This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



# Content-Aware Delivery of Scalable Video in Network Coding Enabled Named Data Networks

Eirina Bourtsoulatze, *Member, IEEE*, Nikolaos Thomos, *Senior Member, IEEE*, Jonnahtan Saltarin, *Student Member, IEEE*, and Torsten Braun, *Senior Member, IEEE*

***Abstract*—We propose a novel network coding (NC) enabled named data networking (NDN) architecture for scalable video delivery. Our architecture utilizes network coding in order to address the problem that arises in the original NDN architecture, where optimal use of the bandwidth and caching resources necessitates the coordination of the Interest forwarding decisions. To optimize the performance of the proposed network coding based NDN architecture and render it appropriate for transmission of scalable video, we devise a novel rate allocation algorithm that decides on the optimal rates of Interests sent by clients and intermediate nodes. The flow of Data packets achieved by this algorithm maximizes the average quality of the video delivered to the client population. To support the handling of Interest and Data packets when intermediate nodes perform network coding, we introduce the use of Bloom filters, which store efficiently additional information about the Interest and Data packets, and modify accordingly the standard NDN architecture. We also devise an optimized Interest forwarding strategy that implements the target rate allocation. The proposed architecture is evaluated for transmission of scalable video over PlanetLab topologies. The evaluation shows that the proposed scheme exploits optimally the available network resources.**

***Index Terms*—Network coding, named data networking, scalable video, rate allocation, forwarding strategy.**

## I. Introduction

In the last decade, we have witnessed a radical change of the video production and communication model. Besides their traditional role as content consumers, users nowadays are able to produce and share their own video content. This change has been fostered by the emergence of affordable-price camera-enabled mobile devices and has resulted in video data dominating the overall IP traffic [1]. The video users typically present significant heterogeneity in terms of display capabilities, processing power, network connectivity, *etc*., which necessitates encoding the video in multiple qualities and resolutions [2]. While this enables users to access the video of their interest encoded in a quality and resolution that matches the capabilities of their device, it also necessitates efficient mechanisms to enable the delivery of scalable data to heterogeneous clients. The Internet Protocol, designed originally for elastic traffic applications, such as messaging and file downloading, fails to deal efficiently with the growing volume of inelastic video traffic. The introduction of protocols like MP-TCP [3], DASH [4], [5] and RTP/RTSP have permitted to partly handle the delivery of large volumes of video traffic. However, these solutions require the establishment of end-to-end connections for each host which makes the use of in-network caching and the dynamic selection of the sources difficult.

To cope with the inefficiencies of the IP host-centric communication model such as scalability, mobility, *etc*., Information-Centric Networking (ICN) [6] has been proposed as alternative solution. The ICN paradigm focuses on the name of the content rather than on its location. The content is searched by its name and can be retrieved from any location where it is permanently or temporarily stored without the need to establish multiple dedicated server-client connections. This content-centric approach makes use of the available network caching capacity and reduces the redundancy of the transmitted content, while it also can exploit efficiently existing network links through multipath data delivery. Thus, there is a significant potential for using the content-centric architectures for the delivery of growing volumes of video data.

Among the existing ICN architectures, Named Data Networking (NDN) [7] has gained significant popularity because of its intuitive naming scheme and the way content requests and data forwarding are handled. The NDN model is receiver driven. The client initiates the content delivery by sending Interest packets with the name identifier of the requested content on its outgoing faces. Once an Interest reaches an uplink node, the node's cache is searched for a matching Data packet. If the requested data exists, it is forwarded backwards on the requesting face. If there is an indication that the data will be available at a later time, *e.g.* another Interest for the same Data packet is pending, the Interest is kept at the node and consumed later when the data arrives. Otherwise, the Interest is forwarded to other nodes according to the employed forwarding strategy. The Data packets that travel towards the end users can be cached in the intermediate nodes and can be used later to consume new requests for the same content.

Although solutions exist that permit to deal with real-time and on-demand video delivery, it is generally acknowledged that ICN architectures are not yet video ready [8]. For example, in live streaming the main bottleneck of NDN is that each Data packet is independently requested by Interests, which increases the network load and raises scalability issues. To address this problem, Interest aggregation [9] and persistent Interest packets [10] can be used. However, the use of such approaches is not trivial, as the loss of a single Interest packet

E. Bourtsoulatze is with the Imperial College London, London, United Kingdom (e-mail: e.bourtsoulatze@imperial.ac.uk). N. Thomos is with the University of Essex, Colchester, United Kingdom (e-mail: nthomos@essex.ac.uk). J. Saltarin and T. Braun are with the University of Bern, Bern, Switzerland (e-mail: {saltarin, braun}@inf.unibe.ch). This work was done while E. Bourtsoulatze was with the University of Bern.

This work has been supported by the Swiss National Science Foundation under grant number 149225.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



can lead to the loss of multiple Data packets. In VoD systems, the use of ICN is problematic due to the lack of reliable estimations of the available end-to-end bandwidth. Adaptive video streaming over ICN is achieved by deploying the DASH protocol over NDN [11]. This is driven by the conceptual similarities of the NDN and DASH protocols. The use of the DASH protocol results in significant performance gains and allows the use of the multiple interfaces of the devices.

The requirement to request each packet of a data stream explicitly by sending an Interest packet can be relaxed by equipping the NDN architecture with network coding capabilities [12]. Network coding [13] can improve the use of the network resources [14], simplify the scheduling, remove the need for coordination, *etc.* With network coding the intermediate nodes linearly combine the received packets prior to forwarding them on the outgoing links. To deploy network coding in practical settings, Randomized Linear Network Coding (RLNC) [15] has been proposed. A header containing the network coding coefficients is prepended to the packets and permits the users to decode the linear combinations. The introduction of the concept of generations [16] helps to limit the overhead information carried by each network coded packet and renders it appropriate for video transmission.

The potential of network coding has motivated researchers to explore the use of coding enabled NDN variants. Montpetit *et al.* propose an architecture called NC3N [12]. In this approach, Interests have a new field, which contains the Data packet availability information of the client. Nodes storing Data packets that match the name prefix of the received Interest, reply only if they can provide a novel network coded Data to the client. However, when there are multiple clients requesting the same content, the aggregation of Interests and the pipelining are problematic. Inspired by NC3N [12], CodingCache [17] focuses on the caching problems and shows that cache diversity due to network coding increases the cache hit rate. However, CodingCache suffers from the same drawbacks as NC3N, namely, the Interest aggregation and Interest pipelining are problematic. In the work presented by Llorca *et al.* [18], multicast delivery in network coding enabled ICN is optimized by finding the evolution of the Data packets that are cached in the network. However, this approach needs a central entity that is aware of the network topology and the Interests, which does not scale well with the number of network nodes. Matsuzono *et al.* [19] have proposed L4C2, a network coding enabled mechanism for low latency, low loss video streaming over CCN. In L4C2, the network nodes estimate the acceptable delay and Data packet loss rate in their uplinks, adjusting the requested video quality accordingly. The clients first request non-network coded Data packets, and only request network coded Data packets when they detect Data packet losses. In this case, network coding is only exploited to deal with lost Data packets. In our recent work, we have proposed the NetCodCCN architecture [20], a network coding enabled CCN protocol which permits Interest aggregation and pipelining, and thus reduces the data retrieval times. Alternatively, Raptor codes have been examined in [21], where RC-NDN, a variant of the NDN, is presented. This scheme shows the benefits of using Raptor codes in mobile networks.

In this paper, we propose a content aware video delivery scheme for network coding enabled NDN architectures. We focus on the transmission of scalable video [2] in order to deal with the heterogeneous requests of users with diverse demands in terms of the video quality. We employ prioritized random linear network coding (PRLNC) [14], [22] in order to respect the unequal importance of the video layers. PRLNC is applied in an embedded way forming packets that belong to classes of decreasing significance. We first present the new features of our NC enabled NDN architecture. These new features include the appropriate naming scheme and the processing functions that permit to handle both the Interest and Data packets when the intermediate nodes perform network coding. In order to deal with the ambiguity that arises from the use of content names that do not specify unique Data packets but rather a set of network coded packets belonging to the same class and generation, we propose the use of Bloom filters that compactly store additional information about the Interest and Data packets. We then derive the optimal rate allocation for the transmission of Interests in order to achieve the flow of Data packets that maximizes the average video quality in the client population. Finally, we design the optimal content-aware Interest forwarding strategy based on the solution of the rate allocation problem. The forwarding strategy guarantees that a sufficient number of Interest packets will be optimally forwarded so that the innovative rate of Data packets remains sufficiently high. We evaluate the performance of the presented scheme for scalable video transmission with respect to the experienced video quality. The evaluation shows that the proposed method results in close to optimal performance in terms of the achieved video quality. Though the proposed scheme is evaluated for H.264/SVC coded video, our scheme is general and can be used with any type of scalable data like scalable HEVC data and multiview scalable data.

In summary, the main contributions in this paper are

- We propose a novel content-aware network coding enabled NDN architecture appropriate for delivering layered data, in general, and scalable video, in particular. For that purpose, we redesign the functions that handle the Interest and Data packets in order to enable the processing of network coded content.
- We formulate the optimal flow rate allocation problem for the transmission of Interest packets so that the achieved rate of Data packets maximizes the average video quality over the client population. Based on the optimal solution of the flow rate allocation problem, we design an optimized forwarding strategy that handles the forwarding of Interest packets.
- We propose the use of Bloom filters in order to resolve problems related to the introduction of network coding in the NDN architecture.
- Finally, we implement the components of the proposed network coding enabled NDN architecture in a NS-3 based simulator and evaluate our Interest forwarding strategy in a network topology taken from the PlanetLab project.

The rest of the paper is organized as follows. In Section II,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



we present the overview of the system and discuss the problem of optimally delivering layered video in our setting. Section III discusses the new features of the network coding enabled NDN architecture. Next, in Section IV, we present the flow rate allocation problem for the transmission of Interest packets along with our subgradient based optimization algorithm, and describe our optimized Interest forwarding strategy. The performance evaluation of the proposed architecture is presented in Section V. Section VI summarizes the paper.

## II. System description

### A. Network and source models

We consider a multi-hop wireline network with links that can simultaneously carry information in both directions between pairs of nodes. The network is modelled by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ and $\mathcal{E}$ denote the set of network nodes and links, respectively. The set $\mathcal{V}$ consists of a server node $s$, a set of intermediate nodes $\mathcal{N}$ and a set of client nodes $\mathcal{U}$, *i.e.*, $\mathcal{V} = s \cup \mathcal{N} \cup \mathcal{U}$. For notational convenience, we assume that every physical link is modelled by a pair of directed links in the graph $\mathcal{G}$, such that $(i,j) \in \mathcal{E}$ iff $(j,i) \in \mathcal{E}$, where the tuple $(i,j)$ denotes the directed link from node $i$ to node $j$. Assuming this convention, the set of network links $\mathcal{E}$ can be written as $\mathcal{E} = \mathcal{E}_I \cup \mathcal{E}_D$, such that $\mathcal{E}_I \cap \mathcal{E}_D = \emptyset$ and $(i,j) \in \mathcal{E}_I$ iff $(j,i) \in \mathcal{E}_D$. The subgraph $\mathcal{G}_I = (\mathcal{V}, \mathcal{E}_I)$ is used to transmit the Interest packets from the clients towards the server, while the subgraph $\mathcal{G}_D = (\mathcal{V}, \mathcal{E}_D)$ is used to transmit the Data packets in the opposite direction. We further assume that the directed graphs $\mathcal{G}_I$ and $\mathcal{G}_D$ are acyclic, which ensures that the information does not cycle within the network. Each pair of links that corresponds to a single physical duplex channel is characterized by the cumulative transmission rate $B_{ij}$ in bits per second (bps) that can be allocated proportionally to the traffic load in each direction.

The server $s$ generates video content, which is delivered to the clients following the NDN architecture. The video delivery is initiated by the clients that transmit Interests for the desired video content. The Interests are routed to the server according to the routing information stored in the nodes' Forwarding Information Base (FIB) tables until a matching Data packet is found. The Data packet is then transmitted back to the client following the reverse path of that followed by the Interest.

Due to the heterogeneity of the clients in terms of bandwidth and video display capabilities, the clients can request video content encoded at different bit rates so as to adapt the quality of the delivered video to the available resources. To meet the diverse clients' demands, the server $s$ encodes the video progressively into $L$ layers with the scalable extension (SVC) of the H.264/AVC standard [2]. The $l$-th video layer, with $l \in \mathcal{L} = \{0, 1, \ldots, L-1\}$, is encoded at rate $R_l$ (in packets/sec). The video layers include the base layer ($l = 0$), which provides the basic video quality, and $L-1$ enhancement layers, which offer an incremental video quality improvement. The $l$-th video layer can be decoded only if all the previous video layers, *i.e.,* $0, 1, \ldots, l-1$, have been successfully decoded [2]. The decoding dependencies between the video layers define a hierarchical structure with the base layer being assigned the highest level of importance, while each subsequent layer has a decreasing degree of importance.

### B. Prioritized delivery of scalable video

In order to improve network performance in terms of throughput and efficient use of available resources, the server and the intermediate nodes combine the video packets with Prioritized Random Linear Network Coding (PRLNC) [14]. To enable the network coding (NC) operations without incurring an additional decoding delay penalty [16], the source data is segmented into generations. Each generation comprises source video packets with similar decoding deadlines, *e.g.*, a Group of Pictures (GOP), since all network coded packets of a generation are decoded simultaneously. Without loss of generality, we consider that the data of the $l$-th layer in each generation is packetized into $\alpha_l$ packets and that the value of $\alpha_l$ is the same for all generations. Prior to transmission, the packets within each generation are encoded by means of PRLNC [14]. Specifically, the packets are first categorized into $L$ classes of decreasing importance, where the $l$-th class consists of source packets that belong to the first $l+1$ layers. The number of source packets that belong to the $l$-th class is $\beta_l = \sum_{k=0}^{l} \alpha_k$. Packets within each class are then encoded with RLNC. The coded packet that results from randomly combining the source packets from class $l$ is hereafter referred to as network coded packet of class $l$. A network coded packet of class $l$ is, therefore, a random linear combination of source packets from layers 0 to $l$.

The decoding of the network coded packets is done only at the client nodes by means of Gaussian elimination. A client can decode a generation of the $l$-th video layer only upon receiving $\beta_l$ innovative packets of this generation from classes $0, 1, \ldots, l$. A network coded packet is considered innovative with respect to a set of network coded packets, when it cannot be generated by linearly combining the packets in the set. Note that class $l$ contains up to $\beta_l$ innovative network coded packets. The PRLNC method respects the intrinsic prioritized structure of the video data by adding more redundancy to more important layers with smaller indices through coding. It thus avoids penalizing clients that do not have sufficient resources to jointly decode all the video layers and offers an adaptive data delivery solution for clients with heterogeneous resources.

### C. Optimal delivery of SVC video in NDN

In our NC enabled NDN architecture for scalable video delivery, we adopt the convention that every Data packet represents a single network coded video packet of some class $l$ and generation $g$. In order to respect the video delivery deadlines and meet the resource constraints of the clients, the flow of network coded Data packets to each client must be optimized. Let $f_{ji}^l$ denote the rate of the flow of Data packets of class $l$ on the link $(j,i)$ and let $\boldsymbol{f} = \{f_{ji}^l : (j,i) \in \mathcal{E}_D,\ l \in \mathcal{L}\}$ be the vector of flow rates. The average video quality in the client population can be written as

$$\overline{Q}(\boldsymbol{f}) = \frac{1}{U} \sum_{u \in \mathcal{U}} Q_u(\boldsymbol{f}^u) \tag{1}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



where $U$ is the number of clients in the network. $Q_u(\boldsymbol{f}^u)$ is the quality of the video delivered to client $u$ as a function of the vector $\boldsymbol{f}^u = \{\sum_{j:(j,u)\in\mathcal{E}_D} f^0_{ju}, \sum_{j:(j,u)\in\mathcal{E}_D} f^1_{ju}, \ldots, \sum_{j:(j,u)\in\mathcal{E}_D} f^{L-1}_{ju}\}$ of cumulative rates of Data packet flows of different packet classes on user's $u$ input links. Note that the quality of the video delivered to user $u$ depends only on the rates of the Data packet flows on the user's $u$ input links. The function $Q_u(\boldsymbol{f}^u)$ is a piecewise constant function. It can be expressed as a linear combination of indicator functions as

$$Q_u(\boldsymbol{f}^u) = \sum_{l=0}^{L-1} (q_l - q_{l-1}) \mathbb{1}_l(\boldsymbol{f}^u) \tag{2}$$

where $q_l$ is the video quality achieved after decoding the $l$-th video layer with $q_{-1} = 0$. The indicator function $\mathbb{1}_l(\boldsymbol{f}^u)$ is defined as $\mathbb{1}_l(\boldsymbol{f}^u) = 1$ if the $l$-th video layer can be decoded at user $u$ given $\boldsymbol{f}^u$; otherwise, $\mathbb{1}_l(\boldsymbol{f}^u) = 0$. Our goal is to maximize the average quality of the video delivered to the clients, and, therefore, we seek for the optimal values of the flow rates $\boldsymbol{f}$ that maximize the objective function in (1).

The Interest packets in our framework express a request for a network coded packet of a certain class and generation, and can be consumed by any network coded packet of the specified class and generation. Since the flow of Data packets in the NDN architecture directly depends on the way that the Interests are propagated in the network, an optimized Interest forwarding strategy is essential in order to deliver the desired flow of Data packets to each client. We assume that the clients generate Interests according to an optimized rate allocation policy and that every Data packet is sent on a network link $(j, i)$ in response to an Interest previously transmitted on the link $(i, j)$. Under these assumptions, the problem of delivering the video to the clients at the optimal flow rates $\boldsymbol{f}$ can be translated into a twofold problem, where we need: (i) to determine the rate of the Interest packets for each class of Data packets that must be transmitted from the clients and intermediate nodes, and (ii) to design the optimal Interest forwarding strategy at clients and intermediate nodes that in combination with the output of the rate allocation algorithm would achieve the optimal target set of flow rates $\boldsymbol{f}$.

## III. Network coding enabled NDN architecture

Before focusing on the solution of the problem introduced in Section II-C, we discuss the new features and functionalities of our network coding enabled NDN architecture. These new features enable the processing of the Data and Interest packets when network coding is performed in the NDN nodes.

### A. Naming

The naming scheme of our NC enabled NDN architecture follows the standard hierarchical naming of the NDN architecture and has the general format */component1/. . . /componentN/NCFlag/PacketId/GenIndx*. The *NCFlag* component enables the identification and subsequent processing of Interest packets that express interest for network coded data. The *NCFlag* is a binary flag. The "0" value signifies a request for an original source packet, while the "1" value corresponds to a request for a network coded Data packet. Thus, the *NCFlag* permits the nodes to distinguish between the two types of Interest packets and to invoke the appropriate processing functions. The *PacketId* specifies the requested packet. When the *NCFlag* is "0", the *PacketId* is interpreted as the sequence number of the packet within the generation. The sequence number along with the generation index uniquely identify the packet within the entire sequence of source video packets. When the request is for a network coded Data packet, the *PacketId* is understood as the class which the network coded Data packet belongs to. In this case, the name in the Interest no longer specifies a unique Data packet but rather refers to any network coded packet of this class and generation. The last new component of the naming structure is the *GenIndx*. It encodes the index of the generation which the packet belongs to.

### B. Bloom filter based processing

Unlike the original NDN architecture, where the content name in the Interest identifies a unique Data packet, in our NC enabled NDN architecture the Interest packets express a request for a network coded packet of a certain class and generation. In this case, the content name does not specify a unique packet, but rather a group of packets with similar information content. On the one hand, the random linear coding of data packets in the network nodes increases the content diversity in the network and facilitates the optimization of the forwarding strategy, since an Interest requesting a network coded Data packet of a certain class and generation can potentially be consumed by any available linear combination of packets of this class and generation. This information diversity significantly contributes to the efficient use of bandwidth and caching resources. On the other hand, from the client's perspective, this approach introduces some degree of ambiguity since clients issue multiple Interest packets with the same content name in order to obtain all the packets of a certain class and generation. This may lead to consuming two or more distinct Interests with the same linear combination of source packets, resulting in reduced innovative packet rate. Thus, in order to maintain a sufficiently high innovative rate of network coded packets and ensure the timely delivery of the video content, it is necessary to design a mechanism to resolve the ambiguity created by the lack of unique mapping between content names and the data.

The mechanism that enables the proper handling of Interest and Data packets relies on the use of Bloom filters [23] that store some additional information about the Interest and the Data packets. Specifically, we add a new field that contains a Bloom filter (BF) in both the Interest and the Data packets. This BF is a compact representation of the set whose elements are the IDs of the clients. When a client generates an Interest packet, it inserts its ID in the originally empty BF. As the Interest packet is forwarded towards the location of the data, the content of its BF is modified by the forwarding strategy according to the rules that will be explained in Section IV-C. The BF of an Interest can be interpreted as the set of clients that are the destination nodes for the Data packet that will

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



consume this Interest. By inspecting the BF of the Interest that arrives on a node's face, the node can decide whether the Interest can be (i) aggregated with other Interests stored in the Pending Interest Table (PIT), (ii) consumed by the Data packets that are stored in the Content Store (CS) or (iii) must be inserted in the PIT as a new entry and forwarded further towards the content location. For example, an Interest that arrives at a node can be aggregated with another Interest with the same content name stored in the PIT as long as the intersection of their BFs is empty. This is due to the fact that they can be consumed by the same Data packet since the Data packet will be delivered to disjoint subsets of clients. When the intersection of the two BFs is not empty, the subsets of clients who are the destination nodes for the Data packet that will consume the Interests are not disjoint. The clients that are present in both BFs expect to receive two innovative Data packets. However, if the two Interests are aggregated, only one Data packet will be delivered to the clients, thus reducing their expected innovative rate. The BF of a Data packet can be interpreted in a similar way. It contains a compact representation of the set of destination nodes where the Data packet can be potentially forwarded. When a network coded Data packet is generated at a node in order to consume an Interest, the content of the BF of the Interest is copied to the BF of the newly generated network coded Data packet. Thus, when the Data packet arrives at a node, it can only consume the Interest whose BF contains a subset of the client IDs stored in the BF of the Data packet. In addition, once this Data packet has been forwarded to some of the clients whose IDs where originally in its BF, these clients are removed from the list of potential consumers of this data so as to avoid sending the same Data packet to the same client multiple times. In [24], we provide an illustrative example that further clarifies how the addition of the BF resolves the ambiguity created by the user of network coding. The formal description of the use of Bloom filters in the processing of Interest and Data packets will be provided in Sections III-C and III-D, respectively.

Finally, since the Bloom filters are an essential element of our architecture and play an important role in the handling of the packets, they are also present in both the PIT and the CS. The modified tables are shown in Fig. 1. In the PIT, the list of incoming faces associated with some content name is replaced by a list of tuples $\langle face, Bloom\ filter \rangle$. Each tuple indicates the incoming face and the Bloom filter of the Interest packet which arrived on this face. For example, the first entry of the PIT table in Fig. 1(a) is an aggregation of two Interest packets for a network coded Data packet with content name */unibe.ch/videos/foreman.qcif/NC/class_0/gen_0*. The intersection of their Bloom filters is empty, which means that the Data packet that will consume the two Interests will be forwarded to two disjoint sets of clients. We can also see that the second entry contains an Interest packet for the same content. However, this Interest could not be aggregated with the first two, since some of the client IDs stored in its Bloom filter are also present in the Bloom filters of the first PIT entry. Similarly, for every Data packet that is stored in the CS, a tuple $\langle original,\ sent \rangle$ of Bloom filters is also stored in the CS along with the content name and the payload, as shown in Fig. 1(b). The *original* element of the tuple is an exact copy of the Bloom filter of the Interest packet that was consumed by this Data packet. The *original* Bloom filter is inserted in the Data packet when it is created by linearly combining matching Data packets stored in the CS with PRLNC, and then transmitted and stored along with the Data packet in the CS of the next hop node. The *sent* element of the tuple is the Bloom filter which stores the "sent" information, *i.e.*, the set of client IDs which the Data packet has already been forwarded to. The *sent* Bloom filter is only stored in the CS and is not transmitted with the Data packets. The addition of a second Bloom filter in each data entry of the CS is necessary, since the removal of elements from a Bloom filter is not permitted as it might cause false negatives [23]. Note that, even though the Data packets stored in the CS may have the same content name, they are linearly independent. Thus, even though the first two entries of the CS depicted in Fig. 1(b) have the same content name, the Data packets which correspond to these entries have linearly independent network coding coefficient vectors.

| **Content name** | **Requesting faces** $\langle face, Bloom filter \rangle$ |
|---|---|
| /unibe.ch/videos/foreman.qcif/NC/class_0/gen_0 | $\langle 0, 0010010 \dots 101 \rangle$<br>$\langle 2, 0101000 \dots 010 \rangle$ |
| /unibe.ch/videos/foreman.qcif/NC/class_0/gen_0 | $\langle 1, 0011110 \dots 110 \rangle$ |
| /unibe.ch/videos/foreman.qcif/NC/class_1/gen_0 | $\langle 1, 0010000 \dots 101 \rangle$<br>$\langle 1, 1001100 \dots 000 \rangle$<br>$\langle 2, 0100010 \dots 010 \rangle$ |
| ... | ... |

(a) Pending Interest Table (PIT)

| **Content name** | **Bloom filters** $\langle original, sent \rangle$ | **Payload** |
|---|---|---|
| /unibe.ch/videos/foreman.qcif/NC/class_0/gen_0 | $\langle 0010010 \dots 101, 0000010 \dots 001 \rangle$ | ... |
| /unibe.ch/videos/foreman.qcif/NC/class_0/gen_0 | $\langle 1011010 \dots 011, 1001000 \dots 010 \rangle$ | ... |
| /unibe.ch/videos/foreman.qcif/NC/class_1/gen_0 | $\langle 0010010 \dots 101, 0010010 \dots 101 \rangle$ | ... |
| /unibe.ch/videos/foreman.qcif/NC/class_0/gen_1 | $\langle 1100011 \dots 100, 1100011 \dots 100 \rangle$ | ... |
| ... | ... | ... |

(b) Content Store (CS)

Fig. 1. (a) Modified Pending Interest Table (PIT). The tuple $\langle face,\ Bloomfilter \rangle$ denotes the face on which the Interest arrived and the content of its BF. (b) Modified Content Store (CS). The tuple $\langle original,\ sent \rangle$ denotes the BF of the Data packet and the BF which stores the set of client IDs which the Data packet has already been forwarded to.

### C. Handling of Interest packets

The order in which the data structures are checked in our NC enabled NDN architecture differs from that of the standard NDN in order to guarantee that a sufficient number of innovative packets will be delivered to all clients. A PIT match is preferred over an FIB match and a FIB match is preferred over a CS match. The latter is due to the fact that some Interests require the linear combination of multiple Data packets in order to be consumed. Since these Data packets arrive at the node asynchronously, it may happen that another Interest with the same content name can arrive at the node

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



between the arrivals of the first and last Data packets required to generate the linear combination. If CS is checked first, this Interest packet will be consumed by the Data packets already stored in the CS. In that case, when the last Data packet arrives, there will not be sufficient innovative data in the CS to consume the pending Interest, which will remain in the PIT until it expires, reducing thus the flow of innovative Data packets. This case will be illustrated with an example in Section IV-C, once we have presented the forwarding strategy.

When an Interest arrives on some face, the PIT is checked for a pending Interest requesting the same content. Unlike the standard NDN architecture, where a positive outcome is observed if the content name of the incoming Interest matches the name of some pending Interest, this criterion is not sufficient for the PIT lookup procedure in the case of NC enabled NDN. This insufficiency stems from the fact that the content name in our framework no longer specifies a unique Data packet, but rather a collection of Data packets, *i.e.*, the set of network coded packets that belong to the same class and generation. In order to obtain all the network coded packets of a certain class and generation, a client will issue as many Interest packets with the same content name as the number of packets in the specified class. It is, therefore, highly likely that two Interests with the same content name may be requesting two linearly independent Data packets and cannot be aggregated in the PIT. In order to decide whether an Interest can be aggregated with another pending Interest, additional criteria are necessary for the PIT lookup procedure. These criteria are based upon the information that is stored in the BFs of the Interests.

Let $I$ denote the incoming Interest packet. Algorithm 1 summarizes the PIT lookup for a pending Interest upon the arrival of $I$. The algorithm takes as input the set $\mathcal{P}_I$ of pending Interests in the PIT whose content name matches the name of $I$, and outputs the Boolean variable *PendingInterestExists*. The latter indicates whether a matching pending Interest was found. If a matching pending Interest exists in the PIT, the algorithm outputs a pointer $I_x$ to the corresponding PIT entry. The PIT lookup is performed as follows. For every pending Interest packet $I'$ in $\mathcal{P}_I$, we first compute the union $\text{BF}_{I'}^{union}$ of the BFs that are stored in the list of requesting faces of $I'$. This union contains the set of clients whose request can be satisfied by the same Data packet. We then compare the set of client IDs stored in $\text{BF}_{I'}^{union}$ with the set $\mathcal{U}_I$ of the client IDs stored in the Bloom filter $\text{BF}_I$ of $I$. If none of the client IDs in $\mathcal{U}_I$ is present in $\text{BF}_{I'}^{union}$, $I'$ is considered as a matching pending Interest packet. This essentially means that $I$ and $I'$ can be consumed by the same Data packet. It is worth mentioning that the use of the BFs and the additional conditions on the PIT lookup procedure enable pipelining, which would not be otherwise feasible due to the fact that the same content name describes more than one network coded Data packet.

If a pending Interest packet is found in the PIT, the arrival face of $I$ as well as its Bloom filter $\text{BF}_I$ are stored in the list of requesting faces of the entry pointed by $I_x$ and no further action is taken. If no matching pending Interest exists, the FIB is checked for a matching entry. The FIB lookup procedure is identical to the one in the standard NDN architecture and is based only on the content name. If a matching FIB entry is found, the Interest packet is forwarded according to the forwarding strategy that will be described in Section IV-C, and then inserted as a new entry in the PIT.

**Algorithm 1** PIT lookup procedure upon arrival of the Interest packet $I$

**Input:** $I$, $\mathcal{P}_I$
**Output:** *PendingInterestExists*, $I_x$
**Initialization:** *PendingInterestExists* $\leftarrow$ **false**, $\mathcal{U}_I \leftarrow \{u \in \mathcal{U} \mid u \text{ in } \text{BF}_I\}$
**while** $\mathcal{P}_I \neq \emptyset$ **do**
  Select the first $I'$ in $\mathcal{P}_I$
  Compute $\text{BF}_{I'}^{union}$, *i.e.* the union of Bloom filters in the list of requesting faces associated with $I'$
  **if** $\text{BF}_{I'}^{union} \cap \text{BF}_u \neq \text{BF}_u$ for all $u \in \mathcal{U}_I$ **then**
    *PendingInterestExists* $\leftarrow$ **true**, $I_x \leftarrow I'$, $\mathcal{P}_I \leftarrow \emptyset$
  **else**
    $\mathcal{P}_I \leftarrow \mathcal{P}_I \backslash I'$
  **end if**
**end while**
**return** *PendingInterestExists*, $I_x$

**Algorithm 2** CS lookup procedure upon arrival of the Interest packet $I$

**Input:** $I$, $\mathcal{D}_I$
**Output:** *MatchingDataExists*, $\mathcal{D}'_I$
**Initialization:** $\mathcal{U}'_I \leftarrow \emptyset$, $\mathcal{D}'_I \leftarrow \emptyset$, *MatchingDataExists* $\leftarrow$ **false**, $\mathcal{U}_I \leftarrow \{u \in \mathcal{U} \mid u \in \text{BF}_I\}$
**while** $\mathcal{U}_I \neq \emptyset$ **and** $\mathcal{D}_I \neq \emptyset$ **do**
  Select the first $D$ in $\mathcal{D}_I$
  **for** every $u \in \mathcal{U}_I$ **do**
    **if** $u \in \text{BF}_D$ **and** $u \notin \text{BF}_D^{sent}$ **then**
      $\mathcal{D}'_I \leftarrow \mathcal{D}'_I \cup D$, $\mathcal{U}'_I \leftarrow \mathcal{U}'_I \cup u$
    **end if**
  **end for**
  $\mathcal{D}_I \leftarrow \mathcal{D}_I \backslash D$, $\mathcal{U}_I \leftarrow \mathcal{U}_I \backslash \mathcal{U}'_I$, $\mathcal{U}'_I \leftarrow \emptyset$
**end while**
**if** $\mathcal{U}_I == \emptyset$ **then**
  *MatchingDataExists* $\leftarrow$ **true**
**end if**
**return** *MatchingDataExists*, $\mathcal{D}'_I$

If both the PIT and the FIB lookup procedures fail to produce a positive outcome, the CS is searched for a Data packet that can potentially consume $I$. In order to determine whether the CS contains such a Data packet, we once again make use of the information stored in the BFs of $I$ and the Data packets in the CS. The CS lookup procedure upon the arrival of $I$ is summarized in Algorithm 2. The algorithm takes as input the set $\mathcal{D}_I$ of Data packets in the CS, whose content names match the content name of $I$. For every client $u$ whose ID is stored in the Bloom filter $\text{BF}_I$ of $I$, the algorithm searches the set $\mathcal{D}_I$ for a Data packet $D$, that has the same client ID stored in its Bloom filter $\text{BF}_D$, but not in the Bloom filter $\text{BF}_D^{sent}$, which keeps track of the clients to whom $D$ has been already forwarded. If a Data packet satisfying these criteria is found for all the clients in $\mathcal{U}_I$, then a Data packet that matches $I$ can be generated from the Data packets stored in the CS. The output Boolean variable *MatchingDataExists* indicates whether a Data packet can be generated to consume $I$. In case of a positive outcome, the algorithm also outputs a set $\mathcal{D}'_I$ of Data packets that must be combined with RLNC in order to consume $I$. This CS lookup procedure guarantees that a Data packet that has already been sent in response to an Interest packet originating from some client $u$, will not be retransmitted multiple times to $u$.

If the outcome of the Algorithm 2 is true, a network coded Data packet is generated by combining the Data packets in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



**Algorithm 3** CS update procedure after transmission of Data packet that consumes the Interest packet $I$

**Input:** $I$, $\mathcal{D}'_I$
**Initialization:** $\mathcal{U}_I \leftarrow \{u \in \mathcal{U} \mid u \in \mathrm{BF}_I\}$
**while** $\mathcal{U}_I \neq \emptyset$ **do**
  Pick $u \in \mathcal{U}_I$
  Find $D \in \mathcal{D}'_I$ s.t. $u \in \mathrm{BF}_D$ **and** $u \notin \mathrm{BF}_D^{sent}$
  Insert $u$ in $\mathrm{BF}_D^{sent}$
  $\mathcal{U}_I \leftarrow \mathcal{U}_I \backslash u$
**end while**

set $\mathcal{D}'_I$ with RLNC. The BF of the newly generated network coded Data packet is set equal to the BF of $I$. The network coded packet is then scheduled for transmission on the arrival face of $I$. The CS is updated according to the procedure described in Algorithm 3. In particular, the IDs of the clients that are stored in the Bloom filter $\mathrm{BF}_I$ of $I$, are inserted in the Data packets in the set $\mathcal{D}'_I$ that where identified as not yet transmitted to those clients. This update of the CS entries ensures that the network coded Data packets stored in CS will not be sent multiple times to the same client.

If after examining all three data structures, *i.e.*, PIT, FIB and CS, a match is not found, the incoming Interest packet is inserted as a new entry in the PIT and remains there until a matching Data packet arrives at the node. The absence of a match in all three data structures indicates that the node has already forwarded the necessary number of Interest packets in order to receive a sufficient amount of innovative content but not all the data has yet arrived at the node.

### D. Handling of Data packets

When a Data packet arrives, the timestamp of the video packet is first inspected. If according to the timestamp the packet has expired, the Data packet is discarded. If the incoming Data packet has not expired, a PIT lookup is performed for a matching pending Interest packet. A PIT match is determined based on not only the content name, as in the standard NDN architecture, but also on the information stored in the BFs of the incoming Data packet and of the pending Interests. The PIT lookup procedure upon arrival of an innovative Data packet $D$ is described in Algorithm 4. The algorithm takes as input the set $\mathcal{P}_D$ of pending Interests whose content name matches the name of $D$, and the set $\mathcal{D}_D$ of Data packets stored in the CS, whose name matches the name of $D$. For every pending Interest $I' \in \mathcal{P}_D$, Algorithm 2 is invoked in order to determine whether the data stored in the CS is sufficient to consume $I'$. If a set of necessary Data packets is found, Algorithm 4 outputs the pending Interest $I$ that can be consumed as well as the set $\mathcal{D}'_D$ of Data packets in CS that are required to consume $I$. A network coded Data packet is then generated by combining all the Data packets in $\mathcal{D}'_D$ with RLNC and is scheduled for transmission on all the faces in the list of requesting faces of the pending Interest $I$. $I$ is then removed from the PIT and the CS is updated using the procedure of Algorithm 3 with input $I$ and $\mathcal{D}'_D$. The above procedure is repeated until no matching pending Interest can be found.

**Algorithm 4** PIT lookup procedure upon arrival of an innovative Data packet $D$

**Input:** $\mathcal{P}_D$, $\mathcal{D}_D$
**Output:** *PendingInteretsExists*, $\mathcal{D}'_D$ and $I$
**Initialization:** *PendingInteretsExists* $\leftarrow$ **false**, $\mathcal{D}'_D \leftarrow \emptyset$
**while** $\mathcal{P}_D \neq \emptyset$ **do**
  Select the first $I'$ in $\mathcal{P}_D$
  Run Algorithm 2 with $I'$, $\mathcal{D}_D$ as input
  Let *MatchingDataExists*, $\mathcal{D}'_D$ be the output of step 6
  **if** *MatchingDataExists* == **true then**
    *PendingInteretsExists* $\leftarrow$ **true**, $I \leftarrow I'$, $\mathcal{P}_D \leftarrow \emptyset$
  **else**
    $\mathcal{P}_D \leftarrow \mathcal{P}_D \backslash I'$
  **end if**
**end while**
**return** *PendingInteretsExists*, $\mathcal{D}'_D$ and $I$

## IV. Optimized content-aware Interest forwarding strategy based on rate allocation

In this section, we focus on the problem of delivering the video to the clients at the optimal flow rates $\boldsymbol{f}$, as introduced in Section II-C, and describe the rate allocation based Interest forwarding strategy for the delivery of scalable video in the NC enabled NDN architecture.

### A. Content-aware rate allocation for Interest packets

We first present our algorithm for determining the rates of Interest packets for each class of network coded packets that must be transmitted from the clients and the intermediate nodes in order to achieve the optimal average quality of the video delivered to the clients. Our approach relies on the observation that Interests generated by different clients and expressing interest for a network coded packet of the same class $l$ and generation $g$ can be aggregated upon arriving at a node and consumed by the same network coded packet. Hence, only a single Interest has to be transmitted further towards the server to fetch the requested Data packet. Taking into account this property, we can make use of the concept of *conceptual flows* to design our content-aware rate allocation algorithm. Conceptual flows are network flows that can co-exist in the network without competing for the link bandwidth [25]. Thus, the overall flow of Interest packets from the clients to the servers can be regarded as a superposition of $U$ unicast conceptual flows from each client to the server, where $U = |\mathcal{U}|$ is the number of clients in the network. The rate of the actual flow on a link is the maximum of the rates of all the conceptual flows that pass through this link.

Let $r^{u,l}$ denote the conceptual flow of Interest packets expressing a request for network coded packets of class $l \in \mathcal{L}$ and originating from client $u \in \mathcal{U}$, and let $r_{ij}^{u,l}$ be the rate of the conceptual flow $r^{u,l}$ on link $(i,j)$, $\forall (i,j) \in \mathcal{E}_I$. The rate of the actual flow of Interests for class $l$ network coded packets on link $(i,j)$ will be $z_{ij}^l = \max_{u \in \mathcal{U}} r_{ij}^{u,l}$, since Interests originating from different clients can be consumed by the same network coded packet. Having assumed that every Interest transmitted on link $(i,j) \in \mathcal{E}_I$ is eventually consumed by a matching Data packet transmitted on link $(j,i) \in \mathcal{E}_D$, the rate of the flow of Data packets on link $(j,i)$ is equal to the rate of the flow of Interests $z_{ij}^l$ on link $(i,j)$. Thus, we have $z_{ij}^l = f_{ji}^l$, $\forall (i,j) \in \mathcal{E}_I$ and $(j,i) \in \mathcal{E}_D$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



Recall from Section II-C that the average video quality $\overline{Q}(\boldsymbol{f})$ is a piecewise constant function. Hence, there can be multiple sets of rate values that maximize the function in (1). To resolve this ambiguity and give higher priority to more important classes of packets, *i.e.*, classes with lower indices $l$, we introduce a cost function for each user $u$ that represents the overall cost of requesting packets of different classes:

$$C_u(\boldsymbol{r}^u) = \boldsymbol{c}^T \boldsymbol{r}^u = \sum_{l=0}^{L-1} c_l \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,l} \tag{3}$$

where $\boldsymbol{c} = (c_0, c_1, \ldots, c_{L-1})^T$ is the cost vector, $\boldsymbol{r}^u = (r^{u,0}, r^{u,1}, \ldots, r^{u,L-1})^T$ is the vector of conceptual flows of Interests associated with user $u$, and $r^{u,l} = \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,l}$, $\forall u \in \mathcal{U}$, due to the flow conservation property. The cost function $C_u(\boldsymbol{r}^u)$ associated with user $u$ is a linear function of the rates of conceptual flows of different classes requested by the user and, essentially, represents the total cost for the bandwidth consumed by the client $u$ for sending the Interest packets. $c_l$ is the bandwidth cost for requesting network coded packets of class $l$. By including the cost function in the problem formulation, we can cast our rate allocation problem as an optimization problem that seeks to maximize the average video quality while minimizing the average cost:

$$\arg\max_{\boldsymbol{f},\boldsymbol{r}^u} \frac{1}{U} \sum_{u\in\mathcal{U}} \Big( Q_u(\boldsymbol{f}^u) - C_u(\boldsymbol{r}^u) \Big) \tag{4}$$

The costs $c_0, c_1, \ldots, c_{L-1}$ can be chosen arbitrarily, but must satisfy the following constraints:

$$0 < c_0 < c_1 < \cdots < c_{L-1} \tag{5a}$$

$$c_l < \frac{q_l - q_{l-1}}{\sum_{k=0}^{l} R_k},\ l = 1, 2, , \ldots, L-1 \tag{5b}$$

Constraint (5a) implies that lower cost is assigned to more important packet classes, while higher cost is assigned to less important packet classes. This guarantees that more important classes, *i.e.*, classes that can be decoded by a larger number of users, are given higher priority in the rate allocation algorithm compared to the less important classes, *i.e.*, classes with higher index $l$ that can be decoded only by a limited number of users with sufficiently large bandwidth resources. Constraint (5b) ensures that a feasible rate allocation vector that leads to decoding video layers up to the $l-1$-th layer is not preferred over a feasible rate allocation vector that leads to decoding video layers up to the $l$-th layer because of the introduction of the cost function. To further explain the intuition behind this constraint, let us consider user $u$ and let $\boldsymbol{r}^u(l)$ be a feasible vectors of conceptual flows of Interests associated with user $u$, that leads to decoding up to layer $l$, *i.e.*, $Q_u(\boldsymbol{r}^u(l)) = q_l$. Similarly, let $\boldsymbol{r}^u(l-1)$ be a feasible vector of conceptual flows of Interests associated with user $u$ that leads to decoding up to layer $l-1$, *i.e.*, $Q_u(\boldsymbol{r}^u(l-1)) = q_{l-1}$. Without loss of generality, we can assume that $r^{u,k}(l) = 0$, $\forall k > l$, since layers higher than $l$ cannot be decoded with this rate allocation. In order to make sure that the vector $\boldsymbol{r}^u(l-1)$ is not preferred over the vector $\boldsymbol{r}^u(l)$, it must hold that

$$Q_u(\boldsymbol{r}^u(l)) - C_u(\boldsymbol{r}^u(l)) > Q_u(\boldsymbol{r}^u(l-1)) - C_u(\boldsymbol{r}^u(l-1) \tag{6}$$

For the inequality (6) to hold $\forall \boldsymbol{r}^u(l)$ and $\boldsymbol{r}^u(l-1)$, we must have that

$$\min Q_u(\boldsymbol{r}^u(l)) - C_u(\boldsymbol{r}^u(l)) > \max Q_u(\boldsymbol{r}^u(l-1)) - C_u(\boldsymbol{r}^u(l-1)) \tag{7}$$

from where after some manipulation it follows that

$$q_l - \max C_u(\boldsymbol{r}^u(l)) > q_{l-1} - \min C_u(\boldsymbol{r}^u(l-1)) \tag{8}$$

Since, by assumption, costs assigned to higher layers are greater than costs assigned to lower layers and $r^{u,k}(l) = 0$ $\forall k > l$, we have that

$$C_u(\boldsymbol{r}^u(l)) = \sum_{k=0}^{l} c_k r^{u,k}(l) < c_l \sum_{k=0}^{l} r^{u,k}(l) \le c_l \sum_{k=0}^{l} R_k$$

Thus, $\max C_u(\boldsymbol{r}^u(l)) = c_l \sum_{k=0}^{l} R_k$. It also holds that $\min C_u(\boldsymbol{r}^u(l-1)) = 0$. Combining these results with the inequality (8) leads immediately to the constraint in (5b).

Using the fact that $f_{ju}^l = z_{uj}^l$ and $z_{uj}^l = \max_{w\in\mathcal{U}} r_{uj}^{w,l} = r_{uj}^{u,l}$ (since $r_{wj}^{u,l} = 0,\ \forall\ w \neq u$), it follows that $f_{ju}^l = r_{uj}^{u,l}$, $\forall u \in \mathcal{U}$ and $\forall j : (j,u) \in \mathcal{E}_D$. We can thus re-express the video quality at user $u$ as a function of the rates of the conceptual flows of Interest packets as follows:

$$\begin{aligned} Q_u(\boldsymbol{f}^u) &= Q_u\Big( \sum_{j:(j,u)\in\mathcal{E}_D} f_{ju}^0, \ldots, \sum_{j:(j,u)\in\mathcal{E}_D} f_{ju}^{L-1} \Big) \\ &= Q_u\Big( \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,0}, \ldots, \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,L-1} \Big) = Q_u(\boldsymbol{r}^u) \end{aligned} \tag{9}$$

By combining (4), (3) and (9), the objective function in the optimization problem in (4) can be rewritten as:

$$\arg\max_{\boldsymbol{r}\in\mathcal{R}} \frac{1}{U} \sum_{u\in\mathcal{U}} \Big( Q_u(\boldsymbol{r}^u) - \boldsymbol{c}^T \boldsymbol{r}^u \Big) \tag{10}$$

where $\boldsymbol{r}$ is the vector of all rate variables $r_{ij}^{u,l}$, $\forall (i,j) \in \mathcal{E}_I,\ \forall u \in \mathcal{U},\ \forall l \in \mathcal{L}$. The polytope $\mathcal{R}$ is defined by the following set of linear equality and inequality constraints:

$$r_{ij}^{u,l} \ge 0, \quad \forall u \in \mathcal{U},\ \forall (i,j) \in \mathcal{E}_I,\ \forall l \in \mathcal{L} \tag{11a}$$

$$\sum_{n:(n,i)\in\mathcal{E}_I} r_{ni}^{u,l} = \sum_{j:(i,j)\in\mathcal{E}_I} r_{ij}^{u,l}, \quad \forall u \in \mathcal{U},\ \forall i \in \mathcal{N},\ \forall l \in \mathcal{L} \tag{11b}$$

$$\sum_{k=0}^{l} \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,k} \le \sum_{k=0}^{l} R_k, \quad \forall u \in \mathcal{U},\ \forall l \in \mathcal{L} \tag{11c}$$

$$x_{ij}^l \ge r_{ij}^{u,l}, \quad \forall u \in \mathcal{U},\ \forall (i,j) \in \mathcal{E}_I,\ \forall l \in \mathcal{L} \tag{11d}$$

$$\sum_{l\in\mathcal{L}} x_{ij}^l (p_I + p_D) \le B_{ij} \quad \forall (i,j) \in \mathcal{E} \tag{11e}$$

where $p_I$ and $p_D$ in (11e) are the size (in bits/packet) of the Interest and Data packets, respectively. The objective function in (10) is a function of only the rates of the conceptual flows of Interest packets on the input links of the clients. Constraints (11a) – (11e) define the set of feasible rate allocations for the conceptual flows of Interest packets. Constraint (11a) guarantees that every conceptual flow is non-negative. Constraint (11b) is the flow conservation constraint that holds for every conceptual flow, since every conceptual flow is a unicast flow from the client to the server. This constraint guarantees that the number of Interests that enter an intermediate node is equal to the number of Interests that are transmitted from the node for every conceptual flow and for every class of network coded

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



packets. Constraint (11c) is the innovative rate constraint, which states that the rate of Interests generated by a client for each class of network coded packets should not exceed the maximum rate of innovative Data packets that the source can provide. Together (11b) and (11c) ensure that the clients do not request Data packets at a rate which is higher than the rate at which innovative Data packets are generated by the server. The variable $x_{ij}^l$ in (11d) is an auxiliary variable that represents the actual transmission rate of Interests for network coded packets of class $l$ on link $(i,j)$. It upper bounds the rates of the conceptual flows of Interests on every link. The actual rates $x_{ij}^l$ are further bounded in (11e) by the available bandwidth on the link. Constraint (11e) states that the bandwidth required to transmit the Interests and the Data packets that consume these Interests on a link, should not exceed the overall bandwidth available for that link.

### *B. Subgradient-based optimization algorithm*

In this section, we present an efficient algorithm based on Lagrangian relaxation and the subgradient method for solving the optimization problem formulated in (10).

We first relax the coupling constraints in (11d) and obtain the Lagrangian dual function. This choice is driven by the fact that the resulting Lagrangian dual problem can be decomposed into several optimization subproblems that can be solved independently. The Lagrangian dual function is given by

$$\begin{aligned} L(\boldsymbol{\mu}) = \max_{\mathcal{R}'} \frac{1}{U} \sum_{u\in\mathcal{U}} \Big( Q_u(\boldsymbol{r}^u) - \boldsymbol{c}^T \boldsymbol{r}^u \Big) - \\ - \sum_{u\in\mathcal{U}} \sum_{(i,j)\in\mathcal{E}_I} \sum_{l\in\mathcal{L}} \mu_{ij}^{u,l} (r_{ij}^{u,l} - x_{ij}^l) \end{aligned} \tag{12}$$

where $\mathcal{R}'$ is the polytope defined by the linear constraints (11a), (11b), (11c) and (11e). The Lagrangian dual of the primal problem in (10) is then

$$\min_{\boldsymbol{\mu}\geq 0} L(\boldsymbol{\mu}) \tag{13}$$

where $\boldsymbol{\mu}$ is the vector of Lagrangian multipliers $\mu_{ij}^{u,l}$. We can observe that the Lagrangian subproblem in (12) can be decomposed into several optimization subproblems that can be solved independently. These optimization subproblems consist of $U$ maximization problems, one for every user $u \in \mathcal{U}$

$$\arg\max_{\boldsymbol{r}\in\mathcal{R}_u} \frac{1}{U} \Big( Q_u(\boldsymbol{r}^u) - \boldsymbol{c}^T \boldsymbol{r}^u \Big) - \sum_{(i,j)\in\mathcal{E}_I} \sum_{l\in\mathcal{L}} \mu_{ij}^{u,l} r_{i,j}^{u,l} \tag{14}$$

where the polytope $\mathcal{R}_u$ is defined for every user $u$ by the constraints (11a), (11b) and (11c), and $|\mathcal{E}_I|$ maximization problems

$$\begin{aligned} \arg\max_{\boldsymbol{x}} \sum_{l\in\mathcal{L}} \Big( \sum_{u\in\mathcal{U}} \mu_{ij}^{u,l} \Big) x_{ij}^l, \quad \forall (i,j) \in \mathcal{E}_I \\ \text{s.t.} \quad \sum_{l\in\mathcal{L}} x_{ij}^l (p_I + p_D) \leq B_{ij} \end{aligned} \tag{15}$$

where $\boldsymbol{x}$ is the vector of the rates $x_{ij}^l, \forall (i,j) \in \mathcal{E}_I$ and $\forall l \in \mathcal{L}$, of the actual flows of Interest packets. The maximization problem in (15) is a continuous knapsack problem that can be solved locally with a greedy algorithm [26]. The optimization problem in (14) consists in maximizing a piecewise linear objective function. Since it is hard to directly optimize the objective function in (14), the optimization problem in (14) can be further transformed into a two level optimization problem:

$$\max_{k\in\mathcal{L}} \max_{\mathcal{R}_u \cap \mathcal{R}_k} \frac{1}{U} \Big( q_k - \boldsymbol{c}^T \boldsymbol{r}^u \Big) - \sum_{(i,j)\in\mathcal{E}_I} \sum_{l\in\mathcal{L}} \mu_{ij}^{u,l} r_{i,j}^{u,l}, \quad \forall u \in \mathcal{U} \tag{16}$$

The polytope $\mathcal{R}_k$ defines the rate region where the class $k$ packets are decodable and is given by the following set of linear constraints

$$\sum_{m=0}^{l} \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,m} \leq \sum_{m=0}^{l} R_m, \ \forall l \in \{0,\ldots,k-1\} \tag{17a}$$

$$\sum_{m=0}^{l} \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,m} = \sum_{m=0}^{l} R_m, \ l = k \tag{17b}$$

$$\sum_{m=0}^{l} \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,m} < \sum_{m=0}^{l} R_m, \ \forall l \in \{k+1,\ldots,L-1\} \tag{17c}$$

At the lower level of the optimization problem in (16), the objective function is linear and is maximized over the rate region where the $k$-th class of network coded packets is decodable. Constraint (17b) guarantees that the rate of the Data packets that will be delivered to the client is sufficient to decode network coded packets of class $k$, while (17a) ensures that the rate of Data packets per each class does not exceed the available innovative rate for this class of packets. Finally, constraint (17c) ensures that the rate of Data packets delivered to the client is not sufficient to decode network coded packets of classes higher than $k$. Constraints (17a)-(17c) define the rate values that permit client $u$ to decode class $k$ packets but not packets of higher classes. The lower level optimization problem in (16) is a minimum-cost multi-commodity flow problem and can be solved distributively in polynomial time [27]. At the higher level of the optimization problem in (16), a point-wise maximization with respect to the packet classes is performed by selecting the solution that yields the best value of the objective function.

To solve the Lagrangian dual problem in (13), we apply the subgradient algorithm. Specifically, we select an initial set of non-negative values of Lagrangian multipliers $\mu_{ij}^{u,l}[0]$, $\forall (i,j) \in \mathcal{E}_I$, $\forall u \in \mathcal{U}$, $\forall l \in \mathcal{L}$, and update them at every iteration $t = 1, 2, \ldots$ according to the following rule:

$$\mu_{ij}^{u,l}[t] = \max\{0, \mu_{ij}^{u,l}[t-1] + \theta[t](r_{ij}^{u,l}[t] - x_{ij}^{u,l}[t])\} \tag{18}$$

where $r_{ij}^{u,l}[t]$ and $x_{ij}^{u,l}[t]$ are the solutions to the optimization problems in (14) and (15) at the $t$-th iteration of the subgradient algorithm given the current value of the Lagrangian multipliers $\mu_{ij}^{u,l}[t]$. The step size $\theta[t]$ controls the convergence properties of the subgradient algorithm. When the step size is chosen such that

$$\theta[t] > 0, \quad \lim_{t\to\infty} \theta[t] = 0, \quad \sum_{t=1}^{\infty} \theta[t] = \infty \tag{19}$$

the subgradient algorithm is guaranteed to converge to the optimal solution of the Lagrangian dual problem in (13) [28]. Here we choose $\theta[t] = a/(b+ct)$, $\forall t$, with $a > 0$, $b \geq 0$ and $c > 0$, which satisfies the conditions in (19). In order to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



**Forwarding Information Base (FIB)**

| **Content name** | **Face list** $\langle face, counter \rangle$ |
|---|---|
| /unibe.ch/videos/foreman.qcif/NC/class_0/gen_2 | $\langle 0, c^0_{0,2} \rangle$ $\langle 1, c^1_{0,2} \rangle$ $\langle 2, c^2_{0,2} \rangle$ |
| /unibe.ch/videos/foreman.qcif/NC/class_1/gen_3 | $\langle 0, c^0_{1,3} \rangle$ $\langle 2, c^2_{1,3} \rangle$ |
| ... | ... |

Fig. 2. Forwarding Information Base (FIB) in our NC enabled NDN architecture. The tuple $\langle f, c^f_{l,g} \rangle$ denotes the outgoing face $f$ and the counter $c^f_{l,g}$ associated with this face. The counter $c^f_{l,g}$ indicates the number of Interest packets for network coded Data packets of class $l$ and generation $g$ to be forwarded on face $f$.

recover the optimal solution to the primal problem in (10), we use the method introduced by Sherali *et al.* [28]. Specifically, at every iteration $t$, the value of the primal variables $r^{u,l}_{ij}$ and $x^l_{ij}$ is constructed according to the following update rule:

$$\hat{r}^{u,l}_{ij}[t] = \sum_{h=1}^{t} \nu_h[t] r^{u,l}_{ij}[h], \quad \hat{x}^l_{ij}[t] = \sum_{h=1}^{t} \nu_h[t] x^l_{ij}[h] \qquad (20)$$

where $\sum_{h=1}^{t} \nu_h[t] = 1$ and $\nu_h[t] \geq 0$, for $h = 0, 1, \ldots, t$. A valid choice for the convex combination weights $\nu_h[t]$ is to set $\nu_h[t] = 1/t$, $\forall h = 0, 1, \ldots, t$, $\forall t$. The accumulation point of the sequence of primal values $\{\hat{r}^{u,l}_{ij}[t]\}$ and $\{\hat{x}^l_{ij}[t]\}$ generated via (20) is then a feasible and optimal solution to the optimization problem in (10) [28].

Though the rate allocation solution found by means of the procedure described above is optimal, it is not guaranteed to be an integer solution. This is due to the fact that the problem in (10) is a variant of a multi-commodity flow problem, which is not guaranteed to admit an integer solution even if the links' bandwidth takes integer values. As we will discuss in Sec. IV-C, our Interest forwarding strategy requires integer values of the rates of the Interest packets. Therefore, in cases where the optimal rate allocation solution found by the subgradient algorithm is fractional, we approximate it by the integer rate allocation vector that has the smallest Euclidean distance from the optimal solution. Though this solution is not guaranteed to be optimal, it is expected to have a similar structure to the optimal solution, *i.e.* to allocate the available bandwidth according to the priority of the video layers giving higher priority to lower classes. Note that the integer version of the multi-commodity problem is NP-complete.

### C. Interest forwarding strategy

We now present the Interest forwarding strategy, which in combination with the optimal rate allocation ensures the optimal delivery of the video to the clients. The forwarding strategy controls the forwarding of Interests, when no matching pending Interest exists in the PIT. It performs the tasks of selecting the outgoing face and of modifying appropriately the Bloom filter of the forwarded Interest in order to achieve the packet rate dictated by the rate allocation algorithm.

**Algorithm 5** Construction of the Bloom filter for an Interest packet that is being forwarded

1: **Input:** $c^f_{l,g}$, $z^l_{ij}$, $r^{u,l}_{ij}$, $\forall u \in \mathcal{U}$
2: **Output** $\mathrm{BF}_I$
3: **Initialization:** $\mathrm{BF}_I \leftarrow \emptyset$, $p \leftarrow z^l_{ij} - c^f_{l,g} + 1$
4: **for** every $u \in \mathcal{U}$ **do**
5: **if** $r^{u,l}_{ij} \neq 0$ **then**
6: $t \leftarrow \left\lfloor \frac{z^l_{ij}}{r^{u,l}_{ij}} \right\rfloor$
7: **if** $p \mod t == 0$ **and** $p/t \leq r^{u,l}_{ij}$ **then**
8: Insert $u$ in $\mathrm{BF}_I$
9: **end if**
10: **end if**
11: **end for**
12: **return** $\mathrm{BF}_I$

The face selection mechanism is implemented in the Forwarding Information Base. The modified FIB table is illustrated in Fig. 2. Each FIB entry consists of a content name, and a list of $\langle f, c^f_{l,g} \rangle$ tuples, where $f$ is the outgoing face and $c^f_{l,g}$ is a counter associated with the outgoing face $f$ and the content name referring to generation $g$ and packet class $l$. At the beginning of the streaming session, the FIB table is initialized with the entries that refer to all packet classes of the first $G$ generations. As the streaming session progresses, the entries that are related to the generations that have expired are deleted from the FIB, while new entries for more recent generations are inserted. This permits to keep the number of FIB entries low and at the same time to store all the information that is needed in order to manage requests within a given time window. For each new entry inserted in the FIB, the counters $c^f_{l,g}$ associated with the outgoing faces are initialized using the values obtained from the rate allocation algorithm. Specifically, we set $c^f_{l,g} = z^l_{ij}$ for every generation $g$, where $z^l_{ij}$ is the rate of the actual flow of Interest packets for network coded Data packets of class $l$ on the link $(i, j)$ and $f$ is the face associated with this link. When a FIB lookup is performed, the content name of the Interest packet is compared against the content names in the entries of the FIB. If a FIB entry with a matching content name is found, the Interest packet is forwarded on all the faces in the list for which the counter is non zero. Once the Interest packet is scheduled for transmission, the corresponding counter is reduced by 1. When all the counters in the list associated with some entry become zero, this entry is removed from the FIB and the node does not forward any more Interest packets with this content name.

The second task of the forwarding strategy is to insert the appropriate client IDs in the Bloom filter of the Interest packet that is being forwarded, in order to ensure that the Interest packets will be consumed by innovative data. The selection procedure is described in Algorithm 5. The variables $r^{u,l}_{ij}$ are the rates of the conceptual flows of Interest packets for network coded packets of class $l$ on link $(i, j)$. The algorithm first computes the counter $p$, which indicates the number of Interest packets of class $l$ and generation $g$ that will have been transmitted on face $f$ including the Interest packet that is being currently forwarded. Then, for every client, the algorithm examines whether this client's ID should be inserted in the Bloom filter of the Interest packet. Specifically, the ID of client $u$ must be inserted in the Bloom filter of every $t$-th forwarded packet, where $t$ is computed in step 6. In [24],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



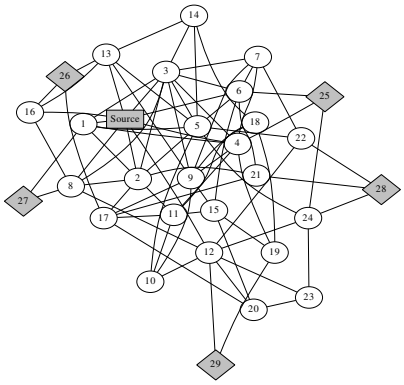


Fig. 3. PlanetLab topology consisting of one source node (node 0), 24 intermediate nodes and 5 client nodes (nodes 25 - 29.)

we provide a numerical example which further illustrates the implementation of the forwarding strategy and the reason for checking the FIB prior to checking the CS.

## V. Performance evaluation

### A. Evaluation scenario

We consider that the server encodes the Foreman video sequence in CIF format into three quality layers with the SVC extension of the H.264 video compression standard [2]. The GOP size is set to 30 frames and the frame rate is set to 30 fps. Each GOP is packetized into 73 packets, which consist of $a_0 = 38$ base layer packets, $a_1 = 15$ first enhancement layer packets and $a_2 = 20$ second enhancement layer packets. The size of the Data packets is $p_D = 1600$ bytes and the size of the Interests is $p_I = 200$ bytes. The generation size is set equal to the size of the GOP, *i.e.*, 73 packets. Thus, each generation corresponds to one second of video. The duration of the entire video sequence is 40 seconds, *i.e.*, there are 40 generations in total. The video quality achieved after decoding each of the three video layers is $q_0 = 36.48$ dB, $q_1 = 37.82$ dB and $q_2 = 39.09$ dB, respectively. The values of the cost coefficients are $c_0 = 0.01$, $c_1 = 0.02$ and $c_2 = 0.025$, and are chosen according to (5a) and (5b). The size of the Galois field, where the network coding operations are performed, is chosen equal to $2^8$. This field size constitutes a common design choice for practical applications using RLNC, as it achieves a good trade-off between the probability of generating non-innovative packets due to the random selection of the network coding coefficients, and the data overhead due to the inclusion of the network coding coefficients in the header [16]. Our proposed architecture is evaluated on the network topology depicted in Fig. 3. This topology was constructed based on real data collected by the PlanetLab project [29]. The topology consists of one source node, 5 client nodes (nodes 25-29) and 24 helpers. To generate this topology we followed the procedure described in [30].

### B. Convergence of the rate allocation algorithm

We first present the convergence of our rate allocation algorithm. Fig. 4 depicts the evolution of the cumulative rate of Interest packets sent by node 29 with the number of iterations of the rate allocation algorithm for different values of the link bandwidth. Each figure shows the convergence of the rate of Interest packets requesting network coded packets that belong to the three available packet classes. When the link bandwidth is low (leftmost figure), the client requests only network coded packets from class 0 which consists of packets that belong to the base layer. The rate of Interest packets requesting class 0 network coded packets is 38 packets/sec, which is the encoding rate of the base layer. As the link bandwidth increases (middle figure), the client starts to request also class 1 network coded packets, which results in decoding the first enhancement layer. The rate of the Interest packets requesting class 1 packets converges to the value of the encoding rate of the first enhancement layer, which is equal to 15 packets/sec. Finally, for sufficiently large values of bandwidth (rightmost figure), the client requests packets from all three classes, which permits to decode the second enhancement layer. It is worth noting that among several possible optimal solutions, our content-aware rate allocation algorithm favors the solution that allocates the rate according to the importance of the delivered packets. In particular, in cases where apart from the base layer additional enhancement layers can be decoded, each class of packets is requested at its encoding rate which minimizes the cost introduced in (3) and reflects the prioritization of the packet classes according to their importance.

### C. Video quality evaluation

In order to validate our architecture, we have implemented a customized simulator in NS-3 [31] with the main components of the NDN architecture as well as the additional features described in Section III, that enable the delivery of network coded scalable content. Each point in the figures is the average of 100 simulations. We assume that the clients join the streaming session with a random delay, which is much smaller than the playback delay. This creates some randomness in the order in which Interests arrive at intermediate nodes. Another source of randomness is the random selection of coding coefficients during the network coding operations. We consider that the CS is sufficiently large, so that it can accommodate all incoming Data packets without deleting those that are already stored in the CS. Data packets whose decoding deadline expires are removed from the CS. Thus, in practice the CS only needs to cache Data packets from a few generations.

We evaluate the proposed architecture for the delivery of scalable video in terms of quality under the scenario described in Section V-A. In Fig. 5 we present the evolution of the PSNR of the delivered video with respect to the links' bandwidth for each network client. UB denotes the upper bound on the video quality that can be attained at the client. The upper bound is calculated based on the maximum achievable flow between each client and the source and represents the video quality that the client would receive if it was using all the network resources. EXP stands for the expected value of the PSNR based on the rate allocation solution obtained with our content-aware rate allocation algorithm. SIM denotes the PSNR values obtained from the simulation of our NC enabled NDN

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



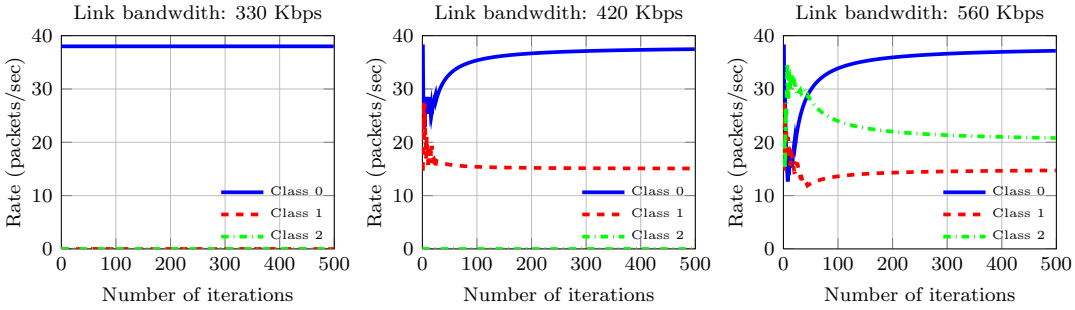


Fig. 4. Cumulative rates of Interest packets sent by node 29 versus the number of iterations of the rate allocation algorithm.

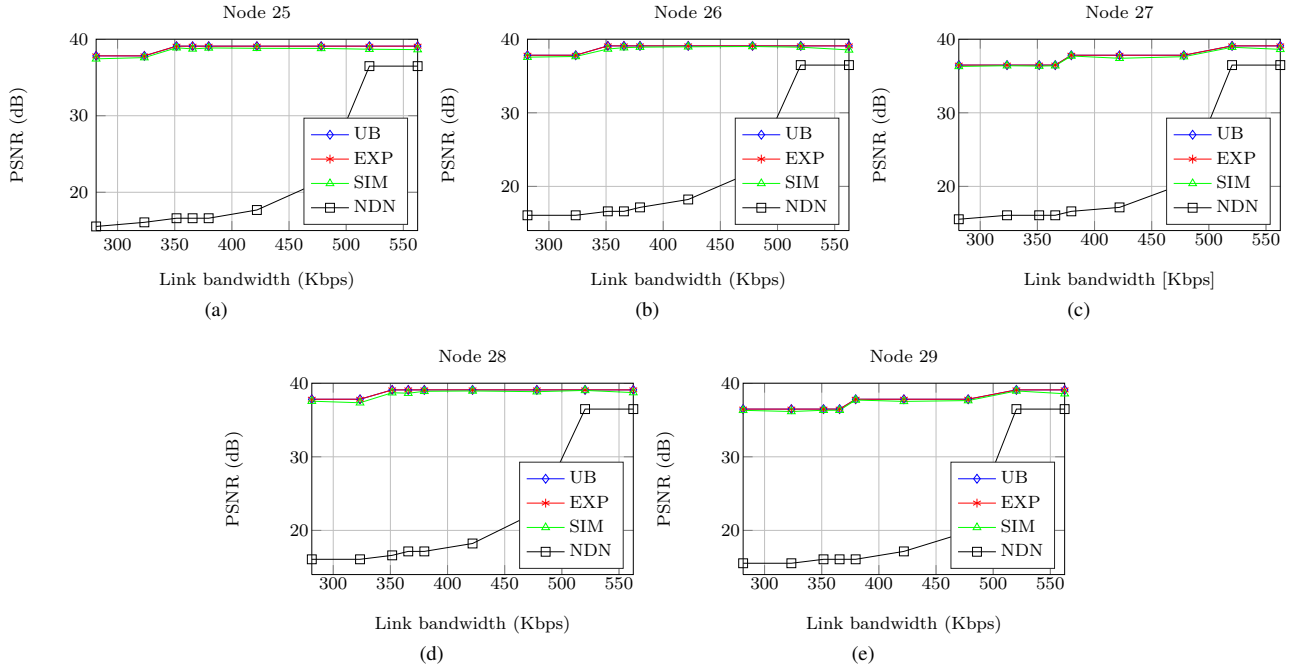


Fig. 5. Average PSNR of the decoded video as a function of the links' bandwidth.

architecture equipped with the proposed forwarding strategy. Finally, NDN denotes the best route strategy of the NDN forwarding deamon [32]. We can see that the performance of our proposed forwarding strategy is very close to the expected one with a deviation that does not exceed 0.5 dB. Moreover, we note that as the links' bandwidth increases, the clients are able to decode more video layers and improve the quality of the delivered video. This improvement is attributed to the use of PRLNC that allows to optimally use the available resources. Our forwarding strategy exploits the optimal rate allocation in order to decide which Interests to forward on which faces. This ensures that there are no packet replicas and the rate of redundant packets is minimized. The small degradation in the quality of the delivered video compared to the expected performance is caused by the randomness in the selection of the network coding coefficients which may result in the generation of linearly dependent network coded packets. Since the decoding of the $l$-th class of packets requires that all the lower classes can be decoded, the delivery of even a single non-innovative packet of class $k$ causes a failure in decoding all classes higher or equal to $k$. This problem can be addressed by reissuing an Interest for an additional Data packet whenever a non-innovative packet arrives at a node. Finally, we can see that the NDN best route strategy is not competitive with our scheme, as clients can only decode the first few generations of the base video layer for low and medium bandwidth values. The NDN scheme manages to decode reliably the base layer only in high bandwidth regime. This is attributed to the fact that the NDN best route strategy is not optimized for video, *i.e.*, it does not take into account the decoding deadlines and the quality of the video data.

Fig. 6 depicts the average PSNR of the delivered video at nodes 26 and 27 as a function of time. The playback delay is set to 1000msec. We can see that our NC enabled NDN architecture achieves low jitter in the video quality. The quality of the decoded video remains close to the value that is expected based on the optimal rate allocation. This indicates that the majority of generations are decoded correctly with only a few of them decoded at lower quality or not decoded at all. This is true even when the bandwidth resources get scarce. It is worth noting that the employed forwarding strategy ensures that the probability of receiving a redundant packet is zero and non-innovative packets can be received only due to the randomness of the network coding operations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2767778, IEEE Transactions on Multimedia



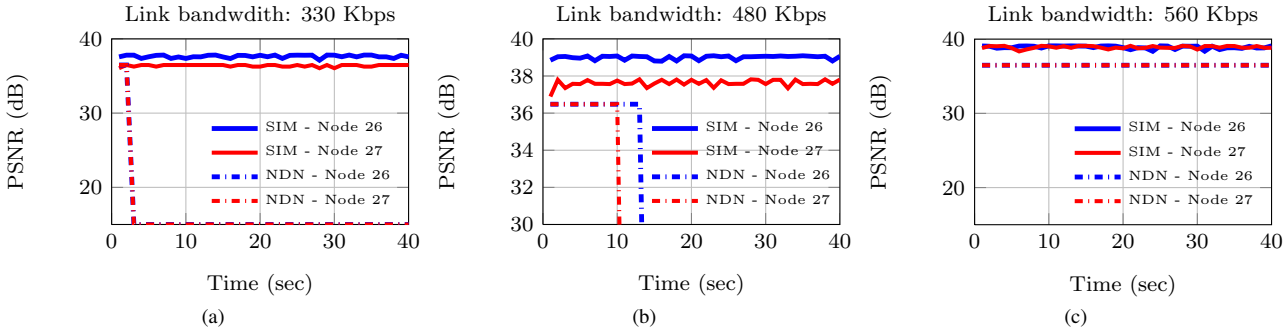


Fig. 6. Average PSNR of the decoded video versus time.

Finally, as we can see, when the bandwidth resources are scarce, with the NDN forwarding strategy clients are only able to decode the first generations of the base layer. As the video streaming progresses, the NDN best route forwarding strategy fails to ensure that the video packets are delivered within the time constraints.

## VI. Conclusions

We have presented a novel network coding enabled NDN architecture for the delivery of scalable video. In order to design the content-aware forwarding strategy, we have formulated a rate allocation problem which decides on the optimal rates of Interest packets sent by clients and intermediate nodes. This rate allocation ensures that the achieved flow of Data packets maximizes the average quality of the video delivered to the client population. We have also proposed the necessary modifications to the standard NDN architecture in order to support the handling of Interest and Data packets when intermediate nodes perform network coding. In order to resolve the practical issues that are caused by the use of network coding, we have proposed the use of Bloom filters that compactly store additional information about the Interest and Data packets. We have evaluated our architecture in terms of the achieved video quality for the delivery of scalable video. The results indicate that the proposed architecture achieves a close to optimal performance. Though the proposed scheme is evaluated for H.264/SVC coded video, our scheme is general and can be used with any type of scalable data like scalable HEVC data and multiview scalable data.

Our future work will include the extension of the proposed architecture to wireless scalable video delivery. Wireless video transmission suffers from high packet losses due to interference and varying channel conditions, which may affect both the Interest and the Data packets. Network coding can partly deal with this challenge by eliminating the need for repeating requests or transmissions of specific Data objects, since all coded Data packets are equivalent in terms of contained information. However, an appropriate mechanism to properly handle the re-transmissions of Interest and/or Data packets would still be needed to compensate for the lost packets and ensure the delivery of the necessary for decoding number of coded Data objects. Another challenge associated with wireless video delivery is user mobility, which renders the communication network highly dynamic. Anastasiades *et al.* [21] have shown that the use of Raptor codes for data delivery in mobile ICN networks reduces significantly the content retrieval time and the number of Data packet transmissions. These improvements are critical for video delivery which is subject to strict time constraints and is typically associated with large volumes of content. Due to user mobility in wireless networks, the optimization of the rate allocation strategy must be performed every time the network topology changes which may result in high computational overhead. Thus, we plan to investigate the design of heuristic algorithms for the forwarding of Interest packets that would adapt quickly to the varying network configuration and at the same time have close to optimal performance. Finally, apart from the challenges, users can take advantage of the overheard transmission intended to other users to limit the data completion delay. This has been showcased by Seferoglu *et al.* [33] who proposed a practical system called MicroCast, that exploits network coding for device-to-device enabled video multicasting. In the context of ICN, the extension of our method to wireless scalable video transmission would require using prioritized network codes and taking into account Interest and Data packet overhearing in the design of the Interest forwarding strategy.

## References

[1] "Cisco Visual Networking Index-Forecast and Methodology, 2014–2019," White paper, May 2015.
[2] ITU-T and ISO/IEC JTC 1, "Advanced Video Coding for Generic Audiovisual Sevices, Amendment 3: Scalable Video Coding," *Draft ITU-T Recommnedation H.264 - ISO/IEC 14496-10 (AVC)*, Apr. 2005.
[3] A. Ford *et al.*, "TCP extensions for multipath operation with multiple addresses," RFC 6824, January 2013.
[4] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.
[5] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles," in *Proc. of the 2nd Annual ACM Conf. on Multimedia Systems*, San Jose, CA, USA, Feb. 2011, pp. 133–144.
[6] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of Information-Centric Networking Research," *IEEE Communication Surveys and Tutorials*, vol. 16, no. 2, pp. 1024–1048, Apr. - June 2014.
[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proc. of ACM CoNEXT*, Rome, Italy, Dec. 2009, pp. 1–12.
[8] C. Tsilopoulos, G. Xylomenos, and G. C. Polyzos, "Are Information-Centric Networks Video-Ready?" in *Proc. of IEEE Packet Video Workshop, PV'13*, San Jose, CA, USA, Dec. 2013, pp. 1–8.
[9] D. Byun, B.-J. Lee, and M.-W. Jang, "Adaptive Flow Control via Interest Aggregation in CCN," in *Proc. of Int. Conf. on Communications, ICC'13*, Budapest, Hungary, June 2013.
[10] C. Tsilopoulos and G. Xylomenos, "Supporting Diverse Traffic Types in Information Centric Networks," in *Proc. of ACM SIGCOMM ICN Workshop*, Toronto, ON, Canada, Aug. 2011, pp. 13–18.

[11] Y. Liu, J. Geurts, J. Point, S. Lederer, B. Rainer, C. Mueller, C. Timmerer, and H. Hellwagner, "Dynamic Adaptive Streaming over CCN: A Caching and Overhead Analysis," in *Proc. of IEEE Int. Conf. on Communications, ICC'2013*, Budapest, Hungary, June 2013.

[12] 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 *Proc. of ACM Workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms and Applications,*, Hilton Head Island, SC, USA, June 2012.

[13] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Trans. Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[14] N. Thomos, J. Chakareski, and P. Frossard, "Prioritized Distributed Video Delivery With Randomized Network Coding," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 776–787, Aug. 2011.

[15] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On Randomized Network Coding," in *Proc. of the 41st Allerton Conf. on Communication, Control and Computing*, Monticello, IL, USA, Oct. 2003.

[16] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *Proc. of the 41st Allerton Conf. on Communication, Control and Computing*, Monticello, IL, USA, Oct. 2003.

[17] Q. Wu, Z. Li, and G. Xie, "CodingCache: Multipath-aware CCN Cache with Network Coding," in *Proc. of ACM SIGCOMM ICN Workshop*, Hong Kong, China, Aug. 2013, pp. 41–42.

[18] J. Llorca, A. Tulino, K. Guan, and D. Kilper, "Network-Coded Caching-Aided Multicast for Efficient Content Delivery," in *Proc. of IEEE Int. Conf. on Communications, ICC'2013*, Budapest, Hungary, June 2013, pp. 3557–3562.

[19] K. Matsuzono, H. Asaeda, and T. Turletti, "Low latency low loss streaming using in-network coding and caching," in *Proc. of IEEE INFOCOM'17*, Atlanta, USA, May 2017.

[20] J. Saltarin, E. Bourtsoulatze, N. Thomos, and T. Braun, "NetCodCCN: a Network Coding Approach for Content-Centric Networks," in *Proc. of IEEE INFOCOM*, San Francisco, CA, USA, April 2016.

[21] C. Anastasiades, N. Thomos, A. Striffeler, and T. Braun, "RC-NDN: Raptor codes enabled named data networking," in *IEEE Int. Conf. on Communications (ICC '15)*, 2015, pp. 3026–3032.

[22] E. Kurdoglu, N. Thomos, and P. Frossard, "Scalable Video Dissemination with Prioritized Network Coding," in *Proc. of Streaming and Media Communication Workshop, StreamComm'11 (in conjunction with ICME'11)*, Barcelona, Spain, July 2011.

[23] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[24] E. Bourtsoulatze, N. Thomos, J. Saltarin, and T. Braun, "Content-Aware Delivery of Scalable Video in Network Coding Enabled Named Data Networks, *arXiv:1510.06659v2[CS.NI]*, Oct. 2017.

[25] Z. Li, B. Li, D. Jiang, and L. C. Lau, "On Achieving Optimal Throughput With Network Coding," in *Proc. of 24th INFOCOM*, Miami, FL, USA, Mar. 2005, pp. 2184–2194.

[26] G. B. Dantzig, "Discrete-variable extremum problems," *Operations Research*, vol. 5, no. 2, pp. 266–277, 1957.

[27] N. Chatzipanagiotis and M. M. Zavlanos, "Distributed Stochastic Multicommodity Flow Optimization," in *Proc. of IEEE Global Conf. on Signal and Information Processing*, Dec. 2013, pp. 883–886.

[28] H. D. Sherali and G. Choi, "Recovery of Primal Solutions When Using Subgradient Optimization Methods to Solve Lagrangian Duals of Linear Programs," *Elsevier Operations Research Letters*, vol. 19, no. 3, pp. 105–113, Sep. 1996.

[29] "PlanetLab," https://www.planet-lab.org/.

[30] N. Cleju, N. Thomos, and P. Frossard, "Selection of Network Coding Nodes for Minimal Playback Delay in Streaming Overlays," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1103–1115, Oct. 2011.

[31] "The network simulator - ns3," http://www.nsnam.org/.

[32] A. Afanasyev *et al.*, "NFD developer's guide," NDN, Technical Report NDN-0021 Revision 7, Oct. 2016.

[33] A. Le, L. Keller, H. Seferoglu, B. Cici, C. Fragouli, and A. Markopoulou, "MicroCast: Cooperative video streaming using cellular and local connections," *IEEE/ACM Trans. Networking*, vol. 24, no. 5, pp. 2983–2999, Oct. 2016.