A Two-layer Intra-domain Routing Scheme for Named Data Networking

Huichen Dai, Jianyuan Lu, Yi Wang, Bin Liu
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University
{dhc10, lu-jy11}@mails.tsinghua.edu.cn, wy@ieee.org, liub@tsinghua.edu.cn

***Abstract*—Routing is undoubtedly the foundation of NDN's data transmission service. We propose a two-layer routing protocol for NDN [1], [2], which is composed of a Topology Maintaining (TM) layer and a Prefix Announcing (PA) layer. The underlying layer (TM) maintains the full topology of an NDN network domain and calculates the shortest-path trees. The upper layer (PA) provides content in two ways: *active publishing* and *passive serving*. However, solely adopting either of them will lead to the problem of scalability. We compare the efficiency and cost of the two methods, and evaluation results show that active publishing is much more efficient than the passive serving method in terms of triggered traffic, but actively publishing all the content will lead to Forwarding Information Base (FIB) explosion. Therefore, we further propose a *popularity*-based active publishing policy and arrive at a compromise between the active and passive methods. Moreover, we put forward several methods to aggregate FIB entries, and the FIB size shrinks effectively after aggregation. This routing protocol is compliant with the NDN characteristics and supports NDN multipath routing.**

***Index Terms*—Named Data Network (NDN), Two-layer Routing Protocol, Active Publishing, Passive Serving**

I. Introduction

Named Data Networking (NDN) [1], [2] has been proposed as a clean-slate future Internet architecture, which concentrates on the content itself, rather than its location. NDN uses names to identify every piece of content, and routes packets by names.

Routing protocol is undoubtedly the foundation of NDN's data transmission service. Though NDN routers can cache content strategically and serve subsequent requests, there must be a protocol that routes requests to content providers if the required content is not cached. As far as we know, there is no routing protocol proposed for NDN yet.

An IP router or host is passively assigned an IP address by an ISP, and it uses the IP address to communicate with other network nodes via address-based IP routing. However, NDN routing is name-based, instead of passive address assignment, an NDN router can *actively* announce/publish *name prefixes* of contents it wants to serve[1]. These announcement packets are propagated throughout the entire network domain, and every router builds its Forwarding Information Base (FIB) based on received announcements. Requests are routed to content providers by looking up the FIBs. Naively flooding these announcements may lead to tremendous traffic and this calls for an efficient solution. Moreover, each announcement will be recorded in the FIB, thus publishing a large number of contents will result in an FIB with numerous entries, *i.e.* FIB explosion.

Contents in NDN can also be passively served, *i.e.*, no announcement about these contents is published, thus FIB does not need to record their information. This is the most preliminary content serving mechanism that NDN provides. When these contents are requested, the requests are propagated via flooding, which incurs a great burden on the network. Therefore, active publishing mode can lead to FIB explosion, while passive content serving mode may result in traffic explosion.

NDN has better support for multipath routing[2] than IP. However, it is of significant importance to point out that NDN multipath property is different from conventional IP multipath. IP multipath means more than one path between a specific pair of source and destination, while NDN multipath means a host can obtain data from multiple content providers via multiple paths. IP routing adopts a single best path to prevent loops. On the contrary, an NDN router can send out a request using multiple interfaces without worrying about loops.

The above characteristics of NDN routing requires a routing scheme that is distinctive from existing ones. we summarize and highlight the following challenges:

1) *Efficient announcement forwarding:* In active publishing mode, the announcements of contents are propagated throughout the NDN network domain. There should be an efficient announcement forwarding solution rather than naively flooding them.
2) *Scalability:* Adopting only active publishing method or passive serving method is problematic and not scalable. A compromise between them should be reached to enable an NDN routing protocol adapt to large-sized network with massive published contents, while keeping the FIB size reasonable and acceptable.
3) *Multipath support:* Multipath property is an elegant feature of NDN routing. A qualified NDN routing protocol should support forwarding requests over multiple paths.

In this paper, we propose, simulate and evaluate a two-layer intra-domain[3] routing protocol for NDN. The underlying layer maintains full topology information of the entire NDN network domain, such as connectivity and bandwidth, and calculates the shortest-path tree rooted at each node, thus is called the Topology Maintaining (TM) layer. This layer provides the shortest paths information as a service to the upper layer.

The upper layer is the Prefix Announcing (PA) layer, which is laid on the TM layer and publishes the content that the router wants to serve by sending out announcements. The announcements are forwarded to all the other nodes via the single source shortest-path tree rooted at this router, which is provided by the

[1]In fact, NDN hosts/servers first announce the name prefixes to the directly connected routers, then the routers announce the prefix to the entire network. For brevity, we omit a description of the server announcing process.

[2]NDN multipath routing will be introduced in Section III-C.

[3]Though NDN is still in the research stage, we believe that there will also be similar administrative regions equivalent to IP domains or ASes, thus we continue to use the terms like *domain* and *AS* in NDN.

TM layer. Routers also build their own FIBs based on received announcements at this layer.

Fig. 1 shows the routing hierarchy, the upmost data communication layer forwards request packets by looking up FIBs and is unaware of where the requested data is located.

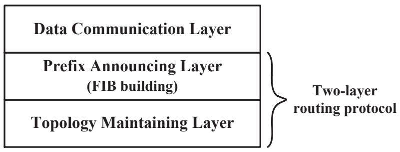


Fig. 1. The layered routing hierarchy.

To solve the aforementioned challenges, we make the following contributions.

1) We propose a two-layer routing protocol for NDN, which decouples the announcement forwarding and FIB building from topology maintaining. Each layer focuses on their own responsibilities. TM layer maintains the topology information, and PA layer efficiently forwards announcements via the shortest-path trees, keeping the announcement traffic to a minimum in active publishing mode.
2) We compare the efficiency and cost of active publishing and passive serving method. Evaluation results show that active publishing method achieves much higher efficiency than passive serving method in terms of triggered packets, but solely adopting it will lead to the problem of FIB explosion. By proposing a *popularity*-based active publishing policy, a compromise between the two methods is reached. Moreover, some FIB entry aggregation methods are also proposed.
3) Due to routers actively announcing names or name prefixes, NDN multipath routing is easily fulfilled by our proposed routing protocol.

The rest of the paper is organized as follows. Section II introduces the communication mechanism of NDN. The details of the two-layered routing scheme is presented in Section III. Section IV addresses the scalability problem. We evaluate the routing scheme in Section V. Section VI is the related work and we conclude our research in Section VII.

II. NDN Background

NDN is a newly proposed future network architecture, which concentrates on the content itself, rather than *where* the content locates. Node in NDN network includes three tables: Forwarding Information Base (FIB), Pending Interest Table (PIT) and Content Store (CS). Communication in NDN is requester-driven, *i.e.*, a consumer sends out a request, or an *Interest* packet, that carries a name that specifies the desired data. When the Interest arrives at an NDN router, the node conceptually consults the three tables in the order of CS, PIT and FIB. The router first checks if the desired data is already cached in the CS, which buffers the content that traverses this router. If the requested data has been cached in the CS, a *Data* packet that carries both the name and content is returned by this router. Otherwise, the router further checks PIT whether a request for the same data has recently been issued. The PIT records the arrival interfaces of Interests that have not been satisfied. If found, append the interface on which the Interest arrives to the interface list of the matched PIT entry. Otherwise, a new PIT entry is created, and the router continues to consult the FIB to determine where to forward the Interest.

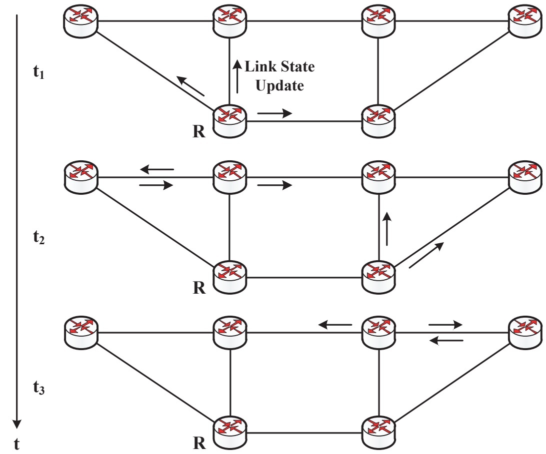


Fig. 2. The flooding process of Link State Updates. Arrows represent Link State Updates (LSU). At t_1, router R sends LSU to its three adjacent routers. Then at t_2, these three routers continue to forward the received LSU to their own adjacent routers, excluding the upstream router, so and so forth.

Announcement

Source Router ID	
Prefix1	property
Prefix2	property
Prefix3	property
......	......

Fig. 3. The format of announcement packet.

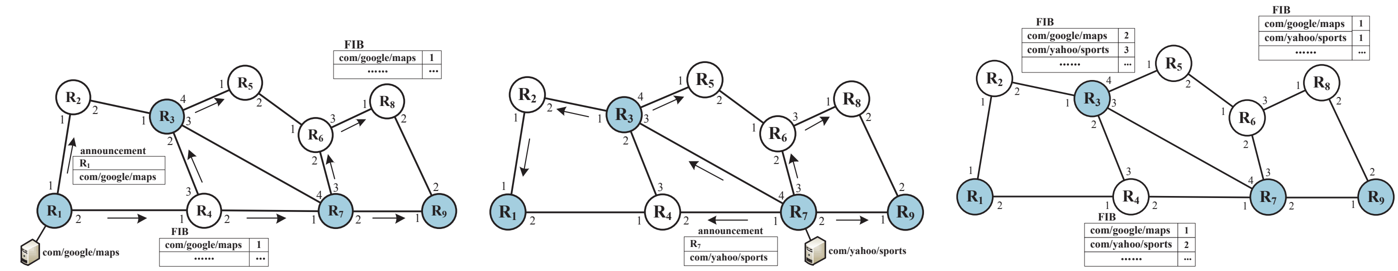
When the Interest arrives at a node that can satisfy the request, *i.e.* the node serves the desired data, the desired content is sent back in a *Data* Packet, which will follow the reverse path of the Interest that requests it. While Data packets traverse a router, the packets may be cached in the CS before being relayed to the requesting interface(s) indicated by the corresponding PIT entry. This PIT entry is subsequently removed.

The naming system is an important piece in the NDN architecture and is still under active research. NDN names are hierarchically structured and are usually composed of explicitly delimited *components*. For the purpose of early exploring the name-based routing mechanisms before the naming specification finally goes to standards, in this paper, we temporarily use *hierarchically* reversed domain names as NDN names. For example, *maps.google.com* is hierarchically reversed to *com/google/maps* ('/' is not part of the name), and *com, google, maps* are three components of the name. Our proposed routing protocol is a *generic* scheme that suits all the naming strategies that satisfy the NDN naming characteristics.

III. NDN Routing

A. Two-layer NDN Routing Scheme

1) Topology Maintaining (TM) Layer: The TM layer is engaged in topology discovering, node/link failure handling, and shortest-path tree calculating, which resembles OSPF's link state information exchange functionality very much. We define a similar concept for NDN – Link State Advertisement (LSA) – an unit of data describing the local state of a router, including the state of the router's interfaces and adjacencies. Every router sends out LSAs (encapsulated in Link State Updates) to other routers, and each LSA is flooded throughout the routing domain. The collected link state advertisements forms the protocol's Link State Database. All the routers get their Link state Databases synchronized at last. Fig. 2 depicts the flooding process of Link State Updates.



(a) R_1 announces its prefix *com/google/maps*. R_4 receives the announcement via interface 1, then builds an FIB entry, and further forwards the announcement to R_3 and R_7.

(b) R_7 advertises its prefix *com/yahoo/sports*.

(c) The FIBs of nodes after R_1 and R_7 announce their prefixes, R_3, R_4 and R_8 are selected as representatives.

Fig. 4. Prefix announcing layer. A sample NDN topology, where solid nodes represent content providers, and hollow ones stand for content requesters.

In order to distinguish a router from others, each router is assigned a device name or ID. It's worth pointing out that, in consideration of routing scalability, the device name should also share the features of NDN name, *e.g.* hierarchical and aggregatable. However, for simplicity and brevity of NDN naming strategy design, device name and NDN content name are not necessarily in the same namespace. In other words, we decouple the device name from the NDN content name, and they have no explicit mapping relationship. In fact, if NDN is laid on IP, the TM layer can adopt OSPF "as-is", and the device ID is simply IP address. NDN content names can adopt the aforementioned hierarchically reversed domain names. Due to the similarity between the TM layer and OSPF, in this paper, we omit some details of the TM layer and focus on the PA layer.

2) Prefix Announcing (PA) Layer: The PA layer is responsible for providing content in both active and passive modes. Here we mainly introduce the active publishing mode, and introduce the passive serving method in Section IV. In active publishing mode, PA conducts efficient prefix announcing and FIB building. Routers can be grouped into content requesters and content providers. As the name – Prefix Announcing layer – suggests, each content provider *actively* advertises the contents it can serve by sending out name prefix announcements, *i.e.*, disseminating announcements to the entire network via the shortest-path trees. Each router builds its FIB based on received announcements and aggregate some entries if possible. An announcement packet, generated by content provider R, includes the name prefixes that this router can serve, the router's ID, as well as some properties (*e.g.* popularity) of the name prefixes. The format of an announcement is illustrated in Fig. 3.

Without loss of generality, we abstract this layer by two functions: an announcement forwarding function f and an FIB building function b. The forwarding function $f : N \times N \rightarrow I$, which, at each router r, given a source node s, returns a *set* of output interfaces $f(s, r)$. N is the set of all the routers in the network, and I is the set of interfaces of the current router r. The building function $b(name_1, name_2, \cdots)$ builds FIB based on received name prefixes.

A layering idea was also observed by [3], while it called for the all-pairs path symmetry property on the upper layer, which was implemented by a global minimum spanning tree in [3]. A spanning tree algorithm treats the network topology as an undirected graph, and there is only one path $u \rightsquigarrow v$ between each pair of node $<u, v>$, which is not necessarily the shortest path. However, our forwarding function f is based on the single-source shortest paths. In this algorithm, network topology is treated as a directed graph and per source computes its own shortest-path tree, which is more coincident to the complicated real network environment. Moreover, the communication mechanism in [3] is different from NDN.

An example of announcement publishing process is shown in Fig. 4(a). Content provider R_1 sends out its announcement, which conveys a prefix: *com/google/maps*. Arrows stand for announcements and their forwarding directions, which imply the shortest paths from R_1 to other routers. When an intermediate router, *e.g.* R_4, receives an announcement, R_4 first invokes function $b(\text{“}com/google/maps'')$ – adds the prefixes carried by this announcement and its arrival interface into FIB, and aggregates some FIB entries (see Section III-C) if possible. Then R_4 extracts the source router ID, R_1, and invokes $f(R_1, R_4)$ to obtain the interface set through which to forward the announcement. In this example, $f(R_1, R_4)$ returns $\{R_3, R_7\}$, and then R_4 forwards the announcement to R_3 and R_7. Fig. 4(b) shows the process of another content provider, R_7, publishing it announcement. Note that due to the complicated network environment, the shortest path from $R_1 \rightarrow R_7$ differs from the shortest path from $R_7 \rightarrow R_1$. After each content provider announces it prefixes, the FIB in each router has the same set of prefixes, but different next-hop interfaces for identical prefixes, as depicted in Fig. 4(c).

B. Requesting Data

After content providers publishing the name prefixes and all the FIBs synchronized, end hosts can solicit data. Content requester sends out Interests that convey the names specifying the desired data. Each time a router receives an Interest, it looks up the name in FIB for the next-hop interface (if not found in PIT), and forwards the Interest to that interface. When the Interest arrives at the content provider, a Data packet is returned by taking the reverse path of the Interest. Fig. 5 illustrates a data requesting process. R_8 wants the data *com/google/maps/USA/CA/Anaheim* and sends out an Interest (dashed arrows). Intermediate routers forward the Interest by looking up FIBs. E.g., $R4$ finds a matched FIB entry according to the Longest Prefix Match principle, and forwards the Interest to interface 1. The Interest arrives at R_1 at last and the responding Data packet (two headed arrow) is symmetrically routed back.

C. Multipath Support

As aforementioned, the NDN multipath routing means a host can obtain data from multiple content providers via multiple paths, which is different from IP multipath routing [4]–[6]. By actively announcing names or name prefixes, the two-layer protocol inherently support multipath routing. For example, Fig. 6 describes that router R_1 and R_7 both provide content

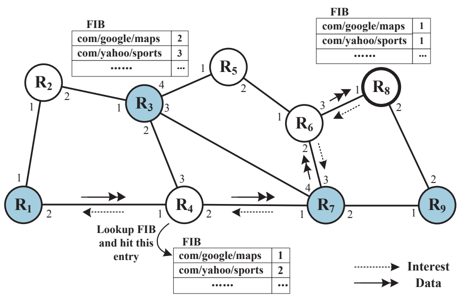


Fig. 5. The data requesting process. R_8 is the requester.

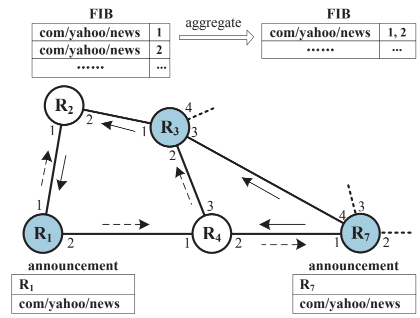


Fig. 6. Multipath Support and Aggregating CDN-like FIB entries.

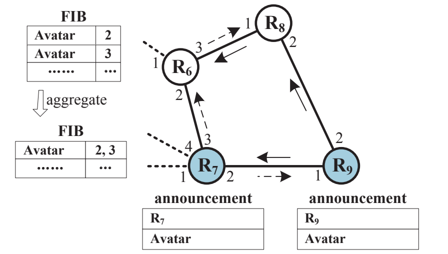


Fig. 7. Multipath Support and Aggregating P2P-like FIB entries.

com/yahoo/news, and send out announcements (dotted and solid arrows), respectively. R_2 receives both announcements and then update its FIB by two entries (which can be further aggregated, refer to Section IV-C). When R_2's Interest favors content *com/yahoo/news*, it will be forwarded along two paths: $R_2 \to R_1$ and $R_2 \to R_3 \to R_7$. In Fig. 7, two servers R_7 and R_9 are both willing to serve the movie *Avatar*, and publish their announcements respectively. If R_6 wants this content, its Interest will be forward along $R_6 \to R_7$ and $R_6 \to R_8 \to R_9$.

IV. Scalability

Each actively published popular name prefix will be recorded in FIB on each router. If a network domain consists of a large amount of nodes and their PA layers publish massive contents, FIB will increase into an extremely large size. Consequently, scalability is an important aspect of the NDN routing protocol.

A. Active Publishing and Passive Serving on the PA Layer

A server may possess too many contents that it is unrealistic to actively publish each of it due to the reason above. Thus we let content providers *actively* publish the content that is *popular* or the provider is willing to serve. And let other contents be *passively* served. When a passively served content is requested for the first time, the corresponding Interest packet is forwarded in a query mode, *i.e.*, through flooding since routing information for these contents has not been established. Though TM layer can provide shortest paths information to each node, the Interest packet does not contain any information about the data requester, and a shortest-path tree cannot be calculated by intermediate routers for these Interests since the root is unknown, thus they can only be disseminated via flooding. When an Interest reaches a node that has the requested data, a Data packet is routed back. A relevant FIB entry may be generated on routers that routes back the responding Data packet, but it depends on a policy.

Passive content serving is the most preliminary function that NDN provides, but it is quite low-efficient and expensive in terms of the triggered flooding traffic. Below we compare the efficiency (in terms of *Interest* packets) of active publishing and passive serving when requesting a piece of content for the *first time*. Given a network domain with N nodes, the degree of each node is d_i $(i = 1 \sim N)$, and node 1 is a content provider.

In passive serving mode, a node, say node N, wants the content on node 1 for the first time. It first floods its Interests throughout the entire network domain, each node broadcasts the Interests except node 1. The number of total triggered *Interests* is $P_{passive} = d_N + \sum_{i=2}^{N-1}(d_i - 1) = \sum_{i=2}^{N} d_i - N + 2$.

In active publishing mode, Node N requests node 1's content by sending out an Interest via the shortest path to node 1. The number of triggered *Interests* is $P_{active} = L_{N,1}$. $L_{N,1}$ is the number of edges of the shortest path from node N to node 1.

We define a variable, *Requesting Efficiency*, as $RE = P_{passive}/P_{active}$. RE is so large (refer to Section V) that active publishing accomplishes the same task as passive serving, but much more efficiently, only at the cost of some extra protocol packets (announcements). The number of protocol packets equals to the number of edges of the shortest-path tree routed at the content provider. When more than one nodes request node 1's content for the first time, RE will be even higher, since different nodes have to flood their Interests respectively in passive mode, while in active mode, all the nodes can access node 1's content via shortest paths.

Due to the high efficiency of active publishing, it seems that we should publish all the contents. However, publishing too many contents can lead to the explosion of FIB, and passive serving all the contents can result in the explosion of traffic. Therefore, in next subsection, we propose a popularity-based publishing strategy on what content should be published.

B. Popularity-based Active Publishing and FIB size suppression

In order to find a compromise between active publishing and passive serving, we propose that content provider actively announce the *popular* contents, and the unpopular contents be passively served. We define the access *frequency* as the *popularity* property of a content. When the popularity exceeds a threshold, it becomes a popular content and get published. It is not easy for a content provider to find an appropriate threshold, since a small threshold leads to large-sized FIB, while a large one causes numerous flooded Interests (This will be evaluated in Section V, and we arrive at a compromise point).

Moreover, a router can suppress its FIB size by adopting a popularity threshold as well. And this requires content providers add popularity property in its announcements. Once a router receives an announcement a, it first compares a's popularity with its threshold p and determine whether to insert it into FIB or ignore it. Each router maintains a different threshold based on its own memory size and other factors. In each FIB, the popularity of a content is not constant, but varies based on the incoming Interests. If many Interests solicits a particular content, its popularity increases, otherwise decreases. And the router can determine to delete an entry if its popularity degrades to less than the threshold. Fig. 8 shows the process of popularity-based FIB size suppression on routers.

C. Name Aggregation in FIB

FIB size can also be reduced in another dimension. NDN names are hierarchically structured and composed of explicitly

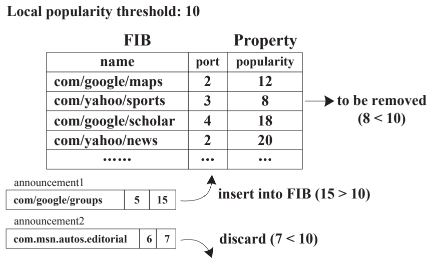


Fig. 8. Popularity-based FIB size suppression.

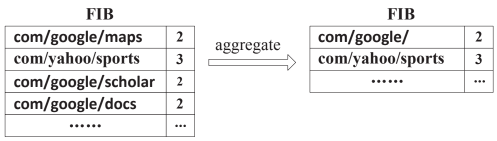


Fig. 9. Aggregating FIB entries: the 1^{st}, 3^{rd} and 4^{th} entries are aggregated into one entry.

delimited *components*, thus multiple NDN names can share a common components-wise name prefix, *e.g. com/google/maps*, *com/google/scholar* and *com/google/docs* share a common prefix *com/google*. If two or more names in the FIB share a common prefix and have the same next-hop interface, these entries can be aggregated into one entry, as depicted in Fig. 9.

Furthermore, NDN's support of multipath routing also enables another two name aggregation scenarios: 1) websites with CDN-like caches, 2) P2P-like file sharing. The first scenario is illustrated in Fig.6, where router R_1 and R_7 both provide content service for *www.yahoo.com* and send out announcements respectively. When a router, say R_2 receives these two announcements, it aggregates them into one entry, which has two next-hop interfaces. The second scenario is depicted in Fig.7, where two servers R_7 and R_9 are both willing to serve the movie *Avatar*, and publish their announcements respectively. After receiving the two announcements, R_6 aggregate them into one entry.

V. Evaluation

Evaluation of the two-layer NDN routing protocol is conducted by simulation in 3 aspects: 1) functionality correctness of our protocol simulation; 2) traffic comparison between active publish and passive serving methods; 3) scalability, including popularity threshold p calculation and FIB size suppression.

A. Experimental Setup

The topologies adopted by our experiments are flat router-level topologies generated by BRITE [7]. The name prefixes we use are hierarchically reversed domain names obtained from DMOZ [8] and ALEXA [9]. For the TM layer, we omit its topology discovering process since topology is known to us by BRITE, so its major task is shortest paths calculation. Thus we focus on simulating the PA layer by implementing the announcement forwarding function f and FIB building function b on a 2.8 GHz CPU with 4 GB memory.

B. Experimental Results

1) Functionality correctness: First we simulate if functions f and b have been correctly implemented in terms of functionality. In a simulated 100-node[4] network, we let 50 nodes publish a prefix respectively, each node forwards the announcements by invoking f and builds FIB by invoking b. Then, the left 50 nodes send out 100 Interests (2 for each) asking for these contents. Fig. 10 shows the CDF of satisfied Interests, revealing that all the Interests are satisfied at last. Next we simulate multipah functionality. We divide the 50 content providers into 25 pairs, and each pair publishes an identical piece of content. The left 50 nodes also send 100 Interests to request these contents, if two Data packets are returned for each Interest, this Interest is satisfied. The corresponding CDF curve is also illustrated in Fig. 10, which shows that all the multipath Interests are satisfied at last.

2) Traffic comparison: Passive content serving is really expensive since it triggers too many Interests, thus costs too much bandwidth and forwarding energy. We evaluate the Requesting Efficiency RE introduced in Section IV-A. In this experiment, a node i requests different contents for the first time, both in active mode and passive mode. Fig. 11 shows the values of RE when node i sends out 1 to 10 requests in a domain with 100, 200 and 300 nodes, respectively. The results reveal that active mode is around 182.8, 321.1 and 435.8 times more efficient than passive mode for 100-, 200- and 300-node domains, respectively. Moreover, a content provider can publish multiple, say M, pieces of contents in a single announcement, the RE can be roughly M times higher when requesting all the M pieces of contents for the first time.

In active publish mode, the total cost C_{active} to request a specific content for the first time is the sum of protocol packets (announcements) and Interests. Suppose the content provider is node j and the requester is i, thus $C_{active} = E_j + L_{i,j}$, where E_j is the number of edges of the shortest path tree rooted at node j. While in passive mode, the corresponding cost $C_{passive}$ equals to $P_{passive}$ (Section IV-A). We define *Cost Ratio* as $CR = C_{passive}/C_{active}$.

We calculate the CR of requesting a particular piece of content for 1, 2, $\cdots$, n times by different nodes in both passive and active mode. Assume that, after requesting P times, this content is actively published[5], and $n < P$. The results of CR is shown in Fig. 12, CR increases as n grows, which means that the total cost of requesting a piece of content under active publishing is very small compared to that under passive mode.

3) Compromise between active mode and passive mode: In Section IV-B we propose a popularity based policy deciding which content to be actively published. If a content's popularity exceeds a threshold p, it is actively published. However, it is non-trivial to calculate a threshold p since we want to keep FIB size and traffic as low as possible. Assume that, in a 100-node domain, there are 500 unique contents, with their popularities obeying Gaussian distribution (the average is 100). Fig. 13 illustrates a relation between FIB size and traffic (Interests) requesting all of these contents for once. As p increases, less contents are published and FIB size decreases, however, more traffic (Interests) are triggered due to more contents are passively served. We finally arrive at a compromise point with $p \approx 47$, where the FIB size and traffic volume are both acceptable.

4) FIB aggregation: At last we evaluate the effects of prefix aggregation. We select 50 routers from a domain with 100 nodes and each of them announce a number of prefixes selected from a name prefix set obtained from DMOZ and ALEXA. To enable the name aggregation, we let some names share a common prefix, or let some nodes publish the same prefix to simulate CDN-like or P2P-like scenarios. Among all the announced prefixes,

[4]The average degree of each node is 3 in our experiments.

[5]This means the passive mode has to flood Interests for n times.

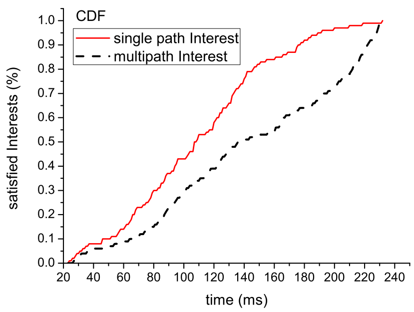


Fig. 10. CDF of satisfied Interests.

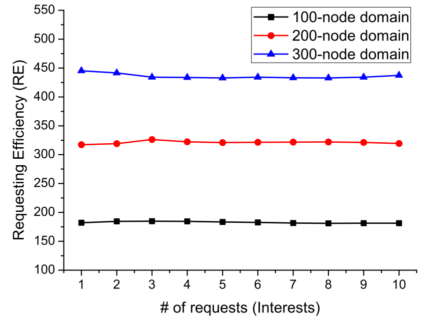


Fig. 11. Requesting Efficiency (RE) of passive publishing and active serving methods.

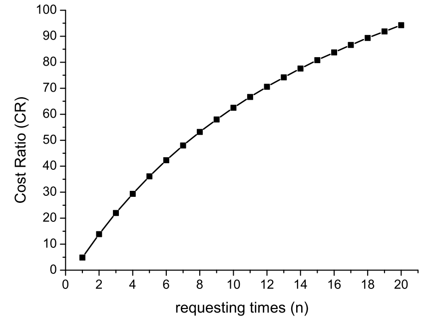


Fig. 12. The Cost Ratio (CR) of requesting a piece of content n times by different nodes.

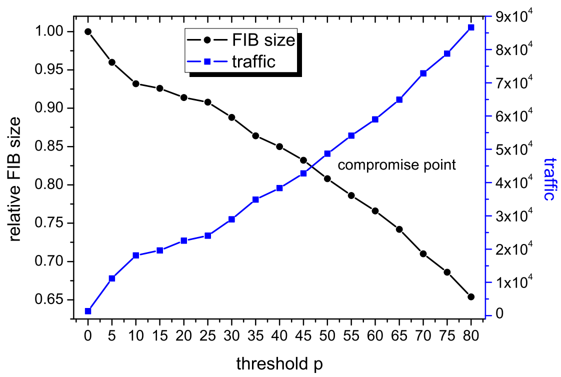


Fig. 13. Results of threshold based policy.

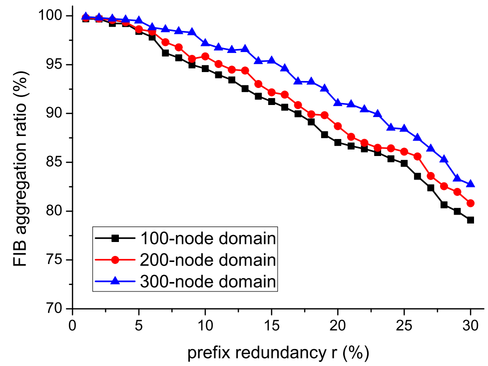


Fig. 14. FIB aggregation ratio.

we denote the number of unique prefixes (neither identical nor share a prefix) by M, and number of those that are identical to or share a common prefix with the unique ones by N. And prefix redundancy r is defined as $r = \frac{N}{M+N}$. The results of name aggregation are shown in Fig. 14, where the y-axis is the aggregation ratio, i.e., the ratio between the number of FIB entries after and before name aggregation. The FIB size shrinks effectively as r increases. We re-conduct the experiment on 200-node and 300-node domains and get similar results.

VI. Related Work

NDN/CCN [1] [2] architecture was proposed recently, but the thought of content-centric or content-based network dates back to a long time ago [10]–[13]. However, most of previous works about content-based networking are based on publish/subscribe systems [3], [14], [15], whose data communication mechanism is different from that CCN/NDN. Some routing protocols [3], [15] under the publish/subscribe framework have been proposed. Particularly, Antonio Carzaniga *et al.* [3] also proposed a layered routing protocol, which requires all-pairs symmetry routes and involves complicated schemes and calculations when actually asking for data. [10] proposed an architecture for content routing in current Internet. This architecture is composed of two protocols – Internet Name Resolution Protocol (INRP) and Name-Based Routing Protocol (NBRP), the former is reverse compatible with DNS, and the latter resembles BGP very much. Moveover, its content routing policies must be consistent with IP routing policies since the decisions made at the content level are eventually carried out by the IP forwarding level.

VII. Conclusion

We present a two-layer (TM layer and PA layer) routing protocol for NDN architecture. TM layer maintains the full topology of a NDN network domain and calculates the shortest-path trees. PA layer provides content in two ways: active publishing and passive serving, but solely adopting either of them is problematic due to scalability. We evaluate and compare the two methods in terms of efficiency and cost, and the proposed popularity-based active publishing policy offers an opportunity to find a compromise between the two methods. Several FIB aggregation methods are also put forward to shrink PIT size, and this routing protocol well supports NDN multipath routing.

References

[1] L. Zhang, D. Estrin, V. Jacobson, and B. Zhang, "Named Data Networking (NDN) Project," in *Technical Report, NDN-0001*, 2010.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proc. of ACM CoNEXT*, 2009.

[3] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A routing scheme for content-based networking," in *Proc. of IEEE INFOCOM*, 2004.

[4] W. Xu and J. Rexford, "Miro: Multi-path interdomain routing," in *Proc. of ACM SIGCOMM*, 2006.

[5] A. Kvalbein, C. Dovrolis, and C. Muthu, "Multipath load-adaptive routing: putting the emphasis on robustness and simplicity," in *Proc. of IEEE ICNP*, 2009.

[6] Y. Liao, L. Gao, R. Guerin, and Z.-L. Zhang, "Reliable interdomain routing through multiple complementary routing processes," in *Proc. of ACM CoNEXT*, 2008.

[7] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: an approach to universal topology generation," in *Proc. of International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, MASCOTS*, 2001.

[8] ODP - Open Directory Project. [Online]. Available: http://www.dmoz.org/

[9] Alexa the Web Information Company. [Online]. Available: http://www.alexa.com/

[10] M. Gritter and D. R. Cheriton, "An architecture for content routing support in the internet," in *Proc. of USENIX Symposium on Internet Technologies and Systems*, 2001.

[11] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A layered naming architecture for the internet," in *Proc. of SIGCOMM*, 2004.

[12] C. Esteve, F. L. Verdi, and M. F. Magalhaes, "Towards a new generation of information-oriented internetworking architectures," in *Proc. of ACM CoNEXT*, 2008.

[13] M. Walfish, H. Balakrishnan, and S. Shenker, "Untangling the web from dns," in *Proc. of USENIX NSDI*, 2004.

[14] A. Carzaniga and A. L. Wolf, "Content-based networking: A new communication infrastructure," *IMWS LNCS*, 2002.

[15] R. Chand and P. Felber, "A scalable protocol for content-based routing in overlay networks," in *Proc. of IEEE NCA*, 2003.