Network Slicing with FlexTE

draft-zzhang-teas-network-slicing-with-flex-te

Z. Zhang, S. Hedge, A. Gulko

Presented by Z. Zhang for TEAS, IETF108
Problem Statement

• Flexible Algorithm (FlexAlgo) is a flexible and lightweight way of implementing multi-topology and network slicing

• What if you have many routers and need many network slices?
  • E.g. 5G Transport
  • FlexAlgo only scales to up to tens of algorithms

• Almighty controllers come to the rescue
  • Routers don’t do per-algo SPFs
  • Controllers calculate SR-TE paths for ingress routers

• This may still not be enough
  • Controllers could be overwhelmed given the scale
    • Keep in mind that calculations may also be needed for many many multicast trees ...
Solution – Flexible TE

• Offload per-algo SR-TE Path calculation to edge routers
  • Traffic steering based on Adjacency SID list
  • Internal routers not aware of Flexible Algorithm Definition (FAD) or Link Administrative Group (LAG) information
  • Edge routers only aware of FADs and LAGs that they care about

• Idea comes from draft-drake-bess-enhanced-vpn
  • Applied to FlexAlgo with very efficient southbound BGP-LS distribution of FADs and LAGs to targeted routers
Targeted Distribution of LAGs

• Controllers provisioned with LAGs for all links
• Controllers originate Link NLRIs and distribute via southbound BGP-LS
  • Instead of advertising LAG bitmask as a TLV in BGP-LS Attribute, advertising it as new Bitmask Route Target
• Bitmask Route Target (RT)
  • Two Bitmask RTs match if the logical AND of the two bitmasks is none-zero
  • An edge router is configured with a local Bitmask RT with the bits set for the LAGs that it cares about
    • So only Link NLRIs that an edge router cares about will be propagated towards and imported by it
Targeted Distribution of FADs

• A FAD NLRI is added to BGP-LS
  • mirroring ISIS FAD sub-TLV
  • with a Bitmask RT to specify the LAGs for links that the FAD includes/excludes

• The same Bitmask RT configured on an edge router to import Link NLRIs also used for importing FAD NLRIs

• Example
  • For any FAD that includes/excludes red links, its FAD NLRI has a Bitmask RT with the bit for red link set
  • For any red link, the LINK NLRI has a Bitmask RT with the bit for red link set
  • For any router that cares about those FADs and red links, its local Bitmask RT has the bit for red link set
  • This gets those FADs and red link's NLRIs propagated to and imported by those routers
Targeted Distribution to Internal Routers

- The SR-TE Path may be too long to encode in packet header
- Some internal routers may be involved
  - Learn FAD/LAG information and do per-algo SR-TE path calculation
    - Just like an edge router
  - Advertise per-algo Binding SIDs for edge routers to use
Controller Signaling of FAD/LAG with FlexAlgo

• Even with plain old FlexAlgo, controller signaling of FAD/LAG to all routers can be used
  • Either have all routers running BGP-LS, or have BGP-LS routers re-flood via IGP
  • Instead of provisioning FAD/LAG on individual routers and then flood
• This allows centralized provisioning/signaling after centralized planning
  • Distributed provisioning/signaling starts with centralized planning anyway
Summary

• Centralized provisioning and signaling of FAD/LAG
  • For ease of provisioning and management

• Targeted distribution of FAD/LAG plus SR-TE path calculation by edge and selected internal routers
  • To offload controllers from overwhelming SR-TE path calculations
  • To relieve other routers from per-algo SPF
  • To limit FAD/LAG information to relevant routers only

• Limitations
  • Link protection is not strictly per-algorithm
  • SR-TE path length issue leads to per-algo Binding SIDs on selected routers