The Impact of FQDN Database Updates on Name-based Routing Architecture

Haesung Hwang*, Shingo Ata†, Masayuki Murata*
*Graduate School of Information Science and Technology, Osaka University, Japan
Email:{h-hwang,murata}@ist.osaka-u.ac.jp
†Graduate School of Engineering, Osaka City University, Japan
Email:ata@info.eng.osaka-cu.jp

***Abstract*—Sophisticated routing protocols, such as content-based routing, have been studied and proposed to replace Internet Protocol (IP) in the future Internet. Routing based on content will make finding, querying, and retrieving the desired information possible within the routers rather than having the routers forward the packets to data servers or domain name servers. However, one of the main reasons that has prevented achieving content-based routing within the routers in the past has been their storage constraints. In our previous work, we showed that name-based routing, which is a good example of content-based routing, is feasible in practice utilizing Ternary Content Addressable Memory (TCAM). Using our proposed algorithms based on longest alphabet match and Distributed Hash Table (DHT), the Fully Qualified Domain Name (FQDN) database was distributed among less than the number of routers that are currently existing in the Internet. In this paper, we evaluate the feasibility of robustly updating the routing information database entries and show that there are no drastic changes in the routers' forwarding tables. We propose an algorithm for mapping a logical network topology to the physical topology to achieve robustness against dynamically changing databases that also considers the characteristic of TCAMs.**

I. INTRODUCTION

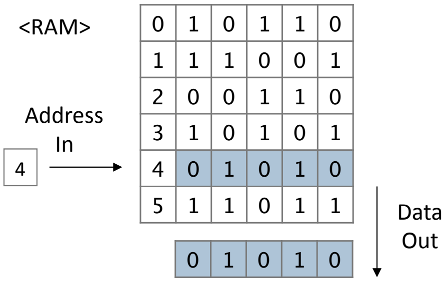
Content-based routing is one of the most anticipated technologies in the future Internet, where many next-generation network research programs such as FIND [1] and FP7 [2] and projects such as AKARI [3] aim at designing a post-IP (Internet Protocol) network architecture. Being able to route with resources and contents such as application names, port numbers, and types of information rather than the aged IP will solve the mismatch between the traffic carried at the time IP was designed and the current traffic trends toward high user mobility and ubiquitous network access.

A typical example of this content-based routing is name-based routing, where names roughly represent the content of the information. One of the easily available databases of 'names' is *Fully Qualified Domain Name* (FQDN). FQDN is used throughout our research as a node ID and the entries constructing forwarding/routing tables in routers. However, one of the problems of using such IDs is that by having variable lengths, typically longer than IP addresses, they require additional hardware space and search overhead. In addition, the scalability issue arises when FQDNs are frequently updated, causing the associated IP addresses in the domain name servers and routing tables to be updated as well.

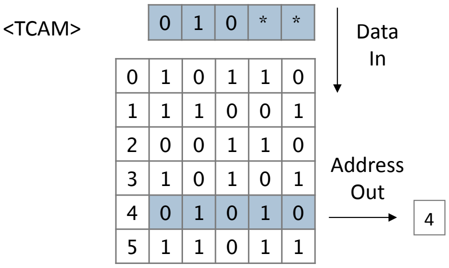
A state-of-the-art memory used for forwarding/routing tables in routers is *Ternary Content Addressable Memory* (TCAM) [4] which is known for its high-speed and its ability to search contents. The difference between conventional Random Access Memory (RAM) and TCAM is shown in Fig. 1. RAM is searched using a memory address as a search key and the content of the memory at that address is returned as the result. On the other hand, TCAM is searched using the memory content and the memory address where the supplied data was found is returned as the result. In the example shown in Fig. 1, RAM takes '`4`' as the input and returns '`01010`' as the result, whereas TCAM takes '`010**`' as the input and returns '4' as the output. Unlike conventional memory, which can represent only binary digits `0` and `1`, TCAM can also represent a third value in each cell: an arbitrary don't care value '$\star$'. This don't care value is a big advantage for TCAM as it can perform partial matching of the entries excluding the $\star$ part, enabling high-speed longest prefix matching.

One of the concerns on TCAM in the past has often been its high power consumption. However, TCAM's memory efficiency has been improved [5], i.e. more transistors can be placed per unit cell space, which shows a vision for solving the additional hardware problem mentioned above.

In our previous work [6], we showed the feasibility of name-based routing by evaluating the required number of routers, where TCAM was used as the memory in the routers. Although we were able to show that the number of required routers is two orders of magnitude smaller than the number of currently available routers deployed in the Internet, the previous work was only preliminary for evaluating the effect of dynamic updates of the FQDN database entries. In this work, we show that the proposed distribution algorithms result in no drastic changes in the routers' forwarding tables against dynamic updates of FQDN databases. In addition, in order to solve the scalability problem mentioned above, we implement a hierarchical architecture which is traditionally known to accommodate growth and to isolate faults [7]. More specifically, we implement a hierarchical *Distributed Hash Table* (DHT), an architecture adequate for storing hierarchically structured FQDN. Another reason that we use DHT is that it is suitable for storing and searching contents. Routers acting as DHT nodes will have far more processing ability than the regular end nodes forwarding packets in Peer-to-peer type of networks.



(a) RAM



(b) TCAM

Fig. 1. Difference between RAM and TCAM

Fig. 2 shows our conceptual logical topology, where routers in the same hierarchy level are grouped in a ring.

Hierarchical DHT designs are known to have better fault isolation and security, effective caching and bandwidth utilization, and hierarchical access control [8]–[10]. However, the mismatch problem in node neighborhood between the overlay and the physical network is a major obstacle in building an effective large-scale overlay network [11], [12]. In [12], Peer-exchange Routing Optimizing Protocols (PROP) are proposed which adaptively adjust the connections of the overlay and efficiently reduce the average logical link latency of the whole system. They are adaptive to dynamic changes in the system and can be deployed on both unstructured and structured P2P systems. The work in [13] uses locality sensitive hash functions instead of a consistent hash function to solve the mismatch problem. The proposed algorithm applies the locality hash function to a node's physical position, where the coordinates are computed from modeling the Internet as a geometric space, and the hashed value is used as the node's ID in the ring. In [14], landmark clustering and *Round-Trip-Time* (RTT) measurements are combined to generate proximity information. Landmark clustering is used to guide the placement of proximity information such that information about nodes that are physically close to each other are stored close to each other on the overlay. A node uses its landmark number, which reflects its physical position in the network, as the DHT key to access relevant proximity information.

Inspired by the location-aware mapping methods above, we come up with a mapping algorithm of our own, utilizing the characteristics of TCAM. In Section III, we present some of the possibilities of making an effective use of the high-speed memory in routers.

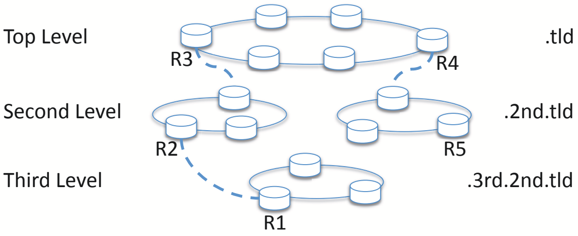


Fig. 2. Hierarchically structured logical topology

Showing that the proposed algorithms result in no drastic changes in the routers' forwarding tables against the updates of the FQDN database and proposing a mapping algorithm between the logical and physical topology are both important in suggesting the feasibility of name-based routing. The former demonstrating the scalability, stability, and robustness and the latter indicating that the logical topology shown in Fig. 2 is applicable to the already deployed physical routers.

Upon demonstrating that routing by FQDN is a feasible method, we aim at generalizing the result and show that content-based routing within the routers is a feasible technology in the future. The advantage of being able to route with content or names is that resolving names to IP addresses will not be necessary anymore. In addition, ID and locator can be separated, which has been one of the main problems of IP addresses [15]. The result we obtain from this work is intended to be used in designing a new hardware forwarding engine, especially a new TCAM architecture suitable for content-based routing by the semiconductor company Renesas Technology Corporation.

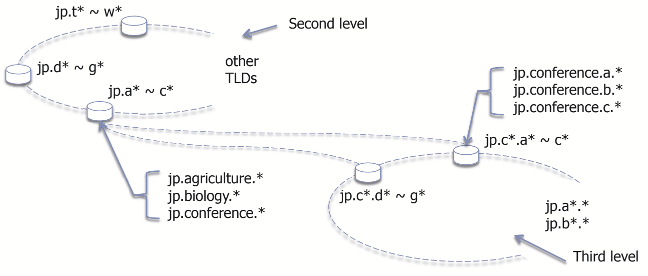
The rest of this paper is organized as follows. Section II evaluates how dynamic updates of the FQDN database affect the topology and the number of routers required to achieve name-based routing. Section III discusses the mapping algorithm which considers the characteristics of TCAM. Section IV concludes this paper by briefly summarizing the main points and sets out a vision for the resource-based routing.

II. Dynamic Updates of Database Entries

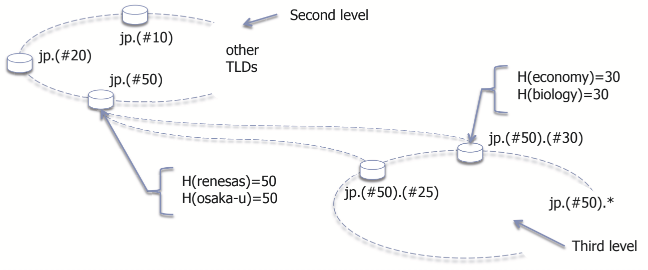
In this section, we evaluate the effect of dynamic updates of database entries by calculating the number of required routers in each quarterly data taken in the period between April 2008 and January 2009. The FQDN database is acquired from the Internet Systems Consortium (ISC) [16]. The number of routers is counted in each level where in the examples in Fig. 2, R3 and R4 are in 'Top Level', R2 and R5 are in 'Second Level', and R1 is in 'Third Level'.

A. *Distribution Algorithms of Domain Names in Routers*

We briefly explain the two main algorithms to distribute the FQDN entries used in this paper, *Hierarchical Longest Alphabet Match (HLAM)* and *Hybrid Distribution (HD)* which are shown in Fig. 3. HLAM is inspired by the longest prefix



(a) Hierarchical Longest Alphabet Match (HLAM)



(b) Hybrid Distribution (HD)

Fig. 3. Examples of HLAM and HD

match which expresses domain names with ASCII code. When alphabets have common bits, they can be stored in the same router. For example, `jp.a⋆`, `jp.b⋆`, and `jp.c⋆` can be stored in the same router since `a`, `b`, and `c`'s ASCII codes are `1100001`, `1100010`, and `1100011`, respectively, which can be expressed as `11000⋆⋆` in ternary logic. The advantage of this method is to make use of the don't care bit (⋆) which is the reason why TCAM is a memory with such a high search speed.

HD distinguishes FQDNs by their *Top Level Domain* (TLD) first and applies a hash function to 2nd level domain name and below. That is, domain names that have the same hash value are stored in the same router. For example, when the hash value of `renesas` (part of `renesas.co.jp`) and `osaka-u` (part of `osaka-u.ac.jp`) is the same, the FQDN is stored in the same router. By using hash values, FQDN entries are more equally distributed among the routers, therefore requiring fewer routers than HLAM.

B. *Update of FQDN Database*

The FQDN database is collected and published quarterly by ISC and the databases used in this paper are from April, July, October 2008, and January 2009. Fig. 4 shows how the change in the FQDN database affects the total number of required routers. A router is assumed to have ten 18 Mbit TCAMs and each entry consumes 180 bits. This is because the 18 Mbit TCAM is a common unit of TCAM and the 180 bits correspond to the popular SHA-1 hash value.

In order to use FQDNs for routing, it is assumed that a new routing table format is required in a clean slate manner. The stability of the very first table is important to minimize the effect of dynamic updates of databases. The more stable the initial table is, the easier it is to adapt to changes.

The topology change in the upper levels have more effect on the lower levels than vice versa. As shown in Fig. 4, the number of required routers in the Second Level does not change too much. Routers in the Second Level store unique 2nd level domain names. In the current FQDN database, none of the TLD's unique 2nd level domain names exceeds the size of a single router except for `com`. For `com`, we assume that there are multiple routers in the top level reserved for `com`. The reason for a slight increase in January 2009 is that five new TLDs were added and two TLDs were deleted.

The increase in the total number of routers from April 2008 to January 2009 was 12.5% and 10.4% for HLAM and HD, respectively. This can be seen as only a minor change to the whole topology.

C. *Effect of Aggregation on Router Memory Utilization*

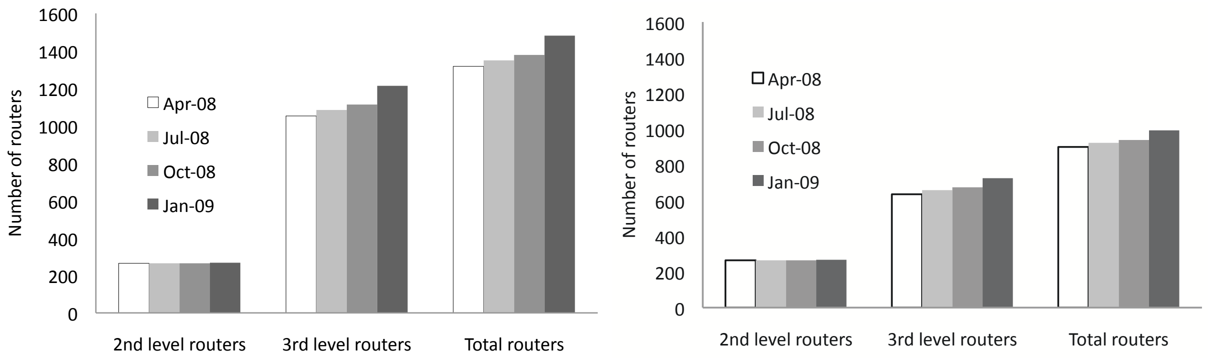
FQDN entries can be aggregated when they are written in the routers to decrease the number of required routers. Figures 5 and 6 show the effect of aggregation on the router utilization and the total number of routers where the exact values are in Table I. The routers are sorted by the utilization in increasing order on the x axis. Although we show the graph for all four databases (April, July, October, and January), there is not much difference among the curves.

$$Utilization = \frac{\text{Number of entries in a router} \times \text{Bits per entry}}{\text{Memory size}}$$

We use 180 bits per entry and ten 18 Mbit TCAMs for each router. Each figure (a) is before the aggregation and (b) is after the aggregation. The goal of aggregation here is to use as few routers as possible when routing information is written. In addition, the routers' memories are better utilized as shown in the figure, where there are more routers with higher utilization after the aggregation.

The aggregation of HLAM refers to the ASCII code of the TLD's alphabet. For example, if `ac`, `ad`, `ae`, `aero`, `af`, `ag`, `ai`, `al`, `am`, `an`, `ao` each use a router (i.e. 11 routers are used), the utilization is low, resulting in an inefficient usage of the memory resource. When these TLDs are aggregated by the ASCII code, `1100001110⋆⋆⋆⋆` can represent everything since `a` is `1100001` and `a - o` is `110⋆⋆⋆⋆`. The average utilization for a router increases by 7% and the number of required routers decreases by 12.6%.

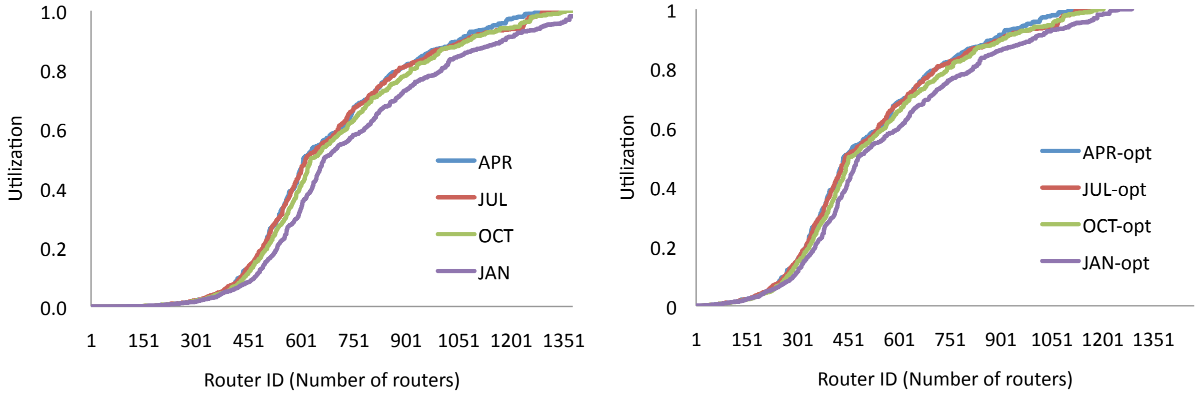
HD classifies FQDNs by the TLDs first and groups the FQDNs by their hash values in the lower level in the hierarchy. Since the algorithm uses hash values, the aggregation method differs from that of the HLAM which shows limitations of the alphabet. For example, suppose H(`osaka-u.ac.jp`) = 1000 (mod x) and H(`renesas.jp`) = 1000 (mod x) (i.e. both FQDNs are in group 1000, where x is an arbitrarily value of TLD's size) and H() is a hash function. Although the name is different, both FQDNs can be written in a router which has ID #1000. In other words, if the entries in routers with low utilization are aggregated to a single router, the average



(a) Increase in number of routers (HLAM)

(b) Increase in number of routers (HD)

Fig. 4. Change in number of required routers from Apr. 08 to Jan. 09



(a) Before aggregation

(b) After aggregation

Fig. 5. Number of required routers before and after aggregation (HLAM)

TABLE I
NUMBER OF REQUIRED ROUTERS FOR HLAM AND HD (BEFORE AND AFTER AGGREGATION)

TLD	Apr 08		Jul 08		Oct 08		Jan 09	
	Before	After	Before	After	Before	After	Before	After
HLAM	1307	1138	1342	1174	1375	1204	1471	1288
HD	922	668	941	685	957	703	1010	756

utilization for a router increases by 25.5% and the number of routers required decreases by 26.6%.

After distributing routing information to multiple routers using HLAM or HD, aggregating the entries will not increase the routing hops when discovering the resource. For HLAM, `1100001110****` in the TCAM as the next hop would match '`ac` (`110001 1100011`)' as well, therefore searching for `ac` would not require extra steps. For HD, if routers can calculate the domain name's hash value and figure out the next output port, no extra hops would be needed. However, the overhead of implementing hashing hardware within the router might be necessary.

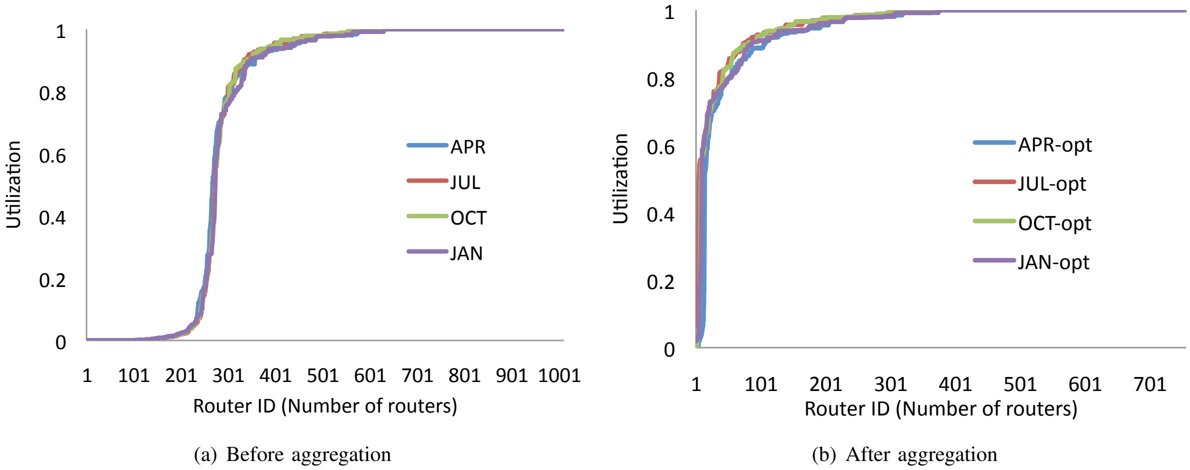
D. Robust Architecture against Dynamic Updates

In order for us to claim that using the distribution algorithms shows no drastic change in the topology upon the update of the database, it is necessary to show that the entries are not moved so frequently. Although Fig. 4 shows that the number of routers increases in scalable manner, it does not necessarily mean that it shows robustness in the topology changes.

Tables II and III show the change of routers where FQDN entries are written. 'Same' is defined as when the entry exists both 'before' and 'after' the update and the storage place is same, where storage place here is router. 'Different' is defined as when the entry exists both 'before' and 'after' the update, but the storage place is different.

$$\textit{Transfer ratio} = \frac{\text{Different}}{\text{Same}} \times 100$$

For three updates HD's transfer ratio is 4.7%, 3.3%, and 4.9% and HLAM's transfer ratio is 250%, 222%, and 218%, respectively.



(a) Before aggregation (b) After aggregation

Fig. 6. Number of required routers before and after aggregation (HD)

TABLE II
CHANGE IN ROUTER ID WHERE A FQDN IS WRITTEN (HD)

	Apr 08 → Jul 08	Jul 08 → Oct 08	Oct 08 → Jan 09
Same	495,106,893	518,627,699	531,823,934
Different	23,253,852	17,010,744	26,056,043

TABLE III
CHANGE IN ROUTER ID WHERE A FQDN IS WRITTEN (HLAM)

	Apr 08 → Jul 08	Jul 08 → Oct 08	Oct 08 → Jan 09
Same	147,836,915	165,916,860	174,961,495
Different	369,471,202	368,211,950	381,208,966

The reason why HLAM shows much higher transfer ratio is in the way that HLAM moves on to the next router when distributing the entries. HLAM aligns FQDN in alphabetical order and stores the ones starting from ‘a’ in routers. It refers to the ASCII codes and stops writing when the entries reach a router’s threshold and moves on to the next router to store further entries. When there are a lot of new FQDNs in one update, the border line that distinguishes the FQDNs in one router from the next router changes and the transfer ratio is increased.

On the contrary for HD, unless there is a sudden increase from the ‘before’ database, the transfer ratio does not change drastically because the storage place for a FQDN is decided by the FQDN’s hash value.

III. MAPPING OF LOGICAL TOPOLOGY TO PHYSICAL TOPOLOGY

As we showed above, there is numerous ongoing research to solve the mismatch between the logical and the physical topology mapping. In our current work, we are seeking for the possibilities of coming up with a modified mapping algorithm of our own using the characteristic of TCAM within the routers.

In our work so far, the abstract mapping of the logical topology to physical topology can be described using Fig. 7. The figure shows the network topology inspired by the Abilene Network [17], which is known to have similar characteristics to the real Internet topology. Let us assume the nodes in Fig. 7 groups routers into three levels from the center outwards to the edge: backbone, local gateway, and edge routers. The routers in the top level in Fig. 2 can be mapped to the routers in the backbone, the routers in the 2nd level to the local gateway, and the routers in the 3rd level to the edge routers.

The backbone routers can be configured statically since there are currently less than 300 TLDs. In order to find out which routers are suitable to store lower level domain names, RTT can be used as in [14]. The routers that return the minimal RTT indicate that they are physically close to the backbone routers.

Storing ‘content’ of the information and some kind of mapping information within the TCAM has been considered difficult in the past due to memory constraints. We aim at eliminating this obstacle by implementing new TCAMs with significantly more space, which is currently being developed by Renesas Technology Corporation. In addition, we are developing a data format containing the information of physical topology using TCAM’s ability to use don’t care bits. For example, a common prefix can be assigned to nodes that are physically closer to each other [11]. Utilizing TCAM’s ability to do a partial matching using the prefix, nodes can select its next hop node among several candidate nodes faster.

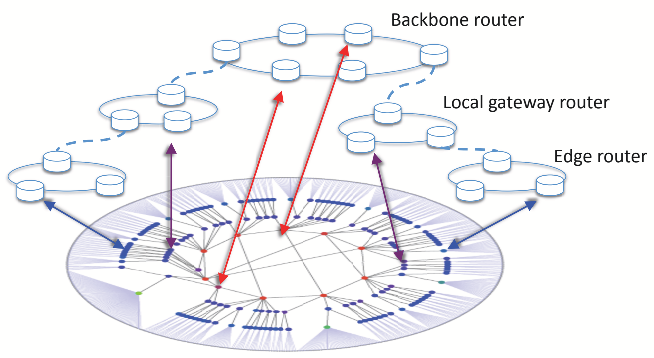


Fig. 7. Mapping of logical topology to Abilene-inspired physical topology

IV. CONCLUSION AND VISION OF RESOURCE-BASED ROUTING

In this paper, we showed that using domain names in routing is feasible and robust against the dynamic update of the database entries. Furthermore, we showed a possibility of developing a new TCAM architecture suitable for content-based routing when logical to physical topology mapping is considered.

Some of the fundamental technologies required to support name-based routing are sufficient hardware resource and robust network architecture. Our study suggests that it is possible to achieve routing with FQDN within the routers is feasible by (i) developing TCAMs that have larger size than currently available TCAMs, (ii) achieving robustness of the network topology against dynamically changing database entries, and (iii) having a locality-aware mapping algorithm. These factors are not limited to only name-based routing. Based on the discussions we had in this paper, we can generalize routing based on a 'resource', regardless of being a name, category, or type of a content. By enhancing our algorithms, we hope to propose an implementation of resource-based routing in the near future.

ACKNOWLEDGMENT

We thank Renesas Technology Corporation for their valuable advice and comments. This research was supported by the National Institute of Information and Communications Technology (NICT) of Japan.

REFERENCES

[1] FIND (Future InterNet Design), http://www.nets-find.net/.

[2] FP7 (Seventh Framework Programme), http://cordis.europa.eu/fp7/home_en.html.

[3] AKARI: Architecture Design Project that Illuminates the Path to the New Generation Network, http://akari-project.nict.go.jp.

[4] Renesas Technology Corporation, "Network Address Search Engine (9 M/18 M-bit Full Ternary CAM)," http://documentation.renesas.com/eng/products/others/rej03h0001_r8a20211bg.pdf.

[5] ——, "Highest Level Performance Ternary Content Addressable Memory (TCAM)," http://www.renesas.com/media/company_info/news_and_events/press_releases/2009/1210/20091210e.pdf.

[6] H. Hwang, S. Ata, and M. Murata, "A Feasibility Evaluation on Name-based Routing," in *Proceedings of the 8th IEEE international workshop on IP Operations and Management (IPOM)*, October 2009.

[7] B. Lampson, "Designing a Global Name Service," in *Proceedings of the 5th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, August 1986, pp. 1–10.

[8] S. Zoels, Z. Despotovic, and W. Kellerer, "On Hierarchical DHT Systems–An Analytical Approach for Optimal Designs," *Computer Communications*, vol. 31, no. 3, pp. 576–590, 2008.

[9] P. Ganesan, K. Gummadi, and H. Garcia-Molina, "Canon in G Major: Designing DHTs with Hierarchical Structure," in *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS)*, March 2004, pp. 263–272.

[10] I. Martinez-Yelmo, A. Bikfalvi, C. Guerrero, R. Cuevas, and A. Mauthe, "Enabling Global Multimedia Distributed Services Based on Hierarchical DHT Overlay Networks," *International Journal of Internet Protocol Technology*, vol. 3, no. 4, pp. 234–244, 2008.

[11] B. Hariri, S. Shirmohammadi, and M. R. Pakravan, "LOADER: A Location-Aware Distributed Virtual Environment Architecture," in *Proceedings of the IEEE Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VECIMS)*, September 2008, pp. 97–101.

[12] T. Qiu, G. Chen, M. Ye, and E. C. amd Bem Y. Zhao, "Towards Location-aware Topology in both Unstructured and Structured P2P Systems," in *International Conference on Parallel Processing (ICPP)*, September 2007.

[13] G. Xue, Y. Jiang, J. You, and M. Li, "A Topology-Aware Hierarchical Structured Overlay Network based on Locality Sensitive Hashing Scheme," in *Proceedings of the second workshop on Use of P2P, GRID and agents for the development of content networks (UPGRADE-CN)*, June 2007, pp. 3–8.

[14] Z. Xu, C. Tang, and Z. Zhang, "Building Topology-aware Overlays Using Global Soft-state," in *International Conference on Distributed Computing Systems (ICDCS)*, vol. 23, May 2003, pp. 500–508.

[15] D. Farinacci, V. Fuller, and D. Oran, "Locator/ID Separation Protocol (LISP)," *"draft-farinacci-lisp-10.txt", IETF Network Working Group, work in progress*, 2008.

[16] Internet Systems Consortium, https://www.isc.org.

[17] Abilene Network, http://www.internet2.edu/network/ .