Two Dimensional Router: Design and Implementation

Shu Yang
Tsinghua University

Mingwei Xu
Tsinghua University

Dan Wang
Hong Kong Polytechnic University

Jianping Wu
Tsinghua University

***Abstract*—Packet classification has attracted research attentions along with the increasing demands for more flexible services in the Internet today. We present our design and implementation of a two dimensional router (TwoD router). It makes forwarding decisions, in hardware level, based on destination and source addresses. This can fundamentally increase routing semantics to support services beyond destination-based routing.**

With one more dimension, it is no longer possible to fit such forwarding table into TCAM, the de facto router standard. In this paper, we propose a new forwarding table structure through a neat separation of TCAM and SRAM where we harvest the speed of TCAM and the storage/flexibility of SRAM. We evaluate our design with an implementation on a commercial router. Our design does not need new hardware.

I. INTRODUCTION

To provide reachability service, conventional Internet routers classify packets based on destination address. Such one dimensional router is adequate for destination-based packet delivery. Nevertheless, there are increasing demands for such services as policy routing, security, traffic engineering, quality of service, etc. The paucity of routing semantics makes innovation in developing higher level services more difficult or restricted to limited scope, e.g., on edge routers only.

There are many studies on higher dimensional routers, among which 5D (destination and source addresses, destination and source ports, transport layer protocol) [4] and 2D (destination and source addresses) [3] have attracted most attention. Many studies are software-based solutions. However, software-based solutions need many accesses to memory, and cause non-deterministic lookup time.

Due to the importance of destination and source addresses, we focus on two dimensional router. There are many challenges of it. For hardware-based forwarding, the de facto standard of the routers is TCAM-based forwarding. It achieves fast and constant lookup time. TCAM, however, has low capacity, large power consumption and high cost. The largest TCAM chip can only accommodate 1 million IPv4 prefixes.

When designing a TwoD router, the immediate change it brings about is the forwarding table size. More specifically, the Forwarding Information Base (FIB) stored in TCAM will tremendously increase. Two dimensional classifiers widely adopt the traditional Cisco Access Control List (ACL) structure (we call it ACL-like structure thereafter) With ACL-like structure, FIB table changes from $\{\text{destination}\} \rightarrow \{\text{action}\}$ to $\{(\text{destination, source})\} \rightarrow \{\text{action}\}$. Let N be the number of destination prefixes and M be the number of source prefixes. In the worst case, the number of TCAM entries can be $O(N \times M)$. This structure increases TCAM size by an order and a practical consequence is that TCAM cannot hold entries of such scale. Note that the number of destination prefixes in current backbone routers is 400,000 [1]. If a TwoD router is implemented by an ACL-like structure, it is beyond the TCAM storage even with a few tens of source prefixes.

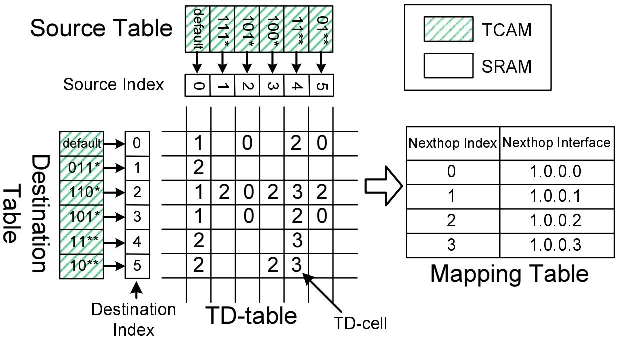
In this paper, we design a new forwarding table structure called FIST (**FI**B **S**tructure for **T**woD-IP). The key of FIST is a neat separation of TCAM and SRAM. SRAM has larger memory and it is 10 times cheaper and consumes 100 times less energy than TCAM. In FIST, TCAM contributes fast lookup and SRAM contributes a large memory space for nexthop storage. This separation reduces the TCAM storage from $O(N \times M)$ to $O(N + M)$.

II. FIST STRUCTURE

A. The TwoD Forwarding Rule

In conventional router, longest matched first (LMF) rule is used to decide which destination prefix will be matched. Here, we first present the definition of the forwarding rules that is used in two dimensional routing. Let d and s denote the destination and source addresses, p_d and p_s denote the destination and source prefixes for d and s. Let a denote an action, more specifically, the nexthop corresponding to p_d and p_s. The storage structure of a TwoD router is a table of entries of 3-tuple (p_d, p_s, a).

Definition 1. ***TwoD forwarding rule**: Assume a packet with s and d arrives at a router. The destination address d should first match p_d according to the LMF rule. The source address s should then match p_s according to the LMF rule among all the 3-tuples given that p_d is matched. The packet is then forwarded to next hop a.*



Destination Index	0	1	2	3	4	5
0	1		0		2	0
1	2					
2	1	2	0	2	3	2
3	1		0		2	0
4	2				3	
5	2			2	3	

Nexthop Index	Nexthop Interface
0	1.0.0.0
1	1.0.0.1
2	1.0.0.2
3	1.0.0.3

Fig. 2: FIST: A forwarding table structure for TwoD-IP

B. FIST Basics and Correctness

1) FIST Basics: The key idea is a separation of TCAM and SRAM (see Fig. 2). In this separation, the destination and source prefixes are stored in TCAM, with an offset table pointing to the nexthop table stored in SRAM. As the nexthop

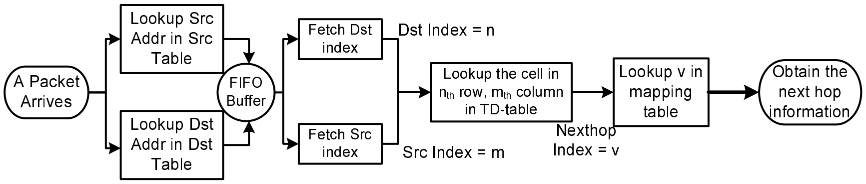


Fig. 1: Lookup action in FIST

information is long, we have another mapping table so that the main SRAM nexthop table only stores an index.

Formally, we design Forwarding Information Base Structure for Two Dimensional Router (FIST). FIST has two tables stored in TCAM and two tables stored in SRAM. In TCAM, one table stores the destination prefixes mapping to a *destination index* (we call the table *destination table* thereafter), and one table stores the source prefixes mapping to a *source index* (we call the table *source table* thereafter). One table in SRAM is a two dimensional table that stores the indexed nexthop of each rule (we call it *TD-table* thereafter) and we call each cell in the array *TD-cell*. The destination and source indexes in TCAM point to a TD-cell in SRAM. The other table in SRAM stores the mapping relations of index values and next hops (we call it *mapping-table* thereafter).

For each rule (p_d, p_s, a), p_d is stored in the destination table, and p_s is stored in the source table. For the (p_d, p_s) cell in the TD-table, there stores an index value. From this index value, a is stored in the corresponding position of the mapping table. Note that the index value is much shorter than the nexthop information a.

With FIST, The FIST TCAM storage is $O(N + M)$ bits. The FIST SRAM storage is $O(N \times M)$ bits.

Clearly, FIST migrates the "multiplication" factor into SRAM, rather than eliminate it. Such migration is based on the following facts: 1) TCAM storage capacity is much smaller than SRAM; 2) TCAM is 10-100 times more expensive than SRAM; 3) TCAM consumes 100+ times more power than SRAM [2].

This migration does not slow down the forwarding speed. The dominant factor for forwarding speed is TCAM lookup, which FIST maintains. In FIST, there are additional SRAM accesses. We discuss this shortly and develop a pipelining scheme so that the router performance is the same with that of conventional routers.

2) TD-cell Saturation: Note that after inserting the rules, there will be empty cells. To address the problem, we develop algorithm TD-Saturation() to saturate the conflicted cells with appropriate index value.

Algorithm 1: TD-Saturation($\mathcal{R}$)

```
begin
    // $\mathcal{R}$ is the set of TwoD forwarding rules
    foreach $p_d, p_s$ do
        if $\nexists (p_d, p_s, a) \in \mathcal{R}$ then
            $\mathcal{S} = \{(\bar{p_s}, \bar{p_d}, \bar{a}) \in \mathcal{R} | \bar{p_d} = p_d\}$
            $\mathcal{S}' = \{(\tilde{p_s}, \tilde{p_d}, \tilde{a}) \in \mathcal{S} | \tilde{p_s}$ is a prefix of $p_s\}$
            Find $(\hat{p_s}, \hat{p_d}, \hat{a}) \in \mathcal{S}', \forall (p'_d, p'_s, a') \in \mathcal{S}', p'_s$ is a prefix of $\hat{p_s}$
            Fill the cell $(p_d, p_s)$ with index value of $\hat{a}$.
```

Theorem 1. *FIST (with* TD-Saturation*()) correctly handles the rule defined in Definition 1.*

Proof: Suppose not. The packet will match another rule other than $(\hat{p_s}, \hat{p_d}, \hat{a})$. If the rule does not belong to $\mathcal{S}$, then p_d is not matched. If the rule does not belong to $\mathcal{S}'$, then p_s should not been matched. If the rule is not $(\hat{p_s}, \hat{p_d}, \hat{a})$, then p_s is not the LMF match given p_d is matched. ■

III. FIST LOOKUP

We first present the basic lookup steps and then show a pipeline lookup. We will show that the pipeline lookup achieves the same performance as the conventional routers for each lookup operation.

The lookup action is shown in Fig. 1. When a packet arrives, the router matches the destination and source prefixes in parallel in the destination and source tables in TCAM. This parallelism is possible since we have a saturated TD-table. The destination table and source table then each outputs the SRAM addresses that point to the destination index and source index. The SRAM addresses are passed to an FIFO buffer, which resolves the un-matching clock-rates between TCAM and SRAM. The router then obtains the destination index and source index. The router can thus identify the cell in the TD-table, and return the index value. Using this index, the router looks up the mapping table, and returns the nexthop.

Theorem 2. *The lookup speed of FIST is one TCAM plus three SRAM clock cycles.*

Proof: The theorem is true because source and destination tables (indexes) can be accessed in parallel. ■

As a comparison, the conventional destination-based routing stores destination prefixes in one TCAM, and accesses both TCAM and SRAM once during a lookup.

We can also develop a pipeline lookup process for further amortizing each individual lookup operation. Using the pipeline, the lookup speed of FIST can achieve one packet per clock rate.

We implement the FIST structure on a commercial router, Bit-Engine 12004. Our implementation is based on existing hardware, and does not need any new hardware. The results show that FIST lookup achieves linecard speeds

REFERENCES

[1] Bgp routing table analysis reports. http://bgp.potaroo.net.

[2] Router fib technology. http://www.firstpr.com.au/ip/sramip-forwarding/router-fib/.

[3] F. Baboescu, P. Warkhede, S. Suri, and G. Varghese. Fast packet classification for two-dimensional conflict-free filters. *Comput. Netw.*, 50(11):1831–1842, 2006.

[4] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In *Porc. ACM CoNEXT'08*, Madrid, Spain, Dec 2008.