RIFT: Zero OPEX Routing Protocol for IP Fabrics

Tony Przygienda prz@juniper.net

Draft-Przygienda-Rift @ IETF
<table>
<thead>
<tr>
<th>Problem / Attempted Solution</th>
<th>Vs. draft-dt-rtgwg-dcrouting-requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>01. As Close to Zero Necessary Configuration as Possible (Contradicts 02)</td>
<td>✓ (next revision)</td>
</tr>
<tr>
<td>02. Peer Discovery/Automatic Forming of Trees/Preventing Cabling Violations (Contradicts 01)</td>
<td>+</td>
</tr>
<tr>
<td>03. Minimal Amount of Routes/Information on ToRs</td>
<td>+</td>
</tr>
<tr>
<td>04. High Degree of ECMP (BGP needs lots knobs, memory, own-AS-path violations) and ideally NEC and LFA</td>
<td>✓</td>
</tr>
<tr>
<td>05. Traffic Engineering by Next-Hops, Prefix Modifications</td>
<td>✓</td>
</tr>
<tr>
<td>06. See All Links in Topology to Support PCE/SR</td>
<td>✓</td>
</tr>
<tr>
<td>07. Carry Opaque Configuration Data (Key-Value) Efficiently</td>
<td>✓</td>
</tr>
<tr>
<td>08. Take a Node out of Production Quickly and Without Disruption</td>
<td>✓ (do we need GR?)</td>
</tr>
<tr>
<td>09. Automatic Disaggregation on Failures to Prevent Black-Holing and Back-Hauling</td>
<td>+</td>
</tr>
<tr>
<td>10. Minimal Blast Radius on Failures (On Failure Smallest Possible Part of the Network “Shakes”)</td>
<td>+</td>
</tr>
<tr>
<td>11. Fastest Possible Convergence on Failures</td>
<td>✓</td>
</tr>
</tbody>
</table>
RIFT: Zero OPEX Routing Algorithm for Clos Underlay

- General Concept
- Automatic Disaggregation
- Optional Horizontal Links
- And More Beyond That

But it’s so new ...

“Man cannot discover new oceans unless he has the courage to lose sight of the shore.” — Andre Gide

Well, you must be ...

“The reasonable man adapts himself to the world: the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man.” — Bernard Shaw
LINK-STATE UP, DISTANCE VECTOR DOWN & BOUNCE

1. Topological Sort

2. Flooding North

3. Distance Vector Down

4. One layer "bounce"
AUTOMATIC DISAGGREGATION

- Remember: South representation of the red spines is reflected by the green layer.
- Lower red spine sees that upper node has no adjacency to the only available next-hop to P1.
- Lower red node disaggregates P1.
**Optional Horizontal Links for Failure Protection**

- Levels can install optional horizontal links.
- Level 0 is special:
  - Leaf-2-Leaf connection that cannot be used except for Leaf-2-Leaf traffic.
- Level > 0 uses horizontal links for failure protection only:
  - Single Node Protection: Node that lost Northbound links but has neighbors that can reach higher layers uses the horizontal link.
  - N:N-1 Protection: Full mesh in a level can provide up to N-2 Northbound protection.
  - Horizontal Disaggregation can heal complex failures (not different from SouthBound Disaggregation).
RIFT DOES ON TOP

- **Automatic Flood Reduction**
- **Leaf-to-Leaf Bi-Directional Shortcuts**
- **Possible Traffic Engineering via “Flooded DV Overlay” With Policies**
- **Completely Model Based Packet Formats**
- **Channel Agnostic Delivery, Could Be Quick, TCP, UDP**
- **Prefixes to Topology Element Mapping Based on Hash Functions Local to Each Node**
  - One Extreme Point is Prefix Per Flooded Element = BGP Update
- **Purging (Given Complexity) is Omitted**
- **Policy Controlled Key-Value Store Support**
SUMMARY OF RIFT ADVANTAGES COMPARED TO DV OR LS MODIFICATIONS

▪ **ADVANTAGES OF BOTH LINK-STATE AND DISTANCE VECTOR**
  ▪ Fastest Possible Convergence
  ▪ Automatic Detection of Topology
  ▪ Minimal Routes on TORs
  ▪ High Degree of ECMP
  ▪ Fast De-commissioning of Nodes
  ▪ Maximum Propagation Speed with Flexible # Prefixes in an Update

▪ **NO DISADVANTAGES OF NEITHER LINK-STATE NOR DISTANCE VECTOR**
  ▪ Reduced Flooding
  ▪ Automatic Neighbor Detection

▪ **UNIQUE RIFT ADVANTAGES**
  ▪ Automatic Disaggregation on Failures
  ▪ Key-Value Store
  ▪ Horizontal Links Used for Protection
  ▪ Minimal Blast Radius on Failures
  ▪ Can Utilize All Paths Through Fabric Without Looping
Is there more than bits over PowerPoint?

- Yes ;-)
THANK YOU FOR YOUR ATTENTION