Adaptive Video Streaming over CCN with Network Coding for Seamless Mobility

Abinesh Ramakrishnan
Huawei Technologies
Santa Clara, CA
Email: ibanesh@gmail.com

Cedric Westphal
Huawei Technologies & University of California, Santa Cruz
Santa Clara, CA
Email:cedric.westphal@huawei.com

Jonnahtan Saltarin
University of Bern
Bern, Switzerland
Email:saltarin@inf.unibe.ch

***Abstract*—We consider the benefits brought by network coding in Information-Centric Networks (ICNs) in the case of a video streaming application. Network coding, when combined with ICN, allows a data transfer session to use multiple sources for the content seamlessly. It permits the client fetching the content to use multiple interfaces at the same time in an asynchronous manner, while using their capacity in an additive manner. This allows to create a *logical* link between the user and the content. In the case of video streaming, this logical link allows the rate adaptation logic to find the proper streaming rate while using multiple links concurrently.**

We implemented a video streaming system which works using network coding and CCN. We have shown that this implementation performs satisfactorily, and has comparable performance to a system without network coding in the unicast single source, single path case, and delivers significant performance gain (better QoE, higher throughput) when the video client retrieves the stream from multiple concurrent sources. We hope to demonstrate that network coding and CCN provides seamless mobility for a video streaming application.

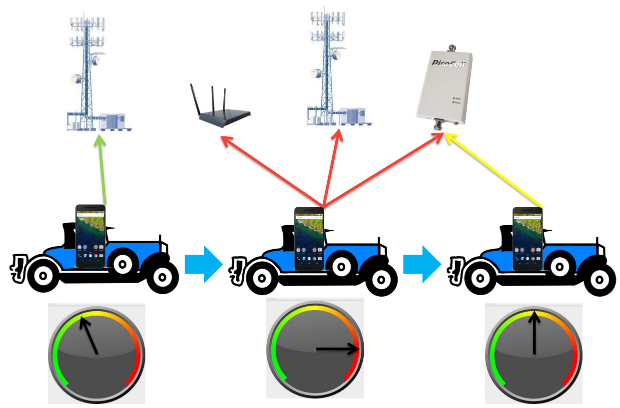

Fig. 1: Network Coding plus ICN enable Seamless Mobility

I. Introduction

Today's Internet traffic is dominated by content distribution services like live-streaming and video-on-demand. Video traffic is expected to account for 82% of all consumer Internet traffic by 2020 [1]. Popular services like Netflix, YouTube, etc have been the driving force behind the explosive traffic growth seen in recent times.

Information-Centric Networking in general, and Named Data Networking (NDN) or Content Centric Networking (CCN) [2] in particular, is a novel network architecture that aims to address the inherent inefficiencies in content delivery.

NDN/CCN does not follow the current IP model for data transfer. In the traditional TCP/IP model (and to some extent, in more modern protocols such as QUIC as well), the client requesting some content needs to first map the URL in the name of the content to a server. This server will mostly stay the same for the length of the session. The session is actually bound to this server, and the session's goal is to transfer the specific content held from this server. Upon a mobility event of the user (for instance, connecting from a LTE interface to a WiFi network), the session has to be reset and it will resume from the new location with the previous server/copy of the content.

It is easy to see this is inefficient along several axes: there is no reason the server is still the most appropriate after a mobility event; the tear down and the reset of the TCP connection is time-consuming; and the session is tied up to an interface and a path. Potentially multiple paths may be concurrently used with MP-TCP, but they still connect the same two endpoints.

CCN/NDN identifies content by its name instead of pointing to the physical location of the content. This makes it possible to store the content anywhere in the network and to connect directly to the content, without binding with a server first. CCN/NDN also removes the session-oriented nature of TCP by adopting a per chunk data transfer. This allow data chunks to be retrieved from different locations, potentially in parallel. In the previously discussed mobility event, the network could seamlessly forward new content requests (termed ”interests” in CCN terminology) to a nearby copy of the content on a different host, rather than sticking with the previously identified copy.

The CCN/NDN architecture has an inherent potential for allowing users to exploit multi-path connectivity available in the network to multiple content locations, as the end user nodes and the intermediate nodes can simultaneously transmit interest over multiple network interfaces to retrieve segments of the requested content. However, to fully achieve this potential, we argue that network coding [3] is necessary to leverage the multiple paths and multiple content locations seamlessly. Network coding with CCN allows content to be requested from multiple copies via multiple interfaces in an asynchronous manner.

We present in this paper an implementation of network coding for ICN and we focus our implementation on the use case of video streaming. There has been some investigation of the behavior of video streaming in Information-Centric Networks [4], but none considered how network coding allows the rate adaptation logic in adaptive video streaming to consider the underlying network between the client and the multiple copies of the content as a single logical link. This is what we describe in this paper.

The paper is organized as follows. In Section II we look at the prior work in this area. Section III presents an overview of DASH and CCN, as well as the big picture of our proposal; section IV outlines the architecture. Section V details the implementation. Section VI shows the performance evaluation benefit as measured in our implementation testbed, and we offer concluding remarks in Section VII.

II. Related Work

To the best of our knowledge, the first application of network coding in ICN has been explored in [3] where the NC3N architecture has been introduced. Inspired by [3], CodingCache has been proposed in [5] which uses network coding to replace the content segments in the cache of the network nodes resulting in an improved cache hit rate. In [6], it is again shown how applying network coding in CCN can enhance throughput and reduce network load.

Wang et al [7] [8] demonstrated how network coding helps achieve better performance when combined with proper content placement in the caches. [9] demonstrates that network coding in content-centric networks provided additional privacy benefits, a clear improvement over the explicit nature of CCN.

Another approach called NetCodCCN was proposed in [10] which addressed some of the shortcomings of the approaches presented in [3] and [5]. Our implementation builds upon the work of [10].

Although multi-path transmission has recently been studied [11] as a means for increasing throughput, CCN usually uses single-path transmission between the end user and the origin server or the content router. [12] considers the case of network-coded caching-aided multicast (NCCAM), and show that random linear coded caching and multicasting is sufficient for achieving close to minimum cost caching-aided multicast in random topologies under Zipf popularity distribution for content.

Even in multicast scenarios, the optimal content delivery rate is only attained if the segments are delivered over the optimal set of multicast trees [13], which does not scale for large and dynamic topologies. Furthermore, the computation of the optimal set of multicast trees needs a central entity that is aware of the network topology, which is hard to be done in dynamic networks.

The optimization of ICN for video streaming was discussed in [4] [14]. Optimization of video transcoding and caching was described in [15] [16] [17]. [18] showed how content caching in a congestion-aware network would improve video quality; [19] uses a rate-adaptation algorithm which considers the availability of caches at the network edges. All these works improve the QoE of the video streaming user, but do not take advantage of network coding. As far as we know, this work is the first to consider video streaming in a network coding enabled information centric network.

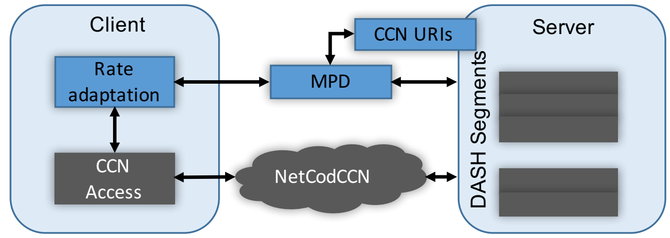

Fig. 2: Overview of the System Architecture

III. Overview

We now describe our architecture for video streaming in a network-coded ICN, after a short background description for network coding and rate adaptive video streaming.

Network coding [20] can bring both distributed data storage and efficient multipath solution to CCN. With network coding all the network nodes perform coding operations on the received packets instead of just replicating and forwarding them as in traditional networks. The receivers decode the information when they receive a decodable set of packets, i.e., as many linearly independent coded packets as the number of source data segments.

The principal mechanism of adaptive video streaming involves the video client downloading some segments from the server and using the completion time of this download to estimate the best rate that the network connection can support.

Dynamic Adaptive Streaming over HTTP (DASH) [21] is an established and widely used standard for adaptive streaming. The basic concept of Dynamic Adaptive Streaming over HTTP (DASH), the predominant standard for adaptive streaming, is to divide the media content into segments of different bit rates, resolutions, etc. A Media Presentation Description (MPD) describes segment information, the relationship between the contents' media segments and the associated bitrate, resolution, and timeline. Using the MPD, a DASH client is able to download the most appropriate segment satisfying the users' context. There are already couple of works on adaptive video stream over CCN, namely [22] and [14]. The former presents an implementation and evaluation of Dynamic Adaptive Streaming over CCN (DASC). The latter studies the impact of using an adaptive video rate on the in-network caching performance and proposes a rate adaptation mechanism that takes into account in-network caching.

Network coding in CCN has been shown to reduce network bandwidth and delay. However, its impact on rate adaptation is unknown. We are presenting a framework to evaluate rate adaptation policies in a CCN network. Our overarching goal is depicted on Figure 1, where a mobile device can associate

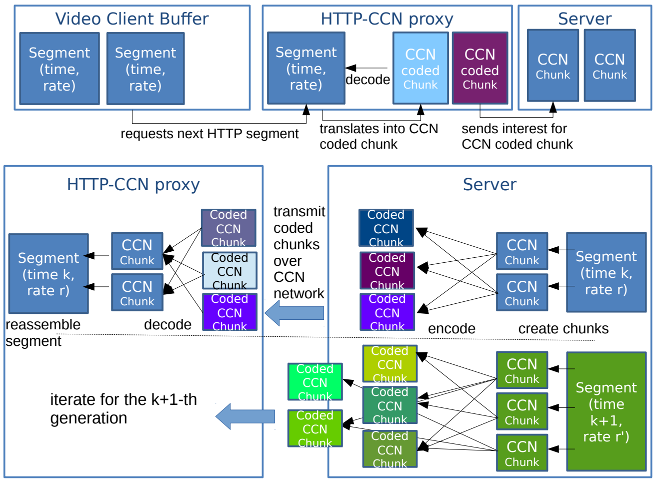

Fig. 3: Encoding and Decoding Operations per Generation

with multiple radio access networks, potentially at the same time, and the rate adaptation mechanism views the mishmash of links between the device and the (potentially multiple copies of the) content as a single logical link upon which to perform rate adaptation.

IV. Architecture

We present an implementation to demonstrate and study the behavior of rate adaptation mechanism for video streaming in CCN with network coding. Figure 2 shows the overview of the system architecture of the testbed. NetCodCCN [10] is used as the underlying framework for building the system.

In order to keep this paper concise, we do not detail all modification done to the NetCodCCN. The end users make use of DASH compatible video clients that has been patched to support DASH over CCN. The server node hosts a CCN test repository with DASH based content, the content is imported in a CCN URI namespace and the MPD corresponding to this repository references this URI.

In DASH the video is divided into fixed duration segments of different bit rates, resolution, etc. In CCN, each segment is further divided in to smaller chunks that fit into Data messages. The size of the segments in a representation with higher bit rate or resolution will be bigger than the segments of a representation with lower bit rate or resolution. Consequently, the segments of a higher quality representation will be divided into larger number of chunks than the segments of a lower quality representation. As described in [10], NetCodCCN restricts the coding operations to a small group of chunks called generations.

In the original NetCodCCN implementation [10] this generation size was fixed to a certain number of chunks. In our implementation, we modify this so that the coding is restricted to all the chunks within a segment, i.e. the generation size is determined by the segment and representations. The generation size varies for each segment based on the content and quality representation. Figure 3 shows the two generation k and $k+1$, where the corresponding segment of length s seconds for a given time interval $(ks, (k+1)s)$ and rate r is decomposed first into chunks, and these chunks then encoded into network coded chunks upon request. Each network encoded chunk is a random liner combination of the original chunks. Figure 3 also shows the $k+1$st generation, with a rate $r' > r$, and the segment is decomposed into more chunks, and more encoded chunks are necessary to decode the original segment.

The video client at the end user requests the underlying CCN daemon of this node for a segment of a particular representation based on the network condition. This user node then forwards an interest message asking for a network coded chunk of this segment to its neighboring nodes through all available network interfaces. The intermediate nodes then follow the Interest processing algorithm of NetCodCCN to either forward the interest or reply to the interest. When a node replies to an interest with a coded data message, the coded data message propagates back through the network with the intermediate nodes following the Content processing algorithm of NetCodCCN.

At the user node, the CCN daemon with network coding does this until it gets a sufficient number of coded chunks from the same generation to be able to decode the requested segment. Once it decodes all the chunks of the requested segment, it will send the segment data to the video client which will be able to play it successfully.

V. Implementation Set-up

Video streaming over NetCodCCN is implemented by integrating the required changes in the CCNx 0.8.2 code. The network coding operations are performed in a finite field of size 2^8. For the video client at the end user, we made use of VLC with DASH over CCN plugin made available by the ITEC-DASH project.

The server node in our implementation is loaded with the CCN test repository that is available at the ITEC-DASH project website. This CCN repository contains 14 representations of the Big Buck Bunny content at 14 different representations, ranging from 100 kbps to 4500 kbps. Each representation has the same resolution of 854×480. Segments are encoded and segmented at a segment length of 2 seconds. Figure 4 shows the testbed environment that we deployed in our lab. Some of the servers are virtual machines, the client runs into another virtual machine, and the CCN network is emulated using virtual switches (bridged over Ethernet LAN if the virtual machines are on different servers).

– Single interface evaluations connect the client to only Server 1. This is used to established the cost of network coding in terms of QoE. We do so by comparing a unicast connection to Server 1 with and without network coding to quantify the rate penalty in Section VI.

– Dual interface evaluations connect the client to both Server 1 and Server 2 through distinct interfaces. The path followed by the interest to Server 1 and Server 2 are also distinct. By turning on and off the interfaces 1 and 2, we can emulate mobility of the user device.

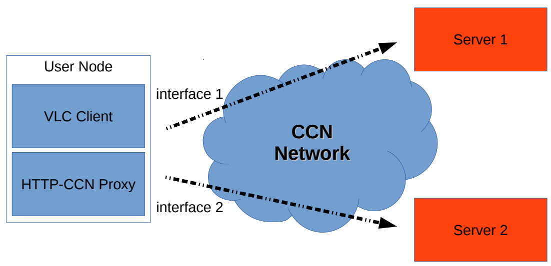

Fig. 4: Testbed Environment

– Not considered here, but worthy of further evaluation would be to insert caching nodes. The evaluation set-up only connects one client with the servers. However, multiple clients requesting the same stream could benefit from intermediary nodes that are able to cache the coded content.

The dual interface setup highlights the effects of rate adaptation on in-network caching efficiency. In particular, it demonstrates the additivity of the link when abstracted by the rate adaptation logic. It would be interesting to see how a rate adaptation scheme like DASH-INC [14] that takes in to account in-network caching would perform in CCN with network coding.

VI. Evaluation

We tested the implementation in our lab. Figure 5 presents the comparison of the rate requested by the video client (that is, the rate computed by the adaptation logic based upon the performance of the download of the previous chunks) in three cases: when one interface is used without network coding and with network coding; this shows that the penalty of using network coding embedded in the overhead and in the additional processing time to decode the packets is not significant. The performance is similar for both these cases.

In the third case of Figure 5, there are two copies available from two distinct servers over two different interfaces. Only the network coding implementation can take advantage of these two copies. If the interest packets of CCN were being broadcast to the two servers, they would retrieve exactly the same data and would serve no other purpose than wasting network resource. The network coding implementation can additively use these two copies and achieve the highest rate throughout the experiment.

In Figure 6, we show how our implementation seamlessly react to mobility events. In this case, from the beginning of the experiment up to about the 20th point in the plot, only interface 1 is available. From the 20th point up to about the 30th point in the plot, both interface 1 and interface 2 are available. From the 30th point onwards, only interface 2 is available. This mimics a device having a soft handover in between two access points.

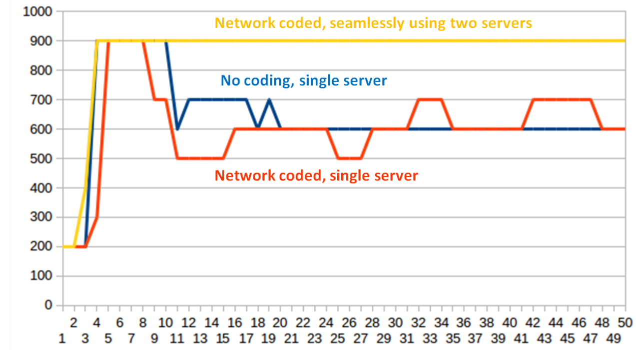

Fig. 5: Video streaming with and without network coding: static comparison

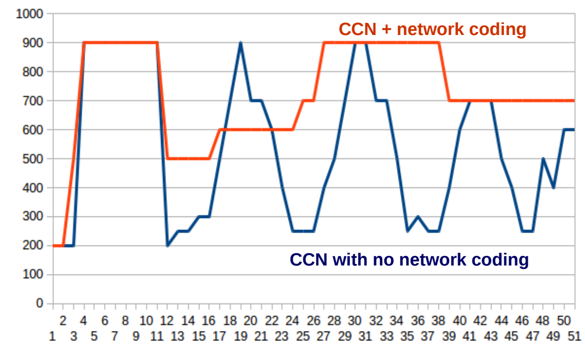

Fig. 6: Video streaming with and without network coding: dynamic comparison

(The exact point at which the interface becomes available is slightly different with and without network coding, owing to different processing delays in either case). We see that there is a lag in the rate adaptation. We can see that CCN + network coding outperforms pure CCN, as it can take advantage of the multiple interfaces as they come up.

VII. Discussion & Conclusions

We have presented an implementation of a video streaming system using network coding and ICN. Because CCN is connectionless on a per chunk basis, and because network coding enables the additive and asynchronous use of multiple interfaces, network coding and ICN easily support seamless mobility across heterogeneous technologies. Other benefits of network coding, such as increased privacy and increased content availability in the cache, are preserved in our implementation, if not evaluated here.

Our evaluation results based upon our implementation show a significant gain for network coded CCN for video streaming. Our per-segment generation of network coded video chunks does not add a significant latency penalty and it allows to use multiple interfaces in a transparent manner to the rate adaptation logic. The rate adaptation views the connection from the CCN engine to multiple copies of the content over heterogeneous technologies and over multiple paths as a single logical link upon which to perform rate adaptation.

Future work would involve taking our implementation out of the lab and into the wild, by having the device connect to WiFi access points when available, and LTE network otherwise and observe the QoE at the video streaming client. Using multiple clients and a larger network with caches would show a wider range of benefits as well.

References

[1] CISCO, "Cisco Visual Networking Index: Forecast and Methodology, 2015-2020," in *CISCO*, 2016.

[2] 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*, pp. 1–12, ACM, 2009.

[3] M.-J. Montpetit, C. Westphal, and D. Trossen, "Network coding meets information-centric networking: an architectural case for information dispersion through native network coding," in *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, pp. 31–36, ACM, 2012.

[4] C. Westphal (Editor) et al, "Adaptive Video Streaming in Information-Centric Networking (ICN)." IRTF RFC7933, ICN Research Group, Aug. 2016.

[5] Q. Wu, Z. Li, and G. Xie, "Codingcache: multipath-aware ccn cache with network coding," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pp. 41–42, ACM, 2013.

[6] S. Miyake and H. Asaeda, "Network coding and its application to content centric networking," 2013.

[7] J. Wang, J. Ren, K. Lu, J. Wang, S. Liu, and C. Westphal, "An optimal cache management framework for information-centric networks with network coding," in *IFIP/IEEE Networking Conference*, June 2014.

[8] J. Wang, J. Ren, K. Lu, J. Wang, S. Liu, and C. Westphal, "A minimum cost cache management framework for information-centric networks with network coding," *Computer Networks*, Aug. 2016.

[9] Q. Wu, Z. Li, G. Tyson, S. Uhlig, M. A. Kaafar, and G. Xie, "Privacy-aware multipath video caching for content-centric networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 2219–2230, Aug 2016.

[10] J. Saltarin, E. Bourtsoulatze, N. Thomos, and T. Braun, "Netcodccn: a network coding approach for content-centric networks," *arXiv preprint arXiv:1512.00259*, 2015.

[11] G. Rossini and D. Rossi, "Evaluating CCN multi-path interest forwarding strategies," *Computer Communications*, vol. 36, no. 7, pp. 771–778, 2013.

[12] J. Llorca, A. M. Tulino, K. Guan, and D. C. Kilper, "Network-coded caching-aided multicast for efficient content delivery," in *2013 IEEE International Conference on Communications (ICC)*, pp. 3557–3562, June 2013.

[13] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," in *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, p. 143, IEEE, 2004.

[14] R. Grandl, K. Su, and C. Westphal, "On the interaction of adaptive video streaming with content-centric networking," in *2013 20th International Packet Video Workshop*, pp. 1–8, IEEE, 2013.

[15] J. Yichao, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach," *Circuits and Systems for Video Technology, IEEE Transactions on*, 2015.

[16] J. Yichao, Y. Wen, and C. Westphal, "Towards joint resource allocation and routing to optimize video distribution over future internet," in *IEEE/IFIP Networking Conference*, May 2015.

[17] G. Gao, Y. Wen, and C. Westphal, "Dynamic resource provisioning with qos guarantee for video transcoding in online video sharing service," in *ACM MultiMedia*, Oct. 2016.

[18] Y.-T. Yu, F. Bronzino, R. Fan, C. Westphal, and M. Gerla, "Congestion-aware edge caching for adaptive video streaming in information-centric networks," in *IEEE CCNC Conference*, Jan. 2015.

[19] F. Bronzino, D. Stojadinovic, C. Westphal, and D. Raychaudhuri, "Exploiting Network Awareness to Enhance DASH over Wireless," in *IEEE CCNC*, Jan. 2016.

[20] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

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

[22] 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 *2013 IEEE International Conference on Communications (ICC)*, pp. 3629–3633, IEEE, 2013.