Name-based Routing with On-Path Name Lookup in Information-Centric Network

Yu Guan*†, Lemei Huang*, Xinggong Zhang*‡[1] , Zongming Guo*‡
*Institute of Computer Science & Technology, Peking University, Beijing, China
†PKU-UCLA Joint Research Institute in Science and Engineering
‡Cooperative Medianet Innovation Center, Shanghai, China.
{shanxigy, milliele, zhangxg, guozongming}@pku.edu.cn

***Abstract*—Name-based routing is one of the core ideas in Information-centric network (ICN). In name-based routing, there is a tradeoff between the cost of name announcement and name lookup. Some ICN architectures introduce an efficient way of name lookup but pay high price in name announcement, others cut off most information exchange in name announcement yet introduce heavy burden in name lookup.**

In order to solve this problem and balance the cost of name announcement and lookup, we propose Name-based routing with On-Path Name Lookup (OPNL). OPNL looks up name prefixes on the path to name's guaranteed destination. It accomplishes distributed name lookup with lighter burden while maintaining little information exchange in name announcement. Results of simulation experiments show that OPNL makes a tradeoff between the cost of name announcement and lookup to have better scalability, eliminates storage overhead and communication overhead compared with prior works and attains even better performance.

I. Introduction

Information-Centric Network (ICN) is a brand new architecture proposed to replace the legacy IP-based network to meet the demand for efficient dissemination of myriad and dynamic network content in large-scale network. Content (occasionally termed as name prefix in ICN papers, we also use name prefix and content name interchangeably) is the focus in ICN rather than its location. Studies has shown that ICN has some advantages over IP-based network, such as in-network caching, data packet based security scheme, and convenient application development [1].

However, when applied to large-scale network, ICN faces several challenges. The most serious problem is that with a huge name space of content items how to do name-based routing. In IP-based network, IP addresses are finite and they are easy to be aggregated. But in a large-scale ICN, the number of different name prefixes could reach 10^{12} or even more [2]. Moreover, usually they have no explicit hierarchical structure to be aggregated. This infinite and aggregation-unfriendly name space in ICN directly leads to the increase of name announcements and routing table entries, etc. , which is detrimental to the feasibility of name-based routing in large-scale network.

This work was supported by National Natural Science Foundation of China under contract No. 61471009 and Beijing Culture Development Funding under Grant No.2016-288

[1]Corresponding author.

In a large-scale network, a name-based routing solution should have the following characteristics:

1) **Low communication overhead**. Each router in the network receives a huge amount of requests continuously. The router must process every request quickly enough and reduce message exchanged between different nodes.
2) **Storage efficiency**. In a large network, the namespace of contents becomes huge. If we simply do name-based routing by recording all name prefixes in each router's routing table, the routing table will be too large to store and efficiently look up. So more storage-efficient methods are needed.

Although name-based routing process may contain several modules, two basic modules are necessary: (1) **Name Announcement**. Content provider needs to announce the existence of the provided content and make the content available in the network, in order to let others know the content, which is called name announcement. (2) **Name Lookup**. When a content request for certain name prefix arrives at a router, the router ought to look up the given name prefix in routing tables and figure out the appropriate next hop to get the content, which is called name lookup.

According to implementation of these two modules, all previous works of name-based routing can be divided into 2 classes: **Distributed Routing** and **Centralized Routing**. (1) In distributed routing method, all routers do name lookup on their own. To achieve this goal, it is necessary to do name announcement by broadcasting name prefixes as routing information to all routers in information exchange, e.g. NLSR [3], LSCR [4]. The main drawback of these methods is that broadcasting all name prefixes causes huge communication and storage overhead. (2) In centralized routing method, they set up a global name lookup system (like DNS in current IP network), so that every node looks up the name from this system, e.g. SNAMP [5]. This method is very efficient in routing information exchange (because a provider only do name announcement to the centralized system, rather than all other nodes in the network), but it causes huge overload of centralized server since name space of content item is much greater than that of IP address. Moreover, when this centralized server is attacked, it will cause many routers unable to forward requests correctly.

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

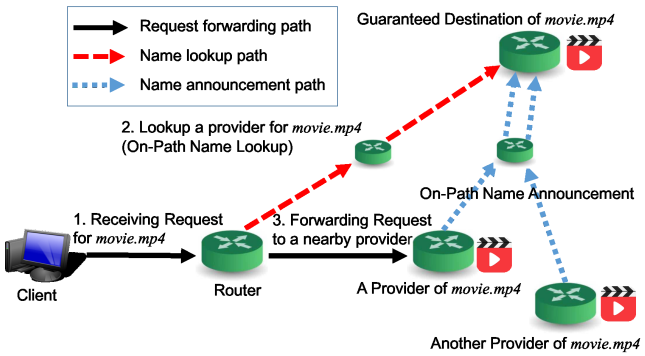


Fig. 1: Brief design of OPNL. When receiving a request, router first look up the requested content name by On-Path Name Lookup to find an appropriate provider, then forward the request to the next hop to this provider.

Since both distributed routing and centralized routing have obvious drawbacks, they have difficulty being applied to large-scale network when facing huge name space. So we need a solution which can (1) eliminate communication and storage overhead compared with distributed routing. (2) avoid dependence on a global lookup system like centralized routing.

To solve this problem, we present **Name-based Routing with On-Path Name Lookup (OPNL)**. In OPNL, router do name announcement and lookup towards content name's guaranteed destination [6] (the first node the content is published on) hop by hop. When a router becomes a new provider of this content, it announces routing information of its name prefix on the path to its guaranteed destination rather than broadcasting to the whole network. Also, when a router receives a request for a content item, it looks up routing information of this name prefix on the path to the guaranteed destination instead of looking up at a center server. Thus, in OPNL, there is no such drawback as costly broadcasting in Distributed Routing or the overburdened and vulnerable centralized server in Centralized Routing, which makes it more feasible in large-scale network. Our design of OPNL is briefly shown in Fig. 1.

The main contribution of our paper is summarized as:

1) We present a novel view of name-based routing and state that every name-based routing approach has two basic modules: name announcement and name lookup. And we show that there is a tradeoff between the cost of name announcement and name lookup.
2) Name-based Routing with On-Path Name Lookup is proposed in this paper. It eliminates communication and storage overhead dramatically compared to local name lookup methods. In the meanwhile, OPNL shares the overhead of name lookup among all name publishers in the network, rather than a single centralized system like centralized routing.

This paper is organized as follows: Section II reviews prior works on name-based routing for ICNs. Section III presents our design of OPNL (*Name-based Routing with On-Path Name Lookup*). Section IV presents simulation experiments implemented to compare OPNL with other approaches in communication overhead and storage overhead. Finally, we conclude this paper in Section V.

II. Related work

Many name-based routing approaches have been proposed. In this section we classify these methods into 2 classes, and analyze their strengths and weaknesses.

The first class is **Distributed Routing**. Its feature is "High-cost name announcement, and low-cost name lookup". Each time a router becomes a provider of a name prefix, it announces this name prefix to all the other routers in the network by broadcasting. As a result, each router knows all name prefixes and their providers, so when a router receive a request for a name, it can look up this name and independently decide where to forward this request. NLSR (Named data Link State Routing) [3] is one pioneering work. It broadcasts information about all providers of all name prefixes to all nodes in the network using link-state protocol. LSCR (Link-State Content Routing) [4] and CCN-RAMP [7] could be regarded as advanced versions of NLSR. Each router only records the nearest provider of each name prefix rather than all providers. Providers of same name prefix can exchange message by a tree-like structure called MIDST.

The main drawback of distributed routing is huge overhead of communication and storage while exchanging routing information among routers. Since in large-scale network, the amount of different name prefixes could reach 10^{12} or even more [2], it is impossible to broadcast all of them in order to enable all routers to do name lookup locally.

The second class is **Centralized Routing**. Its feature is "Low-cost name announcement, and high-cost name lookup". This class can be considered as the opposite of the first class. A centralized system is needed to do name lookup for all clients (which is similar to DNS in current IP network). It records all name prefixes and all their providers. So router does not need to announce name prefix to other routers. When a router receives a request for a content item, it sends a message to this centralized system and looks up the routing information of its providers and then forward the request. SNAMP (Secure namespace mapping) [5] is an approach which applies this method. Only some name prefixes of most popularity are recorded locally in each router. For other name prefixes, it sets up a global DNS to do name lookup, which avoids huge communication overhead caused by name announcement.

These methods also have drawbacks. Current DNS works well because domain name is a hierarchical name, but content name is a flat name. So the number of name prefixes is much larger than that of current domain names, which brings heavy burden on DNS. So a centralized name lookup system is not a good solution in ICN as in current IP network.

III. Name-based routing with On-Path Name Lookup

As mentioned above, traditional designs of name-based routing have difficulty dealing with huge namespace in large-scale network. In order to make both name announcement

and name lookup low-cost, we present *Name-based Routing with On-Path Name Lookup* (OPNL). OPNL neither announce name prefix by broadcasting nor announce name prefix towards a single centralized system. Instead, information about name prefixes are sent directionally to the name's unique guaranteed destination. Moreover, name lookup is also along the path to each name's guaranteed destination. We want to describe OPNL in the following three dimensions:

1. Basic Design: The first dimension is our design of data structures, including tables in routers and content request. After that, we briefly introduce the process of request forwarding.

2. On-Path Name Announcement: The second dimension is how to announce routing information of name prefixes. In OPNL, when a router becomes a provider of a name prefix, the corresponding routing information is announced only on the path from the provider to the name's guaranteed destination.

3. On-Path Name Lookup: The third dimension is how to do name lookup when local router does not have routing information for the requested name prefix. We present On-Path Name Lookup strategy which can look up the name in remote routers on the path to name's guaranteed destination.

A. Basic Design

In OPNL, when content items under a name prefix is locally available on a router, we call this router a provider of the name prefix. Especially, if this name prefix is first published, this router is called **guaranteed destination** of the name prefix. One name prefix could have several providers, but its guaranteed destination is unique.

In OPNL, the data structure of tables in router are similar to that in NDN [8], [9], and we also adopt the design of *Interest* in NDN to refer to content request, with slight modification to be better adapted in OPNL.

1) Design of Tables in Routers: To carry out the Interest and Data packet forwarding functions, each router maintains four tables: Content Store (CS), Pending Interest Table (PIT), Prefix Resolution Table (PRT) and Topology Information Base (TIB). Among them, CS and PIT do the same work as in NDN. Differently, PRT stores the information related to name prefixes and TIB stores the topology information of network similar to the routing table in legacy internet architecture. PRT and TIB work together to perform the same function as FIB in NDN. As the one that concerns name prefix of them, PRT plays an important role in name announcement and name lookup: information of name announcement is recorded in it and name lookup relies on it.

Since the topic of this paper is how to do name-based routing, we are going to describe PRT and TIB detailedly. Each router R in the network has a PRT which maps name prefixes to their providers respectively. In router R, given a name prefix i, its PRT entry PRT_R^i lists some or all providers of i. Fig. 2 shows an example of PRT. TIB is a table which maps a router to the next hop to it. It is used for figuring out the next hop to a given router. TIB can be generated and maintained by traditional Link-State protocol. So in OPNL, a name prefix is first mapped to a provider through PRT, and then mapped to the next hop through TIB.

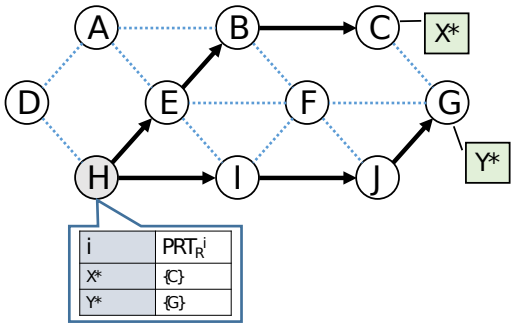


Fig. 2: An example of PRT. In this example, X^* and Y^* are two name prefixes which are published in node C and G. We show node H's PRT in the figure. Solid arrows indicate the route to X^* and Y^* from node H.

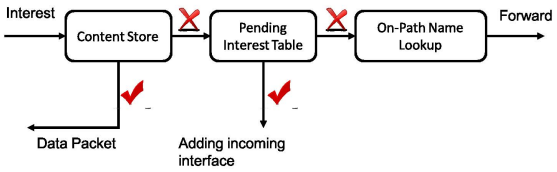


Fig. 3: Name-based routing at a router.

Field	Definition
I.name	Name Prefix.
I.direction	The designated provider of I.name. Will be set up during first name lookup.

TABLE I: Interest Packet

2) Design of Interest Packet: In OPNL, we use Interest Packet to represent a request for a content item which is similar to Interest Packet in NDN. Its data structure is shown in Table I (We omit fields which are irrelevant to our topic). An Interest consists of 2 main fields: name prefix (I.name) and direction(I.direction). Compared to traditional NDN Interest Packet, the biggest difference is that we add a direction field. I.direction records an intermediate result of name lookup. When a router figure out the provider of requested name prefix during name lookup, it embeds the result in I.direction field. Once I.direction is set up, other routers on the path the Interest packet traverses need only to choose appropriate next hop to this provider according to TIB. This reduces computational overhead in name lookup dramatically.

The process of name-based routing is shown in Fig. 3. When an Interest packet arrives, router first checks CS for matching data; if it does not exist, the router looks up the name prefix in its PIT the same as in NDN. If it misses, the proposed On-Path Name Lookup (section III.C) is applied to figure out the appropriate next hop.

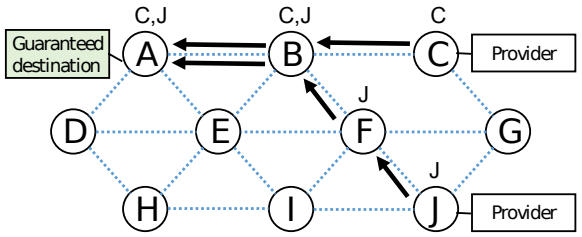


Fig. 4: An example of On-Path Name Announcement. In this example, A is the guaranteed destination of the name prefix. C and J are two new providers. They send NAs on the path to A. (Solid arrow indicates the forwarding path of NA) There are 5 NAs generated and propagated while it is at least 18 in broadcasting scenario. Only {A, B, C, F, J} add one PRT entry and others keep their PRT unchanged.

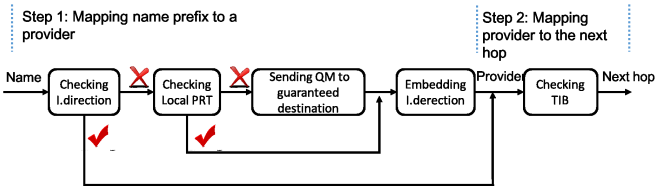


Fig. 5: Process of On-Path Name Lookup.

B. On-Path Name Announcement

When a router becomes a first provider of a content, it will be the unique guaranteed destination of this content. After that, every time a router becomes a new provider of the content, it announces the existence of it by sending Name Advertisement (NA) towards the guaranteed destination. So for each content i, i's guaranteed destination always has i's all providers recorded in it.

Name Advertisement (NA) is to inform information about name prefix and its provider. In OPNL, NA is sent only on the path to name's guaranteed destination through hop-by-hop forwarding instead of broadcasting. Only routers on the traverse path record the information of NA. Compared to name broadcasting, on-path announcement paradigm reduces the communication and storage overhead as less NAs are propagated and routers not on traverse path will not add their PRT size without NAs arriving. Fig. 4 shows an example.

C. On-Path Name Lookup

As shown in Fig. 5, On-path Name lookup strategy consists of two steps: (1) Look up the provider of given name prefix. (2) Figure out the next hop to this content provider.

Since the second step can be done easily through local TIB, we focus mainly on the first step. Given an Interest for a name prefix, router checks if I.direction is empty. If I.direction is not empty, the provider of this name is known already known, so router does not need to go on this step. If I.direction is empty, router looks up requested name prefix in its PRT. If the name prefix hits local PRT, router chooses a proper provider and embeds it into I.direction. If it misses, router generates a Query Message (QM) and sends it on the path to name's guaranteed destination to remotely figure out the provider. Then router waits for response of this Query Message. After receiving response, router writes it into I.direction.

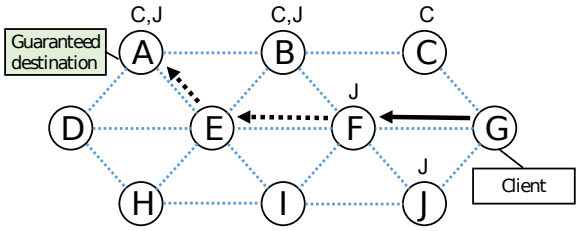
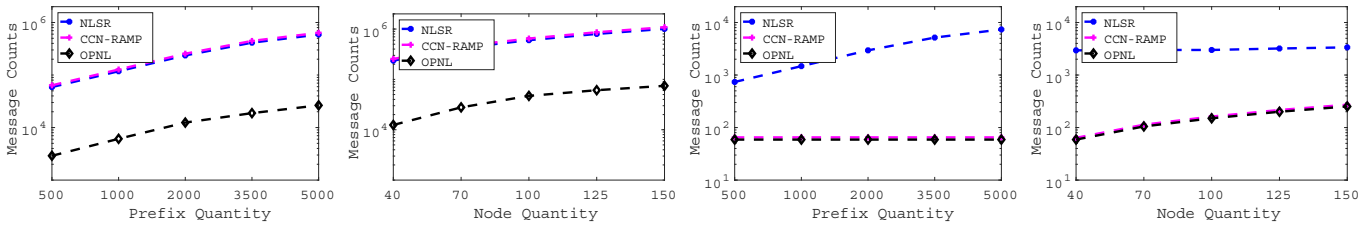


Fig. 6: An example of On-Path Name Lookup. A is the guaranteed destination of the name prefix. G receives an Interest and send Query Message towards A. Query Message stops at F because F has local PRT entry of the name prefix. F sends back the response the message to G and states that J is a provider of the name prefix. (Solid arrow indicates the forwarding path of Query Message)

In this strategy, two main topics are discussed: (1) How does a router know the guaranteed destination of a name prefix? (2) How to do on-path name lookup by sending Query Message?

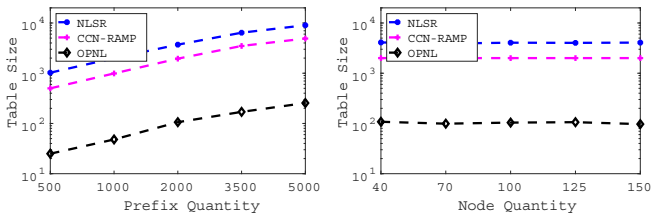
1) How does a router know the guaranteed destination of a name prefix: Once a name prefix is first published on a router, this router becomes this name prefix's guaranteed destination forever, and its name prefix is completely determined. This router creates a new entry for this name prefix in its PRT. Name prefix consists of two parts: (1) **Content Identifier**. (2) **Guaranteed Destination**. Content Identifier is used for distinguishing the name prefix from other name prefixes. Under a guaranteed destination, two different name prefixes must not share a same Content Identifier. So it is forbidden to publish a new content which Content Identifier already exists in the guaranteed destination. Of course one name prefix is likely to be available on multiple providers, but its guaranteed destination is never changed. So we add the Guaranteed Destination into the name prefix. Given a name prefix, a router can figure out its guaranteed destination directly.

2) How to do on-path name lookup by sending Query Message: In OPNL, the Query Message is sent from current router to the guaranteed destination of the requested name prefix. After a Query Message is generated, all routers in the this path will receive this message one by one until the requested name prefix hits PRT in a router. When a router receives the Query Message and it has the corresponding PRT entry locally, it stops forwarding the Query Message, chooses a provider which is the nearest to the Query Message generator and then sends the response back to the Query Message generator. It is easy to prove that On-Path Name lookup will not fail because in the worst case, Query Message will finally arrive at guaranteed destination. Fig. 6 shows an example.



(a) Prefix quantity (b) Node quantity (c) Prefix quantity (d) Node quantity

Fig. 7: Communication overhead in NLSR, CCN-RAMP and OPNL. (a) and (b) illustrate the quantity of messages sent in network initiation while (c) and (d) illustrate the quantity of messages sent in a network change.



(a) (b)

Fig. 8: Average table size in NLSR, CCN-RAMP and OPNL.

IV. Performance Comparison

We implemented Name-based Routing with On-Path Name Lookup (OPNL) based on our algorithm using C++ and Matlab. Then we load our calculation on ndnSIM [10] as a forwarding strategy to evaluate the performance. We compare the proposed OPNL to NLSR and CCN-RAMP. NLSR is a classical name-based routing method based on NDN. CCN-RAMP adopts indirect name lookup (mapping name prefix to provider, then mapping provider to next hop) as we do but announces name prefixes in broadcast.

In this section, we use real request traces to evaluate the relative performance of different architectures. These traces are collected from one-day HTTP requests in Beijing, China, including 22000 different hosts and 10^9 requests.

Three key metrics are considered: (1) the number of messages in name announcement; (2) the average sizes of tables. Link state or DCR [11] protocol is used in the control plane to update information in NLSR and CCN-RAMP. Accordingly, we do not need to consider the signaling overhead, because it is exactly the same in the three approaches we consider.

Since we focus only on name announcement and lookup in this paper, we apply the same caching strategy and PIT to obtain a result of high reliability. We use the GEANT network topology, which is considered to be a realistic topology, for simulations [12]. This topology includes 40 nodes and 59 point-to-point links with 10 ms delay. We also apply some scale-free topologies generated by [13] in order to evaluate the performance in different networks. All nodes have a consumer application simulating local consumers. Different consumers have different frequencies of generating Interests.

A. Communication Overhead

Since OPNL announces a name prefix only on one path instead of by broadcasting, communication overhead is one of the core advantages over other methods.

We evaluate the communication overhead (i.e. the number of exchanged messages) in two scenarios: (1) network initialization: at first all routers in the network possess empty tables. Only content providers know the name prefixes they locally provided. Routers in the network build up their tables according exchanged information (2) network change: A network change (e.g. node failure or link failure) necessitates relevant routers update their tables according to exchanged information.

In our experiment, we simulate the process of information exchange and count the number of messages using C++ and Matlab. Since the quantity of messages is influenced by quantity of name prefixes and nodes in the network, we apply them as two parameters in our experiment.

We present a sensitivity analysis across 2 different configuration parameters: prefix quantity and node quantity. First we give a default setting of these 2 parameters. In our default setting, prefix quantity is 2000, node quantity is 40. Our experiment consists of 2 groups. In each group, we change one of the configuration parameter and keep another unchanged.

Prefix quantity: Number of different name prefixes reflects the diversity of the network. We set the prefix quantity to 500, 1000, 2000 3500 and 5000 in our experiment.

Node quantity: Number of routers reflects the network scale. We show the result when the network scale is 40, 70, 100, 125 and 150.

1) Communication overhead in network initialization: Traditional link-state (LS) or DCR are used to update information in NLSR and CCN-RAMP, while the proposed on-path name announcement is applied in OPNL. Their performance is shown in Fig.7a and Fig.7b. It is obvious that on-path name announcement has great advantages.

Key Observations and Implications:

1. We apply Link-State (LS) method to update information in NLSR and CCN-RAMP. In this method, communication overhead is proportional to prefix quantity and node quantity. (In our experiment, we choose different network topologies

with similar density). We denote CO_{LS} to the communication overhead of LS, N to node quantity and M to prefix quantity, applying a curve fitting according to our experimental data as below:

$$CO_{LS} = k_{LS}NM \tag{1}$$

where k_{LS} is a constant. The value of k_{LS} is around 2.00.

2. In OPNL, the communication overhead also grows with prefix quantity and node quantity. We also apply a curve fitting as equation below:

$$CO_{OPNL} = k_{OPNL}N^{1.259}M^{0.8736} \tag{2}$$

where k_{OPNL} is a constant. The value of k_{OPNL} is around 0.18 which is much smaller than k_{LS}. Furthermore, CO_{OPNL} increases slower than CO_{LS} when network scale increases (M becomes larger).

2) Communication overhead when network change: A main drawback of NLSR is that routers have no concept of nodes and links in the network. So it has difficulty updating FIB when network changes. Although this issue has not been discussed in previous works, it is a real problem and this often happens. In this experiment, we simulate a node failure and evaluate the communication overhead of information updating.

Quantity of name prefixes and nodes in the network are also provided as to parameters. Experimental result is shown in Fig.7c and Fig.7d.

Key Observations and Implications:

1. In NLSR, when a link failure happens, all routes pass this link become invalid. So routers need to recalculate the distance to all name prefixes which is influenced by this broken link. Many FIB entries need to be updated. This recalculation causes unacceptable communication overhead. When the number of name prefixes increase, this issue becomes more serious.

2. CCN-RAMP and the proposed OPNL performs much better because link failure only causes topology information update. It does not influence the process of name lookup. So communication overhead for notifying a broken node depends on the scale of network rather than number of different name prefixes.

B. Average Table Size

Since OPNL only announces a name prefix along one path, routers which are not on this path do not need to store this name prefix. So network storage overhead is reduced dramatically. This is OPNL's another core advantage over other methods. Fig. 8 shows the average table sizes for NLSR, CCN- RAMP, and OPNL on a logarithmic scale.

Key Observations and Implications:

1. According to Fig. 8(a), the table size of OPNL is nearly proportional to quantity of prefixes.

2. Node quantity does not deeply influence the performance of OPNL since the diameter of network topology increases much slower than node quantity increases. So OPNL's performance will not become poorer with the increase of network scale. This is an important reason why OPNL can adapt to large-scale networks.

3. Traditional NLSR uses FIB for name lookup. PRT is used for name resolution in CCN-RAMP and OPNL. Thus, both FIB and PRT store information of name prefixes. CCN-RAMP presents a Forwarding to Anchor Base (FAB) and its function is the same as TIB in proposed OPNL. Experimental results show that storage overhead depends on the scale of name prefixes. Compared to NLSR and CCN-RAMP, OPNL reduces the storage overhead dramatically.

V. Conclusion

In this paper, we summarize the previous name-based routing architectures and state that either Distributed Routing or Centralized Routing has communication or storage problem in large-scale network.

In order to solve these problems of name-based routing, we present OPNL, the name-based routing scheme with On-Path Name Lookup. In OPNL, we propose an On-Path Name Announcement strategy which announces information of name prefix only on a path, and an On-Path Name Lookup method which looks up name prefix remotely in absence of local PRT entry. We prove the correctness of our method and its efficiency is shown in our experiments.

References

[1] W. Shang, L. Zhang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, and B. Zhang, "Named data networking of things (invited paper)," in *IEEE First International Conference on Internet-Of-Things Design and Implementation*, 2016, pp. 117–128.

[2] T. Song, H. Yuan, P. Crowley, and B. Zhang, "Scalable name-based packet forwarding: From millions to billions," in *The International Conference*, 2015, pp. 19–28.

[3] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "Nlsr: named-data link state routing protocol," in *ACM SIGCOMM Workshop on Information-Centric NETWORKING*, 2013, pp. 15–20.

[4] E. Hemmati and J. J. Garcia-Luna-Aceves, "A new approach to name-based link-state routing for information-centric networks," in *The International Conference*, 2015, pp. 29–38.

[5] A. Afanasyev, C. Yi, L. Wang, and B. Zhang, "Snamp: Secure namespace mapping to scale ndn forwarding," in *Computer Communications Workshops*, 2015.

[6] T. Koponen, M. Chawla, B. G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," *Acm Sigcomm Computer Communication Review*, vol. 37, no. 4, pp. 181–192, 2007.

[7] J. J. Garcia-Luna-Aceves, M. Mirzazad-Barijough, and E. Hemmati, "Content-centric networking at internet scale through the integration of name resolution and routing," in *Conference*, 2016.

[8] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *Acm Sigcomm Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[9] "Ndn project, ndn packet format specification." [Online]. Available: http://named-data.net/doc/ndn-tlv/

[10] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnsim: ndn simulator for ns-3," 2012.

[11] J. J. Garcia-Luna-Aceves, "Name-based content routing in information centric networks using distance information," in *The International Conference*, 2014, pp. 7–16.

[12] "Geant project website." [Online]. Available: http://www.geant.net/

[13] [Online]., "Available:http://wwwdi.supelec.fr/software-orig/ashiip/."