March 16, 2014
PARC

Name-Based Content Routing in Information Centric Networks Using Distance Information

J.J. Garcia-Luna-Aceves [1]

Abstract
Prior proposals for name-based content routing in information-centric networks (ICN) require flooding of content requests, or the exchange of information about the locations of content replicas and some routing information regarding the topology of the network. The Distance-based Content Routing (DCR) protocol is introduced, which enables routers to maintain multiple loop-free routes to the nearest sites advertising a named data object or name prefix, and establish content delivery trees over which all sites advertising the same named data object or name prefix can be contacted. In contrast to all prior routing solutions for ICNs, DCR operates without requiring routers to establish overlays, know the network topology, use complete paths to content replicas, or know about all the sites storing replicas of named content. It is shown that DCR is correct and that is far more scalable than prior name-based routing approaches for ICNs, in terms of the speed with which correct routing tables are obtained and the amount of signaling incurred.

Keywords
Content centric networks, information centric networks, name-based content routing, multi-path routing, name prefix routing, loop-free routing

[1] *Palo Alto Research Center*
***Email address**: jjgla@parc.com

Contents

1. Introduction

In response to the shift in Internet usage patterns over the past few years from host-oriented communication to peer-to-peer and user-generated content, several information centric network (ICN) architectures have been proposed [1, 32, 3] as alternatives to the current Internet architecture. The goal of ICN architectures is to enable access to content and services by name, independently of their location, in order to improve system performance and end-user experience.

At the core of all ICN architectures are name resolution and routing of content, and several approaches have been proposed. In some ICN architectures, the names of data objects are mapped into addresses by means of directory servers, and then address-based routing is used for content delivery. By contrast, a number of ICN architectures use name-based routing of content, which integrates name resolution and content routing. With name-based routing, some of the routers (producers or caching sites) advertise the existence of local copies of named data objects (NDO) or name prefixes denoting a set of objects with names sharing a common prefix, and routes to them are established; the consumers of content issue content requests that are forwarded along the routes to the routers that issued the NDO or name prefix advertisements.

Section 2 summarizes the prior work on name-based routing in ICNs. Interestingly, no prior work has been reported using only distance information. This is perhaps due to the commonly-held view that routing based on distance vectors cannot provide multiple routes to destinations replicated multiple times in a graph, as is the case of cached or mirrored content in an ICN.

Our first contribution consists of showing that efficient name-based routing to the nearest instances of content can be attained using only shortest distances to content, without requiring routers to know the network topology, exchange path information, maintain routes to all network sites, or even know about all the sites storing a copy of a given piece of content. Section 3 introduces the first name-based content routing approach for ICNs based solely on distances to the nearest instances of content, which we call *Distance-based*

Content Routing (DCR).

The mechanisms used in the data plane or the uses of name-based routing in ICNs may require the ability to route to any or to all sites advertising a given NDO or name prefix. Our second contribution is introducing an integrated approach for routing to *any or all* sites advertising the same NDO or name prefix in an ICN. DCR builds *anchor based trees* (ABT) using signaling that is much more efficient than the signaling introduced in the past for maintaining shared multicast trees (e.g., CBT [2]) or the spanning-tree approach for publish-subscribe signaling introduced for content-based networking (CBN) [5].

It has been argued that loop-freedom in the control plane is not important in ICNs (e.g., [18, 15]). Our third contribution is showing that content requests may not reach their destinations if the control plane provides multiple routes to content without guarding against routing-table loops, and the data plane forwards content requests on a hop-by-hop basis over a single interface using only the content name, or the content name and a nonce to identify a request. Section 6 shows that DCR provides multiple paths to NDOs or name prefixes without ever creating a routing-table loop, and that it converges to shortest paths to the nearest copies of content over which content requests can flow.

Section 7 compares the communication and time complexities of DCR and routing approaches for ICNs based on DHTs, link-state routing, and distance-vector routing. DCR incurs far less signaling overhead and is much faster to converge to correct routing tables than prior approaches, because it does not require routers to know the network topology or all the sites where content replicas are stored. The results of a simulation experiment comparing DCR with link-state routing is also presented to illustrate the benefits of not requiring routers to communicate topology information or the location of all replicas of content.

2. Related Work

Directed Diffusion [14] was one of the first proposals for name-based routing of content. Requests for named content (called interests) are diffused throughout a sensor network, and data matching the interests are sent back to the issuers of interests. DIRECT [29] uses a similar approach to named-based content routing in MANETs subject to connectivity disruption. Nodes use opportunistic caching of content and flood interests persistently within and across connected network components. LFBL [20] works in much the same way in connected MANETs.

Gritter and Cheriton [13] proposed one of the earliest proposals for name-based routing of content; namely, a name-based routing protocol (NBRP) as an extension of BGP. In essence, name-prefix reachability is advertised among content routers, and path information is used to avoid permanent loops. Another early development on name-based routing of content was the CBCB (combined broadcast and content based) routing scheme for content-based networking [5, 6, 7]. CBCB consists of two components. First, a spanning tree of the network or multiple per-source trees based spanning the network are established. Then publish-subscribe requests for content based on predicates are sent between consumers and producers of content over the tree(s) established in the network.

The ICN architectures proposed recently advocate various ways to accomplish name resolution and routing, and all of them use on-path caching of content [3], [32]. We summarize representative approaches below.

DONA [17] uses flat names for content and either global or local IP addressing and routing to operate. If only local IP routing is used, content requests (FIND messages) gather autonomous-system (AS) path information as they are forwarded, and responses are sent back on the reverse paths traversed by requests. Within an AS, IP routing is used.

Content Centric Networking [15] assumes the use of distributed routing protocols to build the routes over which content requests (Interest messages) are forwarded. A content request (called "Interest") may be sent over one or multiple routes to a name prefix. The use of a link-state routing approach for intra-domain routing was advocated in the original design of CCN, such that routers describe their local connectivity and adjacent resources (content); and adding content prefixes to BGP was proposed for inter-domain content routing.

Several ICN projects have content routing modalities based on the original CCN routing approach (e.g., [8, 9, 11, 22, 27]). NLSR [18] and OSPFN [30] are two protocols for name-based routing of content based on this approach. Routers exchange topology information by flooding two types of link states advertisements (LSA). LSAs can describe the state of physical links just as it is done in traditional link-state routing protocols. In addition, routers flood LSAs about the prefixes for which they have local copies.

To some extent, the routing approach in the Mobility First project [21] is similar to DONA and NBRP, in that it requires using either network addresses or source routing or partial source routing. Several ICN projects (e.g., [25, 27]) have addressed content routing modalities based on distributed hash tables (DHT) running in overlays over the physical infrastructure to accomplish name-based routing. The DHTs are built using underlying routing protocols that discover the network topology (e.g., OSPF).

We observe that all the content routing approaches proposed to date require exchanging information about the physical network in one way or another. More specifically, they use one or more of the following types of mechanisms: (a) maintaining paths to named content or using source routes to content; (b) flooding of information about the network topology and the location of replicas of content; (c) flooding of content requests; (d) establishing trees spanning the network over which name-based publish-subscribe signaling is performed; and (e) maintaining overlays for distributed hash tables (DHT).

3. DCR

3.1 Overview

The operation of DCR assumes that: (a) each network node is assigned a name with a flat or hierarchical structure; (b) each piece of content is a *named data object* (NDO) that can be requested by name; (c) NDOs can be denoted using either flat or hierarchical naming, and the same naming convention is used for the entire system; and (d) routers cache content opportunistically.

If flat names were used to denote NDOs, routers would need to advertise NDOS individually.This calls for indirection mechanisms to allow routing to scale with the number of NDOs. Such mechanisms can be incorporated in DCR, but their definition is beyond the scope of this paper.

There are various ways in which routers can advertise NDOs if hierarchical naming is used. Routers can be allowed to advertise a name prefix only if they have local copies of all NDOs associated with the prefix. Alternatively, routers with subsets of the NDOs corresponding to a name prefix can be allowed to advertise the name prefix. In the first case, a single site can be contacted to resolve requests for any of the NDOs associated with a prefix. By contrast, in the second case, the request must be sent to *all* routers that have advertised the name prefix if the nearest router advertising the prefix does not have a local copy of the NDO requested.

Because DCR provides multiple loop-free routes to the nearest replica or to all replicas of an NDO or name prefix, it can be used to support any type of data-plane strategy for the handling of content requests and content delivery. The way in which content requests are forwarded, how NDOs are sent back in response to requests, and how flow and congestion control are established in the ICN are topics beyond the scope of this paper.

For simplicity, we denote the name of a specific NDO or a name prefix simply as *prefix*. A router that advertises storing some or all the content corresponding to a prefix is called an *anchor* of the prefix. An anchor of a prefix originates distance updates for the prefix periodically; the update includes a sequence number used to avoid routing-table loops. Only the anchors of prefixes are allowed to change the sequence numbers associated with the distances to the prefixes they announce. A router running DCR maintains multiple loop-free routes to one or more anchors for each known prefix.

3.2 Information Stored and Exchanged

In the description of DCR, we denote the lexicographic value of a name i by $|i|$, the set of neighbors of router i by N^i, and the set of next hops of router i for prefix j by S^i_j. The link from router i to router k is denoted by (i,k) and its cost is denoted by l^i_k. The cost of the link (i,k) is assumed to be a positive number that can be a function of administrative constraints and performance measurements made by router i for the link. The specific mechanism used to update l^i_k is outside the scope of this paper.

A router i running DCR maintains four main tables: (a) a *link cost table* (LT^i) listing the cost of the link from router i to each of its neighbors; (b) a *neighbor table* (NT^i) stating routing information reported by each neighboring neighbor for each prefix; (c) a *routing table* (RT^i) that stores routing information for each known prefix; and (d) a *multipoint routing table* (MRT^i) that stores routing information about anchor-based trees created for those prefixes requiring multipoint communication support.

The entry in LT^i for link (i,k) is denoted by LT^i_k, and consists of the name of neighbor k and the cots of the link to it (l^i_k). The row of NT^i for prefix j is denoted by NT^i_j. The information reported by neighbor $k \in N^i$ for prefix j is denoted by NT^i_{jk} and consists of routing information for the nearest anchor and the root anchor of the prefix. The routing information for the nearest anchor consists of: the distance from neighbor k to j (d^i_{jk}); an anchor (a^i_{jk}) storing j; and the sequence number created by a^i_{jk} for j (sn^i_{jk}). The routing information for the root anchor of the prefix consists of: a root anchor (ra^i_{jk}); the distance from neighbor k to that anchor (rd^i_{jk}); and the sequence number created by ra^i_{jk} for j (rsn^i_{jk}).

Router i updates its distance table after receiving an update from neighbor k regarding prefix j or any other input event affecting the same information stored in N^i_{jk}. Router i stores the new information reported by neighbor k if the neighbor sent an up-to-date sequence number by the reported anchor of the prefix. Otherwise, router i does not trust the update, resets the information from that neighbor for the prefix, and schedules an update to correct the neighbor.

The row for prefix j in RT^i is denoted by RT^i_j and specifies: (a) the name of the prefix (j); (b) the routing update information for prefix j (RUI^i_j); (c) the list of neighbors that are valid next hops (S^i_j); and (d) an anchor list (A^i_j) that stores a tuple for each different anchor reported by any next-hop neighbor. The nth tuple in the anchor list is denoted by $A^i_j[n]$ and specifies the anchor and sequence number reported by that anchor. The anchor name and sequence number in that tuple are denoted by $a(A^i_j[n])$ and $s(A^i_j[n])$, respectively.

RUI^i_j states: (a) a flag for each neighbor k denoting whether or not the information needs to be sent in an update to neighbor k (up^i_{jk}); (b) a successor-set counter (sc^i_j) stating the number of valid next hops to j (i.e., $sc^i_j = |S^i_j|$); (c) the ordered distance from i to j (od^i_j); (d) the current distance to j (d^i_j); (e) the anchor of j that has the smallest name among those that offer the shortest distance to j (a^i_j); and (f) the sequence number created by a^i_j for j (sn^i_j).

The entry for prefix j in MRT^i specifies: (a) the name of prefix j; (b) the multipoint update information for prefix j (MUI^i_j); and (c) the list of neighbor routers that have joined the ABT for the prefix (ABT^i_j). MUI^i_j states the root anchor of j (ra^i_j), the distance to the root anchor (rd^i_j), and the sequence number created by ra^i_j for prefix j (rsn^i_j).

An update message sent by router i to neighbor m consists of the name of router i; a message sequence number (msn^i)

used to identify the message; and a list of updates, one for each prefix that needs updating. An update for prefix j sent by router i is denoted by U^i_j and states: the name of the prefix j (j); the distance to j (ud^i_j); an anchor (ua^i_j); and the sequence number created by ua^i_j for prefix j (usn^i_j).

An update message from neighbor k received at i is denoted by $U^i(k)$. The associated message sequence numbers is denoted by $msn^i(k)$, and the entry for prefix j in the update message is denoted by U^i_{jk}. The update entry U^i_{jk} consists of the tuple $[j,\, ud^i_{jk},\, ua^i_{jk},\, usn^i_{jk}]$, which states the prefix name, the distance to it, the anchor for the prefix, and the sequence number assigned to the prefix by the anchor.

Router i sends update messages periodically containing all the updates made to RT^i since the last update message was sent. For each neighbor k, router i includes in an update message for that neighbor all the routing-table entries that have been marked as modified since the previous update was sent to the neighbor. A router can avoid sending its updates at the same time as the other routers in the network using any of the mechanisms proposed in the past for the avoidance of broadcast storms.

3.3 Routing to Nearest Copies of Prefixes

DCR uses sequence numbers for routers to determine which updates carry the most recent information about a destination. Many approaches have been proposed in the past for loop-free routing in networks using distance information ordered by means of sequence numbers (e.g., [23], [24]). However, the mechanisms proposed in the past cannot be applied directly to name-based routing of content that may be replicated anywhere in the network, because an intended destination (i.e., an NDO or name prefix) is multi-homed. For example, using sequence numbers as in DSDV does not work because different sites with copies of the same content (destination) may issue updates with different sequence numbers.

To prevent the creation of routing-table loops when prefixes can be replicated anywhere in the network, DCR establishes a lexicographic ordering of the distances to any given prefix reported by routers based on the distance values, anchor names, and sequence numbers created by the anchors. Instead of remembering the most recent sequence number for a given destination, a router maintains the sequence numbers created by *all* the anchors reported by its neighbors. The information about a given anchor of a prefix is deleted after a finite time that is long enough to ensure that up-to-date information about valid anchors of the prefix is received, before anchor information is deleted.

In a nutshell, a router can select neighbors as next hops to prefixes only if they report up-to-date information and offer shorter distances to the prefixes or the same distances but have smaller names. To address the case in which routers are unable to find viable next hops to some prefixes as a result of topology or prefix dynamics, anchors send updates about their prefixes periodically and increment the sequence numbers they assign to their prefixes.

Let $sn(m)$ denote the sequence number associated with an anchor m in the set of anchors known to router i for prefix j (A^i_j), and let od^i_j denote the smallest value assigned to d^i_j based on distances reported by neighbors. The following condition is sufficient to ensure that no routing-table loops are ever created when routers change their next hops.

Successor-Set Ordering Condition (SOC):
Neighbor $k \in N^i$ can become a member of S^i_j (i.e., be a next hop to prefix j) if one of the following two statements is true:

$$od^i_j < \infty \wedge [\, d^i_{jk} < od^i_j \vee (\, d^i_{jk} = od^i_j \wedge |k| < |i| \,)\,] \wedge [\, \forall m \in A^i_j \; (\, a^i_{jk} \neq m \vee s^i_{jk} \geq sn(m) \,)\,] \quad (1)$$

$$od^i_j = \infty \wedge [\, d^i_{jk} < od^i_j \,] \wedge [\, \forall q \in N^i - k([d^i_{jk} < d^i_{jq}] \vee [\, d^i_{jk} = d^i_{jq} \wedge |k| < |q| \,])] \wedge [\, \forall m \in A^i_j \; (\, a^i_{jk} \neq m \vee s^i_{jk} > sn(m) \,)\,] \quad (2)$$

Eq. (1) addresses the case in which router i has at least one neighbor as a next hop to prefix j. Only those neighbors reporting the most recent sequence numbers from the known anchors of prefix j can be considered as next hops, and they are ordered lexicographically based on their distances to prefix j and their names. Router i can select neighbor k from the set of neighbors reporting the most recent sequence numbers for prefix j if either neighbor k is closer to the prefix than router i, or is at the same distance to the prefix but its name is smaller than the name of router i.

Eq. (2) addresses the case in which router i has no next hops to prefix j. A neighbor can be considered as a next hop to the prefix only if it reports a finite distance to the prefix, the smallest distance to the prefix among all neighbors, and either a more recent sequence number from a known anchor of the prefix or a new anchor. This is a logical extension of the traditional case of loop-free routing based on sequence numbers in which the most recent sequence number is used for the single copy of a given destination.

Based on the above, router i first determines which neighbors report valid sequence numbers. From those neighbors reporting valid sequence numbers, router i selects those that can be next hops (i.e., part of S^i_j) using SOC. The minimum distance to a prefix is computed considering only those neighbors satisfying the constraints imposed by SOC, which prevents routing-table loops. If at least one neighbor is found that satisfies SOC and any changes are made to its distance, anchor or sequence number, router i schedules an update. If no neighbor of router i satisfies SOC, then router i is unable to have any next hop to prefix j. Accordingly, it resets $S^i_j = \emptyset$ by setting the empty successor-set flag $es^i_j = 1$ to indicate that no neighbor can be used as a next hop to prefix j; sets $d^i_j = \infty$, $a^i_j = null$, and $sn^i_j = 0$; and notifies its neighbors of its status by sending an update.

Fig. 1 illustrates how DCR routes to the nearest replica of a prefix. The figure shows the routing information used for a single prefix when three routers (d, o, and u) serve as the anchors, and each link has unit cost. The first tuple listed next to each node indicates the shortest distance from the router

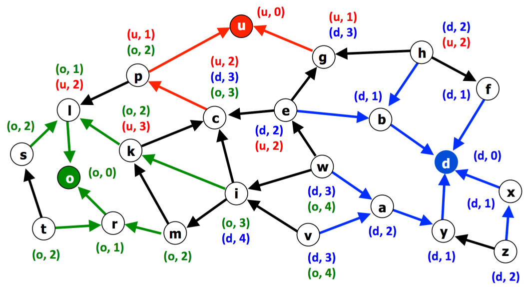

Figure 1. Routing to the nearest instance of a prefix

to the prefix and the anchor with the smallest name at that distance. Updates from each router state only the preferred anchor (e.g., the update from node e states d as the anchor and distance 2 to it). Each additional tuple next to a router, if any, states an alternate anchor for the prefix and the distance to it. It is assumed that all routers have received the most-recent sequence numbers from d, o, or u for the prefix.

As Fig. 1 shows, the updates generated by an anchor propagate only as long as they provide routers with shorter paths to prefixes. In the example, no routing update about the prefix propagates more than four hops, even though the network diameter is eight. In general, independently of how many anchors exist in a network for given prefix, a router only has as many active anchors for the prefix as it has neighbors, because each router reports only the best anchor it knows for each prefix. The arrowheads in the links between nodes indicate the direction in which interest queries could propagate, and the arrow to the lexicographically smallest next hop is shown with the color of the corresponding anchor. It is clear from the figure that DCR does not create a spanning tree for a prefix, but a directed acyclic graph (DAG) with multiple roots, each such root being an anchor.

We observe that, even in this small network of just 23 routers, several routers have multiple paths to prefixes; all links can be used to forward requests for content; very few routers know about all the anchors of the prefix; and traversing any possible directed path in Fig. 1 necessarily terminates at d, o, or u, without traversing a loop.

3.4 Routing to All Copies of Prefixes

DCR supports routing to *all* anchors of the same prefix in two phases. First, all the anchors of a given prefix are connected with one another through an *anchor based tree* (ABT) for the prefix. The ABT of a prefix is rooted at the anchor of the prefix with the smallest name, which we call the *root anchor* of the prefix. The ABT is established using routing updates exchanged by routers located between the root anchor and other anchors. Then, to send data packets to all the anchors of a prefix, a router that is not part of the ABT simply sends the packets towards the nearest anchor of the prefix; the first router in the ABT that receives the data packets broadcasts them over the ABT of the prefix.

Specifying when a prefix requires routing to all its replicas, and hence the need for an ABT, can be done in a number of ways. One approach is for the name of the prefix to denote the need for an ABT; this is the equivalent of a multicast address for the case of traditional routing. An alternative approach is for a special signaling message to request the creation of an ABT for a given prefix.

Establishing an ABT is a problem similar to establishing a shared multicast tree. In the case of core based trees (CBT) [2], all routers in a network are required to have routing information for the core of a multicast group, and routers send join requests towards the core. A join request that reaches the first router in the shared multicast tree causes the routers between that router and the requesting router to join the multicast tree.

The signaling in DCR is more efficient than the signaling used in CBT or similar approaches to multicast routing. To build the ABT for prefix j, routers select the root anchor of the prefix to be that anchor of the prefix that has the smallest name; therefore, at each router i, $ra^i_j = Min\{a^i_{jk}, ra^i_{jk} \mid k \in N^i\}$. Router i then uses the following condition to select the next hop to ra^i_j.

Root-Anchor Ordering Condition (ROC):
Router i can select neighbor $k \in N^i$ as its next hop to its root anchor for prefix j if the following statements are true:

$$(ra^i_{jk} = ra^i_j) \wedge [\, \forall m \in A^i_j \; (\, a^i_{jk} \neq m \vee s^i_{jk} \geq sn(m)) \,] \tag{3}$$

$$\forall m \in N^i_j \; (\, ra^i_{jk} \neq ra^i_{jm} \vee \; rd^i_{jk} \leq rd^i_{jm} \,) \tag{4}$$

$$(rsn^i_j < rsn^i_{jk}) \vee [(\, rsn^i_j = rsn^i_{jk}) \wedge (\, rd^i_{jk} < rd^i_j)] \tag{5}$$

Eq. (3) states that neighbor k must report the same root anchor, which is the anchor with the smallest name known to router i, and must report an up-to-date sequence number from such an anchor. Eq. (4) states that neighbor k must offer the shortest distance to the root anchor. Eq. (5) orders router i with its selected next hop to the root anchor based on the distance to the anchor and the sequence number created by the anchor. ROC uses the same lexicographic ordering based on sequence numbers proposed in several prior routing protocols based on sequence numbers (e.g., [19]), given that a root anchor is the intended destination and cannot be replicated in the network.

The ABT is established in a distributed manner using the distance updates exchanged among routers. A router that knows about multiple anchors for a prefix other than the anchor it considers to be the root anchor sends distance updates about the root anchor along the preferred path to each of the other anchors it knows. Routers that receive updates about the root anchor send their own updates to their preferred next hops to each other anchor they know. This way, distance updates about the root anchor propagate to all other anchors of the same prefix. However, updates about the root anchor propagate only to those routers in preferred paths between the root anchor other anchors.

The distance from router i to the root anchor ra^i_j is $rd^i_j = rd^i_{js} + l^i_s$, where s is the next hop to a^i_j selected by router i. Each time router i changes its routing information for the root anchor of prefix j, it schedules an update about its root anchor to each neighbor that satisfies the following condition.

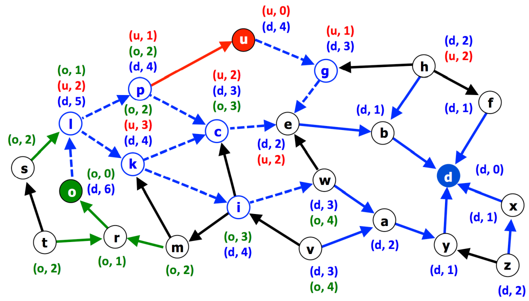

Figure 2. Maintaining routes to a root anchor

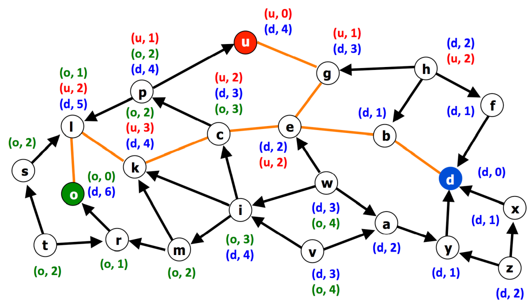

Figure 3. The ABT of a prefix

Root-Anchor Notification Condition (RNC):
Router i sends an update with the tuple $[ra^i_j, rd^i_j, rsn^i_j]$ to each router $q \in N^i$ for which the following two statements are true:

$$a^i_{jq} \neq ra^i_j \tag{6}$$

$$\forall k \in S^i_j \ (a^i_{jq} \neq a^i_{jq} \vee [d^i_{jq} < d^i_{jk} \vee (d^i_{jq} = d^i_{jk} \wedge |q| < |k|)]) \tag{7}$$

Eqs. (6) and (7) state that q is the lexicographically smallest next hop to an anchor known to router i that is not the root anchor.

Each anchor sends a join request towards the root anchor to establish the ABT. When an anchor p of prefix j receives updates from its neighbors stating that the root anchor of j is a router q and it is true that $|q| < |p|$, then anchor p sends a join request to the neighbor with the smallest name among all neighbors reporting the shortest distance to root anchor q. Each router forwarding the joint request stores an entry for the request denoting the neighbor from which it was received and a unique identifier for the request for a finite period of time. The join request traverses the path towards the root anchor q, until it reaches a router x that is already part of the ABT for prefix j, which then sends a join reply to the request that makes each relay router processing the request join the ABT.

Fig. 2 shows how routing information regarding the root anchor of a prefix is propagated. In the example, router d has the smallest name among all the anchors of the prefix. Routers g, c and i know about d and other anchors with larger names than d, and assume that d should be the root anchor for the prefix. Accordingly, each of those routers sends an update about d to the best next hop to each other anchor it knows. In our example, g sends an update to u, c sends an update to p and to k, and i sends an update to k. A router that receives an update about d being the root anchor sends its own update to each best next hop to each other anchor it knows. This way, updates about d reach the other two anchors of the prefix (o and u).

The solid blue lines in Fig. 2 indicate the links over which routing updates propagate for prefix j with anchor d being the nearest anchor. The dashed lines in the figure indicate the links over which updates propagate for anchor d as the root anchor of prefix j, because the link is part of the preferred shortest path to a known anchor. We observe that many routers (i.e., m, o, r, s, t, and u) do not participate in the propagation of updates about d as the root anchor of prefix j, and some (i.e., m, r, s, and t) do not even receive updates about d being the root anchor of prefix j. This contrasts with the traditional approach to building shared multicast trees, in which all routers would have to have a route to d.

Fig. 3 shows the resulting anchor based tree (ABT) for the prefix in the same example. Anchors u and o send their join requests towards d to join the ABT of the prefix. To forward a content requests that must be sent to all anchors of a prefix, routers simply forward the request to one of the anchors of the prefix, and the request is then broadcast over the anchor tree of the prefix as soon as it reaches the first router that has joined the anchor tree. The broadcast mechanism within an ABT is much the same as in shared-tree multicast routing protocols. However, to establish the ABT of a prefix, no router is required to know about all the anchors of the prefix and only routers in the shortest paths between anchors and the root anchor participate in the signaling needed to build the ABT.

4. Name-based Signaling for DCR

DCR can be implemented using different name-based signaling approached, depending on the ICN architecture in which it is being used. The main decisions to be made are: (a) how to name routers, (b) the syntax of update messages, and (c) how to exchange such messages between neighboring routers. The naming of routers is perhaps best done using a hierarchical name space, as has been proposed in NLSR [18]. With this naming scheme, the name of a router would be "/¡network¿/¡site¿/¡router¿", and the name of the DCR daemon running in it would be "/¡network¿/¡site¿/¡router¿/DCR."

As discussed in Section 3, the general semantics of the update process in DCR consists of a router sending incremental routing-table updates to its neighbors. An update message is identified by the name of the router, the protocol (DCR), the type of message, and a sequence number incremented by the sending router. The update messages can be sent periodically and serve as an indication that the router is operational. The syntax of update messages and exactly how messages are exchanged between routers depend on the basic signaling defined in the ICN architecture in which DCR operates. The signaling among routers can be receiver-initiated or sender-initiated.

In NDN, signaling is receiver-initiated, in that data always follow an Interest stated by the receiver. The DCR process in a router using NDN must periodically send a "Routing Interest" (RI) to elicit routing updates from its neighbors. In contrast to NLSR, the state of neighbor routers is different from one another and hence DCR signaling cannot be based on CCNx Sync as it is done in NLSR [18]. To associate one interest with one data object, a router must send an RI to each neighbor requesting routing information. The RI can simply state a name such as "$/ < network > / < site > / < router > /DCR/update/seq\_no$." A router receiving the RI responds with a content object corresponding to a "Routing Update" (RU) with updates made to its routing table that have not been reported to the neighbor sending the RI. A router stores the last sequence number sent by each neighbor in its RI, and associates with that number the list of pending routing updates needed by that neighbor. After receiving an update in response to its RI from a neighbor, a router increments the sequence number stated in the next RI sent to the same neighbor.

A more efficient approach for name-based signaling in DCR is for each router to send a RU to all its neighbors without having to be asked explicitly by any one neighbor. This "push" mechanism is part of new versions of CCNx [8]. The RIs or RUs sent from a router inform its neighbors that the router is alive.

A DCR update in CCN or NDN is digitally signed by the anchor that originates the update, and the update states information that can be used in signature verification [15]. Hence, a router in DCR can attest that it is an anchor of a prefix in much the same way that a router in NLSR attests that the LSA regarding the prefix of local content is valid. Security mechanisms similar to those proposed in NLSR [18] can be implemented for DCR. The approach is aimed at intra-domain routing in which a network administrator (trust anchor) certifies the authenticity of keys; and a hierarchy of keys owned by the network administrator as root, site administrator, operator, router and the process [18]. Various mechanisms can then be used to verify on an end-to-end basis that neighbor routers report valid distances to anchors of prefixes (e.g., [10]). However, the design, verification, and performance of efficient security mechanisms for DCR is outside the scope of this paper.

5. Interactions with Data Plane

The mechanisms used in the data plane of an ICN for the forwarding of content requests or content are independent of the mechanisms used in the control plane to establish routes by DCR. What is important about DCR for the data plane is that it provides multiple loop-free paths to the nearest replicas of content, or to all the sites advertising a name prefix or NDO. Because the routes provided by DCR are loop-free, consumers of content can issue content requests simply stating the name of the content they require and an identifier for the request. Such requests can be forwarded using any strategy to either the nearest site or to all sites advertising the prefix that best matches the name of the requested content. The data plane does not have to rely on topology information (e.g., source routes) or complex search mechanisms aimed at finding valid paths to replicated content.

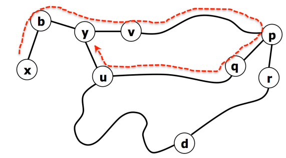

Figure 4. Request propagation over a loop

For example, in CCNx [8, 31] or NDN [22], each router maintains a content store, a forwarding table (FIB), a pending interest table (PIT), and a content store. DCR would populate the FIB with one or multiple routes to name prefixes, i.e., the faces that can be used to forward Interests (without looping). Additional information can be provided by data plane mechanisms to determine how such loop-free routes are used. The PIT has an entry for each Interest that has been forwarded and waiting for data to return; for a given Interest identified by the name of the prefix and a nonce created by the origin of the request, the PIT states the faces (or interfaces in other ICN architectures) over which the Interest has been received and the faces over which the interest has been forwarded, and nothing else.

CCN or NDN routers can forward an Interest in different ways. For example, an Interest can be sent towards the nearest anchor of the prefix that constitutes the best match with the name stated in the Interest. Alternatively, an Interest can be sent to all the anchors of the prefix that is the best match for the name stated in the Interest. Given that routers cache content in their content stores, content can be sent back to the requesting node by the first router that has a copy of the object along the path to an anchor.

6. DCR Correctness

Many ICN architectures (e.g., [15, 9, 11, 22, 27]) rely on the control plane to provide multiple routes to the same name prefix. Content requests used in the data planes of these architectures are such that routers can detect when a given request traverses a loop, and packets carrying content traverse the reverse paths taken by content requests. Because of this, it has been assumed [15, 18, 30] that loop freedom in the control plane is not needed for content requests to be forwarded correctly in the data plane.

We illustrate by example why multi-path loop-free routing is needed for the data plane to operate correctly if a content request is identified simply by the name of the content being requested, or the name and an nonce; and forwarding of content requests in the data plane is done on a hop-by-hop basis using only that information. Consider the network shown in Fig. 4, where CCN or NDN is used and prefix j is announced only by router d. The control plane in this example uses a routing protocol that provides multiple routes to prefixes but *does not* populate the FIBs ensuring loop-free routes. No topology changes occur after routers have computed their shortest paths to advertised prefixes, which makes all the routes to a prefix permanent. The data plane is such that a router does not forward any requests back to any neighbors (faces) from which the requests were received; and uses the name or the name and a nonce in a request to forward the same request only once.

Let router x issue a request for prefix j that is forwarded in the data plane as shown in Fig. 4. Router y receives the same request from neighbors b and u and has sent the request once to neighbor v. According to the information in the PIT of router y and its forwarding strategy, router y simply drops the request. Furthermore, the control and data planes operate correctly in this example, and hence there is no guarantee that future retransmissions of content requests by x for the same content but with different nonces, for example, will reach d.

It is of course straightforward to implement data-plane mechanisms that take into account the fact that a control plane does not provide loop-free routes to content. One example consists of forwarding requests over all the faces leading to a name prefix. Other approaches entail trying different faces after timeouts, or including some information about the path traversed by the interest. However, such approaches amount to data-plane search mechanisms that incur unnecessary communication overhead or delays if the control plane is designed to enforce loop-freedom efficiently.

To ensure that content requests are delivered within a finite time using simple hop-by-hop forwarding mechanisms in the data plane based on names or names and nonces, DCR guarantees that routing-table loops are never formed, even as the locations of content and the network topology change. Furthermore, routers running DCR attain the shortest distances to the nearest instances of each known prefix. The following theorems prove that this is the case.

Theorem 6.1. *No routing-table loops can be formed if routers use SOC to select their next hops to prefixes.* □

Proof. The proof is by contradiction. Assume that a routing loop L_j for prefix j consisting of h hops is created at time t_L when the routers in the vertex set of L_j change successors according to SOC. Let $L_j = (n_1, n_2, ..., n_h)$, with $n_{i+1} = s_j^{n_i}$ for $1 \le i \le h-1$ and $n_1 = s_j^{n_h}$.

According to SOC, each hop $n_i \in L_j$ $(1 \le i \le h)$ can select its next hops (i.e., $S_j^{n_i}$) in only two ways, depending on whether or not $od_j^{n_i} < \infty$ when $n_i \in L_j$ selects its next hops before or at time t_L when L_j is formed.

Assume that there is a subset of hops $I_j \subset L_j$ such that $od_j^{n_m} = \infty$ when n_m joins L_j by adding n_{m+1} to $S_j^{n_m}$ for each $n_m \in I_j$. By assumption, router n_m must then use Eq. (2) in SOC. Therefore, $d_{jn_{m+1}}^{n_m} < \infty$ and router n_{m+1} must report to n_m either an anchor that router n_m did not know before, or a more recent sequence number created by an anchor known to n_m before the update from n_{m+1}. Furthermore, for all $q \in N^{n_m}$, it must be true that $d_{jq}^{n_m} > d_{jn_{m+1}}^{n_m}$ or $d_{jq}^{n_m} = d_{jn_{m+1}}^{n_m}$ and $|n_m| < |q|$. Therefore, the following relation must hold between $d_{jn_{m+1}}^{n_m}$ and $d_{jn_{m-1}}^{n_m}$ for any $n_m \in I_j$:

$$\begin{aligned} & d_{jn_{m-1}}^{n_m} > d_{jn_{m+1}}^{n_m} \\ \vee\ (\ & d_{jn_{m-1}}^{n_m} = d_{jn_{m+1}}^{n_m} \wedge |n_{m-1}| > |n_{m+1}|\ ) \end{aligned} \tag{8}$$

Consider a subset of hops $\{n_m, n_{m+1}, ..., n_{m+c}\} \in I_j$ that forms a contiguous chain in L_j, where $c \le h$. It follows from Eq. (8) that

$$\begin{aligned} & (\ d_{jn_{m-1}}^{n_m} = d_{jn_{m+c+1}}^{n_{m+c}} \wedge |n_{m-1}| > |n_{m+c}|\ ) \\ & \vee (\ d_{jn_{m-1}}^{n_m} > d_{jn_{m+c+1}}^{n_{m+c}}\ ) \text{ for } h \ge c \ge 0. \end{aligned} \tag{9}$$

On the other hand, by assumption, every hop in $L_j - I_j$ must use Eq. (1) in SOC to select its next hops and hence join L_j. Therefore, the following two equations must be satisfied for any $n_i \in L_j - I_j$:

$$d_{jn_i}^{n_i - 1} \ge od_j^{n_i} \tag{10}$$

$$\begin{aligned} & od_j^{n_i} > d_{jn_{i+1}}^{n_i} \ge od_j^{n_{i+1}} \\ \vee\ & (od_j^{n_i} = d_{jn_{i+1}}^{n_i} \ge od_j^{n_{i+1}} \wedge |n_i| > |n_{i+1}|). \end{aligned} \tag{11}$$

Consider a subset of hops $\{n_l, n_{l+1}, ..., n_{l+k}\} \in L_j - I_j$ that forms a contiguous chain in L_j, where $k \le h$. It must be the case that either $od_j^{n_{l+i}} > d_{jn_{l+i+1}}^{n_{l+i}} \ge od_j^{n_{l+i}}$ for at least one hop n_{l+i} in the chain, or that $od_j^{n_{l+i}} = d_{jn_{l+i+1}}^{n_{l+i}} \ge od_j^{n_{l+i}}$ and $|n_{l+i}| > |n_{l+i+1}|$ for each hop n_{l+i} in the chain. Accordingly, it follows from Eqs. (10) and (11) that

$$\begin{aligned} & (\ d_{jn_{l+1}}^{n_l} = d_{jn_{l+k+1}}^{n_{l+k}} \wedge |n_l| > |n_{l+k}|\ ) \\ & \vee (\ d_{jn_{l+1}}^{n_l} > d_{jn_{l+k+1}}^{n_{l+k}}\ ) \text{ for } h \ge k \ge 0. \end{aligned} \tag{12}$$

It follows from Eqs. (9) and (12) that using SOC enforces the same lexicography ordering among the hops of L_j for any given combination of chains of nodes in L_j that belong to I_j or $L_j - I_j$, i.e., use Eq. (2) or Eq. (1) in SOC to select their next hops when they join L_j.

From Eqs. (9) and (12), it must be true that, if at least one hop in $n_i \in L_j$ is such that $d_{jn_{i+1}}^{n_i} > d_{jn_{k+1}}^{n_k}$, where $n_k \in L_j$ and $k > i$, then $d_{jn_{m+1}}^{n_m} > d_{jn_{m+1}}^{n_m}$ for any given $m \in \{1, 2, ..., h\}$, which is a contradiction. On the other hand, if $d_{jn_{i+1}}^{n_i} = d_{jn_{k+1}}^{n_k}$ for any n_i and n_k in L_j, then $|n_m| > |n_m|$ for any given $m \in \{1, 2, ..., h\}$, which is also a contradiction. Therefore, L_j cannot be formed when routers use SOC to select their next hops to prefix j. □

Assume that DCR is executed in a connected finite network G, that a router is able to detect within a finite time who its neighbor routers are, and that any signaling message sent over a working link between two routers is delivered correctly within a finite time. Further assume that topological changes and name prefix changes stop taking place after a given time t_T. The following theorem proves that DCR attains shortest paths to the nearest replicas of known prefixes within a finite time. To simplify the inductive proof, we assume that the cost of any operational link is 1; however, the same approach applies to the case of positive link costs [4].

Theorem 6.2. *If DCR is used in network G, the routes to prefixes converge to the shortest distances to the nearest anchors of the prefixes within a finite time after t_T.* □

Proof. Without loss of generality, we focus on a specific prefix j. The proof is by simple induction on the number of hops (k) that routers are away from the nearest anchors of prefix j. Let the set of anchors in the network for prefix j be $A = \{\alpha_1, \alpha_2, ..., \alpha_r\}$, where r is smaller than or equal to the number of routers in the network.

Basis case: For $k = 1$, consider an arbitrary neighbor of a given anchor α_i of prefix j, with $1 \leq i \leq r$. Given that the signaling between neighbors is reliable and no links fail after time t_T, router n_1 must receive an update $U_j^{\alpha_i}$ from α_i stating $d_j^{\alpha_i} = 0$, $a_j^{\alpha_i} = \alpha_i$, and $sn_j^{\alpha_i} = s(\alpha_i)$ (the most recent sequence number created by α_i) within a finite time after t_T; and it must update $d_{j\alpha_i}^{n_1} = 0$, $a_{j\alpha_i}^{n_1} = \alpha_i$, and $sn_{j\alpha_i}^{n_1} = s(\alpha_i)$.

Because $d_{j\alpha_i}^{n_1} = 0$ and $sn_{j\alpha_i}^{n_1} = s(\alpha_i)$ always satisfy SOC at router n_1 for prefix j, it must be the case that $\alpha_i \in S_j^{n_1}$. Furthermore, any other next hop in $S_j^{n_1}$ must also be an anchor, because the smallest link cost between neighbors equals 1 and hence the smallest value of $d_j^{n_1}$ equals 1. Router n_1 must set $d_j^{n_1} = 1$ and send and update stating that distance, together with the name of that anchor and the sequence number it created, after a finite time $t_1 > t_T$. Therefore, the theorem is true for the basis case.

Inductive step: Assume that the theorem is true for an arbitrary router n_k that is k hops away from its nearest anchors of prefix j. It must then be true that $d_j^{n_k} = k$ after a finite time $t_k > t_1$. By assumption, the signaling between neighbors is reliable and no links fail after time $t_T < t_k$; therefore, each neighbor of n_k must receive updates from n_k a finite time after t_k stating $d_j^{n_k}$, $a_j^{n_k}$, and $sn_j^{n_k}$. Each neighbor $p \in N^{n_k}$ must update $d_{jn_k}^p = k$, $a_{jn_k}^p = a_j^{n_k}$, and $sn_{jn_k}^p = sn_j^{n_k}$ a finite time after t_k.

Let router $q \in N^{n_k}$ be such that it is farther than k hops away from any anchor of prefix j. Because $d_j^{n_k} = k$ is the shortest distance from n_k to prefix j after time t_k, router q cannot have any neighbor reporting a distance to j smaller than k after time t_k. Therefore, $d_{jn_k}^q$ must satisfy Eqs. (1) or (2) of SOC within a finite time after t_k and router q must make n_k a next hop to prefix j a finite time after t_k. Furthermore, any neighbor of q in S_j^q must have also reported a distance of k hops to j. Router q selects the anchor in S_j^q with the smallest name, and sends an update within a finite time $t_{k+1} > t_k$ stating $d_j^q = k+1$, together with the name of its chosen nearest anchor and the most recent sequence number created by that anchor. Therefore, router q and hence any router $k+1$ hops away from the nearest anchors of prefix j must attain a shortest distance of $k+1$ hops to prefix j within a finite time and the theorem is true. □

The following theorems show that DCR is correct for routing to all the sites announcing a prefix. Theorem 6.3 shows that routes to root anchors are loop-free. Using the same assumptions introduced for the proof of Theorem 6.2, Theorems 6.4 and 6.5 show that DCR builds an ABT for a prefix that requires routing to all routers advertising the prefix.

Theorem 6.3. *No routing-table loops can be formed if ROC is used to select next hops to the root anchors of prefixes.* □

Proof. The proof is by contradiction. Assume that a routing loop L_r for root anchor r consisting of h hops is created when nodes in L_r change successors according to ROC. Let $L_r = (n_1, n_2, ..., n_h)$, with $n_{i+1} = s_r^{n_i}$ for $1 \leq i \leq h-1$ and $n_1 = s_r^{n_h}$.

According to ROC, it must be true that $rsn_r^{n_i} \leq rsn_r^{n_{i+1}}$ for $1 \leq i \leq h-1$ and $rsn_r^{n_h} \leq rsn_r^{n_1}$ for each hop $n_i \in L_r$ ($1 \leq i \leq h$). This result implies that $rsn_r^{n_i} = rsn_r^{n_{i+1}}$ for $1 \leq i \leq h-1$ and $rsn_r^{n_h} = rsn_r^{n_1}$. Because of ROC, the above result implies that $rd_r^{n_i} < rd_r^{n_{i+1}}$ for $1 \leq i \leq h-1$ and $rd_r^{n_h} < rd_r^{n_1}$, which implies that $rd_r^{n_i} < rd_r^{n_i}$ for $1 \leq i \leq h$. This is a contradiction and hence the theorem is true. □

Theorem 6.4. *If DCR is used in network G, each anchor of prefix j attains a shortest distance to the root anchor of prefix j within a finite time after t_T.* □

Proof. It follows from Theorem 6.2 that all routers must have a route to the nearest copy of prefix j (i.e., an anchor) after a finite time $t_R \geq t_T$. Let r be the root anchor of prefix j, then $|r| < |u|$ for any router u that is an anchor of prefix j. Let A be the set of anchors of prefix j other than r, and let $a \in A$ be such that the shortest physical paths between a and r in G do not include any other anchor. The number of hops of such paths must be odd or even.

Consider a physical path P_{ar} between a and r, and assume that $|P_{ar}| = 2h$. Because DCR converges to correct routing information for the nearest anchors, there is a router $m \in P_{ar}$ with $d_j^m = h$ and $a_j^m = r$ and two neighbors in P_{ar}. Let $n_1 \in N^m \cap P_{ar}$ be s_j^m. Then neighbor $n_2 \in N^m \cap P_{ar}$ must have reported $d_j^{n_2} = h-1$ and $a_j^{n_2} = a$. According to RNC, router m must send a distance update about root anchor r to neighbor n_2. Furthermore, according to RNC, router n_2 and every router router in the subpath from n_2 to a in P_{ar} must send an update with its own distance to r to the next hop in P_{ar} towards a. Hence, a must have a shortest distance to r in a finite time.

Assume that $|P_{ar}| = 2h+1$. Then there must be a router $n_1 \in P_{ar}$ with $d_j^{n_1} = h$ and $a_j^{n_1} = r$ and a router $n_2 \in N^{n_1} \cap P_{ar}$ with $d_j^{n_2} = h$ and $a_j^{n_2} = a$. According to RNC, router n_1 must inform n_2 of its distance to r, and every router between n_2 and a in P_{ar} must send an update with its own distance to r to the next hop in P_{ar} towards a. Therefore, anchor a must have a shortest distance to r in a finite time.

Consider now an anchor $b \in A$ such that a physical shortest path P_{br} between b and r includes a sequence of $k \geq 1$ other anchors (a_1, a_2, ..., a_k), with a_1 being the closest to r. It follows from the previous argument that a_1 must attain a shortest route to r. By induction on k, it also follows that a_k must also attain a shortest distance to r in a finite time. Accordingly, it follows from the previous argument for anchor

a that anchor b must have a shortest distance to r in a finite time. Therefore, the theorem is true. □

Theorem 6.5. *If DCR is used in network G, an ABT of prefix j is established that includes all anchors of prefix j within a finite time.* □

Proof. If follows from Theorem 6.4 that every anchor other than r must have at least one next hop to r. To join the ABT, each anchor $a \neq r$ sends a join request along its lexicographically smallest next hop to r. Because routes to r are loop free (Theorem 6.3), the join request must reach either r or a router in the ABT, which in turn sends a reply that makes the routers in the path to a join the ABT. Therefore, the theorem is true. □

7. Performance Comparison

7.1 Complexity of The Control Plane

To compare the performance of DCR with other content routing approaches, we focus on the communication and time complexities of the approaches in the control plane, and make a number of simplifying assumptions assuming a network without hierarchical routing. We assume that a separate control message is sent for any given LSA or distance update. In practice, multiple LSAs and distance updates can be aggregated to conserve bandwidth. In fact, aggregating distance updates for multiple prefixes is easier than aggregating LSAs from multiple sources. However, given that the maximum size of a control message is a constant value independent of the growth of N or C, this aggregation does not change the order size of the overhead incurred by the routing protocols.

The number of routers in the network is denoted by N routers and E denotes the number of network links. The number of different name prefixes available in the network is denoted by C, the average number of replicas for a given name prefix is r, the average number of neighbors per router is l, and the network diameter is d. Assuming that all transmissions over any given link are successful, the communication complexity of a routing protocol is the number of messages that must be transmitted for each router to have correct routing information about all the C prefixes. The time complexity of a routing protocol is the maximum time needed for all routers to have correct routing information for all prefixes.

In the link-state routing (LSR) approach used in NLSR and OSPFN for name-based routing, each LSA originated by a router regarding a link or a prefix must be sent to all the other routers in the network, and a router must transmit an LSA for each adjacent link and each prefix that is stored locally. Hence, given that a router can be a maximum of d hops from the source of a given LSA, the time complexity (TC_{LS}) and communication complexity (CC_{LS}) of LSR are:

$$TC_{LSR} = O(d); \quad CC_{LSR} = O(ErC + lEN) \tag{13}$$

If a DHT approach is used with all routers participating in the DHT, the approach that incurs the least amount of overhead is a virtual DHT with one-hop routing [12], such that routers run the DHT locally and maintain routes to all routers in the network. The communication complexity associated with publishing a prefix in the DHT and associating r sites with the prefix (to support routing to any or all copies of the prefix) is $O(rdC)$ assuming no loops. The communication complexity of maintaining routes to all routers is $O(lNE)$, given that link-state routing is typically used. Hence, assuming the fastest possible propagation of routes to all routers, the time complexity (TC_{DHT}) and communication complexity (CC_{DHT}) of the DHT approach equal:

$$TC_{DHT} = O(d); \quad CC_{DHT} = O(drC + lEN) \tag{14}$$

The traditional distance-vector routing (DVR) approach subject to looping problems (e.g., RIP) has far worse performance the the above schemes. Because signaling can traverse long paths and long-term loops and "counting to infinity" can occur, the basic distance-vector approach is known to have $O(N)$ time complexity and $O(N^2)$ communication complexity [16]. Furthermore, given that distance information to all replicas is needed in DVR, we have:

$$TC_{DVR} = O(N); \quad CC_{DVR} = O(N^2 rC) \tag{15}$$

This is much worse than the other approaches and explains why name-based routing using distance vectors has not been considered in the past.

Independently of the number of anchors for a given prefix, the information a router communicates for a given prefix in DCR is only its distance to the nearest anchor of the prefix, plus the anchor name and the latest sequence number created by that anchor. As the number of replicas increases, the distances from a router to the nearest replica of a prefix decreases, and it is always the case that the number of hops from any router to the nearest replica of a prefix (x) is at most d hops. Furthermore, DCR does not incur any routing-table loops. This means that: (a) any routing information propagates as fast as the shortest path between its origin and the recipient; and (b) the number of messages required for all routers to have a correct distance to a given prefix is $O(E)$, regardless of the number of sites r replicating the prefix. Given that there are C prefixes in the network, the time complexity (TC_{DCR}) and communication complexity (CC_{DCR}) of DCR are:

$$TC_{DCR} = O(x); \quad CC_{DCR} = O(EC) \tag{16}$$

From Eqs. (13) to (16), it is clear that the time complexity in DCR is smaller than the time complexity of the other approaches, i.e., routers attain correct routing tables faster than using link states or a DHT. Just as important, as the network size and number of replicas increase, CC_{LSR}, CC_{DHT}, and CC_{DVR} can be orders of magnitude larger than CC_{DCR}. The key reason why DCR is more efficient than the other two approaches is that DCR eliminates the need to communicate information about the network or content replicas.

Table 1. Control-plane overhead

Protocol	10%	20%	40%
DCR	100	100	100
LSR	699	1442	2269

7.2 Simulation Experiment

Due to space limitations we only present the results of a simple simulation experiment to contrast the signaling overheads incurred by DCR and the link-state routing (LSR) approach exemplified by NLSR and OSPFN. We used QualNet [28] (version 5.0) as the discrete event simulator. The implementation of DCR was based on modifications of the Distributed Bellman Ford implementation in Qualnet to use sequence numbers in updates and SOC in the selection of successors to prefixes. The implementation of the LSR approach was based on the OSPF implementation in Qualnet by adding LSAs to advertise content the way NLSR and OSPFN do; however, we use "push" signaling based on RUs sent without interests that request them. The experiment focused on the average number of control packets generated per node by the routing protocols in order for routers to have correct routing information for the nearest replicas of all NDOs. All protocols use the same time period to refresh their routing structures. Each simulation ran for 10 different seed values. We used static networks of 100 nodes, with nodes being uniformly distributed in the network to avoid disconnected nodes.

The transmission of an update to all neighbors of a node is counted as a single transmission, and an update message in DCR and LSR carries all the updated distances or link states, respectively. We increase the number of nodes requesting NDOs from 10% to 40% of the 100 nodes of the network. Each node originally publishes 2 different NDOs, and a node that caches an NDO it requested advertises the NDO. Table I shows the results of the simulation experiment. It is clear that, as the rate of content requests increases and NDOs are replicated more and more in the network, the average signaling incurred per node to update all nodes on the locations of new copies of NDOs in LSR becomes excessive. On the other hand, the signaling overhead of DCR remains constant, because all updates fit in a single message and routers advertise only their distances to the nearest instances of NDOs. Accordingly, regardless of how many times an NDO is replicated in the network, each router advertises the same number of distances.

8. Conclusion

The *Distance-based Content Routing* (DCR) protocol was introduced, which is the first approach for name-based content routing in ICNs based solely on distances to NDOs or name prefixes. DCR does not require any routing information about the physical topology of the network or information about all the replicas of the same content to provide multiple shortest paths to the nearest replica of content, as well as to all sites advertising an NDO or name prefix.

DCR was shown to be loop-free at every instant, to converge to the shortest paths to the closets replicas of content, and to build the anchor based tree of a prefix when routing to all copies of the prefix is needed. DCR was shown to have smaller time and communication complexity than name-based routing based on link states, DHTs, or traditional distance vectors. A simulation experiment was used to illustrate the substantial savings in signaling overhead derived from having to communicate updates about the nearest copies of content, rather than all copies. A more detailed characterization of the performance of DCR and other name-based routing approaches is needed to address the interaction of the control and data planes, the end-to-end delays incurred in obtaining NDOs, and the impact of network size, number of NDOs, and naming hierarchy on the performance of the protocols. This is the subject of a forthcoming publication.

References

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

[2] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT)," *Proc. ACM SIGCOMM 93*, Oct. 1993.

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

[4] D. Bertsekas and R. Gallager, *Data Networks*, Second Edition, Prentice-Hall, 1992.

[5] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Content-Based Addressing and Routing: A General Model and its Application," Technical Report CU-CS-902-00, Department of Computer Science, University of Colorado, January, 2000.

[6] A.Carzaniga et al., "A Routing Scheme for Content-Based Networking," *Proc. IEEE Infocom '04*, 2004.

[7] A. Carzaniga et al., "Is Information-Centric Multi-Tree Routing Feasible?," *Proc. ACM ICN '13*, Aug. 2013.

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

[9] Content Mediator Architecture for Content-aware Networks (COMET) Project [online]. http://www.comet-project.org/

[10] S. Dabideen et al, "An End-To-End Approach to Secure Routing in MANETs," Security and Communication Networks, Wiley, Vol. 3, Issue 2-3, pp. 130–149, 2010.

[11] A. Detti et al., "CONET: A Content-Centric Inter-networking Architecture," *Proc. ACM ICN '12*, 2012.

[12] A. Gupta, B. Liskov and R. Rodrigues, "Efficient Routing for Perr-to-Peer Overlays," *Proc. ACM NDSI '04*, 2004.

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

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

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

[16] M.J. Johnson, "Updating Routing Tables after Resource Failure in a Distributed Computer Network," *Networks*, Vol. 14, No. 3, 1984.

[17] T. Koponen et al., "A Data Oriented (and Beyond) Network Architecture," *Proc. ACM SIGCOMM 07*, 2007.

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

[19] M. Marina and S. Das, "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks," *Proc. IEEE ICNP 2001*, Nov. 2001.

[20] M. Meisel et al., "Listen First Broadcast Later: Topology-Agnostic Forwarding Under High Dynamics," Technical Report 10021, UCLA, 2010.

[21] Mobility First project [online]. http://mobilityfirst.winlab.rutgers.edu/

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

[23] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM 94*, 1994.

[24] C. Perkins and E.M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proc. 2nd IEEE WMCSA*, pp. 90-100, Feb.1999.

[25] Publish Subscribe Internet Technology (PURSUIT) Project [online]. http://www.fp7-pursuit.eu/PursuitWeb/

[26] J. Rajahalme et al., "On Name-Based Inter-Domain Routing," *Computer Networks*, pp. 975-985, 2011.

[27] Scalable and Adaptive Internet Solutions (SAIL) Project [online]. http://www.sail-project.eu/

[28] Scalable Network Technologies, *Qualnet*, http://web.scalable-networks.com/content/qualnet.

[29] I. Solis and J.J. Garcia-Luna-Aceves, "Robust Content Dissemination in Disrupted Environments," *Proc. CHANTS 08: ACM MobiCom 2008 Workshop on Challenged Networks*, Sept. 2008.

[30] L. Wang et al., "OSPFN: An OSPF Based Routing Protocol for Named Data Networking," Technical Report NDN-0003, 2012.

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

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