802.1Qca Tree ordered representation – recipe and attributes

**Construction**
- start at the root;
- encode an arbitrary p2p path to its leaf, recording the branch points in the order they are encountered;
- starting at the 1st branch point, encode the p2p path leading from that, starting that path segment with (a repetition of) the branch point Hop sub-TLV (1);
- recurse until entire tree has been encoded;
- assemble the p2p fragments in branch-point order:
  - to ensure the branch node has been encoded already.

**So what?**

This is simple to compute:
- either **recursively**, after the full tree has been determined;
- or **incrementally**, as a route computation (e.g. CSPF) is performed.

It can represent any structure:
- by using repeated Node Ids for loop closure, this technique can represent any topology, including but not limited to GADAG structures, and shared media LANs.
- Circuit Ids are required only when there are multiple physical links between adjacencies.

This is compact:
- precisely one basic Hop TLV per node in the tree, plus one Hop TLV per branch point:
  - structure size = \((N + \# \text{ branches}) \times 9\) bytes \((= 216\ \text{bytes for the structure above})\)

If there is no benefit from an unordered structure, what is wrong with KISS?