A Fault-Tolerant Forwarding Strategy for Interest-based Information Centric Networks

J.J. Garcia-Luna-Aceves*†
*Palo Alto Research Center, Palo Alto, CA 94304
†Department of Computer Engineering, University of California, Santa Cruz, CA 95064
Email: jj@soe.ucsc.edu

***Abstract*—We show that the forwarding strategies in the named data networking (NDN) architecture and the original content centric networking (CCN) architecture cannot ensure that Interests return the requested data objects when routing-table loops exist in a stable or dynamic network. We also show that no correct Interest forwarding strategy that allows Interest aggregation can be designed solely on the basis of identifying Interests uniquely in order to detect Interest loops. We introduce SIFAH (Strategy for Interest Forwarding and Aggregation with Hop-Counts). SIFAH prevents or detects Interest loops when Interests are aggregated or forwarded over one or multiple paths. As a result, it is far more efficient than the forwarding strategy in NDN and the original CCN proposal. SIFAH operates by having forwarding information bases (FIB) store the next hops and number of hops to named content prefixes, and by using Interests that state the names of requested content and hop counts that reflect the information in their FIBs.**

I. Introduction

Several information-centric networking (ICN) architectures have been proposed to improve the performance and the end-user experience of the Internet [1], [19]. ICN architectures focus on (1) enabling access to content and services by name, rather than by original location, (2) protecting content rather than links or connections, and (3) exploiting in-network storage of content. Section II summarizes prior work on Interest-based ICN architectures.

Interest-based ICN is the most popular ICN approach today, and consists of: populating forwarding information bases (FIB) of routers with routes to name prefixes denoting content, sending content requests (called Interests) for specific named data objects (NDO) over paths implied by the FIBs, and delivering content along the reverse paths traversed by Interests.

Named Data Networking (NDN) [14] and Content Centric Networking (CCN) [3] are the best-known examples of this approach. Section III summarizes the basic operation of the original CCN and NDN Interest forwarding strategies. The developers of these architectures [11], [14], [20], [21] have argued that Interests stating a name of requested content and a nonce or unique identifier can be forwarded correctly towards an intended node advertising a name prefix covering the content name, that routers can aggregate Interests so that a router can forward an Interest for the same content only once, and that Interest loops are detected whenever they occur.

ISBN 978-3-901882-68-5 © 2015 IFIP

Surprisingly, however, no prior work has been reported proving any of the above claims made for NDN and CCN. Section IV demonstrates that the forwarding strategy of NDN [20], [23] does not work correctly, in that some Interests may never return data objects to the consumers who issued the Interests, even if the content exists in the network, the network topology and routing state are stable, and all transmissions are successful. More importantly, it is also shown that there is no correct forwarding strategy with Interest aggregation and Interest-loop detection based on such Interest-identification data carried in Interests as nonces, unique identifiers, or a combination of nonces and the path traversed by an Interest.

Section V introduces the Strategy for Interest Forwarding and Aggregation with Hop-counts (SIFAH). SIFAH is the first forwarding strategy for Interest-based ICNs shown to be correct, and operates by having FIBs store the next hops *and* number of hops to named content, and by forwarding each Interest based on the name of the requested content and a hop count from the forwarding router to the requested content. A router forwards an Interest only if according to its FIB the hop count stated in the Interest is larger than the hop count from the router to the content through some of its neighbors. Similarly, a router that is waiting for an NDO after forwarding an Interest aggregates another Interest received for the same NDO only if the hop count stated in the Interest is larger than the hop count of the Interest sent by the router.

Section VI proves that SIFAH prevents or detects Interest loops even if Interests are aggregated or forwarded over multiple paths. Section VII shows that SIFAH is a far more desirable approach than the NDN approach, because it requires substantially less storage and can reduce latencies.

II. Related Work

Ahlgren et al. [1], and Xylomenos et al. [19] present overviews of the various ICN architectures proposed recently.

Directed Diffusion [10] is one of the first examples of an Interest-based ICN architecture. Requests for named content (called Interests) are diffused throughout a sensor network, and data matching the interests are sent back to the issuers of interests. Subsequent proposals for information-centric disruption-tolerant networking use similar approaches (e.g., DIRECT [16], ICEMAN [18]). Nodes use opportunistic caching of content at every router and flood interests persistently. Depending on the proposal, Interests may state the name of the requested content, a name and a publisher, or a set of attributes describing the content. In all these approaches, Interests are flooded in the network in ways that ensure their dissemination even when the network topology is partitioned. While this Interest-based approach to content dissemination works in connected and disconnected wireless ad hoc networks, it cannot be applied at Internet scale.

The original CCN proposal [11] was the first example of an Interest-based ICN architecture applicable to wired networks in which Interests do not state the identity of the sender. Today, NDN [14] and CCNx [3] are the leading proposals for Interest-based ICN. In general, the forwarding strategy of an Interest-based ICN architecture consists of the creation and forwarding of Interests requesting named content towards nodes that have advertised such content. Such forwarding is based on the information stored in FIBs. Named data objects or NDOs (also called Content Objects) are sent back to the nodes who issue Interests in the reverse direction of the paths traversed by the Interests.

III. Existing Forwarding Strategies for Interest-based ICNs

A router r uses three primary data structures to implement any of the forwarding strategies defined for Interest-based ICNs: (a) a forwarding information base (FIB^r), (b) a pending interest table (PIT^r), and (c) a content store (CS^r). The forwarding strategy determines the interaction among FIB^r, PIT^r, and CS^r needed to forward Interests towards nodes advertising having copies of requested content, send NDOs back to consumers who requested them over reverse paths traversed by Interests, and send any other signal indicating the inability to satisfy an Interest.

FIB^r is used to route incoming interests to the appropriate next hops towards the desired content producer advertising a content prefix name $n(j)^*$. FIB^r is populated using content routing protocols or static routes and matches Interest names stating a specific NDO $n(j)$ to FIB^r entries of prefix names using *longest prefix match*. PIT^r serves as a cache of Interest state, such that content objects that satisfy Interests may follow the reverse Interest path back to the original requester. CS^r is a cache for content objects.

We use the term neighbor instead of interface or face. We denote the name of NDO j by $n(j)$, and the name prefix that includes that NDO name by $n(j)^*$. We denote the existence of an entry for a prefix $n(j)^*$ or NDO with name $n(j)$ in the FIB, PIT or CS of router i by $n(j)^* \in FIB^i$, $n(j) \in PIT^i$, and $n(j) \in CS^i$, respectively.

The forwarding strategies proposed to date for Interest-based ICN architectures are the original CCN strategy [11] and the NDN forwarding strategy [20], [23]. In both strategies, an Interest created by source s for NDO j states $n(j)$ and a nonce $id_j(s)$. The pair $(n(j), id_j(s))$ is used to denote an Interest uniquely with a large-enough probability, and to detect whether an Interest is traversing a loop.

In the context of NDN, we use $I[n(j), id_j(s)]$ to denote an Interest that requests NDO with name $n(j)$ and that is originated by consumer s, who assigns nonce $id_j(s)$ to the Interest. A content-object message sent in response to an Interest $I[n(j), id_j(s)]$, denoted $D[n(j), id_j(s), sig(j)]$, states the name and nonce of the Interest, a signature payload $sig(j)$ used to validate the content object, and the object itself.

The entry in FIB^i for name prefix $n(j)^*$ is denoted by $FIB^i_{n(j)^*}$ and consists of $n(j)^*$ and the list of neighbors that can be used to reach the prefix. If neighbor k is listed in $FIB^i_{n(j)^*}$, then we state $k \in FIB^i_{n(j)^*}$. In NDN [21], the FIB entry for a prefix also contains a stale time after which the entry could be deleted; the round-trip time through the neighbor; a rate limit; and status information stating whether it is known or unknown that the neighbor can bring data back, or is known that the neighbor cannot bring data back.

The entry in PIT^i for NDO with name $n(j)$ is denoted by $PI^i_{n(j)}$ and consists of a vector of one or multiple tuples, one for each nonce processed for the same NDO name. The tuple for a given NDO states the nonce used, the incoming and the outgoing neighbor(s). The tuple created as a result of processing Interest $I[n(j), id_j(s)]$ received from k and forwarded to a set of neighbors $OUTSET$ is denoted by $PI^i_{n(j)}[id_j(s), in : k, out : OUTSET]$, and the set of outgoing neighbors in $PI^i_{n(j)}$ is denoted by $OUTSET(PI^i_{n(j)})$.

Each PIT entry $PI^i_{n(j)}[id_j(s), in : k, out : OUTSET]$ has a lifetime, which should be larger than the estimated round-trip time to a site where the requested NDO can be found.

The NDN forwarding strategy augments the original CCN strategy by introducing negative acknowledgements (NACK) sent in response to Interests for a number of reasons, including: routers identifying congestion, routers not having routes in their FIBs to the requested content, or Interest loops being detected. We denote by $NI[n(j), id_j(s), \mathsf{CODE}]$ the NACK sent in response to $I[n(j), id_j(s)]$, where CODE states the reason why the NACK is sent. The NDN forwarding strategy also differs from the original CCN strategy in other ways related to the retransmission of Interests and the use of multiple paths towards content.

Algorithms 1 and 2 illustrate the NDN Interest processing approach [20], [21] using the notation we have introduced, and correspond to the Interest-processing and forwarding-strategy algorithms in [21]. Algorithm 2 does not include the probing of neighbors proposed in NDN, given that this aspect of NDN is still being defined [21]. Routers forward NACKs received from those neighbors to whom they sent Interests, unless the PIT entries have expired or do not match the information provided in the NACKs.

The use of nonces in NDN and the original CCN approach can be extrapolated to include the case in which the Interest states a nonce and the path traversed by the Interest by assuming that $id_j(s)$ equals the tuple $(id_j(s)[nonce], id_j(s)[path])$. If a nonce and path traversed by the Interest are used, deciding whether an Interest has not traversed a loop can be based on whether $id_j(x)[nonce] \neq id_j(s)[nonce] \vee i \notin id_j(s)[path]$. However, including path information in Interests reveals the identity of originators of Interests. The key aspect of the forwarding strategies that have been proposed for Interest-based ICN architectures to date is that a router determines whether or not an Interest is a duplicate Interest based solely on the content name and Interest-identification data for the Interest (a nonce in the case of NDN). This is done in Line 11 of Algorithm 1. The following section analyzes the correctness of this approach.

Algorithm 1 NDN Processing of Interest at router i

```
1: function Process Interest
2: INPUT: $PIT^i$, $CS^i$, $FIB^i$;
3: INPUT: $I[n(j), id_j(s)]$ received from $k$;
4: if $n(j) \in CS^i$ then
5:     send $D[n(j), id_j(s), sig(j)]$ to $k$
6: else
7:     if $n(j) \notin PIT^i$ then
8:         create $PI^i_{n(j)}[id_j(s), in : k, out : \emptyset]$;
           call Forwarding Strategy($PI^i_{n(j)}$)
9:     else
10:        % There is a PIT entry for $n(j)$
11:        if $\exists\ PI^i_{n(j)}[id_j(x)]$ with $id_j(x) = id_j(s)$ then
12:            % A duplicate Interest is detected
               send $NI[n(j), id_j(s), \text{duplicate}]$ to $k$; drop $I[n(j), id_j(s)]$
13:        else
14:            % Interest can be aggregated
               create $PI^i_{n(j)}[id_j(s), in : k, out : \emptyset]$;
15:            if $RT_i(I[n(j), id_j(s)])$ is expired then
16:                call Forwarding Strategy($PI^i_{n(j)}$);
17:            end if
18:        end if
19:    end if
20: end if
```

Algorithm 2 NDN forwarding of Interest at router i

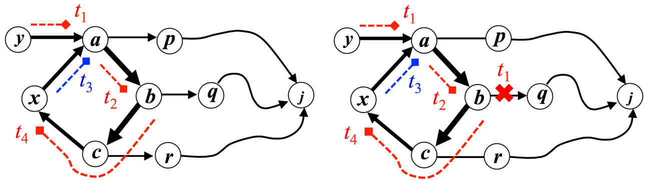
```
function Forwarding Strategy
INPUT: $PIT^i$, $CS^i$, $FIB^i$;
INPUT: $PI^i_{n(j)}[id_j(s), in : k, out : OUTSET]$
if $n(j)^* \in FIB^i$ then
    for each neighbor $m$ in $FIB^i_{n(j)^*}$ by rank do
        if $m \neq in : k$ for all $in : k \in PI^i_{n(j)} \wedge$
          $m \notin SET$ for all $out : SET \in PI^i_{n(j)}$ then
            if $m$ is available then
                $OUTSET(PI^i_{n(j)}) = OUTSET(PI^i_{n(j)}) \cup m$;
                start $RT_i(I[n(j), id_j(s)])$;
                forward $I[n(j), id_j(s)]$ to neighbor $m$;
                return
            end if
        end if
    end for
    send $NI[n(j), id_j(s), \text{congestion}]$ to $k$;
    drop $I[n(j), id_j(s)]$; delete $PI^i_{n(j)}$
else
    send $NI[n(j),\ id_j(s), \text{no data}]$ to $k$;
    drop $I[n(j), id_j(s)]$; delete $PI^i_{n(j)}$
end if
```

IV. Incorrect Operation of The NDN Forwarding Strategy

To discuss the correctness of forwarding strategies, we define an Interest loop as follows.

Interest Loop: An Interest loop of h hops for NDO with name $n(j)$ occurs when one or more Interests asking for $n(j)$ are forwarded and aggregated by routers along a cycle $L = \{v_1, v_2, ..., v_h, v_1\}$ such that router v_k receives an Interest for NDO $n(j)$ from v_{k-1} while waiting for a response to the Interest it has forwarded to v_{k+1} for the same NDO, with $1 \leq k \leq h$, $v_{h+1} = v_1$ and $v_0 = v_h$. ■

According to the NDN forwarding strategy, a router can select a neighbor to forward an Interest if it is known that it can bring content and its performance is ranked higher than other neighbors that can also bring content. The ranking of neighbors is done by a router independently of other routers, which can result in long-term routing loops implied by the FIBs if the routing protocol used in the control plane does not provide loop-free multi-path routing.



(a) Long-term loop (b) Temporary loop

Fig. 1. Interest looping and aggregation in NDN

Figure 1 illustrates Interest looping in NDN. Arrowheads in the figure indicate the next hops to content advertised by router j according to the FIB entries stored in routers. Thick lines indicate that the perceived performance of a neighbor is better than neighbors shown with thinner lines. Dashed lines indicate the traversal of Interests over links and paths. The time when an event is processed at a router is indicated by t_i. Figure 1(a) shows the case of a long-term Interest loop formed because the multi-paths implied in FIBs are not loop-free, even though all routing tables are consistent. Figure 1(b) shows the case of a temporary Interest loop when single-path routing is used and FIBs are inconsistent due to a topology change at time t_1 (link (b, q) fails). In both cases, router a aggregates the Interest from x at time t_3, router x aggregates the Interest from c at time t_4, and the combined steps preclude the detection of Interest looping. This results in routers x and y having to wait for their Interests to time out, before they can retransmit. Furthermore, there is no guarantee that their retransmissions will elicit a response (content or NACK).

As Theorem 4.1 proves, the NDN forwarding strategy specified in [21], [23] cannot work correctly when Interests are aggregated, even if nonces were to denote Interests uniquely. The theorem assumes that all messages are sent correctly and that no routing-table changes occur. Furthermore, Theorem 4.2 shows that no correct forwarding strategy exists that allows Interest aggregation and attempts Interest-loop detection by the matching of Interest-identification data.

Theorem 4.1: The NDN forwarding strategy is not safe in a stable, error-free network in which Interest loops occur, even if nonces denote Interests uniquely.

Proof: Consider the NDN forwarding strategy running in a network in which no two nonces created by different nodes for the same content are equal, all transmissions are received correctly, and no topology or routing-table changes occur after time t_0. Let $LT^{v_k}(I[n(j), id_j(s)])$ denote the lifetime of $I[n(j), id_j(s)]$ at router v_k.

Assume that some Interests traverse loops when they are forwarded according to the NDN forwarding strategy. Let a routing-table loop $L = \{v_1, v_2, ..., v_h, v_1\}$ exist for the name prefix that includes NDO j, and let Interest $I[n(j), id_j(x)]$ start traversing the chain of nodes $\{v_1,\ v_2, ...,\ v_k\} \in L$ (with $1 < k < h$) at time $t_1 > t_0$.

Assume that $I[n(j), id_j(x)]$ reaches router v_k at time $t_3 > t_1$ and that router v_k forwards Interest $I[n(j), id_j(y)]$ to its next hop $v_{k+1} \in L$ at time t_2, where $t_1 \leq t_2 < t_3$, $id_j(x) \neq id_j(y)$, and v_{k+1} may be v_1.

According to the NDN Interest processing strategy, router v_k creates an entry in its PIT for $I[n(j), id_j(y)]$ at time t_2, and perceives any Interest for name $n(j)$ and a nonce different than $id_j(y)$ received after time t_2, and before its PIT entry for $I[n(j), id_j(y)]$ is erased, as a subsequent Interest.

Let $|t_2 - t_3| < LT^{v_k}(I[n(j), id_j(y)])$ when router v_k receives $I[n(j), id_j(x)]$ from router $v_{k-1} \in L$ at time t_3, where $1 < k - 1$. According to the NDN Interest processing strategy, router v_k must treat $I[n(j), id_j(x)]$ as a subsequent Interest for content $n(j)$ that is aggregated, because v_k is waiting for $D[n(j), id_j(y)]$ at time t_3.

Because of L, Interest $I[n(j), id_j(y)]$ is forwarded from v_k to v_1. Let t_4 denote the time when $I[n(j), id_j(y)]$ reaches v_1, where $t_4 > t_2 \geq t_1$, and assume that $|t_1 - t_4| < LT^{v_1}(I[n(j), id_j(x)])$. According to the NDN Interest processing strategy, v_1 must treat $I[n(j), id_j(y)]$ as a subsequent Interest, given that it is waiting for $D[n(j), id_j(x)]$ at time t_4. As a result of the Interest aggregation carried out by nodes v_k and v_1, nodes in the chain $\{v_1, v_2, ..., v_{k-1}\} \in L$ process only $I[n(j), id_j(x)]$, nodes in the chain $\{v_{k+1},\ v_{k+2}, ...,\ v_h\} \in L$ process only $I[n(j), id_j(y)]$, and no Interest loop detection can take place. Therefore, no content can be submitted in response to $I[n(j), id_j(x)]$ and $I[n(j), id_j(y)]$. ■

Similar results to Theorem 1 can be proven for the NDN forwarding strategy and the original CCN forwarding strategy operating in an ICN in which routing tables are inconsistent as a result of network or content dynamics. In this case, Interest loops can go undetected even if the control plane supports only single-path routing of Interests.

Theorem 4.2: No forwarding strategy with Interest aggregation and Interest loop detection based on the matching of Interest-identification data is safe.

Proof: Assume any forwarding strategy in which a router remembers an Interest it has forwarded as long as necessary to detect Interest loops, and detects the occurrence of an Interest loop by matching the Interest-identification data carried in an Interest it receives with the Interest-identification data used in the Interest it forwarded previously asking for the same content. Let $I[n(j), id_j(s)]$ denote the Interest asking for $n(j)$ with Interest-identification data $id_j(s)$ created by router s.

Assume that an Interest loop $L = \{v_1, v_2, ..., v_h, v_1\}$ for NDO with name $n(j)$ exists in an ICN using the forwarding strategy. Let Interest $I[n(j), id_j(x)]$ start traversing the chain of nodes $\{v_1,\ v_2, ...,\ v_k\} \in L$ (with $1 < k < h$) at time t_1.

Assume that $I[n(j), id_j(x)]$ reaches router v_k at time $t_3 > t_1$ and that router v_k forwards Interest $I[n(j), id_j(y)]$ to its next hop $v_{k+1} \in L$ at time t_2, where $t_1 \leq t_2 < t_3$, $id_j(x) \neq id_j(y)$. Let $I[n(j), id_j(y)]$ traverse the chain of nodes $\{v_k,\ v_{k+1}, ...,\ v_1\} \in L$, reaching v_1 at time t_4, where $t_4 > t_2 \geq t_1$.

By assumption, Interest aggregation occurs, and hence v_k aggregates $I[n(j), id_j(x)]$ at time t_3, and v_1 aggregates $I[n(j), id_j(y)]$ at time t_4. Therefore, independently of the amount of information contained in $id_j(x)$ and $id_j(y)$, v_1 cannot receive $I[n(j), id_j(x)]$ from v_h and v_k cannot receive $I[n(j), id_j(y)]$ from v_{k-1}. It thus follows that no node in L can successfully use the matching of Interest-identification data to detect that Interests for $n(j)$ are being sent and aggregated along L and the theorem is true. ■

Theorems 4.1 and 4.2 can also be proven by mapping the Interest processing strategy of NDN and the original CCN to the problem of distributed termination detection over a cycle, where Interests serve as the tokens of the algorithm [6], [13]. Because Interest aggregation erases a token traversing the ring (Interest loop) when any node in the ring has previously created a different token, correct termination detection over the ring (i.e., Interest loop detection) cannot be guaranteed in the presence of Interest aggregation.

V. SIFAH: Strategy for Interest Forwarding and Aggregation with Hop-Counts

A. Design Rationale

It is obvious that a correct Interest processing strategy can be defined by specifying source routes in the Interests. Because a source-routed Interest must traverse the route stated in it or be dropped, no loops can be traversed by any Interest. However, this requires all routers in the ICN to have complete topology information or at least path information for each destination, which does not scale with the number of nodes and content objects. Furthermore, source routing of Interests makes Interest processing overly complex, and reveals the identity of the source router requesting content.

On the other hand, nonces can only ensure that Interests are denoted uniquely with some probability that is large enough to be acceptable in practice, and they still incur considerable storage overhead. More importantly, as Section IV shows, using nonces or identifying Interests uniquely is useless for Interest-loop detection when Interests are aggregated. Hence, for a forwarding strategy to be correct it must be the case that, independently of the identity of an Interest, at least one router detects that it is traversing a path that is not forwarding the Interest closer to a node that has advertised a prefix covering the requested content.

Distance information or some other ordering information is needed in any Interest-based ICN to allow routers to forward Interests towards the nearest instances of requested content, rather than flooding the network with Interests or carrying out random walks of the network searching for content. The same information can also be used to ensure that Interests are forwarded in a way that gets them closer to nodes that advertised the requested content. Given that forwarding information bases (FIB) are populated from the routing tables maintained in the control plane of an ICN, they constitute a readily-available tool to establish the proper interaction between the forwarding strategy operating in the data plane and the distances to advertised content maintained by the routing protocol operating in the control plane.

B. Information Stored and Exchanged

A router maintains a FIB, a PIT, and an optional content store. FIB^i is indexed using content name prefixes. The FIB entry for prefix $n(j)^*$ is denoted by $FIB^i_{n(j)^*}$, and consists of a list of one or more tuples. Each tuple states a next hop to $n(j)^*$ and a hop count to the prefix. The set of next hops to $n(j)^*$ listed in $FIB^i_{n(j)^*}$ is denoted by $S^i_{n(j)^*}$. The hop count to $n(j)^*$ through neighbor $q \in S^i_{n(j)^*}$ is denoted by $h(i, n(j)^*, q)$.

An Interest sent by node k requesting NDO $n(j)$ is denoted by $I[n(j), h^I(k)]$, and states $n(j)$ and the hop count ($h^I(k)$) from node k to the name prefix $n(j)^*$ that is the best match for NDO name $n(j)$ when k forwards the Interest.

A content-object message sent in response to Interest $I[n(j), h^I(k)]$ is denoted by $D[n(j), sig(j)]$, and states the name of the Interest, a signature payload $sig(j)$ used to validate the content object, and the object itself.

The NACK sent by router i in response to an Interest is denoted by $NI[n(j), \mathsf{CODE}]$ where CODE states the reason why the NACK is sent. Possible reasons for sending a NACK include: (a) an Interest loop is detected, (b) no route is found towards requested content, (c) no content is found, and (d) the PIT entry expired.

PIT^i is indexed using NDO names. $PI^i_{n(j)}$ denotes the entry created in PIT^i for NDO with name $n(j)$, and specifies: the name of the NDO; the hop count $h^I(i)$ assumed by router i when it forwards Interest $I[n(j), h^I(i)]$; the set of incoming neighbors from which Interests for $n(j)$ are received ($INSET(PI^i_{n(j)})$); the set of outgoing neighbor(s) ($OUTSET(PI^i_{n(j)})$) to whom router i forwards its Interest; and the remaining lifetime for the Interest ($RT(PI^i_{n(j)})$).

C. Interest Loop Detection

To define a correct forwarding strategy, special attention must be paid to the fact that updates made to the FIBs stored at routers occur independently of and concurrently with the updates made to their PITs. For example, once a router has forwarded an Interest that assumed a given distance to content prefix $n(i)^*$ and waits for its Interest to return a data object, its distance to the same content may change based on updated to its FIB. Hence, simply comparing the minimum distance from a router to content against a distance to content stated in an Interest is not enough to ensure that Interests are not incorrectly forwarded to routers that are farther away form the requested content.

SIFAH takes into account the fact that FIBs and PITs are updated independently by requiring that a router that forwards an Interest for a given piece of content store in its PIT entry the value of the distance to content assumed when it issues its Interest. The following rule is then used for a given router to determine whether an Interest may be propagating over an Interest loop. Hop count to content is used as the metric for the invariant condition. This is done for two reasons, storing hop-count distances in the FIB incurs less storage overhead than storing complex distance values, and the next hops to a prefix stored in the FIB can be ranked based on the actual distances to content. More sophisticated lexicographic forwarding rules could be defined based on the same general approach stated below; however, such a topic is outside the scope of this paper.

Hop-Count Forwarding with Aggregation Rule (HFAR): Router i can accept $I[n(j), h^I(k)]$ from router k if one of the following two conditions is satisfied:

1) $n(j) \notin PIT^i \wedge \exists\, v(\, v \in S^i_{n(j)^*} \wedge h^I(k) > h(i, n(j)^*, v)\,)$
2) $n(j) \in PIT^i \wedge h^I(k) > h^I(i)$

The first condition ensures that router i accepts an Interest from neighbor k only if i determines that it is closer to $n(j)^*$ through at least one neighbor than k was when it sent its Interest. The second condition ensures that router i accepts an Interest from neighbor k only if i was closer to $n(j)^*$ than k was when i and k sent their Interests.

Section VI proves that using HFAR is *sufficient* to ensure that an Interest loop cannot occur without a router in the loop detecting that the Interest has been forwarded incorrectly. This result is independent of whether Interests are aggregated or sent over one or multiple paths, or how Interests are retransmitted. The approach used for Interest-loop detection in SIFAH can be viewed as a case of termination detection based on diffusing computations [5]. Indeed, HFAR is similar to sufficient conditions for loop-free unicast or multicast routing based on diffusing computations (e.g., DUAL [7], MIP [15]). The difference between SIFAH and loop-free routing protocols based on diffusing computations is that SIFAH operates in the data plane using existing FIB entries, while routing protocols operate in the control plane to build routing tables and hence FIB entries.

It should be pointed out that, because HFAR is not *necessary* to detect loops, there are cases in which HFAR is not satisfied even though no Interest loops exist. However, given that FIBs are updated to reflect correct hop counts, a sufficient condition for loop detection operating with multi-path routing is a good baseline for a forwarding strategy in Interest-based ICNs.

D. SIFAH Operation

Algorithms 3 to 8 specify SIFAH, which consists of the steps taken by routers to process Interests, forward Interests, return NDOs, process perceived link failures, handle Interest-lifetime expirations, and send NACKs. Optional steps and data in algorithms are indicated by "**[o]**".

The algorithms used to describe SIFAH were not designed to take into account such issues as load balancing of available paths, congestion-control, or the forwarding of an Interest over multiple concurrent paths. For simplicity, it is assumed that all Interest retransmissions are carried out on an end-to-end basis (i.e., by the consumers of content) rather than routers. Hence, routers do not attempt to provide any "local repair" when a neighbor fails or a NACK to an Interest is received. Depending on the ICN architecture, Interest retransmissions could also be done by routers. The design and analysis of Interest retransmission strategies implemented by routers or by content consumers is a topic deserving further study.

Algorithm 3 implements HFAR. Router i determines that an Interest can be forwarded because Condition 1 in HFAR is satisfied (Line 9 of Algorithm 3), or an Interest can be aggregated because Condition 2 of HFAR is satisfied (Line 17 of Algorithm 3). Content requests from local content consumers are sent to the router in the form of Interests stating infinite hop counts to content, and each router knows which neighbors are remote and which are local.

Algorithm 3 SIFAH Processing of Interest at router i

```
 1: function Process Interest
 2: INPUT: $PIT^i$, $CS^i$, $FIB^i$, $I[n(j), h^I(k)]$;
 3: if $n(j) \in CS^i$ then send $D[n(j), sig(j)]$ to $k$
 4: if $n(j) \notin CS^i$ then
 5:     if $n(j) \notin PIT^i$ then
 6:         if $n(j)^* \notin FIB^i$ then
 7:             % No route exists to $n(j)^*$:
                send $NI[n(j), \text{no route}]$ to $k$; drop $I[n(j), h^I(k)]$
 8:         else
 9:             if $\exists\ v \in S^i_{n(j)^*}$ ( $h^I(k) > h(i, n(j)^*, v)$ ) then
10:                 % Interest can be forwarded:
                    call Forwarding Strategy($PI^i_{n(j)}$)
11:             else
12:                 % Interest may be traversing a loop:
                    send $NI[n(j), \text{loop}]$ to $k$;   drop $I[n(j), h^I(k)]$
13:             end if
14:         end if
15:     else
16:         % There is a PIT entry for $n(j)$:
17:         if $h^I(k) > h^I(i)$ then
18:             % Interest can be aggregated:
                $INSET(PI^i_{n(j)}) = INSET(PI^i_{n(j)}) \cup k$
19:         else
20:             % Interest may be traversing a loop:
                send $NI[n(j), \text{loop}]$ to $k$;  drop $I[n(j), h^I(k)]$
21:         end if
22:     end if
23: end if
24: end function
```

Algorithm 4 SIFAH Interest forwarding at router i

```
1: function Forwarding Strategy
2: INPUT: $PIT^i$, $FIB^i$, $MIL$, $I[n(j), h^I(k)]$;
3: for each $v \in S^i_{n(j)^*}$ by rank do
4:     if $h^I(k) > h(i, n(j)^*, v)$ then
5:         create $PI^i_{n(j)}$;
           $INSET(PI^i_{n(j)}) = \{k\}$; $OUTSET(PI^i_{n(j)}) = \{v\}$;
           $RT(PI^i_{n(j)}) = MIL$; $h^I(i) = h(i, n(j)^*, v)$;
           forward $I[n(j), h^I(i)]$ to $v$; return
6:     end if
7: end for
8: % No neighbor can be used in $S^i_{n(j)^*}$:
   for each $k \in INSET(PI^i_{n(j)})$ send $NI[n(j), \text{no route}]$ to $k$
9: end function
```

The Maximum Interest Life-time (MIL) assumed by a router before it deletes an Interest from its PIT should be large enough to preclude an excessive number of retransmissions. On the other hand, MIL should not be too large to cause the PITs to store too many Interests for which no NDO messages or NACKs will be sent due to failures or transmission errors. A few seconds would be a viable value for MIL. In practice, however, the consumer submitting an Interest to its local router could provide an initial value for the Interest lifetime estimated over a number of Interests submitted for NDOs in the same NDO group corresponding to a large piece of content (e.g., a movie). This is specially the case given our assumption that Interest retransmissions are carried out by content consumers, rather than by routers.

Algorithm 4 describes a simple forwarding strategy in which router i simply selects the first neighbor v in the ranked list of neighbors stored in the FIB for prefix $n(j)^*$ that satisfies the first condition in HFAR (Line 4 of the algorithm).

More sophisticated strategies can be devised that attain load balancing among multiple available routes towards content and can be close to optimum (e.g., [17]). In addition, the same Interest could be forwarded over multiple paths concurrently, in which case content is sent back over each path that the Interest traversed successfully. To be effective, however, these approaches must require the adoption of a loop-free multi-path routing protocol in the control plane (e.g., [8]). In this context, the control plane establishes valid multi-paths to content prefixes using long-term performance measures, and the data plane exploits those paths using HFAR and short-term performance measurements, without risking the long delays associated with backtracking due to looping.

Algorithm 5 outlines the processing of NDO messages received in response to Interests. A router accepts an NDO received from a neighbor if it has a PIT entry waiting for the content and the NDO came from one of the neighbors over which the Interest was sent (Line 5 of the algorithm). The router forwards the valid NDO to any neighbor that requested it and deletes the corresponding PIT entry. A router stores a data object it receives optionally (Step 7 of Algorithm 5) because it stores content according to the caching strategy used in the ICN, which can be path-based or edge-based [4], for example.

Algorithm 5 Process NDO message from q at router i

```
 1: function Process NDO message
 2: INPUT: $PIT^i$, $CS^i$, $FIB^i$, $D[n(j), sig(j)]$ received from $q$;
 3: [o] verify $sig(j)$;
 4: [o] if verification fails then drop $D[n(j), sig(j)]$
 5: if $n(j) \in PIT^i \wedge q \in OUTSET(PI^i_{n(j)})$ then
 6:     for each $p \in INSET(PI^i_{n(j)})$ do send $D[n(j), sig(j)]$ to $p$;
 7:     [o] store the content with name $n(j)$ in $CS^i$;
 8:     delete $PI^i_{n(j)}$
 9: else
10:     drop $D[n(j), sig(j)]$
11: end if
12: end function
```

Algorithm 6 shows a simple approach to handle the case when a PIT entry expires with no NDO or NACK being received. Given that routers do not initiate Interest retransmissions, router i simply sends NACKs to all neighbors from which it received Interests for $n(j)$. A more sophisticated approach would be needed for the case of ICNs in which routers must provide Interest retransmissions.

Algorithm 6 Process Interest life-time expiration

```
function Process Interest Life-time Expiration
INPUT: $PIT^i$, $RT(P^i_{n(j)}) = 0$;
for each $p \in INSET(PI^i_{n(j)})$ do send $NI[n(j), \text{Interest expired}]$
delete $PI^i_{n(j)}$
end function
```

Algorithm 7 states the steps taken to handle NACKs. Router i forwards the NACK it receives for $n(j)$ to all those neighbors from whom it received Interests for $n(j)$ and deletes the Interest entry after that. Supporting Interest retransmissions by routers would require a more complex approach for the handling of NACKs.

Algorithm 8 states the steps taken by a router in response to the failure of connectivity with a neighbor. Reacting to the failure of perceived connectivity with a neighbor over which Interests have been forwarded could be simply to wait for the life-times of those Interests to expire. However, such an approach can be very slow reacting to link failures compared to using Algorithm 8. The algorithm assumes that the control plane updates FIB^i to reflect any changes in hop counts to name prefixes resulting from the loss of connectivity to one or more neighbors. For each Interest that was forwarded over the failed link, router i sends a NACK to all neighbors whose Interests were aggregated.

Algorithm 7 Process NACK at router i

```
function Process NACK
INPUT: $PIT^i$, $NI[n(j), \mathsf{CODE}]$;
if $n(j) \notin PIT^i$ then
    drop $NI[n(j), \mathsf{CODE}]$
else
    if $k \notin OUTSET(PI^i_{n(j)})$ then drop $NI[n(j), \mathsf{CODE}]$;
    if $k \in OUTSET(PI^i_{n(j)})$ then
        for each $p \in INSET(PI^i_{n(j)})$ do send $NI[n(j), \mathsf{CODE}]$;
        delete $PI^i_{n(j)}$
    end if
end if
end function
```

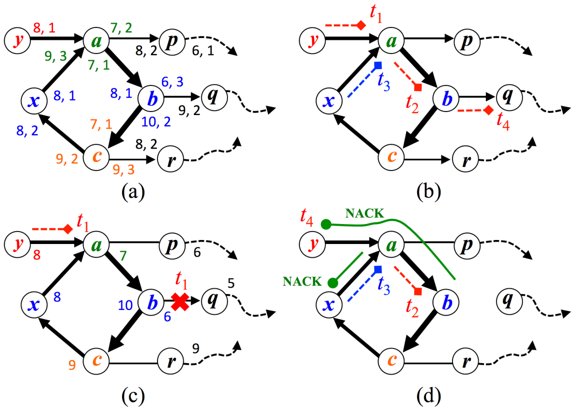
Algorithm 8 Process failure of link (i, k) at router i

```
function Process Link Failure
INPUT: $PIT^i$;
for each $n(j) \in PIT(i)$ do
    if $k \in INSET(PI^i_{n(j)})$ then
        $INSET(PI^i_{n(j)}) = INSET(PI^i_{n(j)}) - \{k\}$;
        if $INSET(PI^i_{n(j)}) = \emptyset$ then delete $PI^i_{n(j)}$;
    end if
    if $k \in OUTSET(PI^i_{n(j)})$ then
        $OUTSET(PI^i_{n(j)}) = OUTSET(PI^i_{n(j)}) - \{k\}$;
        if $OUTSET(PI^i_{n(j)}) = \emptyset$ then
            for each $p \in INSET(PI^i_{n(j)})$ do
                send $NI[n(j), \mathsf{no\ route}]$
            end for
            delete $PI^i_{n(j)}$
        end if
    end if
end for
end function
```

E. Examples of SIFAH Operation

Figures 2(a) to (d) illustrate how SIFAH operates using the same example used in Figure 1. Figures 2(a) and (b) address the case in which the control plane establishes multiple paths to each name prefix but does not guarantee loop-free routing tables. Figures 2(c) and (d) illustrate how SIFAH operates when single-path routing is used. The pair of numbers next to each link outgoing from a node in Figure 2(a) indicates the hop count to $n(j)$ through a neighbor and the ranking of the neighbor in the FIB. The example assumes that: (a) routers execute a routing protocol that does not enforce loop-free FIBs; and (b) the ranking of neighbors is determined independently at each router using some data-plane strategy based on the perceived performance of each path and interface, which is the approach advocated in NDN [14]. Note that the distance value of a path need not be directly proportional to the hop-count value of the path shown in the figure.

Let the tuple (v: h, r) indicate a neighbor, its hop count and its ranking. In Figure 2(a), FIB^a lists (b: 7, 1), (p: 7, 2), and (x: 9, 3). Similarly, FIB^y states (a: 8, 1); FIB^b states (c: 10, 2), (a: 8, 1), and (q: 6, 3); FIB^c states (b: 7, 1), (x: 9, 2), and (r: 9, 3); and FIB^x states (a: 8, 1) and (c: 8, 2). Some of the FIB entries for p, q and r are shown in the figure.



(a) (b) (c) (d)

Fig. 2. Interest looping is prevented or detected with SIFAH

In Figure 2(b), router y originates an Interest for $n(j)$ and sends $I[n(j), h^I(y) = 8]$ to a. Router a receives the Interest from router y at time t_1 and, given that $8 = h^I(y) > h(a, n(j)^*, b) = 7$, it accepts the Interest because it has at least one neighbor that satisfies HFAR. Router a sends $I[n(j), h^I(a) = 7]$ to b because it is the highest-ranked neighbor satisfying HFAR. Router a aggregates $I[n(j), h^I(x) = 8]$ at time $t_3 > t_1$, because it sent $I[n(j), h^I(a) = 7]$ at time t_1 and $8 = h^I(x) > h^I(a) = 7$. Router b receives the Interest from a at time $t_2 > t_1$; accepts it because it has at least one neighbor that satisfies HFAR ($7 = h^I(a) > h(b, n(j)^*, q) = 6$); and sends $I[n(j), h^I(b) = 6]$ to q because q is the highest-ranked neighbor of b that satisfies HFAR.

The above example illustrates that, even if some of the multi-paths implied in the FIBs involve loops, Interests are forwarded along loop-free paths if SIFAH is used and the FIBs maintained by routers have consistent information. The next section proves this result in the general case.

Figure 2(c) shows the hop count values stored in the FIBs for name prefix $n(j)$ when single-path routing is used. Each router has a single next hop and one hop count for each prefix listed in its FIB. Router b updates its FIB to reflect the failure of link (b, q) at time t_1, while router y sends an Interest to router a requesting $n(j)$. Routers have inconsistent FIB states for $n(j)$ while routing updates propagate and Interests are being forwarded.

As shown in Figure 2(d), router b *must* send $NI[n(j), \mathsf{loop}]$ to a, because $7 = h^I(a) \not> h(b, n(j)^*, c) = 10$ and HFAR is not satisfied. In turn, when a receives the NACK from b, it must forward $NI[n(j), \mathsf{loop}]$ to y and to x. Eventually, the routing protocol running in the control plane makes routers a and y change the hop count to $n(j)^*$ in their FIBs to reflect the failure of link (b, q). At that point, a retransmission of the Interest from y would state $h^I(y) = 9$ and would make a forward $I[n(j), h^I(a) = 8]$ to p.

VI. Correctness of SIFAH

The following theorems show that SIFAH enforces correct Interest forwarding and aggregation.

Theorem 6.1: Interest loops cannot occur and be undetected in an ICN in which SIFAH is used.

Proof: Consider a network in which SIFAH is used. Assume for the sake of contradiction that nodes in a loop L of h hops $\{v_1, v_2, ..., v_h, v_1\}$ send and possibly aggregate Interests for $n(j)$ along L, with no node in L detecting the incorrect forwarding of any of the Interests sent over the loop.

Given that L exists by assumption, $v_k \in L$ must send $I[n(j), h^I(v_k)]$ to node $v_{k+1} \in L$ for $1 \leq k \leq h-1$, and $v_h \in L$ must send $I[n(j), h^I(v_h)]$ to node $v_1 \in L$. For $1 \leq k \leq h-1$, let $h(v_k, n(j)^*)^L$ denote the value of $h^I(v_k)$ when node v_k sends $I[n(j), h^I(v_k)]$ to node v_{k+1}, with $h(v_k, n(j)^*)^L = h(v_k, n(j)^*, v_{k+1})$. Let $h(v_h, n(j)^*)^L$ denote the value of $h^I(v_h)$ when node v_h sends $I[n(j), h^I(v_h)]$ to node $v_1 \in L$, with $h(v_h, n(j)^*)^L = h(v_h, n(j)^*, v_1)$.

Because no node in L detects the incorrect forwarding of an Interest, each node in L must aggregate the Interest it receives from the previous hop in L or it must send its own Interest as a result of the Interest it receives from the previous hop in L. This implies that $v_k \in L$ must accept $I[n(j), h^I(v_{k-1})]$ before $RT(PI^{v_k}_{n(j)})$ expires for $1 \leq k < h$, and $v_1 \in L$ must accept $I[n(j), h^I(v_h)]$ before $RT(PI^{v_1}_{n(j)})$ expires.

According to SIFAH, if v_k aggregates $I[n(j),\ h^I(v_{k-1})]$, then it must be true that $h^I(v_{k-1}) > h^I(v_k)$. Similarly, if v_1 aggregates $I[n(j), h^I(v_h)]$, then it must be the case that $h^I(v_h) > h^I(v_1)$. On the other hand, if v_k sends $I[n(j), h^I(v_k)]$ to v_{k+1} as a result of receiving $I[n(j), h^I(v_{k-1})]$ from v_{k-1}, then it must be true that $h^I(v_{k-1}) > h(v_k, n(j)^*)^L = h^I(v_k)$ for $1 < k \leq h$. Similarly, if v_1 sends $I[n(j), h^I(v_1)]$ to v_2 as a result of receiving $I[n(j), h^I(v_h)]$ from v_h, then $h^I(v_h) > h(v_1, n(j)^*)^L = h^I(v_1)$.

It follows from the above argument that, for L to exist when each node in the loop follows SIFAH to send Interests asking for $n(j)$, it must be true that $h^I(v_h) > h^I(v_1)$ and $h^I(v_{k-1}) > h^I(v_k)$ for $1 < k \leq h$. However, this is a contradiction, because it implies that $h^I(v_k) > h^I(v_k)$ for $1 \leq k \leq h$. Therefore, the theorem is true. ■

The result in Theorem 6.1 is independent of whether the ICN is static or dynamic. SIFAH ensures that Interests cannot be incorrectly propagated and aggregated along loops without meeting routers that detect the incorrect forwarding and hence send NACKs in return. A natural consequence of this is that, as long as FIBs are consistent in an ICN and independently of how FIB entries are ranked, Interests are forwarded along loop-free paths when SIFAH is used.

Due to faults or transmission errors, Interests, NDOs or NACKs may be lost. A forwarding strategy is safe if, in the absence of faults or transmission errors, a router that issues an Interest acting on behalf of a consumer locally attached receives the requested NDO or a NACK within a finite time.

Theorem 6.2: SIFAH is safe in an ICN that is free of faults and transmission errors.

Proof: Assume that no network faults or errors occur after time t_0, and consider $I[n(j), h^I(s)]$ being issued at time $t_1 > t_0$ by consumer s. Because no faults or errors occur after t_0, the Interest must either traverse a loop or a simple path to a router d that can reply to the Interest.

If the Interest from s traverses a simple path to router d that advertises $n(j)^*$, then s must receive an NDO or a NACK originated by d, because either type of response must traverse the reverse path traversed by the Interest and no faults or transmission errors occur after t_0.

On the other hand, if $I[n(j), h^I(s)]$ is forwarded along an Interest loop, it follows from Theorem 6.1 that some router k must detect the incorrect forwarding of an Interest asking for $n(j)$ and must issue a NACK $NI[n(j), \text{loop}]$. According to SIFAH, (Algorithm 7) every relay receiving a NACK from the neighbor to whom an Interest was sent must relay a NACK . Because no errors occur after t_0, it follows that s must receive a NACK within a finite time after t_1. ■

VII. Performance Implications

The performance benefits attained with SIFAH compared to NDN are considerable. PITs are much smaller and consumers experience smaller latencies obtaining content or receiving feedback regarding the content they request when routers implement SIFAH instead of the NDN forwarding strategy.

A. PIT Size

In SIFAH, router i uses only the value of $h^I(i)$ to determine whether the Interest it receives from k may be traversing an Interest loop, and does not store $h^I(k)$. Hence, the PIT storage size for SIFAH is $SS_{SIFAH} = O((INT + |mh|)\ |PIT^i|_{SIFAH})$, where $|PIT^i|_{SIFAH}$ is the number of pending Interests in PIT^i when SIFAH is used, $|mh|$ is the number of bits used to store $h^I(i)$, and INT is the average storage required to maintain information about the incoming and outgoing neighbors for a given Interest. For a given NDO with name $n(j)$, the amount of storage needed to maintain the incoming and outgoing neighbors is $INSET(PI^i_{n(j)}) + OUTSET(PI^i_{n(j)})$.

By contrast, NDN requires each router to store the list of different nonces used to denote valid Interests for a given NDO name $n(j)$. With each nonce being of size $|id|$ and router i having up to I neighbors that send valid Interests for an NDO, the PIT storage size for NDN is $SS_{NDN} = O((INT + |id|I)\ |PIT^i|_{NDN})$, where $|PIT^i|_{NDN}$ is the number of pending Interests in PIT^i when NDN is used. Hence, even if $|PIT^i|_{NDN} = |PIT^i|_{SIFAH}$, the amount of additional PIT storage needed in NDN over SIFAH is $(|id|I)(|PIT^i|_{NDN}) - (|mh|)(|PIT^i|_{NDN})$.

A maximum hop count of 255 for an Interest is more than enough, while the size of a nonce in NDN is 16 bytes. Hence, the additional PIT storage required in NDN compared to SIFAH is $(128I-8)\ |PIT^i|_{NDN}$. This is *many orders of magnitude* the number of PIT entries and represents hundreds of gigabytes of RAM. Furthermore, because the NDN forwarding strategy does not detect loops when Interests are aggregated, many Interest entries in PITs have to be stored until their lifetimes expire. Accordingly, as the rate of Interests increases, the probability of undetected Interest loops also increases and $|PIT^i|_{NDN}$ becomes much larger than $|PIT^i|_{SIFAH}$.

The additional FIB storage overhead in SIFAH compared to NDN consists of storing the hop count information for each prefix $n(j)^*$ from each neighbor. This amounts to $(|mh|)(|FIB^i|)D^i$ at router i, where D^i is the number of neighbors of router i and $|FIB^i|$ is the number of entries in FIB^i. Given that D^i and I are of the same order and $O(|FIB^i|) < O(|PIT^i|)$, this is far smaller than the additional PIT storage needed by NDN compared to SIFAH.

B. End-to-End Latencies

SIFAH and NDN incur the same end-to-end latencies in the absence of routing-table loops in FIB entries, given that Interests and their replies traverse shortest paths. However, routing-table loops can lead to much longer end-to-end delays for the retrieval of content or the reception of NACKs with NDN than with SIFAH. In NDN, routers must wait for PIT entries to expire before sending any NACKs to consumers who issued Interests that traversed undetected Interest loops. Even if an NDO is received after the retransmission of an Interest that timed out and generated a NACK, the resulting latency is in the order of seconds, because the lifetimes of Interests in PITs must be set that long in order to avoid unnecessary retransmissions of Interests.

On the other hand, with SIFAH, a consumer must either obtain an NDO or a NACK in response to an Interest, and this must occur within a round-trip-time along the path between the customer and the router sending the NDO or detecting an Interest loop. This corresponds to a few hundred milliseconds in ICNs with topologies similar to today's Internet [2]. Furthermore, prior results on loop-free routing based on diffusing computations [17], [22] illustrate that false detection of Interest loops does not impact significantly the efficiency with which Interests are forwarded to routers with the stored content. This is especially the case if loop-free multi-path routing to name prefixes is provided in the control plane, which can be done with the Distance-based Content Routing protocol [8], [9].

VIII. Conclusions

We showed that the NDN forwarding strategy is not safe in the presence of Interest loops, and that a correct forwarding strategy that supports Interest aggregation cannot be designed simply by uniquely identifying each Interest.

We introduced the Strategy for Interest Forwarding and Aggregation with Hop-counts (SIFAH). It is the first forwarding strategy for Interest-based ICNs shown to be correct in the presence of Interest loops, Interest aggregation, faults, and the forwarding of Interests over multiple paths. SIFAH operates by requiring that FIBs store the next hops and the hop count through such hops to named content, and by having each Interest state the name of the content requested and the hop count from the relaying router to the content. We showed that SIFAH incurs far less storage overhead and renders shortest latencies than NDN.

This work is just a first step in the definition of correct forwarding strategies for Interest-based ICN architectures, and it is applicable to any Interest retransmission approach. More work is needed to understand the performance of SIFAH in large ICNs, the effect of lifetime timers on performance, the effect of load balancing of Interests over multiple routes to content, and the performance implications of the interaction between SIFAH and content routing protocols [8], [12].

References

[1] B. Ahlgren et al., “A Survey of Information-centric Networking,” *IEEE Commun. Magazine*, July 2012, pp. 26–36.

[2] L. Ciavatone et al., “Standardized Active Measurements on a Tier 1 IP Backbone,” *IEEE Comm. Magazine*, June 2003.

[3] Content Centric Networking Project (CCN) [online]. http://www.ccnx.org/releases/latest/doc/technical/

[4] A. Dabirmoghaddam et al., “Understanding Optimal Caching and Opportunistic Caching at "The Edge" of Information-Centric Networks,” *Proc. ACM ICN 2014*, Sept. 2014.

[5] E.W. Dijkstra and C.S. Scholten “Termination Detection for Diffusing Computations,” *Information Processing Letters*, Vol. 11, No. 1, 1980.

[6] E.W. Dijkstra, W. Feijen, and A.J.M. van Gasteren, “Derivation of a Termination Detection Algorithm for Distributed Computations,” *Information Processing Letters*, Vol. 16, No. 5, 1983.

[7] J.J. Garcia-Luna-Aceves, “A Unified Approach to Loop-Free Routing Using Distance Vectors or Link States,” *Proc. ACM SIGCOMM ‘89*, Aug. 1989.

[8] J.J. Garcia-Luna-Aceves, “Name-Based Content Routing in Information Centric Networks Using Distance Information,” *Proc. ACM ICN 2014*, Sept. 2014.

[9] J.J. Garcia-Luna-Aceves, “Efficient Multi-Source Multicasting in Information Centric Networks,” *Proc. IEEE CCNC ‘15*, Jan. 2015.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” *Proc. ACM MobiCom ‘00*, 2000.

[11] V. Jacobson et al., “Networking Named Content,” *Proc. IEEE CoNEXT ‘09*, Dec. 2009.

[12] A.K.M. Mahmudul-Hoque et al., “NSLR: Named-Data Link State Routing Protocol,” *Proc. ACM ICN ‘13*, 2013.

[13] J. Matocha and T. Camp, “A Taxonomy of Distributed Termination Detection Algorithms,” *Journal of Systems and Software*, 1998.

[14] NDN Project [online]. http://www.named-data.net/

[15] M. Parsa and J.J. Garcia-Luna-Aceves, “A Protocol for Scalable Loop-free Multicast Routing,” *IEEE JSAC*, Vol. 15, No. 3, 1997

[16] I. Solis and J.J. Garcia-Luna-Aceves, “Robust Content Dissemination in Disrupted Environments,” *Proc. ACM CHANTS ‘08*, Sept. 2008.

[17] S. Vutukury and J.J. Garcia-Luna-Aceves, “A Simple Approximation to Minimum-Delay Routing,” *Proc. ACM SIGCOMM ‘99*, Aug. 1999.

[18] S. Wood et al., “ICEMAN: A System for Efficient, Robust and Secure Situational Awareness at The Network Edge,” *Proc. IEEE MILCOM ‘13*, Nov. 2013.

[19] G. Xylomenos et al., “A Survey of Information-centric Networking Research,” *IEEE Communication Surveys and Tutorials*, July 2013.

[20] C. Yi et al., “Adaptive Forwarding in Named Data Networking,” *ACM CCR*, Vol. 42, No. 3, July 2012.

[21] C. Yi et al., “A Case for Stateful Forwarding Plane,” *Computer Communications*, pp. 779-791, 2013.

[22] W. Zaumen and J.J. Garcia-Luna-Aceves, “Dynamics of Distributed Shortest-Path Routing Algorithms,” *Proc. ACM SIGCOMM ‘91*, Sept. 1991.

[23] L. Zhang et al., “Named Data Networking,” *ACM SIGCOMM Computer Communication Review*, Vol. 44, No. 3, July 2014.