A Novel Approach to Interior Gateway Routing

Yoshihiro Nozaki, Parth Bakshi, and Nirmala Shenoy
College of Computing and Information Science
Rochester Institute of Technology
Rochester, NY, USA
{yxn4279, pab8754, nxsvks}@rit.edu

Abstract— **Most ISPs and Autonomous Systems (AS) on the Internet today use Open Shortest Path First (OSPF) or Intermediate-System-to-Intermediate-System (IS-IS) as the Interior Gateway Protocol (IGP). Both protocols use the Link-State routing approach and require distribution of link state information to all routers in a network or in an area. Any topological changes require redistributing updates and refreshing routing tables. This results in high convergence times. During convergence, packet routing becomes unreliable. During the years as network sizes have grown, the routing table sizes have also exhibited a linear growth. This is indicative of scalability issues in the current routing approaches and could be a limiting factor for future growth. Future Internet initiatives, which were started worldwide almost a decade ago, have enabled novel approaches to address the routing problem. In this article, such a novel interior gateway routing approach is presented. The approach leverages the tiered structure existing among ISP networks, AS, and in general most networks. The routing protocol was thus named the Tiered Routing Protocol (TRP). Though TRP can be used for both inter- and intra-AS routing, in this article, it is presented as a candidate protocol for intra-AS routing. TRP operations are supported by a tiered addressing scheme. Use of TRP replaces both the Internet Protocol (IP) and the routing protocol. The rationale for TRP and its details followed by its evaluation over the US national testbed namely Emulab are presented in this article. TRP's performance is compared with OSPF to highlight its major contributions to address routing scalability.**

Keywords-Intra-domain Routing; Network Convergence; Internetworking Architectures; Tiered Architectures; Routing Table sizes; Interior Gateway Routing

I. INTRODUCTION

This article is an extension of the conference paper [1] aiming at providing a detailed analysis of ISP network and transition platform between Internet Protocol (IP) routing and the proposed routing protocol.

IP provides best effort reachability for communication across networks and nodes connected to the Internet. In IP networks, routers use routing protocols to discover and maintain routes and also to recover from route failures. Routing tables maintained by current routing protocols increase almost linearly with increase in network size and is an unhealthy trend indicating scalability issues, which can manifest as performance degradation. Also, the time taken for the network to adapt to topological changes increases with increase in network size resulting in higher convergence times during which routing is unpredictable and unstable. With more and more users connecting to networks today, this poses a serious problem. Patch and evolutionary solutions have been and are being proposed and implemented to address the problem both at the inter domain and intra domain level [2][3].

Interior Gateway Protocols (IGP) such as Routing Information Protocol (RIP) and OSPF were designed to work with IP. RIP is a distance vector (DV) protocol and can be used in networks with a maximum diameter of 15 hops. Large ISP networks thus use Link-State (LS) IGPs such as IS-IS or OSPF, which uses the area concept to segment networks into manageable size. LS routing protocols require periodic updates and redistribution of updates to all routers in the network or in an area on link state changes. Each router running the LS routing protocol executes the Dijkstra's algorithm on the collected link state information to populate routing tables. Dissemination of network-wide (or area-wide) link state information also adversely impacts scalability and convergence times in the networks using OSPF. In some cases, the physical location of areas requires use of virtual links to the backbone area further limiting the versatility of OSPF.

A primary contribution in this work is to decouple the dependency of routing table sizes from the network size. However, this had to be found on a solution that would also be acceptable to the Internet service provider community. Thus, the proposed routing protocol adopts an internetworking model that derives from the structures used by ISPs to define their business relationships namely the *tiers*. The routing protocol so proposed is called the *tiered routing protocol* (TRP). A new tiered addressing scheme to enable efficient operation of the TRP was also introduced. The tiered address inherits attributes of the tiered structures and expresses them explicitly in the address to be used for TRP operation and packet forwarding. To decouple dependencies between connected network entities, and enable their easy movement and connections to other entities a nesting concept is introduced [4].

Traditionally, the Internet Protocol was designed to provide logical addresses, application transparency and forwarding of packets based on routing table entries. A routing protocol was thus needed to populate the routing tables. For this purpose, different types of routing algorithms and protocols based on these algorithms were developed. Examples are distance vector algorithm based RIP, link state algorithm based OSPF and path vector algorithm based Border Gateway Protocol (BGP). TRP has been designed to replace both the IP and routing protocol. This is true for inter- and intra-AS routing. Thus, interworking functions and

complexities due to the interworking of two protocols are reduced.

In this article, the rationale and detailed operation of TRP as an IGP are described. TRP has also been evaluated and its performance compared with OSPF using the US national testbed – Emulab [4]. In this article, TRP is applied to an AS and the process of identifying tiers, tiered address allocation, population of routing tables, packet forwarding and failure handling is described. TRP implemented in an AS also provides the network setting for performance evaluation and comparison with OSPF. To provide a more comprehensive performance evaluation, the following metrics were evaluated: initial convergence times, convergence times after link failures, routing tables sizes, and control overhead during initial convergence and convergence after link failure.

Section II describes related work for the reduction of convergence times in IGPs. Section III describes the two routing protocols namely OSPF and TRP. The description is limited to the performance studies targeted, the convergence times, routing table sizes and control overhead. Some foundational studies, which led to the proposal of the tiered routing approach, are also included in this section. Related work on TRP is covered more extensively as compared to OSPF as details about OSPF are RFC standards [2]. Section IV discusses the use of Multi Protocol Label Switching (MPLS) as a transition facilitator and the mechanisms that can enable a successful transition. Section V provides details of the emulations tests and the techniques adopted to collect results in the Emulab testbed. The two AS topologies evaluated are also described. Section VI provides the averaged results collected from several emulation runs over several test sites. This is followed by a detailed analysis of the results for both TRP and OSPF operation. In Section VII, the conclusions and intended future work are discussed.

II. Related Work

Significant research effort can be noticed towards improving and enhancing IGP performance. Some these efforts were directed to the reduction and optimization in IGP convergence time subsequent to link state changes in the network or area. Work in this regard can be broadly categorized into: (a) reducing failure detection time and (b) reducing routing information update time.

A. Reduction in Failure Detection Time

Layer-2 notification is used to achieve sub-second link/node failure detection. However, this relies on types of network interfaces and does not apply to switched Ethernet [6].

Layer-3 notification is the more adopted method for link failure detection. For this purpose, the *Hello* protocol is used. The hello protocol, besides being used to disseminate neighbor information, is also used to identify link/node failure in many routing protocols and is the layer-3 failure detection mechanism. OSPF sends *hello* packets to adjacent routers at an interval of 10 sec by default. The hello packet contains information on all links that a router is connected to. On missing four *hello* packets consecutively from a neighbor, OSPF routers recognize an adjacency failure with that neighbor router. Reducing *hello* packet interval time to sub-seconds can significantly reduce the failure detection time, but at the expense of increased bandwidth usage due to increase in the number of periodic *hello* packets. Increased number of hello packets in a short interval can also increase possibility of route flaps.

B. Reduction in Link State Propagation Time

Although link/node failure detection time can be reduced to sub-seconds, propagating the link status to all routers in the network takes time and is dependent on the network size.

To reduce such delays, an approach that suggests the use of several pre-computed back up routing paths was proposed. Pan et al. [7] proposed the MPLS based on a backup path to reroute around failures. However, having all possible MPLS back up paths in a network is not efficient. Multiple Routing Configurations (MRC) [8] uses a small set of backup routing paths to allow immediate packet forwarding on failure detection. A router in MRC maintains additional routing information on alternative paths. However, MRC guarantees recovery only from single failures. Liu at el. [9] proposed the use of pre-computed rerouting paths if the same can be resolved locally. Otherwise, multi-hop rerouting path had to be set up by signaling to a minimal number of upstream routers. Another approach limits the propagation area of link state update after failure. Narvaez [14] proposed limited flooding to handle link failures. When a link failure occurs, the descendants of the failed link in the shortest path tree are determined and the new shortest path without the failed link is calculated. Then, the updated information is propagated in only the area of descendant nodes.

The two delays discussed above are significant. However, the SPF recalculation time can also be almost a second in large networks [6]. As packet loss/delay or routing loops occur during convergence, it is important to reduce this time. Novel routing approaches under the future Internet initiatives thus provide the opportunity to view the routing problem from a fresh perspective and thus design solutions that are not constrained by the current architectures or implementations.

III. Routing Protocols and Operations

In this section, we describe the operations of the two protocols studied in this article namely OSPF and TRP. In the case of OSPF, only a few basic operations necessary to explain the performance metrics are presented. Details of OSPF operation are publicly available in the Related RFC documents [2].

TRP operation is explained in detail for intra-domain routing. This includes implementing tiered structures within an AS, tiered address allocation to devices in the tiers, routing table population and maintenance with TRP, and the packet forwarding algorithm and link failure handling. Some properties of the tiered address, which makes TRP robust and a few TRP features that result in low convergence times and small routing table sizes are also discussed.

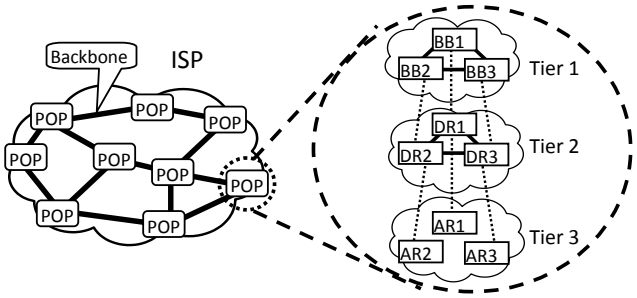


Figure 1. Tiered Topology within an ISP

A. Open Shortest Path First (OSPF)

Link State routing protocols offer faster convergence with theoretically no hop and no network size limitations compared to distance vector (DV) based routing protocol. The small sized update packets consume less bandwidth, as compared to the DV protocols. However, the update packets have to reach all routers in the network or area for successful convergence and stability in routing tables. The routing table update using Dijkstra's algorithm is complex and CPU intensive if the number of entries in the link state database is high [2].

Basic operations of OSPF include: (a) establishing adjacencies with neighbor routers and electing a Designated Router (DR) and a Backup DR (BDR); (b) maintaining Link State Database (LSDB) and; (c) executing the Dijkstra's algorithm on the LSDB to populate the forwarding database or routing tables. These operations are invoked during startup and also when there are link state changes. Convergence in the two cases is impacted differently and thus described separately below.

1) Initial Convergence in OSPF

a) Establishing Adjacencies: OSPF starts by establishing adjacencies with direct neighbor routers using the *Hello* protocol. *Hello* packets are sent on each interface using a multicast address to neighbor routers. Once *Hello* packets are exchanged, each router recognizes if they are connected via a point-to-point network or a multi-point network such as Ethernet, where several routers are in the same subnet. In the case of multi-point networks, OSPF will elect a DR and a BDR using the router *prioity* and the router *ID*. This is necessary to reduce the number of adjacent direct neighbors and the traffic to establish / maintain them.

b) Maintaining Link State Databases: Hello protocol is also used for link state check between established adjacent neighbor routers. On link state establishment as routers come up, distribution of adjacency information to all routers is initiated by flooding Link State Advertisements (LSA). Each router records all the recived link state information that was flooded in a LSDB.

c) Populating Routing Tables: Using the topology information in the LSDB, each router then locally computes the shortest paths from itself to all other routers in the network (or area), using the Shortest Path First (SPF) or Dijkstra algorithm to populate the routing tables or Forwarding Information Bases (FIB).

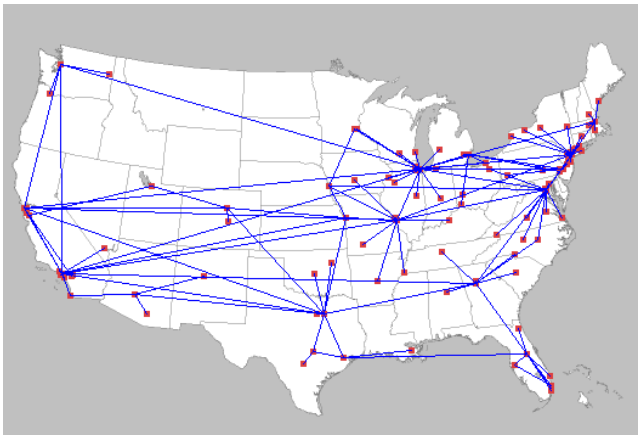


Figure 2. AT&T POP Level Network in the US

2) Convergence After Link / Node Failures

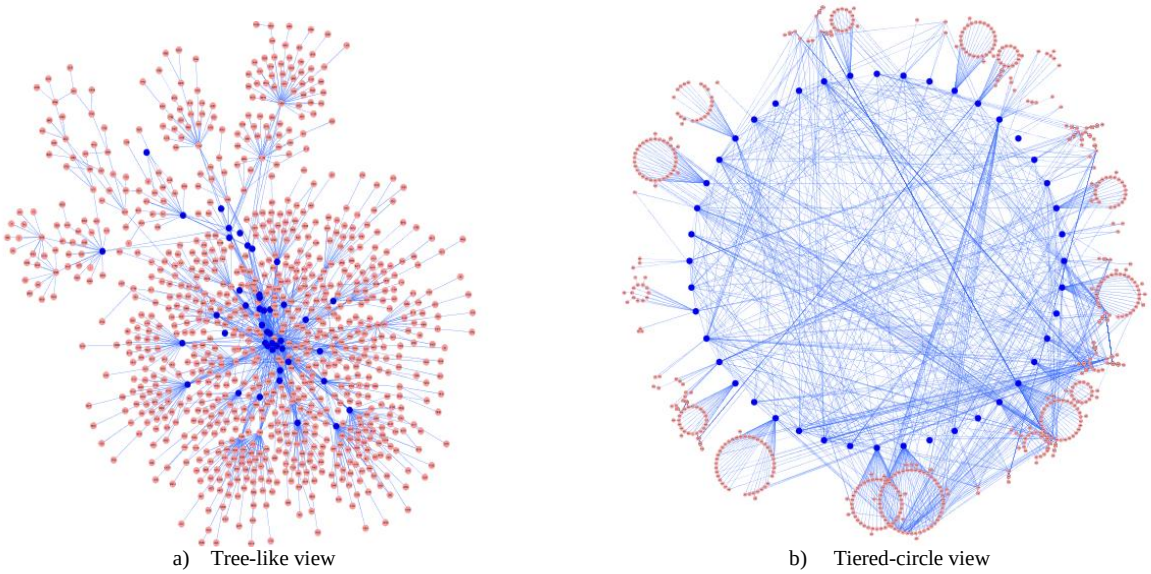
a) Failure Detection: Missing 4 consecutive *Hello* packets from a neighbor indicates link or router failure on that link and hence is one mechanism for failure detection. This is the layer-3 failure detection mechanism and has been adopted predominantly.

b) LSA Propagation: Subsequent to a failure detection, a router generates new LSAs. The LSAs have to be propagated to all routers in the network (area). The time for generating new LSAs on a single failure is between 4 to 12 milliseconds (ms) [9]. OSPF specifies that LSAs cannot be generated within 5 seconds from the last LSA generation time. This provides sufficient time to update the LSDB from the last event and run the Dikjstra alogorithm. LSA propagation time also depends on the number of hops between the routers in the network and the processing delay at each router and transmission delay at each hop.

c) SPF Recalculation Time: When new LSAs update the LSDB they trigger new SPF calculations to update the FIB. Two parameters delay SPF calculations; a *delay timer*, which is 5 seconds and a *hold timer*, which is 10 seconds by default. *Delay timer* is the time between the *new LSA arrival time* and *start of SPF calculation time. Hold timer* limits the interval between two SPF calculations.

B. Tiered Routing Protocol (TRP)

The underlying operational principles of TRP derive from the tiered structure existing in our networks today. However, TRP can run on physical meshed network by creating logical tree-like hierarchical topology through the use of Tiered Routing Addresses (TRA). Hence, in this section, we first describe the process adopted to identify tiers in a given network topology. In large ISP and AS networks, there are backbone routers that connect to one another and extend the connectivity to distribution routers. The distribution routers in turn connect to access routers or sub-networks. In this network scenario, the set of backbone routers can be designated as tier 1 routers, the distribution routers would be the routers at tier 2 and the access routers and sub-networks that they connect would be tier 3. This is the tiered structure adopted for implementing TRP within an AS.



a) Tree-like view b) Tiered-circle view

Figure 3. NY-POP Router-level network in AT&T

To understand the extension of the tiered concept to ISP networks, Fig. 1 is used. Fig. 1 shows a 3-tier structure of that can be identified within an ISP network. Typically, inside of an ISP, there are several Point of Presences (POPs), which form the backbone of that ISP. Each POP in turn comprises several routers, some of which are backbone routers that are primarily meant to connect to other backbone routers in other POPs. Inside of an ISP POP, there is a set of backbone (BB) routers as shown in the projected BB cloud (on the right side of the picture), which can be associated to tier 1 within the POP. The BB routers connect to distribution routers (DR), which can be associated to tier 2. The distribution routers in the DR cloud provide redundancy and load-balancing between backbone and access routers (AR). The ARs then connect to customer or stub networks. The ARs and the stub network can thus be associated to tier 3.

1) Validating the Tiered Approach: In this subsection, we validate the use of the tiered approach using the tiered structure adopted in ISP networks. For this purpose, we conducted some studies using data from the Rocketfuel dataset [15]. This dataset has router-level connectivity information of ISPs. From the Rocketfuel dataset, we imported the AT&T router connectivity information using Cytoscape [16] that also helps to visualize AT&T's router-level topology on the US map (this excludes Hawaii and Alaska). The dataset contains not only the connectivity information, but also the router's location (city) information. Thus, we were able to map each router and city in the visualization shown in Fig. 2.

In total, 11,403 routers and 13,689 links interconnecting the routers were identified under this study. Each city in the topology visualization is a POP that has a large number of routers. A total 110 POPs were identified in the AT&T ISP network in Fig. 2. In each POP, routers connecting with routers in other POPs were identified as BB routers.

2) Associating Routers to Tiers: One of the biggest POP in the AT&T ISP network is the New York POP (NY-POP), which has 946 routers. Among these, 44 of them were identified as BB routers that have link(s) to other POPs. NY-POP router-level topology visualized as a tree structure is shown in Fig. 3 (a). The slightly large dots belong to a node (router) in the tree that has numerous branches. These routers are thus ideal candidates to be the BB routers in tier 1. Using Cytoscape, the visualization was changed to the one shown in Fig. 3 (b), where the BB routers now form the inner circle. Routers that are one hop or a maximum of 5 hops from BB routers were identified as the distribution routers (DR). Some DRs had multiple connection to the BB routers. The edge routers are the access routers that were associated to tier 3 in the POP.

Based on the NY-POP topology observation and the studies conducted, we could identify a total 44 BB routers, 542 DR routers, and 360 AR routers. Once the tiered structure has been identified and the routers associated to tiers explicitly, the tiered address can be allocated as described next.

Once tier 1 nodes are identified, an automated Tiered Routing Addresses (TRA) allocation process can be initiated [10]. This process is explained in the next section. Below we discuss some inherent features of the TRA and the resulting impacts on TRP.

3) TRA Allocation: TRA depends on the tier level in a network and carries the tier value explicilty as the first field. The tier levels can be assigned as described above. Routers closer to a backbone or default gateway have lower tier value and routers near the network edge have higher tier value. TRA can be allocated to a *network cloud* (that comprises of a set of routers used for a specific purpose, such as backbone, distributions and so on) or a router. They are however not allocated to a network interface. Network interfaces are identified by port numbers. However, a router or end node can have multiple TRAs based on its connection to several upper tier routers or networks. This helps to support multi homing.

4) TRA Guarantees Loop-Free Routing: The automated TRA allocation starts from a node at a lower value tier to nodes at higher value tiers. The parent node's address (without the tier value) is part of a child node's address and

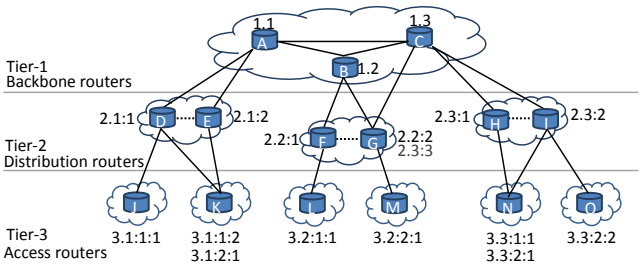


Figure 4. Example Tiered Topology and TRA

precedes a child's unique identifier. As TRAs determine the packet forwarding paths, this feature in a TRA avoids packet looping. However, this dependency can be decoupled at any level through *nesting* without affecting the loop-free packet forwarding.

5) Nested TRA: Let us consider the case where a TRA is assigned to a network cloud. A new tiered structure and TRA can be started for entities within the network cloud, allowing nesting of TRAs. If a network administrator wishes to incorporate clouds in a cloud, nested TRAs can be used where the TRA of an inner cloud does not depend on the TRA of the outer cloud. This decoupling introduces as high level of scalability and flexibility in the internetwork routing operations.

6) Inherent Routing Information: A TRA carries the path information between a lower tier entity and an upper tier entity due to the fact that a child inherits a parent's TRA (without tier value) as part of its address. Thus, a route between two communicating enitites or nodes can be identified by comparing the nodes' TRAs. If a node has multiple TRAs, a sender node may select a communication path based on criteria such as a shorter path or path with better resources.

7) TRP Convergence Time: TRP does not require distribution of routing information due to the inherent route information carried by the TRA. Network convergence in TRP is the time required for direct neighbors to recognize the topology changes in the one-hop neighborhood (in some cases a little more delay may be incurred as infromation may have to propagate down/up a tree branch). However, this time will thus be several magnitudes less than the converegence times experienced by current routing protocols. The extent of information dissemination can also be controlled for optimized operation.

8) TRP Routing Table Size: The packet forwarding decision in TRP is based on next-hop tier level in the direction of packet forwarding, and has only three choices: same tier level, upper tier level, and lower tier level. Thus, the routing table has to be minimally populated with the directly connected neighbor networks / routers. Further optimization is possible by including the two-hop or three-hop neighbors.

C. TRP Operation

Several of the TRP operations such as address allocation, packet forwarding, link / node failure detection / recovery, address re-assignment, and addition / deletion of nodes are explained in this section.

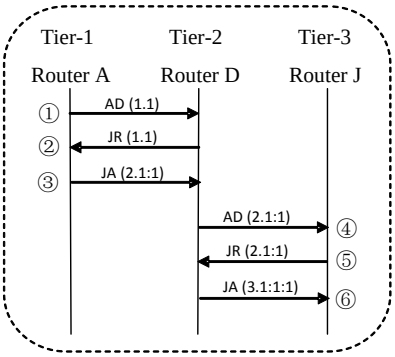


Figure 5. TRA allocation process

TABLE I. ROUTING TABLES OF ROUTER F AND G FROM FIGURE 4

Router F {2.2:1}						Router G {2.2:2, 2.3:3}					
Uplink		*Down*		*Trunk*		*Uplink*		*Down*		*Trunk*	
Port	*Dest*	*Port*	*Dest*	*Port*	*Dest*	*Port*	*Dest*	*Port*	*Dest*	*Port*	*Dest*
1	1.2	3	3.2:1:1	2	2.2:2, 2.3:3	1	1.2	3	3.2:2:1	4	2.2:1
Dest – directly connected neighbor						2	1.3				

1) Address Allocation Process: TRA allows automatic address allocation by a direct upper tier cloud or node. Once tier 1 nodes acquire their TRAs (or have been assigned their TRAs), tier 2 nodes will get their TRA from the serving tier 1 node.

a) TRA Allocation: The process starts from the top tier, i.e., tier 1. A tier 1 node advertises its TRA to all its direct neighbors. A node, which receives an advertisement, sends an address request and is allocated an address. For example in Fig 5, Router A with TRA 1.1 sends Advertisement (AD) packets to Routers B, C, D, and E. Routers D and E send Join Request (JR) to Router A because they do not have a TRA yet. Router B and C do not request address to Router A because they are at the same tier level. Router A allocates a new address (2.1:1) to Router D using a Join Acceptance (JA) packet. Another new address (2.1:2) is allocated to Router E. The last digit of the new address is maintained by the parent router, i.e., Router A. Once Router D registers its TRA, it starts sending AD packets to all its direct neighbors and address assignment continues to the edge routers.

b) Multi-Addressing: If a router has multiple parents, like Router G in Fig. 4, it can get multiple addresses. A router with multiple addresses may decide to use one address as its primary address to allocate addresses to its children routers. This implementation was adopted in the work presented in this article.

2) Routing Tables: TRP maintains three routing tables based on the type of link it shares with its neighbors. In a tiered structure, links between routers are categorized into three different types: up-link that connects to an upper tier router; down-link that connects to a lower tier router; and trunk-link that connects to routers in the same tier level. A router can identify the type of link from which the AD packet arrives by comparing its tier value with the tier value in the received packet.

```
1: if( R.TV == P.TV) then
2:   if( R.TA.the_last_digit == P.TA.the_first_digit) then
3:     if( port_num = find ( P.TA.the_second_digit, down-link table ) ) then
4:       remove( P.TA.the_first_digit);
         P.TV++;
         forward( P, port_num );
         return();
5:     end if
6:   else if( R.TV == 1 ) then  //at Tier1
7:     if( port_num = find ( P.TA.the_first_digit, up-link table ) ) then
8:       forward( P, port_num ),
         return();
9:     end if
10:  end if
11: else if( R.TV– P.TV == 1 && R.TA.the_parent_digit == P.TA.the_first_digit) then
12:   if( port_num = find ( P.TA.the_second_digit, trunk-link table ) ) then
13:     remove( P.TA.the_first_digit);
        P.TV++;
        forward( P, port_num );
        return();
14:   end if
15: else if( R.TV < P.TV) then
16:   discard( P); //wrong packet
      return();
17: end if
18: if( port_num = find( up-link table ) ) then
19:   forward( P, port_num );
      return();
20: end if
21: discard( P);  //no entry in routing tables
    return();
```

Algorithm 1. Packet forwarding at router *R* and incoming packet *P*.

Router F has three different types of links to Routers B, G, and L on port numbers 1, 2, and 3 respectively. Advertisement from Router B is received at port 1 and compared with the tier level of Router B (which is 1) and its own tier level (which is 2). Since tier level of Router B is less than tier level of Router F, the link connected on port number 1 is recognized as up-link and the information is stored in the up-link table. Likewise, information about Router G is stored in the trunk-link table, and information about Router L is stored in the down-link table.

In Table I, the '*port*' column shows the port number of the router and '*dest*' column shows the TRA of direct neighbor obtained from the advertisements. There are multiple entries against a single port in the trunk-link table of Router F because Router G has two TRAs. The routing table for Router G is also shown.

The TRA carries the shortest path information inherently. Hence, initial convergence time in TRP is significantly lower than OSPF because, with one advertisement packet from each direct neighbor, the routing tables converge. This also results in less number of control packets and traffic.

In the network in Fig. 4, three tier levels have been identified, and the TRA for the routers in this network are noted beside them. The TRA is made up of *TV*. *TA*, where *TV* is the tier value to identify the tier level and Tree Addresses (TA) is the address of the router. A '**.**' notation in the tiered address separates a TV and the TA. Thus, the TRA starts with a TV followed by '**:**' separated addresses, which form the TA. Thus, TRA 3.1:1:1 has TV=3 and TA= 1:1:1.

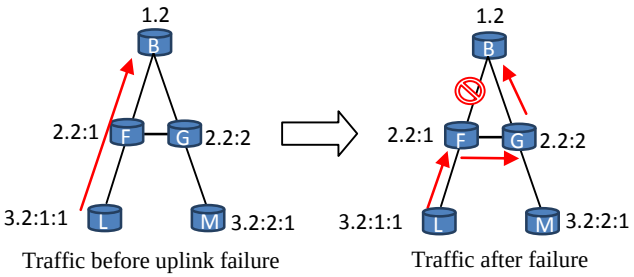


Figure 6. Failure handling with uplink

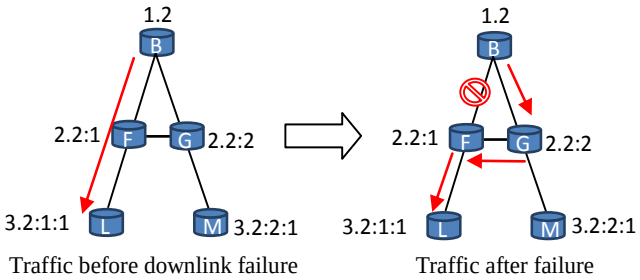


Figure 7. Failure handling with downlink

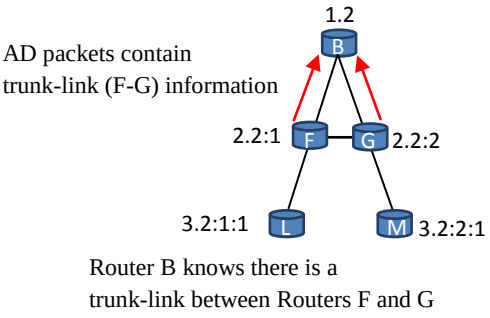


Figure 8. Trunk-link information sharing by the parent router

3) Packet Forwarding in TRP: Packet forwarding in routers running TRP is done as follows. The source router compares the source and destination TRAs to determine the TV of a common parent (grandparent) router between them. Assume source is Router L and destination is Router M in Fig. 4. Source Router L compares TA in its TRA namely ***2***:1:1 with the TA of the destination router's TRA namely ***2***:2:1 from left to right to find the common digit in these addresses. In this case, it happens to be the **1st** digit *2* (shown bold italic character) in the *first place*. This provides the information that a common parent (grandparent) between the two routers resides at *tier 1*. The TV in the forwarding address is thus set to *1*. To this TV is then appended the TA of the destination router to provide the forwarding address 1.2:2:1. Another example, for a forwarding address between source Router J ***1:1***:1 and the destination Router K ***1:1***:2 will be ***2***.1:2 because a common parent is identified at *tier 2*. The pseudo code for the forwarding decisions at a TRP router is provided in Algorithm 1 and it is self-explanatory.

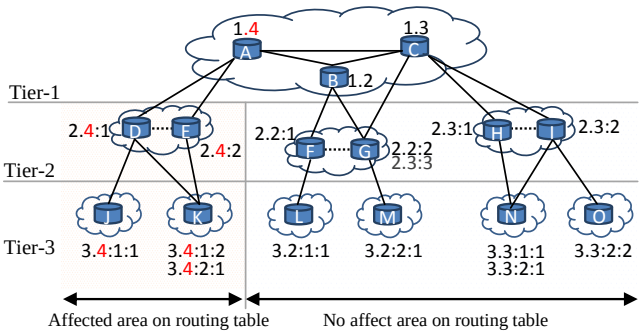


Figure 9. Address changes in TRP

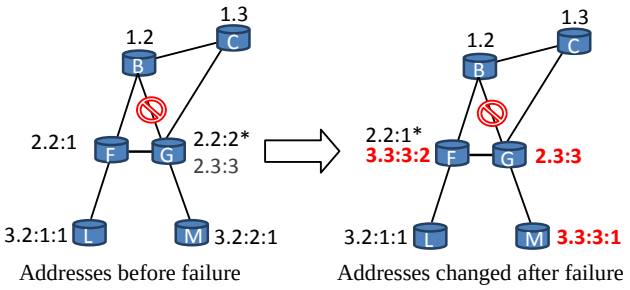


Figure 10. Primary address change

D. Failure Detection and Handling

Failure detection in TRP is *hello* packet based, i.e., typical of layer 3 notification proposed for use with current routing protocols. In TRP, 4 missing AD packets is recognized as link/node failure. A TRP router tracks all neighbors AD packets times and if ADs from a neighbor is missing 4 consecutive times, the TRP router updates its routing table accordingly.

However, in TRP packet forwarding on link/node failure a router does not have to wait for the 4 missing AD packets. An alternative path, if it exists, can be used immediately on missing a single AD packet irrespective of the routing table update. With the current high speed and reliable technologies, it is highly improbable to miss AD packets and redirecting packets on missing one AD packet is justified. However, for a fair comparison with OSPF we adopted the 4 missing hello packets to indicate a link/node failure.

1) Uplink failure: If a node detects an uplink failure and has a trunk link, it can use the trunk link, because trunk link exists between routers that have the same parent route, or it can use an uplink if one exists. In Fig. 6, the sibling router connected to Router F derives its address from the same parent. So, Router F knows that the uplink router on Router G will be its parent Router B.

2) Down link failure: Let a link failure occur between Routers B and F in Fig. 7. To detour around the link failure, down link traffic between Router B and F needs to take a path Router B-G-F. To achieve this, Router B needs to know if there exists a trunk link between Router F and G. A parent router must know all trunk links between its children routers. The trunk link information can be set in AD packets to help a parent router maintain all trunk link information as described in Fig. 8. Due to inheritances, routers can assume responsibilities to forward to their directly connected neighbors as the TRAs carry relationship information.

3) Address Changes: Address changes can happen because of node failure, topology change, or administrative decisions. In TRP, address changes affect limited area and incur very low latency as no updates have to be propagated.

For example, if Router A changed its TRA from 1.1 to 1.4 in Fig. 9, all neighbor Routers B, C, D, and E notice the change from the AD packet sent by Router A. Router D and E will change their TRAs without notifying Router A. Therefore, children of Router A can change their addresses rapidly. The same procedure continues to Routers J and K by the next AD packet from Routers D and E. The pruning operation is triggered on change detection.

4) Primary Address Change: If a node has multiple addresses and a link to a primary address failed, the node changes one of its secondary address to primary address and advertises the same. The child of the node also changes its address in the same manner as described in the case above and keeps the last digit. For example, Router G has two addresses and let 2.2:2 be the primary address in Fig. 10. When a failure occurs between Routers B and G, Router G changes its primary address to 2.3:3 and then advertises it. As a result, Router M changes its address to 3.3:3:1.

IV. TRANSITION WITH MPLS

One major contribution of our work was the study of MPLS as a transition platform to introduce TRP and replace IP and its routing protocols. MPLS achieves similar goals in terms of replacing IP and the routing protocols, but uses the routes from IP routing tables to determine the MPLS paths. Once the paths are established MPLS bypasses the use of IP in the MPLS aware routers. Another feature of MPLS that aided the transition studies was the use of label and label stacking, where in the proposed transition the labels serve to carry the TRP addresses, and label stacking was used to achieve the tiered functionalities, i.e., forwarding across tiers. The packet forwarding decision is the same as Algorithm 1. In this section, the implementation details are presented.

In Fig 11, there are eight MPLS aware routers, Routers A to H. Of these Routers A and F are Label Edge Routers (LER) and the others are Label Switch Routers (LSR). TRAs were assigned to all MPLS aware routers as shown in the figure. Based on the TRAs, it can be noted that Router C is a tier 1 router, Routers B, E, and D are tier 2 routers, while Routers A, G, H and F are tier 3 routers. To conduct the feasibility study, the MPLS tables were manually populated as shown in Figs. 12 and 13. For real implementations using MPLS, the operation of MPLS and its process of populating the tables have to be modified and are not included in this article.

We first explain the use of the tables. The first table in Fig. 12 (a) is for Router A, which is a LER. This router is connected to the IP network 192.168.1.0/24. However, in order to forward a packet to the destination network 10.100.1.0/24, the forwarding table has a dedicated entry. Interpreting this table; when a packet arrives with 10.100.1.0/24 as the destination address, LER A will *push* two labels *1* and *131* where *1* is the outer label (L-1). This

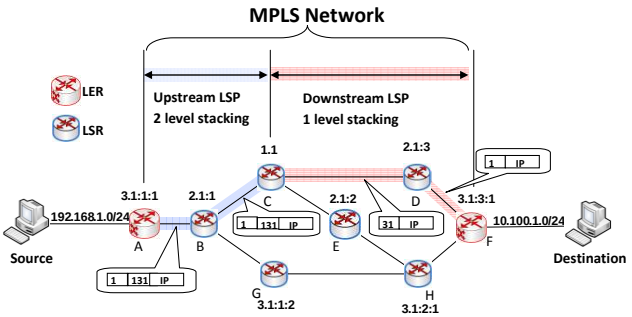


Figure 11. MPLS enabled network with TRP

Destination Network	*Out Label*	*Action*	*Next Hop*
10.100.1.0/24	1(L1) 131(L2)	PUSHx2	Router B

L-1 Label Table			
In Label	*Out Label*	*Action*	*Next Hop*
1	IP header	POP	IP address

a) Router A: 3.1:1:1 (LER)

Destination Network	*Out Label*	*Action*	*Next Hop*
192.168.1.0/24	1(L1) 111(L2)	PUSHx2	Router D

L-1 Label Table			
In Label	*Out Label*	*Action*	*Next Hop*
1	IP header	POP	IP address

b) Router F: 3.1:3:1 (LER)

Figure 12. LER MPLS tables of Routers A and F

packet will then be sent on to the next hop, which is Router B. If a packet arrives to be delivered to network 10.100.1.0/24 at LER A, Router A will *pop* the L-1 label and then forward the packet to the destination IP address in the packet. Similar table entries can be noted for LER F in Fig. 12 (b), which will also perform operations similar to Router A.

At LSR B, it will check the outer label when a packet arrives from Router A and processes the packet forwarding based on the outer label (L-1, tier 1) table. As per this table, when the packet arrives from Router A, if it has a forwarding address where the tier value is *1* (L-1), then the packet will be sent uplink to Router C with a *swapped* label, which will also have a value *1*. If the outer label (L-1) was *2*, it indicates that the anchor tier level is *2* in the forwarding TRA, and Router B is the anchor router (at which time redirection will take place). Hence, Router B will *pop* the L-1 label and the packet will then be processed as per L-2 label table. In the L-2 label, when a packet is received, Router B will *swap* the incoming labels with new labels to deliver the packet to either Routers A or G. Similar entries can be noticed for Routers C and D and their operations will be similar to that explained for Router B and tables are shown in Fig. 13.

Handling tier based forwarding with MPLS can be summarized as:

- For upstream forwarding, a L-1 label indicates that a MPLS packet is to be forwarded until the upper tier level specified in the label is reached. If L-1 label value is lower than router's tier value, it is forwarded to an upper tier.
- For downstream forwarding, if L-1 label value is the same as router's tier value, the router removes (*pop*) L-1 label and forwards the packet to a lower tier based on L-2 label.

L-1 Label Table			
In Label	*Out Label*	*Action*	*Next Hop*
1	1	SWAP	Router C
2	N/A	POP	N/A
11	1	SWAP	Router A
12	2	SWAP	Router G
L-2 Label Table			
In Label	*Out Label*	*Action*	*Next Hop*
11	1	SWAP	Router A
12	2	SWAP	Router G

a) Router B: 2.1:1 (LSR)

L-1 Label Table			
In Label	*Out Label*	*Action*	*Next Hop*
1	N/A	POP	N/A
L-2 Label Table			
In Label	*Out Label*	*Action*	*Next Hop*
111	11	SWAP	Router B
131	31	SWAP	Router D

b) Router C: 1.1 (LSR)
Note: Router C may have more entries

L-1 Label Table			
In Label	*Out Label*	*Action*	*Next Hop*
1	1	SWAP	Router C
2	N/A	POP	N/A
31	1	SWAP	Router F
L-2 Label Table			
In Label	*Out Label*	*Action*	*Next Hop*
31	1	SWAP	Router F

c) Router D: 2.1:3 (LSR)

Figure 13. LSR MPLS tables of Routers B, C, and D

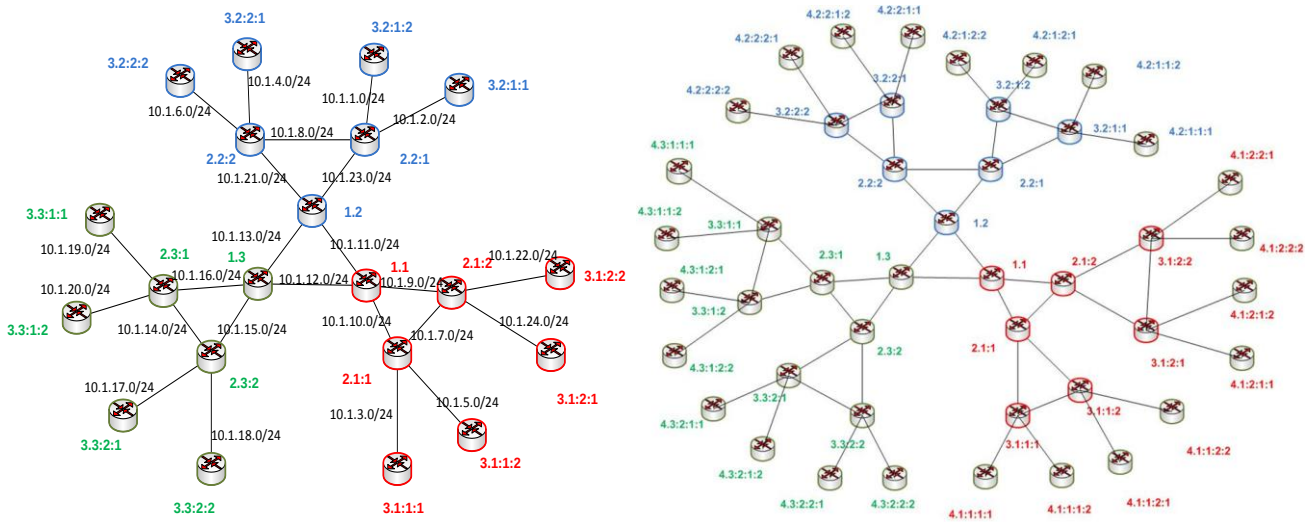
We now work through an example of packet forwarding in the network scenario shown in Fig. 11. Let the source node send a packet to a destination node with destination IP address 10.100.1.x, where x is the host identifier. LER has to be aware of the TRA allocated to network with IP address 10.100.1.0/24. This TRA is 3.1:3:1. Following are the steps.

1) Forwarding TRA calculation: Router A calculates the forwarding TRA to 3.1:3:1 by comparing with own TRA (3.1:1:1) with destination TRA 3.1:3:1. The forwarding TRA will be 1.1:3:1.

2) AddingMPLS header: Router A add two MPLS label to the packet using two *push* operations, where the L-1 label is *1*, L-2 label is *131*. The packet is then forwarded to the next hop Router B.

3) 1st hop: Router B checks the outer label, i.e., L-1 label value of *1*. This is less that Router B's tier value *2*. Thus, the packet will be forwarded to an upper tier based on L-1 label table. In this case, the label will be *swapped* to *1* and then the packet will be forwarded to next hop Router C.

4) 2nd hop: Router C checks L-1 label value of 1. This equals Router C's tier value of 1. Router C will remove the L-1 label through a pop operation and then the packet



a) 21 nodes b) 45 nodes

Figure 14. Testbed Topology with Tiered Addresses

should now be redirected. Router C will hence check the L-2 label, which is *131*, in the packet and compares it with its L-2 label table entry. Then, Router C forwards the packet to the next hop Router D after *swapping* the label from *131* to *31*.

5) 3rd hop: Router D checks L-1 label value *31* and lookups its L-1 label table. It will *swap 31* to *1* and then forward to the next hop Router F.

6) Removing MPLS header: Router F checks the L-1 label value of *1* and lookup its L-1 label table. It will then *pop* (removes the MPLS header from the packet) and checks the IP header to forward to the final destination.

V. EMULATIONS

A. Emulab Test Setup

TRP routers were implemented on Linux machines in Emulab. Emulab is an experimentation facility, which allows setting up networks with different topologies to provide a fully controllable and repeatable experimental environment. Emulab uses different types of equipment for this purpose. Two different types of machines were used during the course of this experiment, as allocated by the Emulab team.

Quagga 0.99.17 [12], a software routing suite for configuring OSPF was used for the comparison studies. IPerf [11] was used to generate data traffic.

A 21-nodes topology is shown in Fig. 14 (a). The configuration details are provided in Table II. In the 45-node topology, the additional 24 nodes were added to the outer circle of the 21 nodes' topology and displayed in Fig. 14 (b). The IP addresses were allocated from address space 10.1.x.x/24 to the segments as shown for OSPF. The TRAs for TRP were allocated using the scheme described in Section III-B.

B. Assumptions

1) More complex or meshed topologies could not be created due to the limitations on the number of interfaces on the Emulab machines. The number of physical network interfaces of Emulab PCs is limited to five, where one interface is used for control. Therefore, only four network interfaces are usable for setting up the test topologies. TRA address allocation mechanism will create logical tree-like topology on a physical meshed topology. Thus, we select tree-like topologies to utilize all links on the emulation because of the limited number of interfaces.

TABLE II. EMULAB TESTBED CONFIGURATIONS

Topology	21 Nodes	45 Nodes
Type of processor	Pentium III	Quad Core Xeon Processor
Number of links	24	54
Connection speed	100 Mbps	100 Mbps

2) TRP code operates on Linux user space and hence the timings and dependent variables such as packet loss during convergence showed a higher value than if the code were run in kernel space. Comparatively the Quagga OSPF code runs in kernel space. However, we present the parameters as collected without any corrections for the higher projected values noted for TRP.

3) To provide a random environment for the tests, they were conducted in two different sets of networks and the experiments repeated five times in each case. For a given 21-node topology or 45-node topology the machines were maintained the same throughout the emulation runs.

4) To emulate link failures, Emulab uses link shaping nodes that can be placed on the segments. We adopted this approach to fail links between Node 1.3 and Node 2.3:2 for both the 21-node and 45-node scenarios.

5) For OSPF evaluations, only one area was defined, as the intention is to demonstrate the performance impacts to increase the number of routers in a network or an area.

C. Tiered Routing Protocol Code

TRP runs above layer 2, *bypassing all layers* between layer 2 and the application layer. It replaces both IP and its routing protocols. To run applications on TRP, a modified

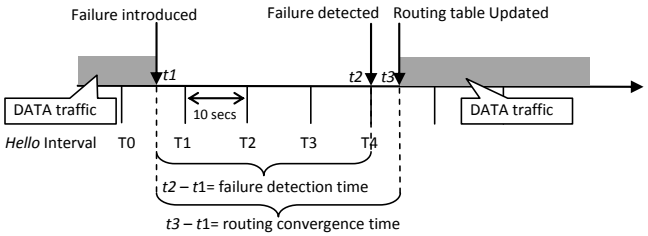


Figure 15. TRP Routing Convergence Time

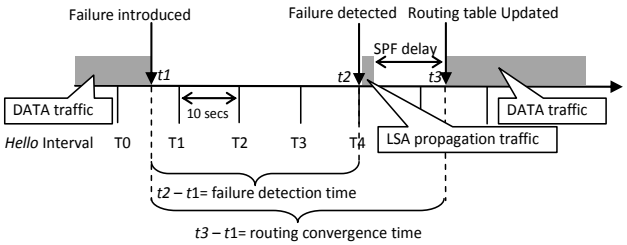


Figure 16. OSPF Routing Convergence Time

clone of IPerf called SIPerf, which allows bandwidth and link quality measurement in terms of packet loss, was used.

D. Initial Convergence Performance Statistics

1) Convergence Times: In OSPF, initial convergence takes place after the FIB update is run on all routers. To improve the veracity of collected data, the timestamps when SPF was run as well as the time when the routing table was updated was logged. For TRP, the timestamp for a new entry in the routing tables is logged and if the routing table at the routers remains unchanged for the next three *hello* intervals then the network was deemed to have converged.

2) Routing Table Size: In OSPF, this value was logged using the built-in commands provided by Quagga. In TRP, this information was logged in a file and sent to the server.

3) Control Overhead: To collect control overhead, Tshark [13], which is similar to Wireshark [13] was utilized to capture packets from which the control packets were accounted for. Tshark is a command-line tool and it was invoked through special scripts during the emulation. Bytes in the packets exchanged during convergence were summed to determine the control overhead at each node and then sent to the server. In TRP, a utility to record the number of control packets exchanged during initial convergence time was built in.

E. Link Failures Performance Statistics

Convergence time after link failure has two components.

1) Link failure detection time: This is the same for OSPF and TRP as they detect a link failure on missing 4 *hello* messages. With a *hello* interval of 10 seconds, this was recorded to be 30 seconds with an additive time - time between the first missing hello packet and the time when the link was actually brought down.

2) Time to update routing tables: This time is different for OSPF and TRP and the differences are explained using Figs. 15 and 16.

3) TRP Response to Link Failures: In Fig. 15, the time t_1 when the link failed is noted along with time t_3 ,which is the time it took to remove the link from the routing table.

Total time for convergence T_c is given by

$$T_c = T_{ru} - T_{fd} \quad (1)$$

where T_{fd} is the failure detection time given by

$$T_{fd} = t_2 - t_1 \quad (2)$$

and T_{ru} is the routing table update time given by

$$T_{ru} = t_3 - t_2 \quad (3)$$

Thus,

$$T_c = t_3 - t_1 \quad (4)$$

T_{fd} will be the same for OSPF, but T_{ru} is negligible in the case of TRP as this is the time for the TRP code to access the routing tables and update its contents. In Figs. 15 and 16, these times are identified based on the operations of TRP and OSPF, respectively.

4) OSPF Response to Link Failure: OSPF uses several timers on link failures, to rerun SPF algorithm and a few other hold times to avoid toggling. They are *Hold_Time*, which is the separation time in milliseconds between consecutive SPF calculations. An *Initial_hold_time* and *Max_hold_time* is also specified. SPF starts with the *Initial_hold_time*. If a new event occurs within the *hold_time* of any previous SPF calculation then the new SPF calculation is increased by *initial_hold_time* up to a maximum of *max_hold_time*.

Let T_{LSA} be the LSA propagation delay, T_{SPF} be the time to run SPF on subsequent LSA messages and T_{TU} be the table update delay, then T_{ru} of OSPF is given by

$$T_{ru} = T_{LSA} + T_{SPF} + T_{TU} \quad (5)$$

T_{SPF}, *initial_hold_time* and *max_hold_time* were set to 200 ms, 400 ms, and 5000 ms respectively for the test. Fig. 16 captures the relationship between the delays for OSPF.

VI. Performance Analysis

The performance of OSPF and TRP, during the initial convergence phase and their response to subsequent link failures are presented in this section. In the histograms, data collected for the two test sites are provided separately, to show the closeness of the two data sets under different environments to reflect the reliability of the experiments.

1) Initial Convergence Times

Fig. 17 records the average initial convergence times in seconds collected from the two test sites and for the two different topologies, one with the 45-router and the other with 21-router. While the convergence times recorded for OSPF range from 55 seconds in the case of the 21-router network to over 60 seconds in the case of the 45-router

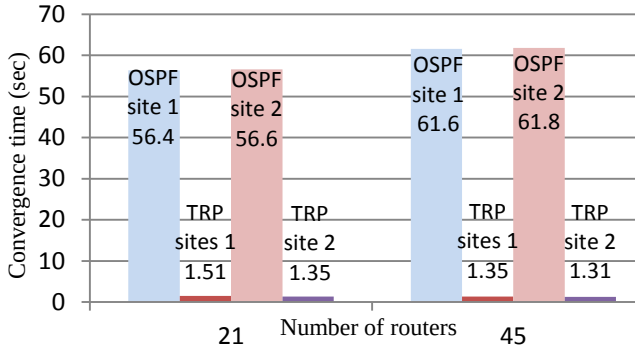


Figure 17. TRP vs. OSPF Initial Convergence Time (sec)

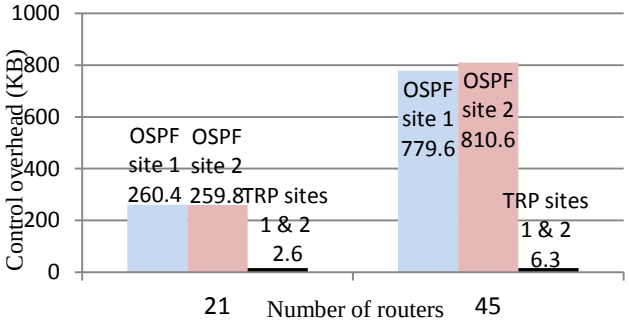


Figure 18. TRP vs. OSPF Routing Control Overhead Size (KB)

network, the convergence times for the network running TRP was around 1 second. While convergence times are stable irrespective of the number of routers running TRP, in the case of OSPF, the convergence times showed an increase by 5 to 6 seconds, indicating dependency of convergence times to the network size. TRP thus has 50-60 times improvement compared to OSPF.

2) Control Overhead During Initial Convergence

Fig. 18 shows the plot of control overhead in Kbytes for OSPF and TRP. Control overhead in the case of OSPF varies from 250 Kbytes for the 21-router network to around 750 to 800 Kbytes for the 45-router network. Increase in overhead almost triples as network size doubles. Control overhead for TRP was 2.6 Kbytes for the 21-router network and around 6 Kbytes for the 45-router network. The improvement achieved with TRP is 100 times in the case of the 21-router network and 120 times in the case of the 45-router network.

3) Routing Table Sizes

In Fig. 19, the routing table sizes collected were the same in the case of OSPF and TRP for the two test sites and hence one graph with the maximum routing table entries is provided. In the case of OSPF, this value is 25 for the 21-router network (as there are 25 segments) and in the case of the 45-router network this value was 55. In the case of TRP, the routing table entries reflects the number of directly connected neighbors, so in both cases, the maximum routing table entry was 4, there is no dependency on the network size.

4) Convergence Time After Link Failure

Fig. 20 displays the routing table update time in seconds subsequent to link failure detection. While OSPF shows an update time of 1.5 to 2 seconds for the 45-router network and

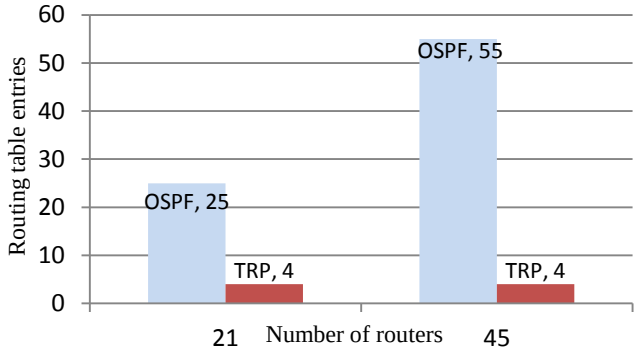


Figure 19. TRP vs. OSPF Routing Table Entry Size

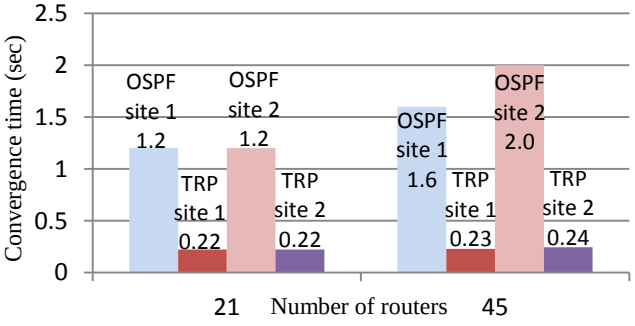


Figure 20. TRP vs. OSPF Convergence Time after Failure (sec)

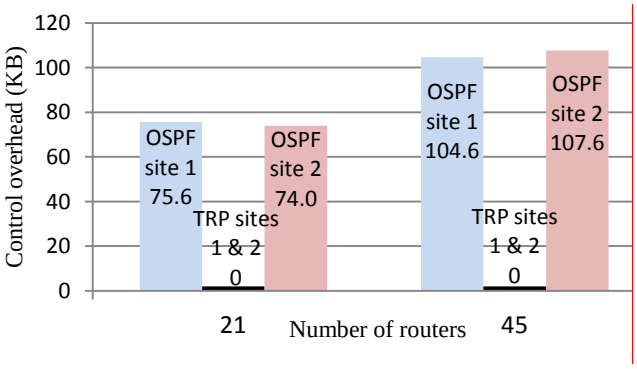


Figure 21. TRP vs. OSPF Control Packet Size after Failure (KB)

around one second for the 21-router network, TRP update times were 200 to 240 milliseconds; a magnitude of 6 improvement for the smaller network and a magnitude of 8 improvement for the larger network. Routing table update time is invariant to the network size in the case of TRP.

5) Control Overhead After Link Failure

Control overhead for TRP and OSPF collected during the convergence times, includes the time to detect a failure and also time to update routing tables. For the given topologies no control overhead was incurred with TRP. In Fig. 21, OSPF required around 100 Kbytes and 70 Kbytes of control packets for the 45-router and 21-router networks respectively. For complex topologies, in TRP change in topology information may have to be propagated to downstream networks. Similarly, upstream router may also have to be informed when a downstream link fails. These features were not tested in the scenarios.

6) Data Packets lost

The packets lost during failure detection will be the same for both protocols as the failure detection time is 4 missing *hello* packets. The time to update routing tables was recorded to be around 0.2 seconds for TRP and 1.2 to 2.0 seconds for OSPF. Thus, the packets lost during routing table update time was a maximum of 1 packet for TRP and a maximum of 10 packets with OSPF at a data rate of 5 packets per second.

From the results presented so far, it would be clear that TRP would be an ideal routing protocol to address scalability concerns as networks grow in number and in size. This is true as the routing table sizes and routing table update time is independent of network size. This in turn will positively impact the routing performance in the network. The convergence times are also very low and changes in network topology do not require network or area-wide dissemination of the changes. This will reduce instability in routing packets and also reduce packet loss.

VII. Conclusions and Future Work

A Tiered Routing protocol was developed under a new tiered Internet architecture. The tiered addresses in this architecture are used by TRP for packet forwarding. In this article, TRP is evaluated as an IGP using Emulab test facility. Initial convergence time and control overhead with networks running TRP is very low as the protocol does not require message flooding or any calculations subsequent to a link status change. Due to the inherent routing information in the tiered addresses, the routing table sizes in TRP are significantly low. Stability in the routing entries and their invariance to network size also indicates the strengths of such new approaches. Comparison with OSPF validates this.

There are several possible directions for future work. OSPF supports area concept for large network, so apply the area concept for larger network to compare with TRP. Validating TRP for inter-domain routing is another direction. Since tier levels in Autonomous System (AS) level topology can also be identified, based on their business relationships such as provider-customer and peer-peer relationship, TRP can be applied for inter-domain routing. Thus, Border Gateway Protocol (BGP) and TRP can be compared to validate TRP as inter-domain routing protocol.

Acknowledgment

This work was sponsored by NSF under grant number 0832008.

References

[1] Y. Nozaki, P. Bakshi, and N. Shenoy, "Tiered interior gateway routing protocol," ICNS 2013, The Ninth International Conference on Networking and Services, pp. 68-75, 2013.

[2] J. Moy, "RFC 1245 - OSPF protocol analysis," RFC Editor, 1991.

[3] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure, "Open issues in interdomain routing: a survey," Network, IEEE, vol. 19, no. 6, pp. 49- 56, 2005.

[4] Y. Nozaki, H. Tuncer, and N. Shenoy, "A tiered addressing scheme based on floating cloud internetworking model," Distributed Computing and Networking, Lecture Notes in Computer Science, vol. 6522, pp. 382-393, 2011.

[5] "Emulab: network emulation testbed," http:// www.emulab.net. (accessed December 2013)

[6] C. Alaettinoglu, V. Jacobson, and H. Yu, "Towards milli-second IGP convergence," Internet Draft, IETF, 2000.

[7] P. Pan, G. Swallow, and A. Atlas, "RFC 4090 - Fast reroute extensions to RSVP-TE for LSP tunnels," May 2005.

[8] A. Kvalbein, A.F. Hansen, T. Ciˇci´c, S. Gjessing, and O. Lysne, "Multiple routing configurations for fast IP network recovery," IEEE/ACM Transactions on Networking, vol. 17, no. 2, pp. 473-486, 2009.

[9] Y. Liu and A.L.N. Reddy, "A fast rerouting scheme for OSPF/IS-IS networks," In Proceedings of ICCCN, pp. 47- 52, 2004.

[10] N. Shenoy, M. Yuksel, A. Gupta, K. Kar, V. Perotti, and M. Karir, "RAIDER: Responsive architecture for inter-domain economics and routing," GLOBECOM Workshops (GC Wkshps), 2010 IEEE, pp. 321-326, 2010.

[11] "Iperf: the TCP/UDP bandwidth measurment Tool," http://www.iperf.sourceforge.net. (accessed December 2013)

[12] "Quagga software routing suit," http://www.quagga.net. (accessed December 2013)

[13] "Tshark and wireshark," http://www.wireshark.org. (accessed December 2013)

[14] P. Narvaez, "Routing reconfiguration in IP networks," Ph.D. dissertation, MIT, June 2000.

[15] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel," IEEE/ACM Transactions on Networking, vol. 12, no. 1, pp. 2-16, 2004.

[16] "Cytoscape," http://www.cytoscape.org. (accessed December 2013)