Towards a Context-Aware Forwarding Plane in Named Data Networking Supporting QoS

Daniel Posch, Benjamin Rainer, and Hermann Hellwagner
Institute of Information Technology
Alpen-Adria-Universität Klagenfurt
{firstname.lastname}@itec.aau.at

ABSTRACT

The emergence of Information-Centric Networking (ICN) provides considerable opportunities for context-aware data distribution in the network's forwarding plane. While packet forwarding in classical IP-based networks is basically predetermined by routing, ICN foresees an adaptive forwarding plane considering the requirements of network applications. As research in this area is still at an early stage, most of the work so far focused on providing the basic functionality, rather than on considering the available context information to improve Quality of Service (QoS). This article investigates to which extent existing forwarding strategies take account of the available context information and can therefore increase service quality. The article examines a typical scenario encompassing different user applications (Voice over IP, video streaming, and classical data transfer) with varying demands (context), and evaluates how well the applications' requirements are met by the existing strategies.

CCS Concepts

•**Networks** → *Network design principles; Traffic engineering algorithms; Public Internet;* Network resources allocation; Network simulations; In-network processing;

Keywords

Information-Centric Networking; Named Data Networking; Forwarding; Context-Awareness, Quality of Service.

1 Introduction

It is evident that today's Internet has to deal with a large variety of applications. Each application requires a specific kind of service, which in general is opaque to the network layer of classical IP-based network infrastructures. On the one hand, there are applications like Voice over IP (VoIP) that demand low latency and jitter while consuming a moderate amount of bandwidth resources. On the other hand, there are applications such as video streaming that consume a large amount of bandwidth resources, but have relaxed requirements with respect to delay and jitter (due to possible pre-buffering capabilities). For an Internet Service Provider (ISP) it would be beneficial to become aware of the individual application requirements, so each transmitted packet can be delivered within the given constraints. This would lead to a high consumer satisfaction, while providing the opportunity to deliver packets in a cost-effective way (e.g., using cheap, but high-delay paths, for classical data traffic).

While the necessary context information is not easily accessible at IP's network layer, the emergence of Information-Centric Networking (ICN) is turning the tide. As surveyed in [1] there are multiple approaches to realize ICN. For this paper the understanding of ICN is coincident with the approach of Named Data Networking (NDN) [2]. In NDN, data is requested by its name following a strictly receiver-driven communication model. The name may include additional data providing information about the requested content characteristics and its application. For instance, name prefixes could be used as indicator for content/application requirements (real-time, delay-tolerant, etc.). This article does not investigate on how to represent this information, but rather focuses on how it can be used by NDN's forwarding plane to support Quality of Service (QoS) considerations.

NDN's adaptive forwarding plane is outlined in [3] and basic technical background will be sketched in Section 2. The flexibility of NDN's forwarding plane has led to a variety of strategies [4, 5, 6, 7] which have been proposed to realize adaptive forwarding. In general each of these strategies pursues a different objective (maximize throughput, minimize delay, load balancing, etc.). This raises the following research questions:

RQ1: *Are forwarding strategies in NDN able to consider different application requirements, or do they just focus on their narrow forwarding objective, oblivious to additional context information?*

RQ2: *Does context awareness in NDN's forwarding plane support the fulfillment of QoS demands?*

In order to answer these questions, we investigate and evaluate selected forwarding strategies with respect to three different types of applications: *VoIP*, *video streaming* and *classical data transfer*. Furthermore, we discuss Stochastic Adaptive Forwarding (SAF) [5], a forwarding strategy proposed by the authors that enables the consideration of extensive context information in the forwarding plane. We present how this is possible and show the benefits of this approach by conducting network simulations using the network simulation framework ns-3/ndnSIM v2.0 [8]. Therefore, we define an evaluation scenario encompassing the aforementioned user applications. We measure the relevant QoS

ACM ISBN (PREPRINT Version — to appear in Jan. 2017).
DOI: http://dx.doi.org/00.0000/0000000.0000000

parameters for each application type and use them as input for existing models to obtain the actual user satisfaction. The detailed investigation of the presented scenario will provide substantial insight into how the individual forwarding strategies and their context awareness influences the user satisfaction with respect to a concrete application. The results indicate that context awareness in the forwarding plane is relevant and can lead to significant QoS improvements.

2 The NDN Forwarding Plane: An Overview

This section introduces the basic technical background of forwarding in NDN. Furthermore, it presents existing strategies and discusses their principles with a focus on context awareness. All discussed strategies are available for the Networking Forwarding Daemon (NFD) [4]. The NFD is a piece of software maintained by the NDN community and provides NDN-based communication over physical networks and in simulated environments using ns-3/ndnSIM [8].

2.1 Packet Forwarding in NDN

In NDN, consumers retrieve content by emitting so called *Interest* messages. These messages are forwarded (to and by other nodes) until they reach a node that can provide the desired content objects. An Interest's final destination could be either the content origin or an intermediate node that holds a cached content replica. The corresponding content object can be generated dynamically as response to an issuing Interest, or it may already exist encapsulated in so-called *Data* packets. An Interest matches a Data packet if its name is a prefix of the Data's name. Once an Interest encounters a match, the corresponding Data packet is returned to the consumer on the reverse path of the issuing Interest.

To provide packet forwarding, a typical NDN node maintains three data structures [2]: *i*) *Content Store* (CS), a cache providing data replicas; *ii*) *Pending Interest Table* (PIT), a table keeping track of the forwarded (still pending) Interests providing the return path for Data packets; and *iii*) *Forwarding Information Base* (FIB), a table essentially maintaining routing information. In contrast to IP, NDN foresees an adaptive forwarding plane with inherent multi-path delivery [3]. A node may register multiple outgoing (inter-)faces per name prefix and the forwarding strategy is responsible to select the "best" outgoing face(s) for each individual Interest that traverses the current node. To obtain a good decision it is necessary to consider a certain amount of context information in the forwarding plane. In the following subsection we provide a basic definition of context and context awareness in the forwarding plane.

2.2 Context Awareness — A View From the Forwarding's Perspective

From the view of NDN's forwarding plane, context awareness could provide the key to effective Interest forwarding. The more information is considered when forwarding Interests, the better the deduced decision will be. It is hard to provide a sharp and unique definition of what is the relevant context information for forwarding as this may vary strongly for different scenarios. Basically, for all scenarios relevant context includes the individual face performances with respect to delay, capacity and packet loss. However, for more sophisticated scenarios, content specific information can be recognized as relevant content information in the forwarding plane. For instance, the name prefix */voip* of a packet may indicate that this is a delay-intolerant transmission, while other name prefixes may indicate the opposite. Even more detailed information concerning the content can be considered as relevant context. Examples are the content's popularity, its availability in nearby caches, and also the availability of various representations (encodings). Although there is the possibility to exploit this rich amount of context information, the majority of the existing baseline strategies focus solely on the classical face performance metrics. This lack in context awareness potentially leads to non-optimal forwarding decisions. In the following, we discuss the most prominent available strategies for NDN and sketch their basis for decision making on what is/are the "best" face(s).

2.3 Strategies: How Context-Aware Are They?

Broadcast [4] is a simple strategy that does not consider any context except the information that is provided by the FIB. Interests are forwarded on all outgoing faces that are registered for the given name prefix. It is evident that this strategy causes a lot of unnecessary overhead that increases with the number of available nodes/links leading to bad performance in scenarios with scarce resources.

BestRoute [4] forwards Interests to the lowest-cost (e.g., in terms of hop count) upstream face. This context information has to be provided by a third component. For instance, this could be the routing plane that provides the number of hops (distance) per outgoing face to the content origin. This setting is the default configuration as implemented in the NFD [4]. Another configuration could consider the imposed latency, or also a combination of both.

NCC [4] forwards Interests to those faces that provide the lowest delay for receiving data packets. Instead of BestRoute that depends on context information provided by a third component, NCC gathers and maintains the latency statistics on its own. Therefore, it measures for each Interest the time it takes to satisfy it on the outgoing face and continuously updates the selected face's assessment. NCC is not an acronym. Its name was derived from flipping the initials from the term Content-Centric Networking (CCN) [9]. NCC was the default forwarding strategy implemented in PARC's CCNx (v0.7.2) and has been ported to the NFD.

The *Request Forwarding Algorithm (RFA)* [6] is part of a set of optimal dynamic multi-path congestion control protocols and request forwarding strategies derived from multi-commodity flow problems. This algorithm monitors for each available prefix the number of PIT entries. Using this information, the forwarding probability of a face is determined by a weight that is actually a moving average over the reciprocal count of the PIT entries. Simply put, the relevant context for RFA is the current load of a face as indicated by the PIT.

On-demand Multi-Path Interest Forwarding (OMP-IF) [7] suggests the forwarding of Interests on node-disjoint paths. In the proposed approach each network node may only use a single face (from the FIB) for forwarding per name prefix to ensure node disjointness. The consumer nodes trigger the multi-path transmission by utilizing a weighted round-robin mechanism based on the path delays, distributing Interests over multiple faces. If a router encounters packet loss on the selected face, subsequent Interests of the corresponding name-prefix are broadcasted. The first face satisfying a broadcasted Interest is selected for further transmission.

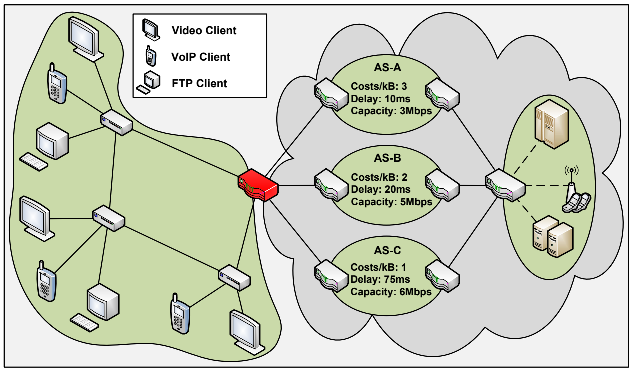


Figure 1: The evaluation scenario considers three different applications: video streaming, VoIP, and data transfer.

Stochastic Adaptive Forwarding (SAF) [5] imitates a self-adjusting water pipe system, intelligently guiding and distributing Interests through the network. SAF uses the returning Data packets as input for deriving a probability distribution, determining forwarding probabilities for the individual face per name prefix. Additionally, SAF employs a virtual (dropping) face that is used to encounter congestion by actively dropping Interests that could cause congestion. The aforementioned probabilities are stored in the so-called *Forwarding Table* (FWT), which are modified by an adaptation engine. SAF is the first strategy that enables an operator to incorporate specific context-aware considerations. Operators may define per name prefix the conditions for considering an Interest as satisfied (e.g., returning Data packets must be retrieved in less than d milliseconds). Furthermore, SAF enables operators to set different weights per name prefix considering their relative importance. The adaptation takes account of the employed context information and arranges the forwarding probabilities in the table accordingly. We present a possible extension to SAF in Section 3.3 that considers context information of the presented scenario (cf. Section 3.1).

3 Investigating NDN's Context-Aware Forwarding Plane: Does it Enhance QoS?

This section investigates the opportunities of introducing context awareness in the forwarding plane to enhance QoS. As a first step, an exemplary scenario is sketched that encompasses network applications with different requirements. Then the employed evaluation methods are discussed. Finally, the results are presented that have been obtained by conducting network simulations using ns-3/ndnSIM [8].

3.1 Scenario Description

Figure 1 depicts the proposed scenario. The left-hand side of the figure illustrates an autonomous system (AS) consisting of several routers and clients representing a typical ISP access network. The clients employ three applications: video streaming, VoIP, and classical file transfer. The relevant files or communication partners for the clients are located in another AS that is depicted on the right hand side of Figure 1. As there is no direct connection between those two autonomous systems, traffic has to be routed/forwarded through the networks of other autonomous systems. We assume that the ISP access network maintains service level agreements (SLAs) with three other (intermediate) providers (AS-A, AS-B, AS-C) that may bridge the gap between the networks. AS-A provides a capacity of 3 Mbps for traffic, imposes a one-way delay of 10 ms and charges 3 units per transmitted kilobyte. The respective values for AS-B are 5 Mbps, 20 ms delay, 2 cost units, and for AS-C 6 Mbps, 75 ms delay, 1 cost unit. In the given scenario all remaining links provide sufficient resources and low latency having negligible negative impact on data transmission. Therefore, the investigated strategies are only changed on the router highlighted by the red color in Figure 1 that connects the access network with the autonomous systems AS-A, AS-B and AS-C. All other routers use the BestRoute strategy.

The video streaming application on the clients is implemented by the principles of Dynamic Adaptive Streaming [10]. Clients use a buffer-based adaptation logic [11] to request video content that is taken from the SVC-DASH dataset [12]. In [12] the video content is encoded in various variants. A variant defines the encoding parameters as well as the scalability domains (temporal, spatial, quality). For this evaluation we have chosen the variant providing Signal-to-Noise-Ratio (quality) scalability only with a segment duration of 2 seconds. The chosen content is provided using a base layer and two enhancement layers. The base layer (henceforth denoted as $L0$) has an average bitrate of approximately 640 kbps. The first enhancement layer ($L1$) has a bitrate of approximately 355 kbps. In order to play back a segment at the quality of $L1$, one has to fetch the same segment of $L0$ yielding a combined multimedia bitrate of $L0 + L1 \approx 995$ kbps. The second enhancement layer ($L2$) has an average bitrate of approximately 407 kbps ($L0 + L1 + L2 \approx 1400$ kbps). The request pattern of the video streaming clients is bursty, thus challenging the individual forwarding strategies.

While IP-based VoIP clients are implemented in a push-based fashion, this is not possible in NDN. NDN follows a strictly pull-based communication approach. A Data packet can only be delivered in response to an Interest. However, this would roughly double the typical one-way delay imposed by push-based approaches impairing user Quality of Experience (QoE) significantly. Pioneering work by Jacobson et al. [13] exploits NDN's hierarchical name space to request data that does not yet exist. This resolves the problem by transmitting an Interest to the producer before the corresponding VoIP data packet is created. The Interest keeps pending until the data is generated and can then immediately be delivered to the requesting client application. Note that here the client application needs to know the packet generation rate, which can be easily predicted from the employed audio codec (e.g, G.711 [14]). We implement the VoIP client following these principles and choose G.711 with Packet Loss Concealment (PLC) as audio codec. Furthermore, we assume a fixed jitter buffer of 50 ms and employ a fixed audio codec bitrate of 64 kbps, which is a typical setting for G.711. The request pattern of the VoIP clients is steady with low demands on capacity, however, it demands that the forwarding strategies choose low latency links. For instance, routing VoIP traffic through AS-C will lead to late packets having a negative impact on user satisfaction.

The File Transfer Protocol (FTP) client is modeled by a simple consumer/producer application provided by [8]. We assume that each FTP client requests a large file from a server demanding roughly 3 Mbps of bandwidth capacity. The traffic pattern of this application is steady, with very low demands only focusing on throughput.

As this paper focuses on the capabilities of context-aware forwarding, we assume that each of the clients depicted in Figure 1 requests unique content. Therefore, caching can be disabled eliminating possible side effects on the data transmission performance. This allows to perfectly investigate the pure forwarding capabilities of the investigated strategies.

3.2 Evaluation Method

To evaluate the performance of the different forwarding strategies and their context-aware capabilities, we investigate each of the applications using a separate evaluation method:

Simplified E-Model for VoIP

For the evaluation of the VoIP performance we use a simplified version of the E-Model [15] that is applicable when only packet loss and delay impairments are considered. This model provides the so called R-value that can be mapped to a Mean Opinion Score (MOS) using Eq. 1. The R-value is basically calculated as $R = 93.2 - I_d - I_{e-eff}$, where I_d represents the impairments caused by the one way delay d, (cf. Eq. 2). I_{e-eff} is defined by Eq. 3 considering the codec impairments (I_e) and the influence of the packet loss percentage (P_{pl}) and its burstiness ($BurstR$) on the employed codec. As in our scenarios clients use G.711 with PLC $I_e = 0$, and $B_{pl} = 34$, which denotes the codecs built-in packet loss concealment ability. For more details on the E-Model and the selected parameters we refer the interested reader to ITU recommendations G.107, G.711, G.113 and to [15].

$$MOS = \begin{cases} 1 & \text{if } R \leq 0, \\ 1 + 0.035R + R \cdot \\ (R-60)(100-R) \cdot 7 \cdot 10^{-6} & \text{if } 0 < R < 100, \\ 4.5 & \text{if } R > 100. \end{cases} \quad (1)$$

$$I_d = \begin{cases} 0.024d & \text{if } d < 177.3, \\ 0.024d + 0.11(d - 177.3) & \text{if } d \geq 177.3. \end{cases} \quad (2)$$

$$I_{e-eff} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{\frac{P_{pl}}{BurstR} + B_{pl}} \quad (3)$$

User Satisfaction Model for Video Clients

To evaluate the performance of the video streaming applications, we use the proposed user satisfaction model from [16]. The model considers the video quality, the quality variations and the re-buffering events:

$$Q = \underbrace{\sum_{k=1}^{K} q(R_k)}_{\text{video quality}} - \lambda \underbrace{\sum_{k=1}^{K-1} |q(R_{k+1}) - q(R_k)|}_{\text{quality variations}} - \mu \underbrace{\sum_{k=1}^{K} b(R_k)}_{\text{re-buffering time}} \quad (4)$$

K denotes the number of segments received by a client. R denotes the available representations (in our case $R = \{L0, L0 + L1, L0 + L1 + L2\}$) and R_k denotes the consumed representation of segment k on a client. $q(R_k)$ denotes the quality of a segment, which we define as the average corresponding representation bitrate. $b(R_K)$ denotes the number of re-buffering seconds before segment R_k is ready for playout. λ and μ are non-negative parameters modeling the particular influence of quality variations and re-buffering events, respectively. In [16] different combination of values for λ and μ are suggested. We vary the parameters in the suggested ranges and provide a 3D plot indicating the user satisfaction considering various user preferences. Please note that we re-formulated the model slightly since in [16] it is used as an objective function for a maximization problem imposing a more complex formulation.

Download Bitrate for FTP Clients

The performance of the FTP clients is measured by the pure download bitrate. We consider the goodput (throughput minus overhead) as the relevant performance indicator.

3.3 Adding Context Information to SAF

As already mentioned, SAF's design [5] allows considering additional context information. In the following we show how to introduce the scenario's context into SAF. We *consider* the following facts as relevant *context information*: VoIP clients cause the lowest amount of traffic, however, their users suffer heavily from packet loss and late packets (cf. Equations. 1, 2, and 3). Therefore, VoIP traffic should be prioritized at the forwarding plane, especially on low latency links. Let us assume that we order the importance of the content classes as follows: $VoIP >_c video >_c data$, where $a >_c b$ denotes that content a should be prioritized over content b. Then our objective is to introduce a weighting mechanism that ensures this ordering by influencing the performance assessment of faces. The weights shall be selected in such a way that low priority content will be dropped in favour of high priority content due to SAF internals (using the dropping face F_D). In the following we show how this demand can be introduced and realized by SAF. Recall that SAF [5] periodically performs updates of the forwarding probabilities for each registered name prefix considering all faces $F_i \in \mathcal{F}$. The updates maximize a combined measure $\mathcal{M}$ that is given by Equation 5. $\mathcal{I}_n$ denotes the set of Interests in a given period n. $S_{F_i}(\mathcal{I}_n)$ denotes a measure for the satisfied Interests, $U_{F_i}(\mathcal{I}_n)$ a measure for the unsatisfied Interests. SAF's default configuration defines $S_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is satisfied by a Data packet on } F_i\}|$ and $U_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is not satisfied on } F_i\}| \; \forall F_i \in \mathcal{F} \setminus \{F_D\}$, where F_D denotes the virtual dropping face that satisfies Interests by definition [5]. Thus, SAF maximizes the throughput for the individual name prefixes.

$$\mathcal{M} = \sum_{F_i \in \mathcal{F}} M_{F_i}(\mathcal{I}_n) = \sum_{F_i \in \mathcal{F}} \left(S_{F_i}(\mathcal{I}_n) - U_{F_i}(\mathcal{I}_n)\right) \quad (5)$$

In order to consider the relative content priorities, we are going to re-define the measure $U_{F_i}(\mathcal{I}_n)$. This leads us to SAF-CAA (Contex-Aware Adaptation) which introduces an additional weight ω to the definition of U_{F_i}. We re-define $U'_{F_i}(\mathcal{I}_n) := U_{F_i}(\mathcal{I}_n) \cdot \omega_{F_i} := \omega_{F_i} \cdot |\{j \in \mathcal{I}_n : j \text{ is not satisfied by a Data packet on } F_i\}|$. As ω_{F_i} is chosen differently for each content (and also specifically per face as indicated by the subscript F_i, which will be disscues later in more detail) it can be used to realize the prioritization. In general, if network congestion is encountered, a large ω leads to earlier pro-active packet dropping of the corresponding content. In the following we provide a rationale for the selection of the individual weights based on the definition of a *reliable* face [5].

According to SAF [5], a reliable face is defined as given by Definition 3.1. The dynamic threshold t_{c_j} tells us how much reliable traffic a specific content currently has, assuming SAF has already converged. We want to introduce an ordering on the specific contents such that we can influence the reliability accordingly. Therefore, we use the weight ω_{F_i,c_j} for the calculation of the reliability of face F_i for a given content c_j as follows: $t_{c_j} \leq \frac{S_{F_i,c_j}}{S_{F_i,c_j}+U_{F_i,c_j}\cdot\omega_{F_i,c_j}}$.

Definition 3.1. A face $F_i \in \mathcal{F}$ is *reliable* for content $c_j \in \mathcal{C}$ if and only if $R_{F_i,c_j} := \frac{S_{F_i,c_j}}{S_{F_i,c_j}+U_{F_i,c_j}} \geq t_{c_j}$, where t_{c_j} denotes the reliability threshold of SAF [5] for c_j.

Definition 3.2 defines an *adaptable set* of contents with respect to a given face $F_i \in \mathcal{F} \setminus \{F_D\}$. Here, the term *adaptable* indicates that for these contents/prefixes adaptation among the individual columns of SAF's FWT is reasonable. Colloquially, we define a set of contents as *adaptable* with respect to F_i if that face performs reliable data delivery for every content in that set. Furthermore, a secondary condition must hold that says that there must exist unsatisfied traffic for each content in that set. In the following we outline why contents in an adaptable set have to satisfy both requirements so that effective adaptation can be performed.

Definition 3.2. We define a set of contents $\mathcal{C}_{F_i}$ as *adaptable* with respect to a given face F_i if and only if for a given content catalogue $\mathcal{C}$, $\mathcal{C}_{F_i}$ contains only contents that are considered as reliably transmitted on F_i, although a number (greater than 0) of Interests cannot be satisfied by F_i, $\mathcal{C}_{F_i} := \{c_j \in \mathcal{C} | R_{F_i,c_j} \geq t_{c_j} \wedge U_{F_i,c_j} > 0 \wedge S_{F_i,c_j} > 0\}$ $\forall F_i \in \mathcal{F} \setminus \{F_D\}$.

The reason for solely encapsulating contents that are reliably transmitted via F_i in an adaptable set is apparent when considering the following. SAF has learned the optimal amount of traffic for each content in the adaptable set that should be forwarded via F_i considering the individual content's reliability threshold t_{c_j}. Only for these contents SAF has converged [5]. Therefore, adaptation among the individual contents in the adaptable set is possible. Contents that are not part of an adaptable set are not transmitted reliably on F_i. Thus, SAF is in an unstable state concerning these contents making meaningful adaptation decisions impossible. The secondary condition ensures that only contents are considered that have unsatisfied Interests on F_i. Therefore, only contents interfering with each other on the given face F_i are considered for adaptation. Given an adaptable set and an ordering of contents, we can state Theorem 3.1 providing a rationale for selecting the individual weights ω.

Theorem 3.1. Given an *adaptable set* $\mathcal{C}_{F_i}$ with $|\mathcal{C}_{F_i}| > 1$ and an ordering on the contents $(\mathcal{C}, >_c)$ (denoting the importance of the contents), one obtains the following result for determining the weights such that the ordering of the contents is established by SAF on F_i:

$$\forall c_k, c_j, c_m \in \mathcal{C}_{F_i}, c_k >_c c_j >_c c_m : \qquad (6)$$
$$\omega_{F_i,c_j} \in \left]\, (*), \frac{S_{F_i,c_j}\cdot U_{F_i,c_m}\cdot\omega_{F_i,c_m}}{S_{F_i,c_m}\cdot U_{F_i,c_j}} \right[,$$

where $(*)$ is $\max\left\{\frac{S_{F_i,c_j}\cdot U_{F_i,c_k}\cdot\omega_{F_i,c_k}}{S_{F_i,c_k}\cdot U_{F_i,c_j}}, \frac{S_{F_i,c_j}\cdot(1-t_{c_j})}{U_{F_i,c_j}\cdot t_{c_j}}\right\}$. We further have two degrees of freedom, ω_{F_i,c_1} and $\omega_{F_i,c_{|C_{F_i}|}}$.

[Proof of Theorem 3.1] The interested reader is referred to the appendix of this article. ■

Algorithm 1 outlines an algorithmic procedure to obtain the weights ω_{F_i,c_j}. The algorithm shall be executed after each iteration of SAF's FWT updates assuming that the contents are ordered by importance (descending) in $\mathcal{C}_{F_i}$.

Algorithm 1 Context-Aware Adaptation for SAF

1: **for each** $F_i \in \mathcal{F} \setminus \{F_D\}$ **do**
2: $\quad w^{(L)}_{F_i,1} \leftarrow 1$
3: $\quad$ **for** $1 \leq j \leq |\mathcal{C}_{F_i}| - 1$ **do**
4: $\quad\quad w^{(L)}_{F_i,c_{j+1}} \leftarrow \max\left\{\frac{U_{F_i,c_j}\cdot w^{(L)}_{F_i,c_j}}{S_{F_i,c_j}\cdot U_{F_i,c_{j+1}}}, \frac{(1-t_{c_{j+1}})}{U_{F_i,c_{j+1}}\cdot t_{c_{j+1}}}\right\}$
5: $\quad\quad w^{(L)}_{F_i,c_{j+1}} \leftarrow S_{F_i,c_{j+1}} \cdot w^{(L)}_{F_i,c_{j+1}}$
6: $\quad w^{(U)}_{F_i,c_{|\mathcal{C}_{F_i}|}} \leftarrow w^{(L)}_{F_i,c_{|\mathcal{C}_{F_i}|}} + 1$
7: $\quad$ **for** $|\mathcal{C}_{F_i}| \geq j \geq 2$ **do**
8: $\quad\quad w^{(U)}_{F_i,c_{j-1}} \leftarrow \frac{S_{F_i,c_{j-1}}\cdot U_{F_i,c_j}\cdot w^{(U)}_{F_i,c_j}}{S_{F_i,c_j}\cdot U_{F_i,c_{j-1}}}$
9: $\quad \omega_{F_i,1} \leftarrow w^{(L)}_{F_i,1}$
10: $\quad \omega_{F_i,|\mathcal{C}_{F_i}|} \leftarrow w^{(U)}_{F_i,c_{|\mathcal{C}_{F_i}|}}$
11: $\quad$ **for** $2 \leq j < |\mathcal{C}_{F_i}|$ **do**
12: $\quad\quad \omega_{F_i,c_j} \leftarrow \frac{w^{(U)}_{F_i,c_j}+w^{(L)}_{F_i,c_j}}{2}$
13: $\quad$ **for each** $c_j \in \mathcal{C} \setminus \{\mathcal{C}_{F_i}\}$ **do**
14: $\quad\quad \omega_{F_i,c_j} \leftarrow 1$

Recall that the following operations are executed for every face $F_i \in \mathcal{F} \setminus \{F_D\}$ (cf. Alg 1, line 1). First, we initialize the lower bound for the content with the highest priority with 1 (cf. Alg. 1, line 2). This, will also be the first degree of freedom deduced in Theorem 3.1 (9). Then, the remaining lower bounds for the weights are determined, which are denoted as $w^{(L)}$ (cf. Alg. 1, lines 3-5). Subsequent, we set the second degree of freedom, which is the upper bound for the content with the lowest priority (cf Alg. 1, line 6 and line 10). Then, the remaining upper bounds are calculated, which are denoted as $w^{(U)}$ (cf. Alg. 1, lines 7-8). Now, the weights ω are chosen. Here we take a value in the middle of the open interval $(w^{(L)}_{F_i,c_j}, w^{(U)}_{F_i,c_j})$ (cf. Alg. 1, lines 11-12). Finally, the algorithm resets the weights for all contents that do not satisfy the conditions for an adaptable set (cf. Alg 1, lines 13-14). We have implemented Algorithm 1 for SAF's adaptation engine as presented. The source code can be found at http://icn.itec.aau.at.

3.4 Results

Figures 2 and 3 summarize the results for the individual applications considering all forwarding strategies discussed in Section 2. Figure 2 depicts the user satisfaction of the video streaming users obtained by Eq. 4 and normalized by the highest possible score. Figure 3a depictes the MOS value for the VoIP clients. Figure 3b depicts the achieved download rate of the FTP clients. Figure 3c depicts the total traffic (incoming and outgoing) triggered by the ISP access network illustrated on the left-hand side of Figure 1. Figure 3d provides the average costs per transmitted kilobyte.

Broadcast: It is evident from Figure 3 that Broadcast performs worst. This can be explained by the strategy's nature to excessively replicate Interests and by the rather limited available resources. Broadcast introduces a lot of unnecessary overhead. Although the strategy causes with 5 GB the highest amount of traffic (and therefore also high costs for the provider, Figure 3c) the consumer/applications demands are not fulfilled. The congestion caused by Interest replication leads to a high packet loss rate and heavily delays packet delivery. This results in the worst possible MOS for the VoIP clients (Figure 3a) and also in the worst achieved

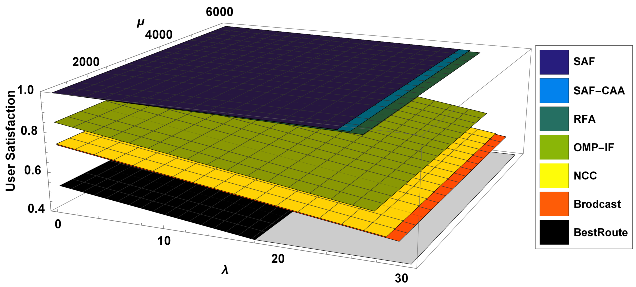


Figure 2: Video streaming satisfaction considering different preferences (λ denotes a weight for a user's sensitivity to quality variations, μ denotes a weight for a user's sensitivity to re-buffering time) [16].

goodput (1672 kbps) for the FTP clients (Figure 3b). Nevertheless, the user satisfaction of the video streaming clients is at an acceptable level, even higher than for BestRoute.

NCC: The overall performance of NCC is actually similar to Broadcast. The only major distinction is the performance of the FTP clients, which are able to achieve a 460 kbps higher goodput. The bad performance of NCC can be explained by the following facts. NCC focuses only on the link that provides the lowest data delivery delay. Therefore, the network traffic is preferably routed through AS-A. The resources of AS-A are overburdened and will cause packet loss due to congestion. Internally NCC issues retransmission for late or lost packets using alternative paths. This excessive retransmission strategy introduces an amount of overhead similar to Broadcast (cf. Figure 3c), resulting in the worst possible MOS of 1 for the VoIP clients.

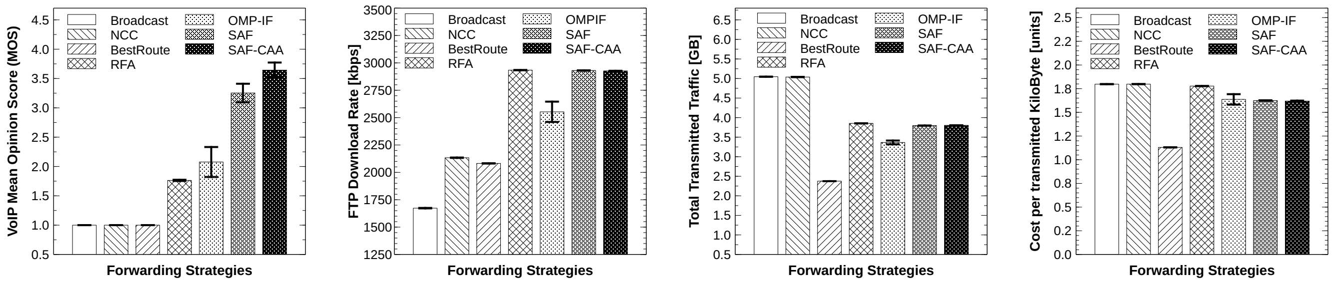
BestRoute: This strategy, configured to consider the ISP's costs as the relevant metric, is no great improvement to Broadcast or NCC. Instead of NCC that focuses on the lowest-delay connection through AS-A, BestRoute focuses on the lowest-cost connection through AS-C (cf. Figure 1). Only if a consumer application issues a retransmission due to late or lost packets, BestRoute considers the next "best" (cheapest) route for data transmission. The low transmitted amount of traffic presented in Figure 3c indicates that BestRoute is unable to effectively use multiple paths for delivery leading to the lowest user satisfaction for the video streaming clients. Furthermore, BestRoute is unable to deliver a single VoIP packet in time due to its strict focus on the "cheapest" route, which is also reflected by the worst possible MOS of 1.

RFA: This is the first algorithm that achieves a MOS greater than 1 for the VoIP clients. As previously mentioned, RFA basically performs a kind of load balancing by considering the number of pending Interests per face. Considering this scenario, the strategy performs fine, in particular the FTP clients are able to achieve their target goodput of about 3000 kbps. Also the user satisfaction for the video streaming clients is acceptable (cf. Figure 4). Nonetheless, a better performance for the VoIP clients is desirable.

OMP-IF: This scheme is able to obtain a MOS of about 2 for the VoIP clients, which is significantly better than RFA. Nevertheless, OMP-IF is inferior with respect to the data transfer and video streaming performance, for the following reasons. While RFA is focused on load balancing and therefore tries to maximize the throughput, OMP-IF considers only node-disjoint paths for individual name prefixes. As in this scenario only three different prefixes are employed (*/voip*, */data*, */video*) OMP-IF is capable of separating the individual streams. However, taking the burstiness of the video traffic into account, none of the autonomous systems AS-A, AS-B and AS-C can fulfill the traffic demands during the bursts. These bursts lead to packet loss or late packets, which will be countered by OMP-IF by frequent path switching. The frequent path switching leads to a lower performance with respect to the video and transfer applications, though, it gives the VoIP clients the opportunity to obtain a better service than in the case of RFA.

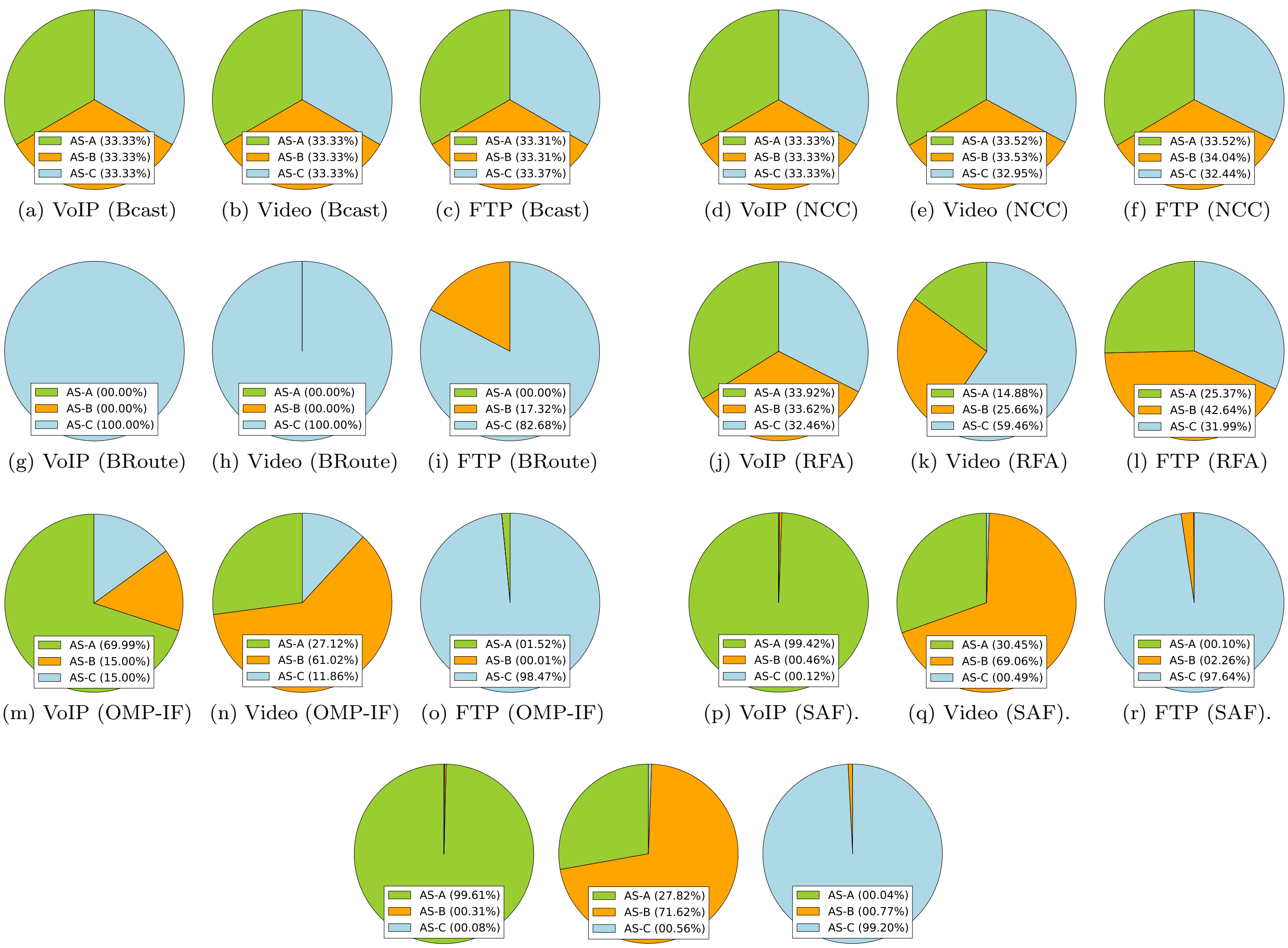
SAF(-CAA): It is evident from Figures 2 and 3 that SAF outperforms all other algorithms. It reaches a MOS of more than 3, while maintaining the target goodput of the data streaming clients and delivering excellent service to the video streaming clients. The key to success for SAF is the consideration of context that allows to maintain this high QoS and QoE levels. As previously mentioned, SAF evaluates for each name prefix and for each face the Interest satisfaction ratio. Considering this ratio, it derives the optimal strategy. For instance, it is able to deliver the VoIP traffic successfully since it implicitly considers the lifetime flag that is provided by an Interest packet. If an Interest times out in the forwarding plane due to a lifetime expiry, SAF uses this information to deduce better decisions in the future. As introduced previously, SAF-CAA considers VoIP traffic as more important than video or data traffic, which are in general more resilient to low values of packet loss. As can be seen from Figure 3, introducing this context into SAF further increases the MOS value for the VoIP clients by about 0.4 without leading to largely negative effects for the data and video streaming applications.

To further investigate the individual behavior of the algorithms in the presented scenario (cf. Figure 1), Figure 4 depicts the individual shares of Interests that are forwarded for each content/application per autonomous system. The figure shows the result from an exemplarily chosen simulation run. Each triple of pie charts presents the Interest shares for the applications VoIP (1st chart), video (2nd chart), and FTP (3rd chart) with respect to the investigated forwarding strategies which may forward Interests via AS-A (green), AS-B (orange), and AS-C (light blue). Figure 4 depicts that Broadcast and NCC are oblivious to the individual application demands, simply transmitting equally sized shares on all available autonomous systems. Although both strategies forward approximately two-thirds of the VoIP Interests via AS-A and AS-B, they are not able to achieve a MOS greater than 1. In contrast, RFA achieves a MOS of about 1.5 (cf. Figure 3a) showing a similar traffic share pattern. Although RFA forwards one-third of the VoIP traffic via AS-C (that is not able to deliver VoIP packets in time), it is able to outperform Broadcast and NCC. The reason for this is that RFA causes less congestion because of: *i*) not excessively replicating Interests (cf. Figure 3c); and *ii*) by effective load balancing of traffic considering the number of PIT entries as an indicator for face utilization (faces with many PIT entries are avoided). BestRoute basically considers only AS-C for forwarding, because it has been configured to prefer the lowest-cost path. Only for lost and/or retransmitted Interests an alternative path via AS-B is used. Therefore, this strategy fails in satisfying the demands of the VoIP clients and also can not take advantage of NDN's multi-path capabilities. In contrast, OMP-IF does a much better job. OMP-IF only forwards 15% of the VoIP Interests via AS-C,



(a) MOS for VoIP clients. (b) FTP download bitrates. (c) Total transmitted traffic. (d) Avg. costs per kilobyte.

Figure 3: Results considering the individual applications' performance with respect to the discussed forwarding strategies.



(a) VoIP (Bcast) (b) Video (Bcast) (c) FTP (Bcast) (d) VoIP (NCC) (e) Video (NCC) (f) FTP (NCC)

(g) VoIP (BRoute) (h) Video (BRoute) (i) FTP (BRoute) (j) VoIP (RFA) (k) Video (RFA) (l) FTP (RFA)

(m) VoIP (OMP-IF) (n) Video (OMP-IF) (o) FTP (OMP-IF) (p) VoIP (SAF). (q) Video (SAF). (r) FTP (SAF).

(s) VoIP SAF-CAA (t) Video SAF-CAA (u) FTP SAF-CAA

Figure 4: Share of Interests forwarded via the available autonomous systems per content/application (cf. Figure 1).

and therefore, obtains a MOS of about 2. We can also see that OMP-IF forwards the largest part of the data traffic to AS-C. However, when compared to SAF(-CAA), we observe that better results are possible. SAF performs excellently, forwarding more than 99% of the latency sensitive VoIP traffic via AS-A and forwarding more than 99% of the latency tolerant FTP traffic via the lowest-cost path (AS-C). SAF forwards the biggest share of the video traffic via AS-B, using resources of AS-A that are not required by the VoIP traffic. The results show that SAF optimally separates and distributes the Interests for the individual applications on the available autonomous systems. Furthermore, we can observe that the weighting of contents has a significant positive effect on the MOS (cf. Figure 3a), and it has basically no influence on the traffic shares. The weighting only ensures that data and video packets are dropped earlier than VoIP packets.

4 Conclusion and Outlook

In this paper we investigated the importance of considering context information in NDN's forwarding plane. We specified a scenario encompassing different network applications (VoIP, video streaming, data transfer) with various demands and evaluated their performance using prominent forwarding strategies. We conducted network simulations using ns-3/ndnSIM [8] and obtained the relevant QoS parameters. We mapped these values to user satisfaction models to asses their actual benefit. The results indicate that the more context information is considered by the forwarding strategies, the better becomes the provided QoS, leading to higher user satisfaction. Especially the strategy Stochastic Adaptive Forwarding, which can be easily configured to consider context information, performs excellently in the presented scenario. Our results indicate that further research in this area should focus on the available context information to unlock the full capabilities of NDN's forwarding plane.

5 References

[1] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A Survey of Information-Centric Networking Research," *IEEE Comm. Surveys and Tutorials*, vol. 16, no. 2, 2014.

[2] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM SIGCOMM Computer Comm. Review*, vol. 44, no. 3, pp. 66–73, 2014.

[3] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A Case for Stateful Forwarding Plane," *Computer Communications*, vol. 36, no. 7, pp. 779 – 791, 2013.

[4] A. Afanasyev, J. Shi, B. Zhang, L. Zhang *et al.*, "NFD Developer's Guide," Technical Report NDN-0021, 2014.

[5] D. Posch, B. Rainer, and H. Hellwagner, "SAF: Stochastic Adaptive Forwarding in Named Data Networking," *IEEE/ACM Transactions on Networking*, vol. (to appear), p. 14, 2016, http://dx.doi.org/10.1109/TNET.2016.2614710.

[6] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang, "Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks," in *Proc. of 21st IEEE Int. Conf. on Network Protocols*, 2013.

[7] A. Udugama, X. Zhang, K. Kuladinithi, and C. Goerg, "An On-demand Multi-Path Interest Forwarding Strategy for Content Retrievals in CCN," in *IEEE NOMS*, 2014.

[8] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A New Version of the NDN Simulator for NS-3," Technical Report NDN-0028, 2015.

[9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proc. of the 5th Int. Conference on Emerging Networking Experiments and Technologies*. ACM, 2009.

[10] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.

[11] C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and User-centric Comparison of a Novel Adaptation Logic for DASH with SVC," in *IFIP IEEE Int. Symp. on Integrated Network Management*, 2013.

[12] C. Kreuzberger, D. Posch, and H. Hellwagner, "A Scalable Video Coding Dataset and Toolchain for Dynamic Adaptive Streaming over HTTP," in *Proc. of the 6th ACM Conference on Multimedia Systems*, 2015.

[13] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice over Content-Centric Networks," in *Proc. of the 2009 Workshop on Re-architecting the Internet*. ACM, 2009.

[14] ITU-T, "Pulse Code Modulation (PCM) of Voice Frequencies," *ITU-T Recommendation G.711*, 1990.

[15] L. Sun, I.-H. Mkwawa, E. Jammeh, and E. Ifeachor, *Guide to Voice and Video over IP*. Springer, 2013.

[16] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," *ACM SIGCOMM Computer Comm. Review*, vol. 45, no. 4, pp. 325–338, Aug. 2015.

Appendix

[**Proof** Theorem 3.1] It suffices to show that the selection of the weights ω_{F_i,c_j} as given in Theorem 3.1 results in the same ordering of reliabilities as the ordering of the contents such that $R_{F_i,c_1} > R_{F_i,c_2} > \cdots > R_{F_i,c_{n-1}} > R_{F_i,c_n}$, where R_{F_i,c_1} corresponds to the reliability of the most important content with $c_1 >_c c_2 >_c \cdots >_c c_{n-1} >_c c_n$. SAF then shifts the traffic such that the faces become reliable again. The lower bound can be easily obtained from the required ordering of the reliabilities and that we want to force SAF to shift traffic from the given face to other faces or the dropping face. Therefore, we have according to Definitions 3.1 and 3.2:

$$\forall c_j, c_{j+1} \in \mathcal{C}_{F_i}, c_j >_c c_{j+1} :$$

$$\frac{S_{F_i,c_j}}{S_{F_i,c_j} + U_{F_i,c_j} \cdot \omega_{F_i,c_j}} > \frac{S_{F_i,c_{j+1}}}{S_{F_i,c_{j+1}} + U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}}$$

$$S_{F_i,c_j} > \frac{S_{F_i,c_{j+1}} \cdot S_{F_i,c_j} + S_{F_i,c_{j+1}} \cdot U_{F_i,c_j} \cdot \omega_{F_i,c_j}}{S_{F_i,c_{j+1}} + U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}}$$

$$S_{F_i,c_j} \cdot U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}} > S_{F_i,c_{j+1}} \cdot U_{F_i,c_j} \cdot \omega_{F_i,c_j}$$

$$\omega_{F_i,c_{j+1}} > \frac{S_{F_i,c_{j+1}} \cdot U_{F_i,c_j} \cdot \omega_{F_i,c_j}}{S_{F_i,c_j} \cdot U_{F_i,c_{j+1}}}$$

We derive the upper bound analogously as follows:

$$\forall c_{j+1}, c_{j+2} \in \mathcal{C}_{F_i}, c_{j+1} >_c c_{j+2} :$$

$$\frac{S_{F_i,c_{j+1}}}{S_{F_i,c_{j+1}} + U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}} > \frac{S_{F_i,c_{j+2}}}{S_{F_i,c_{j+2}} + U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}}}$$

$$S_{F_i,c_{j+1}} > \frac{S_{F_i,c_{j+2}} \cdot S_{F_i,c_{j+1}} + S_{F_i,c_{j+2}} \cdot U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}}{S_{F_i,c_{j+2}} + U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}}}$$

$$S_{F_i,c_{j+1}} \cdot U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}} > S_{F_i,c_{j+2}} \cdot U_{F_i,c_{j+1}} \cdot \omega_{F_i,c_{j+1}}$$

$$\omega_{F_i,c_{j+1}} < \frac{S_{F_i,c_{j+1}} \cdot U_{F_i,c_{j+2}} \cdot \omega_{F_i,c_{j+2}}}{S_{F_i,c_{j+2}} \cdot U_{F_i,c_{j+1}}}$$

Taking into account the requirement that the reliability should be below the corresponding reliability threshold. It follows that:

$$\forall c_j \in \mathcal{C}_{F_i}, c_j >_c c_{j+1} :$$

$$\frac{S_{F_i,c_j}}{S_{F_i,c_j} + U_{F_i,c_j} \cdot \omega_{F_i,c_j}} < t_{c_j}$$

$$\frac{S_{F_i,c_j}}{t_{c_j}} < S_{F_i,c_j} + U_{F_i,c_j} \cdot \omega_{F_i,c_j}$$

$$-U_{F_i,c_j} \cdot \omega_{F_i,c_j} < S_{F_i,c_j} - \frac{S_{F_i,c_j}}{t_{c_j}}$$

$$-U_{F_i,c_j} \cdot \omega_{F_i,c_j} < \frac{S_{F_i,c_j} \cdot (1 - t_{c_j})}{t_{c_j}}$$

$$\omega_{F_i,c_j} > \frac{S_{F_i,c_j} \cdot (1 - t_{c_j})}{t_{c_j} \cdot U_{F_i,c_j}}$$

$\omega_{F_i,c_1} \geq 1$ can be chosen arbitrarily and by calculating all the lower bounds we finally get to $\omega_{F_i,c_{|\mathcal{C}_{F_i}|}}$. The upper bound for $\omega_{F_i,c_{|\mathcal{C}_{F_i}|}}$ can be chosen arbitrarily big with the restriction that it has to be bigger than the indicated lower bound. Then we may calculate the upper bounds for all other weights and choose the weights within the determined bounds. This will instruct SAF to prioritize the contents as given by the ordering $(\mathcal{C}_{F_i}, >_c)$ on the network level. This concludes the proof. ■

How-To Repeat the Results

In the following we provide detailed information for researchers that want to reproduce the presented results. The configuration has been tested on an ordinary desktop computer using Ubuntu 14.04 64bit. To get started, please install the following packages via the **apt-get install** command: `python-dev python-pygraphviz python-kiwi python-pygoocanvas python-gnome2 python-rsvg ipython python-numpy python-scipy python-matplotlib libsqlite3-dev libcrypto++-dev libboost-all-dev git-core cmake libxml2-dev libcurl4-openssl-dev mercurial`. The next step is to checkout, compile and install:

- **ns-3**, an open source network simulator;
- **(amus)-ndnSIM v2.0**, a customized version of the ndnSIM plugin for ns-3 including DASH-based clients/server implementations for NDN;
- **libdash**, an open-source library that provides an object-oriented interface to the MPEG-DASH standard;
- **Brite**, a network topology generator that can be used as plugin for ns-3;
- **ndn-cxx**, a NDN C++ library;
- **itec-ndn**, a collection of experiments provided by the authors including the experiment presented in this paper.

Create a new folder (e.g., `ndn`) and set it to your current working directory. Install **BRITE** by using the following commands:

```
hg clone http://code.nsnam.org/BRITE
cd BRITE
make
sudo cp libbrite.so /usr/lib/
cd ..
```

Install **libdash** by using the following commands:

```
git clone https://github.com/bitmovin/libdash.git
cd libdash/libdash
mkdir build
cd build
cmake ../
make dash
cd ../../../
sudo cp ./libdash/libdash/build/bin/libdash.so  /usr/local/
      lib/
sudo mkdir /usr/local/include/libdash
sudo cp -r ./libdash/libdash/libdash/include/* /usr/local/
      include/libdash/
```

Fetch the code repositories for **ndn-cxx**, **(amus)-ndnSIM**, **itec-ndn**, and checkout the recommended versions:

```
git clone https://github.com/named-data/ndn-cxx.git ndn-cxx
git clone https://github.com/cawka/ns-3-dev-ndnSIM.git ns-3
git clone https://github.com/cawka/pybindgen.git pybindgen
git clone https://github.com/ChristianKreuzberger/amus-
      ndnSIM.git ns-3/src/ndnSIM
git clone https://github.com/danposch/itec-ndn.git
cd pybindgen
git checkout e11c02d87924d92ee80991c9d663e1398a468008
cd ../ndn-cxx
git checkout cbf054dd31596160b181ed60befe25ef388cb674
cd ../ns-3
git checkout 4e388e47d715c3206374974a40cbab7ce428936f
cd src/ndnSIM/
git checkout 86a881d9898df74fa4cfd8e85684a3ae81ab02e6
cd ../../../
```

Patch the NDN forwarder to enable the usage of third-party strategies provided by the authors:

```
cp itec-ndn/extern/forwarder.cpp ns-3/src/ndnSIM/NFD/daemon
      /fw/forwarder.cpp
cp itec-ndn/extern/forwarder.hpp ns-3/src/ndnSIM/NFD/daemon
      /fw/forwarder.hpp
cp itec-ndn/extern/ndn-content-store.hpp ns-3/src/ndnSIM/
      model/cs/ndn-content-store.hpp
cp itec-ndn/extern/content-store-impl.hpp ns-3/src/ndnSIM/
      model/cs/content-store-impl.hpp
cp itec-ndn/extern/content-store-nocache.hpp ns-3/src/
      ndnSIM/model/cs/content-store-nocache.hpp
cp itec-ndn/extern/content-store-nocache.cpp ns-3/src/
      ndnSIM/model/cs/content-store-nocache.cpp
cp itec-ndn/extern/strategy.cpp ns-3/src/ndnSIM/NFD/daemon/
      fw/strategy.cpp
cp itec-ndn/extern/strategy.hpp ns-3/src/ndnSIM/NFD/daemon/
      fw/strategy.hpp
```

Build and install **ndn-cxx**:

```
cd ndn-cxx
./waf configure
./waf
sudo ./waf install
cd ../
```

Build and install **ns-3** with the **(amus)-ndnSIM** and the **BRITE** plugin:

```
cd ns-3
./waf configure -d optimized --with-brite=../BRITE
./waf
sudo ./waf install
cd ./build
sudo cp ./libns3-dev-brite-optimized.so /usr/local/lib/
cd ../../BRITE
sudo cp *.h /usr/local/include/ns3-dev/ns3
sudo mkdir /usr/local/include/ns3-dev/ns3/Models
cd Models/
sudo cp *.h /usr/local/include/ns3-dev/ns3/Models
cd ../..
```

Build **itec-ndn** scenarios:

```
cd itec-ndn
./waf configure
./waf
```

Now everything required is installed. The scenario file for the experiment conducted in this paper can be found in `./itec-ndn/scenarios/ccr_scenario.cc`. The scenario requires a MPEG-DASH dataset that is hosted at ftp://ftp-itec.aau.at/pub/icn/ccr_dataset. Please download the dataset to you local machine and adapt line 303 in `./itec-ndn/scenarios/ccr_scenario.cc` to point to the dataset directory.

To conduct the simulations please set your current working directory to `./itec-ndn`. Open the python script `./python_scripts/ccr_scenario.py` and adapt the settings from lines 490–530 to your needs (e.g., set the number of used threads or conducted simulation runs) before executing it. The results of the simulations will be available in the folder `./output_crr` organized in sub-folders per investigated forwarding strategy. To visualize your results please copy these sub-folders to `./ccr_vis/output/*`. Set your current working directory to `./ccr_vis/` and run the python script `data_extract.py`. Once the required data is extracted from the logfiles you may use the supplied Veusz (http://home.gna.org/veusz/) and Wolfram Mathematica (http://www.wolfram.com/mathematica/) files to display your results. However, you may have to edit the files previously in a text editor in order to adapt the employed directory paths to your needs. For further information please visit http://icn.itec.aau.at or contact the authors of this paper.

Acknowledgment

The authors want to thank Michele Tortelli for his support and testing concerning the reproducibility guide.

This work was supported in part by the Austrian Science Fund (FWF) under the CHIST-ERA project CONCERT, project nr. *I1402* at Alpen-Adria-Universität Klagenfurt.