Title
Eliminating Undetected Interest Looping in Content-Centric Networks

Permalink
https://escholarship.org/uc/item/4cz721p4

Author
Garcia-Luna-Aceves, J.J.

Publication Date
2015-09-30

Peer reviewed

Eliminating Undetected Interest Looping in Content-Centric Networks

J.J. Garcia-Luna-Aceves[1,2]
[1]Palo Alto Research Center, Palo Alto, CA 94304
[2]Department of Computer Engineering, University of California, Santa Cruz, CA 95064
Email: jj@soe.ucsc.edu, maziar@soe.ucsc.edu

Abstract—**It has been shown that Interest loops can go undetected in NDN (named data networking) and CCNx (content-centric networking) when Interests are aggregated. To solve this problem, we introduce CCN-ELF, a simple variation of the way in which CCNx and NDN work based on a new type of forwarding information base (FIB) that stores distance information about name prefixes for neighbors of a content router that can serve as next hops, rather than just a ranked list of those next hops. CCN-ELF uses a loop-free forwarding algorithm based on the information available in the new FIBs that allows Interests to be forwarded and aggregated, without the risk of undetected Interest loops and without requiring any changes to the packet formats used in NDN and CCNx.**

I. INTRODUCTION

Several information-centric networking (ICN) architectures have been proposed as an alternative to today's Internet, and the leading ICN approach can be characterized as *Interest-based*. This approach consists of: populating forwarding information bases (FIB) maintained by 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. The original content-centric networking proposal was the first example of an Interest-based ICN architecture in which Interests need not be flooded and do not state the identity of the sender. Today, named data networking (NDN) [10] and CCNx [2] are the leading Interest-based ICN approaches.

Section II summarizes the operation of the NDN and CCNx forwarding planes. Since the introduction of the original content-centric networking proposal [2], the research community (e.g., [10], [12], [14]) has assumed that the forwarding planes of NDN and CCNx are such that they can recover from resource failures and congestion problems, because packets containing data are sent back in response to Interests. However, we have shown [6], [7] that this is not the case in general. More specifically, we have proven that Interest loops may go undetected when Interests from different consumers requesting the same content are aggregated and Interests are forwarded along routing loops, which may occur due to rankings of routes, failures, mobility, or congestion.

We have also shown [6] that no forwarding strategy can be designed that works correctly in the presence of Interest aggregation and uses nonces and the names of NDOs as the basis of Interest-loop detection. Section III shows an example of the occurrence of undetected Interest loops in NDN and CCNx.

We have recently proposed an approach [6] that remedies the Interest-loop-detection problems in NDN and CCNx. This approach requires an Interest to state a hop count to the intended name prefix. We have proven that this approach prevents Interests from looping, independently of the state of FIBs. However, a limitation of this approach is that the routing protocol operating in the control plane of the network must maintain hop counts to name prefixes in addition to any other type of distance information that may be used in the network (e.g., congestion- or delay-based distances).

We present the first solution to the Interest looping problems in NDN and CCNx that works correctly in the presence of aggregation and does not require any modifications to the Interest packet formats used in NDN and CCNx. We call this new approach **CCN-ELF** (*CCN with Expanded Look-up of FIB*), because the detection of Interest loops relies on a simple look-up of an expanded FIB that stores the distances to name prefixes reported by neighbors of a content router, rather than just the set of next hops to name prefixes.

Sections IV describes CCN-ELF, which ensures that Interest loops are detected if they occur, even if Interests from different consumers are aggregated. Sections VI proves that CCN-ELF ensures that no Interest loop can go undetected and that any Interest must receive a response within a finite time. Section VII addresses performance implications of CCN-ELF.

II. ELEMENTS OF THE FORWARDING PLANE IN NDN AND CCNX

In NDN and CCNx, a given router uses three primary data structures: a forwarding information base (FIB), a pending interest table (PIT), and a content store (CS). The forwarding plane uses these three tables to forward Interests towards routers advertising having copies of requested content, and send named data objects (NDO) or other responses back to consumers over reverse paths traversed by Interests.

A router uses its FIB to route Interests towards the desired content producer advertising a content-name prefix . A FIB is populated using content routing protocols or static routes. The FIB entry for a given name prefix lists the interfaces that can be used to reach the prefix. In NDN [12], the FIB entry for a name prefix also contains a stale time after which the entry could be deleted; the round-trip time through the interface; a rate limit; and status information stating whether it is known or unknown that the interface can bring data back, or is known that the interface cannot bring data back. A CS is a cache for content objects. With on-path caching, routers cache the content they receive in response to Interests they forward.

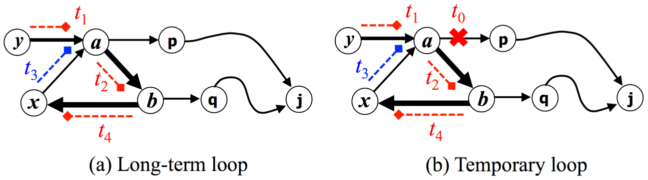
PITs are used in NDN and CCNx to keep track of the neighbors to which NDO messages or NACKs should be sent back in response to Interests, allow Interests to not disclose their sources, and enable Interest aggregation. A PIT entry in NDN lists the name of a requested NDO, one or multiple tuples stating a nonce received in an Interest for the NDO and the incoming interface where it was received, and a list of the outgoing interfaces over which the Interest was forwarded. A PIT entry in CCNx is similar, but no nonces are used.

When a router receives an Interest, it checks whether there is a match for the content requested in the Interest in its CS. The Interest matching mechanisms differ in NDN [10] and CCNx [2], with the latter supporting exact Interest matching only. If a match to the Interest is found, the router sends back an NDO over the reverse path traversed by the Interest. If no match is found in the CS, the router determines whether the PIT stores an entry for the same content. In NDN, if the Interest states a nonce that differs from those stored in the PIT entry for the requested content, then the router "aggregates" the Interest by adding the incoming interface from which the Interest was received and the nonce to the PIT entry without forwarding the Interest. On the other hand, if the same nonce in the Interest is already listed in the PIT entry for the requested content, the router sends a NACK over the reverse path traversed by the Interest. In CCNx, aggregation is done if the Interest is received from an interface that is not listed in the PIT entry for the requested content. A retransmitted Interest received from the same interface is forwarded [2].

If a router does not find a match in its CS and PIT, the router forwards the Interest along a route listed in its FIB for the best prefix match. In NDN, a router can select an interface to forward an Interest if it is known that it can bring content and its performance is ranked higher than other interfaces that can also bring content. The ranking of interfaces is done by a router independently of other routers.

III. UNDETECTED INTEREST LOOPS IN NDN AND CCNX

Figure 1 illustrates Interest looping in NDN and CCNx. 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 an interface is better than interfaces shown with thinner lines. Dashed lines indicate the traversal of Interests over links and paths. The time when an event arrives at a router is indicated by t_i. Figure 1(a) shows the case of a long-term Interest loop caused by multi-paths implied in FIBs not being loop-free, even though all routing tables are consistent. In this case, the ranking of interfaces in a FIB can be such that a path with a larger hop count may be ranked higher than a path with a smaller hop count, because of the perceived performance of the interfaces or paths towards prefixes. 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.



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

Fig. 1. Undetected Interest looping in NDN and CCNx

In both cases, router a aggregates the Interest from x and router x aggregates the Interest from y, and the combined steps preclude the detection of any Interest looping. In this example, it would appear that the looping problems could be avoided by forcing router b to use q rather than x for Interests regarding prefixes for which router j is an origin. However, the same looping problems would exist even if link (b, q) were removed in the example, and the ways in which FIBs are populated and interfaces are ranked are independent of updates made to PITs.

We have proven [6], [7] that undetected Interest loops can occur in NDN and CCNx, and that no forwarding strategy that supports Interest aggregation can be defined to ensure the detection of Interest loops by using the names of NDOs or nonces stated in the Interests. Although an Interest cannot recirculate along a routing loop in NDN, an undetected Interest loop causes PIT entries to remain in storage until they time out, given that no response are sent to aggregated Interests that traverse routing loops. As the results in [7] indicate, this can cause large increases in end-to-end delays and the number of PIT entries stored by content routers, even when the percentage of Interests traversing loops is small.

IV. CCN-ELF OPERATION

The design objective of CCN-ELF is to ensure that no Interest loops can go undetected even when Interests are aggregated, and without requiring any changes to the packet formats used in NDN and CCNx. The design rationale in CCN-ELF is twofold. First multi-path content routing protocols [5], [9], [8] are a much more attractive alternative than single-path routing in content-centric networks. Second, the FIB of each content router is a readily-available tool to enforce the needed ordering in the forwarding strategy operating in the data plane.

The operation of CCN-ELF differs from the current specifications of NDN and CCNx only in the way in which Interests are forwarded and the modifications needed in FIBs and PITs. Accordingly, we only describe those aspects of CCN-ELF that differ from NDN and CCNx. In our description, we assume that Interests are retransmitted only by the consumers that originated them, rather than routers that relay Interests. Routers are assumed to know which interfaces are neighbor routers and which are local consumers, and forward Interests on a best-effort basis. Furthermore, given that no Interest matching policy has been shown to work better than simple exact matching of Interests, we assume that routers use exact Interest matching as in CCNx [2] to forward Interests.

A. Information Exchanged and Stored

The information used to enable correct forwarding of Interests, NDO messages, and NACKs are the name of NDOs, *distance* information stored in FIBs, and link-cost information to each neighbor. Interests, NDO messages and NACKs (called Interest return in CCNx [2]) are assumed to specify the same information used in NDN or CCNx.

The name of NDO j is denoted by $n(j)$. The terms neighbor and interface are used interchangeably, and the set of neighbor routers of router i is denoted by N^i. An Interest forwarded by router k requesting NDO $n(j)$ is denoted by $I_k[n(j)]$. An NDO message sent by router k in response to an Interest is denoted by $D_k[n(j), sp(j)]$, where $sp(j)$ is the security payload used optionally to validate the NDO. The NACK to an Interest sent by router k is denoted by $NI_k[n(j), \mathsf{CODE}]$, where CODE is a code indicating the reason why the NACK is sent. Possible reasons for sending a NACK include: an Interest loop is detected, or no route is found towards the requested content.

The PIT, FIB and CS maintained at router i are denoted by PIT^i, FIB^i, and CS^i, respectively. Router i also maintains a link-cost table (LT^i).

The CS in CCN-ELF is the same as in NDN and CCNx. FIB^i in CCN-ELF is updated by the routing protocol operating in the control plane. It is indexed using content name prefixes and stores additional information than in NDN or CCNx. The entry in FIB^i for name prefix $n(j)^*$ is denoted by $FIB^i_{n(j)^*}$ and consists of a set of tuples, one for each neighbor of router i. The tuple for neighbor $q \in N^i$ states the name of neighbor q and the distance from q to $n(j)^*$ ($D(i, n(j)^*, q)$).

PIT^i in CCN-ELF is slightly different with respect to NDN and CCNx. The entry in PIT^i for NDO with name $n(j)$ is denoted by $PIT^i_{n(j)}$ and, in addition to the information maintained in NDN or CCNx, it stores the distance assumed by router i to name prefix $n(j)^*$ when it forwarded $I_i[n(j)]$. This distance is denoted by $D(i, n(j))$. A point worth mentioning is that the nonces used in NDN are not needed for Interest-loop detection.

LT^i stores the cost of the link from router i to each of its neighbor routers. It can be updated based on the congestion perceived over the link. The cost of the link from router i to neighbor v is denoted by $c(i, v)$.

B. Interest Loop Prevention and Detection

Interest loops resulting from inconsistencies in FIB entries maintained at different routers are avoided or detected if they occur using the following rule.

ELF Rule: Router i accepts Interest $I_k[n(j)]$ from router k if one of the following two conditions is satisfied:

1) $n(j) \notin PIT^i \wedge$
   $\exists v \in N^i(D(i, n(j)^*, k) > D(i, n(j)^*, v)) + c(i, v));$
   set $D(i, n(j)) = D(i, n(j)^*, v)) + c(i, v))$
2) $n(j) \in PIT^i \wedge D(i, n(j)^*, k) > D(i, n(j))$

The first condition ensures that router i accepts an Interest from neighbor k only if router i determines that it can forward the Interest for $n(j)$ through a neighbor that is closer to prefix $n(j)^*$ than neighbor k is. The second condition ensures that router i accepts an Interest from neighbor k only if router i was closer to $n(j)^*$ when it sent its Interest for $n(j)$ than neighbor k is when the Interest from k is received.

The ELF rule is independent of the specific metric used to measure distances from routers to name prefixes, or the specific way in which a router selects next hops towards a name prefix. Section VI proves that using the ELF rule is *sufficient* to ensure that an Interest loop cannot occur in CCN-ELF, without a router in the loop detecting that the Interest has been forwarded incorrectly.

C. Interest Forwarding in CCN-ELF

CCN-ELF operates in much the same way as NDN and CCNx do. The difference is in the way in which Interests are forwarded according to the ELF rule using the expanded FIBs and the distance information stored in PITs.

Algorithms 1 describes the steps taken by routers to process Interests. Our description does not take into account such issues as load balancing of available paths to name prefixes, congestion-control, or the forwarding of an Interest over multiple paths concurrently. 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 relaying routers. Hence, routers do not attempt to provide any "local repair" when a neighbor fails or a NACK to an Interest is received.

Algorithm 1 implements the ELF rule to ensure that no Interest looping goes undetected. Router i forwards a new Interest when Condition 1 in the ELF rule is satisfied (Line 10 of Algorithm 1), or aggregates an Interest when Condition 2 of the ELF rule is satisfied (Line 18 of Algorithm 1). For simplicity, we assume that content requests from local content consumers are sent to the router in the form of Interests, and each router knows which neighbors are remote and which are local.

$INSET(PIT^i_{n(j)})$ denotes the set of neighbors from which router i has received an Interest for NDO $n(j)$; $OUTSET(PIT^i_{n(j)})$ denotes the set of neighbors to which router i has sent an Interest for NDO $n(j)$; and $RT(PIT^i_{n(j)})$ denotes the lifetime of the PIT entry. The Maximum Interest Life-time (MIL) assumed by a router before it deletes an Interest from its PIT is large enough to preclude unnecessary retransmissions, and not too large to cause the PITs to store too many Interests for which no NDO messages or NACKs can be sent due to failures or transmission errors.

Algorithm 1 describes a simple forwarding strategy for Interests 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 the ELF rule (Line 10 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., [11]). 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., [5], [8]). In this context, the control plane establishes valid multi-paths to prefixes using long-term performance measures, and the data plane exploits those paths using the ELF rule and short-term performance measurements, without risking the long delays associated with backtracking due to looping.

Algorithm 1 CCN-ELF Processing Interest from router k

```
function Process Interest
INPUT: $PIT^i$, $CS^i$, $FIB^i$, $LT^i$, $I_k[n(j)]$;
if $n(j) \in CS^i$ then send $D_i[n(j), sp(j)]$ to $k$
if $n(j) \notin CS^i$ then
    if $n(j) \notin PIT^i$ [% No prior Interest is pending for $n(j)$] then
        if $n(j)^* \notin FIB^i$ [% No route exists to $n(j)^*$] then
            send $NI_i[n(j), \mathsf{no\ route}]$ to $k$; drop $I_k[n(j)]$
        else
            for each $v \in N^i$ by rank do
                if $D(i, n(j)^*, k) > D(i, n(j)^*, v) + c(i, v)$ then
                    [% Interest can be sent to $v$: ]
                    create $PIT^i_{n(j)}$;
                    $INSET(PIT^i_{n(j)}) = \{k\}$; $OUTSET(PIT^i_{n(j)}) = \{v\}$;
                    $D(i, n(j)) = D(i, n(j)^*, v) + c(i, v)$;
                    $RT(PIT^i_{n(j)}) = MIL$;
                    send $I_i[n(j)]$ to $v$; return
                end if
            end for
            [% Interest may be traversing a loop: ]
            send $NI_i[n(j), \mathsf{loop}]$ to $k$;   drop $I_k[n(j)]$
        end if
    else
        % There is a PIT entry for $n(j)$:
        if $D(i, n(j)^*, k) > D(i, n(j))$ then
            [% Interest can be aggregated: ]
            $INSET(PI^i_{n(j)}) = INSET(PI^i_{n(j)}) \cup k$
        else
            [% Interest may be traversing a loop: ]
            send $NI_i[n(j), \mathsf{loop}]$ to $k$;  drop $I_k[n(j)]$
        end if
    end if
end if
end function
```

V. EXAMPLES OF CCN-ELF OPERATION

Figures 2(a) and (b) illustrate how CCN-ELF operates when a multi-path routing protocol is used to populate the FIBs. The same example shown in Figure 1 is used.

The pair of numbers next to a router in Figure 2(a) indicate the distance from that router to prefix $n(j)^*$ through a neighbor and the ranking of the neighbor according to the FIB of the router. Let the triplet (v, h, r) denote a neighbor, a distance to the prefix through that neighbor, and its ranking in the FIB of a router. Assuming that *all* neighbors of a router are listed in the FIB entry for $n(j)^*$ in Figure 2(a), the choices at router a are $(b, 4, 1)$, $(p, 4, 2)$, and $(x, 6, 3)$, and $(y, 6, 4)$. The choices at router b are $(x, 6, 1)$, $(a, 5, 2)$, and $(q, 3, 3)$. The choices are router x are $(a, 5, 2)$ and $(b, 5, 1)$.

As Figure 2(b) shows, when router a receives $I[n(j)]$ from router y at time t_1, it forwards $I[n(j)]$ to b because b offers the highest ranked distance to $n(j)$ satisfying the ELF rule, i.e., $D(a, n(j)^*, y) = 5 > 4 = D(a, n(j)^*, b) + c(a, b)$. Router a sets $D(a, n(j)) = 4$ in its PIT. Router b receives the Interest from a at time t_2 and accepts it , because the ELF rule is also satisfied by neighbor i.e., $D(b, n(j)^*, a) = 4 > 3 = D(b, n(j)^*, q) + c(b, q)$. The Interest generated by router x is aggregated by router a at time t_3, because the ELF rule is satisfied, i.e., $D(a, n(j)^*, x) = 5 > 4 = D(a, n(j))$. In contrast to the case shown in Fig. 1 for NDN and CCNx, no loop occurs in CCN-ELF.

Figures 2(c) to (e) illustrate how CCN-ELF operates when FIB entries are inconsistent among routers due to topology changes. Router a updates its FIB at time time t_0 and router b updates its FIB at time t_1 as shown in Figure 2(c). Routers have inconsistent FIB states for $n(j)^*$ because routing-table updates are being sent in the control plane while Interests are being forwarded in the data plane. The figure shows the snapshot of values stored in FIBs at the times Interests propagate after link (a, p) fails and link (b, q) increases its cost from 1 to 6. Each number next to a router indicates the distance to prefix $n(j)^*$ through a given neighbor.

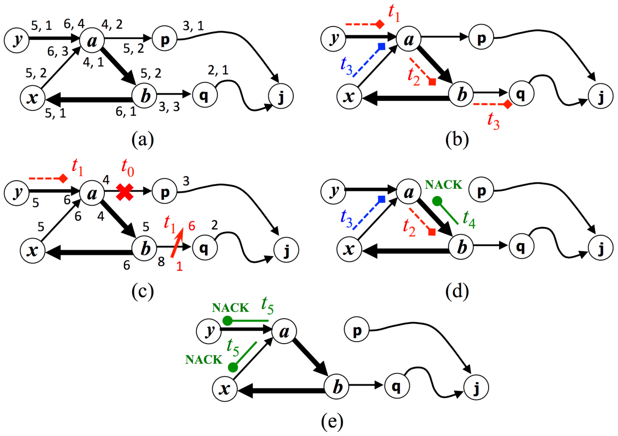


Fig. 2. CCN-ELF prevents or detects Interest loops.

As Figure 2(c) shows, when the Interest for $n(j)$ from router y arrives at router a at time t_1, router a forwards the Interest because the ELF rule is satisfied by router b, i.e., $D(a, n(j)^*, y) = 5 > 4 = D(a, n(j)^*, b) + c(a, b)$. Router a sets $D(a, n(j)) = 4$ in its PIT.

As shown in Figure 2(d), even though FIBs are inconsistent, router b sends a NACK to router a when the Interest arrives at time t_2, because b cannot find any neighbor v such that $D(b, n(j)^*, v) + c(b, v) < 4$. Router a aggregates the Interest from router x at time t_3, because the ELF rule is satisfied, i.e., $D(a, n(j)^*, x) = 5 > 4 = D(a, n(j))$. Router a forwards the NACK it receives from b at time t_4 to routers y and x.

Within a finite time, the FIBs of all routers are updated to reflect the new shortest paths that take into account the changes to links (a, p) and (b, q). Once FIBs are consistent, Interests regarding objects in the name prefix $n(j)^*$ are forwarded along shortest paths towards $n(j)^*$.

The ELF rule is only a sufficient condition to avoid Interest looping, and it is possible for a router to assume that an Interest is traversing a loop when this is not the case. In the example in Figure 2(d), router b could forward the Interest to router q without causing a loop. However, the ELF rule is not satisfied by router q and b cannot select it.

Given the speed with which FIBs are updated to reflect correct distances computed in the control plane, false loop detections are rare, and their occurrence is better than having to store PIT entries for Interests that cannot receive responses, until their lifetimes expire after many seconds.

VI. CORRECTNESS OF CCN-ELF

The following theorems show that no Interest loops can occur and be undetected if CCN-ELF is used, and that every Interest must receive a response (an NDO message or a NACK) within a finite time. These results are independent of whether the network is static or dynamic, the specific caching strategy used in the network (e.g., at the edge or along paths traversed by NDO messages [3]), the retransmission strategy used by content consumers or relay routers after experiencing a timeout or receiving a NACK, or how many paths are used to forward an Interest.

Theorem 6.1: Interest loops cannot occur and be undetected in a network in which CCN-ELF is used.

Proof: Consider a network in which CCN-ELF is used. Assume for the sake of contradiction that routers 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 router in L detecting the incorrect forwarding of any of the Interests sent over the loop. Let $D^L(u, n(j)^*, v)$ and $c^L(u, v)$ denote the distance from v to $n(j)^*$ stored in FIB^u and the cost of link (u, v) stored in LT^u when router u becomes part of loop L, respectively.

Given that L is assumed to exist, $v_k \in L$ must send $I_{v_k}[n(j)]$ to router $v_{k+1} \in L$ for $1 \leq k \leq h-1$, and $v_h \in L$ must send $I_{v_h}[n(j)]$ to router $v_1 \in L$. For $1 \leq k \leq h-1$, let $D^L(v_k, n(j))$ be the distance to $n(j)^*$ that v_k computes and stores in PIT^{v_k} when it sends $I_{v_k}[n(j)]$ to router v_{k+1}. Similarly, let $D^L(v_h, n(j))$ be the distance to $n(j)^*$ that v_h stores in PIT^{v_h} when it sends $I_{v_h}[n(j)]$ to router $v_1 \in L$

Because no router in L detects the incorrect forwarding of an Interest, each router in L must either 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 $I_{v_{k-1}}[n(j)]$ before the PIT timer expires for $1 \leq k < h$, and $v_1 \in L$ must accept $I_{v_h}[n(j)]$ before the PIT timer expires.

According to the ELF rule, if v_k aggregates $I_{v_{k-1}}[n(j)]$, then it must be true that $D^L(v_k, n(j)^*, v_{k-1}) > D^L(v_k, n(j))$. Similarly, if v_1 aggregates $I_{v_h}[n(j)]$, then it must be the case that $D^L(v_1, n(j)^*, v_h) > D^L(v_1, n(j))$.

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

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

An Interest forwarding strategy must ensure that either an NDO message or a NACK is received within a finite time by the consumer who issues an Interest. The following theorem shows that this is the case for CCN-ELF, independently of the state of the topology or the fate of messages.

Theorem 6.2: CCN-ELF ensures that a consumer that issues an Interest for a valid NDO with name $n(j)$ receives an NDO message for name $n(j)$ or a NACK within a finite time.

Proof: Consider an Interest for $n(j)$ being issued by consumer s at time t_1. The forwarding of Interests assumed in CCN-ELF is based on the best match of the requested NDO name with the prefixes advertised in the network. A router sends back an NDO message to a neighbor that sent an Interest for NDO $n(j)$ only if has an exact match of the name $n(j)$ in its content store, and a router that receives an NDO message in response to an Interest it forwarded must forward the same NDO message. Hence, the wrong NDO message cannot be sent in response to an Interest. There are three cases to consider next: (a) there are no routes to the name prefix $n(j)^*$ of the requested NDO, (b) the Interest traverses a routing loop, or (c) the Interest traverses a simple path towards a router d that can reply to the Interest.

Case 1: If there is no route to $n(j)^*$, then it follows from the operation of CCN-ELF that a router issues a NACK stating that there is no route. That NACK is either forwarded successfully back to s or is lost due to errors or faults. In the latter case, a router must send a NACK back towards s stating that the Interest expired or the route failed.

Case 2: If an Interest for $n(j)$ is forwarded along a loop and does not reach any router with a copy of $n(j)$, then it follows from Theorem 6.1 that the Interest must either reach some router k that detects the incorrect forwarding of the Interest and issues a NACK stating that there is a loop, or the Interest is dropped due to faults or transmission errors before reaching such router k. Each router that receives a NACK in response to an Interest sends NACKs back to all neighbors from which it received Interests for $n(j)$. Hence, if no errors or faults prevent the NACK from reaching s, the consumer receives a NACK stating that an Interest loop was found. On the other hand, if either the Interest traversing an Interest loop or the NACK it induces at some router k is lost, a router between s and router k must send a NACK towards s indicating that the Interest expired or that the route failed. Accordingly, consumer s must receive a NACK within a finite time after issuing its Interest in this case.

Case 3: If the Interest traverses a simple path towards a router d that advertises $n(j)^*$ or has a content store containing $n(j)$, then the Interest must either reach d or is lost before reaching d. If the Interest is lost before reaching d, then a router between s and router d must send a NACK towards s indicating that the Interest expired or that the route failed. As a result, s must receive a NACK originated by some router between s and d. If the Interest reaches d, then that router must send the requested NDO back. The NDO message originated by d is forwarded back towards s along the reverse simple path traversed by the Interest. If no fault or errors occur between d and s, it follows that the theorem is true for this case. Alternatively, if the NDO message originated by d is lost due to faults or errors, a router between s and d must send a NACK towards s indicating that the Interest expired or that the route failed. ∎

VII. PERFORMANCE IMPLICATIONS

The performance benefits attained with CCN-ELF compared to NDN and CCNx as currently implemented can be considerable in the presence of Interest looping.

The additional storage requirements for the link-cost table are negligible and independent of the number of pending Interests. Compared to CCNx and NDN, CCN-ELF requires additional storage for each FIB entry maintained for a name prefix and each PIT entry maintained for an Interest.

The additional FIB storage required in CCN-ELF consists of storing the distance reported by each neighbor for each prefix $n(j)^*$. This is in the order of $(|D|)(|FIB^i|)(|N^i|)$ bytes at router i, where D is the number of bytes needed to represent a distance, $|N^i|$ is the number of neighbors of router i and $|FIB^i|$ is the number of entries in FIB^i.

The additional PIT overhead incurred with CCN-ELF consists of storing a distance value for each PIT entry. This corresponds to just $(|D|)(|PIT^i_{ELF}|)$ bytes at router i, where $|PIT^i_{ELF}|$ is the number of PIT entries.

Given that CCN-ELF does not need nonces to detect Interest looping, NDN PITs could be simplified by not storing the nonces stated in Interests, which represents storage savings in the order of $(|id|)(|PIT^i_{NDN}|)(|N^i|)$ bytes, where $|id| >> |D|$ is the number of bytes needed to state a nonce and where $|PIT^i_{NDN}|$ is the number of PIT entries.

CCN-ELF incurs the same end-to-end latencies as NDN and CCNx in the absence of routing-table loops in FIB entries, given that Interests and their replies traverse shortest paths. However, latencies and the number of PIT entries can increase drastically when Interests are aggregated while traversing routing loops.

Interests that are aggregated along routing loops in NDN and CCNx must remain in the PIT until they expire before any NACKs can be sent to the consumers who issued the Interests. The resulting latency incurred in responding to such Interests 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 CCN-ELF, 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 topologies similar to today's Internet. Recent simulation results [7] indicate that, even if only a small percentage of Interests are aggregated along routing loops, undetected Interest loops result in large increases in the number of PIT entries stored in content routers (i.e., $|PIT^i_{NDN}| >> |PIT^i_{ELF}|$) and end-to-end delays in obtaining responses to Interests.

Prior results on loop-free routing [11], [13] illustrate that false detection of Interest loops should not impact significantly the efficiency with which Interests are forwarded to routers with the requested content. This is especially the case if loop-free multi-path routing to name prefixes is provided in the control plane (e.g., DCR [5]).

VIII. CONCLUSIONS

Undetected Interest loops have been shown to occur in NDN and CCNx, which causes Interests to timeout without content or negative acknowledgments being received in response. We introduced CCN-ELF, the first approach to content-centric networking that eliminates the possibility of undetected Interest loops without requiring packet formats to be modified in CCNx or NDN.

Compared to NDN, the additional storage needed to maintain distances to prefixes through each neighbor and the distance assumed to a name prefix when an Interest is forwarded is more than compensated by the storage savings derived from not having to store the nonces included in Interests. The mechanisms needed for CCN-ELF can be adopted in NDN and CCNx, because it does not change any of the packet formats, and the additional storage needed to implement the ELF rule is proportional to the number of FIB entries.

The ELF rule also points out a way in which SIFAH [6] can be modified to use distance values other than minimum-hop counts. The end result would be the ability to eliminate Interest looping by having an Interest state the distance to the requested content and FIBs maintain distances reported by neighbors considered to be next hops to prefixes.

REFERENCES

[1] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for ns-3", *University of California, Los Angeles, Tech. Rep*, 2012.

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

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

[4] S. Fayazbakhsh et al., "Less Pain, Most of the Gain: Incrementally Deployable ICN," *Proc. ACM SIGCOMM '13*, 2013.

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

[6] J.J. Garcia-Luna-Aceves, "A Fault-Tolerant Forwarding Strategy for Interest-based Information Centric Networks," *Proc. IFIP Networking'15*, May 2015.

[7] J.J. Garcia-Luna-Aceves and M. Mirzazad-Barijough, "Enabling Correct Interest Forwarding and Retransmissions in a Content Centric Network," *Proc. ACM/IEEE ANCS '15*, May 2015.

[8] E. Hemmati and J.J. Garcia-Luna-Aceves, "A New Approach to Name-Based Link-State Routing for Information-Centric Networks," *Proc. ACM ICN 2015*, Sep. 2015.

[9] A.K.M. Mahmudul-Hoque et al., "NSLR: Named-Data Link State Routing Protocol," *Proc. ACM ICN '13*, 2013.

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

[11] S. Vutukury and J.J. Garcia-Luna-Aceves, "A Simple Approximation to Minimum-Delay Routing," *Proc. ACM SIGCOMM '99*, Aug. 1999.

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

[13] W. Zaumen and J.J. Garcia-Luna-Aceves, "Dynamics of Distributed Shortest-Path Routing Algorithms," *Proc. ACM SIGCOMM '91*, Sept. 1991.

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