ERICSSON

Maximally Redundant Trees (MRT) for 802.1Qca

János Farkas and Gábor Enyedi

January 22, 2014

Summary

- Fast-Reroute with Maximally Redundant Trees (MRT-FRR) is being standardized, hence becoming available for IS-IS
- MRT-FRR has attractive features
  - 100% coverage for node and link failures
  - No engineering for fault coverage
  - Extremely efficient computation
- IEEE 802.1Q could leverage MRT
  - IS-IS has been already introduced
  - The architecture provides a good basis to support MRT
  - The add-ons seem to be simple
  - MRT could be great for seamless redundancy
  - P802.1Qca addresses enhancements for redundancy

Outline

MRT Background

References

[1] G. Enyedi, A. Császár, A. Atlas, C. Bowers, and A. Gopalan, "Algorithms for computing Maximally Redundant Trees for IP/LDP Fast-Reroute," October 2013, work in progress, http://tools.ietf.org/html/draft-enyedi-rtgwg-mrt-frr-algorithm-04 (it is [B-Qca-3] in P802.1Qca D0.5)

[2] A. Atlas, R. Kebler, G. Enyedi, A. Császár, J. Tantsura, M. Konstantynowicz, and R. White, "An Architecture for IP/LDP Fast-Reroute Using Maximally Redundant Trees," July 2013, work in progress, http://tools.ietf.org/html/draft-ietf-rtgwg-mrt-frr-architecture-03

[3] Z. Li, N. Wu, A. Atlas, C. Bowers, and J. Tantsura, "IS-IS Extensions for Maximally Redundant Trees," October 2013, work in progress, http://tools.ietf.org/html/draft-li-isis-mrt-00

MRT Highlights

- Fast-Reroute with Maximally Redundant Trees (MRT-FRR) is a technology that gives link-protection and node-protection with 100% coverage in any network topology that is still connected after the failure
- MRT removes all need to engineer for fault coverage
- The MRT architecture [2] relies on fully distributed path computation
  - MRT computation is extremely efficient
- A **standard MRT algorithm** is required
- The **MRT Lowpoint algorithm** is being standardized in [1] as the default algorithm for the distributed computation of Maximally Redundant Trees (MRT) = Maximally Disjoint Trees (MDT)

Note

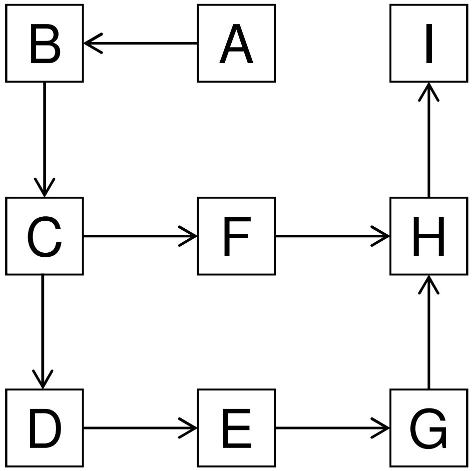
- Maximally Redundant Trees (MRT) = Maximally Disjoint Trees (MDT)
  - MRT and MDT are the same!
  - It is just two terms for the same thing
- Further terms are used in the literature for the same thing
  - colored trees
  - independent trees
  - recovery trees

Some Definitions

- **digraph**: directed graph
- **DAG**: Directed Acyclic Graph – a digraph containing no directed cycle.
- **ADAG**: Almost Directed Acyclic Graph – a digraph that can be transformed into a DAG by removing the edges directed to the Root vertex.
- **GADAG**: Generalized ADAG – a digraph, which has only ADAGs as all of its blocks, where a block is either a 2-connected cluster, a cut-edge, or an isolated vertex. The root of such a block is the vertex closest to the global root (e.g. with uniform link costs).

Directed Acyclic Graph (DAG)

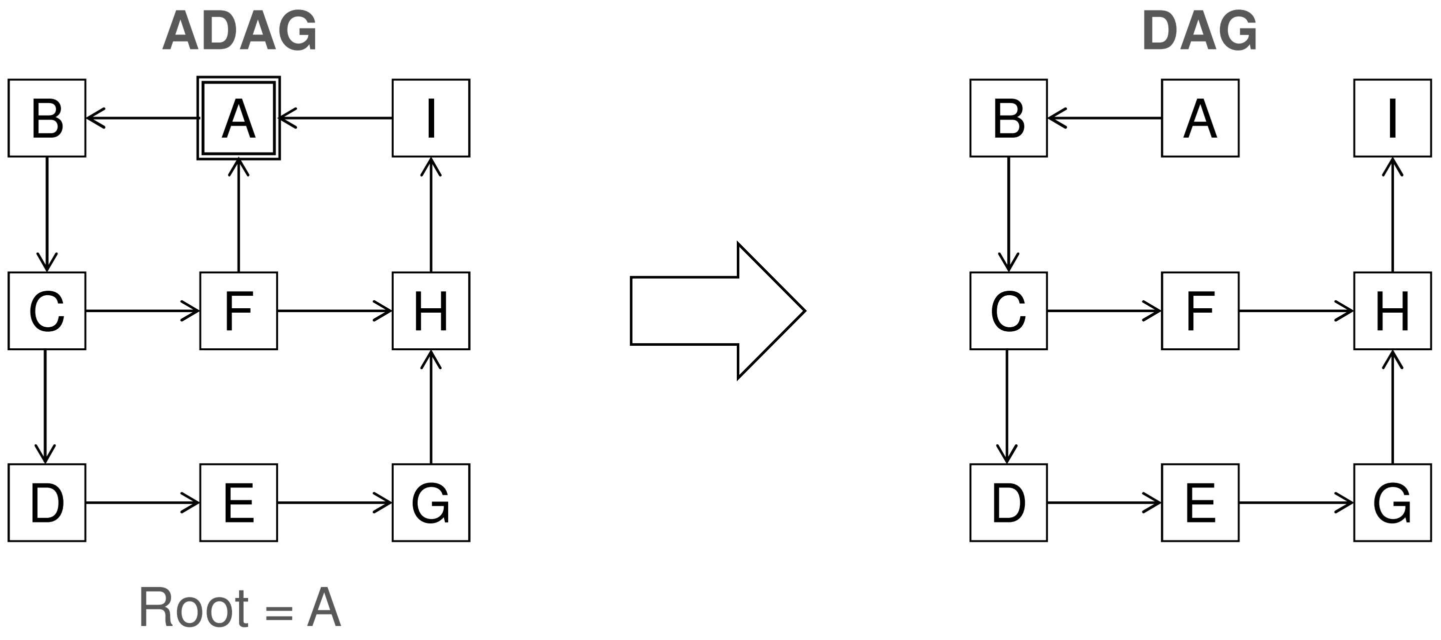
- A digraph containing no directed cycle, e.g.


$$\begin{tikzcd}
B \arrow[d] & A \arrow[l] & I \\
C \arrow[r] \arrow[d] & F \arrow[r] & H \arrow[u] \\
D \arrow[r] & E \arrow[r] & G \arrow[u]
\end{tikzcd}$$

- A DAG can represent a partial order of vertices
  - The direction of an edge in the example is according to the lexicographical order
- It is only a partial order because not necessarily all relations are captured by a DAG
  - The example does not provide any information on the relation between F ? D, F ? E, and F ? G

ADAG and GADAG

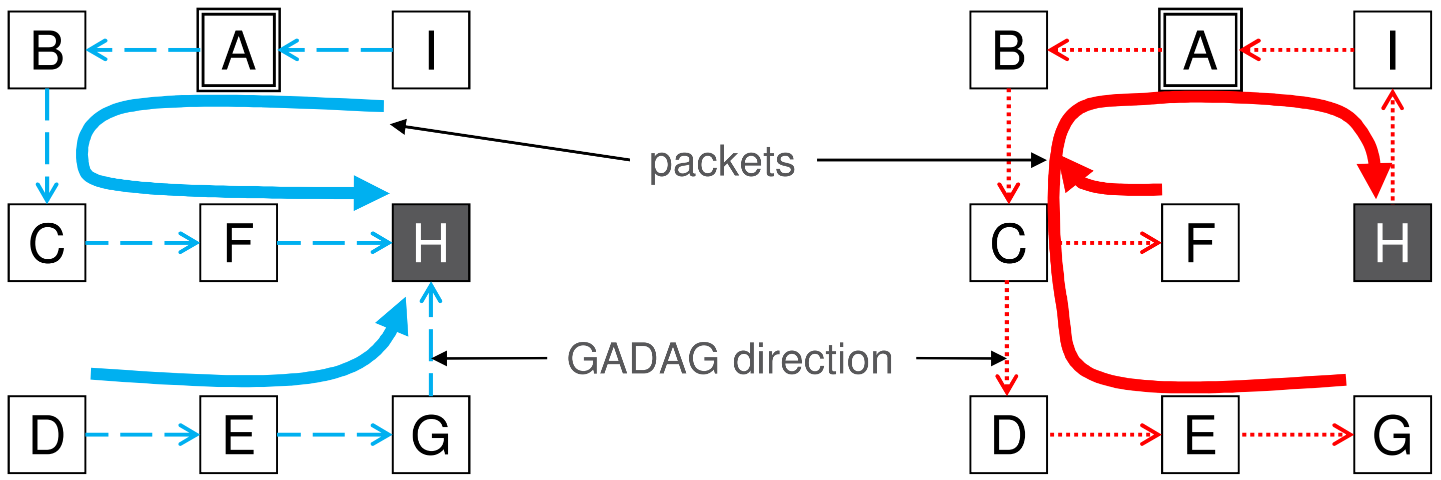
- ADAG = Almost DAG
  - We get a DAG from an ADAG if we remove the edges directed to the Root vertex, e.g.



- GADAG = Generalized ADAG for graphs that are not 2-connected

MRT-Blue and MRT-Red

- MRT-Blue: the increasing MRT, e.g.
- MRT-Red: the decreasing MRT, e.g.



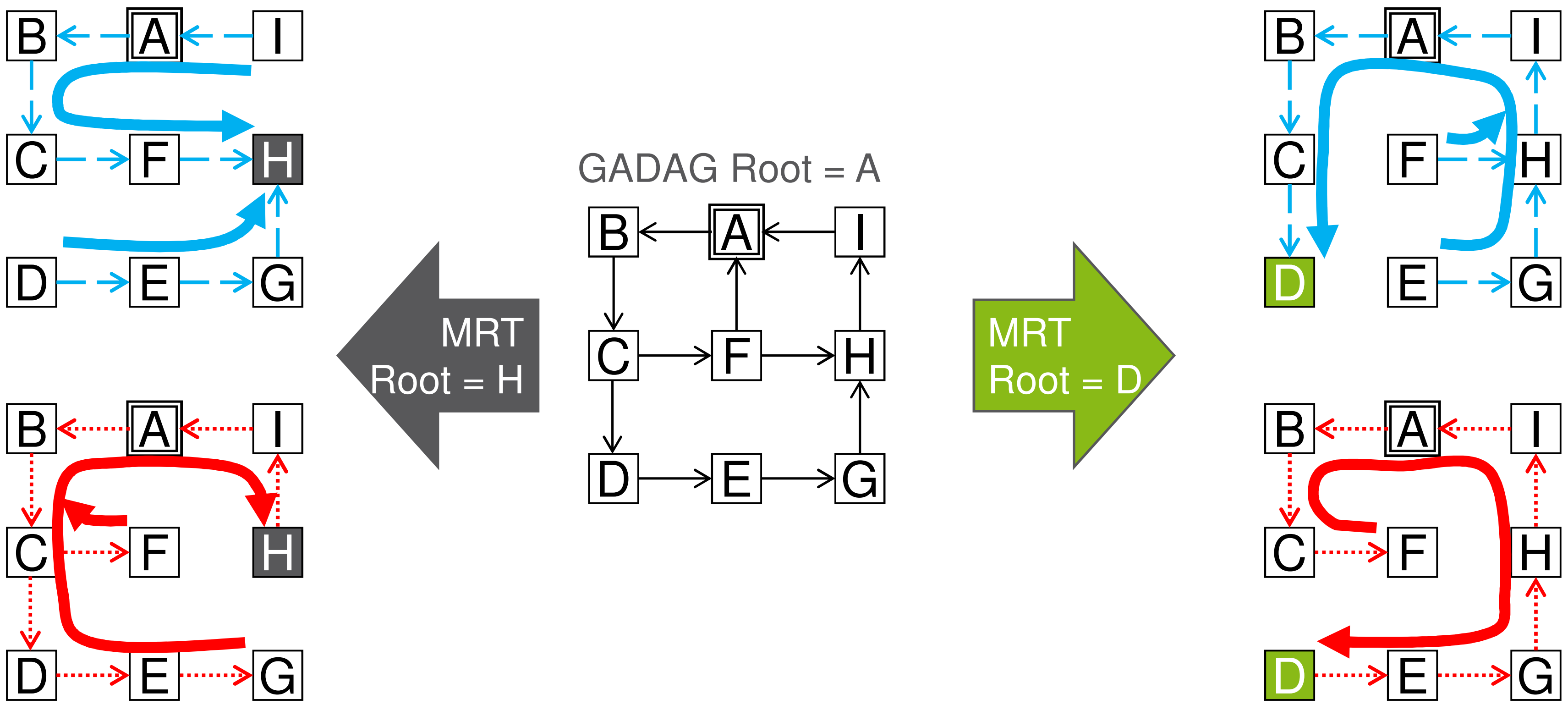
MRT Root = **H**
GADAG Root = A

MRT Root = **H**
GADAG Root = A

- Packets use the links along ascending GADAG order
- Packets use the links along descending GADAG order
- Along the different MRTs, each source uses different Port to reach the MRT Root (if topology allows)

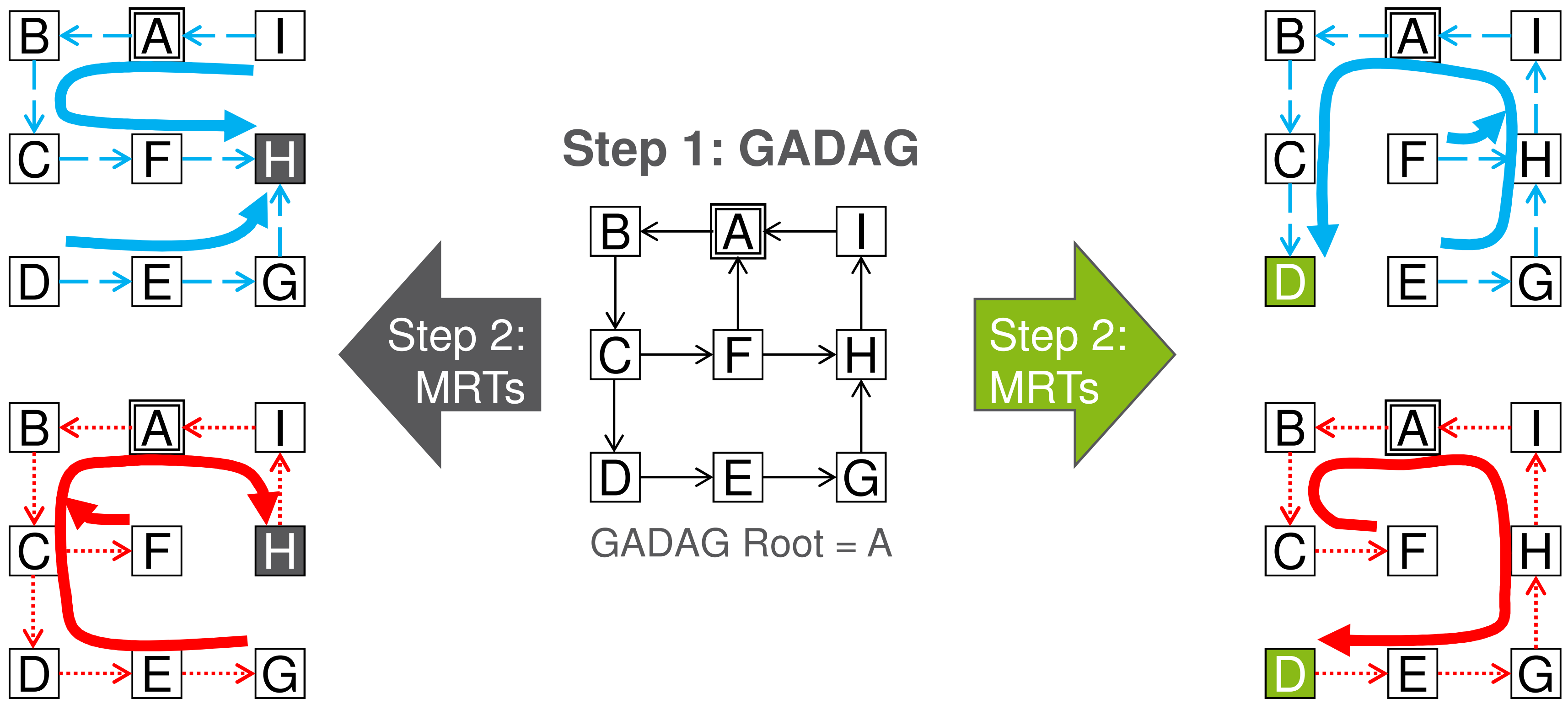
GADAG is Identical for All MRTs

- The **same GADAG** is used **for** computing **all** the **MRTs** for all the MRT Roots (within a given MRT algorithm)
- **MRT Root** and **GADAG Root** are two **different** things!



Computation

- Computation steps of an MRT Algorithm:
  1. Determine GADAG, which is common and identical for all MRTs
  2. Find the MRT pairs on the GADAG for each MRT Root



Use of MRTs

- MRTs are used for the protection of the shortest path by means of FRR local repair, see [2]
- Point of Local Repair (PLR) router sends the packet on the unaffected MRT rooted by the destination address if the shortest path link goes down
  - It is determined a priori for each link of a router which MRT (Blue or Red) to be used by the router if the given link goes down
- Packets do not get redirected from an MRT

MRT Recomputation

- Section 11.2 of [2] specifies the restoration of MRTs as follows:
  1. Receive failure notification
  2. Recompute SPT
  3. Install new SPT
  4. If the network was stable before the failure occurred, wait a configured (or advertised) period for all routers to be using their SPTs and traffic to drain from the MRTs.
  5. Recompute MRTs
  6. Install new MRTs
- Note that shortest path gets restored first, MRTs are only touched when traffic uses shortest path again

IS-IS sub-TLVs for MRT

- An MRT Profile [2] determines parameters and behavioral rules, e.g.:
  - The MRT algorithm (MRT Lowpoint for the default MRT Profile)
  - How MRT recomputation is handled
  - How area/level boundaries dealt with
- A router has to advertise its MRT capabilities
- IS-IS extensions proposed by [3]
  - **M bit**: new bit in the Multi-Topology TLV (type = 229) for indicating MRT capability
  - **MRT Profile sub-TLV** in IS-IS Router Capability TLV
  - **MRT-Ineligible Links sub-TLV** in IS-IS Router Capability TLV

ERICSSON

Leveraging MRT at L2

Basic Idea

- It would be great to leverage MRT by 802.1Qca
  - Redundancy (protection, restoration) is in scope, being addressed by subclause 45.3
- IS-IS TLVs are being specified anyways
  - M flag becomes available as e.g. the SPBV MAC address sub-TLV and the SPBM Service Identifier and Unicast Address sub-TLV are carried in the Multi Topology Capability TLV
- Let’s use VIDs for the identification of MRT-Blue and MRT-Red (instead of or in addition to Multi-Topology ID)
- Let’s define ECT Algorithm for standard MRT operations at L2
  - Let’s use the MRT Lowpoint algorithm [1] as the standard algorithm
  - Let’s specify the rest of the operation details

Possible Use of MRTs at L2

I. For the protection of the shortest path
   - This is the same usage as specified by [2], see also page 14

II. For the protection of each other
   - MRT-Blue and MRT-Red can protect each other in the fashion of seamless redundancy or 1+1 protection
     - A copy of the packet is sent on both trees
   - SPT optionally can be used for a third copy of the packet
   - Cautious restoration is required: One of the trees has to remain untouched while updating the other one

- Maximally disjoint point-to-point paths
  - Note that it is possible to use only the branches of MRT-Blue and MRT-Red between the MRT Root and a Leaf
  - Thus we get, two maximally disjoint paths

Possible Operation Modes

A. Fully distributed computation
   - Each node performs both computation steps: GADAG + MRTs

B. Centrally computed GADAG
   - Computation steps are split:
     - GADAG is only computed by a PCE
     - PCE floods the descriptor of GADAG to every network node
     - Nodes only determine the MRTs on top of the GADAG

C. Centrally computed MRTs
   - Each MRT is computed by the PCE
   - PCE floods the descriptor of the MRTs

Which Bridges are the MRT Roots?

- Default approach
  - Each SPT Root is also an MRT Root
  - MRT-Blue and MRT-Red are then set-up and maintained for all SPT Roots
- A more sophisticated approach
  - Allow to configure which bridges are MRT Roots
  - MRT-Blue and MRT-Red are then only set-up and maintained for the MRT Roots
- Both approaches are applicable to all the three modes: A, B, and C

ECT Algorithm

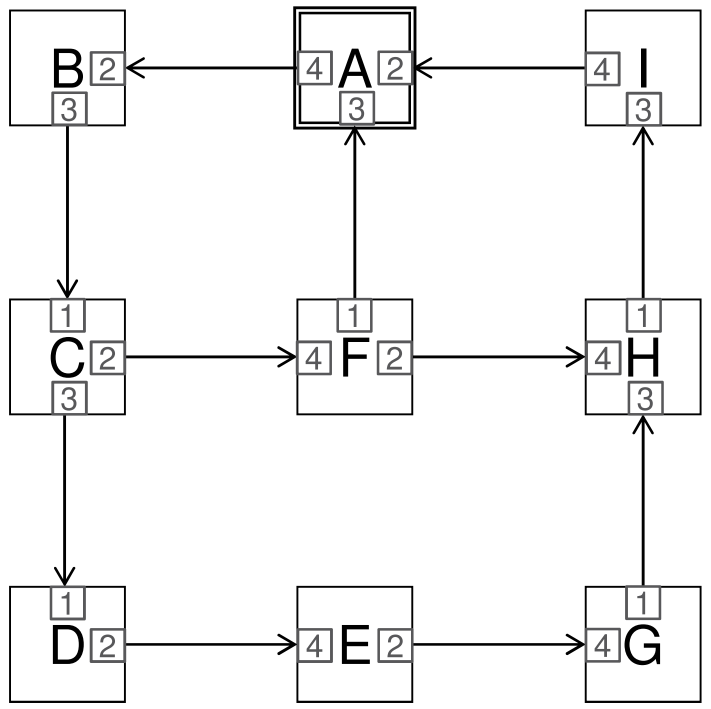
- Let's use ECT Algorithm to identify an exact operation mode
- In other words, let's use ECT Algorithm as the MRT Profile ID at L2
  - Instead of / aside the Profile ID field of the MRT Profile sub-TLV in IS-IS Router Capability TLV of [3]
  - An MRT Profile also determines the exact MRT algorithm to be used, where the MRT Lowpoint algorithm [1] is the standard algorithm for the Default Profile

GADAG

- Lets use the Bridge IDs for GADAG Root selection in all operation modes
  - Instead of the GADAG Priority field of the MRT Profile sub-TLV in IS-IS Router Capability TLV of [3]
  - Note that Bridge ID has a Priority field
- GADAG descriptor is needed for Mode B
  - GADAG descriptor is flooded by PCE in an LSP
  - Note that the GADAG descriptor is relevant for each node because it is identical thorough the Domain and the same GADAG is used for the computation of the MRTs of each MRT Root
  - GADAG is given by the list of its edges, where the first entry should refer to the GADAG Root
  - An edge of a GADAG is given by the Nodal ID identifying the lower vertex of the edge in the GADAG plus a Port ID identifying the port Connecting to the higher vertex of the GADAG edge

GADAG Description Example

- GADAG



GADAG Root = A

- GADAG Descriptor
  - (Entry Format: Nodal ID, Port ID)

A, 4
F, 1
E, 2
F, 2
I, 4
B, 3
H, 1
C, 2
D, 2
C, 3
G, 1

  - Note that the GADAG Root is the first in the list, the order to the rest is arbitrary

Cautious Restoration for MRTs

- Failure ➔ MRT-Blue or MRT-Red is used for data traffic
- Goal: do not cause traffic outage by restoration
- Therefore, at least one of the trees (out of the two or three) should be untouched while another being updated
- 3-three case: SPT + MRT-Blue + MRT-Red (the case of [2])
  - SPT is restored first,
  - After the network converges and traffic is directed back to SPT,
    - The method of [2] for checking that is: "wait a configured (or advertised) period for all routers"
  - MRT-Blue and MRT-Red are updated at the same time, see [2] and/or page 15
- Two-tree case: MRT-Blue and MRT-Red protect each other
  - Meanwhile the broken one gets restored, the other one should not be touched
  - That is, the MRT-Blue and MRT-Red should not be updated at the same time
- The current approach of [2]: "waiting long enough" did not seem the best method during the comment resolution of P802.1Qca D0.3
  - Note that a more sophisticated method has been proposed in http://www.ieee802.org/1/files/public/docs2013/ca-farkas-path-status-notification-0513-v01.pdf

Summary – same as page 2

- Fast-Reroute with Maximally Redundant Trees (MRT-FRR) is being standardized, hence becoming available for IS-IS
- MRT-FRR has attractive features
  - 100% coverage for node and link failures
  - No engineering for fault coverage
  - Extremely efficient computation
- IEEE 802.1Q could easily leverage MRT
  - IS-IS has been already introduced
  - The architecture provides a good basis to support MRT
  - The add-ons seem to be simple
  - MRT could be great for seamless redundancy
  - P802.1Qca addresses enhancements for redundancy

Questions

- Shall we add the support for MRT to 802.1Qca?
- If yes, then which operation mode?
  - Note that they do not exclude each other, it is just using different ECT-Algoritms for the different modes and leveraging almost the same add-ons