

802.1Qbp
Shared Tree (*.G) Algorithms
(for head end MCAST ECMP)

Peter Ashwood-Smith
peter.ashwoodsmith@huawei.com

bp-ashwood-shared-trees-1011-v3

Motivation

- 802.1Qbp is introducing new ECMP behavior in an 802.1aq network.
- There is a requirement to do ECMP (head end) over multicast trees.
- So far we have only discussed the (S,G) multicast trees (existing .1aq style and Ben's alternatives).
- I'd like to discuss some simple (*,G) options since state reduction without loss of functionality is possible especially in DC networks.

N.B (S,G) is source/group specific tree, i.e. <SpSourceID>||<SID> in the DA
(* ,G) is shared by all sources but one group i.e. <Constant>||<ISID> in the DA

Considerations

- What we really want is a minimum spanning tree that covers just a subset of the nodes (those in the ISID).
- This is referred to as a Steiner Tree.
- A Steiner Tree computation is **NP-Complete**.
- “**Non Polynomial**” means its $\gg O(N^c)$ for any constant **c**.
- “**Complete**” is a way of saying we won’t likely solve it here ..
- Basically its one of those problems that you have to enumerate all $O(n!)$ solutions and pick the best.

Solutions

There are a few less optimal (*,G) solutions:

1. Pick some node as a root and use SPF from 'it' as the tree.
 - This is $O(n \cdot \log N)$ but sends traffic everywhere!!!
 - So .. modify above by pruning per ISID (SPF is template).
 - This is $O(N \cdot \log N + I \cdot \log I)$
 - Still non shortest path routing but state is minimal
2. Other solutions aimed at reducing non shortest path routing issues but increase CPU.. these are FFS. (e.g. enforce root = member of ISID)

One possible proposal

- The 802.1aq CIST algorithm (which is just the STP algorithm done as a computation), can be reused for per ISID (*,G) trees in .1Qbp
- The multicast address format can be the existing PBB format i.e: **00-1e-83-xx-xx-xx** (where **xx-xx..** is the ISID)
- 16 different shared trees can be computed by finding the lowest BridgIdentifier under the 16 802.1aq ECT masks i.e. 0x00..., 0xff..., 0x11..., 0x22... ... 0xee...
- These shared trees produce almost symmetric congruent results to the .1aq (S,G) trees in fat tree networks.
- Root selection automatic based on algorithm, auto recovery to new root etc. No explicit encoding of root in DA required.
- Can use F-TAG with TTL, or can rely on digest for loop prevention, or both....
- Can use same B-VID as unicast (no SVL), or different (with SVL) or even no B-VID.

Example #1

The network topology shows a central node 3 (yellow) connected to nodes 1, 4, 5, 6, 8, 9, and 11. Node 1 is the root. Nodes 4, 6, 8, 9, and 11 are connected to node 3 via dashed green lines. Nodes 1, 2, 3, 5, 7, and 10 are connected to node 3 via solid pink lines. A thick dashed black arrow points from node 1 to node 3, labeled 'Root'. A label 'FDB @3' is positioned above the FDB table.

Area ID Configuration:
 Area ID: 10.0001
 AreaName: AreaTestNet
 Level 1 Level 2
 ECT-ALG: 40
 I-SID: Both-17

 Inst Real Virt

Link Configuration:
 Link Name: Link5-6
 Metric: 10
 Bandwidth: 1000
 AreaName: AreaTestNet

Connection Configuration:
 Address: 127.0.0.1
 Port: 7001

Add Configuration:

Display Configuration:
 Node ID Metric 10
 Interface I-SID C
 Label BW

FLG	IN/IF	DESTINATION ADDR	BUID	OUT/IF(s)
	if/00	0000-0000-0000	0039	<if/1,if/5,if/6 >
	if/00	011e-8300-0010	0040	<if/5,if/6 >
	if/00	011e-8300-0011	0040	<if/1,if/5,if/6 >
	if/**	4455-6677-0101	0040	<if/1 <4455-6677-0101 >
	if/**	4455-6677-0101	0041	<if/1 <4455-6677-0101 >
	if/**	4455-6677-0102	0040	<if/2 <4455-6677-0102 >

A (*,G) is computed using the Lowest Bridge Identifier (node 1) CIST algorithm.

The full tree is shown in pink.

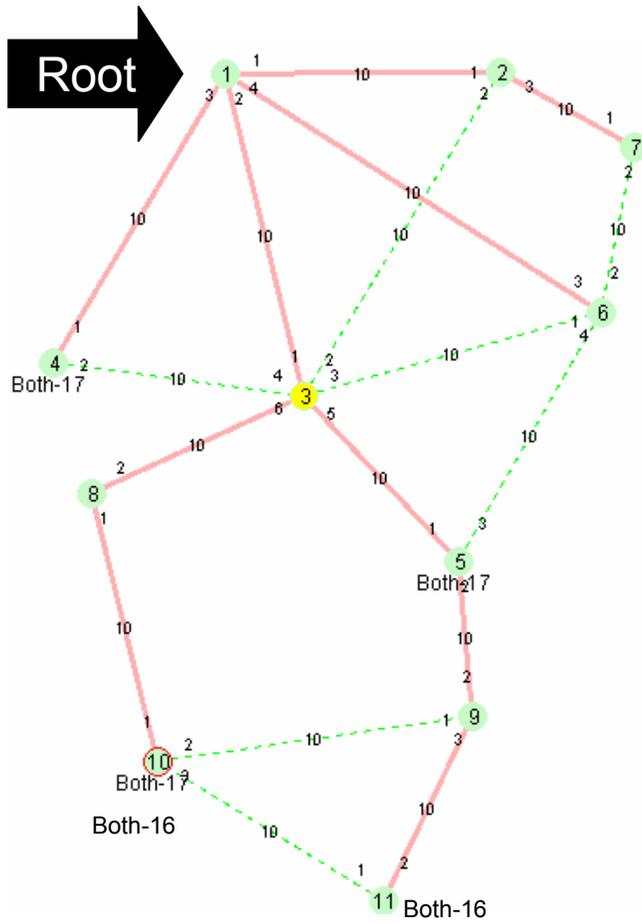
Two ISIDs are pruned against this tree for Multicast, sub trees below: ISID 17 and ISID 16.



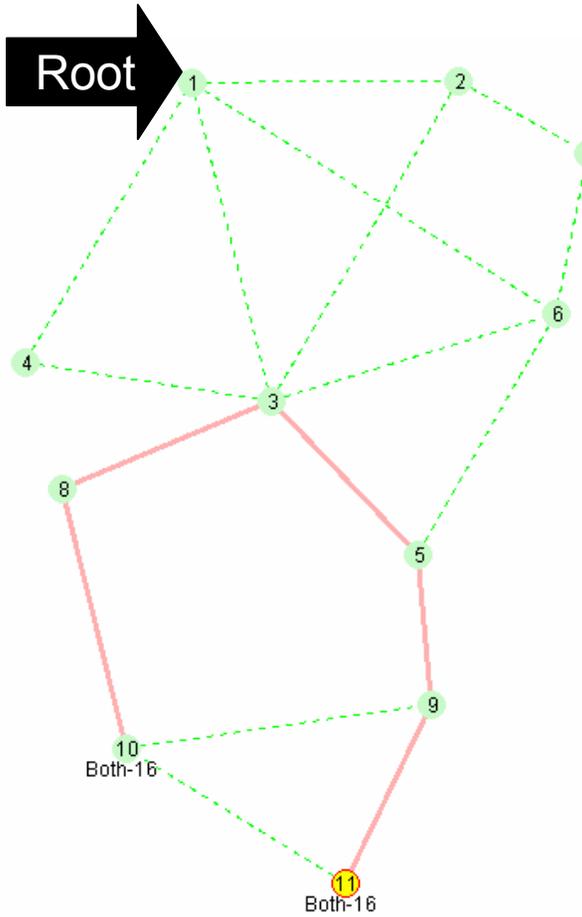
We show the Mcast state at node 3 for Each ISID.

- CIST
- Pruned for ISID 16
- Pruned for ISID 17

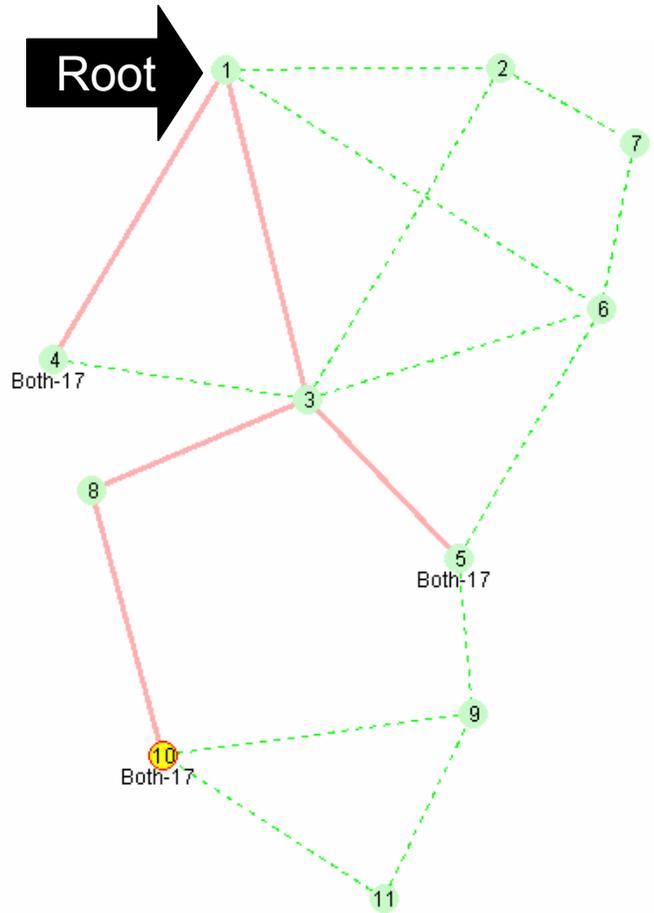
Example #1 – pruning



FULL MASK 0x00
(ROOT=1) TREE



ISID 16
PRUNED



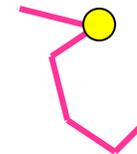
ISID 17
PRUNED

Example#2

A (*,G) is computed using the **highest** Bridge Identifier (node 11) i.e. CIST algorithm XOR 0xff.

The full tree is shown in pink.

One ISIDs is pruned against this tree for Multicast, sub trees below: ISID 18



We show the Mcast state at node 3 for Each ISID.

File Config Replication Zoom

Area ID: 10.0001
 AreaName: AreaTestNet
 Level 1 Level 2
 ECT-ALG: 41
 I-SID: New

 Inst Real Virt

Link
 Link Name: Link10-11
 Metric: 10
 Bandwidth: 1000
 AreaName: AreaTestNet

Connection
 Address: 127.0.0.1
 Port: 7001
 con1

Add

Display
 Node ID Metric: 10
 Interface I-SID: C
 Label BW

FDB @3

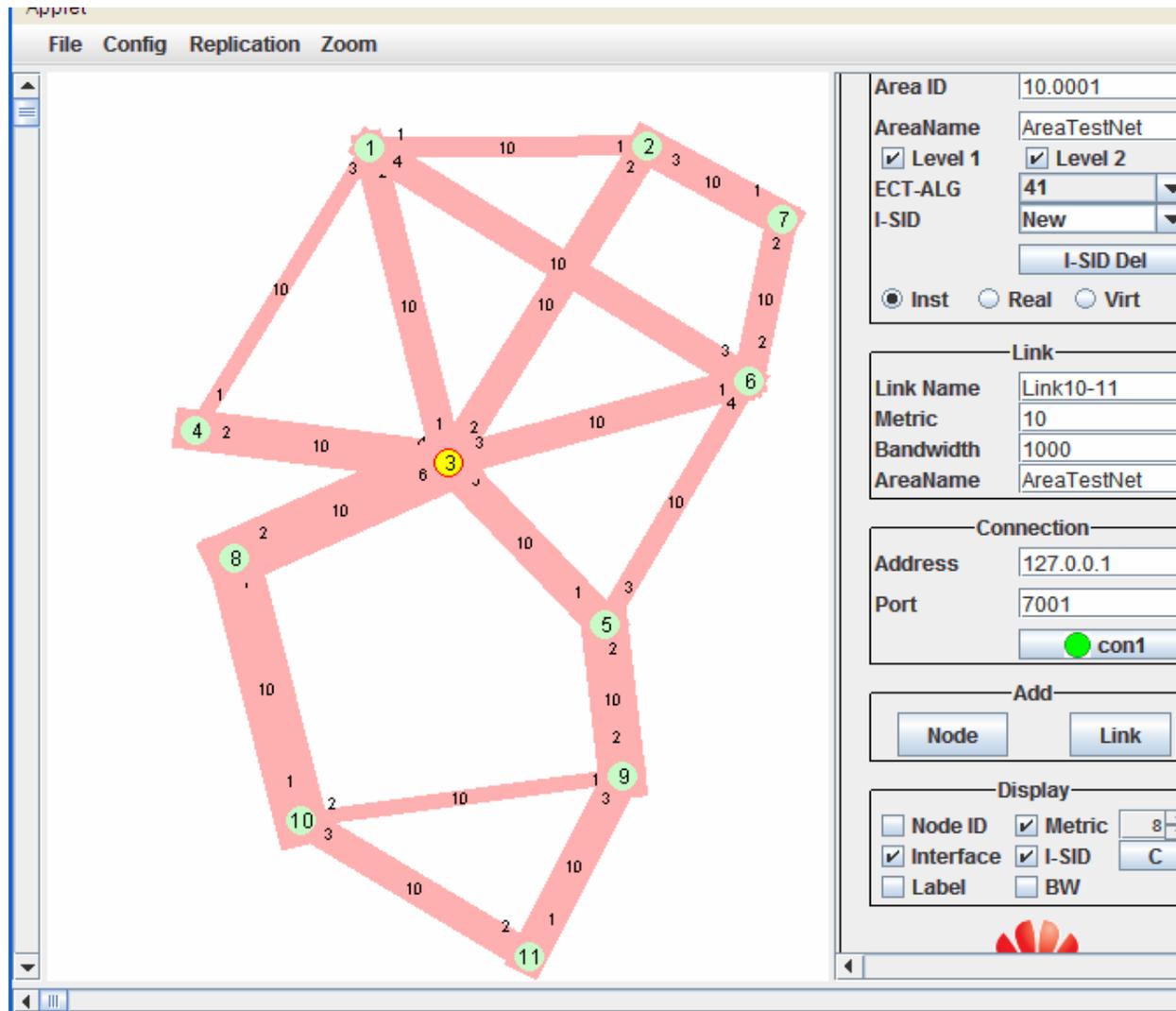
Root

FLG	IN/IF	DESTINATION ADDR	BUID	OUT/IF(s)
	if/00	0000-0000-0000	0039	<if/1,if/5,if/6 >
	if/00	011e-8300-0010	0040	<if/5,if/6 >
	if/00	011e-8300-0011	0040	<if/1,if/5,if/6 >
	if/00	011e-8300-0012	0041	<if/4,if/6 >



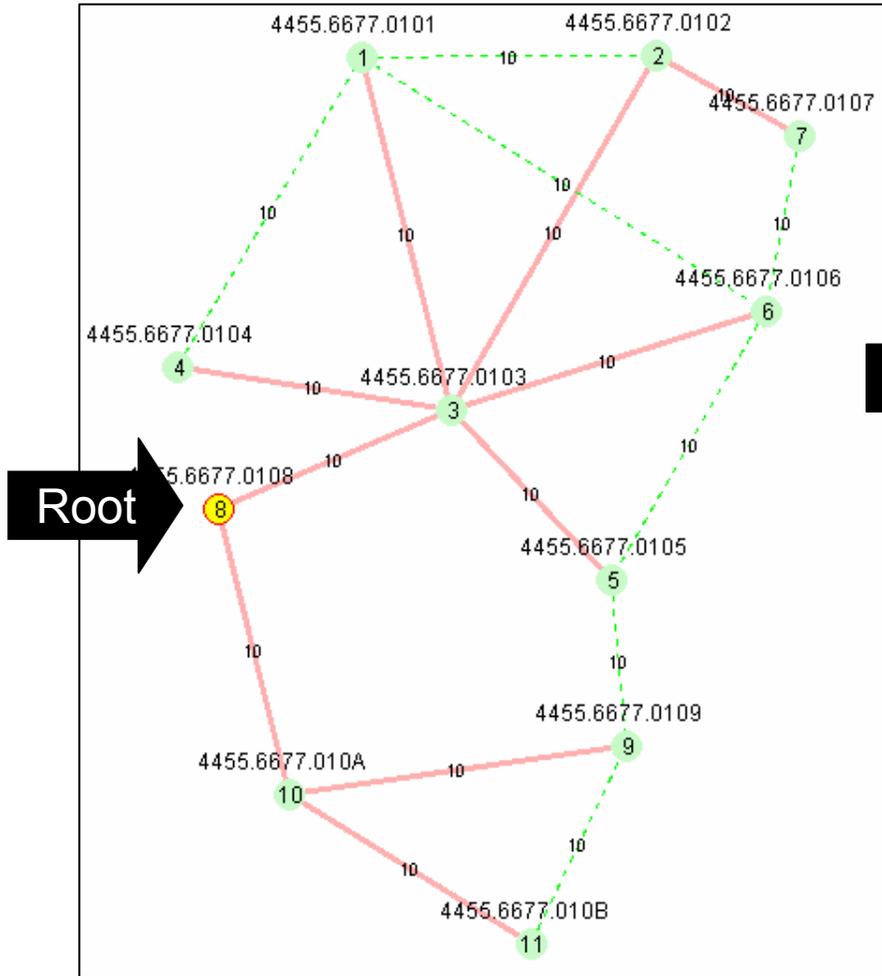
- CIST
- Pruned for ISID 16
- Pruned for ISID 17
- Pruned for ISID 18

Example#3- Coverage is not bad

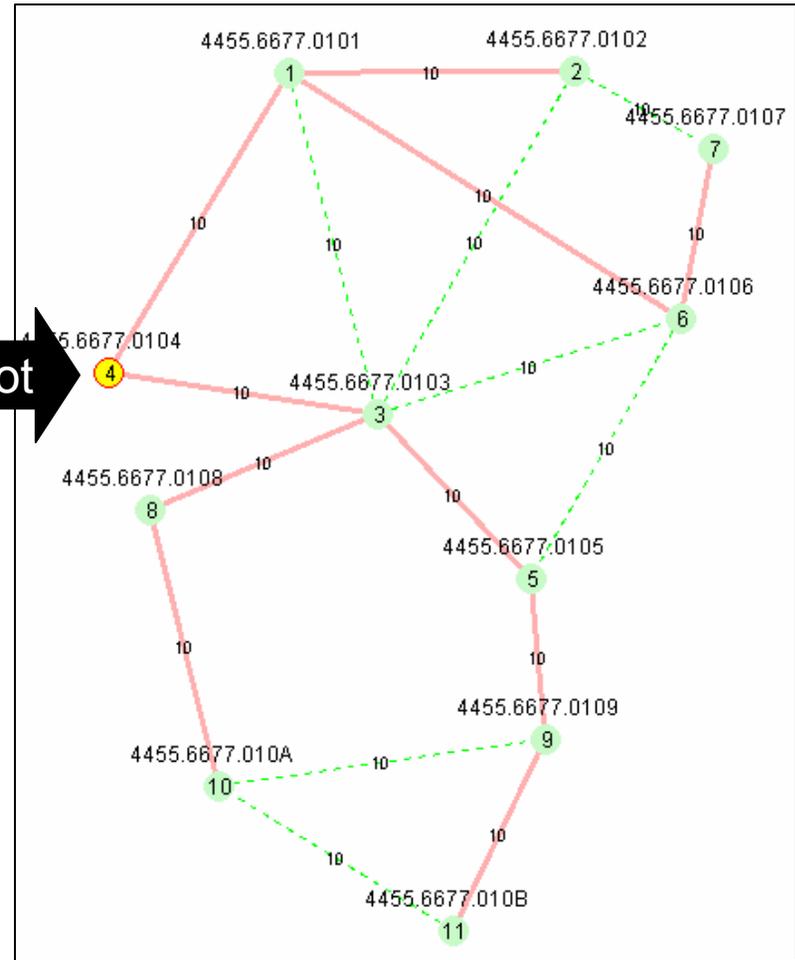


ALL 16 (*,G) Trees shown superimposed. Basically the CIST algorithm 16 times but with different root choices based on BridgeIdentifier XOR Mask[i]

Example#3- Some of the individual trees



ALG MASK=0x8888.
So node .. 108 is root.



ALG MASK=0x444444.
so node 104 is root.

Basic Algorithm

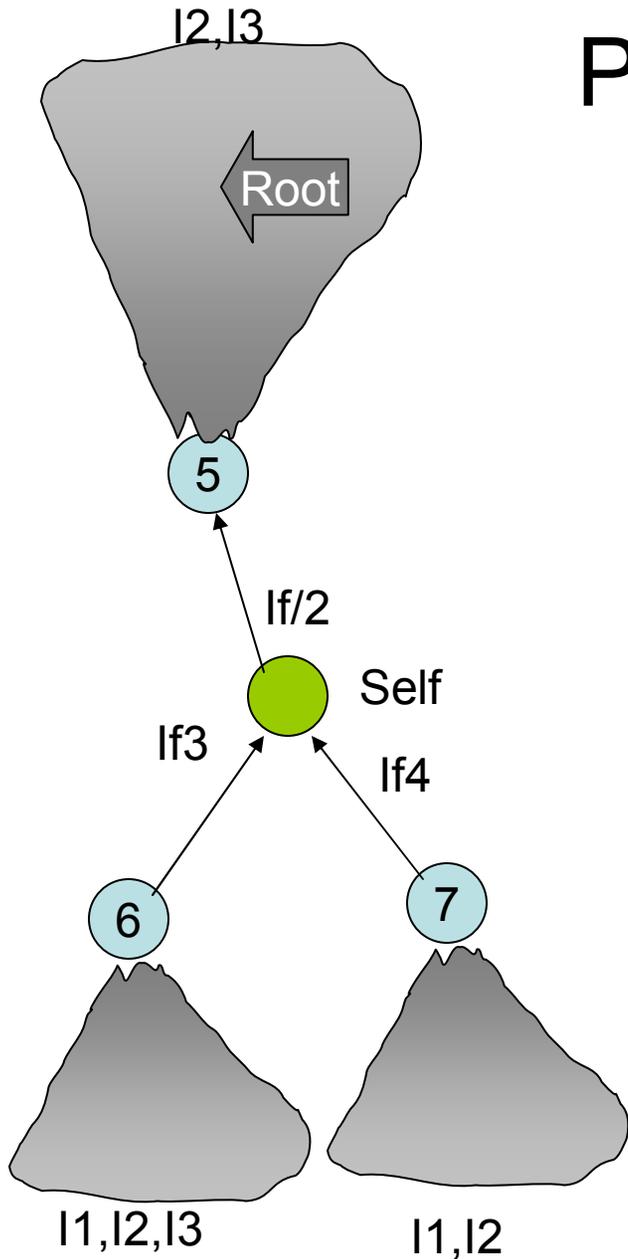
```
Compute Shared Tree (alg, self) { // alg==0 => .1aq CIST  
  
    root = find lowest Bridgelfdentifier XOR Mask[alg]*  
  
    run SPF from root where  
        tie break on equal cost winner =  
            lowestBridgelfdentifier XOR Mask[alg]*  
}
```

Multicast DA per ISID can then easily be generated by sorting the set of all ISIDs and the interface to reach that that ISID... Next slide..

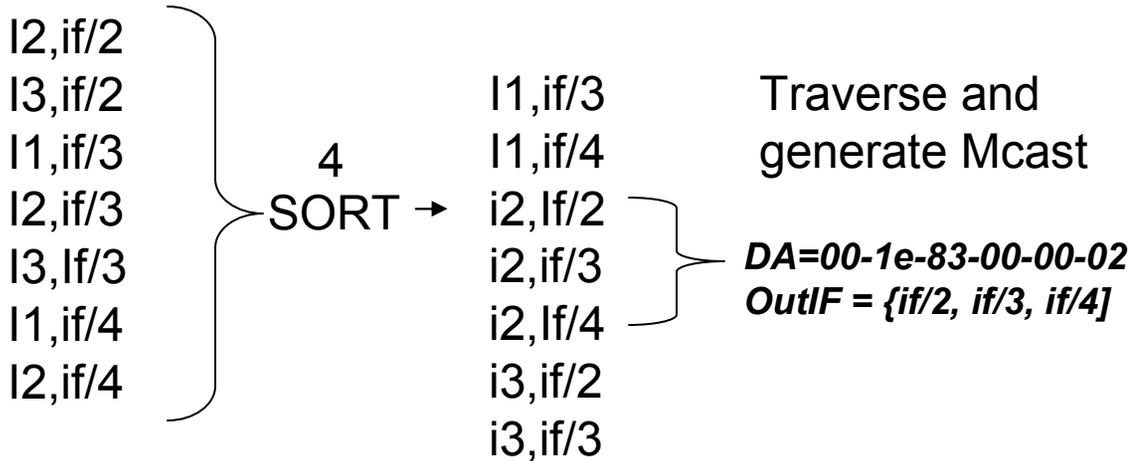
So total run time is $O(16 \times [(N \times \text{Log}(N)) + (I \times \text{Log}(I))])$

* Recall Mask[] = {0x00..., 0xff., 0x11..., 0x22..., 0x33..., 0xee.. }

Pruning - One Possibility



1. At self do the SPF from selected root.
Result is upward pointing parent pointers to root.
2. For each node in network assign it the local interface that reaches it. Eg: 5 and everything above it via if/2; 7 and everything below it by if/4 etc.
3. Then traverse network and generate a list of.
<ISID, IF/#> records ..will have lots of duplicates.



Ignore if only reachable via one interface ..

100+ node example – ISID 100 with 4 attachment points

The screenshot displays a network management interface with a large network topology. The topology consists of numerous nodes (represented by numbers 1-100) connected by links. Four attachment points are highlighted in red, labeled 'both-100'. A central window titled '802.1aq calculation engine' shows the following output:

```
if/00 011e-8300-0064 0040 < if/5, if/6, if/21 >
if/* 6655-4433-2201 0040 < if/1 < 6655-4433-2201 > >
if/* 6655-4433-2202 0040 < if/3 < 6655-4433-2228 > >
if/* 6655-4433-2203 0040 < if/3 < 6655-4433-2228 > >
```

The right sidebar shows configuration details for a node and link:

Information

Node

- Instance ID: 77
- Node ID: 6655.4433.224D
- Area ID: 10.0001
- AreaName: AreaTestNet
- Level 1 Level 2
- ECT-ALG: 40
- I-SID: both-100
- I-SID Del: [button]
- Inst Real Virt

Link

- Link Name: link0-0
- Metric: 10
- Bandwidth: 1000
- AreaName: AreaTestNet

Connection

- Address: 127.0.0.1
- Port: 7001
- con1 [button]

Add

- Node [button]
- Link [button]

Display

- Node ID Metric: 10
- Interface I-SID: C
- Label BW

HUAWEI

Notes : Addressing Options

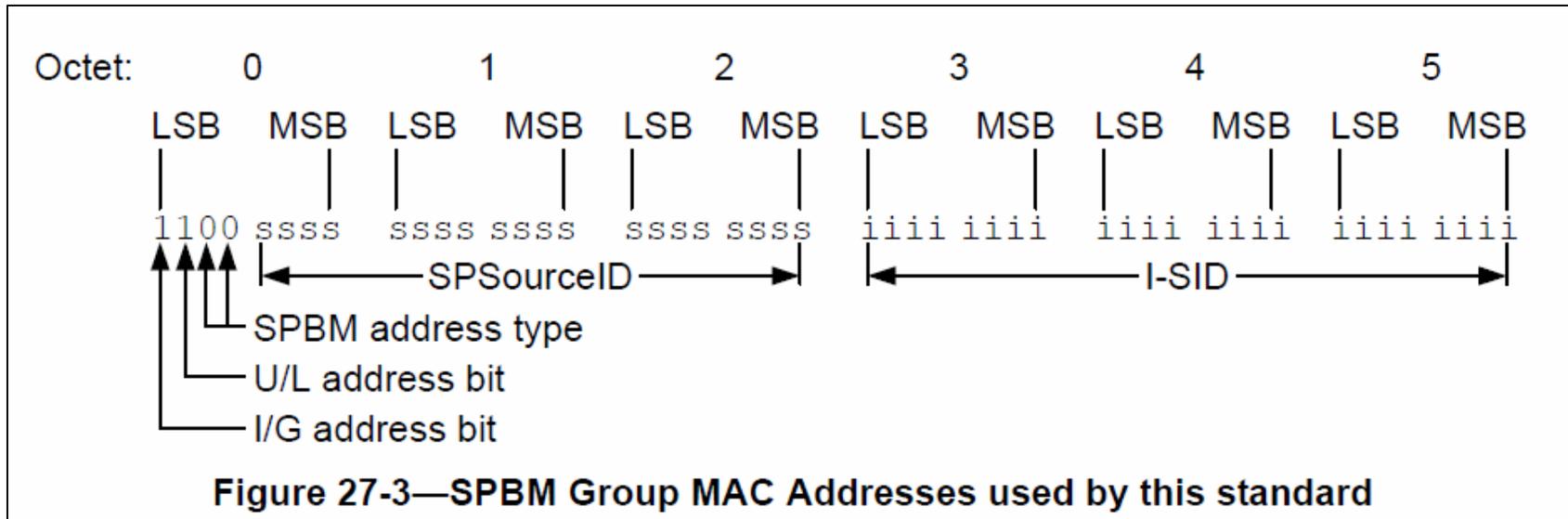
1. ***DA = 01-1e-83-xx-xx-xx & VID = F(ALG)***
 - ***Tree is identified by the VID so overlapping ISIDs (as used for ECMP) requires the VID to differentiate.***
 - ***Local bit NOT set so can co-exist with (S,G) .1aq trees.***
2. ***DA = F(ALG)-xx-xx-xx & VID = Const | Absent***
 - ***Tree is identified by the DA so overlapping ISIDs do not require VID to differentiate and in fact VID can be absent even with overlapping ISIDs.***

Note that encoding root of (,G) tree in address appears unnecessary as root is a function of the Algorithm used to pick tree.*

Note 24 bit ISID value represented as xx-xx-xx

Addressing Option 2 Cont'd

802.1aq Group MAC format



SPBM uses only address type 00, therefore we could use address type 01 to implement:

$$DA = 1101-F(ALG):20-xxxxxxxxxxxx.....xxxxxxxx$$

So this gives up to 2^{20} shared trees more than we can do with B-VID!

Questions

- Do we need new *BridgePriority* for (*,G) root selection?
 - We could allow greater flexibility with separate ECT tie breaking *BridgePriority* and ECMP root selection *BridgePriority* but it adds more complexity...
 - We have opaque TLV that can be used to carry new ‘things’ without involving ISIS-wg.