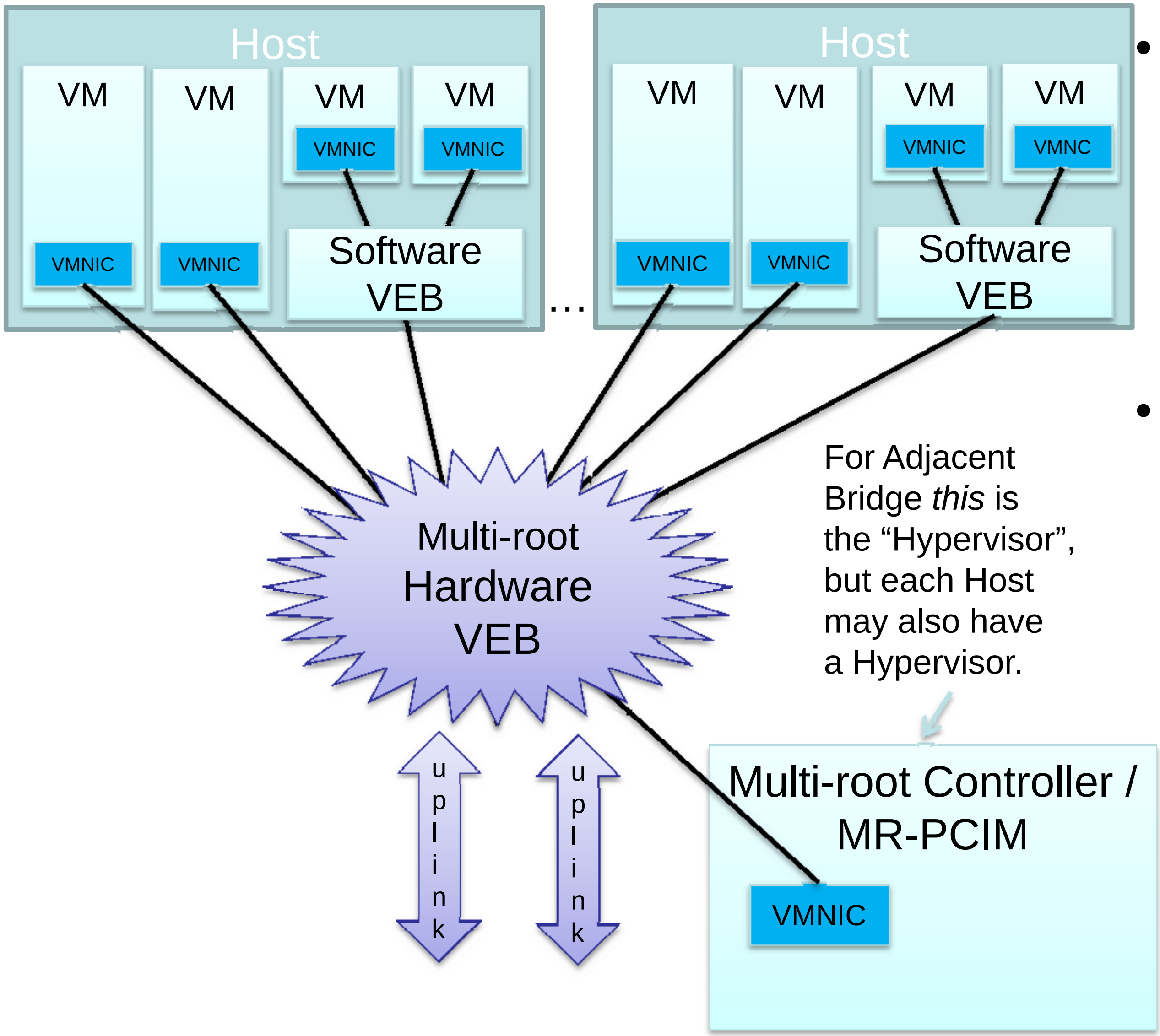
# Multi-Root VEBs

Caitlin Bestler

cait@asomi.com

# Multi-Root VEB



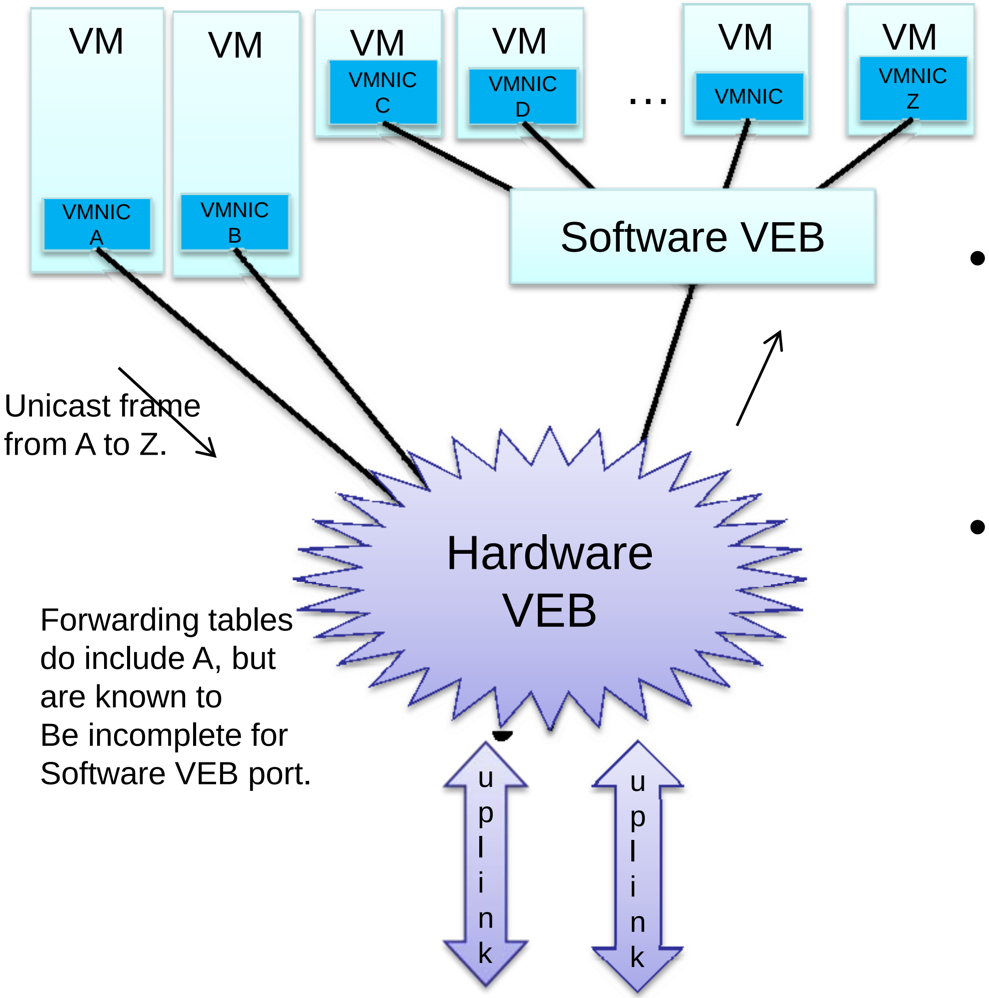
For Adjacent Bridge *this* is the “Hypervisor”, but each Host may also have a Hypervisor.

- As with a typical Hardware VEB, one or more uplinks are Bridged with local ports that are PCIe Functions.
- However there is only one multi-root master function, and one local master per host.

# Multi-Root Issues

- Terminology: "Hypervisor" is not the entity that admits/controls the VSIs in a Multi-Root VEB.
  - A term that emphasizes the role (assigning VSIs) rather than the usual occupant would be better.
  - "Virtualization Intermediary" works, but is vague.
- VEB Stacking
  - A Multi-Root Hardware VEB has Single-Root VEBs in the role of End Stations.
  - The definition of a "VEB" should be compatible with this.
- Hairpin Reflection
  - Without Hairpin reflection, the multi-root Hardware VEB must know *all* VMNICs supported by *all* Software VEBs.
  - Avoid forcing an "all or nothing" choice on Hairpin Reflection.

# Hairpin Reflection also an Issue for Single-Root VEBs

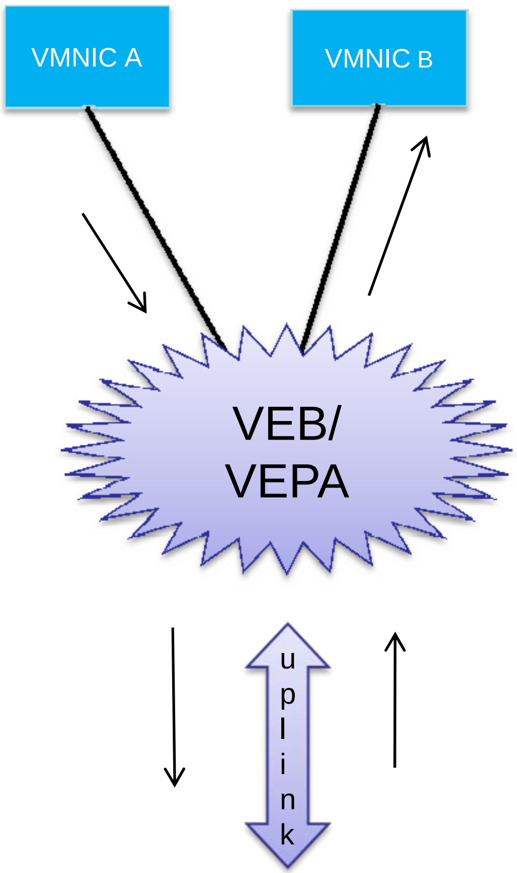


Unicast frame from A to Z.

Forwarding tables do include A, but are known to Be incomplete for Software VEB port.

- Anytime the Hardware VEB’s forwarding tables cannot hold all of the Software VEBs VMNICs Hairpin Reflection is useful.
- Multi-root environment just makes this more likely because more Software VEBs typically means more VMNICs.
- Without Hairpin reflection Where does Hardware VEB send frame from A to Z (not in its tables)?
  - Uplink Only?
    - Probably correct, but not always.
  - Uplink and Software VEB?
    - Software VEB may not have same capacity as the Uplink.
    - PAUSE from Software VEB can delay outbound traffic.

# Unnecessary Reflection Costs More than 2X.



Direct: Frame from A to B via VEB

1. Transmit from A to VEB
2. Wait in VEB Output Queue for B
3. Transmit from VEB to B.

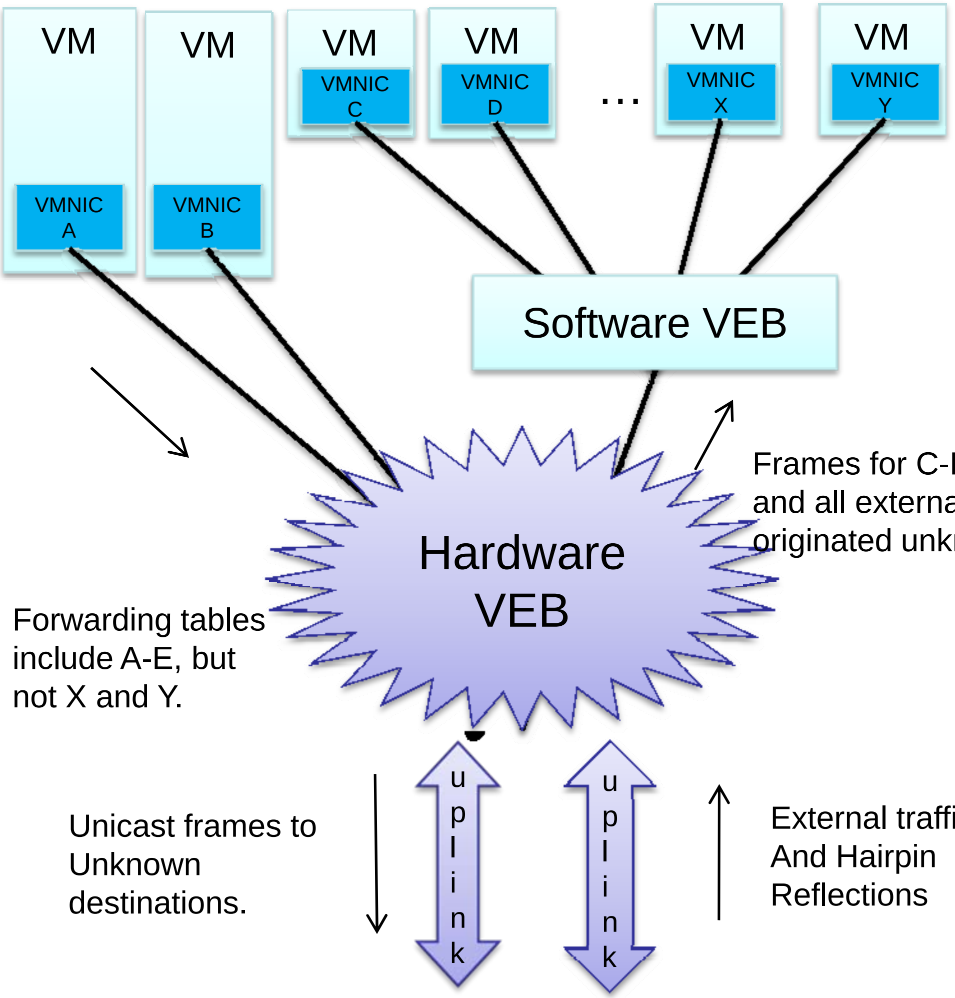
- Two Hops
- One Queue Wait

Reflection: Frame from A to B via VEPA

1. Transmit from A to VEPA
2. Wait in VEPA Output Queue to Adjacent Bridge.
3. Transmit from VEPA to Adjacent Bridge
4. Wait in Adjacent Bridge Output Queue for VEPA.
5. Transmit from Adjacent Bridge to VEPA.
6. Wait in VEPA Output Queue for B.
7. Transmit from VEPA to B.

- Four Hops
- Three Queue Waits
- Even worse if any of the extra queues delays trigger any form of Pause, CNM or drop.

# Avoiding All or Nothing Trap



Frames for C-E and all externally originated unknown

Forwarding tables include A-E, but not X and Y.

Unicast frames to Unknown destinations.

External traffic And Hairpin Reflections

- Because of limitations in Hardware VEB’s forwarding table size there are VMNICs known to the Software VEB but not to the Hardware VEB.
- Traffic from A to B,C,D and E can still be directly forwarded.
  – **Major** performance improvement.
- Traffic from between VMNICs on the Software VEB can always be direct.
  – And any to Hardware VEB supported VMNIC (A,B) can be direct through Hardware VEB.
- Some frames from each VMNIC can be directly forwarded, even if all cannot be.

# Established Connection Direct Forwarding

- Another example of partial use of Hairpin Reflection – Frames establishing TCP connections (SYN/SYN-ACK) are forwarded to the Adjacent Bridge.
    - This enables the Adjacent Bridge to let a Firewall (internal or external to it) approve and track the connection.
- Only after the connection is established are the frames directly forwarded.
- This is still falls under one simple rule:
    - “the VEB forwards some subset of the frames internally”.
- The benefits of the external firewall can be gained *without* requiring the entire connection flow through it.
    - An “All or Nothing” rule would block this functionality.

# Distributed vs. Central Execution

- There is no benefit in mandating central execution of forwarding.
  - Forwarding table capacity and/or ACLs that will be commonly used can be implemented more efficiently in a distributed fashion.
- But there are benefits for shared/central capacity:
  - Distributed resources are statically allocated to their physical location.
  - Central implementation has larger base to justify more complex, less frequently used, logic.
- Allowing VEB and/or VEPA enables benefits of distributed and centralized forwarding.

# No Impact on Adjacent Bridge

- When a VEB requests enabling Hairpin Reflection, simply honor the request.
  - The VEB is responsible for properly handling all reflected frames.
    - It does not reflect them itself.
    - It does not deliver them back to the source VMNIC.
      - But an Outer VEB might re-deliver to an Inner VEB
        - Assuming the Inner VEB requested Hairpin Reflection.
  - VEB must control / track all frames it directly delivers.
    - But there is no simple characterization of what frames will be direct delivered.
    - In example, direct delivery to X,Y was supported for C thru Y,
      but not for frames originating from A and B.
    - Simply assume that the VEB will directly deliver when convenient.
      - Query its statistics to find out how often it is actually doing so.

# Proposal

- A “VEB”, as opposed to a “VEPA” is *allowed* to do internal port to port forwarding.
  - It MAY request Hairpin Reflection.
  - If it does so it MUST prevent reflected frames from being re-delivered to their source.
- There should be a formal definition of a VEB as a subset of an 802.1Q Bridge, much as for a Two Port MAC Relay.