Dynamic Adaptive Streaming over CCN: A Caching and Overhead Analysis

Yaning Liu, Joost Geurts and Jean-Charles Point
JCP-Consult, France
firstname.lastname@jcp-consult.com

Stefan Lederer, Benjamin Rainer, Christopher Müller, Christian Timmerer, and Hermann Hellwagner
Alpen-Adria-Universität (AAU) Klagenfurt, Austria
firstname.lastname@itec.aau.at

***Abstract*—In this paper, we present our implementation and evaluation of Dynamic Adaptive Streaming over Content centric networking (DASC) which implements MPEG Dynamic Adaptive Streaming over HTTP (DASH) utilizing a Content Centric Networking (CCN) naming scheme to identify content segments in a CCN network. In particular, video segments formatted according to MPEG-DASH are available in different quality levels but instead of HTTP, CCN is used for referencing and delivery. Based on the conditions of the network, the DASC client issues interests for segments achieving the best throughput. Due to segment caching within the network, subsequent requests for the same content can be served quicker. As a result, the quality of the video a user receives progressively improves, effectively overcoming bottlenecks in the network. We present two sets of experiments to evaluate the performance of DASC showing that throughput indeed improves. However, the generated overhead is relatively large and the adaptation strategy used for DASH that assumes an end-to-end connection could be revised for the hop-by-hop architecture of CCN.**

***Index Terms*—Content Centric Networking, Dynamic Adaptive Streaming, HTTP Video Streaming, MPEG-DASH.**

I. Introduction

Video is expected to account for more than 55% of all Internet traffic [1] in 2016. According to comScore, the top two rankings in online video market in July 2012 in the US are the social media Web sites YouTube and Facebook [2].

In general, it is challenging to stream video over the current Internet architecture as this service is latency-sensitive and bandwidth-hungry. First, intermediate nodes such as Internet Protocol (IP) routers are designed to discard packets after they have been forwarded. This leads to a waste of network bandwidth and network congestion when the content is requested simultaneously and frequently respectively. Second, as IP provides a best-effort delivery, Quality of Service (QoS) is likely to degrade when the distance between the user and content provider is huge.

Many overlay solutions such as Content Delivery Network (CDN) and Peer-to-Peer (P2P) networks have been proposed and successfully deployed. However, these systems also have certain drawbacks [3]. CDN solutions are relatively complex to implement and the management is costly. On the other hand, CDNs are scalable but it is not easy to achieve a rapid network expansion due to the administrative and operation costs. P2P networks allow end users to share their resources (e.g. bandwidth and storage) with a low cost. However, P2P introduces more duplicated traffic and causes various inefficiencies in resource distribution and network usage. P2P has been criticized by network operators [4] since the popularity of P2P applications causes increased network traffic.

Content Centric Networking (CCN) [5] is a novel Internet architecture aiming to address the inherent inefficiencies in content delivery. Instead of referring to the physical location of data, CCN identifies content by its name. As a result, content can be stored anywhere in the network realizing in-network caching and allowing for efficient content distribution. Although the long-term goal of CCN is to replace IP-based network, it is also deployable as an overlay on IP networks to enable an smooth transition.

In addition to inefficiency in content distribution, QoS also decreases due to network bottlenecks that may be caused by heterogeneous network connections (e.g., WiFi, 3G, and ADSL), or difficult network conditions (i.e., large round trip times and significant bandwidth fluctuations). Dynamic Adaptive Streaming over HTTP (DASH) is an emerging standard for adaptive streaming that was recently ratified by ISO/IEC MPEG to overcome these problems [6, 7]. DASH is agnostic to the used codecs and supports different transport formats such as the ISO Base Media File Format (ISOBMFF) [8]. It divides the media content into segments of different bit rates, resolutions, etc. and each segment is identified by an HTTP-URL. The relationship between the different versions is described in a so-called Media Presentation Description (MPD). Based on the MPD, a DASH client is able to download the most appropriate segment satisfying the users' context, i.e., bandwidth conditions, preferences, etc.

In this paper we present our implementation and evaluation of DASH over CCN (DASC). We will demonstrate that the throughput progressively converges to allow for high quality video overcoming bottlenecks in the network. The remainder of the paper is organized as follows. Section II briefly highlights related work. We describe the system design of DASC in Section III. Section IV shows the experimental results and discusses the pros and cons of DASC. Finally, Section V concludes the paper including future work.

II. Related Work

The first concepts similar to Content Centric Networking are referred to as Named Data Networking and have been

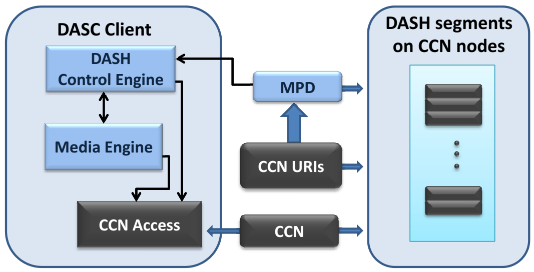


Fig. 1. System Architecture of DASC.

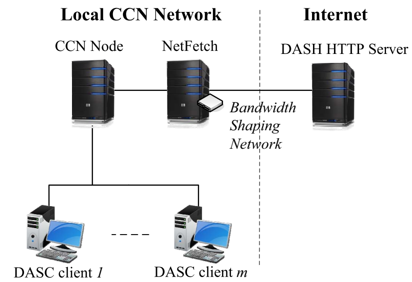


Fig. 2. An Integrated Architecture of DASH and DASC.

proposed back in the 1970s [9]. There have been recently new research interests on emerging future Internet architectures, such as Cache and Forward Network (CNF) [10], NetInf [11], and CCN [5]. Their objectives are to explore new network architectures and protocols to support future content-oriented services. Among these architectures, CCN gets a lot of attention due to prototype implementations like [12].

Several applications over CCN [13]–[15] have been recently designed and implemented. For example, [13] implemented and evaluated voice over IP for CCN. The authors show that the concept of CCN is applicable to real-time multimedia transmission and that the computational overhead introduced by encryption and signing of the content can be neglected. The Audio Conference Tool (ACT) [14] is another real-time service on top of CCN. Authors denote that the design of ACT over CCN provides improved scalability and better security.

Time-shifted TV has been investigated in CCN [16]. A cooperative caching strategy is designed to reduce the cross-domain traffic among Internet service provider by means of the in-network caching and evaluated by simulation. Most closely related to ours, is the work about Apple's HTTP Live Streaming (HLS) over CCN proposed in [15]. Authors showed the caching efficiency of CCN by leveraging an HTTP proxy-based approach. In contrast to that paper, our work focuses on the implementation and evaluation of a CCN video delivery system with dynamic adaptive behavior. To the best of our knowledge, this work is the first prototype implementation for dynamic adaptive video streaming over CCN.

III. System Design: DASC

In this section we present the DASC system [1] that is designed to enable DASH over CCN. Since CCN is at an early stage of development, it is important for DASC to be compatible with the current Internet to account for content that cannot be satisfied within the CCN network.

Figure 1 presents the DASC architecture. The original DASH modules are shown in light blue and the components adapting with respect to the traditional DASH architecture are highlighted in dark gray. In our architecture, DASC provides a native CCN interface to DASH and the CCN naming scheme to denote segments in the MPD.

[1] DASH over CCN patches for the VLC player have been released on DASH research website: http://dash.itec.aau.at

Although DASH is primarily designed for HTTP, its modular design allows HTTP to be substituted by CCN as in CCN content names are represented as Uniform Resource Identifiers (URIs) whereas HTTP uses Uniform Resource Locators (URLs). Figure 3 presents an example showing the CCN URI of a DASC segment compared to the HTTP URL of a DASH segment. The naming scheme of the CCN URIs for DASC segments is compatible to [5] and simply reflects the HTTP URL of a DASH segment prefixed by `ccnx:/DASC` where `ccnx:` identifies the CCN URI scheme and `DASC` indicates a DASC segment (which is also available by HTTP in case the segment is not available within the CCN network).

The name for a DASH segment

http://www-itec.uni-klu.ac.at/BigBuckBunny/bunny_2s/bunny_2s_150kbit/bunny_2s1.m4s

The name for a DASC segment

ccnx:/DASC/http/www-itec.uni-klu.ac.at/BigBuckBunny/bunny_2s/bunny_2s_150kbit/bunny_2s1.m4s

Fig. 3. Example DASH Segment Name and its CCN Name.

We implemented the DASC prototype on top of the DASH VLC plugin [17] using the CCNx implementation [12]. Figure 2 shows an integrated architecture for DASC which allows the retrieval of video segments via both CCN and HTTP. The local network presented on the left deploys a CCN network. The *NetFetch* component, which is provided by CCNx, listens for unsatisfied CCN interests message and turns them into HTTP GET messages. Once the requested video segment is retrieved on the Internet via HTTP, *NetFetch* packages it into a CCN content object that satisfies the pending interest. The content object (i.e., video segment) is cached in the CCN network so that a subsequent request for the same video segment can be satisfied immediately.

IV. Evaluation

In this section we describe the evaluation of DASC in two sets of experiments. In the first set of experiments, we investigate the performance and caching of CCN in the context of dynamic and adaptive streaming. In the second set, we analyze the protocol overhead of DASC compared to DASH.

For the DASH server we use an Apache HTTP Server that provides DASH content in multiple bitrates. We make use of the DASH dataset [18], notably the Big Buck Bunny

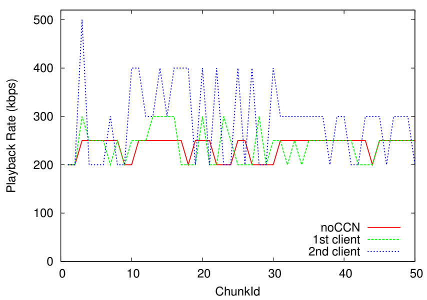


Fig. 4. Comparison of playback bitrates for DASH/DASC streaming

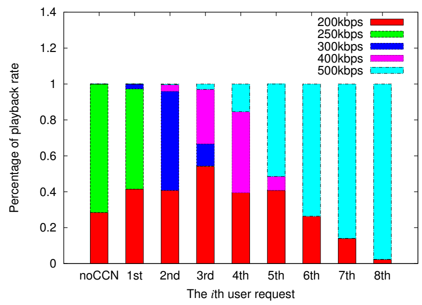


Fig. 5. Comparison of percentage of playback bitrates for DASH/DASC clients

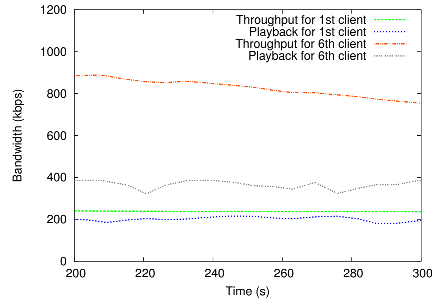


Fig. 6. Estimated throughput by DASH logic *v.s* actual playback bitrates for DASC clients.

sequence [19]. The used content bitrates differ per experiments and are specified in the respective section.

For the DASH client we used our DASH VLC plugin [17]. The DASC client is also based on our VLC plugin, but modified to retrieve content via the local ccnd daemon of CCNx [12]. For the bandwidth shaping the Linux traffic control system (tc) is used in combination with a hierarchical token bucket (htb) packet scheduler. The produced traffic is analyzed using Wireshark [20].

A. Performance and Caching Evaluation

1) Methodology: We run this experiment using the CCNx test-bed described in Figure 2. The bandwidth between the DASH server on the Internet and the NetFetch component on the local network is limited to 250kbps. The connections within the CCN network are set to 100Mbps. As test content we use the Big Buck Bunny sequence with a segment size of two seconds, a resolution of 480x360 pixels, and five representations: 200kbps, 250kbps, 300kbps, 400kbps and 500kbps. As the objective of this experiment is to understand the impact of a CCN cache on the DASH performance, we simply allow DASC clients to retrieve the movie in sequence (i.e. no simultaneous requests). In total, we ran eight DASC clients requesting the same movie after each other.

2) Performance of DASC Streaming: Whenever a DASC client requests a segment that has been already requested by a neighboring client in CCN, this segment will be served from nearby CCN caches rather than from the origin node. However, this may impact the DASH adaptation logic which is based on measuring the available bandwidth, since the request may be responded either by the origin node or by a nearby cache. In order to understand this behavior, we evaluate the performance of DASC and show the results as follows.

Figure 4 shows the playback rate of DASH streaming, and DASC streaming for the first and second client respectively. Figure 5 presents a global comparison of the playback rates for all DASH/DASC clients. Comparing playback rates of DASH client and the first DASC client from Figure 4 and Figure 5, DASH client requests more segments with 250kbps whereas the first DASC client requests a few segments with 300kbps. The second DASC client requests segments with higher quality (even a few segments with 500kbps) and the eighth DASC client can get almost all segments with highest quality. Thanks to the in-network caches, CCN allows subsequent DASC clients to retrieve the video with a high bitrate even though the bandwidth – between client and the origin node – is lower than the actual bitrate. When the number of clients requesting the video is high enough, video segments with the best quality will be all cached on the nearby CCN nodes, which allows the following DASC clients to achieve a higher throughput.

However, some DASC clients (such as the 3rd and the 5th) request a big amount of segments with the lowest quality, i.e, 200kbps. This is mainly contributed to the strategy of the DASH adaptation logic. That is, when the buffer underruns a certain threshold, the adaptation logic decides to retrieve the lowest quality of the video to maintain a smooth playback. As only a subset of the segments is available in nearby CCN nodes, DASC clients may retrieve segments also from the origin node. The DASH adaptation logic consequently gets mixed throughput measurements resulting of two network paths: (a) between DASC clients and the origin node; (b) between DASC clients and nearby CCN nodes. Following the DASH adaptation logic, DASC clients demand segments with high quality. However, the buffer cannot be filled on time if the requested segments are not in nearby CCN nodes. In this case, DASC clients have buffer underruns resulting in stalls and re-buffering events. The cause of this problem is a misunderstanding of network conditions by the DASH adaptation logic. We observe that latter arrived clients (from the 2nd to 7th client) suffer from stalls. As soon as chunks with best quality have been cached on nearby nodes, the following clients (e.g the 8th client) will not have the problem of stall.

	200kbps	250kbps	300kbps	400kbps	500kbps
DASH client	0.284	0.716	0	0	0
1st DASC client	0.417	0.557	0.026	0	0
2nd DASC client	0.657	0.557	0.570	0.037	0.003
3rd DASC client	0.827	0.557	0.610	0.317	0.030
4th DASC client	0.893	0.557	0.610	0.653	0.170
5th DASC client	0.943	0.557	0.610	0.680	0.563
6th DASC client	0.950	0.557	0.610	0.680	0.873
7th DASC client	0.953	0.557	0.610	0.680	0.977
8th DASC client	0.953	0.557	0.610	0.680	0.990

TABLE I
RATIO OF CACHED SEGMENTS ON CCN ROUTER

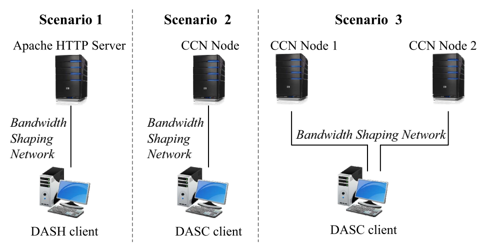


Fig. 7. Experiment Setup for Overhead Analysis.

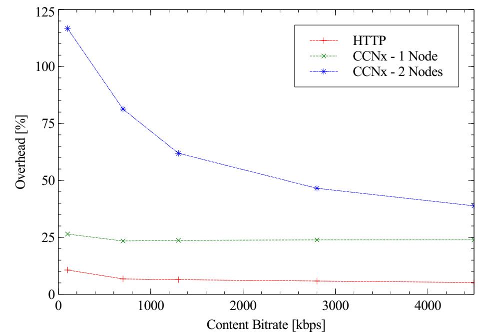


Fig. 8. Protocol Overhead Analysis.

This can be explained by Figure 6 that compares the throughput estimated by the DASH adaptation logic to the actual playback rate of DASC clients. Since for the first arrival client no content has been cached, the estimation is quite accurate as the estimated throughput does not differ much from the actual playback rate (this situation compares to classical DASH). Whereas, for the sixth client most content are available from the cache resulting in a high estimated throughput. However, throughput drops significantly for the part of the video that is not yet available in the cache. The DASH adaptation logic cannot distinguish between these cases resulting in a measured throughput that is higher than actually achieved.

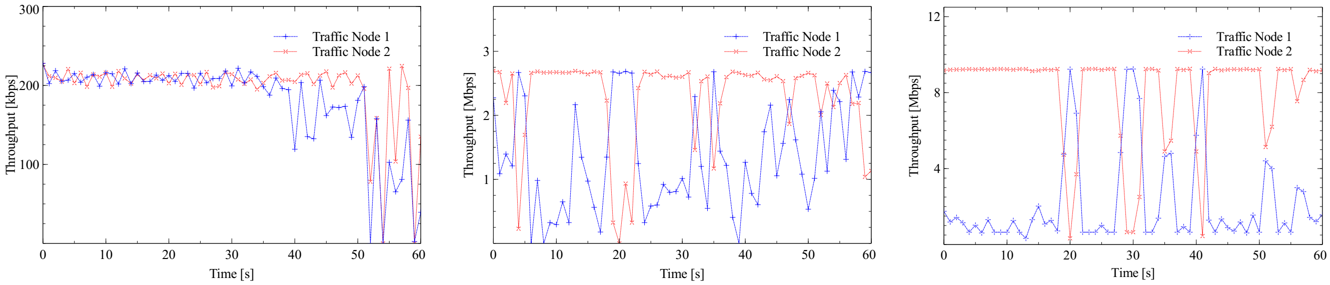
3) Content Caching on CCN Nodes: We also investigate the ratio of cached segments on CCN nodes in Table I with the first column identifying the sequence number of DASC clients and the first row showing the playback rates of cached segments on the CCN node. This result shows that the more DASC clients are requesting the same content, the more segments with higher playback rate will become available on the CCN nodes. In the end, segments with lowest and highest quality will be approximately all cached whereas the cache ratio of the medium quality remains constant according to the strategy of the used DASH adaptation logic.

B. Overhead Evaluation

1) Methodology: In the following we evaluate the protocol overhead produced by DASH (Scenario 1), DASC (Scenario 2), and DASC in case the content is retrieved from multiple CCN nodes in parallel (Scenario 3). For the three scenarios we used three different evaluation setups as depicted in Figure 7: *Scenario 1* presents a traditional DASH scenario using HTTP. *Scenario 2* uses a CCN node with a repository containing the same DASH content as used in Scenario 1. *Scenario 3* represents a scenario where the same content is cached in two distinct CCN nodes, both leveraging the same repository as in Scenario 2. Consequently, requested interest messages for this content may be satisfied by both nodes. In all three scenarios a bandwidth shaping network is used to connect the network nodes and limit the network throughput which is always set to twice the content bitrate. For example, the bandwidth is set to 200 kbps for the 100kbps DASH representation. Each scenario uses the Big Buck Bunny sequence with a two second segment length and representations of 100, 700, 1300, 2800 and 4500 kbps. Furthermore, both DASH and DASC maintain a maximum buffer size of 30 seconds, but player does not need to fill this buffer before starting to decode and render frames, it starts decoding as soon as the first data is downloaded.

2) Results: Figure 8 depicts the protocol overhead for the three scenarios. As expected, the overhead of DASH is relatively low, starting at 10.7 % for the representation with 100 kbps and going down to 5.2 % for the representation with 4500 kbps. Similarly, DASC maintains a constant overhead of around 24 % on average but higher as for DASH due to the larger header of CCN compared to HTTP. Interestingly, for small bitrates the overhead in both cases is slightly higher which is caused by small segments sizes. When using two CCN nodes, the protocol overhead from both nodes is shown. For example, when issuing the first interest message for a DASH segment, the ccnd daemon has to send an interest to all associated faces (i.e., 3G and WiFi in parallel) as it does not know the status of the nodes which can satisfy the interest. Furthermore, for representations with a low bitrate such as 100 kbps, most of the data is retrieved from both nodes resulting in a total overhead of nearly 117 % due to the small segment size. This overhead is reduced for larger segments, e.g., about 81 % for the representation with 700 kbps and 39 % for the representation with 4500 kbps.

This effect can be further observed by investigating the throughput of the two CCN nodes. Figure 9(a) shows the first 60 seconds of the streaming session in Scenario 3 for the representation with 100 kbps. Due to small segment sizes both nodes show an almost equal throughput until the clients' buffer reaches its maximum (>30 seconds). Note that this introduces redundancy as the same segments are downloaded twice over both links. Once the client starts requesting only those segments that are actually needed to maintain the buffer level, the throughput changes as the local ccnd daemon can concentrate on one of the available nodes. For representations with higher bitrates (e.g., 1300 kbps) and, thus, larger segment sizes as shown in Figure 9(b), a different behavior can be observed showing a kind of handover between the throughput of the two CCN nodes. This behavior is further noticeable for even larger segment size as depicted in Figure 9(c) for the



(a) 100 kbit representation (b) 1300 kbps representation (c) 4500 kbps representation

Fig. 9. Results for Scenario 3 using Different Representations.

representation of 4500 kbps. Interestingly, this is an effect of the CCN strategy layer implementation which is also described in [5].

V. Conclusions and Future Work

In this paper we presented our implementation and evaluation of DASC, a dynamic adaptive streaming system that implements DASH over CCN. We showed that DASH can be adapted relatively straightforward to a CCN environment taking advantage of the caching features offered by CCN. We also showed that DASC remains compatible with legacy HTTP infrastructures to account for the case that requests cannot be satisfied within the CCN network.

In order to validate our implementation, we performed two sets of experiments: the first set of experiments shows that the throughput of DASC progressively converges to the best available quality, effectively overcoming bottlenecks in the network. However, our current DASH adaptation logic assumes an end-to-end connection which is less effective in a hop-by-hop architecture used by CCN. Therefore, we believe that the throughput can be further improved which is subject to future work, e.g., by taking into account the CCN network architecture in the DASH adaptation logic.

The second set of experiments shows that the protocol overhead of DASC is higher compared to DASH. However, DASC is more efficient for larger segment sizes, specifically in scenarios where segments are retrieved from multiple CCN nodes in parallel (e.g., a mobile device with 3G and WiFi used in parallel). Additionally, redundancy is decreased when using larger segment sizes thanks to the current implementation of the strategy layer. On the other hand, larger segment sizes may affect the delay which will be studied in the future.

Finally, we intend to extend our evaluations of DASC in a large scale test-bed, such as PlanetLab, to asses its performance in case of massive simultaneous requests under heterogeneous network conditions. We also plan to test DASC in mobile environments.

VI. Acknowledgements

This work was supported in part by the EC in the context of the projects of SocialSensor (FP7-287975) and COMBO (FP7-317762), and partly performed in the Lakeside Labs research cluster at AAU.

References

[1] CISCO, "Cisco visual networking index: Forecast and methodology, 2011-2016," CISCO, Tech. Rep., May 2012.

[2] comScore, "comscore releases july 2012 u.s. online video rankings," comscore, Tech. Rep., July 2012.

[3] A. Passarella, "A survey on content-centric technologies for the current internet: Cdn and p2p solutions," *Computer Communications*, vol. 35, no. 1, pp. 1–32, 2012.

[4] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can isps and p2p users cooperate for improved performance?" *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 29–40, July 2007.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT)*, 2009, pp. 1–12.

[6] T. Stockhammer, "Dynamic adaptive streaming over http - standards and design principles," *ACM Multimedia Systems*, pp. 133–143, Feb. 2011.

[7] I. MPEG, "Information technology - dynamic adaptive streaming over http (dash) - part 1: Media presentation description and segment formats," ISO/IEC MPEG, Tech. Rep., 2012.

[8] ——, "Information technology - coding of audio-visual objects - part 12: Iso base media file format," ISO/IEC MPEG, Tech. Rep., 2005.

[9] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *Communications Magazine, IEEE*, vol. 49, no. 7, pp. 26 –36, july 2011.

[10] S. Paul, R. Yates, D. Raychaudhuri, and J. Kurose, "The cache-and-forward network architecture for efficient mobile content delivery services in the future internet," in *Innovations in NGN: Future Network and Services*, May 2008, pp. 367 –374.

[11] C. Dannewitz, "Netinf: An information-centric design for the future internet," in *Proceedings of 3rd GI/ITG KuVS Workshop on The Future Internet*, May 2009.

[12] Ccnx project v0.6.0. [Online]. Available: http://www.ccnx.org

[13] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "Voccn: voice-over content-centric networks," in *Proceedings of the 2009 workshop on Re-architecting the internet*, ser. ReArch '09. New York, NY, USA: ACM, 2009, pp. 1–6.

[14] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang, "Act: audio conference tool over named data networking," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking (ICN '11)*. ACM, 2011, pp. 68–73.

[15] H. Xu, Z. Chen, R. Chen, and J. Cao, "Live streaming with content centric networking," in *3rd Int. Conf. on Networking and Distributed Computing*, Oct. 2012.

[16] Z. Li and G. Simon, "Time-shifted tv in content centric networks: The case for cooperative in-network caching," in *IEEE International Conference on Communications (ICC)*, June 2011, pp. 1 –6.

[17] C. Müller and C. Timmerer, "A vlc media player plugin enabling dynamic adaptive streaming over http," in *Proceedings of the 19th ACM international conference on Multimedia*. New York, NY, USA: ACM, 2011, pp. 723–726.

[18] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, ser. MMSys '12, 2012, pp. 89–94.

[19] Big buck bunny. [Online]. Available: http://www.bigbuckbunny.org/

[20] Wireshark analyzer. [Online]. Available: http://www.wireshark.org