Meshed Tree Protocol for Faster Convergence in Switched Networks

Kuhu Sharma, Bruce Hartpence, Bill Stackpole, Daryl Johnson, Nirmala Shenoy
College of Computing and Information Sciences
Rochester Institute of Technology,
Rochester, NY, USA
kxs3104@rit.edu, bhhics@rit.edu, Bill.Stackpole@rit.edu, daryl.johnson@rit.edu, nxsvks@rit.edu

***Abstract*—Loop free forwarding is a continuing challenge in switched networks that require link and path redundancy. Solutions to overcome looping frames are addressed by special protocols at layer 2, which block ports in the bridges to build a logical spanning tree for frame forwarding. However, due to the continuing convergence issues in the Spanning Tree algorithm, IETF RFC 5556 *Transparent Interconnection of Lots of Links* on RBridges (router bridges) and IEEE 802.1aq Shortest Path Bridging both use link state routing techniques to build Dijkstra trees from every switch. Both techniques have the expense of higher processing complexity. In this paper, a novel meshed tree algorithm (MTA) is investigated to address convergence issues faced by STA while also avoiding the complexity of Link State routing. The MTA based protocol is compared with Rapid Spanning Tree Protocol using OPNET simulations. The significant reduction in convergence time combined with the simplicity in implementation indicates that the Meshed Tree Protocol would be superior candidate to resolve looping issues in switched networks.**

Keywords- Loop Avoidance; Switched Networks; Meshed Trees.

I. INTRODUCTION

Loop free forwarding is a continuing challenge in switched networks. The mandate for link and path redundancy to provide a continued communications path between pairs of end switches in the event of switch or link failure often results in a physical network topology that has loops. The physical loops in turn cause broadcast storms when forwarding broadcast packets. Implementing a loop free logical topology over the physical topology is one way to avoid broadcast storms. The first logical loop-free forwarding technique based on Spanning Tree Algorithm (STA) was proposed by Radia Perlman [1]. Spanning tree in switched networks was constructed by logically blocking some of the bridge's ports. The Rapid Spanning Tree protocol (RSTP) was subsequently developed to reduce the convergence times on topology changes in the basic STP. Transparent interconnection of lots of links (TRILL) on Rbridges (router bridges) was proposed by the same researcher to overcome the disadvantages of STA-based loop avoidance. This came at the cost of some overhead and implementation complexity through the adoption of the Intermediate system to Intermediate system (IS-IS) routing protocol. IS-IS related messages are encapsulated in special frames by Rbridges. This is currently an Internet Engineering Task Force (IETF) draft [4]. Shortest Path Bridging (SPB) was developed along similar lines, adopting the IS-IS protocol. Its specifications can be found in the IEEE 802.1aq standard.

The premise for these solutions is that a single logical tree from a root switch that operationally eliminates physical loops is necessary to resolve the conflicting requirements of physical link redundancy and loop free forwarding. In the event of link failure, the tree has to be recomputed. While spanning tree is a single tree constructed from a single elected root switch, the Dijkstra algorithm adopted in IS-IS allows building a tree from every switch. IS-IS requires link state information in the whole network to be made available to every switch so that each can build its own tree. The Dijkstra algorithm uses the connectivity information to compute the tree.

In this paper, a novel meshed tree algorithm (MTA) is proposed to address the convergence issues faced with STA based protocols and at the same time avoid the complexity in adopting *Link State* routing at layer 2. MTA allows creation and maintenance of *multiple* overlapping tree branches from *one* root switch. The multiple branches mesh at switches, and of failure of a link (or branch) the switch can immediately fall back on another branch. Packet forwarding can continue while the broken branch is pruned. This eliminates temporary inconsistent topologies and latencies resulting from tree reconstruction. It is important to have a tree (logical or physical) for forwarding broadcast packets. But, that should not preclude the construction of multiple tree branches simultaneously or the overlapping of the tree branches if this can be achieved without loops. Redundant tree branches will thus take over packet forwarding seamlessly in the event of a link or failure.

Meshed trees (MT) can be implemented through a simple numbering scheme called MT_VIDs (virtual IDs) that will be assigned to a switch in the bridged network. The MT_VID defines a tree branch or logical packet-forwarding path from the root switch to the switch with the MT_VID. A switch can acquire several MT_VIDs as it is allowed to join multiple tree branches. In this way, meshed trees leverage the redundancy in meshed topologies to set up several loop-free logical forwarding paths without blocking switch ports. Meshed trees can also be built from multiple root switches although this aspect of the Meshed Tree Protocol (MTP) is not covered here.

In this paper, the implementation details of MTP are presented. The performance of MTP is evaluated and compared with RSTP. The comparison was conducted using OPNET simulation tool [7]. RSTP models are available

 ISBN: 978-1-61208-330-8

with OPNET and hence the comparison studies were limited to RSTP. However, the significant improvement in the convergence times and the hops taken by frames to reach destinations indicate the superior capabilities of MTP. The operational simplicity of MTP also provides advantages over complex Link State solutions. MT loop free forwarding at layer 2 is currently the IEEE 1910.1 working group [8] and the authors lead the effort. The rest of the paper is organized as follows. Section II discusses related work in the context of STP and *Link State* based solutions highlighting the comparable features of MT based solutions. In Section III, operational details of the MT algorithm and protocol are presented. Section IV describes the optimized unicast frame forwarding schema adopted in MTP. Section V provides the simulation details and performance results. Section VI follows with conclusions.

II. Related Work

In this section, we focus on the two primary techniques adopted for loop resolution in bridged networks. The first of these is based on the (Rapid) Spanning Tree Protocol (STP and RSTP) and the second is based Link State (LS) Routing. STP and RSTP both use the spanning tree approach. TRILL on RBridges and SPB are two efforts based on LS routing. The presentation in this section focuses on some distinct features of these techniques without describing operational details as such information is publicly available.

A. *Protocols on Spanning Tree Algorithm*

The STP is based on the STA. To avoid loops in the network while maintaining access to all the network segments, the bridges collectively elect a root bridge and then compute a spanning tree from the root bridge. In STP, each bridge first assumes that it is the root and announces its bridgeID. This information is used by the neighboring bridges to elect the root bridge. The unique bridgeID is a combination of a bridge priority and the bridge medium access control (MAC) address. A bridge may supplant the current root if its bridgeID is lower. Once a root bridge is elected, other bridges then resolve their connection to the root bridge by listening to messages from their neighbors. These messages also include path cost information. This continues until the topology converges on a single tree.

STP has high convergence times subsequent to topology changes. To reduce the convergence times the *Rapid Spanning Tree* protocol (RSTP) was proposed [2]. RSTP is a refinement of the STP and therefore shares most of its basic operation characteristics, with some notable differences. The differences are: 1) The detection of root bridge failure is 3 'hello' times. 2) Response to Bridge Protocol Data Units (BPDUs) are sent only from the direction of the root bridge, allowing RSTP bridges to 'propose' their spanning tree information on their designated ports. This allows the receiving RSTP bridge to determine if the root information is superior, and set all other ports to 'discarding' and send an 'agreement' to the first bridge. The first bridge, can rapidly transition that port to forwarding bypassing the traditional listening/learning states. 3) Backup details regarding the discarding status of ports are maintained to avoid failure timeouts of forwarding ports.

Advantages: STA based implementation is simple as the spanning tree is executed with the exchange of BPDUs that carry *tree formation* information.

Disadvantages: Several disadvantages of STA based protocols are noted in [2]. These include: 1) Traffic is concentrated on the spanning tree path, and all traffic follows that path even when other more direct paths are available. This causes traffic to take potentially sub-optimal paths, resulting in inefficient use of the links and reduction in aggregate bandwidth. 2) Spanning tree is dependent on the way the bridges are interconnected. Small changes due to link failure can cause large changes in the spanning tree. Changes in the spanning tree take time to propagate and converge, especially for non-RSTP protocols. 3) Though 802.1Q supports multiple spanning trees, it requires additional configuration, the number of trees is limited, and the defects apply within each tree [3].

B. *TRILL Protocol on RBridges and Shortest Path Bridging*

These two techniques overcome the shortcomings of RSTP as they combine the routing functionality of layer 3 by using the IS-IS protocol [4] at layer 2 to compute pair-wise optimal paths between two bridges. The computed pair-wise optimal paths will be used for forwarding frames at layer 2. Inconsistencies and loop formations during topology change are overcome by a *hop count* used in inter-bridge forwarding. TRILL encapsulates link state routing messages in special headers and uses protocols to learn end station addresses. SPB has two versions; one which creates shortest path trees that are identified by the base VLAN ID called SPBV, and the other which uses the source MAC address to identify the trees and uses MAC in MAC encapsulation. The second technique requires MAC address information dissemination

Advantages over 802-style bridging [4]: 1) Frames travel via an optimal path. 2) Transit frames are routed with a hop count; temporary loops will result in frames being discarded when the hop count reaches zero. 3) Route changes can be made quickly and safely based on local information.

C. *Meshed Tree Protocol*

Tree like structures imposed on topologies may reduce or eliminate loops but also create an environment in which there are failover delays to alternate links. These topologies also lack redundancy or the ability to load balance. Protocols such as SPB and TRILL work to alleviate these problems but are complex, incorporating routing at layer 2 and requiring additional encapsulation. Link State routing requires that link state database be stable for a certain interval of time before running the Dijkstra algorithm to compute forwarding paths; routing and forwarding can be unstable during this time.

MTP seeks to address these same issues with less complexity and even shorter failover times upon discovery of link failure. The core of the protocol is the ability of each Meshed Tree Switch (MTS) to be a member of more than one tree. This provides redundancy and optimized traffic forwarding to hosts, while supporting redundant paths that takeover upon link or switch failures.

III. THE MESHED TREE ALGORITHM

The *meshed tree* algorithm allows construction of logically *meshed trees* from a single root switch in distributed fashion and with local information [5]. The discussion presented in this article does not include the election of a root bridge as the focus is on the loop resolution / avoidance capability of MT algorithms. A process similar to that adopted by STA can be used to elect a root bridge. In this article we assume a designated root bridge, which is an option advocated in IEEE1910.1.

Bridge ID: For the operation of the MT algorithm bridgeIDs are necessary. These have to be unique only within the switched network (a simple MAC address derivative can be used). The MT_VIDs would be thus simple, and the first value in the MT_VID will be the root bridgeID. In this article without loss of generality we used a single digit ID for the root switch. Resolution upon root failure is not included in this work.

An MT_VID describes a path that connects the root to a particular switch. The elements of the MT_VID are derived from the root bridgeID and the outbound port numbers of the switches in the path to that particular switch. In a single physical topology, a switch can be associated with more than one MT_VID and thus:

- A *Meshed Tree* could contain ***all*** of the possible paths from the root to each switch.
- More than one path to each switch is supported

Consider a three-switch single loop topology shown in Figure 1. In the upper left is the physical loop topology. In order to prevent traffic from looping, we might impose any one of several logical tree topologies like those shown. In the upper right, the topology is optimized for transmissions associated with switches connected to the root. But in the lower left and lower right, the topology is optimized for nodes connected to switches A and B, respectively. These tree topologies do not provide for redundancy. Meshed trees utilize all of the pathways and because the pathways are pre-established, *failover times to redundant links are near zero.*

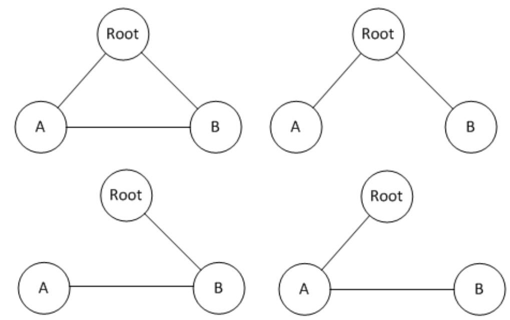

Figure 1: One physical topology - three logical tree topologies

A. Protocol Description

The topology resolved under MTA will have multiple paths between the root and other switches. These overlapping trees are created and maintained through the MT_VIDs. A Meshed Tree Switch (MTS) that has membership on a tree will be assigned an MT_VID that is associated with that tree and a particular path back to the root. Critically, switches having more than one pathway back to the root will have primary, secondary, tertiary, etc., memberships in multiple trees, each having a separate and unrelated MT_VID. MT_VIDs are stored in a table and have an association with ports through which they were established. Examples of trees from a single root and associated MT_VIDs are shown in Figure 2.

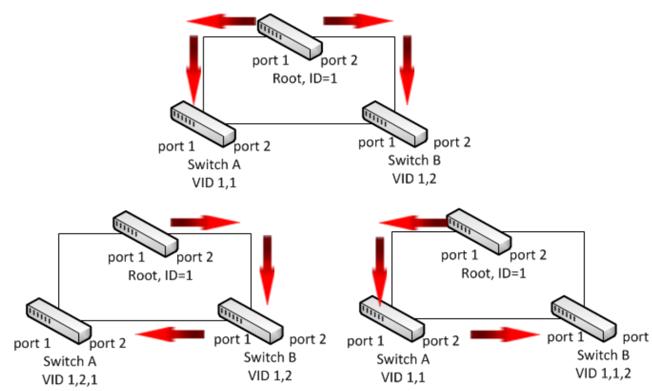

Figure 2 MT topologies and MT_VID Creation

On the top we can see that the topology is optimized to the root. The MT_VIDs (identified in the figure as VIDs) and the tree are derived based on this perspective. However, in a looped topology, the downstream or child switches have alternate paths. In the bottom left and bottom right we see the MT_VIDs that would be derived in these alternate logical topologies.

Another way to look at this is to consider the traffic that might flow between switches A and B. Clearly the topology that would be derived per spanning tree would be suboptimal. It is noteworthy that these alternate paths might be used to optimize transmissions between the hosts connected to the switches. So, another important aspect of MTP is that meshed tree switches do not possess source address tables or SATs. Instead they use a virtual SAT or

 ISBN: 978-1-61208-330-8

VSAT. MAC addresses of nodes connected locally will be learned in much the same way as described in 802.1D. Neighboring switches can exchange VSAT information in order to obtain more efficient pathways to the end hosts. This is possible as the MTP does not block ports. Within the VSAT, nodes are associated with an MT_VID for forwarding. Ports connecting the switch to a host are the *Host* ports. A port connecting a switch to another switch participating in the MTP is called an MT port because it is active in the MT topology. Port roles are shown in Fig. 3.

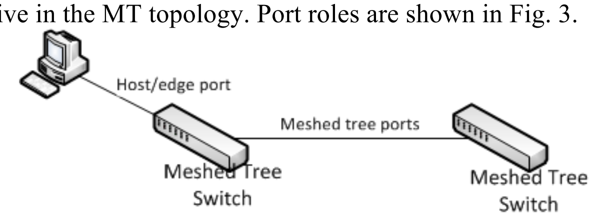

Figure 3 – Meshed Tree Switch Port Roles

B. *Basic Protocol Operation*

Switches join a meshed tree topology by either advertising themselves or hearing an advertisement from another MTS. Switches exchange *Hello* messages and establish an MT_VID. The MT_VID is derived from the parent MTS and the port transmitting the *Hello* message. This is explained with two switches in Figure 4.

Once all switches have at least one MT_VID, the forwarding topology can be viewed as an MT_VID tree. One of these MT_VID trees will be identified as the primary VID (PVID) tree. Unknown MAC addresses, broadcast and multicast traffic will be forwarded via this tree.

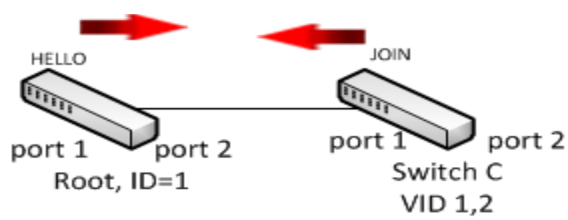

Figure 4 – Meshed Tree Hello and Join Process

Once switches have joined the MT topology and understand their parent and child relationships via the MT_VID, they exchange information contained in their VSATs via *VSAT Update* messages (VUM). Upon receipt, the VSAT in the receiving switch is modified in order to provide optimized forwarding to destination host MAC addresses. In more complex topologies, there will be superior pathways between some hosts and these can easily be identified through the VID structure. For example, parent and child switches are direct neighbors.

On discovery of a link failure or other problem, the meshed tree topology responds by deleting MT_VIDs from a Switch's MT_VID table and any VSAT entry associated with the lost MT_VID. Because redundant paths are permitted, the topology may have an alternative pathway immediately available. This path may now be elevated to the PVID. Generally speaking, shorter MT_VIDs are preferred as they represent a shorter path, though allowance for cost can be implemented.

Broadcast Packets: For forwarding broadcast packets or packets to unknown destinations, the switches should associate the MT_VIDs to the ports through which they were acquired. Thus, when forwarding to an MT_VID, the switch is correctly and efficiently forwarding the frame. Non-root switches forward broadcast frames using the following guidelines; If the broadcast frame is received from the port of PVID, it is sent out on all ports that have an MT_VID derived from the PVID and all host ports. However, if the broadcast frame is received from any other port, it is sent out on ports associated with the PVID and all host ports.

IV. OPTIMIZED FORWARDING

All switches that have MT_VIDs populate a VSAT that is indexed by Host MAC address. Locally connected hosts are added to the VSAT and in this case the port field is populated with the local switch port. Hosts connected to other switches will be represented in the VSAT with a field listing all of the MT_VIDS of switches handling traffic for the hosts. This indicates that at VSAT entry for a host may have more than one possible pathway back to the host. For non-local hosts the port field will also contain the egress port for packets destined for that host MAC address. Every time a VSAT entry is changed the forwarding port field is updated to reflect this change.

A. *VSAT Update Message*

When a Host leaves, its timer expires, or when a new host connects on a port, the switch creates a VSAT Update Message (VUM) and sends the VUM as shown in Fig. 5. A VUM;

- Includes only the changes to the VSAT
- Is sent out on all MT ports using an MT multicast destination address
- Includes Host MAC addresses and list of MT_VIDs of the associated switch
- Includes a flag to indicate addition or removal
- Contains a sequence number to avoid duplication of activity and ordering

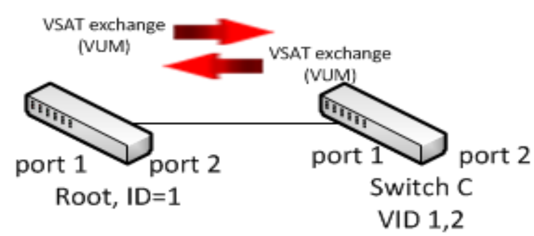

Figure 5 – VSAT updates

For each Host MAC address in the received VUM, MTS processes the message as follows;

- If the information is different than an existing VSAT entry – replace if the VUM sequence number is higher

- If not already in the VSAT – add an entry
- If a matching entry exists in the VSAT – do nothing

If changes were made to the VSAT, the switch creates a new VUM to reflect the changes and multicasts the VUM on all MT ports except the port that received the change. In this way, all of the switches in the topology learn of the VSAT changes.

C. Frame Delivery Process

Following cases can occur when forwarding frames:

1. Destination is this switch
2. Destination is in this MT branch away from root
3. Destination is in this MT branch towards root
4. Destination is in a different MT branch off of a switch towards the root
5. Destination is in another MT branch off of the root
6. Destination is on a switch that no longer exists
7. Destination has moved

The switch receiving a frame to forward will look up the destination MAC in its VSAT to obtain the switch's MT_VIDs associated with the Host. The switch must then follow a standard decision tree.

Case1: Is there an exact match in the local MT_VID table to the destination host switch's MT_VID?

- YES – the host is on the local switch and the frame has to be delivered through the local port.
- NO – frame forwarding will be handled by one the following cases

Case 2: Find shortest entry in the forwarding switch's MT_VIDs that is a parent (or grandparent, etc.) to the destination MT_VID. Select the next digit from the MT_VID after the matching pattern – this will be the port to forward the frame.

Case 3: Find shortest entry in the forwarding switch's MT_VID for which the destination switch's MT_VID is a parent (or grandparent, etc.). If there is a tie, pick one. Retrieve the port from the VID table – this will be the port to forward the frame.

Case 4: Find an entry in the forwarding switch's VID list that has a common parent (or grandparent, etc.) with the destination switch's MT_VID. This will resolve to the forking switch that leads to the destination. When that switch receives the frame it will use case 3 to direct the frame down the correct branch.

Case 5: This is a special instance of Case 4 where the common parent (or grandparent, etc.) is the root switch. When the root switch gets the frame it will follow case 2 to determine correct branch to send the frame on.

The above process can be executed on receiving a VUM and the ports associated with the host MAC address can be populated in the VSAT. A typical VSAT entry would be as shown in Fig. 6.

MAC	port	VID
00:01:02:03:04:05	23	1,1 1,2,3

Figure 6. VSAT entry

V. SIMULATIONS AND PEFORMANCE

The models for MTP were developed in OPNET using two scenarios; one with four switches and 1 loop, the other with six switches and 2 loops. For comparison the OPNET model for RSTP was utilized. The following performance parameters were targeted; ;.

MTP Single Tree Creation (MSTC) Time: The interval required for all switches to receive at least one MT_VID and can start forwarding frames.

MTP Meshed Tree Creation (MMTC) Time: Each Switch was allowed a maximum of three MT_VIDs. The time taken by all switches to record a maximum of the three different best paths was recorded. In MTP this would be the time when on link failures the backup paths can be used without new tree resolution.

MTP VSAT Update (MVSAT) time: The time taken for all switches to record a path to all hosts subsequent to receiving VUMs. At this time unicast frames can be optimally forwarded via other VIDs other than the primary.

RSTP initial convergence (IC) time was recorded when the spanning tree was formed. RSTP broadcasts unicast frames to unknown destinations at this time, as learning time is removed to improve convergence time.

Maximum hops taken by frames.

The converged topologies for MTP and RSTP in the case of the 4-switch scenario are shown in Fig. 7.

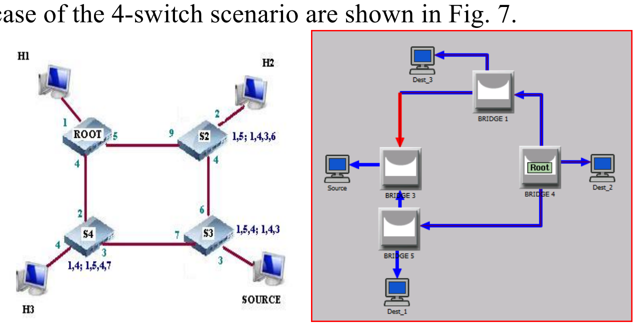

Figure 7. Meshed trees (left), spanning tree (right)

The MT_VIDs in [Figure 7] identify the three trees on which switches S2, S3 and S4 reside. The red line indicates the blocked port in the spanning tree. A host was connected to every switch. One host was identified as the source, which sent packets continuously, while the other hosts sent only for 3 seconds from the start of the simulation.

Packet exponential inter-arrival time at the hosts was set to 0.01 sec. At the switches the control traffic service rate was set to 100,000 packets per sec, while the data traffic service rate was 500,000 packets per sec. Duplex Link speed were maintained at 100 Mbps. Packet sizes were1500 bytes. The duration of simulation was set to 20 secs.

A. 4-Switch Single Loop Scenario

In this scenario, MSTC was recorded as 0.000037 sec, MMTC = 0.000047 sec, while MSAT was 0.0209882 sec. In the case of RSTP, IC was recorded to be 0.55 seconds. In MTP even if we avoided the flooding during the time that switches learn the host addresses through VUMs, the improvement in convergence is 26 times faster than RSTP. If we allow for frame flooding then the convergence time improvement is several thousand times. The hops taken by packets in MTP were recorded to be a maximum of 3 hops. In the case of RSTP the maximum hops would be 4.

Table 1: CONVERGENCE TIMES IN MTP

SEED	MSTC	MMTC	MSAT
127	0.000037	0.000047	0.028708
317	0.000037	0.000047	0.007826
509	0.000037	0.000047	0.024935
1009	0.000037	0.000047	0.019308
1721	0.000037	0.000047	0.024164

Note in Table 1, for seed 317, the MSAT was as low as 0.007826. The reason for the variance; when the switch gets the first data packet, it may not have had an MT_VID and hence that packet would have been discarded. The arrival of the second data packet would depend on the seed since the inter-arrival time for data packets is an exponential distribution. So if the second data packet were to trigger VSAT updates from some of the switches, the convergence time would be different for different seeds. Hence this convergence time depended on the packet inter-arrival at the host. If the inter-arrival were low then the MSAT would be also very low.

B. 6-Switch – Two Loop Scenario

In this scenario, the MSTC, MMTC and MVSAT were recorded to be 0.000047 sec, 0.000070 sec and 0.0225622 seconds. The RSTP IC time was 0.56 seconds. MTP records several thousand times improvement if packets could be forwarded before learning end host addresses (i.e. without a VSAT update) and 24 times better after all host addresses were recorded in all switches. The hop counts for packets were recorded to be 6 hops as compared to a maximum of 4 hops with MTP.

The convergence times noted and the hop counts depend on the topology. With more complex and meshed topologies the convergence times and hop counts can vary significantly. For example, in a full meshed topology the maximum hop count for frames in MTP would be 2, whereas for RSTP the frames will have to travel through the root switch. The control message overhead and excess traffic due to frame flooding also would significantly differ.

C. Comparison with Link State Protocols

In the case of TRILL on RBridges and SPB, optimal pairwise paths are computed and used for frame forwarding. However, the processing complexity has increased by several magnitudes. In the case of single meshed tree MTP, optimal paths can be computed based on the MT_VIDs acquired by the switches. Since switches may not record all MT_VIDs offered, some paths may not be the shortest.

In terms of convergence, link state routing requires that all link state information to be flooded to all switches. Subsequently the Dijkstra algorithm will be run to compute the forwarding paths. During this time the SAT may not be updated and could result in unstable operation. Comparatively in MTP, the tree is built using information received from neighbor switches and flooding of information is avoided for tree resolution. In the event that tree pruning is required, the switches can still use the backup paths to forward frames.

VI. Conclusions and Future Work

Loop free forwarding in networks with redundant paths has been hitherto addressed on the premise that a single logical tree topology originating from a root switch is essential. This resulted in the spanning tree algorithm, which had high convergence delays. This was addressed by RSTP, which continued to face several disadvantages as stated by their inventors. More complex IS-IS based routing solutions are being adopted at layer 2. This article describes a simple solution that can replace STA algorithm at layer 2, without its disadvantages, while at the same time avoid the complexity from adopting layer 3 routing solutions at layer 2. Specification of the MTP is currently being developed under a new IEEE standard [8].

MTP performance has been compared with RSTP in terms of convergence times and path hop counts taken by framed. The superior performance achieved with MTP can be noted from these results. These results can also be used as benchmark when TRILL and SPB are evaluated.

References

[1] LAN/MAN Standards Committee of the IEEE Computer Society, ed. (1998). *ANSI/IEEE Std 802.1D, 1998 Edition, Part 3: Media Access Control (MAC) Bridges*. IEEE

[2] W. Wodjek, "Rapid Spanning Tree Protocol: A new solution from old technology", http://www.redes.upv.es/ralir/MforS/RSTPtutorial.pdf Reprinted from CompactPCI Systems / March 2003 [Retreived Feb, 2014]

[3] R. Perlman, D. Eastlake, G. D. Dutt and A. G. Gai, "Rbridges: Base Protocol Specification", RFC 6325, July 2011.

[4] J. Touch, R. Perlman, "Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement", RFC 5556.

[5] N. Shenoy, Y. Pan, D. Narayan, D. Ross and C. Lutzer, "Route robustness of a multi-meshed tree routing scheme for internet MANETs", Proceeding of IEEE Globecom 2005. 28th Nov – 2nd Dec. 2005 St Louis, pp. 3346-3351.

[6] N. Shenoy and S. Mishra, "Multi Hop routing and load balancing in Mobile Ad hoc networks", Book Chapter in Encyclopedia on Ad Hoc and Ubiquitous Computing, Published by World Scientific Book Company, 2008.

[7] Network Simulation (OPNET Modeler Suite), Riverbed Technologies

[8] 1910 Working Group for Loop-Free Switching and Routing, Project 1910.1 Standard for Meshed Tree Bridging with Loop Free Forwarding.