Design and Performance of a Simple Data Center Network Protocol

Dr. –Ing. Nirmala Shenoy, Professor, ISchool, School of Information,
Prof. Bill Stackpole, Cybersecurity Department
Golisano College of Computing and Information Sciences,
Rochester Institute of technology, Rochester, New York

Growth of Datacenter Networks

- Datacenters comprise of thousands of servers per location, extend to multiple locations
- Data Center Network (DCN) is the backbone that connects servers in the datacenter
- Research on DCN architectures and topologies continue
- Growing energy and carbon footprint concerns,
  - Costs – OPEX and CAPEX from RFC 7938
- High maintenance and troubleshooting efforts / costs
- Several other challenges …
- FOR OUR TESTS
- Adopted the popular Folded-Clos topology
- Research - How to simplify the protocols for a folded-clos topology DCN?

Protocols for Datacenter Networks

- Several protocol suites have been investigated for folded-Clos topology DCNs.
- Minimally - a routing protocol, a load balancing protocol and if required a protocol to speed up failure detection.
- A popular protocol suite used in folded-Clos topology DCN
  - Border Gateway Protocol (BGP) for routing
  - Equal Cost Multipath protocol (ECMP) for load balancing
  - Bidirectional Forwarding Detection (BFD) to speed up failure detection
  - BGP requires Transport Control Protocol (TCP) for its operation and
  - BFD requires User Datagram Protocol (UDP) for its operation
- Increased set of protocols - increased operational complexity - increased configurational, troubleshooting and management needs.
- Increased energy, cooling and equipment cost

Routing in a DCN – A New Approach

- To route traffic between servers in a datacenter, the routers require information about the server networks (IP addresses) and how to reach them
- Routers store multiple paths between the servers – fallback
- Currently we disseminate network information to all routers to set up multiple routing paths between server racks

Proposed Multi-Root Meshed Tree Protocol (MR-MTP)

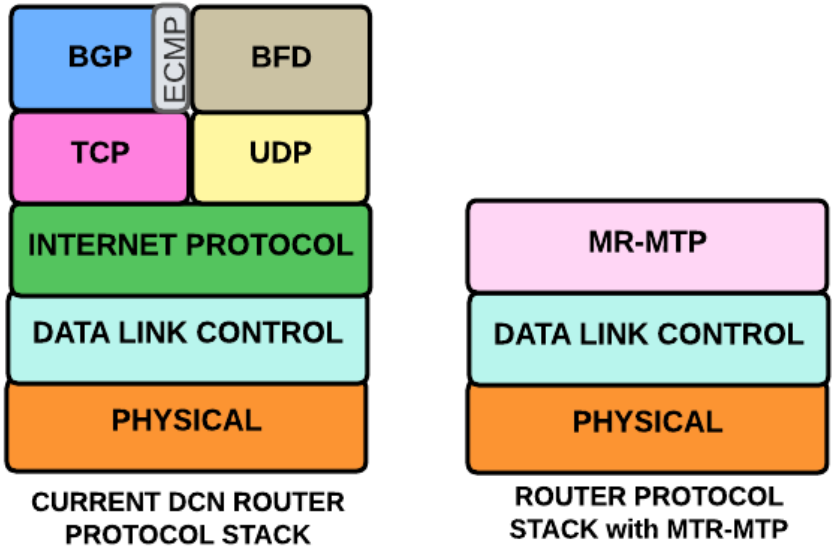
- Uses the Folded-Clos topology (structure) to simplify routing operations.
- Establishes all loop-free paths from ToRs (top of rack) switches to all top tier spines
- Trees start at ToRs and mesh at the upper tier spines (no loops)

The Multi-Root Meshed Tree Protocol

- MR-MTP establishes all routes from ToRs to top tier spines using Virtual IDs (VIDs)
  - VIDs are auto assigned by MR-MTP
- MR-MTP unifies routing, load balancing and fast failure detection in a single protocol
- MR-MTP encapsulates and forwards IP packets between servers.
- MR-MTP is independent of Layer 3 – Layer 3 agnostic.
- MR-MTP defines its own headers (introduced later)
- MR-MTP is backward compatible to Ethernet (MR-MTP messages are carried in Ethernet frames) and IP

MR-MTP Protocol Stack

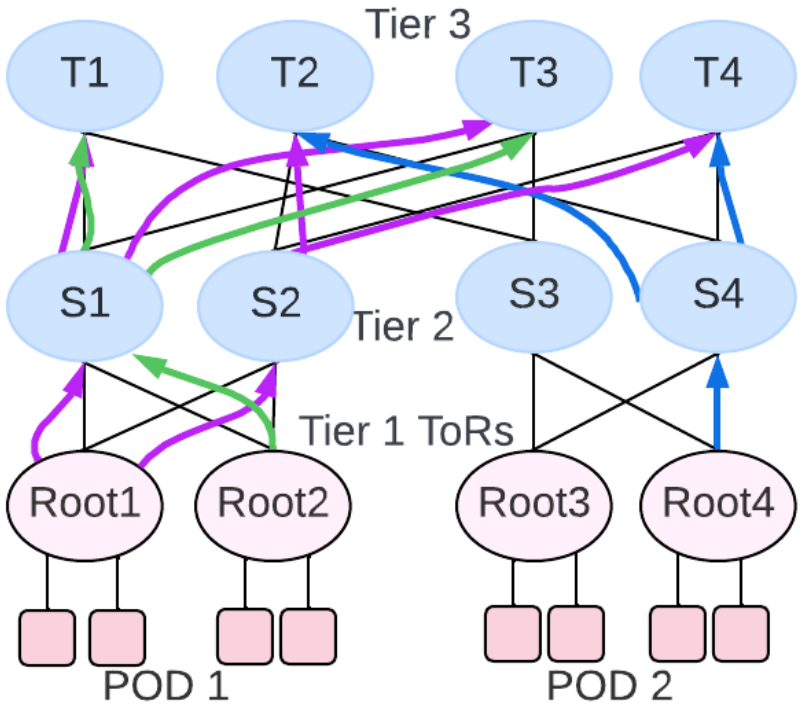
- In its current version MR-MTP replaces BGP, ECMP, BFD and IP. It also avoids the need for TCP (required by BGP) and UDP (required by BFD)
- The router protocol stack is cut down significantly – see figure below
- The benefits can be multifold



MR-MTP Features / Configuration Needs

- MR-MTP routers require tier information be configured
  - ToRs at tier 1, spines at tier 2, 3 etc
  - ToRs need a Virtual ID – currently auto derived
- MR-MTP (C code) executable code size is 40 Kbyes.
  - https://github.com/pjw7904/CMTP
  - FABRIC testbed scripts - https://github.com/pjw7904/FABRIC-Automation
- MR-MTP can be turned off to fall back to current protocols
  - This will help in incremental deployment
- MR-MTP in one location can communicate with BGP in another location

Meshed Trees in a Folded-Clos topology – The Concept



Picture shows meshed trees constructed by protocol

- **ToRs are roots of the meshed trees**
- **Note the purple tree from Root1**
- **A partial green tree from Root2**
- **--- so on**
- **A partial blue tree from Root4**
- **All trees mesh at all upper tier spines.**
- **Each tree from a ToR reaches all the top tier spines.**
- **The meshed trees cover all loop-free paths from each ToR to every top tier spine.**
- **How to implement this?**

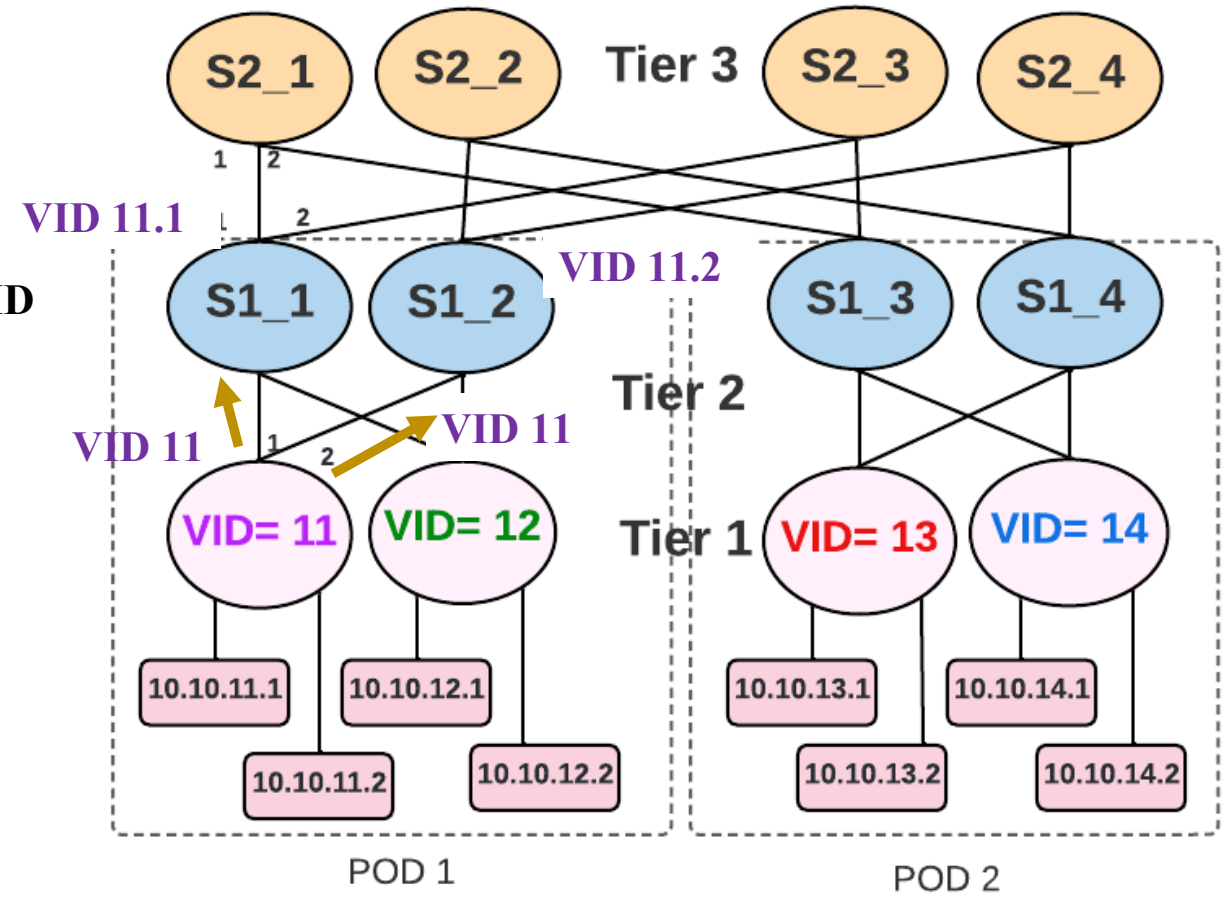
Establishing Meshed Trees with Virtual IDs – MR-MTP Operation

And then

Spines S1_1, S1_2 send in a request. ToRs assign VIDs 11.1 and 11.2 by appending the port number to their VID

ToRs advertise their VIDs

Assume ToRs have assigned VIDs – such as 11, 12, 13, 14



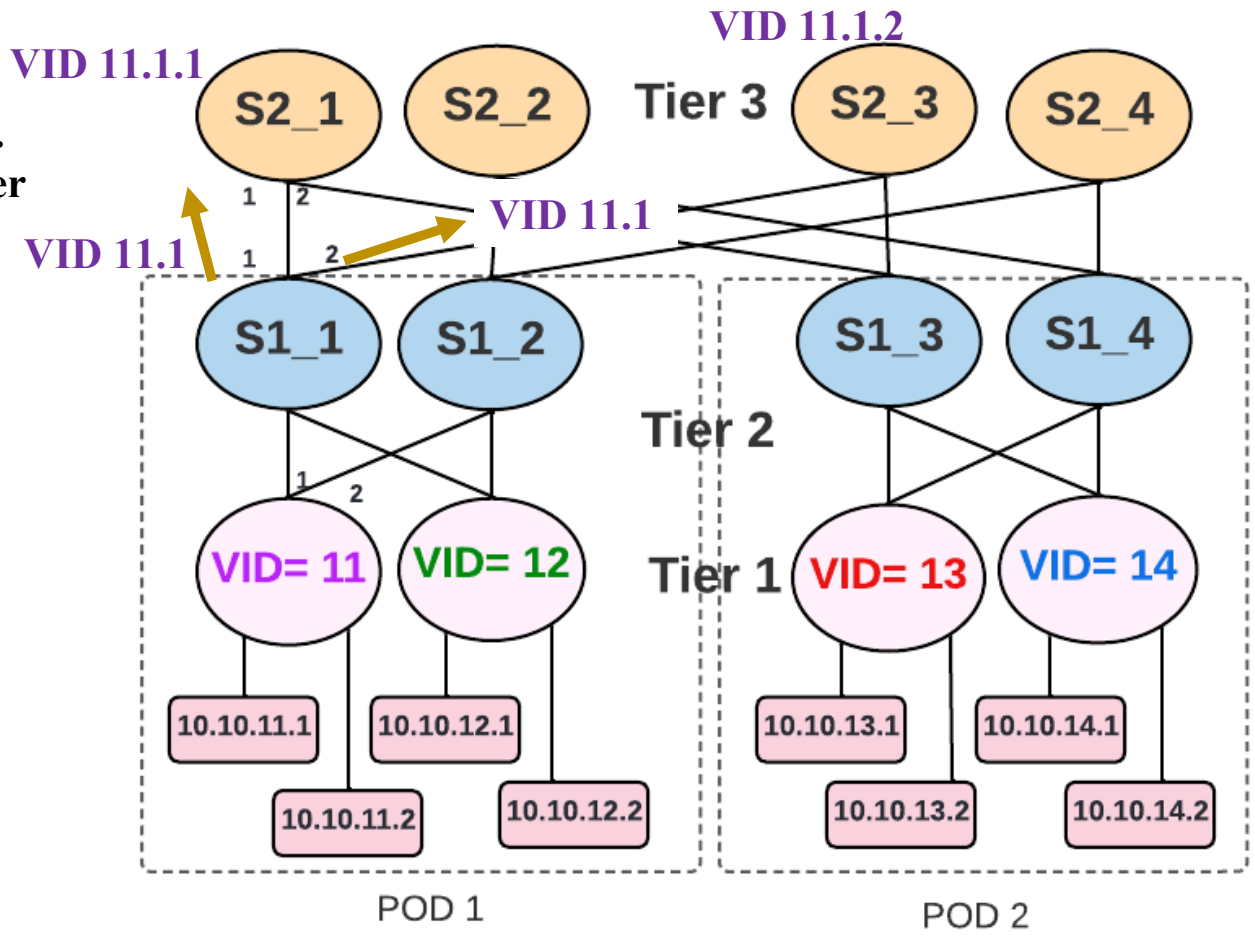
NANOG™

Establishing Meshed Trees with Virtual IDs

Spines S2_1, S2_3 send in a request. S1_1 assigns VIDs 11.1.1, 11.1.2 after appending the port number (on which the request arrived), to their VID

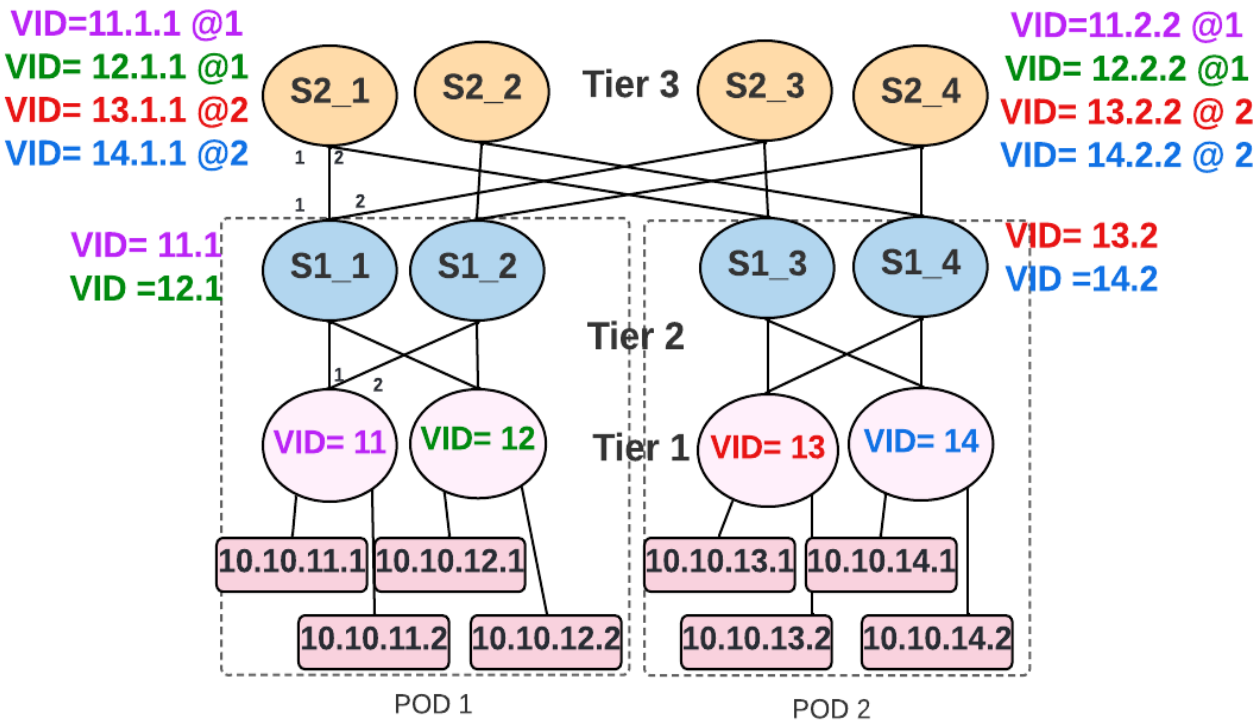
Spines S1_1advertise its VIDs

ToRs derive a unique VID from the subnet IP address (other secure algorithms to auto derive ToR VIDs have been tested)



NANOG™

Virtual IDs Maintain Routing Paths



Spines store acquired VIDs @ ports of acquisition

All route paths are established using simple VIDs. Trace the color.

No routing protocols,

No route or network reachability dissemination

No IP addresses to networks, devices.

THE ONLY CONFIGURATION REQUIRED IS TIERS OF THE DEVICES

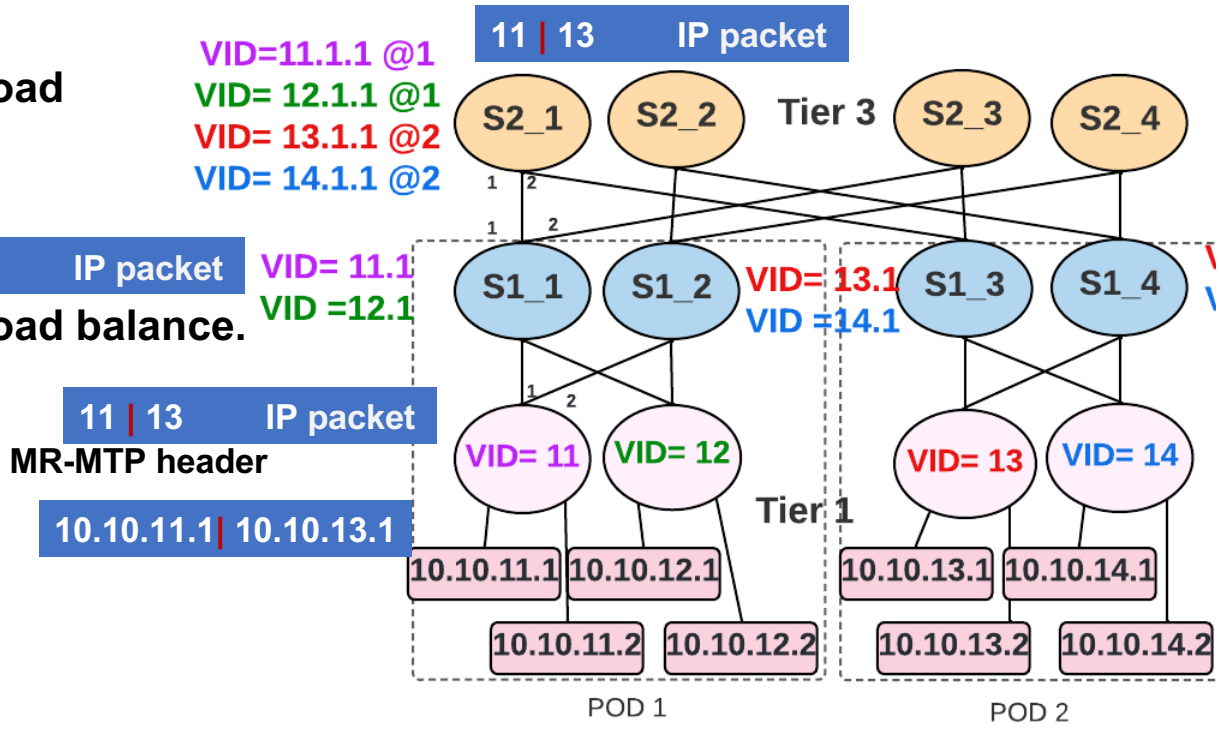
IP Packet Forwarding Between Servers

Spine S1_1 checks its VID table.
No entry for dst VID 13.
Default - send to upper tier after load balance. Send to S2_1

ToR 11 Checks VID table.
No entry for dst VID 13.
Default - send to upper tier after load balance.
Send to S1_1

ToR encapsulates with headers.
Src, dst VID derived from subnet address

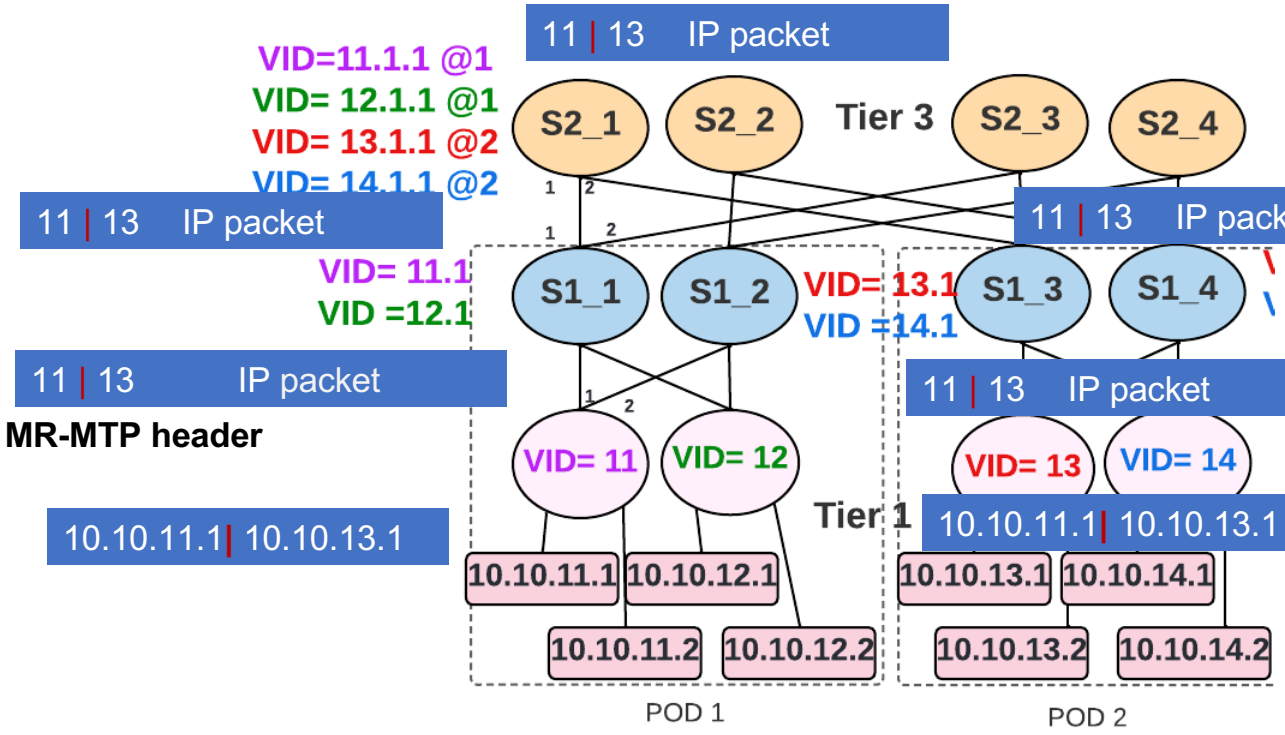
IP packet arrives at ToR 11.
Src=10.10.11.1, Dst = 10.10.13.1



NANOG™

IP Packet Forwarding Between Servers

Spine S2_1 checks its VID table. Dest VID 13 at port 2. Send to S1_3



Spine S1_3 checks VID table. Dest VID 13 at port 1. Send to Tor VID 13

ToR checks Dst VID. This is the Dest ToR. De-encapsulate IP packet. Send to server 10.10.13.1

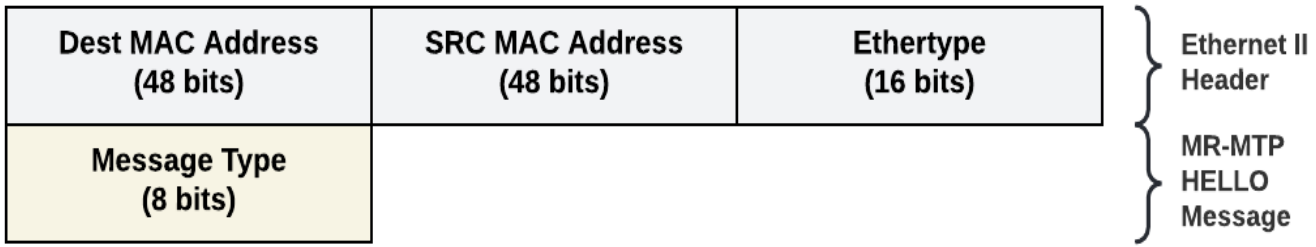
No IP addresses to route. No routing protocol.
The whole IP packet (with IP addresses) can be encrypted.

NANOG™

Improved Availability With MR-MTP

- QUICK TO DETECT- SLOW TO ACCEPT
- All MR-MTP messages are keep-alive
- If there are NO MR-MTP messages to send for the duration of 'hello timer' send a 1-byte hello message
- Missing messages for 1.5 * hello timer – assume neighbor down.
  - Speeds up failure detection – QUICK TO DETECT

Keepalive/Hello



Improved Availability With MR-MTP

- To handle route /interface flapping and dampening – SLOW TO ACCEPT
  - After receiving three consecutive messages – assume neighbor up
  - Benefits - failure detection is 3 times faster.
- Update dissemination message carries only lost/added VID
- On receiving an update, a node notes
  - A Dest VID is inaccessible on the port on which the failure message was received.
  - Low control overhead

UPDATE

Dest MAC Address (48 bits)	SRC MAC Address (48 bits)	Ethertype (16 bits)
Message Type (8 bits)	Update Option (8 bits)	Num of VIDs (16 bits)
VID 1 Size (8 bits)	VID 1 (variable length)	
VID 2 Size (8 bits)	VID 2 (variable length)	
...	...	

Ethernet II Header

MR-MTP UPDATE Message

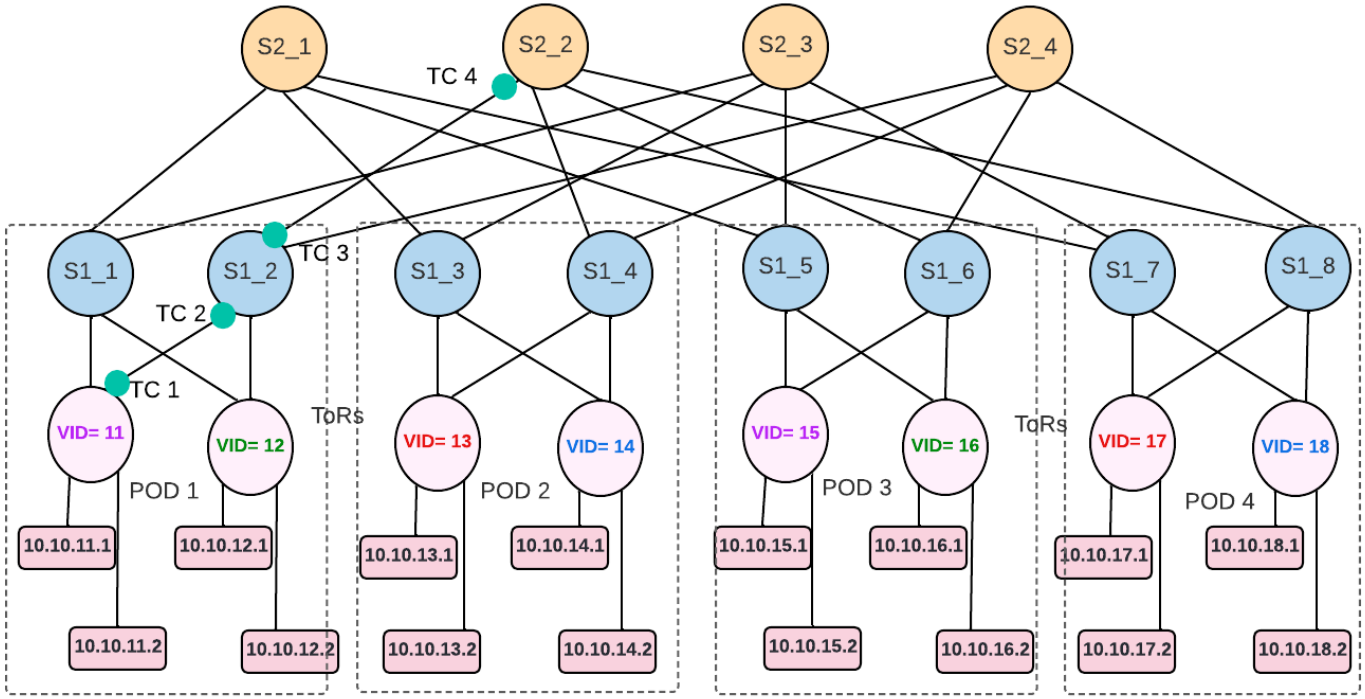
Performance Comparison

- Test Topology - 4 –Pod folded-Clos topology
- BGP/ECMP/BFD protocol suite used for comparison studies
  - BGP modified to work on folded-Clos topology, adjusted AS numbers – requires TCP
  - Used Bidirectional Forwarding Detection to speed up failure detection – requires UDP
  - Equal Cost Multipath Protocol – used with BGP for load balancing.
  - IP for Packet Forwarding
- Presented work – eBGP with modified AS numbers for DCN operation
- Current work – BGP for DCN as per RFC7938
  - Results and demo on FABRIC available

MR-MTP Operation Summary

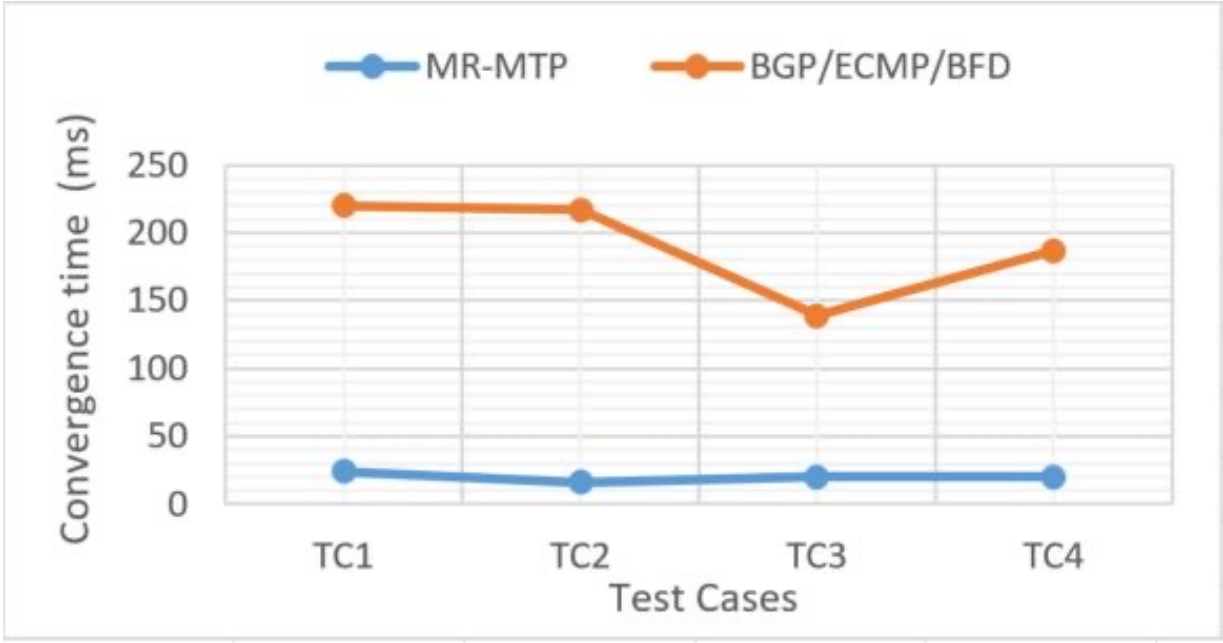
- MR-MTP operates over Ethernet (Layer 2)
- Agnostic to any layer 3 protocols. Can take packets from any network
- Replaces BGP, ECMP, BFD, TCP, UDP, IP
  - Heavy reduction in operational complexity and memory needs
- Backward compatible with IP (v4, v6) and Ethernet
  - Communicate with another DCN running IP, Ethernet etc.
- MR-MTP can be turned on/off

Performance – Test Topology and Test cases



Test Topology and Test Cases (Green Dots)

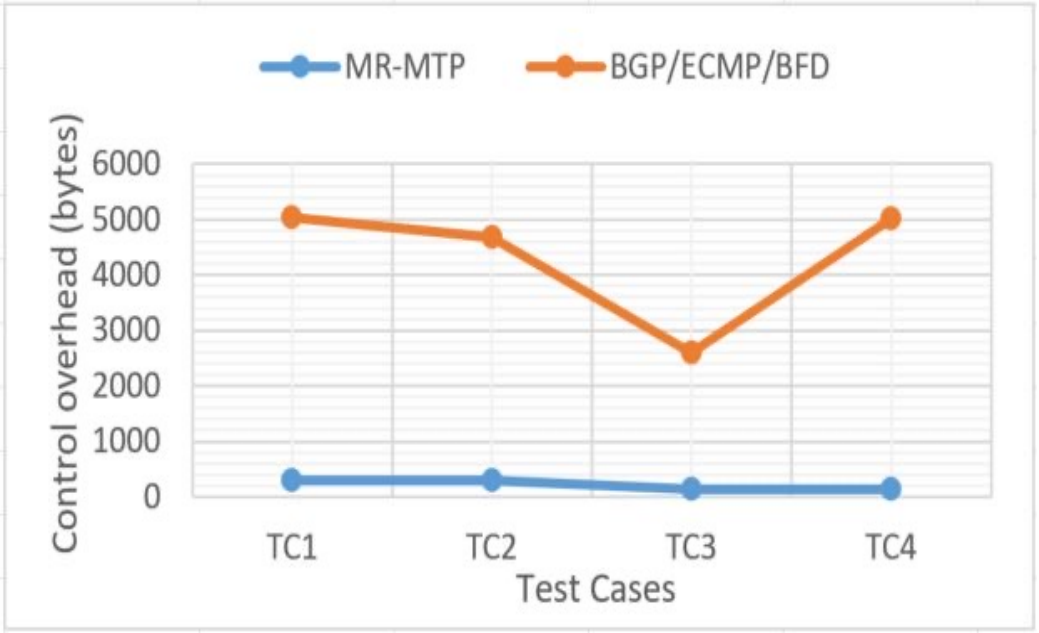
Convergence in ms – Routing Table Stabilization time



BGP/ECMP/BFD convergence time (**140 to 220 ms**)
MR-MTP – convergence time (**around 25 ms**)

Control Overhead

```
Frame 5: 565 bytes on wire (4520 bits), 565 bytes captured (4520 bits) on interface eth1, id 0
Ethernet II, Src: 02:b2:8e:b0:79:04 (02:b2:8e:b0:79:04), Dst: 02:7f:1a:ad:9e:35 (02:7f:1a:ad:9e:35)
Internet Protocol Version 4, Src: 10.10.17.1, Dst: 10.10.17.2
Transmission Control Protocol, Src Port: 179, Dst Port: 36886, Seq: 39, Ack: 39, Len: 499
Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 59
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 36
  Withdrawn Routes
    10.10.5.0/24
    10.10.6.0/24
    10.10.7.0/24
    10.10.8.0/24
    10.10.13.0/24
    10.10.14.0/24
    10.10.15.0/24
    10.10.16.0/24
    10.10.18.0/24
    Total Path Attribute Length: 0
Border Gateway Protocol - UPDATE Message
Border Gateway Protocol - UPDATE Message
Border Gateway Protocol - UPDATE Message
Border Gateway Protocol - UPDATE Message
Border Gateway Protocol - UPDATE Message
Border Gateway Protocol - UPDATE Message
Border Gateway Protocol - UPDATE Message
```



MR-MTP updates – add remove a port against a VID

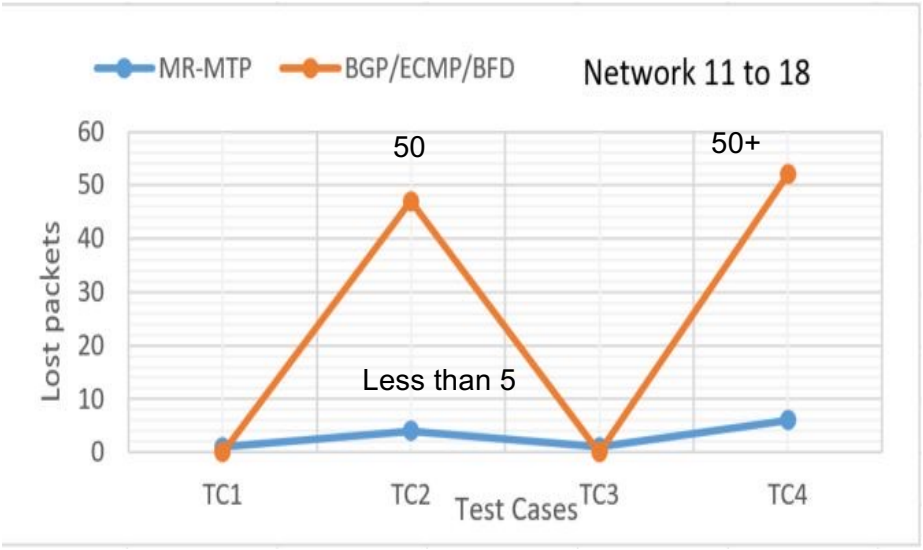
BGP/ECMP/BFD control overhead (**upto 5000 bytes**)

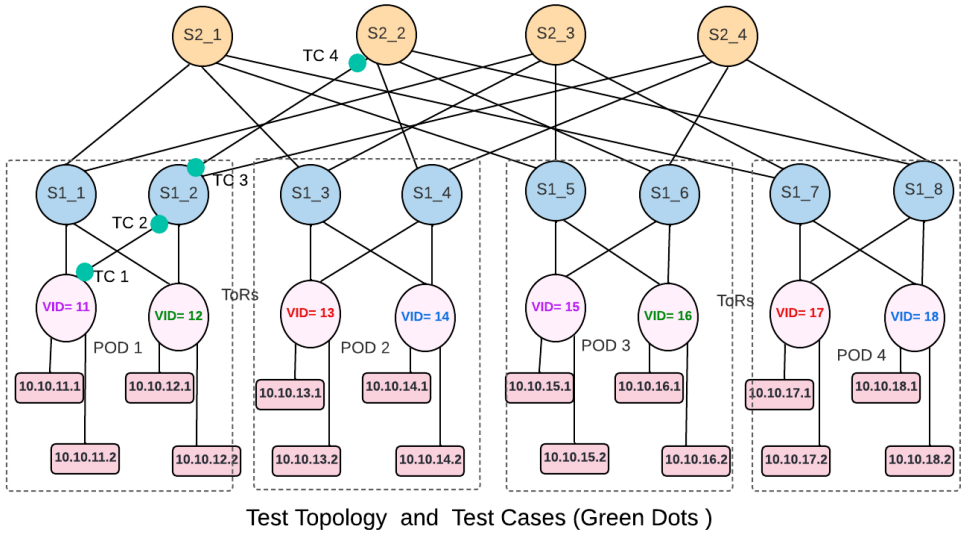
MR-MTP – control overhead (**below 300 bytes)**

MR-MTP is more stable

NANOG™

Packet Loss – Traffic from Network 11-18



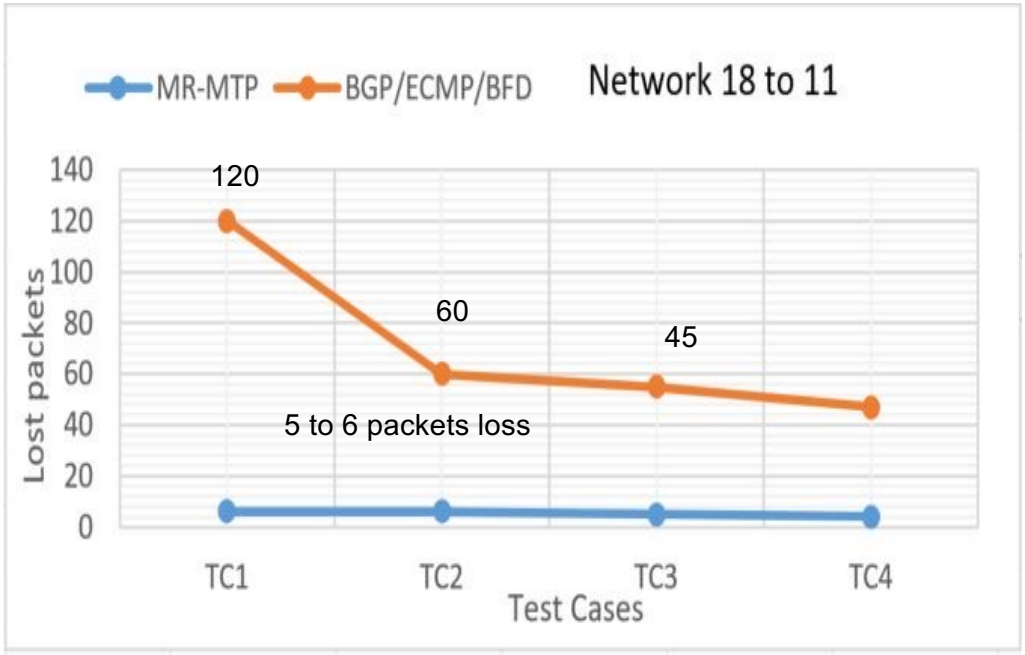


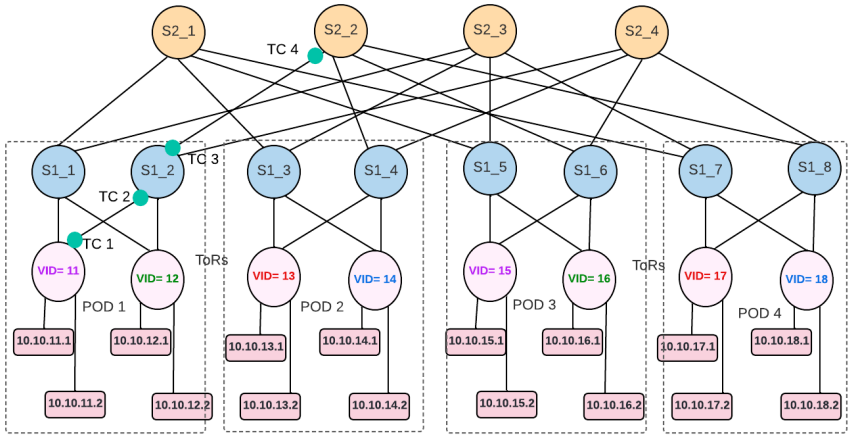
Test Topology and Test Cases (Green Dots)

On failure at TC1, TC3, BGP router flips to other interface immediately.

MR-MTP – code in user space (no link layer failure detection)
BGP/ECMP/BFD – kernel space

Packet Loss – Traffic from Network 18 - 11

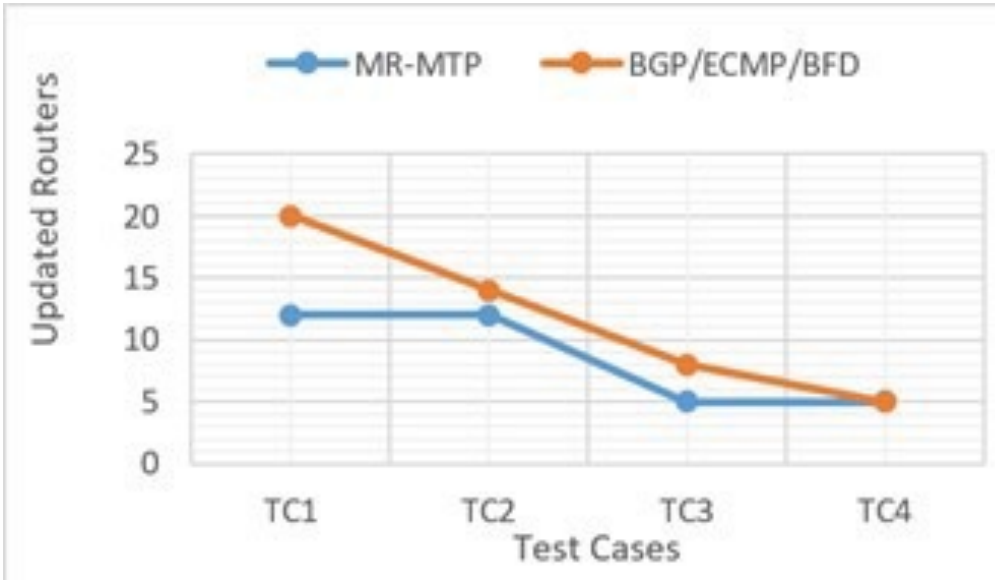


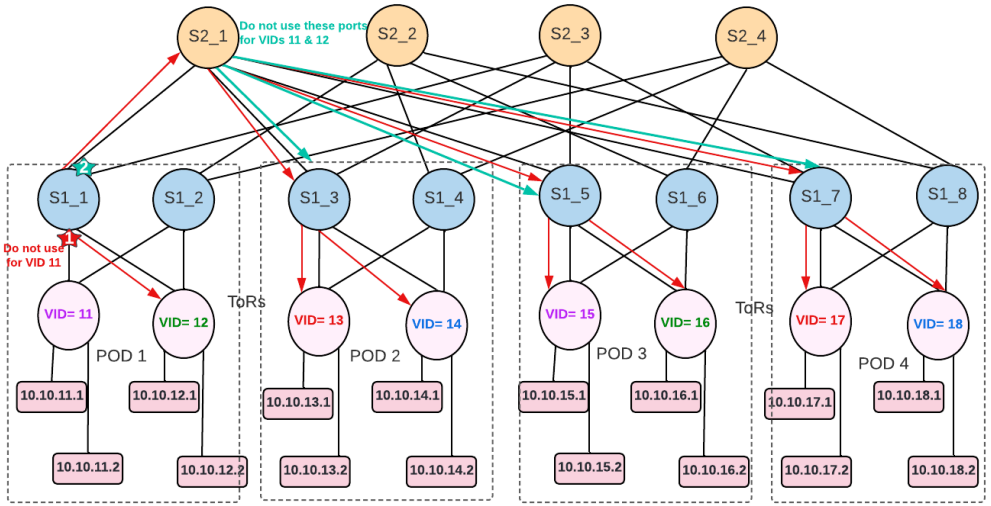


Test Topology and Test Cases (Green Dots)

NANOG™

Blast Radius – Routers Updating Routing Tables





NANOG™

Future work

- Scale the folded clos topology to multiple spine tiers - Mininet
- Tuning of timers
- Extended failure test cases
- More traffic in the network
- Use an algorithm that can be seeded to generate ToR VIDs – secure
- Encrypt IP packets originating from servers.
- Security – no BGP, TCP, IP
- Impact on energy consumption and carbon footprint, CPU and memory utilization
- Impact on cost and investment
- Interested in Collaborations - security, sustainability, economics benefits

Demo on FABRIC

FABRIC testbed

- FABRIC is Adaptive ProgrammaBle Research Infrastructure for Computer Science and Science Applications
- FABRIC is an International infrastructure that enables cutting-edge experimentation and research at-scale in the areas of networking, cybersecurity, distributed computing, storage, virtual reality, 5G, machine learning, and science applications.
- The FABRIC infrastructure is a distributed set of equipment at commercial collocation spaces, national labs and campuses.

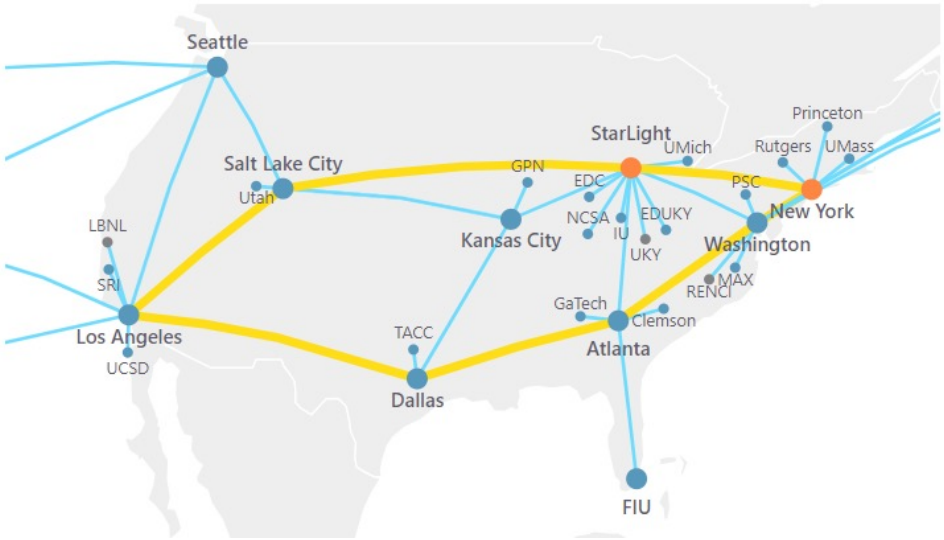
FABRIC testbed

- Each of the 29 FABRIC sites has large amounts of compute and storage, interconnected by high speed, dedicated optical links. It also connects to specialized testbeds (5G/IoT PAWR, NSF Clouds), the Internet and high-performance computing facilities to create a rich environment for a wide variety of experimental activities.

- FABRIC Across Borders (FAB) extends the network to 4 additional nodes in Asia and Europe.

Demo set up in FABRIC

- Fabric test bed - portal.fabric-testbed.net
- Reserve VM - connect
- Use desired OS
- Upload code
- Run the experiment
- Collect results
- Analyze



NANOG™

Demo on Fabric

- Results presented – GENI testbed (ww.geni.net)
- Moved experiments to FABRIC testbed

- Demo of BGP/ECMP/BFD from FRrouting and MR-MTP ready – takes time
- Used BGP for DCN as per RFC 7938
- Interested – please check with me

From Fabric Experiments– Configuration

BGP: SHOW RUNNING CONF

```
frr version 10.0
frr defaults datacenter
hostname T-1
log file /var/log/frr/bgpd.log
log timestamp precision 3
no ipv6 forwarding
debug bgp updates in
debug bgp updates out
debug bgp updates detail
router bgp 64512
timers bgp 1 3
neighbor 172.16.0.2 remote-as 64513
neighbor 172.16.0.2 bfd
neighbor 172.16.1.2 remote-as 64514
neighbor 172.16.1.2 bfd
neighbor 172.16.2.2 remote-as 64515
neighbor 172.16.2.2 bfd
neighbor 172.16.3.2 remote-as 64516
neighbor 172.16.3.2 bfd
bfd
profile lowerIntervals
transmit-interval 100
peer 172.16.0.2
profile lowerIntervals
peer 172.16.1.2
profile lowerIntervals
peer 172.16.2.2
profile lowerIntervals
peer 172.16.3.2
profile lowerIntervals
```

BGP configuration at ONE router

```
topology: {
   leaves: [L-1-1,L-1-2,L-2-1,L-2-2,L-3-1,L-3-2,L-4-1,L-4-
2],
                leavesNetworkPortDict:
                {
                 L-1-1 :  eth3,
                 L-1-2 :  eth3,
                 L-2-1 :  eth3,
                 L-2-2 :  eth3,
                 L-3-1 :  eth1,
                 L-3-2 :  eth3,
                 L-4-1 :  eth3,
                 L-4-2 :  eth2
                },
   topSpines : [ T-1 , T-2 , T-3 , T-4 ],
    pods : [
          topSpines : [ S-1-1 , S-1-2 ]
          topSpines : [ S-2-1 , S-2-2 ]
          topSpines : [ S-3-1 , S-3-2 ]
          topSpines : [ S-4-1 , S-4-2 ]
          ]
   }
}
```

MR-MTP 4-POD configuration file – all routers

From FABRIC Testbed – Routing Tables

T-1 Routing table

10.30.0.0/19 dev eth0 proto kernel scope link src 10.30.8.203 metric 100
169.254.169.254 via 10.30.6.11 dev eth0 proto dhcp src 10.30.8.203 metric 100
172.16.0.0/24 dev eth4 proto kernel scope link src 172.16.0.1
172.16.1.0/24 dev eth2 proto kernel scope link src 172.16.1.1
172.16.2.0/24 dev eth3 proto kernel scope link src 172.16.2.1
172.16.3.0/24 dev eth1 proto kernel scope link src 172.16.3.1
192.168.0.0/24 via 172.16.0.2 dev eth4 proto bgp metric 20
192.168.1.0/24 via 172.16.0.2 dev eth4 proto bgp metric 20
192.168.2.0/24 via 172.16.1.2 dev eth2 proto bgp metric 20
192.168.3.0/24 via 172.16.1.2 dev eth2 proto bgp metric 20
192.168.4.0/24 via 172.16.2.2 dev eth3 proto bgp metric 20
192.168.5.0/24 via 172.16.2.2 dev eth3 proto bgp metric 20
192.168.6.0/24 via 172.16.3.2 dev eth1 proto bgp metric 20
192.168.7.0/24 via 172.16.3.2 dev eth1 proto bgp metric 20

VID table at T-1

eth1 33.1.1, 34.1.1
eth2 35.1.1, 36.1.1
eth3 37.1.1, 38.1.1
eth4 39.1.1, 40.1.1

S-1-1 Routing Table

10.30.0.0/19 dev eth0 proto kernel scope link src 10.30.6.239 metric 100
169.254.169.254 via 10.30.6.11 dev eth0 proto dhcp src 10.30.6.239 metric 100
172.16.0.0/24 dev eth3 proto kernel scope link src 172.16.0.2
172.16.8.0/24 dev eth4 proto kernel scope link src 172.16.8.2
172.16.16.0/24 dev eth2 proto kernel scope link src 172.16.16.1
172.16.17.0/24 dev eth1 proto kernel scope link src 172.16.17.1
192.168.0.0/24 via 172.16.16.2 dev eth2 proto bgp metric 20
192.168.1.0/24 via 172.16.17.2 dev eth1 proto bgp metric 20
192.168.2.0/24 proto bgp metric 20
nexthop via 172.16.0.1 dev eth3 weight 1
nexthop via 172.16.8.1 dev eth4 weight 1
192.168.3.0/24 proto bgp metric 20
nexthop via 172.16.0.1 dev eth3 weight 1
nexthop via 172.16.8.1 dev eth4 weight 1
192.168.4.0/24 proto bgp metric 20
nexthop via 172.16.0.1 dev eth3 weight 1
nexthop via 172.16.8.1 dev eth4 weight 1
192.168.5.0/24 proto bgp metric 20
nexthop via 172.16.0.1 dev eth3 weight 1
nexthop via 172.16.8.1 dev eth4 weight 1
192.168.6.0/24 proto bgp metric 20
nexthop via 172.16.0.1 dev eth3 weight 1
nexthop via 172.16.8.1 dev eth4 weight 1
192.168.7.0/24 proto bgp metric 20
nexthop via 172.16.0.1 dev eth3 weight 1
nexthop via 172.16.8.1 dev eth4 weight 1

Takeaway

- Do we need routing protocols?
- Simple techniques can automatically establish paths.
- Benefits of auto-configuration and auto address assignment
- Non-IP based solutions can be very efficient and be backward compatible with IP and Ethernet.
- Better ways to cut down on costs – energy, equipment and maintenance
- No BGP, TCP, IP – improves security

Thank you

Contact: nxsvks@rit.edu

NANOG™