RIFT: Strawman Proposal of a Novel DC Fabric Routing Protocol

Key Customers' Deck

draft-przygienda-rift @ IETF

Disclaimers and Expectations

- This is a "Working Straw-Man Proposal" with Serious Amount of High-Performance Pre-Production Code in Place
- Standard Boilerplate: None of those Things Constitute Commitments to Product Specifications, Offerings or Release Dates by Juniper at this Point in Time

Agenda

- Blitz Overview of Today's Routing (if Needed)
- "Fabric Routing" is a Specialized Problem
- RIFT: a Novel Routing Algorithm for Clos Underlay

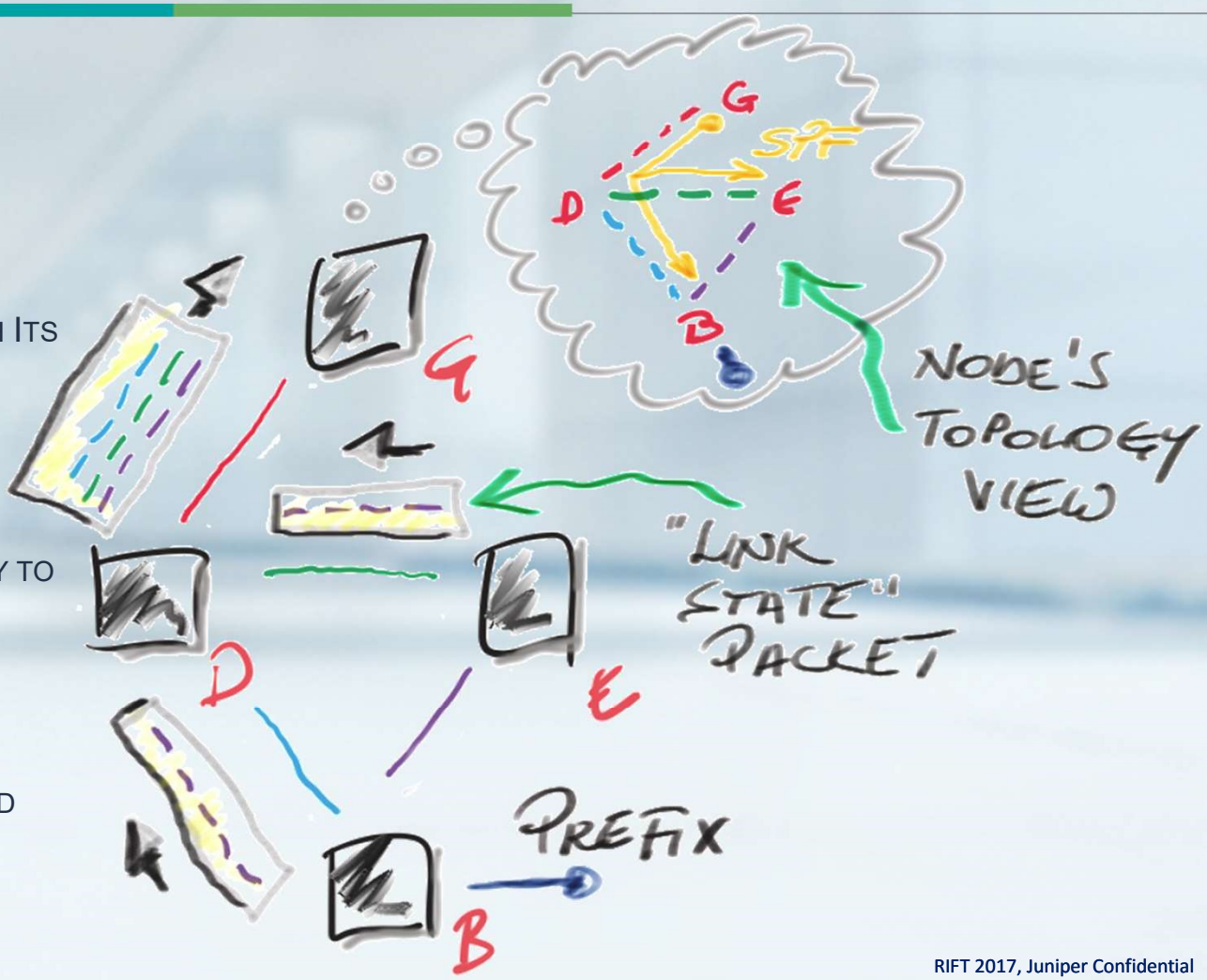
- First Performance Indicators

Blitz Overview of Today's Routing

- Link State & SPF
- Distance/Path Vector

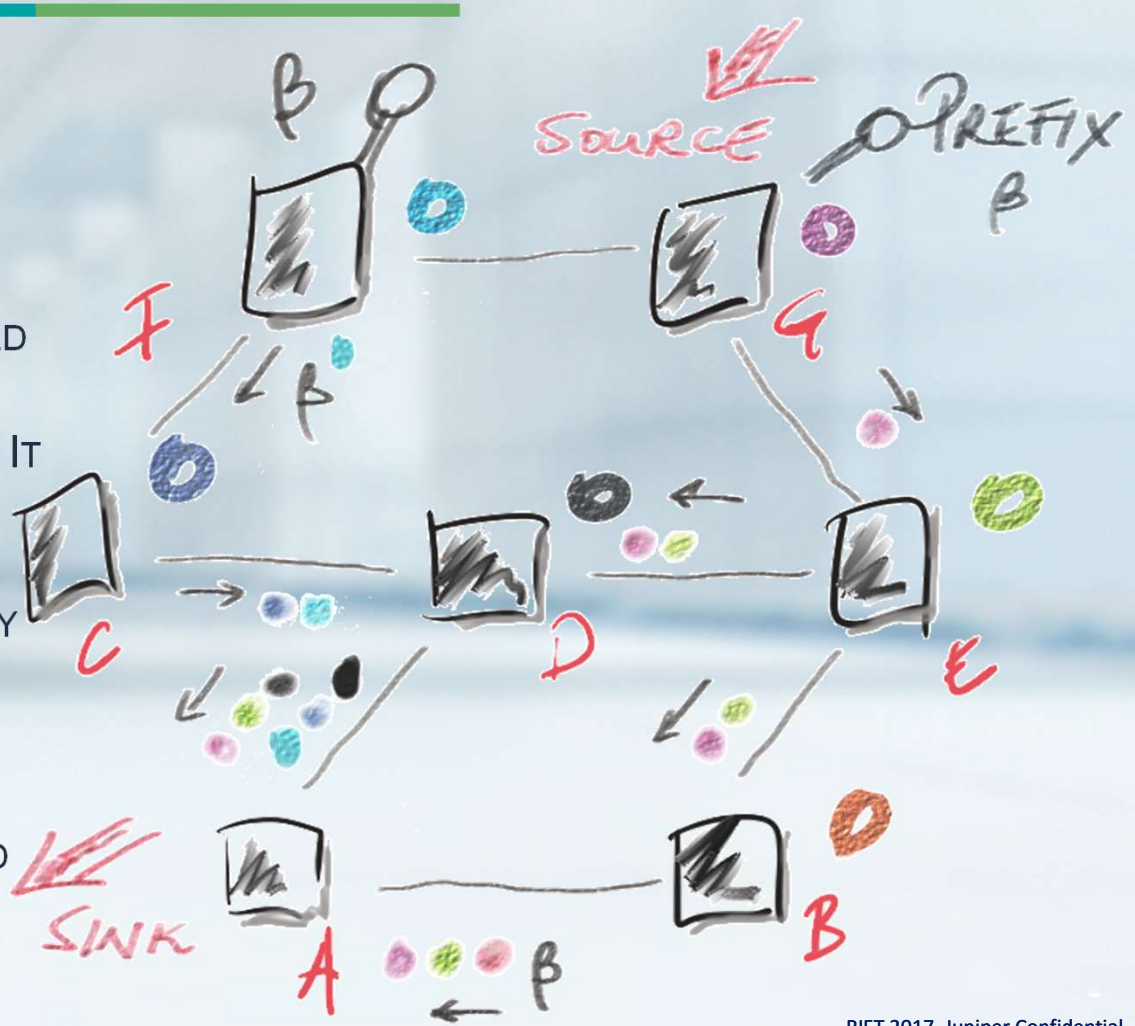
Link State and SPF = *Distributed* Computation

- Topology Elements
  - Nodes
  - Links
  - Prefixes
- Each Node Originates Packets with Its Elements
- Packets are ”Flooded”
- ”Newest” Version Wins
- Each Node “Sees” Whole Topology
- Each Node “Computes” Reachability to Everywhere
- Conversion is Very Fast
- Every Link Failure Shakes Whole Network (modulo Areas)
- Flooding Generates Excessive Load for Large Average Connectivity
- Periodic Refreshes (not Strictly Necessary)



Distance/Path Vector = *Diffused* Computation

- Prefixes "Gather" Metric When Passed Along Links
- Each Sink Computes "Best" Result And Passes It On (Add-Path Changed That)
- A Sink Keeps All Copies, Otherwise It Would Have to Trigger "Re-Diffusion"
- Loop Prevention is Easy on Strictly Uniformly Increasing Metric
- Ideal for "Policy" Rather Than "Reachability"
- Scales When Properly Implemented to Much Higher # of Routes Than Link-State

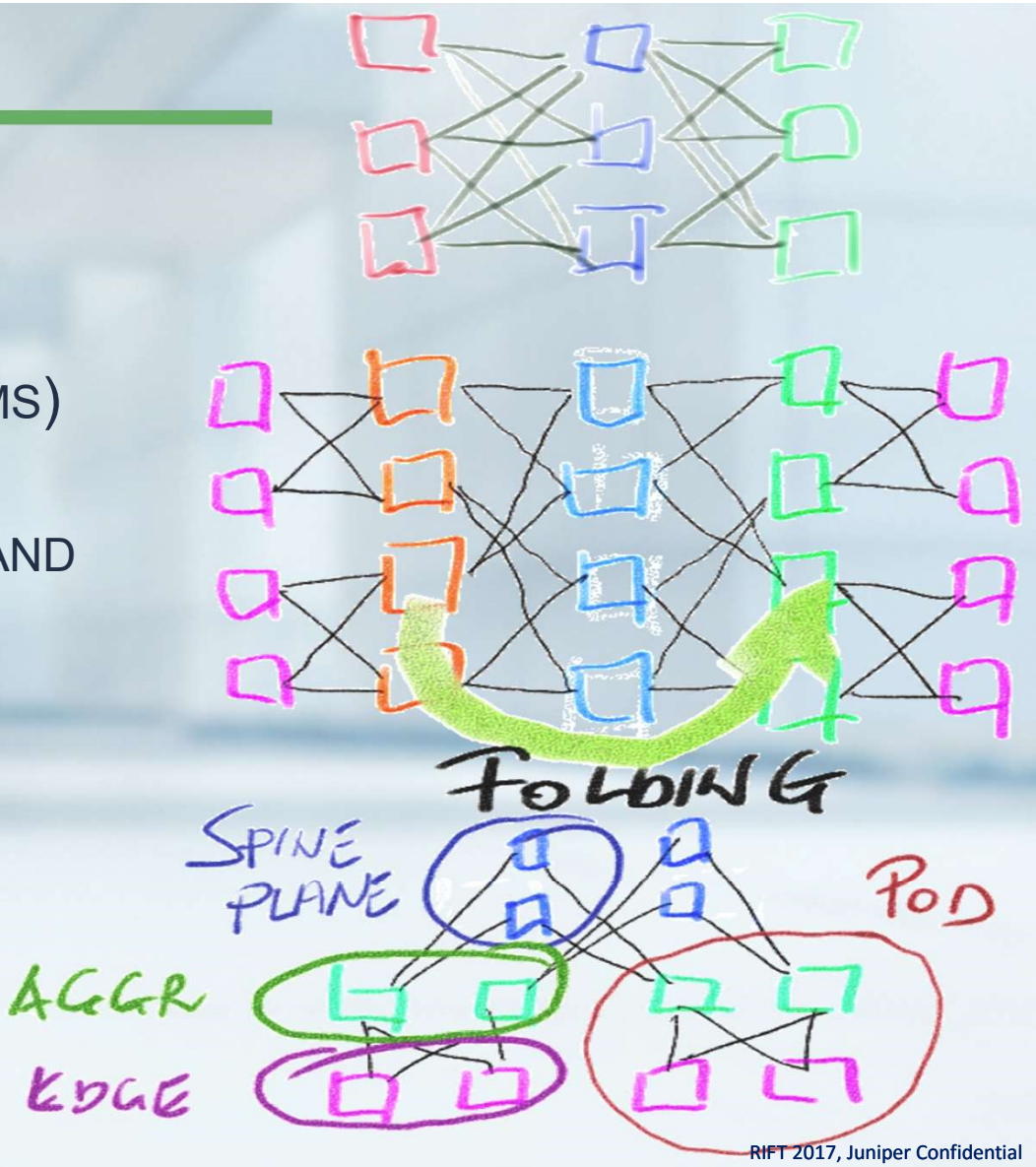


DC Fabric Routing: a Specialized Problem

- Clos Topologies Are Dominant Today
  - Toroidal [and Diagonal] Meshes have Long Paths, Small Bisection Width and Poor Blocking Properties
  - Dragonfly is very Novel and Unproven
    - ½ Throughput of CLOS at Same Cost Due to Low ECMP
    - RIFT will Work Well in a Practical Modification (One Level CLOS and Dragonfly Core)
- Current State of Affairs
- Requirements Matrix

Clos Topologies

- Clos Offers Well-Understood Blocking Probabilities
- Work Done at AT&T (Bell Systems) in 1950s
- Fully Connected Clos is Dense and Expensive
- Data Centers Today Tend to Be Variations of "Folded Fat-Tree"
  - Input Stages = Output Stages
  - Clos is "Partial"
  - Links get "Fatter" Up the Tree



Current State of Affairs

- Several of Large DC Fabrics Use E-BGP with Band-Aids as De-Facto IGP (RFC7938)
  - Numbering Schemes to Control "Path Hunting"
    - "Looping Paths" (allow-own-as Under AS Private Numbering)
    - "Relaxed Multi-Path ECMP" Since ECMP over Different AS in EBGP Does Not Work Normally
  - Add Paths to Support Multi-Homing, N-ECMP, Prevent Oscillations
  - Efforts to Get Around 65K ASes and Limited Private AS Space
  - Proprietary Provisioning and Configuration Solutions, LLDP Extensions
  - "Violations" of FSM Like Restart Timers and Minimum-Route-Advertisement Timers
  - Emerging Work for "Peer Auto-Discovery" and "SPF" Diametrically Opposite to BGP Design Principles
  - Reliance on "Update Groups" ~ Peer Groups to Prevent Withdrawal and Path Hunting After Server Link Failures
- Others Run IGP (ISIS)
- Yet Others Run BGP over IGP (Traditional Routing Architecture)
- Less Than More Successful Attempts @ Prefix Summarization, Micro- and Black-Holing

Requirements Breakdown (RFC7938+) For a "Minimal OPEX Fabric"

Problem / Attempted Solution	BGP modified for DC (all kind of "mods")	ISIS modified for DC (RFC7356 + "mods")	RIFT Native DC
Peer Discovery/Automatic Forming of Trees/Preventing Cabling Violations	⚠	⚠	✓
Minimal Amount of Routes/Information on ToRs	✗	✗	✓
High Degree of ECMP (BGP needs lots knobs, memory, own-AS-path violations) and ideally NEC and LFA	⚠	✓	✓
Traffic Engineering by Next-Hops, Prefix Modifications	✓	✗	✓
See All Links in Topology to Support PCE/SR	⚠	✓	✓
Carry Opaque Configuration Data (Key-Value) Efficiently	✗	⚠	✓
Take a Node out of Production Quickly and Without Disruption	✗	✓	✓
Automatic Disaggregation on Failures to Prevent Black-Holing and Back-Hauling	✗	✗	✓
Minimal Blast Radius on Failures (On Failure Smallest Possible Part of the Network "Shakes")	✗	✗	✓
Fastest Possible Convergence on Failures	✗	✓	✓
Simplest Initial Implementation	✓	✗	✗

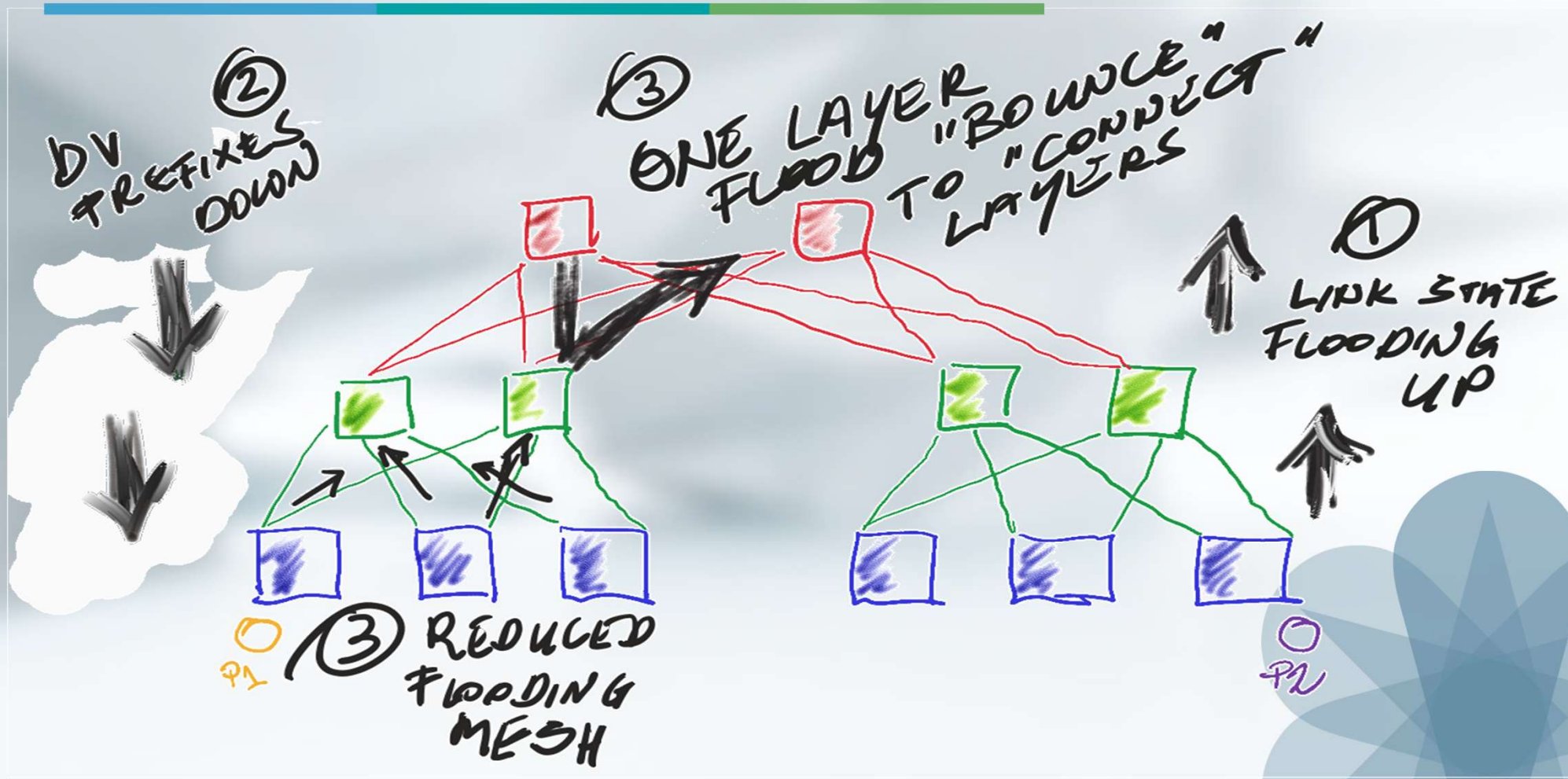
RIFT: Novel Routing Algorithm for Clos Underlay

- General Concept
- Automatic Topology Constraints
- Automatic Disaggregation
- Automatic Flooding Reduction
- And More

"Just because the standard provides a cliff in front of you, you are not necessarily required to jump off it."

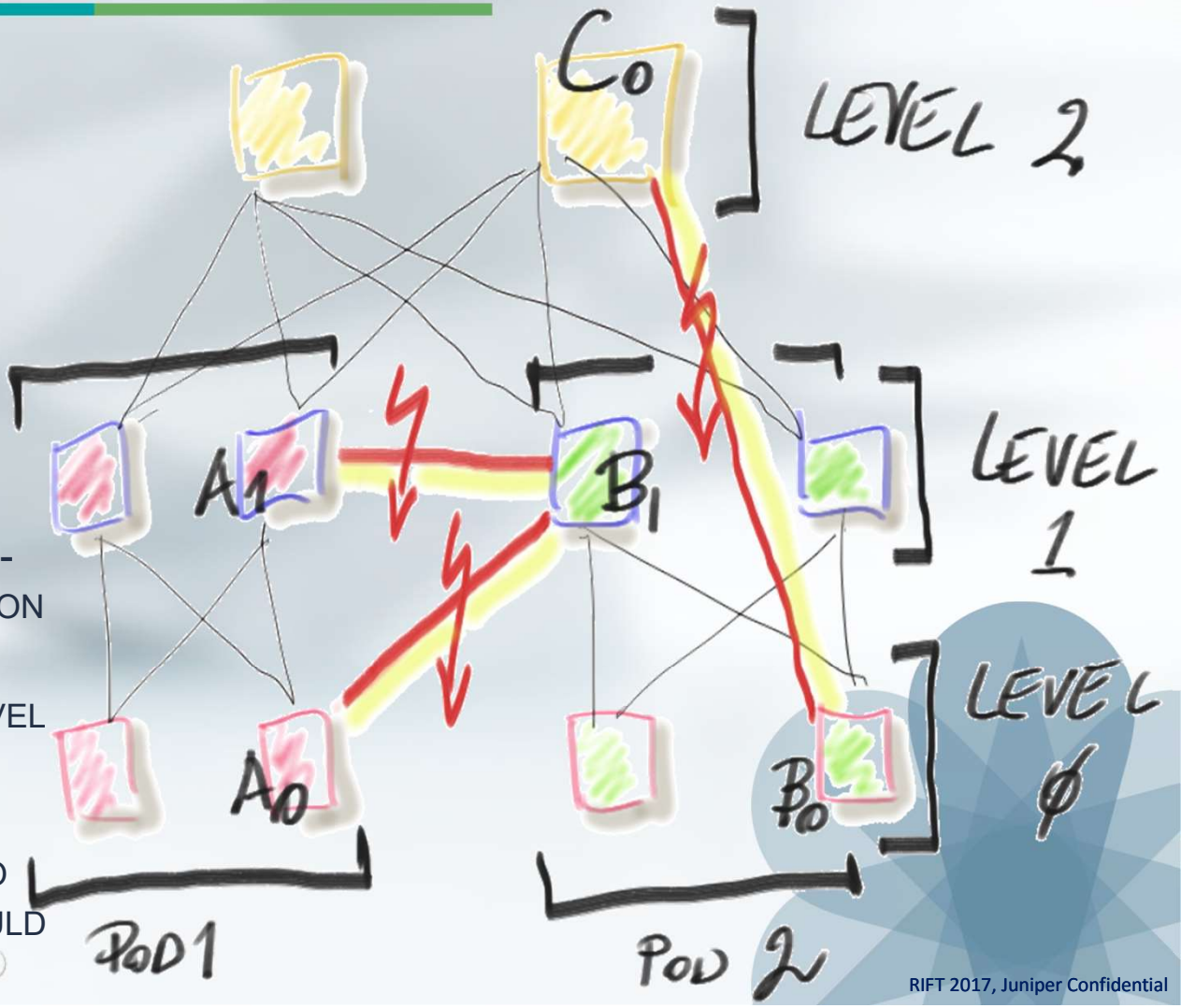
— Norman Diamond

Link-State Up, Distance Vector Down & Bounce

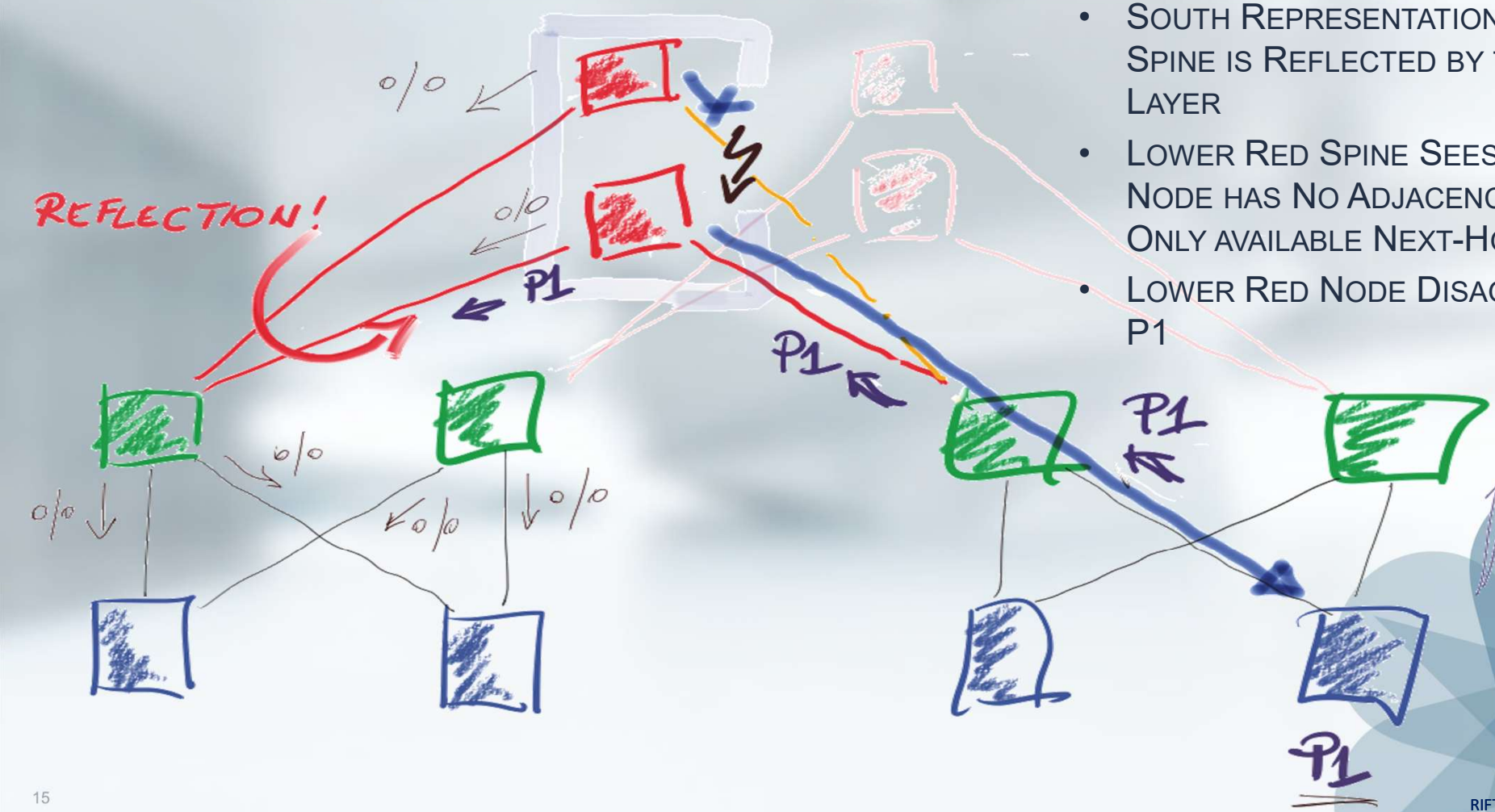


Automatic Topology Constraints

- Level 0 = Leaf
- Pod 0 = Any Pod
- Automatic Rejection of Adjacencies Based on Minimum Configuration
- A1 to B1 Forbidden Due to POD Mismatch
- A0 to B1 Forbidden Due to POD Mismatch (A0 Already Formed A0-A1 Even If POD Not Configured on A0)
- B0 to C0 Forbidden Based on Level Mismatch
- Protocol Will Work as Well If Level 0 is Allowed to Connect to Level 2 but Optimal Routing Would Need PGPs



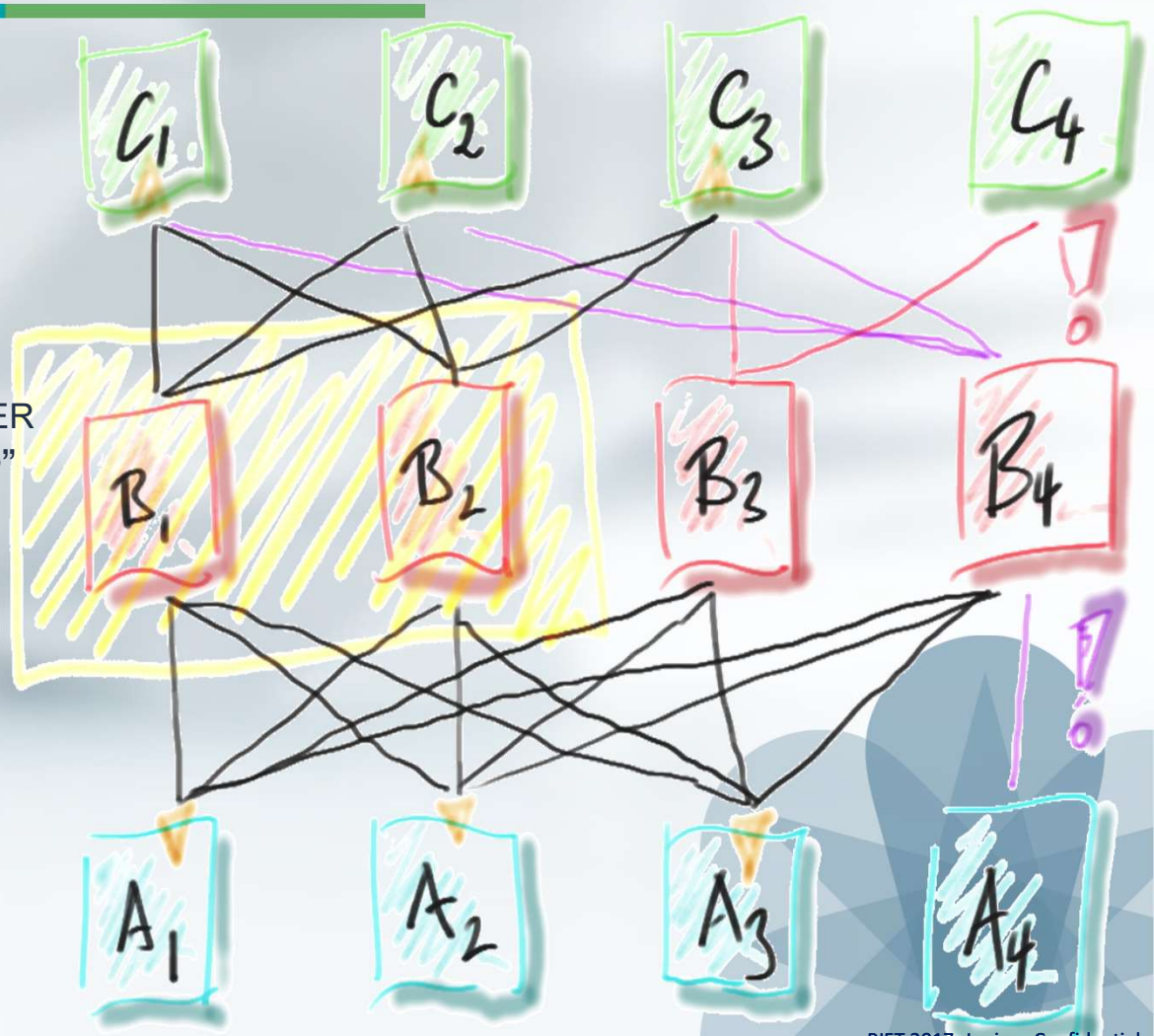
Automatic De-Aggregation



- South Representation of the Red Spine 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

Automatic Flooding Reduction

- Each "B" Node Computes From Reflected South Representation of Other "B" Nodes
  - Set of South Neighbors
  - Set of North Neighbors
- Nodes Having Matching Sets Consider Themselves "Flood Reduction Group"
- Fully Distributed, Unsynchronized Election
- In our Case B1 & B2
- Each Node Chooses Based on Hash Computation which Other Nodes' Information it Forwards on ***First*** Flood Attempt
- Similar to DF Election in EVPN



Moreover

- Traffic Engineering via "Flooded DV Overlay" With Policies
- Completely Model Based Packet Formats
- Channel Agnostic Delivery, Could Be QUICK, TCP, UDP, UDT
- 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

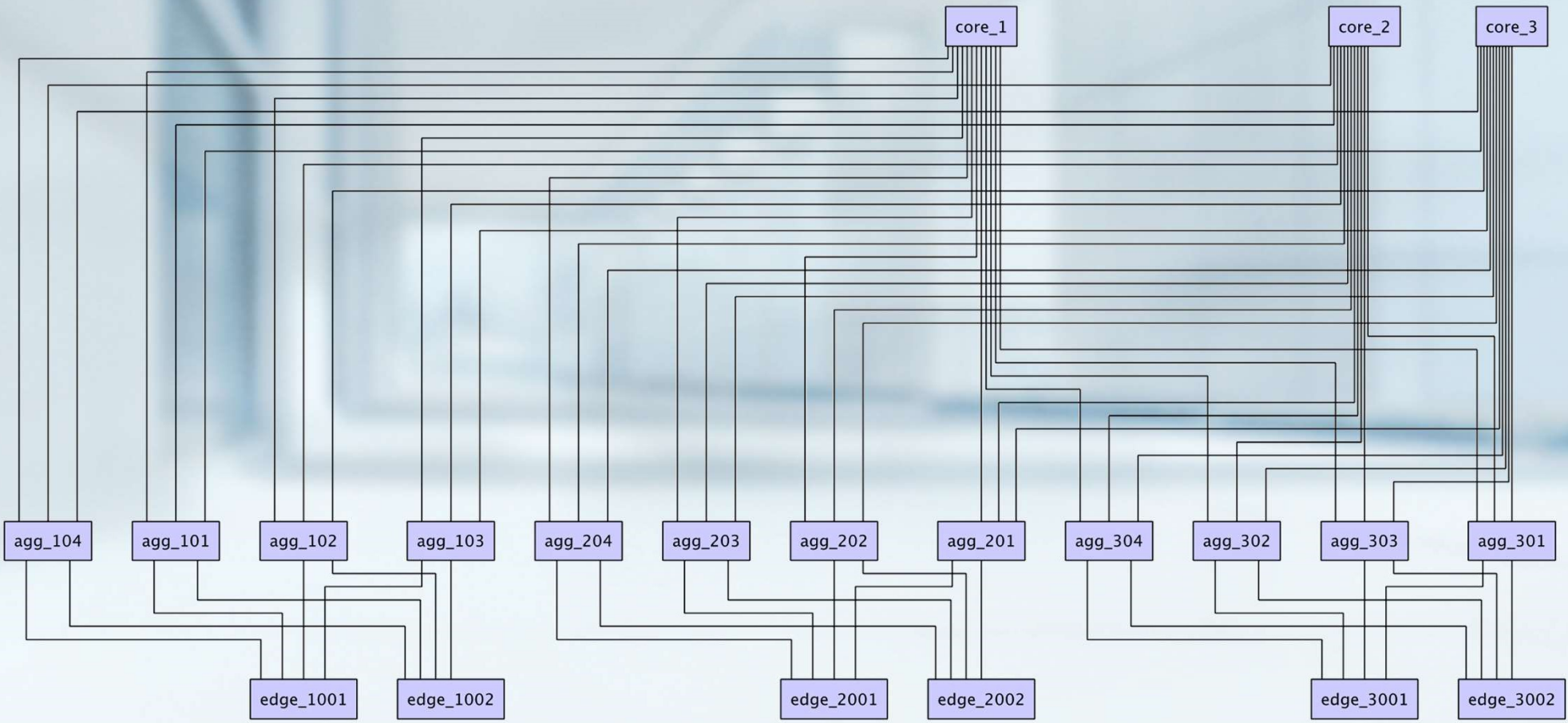
Summary of RIFT Advantages

- Advantages of Link-State and Distance Vector
  - Fastest Possible Convergence
  - Automatic Detection of Topology
  - Minimal Routes on TORs
  - High Degree of ECMP
  - Minimal Blast Radius on Failures
  - Fast De-comissioning of Nodes
  - Maximum Propagation Speed with Flexible # Prefixes in an Update
- No Disadvantages of Link-State or Distance Vector
  - Reduced Flooding
  - Automatic Neighbor Detection
- And Some Neither Can Do
  - Automatic Disaggregation on Failures
  - Key-Value Store

Early Juniper Implementation

- Pre-Production Code
- Flooding/Hello/SPF/RIB/Flood Reduction Implemented
- Optional REDIS Persistency for Link-State DB and Statistics Implemented
- Very Heavily Multi-Threaded to Fully Utilize Modern CPU Architectures
  - Memory and Thread Safe Unlike C
- Own Topology Environment for Heavy-Duty Testing
- Easily Portable Across All Modern OS Platforms

INDICATIVE NUMBERS FOR RIFT: TOPOLOGY



Indicative Numbers for RIFT Implementation

- MacPro (Low Power I7 with 4 Real Cores)
- 21 Nodes/60 links/600 Prefixes Executed
- ~480 Threads
- Flooding on Hysteresis
  - Around 10+K TIEs/Sec per Adjacency

- **Convergence From Cold Start**
  - Total CPU Use (4 Cores I7)
    - Flat: 11secs ~ 500 msecs/Node
    - Rift: 3.2secs ~ 150 msecs/Node
  - RIB Convergence for Core Super-Spines
    - Flat: 900msec
    - Rift: ~250msec
  - Amount of Flooding **(no Flood Reduction Applied Yet)**
    - Flat: ~13K Ties, 96K Total Transmissions
    - Rift: ~5K Ties, 21K Total Transmissions

- **Single Link Flap (Core_1 to Agg_104)**
  - Flat: Avg: ~60 msecs, Max: ~75 msecs
  - Rift: Avg: ~35 msecs, Max: ~ 70 msecs

Ongoing/Future Work

- Automatic De-Aggregation
- Northbound Computation
- Large Scale Runs on Server Farms
- Prefix Guided Prefixes
- Integration with One of Our Platforms
- Ti[DR]es on ToS (or Different Port)

- Key Customer Engagements
  - Requirements/Use Cases
  - Draft Co-Authorship Desired


JUNIPER
NETWORKS
THANKS