802.1aq: Link State Protocol & Loop Mitigation Options

Ali Sajassi

IEEE 802.1 - Geneva

May 30, 2007

Agenda

- 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

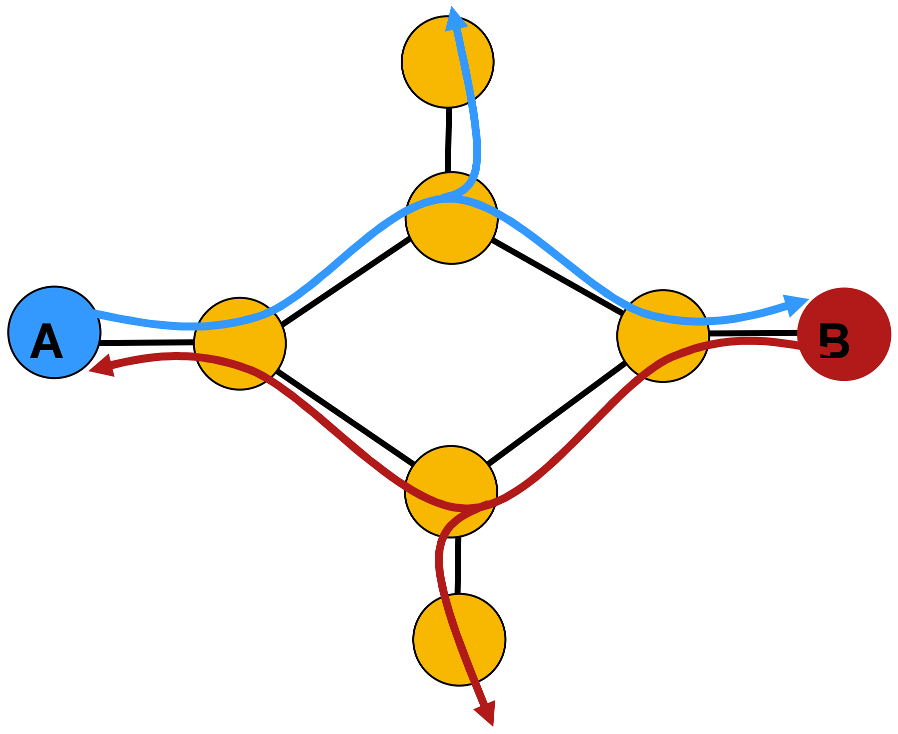
Agenda

- 802.1Q congruency requirements for forward & reverse paths
- 802.1Q congruency requirements for unicast & mcast
- B-MAC learning in control-plane & congruency requirements

Congruency in Forward and Reverse Paths - I

If the forward and reverse paths for a given VLAN are different, then it creates the following issue:

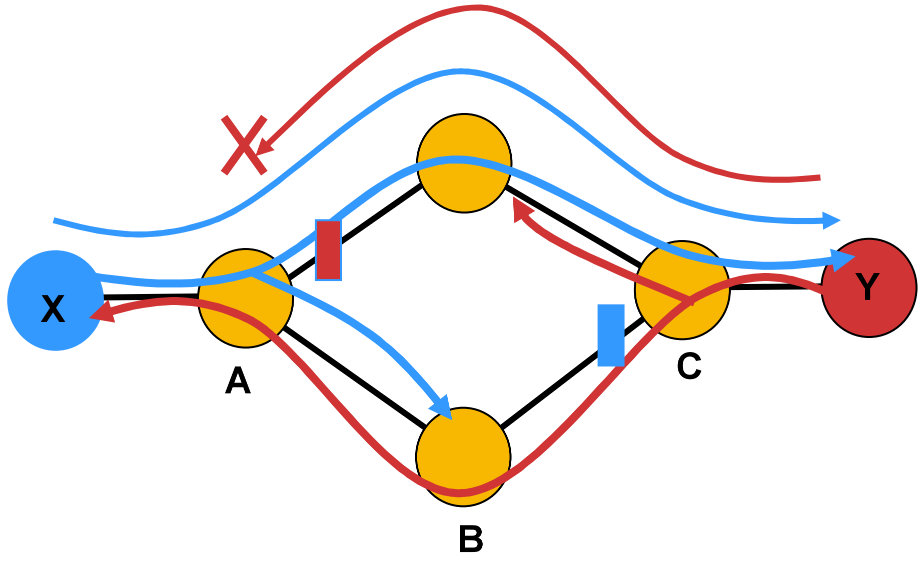
- Because of flooding mechanism of unknown unicast frames in 802.1Q, the reverse and forward paths must be the same – otherwise, there will be no learning of MAC addresses toward destination and the frames gets flooded forever.



Congruency in Forward and Reverse Paths - II

If the forward and reverse paths for a given VLAN are different, then it creates the following issues:

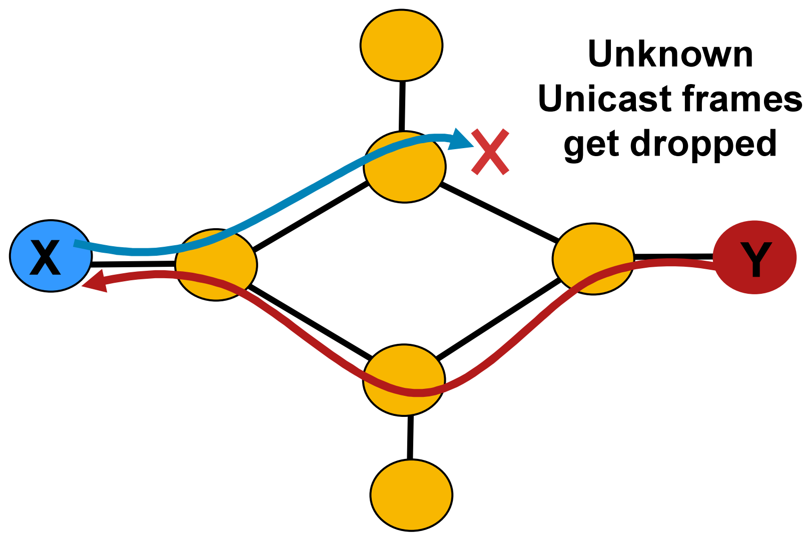
- Traffic in one direction is dropped because of the blocking by the spanning tree – e.g., X's blue STI is not congruent with Y's red STI and thus when X floods its unknown unicast to Y along X's blue STI and Y is responding along its learned path, it gets dropped by the Y's red STI at node A.

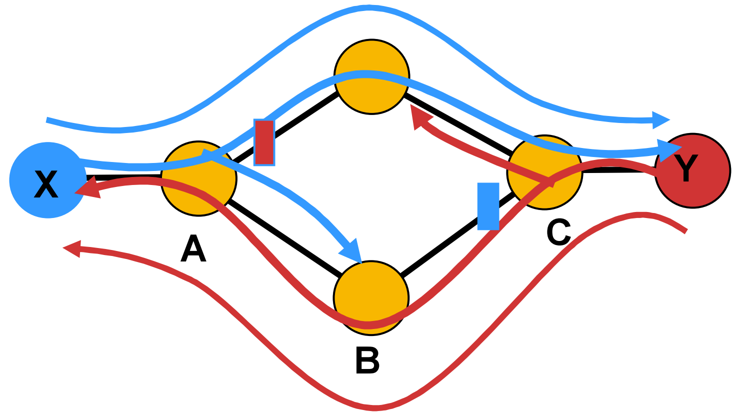


Congruency in Forward and Reverse Paths- III

Performing MAC address learning in control plane removes the requirement for congruency of forward and reverse paths because:

1. When learning is performed in control plane there is no flooding of unknown unicast - e.g., if a node receives an unknown unicast, it drops it.
2. When learning is performed in control plane, frames from X to Y can take a different path than frames from Y to X because there is no flooding of unknown unicast.





Congruency in Forward and Reverse Paths- IV

Conclusion:

When using B-MAC learning in control-plane, there is no need to:

1. Make reverse and forward paths congruent and
2. Use mechanism such as Path Vector or Reflection Vector to ensure such congruency

Congruency in unicast &mcast paths – part I

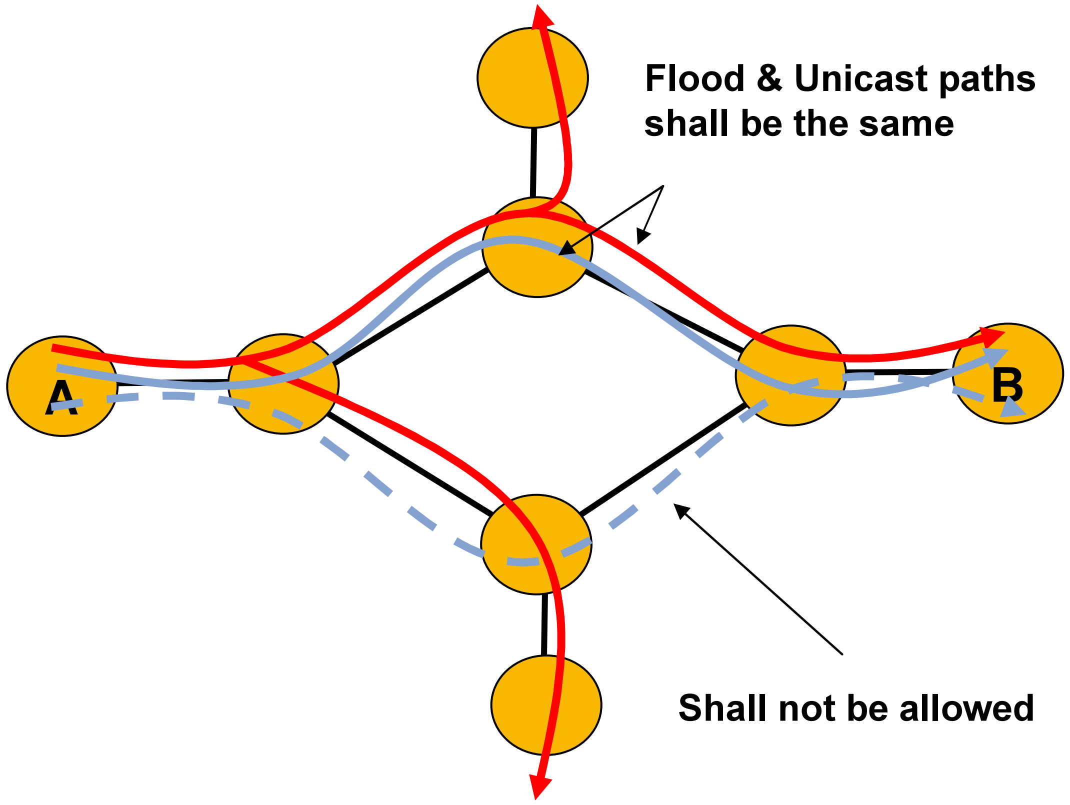
- Since unknown unicast frames are flooded along the broadcast path, the following issues can happen if known unicast and broadcast paths are different:

  a) Out-of-order delivery

  b) Issues with Ethernet OAM (802.1ag)

  c) Black holing in customer's network

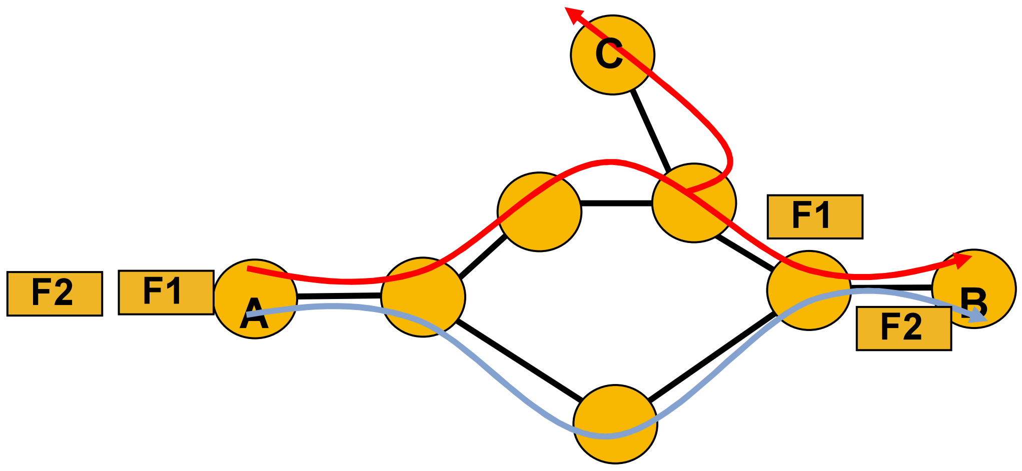
  d) Loop creation in customer's network



Congruency in unicast &mcast paths – part II Out of Order Delivery

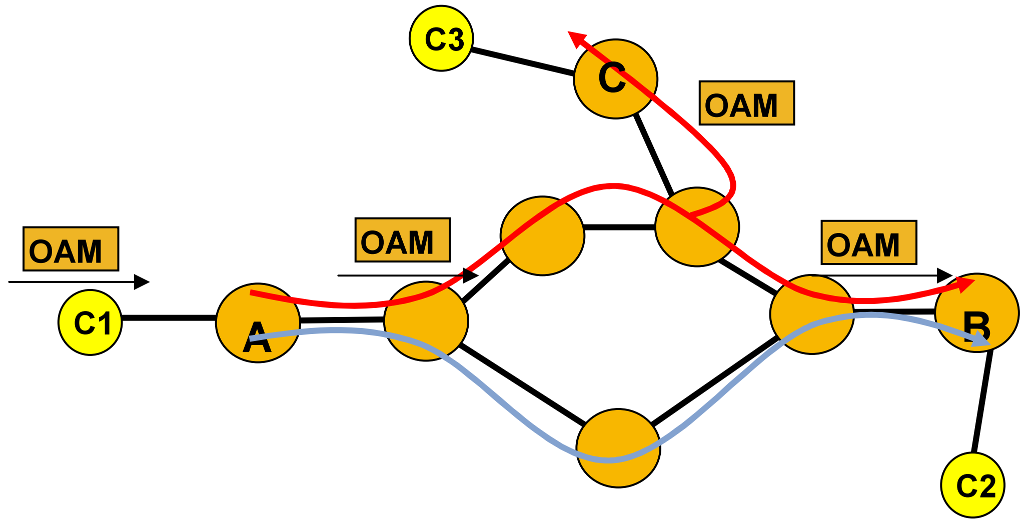
Consider the following scenario:

- **A wants to send frames F1 & F2 to B. When A sends F1, it hasn't learned the B's MAC address and thus it floods the frame over its flood path.**
- **During or right after A sending the frame F1, it learns the MAC address of B, so the next frame, F2, is sent via the unicast channel.**
- **If the delay along the multicast path is longer than the unicast path, then F2 can arrive before F1 at the B and thus causing packet re-ordering in the steady state.**



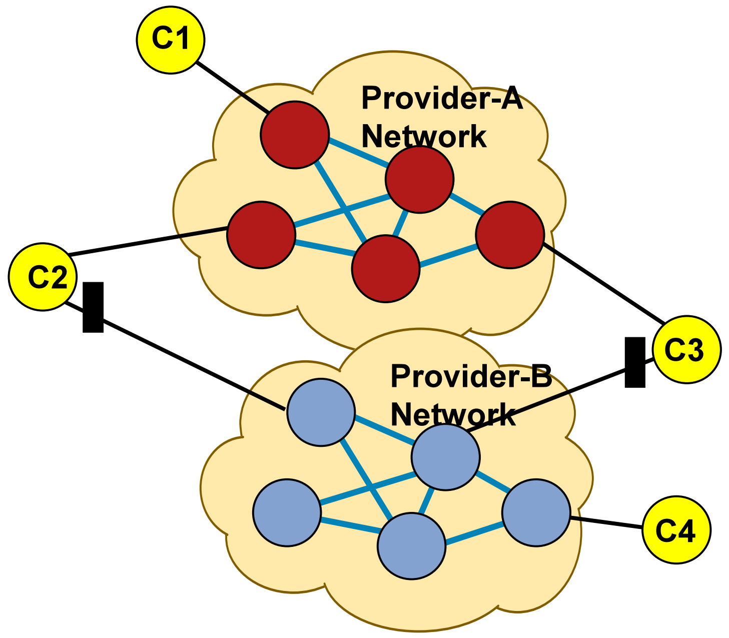
Congruency in unicast &mcast paths – part II Ethernet OAM Issues

- Since Eth OAM periodic Connectivity Check (CC) messages are of type multicast, they go over service provider's multicast path and thus if there is a break on the unicast path, it cannot be detected by CC mechanism. In regular bridge network, Ethernet OAM (802.1ag) provide coverage for both unicast and multicast traffic.



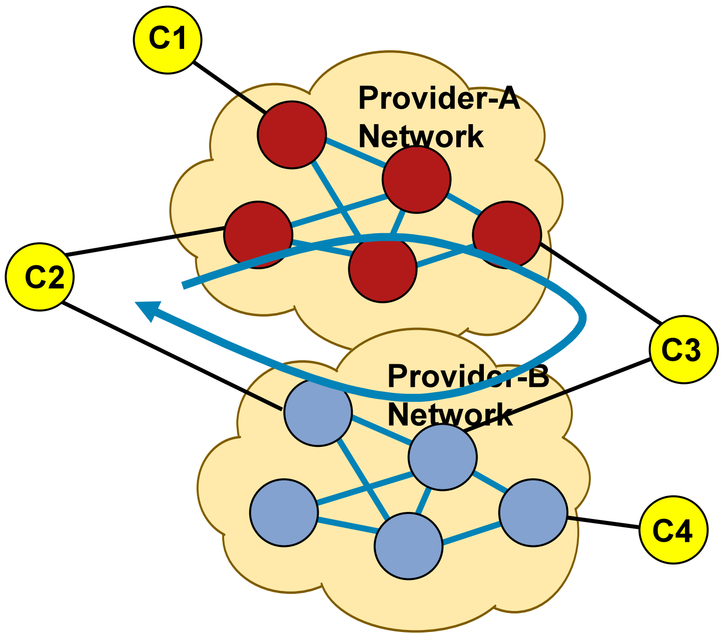
Congruency in unicast &mcast paths – part III Black hole

- If customer’s multicast & unicast paths within the provider’s network are different and if there is a failure in the provider’s ***unicast*** path, then the loss of unicast path is not detected by customer bridges (since customer’s BPDUs still gets exchanged) and this results in customer’s traffic to get black holed.



Congruency in unicast &mcast paths – part IV Loop Creation

- If customer’s multicast & unicast paths within the provider’s network are different and if there is a failure in the provider’s ***multicast*** path such that it results in loss of customer’s BPDUs, then customer’s bridges unblock their alternate ports which results in a loop within the customer’s network because the unicast path is still up and running.



Congruency in unicast & mcast paths - VI

Conclusion:

- B-MAC learning in control-plane does not alleviate congruency requirement for unicast & mcast paths in B-space because if they are not congruent, it results in the above issues for the customer.

A Single Control Protocol

A Single Control Protocol

- 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

Self-Built Auto-Discovery

- There is no need to distribute I-SID information via this protocol. There is no need for auto-discovery of BEBs participating in a given service instance because:
  - o I-SIDs are mapped into B-mcast groups and unknown C-MAC unicast gets flooded over this B-mcast tunnel
  - o Only I-SID to B-mcast group mapping is needed. This is a local matter configured on each BEB
  - o B-mcast group info is distributed using IS-IS control protocol

Agenda

- Neighbor & Topology Discovery
- Shortest Path Tree - SPT
- Multiple Spanning Trees - MST
- Multicast Distribution Trees – MDT

Neighbor & Topology Discovery

- IS-IS is used as a link-state protocol for distribution of:
  - o B-MAC addresses as identifiers for BCBs and BEBs among all the SPT bridges along with neighbor link cost
    - ➢ multiple B-MAC addresses can be used to identify the same bridge – e.g., one B-MAC per line card
- It can also be used for distribution of:
  - o B-VIDs associated with a given SPT bridge – each B-VID identifies a tree rooted at that SPT bridge (multiple trees are needed for ECMPs)
  - o Mcast group addresses (in B-space) for limiting the scope of broadcast on trees identified by B-VIDs
  - o Indication of whether a tree is bi-directional or uni-directional (optimal trees are unidirectional but sub-optimal trees are bidirectional)

Neighbor and Topology Discovery

- Per clause 27.2 of 802.1aq, the following configs are required:
  - Each bridge has a globally unique MAC address
  - Each bridge port has a 12-bit ID, unique within the bridge
  - Each bridge has a relative priority within the network
  - Each bridge port has a relative priority within the bridge
  - Each Port has a Path Cost
- The link state protocol can use the above info in building its link-state database and subsequently building its SPT and MST trees

SPT Configuration & SPVID Assignments

- In an SPT region, all bridges must agree on same allocation of VIDs to specific SPT & MST trees. Currently clause 27 of 802.1aq describe a mechanism for
  - SPT configuration, configuration identification
  - SPT region determination and identification
  - SPVID allocation
- When 802.1ah is used with B-MAC learning in control plane, then I-SID (rather than VID) is used for community identification. For limiting the scope of broadcast within the provider network, B-Mcast addresses can be used instead of B-VID as it scales lot better and provides many broadcast domains. Therefore, B-VID can be used just for tree identification (e.g., 1:1 mapping between B-VID and Tree ID)

SPVID Assignments Options

- 802.1aq describes a centralized mechanism for SPVID assignment using the master bridge
- In 802.1ah network where SPVIDs can only be used as tree IDs as described above, we can support upto 4K such trees – e.g., 4K bridges in a single region (or area)!!
- Since there can be a one-to-one association between SPVIDs and bridges, one can assign SPVIDs along with a bridge MAC address at the time of configuration
- IS-IS protocol can be used as an alternative mechanism to distribute this info among different bridges allowing each bridge to have visibility of different SPTs rooted at different bridges as identified by their SPVIDs

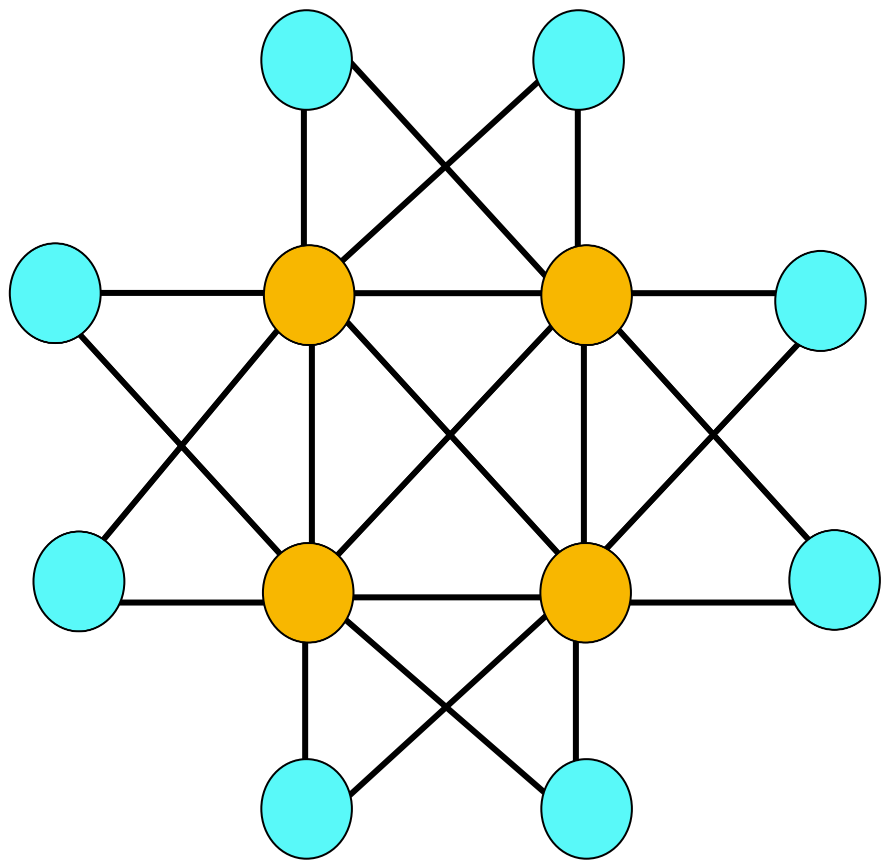
Agenda

- Neighbor & Topology Discovery
- Shortest Path Tree - SPT
- Multiple Spanning Trees – MST
- Shortest Path Tree w/ ECMP
- Multicast Distribution Trees – MDT

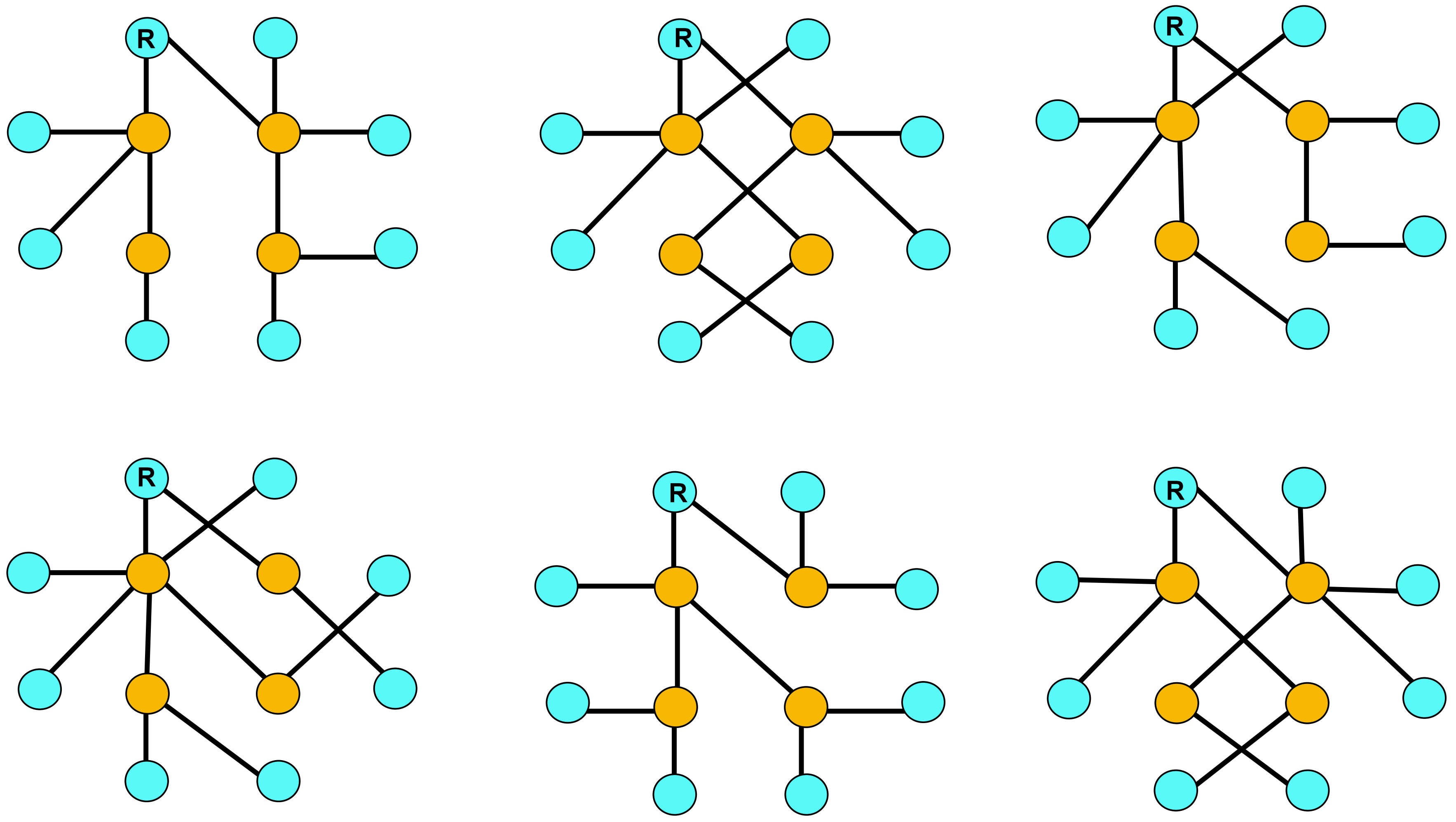
Optimal Distribution Tree - SPT

🔵 BEB

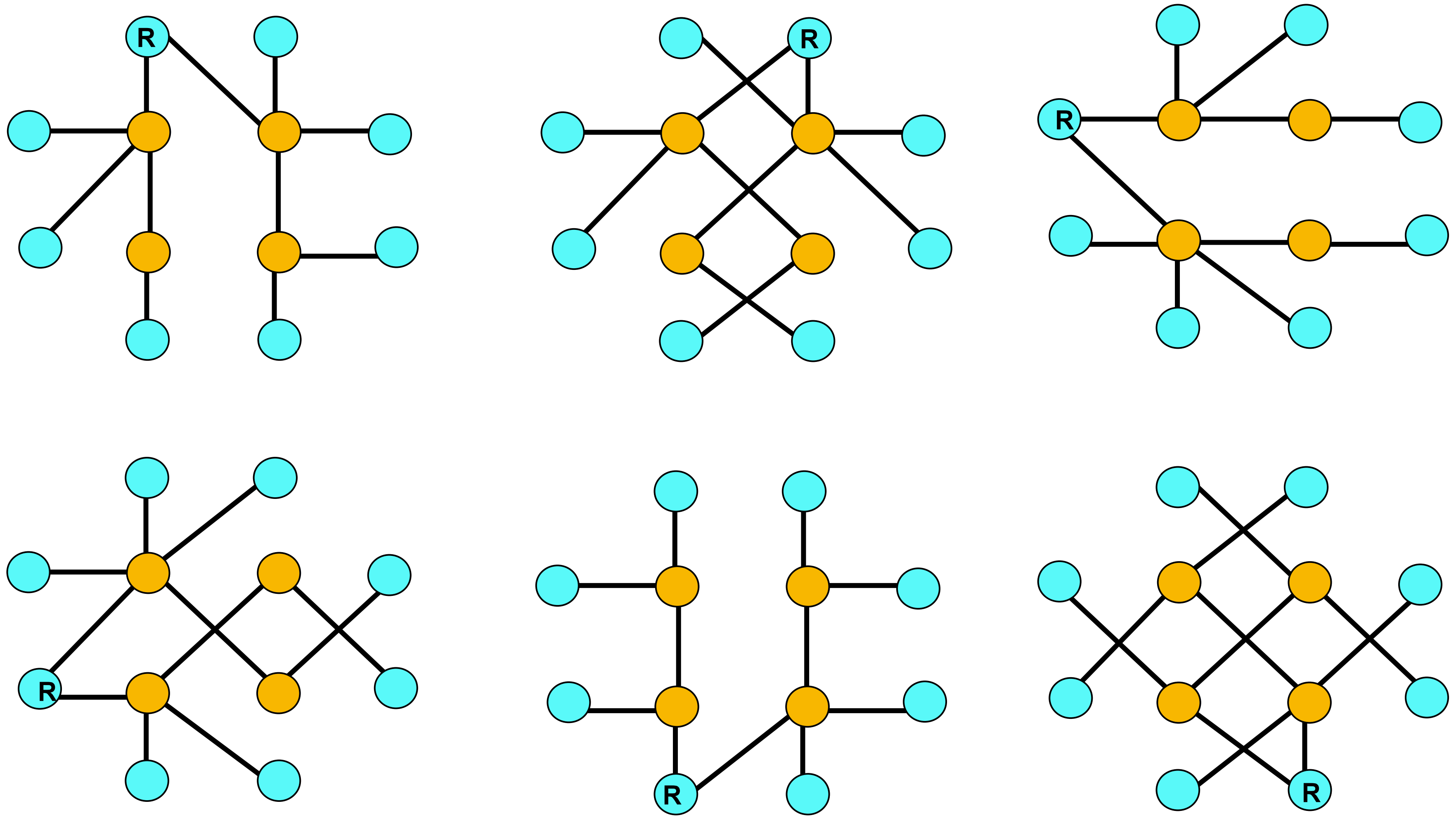
🟠 BCB



SPTs for different ECMP rooted at a BEB



Different SPTs rooted at different BEBs taking different paths – Alleviating the need for ECMP SPTs



One SPVID per SPT

- In a typical configuration where edge switches are dual-homed to core switches and core switches are fully or partially meshed among themselves, then having a single tree per edge switch should be sufficient. Since as number of edge switches becomes large, trees becomes more evenly distributed over the network graph.
- ECMP load balancing among different links is basically achieved through different trees.
  - o However, ECMP achieved this way can be sub-optimal
  - o Building SPTs with minimal overlap required further study

How to Build SPT using Link State

a) Need at least one tree per BEB (in case of ECMP multiple trees may be used per BEB)

b) The trees are uni-directional with the root at the BEB

c) A given tree is used for both unicast and mcast traffic originating from the BEB who is the root of the tree

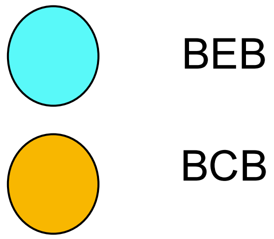
d) Each SPT bridge in the region will calculate the distribution trees (identified by their SPVIDs) based on LSP information

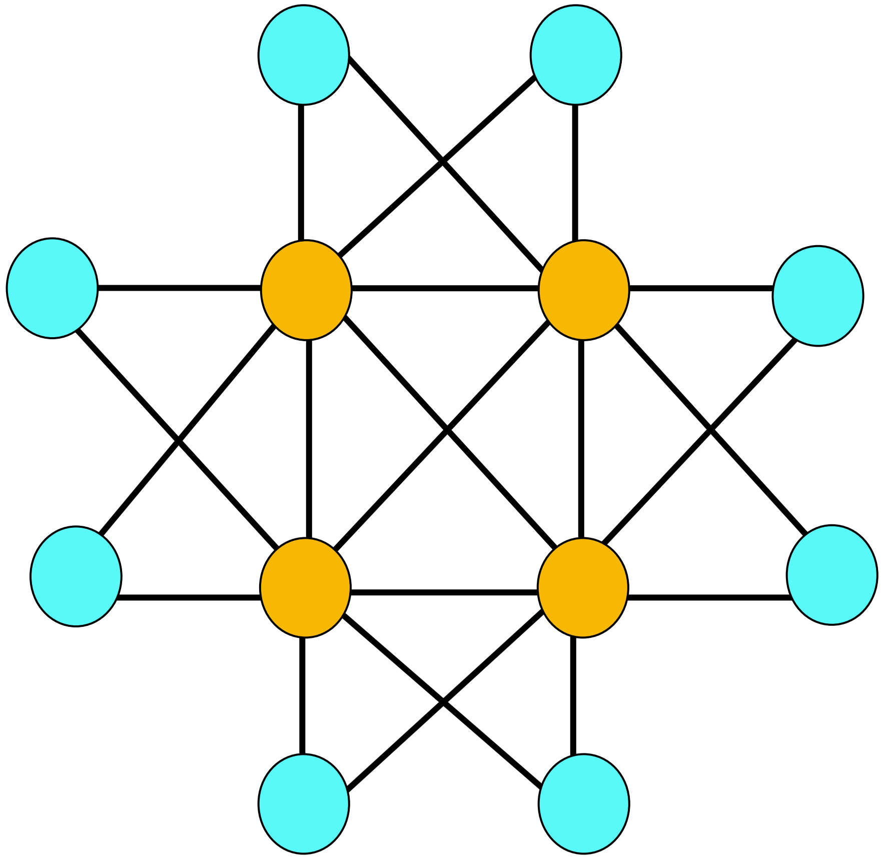
e) Each SPT bridge performs Reverse Path Forwarding Check (RPFC) on that tree (aka B-VID) in order to mitigate transient loops

Agenda

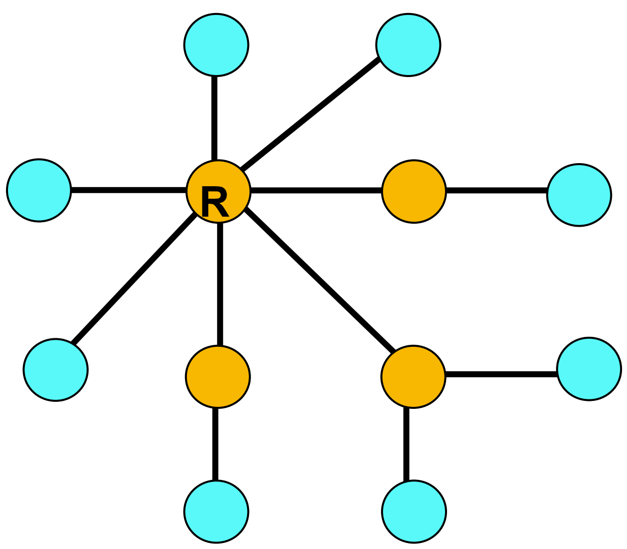
- Neighbor & Topology Discovery
- Shortest Path Tree - SPT
- Multiple Spanning Trees – MST
- Shortest Path Tree w/ ECMP
- Multicast Distribution Trees – MDT

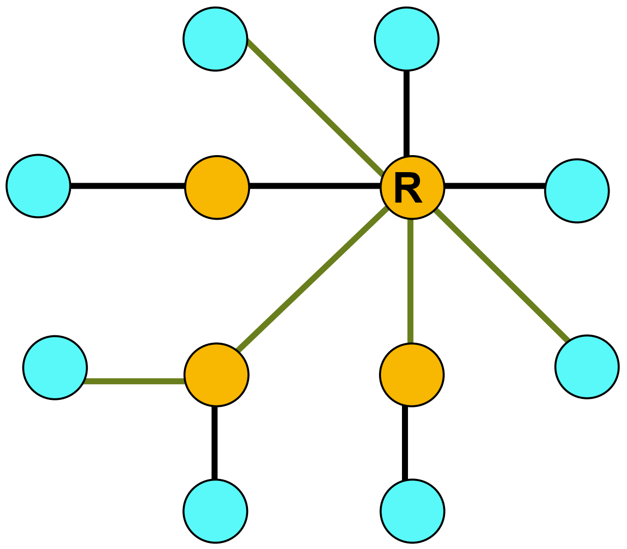
Sub-optimal Distribution Tree





MST Tree





As it can be seen in the figure, different MSTs rooted at different BCB can utilize different links in the network, therefore mitigating the need for ECMP trees when there are more of these trees – e.g., four trees in this example

How to Build it ?

a) Each BCB is configured to indicate whether it should be a root of a sub-optimal tree and if it is a root, then how many ECMP it should supports

b) The trees are bi-directional with the root at the BCBs

c) Each bridge in the region will calculate the distribution trees rooted at a given BCB (one per B-VID) based on LSP information

d) Each bridge performs Reverse Path Forwarding Check (RPFC) on that tree (aka B-VID) in order to mitigate transient loops

Failure: Link, Node, and Root

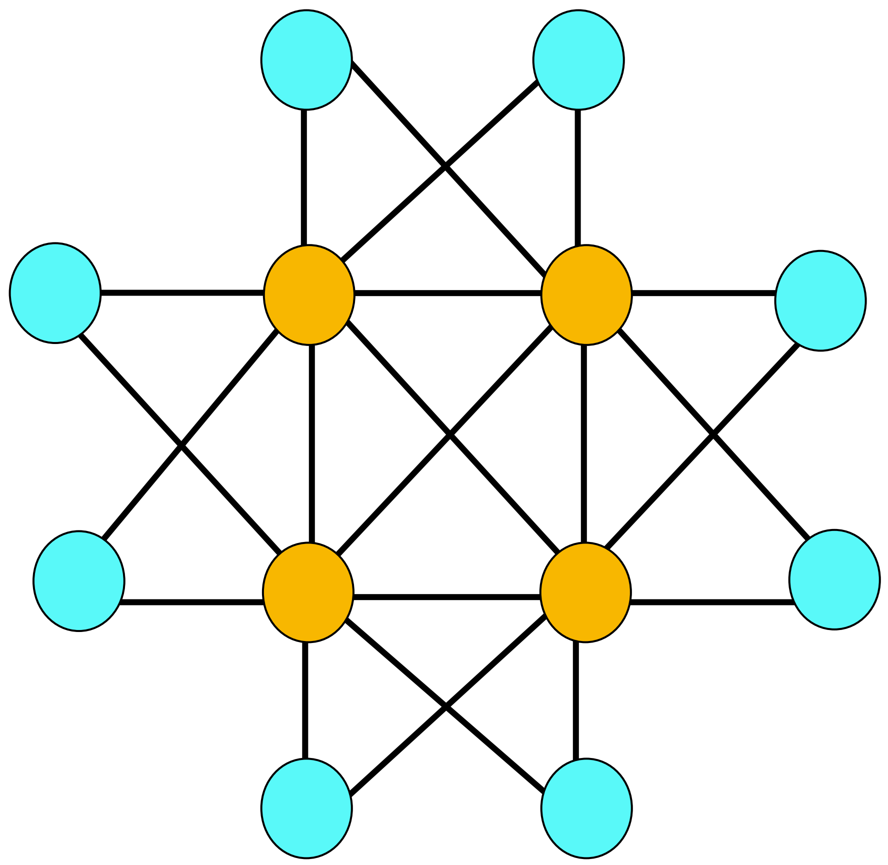
- Link Failure:
- Node Failure
- Root Failure

    If an MST root fails, then the tree has failed, and BEBs need to join a different tree rooted a different BCB for their corresponding MST mcast groups

Agenda

- Neighbor & Topology Discovery
- Shortest Path Tree - SPT
- Multiple Spanning Trees - MST
- Shortest Path Tree w/ ECMP
- Multicast Distribution Trees – MDT

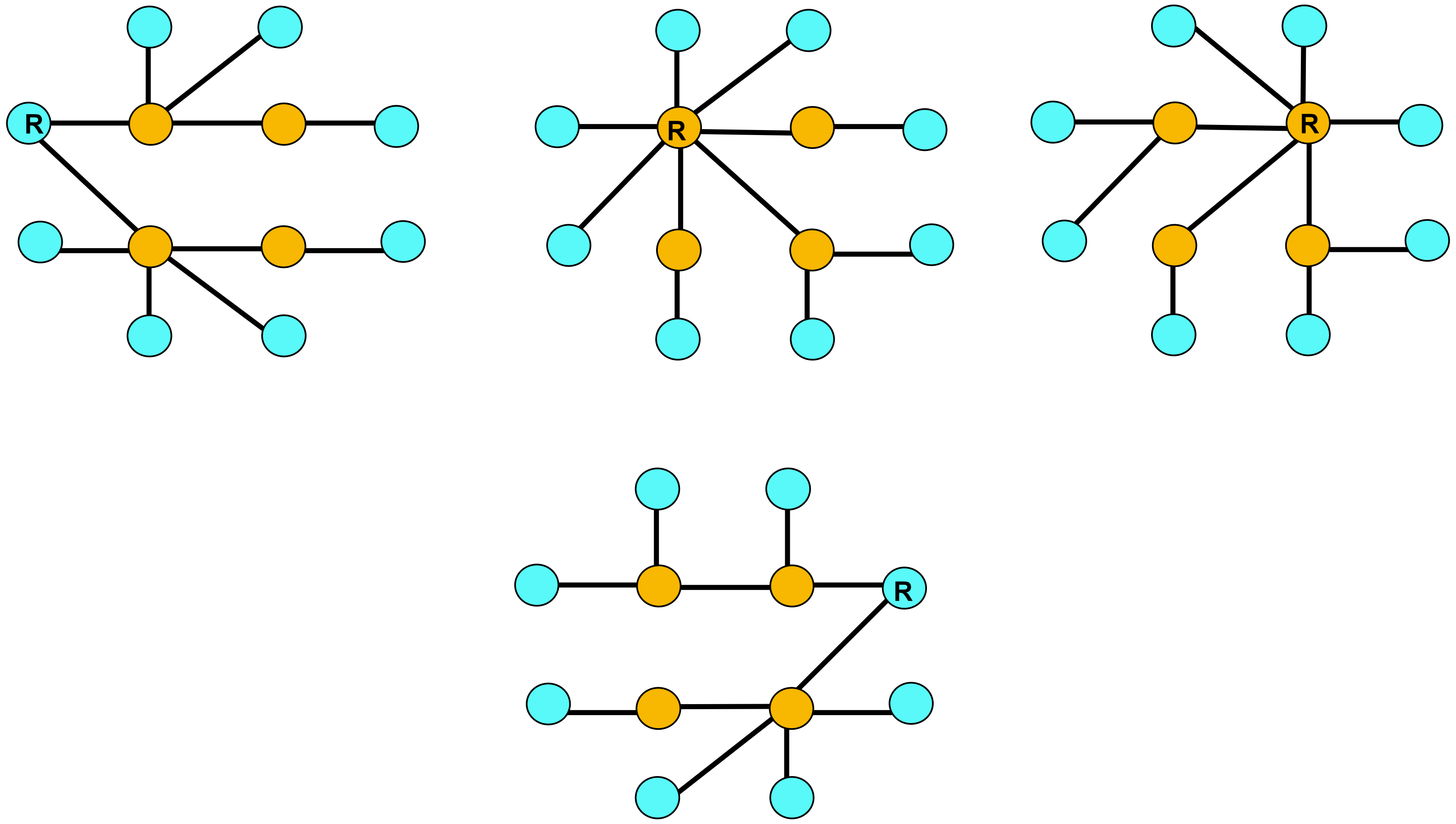
SPT for ECMPs - Optional



● BEB

● BCB

Different SPTs – Each rooted at a BEB or BCB along the path



SPT Tree for ECMP

- There is a SPT tree rooted at each node
- If there are ECMPs, then it gets reflected in the link state database for that tree (refer to Appendix A: Link State Protocol)
- This is the typical way where Link State protocol operates

How to Build SPT using Link State

a) Each node builds a tree rooted at itself

b) This tree is only used for unicast data

c) A single SPVID can be used for all such trees so that it indicates to the receiving bridge that the receiving frame must be forwarded based the receiving node SPT (e.g., SPT rooted at the receiving node).

d) Each SPT bridge in the region only calculates its own SPT (a single SPT per SPT bridge).

ECMP SPT Tree

- This tree has very little overhead since each node requires to build one such tree
- This SPT is an optional SPT
- This SPT can be used in applications where congruency requirements can be relaxed.

Agenda

- Neighbor & Topology Discovery
- Shortest Path Tree - SPT
- Multiple Spanning Trees – MST
- Shortest Path Tree w/ ECMP
- Multicast Distribution Trees – MDT

Limiting Scope of Broadcast

- Each BEB sends a list of mcast groups that it is interested (encoded based on MMRP for efficiency)
- Mcast groups need to indicate if they are on MST or SPT trees
- For a mcast group on a MST, the pruning is done along the branches of that MST
- For a mcast group on a SPT, the pruning is done along the branches of that SPT
- If a mcast group wants to be on all SPTs, then it would indicates that accordingly
- All the bridges after computing the corresponding trees, they prune it based on these mcast information

Loop Mitigation

Agenda

- Reverse Path Forwarding Check
- Hop Count

802.1Q Ingress Filtering Capabilities

- 802.1Q clause 8.6.2 describes ingress filtering function as follow:
  - ➢ Each Port may support an Enable Ingress Filtering parameter. A frame received on a Port that is not in the member set (8.8.9) associated with the VID shall be discarded if this parameter is set. The default value for this parameter is reset, i.e., Disable Ingress Filtering, for all Ports.
- 802.1Q clause 8.8.7 describes active VLAN as: A VLAN is active if either of the following is true:

  1) The VLAN’s member set (8.8.9) contains one Port that is in a forwarding state, and at least one other Port of the Bridge is both in a forwarding state and has Ingress Filtering (8.6.2) disabled;

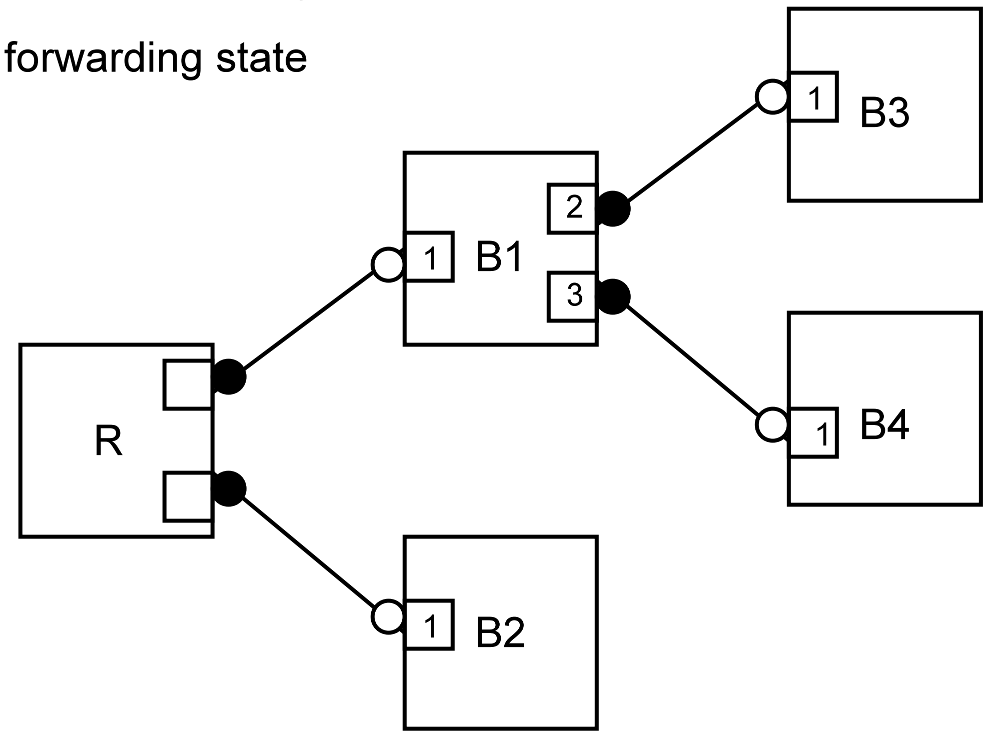
  2) The VLAN’s member set contains two or more Ports that are in a forwarding state.

Ingress Filtering & RPF Check

- Ingress Filtering feature can be used to perform PRFC on a VLAN basis – e.g., a given bridge along the path of SPT has one ingress port and one or more egress ports
- If ingress filtering is enabled on all ports except the ingress port for a given SPVID, then only frames with that SPVID can come through the ingress port. And if frames with that SPVIDs come through any other ports, they get discarded.
- This ingress filtering function provides an “RPF Check”

Ingress Filtering & RPF Check

●———	Designated port – forwarding state
○———	Root port – forwarding state



- Bridge B1 has ingress filtering disabled on its P1 but enabled on P2 & P3
- Bridge B2, B3, and B4 have ingress filtering disabled on their P1 ports

802.1Q Modifications

- Clause 8.6.2:

  “Each Port may support an Enable Ingress Filtering parameter. A frame received on a Port that is not in the member set (8.8.9) associated with the VID shall be discarded if this parameter is set. The default value for this parameter is reset, i.e., Disable Ingress Filtering, for all Ports.“

- Currently there is only one bit that indicates whether a port is in a VLAN member set or not. If this bit is set and ingress filtering is disabled, then frames can come and leave this port. If this bit is not set and ingress filtering is enabled, then frames can not come and leave this port.

802.1Q Modifications – Cont.

- What we need is two bits per port to indicate the allowed direction for the frames – one bit for ingress direction and another bit for egress direction.
    - If both bits are set and ingress filtering is enabled, then frames can come and leave this port.
    - If ingress bit is not set but egress bit is set, then frames can leave this port but can NOT come in through this port - which is what we need for a a leaf port of a uni-directional tree
    - If both ingress and egress bits are not set, then frames can NOT come and leave this port
    - If ingress bit is set but egress bit is not set, then frames can come through this port but not leave this port – which is what we need for a root port of a uni-directional root tree

RPFC: VLAN versus MAC-SA

- One of the main advantages of doing MAC learning in control plane is that it cuts the no. of lookups in half – e.g., lookup for MAC-SA is eliminated because there is no learning in data-plane, thus improving performance
- However, some argue that current bridges already perform two lookups per packet and thus nothing is lost. But that is not the point. The point is that RPFC based on MAC-SA is new functionality which doesn't exist and the question is what is the best & most efficient way to implement RPFC when B-MAC learning is performed in control plane.
- Many of the early-adopter bridges use network processor to implement the functionality and thus affecting the throughput of the device directly – e.g., forwarding N # of packets/sec versus forwarding 2N

Agenda

- Reverse Path Forwarding Check
- Hop Count - Optional

Intra-AS: Possible Hop-Count Options

1) Use a separate EthType for TTL

0x88a8	B-VID	EthType	TTL	EthType	I-SID

2) Use a extended VLAN-tag

EthType	B-VID + TTL	EthType	I-SID

3) Use part of I-SID tag

0x88a8	B-VID	EthType	TTL+ I-SID

Pros & Cons of Different Options

Options	Pros	Cons
1. Use a separate tag	➢ Backward compatibility w/ existing 802.1Q	➢ Requires additional 4 bytes ➢ Requires additional lookup in BCBs
2. Use extended VLAN tag	➢ More efficient in size than option (1) ➢ Requires no additional lookup in BCBs	➢ Still requires additional 2 bytes ➢ No backward compatibility
1. Use I-tag	➢ Most efficient in size than option (1) ➢ Backward compatible w/ existing 802.1Q	➢ Requires additional lookup in BCBs

Inter-AS: Only a Single Hop-Count Option

In case of inter-AS, there is basically a single option for Hop-Count and that is to be included as part of I-tag because the service interface for inter-AS scenarios only contains I-tag.

0x88a8	B-VID	EthType	TTL+ I-SID