Low Latency Low Loss Streaming using In-Network Coding and Caching

Kazuhisa Matsuzono, Hitoshi Asaeda, Thierry Turletti

Low Latency Low Loss Streaming using In-Network Coding and Caching

Kazuhisa Matsuzono*, Hitoshi Asaeda*, and Thierry Turletti†
*National Institute of Information and Communications Technology (NICT), Japan.
Email: {matsuzono, asaeda}@nict.go.jp
†Université Côte d'Azur, Inria, France. Email: thierry.turletti@inria.fr

***Abstract*—Owing to the rapid growth in high-quality video streaming over the Internet, preserving high-level robustness against data loss and low latency, while maintaining higher data transmission rates, is becoming an increasingly important issue for high-quality real-time delay-sensitive streaming.**

In this paper, we propose a low latency, low loss streaming mechanism, L4C2, convenient for high-quality delay-sensitive streaming. With L4C2, nodes in the network estimate the acceptable delay and packet loss probability in their uplinks, aiming at retrieving lost data packets from in-network cache and/or coded data packets using in-network coding within an acceptable delay, by extending the Content-Centric Networking (CCN) approach. Further, L4C2 naturally provides multiple paths and multicast technologies to efficiently utilize network resources while sharing network resources fairly with competing data flows by adjusting the video quality when necessary.

We validate through comprehensive simulations that L4C2 achieves a high success probability of data transmission considering the acceptable one-way delay, and higher QoE while suppressing the interest and redundant data traffic than the proposed multipath congestion control mechanism in CCN.

I. Introduction

Because of the huge bandwidth improvement in both access links and backbone networks and the increasing demand of video applications [4], end users are expecting to receive higher quality video streaming such as ultra-high definition (UHD) video termed 4K or 8K video. However, since high quality video streaming consumes a large amount of network bandwidth (e.g., approximately 2 to 18 Gbps for uncompressed 4K video [6]), video codecs with strong compression are required. Considering the rapid increase in computing capabilities that allows efficient video compression, it is expected that these applications will become increasingly popular in the near future [5]. However, the perceived quality of video compressed with a higher compression ratio is more likely to be affected by data losses. In the case of delay-sensitive applications such as real-time live video broadcast services and interactive video communication, it is more critical to maintain end-to-end delay within a range as the acceptable one-way delay must be approximately 150 ms for interactive applications [10], [13]. Preserving low latency and high-level robustness against data loss while maintaining higher data transmission rates will become an increasingly important issue for high-quality real-time delay-sensitive streaming.

Recently, Content-Centric Networking (CCN) [1] and Named-Data Networking (NDN) [2] have been proposed as a new network architecture promising to form the cornerstone of the future Internet. The name-based communication in CCN/NDN enables consumers (i.e., data receivers) to obtain desired data or content without establishing and maintaining continuous end-to-end communication channels between hosts. Hence, CCN/NDN communication allows consumers and intermediate nodes (or routers) to utilize in-network caching and flexibly exploit multiple paths at the network core layer. This feature contributes to improved use of network resources and reduces the time for consumers to obtain the desired data. Further, CCN/NDN intrinsically supports multicast communication because identical request packets (called interest) are aggregated at the CCN/NDN routers, which is a key feature for real-time live video streaming in terms of eliminating duplicate data traffic.

The CCN/NDN architecture is not only suitable for the massive amounts of archive content exchanged via the Web but can also facilitate a class of applications that involve delay-sensitive applications shared with consumers [7], [8]. However, despite the potential advantages, certain aspects of the CCN/NDN transport to support high-quality delay-sensitive video applications have not yet been fully investigated. As reported in [18], [28], it is reasonable to integrate MPEG-DASH [29] into CCN, because (1) it assumes a receiver-driven (pull-based) approach as employed in CCN and (2) the protocol layers to which they belong are different, i.e., MPEG-DASH belongs to the application layer and CCN to the network layer. However, these approaches or today's VoD solutions (such as Netflix) basically maintain a large data buffer (e.g., 5 seconds) and focus on avoiding playout stopping (i.e., rebuffering) by adjusting video quality according to network conditions, rather than on providing robustness against data losses in the network and low latency or low delay communication.

In this paper, envisioning a scenario where the traffic volume of high-quality delay-sensitive video applications competing with other traffic will increase, we propose a low latency, low loss streaming mechanism using in-network coding and caching, named L4C2. The principal idea of L4C2 is that nodes in the network estimate the acceptable delay and packet loss probability in their uplinks, aiming at retrieving lost data packets from in-network cache and/or coded data packets using in-network coding within the acceptable delay, by extending the CCN approach. Further, to provide the best possible video

quality, L4C2 naturally cooperates with multiple paths and multicast technologies to efficiently utilize network resources while fairly sharing network resources with competing data flows by adjusting the video quality (i.e., consumed bandwidth) as necessary. Our main contribution is that we present the system architecture of L4C2 that leverages the following three types of interest packets based on the estimated transmission success probability; 1) "symbolic interest (SMI)" used for continuously receiving real-time video data while suppressing high-rate interest packets, 2) "control interest (CNI)" used for controlling redundant data packets generated by in-network coding, and 3) "regular interest (RGI)", which is the common interest in CCN/NDN, used for triggering retransmissions to recover lost video data packets.

The reminder of this paper is organized as follows: In Section II, the background for this research and related work are provided. The detailed design of L4C2 is provided in Section III. In Section IV, we evaluate the L4C2 performance and demonstrate that the adaptation mechanism is more feasible than the existing method through comprehensive simulations. We finally present our concluding remarks and further research issues in Section V.

II. Background and Related Work

A. CCN/NDN Basics

CCN/NDN advocates receiver-driven, hop-by-hop communication, where a consumer initiates communication by sending an "interest" packet for a specific content to receive the corresponding "data" packet(s). Routers forward the interest to routers or the publisher holding the content via name prefixes for routing. A Forwarding Information Base (FIB) is a lookup table used to determine incoming interfaces (called Faces) for each set of content. Its entries consist of a name prefix and incoming Face pairs. If there are multiple incoming Faces for a certain name prefix in the FIB, the "forwarding strategy" assumes a role in finally determining which and how many of the available incoming Faces to use (multipath data retrieval). Each router also maintains a Pending Interest Table (PIT), which is a lookup table containing the name prefix and outgoing Faces used to forward the received data packets.

When a router receives an interest, it examines the data name to determine if it is cached in its Content Store (CS) (or in-network cache). If the named data is in the CS of the router, the router forwards the cached data to the Face on which it arrived. If the named data is not present in the CS of the router, the router searches its PIT to determine whether another interest for the same data has already been processed. If the relevant data name is in the PIT of the router and the arrival Face is not set in its PIT, the router updates the PIT entry by adding the arrival Face as the outgoing Face and discards the interest. If the relevant data name is not in the PIT of the router, the router creates a new PIT entry and forwards the interest through the incoming Faces determined by the forwarding strategy.

If no router along the path of the content to the publisher has the requested data in its cache, the interest packet will finally reach the publisher. When an interest is received, the publisher forwards the named data to its downstream router. Each router along the path that receives the interest forwards the data through the outgoing Faces, stores the forwarded content in its CS, and deletes the corresponding PIT entry.

B. Motivation

For high-quality delay-sensitive video streaming, a consumer must send high-rate interests with unique packet identifiers (i.e., sequence or segment numbers such as "/example.com/video1/seq=10") continuously and consecutively. For a 30 Mbps stream whose packet size is about 1,000 bytes as in [20], the consumer must send about 3,750 different interest packets per second. Such high-rate interests may cause network congestion resulting in interest and data packet losses. Increasing the packet size can reduce the traffic. However, a large data packet can cause packet fragmentation and can degrade throughput because a single fragment loss initiates the entire data packet retransmission in the network. Furthermore, to receive streaming data in real time, the consumer must execute highly complex interest operations; the consumer must manage the timing of sending the corresponding interest packets such that they arrive at the publisher or the forwarding router at the appropriate time. Finally, CCN/NDN multicast communication, which is essential to eliminate duplicate data packets, heavily relies upon each interest arrival time at the router aggregating the interest and on the presence of the data in the CS.

To address these issues, a CCN extension called "symbolic interest" was proposed in [23]. Symbolic interest enables a wildcard approach for content name specification: the consumer specifies the name prefix with a flag "++" and a label "segment" (e.g., "/example.com/video1/++segment") without a sequence number. When a router receives the symbolic interest, it registers the name prefix in its PIT, and upon transmission considers (replaces) the "++segment" as the sequence numbers. It allows the nodes to consecutively forward streaming data during the specified period (i.e., lifetime) while receiving streaming data with the relevant content name prefix. Before the router's PIT entry expires at the end of the specified interest lifetime, the consumer sends the symbolic interest. The upstream router then updates the lifetime in its PIT entries and forwards the received symbolic interests to its upstream router(s). This allows a consumer to continuously obtain data packets while avoiding high-rate interests.

The symbolic interest approach is simple and can resolve the aforementioned issues. However, this modification does not fully exploit the CCN/NDN benefits such as in-network caching and multipath data retrieval for low latency, low loss streaming, because it is optimized for continuously receiving real-time video data with a single path. With regards to this topic, it is necessary to propose an additional CCN/NDN extension that can provide robustness against data loss while preserving higher video quality and minimizing delay and network cost by eliminating duplicate or unnecessary traffic.

C. Related Work

NDN-RTC [27] was developed for a real-time video conferencing library utilizing all NDN features. It demonstrates low-latency HD video communication over NDN while preserving network-supported scalability through the help of the network rather than the help of the producer. It can provide robustness using Forward Error Correction (FEC) at the publisher side in an end-to-end manner. However, this scheme does not provide robustness with consideration for minimizing delay and it is difficult for consumers to manage the timing of sending interests to obtain data in a real-time manner and cooperate with multicast technologies [23].

Han et al. proposed a new loss protection scheme called redundant packet transmission (RPT) [9] to achieve low latency and high-level robustness against network loss in content-aware networks. The key idea of RPT is that the duplicate copies of packets that the router transmits are encoded into short packets utilizing router caching, which results in a low bandwidth overhead for the redundancy. Compared to traditional FEC-based schemes, this technique contributes to lower latency because FEC requires the decoder to wait for receiving a sufficient number of packets for loss recovery. The proposed scheme assumes that routers must cache all data packets sent within the last tens of seconds, which requires a huge volume of caching space for high quality video streaming. RPT is therefore not suitable for high-quality video streaming owing to cache space limitations [17].

In [14], [15], a network coding approach for CCN was proposed. Routers cache coded data packets using the received original and/or coded data packets to improve the cache hit probability resulting in higher throughput. However, this approach focuses on archived data retrieval rather than real-time data generated sequentially by the publisher; hence, it does not provide data loss protection within the acceptable delay.

Carofiglio et al. proposed a multipath congestion control and interest forwarding mechanism [25] where consumers adjust the number of outstanding interests according to a window-based Additive-Increase/Multiplicative-Decrease (AIMD) mechanism. In this proposal, the timing for transmission of interests must be perfectly adjusted by monitoring the reception bandwidth to minimize playback delay. However, the transport protocol imposes highly complex interest operations on consumers and leads to longer playback delay.

III. L4C2 Proposal

A. Design Rationale

1) ***Cache for Lost Packet Retransmission*****:** A retransmission scheme requires a minimum of one Round Trip Time (RTT) to recover from a packet loss. This leads to a performance issue because the RTT can easily be sufficiently large to delay reception times of streaming data packets to the point where they become unplayable [16]. Conversely, the retransmission scheme in the hop-by-hop manner intrinsically supported by CCN/NDN has a higher possibility of success because each RTT in the link between routers is likely to be smaller than that in the link between source and receiver. Furthermore, a router maintaining an in-network cache, which does not have to be as large because it is used for delay-sensitive streaming, can potentially retransmit lost packets to the consumer if the router thinks that the retransmission will be completed within an acceptable delay for the streaming. To adopt the hop-by-hop retransmission, a downstream router must only send the interest for the lost data to request the upstream router to retransmit it from the in-network cache.

2) ***Coding with Low Performance Penalty*****:** In the case where the success probability of only retransmissions in a certain link is low because of a large RTT, it is more efficient for the upstream router to add some redundant data using a coding scheme. Because CCN/NDN can support a rate adaptation scheme in a hop-by-hop manner, we need to ensure that 1) each router flexibly generate coded data packets using both original and coded data packets and 2) the time that is required for encoding/decoding be affordable such that the consumer can recover lost data packets within a low delay. In the case of typical erasure codes such as Reed-Solomon (RS) codes [19], routers in the path must obtain all the original data packets in a certain block to generate new coded packets. This means that they may be required to decode all the missing original data packets because receiving packets at each router are not always original data packets. The time required for RS decoding is not compatible with real-time data transmission, particularly considering high-quality video streaming [20]. In the case of Random Linear Network Coding (RLNC) [21], routers in the path can flexibly generate new coded packets using both original and coded data packets and the encoding/decoding complexity is reasonable as efficient implementations of network coding [22] are available.

3) ***Congestion and Rate Control*****:** Adding redundancy may cause further network congestion and adversely affect the streaming QoE. In particular, in a situation where fair bandwidth sharing is more desirable, each streaming flow must adapt to the network conditions to fairly consume the available link bandwidth. It is thus indispensable that each streaming data flow cooperatively implements congestion control to adjust the consumed bandwidth to stabilize the network condition (i.e., to achieve low packet loss rate, delay, and jitter). In CCN/NDN, the transport protocol in each router can estimate and control data traffic per content flow by adjusting the sending rate of the regular interest packets [24]. For high-quality real-time video streaming, yet another rate control such as the adjustment of video rate is useful. Assuming that the total available bandwidth for each streaming flow is based on the bandwidth sharing policy defined by the network operator, it is possible to adjust the total data traffic such that it does not exceed the upper limit. Note that even in such a situation, owing to uncertainty regarding content flow dynamics such as when/what types of content flow occur and data traffic burstiness, network congestion can arise as long as the transport mechanism attempts to fully utilize the network resources [24].

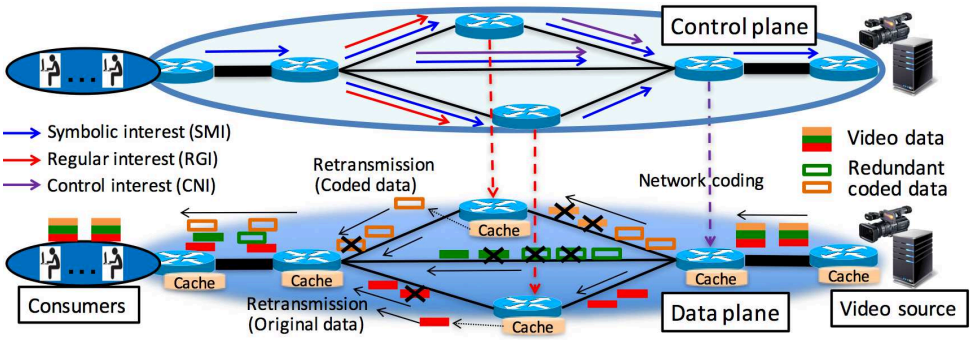


Fig. 1. The system architecture of L4C2.

B. System Description

In this section, we describe the design and algorithm of L4C2. Figure 1 illustrates the system architecture of L4C2. It concomitantly uses symbolic interest (SMI), regular interest (RGI), and control interest (CNI) to fully leverage CCN/NDN benefits.

*1) **Model Assumptions**:* We consider real-time video content viewing over CCN/NDN-enabled wired networks to be modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with bidirectional arcs, where there are consumers $\mathcal{U} \subset \mathcal{V}$ sending SMIs to receive various video content, and each of the corresponding sources $\mathcal{S} \subset \mathcal{V}$ generates the video content in real-time. Each node $i \in \mathcal{V}$, which is connected to neighborhood node(s) $j \in \mathcal{V}$, sends SMI and RGI to node j, and then the SMI and RGI traverse the link(s) $l(i,j) \in \mathcal{E}$. The corresponding data that the source $s \in \mathcal{S}$ generates will arrive at the node i over $l(j,i)$.

Real-time video streaming source s provides a set of the L different video qualities $\mathcal{Q} = \{q_1, q_2, ..., q_L\}$ predefined by the video content producer; these are produced by video encoder(s) or by using a layered encoding technique, such as Scalable Video Coding (SVC) [12]. A produced video segment with quality q is denoted by $C_n^q, q \in \mathcal{Q}$ with the segment number $n = 0, ..., N$. According to the network condition, each node receiving data with video quality $q_m, 1 \leq m \leq L$, can change the sending video rate r from the set of available rates $\mathcal{R} = \{r_{q_1}, r_{q_2}, ..., r_{q_i}\}$. Each node can recognize the information about $\mathcal{Q}$ and $\mathcal{R}$ by receiving and reference to meta-information the source s provides.

RLNC is applied into each coding group $\mathcal{N}_{id}^q$ with the group index $id = 0, ..., \lfloor N/k \rfloor$, which consists of the k different original data packets with the same video quality q (i.e., $C_n^q, n = k \times id, ..., (k \times id) + k - 1$). With the encoding vector $g = (g_0, g_1, ..., g_{k-1})$, where each coefficient was randomly chosen in $GF(2^8)$, a node can generate a coded packet $\acute{N}_{id}^q = \sum_{j=0}^{k-1} g_j C_{(k \times id)+j}^q$ from N_{id}^q. Encoding can be performed recursively, namely, using already encoded packets. Thus, from the k packets including the original and/or coded packets within the same group id, a new coded packet can be easily generated. When using coefficients over $GF(2^8)$, the probability of selecting linearly dependent combinations is negligible [26]. The consumer, therefore, can recover all of the missing packets from any set of exactly k packets within the coding group. The k value is related to the waiting time for the consumer to detect whether the lost original packets can be recovered. Thus, according to the video rate $r \in \mathcal{R}$, in advance, the video content producer defines the constant k value such that the consumer can recover the lost original packets without a large recovery processing delay.

*2) **Data Forwarding and Control using Interest**:* In the control plane, a consumer $u \in \mathcal{U}$ sends SMIs specifying the video content name with or without the video quality $q \in \mathcal{Q}$. In the case where q is not specified, each router attempts to provide streaming data with the maximum video quality. According to the network conditions and estimated acceptable delay in the uplink(s), a downstream router is able to determine which uplink to use and what level of video quality to receive through the link; it then sends the SMI specifying q through the determined uplink. Although the SMI eliminates the tight feedback loop as in receiver-driven congestion control using only RGIs, each L4C2 node estimates and promptly controls the receiving data traffic to avoid network congestion by adjusting q specified by the SMI.

The downstream can request retransmission for lost original packets using RGIs and control the redundancy level to be added using the control interest (CNI). In the case where the downstream router changes the uplink used for receiving data packets C_n^q to another uplink, the SMI is sent through the new uplink and a CNI is sent through the previous uplink for notifying the upstream router to stop the data forwarding in order to avoid receiving duplicate data packets.

In the data plane, based on the PIT entries, each router immediately sends the received real-time original video data packets to the downstream router(s). After receiving coded data packets from the upstream router(s), each router sends them only when the total number of transmitted original and coded data packets within the same coding group can be k. This procedure avoids unnecessary transmission of coded redundant data packets; if the total number of data packets is less than k, it is impossible for the consumer to recover lost original data packets. Each router caches the received original and coded data packets (during the acceptable end-

to-end delay time at most). Based on the CNI issued by the downstream routers, each router adjusts both the redundancy level to be added (if possible) and video quality to be sent. After receiving the RGI for requesting retransmission of the original data packet, the router retransmits the original data if it is cached. Otherwise, the router can transmit newly generated coded packets instead of the original data if a set of k packets within the same coding group is cached.

*3) **Path Probing**:* The simplicity brought by SMI requires a continuous probing of paths $l(i,j) \in \mathcal{E}$ at every node $i \in V$. Every node i examines the link metrics required for transmission adaptation (described later in Section III-B5): 1) the packet loss probability $P^{loss}_{l(i,j)}$, 2) the maximum and minimum values of RTT of the link $l(i,j)$, $RTT_{l(i,j)}$, and 3) the maximum and minimum values of propagation delay between a consumer u and the node i, $D_{(u,i)}$, and between a source s and the node j, $D_{(s,j)}$.

Packet loss can be detected with reference to the segment number and RLNC parameters such as k and the transmitted number of coded packets and its coding group $\mathcal{N}^q_{id}$, which are attached into the optional header of data packet. RLNC parameters to be inserted into this field are updated when nodes receive CNI requiring the parameter change. Nodes detect one round transmission of k packets is complete when receiving data belonging to the next coding group. The sampled packet loss probability $Ps_{l(i,j)}$ is periodically measured at a constant interval of 100 ms and calculated using an exponential weighted moving average with a smoothing factor α, as follows: $P_{l(i,j)} = \alpha \cdot P_{l(i,j)} + (1-\alpha) \cdot Ps_{l(i,j)}$. We chose $\alpha = 0.2$, which allows $P_{l(i,j)}$ to adapt to the changes of $Ps_{l(i,j)}$ rapidly.

To measure $RTT_{l(i,j)}$, a downstream node i sends the *empty* CNI periodically at interval of 1.0 s to the upstream node j. The node j returns the response after receiving it and the node i measures the difference between the sending time of the CNI and the receiving time of the response. Node j retains the maximum and minimum values, $RTT^{min}_{l(i,j)}$ and $RTT^{max}_{l(i,j)}$.

To obtain $D_{(u,i)}$, every node i refers to the maximum and minimum values specified in the elapsed time of the optional field of SMI, and adds $RTT^{max}_{l(i,j)}/2$ and $RTT^{min}_{l(i,j)}/2$ to the referred value, respectively. The elapsed time of the optional field of the SMI sent to a node j is replaced with the sum value. By referring to this field of the SMI, a node i can derive the maximum and minimum values of propagation delays, $D^{max}_{(u,i)}$ and $D^{min}_{(u,i)}$. In the case of $D_{(s,j)}$, the elapsed time of the optional field of the data is used in the same manner as $D_{(u,i)}$ using SMI. Every node periodically inputs the sum value into the optional field of the data packet. A node i refers to the elapsed time and derives and updates the max/min value of $D_{(s,j)}$ by calculating the value minus the max/min $RTT_{l(i,j)}/2$. In the next section, we describe the L4C2 adaptation algorithm in detail.

*4) **Deriving Metrics for L4C2**:* L4C2 uses the success probability of the receiving the data within the estimated acceptable delay to execute the efficient data transmission for all the receiving video flows $\mathcal{F}$.

Let us now consider how to derive the success probability of the data transmission by focusing on the link $l(i,j)$. Because the time that is required for node i to receive the requested data is at least $RTT_{l(i,j)}$, the maximum number of times that node j can send an arbitrary data packet considering the first transmission and retransmission is given by:

$$Tr^{max}_{l(i,j)} = \lfloor (2D^{allow}_{l(i,j)} + RTT_{l(i,j)})/2RTT_{l(i,j)} \rfloor \quad (1)$$

where $D^{allow}_{l(i,j)}$ is the estimated acceptable delay in the link, and we assume $Tr^{max}_{l(i,j)} \geq 1$. When we consider only retransmission to recover lost data, the success probability of the data transmission subject to $D^{allow}_{(i,j)}$ is given by:

$$P^{Ret}_{l(i,j)} = 1 - (P^{loss}_{l(i,j)})^{Tr^{max}_{l(i,j)}} \quad (2)$$

We now consider that node j applies both retransmission and RLNC. Node j transmits k original data packets and $n-k$ coded packets. When the number of original and coded packets received by node i is not sufficient (i.e., less than k) owing to packet losses, node i requires node j to transmit newly generated coded packets to receive k packets in total. The probability to receive k packets when node j sends n packets is given by:

$$P(n,k) = \binom{n}{k} (1 - P^{loss}_{l(i,j)})^k (P^{loss}_{l(i,j)})^{n-k} \quad (3)$$

Assuming that there is no loss of interest packets for requesting retransmission, when node i receives Rc_x packets in the x-th $(1 \leq x \leq Tr^{max}_{l(i,j)})$ transmission from node j, the probability that the total number of received packets becomes Rc^{total}_x up to the x-th transmission is derived by:

$$\begin{aligned} P(Rc^{total}_x) = P(n, Rc_1) P(n - Rc_1, Rc_2) ... \\ P(n - \sum_{m=0}^{x-1} Rc_m, Rc_x) \end{aligned} \quad (4)$$

where $Rc^{total}_x = \sum_{m=1}^{x} Rc_m$.

Let N_{loss} denote the number of lost packets in the n consecutive packets. The expected number of lost packets is given by:

$$E[N_{loss}] = \sum_{j=1}^{k-1} (n-j) P(Rc^{total}_x = j) \quad (5)$$

The success probability of data transmission in the case of using RLNC is given by:

$$\begin{aligned} P^{NC(n,k)}_{l(i,j)} &= 1 - (E[N_{loss}]/n) \\ &= 1 - (\sum_{j=1}^{k-1} (n-j) P(Rc^{total}_x = j))/n \end{aligned} \quad (6)$$

The transmission success probability on $l(i,j)$ strongly depends on $D^{allow}_{l(i,j)}$ and therefore, to accurately derive this, node i must know the time required for each data to traverse $l(u,i)$ and $l(j,s)$. However, it is difficult for node i to estimate the accurate time owing to the diversities on network conditions of consumers and variations in the network condition on each

link. Thus, L4C2 uses the average value of the transmission success probability as follows:

$$P^{suc}_{l(i,j)} = \frac{1}{(D^{MaxAlw}_{l(i,j)} - D^{MinAlw}_{l(i,j)})} \int_{D^{MinAlw}_{l(i,j)}}^{D^{MaxAlw}_{l(i,j)}} P^{\mathcal{W}}_{l(i,j)}(x)dx, \quad \mathcal{W} \in \{Ret, NC\} \tag{7}$$

where $\mathcal{W}$ is the transmission method using only retransmission (Ret) or applying RLNC (NC). Node i can derive both $D^{MaxAlw}_{l(i,j)}$ and $D^{MinAlw}_{l(i,j)}$ as follows:

$$D^{MaxAlw}_{l(i,j)} = D^{allow}_{f} - (D^{min}_{(u,i)} + D^{min}_{(s,j)}) \tag{8}$$

$$D^{MinAlw}_{l(i,j)} = D^{allow}_{f} - (D^{max}_{(u,i)} + D^{max}_{(s,j)}) \tag{9}$$

where D^{allow}_{f} is the acceptable one-way delay defined per video flow f.

*5) **Adaptation of L4C2**:* Algorithm 1 presents the pseudo-code for the acceptable delay aware adaptation of L4C2. Using the transmission success probability $P^{suc}_{l(i,j)}$ and available bandwidth $B^{avail}_{f,l}$ for flow f, L4C2 adjusts the video rate r and the redundancy level (i.e., $n-k$) if necessary, such that the transmission success probability $P^{suc}_{l(i,j)}$ remains greater than the threshold value η. $B^{avail}_{f,l}$ is determined based on the bandwidth sharing policy as described in Section III-A. Under this condition, the algorithm aims at maximizing the receiving video quality for each flow $f \in \mathcal{F}$ while minimizing the amount of redundant data traffic.

When only the retransmission method is sufficient to maintain $P^{suc}_{l} \geq \eta$ (in Line 5), a node i does not ask its upstream router j to transmit redundant coded data to avoid wasting bandwidth with redundant data transmission. If the success probability of retransmission becomes less than η, using Eq.(6) and (7) L4C2 searches the minimum redundancy level considering the available bandwidth of the link (in Line 8) such that it achieves the success probability of η at least, and asks node j to apply NC(N_{min}, k) if any exists.

When there are a small number of available Faces (i.e., links), node i utilizes each link to maximize the received video quality q for each flow $f \in \mathcal{F}$ through the links. Because SMI cannot be divided in such a manner as to divide RGIs per sequence number, node i divides the SMIs per video quality q and sends them to the links considering the available bandwidth of each link and the video rate r_q. The dividing algorithm is greedy in the sense that it allocates the available bandwidth to the most valuable SMI at first and then fills the remaining network resources with less valuable SMIs. In the case of SVC, the SMI for the base-layer video (i.e., q_1), which is the most valuable for playback of the video, will be sent to the link with the greatest available bandwidth (as indicated in Lines 2 and 3). If the video is encoded to generate independent video streams with different video quality (called simulcast), the available bandwidth is allocated to the highest video quality.

Once determining the transmission method to request for flow f (e.g., $\mathcal{W} = NC, N_{min} = k+1, q = q_3$), the corresponding CNI is sent through the uplink l. The time complexity of L4C2 adaptation is not strongly impacted by the number of video flows $|\mathcal{F}|$. Indeed, the number of provided video qualities $|\mathcal{Q}_f|$ is typically small (e.g., 5), and furthermore, the time complexity of searching the minimum redundancy level (Line 8) using Eq.(6) remains low, as both $Tr^{max}_{l(i,j)}$ and k are small (at most 10) due to the small acceptable delay of data transmission. Hence, the asymptotic time complexity is of polynomial time of $\mathcal{O}(\sum_{f \in \mathcal{F}} |\mathcal{Q}_f|)$.

Algorithm 1 L4C2 Adaptation

INPUT:
η: the threshold value of success probability,
$B^{avail}_{f,l}$: the available bandwidth of the link $l \in \mathcal{E}$ for flow f,
$\mathcal{L}(f) = \{l \in \mathcal{E} : B^{avail}_{f,l} > 0\}$: the set of links flow f can use,
$\mathcal{Q}_f$: the set of video qualities for flow f specified by received SMIs
OUTPUT:
$w(q,l) = \{w \in \mathcal{W} : q \in \mathcal{Q}_f, l \in \mathcal{L}(f)\}$: the transmission method to request for flow f.

```
1:  for all f ∈ F do
2:    for all q ∈ Q_f /* in ascending order */ do
3:      l ← Get_Link_MaxBavail(L(f));
4:      if r_q ≤ B^avail_{f,l} then
5:        if P^suc_l ≥ η (W = Ret) then
6:          w(q,l) ← Ret /* retransmission only*/
7:        else
8:          if Search_NC_N_min(B^avail_{f,l}) then
9:            /* add redundancy */
10:           w(q,l) ← NC(N_min, k)
11:         else
12:           w(q,l) ← NULL
13:         end if
14:       end if
15:     else
16:       w(q,l) ← NULL
17:     end if
18:   end for
19: end for
```

IV. Evaluation

A. Simulation Setup

In this section, we evaluate the performance of L4C2 to validate the adaptation mechanism and demonstrate that L4C2 outperforms the existing one. We used an NS-3 based NDN simulator (ndnSim) [3] and extended it for L4C2 and for the multipath congestion control mechanism proposed in [25] (hereafter, called as MCC), respectively. In the case of MCC multipath data retrieval, routers send received interest packets to upstream routers using the proposed weighted round robin logic by monitoring the number of pending interests on each Face. Concerning L4C2 parameters in all the simulations, the threshold value $\eta = 0.995$ was used, and the k value was set 5,

where the time required to buffer all the successive k original packets was short (e.g., at least 2.1 ms at a rate of 35 Mbps). L4C2 consumers set SMI lifetime to 5.0 s, and sent an SMI per 2.0 s.

All the sources of real-time video transmitted video data at a rate of 35 Mbps, which consists of three video layers; the rate of the base-layer/enhanced-layer-1/enhanced-layer-2 was set to 20 Mbps/10 Mbps/5 Mbps. The interest and data packet size in bytes were set to 120 and 1,024 respectively.

To conduct a performance comparison of L4C2, a slightly modified MCC was used (hereafter, called as MCC-Video). MCC-Video at consumer side adjusts the interest sending rate to receive video data at a rate between 20 Mbps and 35 Mbps. Hence, consumers try to receive real-time video data of basic-layer at least.

B. Performance Metrics

We used the following performance metrics for L4C2.

1) Data Transmission Success Ratio:
This value denotes the ratio of total number of data packets that consumer received within the acceptable delay, to the total number of data packets the real-time video source transmitted.

2) Redundancy Ratio:
This value is defined as $(n-k)/k$.

3) Normalized QoE:
The key metrics of QoE are the following: 1) the average video quality, 2) the average quality variations, and 3) the unplayable data rate. Let $VQ(\cdot) : \mathcal{R} \rightarrow \mathbf{R}_+$ be the function that maps the video rate $r \in \mathcal{R}$ to video quality experienced by a consumer receiving the video. With reference to the QoE model [11], we defined the QoE of flow f at time t, using a weighted sum of the above three metrics:

$$QoE_f(t) = VQ(\overline{r}(t)) - \lambda |VQ(\overline{r}(t)) - VQ(\overline{r}(t-1))| - \sum_{L=0}^{2} \mu_L N_{loss}^L \quad (10)$$

where $\overline{r}(t)$ denotes the average rate of received video data per second at time t; N_{loss}^L denotes the number of unplayable data packets of video layer L; λ and μ_L for video layer L denote the positive weighing parameters, which reflect the QoE affected by user preferences concerning the smoothness of video rate changes and video disruptions of caused by data loss. We used the normalized value defined as $Nml\_QoE_f(t) = QoE_f(t)/QoE_f^{max}$, where QoE_f^{max} denotes the maximum value when the real-time video flow keeps the highest data rate without data loss. Considering the video rate of each video layer, we set $\lambda = 1$, and $\mu_0 = 50,000, \mu_1 = 25,000, \mu_2 = 12,500$.

4) Total Interest/Data Traffic Rate:
We observed the total transmission rate of interest and video data packets at each node, in the expectations of 1) low interest rate thanks to SMI, and 2) fairly competing with non-real-time data flows.

5) Non-real-time Flow Friendliness Index:
We observed the average data throughput of non-real-time MCC flows at consumer, and defined an index of friendliness to non-real-time data flows as:

$$NRF_{index} = \frac{NonRealTime\_Thuput}{RealTimeVideo\_Thuput} \quad (11)$$

where $RealTimeVideo\_Thuput$ is the result of the average data throughput of L4C2 or MCC-Video flows; $NonRealTime\_Thuput$ is the result of the average throughput of MCC flows.

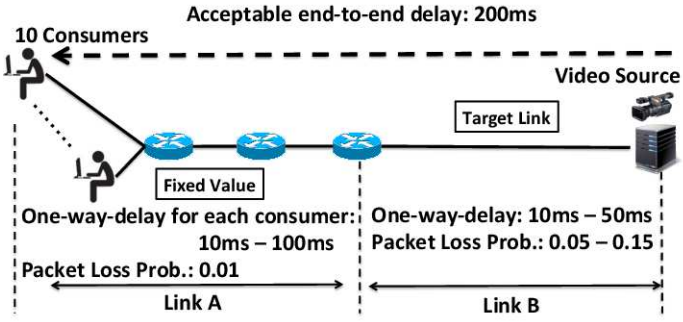
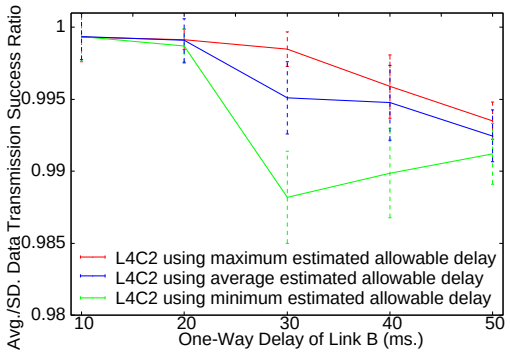
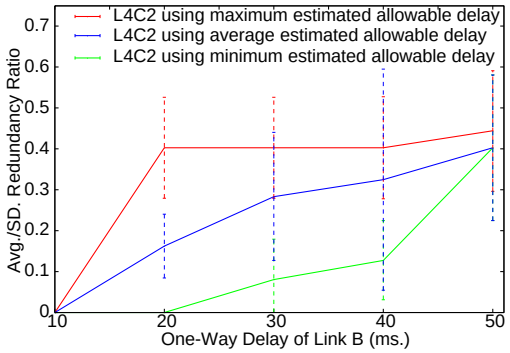


Fig. 2. Network model, where the access link delay of each consumer varies, which largely affects the estimated acceptable delay used for the adaptation at the router in the link B. The one-way delay and packet loss probability of the link B vary, which also influences the adaptation method at the router.



(a) Average data transmission success ratio under the acceptable end-to-end delay of 200ms



(b) Average redundancy ratio added at the router in the link-B

Fig. 3. The results of L4C2 adaptation, compared to the case of using estimated maximum/minimum acceptable delay.

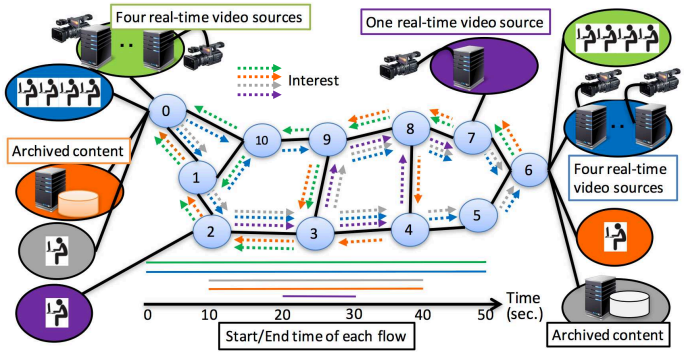


Fig. 4. Network topology used for the evaluation scenario.

C. Delay Diversity Impact on L4C2 Success Probability

The performance of the L4C2 adaptation scheme depends on whether or not each node i can accurately set $D^{allow}_{l(i,j)}$ in the link for each data. However, knowing the actual $D^{allow}_{l(i,j)}$ in advance is difficult due to the uncertainty of network conditions, especially in the case that there are a number of consumers in different links. To deal with this issue, L4C2 uses the average value of the transmission success probability with the estimated $D^{allow\_max}_{l(i,j)}$ and $D^{allow\_min}_{l(i,j)}$, in order to determine whether or not to add redundancy and its level. From this viewpoint, we evaluate the association between network conditions and the L4C2 adaptation method of applying redundancy and its impact on the data transmission success ratio.

Fig. 2 shows our network model. In accordance with the link A, the estimated $D^{allow\_max}_{link\_B}$ and $D^{allow\_min}_{link\_B}$ used at the router in the link B were 140 ms and 50 ms, respectively. We changed the one-way delay in the link B from 10 ms to 50 ms, and the packet loss probability from 0.05 to 0.15. The acceptable end-to-end delay was set to 200 ms.

Fig. 3 shows the performance results of L4C2, and its variants using only $D^{allow\_max}_{link\_B}$ or $D^{allow\_min}_{link\_B}$. As we can see in Fig. 3(a), all the average data transmission success ratio becomes almost equal or greater than $\eta = 0.995$, except the case of using the maximum estimated acceptable delay, thanks to the retransmission and/or redundancy using in-network coding and caching. Regarding the average ratio of redundancy added by the router in the link B, because the two L4C2 variants consider the consumer in the longest or shortest link, the applied redundancy ratio results in a higher or lower redundancy ratio than L4C2, as shown in Fig. 3(b). Such a condition leads to a significant performance issue when there are many consumers in widely different links, because unnecessary redundant data traffic is likely to occur or small redundancy cannot improve the data transmission success ratio within the acceptable delay. In this context, the L4C2 adaptation method is efficient.

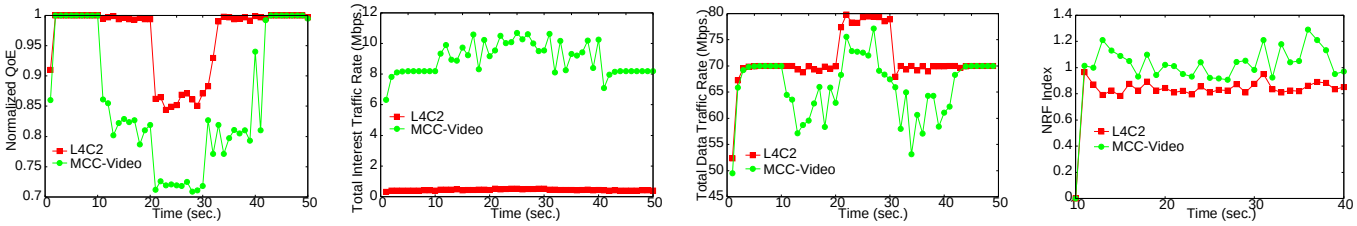
D. Competition between Real-Time and Non-Real-Time Flows

Fig. 4 shows the Abilene network topology used for this evaluation scenario, where we observed interactions between real-time and non-realtime flows, and investigated the performances of L4C2 multipath data retrieval using the rate control adaptation method. The one-way propagation delay and capacity of each link were set to 5 ms and 100 Mbps, respectively. A DropTail queue was used, and the queue size in packets was set to 100. At the beginning, the eight real-time video flows were generated from the corresponding video sources in the green and blue color area. Another real-time video source in the purple area transmitted video data from 20 s to 30 s. The consumers requesting real-time video data used L4C2 or MCC-Video. In the case of L4C2, all the routers fairly allocated the maximum available bandwidth to real-time video flows so that the total consumed bandwidth does not exceed the estimated total bandwidth consumed by competing non-real-time data flows per Face. From 10 s to 40 s, the two non-real-time data flows were generated from the archived content sources in the orange and gray color area. The corresponding consumers used MCC. The acceptable end-to-end delay of all the real-time flows was set to 150 ms.

Fig. 5 shows the trace results of the performance of L4C2 and MCC-Video flows. As shown in Fig. 5(a), L4C2 achieved higher normalized QoE, except during the time between 20 s and 30 s. This is because data losses caused by network congestion were immediately recovered, and the highest video quality was maintained. Furthermore, SMI drastically suppressed the interest traffic and enabled consumers to continue receiving video data even when interest loss occurred. The total interest traffic rate of MCC-Video at link(9,8) in which the most amount of interest packets traversed was more than about 8 Mbps, while that of L4C2 was less than about 0.5 Mbps (Fig. 5(b)). The corresponding data traffic rate of L4C2 became about 80 Mbps from 20 s to 30 s (Fig. 5(c)). This means that although the total maximum rate of real-time data flows exceeds the link capacity of 100 Mbps after the occurrence of the new real-time data flow at 20 s, L4C2 rate control adaptation method with multipath data retrieval effectively utilized network resources so as to archive the best possible QoE. Meanwhile, although MCC-Video can gain the multipath data retrieval as well as L4C2, MCC-Video suffered from the QoE degradations from 10 s to 40 s. This is because, 1) MCC-Video cannot benefit from reducing the number of high-rate interests, 2) the retransmission scheme often failed due to the constraint of acceptable end-to-end delay, and 3) the interest rate control method with the AIMD algorithm at a consumer tended to decrease the video quality to receive (i.e., after packet loss detection, the interest sending rate was reduced by half at most). Such drops in video quality posed lower QoE, while the decrease in data traffic results in higher NRF_{index} than L4C2 (Fig. 5(d)). On the other hand, owing to the L4C2 adaptation method, NRF_{index} of L4C2 achieved more than about 0.8 with higher normalized QoE.

V. Conclusion

We proposed a low latency, low loss streaming mechanism, named L4C2. L4C2 exploits the architectural benefits of CCN/NDN such as in-network caching and multipath data



(a) Normalized QoE of real-time video streaming flows (b) Interest traffic rate at link (9,8) (c) Data traffic rate at link (8,9) (d) Non-real-time flow friendliness index

Fig. 5. The trace results of L4C2 and MCC-Video flows that compete with each other and the non-real-time content flows.

retrieval to maximize the streaming quality while suppressing the duplicate and redundant data packets. A large number of signaling messages, known as interest packets in CCN/NDN, and required for high-quality streaming reception, are also suppressed by using symbolic interest. Based on the estimated transmission success probability considering the acceptable delay, L4C2 utilizes a retransmission scheme using in-network cache and in-network coding and adjusts the video rate and redundancy level in a hop-by-hop fashion. To execute congestion control, the bandwidth required each video flow can be fairly allocated to maximize the video quality. We analyzed the efficacy of the L4C2 adaptation scheme based on transmission success probability and evaluated the performance of L4C2 using a comprehensive simulation confirming that the proposed mechanism was more feasible for high-quality delay-sensitive streaming than the proposed multipath congestion control mechanism.

In our future work, we will make deeper analysis of the optimality and approximation of the adaptation scheme. Finally, we will investigate a mechanism to make L4C2 routers quickly execute state management and timers using an actual implementation, and evaluate L4C2's performance in a real-world setting.

ACKNOWLEDGMENT

This work has been partially supported by the Japanese Society for the Promotion of Science (JSPS) and Inria in the context of the UHDon5G associated team.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," Proc. ACM CoNEXT, Dec. 2009, pp.1–12
[2] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," ACM Comput. Commun. Rev., vol. 44, no. 3, 2014, pp. 66–73.
[3] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, 2012.
[4] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019 White Paper.
[5] P. Seeling and M. Reisslein, "I. want. pixels. (Entering the age of 4K)," IEEE Potentials, vol. 33, no. 6, 2014, pp. 27–30.
[6] P. Troubil, H. Rudová, and P. Holub, "Media streams planning with transcoding," Proc. IEEE International Symp. Network Computing and Applications, 2013, pp. 41–48.
[7] V. Jacobson, D. Smetters, N. Briggs, M. Plass, J. Thornton, and R. Braynard, "VoCCN: Voice-over content centric networks," Proc. ACM ReArch Workshop, Dec. 2009, pp. 1–6.
[8] Z. Zhu, S. Wang, and X. Yang, "ACT: Audio conference tool over named data networking," Proc. ACM SIGCOMM ICN Workshop, Aug. 2011, pp. 68–73.
[9] D. Han, A. Anand, A. Akella, and S. Seshan, "RPT: re-architecting loss protection for content-aware networks," Proc. USENIX NSDI, 2012.
[10] ITU-T recommendation G.114, International Telecommunication Union, Tech. Rep., 2009.
[11] X. Yin, V. Sekar, and B. Sinopoli, "Toward a principled framework to design dynamic adaptive streaming algorithms over HTTP," Proc. ACM Hotnets, Oct. 2014.
[12] H. Schwarz, D.Marpe and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," IEEE Trans. Circuits and Syst. for Video Tech., vol. 17, no. 9, Sept. 2007, pp. 1103–1120.
[13] L. Pantel and L. C. Wolf, "On the impact of delay on real-time multiplayer games," Proc. NOSSDAV, 2002, pp. 23–29.
[14] M. -J. Montpetit, C. Westphal, and D. Trossen, "Network coding meets information-centric networking: an architectural case for information dispersion through native network coding," Proc. ACM NoM Workshop, 2012, pp. 31–36.
[15] J. Saltarin, E. Bourtsoulatze, N. Thomos, and T. Braun, "NetCodCCN: a network coding approach for content-centric networks," Proc. IEEE INFOCOM, 2016.
[16] D. Loguinov and H. Radha, "On retransmission schemes for real-time streaming in the Internet," Proc. IEEE INFOCOM, 2001, pp. 1310–1319.
[17] D. Rossi and G. Rossini, "On sizing CCN content stores by exploiting topological information," Proc. IEEE INFOCOM, NOMEN Workshop, 2012, pp. 280–285.
[18] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, and H. Hellwagnet, "Adaptive streaming over content centric networks in mobile networks using multiple links," Proc. IEEE ICC, 2013, pp. 677–681.
[19] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," Proc. ACM Comput. Commun. Rev., vol. 27, no. 2, 1997, pp. 24–36.
[20] K. Matsuzono, J. Detchart, M. Cunche, V. Roca, and H. Asaeda, "Performance analysis of a high-Performance real-time application with several AL-FEC schemes," Proc. IEEE LCN, 2010, pp. 1–7.
[21] R. Ahlswede, N. Cai, S.-Y, Li, and R. Yeung, "Network information flow," IEEE Trans. Information Theory, vol. 46, no. 4, July 2000, pp. 1204–1216.
[22] M. Pedersen, J. Heide, P. Vingelmann, and F. Fitzek, "Network coding over the $2^(32)$ – 5 prime field," Proc. IEEE ICC, 2013, pp. 2922–2927.
[23] K. Matsuzono and H. Asaeda, "NRTS: content name-based real-time streaming," Proc. IEEE CCNC, Jan. 2016, pp. 537–543.
[24] G. Carofiglio, M. Gallo and L. Muscariello, "Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks," Proc. ACM SIGCOMM ICN Workshop, Aug. 2012, pp. 37–42.
[25] G. Carofiglio, M. Gallo, L. Muscariello and M. Papalini, "Multipath congestion control in content-centric networks," Proc. IEEE ICNP, 2013, pp. 363–368.
[26] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," Proc. IEEE ISIT, 2004.
[27] P. Gusev and J. Burke, "NDN-RTC: real-time videoconferencing over named data networking," Proc. ACM ICN'15, 2015, pp. 117–126.
[28] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagnet, "Adaptive multimedia streaming in information-centric networks," Proc. IEEE Network Magazine, vol. 28, no. 6, 2014, pp. 91–96.
[29] T. Stockhammer, "Dynamic adaptive streaming over HTTP-design principles and standards," Proc. ACM MMSys, 2011, pp. 133–144.