A Framework for Computed Multicast applied to MPLS based Segment Routing

draft-allan-spring-mpls-multicast-framework-00

Dave Allan, Jeff Tantsura; Ericsson

IETF95

What is the draft about?

- The application of computation to determining the routing of multicast segments in an MPLS based SR network, and how unicast tunnels can be used as part of multicast tree construction to minimize overall network state
- The draft describes
  - Terminology
  - Overall approach
  - Loose and Specified multicast distribution trees
  - Algorithm
  - FIB installation procedures

Approach

- The draft postulates an architecture whereby multicast trees are a hybrid of roots, leaves, and replication points interconnected with unicast tunnels, with the routing of the tree determined entirely from information in the IGP
  - Which has been augmented to add TLVs for multicast interest
- This provides multiple benefits
  - Minimized messaging the converge the network
  - Reduced dataplane state
  - Minimized bandwidth requirements
  - Unicast recovery for most failures
  - Re-use of the existing MPLS dataplane
  - Use of tunnels reduces computation requirements

Tree Generation

- The use of tunnels necessitates a minimum cost or near minimum cost tree in order to be ECMP “friendly”
  - No duplicate packets on any link
- This necessitates a unique solution per S,G tree

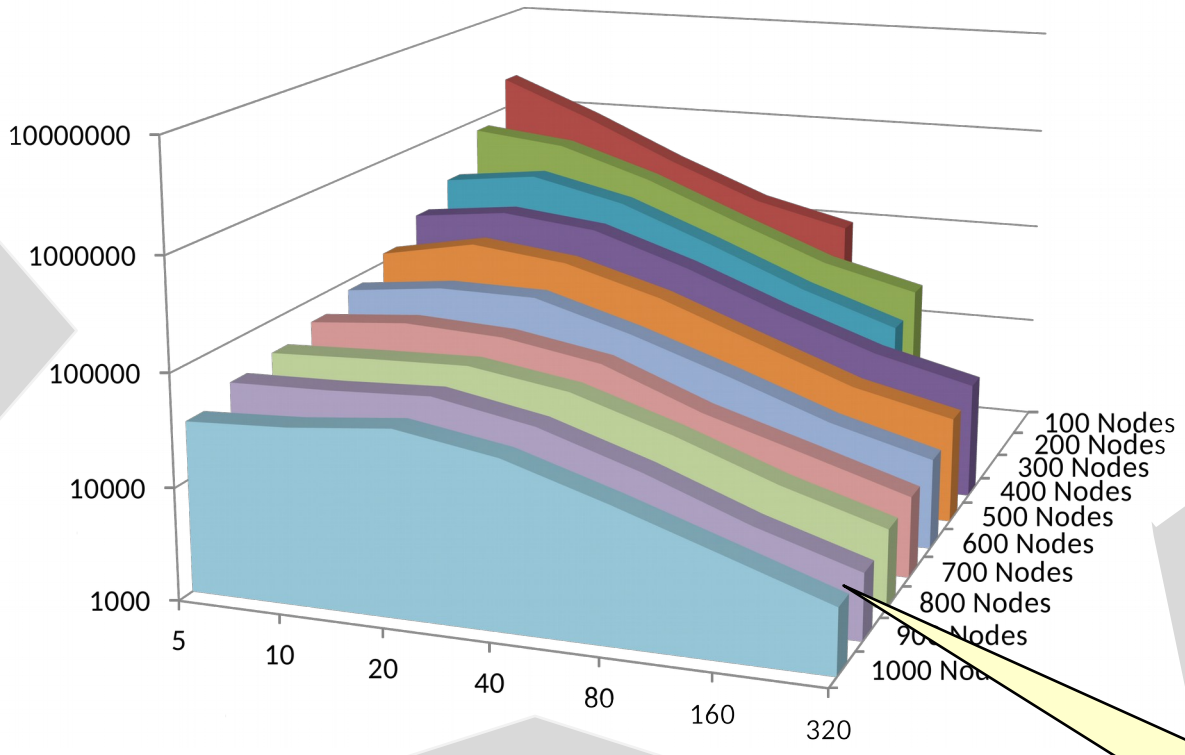
Tree Pruning

- Computed trees are determined by a series of pruning steps applied to the shortest path tree from a root to the set of leaves
- Two classes of prune
  1. Those that if they fully resolve the tree are known to produce a minimum cost tree
     - This will sort out 97%+ of leaves
  2. If these do not completely resolve the tree (unique shortest path to the root from every leaf), then we start to apply “guesses”, and audit the tree for correctness at the end
     - With good “guesses” only a tiny fraction of the remainder require “fixing”

Yes…

- This is computationally expensive
- But the use of tunnels means most nodes will not install state for a given MDT
    - And they can figure this out early in the process
- So not only do they not have to fully resolve every S,G tree, but the amount of state to be generated and synchronized with the FIB is minimized

Example results



Tests performed on a 1.8Ghz i5CPU

Loose and Specified Trees

- A loose tree is composed of a single multicast segment (with a SID), where only the root and the leaves have been specified in the IGP
  - The routing of the tree is wholly computed based on the current network topology
- A specified tree is composed of a concatenation of multicast segments where the roots, waypoints and leaves have been specified in the IGP
  - The routing of individual segments is still computed
  - The routing of an MDT can then be specified to an arbitrary level of granularity
  - A unique SID per segment ensures the resulting hybrid of pinned and computed components is loop free, even if not planar

Next Steps

- We have framework draft 00
  - Collect feedback
- We will bring forth in future drafts:
  - The required IGP extensions
  - Interworking with existing mechanisms
- We will pursue standards track

Questions?