Performance of Meshed Tree Protocols for Loop Avoidance in Switched Networks

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

Abstract—**Loop free frame forwarding in layer 2 switched networks that use meshed topologies to provision for link and path redundancy is a continuing challenge. The challenge is addressed through special protocols at layer 2 that build logical trees over the physically meshed topologies, along which frames can be forwarded. The first such protocol was based on the spanning tree. The spanning tree protocol (STP) had high convergence times subsequent to topology changes. Rapid STP and IETF RFC 5556 *Transparent Interconnection of Lots of Links* (TRILL) on Router Bridges (RBridges) were then developed to reduce the convergence times. RSTP continued to use the spanning tree while TRILL adopted link state routing to support a tree from every switch. TRILL introduces high processing complexity into layer 2 networks. In this article a new meshed tree algorithm (MTA) and a loop avoidance protocol based on the MTA, namely the meshed tree protocol (MTP) are discussed. The MTA allows constructing several overlapping trees from a single root switch. This speeds up convergence to link failures. The MTP proposes a simple numbering scheme to implement meshed trees – thus, the processing complexity is low. The specification for the MTP is currently an ongoing IEEE standard Project 1910.1. In this article the operational details of MTP are presented and its performance evaluated and compared with RSTP.**

Keywords- Loop Avoidance, Switched Networks, Meshed Trees Protocol, Link Failure and Recovery

I. INTRODUCTION

Loop free forwarding is a continuing challenge in layer 2 switched networks. The need for link and path redundancy to provide a continuous communications path between pairs of end switches in the event of switch or link failure requires a physical network topology that is meshed. However, the physical loops in a mesh topology cause broadcast storms when forwarding broadcast frames. Hence, it is important to have a logical tree topology overlaid on the physical meshed topology to forward broadcast frames. The first such logical loop-free forwarding solution was based on the Spanning Tree. Radia Perlman [1] proposed the specifications of a protocol called the Spanning Tree protocol (STP) based on the Spanning Tree Algorithm (STA). A spanning tree in a switched network was constructed by logically blocking some of the switch's ports from forwarding frames. The basic STP had high convergence times during topology changes. Rapid Spanning Tree Protocol (RSTP) was developed to reduce the convergence times in the basic STP. However, RSTP still retained some of the inefficiencies of spanning trees, one of which is forwarding all frames through the root and a second is the root re-election on topology changes. Radia Perlman then proposed Transparent Interconnection of Lots of Links (TRILL) on RBridges (router bridges) to overcome the disadvantages of STA-based loop avoidance. This came at the cost of processing overhead and implementation complexity as TRILL used the Intermediate System to Intermediate System (IS-IS) routing protocol at layer 2. The goal was to use optimal paths for frame forwarding between pairs of switches and also avoid root election. IS-IS is a link state routing protocol that can operate independently of the network layer. The TRILL protocol was implemented above layer 2 and used special headers to encapsulate the TRILL and IS-IS related link state routing messages. The bridges were called RBridges as they implemented routing protocols. TRILL on RBridges is currently an Internet Engineering Task Force (IETF) draft [2]. Shortest Path Bridging (SPB) was developed along similar lines by IEEE 802.1aq, also adopting the IS-IS routing protocol at layer 2. TRILL is considered as superior to RSTP due to the redundant links that are established. Shortest Path Bridging (SPB) based loop avoidance primarily targets high switching speeds as required in service provider and backbone provider networks. Thus, two versions of SPB have been specified, SPBV (SPB with VLAN Ids) and SPBM (SPB with MAC addresses) [3] for service provider networks and backbone provider networks, respectively.

The premise for the loop avoidance solutions discussed above 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 frame forwarding. Under this approach, 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 used in IS-IS based routing builds a tree from every switch. The operation of IS-IS requires link state information in the entire network to be disseminated to every switch so that each switch can compute its own tree by running the Dijkstra algorithm on the connectivity information that it collects and stores in a Link State database. On topology changes, link state information must again be disseminated to all switches and the Link State database should be stable for some time before the Dijsktra algorithm can be run. During this period the frame forwarding information is unstable. TRILL on RBridges uses a hop count to avoid looping of frames.

In [4], a novel meshed tree algorithm (MTA) and the associated Meshed Tree Protocol (MTP) was introduced. Its performance was evaluated and compared with RSTP. Unlike the trees discussed above the MTA allows construction of multiple trees from a single root by using the multiple paths provisioned by the meshed topology. Loop-free frame forwarding can happen using any one of the multiple trees. The MTP based on the MTA allows for creation and maintenance of *multiple* overlapping tree branches from *one* root switch. The multiple branches mesh at the switches, and thus on the failure of a link (or branch) the switch can fall back on another branch without waiting for re-computation of the tree. Frame forwarding can continue while the broken branch is pruned. This eliminates temporary inconsistent topologies and latencies resulting from tree reconstruction. The premise of the MTP is to leverage the multiplicity of connections in a meshed topology by constructing and maintaining several trees from a single root concurrently [5-9]. Thus, the MTA addresses the convergence issues facing STA based protocols and also avoids the complexity of IS-IS based loop avoidance solutions. In addition, there can be multiple root switches, where each root supports its own meshed trees. This extends the MTA to Multi Meshed Trees (MMT), which can be used to introduce redundancy in the event of root failure. This feature of MTP is not covered in this article.

The novel feature of the MTA is implemented through a simple numbering scheme. Meshed Tree Virtual IDs (MT_VIDs) are allocated to each switch in the network. The MT_VID acquired by a switch defines a tree branch or logical frame-forwarding path from the root switch to that switch. A switch can acquire multiple MT_VIDs based on the MT_VIDs advertised by its neighboring switches and thus join multiple tree branches providing a switch with several paths to the root switch. In this way, meshed trees leverage the redundancy in meshed topologies to set up several loop-free logical frame-forwarding paths. No ports are blocked from forwarding frames.

In this paper, some basic operational specifications of the MTP are presented. These include meshed tree creation through the use of MT_VIDs, the limits on the level of meshing, the criteria and process for a switch to forward broadcast and unicast frames, and the handling of link failures. The specification of the MTP in this article is limited to customer VLANs where RSTP is the primary candidate solution. Thus, the performance of the MTP is evaluated and compared with RSTP. The comparison was conducted using OPNET simulation tool [10]. Though TRILL is considered as another candidate protocol for RSTP replacement, models of TRILL were not available for a comparative study. However, under Section II.D a detailed operational comparison of TRILL with the MTP is provided.

The significant improvement in the convergence times and the hops taken by frames to reach destinations indicate the superior features of the MTP. The operational simplicity of the MTP also provides advantages over complex Link State solutions. MT loop free forwarding at layer 2 is currently an IEEE project (1910.1) under the IEEE 1910 working group [11] lead by the authors. 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 MTP are presented. Section IV describes the optimized unicast frame forwarding schema adopted in the MTP. Section V provides the link failure handling mechanism adopted in the MTP. Section VI provides the simulation details and performance results. Section VII follows with conclusions.

II. Related Work

In this section, we discuss the two primary techniques proposed for loop resolution in layer 2 switched networks. The first of these is based on the Spanning Tree Protocols (STP and RSTP) and the second is based on Link State (LS) Routing namely the TRILL on RBridges. This article does not describe all the operational details as such information is publicly available [12-14].

A. Protocols Based on Spanning Tree Algorithm

Both the STP and RSTP are based on the STA. To avoid loops in the network while maintaining access to all the network segments, the bridges compute a spanning tree after collectively electing a root bridge. For root election bridgeIDs are used. In (R)STP, each bridge first assumes that it is the root and announces its bridgeID. Upon receiving the bridgeID, neighbors compare it with their bridgeID and allow the bridge with a lower bridgeID to continue as a root. The unique bridgeID is a combination of a bridge priority and the bridge's medium access control (MAC) address. A bridge may supplant the current root if its bridgeID is lower.

One major disadvantage of STA based protocols is that all traffic flows via the root switch. It is thus important to have a root switch that has adequate processing capability and an optimal location within the topology. For this purpose the priority field in the bridgeID can be manually set by an administrator. 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 include the path cost information from the root bridge. Bridges accept a connection to another bridge based on the lowest path cost. With the STP, other ports are blocked from frame transmission. Within a network deploying RSTP these ports are maintained in readiness (alternate, backup) to takeover on the failure of the unblocked ports in RSTP.

The STP has high convergence times after a topology change. To reduce the convergence times the *Rapid Spanning Tree* protocol (RSTP) was proposed [12]. The RSTP is a refinement of the STP and therefore shares most of its basic operation characteristics, with some notable differences including: 1) the detection of root bridge failure is done in 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. The second feature 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 and bypass the traditional listening/learning states. 3) Lastly, backup details regarding the discarding status of ports are maintained to avoid failure timeouts of forwarding ports.

STP and RSTP: STA based implementation is simple as the spanning tree is executed with the exchange of BPDUs among neighboring bridges that carry *tree formation* information. Several disadvantages of STA based protocols are noted by the inventors of STA [14]. These include: 1) Traffic concentration on the spanning tree path, as all traffic follows the tree 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 logical spanning tree topology. Changes in the spanning tree take time to propagate and converge, especially for non-RSTP protocols. 3) Though IEEE 802.1Q describes multiple spanning trees, this requires additional configuration, the number of trees is limited, and the defects previously noted apply within each tree [3].

B. TRILL Protocol on RBridges

The TRILL protocol overcomes many of the shortcomings in STA based protocols. Convergence times are improved by supporting a tree from each switch. TRILL incorporates the routing functionality of layer 3 by using the IS-IS protocol [13, 14] at layer 2. The IS-IS protocol is used to compute pair-wise optimal paths between two bridges. The computed pair-wise optimal paths is used for forwarding frames at layer 2. Thus, the frame forwarding inefficiency in STA based protocols is avoided. Inconsistencies and loop formations during topology change can occur but are overcome by a *hop count* used in inter-bridge forwarding. TRILL encapsulates link state routing messages of IS-IS in special headers and uses special protocols to learn end station addresses.

Advantages and Disadvantages of TRILL: Advantages of the TRILL protocol include: 1) Frames are forwarded via an optimal path. 2) Transit frames are routed with a hop count, thus 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. The disadvantages of IS-IS based protocols are: 1) They have to encapsulate all messages required for the operation of IS-IS. 2) The operations of IS-IS are distinct from layer 2 operations and VLANs in layer 2. This adds to the processing complexity at layer 2 which is compounded by the need for integrated operations of layer 2 and IS_IS routing functions. 3) All link state routing protocols require that Dijkstra algorithm be run only after the Link State database has stabilized for a certain time interval after the last link state update received. During this time the forwarding (routing) operations are unstable and this contributes to the convergence time of the network topology at layer 2. During this time, looping packets cannot be avoided which required IS-IS based solutions to include a hop-count to discard such packets. Dijsktra algorithm is also known for its processing complexity [15], which is proportional to the number of switches / links.

C. The Meshed Tree Protocol

Single-tree like structures imposed on topologies 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 build trees from all nodes to alleviate these problems. However, as redundancy is introduced the complexity becomes very high due to the creation and maintenance of as many trees as there are switches. The 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 switch to be a member of more than one tree. This provides path redundancy and quick fail-over to the redundant paths on link failure detection. Ports are not blocked which allows for optimized frame forwarding paths. Root redundancy requirements in single meshed-tree based on the MTP can be addressed by multiple meshed-trees (MMT) [5 - 9], where several switches can be roots and each can support a meshed tree. The number of roots can be optimized to improve redundancy and performance while keeping the complexity low.

D. Comparison with Link State Protocols

In the case of TRILL on RBridges optimal pairwise paths are computed and used for frame forwarding. However, the processing complexity has increased by several orders of magnitude. 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 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 source address table (SAT) may not be updated and could result in unstable operation. Using the 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.

Table I lists the major difference between TRILL and MTP.

Table I. Comparison of 'MTP on Bridges' vs 'TRILL on RBridges'

Feature	TRILL on RBridges	Meshed tree on bridges
Tree structure	• One shortest path spanning tree originating at the root Rbridge • Each Rbridge is present on only one branch of a single tree originating from a root bridge	• Several overlapped spanning trees with one of them being the shortest path spanning tree • Each bridge can reside on multiple branches of a single meshed tree originating from a root bridge
Multiple trees originating at different bridges	Possible	Possible
Knowledge of network topology	required	NOT required
Flooding of topology messages	required	NOT required
Action on link failure and addition /removal of bridges and links	• Generate link state updates and disseminate. • Flood topology control messages	• Repair locally. • Inform bridges downstream that have an MT_VID which is derived from the lost MT_VID. • Build tree branches as nodes join
Formation of temporary loops	Yes. Loop is broken when hop count (6 bits in the header) reaches 0.	Loop formation prevented
Avoidance of loop formation	Not completely avoided. Uses hop counts	Avoided due to the numbering scheme
Unicast frames (known destination address)	• Forwarded on pair-wise optimal paths determined by the link state routing protocol if End System Address Distribution Information (ESADI) is used. • Next hop path should be specified. • Encapsulated in TRILL header • Every Rbridge that forwards decapsulates and encapsulates again	• Neighboring bridges can forward directly to the appropriate port. • Forwarded on the optimal path decided by primary VID tree at the originating bridge.
Multicast traffic Unicast frames (destination unknown)	• Forwarded on distribution trees, using multi pathing to multiple destination. • Tree pruning advised (no specifications provided)	• Can follow the current process using multicast addresses at layer 2. • Meshed tree at originating bridge can be used.
End node address learning	• Open the internal Ethernet frame to determine the source address • Use ESADI protocol and inform all RBRridges	• Learn from source address as no encapsulation is used • Can exchange infromation between neighboring switches.
Computing complexity of Dijkstra's algorithm	• O(n^2) in a dense network for node selection with 'n' nodes. • O(m) for edge (link) updates with 'm' edges • O(m log n) using an adjacency list representation and a partially ordered tree data structure for organizing the set of edges [15].	O(1) –See Appendix A
Implementations	• Dynamic nickname protocol to reduce TRILL header • Topology control message dissemination • Encapsulation and de-encapsulation at forwarding Rbrdiges. Every transit frame has to be encapsulated with an external Ethernet header. Overhead per encapsulation equals 144 bits • End Station Address Dissemination Information (ESADI) protocol is optional • Election of a designated Rbridge per link • Designated VLAN required for Rbridge communication • Differentiate between IS_IS at layer 2 and layer 3 • Requires 'reverse path forwarding check" to control looping traffic **See schematic in Appendix B**	• Replace the ST algorithm with the MT algorithm. • Define software to run the MT algorithm • Works on the same principle as STA. MT_VIDs will be sent in BPDUs. **See schematic in Appendix B**

III. The Meshed Tree Protocol

The *MTA* allows construction of logically *meshed trees* from a single root switch in distributed manner using local information shared among neighbor switches [5-9]. In this article, MTP operations related to the construction of meshed trees are described. The discussion presented in this article does not include the election of a root bridge as the focus is on the loop resolution / avoidance feature of MTA based protocols. Hence we assume a designated root bridge.

Bridge ID: For the operation of the MTP bridgeIDs are necessary. These have to be unique only within the switched network. The MT_VIDs of switches are derived by appending the outgoing port number to the MT_VID of the switch that offers an MT_VID to a downstream switch. The root switch uses its unique ID as its MT_VID, thus the first value in the MT_VID acquired by other bridges will be the root bridgeID. In this article without loss of generality we used a single digit ID for the root switch though a simple MAC address derivative could be used.

Because of the way in which MT_VIDs are constructed, an MT_VID describes a path that connects a bridge to the root bridge. In a single physical meshed 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 switch to each switch in the topology.
- More than one path to each switch can coexist

Consider a three-switch single loop topology shown in Fig. 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. By themselves, these three logical topologies do not provide for redundancy. The MTA allows for building and using all of the logical trees simultaneously and because multiple pathways are pre-established, *failover times to redundant links are near zero.*

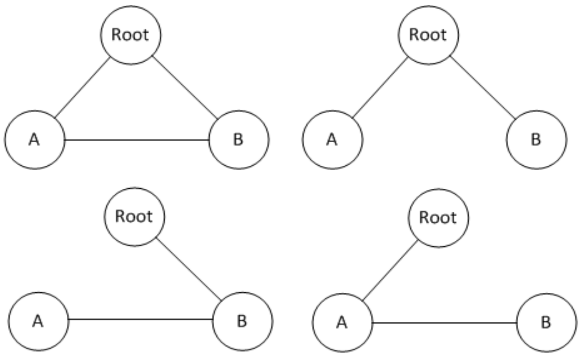


Figure 1. One physical meshed topology - three logical tree topologies

A. Basic Protocol Operation

The topology resolved using the MTP will support overlapping trees that are created and maintained through the MT_VIDs. A Meshed Tree Switch (MTS) that has membership on a tree will be assigned at least one MT_VID that is associated with that tree and a particular path back to the root. Significantly, 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 Fig. 2.

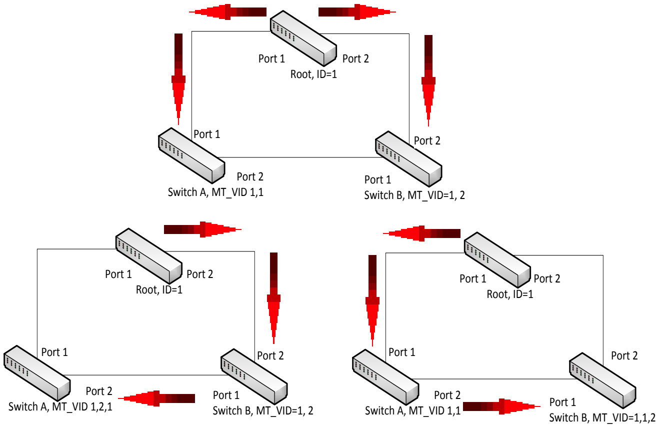


Figure 2. MT topologies and MT_VID Creation

On the top of Fig. 2 it can be seen that the topology is optimized to the root. The MT_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, switches A and B also have MT_VIDs that would be derived in these alternate logical tree 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 as all traffic must first flow to the root switch and then back down. It is noteworthy that these alternate paths might be used to optimize transmissions between the hosts connected to the switches. Another important aspect of the MTP is that MTS's do not populate the SAT in the traditional manner; learning the source addresses of end hosts based on the port upon which they arrive. Switches in the meshed tree topology share information regarding their directly connected hosts and this information is contained in a virtual SAT or VSAT. Using this information, the paths taken by frames can be optimized because each switch is aware of the switch MT_VID to which an end host is connected. The optimum path can be determined by comparing known MT_VIDs and ports with the VSAT entry. This is possible due to inherent attribute of MT_VIDs. MAC addresses of nodes directly connected to a switch will be learned in much the same way as described in 802.1D; when the hosts generates a frame and it arrives at a non-MTS or *host* port. Ports connecting the switch to a host are the *Host* ports. A port connecting an MTS 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. Switches populate their tables with addresses from local hosts and map it to their MT_VIDs. They then advertise the Virtual Source Address Table (VSAT) to the neighbors. All switches can exchange VSAT information with their neighbors and add learned information to their own VSAT. This is possible as the MTP does not block ports.

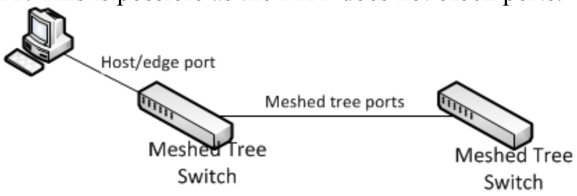


Figure 3. Meshed Tree Switch Port Roles

B. Messages in MTP

Switches join a meshed tree topology by either advertising themselves or hearing an advertisement from another MTS. Switches advertise their MT_VIDs using *Hello* messages. While advertising on a particular port they append the port number to their MT_VIDs and offer the MT_VID to a neighbor switch. A switch that accepts an MT_VID from an advertising switch responds with a JOIN message. Switches record the ports on which they hear the join message to retain the child MTS connected on that port. This information is useful in forwarding broadcast frames as described later in this section. The message exchange process is explained with two switches in Fig. 4. Once all switches have at least one MT_VID, the forwarding topology can be viewed as an MT_VID tree. When switches have acquired multiple MT_VIDs, one of these MT_VID trees will be identified as the primary MT_VID (PMT_VID) tree. Unknown MAC addresses, broadcast and multicast traffic will be forwarded via the PMT_VID tree.

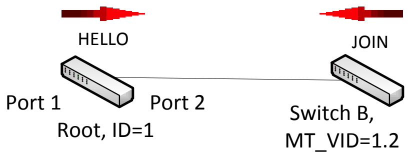


Figure 4. Meshed Tree Hello and Join Process

Once switches have joined the MT topology and understand their parent and child relationships via the MT_VIDs, they exchange information contained in their VSATs via a *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 MT_VID structure. For example, parent and child switches are direct neighbors and an optimal shortest path will exist unless otherwise defined differently due to path cost.

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. The MT_VID associated with this path may now be elevated to the PMT_VID. Generally speaking, shorter MT_VIDs are preferred as they represent a shorter path, unless the costs of the links define otherwise.

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

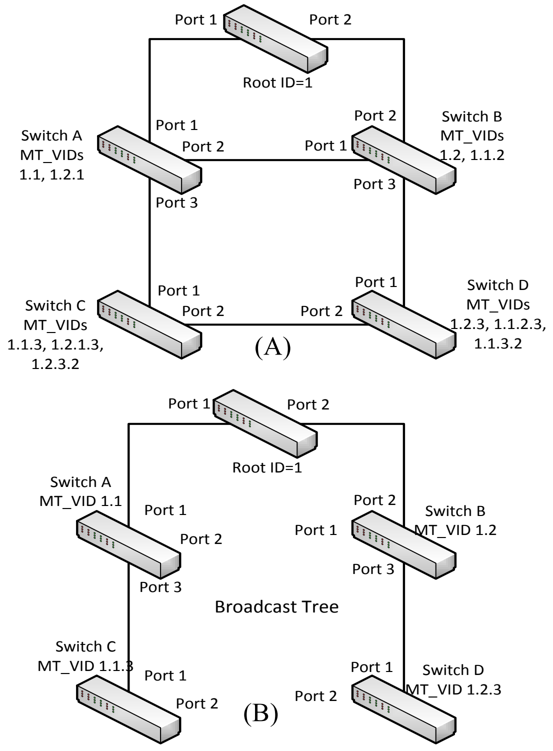


Figure 5. (A) Two Loop Meshed Topology with MT_VIDs.
(B) Broadcast Tree for the two-loop Topology

Fig. 5A shows a two-loop topology, which is an extension of Fig. 2 and includes switches C and D. Switches C and D each have acquired three MT_VIDs. Of the multiple MT_VIDs a switch records one MT_VID as the primary MT_VID or PMT_VID. The others are stored in order of preference. The MT_VID tables from all switches are shown in Table II. The shaded MT_VID is the PMT_VID as it has the lowest cost (hops in this case) to the root. In this article all links are assumed to be of equal cost. Based on this information, the PMT_VID tree or broadcast tree is shown in Fig. 5B. In the case of a tie, the MT_VID acquired first would be assumed to be the PMT_VID and this assumption would not impact the operations.

Table II . MT_VID Table at the Switches

Switch	MT_VIDs stored in order of preference
Root	1
A	1.1, 1.2.1
B	1.2, 1.1.2
C	1.1.3, 1.2.1.3, 1.2.3.2
D	1.2.3, 1.1.2.3, 1.1.3.2

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 that are directly connected to the hosts. This indicates that a 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. The algorithm used in this case is provided in sub-section B.

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.

A. VSAT Update Message

When a host leaves, its VSAT 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. 6.

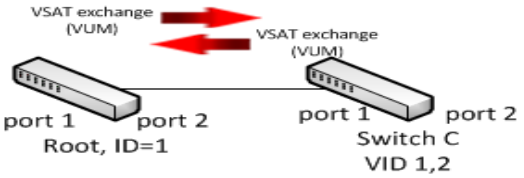


Figure 6. Exchange of Virtual Source Address Tables

A VSAT Update Message (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

For each host MAC address in the received VUM, the MTS processes the message as follows:

- If the information is different than an existing VSAT entry; replace if the VUM sequence number if higher
- If not already in the VSAT; add an entry
- If a matching entry exists in the VSAT; do nothing

B. Egress Ports for Frame Delivery

Following cases will be considered to determine the egress port.

Case 1: Destination is this switch, then the egress port is one of the host ports

Case 2: Destination is in this MT branch away from root. 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: Destination is in this MT branch towards root. 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: Destination is in a different MT branch off of a switch towards the root. 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: Destination is in another MT branch off of the root. 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.

On receiving a VUM, the above process will be executed 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. 7.

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

Figure 7. Virtual Source Address Table Entry

V. LINK FAILURE HANDLING BY MTP

Let the link between switches B and D in Fig. 5A fail at time *t*. The time taken by switch D to switch its PVIDs after link failure has been detected is constrained only by the internal hardware / firmware and MTP processing delays. Since MT_VIDs 1123 and 123 were acquired on port 1 of switch D (the port that detected the failure), MT_VID 1132 will take over as the PMT_VID. Switch D then sends a prune message to the switch that has an MT_VID acquired from switch D and derived from the MT_VIDs no longer available.

Switch C continues to use the other paths supported by its other MT_VIDs. In the case of switch B, as it has no MT_VIDs acquired from port 3 (the port that detected link failure) it makes no changes. The broadcast tree after pruning will look as shown in Fig. 8.

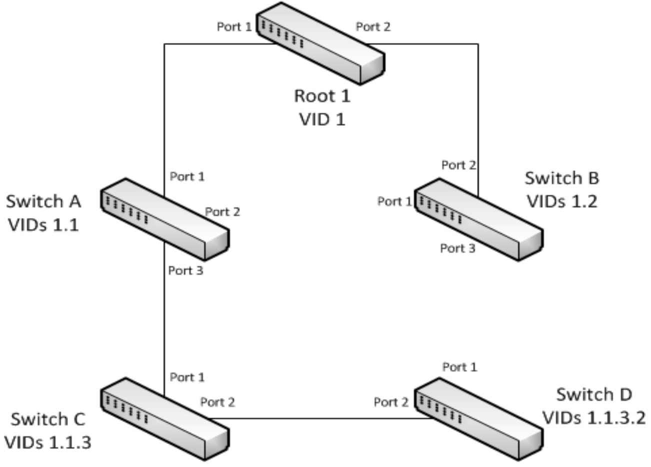


Figure 8. Broadcast Tree After Link Failure

A. Impact on Broadcast Packets:

The switchover of broadcast packets in the midst of PVID tree change will impact the performance. Current schemes do not conisder this. For example: It may happen that some broadcast packet, which should have gone via switch B to switch D may not reach switch D. If the PMT_VID tree at switch C resolves in a timely manner then it may forward the stored or intransit broadcast frames to switch D.

B. Impact on Unicast Packets:

The failover to a backup tree branch after a link failure should occur in near-zero time with MTP. In most cases any performance impact on unicast packets should be negligible. However, there could be cases where frames in transit may experience slight delays. For example a frame from an end node attached to switch C has reached switch D enroute to a node connected to switch B.

Switch D will redirect the frame back on to port 2 towards switch C, which will then forward the frame using another MT_VID. No disruption should occur if the tree pruning and VSAT update occurs before the frame is resent to switch C.

C. Link Failure Process

Two types of messages are used in MTP to detect link failures. The small periodic Hello message containing no information is sent out every 2 seconds to inform the neighbor switch of its continued participation in the MTP. When there are changes in the MT_VID, then *change* Hello messages are sent to inform the neighboring switches of the changes in MT_VID. Timers are used in the switches for the purpose. When the timer for a periodic Hello expires, the switch enters the “Timer Expired” state. The items of relevance are the node’s MT_VIDs and the associated port on the expired link. Following actions are taken.

- VSAT outgoing ports resolved to this port are recalculated and the next best MT_VID of the Destination host MAC address is determined using the algorithm described above.
- Any MT_VID that was received from that switch is flagged as unreachable. VSAT entries for the local hosts are adjusted if required and VSAT updates are sent.
- Since the MT_VID table has now changed, a change_hello packet is sent with the new active MT_VIDs. Downstream nodes will use this information to remove stale MT_VID entries, make corresponding VSAT changes and send VSAT updates.

After the link down event is detected or a timer expires, the first action is the recalculation of best outgoing port. At this point, packets will be forwarded to the correct destination even as the rest of the network heals. While the network converges, some packets may not follow the best possible route, but packet flow will continue. Thus, the convergence time will depend on the failure detection, i.e., the hello timer only. The failover time is almost negligible. When the backup paths can be used without new tree resolution.

VI. SIMULATIONS AND PEFORMANCE

The models for MTP were developed in OPNET. OPNET already had models for RSTP. The performance parameters targeted were the following. Two scenarios were used for the purpose; one with four switches and 1 loop, the other with six switches and 2 loops.

MTP Single Tree Creation (MSTC) Time: this was the time that all switches received 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: This is the time taken for all switches to record a path to all hosts subsequent to receiving VUMs. At this time unicast frames can be forwarded without broadcasting.

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.

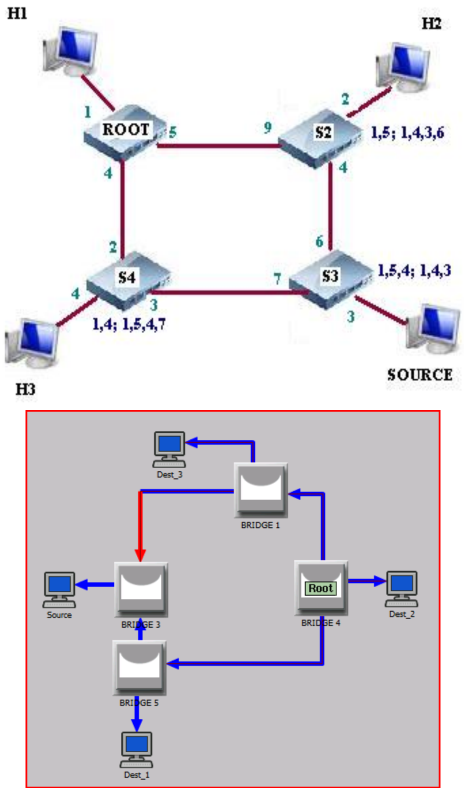


Figure 9. Meshed trees (top), spanning tree (bottom)

The MT_VIDs in Fig. 9 identify the three trees on, which switches S2, S3 and S4 reside. The red line in the picture shows 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 the MTP even if flooding of traffic was avoided during the time that switches learn the host addresses through VUMs, the improvement in convergence is 26 times compared to RSTP. If we allow for frame flooding then the convergence time

The resolved topologies for MTP and RSTP in the case of the 4-switch scenario are shown in Fig. 9. improvement is several thousand times. The hops taken by packets in the MTP were recorded to be a maximum of 3 hops. In the case of RSTP the maximum hops would be 4.

Table III. Convergence In MTP – One-Loop Topology

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 III, for seed 317, the MSAT was as low as 0.007826. The reason for the variance is: 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 as recorded in Table IV. The RSTP IC time was 0.56 seconds. MTP records several thousand times improvement if packets could be forwarded before learning end host addresses 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.

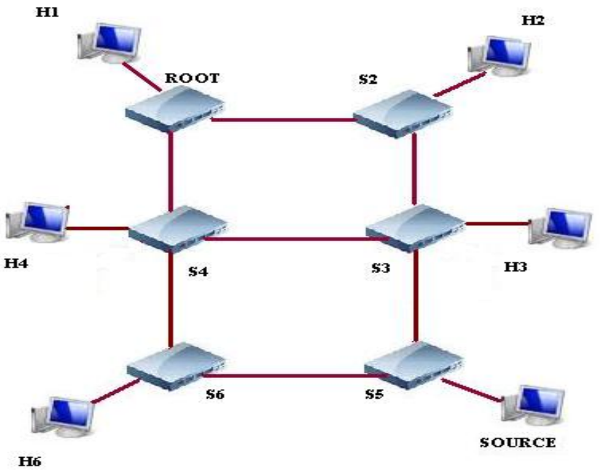


Figure 10. 6-switch, two-loop scenario

Table IV. Convergence In MTP – Two -Loop Topology

SEED	MSTC	MMTC	MSAT
127	0.000047	0.000070	0.033301
317	0.000047	0.000070	0.020941
509	0.000047	0.000070	0.024952
1009	0.000047	0.000070	0.016955
1721	0.000047	0.000070	0.016662

VII. Conclusions

Loop free forwarding in networks with redundant paths has been previously addressed with the premise that a single logical tree topology originating from a root switch is essential. This resulted in the STA based STP, which had high convergence delays. This was improved by RSTP, which continued to face several disadvantages as stated by the protocol inventors. More complex IS-IS based routing solutions are being adopted at layer 2. This article describes a simple solution that can replace (R)STP at layer 2, without its disadvantages, while at the same time avoid the complexity of using layer 3 routing at layer 2. The specification of the MTP is currently being developed under a new IEEE standard [11].

In this article the MTP performance has been compared with RSTP in terms of convergence times and path hop counts taken by frame traffic. The superior performance achieved with the 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 Standard.

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

[3] Standard IEEE 802.1Q, Media Access Control Bridges and Virtual Bridged Local Area Networks.

[4] K. Sharma, B. Hartpence, B. Stackpole, D. Johnson and N. Shenoy, "Meshed tree protocol for faster convergence in switched networks," IARIA Sponsored Tenth International Conference on Networking and Services, April 20-24, Chamonix, France, pp 90-95.

[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 to 2nd Dec. 2005 St Louis, pp. 3346-3351.

[6] N. Shenoy and S. Mishra, "Multi-Hop, Multi-Path and Load Balanced Routing in Wireless Mesh Networks," Book Chapter in Encyclopedia on Ad Hoc and Ubiquitous Computing, Published by World Scientific Book Company, 2008.

[7] N. Shenoy, Y. Pan and V.Reddy "Quality of service in Internet MANETs*,"* Proc. of IEEE 16th Intl. Symposium on Personal, Indoor & Mobile Radio Communications (PIMRC), 2005, pp. 1823-1829.

[8] N. Shenoy and Y. Pan. "Multi-meshed tree routing for Internet MANETs," Proc. of 2nd International Symposium on Wireless Communication Systems, pp. 145-149.

[9] S. Pudlewski, N. Shenoy, Y. Al-Mousa, Y. Pan Y and J. Fischer, "A hybrid multi meshed tree routing protocol for wireless ad hoc networks," Second IEEE International Workshop on Enabling Technologies and Standards for Wireless Mesh Networking, September 29, 2008. Atlanta, GA, USA, pp 635-641.

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

[11] 1910 Working Group for Loop-Free Switching and Routing, Project 1910.1 Standard for Meshed Tree Bridging with Loop Free Forwarding.

[12] 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 Dec, 2014]

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

[14] R. Perlman, "Rbridges: Transparent Routing," IEEE Proceedings of Infocomm 2004, pp 1211 -1218.

[15] http://mathworld.wolfram.com/DijkstrasAlgorithm.html. [Retreived Dec, 2014]

Appendix A: Computational Complexity of 'Meshed Tree Algorithm'

Assume root is elected, which will not be a consideration if some bridge is already designated to be 'root' or if all bridges would like to set up their own 'meshed tree' as under multi-meshed trees

- We can set a limit on the length of a tree branch and number of 'VIDs' that can be derived from a single bridge without loss of generality. Hence
  a. Let the number of maximum hops in a tree branch from a root node be ≤ B
  b. Let the number of derived MT_VIDs that each bridge can allocate be ≤ C
  c. Bridged network size (Number of bridges in the meshed tree) $\leq 1+C^1+C^2+\ldots+C^B$
- The convergence occurs in $N_{iter} = O(1)$ iterations

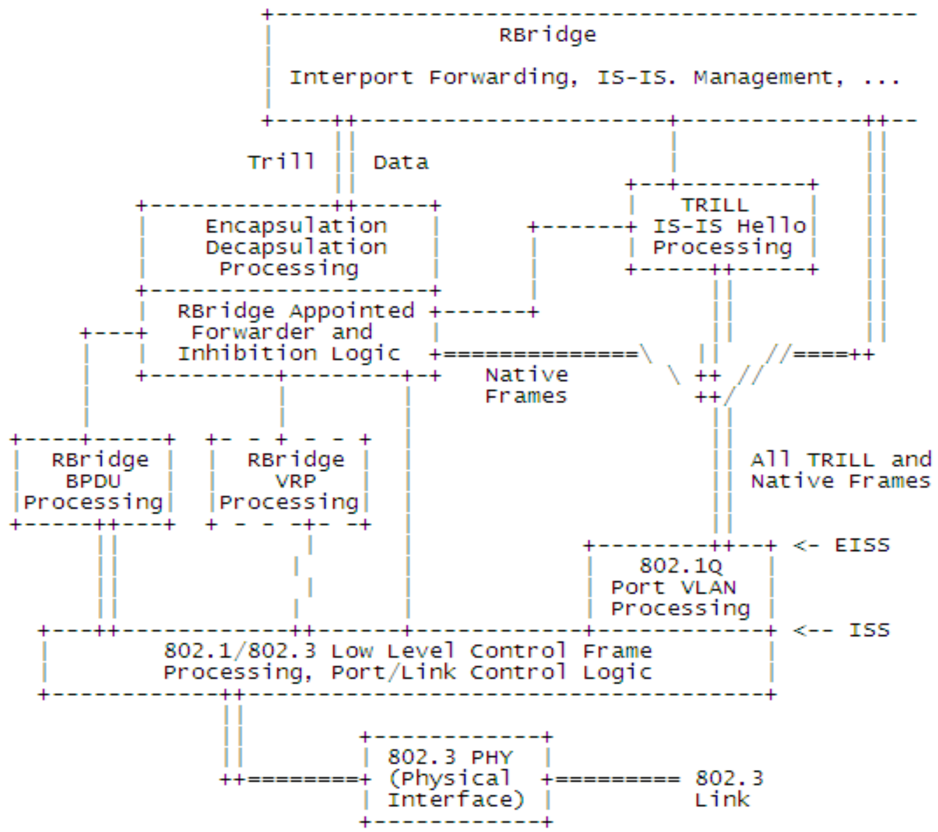
Pseudo Code and Complexity Analysis

This pseudo code is for a bridge attachment to a tree branch in the 'meshed tree' algorithm.

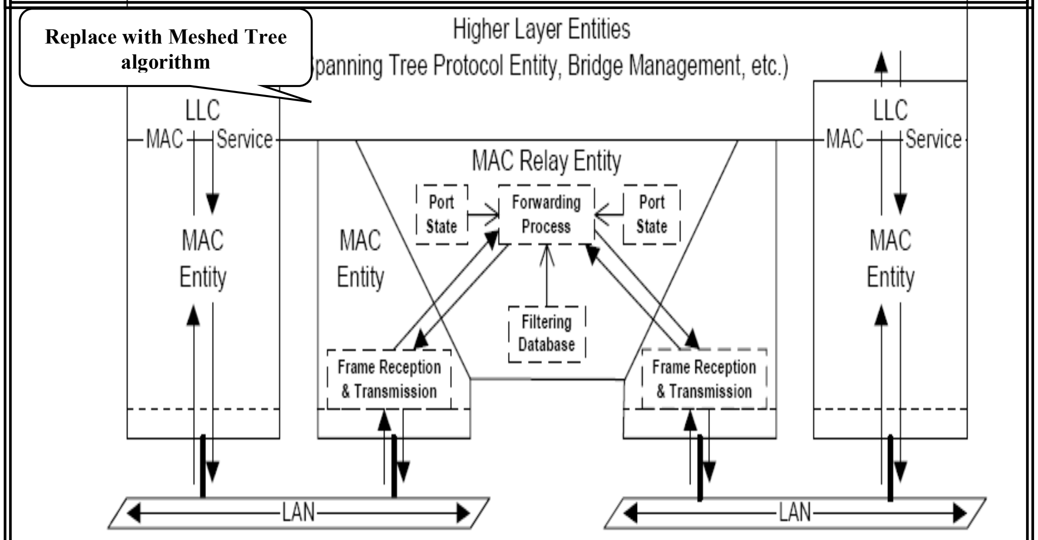
```
Repeat {
If ((hear a 'hello' message)            # a regular 'hello' from my neighbor
                                        # could be a new MT_VID offer
        - Scan the MT_VIDs
        - Compare with my existing MT_VIDs
        - If (new MT_VIDs)
                Repeat for all new MT_VIDs
                { Decision
                Criteria 1: Will the cost be better if I join this MT_VID
                Criteria 2:  Will the hops be within the limit of 'maximum hops'
                Criteria 3:             #any number of other decisions

                Send in a join request for the new MT_VID}
                Else (update the keep_alive timer of my MT_VIDs)
                }
```

As can be seen convergence or decision making iteration is of O(1) on every new MT_VID that is heard.

Appendix B Implementation requirements of ‘TRILL on RBridges’ and ‘Meshed tree on bridges’


RBridge
Interport Forwarding, IS-IS. Management, ...
Trill
Data
Encapsulation Decapsulation Processing
TRILL IS-IS Hello Processing
RBridge Appointed Forwarder and Inhibition Logic
Native Frames
RBridge BPDU Processing
RBridge VRP Processing
All TRILL and Native Frames
<- EISS
802.1Q Port VLAN Processing
<-- ISS
802.1/802.3 Low Level Control Frame Processing, Port/Link Control Logic
802.3 PHY (Physical Interface)
802.3 Link

From http://www.ietf.org/internet-drafts/draft-ietf-trill-rbridge-protocol-11.txt


Replace with Meshed Tree algorithm

From IEEE 802.1D “Replaced STA with MTA”