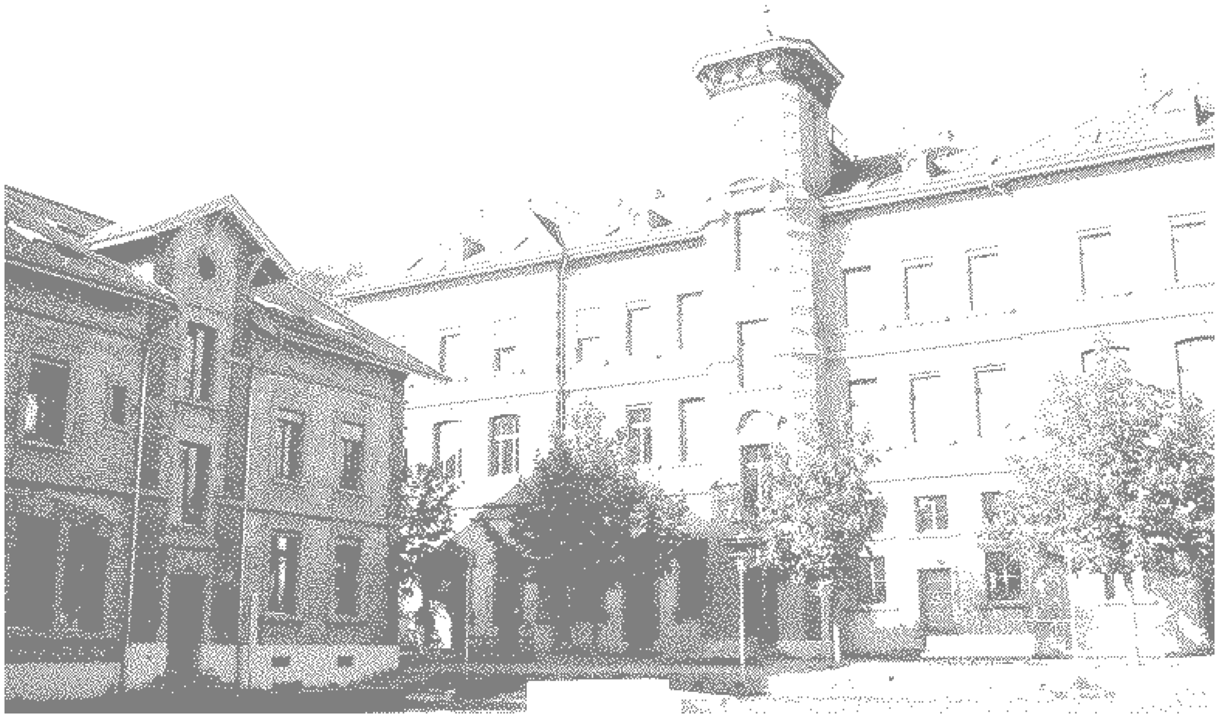
b
UNIVERSITÄT
BERN

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

E. Bourtsoulatze, N. Thomos, J. Saltarin and T. Braun

Technischer Bericht INF-15-003 vom 21. September 2015

Institut für Informatik, www.inf.unibe.ch



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

Eirina Bourtsoulatze, Nikolaos Thomos, Jonnahtan Saltarin and Torsten Braun

Technischer Bericht INF-15-003 vom 21. September 2015

CR Categories and Subject Descriptors:
C.2.1 [Computer-Communication Networks]: Network Architecture and Design; C.2.2 [Computer-Communication Networks]: Network Protocols;

General Terms:
Design, Algorithms, Performance

Additional Key Words:
Network coding, information-centric networking, rate allocation, scalable data, forwarding strategy.

Institut für Informatik, Universität Bern

Abstract

In this work, we propose a novel network coding enabled NDN architecture for the delivery of scalable video. Our scheme utilizes network coding in order to address the problem that arises in the original NDN protocol, where optimal use of the bandwidth and caching resources necessitates the coordination of the forwarding decisions. To optimize the performance of the proposed network coding based NDN protocol and render it appropriate for transmission of scalable video, we devise a novel rate allocation algorithm that decides on the optimal rates of Interest messages sent by clients and intermediate nodes. This algorithm guarantees that the achieved flow of Data objects will maximize the average quality of the video delivered to the client population. To support the handling of Interest messages and Data objects when intermediate nodes perform network coding, we modify the standard NDN protocol and introduce the use of Bloom filters, which store efficiently additional information about the Interest messages and Data objects. The proposed architecture is evaluated for transmission of scalable video over PlanetLab topologies. The evaluation shows that the proposed scheme performs very close to the optimal performance.

Contents

1 Introduction **1**

2 System description **5**
- 2.1 Network model 5
- 2.2 Prioritized delivery of scalable video 5
- 2.3 Optimal delivery of SVC video in NDN 7

3 Optimal content-aware flow allocation **8**
- 3.1 Rate allocation problem 8
- 3.2 Subgradient-based algorithm 11

4 Protocol design **15**
- 4.1 Naming 15
- 4.2 Bloom filter based forwarding 16
- 4.3 Handling of Interest messages 17
- 4.4 Handling of Data objects 22
- 4.5 Forwarding strategy 23

5 Performance evaluation **26**
- 5.1 Evaluation scenario 26
- 5.2 Convergence 26
- 5.3 Video quality evaluation 28

6 Conclusions **32**

References **33**

1 Introduction

During 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 often able to produce and share their own video content. This change has been fostered by the emergence of affordable price camera enabled mobile devices with various capabilities, and has resulted in video data dominating the overall Internet traffic [1]. To address users' heterogeneity in terms of display capabilities, processing power, network connectivity, *etc.*, the video is often encoded in multiple qualities and resolutions [2]. This enables users to access the video of their interest encoded in a quality and resolution that matches the capabilities of their device. At the same time, however, this necessitates efficient mechanisms that can enable the delivery of scalable data to heterogeneous clients.

The Internet protocol, designed originally for delay-tolerant applications, such as messaging and file downloading, fails to deal efficiently with the growing volume of time sensitive video traffic. The introduction of protocols like TCP, DASH [3, 4] and RTP/RTSP have permitted to partly handle the real-time and on demand delivery of large volumes of video traffic. However, these solutions have rendered the network operation and management complex. To cope with the inefficiencies of the IP host-centric communication model such as scalability, mobility, *etc.*, Information-Centric Network (ICN) architectures [5] have been proposed as an alternative solution. The ICN communication paradigm focuses on the name of the content rather than on its location. Thus, the content is searched by its name and can be retrieved from any location where it may be 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.

Among the existing ICN architectures, Named Data Networking (NDN) [6] 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 receiver (client) initiates the content delivery by sending Interest messages with the name identifier of the requested content to some (or all) of its outgoing faces. Once an Interest message reaches an uplink node, the node's cache is searched for a matching Data object. If the requested data exists, the node sends the data backwards on the requesting face. If there is an indication that the data will be available at a later time instant, 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 objects 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 NDN and in overall ICN architectures are not yet video ready [7]. For example, in live streaming the main bottleneck of NDN is that each content packet is independently requested by Interest messages, which increases the network load and raises scalability issues. To address this problem, Interest aggregation [8] and persistent Interest packets [9] can be used. However, the use of such approaches is not trivial, as the loss of a single Interest message can lead to the loss of multiple Data objects. In Video-on-Demand systems, the main problem of using ICN arises from the fact that there are no reliable estimations of the available end-to-end bandwidth. Adaptive video streaming over ICN is achieved by deploying the DASH protocol over NDN [10]. 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 that each packet of a data stream should be requested explicitly by sending an Interest message is relaxed by equipping the NDN protocol with network coding capabilities [11]. Network coding [12] can improve the use of the network resources [13], enhance the resilience of the communication, simplify the scheduling, remove the need for coordination, *etc.* With network coding the intermediate network nodes linearly combine the received packets prior to forwarding them to the outgoing links. The linear combinations are performed in a Galois field. To deploy network coding in practical settings, Randomized Linear Network Coding (RLNC) [14] has been proposed. The network coded packets have a header that contains the network coding coefficients and permits the end users to decode the linearly combined packets. The introduction of the concept of generations [15] helps to limit the overhead information carried by each network coded packet and renders it appropriate for video transmission.

The potential of network coding enabled NDN protocols in [11] motivated researchers to explore the use of coding enabled NDN variants. CodingCache [16] has focused on the caching problems and has shown that cache diversity due to network coding helps to increase the cache hit rate. The solution in [11] has been analytically studied in [17] for the butterfly network. Other coding solutions such as Raptor codes have been examined in [18] where RC-NDN, a variant of the NDN, is presented. This

scheme shows the benefits of using Raptor codes in mobile networks.
In this work, we propose a content aware video delivery scheme for network coding enabled NDN architectures. We focus on the transmission of scalable video content [2] in order to deal with the requests of users that have diverse demands in terms of the video quality they want to receive. We employ prioritized random linear network coding (PRLNC) [13, 19] 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 derive the optimal rate allocation for the transmission of Interest messages in order to achieve the flow of Data objects that maximizes the average video quality in the client population. We then present in detail the new features of our network coding enabled NDN architecture. These new features include the appropriate naming scheme and the processing functions that permit to handle both the Interest messages and the Data objects 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 objects but rather a set of network coded packets that belong to the same class and generation, we propose the use of Bloom filters that compactly store additional information about the Interest messages and Data objects. Finally, we design the optimal content-aware forwarding strategy based on the solution of the rate allocation problem. The forwarding strategy guarantees that a sufficient number of Interest messages will be optimally forwarded so that the innovative rate of Data objects 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.
In summary, the main contributions in this work are the following:

- we propose a novel content-aware network coding enabled NDN architecture appropriate for delivering layered data in general and scalable video in particular;

- we formulate the optimal flow rate allocation problem for the transmission of Interest messages so that the achieved rate of Data objects maximizes the average video quality over the client population;

- we redesign the functions that handle the Interest messages and Data objects in order to enable the processing of network coded content;

- we propose the use of Bloom filters in order to resolve problems related to the introduction of network coding in the NDN architecture;
- we design an optimized forwarding strategy that handles the forwarding of Interest messages based on the optimal solution of the flow rate allocation problem.

The rest of the report is organized as follows. In Section 2, we present the overview of the system and discuss the problem of optimally delivering layered video in our setting. Next, in Section 3, we formulate the flow rate allocation problem for the transmission of Interest messages and present the subgradient based optimization algorithm for obtaining the optimal solution. Section 4 discusses the design of the network coding enabled NDN protocol. The performance evaluation of the proposed protocol is presented in Section 5. Finally, Section 6 summarizes the work.

2 System description

2.1 Network model

We consider a multi-hop wireline network with full duplex links that can simultaneously carry information in both directions between pairs of connected nodes. The network is modeled by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of network nodes and network links, respectively. For notational convenience, we assume that every physical link in the network is modelled by a pair of directed links in the graph $\mathcal{G}$, such that $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$, where the tuple (i, j) denotes the directed link from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$. Assuming this convention, the set of network links $\mathcal{E}$ can be written as the union of two disjoint sets $\mathcal{E}_I$ and $\mathcal{E}_D$, *i.e.*, $\mathcal{E} = \mathcal{E}_I \cup \mathcal{E}_D$, such that $(i, j) \in \mathcal{E}_I$ if and only if $(j, i) \in \mathcal{E}_D$. We further assume that the directed graphs $\mathcal{G}_I = (\mathcal{V}, \mathcal{E}_I)$ and $\mathcal{G}_D = (\mathcal{V}, \mathcal{E}_D)$ 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 set of network nodes $\mathcal{V}$ consists of a server node s, a set of intermediate nodes $\mathcal{I}$ and a set of client nodes $\mathcal{U}$. We have $\mathcal{V} = s \cup \mathcal{I} \cup \mathcal{U}$. The server generates video content which is subsequently delivered to the clients through the intermediate nodes following the NDN protocol. The video delivery is initiated by the clients that transmit Interest messages for the desired video content. The Interest messages are routed towards the server according to the routing information stored in the intermediate nodes' Forwarding Information Base tables until a matching Data object is found. The Data object is then transmitted back to the client following the reverse path of that followed by the Interest message.

2.2 Prioritized delivery of scalable video

Due to the heterogeneity of the network clients in terms of bandwidth resources and video display capabilities, the clients may request video content encoded at different bit rates so as to better adapt the quality of the delivered video to the available resources. In order to meet the diverse clients' demands, the video server s encodes the video progressively into L video layers with the scalable extension (SVC) of the H.264/AVC compression standard [2]. The l-th video layer is encoded at rate R_l expressed in packets per second. The video layers include the base layer, which provides the basic video quality, and $L-1$ enhancement layers, which offer an incremental improvement of the video quality. According to the SVC standard, the l-th video layer can be decoded only if all the previous video layers have been successfully decoded. 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 with respect to the previous layers.

In order to improve the network performance in terms of throughput and efficient use of available resources, the server and the intermediate nodes combine the data packets with Prioritized Random Linear Network Coding (PRLNC) [13]. To enable the network coding operations without incurring an additional decoding delay penalty [15], the source data is segmented into generations. Each generation comprises source video packets with similar decoding deadlines. We consider that the data of the l-th layer in each generation is packetized into α_l packets. Prior to transmission, the packets within each generation are encoded by means of PRLNC [13]. 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 layers. The number of source packets that belong to the l-th class is $\beta_l = \sum_{k=0}^{l} \alpha_k$, while class $L-1$ contains $\beta_{L-1} = \sum_{k=0}^{L-1} \alpha_k$ packets and thus includes all the packets of a generation. Packets within each class are then encoded with random linear coding. 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.

The decoding of the network coded packets is done only at the client nodes with the help of Gaussian elimination upon receiving a sufficient number of network coded packets. A client can decode a generation of the l-th video layer only upon receiving β_l innovative packets of this generation from classes $0, 1, \ldots, l$. Note that class l contains up to β_l innovative network coded packets. The PRLNC method respects the intrinsic prioritized structure of the video data by assigning higher importance to layers with smaller index. 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.

2.3 Optimal delivery of SVC video in NDN

In our network coding enabled NDN architecture for scalable video delivery, we adopt the convention that every Data object represents a single network coded packet of some class l and generation g. The Interest messages 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. We assume that the clients generate Interest messages at a constant rate. In this context, our goal is to ensure a constant streaming bit rate and playback quality at each client.
The time constrained nature of the video imposes strict requirements on the video delivery deadlines. An optimized forwarding strategy is therefore essential in order to achieve the delivery deadlines set by the video application. In this work, we propose to design the forwarding strategy based on the optimal rate allocation for Interest messages that maximizes the average quality of the video delivered to the client population.
Let $\boldsymbol{f} = \{f^0, f^1, \ldots, f^{L-1}\}$ be a set of flows, where f^l, $\forall l \in \mathcal{L}$, is the flow of Data objects of class l in the network and $\mathcal{L} = \{0, 1, \ldots, L-1\}$ is the set of packet classes. The average video quality in the client population can be written as

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

where $U = |\mathcal{U}|$ is the number of clients in the network and $Q_u(\boldsymbol{f})$ is the quality of the video delivered to client u as a function of the flows f^l of the Data objects. The function $Q_u(\boldsymbol{f})$ is a piecewise constant function. It can be expressed as a linear combination of indicator functions as

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

where q_l is the video quality achieved after decoding the l-th video layer with $q_{-1} = 0$. The indicator function $\mathbb{1}_l(\boldsymbol{f})$ is defined as

$$\mathbb{1}_l(\boldsymbol{f}) = \begin{cases} 1, & \text{if the } l\text{-th video layer can be decoded given } \boldsymbol{f} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Given that our objective is to maximize the average video quality in (1), the problem boils down to (i) determining the rate of the Interest messages for each class of packets that must be transmitted from the clients, and (ii) designing the optimal forwarding strategy at clients and intermediate nodes, in order to achieve the set of flows $\{f^0, f^1, \ldots, f^{L-1}\}$ that maximize the average video quality.

3 Optimal content-aware flow allocation

We now present our algorithm for determining the optimal forwarding strategy at the clients and at the intermediate nodes in a network coding enabled NDN architecture for the delivery of scalable video. We derive the forwarding strategy by casting this problem in a rate allocation problem that aims at maximizing the average video quality in the client population. Our approach relies on the observation that Interest messages 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 only a single Interest message for a network coded packet of this class and generation has to be transmitted further towards the server to fetch the Data object that will consume all the aggregated Interest messages. Taking into account this property, we can make use of the concept of *conceptual flows* presented in [20] to design our content-aware algorithm for optimal forwarding of Interest messages. In particular, the overall flow of Interest messages from the clients to the servers can be regarded as consisting of U unicast conceptual flows from each client to the server, where $U = |\mathcal{U}|$ is the number of clients in the network. Conceptual flows are network flows that can co-exist in the network without contending for the link bandwidth. Therefore, 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.

3.1 Rate allocation problem

Let $r^{u,l}$ denote the conceptual flow of Interest messages expressing a request for network coded packets of class 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 Interest packets for class l network coded packets on link (i, j) will be $z_{ij}^l = \max_{u \in \mathcal{U}} r_{ij}^{u,l}$. Let f_{ji}^l denote the rate of the flow of Data objects of class l on the link $(j, i) \in \mathcal{E}_D$. Since every Interest message transmitted on link $(i, j) \in \mathcal{E}_I$ is eventually consumed by a matching Data object transmitted on link $(j, i) \in \mathcal{E}_D$, the rate of the flow of Data objects on link (j, i) is equal to the rate of the flow of Interest messages z_{ij}^l on link (i, j), *i.e.*, we have $z_{ij}^l = f_{ji}^l$, $\forall (i, j) \in \mathcal{E}_I$ and $(j, i) \in \mathcal{E}_D$.
Recall from Section 2.3 that the average video quality function $\overline{Q}(\boldsymbol{f})$ is a

sum of piecewise constant functions, and is therefore also a piecewise constant function. That means that there can be multiple sets of flow values that maximize the function in (1). In order to resolve this ambiguity and give higher priority to more important classes of packets, we introduce a flow cost function for each user u that represents the overall cost of requesting packets of different classes:

$$C_u(\boldsymbol{r}^u) = \boldsymbol{c}^T \boldsymbol{r}^u = \sum_{l=0}^{L-1} c_l \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,l} \tag{4}$$

where $\boldsymbol{c} = (c_0, c_1, \dots, c_{L-1})^T$ is the cost vector, $\boldsymbol{r}^u = (r^{u,0}, r^{u,1}, \dots, r^{u,L-1})^T$ is the vector of conceptual flows of different classes 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. c_l is the cost for requesting network coded packets of class l. The costs can be chosen arbitrarily, and must satisfy the following constraints:

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

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

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 are given higher priority in the rate allocation algorithm compared to the less important classes. Constraint (5b) ensures that the introduction of the cost function into the optimization problem does not alter the value of the objective function in (1) at the optimal rate allocation solution.

By including the cost function in (4) 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 flow cost. Formally, this optimization problem can be written as:

$$\arg\max_{\boldsymbol{f}, \boldsymbol{r}^u} \frac{1}{U} \sum_{u\in\mathcal{U}} \Big(Q_u(\boldsymbol{f}) - C_u(\boldsymbol{r}^u) \Big) \tag{6}$$

Taking into account that the quality of the delivered video at client u depends only on the rate of the flow of Data objects on the input links of the client, the video quality at user u can be expressed as a function of the rates of the conceptual flows of Interest messages as follows:

$$\begin{aligned} Q_u(\boldsymbol{f}) &= Q_u\Big(\sum_{j:(j,u)\in\mathcal{E}_D} f_{ju}^0, \sum_{j:(j,u)\in\mathcal{E}_D} f_{ju}^1, \dots, \sum_{j:(j,u)\in\mathcal{E}_D} f_{ju}^{L-1}\Big) \\ &= Q_u\Big(\sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,0}, \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,1}, \dots, \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,L-1}\Big) \\ &= Q_u(\boldsymbol{r}^u) \end{aligned} \tag{7}$$

where we have used the fact that $f_{ju}^l = z_{uj}^l$ and $z_{uj}^l = \max_{w\in\mathcal{U}} r_{wj}^{w,l} = r_{uj}^{u,l}$. By combining (6), (4) and (7), the objective function in the optimization problem in (6) can be rewritten as:

$$\arg\max_{\boldsymbol{r}\in\mathcal{R}} \frac{1}{U}\sum_{u\in\mathcal{U}}\Big(Q_u(\boldsymbol{r}^u) - \boldsymbol{c}^T\boldsymbol{r}^u\Big) \tag{8}$$

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:

$$r_{ij}^{u,l} \geq 0, \quad \forall u\in\mathcal{U},\ \forall(i,j)\in\mathcal{E}_I,\ \forall l\in\mathcal{L} \tag{9a}$$

$$\sum_{n:(n,i)\in\mathcal{E}_I} r_{ni}^{u,l} = \sum_{j:(i,j)\in\mathcal{E}_I} r_{ij}^{u,l}, \quad \forall u\in\mathcal{U},\ \forall i\in\mathcal{I},\ \forall l\in\mathcal{L} \tag{9b}$$

$$\sum_{k=0}^{l}\sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,k} \leq \sum_{k=0}^{l} R_k, \quad \forall u\in\mathcal{U},\ \forall l\in\mathcal{L} \tag{9c}$$

$$x_{ij}^l \geq r_{ij}^{u,l}, \quad \forall u\in\mathcal{U},\ \forall(i,j)\in\mathcal{E}_I,\ \forall l\in\mathcal{L} \tag{9d}$$

$$\sum_{l\in\mathcal{L}} x_{ij}^l(p_I + p_D) \leq B_{ij} \quad \forall(i,j)\in\mathcal{E}_I \tag{9e}$$

where p_I and p_D in (9e) are the size (in bits) of the Interest messages and Data objects, respectively. The objective function in (8) is a function of only the rates of the conceptual flows of Interest messages on the input links of the clients. The set of constraints (9a) -(9e) defines the set of feasible rate allocations for the conceptual flows of Interest messages. Specifically, constraint (9b) is the flow conservation constraint that holds for every conceptual flow of Interest messages, since every conceptual flow is a unicast flow from the client to the server. This constraint guarantees that the number of Interest messages that enter an intermediate node is equal to the number of Interest messages that are transmitted from the node for every conceptual flow and for every class of network coded packets. Constraint

(9c) is the innovative[1] rate constraint which states that the rate of Interest messages generated by a client for each class of network coded packets should not exceed the maximum rate of innovative Data objects that can be provided by the source. Constraints (9b) and (9c) together ensure that the clients do not request Data objects at a rate which is higher than the rate at which innovative Data objects are generated by the server. The variable x_{ij}^l in (9d) is an auxiliary variable that represents the actual transmission rate of Interest messages for network coded packets of class l on link (i,j). It upper bounds the transmission rates of the conceptual flows of Interest messages on every link. The actual rates x_{ij}^l are further bounded by the available bandwidth on the link which is captured by the constraints in (9e). Constraint (9e) states that the bandwidth required to transmit the Interest messages and the Data objects that consume these Interest messages on a link, should not exceed the overall bandwidth available for that link.

3.2 Subgradient-based algorithm

In this section, we present an efficient algorithm based on Lagrangian relaxation and the subgradient method for solving the optimization problem formulated in (8). As previously discussed, the video quality function in (7) is a piecewise constant function and therefore, the objective function in (8) is piecewise affine and thus also piecewise concave.
We first relax the coupling constraints in (9d) and obtain the Lagrangian dual function. This choice is driven by 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

$$\begin{aligned} L(\boldsymbol{\mu}) &= \max_{\mathcal{R}'} \frac{1}{U} \sum_{u \in \mathcal{U}} \Big(Q_u(\boldsymbol{r}^u) - \boldsymbol{c}^T \boldsymbol{r}^u \Big) - \sum_{u \in \mathcal{U}} \sum_{(i,j) \in \mathcal{E}_I} \sum_{l \in \mathcal{L}} \mu_{ij}^{u,l} (r_{ij}^{u,l} - x_{ij}^l) \\ &= \max_{\mathcal{R}'} \sum_{u \in \mathcal{U}} \Big(\frac{1}{U} (Q_u(\boldsymbol{r}^u) - \boldsymbol{c}^T \boldsymbol{r}^u) - \sum_{(i,j) \in \mathcal{E}_I} \sum_{l \in \mathcal{L}} \mu_{ij}^{u,l} r_{i,j}^{u,l} \Big) \\ &+ \sum_{(i,j) \in \mathcal{E}_I} \sum_{l \in \mathcal{L}} \Big(\sum_{u \in \mathcal{U}} \mu_{ij}^{u,l} \Big) x_{ij}^l \end{aligned} \tag{10}$$

where $\mathcal{R}'$ is the polytope defined by the linear constraints (9a), (9b), (9c)

[1]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.

and (9e). The Lagrangian dual of the primal problem in (8) is then

$$\min_{\boldsymbol{\mu}\geq 0} L(\boldsymbol{\mu}) \tag{11}$$

where $\boldsymbol{\mu}$ is the vector of Lagrangian multipliers $\mu_{ij}^{u,l}$. We can observe that the Lagrangian subproblem in (10) can be decomposed into several optimization subproblems that can be solved independently. These optimization subproblems consist of U maximization problems

$$\arg\max_{\boldsymbol{r}\in\mathcal{R}_u} \frac{1}{U}\Big(Q_u(\boldsymbol{r}^u) - \boldsymbol{c}^T\boldsymbol{r}^u\Big) - \sum_{(i,j)\in\mathcal{E}_I}\sum_{l\in\mathcal{L}} \mu_{ij}^{u,l} r_{i,j}^{u,l}, \quad \forall u \in \mathcal{U} \tag{12}$$

where the polytope $\mathcal{R}_u$ is defined for every user u as

$$r_{ij}^{u,l} \geq 0, \quad \forall (i,j) \in \mathcal{E}_I,\ \forall l \in \mathcal{L} \tag{13a}$$

$$\sum_{n:(n,i)\in\mathcal{E}_I} r_{ni}^{u,l} = \sum_{j:(i,j)\in\mathcal{E}_I} r_{ij}^{u,l}, \quad \forall i \in \mathcal{I},\ \forall l \in \mathcal{L} \tag{13b}$$

$$\sum_{k=0}^{l}\sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,k} \leq \sum_{k=0}^{l} R_k, \quad \forall l \in \mathcal{L} \tag{13c}$$

and $|\mathcal{E}_I|$ maximization problems

$$\begin{aligned} &\arg\max_{\boldsymbol{x}} \sum_{l\in\mathcal{L}}\Big(\sum_{u\in\mathcal{U}}\mu_{ij}^{u,l}\Big)x_{ij}^{l}, \quad \forall (i,j)\in\mathcal{E}_I \\ &\text{s.t.} \quad \sum_{l\in\mathcal{L}} x_{ij}^{l}(p_I + p_D) \leq B_{ij} \end{aligned} \tag{14}$$

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 messages. The maximization problems in (14) are linear programs that can be solved with one of the general LP optimization methods, such as the Simplex algorithm. The optimization problem in (12) consists in maximizing a piecewise linear objective function. Since it is hard to directly optimize the objective function in (12), the optimization problem in (12) can be further transformed into a two level optimization problem:

$$\max_{k\in\mathcal{L}}\max_{\mathcal{R}_u\cap\mathcal{R}_k} \frac{1}{U}\Big(q_k - \boldsymbol{c}^T\boldsymbol{r}^u\Big) - \sum_{(i,j)\in\mathcal{E}_I}\sum_{l\in\mathcal{L}} \mu_{ij}^{u,l} r_{i,j}^{u,l}, \quad \forall u \in \mathcal{U} \tag{15}$$

where $\mathcal{R}_u$ is defined in (13a)-(13c). 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

$$\sum_{m=1}^{l} \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,m} \leq \sum_{m=1}^{l} R_m, \ \forall l \in \{0,\dots,k-1\} \tag{16a}$$

$$\sum_{m=1}^{k} \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,m} = \sum_{m=1}^{k} R_m \tag{16b}$$

$$\sum_{m=1}^{l} \sum_{j:(u,j)\in\mathcal{E}_I} r_{uj}^{u,m} < \sum_{m=1}^{l} R_m, \ \forall l \in \{k+1,\dots,L\} \tag{16c}$$

At the lower level of the optimization problem in (15), the objective function is linear and is maximized over the rate region where the k-th class of network coded packets is decodable. Constraints (16a) and (16b) guarantee that the rate of the Data objects that will be delivered to the client is sufficient to decode network coded packets of class k, while at the same time the rate of Data objects per each class does not exceed the available innovative rate for this class of packets. Constraint (16c) ensures that the rate of Data objects delivered to the client is not sufficient to decode network coded packets of classes higher than k. At the higher level of the optimization problem in (15), a pointwise 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 (11), 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 $t = 1, 2, \dots$ according to the following rule:

$$\mu_{ij}^{u,l}[t] = \max\{0, \mu_{ij}^{u,l}[t-1] + \theta[t](r_{ij}^{u,l}[t] - x_{ij}^{u,l}[t])\}, \quad \forall(i,j) \in \mathcal{E}_I, \ \forall u \in \mathcal{U}, \ \forall l \in \mathcal{L} \tag{17}$$

where $r_{ij}^{u,l}[t]$ and $x_{ij}^{u,l}[t]$ are the solutions to the optimization problems in (12) and (14) 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

$$\theta[t] > 0, \quad \lim_{t\to\infty} \theta[t] = 0, \quad \sum_{t=1}^{\infty} \theta[t] = \infty \tag{18}$$

the subgradient algorithm is guaranteed to converge to the optimal solution of the Lagrangian dual problem in (11)[21]. Here we choose $\theta[t] =$

$a/(b+ct)$, $\forall t$, with $a > 0$, $b \geq 0$ and $c > 0$, which satisfies the conditions in (18). In order to recover the optimal solution to the primal problem in (8), we use the method introduced by Sherali *et al.* [21]. Specifically, at every iteration t, the value of the primal variables $r_{ij}^{u,l}$ and x_{ij}^{l} is constructed according to the following update rule:

$$\hat{r}_{ij}^{u,l}[t] = \sum_{h=1}^{t} \nu_h[t] r_{ij}^{u,l}[h] \tag{19a}$$

$$\hat{x}_{ij}^{l}[t] = \sum_{h=1}^{t} \nu_h[t] x_{ij}^{l}[h] \tag{19b}$$

where $\sum_{h=1}^{t} \nu_h[t] = 1$ aned $\nu_h[t] \geq 0$, for $h = 0, 1, \dots, t$. A valid choice for the convex combination weights $\nu_h[t]$ is to set $\nu_h[t] = 1/t$, $\forall h = 0, 1, \dots, t, \ \forall t$. The accumulation point of the sequence of primal values $\{\hat{r}_{ij}^{u,l}[t]\}$ and $\{\hat{x}_{ij}^{l}[t]\}$ generated via (19a) and (19b) is then a feasible and optimal solution to the optimization problem in (8) [21]. For this particular choice of convex combination weights, the sequences of primal values generated via (19a) and (19b) can be calculated recursively as

$$\hat{r}_{ij}^{u,l}[t] = \frac{t-1}{t}\hat{r}_{ij}^{u,l}[t-1] + \frac{1}{t} r_{ij}^{u,l}[t] \tag{20a}$$

$$\hat{x}_{ij}^{l}[t] = \frac{t-1}{t}\hat{x}_{ij}^{l}[t-1] + \frac{1}{t} x_{ij}^{l}[t] \tag{20b}$$

4 Protocol design

Since both the server and the intermediate nodes perform network coding operations on the data packets, the standard NDN protocol requires mechanisms that would permit to handle Interest messages and Data objects in a network coding enabled setting. In this section, we discuss the new features and functionalities that we introduce in the NDN architecture, in order to enable the processing of the Data objects and Interest messages when network coding is performed in the NDN nodes. We also present our content aware forwarding strategy that builds upon our rate allocation algorithm described in Section 3.2 and aims at the efficient delivery of the video content to the clients at the best possible quality. Central to our protocol and forwarding strategy design is the requirement to maintain a sufficiently high innovative rate of network coded packets in order to ensure the timely delivery of the video content.

4.1 Naming

The naming scheme of our network coding enabled NDN architecture follows the standard hierarchical naming of the NDN protocol and has the general format */component1/.../componentN/NCFlag/PacketId/GenIndx*. The *NCFlag* component enables the identification and subsequent processing of Interest messages 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 object. Thus, the *NCFlag* permits the nodes to distinguish between the two types of Interest messages and to invoke the appropriate processing functions. The *NCFlag* is followed by the *PacketId* name component, which 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 object, the *PacketId* is understood as the class which the network coded Data object belongs to. In this case, the name in the Interest message no longer specifies a unique Data object but rather refers to any network coded packet that belongs to 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.

4.2 Bloom filter based forwarding

Unlike the original NDN protocol, where the content name in the Interest message identifies a unique Data object, in our network coding enabled NDN architecture the Interest messages 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 message requesting a network coded Data object of a certain class and generation can 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 messages with the same content name in order to obtain all the packets of a certain class and generation. This impedes the use of pipelining and entails a potential risk of consuming two distinct Interest messages with the same linear combination of source packets, thus reducing the innovative packet rate. It is thus necessary to provide the NDN nodes with a mechanism that would permit to distinguish between two Interest messages with the same content name and to decide whether they can be consumed by the same Data object.

In order to resolve the ambiguity created by the lack of unique mapping between content names and the data, we make use of Bloom filters [22] that store some additional information about the Interest messages and the Data objects. Specifically, we add a new field that contains a Bloom filter in both the Interest message and the Data object. This Bloom filter is a compact representation of a set whose elements are the IDs of the clients. When a client generates an Interest message, it inserts its ID in the originally empty Bloom filter. As the Interest message is forwarded towards the location of the data, the content of its Bloom filter is modified by the forwarding strategy according to some rules derived based on the optimal rate allocation, as will be explained in Section 4.5. The content of the Bloom filter included in an Interest message can be interpreted as the set of clients that are the destination nodes for the Data object that will consume this Interest message. Thus if, for example, two Interest messages with the same content name arrive at a node, they can be consumed by the same Data object as long as the intersection of their Bloom filters is empty, which means that the same data will not be forwarded multiple

times to the same client. When a network coded Data object consumes an Interest message, the content of the Bloom filter of the Interest message is copied to the Bloom filter of the newly generated network coded Data object. In that way, the Data object contains a compact representation of the set of destination nodes where this Data can be potentially forwarded. Thus, when a Data object arrives at a node, it can only consume the Interest message whose Bloom filter contains a subset of the client IDs stored in the Bloom filter of the Data object. In addition, once this Data object has been forwarded to some of the clients whose IDs where originally in its Bloom filter, these clients are removed from the list of potential consumers of this data so as to avoid sending the same Data object to the same client multiple times. More details on the use of Bloom filters for the processing of Interest messages and Data objects will be given in Sections 4.3 and 4.4, respectively.
Finally, since the Bloom filters are an essential element of our protocol design and play an important role in the handling of the packets, they are also present in both the Pending Interest Table (PIT) and the Content Store (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 $\langle face, Bloom\ filter \rangle$, which indicates the incoming face and the Bloom filter of the Interest message which arrived on this face. Similarly, for every Data object that is stored in the CS, its Bloom filter is also stored in the CS along with the content name and the payload. Since the removal of elements from a Bloom filter is not permitted [22], a second Bloom filter is added for each data entry, which stores the “sent” information, *i.e.*, the set of client IDs which the Data object has already been forwarded to.

4.3 Handling of Interest messages

When an Interest message arrives on some face, the PIT is first checked for a pending Interest message requesting the same content. Unlike the standard NDN protocol, where a positive outcome is observed if the content name of the incoming Interest message matches the content name of some pending Interest message, this criterion is not sufficient for the PIT lookup procedure in the case of the network coded data. This insufficiency stems from the fact that the content name in our framework no longer specifies a unique Data object, but rather a collection of Data objects, *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 messages

Pending Interest Table (PIT)

Content name	**Requesting faces** $\langle face, Bloomfilter\rangle$
/unibe.ch/videos/foreman.qcif/NC/class_0/gen_2	$\langle 2, 0010010\ldots 101\rangle$
	$\langle 3, 0101010\ldots 010\rangle$
/unibe.ch/videos/foreman.qcif/NC/class_1/gen_3	$\langle 1, 0010010\ldots 101\rangle$
	$\langle 0, 1001110\ldots 000\rangle$
	$\langle 3, 0101010\ldots 010\rangle$
$\cdots$	$\cdots$

(a)

Content Store (CS)

Content name	**Bloom filters** $\langle original, sent\rangle$	**Content**
/unibe.ch/videos/foreman.qcif/NC/class_0/gen_2	$\langle 0010010\ldots 101, 0000010\ldots 001\rangle$	$\cdots$
/unibe.ch/videos/foreman.qcif/NC/class_1/gen_3	$\langle 0010010\ldots 101, 0010010\ldots 101\rangle$	$\cdots$
$\cdots$	$\cdots$	$\cdots$

(b)

Figure 1: (a) Pending Interest Table (PIT) and (b) Content Store (CS) in our network coding enabled NDN architecture.

with the same content name as the number of packets in the specified class. It is therefore highly likely that two Interest messages with the same content name may be requesting two linearly independent network coded packets and cannot be consumed by the same Data object. In order to distinguish between the case where the incoming Interest message can be consumed by the same Data object as a pending Interest message and the case where the incoming Interest message must be treated as a request for a new Data object, additional criteria must be fulfilled for the PIT lookup procedure. These criteria are based upon the information that is stored in the Bloom filters of the Interest messages.

Let I denote the incoming Interest message. Algorithm 1 summarizes the PIT lookup for a pending Interest message upon the arrival of I. The algorithm takes as input the set $\mathcal{P}_I$ of pending Interest messages in the PIT whose content name matches the content name of I, and outputs the Boolean variable *PendingInterestExists*, which indicates whether a matching pending Interest message was found. If a matching pending Interest message exists in the PIT, the algorithm also outputs a pointer I_x to the PIT entry where it is stored. The PIT lookup is performed as follows. For every

Algorithm 1 PIT lookup procedure upon arrival of the Interest message I

```
1: Input: $I$, $\mathcal{P}_I$
2: Output: PendingInterestExists, $I_x$
3: Initialization: PendingInterestExists $\leftarrow$ false,
   $\mathcal{U}_I \leftarrow \{u \in \mathcal{U} \mid u$ in BF$_I\}$
4: while $\mathcal{P}_I \neq \emptyset$ do
5:    Select the first $I'$ in $\mathcal{P}_I$
6:    Compute BF$_{I'}^{union}$, i.e. the union of Bloom filters in the list of request-
      ing faces associated with $I'$
7:    if BF$_{I'}^{union} \cap$ BF$_u \neq$ BF$_u$ for all $u \in \mathcal{U}_I$ then
8:       PendingInterestExists $\leftarrow$ true, $I_x \leftarrow I'$, $\mathcal{P}_I \leftarrow \emptyset$
9:    else
10:      $\mathcal{P}_I \leftarrow \mathcal{P}_I \backslash I'$
11:   end if
12: end while
13: return PendingInterestExists, $I_x$
```

pending Interest message I' in $\mathcal{P}_I$, we first compute the union BF$_{I'}^{union}$ of the Bloom filters 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 object. We then compare the set of client IDs stored in BF$_{I'}^{union}$ with the set $\mathcal{U}_I$ of the client IDs stored in the Bloom filter BF$_I$ of I. If none of the client IDs in $\mathcal{U}_I$ is present in BF$_{I'}^{union}$, I' is considered as a matching pending Interest message. This essentially means that I and I' can be consumed by the same Data object. It is worth mentioning that the use of the Bloom filters 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 objects.

If a pending Interest message is found in the PIT, the arrival face of I as well as its Bloom filter BF$_I$ are stored in the list of requesting faces of the entry pointed by I_x and no further action is taken. If the PIT lookup procedure does not yield any matching pending Interest message, the Forwarding Information Base (FIB) is checked for a matching entry. The FIB lookup procedure is identical to the one in the standard NDN protocol and is based only on the content name. If a matching FIB entry is found, the Interest message is forwarded according to the forwarding strategy that will be described in Section 4.5, and then inserted as a new entry in the PIT.

Algorithm 2 CS lookup procedure upon arrival of the Interest message I

```
1:  Input: $I$, $\mathcal{D}_I$
2:  Output: MatchingDataExists, $\mathcal{D}'_I$
3:  Initialization: $\mathcal{U}'_I \leftarrow \emptyset$,  $\mathcal{D}'_I \leftarrow \emptyset$,
    MatchingDataExists $\leftarrow$ false, $\mathcal{U}_I \leftarrow \{u \in \mathcal{U} \mid u \in \text{BF}_I\}$
4:  while $\mathcal{U}_I \neq \emptyset$ and $\mathcal{D}_I \neq \emptyset$ do
5:      Select the first $D$ in $\mathcal{D}_I$
6:      for every $u \in \mathcal{U}_I$ do
7:          if $u \in \text{BF}_D$ and $u \notin \text{BF}_D^{sent}$ then
8:              $\mathcal{D}'_I \leftarrow \mathcal{D}'_I \cup D, \mathcal{U}'_I \leftarrow \mathcal{U}'_I \cup u$
9:          end if
10:     end for
11:     $\mathcal{D}_I \leftarrow \mathcal{D}_I \backslash D, \mathcal{U}_I \leftarrow \mathcal{U}_I \backslash \mathcal{U}'_I, \mathcal{U}'_I \leftarrow \emptyset$
12: end while
13: if $\mathcal{U}_I == \emptyset$ then
14:     MatchingDataExists $\leftarrow$ true
15: end if
16: 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 object that can potentially consume I. In order to determine whether the CS contains such a Data object, we once again make use of the information stored in the Bloom filters of I and the Data objects 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 objects in the CS, whose content name matches the content name of I. For every client u whose ID is stored in the Bloom filter BF$_I$ of I, the algorithm searches the set $\mathcal{D}_I$ for a Data object D, that has the same client ID stored in its Bloom filter BF$_D$, but not in the Bloom filter BF$_D^{sent}$, which keeps track of the clients to whom D has been already forwarded. If a Data object satisfying these criteria is found for all the clients in $\mathcal{U}_I$, then we consider that a Data object that matches I can be generated from the Data objects that are stored in the CS. The algorithm outputs the Boolean variable *MatchingDataExists*, which indicates whether a Data object can be generated to consume I. In case of a positive outcome, the algorithm also outputs a set $\mathcal{D}'_I$ of Data objects that must be combined with RLNC in order to consume I. This CS lookup procedure guaranties that a Data object that has already been sent in response to an Interest message originating from some client u, will not be retransmitted multiple times to u.

Algorithm 3 CS update procedure after transmission of Data object that consumes the Interest message I

```
1: Input: $I$, $\mathcal{D}'_I$
2: Initialization: $\mathcal{U}_I \leftarrow \{u \in \mathcal{U} \mid u \in \text{BF}_I\}$
3: while $\mathcal{U}_I \neq \emptyset$ do
4:     Pick $u \in \mathcal{U}_I$
5:     Find $D \in \mathcal{D}'_I$ s.t. $u \in \text{BF}_D$ and $u \notin \text{BF}_D^{sent}$
6:     Insert $u$ in $\text{BF}_D^{sent}$
7:     $\mathcal{U}_I \leftarrow \mathcal{U}_I \backslash u$
8: end while
```

If the outcome of the Algorithm 2 is true, a network coded Data object is generated by combining the Data objects in the set D_I with RLNC.[2] Its Bloom filter is set equal to the Bloom filter 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 BF_I of I, are inserted in the Data objects 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 objects 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 message is inserted as a new entry in the PIT and remains there until a matching Data object 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 messages in order to receive a sufficient amount of innovative content but not all the data has yet arrived at the node.

Remark: The order in which the data structures are checked in our network coding enabled NDN protocol differs from that of the standard NDN in order to guarantee that a sufficient number of innovative packets will be delivered to all clients. The FIB match is preferred over a CS match due to the fact that some Interest messages require the linear combination of multiple Data objects in order to be consumed. Since these Data objects arrive at the node asynchronously, it may happen that another Interest, that can be consumed by the Data objects already stored in the CS, arrives at the node before the last Data object. Thus, if the CS is checked first, some

[2]Note that this is equivalent to combining only the Data objects in $\mathcal{D}'_I$.

Algorithm 4 PIT lookup procedure upon arrival of an innovative Data object D

```
 1: Input: $\mathcal{P}_D$, $\mathcal{D}_D$
 2: Output: PendingInteretsExists, $\mathcal{D}'_D$ and $I$
 3: Initialization: PendingInteretsExists $\leftarrow$ false, $\mathcal{D}'_D \leftarrow \emptyset$
 4: while $\mathcal{P}_D \neq \emptyset$ do
 5:     Select the first $I'$ in $\mathcal{P}_D$
 6:     Run Algorithm 2 with $I'$, $\mathcal{D}_D$ as input
 7:     Let MatchingDataExists, $\mathcal{D}'_D$ be the output of step 6
 8:     if MatchingDataExists == true then
 9:         PendingInteretsExists $\leftarrow$ true, $I \leftarrow I'$, $\mathcal{P}_D \leftarrow \emptyset$
10:     else
11:         $\mathcal{P}_D \leftarrow \mathcal{P}_D \backslash I'$
12:     end if
13: end while
14: return PendingInteretsExists, $\mathcal{D}'_D$ and $I$
```

of the Data objects may be used to consume the newly arrived Interest message. In that case, when the last Data object arrives, there is not sufficient innovative data to consume the pending Interest message, which will remain in the PIT until it expires, reducing thus the flow of innovative Data objects.

4.4 Handling of Data objects

When a Data object arrives, the timestamp of the video packet is first compared to the current time. If according to the time stamp the packet has expired, the Data object is discarded, since it is no longer useful for the clients.

If the incoming Data object has not expired, a PIT lookup is performed for a matching pending Interest message. A PIT match is determined based on not only the content name, as in the standard NDN protocol, but also on the information stored in the Bloom filters of the incoming Data object and of the pending Interest messages in the PIT. The PIT lookup procedure upon arrival of an innovative Data object D is described in Algorithm 4. The algorithm takes as input the set $\mathcal{P}_D$ of pending Interest messages in the PIT whose content name matches the content name of D, and the set $\mathcal{D}_D$ of Data objects stored in the CS, whose content name matches the content name of D. For every pending Interest message I' in the set

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	$\langle 0, c^0_{1,3}\rangle$ $\langle 2, c^2_{1,3}\rangle$
...	...

(a)

Figure 2: Forwarding Information Base (FIB) in our network coding enabled NDN architecture.

$\mathcal{P}_D$, the CS lookup procedure described in 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 objects is found, Algorithm 4 outputs the pending Interest message I that can be consumed as well as the set $\mathcal{D}'_D$ of Data objects in CS that are required to consume I. A network coded Data object is then generated by combining all the Data objects 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. The pending Interest message I is then removed from the PIT and the CS is updated using the procedure described in Algorithm 3 with input I and $\mathcal{D}'_D$. The above procedure is repeated until no matching pending Interest message can be found.

4.5 Forwarding strategy

The forwarding strategy controls the forwarding of Interest messages, when no matching pending Interest message can be found in the PIT, and performs the tasks of selecting the outgoing face and of modifying appropriately the Bloom filter of the forwarded Interest message in order to achieve the packet rate dictated by the rate allocation algorithm.

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 lm. 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. Since our forwarding strategy requires integer rate values, we recover the integer rate allocation solution by choosing the integer rate allocation vector which has the minimum Euclidean distance from the optimal rate allocation vector derived in Section 3, under the constraint that the integer rate allocation vector results in decoding the same set of video layers as the non-integer solution. We then set $c^f_{l,g} = \tilde{z}^l_{ij}$ for every generation g, where $\tilde{z}^l_{ij}$ is the integer rate of the actual flow of Interest messages for network coded Data objects of class l on the link (i, j) and f is the face associated with this link. When an FIB lookup is performed, the content name of the Interest message 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 message is forwarded on all the faces in the list for which the counter is non zero. Once the Interest message is scheduled for transmission, the corresponding counter is reduced by one. 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 messages 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 message that is being forwarded. The selection procedure is described in Algorithm 5. The variables $\tilde{r}^{u,l}_{ij}$ are the integer rates of the conceptual flows of Interest messages for network coded packets of class l on link (i, j). The algorithm first computes the counter p which indicates the number of Interest messages of class l and generation g that will have been transmitted on face f including the Interest message 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 message. 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. To better illustrate the intuition behind this algorithm let us give an example. Let $\tilde{z}^l_{ij} = 9$, $\tilde{r}^{u_1,l}_{ij} = 9$ and $\tilde{r}^{u_2,l}_{ij} = 3$. Since the rate of the conceptual flow of client u_1 is equal to the actual rate of Interest

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

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

messages, the ID of the client u_1 will be inserted in all Interest messages forwarded on this link. On the other hand, only $\tilde{z}^l_{ij}/\tilde{r}^{u_2,l}_{ij} = 1/3$ of the Interest messages originate from client u_2, which means that only $1/3$ of the Data objects delivered on this link will be useful for client u_2. Therefore, the ID of the client u_2 will be inserted only in every third Interest message forwarded on this link.

5 Performance evaluation

5.1 Evaluation scenario

We consider that the server encodes the Forman 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 78 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 objects is $p_D = 1600$ bytes and the size of the Interest Messages is $p_I = 200$ bytes. The generation size is set equal to the size of the GOP, *i.e.*, 78 packets. That means that 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 cost coefficients used in the rate allocation algorithm 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 [15].
We evaluate the performance of our proposed architecture on the network topology depicted in Fig. 3. This topology was constructed based on real data collected by the PlanetLab project [23]. 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 [24].

5.2 Convergence

We first investigate the convergence of our rate allocation algorithm presented in Section 3. In Fig. 4 we illustrate the evolution of the cumulative rate of Interest messages sent by node 29 with the number of iterations of the rate allocation algorithm presented in Section 3 for different values of the link bandwidth. Each figure shows the convergence of the rate of Interest messages requesting network coded packets that belong to the three

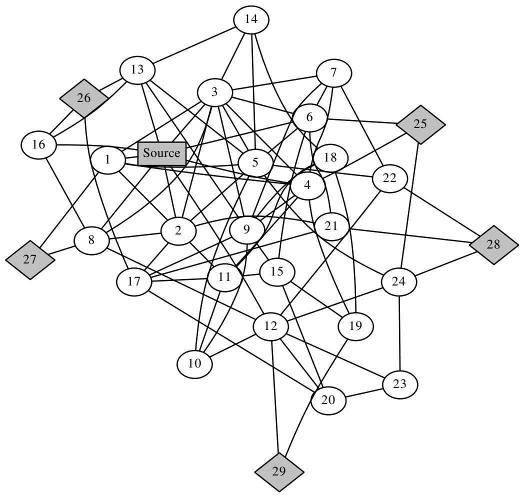


Figure 3: PlanetLab topology consisting of one source node (node 0), 24 intermediate nodes and 5 client nodes (nodes 25 - 29.)

available packet classes. We can see that when the link bandwidth is low, the client requests only network coded packets from class 0 which consists of packets that belong to the base layer. The rate of Interest messages requesting class 0 packets converges to the value of 38 packets/sec which is the encoding rate of the base layer. As the link bandwidth increases, the client starts to request also class 1 network coded packets, which results in decoding the first enhancement layer. The rate of the Interest messages 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, 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, we can observe that 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 (4) and reflects the prioritization of the packet classes according to their importance.

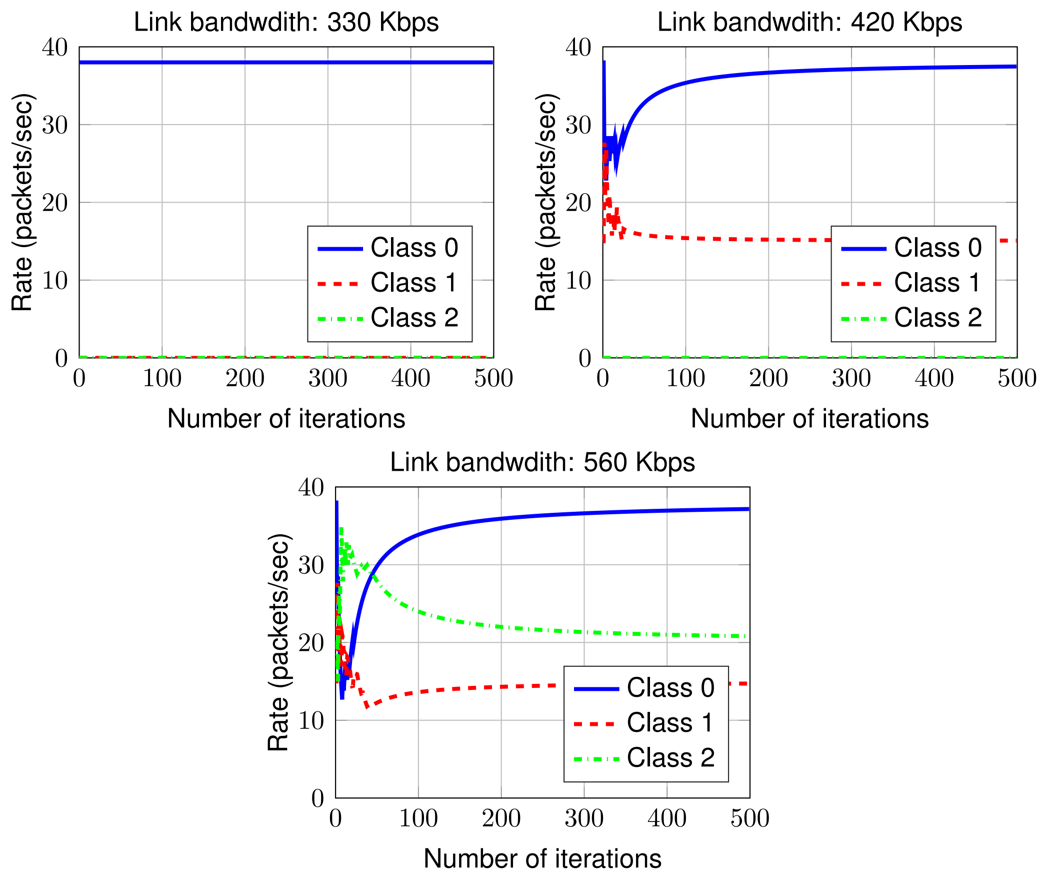


Figure 4: Cumulative rates of Interest messages sent by node 29 versus the number of iterations of the rate allocation algorithm.

5.3 Video quality evaluation

In order to validate our architecture, we have implemented a customized simulator in NS-3 [25] with the main components of the NDN architecture as well as the additional features described in Section 4, 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 in order to create some randomness in the order in which Interest messages arrive at intermediate nodes. Another source of randomness is the random selection of coding coefficients during the network coding operations.

We evaluate the proposed protocol for the delivery of scalable video in terms of quality. In Fig. 5 we present the evolution of the PSNR of the de-

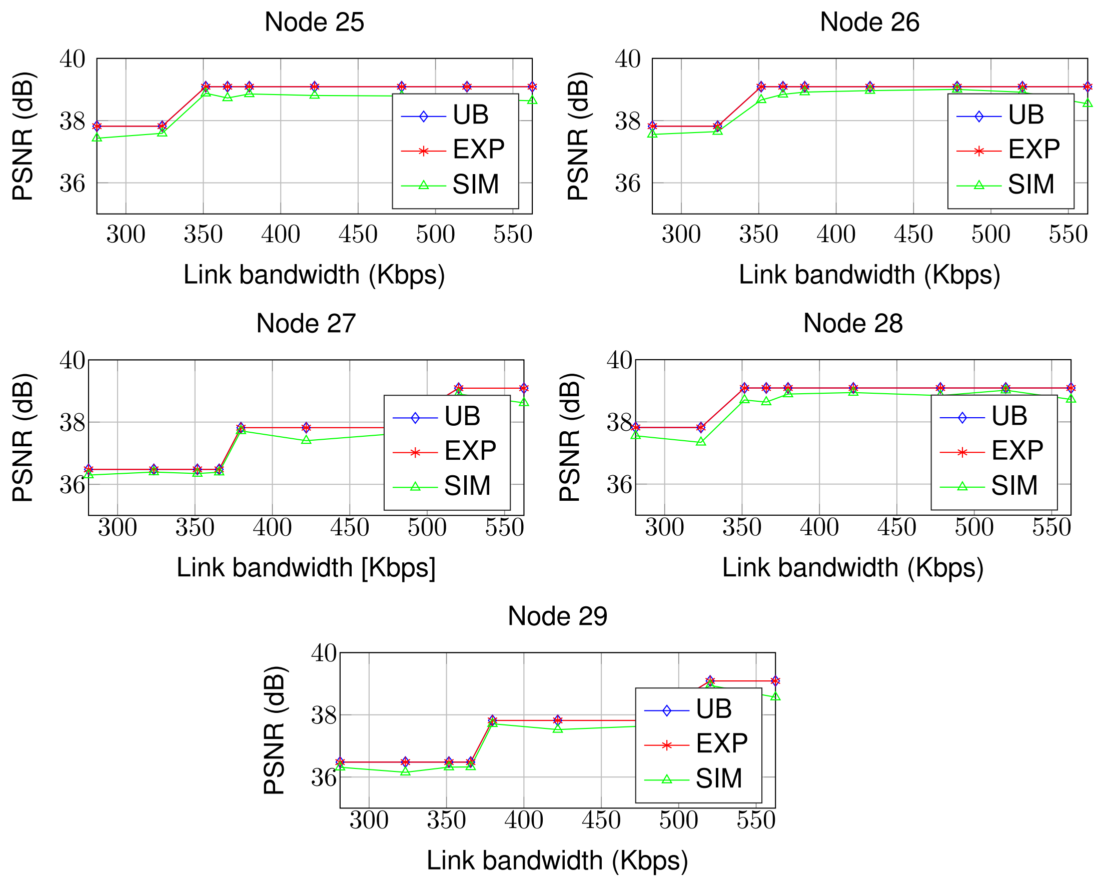


Figure 5: Average PSNR of the decoded video as a function of the links' bandwidth.

livered 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 assuming that the client is 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. Finally, SIM denotes the PSNR values obtained from the simulation of our network coding enabled NDN protocol equipped with the forwarding strategy described in Section 4.5. 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.

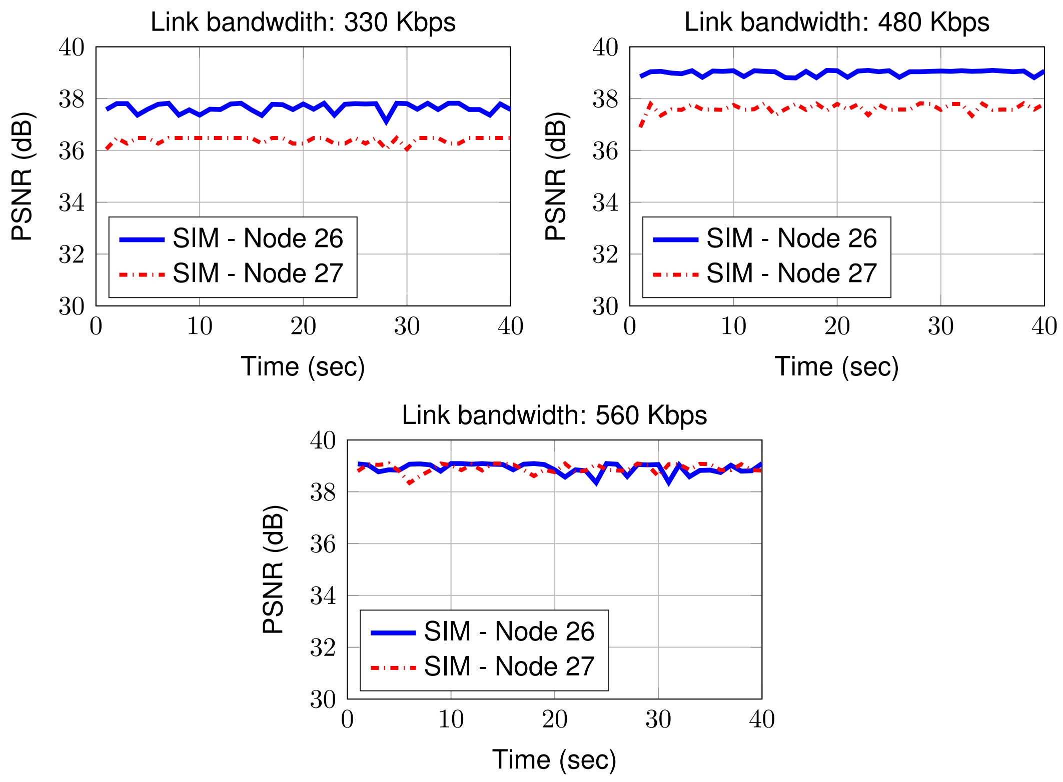


Figure 6: Average PSNR of the decoded video versus time.

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 Interest messages 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 optimal 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 network coded packets requires that all the lower packet 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.

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 network coding enabled NDN protocol 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. The results indicate 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 (leftmost figure in Fig. 6). Finally, 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.

6 Conclusions

In this work, we have presented a novel network coding enabled NDN architecture for the delivery of scalable video. In order to design the optimal content-aware forwarding strategy, we have formulated a rate allocation optimization problem which decides on the optimal rates of Interest messages sent by clients and intermediate nodes. This rate allocation ensures that the achieved flow of Data objects maximizes the average quality of the video delivered to the client population. We have also proposed the necessary modification to the standard NDN architecture in order to support the handling of Interest messages and Data objects 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 messages and Data objects. We have evaluated our protocol 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.

Our future work will include the extension of the proposed architecture to the cases where link losses affect the delivery of Interest messages and Data packets. In this case, appropriate mechanisms need to be designed in order to properly handle the re-transmission of Interest and Data packets. We will also investigate the design of distributed/heuristic algorithms for the forwarding of Interest messages that will be able to guarantee a close to optimal performance of our system without the need for central control.

References

[1] “Cisco Visual Networking Index-Forecast and Methodology, 20142019.” 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] I. Sodagar, “The MPEG-DASH Standard for Multimedia Streaming Over the Internet,” *IEEE Multimedia*, vol. 18, pp. 62–67, Apr. 2011.

[4] 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), pp. 133–144, Feb. 2011.

[5] 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, pp. 1024–1048, Apr. - June 2014.

[6] 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), pp. 1–12, Dec. 2009.

[7] 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), pp. 1–8, Dec. 2013.

[8] 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.

[9] C. Tsilopoulos and G. Xylomenos, “Supporting Diverse Traffic Types in Information Centric Networks,” in *Proc. of ACM SIGCOMM ICN Workshop*, (Toronto, ON, Canada), pp. 13–18, Aug. 2011.

[10] 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.

[11] 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.

[12] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network Information Flow,” *IEEE Trans. Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[13] N. Thomos, J. Chakareski, and P. Frossard, “Prioritized Distributed Video Delivery With Randomized Network Coding,” *IEEE Trans. Multimedia*, vol. 13, pp. 776–787, Aug. 2011.

[14] 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.

[15] 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.

[16] 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), pp. 41–42, Aug. 2013.

[17] J. Llorea, 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), pp. 3557–3562, June 2013.

[18] C. Anastasiades, N. Thomos, A. Striffeler, and T. Braun, “RC-NDN: Raptor Codes Enabled Named Data Networking,” in *Proc. of IEEE Int. Conf. on Communications, ICC’2015*, (London, UK), 2015.

[19] 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.

[20] 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), pp. 2184–2194, Mar. 2005.

[21] 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, pp. 105–113, Sep. 1996.

[22] B. H. Bloom, “Space/Time Trade-offs in Hash Coding with Allowable Errors,” *Communications of the ACM*, vol. 13, pp. 422–426, Jul. 1970.

[23] “PlanetLab.” https://www.planet-lab.org/.

[24] N. Cleju, N. Thomos, and P. Frossard, “Selection of Network Coding Nodes for Minimal Playback Delay in Streaming Overlays,” *IEEE Trans. Multimedia*, vol. 13, pp. 1103–1115, Oct. 2011.

[25] “The network simulator - ns3.” http://www.nsnam.org/.