

# Configuring Overlay Protocols

---



**Sean Douglas**

DATA CENTER ENGINEER

@ocdlearning



# Overview



Overlay protocols provide flexibility, scalability, and manageability

Examine overlay protocols used to connect geographically disparate L2 data centers over L3 networks

- Cisco OTV
- VXLAN



OTV



# IP Connectivity Between Data Centers



The only requirement from the transport infrastructure is providing IP connectivity between remote data center sites



OTV does not extend the Spanning Tree Protocol (STP) across sites. It also provides other benefits, such as ARP caching, multicast optimization



OTV provides an overlay that enables Layer 2 connectivity between separate switched domains; keeping these domains independent and preserving the fault-isolation, resiliency, and load-balancing

# Cisco OTV



Modern data centers must meet different requirements for speed and flexibility to accelerate application deployment and fulfill operations needs



Extended Layer 2 connectivity between data centers is required to enable server clustering across data centers and cloud integration



OTV provides Layer 2 extension capabilities over any transport infrastructure: Layer 2, Layer 3, MPLS, and so on



OTV provides fault-isolation, resiliency, and load-balancing as well as ARP caching, and multicast optimization



# Cisco OTV Characteristics

## MAC routing

CCP used to exchange  
MAC reachability info  
No data plane flooding

## L2 frames into L3 packets

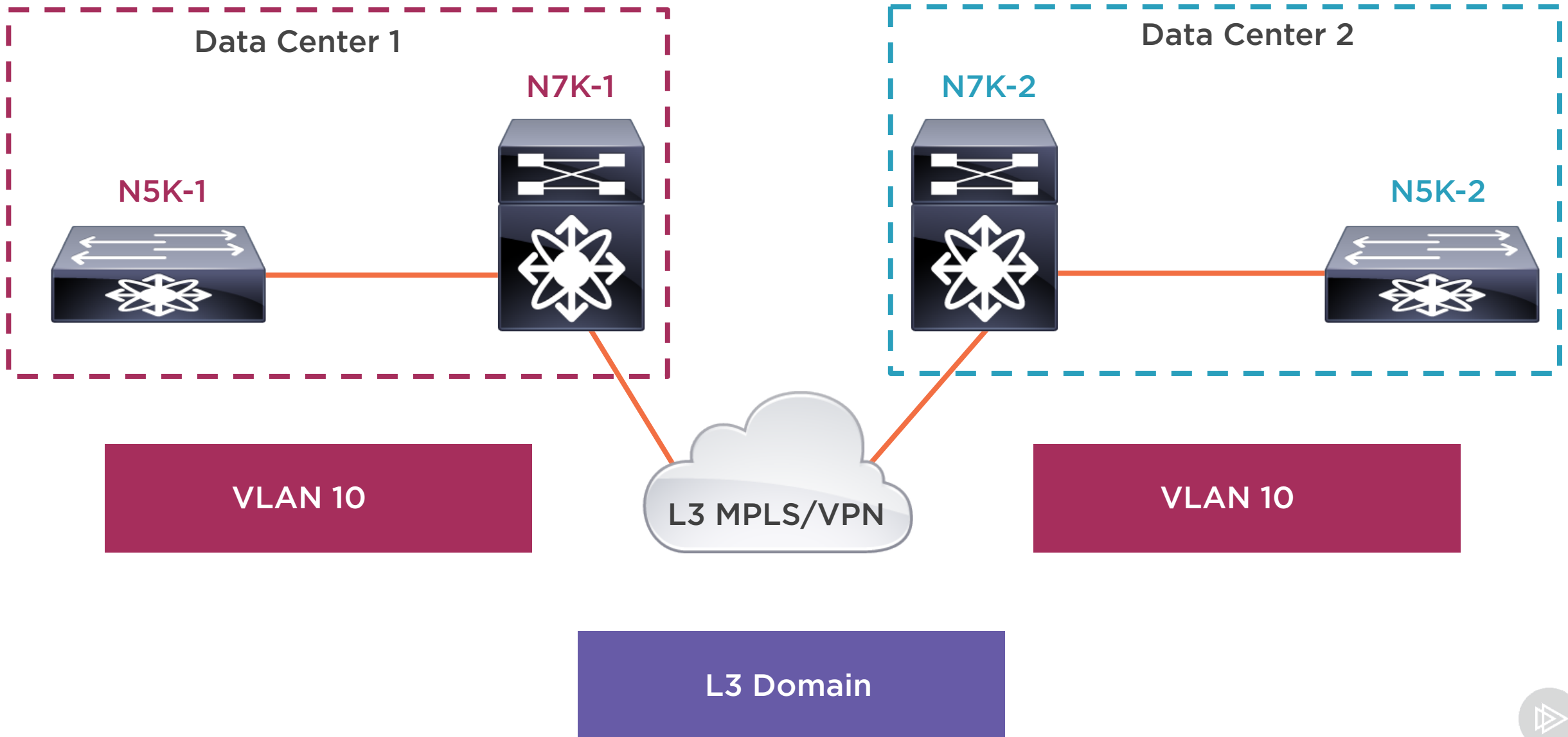
No MPLS  
No pseudowires

## Native multihoming

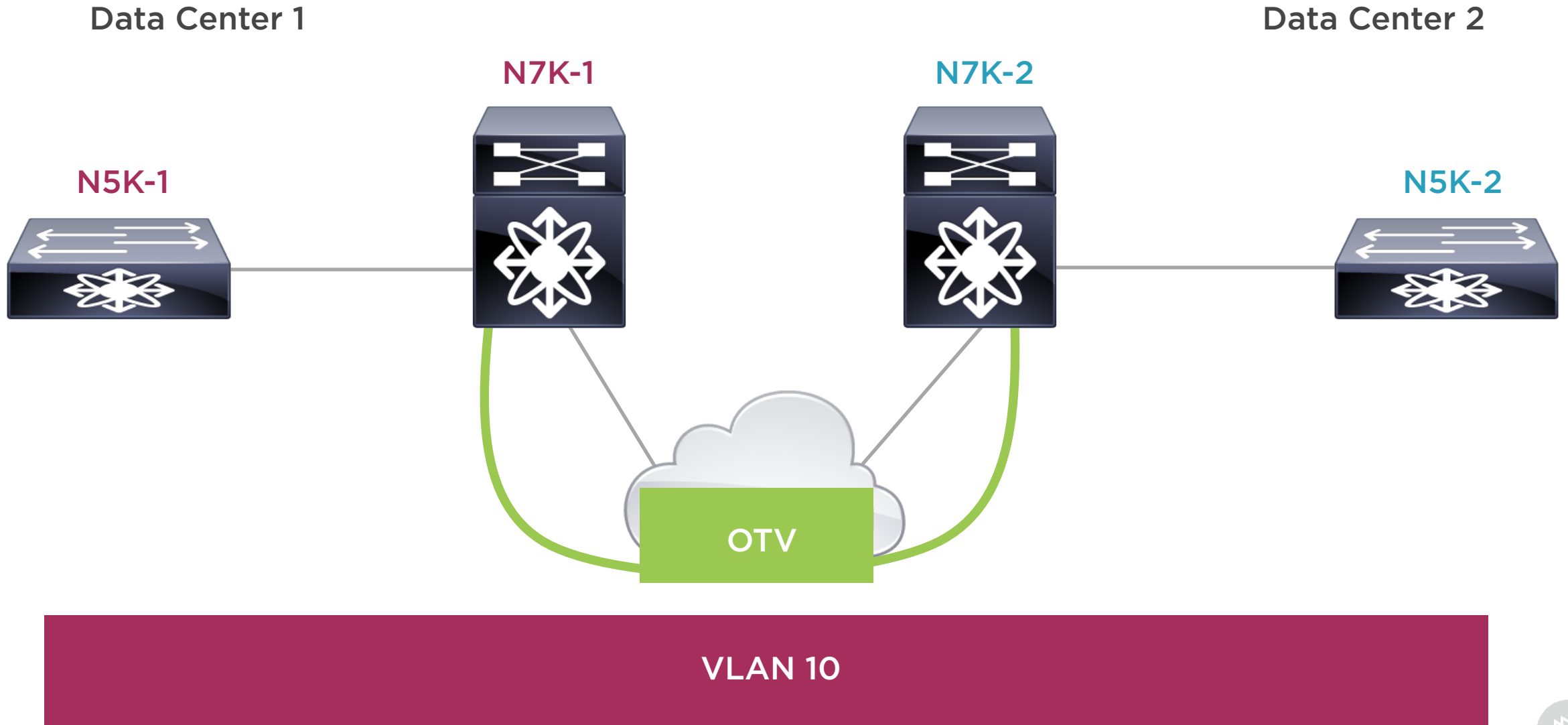
No STP  
ARP caching



# OTV Data Center Extension



# OTV Extends VLANs over IP





# OTV Operation

---



# OTV Operation



OTV uses MAC routing - this is a control plane protocol used to exchange the MAC address info between network devices



This varies from traditional switching that uses data plane learning. Limits flooding of the Layer 2 traffic across the network

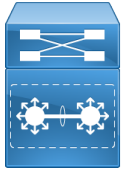


OTV encapsulation is performed at the OTV edge devices



If the destination MAC address information is unknown, switch drops the traffic, instead of flooding it, which prevents wasting bandwidth

# OTV Operation MAC Addresses



OTV edge device learns a new MAC address, creates update message about the MAC address and sends it to all other OTV edge devices



MAC information is added to the OTV edge device Content Addressable Memory (CAM) table



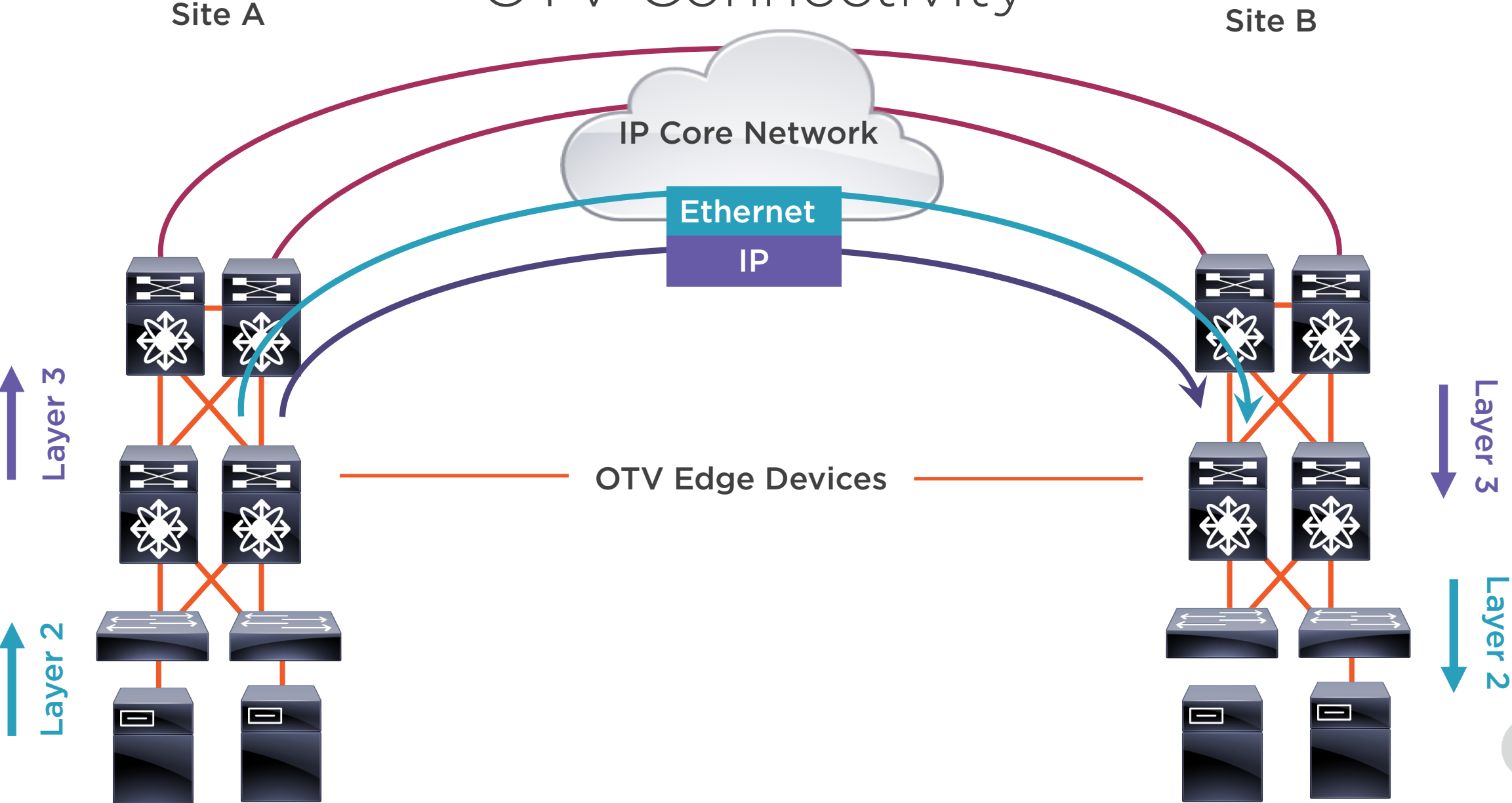
Instead of being associated with a physical interface, the OTV entries refer to the IP address of the originating OTV edge device



OTV edge device receives a Layer 2 frame, it performs a lookup. If it points to IP of remote OTV device, switch sends it to that OTV device



# OTV Connectivity



# OTV CAM Table

Site A

Site B

Layer 3

Layer 3

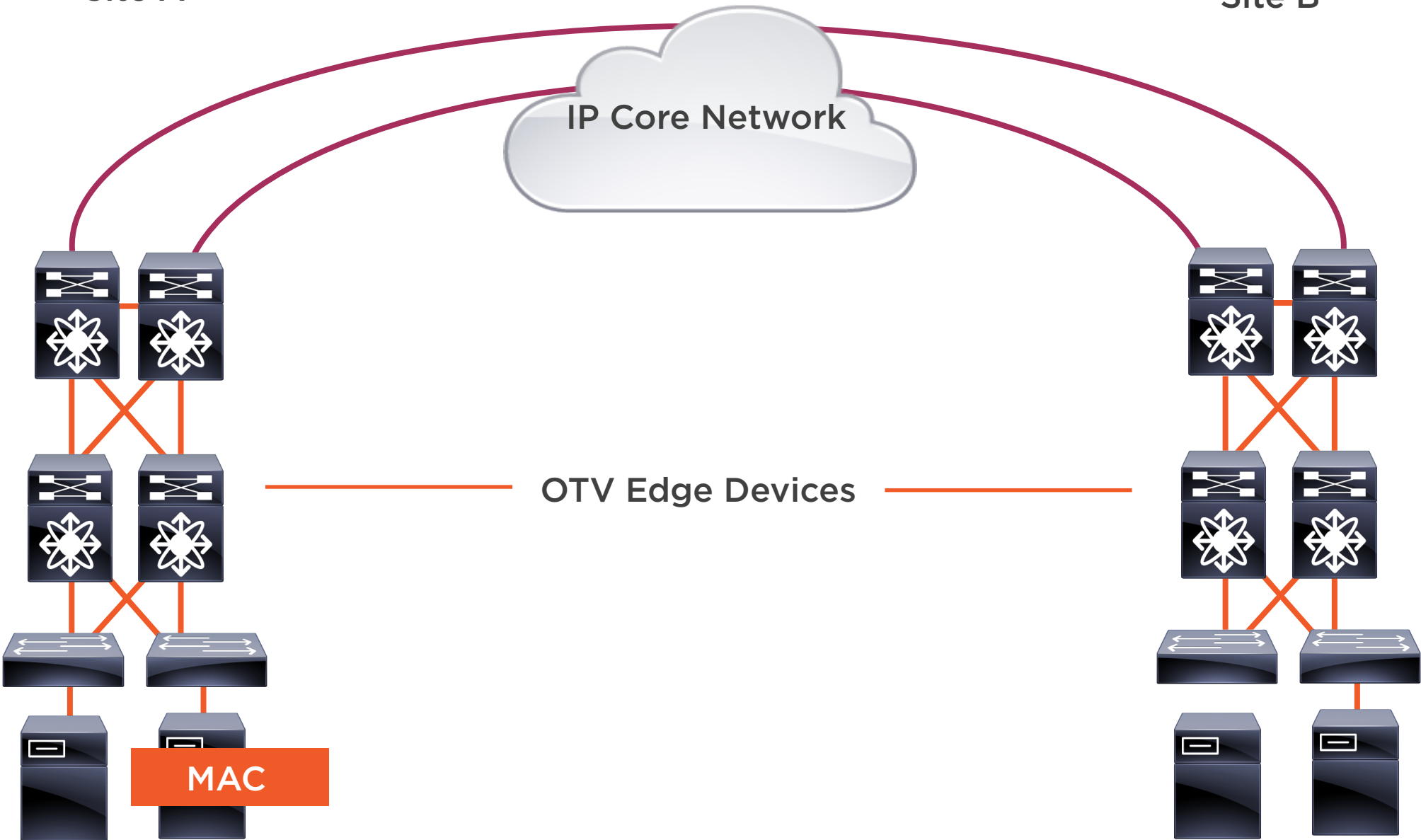
Layer 2

Layer 2

IP Core Network

OTV Edge Devices

MAC



# OTV CAM Table

Site A

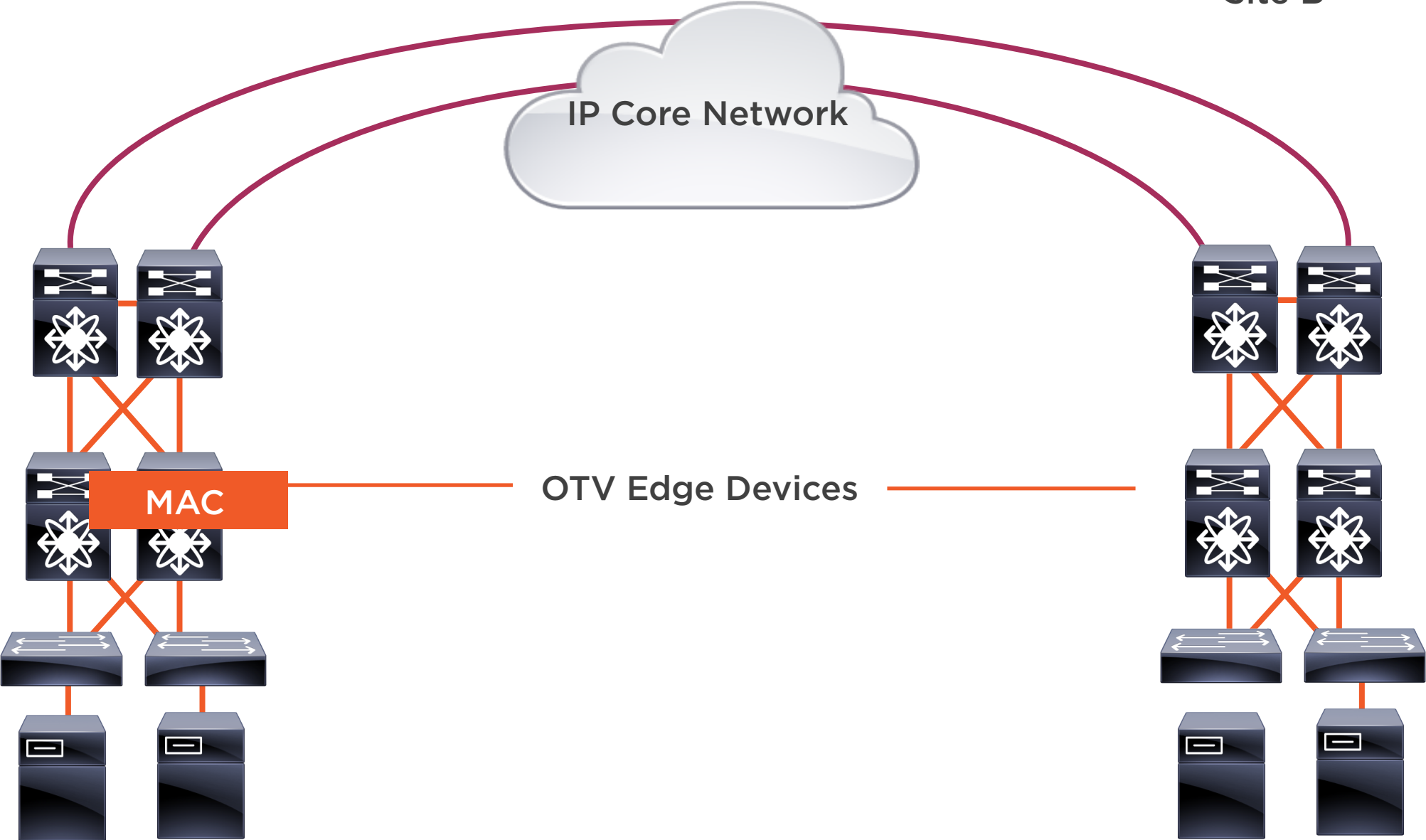
Site B

Layer 3

Layer 3

Layer 2

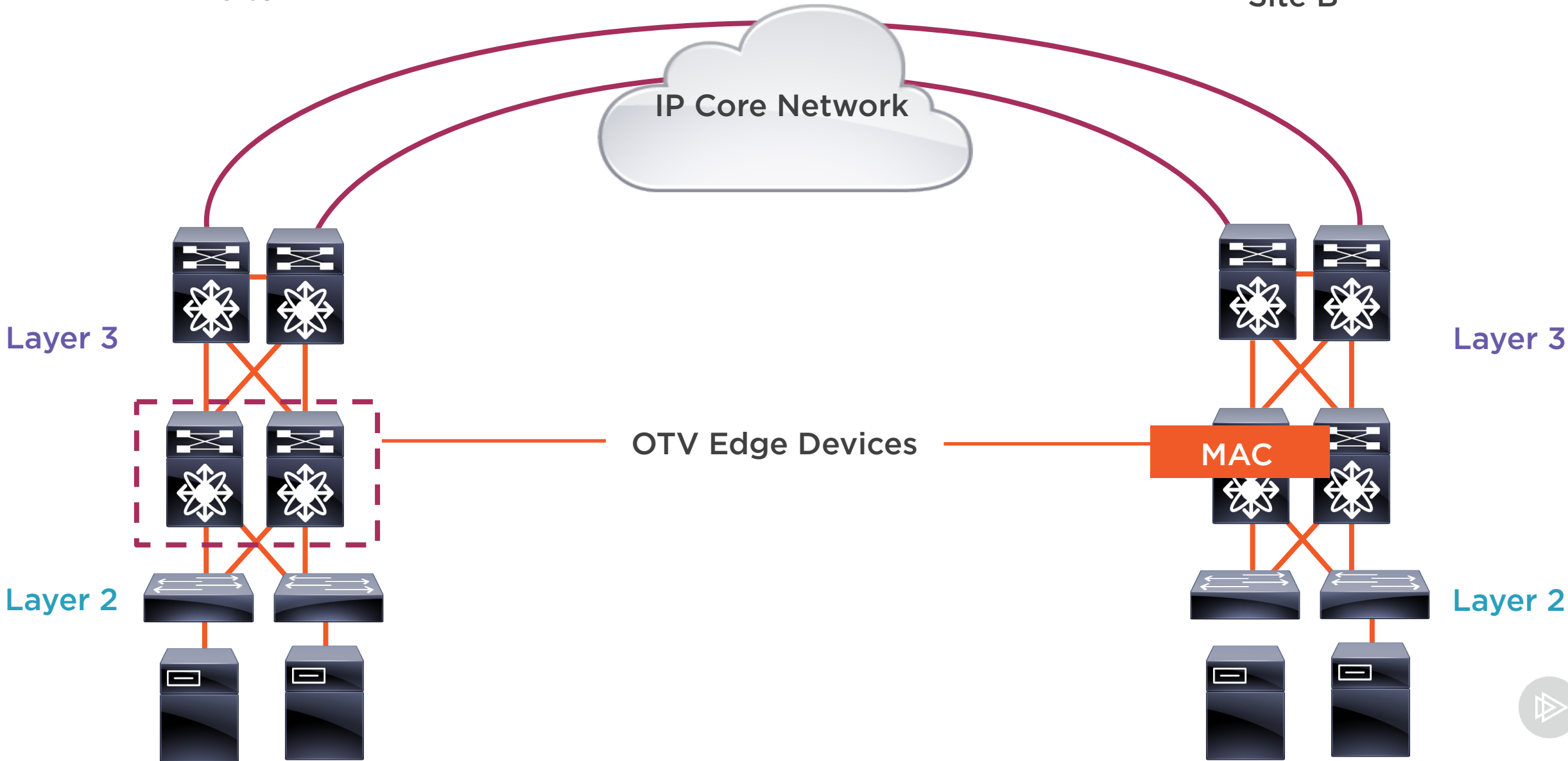
Layer 2



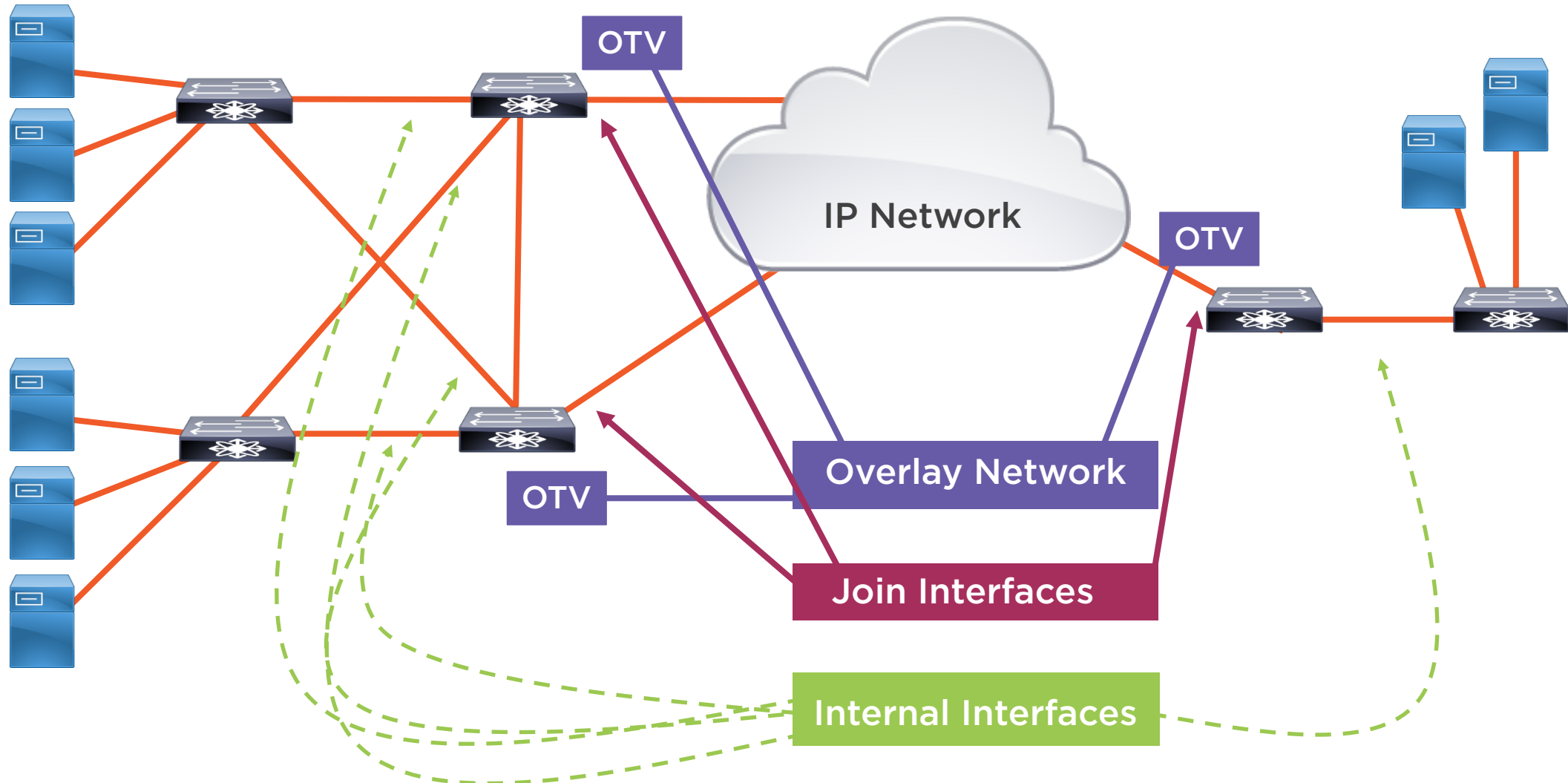
# OTV CAM Table

Site A

Site B



# OTV Components

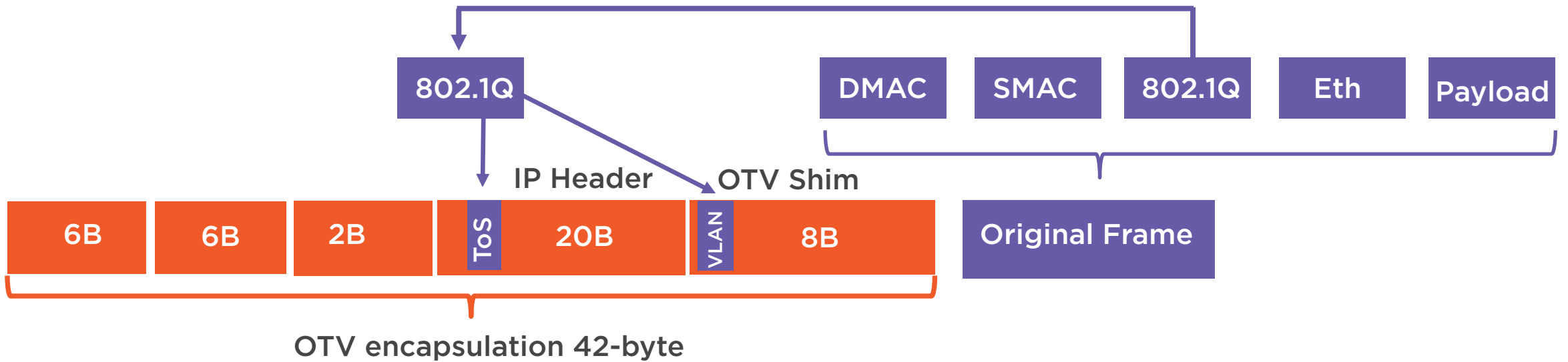




OTV is supported on Nexus  
7000, 7700, 1000V CSR,  
and 1000 ASR



# OTV Data Encapsulation



# Configuring OTV

---



# Guidelines and Limitations for Cisco OTV



Use a separate nondefault VDC for OTV for better manageability and maintenance



An overlay interface will only be operational if the overlay interface configuration is complete and enabled



Configure the join interface and all interfaces that face the core with the highest MTU size that is supported by the core



OTV supports multicast and unicast between sites. If multicast mode is used, the network must support PIM sparse mode or PIM-Bidir



```
N7K-1(config)# feature otv
```

```
N7K-1(config)# otv site-vlan 108
```

```
N7K-1(config-site-vlan)# otv site-identifier 1.1.1
```

## Configuring OTV



## OTV MTU and IGMPv3

```
N7K-1(config)# interface ethernet 2/1
```

```
N7K-1(config-if)# mtu 9000
```

```
N7K-1(config-if)# ip igmp version 3
```

# Configuring OTV

```
N7K-1(config)# interface overlay 1
```

```
N7K-1(config-if-overlay)# otv join-interface ethernet 2/1
```

```
N7K-1(config-if-overlay)# otv control-group 239.1.1.1
```

```
N7K-1(config-if-overlay)# otv data-group 239.1.1.0/28
```



```
N7K-1(config-if-overlay)# otv extend-vlan 100
```

```
N7K-1(config-if-overlay)# no shutdown
```

## OTV VLAN Extension

**Only extend VLANs across the network that you need to extend**

**Do not extend the site VLAN**





```
N7K-1# show otv
```

```
N7K-1# show otv adjacency
```

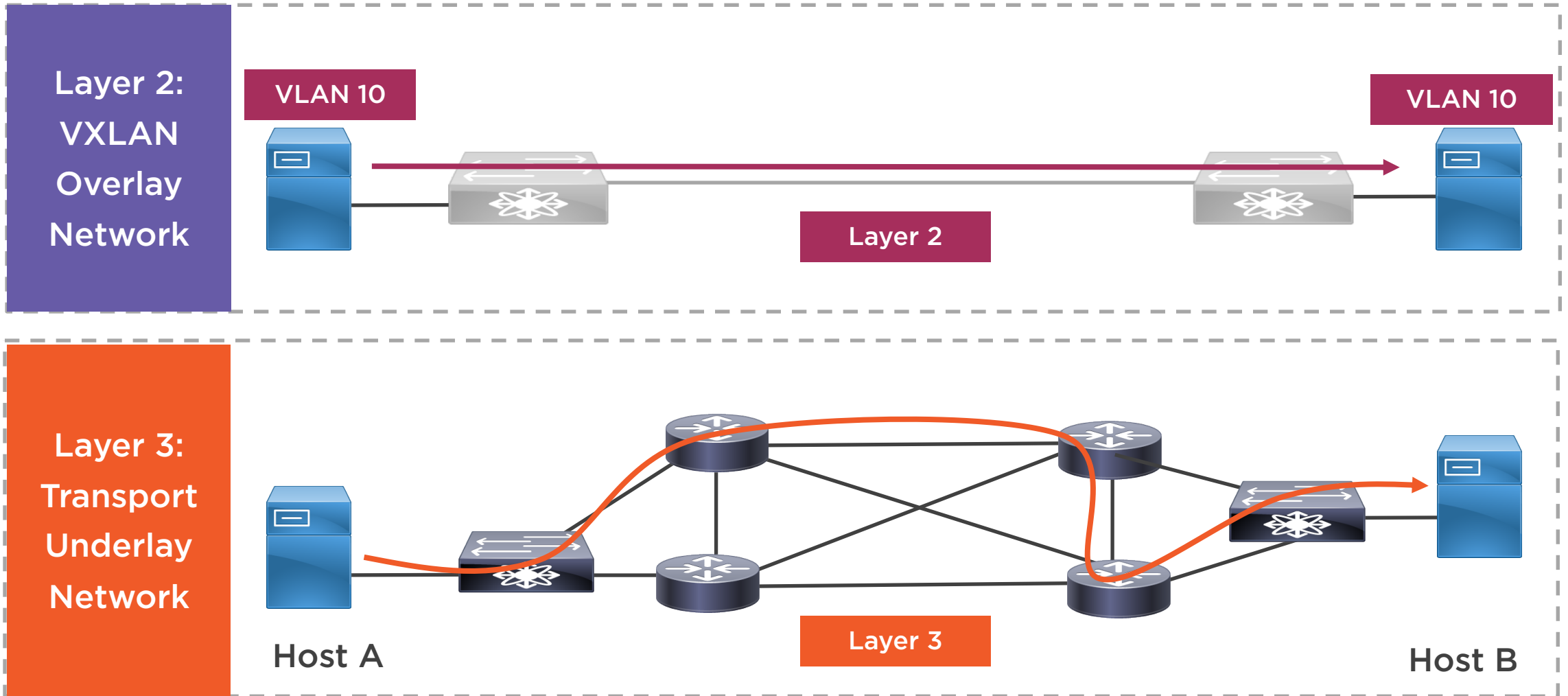
```
N7K-1# show otv route
```

```
N7K-1# copy running-config startup-config
```

Show and save



# NOT USED



# VXLAN

---



# Traditional VLANs



**4094 segments**

**Expansion limited to Layer 2 boundaries**

**Forwarding at Layer 2**

- Loop prevention
- STP

# VXLAN

**16 million segments**

**Expansion can extend beyond Layer 2 boundaries**

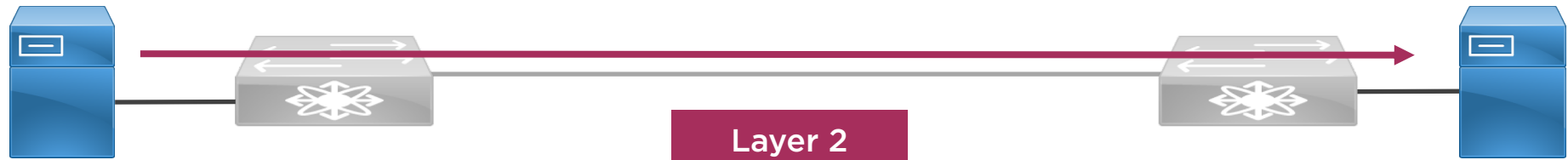
**Forwarding at Layer 3**

- Routing
- Link aggregation
- ECMP

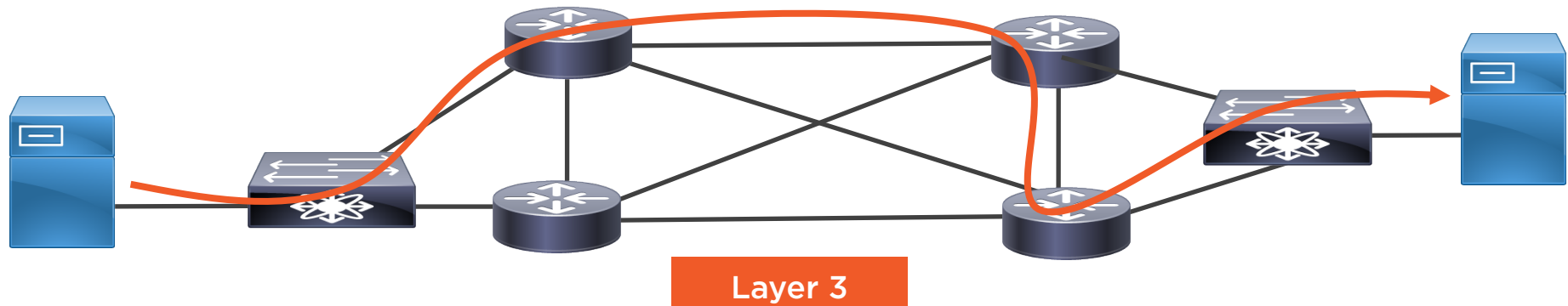


# VXLAN Overlay

Layer 2:  
VXLAN  
Overlay  
Network



Layer 3:  
Transport  
Underlay  
Network



# VXLAN Encapsulation



VXLAN is a Layer 2 overlay using a Layer 3 network that uses encapsulation to extend Layer 2 across the data center network



VXLAN defines MAC-in-UDP encapsulation where the original Layer 2 frame has a VXLAN header added and is then placed in an IP packet



VXLAN header is 8-bytes with a 24-bit VNI used to identify Layer 2 segments and to maintain isolation between the segments



With MAC-in-UDP encapsulation, VXLAN tunnels Layer 2 network over Layer 3 network



# VXLAN MAC-in-UDP Encapsulation



VXLAN uses MAC-in-UDP encapsulation. The original Layer 2 frame has a VXLAN header added and is then placed in a UDP-IP packet



VXLAN header is 8-byte that consists of a 24-bit VNI. This and the original Ethernet frame are encapsulated in the UDP payload



24-bit VNI is used to identify Layer 2 segments and to maintain Layer 2 isolation. VXLAN can support 16 million LAN segments



With this MAC-in-UDP encapsulation, VXLAN tunnels the Layer 2 network over the Layer 3 network





# VTEP



VXLAN uses VXLAN tunnel endpoint (VTEP) devices to map devices in local segments to VXLAN segments. VTEP performs encapsulation and de-encapsulation of the Layer 2 traffic



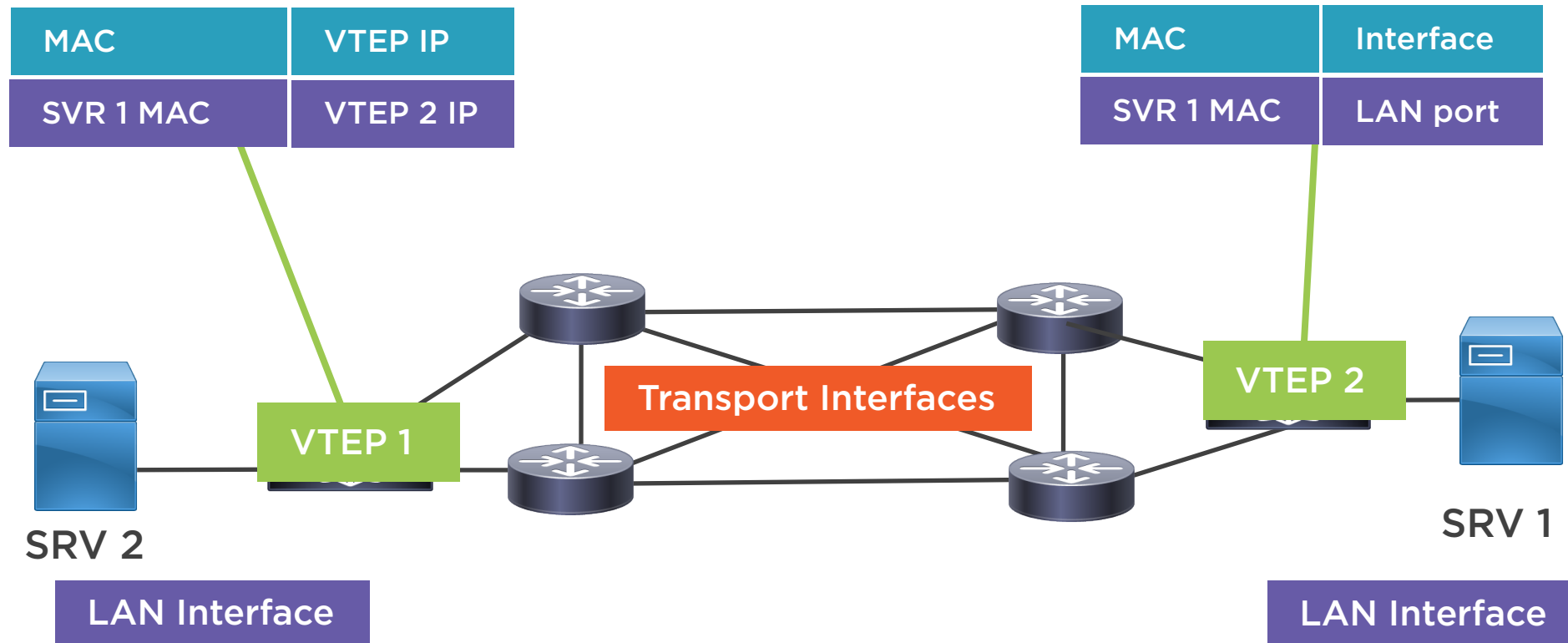
Each VTEP has at least two interfaces: a switch interface on the local LAN and an IP interface in the transport IP network



The encapsulated packets are routed based on the outer IP address header, which has the initiating VTEP as the source IP address and the terminating VTEP as the destination IP address



# VXLAN Transport



Outer IP Header

Outer UDP Header

VXLAN Header

Original Layer 2 Frame



# VXLAN Components

---



# VXLAN Packet Structure



**VXLAN Header:** 24-bit VNID field that identifies the VXLAN segments. Provides expanded address spaces for Layer 2 networks



**UDP header:** destination port 4789 in the UDP header indicates that the packet is a VXLAN encapsulated packet



**Outer IP header:** source IP address in the outer IP header is the local VTEP address. Destination IP address is the remote VTEP address



**Outer MAC address or Layer 2 header:** used to forward the encapsulated packets to the immediate next-hop device



# VXLAN Packet Structure - Find on google

## VXLAN Packet Structure

MAC-in-UDP encapsulation

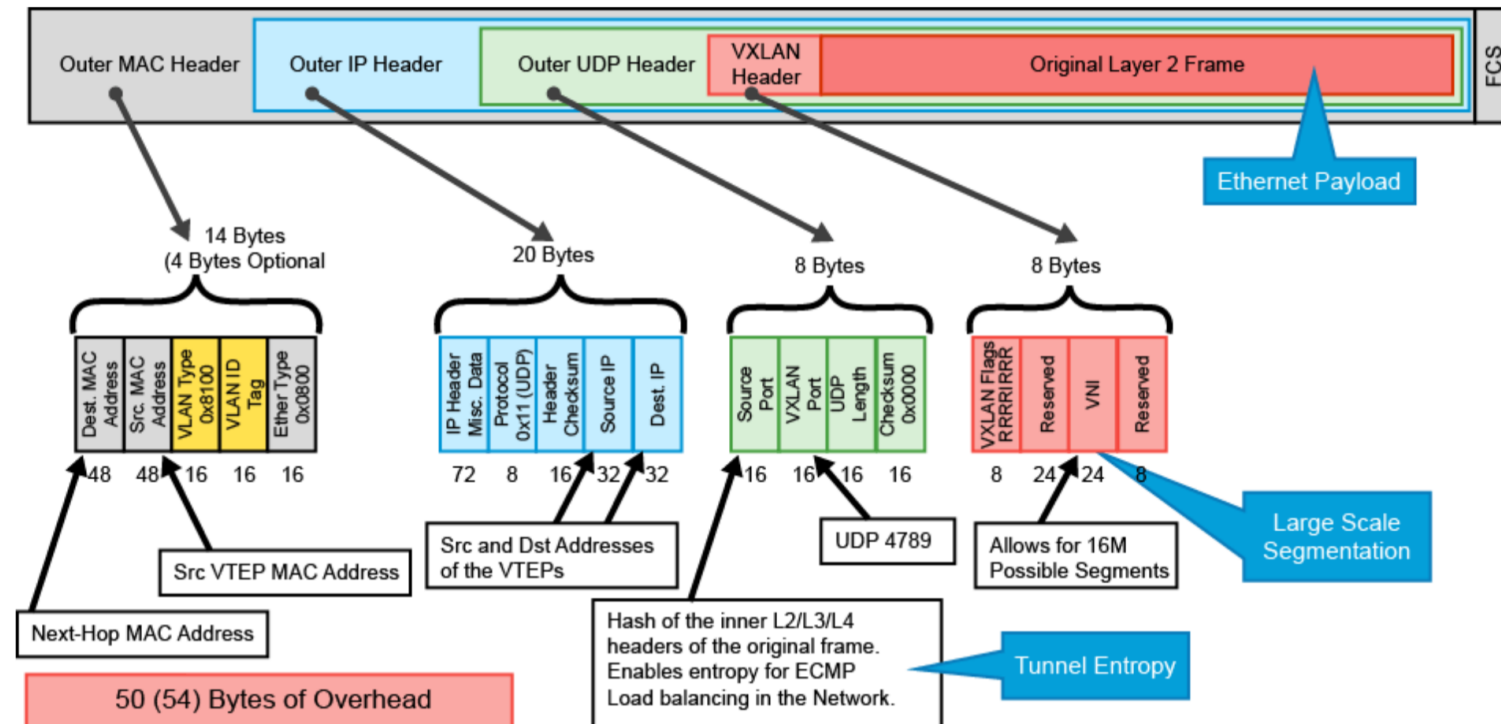


Figure shows the VXLAN packet format. The packet has an 8-byte VXLAN header, UDP header, outer IP header, and outer MAC header.



# VXLAN Control Plane Options



VXLAN scales based on how well VTEP devices handle broadcast, unknown unicast, and multicast (BUM) traffic



VTEP devices use host MAC address to VTEP IP address mappings to forward encapsulated frames across the IP transport network



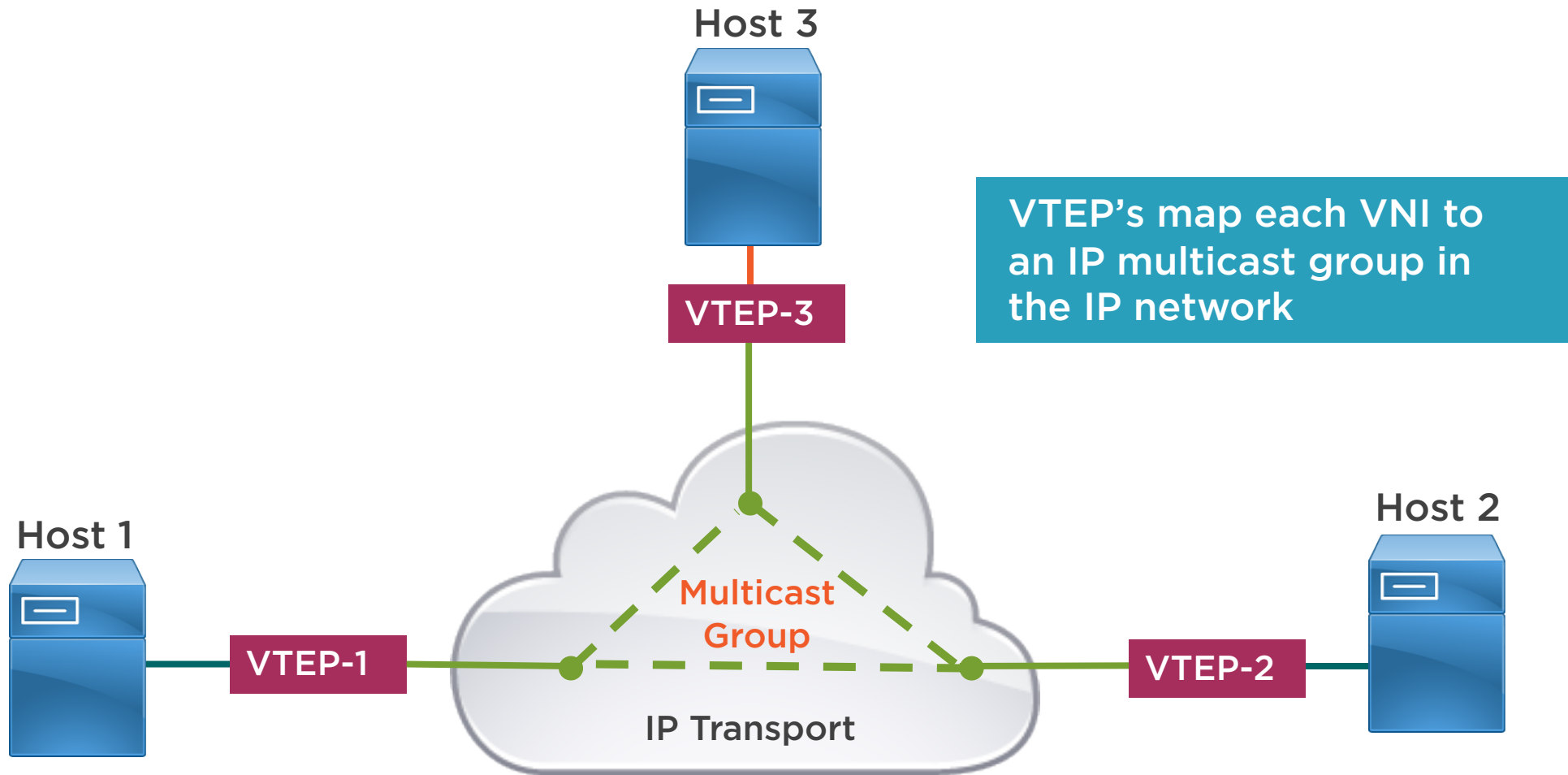
Forwards BUM traffic using a multicast forwarding tree which can be static ingress replication or MP-BGP EVPN ingress replication



With static ingress replication, remote peers are statically configured. Multi-destination packets are unicast and sent to remote peers



# VXLAN Using BIDIR-PIM



# VXLAN Using BIDIR-PIM



Using multicast limits Layer 2 flooding to devices in the same VXLAN segment



Devices send unicast data from one VTEP to another using the unicast address of a VTEP



No routing protocol – multicast uses layer 2 data plane flooding and learning





# VXLAN Using BIDIR-PIM



When VXLAN uses BIDIR-PIM, VTEPs map each VNI to an IP multicast group IP network



Each VTEP is configured independently, and then it will use IGMP to join this multicast group



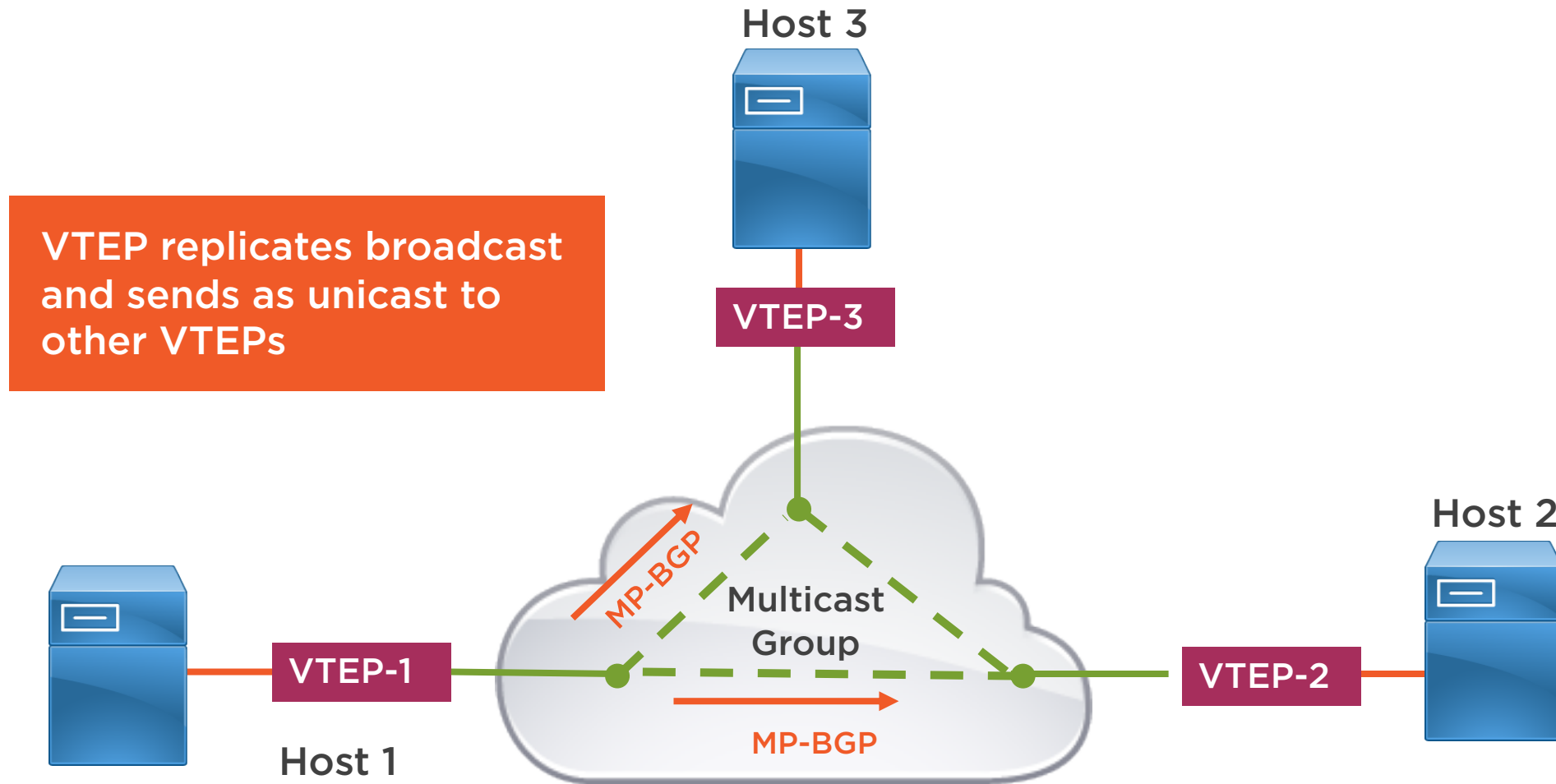
VXLAN uses the multicast group to transmit the BUM traffic through the IP network



Devices send unicast data from one VTEP to another using the unicast address of a VTEP



# VXLAN Control Plane Using MP-BGP EVPN



# VXLAN Gateways



Data centers contain devices that are not capable of supporting VXLAN, such as legacy hypervisors and physical servers



A VXLAN layer 2 gateway is a VTEP device that combines a VXLAN segment and a VLAN segment into one common layer 2 domain



VXLAN layer 3 gateway, a VXLAN router, routes between different VXLAN segments. This translates frames from one VNI to another



In MP-BGP EVPN, any VTEP can act as a gateway for hosts in its IP subnet by supporting the same virtual gateway IP and MAC address



# VXLAN Configuration

---



```
N7K-1(config)# feature pim
```

```
N7K-1(config)# ip pim rp-address 192.168.1.1 group-list 224.0.0.0/4
```

## PIM Rendezvous Point

Use the RP of interface of Loopback 0



# Enable PIM Sparse Mode

```
N7K-1(config)# interface ethernet 3/1
```

```
N7K-1(config-if)# ip pim sparse-mode
```

```
N7K-1(config-if)# interface loopback 0
```

```
N7K-1(config-if)# ip pim sparse-mode
```

```
!
```

```
N7K-1# show ip pim neighbor
```



```
N7K-1(config)# feature nv overlay
```

```
N7K-1(config)# feature vni
```

## Configure VXLAN

NVE is what Cisco Nexus 7000 implementation of VXLAN calls a VTEP interface

VNI is a VXLAN identifier



# Bridge Domain Interface

```
N7K-1(config)# system bridge-domain 21
```

```
N7K-1(config)# vrf context Tenant
```

```
N7K-1(config-vrf)# interface bdi21
```

```
N7K-1(config-if)# no shutdown
```

```
N7K-1(config-if)# vrf member Tenant
```

```
Warning: Deleted all L3 config on interface Bdi21
```

```
N7K-1(config-if)# ip address 192.168.21.2/24
```





```
N7K-1(config)# vni 10201
```

```
N7K-1(config-if-nve)# bridge-domain 21
```

```
N7K-1(config-bdomain)# member vni 10201
```

## Configure VNI

We can't use matching bridge domain and VNI numbers because their ranges are excluding

VNI range is from 4096 to 16777215 and the bridge domain range is from 2 to 4094



```
N7K-1(config)# interface nve 1
```

```
N7K-1(config-if-nve)# no shutdown
```

```
N7K-1(config-if-nve)# source-interface loopback0
```

```
N7K-1(config-if-nve)# member vni 10201 mcast-group 225.1.2.1
```

## Create VNI Interface

### Create the NVE interface

On Nexus 7000's, you can use only a loopback interface as the source interface for an NVE



```
N7K-1(config)# encapsulation profile vni DC_VXLAN
```

```
N7K-1(config-vni-encap-prof)# dot1q 201 vni 10201
```

## Application Profile

The encapsulation instance is also called VSI (VN-Segment Service Instance)



```
(N7K-1(config)# interface ethernet 2/1  
N7K-1(config-if)# service instance 1 vni  
N7K-1(config-if-srv-vni)# encapsulation profile DC_VXLAN default  
N7K-1(config-if-srv-vni)# no shutdown
```

## Apply the Application Profile

**Do not forget to enable the service instance; otherwise, it will not work**



# VNI Verification

```
N7K-1# show bridge-domain
```

```
Bridge-domain 21 (2 ports in all)
```

```
Name:: Bridge-Domain21
```

```
Administrative State: UP
```

```
Operational State: UP
```

```
    vni10201
```

```
    VSI-Eth2/1
```

```
    nve1
```



# Summary



Overlay protocols provide flexibility, scalability, and manageability

Examine overlay protocols used to connect geographically disparate L2 data centers over L3 networks

- Cisco OTV
- VXLAN

