Two Dimensional-IP Routing

Mingwei Xu [1], Shu Yang [1], Dan Wang[2], and Jianping Wu [1]
[1] Tsinghua University, Beijing 100084, China
[2] Hong Kong Polytechnic University, Hong Kong, China
Email: xmw@cernet.edu.cn; yangshu@csnet1.cs.tsinghua.edu.cn; csdwang@comp.polyu.edu.hk; jianping@cernet.edu.cn

Abstract—Traditional IP networks use single-path routing, and make forwarding decisions based on destination address. Source address has always been ignored during routing. Loss of source address makes the traditional routing system inflexible and inefficient. The current network can not satisfy demands of both the network users and the ISP operators. Although many patch-like solutions have been proposed to bring the source address back to the routing system, the underlying problems of the traditional routing system can not be solved thoroughly. In this paper, we propose Two Dimension-IP Routing (TwoD-IP), which makes forwarding decisions based on both source and destination addresses. However, combining with source address, both the forwarding table and routing protocol have to be redesigned. To overcome the scalability problem, we devise a new forwarding table structure, which achieves wire-speed forwarding and consumes less TCAM storage space. To satisfy demands of users and ISPs, we also design a simple TwoD-IP policy routing protocol. At last, we discuss the deployment problem of TwoD-IP.

Index Terms—TwoD-IP Routing, Forwarding Table, Incremental Deployment

I. Introduction

Internet has become one of the most successful communication networks world-wide, attracting billions of users and creating great number of applications. However, with more users, the Internet faces many challenges. For example:

Traffic inside an ISP network is unevenly distributed;

Complex network measurement and anomaly detection always annoy the network operators;

Multi-path routing is hard to be used, and multi-homing faces problem with ingress-filtering [1];

Flexible traffic management or policy routing is quite difficult, within destination-based single-path routing.

The current Internet makes forwarding decisions independently at each node according to the destination address of each packet. This simplicity, or dump core principle, of the traditional Internet pushes all complexities to the edges. However, for simplicity, traditional networks over-emphasize on their reachability to destinations, but do not pay much attention to other aspects related to sources. With the tremendous growing of the Internet, there are increasing demands for identifying the sources of traffic, e.g., ISPs usually desire to divert traffic from one customer network to an egress router, rather than the one selected by the best path selection algorithm of BGP [2]. The absence of source address identity in the routing system causes many problems. For example, it is difficult for malicious traffic from hackers to be filtered, and difficult for traffic for emergency service to take precedence.

To make up the deficiencies of the traditional network, there are many patch-like solutions. Such as source routing [3], where the sender can specify the routing path of a packet, and MPLS [4], that sets up static routing paths using label switching. However, source routing hands most control to the end users, while MPLS brings additional protocol complexity and control overheads. In addition to source routing and MPLS, overlay [5] can also be used. This however, is beyond the network layer. All these solutions are not widely deployed, and fail to satisfy the demands from sources. For an ISP, a light weight, pure IP-based, more network controllable solution is desired.

For security reasons, China Education and Research Network 2 (CERNET2), the world's largest IPv6 backbone network (including 59 Giga-PoPs), has deployed SAVI (source address validation improvement) [6], where the source address of each packet is checked. SAVI guarantees that each packet will hold an authenticated source IP address, and thus enhances the security of the network.

To achieve better manageability and flexibility during routing, we are now deploying Two Dimensional-IP (TwoD-IP) routing. More specifically, the forwarding decisions of intermediate routers will be based on both the destination addresses and the source addresses. Packets from different sources towards the same destination may be delivered to different next hops in TwoD-IP routing, rather than the same one that is on the shortest path in traditional routing.

With TwoD-IP, the routing system will become more flexible, manageable and reliable. However, the new TwoD-IP routing architecture will cause additional overheads in both data and control planes, which can be seen as a trade-off between simplicity and flexibility. In data plane, storage cost may increase explosively with one more dimension in the forwarding table. In control

Manuscript received March 7, 2013; revised April 17, 2013.
doi:10.12720/jcm. 8.4.249-258

plane, we need new routing protocols to control the routing paths of traffic from different sources.

We devise a new forwarding table structure for TwoD-IP. The new TwoD-IP forwarding table structure uses two separate TCAM tables to store source and destination prefixes, and a larger SRAM array to store the next hop information. When packets arrive, the router first lookups both source and destination addresses in the two TCAMs, and then use the output information to access the SRAM array and obtain the next hop information. Within the new structure, we can almost keep the same speed as the traditional destination-based forwarding table, and also achieve a tolerable growth of storage.

We design a policy routing protocol based on extensions of OSPF. It can divert traffic from a customer network to another egress router rather than a default one. ISP operators can flexibly use the new protocol to carry out their policies.

We have developed prototypes of the TwoD-IP routers and new protocols on a commercial router, Bit-Engine 12004, and set up small scale tests under our testbed as well. The results show that TwoD-IP routers can achieve line speeds.

II. Related Work

There is little work on two dimensional routing since destination-based IP routing won over circuit based routing such as PNNI [7]. Because of the important semantic in source address, recent years see more research on giving sources control.

IP (loose/strict) source routing [3], where the route is carried in the packet, is naturally combined with IP protocol, and allows the sender to take full control of the routing path. However, due to security reasons [8], source routing is disabled in most networks. In addition, source routing hands most control to the end users, which is unfavorable for ISP operators. MPLS [4] is often used to manage traffic per flow. However, due to the control and management overheads, MPLS raises concern about scaling when the number of label switching paths (LSPs) increases [9]. The more the LSPs, the heavier the system burden [10]. Overlay [5] can also be used. This however, is beyond the network layer. For an ISP, a lightweight, pure IP-based, and more network-controllable solution is desired.

There are many other routing schemes that have been combined with source address lookup, such as policy-based routing (PBR) [11], customer-specific routing [12], user-directed routing [13], multi-topology routing, where traffic flows on user-specific topology [14]. In our paper, we try to design a routing architecture which is well combined with source address lookup, and scales in both control and data planes.

Source Address Validation Improvement (SAVI) is a working group in IETF, which is aiming at providing a standardized mechanism for IP source address validation at a finer granularity. CERNET2 (China Education and Research Network 2) has deployed SAVI [6], that guarantees that each packet will hold an authenticated source IP address. Currently, confirmed SAVI users are more than 900,000. CERNET2 then plans to further deploy TwoD-IP Routing based on authenticated source IP addresses in its network.

III. Adding Source Address to the Routing System

There is little work on two dimensional routing since destination-based IP routing won over circuit based routing such as PNNI [7]. Because of the important semantic in source address, recent years see more research on giving sources control.

In the current Internet routing, only destination address is used for forwarding decision. This fundamentally limits the diversity of the functions and services that the Internet routing system can provide. To improve quality-of-services facing the demands from the users and applications, there are such proposals as source routing [3], ingress filtering [15], MPLS [4], and even overlay routing [5] at application layer. Each of these proposals provides one or a few QoS functions. Many of these proposals include source addresses, explicitly or implicitly, in their decision making; however, with their own syntax. It is widely accepted that the routing system today is less expressive and provides less basic primitive functions.

In this paper, we propose to add source address into the Internet routing system so that routers can make forwarding decisions based on both source and destination addresses. This greatly enriches the semantics of the routing system. Some services are illustrated as the following.

Example 1, Policy routing: An ISP wants the traffic from source address A to destination address B passes by router C. With TwoD-IP routing, routers in the network make forwarding decisions based on both destination and source addresses, thus they can easily recognize packets from A to B, and divert the traffic to router C.

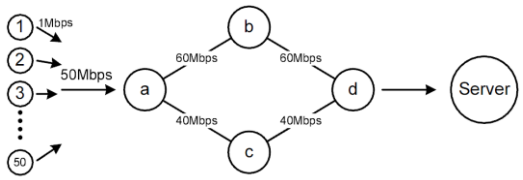


Figure 1. TwoD-IP routing for better traffic distribution

Example 2, Traffic engineering with Load-Balancing: Assume an ISP has four routers with the topology shown in Fig. 1. Assume there are 50 hosts attached to the ingress router a, and each host sends traffic to the server attached to the egress router d at 1Mbps. The total traffic demand is 50Mbps. Using current destination-based single-path routing, traffic towards the same destination should take the same route. To achieve Min-max link utilization, all traffic will take the route through b and the

maximum link utilization is 83.3%. With TwoD-IP routing, router a could differ according different sources. The optimal distribution is to let traffic of 30 hosts take the route through b, and traffic of the other 20 hosts take the route through c; the maximum link utilization is 50.0%.

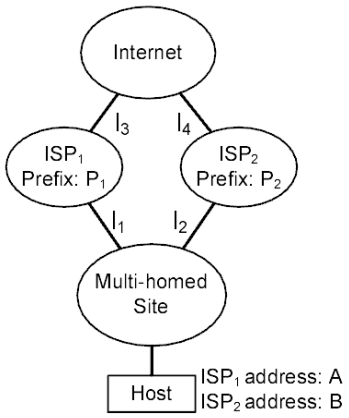


Figure 2. TwoD-IP routing for multi-homing.

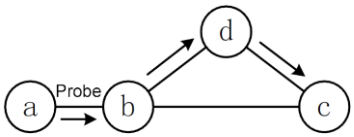


Figure 3. TwoD-IP routing for network monitoring.

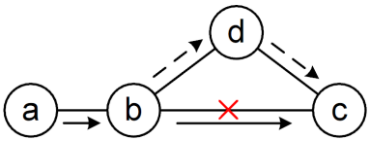


Figure 4. TwoD-IP routing for better reliability.

Example 3, Traffic engineering with Multi-homing: Traditionally, provider-independent (PI) address is used for multi-homing. As the PI address causes the routing table inflation problem, provider-aggregatable (PA) address is recommended. PA is inflexible, however, and it imposes heavy configurations on ISP administrators [16]. In Fig. 2, assume a multi-homed site is connected to two ISPs, ISP1 that owns prefix P1 and ISP2 that owns prefix P2. A host in this site has two addresses, address A that belongs to P1 and address B that belongs to P2. With TwoD-IP routing, the routers of this site can forward packets towards the Internet according to the source address, i.e., packets with source address A will be forwarded to ISP1 and packets with source address B will be forwarded to ISP2.

Example 4, Diagnosis: In Fig. 3, assume an ISP has four routers. To monitor link (b, c), the ISP sets up a monitor at router a. With destination-based routing, link (b, c) is on the shortest path from a to c. Therefore, a can just send the probe packets with destination of c to monitor the (b, c) link. However, the network provides traffic engineering capabilities. If there is congestion on link (b, c), and router b moves the traffic towards c by path (b, d), (d, c), the probe packets from a will fail to monitor link (b, c). With TwoD-IP routing, through identifying the source address from a, b will recognize that these packets are for probing and can forward them through link (b, c) and the link can be monitored.

Example 5, Reliability: In Internet routing, failures happen everyday [17]. Combined with source address, the routing system can intrinsically provide multiple paths to the same destination. Thus, TwoD-IP routing can be used for link/node, or even path-level [18] protection. In Fig. 4, router b forwards packets from a to c through link (b, c). In traditional routing, packet forwarding will be interrupted once link (b, c) fails. With TwoD-IP routing, router b can select d as the backup next hop towards c, and set up a backup route for destination c. When b detects the failure of the link between b and c, it can automatically reroute the packets to the backup path. Such protection scheme is the goal for many fast reroute schemes [19]. TwoD-IP routing directly support fast reroute for link/node protection.

Example 6, Multi-path Routing: The Internet is over-provisioned with links and bandwidth; it is well-known that the Internet routing can be more efficient with multipath routing. However, it is not straightforward for an ISP to support flexible multi-path in a traditional routing system. ISPs have to go over through MPLS or overlay network, both of which bring overheads and complication to the network. It is much simpler given TwoD-IP routing. See the example in Fig. 5, where the network has four routers, a host connected to router a sends packets to d. With TwoD-IP routing, we can provide multiple paths towards the same destination at the same time. To achieve this, we only need to let the host own multiple source addresses, e.g., A, B and C. Router a can make forwarding decisions based on these source addresses (together with the destination address). For example, a can forward the packets with source address A directly to d, the packets with source address B to b, and the packets with source address C to c.

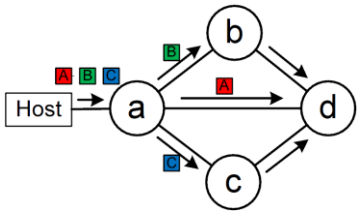


Figure 5. TwoD-IP routing for Multi-path.

Example 7, Security: Traditionally, for security reasons, most networks deploy source IP checking functionality on the border routers. Facing larger and larger DDoS attacks from multimillion-node botnets, border routers can hardly stop them [20]. Assume in Fig. 6, three source hosts are attacking the victim by sending large amounts of traffic data. Traditionally, only router a can install filters. However, Due to limited filter capacity, router a can only set up one filter, that can only block traffic from one source host. Thus the victim still suffers attacks from two source hosts. With TwoD-IP routing, source IP

checking functionality is deployed deeper in the network, such that both b and c can set up one more filter. Suppose that router a, b and c each filters one source host, the network can successfully filter all malicious traffic.

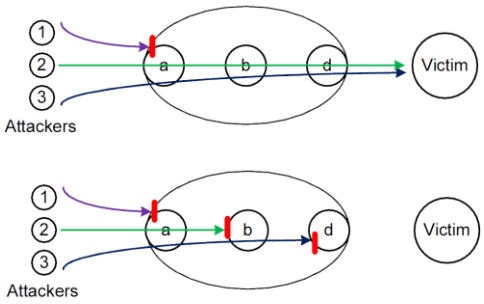


Figure 6. TwoD-IP routing for better security.

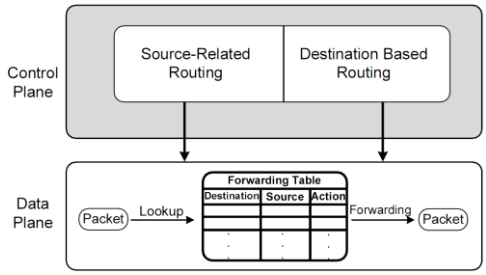


Figure 7. TwoD-IP Routing Framework.

The benefit from adding source addresses to the routing system is not limited to the above examples. Intrinsically, we enrich the semantics of the entire Internet routing system.

IV. Overview of TwoD-IP

Fig. 7 shows the architecture of our TwoD-IP routing. Similar to the traditional architecture, it is separated into data plane and control plane.

A. Data Plane

Each entry of the TwoD-IP forwarding table is a 3-tuple, i.e., {source prefix, destination prefix, next hop}. When a packet arrives at a router, the router checks both destination address and source address, and then outputs a corresponding action (e.g., the next hop to be forwarded).

Compared with traditional destination-based table, the forwarding table in TwoD-IP routing may be much larger. If M is the size of source address space, a straightforward implementation will result in an increase of an order of M. We will discuss a neat structure to address this problem in Section V.

B. Control Plane

Each entry of the TwoD-IP forwarding table is a 3-tuple, i.e., {source prefix, destination prefix, next hop}. When a packet arrives at a router, the router checks both destination address and source address, and then outputs a corresponding action (e.g., the next hop to be forwarded).

Traditional routing protocol only exchanges network status information (e.g., network topology). TwoD-IP routing can meet more demands of the network users and ISPs. Therefore, the control plane can be more flexible. The key component is the routing protocols with updates according to both topology changes and policy changes. There are two components of the control plane of our TwoD-IP framework: destination-based routing protocols and source-related routing protocols.

Destination-Based Routing Protocols: Traditional destination-based protocols, e.g., IS-IS and OSPF protocols that can run directly within the new architecture. The objective of these protocols is to provide connectivity services for users to reach the destinations. To provide better connectivity services, destination-based routing protocols should respond instantly to the changes of network topology.

Source-Related Routing Protocols: Based on the combination of network status and user demands, we can make better decisions on routing for both users and ISPs. We will present one in Section V. Different routing protocols can coexist, although they need to be consistent. Source-related routing protocol should respond to the changes of user demands or ISP policies. Depending on the specific user requirements, some source-related routing protocols need real-time updates, while others do not.

C. Key Challenges

Many opportunities can be explored, given that the TwoD-IP routing is deployed. To establish TwoD-IP routing, we consider the following main technical challenges.

Forwarding Table Design: The immediate change that TwoD-IP routing brings to the picture is the routing table size. More specifically, the Forwarding Information Base (FIB) will tremendously increase. Note that a first thought might think that the routing table only doubles. But this is not true, as for each destination address, it may correspond to different source addresses. A straightforward implementation means the FIB table should change from {destination} -> {action} to {(source address, destination address)} -> {action}.

This increases the FIB size for an order. The practical consequences can be calculated as follows. TCAM storage is 1 million. The current destination address space is 400,000. If TwoD-IP is used, and even if we only need to store 100 source prefixes, the total required storage is 40,000,000. This is far beyond a practical level. We solve this problem by proposing a neat FIST storage framework (see Section V).

New Source-Related Protocol: If all routers are equipped with source address checking functionality, we can design many source-related routing protocols for different purposes. Besides working correctly, the new protocols should be:

- Consistent: The protocols must be consistent with destination-based protocol and other source-related

protocols. There must be no loops, and no policy conflicts.

- Efficiency: The protocol overheads should be low, e.g., maintaining minimum states on routers and bringing minimum exchanged messages between routers.

To illustrate source-related protocols, we develop a simple policy protocol in Section VI.

Incremental Deployment: Deployment is always a difficult problem for Internet routing systems. For TwoD-IP routing, it can be changed within an AS. Nevertheless, an incremental deployment is still greatly needed. The goals can be grouped into three levels: 1) backward compatiblity, 2) visible gain if only partial routers are deployed, 3) an upgrade sequence that can maximize the gain in each step. We believe 1) and 2) are a must and 3) needs to be greatly favored. We discuss incremental deployment in Section VI-A.

The TwoD-IP design has three main components: forwarding table, routing protocol, and deployment scheme. We describe each design component in turn.

V. FIST: Forwarding Table Design

We propose a novel forwarding table structure FIST (FIB Structure for TwoD-IP) for our TwoD-IP forwarding table. It achieves fast lookup and small memory space. The key of our design is a neat combination of TCAM and SRAM. TCAM contributes to fast lookup and SRAM contributes to a larger memory space. Overally, our TwoD-IP forwarding table consumes O(N +M) TCAM storage space only, where N is the size of destination address space and M is the size of source address space.

We first present a clear definition of the forwarding rules that should be used in two dimensional routing. Let d and s denote the destination and source addresses, pd and ps denote the destination and source prefixes. Let a denote an action, more specifically, the next hop. The storage structure should have entries of 3-tuple (pd, ps, a).

Definition 1: Assume a packet with source address s and destination address d arrives at a router. The destination address d should first match pd according to the Longest Match First (LMF) rule. Then source address s should match ps according to the LMF rule among all the 3-tuple given pd is matched. The packet is then forwarded to the next hop a.

Different with traditional two dimensional layer-4 classification [21], which assigns the same priority to both source and destination addresses. We give higher priority to destination address, because 1) reachability to the destination addresses still belongs to our primary goals; 2) we can guarantee conflict-free [22], i.e., a packet will only match one entry.

The new structure FIST is made up of two tables stored in TCAMs and other two tables stored in SRAM (see Fig. 8). One table in TCAM stores the destination prefixes (we call it destination table thereafter), and the other table in TCAM stores the source prefixes (we call it source table thereafter). One table in SRAM is a two dimensional table that stores the indexed next hop of each rule in TwoD-IP (we call it TD-table thereafter) and we call each cell in the array TD-cell (or in short cell if no ambiguity). Another table in SRAM stores the mapping relation of index values and next hops (we call it mapping-table thereafter).

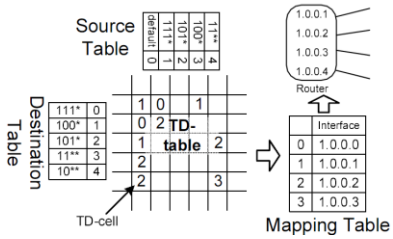


Figure 8. FIST: A forwarding table structure for TwoD-IP.

For each rule (pd, ps, a), pd is stored in the destination table, and ps is stored in the source table. We can obtain a row address in TD-table through pd, and a column address in TD-table through ps. Combining the row and column addresses, we can access a cell ((pd, ps) is used to denote the cell) in TD-table, and obtain an index value. According to the index value, a is stored in the corresponding position of mapping table. We store the index value rather than the next hop a in the TD-table, because next hop information is much longer.

For example, in Fig. 8, for rule (100*, 111*, 1.0.0.2), 100* is stored in destination table and is associated with the 1st row, and 111* is stored in source table and associated with the 1st column. In the TD-table, the cell (100*, 111*) that corresponding to 1st column and 1st row has index value 2. In the mapping table, the next hop that is related with index value 2 is 1.0.0.2.

To provide better connectivity, each destination prefix is associated with one or more default next hops. If no source prefix matches the source address of a packet, then routers will forward the packet to the default next hop associated with the matched destination prefix. The default next hop can be seen as a string composed of wildcards (**** in Fig. 8). Thus, for any arrived packet, there will be at least one source prefix that matches its source address.

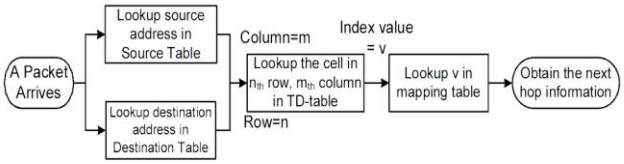


Figure 9. Lookup action in FIST.

The lookup action lookup (d, s) is shown in Fig. 9. When a packet arrives, the router first extracts the source address s and destination address d. Using the LMF rule, the router finds the matched source and destination prefixes in both source and destination tables that reside in the TCAMs. According to the matched entry, the

source table will output a column address and the destination table will output a row address. Combined with the row and column addresses, the router can find a cell in the TD-table, and return an index value. Using the index value, the router looks up the mapping table, and returns the next hop that the packet will be forwarded to.

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

```
1 begin
2   foreach p_d, p_s do
3     if ∄(p_d, p_s, a) ∈ R then
4       S = {(p̄_s, p̄_d, ā) ∈ R | p̄_d = p_d}
5       S' = {(p̄_s, p̄_d, ā) ∈ S | p̄_s is a prefix of p_s}
6       Find (p̂_s, p̂_d, â) ∈ S', ∀(p'_d, p'_s, a') ∈ S', p'_s is a prefix of
        p̂_s
7       Fill the cell (p_d, p_s) with â.
```

Filling up the TD-cells: For the example in Fig. 8, if a packet with destination address 1011 and source address 1111 arrives at the router, rule (101*, 11**, 1.0.0.2) should be matched. This is because according to LMF rule, the destination prefix 101* should be first matched. There are two rules (including the default rule) associated with the destination prefix 101*. Consequently, source prefix 11** will be matched. With the new structure, destination prefix 101* will be matched and source prefix 111* will be matched. However, the cell (101*, 111*) (2nd row and 1st column) in TD-Table does not have any index value. Intrinsically, consider a packet that should match destination and source prefix pairs (pd, ps). If there exists a source prefix ps' that is longer than ps, cell (pd, ps') rather than (pd, ps) will be matched.

To address the problem, we should pre-compute and fill the conflicted cells (such as (101*, 111*) in the above example) with appropriate index value. We develop algorithm TD-Saturation() to resolve this confliction.

Theorem 1: FIST can correctly handle the rule defined in Definition 1.

Proof: When a packet arrives, and matches the cell (pd, ps) according to our rule. If cell (pd, ps) is set with an index value. Then this cell stores the rule that should be matched.

Else if cell (pd, ps) does not have an index value. Then according to the first step in the above algorithm, S contains all the rules given pd is matched. In the second step, $\tilde{ps}$ is a prefix of ps, thus the packet also match the rule ($\tilde{ps}$,$\tilde{pd}$,$\tilde{a}$) belong to S. Because there does not exist ($\hat{ps}$,$\tilde{pd}$,$\hat{a}$) belong to S where $\hat{ps}$ is longer than $\tilde{ps}$, $\tilde{ps}$ is the longest match among all the rules given pd is matched. So the cell (pd, ps) should be set with $\tilde{a}$ according to Definition 1.

	1	0	1	1	1
	0	2	0	0	0
	1	2	1	1	2
	2	2	2	2	2
	2	3	2	2	3

Figure 10. TwoD array after setting all conflicted cells.

We show the TD-table after filling up all the conflicted cells in Fig. 10. The above algorithm guarantee the correctness of FIST, however, the algorithm requires all unset cells to be re-computed during an update. The update frequency of Internet router is hundreds per second [23], thus we need an incremental update algorithm to avoid resources over-consumed on updating.

We can construct a tree using all source prefixes in the source table. In the tree, ps is the ancestor of ps' if and only if ps is a prefix of ps'. Continue the example in Fig. 8, we show the tree in Fig. 11. To fill up the unset cell (pd, ps), for the node ps' in the colored tree, we set it to be black if there is a rule associated with pd and ps', else set it to be white. We call the tree colored tree, because the colored trees associated with the same destination are the same, we use CT (pd) to denote the colored tree associated with pd. Then we should fill the cell (pd, ps) with the index value of (pd, $\tilde{ps}$) where $\tilde{ps}$ is the highest level (suppose the level of root node is 0) black ancestor node in CT (pd). Fig. 11 shows the colored tree for filling up (101*, 111*), the highest level black ancestor node of 111* is 11**, so (101*, 111*) should be filled with 2, which is the index value of (101*, 11**).

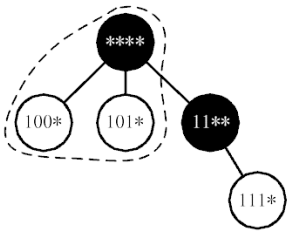


Figure 11. Colored tree CT(101*) for Fig. 8.

With the colored tree, FIST supports incremental update, such that we do not have to re-compute the index value of an unset cell during an update. When the index value (pd, ps) changes, only the largest sub-tree (of CT (pd)), that rooted at ps and does not include black node (except the root node itself), has to change correspondingly. For example, when the index value of (101*, ****) changes, only the sub-tree inside the dashed line, which includes (101*, ****), (101*, 100*) and (101*, 101*) have to be reset correspondingly.

Theorem 2: The TCAM storage space of FIST is O(N +M) bits. The SRAM storage space of FIST is O(N × M) bits. The look up speed is one clock cycle of TCAM plus three clock cycles of SRAM. The update speed is O(M) clock cycles of SRAM.

TABLE I: COMPARISON OF DIFFERENT STORAGE STRUCTURE [28] [29] [30] [31]

Metrics	Structure		
	Regular	Bold	Italic
Maximum clock rate (MHz)	400	266	125
Maximum storage space (Mb)	144	36	576
Price ($/Mb)	≈10-50	200-259	1
Power Consumption(watts/Mb)	≈0.1	15	≈0.1

Proof: In FIST, the destination table has N entries, and the source table has M entries. Each entry consumes w bits, where w is the width of each TCAM entry. Thus the TCAM storage space is O(N +M) bits. The SRAM storage is dominated by the TD-table, which has N rows and M columns, so the total storage space is O(N ×M) bits.

During a lookup process, the router will access source table (in TCAM) and destination table (in TCAM) for one time. The source and destination tables can be accessed in parallel. Thus one clock cycle of TCAM is enough. With the outputs of source and destination tables, we can obtain the row and column addresses in SRAM, which can be done within one SRAM clock cycle in a dual-port SRAM. Then the router will access the TD-table, and mapping table next hop. In total, the look up process speed is one clock cycle of TCAM plus three clock cycles of SRAM.

An update process, in the worst case, will cause updating on all cells in a row of TD-table. For example, if we update (11 **, ****) in Fig. 8 with index value 1, then all cells in the 3rd row should be updated with index value 1.

As a comparison, the traditional destination-based routing usually stores destination prefixes in one TCAM, and accesses both TCAM and SRAM for one time during a lookup process. Since the speed of current SRAM is very fast, (the maximum clock rate of SRAM can reach 400MHz), the look up speed is roughly the same with destination-based routing.

FIST greatly reduces the TCAM storage space, and achieves fast lookup speed. Although FIST increases the SRAM storage space, the size of SRAM is much larger than TCAM. current largest available SRAM chip in the market can reach 144Mb/chip [24]. Besides, other memory products can be used to take the place of SRAM, such as RLDRAM (Reduced Latency DRAM), that has 576Mb/chip and takes only 8ns during a random access [25].

FIST increases the number of accesses to memory during updating. However, the problem can be alleviated with the following two facts: 1) the frequency of updates is much slower (about two orders of magnitude) than lookup [26]; 2) most prefixes in the current routing tables are 24 bits prefixes [27], indicating that most black nodes are leaf nodes in colored trees, and update on leaf node will not cause updates on other nodes.

VI. PORPT: ROUTING PROTOCOLS DESIGN

The TwoD-IP architecture provides great opportunities and flexibility for the ISPs to deploy routing protocols for different purposes. In this section, we design a policy routing protocol PORPT (Policy Routing Protocol for TwoD-IP), which illustrates an example for a TwoD-IP routing protocol.

PORPT is designed to satisfy real load balancing demands for CERNET2. CERNET2 has two international exchange centers (one is in Beijing, CNGI-6IX, and the other one is in Shanghai, CNGI-SHIX) connected to the foreign Internet (see Fig. 12). During daily operation, we find that the CNGI-6IX is very congested (with an average throughput of 1.18Gbps in February 2011), while the CNGI-SHIX has much more spare capacity (with a maximal throughput of 8.3Mbps in February 2011). Thus CERNET2 desires to move part of the traffic from CNGI-6IX to CNGI-SHIX.

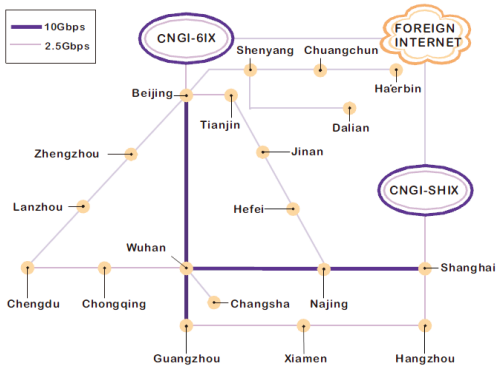


Figure 12. CERNET2 topology.

Our goal is to divert traffic from some specified customer network to any edge router. For example, in Fig. 13, customer networks are connected to ISP network through provider edge routers (PE routers, e.g., PE0 and PE1), and ISP network is connected to Internet through edge routers (e.g., E0, and E1). Besides the PE and edge routers, there are other routers (P routers, e.g., I0, I1, I2, I3) in the network. Currently, traffic from customer networks to the Internet all passes through E0. The objective of the ISP is to move the traffic from PE1 towards the Internet to E1.

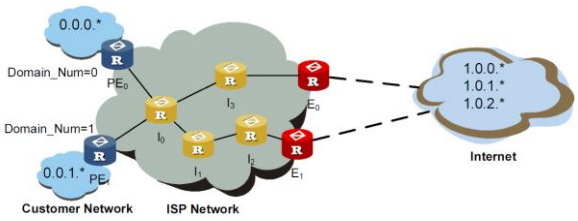


Figure 13. Example of Policy Routing

We design an intra-domain routing protocol combined with OSPF. Additional information is disseminated and received through extensions of OSPF [32]. We clarify a few aspects about the protocol description:

- Edge routers have the information of foreign Internet prefixes through inter-domain protocols like BGP.
- PE routers have the domain numbers of the customer networks that attach to them. PE routers have the information about the preference of the customer network on the edge routers. This can be obtained through manual configuration or automatical selection (e.g., selecting the edge router that has lower utilization).

With these conditions, edge router will announce foreign Internet prefixes information to intra-domain,

including the identity of the edge router itself. The PE router will announce its preferences on edge routers, and the binding information between its customer prefixes and customer domain number. The routers of the ISP can compute the TwoD-IP forwarding table based on these information. We first describe the PORPT protocol details and then describe how to transform the information to the two dimensional routing table.

Let Foreign_Prefix be a foreign Internet prefix, Customer_Prefix be a customer prefix, Router_ID be the IP address of a router and Domain_Num be the domain number of a customer network. We define three types of messages:

- Announce(Foreign_Prefix,Router_ID): This message is sent by an edge router of Router_ID to announce an Internet prefix IP_Prefix.
- Bind(Customer_Prefix,Domain_Num): This message is sent by a PE router to announce the binding information between a customer prefix and domain number of this customer network
- Pref(Domain_Num,Router_ID): This message is sent by a PE router, to announce the preference for a customer network on an edge router.

Fig. 14 shows the time line of PORPT. The edge routers just have to announce the foreign Internet prefixes combined with its own router identification to intra-domain, and does nothing else unless it is also a member of PE or P routers. The PE routers have to announce the binding between its customer domain number and customer prefixes, PE routers also have to announce the preferences on edge routers. After obtaining the foreign Internet prefixes and preferences of customer networks, both PE and P router should compute the two dimensional routing table, which include a set of 3-tuple rules defined in Section V. Combined with traditional routing table, two dimensional routing table can be transformed to two dimensional forwarding table. We present a algorithm for computing the routing table as follows.

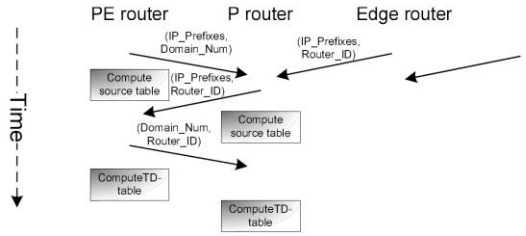


Figure 14. Time line of the policy routing protocol.

For example, in Fig. 13, PE router PE0 will announce the binding information by sending Bind(0.0.0.*, 0), PE1 will announce Bind(0.0.1.*, 1). Edge router E0 will announce three foreign Internet prefixes combined with its own identification by sending Announce(1.0.0.*,E0), Announce(1.0.1.*,E0), Announce(1.0.2.*,E0), E1 will announce Announce(1.0.0.*,E1), Announce(1.0.1.*,E1), Announce(1.0.2.*,E1). At last, PE1 will announce Pref(1,E1). Receiving these messages, PE and P routers can construct the two dimensional routing tables, we show the routing table on router I0 in Table II.

Algorithm 2: Compute-RIB()

```
Output : R
Initialize: R ← ∅
1 begin
2     foreach Pref(Domain_Number, Router_ID) do
3         NH ← Lookup_Next_Hop(Router_ID)
          // Lookup the next hop towards Router_ID
4         foreach Announce(Foreign_Prefix, Router_ID) do
5             NH' ← Lookup_Next_Hop(Foreign_Prefix)
6             if NH = NH' then
7             foreach Bind(Customer_Prefix, Domain_Num)
              do
8                 R = R ∪
                  {(Foreign_Prefix, Customer_Prefix, NH)}
```

We have developed a prototype of the protocol, and set up a small scale test under VegaNet [33], a high performance virtualized testbed.

TABLE II: TWO DIMENSIONAL ROUTING TABLE ON THE P ROUTER I0

Destination Prefix	Source Prefix	Next hop
1.0.0.*	0.0.1.*	I1
1.0.0.*	0.0.1.*	I1
1.0.0.*	0.0.1.*	I1

A. Deployment

It is widely known that making changes to the network layer is notoriously difficult. We consider two important problems in the deployment. First, during the deployment, the proposed protocols should have small impact on the Internet protocols and infrastructure. Second, at the initial stage, a node-by-node incremental deployment scheme is highly preferred to minimize error and support efforts.

Currently, we mainly focus on a node-by-node incremental deployment scheme. We consider the most important factor for the success is that the deployment should have visible benefits after each node is deployed. We have a separate study on this problem [34]. The key investigated problem is that without full deployment, the resulting paths for traffic from some sources may deviate from the required ones, (i.e., pre-defined by users or ISP providers), then how to find node deployment sequences to minimize the deviation. We rigidly defined the deviation and mathematically formulated the problem.

We developed several algorithms for different practical scenarios and a case study on CERNET2. Our main observation is that we can gain the majority of the performance when only a small percentage of carefully selected nodes are deployed.

VII. DISCUSSION ON PROBLEMS TO SOLVE IN THE FUTURE

With our forwarding table design FIST, routing protocol PORPT, and deployment study, we believe the TwoD-IP routing architecture can work. Nevertheless, there are still many problems to solve so that TwoD-IP

routing can work better. We discuss a few that we consider the most urgent problems.

A. Forwarding Table Improvement

1) Forwarding Table Storage: Our new TwoD-IP forwarding table structure FIST is composed of TCAMs and SRAMs. Current largest available TCAM chip on the market has 36Mb [30], and can accommodate 1 million IPv4 prefixes. The latest reported destination-based forwarding table size is 400,000 [35]. Thus we believe TCAM storage space is large enough for TwoD-IP routing.

FIST consumes O(N × M) SRAM storage space. This is enough for CERNET, which has only 6493 destination prefixes. However, this might be large when both N andM are large. For example, when both N and M are 10000, the TD-table consumes 800Mb, which is too large for current SRAM storage.

We see the following directions promising, 1) Note that SRAM is not highly customized, compared to TCAM. Thus we can use various techniques to compress SRAM space; 2) In our FIST structure, each row occupies a row in the TD-table, in practice, we only need to divert traffic for a small part of the destination prefixes, rather than all. Thus, we can divide the destination table into two parts, each prefix in the first part points to a row in TD-table, each prefix in the second part points directly to an index value.

2) Forwarding Table Update: Forwarding table will be updated mainly due to two reasons: 1) topology change, which will incur destination address-oriented update and 2) policy change, which may incur source address-oriented update. Topology change will have the same update time as in the current Internet. For policy change, we believe that there does not need a frequent and on-time update. Therefore, for such update, it can be carried out during intervals when the routers have lower load. Nevertheless, we consider it is still necessary to give a deeper study on this issue.

B. Protocol Properties for TwoD-IP Routing

We only developed a simple protocol. The current Internet intra-domain protocol naturally avoid loops. We need to study protocol properties for the policy routing, and develop a mathematical foundation to avoid loops and conflicts. Several existing studies may be helpful to picture out the guidelines; for example, general algebra [36], and valley-free properties [37] [38].

VIII. Conclusion and Future Work

We presented the TwoD-IP architecture, which is closely combined with source address during routing. With TwoD-IP, the semantics that the routing system can provide are greatly enriched. There are also great challenges that we should face during designing and implementing TwoD-IP. In this paper, we described our initial design for TwoD-IP.

Acknowledgment

The research is supported by the National Basic Research Program of China (973 Program) under Grant 2009CB320502, the National Natural Science Foundation of China (61073166 and 61161140454),the National High-Tech Research and Development Program of China (863 Program) under Grants 2011AA01A101, the National Science & Technology Pillar Program of China under Grant 2011BAH19B01.

References

[1] M. Handley, "Why the internet only just works," *BT Technology Journal*, vol. 24, pp. 119–129, 2006.

[2] E. Chen and S. Sangli, "Avoid BGP best path transitions from one external to another," *Internet Engineering Task Force*, Sep. 2007.

[3] D. Estrin, T. Li, Y. Rekhter, K. Varadhan, and D. Zappala, "Source demand routing: Packet format and forwarding specification (Version 1)," *Informational Internet Engineering Task Force*, May 1996.

[4] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," *Internet Engineering Task Force*, Jan 2001.

[5] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. ACM SOSP'01*, Banff, Canada, October 2001.

[6] J. Wu, J. Bi, M. Bagnulo, F. Baker, and C. Vogt, "Source address validation improvement framework," *Internet Draft*, Mar 2011.

[7] *Private Network-Network Interface Specification v1. 0(PNNI)*, ATM Forum Technical Committee, Mar. 1996.

[8] C. Perkins, "IP Encapsulation within IP," *Internet Engineering Task Force*, Oct. 1996.

[9] S. Yasukawa, A. Farrel, and O. Komolafe, "An analysis of scaling issues in MPLS-TE core networks," *Informational Internet Engineering Task Force*, Feb. 2009.

[10] C. Metz, C. Barth, and C. Filsfils, "Beyond mpls-less is more," *Internet Computing, IEEE*, vol. 11, no. 5, pp. 72 –76, Sep 2007.

[11] *Policy-Based Routing (white paper)*, Cisco, 1996.

[12] J. Fu and J. Rexford, "Efficient ip-address lookup with a shared forwarding table for multiple virtual routers," in *Proc. ACM CoNEXT Conference*, Madrid, Spain, Dec 2008.

[13] P. Laskowski, B. Johnson, and J. Chuang, "User-directed routing: from theory, towards practice," in *Proc. ACM CoNEXT Conference*, Seattle, WA, USA, Aug 2008.

[14] *Multi-topology routing (White Paper)*, Juniper, Aug 2010.

[15] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial f service attacks which employ ip source address spoofing," *Internet Engineering Task Force*, May 2000.

[16] C. De Launois and M. Bagnulo, "The paths toward ipv6 multihoming," *IEEE, Communications Surveys Tutorials*, vol. 8, no. 2, pp. 38 –51, 2006.

[17] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C. Diot, "Characterization of failures in an operational ip backbone network," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 749–762, 2008.

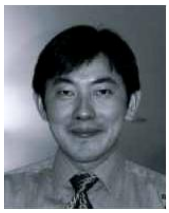
[18] M. Suchara, D. Xu, R. Doverspike, D. Johnson, and J. Rexford, "Network architecture for joint failure recovery and traffic engineering," in *Proc. ACM SIGMETRICS'11*, San Jose, CA, Jun 2011.

[19] A. Atlas and A. Zinin, "Basic specification for IP fast reroute: loop- free alternates," *Internet Engineering Task Force*, Sep. 2008.

[20] X. Liu, X. Yang, and Y. Lu, "To filter or to authorize: Network layer dos defense against multimillion-node botnets," in *Proc. ACM SIGCOMM'08*, Seattle, WA, USA, Aug 2008.
[21] P. Gupta and N. McKeown, "Packet classification on multiple fields," in *Proc. ACM SIGCOMM'99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Cambridge, Massachusetts, United States, Aug 1999, pp. 147-160.
[22] A. Hari, S. Suri, and G. Parulkar, "Detecting and resolving packet filter conflicts," in *Proc. IEEE INFOCOM'00*, Tel-Aviv, Israel, Mar 2000.
[23] J. Zhao, X. Zhang, X. Wang, Y. Deng, and X. Fu, "Exploiting graphics processors for high-performance ip lookup in software routers," in *Proc. IEEE INFOCOM'11*, Shanghai, China, Apr 2011.
[24] Cypress Semiconductor. [Online]. Available: http://www.cypress.com
[25] Rldram Memory Technology. [Online]. Available: http://www.micron.com/products/dram-/rldram-memory
[26] T. Mishra and S. Sahni, "Duos-simple dual tcam architecture for routing tables with incremental update," in *Proc. IEEE ISCC'10*, Riccione, Italy, Jun 2010.
[27] A. Basu and G. Narlikar, "Fast incremental updates for pipelined forwarding engines," *IEEE/ACM Trans. Netw.*, vol. 13, pp. 690–703, 2005.
[28] W. Jiang, Q. Wang, and V. Prasanna, "Beyond tcams: An sram-based parallel multi-pipeline architecture for terabit ip lookup," in *Proc. IEEE INFOCOM'08*, Phoenix, AZ, Apr 2008.
[29] A. Liu, C. Meiners, and E. Torng, "Tcam razor: A systematic approach towards minimizing packet classifiers in tcams," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 490 – 500, 2010.
[30] Y. Chiba, Y. Shinohara, and H. Shimonishi, "Source flow: handling millions of flows on flow-based nodes," in *Proc. ACM SIGCOMM'10*, New Delhi, India, Sep 2010.
[31] Router Fib Technology. [Online]. Available: http://www.firstpr.com.au/ip/sram-pforwarding/router-fib/
[32] A. Zinin, A. Roy, L. Nguyen, B. Friedman, and D. Yeung, "OSPF link local signaling," *Internet Engineering Task Force*, Aug. 2009.
[33] C. Wenlong, X. Mingwei, Y. Yang, L. Qi, and M. Dongchao, "Virtual network with high performance: Veganet," *Chinese Journal of Computers*, vol. 33, no. 1, pp. 63–73, 2010.
[34] S. Yang, D. Wang, M. Xu, and J. Wu, "Efficient two dimensional-ip routing: An incremental deployment design," Tsinghua University, Tech. Rep., July 2011.
[35] Bgp Routing Table Analysis Reports. [Online]. Available: http://bgp.potaroo.net.
[36] J. Sobrinho, "Algebra and algorithms for qos path computation and hop-by-hop routing in the internet," *IEEE/ACM Trans. Netw.*, vol. 10, pp. 541–550, 2002.
[37] L. Gao and J. Rexford, "Stable internet routing without global coordination," in *Proc. ACM SIGMETRICS'00*, Santa Clara, CA, Jun 2000.
[38] Y. Liao, L. Gao, R. Gu ́erin, and Z.-L. Zhang, "Safe interdomain routing under diverse commercial agreements," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 1829–1840, 2010.

Mingwei Xu received the B.Sc degree in 1994 and Ph.D. degree in 1998 both from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. Now he is a professor in Tsinghua University. His research interests include computer network architecture, Internet routing and high-speed router architecture. He is a member of IEEE.

Shu Yang received the B.Sc degree from the Department of Computer Science and Technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2009. He is current a Ph.D. candidate in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His current research interests include Internet Routing and High Performance Router Design.



Dan Wang received the B.Sc degree from Peking University, Beijing, China, in 2000, the M.Sc degree from Case Western Reserve University, Cleveland, Ohio, USA, in 2004, and the Ph. D. degree from Simon Fraser University, Burnaby, B.C., Canada, in 2007; all in computer science. He is currently an assistant professor at the Department of Computing, The Hong Kong Polytechnic University. His research interests include wireless sensor networks, Internet routing and applications. He is a member of the IEEE.

Jianping Wu received his B.S., M.S., and Ph.D. from Tsinghua University. He is a Full professor and director of Network Research Center, Ph.D. Supervisor of Department of Computer Science, Tsinghua University. From 1994, he has been in charge of China Education and Research Network (CERNET). He is a member of Information Advisory Committee, Ofce of National Information Infrastructure, Secretariat of State Council of China, and is also a vice president of Internet Society of China (ISC). His research interests include next generation Internet, IPv6 deployment and technologies, Internet protocol design and engineering. He is an IEEE fellow.