Content-Centric Networking at Internet Scale through The Integration of Name Resolution and Routing

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

ABSTRACT

We introduce CCN-RAMP (Routing to Anchors Matching Prefixes), a new approach to content-centric networking. CCN-RAMP offers all the advantages of the Named Data Networking (NDN) and Content-Centric Networking (CCNx) but eliminates the need to either use Pending Interest Tables (PIT) or lookup large Forwarding Information Bases (FIB) listing name prefixes in order to forward Interests. CCN-RAMP uses small forwarding tables listing anonymous sources of Interests and the locations of name prefixes. Such tables are immune to Interest-flooding attacks and are smaller than the FIBs used to list IP address ranges in the Internet. We show that no forwarding loops can occur with CCN-RAMP, and that Interests flow over the same routes that NDN and CCNx would maintain using large FIBs. The results of simulation experiments comparing NDN with CCN-RAMP based on ndnSIM show that CCN-RAMP requires forwarding state that is orders of magnitude smaller than what NDN requires, and attains even better performance.

CCS Concepts

•**Networks** → **Network architectures; Network design principles; Naming and addressing;**

1. INTRODUCTION

Several Information-Centric Networking (ICN) architectures have been proposed [4, 5, 44] to improve on the performance of the IP Internet by enabling packet forwarding based on the names of content or services required, rather than the addresses where they may be hosted. They attempt to accomplish this by means of new ways to integrate name resolution (mapping of names to locations) and routing (establishing paths between locations) functions.

Section 2 summarizes prior work on forwarding planes for ICN architectures. Most architectures keep name resolution and routing independent of each other, but offer major improvements over the current use of the Domain Name System (DNS) to map domain names to IP addresses. By contrast, the Content-Centric Networking (CCNx) [6] and the Named Data Networking (NDN) [26, 46] architectures *merge* name resolution with routing to content and services in order to allow consumers to request content objects (CO) or services by name. Routers provide this service using three tables. A content store (CS) lists the COs that are cached locally. A Pending Interest Table (PIT) keeps forwarding state for each Interest (a request for a CO) processed by a router, such that a single copy of an Interest for the same CO is forwarded and responses to Interests can be sent over the reverse paths traversed by the Interests. A forwarding information base (FIB) listing the next hops to name prefixes is used to forward Interests towards content producers.

The simplicity of merging name resolution with routing is very attractive. However, it comes at a big price, in terms of storage requirements associated with FIBs and PITs [9, 33, 35, 37, 38, 41] and additional vulnerabilities to DDoS attacks associated with PITs [19, 41].

Consider a network in which a name-based routing protocol establishes routes to all known name prefixes and to those routers that announce name prefixes being locally available, which we call anchors of the prefixes. Section 3 shows that the path traversed by an Interest when routers maintain FIBs with entries for name prefixes is the same as the path traversed by the Interest if the first router binds the name prefix that provides the best match for the CO name to the name of an anchor of that prefix, and routers forward the Interest towards that anchor.

Section 4 describes CCN-RAMP, which is based on two key innovations. First, CCN-RAMP integrates name resolution with routing to name-prefix anchors using the name-based content routing protocol running in the control plane (e.g., [11, 23, 24]). The approach takes advantage of the result presented in Section 3, and the consequence is that routers can forward Interests using forwarding tables that are orders of magnitude smaller than the FIBs required in NDN and CCNx, and even smaller than the FIBs needed in the IP Internet. Second, CCN-RAMP extends recent results in [16, 17] to eliminate forwarding loops, and to replace PITs with small forwarding tables listing the next hops towards the origins of Interests, without identifying such origins.

Section 5 compares the performance of CCN-RAMP with NDN when either no caching or on-path caching is used in a 153-router network. CCN-RAMP incurs very similar Interest traffic than NDN to retrieve content, requires orders of magnitude fewer entries in forwarding tables than NDN, and needs fewer table look ups to retrieve any given CO than NDN.

ACM-ICN '16, September 26–28, 2016, Kyoto, Japan
© 2016 ACM. ISBN 978-1-4503-4467-8/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2984356.2984359

2. RELATED WORK

Previous work attempting to make the forwarding planes of ICN architectures efficient is vast and excellent reviews exist of this prior work [4, 5, 44]. We only highlight the main points regarding name resolution and routing that motivate our design. In this light, it is important to first note that a major impediment for the IP Internet to provide efficient access to content and services by name is the poor interaction between name resolution and routing that it currently supports. A client must be an integral part of name resolution and is required to bind the name of the CO or service to be requested to the location where the content or service is offered. A client first interacts with a local Domain Name System (DNS) server to obtain the mapping of a domain name to an IP address in order for a request for content or service can be sent to a specific IP address. Adding to this, the DNS is based on a hierarchical, static caching structure of servers hosting the mappings of domain names to addresses built and maintained independently of routing and requiring servers to be configured on how to contact other servers over the Internet.

2.1 ICN Architectures

TRIAD [21] was one of the first projects to advocate using names for routing rather than addresses. Since then, many ICN architectures have been proposed that support name resolution and content routing functionality to enable consumers to ask for content objects (CO) or services by name. These architectures differ on how a CO name is mapped to a producer or source that can provide the requested CO or service, and the way in which paths are established for requests for COs or services and the associated requests.

Some ICN architectures [4, 5, 44], including DONA, PURSUIT, SAIL, COMET, and MobilityFirst, implement name resolution and routing as independent functions. In these architectures, name resolution servers (called by different names) are organized hierarchically, as multi-level DHTs, or along trees spanning the network [5], and consumers and producers contact such servers to publish and subscribe to content in various ways. Consumers obtain the locations of publishers from name resolution servers, and send their content requests to those locations to get the required content or services, and address-based routing is used to establish paths between consumers and subscribers or between resolution servers and subscribers or consumers.

A major limitation of keeping name resolution independent of routing stems from the complexity incurred in keeping name-resolution servers consistent with one another, and allowing consumers and producers to interact with the name-resolution system. Enabling the updates of name-to-address mapping is a non-trivial problem using hierarchical structures, spanning trees, or DHT-based organizations of servers. Another design consideration in these architectures is that a solution is still needed to preserve the anonymity of the sources of requests for COs or services, which may induce additional complexity in the forwarding plane or require the use of the forwarding mechanisms used in NDN and CCNx.

In contrast to most prior ICN architectures, NDN and CCNx *merge* name resolution and routing, such that routers are the facto name resolvers by establishing routes to name prefixes on a hop-by-hop basis. A major advantage of doing this is that it eliminates the complexity of designing and maintaining a network of name-resolution servers that replace the DNS. This merging of functionalities is supported by: (a) a name-based routing protocol operating in the control plane, which updates the entries in FIBs listing the next hops to known name prefixes, and (b) forwarding of Interests based on the longest prefix match (LPM) between the CO name in the Interest and a name prefix listed in the FIBs.

However, as attractive as the simplicity of merging of name resolution with routing in NDN and CCNx is, it comes at a very big price. Because the name of a CO or service is bound directly to a route on a hop-by-hop basis, each router along the path traversed by an Interest must look up a FIB listing the known name prefixes. To operate at Internet scale, FIB sizes in NDN are acknowledged to eventually reach billions of entries [33]. This is easy to imagine, given that the number of registered domains in the IP Internet was more than 300 million by the end of 2015. This is orders of magnitude larger than the largest FIB size for the IP Internet, which is smaller than 600,000 today.

To make matters worse, the name prefixes assumed in ICN architectures are variable length and much more complex than IP addresses. This means that efficient LPM algorithms developed for the IP Internet cannot be applied directly to NDN and CCNx. Indeed, it has been noted that NDN and CCNx cannot be deployed at Internet scale without further advances in technology [28].

2.2 Limitations of Using PITs

The size of a PIT grows linearly with the number of distinct Interests received by a router as consumers pipeline Interests (e.g., to support HD video streams) or request more content, or more consumers request content [9, 35, 37]. Unfortunately, as the following paragraphs summarize, PITs do not deliver substantial benefits compared to much simpler Interest-forwarding mechanisms, and can actually be counter-productive.

We have shown [8, 17] that the percentage of aggregated Interests is minuscule when in-network caching is used, even Interests exhibit temporal correlation. We have also shown that per-Interest forwarding state is not needed to preserve the privacy of consumers issuing the Interests [16, 17].

Supporting multicast content delivery efficiently in the data plane has been viewed as a major reason to use PITs. However, as we demonstrate in [18], maintaining per-Interest forwarding state is unnecessary to implement pull-based multicast content dissemination. In a nutshell, a source-pacing algorithm can be used with routers maintaining per-source rather than per-Interest forwarding state. Routers forwarding traffic from a given multicast source maintain the most recent *multicast-counter value* (*mv*) for the source. A router forwards an Interest towards a source only if the *mv* stated in the Interest is larger than the value it currently stores for the source and discards the Interest otherwise. The net effect is the same as Interest aggregation, but without the large overhead of per-Interest forwarding state [18].

We have also shown [13, 14, 15] that Interest aggregation combined with the Interest-loop detection mechanisms used in NDN and CCNx can lead to Interests being aggregated while traversing forwarding loops without such loops being detected. This results in aggregated Interests "waiting to infinity" for responses that never come. In addition, using PITs makes routers vulnerable to Interest-flooding attacks [19, 38, 40, 41] in which malicious users can send malicious Interests aimed at making the size of PITs explode. Unfortunately, the countermeasures that have been proposed for these attacks [2] simply attempt to reduce the rates at which suspected routers can forward Interests, and this can be used to mount other types of denial-of-service attacks.

2.3 Limitations of Using FIBs Listing Name Prefixes

A number of proposals have been advanced to allow Interest forwarding in NDN based on LPM to keep up with new wire speeds while coping with the required FIB sizes and name-prefix structures. A big challenge for name-based Interest forwarding is to attain 100 Gbps rates or higher, given that FIBs listing name prefixes at Internet scale are much too large to fit into SRAM or TCAM [28].

Several proposals for the implementation of FIBs for NDN rely on the use of tries and massive parallelism in order to avoid bottlenecks in the encoding process needed to use the tries [42, 43]. Other approaches are based on hash tables for FIB lookups [31, 32, 36, 45], which requires larger memory footprints and is not scalable to prefix names with a large number of name components. Hash-based approaches for name-based forwarding are based on DRAM technology and rely on massive parallel processing, because they would require hundreds of MiBs for just a few million prefixes.

Given the major limitations of using LPM in FIBs containing billions of name prefixes, a few proposals have been advanced to either reduce the size of FIBs listing name prefixes or eliminate the use of such FIBs.

Song et al. [33] introduced the concept of "speculative forwarding" based on longest-prefix classification (LPC) rather than LPM. LPC behaves just like LPM when a match is found in the FIB for the name stated in an Interest; however, with LPC a packet is forwarded to a next hop given by the FIB even if no match is found. Unfortunately, as described in [33], Interests may be forwarded along loops. Although the NDN forwarding strategy can prevent Interests from traversing the same forwarding loop multiple times, it cannot guarantee that Interests will not be aggregated while they traverse forwarding loops [14].

SNAMP [3] and PANINI [30] reduce FIB sizes by means of default routes and default-free zones. Edge routers resolve and keep track of local name prefixes, and forward all other interests toward backbone routers that create a default-free zone and map name prefixes to globally-routed names.

TagNet [27] uses content descriptors and host locators for forwarding. Content descriptors are variable-length sets represented with fixed-length Bloom filters for forwarding. Host locators are routing labels assigned to routers and hosts along one or multiple spanning trees. The labeling approach used in TagNet to assign locators is due to Thorup and Zwick [34]. TagNet results in FIBs that are much smaller than the FIBs required in NDN; however, it has a number of limitations. Content requests (Interests) must state the locators of their sources, which eliminates the anonymity provided in NDN and CCNx. Like any other scheme based on compact-routing, the paths traversed over the labeled spanning trees can have some stretch over the shortest paths. The Thorup-Zwick labeling used in TagNet is a depth-first search approach, and entire spanning trees may have to be relabeled after a single link failure. No prior work exists showing that TagNet or similar routing schemes based on interval-routing labels are suitable for large networks subject to topology changes or mobility of hosts and routers.

3. ROUTING TO NAME PREFIXES VS. ROUTING TO ANCHORS

We use the term *anchor* to denote a router that, as part of the operation of the name-based routing protocol, announces the content corresponding to a name prefix being locally available. If multiple mirroring sites host the content corresponding to a name prefix, then the routers attached to those sites announce the same name prefix. However, a router simply caching COs from a name prefix does not announce the name prefix in the name-based routing protocol.

An anchor announcement can be done implicitly or explicitly as part of the operation of the name-based routing protocol running in the control plane, and routers simply caching COs are not anchors. For example, in NDN [24] and other name-based routing protocols based on link-state information [23], routers exchange link-state announcements (LSA) corresponding to either name prefixes or adjacencies to networks or routers. In this context, any router that originates an LSA for a name prefix is an anchor of the prefix. On the other hand, DCR [11] and other name-based content routing protocols (e.g., [29]) based on distance information to name prefixes use the names of anchors to ensure the correctness of the multi-path route computation [12].

Figure 1 shows an example of a content-centric network in which router y is an anchor for prefixes P^*, Q^*, and R^*; and router z is an anchor for prefixes A^*, C^*, and P^*. Dark solid arrowheads show the best next hop towards name prefixes. A red arrowhead indicates the best next hop to name prefixes with COs locally available at router y, and a dashed arrow head indicates the best next hop towards name prefixes with COs locally available at router z. The FIB entries at router p are shown when FIB entries are maintained for all instances of each name prefix, the nearest instances of a name prefix, or only for the anchors of name prefixes.

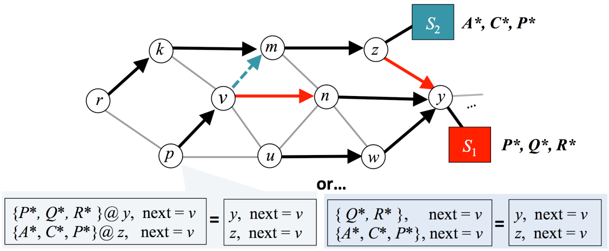


Figure 1: FIB entries for name prefix instances and FIB entries for anchors of name prefix instances

This example illustrates the fact that routes to instances of name prefixes must also be routes to the anchors announcing those instances. Therefore, the paths obtained from FIB entries listing name prefixes are the same as the paths obtained from FIB entries listing the anchors of name prefixes.

It is important to note that a router acts as the anchor of a name prefix over time scales that are many orders of magnitude larger than either the time scale at which congestion varies in a network due to traffic or even topology changes, or the time needed for all routers to respond to congestion changes.

Even though current Internet routing protocols do not handle congestion or multi-path routing well, the necessary and sufficient conditions for minimum-delay routing (or minimum-congestion routing) are well known [10] and practical approaches that provide very good approximations to optimum routing using datagrams have existed for some time [25, 39]. Given the long time periods over which routers act as anchors of name prefixes, these approaches can be used in either routing to name prefixes or routing to the anchors of name prefixes. Furthermore, existing name-based routing protocols can easily apply mechanisms to react to congestion, namely: multi-path routing, maintaining congestion information about local interfaces, and path-based measurements of congestion. Accordingly, it can be safely assumed that congestion-oriented multi-path routing is attained independently of whether routes are established for name prefixes or anchors.

The following theorem formalizes the result illustrated in Figure 1 for the case of shortest-path routing (single path or multi-path) to name prefixes. We show that, if the same name-based routing protocol is used to establish routes to anchors and to name prefixes, the paths traversed by Interests are the same or mostly the same.

THEOREM 1. *The paths to name prefixes obtained using forwarding entries listing the name prefixes are the same as the paths obtained using forwarding entries listing the anchors of name prefixes in a stable network in which a correct name-based routing protocol is executed.*

PROOF. Consider a content-centric network in which forwarding entries list the next hops to the anchors of name prefixes. Assume that the routing protocol computes the paths to all the known anchors by time t_0 and that no changes occur in the network after that time.

Consider a name prefix P for which router a is an anchor, and assume for the sake of contradiction that, at some time $t_1 > t_0$, there is a route from a router r to the instance of prefix P announced by anchor a that is shorter than any of the routes from r to a implied by the forwarding tables maintained by routers at t_1. This is a contradiction to the definition of an anchor and the assumption that the routing protocol computes correct routes to all anchors by time t_0 and no changes occur after that. □

COROLLARY 1. *For any name prefix that has a single anchor, the paths to the name prefix obtained using forwarding entries listing name prefixes are the same as the paths obtained using forwarding entries listing the anchors of name prefixes.*

PROOF. The proof follows from Theorem 1 for the case of a stable topology. If forwarding tables are inconsistent due to network dynamics, the result follows from the one-to-one correspondence between a name prefix and its anchor, given that the prefix is hosted at a single site. □

Figure 2 illustrates a content-centric network in which name prefix P^* is multi-homed. Assume that router o receives an Interest for a CO with a name in prefix P^* from consumer c, and the nearest anchor of P^* to router o is a_i. In the example, D_{ij} denotes the length of the pat from router i to router j, and $D^*_{ra_i}$ denotes the distance from r to a_i along a path that does not include p.

If routing is based on anchor names, router o binds the CO name to anchor a_i. Assume that the shortest path from o to a_i includes router r and that p is the next hop from r to a_i. Consider the case in which, because link (r, p) fails or becomes too congested, router r must find an alternative path to forward an Interest intended for anchor a_i. Anchor a_j is the closest anchor of prefix P^* to router r after the change in link (r, p). Routing based on name prefixes can provide more efficient forwarding than routing based on anchor names only if one of the following conditions is true:

$$D^*_{ra_i} < \infty \;\wedge\; D_{ra_j} < D^*_{ra_i} \qquad (1)$$

$$D^*_{ra_i} = \infty \;\wedge\; D_{ra_j} < D_{ro} + D_{oa_j} \qquad (2)$$

Only a small number of Interests pertaining to name prefixes that are multi-homed may be forwarded more efficiently if routing based on name prefixes is used. The reason for this is that Equations 1 and 2 cannot be satisfied over extended time periods. The name-based routing protocol operating in the control plane must make router o update its distances to anchors reflecting the change in link (r, p). Once router o updates its forwarding table, it must select a_i or another anchor as the new nearest anchor for P^* and Interests would traverse new shortest paths. We observe that this is the case even if mobility of hosting sites or consumers occurs, because the efficiency of Interest forwarding is determined by the distances between the routers attached to consumers and the routers attached to hosting sites (anchors).

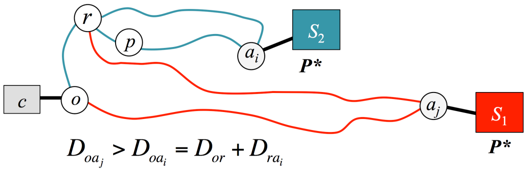


Figure 2: Forwarding Interests based on routes to anchors or routes to name prefixes

4. CCN-RAMP

4.1 Design Overview

CCN-RAMP allows routers to forward Interests by looking up small forwarding tables listing next hops to anchors. Clearly, for this to work, routers need to first obtain the mapping of the CO name stated in an Interest to an anchor of the name prefix that best matches the CO name.

CCN-RAMP takes advantage of the fact that existing name-based routing protocols communicate the anchor of a name prefix as part of an LSA or a distance update for the prefix [11, 23, 24]. Using the information disseminated in the name-based routing protocol, a router builds and maintains two tables: A Forwarding to Anchors Base (FAB) listing the routes to anchors, and a Prefix Resolution Table (PRT) listing the anchors of each name prefix.

A router receiving an Interest from a local consumer (call it origin router) uses its PRT to bind the CO name to the nearest anchor for the name prefix that is the best match for the CO name. To allow relaying routers to use only their FABs to forward Interests, an Interest states the name of the anchor chosen by the origin router. The origin router and other relaying routers forward the Interest as needed using their FABs and the anchor name in the Interest.

Routers in CCN-RAMP use exactly the same amount of routing signaling as routers in NDN and CCNx, and a PRT in CCN-GRAM has as many entries as a FIB in NDN and CCNx. However, the amount of routing signaling is not a problem, because anchors of name prefixes change infrequently and the name-based routing protocol can send updates regarding anchor-prefix bindings independently of updates regarding how to reach anchors. The limitation of using a FIB listing name prefixes is not its size but the need to look up the FIB in real time for each Interest being forwarded, which either requires very expensive memory or renders very slow lookup times.

In NDN and CCNx, each router forwarding an Interest must look up a FIB based on name prefixes because name resolution and routing are merged into one operation. By contrast, name resolution in CCN-RAMP is delegated to the origin routers that receive Interests from local consumers. Each router is in effect a name resolver. An origin router that binds a CO name to an anchor by looking up its PRT in response to an Interest from a local consumer replaces the role of the DNS to resolve a name into an address. However, consumers are relieved from being involved in name resolution. What can take hundreds of milliseconds using the DNS takes just a lookup of the PRT, which can be implemented using tries and slow storage. Once the binding of a CO name to an anchor is done by an origin router, forwarding an Interest involves fast lookups of FABs that can be even smaller than the FIBs used in the IP Internet today because only anchors are listed.

The tradeoff made in CCN-RAMP compared to prior approaches that separate name resolution from address-based routing is the need for each router to store a PRT. This is acceptable, given the cost of memory today and the infrequency with which PRT entries must be updated.

The design of CCN-RAMP eliminates forwarding loops by ordering the routers forwarding Interests based on their distances to destinations [14]. To attain this, each Interest carries the distance to an anchor and FABs list the next hops *and* the distances to anchors.

CCN-RAMP extends our prior work [16, 17] to eliminate the use of PITs while providing the same degree of Interest anonymity enabled in NDN and CCNx. A router maintains a Label Swapping with Anchors Table (LSAT) to remember the reverse paths traversed by Interests, and uses anonymous identifiers (AID) with local scope to denote the origins of Interests. The origin of an Interest is denoted with an AID that is swapped at each hop.

4.2 Assumptions

We make a few assumptions to simplify our description of CCN-RAMP; however, they should not be considered design requirements. For convenience, a request for content from a local user is sent to its local router in the form of an Interest.

Interests are retransmitted only by the consumers that originated them, rather than routers that relay Interests, routers forward Interest based on LPM, and a router can determine whether or not the CO with the exact same CO name is stored locally. Routers know which interfaces are neighbor routers and which are local consumers, and forward Interests on a best-effort basis.

Allowing anchors with only subsets of the COs in the name prefixes they announce requires routers to determine which of the anchors announcing the name prefix actually host the requested CO. This can be accomplished in CCN-RAMP using the multi-instantiated destination spanning trees (MIDST) described in [11, 12]. The pros and cons of allowing anchors to host only subsets of the COs in name prefixes, and the search mechanisms needed to support it, are the subject of another publication. In the rest of this paper, we assume that each anchor of a name prefix is required to have *all* the COs in the name prefix locally available.

The name of content object (CO) j is denoted by $n(j)$ and the name prefix corresponding to the longest prefix match for name $n(j)$ is denoted by $n(j)^*$. The set of neighbors of router i is denoted by N^i.

4.3 Information Exchanged

An Interest forwarded by router k requesting CO with name $n(j)$ is denoted by $I[n(j), AID^I(k), a^I(k), D^I(k)]$, and states the name of the CO ($n(j)$), a fixed-length anonymous identifier ($AID^I(k)$) denoting the origin router of the Interest, the anchor selected by the first router processing the Interest ($a^I(k)$), and the distance ($D^I(k)$) from k to $a^I(k)$.

A data packet sent by router i in response to an Interest is denoted by $DP[n(j), AID^R(i), sp(j)]$ and states, in addition to a CO, the name of the CO being sent ($n(j)$), an anonymous identifier ($AID^R(i)$) denoting the router that should receive the data packet, and a security payload ($sp(j)$) used optionally to validate the CO.

An error message sent by router i in response to an Interest is denoted by $ERR[n(j)$, $AID^R(i)$, $a^R(i)$, $\mathsf{CODE}]$ and states the name of a CO ($n(j)$), an anonymous identifier ($AID^R(i)$) that states the intended recipient of the reply, the selected anchor for the name prefix ($a^R(i)$), and a code (CODE) indicating the reason why the reply is sent. Possible reasons for sending a reply include: an Interest loop is detected, no route is found towards requested content, no content is found, and an upstream link is broken.

4.4 Information Stored

Router i maintains three tables for packet forwarding: A Prefix Resolution Table (PRT^i), a Forwarding to Anchors Base (FAB^i), and a Label Swapping with Anchors Table ($LSAT^i$). If router i has local consumers, it maintains a Local Request Table (LRT^i). Router i maintains a Content Store (CS^i) if it provides content caching locally.

PRT^i is indexed by the known name prefixes advertised by their anchors. Each entry of the PRT^i states the names of the selected anchors that advertised the prefix. Depending on the specific approach, the list may state the nearest anchors or all the anchors of the name prefix. However, with the assumption that anchors must have all COs of the name prefixes they announce, listing the nearest anchors for a name prefix suffices.

FAB^i is indexed by anchor names and each entry in FAB^i states available next hops to the anchor. The distance stored for neighbor q for anchor a in FAB^i is denoted by $D(i, a, q)$. This information is updated by means of a name-based routing protocol running in the control plane.

$LSAT^i$ is indexed by anonymous identifiers denoting origin routers. An anonymous identifier (AID) is simply a fixed-length number. Each entry in $LSAT^i$ states an AID locally created or received in Interests from a previous hop, the previous hop ($PH^i[AID]$) that provided the AID, a next hop ($NH^i[AID]$), the mapped AID ($MAP^i[AID]$) that should be used for the next hop, and the distance ($D^i[AID]$) to the anchor that should receive the forwarded Interests.

LRT^i lists the names of the COs requested by router i on behalf of local consumers. The entry for CO name $n(j)$ states the name of the CO ($n(j)$) and a list of the identifiers of local consumers (denoted by $lc[n(j)]$) that have requested the CO. CS^i lists the COs cached locally. The entry for

CO name $n(j)$ states a pointer to the content of the CO (denoted by $p[n(j)]$).

4.5 Avoiding Forwarding Loops

Let S^i_a denote the set of next-hop neighbors of router i for anchor a. Router i uses the following rule to ensure that Interests cannot traverse forwarding loops, even if the forwarding data maintained by routers regarding name prefixes and anchors is inconsistent or contains routing-table loops.

Anchor-Based Loop-Free Forwarding (ALF): If router i receives $I[n(j), AID^I(k), a^I(k) = a, D^I(k)]$ from router k, it can forward $I[n(j), AID^I(i), a^I(i) = a, D^I(i)]$ if:

1. $AID^I(k) \notin LSAT^i \wedge \exists v \in S^i_a \; (\; D^I(k) > D(i, a, v) \;)$
2. $AID^I(k) \in LSAT^i \; \wedge \; (\; D^I(k) > D^i[AID^I(k)] \;)$

THEOREM 2. *No Interest can traverse a forwarding loop in a content-centric network in which CCN-RAMP is used.*

PROOF. Consider a network in which CCN-RAMP is used and assume that, following the operation of CCN-RAMP, there is a router v_0 that originates an Interest for CO $n(j)$, uses longest match prefix to obtain the name prefix $n(j)^*$, and binds that name prefix to anchor a. The Interest sent by v_0 is $I[n(j), AID^I(v_0), a^I(v_0) = a, D^I(v_0)]$.

For the sake of contradiction, assume that routers in a forwarding loop L of h hops $\{v_1, v_2, ..., v_h, v_1\}$ forward the Interest for CO $n(j)$ originated by v_0 along L, with no router in L detecting that the Interest has traversed loop L.

Given that L exists by assumption, router $v_k \in L$ must forward $I[n(j), AID^I(v_k), a^I(v_k) = a, D^I(v_k)]$ to router $v_{k+1} \in L$ for $1 \leq k \leq h-1$, and router $v_h \in L$ must forward $I[n(j),\ AID^I(v_h),\ a^I(v_h) = a, D^I(v_h)]$ to router $v_1 \in L$. According to ALF, if router v_k forwards Interest $I[n(j), AID^I(v_k), a, D^I(v_k)]$ to router v_{k+1} as a result of receiving $I[n(j),\ AID^I(v_{k-1}),\ a, D^I(v_{k-1})]$ from router v_{k-1}, then it must be true that

$$AID^I(v_{k-1}) \notin LSAT^{v_k} \wedge [\; D^I(v_{k-1}) > D(v_k, a, v_{k+1})]$$

or

$$AID^I(v_{k-1}) \in LSAT^{v_k} \wedge [\; D^I(v_{k-1}) > D^{v_k}[AID^I(v_{k-1}) \;].$$

Similarly, if router v_1 forwards Interest $I[n(j), AID^I(v_1), a, D^I(v_1)]$ to router v_2 as a result of receiving $I[n(j), AID^I(v_h), a,\ D^I(v_h)]$ from router v_h, then

$$AID^I(v_h) \notin LSAT^{v_1} \wedge [\; D^I(v_h) > D(v_1, a, v_2)]$$

or

$$AID^I(v_h) \in LSAT^{v_1} \wedge [\; D^I(v_h) > D^{v_1}[AID^I(v_h)] \;].$$

Given that each router in loop L that forwards an Interest for a given AID for the first time must create an entry in its LSAT, it follows from the above argument that, for loop L to exist and be undetected when each router in the loop uses ALF to forward the Interest originated by router v_0, it must be true that

$$D^I(v_{k-1}) > D^{v_k}[AID^I(v_{k-1})] \; for \; 1 < k \leq h \tag{3}$$

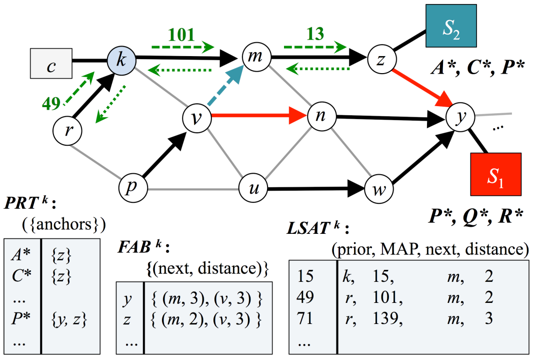
$$D^I(v_h) > D^{v_1}[AID^I(v_h)]. \tag{4}$$

However, Eqs. (3) and (4) constitute a contradiction, because they imply that $D^I(v_k) > D^I(v_k)$ for $1 \leq k \leq h$. Therefore, the theorem is true. □

Theorem 2 is independent of whether the network is static or dynamic, the specific caching strategy used in the network, the retransmission strategy used by content consumers or relay routers after experiencing a timeout or receiving a reply, or whether routers use multiple paths or a single path to forward Interests towards a given anchor. We should also point out that ALF is a sufficient condition to ensure loop-free Interest forwarding, and it is possible that more flexible loop-free forwarding rules could be found. This is the subject of future work.

4.6 Interest Forwarding

Figure 3 illustrates the forwarding of Interests in CCN-RAMP showing two anchors (y and z) and five name prefixes, one of them (P^*) is multi-homed at y and z. The figure shows the forwarding tables used at router k (PRT, FAB and LSAT). Not shown are the content store and the LRT at router k.



A*	{z}
C*	{z}
...	
P*	{y, z}
...	

y	{ (m, 3), (v, 3) }
z	{ (m, 2), (v, 3) }
...	

15	k,	15,	m,	2
49	r,	101,	m,	2
71	r,	139,	m,	3
...				

Figure 3: Interest forwarding in CCN-RAMP

When router k receives an Interest from a local consumer c for a CO with name $n(j)$, it looks up its content store (CS^i) to determine if the CO is stored locally. If the CO is remote, router k adds c to an entry in LRT^k stating the list of consumers that have requested $n(j)$ and proceeds to create Interest $I[n(j),\ AID^I(k),\ a^I(k),\ D^I(k)]$. The LRT entries allow router k to demultiplex responses it receives for AIDs that it originates and send the responses to the correct consumers.

Router k looks up PRT^k for the name prefix that provides the best match and selects an anchor of that prefix to be included in the Interest ($a^I(k)$). The router then looks up FAB^k to obtain the next hop m and the distance to the selected anchor. Router k includes its distance to the anchor ($D^I(k)$) in its Interest, so that forwarding routers can apply ALF as described in Section 4.5. If no entry exists in $LSAT^k$ with Router k as the prior hop, it selects an anonymous identifier ($AID^I(k)$) to denote itself as the origin of the Interest, such that the identifier is not being used by k to denote any other origin of Interests in $LSAT^k$. To select new AIDs, router i maintains a hash table or an array of bits that keeps track of previously used random numbers. An alternative approach could be using a counter that is increased after creating a new mapped AID (MAP).

In the example of Figure 3, router k has forwarded an Interest from a local consumer and used 15 as the anonymous identifier (AID) to identify itself. Router k can use the same AID in all Interests it sends towards any anchor on behalf of local consumers, or use different AIDs.

If router k forwards Interest $I[n(j), AID^I(r), a^I(r), D^I(r)]$ from neighbor r to neighbor m, ALF is satisfied, and no entry for $AID^I(r)$ exists in $LSAT^k$, then router k computes an AID that is not used as the MAP in any entry in $LSAT^k$. Router k then creates the entry for $AID^I(r)$ in $LSAT^k$ stating: $PH^k(AID^I(r)) = r$, $NH^k(AID^I(r)) = m$, $MAP^k\ (AID^I(r)) = n$, and $D^k(AID^I(r)) = D(k, a^I(r), m)$.

Note that only the ingress router receiving an Interest from a local consumer (e.g., router k receiving an Interest from c in the example) needs to look up its PRT, which is of the same size as a FIB in NDN and CCNx.

Forwarding routers use only their FABs and LSATs. Furthermore, a router needs to lookup its FAB only when no forwarding state exists in its $LSAT$ for the AID given in an Interest received from a neighbor router. A forwarding router receiving an Interest looks up its LSAT. If forwarding state is already set in its LSAT for the AID stated in an Interest from a given neighbor, the router forwards the Interest without involving its FAB.

The forwarding state in $LSAT^k$ specifies the next hop and AID to be used to forward an Interest received with a given AID from a previous hop. In the example, router k maps $(AID = 49, prior = r)$ in the Interests received from r to $(AID = 101, next = m)$ in the Interests it forwards to m towards anchor z. The dashed green arrows in Figure 3 show the flow of Interests from r to anchor z and the flow of responses from z to r.

The AID swapping approach adopted in CCN-RAMP simplifies the approaches we introduced in [16, 17]. Its intent is to make Interest forwarding as simple and fast as label swapping in IP networks, without revealing the identities of the consumers or routers that originate the Interests.

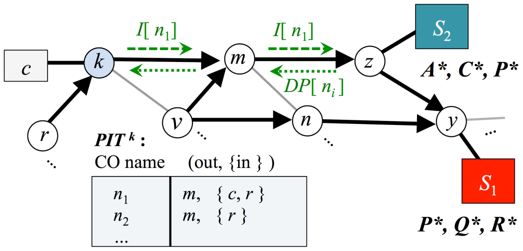


Figure 4: Interest forwarding in NDN

Figure 4 helps to illustrate the level of anonymity provided in NDN and CCNx. Interests in NDN only state the name of the requested CO and a nonce, and the per-Interest forwarding state stored in PITs is what allows routers to forward responses to Interests over the reverse paths traversed by the Interests. A third party monitoring traffic cannot determine the origin of an Interest simply from the information in the headers of Interests. However, the origin of an Interest can be obtained if routers collaborate and trace back the path the Interest traversed using the PIT entries listing the CO name in the Interest.

The same type of anonymity is provided in CCN-RAMP, but without the need for per-Interest forwarding state. A third party monitoring traffic cannot determine the source of an Interest simply by reading the information in the header of the Interest, because a local identifier is used to denote the source of an Interest at each hop. However, routers can collaborate to trace back the origin of Interests by means of the LSAT entries stored by the routers.

4.7 Updating Forwarding State

Algorithms 1 to 4 show the steps taken by routers to maintain the forwarding state needed to forward Interests, COs and error messages. The algorithms assume that FAB^i and PRT^i are initialized and maintained by a routing protocol operating in the control plane (e.g., NLSR [24] or DCR [11]).

Algorithm 1 shows the steps taken by a router to process an Interest received from a local consumer or a neighbor router, which were discussed in Section 4.6. An Interest from a consumer is assumed to specify the name of a requested CO with the rest of the information being nil.

Algorithm 1 Processing Interest from p

```
function Interest_Forwarding
INPUT: $LIST^i$,$FAB^i$, $PRT^i$, $LSAT^i$;
INPUT: $I[n(j), AID^I(p), anchor, D^I(p)]$;
if $p[n(j)] \in CS$ then
  retrieve CO $n(j)$; send $DP[n(j), AID^R(i), sp(j)]$ to $p$
else
  if p is consumer then
    $lc[n(j)] = lc[n(j)] \cup c$;
    for each $a$ by rank in $PRT^i(n(j))$ do
      $anchor = a$; break;
    end for
    $aid = f(anchor)$;
  else
    $aid = AID^I(p)$;
  end if
  $entry = nil$;
  for each $e \in LSAT^i_{aid}(aid)$ do
    if $PH(e) = p$ then
      $entry = e; break$;
    end if
  end for
  $loop = true; noRoute = true$;
  if $entry = nil$ then
    for each $s \in N^i$ by rank in $FAB^i(anchor)$ do
      NoRoute=false;
      if $D^I(p) > D(i, n(j)^*, s)$ (% ALF is satisfied) then
        loop=false; $D^I(i) = D(i, n(j)^*, s)$; $NH = s$;
        break;
      end if
    end for
    select unused random number $map$;
    $entry$ =create entry $LSAT^i[aid, p, map, NH, D(i, n(j)^*, s)]$;
  end if
  if $entry \neq nil$ then
    send $I[n(j), MAP(entry), anchor, D(entry)]$ to $NH(entry)$; return;
  end if
  if $noRoute = false \wedge loop = true$ then
    send $ERR[n(j), AID^I(p), anchor,$ loop] to $p$ ;
  else
    send $ERR[n(j), AID^I(p), anchor,$ no route] to $p$
    (% No route to $n(j)^*$ exists);
  end if
end if
```

Algorithm 2 shows the steps taken when a data packet from router s is received at router i. Like an interest, a data packet contains an anonymous identifier $AID^R(s)$. Router looks up $LSAT^{(}i)$ for $AID^R(i)$. If no entry with $MAP = AID^R(i)$ exists, the router does not forward the packet any further. If a matching entry is found, the router checks if the previous hop stated in the $LSAT^{(}i)$ entry is a neighbor router or the router itself. If it is a neighbor node, it forwards the packet to the previous hop PH stated in the matched *entry*. Otherwise, the data packet is forwarded to the local consumers listed in $lc[n(j)]$ of the entry for $n(j)$ in LRT^i.

Algorithm 3 shows the steps taken when the link connecting router i to router s fails. In such a case, for each entry in $LSAT^i$ that states the next hop as router s, an error message including the AID of the $entry$ is created and is sent back to the previous hop stated in the AID entry. This way, router i informs neighbor routers of the link failure. Router i also invalidates all the matching entries in $LSAT^i$.

Algorithm 4 shows the steps followed after an error message from router s is received at router i. The received error message contains an AID set by the neighbor router. Router i looks up $LSAT^i$ and for any entry with $MAP = AID^R(s)$, it creates an error message containing $AID(entry)$ and sends it to the previous hops or the local consumers. The router also invalidates the entry found.

Algorithm 2 Processing data packet from router s

```
function Data Packet
INPUT: $LIST^i$, $LST^i$, $DP[n(j), AID^R(s), sp(j)]$;
[o] verify $sp(j)$;
[o] if verification with $sp(j)$ fails then
     discard $DP[n(j), AID^R(s), sp(j)]$;
entry = $LSAT^i_{map}(AID^R(s))$; ( %LST in case of CCN-RAMP)
if $entry = nil$ then
   $drop$; $return$;
end if
if $preHop(entry) = local$ (% router $i$ is the origin) then
   for each $c \in lc[n(j)]$ do
      send $DP[n(j), nil, sp(j)]$ to $c$; $lc[n(j)] = lc[n(j)] - \{c\}$
   end for
else
   send $DP[n(j), AID(entry), sp(j)]$ to $preHop(entry)$;
end if
store CO in CS
```

Algorithm 3 Failure of link l connected to router i to p

```
function Data Packet
INPUT: $LSAT^i$;
for each $entry \in LSAT^i$ with $NextHop(entry) = p$ do
   $preHop = getNodeID(AID(entry))$;
   send $ERR[nil, AID(entry),$ link failure$]$ to $preHop$
   INVALIDATE(entry);
end for
```

Algorithm 4 Processing Error Message from Router s at router i

```
function ERR
INPUT: $LSAT^i$, $ERR[n(j), AID^R(s), reason]$;
for each $entry \in LSAT^i_{map}(AID^R(s))$ do
   $preHop = getNodeID(AID(entry))$;
   if $preHop = local$ (% router $i$ was the origin of the Interest)
   then
      for each $c \in lc[n(j)]$ do
         send $DP[n(j), c, sp(j)]$ to $c$; $lc[n(j)] = lc[n(j)] - \{c\}$
      end for
   else
      send $ERR[nil, AID(entry),$ reason$]$ to $preHop$;
   end if
   INVALIDATE(entry);
end for
```

5. PERFORMANCE COMPARISON

We implemented CCN-RAMP in ndnSIM [1] based on Algorithms 1 to 4 and used the NDN implementation in ndnSIM without modifications to compare NDN with CCN-RAMP. We also compare CCN-RAMP against our previous proposal for the elimination of PITs using datagrams, CCN-GRAM [17], which also relies on name-based routing in the control plane and uses FIBs listing the next hops to known name prefixes to forward Interests.

The performance metrics used for comparison are the average sizes of forwarding tables, the average number of table lookups needed to obtain one CO, the average end-to-end delays, and the average number of Interests sent by routers. DCR [11] is used in the control plane to update FIBs listing name prefixes in NDN and CCN-GRAM, or the Prefix Resolution Tables (PRT) and Forwarding to Anchor Bases (FAB) used in CCN-RAMP. Accordingly, we do not need to consider the signaling overhead of the name-based routing protocol, because it is exactly the same in the three approaches we consider.

We considered networks with no caching and with on-path caching, with each cache being able to store only 1000 COs. We used the AT&T network topology, which is considered to be a realistic topology for simulations [22]. This topology includes 153 nodes and 184 point-to-point links with 30 ms delay. To reduce the effects derived from sub-optimal implementations of CCN-RAMP, NDN, or CCN-GRAM, we set the data rate of point-to-point links to 10Gbps.

We selected 70 nodes randomly to have a consumer application simulating local consumers. All consumers generate Interests requesting COs from all name prefixes following a Zipf distribution with parameter $\alpha = 0.7$. The total number of COs is only 10^7, with 1000 COs per name prefix. We used a relatively small content population, because using larger numbers for COs and name prefixes would just make CCN-RAMP look better compared to NDN.

We selected 20 of the nodes randomly to be anchors of 500 different name prefixes each, and each name prefix has a single anchor.

Given that prefixes are not multi-homed in our simulation scenarios, the results in Section 3 indicate that the paths traversed by Interests should be the same whether forwarding tables maintain routes to anchors or to prefixes. This is the case for both single-path and multi-path routing. Accordingly, for simplicity, our simulation experiments assume single-path routing and static topologies. The difference between CCN-RAMP and CCN-GRAM [17] (its counter-part based on FIBs listing name prefixes) is that the latter incurs more forwarding overhead by its use of FIBs.

5.1 Average Table Sizes

Figure 5 shows the average table sizes for NDN, CCN-RAMP, and CCN-GRAM on a logarithmic scale as a function of the rate at which Interests arrive at routers with local consumers, ranging from 100 to 2000 Interests per second.

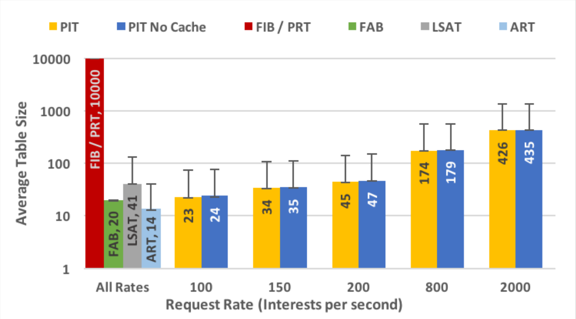


Figure 5: Average number of entries in forwarding tables for NDN, CCN-GRAM, and CCN-RAMP

The number of entries in the PRTs used in CCN-RAMP is the same as the number of entries in the FIBs used in NDN and CCN-GRAM, because both list an entry for each known name prefix. Given that the topology contains 20 producer nodes and each node has 500 different name prefixes, each PRT and FIB table is expected to have 10,000 entries.

For CCN-RAMP, the average number of FAB entries for all Interest rates is only 20 and the average size of an LSAT, which is used to forward responses to Interests back to consumers, is only 41 entries for all values of Interest rates. The small sizes of FABs and LSATs should be expected, because there are only 20 routers acting as anchors, and each router acts as a relay of only a fraction of the paths to such anchors.

The average size of the forwarding table used in CCN-GRAM to send responses to Interests towards consumers (called ART) is only 14 entries for all Interest rates.

By contrast, the number of PIT entries depends on network conditions and traffic load. The average PIT size varies from 23 or 24 to 426 or 435, depending on whether on-path caching is used. The size of PITs at some core routers can be more than 1000 entries when the request rate at routers with local consumers is 2000 Interests per second. Interestingly, the average number of PIT entries is not much smaller when on-path caching is used compared to the case in which no caching is used.

5.2 Average Number of Table Lookups

Figure 6 shows the average number of table lookups required to retrieve a single CO, and includes all lookups done in forwarding the Interest for the CO and sending back the corresponding CO to the requesting origin router.

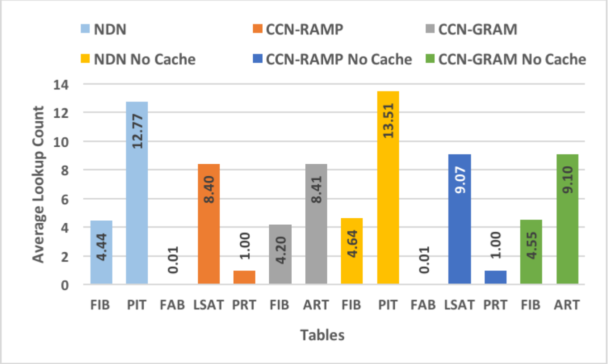


Figure 6: Average number of table lookups needed to retrieve one CO

In NDN, forwarding an Interest requires a PIT lookup at each hop along the path from consumer to caching site or anchor. If no PIT entry is found, a FIB lookup is required to obtain the next hop for the Interest. Forwarding a data packet sent from an anchor or a caching site requires a PIT lookup at every hop along the way to the consumer.

In CCN-RAMP, retrieving a remote CO includes one PRT lookup at the router that receives the Interest from a local consumer. That router binds the CO name to an anchor and each router along the path towards the anchor must do one FAB lookup to forward the first Interest with an AID that does not exist in the LSAT of the router, and one LSAT lookup for every Interest being forwarded. Once forwarding state is established along a path from an originating router to an anchor, no FAB lookups are needed for Interests that carry AIDs already listed in the LSATs of the relaying routers. As Figure 6 shows, the average number of FAB lookups per Interest is a very small fraction, because only a small fraction of Interests are forwarded without having any forwarding state already established in the LSATs of routers.

In CCN-GRAM, retrieving a remote CO involves one FIB lookup at each hop along the path from consumer to caching site or anchor, as well as a lookup of the ART in order to carry out the proper swapping of AIDs. Forwarding a data packet sent from an anchor or a caching site requires only an ART lookup at each hop along the way to the consumer.

Compared to NDN and CCN-GRAM, CCN-RAMP results in relay routers doing fewer lookups of tables that are three orders of magnitude smaller than FIBs listing name prefixes, even for the small scenario we consider.

It can be inferred from Figure 6 that the average hop count for paths traversed by Interests is around four or five hops, because this is the approximate number of average FIB lookups needed in NDN and CCN-GRAM when no caching is used. As should be expected, the average number of table lookups in NDN, CCN-GRAM, and CCN-RAMP is slightly larger when no on-path caching is available, because the average paths between consumers requesting COs and the anchors are longer than the average paths between consumers and caching sites.

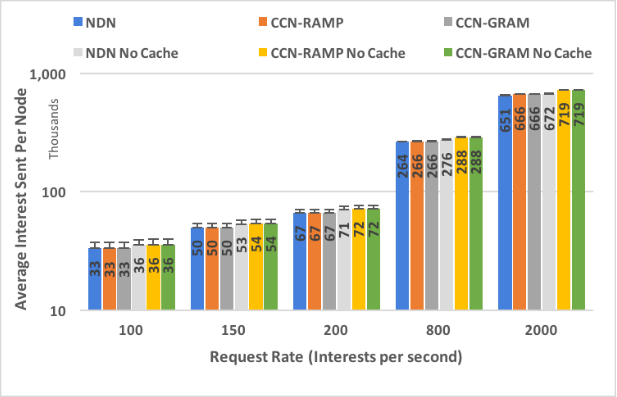


Figure 7: Average number of Interests forwarded per router

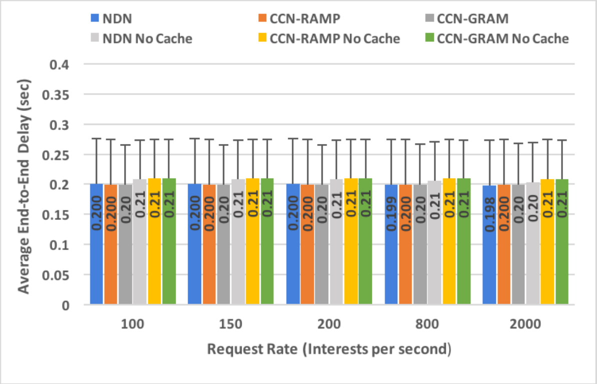


Figure 8: Average end-to-end delays

5.3 Average Number of Interests and End-to-End Delays

Figure 7 shows the average number of Interests sent by each router. As the results indicate, the average encumber of Interests sent by each router is essentially the same for all three approaches and the percentage of Interests that benefit from aggregation using PITs is insignificant.

Figure 8 shows that the average end-to-end delays are very similar in all cases for NDN and CCN-RAMP. Given that the simulations assume zero delays for table lookups (i.e., the differences in forwarding table sizes are not taken into account), these simulation results indicate that the paths traversed by Interests are the same for NDN, CCN-GRAM, and CCN-RAMP. This confirms that the paths traversed by Interests when routers maintain FIBs with entries for name prefixes tend to be the same as the paths traversed by Interests if the origin routers select the anchors of name prefixes and routers forward Interests towards anchors.

6. CONCLUSIONS

Scaling has been identified as a major research problem for content-centric networks [20]. We introduced CCN-RAMP, a new approach to content-centric networking that can be deployed at Internet scale and is able to handle billions of name prefixes, because it eliminates the need to lookup large FIBs listing name prefixes, and the use of PITs that make routers vulnerable to DDoS attacks on the routing infrastructure. CCN-RAMP provides all the benefits sought by NDN and CCNx, including native support of multicasting without the need for a new multicast routing protocol (see [17, 18]). In contrast to NDN and CCNx, Interests cannot traverse forwarding loops and no Interest-flooding attacks can be mounted.

The results of simulation experiments based on implementations of NDN, CCN-GRAM, and CCN-RAMP in ndnSIM show that CCN-RAMP is more efficient than NDN and CCN-GRAM. CCN-RAMP rendered similar end-to-end delays, incurred similar Interest overhead in the data plane, and resulted in forwarding state with a number of entries that can be orders of magnitude smaller than the forwarding state required in NDN.

Our results open up several research avenues on content-centric networking at Internet scale. Important next steps include: analyzing the performance impact of the dynamics of name-based routing protocols adapting to congestion and topology changes, analyzing the impact of multihoming of name prefixes, enabling hierarchical name-based routing in the control plane, defining the role of autonomous systems, and analyzing policy-based routing and forwarding across autonomous systems.

7. REFERENCES

[1] A. Afanasyev et al., "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] A. Afanasyev et al., "SNAMP: Secure Namespace Mapping to Scale NDN Forwarding," *in Proc. IEEE Global Internet Symposium ‘15*, 2015.

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

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

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

[7] A. Dabirmoghaddam et al., "Understanding Optimal Caching and Opportunistic Caching at The Edge of Information Centric Networks," *Proc. ACM ICN ‘14*, Sept. 2014.

[8] A. Dabirmoghaddam et al., "Characterizing Interest Aggregation in Content-Centric Networks," *Proc. IFIP Networking 2016*, May 2016.

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

[10] R.G. Gallager, "A Minimum delay Routing Algorithm Using Distributed Information," *IEEE Trans. Commun.*, 1977.

[11] J.J. Garcia-Luna-Aceves, "Name-Based Content Routing in Information Centric Networks Using Distance Information," *Proc. ACM ICN ‘14*, Sept. 2014.

[12] J.J. Garcia-Luna-Aceves, "Routing to Multi-Instantiated Destinations: Principles and Applications," *Proc. IEEE ICNP ‘14*, Oct. 2014.

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

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

[15] J.J. Garcia-Luna-Aceves, "Eliminating Undetected Interest Looping in Content Centric Networks," *Proc. IEEE NOF ‘15*, Sept. 30-Oct. 2, 2015.

[16] J.J. Garcia-Luna-Aceves and M. Mirzazad-Barijough, "A Light-Weight Forwarding Plane for Content Centric Networks," *Proc. IEEE ICNC ‘16*, Feb. 2016.

[17] J.J. Garcia-Luna-Aceves and M. Mirzazad-Barijough, "Content-Centric Networking Using Anonymous Datagrams," *Proc. IFIP Networking ‘16*, May 2016.

[18] J.J. Garcia-Luna-Aceves and M. Mirzazad-Barijough, "Efficient Multicasting in Content-Centric Networks Using Datagrams," *IEEE Globecom ‘16*, Washington, D.C., 4–8 Dec. 2016.

[19] P. Gasti et al., "DoS and DDoS in Named Data Networking," *Proc. IEEE ICCCN ‘13*, 2013.

[20] A. Ghodsi et al., "Information-Centric Networking: Seeing The Forest from The Trees," *Proc. ACM Hotnets ‘11*, 2011.

[21] M. Gritter and D. Cheriton, "An Architecture for Content Routing Support in The Internet," *Proc. USENIX Symposium on Internet Technologies and Systems*, Sept. 2001.

[22] O. Heckmann et al., "On Realistic Network Topologies for Simulation," *Proc. ACM SIGCOMM MoMeTools ‘03*, Aug. 2003.

[23] E. Hemmati and J.J. Garcia-Luna-Aceves, "A New Approach to Name-Based Link-State Routing for Information-Centric Networks," *Proc. ACM ICN ‘15*, Oct. 2015.

[24] V. Lehman et al., "A Secure Link State Routing Protocol for NDN," Technical Report NDN-0037, Jan. 2016.

[25] S. Murthy and J.J. Garcia-Luna-Aceves, "Congestion-Oriented Shortest Multipath Routing," *Proc. IEEE INFOCOM ‘96*, 1996.

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

[27] M. Papalini et al., "Scalable Routing for Tag-Based Information Centric Networking," *Proc. ACM ICN ‘14*, Sept. 2014.

[28] D. Perino and M. Varvello, "A Reality Check for Content Centric Networking," *ACM ICN ‘11*, 2011.

[29] J. Raju et al., "System and Method for Information Object Routing in Computer Networks," U.S. Patent 7,552,233, June 23, 2009

[30] T.C. Schmidt et al., "Let's Collect Names: How PANINI Limits FIB Tables in Name Based Routing," *Proc. IFIP Networking ‘16*, May 2016.

[31] W. So et al., " Toward Fast NDN Software Forwarding Lookup Engine Based on Hash Tables," *Proc. ACM ANCS ‘12*, 2012.

[32] W. So et al., "Named Data Networking on A Router: Fast and DoS-Resistant Forwarding with Hash Tables," *Proc. ACM ANCS ‘13*, 2013.

[33] T. Song et al., "Scalable Name-Based Packet Forwarding: From Millions to Billions," *Proc. ACM ICN 2015*, Sept. 2015.

[34] M. Thorup and U. Zwick, "Compact Routing Schemes," *Proc. ACM Symp. Parallel Algorithms and Architectures*, 2001.

[35] C. Tsilopoulos et al., "Reducing Forwarding State in Content-Centric Networks with Semi-Stateless Forwarding," *Proc. IEEE INFOCOM ‘14*, April 2014.

[36] M. Varvello et al., "Caesar: A Content Router for High Speed Forwarding," *Proc. ACM ICN ‘12*, 2012.

[37] M. Varvello et al., "On The Design and Implementation of a Wire-Speed Pending Interest Table," *Proc. IEEE Infocom NOMEN Workshop ‘13*, April 2013.

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

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

[40] M. Wahlisch et al., "Lessons from the Past: Why Data-driven States Harm Future Information-Centric Networking," *IFIP Networking ‘13*, May 2013.

[41] M. Wahlisch et al., "Backscatter from The Data Plane? Threats to Stability and Security in Information-Centric Network Infrastructure," *Computer Networks*, Vol. 57, No. 16, Nov. 2013.

[42] Y. Wang et al., "Scalable Name Lookup in NDN Using Effective Name Component Encoding," *Proc. ICDCS ‘12*, 2012.

[43] Y. Wang et al., "Wire Speed Name Lookup: A GPU-based Approach," *USENIX NSDI ‘13*, 2013.

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

[45] H. Yuan and P. Crowley, "Reliably Scalable Name Prefix Lookup," *proc. IEEE ICCCN ‘12*, 2012.

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