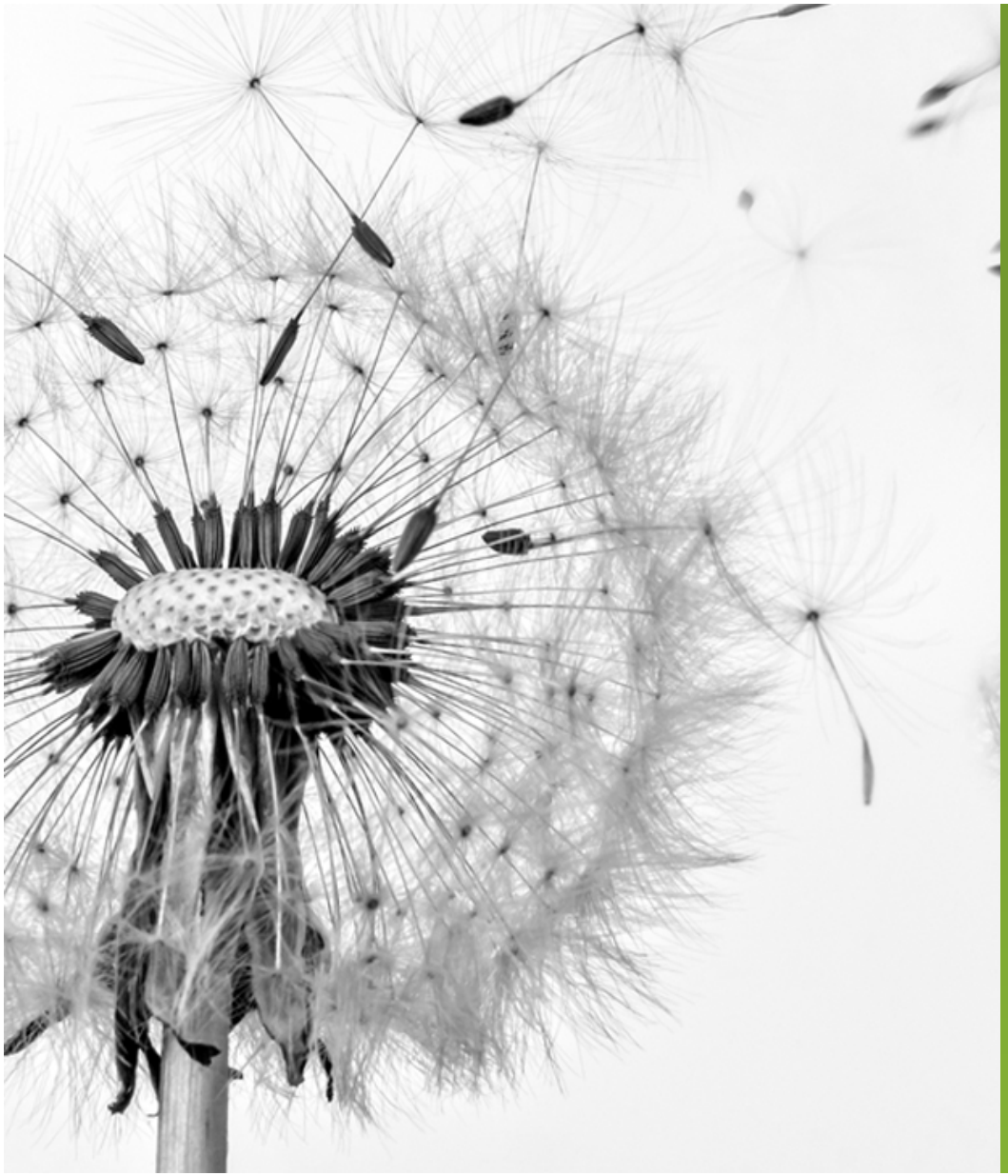
AUTO EVPN

IETF 110

DRAFT-HEAD-RIFT-AUTO-EVPN

JORDAN HEAD, ED
WEN LIN
TONY PRZYGIENDA

JUNIPER NETWORKS | Engineering Simplicity



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

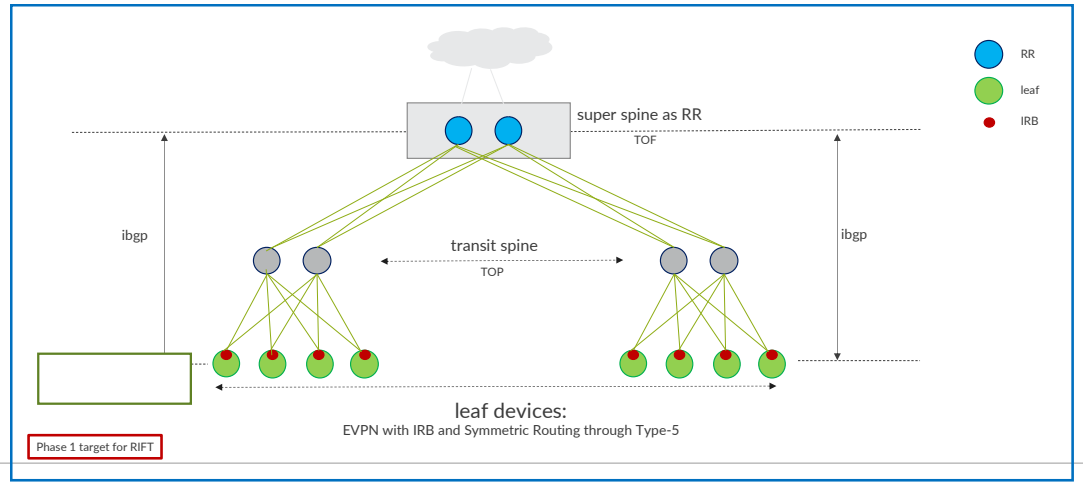
  And Then

  - 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 Forgot?
    - 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



SOME FIRST DETAILS

- Example of Variables on All Leaves and RRs

```
cluster_id: fid2clusterid(fid) as _,
v6_loopback: v62octets(sidfid2loopback(
    v6pref, fid, sid)?),
type5_v6_loopback:  v62octets(sidfid2loopback(
    v6t5pref, fid, sid)?),
type5_v4_loopback: sidfid2v4loopback(fid, sid),
bgp_router_id: sidfid2bgpid(fid, sid) as _,
autonomous_system: fid2private_AS(fid) as _,
```

- BGP Router ID

```
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( v1: 1,  v2: hs ^ ls) // never a 0
};
```

- Decoding Further

```
let fid2private_AS : fn(il6) → 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(il6) → u32 = fid2private_AS;
```

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