Edge Virtual Bridging with VEB and VEPA

May 2009

Chuck Hudson (HP) c.hudson@hp.com
Paul Congdon (HP) ptcongdon@ucdavis.edu

Agenda

- Life on the Edge
- VEBs are here to stay
- Extending VEBs with Tag-less VEPAs
- Comparing EVB Approaches
- Using Promiscuous vPorts with VEPA
- Other 'Case Studies'
- Conclusion

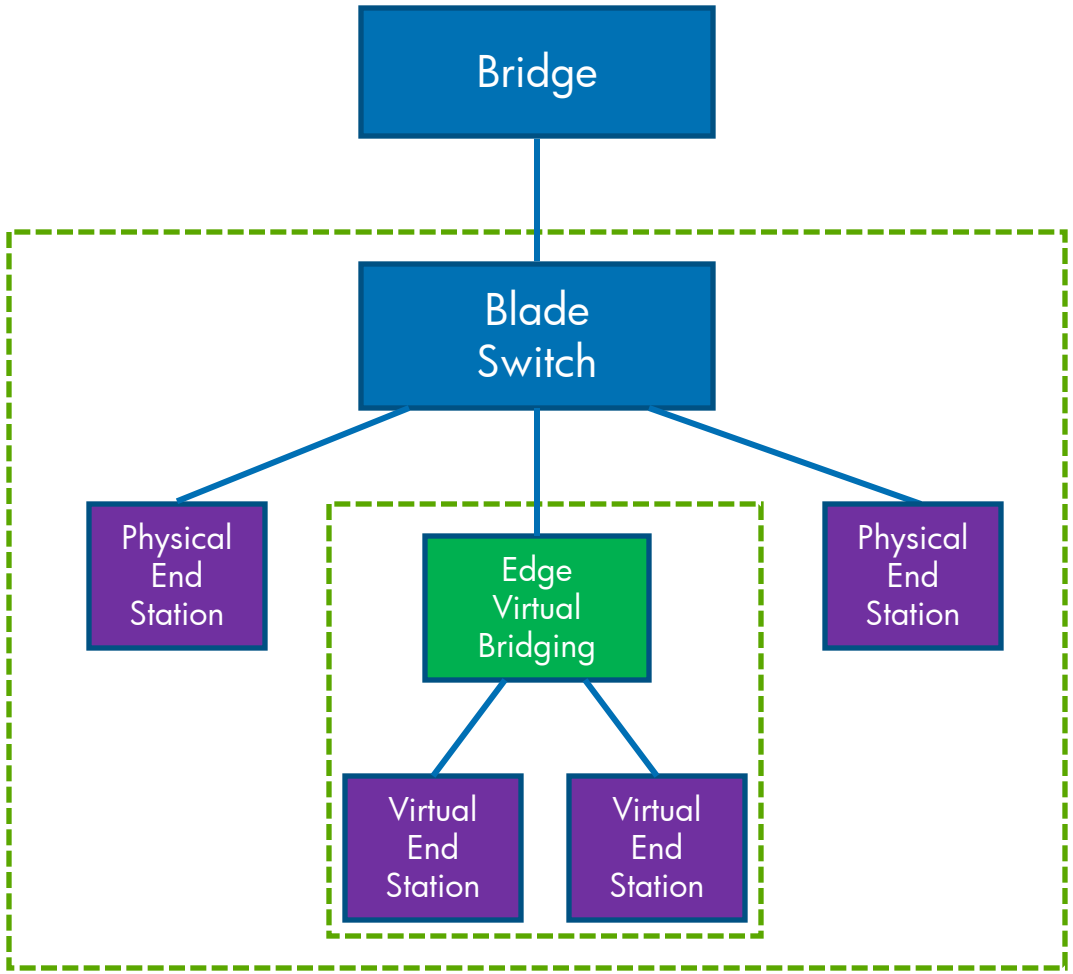
Life on the Edge

Edge Virtual Bridging
A Definition

Edge Virtual Bridging (EVB) is an environment where physical end stations contain multiple virtual end stations that participate in the Ethernet network environment.

Note: EVB environments are unique in that vNIC configuration information is available that is not normally available to an 802.1Q bridge.

Edge Virtual Bridging
At the edge, in the physical end station



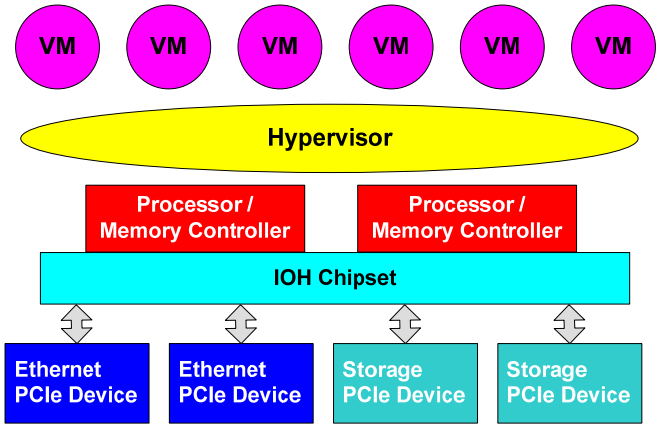
Edge Virtual Bridging
includes…

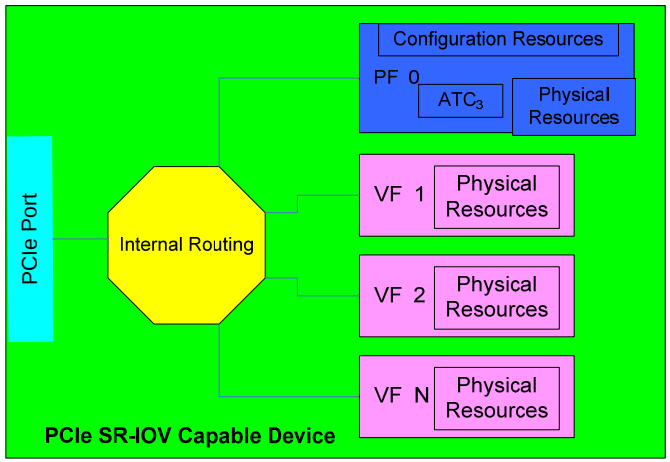
- Virtual Machine Environments (Virtual Switch)
  - VMware ESX Server
  - Microsoft HyperV
  - Citrix XEN
  - Linux KVM (linux-kvm.org)
- Proprietary offerings from HP, IBM, Sun, Oracle, etc.
- NICs with multiple vNICs that share a single link
  - PCI Single-root or Multiple-root IO Virtualization (SR-IOV, MR-IOV)
  - Other multi-vNIC technologies

PCI Standards for IO Virtualization

- PCI Standards for
  - Single Root IO Virtualization (SR-IOV)
  - Multiple Root IO Virtualization (MR-IOV)
- Allows for many PCI functions to be created…
  - That share a single physical device
  - That share a single physical uplink
  - That appear as multiple, separate devices to the operating system
  - So each virtual machine has direct access to its own buffer
- SR-IOV NICs
  - Separate buffer for each virtual function (vNIC)
  - May implement hundreds of virtual functions
  - Will usually implement ETS queues at the shared link

SR-IOV Device



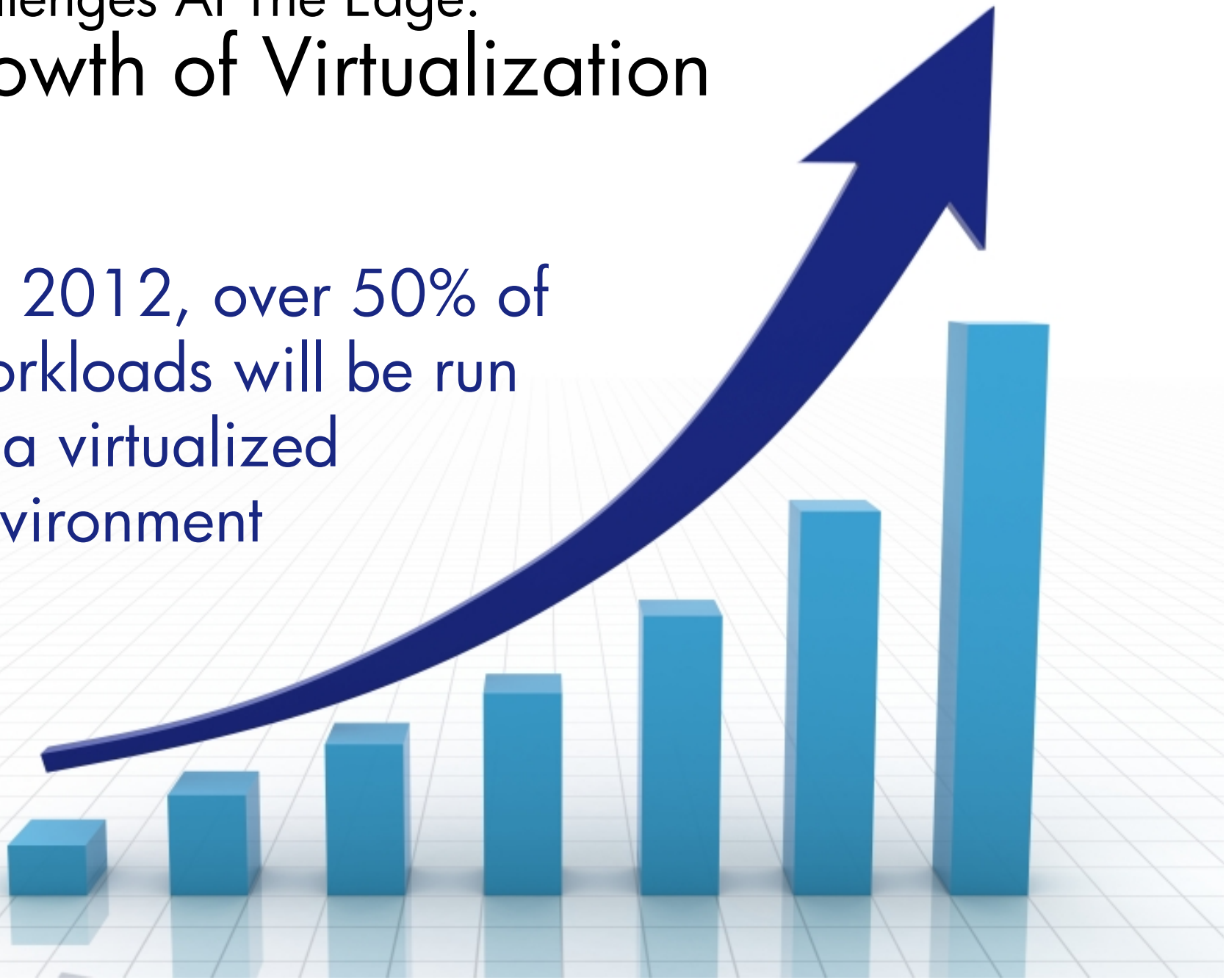


- Any type of I/O device
  - Ethernet, Fibre Channel, Graphics, SAS, etc.
- PCI Express (PCIe) Device
  - 1-256 Physical Functions (PF)
    - Typically 1-16 PF per device
    - Full PCIe Function + SR-IOV capability
    - Owned by hypervisor
    - Device-specific management function and control of shared resources, e.g. Ethernet Port
  - 1-64K Virtual Functions (VF)
    - Typically 32-255 per device
    - Light-weight hardware resources to reduce cost and device complexity
    - Owned by Virtual Machine (VM) Guest
    - Direct VM hardware access for data movements – no hypervisor overhead
      - Infrequent configuration operations trap to hypervisor
  - Typical device <= 256 PF+VF

Challenges At The Edge:

Growth of Virtualization

By 2012, over 50% of workloads will be run in a virtualized environment



Challenges At The Edge: Visibility & Control

- System administrators control embedded switching in their end stations.
  - System admins have physical end station 'root access'
  - Hypervisor consoles integrate support for multiple virtual machine hosts
- No standards in place for coordinating management between hypervisors and external bridging.
  - Multiple, vendor-specific vSwitches/VEBs are impractical
- Lack of network admin control can mean inadequate:
  - Control of network access
  - Visibility of networking traffic
  - Support for debugging network issues

Challenges At The Edge:
Limited Embedded Capability

- NICs have cost & complexity constraints
  - Usually do not support TCAMs, etc.
- NICs usually focused on end-station IO capability
  - Specialized device types (NIC, iSCSI, FCoE)
  - NIC teaming
  - Multi-OS support
- End-stations and bridges evolve independently

Virtual Ethernet Bridges (VEBs) are here to stay

Virtual Ethernet Bridge
A Definition

A Virtual Ethernet Bridge (VEB) is a capability within a physical end station that supports local bridging between multiple virtual end stations and (optionally) the external bridging environment.

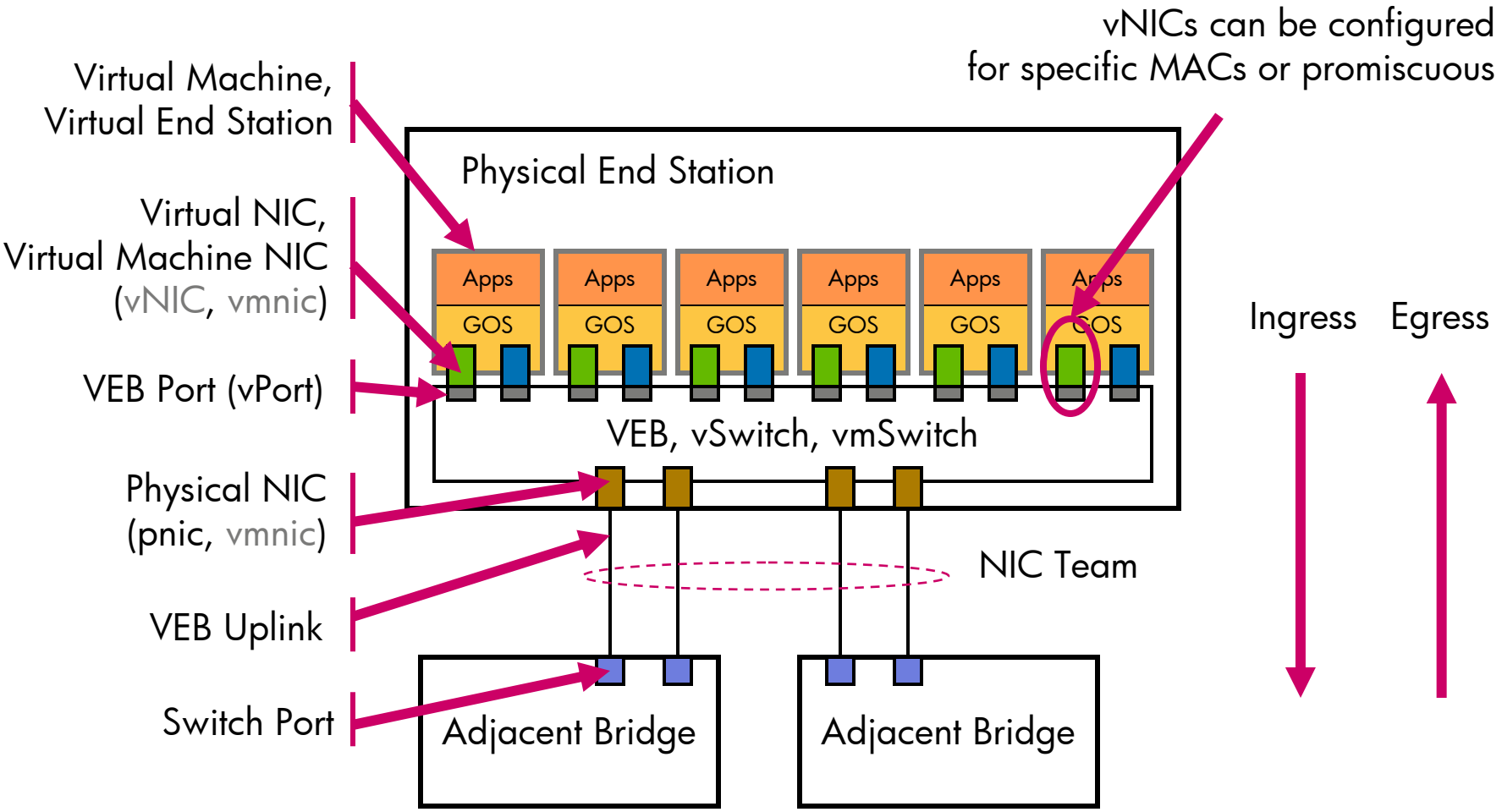
May be implemented in software as a virtual switch (vSwitch) or with embedded hardware.

Note: With VEBs, vNIC configuration information is available that is not normally available to an 802.1Q bridge.

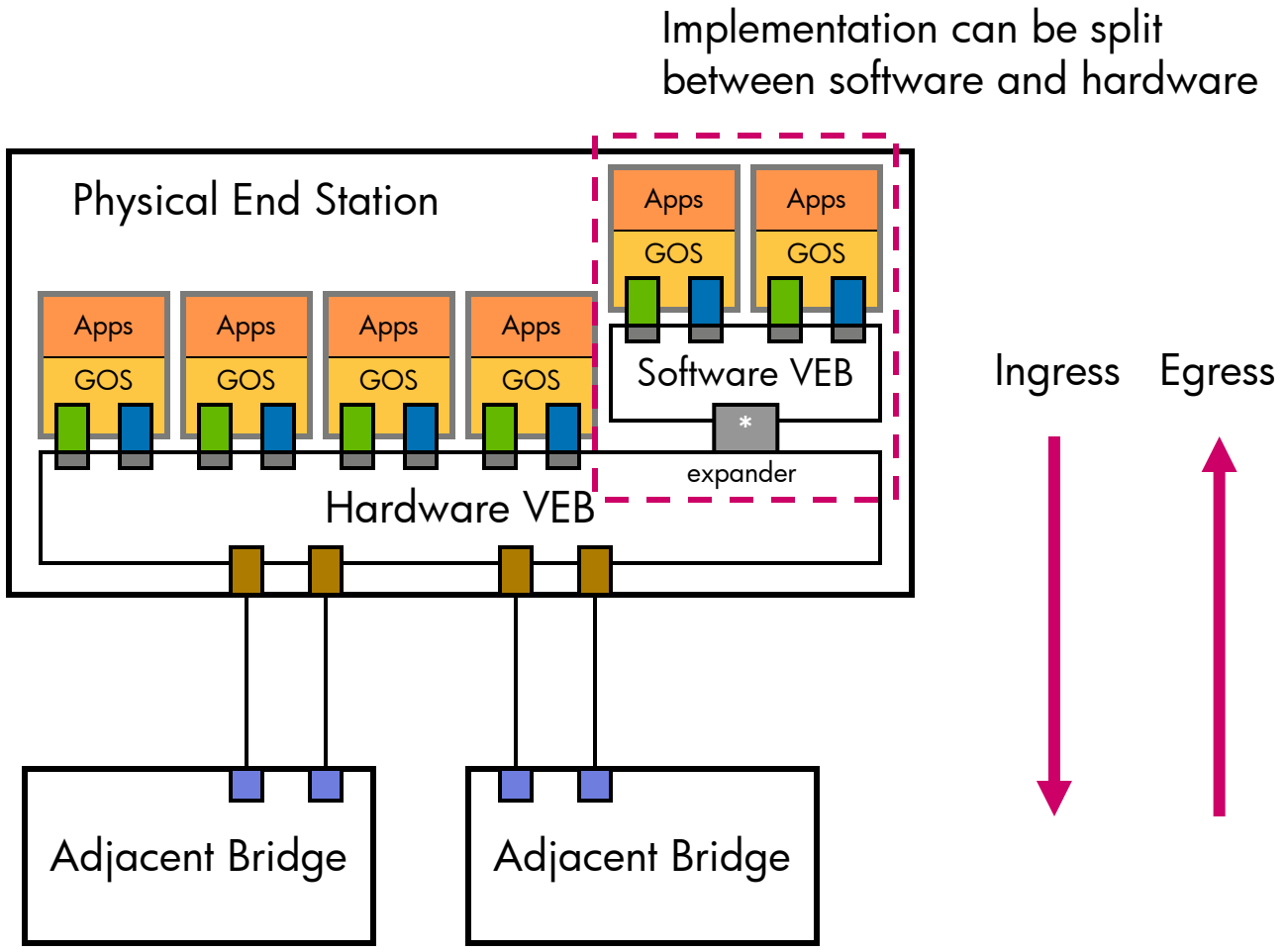
VEBs Are Here To Stay

- All hypervisor environments support some form of VEB capability today.
- Local bridging with VEB is needed to allow hypervisors to:
  - Operate without external bridges attached
  - Operate with a broad range of Ethernet environments
  - Maximize local bandwidth
  - Minimize local latency
  - Minimize local packet loss
- VEB capability will still be required for hypervisors and SR-IOV NICs

Basic VEB Anatomy and Terms

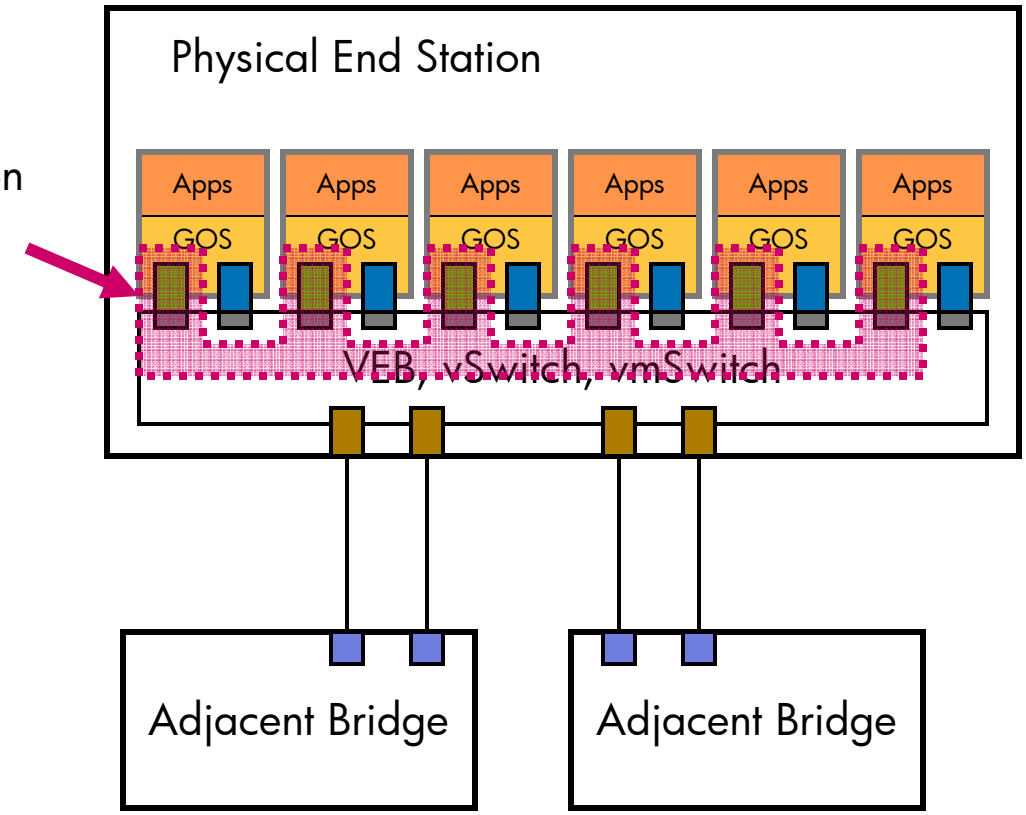


Basic VEB Anatomy and Terms
Showing Hardware + Software VEB

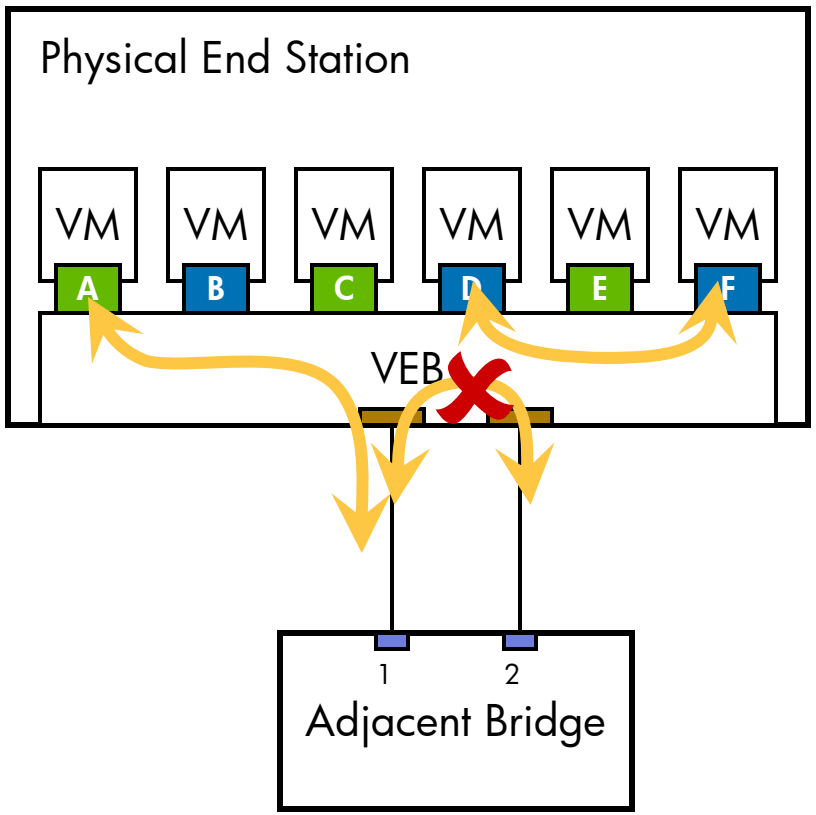


Basic VEB Anatomy and Terms
Showing Port Group

Port Group
A set of vPorts with similar configuration (such as VLAN ID)



VEB Loop-free Forwarding Behavior



- Forwards based on MAC address (and port group or VLAN)
- Forwards
  - vNIC ←→ vNIC
  - vNIC ←→ Uplink
- Does NOT forward from uplink to uplink
  - Single active logical uplink
  - Multiple uplinks may be 'teamed' (802.3ad and other algorithms)
- Does not participate in (or affect) spanning tree

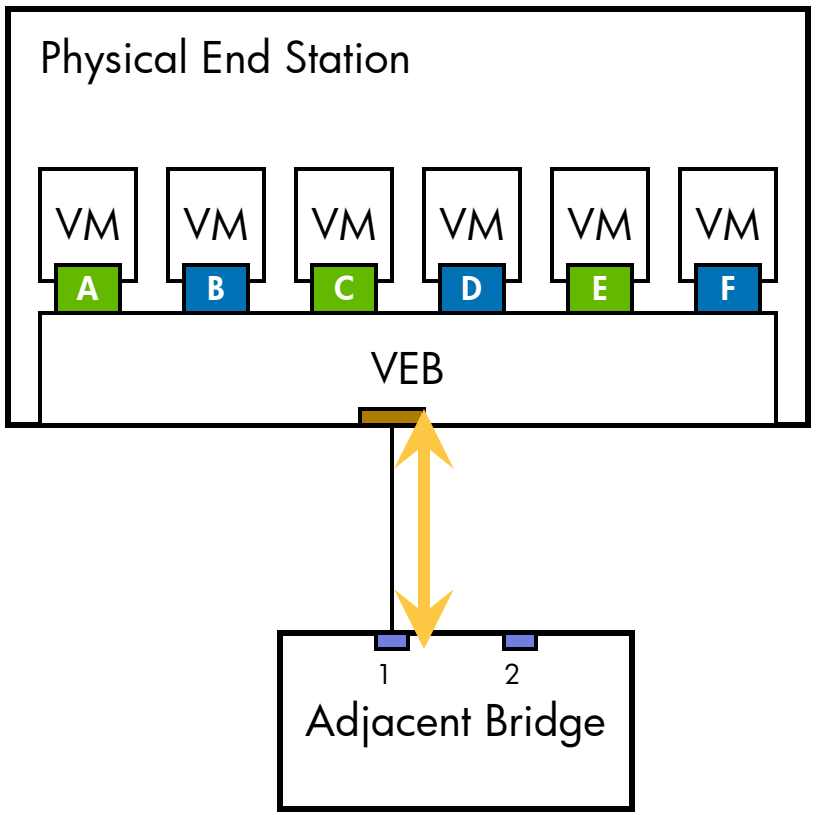
Mgmt & Config of VEB vPorts

- General VLAN mode
  - VEB/vSwitch can terminate or pass VIDs through to VM
  - Affects all ports in VEB
- Per-port VLAN Settings
  - vPort VLAN ID
  - egress VLAN IDs (VIDs that can reach the port)
- Addressing Security
  - Limit to assigned MAC
  - Allow guest-OS specific MACs
  - Promiscuous
- Default priority and/or priority mapping
- Traffic shaping & bandwidth management

Mgmt & Config of VEB Uplinks

- General VLAN mode
  - VEB/vSwitch can terminate or pass VIDs through to VM
  - Affects all ports in VEB
- Uplinks (NICs) associated with a VEB
- NIC Teaming Mode
  - Fail-over
  - Transmit load-balancing
  - Bi-direction load-balancing (802.3ad, etc.)
- DCBX Configuration
  - ETS Queues
  - Priority Flow Control

LLDP and DCBX

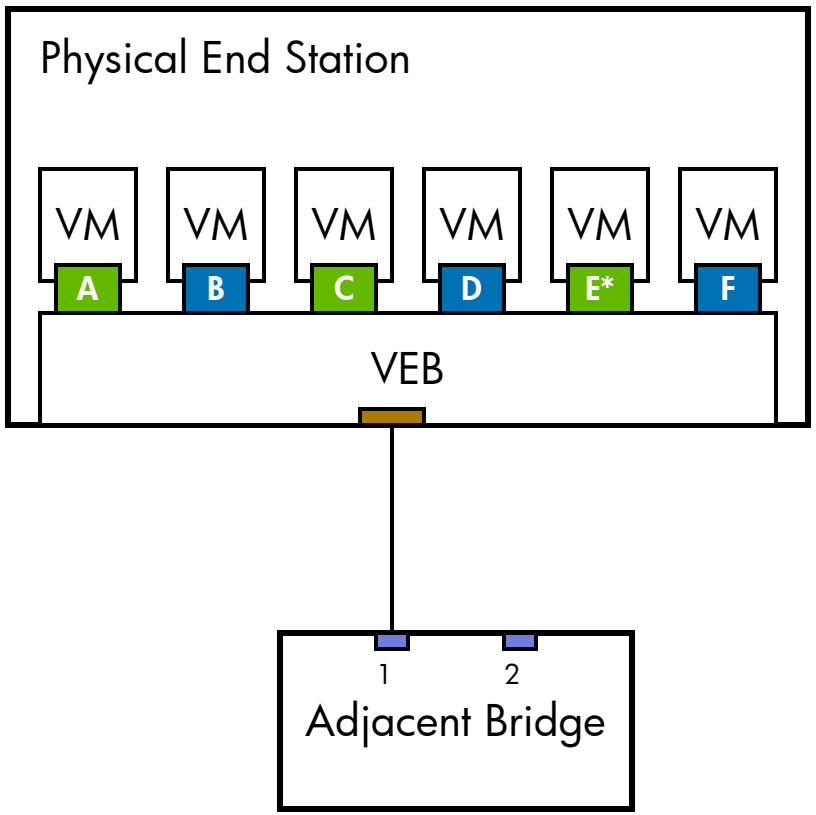


- LLDP & DCB are between…
  - VEB uplink and
  - Adjacent bridge port
  - Uses MAC of physical NIC
- LLDP
  - Identifies physical NIC of the physical end station
- DCBX
  - Configures the physical NIC
  - Physical NIC ETS queues
  - Physical NIC PFC settings
- vNICs typically implement a single (lossless) queue

VEB Address Table Management

- VEBs typically don't do learning
  - Intended to be at the edge of the network, not the middle
- Instead, MAC addresses can be known by registration
  - Hypervisors set vNIC default MAC address
  - Hypervisors can intercept when a guest OS sets receive filters on a vNIC
    - Locally Administered Address (LAA)
    - Multicast addresses
- VEB Address Table entries
  - Provide forwarding information
  - Provide the receive filtering for the vNICs
  - Provides multicast filtering without IGMP snooping

VEB Address Table
Populated via MAC registration



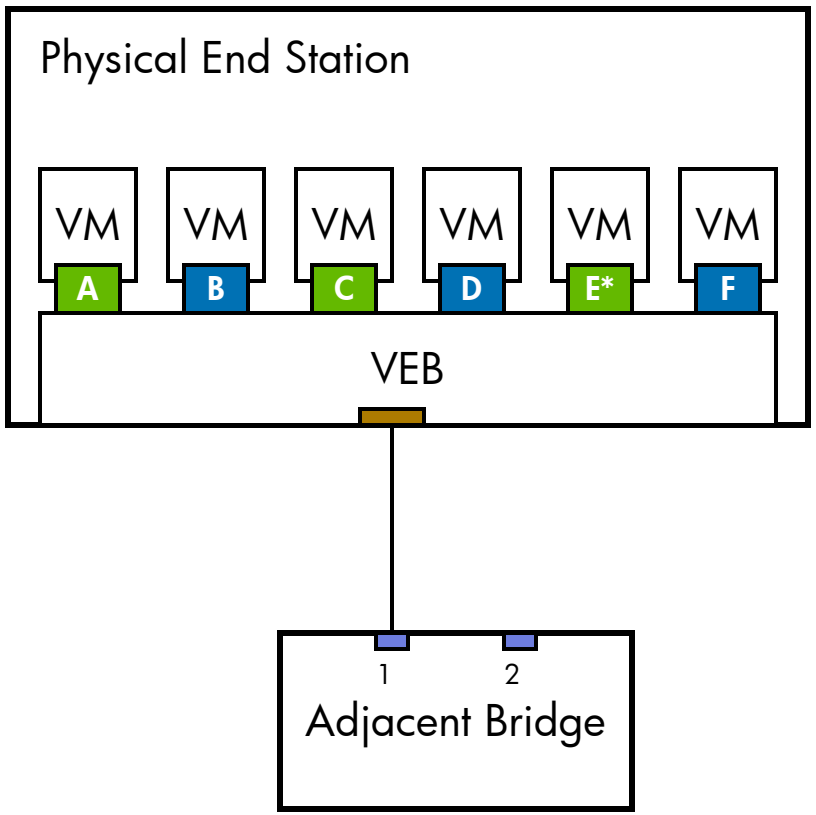
VEB Address Table

via registration

DST MAC	VLAN	Copy To (ABCDEF Up)
A	1	100000 0
B	2	010000 0
C	1	001000 0
D	2	000100 0
E	1	000010 0
F	2	000001 0

* Promiscuous vPort

VEB Address Table
Broadcast entries



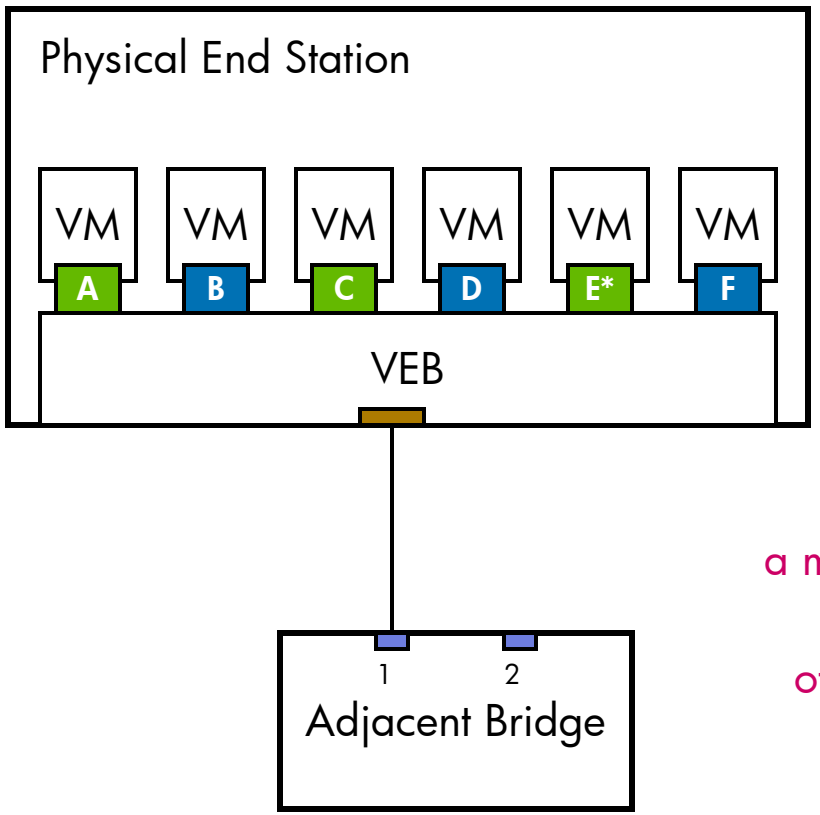
VEB Address Table

DST MAC	VLAN	Copy To (ABCDEF Up)
A	1	100000 0
B	2	010000 0
C	1	001000 0
D	2	000100 0
E	1	000010 0
F	2	000001 0
Bcast	1	101010 1
Bcast	2	010101 1

Based on VLAN ID (Port Groups)

* Promiscuous vPort

VEB Address Table
Multicast entries



VEB Address Table

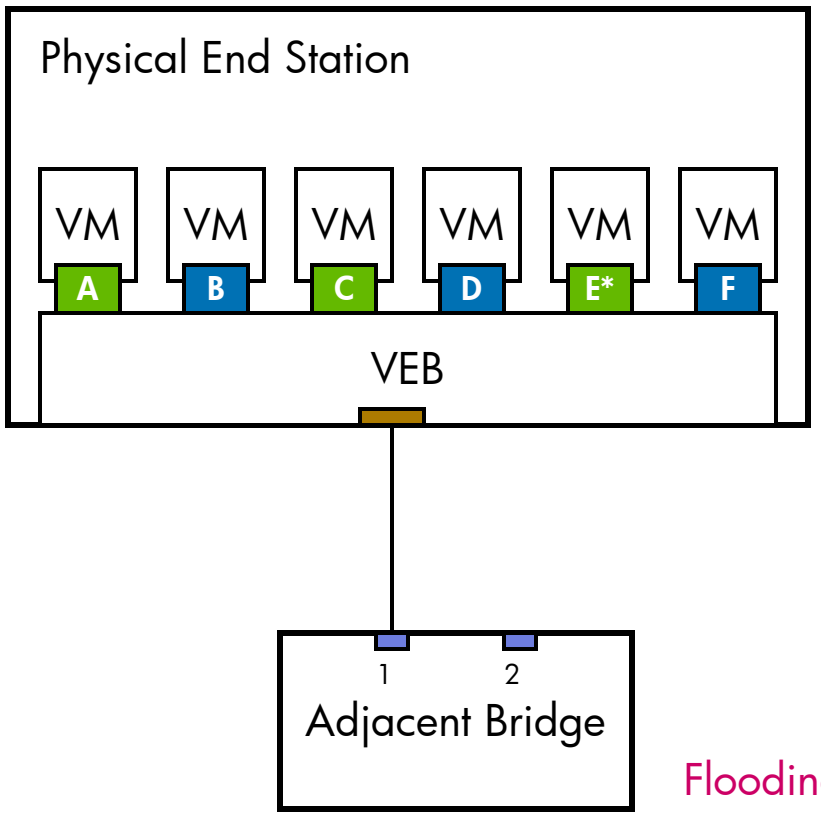
DST MAC	VLAN	Copy To (ABCDEF Up)
A	1	100000 0
B	2	010000 0
C	1	001000 0
D	2	000100 0
E	1	000010 0
F	2	000001 0
Bcast	1	101010 1
Bcast	2	010101 1
MulticastC	1	101010 1
Unk Mcast	1	100010 1
Unk Mcast	2	010101 1

C registers a multicast listen →

C avoids other multicasts →

* Promiscuous vPort

VEB Address Table
Unknown unicast entries



VEB Address Table

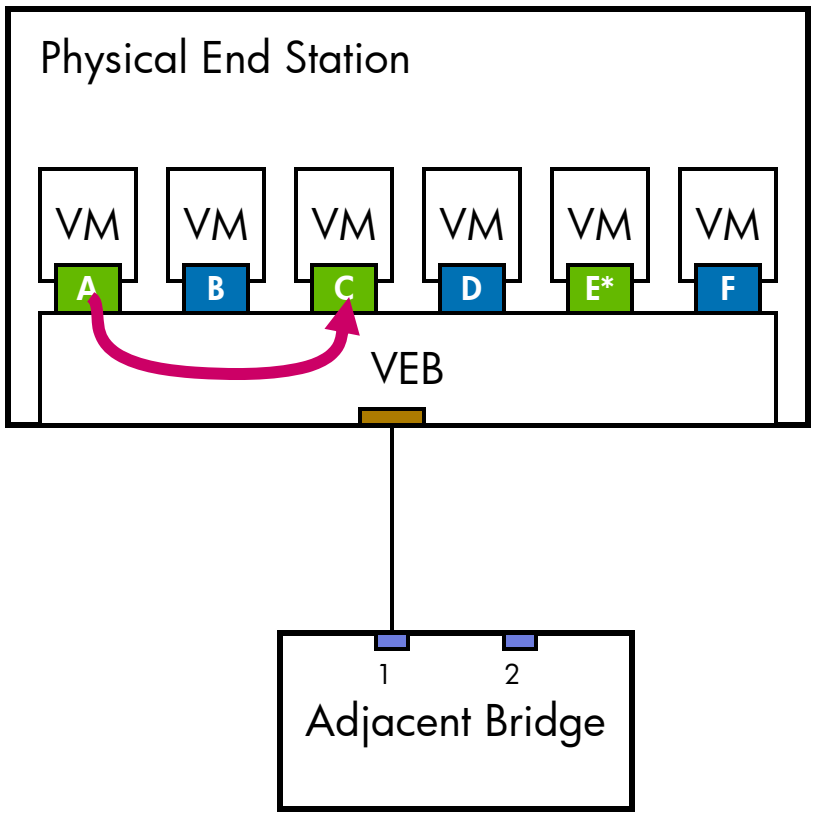
DST MAC	VLAN	Copy To (ABCDEF Up)
A	1	100000 0
B	2	010000 0
C	1	001000 0
D	2	000100 0
E	1	000010 0
F	2	000001 0
Bcast	1	101010 1
Bcast	2	010101 1
MulticastC	1	101010 1
Unk Mcast	1	100010 1
Unk Mcast	2	010101 1
Unk Ucast	1	000010 1
Unk Ucast	2	000000 1

Flooding of unknown unicast limited to promiscuous ports and uplink

* Promiscuous vPort

VEB Address Table Example
Local Unicast

SRC = A; DST = C



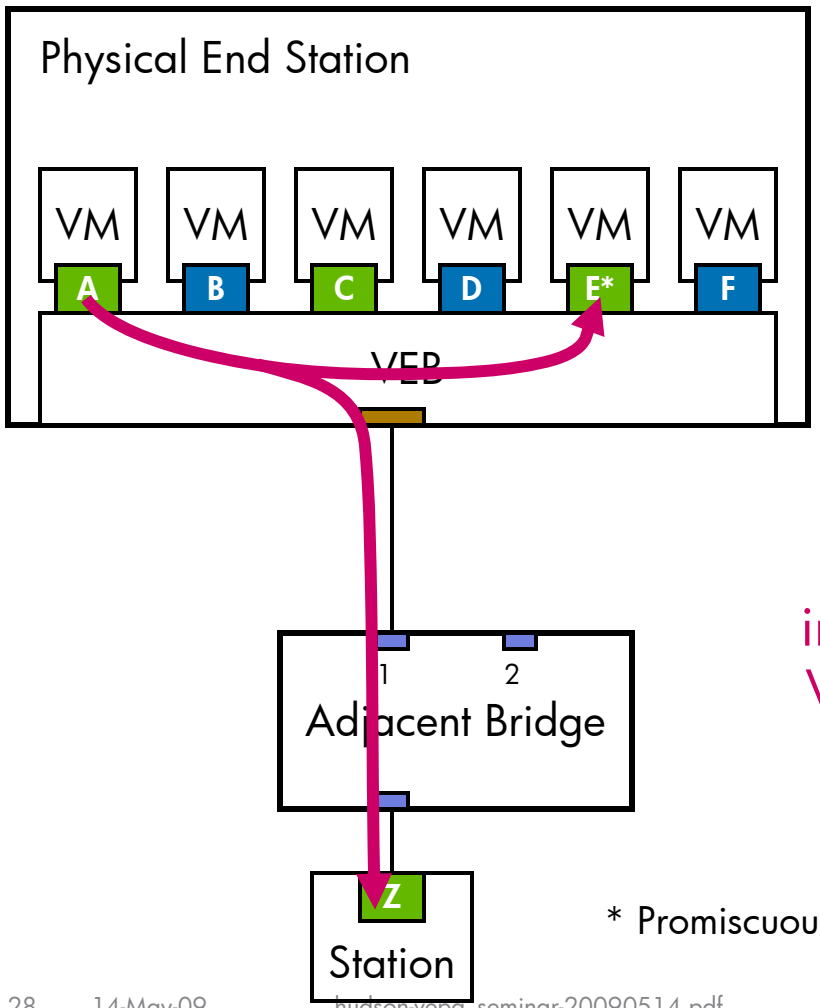
VEB Address Table

DST MAC	VLAN	Copy To (ABCDEF Up)
A	1	100000 0
B	2	010000 0
C	1	001000 0
D	2	000100 0
E	1	000010 0
F	2	000001 0
Bcast	1	101010 1
Bcast	2	010101 1
MulticastC	1	101010 1
Unk Mcast	1	100010 1
Unk Mcast	2	010101 1
Unk Ucast	1	000010 1
Unk Ucast	2	000000 1

* Promiscuous vPort

VEB Address Table Example
External Unicast

SRC = A; DST = Z



VEB Address Table

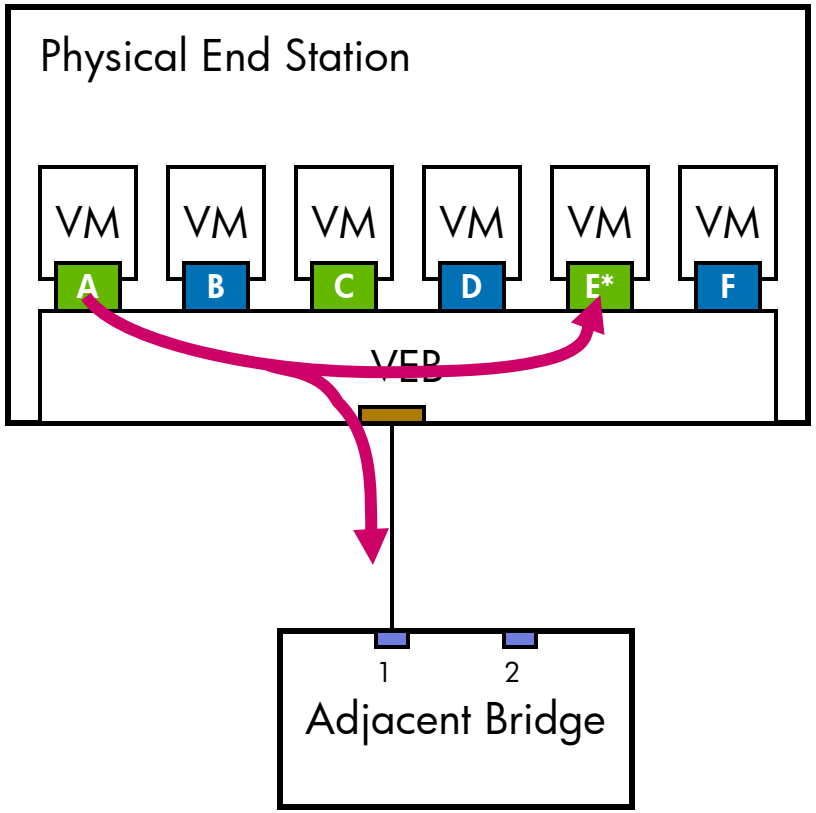
DST MAC	VLAN	Copy To (ABCDEF Up)
A	1	100000 0
B	2	010000 0
C	1	001000 0
D	2	000100 0
E	1	000010 0
F	2	000001 0
Bcast	1	101010 1
Bcast	2	010101 1
MulticastC	1	101010 1
Unk Mcast	1	100010 1
Unk Mcast	2	010101 1
Unk Ucast	1	000010 1
Unk Ucast	2	000000 1

DST Z is not in table since VEB typically doesn't do learning

* Promiscuous vPort

VEB Address Table Example
Multicast

SRC = A; DST = MulticastX, VLAN 1

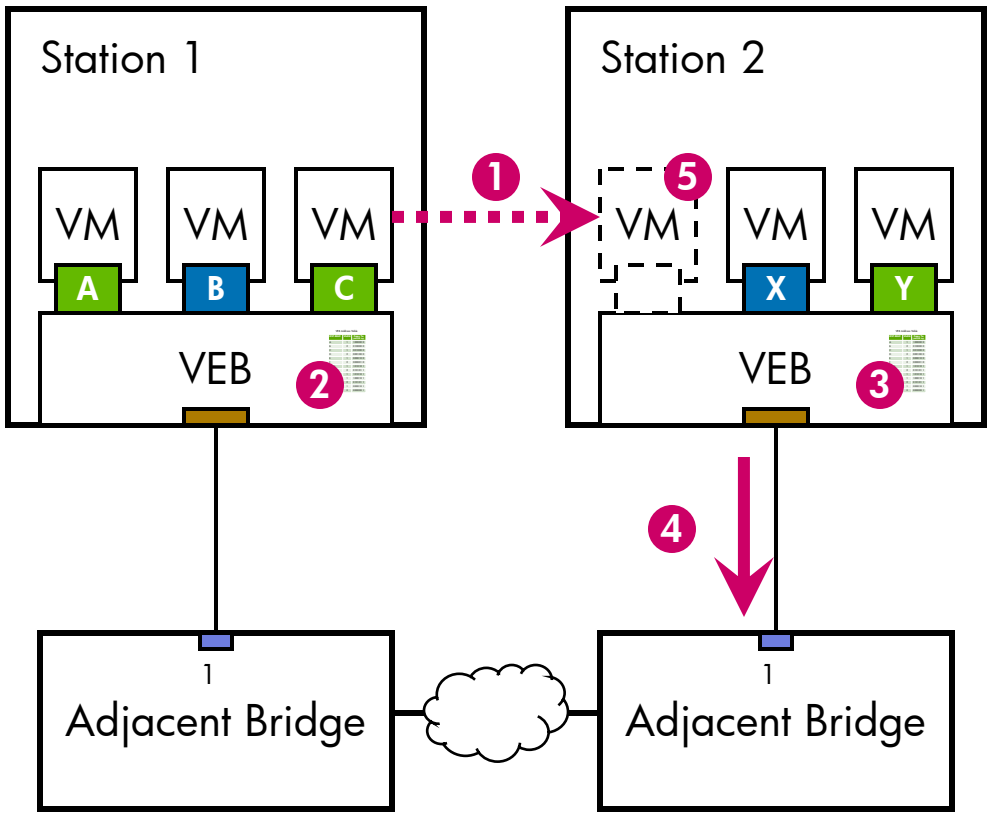


VEB Address Table

DST MAC	VLAN	Copy To (ABCDEF Up)
A	1	100000 0
B	2	010000 0
C	1	001000 0
D	2	000100 0
E	1	000010 0
F	2	000001 0
Bcast	1	101010 1
Bcast	2	010101 1
MulticastC	1	101010 1
Unk Mcast	1	100010 1
Unk Mcast	2	010101 1
Unk Ucast	1	000010 1
Unk Ucast	2	000000 1

* Promiscuous vPort

VM Migration Example



1. Hypervisors copy VM state to target destination.
2. Station 1 hypervisor halts VM & removes address table entries.
3. Station 2 hypervisor adds address table entries.
4. Station 2 hypervisor sends out gratuitous ARP to update external switch caches.
5. Station 2 activates VM.

Limitations of VEBs (today)

- Limited feature set compared to external switches
  - Limited or no packet processing (TCAMs, ACLs, etc.)
  - Limited support for security features (e.g., DHCP guard, ARP monitoring, source port filtering, dynamic ARP protection/inspection, etc.)
- Limited monitoring capabilities
  - Limited support for statistics and switch MIBs
  - No NetFlow, sFlow, rmon, port mirroring, etc.
- Limited integration with external network management systems
- Limited support for promiscuous ports (typically no learning)
- Limited support for 802.1 protocols (e.g., STP, 802.1X, LLDP)

References on Hypervisor vSwitches

- VMware
  - "VI3: Networking Concepts & Best Practices", Session #TA2441, Guy Brunsdon, vmworld 2008
  - http://vmware.com/go/networking
  - VMware Infrastructure 3 Documentation http://www.vmware.com/support/pubs/vi_pubs.html
- Microsoft
  -
- XEN
  - http://wiki.xensource.com/xenwiki/XenNetworking

Extending VEBs with Tag-less VEPA
(Virtual Ethernet Port Aggregator)

Tag-less VEPA can address most VEB limitations with minimal cost, minimal NIC changes, minimal bridge changes, no frame format changes, and minimal IEEE specification changes.

Virtual Ethernet Port Aggregator
A Definition

A Virtual Ethernet Port Aggregator (VEPA) is a capability within a physical end station that collaborates with an adjacent, external bridge to provide bridging support between multiple virtual end stations and external networks. The VEPA collaborates by forwarding all station-originated frames to the adjacent bridge for frame processing and frame relay (including 'hairpin' forwarding) and by steering and replicating frames received from the VEPA uplink to the appropriate destinations.

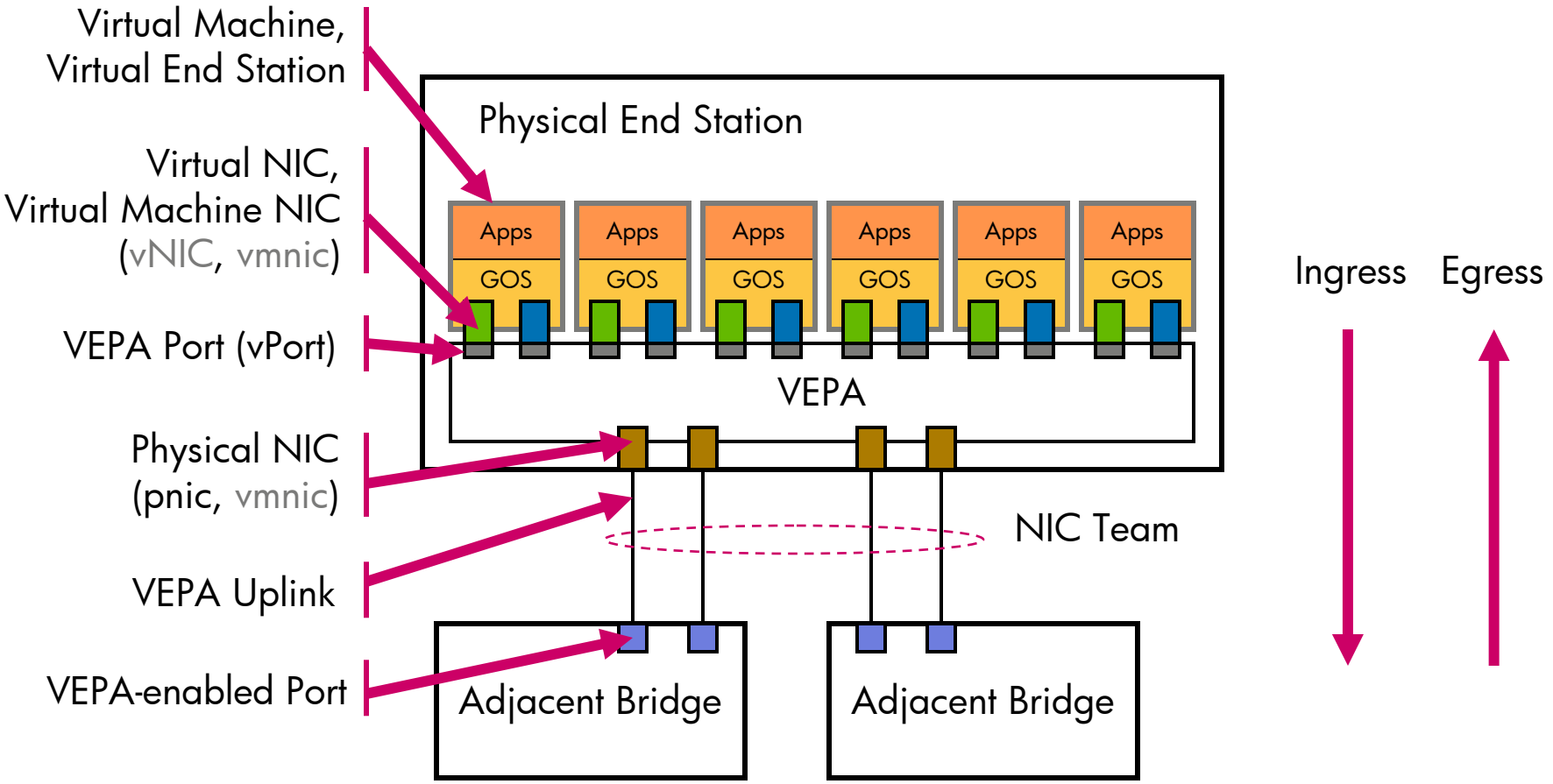
May be implemented in software or in conjunction with embedded hardware.

Note: As with the case of VEBs, VEPAs have access to vNIC configuration information that normally not available to an 802.1Q bridge.

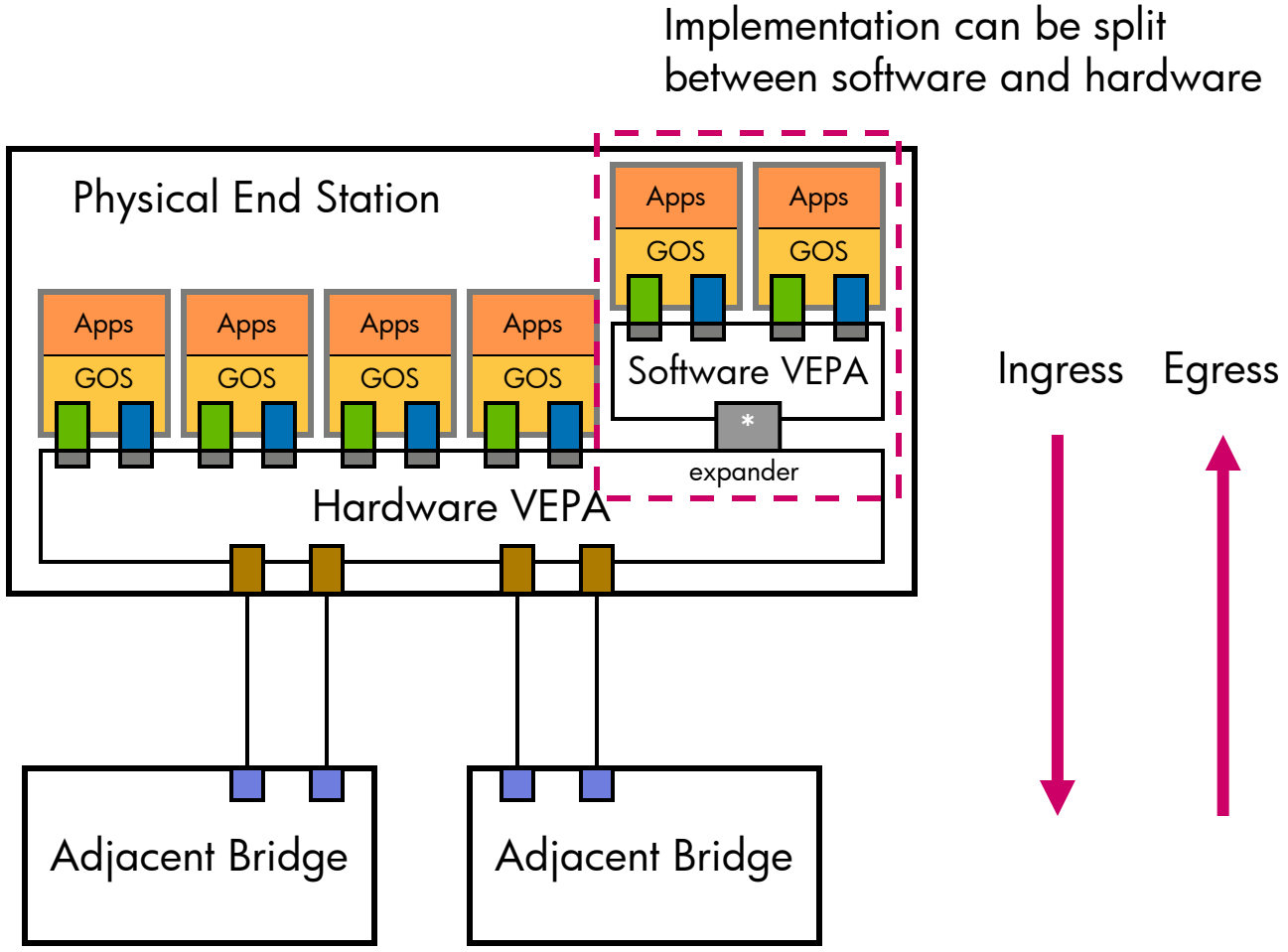
Benefits VEPA adds to VEB

- Gains access to external switch features
  - Packet processing (TCAMs, ACLs, etc.)
  - Security features such as: DHCP guard, ARP monitoring, source port filtering, dynamic ARP protection/inspection, etc.
- Enhances monitoring capabilities
  - Statistics
  - NetFlow, sFlow, rmon, port mirroring, etc.

Basic VEPA Anatomy and Terms

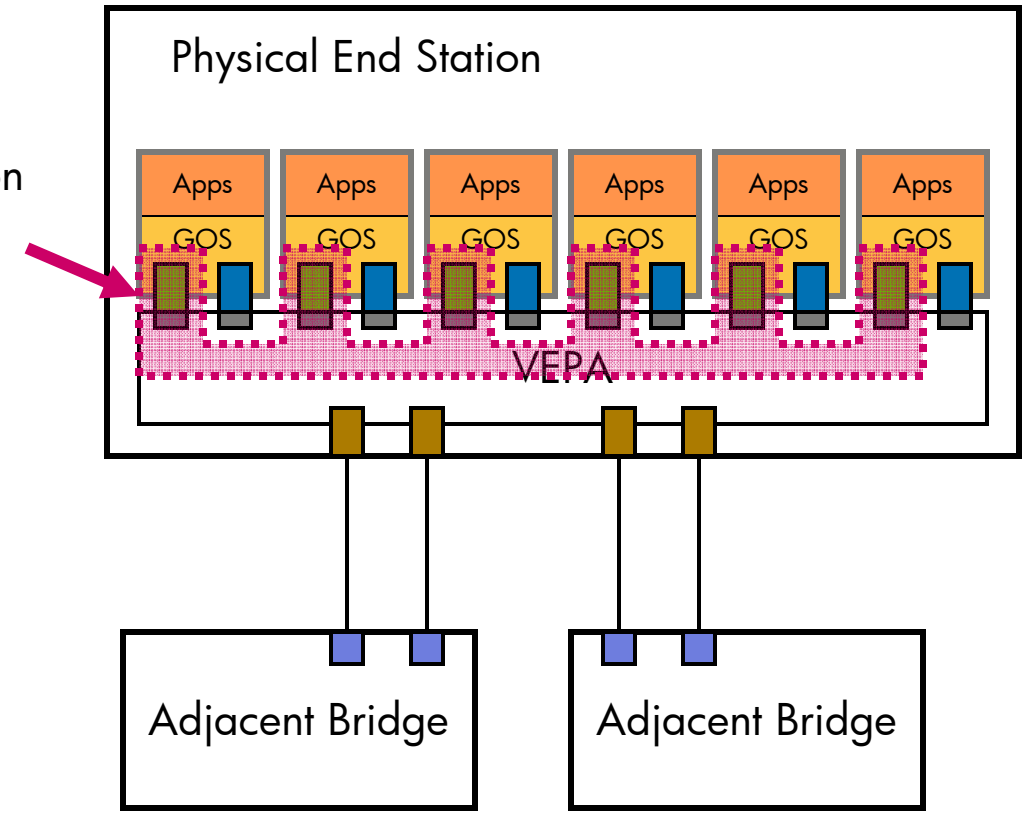


Basic VEPA Anatomy and Terms
Showing Hardware + Software VEPA



Basic VEPA Anatomy and Terms
Showing Port Group

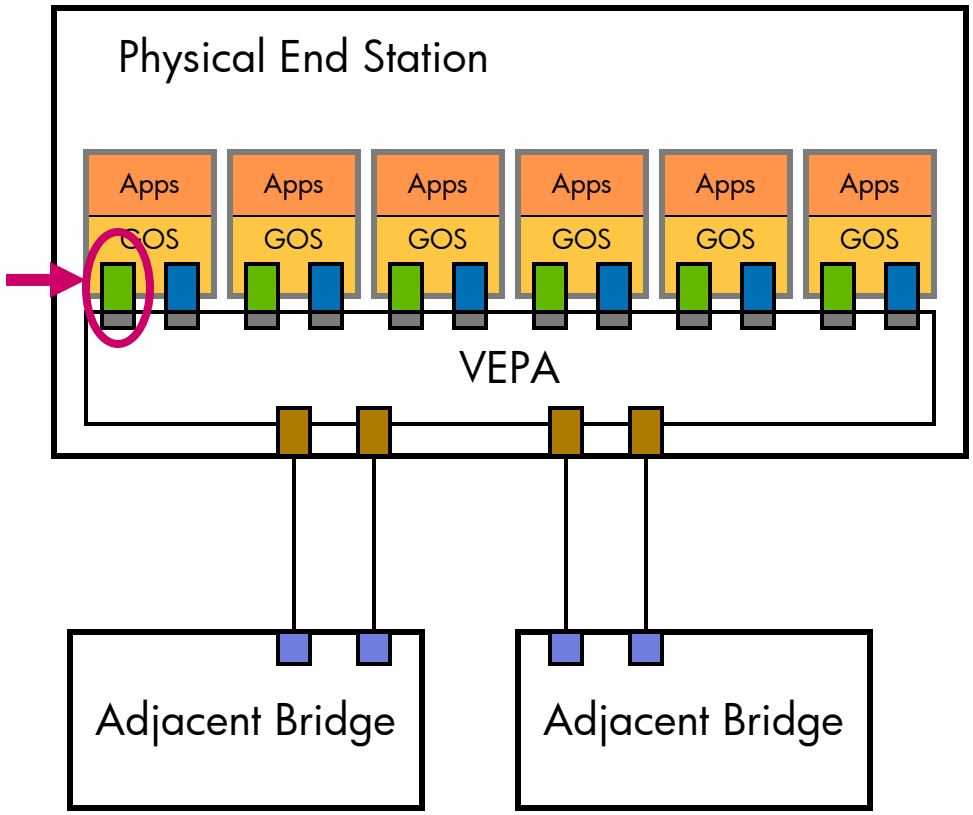
Port Group
A set of vPorts with similar configuration (such as VLAN ID)



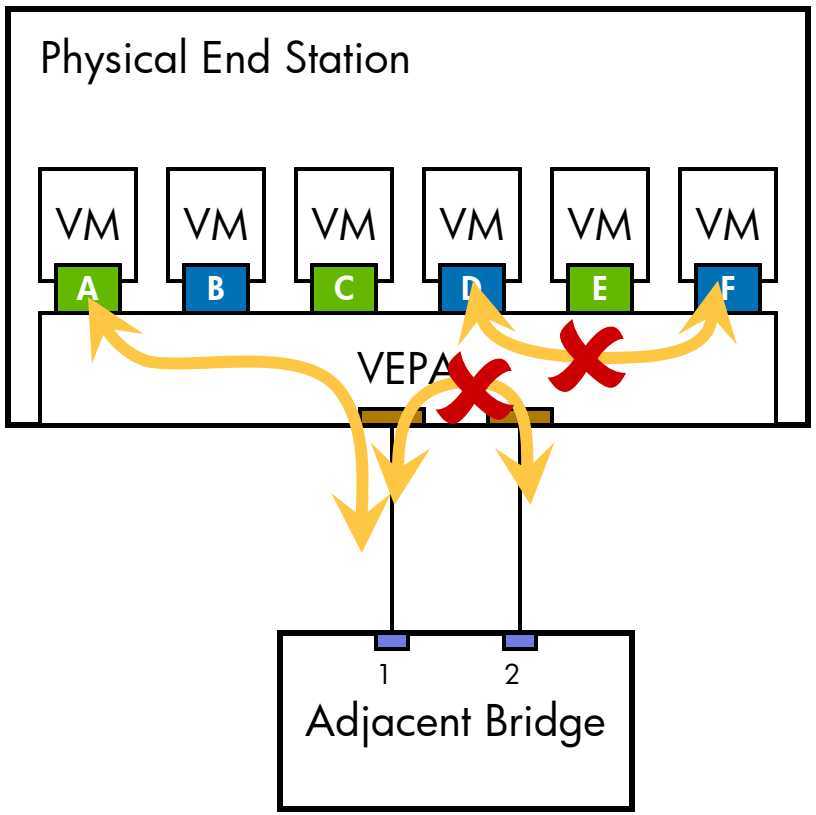
Basic VEPA Anatomy and Terms
Key constraint for basic tag-less VEPA mode…

vNICs in basic tag-less VEPA mode are NOT configured for promiscuous operation

This will be addressed in a later section



VEPA Loop-free Forwarding Behavior



- Forwards based on MAC address (and port group or VLAN)
- Forwards
  - VM ←→ Uplink
- Never from VM to VM
- Does NOT forward from uplink to uplink
  - Single active logical uplink
  - Multiple uplinks may be 'teamed' (802.3ad and other algorithms)
- Does not participate in (or affect) spanning tree

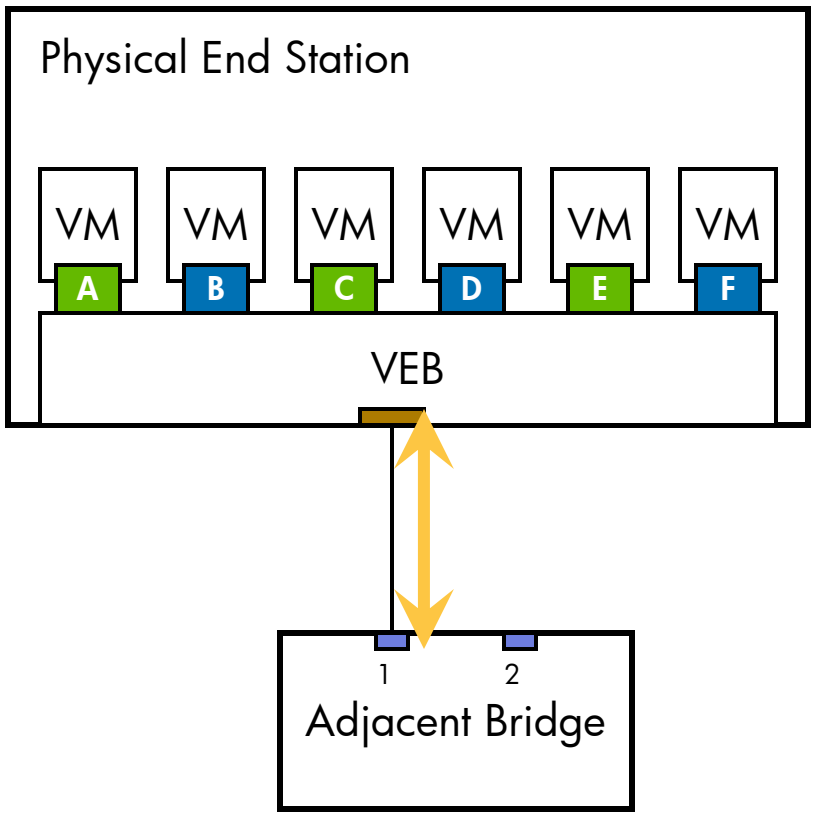
Mgmt & Config of VEPA vPorts
is very similar to Mgmt & Config of VEB vPorts

- General VLAN mode
  - VEB/vSwitch can terminate or pass VIDs through to VM
  - Affects all ports in VEB
- Per-port VLAN Settings
  - vPort VLAN ID
  - egress VLAN IDs (VIDs that can reach the port)
- Addressing Security
  - Limit to assigned MAC
  - Allow guest-OS specific MACs
  - ~~No promiscuous for basic tag-less VEPA~~
- Default priority and/or priority mapping
- Traffic shaping & bandwidth management

Mgmt & Config of VEPA Uplinks
is very similar to Mgmt & Config of VEB Uplinks

- General VLAN mode
  - VEB/vSwitch can terminate or pass VIDs through to VM
  - Affects all ports in VEB
- Uplinks (NICs) associated with a VEB
- NIC Teaming Mode
  - Fail-over
  - Transmit load-balancing
  - Bi-direction load-balancing (802.3ad, etc.)
- DCBX Configuration
  - ETS Queues
  - Priority Flow Control
- EVB Mode (VEB/VEPA)

LLDP and DCBX
is very similar to VEB

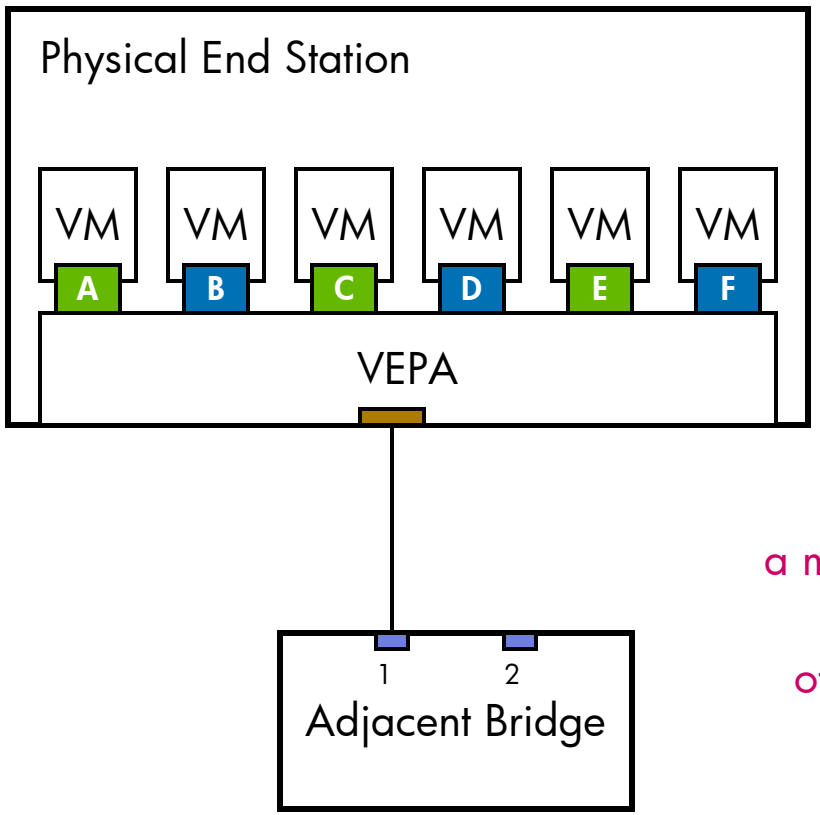


- LLDP & DCB are between
  - VEB uplink and
  - Adjacent bridge port
  - Uses MAC of physical NIC
- LLDP
  - Identifies physical NIC of the physical end station
- DCBX
  - Configures the physical NIC
  - Physical NIC ETS queues
  - Physical NIC PFC settings
  - Select EVB Mode (VEB/VEPA)
- vNICs typically implement a single (lossless) queue

VEPA Address Table Management
is like VEB Address Table Management

- VEPAs typically don't do learning
  - Intended to be at the edge of the network, not the middle
- Instead, MAC addresses can be known by registration
  - Hypervisors set vNIC default MAC address
  - Hypervisors can intercept when a guest OS sets receive filters on a vNIC
    - Locally Administered Address (LAA)
    - Multicast addresses
- VEPA Address Table entries
  - Provide the receive filtering for the vNICs
  - Provides multicast filtering without IGMP snooping

VEPA Address Table
Showing MAC, multicast, & unknown unicast



VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
A	1	100000
B	2	010000
C	1	001000
D	2	000100
E	1	000010
F	2	000001
Bcast	1	101010
Bcast	2	010101
MulticastC	1	101010
Unk Mcast	1	100010
Unk Mcast	2	010101
Unk Ucast	1	000000
Unk Ucast	2	000000

via registration

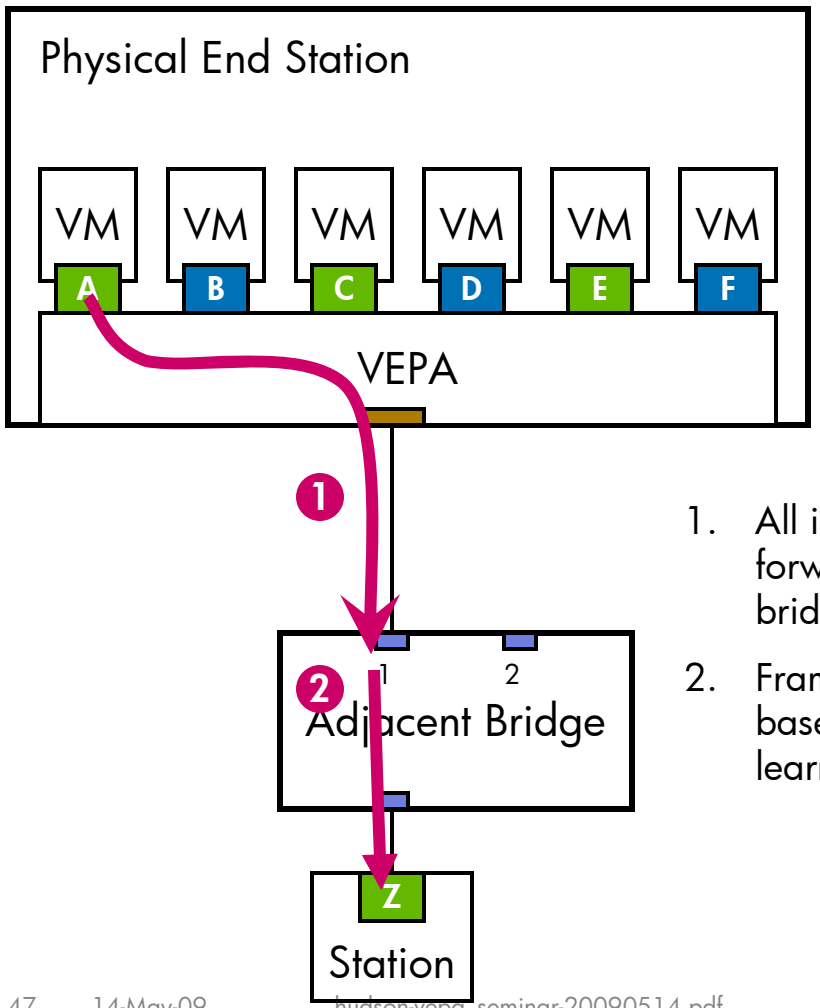
C registers a multicast listen

C avoids other multicasts

This example assume no promiscuous ports

Basic VEPA Operation
Unicast to external address

SRC = A; DST = Z



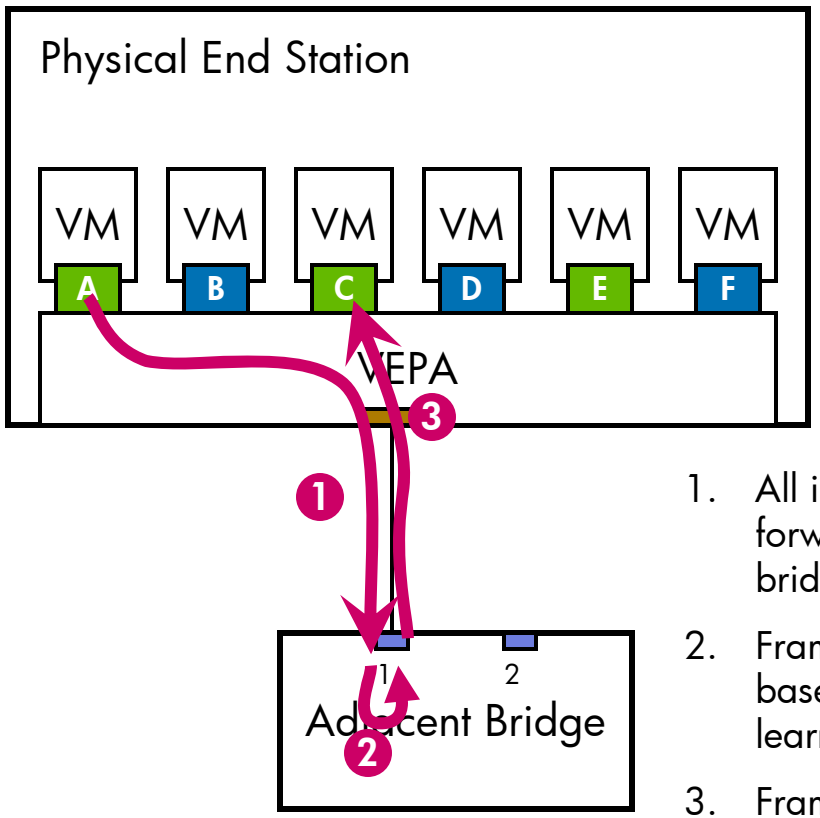
1. All ingress frames forwarded to adjacent bridge
2. Frame forwarded based on adj. bridge learning.

VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
A	1	100000
B	2	010000
C	1	001000
D	2	000100
E	1	000010
F	2	000001
Bcast	1	101010
Bcast	2	010101
MulticastC	1	101010
Unk Mcast	1	100010
Unk Mcast	2	010101
Unk Ucast	1	000000
Unk Ucast	2	000000

Basic VEPA Operation
Unicast to local address

SRC = A; DST = C



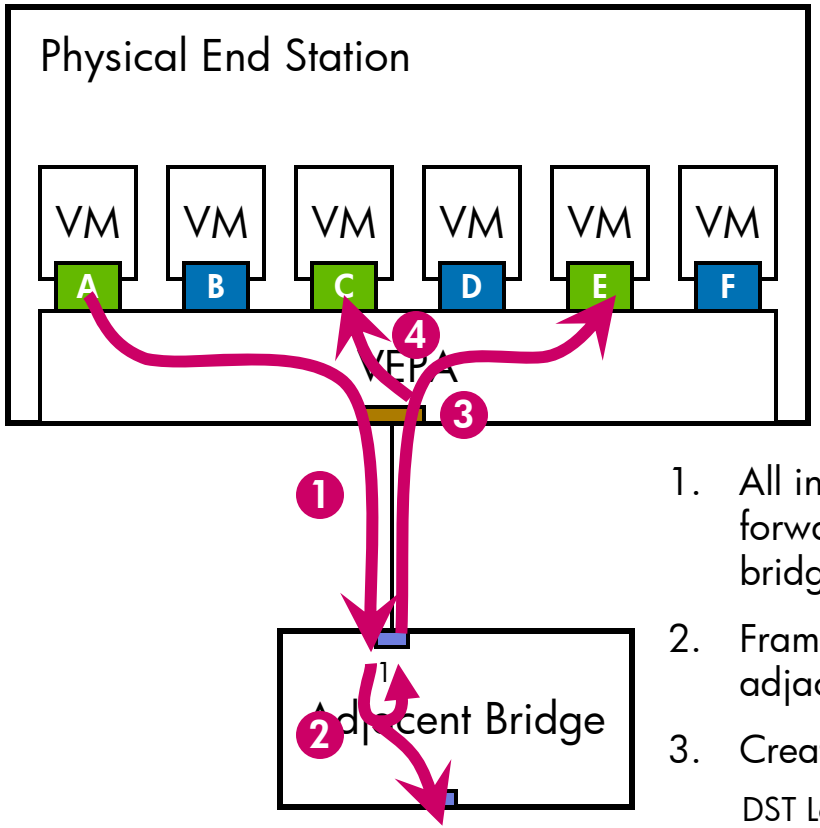
1. All ingress frames forwarded to adjacent bridge
2. Frame forwarded based on adj. bridge learning.
3. Frame forwarded based on delivery mask generated from VEPA address table

VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
A	1	100000
B	2	010000
C	1	001000
D	2	000100
E	1	000010
F	2	000001
Bcast	1	101010
Bcast	2	010101
MulticastC	1	101010
Unk Mcast	1	100010
Unk Mcast	2	010101
Unk Ucast	1	000000
Unk Ucast	2	000000

Basic VEPA Operation
Multicast

SRC = A; DST = MulticastC



1. All ingress frames forwarded to adjacent bridge
2. Frame forwarded by adjacent bridge.
3. Create delivery mask

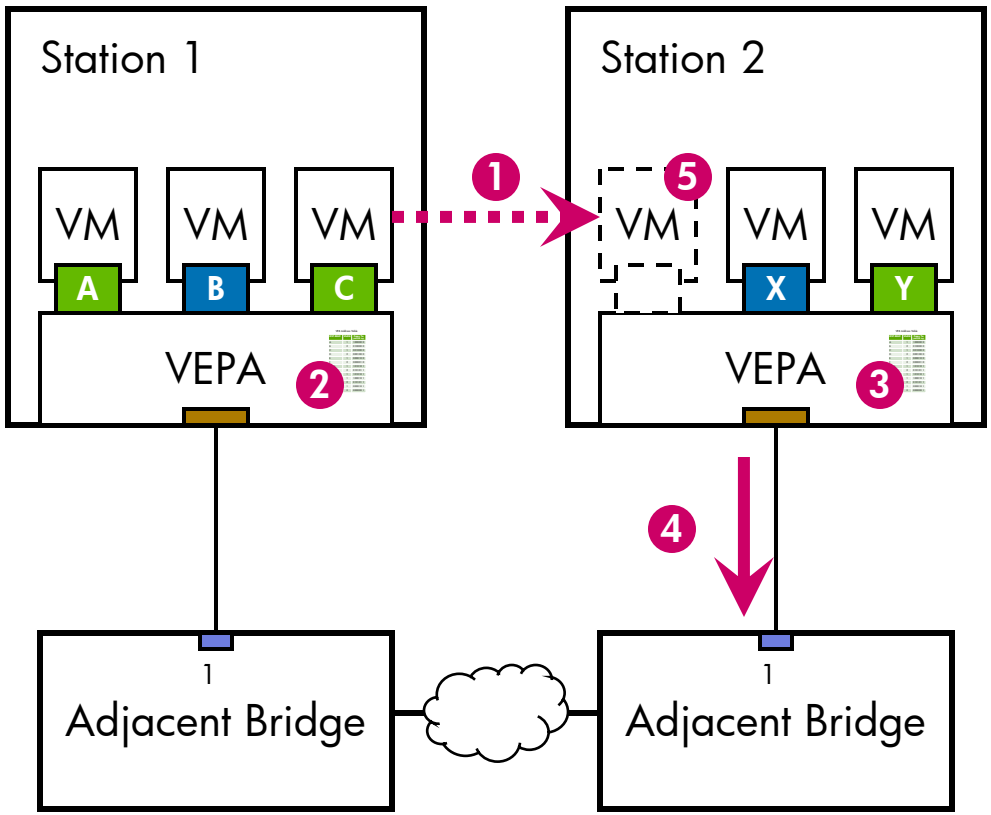
   DST Lookup = 101010
   SRC Lookup = 100000
   Delivery Mask = 001010

4. Deliver Frame Copies

VEPA Address Table

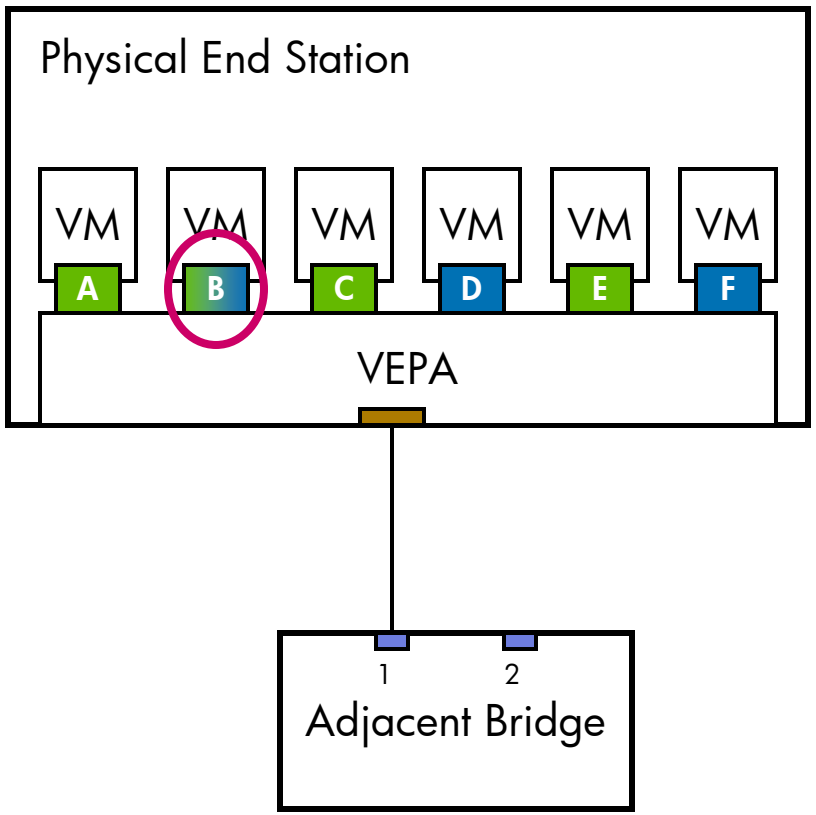
DST MAC	VLAN	Copy To (ABCDEF)
A	1	100000
B	2	010000
C	1	001000
D	2	000100
E	1	000010
F	2	000001
Bcast	1	101010
Bcast	2	010101
MulticastC	1	101010
Unk Mcast	1	100010
Unk Mcast	2	010101
Unk Ucast	1	000000
Unk Ucast	2	000000

VM Migration Example



1. Hypervisors copy VM state to target destination.
2. Station 1 hypervisor halts VM & removes address table entries.
3. Station 2 hypervisor adds address table entries.
4. Station 2 hypervisor notifies adjacent bridge of EVB configuration change & sends out gratuitous ARP to update external switch caches.
5. Station 2 activates VM.

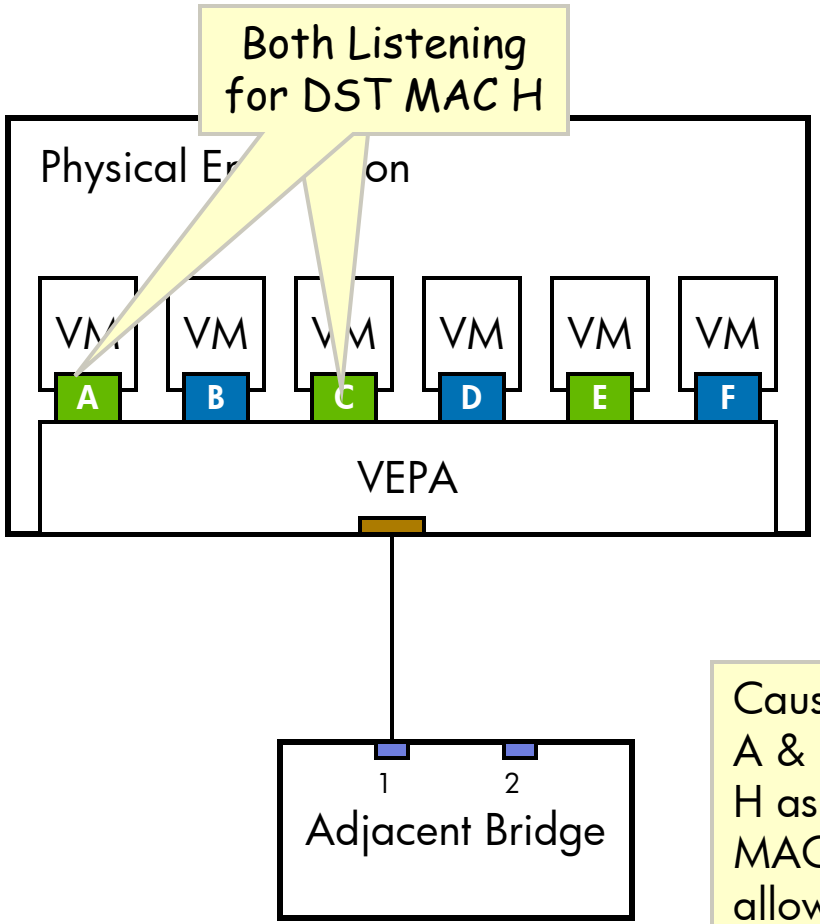
VEPA Address Table
vPort on multiple VLANs



VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
A	1	100000
B	1,2	010000
C	1	001000
D	2	000100
E	1	000010
F	2	000001
Bcast	1	111010
Bcast	2	010101
MulticastC	1	111010
Unk Mcast	1	110010
Unk Mcast	2	010101
Unk Ucast	1	000000
Unk Ucast	2	000000

VEPA Address Table
vPorts in Dual Listening Mode



Caused by VMs A & C registering H as a listening MAC address (if allowed by hypervisor)

VEPA Address Table

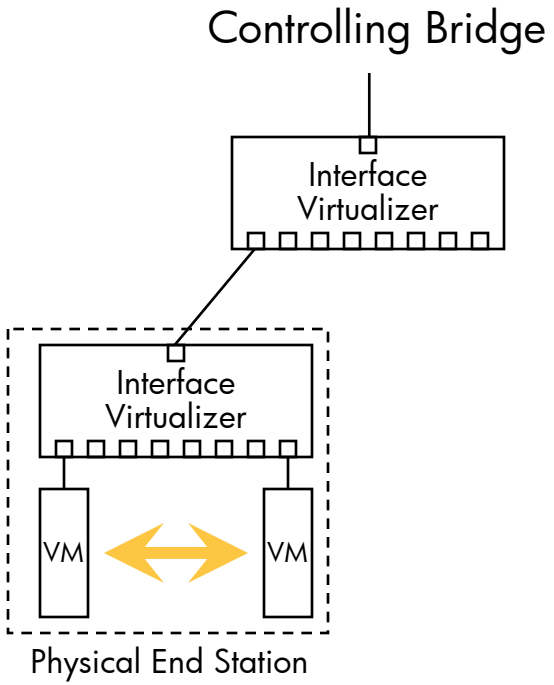
DST MAC	VLAN	Copy To (ABCDEF)
A	1	100000
B	2	010000
C	1	001000
D	2	000100
E	1	000010
F	2	000001
H	1	101000
Bcast	1	101010
Bcast	2	010101
MulticastC	1	101010
Unk Mcast	1	100010
Unk Mcast	2	010101
Unk Ucast	1	000000
Unk Ucast	2	000000

Comparing EVB Approaches

Benefits of VEB/VEPA Solution

- Simple extension to VEB
  - Similar port configuration
  - Similar address table
  - Minor changes to frame forwarding behavior
- Solves nearly all of the issues with VEBs
- Allows easy migration between VEB and VEPA modes
  - Could allow simultaneous operation of VEB and VEPA
- Requires minimal extension to 802.1Q
  - Configuration of hair-pin mode
- Can be implemented in many existing switches with a firmware update

Issues with VN-Tag: Performance Choke Point



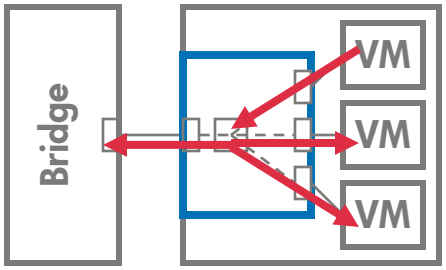
- Total switching capacity of
  - N physical end stations and
  - N*M virtual servers

  is constrained to the speed of the uplink to the controlling bridge.
- Introduces additional latency.
- Breaks distributed computing approaches that can exploit physical proximity.

Issues with VN-Tag Approach

- Using multiple layers of VN-Tag network concentrators…
  - Significantly limit the network cross-sectional bandwidth
  - Often increases the number of links traversed
  - Increases congestion
- Constrains innovations in distributed computing
  - Blocks advantages of locality in distributed systems
    - Distributed storage solutions, nearby caching servers, etc.
  - Blocks benefits of increased end-station capabilities over time
- VN-Tags increases hardware complexity to end stations
  - Significantly different than already-required VEB
  - New forwarding and frame replication mechanisms
- VN-Tags require significant new standards efforts
  - New tag format
  - Management of remote frame replication
- VN-Tags will not work with any switch not specifically designed for it

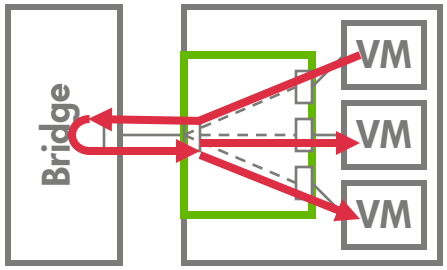
Summary of Edge Virtual Bridging (EVB) Approaches



Virtual Ethernet Bridge (VEB)

uses MAC+VID to steer frames

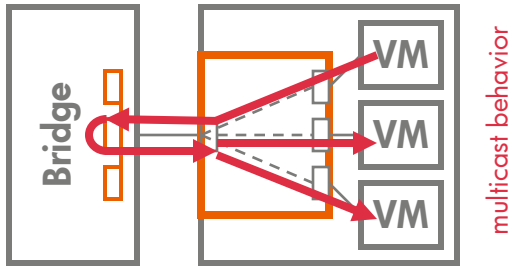
- Emulates 802.1 Bridge
- Works with all existing bridges
- No changes to existing frame format.
- Limited bridge visibility
- Limited feature set
- Best performance.
- Will always be there



Tag-less VEPA

uses MAC+VID to steer frames

- Exploits 802.1 Bridge
- Works with many existing bridges
- No changes to existing frame format.
- Full bridge visibility
- Access to bridge features
- Constrained performance
- Leverages VEB



VN-Tagged

uses new tag to steer frames

- Extends 802.1 Bridge
- Works with few or no existing bridges
- Changes existing frame format.
- Full bridge visibility
- Access to bridge features
- Constrained performance
- Doesn't leverage VEB

Call For Action

- IEEE 802.1 standardization of
  - Switch port operation when in 'hairpin' mode
  - Configuration of 'hairpin' mode
    - LLDP/DCBX capabilities exchange & configuration
    - Managed object definition
- Industry Standardization of EVB management
  - Coordinated configuration of vPort settings
  - Mechanism & standards forum is still TBD
- Join the Edge Virtual Bridging Ad Hoc
  - http://tech.groups.yahoo.com/group/evb/
  - Conference Calls Tuesdays 1PM Central US

Agenda

- Life on the Edge
- VEBs are here to stay
- Extending VEBs with Tag-less VEPAs
- Comparing EVB Approaches
- Using Promiscuous vPorts with VEPA
- Other 'Case Studies'
- Conclusion

Advanced Topics: Supporting Promiscuous vPorts

Why Promiscuous Ports Matter…

- Promiscuous ports are not common at the edge of the virtualization environment… However,

- Simultaneous operation of a VEB and VEPA provides both performance and flexibility
- A small number of inline virtual appliances may be useful
- The VN-Tag alternative believes it is necessary

Approaches for handling promiscuous vPorts

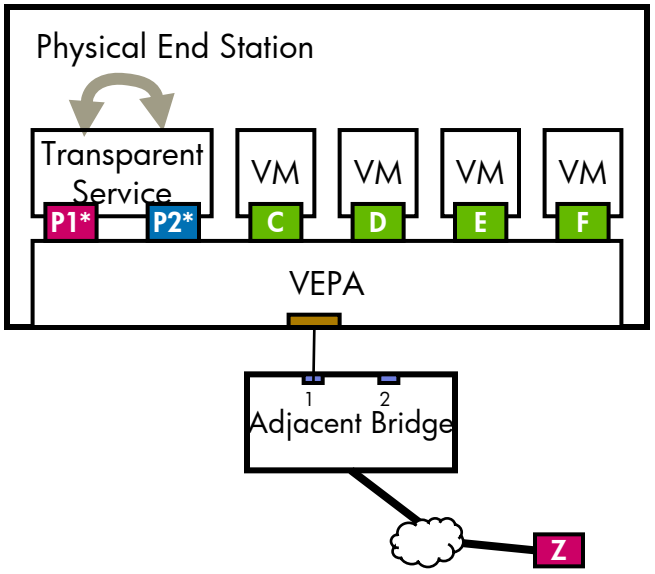
- Use a VEB
- Use security APIs instead
  - Transparent services accessed through vNIC extensions or hypervisor APIs
- Have the VEPA learn
  - Not really practical since the VEPA must have a complete address table to filter properly and needs assistance from adjacent bridge
- Use VLANs to isolate promiscuous ports
  - Use private-VLAN-like isolation
  - Requires hairpin mode on a per VLAN basis on the adjacent bridge
  - VLAN translation is helpful

VLAN Isolation Approach
Summary

- Use a set of VLANs internal to the VEPA and adjacent bridge to create proper filtering behavior
- Promiscuous vPorts send on a different set of VLANs than they receive on
- Non-promiscuous vPorts send and receive on a set of VLANs and may also receive on promiscuous vPort VLANs
- The adjacent bridge enables “hairpin” mode on some VLANs, but not others

VEPA Promiscuous Mode Example
Address Table & Port Setup

- Non-promiscuous vPorts may egress internal VLANs.
- Internal Promiscuous vPort PVID and Egress list are different
- External Promiscuous vPort is isolated
- Broadcast table entries are created based on port Egress list



Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

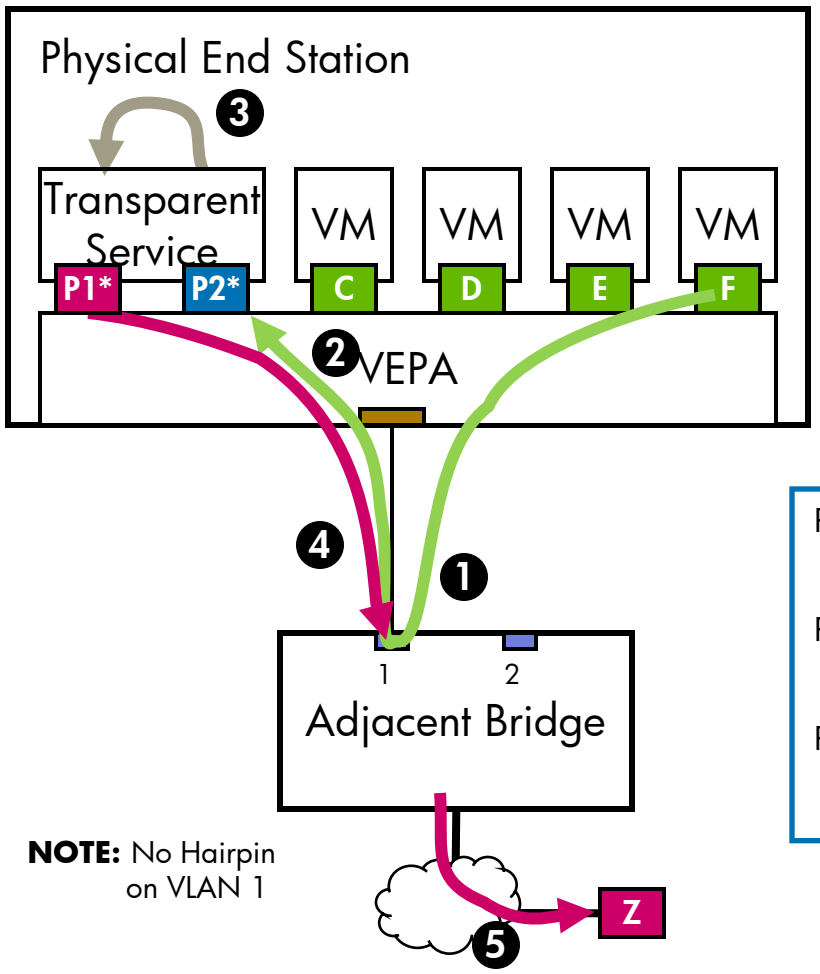
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Unicast

SRC = F; DST = Z



VLAN 1
VLAN 2*
VLAN 3*
* Internal VLAN

Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

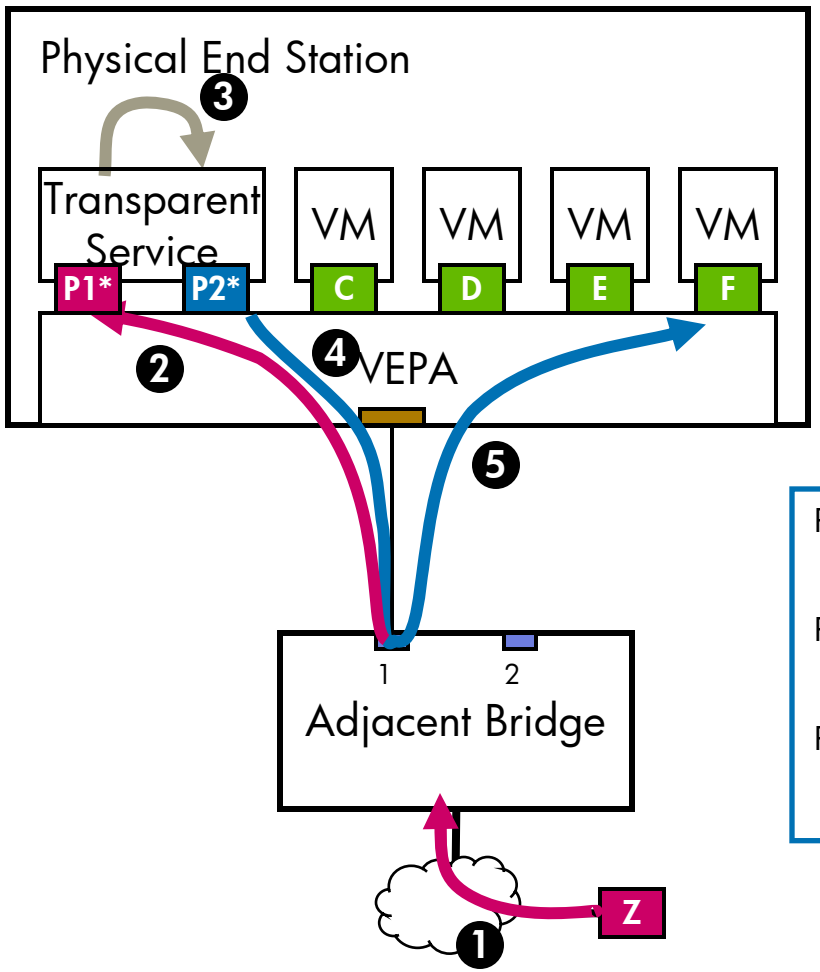
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Unicast

SRC = Z; DST = F



VLAN 1
VLAN 2*
VLAN 3*
* Internal VLAN

Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

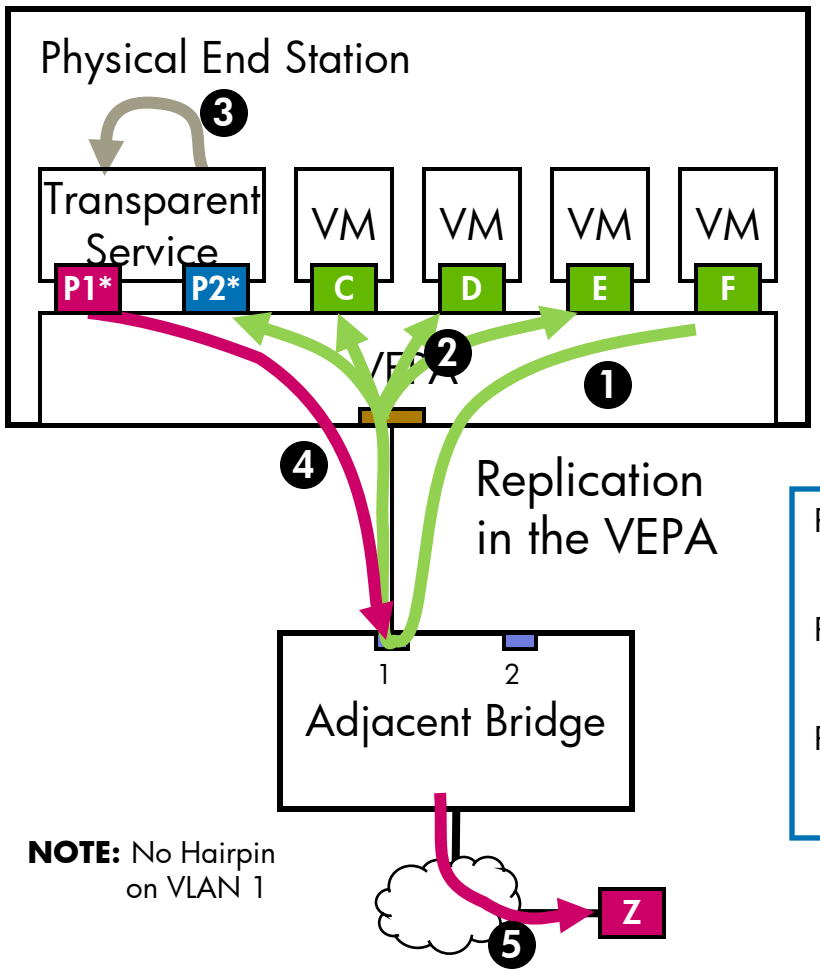
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Broadcast

SRC = F; DST = Bcast



VLAN 1
VLAN 2*
VLAN 3*
* Internal VLAN

Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

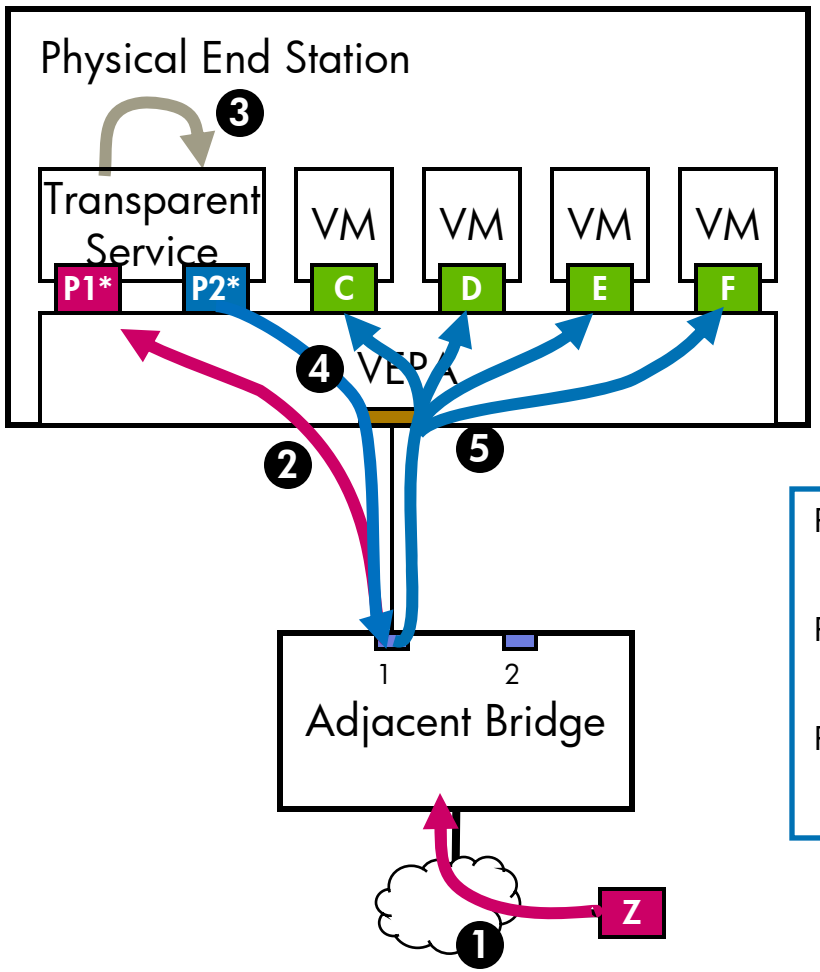
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Broadcast

SRC = Z; DST = Bcast



VLAN 1
VLAN 2*
VLAN 3*
* Internal VLAN

Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

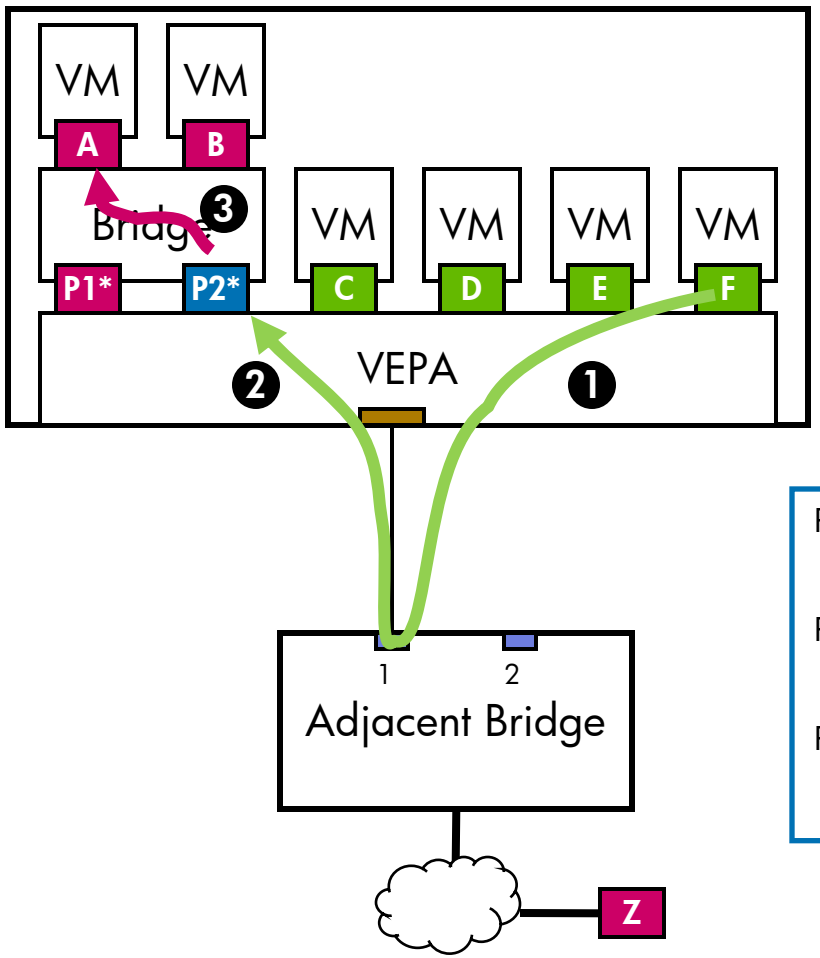
VEPA Address Table

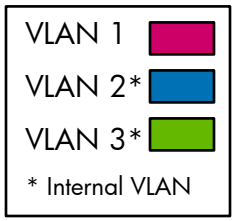
DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Collocated Bridge

SRC = F; DST = A





Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

VEPA Address Table

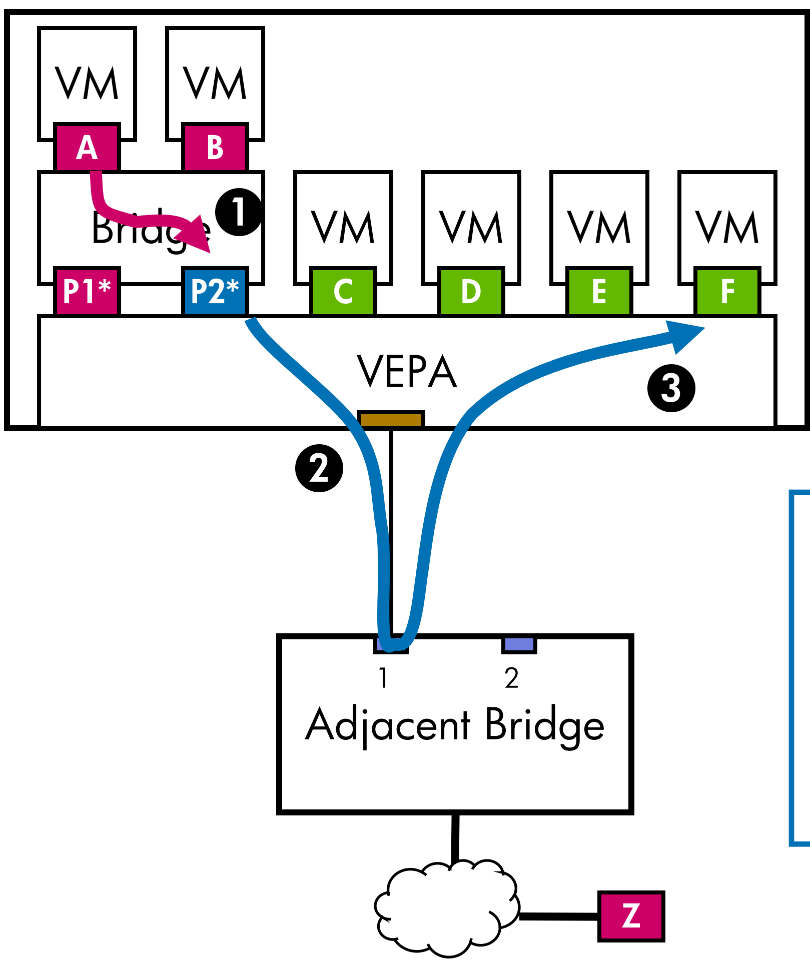
DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example

Collocated Bridge

SRC = A; DST = F



VLAN 1
VLAN 2*
VLAN 3*
* Internal VLAN

Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

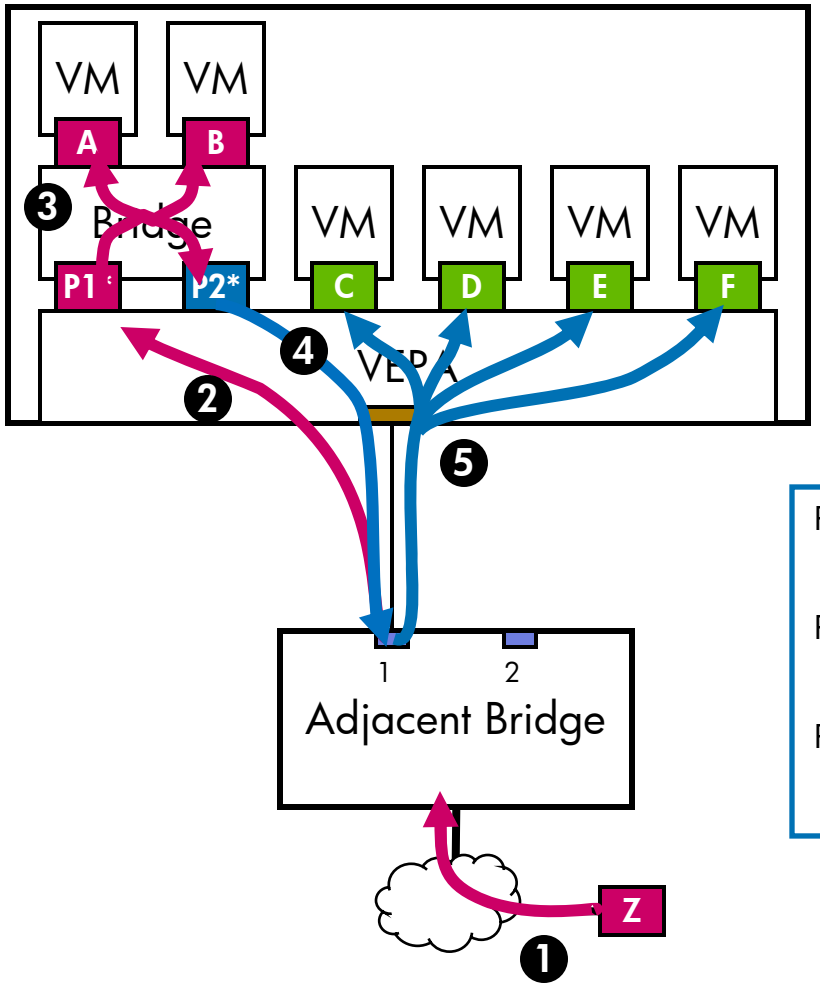
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Collocated Bridge

SRC = Z; DST = Bcast



VLAN 1
VLAN 2*
VLAN 3*
* Internal VLAN

Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

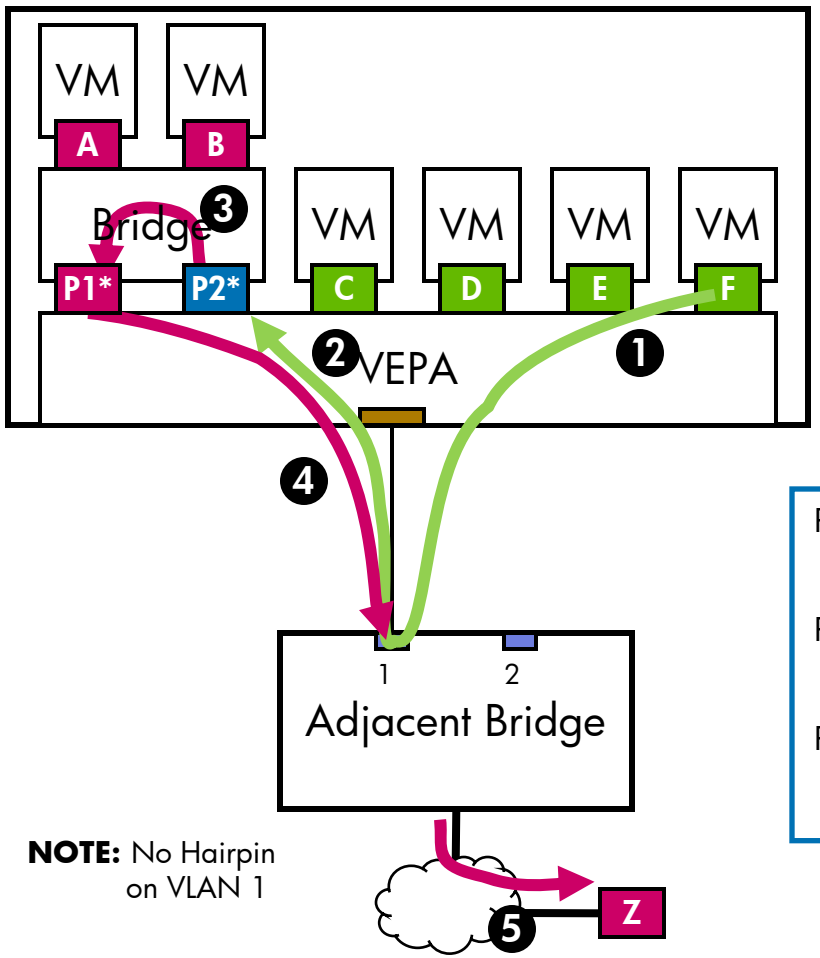
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Collocated Bridge

SRC = F; DST = Z



VLAN 1
VLAN 2*
VLAN 3*
* Internal VLAN

Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

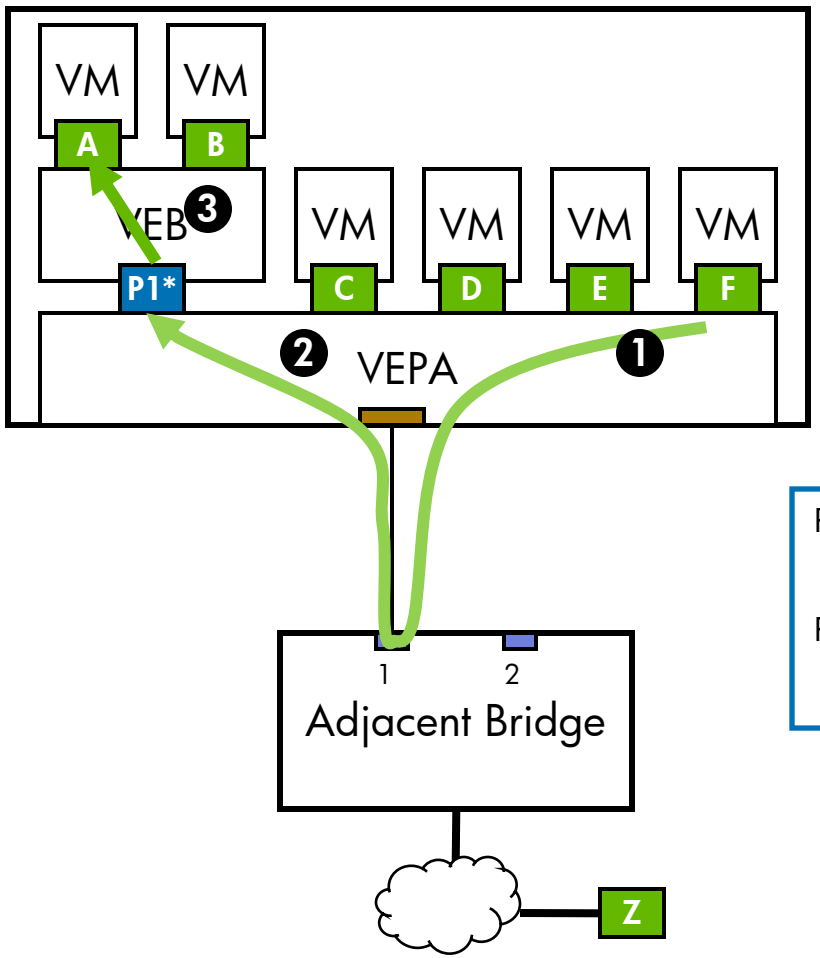
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Collocated VEB

SRC = F; DST = A



VLAN 1
VLAN 2*
VLAN 3
* Internal VLAN

Port P1 *
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

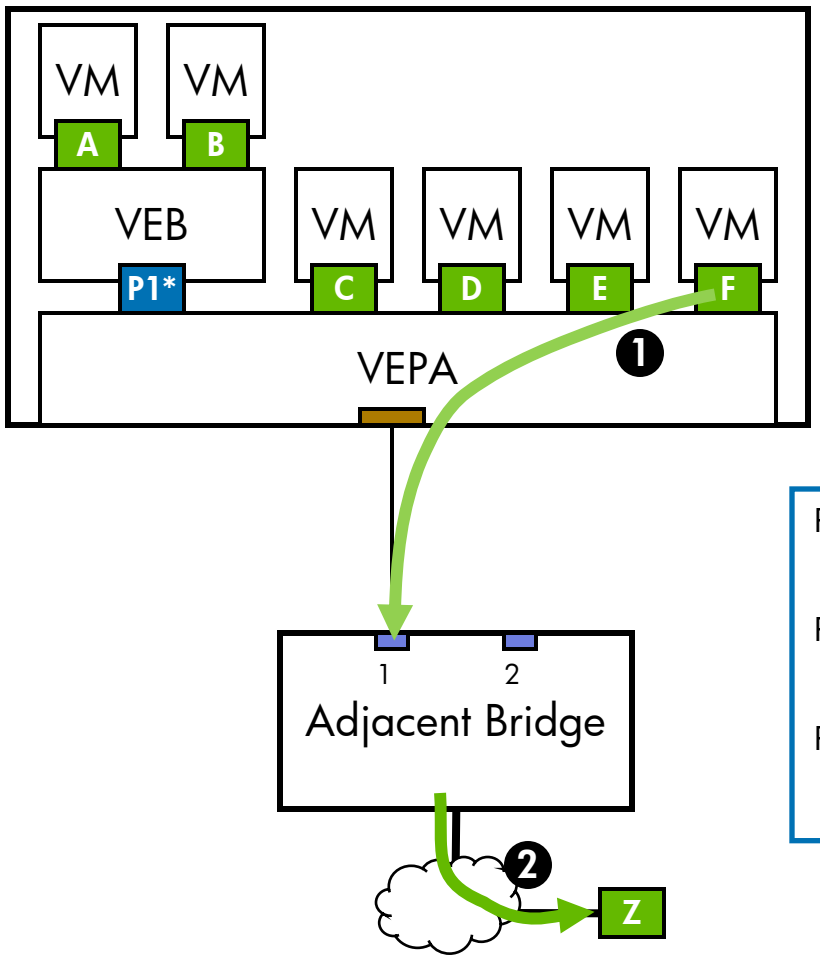
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	100000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Collocated VEB

SRC = F; DST = Z



VLAN 1
VLAN 2*
VLAN 3
* Internal VLAN

Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

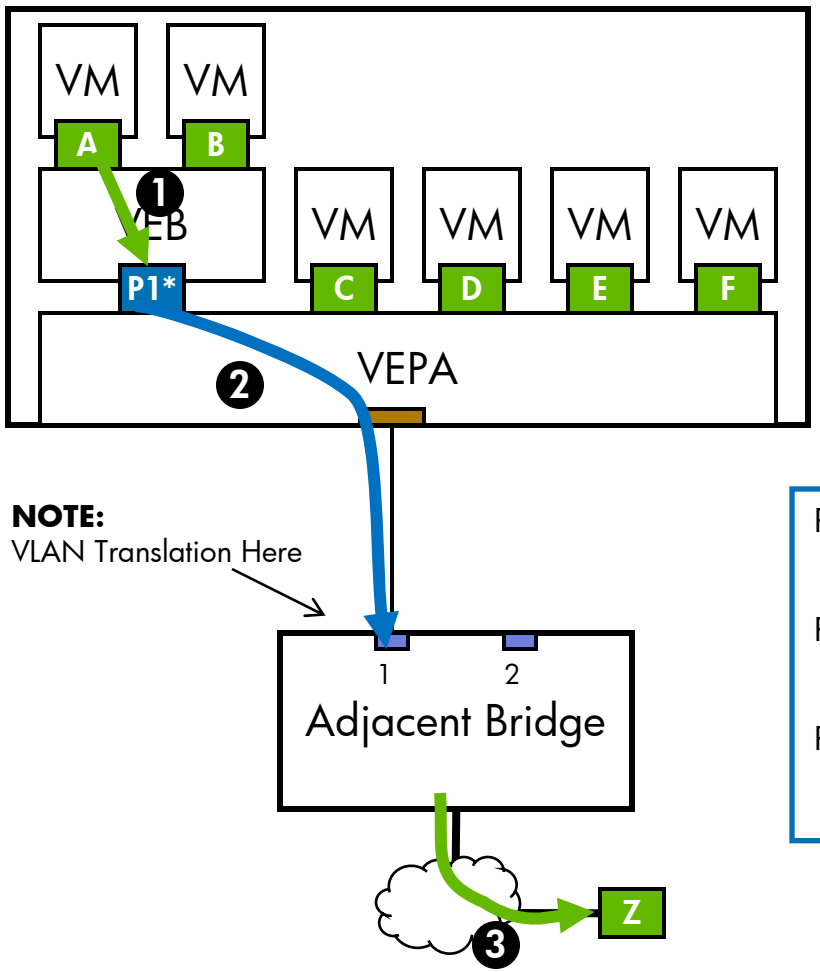
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Collocated VEB

SRC = A; DST = Z



Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

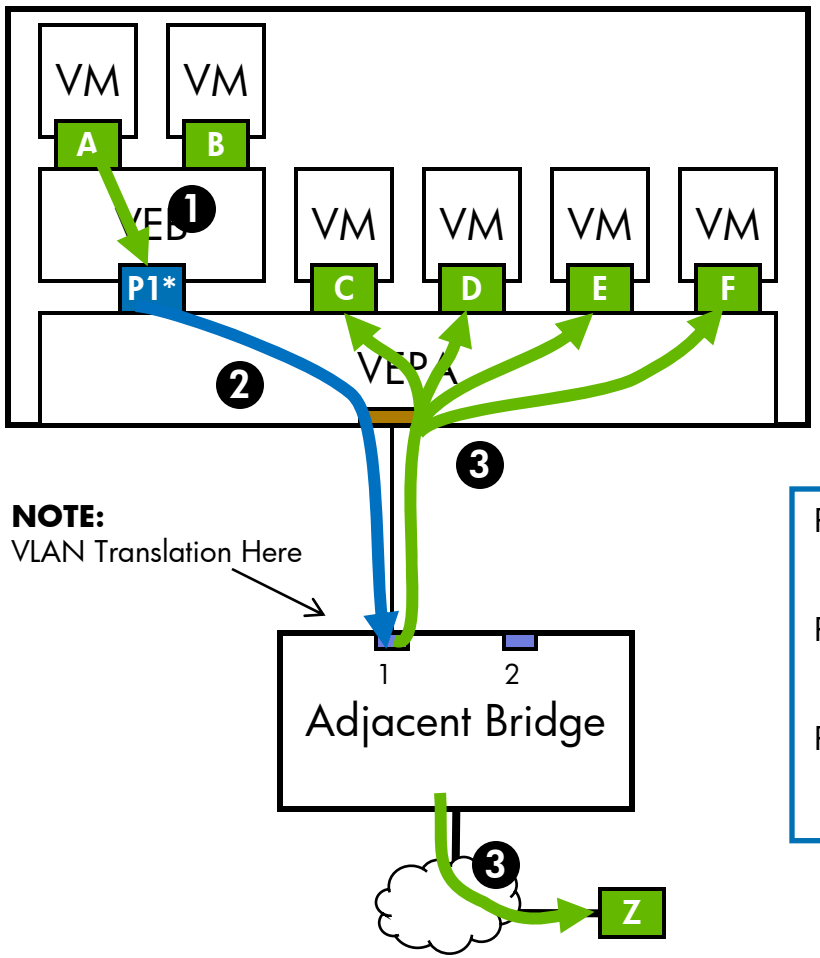
VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

VEPA Promiscuous Mode Example
Collocated VEB

SRC = A; DST = Bcast



VLAN 1
VLAN 2*
VLAN 3
* Internal VLAN

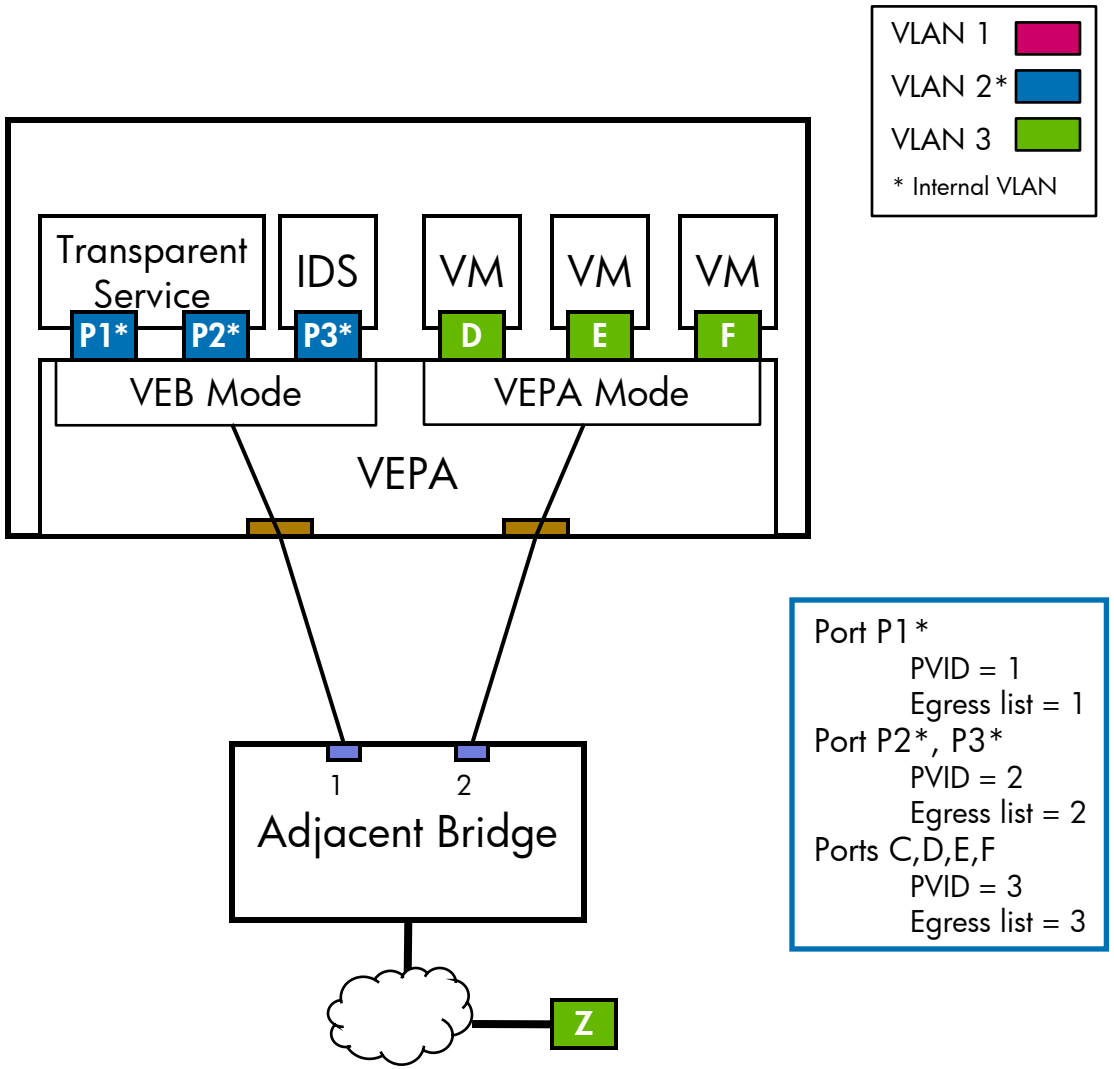
Port P1*
PVID = 1
Egress list = 1
Port P2*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

VEPA Address Table

DST MAC	VLAN	Copy To (ABCDEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	100000
Bcast	2	001111
Bcast	3	011111
Unk Ucast	1	100000
Unk Ucast	2	000000
Unk Ucast	3	010000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	Y	00UUUU T
3	Y	0U0000 T

Combined VEPA/VEB Example
Multiple Transparent Services



Port P1*
PVID = 1
Egress list = 1
Port P2*, P3*
PVID = 2
Egress list = 2
Ports C,D,E,F
PVID = 3
Egress list = 3

VEPA Address Table

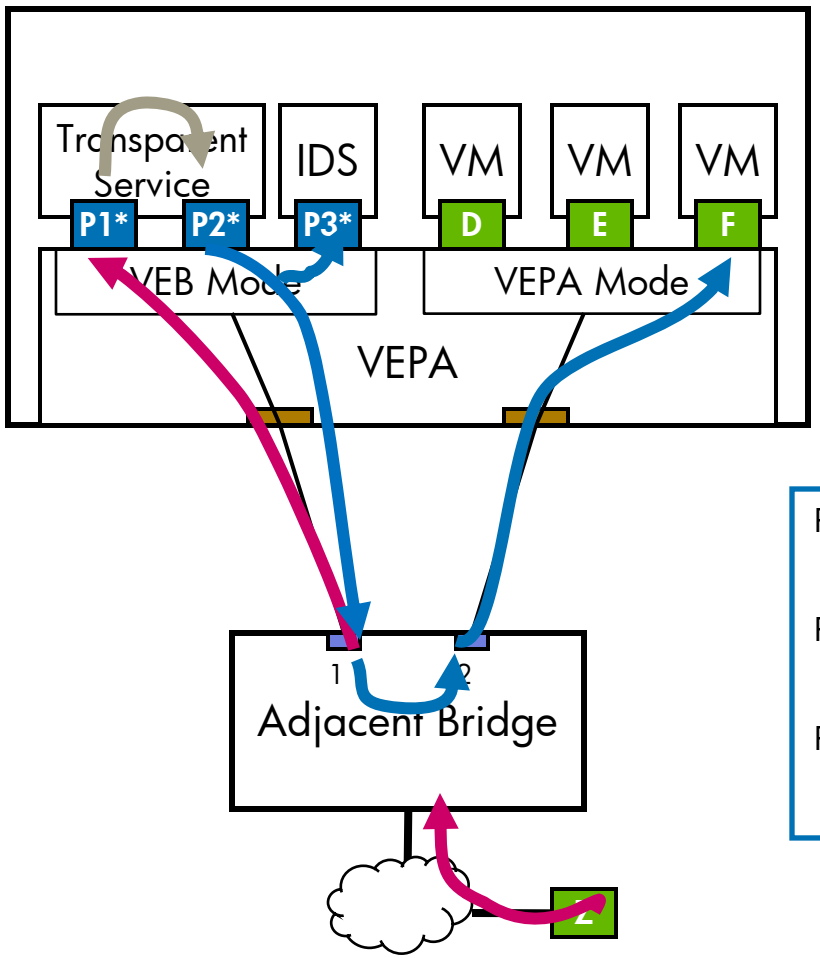
DST MAC	VLAN	Copy To (123DEF)
C	3	001000
D	3	000100
E	3	000010
F	3	000001
Bcast	1	100000
Bcast	2	011000
Bcast	3	000111
Unk Ucast	1	100000
Unk Ucast	2	011000
Unk Ucast	3	000000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	N	0UU000 T
3	Y	000UUU T

Combined VEPA/VEB Example

Multiple Transparent Services

SRC = Z; DST = F



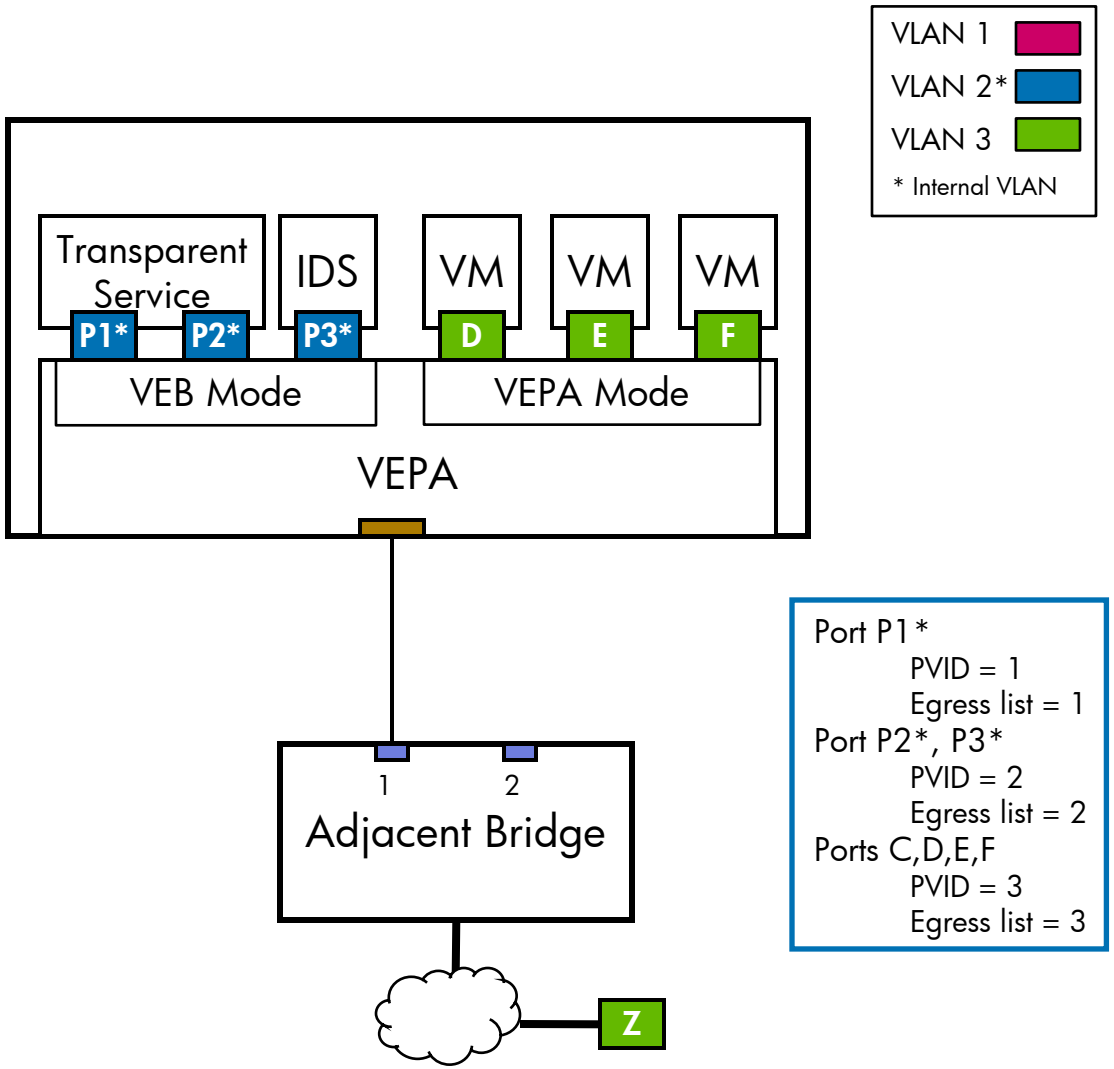
Port P1*
PVID = 1
Egress list = 1
Port P2*, P3*
PVID = 2
Egress list = 3
Ports C,D,E,F
PVID = 3
Egress list = 2,3

VEPA Address Table

DST MAC	VLAN	Copy To (123DEF)
C	2,3	001000
D	2,3	000100
E	2,3	000010
F	2,3	000001
Bcast	1	000000
Bcast	2	000111
Bcast	3	000111
Unk Ucast	1	000000
Unk Ucast	2	000000
Unk Ucast	3	000000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	N	0UU000 T
3	Y	000UUU T

Combined VEPA/VEB Example
Multiple Transparent Services



Port P1*
PVID = 1
Egress list = 1
Port P2*, P3*
PVID = 2
Egress list = 2
Ports C,D,E,F
PVID = 3
Egress list = 3

VEPA Address Table

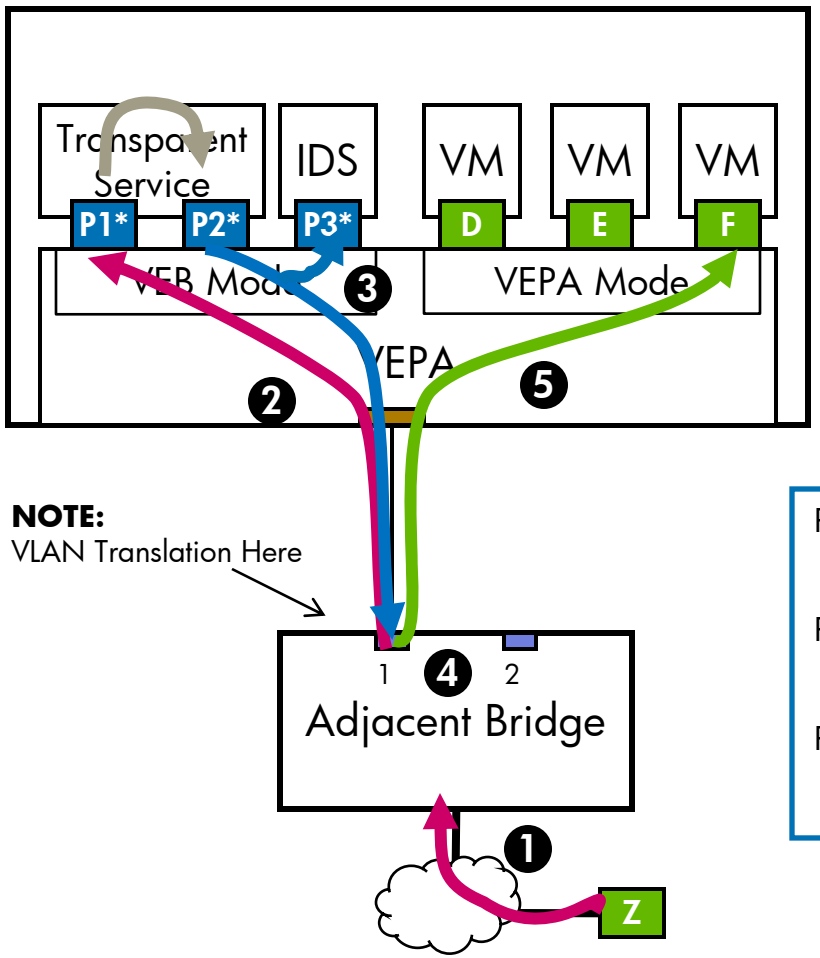
DST MAC	VLAN	Copy To (123DEF)
C	3	001000
D	3	000100
E	3	000010
F	3	000001
Bcast	1	100000
Bcast	2	011000
Bcast	3	000111
Unk Ucast	1	100000
Unk Ucast	2	011000
Unk Ucast	3	000000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	N	0UU000 T
3	Y	000UUU T

Combined VEPA/VEB Example

Multiple Transparent Services

SRC = Z; DST = F



Port P1*
PVID = 1
Egress list = 1
Port P2*, P3*
PVID = 2
Egress list = 2
Ports C,D,E,F
PVID = 3
Egress list = 3

VEPA Address Table

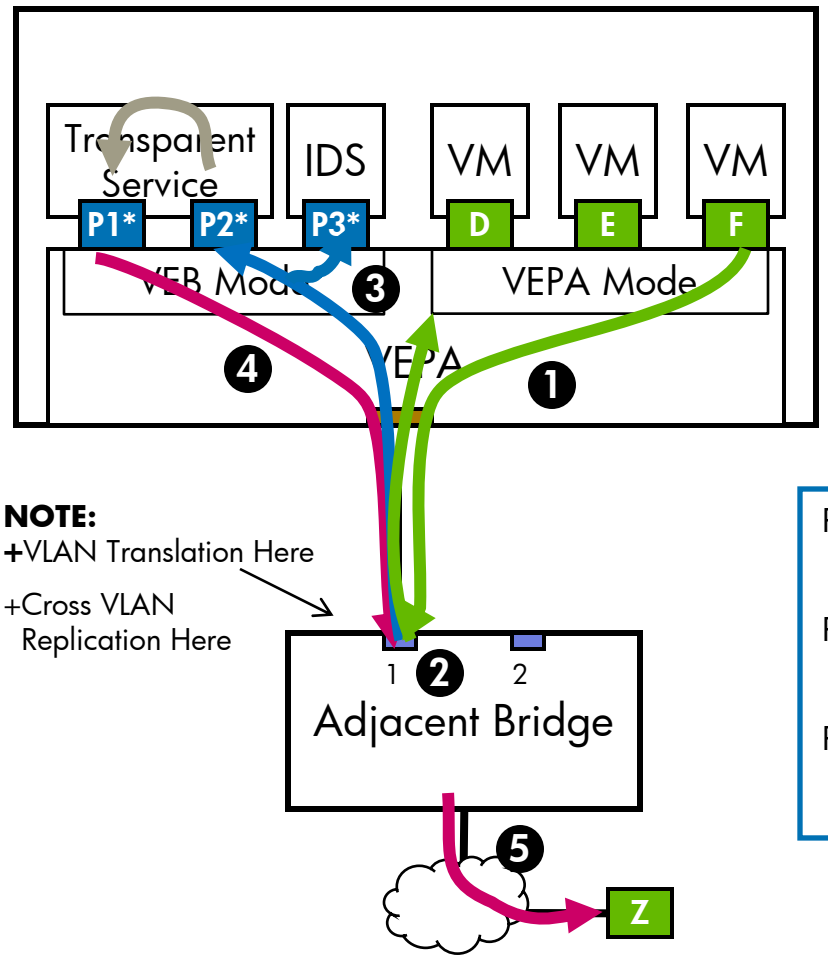
DST MAC	VLAN	Copy To (123DEF)
C	3	001000
D	3	000100
E	3	000010
F	3	000001
Bcast	1	100000
Bcast	2	011000
Bcast	3	000111
Unk Ucast	1	100000
Unk Ucast	2	011000
Unk Ucast	3	000000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	N	0UU000 T
3	Y	000UUU T

Combined VEPA/VEB Example

Multiple Transparent Services

SRC = F; DST = Z



Port P1*
PVID = 1
Egress list = 1
Port P2*, P3*
PVID = 2
Egress list = 2
Ports C,D,E,F
PVID = 3
Egress list = 3

VEPA Address Table

DST MAC	VLAN	Copy To (123DEF)
C	3	001000
D	3	000100
E	3	000010
F	3	000001
Bcast	1	100000
Bcast	2	011000
Bcast	3	000111
Unk Ucast	1	100000
Unk Ucast	2	011000
Unk Ucast	3	000000

VLAN	Reflect	Egress (**CDEF Up)
1	N	T00000 T
2	N	0UU000 T
3	Y	000UUU T

Advanced Topics: Other "Case Studies"

General Comments

- Many of the case studies show stacked VEPAs.
  - While VEPAs can be stacked, we don't expect them to be
  - VEPA is focused on the edge
- Some examples assume that the logic to support VEBs is not there.
  - VEBs are a necessity at the edge
  - VEPA is a simple, low-cost extension to VEBs
- Responses assume that you are already familiar with the case studies.

'Case Study'

Address Learning & Forwarding Transparent Services Example

Description:

- Transparent services may be placed in-line. Since they may pass packets with many MAC addresses and VEPAs don't do learning, it is stated that VEPAs can't support these types of services.

Note:

- Example shows stacked VEPAs (not the focus for EVB)

Response:

- VEPAs can be configured to isolate the traffic into and out of transparent services using private VLAN techniques.
  - Such an approach does not require new frame formats, new number/name spaces, and new address tables.
- Users may also opt for other approaches
  - Hypervisor transparent services APIs rather than a separately-attached VM
  - Use a VEB
  - VEPAs could also be extended to have limited learning capabilities, but this is not the recommended solution.

‘Case Study’
Ingress VLAN Enforcement

Description:

- 802.1Q provides ability for a switch port to restrict the egress VLANs to a specific set of the 4K available VLANs.

Note:

- VEBs and VEPAs have similar behavior

Response:

- There is limited need to support multiple VLANs at the vPort
  - End stations using several VLANs are usually VM hosts (therefore not a virtual end station)
  - Hypervisors can simply create an additional vNICs instead
- Address table can be used to validate VLAN IDs
- vPorts with VLAN choices of: (none, one, few, all) meet market requirements
- VEBs/VEPAs could provide full 4K if market requires it
  - Hardware to support 4K VLANs is the same regardless of whether it is in the VEB/VEPA or built into the switch to support a virtual switch port.
  - Many 802.1 devices do not support 4K VLANs

‘Case Study’
ACLs for FCoE

Description:

- FCoE needs ACLs created via FIP snooping in order to prevent impersonation.

Note:

- VEBs and VEPAs have similar behavior
- Example shows stacked VEPAs (not the focus for EVB)

Response:

- Hypervisors do not usually expose FC/FCoE devices to the VMs
  - Storage adapters terminate in the hypervisor
  - Exposed to VMs as generic SCSI devices
- MAC filtering is supported by VEB/VEPA
  - Stops any ‘evil initiators’ at the VEB/VEPA
- Hypervisors can intercept vNIC requests to add MACs with FC-OUIs

‘Case Study’
Bridge Stacking

Description:

- Attaching a bridge beneath a VEPA appears beyond the scope of the VEPA proposal.

Note:

- Example shows stacked VEPAs (not the focus for EVB)

Response:

- VEPA is focused at the Edge Virtual Bridging issue described up-front.
- VEPA can handle transparent services and promiscuous ports if and when required.

‘Case Study’
Multicast Egress ACLs

Description:

- Creating egress ACLs based on the vPort number (odd, even) is not possible, so one can do service load balancing.

Note:

- VEBs and VEPAs have similar behavior
- Example shows stacked VEPAs (not the focus for EVB)

Response:

- A similar load-balancing approach can be achieved in a edge VEPA/VEB environment as follows:
  - Isolate the services a separate VLAN (e.g., VID translation)
  - Place an egress ACL on the traffic going to each instance of the service (which is now isolated from other VEB/VEPA ports by the VLAN).
  - Have the egress ACL block traffic with odd or even source MACs.
  - Services will now distributed by the client MAC

‘Case Study’
Comparative Forwarding Logic

Description:

- Without a new tag, the forwarding table of a VEPA is essentially the same as that required by a bridge.

Note:

- Example shows stacked VEPAs (not the focus for EVB)

Response:

- Yes! VEPAs and VEBs have extremely similar address tables and forwarding logic. Since you already have a VEB, the VEPA mode is essentially free.
- If you combine a VEB and an Interface Virtualizer, then now get extra VN-Tag table, logic, and testing.
- Also, the forwarding complexity is grossly over stated. Edge devices can typically work well without learning.

Conclusion

Call For Action

- IEEE 802.1 standardization of
  - Switch port operation when in 'hairpin' mode
  - Configuration of 'hairpin' mode
    - LLDP/DCBX capabilities exchange & configuration
    - Managed object definition
    - None, Always, by VLAN
- Industry Standardization of EVB management
  - Coordinated configuration of vPort settings
  - Mechanism & standards forum is still TBD
- Join the Edge Virtual Bridging Ad Hoc
  - http://tech.groups.yahoo.com/group/evb/
  - Conference Calls Tuesdays 1PM Central US