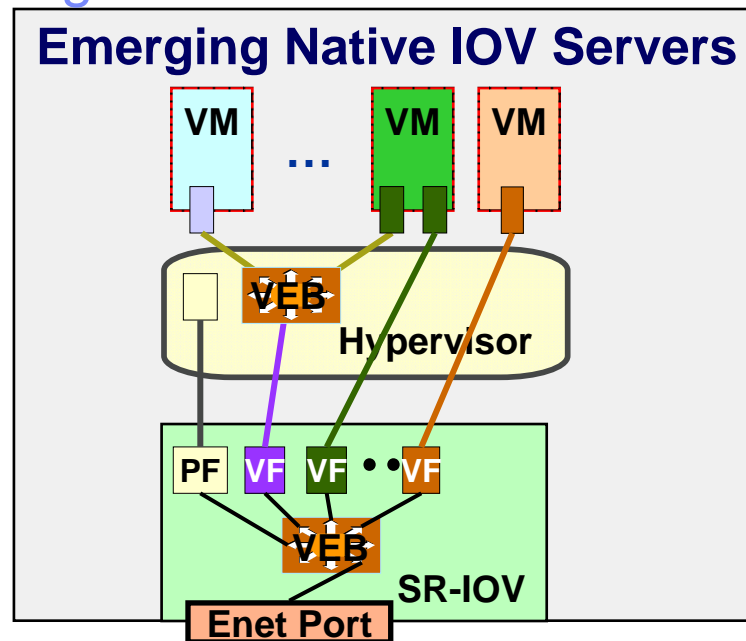
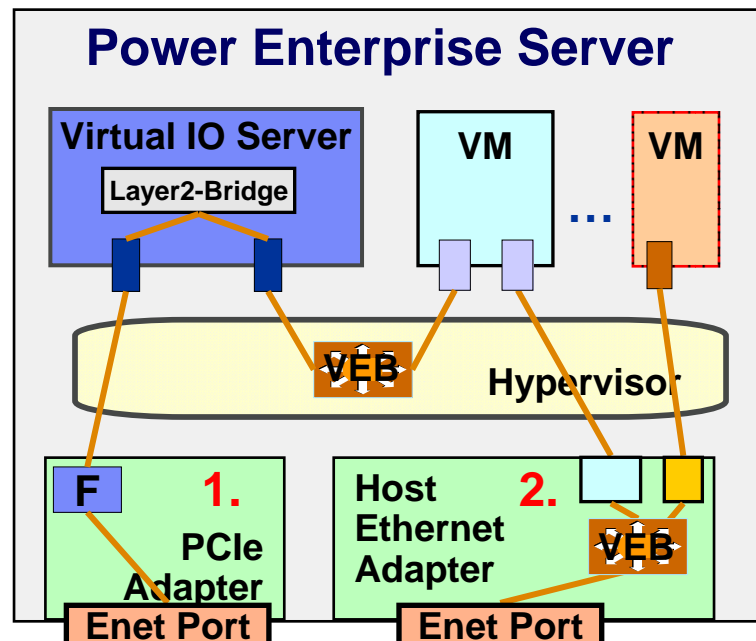


# Automated Ethernet Virtual Bridging

Renato Recio, DE,  
IBM Data Center Networking CTO

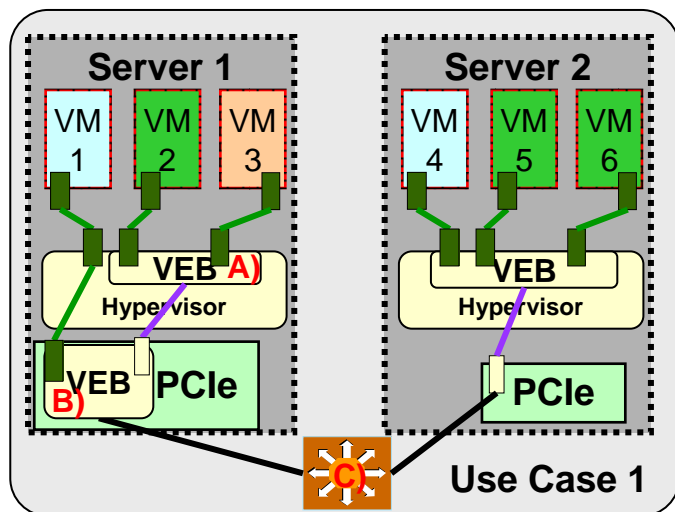
Omar Cardona,  
AIX Ethernet Virtual Switching Development

## Server Ethernet Virtualization Technologies



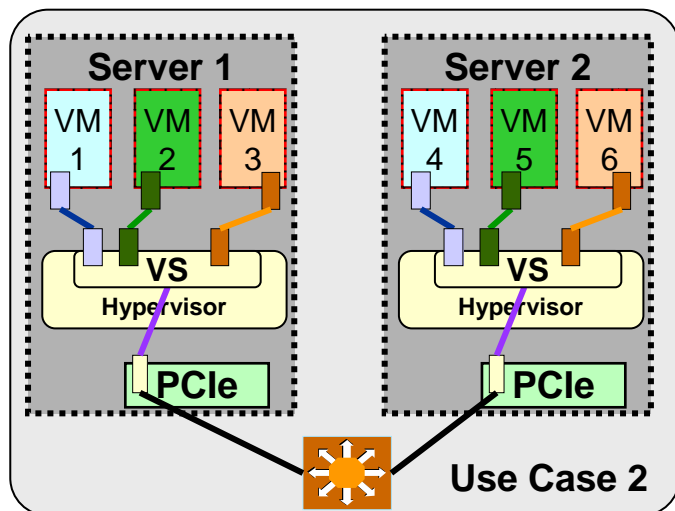
- Today's virtual IO technology (e.g. Power)
  1. IO shared through Virtualization Intermediary (e.g. VIOS on Power).
    - All IO is performed through VI.
    - Adds overhead to every IO operation.
  2. Native IO Virtualization (IOV)
    - IO directly shared by adapter hardware.
    - Adds Adapter Virtual Ethernet Bridge (VEB)
- IBM worked with the industry (PCI SIG) to standardize IO Virtualization technologies.
  - PCIe adapters are coming to market that support direct IO sharing through multi-queue, multi-function or Single-Root IO Virtualization.
- PCIe IOV enables VMs to bypass the Hypervisor and directly share a PCIe adapter.
  - PCI Special Interest Group (SIG) standardized the north side mechanisms.
  - However, the south side mechanisms (e.g. VEB), weren't standardized (not in PCI SIG scope).

## Server Virtual Ethernet Bridging Use Cases & Placement Options



▪ Two Virtual Ethernet Bridging (VEB) use cases emerging:

- **Use case 1: *Flat (singly restrictive) layer-2 fabric*** with common access/security controls for all systems and VMs.
  - Targets clusters of virtualized systems that run application(s) with common access controls & want to exploit CPU trends (i.e. more cores → more VMs → more local VM-VM IO).
  - Access controls need not be as sophisticated as those available in external switches.

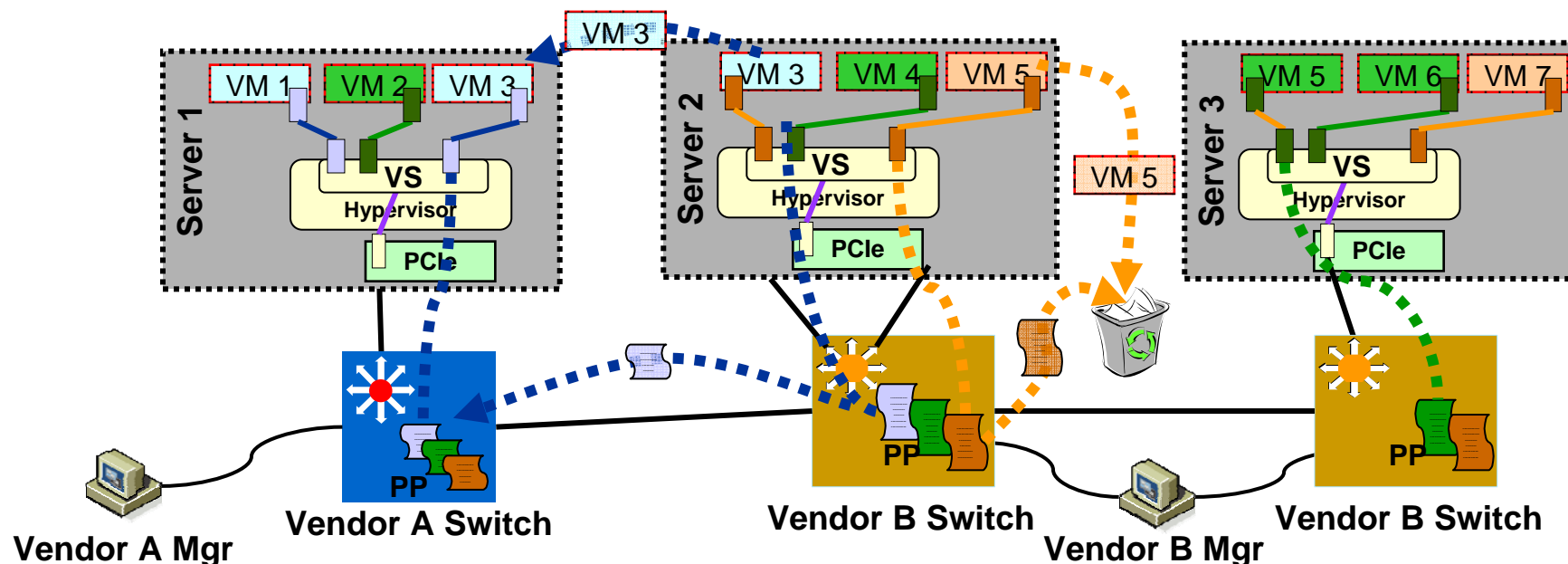


- **Use case 2: *Profile restrictive layer-2 fabric*** with access/security control profiles for each system and VM type.
  - Targets multi-tier environments that want to host multiple VM types (e.g. Web, e-mail, SAP, database) in the same layer-2 fabric.
  - In this model local VM-VM communications require advanced, layer-2 access controls.

▪ VEB placement options:

VM-VM switching	A) Software	B) Adapter	C) External Switch
<b>Bandwidth</b>	10 GB/s	4 → 8 GB/s	1 GB/s
<b>Latency</b>	< 1 us	< 1 us	2+ us
<b>CPU overhead</b>	High	Low	Low

## Virtual Ethernet Bridges Requirements

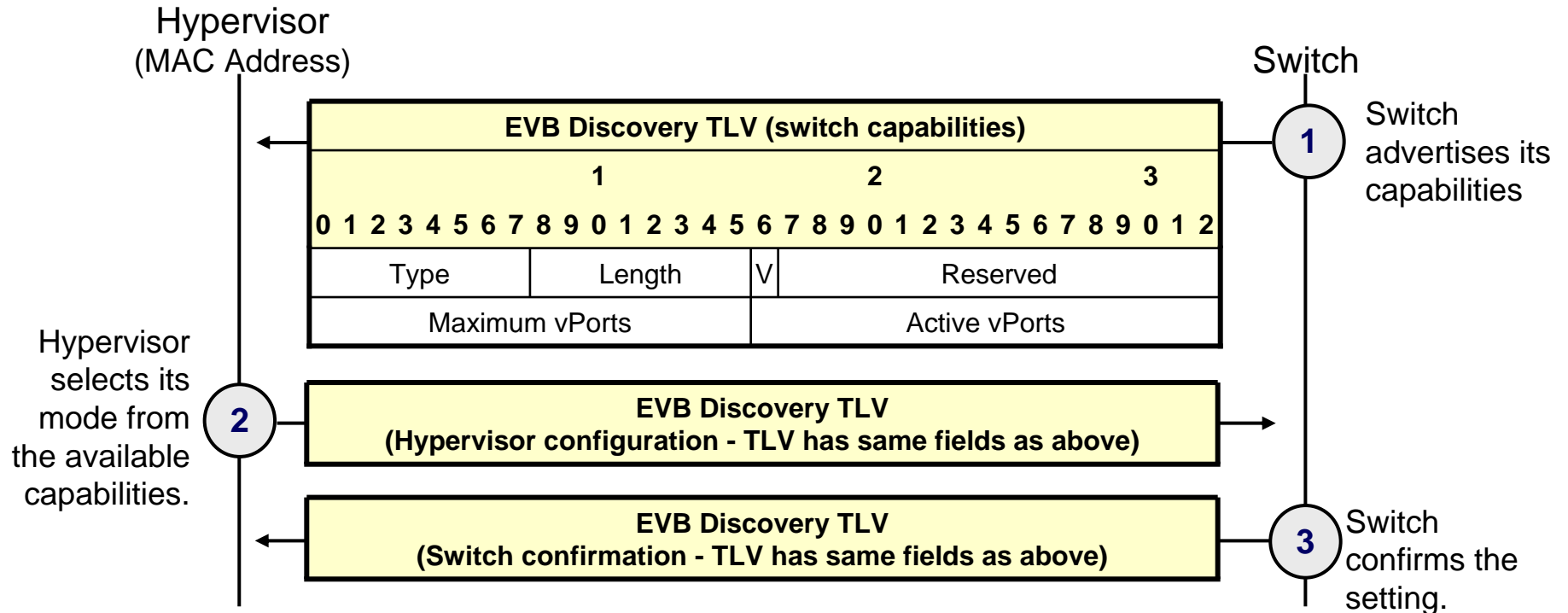


VM-VM bridging needs to:

- Enable sophisticated switch port profiles (i.e. access, traffic & security controls).
- Be visible to network management tools.
- Enable automated VM migration, including migration of the VM's switch port profile across layer-2 switches, homogeneous (e.g. vendor B → vendor B above) & heterogeneous (vendor B → vendor A).
- Enable the use case and VEB placement options covered earlier.
- Differentiate between a re-incarnated and a migrated MAC Address:
  - A **re-incarnated** MAC Address (e.g. **VM 5's MAC Address**) is one that was previously in use by a recently destroyed VM and is now in use by a different VM, which may require a completely different external network port profile.
  - A **migrated** MAC Address (e.g. **VM 3's MAC Address**) is one that is associated with a VM that has been migrated across two physical servers in the fabric and retains the same port profile association after the VM migration.



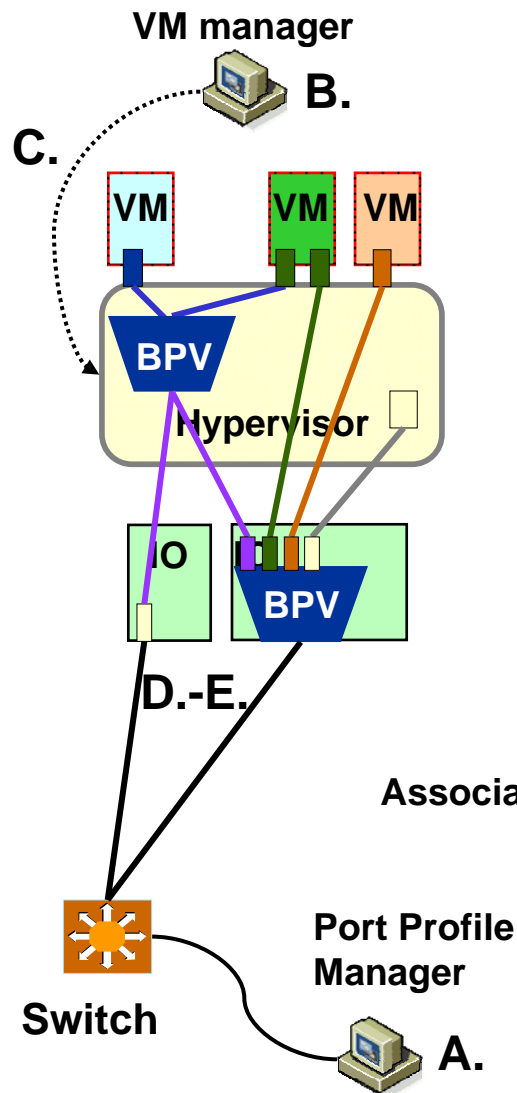
## Proposed Ethernet Virtual Bridging (EVB) Data Center Bridging eXchange (DCBX) protocol Type, Length, Value (TLV)



### Where in above TLV:

- V = Ethernet Virtual Bridging mode:
  - If VEB = 0, then Server Virtual Ethernet Port Aggregation is used. Where all VM-VM bridging is performed by the external switch.
  - If VEB = 1, then Server Virtual Ethernet Bridging is used. Where all VM-VM bridging is performed by either the Hypervisor or Adapter.
- Maximum vPorts = Maximum number of vPorts the Hypervisor may have activate.
- Active vPorts = Active number of vPorts the Hypervisor currently has activate.

## B. Automating External Port Profile Migration



To migrate the Port Profile associated with a Virtual Machine's virtual NIC port's MAC Address, we propose an Automated Migration of a Port Profile mechanism:

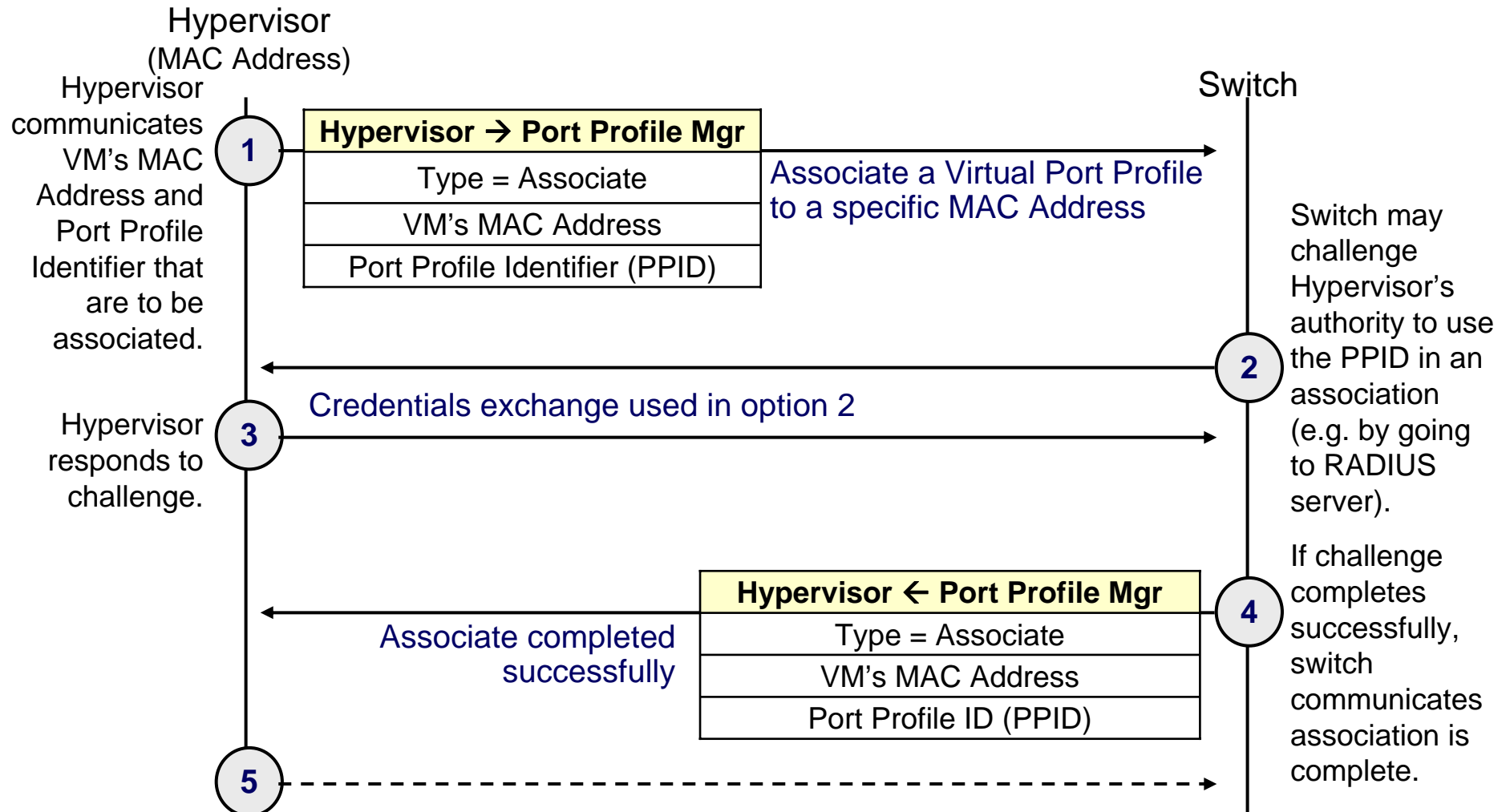
- The approach allows automated migration of a VM's external network profile on activation and during VM migration, by:
  1. Using **management plane** protocols to:
    - A. Create/modify/destroy virtual switch port profiles; and
    - B. Communicate "port profile" identifiers" to the Hypervisor manager
    - C. Hypervisor manager communicates "port profile to VM associations" to Hypervisor.
  2. Using a **control plane** protocol to:
    - D. Pre-associate a port profile with a specific VM's MAC Address;
    - E. Associate a port profile with a specific VM's MAC Address; and
    - F. De-associate & migrate that profile when the VM migrates.

Hyp. → Switch / Port Pr. Mgr
Associate
VM MAC Address
Port Profile ID

Hyp. ← Switch / Port Pr. Mgr
Associated
VM MAC Address
Port Profile ID

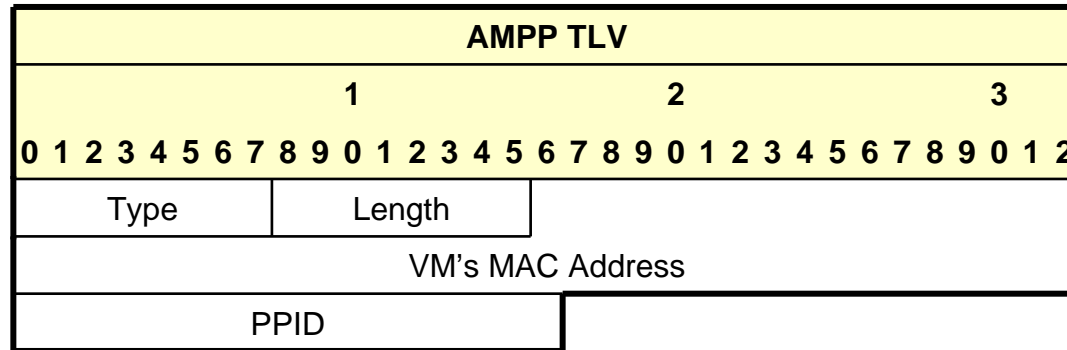
- Hypervisor & PCI SR-IOV adapters implement BPV (Bridge Port Virtualizer) to associate set switching mode & AMPP usage.
  - Select internal vs external bridging
  - Select Automated Migration of a Port Profile (AMPP)

## B. Automating External Port Profile Migration (AMPP) → Associate



Note: Hypervisor may send a gratuitous ARP with VM's MAC Address, to accelerate learning. Hypervisor can now bring up VM and VM can begin using the MAC Address.

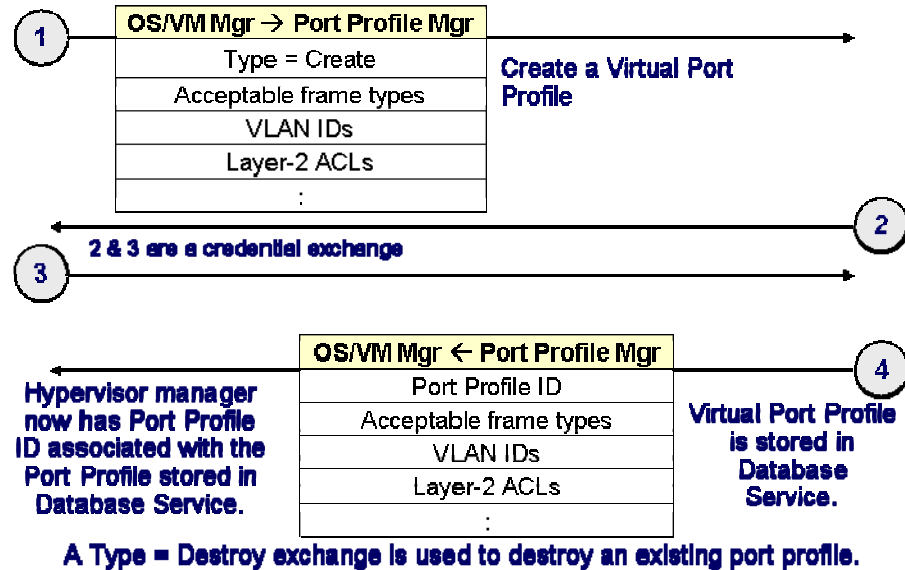
## B. Automating External Port Profile Migration → AMPP TLV



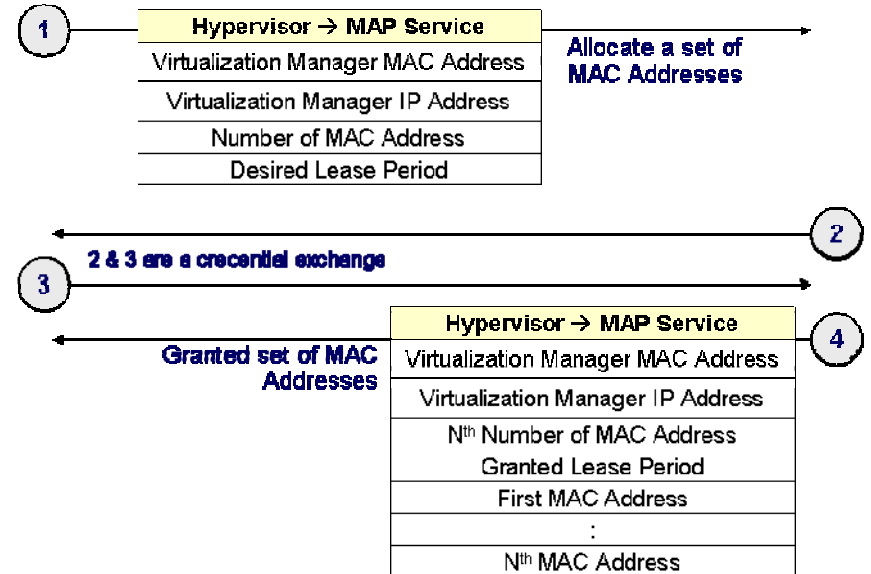
- **Types**
  - Pre-Associate - used, specially during migration, to provide an early warning to the switch that a new VM virtual port instance will be sharing the port; the MAC Address of the VM virtual port instance and the Port Profile Identifier the VM virtual port instance will associated with.
  - Associate - used to associate a new VM virtual port instance's VM MAC Address to the PPID.
  - De-Associate - used to de-associate a VM virtual port instance's VM MAC Address from the PPID
- **Length**
  - 10 octets.
- **VM's MAC Address**
  - The VM's virtual port instance's MAC Address that the server's virtualization infrastructure is asking to be associated with the Port Profile Identifier.
- **PPID**
  - A Port Profile Identifier for a specific switch port profile.



## Additional Areas for *Possible* Standardization

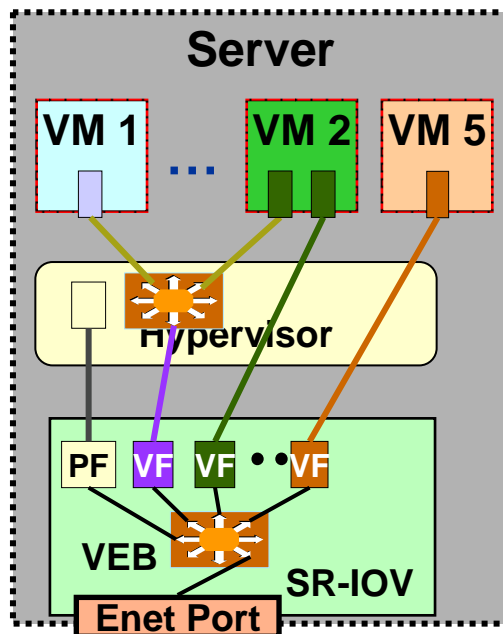


- Problem
  - Simplifying Port Profile management, specially in environments with heterogeneous layer-2 switches.
- Port Profile Management Protocol
  - Used by Hypervisor manager to create/destroy Port Profiles through a control plane protocol between the Hypervisor manager and the Port Profile manager.
  - Would require standardization of a base set of Port Profile attributes, with additional proprietary attributes
- Port Profile Identifier Dissemination Protocol
  - Used by the Port Profile manager to distribute the Port Profiles to switches.



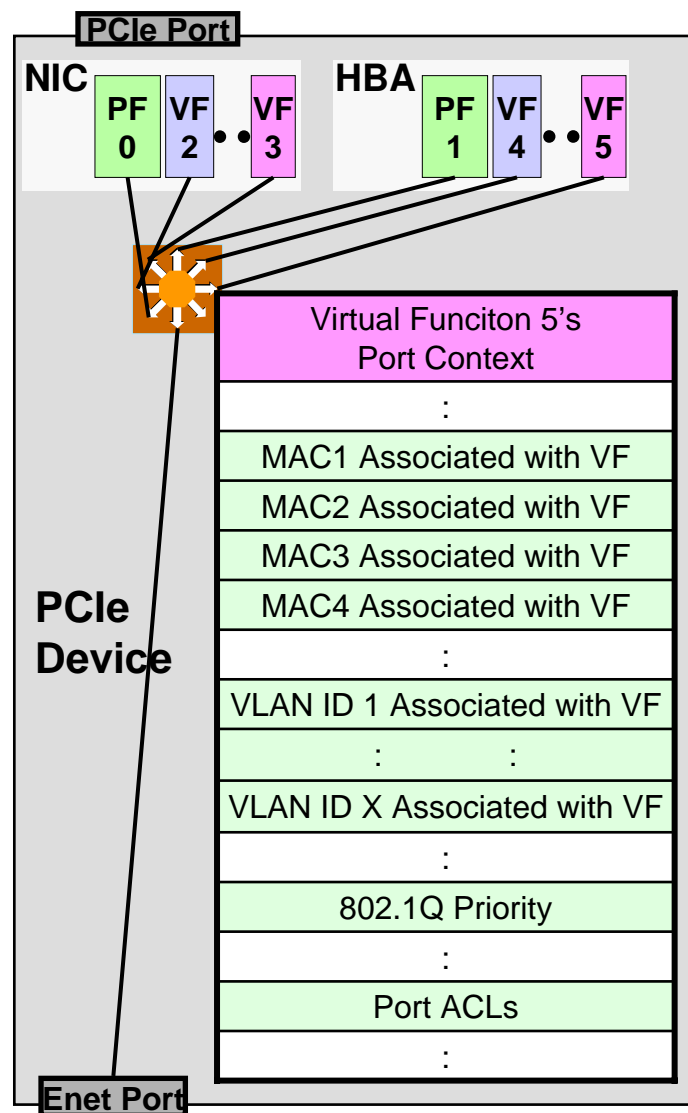
- Problem
  - As layer-2 fabric sizes continues to grow, the possibility of multiple virtualization domains in the same layer-2 fabric increases and the risk of having collisions in locally administered MAC addresses.
- MAC Address Provider Service Protocol
  - Used to assign or remove a set of MAC addresses to a specific device (e.g. a Hypervisor Manager), where the assignment has a “time to live” lease period.

## Virtual Ethernet Bridging for Directly Shared IO Adapters



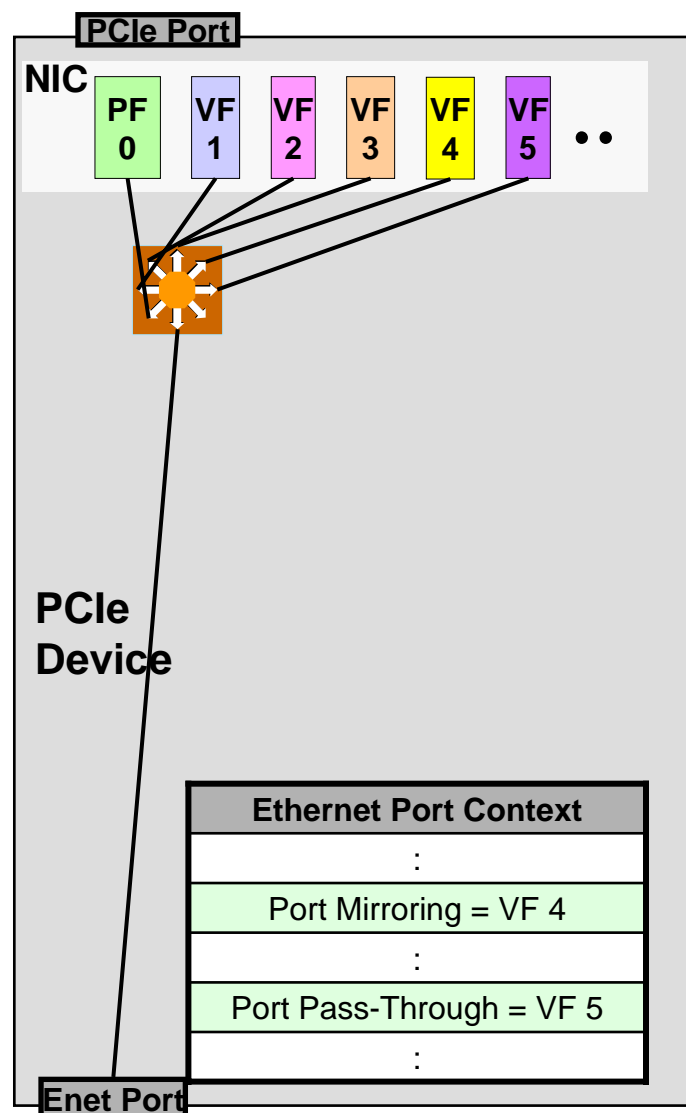
- As covered earlier, **direct sharing** allows VM-Adapter communication to bypass the Hypervisor
  - This approach circumvents the resource and processing overhead inherent in Hypervisor based Ethernet bridging.
- The PCI specification only provided 8 Functions per Device.
  - Limits the ability to directly share an adapter by multiple VMs.
- PCI SIG defined enhancements to improve the ability of a PCI adapter to be directly shared by multiple VMs.
  - SR-IOV “Alternate Route Identifier” increased the number of Functions to 256.
  - The SR-IOV Base Specification increased the number of functions further to be able to use the full 16 bit “Routing Identifier” space, minus the bits used to identify intervening busses.
- However, SR-IOV did not define the Virtual Ethernet Bridging (VEB) mechanisms needed to bridge Ethernet traffic between VMs and the external port.
  - PCI SIG viewed VEB mechanisms as outside its scope.
  - The following slides describe options for these VEB mechanisms.

# 1) MAC/VLAN Management and Access Controls



- A mechanism is needed to prevent MAC Address spoofing between VMs sharing the same PCIe adapter.
- One approach is to have the Hypervisor populate the VEB with the allowed set of MAC Addresses that a given VM is allowed to use.
  - The VM can then select a single MAC Address from a set of MAC Addresses the Hypervisor previously populated.
  - On egress, the adapter compares the Source MAC Address used by the VM to the MAC Address stored in the VF context.
    - If the frame's Source MAC Address equals one of Hypervisor populated MAC Addresses, the frame is forwarded to its destination.
    - Otherwise the frame is discarded.
- A similar mechanism can be used to associate & check:
  - The VLAN Identifier(s) associated with the VM port;
  - The 802.1Q Priority associated with the VM's port; and
  - Access controls associated with the VM's port.

## 2) Network Security and Diagnostics

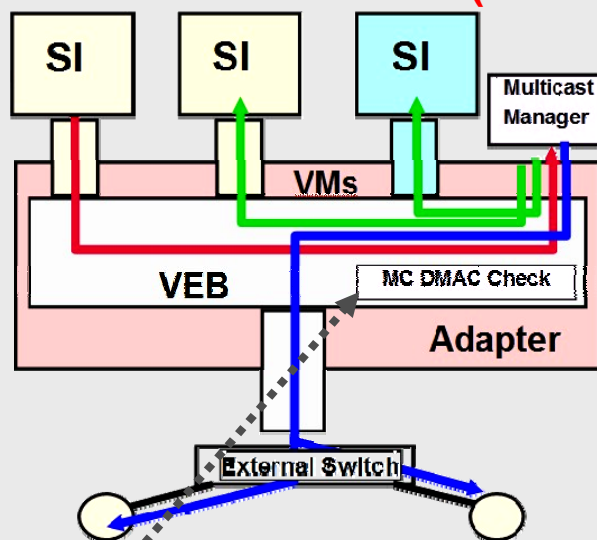


- In a data center, security appliances are used to provide Intrusion Detection and Prevention (IDP) functions.
  - Virtual server consolidation requires the ability to have VM-to-VM communications inspected by security appliances.
  - For directly shared adapters a mechanism is needed to enable intrusion detection and prevention appliances.
  
- We propose two mechanisms to meet the above need:
  - Port-Mirroring supports a virtual intrusion detection appliance by **mirroring** all frames received by the adapter to the Virtual Port used by the Virtual Appliance (e.g. to the VF used by the Virtual Appliance).
    - If a frame is benign, it is silently dropped.
    - If it is found to have a possible malignancy, an alert is surfaced through the virtual security appliance's manager
  
  - Port-Pass-Through supports a virtual intrusion prevention appliance by **forwarding** all frames received by the adapter through to the Virtual Port used by the Virtual Appliance.
    - If forwarded frames are determined to be benign, they are forwarded to the destination VM (or VMs for multicast/broadcast) or external network through the VEB.
    - Otherwise the frames are dropped.

## 4) Multicast Management

Directly shared adapters must be able to forward Multicast (MC) and Broadcast (BC) frames.

### 2. SW based multicast (shown)



MC Address Table	
MB Address 1	Pointer to VF list 1
:	:
MB Address i	Pointer to VF list i
:	:
MB Address N	Pointer to VF list N

List of VFs Associated with MC Address i
VF Number of first VF
:
VF Number of last VF

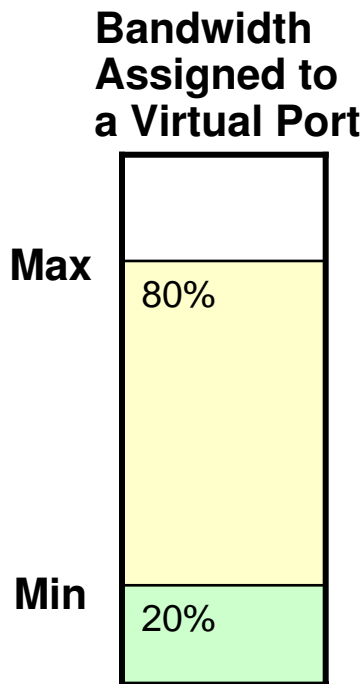
One table per MB Address.

- Note the following uses runtime para-virtualization calls from the VM to the Hypervisor to dynamically adds and deletes MC addresses.
  - Options for where to place MC & BC forwarding:
    1. Through a special purpose (Physical or Virtual) function that is assigned for MC/BC forwarding; or
    2. Perform function in the adapter.
1. SW based multicast
    - The adapter VEB redirects MC/BC traffic to the PCIe function assigned to perform MC/BC forwarding.
    - For Multicast, the PCIe MC/BC function forwards the frame to each VF associated with the Multicast Address, except to source. For Broadcast, the PCIe MC/BC function forwards the frame to all VFs.
  2. Adapter based multicast/broadcast forwarding
    - For Multicast, the PCIe VEB forwards the frame to each VF associated with the Multicast Address, except to source.
    - For Broadcast, the PCIe VEB forwards the frame to all VFs.

## 6) Traffic Scheduling Across VFs

- A mechanism is needed to schedule traffic across PCIe functions (PCIe VF, PF or Function).
- Proposed approach consists of using 3 variables to schedule traffic across functions:

- **Maximum capacity** defines the maximum percentage of the egress link bandwidth the adapter will make available to the function even if there is no link contention. That is, the function's egress bandwidth will not exceed this value.
- **Minimum capacity** defines the minimum percentage of the egress link bandwidth the adapter must make available to the function.
- **Weight** defines a weighted priority at which each function competes for excess capacity on the link. The weight value allows for prioritizing the functions relative to each other such that a higher priority function is favored over a lower priority function by the weight associated with each.
  - The weight allows for prioritizing virtual ports relative to each other, so that higher priority virtual ports are favored to get the excess capacity over the lower priority virtual ports



---

## Conclusion

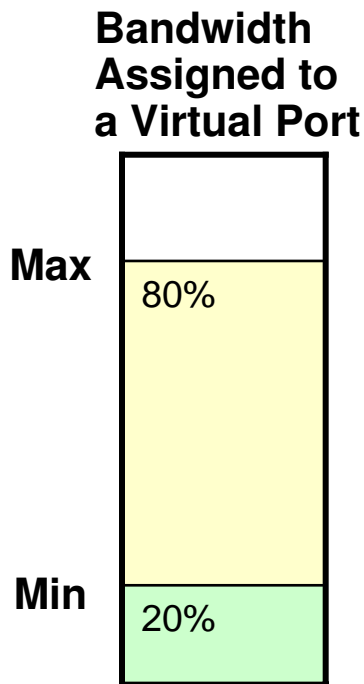
- This paper described use cases for when to perform Ethernet virtual bridging within the server vs in the external fabric.
  
- For the case where Ethernet Virtual Bridging is performed in the fabric, we described mechanisms for:
  - Automating migration of an external switch's port profile when a VM is migrated across physical servers.
  - Providing unique locally administered MAC Addresses within a layer-3 domain.
  
- For the use case where Ethernet virtual bridging is performed within the server, we described mechanisms for the necessary VEB functions.

---

# Back-ups

## Example of Traffic Scheduling Across VFs

- Take an example where you have 5 VPs, each configured to get entitled capacity of 20% and max capacity of 50%.
  - VP1 has a weight of 1, VP2 has a weight of 2, VP3 has a weight of 3, VP4 has a weight of 4, and VP5 has a weight of 5. This gives you a total weight of 15.
  - Assume in a given instant that VPs 2-5 are all consuming their entitled capacity - so 80% of the BW is being consumed and they all need more. VP1 is idle so is not consuming hardly any of its entitled capacity so the excess available capacity is 20%.
    - The sum of the weights of the VPs competing for the excess capacity is 14.
    - The available excess capacity (that unused 20%) should be divided proportionally based on these weights.
      - VP5 would get  $5/14$  - so it would consume 27.1% of the total BW
      - VP4 would get  $4/14$  - so it would consume 25.7% of the total BW
      - VP3 would get  $3/14$  - so it would consume 24.3% of the total BW
      - VP2 would get  $2/14$  - so it would consume 22.9% of the total BW



**Weight =  $N/M$ , where  $M$  is the sum of the weights for the Virtual Ports that have consumed their min capacity and are competing for excess capacity, which can be up to but no more than the max**