Title

CORD: Content Oriented Routing with Directories

Permalink

https://escholarship.org/uc/item/8g21p4hr

Authors

Garcia-Luna-Aceves, J.J.
Li, Q.
Karadeniz, T.

Publication Date

2015-02-16

Peer reviewed

CORD:
Content Oriented Routing with Directories

J.J. Garcia-Luna-Aceves,[†][*] Qian Li,[*] and Turhan Karadeniz[*]
[*]Computer Engineering Department
University of California, Santa Cruz, CA 95064
[†]PARC, Palo Alto, CA 94304
Email: { jj, liqian, turhan}@soe.ucsc.edu

Abstract—**Recent proposals for content routing in information-centric networks (ICN) require the use of content-based routing tables listing routes to name prefixes or individual named data objects (NDO), and a single naming space for NDOs. We present CORD (Content Oriented Routing with Directories) as an alternative to content routing in ICNs. CORD eliminates the need for large content-based routing tables by establishing routes to directories using distance-vector signaling and by mapping name prefixes or names of NDOs to directories using publish-subscribe mechanisms. Simulation experiments using the topology of a real ISP network are used to compare CORD with name-based content routing approaches based on link-state and distance information. The results show that CORD attains comparable data delivery and end-to-end delays, but incurs orders of magnitude less control overhead. In addition, CORD supports multipath forwarding of content requests and content.**

I. INTRODUCTION

Several information centric network (ICN) architectures have been proposed [1], [2], [9] as alternatives to the current Internet architecture to address the fact that Internet usage is dominated by peer-to-peer communication and user-generated content. All ICN architectures aim at accessing content and services by name, independently of their location, in order to improve system performance and end-user experience.

Section II summarizes the prior work on routing schemes for ICN architectures. Most ICN architectures rely on name-based routing of content, which integrates name resolution and content routing. Routers advertise or compute routes to named data objects (NDO) or name prefixes, and content requests for specific NDOs are forwarded towards the nearest routers storing those NDOs. As our review of prior work in Section II indicates, prior content routing approaches assume that the entire ICN use the same naming space for NDOs and that routing tables list routes to NDOs or name prefixes, which incurs more overhead than routing to address ranges.

Section III presents CORD (*Content Oriented Routing with Directories*), which is an approach to content routing within autonomous systems in which directory nodes act as intermediaries to establish *virtual cords* linking consumers of content with content producers or caching sites. The primary objective of using directories between content producers or caches and the consumers of content is to reduce control overhead in the ICN. Instead of having routing tables listing routes to individual NDOs or name prefixes, they only list routes to the directories that maintain the mappings between name prefixes or NDO names and the locations where their copies reside. This type of indirection in routing is inspired by McQuillan's work [13] on message addressing capabilities in the early days of the ARPANET.

CORD consists of three main elements: (a) maintaining multiple loop-free routes to directories that maintain the mappings from NDOs and name prefixes to the addresses of sites storing the content; (b) maintaining loop-free routes from directories to destination nodes nearby; and (c) publish-subscribe mechanisms for publishers and consumers of content to advertise and request content.

Section IV describes the results of simulation experiments used to compare the performance of CORD to the link-state approach advocated in NLSR [12] and OSPFN [19], and a loop-free routing approach based on distance vectors in which all the replicas of each NDO or name prefix are known. The protocols are compared using the AT&T network topology. The impact that network size, traffic load, and opportunistic caching have on performance is examined using packet-delivery ratio, average end-to-end delay, control plane overhead, and data plane overhead as the performance metrics. The results from these experiments show that CORD incurs orders of magnitude less control plane overhead than routing schemes that require all content replicas to be known while attaining the same packet delivery ratios and similar average delays.

II. RELATED WORK

The ICN architectures proposed recently advocate various ways to accomplish name resolution and routing, and all of them use on-path caching of content [2], [9]. Due to space limitations we only mention very few of these to contrast them to CORD.

Several ICN projects advocate using a link-state routing approach for intra-domain content routing, and adding content prefixes to BGP for inter-domain content routing (e.g., [5], [6], [4], [15], [18]). NLSR [12] and OSPFN [19] are two protocols for name-based routing of content within an autonomous-system. 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 prefixes for which they have copies. Gritter and Cheriton [8] proposed the 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. The routing approach in the Mobility First project [14] requires using network addresses or source routing or partial source routing.

In many approaches, the names of data objects are mapped into addresses by means of directory servers or overlays, and address-based routing is used for content delivery (e.g., [7], [17]). Several ICN projects (e.g., [16], [18]) have addressed content routing modalities based on distributed hash tables (DHT) running in overlays over the physical infrastructure and accomplish name-based routing on top of link state routing protocols.

Content routing approaches proposed to date require 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).

III. CORD

A. Basic Operation

CORD assumes that: (a) each router or host is assigned a flat or hierarchical name; (b) each piece of content is a *named data object* (NDO) that can be requested by name; (c) NDOs can be denoted using flat or hierarchical naming, with multiple naming conventions possibly being used in the same ICN; and (d) routers cache content opportunistically.

Fig. 1 illustrates how CORD operates. In this example, nodes *a, k,* and *r* maintain directories, and all nodes maintain routes to such directories. The *anchor* of a name prefix or NDO is a directory responsible for maintaining the mappings between the name prefix or NDO to the locations where copies of the prefix or NDO are stored. Directories advertise to the entire network the name prefixes and intervals of NDOs for which they serve as *anchors*, and hence all routers know which directory to contact regarding any NDO or name prefix.

In contrast to the content routing approaches in prior ICN architectures, hierarchical naming and flat naming can be used in the same ICN running CORD. This is attained by stating the name space in which a prefix name or NDO name is defined as part of the advertisements sent by directories, as well as the publish-subscribe requests exchanged with directories.

All routers maintain multiple loop-free routes to directories using sequence-numbered distances. Router d maintains this information in its *directory table* (DT^d), and uses it to contact a subset of directory nodes within a maximum distance r from itself to publish its presence. To do this, router d sends a publish message to its selected local directories with the mapping (d, $\{l_d^1, ..., l_d^k\}$), where l_d^i $(1 \leq i \leq k)$ is a local directory for router d. Each local directory l_d^i of d and each relay between d and the directory receiving the publish request from d stores a tuple stating d, the next hop to d, and $\{l_d^1, ..., l_d^k\}$. Router i maintains a *local directory list* (LDL^i) stating the information about its local directories. In addition, each router i maintains a *neighbor table* (NT^i) stating routing information communicated by its neighbor routers, and a *content store table* (CST^i) listing all content cached by i. Directories also maintain routing information for those routers that select them as local directories.

Routers exchange control information using HELLO messages sent periodically. A HELLO includes some or all the updates made to the sending node's tables. Each node stores all the information from the HELLOs it receives from its neighbors, and also caches content it receives. Entries in CST^i are populated by the publish-subscribe signaling described subsequently.

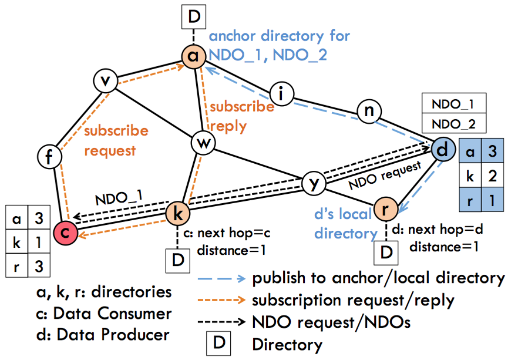


Fig. 1: Example of CORD operation

In contrast to prior approaches, the data plane in CORD is assisted by the publish-subscribe signaling between routers and directories to support content requests without routing tables listing entries for NDOs or name prefixes. The router of the producer or a caching site of an NDO publishes the local copy of the NDO by sending a publish message to its local directories and the anchor of the NDO. The message states the name space used, the NDO name in that name space, the router identifier, and its local directories.

A consumer of content asks for an NDO by sending a content request to its local router. In turn, the router follows a two-step process to request the NDO. To request content, a router first sends a subscription request to one or more of its local directories. If the local directory or directories cannot provide a mapping, the router sends its subscription request to the known anchor of the NDO, as exemplified in Fig. 1. The subscription request specifies the name space used, the NDO name, the identifier of the consumer, and the local directories for the requesting router. A directory with the requested mapping sends the subscription reply to one of the local directories of the requesting router, and the reply is sent to the requesting router from that local directory or a router with a route to the requesting router.

Once a router receives a subscription reply from a directory, it knows the names or identifiers of one or multiple sites hosting the NDO. It can then select the site whose local directory is closest, and can send a content request based on the mechanisms defined for the data plane of the network. The operation of CORD in the control plane is independent of the data plane mechanisms once a router obtains the identifier or name of a site hosting the required content. In the data planes assumed in most prior ICN architectures (e.g., [10], [15]), an NDO request specifies the name of the NDO, does not state the name of the requestor, and is forwarded by routers towards the nearest site known to store the NDO. The NDO is sent back to the consumer by the NDO producer or a caching site over the reverse path traversed by the content request. Hence, CORD supports name-based content routing with routers having to know only how to reach directories, and directories having to know how to reach some routers.

B. Updating Routing Information

CORD uses a distance-vector routing approach to maintain routes to directories. To guarantee loop-free routes, CORD uses sequence numbers that restrict the selection of next hops towards a given directory, such that only those neighbors with shorter distances to the directory or with a more recent sequence number reported by the directory can be considered as successors. Algorithm 1 (DSU) is used to update routing information for directories.

Algorithm 1 DSU: Directory Status Update

```
Input: $DT^i$, $NT^i$;
if $\exists q \in NT^i \mid sn^i_{cq} > sn^i_c$ then
    if $(v = s^i_c) \wedge (sn^i_{cv} > sn^i_c) \wedge (d^i_{cv} = \infty)$ then
        $d^i_c = \infty$; $sn^i_c = sn^i_{cv}$;
    else
        $d^i_c = Min\{d^i_{cf} + 1 \mid (f \in NT^i) \wedge$
               $(sn^i_{cf} = Max\{sn^i_{cv} \mid v \in NT^i)\}\}$;
        $s^i_c = j \mid (j \in NT^i) \wedge (d^i_{cj} = d^i_c - 1)$;
        $sn^i_c = Max\{sn^i_{cv} \mid v \in NT^i\}$;
    end if
else
    $d^i_c = Min\{d^i_{cf} + 1 \mid (f \in NT^i) \wedge (sn^i_{cf} = sn^i_c) \wedge (d^i_{cf} < d^i_c)\}$;
    $s^i_c = j \mid (j \in NT^i) \wedge (d^i_{cj} = d^i_c - 1)$;
end if
```

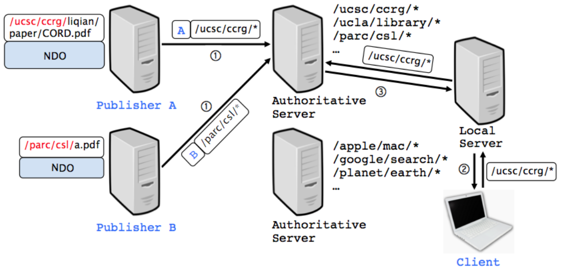
Let N^i be the set of one-hop neighbors of node i. Node i updates DT^i_j as a result of HELLOs from neighbor $j \in N^i$ or the loss of connectivity to neighbor j. If node i loses connectivity to node j, the entries in DT^i_j are deleted. The node, which is predefined as a directory, is the only one that can change the sequence number for its own entry in directory table updates sent in HELLOs.

If node i receives a HELLO from j or a link failure occurs that makes it update DT^i_j for entry $c \neq i : \{nid^i_{cj}, d^i_{cj}, sn^i_{cj}\}$, node i updates its entry for c in DT^i according to Algorithm 1 (DSU), which forces node i to propagate a reset update or to select a successor to directory c that is either closer to c or has reported a more recent sequence number from c.

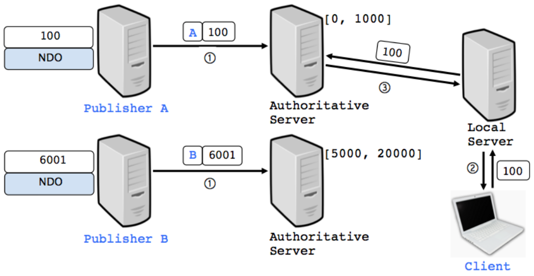
C. Mapping of Content to Directories

All routers maintain multiple loop-free routes to directories using sequence-numbered distances. The number of directories in the network is related to the network size and traffic load, which means the larger network and heavy traffic load the more directories are needed in the network to support scalability and provide highly efficient query processing. With the utilization of consistent hashing, changes of directories size do not affect the correctness and efficiency in CORD. CORD allows directories to announce to the entire network the name prefixes or intervals of NDOs for which they serve as anchors through the HELLO messages they send periodically. A router with content to publish extracts the name space of the NDO, and finds the corresponding anchor from its DT^i.

If hierarchical names are used, directories send updates about the list of the name prefixes for which they are anchors. For instance, a directory could announce being the anchor for the name prefix "/ucsc/ccrg/*" and a content request for the NDO with name "/ucsc/ccrg/liqian/paper/CORD.pdf" would be sent to that anchor.



(a) Mapping of Hierarchical Name



(b) Mapping of Flat Name

Fig. 2: Mapping of Content to Directory

If flat names are used, directories announce the range of NDO identifiers they serve. Assuming an NDO has flat name 100, a directory could advertise a range from 0 to 1,000. Fig. 2 illustrates the consistent mapping to directories using NDO's hierarchical name and flat name when publishing and subscribing to content.

D. Publish-Subscribe Mechanisms

1) Publishing Content: Publishing in CORD consists of having a few local directories know the routes to a given node storing the NDOs in a name prefix or a given range of flat names, and having an anchor directory know the mapping from an NDO range or name prefix to a list of local directories.

Algorithm 2 Publishing to Directories

```
Input: $o^k$, $DT^i$, $LDL^i$, $CST^i$, $NT^i$;
if $o^k \notin CST^i$ then
    /* data object $o^k$ is new to node $i$ */
    $CST^i \leftarrow \{o^k\}$;
    if $i$ is origin of $o^k$ then
        $i$ publish $o^k$ to $a_k$ with $(i \leftarrow \{o^k_i\}, LDL^i)$;
        /* $a_k$ is the anchor directory for $o^k$ */
    end if
    $i$ publish $o^k$ to $l_k = hash(o^k)$ with $(d \leftarrow \{o^k_d\}, i, LDL^i)$;
    /* $l_k$ is the selected local directory from $LDL^i$ for $o^k$ */
end if
```

Each non-directory node i publishes itself with the k directories listed in its *local directory list* (LDL^i). If node i is attached to a content producer, it also publishes the existence of the content with one or more anchor directories, and with its local directories. The local directories in LDL^i are within r hops of node i and serve as the "landmarks" for other nodes to reach node i, given that nodes far away from node i do not have routes to node i. Accordingly, a local directory for node i must maintain updated routes to node i, and it also maintains the mapping $(i \leftarrow \{o_i^1, ..., o_i^n\},\ LDL^i)$, so that it can find alternate ways to reach node i if its route to i fails, and it can resolve subscription requests for content stored at i. The anchor directories are needed for nodes far away from content to obtain the mappings between the content and the local directories of the original content producer as a way to obtain the content. Algorithm 2 explains when and how to publish content to directories. For simplicity, we assume that a single anchor directory is selected for any one original content producer.

The forwarding of a publication message from a node to its local directories is done by the exchange of HELLOs. The routes maintained by local directories to nearby nodes are refreshed periodically based on a HELLO interval. A node that is the original source of an NDO publishes the existence of the NDO by sending a publication message to the anchor directory known to be in charge of the name prefix or NDO range to which the NDO belongs. An anchor directory stores the mapping from the identifier of a node i to the name prefixes or NDO ranges corresponding to NDOs stored at node i, and the list of local directories for node i (LDL^i). Nodes caching an NDO do not publish the NDO with the anchor directory of the NDO; they simply inform their local directories.

The submission of a publication message from node i to an anchor directory regarding NDO o_i^k is done by node i sending the message with the mapping $(i \leftarrow \{o_i^k\},\ LDL^i)$ towards anchor a. Each node v in the route from node i to directory a forwards the publication packet towards a and caches the mapping. Hence, the anchor directory and each node processing a publication message is able to redirect nodes sending subscription requests for NDO o_i^k to the local directories of node i.

2) Subscribing to Content: When node t needs to request NDO o^k, it first sends a subscription request to a local directory in LDL^t, which is selected using a hash function that computes $hash(o^k) = l_t$, where $l_t \in LDL^t$, and sends its subscription request towards l_t.

If directory l_t has received publication from local nodes regarding o^k before, it replies with the identifier of a node p where o^k exists, as well as the local directories of p. If l_t does not know about o^k, it sends a negative reply to node t. At that point, node t sends its request to the anchor directory a_k for o^k, based on the name prefixes and NDO ranges advertised by anchor directories. In turn, anchor directory a_k responds with the mapping $(d \leftarrow \{o_d^k\},\ LDL^d)$ towards the nearest local directory of node t selected from LDL^t. The answer is redirected to t by either the selected directory l_t^j or the first relay node along the path from a_k to directory l_t^j with a route to t. Hence, node t obtains a subscription response from either one of its local directories or the anchor directory of o^k. Node t is then able to send a content request according to the data-plane mechanisms defined for the ICN in which it operates.

IV. PERFORMANCE COMPARISON

We implemented CORD and other content routing protocols using the discrete event simulator QualNet [21] (version 5.0). We ran simulation experiments using the 154-node AT&T topology, which is well known. We compared the performance of CORD with that of NLSR, which is based on the link-state approach, and a loop-free distance-vector approach to content routing. The distance vector approach uses sequence numbers to ensure that routes to destination nodes are loop-free, and nodes learn about all the replicas of each NDO in the ICN.

We use end-to-end delay, control plane overhead, data plane overhead, and packet delivery ratio as our performance metrics. The control plane overhead is the average number of control packets generated by the routing protocols, and the data plane overhead is the average number of data plane packets generated by the routing protocols, including subscription requests sent from data consumers to producers and forwarded content packets. We evaluated the three protocols in wired networks. The three protocols used the same time period to refresh their routing structures. For CORD we used a maximum distance of 3 hops to select local directories. Each simulation ran for 10 different seed values.

A few scenarios were used to evaluate the performance of three protocols in ICN by means of simulation experiments. We randomly selected a few groups in the AT&T topology, and within each group nodes are connected with each other through different paths. Nodes in selected groups can be content consumers requesting content. Each requested NDO exists in the network in such way that it is generated by one original content producer, but may also be cached anywhere else in the network. We investigated the impact of different scenarios on the performance of these three content routing protocols.

A. Impact of Number of Content-Requesting Group

In this scenario we evaluated the impact of increasing the number of consumer groups on the performance of the protocols. We started from 5 groups up to 10 groups, and the number of nodes in each group varied from 10 to 20. To minimize the influence from other parameters, there was only one content consumer from each group in this scenario. In addition, the number of content producers varied from 1 to 10, so that the number of content flows grew with the number of consumer groups at the same pace to avoid content sparsity in the network.

The results of this scenario with performance metrics used in comparison are shown in Fig. 3, where “1-data” means only one content producer existing in the network, and “5-group” means there are five content requesting groups. Given that the three approaches we simulated attain close to 100% delivery in all cases, so we do not show that metric in our results due to space limitations.

Fig. 3-a shows that CORD attains similar end-to-end delays of delivered content when the protocols have the same delivery ratios.

Fig. 3-b shows the average control plane overhead induced by the protocols. CORD incurs much smaller overhead and contrasts with the overhead induced by NLSR, which experiences a steep overhead increase for 10 groups. CORD incurs limited and fairly constant control overhead, because only unicast publish-subscribe requests to directories are sent other than HELLOs. By contrast, NLSR needs to flood link state advertisements (LSA) regarding the existence of new copies

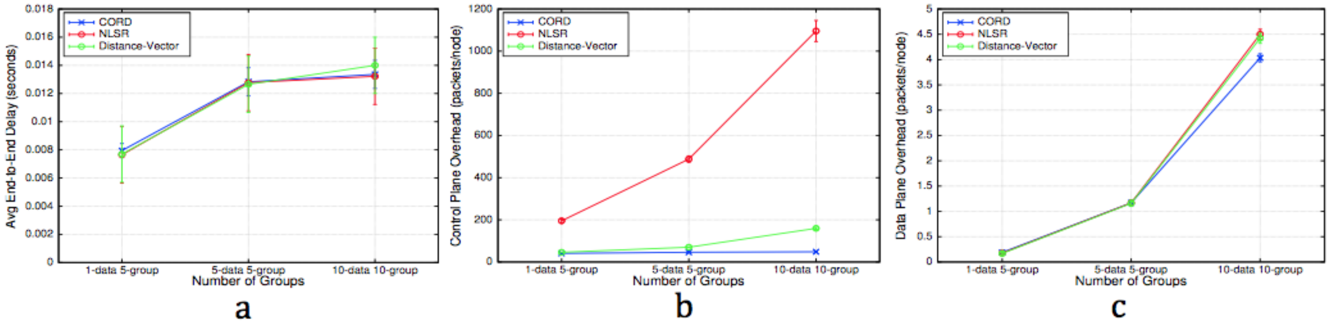


Fig. 3: Impact of increasing number of content requesting groups: (a) End-to-end delay of delivered data packets, (b) average number of control plane packets sent per node, (c) average number of data plane packets sent per node.

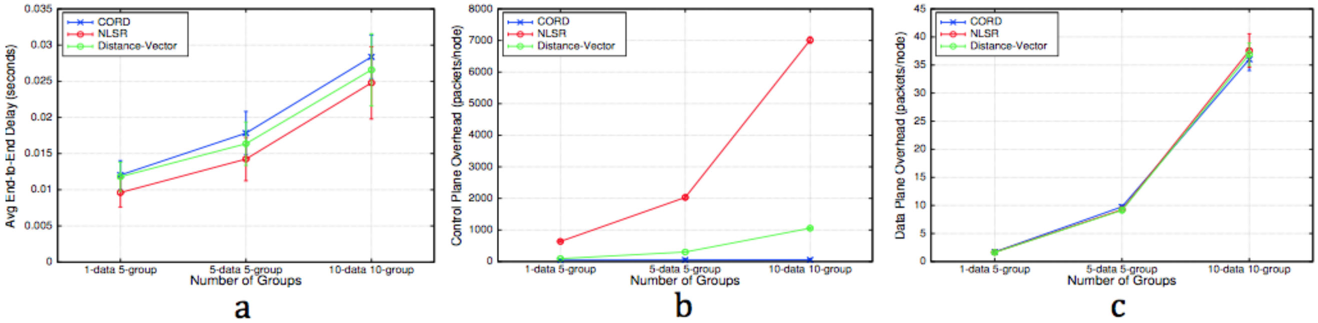


Fig. 4: Impact of increasing number of flows: (a) End-to-end delay of delivered data packets, (b) average number of control plane packets sent per node, (c) average number of data plane packets sent per node.

of NDOs, and the distance-vector routing approach needs to flood the network about the existence of new copies of NDOs at different nodes. Fig. 3-c shows the data plane overhead induced by three protocols. We can see that as the number of groups increases, the other two protocols incur more data plane overhead than CORD.

B. *Impact of Increasing Number of Flows*

In this scenario we increased the number of content consumers, which is the other way to increase the number of content flows, to see the impact on performance. We used the same selected consumer groups; however, nodes in each group are all content consumers and request NDOs at the same time. Fig. 4 shows the results in this scenario.

Fig. 4-a shows that CORD incurs slightly higher end-to-end delays to deliver NDOs than the other two protocols, which is mostly due to the fact that, in some cases, packets may take routes that are slightly longer than the shortest paths attained with the other two approaches.

Fig. 4-b shows the control plane overhead induced by the protocols. CORD incurs very limited and fairly constant control overhead, which contrasts with the overhead incurred by other two approaches for 10 groups with 10 data producers in the network.

Fig. 4-c shows that the data plane overhead induced by the three protocols is similar, even with an increasing number of flows. The key reason why CORD is more efficient than the other approaches is that CORD eliminates the need to communicate information about the network or content replicas.

C. *Impact of Caching Scheme*

To evaluate our caching scheme as well as compare the performance with the impact of caching, this scenario took in-network caching into consideration. We considered "path caching," where all nodes cache content opportunistically when they forward NDOs to requesting nodes; and "edge caching," in which only those nodes that request NDOs cache them. We evaluated the impact of these two caching schemes using the same scenarios described above.

Fig. 5, 6 show the results of using "path caching", and Fig. 3, 4 present the results of using "edge caching". In Fig. 5, there is only one content consumer in each group, whereas nodes are all content consumers in each group in Fig. 6. When we compare Fig. 3, 4 with Fig. 5, 6, "path caching" helps the three protocols to attain slightly lower end-to-end delays to deliver NDOs; however the difference is very small. When we compare the control plane overhead and data plane overhead in the figures, we find the difference by using two caching schemes is very small. Hence, the results indicate that "edge caching" provides most of the advantages of "path caching" with far less storage overhead.

V. CONCLUSION

We introduced CORD (Content Oriented Routing with Directories) as an alternative to content routing in ICNs. CORD eliminates the need for content-based routing tables by establishing routes to directories using distance-vector signaling and by mapping name prefixes or names of NDOs to directories using publish-subscribe mechanisms. CORD

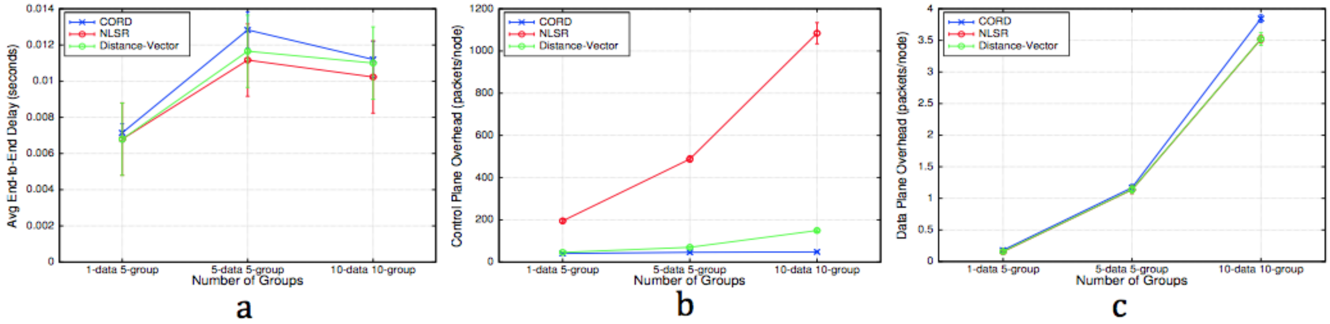


Fig. 5: Impact of caching, one consumer each group: (a) End-to-end delay of delivered data packets, (b) average number of control plane packets sent per node, (c) average number of data plane packets sent per node.

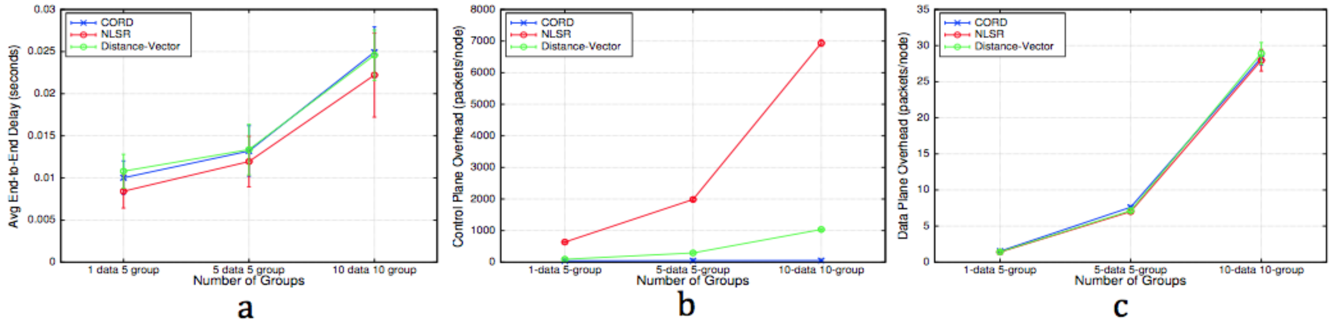


Fig. 6: Impact of caching, nodes are all consumers in each group: (a) End-to-end delay of delivered data packets, (b) average number of control plane packets sent per node, (c) average number of data plane packets sent per node.

constitutes the first approach for name-based content routing based on distance information to directories.

We used simulation experiments to compare its performance with that of name-based routing of content using the link-state approach advocated in NLSR [12] for ICN and a loop-free distance-vector approach. CORD achieves the same high data delivery, attains comparable delays to deliver NDOs, and incurs substantially less control plane overhead than the alternatives. The key reason why CORD outperforms the other name-based routing approaches is that it eliminates the need to maintain topology information or routing information for all the replicas of the same content.

VI. ACKNOWLEDGMENTS

This research was sponsored in part by the Jack Baskin Chair of Computer Engineering at UCSC.

REFERENCES

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

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

[3] A.Carzaniga et al., “A Routing Scheme for Content-Based Networking,” *Proc. IEEE Infocom ’04*, 2004.

[4] A. Detti et al., “CONET: A Content-Centric Inter-networking Architecture,” *Proc. ACM ICN ’12*, 2012.

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

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

[7] J.J. Garcia-Luna-Aceves, ”System and Method for Discovering Information Objects and Information Object Repositories in Computer Networks,” U.S. Patent 7,162,539, 2007.

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

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

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

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

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

[13] J.M. McQuillan, “Enhanced Message Addressing Capabilities for Computer Networks”, *Proceedings of the IEEE* , Vol. 66, No. 11, Nov. 1978.

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

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

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

[17] J. Raju, J.J. Garcia-Luna-Aceves and B. Smith, “System and Method for Information Object Routing in Computer Networks,” U.S. Patent 7,552,233, June 23, 2009

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

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

[20] C Dannewitz, “NetInf: An information-centric design for the future internet”. 2009.

[21] S. N. Technologies. Qualnet. Available: http://www.scalable-networks.com/.

[22] C. Tsilopoulos, D. Makris, and G. Xylomenos, “Bootstrapping a Publish/Subscribe Information Centric Network,” to appear In *Future Network and Mobile Summit 2011*, 2011.