

SAF: Stochastic Adaptive Forwarding in Named Data Networking

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

Abstract—Forwarding decisions in classical IP-based networks are predetermined by routing. This is necessary to avoid loops, inhibiting opportunities to implement an adaptive and intelligent forwarding plane. Consequently, content distribution efficiency is reduced due to a lack of inherent multi-path transmission. In Named Data Networking (NDN) instead, routing shall hold a supporting role to forwarding, providing sufficient potential to enhance content dissemination at the forwarding plane. In this paper we design, implement, and evaluate a novel probability-based forwarding strategy, called Stochastic Adaptive Forwarding (SAF) for NDN. SAF imitates a self-adjusting water pipe system, intelligently guiding and distributing Interests through network crossings circumventing link failures and bottlenecks. Just as real pipe systems, SAF employs overpressure valves enabling congested nodes to lower pressure autonomously. Through an implicit feedback mechanism it is ensured that the fraction of the traffic forwarded via congested nodes decreases. By conducting simulations we show that our approach outperforms existing forwarding strategies in terms of the Interest satisfaction ratio in the majority of the evaluated scenarios. This is achieved by extensive utilization of NDN's multipath and content-lookup capabilities without relying on the routing plane. SAF explores the local environment by redirecting requests that are likely to be dropped anyway. This enables SAF to identify new paths to the content origin or to cached replicas, circumventing link failures and resource shortages without relying on routing updates.

Index Terms—Information-Centric Networking; Named Data Networking; Adaptive Forwarding; Routing.

I. INTRODUCTION

Today's Internet is based on a legacy host-based architecture, resulting in limitations. One key issue is forwarding, which is strictly predetermined by routing to ensure loop-free communication. There are few opportunities to implement an adaptive forwarding plane in classical IP-based networks since routing dictates the forwarding options. In IP, forwarding planes are stateless and routing protocols are responsible to deal with all kinds of short- and long-term topology changes. Routing protocols struggle with all the imposed responsibility. For instance, Labovitz et al. [1], [2] showed that BGP convergence times up to 15 minutes recovering from a single multi-homed fault (switching to an alternative, redundant route) are possible, if no additional mechanisms (e.g., backup configurations providing alternative routes [3]) are used. The concept of Information-Centric Networking (ICN) [4], [5] instead rests on a content-centric communication model, where content only is addressed. There are a variety of approaches for ICN architectures [6], [7], [8], [9], [10]. In this

paper, our understanding of an ICN is coincident with Named Data Networking (NDN) [10]. In NDN, data is requested by its name following a strictly receiver-driven communication model. In order to retrieve content, a consumer emits an *Interest* message, which is then forwarded by other NDN nodes until it reaches the desired content. An Interest's final destination could be the content origin, or any intermediate node that holds a cached replica. A *Data* packet carries the requested content object and is always returned on the reverse path of the requesting Interest. A typical NDN router maintains three data structures: *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 order to detect duplicate or looping Interests, a NONCE (unique bit pattern) is added to each Interest. Loop prevention is achieved by matching name and NONCE of received Interests to those already in the PIT. This enables an adaptive forwarding plane with inherent multi-path delivery, in contrast to routing in IP that avoids loops in the first place.

It has been shown by Yi et al. [11] that NDN's stateful forwarding plane handles typical network issues, such as short-term link failures and congestion, more effectively than IP networks. Furthermore, in [12] Yi et al. argue that *routing in NDN shall hold a supporting role to forwarding*. Routing should only provide a reasonable starting point for the forwarding plane, which then should explore different multipath opportunities. In return, adaptive forwarding enables a more scalable routing plane with relaxed requirements in terms of convergence time and completeness. In this paper our understanding of routing, forwarding and caching and their clear separation is in accordance with Yi et al. [11]. This assumption does not restrict the coupling of these mechanisms as suggested by [13] and [14], but clearly separates their areas of responsibility. The existing adaptive forwarding strategies in NDN do not provide all mechanisms foreseen in [11] and [12]. For this reason we propose *Stochastic Adaptive Forwarding (SAF)*, a probability-based forwarding strategy. As application field we envisage the infrastructure of Internet Service Providers (ISPs) interconnecting several autonomous systems and/or access networks. SAF is specifically designed to meet the following objectives:

- 1) Perform stochastic adaptive forwarding on a per-

content/per-prefix basis.

- 2) Provide effective forwarding even with incomplete or partly invalid routing information.
- 3) Deal with unexpected network topology changes, e.g., link failures, without relying on the routing plane.
- 4) Discover unknown paths to cached replicas.

While these attributes are beneficial for the rather versatile edge networks, they are not suitable for the rigid Internet core. Here, simpler approaches may perform better considering the well known scalability principle for networks: “*complexity at the edge, aggregation at the core*” [15].

SAF accomplishes the previously mentioned objectives by imitating a water pipe system. Network nodes act as crossings for an incoming flow of water (Interests). Returning Data packets act as input for a probability distribution which determines the share of the flow that is forwarded via the different pipes (outgoing interfaces). Each crossing maintains an overpressure valve. If the pressure on a node increases, e.g., due to network congestion, it may use the overpressure valve to lower the pressure autonomously. The Interests that pass the overpressure valve can be used for different purposes. For instance, they can be used as scouts to investigate unknown paths to complement routing information. However, in some cases it is best to simply discard these Interests. The discarded and therefore unsatisfied Interests provide implicit feedback to the requesting nodes, indicating that they should reduce the number of requests forwarded to the considered node.

SAF is based on an exchangeable measure that defines the target of the adaptive forwarding. The objective is to maximize the Interest/Data satisfaction ratio with (optional) respect to delay, hop-count, and/or transmission cost considerations. Based on the requested content, the specific service and/or the network operator's ambitions, this measure has to be chosen carefully. It determines the preferred paths for forwarding. However, adaptivity provided by SAF is not only achieved by intelligent multi-path transmission using redundant paths. SAF is also able to exploit content-based information to further improve the forwarding decisions. For instance, this is beneficial for multimedia scenarios where the relative priority of packets (e.g., VoIP vs. file transfer) is more important than a purely throughput- and/or delay-based metric may indicate.

The remainder of this paper is organized as follows. Section II discusses existing forwarding strategies for the ICN/NDN approach. Section III provides the stochastic forwarding model SAF rests upon. Section IV presents an evaluation of SAF, comparing it to other state-of-the-art forwarding strategies. Section V concludes our findings and discusses potential future work.

II. RELATED WORK

As the work in this paper focuses on forwarding in NDN, we first discuss the proposed forwarding strategies introduced by the NDN community. Yi et al. [11], [16] classify (inter-)faces based on a simple color scheme. Faces can be marked as GREEN, YELLOW and RED, which corresponds to the meaning that faces return data (GREEN), may or may not return data (YELLOW), or they do not work at all (RED). Within

this classification, faces are ranked, e.g., based on the delay of receiving Data packets. This basic scheme is used by all proposed forwarding strategies that are provided within version 1.0 of the ns3/ndnSIM simulator [17]:

- **Flooding:** Interests are forwarded to all GREEN and YELLOW faces supplied by the FIB.
- **SmartFlooding:** Interests are forwarded to the highest-ranked GREEN face. If no GREEN face is available, an Interest is forwarded via all YELLOW faces.
- **BestRoute:** Interests are forwarded to the highest-ranked GREEN face. If no GREEN face is available, an Interest is forwarded via the highest-ranked YELLOW face.

Furthermore, forwarding strategies in ns3/ndnSIM v1.0 can be supplemented by several enhancements [17]. One example are *Interest Limits*, which conceptually are Token Bucket filters. For instance, if the highest ranked interface reaches its transmission limit, an *Interest Limit* ensures that another face is selected for forwarding further requests. Further supplements are negative acknowledgment messages (NACKs), which can be returned to an Interest issuer to provide immediate feedback if a request can not be satisfied (e.g., due to the imposed Interest Limits).

Recently ns3/ndnSIM v2.0 [18] was released. The newer version no more re-implements basic NDN primitives, such as forwarding, but uses code from the *NDN Forwarding Daemon* (NFD) [19]. This allows realistic simulations since the code-base of the NFD is used, which has been developed for physical hardware. The step towards realistic simulations resulted in major changes in the simulator also affecting the implemented forwarding strategies. In ns3/ndnSIM v2.0 strategies no longer rest on the aforementioned color scheme and cannot take advantage of additional features such as *Interest Limits* or NACKs. We shortly outline the forwarding strategies available in the NFD [19]:

- **Broadcast:** Interests are forwarded to all faces supplied by the FIB.
- **ShortestRoute:** Interests are forwarded to the lowest-cost (e.g., hop count) upstream face indicated by routing. Actually this strategy is referred to as BestRoute in the NFD [19]; however, to avoid confusion with BestRoute in ns3/ndnSIM v1.0 [17], we renamed it for this paper.
- **NCC:** Interests are forwarded to those faces that provide data packets with the lowest delay.

As the aforementioned strategies are available in the ns3/ndnSIM simulator, they can easily be used for comparison with new approaches. Since we are focusing on a realistic approach with SAF, we regard ns3/ndnSIM v2.0 as the platform most suitable for evaluations and performance measurements. For this reason, we consider the algorithms Broadcast, ShortestRoute, and NCC for comparison to SAF, and do not take into account their predecessors, Flooding, SmartFlooding, and BestRoute in ndnSIM v1.0.

In [13] Rossini and Rossi propose [ideal] Nearest Replica Routing ([i]NRR), an approach to couple caching and forwarding extending aNET [20]. The iterative algorithm provided in [13] makes use of an oracle providing information on the availability of content in all caches in the network. The strategy

selects the face with the shortest distance to the content, which prefers nearby caches rather than forwarding the Interest towards the content origin. iNRR is implemented in the ccnSIM simulator [21]. Although the authors indicate that an implementation of a perfect oracle is not feasible in a real environment, we consider iNRR with perfect knowledge about the individual content chunks in the caches as a competitor to SAF. Please note that in [13] some practical approaches using off-path exploration to assess the necessary information provided by the oracle for the concept of NRR are proposed.

Chiocchetti et al. [22] developed INFORM, which is an adaptive hop-by-hop forwarding strategy using reinforcement learning inspired by the Q-routing framework. INFORM is able to discover temporary copies of content not present in the routing table, thus increasing the effectiveness of forwarding. The authors have implemented and evaluated INFORM within ccnSIM [21]. Unfortunately, they indicate on their website (<http://goo.gl/2XJa9R>) that due to a lack of manpower no release candidate or source code package for this strategy can be provided. However, the authors also indicate that one should prefer iNRR over INFORM for comparison, as it outperforms INFORM. Since iNRR is already on our list as a competitor to SAF, we do not consider INFORM.

In [23] Carofiglio et al. derive a set of optimal dynamic multipath congestion control protocols and request forwarding strategies from a multi-commodity flow problem. The proposed Request Forwarding Algorithm (RFA) in [23] is outlined in detail and is a good candidate for comparison. The idea of the algorithm is simple yet effective. For each content-prefix and for each face, RFA monitors the PIT entries. The forwarding probability of a face is then determined by a weight, which is actually a moving average over the reciprocal count of the PIT entries. We re-implemented this algorithm for ndnSIM v2.0 and NFD for comparison (more details are provided in the evaluation section).

Qian et al. [24] proposed the concept of Probability-based Adaptive Forwarding. The basic idea is to select faces based on a probability distribution, which is also similar to our approach. However, noticeable differences are that [24] is inspired by ant colony optimization and focuses on delay minimization. As described later on in the paper, SAF is generic providing opportunities to adapt forwarding in additional dimensions. We introduce a virtual face that enables content- and context-aware adaptation. Furthermore, we do not introduce distinguished Interest packets for probing only.

Yeh et al. [14] proposed VIP, a framework for *joint dynamic forwarding and caching* in NDN. In this system, *Virtual Interest Packets* (VIPs) capture the measured demand for respective data objects. The VIP count in a part of a network represents the local level of interest in a given object. The VIP framework employs a virtual control plane which operates on the VIPs. Distributed control algorithms are used to guide caching and forwarding strategies. We do not consider the VIP framework as a competitor due to: *i)* absence of a reference implementation; *ii)* difficulties to precisely reproduce the results: the illustration in [14] leaves room for interpretation regarding the individual parts (virtual control plane, communication protocols, control algorithms) and their interplay. *iii)* no

estimation regarding the communication overhead is provided.

Recently, Udugama et al. [25] published On-demand Multi-Path Interest Forwarding (OMP-IF). This forwarding strategy uses multiple node disjoint paths for Interest forwarding simultaneously. Each router may only use a single face (from the FIB) for forwarding per content-prefix to ensure node disjointness. The client is responsible for triggering the multipath transmission by utilizing a weighted round-robin mechanism based on path delays distributing Interests over multiple faces. However, considering only node disjoint paths may leave some network resources unused. We consider OMP-IF as a competitor to SAF.

III. STOCHASTIC ADAPTIVE FORWARDING

This section deals with the terminology and design of SAF. First, the network, content and node models are discussed, which state the necessary preconditions for the presented approach. Subsequently, we show that SAF enables adaptive forwarding based on a given measure. The design of \mathcal{M}_T , an exemplary measure maximizing the Interest satisfaction ratio, is illustrated. SAF defines update operations that modify the forwarding probabilities for (inter-)faces. The ultimate goal of SAF is to optimize a node's forwarding behavior such that it performs optimally in terms of a given measure. The design of these update operations is presented and exemplarily discussed based on \mathcal{M}_T . Finally, two examples are presented illustrating the functionality of SAF.

A. Network, Content and Node Models

SAF rests on the following network model: $\mathcal{N}(\mathcal{V}, \mathcal{E})$ denotes a network consisting of a set of nodes \mathcal{V} and a set of edges/links $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each node $v \in \mathcal{V}$ maintains a physical face $F_{(v,u)}$ (e.g., wireless network interface) for each tuple $(v,u) \in \mathcal{E}$. We define the list of physical faces on v as $\mathcal{F}'_v := \bigcup_{(v,u) \in \mathcal{E}} F_{(v,u)}$, where $|\mathcal{F}'_v|$ denotes the number of physical links/faces of v . A node may receive Interests on any $F_{in} \in \mathcal{F}'_v$ and tries to satisfy these requests by either returning a locally stored copy of the requested data or by forwarding the Interest to a suitable face $F_{out} \in \mathcal{F}'_v \setminus \{F_{in}\}$. The content catalogue in \mathcal{N} is determined by a set \mathcal{C} . Each $c \in \mathcal{C}$ denotes content that can be retrieved using a common prefix.

In addition to the physical faces \mathcal{F}'_v , each node maintains a distinguished virtual face F_{D_v} , which acts as overpressure valve. The set $\mathcal{F}_v = \mathcal{F}'_v \cup \{F_{D_v}\}$ denotes the entire set of faces known to v . SAF is an algorithm local to each node, requiring no explicit communication between nodes. Therefore, further discussion on SAF focuses only on a single node, which allows us to omit the subscripts v identifying a specific node for the remainder of this paper. The virtual face F_D is treated as an ordinary face by SAF, however, any Interest forwarded to this face is discarded; the use of this *dropping face* will be made clear in the course of the paper.

Every node maintains a so called *Forwarding Table* (FWT). This table is a two-dimensional matrix, where the rows correspond to the set of faces \mathcal{F} and the columns correspond to the different contents from the catalogue \mathcal{C} . The elements of the matrix indicate the confidence (probability) with which a

certain outgoing face can provide data for a certain prefix. For instance, the following matrix represents an example FWT for a node with $\mathcal{F} = \{F_D, F_0, F_1, F_2\}$ and $\mathcal{C} = \{c_0, c_1, c_2\}$:

$$\text{FWT} = \begin{matrix} & c_0 & c_1 & c_2 \\ F_D & 0 & 0 & 1/3 \\ F_0 & 1/3 & 1/2 & 0 \\ F_1 & 2/3 & 0 & 2/3 \\ F_2 & 0 & 1/2 & 0 \end{matrix}$$

The FWT provides the probability of forwarding an Interest for content c_l on face F_i . We denote $p(F_i, c_l)$ as the forwarding probability that an Interest asking for c_l will be forwarded on F_i , for instance, $p(F_1, c_0) = \frac{2}{3}$. Note that, for any c_l , the corresponding column of the FWT specifies a discrete probability distribution: $\sum_{F_i \in \mathcal{F}} p(F_i, c_l) = 1$. The decision to forward a given Interest on a face is as simple as drawing a random number from a uniform distribution $U(0, 1 - p(F_{in}, c_l))$. Algorithm 1 sketches the face selection process, which is also known as *inverse transform sampling*. The function *nextDouble()* (cf. Alg. 1 line 2) draws a number from the distribution U and *pop()* (cf. Alg. 1 line 4) removes and returns the top element of a list (F_{list}). The algorithm requires $O(|\mathcal{F}|)$ steps. As the number of faces is usually constant on a node, the algorithm is actually in $O(1)$.

Algorithm 1 Select outgoing Face for an Interest

```

1:  $F_{list} \leftarrow \mathcal{F} \setminus \{F_{in}, F_D\}$ ,  $limit \leftarrow 0.0$ 
2:  $rand \leftarrow U(0, 1 - p(F_{in}, c_l)).nextDouble()$ 
3: while  $F_{list} \neq \emptyset$  do
4:    $F_{cur} \leftarrow F_{list}.pop()$ 
5:    $limit \leftarrow limit + p(F_{cur}, c_l)$ 
6:   if  $rand \leq limit$  then
7:     return  $F_{cur}$ 
8:   end if
9: end while
10: return  $F_D$ 

```

Figure 1 illustrates an NDN node using SAF. The design of the FWT provides two opportunities to perform adaptive forwarding, which are implemented by the *Adaptation Engine*. First, modifications of the probabilities within a single column (solid, red lines in Figure 1) of an FWT change the forwarding probabilities for Interests asking for a specific c_l . These updates modify which faces/paths are preferred for forwarding Interests. Second, shifting forwarding probabilities among different columns (dashed, blue lines in Figure 1) allows prioritization of specific content types/prefixes. For instance, assume c_l is more important than c_k . In this case it can be beneficial to increase the probability of dropping Interests for c_k in favor of c_l . The necessary statistical information for these operations is provided by the *Statistic Collector*, which monitors the Interests received and satisfied by faces.

This paper focuses on the core forwarding part of SAF (cf. solid, red lines in Figure 1). Additional context- and content-aware adaptations (cf. dashed, blue lines in Figure 1) are inherently supported by the design of SAF, however, they are considered as out of scope for this paper. Note that the forwarding core of SAF operates separately within a single column (content prefix) of the FWT. Therefore, and for the sake of readability we omit the notion of different $c \in \mathcal{C}$ for the remainder of this paper.

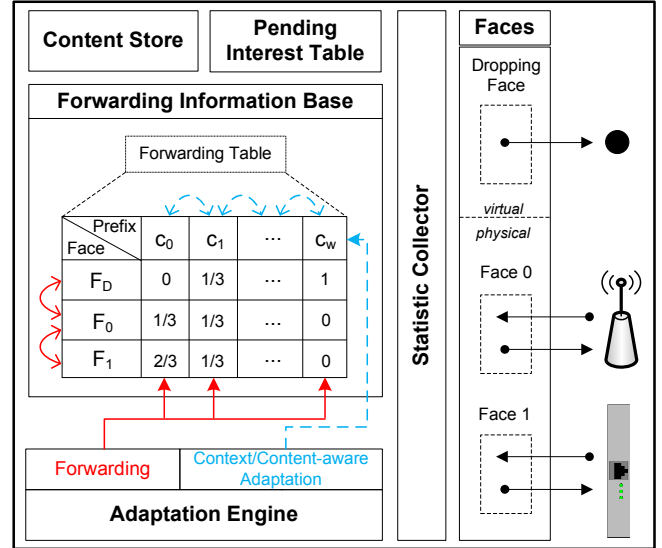


Fig. 1: The model of an NDN node using SAF.

B. A Throughput-based Forwarding Measure

SAF is based on the measure \mathcal{M} as defined as follows:

Definition III.1. Let (Ω, \mathcal{A}) be a measurable space with a set Ω , the corresponding σ -Algebra $\mathcal{A} \subseteq \mathcal{P}(\Omega)$ and $\mathcal{M}, S, U : \mathcal{A} \rightarrow \mathbb{R}$ functions with:
 $\forall F_i \in \mathcal{F} \setminus \{F_D\}, \forall A \in \mathcal{A} : \mathcal{M}_{F_i}(\emptyset) = 0$,
 $\mathcal{M}_{F_i}(A) = S_{F_i}(A) - U_{F_i}(A)$ and $\forall A \in \mathcal{A} : \mathcal{M}_{F_D}(A) = 0$.

$S_{F_i}(A)$ provides a measure for *satisfying Interests* and $U_{F_i}(A)$ defines a measure for *not satisfying Interests*. Note that, since $U_{F_i}(A)$ is the complementary measure to $S_{F_i}(A)$, it suffices to define $S_{F_i}(A)$. $S_{F_i}(A)$ is not predefined by SAF, which allows to use an arbitrary measure. For instance, one may define $S_{F_i}(A)$ as the number of Interests that are satisfied by Data packets below a certain delay threshold. The objective of \mathcal{M} , therefore the definition of $S_{F_i}(A)$, guides the update operations for the FWT. These operations are issued periodically, e.g., once every second. SAF finds the optimal forwarding strategy by maximizing \mathcal{M} over the periods. The forwarding core of SAF operates separately on each column (prefix) of the FWT, thus providing for decoupling of periods for the different contents (prefixes). This allows to spread the node's work load over time.

In this paper we use the measure \mathcal{M}^T , which maximizes the throughput by investigating the Interest satisfaction ratio on individual faces. Therefore we consider for each node the set of forwarded Interests \mathcal{I}_n during period n , for which a Data packet has been received (*satisfied* Interest) or a timeout has occurred (*unsatisfied* Interest) within that period n . Interests neither satisfied nor unsatisfied within a period are termed *pending*. We define $S_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is satisfied by a Data packet on } F_i\}|$ and $U_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is not satisfied on } F_i\}| \forall F_i \in \mathcal{F} \setminus \{F_D\}$, and $S_{F_D}(\mathcal{I}_n) = |\{j \in \mathcal{I}_n : j \text{ is satisfied by } F_D\}|$ (F_D satisfies Interests by definition, but note that $\mathcal{M}_{F_D}(A) = 0$; $U_{F_D}(\mathcal{I}_n) = 0$). This implicitly defines \mathcal{M}^T .

Remark III.1. The ongoing discussion of SAF is agnostic to a

VAR./EXP.	DEFINITION / EXPLANATION
S_{F_i}	Number of satisfied Interests on F_i . F_D satisfies Interests by definition.
U_{F_i}	Number of unsatisfied Interests on F_i .
I	$\sum_{F_i \in \mathcal{F}} [S_{F_i} + U_{F_i}]$, satisfied and unsatisfied without pending Interests.
ST_{F_i}	$\begin{cases} \frac{S_{F_i}}{I} & \text{if } I > 0 \\ 0 & \text{otherwise.} \end{cases}$ Satisfied traffic fraction on F_i .
UT_{F_i}	$\begin{cases} \frac{U_{F_i}}{I} & \text{if } I > 0 \\ 0 & \text{otherwise.} \end{cases}$ Unsatisfied traffic fraction on F_i .
R_{F_i}	$\begin{cases} \frac{S_{F_i}}{S_{F_i} + U_{F_i}} & \text{if } S_{F_i} + U_{F_i} > 0 \\ 1 & \text{otherwise.} \end{cases}$ Reliability of F_i .
$p(F_i)$	Forwarding probability for face F_i .
t	$t \in [t_{min}, t_{max}]$, reliability threshold. $t_{max} \in]0, 1[$, and $t_{min} \in]0, t_{max}[$.
\mathcal{F}_R	$\{F_i \in \mathcal{F} \setminus \{F_D\} \mid R_{F_i} \geq t\}$, reliable faces.
\mathcal{F}_U	$\mathcal{F} \setminus (\mathcal{F}_R \cup \{F_D\})$, unreliable faces.
\mathcal{F}_S	$\{F_i \in \mathcal{F}_R \mid ST_{F_i} + UT_{F_i} > 0\}$, faces that may take additional traffic.
\mathcal{F}_D	$\mathcal{F}_R \setminus \mathcal{F}_S$, faces used for probing.
$\delta \in [0, 1]$	Total unsatisfied traffic fraction.
$\delta_U \in [0, 1]$	Unsatisfied traffic on faces in \mathcal{F}_U .
$\alpha_{F_i} \in]0, 1]$	$\frac{1}{1 + \sqrt{\text{var}(X)}}$, traffic stability indicator, where X is a window over S_{F_i} with length N .
$\Delta \in [0, 1]$	δ_U with respect to $\alpha_{F_i} \forall F_i \in \mathcal{F}_U$.
σ_{F_i}	Resources for additional Interests on F_i .
$\rho \in [0, 1]$	Traffic fraction used for probing.

TABLE I: Variables and expressions for SAF.

concrete instantiation of a measure \mathcal{M} . One may use other measures than \mathcal{M}^T considering the hop count or delay of received Interests. For instance, one may define a hop count-based measure \mathcal{M}^H as follows: $S_{F_i}(\mathcal{I}_n) := |\{j \in \mathcal{I}_n : j \text{ is satisfied by a Data packet } d \text{ on } F_i, \text{ where } d \text{ traversed fewer than } h \text{ hops}\}|, \forall F_i \in \mathcal{F} \setminus \{F_D\}$, and $S_{F_D}(\mathcal{I}_n) = |\{j \in \mathcal{I}_n : j \text{ is satisfied by } F_D\}|$.

For the sake of simplicity we write S_{F_i} for $S_{F_i}(\mathcal{I}_n)$ and U_{F_i} for $U_{F_i}(\mathcal{I}_n)$ for the remainder of the paper. Table I depicts variables and expressions which are used for the definition of \mathcal{M} and for SAF's update operations. Since these operations are executed during the transition from one period to another, variables hold the observed system state that is observable at the *end* of a given period.

Note that SAF is also scalable concerning space complexity. While classical approaches maintain a list of outgoing faces per prefix, SAF requires a vector of forwarding probabilities instead (cf. Figure 1). For instance, assume that those probabilities are quantized into the range of a byte. Furthermore, for each combination of prefixes and faces, SAF requires two counting variables (e.g., 4 bytes) representing S_{F_i} and U_{F_i} . Then the space complexity for SAF is given by $|\mathcal{C}| \cdot |\mathcal{F}| \cdot 9$ bytes. Since \mathcal{F} is usually fixed on a node, space complexity is in $O(|\mathcal{C}|)$, as for most forwarding strategies (cf. Section II).

C. Identifying Unsatisfied Traffic

For all update operations, SAF requires knowledge about the total unsatisfied traffic fraction δ . It provides information

about the traffic percentage that has been forwarded towards wrong faces, given the measure \mathcal{M} . Using the expressions from Table I, δ can be defined as given by Equation 1.

$$\delta = \begin{cases} 1 - \sum_{F_i \in \mathcal{F}} ST_{F_i} = \sum_{F_i \in \mathcal{F}} UT_{F_i} & \text{if } I > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

δ denotes the amount of traffic that should be forwarded on other faces during the next period. However, it is not yet known which faces provide a *poor service* and should therefore receive less traffic, and vice versa. For this reason, SAF splits the set of all physical faces $\mathcal{F} \setminus \{F_D\}$ into two disjoint subsets: (i) \mathcal{F}_R , the set of reliable faces, and (ii) \mathcal{F}_U , the set of unreliable faces. This partitioning is based on the definition of the reliability of a face R_{F_i} and the dynamic reliability threshold t . The threshold t is adapted based on a node's *health* status in the interval $t \in [t_{min}, t_{max}]$, which will be discussed later. Note that the definition of R_{F_i} is based only on the defined measure S_{F_i} .

The partitioning of $\mathcal{F} \setminus \{F_D\}$ into \mathcal{F}_R and \mathcal{F}_U provides a starting point to improve a node's forwarding decisions. SAF's update operations focus on shifting traffic from the unreliable faces towards the reliable faces. The next step for SAF is to evaluate the amount of traffic from faces in \mathcal{F}_U that can be shifted to faces in \mathcal{F}_R without overloading those. For this purpose, we define δ_U (cf. Equation 2), which specifies the accumulated unsatisfied traffic from faces in \mathcal{F}_U only.

$$\delta_U = \begin{cases} \sum_{F_i \in \mathcal{F}_U} UT_{F_i} & \text{if } I > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

D. Update Operations

At the end of each period, SAF's objective is to shift the traffic fraction δ_U from \mathcal{F}_U to \mathcal{F}_R , or, in the worst case, to the virtual face F_D . In order to neglect short-term effects, the shifting of traffic is relaxed by $\alpha_{F_i} \in]0, 1]$. α_{F_i} is an indicator for the stability of the satisfied traffic over F_i and is defined in Table I. It is determined by the standard deviation of the satisfied Interests over a given number of periods. Note that, the larger the standard deviation of S_{F_i} over the periods, the smaller α_{F_i} , and vice versa. This ensures that SAF balances the updates of the FWT taking into account traffic stability, which allows stronger changes if the observed state is steady over the periods. Plugging α_{F_i} into Equation 2 provides the relaxed unsatisfied traffic fraction Δ , as denoted in Equation 3.

$$\Delta = \begin{cases} \sum_{F_i \in \mathcal{F}_U} UT_{F_i} \cdot \alpha_{F_i} & \text{if } I > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Algorithm 2 outlines SAF's update procedure, which can be used as the road map for Subsections III-D to III-F. In line 2 of Algorithm 2, Γ is introduced. Γ denotes the sum of the unsatisfied traffic fraction Δ and the current forwarding probability $p(F_D)$ of the virtual face F_D . It is important to consider the sum of Δ and $p(F_D)$, as Δ does not consider the discarded traffic fraction on F_D . Note that $\Delta > 0 \Leftrightarrow \mathcal{F}_U \neq \emptyset$ as indicated by Proposition III.1. *Remark:* $\Delta = 0 \not\Leftrightarrow UT_{F_i} = 0 : \forall F_i \in \mathcal{F}$.

Algorithm 2 Pseudocode for the FWT updates in SAF

```

1:  $\Delta \leftarrow \text{determineUnsatisfiedTraffic}()$ 
2:  $\Gamma \leftarrow \Delta + p(F_D)$ 
3: if  $\Gamma > 0$  then
4:    $\mathcal{F}_S, \mathcal{F}_P \leftarrow \text{splitSet}(\mathcal{F}_R)$ 
5:    $\text{shiftTraffic}(\mathcal{F}_U, \mathcal{F}_S, \Gamma)$ 
6:    $p(F_D) = \Gamma - \Gamma'$ 
7:   if  $p(F_D) > 0$  then
8:      $\text{probeOnFaces}(\mathcal{F}_P)$ 
9:     if  $p(F_D) > (1 - t)$  then
10:       $\text{decreaseReliability}(t)$ 
11:     end if
12:   end if
13: else if  $I > 0$  then
14:    $\text{increaseReliability}(t)$ 
15: end if
    
```

Proposition III.1. Suppose $\forall F_i \in \mathcal{F} : \alpha_{F_i} \in]0, 1]$ and $I > 0$. Then, it holds that $\Delta > 0 \iff \mathcal{F}_U \neq \emptyset$.

[Proof of Prop. III.1] Follows directly from the definition of Δ and \mathcal{F}_U (cf. Table I). ■

Algorithm 2 has a trivial case, which eventuates if $\Gamma = 0$. In this case no changes in the FWT are required, since neither unsatisfied traffic exists nor any traffic is dropped in advance. In this favorable case, t is increased if $I > 0$ in this period. However, if $\Gamma > 0$, SAF resolves the unsatisfied traffic using the two following approaches: *i*) adaptation of the forwarding probabilities within the FWT (cf. Alg. 2 line 4-5); *ii*) identification of yet unknown paths to the desired content via probing (cf. Alg. 2 line 8). For the sake of simplicity, we separate the further discussion of Algorithm 2 into these two parts. The adaptation of the reliability threshold (cf. Alg. 2 lines 10 and 14) is discussed at the end of the second part.

Note that Algorithm 2 is executed for each content prefix in the FWT (cf. Figure 1). If we assume the worst case for Algorithm 2 ($\Gamma > 0$, $p(F_D) > 0$, and $p(F_D) > (1 - t)$), then the asymptotic time complexity is given by $O(|\mathcal{C}|)$. This is due to the fact that only simple arithmetic operations are used (cf. Subsection III-E and III-F), and their quantity solely depends on the number of faces $|\mathcal{F}|$, which is usually constant.

E. Adaptation of Forwarding Probabilities

SAF shifts traffic between faces only if $\Gamma > 0$. The principal objective in this step is to shift the unsatisfied traffic from the unreliable faces \mathcal{F}_U towards the reliable faces \mathcal{F}_R (cf. Alg. 2 line 5) without overloading them. Of course, this is not always possible, which in the worst case forces the algorithm to forward some Interests towards the virtual face F_D (cf. Alg. 2 line 6). Before any actions are taken, SAF splits the set \mathcal{F}_R into two disjoint subsets. The subset $\mathcal{F}_S \subseteq \mathcal{F}_R$ includes only faces from \mathcal{F}_R which have successfully forwarded Interests in the current period. The second subset, $\mathcal{F}_P = \mathcal{F}_R \setminus \mathcal{F}_S$ (cf. Table I), includes faces that are considered as reliable only because they have not forwarded any Interests ($\forall F_i \in \mathcal{F}_P : S_{F_i} = U_{F_i} = 0$) in the current period. So it is very likely that faces in \mathcal{F}_P cannot fulfill requests, which is why we do not consider them for *attracting additional* traffic.

Before SAF performs the shifting, it determines how much additional traffic the faces in \mathcal{F}_S may take, without decreasing

their reliability below t . Proposition III.2 provides σ_{F_i} , which denotes the number of additional Interests the face $F_i \in \mathcal{F}_S$ may take without dropping R_{F_i} below t .

Proposition III.2. For a given reliability t , every $F_i \in \mathcal{F}_S$ can satisfy $0 \leq \sigma_{F_i} \leq \lfloor \frac{S_{F_i}}{t} - S_{F_i} - U_{F_i} \rfloor$ additional Interests.

[Proof of Prop. III.2] Follows directly from the definition of R_{F_i} (cf. Table I). ■

Given Γ and $\sigma_{F_i} \forall F_i \in \mathcal{F}_S$, SAF is able to determine the maximum traffic that can/should be shifted from \mathcal{F}_U to \mathcal{F}_S . We denote this amount as Γ' as defined in Equation 4.

$$\Gamma' = \min\left(\frac{1}{I} \cdot \sum_{F_i \in \mathcal{F}_S} \sigma_{F_i}, \Gamma\right) \quad (4)$$

The next step for SAF is to determine the forwarding probabilities for period $n + 1$ by: *i*) decreasing the forwarding probabilities for faces in \mathcal{F}_U by Γ' , Equation 5; *ii*) increasing the forwarding probabilities for faces in \mathcal{F}_S by Γ' , Equation 6.

$$\forall F_i \in \mathcal{F}_U : p_{n+1}(F_i) \leftarrow p_n(F_i) - UT_{F_i} \cdot \alpha_{F_i} \quad (5)$$

$$\forall F_i \in \mathcal{F}_S : p_{n+1}(F_i) \leftarrow p_n(F_i) + \Gamma' \cdot \frac{\sigma_{F_i}}{\sum_{F_i \in \mathcal{F}_S} \sigma_{F_i}} \quad (6)$$

Note that Equation 5, does not use Γ' to determine the amount of the traffic reduction. Instead, the complete unsatisfied traffic from a face $F_i \in \mathcal{F}_U$ considering α_{F_i} is removed, which exactly adds up to Δ . As the residual traffic ($\Gamma - \Gamma'$) can not be satisfied by any $F_i \in \mathcal{F} \setminus \{F_D\}$, it is beneficial to drop this portion of the traffic. Forwarding those Interests would likely cause congestion and impair the performance. So SAF simply determines the amount of residual unsatisfied traffic and puts it on the virtual face F_D , denoted by Equation 7 (cf. Alg. 2 line 6). Theorem III.1 shows that SAF converges to a steady state, which is defined as a state where $\mathcal{F}_U = \emptyset$.

$$p(F_D) = \Gamma - \Gamma' \quad (7)$$

Theorem III.1. Given any state with $\mathcal{F}_U \neq \emptyset$, $t \in [t_{min}, t_{max}]$ and without the loss of generality set α sufficiently small (i.e., $\alpha = \min(\alpha_{F_i} : F_i \in \mathcal{F}_U)$). d_{F_i} denotes the number of Interests that can be satisfied on face F_i , $p^*(F_i) = \frac{d_{F_i}}{I}$ denotes the optimal forwarding probability for face F_i and I denotes the number of Interests that shall be forwarded in every period. Assume that I is constant for every period and that $I \cdot p(F_i) \geq \frac{d_{F_i}}{t}$. According to Equations 5, 6 and 7 SAF converges to $\mathcal{F}_U = \emptyset$ after n periods (iterations) bounded by:

$$n \leq \max_{F_i \in \mathcal{F}_U} \left(\left\lceil \frac{\ln\left(\frac{d_{F_i}}{t} - d_{F_i}\right) - \ln(p_0(F_i) \cdot I - d_{F_i})}{\ln(1 - \alpha)} \right\rceil \right), \quad (8)$$

where p_0 denotes the initial forwarding probability.

[Proof of Theorem III.1] In the case of $\mathcal{F}_U \neq \emptyset$, SAF uses Equation 5 to reduce the forwarding probabilities on every $F_i \in \mathcal{F}_U$ until all become reliable. According to the definition of R_{F_i} , $F_i \in \mathcal{F}_U$ iff $p_n(F_i) \cdot I > \frac{d_{F_i}}{t}$. In this case we either shift the probabilities $\forall F_i \in \mathcal{F}_U$ to faces in \mathcal{F}_S , or we drop the traffic by increasing the forwarding probability of F_D . Since we may express UT_{F_i} as $p_n(F_i) - \frac{d_{F_i}}{I}$ assuming a

perfect random distribution for Algorithm 1, and according to Equation 5 we have, $\forall F_i \in \mathcal{F}_U$:

$$p_{n+1}(F_i) = p_n(F_i) \cdot (1 - \alpha) + \alpha \cdot \frac{d_{F_i}}{I}. \quad (9)$$

Solving the recursion we have,

$$p_n(F_i) = p_0(F_i) \cdot (1 - \alpha)^n + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=0}^{n-1} (1 - \alpha)^j. \quad (10)$$

Equation 10 and $\lim_{n \rightarrow \infty} p_n(F_i) = \frac{d_{F_i}}{I}$ provides the claim that $p_n(F_i) \rightarrow p^*(F_i)$ and with $t \rightarrow 1$, $p^*(F_i) = \frac{d_{F_i}}{I}$ denotes the optimum for face F_i . Further, one easily shows that the convergence speed ($|p_{n+1}(F_i) - p^*(F_i)| \leq M \cdot |p_n(F_i) - p^*(F_i)|$) is at least linear with $M = (1 - \alpha)$. By plugging $p_n(F_i)$ into $p_n(F_i) \cdot I > \frac{d_{F_i}}{t}$ and using the formula for the geometric series ($0 < 1 - \alpha < 1$) we have,

$$(1 - \alpha)^n \cdot (p_0(F_i) \cdot I - d_{F_i}) + d_{F_i} > \frac{d_{F_i}}{t}. \quad (11)$$

Then the number of periods n until all unreliable faces become reliable is bounded by Equation 8. ■

Remark III.2. In Theorem III.1 we assumed a constant number of Interests in each period. In order to show that Theorem III.1 is still valid given a varying number of Interests for each time dependent period, we may define a period by the number of Interests. Thus, we again have a constant number of Interests during a period and Theorem III.1 holds.

Remark III.3. Let $(\mathbb{R}, |x - y|)$ be a metric space. According to Equation 9, $p^*(F_i)$ is a fixed point. This fixed point is globally asymptotically stable for $\alpha \in]0, 1]$.

[Proof of Remark III.3] Let $\Phi(k, \kappa, \xi) = \xi(1 - \alpha)^{k-\kappa} + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=\kappa}^{k-1} (1 - \alpha)^{k-(j+1)} \cdot \mathbf{1}_{\{k \geq \kappa\}}$ with initial pair $(\xi, \kappa) \in \mathbb{R} \times \mathbb{N}$, $p_\kappa = \xi$ and $k \geq \kappa$ be the general solution to our autonomous difference equation given in Equation 9. First, we show that $p^*(F_i)$ is a fixed point and it is globally attractive.

$$p_n(F_i) \cdot (1 - \alpha) + \alpha \cdot \frac{d_{F_i}}{I} = p_n(F_i) \iff p_n(F_i) = \frac{d_{F_i}}{I}.$$

According to Theorem III.1, it follows that for $\alpha \in]0, 1]$ the fixed point $p^*(F_i)$ is globally attractive ($\forall (\xi, \kappa) \in \mathbb{R} \times \mathbb{N} : \lim_{n \rightarrow \infty} |p_n(F_i) - p^*(F_i)| = 0$). Second, we show that the solution $p^* = \frac{d_{F_i}}{I}$ is stable for all $\alpha \in]0, 1]$. For $\alpha \in]0, 1]$, $\forall k \in \mathbb{N}, k \geq \kappa$ and for any $\varepsilon > 0$ we have,

$$\begin{aligned} |\Phi(k, \kappa, \xi) - p^*| &= \\ |\xi \cdot (1 - \alpha)^{k-\kappa} + \alpha \cdot \frac{d_{F_i}}{I} \cdot \sum_{j=\kappa}^{k-1} (1 - \alpha)^{k-(j+1)} \cdot \mathbf{1}_{\{k \geq \kappa\}} - \frac{d_{F_i}}{I}| &= \\ |(1 - \alpha)^{k-\kappa} \cdot \left(\xi - \frac{d_{F_i}}{I} \right)| \leq \left| \xi - \frac{d_{F_i}}{I} \right| < \varepsilon \quad \forall \xi \in B_\delta \left(\frac{d_{F_i}}{I} \right), \end{aligned}$$

with $\delta = \frac{\varepsilon}{1 - \alpha}$, where $B_\delta(y) := \{x \in \mathbb{R} : |x - y| < \delta\}$. For $\alpha = 1$ the proof is trivial. Thus, p^* is globally asymptotically stable. ■

F. Searching for Unknown Paths and Cached Replicas

After shifting traffic in the previous step, $p(F_D)$ holds the residual traffic that cannot be forwarded on the physical faces. This traffic is going to be dropped by the virtual face F_D . SAF's probing mechanism takes a share of this traffic and uses it to discover new paths towards the content origin or to

discover nodes holding cached replicas. The fraction of the traffic that is used for probing is limited by ρ as denoted in Equation 12. The probe is defined by Equation 13.

$$\rho = 1 - \sum_{F_i \in \mathcal{F} \setminus F_D} ST_{F_i} = \delta - ST_{F_D} \quad (12)$$

$$probe = p(F_D) \cdot \rho \quad (13)$$

ρ increases based on the proportion of the satisfied and the unsatisfied traffic on physical faces. The larger the fraction of the unsatisfied traffic, the larger ρ . The idea behind this is that the more unsatisfied traffic we have, the more important it is to discover additional paths to the content. In the worst case where $p(F_D) = 1$ the entire traffic is used for probing. For instance, this can happen when a previously working path to the content suffers from link failure(s). In this case probing may help to circumvent the broken link or to identify yet unknown path(s) to cached content replicas. The probe is uniformly distributed on all faces in \mathcal{F}_P as denoted in Equation 14. Finally, the forwarding probability of the virtual face F_D needs to be adjusted as outlined in Equation 15.

$$\forall F_i \in \mathcal{F}_P : p_{n+1}(F_i) = p_n(F_i) + \frac{probe}{|\mathcal{F}_P|} \quad (14)$$

$$p_{n+1}(F_D) = p_n(F_D) - probe \quad (15)$$

After the probing has been carried out, SAF checks if it has to decrease the reliability threshold (cf. Algorithm 2 lines 9–10). If $p(F_D)$ is larger than $(1 - t)$, the reliability threshold has to be decreased since it cannot be retained. In contrast to this, the reliability threshold should be increased if currently all Interests can be satisfied with a reliability of at least t (cf. Algorithm 2 lines 13–14). We suggest Equations 16 and 17 for the adjustment of t , where λ denotes the rate of change.

$$t_{n+1} = (1 - \lambda) \cdot t_n + \lambda \cdot t_{max} \quad (16)$$

$$t_{n+1} = (1 - \lambda) \cdot t_n - \lambda \cdot t_{min} \quad (17)$$

G. SAF Examples

We present two examples, one illustrating how SAF approaches the optimal FWT, and a second one illustrating the probing mechanism that allows SAF to recover from a link failure in the given scenario. Figure 2 depicts the example network with the corresponding link capacities in Mbps. Our object of investigation is the router R . This router has to forward Interests for a given content prefix c , which can be retrieved at the content provider. R maintains three physical faces which have links to the routers F_0, F_1 , and F_2 (we use faces from R interchangeably with routers F_0, F_1 , and F_2). Interests forwarded via F_1 and F_2 reach the content provider. We assume that R has to satisfy a constant flow of Interests requesting 3 Mbps of Data packets. Furthermore, we assume that the initial value for $t = 0.5$ and $\forall F_i \in \mathcal{F} : \alpha_{F_i} = 1$ for all periods. The initial FWT for R can either be provided by the routing layer, or, as in this example, the traffic is uniformly distributed among the faces (cf. Figure 3a). During the first period each of the faces F_0, F_1 , and F_2 receives Interests requesting Data packets for a bitrate of 1 Mbps. While F_1 and F_2 satisfy all Interests, F_0 is not able to satisfy any. Therefore, F_0 is classified as an unreliable face and since $\alpha_{F_0} = 1$ the entire unsatisfied traffic is removed (cf. Eq. 5). The forwarding

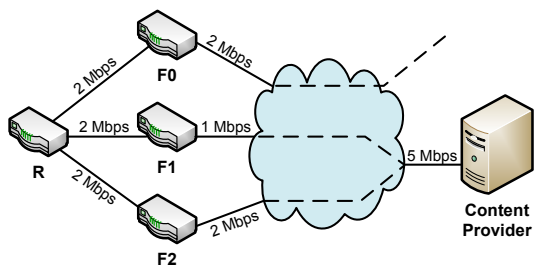


Fig. 2: Example network.

probability of F_0 is shifted towards F_1 and F_2 (cf. Eq. 6 and Figure 3b). In the second period both, F_1 and F_2 , receive Interests requesting Data packets for 1.5 Mbps. While F_2 is able to satisfy all Interests, F_1 satisfies only $\frac{2}{3}$. However, based on t both faces are considered as reliable and no changes are performed on the FWT (cf. Figure 3c). Only the threshold t is increased (cf. Alg. 2 line 14), assume t is increased to 0.75. In the third period the traffic is distributed as in the second period, with the same issue that F_1 can satisfy only $\frac{2}{3}$ of the Interests. However, this time $t > \frac{2}{3}$, which classifies F_1 as unreliable. Therefore, the unsatisfied traffic (≈ 0.08) is shifted towards F_2 (cf. Figure 3d). With each further period, the FWT approaches the optimal distribution ($p(F_1) = \frac{1}{3}$, $p(F_2) = \frac{2}{3}$).

For the second example we consider the same network as before with one major change. The path traversing F_0 ends at a content replica of the content provider. We use the optimal FWT from the previous example as the starting point for the second example, $t = 0.99$ and $\forall F_i \in \mathcal{F} : \alpha_{F_i} = 1$ for all periods. We further assume that, at the beginning of the first period, a link failure on the path between F_2 and the content provider occurs. Therefore, none of the Interests can be satisfied via F_2 , which causes this face to be marked as unreliable. As $t = 0.99$, F_1 is not able to take any additional Interests from F_2 causing this traffic portion ($\frac{2}{3}$) to be shifted to the virtual face F_D . As $p(F_D) > 0$, probing is issued (cf. Alg. 2 line 7). This results in the situation that a share of $p(F_D) \cdot \rho = \frac{2}{3} \cdot \frac{2}{3}$ Interests is used for probing on face F_0 (cf. Eq 12 and 14). Figure 4b illustrates the FWT after the first iteration. Since $p(F_D) > 1 - t$, t is decreased. Assume t is decreased to $t = 0.75$. In the second period, all Interests forwarded via F_0 and F_1 can be satisfied and SAF is able to discover a new path to the content via F_0 . After the first iteration, still 22.2% of the Interests are dropped in advance. Since in the second period all Interests forwarded via F_0 and F_1 can be satisfied and $\sigma_{F_0} + \sigma_{F_1} > p(F_D)$ ($\approx 0.148 + 0.111 > 0.222$) the forwarding probabilities for F_D are distributed according to Eq. 6 (cf. Figure 4c). As the capacity of F_1 was already exhausted, a part of the traffic ($\approx 0.43 - 0.33 \approx 0.1$) is dropped during the third period due to congestion. However, according to $t = 0.75$, F_1 is still reliable and since all faces are considered as reliable during the third period the FTW is not modified between iterations two and three. Only the reliability threshold is increased, assume t is increased to 0.85. Therefore, after the fourth period F_1 is considered as unreliable and the unsatisfied traffic is completely taken away according to Eq. 5 (as $\forall F_i \in \mathcal{F} : \alpha_{F_i} = 1$ is assumed). This traffic portion is then distributed among the reliable face(s) in

$t=0.5$	c	$t=0.5$	c	$t=0.75$	c	$t=0.75$	c
F_D	$\begin{pmatrix} 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$	F_D	$\begin{pmatrix} 0 \\ 0 \\ 1/2 \\ 1/2 \end{pmatrix}$	F_D	$\begin{pmatrix} 0 \\ 0 \\ 1/2 \\ 1/2 \end{pmatrix}$	F_D	$\begin{pmatrix} 0 \\ 0 \\ 0.42 \\ 0.58 \end{pmatrix}$
F_0		F_0		F_0		F_0	
F_1		F_1		F_1		F_1	
F_2		F_2		F_2		F_2	

(a) Initial (b) Iteration 1 (c) Iteration 2 (d) Iteration 3

Fig. 3: FWTs of router R for the **first** example.

$t=0.99$	c	$t=0.75$	c	$t=0.75$	c	$t=0.85$	c
F_D	$\begin{pmatrix} 0 \\ 0 \\ 1/3 \\ 2/3 \end{pmatrix}$	F_D	$\begin{pmatrix} 2/9 \\ 4/9 \\ 1/3 \\ 0 \end{pmatrix}$	F_D	$\begin{pmatrix} 0 \\ 0.57 \\ 0.43 \\ 0 \end{pmatrix}$	F_D	$\begin{pmatrix} 0 \\ 2/3 \\ 1/3 \\ 0 \end{pmatrix}$
F_0		F_0		F_0		F_0	
F_1		F_1		F_1		F_1	
F_2		F_2		F_2		F_2	

(a) Initial (b) Iteration 1 (c) Iter. 2 and 3 ($t = 0.85$) (d) Iteration 4

Fig. 4: FWTs of router R for the **second** example.

\mathcal{F}_S (in this case only $\mathcal{F}_S = \{F_0\}$), and since $\sigma_{F_0} > UT_{F_1}$, SAF reaches the optimal FTW as shown in Figure 4d.

IV. EVALUATION

In this section we investigate the performance of SAF by comparing it to related work and standard forwarding algorithms already present in the ns3/ndnSIM 2.0 simulator [17]. The selected algorithms from related work are: ideal Nearest Replica Routing (iNRR) [13], the Request Forwarding Algorithm (RFA) [23], and the On-demand Multi-Path Interest Forwarding (OMP-IF) [25]. As already mentioned in Section II, iNRR uses an oracle to determine for every Interest the nearest node which holds a cached copy of the corresponding Data packet and to provide the shortest path to this node [13]. RFA has been implemented as described in [23]. OMP-IF has been implemented according to the descriptions given in [25]. We provide an open source implementation of SAF together with the implementations of iNRR, RFA, and OMP-IF in ns3/ndnSIM 2.0 at github.com/danposch/SAF. For SAF we use the presented measure \mathcal{M}^T , thus, maximizing the throughput for specific content prefixes at every node in the network.

A. Network Topology Generation

For generating random network topologies we employ the network topology generator BRITE [26]. BRITE was configured to build scale-free networks in a top-down fashion since the Internet topology is likely to be described by power-laws [27]. We model the infrastructure of large ISPs, interconnecting several autonomous systems and access networks. The top level represents $\mu = 5$ autonomous systems (AS). Each AS maintains $\nu = 20$ nodes (bottom level) acting as ICN routers (in total we have $5 \cdot 20 = 100$ ICN routers) and serving as access nodes for later on added client and server nodes. Both the top- and the bottom-level graphs are randomly generated based on the Barabási-Albert model. The model was configured to connect each node with exactly one neighbor. This generates scale-free networks with no redundant paths. As ICN characteristics such as multi-path delivery and link-failure recovery can only be evaluated properly if redundant paths exist, we extended the generated graphs with additional random edges. This was necessary because BRITE only enables to set an integer number of links per node, and therefore does not

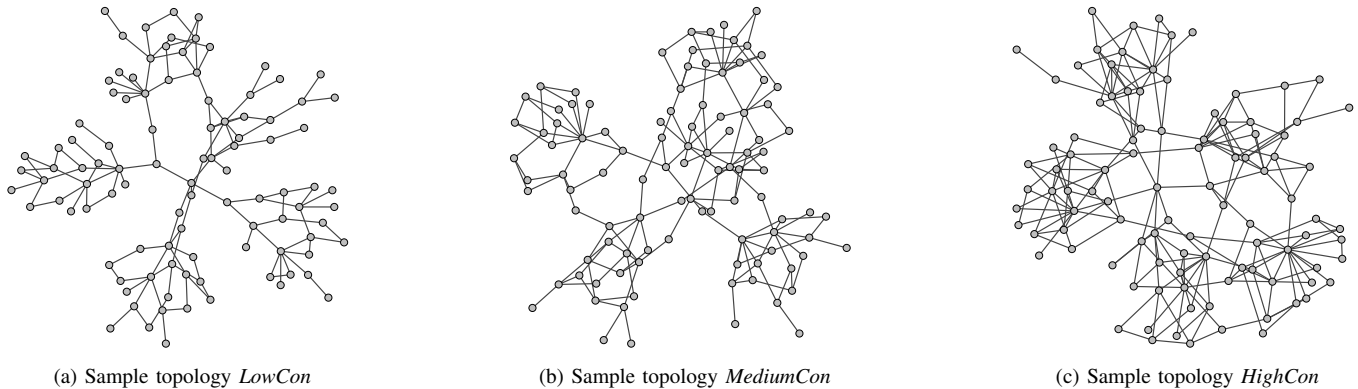


Fig. 5: Sample topologies for the different network **connectivities** as defined in Table II (without client/server nodes).

IDENTIFIER	TOP LEVEL	BOTTOM LEVEL	$C(\mathcal{N})$
LowCon	$\lfloor \mu/2 \rfloor$	$\lfloor \nu/3 \rfloor$	0.0265
MediumCon	μ	$\lfloor \nu/2 \rfloor$	0.0311
HighCon	$\lfloor \mu * 2 \rfloor$	ν	0.0422

TABLE II: Additional edges per connectivity variant.

IDENTIFIER	TOP LEVEL	BOTTOM LEVEL
LowBW	[2, 4] Mbps	[1, 2] Mbps
MediumBW	[3, 5] Mbps	[2, 4] Mbps
HighBW	[4, 6] Mbps	[3, 5] Mbps

TABLE III: Assigned link capacities.

support a “probability-based” link insertion. The alternative, to set the number of neighbours to values greater or equal two, results in already very well-connected networks inhibiting a fine-granular investigation of the influence of redundant paths on the forwarding strategies.

Based on the aforementioned considerations, we generate nine different topology variants. These topology variants differ in graph connectivity and available bandwidth resources. Table II specifies the configured graph connectivities, where the second and third columns specify the number of additionally added edges at the top and at the bottom level, respectively. We define the connectivity $C(\mathcal{N})$ of a network \mathcal{N} as $C(\mathcal{N}) = \frac{1}{(|\mathcal{V}|-1) \cdot |\mathcal{V}|} \cdot \sum_{v \in \mathcal{V}} \text{deg}(v)$, where $\text{deg}(v)$ denotes the (edge) degree of node v . Figure 5 illustrates three sample topologies for the chosen connectivity values. Table III lists the bandwidth resources provided to the links. The capacity of each link is randomly drawn from a uniform distribution limited by the indicated intervals. The nine topology variants arise from the cross product of Table II and Table III. For instance, if we refer to variant *LowConLowBW*, this refers to a topology defined by settings given in the first two lines of Table II and Table III.

B. Scenario Description

We evaluate the mentioned algorithms under two different request scenarios. First, the popularity of content is uniformly distributed among all clients. This leads to the fact that most of the content is concentrated at the caches in the core of the network [28]. Second, we distribute the popularity of content according to a Zipf distribution which has the opposite effect (caches at the edges are utilized) [28]. Before simulations are conducted, we perform a limited parameter investigation in Subsection IV-C regarding the duration of the period (the time between two iterations of Algorithm 2) for SAF, as Theorem III.1 suggests that the duration of the period has significant influence on SAF’s performance.

Given a generated topology we randomly placed $a = 100$ clients and $b = 10$ servers in the network. Clients are configured to request content from a single server with a rate of 30 Interests per second, uniformly distributed during a second. This corresponds to a download rate of approximately 1 Mbps, as a single Interest always requests a 4 kB Data packet. The uniformly distributed request rate represents a steady consumption of content, e.g., streaming video at a certain bit-rate. As already mentioned, for the first scenario we assume a perfect uniform content popularity. A client randomly starts to consume content within the first 30 seconds of the simulation. Each server provides unique content identified by an arbitrary prefix, e.g., */server_id/*. Each node in the network is equipped with a 25 MB large cache which corresponds to approximately 1% of the available content catalogue. We consider this size as sufficient and realistic for the following reason. For instance, consider the *MediumConMediumBW*-scenario. A node in this scenario maintains on average 3.11 links (cf. Table II). The maximum link capacity for this scenario is limited by 5 Mbps (cf. Table III), which leads to an average traffic of $3.11 \cdot 5 \text{ Mbps} = 15.55 \text{ Mbps}$. So, given a node that fully utilizes all links may still cache more than 10 seconds of traffic on average. Considering the findings in [29] that 40% of all cache hits occur in the first 10 seconds, and that clients in the selected scenarios start to request content within a 30 second window, we consider the cache to be sufficiently large. We use the Least Recently Used (LRU) cache replacement strategy [28] and nodes cache every packet they receive.

We further introduce 0, 50, or 100 random link failures during each simulation run. A link failure’s point of occurrence and its duration are distributed uniformly. The duration of a link failure is drawn from the interval of $[0, \lfloor \frac{\text{SimTime}}{10} \rfloor]$, where *SimTime* denotes the duration of a simulation run. Nodes are configured such that they know all possible routes to any content server at the beginning of each simulation run. This is necessary as many of SAF’s competitors require

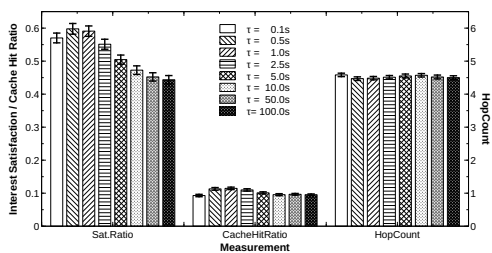


Fig. 6: Influence of different period durations (τ) on SAF, given scenario *MediumBWMediumCon* with 50 link failures.

that this information is provided by the routing layer. In this experiment, SAF uses the routing information only as the starting point for the initial FWT. During the simulation no routing updates are enforced, which puts the responsibility to deal with short-term topology changes to the forwarding strategies. We simulate 1800 seconds of network traffic using the aforementioned parameters and conduct 50 runs per setting.

C. SAF: Influence of the Period on the Performance

In order to investigate the influence of different values for the duration of a period (τ), we considered the scenario *MediumConMediumBW* with 50 link failures given uniform content popularity. Figure 6 depicts the 95% confidence intervals of the average Interest satisfaction ratio, the cache hit ratio and the hop count for $\tau \in \{0.1, 0.5, 1.0, 2.5, 5.0, 10.0, 50.0, 100.0\}$ seconds. The Interest satisfaction ratio denotes the ratio between received Data packets and generated Interests by all clients. The cache hit ratio is averaged over all network nodes (clients and servers maintain no cache). The hop count provides the number of links a Data packet traversed to satisfy an issued Interest by a client considering cache hits.

One can observe in Figure 6 that in general shorter periods increase the performance regarding the Interest satisfaction ratio. This is consistent with Theorem III.1, which provides the number of steps for SAF to converge to the optimum (shorter periods, faster steps). However, one can also see that a too small period, e.g. $\tau = 0.1s$, may have a negative influence on the performance. This is due to the fact that the observed traffic within extremely short periods is not representative enough to deduce a suitable decision. The cache hit ratio varies only slightly considering different period durations. Nevertheless, one can observe that the highest ratios are achieved with period durations of few seconds. The average hop count does not change significantly, regardless of the selected period. We use $\tau = 1.0$ seconds for all further evaluations, as this setting achieves a good performance with acceptable computational overhead (cf. Section III).

D. Performance under Uniform Content Popularity

Figures 7, 8, and 9 depict the 95% CI of the Interest satisfaction ratio for each pairing (topology variant, forwarding strategy) having 0, 50, or 100 random link failures during each simulation. It is evident that SAF outperforms its competitors in nearly all simulation scenarios in terms of Interests satisfied, regardless of the number of link failures, the connectivity of the underlying network and/or network topology. The two

STRATEGY	LOWBW		MEDIUMBW		HIGHBW	
Broadcast	0.024	0.133	0.097	0.242	0.195	0.333
NCC	0.033	0.155	0.102	0.229	0.181	0.287
ShortestRoute	0.012	0.098	0.065	0.188	0.135	0.247
RFA	0.060	0.218	0.109	0.267	0.111	0.263
iNRR	0.016	0.107	0.112	0.226	0.204	0.263
OMP-IF	0.045	0.177	0.135	0.233	0.252	0.298
SAF	0.090	0.240	0.253	0.334	0.288	0.320

TABLE IV: **Actual** (first value) and **relative** (second value) performance loss comparing Figures 7b and 9b.

strongest competitors to SAF are iNRR and OMP-IF, especially in scenarios with more resources available. SAF's good performance considering all scenario settings can be explained by its ability to: *i*) smartly circumvent congested nodes and link failures by taking detours into account; *ii*) prevent further transmission of Interests on congested links by redirecting them to the virtual dropping face; *iii*) discover paths to cached content that are not indicated by the FIB. This is also reflected by Figure 10 depicting the cache hit ratio for the scenarios with zero link failures. In most of the cases SAF outperforms the other algorithms, except for iNRR, which has perfect knowledge about the content chunks in the individual caches. Figure 11 illustrates the average hop count per satisfied Interest for the scenarios with zero link failures. As expected, SAF maintains a higher hop count than some of the other algorithms due to the detours it takes for maximizing the Interest satisfaction according to \mathcal{M}^T . The very low hop count of Broadcast and NCC show that both algorithms are able to obtain nearby replicas, however, due to their poor Interest satisfaction and cache hit ratio the low hop count is valueless.

An increasing number of link failures has a significant negative impact on the Interest satisfaction ratio for all algorithms. For instance, consider Figure 7b and 9b illustrating the results for 0 and 100 link failures with *medium* graph connectivity. Table IV depicts the actual and relative performance loss for each strategy. SAF has the highest actual performance loss, however, the relative loss is in a similar range as for its competitors. In these scenarios SAF is not able to fully circumvent link failures due to the resource shortages that are inherent in the selected scenarios. iNRR also suffers from link failures as the oracle used for finding cached replicas does only consider the distance to the content and does not consider link attributes such as reliability or capacity. Interestingly, ShortestRoute maintains the smallest actual and relative performance loss. First, these numbers must be considered with respect to the absolute number of satisfied Interests, and second, this can be explained due to the rather short paths used to satisfy Interests (cf. Figure 11b), which of course have a lower chance of being affected by randomly emerging link failures. As will be shown later, Broadcast and NCC cannot take advantage of their low hop counts due to extensive Interest replication (cf. Figure 15).

Especially if the network resources and the connectivity are very limited, RFA is the closest competitor to SAF considering the Interest satisfaction ratio. This is due to the extensive use of all provided routes to the content origins by RFA. Thus, it is able to distribute the load equally on congested nodes/paths. Nevertheless, in better connected scenarios and with higher

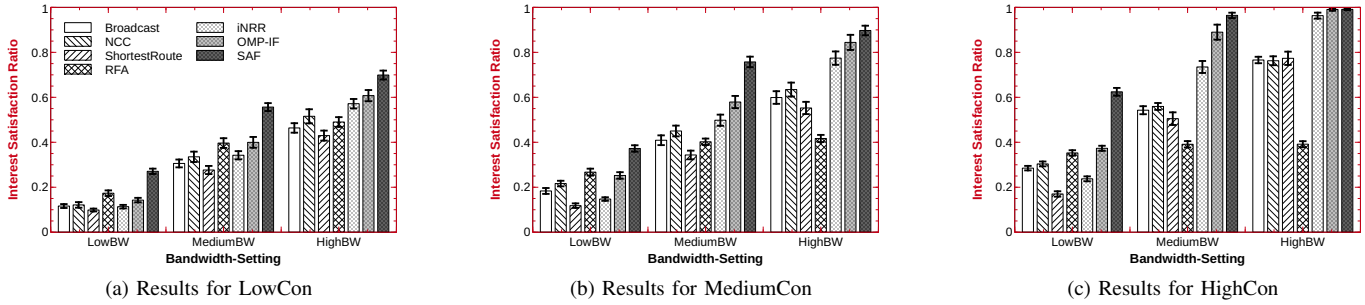


Fig. 7: Average **Interest satisfaction ratio** and 95% CI with **0 link failures** per simulation run (higher is better) [uniform].

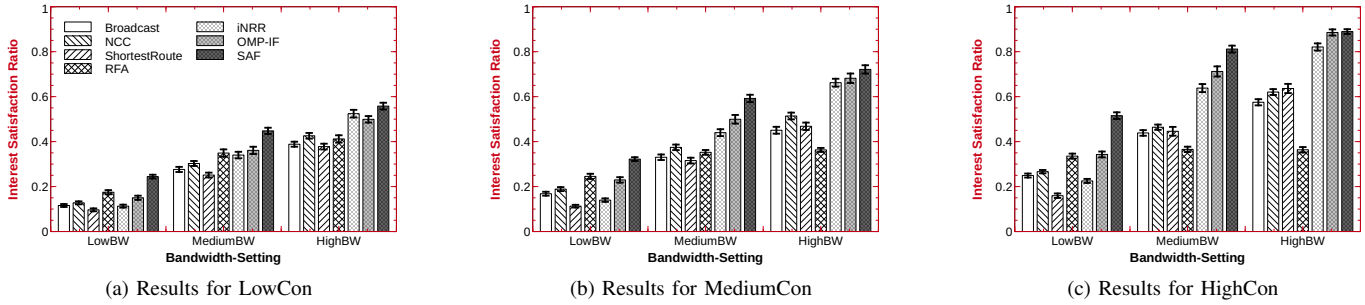


Fig. 8: Average **Interest satisfaction ratio** and 95% CI with **50 link failures** per simulation run (higher is better) [uniform].

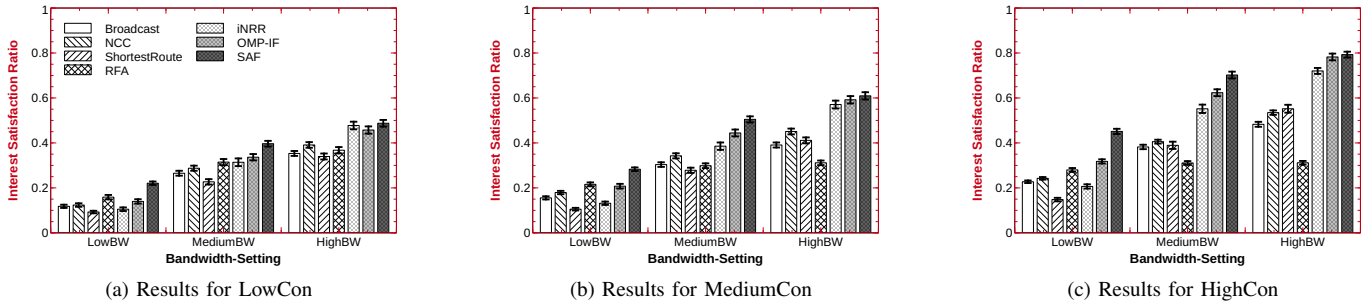


Fig. 9: Average **Interest satisfaction ratio** and 95% CI with **100 link failures** per simulation run (higher is better) [uniform].

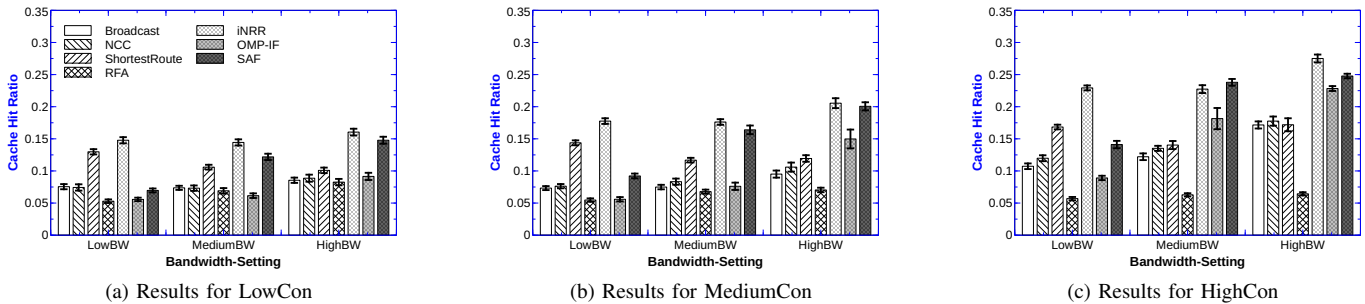


Fig. 10: Average **cache hit ratio** and 95% CI in the network with **0 link failures** per simulation run (higher is better) [uniform].

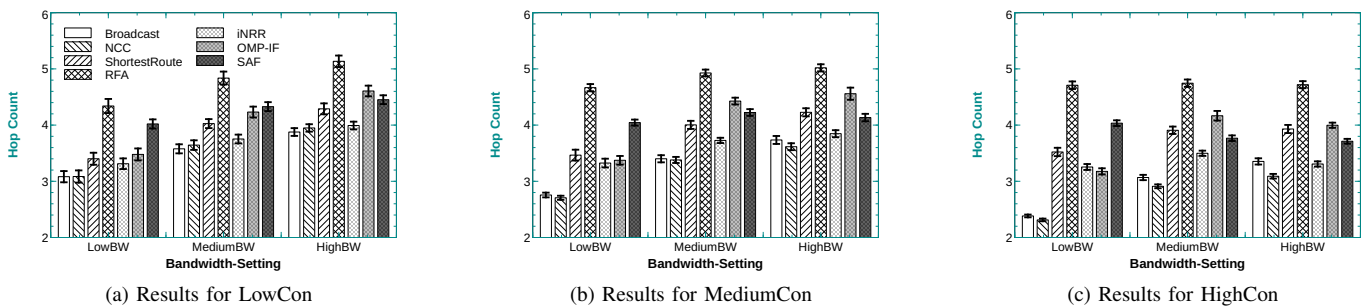


Fig. 11: Average **hop count** per *satisfied* Interest and 95% CI with **0 link failures** per simulation run (lower is better) [uniform].

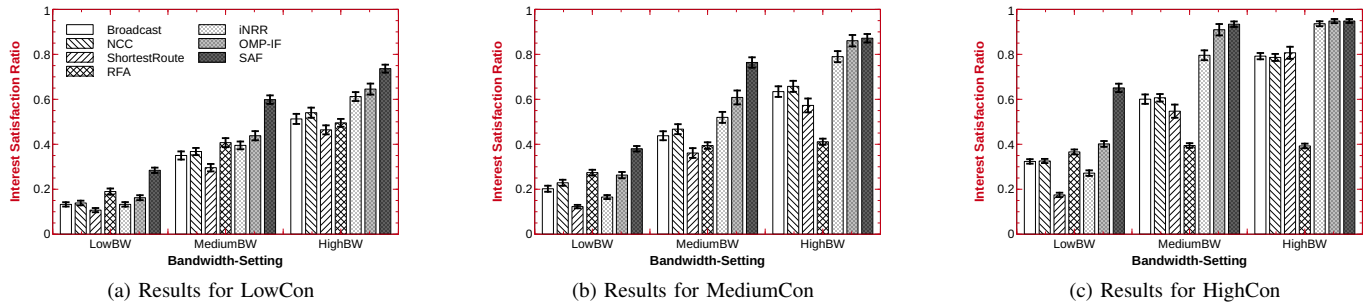


Fig. 12: Average **Interest satisfaction ratio** and 95% CI with **0 link failures** per simulation run (higher is better) [zipf].

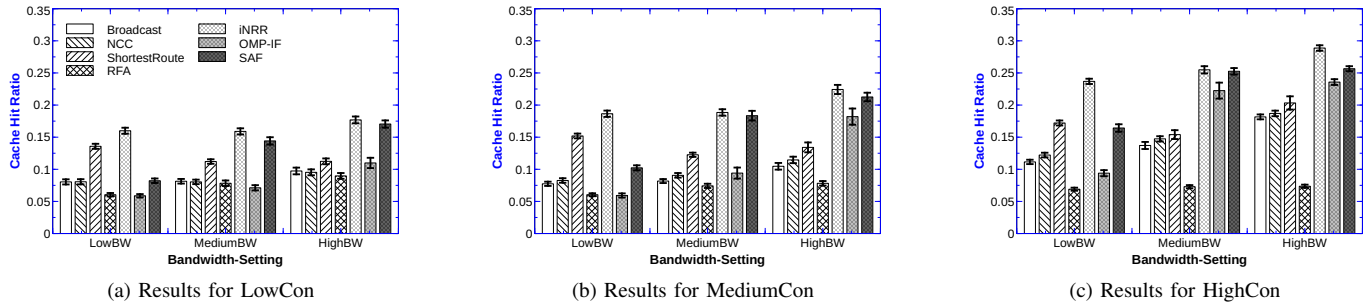


Fig. 13: Average **cache hit ratio** and 95% CI in the network with **0 link failures** per simulation run (higher is better) [zipf].

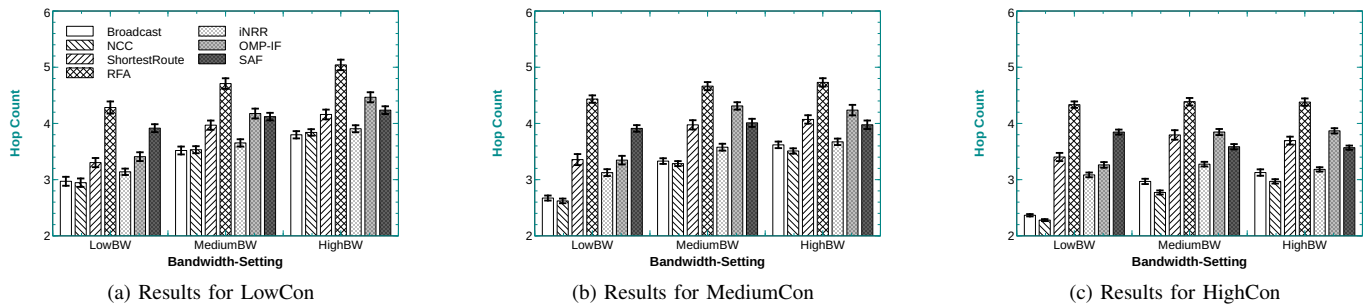


Fig. 14: Average **hop count** per *satisfied* Interest and 95% CI with **0 link failures** per simulation run (lower is better) [zipf].

network resources RFA is not anymore able to reach the performance of the *standard algorithms* (Broadcast, NCC, ShortestRoute). RFA uses the number of pending Interests on a face (the one with the lowest number pending is selected) for deciding to which one of the faces in the FIB an Interest shall be forwarded. It does not store any *long term* state or classification of a face. Therefore, the more routes are available to RFA and the higher the network resources, the less RFA is able to determine which faces are the most appropriate for specific content prefixes. For instance, this can be observed in Figure 7c. This is also reflected by the low cache hit ratio and high hop count shown in Figures 10 and 11 indicating that RFA is not able to select those faces that provide good service (e.g., cache hits), but rather takes unnecessary detours.

In scenarios with plenty resources available, iNRR and OMP-IF are able to catch up with SAF. Figure 10 shows that iNRR achieves the highest cache hit ratio, which also is an explanation for iNRR's good performance. The higher hop count compared to Broadcast and NCC (cf. Figure 11), can be explained by the deletion of cache entries and the resulting detours iNRR imposes on Interests. In order to understand iNRR's behavior, suppose three nodes $a, b, c \in \mathcal{V}$, whereof

c is the content origin. Assume that a receives an Interest and searches for caches that can satisfy it. The nearest cache is b satisfying the aforementioned constraints (cf. Section II). Therefore, a forwards the Interest to b , but at arrival or on the way to b , b has already evicted the desired content. Thus, the Interest is forwarded to c . Therefore, the triangle inequality ($d(a, c) \leq d(a, b) + d(b, c)$) explains the higher hop count of iNRR compared to Broadcast and NCC. This leads to the assumption that larger caches will improve iNRR's performance.

The strong performance of OMP-IF in scenarios with many resources can be explained by its principle of node disjointness. Subsequent Interests requesting the same content will be forwarded with high probability on the same paths resulting in a high cache hit ratio (cf. Figure 10c). Furthermore, the usage of probing Interests (which are broadcasted on all faces in the FIB to identify the best performing path) performs well in these scenarios. However, as resources become scarce (cf. Figure 7b) these probing Interests amplify congestion. This leads to the fact that the used paths are switched frequently impairing the cache hit ratio (cf. Figures 10b and 10c) and the overall performance significantly.

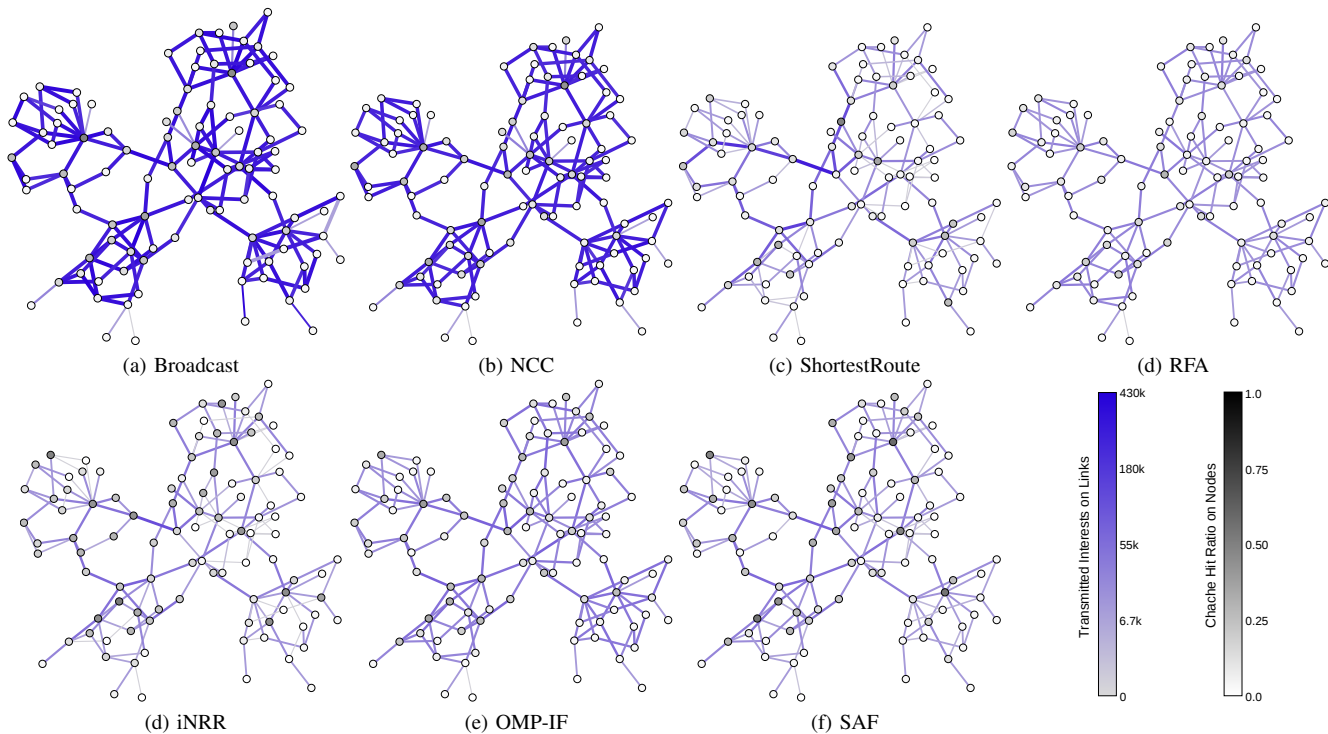


Fig. 15: Heat maps illustrating the distribution of the transmitted Interests and cache hits. The wider and the more saturated a link, the more Interests are transmitted over the given link. The darker the coloring of a node, the higher the cache hit ratio.

E. Performance under Zipf-like Content Popularity

In order to compare the results using a uniform content popularity and the results using a Zipf distribution for popularity, we use the same network topologies and simulation parameters as for the previous evaluation (cf. Section IV-D). Only the distribution of the content popularity was changed to a Zipf distribution with $\alpha = 0.668$ according to [30]. We again conduct 50 simulation runs for each scenario. In Section IV-D we presented the results for scenarios with 0, 50, and 100 link failures. It can be observed that the performance of the algorithms decreases proportionally with an increase in link failures (cf. Figures 7, 8 and 9). Therefore, we omit the results for 50 and 100 link failures for the following discussion. Figures 12, 13, and 14 depict the average Interest satisfaction ratio per node, the average cache hit ratio, and the average hop count per satisfied Interest, respectively. The Interest satisfaction ratio for Zipf-distributed content popularity (cf. Figure 12) paints a picture similar to Figure 7 for uniform content popularity. SAF outperforms its competitors in terms of satisfied Interests. Again iNRR and OMP-IF are the strongest competitors to SAF and in well-connected scenarios with many resources available these algorithms achieve similar performance. As expected, iNRR beats SAF in terms of cache hit ratio since iNRR is designed to forward Interests to the nearest cache holding the desired Data packet. Broadcast and NCC again provide the lowest average hop count (cf. Figure 14) outperforming the other algorithms regarding this metric for the very same reasons as outlined in Section IV-D. Although iNRR always maintains a lower hop count and higher cache hit ratio than SAF, SAF outperforms iNRR regarding the number of satisfied Interests. This is due

to the extensive usage of multi-path forwarding considering paths which may not provide optimal performance regarding cache hit ratio or hop count, however, maximize the number of satisfied Interests. OMP-IF also focuses on the usage of multiple but node-disjoint paths per content. Nevertheless, as can be seen from Figures 13 and 14 in most of the cases it maintains a higher hop count and lower cache hit ratio than SAF resulting in a smaller number of overall satisfied Interests.

Figure 15 depicts the distribution of the transmitted Interests and cache hits in the network. For each of the evaluated algorithms, a single simulation run is depicted considering a fixed scenario (MediumCon, HighBW; cf. Figure 5b) with a Zipf-distributed content popularity. This figure illustrates how differently the forwarding algorithms behave. Broadcast replicates Interests on every node and pushes them to all neighboring nodes (cf. Figure 15a). NCC shows a similar behavior. ShortestRoute focuses on the shortest paths from a client to the content provider and, therefore, the caches on the shortest paths are heavily used. RFA performs a kind of load balancing distributing Interests equally on all routes. Thus, it achieves a low cache hit ratio and Interest satisfaction ratio (cf. Figure 12). iNRR tries to maximize the cache hit ratio by explicitly preferring nearby cached Data packets in the network instead of retrieving them from the content origin. This circumstance is reflected in Figure 15d showing several nodes with high cache hit ratios (dark coloring). OMP-IF also achieves an acceptable cache hit ratio due to its principle of node disjointness. Frequent probing and path switching leads to a similar load balancing behavior as in RFA. SAF is able to provide both, effective load balancing and high cache hit ratios at the relevant nodes (cf. Figure 15f). Compared to OMP-IF,

which balances the load very equally on all available links, SAF focuses on paths that provide better performance, leading to a better cache hit ratio and an overall better performance.

V. CONCLUSION AND FUTURE WORK

This paper introduced Stochastic Adaptive Forwarding (SAF), a novel forwarding strategy for Named Data Networking. SAF provides probability-based forwarding on a per-content/per-prefix basis. It ensures effective forwarding with incomplete and/or invalid routing information, and resolves unexpected network topology changes without relying on the routing plane. The extensive usage of multi-path transmission is the foundation for SAF's performance. SAF is flexible in that it can be configured with various measures defining the forwarding objectives. The effectiveness of SAF was illustrated by conducting extensive simulations using the ns-3/ndnSIM framework. We presented the throughput-based measure \mathcal{M}^T optimizing the Interest satisfaction ratio in a given network. Simulations show that SAF is able to outperform existing algorithms. Our results can be reproduced as we provide an open source implementation of SAF and its competitors (iNRR, OMP-IF, RFA) at github.com/danposch/SAF.

SAF allows to optimize its forwarding behavior with respect to a measure. The only assumption that SAF has on the measure is that it allows classifying Interests into satisfied and unsatisfied ones. Thus, SAF can be easily tailored to specific use cases. For example, if Interests shall not exceed a specific hop count, one may introduce a measure that classifies Interests as satisfied if the hop count is below this specific threshold. SAF further allows to make decisions on a context and content level. Since SAF keeps track of the forwarding probabilities for all content prefixes, it is possible to adapt the forwarding probabilities with respect to content prefixes. This can be used to prioritize traffic from specific content prefixes. For instance, (real-time) multimedia traffic that takes the same path through the network as bulk file transfer can be assigned higher priority. In this case, one may introduce a weighting such that the probability of the virtual dropping face (F_D , cf. Section III-A) for bulk file transfer is increased. This leads to a decrease of the forwarding probability of F_D for multimedia traffic and, therefore, to an increase of the forwarding probability on the desired faces. We plan to design and investigate algorithms for content- and context-aware forwarding in our future work.

ACKNOWLEDGMENT

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. We are grateful for the feedback from the anonymous reviewers, which greatly improved the article.

REFERENCES

- [1] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," *SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 175–187, Aug. 2000.
- [2] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The Impact of Internet Policy and Topology on Delayed Routing Convergence," in *Proceedings of 20th Joint Conference of the IEEE Computer and Communications Societies*, 2001, pp. 537–546.
- [3] S. Kini, S. Ramasubramanian, A. Kvalbein, and A. Hansen, "Fast Recovery from Dual Link Failures in IP Networks," in *Proc. of the 28th IEEE INFOCOM*, April 2009, pp. 1368–1376.
- [4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.
- [5] 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 Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.
- [6] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of Information (NetInf) - An Information-Centric Networking Architecture," *Computer Comm.*, vol. 36, no. 7, 2013.
- [7] N. Fotiou, D. Trossen, and G. Polyzos, "Illustrating a Publish-Subscribe Internet Architecture," *Telecomm. Systems*, vol. 51, no. 4, 2012.
- [8] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," in *Proc. of the 5th ACM Int. Conf. on Emerging Networking Experiments & Technologies*, 2009.
- [9] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-oriented (and Beyond) Network Architecture," *SIGCOMM Comp. Comm. Rev.*, vol. 37, no. 4, 2007.
- [10] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM SIGCOMM Comp. Comm. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.
- [11] 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.
- [12] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "On the Role of Routing in Named Data Networking," in *Proc. of the 1st Int. Conference on Information-Centric Networking*. ACM, 2014.
- [13] G. Rossini and D. Rossi, "Coupling Caching and Forwarding: Benefits, Analysis, and Implementation," in *Proc. of the 1st Int. Conference on Information-Centric Networking*. ACM, 2014, pp. 127–136.
- [14] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, and D. Leong, "VIP: A Framework for Joint Dynamic Forwarding and Caching in Named Data Networking," in *Proc. of the 1st Int. Conference on ICN*. ACM, 2014.
- [15] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC 3272, IETF, 2002.
- [16] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive Forwarding in Named Data Networking," *SIGCOMM Computer Communication Review*, vol. 42, no. 3, pp. 62–67, Jun. 2012.
- [17] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN Simulator for NS-3," Tech. Report. NDN-0005, 2012, University of California, LA.
- [18] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new Version of the NDN Simulator for NS-3," Tech. Report NDN-0028, 2015, <http://named-data.net/publications/techreports/>.
- [19] A. Afanasyev, J. Shi, B. Zhang, L. Zhang *et al.*, "NFD Developer's Guide," Tech. Report NDN-0021, 2014, University of California, LA.
- [20] E. Rosensweig, J. Kurose, and D. Towsley, "Approximate Models for General Cache Networks," in *Proc. of the 29th IEEE INFOCOM*, 2010.
- [21] R. Chiocchetti, D. Rossi, and G. Rossini, "ccnSim: An Highly Scalable CCN Simulator," in *Proc. of IEEE ICC*, 2013, pp. 2309–2314.
- [22] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini, "INFORM: A Dynamic Interest Forwarding Mechanism for ICN," in *Proc. of the 3rd ACM SIGCOMM Workshop on ICN*, 2013, pp. 9–14.
- [23] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang, "Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks," in *Proceedings of 21st IEEE International Conference on Network Protocols (ICNP)*, Oct 2013.
- [24] H. Qian, R. Ravindran, G.-Q. Wang, and D. Medhi, "Probability-based Adaptive Forwarding Strategy in Named Data Networking," in *IFIP/IEEE Int. Symposium on Integrated Network Management*, 2013.
- [25] A. Udugama, X. Zhang, K. Kuladinithi, and C. Goerg, "An On-demand Multi-Path Interest Forwarding Strategy for Content Retrievals in CCN," in *Proceedings of IEEE NOMS*, 2014.
- [26] A. Medina, I. Matta, and J. Byers, "BRIT: A Flexible Generator of Internet Topologies," 2000, www.cs.bu.edu/brite/ (Boston University).
- [27] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-law Relationships of the Internet Topology," *SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 251–262, Aug. 1999.
- [28] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Optimal Cache Allocation for CCN," in *Proceeding of 21st IEEE ICNP*, 2013.
- [29] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in Network Traffic: Findings and Implications," *SIGMETRICS Performance Evaluation Rev.*, vol. 37, no. 1, pp. 37–48, 2009.
- [30] X. Cheng, C. Dale, and J. Liu, "Statistics and Social Network of YouTube Videos," in *Proc. of the 16th Int. Workshop on QoS*, 2008.