Multiple 802.1Q Spanning Trees

802.1 HILI Working Group
IEEE LMSC Meeting, July 1998

Norman Finn
Michael Smith
Cisco Systems, Inc.

Outline

- Why use one spanning tree or more than one?
- How many spanning trees?
- How to encapsulate BPDUs?
- How to identify different bridges in same box?
- How to assign VLANs to spanning trees?
- How to interoperate with 1-spanning-tree .1Q bridges?
- What about GMRP and GVRP?
- What questions remain to be answered?

Two Models

- VLANs as filters: Single Spanning Tree
  - One Bridged LAN for each router interface
  - Each Bridged LAN subdivided into VLANs to effect filtering and separation of traffic.
  - One filtering database (per bridge) shared by all VLANs.
  - One spanning tree for the one Bridged LAN.
- VLANs as subnets: Multiple Spanning Tree
  - One VLAN for each router interface.
  - Each VLAN is an independent Bridged LAN utilizing a common physical plant.
  - One filtering database (per bridge) per VLAN.
  - One spanning tree per VLAN.

Number of Spanning Trees

- More than one spanning tree in a single Filtering Database is not practical.
  - E.g., you learn address X on the red VLAN on port 2. You then want to forward a packet to X on the green VLAN, but green spanning tree is blocked on port 2.
- One spanning tree per FDB is simple to specify and is known to work.
  - This is equivalent to multiple separate bridged LANs sharing a single physical plant, but kept logically separate.
- Fewer spanning trees than FDBs are possible.
  - This would require a means for specifying the spanning tree associations.
  - Is this really necessary?

BPDU Encapsulation

- Just add 802.1Q tags.
- Let us call a “Spanning Tree Group” (STG) a set of VLANs sharing a spanning tree.
- Send one BPDU per physical port per STG. Can use any VLAN in the STG.
- Incoming BPDUs assigned to the STG to which VLAN belongs.
- Same rules for assigning untagged BPDUs to a VLAN as for normal data packets.

Different Bridges in Same Box

- With multiple instances of spanning tree in one box, what do we use for the 802.1D bridge ID for each instance?
  - Could use same ID for all.
  - Using a different bridge ID for each instance gives the opportunity to notice mis-wiring or mis-configuration when same bridge ID shows up as the root bridge ID for two different STGs.
  - Could use a different MAC address for each instance. This can use up lots of MAC addresses, especially if there are more bridges than physical ports in a backbone box.
  - Could use low-order bits of priority field in bridge ID.
- No matter what choice is made by a single box, spanning tree protocol still works.

Assigning VLANs to Spanning Trees

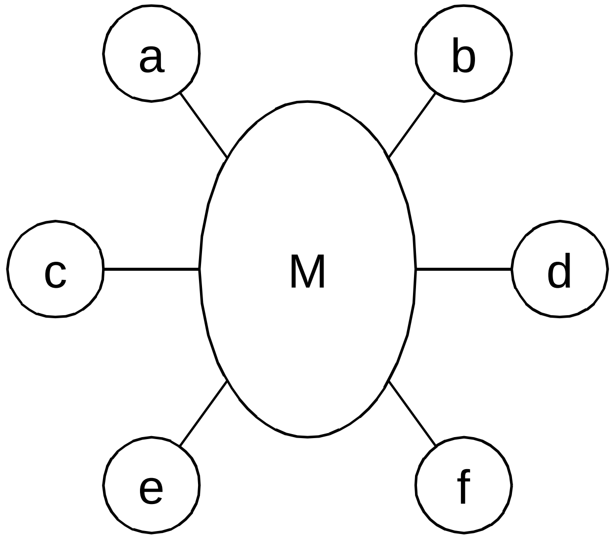
- We have a mechanism for assigning VLANs to FDBs.
- Could use the same mechanism for a parallel assignment of VLANs to STGs.
- Could simply add a single flag which differentiates between “One STG per FDB” and “One STG per box”.
- In either case, the assignment of VLANs to STGs <u>must not</u> be ambiguous, or loops will immediately result.
- When VLAN/STG assignment is changing, must either disable those VLANs, or have a protocol to distribute the assignments and prevent temporary loops.

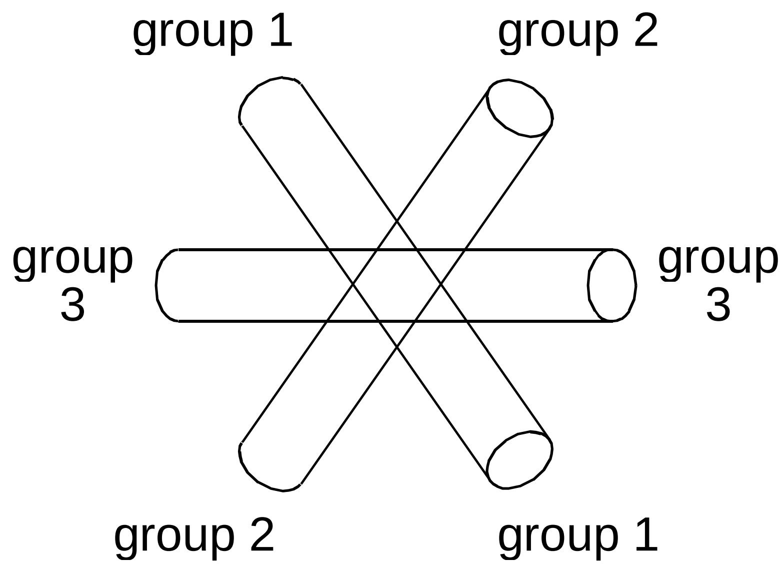
Interoperation Scenarios

- Arbitrary mixing of switches employing the different modes is not practical.
  - Single spanning tree switches collapse the multiple spanning trees into one large spanning tree with too many hops.
  - Switches using multiple filtering databases causes unnecessary flooding of traffic flows designed for a single database.
  - Routers may have features and capabilities which expect one or the other model for the underlying VLANs.
- One would expect many installations to employ one model or the other, but not both.

Practical Interoperation

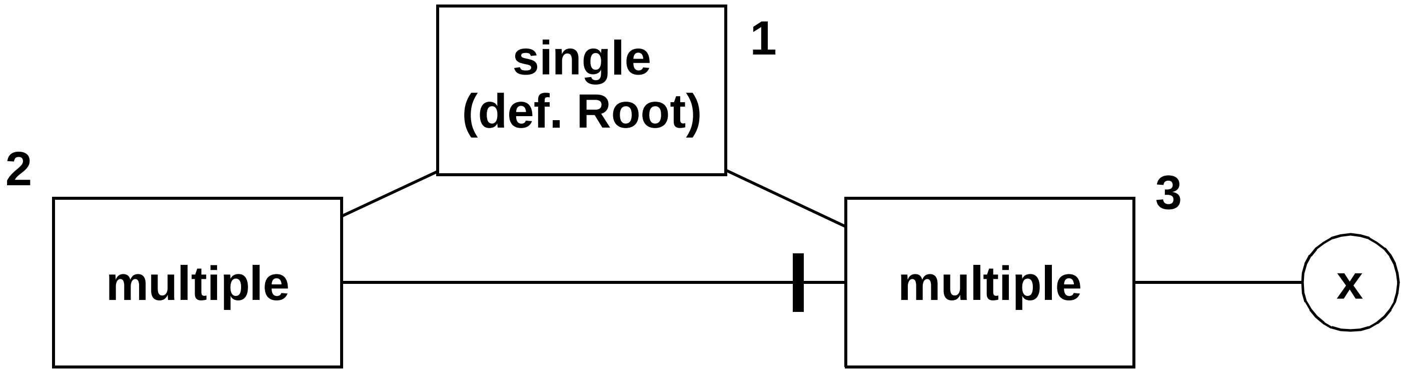
- One possible logical model for genuine interoperation:





- Central multi-spanning tree, multi-database cloud provides connectivity between single-spanning tree, single database islands.

Nightmare Connection



- Bridge 1 sees "red" tagged BPDUs as invalid, and drops them. "Red" VLAN is not blocked anywhere. Loop!!!
  - — Disallow and ignore this fundamentally dangerous configuration?
  - — Discover that bridge 1 is a single-spanning-tree bridge?
  - — Use a different MAC address for tagged BPDUs?
- Still have temporary loops when bridge 1 unblocks a port.
  - — Various games with priorities and timers can largely prevent this.

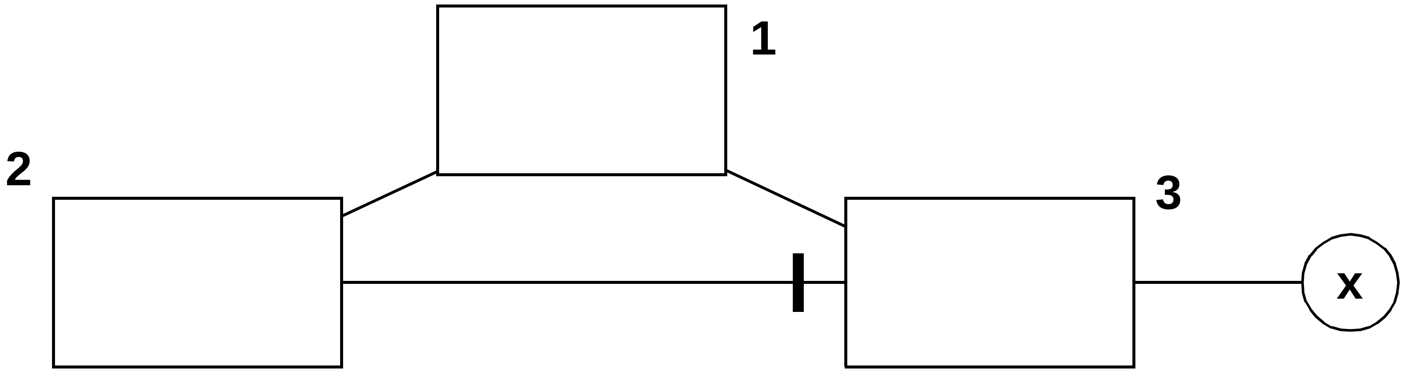
GMRP

- Current 802.1Q definition allows a GMRP participant to exist on a per VLAN basis
- Operates with the GIP context being the VLAN context
  - In a sense, creates an instance of GMRP for each VLAN context
- In a multiple ST environment, the GIP propagation would occur using the ST state of the ST for that VLAN
- Quite simply, very little changes

GVRP

- In the current 802.1Q model, a single GVRP instance runs on the switch
- Currently, GIP context is Base Spanning Tree Context; the only spanning tree
- In a multiple spanning tree environment, which spanning tree do we choose ?
- Allow a single spanning tree to go everywhere on the bridged LAN untagged will be interoperable for ST with current .1D/Q bridges but not GVRP
  - Interoperability with ST assuming other issues are resolved

GVRP Propagation Problem



- Assuming Base ST blocks at 3 but X is on Red
- Assume Red’s ST blocks at 1
- Current GVRP propagation model calls for Joins from X to be sent to 1 and not to 2
- Result is that data will be dropped at 1 and connectivity problems occur

Multiple GIP Contexts

- Still use a single GVRP participant
- Allow it to participate it multiple GIP contexts
  - Actually, it should participate in all GIP contexts
- The propagation occurs based on the GIP context corresponding to the GIP context for the VID
- Propagation behaviour similar to GMRP except that the GIP context is chosen based on the attribute value and not statically set as 1 GIP context
  - In the reference implementation, could easily be added by using an array of is_connected flags and an array of next_in_connected_ring pointers. GMRP would simply only use an array size of 1.

Other Issues

- Untagged BPDUs being assigned to different VLANs and, thus, different STs.
  - This can lead to STs collapsing which, in the worst case, becomes today's model.
- Dynamic VLAN creation is slower with a single VLAN per ST (using traditional ST leads to 30 sec delay)
- Dynamic VLAN creation using GVRP impacts topology changes

Questions to be Answered

- Do we support having fewer filtering databases than spanning trees (e.g. VLANs split among nine FDBs per bridge, collected into three spanning trees, or just one spanning tree per FDB), or just one spanning tree per FDB?
- How far do we go (how many mechanisms do we provide) to prevent temporary ambiguities in spanning tree group assignments?
- How far do we go to detect, prevent, and/or localize damage from the nightmare connection?