VxLAN技术 探讨和方案选择

v1.0版本

马旻

电子邮件：minma@cisco.com

行业解决方案部 顾问工程师

思科系统（中国）网络技术有限公司

2014年3月

前言

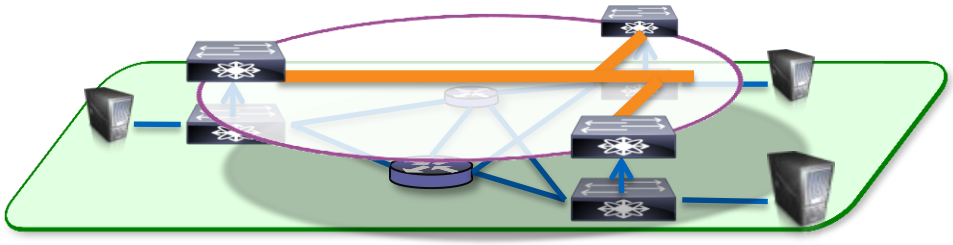
- VXLAN 的起源和功能
- VXLAN 的定位和使用场景和应用范围
- 现阶段VXLAN 实现情况
- 澄清VXLAN 相关的一些观点

议程

- Overlays 叠加网络介绍
- VXLAN概念介绍
- VXLAN 网络设计
- 现阶段VXLAN部署的问题
- VXLAN发展的下一阶段
- SDN与VXLAN


Overlays

为什么需要 Overlays?



灵活的叠加虚拟网络

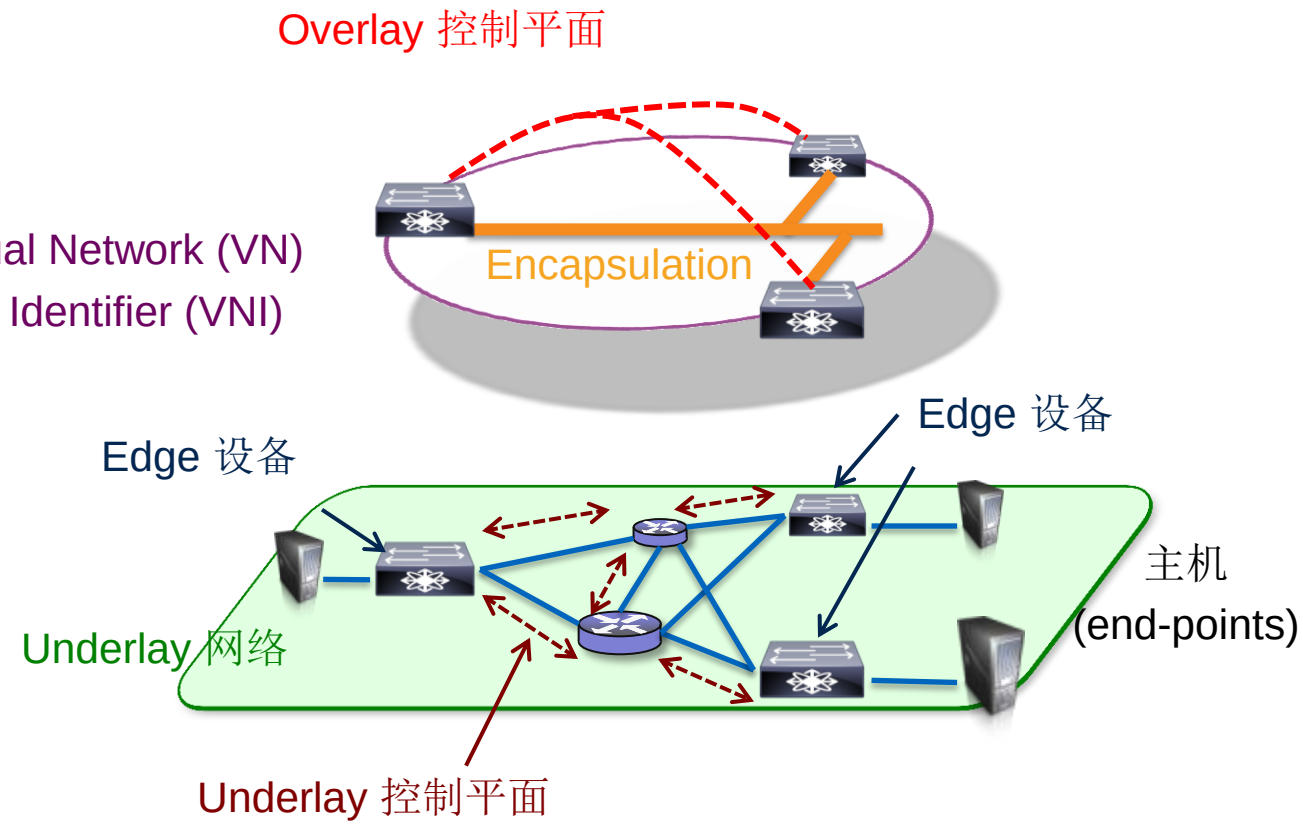
- 移动性 – 跟踪边缘设备连接的主机
- 伸缩性 – 减少核心网络状态，网络边缘分布式和分区处理
- 多租户 – 共享网络资源
- 灵活性/可编程，减少维护端点数量

按需在底层承载网络之上快速部署虚拟业务网络

Overlay 术语

Service = Virtual Network (VN)

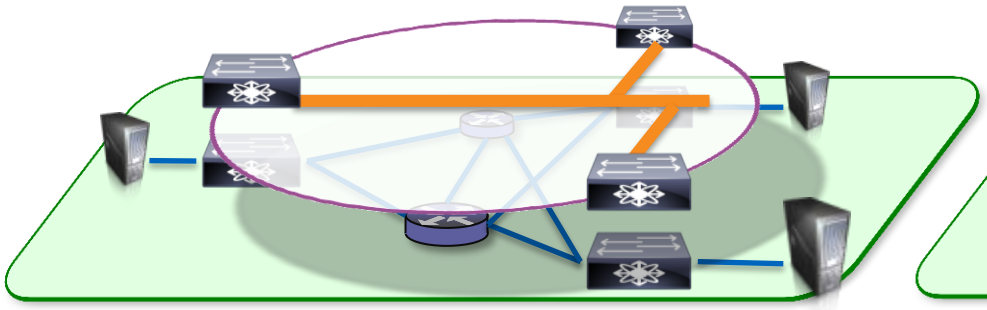
Identifier = VN Identifier (VNI)

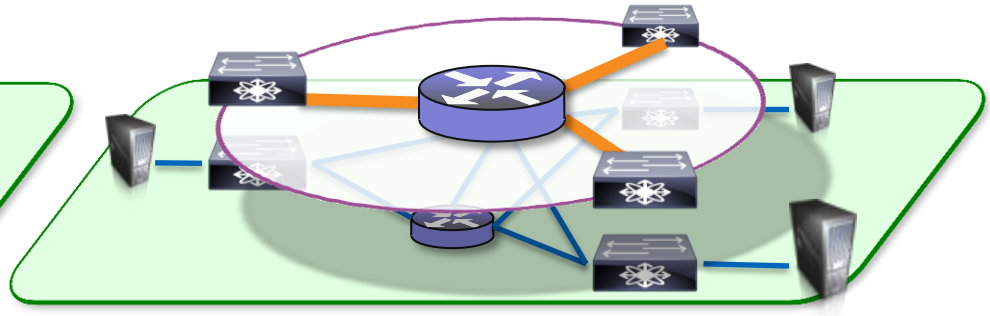


Overlay 属性

服务	边缘设备	信令
Layer 2 服务 Layer 3 服务	主机式 Overlays 网络式 Overlays 混合式 Overlays	数据平面学习 控制平面学习

Overlay 服务类型





Layer 2 Overlays

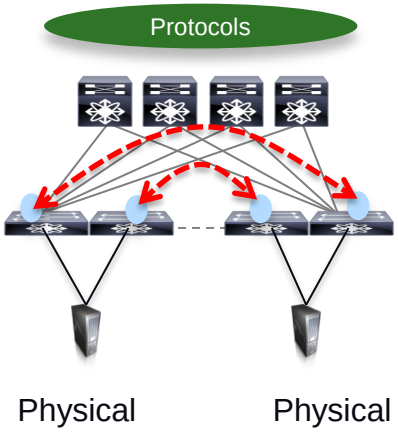
- Emulate a LAN segment
- Transport Ethernet Frames (IP and non-IP)
- Single subnet mobility (L2 domain)
- Exposure to open L2 flooding
- Useful in emulating physical topologies

Layer 3 Overlays

- Abstract IP based connectivity
- Transport IP Packets
- Full mobility regardless of subnets
- Contain network related failures (floods)
- Useful in abstracting connectivity and policy

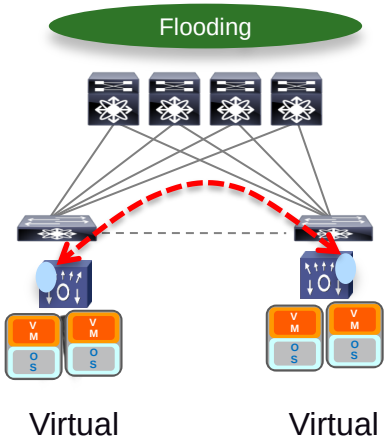
Overlay边缘设备类型

网络式 Overlays



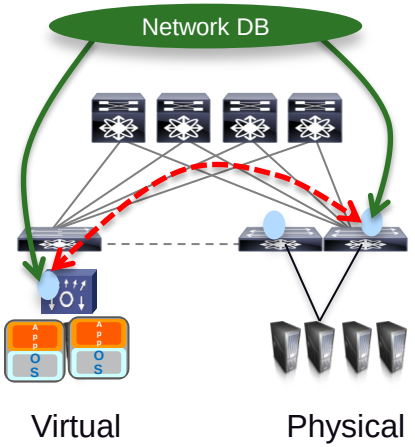
- Router/switch end-points
- Protocols for resiliency/loops
- Traditional VPNs
- **OTV, VPLS, LISP, FP**

主机式 Overlays



- Virtual end-points only
- Single admin domain
- **VXLAN, NVGRE, STT**

混合式 Overlays

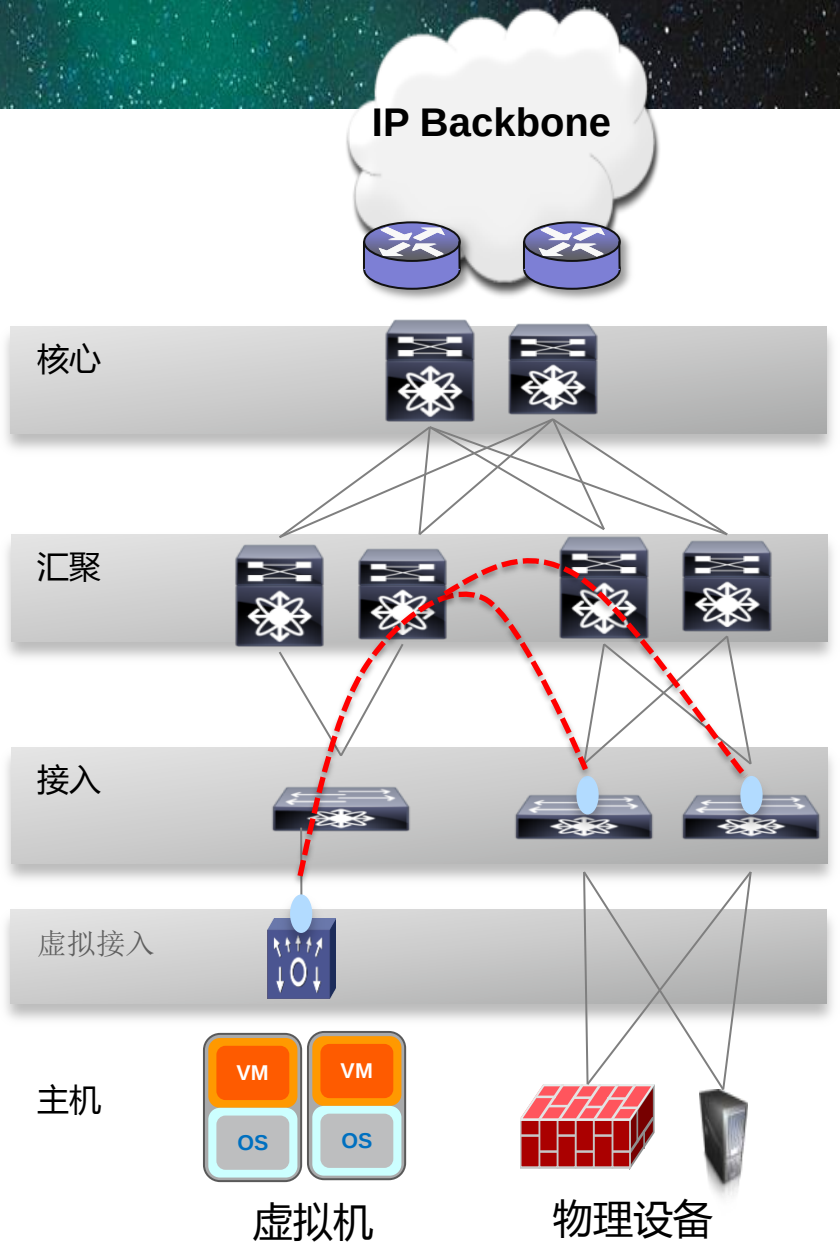


- Physical and Virtual
- Resiliency + Scale
- x-organizations/federation
- Open Standards

Tunnel End-points

混合式 Overlays

- **VMs** 连接到虚拟交换机
  - **主机式 overlays** 发起在虚拟交换机
  - 基于虚拟化的弹性: **单挂方式**
- **Physical hosts** 连接到物理接入交换机
  - **Network overlays** 发起在物理接入层
  - 网络的弹性: **多宿主方式**
- 混合式 Overlay允许联合物理和虚拟化的资源



VXLAN概念介绍

VXLAN 综述

VXLAN 解决的问题:

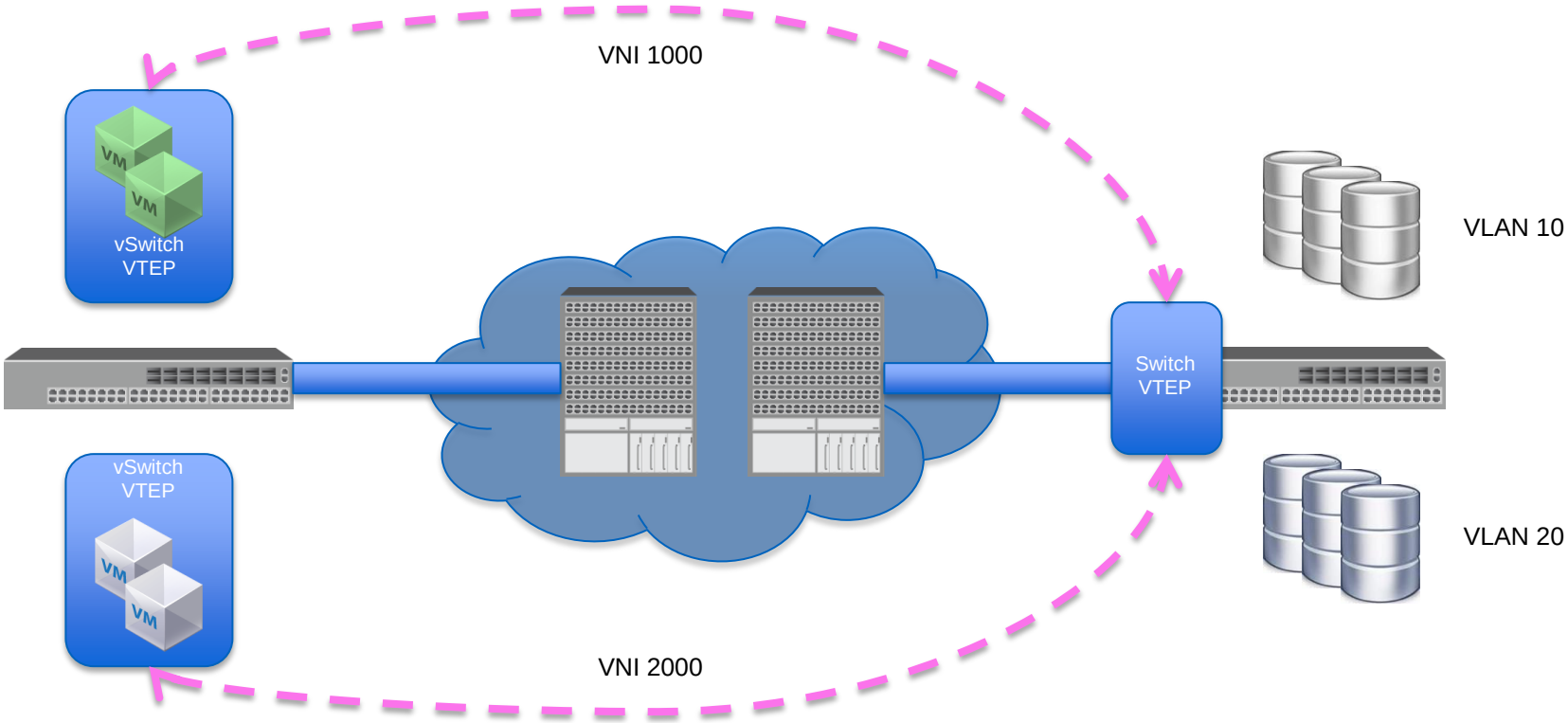
- VLAN 扩展性问题 (4K) – VXLAN 扩展二层分段 识别ID 字段到24位，理论上可以在同一个网络上 标识 1600万个唯一的二层网络分段
- VM 移动被限制在本地VLAN – VXLAN 封装二层数据帧在 IP-UDP 报头，允许二层网络连接跨越三层路由网络。

VXLAN 技术综述:

- MAC-in-UDP 封装
- 利用底层传输网络的组播能力仿真二层网络中的BUM帧
- 利用 ECMP 实现底层网络之上的路径优化

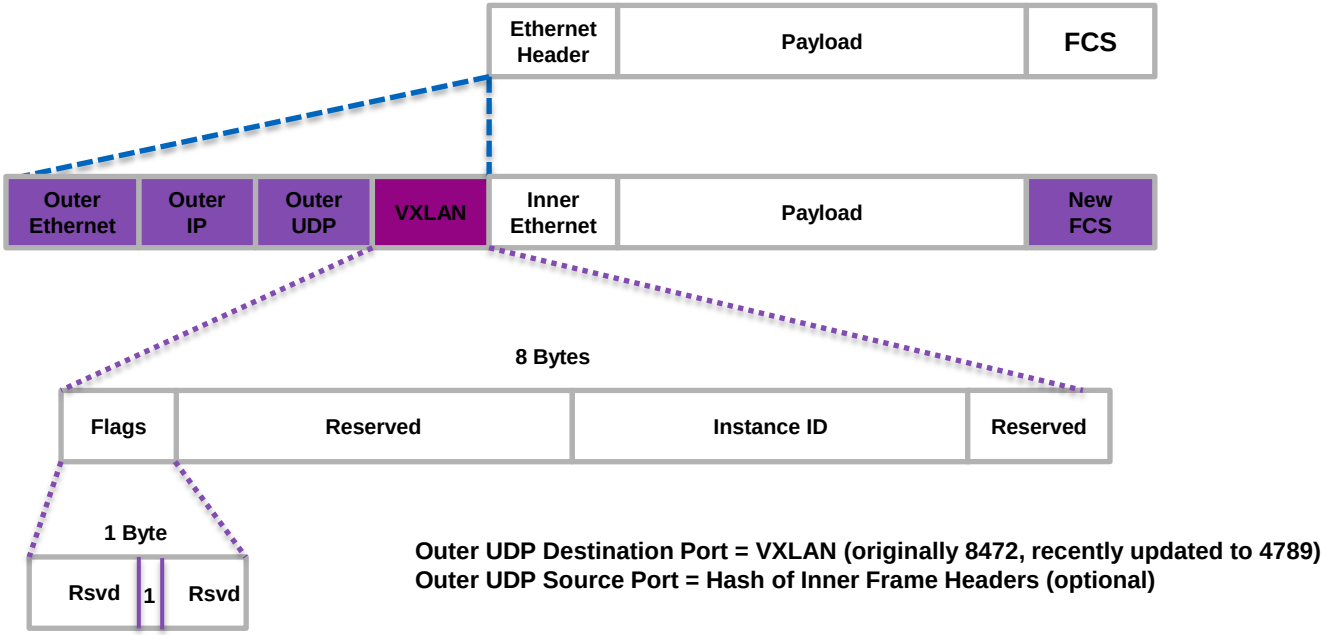
VXLAN 综述

VXLAN can be implemented on both Hypervisor-based Virtual Switches to allow for scalable VM deployments, as well as on Physical switches, which provides the ability to bridge VXLAN segments back into VLAN segments. In these cases, the Physical Switch instantiates a VTEP, and function as a VXLAN Gateway…

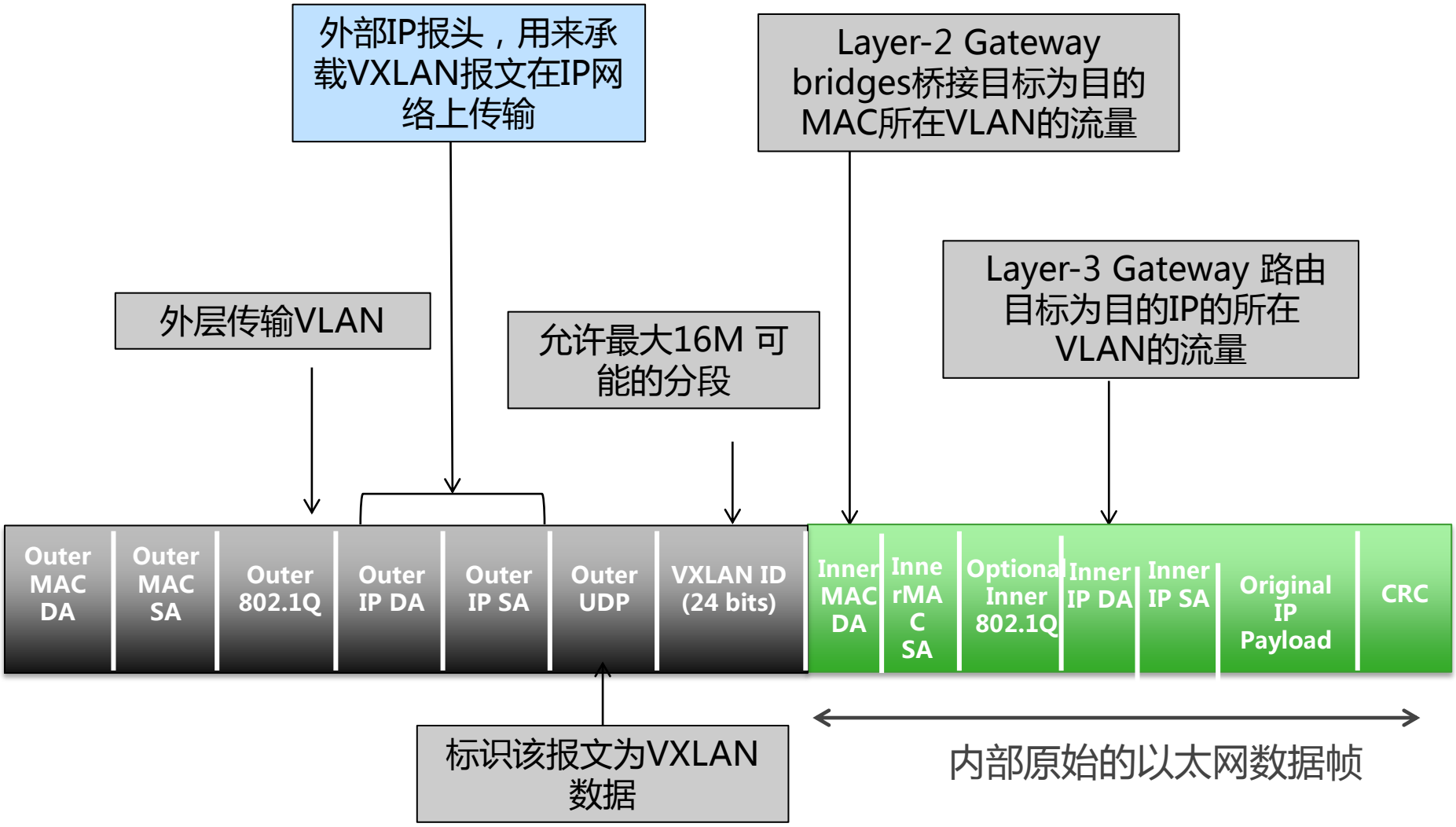


Virtual eXtensible LAN (VXLAN)

- Virtual eXtensible LAN (VXLAN) 是建立 Layer 3 network之上的一个 Layer 2 overlay 处理方法
- 一个 24-bit VXLAN Segment ID 或者 VXLAN Network Identifier (VNI) 包含在数据帧头提供达到16M VXLAN 分段，相比VLAN只能提供 4K 个分段。每个分段代表一个唯一的二层广播域，子网或租户地址

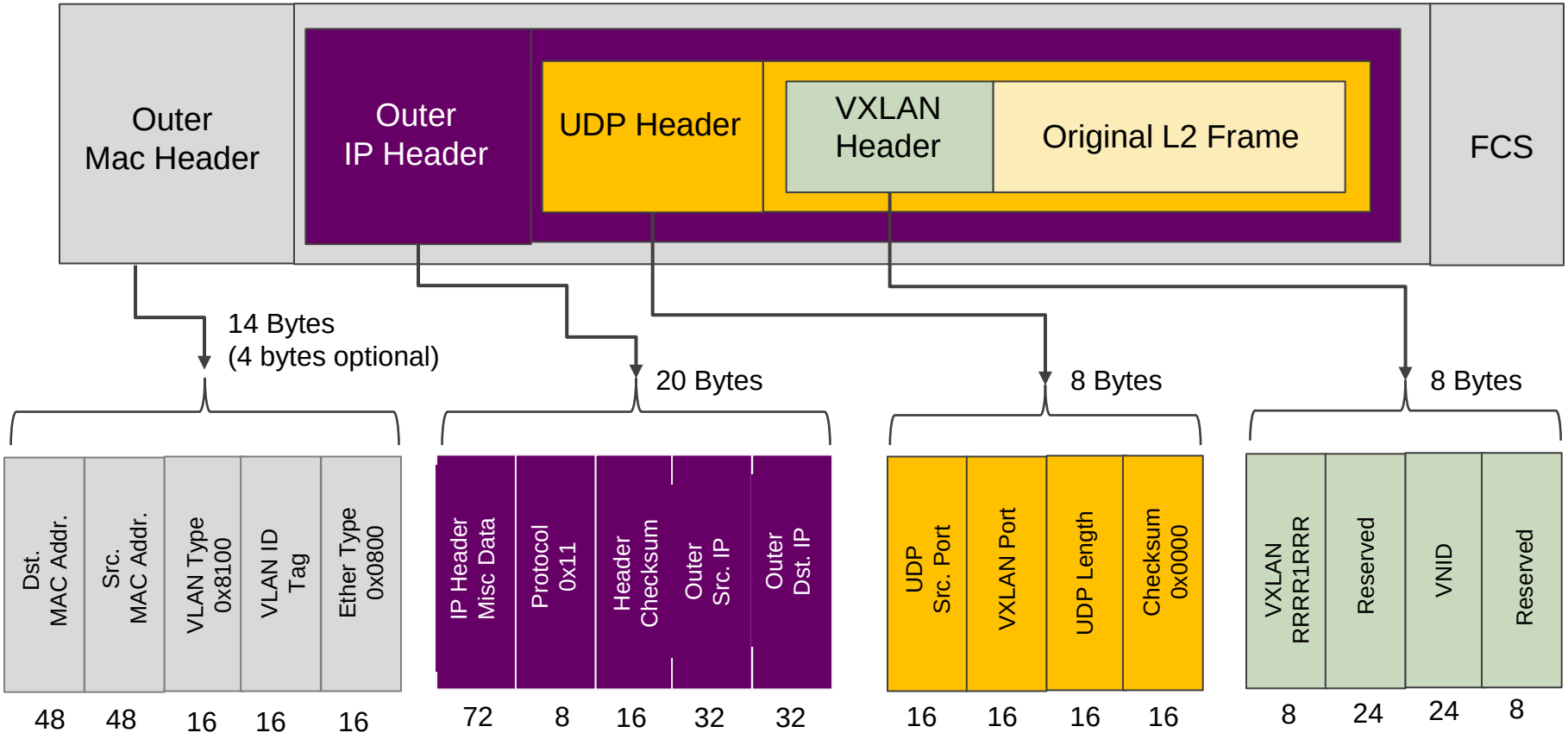


VXLAN 帧格式



VXLAN 帧格式 (Cont.)

增加50字节的开销，注意调整设备接口下的MTU值

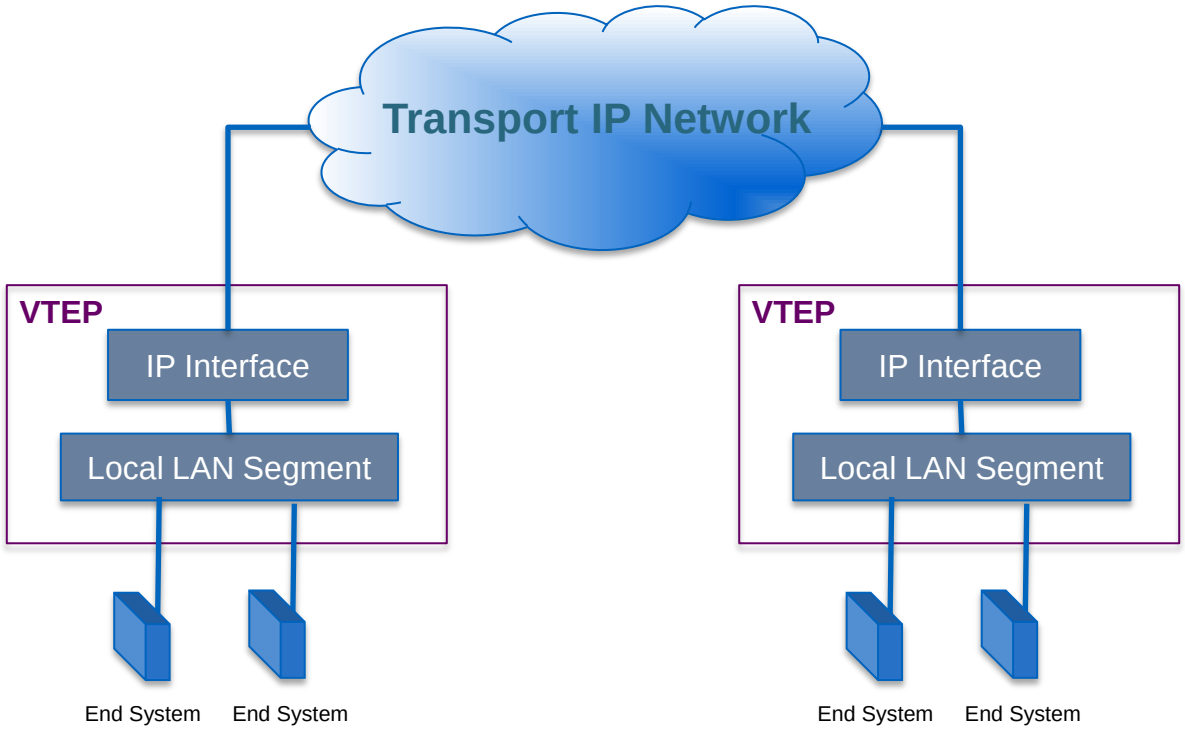


Outer UDP Destination Port = VXLAN (originally 8472, recently updated to 4789)
Outer UDP Source Port = Hash of Inner Frame Headers (optional)

VXLAN VTEP

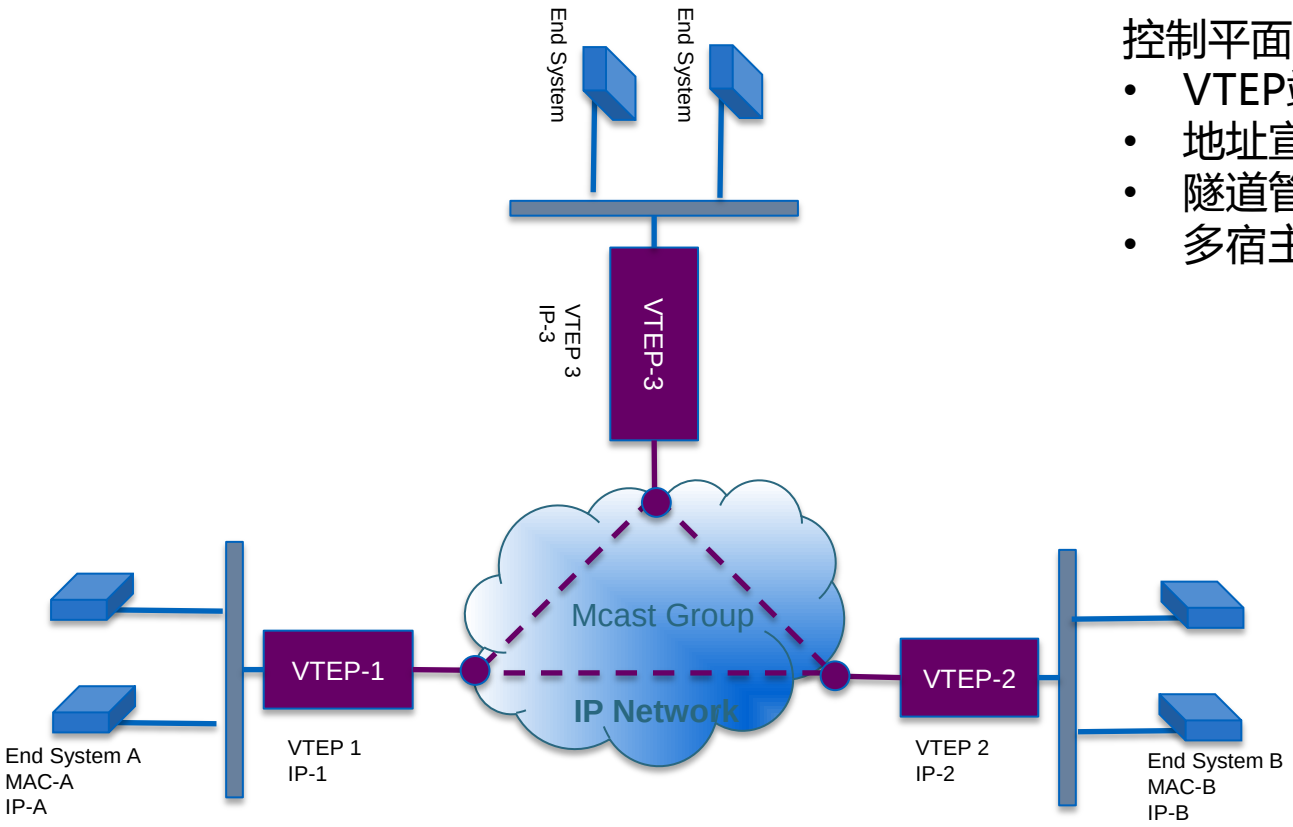
VXLAN 在 VTEPs (Virtual Tunnel End Point) 终结隧道服务

每个VTEP提供两个接口：
一个负责本地主机桥接功能， 另外一个连接核心网络提供VxLAN封装和解封装



多目的地流量的处理

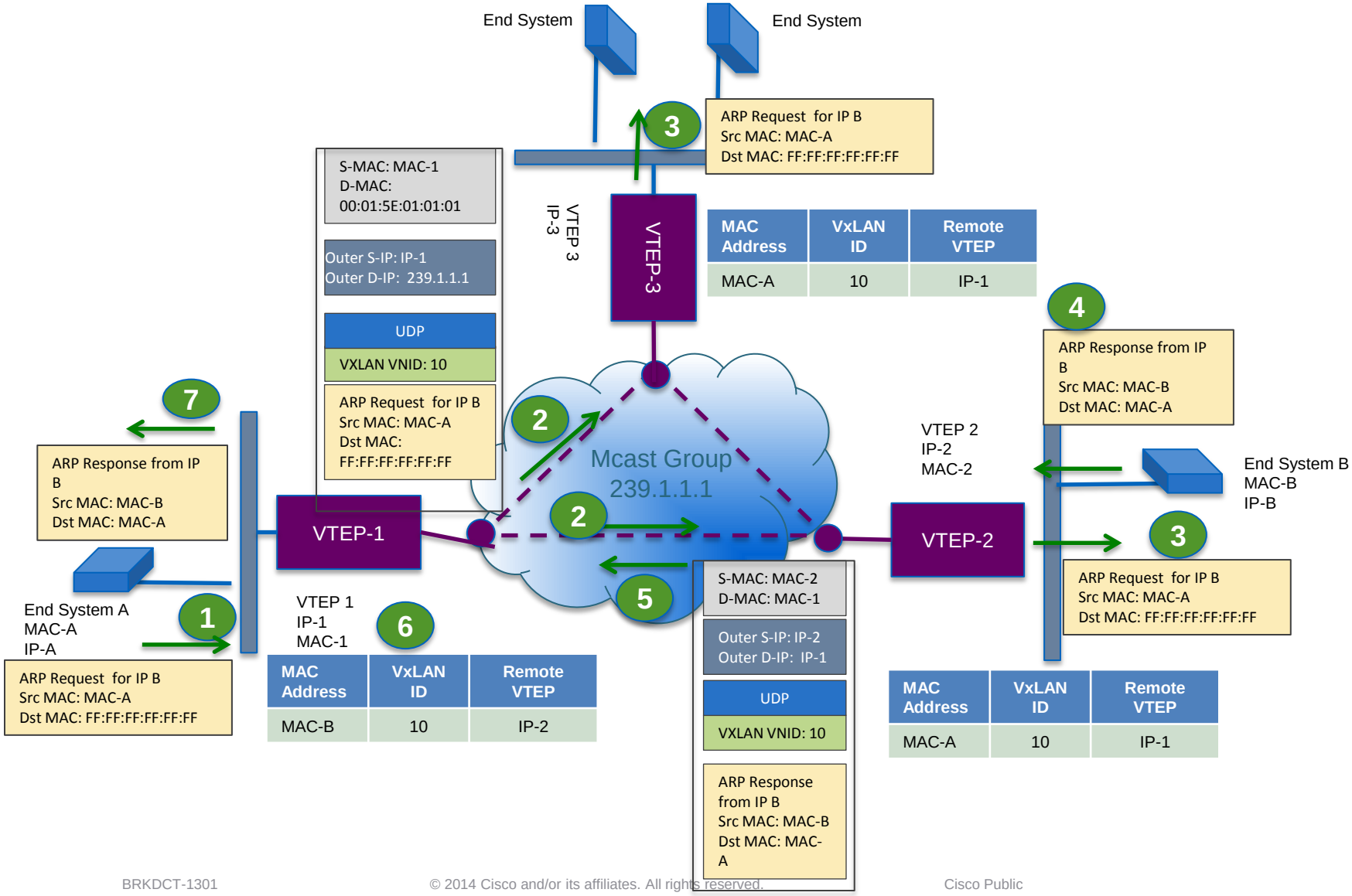
由于目前控制信令协议尚未实现，多目的流量 (Broadcast, Multicast, Unknown Unicast) 利用底层IP网络提供的组播服务实现，即借助数据转发平面实现。



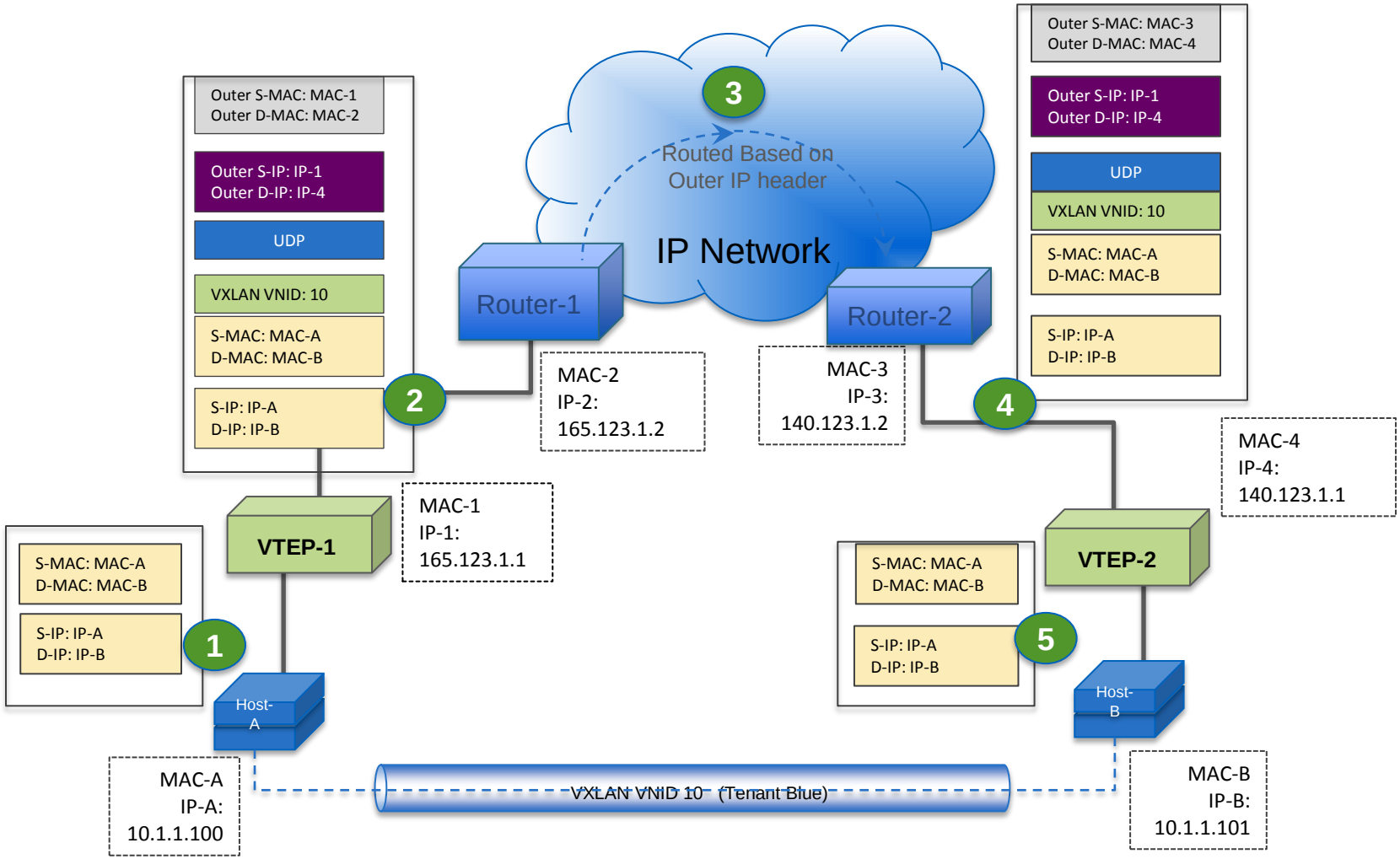
控制平面的功能：

- VTEP站点服务发现
- 地址宣告和映射
- 隧道管理
- 多宿主连接

VTEP 发现和地址学习过程



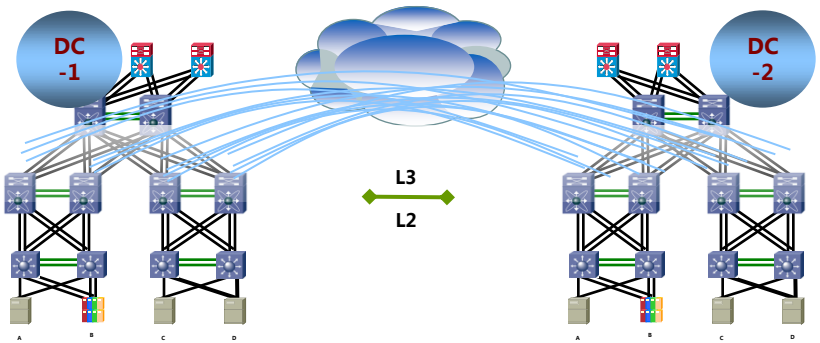
VXLAN单播数据流转发过程



L2 Overlays 中的泛洪

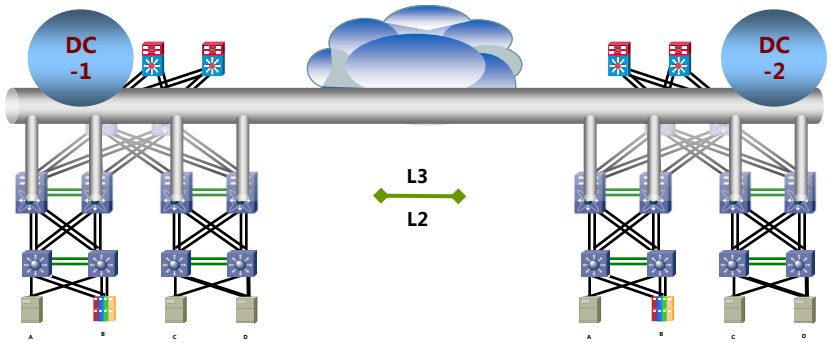
采用控制平面信令消除泛洪发生

数据平面学习方式



- Pre-set flood facility
- MAC learning based on flooding
- Flood L2 protocols and unknown unicast
  - ➔ Failure propagation
- Fail Open
- Suitable for small domains (failure scope)

控制平面学习方式



- No predetermined flood tree
- MAC learning by control protocol
  - ➔ Contain Failures and L2 protocols
  - ➔ Rich information
- Fail Closed
- Better suited for broad scope

在L2 Overlays中泛洪

MAC 路由

L2 Overlay 演进过程

Inter-DC (DCI)

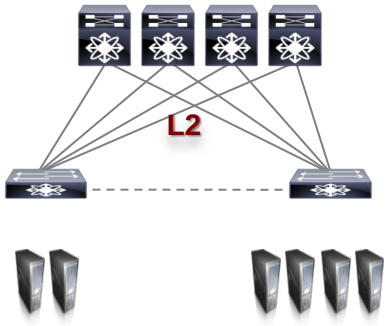
VPLS ➔ A-VPLS
Flood and Learn

OTV
MAC routing (IS-IS)

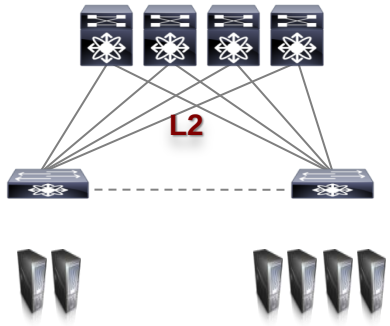
EVPN / L2-LISP
MAC routing (BGP/LISP)

Backbone Network

Intra-DC
(Fabric)



FP （TRILL）
Flood and Learn



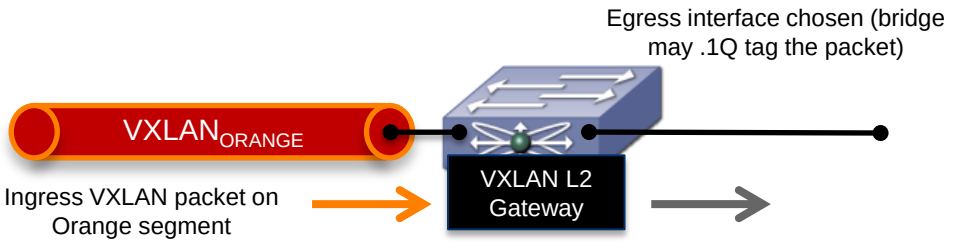
Enhanced Forwarding (hybrid)
Control Plane

Host Overlays

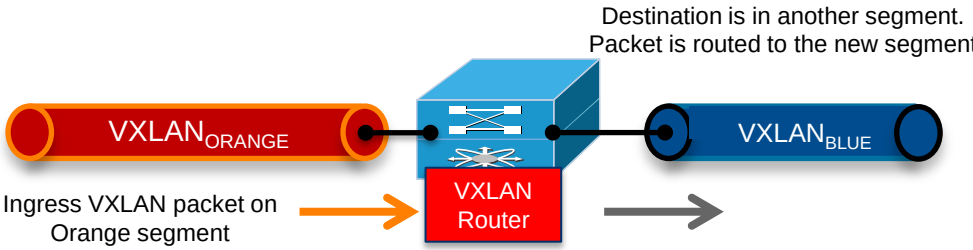
VXLAN
Flood and Learn

VXLAN 网关设备处理方式

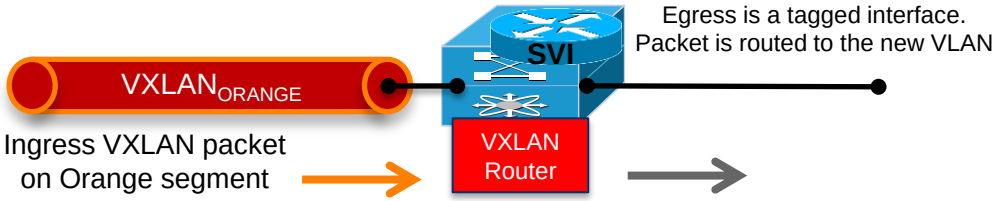
- VXLAN to VLAN （/ VXLAN）Bridging (L2 Gateway)



- VXLAN-to-VXLAN Routing (L3 Gateway)

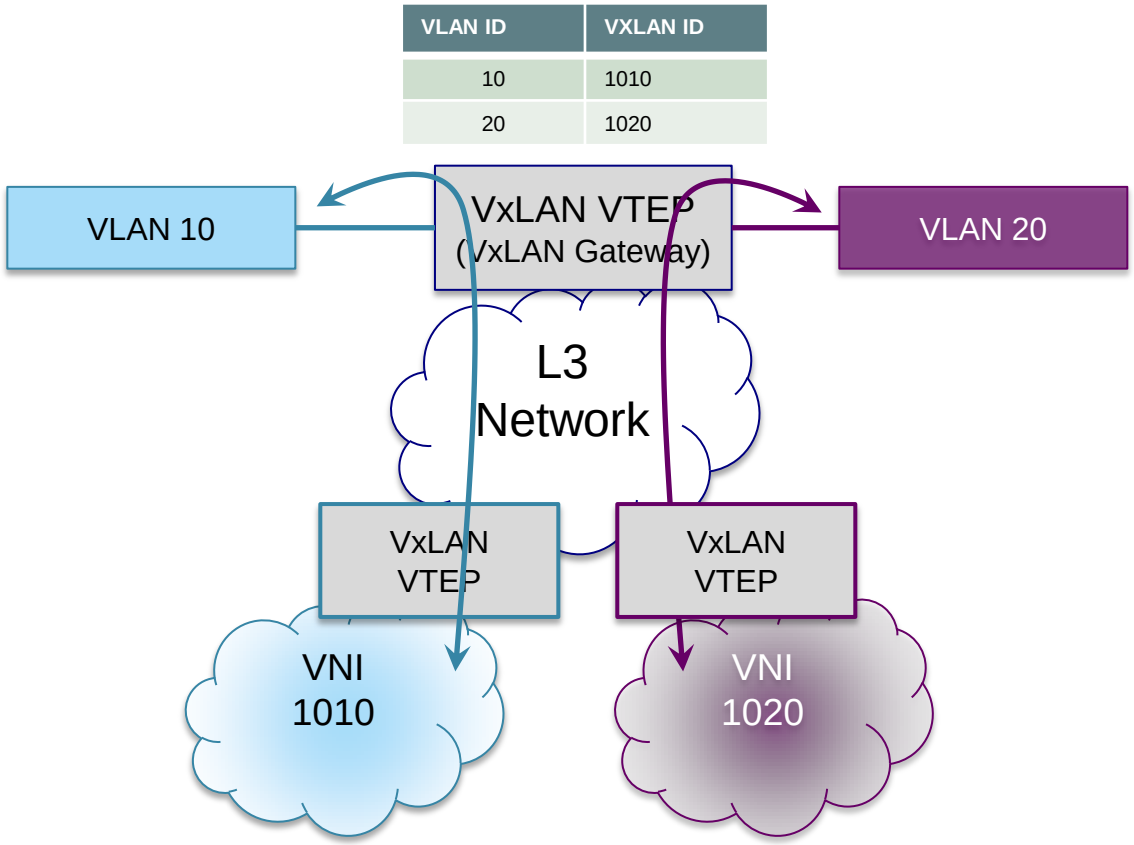


- VXLAN-to-VLAN Routing (L3 Gateway)



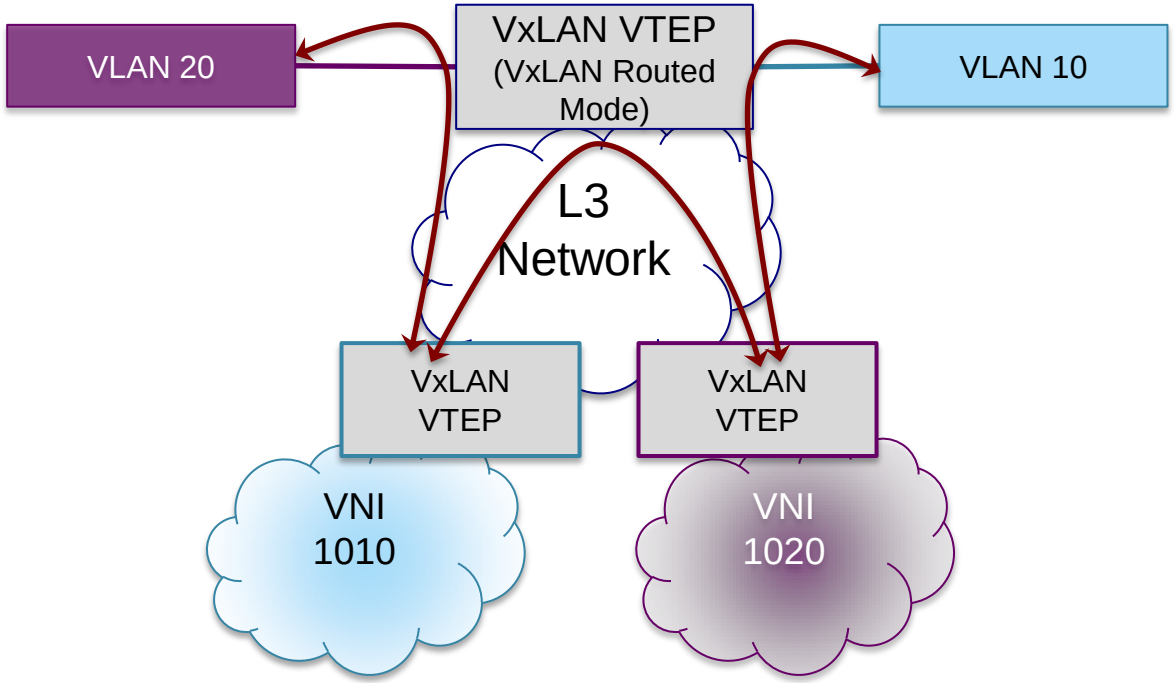
VxLAN Gateway

VXLAN gateway bridges traffic between VXLAN segment and another physical / logical layer 2 domain (such as a VLAN)…



Routing VxLAN

VxLAN routed mode ‘routes’ traffic between VxLAN segments and between VxLAN another physical / logical layer 2 domain (such as a VLAN)…



Nexus 9000 系列
VXLAN 支持情况

VXLAN is supported across the Nexus 9000 series platforms. The VXLAN Gateway functionality is supported across all form factors and line cards. Integrated routing functionality is only supported on ACI-enabled Modules…

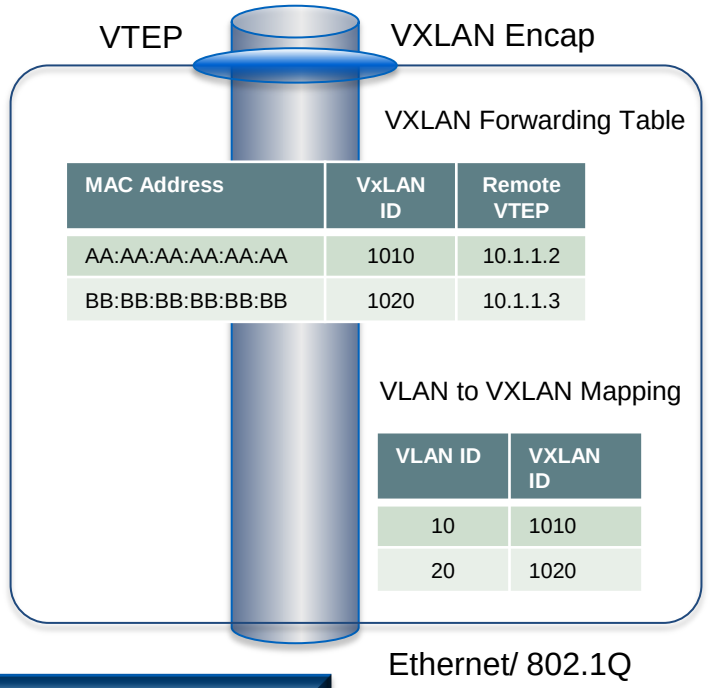
Nexus 9300 Series

Nexus 9500 Series

VxLAN Gateway – Mapping

The Nexus 9000 series supports VxLAN Gateway function, allowing VLANs to be bridged/mapped to VxLAN Segments and vice versa…

```
feature nv overlay
feature vn-segment-vlan-based

interface et4/13
  switchport
  switchport access vlan 10
  no shut
interface nve1
  no shutdown
  source-interface loopback0
  overlay-encapsulation vxlan
  member vni 1010 mcast-group 230.1.1.1

vlan 10
  vn-segment 1010
```




```
switch# show nve vni
Interface          VNI        Multicast-group   VNI State
---------------    --------   ---------------   ---------
nve1               1010        230.1.1.1            up
switch# show nve peers
Interface           Peer-IP              VNI         Up Time
-------------       ----------------     --------    -------
nve1                10.1.1.2             1010       00:52:24
switch#
```

思科产品支持VxLAN的情况

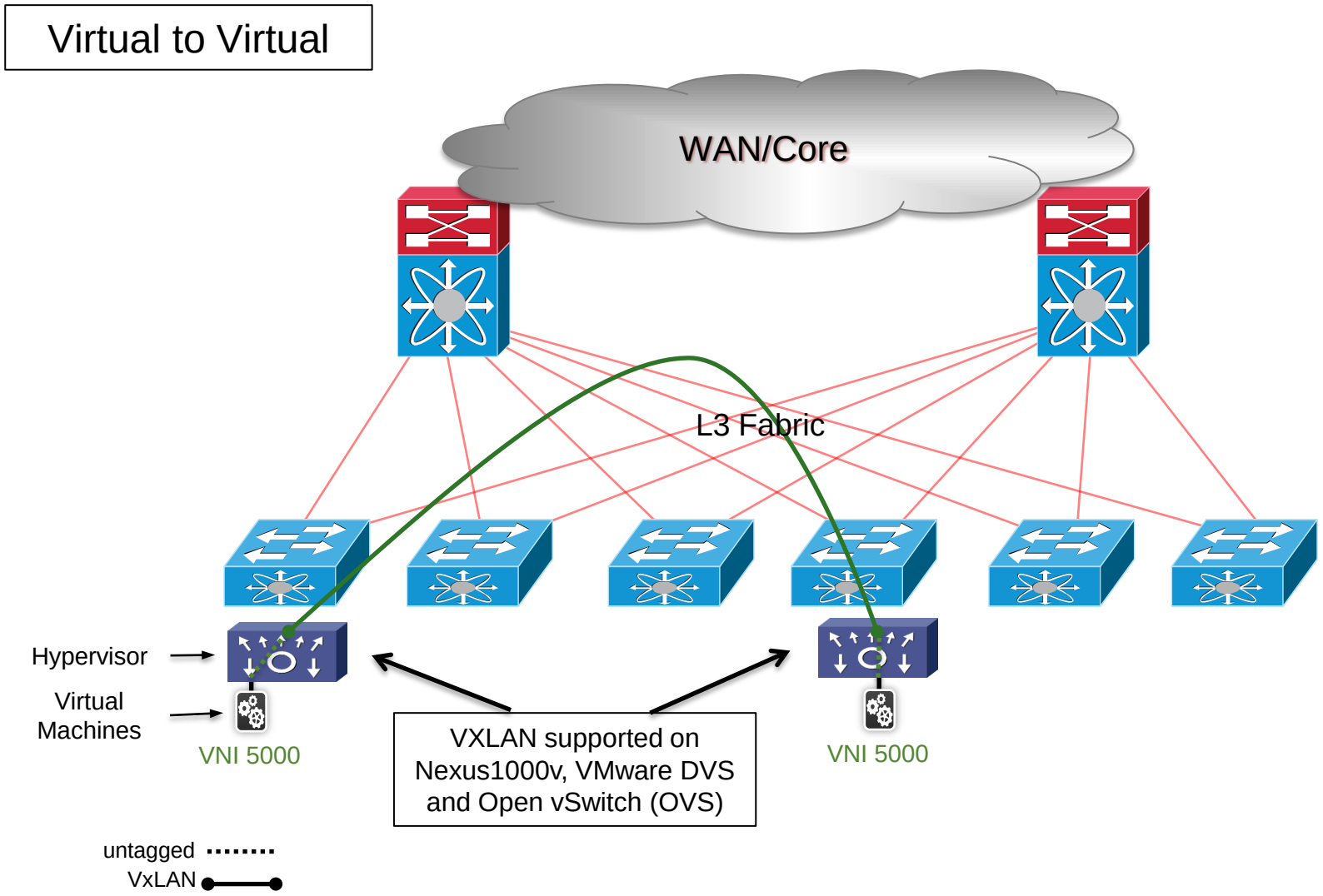
- **Nexus 9500/9300**
- **Nexus 7000 (F3)**
- **Nexus 6000**
- **Nexus 5600**
- Nexus 3100
- Nexus 1000v

- **CSR 1Kv / ASAv**

- **ASR 9000**
- **ASR 1000**

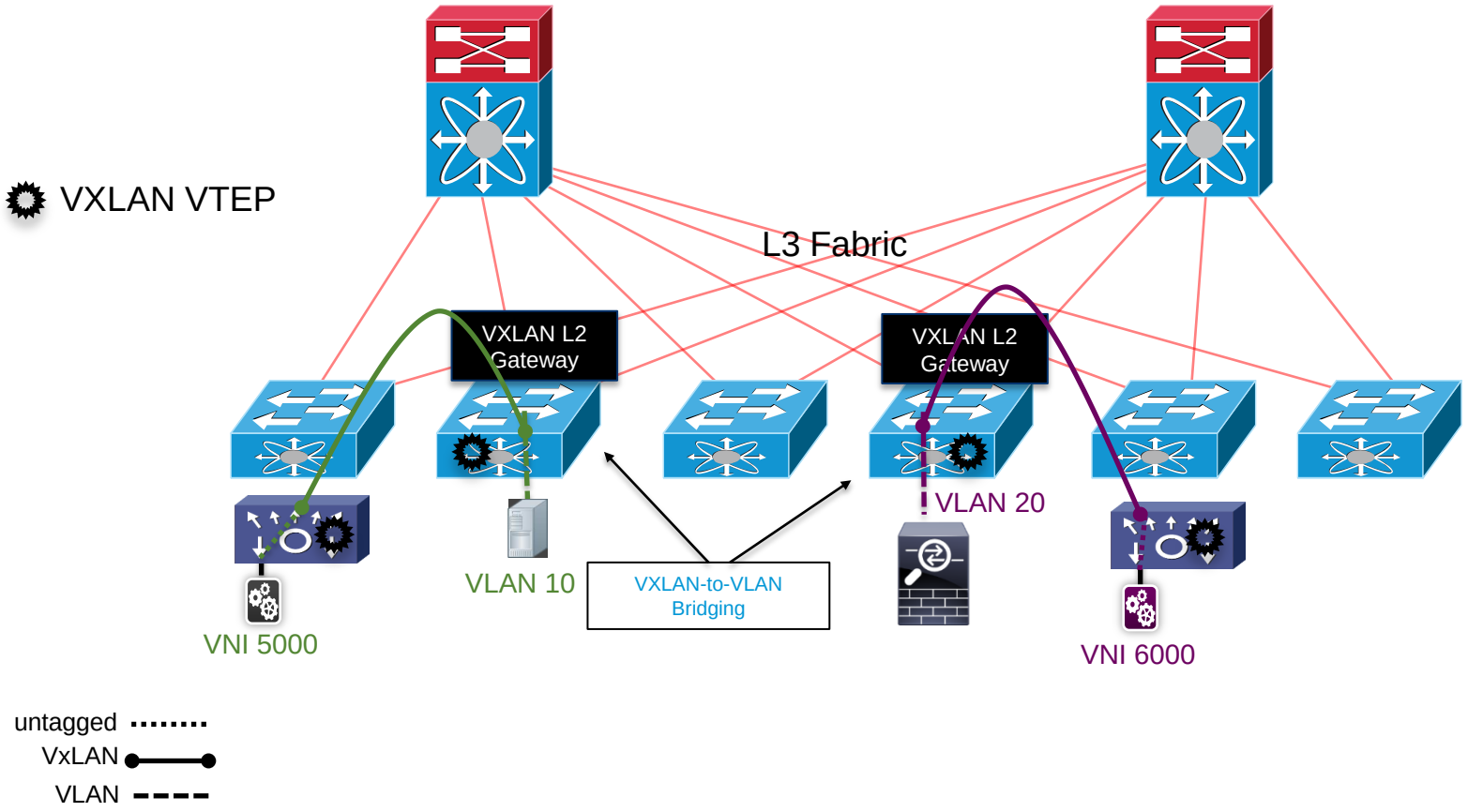
VXLAN 网络设计

“Virtual” VXLAN Layer-2 Gateway Purely an Host Overlay Solution

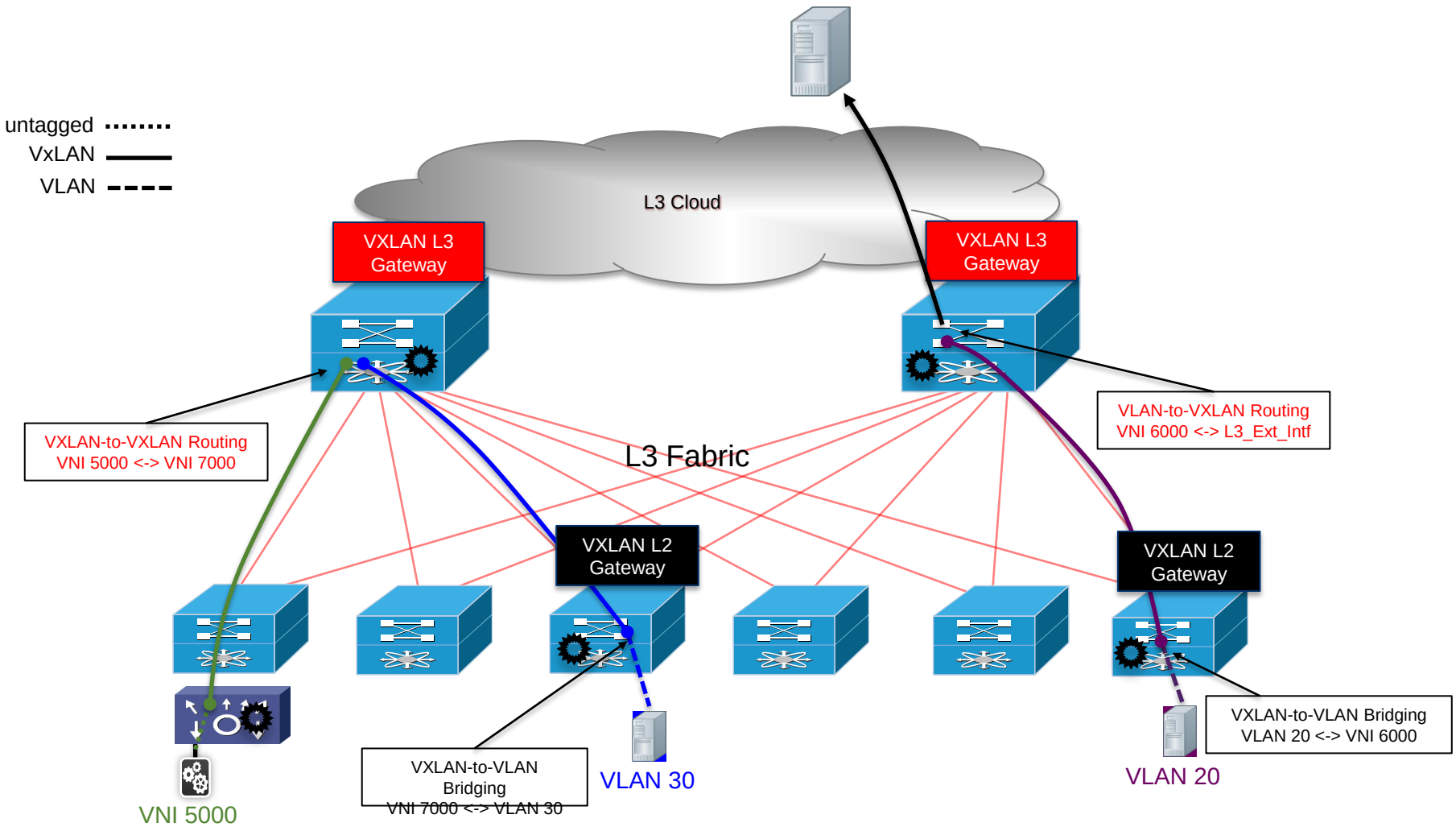


HW VXLAN L2 Gateway
Intra-Subnet Communication

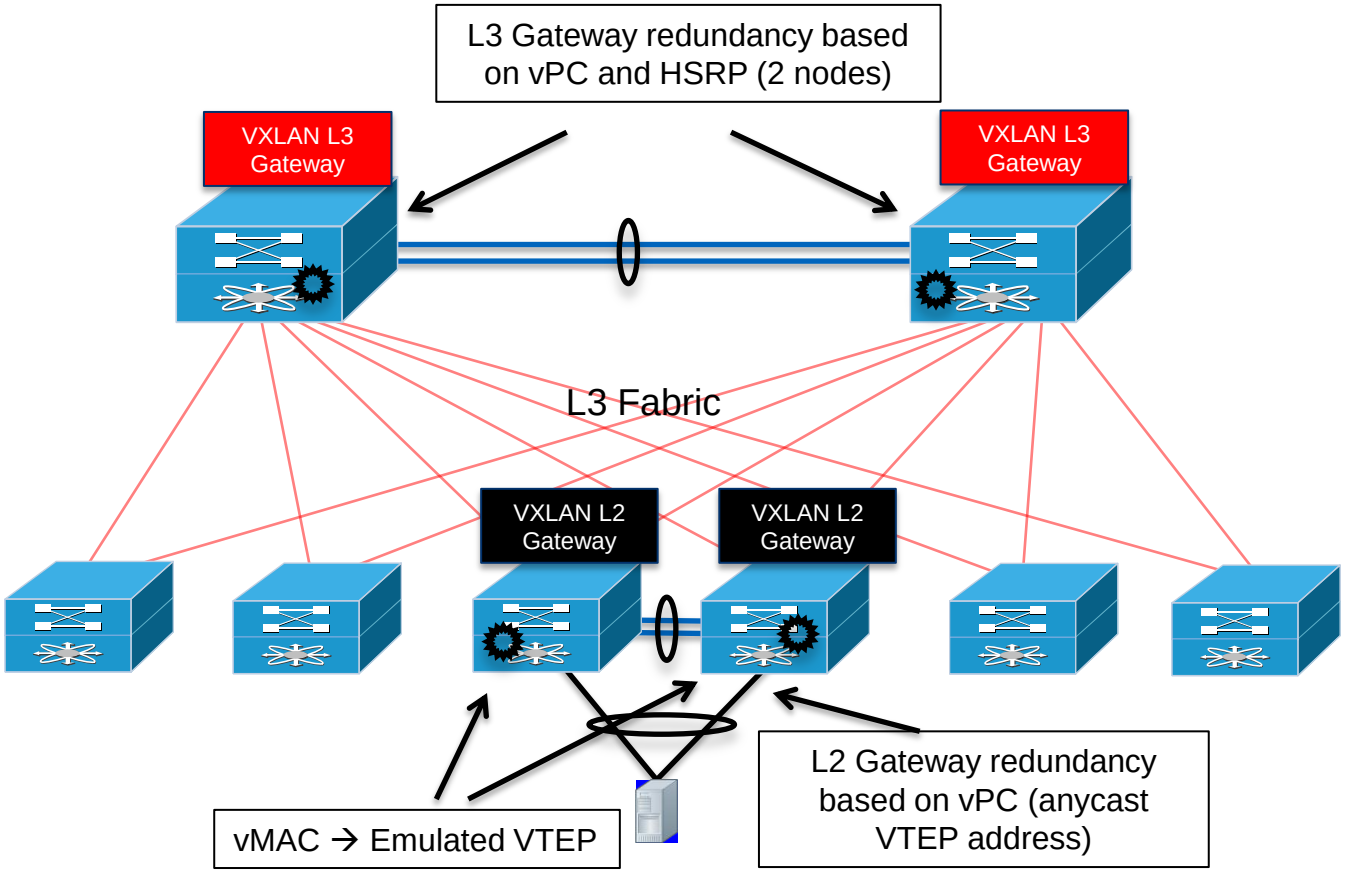
Virtual to Physical



HW VXLAN Routing
Inter-Subnets Communication

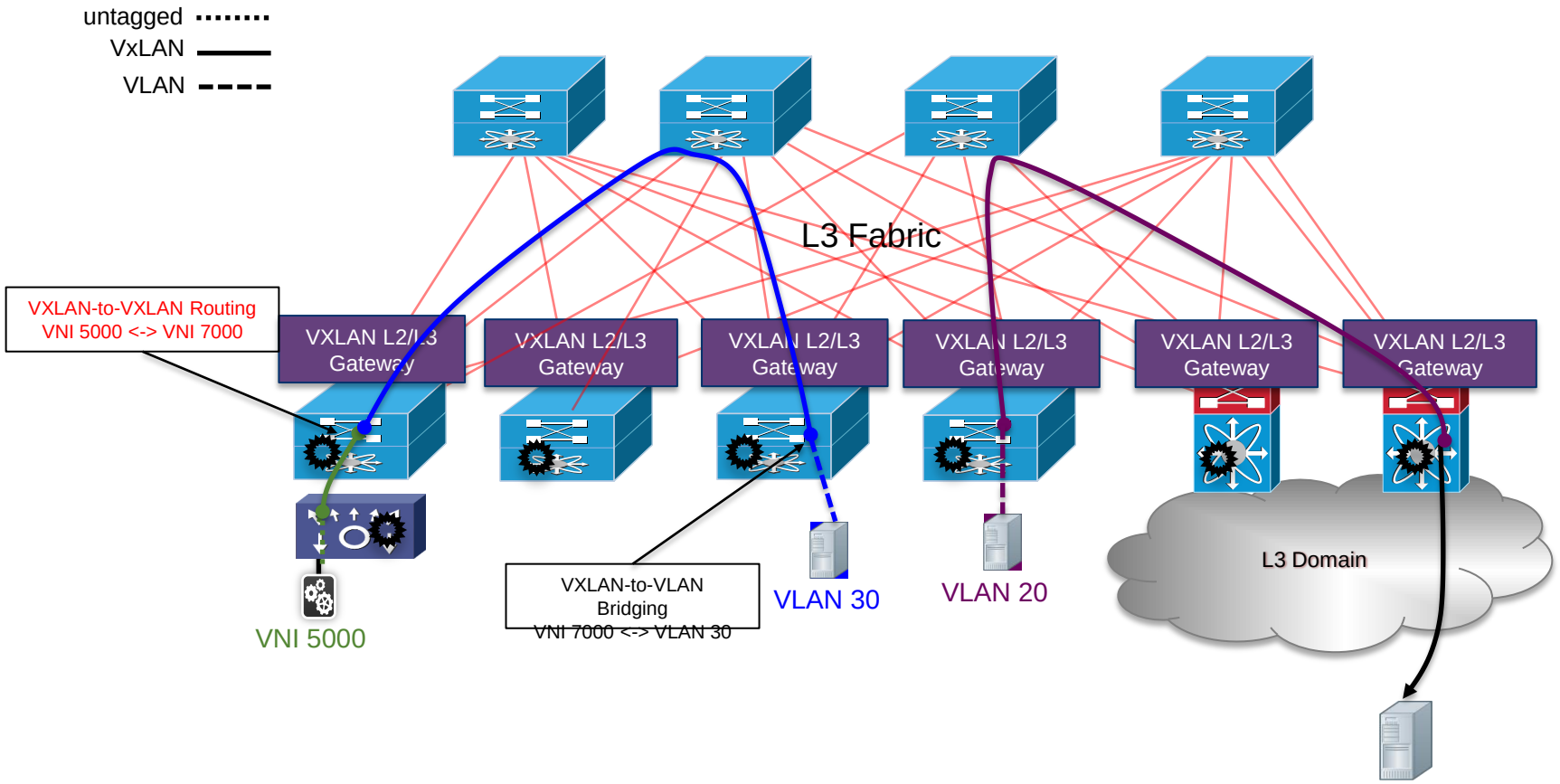


HW VXLAN Routing
VTEP Redundancy

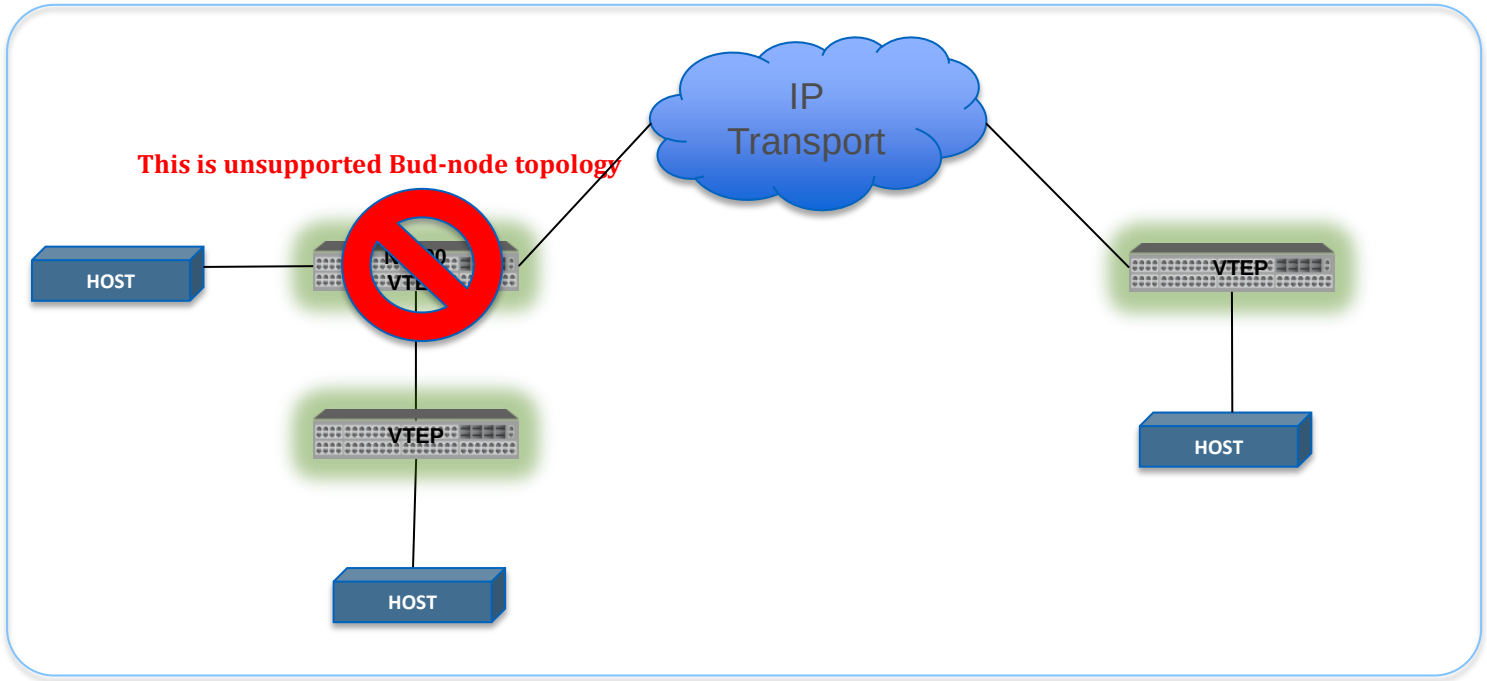


HW VXLAN Routing
Distributed Gateways

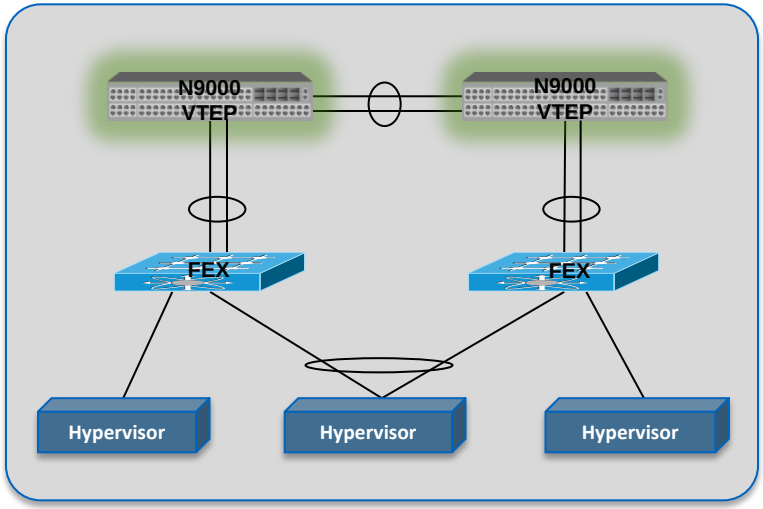
Traffic is forwarded via the optimal path (no hair-pinning)

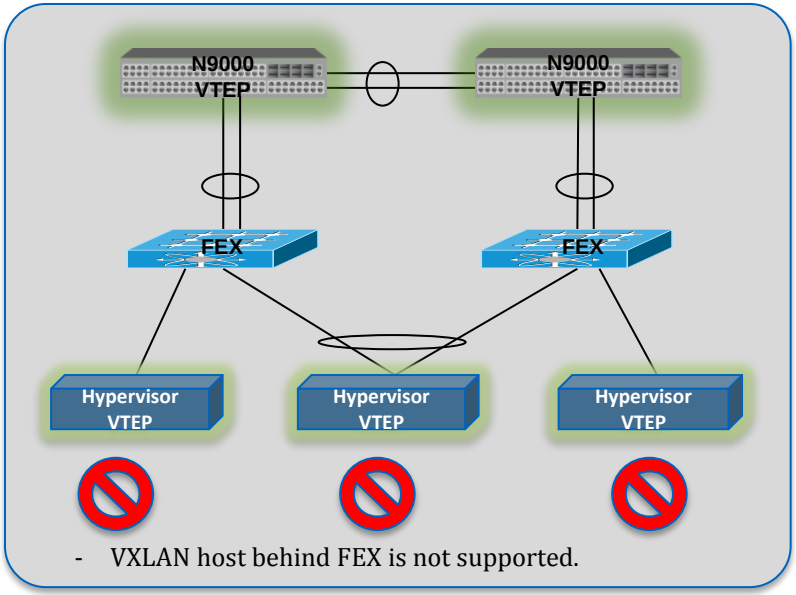


VXLAN – Unsupported Bud-node Topology



VXLAN – Supported and Unsupported FEX Topology

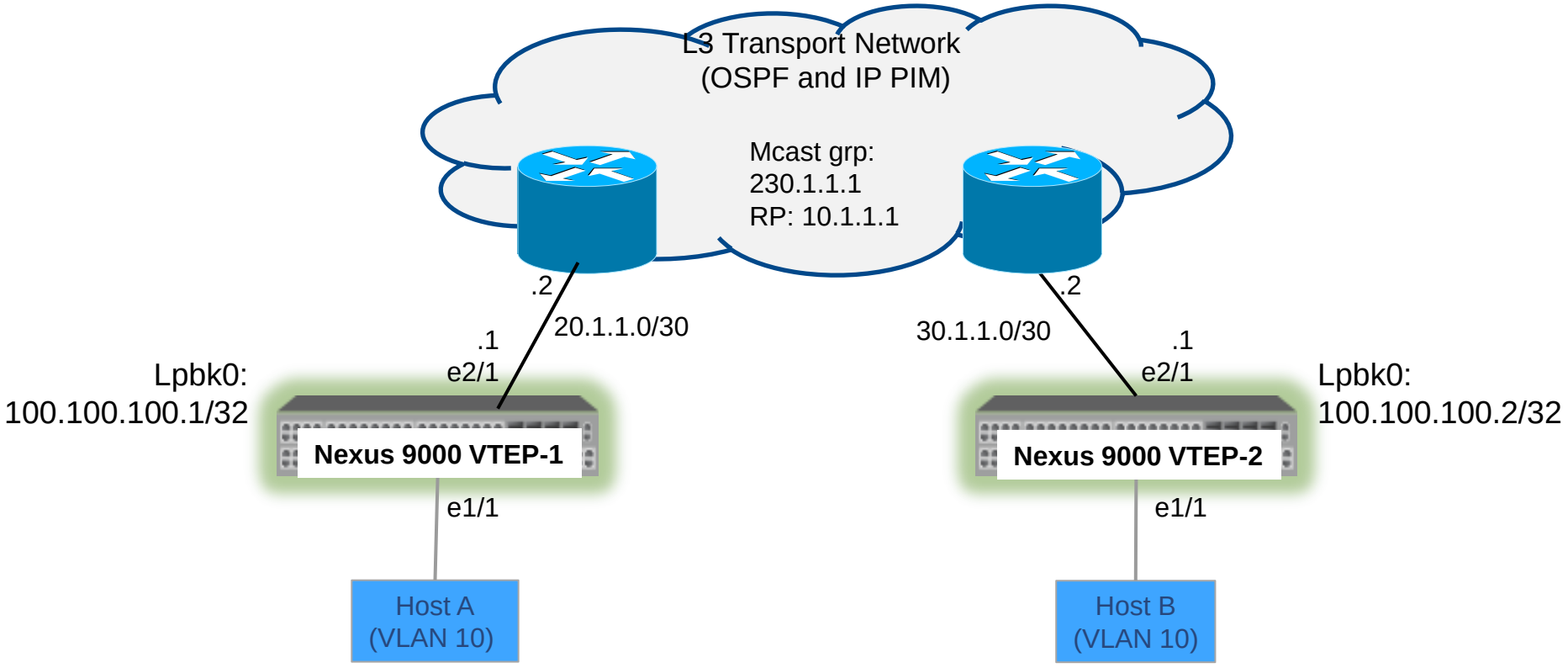




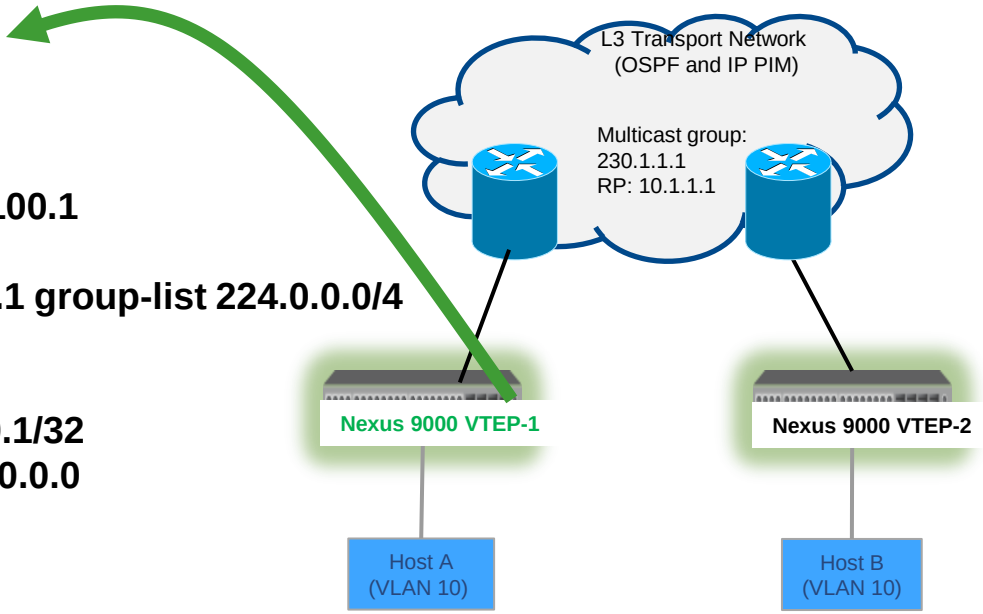
- VXLAN host behind FEX is not supported.

VXLAN 配置示例

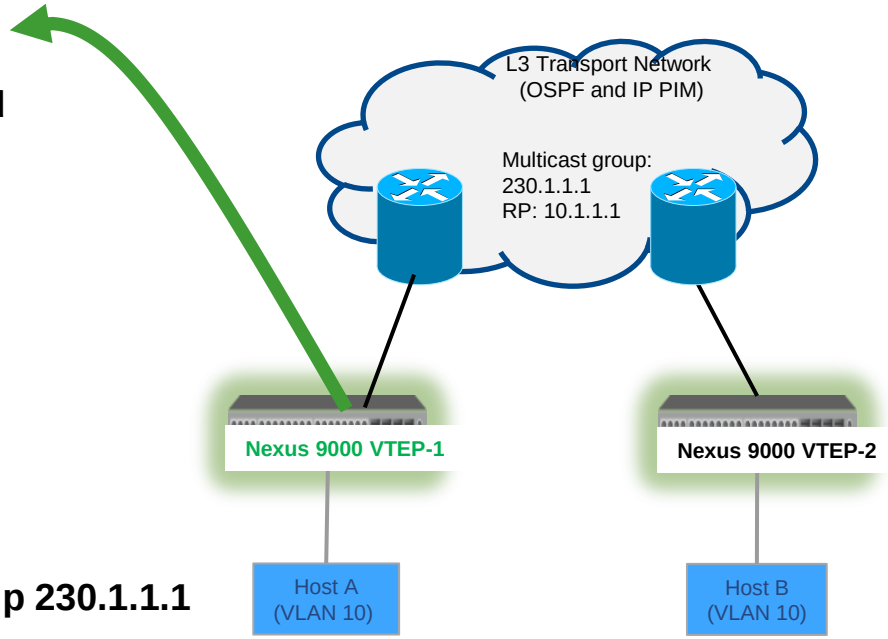
VXLAN Sample Topology - VTEP



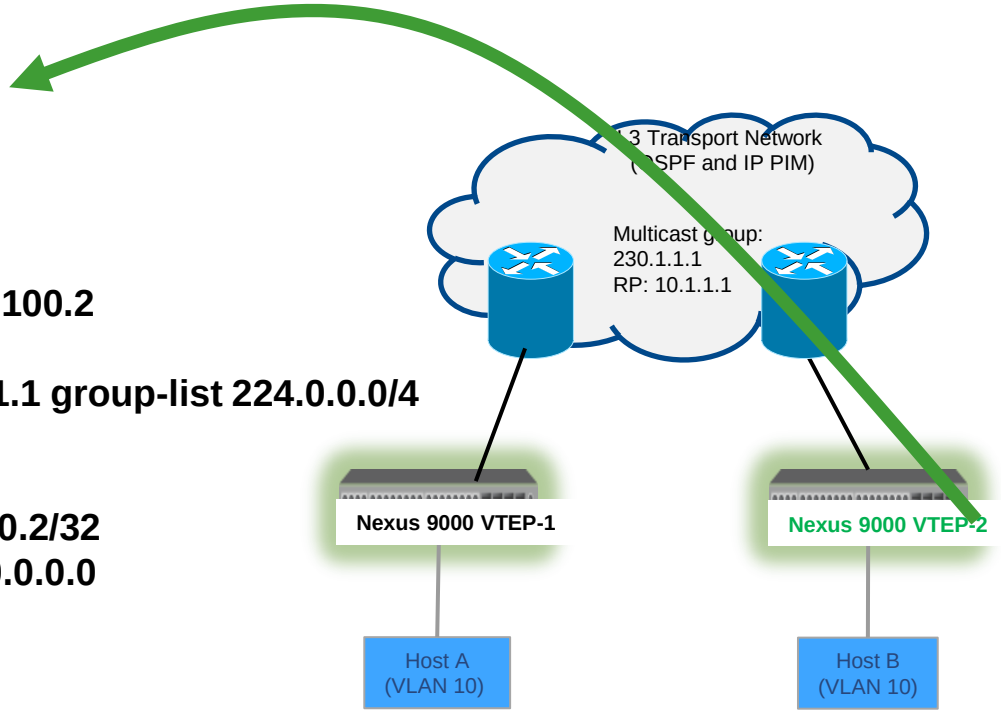
```
n9k-vtep-1(config)# feature ospf
n9k-vtep-1(config)# feature pim

n9k-vtep-1(config)# router ospf 1
n9k-vtep-1(config-router)# router-id 100.100.100.1

n9k-vtep-1(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4

n9k-vtep-1(config)# interface loopback0
n9k-vtep-1(config-if)# ip address 100.100.100.1/32
n9k-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
n9k-vtep-1(config-if)# ip pim sparse-mode

n9k-vtep-1(config)# interface e2/1
n9k-vtep-1(config-if)# ip address 20.1.1.1/30
n9k-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
n9k-vtep-1(config-if)# ip pim sparse-mode
```



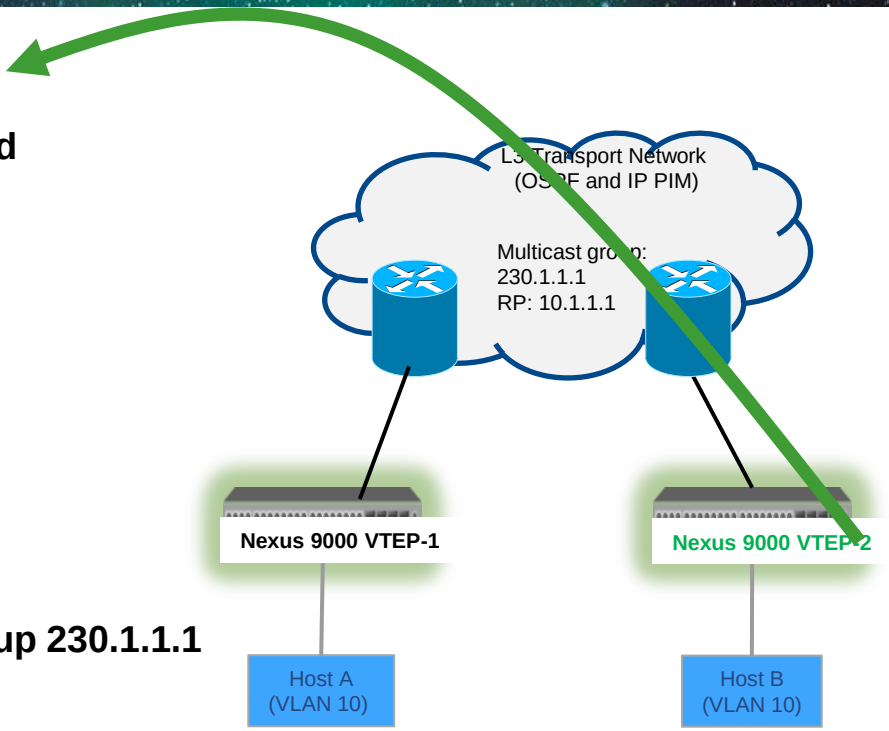
```
n9k-vtep-1(config)# feature nv overlay
n9k-vtep-1(config)# feature vn-segment-vlan-based

n9k-vtep-1(config)# interface e1/1
n9k-vtep-1(config-if)# switchport
n9k-vtep-1(config-if)# switchport access vlan 10
n9k-vtep-1(config-if)# no shutdown

n9k-vtep-1(config)# interface nve1
n9k-vtep-1(config-if)# no shutdown
n9k-vtep-1(config-if)# source-interface loopback0
n9k-vtep-1(config-if)# overlay-encapsulation vxlan
n9k-vtep-1(config-if)# member vni 1010 mcast-group 230.1.1.1

n9k-vtep-1(config)# vlan 10
n9k-vtep-1(config-vlan)# vn-segment 1010
```



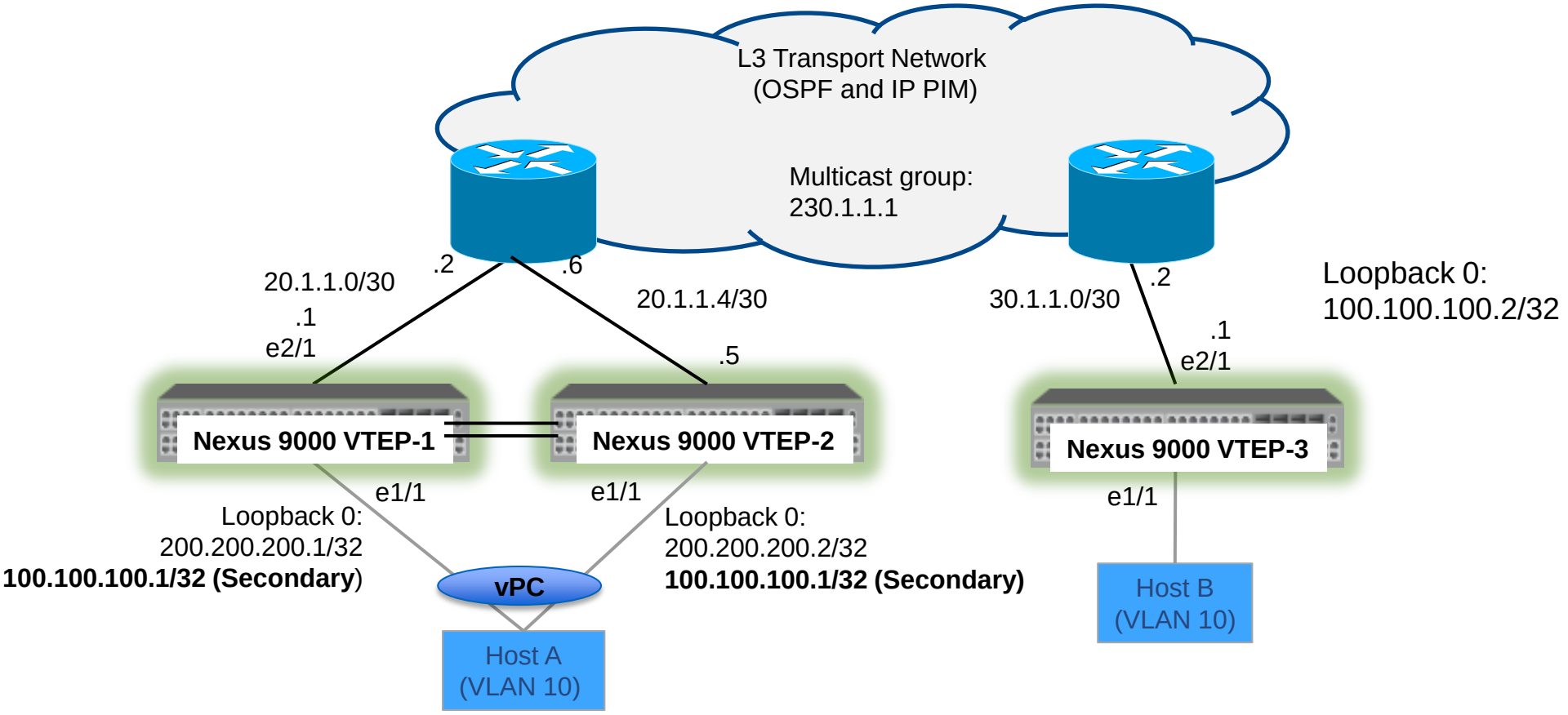
```
n9k-vtep-2(config)# feature ospf
n9k-vtep-2(config)# feature pim

n9k-vtep-2(config)# router ospf 1
n9k-vtep-2(config-router)# router-id 100.100.100.2

n9k-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4

n9k-vtep-2(config)# interface loopback0
n9k-vtep-2(config-if)# ip address 100.100.100.2/32
n9k-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
n9k-vtep-2(config-if)# ip pim sparse-mode

n9k-vtep-2(config)# interface e2/1
n9k-vtep-2(config-if)# ip address 30.1.1.1/30
n9k-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
n9k-vtep-2(config-if)# ip pim sparse-mode
```



```
n9k-vtep-2(config)# feature nv overlay
n9k-vtep-2(config)# feature vn-segment-vlan-based

n9k-vtep-2(config)# interface e1/1
n9k-vtep-2(config-if)# switchport
n9k-vtep-2(config-if)# switchport access vlan 10
 n9k-vtep-2(config-if)# no shutdown

n9k-vtep-2(config)# interface nve1
n9k-vtep-2(config-if)# no shutdown
n9k-vtep-2(config-if)# source-interface loopback0
n9k-vtep-2(config-if)# overlay-encapsulation vxlan
n9k-vtep-2(config-if)# member vni 1010 mcast-group 230.1.1.1

n9k-vtep-2(config)# vlan 10
n9k-vtep-2(config-vlan)# vn-segment 1010
```



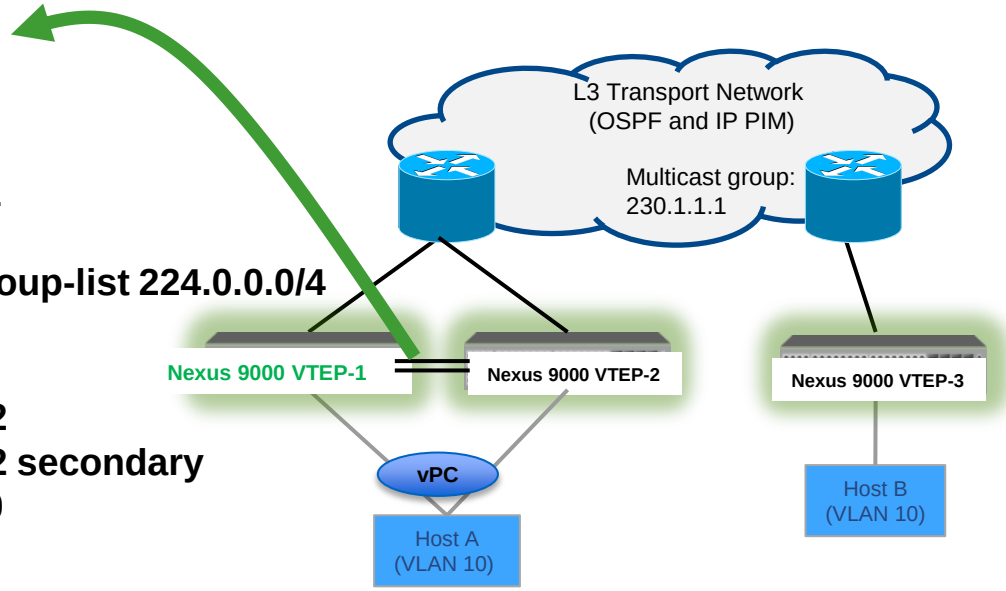
VXLAN Sample topology with vPC



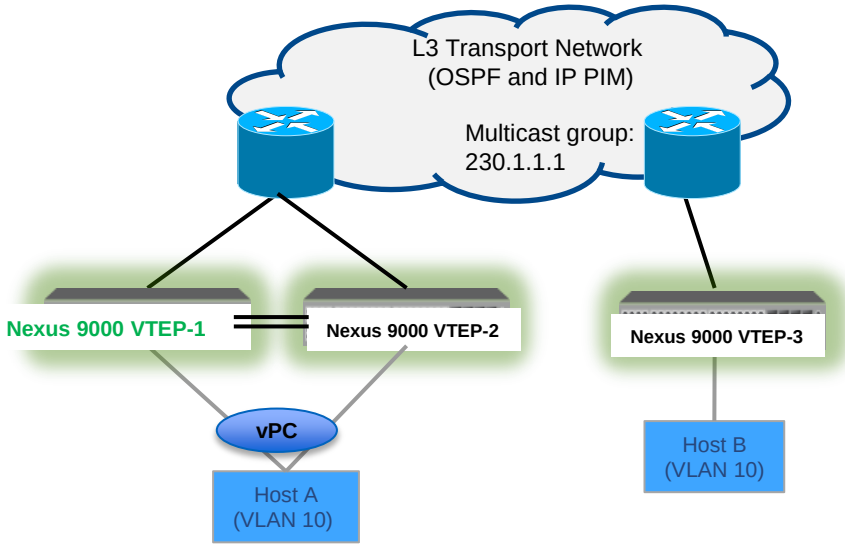
The secondary ip address is used by the emulated VTEP for VXLAN

Just make sure all the configs are identical between vPC Primary and vPC Secondary

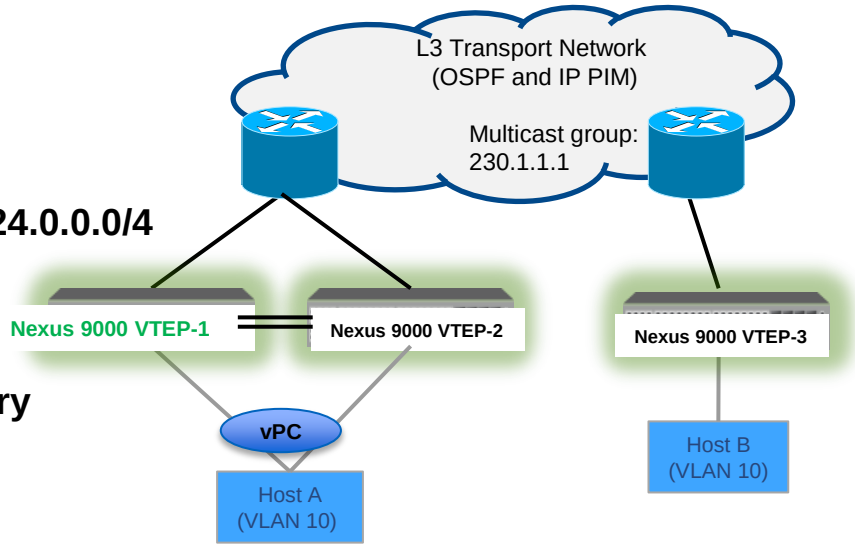
```
n9k-vtep-1(config)# feature ospf
n9k-vtep-1(config)# feature pim

n9k-vtep-1(config)# router ospf 1
n9k-vtep-1(config-router)# router-id 200.200.200.1

n9k-vtep-1(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4

n9k-vtep-1(config)# interface loopback0
n9k-vtep-1(config-if)# ip address 200.200.200.1/32
n9k-vtep-1(config-if)# ip address 100.100.100.1/32 secondary
n9k-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
n9k-vtep-1(config-if)# ip pim sparse-mode

n9k-vtep-1(config)# interface e2/1
n9k-vtep-1(config-if)# ip address 20.1.1.1/30
n9k-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
n9k-vtep-1(config-if)# ip pim sparse-mode


n9k-vtep-1(config)# feature nv overlay
n9k-vtep-1(config)# feature vn-segment-vlan-based
```



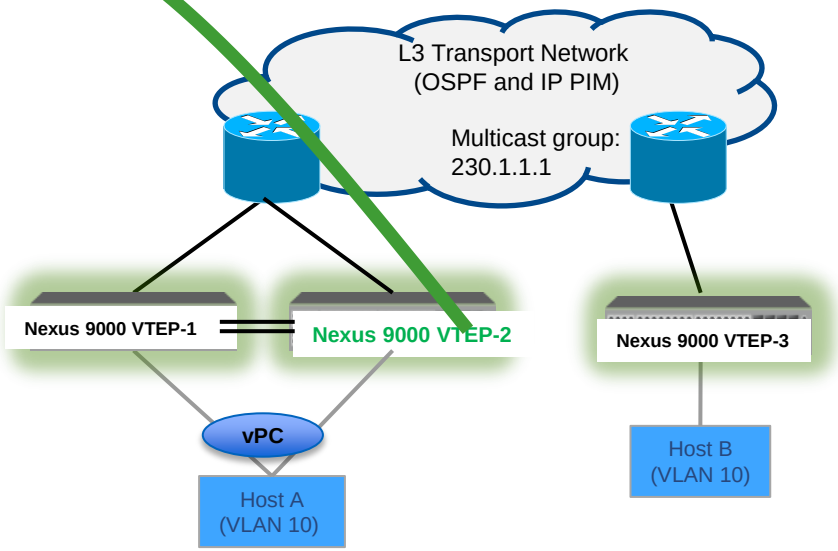
```
n9k-vtep-1(config)# interface port-channel 10
n9k-vtep-1(config-if)# vpc 10
n9k-vtep-1(config-if)# switchport
n9k-vtep-1(config-if)# switchport mode access
n9k-vtep-1(config-if)# switchport access vlan 10
n9k-vtep-1(config-if)# no shutdown

n9k-vtep-1(config)# interface e1/1
n9k-vtep-1(config-if)# channel-group 10 mode active
n9k-vtep-1(config-if)# no shutdown

n9k-vtep-1(config)# interface nve1
n9k-vtep-1(config-if)# no shutdown
n9k-vtep-1(config-if)# source-interface loopback0
n9k-vtep-1(config-if)# overlay-encapsulation vxlan
n9k-vtep-1(config-if)# member vni 1010 mcast-group 230.1.1.1

n9k-vtep-1(config)# vlan 10
n9k-vtep-1(config-vlan)# vn-segment 1010
```



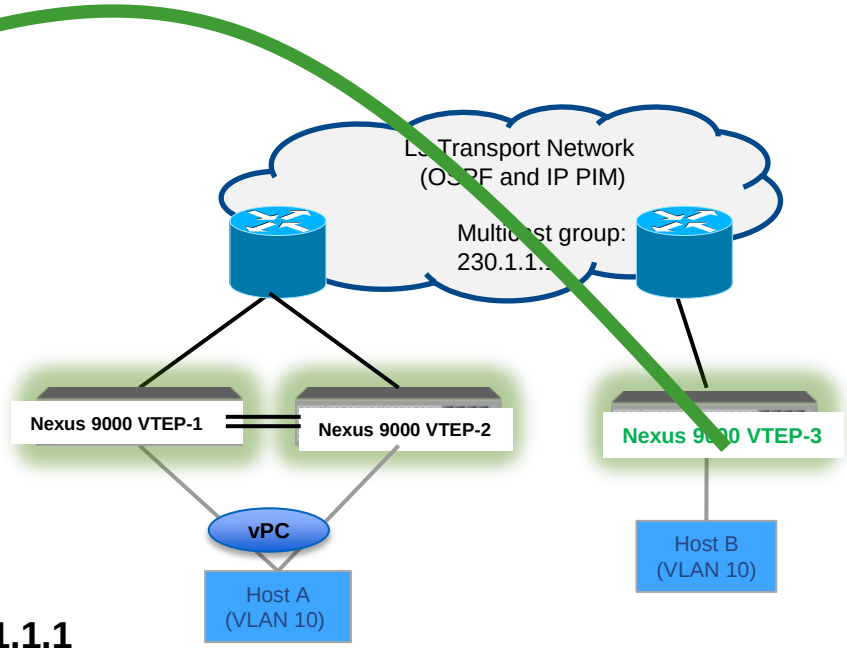
```
n9k-vtep-2(config)# feature ospf
n9k-vtep-2(config)# feature pim

n9k-vtep-2(config)# router ospf 1
n9k-vtep-2(config-router)# router-id 200.200.200.2

n9k-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4

n9k-vtep-2(config)# interface loopback0
n9k-vtep-2(config-if)# ip address 200.200.200.2/32
n9k-vtep-2(config-if)# ip address 100.100.100.1/32 secondary
n9k-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
n9k-vtep-2(config-if)# ip pim sparse-mode

n9k-vtep-2(config)# interface e2/1
n9k-vtep-2(config-if)# ip address 20.1.1.5/30
n9k-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
n9k-vtep-2(config-if)# ip pim sparse-mode

n9k-vtep-2(config)# feature nv overlay
n9k-vtep-2(config)# feature vn-segment-vlan-based
```



```
n9k-vtep-2(config)# interface port-channel 10
n9k-vtep-2(config-if)# vpc 10
n9k-vtep-2(config-if)# switchport
n9k-vtep-2(config-if)# switchport mode access
n9k-vtep-2(config-if)# switchport access vlan 10
n9k-vtep-2(config-if)# no shutdown

n9k-vtep-2(config)# interface e1/1
n9k-vtep-2(config-if)# channel-group 10 mode active
n9k-vtep-2(config-if)# no shutdown

n9k-vtep-2(config)# interface nve1
n9k-vtep-2(config-if)# no shutdown
n9k-vtep-2(config-if)# source-interface loopback0
n9k-vtep-2(config-if)# overlay-encapsulation vxlan
n9k-vtep-2(config-if)# member vni 1010 mcast-group 230.1.1.1

n9k-vtep-2(config)# vlan 10
n9k-vtep-2(config-vlan)# vn-segment 1010
```



```
n9k-vtep-3(config)# feature nv overlay
n9k-vtep-3(config)# feature vn-segment-vlan-based

n9k-vtep-3(config)# interface e1/1
n9k-vtep-3(config-if)# switchport
n9k-vtep-3(config-if)# switchport access vlan 10
n9k-vtep-3(config-if)# no shutdown

n9k-vtep-3(config)# interface nve1
n9k-vtep-3(config-if)# no shutdown
n9k-vtep-3(config-if)# source-interface loopback0
n9k-vtep-3(config-if)# overlay-encapsulation vxlan
n9k-vtep-3(config-if)# member vni 1010 mcast-group 230.1.1.1

n9k-vtep-3(config)# vlan 10
n9k-vtep-3(config-vlan)# vn-segment 1010
```



Troubleshooting

Verify Overlay Peer

```
switch# show nve peers
Interface           Peer-IP                 VNI         Up Time
-------------       -----------------       --------    -------
nve1                10.1.1.2                1010        00:52:24
```

Verify Overlay interface

```
switch# show nve vni
Interface           VNI         Multicast-group    VNI State
----------------    --------    ---------------    ---------
nve1                1010        230.1.1.1             up
```

验证 MAC地址学习情况

```
switch# show mac address-table
Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O -
Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC
Peer-Link,
        (T) - True, (F) - False
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+-------
-----------
*  10     0000.aa01.0001   dynamic  0         F      F    nve1
*  10     0000.bb01.0001   dynamic  0         F      F    nve1
*  10     0000.bb01.0001   dynamic  0         F      F    Eth1/46
```

- It displays VTEP peers and the associated VNI with them. The output displays time since the peer was last detected / learnt.
- List of all VNIs that are associated with various nve interfaces and the associated multicast IP address.

现阶段VXLAN 技术小结

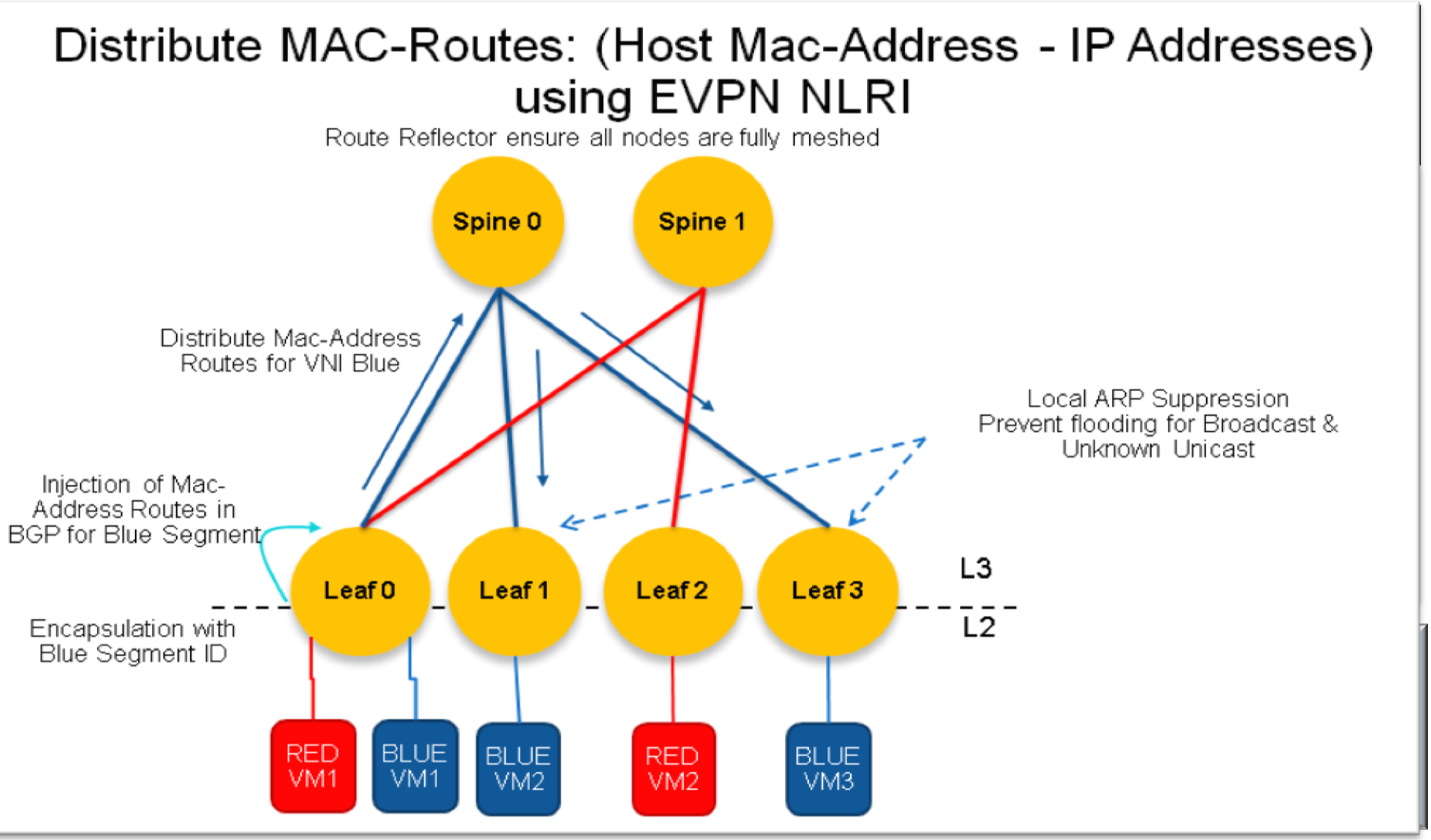
- VXLAN 简单配置即可实现二层网络的扩展
- VXLAN 扩展二层交换域超越 4K VLAN 限制
- VXLAN 可以提供ECMP能力，但是部署必须依靠底层网络交换矩阵架构的稳定性、可靠性、扩展性。
- VXLAN 的控制平面问题

现阶段VXLAN部署的问题—— 控制平面

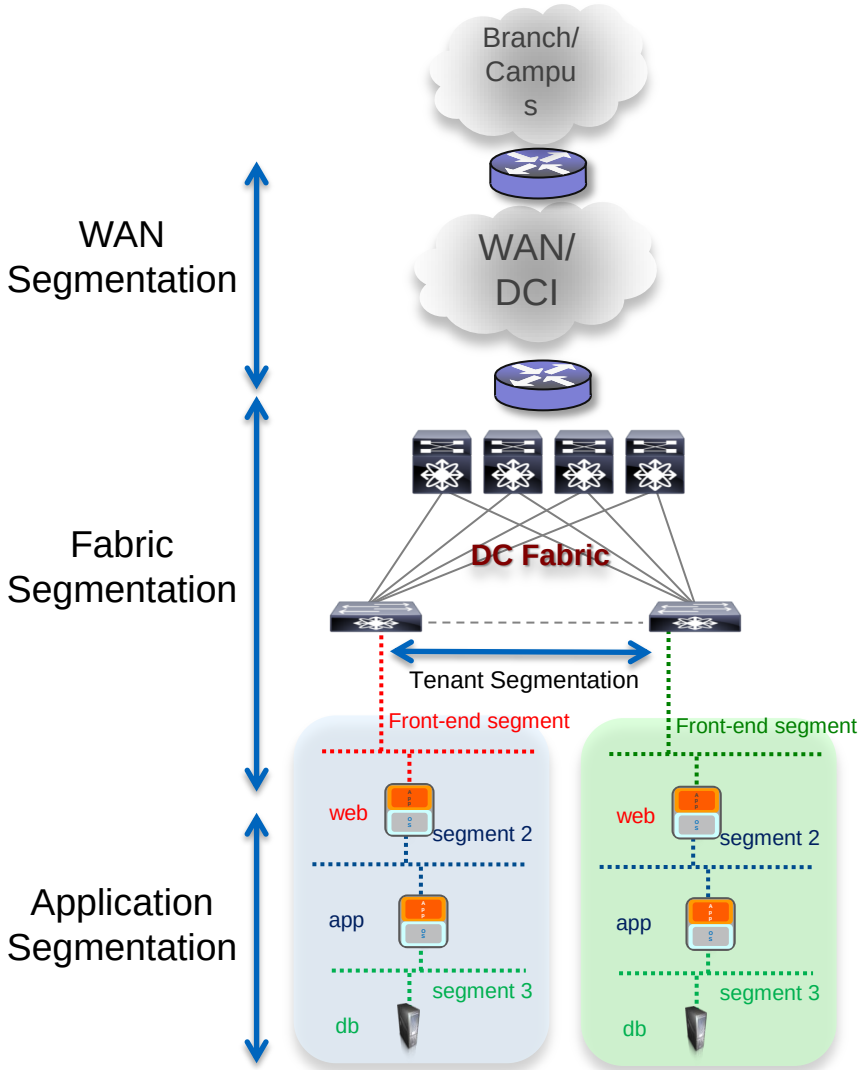
- 挑战
  - 数据中心核心网络不支持组播
  - 组播泛洪造成的扩展性问题
  - VNI到组播组的映射带来的管理问题
  - VXLAN成员 VTEP自动发现
  - 多宿主网关

- 实现方法选择:
  - **扩展路由协议提供控制平面**
    - EVPN （MP-BGP）
    - LISP
  - **提供API 接口，依靠策略服务器控制**
    - REST Based APIs ， Python 脚本 提供给第三方业务编排软件 （SDN软件方式）
    - 以应用为中心的基础架构 ACI （硬件化的SDN?）

VXLAN 的控制平面的选择示例 - EVPN

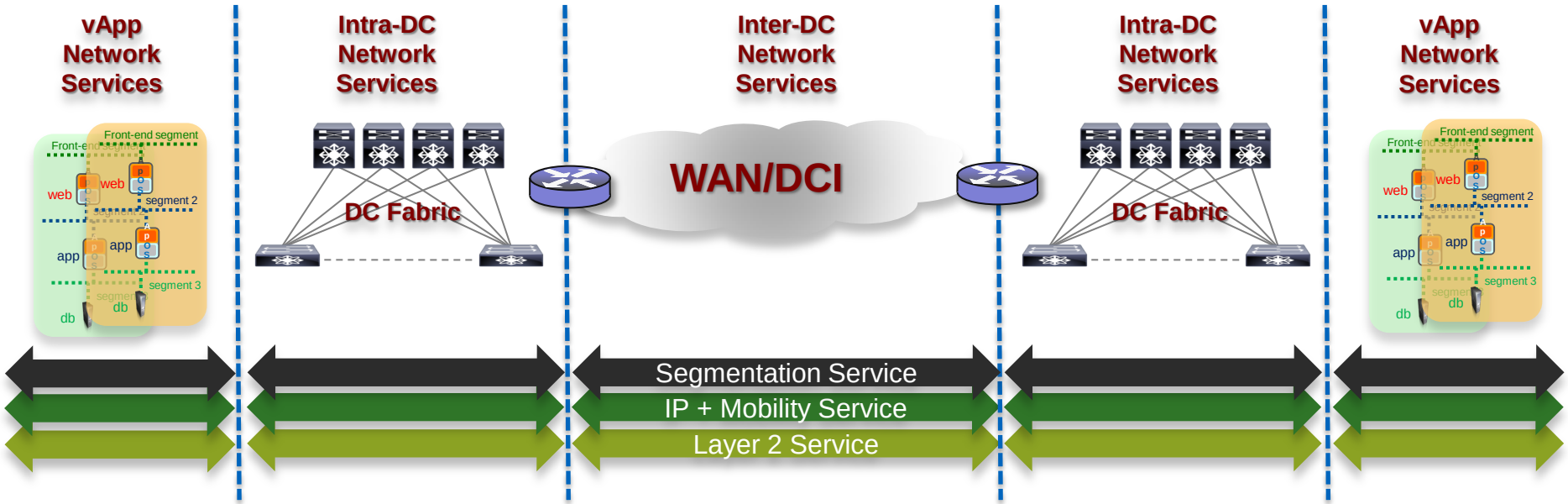


端到端部署网络分区



- Segmentation at many levels
- Must be given continuity
  - Across the different network places
  - Across organizations and administrative boundaries
- All relevant technologies include the required segmentation semantics
- The network maps the segments together to provide a scalable and interoperable e2e segmentation solution

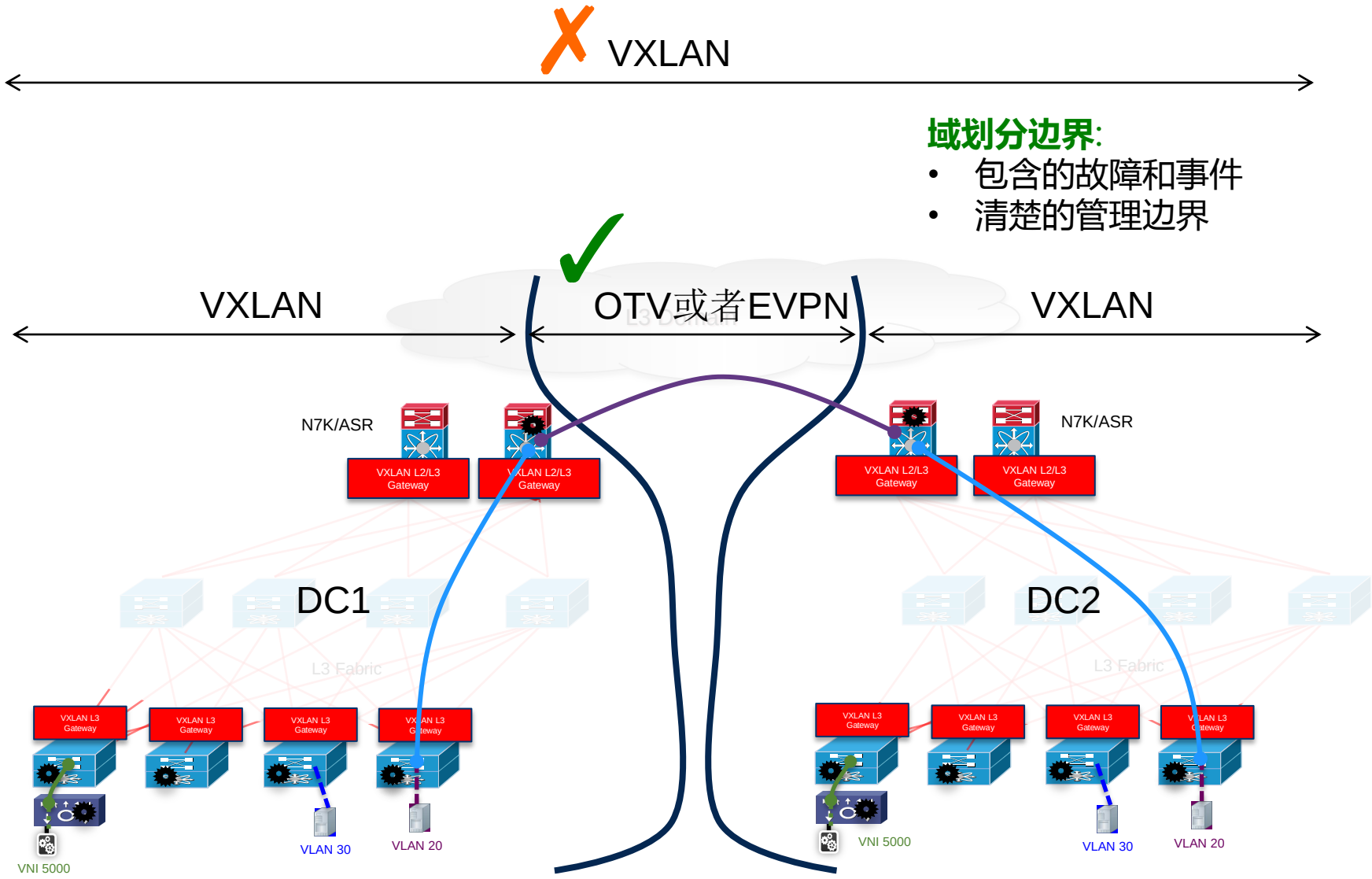
故障域范围划分



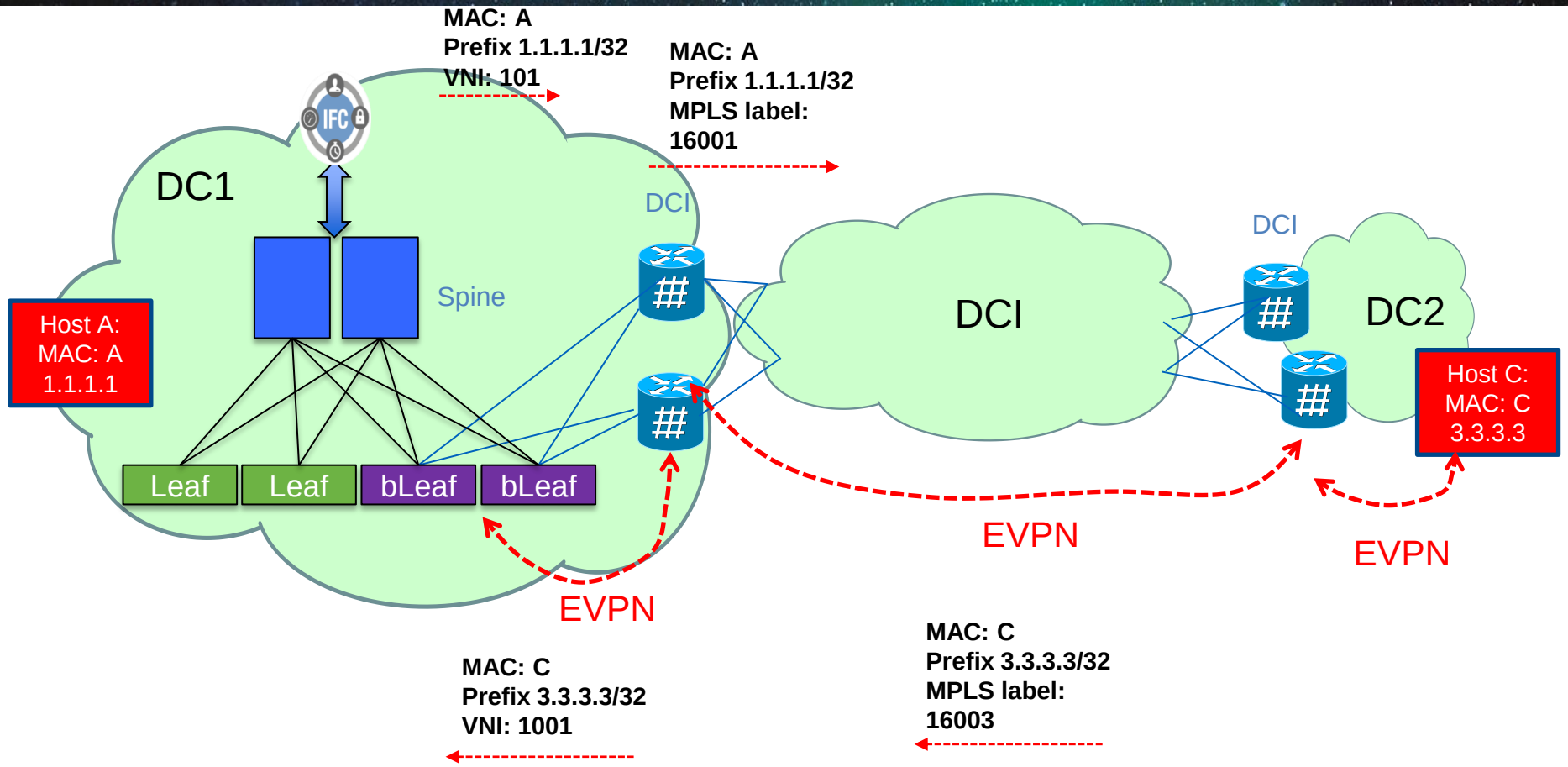
Core Principles of Network Resiliency/Scale applied to Overlay Services

- Clearly delineated Fault Boundaries and service domains
- Control Plane Hierarchy and Federation within and across domains
- Data Plane Boundaries
- Administrative Domain Delineation and Federation

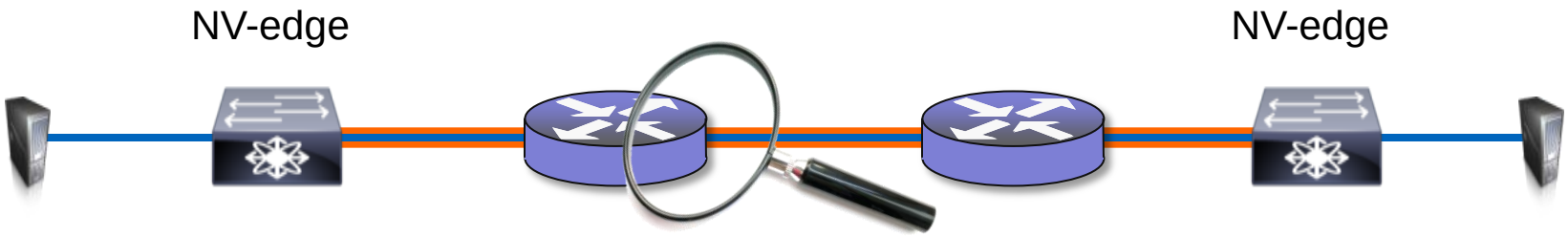
多个数据中心的互联—— LAN 扩展



端到端EVPN 控制平面示例

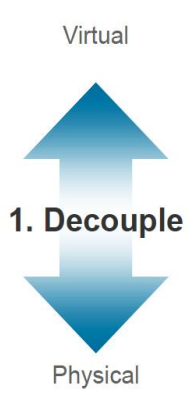


另外的问题：如何对Overlay 报文可视化监控和处理

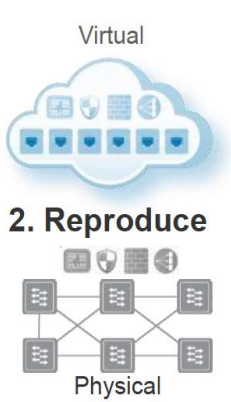


- Infrastructure awareness of encapsulated traffic:
  - Outer/Encapsulation header
  - Overlay shim header
  - Internal/Payload header
  - Payload
- Overlay aware Switching & Routing infrastructure:
  - ACLs, QoS, Netflow
- Network Analysis Module (NAM) inspects encapsulated traffic

SDN（软件定义的网络）

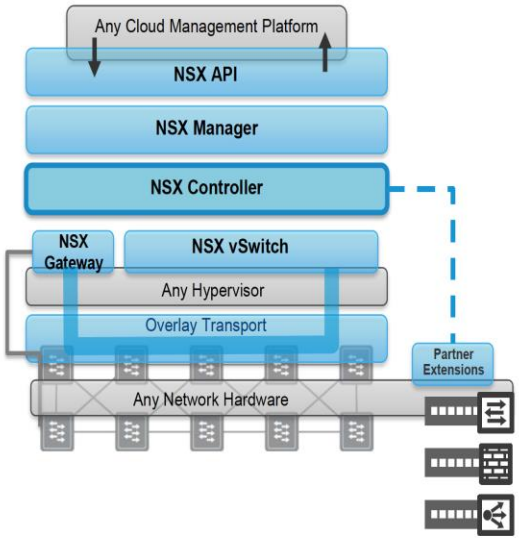


Hardware independence



No change to network from end host perspective

Operational benefits of virtualization



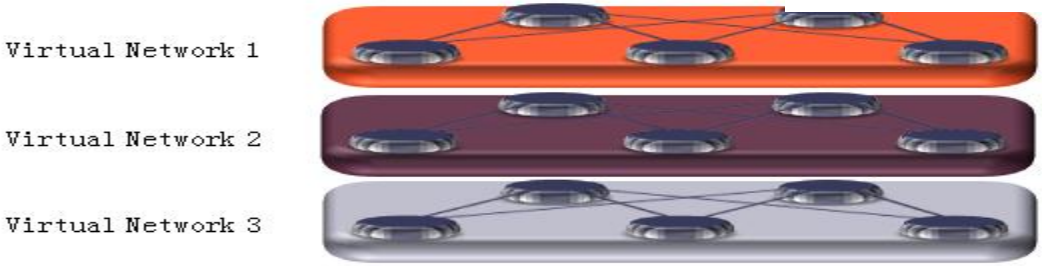
NSX Controller Cluster
Advanced distributed state management system manages virtual networks and overlay transport tunnels

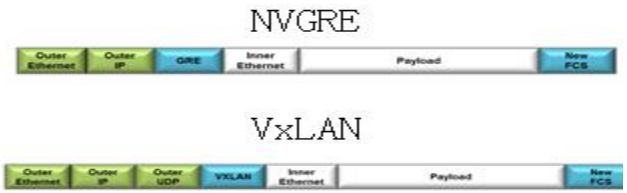
NSX vSwitch
Intelligence moves to the edge, in software, embedded in server hypervisor

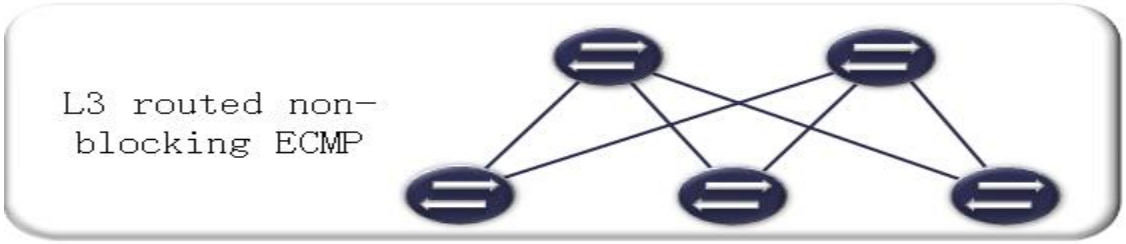
Logical Switches
Hypervisor to hypervisor high performance, vendor independent, transport on any physical fabric architecture

NSX Gateway
Supports connection between logical and physical networks – VLANs, Routing

Network API
RESTful web services API opens programmatic interface to virtual network infrastructure

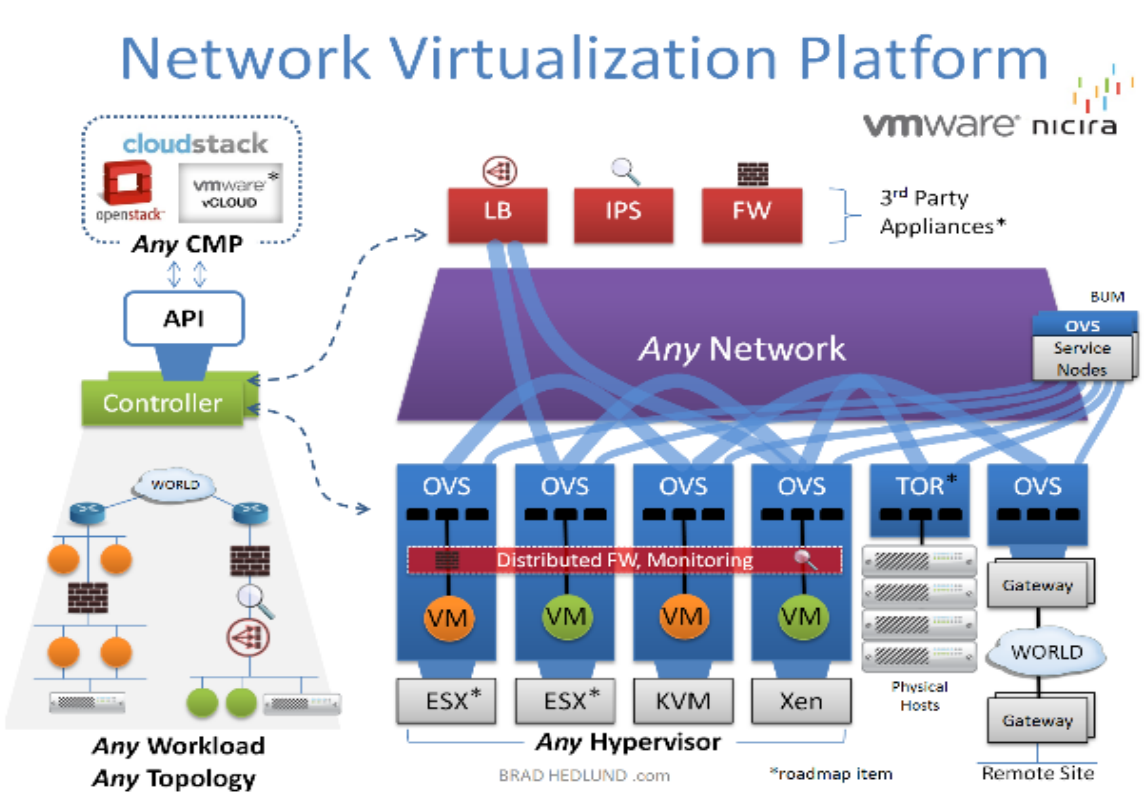






Encapsulated traffic carried over L3 fabrics

SDN 的例子
Vmware Nicira的NVP



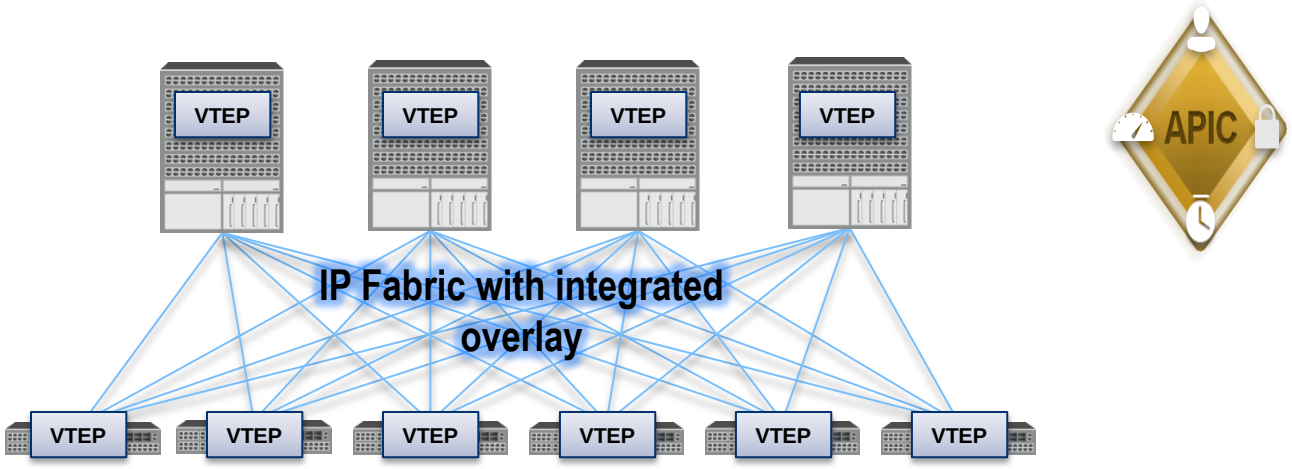
NVP 忽略了应用对底层硬件网络的要求，代价:

- 每台服务器和虚拟机的开销增加
- 需要第3方硬件网关设备支持，集成组件多，管理复杂
- 数据中心非虚拟化的服务器与其交互复杂
- 管理节点受限，扩展性差
- 无法感知应用，而虚拟机上应用的信息被NSX增加的包头所掩盖，物理网络的QoS无法生效
- 无法实现应用可视，不利于故障的检测、隔离和排除

纯软件SDN的解决方式 ，实际上用户的应用性能得不到保证。

Cisco ACI Fabric

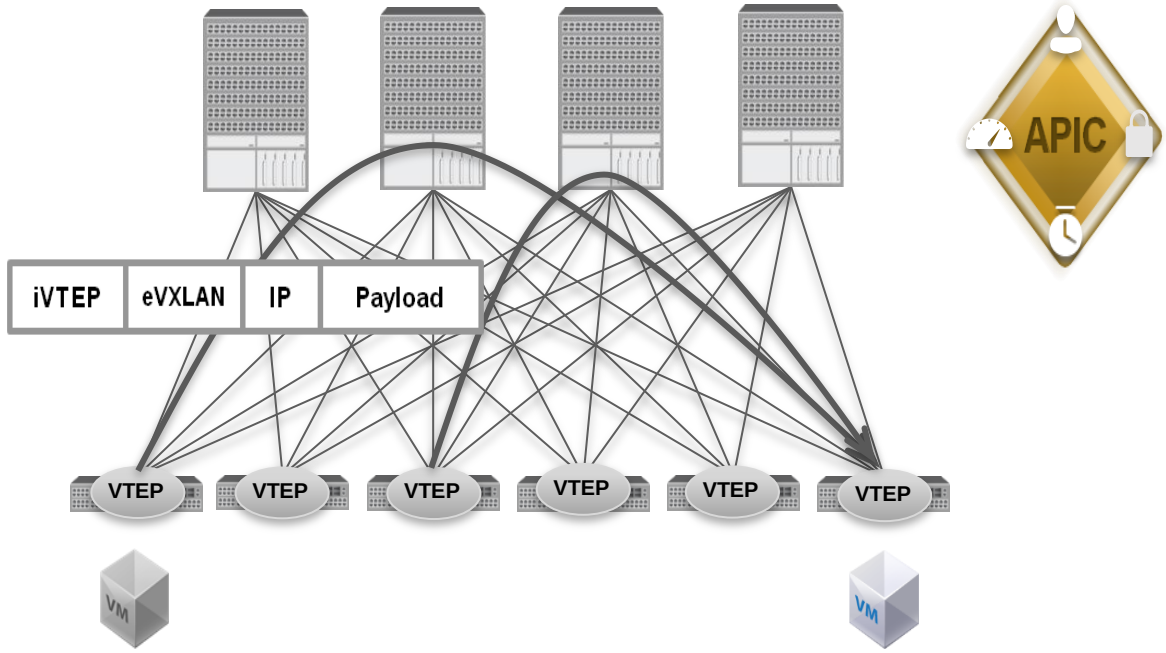
Why an Integrated Overlay?



- Mobility, Scale, Multi-Tenancy & integration with emerging hypervisor designs
- Flexible any-to-any Addressing and Host connectivity
- Data Traffic can now carry explicit Meta-Data that allows for distributed policy and extensions

Cisco ACI Fabric

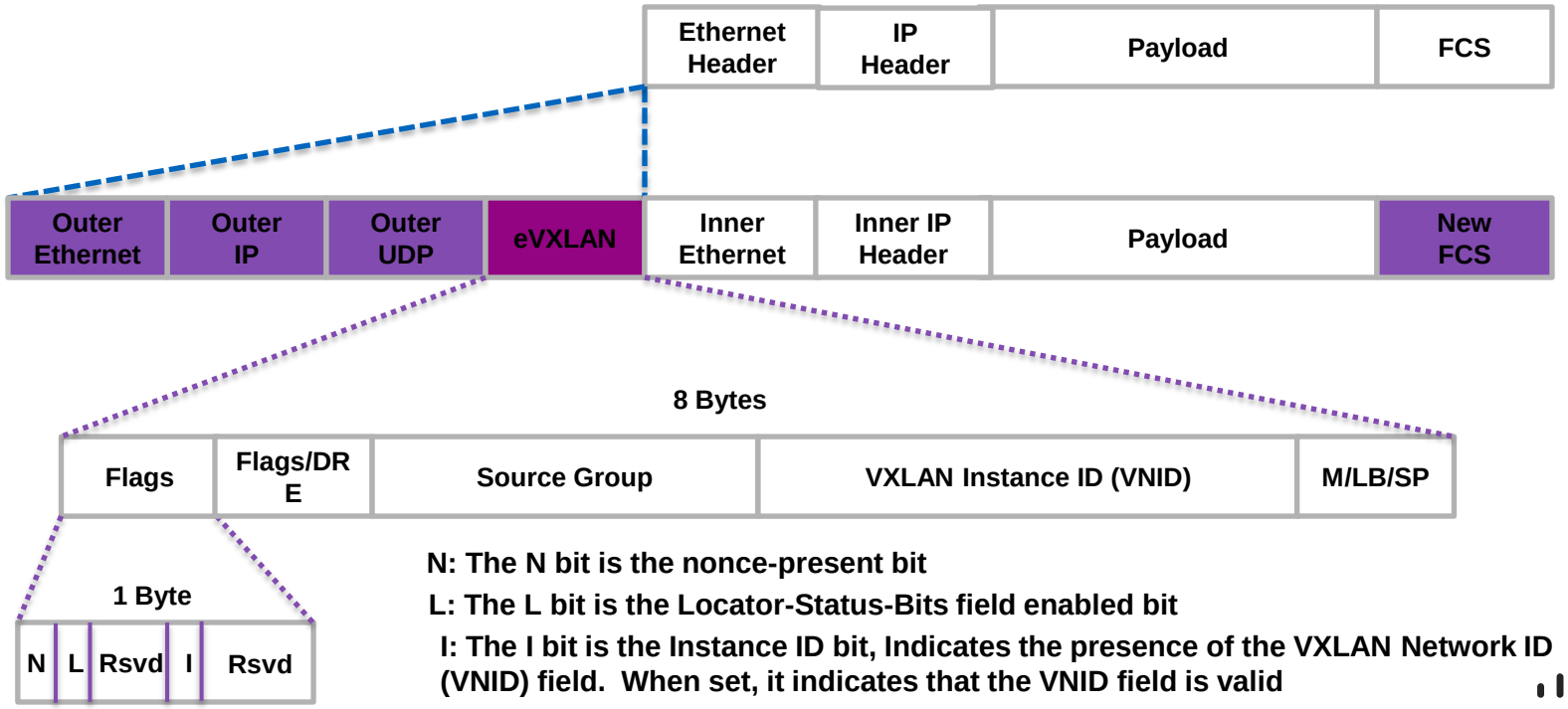
Decoupled Identity, Location & Policy



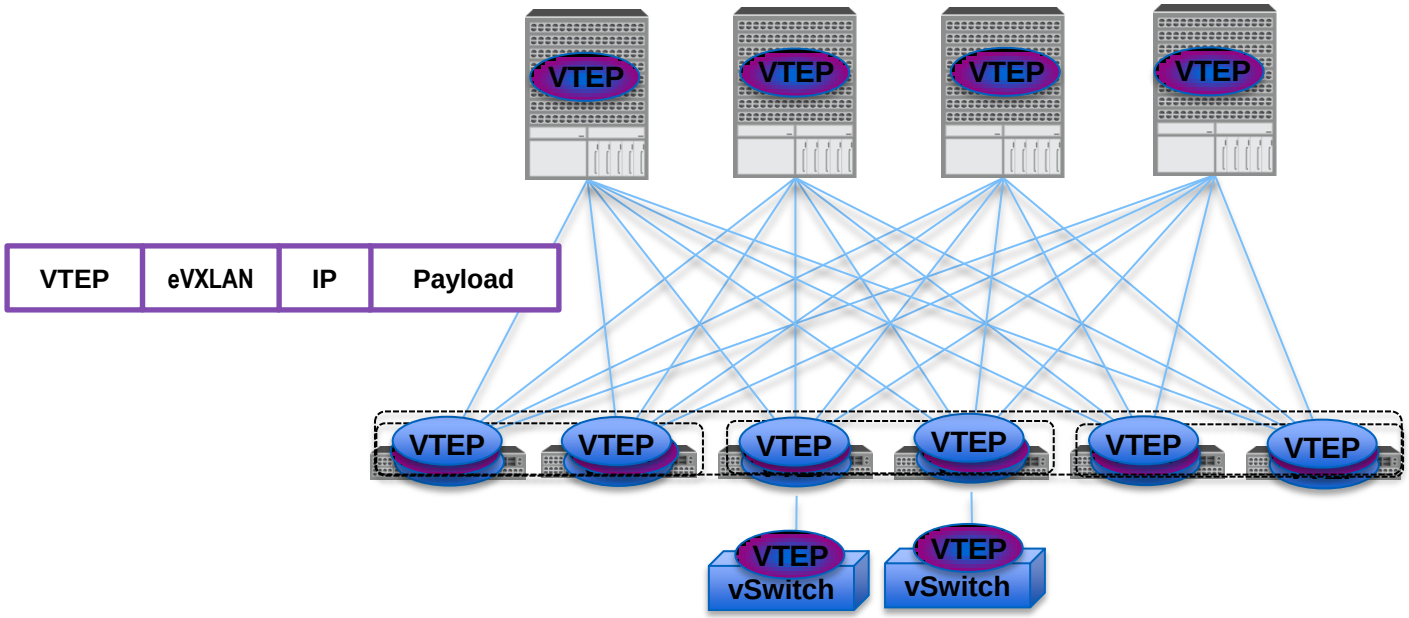
- ACI Fabric decouples the tenant end-point address, its “identifier” from the location of that end-point which is defined by its “locator” or VTEP address
- Forwarding within the Fabric is between VTEPs (eVXLAN tunnel endpoints) and leverages an enhanced VXLAN header format referred to as the eVXLAN policy header
- The mapping of the internal tenant MAC or IP address to location is performed by the VTEP using a distributed mapping (reachability) database

ACI Fabric
ACI VXLAN (eVXLAN) Header

ACI VXLAN (eVXLAN) header provides a tagging mechanism to identify properties associated with frames forwarded through an ACI capable fabric. It is an extension of the Layer 2 LISP protocol (draft-smith-lisp-layer2-01) with the additional of **policy group, load and path metric, counter and ingress port and encapsulation** information. The eVXLAN header is not associated with a specific L2 segment or L3 domain but provides a multi-function tagging mechanism used in ACI Application Defined Networking enabled fabric.



ACI Fabric – Integrated Overlay
Decoupled Identity, Location & Policy

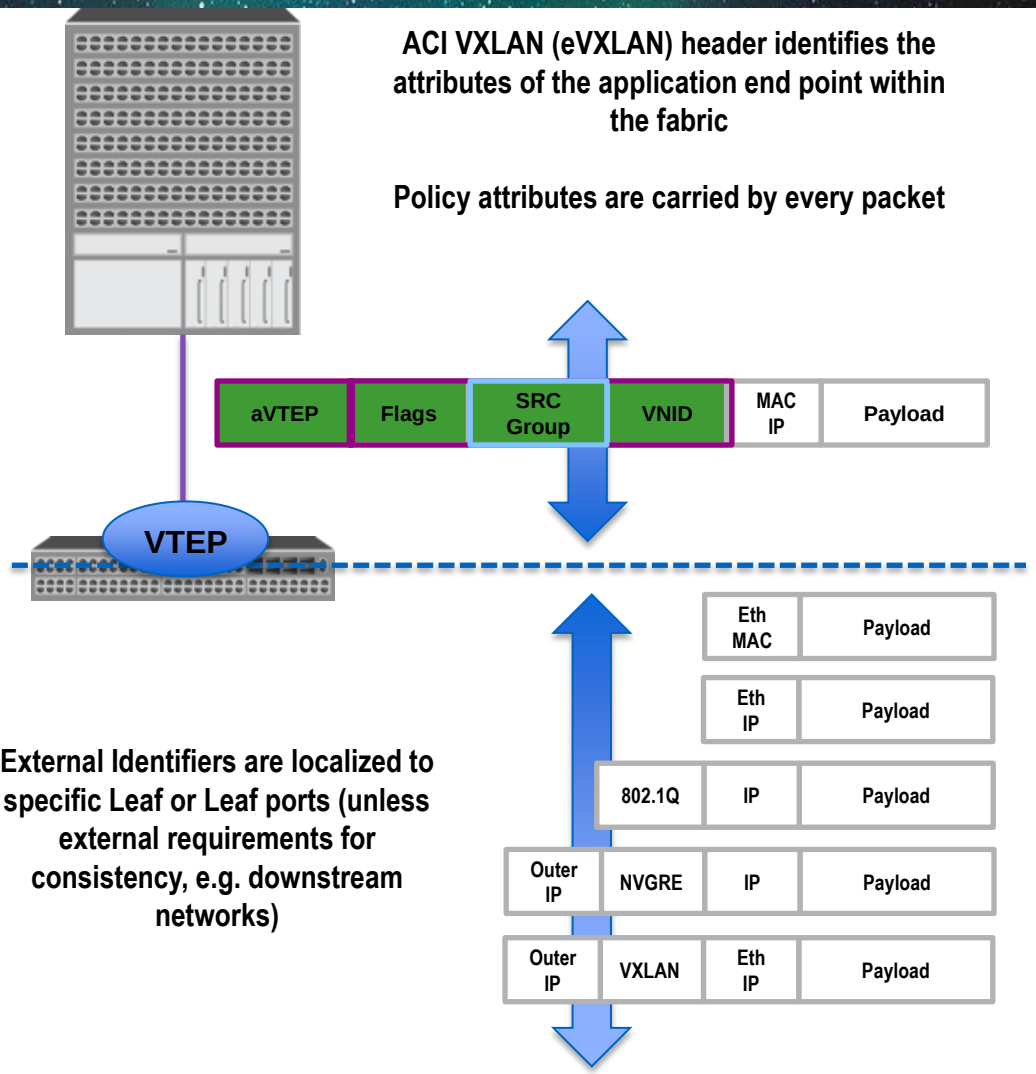


- Different VTEP's are used for different forwarding functions within the fabric
- Known unicast traffic forwarded directly between Leaf VTEP's
- Unknown unicast traffic is forwarded to anycast spine proxy VTEP's
- Logical vPC switch is represented by anycast Leaf vPC VTEP's
- Multicast and any allowed broadcast traffic is forwarded to a Group VTEP that exists on any leaf with membership for that specific group
- VTEP's may exist in physical or virtual switches

ACI Fabric – Integrated Overlay

ACI VXLAN (eVXLAN) Header

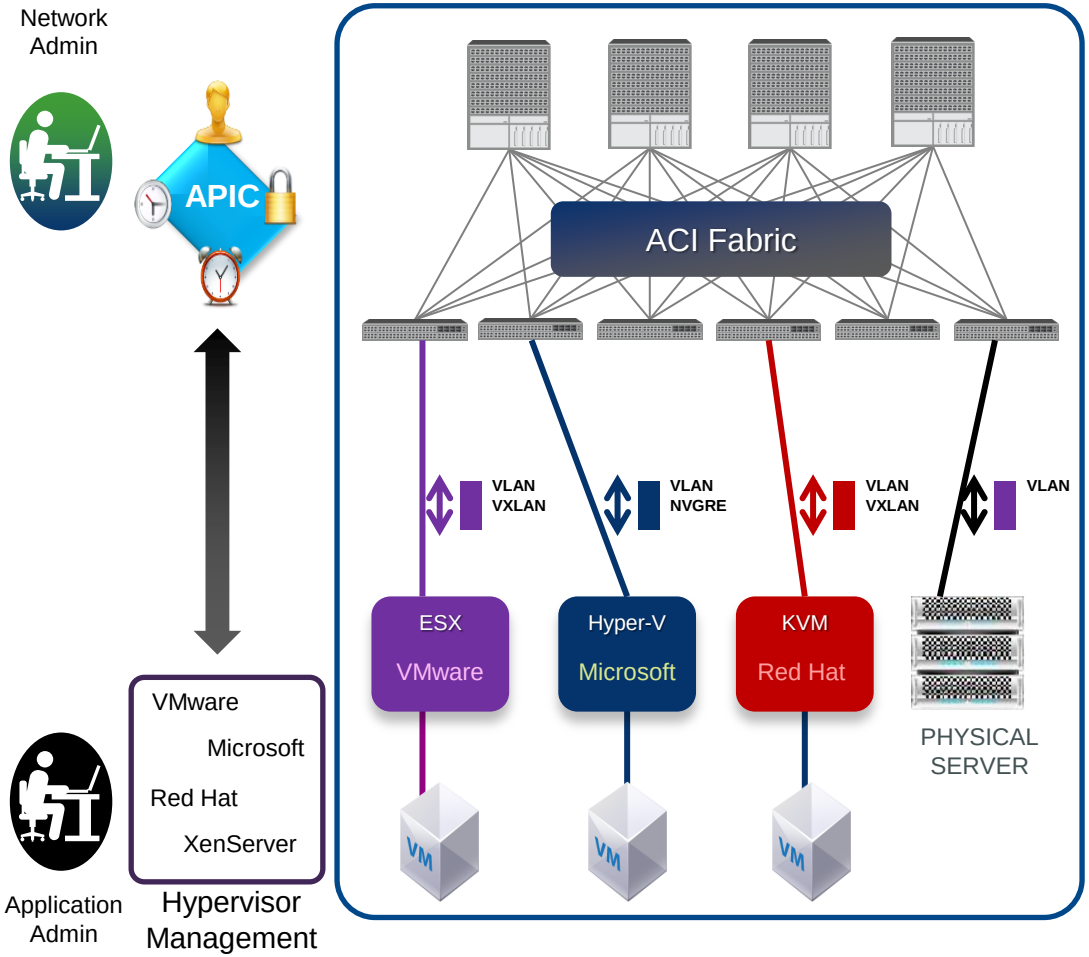
- All Tenant traffic within the Fabric is tagged with an ACI VXLAN (eVXLAN) header which identifies the policy attributes of the application end point within the fabric
  - Policy Group (source group)
  - Forwarding Group (Tenant, VRF, Bridge Domain)
  - Load Balancing Policy
  - Telemetry Policy
- At the ingress port the Fabric translates an external identifier which can be used to distinguish different application end points via the internal eVXLAN tagging format



ACI Fabric – Integrated Overlay

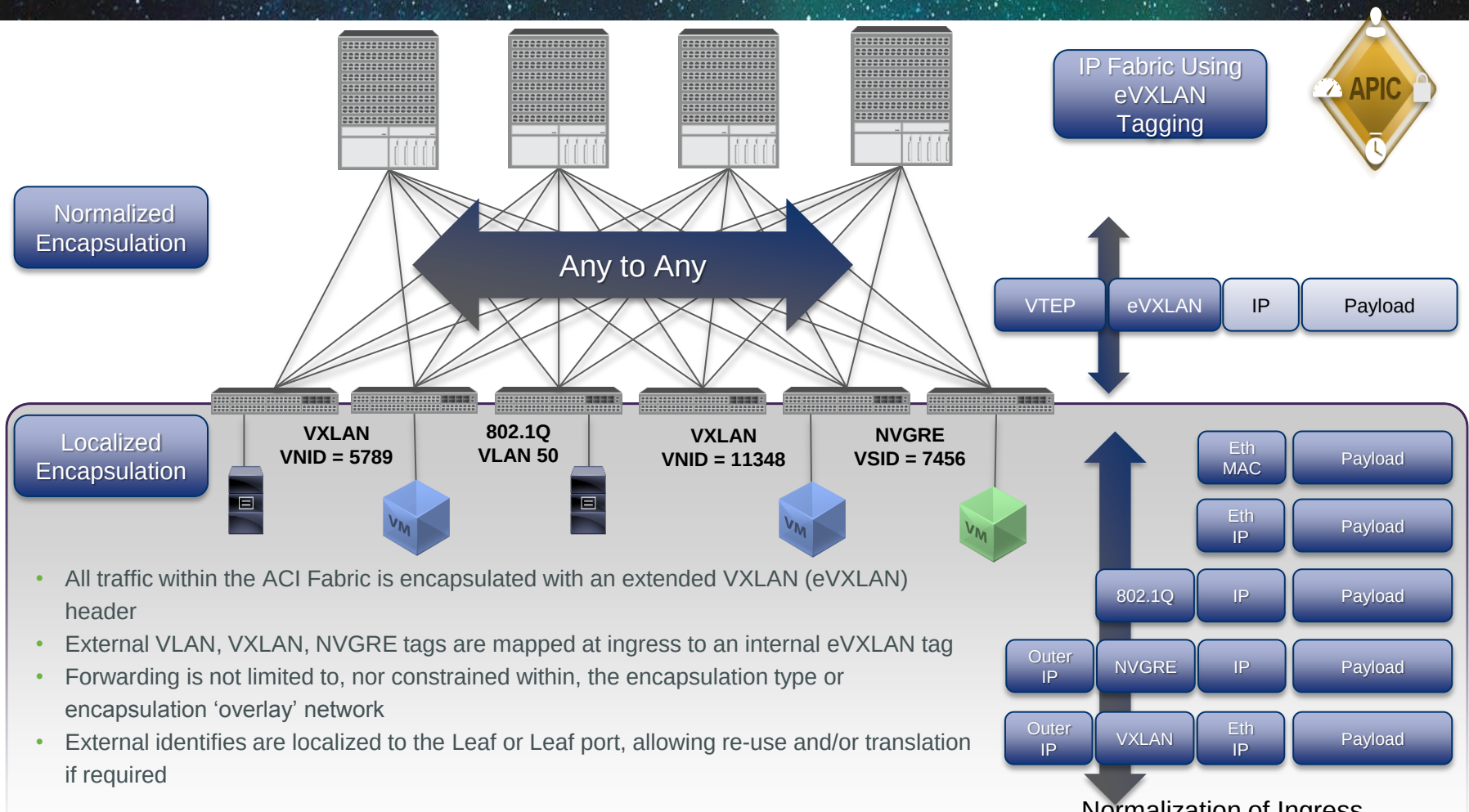
Multi-Hypervisor Encapsulation Normalization

- Integrated gateway for VLAN, VxLAN, and NVGRE networks from virtual to physical
- Normalization for NVGRE, VXLAN, and VLAN networks
- Customer not restricted by a choice of hypervisor
- Fabric is ready for multi-hypervisor



ACI Fabric – Integrated Overlay

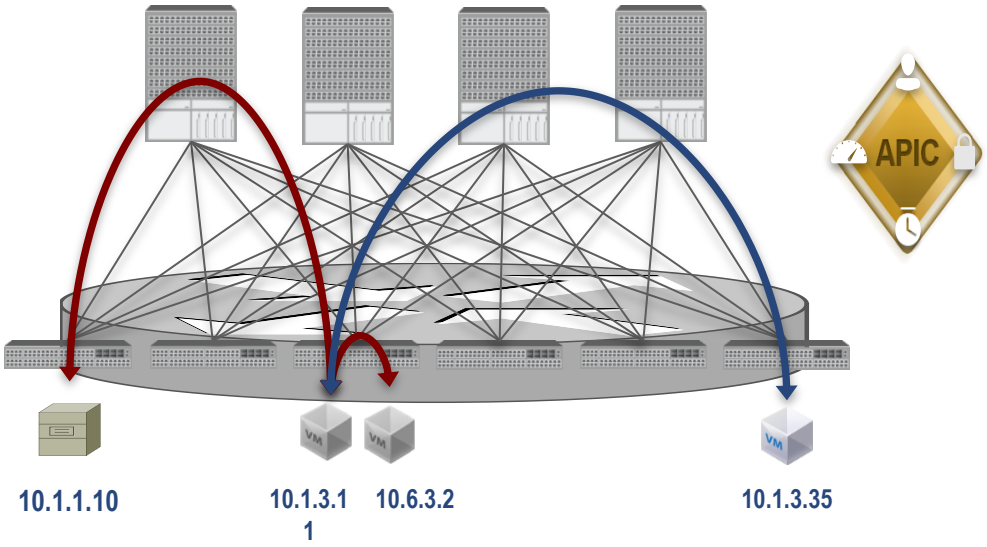
Multi-Hypervisor Encapsulation Normalization



- All traffic within the ACI Fabric is encapsulated with an extended VXLAN (eVXLAN) header
- External VLAN, VXLAN, NVGRE tags are mapped at ingress to an internal eVXLAN tag
- Forwarding is not limited to, nor constrained within, the encapsulation type or encapsulation 'overlay' network
- External identifies are localized to the Leaf or Leaf port, allowing re-use and/or translation if required

Location Independent Forwarding

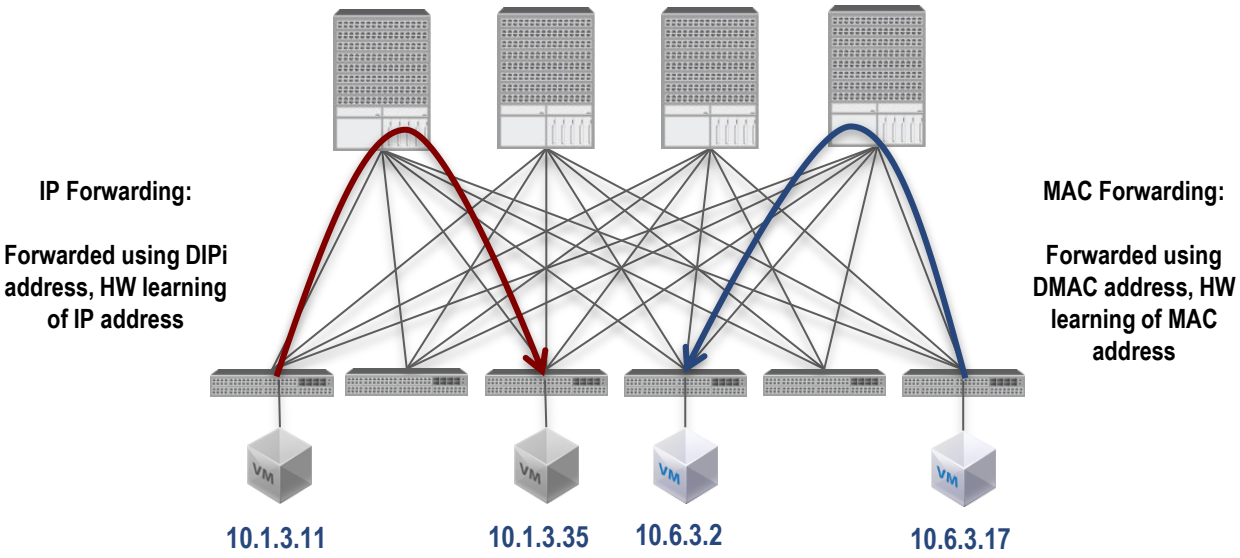
Distributed Default (Pervasive) Gateway



- ACI Fabric supports full layer 2 and layer 3 forwarding semantics, no changes required to applications or end point IP stacks
- ACI Fabric provides optimal forwarding for layer 2 and layer 3
  - Fabric provides a Pervasive SVI which allows for a distributed default gateway
  - Layer 2 and layer 3 traffic is directly forwarded to destination end point

Location Independent Forwarding

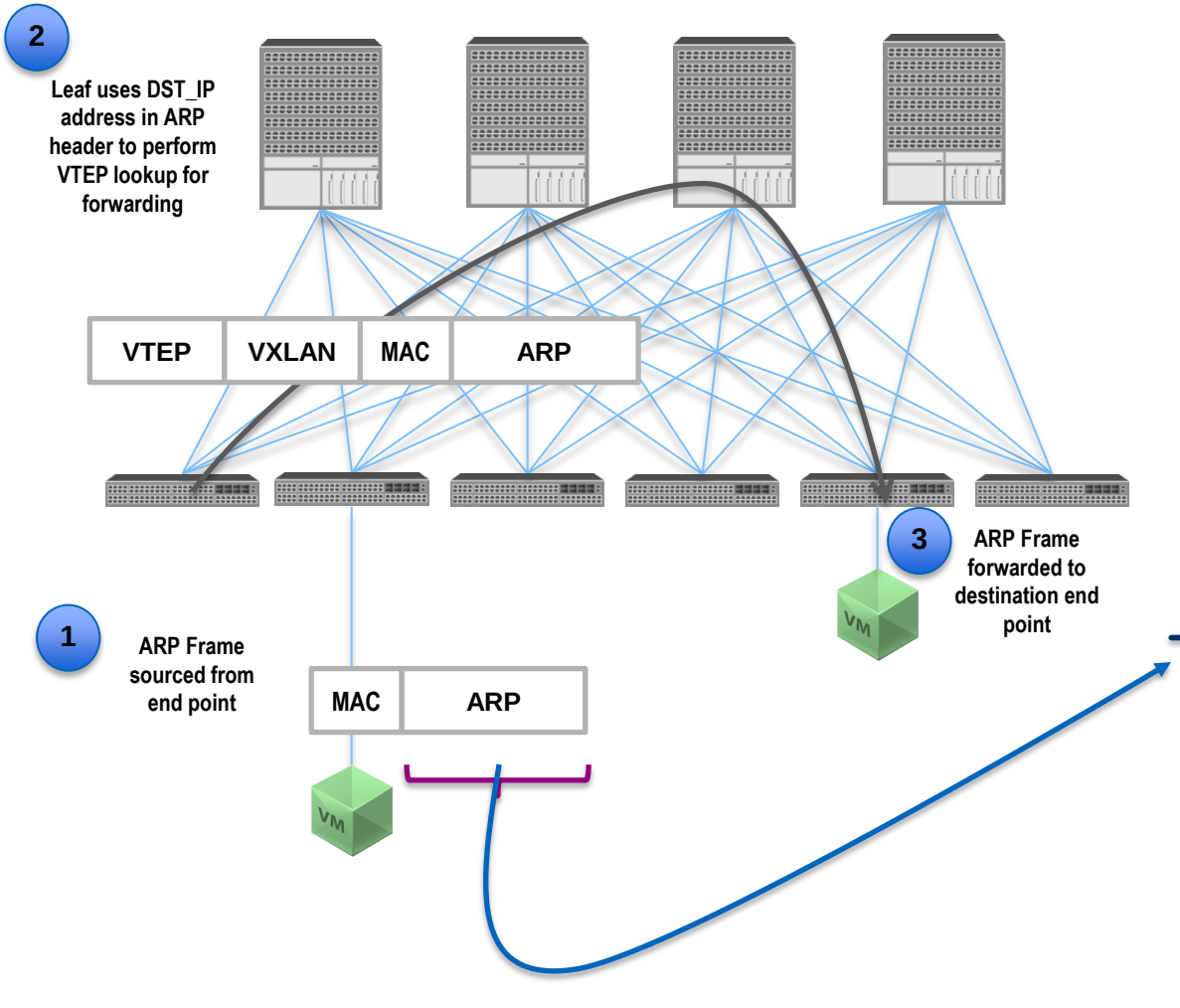
Layer 2 and Layer 3 Handling



- Forward based on destination IP Address for intra and inter subnet (Default Mode)
- Bridge semantics are preserved for intra subnet traffic (no TTL decrement, no MAC header rewrite, etc.)
- Non-IP packets will be forwarded using MAC address. Fabric will learn MACs for non-IP packets, IP address learning for all other packets
- Route if MAC is router-mac, otherwise bridge (standard L2/L3 behavior)

ACI Fabric Scale

Hardware based directed ARP Forwarding

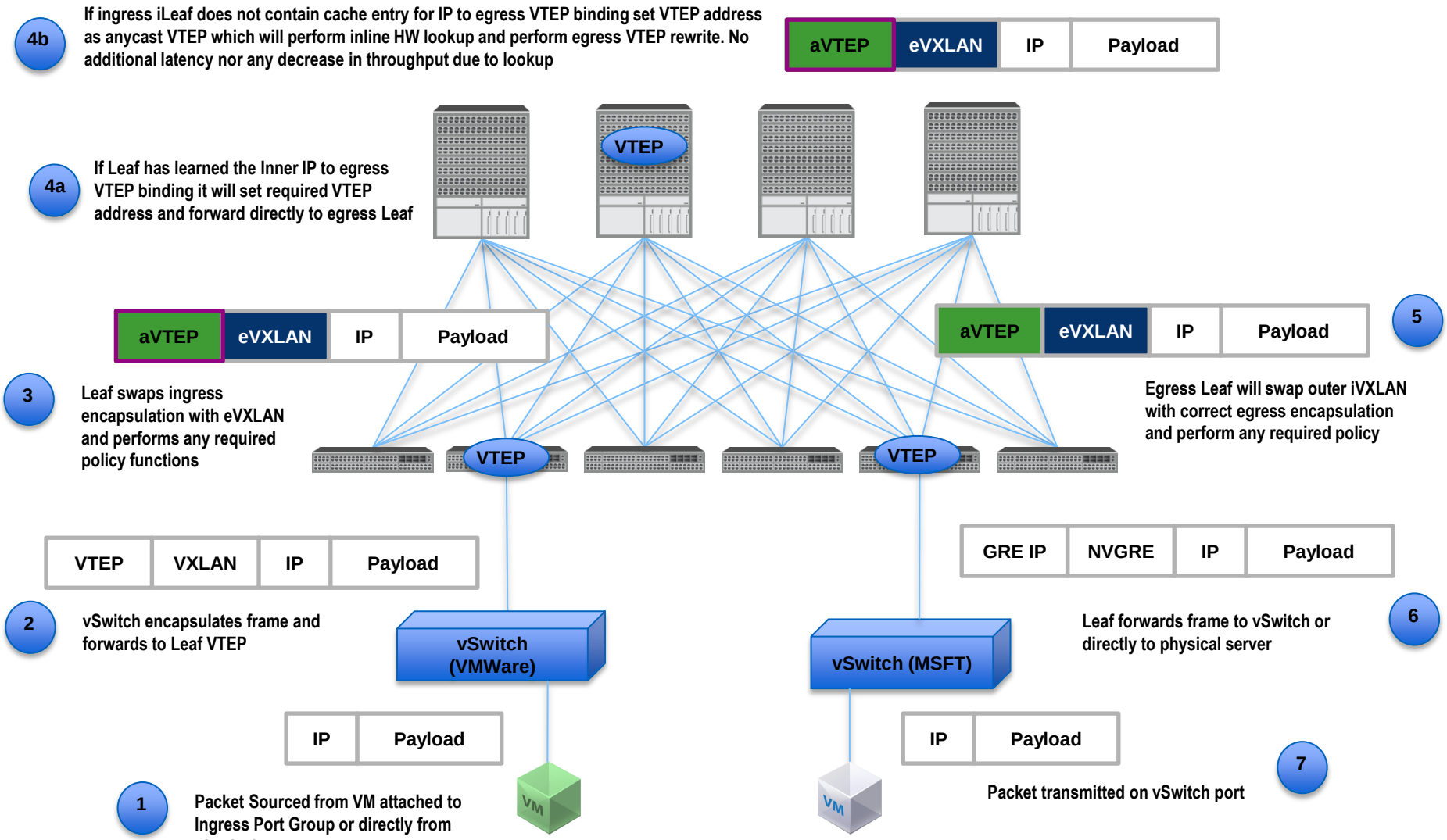


- Leaf ASIC forwards ARP packet to target destination IP identified in ARP payload

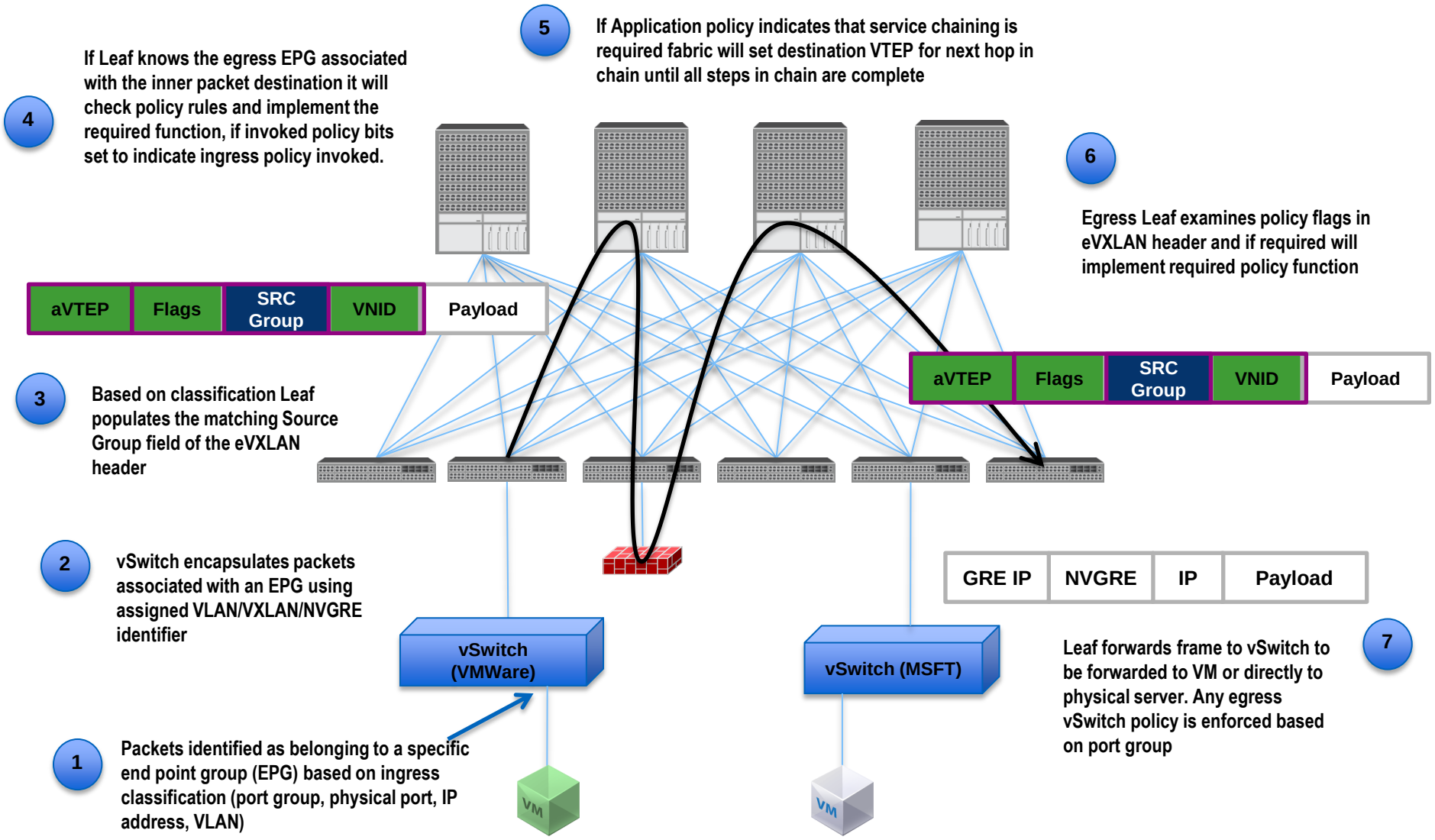
ARP Payload

Hardware type		Protocol type
Hardware size	Protocol size	Operation
Sender MAC address		
Sender MAC address (cont.)		Sender IP address
Sender IP address (cont.)		Destination MAC address
Destination MAC address (cont.)		
Destination IP address		

Overview of ACI Fabric Unicast Forwarding

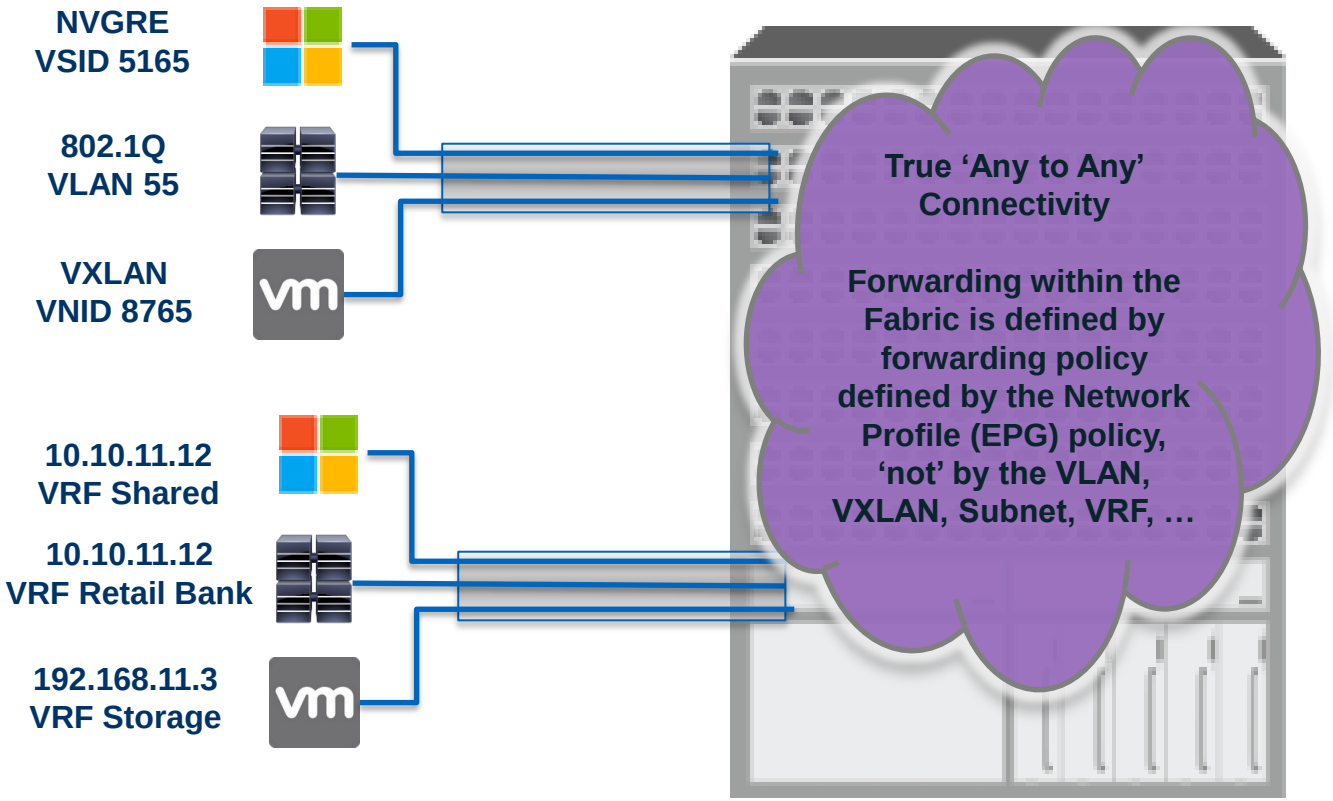


Overview of ACI Fabric Policy Mechanisms

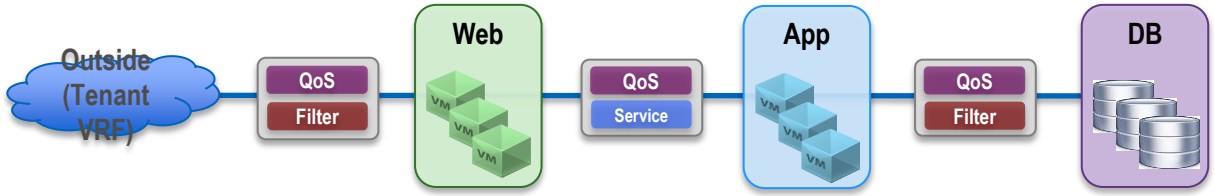


Fabric Infrastructure
Endpoint based forwarding with distributed policy

All single port can support all encapsulations simultaneously



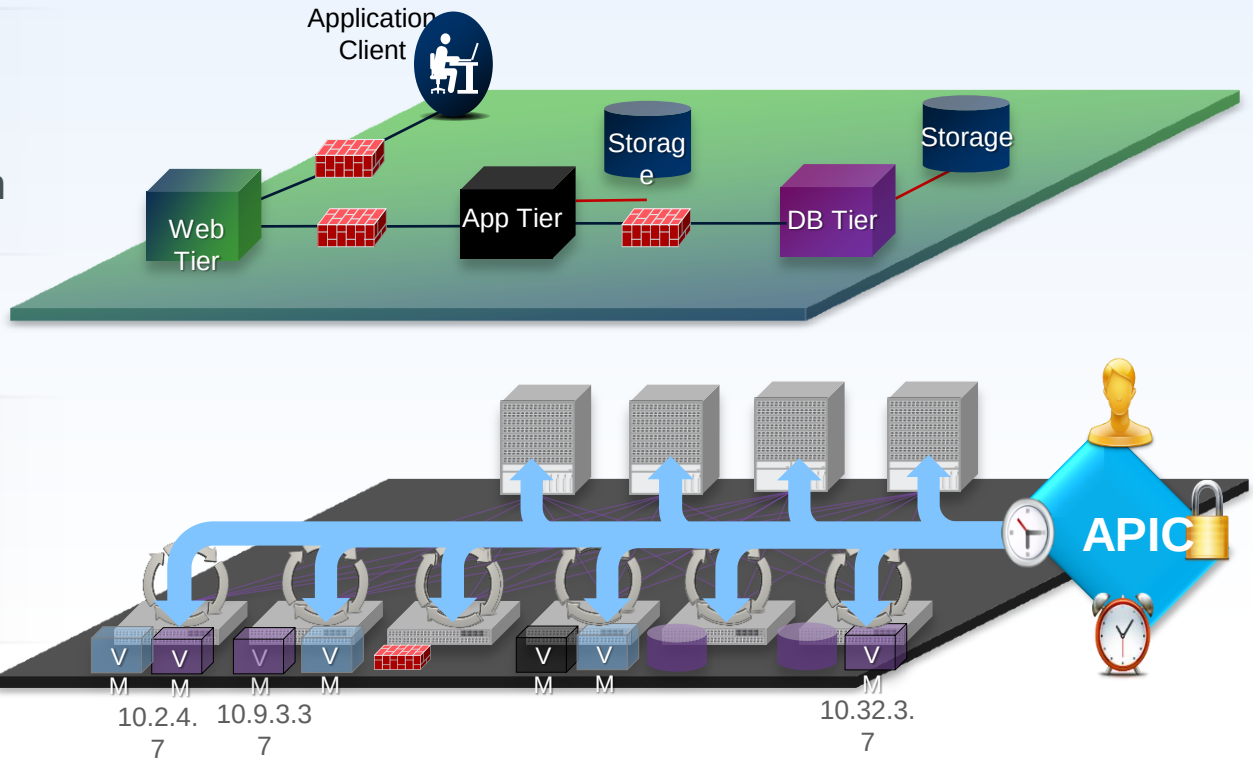
Forwarding is defined by Policy EPG 'Web' can talk to EPG 'DB' independent of IP subnet, VLAN/VXLAN, VRF is Policy says it should



Application Policy Model and Instantiation

Application policy model: Defines the application requirements (application network profile)

Policy instantiation: Each device dynamically instantiates the required changes based on the policies



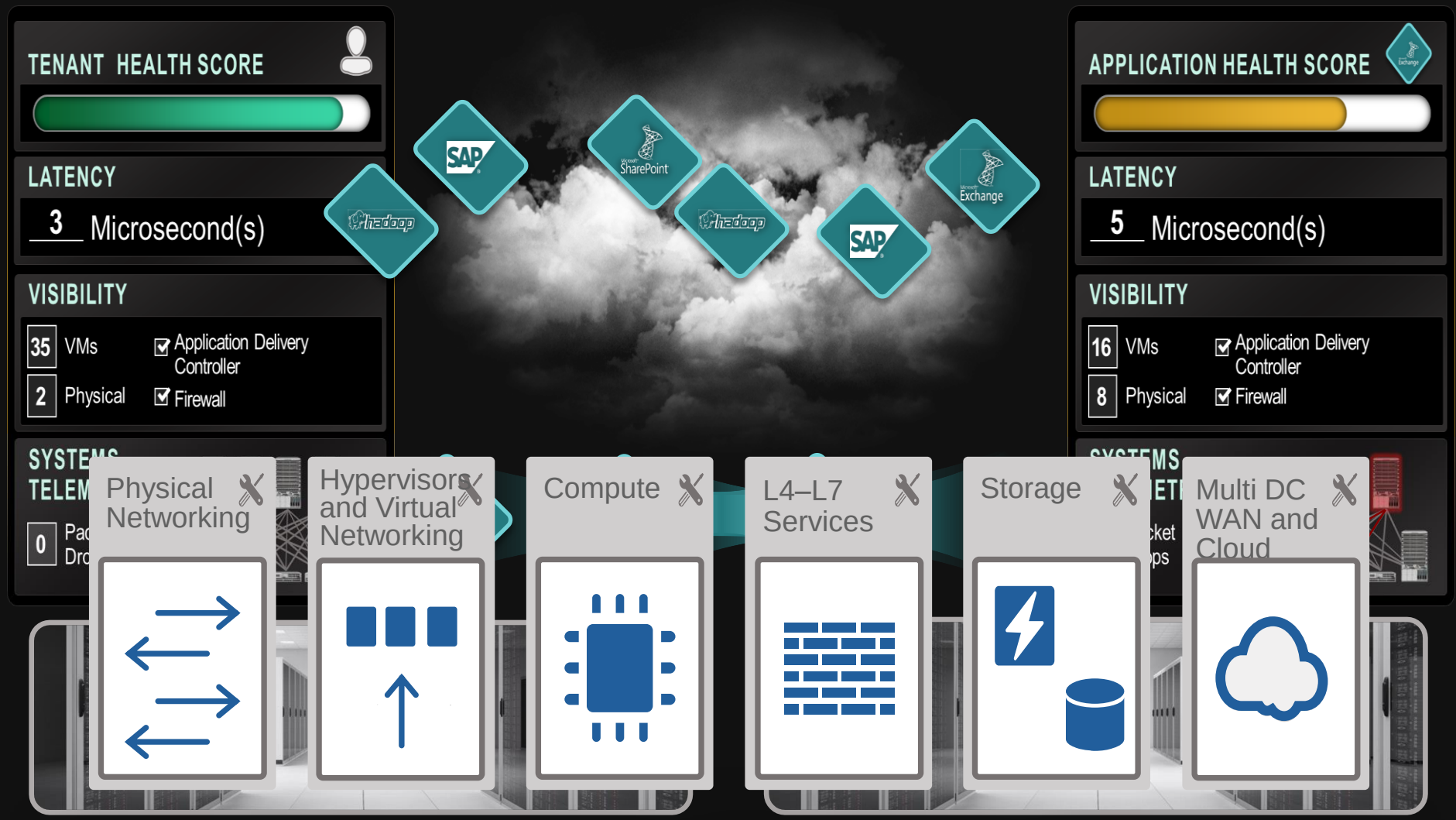
All forwarding in the fabric is managed through the application network profile

- IP addresses are fully portable anywhere within the fabric
- Security and forwarding are fully decoupled from any physical or virtual network attributes
- Devices autonomously update the state of the network based on configured policy requirements

总结

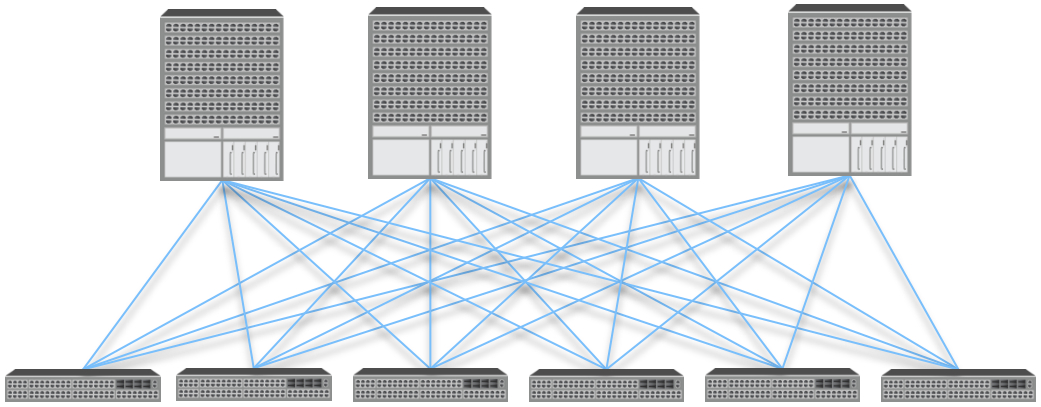
- VXLAN 是Overlay网络的一种部署形式，简单配置即可实现二层网络的扩展，超越 4K VLAN 限制。
- VXLAN 是SDN实现的技术手段之一。
- 必须解决VXLAN 控制平面的问题，才具备大规模部署的条件。
- 通过VXLAN方式实现二层互联的方法，通常只建议部署在DC内部
- VXLAN标准仍然在发展完善，思科可提供VXLAN完整的解决方案

ACI: RAPID DEPLOYMENT OF APPLICATIONS ONTO NETWORKS WITH SCALE, SECURITY AND FULL VISIBILITY



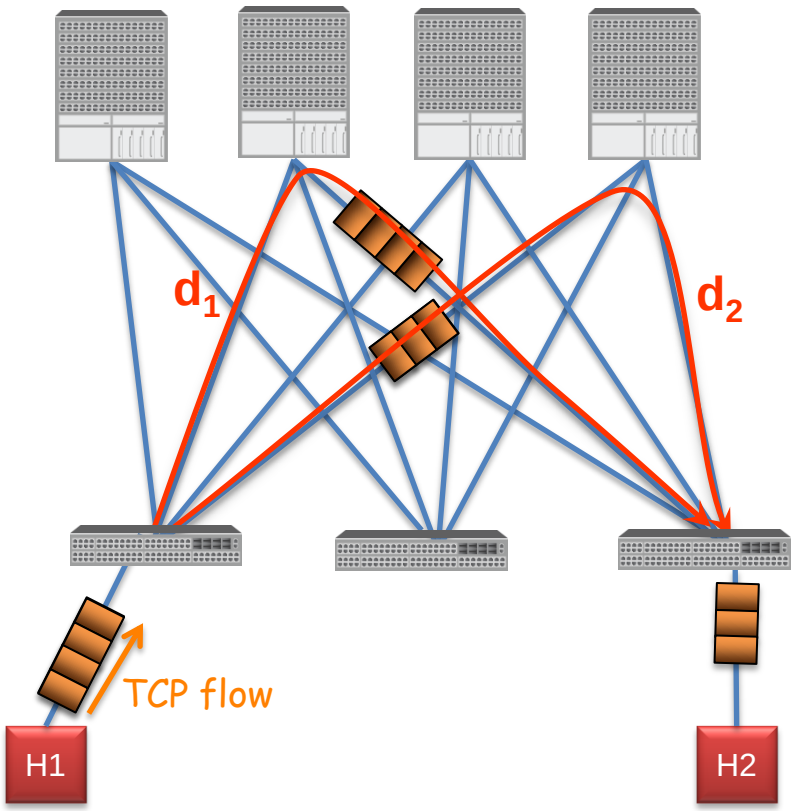
ENABLED BY PHYSICAL AND VIRTUAL INTEGRATION

ACI Fabric
Focus on the application response time

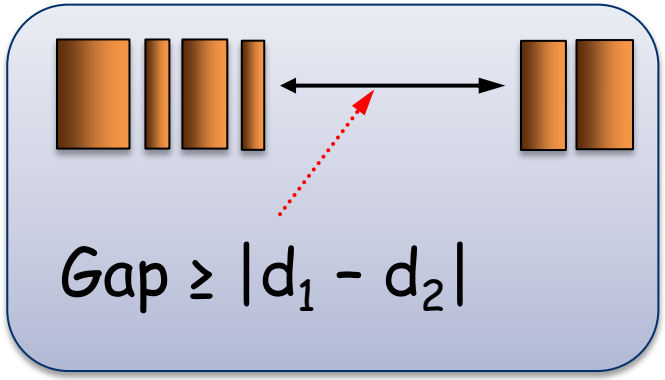


- Building Blocks of ACI Fabric Load Balancing (IFLB)
  - Flowlet Switching
  - End to End congestion monitoring
  - Dynamic Load Balancing
  - Dynamic Flow Prioritization
- Objective is to address fundamental queuing and congestion problems
  - Elephant and Mice problems with flow hashing
  - Random congestion events due to link loss or shifting application traffic patterns

ACI Fabric Load Balancing
Flowlet Switching



- State-of-the-art ECMP hashes flows (5-tuples) to path to prevent reordering TCP packets.
- ***Flowlet switching**** routes bursts of packets from the same flow independently.
- No packet re-ordering

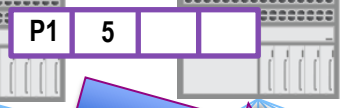


*Flowlet Switching (Kandula et al.'04)

ACI Fabric Load Balancing
Congestion Monitoring

1 Every packet is sourced from an Leaf with an identifier for the ingress port to the Fabric and a DRE = 0 metric

2 At every ASIC hop along the path DRE field is updated with the congestion metric (if my DRE > current DRE then update DRE field else keep current DRE)



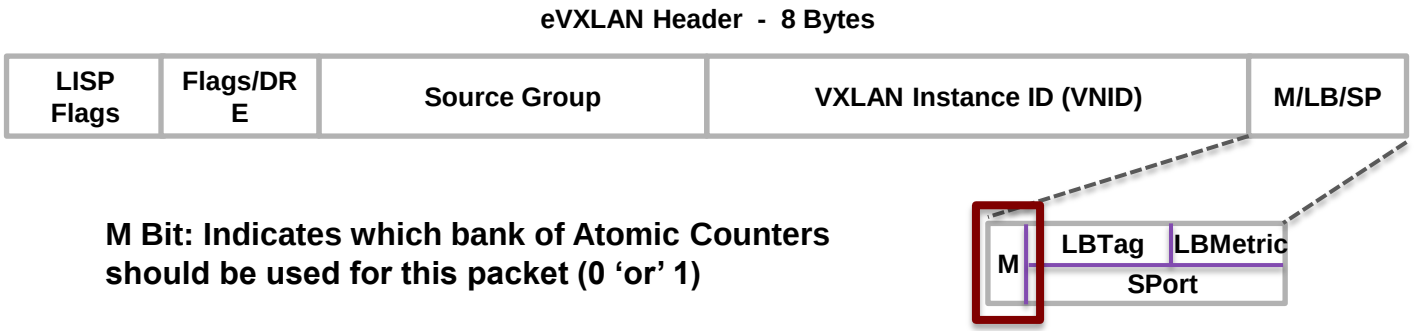
3 On receipt of each packet update the DRE weight in feedback table tracking LBTag (source identifier)

4 Next packet sent back to the originating Leaf will carry with it feedback information on the max load recorded on the ingress path

5

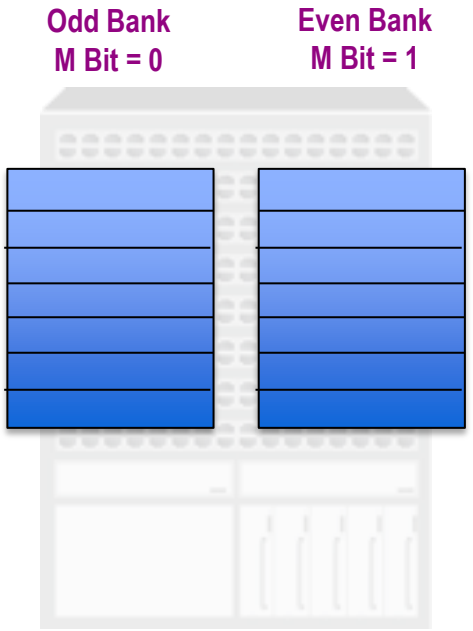
6 On receipt of the metric feedback the original leaf will update the Flowlet load balancing weights (It will also receive indication of the load from the other direction)

Telemetry
Atomic Counters

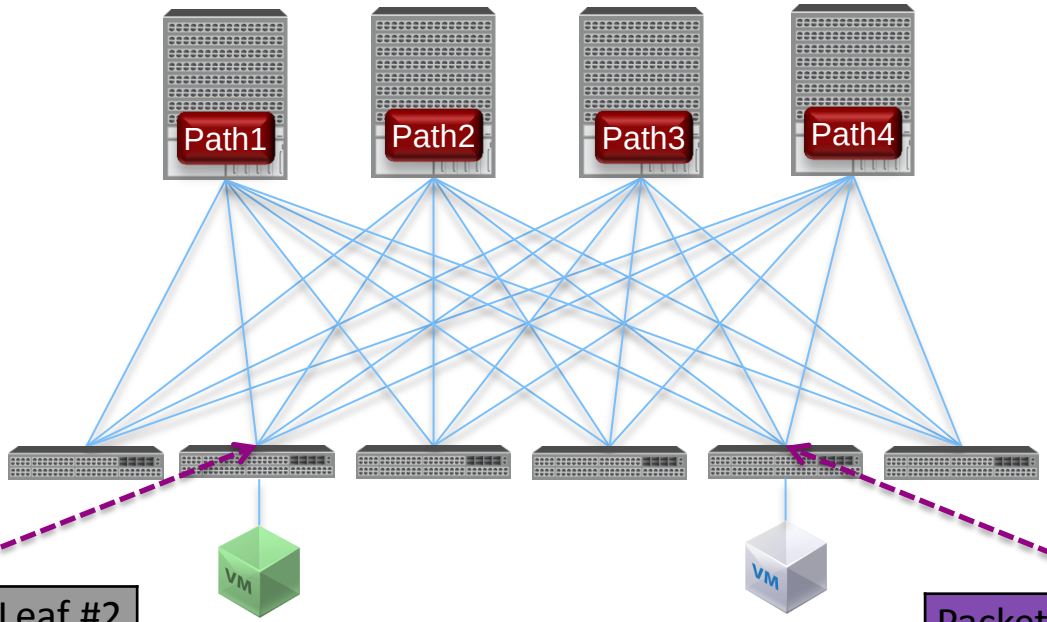


M Bit: Indicates which bank of Atomic Counters should be used for this packet (0 ‘or’ 1)

- Atomic Counters: Increment counters with respect to the ‘packet’ and not with respect to time
  - Atomic Counters counted based on identification of the packet
  - Counters incremented consistently with respect to the ‘packet’
  - Each packet is counted against one of two banks (odd or even)



Telemetry
Atomic Counters



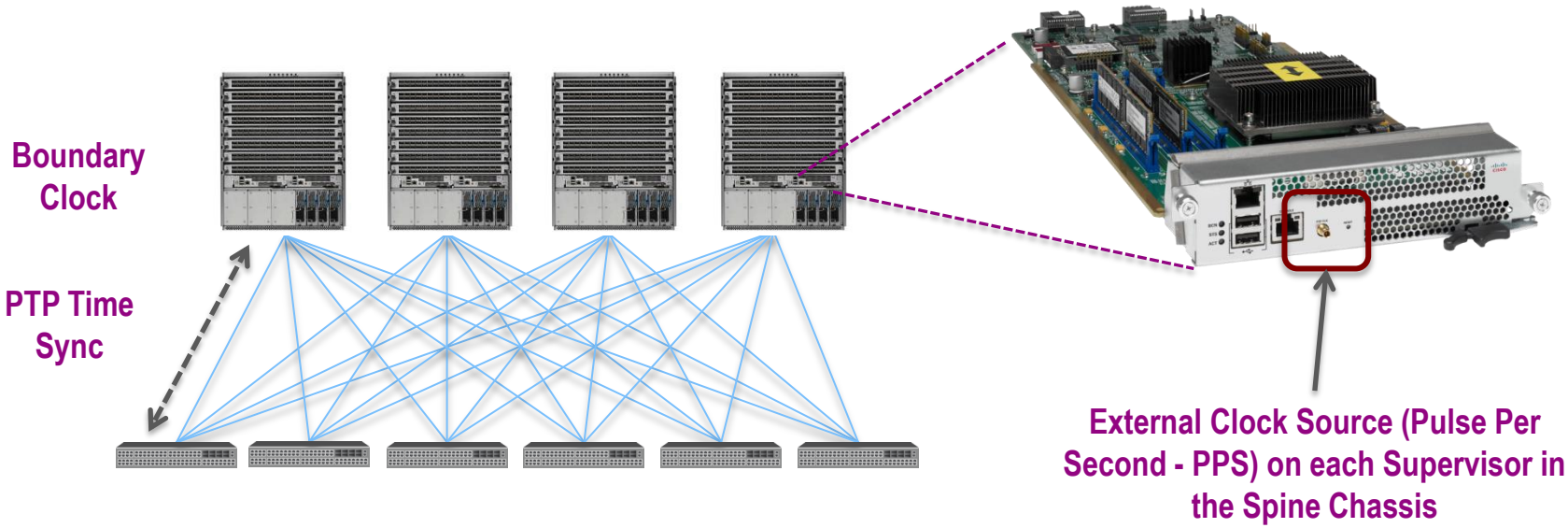
Packets sent from Leaf #2 to Leaf #5	
Path 1	2068
Path 2	2963
Path 3	2866
Path 4	2506

Difference	
Path 1	2
Path 2	0
Path 3	-3
Path 4	0

Packets Received on Leaf #5 sent from Leaf #2	
Path 1	2066
Path 2	2963
Path 3	2869
Path 4	2506

Telemetry
Fabric Latency Measurements

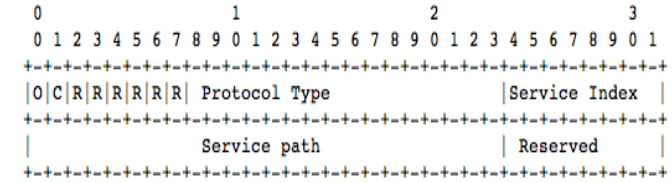
- Matrix of Latency Measurements between all iLeaves is tracked at each Leaf
- IEEE 1588 time sync between nodes in the Fabric
- External Clock Source on the Spine (Spine is Boundary Clock)
- Leaf nodes operate as PTP slaves

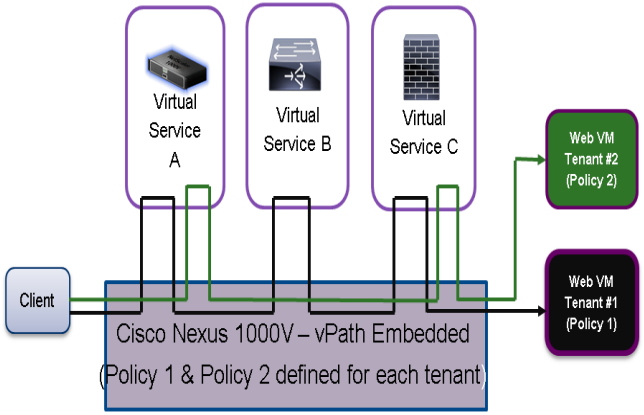


LISP, OTV and VXLAN GPE plus Network Service Header

draft-quinn-nsh-01
standardizes the definition for the vPath 3.0 header

Base Service Header:





Protocol Type = 0xNSH

Protocol Type = IP