UC Santa Cruz
UC Santa Cruz Previously Published Works

Title
New Directions in Content Centric Networking,

Permalink
https://escholarship.org/uc/item/12f0q0bs

Author
Garcia-Luna-Aceves, J.J.

Publication Date
2015-09-04

Peer reviewed

eScholarship.org
Powered by the California Digital Library
University of California

New Directions in Content-Centric Networking

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 revisit some of the basic premises of the Content-Centric networking (CCN) and Named-Data Networking (NDN) architectures, which have been proposed as alternatives to the IP Internet architecture in order to support more efficient access to content available in the Internet. We address the large overhead incurred in NDN and CCN by maintaining forwarding state for each Interest traversing a router and for each content-name prefix known to each router. We introduce a new approach designed to provide orders of magnitude reduction in the complexity of the data plane to make content-centric networking much more viable at Internet scale.**

I. INTRODUCTION

Several information-centric networking (ICN) architectures have been proposed to support more efficient access to and dissemination of content than the IP Internet architecture provides today [3]. CCN was the first ICN approach based on an Interest-based approach [12] consisting of sending content objects or feedback to consumers in response to Interests, which are requests for such content by name. The named-data networking (NDN) architecture [14] and CCNx [5] have evolved directly from CCN, and are the leading approaches to Interest-based content-centric networking.

The key advantages offered by CCNx and NDN compared to the IP Internet include: (a) content providers and caching sites do not know the identity of the consumers requesting content; (b) content can be obtained by name from those sites that are closer to consumers; (c) data packets carrying content cannot traverse loops, because they are sent over the reverse paths traversed by Interests; and (d) content-oriented security mechanisms can be implemented as part of the content-delivery mechanisms.

However, the benefits attained with NDN and CCN come at a big price, which is the large complexity of the forwarding plane they require to operate. A Pending Interest Table (PIT) is needed to store forwarding state for each Interest traversing the router pending a response, and a forwarding information base (FIB) is required to store forwarding state for each name prefix known to the router.

The large storage requirements of maintaining PITs in NDN or CCN deployments at Internet scale has been addressed by a number of authors [6], [15], [16]. By some estimates, the number of PIT entries needed for NDN and CCN to operate at Internet scale is $O(10^6)$. This is comparable to the number of entries of the FIBs maintained by high-end Internet routers today. However, the number of FIB entries needed for NDN and CCN to operate at Internet scale is larger by far than the number of PIT entries. The number of Internet web sites is already $O(10^9)$, with the growth of web sites being exponential, and the number of top level domain (TLD) names by the end of last year was $O(10^8)$. Hence, the number of FIB entries of name prefixes required to deploy NDN and CCN today is $O(10^8)$ or even larger.

The "stateful forwarding plane" [17] (one that maintains per-Interest forwarding state at each router) used in NDN and CCN requires orders of magnitude more storage complexity in routers. This could be viewed as a goal that must be attained if maintaining per-Interest state were necessary to derive the benefits of content-centric networking. However, a stateful forwarding plane also introduces new types of vulnerabilities and does not solve the problems found in a stateless forwarding plane.

Interest flooding attacks [2], [16] can be mounted in which end users simply send Interests requesting valid or invalid content corresponding to routable content-name prefixes at rates that overload PITs.

On the other hand, even though data packets carrying content cannot loop in NDN or CCN, we have shown [9], [10] that Interests may be aggregated[1] while traversing forwarding loops. This can result in Interests never receiving a response because the Interests are unable to reach a router with a local content producer or cached content. Even temporary inconsistencies among the FIBs of various routers can result in Interests "waiting to infinity" in PITs, resulting in longer latencies for those flows affected by the loops [10].

Furthermore, we have obtained simulation results of NDN [11] illustrating that in-network caching can obviate the need for Interest aggregation, and that the percentage of Interests that are aggregated is a very small fraction of Interests. This result holds independently of the rates at which Interests are submitted or whether or not Interests exhibit temporal correlation.

In light of the above numbers and recent results, we argue that the mechanisms used in the forwarding plane of a

---

[1] A router forwards only the first Interest it receives asking for some content and suppresses subsequent Interests for the same content.

content-centric-networking architecture must be re-examined in order to enable deployments at Internet scale. We use a specific example to illustrate possible research directions in content-centric networking at Internet scale. The following sections introduce CCN-GRAM (*Gathering of Routes for Anonymous Messengers*) to illustrate how content-centric networking can operate with a stateless forwarding plane (i.e., no per-Interest state) by forwarding Interests and responses to them as *anonymous* datagrams using destination-based forwarding tables.

CCN-GRAM should be viewed simply as a proposal to elicit much needed discussion on new research directions regarding the algorithms and mechanisms that can make the control and forwarding planes of content-centric networks at least as efficient as the control and forwarding planes of today's IP Internet, or even more efficient.

II. CCN-GRAM OVERVIEW

CCN-GRAM is based on Interests, Data packets, and negative acknowledgments (NACK). Similar to IP datagrams, the messages sent in CCN-GRAM specify a source and a destination. For an Interest, the destination is the name of a content object and the source is an anonymous identifier. For Data packets and NACKs, the source is the name of a content object and the destination is an anonymous identifier.

Forwarding in CCN-GRAM takes place by means of four tables: a LIGHT (Local Interests GatHered Table), a FIB, an ART (Anonymous Routing Table) and a LIST (Local Interval Set Table).

The LIGHT of a router is an index of content locally available, as well as content that is remote and has been requested by local users. The FIB of each router states the distance to each known name prefix attained by each neighbor router. The ART is maintained on-demand using Interests, and states the paths to destinations denoted with local identifiers from which routers cannot discern the origins of Interests. The LIST states the intervals of local identifiers that a router assigns to its neighbors and that each neighbor assigns to the router.

A proactive routing approach is used by routers to maintain routes to known name prefixes. A novel on-demand routing approach is also used to maintain anonymous routes to the routers that originate Interests for specific content on behalf of content consumers. Only the local router serving a user knows the identity of the user; and no other router, content provider, or caching site can determine the router that originated an Interest on behalf of a user.

For simplicity, we make a number of assumptions in the description of CCN-GRAM that should not be considered design requirements. We assume that Interests are retransmitted only by the consumers that originated them, rather than routers that relay Interests. Interest retransmissions can be attempted by relays in much the same way that "local repair" mechanisms exist for on-demand routing protocols, and they could enable faster reaction to congestion or topology changes. We assume that routers use exact Interest matching, and that a router that advertises being an origin of a content-name prefix stores all the content objects associated with that prefix at a local store. Routers know which interfaces are neighbor routers and which are local users, and forward Interests on a best-effort basis. For convenience, a request for content from a local user is sent to its local router in the form of an Interest.

III. INFORMATION EXCHANGED AND STORED IN CCN-GRAM

CCN-GRAM supports content dissemination using Interests, Data packets, and NACKs. The name of content object (CO) j is denoted by $n(j)$ and the name prefix that is the best match in the FIB for name $n(j)$ is denoted by $n(j)^*$

An Interest forwarded by router k requesting CO $n(j)$ is denoted by $I[n(j), D^I(k), AID^I(k)]$, and states the name of the requested CO ($n(j)$), a distance to the requested content ($D^I(k)$), and an assigned identifier ($AID^I(k)$) that is modified on a hop-by-hop basis to hide the identity of the origin of the Interest.

A Data packet sent by router i in response to an Interest is denoted by $DP[n(j), sp(j), AID^R(i)]$, and states the name of the CO being sent ($n(j)$), a security payload ($sp(j)$) used optionally to validate the CO, and an assigned identifier ($AID^R(i)$) that denotes the intended recipient of the Data packet without revealing its true identity.

The NACK sent by router i in response to an Interest is denoted by $NA[n(j), \mathsf{CODE}, AID^R(i)]$ and states the name of a CO ($n(j)$), a code (CODE) indicating the reason for the NACK, and an assigned identifier ($AID^R(i)$) that denotes the intended recipient of the NACK. Possible reasons for sending a NACK include: an Interest loop is detected, no route is found towards requested content, and no content is found.

Router i maintains the following tables for forwarding: an optional Local Interests GatHered Table ($LIGHT^i$), a forwarding information base (FIB^i), an Anonymous Routing Table (ART^i), and a Local Interval Set Table ($LIST^i$).

$LIGHT^i$ lists the names of the COs that either are remote and have been requested by router i or are stored locally at router i. It is indexed by the CO names. The entry for CO name $n(j)$ states the name of the CO ($n(j)$), a pointer to the content of the CO ($p[n(j)]$), and a list of zero or more identifiers of local consumers ($lc[n(j)]$) that have requested the CO while the content is remote.

FIB^i is indexed using known content-name prefixes. The entry for prefix $n(j)^*$ states the hop-count distance reported by each neighbor router for the prefix. The distance stored for neighbor q for prefix $n(j)^*$ in FIB^i is denoted by $D(i, n(j)^*, q)$. Each entry in FIB^i is assumed to be updated as needed by the routing protocol running in the control plane.

ART^i is indexed using local identifiers. Each entry states a local identifier, a previous hop, and a name prefix. $ART^i(a, p, n)$ is used to denote the entry for local identifier a, previous hop p, and name prefix n.

An entry in ART^i can result from router i originating or forwarding an Interest. An entry created for a given local identifier a resulting from router i originating Interests for COs with names that are best matched by name prefix $n(j)^*$ states a, router i as the previous hop, and prefix $n(j)^*$. On the other hand, an entry for local identifier a resulting from router i forwarding Interests from another router simply states identifier a and the previous hop $p^i(a)$ of the path traversed by the Interests in which router i uses the local identifier a. Each entry in ART^i is stored for a few seconds.

$LIST^i$ maintains the intervals of local identifiers used by router i. For each neighbor k, $LIST^i$ states the interval of local identifiers (local interval) assigned to router i by router k ($LI^i(k, i)$) and the local interval assigned to router k by router i ($LI^i(i, k)$).

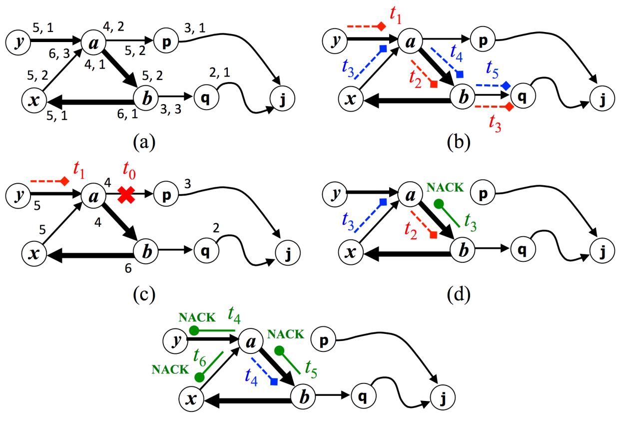
IV. AVOIDING FORWARDING LOOPS

Let $S^i_{n(j)^*}$ denote the set of neighbors of router i considered to be next hops to prefix $n(j)^*$. The following rule is used to ensure that Interests cannot traverse routing loops, even if the routing data stored in FIBs regarding name prefixes is inconsistent and leads to routing-table loops.

Loop-Free Forwarding Rule (LFR):
Router i accepts $I[n(j), AID^I(k)]$ from router k if:

$$\exists\ v \in S^i_{n(j)^*}(\ D^I(k) > D(i, n(j)^*, v)\ )$$

LFR is the same condition for loop-free forwarding we have proposed in [11] and have shown to prevent Interests from traversing forwarding loops even when the prefix entries in the FIBs maintained by routers incur routing loops.



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

Figure 1. CCN-GRAM prevents forwarding loops

Figures 1(a) and (b) illustrate how CCN-GRAM prevents Interests from traversing loops when a multi-path routing protocol is used to populate the FIBs and FIB entries state next hops to prefixes that lead to forwarding loops. The pair of numbers next to a node in Figure 1(a) indicate the hop count from that node to $n(j)$ over an interface and the ranking of the interface according to the FIB of the node. Let (v, h, r) denote the triplet indicating an interface, its hop-count distance to content, and its ranking.

In Figure 1(a), FIB^a states $(b, 4, 1)$, $(p, 4, 2)$, and $(x, 6, 3)$; FIB^b states $(x, 6, 1)$, $(a, 5, 2)$, and $(q, 3, 3)$; and FIB^x states $(a, 5, 2)$ and $(b, 5, 1)$. As Figure 1(b) shows, router a receives $I[n(j), D^I(y) = 5, AID^I]$ from router y at time t_1. Router a forwards $I[n(j), D^I(a) = 4, AID^I]$ to b, because $5 = D^I(y) > D(a, n(j)^*, b) = 4$ and b is ranked above p. Router b receives the Interest at time t_2 and accepts it, because $4 = D^I(a) > D(b, n(j)^*, q) = 3$. Router b must use neighbor q as the next hop for the Interest, because q is the highest ranked neighbor satisfying LFR. Similarly, the Interest generated by router x is forwarded to router q towards j without traversing a loop, because each relaying router must satisfy LFR.

Figures 1(c) to (e) illustrate how CCN-GRAM operates when single-path routing is used and a temporary routing-table loop exists in the FIBs. Each router has a single next hop and hop count for each prefix in its FIB. The distance from a router to name prefix $n(j)^*$ need not be directly proportional to the hop counts of the paths. For example, link (b, q) may have limited bandwidth or long delays and hence b prefers the path through x to reach $n(j)^*$. Router b updates its FIB at time t_0 as shown in Figure 1(c), and routers have inconsistent FIB states for $n(j)$ while Interests are being forwarded. As shown in Figure 1(d), router b *must* send $NI[n(j), \mathsf{loop}, AID^R]$ to a, because $4 = D^I(a) \not> D(b, n(j)^*, x) = 6$. In turn, a forwards a NACK to y. The Interest from x also prompts a NACK from b because LFR is not satisfied. Within a finite time, FIB^a, FIB^x, and FIB^b are updated to reflect the new topology state, and Interests from y regarding objects in $n(j)^*$ can be forwarded along the chain of nodes a, b, and q towards $n(j)^*$. Similarly, within a finite time, Interests from x regarding $n(j)^*$ can be forwarded to b and q towards $n(j)^*$.

V. FORWARDING TO ANONYMOUS DESTINATIONS

To forward Data packets and NACKs using a destination-based forwarding table approach, routers need identifiers that unambiguously denote the intended destinations. In contrast to IP datagram forwarding, the destinations must be anonymous, in the sense that no forwarding router, caching site, or content producer should be able to identify the router that originated the Interest prompting a Data packet or NACK.

The identifiers used in Interests could be nonces assigned randomly by an originated router and the name space for such nonces can be large enough to ensure a small probability of collision (i.e., more than one router using the same nonce). However, such a sender-initiated approach to the assignment of identifiers may allow a team of routers to find out the identities of some origins of Interests. Accordingly, it is subject to attacks based on the creation of fake identifiers. In addition, the length identifiers must be large to reduce collisions.

CCN-GRAM adopts a novel receiver-initiated approach for the assignment of local identifiers used to forward Data packets and NACKs. This approach takes advantage of the fact that Data packets and NACKs are sent over the reverse paths traversed by the Interests that prompt them.

Each router has a large set of local identifiers that it can assign to its neighbors, so that those routers can in turn use them as the AIDs in the Interests they forward to the router itself. We assume that a router assigns non-overlapping continuous local intervals of identifiers (*local intervals*) of equal length to its neighbors, such that a router cannot give the same local identifier to more than one neighbor. Given that all all local intervals have the same length, a local interval is uniquely defined by the identifier at the start of the interval.

An identifier i in local interval LI is denoted by $LI(i)$. Let two local intervals be $LI_A = [LI_A(s), LI_A(e)]$ and $LI_B[LI_B(s), LI_B(e)]$, a simple way to obtain an identifier $AID_B \in LI_B$ from an identifier $AID_A \in LI_A$ consists of computing $AID_B = AID_A - LI_A(s) + LI_B(s)$ modulo the length of an interval. The inverse function is the same.

To simplify our description of CCN-GRAM, we assume that routers have exchanged their local intervals with their neighbors and have populated their LISTs accordingly, and also assume that local intervals do not change for extended periods of time after they are assigned.

Routers can exchange local intervals with their neighbors in a number of ways. The exchange can be done in the data plane using Interests and Data packets. An example would be having a router send an Interest stating a common name denoting that a neighbor interval is requested, and an empty AID. Given the succinct way in which local intervals can be stated (an identifier denotes its interval), the exchange can also be easily done as part of the routing protocol running in the control plane. Routers could exchange local intervals in HELLO messages, link-state advertisements or distance updates.

Let router i originate an Interest regarding CO $n(j)$ on behalf of a local consumer and forward Interest $I[n(j), D^I(i), AID^I(i)]$ to neighbor k towards name prefix $n(j)^*$. Router i must use as $AID^I(i)$ an identifier from $LI^i(k, i)$ that it is not currently using for a different destination. Router i stores the entry $[AID^I(i), i]$ in ART^i and adds to the entry the prefix name $n(j)^*$.

When router k forwards an Interest $I[n(j),\ D^I(k),\ AID^I(k)]$ from router i to neighbor y towards name prefix $n(j)^*$, router k must use in its own Interest an identifier from the interval $LI^k(y, k)$ using $AID^I(i)$ as the key. Given that all local intervals are of the same length, router k can use any one-to-one mapping from $LI^k(k, i)$ to $LI^k(y, k)$ to determine the value of $AID^I(k)$. We denote this mapping by $f_k(i, y) : LI^k(k, i) \rightarrow LI^k(y, k)$. Router k then stores the entry $[AID^I(k), y]$ in ART^k.

When router k forwards the Data packet $DP[n(j),\ sp(j),\ AID^R(y)]$ received from neighbor y, it uses $AID^R(y)$ as the key in ART^k to determine the next hop x that the Data packet should take. Given the next hop x, router i uses the inverse function $f_k^{-1}(x, y)$ to map $AID^R(y) \in LI^k(y, k)$ into $AID^R(k) \in LI^k(k, x)$, and forwards $DP[n(j), sp(j), AID^R(k)]$ to router x. The same approach is followed for the forwarding of NACKs.

Many bijections can be used by routers to map identifiers from one local interval to another. What is important to note is that, once a router originates an Interest with a given local identifier, it is not possible to have collisions with the identifiers created by other origins, because of two reasons. First, each router uses new AIDs for new destinations when it originates an Interest. Second a router uses a bijection to map the AID it receives in an Interest into the AID it uses in the Interest it forwards, which must be one of the identifiers of the local interval given to the router by its next hop.

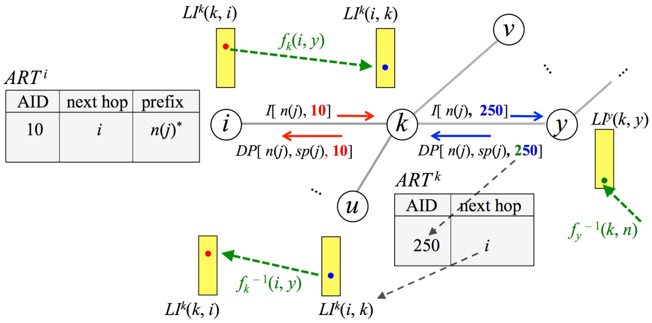


Figure 2. CCN-GRAM supports anonymous forwarding of Interests and Data packets

Figure 2 illustrates how CCN-GRAM uses announced identifiers and ARTs to forward Data packets and NACKs back to the origins of Interests without any router being able to discern the true identity of routers serving as Interest origins. In the figure, router i uses $AID^I(i) = 10$ in the Interests it sends towards prefix $n(j)^*$ through router k and stores the entry $[10, i]$ in ART^i. The mapping function at k for neighbor y transforms $AID^I(i) = 10$ to $AID^I(k) = 250$ and router k stores the entry $[250, i]$ in ART^k. Router y would do a similar mapping, or reply with the requested content. When router k receives a Data packet from y with $AID^R(y) = 250$, the entry in ART^k instructs router k to use the inverse mapping function at k for neighbor i, which results in mapping $AID^R(y) = 250$ to $AID^R(k) = 10$ and router k forwards the Data packet to i.

It is clear from the example that, even when a very small number of routers is involved, only the router that originates an Interest is able to determine that fact, because the identifiers used for forwarding over reverse paths traversed by Interests are assigned by the receivers of Interests at each hop. The role of the announced identifiers is also clear, they are only an aid for a router to select the inverse function to use when it forwards Data packets or NACKs.

VI. MAINTAINING ANONYMOUS FORWARDING STATE

Routers maintain routes to those sites advertising being the origins of content-name prefixes, and populate their routing tables using a routing protocol operating in the control plane (e.g., DCR [7] or NLSR [13]). Routers populate their FIBs with routes to name prefixes based on the data stored in their routing tables. How long FIB entries are maintained is determined by the operation of the routing protocol. We assume that a multipath routing protocol is used to populate FIBs and that a router is able to know a hop-count distance from each neighbor router to each name prefix. This information is stored in the FIB of a router.

The following algorithms specify the steps taken by routers to process and forward Interests, and return Data packets or NACKs. We assume that each router is initialized properly, knows the identifiers used to denote local consumers, knows all its neighbors, and knows the local identifier intervals associated with each neighbor.

Algorithm 1 Processing Interest from user c at router i

```
function Interest_Source
INPUT: $LIGHT^i$, $LIST^i$, $FIB^i$, $ART^i$, $I[n(j), nil, c]$;
if $n(j) \in LIGHT^i$ then
    if $p[n(j)] \neq nil$ then
        retrieve CO $n(j)$; send $DP[n(j), sp(j), c]$ to $c$
    else
        $lc[n(j)] = lc[n(j)] \cup c$; $p[n(j)] = nil$ (% Interest is aggregated)
    end if
else
    if $n(j)^* \in LIGHT^i$ then
        send $NA[n(j), \text{no content}, c]$ (% $n(j)$ does not exist)
    else
        if $n(j)^* \notin FIB^i$ then
            send $NA[n(j), \text{no route}, c]$ to $c$ (% No route to $n(j)^*$ exists)
        else
            create entry for $n(j)$ in $LIGHT^i$: (% Interest from $c$ is recorded)
            $lc[n(j)] = lc[n(j)] \cup c$; $p[n(j)] = nil$;
            for each $v \in S^i_{n(j)*}$ by rank in $FIB^i$ do
                $D^I(i) = D^i(i, n(j)^*, v)$;
                if $\exists ART^i(a, p, n)$ $(a \in LI^i(v, i) \wedge p = i \wedge n = n(j)^*)$ then
                    $AID^I(i) = a$; send $I[n(j), D^I(i), AID^I(i)]$ to $v$;
                    return
                else
                    select $AID \mid AID \in LI^i(v, i) \wedge AID \neq a \;\forall\; ART^i(a, i, n)$;
                    create new entry $ART^i(AID, i, n(j)^*)$;
                    $AID^I(i) = AID$; send $I[n(j), D^I(i), AID^I(i)]$ to $v$;
                    return
                end if
            end for
        end if
    end if
end if
```

Algorithm 1 shows the steps taken by router i to process Interests received from local consumers. For convenience, content requests from local consumers are assumed to be Interests stating the name of a CO together with the name of the consumer.

Router i first searches its LIGHT to determine if the content is stored locally or a request for the same content is pending. If the content is stored locally, a Data packet is sent back to the user requesting the CO. If a request for the same content is pending, the name of the user is added to the list of users that have requested the CO. In our description, a router that advertises being the origin of a prefix must have all the COs associated with the prefix stored locally. If router i states that it is the origin of the name prefix $n(j)^*$ and a specific CO with a name that is in that prefix is not found locally, a NACK is sent back stating that the content was not found.

If the CO is remote and no FIB entry exists for a name prefix that can match $n(j)$, a NACK is sent back stating that no route to the CO could be found. Otherwise, router i forwards the Interest through the highest ranked neighbor in its FIB for the name prefix matching $n(j)$. How such a ranking is done is left unspecified, and can be based on a distributed or local algorithm. If an ART entry exists for the selected successor that should receive the Interest, the existing route is used; otherwise, a new ART entry is created before the Interest is forwarded, making sure that the assigned identifier selected by router i has not been used for any other ART entry created as a result of router i being the origin of Interests.

Algorithm 2 Processing Interest from router k at router i

```
function Forwarding
INPUT: $LIGHT^i$, $LIST^i$, $FIB^i$, $ART^i$, $I[n(j), AID^I(k)]$;
if $n(j) \in LIGHT^i$ then
    if $p[n(j)] \neq nil$ then
        retrieve CO $n(j)$;
        $AID^R(i) = AID^I(k)$; send $DP[n(j), sp(j), AID^R(i)]$ to $k$
    end if
else
    if $n(j)^* \in LIGHT^i$ then
        $AID^R(i) = AID^I(k)$;
        send $NA[n(j), \text{no content}, AID^R(i)]$ to $k$ (% $n(j)$ does not exist)
    else
        if $n(j)^* \notin FIB^i$ then
            $AID^R(i) = AID^I(k)$;
            send $NA[n(j), \text{no route}, AID^R(i)]$ to $k$ (% No route to $n(j)^*$ exists)
        else
            for each $v \in S^i_{n(j)*}$ by rank in $FIB^i$ do
                if $D^I(k) > D(i, n(j)^*, v)$ (% LFR is satisfied) then
                    $AID = f_i(k, v)[AID^I(k)]$;
                    if $\exists ART^i(a, p, n)$ $(a = AID \wedge p = k)$ then
                        $AID^I(i) = AID$; $D^I(i) = D(i, n(j)^*, v)$;
                        send $I[n(j), D^I(i), AID^I(i)]$ to $v$
                    else
                        create new entry $ART^i(AID, k, nil)$
                    end if
                    return
                end if
            end for (% LFR is not satisfied; Interest may be traversing a loop)
            $AID^R(i) = AID^I(k)$;
            send $NA[n(j), \text{loop}, AID^R(i)]$ to $k$
        end if
    end if
end if
```

Algorithm 2 shows the steps taken by router i to process an Interest received from a neighbor router k. The main differences in the steps taken by router i compared to Algorithm 1 are that: (a) no Interest aggregation is done for Interests received from neighbor routers; and (b) router i simply maps the AID it receives in the Interest from the previous hop to the AID it should use in the Interest it sends to the next hop using a simple mapping function.

If the requested content is cached locally, a Data packet is sent back. If router i is an anchor of $n(j)^*$ and the CO with name $n(j)$ is not found locally, a NACK is sent back stating that the content could not be found. If the CO is remote and no FIB entry exists for $n(j)^*$, then the router sends a NACK stating that no route could be found for the CO.

Router i tries to forward the Interest to a next hop v to $n(j)^*$ (the best prefix match for $n(j)$) that satisfies LFR. The highest-ranked router satisfying LFR is selected as the successor for the Interest and router i. The AID used in the Interest is obtained with a simple mapping function from the AID received from router k to the corresponding AID that should be used from the local identifier interval provided by the selected next hop v. If no entry in ART^i exists for the same AID and previous hop k, the entry is created. The entry is valid for any name prefix for which the same path and AID are used, and hence the name prefix of the entry is left empty.

Algorithm 3 Processing Data packet at router i

```
function Data Packet
INPUT: $LIGHT^i$, $LIST^i$, $ART^i$, $DP[n(j), sp(j), AID^R(q)]$;
[o] verify $sp(j)$;
[o] if verification fails then discard $DP[n(j), sp(j), AID^I(q)]$
if $\exists ART^i(a, p, n)$ $(p = i)$
  (% router $i$ was the origin of the Interest) then
    for each $c \in lc[n(j)]$ do
        send $DP[n(j), sp(j), c]$ to $c$; $lc[n(j)] = lc[n(j)] - \{c\}$
    end for
end if
if $\exists ART^i(a, k, n)(k \in N^i)$ then
    $AID^R(i) = f_i^{-1}(k, q)[AID^R(q)]$;
    send $DP[n(j), sp(j), AID^R(i)]$ to $k$
end if
[o] if no entry for $n(j)$ exists in $LIGHT^i$ then
        create $LIGHT^i$ entry for $n(j)$: $lc[n(j)] = \emptyset$
     end if
[o] store CO in local storage; $p[n(j)] =$ address of CO in local storage
```

Algorithm 3 outlines the processing of Data packets. If the router has local users that requested the content, the Data packet is sent to those consumers based on the information stored in $LIGHT^i$. If the Data packet is received in response to an Interest that was forwarded from router k, router i forwards the Data packet doing the proper mapping of AIDs. Router i stores the data object if edge or on-path caching is supported. The steps needed to process NACKs are very similar.

VII. CONCLUDING REMARKS

Much more work remains to be done in order to make content-centric networking more efficient than today's Internet forwarding plane. In particular, a major additional impediment for the deployment of content-centric networking is the size of the FIBs needed to support routing of Interests and content by name, which must be in the order of published name prefixes in NDN and CCN. This is a direct consequence of the fact that NDN and CCN integrate name resolution with routing. However, as recent surveys of naming and routing in information-centric networks show [4], routing Interests by name does not have to implemented this way.

It is worth pointing out that redirection can be used as an integral part of the content routing protocol itself (e.g., CORD [8]) so that routing tables store entries to publishing sites without the need to keep track of specific name prefixes. Doing so is very promising, because it can reduce the number of FIB entries by orders of magnitude. We view this as a critical area of future research that needs to be undertaken together with approaches aimed at simplifying the way in which anonymous Interests are forwarded.

REFERENCES

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

[2] A. Afanasyev et al., "Interest Flooding Attack and Countermeasures in Named Data Networking," *Proc. IFIP Networking '13*, May 2013.

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

[4] M.F. Bari et al., ?A Survey of Naming and Routing in Information-Centric Networks,? IEEE Commun. Magazine, July 2012, pp. 44?53.

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

[6] H. Dai el al., "On Pending Interest Table in Named Data Networking," *Proc. ANCS '12*, Oct. 2012.

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

[8] J.J. Garcia-Luna-Aceves, Q. Li, and Turhan Karadeniz, "CORD: Content Oriented Routing with Directories," *Proc. IEEE ICNC 2015*, Feb. 2015.

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

[10] 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.

[11] J.J. Garcia-Luna-Aceves and M. Mirzazad-Barijough, "Content Centric Networking without Pending Interest Tables," *Proc. ACM ICN '15*, Sept. 2015.

[12] V. Jacobson et al., "Networking Named Content," *Proc. IEEE CoNEXT '09*, Dec. 2009.

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

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

[15] M. Varvello et al., "On The Design and Implementation of a Wire-Speed Pending Interest Table," *Proc. IEEE Infocom NOMEN Workshop*, April 2013.

[16] M. Virgilio et al., "PIT Overload Analysis in Content Centric Networks," *Proc. ACM ICN '13*, Aug. 2013.

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