Poster Abstract: Routing Meets Caching in Named Data Networks

Chavoosh Ghasemi[1], Hamed Yousefi[2], Kang G. Shin[2], and Beichuan Zhang[1]
[1]Department of Computer Science, The University of Arizona
[2]Department of Electrical Engineering and Computer Science, The University of Michigan

***Abstract*—In-network caching is a distinct feature of named data networking (NDN). The current NDN routing protocols, however, still focus on the traditional problem of forwarding content requests to content producers, without explicit support of in-network caching, which will limit NDN's potential and benefits to applications. In this poster, as part of a new routing protocol, we equip the forwarding plane with a path which significantly improves the probability of meeting the cached contents before reaching the producers.**

I. INTRODUCTION

As an important representative of Information-Centric Networking (ICN), *Named Data Networking* (NDN) makes a fundamental paradigm shift from host-centric to data-centric Internet architecture [5]. In NDN, content is identified by a hierarchical name, and both the requests (i.e., *Interests*) and the responses (i.e., *Data*) carry the content name rather than a source/destination address. This enables native in-network caching, and thus intermediate routers can cache Data in its returning path to the requester(s) to serve future Interests.

Full utilization of in-network caching capability needs support from the underlying routing protocol. At the heart of NDN, the forwarding engine needs a routing protocol to efficiently compute and install proper forwarding entries in order to forward Interests towards the corresponding content provider(s). While some studies have been done on NDN routing protocols [3], [4] or similar content-centric routing protocols [1], [2], they all focus on the traditional problem of computing the shortest path towards a content producer, without explicit support of in-network caching. Thus, the result is opportunistic for caching. Actually, simply exploiting this opportunity in the forwarding plane can degrade the performance in terms of both content retrieval delay and traffic (Interest/Data) overheads. Fig. 1 shows how ignoring the in-network caching capability in the state-of-the-art can cause forwarding R1&R2&R3's Interests for the same content through late, even never, merging paths (R2&R3 and R1&R2, respectively).

As part of a future comprehensive link-state routing protocol for NDN, our idea is to simply label incoming route announcements and updates (advertised in the case of both topology and name prefix changes) at border routers such that for the same name prefix, all internal routers will select the same border router to forward their Interests through. As shown in Fig. 1, this not only guarantees that requests for the same content will always merge before going out of the network (see R1&R2&R3's paths), but also improves the chance that they merge even before reaching the border router (see R1&R2's paths). Thus, we equip the forwarding plane with a new path (referred to as *MPP*), explicitly exploiting in-network caching in NDN. Our preliminary results demonstrate the effectiveness of utilizing MPPs over the state-of-the-art in NDN.

II. DESIGN

As part of any link-state routing protocol to synchronize all the LSDBs (Link State Databases), each router needs to detect a new update in the case of both topology and name prefix changes and disseminate it throughout the network. (The LSDB at each router contains information on reachability to both routers and name prefixes.) We consider the case where routing announcements/updates are received in an area through multiple border routers. (The areas simply follow the network partitioning in OSPF.) Then, the entry points (i.e., border routers) simply update a field *Modified Time* in receiving announcements/updates—this is referred to as *labeling process*. Finally, the internal routers choose the border router informed of a new name prefix before the others (i.e., that with the least *Modified Time*) and use their shortest paths towards this border router to forward future Interests looking for the same name prefix. Finally, from a given router's point of view, there is a path (maybe longer than the shortest path) which eventually reaches the producer(s), but with a higher probability of satisfying its request by an intermediate router.

Fig. 2 shows an example in which a new `/a/b/c` server announces the content it serves by disseminating a routing update in the network. This advertisement is received by router R1 at the second hop and propagated into the network (follow pentagon-shaped updates) before router R4 (follow triangle-shaped updates) receiving the update at the third hop. Thus, MPPs to `/a/b/c` server from all internal routers in A1 go through router R1. Note that the resolved MPP in a router may be the same as its shortest path (for routers R0 and R2), or different (for router R3).

Although MPPs may take longer paths than the shortest paths, they can effectively reduce content retrieval delay. Besides, by presenting MPP, we prevent scattered caching and forwarding Interests through improper paths or towards deprecated copies of contents. Moreover, by avoiding sending several similar Interests throughout the network, MPPs reinforce the role of PIT (Pending Interest Table), as one of the main NDN design principles. This way, we save more network bandwidth and decrease the possibility of congesting the intermediate links/routers, and reduce the transport cost incurred by traffic between networks. By receiving fewer Interest packets at the producers, the servers' load will also be reduced.

III. EVALUATION

We compare the NDN forwarding based on MPPs versus shortest paths (utilized by NLSR [4], as the current *de facto* routing protocol of NDN testbed) via ndnSIM. Fig. 3 illustrates a network of 44 nodes partitioned into three areas, where four consumers request 20 name prefixes served by one server. In order to test the network operation, each time step is

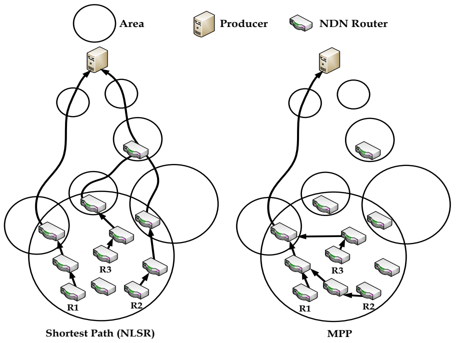


Fig. 1: Utilizing MPPs to merge the forwarding paths for the same content as early as possible (in the worst case, at the same border router). (The areas simply follow the network partitioning in OSPF.)

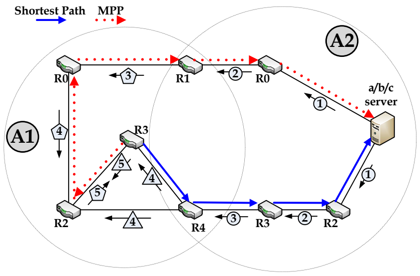


Fig. 2: Update dissemination after adding the `/a/b/c` server. The blue and red lines illustrate the shortest path and MPP, respectively, from R3 to retrieve `/a/b/c`

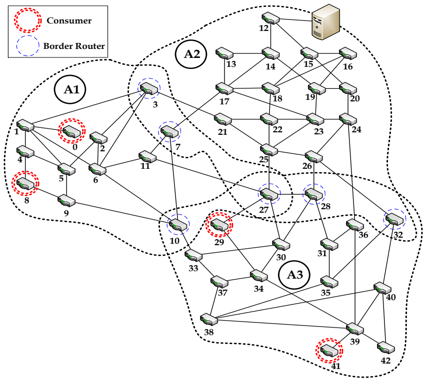
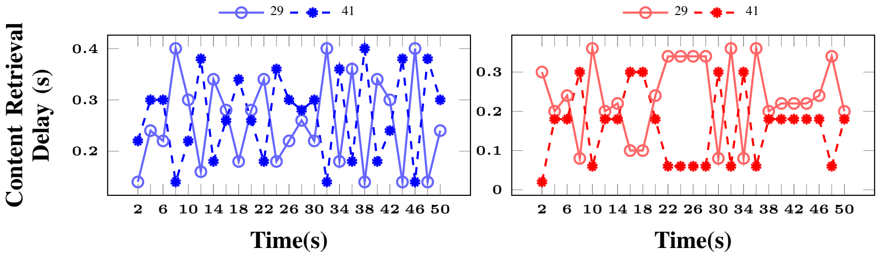


Fig. 3: Network topology to evaluate the MPP performance



(a) Shortest Path (NLSR) **(b)** MPP

Fig. 4: Content retrieval delay in nodes 29 and 41 using different forwarding schemes

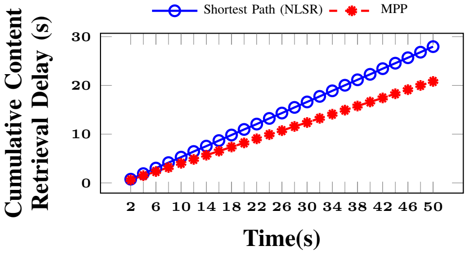


Fig. 5: Cumulative content retrieval delay using different forwarding schemes

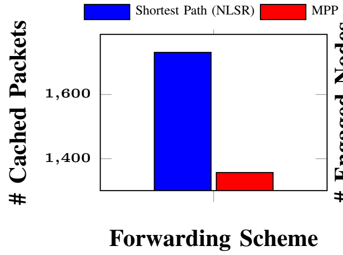


Fig. 6: Number of packets cached in the network using different forwarding schemes

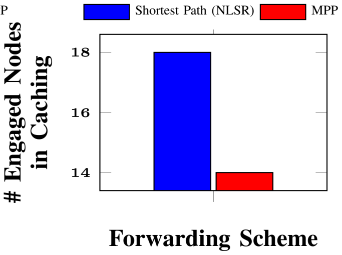


Fig. 7: Number of nodes engaged in caching using different forwarding schemes

randomly chosen between 0 and 4 seconds, in which each of four consumers requests one of the prefixes. At the end of simulation that lasts for 50 seconds, each node has requested at least a half of the existing name prefixes. Thus, it is highly possible that a content is repeatedly requested by different consumers (time-locality principle).

Fig. 4 shows the content retrieval delay for nodes 29 and 41 (A3's consumers). This delay for the nodes under the MPP-based scheme is shown to be smaller than that under the shortest paths (NLSR). Indeed, although leveraging only shortest paths can result in the same or even better performance under a special condition (when the content requisition does not follow the time locality principle), MPPs can in general reduce the retrieval delay. Besides, nodes 29 and 41 have relatively opposite retrieval delay trends (i.e., high vs. low). This verifies that by leveraging in-network caching, if a node needs a content which has already been requested, it will meet a cached version with high probability. Fig. 5 shows that the MPP-based scheme outperforms the shortest-path-based scheme (NLSR) by an average of 26% in terms of the cumulative retrieval delay for all consumers. Furthermore, Figs. 6 and 7 show the MPP-based scheme reduces the overall cache memory usage and the number of nodes engaged in caching a specific content on average by 27% and 22%, respectively. Obviously, fewer engaged nodes mean less scattered caching while also providing a lower retrieval delay.

Knowing the benefits of using MPP over shortest paths in different aspects, we can reject adopting the assumption of several studies (like NLSR) which implies using path cost as a metric to rank the available paths. Instead, we believe that the paths with a higher probability to meet the cached content have higher priority over the shortest paths towards the provider(s) in NDN.

IV. FUTURE WORK

As another important part of a comprehensive routing protocol, it is expected to install multiple paths towards the content provider(s). In future, we plan to study multipath forwarding by employing MPP and evaluate how it improves network resiliency and survivability.

ACKNOWLEDGMENTS

This work was supported in part by NSF under Grants CNS-1345142 and CNS-1629009.

REFERENCES

[1] J. Garcia-Luna-Aceves, "Name-based content routing in information centric networks using distance information," in *ICN*, 2014, pp. 7–16.
[2] E. Hemmati and J. Garcia-Luna-Aceves, "A new approach to name-based link-state routing for information-centric networks," in *ICN*, 2015, pp. 29–38.
[3] A. Hoque *et al.*, "NLSR: Named-data link state routing protocol," in *ICN*, 2013, pp. 15–20.
[4] V. Lehman *et al.*, "A secure link state routing protocol for NDN," Tech. Rep. NDN-0037, Jan. 2016.
[5] L. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.