ERICSSON

Description of Explicit Topologies

János Farkas
janos.farkas@ericsson.com

March 18, 2014

Format *A*: Port ID Based

- This is the format of 802.1Qca D0.6
- Format A is based on listing Bridge Ports that are part of the topology, where a Bridge Port is identified by an IS-IS System ID, Circuit ID tuple
- The connectivity provided by a Bridge Port is included in the topology if the Port ID is included; therefore, each bridge or station connected to the same LAN is also included in the topology
- Format A only requires ordering for a loose hop of a p2p path that mixes loose and strict hops
  - Ordering is not required either in fully specified or in completely loose cases
  - A tree (mp2mp) is always either fully specified or completely loose
- Otherwise, Format A does not require any particular ordering of the hops, but ordering is allowed in case of p2p paths
- Tie-breaking for a link: use the numerically lower System ID

Format *B*: Order Based

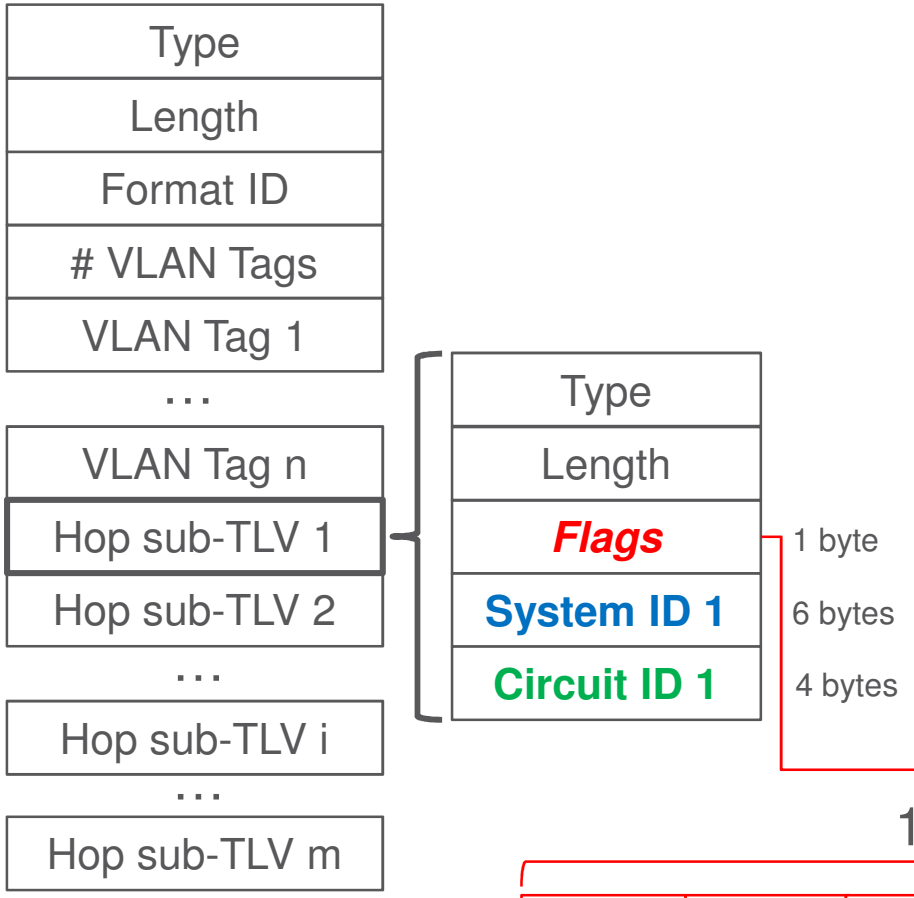
- Format B is based on the ordered list of Nodal IDs for describing all kinds of topologies
- A chain (or ear) out of the topology is described by an ordered list
  - A p2p path is a single chain
  - The smallest chain is a single link
- Arbitrary order between chains
- Each node involved in the topology appears at least once in the descriptor
- The System ID is the Nodal ID for IS-IS

Parallel Links

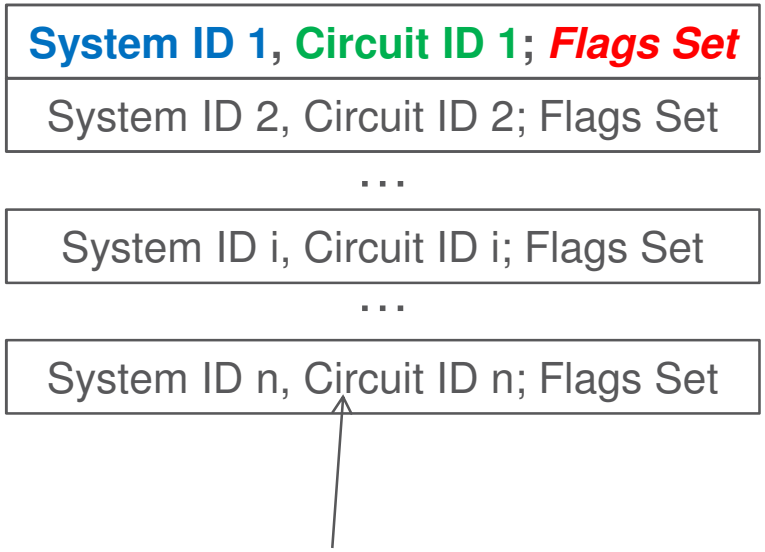
- Port ID has to be also supported in case of Format B in order to be able to distinguish parallel links between a pair of bridges
- Therefore, the same TLV structure can be used for both formats

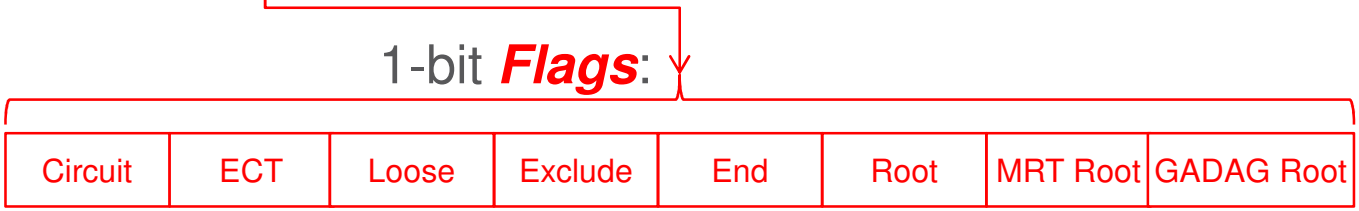
Descriptor

- 802.1Qca D0.6

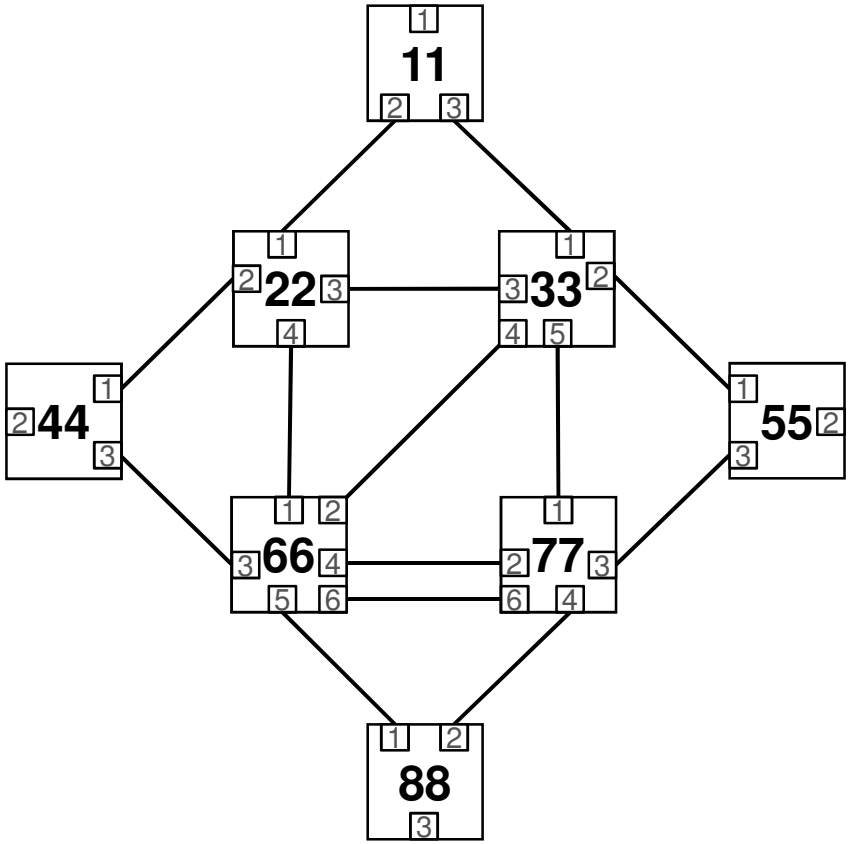


- This ‘translated’ version is used in the following:





Example Network

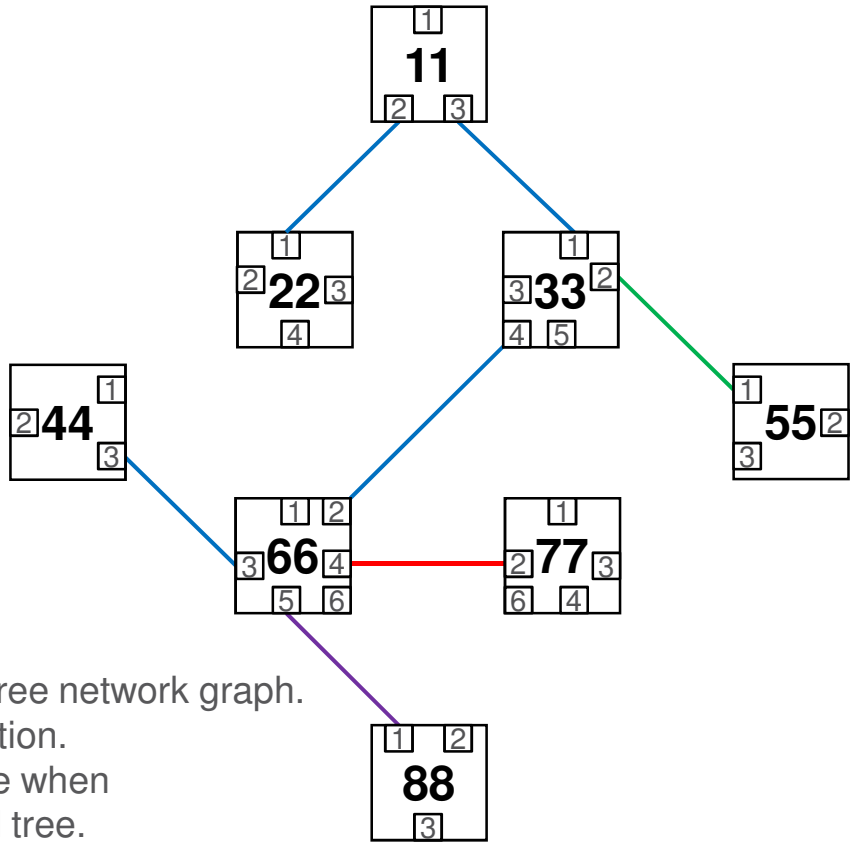


A Fully Specified Spanning Tree

Format A
arbitrary order

11, 2; Circuit, End
11, 3; Circuit, End
44, 3; Circuit, End
55, 1; Circuit, End
88, 1; Circuit, End
33, 4; Circuit
66, 4; Circuit

77 bytes



Format B
exact order
for each chain

22
11; End
33
66
44; End
33
55; End
66, 4; Circuit
77
66
88; End

81 bytes

Note that a tree is just a loop-free network graph.
Root only matters for computation.
Root does not matter any more when
just describing a fully specified tree.

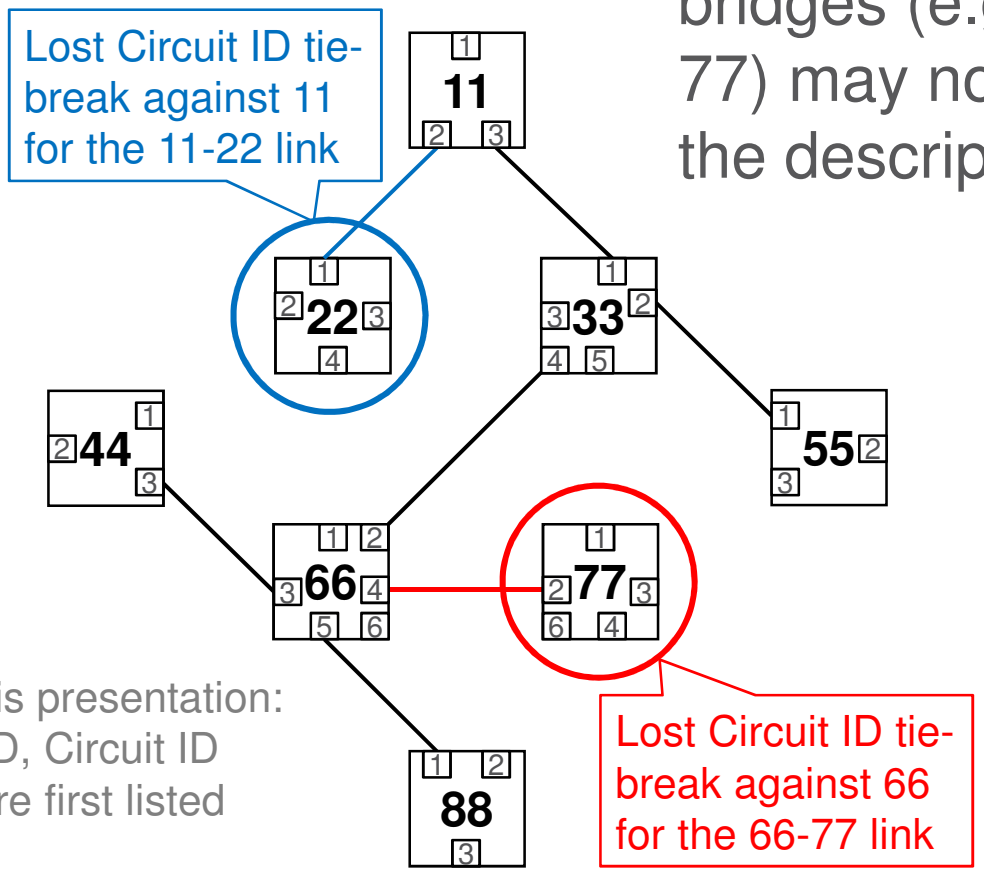
Circuit ID has to be used for parallel links in every case

A Fully Specified Spanning Tree Format *A* Peculiarities

Format A
arbitrary order

11, 2; Circuit, End
11, 3; Circuit, End
44, 3; Circuit, End
55, 1; Circuit, End
88, 1; Circuit, End
33, 4; Circuit
66, 4; Circuit

77 bytes



- Tie-breaking looser bridges (e.g. 22 and 77) may not appear in the descriptor

The order applied in this presentation: Ascending in System ID, Circuit ID such that End Points are first listed

A Fully Specified Spanning Tree Format *A* Peculiarities – cont'd

Format A
arbitrary order

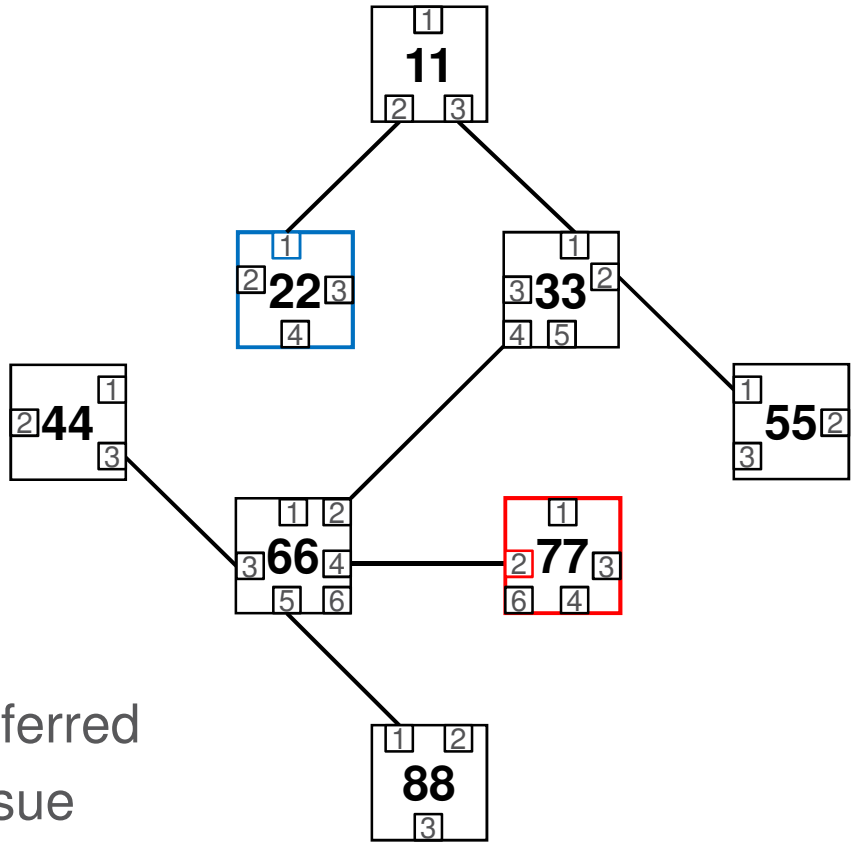
11, 2; Circuit, End
11, 3; Circuit, End
44, 3; Circuit, End
55, 1; Circuit, End
88, 1; Circuit, End
33, 4; Circuit
66, 4; Circuit

77 bytes

Format A
arbitrary order

11, 2; Circuit, End
11, 3; Circuit, End
44, 3; Circuit, End
55, 1; Circuit, End
88, 1; Circuit, End
22, 1; Circuit
33, 4; Circuit
66, 4; Circuit
77, 2; Circuit

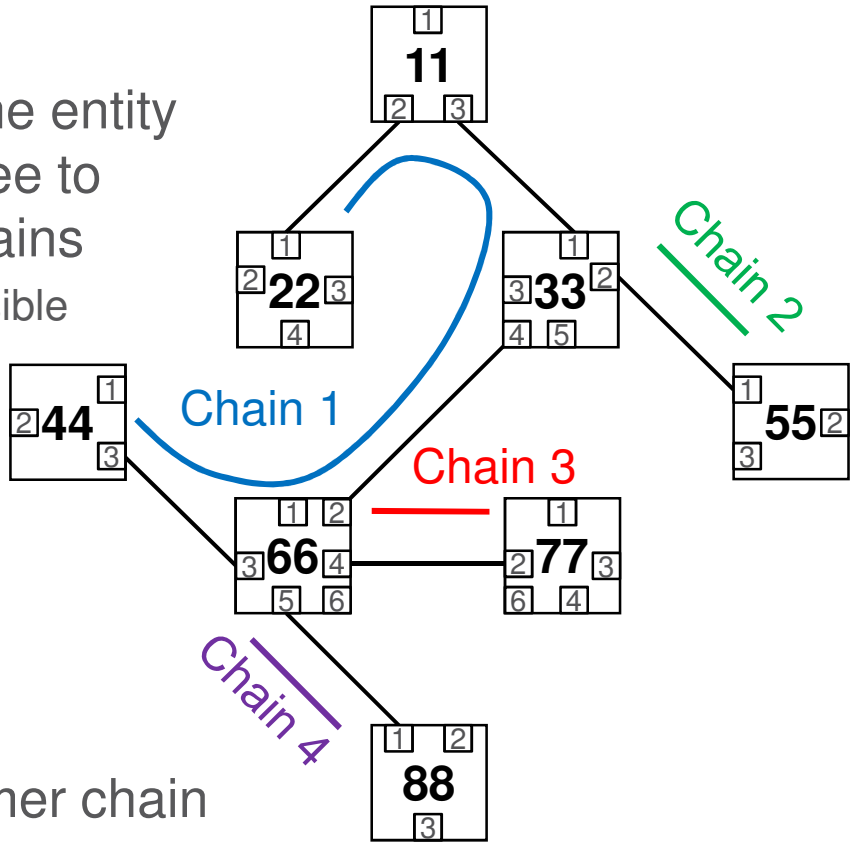
99 bytes



- Each bridge can be listed if that is preferred
- Redundant items do not cause any issue

A Fully Specified Spanning Tree Format *B* Peculiarities

- Exact order for each chain
- Arbitrary order between chains
- It is the task of the entity describing the tree to figure out the chains
  - e.g. longest possible chains for least bytes descriptor
- Beginning of new chain is indicated by a System ID that already appears in a former chain



Format B

22	Chain 1
11; End	
33	
66	
44; End	
33	Chain 2
55; End	
66, 4; Circuit	Chain 3
77	
66	Chain 4
88; End	

81 bytes

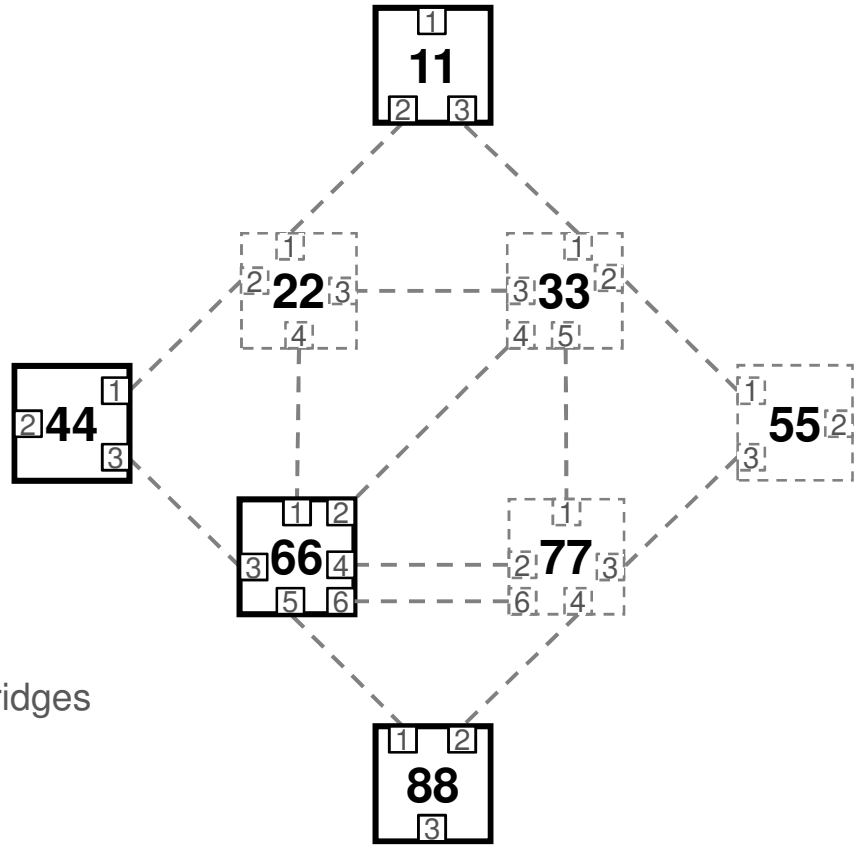
A Completely Loose Tree

Note that order does not matter in either format

Format A

11; Loose, End
44; Loose, End
88; Loose, End
66; Loose, Root

28 bytes



Format B

11; Loose, End
44; Loose, End
66; Loose, Root
88; Loose, End

28 bytes

Root matters because the bridges have to compute.

A Fully Specified P2P Path

Format A
in arbitrary order

11, 2; Circuit, End
88, 2; Circuit, End
22, 3; Circuit
33, 4; Circuit
66, 6; Circuit

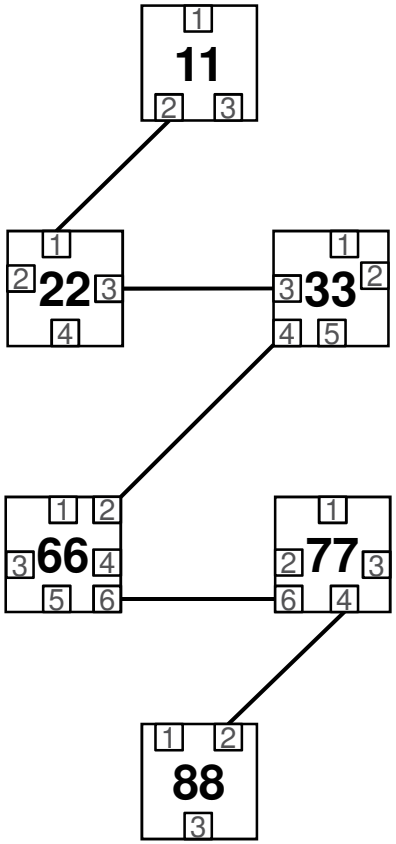
55 bytes

77 lost Circuit ID tie-break against 66 for the 66-77 link

Format A
in exact order

11; End
22
33
66, 6; Circuit
77
88; End

46 bytes



Format B
exact order

11; End
22
33
66, 6; Circuit
77
88; End

46 bytes

Format A (802.1Qca D0.6) allows
exact order of System IDs for p2p paths:
Exact order has to be followed if Circuit ID is not present

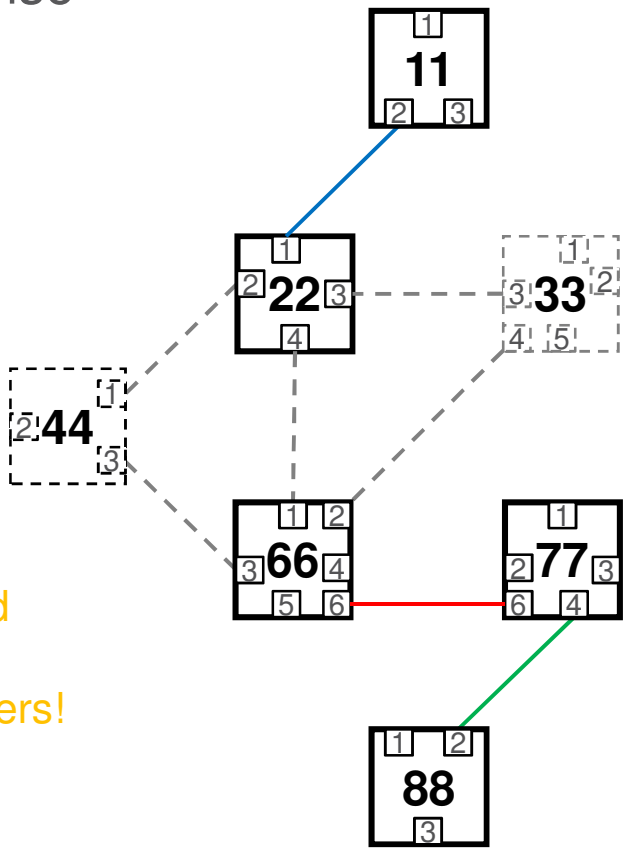
A Mixed P2P Path (Mixed Strict and Loose Hops)

Format A

exact order for loose hop
arbitrary order otherwise

11, 2; Circuit, End
66, 6; Circuit, Loose
88, 3; Circuit, End

33 bytes



a loose hop is related
to the previous hop;
therefore, order matters!

Format B

exact order

11; End
22
66, 6; Circuit, Loose
77
88; End

39 bytes

Circuit ID has to be used for parallel links in every case

A Mixed P2P Path Format *A* Peculiarities

Format A

exact order for loose hop
arbitrary order otherwise

11, 2; Circuit, End
66, 6; Circuit, Loose
88, 3; Circuit, End

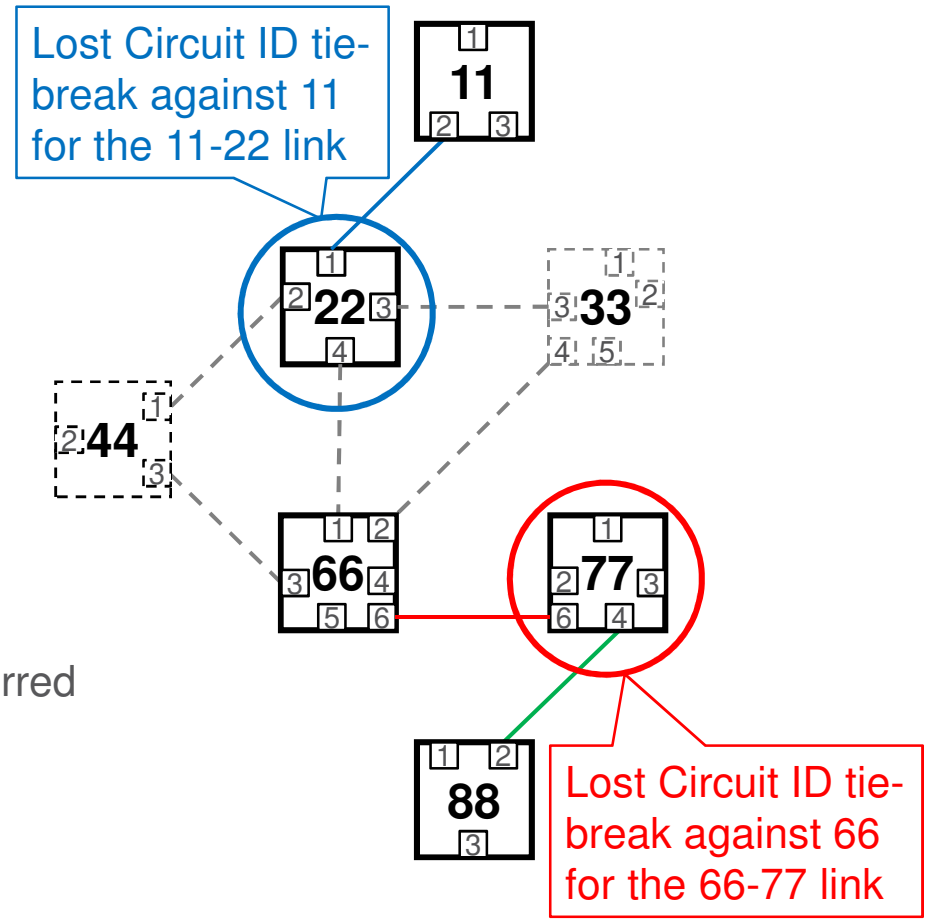
33 bytes

a loose hop is related to the previous hop; therefore, order matters!

11, 2; Circuit, End
88, 3; Circuit, End
22, 1; Circuit
66, 6; Circuit, Loose
77, 6; Circuit

55 bytes

Each bridge can be listed if that is preferred

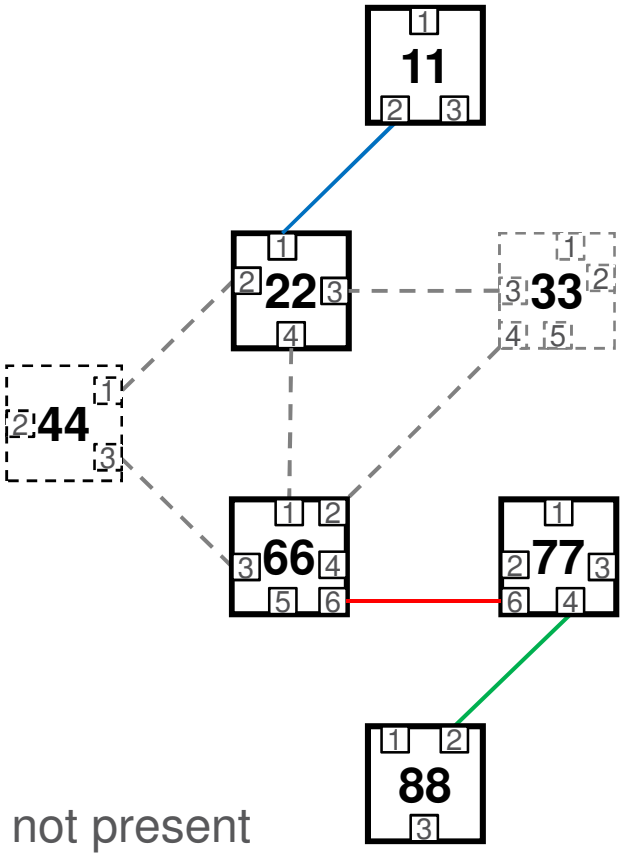


A Mixed P2P Path
Format ***A*** Peculiarities – cont'd

Format A
exact order

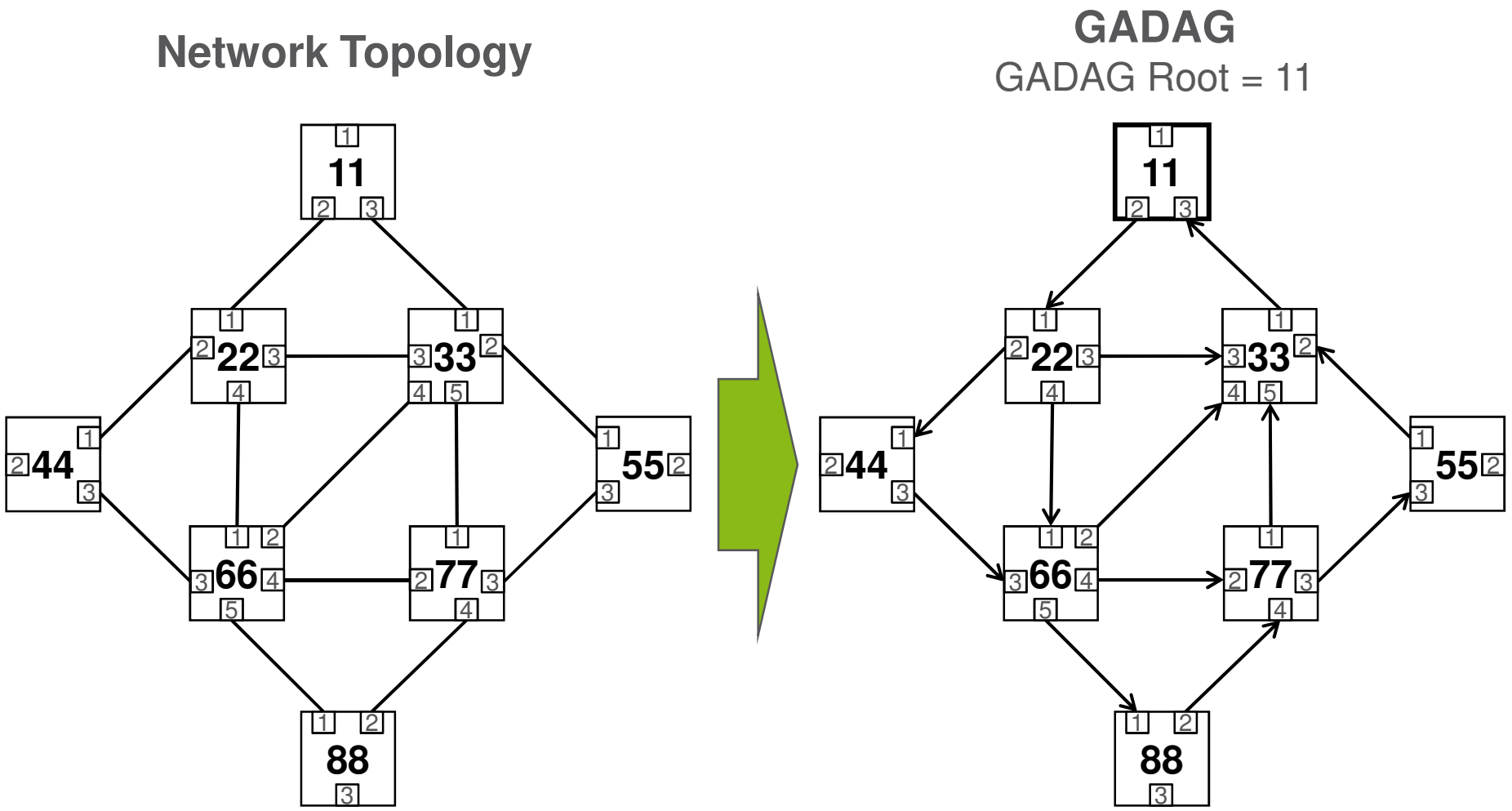
11; End
22
66, 6; Circuit, Loose
77
88; End

39 bytes



Format A (802.1Qca D0.6) allows
exact order of System IDs for p2p paths:
Exact order has to be followed if Circuit ID is not present

A GADAG Example

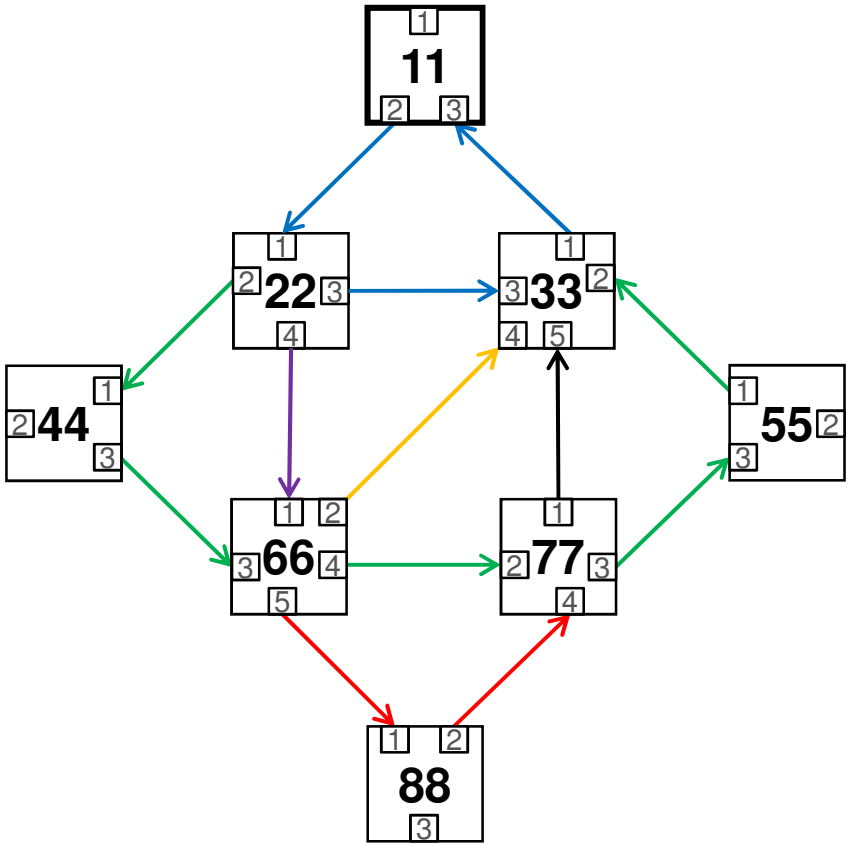


GADAG Description

Format A
arbitrary order

11, 2; Circuit, GADAG Root
22, 2; Circuit
22, 3; Circuit
22, 4; Circuit
33, 1; Circuit
44, 3; Circuit
55, 1; Circuit
66, 2; Circuit
66, 4; Circuit
66, 5; Circuit
77, 1; Circuit
77, 3; Circuit
88, 2; Circuit

143 bytes



Format B
specific order

11; GADAG Root
22
33
11; GADAG Root
22
44
66
77
55
33
66
88
77
22
66
66
33
77
33

133 bytes

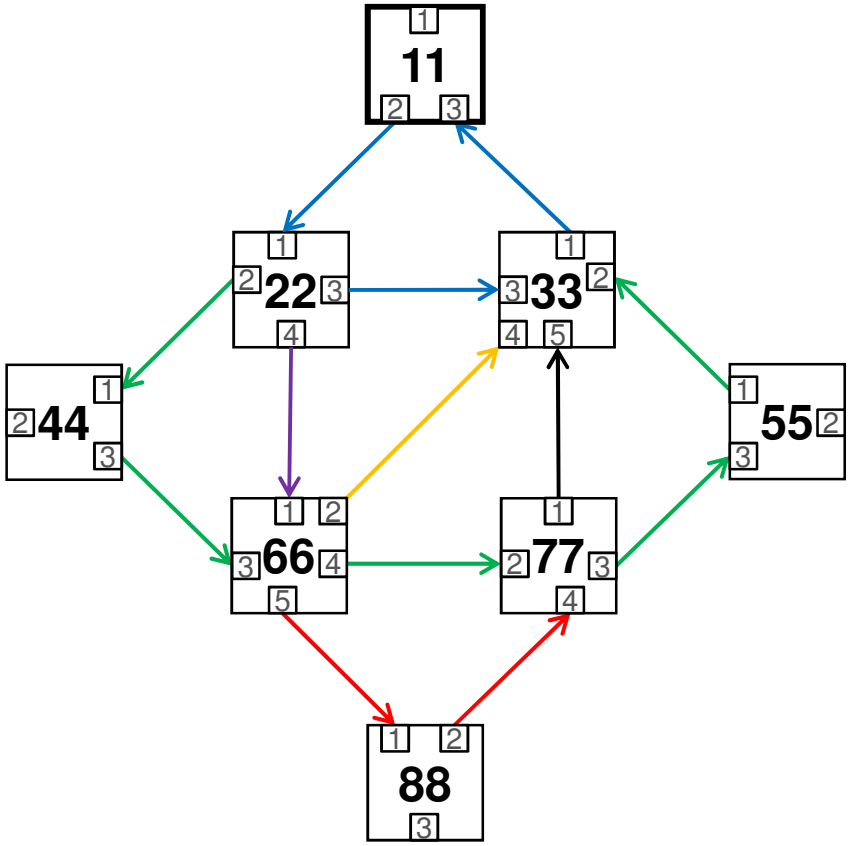
GADAG Description Format *A* Peculiarities

Format A
11, 2; Circuit, GADAG Root
22, 2; Circuit
22, 3; Circuit
22, 4; Circuit
33, 1; Circuit
44, 3; Circuit
55, 1; Circuit
66, 2; Circuit
66, 4; Circuit
66, 5; Circuit
77, 1; Circuit
77, 3; Circuit
88, 2; Circuit

Bridge, Port order

Format A
11, 2; Circuit, GADAG Root
22, 3; Circuit
33, 1; Circuit
22, 2; Circuit
44, 3; Circuit
66, 4; Circuit
77, 3; Circuit
55, 1; Circuit
66, 5; Circuit
88, 2; Circuit
22, 4; Circuit
66, 2; Circuit
77, 1; Circuit

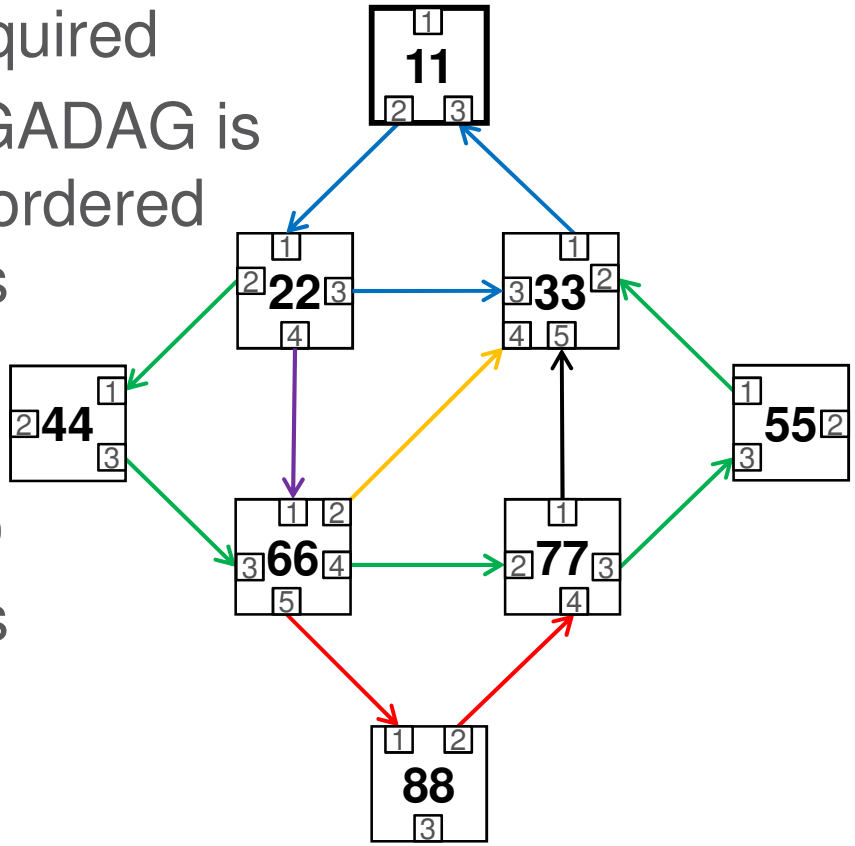
ear order



- Each edge of the graph is specified by the outbound port
- Arbitrary order can be applied; therefore,
- The graph can be described bridge by bridge and port by port

GADAG Description Format *B* Peculiarities

- Specific order required
- Each ear of the GADAG is described by an ordered list of System IDs
- Arbitrary order among ears (e.g. comp order)
- A new ear begins and ends with a System ID that is already in the list



Format B
specific order

11; GADAG Root	Ear 1
22	
33	
11; GADAG Root	
22	Ear 2
44	
66	
77	
55	
33	
66	Ear 3
88	
77	
22	Ear 4
66	
66	Ear 5
33	
77	Ear 6
33	

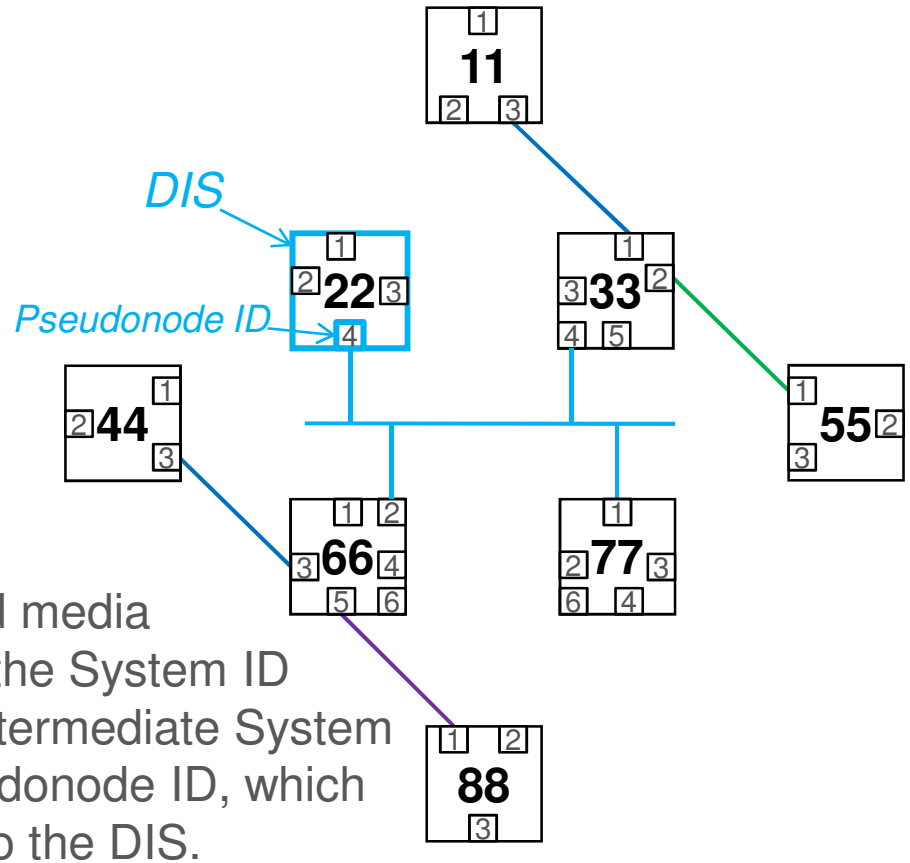
133 bytes

Shared Media LAN Example

Format A
arbitrary order

11, 3; Circuit, End
44, 3; Circuit, End
55, 1; Circuit, End
88, 1; Circuit, End
22, 4; Circuit

55 bytes



ISO 10589: A shared media LAN is identified by the System ID of the Designated Intermediate System (DIS) and by a Pseudonode ID, which is a Circuit ID local to the DIS.

Format B
exact order
for each chain

11; End
33
22, 4; Circuit
66
44; End
33
55; End
22, 4; Circuit
77
66
88; End

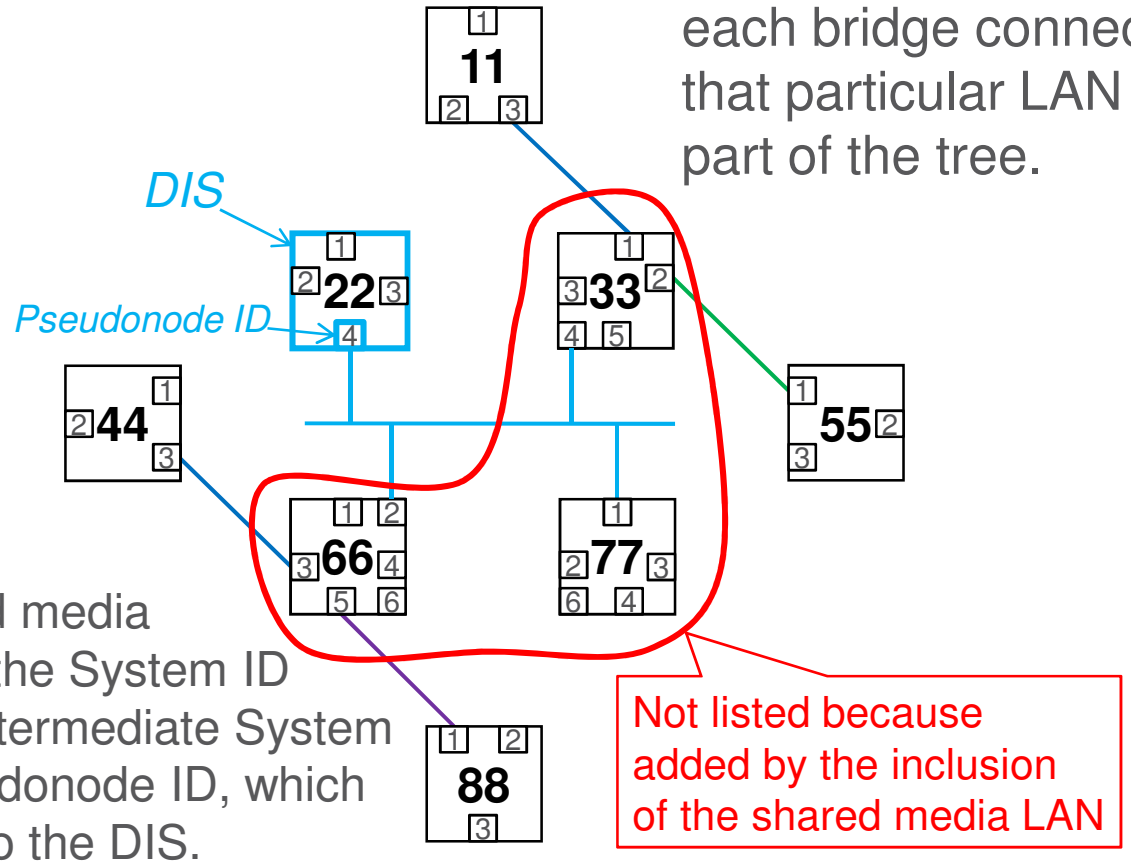
85 bytes

Shared Media LAN Example Format *A* Peculiarities

Format A
arbitrary order

11, 3; Circuit, End
44, 3; Circuit, End
55, 1; Circuit, End
88, 1; Circuit, End
22, 4; Circuit

55 bytes



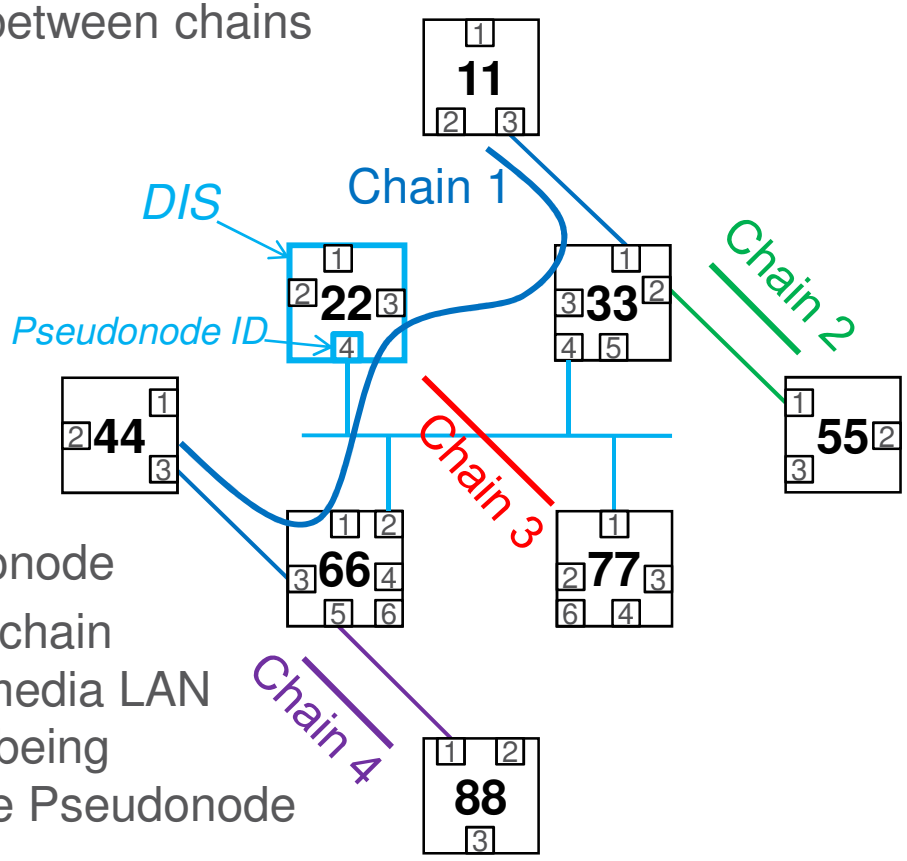
- If a shared media LAN is part of an explicit tree, then each bridge connected by that particular LAN is also part of the tree.

ISO 10589: A shared media LAN is identified by the System ID of the Designated Intermediate System (DIS) and by a Pseudonode ID, which is a Circuit ID local to the DIS.

Not listed because added by the inclusion of the shared media LAN

Shared Media LAN Example Format *B* Peculiarities

- Exact order for each chain
- Arbitrary order between chains
- Beginning of new chain is indicated by a System ID that already appears in a former chain
- Circuit ID to be used for Pseudonode
- Taking part in a chain via the shared media LAN is described by being connected to the Pseudonode



Format B
exact order
for each chain

11; End	Chain 1
33	
22, 4; Circuit	
66	
44; End	
33	Chain 2
55; End	
22, 4; Circuit	Chain 3
77	
66	Chain 4
88; End	

85 bytes

Note

- 802.1Qca is not about p2p paths
- Mixing strict and loose hops in an explicit tree makes it too complicated
- Mixing strict and loose hops in a p2p path may be not that useful
- Order is only mandatory for a loose hop, because it is related to the preceding hop
- Ordering is unnecessary if it is not allowed to mix strict and loose hops

Programming

- ***Format A***
- Easy
- PCE
  - e.g. go through the topology sequentially per bridge per port
- Bridge
  - Just include the hops to the topology

- ***Format B***
- More complex
- PCE
  - Longest possible chains to be find
  - Encode the chain as ordered list
- Bridge
  - It has to be detected when a chain begins and ends
  - Worst case: each link is an individual chain

Summary

- The original intention determines the pros and cons
  - Format A: describe a generic graph, network topology
  - Format B: describe a p2p path
- ***Format A***
- Easier to program
- Shared media LAN
  - Simple, in-line with IS-IS
- Size
  - Can be 2 bytes smaller per hop
- ***Format B***
- Easier to read by human
- Shared media LAN
  - Messy
- Size
  - 2 bytes larger in worst case (single hop chain)
- Same TLV structure can be used for the two formats