CoMon-DAS: A Framework for Efficient and Robust Dynamic Adaptive Streaming over NDN

Muhammad Hassan[*], Hani Salah[†], Mauro Conti[*], Frank H. P. Fitzek[†], Thorsten Strufe[†]
[*]University of Padova, Italy [†]TU Dresden, Germany
last@math.unipd.it first.last@tu-dresden.de

***Abstract*—Implementing DASH, the most popular method for multimedia streaming, over NDN, a potential future Internet architecture, can substantially increase the network bandwidth utilization. However, inherent features of NDN can create new security risks for adaptive multimedia streaming. We propose a novel attack called Bitrate Oscillation Attack (BOA), which adversely exploits NDN's autonomous on-path caching and interest aggregation to unsettle DASH functionality. BOA forces the resolution and quality of video received by the attacked client to oscillate with high frequency and amplitude. Subsequently, we present CoMon-DAS, a framework for lightweight coordination that mitigates BOA and other attacks in NDN. Through extensive simulations, we demonstrate that BOA is very harmful for DAS over NDN, but can be significantly mitigated by CoMon-DAS.**

I. Introduction

The importance of multimedia data is definite in the traffic trends of future Internet. Cisco VNI estimates that multimedia data will consume 82% of the IP traffic till 2022 [1]. In fact, Netflix and YouTube jointly hold 50% of the Internet traffic. Due to this rapid gain in multimedia traffic, challenges are being faced by the network operators to meet the bandwidth requirements of the end users.

In response to the aforementioned challenges, researches have resulted in new networking paradigms that are able to cache the traffic within the network. Named Data Networking (NDN) [2] is one of those paradigms that provide native support for caching at the network layer to ease the bandwidth blockage. NDN is widely accepted as an architecture for the future Internet. It replaces the host-centric communication model with a content-centric approach where routers perform forwarding and caching of data packets. In NDN, client issues an exclusive *interest* packet to directly request a *content* regardless of the provider location. The network then is in charge to find the closest copy of the requested content that satisfies the interest as efficiently as possible. To this end, NDN introduces ubiquitous *in-network caching*, i.e, any node can store a copy of recently received/forwarded contents, and utilize it to satisfy subsequent interests. NDN also provides a native support for multicast by enabling *interest aggregation*, i.e., only the first of multiple closely spaced interests (for the same content) is forwarded by each network element.

The most adopted multimedia streaming method to enhance bandwidth utilization (e.g., adopted by Netflix, YouTube, and HBO) is HTTP based Dynamic Adaptive Streaming (DASH) [3]. It provides a dynamic approach to time-shift control on media requests in response to fluctuating bandwidth conditions experienced by individual users. In particular, DASH strives to adopt the most appropriate resolution via real-time bandwidth measurements in unstable network conditions, to deliver the best possible Quality of Experience (QoE). Taking into account the significance of NDN in reducing bandwidth utilization, and to overwhelm the constraints of multimedia streaming, recently, the research community has investigated the implementation of Dynamic Adaptive Streaming (DAS) over NDN [4]. Several studies, e.g., [5]–[7], have shown that NDN's receiver driven content delivery with in-network caching can significantly enhance the performance of adaptive multimedia streaming with DASH.

Motivated by the importance of addressing security problems of a potential future Internet architecture (i.e., NDN), we study a vulnerability in DASH over NDN. In this vulnerability, two fundamental NDN features, *in-network caching* and *interest aggregation* [2], can be adversely exploited to attack the DASH streaming control system to degrade user perceived QoE. In particular, the adversary is able to force the DASH client to compute false bandwidth estimations during bitrate (i.e., resolution) adaptation process, causing video streaming with highly variable bitrates. The initial results verifying the impact of our proposed Bitrate Oscillation Attack (BOA) on perceived QoE is reported in [8].

In this paper, we broaden the attack scenarios, and subsequently propose a mitigation approach that is both effective and efficient. In particular, NDN's autonomous, on-path cache management initiates enormous cache redundancy, results in sub-optimal caching decisions, and inherits cache-ignorant routing [9]. This makes DAS more challenging in NDN, and makes NDN exposed to new security risks. Therefore, we propose to mitigate BOA based on timely and global knowledge of content access information. This enables DAS to realize network-wide caching goals and cache-aware routing.

Our contribution in this paper is twofold. First, we propose an effective countermeasure to mitigate BOA, called CoMon-DAS. It implements **Co**ordination with lightweight **Mon**itoring for DAS to enable network-wide coordinated caching and cache-aware routing. By this, it aims to reduce bitrate oscillations and cache content redundancy in presence of both BOA and inherent content source variations, thus to enhance the perceived QoE. Second, we evaluate BOA and CoMon-DAS, through an extensive simulation study. Our results show the adverse impact of BOA, as well as the high effectiveness and feasiblity of CoMon-DAS.

978-1-7281-2999-0/19/$31.00 ©2019 IEEE

The rest of the paper is structured as follows. We overview DASH and related work in Section II. Next, Section III describes the aforementioned vulnerability. CoMon-DAS and its evaluation are detailed in sections IV and Section V, respectively. Finally, we conclude the paper in Section VI.

II. Background and Related Work

DASH [3] has been considered as the most widely used on-demand, real-time multimedia streaming method, and is ratified by ISO/IEC as MPEG-DASH. DASH specifies the explanation of multimedia content accessibility and the procedure of by what means it shall be segmented for delivery. In DASH, the media content is provided in various encoded versions, which are further decomposed into segments of specific durations and characteristics. The association between a segment's characteristics (e.g., bitrate, resolution, codec, timeline) and the location is provided to the client by the so-called XML based *Media Presentation Description (MPD)*. The client implements a pull-based mechanism to request each segment individually. Hence, using the information in the MPD file, the DASH client dynamically adopts the highest suitable bitrate by considering the user's current context, i.e., bandwidth fluctuations, preferences, etc. [3]. The strategies for dynamic adaptation are mainly classified into two categories depending on their functionality: (i) Rate Based (*RB*) and (ii) Buffer Based (*BB*). The *RB* strategy makes use of network bandwidth estimation to download the appropriate bitrate, as in Probe and Adapt (PANDA) [10]. The *BB* strategy functions independent from bandwidth estimation. Instead, it selects the bitrate according to the current buffer occupancy of the DASH client, as in Buffer Occupancy based Lyapunov Algorithm (BOLA) [11]. Rate and Buffer based *(R&B)* adaptation algorithm [12] is a stronger coexistence between *RB* and *BB*.

As NDN [2] proved to be a possible replacement of existing Internet architecture, implmenting DASH over NDN has gained significant attention by the esearch community recently. Several studies [4]–[7] have shown NDN's native support for in-network caching and content-oriented delivery as a provision for DASH. For instance, the authors in [7] demonstrate the combination of DASH besides NDN while implementing a proxy provision between HTTP and NDN. The authors in [5] exploit the advantages of NDN by implementing the DASH client as an instinctive NDN interface. These works transform the request and reply messages of HTTP into the corresponding NDN's interest and content messages. In DASH over NDN, MPD lists the NDN names (i.e., URIs) for the segments as an alternative of URLs [5]. In addition, the hierarchical naming structure of NDN explicitly supports DASH's versioning and segmentation.

The starring role of DASH's streaming control system is to adjust the client requests based on the available bitrates and network bandwidth, to deliver a smooth streaming session with high QoE. The work in [5] confirms that DASH over NDN is able to provide enhanced performance in terms of average download bitrate, smooth streaming sessions, and reduced bandwidth requirements. Also, the outcome of [5] demonstrates the effectiveness of in-network caching in the case where multiple clients request similar contents, subsequently showing improved video quality over time. Furthermore, the authors in [6] exhibit the gain of NDN-based dynamic adaptive streaming while using Scalable Video Coding (SVC), proving that integration of the layered data approach with in-network caching increases the performance of bitrate adaptation process and provides smooth playback without stalling. Although these result show that NDN features are advantageous for adaptive multimedia streaming, DASH streaming control system exposes new security vulnerabilities when intersects with NDN's architecture [8]. In this paper, we observe the ineffectual behaviour of the DASH control system and propose a mitigation approach for efficient and robust adaptive streaming over NDN.

III. Vulnerability in DAS over NDN

To illustrate the vulnerability in DAS over NDN, we consider the streaming scenario for *client* (C), where DAS-compatible data (S) is provided by *producer* (P), as illustrated in Figure 1. S includes n equal-length segments, each of them is available in various media encoded bitrates, $b_{(i,j)}$ such as $i < j$. S is publicly available, i.e., consumers can access it through various Access Points (AP). Furthermore, each interest packet (from *Adversary* (*Adv*) or from C) traverses one more NDN routers (R_l). For DAS, both C and P utilize the aforesaid DAS over NDN models [5], and Adv is also aware of DAS.

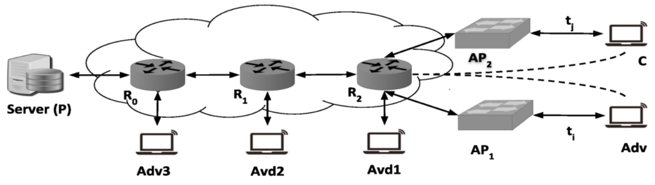

Figure 1: Considered topology

A. Adversary Model

For the adversarial model, we first assume that Adv is attached to at least one on-path router between C and P. Existing geo-locating techniques such as [13] justify the first assumption, which Adv could use to identify the routers closest to the victim and simply connects to the network through them. Secondly, we assume that Adv has prior information about S that C is going to stream in near future. Apart from the preliminary information required to execute the attack subjective to C, various standing practices provision this assumption. For instance, Adv could exploit the timing attacks [14] and probe the initial portions of video, e.g., MPD files, to discover the S which C is going to request. Moreover, in case of identical wireless links, eavesdropping techniques on the exposed traffic traces could also be exploited to analyze the online activities of the victim and predict the video and its source location as well. Lastly, Adv can access S by means of generic ISPs.

B. A Bitrate Oscillation Attack for DAS over NDN

In this section, we describe the proposed Bitrate Oscillation Attack (BOA) in NDN [8]. To trigger oscillations for C while streaming a video file S, Adv requests particular segments of S in advance, as illustrated in Algorithm 1. When issuing requests for S, initially Adv obtains the MPD file containing the list of all available segments (S_n) and various encoded bitrates ($b_{i,j}$) of each S_n [5]. Adv exploits the information presented in the list to request a non-sequential subset of S in a way leading to higher bitrate oscillations for C. Although Adv holds information of S that C will stream, she is not aware of precise bitrates of S requested by C. To attain the maximum adversarial impact, for each selected segment of S, Adv requests all offered bitrates of that segment to guarantee that any variety of that segment requested by C should have prior forwarding state in PIT or cached in an intermediate router. In particular, Adv issues a series of interests following an ascending directive (i.e., $S(n+\gamma)$), where γ is the consecutive gap of persistent or variable length, which is used to generate discontinues requests for S with all offered bitrates (i.e., ($b_{(i,j)}$)) towards P.

Algorithm 1 Adversary algorithm (Adv)

```
procedure SEQUENCE_OF_INTEREST $(S, i, j, \gamma)$
    $MPD \leftarrow Send\_\ requests\_\ to\_\ P$    ▷ $MPD = \{S(n)_{b_{i,j}}\}$
    for $n = 1, n \leqslant N, n + \gamma$ do
        $Content(S(n)_b) \leftarrow Interest(S(n)_b)$
        for $i \leqslant f \leqslant j$ do
            $Content(S(n)_{b_f}) \leftarrow Interest(S(n)_{b_f})$    ▷ $S(n)_{b_i}, \ldots S(n)_{b_j}$ caches on $CS_k$
        end for
    end for
close;
```

Considering a specific segment S_n requested by C, as shown in Figure 1, if Adv has previously requested it from P, a copy of S_n is then available in the CS of R_2, and is returned to C in the round-trip time between R_2 and C. In case the earlier requested S_n (i.e., by Adv) has not yet reached R_2, the interest of C is aggregated at R_2, and later the request (for S_n) will be satisfied by R_2 in the round trip time less than the full one (i.e., between C and P). Lastly, if S_n has not earlier being requested by Adv, the request is forwarded all the way to P. Thus, S_n is delivered in the full round trip time between P and C. Subject to this occurrence, C interprets extremely rapid delivery times for the segments pre-issued by Adv as indicating high network bandwidth availability. In contrast, C interprets longer delivery times for the remaining segments as indicating comparatively less bandwidth availability. This makes DAS adaptation strategy at C frequently switches between highly different bitrates, which translates into QoE degradation.

IV. CoMon-DAS: Coordinated Caching and Cache-Aware Routing for DAS

We aim to mitigate BOA in an effective, yet inexpensive, way. Our experience in defending against distributed attacks in NDN [15]–[17] learned us that effectiveness can be achieved if the attacks are mitigated based on current, network-wide view of attack-related information.

For DAS over NDN, unlike native NDN's autonomous on-path cache management scheme, caching decisions should be made based on network-wide knowledge of content requests, and routing should be aware of cache configurations (i.e., which contents are cached, and in which routers). This is because DAS client estimates the bitrate(s) of the subsequent segment(s) just by considering the measurements of the earlier received segment(s). Also, the producer location keeps on shifting due to content source variations triggered by on-path caching. Categorically, the segment(s) retrieved from various caches results in different measurements as compared to the ones received from the origin producer. If the change in the positions of consecutive segments within a session is too numerous, DAS erroneously adopts higher or lower bitrates for the subsequent segments. This leads to a vulnerability in DAS over NDN, as discussed in Section III-B.

To be robust against BOA and varied content source locations, the network should effectively utilize the global cache capacity, i.e., reduce bitrate oscillations and avoid cached content redundancy, thus efficiently deliver the best possible perceived QoE. The obligation indicates that each network node should have a timely and network-wide level view of cached content information. However, such a solution requires to exchange and process massive amounts of information very frequently.

We propose to address the aforementioned problem by adapting CoMon, our framework for **Co**ordination with lightweight **Mon**itoring. Our choice is influenced by CoMon's demonstrated ability to address two similar problems in NDN: (i) network-wide cache coordination [9] and (ii) mitigation of distributed DoS attacks [15]–[17].

We call our solution CoMon-DAS. We give an overview of the system architecture and monitoring techniques in Subsection IV-A.[1] Next, we describe the defense mechanism in Subsection IV-B.

A. System Architecture and Monitoring Techniques

System architecture: CoMon-DAS is designed to work within a domain network (i.e., autonomous system). As shown in Figure 2, the network includes a Domain Controller (DC) and a set V of routers divided into two groups: (i) NDN Routers (NRs) and (ii) Monitoring Routers (MRs). In the following, we introduce these components and describe how do they work with each others:

1) *Domain Controller (DC)*: This is a (logically) centralized controller. It periodically receives a summary of MRs

[1] For more details about the monitoring techniques, the reader is referred to our previous work [15].

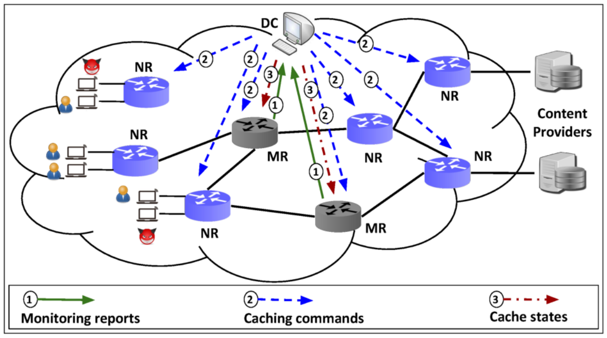

Figure 2: System architecture (adapted from [17]): ”DC” stands for Domain Controller, ”NR” for NDN Router, and ”MR” for Monitoring Router.

observations. The DC aggregates and processes these information. It then commands the routers to perform certain actions accordingly.

2) *NDN Routers (NRs)*: They work similar to standard NDN routers, as described in [2]. However, the routing protocol and the cache replacement algorithm are modified in the NRs.

3) *Monitoring Routers (MRs)*: Each MR, in addition to the routing and caching tasks, persistently monitors the interest packets passing through it. At the end of each observation period, the MR sends a report to the DC summarizing the names of the requested segments along with information about their quality and request statistics. The MRs also receive instructions from the DC, and adapts their routing and caching decisions accordingly.

Monitoring techniques: CoMon-DAS employs a lightweight algorithm called *PRCS* (Placement based on covered Routes and Closeness to Sources) [15] to select the MRs. In principle, PRCS selects a subset $M \subset V$ of routers[2] that jointly maximize routes coverage. At the same time, it gives preference to the routers located close to clients (thus to potential attack sources), so that attacks can be defended at an early stage.

In order to achieve full coverage, which cannot be guaranteed by PRCS alone, CoMon-DAS implements two monitoring techniques: (i) Forward-Till-Be-Monitored (FTBM) and (ii) Monitor-Aware Routing (MAR). FTBM deals with the satisfied interest packets that are not monitored earlier. Its functionality requires to add two flags to the standard interest packet: (i) *satisfied* flag and (ii) *monitored* flag.[3] When a router satisfies an interest packet [4], it sets the *satisfied* field to 1, and then forwards the packet to the closest MR. The designated MR, in turn, records the packet information, and drops it afterwards.

MAR enforces each interest packet, thus the corresponding data packet, to pass through an MR. This requires to modify the original routing protocol. More specifically, each interest packet is first forwarded to an MR (e.g., the closest one). The designated MR then forwards the packet to its original destination.

B. Defense Mechanism

Our defense mechanism is composed of three techniques: (i) selection of cached contents, (ii) traffic shaping, and (iii) dynamic prefetching. In the following, we describe these techniques and explain how they together enable to mitigate BOA effectively.

Selection of cached contents: At the end of each observation period, the DC uses the reports received from the MRs to identify $|V| \times c$ segments to be cached in the network during the next observation window, where c denotes the router’s cache capacity. Instead of assigning the segments to the routers randomly, which results in low routing performance [9], the DC considers the topological properties of the routers. In particular, it implements the allocation algorithm that we proposed in [9], which is based on the betweenness centrality (BC).

The DC makes the caching decisions corresponding to network-wide caching goals co-related with DAS requirements. In particular, a segment is cached only if it is favorable to perceived QoE. To this end, the DC exploits the aggregated report comprising the list of currently requested segments into the network, denoted as L. Using the naming information[5] from URI structure [5], the DC determines the sequence and characteristics of the segments being requested in a given period of time δ_t. Then, for each request in L, the DC checks whether a request for a subsequent segment with higher bitrate(s) also exists in the list. If so, the DC instructs to cache the segments of both requests. Otherwise, the request is left untreated for caching.

The procedure is outlined in the first part of Algorithm 2 (lines 5 – 11). In particular, the DC checks for each request $S(r)_{b_i}$, whether the subsequent request with higher bitrate (i.e., $S(r+1)_{b_k}$) exists in L, where $i < k \leq j$. If so, both $S(r)_{b_i}$ and $S(r+1)_{b_k}$ are cached. Otherwise, the segment $S(r)_{b_i}$ is not cached. The cache assignments are then sent to the routers according to their BC values (line 12).

Traffic shaping: The DC informs the MRs about caching decisions, i.e., the *segment-to-router* assignments. The MRs use these information every time they receive an interest packet not monitored before (i.e., *monitored* $= 0$) to check whether the requested segment is cached inside the network or not.

Each MR, when receiving an interest packet not monitored before, checks whether both (i) the requested segment and (ii) its successor with a higher quality are cached. If so, the MR reroutes the interest packet towards the cached copy. Otherwise, the original route is preserved. Next, the MR sets the *monitored* flag to 1 in order to avoid repeating the aforementioned checking step by other MRs. This way, CoMon-DAS enables *cache-aware routing*.

[2] M is predetermined; $|M| \ll |V|$

[3] The two flags neither significantly change the packet structure (only one bit each) nor breach the standard protocol.

[4] The packet matches either a PIT entry or a cached segment.

[5] The naming information are inferred from the MPD file as part of the setup phase of DASH stream.

With the above described caching and routing strategies, a request from a DAS client results in a *cache-hit* only if the CS is capable to compete the DAS bandwidth estimation requirements for the subsequent segment. This is driven by the fact that DAS clients utilize the bandwidth estimation of the retrieved segment while processing the bitrate selection process of the subsequent segment. This way, CoMon-DAS shapes the requests to avoid bitrate oscillations triggered by BOA and varying content source locations.

Dynamic prefetching: To improve the QoE further, CoMon-DAS additionally takes proactive measures while effectively utilizing the global cache capacity. Specifically, based on the available timely content request information, the DC predicts the contents to be requested by the DAS clients in the near future.

As outlined in the second part of Algorithm 2 (lines 14 – 17), the DC decides to prefetch a segment $S(r+1)_{b_k}$ if a request of $S(r)_{b_i}$ already exists in the requested content list, where $i < k \leq j$. Subsequently, the DC assigns the prefetching tasks to the routers that are located close to the clients[6].

Algorithm 2 Defense mechanism against BOA

```
 1: procedure FUNCTIONALITY_OF_DC ($L, \delta_t, S(n)b, i, j$)
 2:     $L \leftarrow \delta_t$
 3:     $\{S(n)_{b_{i,j}}\} \leftarrow MPD$
 4:     Check requested content
 5:     for Each request $S(r)_{b_i}$ do
 6:         if $S(r+1)_{b_k} == L$ then                $\triangleright\ i < k \leqslant j$
 7:             Cache the content $S(r)_{b_i}$ and $S(r+1)_{b_k}$
 8:         else
 9:             Do not cache $S(r)_{b_i}$
10:         end if
11:     end for
12:     $NR_{BC} \leftarrow$ Assign cache configurations
13:     Prefetching after $\delta_t$
14:     if $S(r)_{b_i} == L$ then
15:         $Content(S(r+1)_{b_k}) \leftarrow Interest(S(r+1)_{b_k})$ $\triangleright\ i < k \leqslant j$
16:         $NR_{BC} \leftarrow$ Instructions
17:     end if
18: close;
```

V. Evaluation and Result Analysis

In this section, we evaluate the BOA attack as well as CoMon-DAS. The evaluation is established on simulations. We describe our experimental setup and evaluation metrics in Subsection V-A. After that, we discuss the impact of the attack and the effectiveness of CoMon-DAS in Subsection V-B and Subsection V-C, respectively.

A. Setup and Evaluation Metrics

We implemented BOA and CoMon-DAS over AMuSt-ndnSIM [12], an adaptive multimedia streaming framework over ndnSIM. AMuSt-ndnSIM delivers a set of applications grounded on the official DASH standard [18].

We simulated with a real ISP toplogy measured by the Rocketfuel project [19]. Specifically, we implemented the AS 3967 topology (79 nodes and 147 bidirectional edges), with a single producer (P), single DASH client (C), three adversaries ($Adv1$, $Adv2$, $Adv3$). P hosts a real-time existing MPEG-DASH video (*BigBuckBunny* movie), both AVC-encoded [20] and SVC-encoded [21]. We separately simulated using three different DASH adaptation strategies: (i) Rate-Based (*RB*) [12], (ii) Buffer-Based (*BB*) [22], and (iii) Rate-Buffer-based (*R&B*) [12]. Other simulation parameters and their values are summarized in Table I.

TABLE I: Simulation parameters

Parameters	Value
No. of video segments	250
Video period (sec.)	240
Available bitrates (AVC)	20
Layers of quality (SVC)	4
Duration per segment (sec.)	2
Delay between P and edge router (µs)	200
Consecutive gap α	2
Fragment size s (byte)	1449
MTU (byte)	1449
Drop Tail Queue (max. packets)	20
Default cache policy	LRU
Startup delay (sec.)	0.1
Max. buffered time (sec.)	30

Following [23], [24], we evaluate the effectiveness of BOA and CoMon-DAS using the following two metrics:

i) *Oscillation frequency:* The frequency of video quality oscillations within a streaming session.

ii) *Average oscillation magnitude:* The average amplitude of the video quality oscillations within a streaming session.

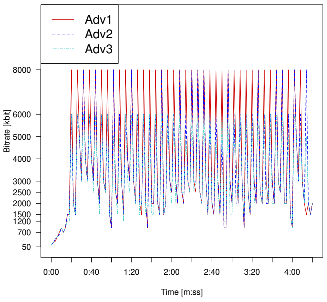

Figure 3:
DAS RB (AVC) for multiple Adv(s)

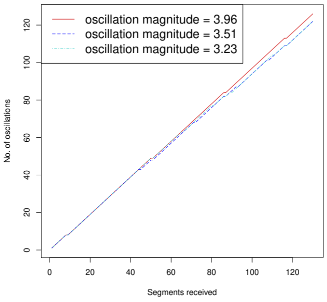

Figure 4:
Oscillation frequency to various adversarial locations

B. Attack Impact

Figures 3 and 4 show the adversarial impact of BOA. Specifically, the two figures show an increase in the average oscillation magnitude and the oscillation frequency, respectively,

[6] The routers are selected by the algorithm that we proposed in [9].

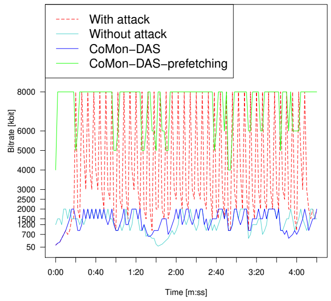

Figure 5:
DAS applying RB (AVC)

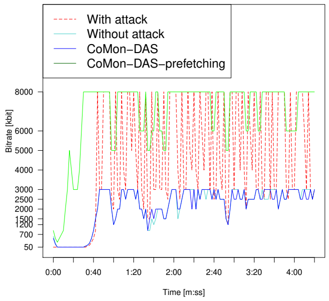

Figure 6:
DAS applying R&B (AVC)

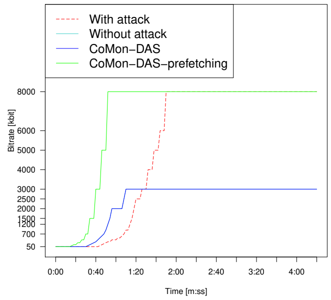

Figure 7:
DAS applying BB (AVC)

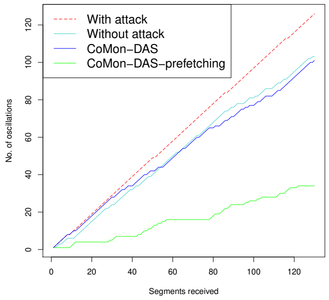

Figure 8:
Oscillation frequency RB (AVC)

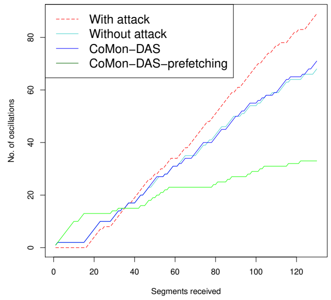

Figure 9:
Oscillation frequency R&B (AVC)

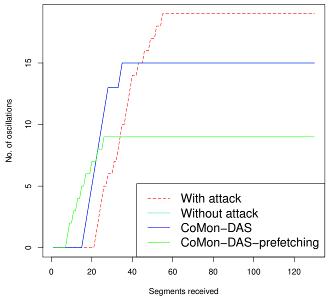

Figure 10:
Oscillation frequency BB (AVC)

encountered by C for all the cases. These results represent various adversarial locations. Due to space limitations, the rest of the results represent only the case of $Adv1$. However, the results of $Adv2$ and $Adv3$ are very similar, and lead to the same conclusions.

In the case of AVC, Figures 5, 6, and 7 report the bitrate request patterns of the DAS client, with and without BOA, applying RB, R&B, and BB adaptation logic, respectively.

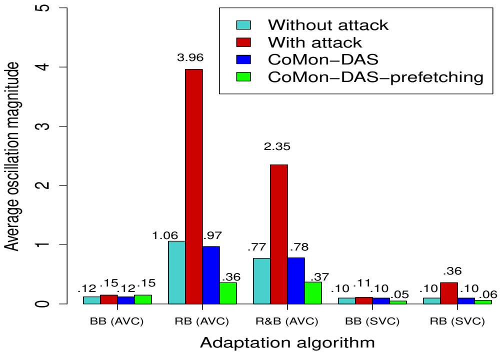

Figure 11:
Average oscillation magnitude

The corresponding oscillation frequency results are plotted in Figures 8, 9, and 10. The results show that frequency of bitrate oscillation in the three adaptation logic increases by about 25%. In addition, as can be seen in Figure 11, BOA also massively increases the average oscillation magnitude of bitrate fluctuations. Specifically, the average increases by up to 273%, 157%, and 25%, in RB, R&B and BB, respectively.

In SVC, BOA increases the oscillation frequency for RB, as can be seen in Figures 12 and 13, by about 275%. Also, Figure 11 depicts an increase in the oscillation magnitude by about 260%, which translates into significant QoE degradation.

The results also reveal that BB in SVC is resilient to BOA (see Figures 14 and 15). This is because the buffer capacity is resilient to short term bandwidth fluctuations. However, the use of BB with SVC still remains an open question due to buffer size management issues in small devices such as smart phones.

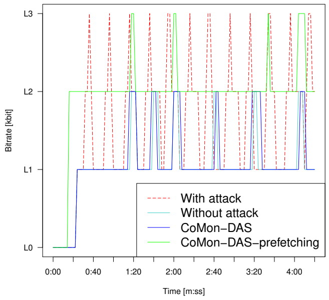

Figure 12:
DAS applying RB (SVC)

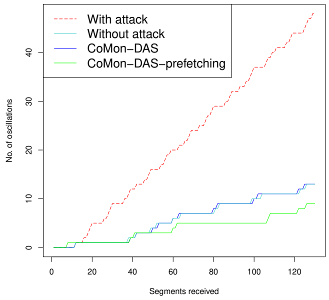

Figure 13:
Oscillation frequency RB (SVC)

C. CoMoN-DAS Effectiveness

We discuss here the effectiveness of CoMon-DAS with and without dynamic perfecting. Figures 5, 6, and 7 report the performance of CoMon-DAS in presence of BOA, in AVC using RB, R&B, and BB, respectively. The results highlight

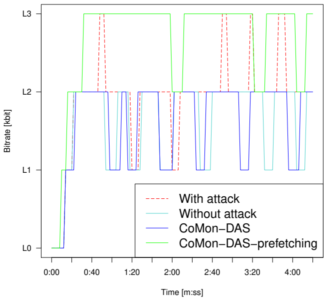

Figure 14:
DAS applying BB (SVC)

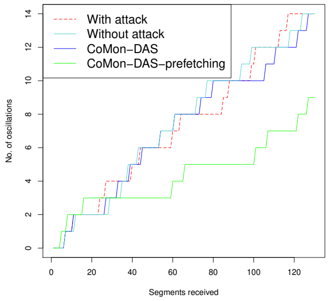

Figure 15:
Oscillation frequency BB (SVC)

the phenomena where the victims follow the similar bitrate request pattern as if no attack exists. This is because malicious segments requested by Adv are not allowed to cache, thus C's requests follow the original path. Furthermore, the results show that dynamic prefetching of segments provides higher bitrates to C, because sequential segments of higher bitrates are retrieved from caches.

Figures 8, 9, and 10 exhibit that the oscillation frequency in AVC with RB, R&B, and BB, respectively, decreases remarkably in presence of BOA. In addition, CoMon-DAS-prefetching enhances QoE by reducing bitrate fluctuations as compared to base-line scenario, by about 70%, 57%, and 46% for RB, R&B, and BB, respectively.

The results depicted in Figure 11 confirm the robustness of CoMon-DAS. Specifically, they show a low average oscillation magnitude in presence of BOA. Also, an additional improvement of 50% to 70% less oscillation magnitude, for RB and R&B, respectively, is achieved by dynamic prefetching.

For SVC, CoMon-DAS results in a reduced bitrate oscillations frequency (see Figures 12 and 13) and in a reduced average oscillation magnitude (see Figure 11) for RB. We can also see that BB takes advantage of BOA. However, with dynamic prefetching of CoMon-DAS, the client can achieve even better resolution, with approximately 50% less bitrate oscillation frequency and magnitude (see Figures 14 and 15).

All over all, the results report that CoMon-DAS is able to maintain the perceived QoE in presence of BOA. That is, the client would experience higher bitrates with reduced oscillation frequency and magnitude.

VI. Conclusion

We have shown how an adversary can exploit features of NDN to launch the so-called Bitrate Oscillation Attack (BOA), and explained how BOA can significantly degrade the performance of dynamic video steaming over NDN. Subsequently, we have proposed a countermeasure, called CoMon-DAS, to protect the network against BOA. CoMon-DAS alleviates the effects of adversaries by enabling network-wide coordinated caching and cache-aware routing. We have extensively simulated BOA and CoMon-DAS in realistic settings. The results show that: (i) high frequency of bitrate switching increases the annoyance factor in spatial dimension, (ii) high amplitude of oscillations decreases the video quality, and (iii) CoMon-DAS can significantly enhance the perceived QoE in presence of varied content source locations and attacks.

In the future, we plan to investigate in the direction of making caching decisions more intelligent. For example, by enabling the network to convert high-resolution segments to lower ones (at line rate), it becomes possible to cache the highest available qualities and dynamically transforming them (if necessary) to provide the best possible QoE.

Acknowledgment

This work was supported by Deutsche Telekom AG as part of a research and development project with TU Dresden, Deutsche Telekom Chair of Communication Networks.

References

[1] "White paper: Cisco Visual Networking Index (VNI): Forecast and methodology, 2017–2022."
[2] L. Zhang *et al.*, "Named data networking," *ACM SIGCOMM CCR*, 2014.
[3] S. Lederer *et al.*, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, 2012.
[4] M. F. Majeed *et al.*, "Multimedia streaming in information-centric networking: A survey and future perspectives," *Computer Networks*, 2017.
[5] S. Lederer *et al.*, "Adaptive multimedia streaming in information-centric networks," *IEEE Network*, 2014.
[6] S. Petrangeli *et al.*, "Towards svc-based adaptive streaming in information centric networks," in *IEEE ICMEW*, 2015.
[7] A. Detti *et al.*, "Offloading cellular networks with information-centric networking: The case of video streaming," in *IEEE WoWMoM*, 2012.
[8] M. Conti *et al.*, "Qoe degradation attack in dynamic adaptive streaming over icn," in *IEEE WoWMoM*, 2018.
[9] H. Salah and T. Strufe, "Comon: An Architecture for Coordinated Caching and Cache-aware Routing in CCN," in *IEEE CCNC*, 2015.
[10] Z. Li *et al.*, "Probe and adapt: Rate adaptation for http video streaming at scale," *IEEE JSAC*, 2014.
[11] K. Spiteri *et al.*, "Bola: Near-optimal bitrate adaptation for online videos," in *IEEE INFOCOM*, 2016.
[12] C. Kreuzberger *et al.*, "AMuSt Framework - Adaptive Multimedia Streaming Simulation Framework for ns-3 and ndnSIM," 2016.
[13] A. Compagno *et al.*, *Violating Consumer Anonymity: Geo-Locating Nodes in Named Data Networking*. Springer International Publishing, 2015.
[14] G. Acs *et al.*, "Privacy-aware caching in information-centric networking," *IEEE Transactions on Dependable and Secure Computing*, 2017.
[15] H. Salah *et al.*, "Coordination supports security: A new defence mechanism against interest flooding in NDN," in *IEEE LCN*, 2015.
[16] H. Salah and T. Strufe, "Evaluating and mitigating a collusive version of the interest flooding attack in NDN," in *IEEE ISCC*, 2016.
[17] H. Salah *et al.*, "CoMon++: Preventing Cache Pollution in NDN Efficiently and Effectively," in *IEEE LCN*, 2017.
[18] S. Mastorakis *et al.*, "ndnSIM 2: An updated NDN simulator for NS-3," Technical Report, 2016.
[19] N. Spring *et al.*, "Measuring isp topologies with rocketfuel," *IEEE/ACM Trans. Netw.*, 2004.
[20] S. Lederer *et al.*, "Dynamic Adaptive Streaming over HTTP Dataset," in *ACM SIGMM MMSys*, 2012.
[21] C. Kreuzberger *et al.*, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over http," in *ACM Multimedia Systems Conference*, 2015.
[22] C. Sieber *et al.*, "Implementation and user-centric comparison of a novel adaptation logic for dash with svc," in *IFIP/IEEE IM*, 2013.
[23] Y. Liu *et al.*, "User experience modeling for dash video," in *International Packet Video Workshop*, 2013.
[24] P. Ni *et al.*, "Flicker effects in adaptive video streaming to handheld devices," in *ACM International Conference on Multimedia*, 2011.