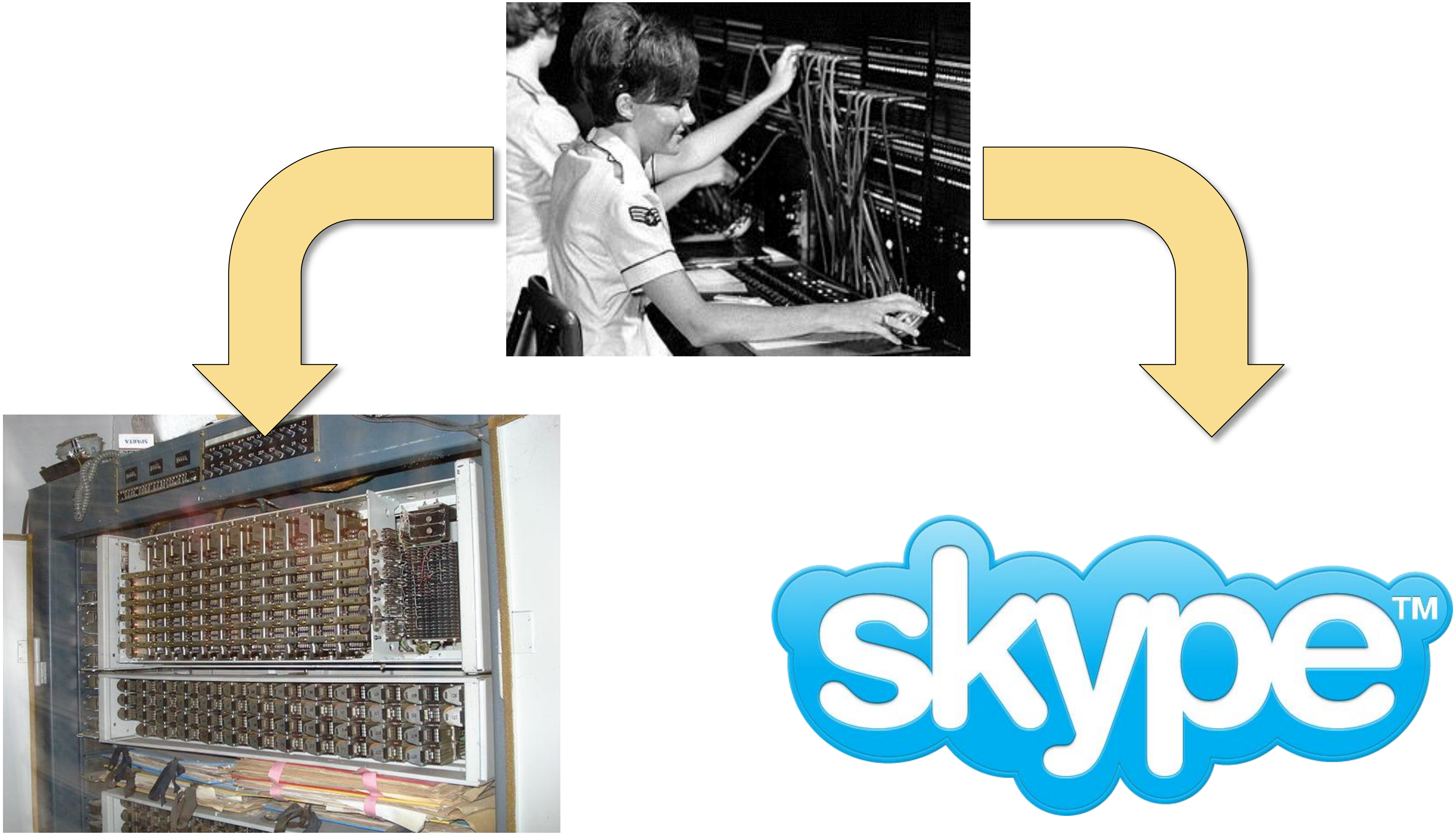
Overlay Virtual Networking Explained

Ivan Pepelnjak (ip@ipSpace.net)
NIL Data Communications

http://commons.wikimedia.org/wiki/File:Telefon_BW_2012-02-18_13-44-32.JPG

http://en.wikipedia.org/wiki/File:Offutt_Air_Force_Base_operator.jpg

Where To?



http://commons.wikimedia.org/wiki/File:1970's_Czechoslovka_TESLA_automatic_telephone_exchange.jpg

Who is Ivan Pepelnjak (@ioshints)

- Networking engineer since 1985
- Technical director, later Chief Technology Advisor @ NIL Data Communications
- Consultant, blogger (blog.ioshints.info), book and webinar author
- Currently teaching “Scalable Web Application Design” at University of Ljubljana

Focus:

- Large-scale data centers and network virtualization
- Networking solutions for cloud computing
- Scalable application design
- Core IP routing/MPLS, IPv6, VPN

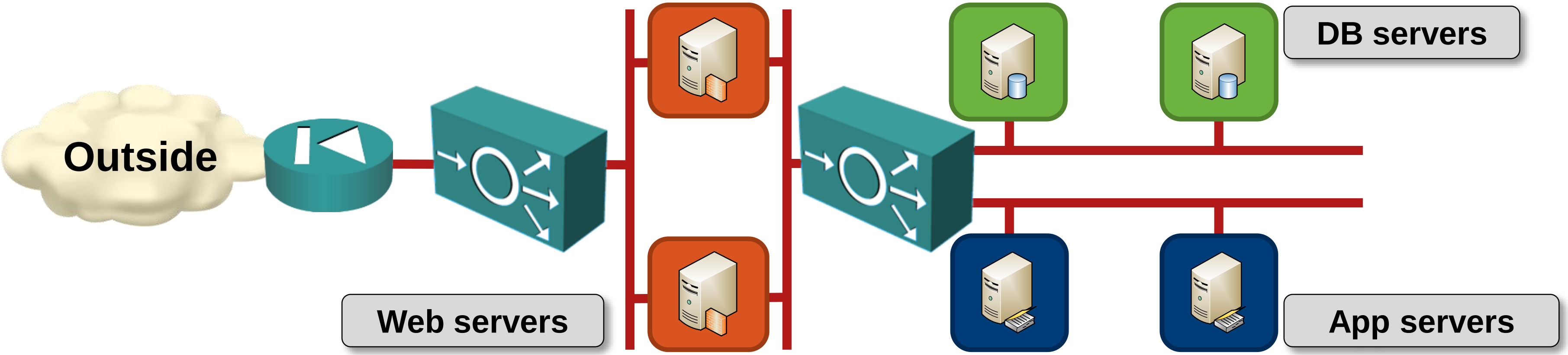
Disclaimers

- This presentation is an analysis of currently available virtual networking architectures
- It’s not an endorsement or bashing of companies, solutions or products mentioned on the following slides
- It describes features shipping in September 2013, not futures announced by individual vendors
- The crucial question: Does It Scale?

Your mission, should you choose to accept it, is to build stable and scalable cloud infrastructure supporting thousands of virtual networks

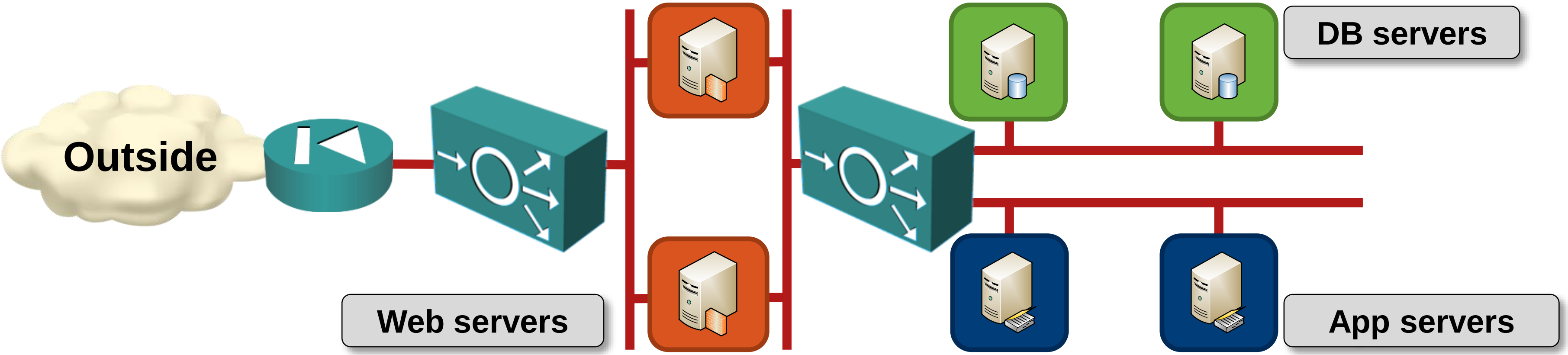
Why Do We Care?

Complex Application Stacks Need Network Services



- L2/L3 packet forwarding with multiple address spaces
- Firewalling (inter-subnet and VM-level)
- Load balancing
- NAT
- VPN access (public cloud deployments)

Cloud Services Must Support Multi Tenancy



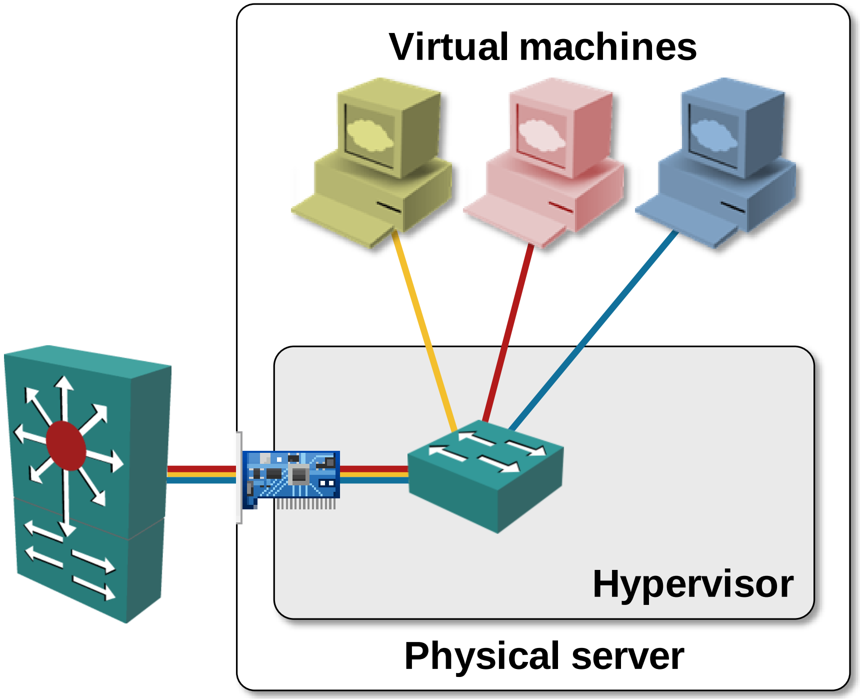
Each application stack deployed in a cloud must be independent

- Retain existing connectivity model (internal addressing, network services and security model)
- Isolated virtual segments and network services
- Per-tenant QoS limits (prevent noisy neighbor problems)

Ideal case: every application is an independent tenant

The Traditional Approach

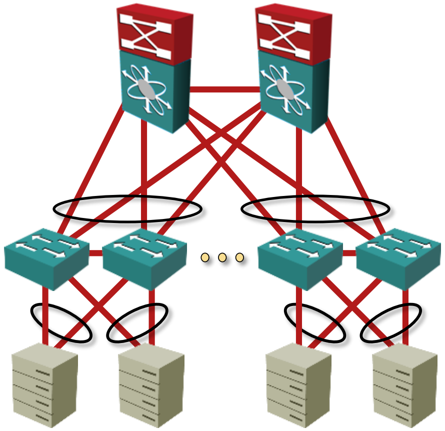
The Traditional Solution: VLANs



- Virtual segments implemented with VLANs
- Layer-2 connectivity also required for VM mobility
- Manual VLAN provisioning or every-VLAN-on-every-port approach

Warning: Layer-2 network = single failure domain

VLANs: Scalability Constraints



Number of VLANs: 4K

VM MAC addresses usually visible in the core

Hypervisor NICs work in promiscuous mode

- Flooded packets handled by hypervisor CPU

Common design: every VLAN on every server port

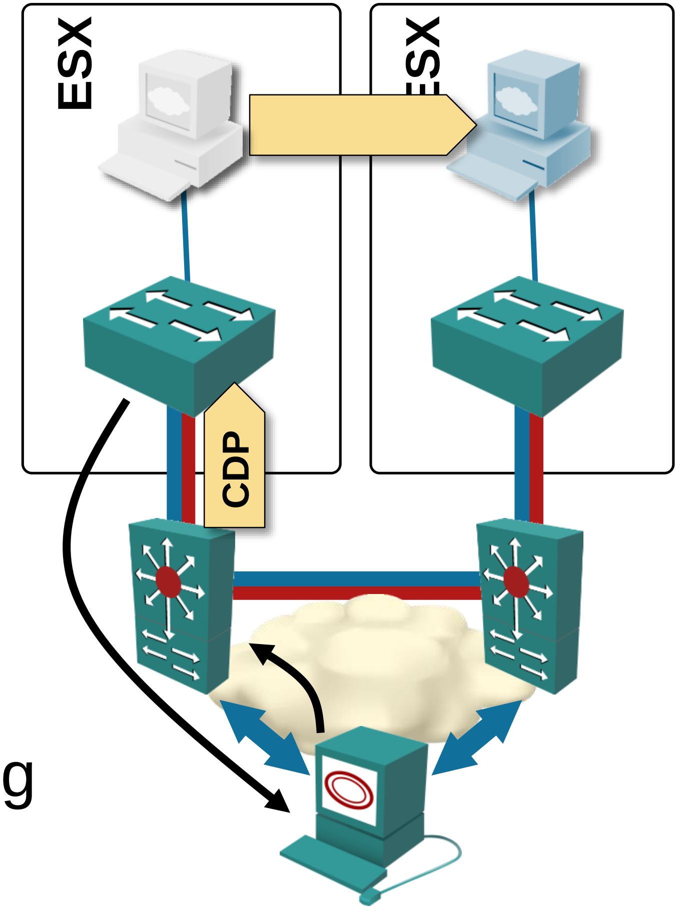
- Every hypervisor processes multicasts for every VLAN
  Even when it has no active VM in that VLAN
- Identical to single VLAN design from scalability perspective

"… *current broadcast domains can support … around 1,000 end-hosts in a single bridged LAN of 100 bridges*" (RFC 5556 - TRILL)

We never mentioned STP – applies equally well to SPB or TRILL

The Networking Industry's Way

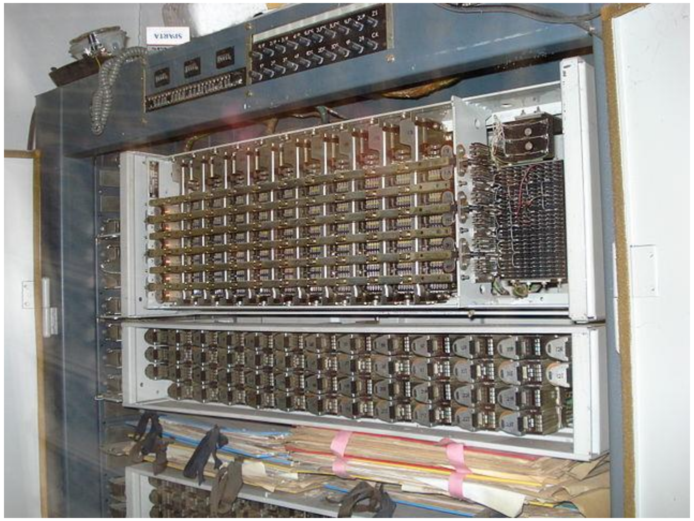
- Hypervisors flooded with broadcasts ➔ VM-aware networking (edge VLAN pruning)
- Spanning Tree problems ➔ Routing protocol (IS-IS) for MAC addresses
- Links blocked by STP ➔ ECMP Brouting (routing at layer-2)
- VLAN limits ➔ Add another VLAN tag
- MAC address limits ➔ Add another MAC header (PBB, TRILL)
- Still too much flooding ➔ core VLAN pruning



Let’s Recap

You need:

- EVB (802.1Qbg) or equivalent (VM tracer, HyperLink, VM-FEX ...)
- TRILL, SPB (802.1Qaq) or equivalent (FabricPath, VCS Fabric, QFabric)
- 802.1ad (Q-in-Q) or 802.1ah (PBB)
- 802.1ak (MVRP) or equivalent (VTP)
- Numerous other features (e.g. BPDU guard, storm control)



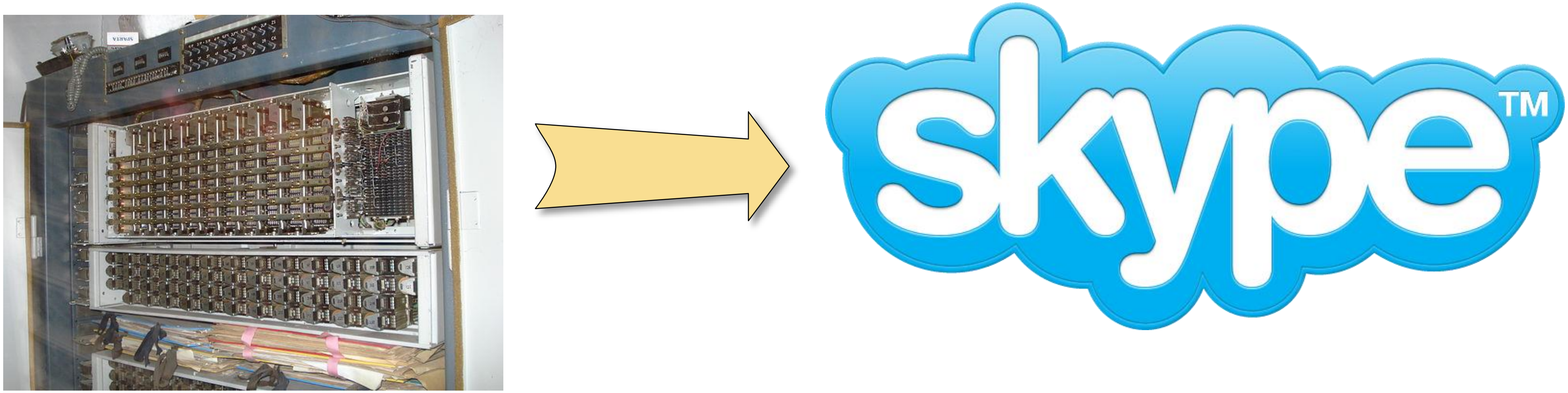
... and you still have a single failure domain

With sufficient thrust, pigs fly just fine **RFC 1925**

Can we afford the fuel costs? And who wants to fly pigs anyway?
Randy Bush

Can We Do Better?

Decouple Virtual Networking From Physical Transport



Principles:

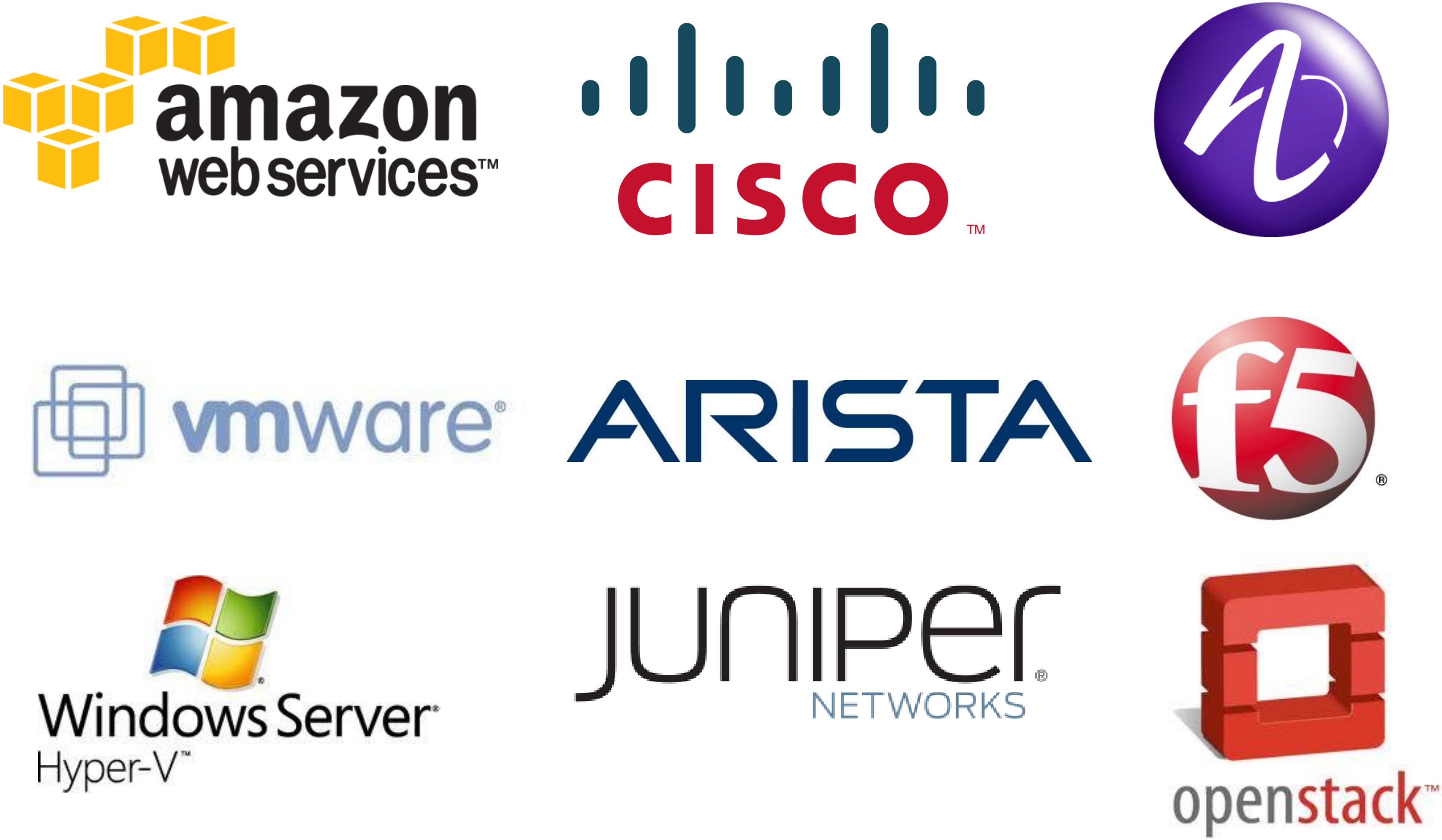
- Network provides simple IP transport
- Complex operations performed in virtual switches
- Virtual network traffic encapsulated in IP datagrams ➔ just another IP application

Decision points:

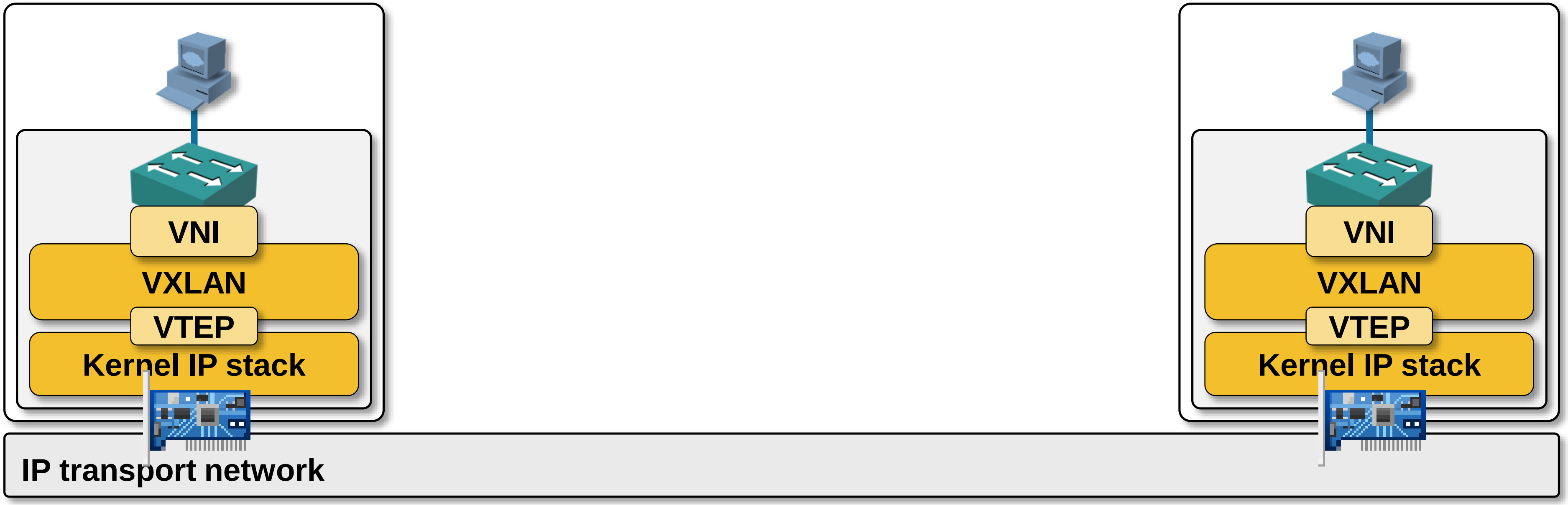
- L2 or L3 hypervisor switching
- Encapsulation: VXLAN, NVGRE, STT
- Control plane or flooding
- Connectivity with the outside world

Smart edge, simple core ... Sounds like Internet, right?

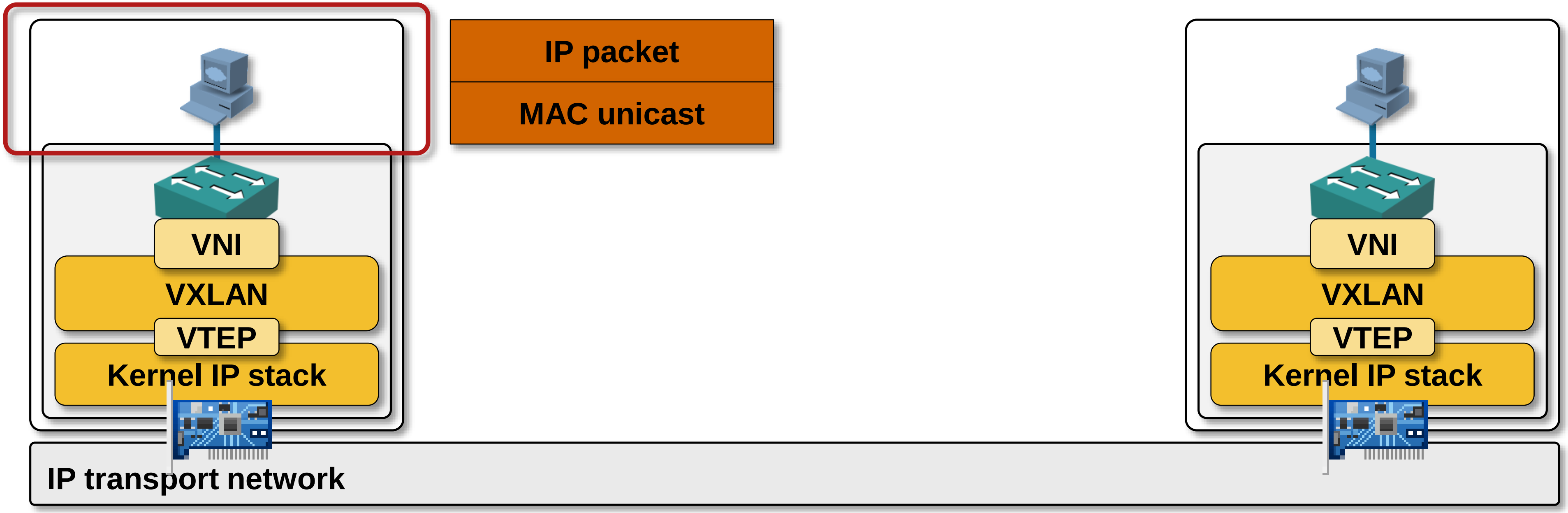
Shipping Products



A Day in the Life of an Overlaid Packet

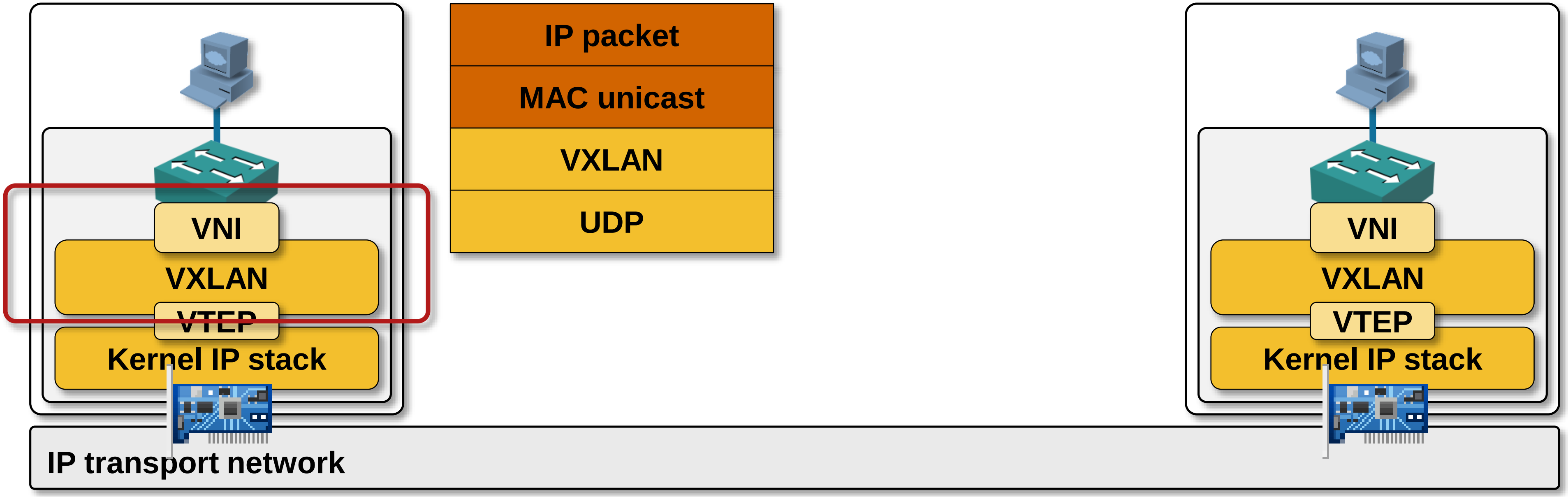


A Day in the Life of an Overlaid Packet



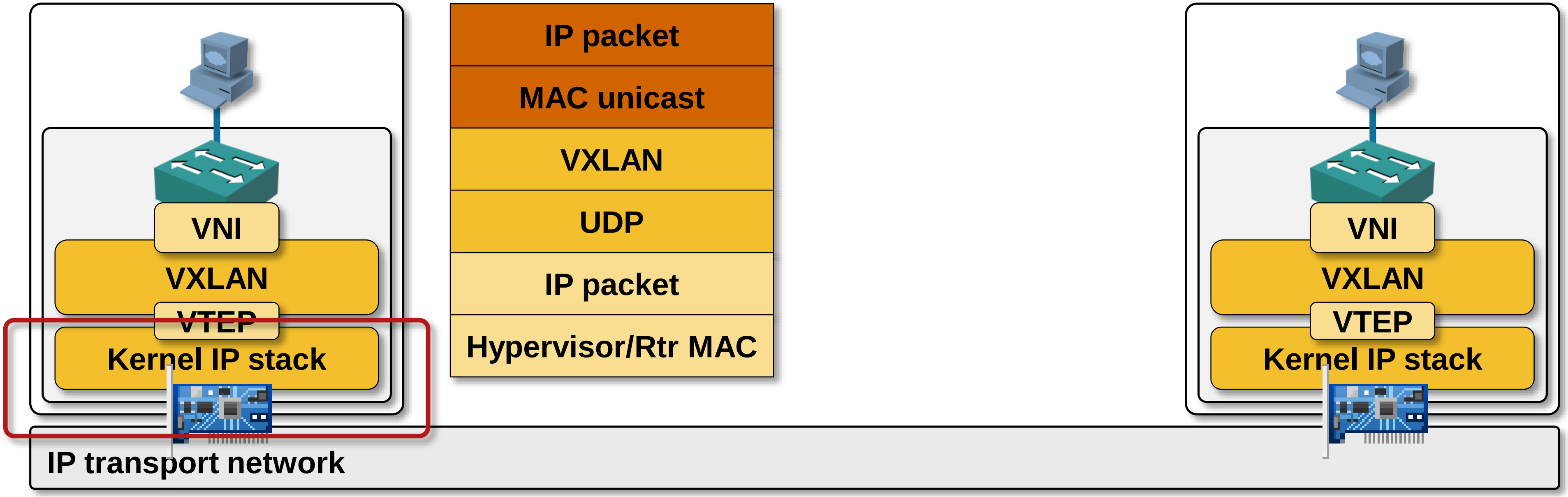
- VM generates Ethernet packet for a MAC destination in the same segment

A Day in the Life of an Overlaid Packet



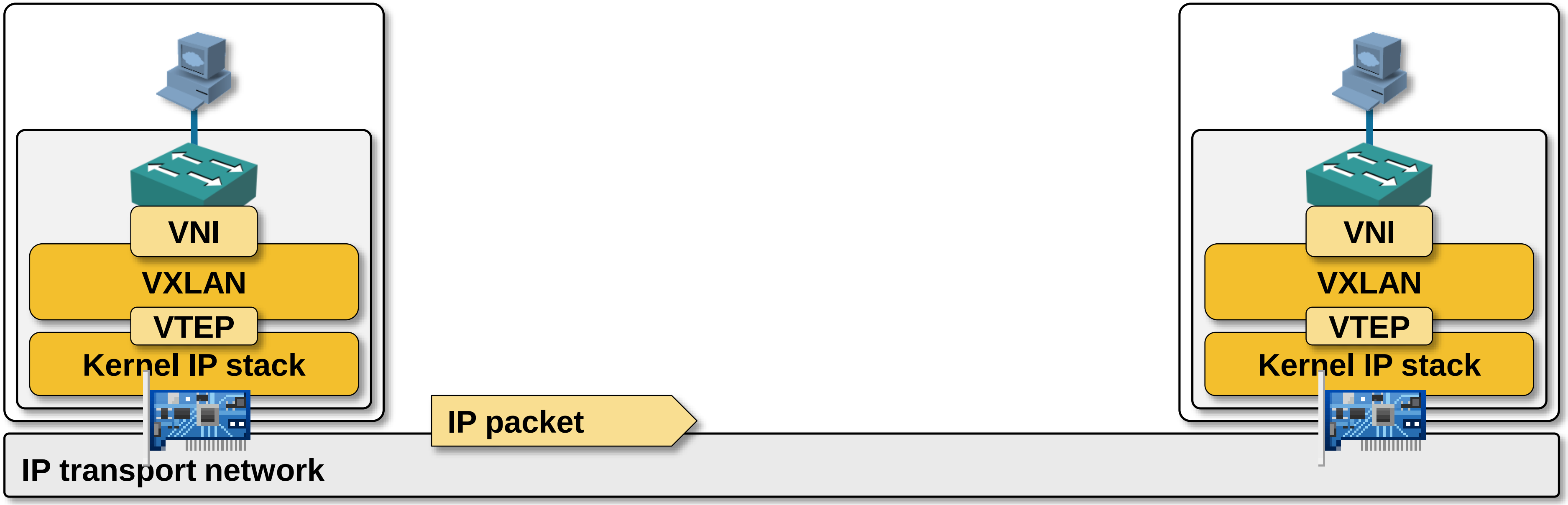
- VM generates Ethernet packet for a MAC destination in the same segment
- VXLAN adds VXLAN/UDP envelope (maps remote MAC to hypervisor IP)

A Day in the Life of an Overlaid Packet



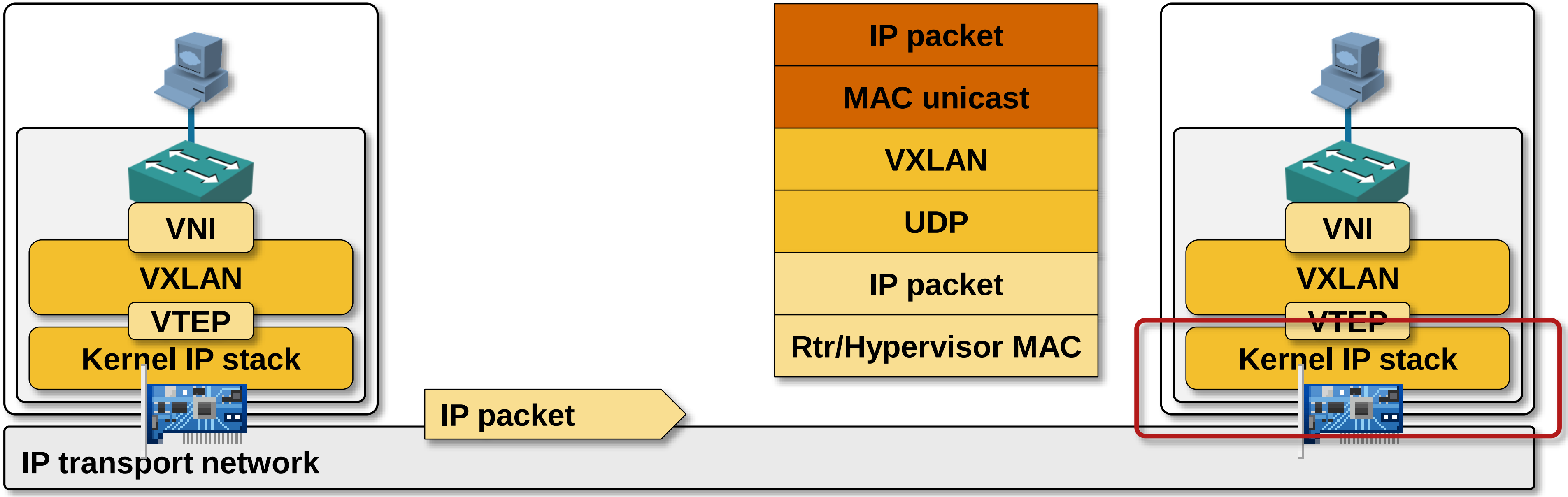
- VM generates Ethernet packet for a MAC destination in the same segment
- VXLAN adds VXLAN/UDP envelope (maps remote MAC to hypervisor IP)
- Kernel IP stack sends IP packet toward target hypervisor

A Day in the Life of an Overlaid Packet



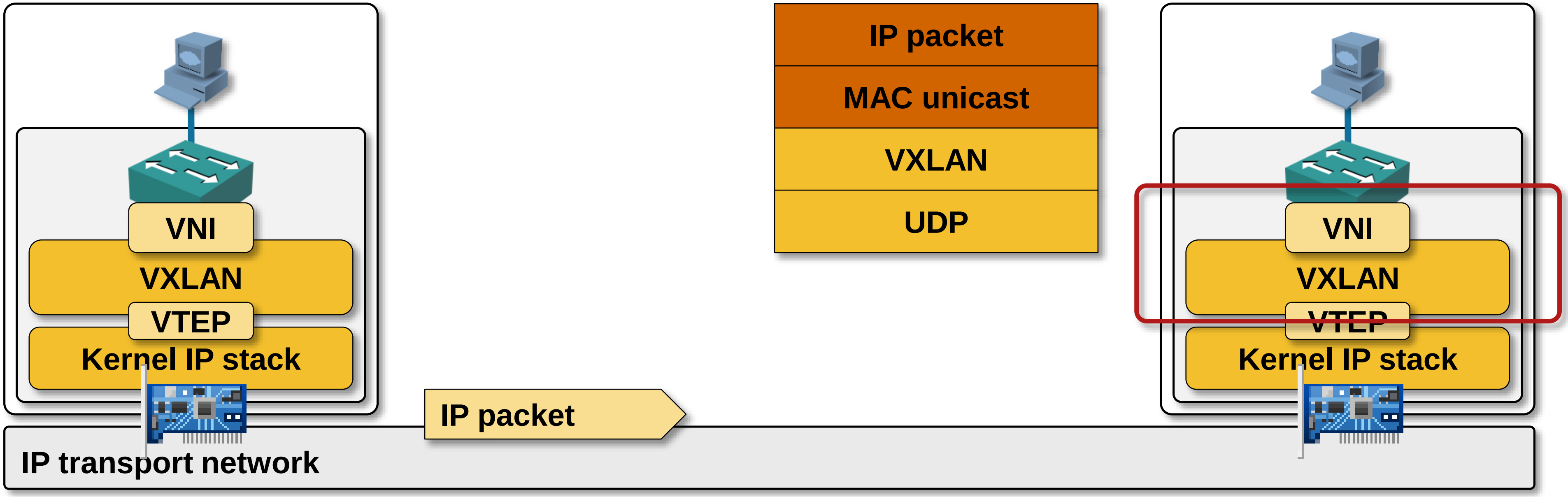
- VM generates Ethernet packet for a MAC destination in the same segment
- VXLAN adds VXLAN/UDP envelope (maps remote MAC to hypervisor IP)
- Kernel IP stack sends IP packet toward target hypervisor

A Day in the Life of an Overlaid Packet



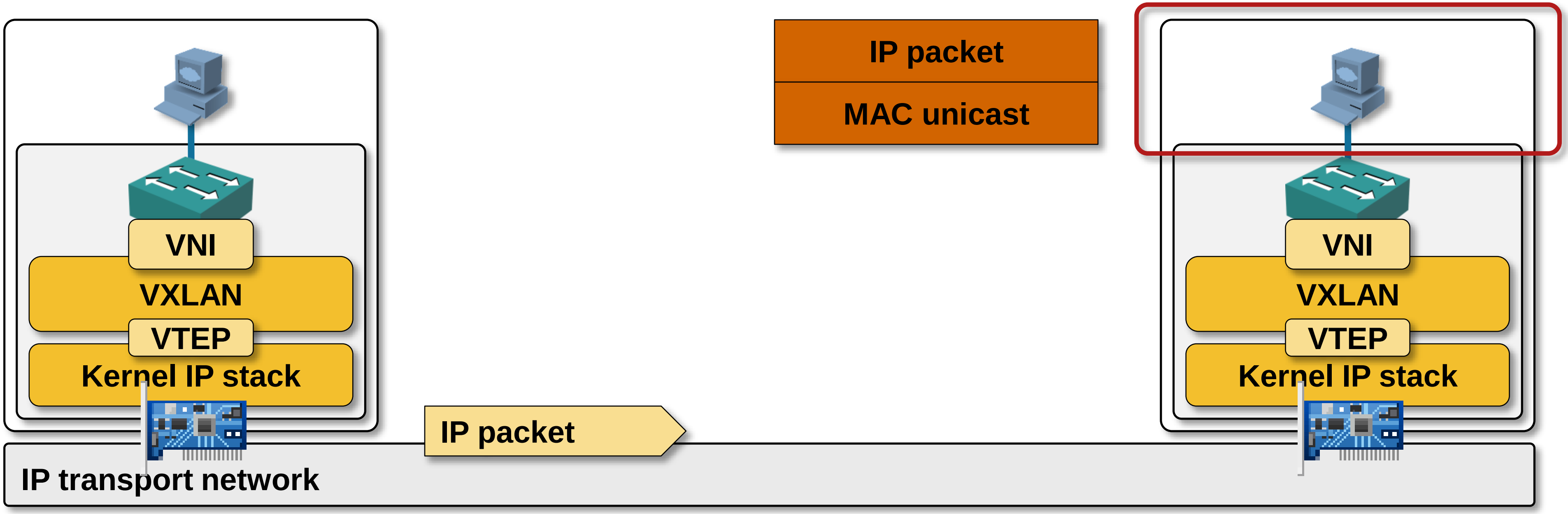
- VM generates Ethernet packet for a MAC destination in the same segment
- VXLAN adds VXLAN/UDP envelope (maps remote MAC to hypervisor IP)
- Kernel IP stack sends IP packet toward target hypervisor
- IP packet received by hypervisor kernel

A Day in the Life of an Overlaid Packet



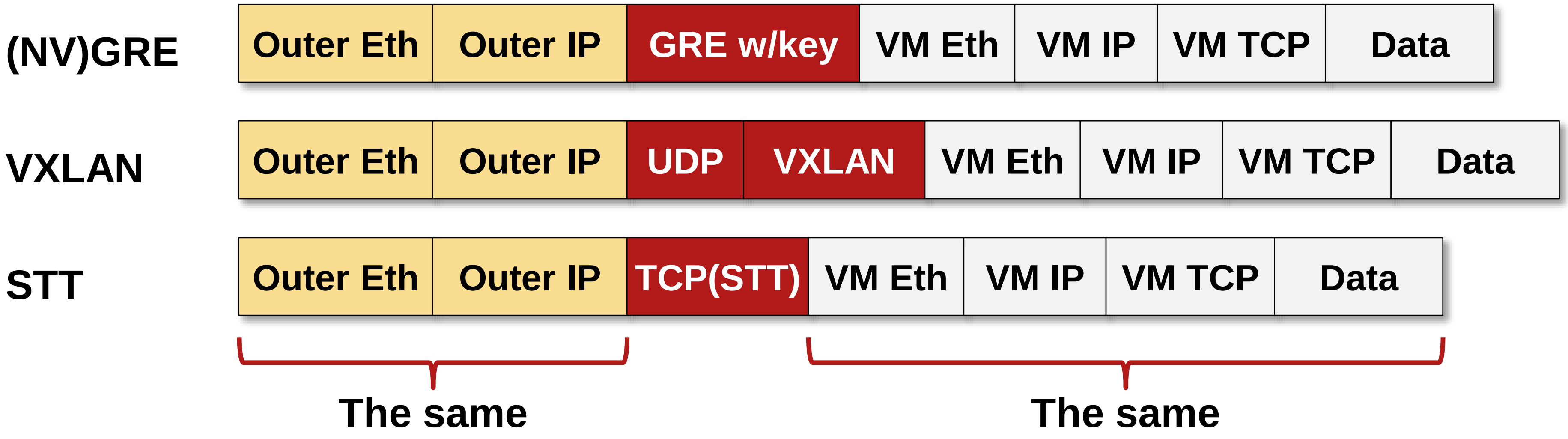
- VM generates Ethernet packet for a MAC destination in the same segment
- VXLAN adds VXLAN/UDP envelope (maps remote MAC to hypervisor IP)
- Kernel IP stack sends IP packet toward target hypervisor
- IP packet received by hypervisor kernel
- UDP packet delivered to VXLAN

A Day in the Life of an Overlaid Packet



- VM generates Ethernet packet for a MAC destination in the same segment
- VXLAN adds VXLAN/UDP envelope (maps remote MAC to hypervisor IP)
- Kernel IP stack sends IP packet toward target hypervisor
- IP packet received by hypervisor kernel
- UDP packet delivered to VXLAN
- Inner Ethernet packet delivered to target VM

The Encapsulation Wars (Are Stupid)



(NV)GRE	Outer Eth	Outer IP	GRE w/key	VM Eth	VM IP	VM TCP	Data	
VXLAN	Outer Eth	Outer IP	UDP	VXLAN	VM Eth	VM IP	VM TCP	Data
STT	Outer Eth	Outer IP	TCP(STT)	VM Eth	VM IP	VM TCP	Data	

- Three competing encapsulations
- Minor technological differences (load balancing, TCP offload)
- None supported by legacy networking hardware or IDS/IPS gear
- No security features ➔ transport network MUST be secure
- What really matters is the control plane

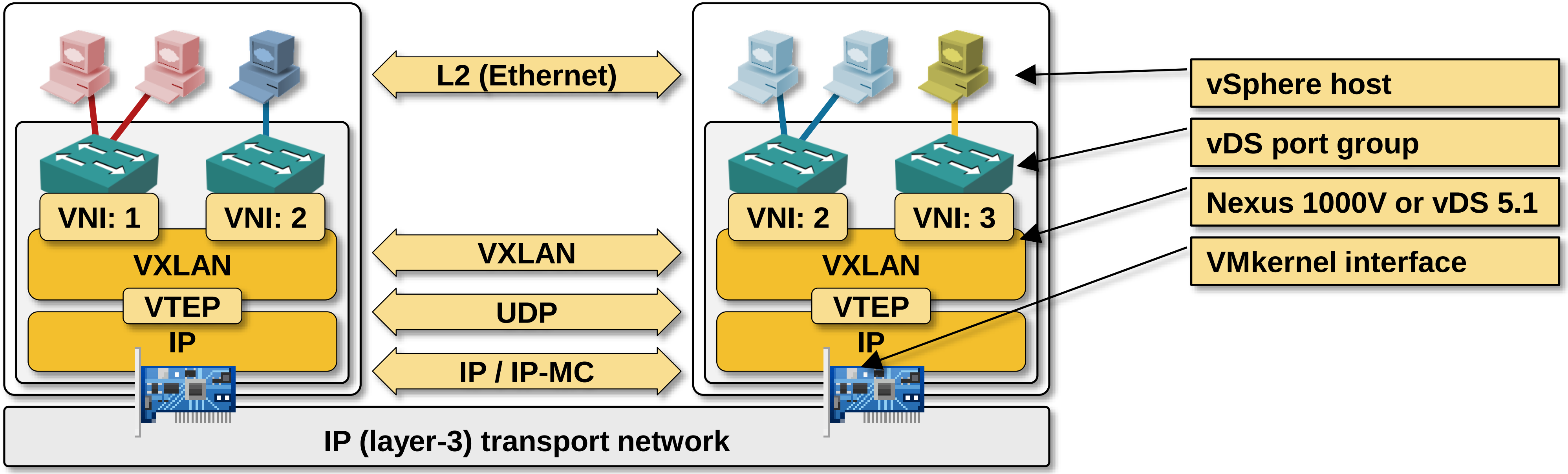
Control Plane Matters Most

What Really Matters in Overlay Virtual Networking

Questions that impact an overlay virtual network scalability:

- How will the source hypervisor find out the IP address of the target hypervisor (MAC-to-VTEP mapping)?
- How much information (and state) must be kept in hypervisors and central controllers or databases?
- Does the forwarding information change in real-time or only on topology change?
- What is a topology change?

VXLAN (Lack Of) Control Plane

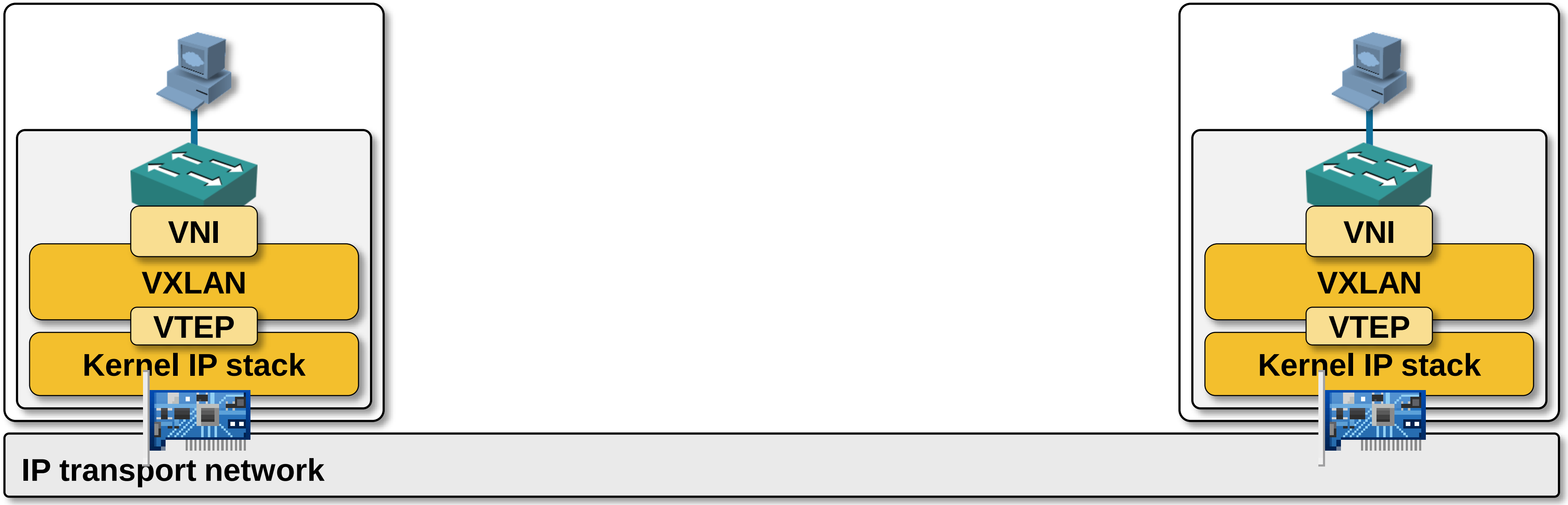


Relies on traditional L2 flooding/learning behavior

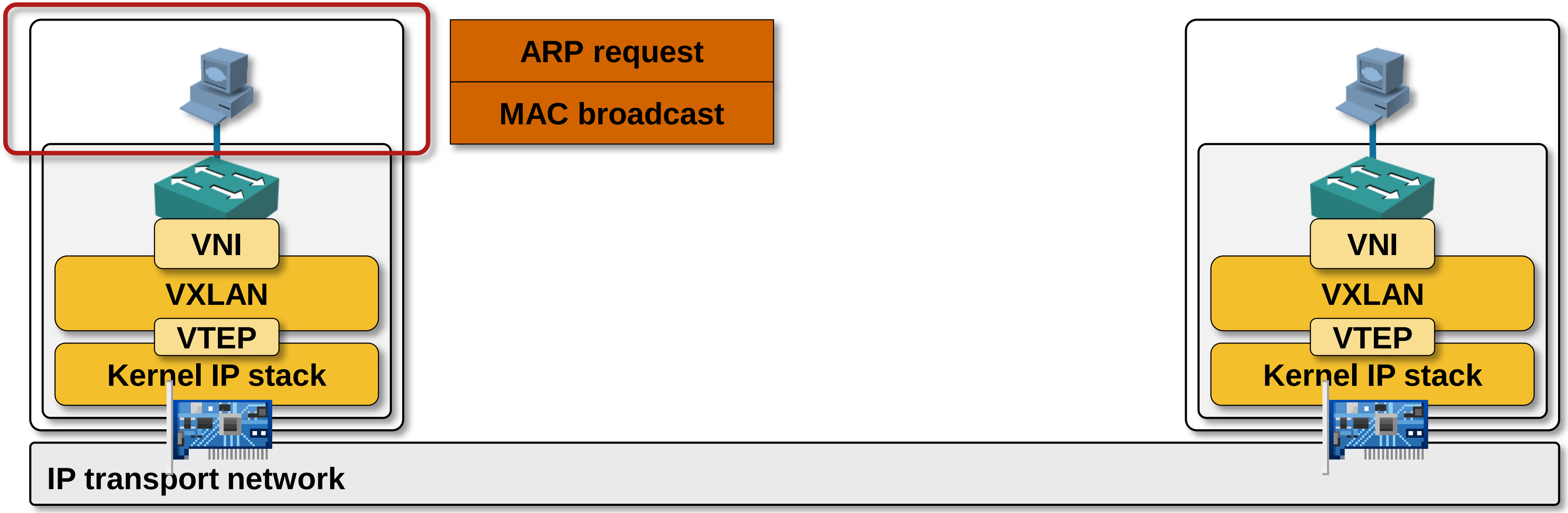
- BUM (Broadcast, Unknown Unicast, Multicast) frames are flooded
- IP multicast in transport network is used to flood VM L2 frames
- Pool of IP multicast addresses or IP multicast address per VXLAN segment
- Hypervisors build VM-MAC-to-host-IP maps by listening to flooded frames

Unicast VXLAN shipping since June 2013

VXLAN Flooding and Learning

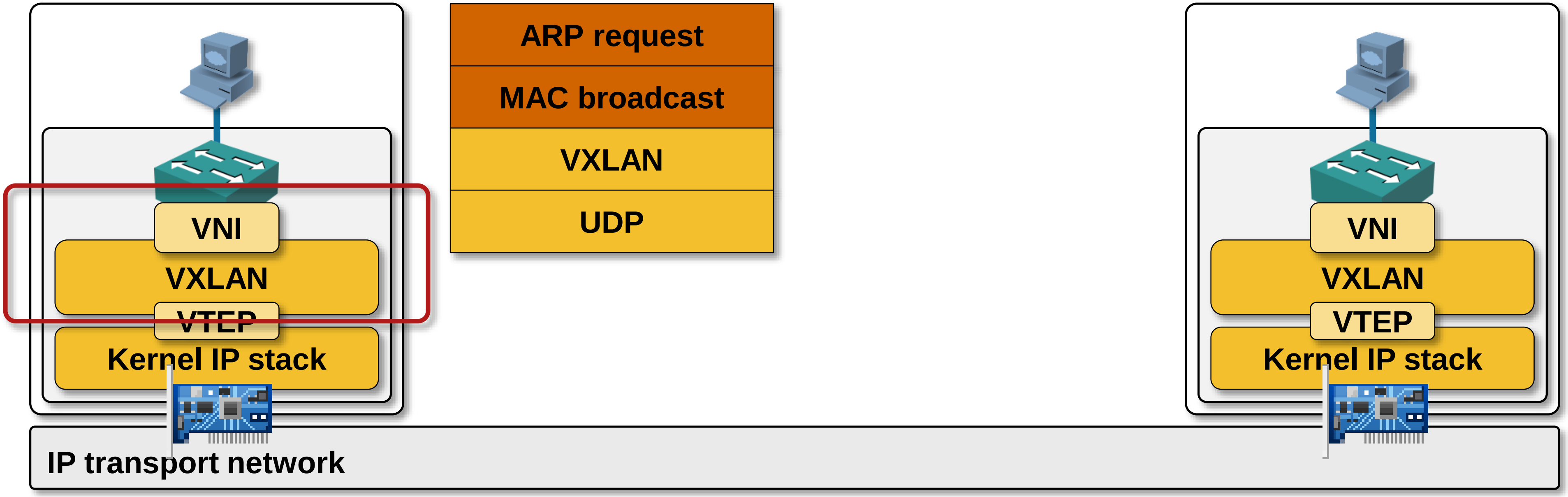


VXLAN Flooding and Learning



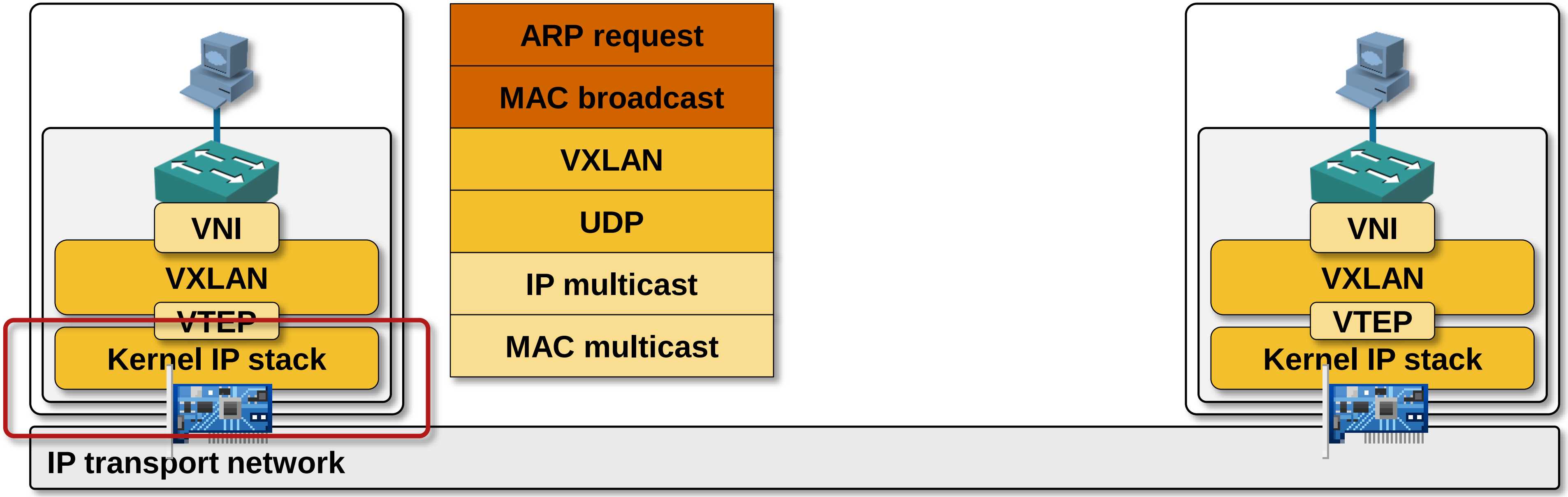
- VM generates ARP broadcast

VXLAN Flooding and Learning



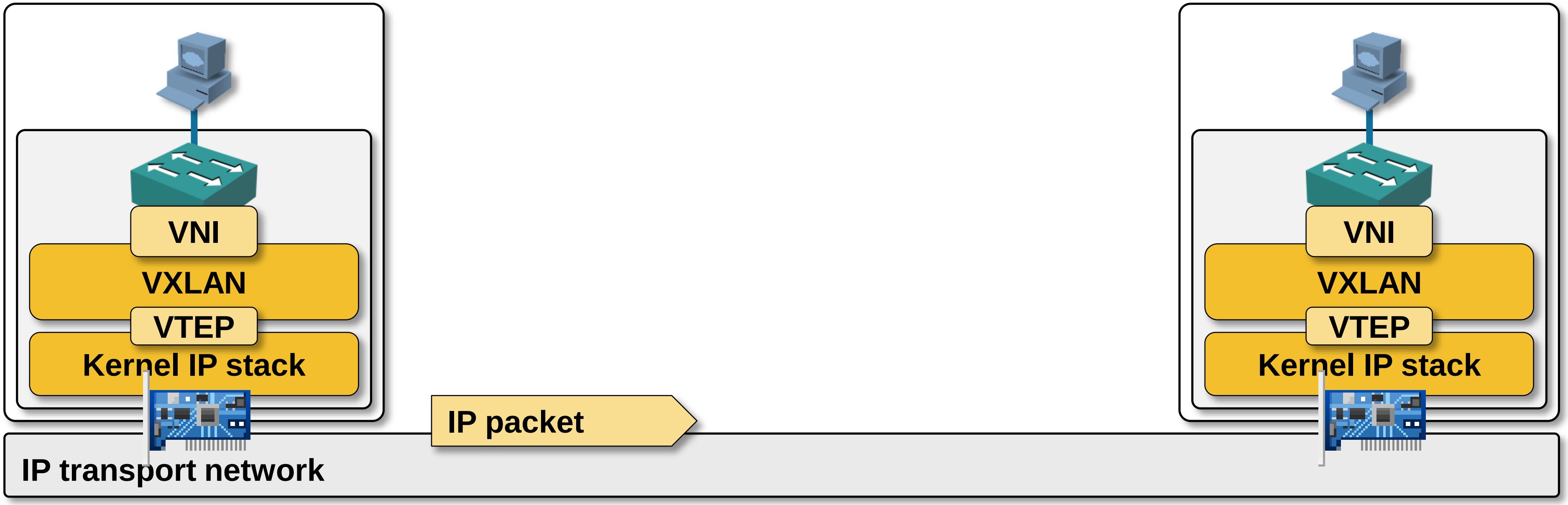
- VM generates ARP broadcast
- VXLAN module sends UDP packet to segment-specific multicast destination

VXLAN Flooding and Learning



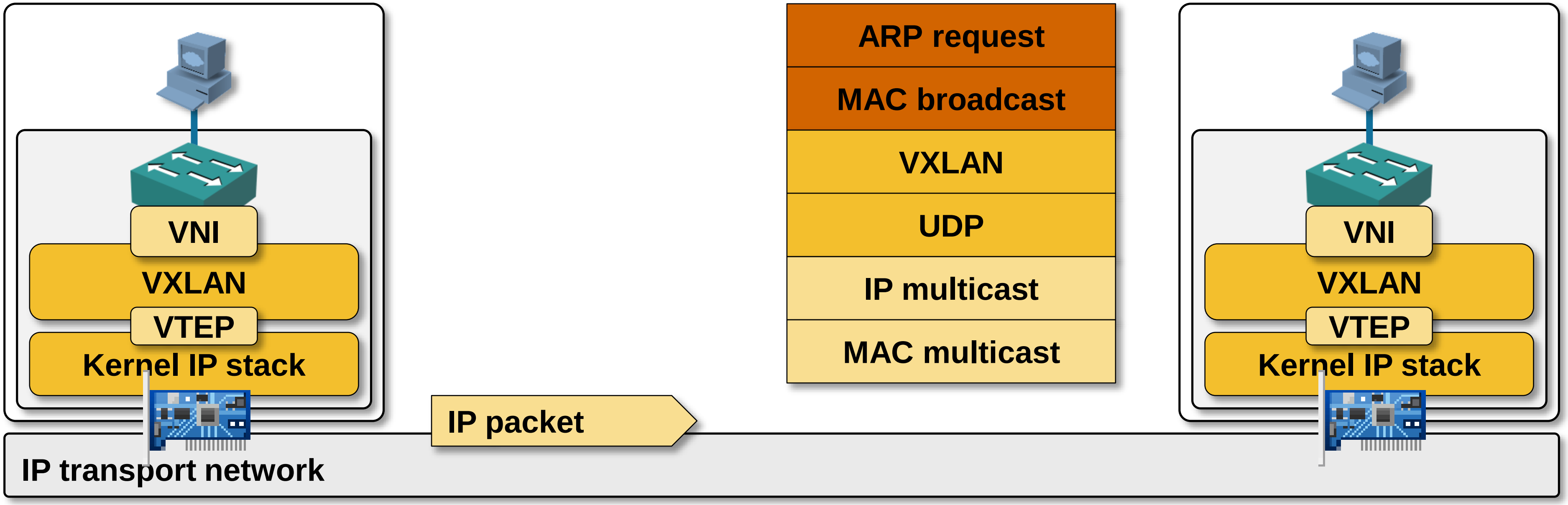
- VM generates ARP broadcast
- VXLAN module sends UDP packet to segment-specific multicast destination
- Kernel IP stack sends multicast IP packet

VXLAN Flooding and Learning



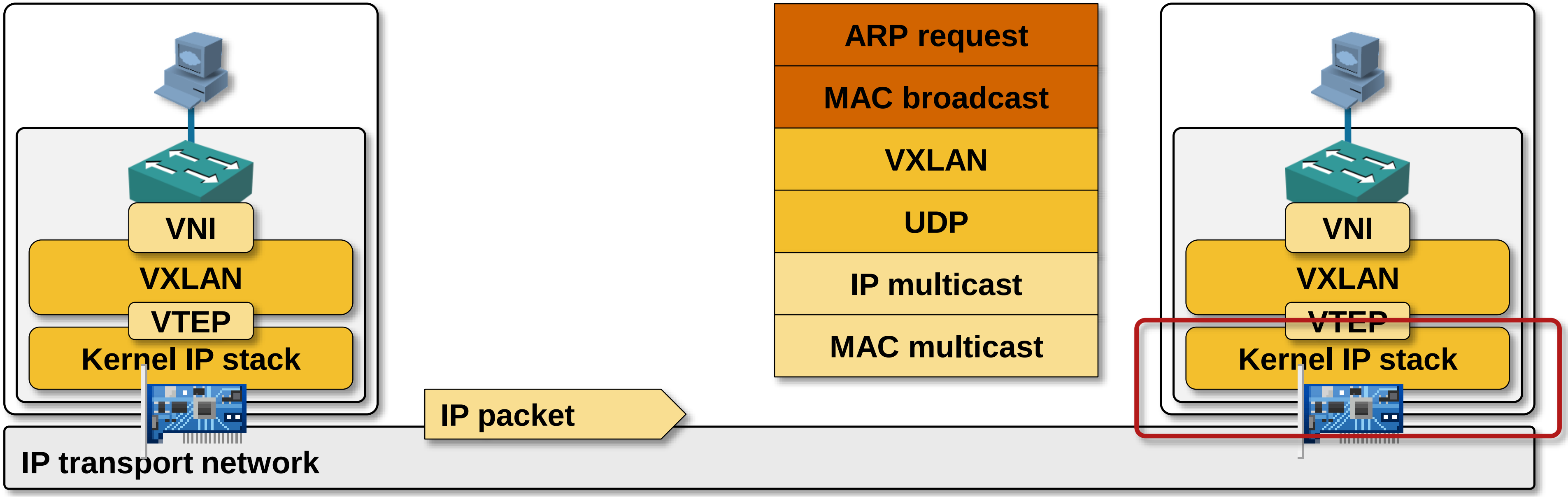
- VM generates ARP broadcast
- VXLAN module sends UDP packet to segment-specific multicast destination
- Kernel IP stack sends multicast IP packet

VXLAN Flooding and Learning



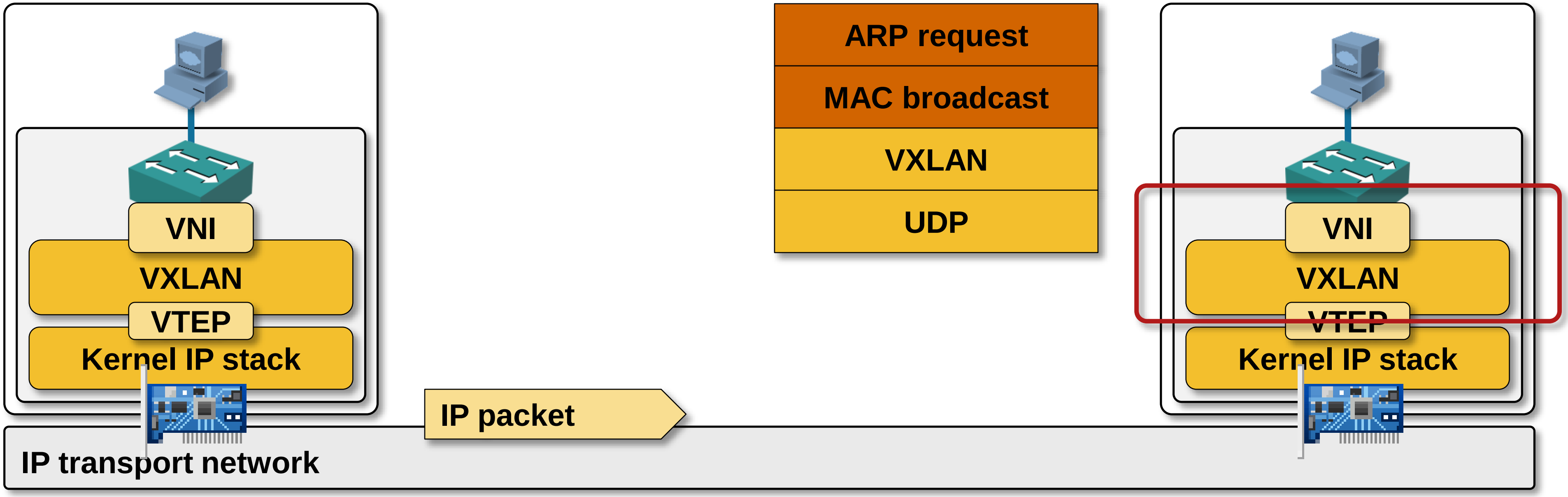
- VM generates ARP broadcast
- VXLAN module sends UDP packet to segment-specific multicast destination
- Kernel IP stack sends multicast IP packet

VXLAN Flooding and Learning



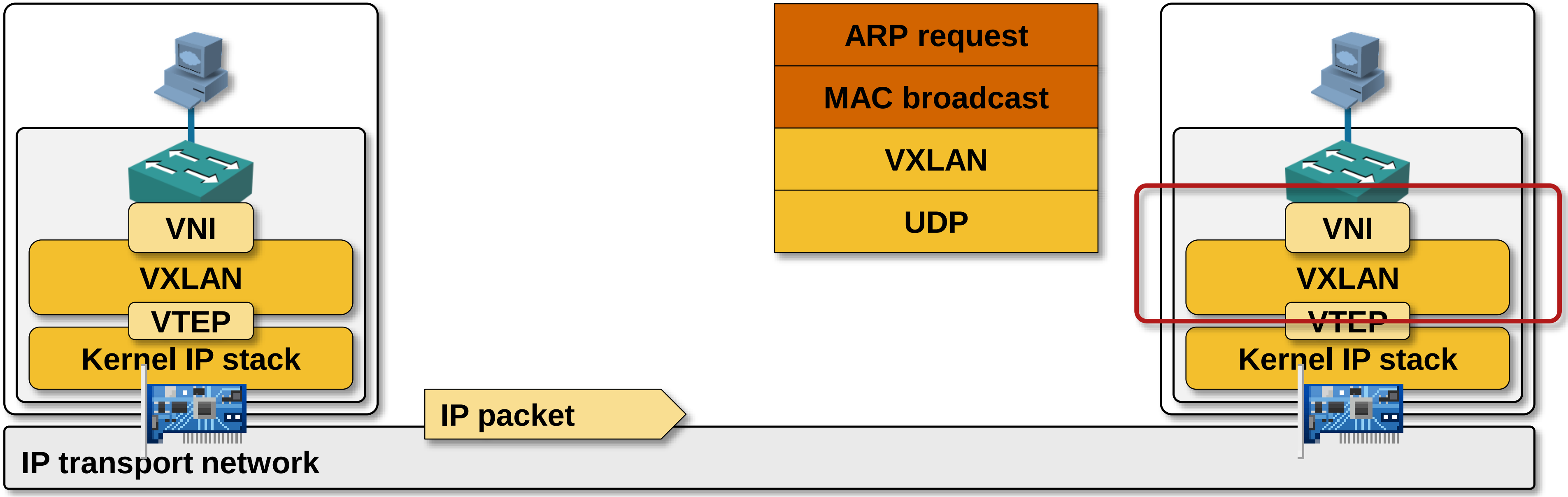
- VM generates ARP broadcast
- VXLAN module sends UDP packet to segment-specific multicast destination
- Kernel IP stack sends multicast IP packet
- Kernel IP stack receives IP multicast

VXLAN Flooding and Learning



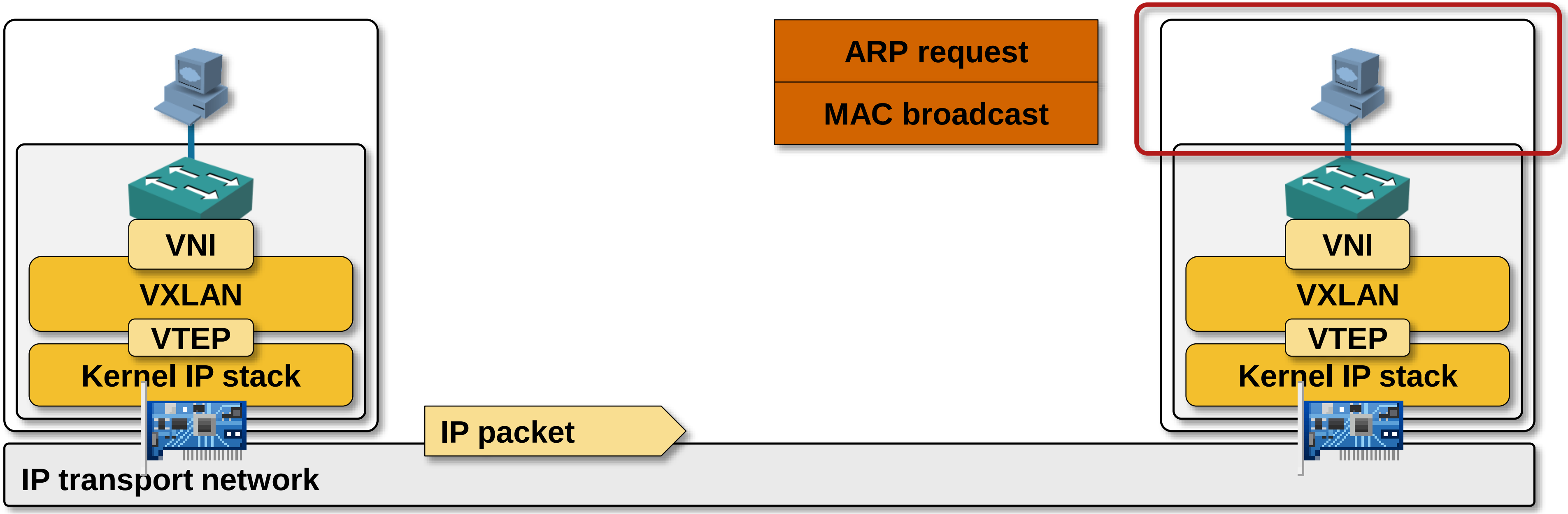
- VM generates ARP broadcast
- VXLAN module sends UDP packet to segment-specific multicast destination
- Kernel IP stack sends multicast IP packet
- Kernel IP stack receives IP multicast
- UDP packet delivered to VXLAN

VXLAN Flooding and Learning



- VM generates ARP broadcast
- VXLAN module sends UDP packet to segment-specific multicast destination
- Kernel IP stack sends multicast IP packet
- Kernel IP stack receives IP multicast
- UDP packet delivered to VXLAN
- VXLAN module remembers remote MAC-to-VTEP mapping

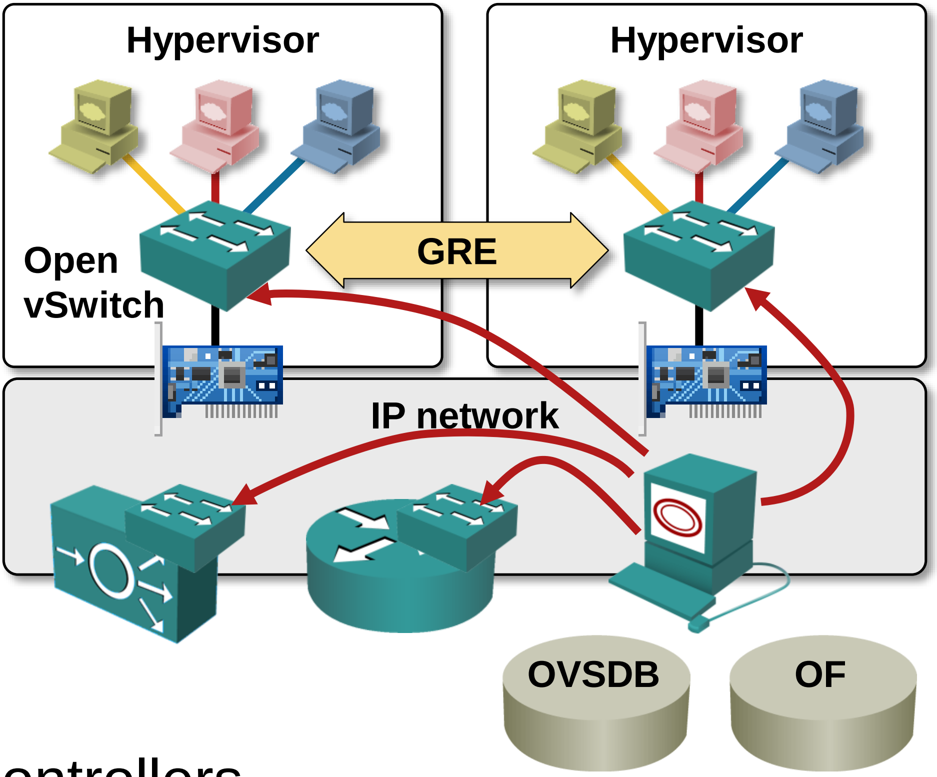
VXLAN Flooding and Learning



- VM generates ARP broadcast
- VXLAN module sends UDP packet to segment-specific multicast destination
- Kernel IP stack sends multicast IP packet
- Kernel IP stack receives IP multicast
- UDP packet delivered to VXLAN
- VXLAN module remembers remote MAC-to-VTEP mapping
- ARP request is delivered to VM

VMware NSX (Nicira NVP)

- OpenFlow-capable hypervisor switches (OVS)
- L2 segments implemented with VLANs or IP tunneling (GRE, STT)
- Logical routers with NAT
- Stateful firewalls
- x86-based L2 or L3 gateways with outside world



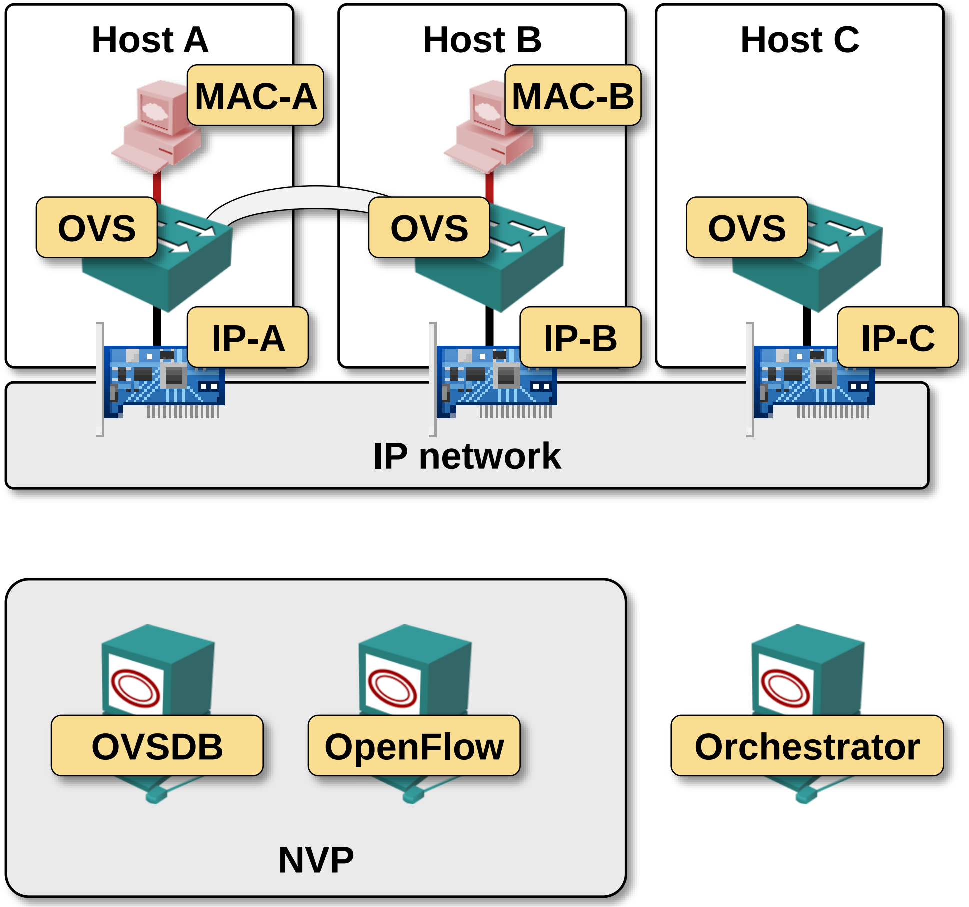
Scalability aspects

- Control plane = cluster of OpenFlow controllers
- Proactive flow setups ➔ controllers are not involved in data plane forwarding
- L2 BUM flooding implemented in hypervisors or through service nodes

No IP multicast required, no network-wide flooding

NSX Control Plane Example

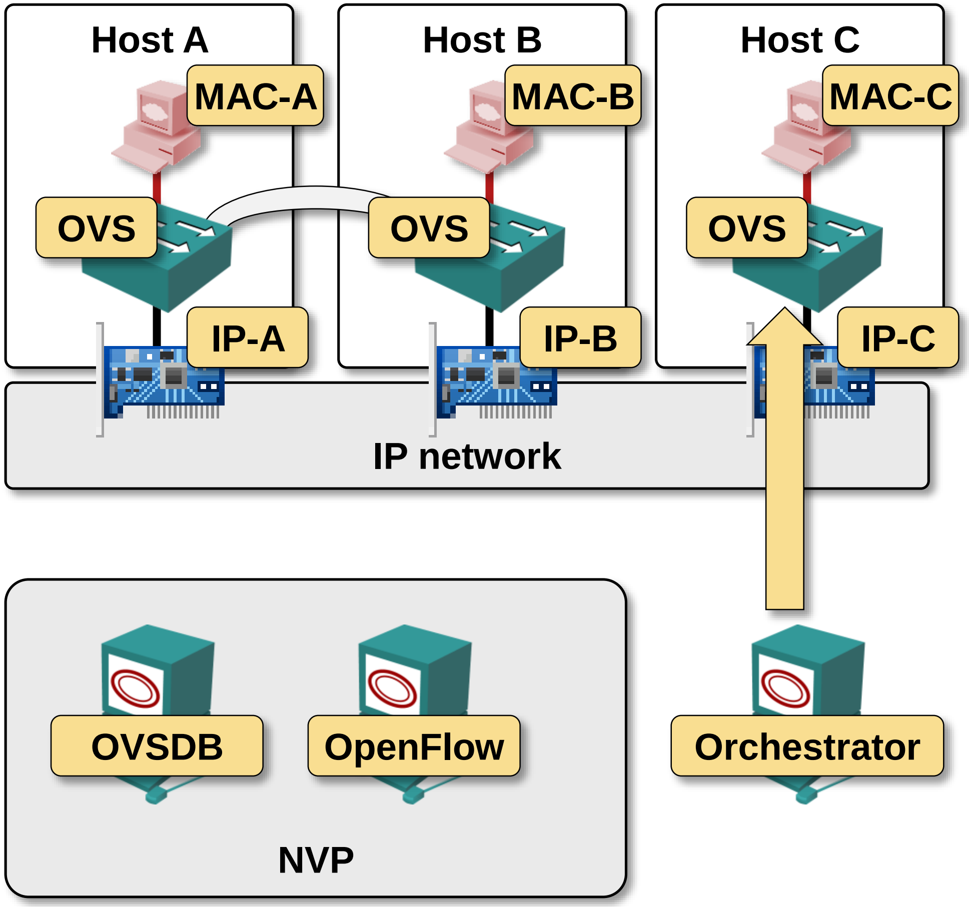
Starting point: GRE-based virtual segment between VMs on A and B



NSX Control Plane Example

Starting point: GRE-based virtual segment between VMs on A and B

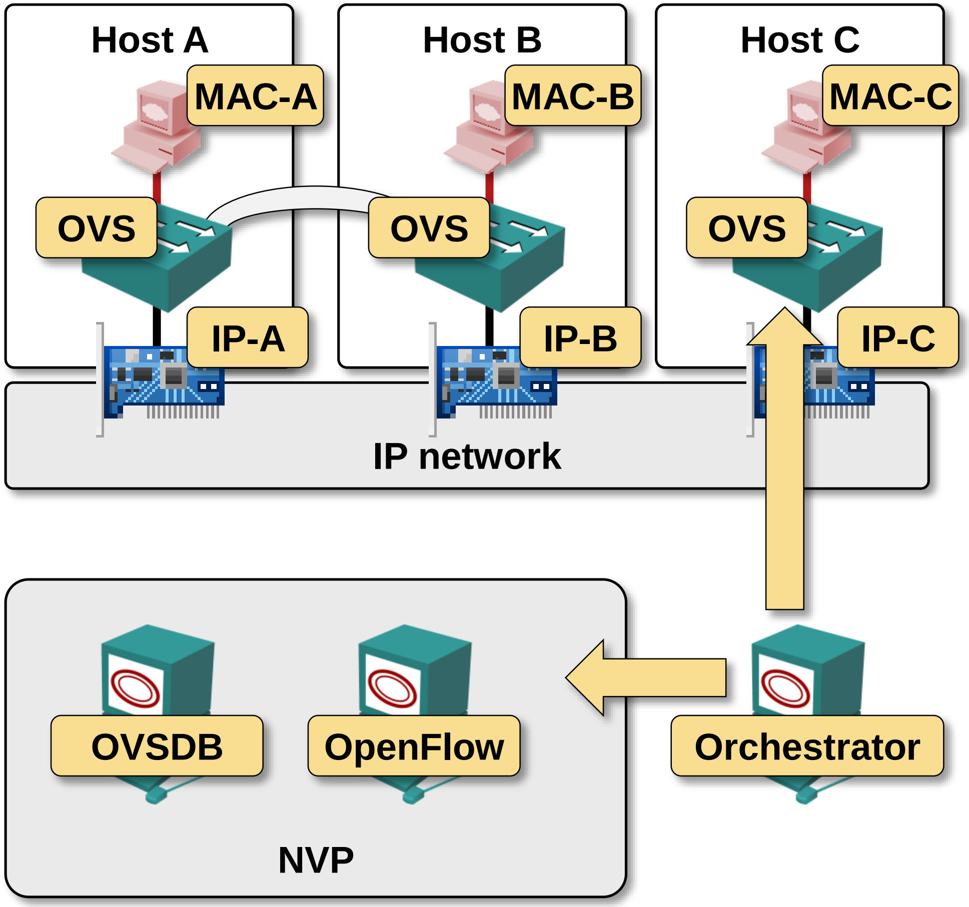
1. Cloud orchestration platform starts a new VM @ Host C



NSX Control Plane Example

Starting point: GRE-based virtual segment between VMs on A and B

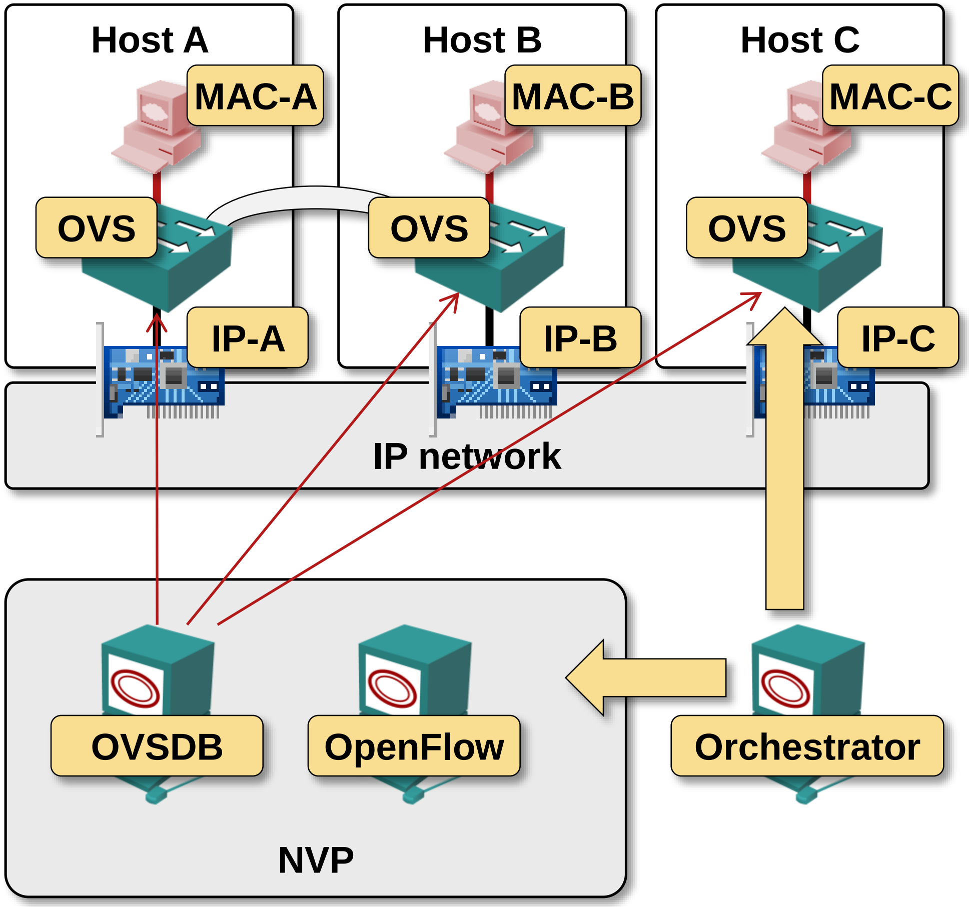
1. Cloud orchestration platform starts a new VM @ Host C
2. NVP receives information from orchestration platform



NSX Control Plane Example

Starting point: GRE-based virtual segment between VMs on A and B

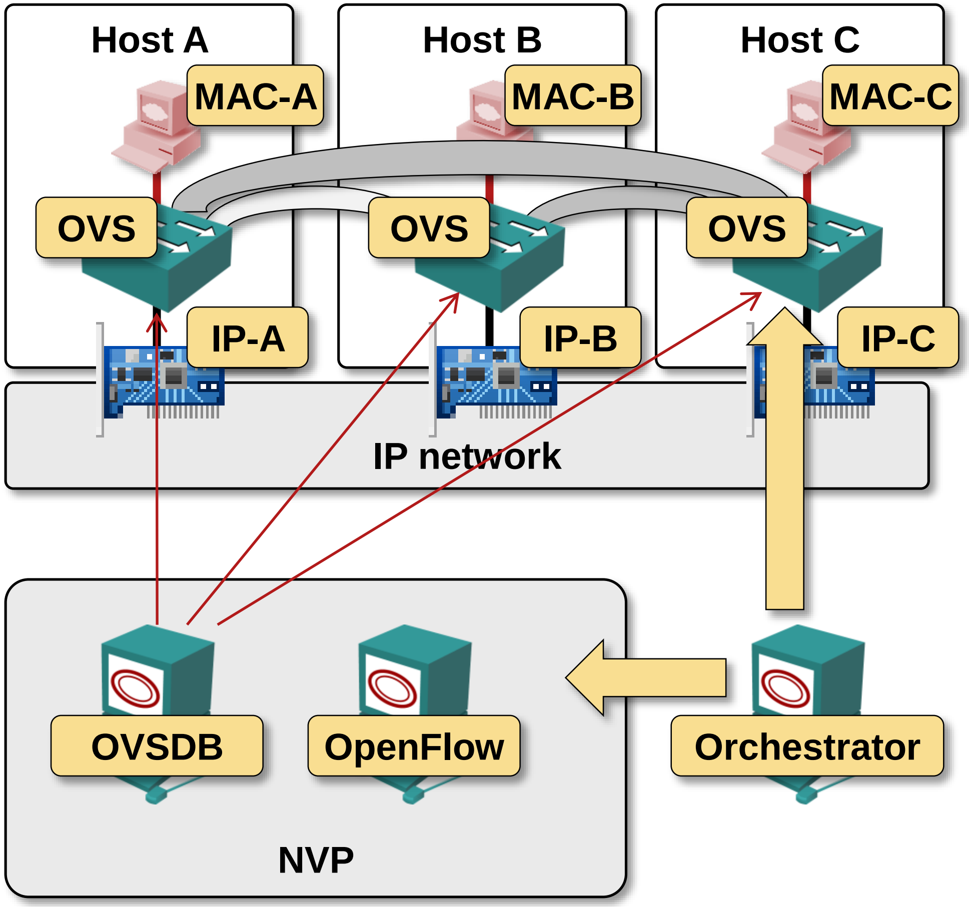
1. Cloud orchestration platform starts a new VM @ Host C
2. NVP receives information from orchestration platform
3. NVP creates OVSDB entries for new GRE/STT tunnels (A-C, B-C)



NSX Control Plane Example

Starting point: GRE-based virtual segment between VMs on A and B

1. Cloud orchestration platform starts a new VM @ Host C
2. NVP receives information from orchestration platform
3. NVP creates OVSDB entries for new GRE/STT tunnels (A-C, B-C)
4. OVS switches create new interfaces

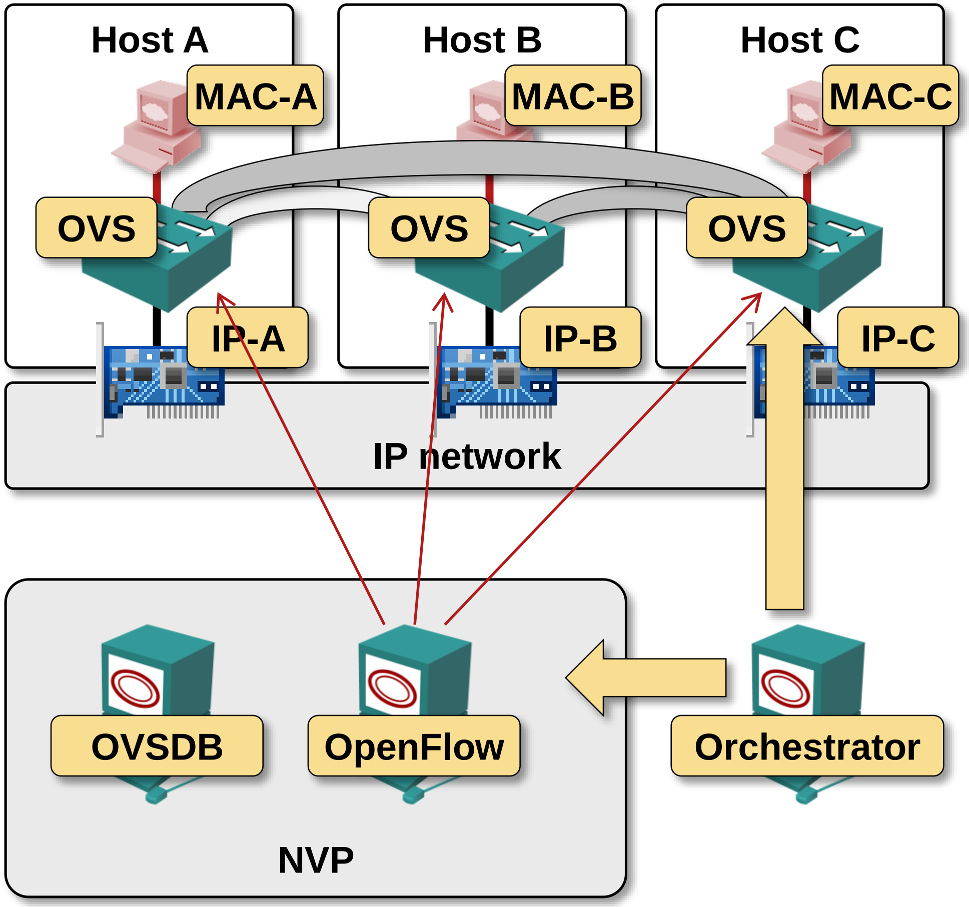


NSX Control Plane Example

Starting point: GRE-based virtual segment between VMs on A and B

1. Cloud orchestration platform starts a new VM @ Host C
2. NVP receives information from orchestration platform
3. NVP creates OVSDB entries for new GRE/STT tunnels (A-C, B-C)
4. OVS switches create new interfaces
5. NVP pushes new OpenFlow entries to hypervisor hosts

Switch	OpenFlow entry	Interface
Host-A	DMAC = MAC-C	A-C tunnel
Host-B	DMAC = MAC-C	B-C tunnel
Host-C	DMAC = MAC-A	C-A tunnel
Host-C	DMAC = MAC-B	C-B tunnel



Gateway to the Outside World

Gateways? What Gateways?

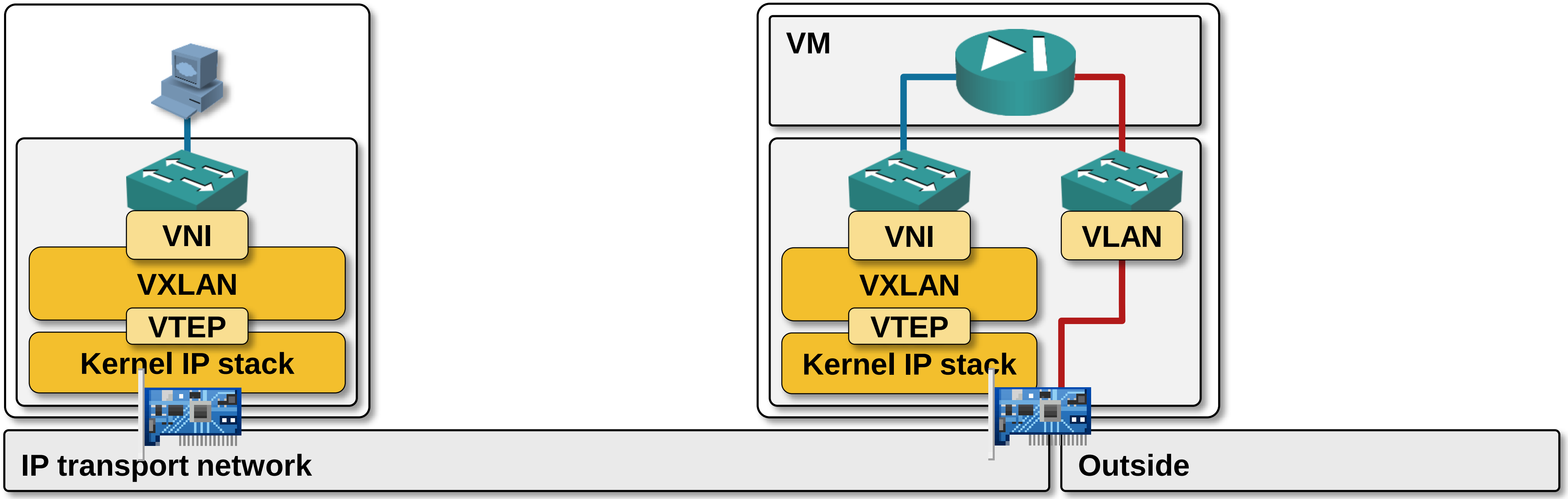
Existing networking gear rarely supports overlay networks

- Shipping: VXLAN support on Arista 7150 (L2 gateway), VXLAN on F5 BGI-IP, NVGRE on IRON Networks MNV Appliance
- Announced: NVGRE on F5 BIG-IP, VXLAN on Brocade ADX, NVGRE on Dell Force10 ...
- Brocade “reaffirmed its commitment to NVGRE”
- Avaya “is actively supporting the VXLAN initiative”

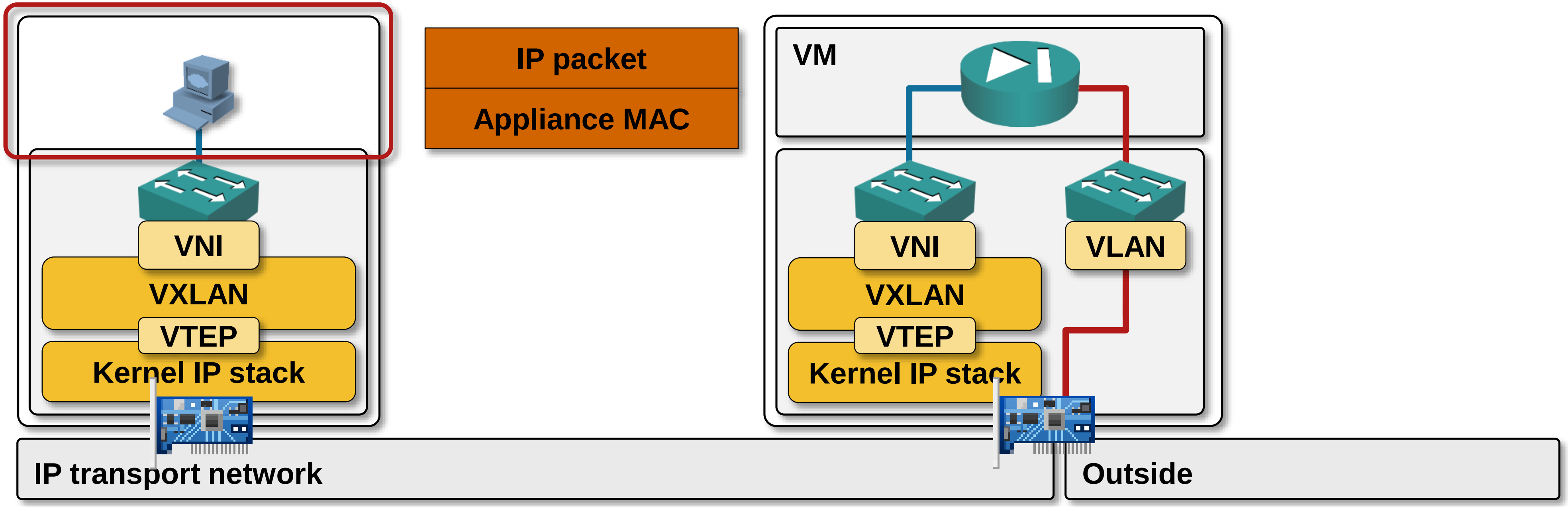
Don’t despair, focus on VM-based appliances

- Reasonable performance
- Acceptable feature set
- Much higher flexibility and ease-of-deployment

Sample Layer-3 VM Appliance Integration Scenario

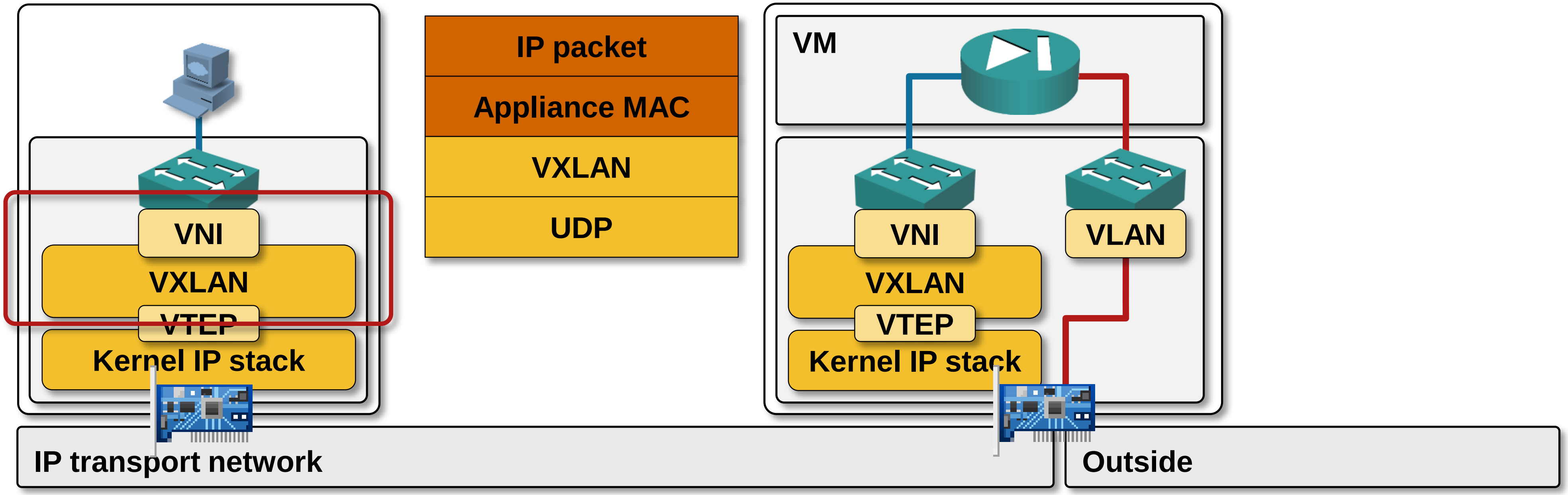


Sample Layer-3 VM Appliance Integration Scenario



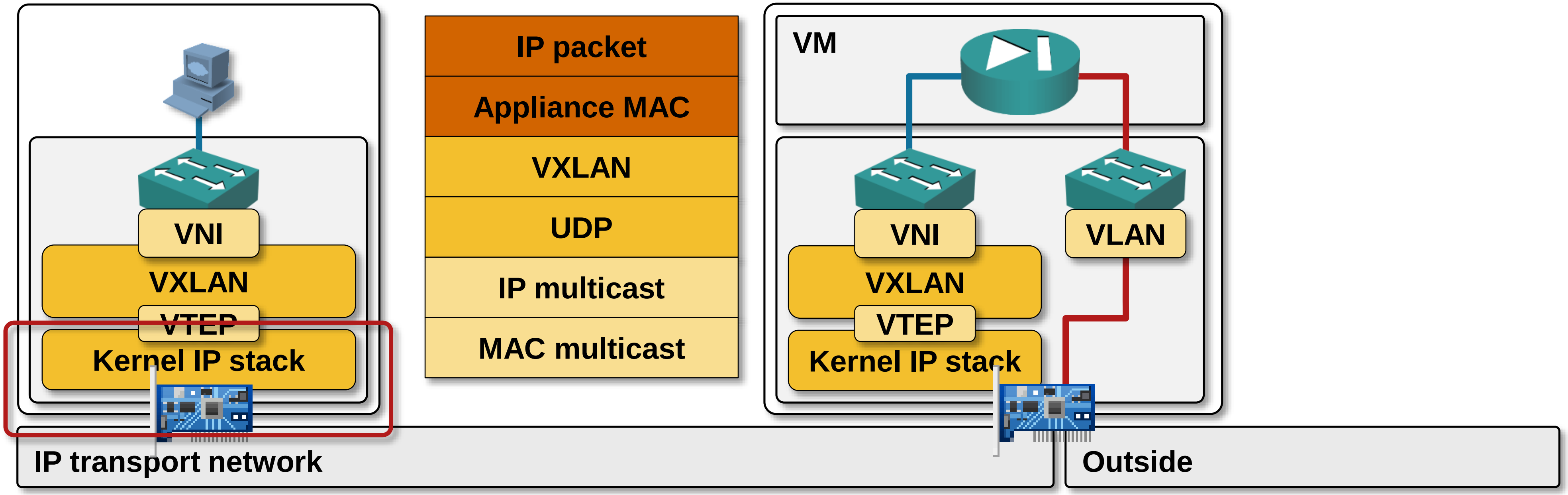
- VM sends IP packet to appliance MAC address

Sample Layer-3 VM Appliance Integration Scenario



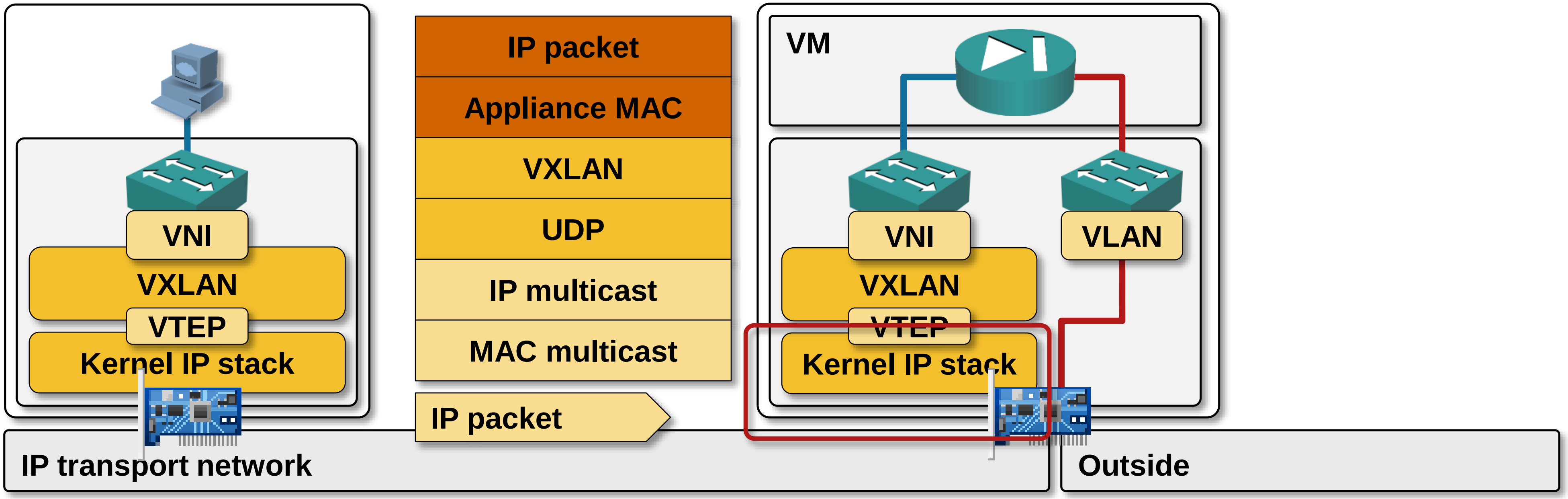
- VM sends IP packet to appliance MAC address
- VXLAN module encapsulates VM packet in VXLAN+UDP envelope

Sample Layer-3 VM Appliance Integration Scenario



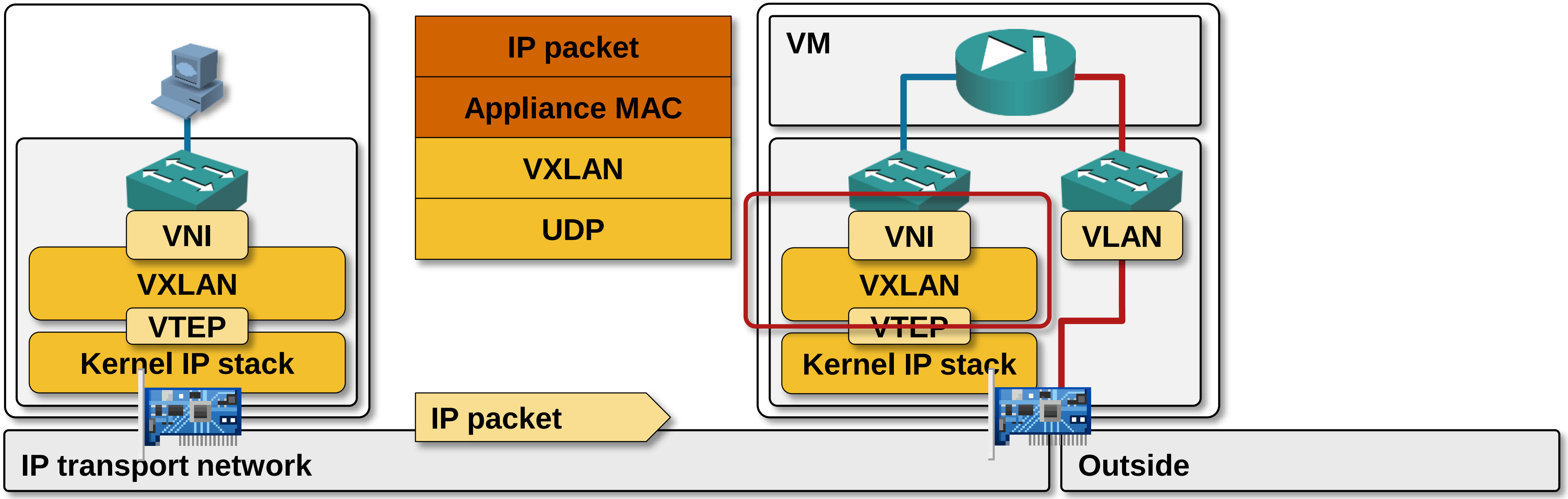
- VM sends IP packet to appliance MAC address
- VXLAN module encapsulates VM packet in VXLAN+UDP envelope
- Kernel IP stack sends IP packet to hypervisor running the appliance

Sample Layer-3 VM Appliance Integration Scenario



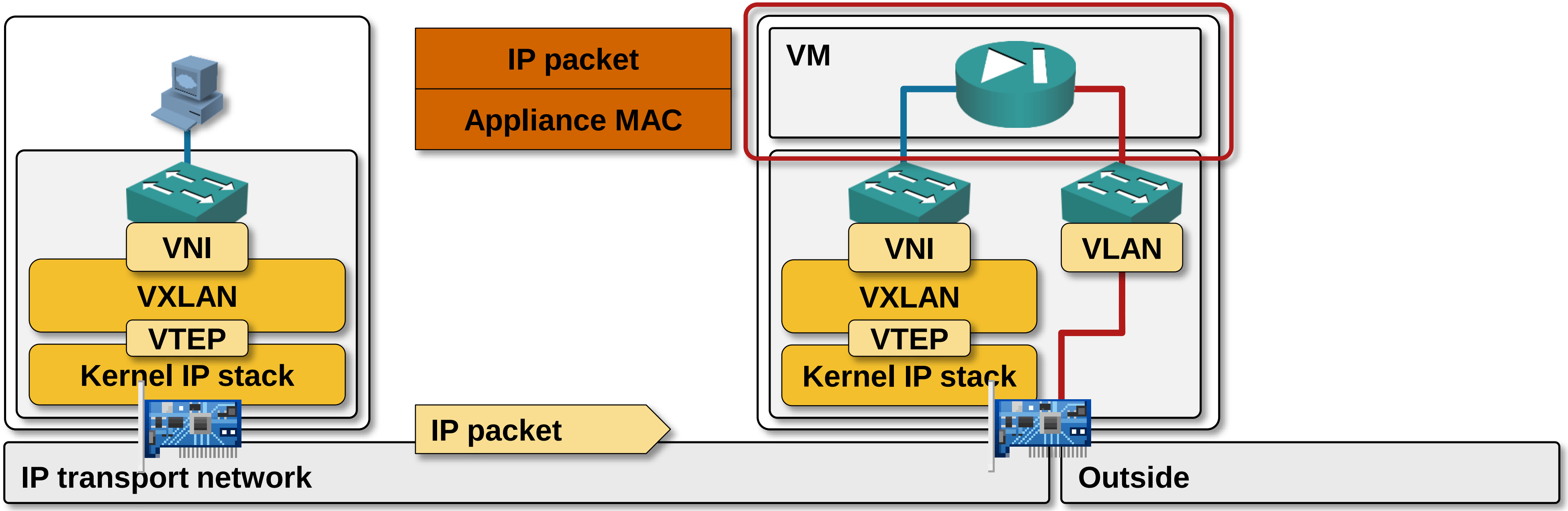
- VM sends IP packet to appliance MAC address
- VXLAN module encapsulates VM packet in VXLAN+UDP envelope
- Kernel IP stack sends IP packet to hypervisor running the appliance
- IP packet received by kernel IP stack

Sample Layer-3 VM Appliance Integration Scenario



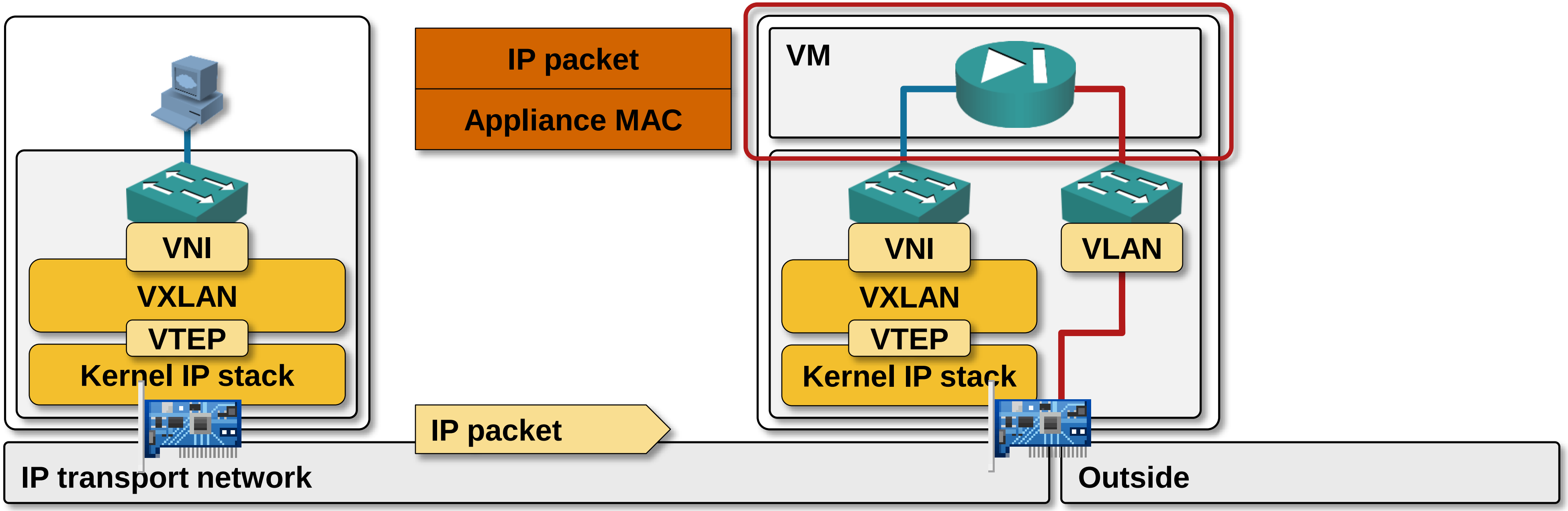
- VM sends IP packet to appliance MAC address
- VXLAN module encapsulates VM packet in VXLAN+UDP envelope
- Kernel IP stack sends IP packet to hypervisor running the appliance
- IP packet received by kernel IP stack
- UDP packet delivered to VXLAN

Sample Layer-3 VM Appliance Integration Scenario



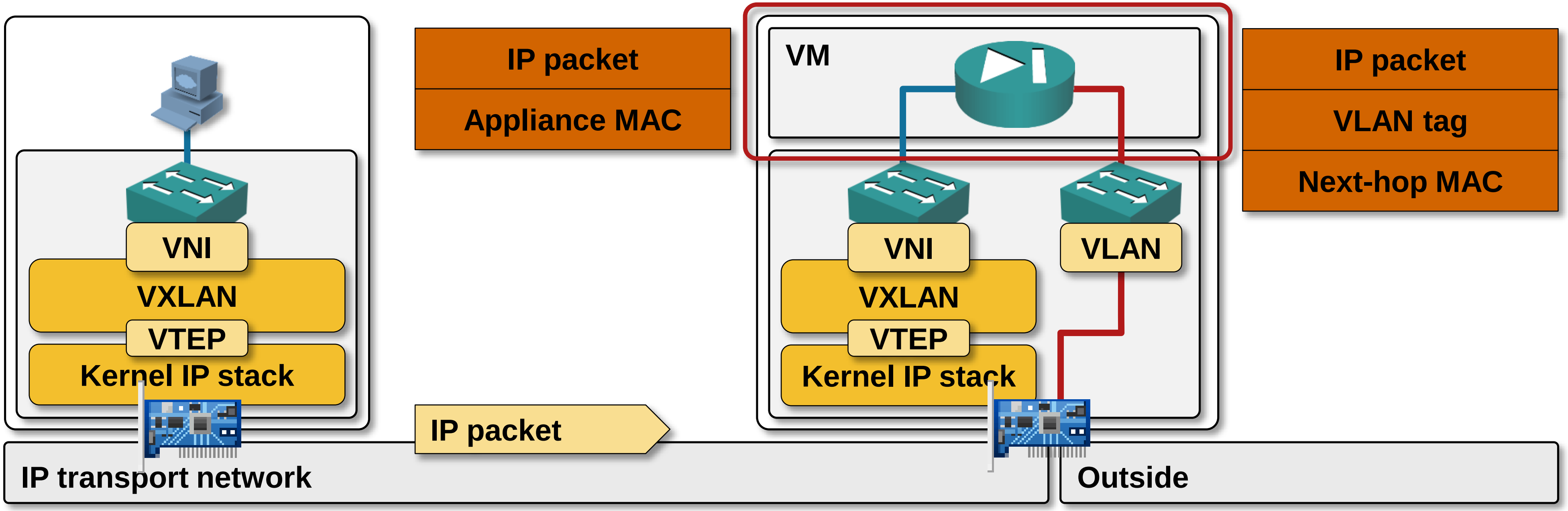
- VM sends IP packet to appliance MAC address
- VXLAN module encapsulates VM packet in VXLAN+UDP envelope
- Kernel IP stack sends IP packet to hypervisor running the appliance
- IP packet received by kernel IP stack
- UDP packet delivered to VXLAN
- Inner IP packet delivered to VM appliance

Sample Layer-3 VM Appliance Integration Scenario



- VM sends IP packet to appliance MAC address
- VXLAN module encapsulates VM packet in VXLAN+UDP envelope
- Kernel IP stack sends IP packet to hypervisor running the appliance
- IP packet received by kernel IP stack
- UDP packet delivered to VXLAN
- Inner IP packet delivered to VM appliance
- VM appliance processes IP packet

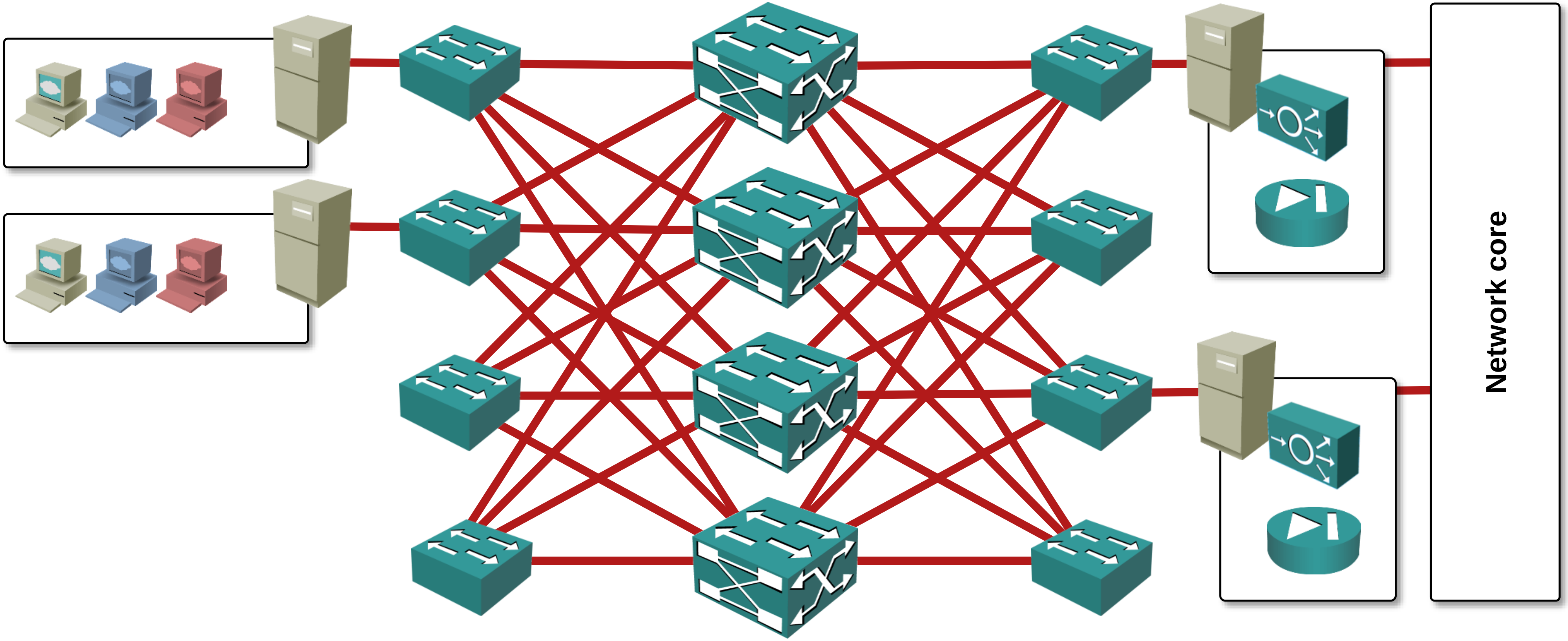
Sample Layer-3 VM Appliance Integration Scenario



- VM sends IP packet to appliance MAC address
- VXLAN module encapsulates VM packet in VXLAN+UDP envelope
- Kernel IP stack sends IP packet to hypervisor running the appliance

- IP packet received by kernel IP stack
- UDP packet delivered to VXLAN
- Inner IP packet delivered to VM appliance
- VM appliance processes IP packet
- VM appliance sends IP packet to outside VLAN

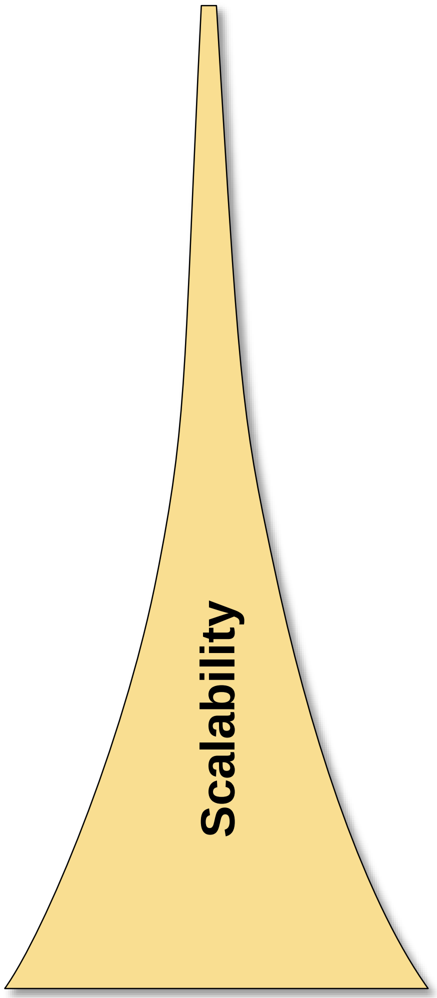
Putting It All Together: Network Fabric = Resource Pool



- Build a leaf-and-spine (or multistage Clos) fabric
- Connect compute + storage capacity to the fabric
- Deploy virtual appliances in a dedicated cluster linked to the network core

Conclusions

Solution Space and Scalability



VLANs

4096 segments

VM-aware Networking (Arista VM Tracer)

Edge Virtual Bridging (EVB, 802.1Qbg)

Emerging

EVB with PBB/SPB (L2 over L2)

Theoretical

VXLAN (Cisco / VMware)

No control plane

Unicast VXLAN

VMware NSX (L2+L3 over IP + Control Plane)

Juniper Contrail (L3 over IP + Control Plane)

Microsoft WNV (L3 over IP + Control Plane)

Based on features shipping in September 2013

When Should You Use Overlay Networking?

Private or public clouds

- Use overlay virtual networks for stability and scalability
- Use virtual appliances for fast and more flexible deployment

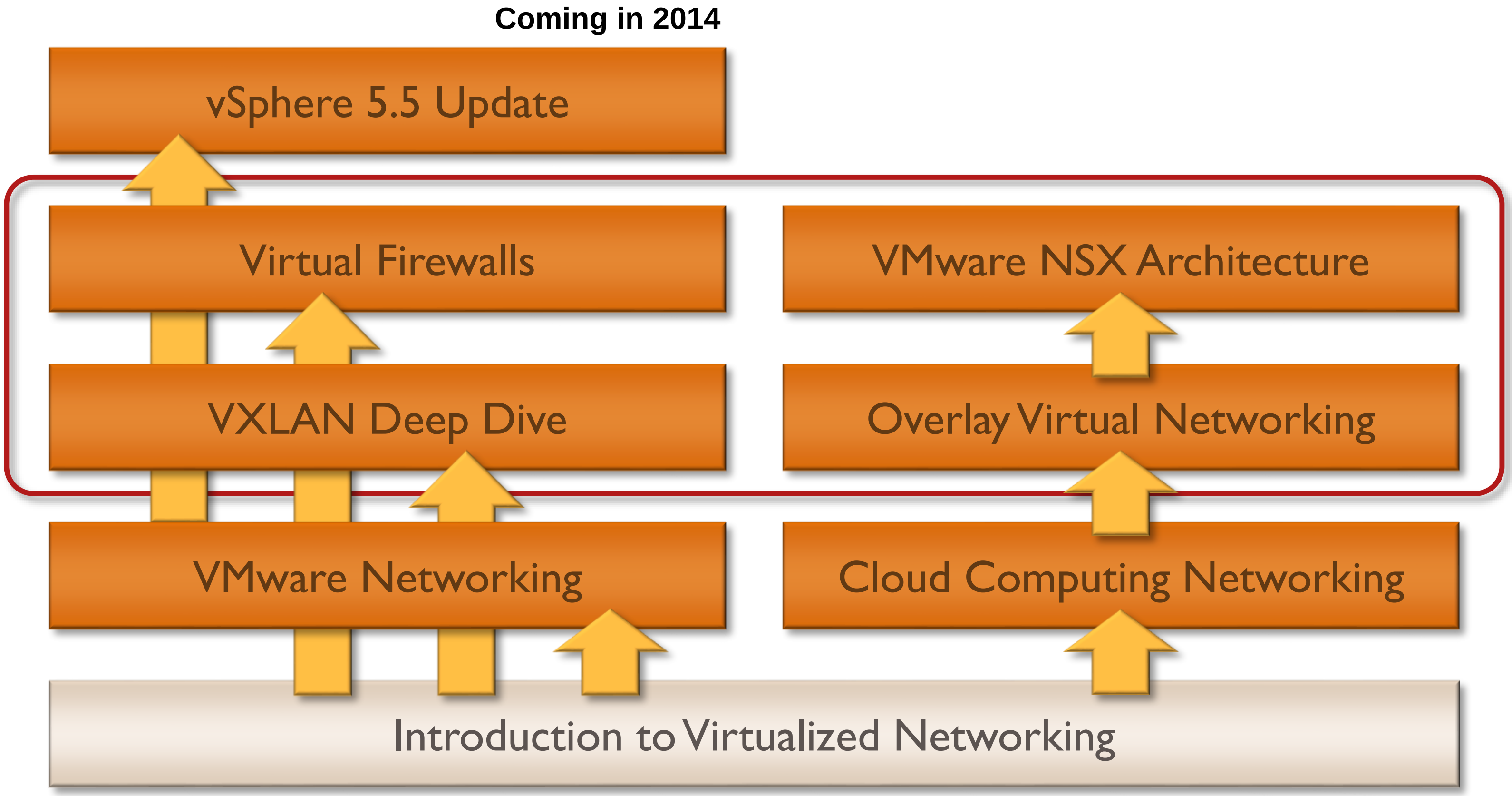
Traditional server virtualization

- Migrate toward virtual appliances
- Overlay networks will enable faster virtual network provisioning
- Use them if you’re approaching VLAN scalability limits

Overlay virtual networks are not a good fit

- Siloed IT not yet ready for restructuring / convergence
- Small deployments where existing solutions are good enough

Reference: Virtualization Webinars on ipSpace.net



Availability

- Live sessions
- Recordings of individual webinars
- Yearly subscription

Other options

- Customized webinars
- ExpertExpress
- On-site workshops

More information @ http://www.ipSpace.net/Webinars

Reference: Blogs and Podcasts

- Packet Pushers Podcast & blog (packetpushers.net)
- The Cloudcast (.net)
- Network Heresy (Martin Casado, Nicira/VMware)
- Blog.scottlowe.org (Scott Lowe, VMware)
- BradHedlund.com (Brad Hedlund, VMware)
- RationalSurvivability.com (Christopher Hoff, Juniper)
- High Scalability Blog
- it20.info (Massimo Re Ferre, VMware)
- NetworkJanitor.net (Kurt Bales)
- Yellow bricks (Duncan Epping, VMware)
- blog. ipspace.net (yours truly)

Questions?

Send them to ip@ipSpace.net or @ioshints