Tiered Interior Gateway Routing Protocol

Yoshhiro 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 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 are Link-State routing protocols and require distribution of link state information to all routers. Topological changes require redistributing updates and refreshing routing tables, resulting in high convergence times. Routing table sizes grow linearly with network size, indicating scalability issues. Future Internet initiatives provide new venues to address the routing problem. In this article, a Tiered Routing Protocol (TRP) is presented as a candidate protocol for intra-AS routing. TRP is supported by a tiered addressing scheme. TRP replaces both IP and the routing protocol. TRP's performance is compared with OSPF using Emulab test-beds.**

Keywords-Intra-domain Routing; Network Convergence; Internetworking Architectures; Tiered architectures; Routing Table sizes.

I. INTRODUCTION

In IP networks, routers use routing protocols to discover and maintain routes to other networks. Routing table sizes maintained by current routing protocols increase linearly with increase in network size and is indicative of scalability issues which can manifest as performance deterioration. Also, the time taken for the network to adapt to topological changes increases with network size resulting in higher convergence times during which routing is unreliable. Patch and evolutionary solutions address the problem both at inter and intra domain level [1, 2].

Interior Gateway Protocols (IGP) such as Routing Information Protocol and OSPF were designed to work with IP. Large ISP networks 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 on link state changes. Each router running the LS routing protocol executes the Dijkstra's algorithm on the link state information to populate routing tables. Dissemination of network-wide (or area-wide) link state information adversely impacts scalability and convergence time when using OSPF.

A primary contribution in this work is the decoupling of the routing table sizes from the network size. A major goal was to investigate a solution that is acceptable to the service provider community. Thus, the proposed internetworking model derives from the structures used by ISPs to define their business relationships namely the *tiers*. The routing protocol proposed under this internetworking model is called the *tiered routing protocol* (TRP). A new tiered addressing scheme was introduced. The tiered address inherits attributes of the tiered structures. To decouple dependencies between connected entities, a nesting concept is introduced [3].

TRP replaces both IP and routing protocol. In this article, TRP operation as an IGP is described and evaluated. The tiered structure within an AS is identified and used for the purpose. Its performance is compared with OSPF using Emulab [4] test-beds. The performance metrics evaluated were: 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 some related work in reduction of convergence times in IGPs. Section III describes the two routing protocols under study. Section IV provides details of the emulations tests and Section V analyses the results of the tests. Section VI provides the conclusions.

II. RELATED WORK

Significant research effort has been directed towards reduction and optimization in IGP convergence time to link state changes in the network. The work 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. This relies on types of network interfaces and does not apply to switched Ethernet [5].

Hello protocol is used to identify link/node failure in many routing protocols and is called layer-3 failure detection. OSPF sends *hello* packets to adjacent routers at regular intervals. On missing four *hello* packets consecutively, OSPF routers recognize an adjacency failure with a neighboring router. Reducing *hello* packet interval time to sub-seconds can significantly reduce the failure detection time at the expense of increased bandwidth use.

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, several pre-computed back up routing path approaches have been proposed. Pan et al. [6] proposed the Multi Protocol Label Switching (MPLS) based on a back up path to reroute around failures. However, having all possible MPLS back up paths in a network is not efficient. Multiple Routing Configurations (MRC) [7] 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. MRC guarantees recovery from only single failures. Liu at el. [8] proposed use of pre-computed rerouting paths if resolved locally. Otherwise multi-hop rerouting path had to be set up by signaling to a minimal number of upstream routers.

While the above two delays are of significance, SPF recalculation time can also be almost a second in large networks [5]. As packet loss/delay or routing loops occur during convergence, it is important to reduce this time.

III. Routing Protocols and Operations

In this section, we describe the operations of the two protocols that are OSPF and TRP. In the case of OSPF, only a few basic operations necessary to explain the performance metrics are presented. Details are available in [1]. TRP operations include implementing tiered structures within an AS, tiered address allocation to devices in the tiers, routing table maintenance with TRP, and the packet forwarding algorithm and failure handling.

A. Open Shortest Path First (OSPF)

Basic operations of OSPF include: (a) establishing adjacencies with neighbors and electing a Designated Router (DR) and a Backup DR (BDR); (b) maintaining Link State Database (LSDB) and; (c) executing Dijkstra's algorithm. The operations are invoked during startup and also in response to link state changes. Convergence in each case is impacted differently and described separately below.

1) Initial Convergence in OSPF

a) Establishing Adjacencies: OSPF establishes adjacencies with direct neighbors using the *Hello* protocol. Once *hello* packets are exchanged, each router recognizes the adjacent routers and elect the DR and BDR.

b) Maintaining Link State Databases: On link state establishment as nodes come up, distribution of adjacency information to all routers is initiated by flooding Link State Advertisements (LSA). Each router maintains the flooded link state information in LSDBs.

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

2) Convergence After Link / Node Failures

a) Failure Detection: Missing 4 *hello* consecutive packets from a neighbor indicates link or router failure on that link and hence is one mechanism for failure detection.

b) LSA Propagation: After failure detection, a router generates new LSAs to be propagated to all routers in the network (area). The time for generating new LSAs for a single failure is between 4ms and 12ms [8] and OSPF specifies that LSAs cannot be created within 5 seconds from the last LSA generation time to provide sufficient time to update the LSDB from the last event. LSA propagation time also depends on the number of hops between the routers in the network and the processing delay at each router/hop.

c) SPF Recalculation Time: New LSAs update the LSDB and 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)

Identifying the tiered structure is described first. In large ISP and AS networks, backbone routers provide connectivity between distribution routers, which, in turn, connect to access routers or sub-networks. In the proposed tiered architecture, the set of backbone routers are designated as tier 1 routers, distribution routers as tier 2 routers and the access routers and sub-networks that they connect as tier 3. This is adopted in the presented studies. A Tiered Routing Addresses (TRA) is required [9] for the purpose. Some features of TRA and resulting impacts on TRP are described below.

1) TRA Allocation: TRA depends on the tier level in a network and carries the tier value as the first field. The tier levels were assigned as stated above. Basically, nodes near backbone or default gateway have lower tier value and nodes near 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 node. It is not allocated to network interface, which will be identified by port numbers. TRA assignment is made to the node. However, a node can have multiple TRAs based on its connection to the upper tier nodes or networks to support multi homing.

2) TRA Guarantees Loop-Free Routing: TRA allocation starts from a lower value tier to higher value tiers. The parent's address (without the tier value) precedes a child's address. As TRAs determine the packet forwarding paths, this attribute avoids packet looping. However, the dependency can be decoupled at any level through *nesting*.

3) Nested TRA: TRAs can be assigned to network cloud. A new TRA can be started for entities within a network cloud, allowing nesting of TRAs. If a network administrator wishes to incorporate clouds in a cloud, nested TRAs can be used where TRA of an inner cloud does not depend on the TRA of the outer cloud. This decoupling provides a high level of scalability and flexibility in the internetworking.

4) Inherent Routing Information: TRA carries the path information between a lower tier entity and an upper tier entity due to the inheriting the parent's TRA in the child TRA (without tier values). Thus, a route between two communicating nodes can be identified by comparing the nodes' TRAs. If a node has multiple TRAs, a sender node can select a communication path based on criteria such as a shorter path or better resources.

5) 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 change in the neighborhood. This will be several magnitudes less than that required by current routing protocols. The extent of information dissemination can be controlled for optimization.

6) 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

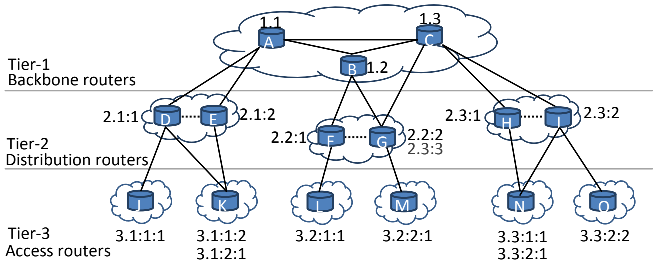


Figure 1. Example Tiered Topology and TRA

directly connected neighbor networks /routers. Optimization is possible by including the two-hop or three-hop neighbors.

C. TRP Operation

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

1) Address Allocation Process: TRA allows automatic address allocation by a direct upper tier cloud / node. Once tier 1 nodes acquire 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 (tier 1). A tier 1 node advertises its TRA to all direct neighbors. A node, which receives an advertisement, sends an address request and is allocated an address. For example in Fig 1, 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 TRA yet. Router B and C do not request address to Router A because they are at the same tier level. In Fig. 2, Router A allocates 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 in Fig. 1. The last digit of the new address is maintained by the parent router - 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.

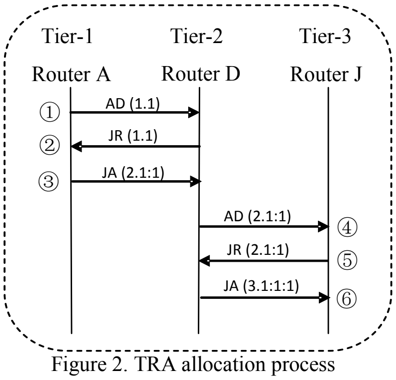


Figure 2. TRA allocation process

TABLE I. ROUTING TABLES OF ROUTER F AND G FROM FIGURE 1

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				

b) Mutli-Addressing: If a router has multiple parents, like Router G in Fig.1, 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 this work.

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 which connects to an upper tier router; down-link which connects to a lower tier router; and trunk-link which 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.

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

```
if( R.TV == P.TV ) then
   if( R.TA.last_digit == P.TA.1st_digit ) then
      if( port_num = find ( P.TA.2nd_digit, down-link table ) ) then
         remove( P.TA.1st_digit );
         P.TV++;
         forward( P, port_num );
         return();
      end if
   else if( R.TV == 1 ) then  //at Tier-1
      if( port_num = find ( P.TA.1st_digit, up-link table ) ) then
         forward( P, port_num ),
         return();
      end if
else if( R.TV – P.TV == 1 && R.TA.parent_digit == P.TA.1st_digit ) then
   if( port_num = find ( P.TA.2nd_digit, trunk-link table ) ) then
      remove( P.TA.1st_digit );
      P.TV++;
      forward( P, port_num );
      return();
   end if
else if( R.TV < P.TV ) then
   discard( P ); //wrong packet
   return();
end if
if( port_num = find (up-link table ) ) then
   forward( P, port_num );
   return();
end if
discard( P );  //no entry in routing tables
return();
```

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

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.

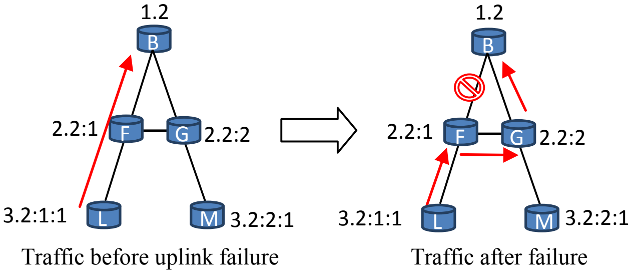
In the network in Fig. 1, three tier levels have been identified, and the TRA for the routers in this network are noted beside them. The TRA is made up of *TV. TA*, where *TV* is the tier value to identify the tier level and *TA* is the address of the router. A '.' notation in the tiered address separates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

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 TV of a common parent (grandparent) router between them. Assume source is Router L and destination is Router M in Fig. 1. 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.

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 interval and if ADs from a neighbor is missing 4 consecutive times, the TRP router updates its routing table accordingly.

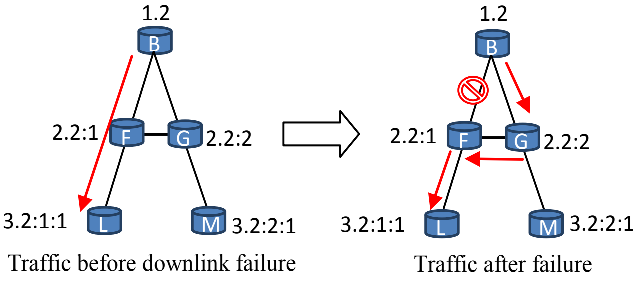
However, in TRP packet forwarding on link/node failure does not have to wait for the 4 missing AD packets. An alternate path, if it exists, can be used on detecting a single missed AD packet irrespective of the routing table update. With the current high speed and reliable technologies, it is highly improbable to miss AD packets and redirecting packets on missing one AD packet is justified.



Traffic before uplink failure

Traffic after failure

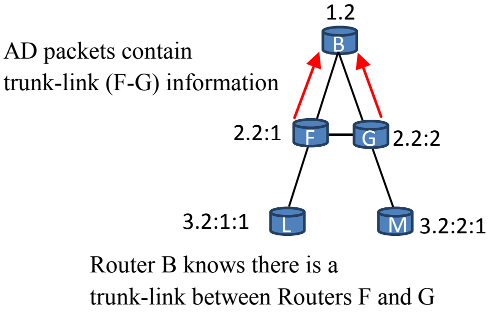
Figure 3. Failure handling with uplink



Traffic before downlink failure

Traffic after failure

Figure 4. Failure handling with downlink



AD packets contain trunk-link (F-G) information

Router B knows there is a trunk-link between Routers F and G

Figure 5. Trunk-link information sharing by the parent router

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 router, or if a router has another uplink, it can use it. In Fig. 3, 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 link failure occur between Routers B and F in Fig. 4. 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. 5. Due to the inheritances, routers can assume responsibilities to forward for their directly connected neighbors as the TRAs carry relationship information.

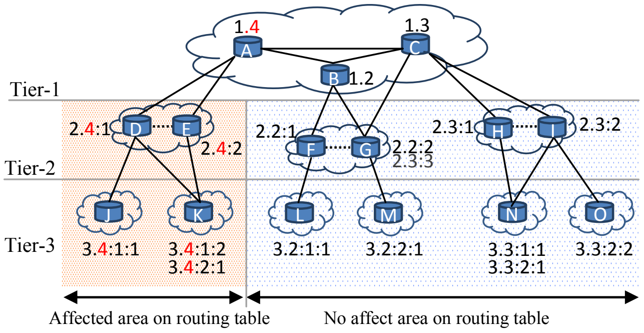
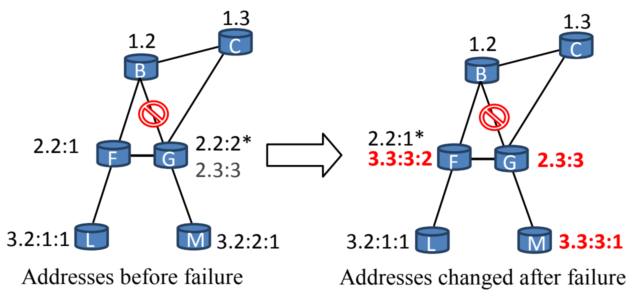


Figure 6. Address changes in TRP



Addresses before failure

Addresses changed after failure

Figure 7. Primary address change

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. 6, 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. 7. When a failure occurs between Routers B and G, Router G changes its primary address to 2.3:3 and then advertises it. As the result, Router M changes its address to 3.3:3:1.

IV. EMULATIONS

A. Emulab Test Setup

A TRP router was implemented on Linux machines in Emulab. Emulab is an experimentation facility which allows creation of 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 [11], a software routing suite for configuring OSPF was used for the comparison studies. Iperf [10] was used to generate traffic.

A 21 node topology is shown in Fig. 8. The configuration details are provided in Table II. In the 45 nodes topology, the additional 24 nodes were added to the outer circle of routers utilizing a topological connection similar to that of the outer routers in Fig. 8. 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.

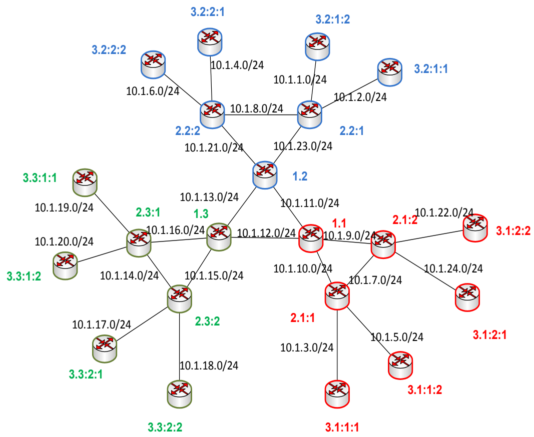


Figure 8. Testbed Topology with IP and Tiered Addresses

TABLE II. EMULAB TESTBED CONFIGURATIONS

Topology	**21 Nodes**	**45 Nodes**
Type of processor	Pentium III	Quad Core Xeon Processor
Number of links	24	54
Link shaping nodes	12	20
Connection speed	100 Mbps	100 Mbps

2) TRP code operates on Linux user space and hence timings and dependent variables such as packet loss during convergence project a higher value than if the code were run in kernel space. Quagga OSPF code runs in kernel space.

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.

4) To emulate link failures, Emulab uses link shaping nodes that can be placed on the segments.

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

B. 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, SIPerf, a modified clone of Iperf which allows bandwidth and link quality measurement in terms of packet loss, was used.

C. Performance Statistics on Initial Convergence

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*: Tshark [12] which is a command line tool similar to Wireshark [12] was utilized for the purpose. 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.

D. *Performance Statistics on Link Failures*

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 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 are explained using Figs 9 and 10.

3) TRP Response to Link Failures: In Fig. 9, the time t_1 when the link failed is noted along with time t_3 it took to remove the link from the routing table. Total time for convergence T_c is then given by

$$T_c = T_{ru} - T_{fd} \tag{1}$$

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

$$T_{fd} = t_2 - t_1 \tag{2}$$

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

$$T_{ru} = t_3 - t_2 \tag{3}$$

Thus,

$$T_c = t_3 - t_1 \tag{4}$$

T_{fd} will be the same for OSPF, but T_{ru} is negligible in the case of TRP as this is the time to for the TRP code to access the routing tables and update its contents. In Figs 9 and 10, 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 seperation time in ms 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 Tru of OSPF is given by

$$T_{ru} = T_{LSA} + T_{SPF} + T_{TU} \tag{5}$$

T_{SPF}, *initial_hold_time* and *max_hold_time* were set to 200ms, 400ms, and 5000ms respectively for the test. Fig. 10 captures the relationship between the delays for OSPF.

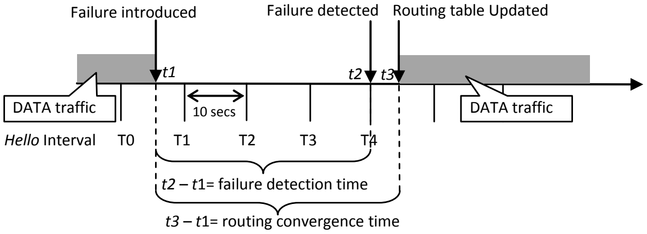


Figure 9. TRP Routing Convergence Time

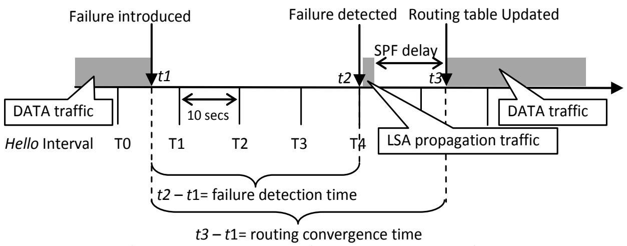


Figure 10. OSPF Routing Convergence Time

V. 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. 11 records the average initial convergence times in seconds collected from the two test sites and for the two different topologies, one with 45 routers and the other with 21 routers. While the convergence times recorded for OSPF range from 55 secs in the case of the 21 router network to over 60 secs in the case of the 45 router network, the convergence times for the network running TRP was around 1 sec. 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 secs, indicating dependency of convergence times to the network size. TRP has 50-60 times improvement compared to OSPF.

2) Control Overhead During Initial Convergence

Fig. 12 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 21-router network and around 6 Kbytes for 45-router network. The improvement achieved with TRP 100 times in the case of the 21-router network and 130 times in the case of the 45-router network.

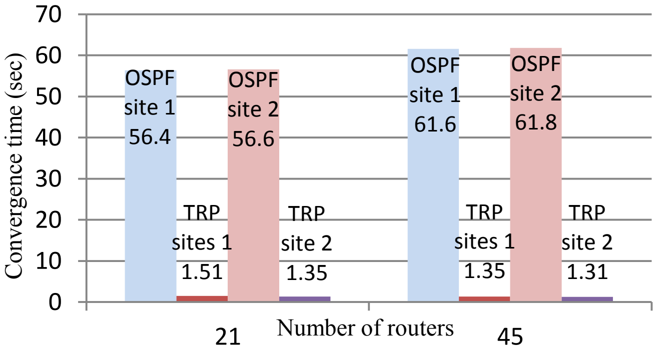


Figure 11. TRP vs. OSPF Initial Convergence Time (sec)

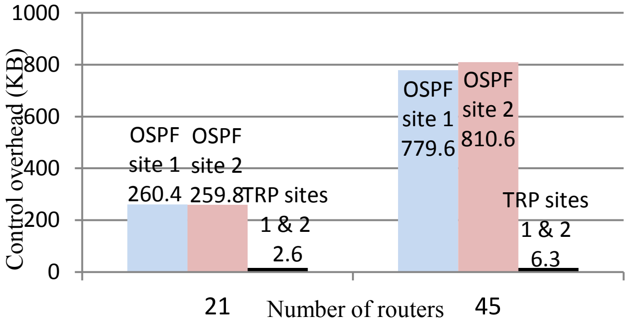


Figure 12. TRP vs. OSPF Routing Control Overhead Size (KB)

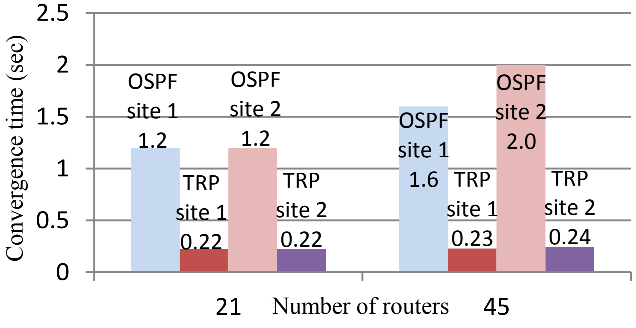


Figure 14. TRP vs. OSPF Convergence Time after Failure (sec)

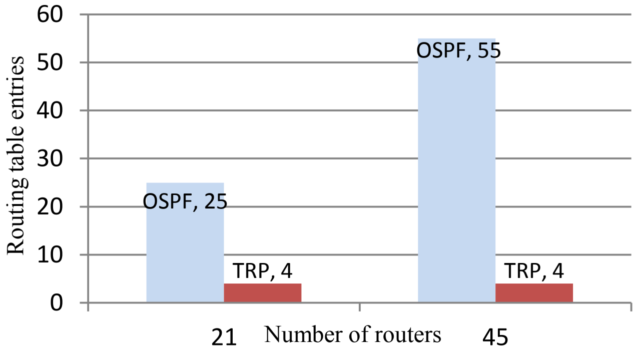


Figure 13. TRP vs. OSPF Routing Table Entry Size

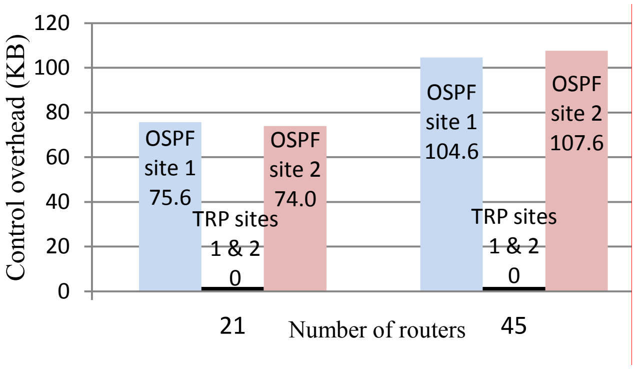


Figure 15. TRP vs. OSPF Control Packet Size after Failure (KB)

3) *Routing Table Sizes*

In Fig. 13, 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 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 45-router network this value was 55. In the case of TRP, the routing table entries reflects 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. 14 has the routing table update time in seconds subsequent to link failure detection. While OSPF shows an update time of 1.5 to 2 secs for the 45-router network and around a 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. 15, 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 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.

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 sec for TRP and 1.2 to 2.0 sec 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.

VI. 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 are compared to validate TRP as inter-domain routing protocol.

ACKNOWLEDGMENT

This work was sponsored by NSF under grant number 0832008.

REFERENCES

[1] J. Moy, “RFC 1245 - OSPF Protocol Analysis,” RFC Editor, 1991.

[2] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure, “Open issues in interdomain routing: a survey,” *Network, IEEE* , vol.19, no.6, pp. 49- 56, Nov.-Dec. 2005

[3] 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 6522/2011, pp. 382-393. 2011.

[4] “Emulab: Network Emulation Testbed,” http:// www.emulab.net. (accessed March 2013)

[5] C. Alaettinoglu, V. Jacobson, and H. Yu, “Towards Milli-Second IGP Convergence,” Internet Draft draft-alaettinoglu-isisconvergence-00.txt, IETF, November 2000.

[6] P. Pan, G. Swallow, and A.Atlas, “RFC-4090, Fast Reroute Extensions to RSVP-TE for LSP Tunnels.” May 2005.

[7] A. Kvalbein, A.F. Hansen, T. Ciˇci´c, S. Gjessing, and O. Lysne, “Multiple routing configurations for fast IP network recovery,” IEEE/ACM Trans. Netw. 17, 2, pp. 473-486, 2009

[8] Y. Liu, and A.L.N. Reddy, “A fast rerouting scheme for OSPF/IS-IS networks,” In Proceedings of ICCCN 2004, pp. 47- 52, 11-13 Oct. 2004.

[9] 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 , pages 321-326, 6-10 Dec. 2010.

[10] “Iperf: The TCP/UDP Bandwidth Measurment Tool,” http://www.iperf.sourceforge.net. (accessed March 2013)

[11] “Quagga Software Routing Suit,” http://www.quagga.net. (accessed March 2013)

[12] “Tshark and Wireshark,” http://www.wireshark.org. (accessed March 2013)