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
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

Virtual Machine
Higher Layers
MAC Client

Higher Layers
MAC Relay
MAC Client

MAC

Virtual Machine
Higher Layers
MAC Client

Higher Layers
MAC Client

Routing Protocols, Storage Protocols, Availability Protocols, IDS/IPS, etc

Access Control Lists

Traffic Monitoring

Port Mirroring

PAE
LLDP
Higher Layers
(Bridge Protocol)

MAC Relay

MAC Sec

MAC

MACRelay

MAC Sec

MAC Sec
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

- Virtual Machine
- Virtual NIC (vNIC)
- Virtual Station Interface (VSI)
- Physical NIC
- Uplink
- Switch Port
- Physical End Station
- VEB/VEPA
- NIC Team
- Adjacent Switch
- Expander port
- Software VEB/VEPA
- expander
- Apps
- GOS
- Apps
- GOS
- Apps
- GOS
- Apps
- GOS
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

<table>
<thead>
<tr>
<th>DST MAC</th>
<th>VLAN</th>
<th>Copy To (ABCDEF Up)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>100000 0</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>010000 0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>001000 0</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>000100 0</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>000010 0</td>
</tr>
<tr>
<td>F</td>
<td>2</td>
<td>000001 0</td>
</tr>
<tr>
<td>Bcast</td>
<td>1</td>
<td>101010 1</td>
</tr>
<tr>
<td>Bcast</td>
<td>2</td>
<td>010101 1</td>
</tr>
<tr>
<td>Multicast</td>
<td>1</td>
<td>101010 1</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>1</td>
<td>100010 1</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>2</td>
<td>010101 1</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>1</td>
<td>000010 1</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>2</td>
<td>000000 1</td>
</tr>
</tbody>
</table>
VEB Unicast Example

**SRC = A; DST = C**

**VEB Address Table**

<table>
<thead>
<tr>
<th>DST MAC</th>
<th>VLAN</th>
<th>Copy To (ABCDEF Up)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>100000 0</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>010000 0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>001000 0</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>000100 0</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>000010 0</td>
</tr>
<tr>
<td>F</td>
<td>2</td>
<td>000001 0</td>
</tr>
<tr>
<td>Bcast</td>
<td>1</td>
<td>101010 1</td>
</tr>
<tr>
<td>Bcast</td>
<td>2</td>
<td>010101 1</td>
</tr>
<tr>
<td>MulticastC</td>
<td>1</td>
<td>101010 1</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>1</td>
<td>100010 1</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>2</td>
<td>010101 1</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>1</td>
<td>000010 1</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>2</td>
<td>000000 1</td>
</tr>
</tbody>
</table>

* Promiscuous VSI
**VEPA Unicast Example**

**SRC = A; DST = C**

**VEPA Address Table**

<table>
<thead>
<tr>
<th>DST MAC</th>
<th>VLAN</th>
<th>Copy To (ABCDEF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>100000</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>010000</td>
</tr>
<tr>
<td><strong>C</strong></td>
<td>1</td>
<td><strong>001000</strong></td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>000100</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>000010</td>
</tr>
<tr>
<td>F</td>
<td>2</td>
<td>000001</td>
</tr>
<tr>
<td>Bcast</td>
<td>1</td>
<td>101010</td>
</tr>
<tr>
<td>Bcast</td>
<td>2</td>
<td>010101</td>
</tr>
<tr>
<td>MulticastC</td>
<td>1</td>
<td>101010</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>1</td>
<td>100010</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>2</td>
<td>010101</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>1</td>
<td>000000</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>2</td>
<td>000000</td>
</tr>
</tbody>
</table>

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 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**

<table>
<thead>
<tr>
<th>DST MAC</th>
<th>VLAN</th>
<th>Copy To (ABCDEF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>100000</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>010000</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>001000</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>000100</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>000010</td>
</tr>
<tr>
<td>F</td>
<td>2</td>
<td>000001</td>
</tr>
<tr>
<td>Bcast</td>
<td>1</td>
<td>101010</td>
</tr>
<tr>
<td>Bcast</td>
<td>2</td>
<td>010101</td>
</tr>
<tr>
<td>MulticastC</td>
<td>1</td>
<td>101010</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>1</td>
<td>100010</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>2</td>
<td>010101</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>1</td>
<td>000000</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>2</td>
<td>000000</td>
</tr>
</tbody>
</table>
‘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**

<table>
<thead>
<tr>
<th>DST MAC</th>
<th>VLAN</th>
<th>Copy To (ABCDEF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>100000</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>010000</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>001000</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>000100</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>000010</td>
</tr>
<tr>
<td>F</td>
<td>2</td>
<td>000001</td>
</tr>
<tr>
<td>Bcast</td>
<td>1</td>
<td>101010</td>
</tr>
<tr>
<td>Bcast</td>
<td>2</td>
<td>010101</td>
</tr>
<tr>
<td>MulticastC</td>
<td>1</td>
<td>101010</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>1</td>
<td>100010</td>
</tr>
<tr>
<td>Unk Mcast</td>
<td>2</td>
<td>010101</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>1</td>
<td>000000</td>
</tr>
<tr>
<td>Unk Ucast</td>
<td>2</td>
<td>000000</td>
</tr>
</tbody>
</table>
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

Physical End Station

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

Physical End Station

PVID set at VSI

SVID added at vUplink

SVID removed
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
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

<table>
<thead>
<tr>
<th>CPU Utilization (top)</th>
<th>Throughput (Mbps)</th>
<th>RTT Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>External</td>
<td>Internal</td>
</tr>
<tr>
<td>VEB</td>
<td>VEPA</td>
<td>VEB + FW</td>
</tr>
<tr>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>10.00</td>
<td>2.00</td>
<td>1.00</td>
</tr>
<tr>
<td>20.00</td>
<td>3.00</td>
<td>2.00</td>
</tr>
<tr>
<td>30.00</td>
<td>4.00</td>
<td>3.00</td>
</tr>
<tr>
<td>40.00</td>
<td>5.00</td>
<td>4.00</td>
</tr>
<tr>
<td>50.00</td>
<td>6.00</td>
<td>5.00</td>
</tr>
<tr>
<td>60.00</td>
<td>7.00</td>
<td>6.00</td>
</tr>
</tbody>
</table>

Comparing internal vs. external for VEB and VEPA configurations, the charts illustrate performance metrics such as CPU utilization, throughput (Mbps), and round trip time latency (ms). The graphs show significant differences in performance between the configurations, highlighting the impact of each setup on network efficiency and latency.
A Measure of Efficiency
VM Appliance Comparison Topologies

- **CPU Utilization (top)**
- **Throughput (Mbps)**
- **RTT Latency (ms)**

*Compare internal vs. external firewall configurations.*
A Measure of Efficiency

![Diagram showing Mbps per Percentage of CPU for different scenarios: VEPA, VEB + VM FW, and VEB + XFW. The graph compares internal and external conditions.]
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