AUTO EVPN
IETF 110
DRAFT-HEAD-RIFT-AUTO-EVPN
JORDAN HEAD, Ed
WEN LIN
TONY PRZYGIEienda
AGENDA

• DRIVERS & VISION
• ARCHITECTURE
• FIRST DETAILS
• NEXT STEPS
DRIVERS & VISION

- **Ethernet L2 Switches Require Exactly “No Configuration at All” to Be Plugged In and Work**

- **Compare to EVPN/VxLAN Today Which Is Basically Aiming to Replace L2 As Technology**
  - Either Take a Magic Controller (May Be Best Solution in Seriously Large Deployments)
    - How Do the Boxes Find It?
    - How Does it Know Which Box is What?
  - Or
    - **Configure L3 IP Underlay**
      - Addressing (In Case of V6 LLC Lessens the Load a Bit)
      - Underlay Protocol
        - IDs
        - Peers (IGP Less So, BGP with More Special Hacks Also Less)
        - BFD?
    - **Debug Underlay**
  - **And Then**
    - **Configure L3 IP Overlay**
      - Bring Up RRs
        - Cluster IDs
        - Loopbacks

- **Bring Up Leaves**
  - Cluster IDs
  - Loopbacks for VxLAN
  - Loopbacks for IRB (If You Want Get Out the LAN)
  - Route Distinguishers For Type-2 and Type-5
  - Route Targets
  - VLANs
  - VNIs
  - Configure CE Interfaces
  - How Much Else Did I Forget?

- **Debug Overlay**

- Where Is the L2 Simplicity, Can We Have It Back?
ARCHITECTURE

• Key Observations
  – RIFT Basically Provides a Configuration Free Underlay That Boots Up Like L2
  – RIFT Has Enough Topology Understanding to Figure Out Leaves & RR Position for EVPN
  – RIFT Provides to All Nodes Enough Topology Information To Run All Elections/Value Computation in a Fully Distributed Fashion

• Reference Model Can Be Automatically Provisioned Without Any Further Configuration
  – Works on Fabrics with Different PoD Heights and Fanouts
  – A Fabric ID (Optionally Provided) Can Be Used to Derive Different Values
  – Both Stretched and Non-Stretched LANs Can Be Derived

![Diagram of RIFT architecture with leaf devices, super spine as RR, and iBGP transit spine.](image_url)
SOME FIRST DETAILS

- Example of Variables on All Leaves and RRs

```javascript
cluster_id: fn2clusterid(fid) as _,
    v6_loopback: v6octets(sidfid2loopback(
        v6pref, fid, sid)),
    type5_v6_loopback: v6octets(sidfid2loopback(
        v6tpref, fid, sid)),
    type5_v4_loopback: sidfid2v4loopback(fid, sid),
    bgp_router_id: sidfid2bgpid(fid, sid) as _,
    autonomous_system: fid2private_AS(fid) as _,
```

- BGP Router ID

```javascript
let sidfid2bgpid : fn(_, _) = |fid: FabricIDType, sid: UnsignedSystemID| {
    assert!(fid != 0);
    let hs: u32 = ((sid & 0xffff_ffff_0000_0000) >> 32) as _;
    let mut ls: u32 = (sid & 0xffff_ffff) as _;
    ls = ls.rotate_right(7) ^ (fid as u32).rotate_right(13);
    max( 1, v1: hs, v2: ls ) // never a 0
};

let fid2private_AS : fn(i16) → u32 = |fid: FabricIDType| {
    assert!(fid != NO_FABRIC_ID);
    // range 64496 - 65534
    const DIFF: u32 = 65534 - 64496;
    64496 + ((fid as u32) << 3) % DIFF
};

let fid2clusterid : fn(i16) → u32 = fid2private_AS;
```

- Decoding Further
RIFT EXTENSIONS

• Minimal Optional AUTO EVPN Version Carried in LIEs and Node TIEs to Prevent Forming of Adjacencies on Mismatched Versions

• Multi-Plane Support Will Need a New TIE on ToFs Only Providing the System IDs of All Routers in Same Plane
  - Necessary for RR Election Across Multiple Planes

• Optional Well-Known Northbound KV Values for a Minimal AUTO EVPN Health Check At ToF
NEXT STEPS

• Next Revision of the Draft Will
  – Fill In the Algorithms
  – Fill In the type-5 IRB Variables and Algorithms

• Additional Draft (or Subsection) for KV Fabric Status

• Co-Authorship Welcome

• With Open-Source Minimal Leaf Only Implementation of RIFT a Node Could Easily Participate in AUTO EVPN as Leaf