University of Rome Tor Vergata
ICT and Internet Engineering

Network and System Defense

Alessandro Pellegrini, Angelo Tulumello

A.A. 2023/2024

VXLAN and EVPN

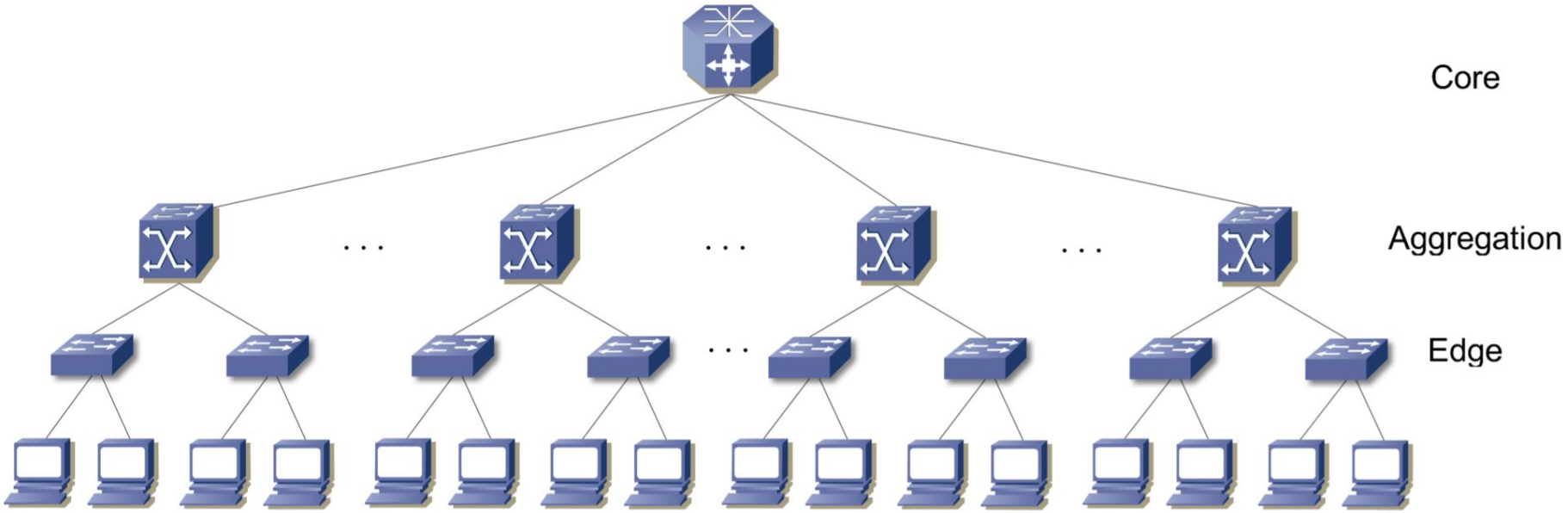
Angelo Tulumello

TOC

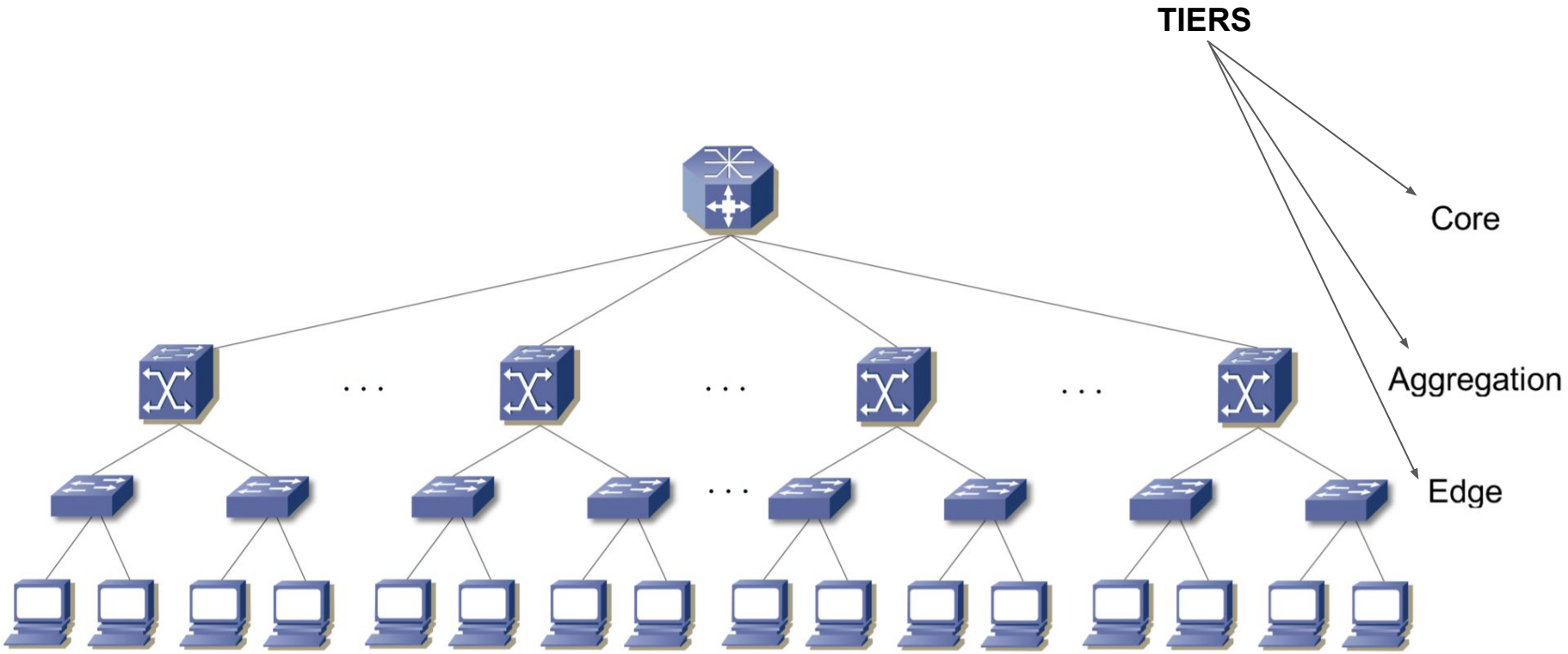
- ❏ Datacenter networks - two/three-tier topologies
- ❏ VXLAN
- ❏ LAB: configure VXLAN static tunnels
- ❏ EVPN
- ❏ LAB: configure EVPN with MP-BGP

Intro: Datacenter Networks

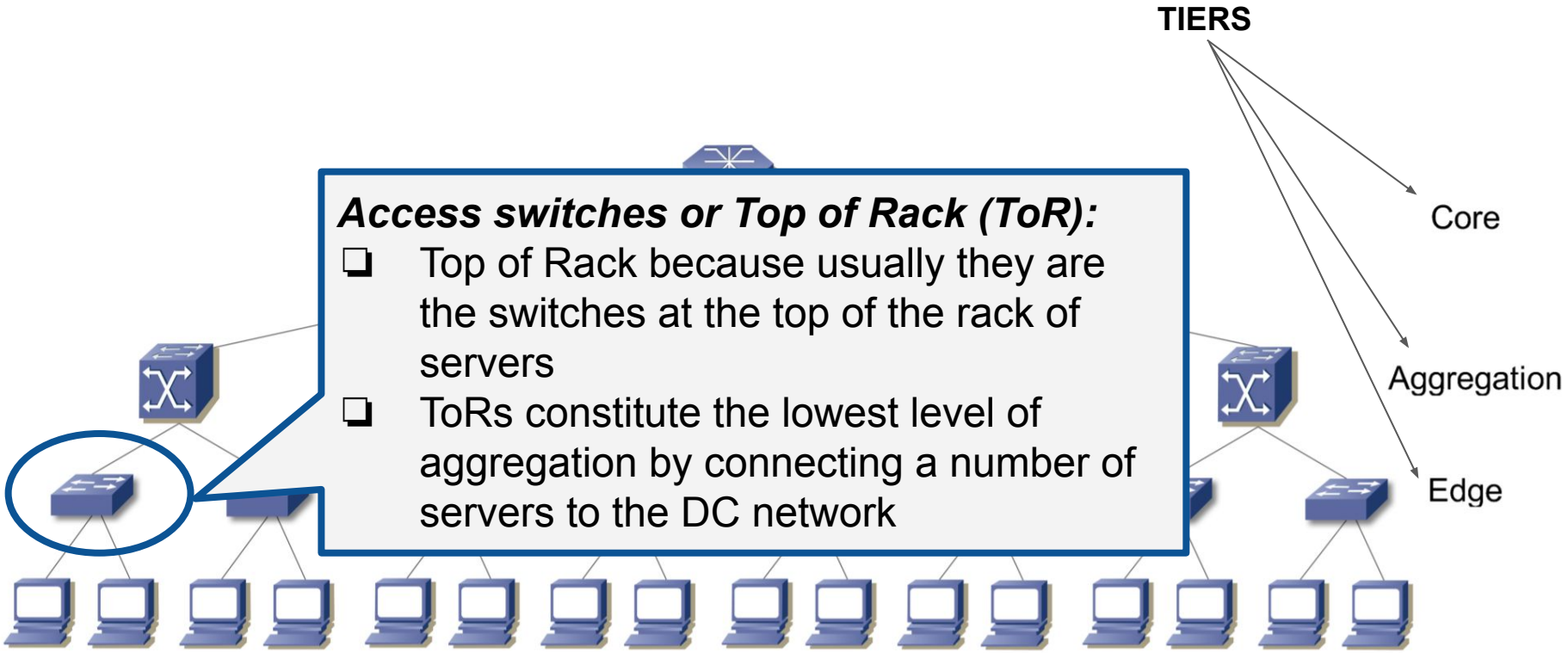
Datacenter Networks: Basic Tree



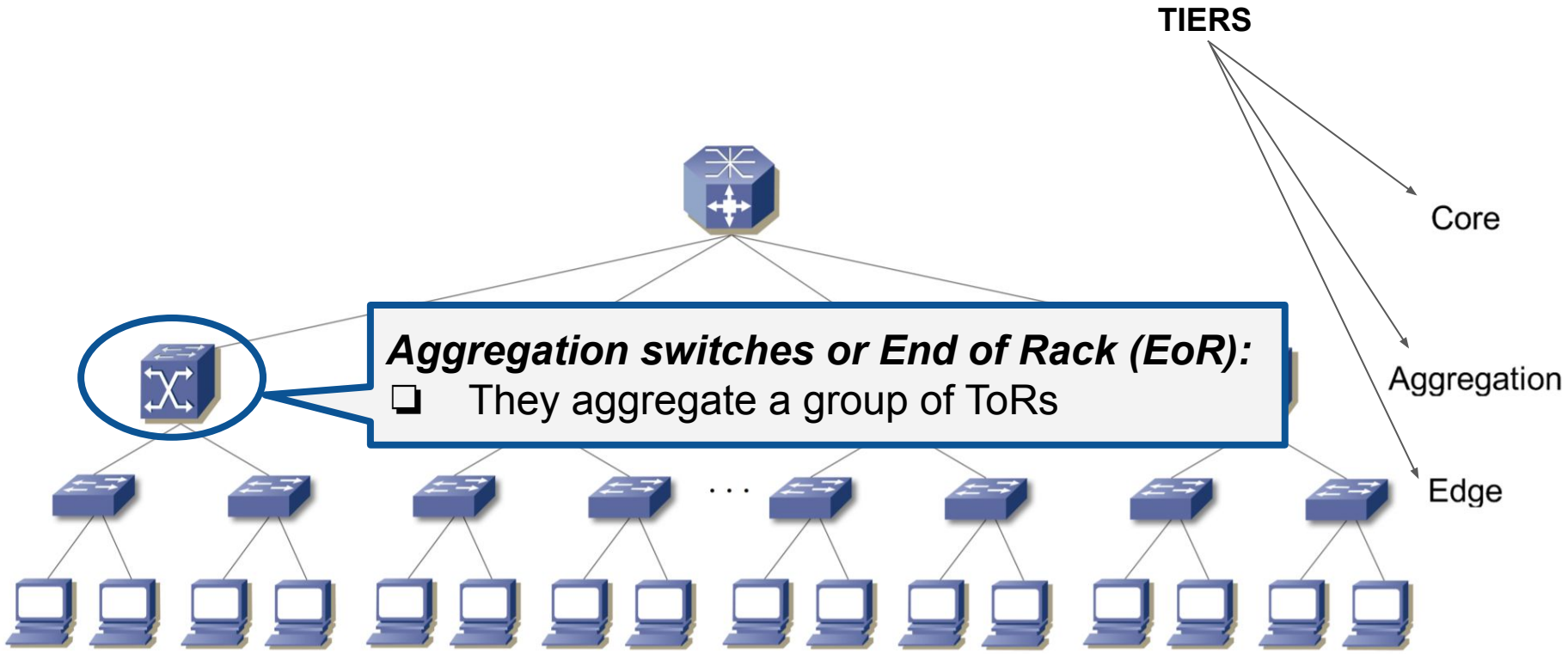
Basic Tree: tiers



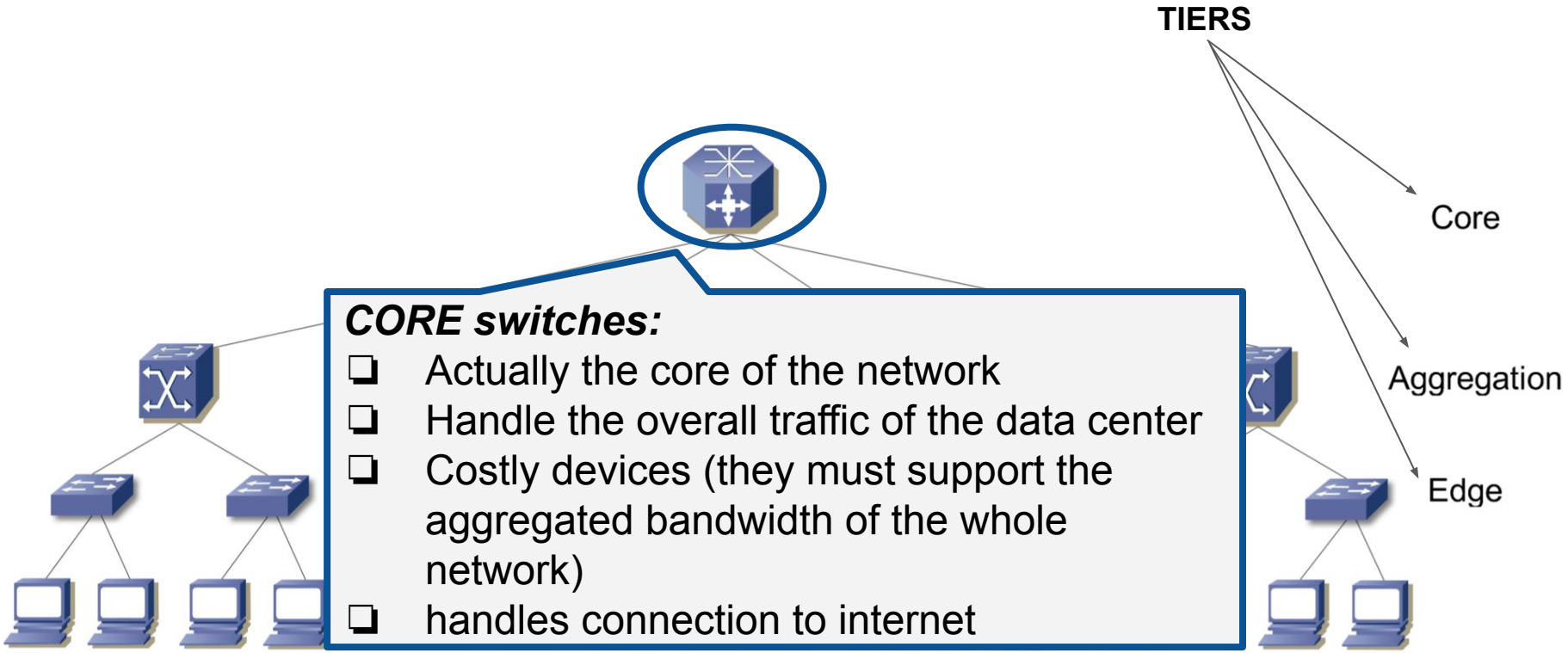
Basic Tree: ToR



Basic Tree: EoR



Basic Tree: Core

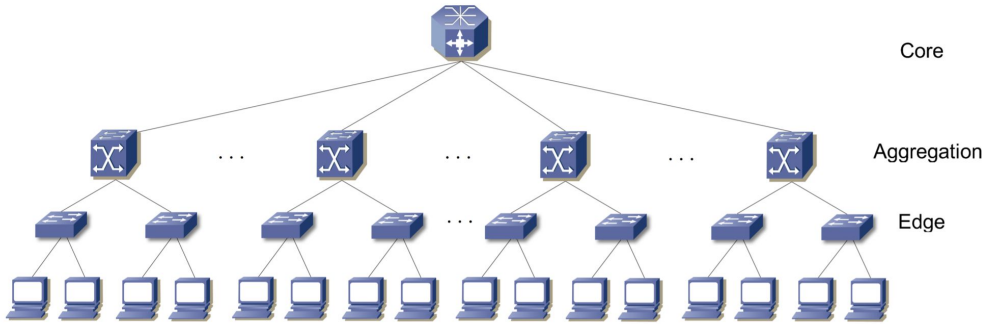


CORE switches:

- ❏ Actually the core of the network
- ❏ Handle the overall traffic of the data center
- ❏ Costly devices (they must support the aggregated bandwidth of the whole network)
- ❏ handles connection to internet

Basic Tree: Limitations

- Serious **oversubscription** over *core* network tier, i.e. the core switch(es) must support the **full** aggregated bandwidth of lower tiers
- Poor **fault tolerance** → if a node in the tree fails, all its smaller “branches” are out of service
- Higher tier switches are very **expensive**



Today's Data Center networks

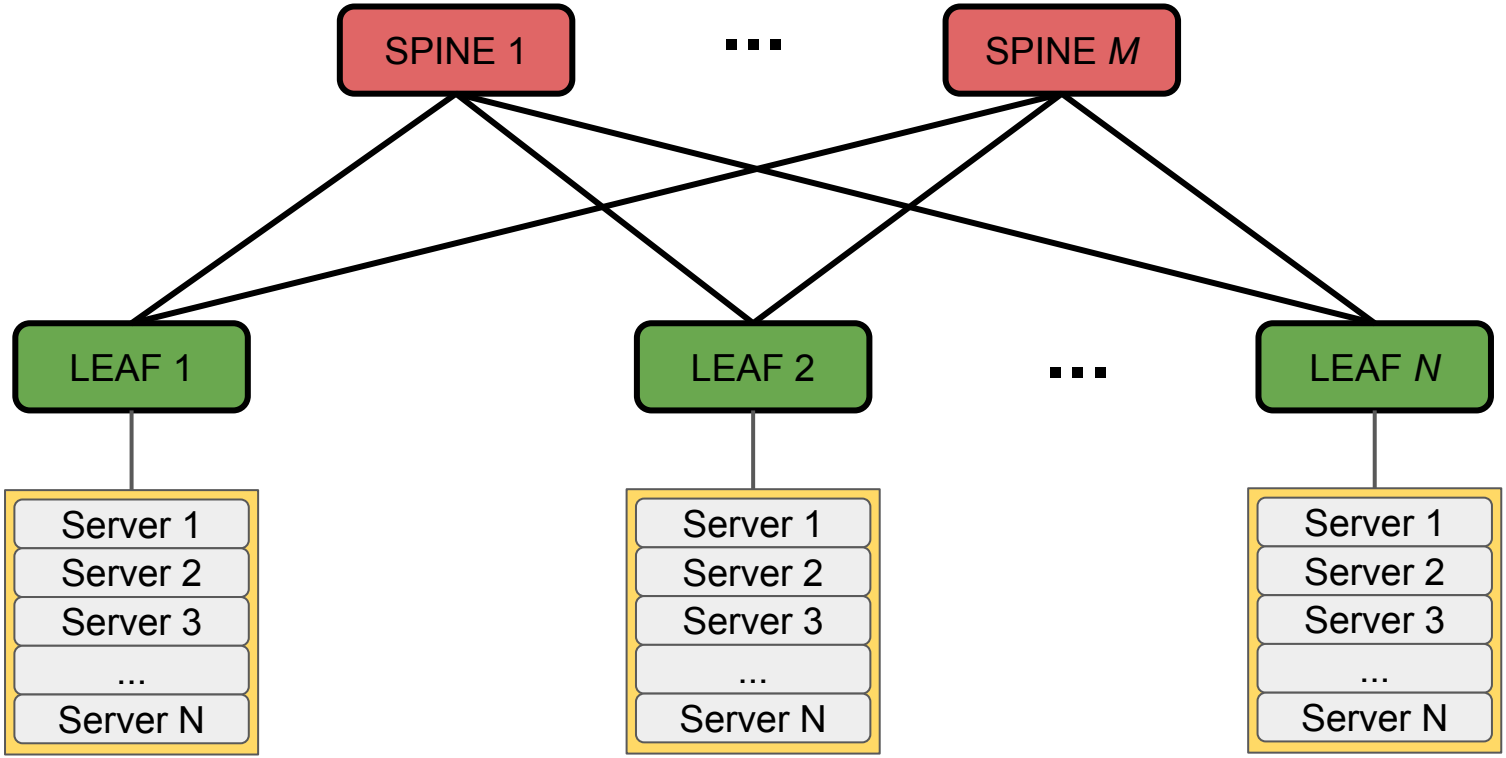
- *Key objectives:*
  - ***Scalability:*** data center networks must support incremental expansion of the compute nodes without affecting the existing architecture
  - ***Redundancy****:* a data center networking node cannot be a single point of failure. Redundancy of networking nodes at different levels to get a good level of *resiliency/fault-tolerance.*
  - ***Latency:*** latency is mostly dependent on the *number of hops* a packet must traverse to reach its destination. Need of a smaller network *diameter.*
  - ***Network capacity:*** Large scale data centers need high bandwidth (e.g. for distributed applications like memcached, network file systems, etc.)
  - ***Tenants Isolation:*** Virtualization with Cloud require *isolation* between tenants both for network performance and for security

Today's Data Center networks = Clos Topologies

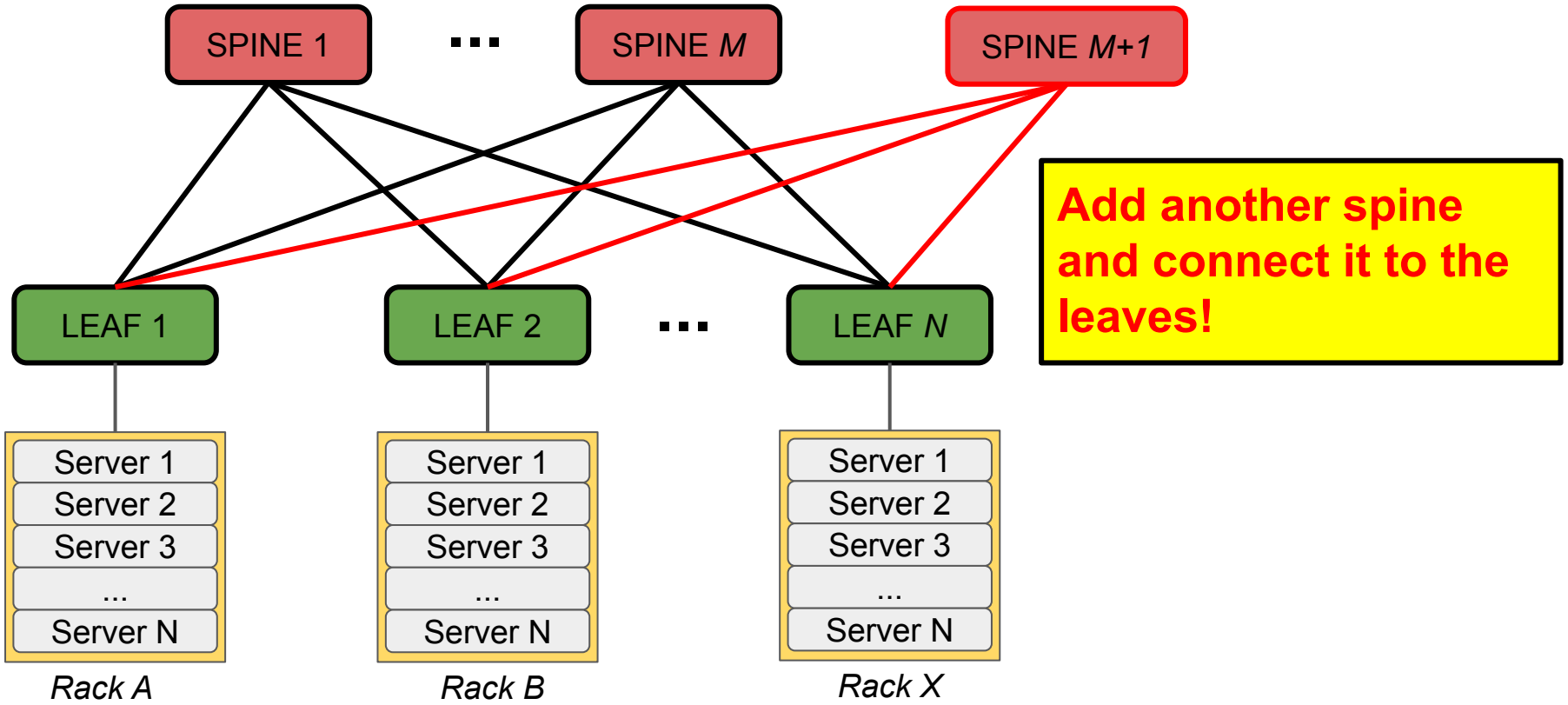
- Clos is not an acronym → formalized by Charles Clos in 1952
  - initially used in circuit switched networks for telephones
  - basic idea → many cheaper devices with redundant interconnections between each other
    - instead of bigger cross-connect switches

- Can have different levels (tiers). In DC networks usually we have:
  - **Two-tier** *(aggregation - core)* → *leaf-spine* topologies
  - **Three-tier** *(access - aggregation - core)* → *fat-tree* topologies

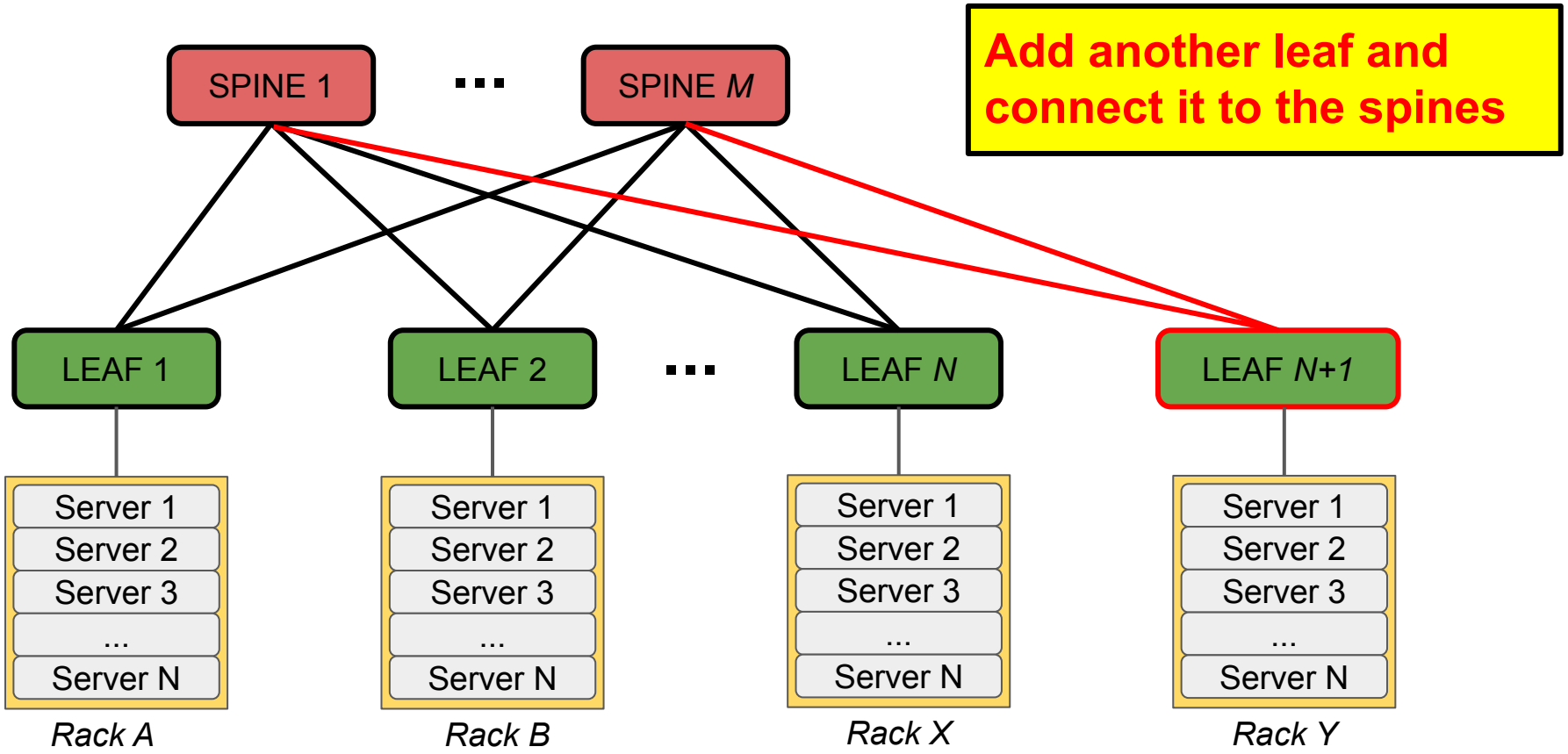
Clos topologies: two-tier leaf-spine topology



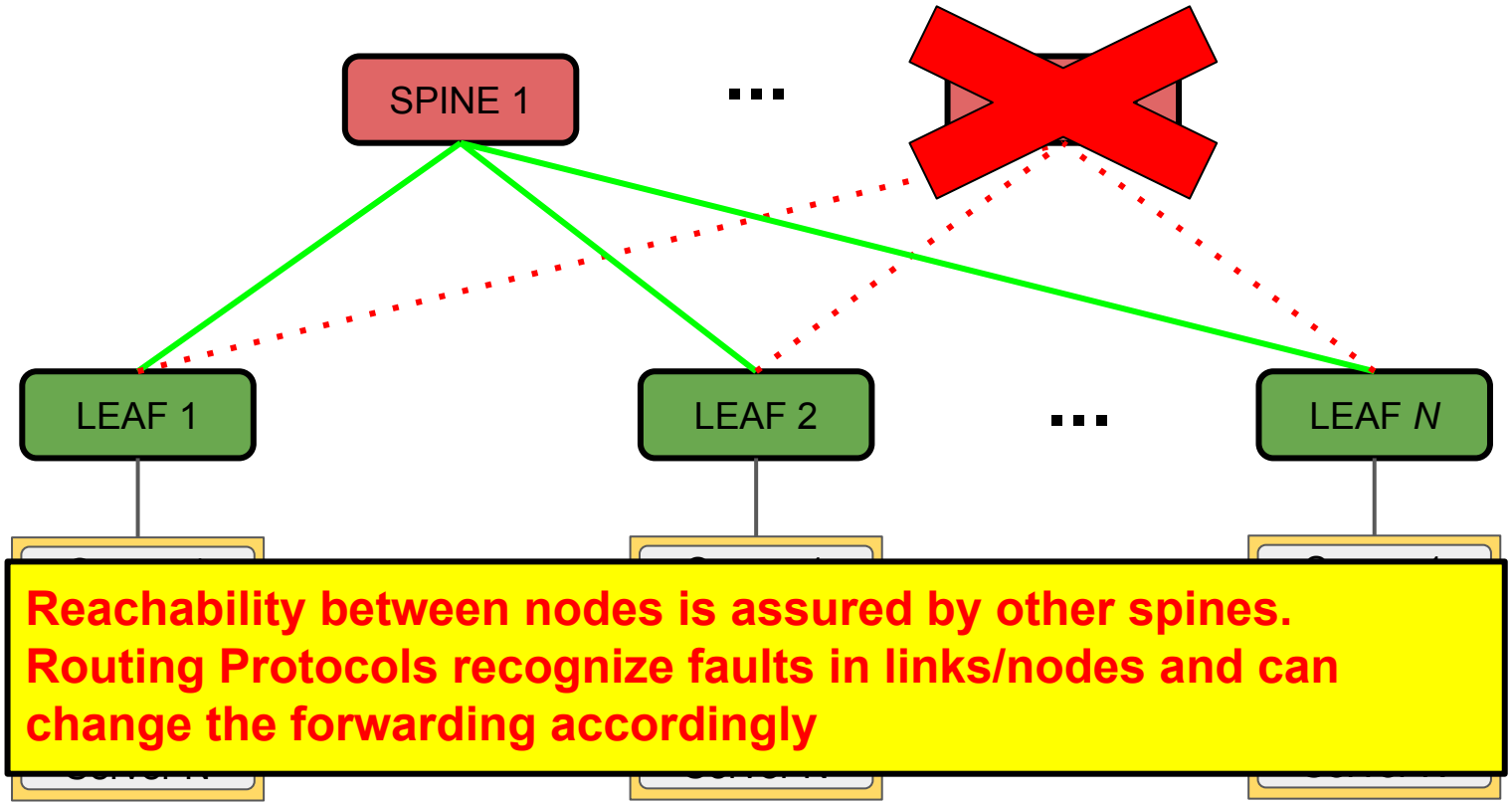
Scalability: Need more bandwidth?



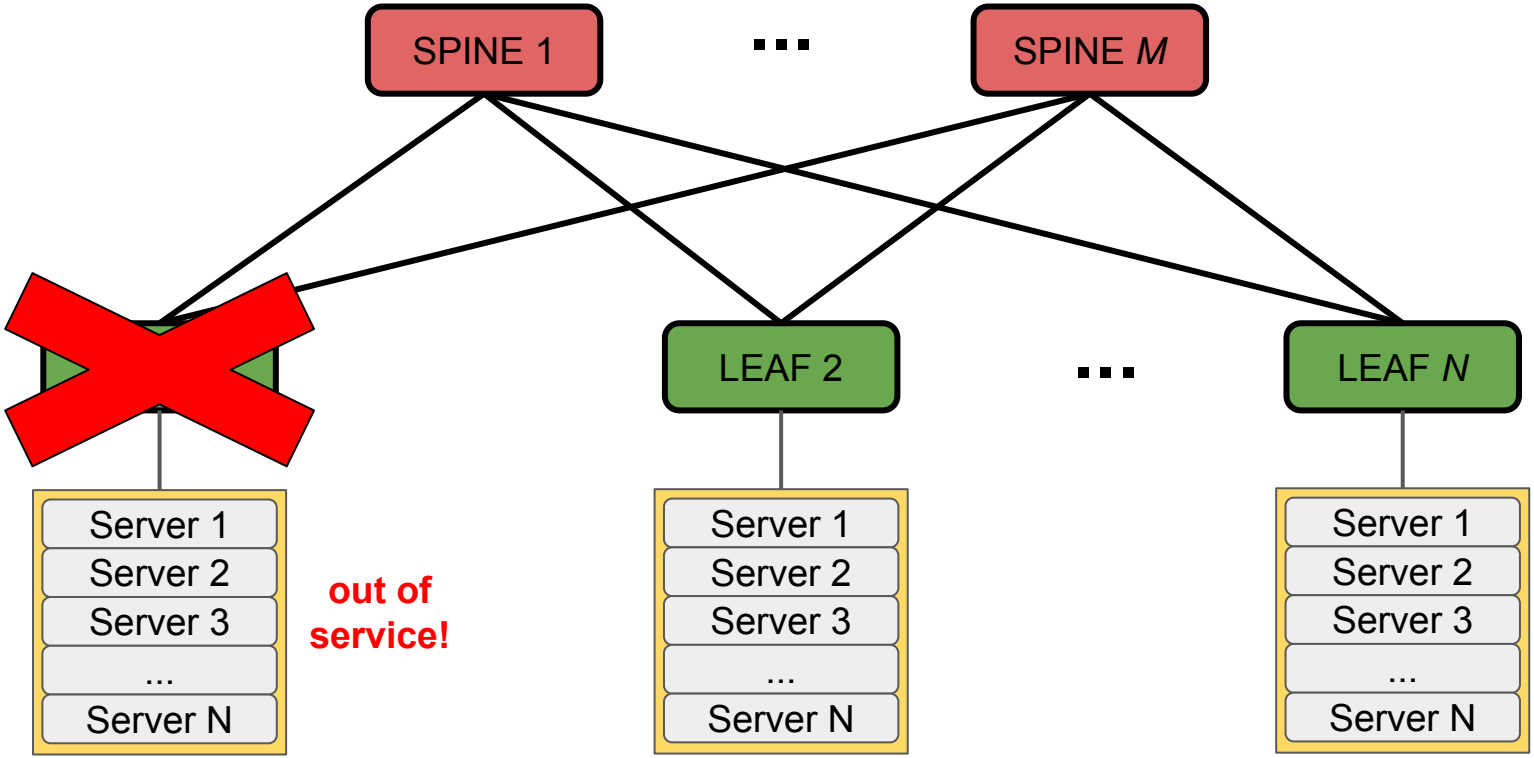
Scalability: Need to extend the compute capacity?



Fault Tolerance: Spines

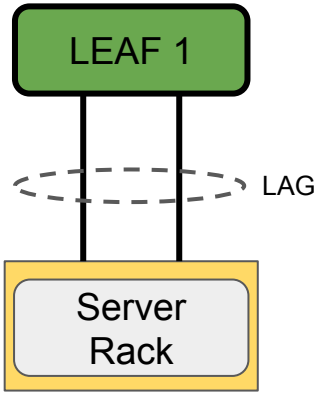


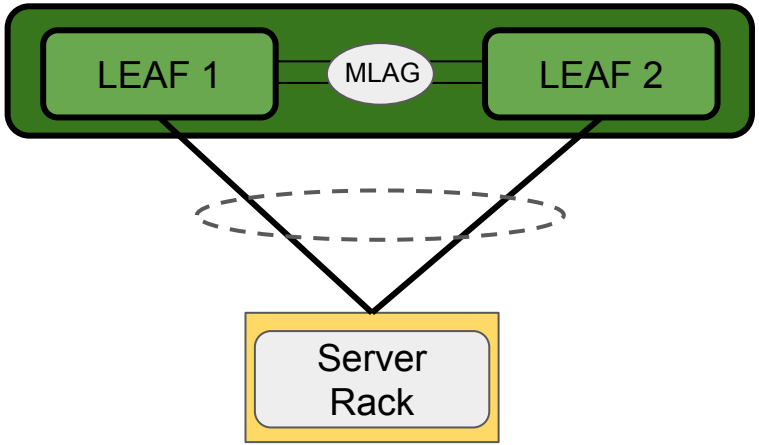
Fault tolerance: Leaves



Fault tolerance: LAG and MLAG

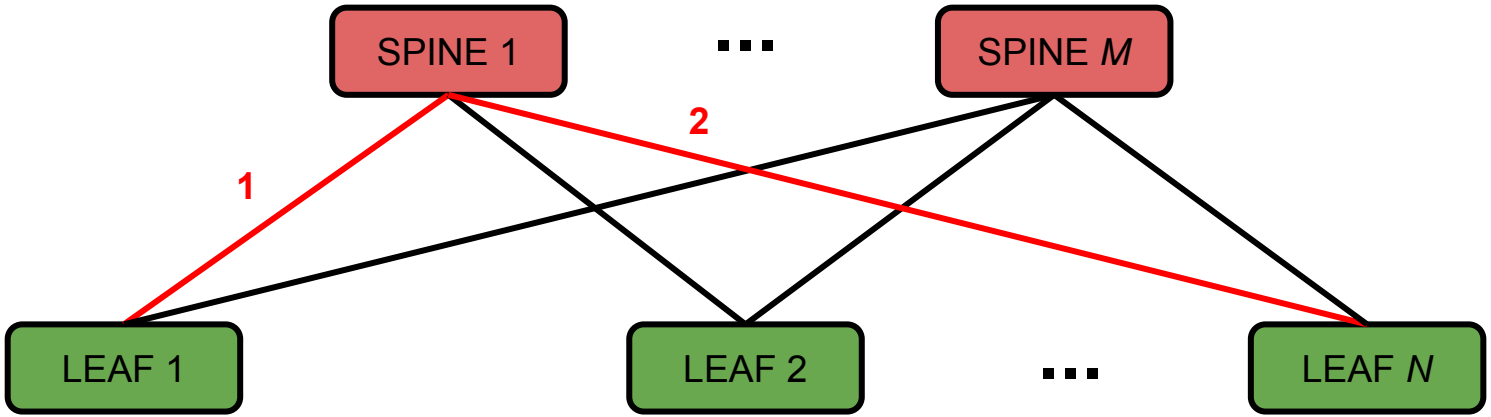
- Multi Chassis Link Aggregation Group (MLAG or MC-LAG)
- Extended version of Link Aggregation Group (LAG → IEEE 802.3ad)
  - Sort of *inverse multiplexing* technique → a group of links are aggregated both for *redundancy* and for *bandwidth multiplication* acting as **ONE** single virtual link
  - Managed by Link Aggregation Control Protocol (LACP)
- MLAG adds redundancy at a “switch” level, not only link level





Latency

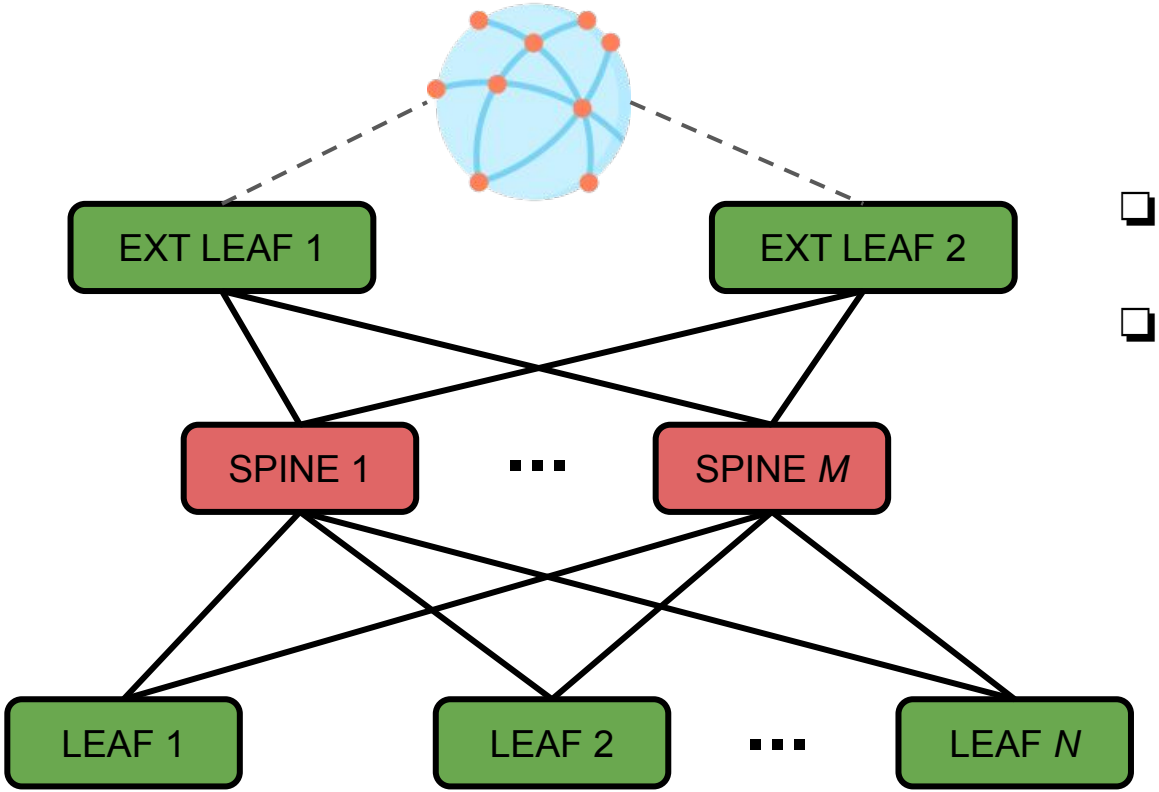
- If we ignore the transmission time of packets (time in the wire) that is negligible in a DC, the latency of a network is mostly related to the number of hops
- Network ***diameter:*** maximum number of hops of the shortest paths between two nodes in the network
- *Example*: in leaf-spine topologies the *diameter* is **2** and it is equal for every pair of end-nodes



Bandwidth

- ***Bisection bandwidth:*** split the network in half. The *bisection bandwidth* is the minimum bandwidth when half of the nodes simultaneously send at maximum link capacity to the other half of the nodes
- ***Oversubscription ratio:*** is the ratio between worst case achievable aggregate bandwidth between two end-hosts, and the bisection bandwidth
- Let N be the number of end-nodes and B bits/s the capacity of end-nodes links
- Ideal condition → *Bisection bandwidth* = N/2 * B
  - i.e. an oversubscription ratio of 1:1

Connection to the external world

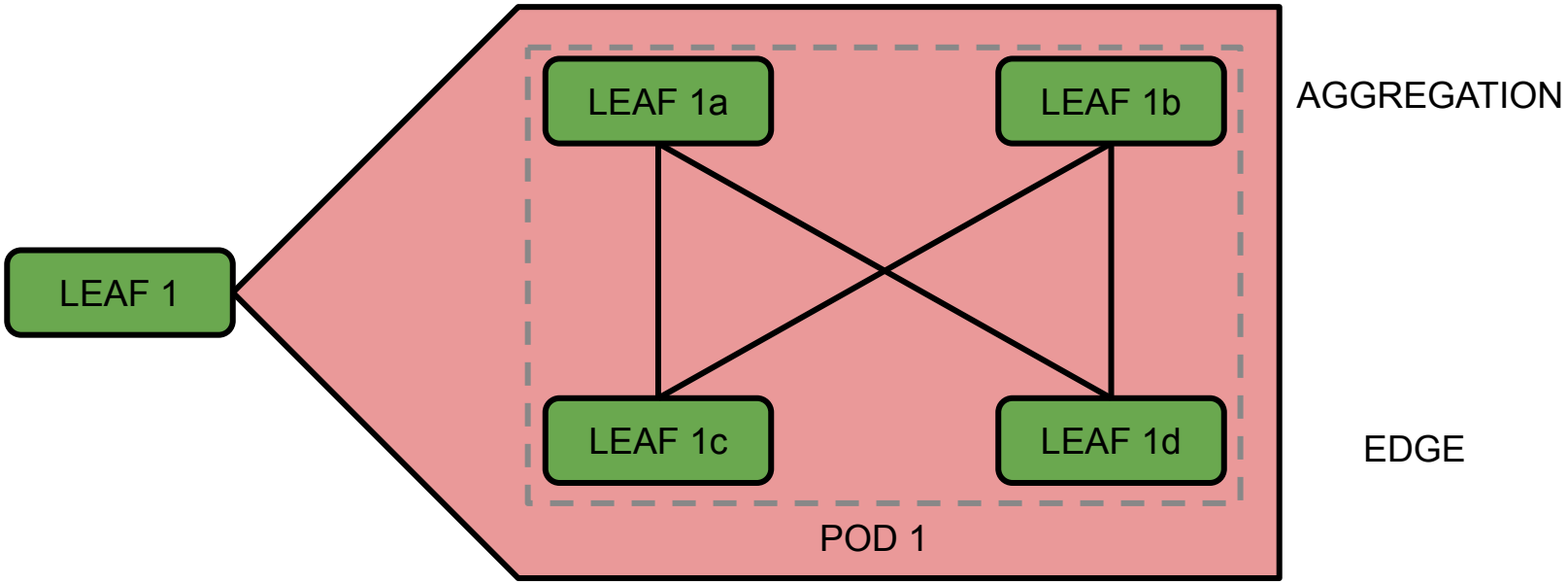


- Special leaves handle traffic from/to Internet
- Usually they are (at least) 2 for fault-tolerance

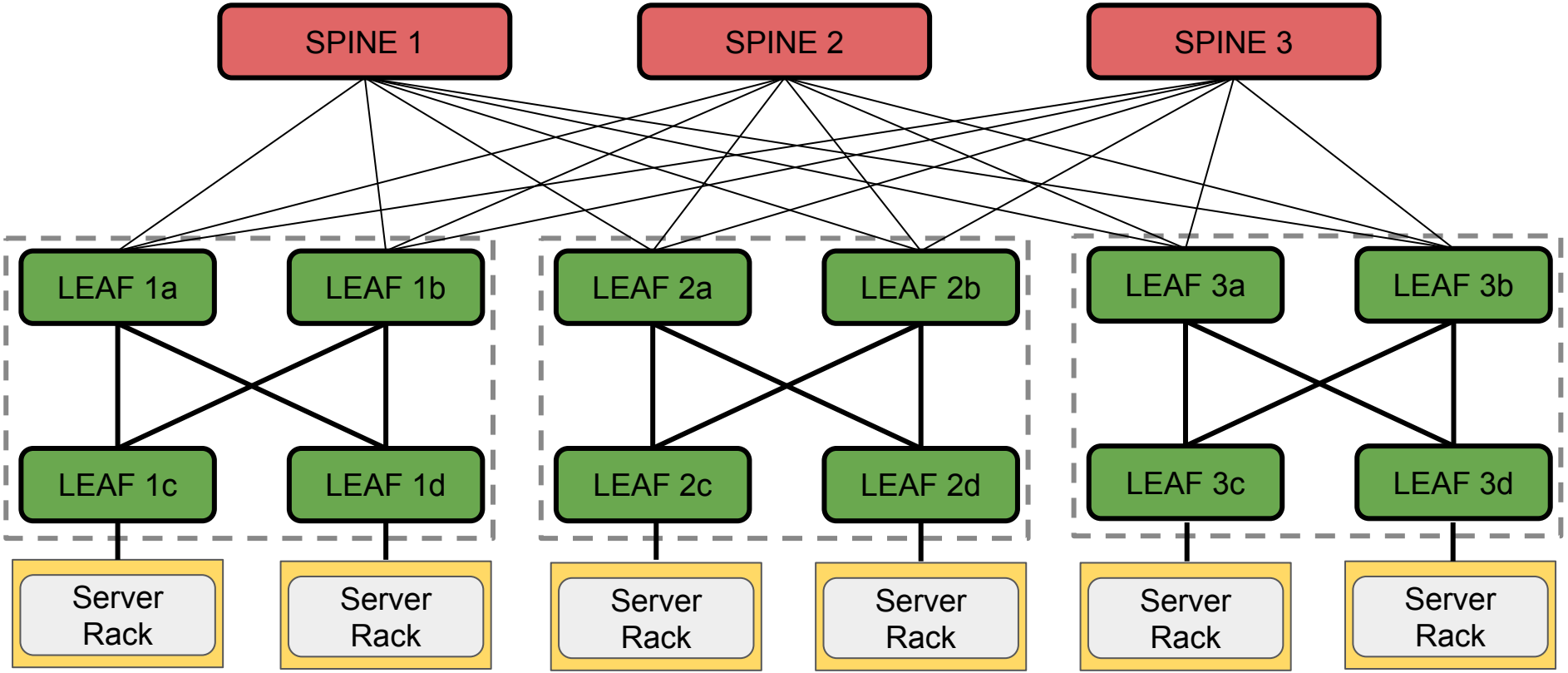
Three-tier Clos networks:
Fat Tree topology

Fat Tree Topology

- The **Fat-tree** adds one tier to the leaf-spine Clos topology
- Can be seen as a leaf exploded in four leaves → this group of 4 leaves is called **pod**



Example of Fat Tree Topology



Fat Tree Topology

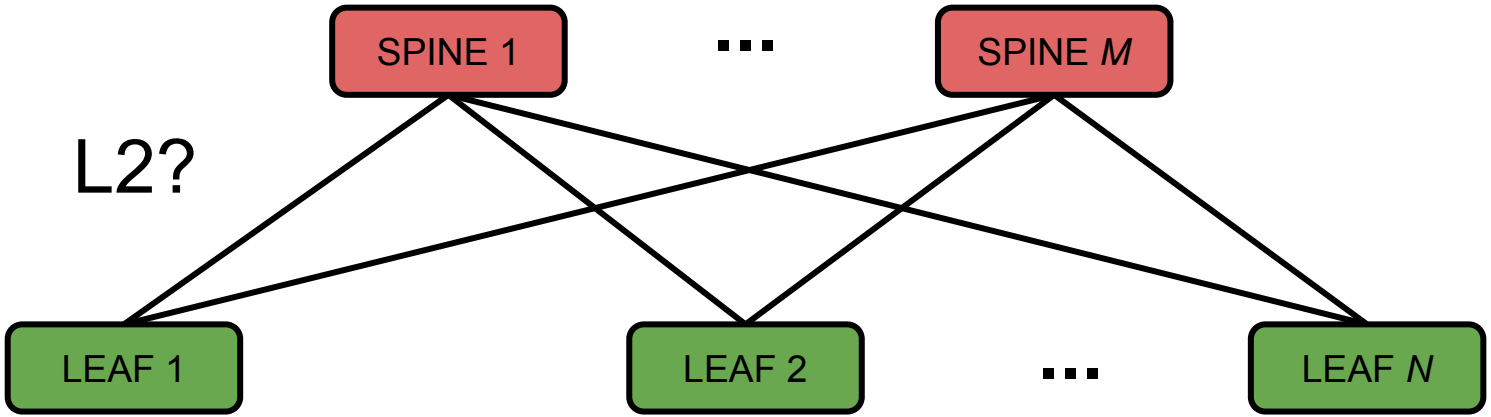
- Capable to scale up to 27k+ end-nodes with 48 ports switches
  - maintaining **full** bisection bandwidth
- Switches are *commodity* switches → low cost
- Backward compatibility with Ethernet/IP/TCP solutions

See the Fat-tree whitepaper for more information:
http://www.cs.kent.edu/~javed/class-CXNET09S/papers-CXNET-2009/FaLV08-DataCenter-interconnect-p63-alfares.pdf

Data center networks: Packet Forwarding
From L2 to VXLAN

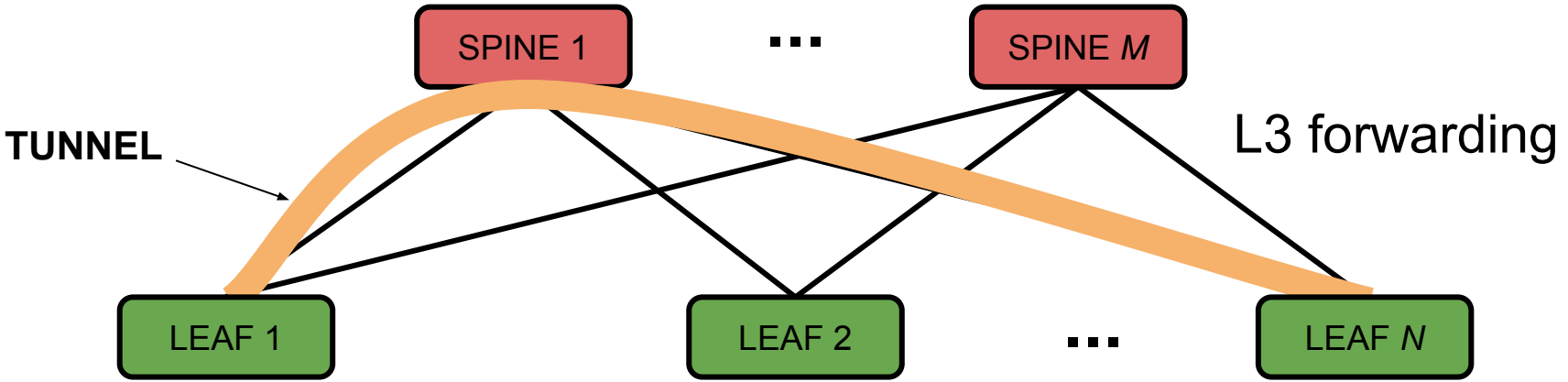
L2 forwarding in clos topologies

- L2 forwarding is not feasible in large scale datacenter topologies, for a number of reasons:
  - Spanning Tree Protocol, needed to avoid loops, selects only one output port per dest MAC → ***load balancing*** not feasible
  - ***Virtualization***: VLANs are limited → less than 4k virtual network identifiers available



Network Virtualization

- Cloud multi-tenancy requires:
  - ***Traffic isolation:*** tenants must see only packets belonging to their virtual network
  - Not only private IP subnets can be reused → also MAC addresses of VMs can be reused in different virtual networks
- These requirements imply that L2 frames must be **tunnelled** through the underlay network → in this case an L3 network

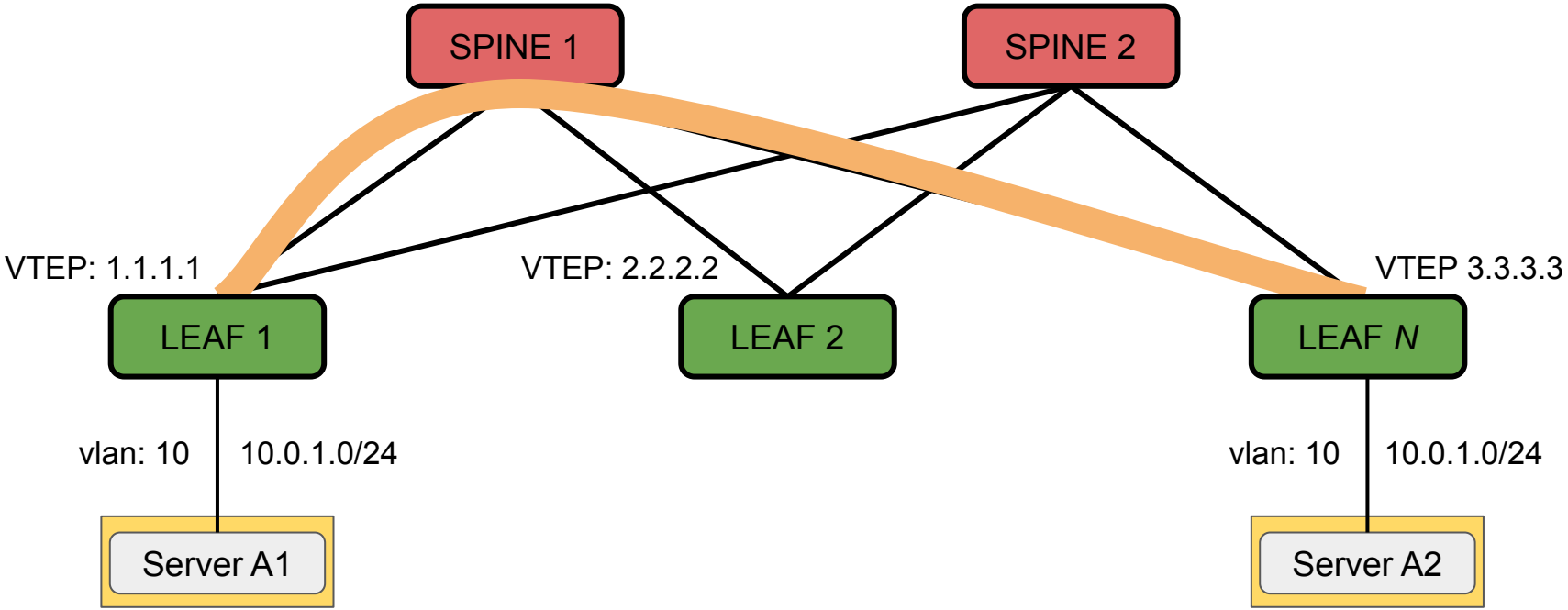


VXLAN comes to the rescue

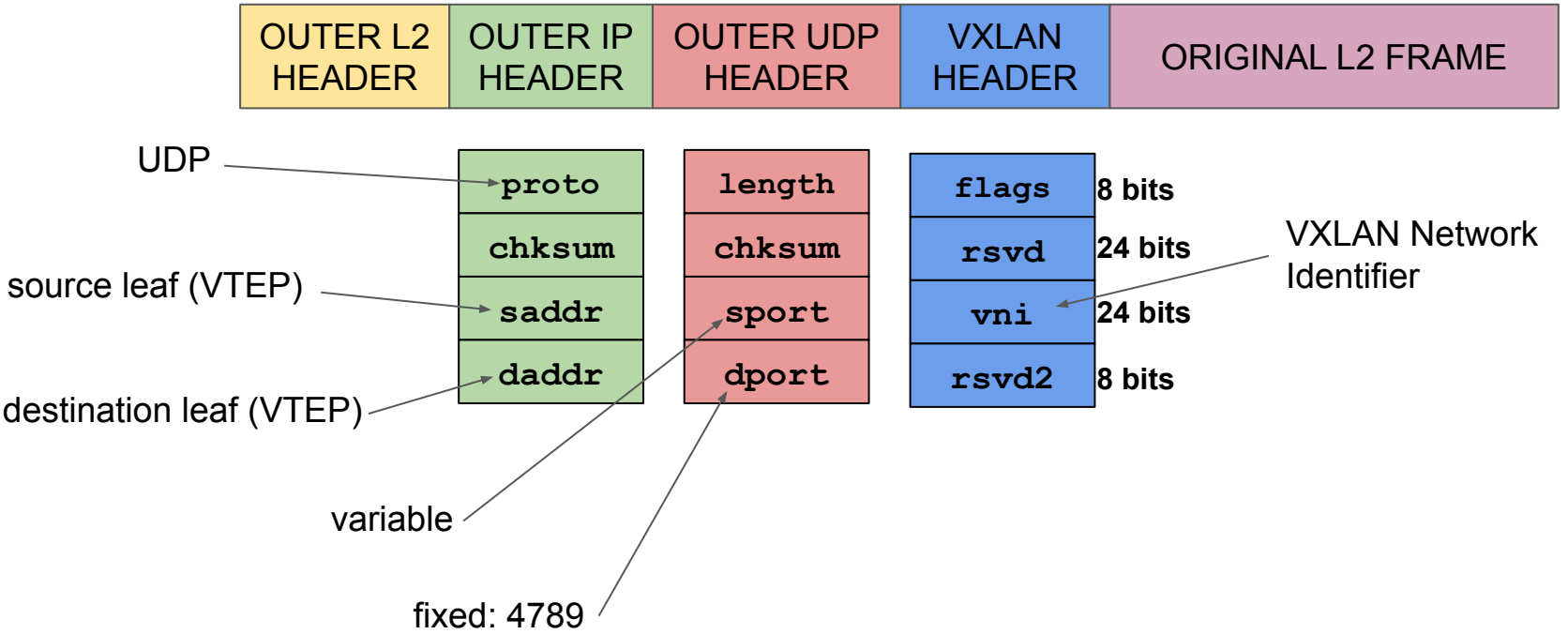
- *Virtual eXtensible Local Area Network* (VXLAN - rfc7348) is an encapsulation protocol that permits to enclose L2 frames in IP/UDP datagrams
- Identifiers are 24 bits → over 16M possible broadcast domains
- It operates at L3 level, packets are *IP routed*
  - load balancing is thus permitted → any routing device knows how to load balance IP/UDP packets
- Only the "*edge*" nodes must support encapsulation → the rest of the network sees simple IP datagrams

VXLAN in action

- ***VTEP*** = VXLAN Tunnel End Point
  - it is the terminal that must support VXLAN encap/decap and initiates/terminates a VXLAN tunnel



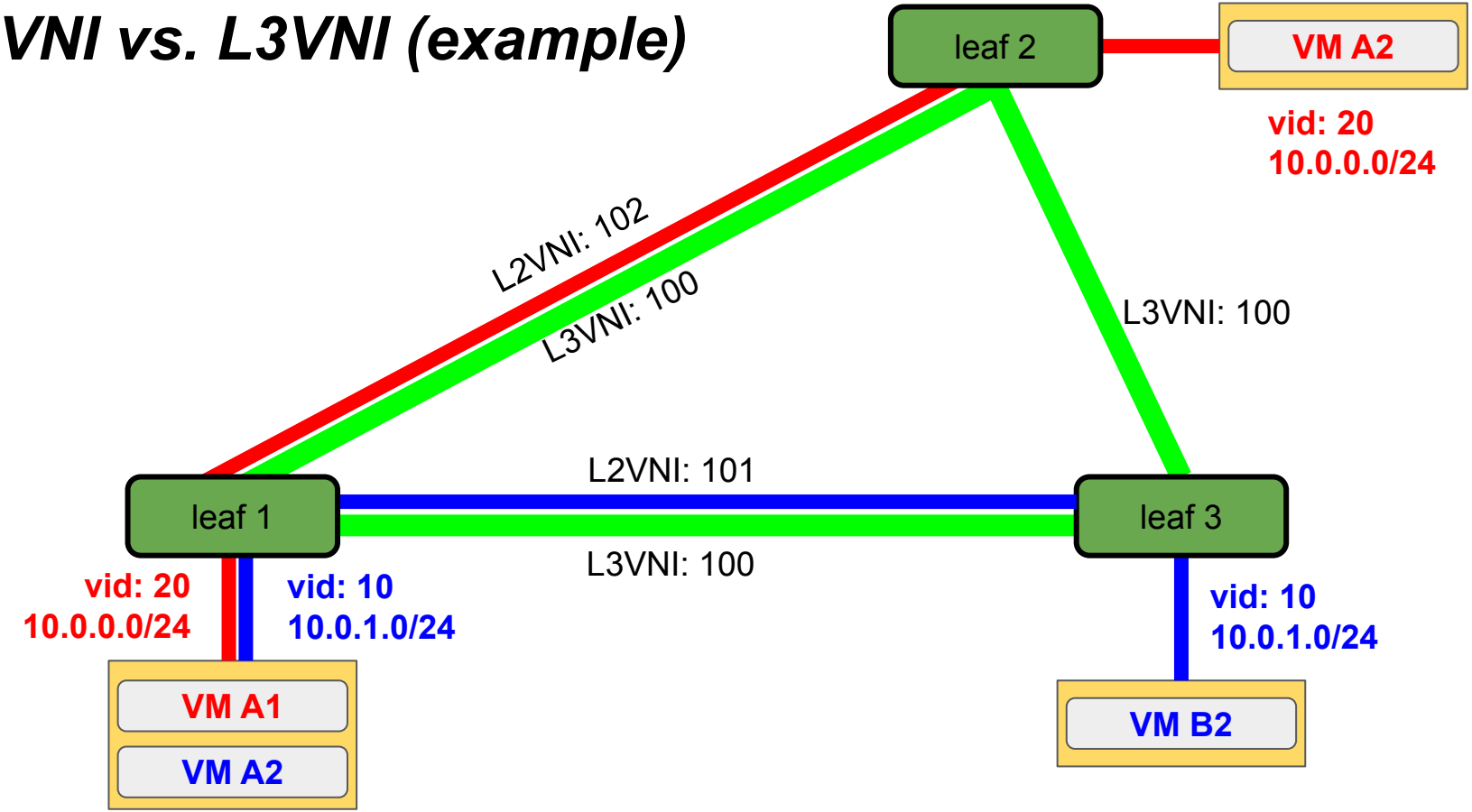
VXLAN packet format



L2VNI vs. L3VNI

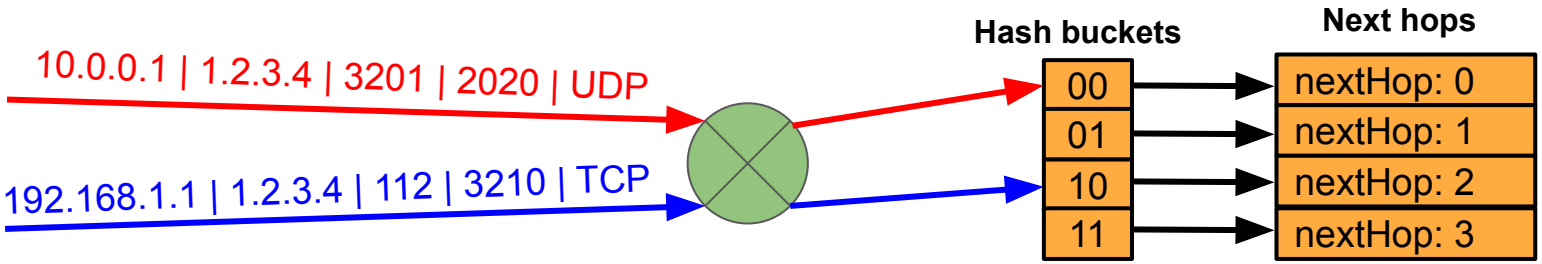
- ***L2VNI*** → Layer 2 VXLAN Network Identifier
  - is the VNI associated with **ONE** broadcast domain (BD) of a specific tenant
- Anyway, a tenant may have more than one broadcast domain, and may need forwarding of traffic between different subnets
- ***L3VNI*** → Layer 3 VXLAN Network Identifier
  - is the VNI (usually) associated with **ONE tenant** for layer 3 routing across different subnets
- An L3VNI is mapped one-to-one to the VRF configured for the tenant
- Within the same VRF there can be multiple L2VNIs

L2VNI vs. L3VNI (example)



Load balancing: ECMP

- As we said, Clos topologies need load balancing to be carried out at Layer 3
- One of the most used solutions is *Equal Cost MultiPath* routing (**ECMP**)
- An hash of the 5-tuple extracted from the packet is calculated to evenly spread the traffic in the available links for a single destination
- The hash will fold into hash *buckets* that will tell the destination to choose
- Why adding complexity with hashes instead of using simpler solutions like round robin?
  - packets belonging to the same flow must go through the same path across the network
  - For example: a burst of TCP packets spread across the network could cause out-of-order reception of packets, which is an important performance issue in datacenter networks



VXLAN & ECMP

- ECMP is calculated on the 5-tuple of the **outer headers** → remember that not all the nodes in the L3 underlay network may handle VXLAN encapsulation
- The 5-tuple of a VXLAN tunnel is always the same...
  - IP source address = IP address of ingress leaf
  - IP destination address = IP address of egress leaf
  - Proto = UDP
  - UDP destination port = VXLAN port
- … except for the UDP source port → the only one that can vary
- The UDP source port is chosen by the ingress VTEP that makes an hash of the **inner** packet and selects the udp.sport based on the result

VXLAN VTEPs (NIC and Switches)

- VTEPs, i.e. the tunnel terminations, can be in *leaf* switches as well as in the *end-hosts*, i.e. on each server
- Modern switches intrinsically have Hardware support for VXLAN encapsulation → fast line-rate VXLAN processing
- In end-hosts not all the NICs support hardware acceleration of VXLAN processing → performing the encap/decap operations in SW is a heavy task
- Hypervisors host an OvS instance that handle network virtualization

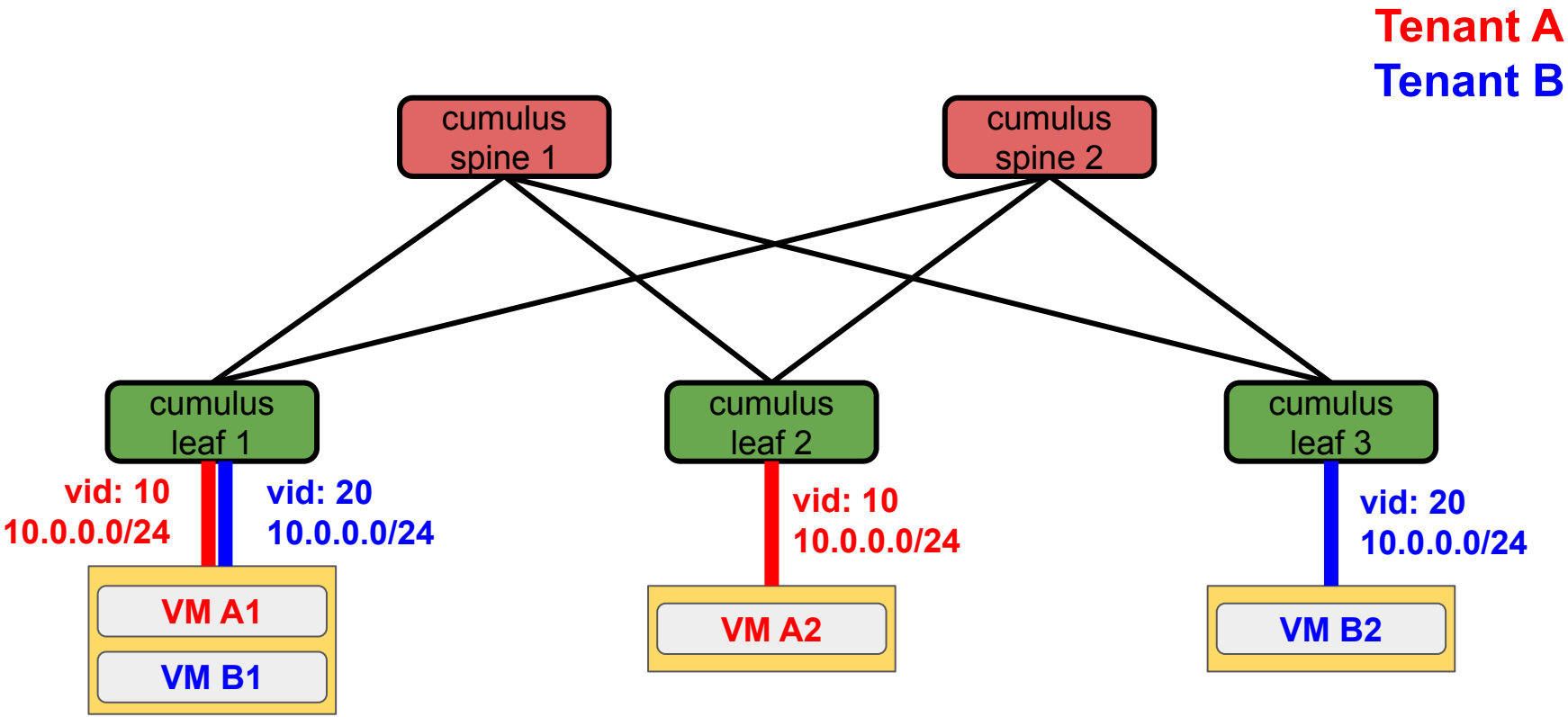
Why VXLAN and not e.g. MPLS?

- Short answer (which applies for most networking protocols…)
  - MPLS was not designed for network virtualization
  - MPLS was designed to speed up packet processing bypassing the routing in core switches (forwarding is done with label switching)
  - VPLS (Virtual Private LAN Service) is used to extend L2 networks using the MPLS dataplane
    - not really an overlay network (MPLS is anyway a *switched* network)
    - usually not employed for datacenter networks
- On the other end…
  - VXLAN is **designed** to enable L2 frames to be forwarded in ***any*** overlay network, thus implementing network virtualization at Layer 2
  - Due to these functionalities, VXLAN is well suited to support the requirements of cloud networking like VMs mobility

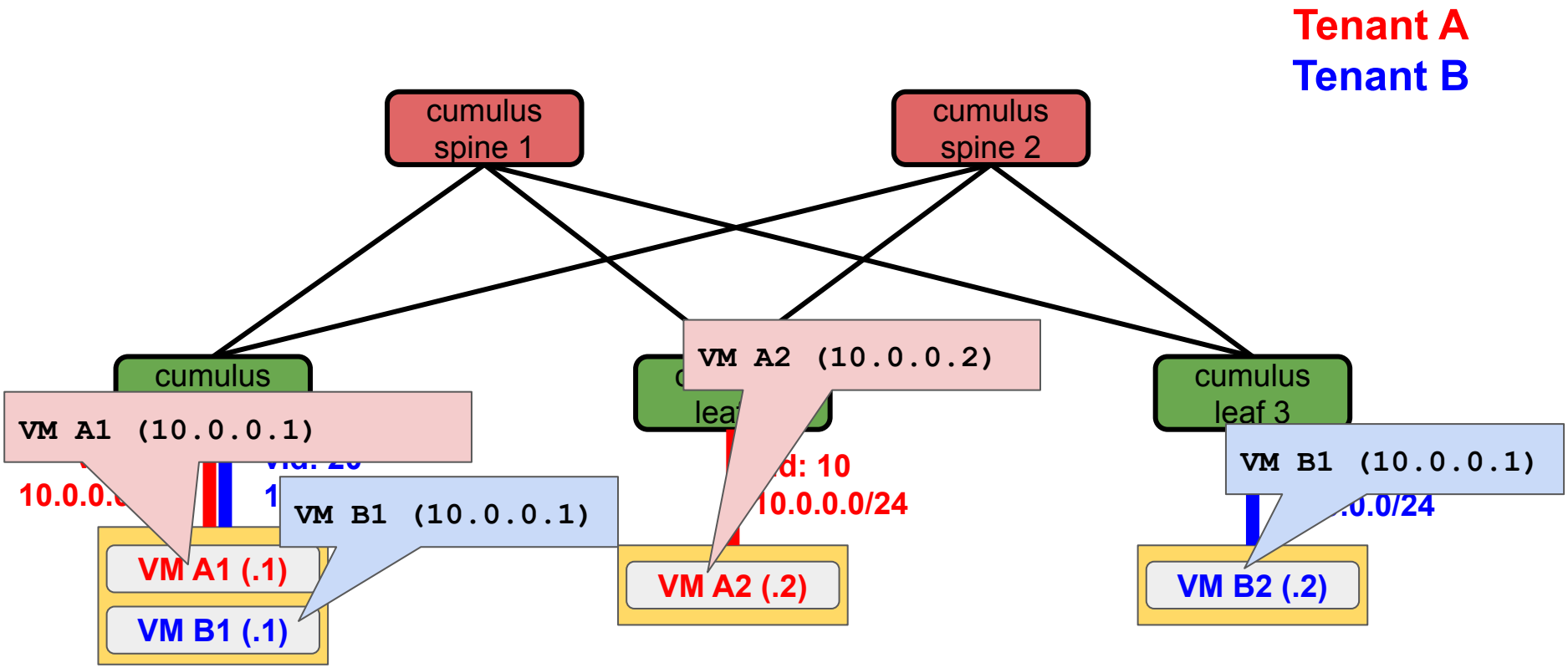
LAB: VXLAN static tunnels in leaf-spine fabric

with cumulus linux

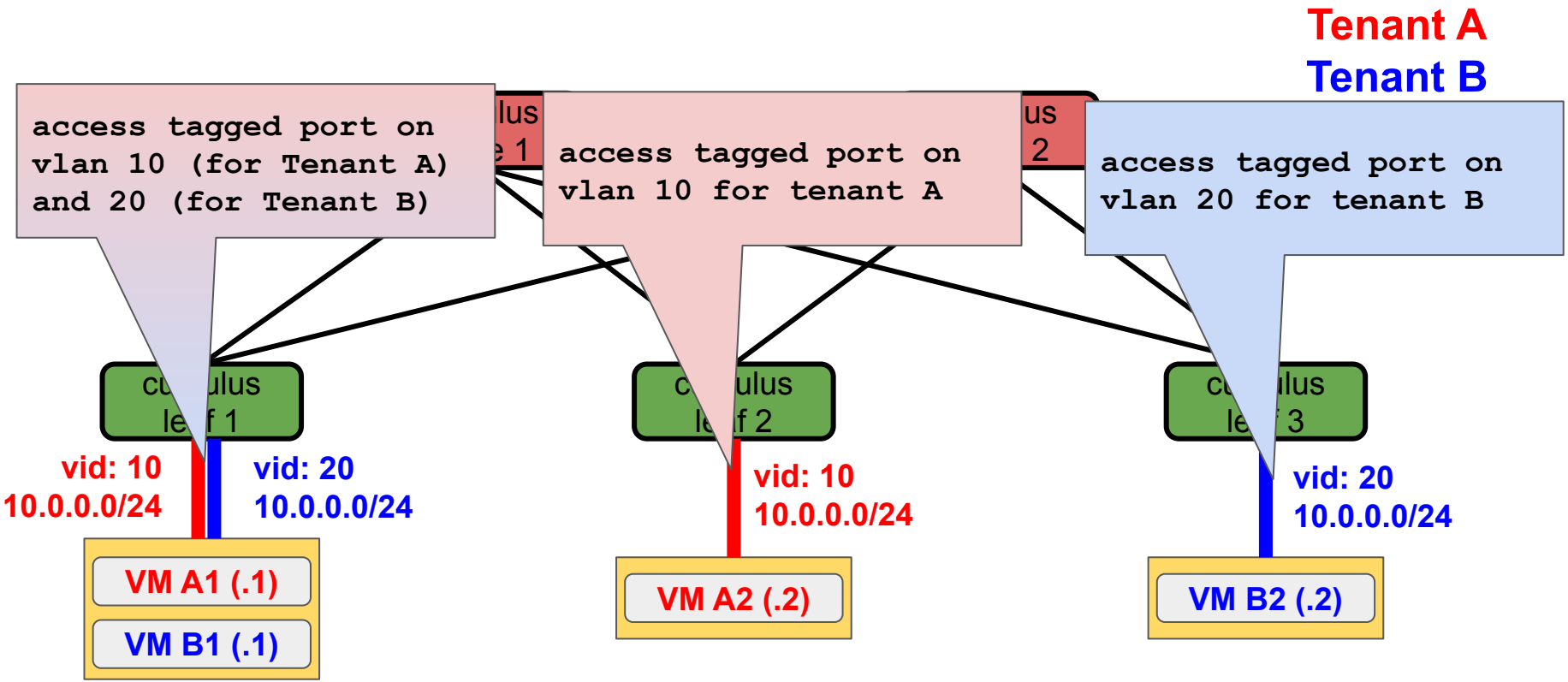
Topology (tenants VM)



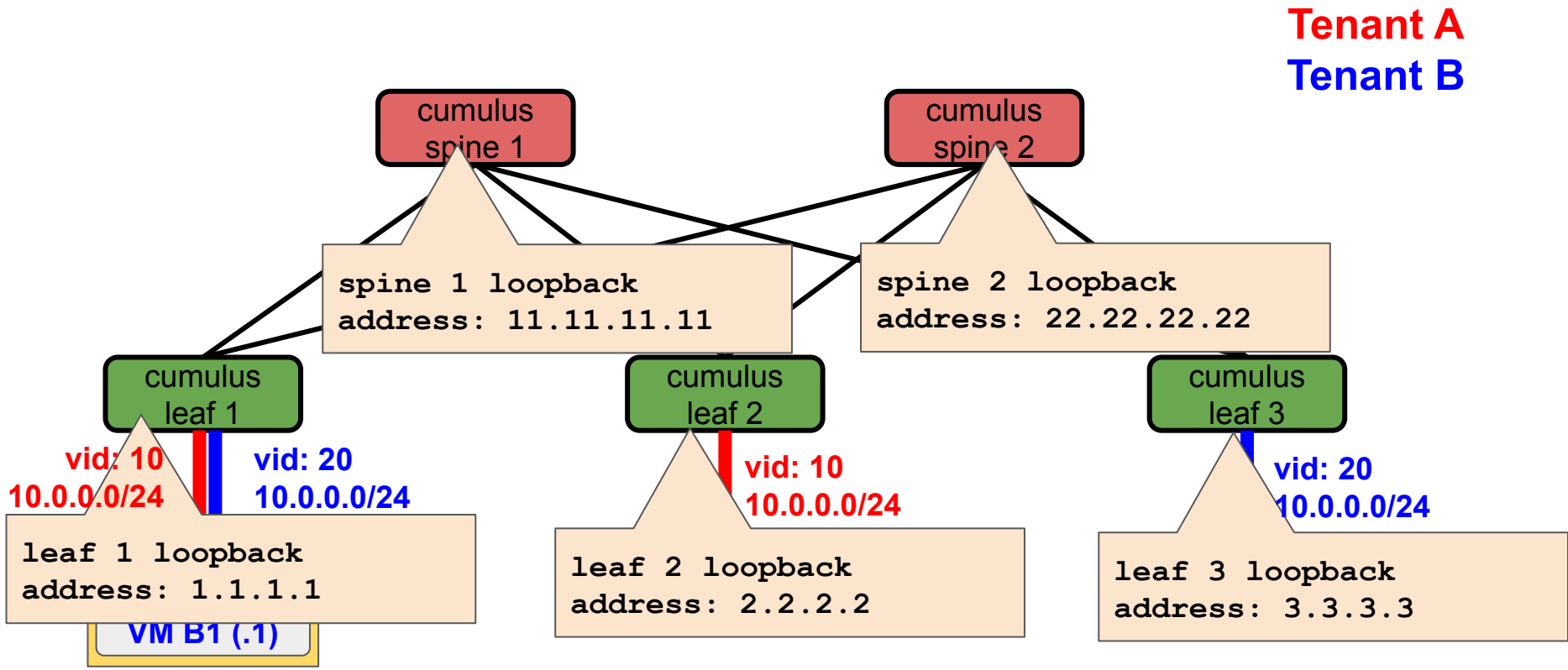
Topology (tenants VM)



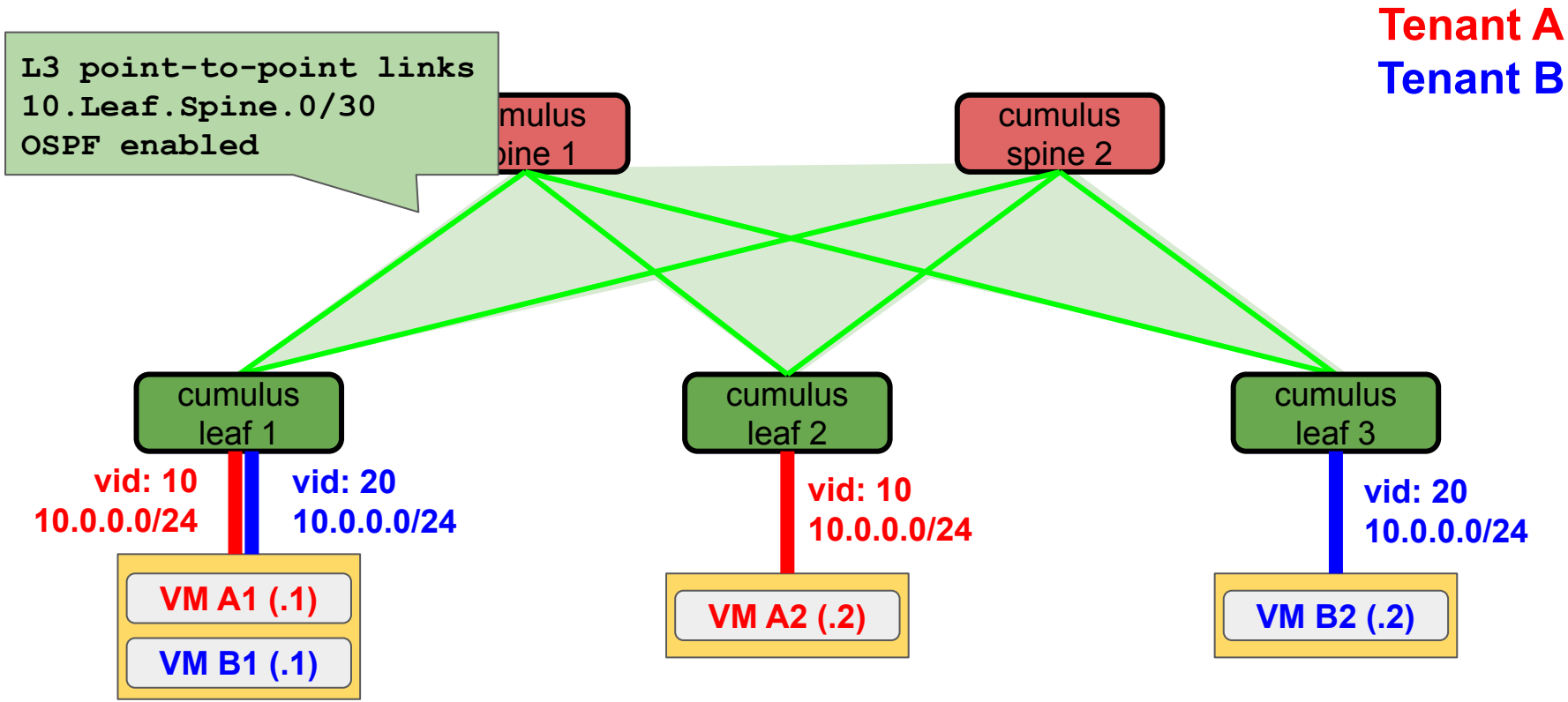
Topology (tenants VM)



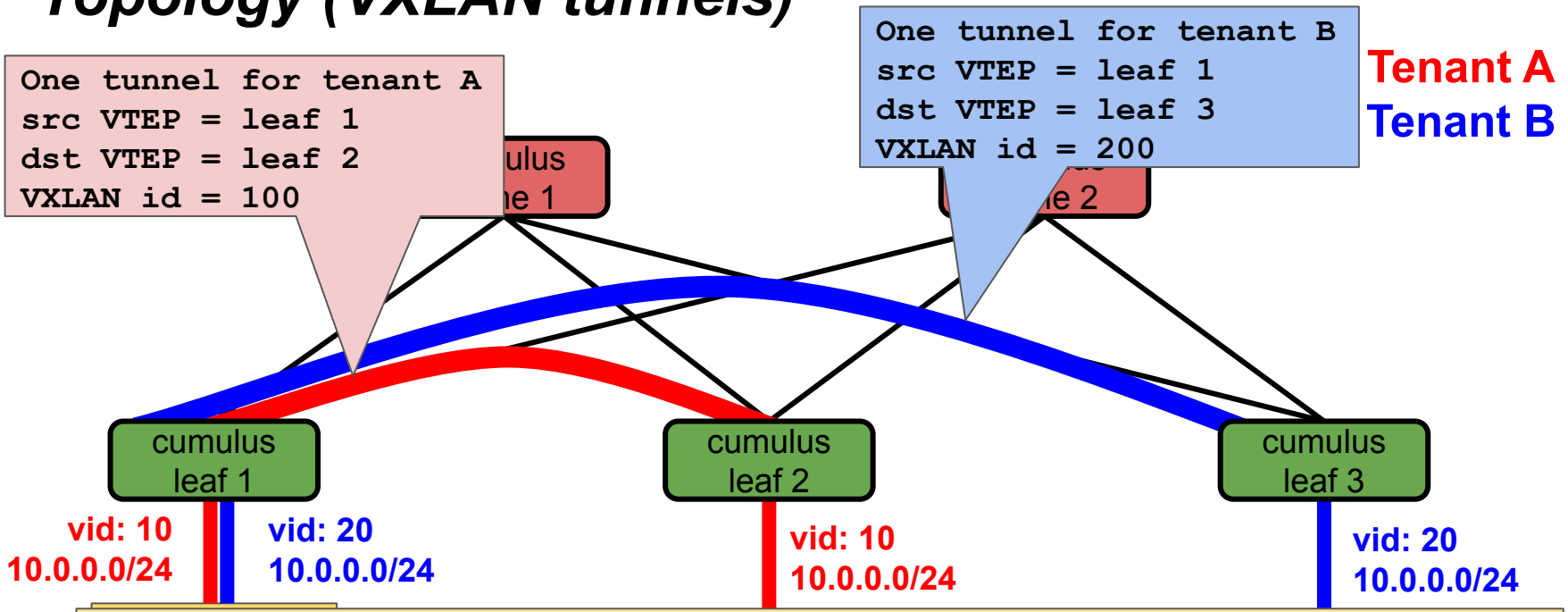
Topology (leaf-spine)



Topology (leaf-spine)

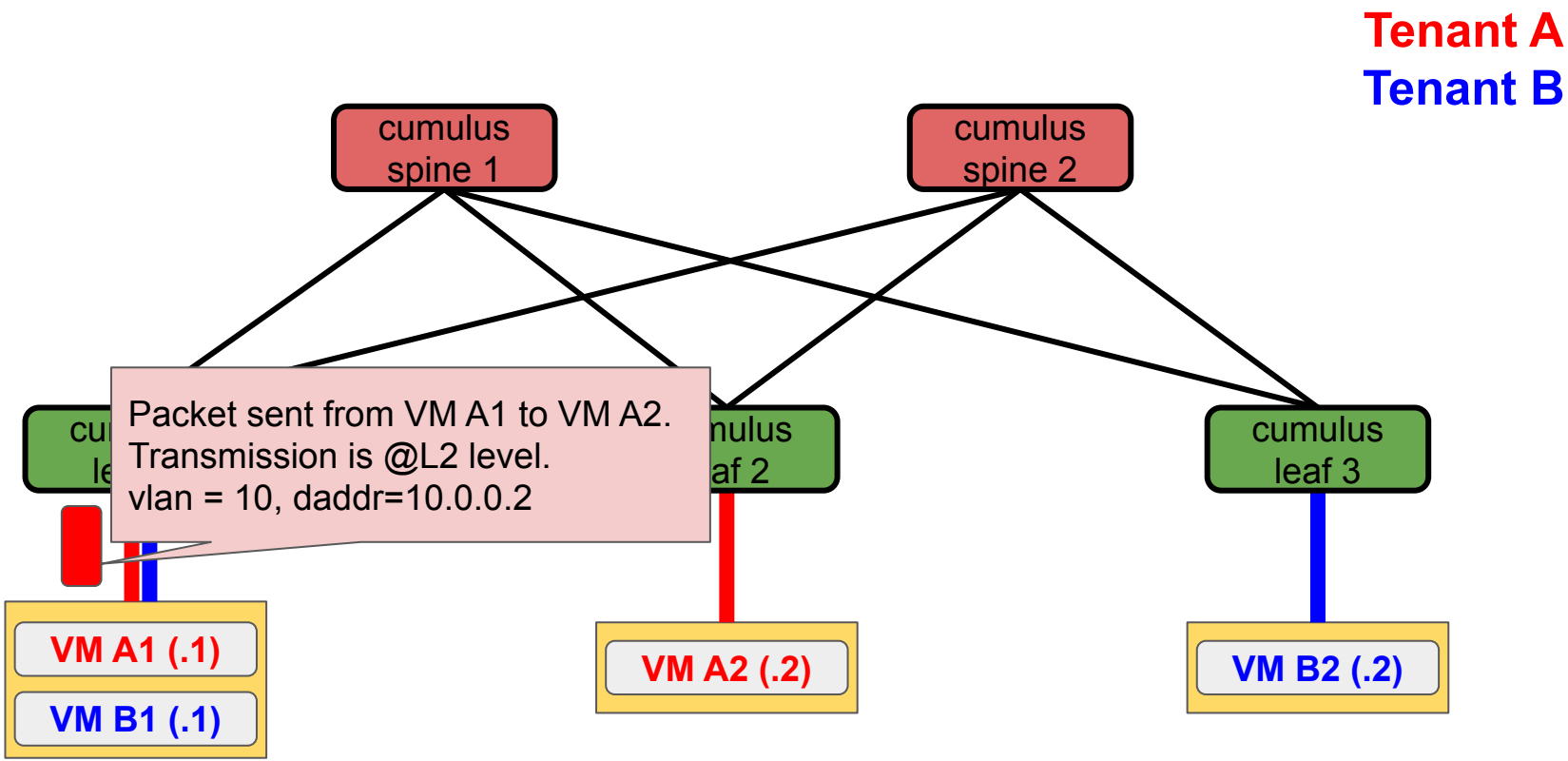


Topology (VXLAN tunnels)

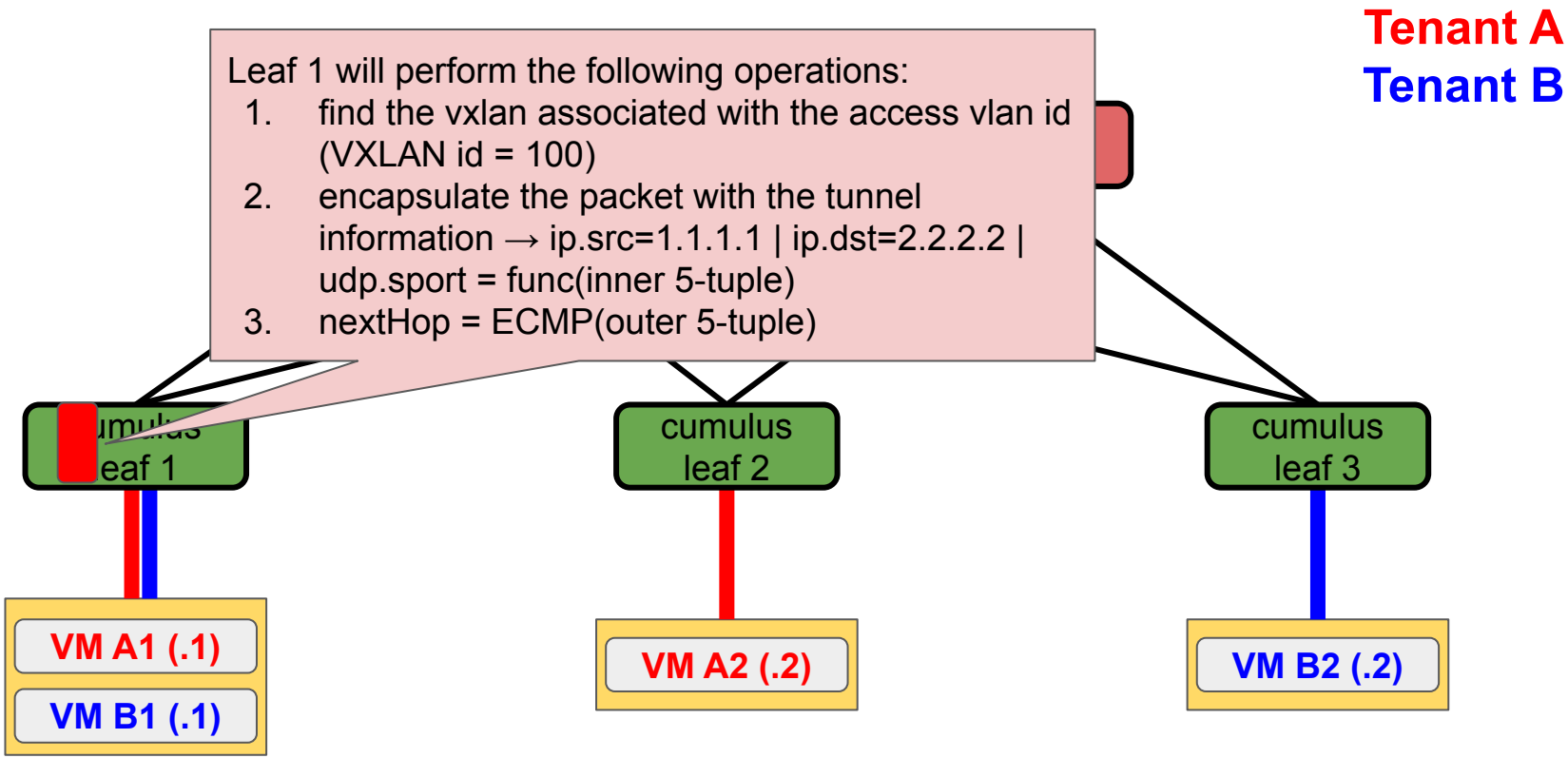


NOTE: A tunnel is established from a source VTEP to a destination VTEP. No need to manually configure the tunnel for each individual path, ECMP will handle that.

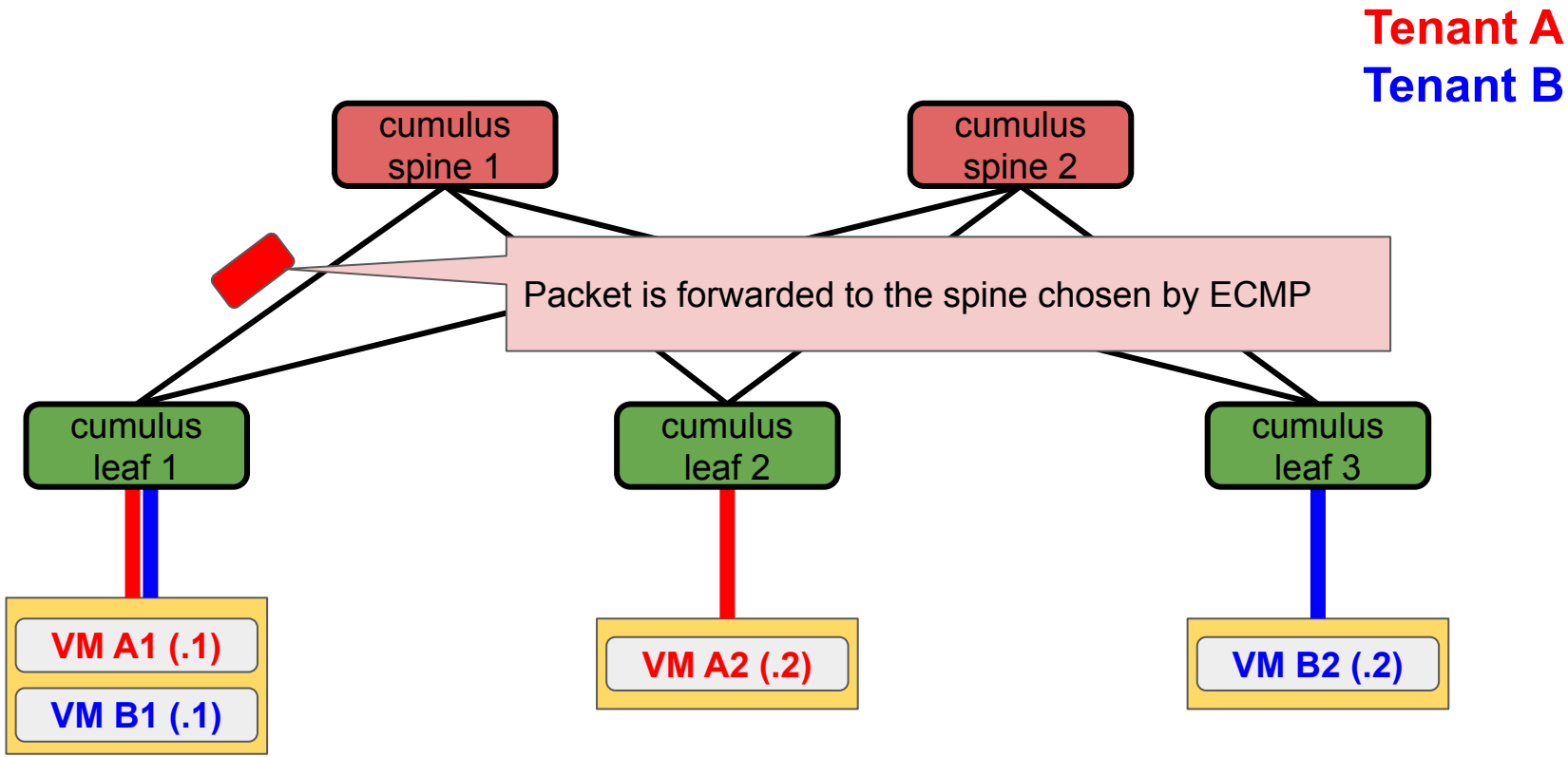
Forwarding behavior



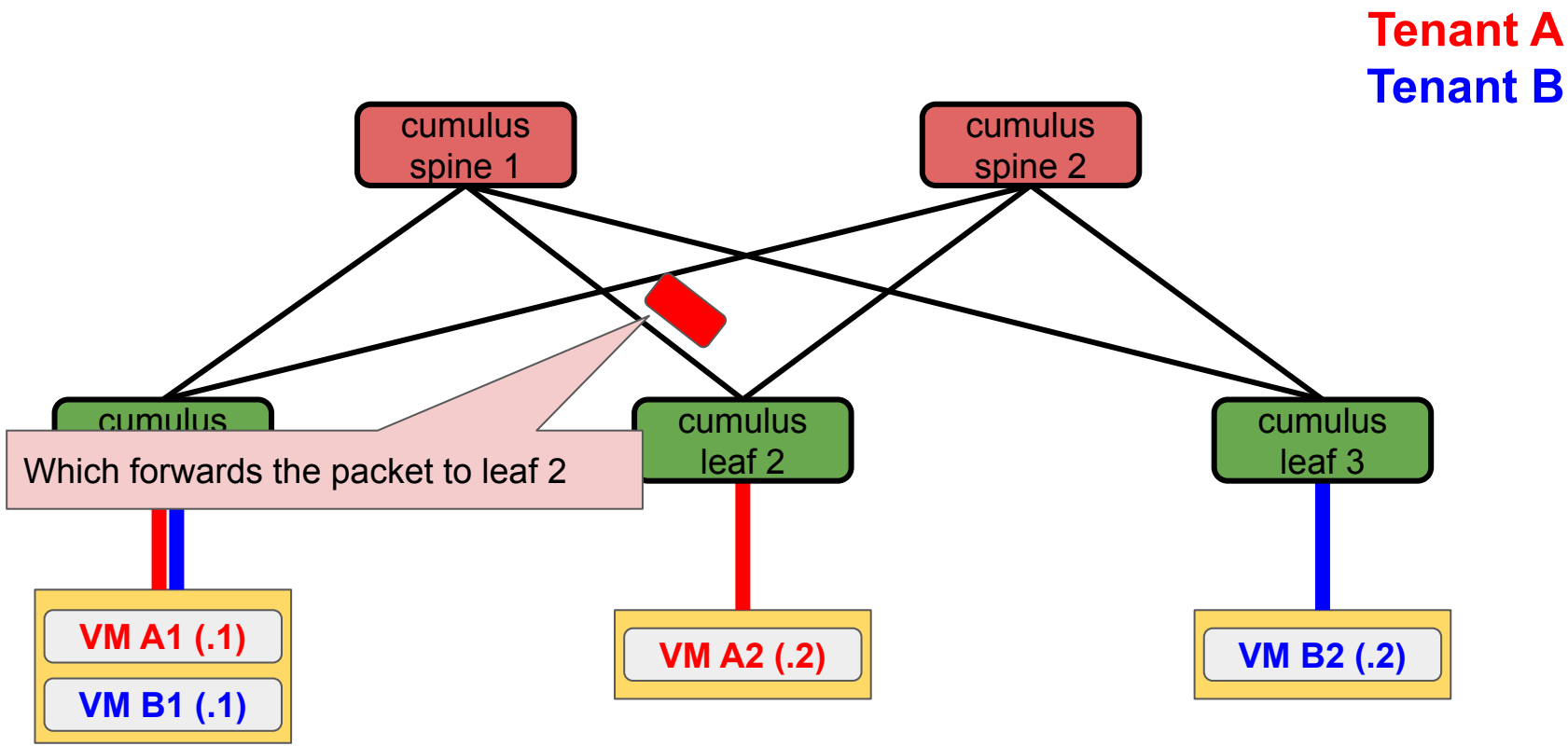
Forwarding behavior



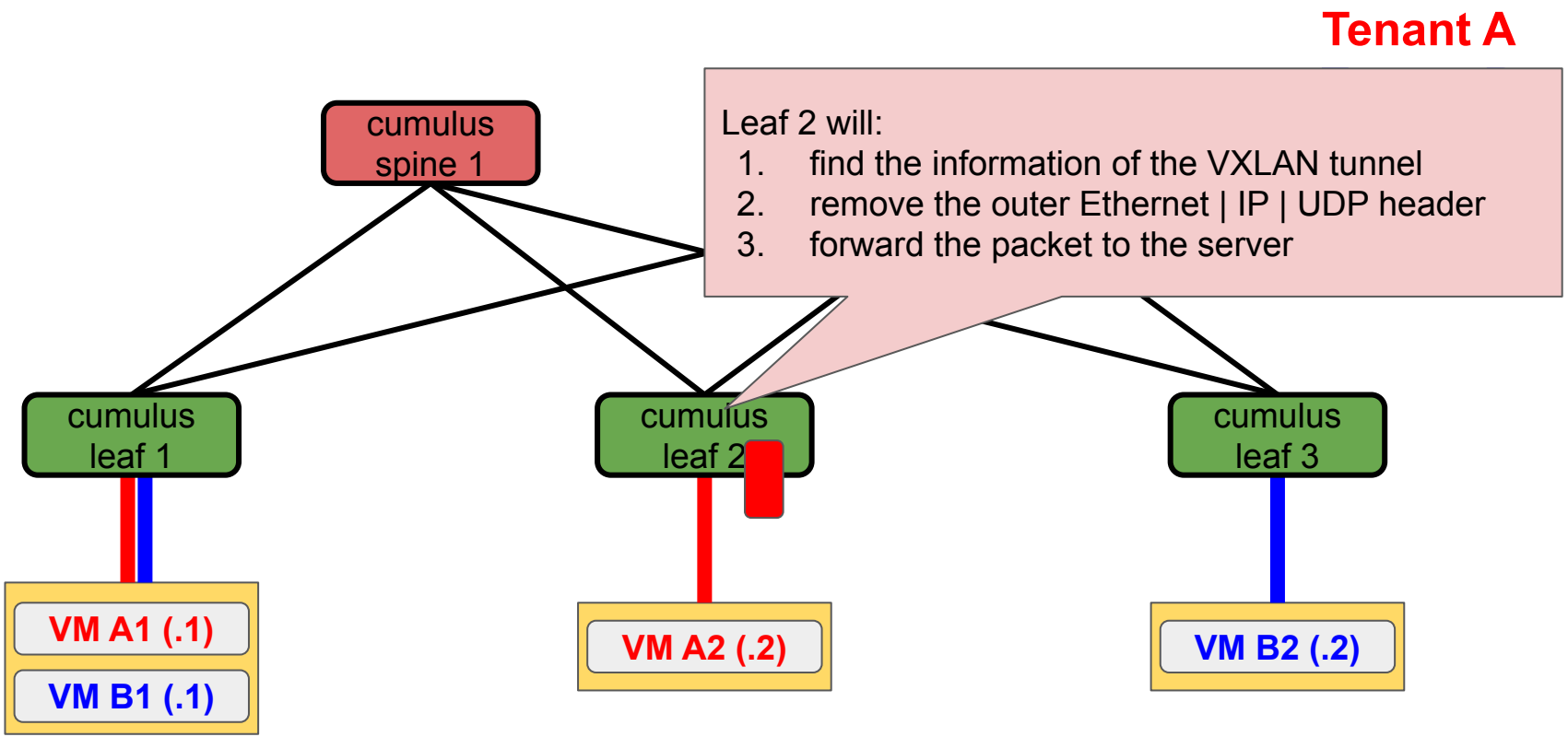
Forwarding behavior



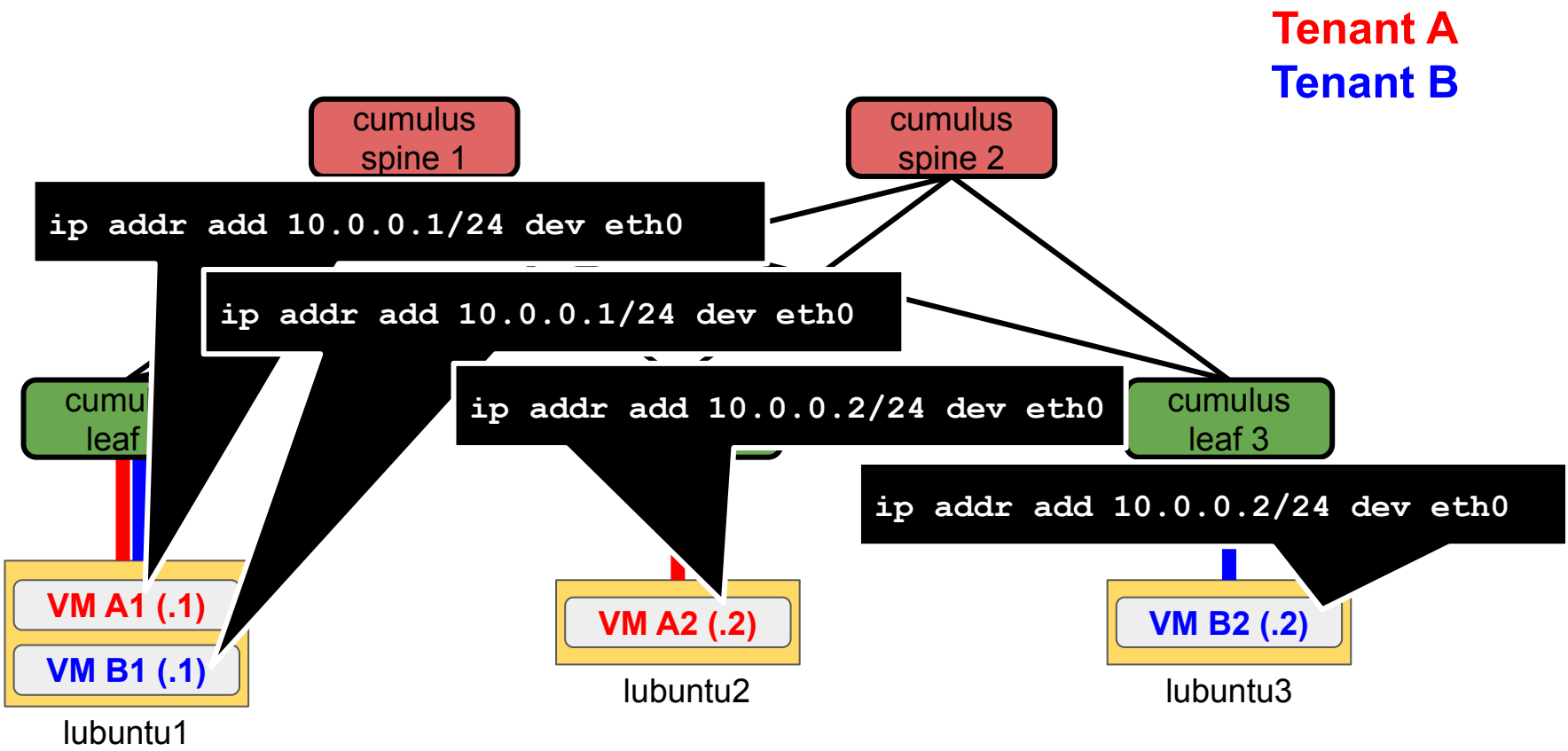
Forwarding behavior



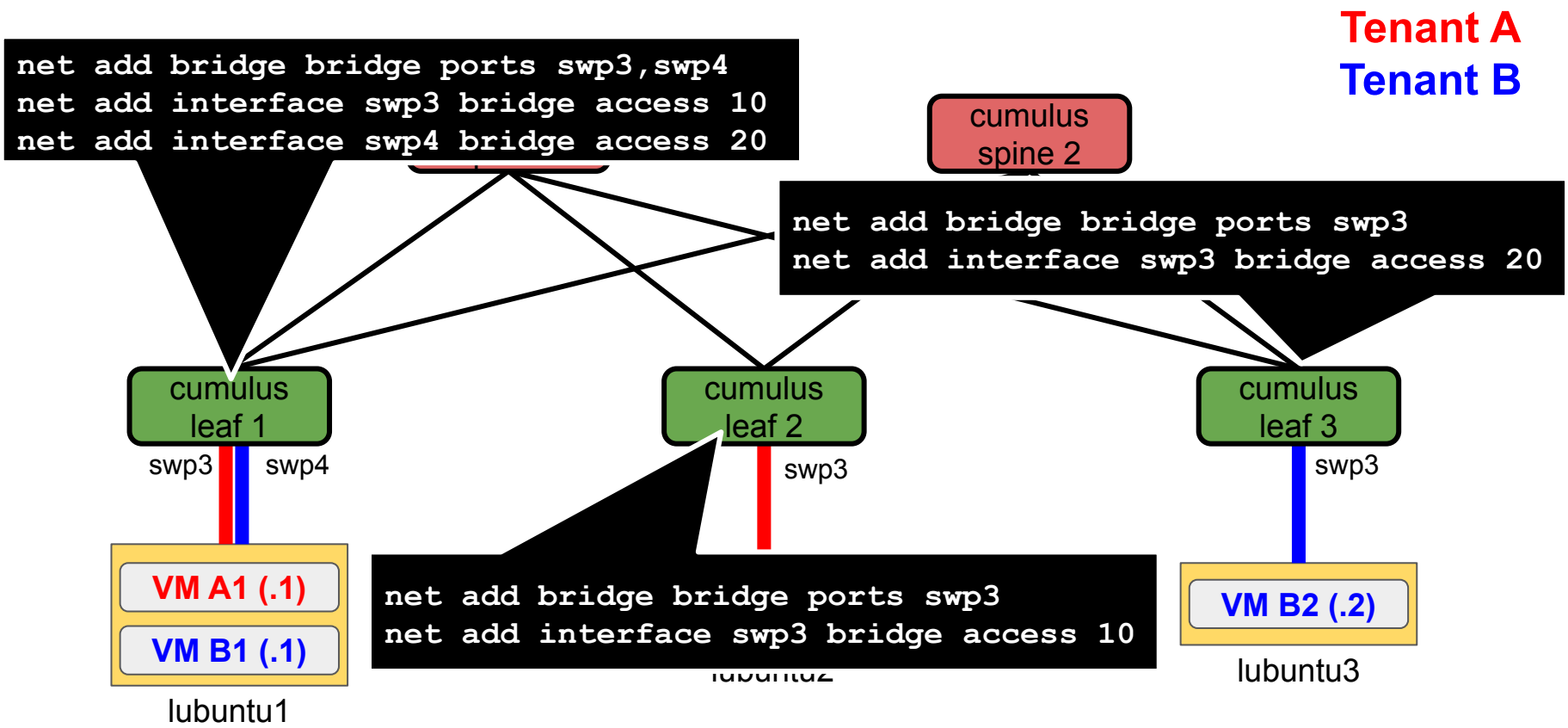
Forwarding behavior



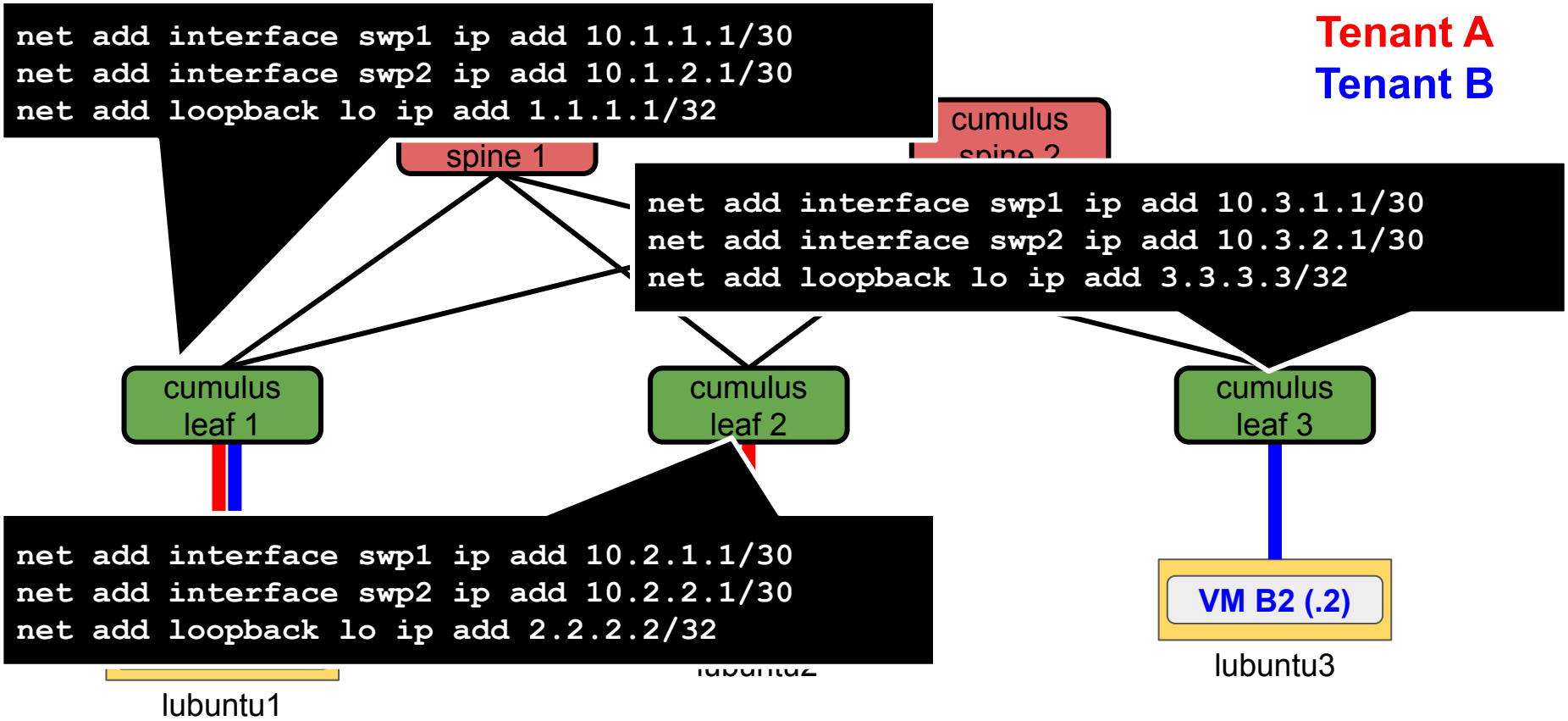
Configuration (end-hosts)



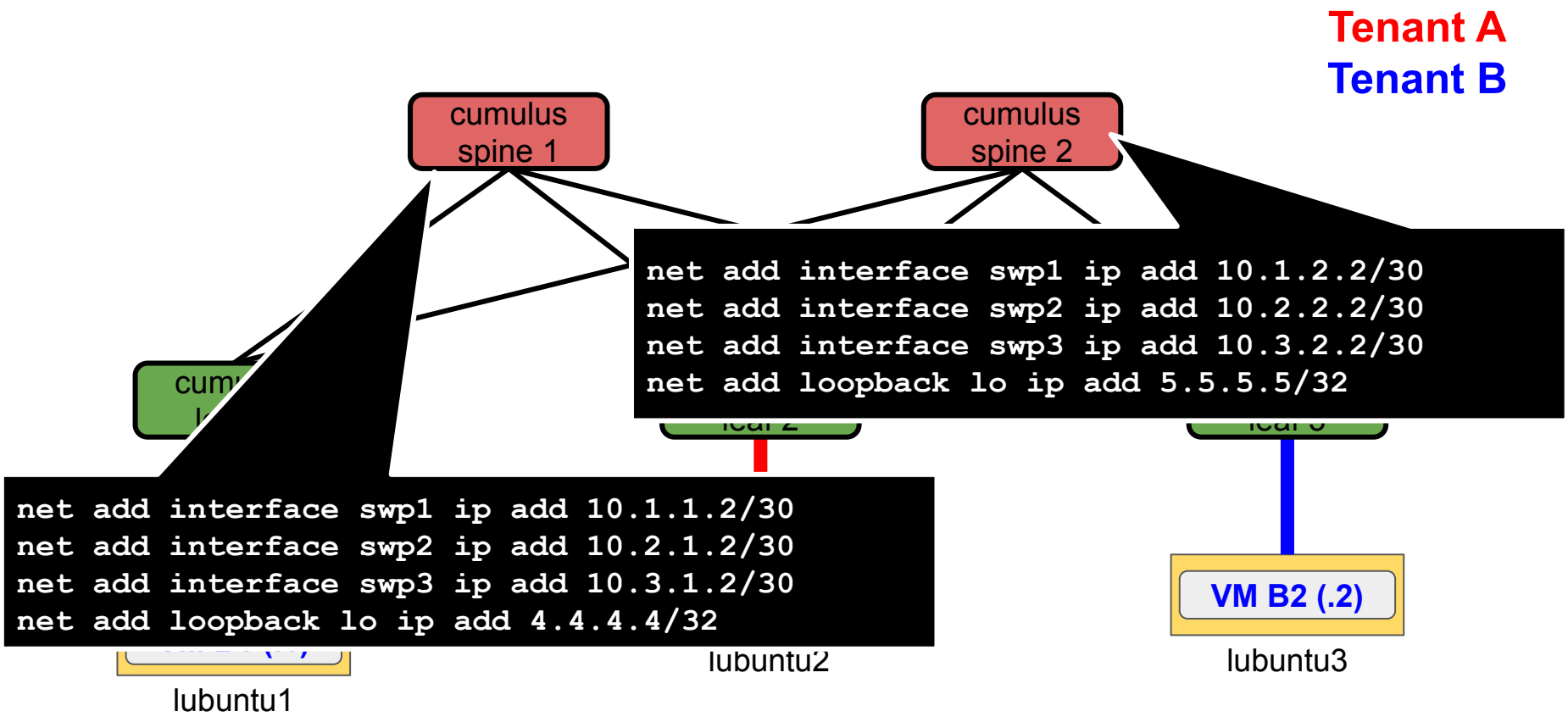
Configuration (leaves - bridge)



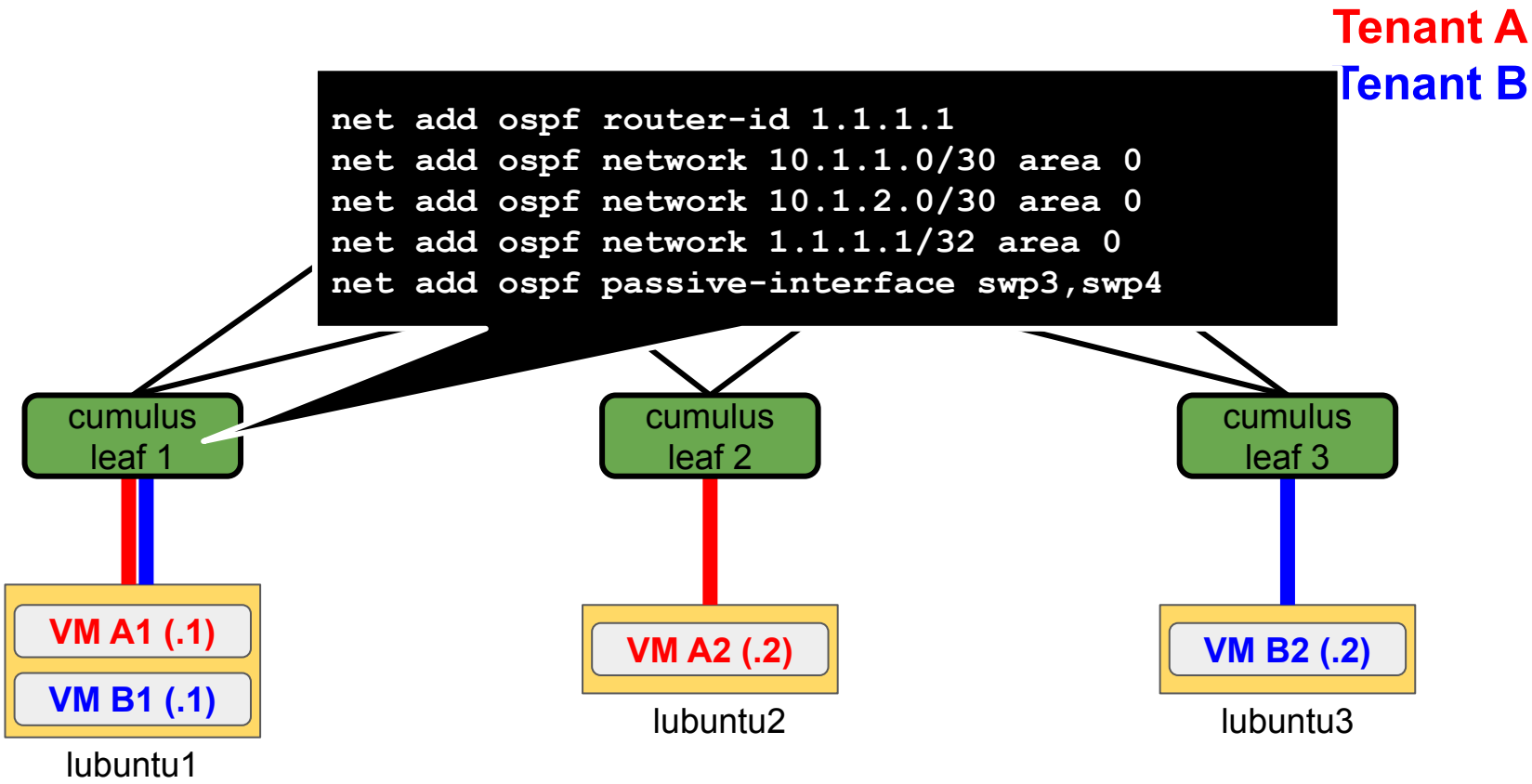
Configuration (leaves - ip addresses)



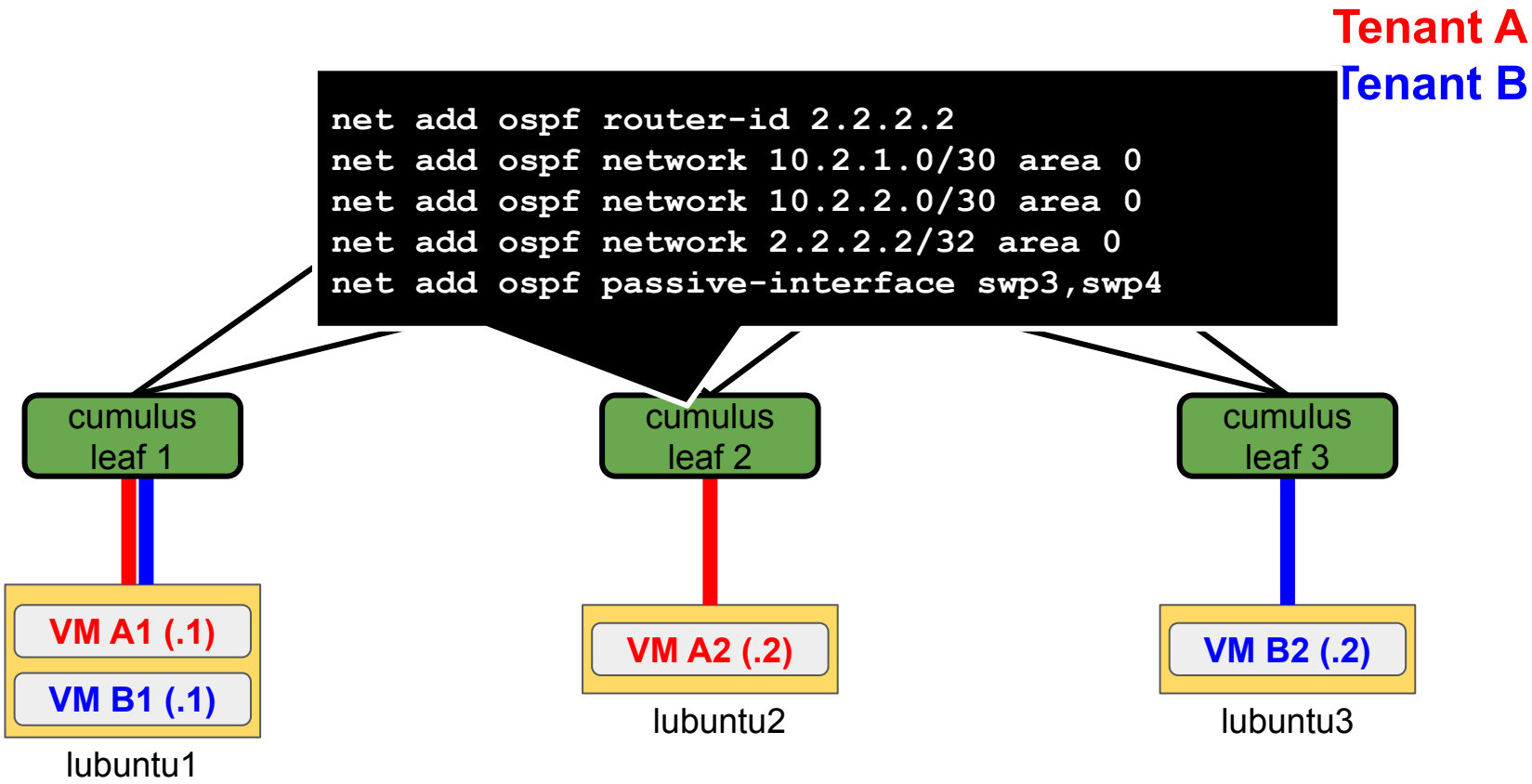
Configuration (spines - ip addresses)



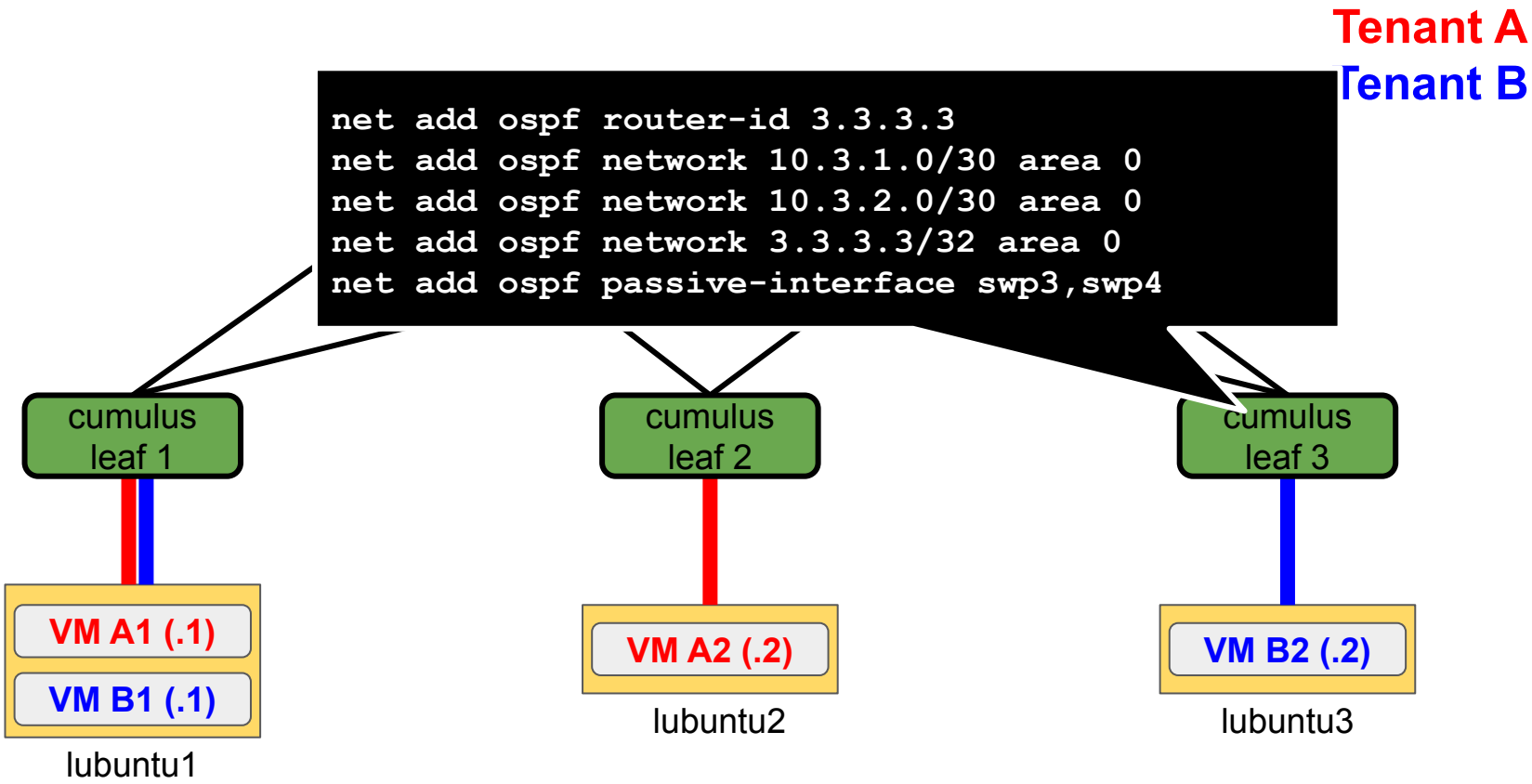
Configuration (OSPF)



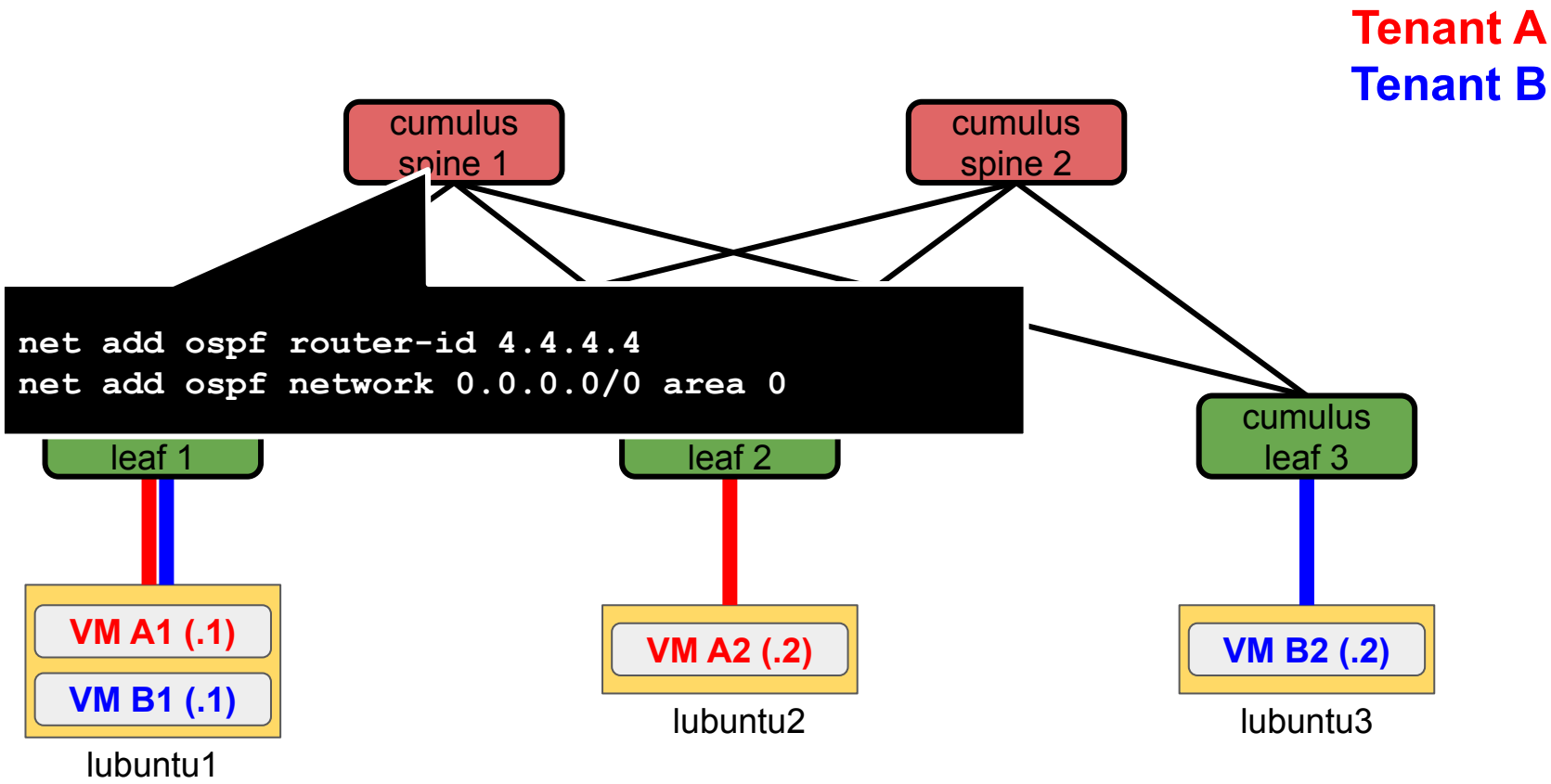
Configuration (OSPF)



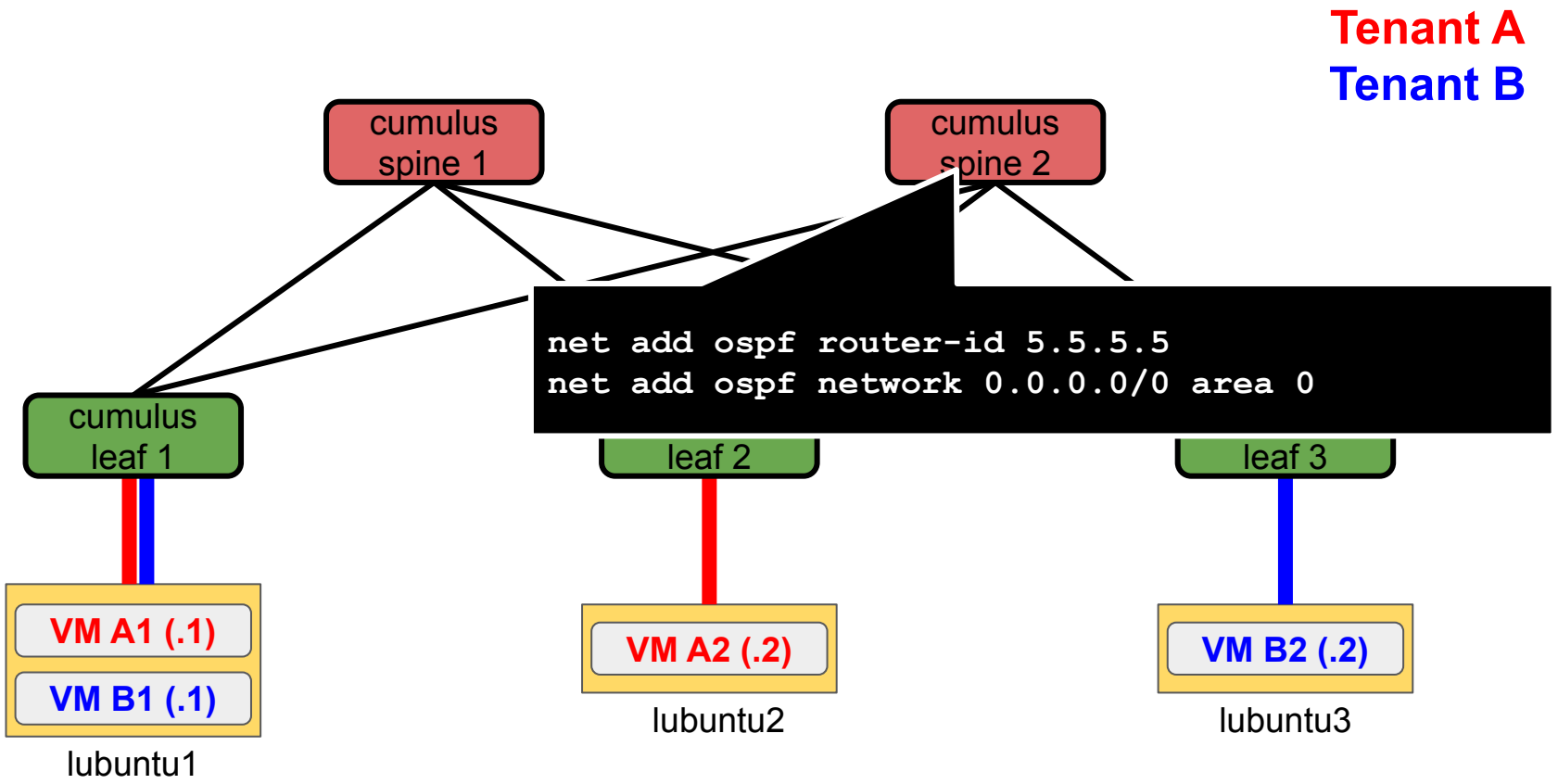
Configuration (OSPF)



Configuration (OSPF)



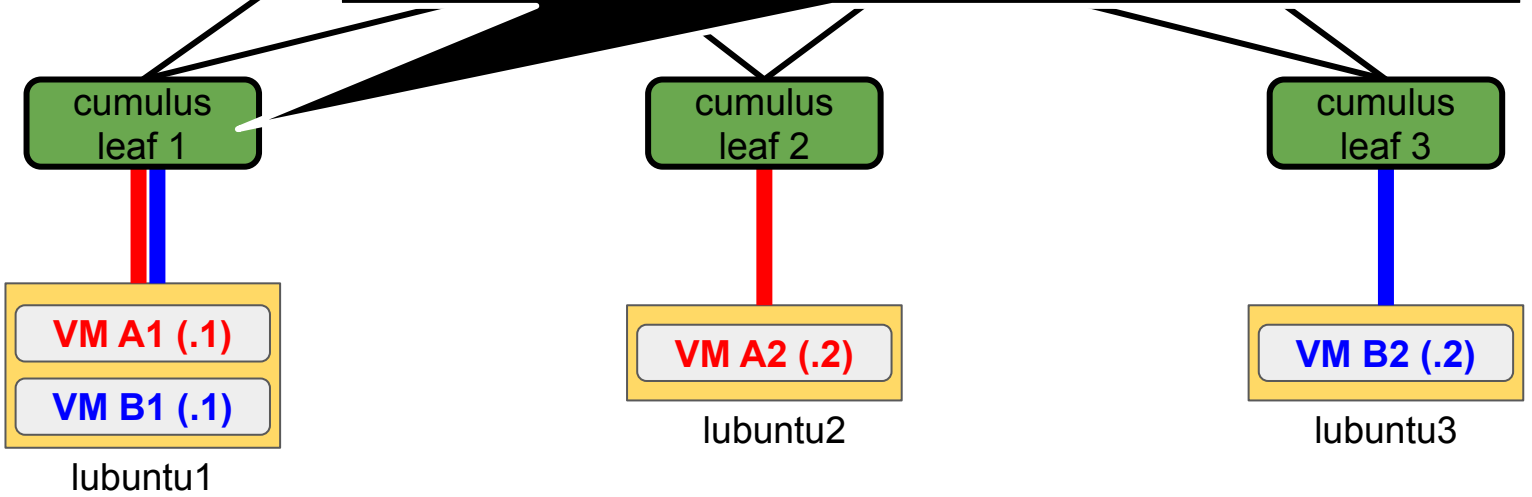
Configuration (OSPF)



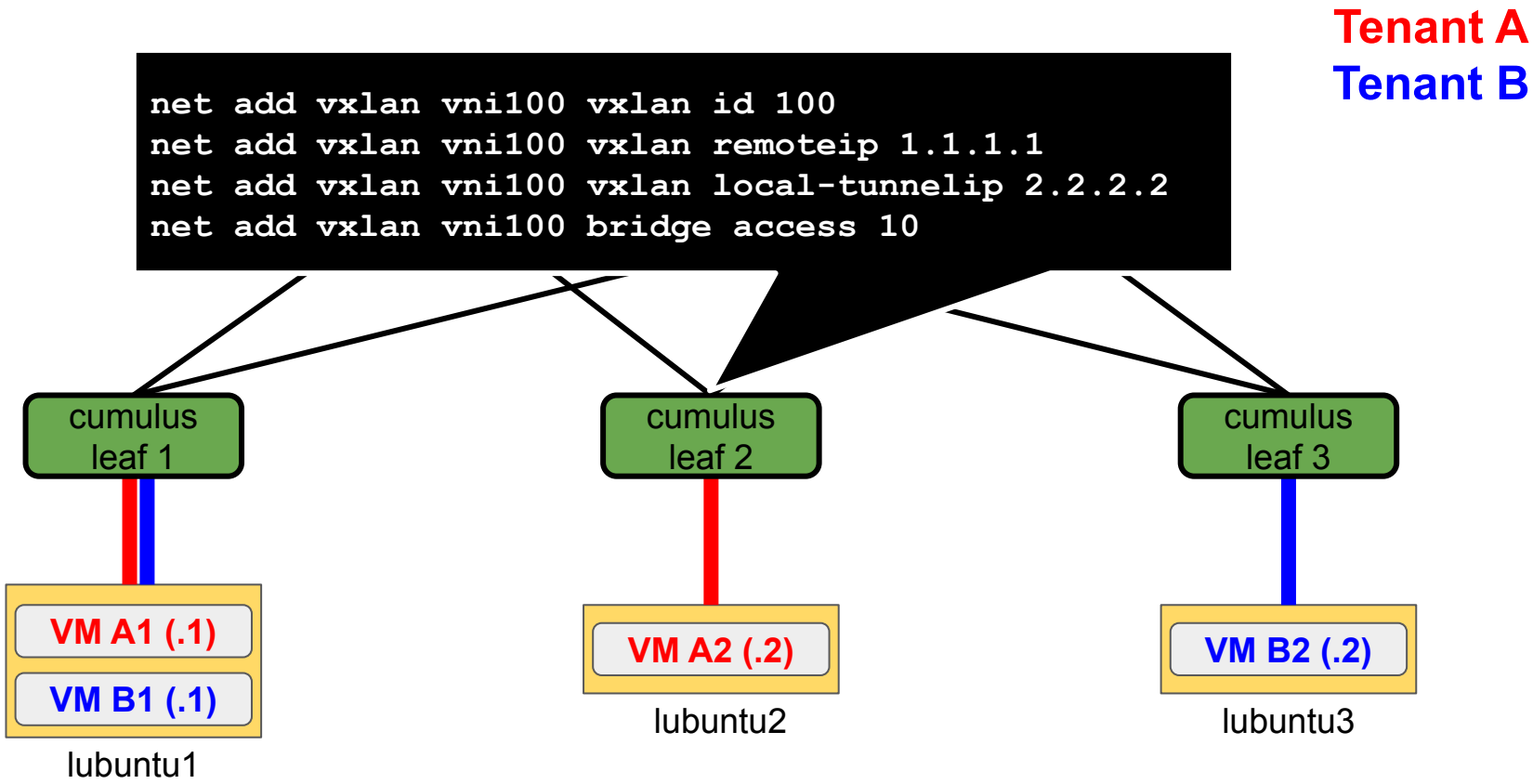
Configuration (VXLAN)

```
net add vxlan vni100 vxlan id 100
net add vxlan vni100 vxlan remoteip 2.2.2.2
net add vxlan vni100 vxlan local-tunnelip 1.1.1.1
net add vxlan vni100 bridge access 10
net add vxlan vni200 vxlan id 200
net add vxlan vni200 vxlan remoteip 3.3.3.3
net add vxlan vni200 vxlan local-tunnelip 1.1.1.1
net add vxlan vni200 bridge access 20
```

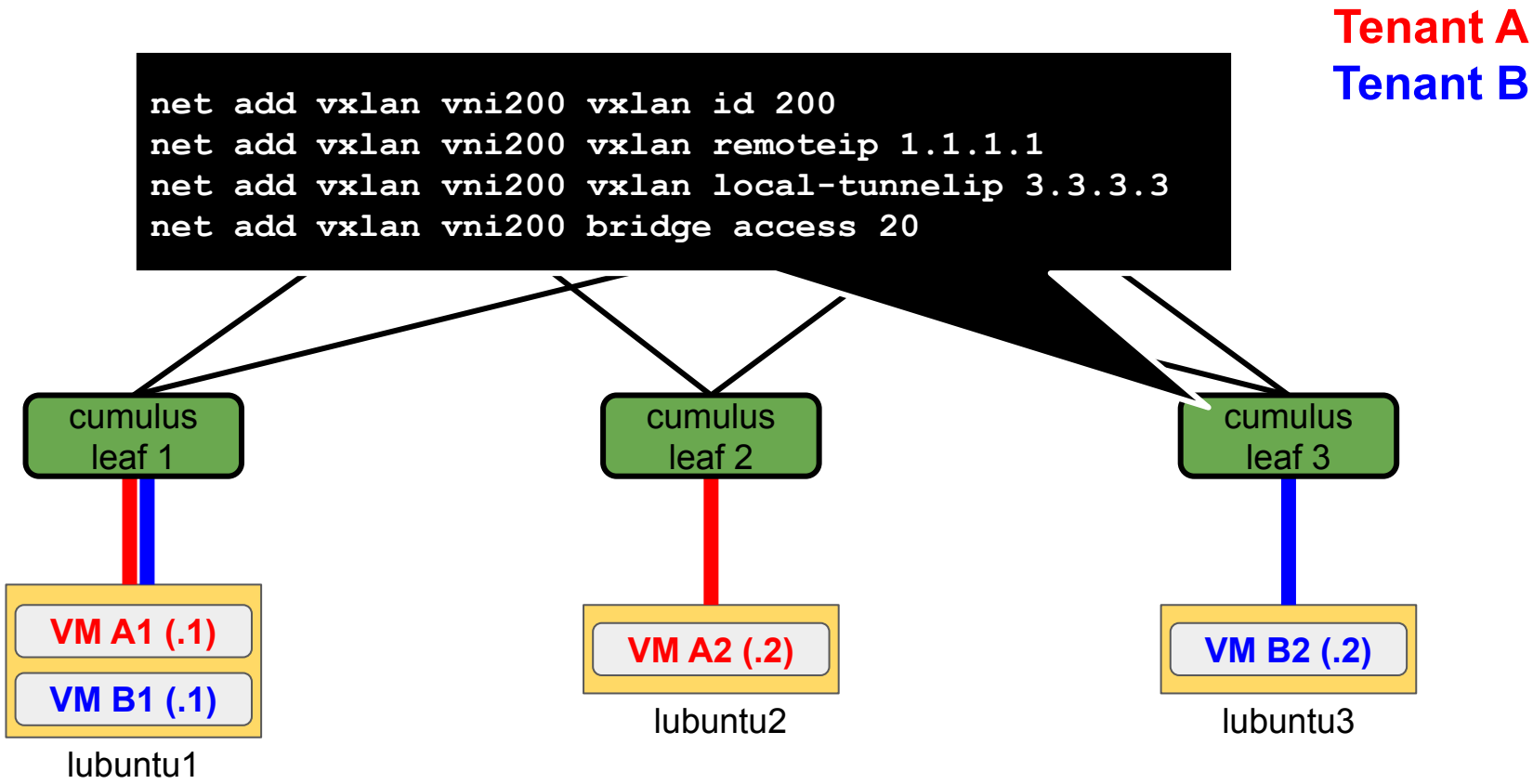
ant A
ant B



Configuration (VXLAN)



Configuration (VXLAN)



VXLAN control plane: Ethernet VPN (EVPN)

A control plane for VXLAN

- The VXLAN protocol described in RFC 7348 did not define a control plane
  - Manually configured tunnels are not scalable → difficult network expansion
  - Moreover, in this way flooding traffic is replicated through the underlay network
- Ethernet VPN (EVPN) is the control plane for network virtualization with VXLAN.
- Uses MP-BGP mechanism (like MPLS VPNs) defining a new address family
- EVPN features:
  - VTEPs are automatically discovered and VXLAN tunnels automatically established
  - Flooding traffic is reduced in the network (e.g. MAC learning is carried out in the control plane)
  - Can advertise both Layer 2 MAC address info and Layer 3 routing info
- The EVPN control plane is also used for MPLS VPNs

EVPN message types

- EVPN has 5 different message types to exchange information:

- Type 1 → Ethernet auto-discovery (A-D) route
- **Type 2** → MAC/IP advertisement route
- **Type 3** → Inclusive multicast Ethernet tag route
- Type 4 → Ethernet segment route
- **Type 5** → IP prefix route

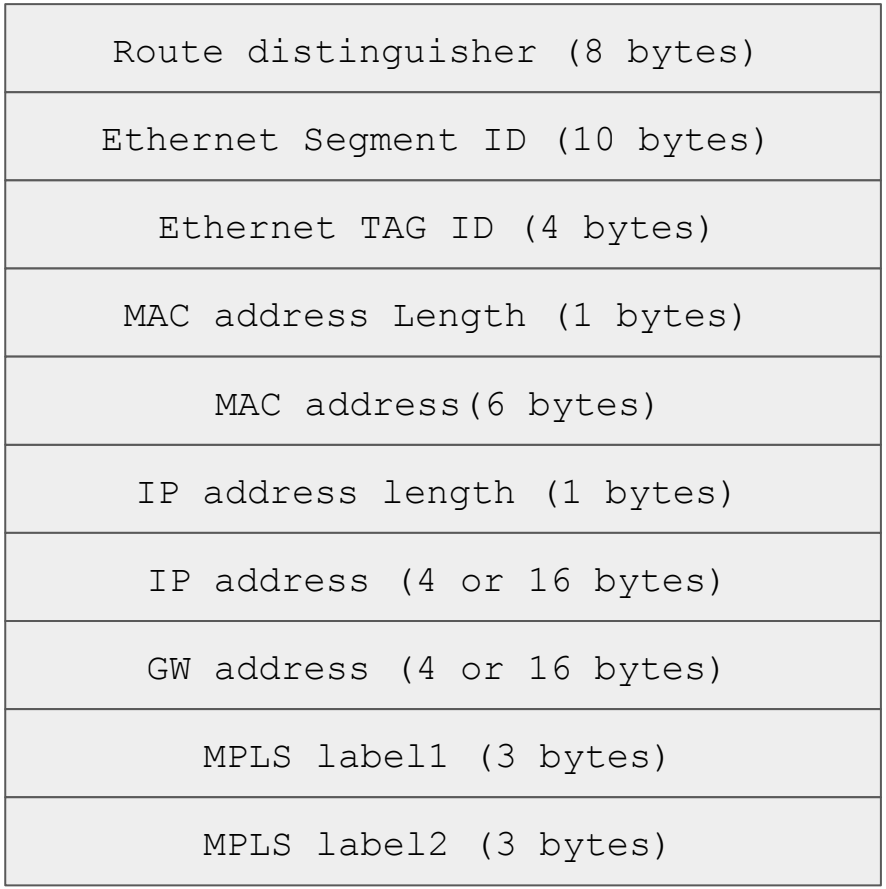
NOTE: Type 1 and 4 are used in scenarios that allows VM multi-homing and to improve reliability in the access side. We will focus on message types 2, 3 and 5

EVPN Type 2

- Type 2 routes are exchanged by VTEPs to advertise each other IP and MAC addresses of hosts in a particular broadcast domain
- Basically, the goal of EVPN Type 2 routes is to synchronize the Forwarding Information Base (FIB) and ARP table across all VTEPs of a particular L2VNI
- Every time a VTEP learns a MAC address or IP address of a host, it generates a Type 2 route advertisement
- L2 flooding traffic usually is not forwarded in the overlay network
  - Type 2 routes actually replace MAC learning and ARP requests

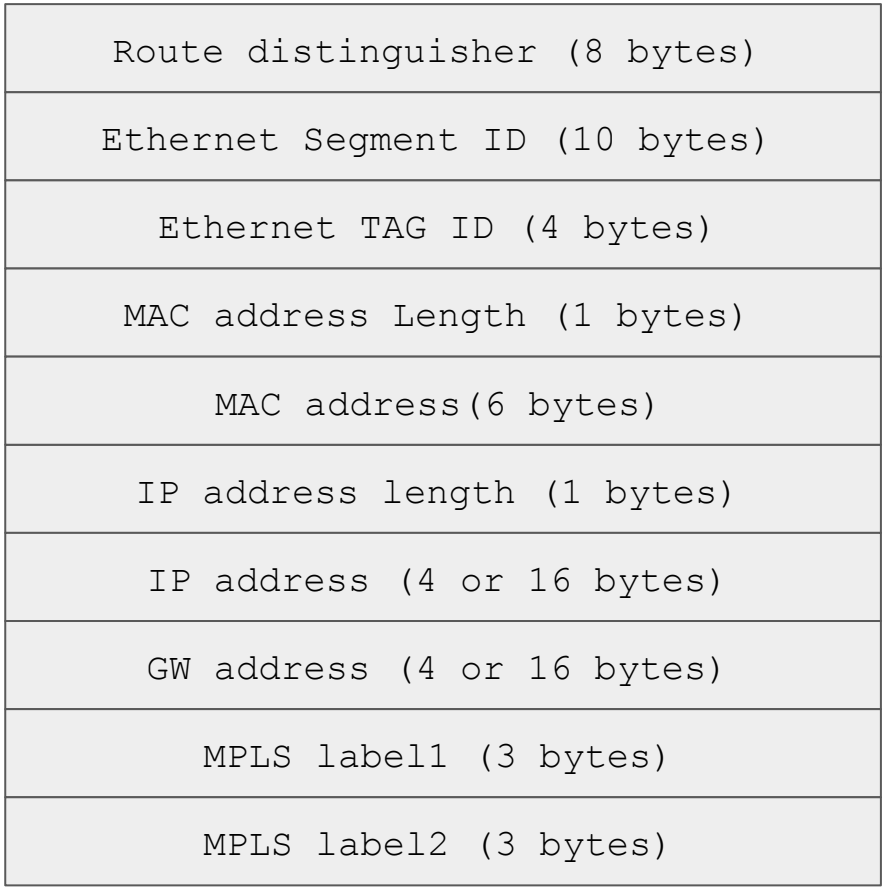
EVPN Type 2 Format

- ***Route distinguisher*** → not like in MPLS-BGP VPNs, a unique identifier of the EVPN ***instance***
- ***Ethernet Segment ID (ESI)*** → (advanced) a unique identifier used for LAG and multihoming
- ***Ethernet TAG ID*** → contains the VLAN id configured on the device
- ***MAC Address Length*** → length of the MAC address (basically if it's present or not)
- ***MAC Address*** → mac address of the host



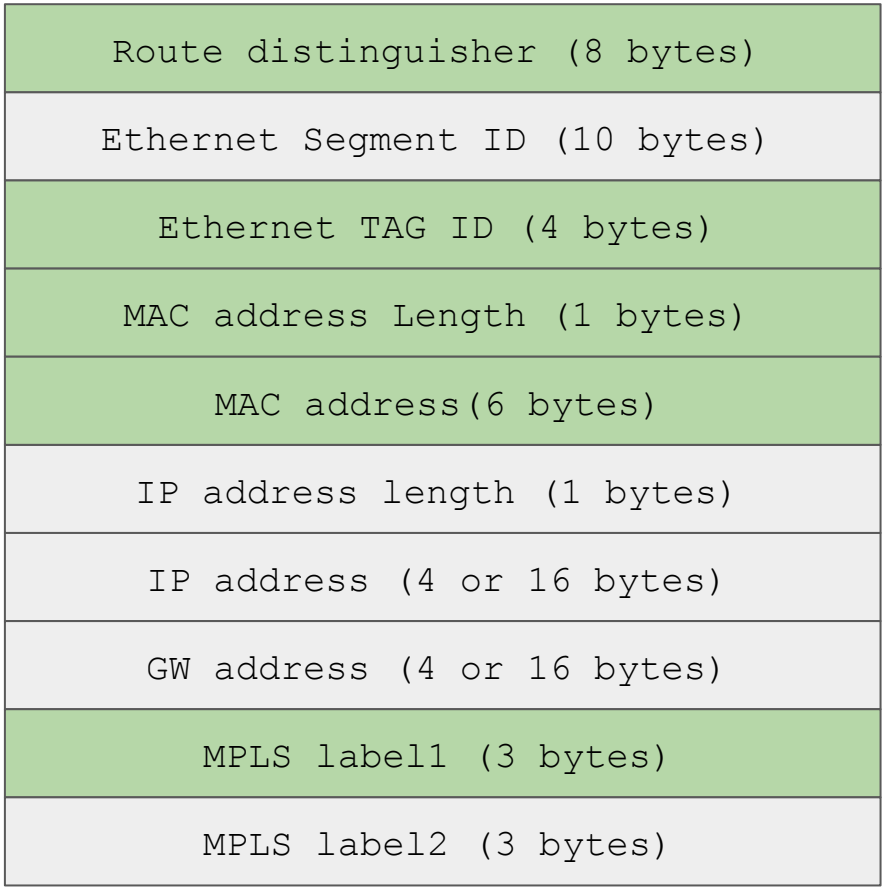
EVPN Type 2 Format

- ***IP address length*** → network mask (e.g. 24, 32)
- ***IP address*** → IP address of the host in the advertisement
- ***GW address*** → IP address of the gateway
- **MPLS Label 1** → contains the VXLAN id used for L2VNIs (3bytes = 24bits)
- **MPLS Label 2** → contains the VXLAN id used for L3VNIs



EVPN Type 2 scenarios - Host MAC advertisement

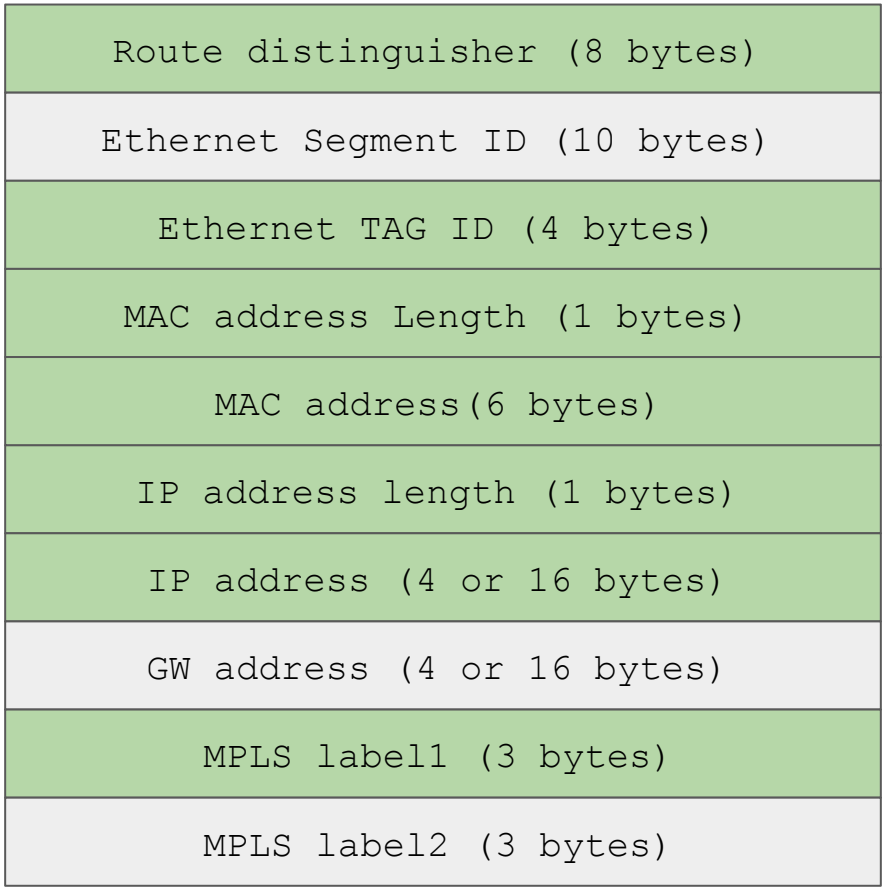
- For seamless L2 communication between hosts on the same subnet, the VTEPs at both ends must learn host MAC addresses from each other
- After EVPN peering between VTEPs, each VTEP will advertise their FIB table entries to the others, i.e. performing MAC address learning via EVPN BGP announcements



EVPN Type 2 scenarios - Host ARP advertisement

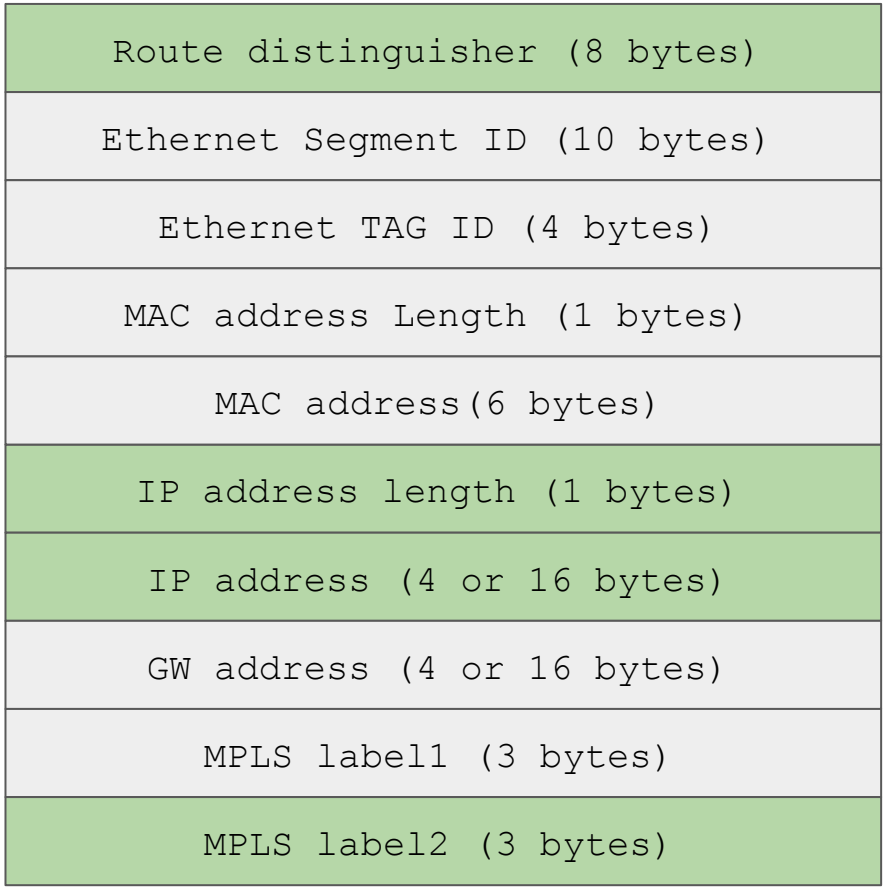
- This EVPN route carries an ARP entry (MAC & IP addresses), so it implements host ARP advertisement
- **ARP broadcast suppression**
  - to suppress the broadcast ARP requests from VMs, the VTEPs announce their ARP table to other VTEPs as soon as they receive ARP information from the hosts (e.g. with gratuitous ARPs)
  - when a broadcast ARP request is received by a VTEP, it will replace the broadcast MAC address (ff:ff:ff:ff:ff:ff) with the host MAC address as destination, therefore it unicasts the packet
- **VM migration**
  - When a VM is migrated behind another VTEP, the VXLAN gateway learns the ARP information from the VM, thereby it announces this information to the VTEP that previously hosted the VM
  - The VTEP that once hosted the migrated VM recognizes the migration and triggers an ARP probe to the VM, to check if it's still running or not. If it fails, it withdraws the routes.

EVPN Type 2 scenarios - Host IP advertisement



EVPN Type 2 scenarios - IP route advertisement

- For Layer 3 communication between hosts in different subnets, the VTEPs must learn the host IP routes from each other
- VTEPs at both ends exchange routing information using type 2 EVPN routes to learn the subnets belonging to the same L2VPN behind a VTEP

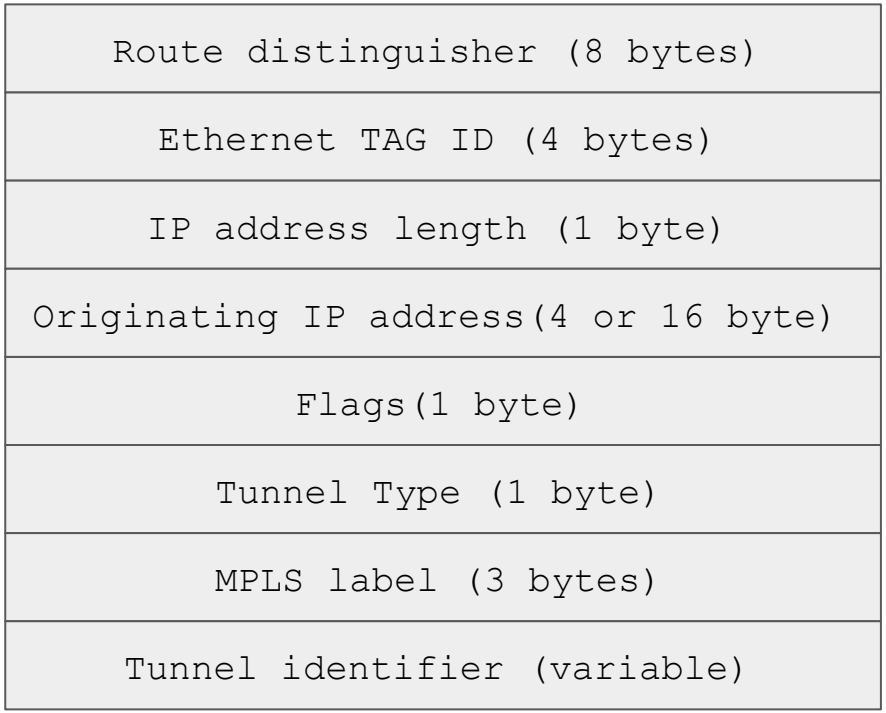


EVPN Type 3

- Type 3 routes are used to advertise L2VNIs and VTEP IP addresses among VTEPs to create the replication list
- So they are used to automatically discover VTEPs and dynamic VLAN tunnel establishment → when a type 3 route is received, a VXLAN tunnel is setup from the local VTEP to the peer VTEP
- Basically the type 3 is used to discover the VXLAN terminations of all the configured L2 VPNs in the overlay BGP network
- This information is contained in another Extended Community field of BGP called PMSI (P-Multicast Service Interface)

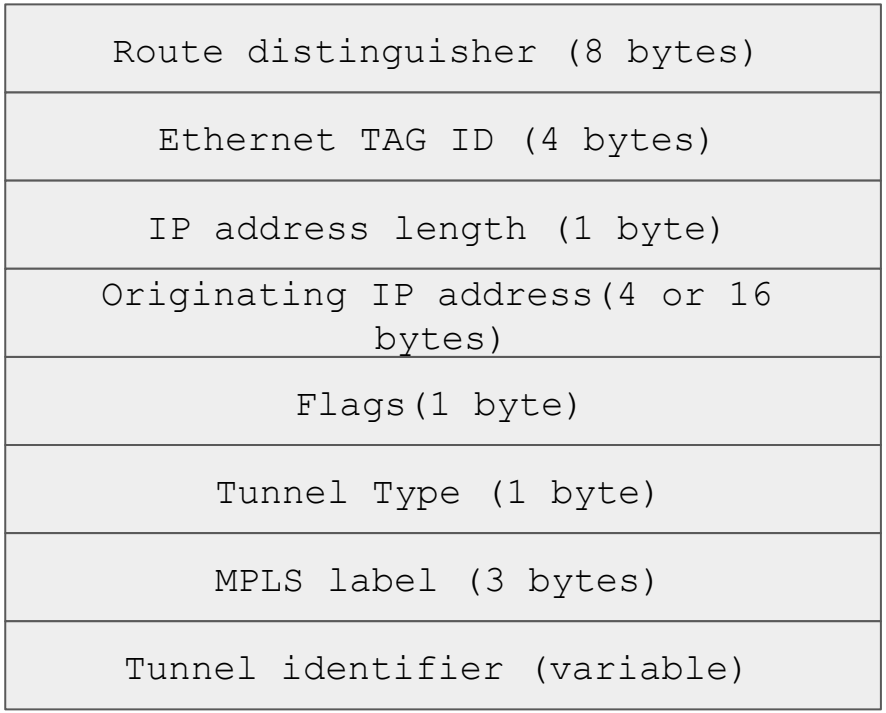
EVPN Type 3 format

- ***Route distinguisher*** → same as before
- ***Ethernet TAG ID*** → contains the VLAN id configured on the device
- ***IP address length*** → Mask length of the local VTEP IP address
- ***Originating IP address*** → Local VTEP IP address originating the route



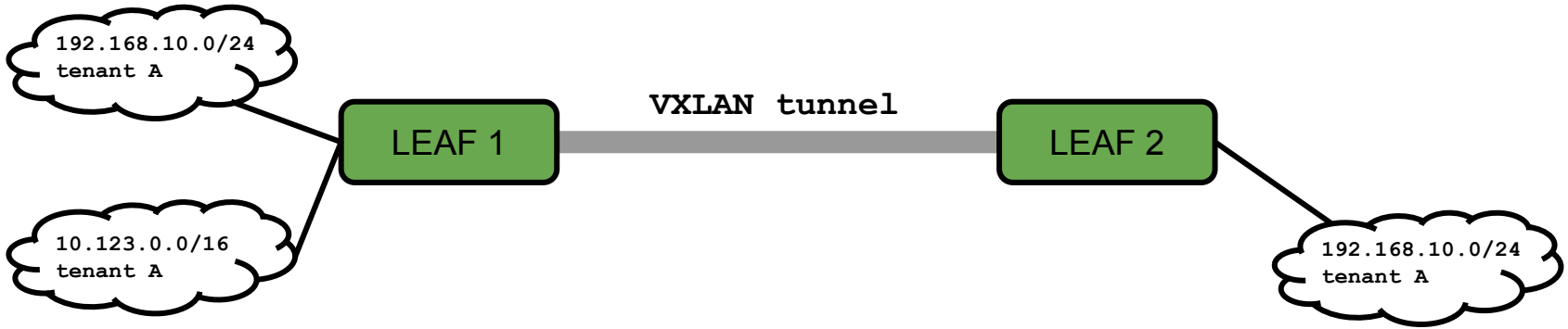
EVPN Type 3 format

- ***Flags*** → Additional information not used in VXLAN
- ***Tunnel Type*** → Tunnel type carried in the route (in VXLAN it is 6)
- ***MPLS label*** → L2VNI carried in the route
- ***Tunnel Identifier*** → Tunnel information carried in the route. It contains also the local VTEP address

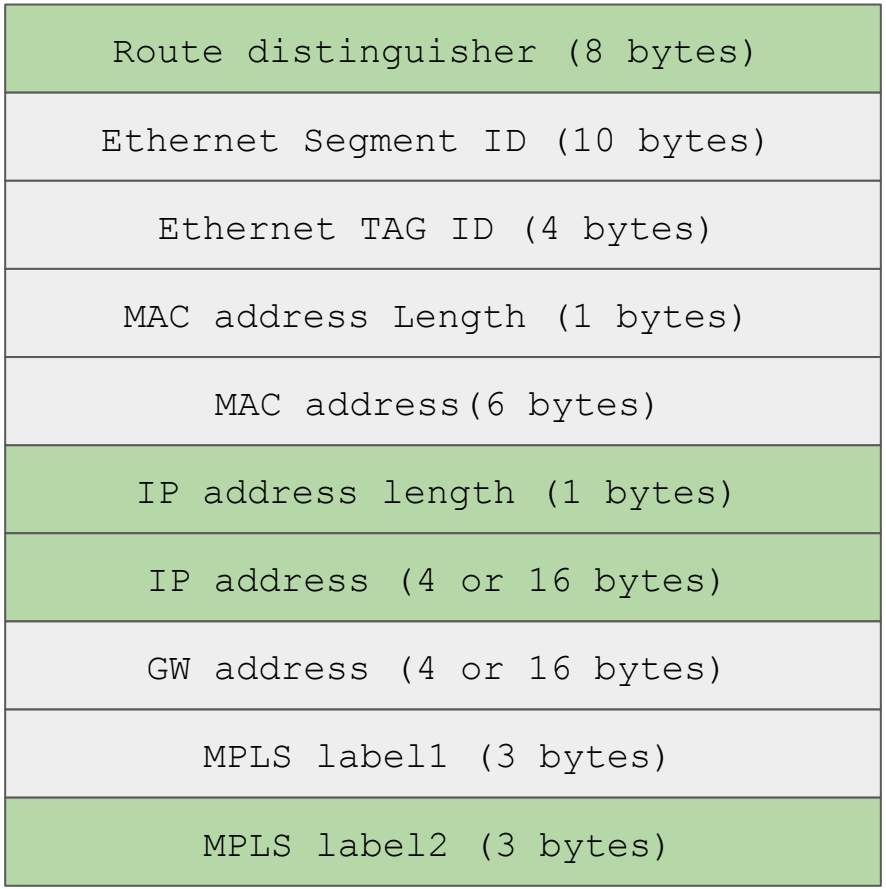


EVPN Type 5

- Type 5 routes are similar to the Type 2 routes for host IP advertisement of host /32 (or /128 for IPv6) prefixes
- In the case of Type 5, you can advertise subnets with variable prefix lengths
- Used for L3VNIs reachability information not for L2VNI Broadcast domains
- So, it is used for inter-subnet routing among the L2VNIs of a same tenant



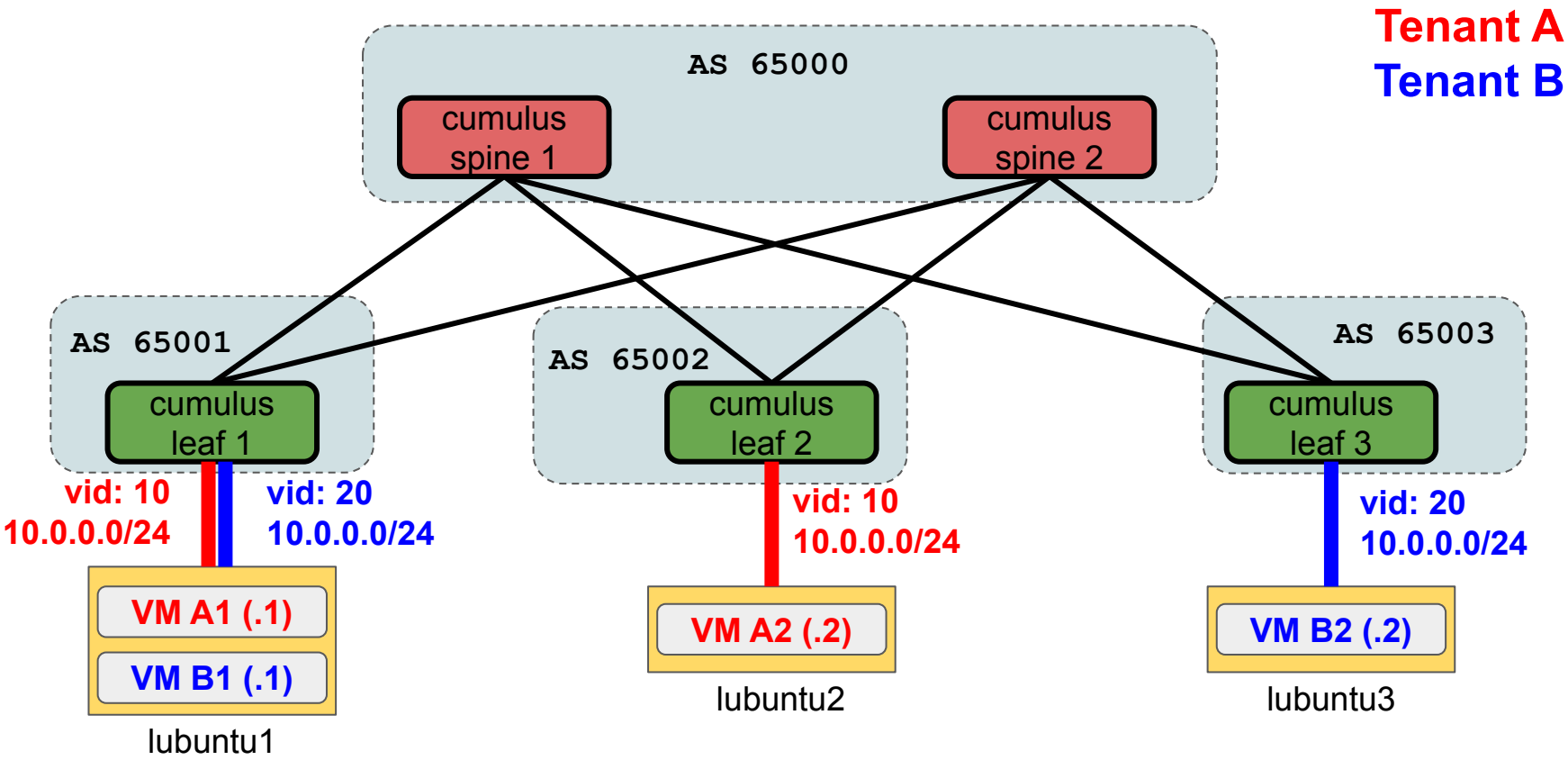
EVPN Type 5


Route distinguisher (8 bytes)
Ethernet Segment ID (10 bytes)
Ethernet TAG ID (4 bytes)
MAC address Length (1 bytes)
MAC address(6 bytes)
IP address length (1 bytes)
IP address (4 or 16 bytes)
GW address (4 or 16 bytes)
MPLS label1 (3 bytes)
MPLS label2 (3 bytes)

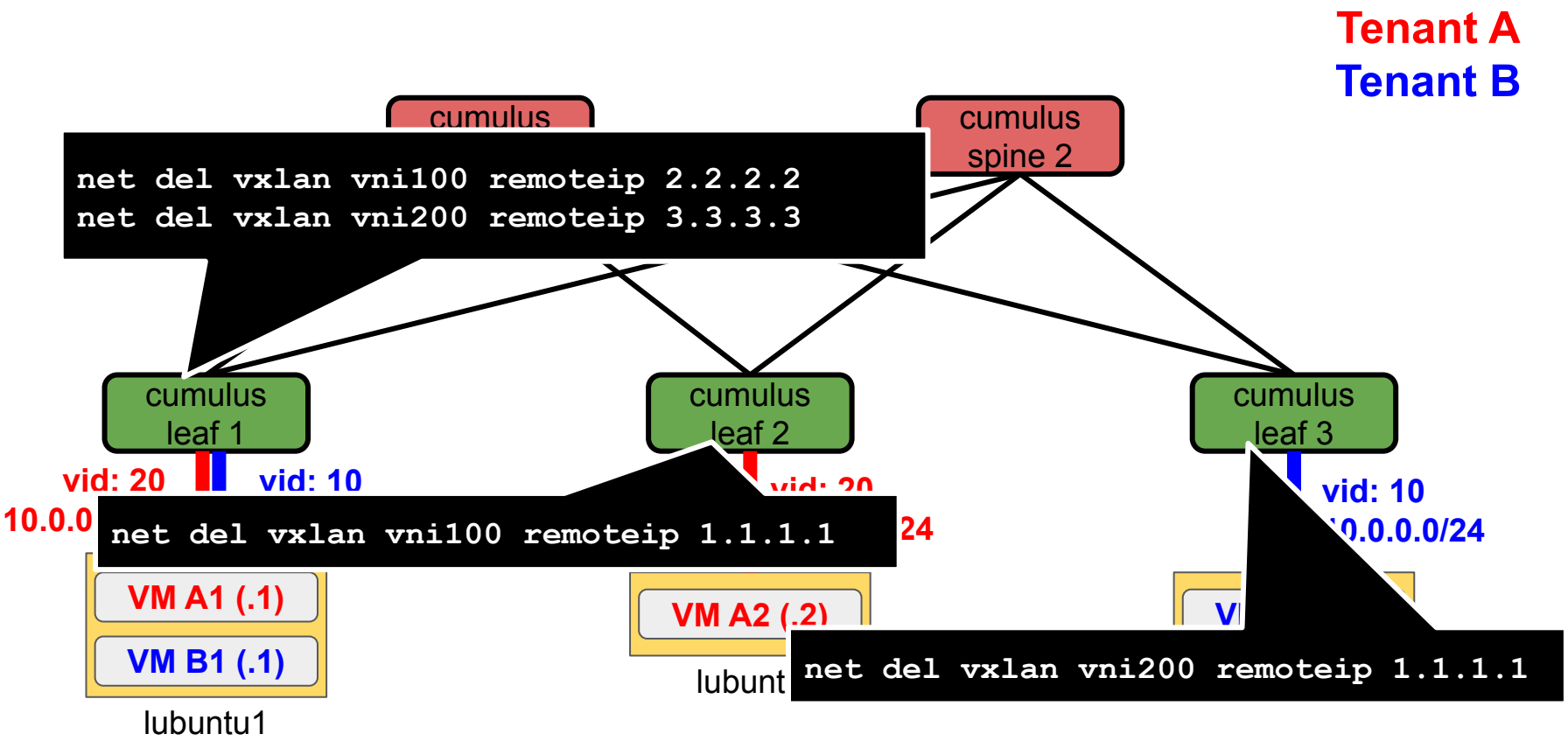
LAB: VXLAN with BGP EVPN in leaf-spine fabric

starting from the previous LAB configuration

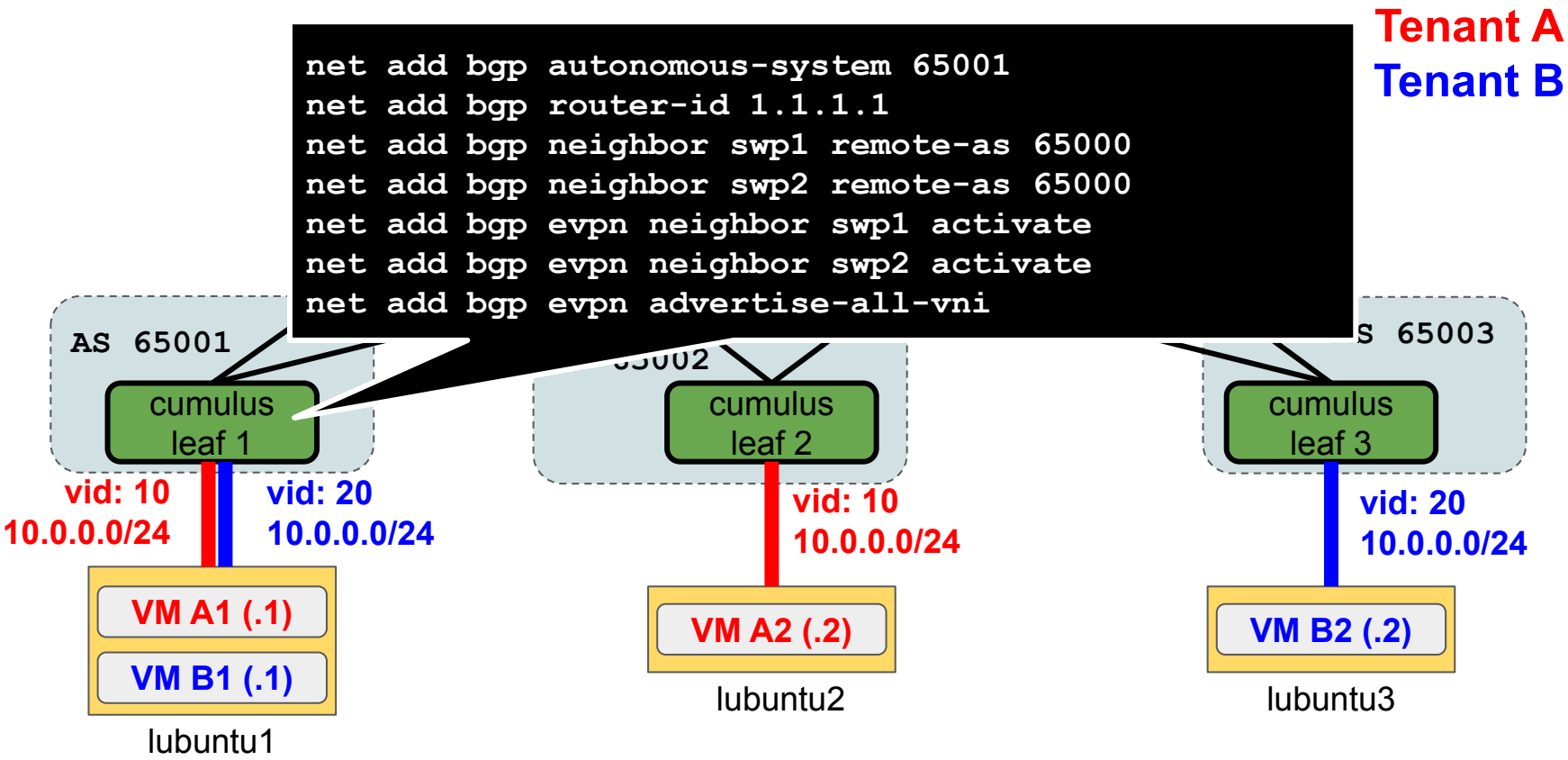
Add MP-eBGP peerings



Remove the static VXLAN VTEPs of previous lab



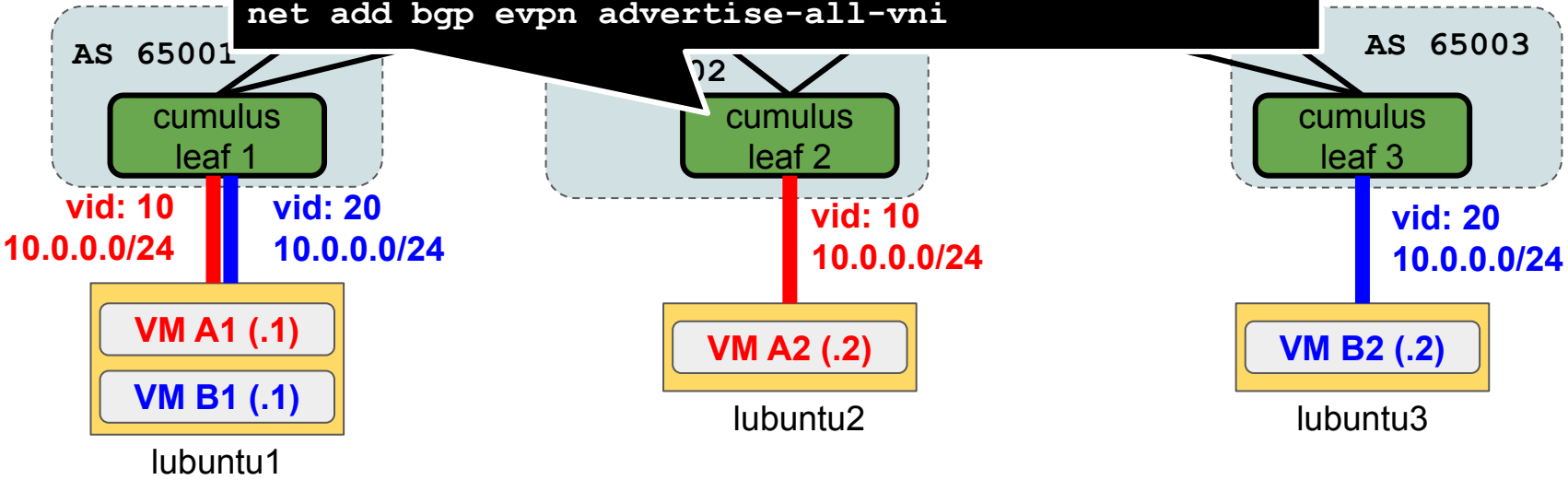
Add MP-eBGP peerings



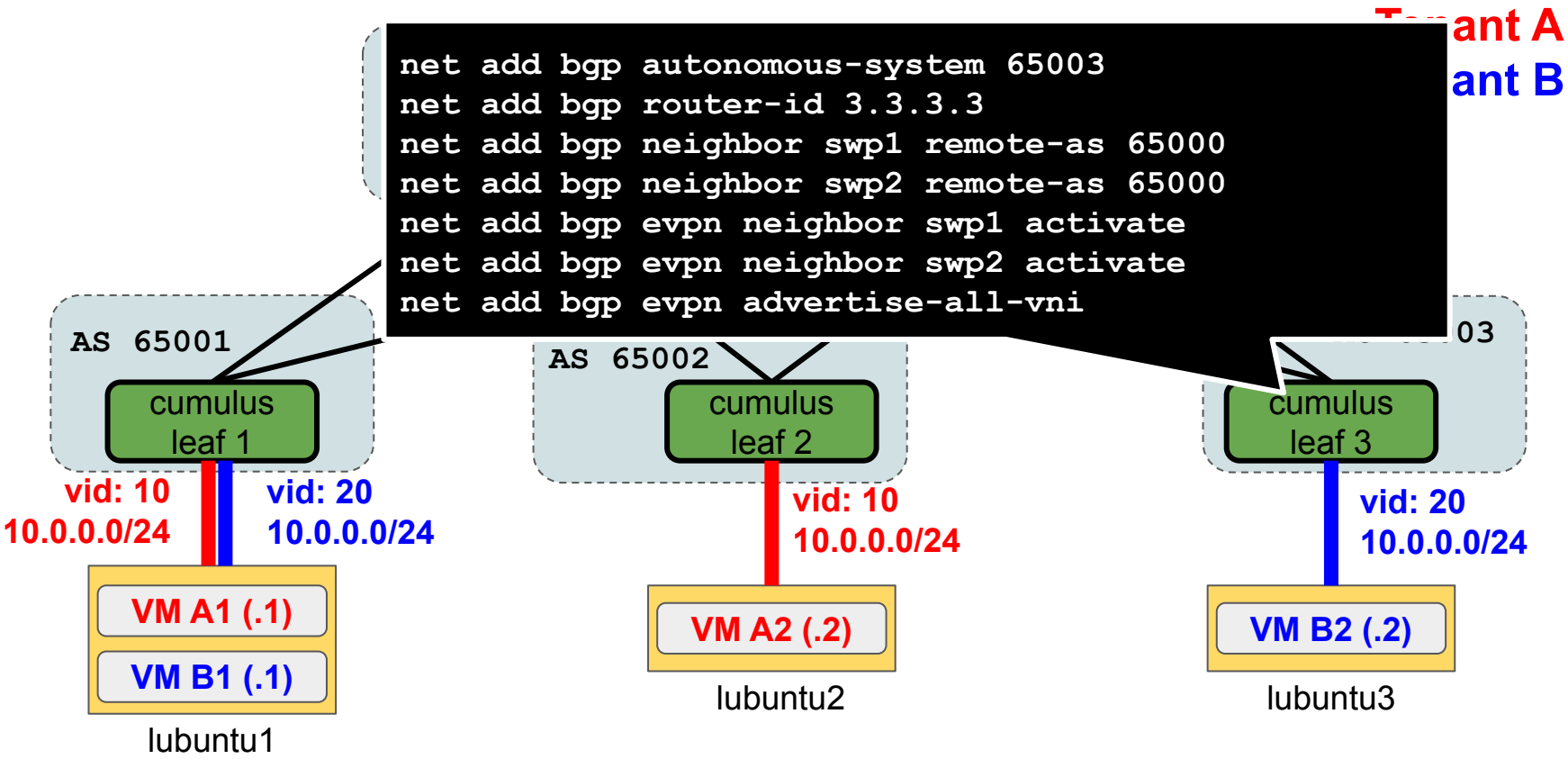
Add MP-eBGP peerings

```
net add bgp autonomous-system 65002
net add bgp router-id 2.2.2.2
net add bgp neighbor swp1 remote-as 65000
net add bgp neighbor swp2 remote-as 65000
net add bgp evpn neighbor swp1 activate
net add bgp evpn neighbor swp2 activate
net add bgp evpn advertise-all-vni
```

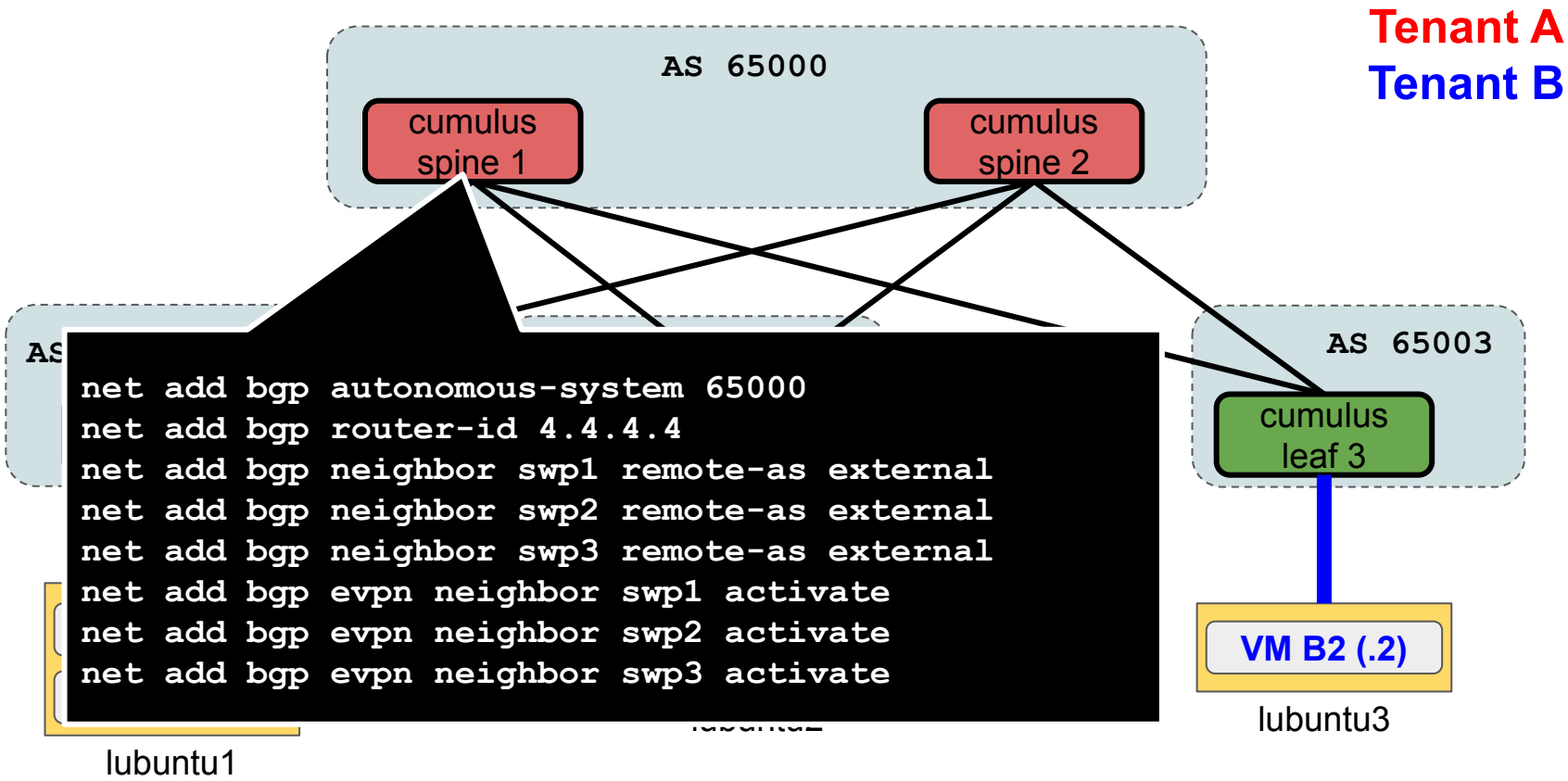
Tenant A
Tenant B



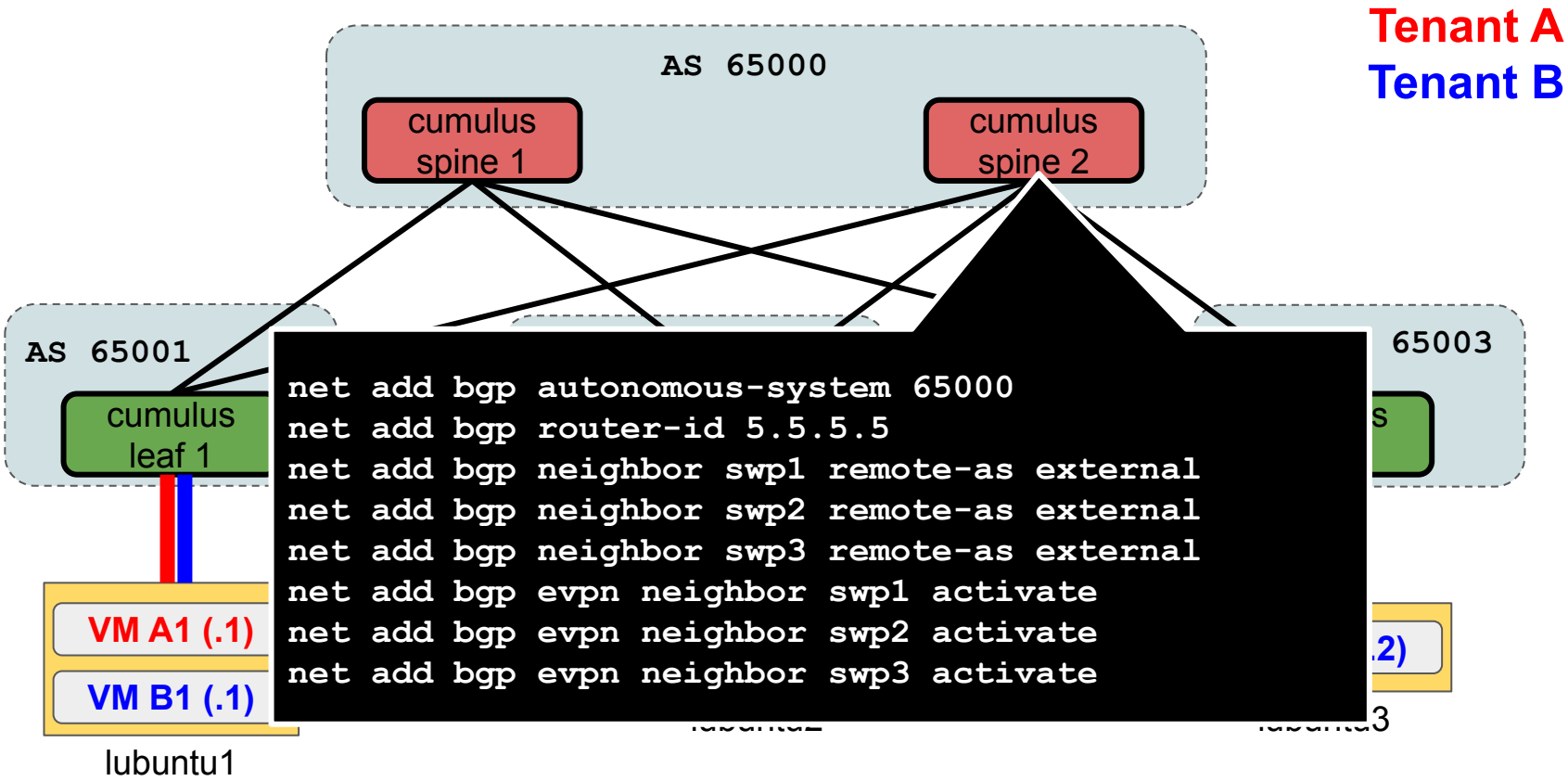
Add MP-eBGP peerings



Add MP-eBGP peerings



Add MP-eBGP peerings



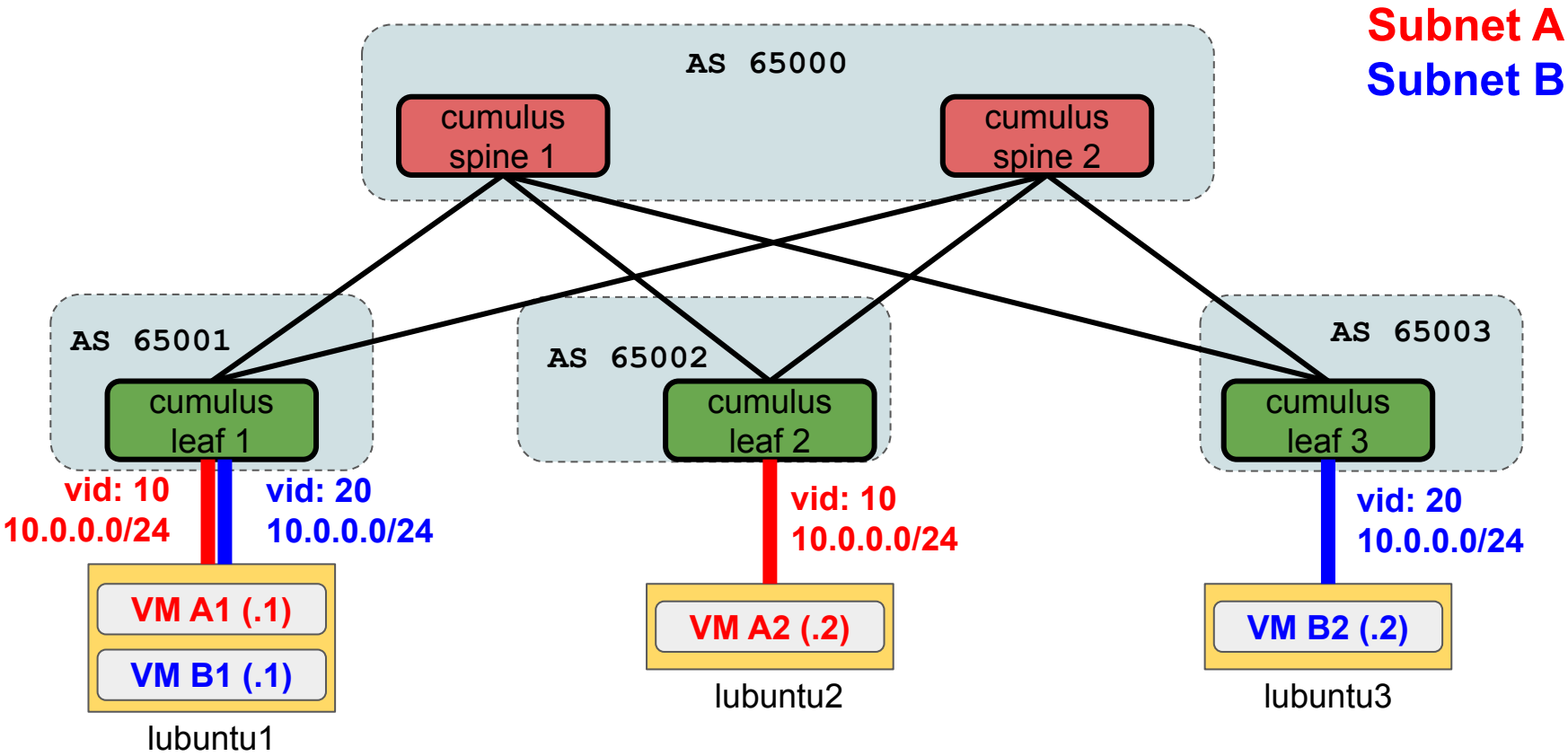
LAB: VXLAN with BGP EVPN in leaf-spine fabric

let’s add L3VNIs

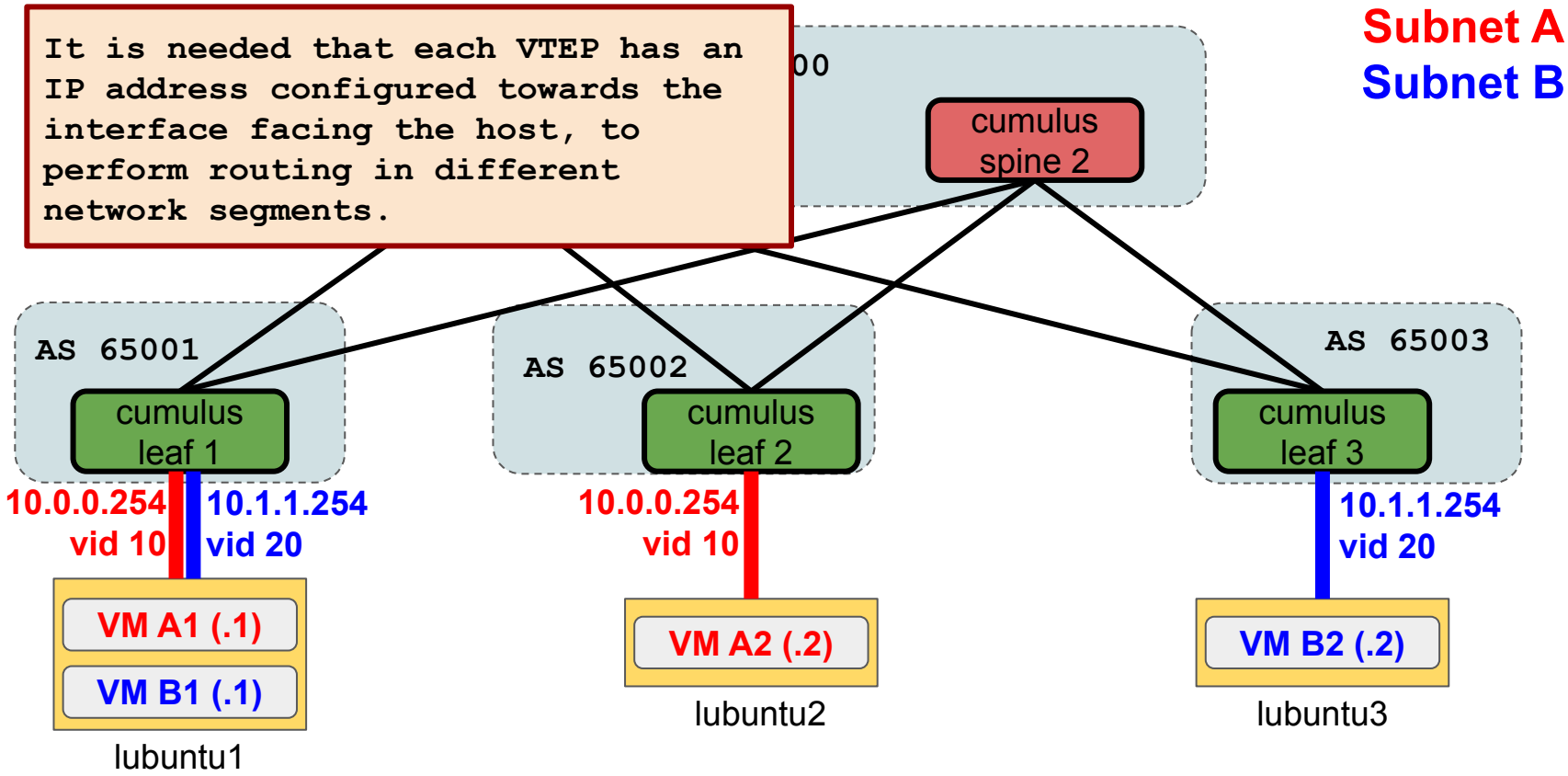
Scenario

- Let’s assume that Tenant A and Tenant B ***merge in a single tenant***
- VM A1 and VM A2 will be in the same broadcast domain, and belong to the network segment with address 10.0.0.0/24 (as before)
- VM B1 and VM B2, as well, will be in the same broadcast domain, and belong to the network segment with address 10.1.1.0/24
- We want to make these network segments reachable between each other
- We do this by creating an L3VNI 1020, common to both broadcast domains, i.e. L2VNI 100 and L2VNI 200

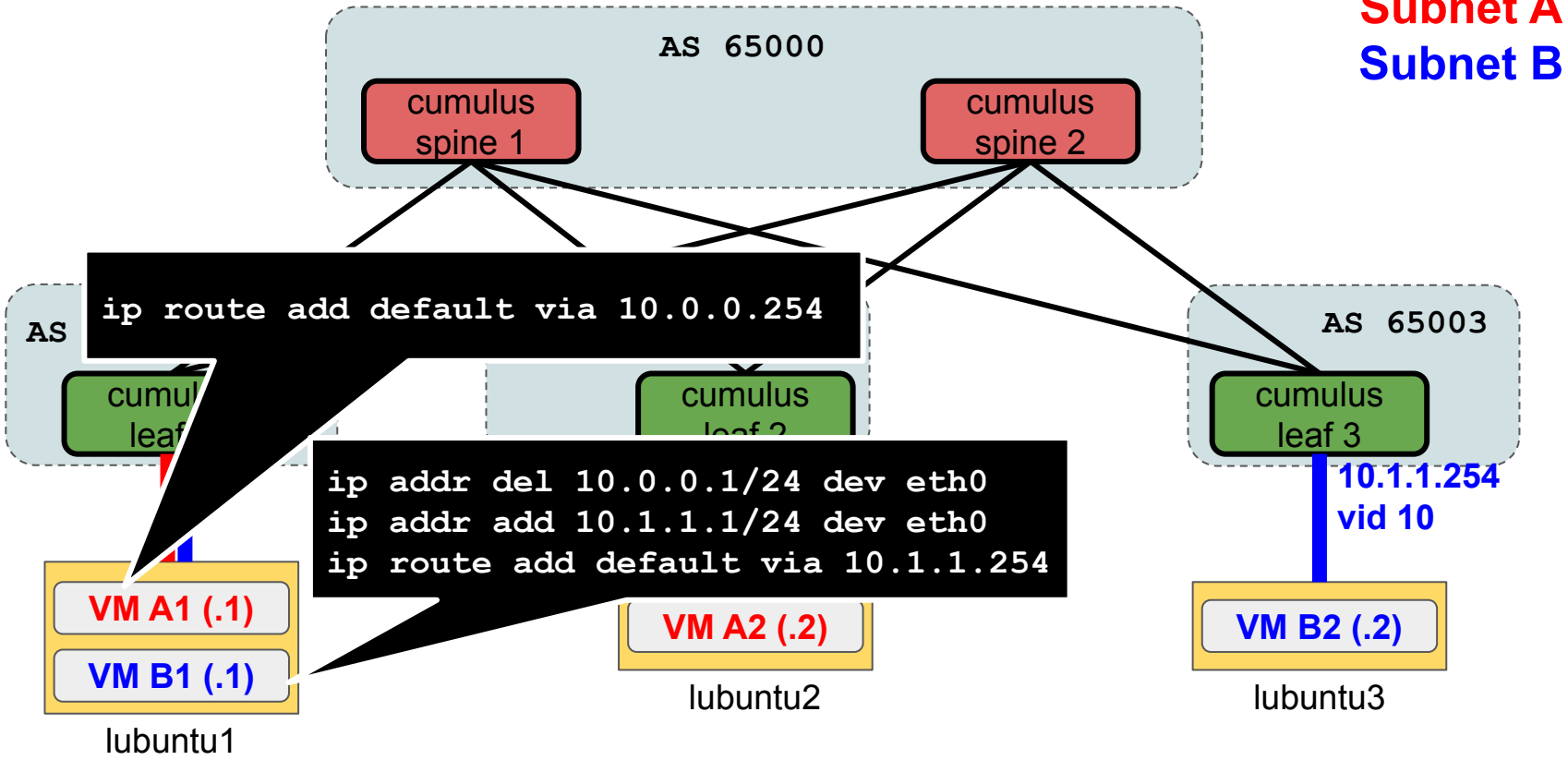
Topology


Subnet A
Subnet B
AS 65000
cumulus spine 1
cumulus spine 2
AS 65001
cumulus leaf 1
AS 65002
cumulus leaf 2
AS 65003
cumulus leaf 3
vid: 10
10.0.0.0/24
vid: 20
10.0.0.0/24
vid: 10
10.0.0.0/24
vid: 20
10.0.0.0/24
VM A1 (.1)
VM B1 (.1)
lubuntu1
VM A2 (.2)
lubuntu2
VM B2 (.2)
lubuntu3

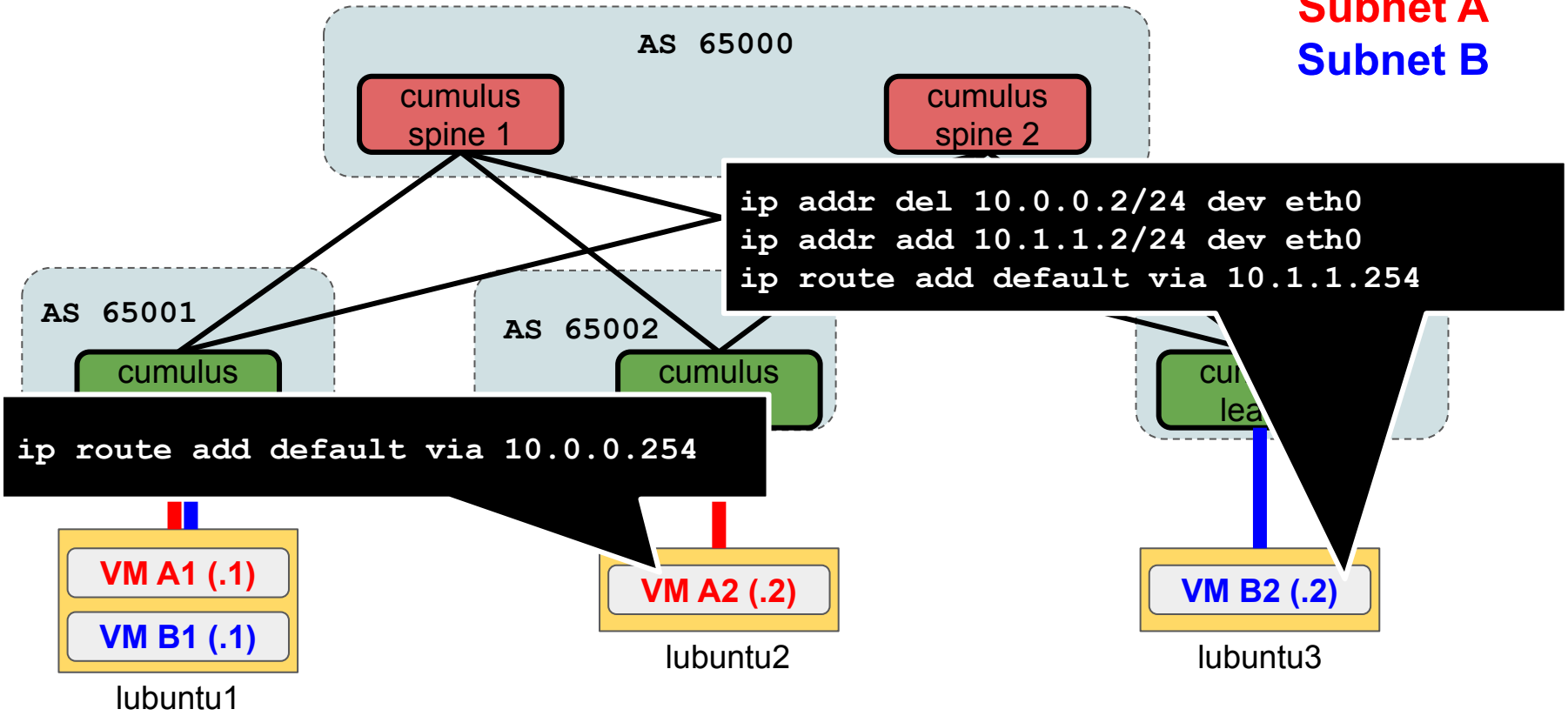
VTEP IP address in each subnet



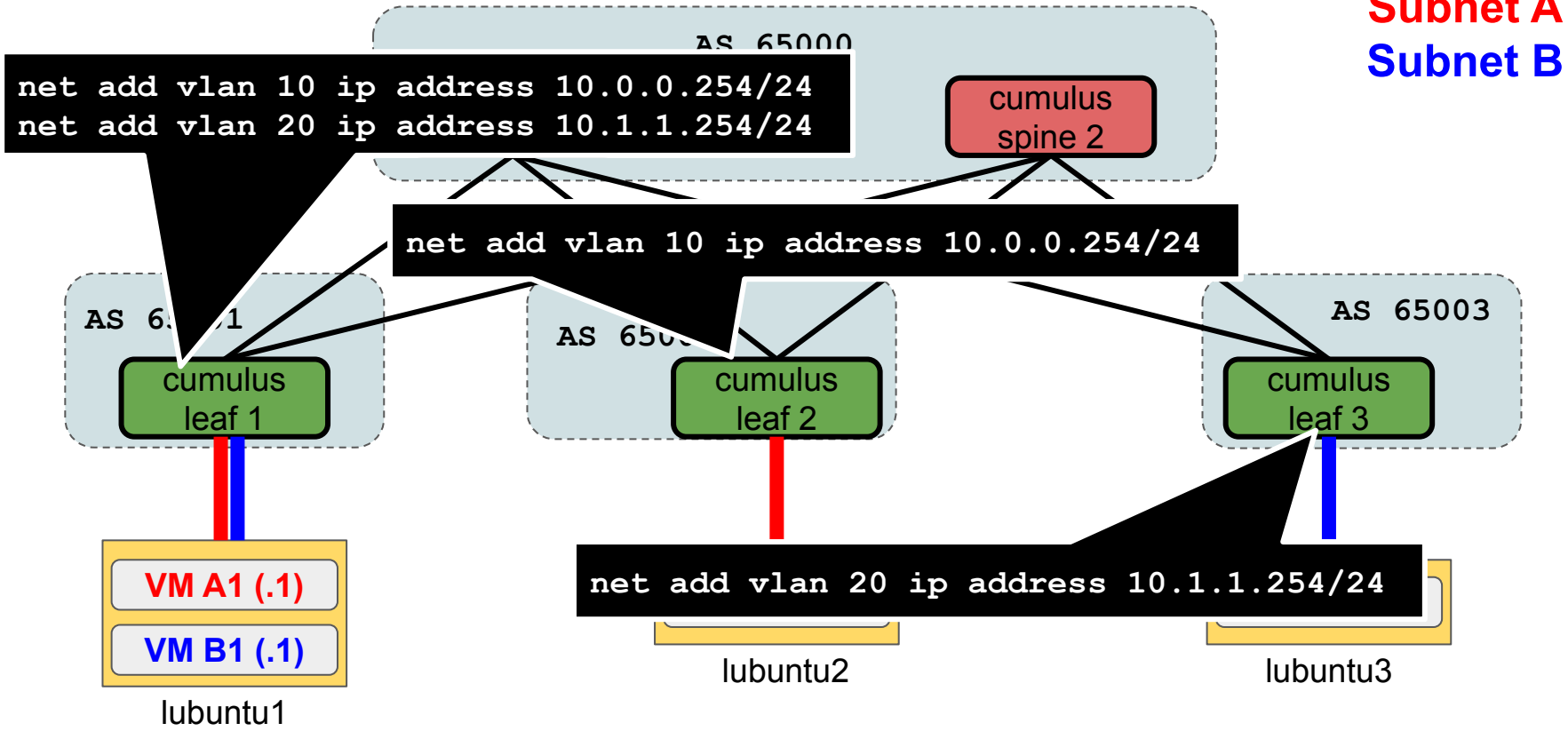
Add default route via VTEP IP address



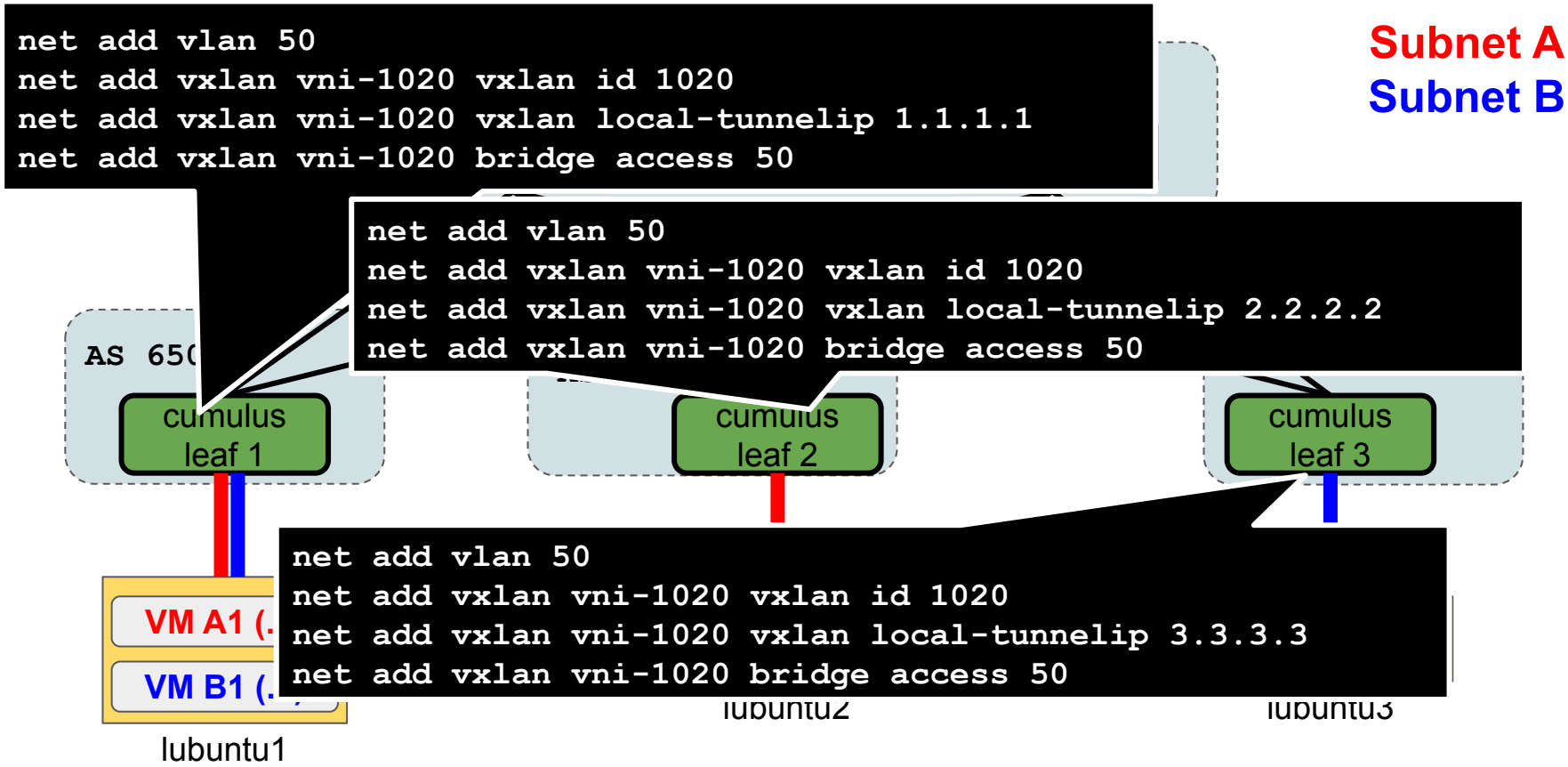
VTEP IP address in each subnet



Add IP address in VTEPs



Add a new VXLAN interface for L3VNI



Bridge VIDs and VNIs and create tenant VRF