DC Routing
Current Challenges and RIFT
(Routing In Fat Trees)

- Alankar Sharma
  Sr. Principal Architect
  (Datacenter Design & Strategy)
  Comcast
Networking Progression (IPv4)

NETWORKING

NAT-WORKING

NOT-WORKING

IPv6
Hyper Scale Datacenter (BGP)

RFC 7938

SAS01
SAS02

AS01
AS02
AS03
AS04

S01
S02
S(n)
S(n+1)

LEAF 65004
LEAF 65005

POLICY/ACL

Backbone/ISP

EBGP
EBGP
EBGP
EBGP

Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes

LEAF 65004
LEAF 65005

POLICY/ACL

Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes

LEAF 65004
LEAF 65005

POLICY/ACL

Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes

LEAF 65004
LEAF 65005

POLICY/ACL

Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes

LEAF 65004
LEAF 65005

POLICY/ACL

Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes

LEAF 65004
LEAF 65005

POLICY/ACL

Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes
Default Route
Specific Routes

LEAF 65004
LEAF 65005

POLICY/ACL
**Hyper Scale Datacenter (BGP)**

**Loopback Reachability Issue**

- Lo0 not reachable on loss of north bound links
- May show impact on monitoring systems
- Loose in-band telemetry and other polling information
- Requires E-W links/IGP/IBGP (breaking L/S architecture)
Hyper Scale Datacenter (BGP)

Centralized services
SNAT

Backbone/ISP

Centralized Services e.g. Load Balancer

POLICY/ACL

SUPER AGG SPINE 65001

AGG SPINE 65002

SPINE 65003

LEAF 65004

SAS01 SAS02 AS01 AS02 AS03 AS04 S01 S02 sw001a

Specific Routes Default Route

EBGP

L3 P2P

Default Route

Specific Routes

Centralized services

SNAT
Hyper Scale Datacenter (BGP)

Evolution with more Centralized Services
Requirement of being in the routed path
Hyper Scale Datacenter (BGP)

- Break the BGP Loop Prevention (Allow-as in)
- Erase the AS Path (build a policy to replace all path info with null)
- Policies and filtering for damage control

![BGP Diagram](image)

More Centralized services
Requirement of being in the routed path (Security, Firewall etc.)
Hyper Scale Datacenter (BGP)

• Missing topology information as Link State (BGP LS)
  • Topology models are required by external entities like SDN Controller
  • Other custom apps rely on this information
  • As a work around, run IGP but not use it for routing.
    o Comes with its own issues
Hyper Scale Datacenter (BGP)

Practical Scenario: Reality Check

- Issues with Loopback reachability
- Remove AS Path
- Allow-AS-In
- Apply manual/static policy to send/accept default south bound (Prefix-Set, Policy, Apply to a Neighbor)
- Other policies for filtering unwanted prefixes for loop prevention
- No LS info for topology modelling
- Explicit neighbor statements are like static config
- When the desired behavior is always EBGP (IBGP) like, ASN may not be significant.

- IGP: Issues- Scalability, Stability, Management/Ops…)

COMCAST
Sure, It Works!

We can tweak it!
router bgp 65003
  router-id 96.108.202.21
  maximum-paths 32
neighbor V4-EXT-TO-BLUE-FW peer-group
neighbor V4-EXT-TO-BLUE-FW remote-as 65005
neighbor V4-EXT-TO-BLUE-FW fall-over bfd
neighbor V4-EXT-TO-BLUE-FW allowas-in 3
neighbor V4-EXT-TO-BLUE-FW password 7 xxxxxxx
neighbor V4-EXT-TO-BLUE-FW send-community

neighbor 10.144.33.101 peer-group V4-EXT-TO-BLUE-FW
neighbor 10.144.33.101 description fw01

neighbor V4-EXT-TO-BLUE-FW activate
neighbor V4-EXT-TO-BLUE-FW route-map V4-FW-EXT-NEXTTHOP in
neighbor V4-EXT-TO-BLUE-FW route-map V4-DEFAULT-ONLY out

# Policy for default route
route-map V4-DEFAULT-ONLY permit 10
  match ip address prefix-list V4-DEFAULT-ONLY-PFX

# Policy to clear AS Path
route-map V4-FW-EXT-NEXTTHOP permit xx
  set as-path match all replacement none

- Static Config repeated many times (IPv4 & IPv6)
- Most of lines are to hack BGP
- Make these defaults
- Ends up in thousands of lines of code

Vs
  #router rift
**RIFT (Routing In Fat Trees)**

Next-Gen DC Routing: Time to restart, build new rather overloading, complicating existing protocols.


- Purpose built for growing CLOS (Leaf/Spine) architectures in datacenters

- Taking advantage of somewhat deterministic nature of these topologies with fabric like lots of links
  - Unlike most of the current routing protocols which are well suited for internet like (irregular) topologies with low degree of connectivity
RIFT (Routing In Fat Trees)

**Significant Distinguishers**

- Hybrid, as best of both protocols- DV and LS
  - DV down
  - LS up
- Minimize routing state at each level. Leaf layer has just the default route
  - Default route south
  - Specific routes north
- Topology awareness (Leaf/Spine levels)
  - Concept of Aggregation Levels
  - Detection of mis-cabling
- Smart Disaggregation
  - To prevent black-holing or sub-optimal routing upon a link/node failure
RIFT (Routing In Fat Trees)

Significant Distinguishers

- Unequal Cost Multi-pathing support
- Load balancing of inbound traffic based on outgoing bandwidth across multiple nodes in the same layer
- Built-in Route Reflection
  - To overcome East-West links in CLOS
- Flood Reduction
  - Prunes the protocol routing updates to an optimized subset of links
- Autonomous Routing (ZTP-ZT, No controller, AI)
  - Seed based automatic construction of tree
  - No routing configurations required via controller or automation/ZTP
  - A node could figure out its position and start routing
Other features

- Overload Bit
- Key-Value Store: KV-TIE
- BFD
- MT (Multi-topology)
- Policy-Guided Prefixes
- Label Binding
- Support for Segment Routing
Levels (Topology Awareness)

- Aggregation Level
- Backbone
- Level 3
- Level 2
- Level 1
- Level 0
- POD 1
- POD 2
- POD M
- POD N

- X = Routing Node
- = Server Rack
- = Network Links

Number of PODs Or Racks
Smart Disaggregation

Prevents Black-holing or Backhauling

- Nodes at same level detect different set of south neighbors
- Lost prefix gets advertised by other nodes as a specific route. Gets disaggregated from default route
- Routing change is confined within the level
- Disaggregated prefixes are not reflected north
Built-in Route Reflection

- L/S natively does not have East West links
- Every *NODE* S-TIE is "reflected" northbound to level from which it was received

Loopback Injection

- Node can inject Loopback into N-Prefix TIEs (Topology Information Element) for reachability under normal operations
- Node can inject Loopbacks on north connectivity failures into S-PGP TIEs for reachability “From the South”
Flooding Reduction

- Starting from bottom each node picks its FL and Backup FL based on level on connectivity and running a hash.
- Each prefix is flooded twice.
- Distributed flooding topology. Fast and efficient.
Upon northbound link failure from AS01 to BB, Spine layer will adjust accordingly to send $\frac{1}{3}$rd traffic to AS01 and $\frac{2}{3}$rd traffic to AS02.

- Nodes compute/advertise Bandwidth Adjusted Distance (BAD).

- Weighted ECMP behavior for balancing in the fabric.
Autonomous Routing (AI/Self-Aware)

- Routers can autonomously figure out their roles
- No automation or controller required to deliver the configurations

- **Seed**: A node (topmost) is needed as seed and given the concept of levels and deterministic nature of L/S topology, other node can derive their relative positions
- Link Id auto generated
- Interfaces can use “ip-unnumbered”

- Mis-cabling: A node can generally connect to adjacent levels. More than 3 unique values received or difference of gt 2 in received values, indicates cabling issue
Auto-configure (Concept- High Level)*

Case I: 1 value (X) received for LEVEL.  
My LEVEL is (X-1)

Case II: 2 values (X, Y) received for LEVEL. Higher is X and lower is Y
If X-Y <=2 , my LEVEL = (X-1)
If X-Y > 2, my LEVEL = INVALID.
Indicate cabling error

Case III: 3 values (X, Y, Z) received for LEVEL. X>Y>Z
If X-Y = 1 AND Y-Z = 1, my LEVEL = (X-1)
If X-Y != 1 OR Y-Z != 1, INVALID.
Indicate cabling error

Case IV: More than 3 values received for LEVEL
Indicate cabling error

So What?

https://funnymemes.co/
RIFT as Open Standard in IETF

- Standards Track Working Group in IETF
  - @ https://datatracker.ietf.org/wg/rift/about/

- Specification Completely Open
  - @ https://datatracker.ietf.org/doc/draft-ietf-rift-rift/

- Co-Authorship by Major Vendors

- Drafts for YANG & Other Necessary Stuff Forthcoming

- Freely available binary package with implementation
  - @ https://www.juniper.net/us/en/dm/free-rift-trial/

- First Hackathon @ IETF 102
  - @ https://trac.ietf.org/trac/ietf/meeting/wiki/102hackathon