A Case for VEPA:
Virtual Ethernet Port Aggregator

September 2010

Paul Congdon - ptcongdon@ucdavis.edu
Anna Fischer - anna.fischer@hp.com
Prasant Mohapatra – prasant@ucdavis.edu

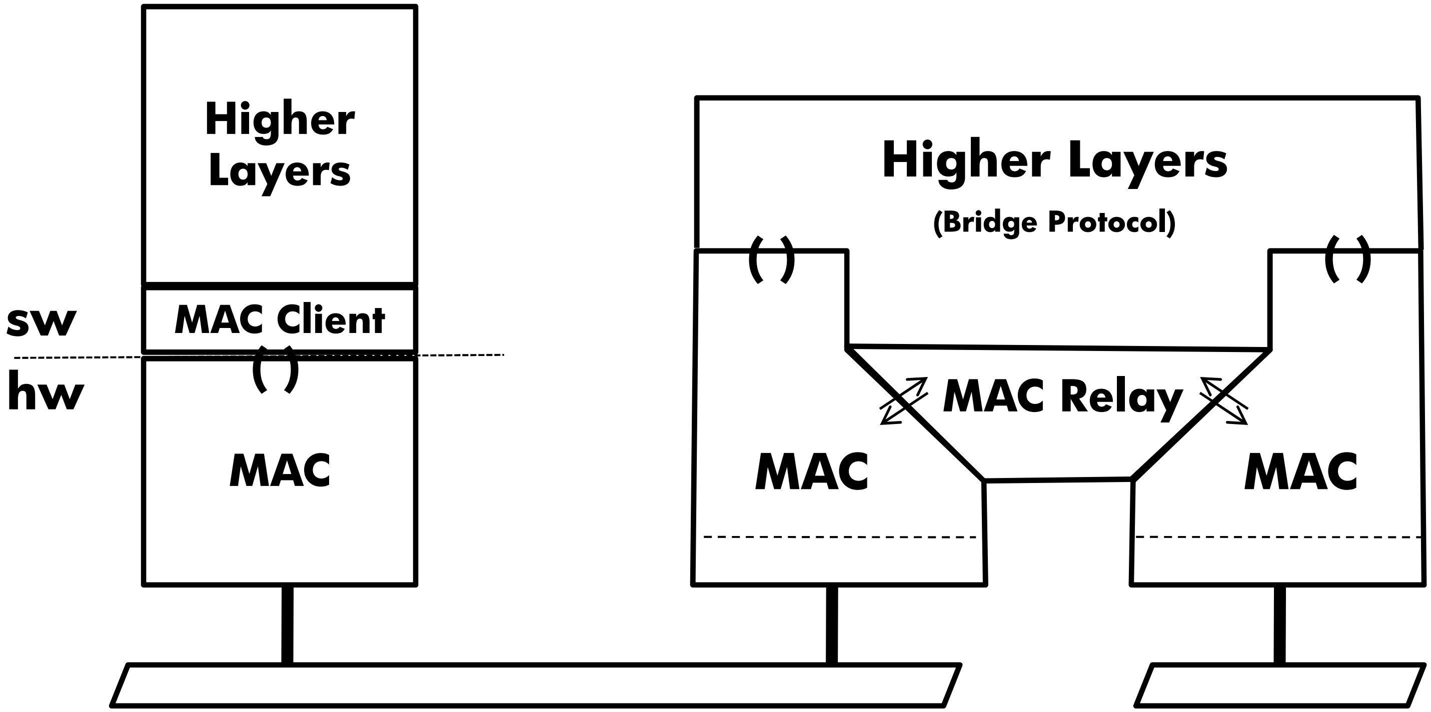
UC DAVIS

Agenda

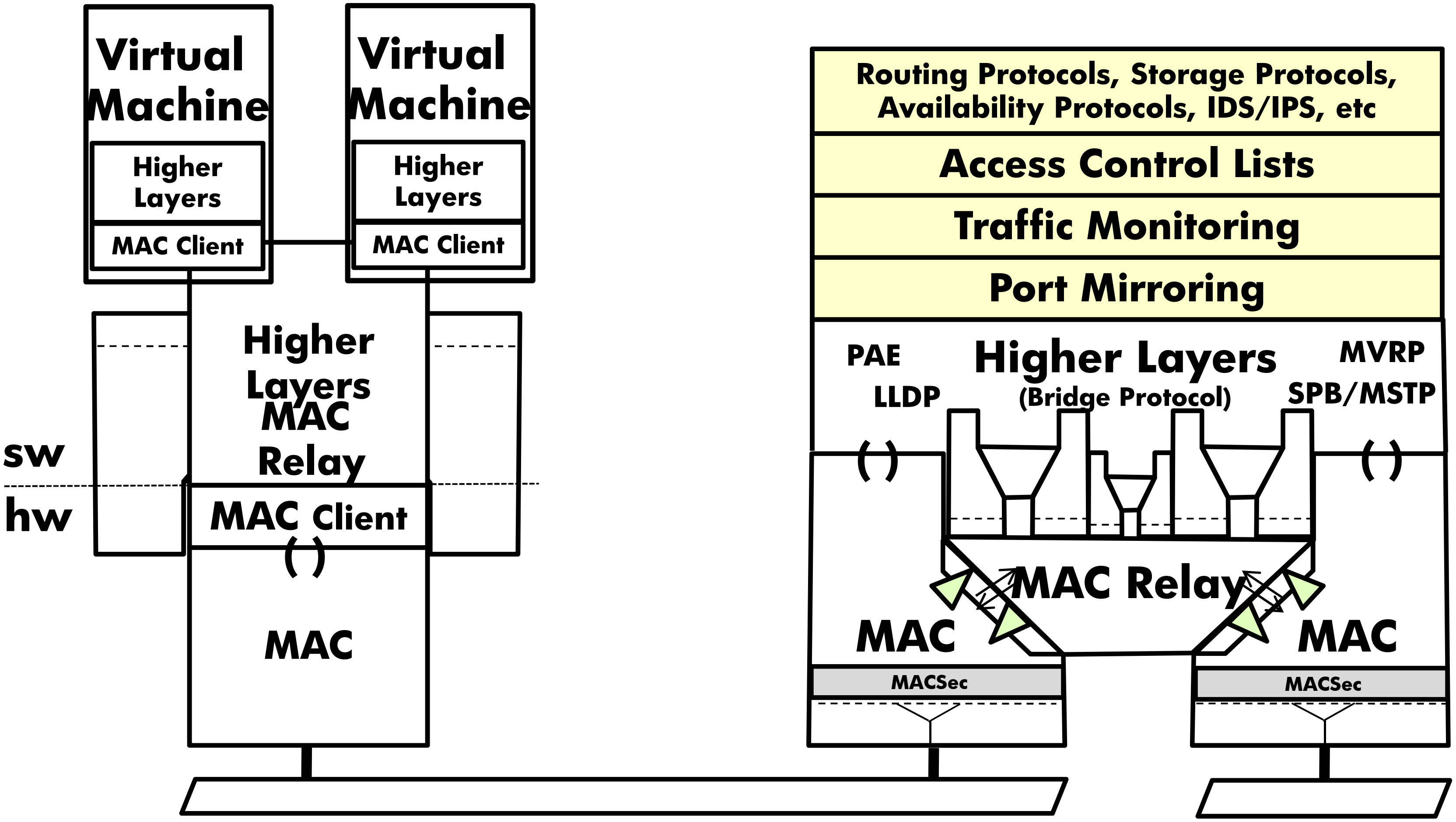
- Motivation
- Approach
- Solutions
  - VEB
  - VEPA
  - Multi-Channel
- Prototype and Evaluation Results
- Related Work
- Status and Likely Future

Traditional Networking

The end-station and bridge



Modern Networking
The end-station and bridge



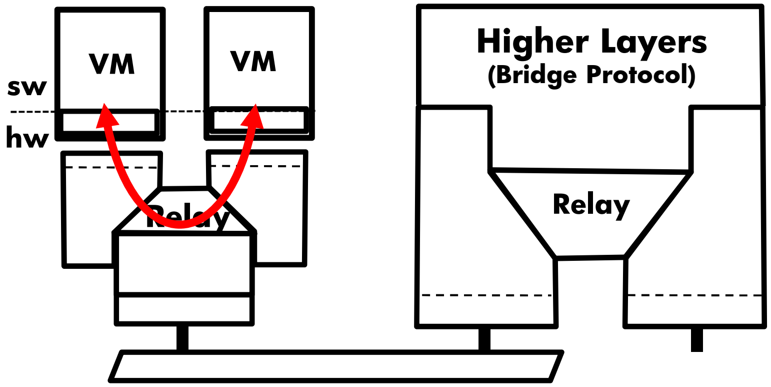
Issues with Virtualization and Networking

- –Performance
  - – I/O virtualization has high overhead
  - – Software forwarding doesn’t scale as inline features increase
- –Consistent Policy Enforcement
  - – physical network equipment is often used to enforce policy (firewalls, bandwidth control, QoS)
  - – policy controls and mechanisms within Server virtual networks are limited
- –Visibility
  - – Traditional network management tools can’t ‘see’ internal VM-to-VM traffic
- –Creating Secure Topologies
  - – Pools of VMs on different physical machines need to be interconnected on their own isolated network with full access to features

Exploiting Switch Adjacency

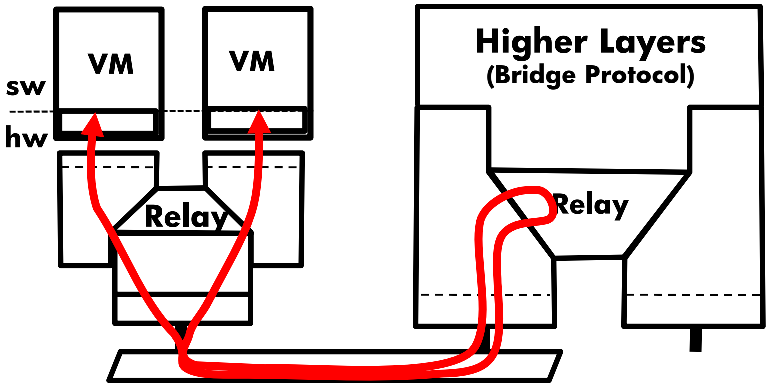
Virtual Ethernet Bridge (VEB)

- Well understood today
- Results in difficult trade-offs between cost and capability
- Debugging and administrative challenges



Virtual Ethernet Port Aggregator (VEPA)

- New method of forwarding
  - Simplifies management
  - Visibility
  - Traffic Control
  - Consistency
- Lower Cost Implementation

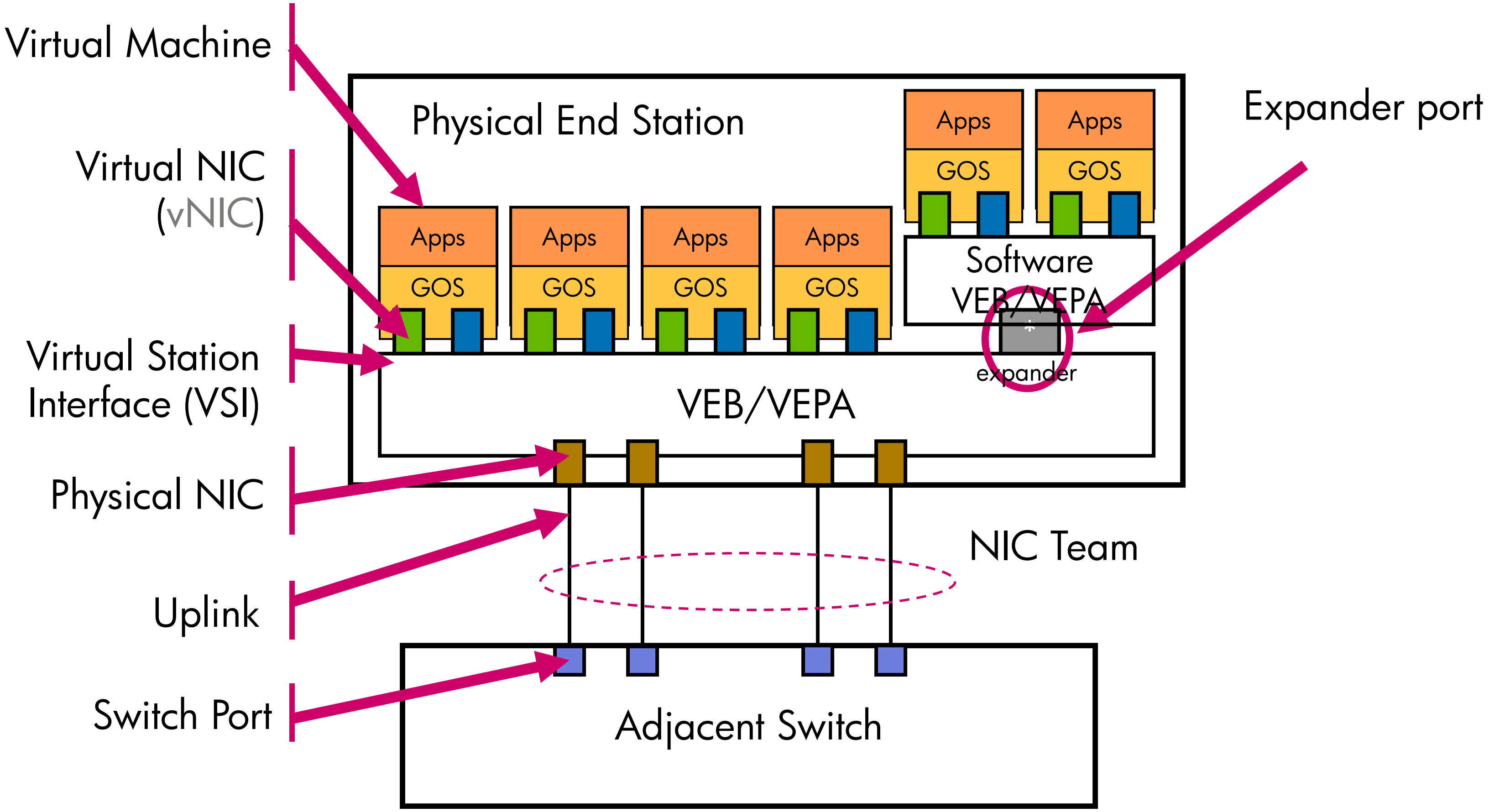


The case for VEPA

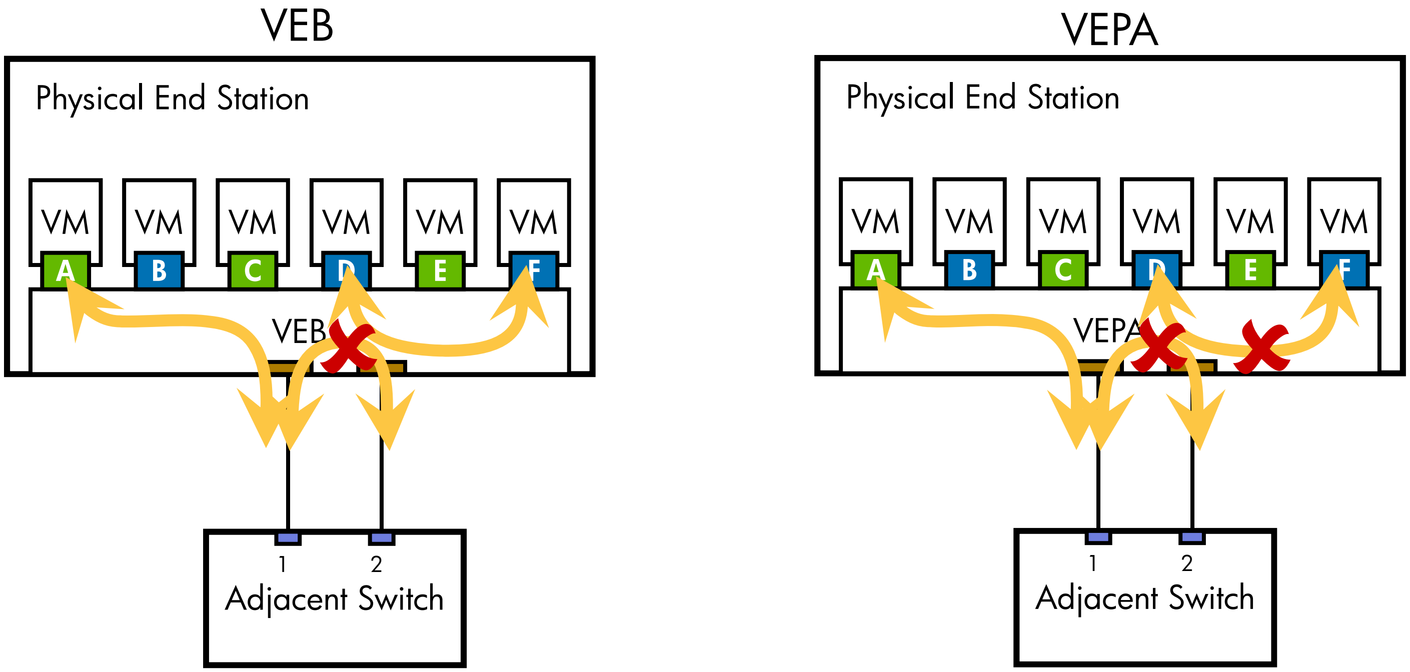
- Improve virtualized network I/O performance and reduce complex features in software based hypervisor switches
- Allow NICs to maintain low cost circuitry
- Enable consistent network policy enforcement
- Provide visibility to all VM traffic
- Reduce network configuration requirements on server administrator

Virtual Ethernet Bridges (VEBs)
Virtual Ethernet Port Aggregators (VEPAs)

Basic VEB/VEPA Anatomy and Terms



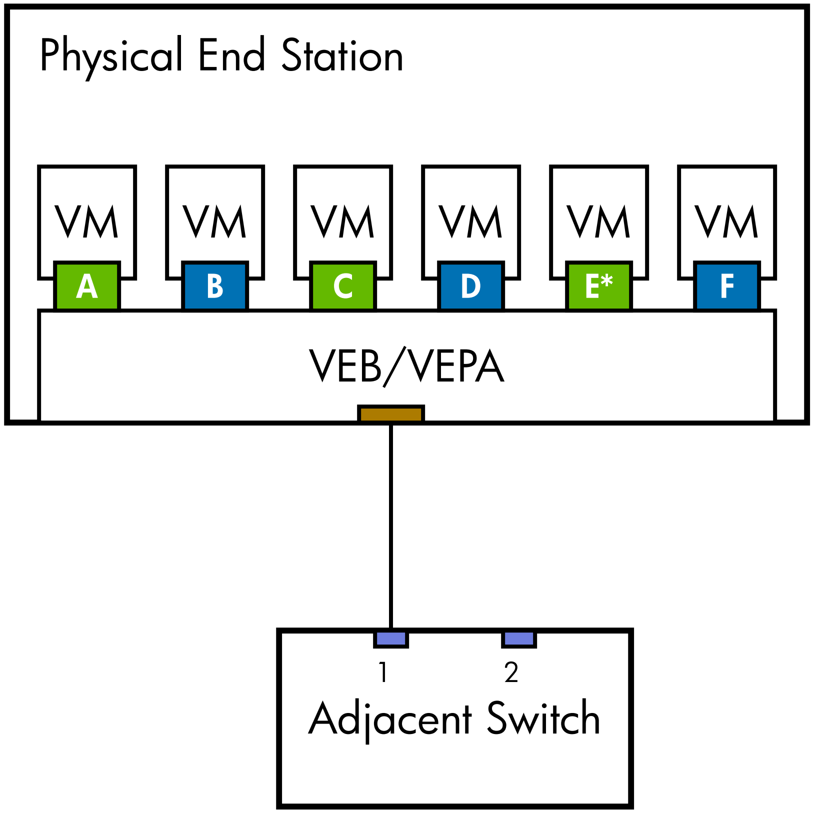
Loop-free Forwarding Behavior



- Forward based on MAC address (and port group or VLAN)
- Do NOT forward from uplink to uplink
  - Single active logical uplink
  - Multiple uplinks may be ‘teamed’ (802.3ad and other algorithms)
- No need to participate in (or affect) spanning tree

VEB/VEPA Address Table

Populated via MAC registration



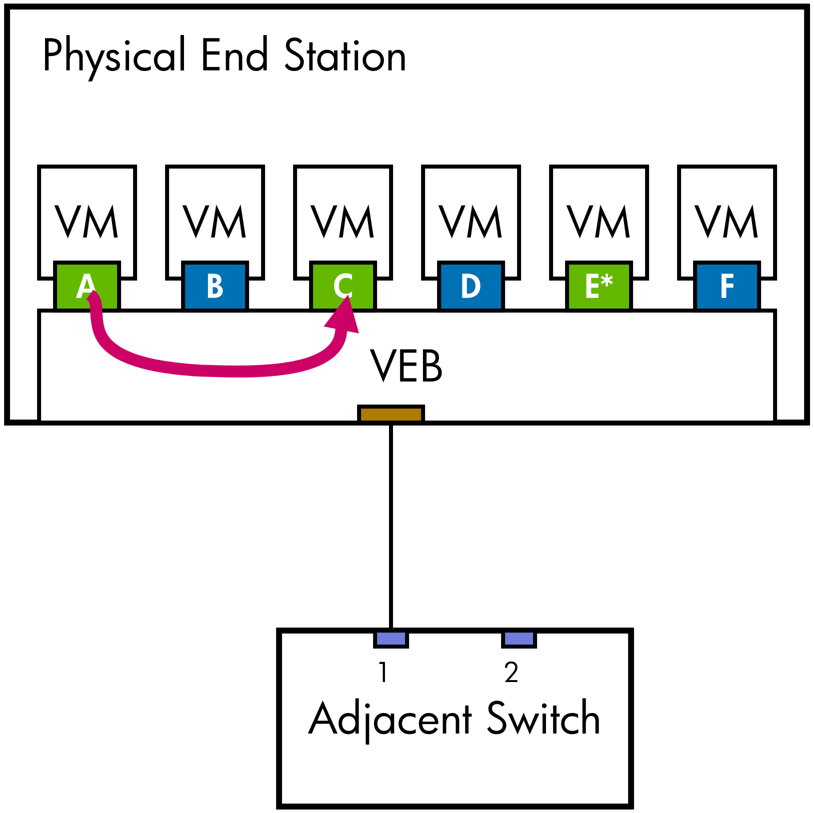
* Promiscuous VSI

VEB/VEPA 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

VEB Unicast Example

SRC = A; DST = C



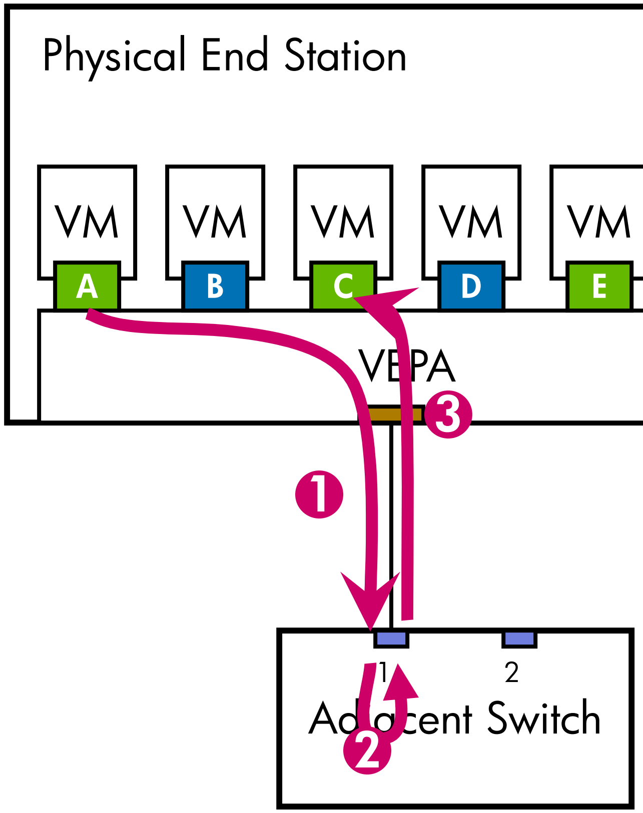
* Promiscuous VSI

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

VEPA Unicast Example

SRC = A; DST = C



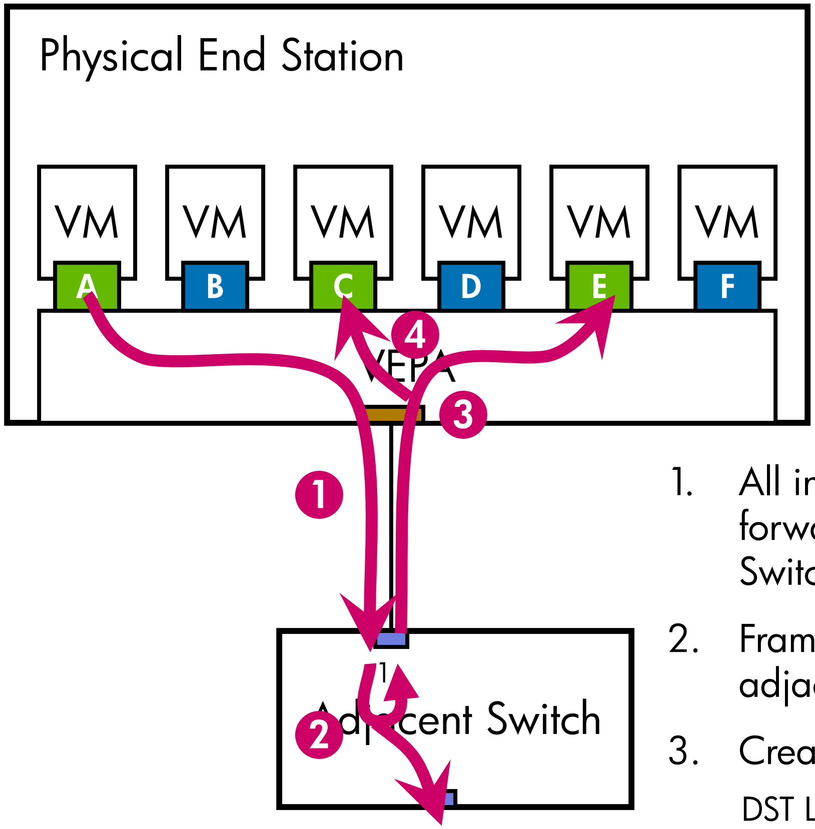
1. All ingress frames forwarded to adjacent Switch
2. Frame forwarded based on adj. Switch learning.
3. Frame forwarded based on delivery mask generated from VEPA address table

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

VEPA Multicast Example

SRC = A; DST = MulticastC



1. All ingress frames forwarded to adjacent Switch
2. Frame forwarded by adjacent Switch.
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

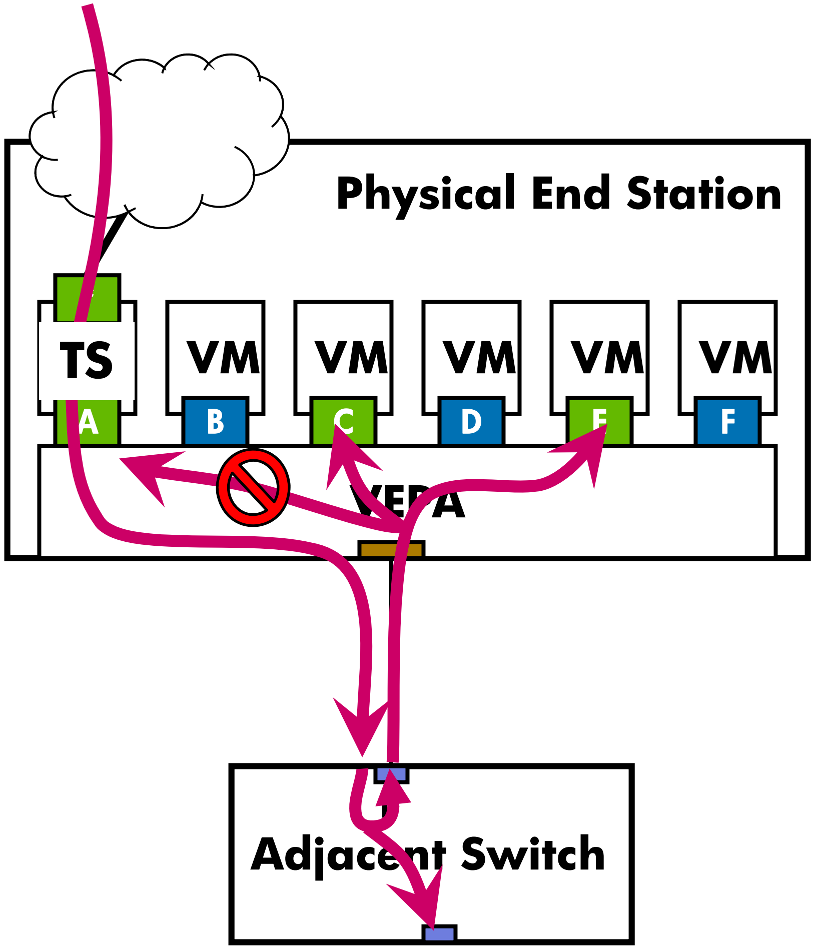
‘Basic VEPA’ Limitations

- Basic VEPA is challenged by promiscuous ports
  - Must have complete address table and learning is problematic
  - Difficult to create proper destination mask to account for promiscuous ports
  - Useful to support inline transparent services

- Mixing VEPA and VEB ports on single physical link
  - Allow for optimized performance configuration

Problem with Dynamic Addresses

SRC = Z; DST = MulticastC



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

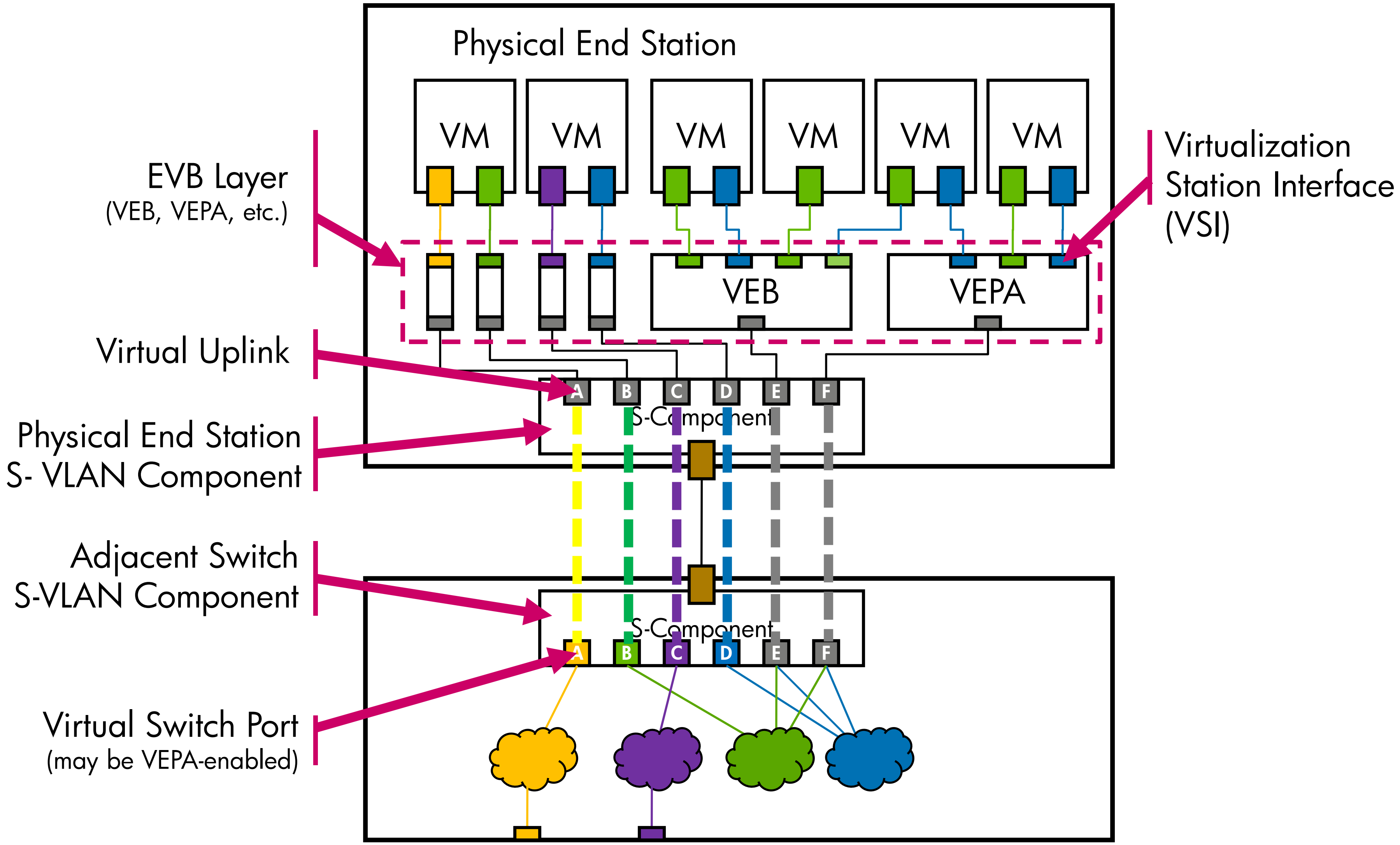
Tagging Schemes to the Rescue

- Filtering problem is eliminated by ‘isolating’ the VSIs
- Tagging schemes provide a virtual port indication for the adjacent Switch
- Normal Switch learning and flooding are extended can be extended to VSIs

- New problems arise…

MultiChannel

New Anatomy and Terms



EVB Layer
(VEB, VEPA, etc.)

Virtual Uplink

Physical End Station
S- VLAN Component

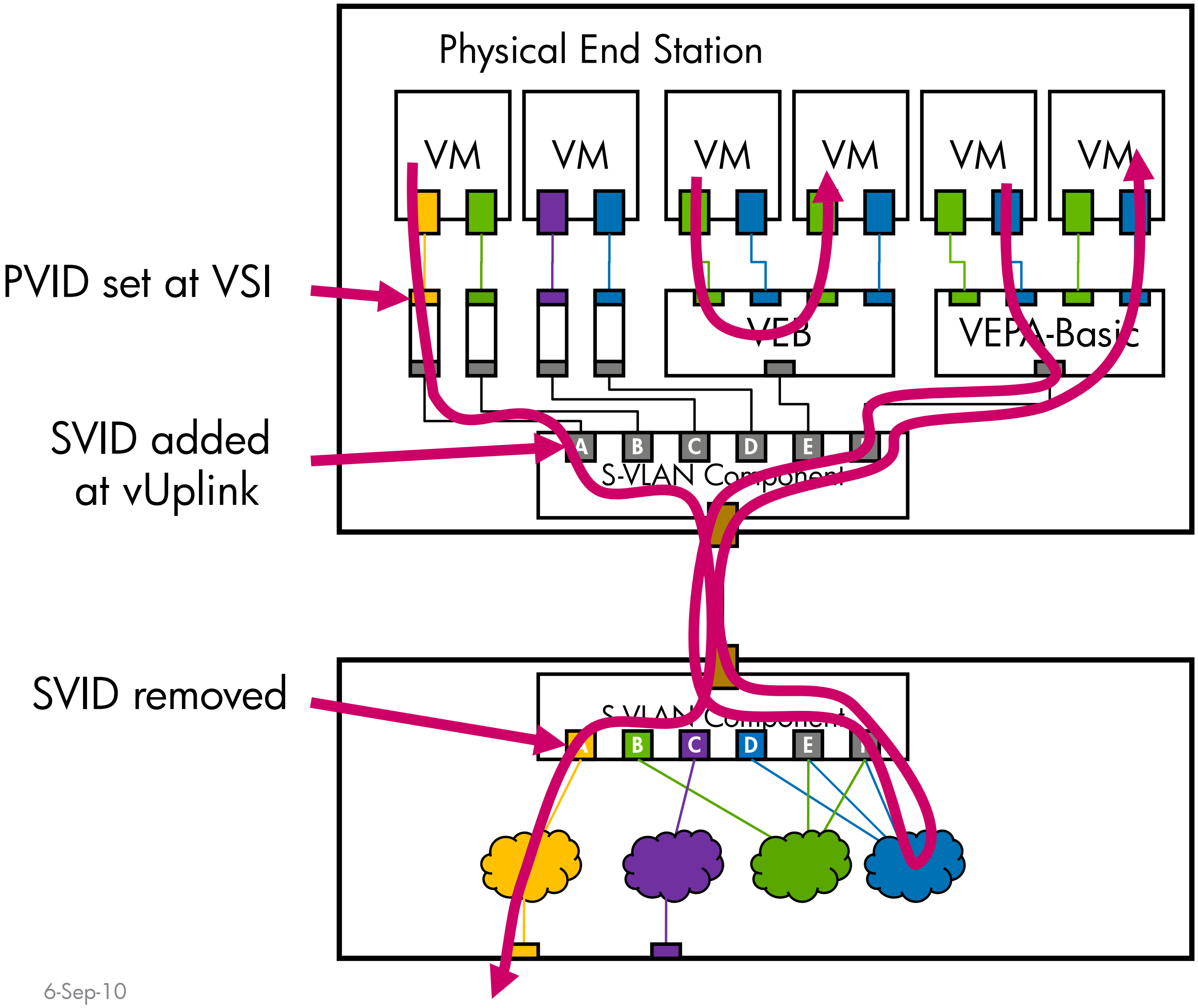
Adjacent Switch
S-VLAN Component

Virtual Switch Port
(may be VEPA-enabled)

Virtualization
Station Interface
(VSI)

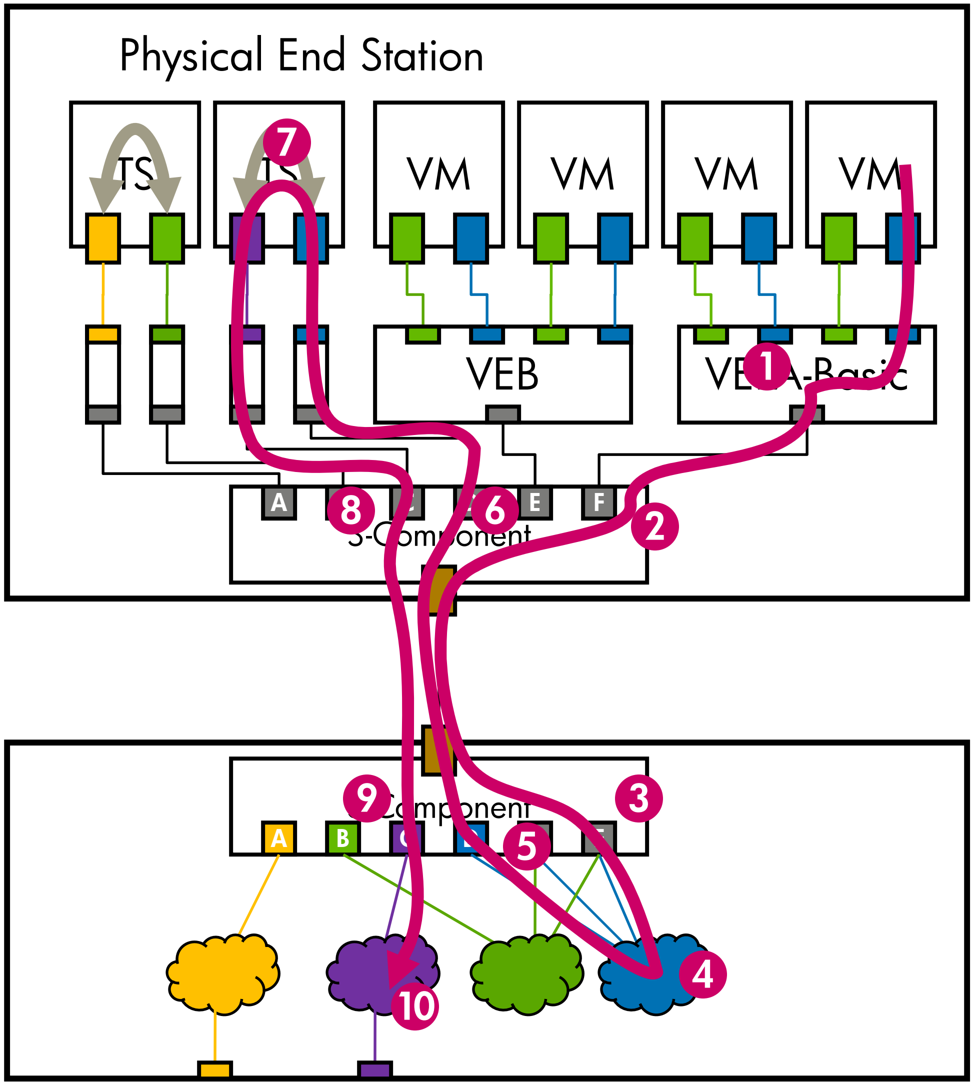
MultiChannel Approach

Isolation VSI, VEB and VEPA simultaneously



MultiChannel Approach

Example: Using Transparent Service Separating Blue & Purple VLANs



1. VEPA ingress frame from VM forwarded out VEPA uplink to S-Component
2. Station S-Component adds SVID (F)
3. Switch S-Component removes SVID (F)
4. Forwards based on Switch forwarding table to virtual switch port E.
5. Switch S-Component adds SVID (D)
6. Station S-Component removes SVID (D)
7. Transparent service switches across to purple VLAN.
8. Station S-Component adds SVID (C)
9. Switch S-Component removes SVID (C)
10. Switch forwards frame on purple VLAN.

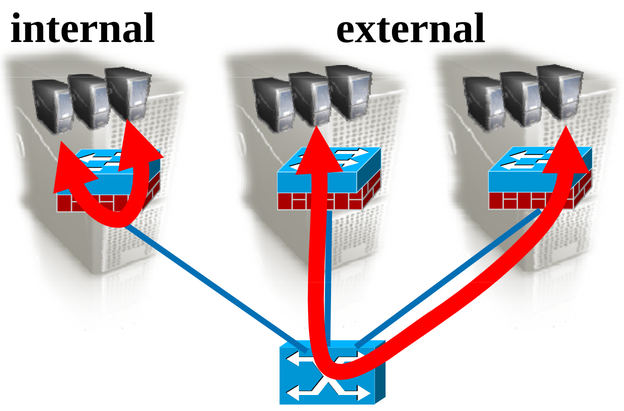
Implementation and Results

UCDAVIS

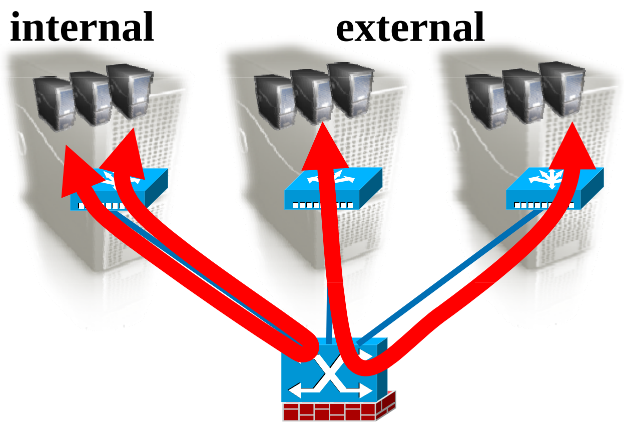
VEPA Open Source Implementation

- Patches available for VEPA and hairpin mode:
  - net/bridge: base 2.6.30 kernel, Xen’s 2.6.18.8 Dom0
  - bridge-utils: brctl commands to enable/disable modes
  - tools: Xen tools equivalent
- Very minor changes required
  - 37 lines of code in VEPA data path
  - 2 lines of code for hairpin mode
- Tested in KVM and Xen
- Tested against 3rd party switch with hairpin mode

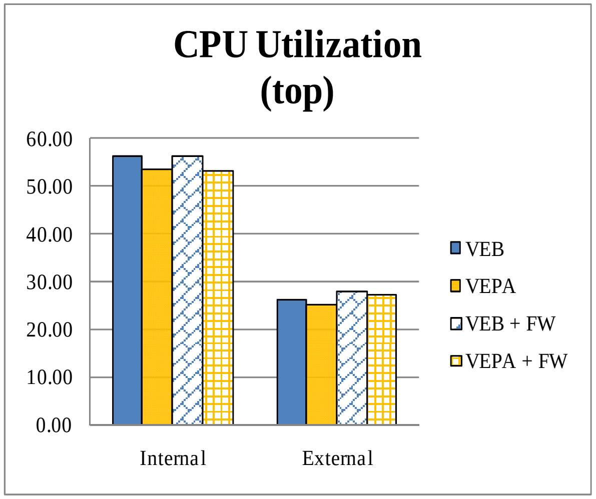
VEB/VEPA Comparison

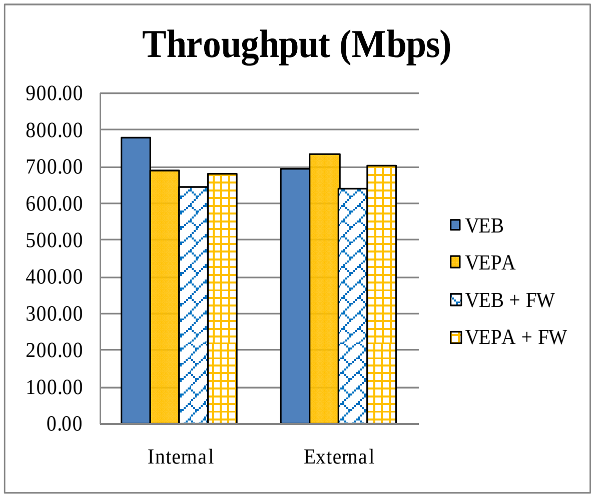


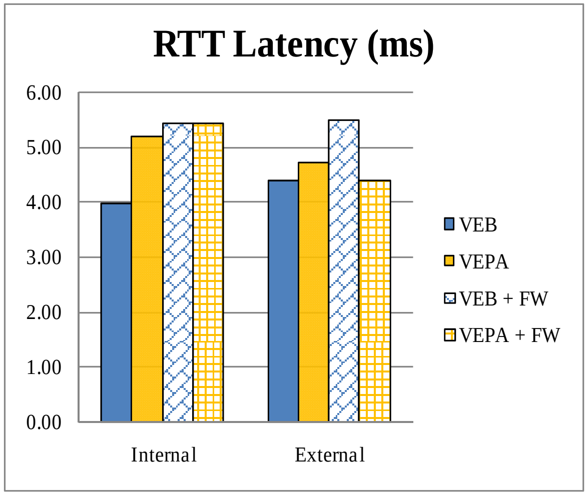
VEB configurations



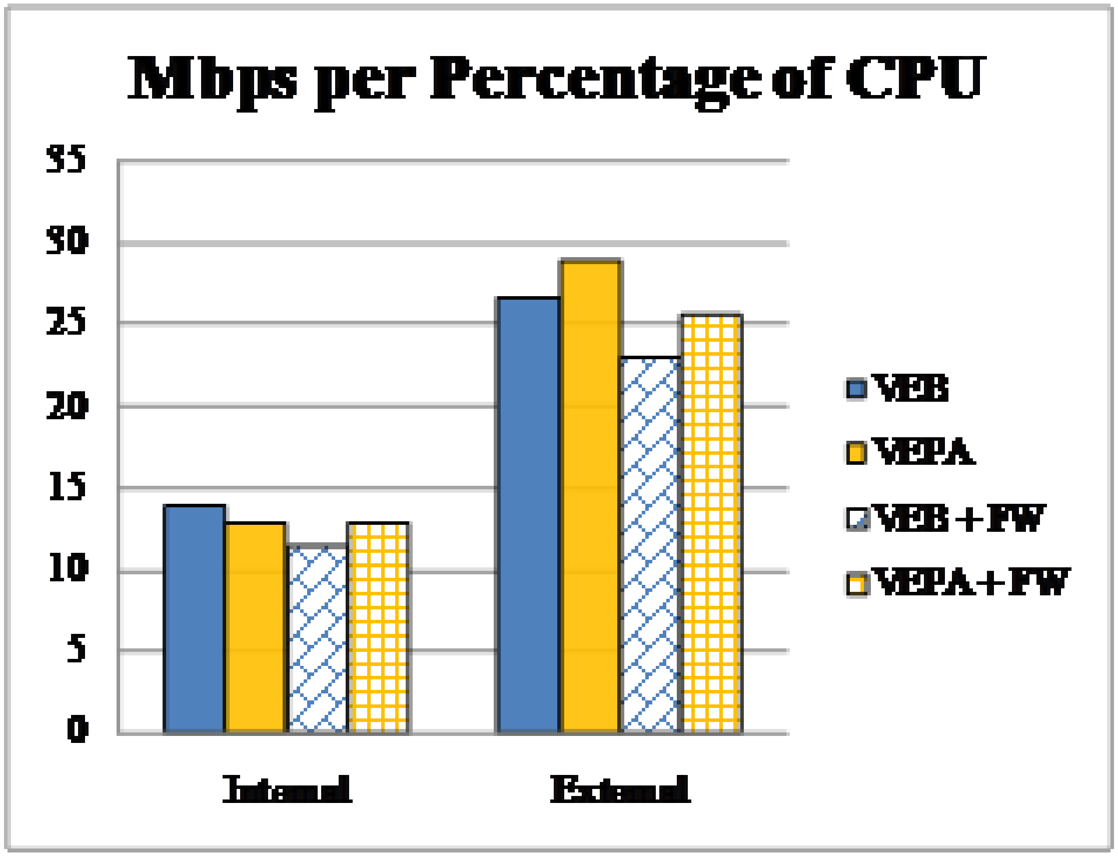
VEPA configurations


CPU Utilization (top)

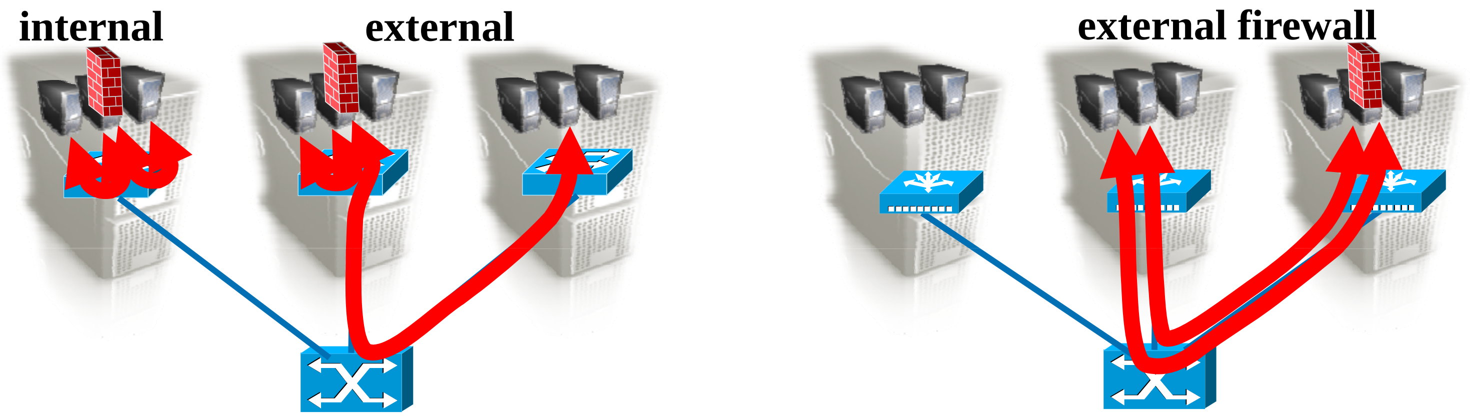

Throughput (Mbps)

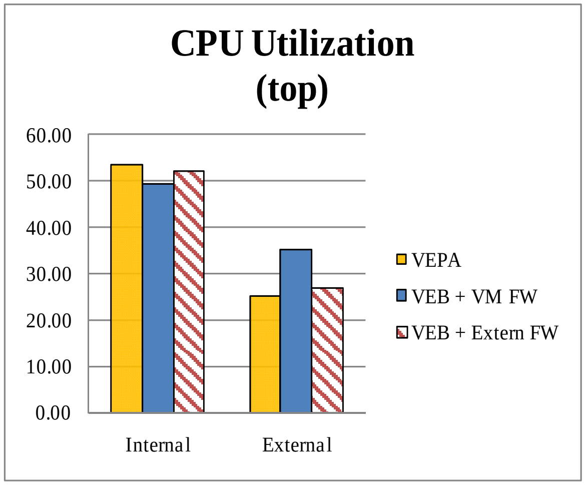

RTT Latency (ms)

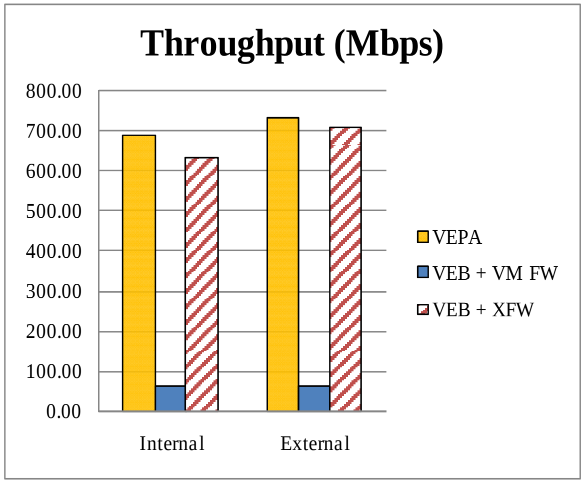
A Measure of Efficiency

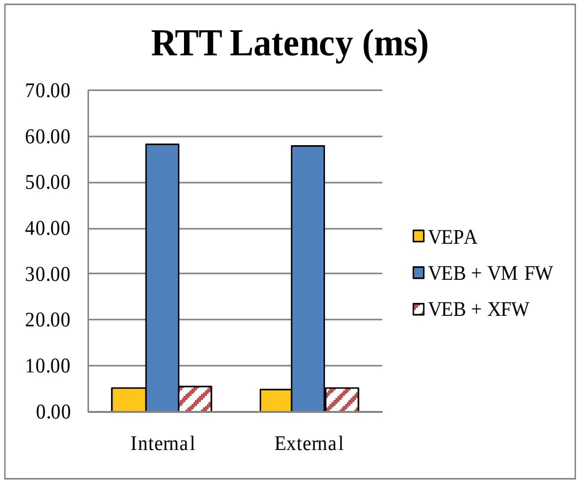


VM Appliance Comparison Topologies

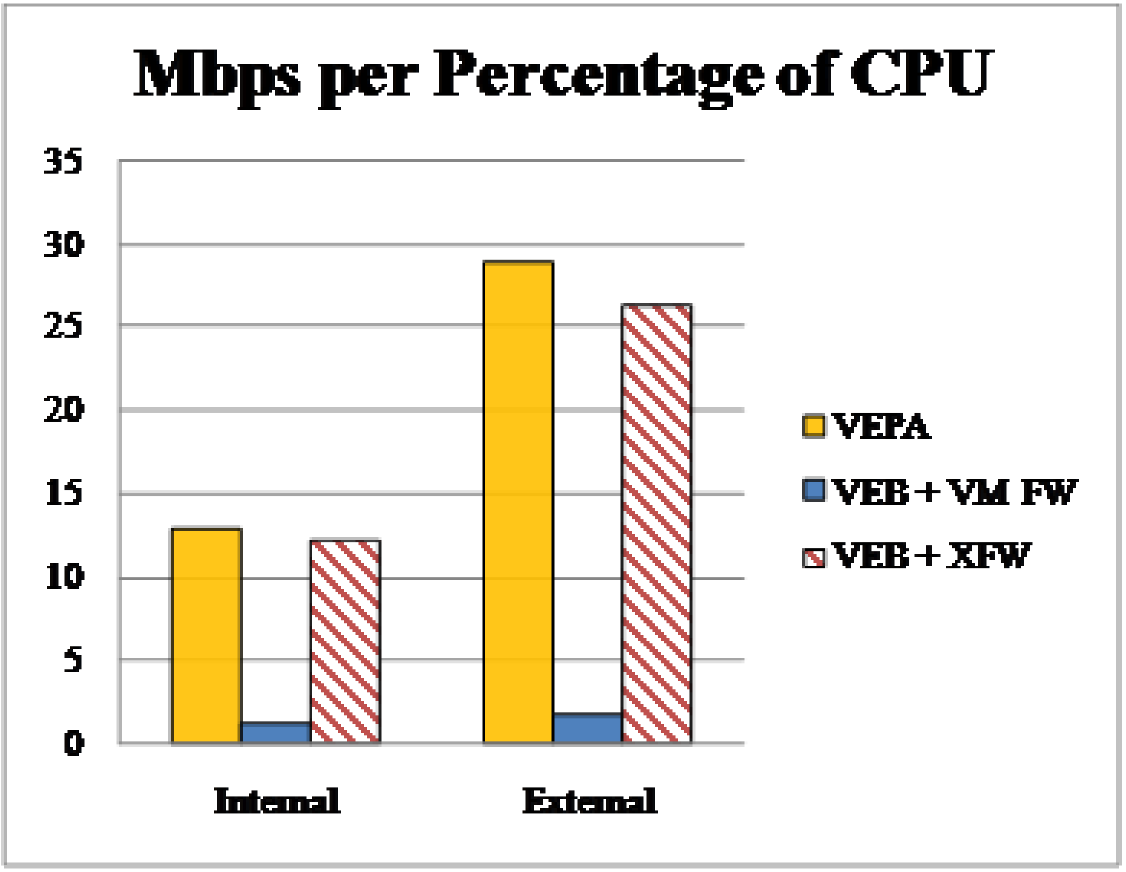



CPU Utilization (top)


Throughput (Mbps)


RTT Latency (ms)

A Measure of Efficiency



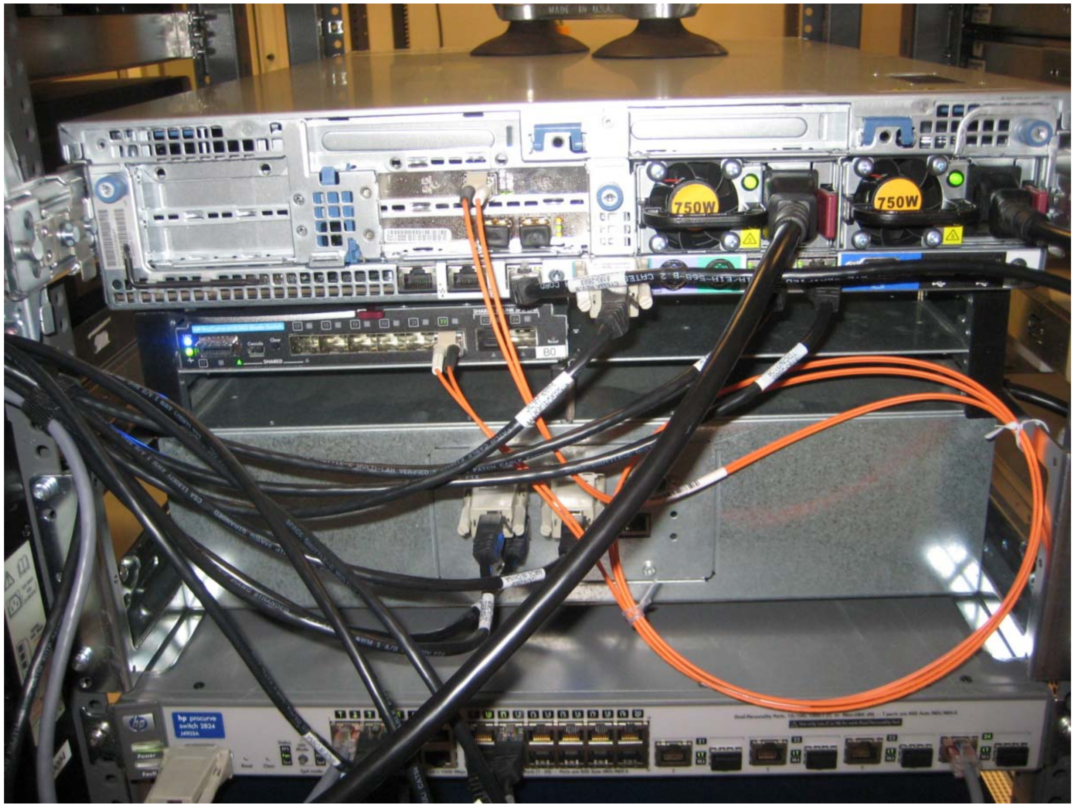
Status and future directions

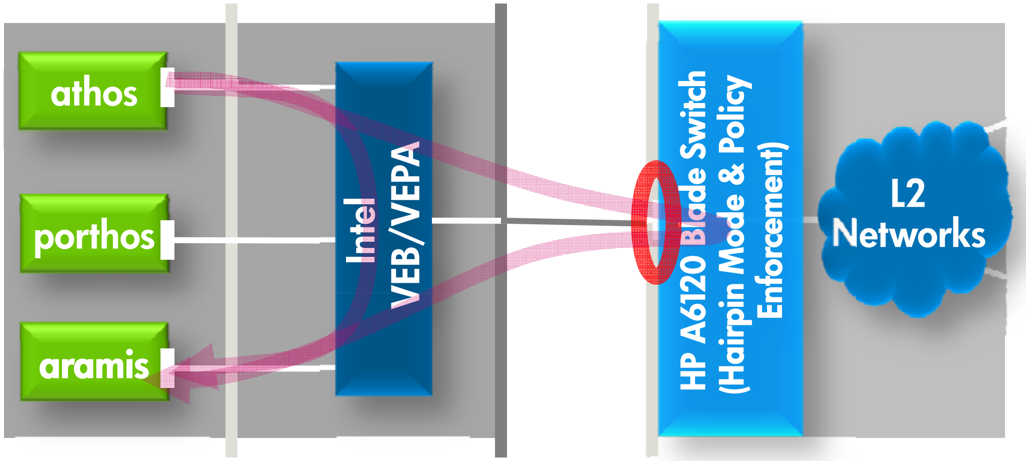
IEEE 802.1 Standards Activities

- IEEE 802.1Qbg – Edge Virtual Bridging
  - Reflective Relay (Hairpin)
  - Multi-channel
  - VEPA filtering requirements
  - Protocols
    - EDCP - Edge Discovery and Configuration Protocol (LLDP extension)
    - CDCP - S-channel Discovery and Configuration Protocol (LLDP extension)
    - ECP - Edge Control Protocol
    - VDP - VSI Discovery and Configuration Protocol (ECP extension)
- IEEE 802.1Qbh – Port Extension
  - Remote Replication services
  - Replication tag and forwarding database
  - Protocols
    - PECSP – Port Extension Control and Status Protocol (ECP extension)

SR-IOV VEPA Demo

- 3 VMs on a single server running Xen
- Intel 82599 SR-IOV NIC with VEPA capability
- HPN A6120 switch with hairpin mode enabled
- ACLs and sFlow available on A6120 edge switch





Thank You

UC DAVIS