RIFT: Open Standard, Zero OPEX, IP Fabric Routing Underlay

IEEE 802/IETF DC Workshop

Tony Przygienda
Distinguished Engineer, Juniper Networks

draft-ietf-rift-rift @ IETF

Disclaimers and Expectations

- This is an IETF "Working Straw-Man Proposal"
- Unless Specifically Stated None of those Things Constitute Commitments to Product Specifications, Offerings or Release Dates by Juniper Networks at this Point in Time

WHAT AND WHY ?

- HYPER-SCALERS ARE EXTRAPOLATING THE THINGS TO COME
  - VAST AMOUNT OF BANDWIDTH CLOSE TO PRODUCER & CONSUMER NECESSARY
    - IP FABRICS IN DC (SERVER FARMS)
    - METRO (CACHES AND ACCESS)
    - DISAGGREGATED CHASSIS ARCHITECTURES
  - THOSE TOPOLOGIES ARE BECOMING UNIFORM, LOCAL AND REGULAR
  - WAN-STYLE TRAFFIC ENGINEERING & PROTECTION IS BEING REPLACED BY WIDE FAN-OUT & DISTRIBUTED SYSTEMS REDUNDANCY (RATHER THAN CHASSIS & FRR)
  - HYPER-SCALERS ARE BUILDING CUSTOMIZED HIGH-OPEX SOLUTIONS TO MANAGE THOSE FABRICS
- IP FABRIC IS BECOMING THE NEW "RAM CHIP" TO CONSUME BANDWIDTH
  - NO'ONE CONFIGURES RAM BANKS AND CAS/RAS MANUALLY IN EVERY LAPTOP
  - IP FABRICS HW IS LARGELY COMMODITY ALREADY
  - IP FABRICS MUST "OPEX COMMODITIZE"
- CUSTOMERS ARE HOSTING THEIR CONTENT & CRITICAL BUSINESS PROCESSES
  - HYBRID CLOUD FOR MANY REASONS, ONE OF THEM TO KEEP REAL-ESTATE FROM HYPER-SCALERS
  - NEED TO BUILD OWN FABRICS
  - HARD TO SUSTAIN PROPRIETARY OPEX EFFORTS

Agenda

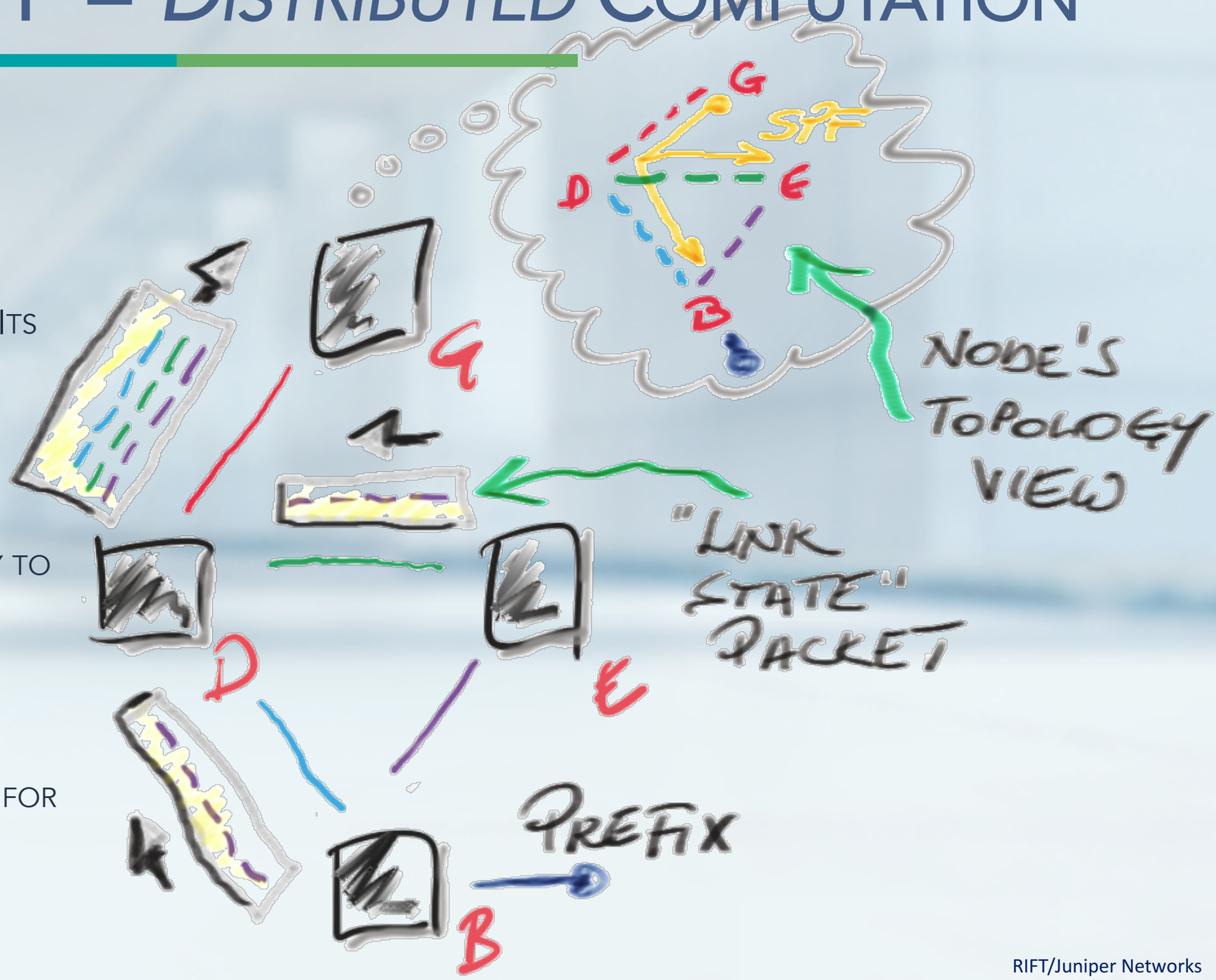
- Blitz Overview of Today's Routing (if Needed)
- "Fabric Routing" is a Specialized Problem
- RIFT: a Novel Routing Algorithm for IP Fabric Underlay

Blitz Overview of Today's Routing

- Link State & SPF
- Distance/Path Vector

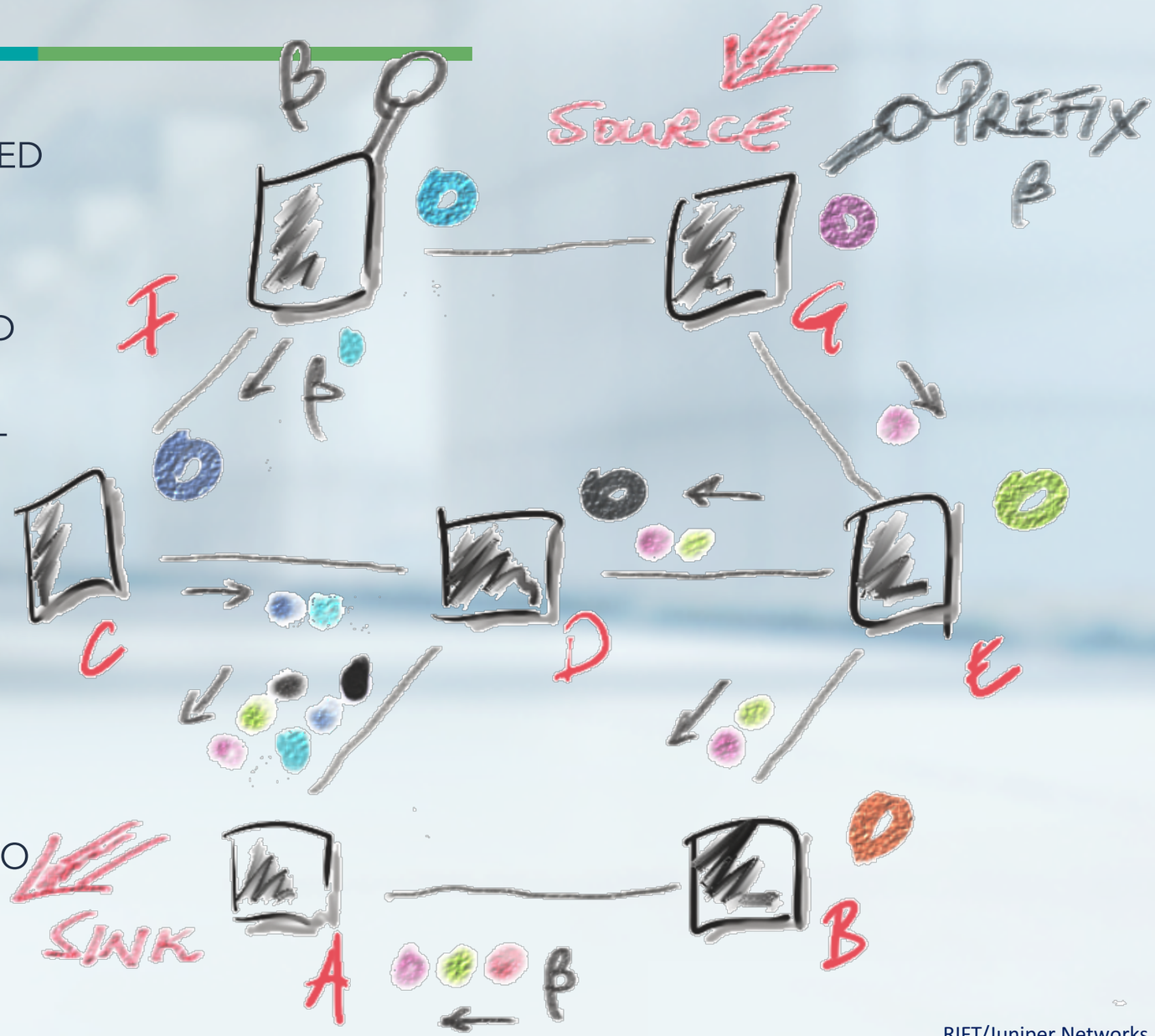
Link State and SPF = *Distributed* Computation

- 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

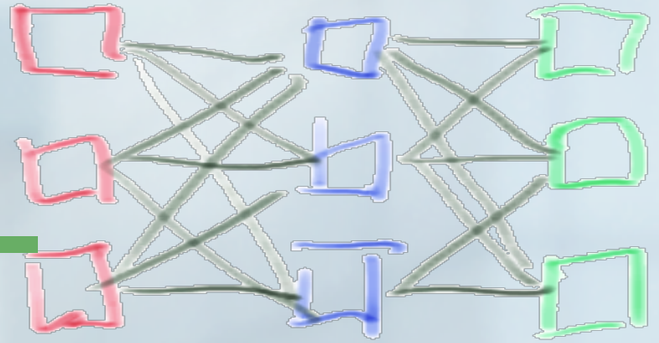
- Prefixes "Gather" Metric When Passed Along Links
- Each Sink Computes "Best" Result And Passes It on (Add-Path Changed That)
- A Sink Keeps All Copies, Otherwise It Would Have to Trigger "Re-Diffusion"
- Loop Prevention is Easy on Strictly Uniformly Increasing Metric
- Ideal for "Policy" Rather Than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



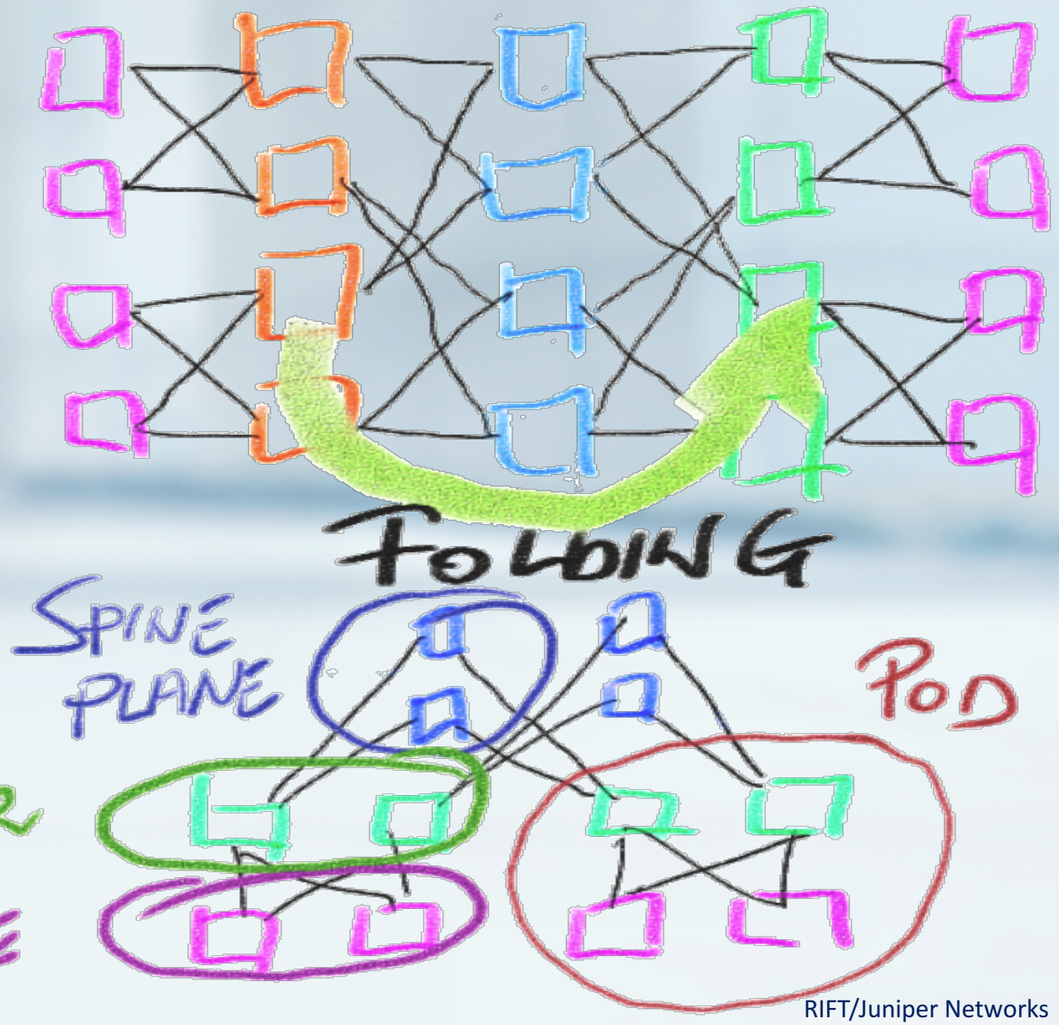
DC Fabric Routing: a Specialized Problem

- Clos Topologies Are Dominant Today
  - Toroidal [and Diagonal] Meshes have Long Paths, Small Bisection Width and Poor Blocking Properties
  - Dragonfly (and Some Probabilistic Variants) is very Novel and Unproven
    - ½ Throughput of CLOS at Same Cost Due to Low ECMP
    - RIFT Could 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)
  - Generally a "Better" Approach to Faster Convergence
  - Current Attempts to Deal with Some "Spot Problems" like Flooding Reduction
- Yet Others Run BGP over IGP (Traditional Routing Architecture)
- Less Than More Successful Attempts @ Prefix Summarization, Micro- and Black-Holing, Blast Radius Containment
- Server Multi-Homing not Possible Using IP Due to Equal Cost and Scaling Constraints, Hence MC-LAG'ed Solutions or EVPN
- In Summary: High OPEX Solutions Not Necessarily Viable for Customers Who Cannot or Do Not Want to Build Sophisticated Talent Pool to Deal with Their "Unicorn" Fabrics

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
Optional Peer Discovery/True ZTP/Preventing Cabling Violations	⚠	⚠	✓
Minimal Amount of Routes/Information on ToRs, light-weight enough for servers, Can Scale to Multi-Homed Server Architectures	✗	✗	✓
High Degree of ECMP (BGP needs lots knobs, memory, own-AS-path violations) and ideally NEC and LFA	⚠	✓	✓
Non Equal Cost Multi-Path, Equal Cost Independent Anycast, MC-LAG Replacement	✗	✗	✓
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	✗	✓	✓
Bandwidth Load Balancing	✗	✗	✓
Simplest Initial Implementation	✓	✗	✗

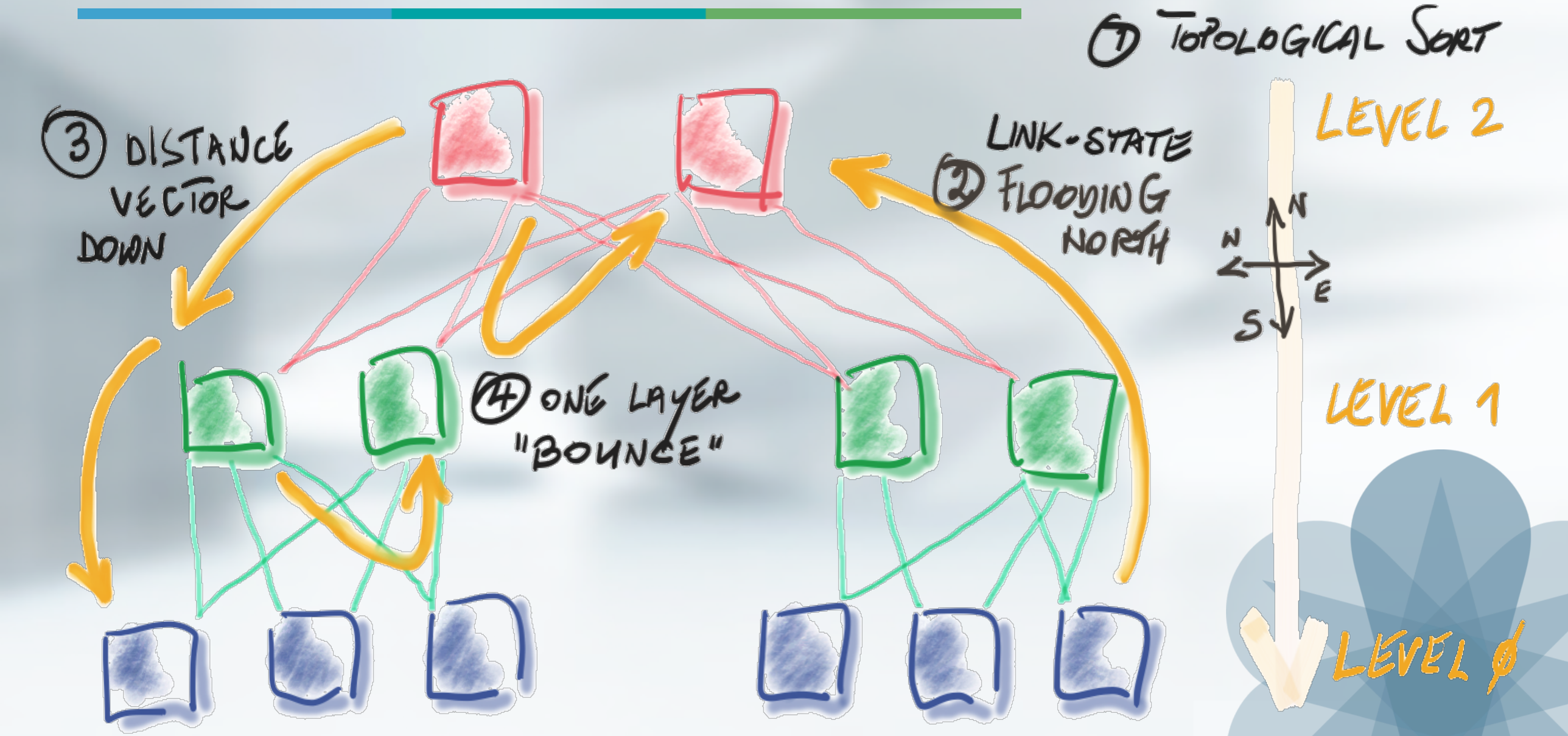
RIFT: Novel Routing Algorithm for Clos Underlay

- General Concept
- Automatic Disaggregation
- Automatic Bandwidth Balancing
- Fast Mobility Support
- And More

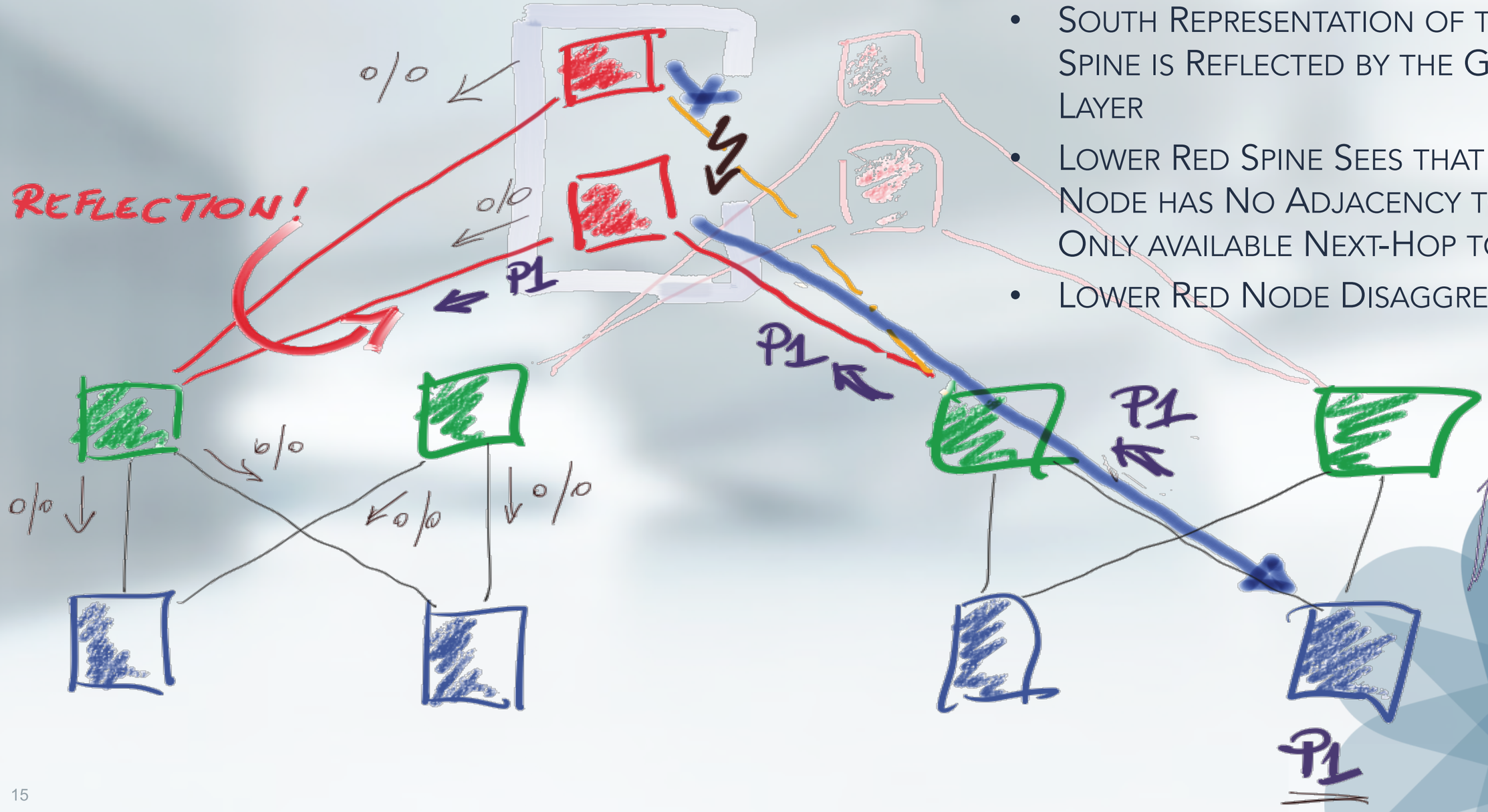
"Just because the standard provides a cliff in front of you, you are not necessarily required to jump off it."

— Norman Diamond

Link-State Up, Distance Vector Down & Bounce

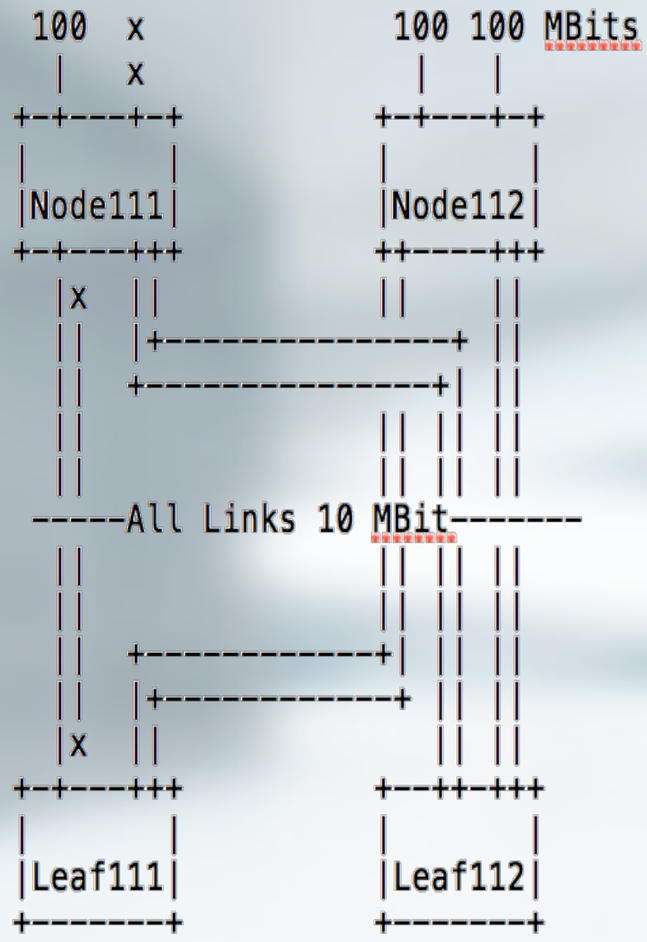


Automatic De-Aggregation



- 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

NORTHBOUND BANDWIDTH BALANCING



- RIFT calculates the amount of northbound bandwidth available towards a node compared to other nodes at the same level and adjusts the default route distance accordingly to allow for the lower level to have different weights on load balancing.
- **BAD_N:** Bandwidth Adjusted Metric to N
- **L_N_u:** as sum of the bandwidth available from L to N
- **N_u:** as sum of the uplink bandwidth available on N
- **T_N_u:** L_N_u + N_u
- **M_N_u:** log_2(next_power_2(T_N_u))
- **BAD_N:** D * (1 + maximum_of_all(M_N_u) - M_N_u)

Node	N	T_N_u	M_N_u	BAD
Leaf111	Node111	110	7	2
Leaf111	Node112	220	8	1
Leaf112	Node111	120	7	2
Leaf112	Node112	220	8	1

Mobility Support

- Optional clock attribute on prefix
- If clock not present, address is anycast
- If present, always better than none
- If both present, RFC5905 or better on fabric assumed
  - If IEEE802_1 less than 200msec diff, transactionid (TID) if present tie-breaks
  - Otherwise timestamp compares
- TIDs are coming from draft-ietf-6lo-rfc6775-update or similar mechanisms

And Then …

- Complete ZTP
  - Not Even Addressing Necessary (Except v6 Local via ND)
  - IPv4 over IPv6 Forwarding
  - Arbitrary Number of Levels
  - Heterogeneous PoD Height Possible
- Loop-Free, i.e. All Paths Through IP Fabric Can be Saturated
- Normal Operation Has Only Default Routes on Leafs
- Minimal Blast Radius
  - Automatic Optimal Flooding Pruning and Load Balancing on Changes for Maximum Scaling
- Northbound Bandwidth Balancing on Link Loss
- K/V Store
- Completely Model Based Packet Formats
- Flooding over UDP for Fastest Convergence
- Policy Controlled Key-Value Store Support
- Possible SR Support

Standardization & Open Source

- Standards Track Working Group in IETF
  - Juniper & Apstra Co-Chair
- Cisco, ComCast, Yandex, Mellanox, HPE Co-Authors
  - Bloomberg, Criteo & Others Engaged
- YANG, SR & Other Things under Works
- Open Source Implementation
  - https://github.com/brunorijsman/rift-python

Summary of RIFT Operational Advantages

- Open IETF Standard
  - Can Build Hybrid Vendor Fabrics
  - Protocol is Well Reviewed and Understood by World-Class Experts
- True ZTP
  - No Configuration Necessary
  - V4 over V6 Forwarding
  - Mis-cabling Handled
- Can Operate on Asymmetric Bandwidth Fabrics and Handle "Fat Link" Failures By Adjusting Automatically
- Can Scale to and Multi-Home Servers
  - No Need for Service Migration on ToR Upgrades
  - Can Talk Directly to Hyper-Visors/Kubernetes GW
- BFD is "Built In"
  - Can Be Used for Fast Rehash or Early Loss Detection
- Runs on UDP
  - Trivial Kernel Support on All Platforms
  - Allows for Max. Speed Flooding
  - Easy to "Multi-Instantiate" for Different Purposes
- Minimal Blast-Radius
  - Failures/Bring-Up on Fabric Only Affects the Smallest Viable Radius
- RIFT Flooding is ~30% of Normal Flat IGP
  - Built-In Flood Reduction Reduces Flood Traffic to <20% of Flat IGP
- Loop-Free
  - Can Utilize **All** Viable Paths Through Fabric
  - Can Support True Anycast
- Model Based
  - Much Less Possibility for Weird Parser and Formatter Bugs Plaguing Today's Networking Protocols
- Specification is Written for Maximum parallelization
  - With Enough Cores IP Switches Should Be Able to Converge @ Speeds Making FRR Unnecessary (Assuming Fast Rehash)
- KV Store Allows to Replace Out-Of-Band Applications
  - IP/MAC Binding Can Be Flooded to Top-of-Fabric

Summary of RIFT Protocol Advantages

- Open IETF Standard
- Advantages of Link-State and Distance Vector
  - Fastest Possible Convergence
  - Automatic Detection of Topology
  - Minimal Routes on TORs
  - High Degree of ECMP
  - Fast De-comissioning of Nodes
  - Maximum Propagation Speed with Flexible # Prefixes in an Update
  - Vector
  - Reduced Flooding
  - Automatic Neighbor Detection
- Unique RIFT Advantages
  - Bandwidth Re-Balancing
  - Automatic Disaggregation on Failures
  - Key-Value Store
  - Horizontal Links Used for Protection Only
  - Minimal Blast Radius on Failures
  - Can Utilize All Paths Through Fabric Without Looping
  - Supports Non-Equal Cost Multipath and Can Replace MC-LAG
  - True ZTP

More Material

- Specifications in IETF Working Group
  - https://datatracker.ietf.org/doc/draft-ietf-rift-rift/
- Walk Through Major Concepts & Package Explanation (RIFT Interim Recording)
  - May 3, 2018: https://www.youtube.com/watch?v=dtxNoCkC7MA
  - May 2, 2018: https://www.youtube.com/watch?v=BZtFPTgCSBs
- Juniper's Public Stand-Alone Package Downloadable
  - https://www.juniper.net/us/en/dm/free-rift-trial/
- Open Source Implementation
  - https://github.com/brunorijsman/rift-python

THANKS