802.1aq Link State Protocol – Part II

Ali Sajassi

IEEE Plenary Meeting

July 19, 2007

Agenda

- Congruency Requirements - Revisited
- Building Congruent Trees
- Comparison between VID versus MAC SPT

Last time we covered

- 802.1Q Congruency Requirements & B-MAC Learning in Control-plane
- B-MAC Learning using Link-state Protocol
- Loop Prevention & Mitigation Mechanisms
  - Intra-domain: RPFC & TTL
  - Inter-domain: TTL

Last time we covered – slide 15

- A single instance of link-state control protocol can be used for:
  - o neighbors & topology discovery
  - o building distribution trees (used for both unicast & mcast traffic)
    - ➢ One tree per BEB, if optimal forwarding is required
    - ➢ One tree per BCB (for selected number of BCBs), if sub-optimal forwarding is required
  - o Distribution of B-VIDs for tree identification (one B-VID per tree)
  - o Distribution of mcast groups (for B-space) - flooding scope is limited using mcast pruning and NOT VLAN pruning

Congruency Requirements

From my slide deck presented in May, the following conclusion was made when B-MAC learning to be performed in control plane.

1. Reverse and forward paths do NOT need to be congruent but
2. Unicast and Multicast paths do NEED to be congruent

Congruency Requirements – Cont.

- However, if reverse and forward paths are not congruent, then its impact to CFM and CM need to be evaluated since
  - It impacts the operation of Link Trace and Loopback in CFM
  - It may impacts the operation of CM

Congruency & CFM

- Currently CFM assumes congruency in forward and reverse paths in a bridged network so that without such congruency:

  - Loopback mechanism can only check the data path from the originator to the loopback point in forward direction

  - For checking the data path in the reverse direction, another loopback message would be needed from loopback point to the originator

  - The same goes with Link Trace such that it would only check the forward path from the originator to the trace point but not the reverse path

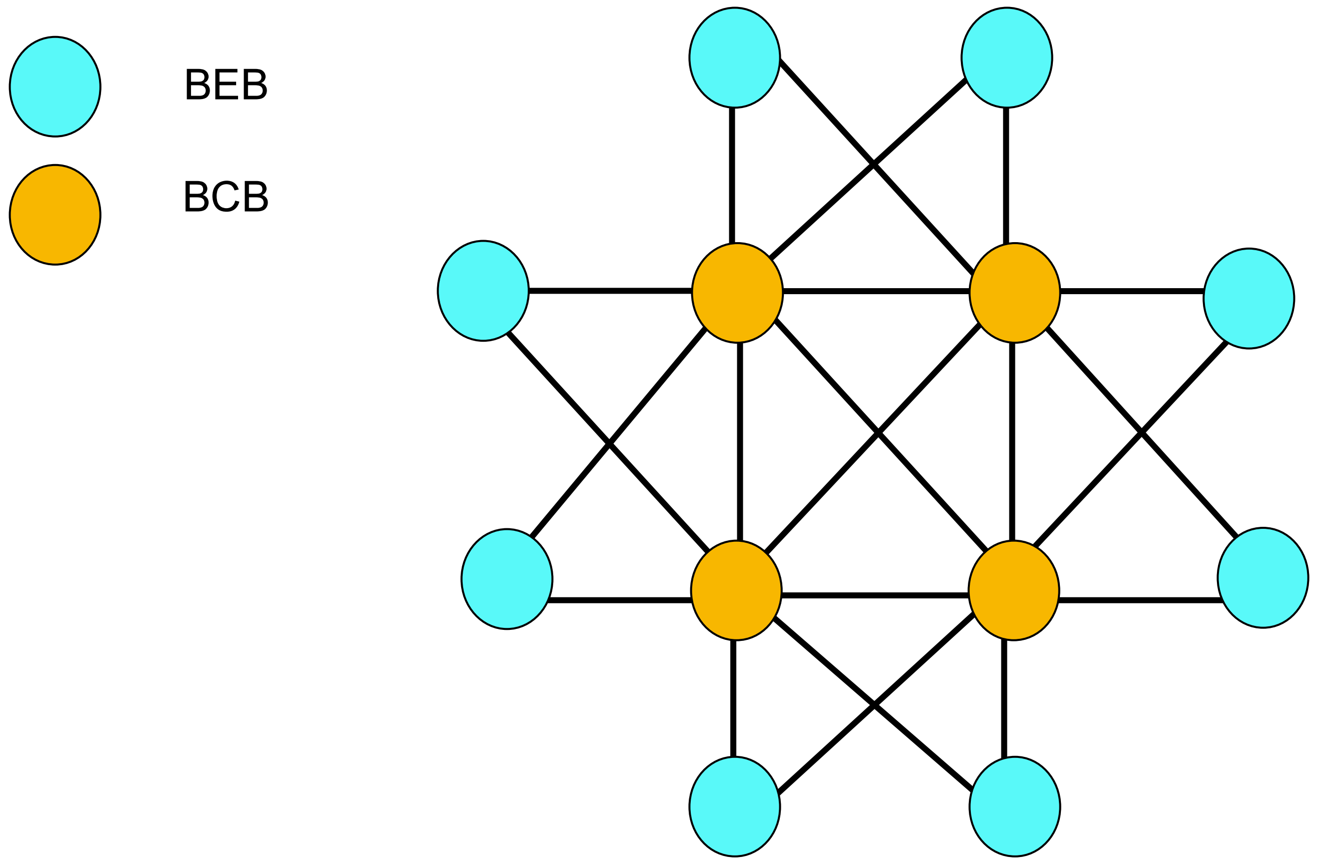
Conclusion

- If congruency would have taken a lot of effort in link-state protocol, then we could have consider the changes to CFM & CM to accommodate uni-directional trees/paths.
- However, congruency can be done relatively easily in link-state protocol, so we can keep the operation & coverage of CFM & CM same as before.

Agenda

- Congruency Requirements - Revisited
- Building Congruent Trees
- Comparison between VID versus MAC SPT

Congruency between Forward & Reverse Paths

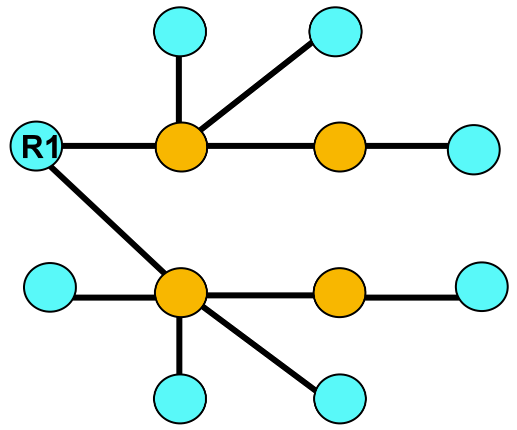


Each Bridge in PBBN runs link-state protocol (IS-IS) and thus has a full picture of the network topology

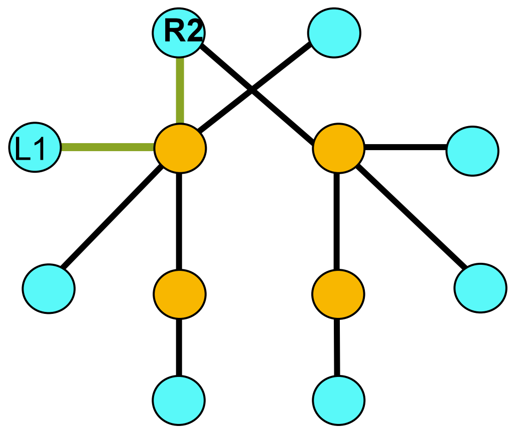
Example Only – Building Congruent Trees

- Each bridge builds its SPT (one per BEB) in an ordered way – e.g., starting with higher BEB IDs (or lower BEB IDs) which translates into higher B-VIDs (or lower B-VIDs) since B-VIDs are used as SPT identifier
- After building the 1st SPT tree for a BEB, then the 2nd SPT tree (next higher or lower B-VID) is build such that it uses the same branch between the two roots as in the 1st tree
- Next the third tree is built such that the first two branches are the same as the ones in 1st and 2nd tree
- And so on till all SPT trees are built on a given node
- Since all nodes follow the same algorithm, all the SPTs trees built in this way on each node are exactly the same as any other nodes – e.g., the same set of SPT trees are computed on every node.
- Since each node needs to compute a SPT tree for every BEB anyway, the cost of such congruency is minimal because the information already exist.

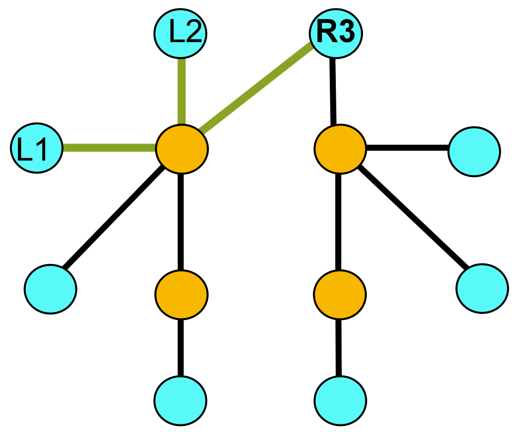
Example of Building Congruent SPT Trees



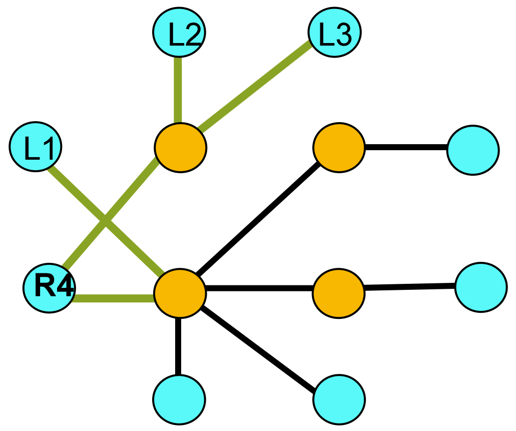
SPT-1



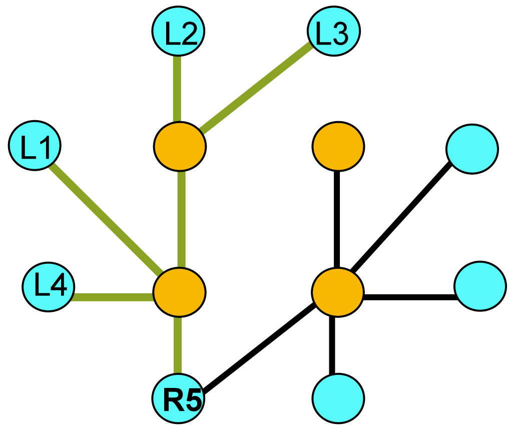
SPT-2



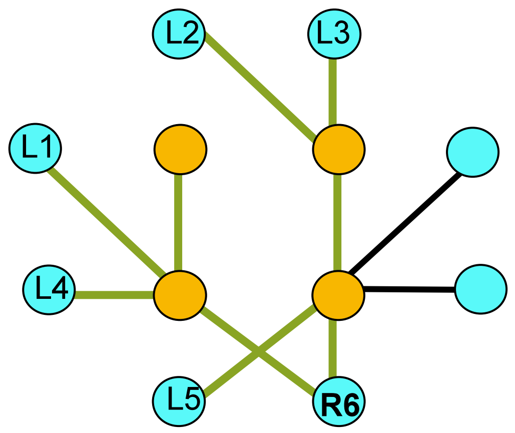
SPT-3



SPT- 4

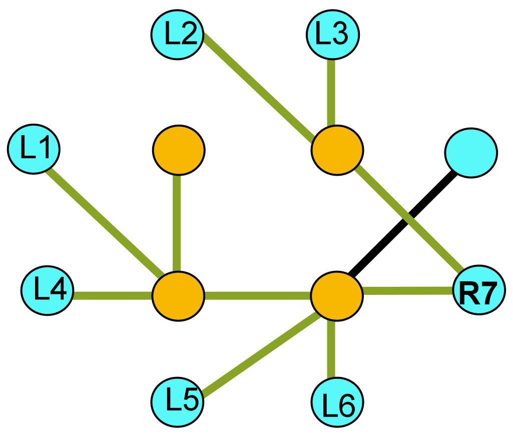


SPT-5

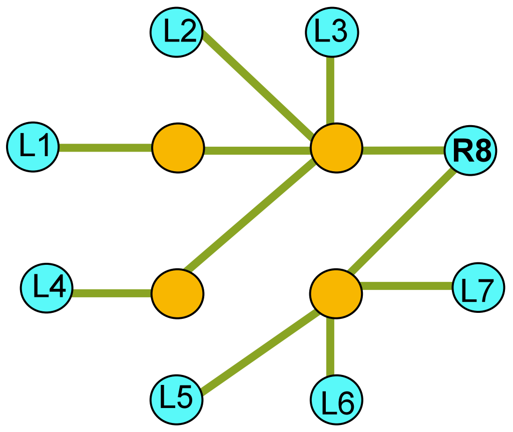


SPT-6

Example of Building Congruent SPT Trees - II



SPT-7



SPT-8

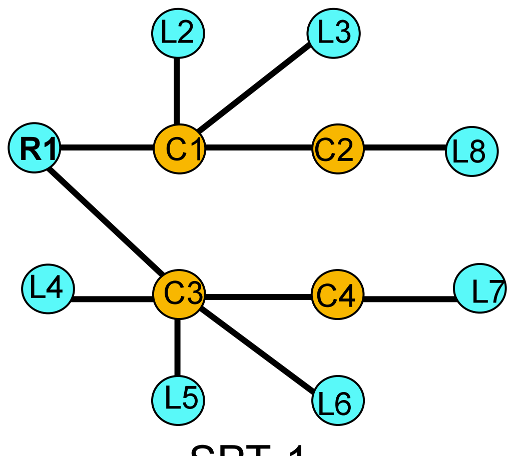
As it can be see in these figures, the path from root Rx to leave Ly on SPT-x is the same as the path from root Ry on SPT-y to leaf Lx

Modification to CFM Procedures

- Currently CCM, LBM, LBR, LTM, and LTR are all using the same VID for a given MA
- LBM and LTM should be modified to carry the B-VID associated with the reverse path so that when a MEP or MIP needs to send a LBR or LTR, it can simply use the associated B-VID
- This modification is needed anyway to support CFM for E-TREE !

Example of CFM Enhancements

1. R1 sends CC messages to the leaves
2. Lets say L3 doesn't receive it
3. L3 then sends a LBM to C1 on the B-VID corresponding to its SPT tree (B-VID-3) and <u>an indication that the response should be sent on B-VID-1 (corresponding to R1 whose CC messages were not received)</u>



SPT-1

Congruency & Congestion Management

- When using per-VID SPT, then Backward Congestion indication can be sent on the VID associated with node originating the Backward Congestion Indication.

Agenda

- Congruency Requirements - Revisited
- Building Congruent Trees
- Comparison between VID versus MAC SPT

Comparison

- VID-based SPT was described in my preso (May 2007) and some comparisons were made between VID-based and MAC-based SPT, here are some additional comparisons
- Comparison between the two approaches are made based on the following factors
  - - Tree Identification
  - - RPF Check
  - - Scalability

Tree Identification

- For unicast forwarding, SVL mode can be used and a single Filtering database can be used at a given node for RPF check and for unicast forwarding
- For multicast forwarding, if the same group address is used for different trees, then we need a way to identify the tree source – e.g., need to identify (S,G) and the associated forwarding table (SVL can not be used or else it can result in loop)
- Therefore, we need to have IVL mode for mcast which means we need an index identifier for FID (e.g., need to be able to identify the tree).

Tree Identification – Cont.

- In VID SPT approach, VID is used to identify the SPT tree as done currently with 802.1Q bridges
- In MAC SPT approach, source B-MAC address is used to identify the SPT tree which has the following implications in PBBN:

  - Need to have one tree per B-MAC address – which is not practical and doesn’t scale

  - Need to have source tree-ID embedded in B-MAC-SA which would then require partial lookup of MAC addresses in bridges (similar to routers !!) and would require partitioning of MAC space based on hierarchical assignment (box + line-card/port)

  - hierarchical assignment would require either a global scheme or would require flexible bit-masking to accommodate different assignments in different admin domains (which makes it very similar to IP subnet mask !!)

RPF Check

- VID SPT mechanism requires RPF check based on VLAN which is inline with current bridge operation
- To support uni-direction tree, we need some modifications to separate ingress filtering from egress filtering as described in my preso in May (2-bit vector rather than 1-bit vector)
- MAC SPT mechanism requires RPF check based on B-MAC-SA which requires two MAC lookups per frame – one on MAC-DA for forwarding and the other for MAC-SA for RPFC

Scalability

- VID SPT mechanism gives us max of 4K trees which typically translates into number of nodes in a single network but it can translates into fewer nodes if multiple trees are used per node (e.g., one way of doing ECMP)
- MAC SPT mechanism can potentially support more than 4K trees but it can also be limited to fewer than 4K based on hierarchical assignment in MAC address (e.g., allocation of fewer than 12 bits for node ID in MAC address)
- A single bridged network typically contains in order of tens or hundreds of nodes but not thousands !! With thousands of nodes in network, then IGP convergence becomes an issue specially when calculating thousands of trees per node simultaneously. Add to that number of mcast trees per node tree, then the convergence can really become an issue.