Realization of Name Lookup Table in Routers Towards Content-centric Networks

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

***Abstract*—The future Internet is expected to be a highly intelligent that can route packets not only with an explicit destination address but also with the content. This paradigm shift in the network architecture from host- to content-centric communication naturally leads to the contemplation of the shift in the network layer devices, i.e. routers. In other words, the hardware architecture of routers should also be able to support content-centric communication. In this paper, we propose a new router architecture to manage a large information of contents and large-scale number of users . In order to complete the packet forwarding within the network layer, routers acting as the brokers of a publish/subscribe system should maintain the information of content names and the subscribers. We propose three memory structures for name lookup tables in routers, each with a different combination of memory types depending on the usage purpose. The proposed memory architecture is evaluated with parameters such as memory cost, latency, and utilization using real-life and synthetic databases that have a Zipf distribution. We show the memory architecture which has the lowest manufacturing cost and the lowest latency for storing the given database within a fixed budget.**

I. Introduction

The Internet in its early phase was designed around *who (host)*. The communication is performed among the nodes with known destination Internet protocol (IP) addresses but this is not an optimal design for the current usage of the Internet where *what (data)* rather than *who* is considered to be more important [1]. Future Internet is expected to be highly intelligent, indicating that routing is possible not only by the numerical identifier but also by the *content* [2]. In other words, when a content source sends data to an end node that requests some piece of information, the source can specify the content destination by the content's metadata instead of allocating the content with a single IP address. The advantage of content-centric networks is that the content source does not have to worry about the destination IP address of the receiver and send out the message to the network [3].

Determining the content's destination by its metadata indicates that a message that matches one or more conditions specified by multiple users should be sent to every user who requests the message. This is one of the biggest differences between host- and content-centric network and exchanging duplicate packets for one-to-many communication among the individual nodes makes the complex large-scaled Internet more and more congested. In such environment, publish/subscribe (pub/sub) is considered as an effective communication model [4] since relevant data is delivered to the consumers according to the interests they have expressed. Pub/sub consists of *publisher*, *subscriber*, and *broker*. The subscriber sends out a message for the content which it has an *interest* for. When the *publication* of the publisher matches this subscription's condition, the content is sent to the subscriber. The advantage of this interaction based on events is that the publisher and the subscriber do not have to communicate at the same time nor know about each other, since a server-like event service (*broker*) operates between the publisher and the subscriber [5]. The communication style of pub/sub also shows the need for paradigm shift in the router architecture. That is, it is only natural that network layer devices have to support the change in the network architecture from the host- to content-centric.

In this paper, we propose a hardware architecture where the network layer routers behave as the brokers in pub/sub that can perform well even when the number of the subscribers increases drastically. Specifically, we evaluate the proposed router memory architecture for storing topic names and the subscribers where topic names express the interest of a content. We share the fundamental idea with Jacobson et. al [6] which matches the content from the provider and the interest from the consumer by content names. We assume that the contents are already named and the router memory architecture is evaluated with parameters such as memory cost, latency, and utilization using the topic name and the subscriber databases that have Zipf distribution. In addition, the goal of this research is to propose a router architecture for future Internet, not to confront nor compete with application level solutions such as overlays [7] to realize content-centric networks. Implementing content-centric routing on network layer can replenish the conventional overlays by adding features such as accelerated lookup algorithm and optimized routing path that utilizes the physical topology. Furthermore, overlays can be used as a method to aid the migration from the host- to content-centric when the routers have a legacy architecture.

The rest of this paper is organized as follows. Section II surveys existing technologies on hardware architecture of conventional multicast and multimatch. Section III presents the proposed algorithm and Section IV explains the simulation settings that are used for evaluating the proposed algorithm. Section V concludes this paper by briefly summarizing the main points and mentioning future work.

II. Background Technologies

In this section, we introduce two background technologies of memory architectures in conventional routers.

- Multicasting: The multicasting forwarding table is typically composed of forwarding information base (FIB), adjacency table (ADJ), and multicast expansion table (MET) [8]. FIB has the information of source and multicast group, and ADJ has *rewrite MAC* and *MET index* which are all memories with a finite capacity. Especially for the MET that keeps the information of multiple output interface, the maximum number of output interfaces it can store is up to 64,000 [8]. When the table is full, multicast packets can not be processed in hardware and software switching in the CPU is required which drastically degrades the performance due to its slow speed [9].

- Multimatching: The address lookup in routers compares the input packet's destination IP address and the router's forwarding table's entry. Generally a single result that matches the longest bit sequence is returned which is also known as longest prefix match. The router's forwarding table is written in a special memory called ternary content addressable memory (TCAM) [10] which excels in high-speed searching and the memory address of a single entry that satisfies the search key is returned to SRAM by priority encoder (PE). However in a situation where each entry that matches the condition has to be returned (e.g. network intrusion detection system) [11], TCAM is searched multiple times. This is performed by setting the force no hit (FNH) bit to `1` to further prevent the entry from being matched for the given key.

III. Proposed Scenarios for Storing Database

In conventional routers, multicast is performed by matching a single TCAM entry with a given search key and the candidate output interfaces are stored in MET. However for pub/sub, the number of the subscribers of a content with a topic name can exceed several tens of thousands which can lead to a problem since the current memory structure is not optimized to support the system well for large number of subscribers. In order to support pub/sub notification service in routers, we propose three memory scenarios as shown in Fig. 1 that make use of the multimatch capability of TCAM and general purpose memory such as SRAM and DRAM.

A. Scenario A: Active TCAM and Passive SRAM

Scen. A maximizes the usage (active) of the multimatching in TCAM and minimizes the usage (passive) of SRAM by storing a small number of output interfaces for a topic per SRAM entry.

- Pros: Ability to utilize TCAM for high-speed searching. Number of keys can be searched in parallel using the latency between returning the result and the next search step.

- Cons: If two or more search keys are related to each other, single FNH bit is insufficient. For example, FNH of Addr `011` is set to `1` after being searched with the first key. If the next search key has a same prefix, this also matches the entry addressed `011` but ends up skipping the entry since the FNH bit is set to `1` after the first key.

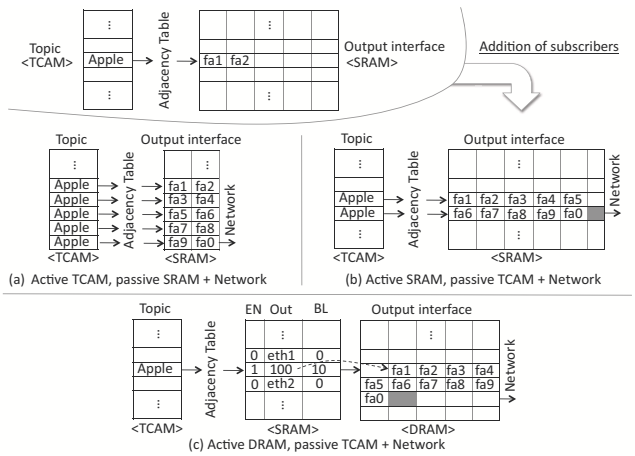


Fig. 1. Three scenarios to solve problems in current multicast hardware

B. Scenario B: Active SRAM and Passive TCAM

Contrary to the Scen. A, Scen. B makes an active usage of SRAM and minimizes the usage of TCAM and its multimatching process. For the number of users that exceeds the capacity of a single SRAM entry, multiple entries have to be used and the corresponding number of entries are used in TCAM as well as shown in Fig. 1(b).

- Pros: Chip cost per unit area of SRAM is 20% compared to TCAM and less FNH bits are required by minimizing the number of multimatch in TCAM. Minimizing the memory space used in TCAM also implies that the number of searches in TCAM decreases, resulting in lower power consumption.

- Cons: If a large number of horizontal bits in a row is reserved for the SRAM in order to minimize the number of required entries, the utilization of the memory can decrease if there are a lot of topics with a small number of subscribers.

C. Scenario C: Active DRAM and Passive TCAM

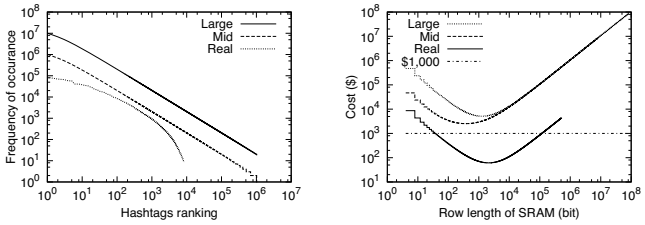
The output interface is written in DRAM. Unlike the Scen. A and B, only a single TCAM entry is consumed for a topic name. The only information stored in SRAM is whether DRAM referral is necessary by setting the EN bit to `0` (a single output interface is written in the SRAM) or `1` (DRAM address that has to be referred is written in the *Out* field with the burst length (BL) field).

- Pros: The chip cost per unit area of DRAM is 0.1% compared to SRAM and can handle a large number of subscribers at a low cost. In addition, by using a single TCAM entry for a topic name the power consumption can be reduced.

- Cons: DRAM has a high latency value compared to TCAM and SRAM, affecting the speed of the overall search process.

IV. Evaluation

The database of topic names and the number of subscribers used is taken from Hashtagsjp [12] and Twitter [13]. The database is assumed to have a rough Zipf distribution where the topic name subscribed by the ith most users has the number of $\frac{1}{i}$ users compared to that of the most popular topic name.



(a) Three databases (b) Evaluation of actual cost

Fig. 2. Cost evaluation of three databases (Scen. A and B)

TABLE I
COMPARISON OF MEMORIES

	DRAM	SRAM	TCAM
Wire speed	20 ns	1 ns	360 Msps
Read latency	20 ns	2 ns	75 ns
Cost per 10 Mbit	$1	$10	$50

In addition, two synthetic databases (Mid and Large) that have larger number of topics and subscribers are used. The characteristic of these database is shown in Fig. 2(a).

As for TCAM, SRAM, and DRAM in this evaluation, we follow currently used unit, 20 Mbit, 72 Mbit, 2 Gbit, respectively [14]. Referring to the cost per 10 Mbit in Table I, the chip cost for TCAM, SRAM, and DRAM is $100, $72, and $200, respectively. For example, if the combination for scenario A is one TCAM and one SRAM, the total chip cost is $172 which is calculated regardless of the used entries. The term actual cost is defined as 'used entries$\times$ cost per entry' of each memory.

A. Actual Cost and Read Latency

We first calculate the actual cost and the read latency when storing the database. The number of subscribers that can be stored in a router depends on the horizontal number of bits (row length) of the SRAM for Scen. A and B and the row length of DRAM for Scen. C. A long row in the memory is suitable for storing topics with a large number of subscribers but when there are many topics with a small number of subscribers, it can lead to the decreased utilization of the memories. We evaluate the actual cost and the utilization of the scenarios by setting this row length of the SRAM and DRAM as variables. When an SRAM entry is added, an additional entry in TCAM is required to be able to refer to that new SRAM entry. Therefore, the necessity for the multimatch increases and more FNH bits in the TCAM are required.

Fig. 3 shows the relationship between the read latency and the actual memory cost of the three scenarios. The x axis of Fig. 3(a) and 3(b) is the row length of SRAM and DRAM, respectively. When the row is short it is closer to Scen. A and otherwise closer to Scen. B. The extreme example of the Scen. A is when only a single user can be written in an SRAM entry, consuming 'number of topic names $\times$ number of subscribers in each topic' entries. The read latency is the worst-case value which is defined by the time consumed for searching and returning the result (list of subscribers) of the most popular topic name.

It seems that the Scen. C is the best solution since the information of subscribers is stored in the most inexpensive DRAM. However, DRAM has the highest read latency value, affecting the overall processing speed. Nevertheless, the read latency of Scen. A and B are higher than that of Scen. C in Fig. 3. This is because the latency of TCAM is calculated with its search latency which is in fact 75 ns. However, this should not form a hasty conclusion that Scen. A and B are slower than Scen. C. No matter how long a data entry is, it takes 75 ns to search TCAM with a key whereas for SRAM and DRAM, it takes $\frac{EntryLength}{WordLength} \times Latency$. In this simulation the parameters for $WordLength$ of SRAM and DRAM are both 18 bits. Therefore, the latency of SRAM and DRAM is only higher than TCAM when the row length is large enough, which is 675 bits and 6,750 bits for SRAM and DRAM, respectively.

The actual cost is defined as 'used entries$\times$ cost per entry of each memory'. For the extreme example of Scen. A, the actual cost ends up being approximately $8,600. Furthermore, it uses the entry of the TCAM most among the three scenarios, resulting in high search latency. Extending the row of the SRAM reduces the number of entries to store the subscribers for each topic up to a certain row length, also reducing the actual memory cost. This certain row length of SRAM, i.e. the value which minimizes the cost can be attained by differentiating Eq. (1) by x, where x is the SRAM row length.

$$Cost = (T + sx) \sum_{i=1}^{R} \left\lceil \frac{b_i P}{x} \right\rceil \tag{1}$$

The parameters are as follows:

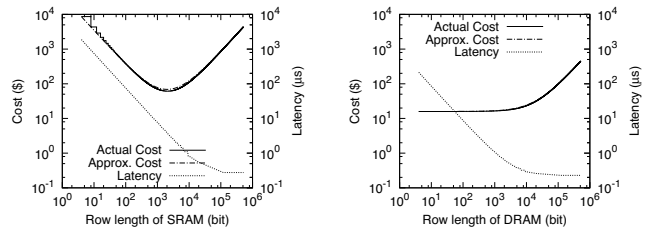
- Cost per entry of TCAM: $T = 320 \times 0.5 \times 10^{-5}$ (we assume 320 bits per a TCAM entry)
- Cost per bit of SRAM: $s = 0.1 \times 10^{-5}$
- Number of topic names in the database: R
- Number of subscribers for ith most popular topic: b_i
- Number of bits per output interface: P (= 4 bits)

The term on the right hand side consists of a product of linear increase and exponential decrease. However, due to the ceiling function, the sum does not approach 0, but is truncated at 1 for each $x \geq b_i P$. Therefore, Eq. (1) initially decreases then increases. Since the ceiling function is not differentiable, we approximate Eq. (1) as the following where $B = \sum_{i=1}^{R} b_i$.

$$\begin{aligned} Cost &= (T + sx) \sum_{i=1}^{R} \left(\frac{b_i P}{x} + 1 \right) \\ &= (T + sx) \left(\frac{BP}{x} + R \right) \\ &= \frac{BPT}{x} + sRx + (sBP + RT) \end{aligned}$$

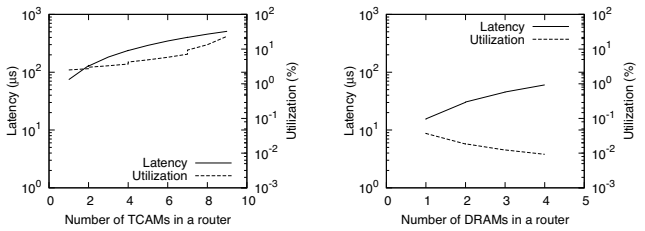
Differentiating *Cost* by x and the x that satisfies $Cost' = 0$ is the value which minimizes the *Cost*.

$$Cost' = -\frac{BPT}{x^2} + sR$$



(a) Scen. A and B (b) Scen. C

Fig. 3. Evaluation of actual cost and latency using real-life database



(a) Scen. A and B (b) Scen. C

Fig. 4. Evaluation of latency and utilization using real-life database

TABLE II
SRAM ROW LENGTH (x) THAT MINIMIZES COST IN SCEN. A AND B

	Real	Mid	Large
x bits from Eq. (1)	1,948	376	1,272
Min. Cost from Eq. (1)	\$61	\$2,526	\$5,111
x bits from Eq. (2)	2,024	419	1,335
Min. Cost from Eq. (2)	\$69	\$2,697	\$5,699

$$x = \sqrt{\frac{BPT}{sR}} \tag{2}$$

Table II shows the result of the Eq. (1) and (2). For all databases the approximated x and the cost from the Eq. (2) are slightly larger than the actual x and cost from the Eq. (1) but generally show a good resemblance as shown in Fig. 3 (Actual Cost and Approx. Cost). In addition, Fig. 2(b) shows the actual cost using the three databases.

Fig. 3(b) shows the result of Scen. C where a topic name consumes an entry in each TCAM and SRAM and the subscribers are written in the DRAM. Since the cost per bit in DRAM is lower than any of the memory types, increasing the row of the DRAM does not affect the overall actual memory cost even though the required entries decrease.

Due to the space limitations we omit the additional parameters and the calculation of the cost. The differentiated value is constant however and the minimal $Cost_c$ does not exist. Nevertheless, drastically increasing the row length creates wasted space after all, making the actual memory cost approximately \$450 which is still trivial compared that of Scen. A and B.

B. Read Latency and Utilization with Fixed Budget

In reality, the row length of SRAM and DRAM can neither be adjusted nor increased to a large value of tens of thousands as one desires as shown in Fig. 3. Instead we add more restrictions and realism to get the results shown in Fig. 4 by setting the manufacturing chip cost to \$1,000 and the row length of the SRAM and the DRAM to 32 bits. To compare the memory combinations that have a similar manufacturing chip cost, only those between \$900 and \$1,000 are selected for the evaluation. The x axis of Fig. 4(a) is the number of TCAM chips and since the manufacturing chip cost is fixed, the number of SRAM chips decreases as TCAMs increase. Two different utilization values exist when there are four and seven TCAM chips because there are two combinations each that satisfy \$900 – \$1,000 condition. The x axis of Fig. 4(b) is the number of DRAM chips and since Scen. C puts emphasis on the memory capacity, the number of SRAMs increase and TCAMs decrease as the DRAMs increase.

The utilization of Scen. A and B increase as the number of TCAM chips increase. This is because for a given budget, the number of SRAM decreases as the number of TCAM increase, resulting in lower memory capacity for storing the output interface. However, the utilization decreases as the number of DRAM chips increase for the Scen. C. Although more DRAM means less TCAM and SRAM, the unit of DRAM chip is 2 Gbit each, therefore reducing the utilization of the total memory. For larger databases, the utilization can exceed 100% meaning all memory space in a router is used. When this occurs, the searching of topic names and forwarding is delegated to other routers in the network.

From Fig. 4(a), it is shown that the overall latency is largely affected by the number of TCAMs whereas in Fig. 4(b), the number of DRAMs affect the overall latency. TCAM is a special kind of memory generally used for high-speed searching but the total latency accumulates to become high since the whole memory chip has to be searched regardless of the data length.

V. DISCUSSION AND CONCLUSION

In this paper, we discussed the advantages of realizing content-centric routing in the network layer and evaluated the proposed router memory architecture for storing topic names of contents. As a result, we have shown that the memory architecture is affected by the database of topic names and users having Zipf distribution, and the latency of each memory.

The current work in progress is on proposing an algorithm specifying the '+network' part for delegating the searching and forwarding of topic names to other routers in the network. This delegation occurs when the utilization of the router memory exceeds 100% in Fig. 4 or the given budget of \$1,000 shown in Fig. 2(b). After extending our algorithms with further evaluations, we hope to propose an implementation of content-centric network in the network layer in the near future.

ACKNOWLEDGMENT

This research was partially supported by the Promotion program for Reducing global Environmental loaD through ICT innovation (PREDICT) of Ministry of Internal Affairs and Communications, Japan.

REFERENCES

[1] S. Paul, J. Pan, and R. Jain, "Architectures for the Future Networks and the Next Generation Internet: A Survey," *Computer Communications*, vol. 34, no. 1, pp. 2–42, January 2011.

[2] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A Routing Scheme for Content-Based Networking," in *Proc. IEEE INFOCOM*, vol. 2, March 2004, pp. 918–928.

[3] R. Chand and P. Felber, "A Scalable Protocol for Content-based Routing in Overlay Networks," in *Proc. IEEE NCA*, April 2003, pp. 123–130.

[4] G. Li, V. Muthusamy, and H. Jacobsen, "Adaptive Content-based Routing in General Overlay Topologies," in *Proc. IFIP/ACM Middleware*, December 2008, pp. 1–21.

[5] S. Arianfar, P. Nikander, and J. Ott, "On Content-centric Router Design and Implications," in *Proc. ReArch*, November 2010, pp. 1–6.

[6] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," in *Proc. CoNEXT*, December 2009, pp. 1–12.

[7] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. 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, October 2007.

[8] Cisco Networkers, "Cisco Catalyst 6500 IP Multicast Architecture and Troubleshooting," 2006.

[9] Cisco Systems, "Understanding and Configuring IP Multicast, Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.1(12c)EW ."

[10] Renesas Electronics Corporation, "Network address search engine (9m/18m-bit full ternary CAM)," http://documentation.renesas.com/eng/products/others/rej03h0001_r8a20211bg.pdf.

[11] W. Jiang and V. K. Prasanna, "Field-split Parallel Architecture for High Performance Multi-match Packet Classification using FPGAs," in *Proc. SPAA*, August 2009, pp. 188–196.

[12] Hashtagsjp, http://hashtagsjp.appspot.com/.

[13] Twitter, http://twitter.com.

[14] Renesas Electronics Corporation, "Memory products," http://www.renesas.com/prod/memory/.