

A Flat Name Based Routing Scheme for Information-Centric Networking

Jungha Hong*, Woojik Chun**, Heeyoung Jung*

* ID Comm. Research Lab, ETRI, Daejeon, Republic of Korea

** Hankuk University of Foreign Studies, 81, Oedae-ro, Mohyeon-myeon,
Cheoin-gu, Yongin-si, Gyeonggi-do, Korea

jhong@etri.re.kr, woojikchun@gmail.com, hyjung@etri.re.kr

Abstract— The paradigm shift from the current host-centric networking to information-centric networking (ICN) generates much interest in the research community. In ICN, the primary communication entity is contents and uniquely identifiable names are assigned directly to the contents, where the names are independent of their locations. Thus, content naming and routing are the key challenges in ICN. In this paper, we propose a flat name based routing scheme for ICN. For scalable routing, we exploit the structure of container which is a space where contents or information objects reside. We also utilize bloom filters as an aggregated form of the flat names and enable the deepest match routing.

Keywords— Internet Architecture, Information Centric Networking, Container, Flat Name Based Routing

I. INTRODUCTION

The current Internet is founded on a host-centric networking: hosts are uniquely identified with IP addresses and communication is possible between any pair of hosts. Thus, contents in the current Internet are identified by the name of hosts where the contents are stored. In contrast to the host-centric networking, the primary communication object in information-centric networking (ICN) is contents, where uniquely identifiable and location independent names are assigned directly to contents.

The shift from host-centric to information-centric requires fundamental changes in communication model, which includes separation of name and location, dissemination and in-network caching of information, discovery of information, and delivery of information. Over last decade, several proposals for ICN architecture have been published [1-2]. Among several challenges in ICN research community, content naming and name based routing are the key challenges and there are several projects on them [3-10].

Names assigned directly to contents are categorized into two parts: hierarchical names and flat names. One of the key issues with flat names in ICN is scalability, where names are uniquely identifiable. A. Ghodsi et al. [10] claimed that self-certifying names are the correct choice based on the architectural goals of security, scalability, and flexibility. They constructed explicit aggregation which is more flexible than inherent aggregation by using concatenations of globally

unique names. The concatenated names were considered as fetch-terms and included in metadata associated with the name. The semantics of such concatenations were that the object would be eventually found when following routing entries for each concatenated names, which they called the aggregation invariant. They used the fetch-terms to retrieve the object more easily and the deepest match for efficient routing. However, the binding of names and fetch-terms in a content request message may yield some issues when contents are relocated or cached. It also reduces flexibility compared to their separation.

Thus, in this paper, we propose a purely flat name based routing scheme for ICN with no use of such fetch-terms or no concept of locators. However, we keep the aggregation invariant by using bloom filter. We utilize the concept of a container which is a space where contents or information objects reside [9]. We construct the containers hierarchically for scalable routing instead of constructing a hierarchy in naming. We also utilize bloom filters for efficient lookup process at each container. Our explicit aggregation takes place in the use of bloom filter.

In general, projects on naming and routing in ICN start with proposing naming schemes and then present correspondent routing schemes to their own naming schemes. However, it is remarkable that there is no need of a specific scheme for flat naming in this paper. In other words, our flat name based routing scheme for ICN works for any schemes of flat naming.

The remainder of this paper is organized as follows. We separately describe our communication model and key operations for the proposed flat name based routing scheme for ICN in Section 2 and Section 3, respectively. The contribution of this paper along with our future work is described in Section 4.

II. A COMMUNICATION MODEL

In this section, we describe our communication model. We assume that a requestor who sends an interest packet to retrieve contents is outside of an ICN network. We also assume that containers are well structured hierarchically for scalable routing and the structure retains the loop-free property for using bloom filter.

A. Containers

A container is a space where content or information objects reside [9]. We can imagine lots of diverse forms of containers such as file systems, databases, ID center, a server or server farms. The abstract concept can be easily extended to realistic networks such as local, regional or provider networks. Similarly, a single file can be viewed as a container that holds much other information linked via hyperlinks.

In host-centric networking, each named host takes the role of communication endpoint, whereas in ICN, the named content itself cannot participate in communication. In such a sense, we may call the named content as a “passive entity” in contrast to host as an “active entity”. Thus, every passive entity must be associated with an active component to participate in communication.

Contents must reside in at least one container to be fetched by requestors and the container must provide access function to requestors. Thus, we define a “controller” as an entry point of containers and it operates only on the control plane. In the networking view, the controller takes the role of communication endpoint on the behalf of the contents within the container. The controller must have proper mechanism to fetch all contents within its container.

In addition to normal fetching function the controller may performs routing or security functions. For example, the controller may perform the authentication of the request and verify authorization in the routing phase. It may also perform cryptographic functions or digital signatures in the delivery phase.

In order to fetch contents, interest packets by requestor must be forwarded into the container where the interested contents reside. Since it is assumed that requestor who tries to retrieve contents is outside of ICN, there must be an access point so interest packets can first access to ICN network. We define it as “access controller” and assume that every requestor knows one or more access controllers where it can send interest packets to.

Figure 1 depicts the communication model where the structure of containers is omitted. It will be described in the following subsection.

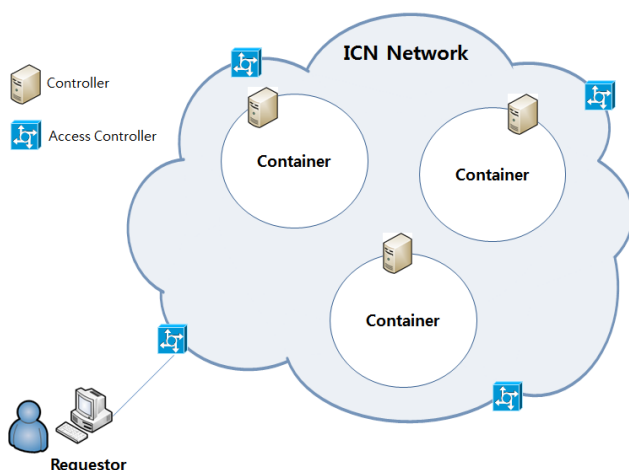
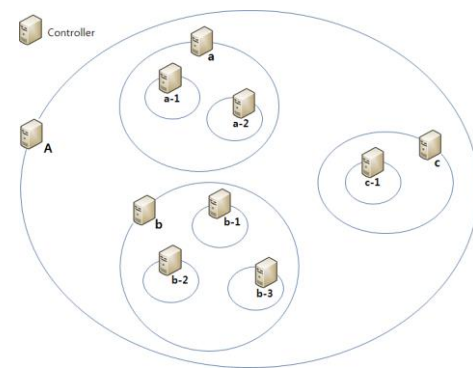
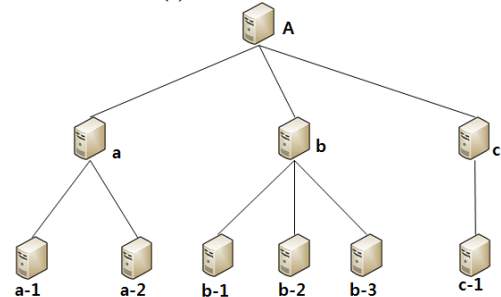


Figure 1. Communication model



(a) Container structure



(b) Tree structure of controllers

Figure 2. Structures of container and controller

B. Recursive Structure of Containers

In this paper, a container may include nested containers and may be involved in a larger container. Thus, containers can be structured in a recursive manner such as novel/literary novel/short literary novel/short literary novel written in English/etc. Of course, a piece of content as well as a container may be involved in more than one container simultaneously, which is exactly same as multi-homing in host-centric networking. We first allow operators to structure the containers in any recursive manner so they can utilize it for their better management. Then, we let contents reside in at least one container which is well structured by operators.

When requestor wants to retrieve contents, the interest packets have to be forwarded to the containers exactly where the contents reside. Thus, we build a forest of controllers based on the recursive structure of containers for interest packet routing. We utilize bloom filters as an aggregated form of names at each controller instead of announcing the whole list of names to its peers or parents for flat name based routing. The bloom filters are in the role of forwarding table. The explicit aggregation mentioned in [10] takes place by the use of bloom filter. We note that the forest of controllers retains the loop-free property for the use of bloom filter.

Figure 1 can be viewed that the ICN network is consisted of a forest by 3 disjoint trees. One of them may have a structure like in figure 2 (a) and the hierarchy in figure 2 (b) can be built based on its structure shown in figure 2 (a). The key components of containers for interacting with other containers are the access interface and the peering interface. The access interface provides container management functionalities, such

as bootstrapping, configuration, and life cycle management. The peering interface allows collaboration between the peering containers by extending their transport, control, management, and inference planes among each other. In each of the extended planes, the peers share information via some agreed interactions, often referred as protocols. Although shared by multiple containers, protocols are defined in container-specific context independent of other containers.

Basically, it is assumed that interest packets are forwarded to the roots of each tree associated with an access controller. Then, by bloom filter test, they are eventually forwarded to the containers where the interested contents reside. In such a sense, we keep the aggregation invariant defined in [10].

III. KEY OPERATIONS

In general, ICN routing may comprise three communication steps: content lookup, content discovery, and content delivery. The first phase of content retrieval, content lookup, is to find out where the information actually exists, possibly in multiple locations. Each location is identified by the specific notations defined by the given network space. For example, if content is stored in the file system in a host, the location can be given in the form of IP address of the host and path name within the file system.

Once obtaining the locations of content, requestor must express his interest on the content into the network. This content discovery phase is usually done by finding appropriate path from requestor to one of the content locations.

The last step, content delivery, is that interest content has to be delivered from the origin or one of the caches to the requestor. This phase must be able to handle dynamic status change in the network.

Since the flat name based routing in this paper do not use such a concept of locator, the communication step can be shorten by two steps: content discovery and content delivery.

Practically we can easily expect billions of names registered in a container and millions of containers in the network. In order to address the scalability issue when routing by name, we must borrow some aggregation techniques. The well-known prefix techniques, such as container name or IP address masking, cannot be used for our routing because it is assumed that the name of content is flat and has no structure to be aggregated. Thus, how to register content to containers, which has a flat name, is also one of the key operations.

A. Content Registration

As we have stated in the communication model in Section 2, operators are allowed to construct the container structure in any way they want and the named contents can be registered into one or more arbitrary containers since location independent flat name is assumed in this paper. Here, we emphasize that the named contents can reside in any containers they want regardless of the structure of containers. In other words, there is no constraint on registering the named content to containers. However, contents may practically reside in containers according to their structure for fast discovery.

Since flat name is assumed in this paper, we need to aggregate names in any shape of type to address the scalability issue. One popular technique for flat name is Distributed Hashing Table (DHT), where multiple servers (in our case containers) form circular linked list and the bindings are stored in the appropriate server. However, the DHT technique has some drawbacks; the binding must be stored in the server other than the owner's and lookup message may be propagated through the long paths.

Thus, we utilize bloom filter as an aggregated form of names. Controllers of containers announce their content name set to the other containers. Instead of announcing the whole list of names, bloom filter as an aggregated form of names is announced. When announcing its name set to its peers or parents, the controller announces the union of name sets of all child containers. Union of child name set can be built by using the characteristic of bloom filter that bloom filter for union of sets can be built merely by bitwise 'OR' operation on all the sets.

Our content registration is along with bloom filter update. When content resides in a certain container, its name has to be inserted first into its own container's bloom filter. Then the controller announces the new name information to the controllers of its parents and peers and they update their corresponding bloom filters, where this recursion holds until bloom filters at the root are completely updated. When contents are cached, their information has to be inserted into bloom filters just like when registered.

When contents are deleted from containers, we need to adopt a certain mechanism to update the bloom filters for the deletion since bloom filter can handle the deletion by itself. Thus, we use the periodic refresh technique that bloom filters with registered names are rebuilt periodically and followed by bloom filter updates.

B. Content Discovery

The primary role of the content discovery is generally to setup the path from a requestor to one of locators where the interest content is stored so interest packet can be forwarded along the path. However, the discovery phase in this paper is to route the interest packet by name to the container where the interest content reside.

Controllers are in the role of entry point of containers and logically build a forest of hierarchies based on the container structure. The relationships of parents and child are determined by the tier information. Controllers of tier-0 container are the roots of the hierarchies and ones of the highest tier number containers are the leaves. Each controller consists of bloom filters of its own, child, and peer and the bloom filters are in the role of forwarding table.

A requestor sends an interest packet and is forwarded into ICN network through an access controller. Basically, access controller forwards the interest packet to the roots of the controller tree, i.e. tier-0 controllers, which are associated with the access controller. However, access controller may associate with some intermediate controllers for better performance in the respect of latency. Access controller consists of bloom filters of controllers which the access

controller is associated with. Among controllers passed by bloom filter test, access controller forwards the interest packet first to the one which has the highest tier number. This method is conceptually identical to the deepest match in [10]. It is possible that the requestor may not retrieve the interest contents by the first discovery to the deepest controller due to the false positive error of bloom filter. In this case, the access controller tries the second deepest match for content discovery. This recursion works until the interest content is discovered.

In case that an access controller is associated with several intermediate controllers, we may need a specific forwarding policy at access controller for better performance. For example, interest packet can be forwarded to a couple of controllers according to the deepest match.

C. Content Delivery

In this paper, it is assumed that the control plane for content discovery and data plane for content delivery are separated. So, the interest content discovered in the control plane is delivered directly to the requestor through the data plane as depicted in figure 3.

Figure 4 depicts the procedures of the flow shown in figure 3, where content “C” is registered in controller b-3, bloom filter (BF) updates occur in succession, and a requestor looks for the content “C”.

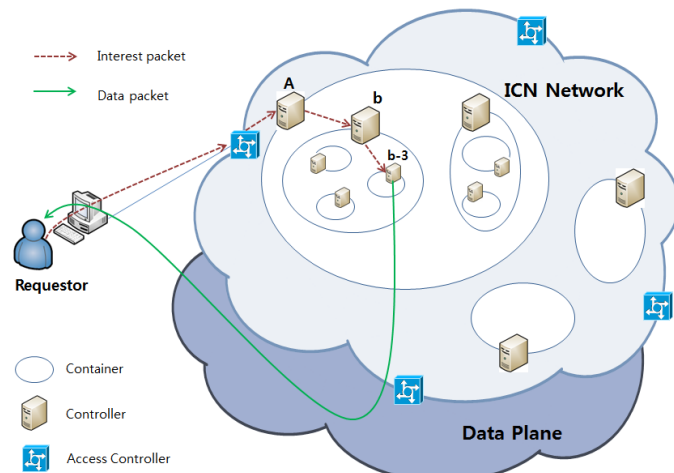


Figure 3. Flow of interest and data packets

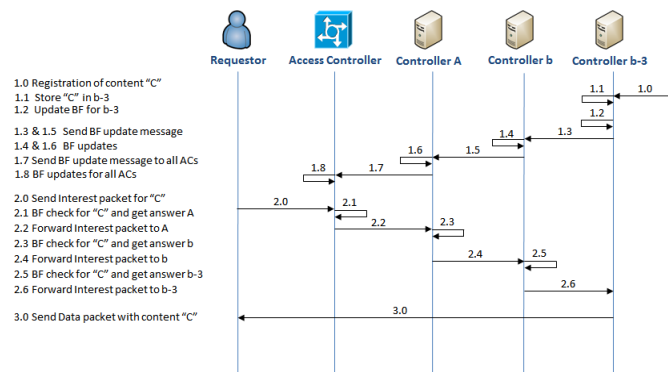


Figure 4. Procedures of key operations

IV. DISCUSSION AND FUTURE WORK

In the traditional IP network, the aggregation of the IP addresses leads to a compact forwarding table. However, since the forwarding table is implemented often by using expensive SRAMs for performance reasons, a large forwarding table is not economically viable. In contrast to IP network, a compact forwarding table cannot be created for ICN with flat name. Hence, the forwarding table entry explosion problem, i.e., scalability to the number of named object must be addressed for a flat name based routing.

In this paper, we proposed a novel flat name based routing on structured containers for ICN. We assumed that containers can be well structured by operators and flat named contents can reside in any containers. In such a way, lots of flexibility can be provided. We utilized bloom filters as an aggregated form of the names and enabled the deepest match routing.

We intend to investigate the performance of our flat name based routing for ICN by modelling the communication system in terms of the average number of controller accesses per request.

REFERENCES

- [1] B. Ahlgren et al., “A Survey of Information-Centric Networking,” *IEEE Commun. Mag.*, vol. 50, no. 7, pp.26–36, July 2012.
- [2] J. Choi et al., “A Survey on Content-Oriented Networking for Efficient Content Delivery,” *IEEE Commun. Mag.*, vol. 49, no. 3, Mar. 2011, pp. 121–27.
- [3] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, “A survey of naming and routing in information-centric networks,” *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 44–53, 2012.
- [4] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 181–192, August 2007.
- [5] C. Dannewitz, “NetInf: An Information-Centric Design for the Future Internet,” in *Proc. GI/ITG KuVS Workshop on The Future Internet* 2009.
- [6] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. Braynard, “Networking named content,” in *CoNEXT*, 2009.
- [7] A. Carzaniga, M. Rutherford, and A. Wolf, “A Routing Scheme for Content-Based Networking,” *INFOCOM 2004, 3rd Annual Joint Conf. IEEE Computer and Commun. Societies*, vol. 2, Mar. 2004, vol. 2, pp. 918–28.
- [8] D. Cheriton and M. Gritter, “TRIAD: a scalable deployable NAT-based Internet architecture,” *Technical Report*, January 2000.
- [9] C. Yao, R. Wang, Y. Xuan, and Z. Yan, “Container Assisted Naming and Routing for ICN,” *draft-ietf-icnrg-naming-routing-00*, February 17, 2013.
- [10] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and Shenker, “Naming in Content-Oriented Architectures,” In *Proceedings of the SIGCOMM ICN’11*, August 19, 2011, Toronto, Ontario, Canada.