Meshed Tree Routing in Folded-Clos Topologies

PETER WILLIS & NIRMALA SHENOY, PH.D.

08/22/2022

Research Motivation

Data centers are growing in size and operational complexity.

- Thousands of servers.
- Variety of use-cases and applications are growing.

The Data Center Network (DCN) connects the servers so they can exchange data.

- DCNs have to be highly reliable and resilient.
- DCNs have to scale and should have high availability.

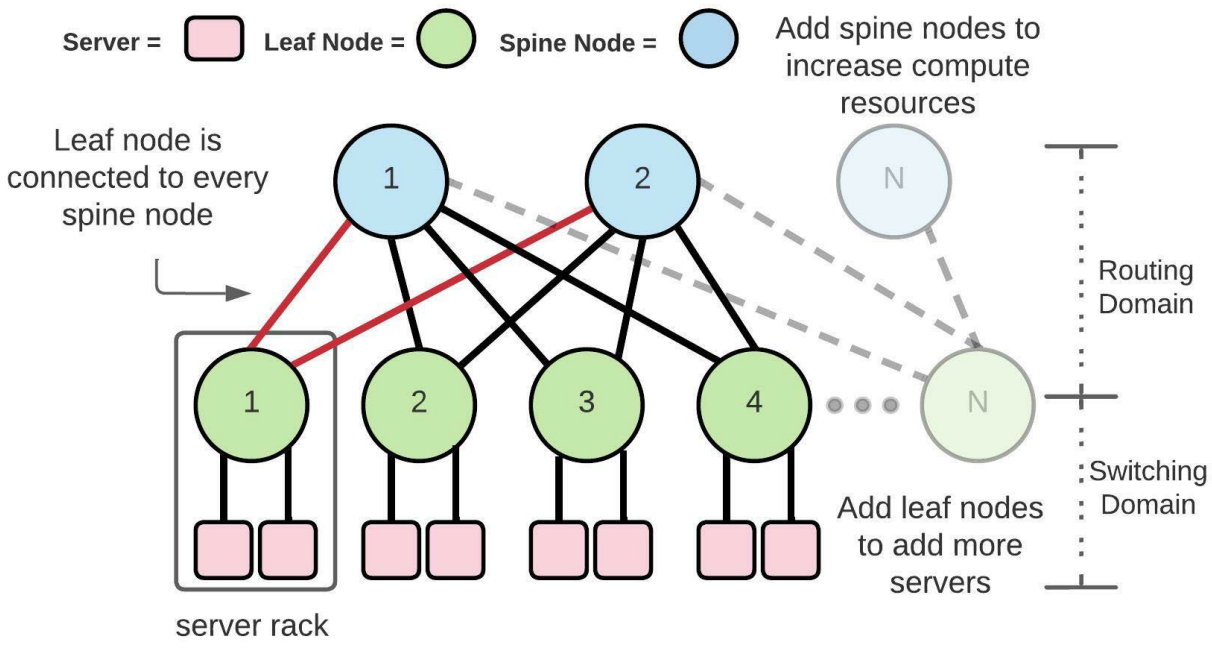
Innovation in the DCN control plane needs to keep pace with the advancement of other aspects of data centers.

- Trends in modern DCNs lends itself to optimizations.

Folded-Clos Topology

- **Folded-Clos topology is popular in DCNs and has several attractive features.**
  - High bisection bandwidth.
  - Rearrangably non-blocking.
  - Scales out via commodity hardware.
  - Equal-cost multi-path routing between servers.

Hence we used the folded-Clos topology in our project.



Existing Solutions

Utilize a Link-State Routing Protocol.

- OSPF, IS-IS.
- Flooding is problematic.

Utilize the Border Gateway Protocol (BGP).

- Path-vector based.
- Built for inter-AS routing, modified (retrofitted) for DCNs (RFC 7938).

Utilize a Topology-Aware protocol.

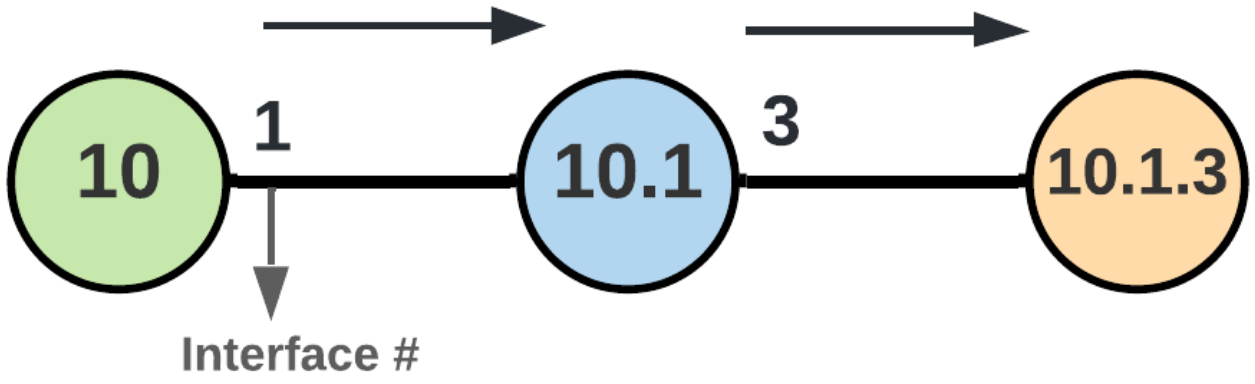
- Routing in Fat Trees (RIFT).
- Link State Vector Routing (LSVR) → Another BGP modification.

The Meshed Tree Algorithm and Protocol

A control and data plane solution designed to use the attributes of folded-Clos topologies.

Uses virtual identifiers (VID) – a simple way to build path vectors in the control plane.

Forwarding decisions in the data plane are made using VIDs, not IP addresses.

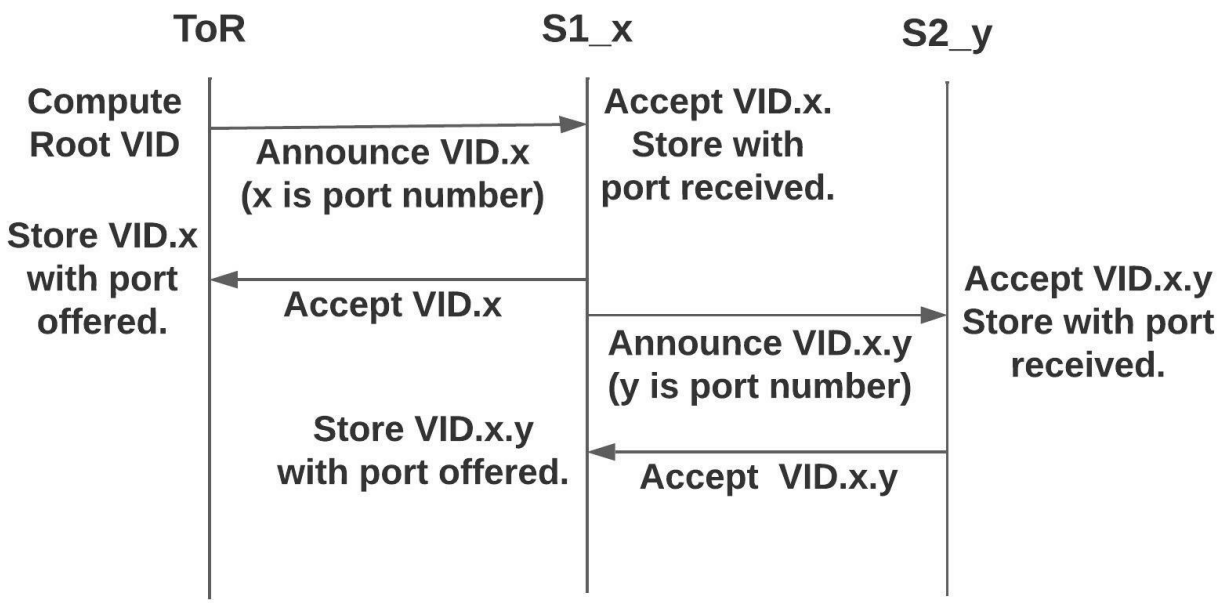


MTP Route Establishment

- **VID = Virtual Identifier.**

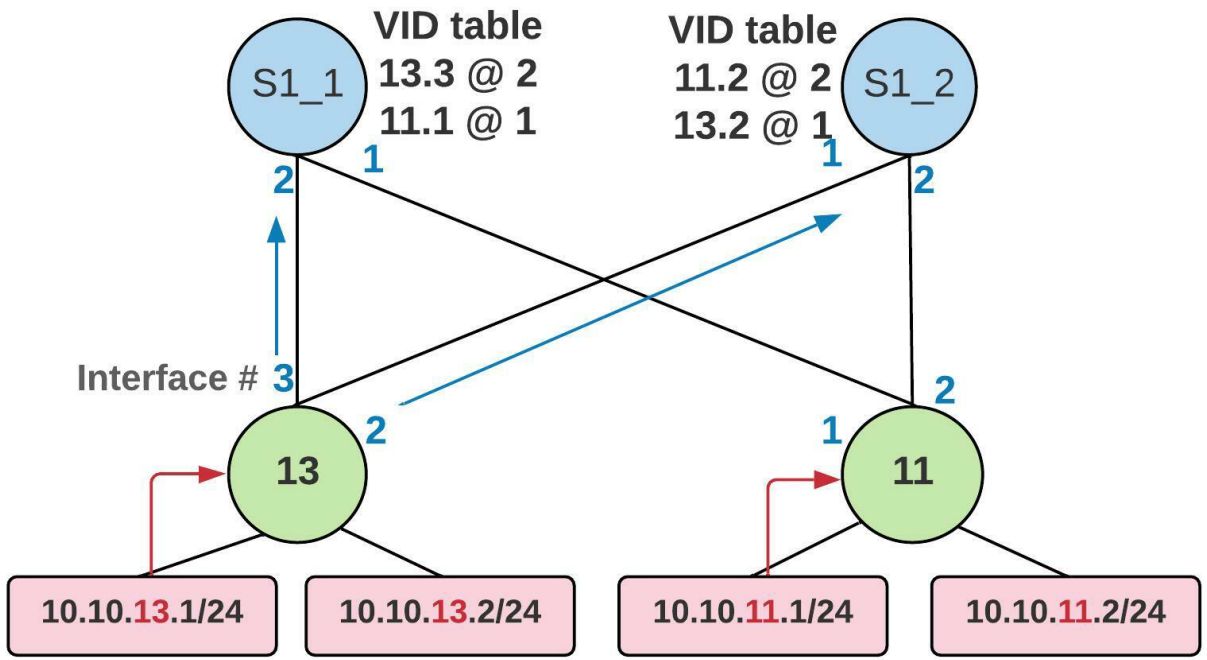
Defines a path from a leaf (ToR – the root of the meshed tree) to a spine. (one-way only).

- **Each ToR derives a root VID (can be assigned).**
- **Announcement messages carry the VID information to its neighbor spines.**
- **Spines at the highest tier are configured to stop VID propagation down the topology.**



Root (ToR) VID derivations

- **The Root VIDs can be derived from the physical subnet IP address used by servers connecting to ToR.**
- **Current method utilizes an octet of the server subnet.**
  - In this case, all server subnets are /24.
- **Other methods can be defined.**

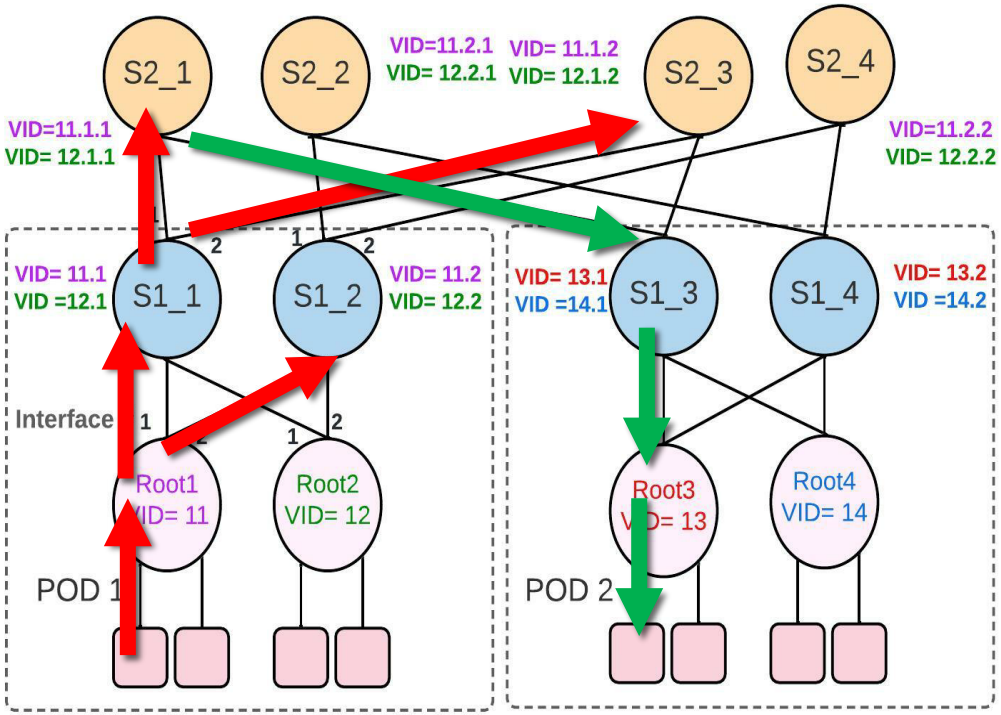


MTP Routing Tables

Root1 Forwarding Table	
Spine	**Port**
11.1	1
11.2	2

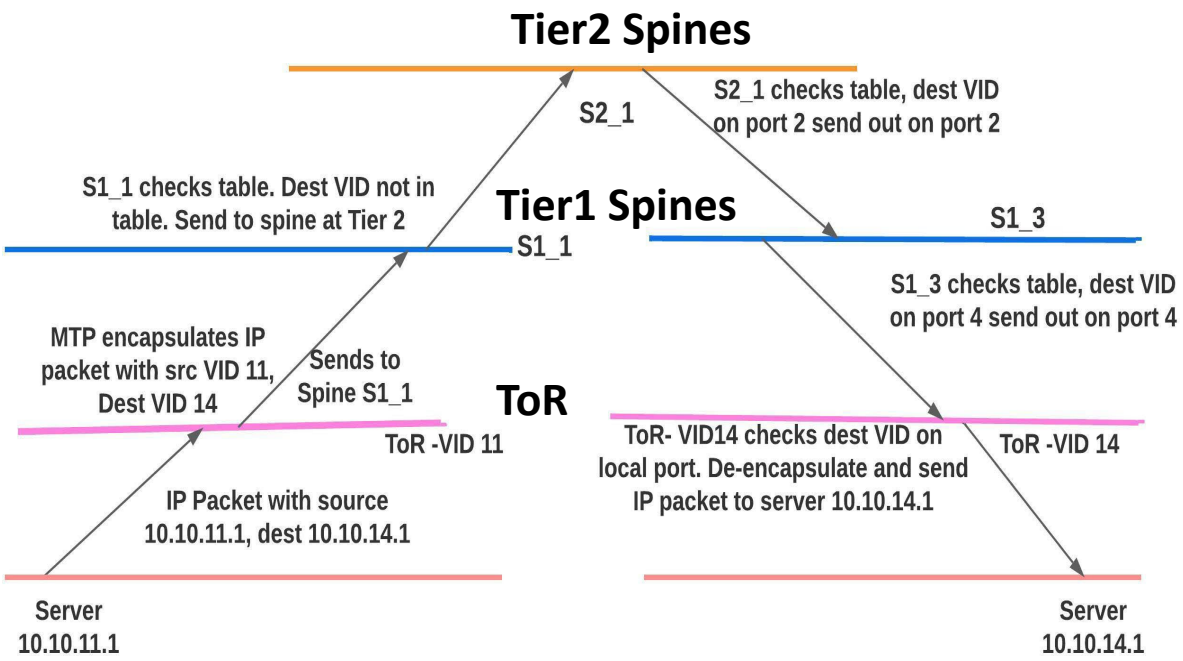
S2_1 Forwarding Table	
Downstream	**Port**
11.1.1	1
12.1.1	1
13.1.1	2
14.1.1	2

S1_1 Forwarding Table	
Upstream	**Port**
11.1.1	1
12.1.1	1
11.1.2	2
12.1.2	2
Downstream	**Port**
11.1	3
12.1	4



Example

- Source ToR- checks src-dst IP address. Derives the src-dst ToR VIDs – populates MTP header.
- Encapsulates incoming IP packet.
- Sends to any Tier 1 spine – hash algorithm.
- Tier 1 spine checks dst ToR VID in MTP header – forwards by default to Tier 2 spine – hash algorithm.
- Tier 2 spine check dst ToR VID, its VID table, sends on port – that records the dst VID.
- So on…



Comparison – Current BGP Implementation

MTP

- Route established with VIDs – no traditional routing protocols, no route discovery.
- One-way route from ToR to Spine.

Modified BGP

- BGP modified to avoid path hunting.
- ASNs adjusted.
- Route discovery flooding – periodic.

Current Status

Working MTP code in Python tested on the GENI testbed.

- Has algorithm tested.
- Check functionality.
- Demo available.

Code available on public Github repository.

C Code being developed, will integrate hashing and link aggregation.

Other features – fast failure recover, energy studies.

Seeking industry collaboration.

Thank you!