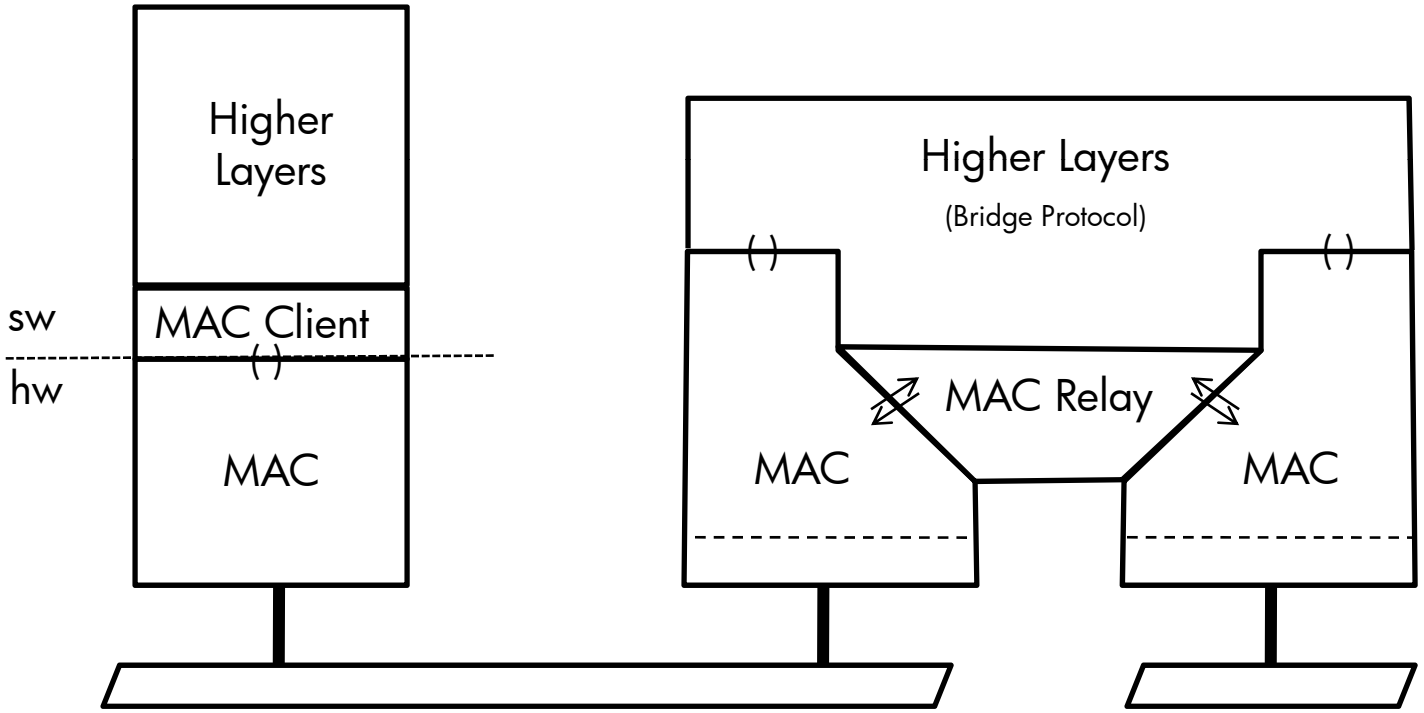
# Virtual Ethernet Port Aggregator Standards Body Discussion

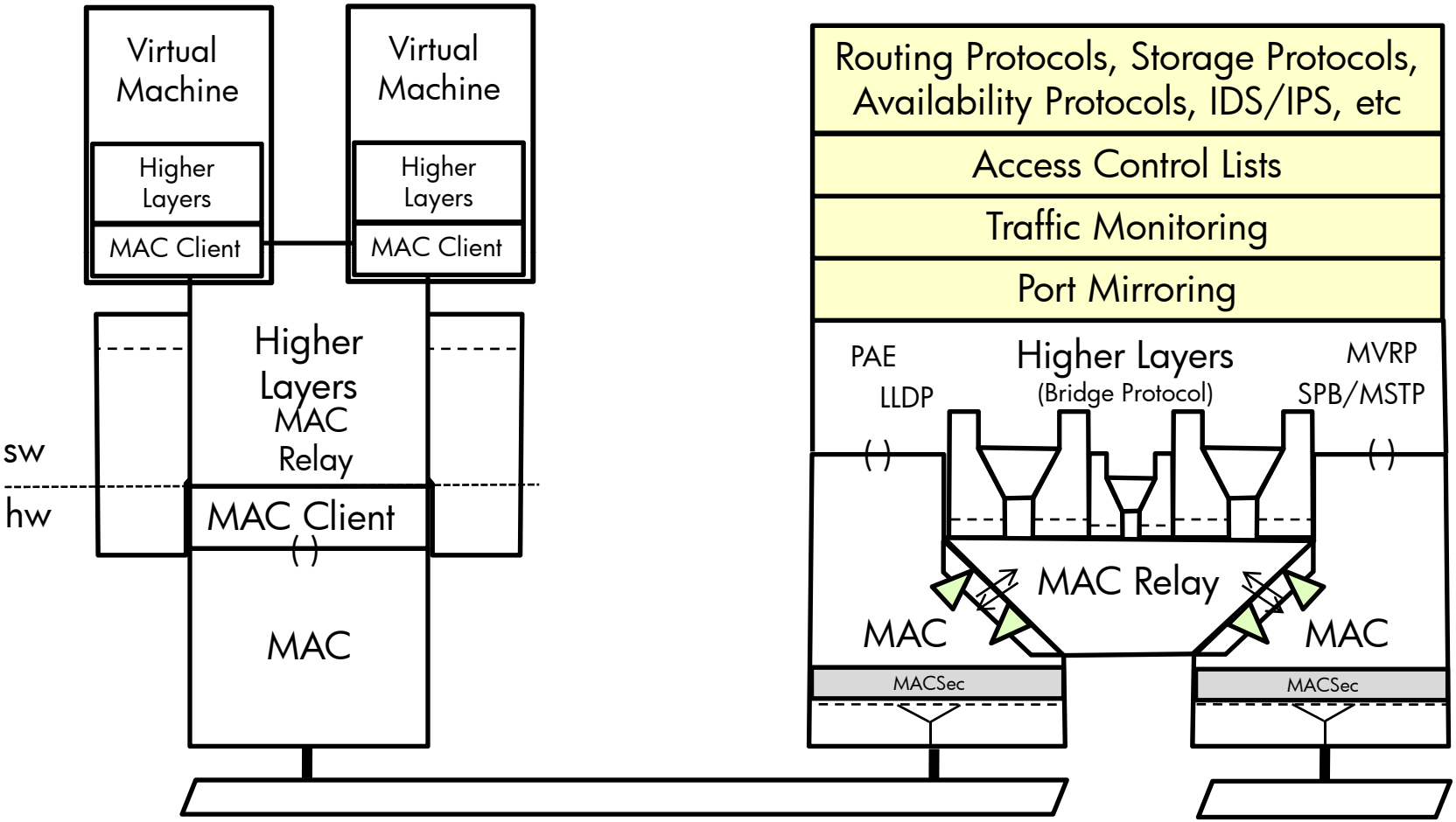
**November 10, 2008**

**Paul Congdon**

# Traditional End-Station and Bridge



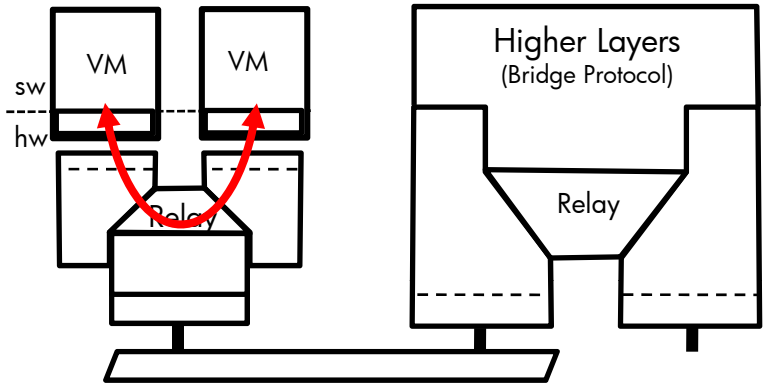
# Current End-Station and Bridge



# High-Level Traffic Flow diagram

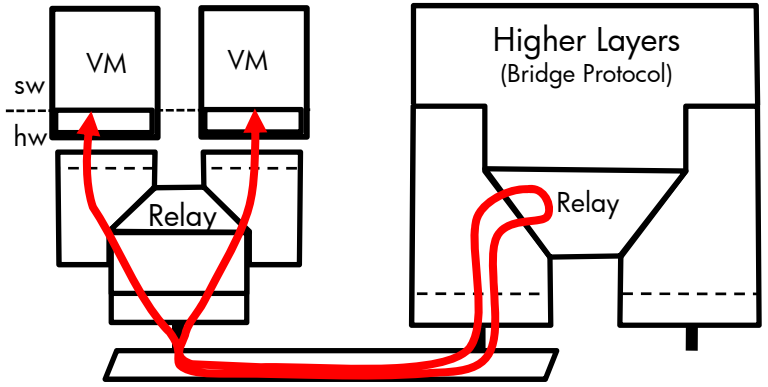
- If you want this…

  Fine.. It's called a "bridge" and we have standards for that




- If you want this…

  New forwarding modes need to be defined.




- The component on the host is no longer a virtual switch, but rather a Virtual Ethernet Port Aggregator (VEPA)

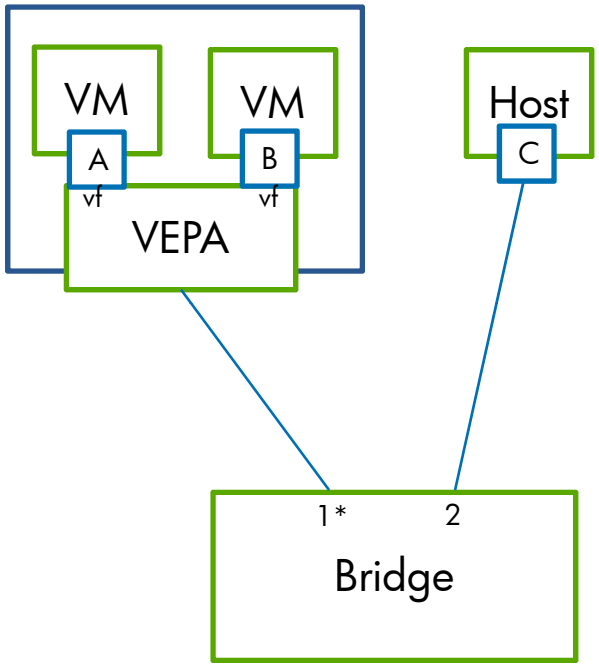
# Goals for a Virtual Ethernet Port Aggregator

- Provide external network visibility and management of all per VM traffic
- Partition the work between NICs and Bridges to leverage their respective strengths
- SR-IOV Virtual Functions assigned to VMs for performance (and don't break anything else)
- Correctly and efficiently solve the unicast, multicast and broadcast problems
- Allow the Hypervisor Virtual Switch to become optional
- Align with established IEEE practices

# Possible Technical Approaches

- Untagged
  - No modifications to existing packets
  - No modifications to existing Bridge tables or learning behavior
  - Policy enforcement and network visibility done on a per-MAC basis
  - Leverage the potential existence of a MAC address table on the NIC to "steer" and filter traffic to VMs
  - Multicast/Broadcast replication is done on the NIC
- Tagged
  - Tag packets to explicitly indicate the Virtual Machine port
  - Bridge forwards between virtual ports within the Bridge
  - Policy enforcement and network visibility done on a per-Port basis
  - Tag to Virtual Function mapping table "steers" traffic. No MAC address table needed on the VEPA. The tag is essentially a new address space.
  - Multicast/Broadcast replication *may* be done on the Bridge or on the VEPA. The later requires additional tags to represent multicast groups.

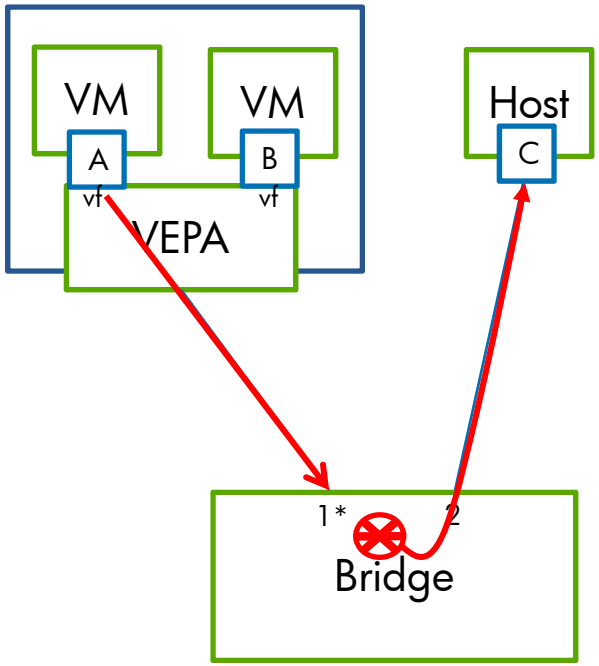
# Virtual Ethernet Port Aggregator




1. Provides Multiple Virtual Functions (VFs) as vNICS to Virtual Machines
2. Each VF is configured as individual NIC (i.e. MAC addr, Multicast addrs, Promiscuous, VLAN tags or passthru). VEPA aggregates configurations.
3. May support all traditional NIC features (e.g. TCP Checksum, RSS, Large Segment Send)
4. Does NOT perform Local Bridgeing. Not a Virtual Ethernet Bridge (VEB)
5. Sends all outbound traffic to the wire
6. Replicates mcast/bcast received traffic
7. VLAN aware
8. May provide QoS and BW management
9. Invoked by special Bridge mode negotiation

Note: This proposal does NOT require new tags, but could work with them.

# VEPA Forwarding



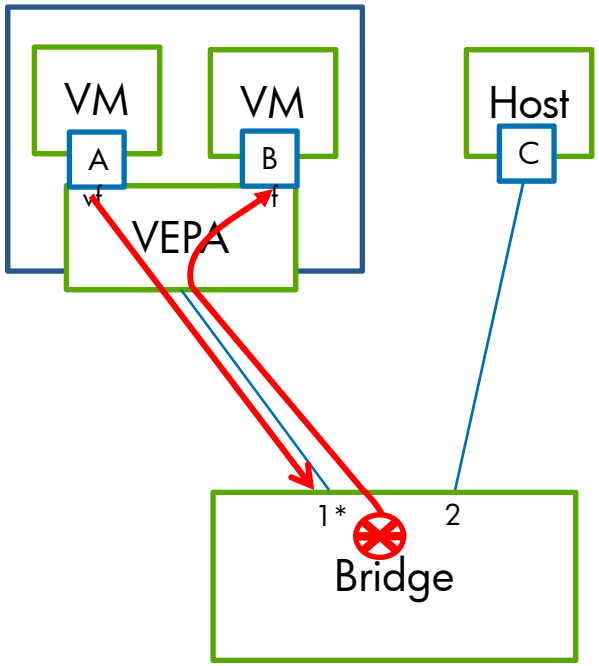

1. A->C
2. A->B
3. A->Bcast
4. C->Bcast

Bridge Address Table

| Address | Port |
|---|---|
| A | 1 |
| B | 1 |
| C | 2 |

* = Bridge Port Configured for VEPA attach

# VEPA Forwarding



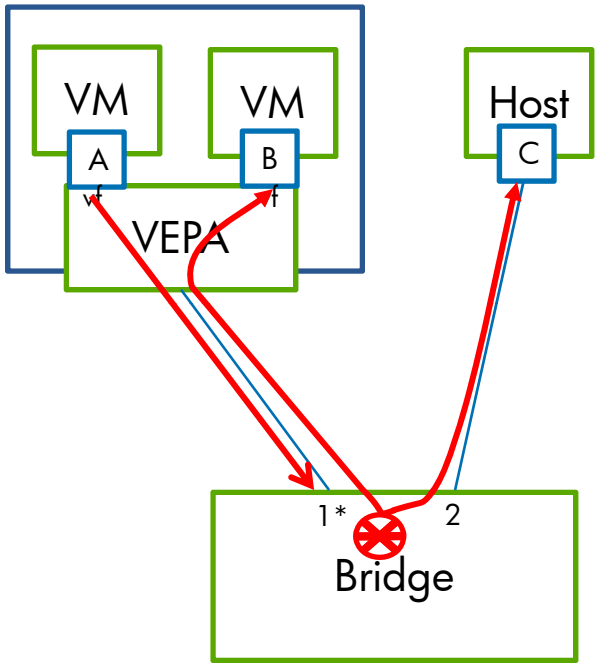

1. A->C
2. A->B
3. A->Bcast
4. C->Bcast

Bridge Address Table

| Address | Port |
|---|---|
| A | 1 |
| B | 1 |
| C | 2 |

* = Bridge Port Configured for VEPA attach

# VEPA Forwarding



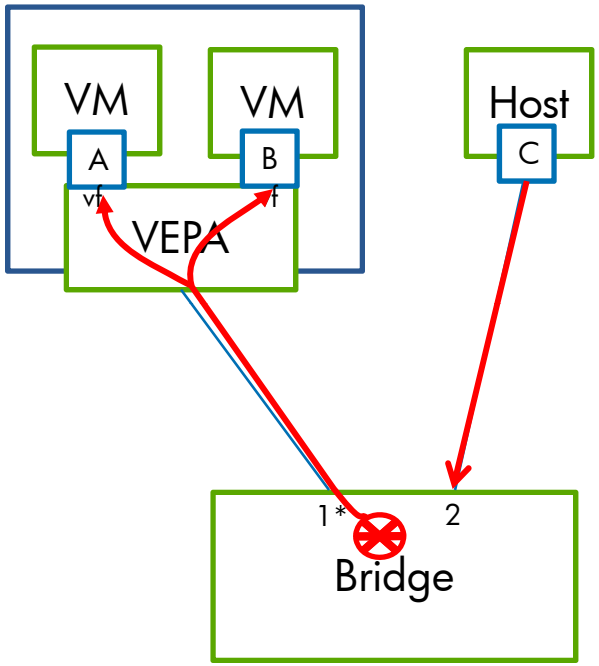

1. A->C
2. A->B
3. A->Bcast
4. C->Bcast

Bridge Address Table

| Address | Port |
|---|---|
| A | 1 |
| B | 1 |
| C | 2 |

* = Bridge Port Configured for VEPA attach

# VEPA Forwarding



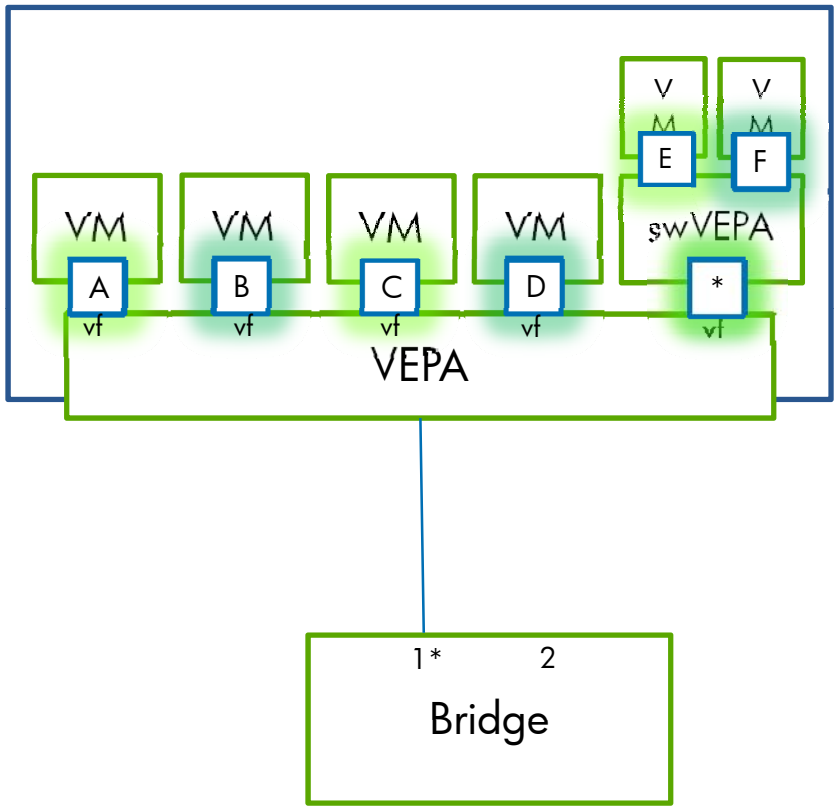

1. A->C
2. A->B
3. A->Bcast
4. C->Bcast

Bridge Address Table

| Address | Port |
|---|---|
| A | 1 |
| B | 1 |
| C | 2 |

* = Bridge Port Configured for VEPA attach

# VEPA Multicast and VLANs




Example: VEPA Address Table

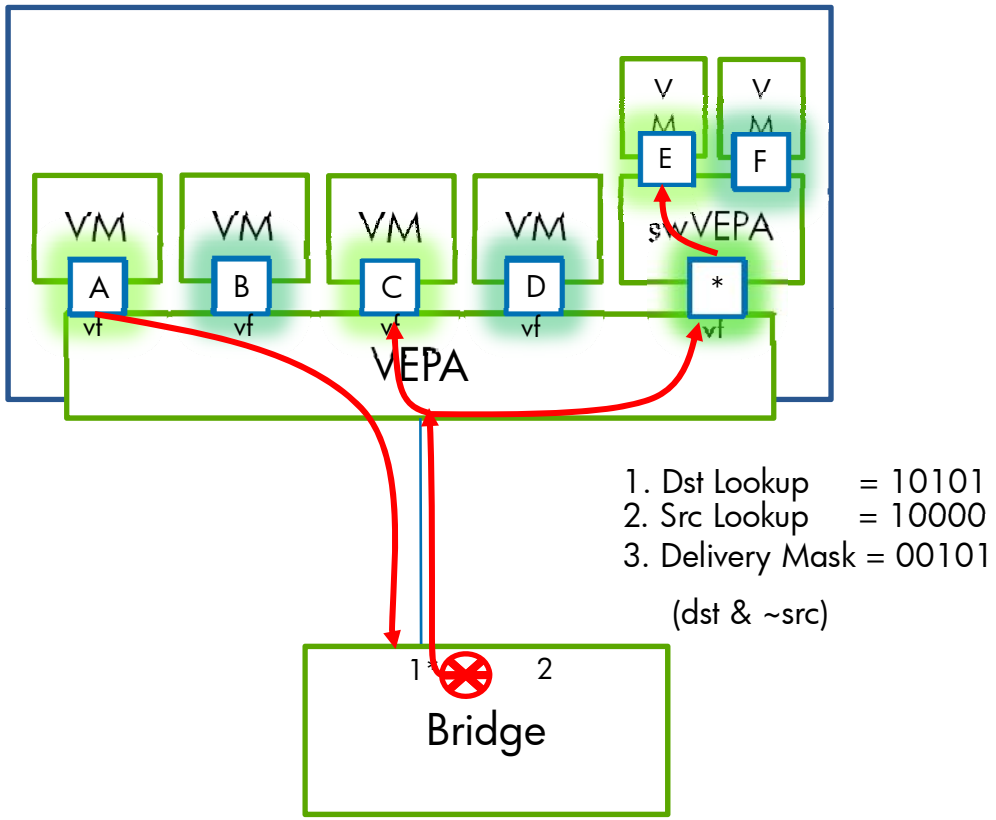
| Address | VLAN | VF Mask |
|---|---|---|
| A | 1 | 10000 |
| B | 2 | 01000 |
| C | 1 | 00100 |
| D | 2 | 00010 |
| E | 1 | 00001 |
| F | 2 | 00001 |
| Bcast | 1 | 10101 |
| Bcast | 2 | 01011 |
| Mcast1 | 1 | 10100 |
| Mcast1 | 2 | 01000 |
| Mcast2 | 2 | 01010 |

* = Bridge Port Configured for VEPA attach

VLAN 1 Tag Mask = UUUUT
VLAN 2 Tag Mask = UUUUT

# VEPA Multicast and VLANs

A -> Bcast




1. Dst Lookup = 10101
2. Src Lookup = 10000
3. Delivery Mask = 00101

(dst & ~src)

Example: VEPA Address Table

| Address | VLAN | VF Mask |
|---|---|---|
| A | 1 | 10000 |
| B | 2 | 01000 |
| C | 1 | 00100 |
| D | 2 | 00010 |
| E | 1 | 00001 |
| F | 2 | 00001 |
| Bcast | 1 | 10101 |
| Bcast | 2 | 01011 |
| Mcast1 | 1 | 10100 |
| Mcast1 | 2 | 01000 |
| Mcast2 | 2 | 01010 |

VLAN 1 Tag Mask = UUUUT

VLAN 2 Tag Mask = UUUUT

* = Bridge Port Configured for VEPA attach

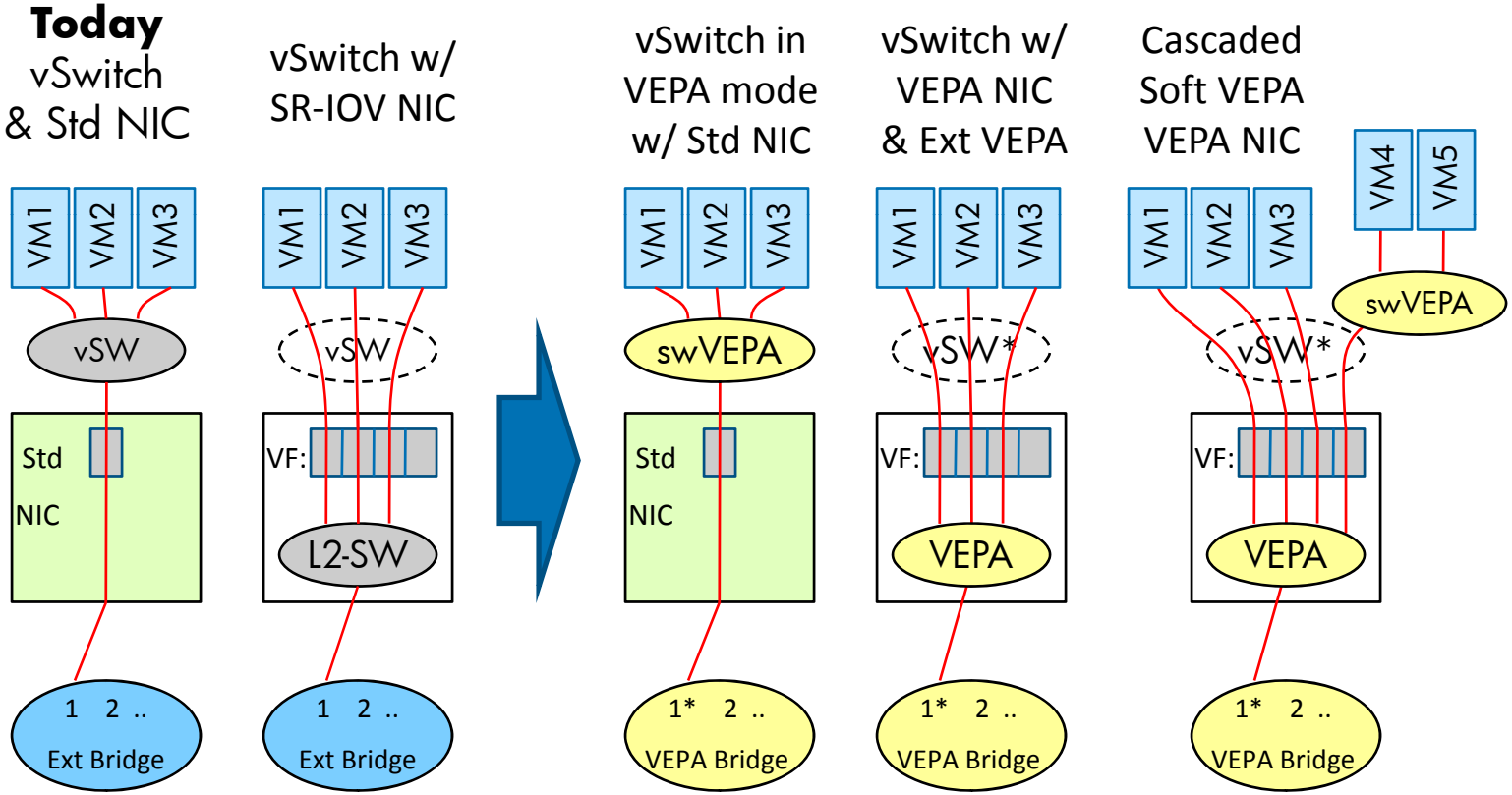
# Untagged VEPA Limitations and Issues

1. Topology Restrictions
   - VEPA must be directly attached to a Bridge in special 'turn-around' mode
   - Multiple VEPAs can be stacked, only the Bridge port can do 'turn-around'
2. Promiscuous Mode
   - VEPA needs pass all multicast, broadcast and unknowns up to a software VEPA above a port in promiscuous mode if multiple source MACs are above
   - A vSwitch attached to a VF of a VEPA needs to know the multicast flooding behavior to avoid address learning thrash.
3. VM Recommendations
   - VEPA Attached VMs should not forward between multiple vNICs (e.g. Transparent Firewall)
   - VMs should be application end-points, not network forwarding devices
4. Table Sizes
   - VLAN awareness requires tag and pass thru configuration
   - Multicast address filters are per VLAN per VF

# Untagged VEPA Work Items, Impact

- Define Port Peer Mode Negotiation (LLDP?)
  - VEPA port and terminus entities
  - VEPA port may have stackable mode (no turnaround)
- NIC vendors
  - Negotiation of VEPA mode with port peer
  - Per VF multicast membership and MAC assignment
    - OS -> Driver -> VF hardware
  - Ingress packet data replication
    - MAC/VLAN match could go to multiple VF ingress queues
- Bridge Vendors
  - Negotiation of VEPA mode with port peer
  - Define Turnaround mode on Bridge ports to VEPAs
  - Otherwise process like any other packet

# Adding VEPA to Today's Solutions



# Tagging Schemes

## Objectives:

1. Eliminate the need for the VEPA to have a MAC address table
2. Provide explicit indication of what VFs need to receive a packet

Note: If tagging scheme includes address encapsulation then VEPA and external Bridge need not be directly connected

## Existing Candidates:

1. MACSec Tag (aka SecTAG)
2. 802.1Q Provider Tag (limited combinations)
3. 802.1ah Backbone Provider Tag (encapsulation)

# MACSec Scheme

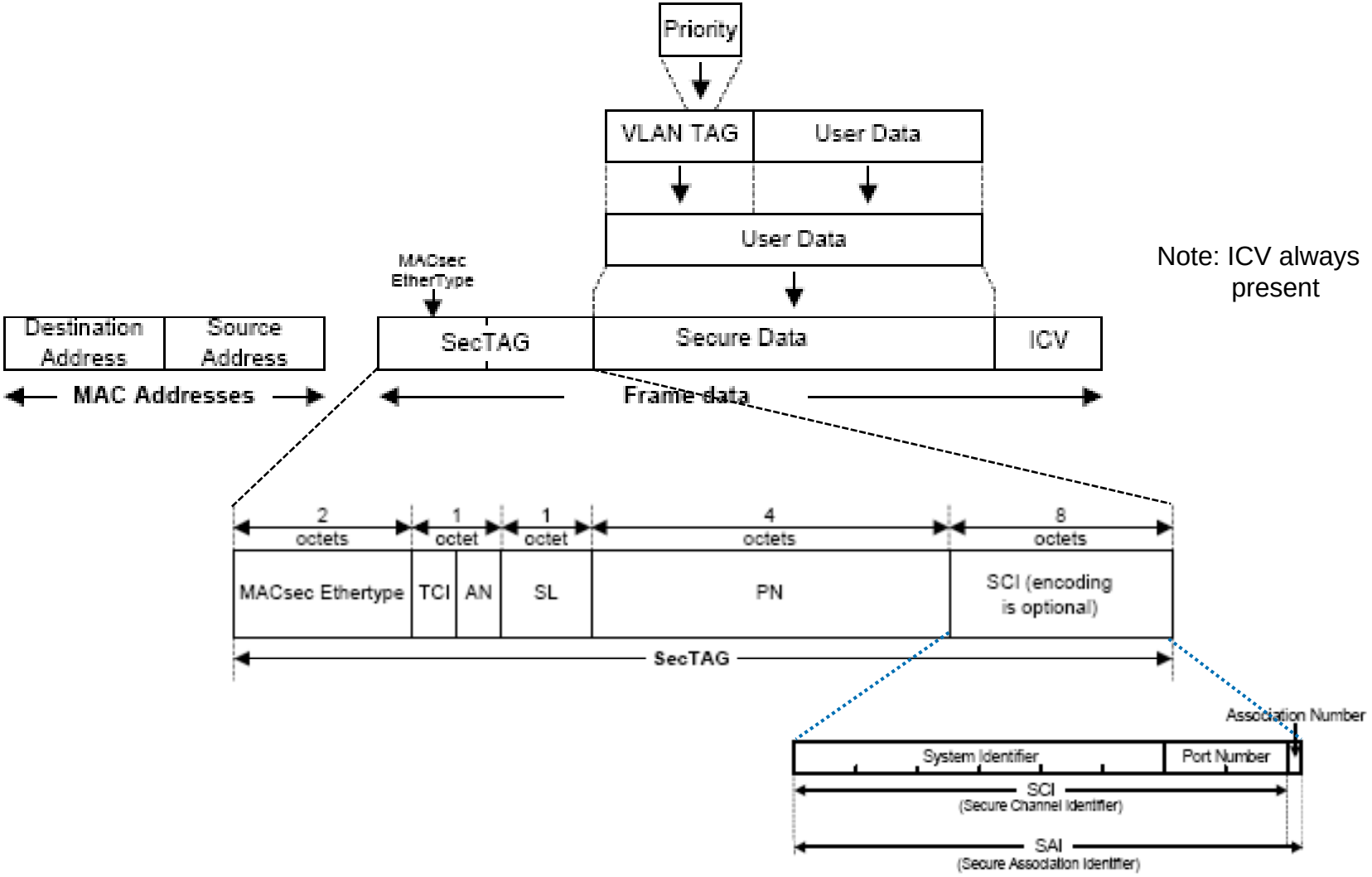
Advantages:

1. Leverages existing standard for virtual ports and tags
2. Already includes the ability to secure connections between VEPA and bridges

Disadvantages:

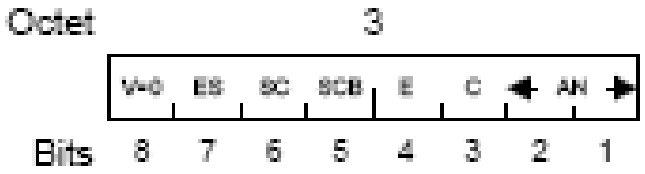
1. Small modifications to existing specification are required
2. Requires between 16-32 bytes of overhead
3. VEPA and bridge must be directly attached

# MACSec Frames




Note: ICV always present

# SecTAG Control Information



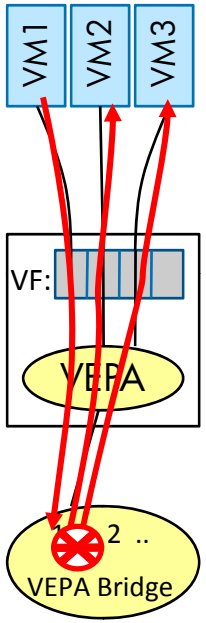

```
V   := Version bit (v=0)
ES  := End-Station
SC  := SCI included
SCB := Single copy broadcast (EPON)
E   := Encryption
C   := Changed Text
AN  := Association Number
```

- Version is 0, but if necessary could bump to 1 and define additional bits (not desired)
- End-Station bit needs to be 0 to allow SCI to be used to encode source virtual port number
- SCI must be included to allow 8 bytes of SCI to be included
- Single copy broadcast can only be used when SC is 0, but we need SC to encode port group
- Encryption may or may not be used as desired, but ICV is always included
- Changed Text is only set if the user data has been encrypted

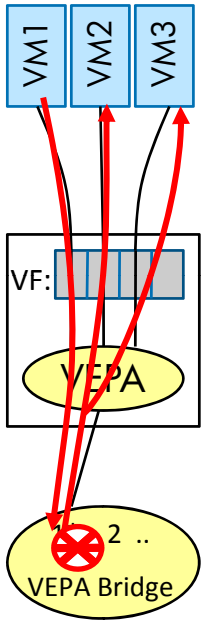
# Making MACSec work on a VEPA

- Always include a SecTAG on all traffic between VEPA and external bridge
- Always include the SCI in each SecTAG
- VEPA uses SCI to indicate internal virtual port number
- Bridge uses SCI to indicate VEPA internal port number and/or multi-destination port groups
- Multicast/Broadcast behavior (choices)
  - Bridge replicates multicast/broadcast
  - Allow Single Copy Broadcast bit to be set while including SCI from bridge
- Protocol between VEPA and bridge is needed to define multi-destination mappings and VEPA port resource limits.

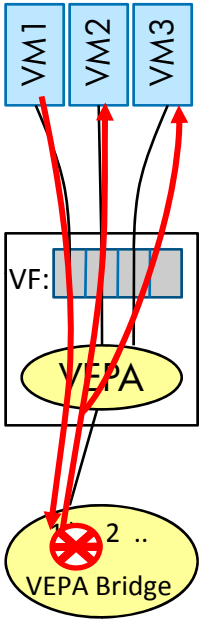
# Tagged Multicast/Broadcast Behavior




**Bridge Replication**
- Unique copy for each VF
- SCI describes dest VF
- Almost MACSec today




**VEPA Replication (1)**
- Unique SCI for port set
- Limited combinations
- Large bridge tables needed




**VEPA Replication (2)**
- Unique SCI for group
- Source VF encoded in SCI to allow source filtering
- New SCI definition

# SecTAG Scheme Details

with VEPA replication

1. Bridge creates virtual ports per 802.1AE specification
2. Bridge creates Single Copy Broadcast port for each VEPA
3. Bridge virtual ports are associated with VEPA virtual functions
4. A Null cipher is desired that also eliminates ICVs.
5. Bridge sends single multi-destination frame to VEPA on a physical port
6. For VEPA to replicate broadcasts:
   a) Bridge sends SecTAG frames with SCB bit set and SCI
   b) SCI port identifies explicit set of VEPA ports for replication
   c) SCI system identifier from bridge identifies VEPA and source port
7. VEPA must communicate to bridge virtual function configuration
   a) Number of virtual functions
   b) VLAN configuration
   c) Known multicast filter membership
8. Bridge must communicate to VEPA multi-destination definitions

# Current Specification Issues

1. Presence of ICV in all SecTAG frames implies required key management
   Change: Define a null cipher that doesn't require ICVs

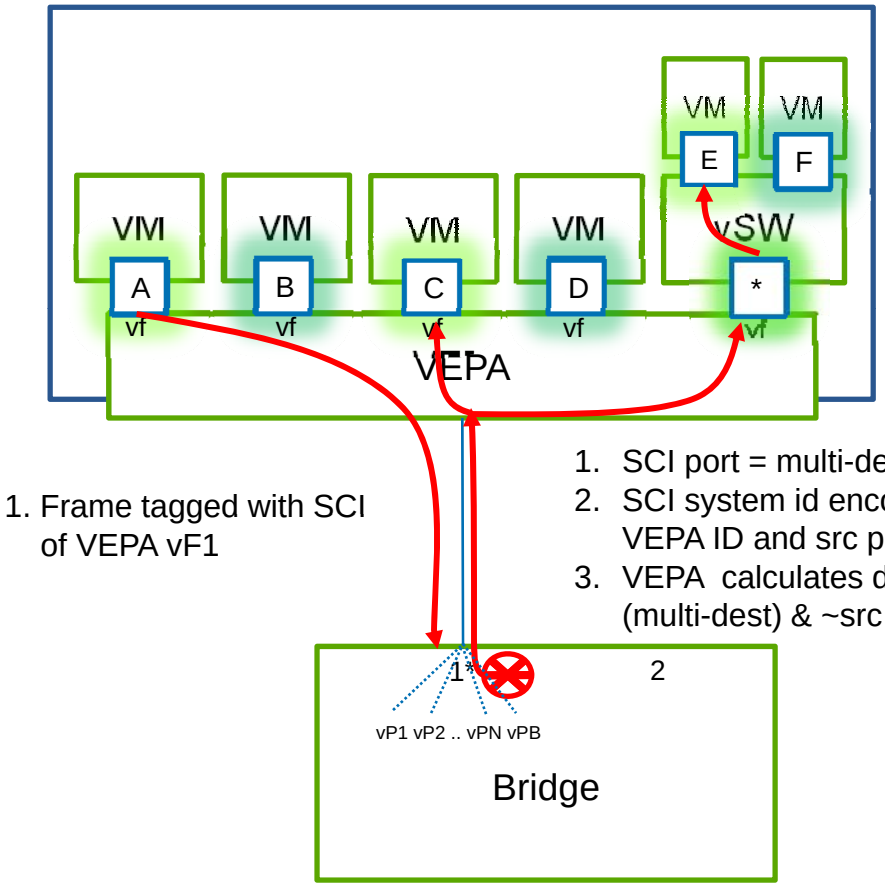
2. Single copy broadcast frames don't also allow presence of SCI
   Change: Remove text preventing behavior

3. Using port number to represent multi-destination replication requires unique combinations to eliminate sources
   Change: Encode VEPA source port in SCI system identifier on frames from bridge. Or….
   Change: Modify SecTAG to include additional source port field

4. Current definition of SCI system identifier does allow other uses
   Change: System identifier on frames from bridge could identify VEPA

# VEPA Multicast and VLANs

A -> Bcast




1. Frame tagged with SCI of VEPA vF1

1. SCI port = multi-dest
2. SCI system id encodes VEPA ID and src port
3. VEPA calculates delivery mask (multi-dest) & ~src

* = Bridge Port Configured for VEPA attach

Example: VEPA Port Replication Table

| SCI Port # | VF Internal Mask |
|---|---|
| 1 | 100000000000 |
| 2 | 010000000000 |
| 3 | 001000000000 |
| 4 | 000100000000 |
| 5 | 000010000000 |
| … | 0000000X0000 |
| N | 000000000001 |
| N+1 (V1) | 101010000000 |
| N+2 (V2) | 010110000000 |
| … | |
| M (Mcast1 V1) | 101000000000 |
| M+1 (Mcast1 V2) | 010000000000 |
| M+2 (Mcast2 V2) | 010100000000 |

VLAN 1 Tag Mask = UUUUT

VLAN 2 Tag Mask = UUUUT

# Conclusion

1. Existing SW/HW vSwitches are not going away, so they should follow existing 802.1 standards
2. Adding untagged VEPA mode allows external traffic flow with high leverage and little impact to existing solutions
3. Tagged VEPA mode already exists using MACSec model requiring external bridge replication
4. Modest adjustments to MACSec could be done to support tagged mode VEPA replication
5. Yet another tagging scheme is not needed