Interim LSR meeting

May 30, 2019
Issues discussed at Prague

- LANs in the FT: LANs can be in the FT. In centralized mode, if a LAN (pseudonode) is included, then all nodes on that pseudonode are on the FT through the LAN. In distributed mode, more selective flooding is optional.

- Temporary additions: Agreed that nodes should rate limit temporary additions to the FT. Too slow, we impact convergence. Too fast, we risk cascade failure.
Other changes since Prague

• Advertising the FT: We’ve added a bit in the Link Attributes sub-TLV (IS-IS) and added a Link Attributes TLV to OSPF. This is taken from draft-cc-Isr-flooding-reduction’s FT bit.

• Bug fix: In the Flooding Request TLV in an IIH, we listed a field as CircuitType. This was confusing. It’s a bit mask of the levels that are requesting flooding. It should be a subset of the Circuit Type.
Temporary additions vs. Backup Paths

- Problems we’re addressing:
  - If a new node is added to the topology, it is not on the Flooding Topology (FT). How does its LSP/LSA get flooded?
  - If we have multiple failures, the FT may partition. How do we recover?
Temporary addition algorithm

On each node, in parallel:
  On a topology change:
    For each adjacency:
      Is there a path to the adjacent node via the flooding topology?
      If not, add it to a set of candidates.
    While the set of candidates is not empty:
      Remove one candidate from the set.
      Temporarily add it to the flooding topology
      Delay (amount is implementation defined)
      If there has been another topology change:
        Clear the candidate set
        Restart the algorithm
Benefits of the algorithm

• Guaranteed to converge. Each iteration adds links to the FT.
  • Converges in the face of arbitrary failures, including multiple partitions.

• Does not rely on stale information. Information required is the FT and LSDB for the local partition.
  • If multiple repairs are necessary, leverages updated information as it’s learned.
  • No reliance on knowing what links failed or LSDB prior to failure.

• Computationally cheap. Partition check is $O(n)$. Evaluating adjacencies is $O(n)$.

• Right trade off: gives both stability and convergence.

• Implemented. Works in the lab.
Downsides of the algorithm

- Does not necessarily enable the minimal number of links for repair. This requires an oracle.

- Does not converge instantly. This also requires an oracle.