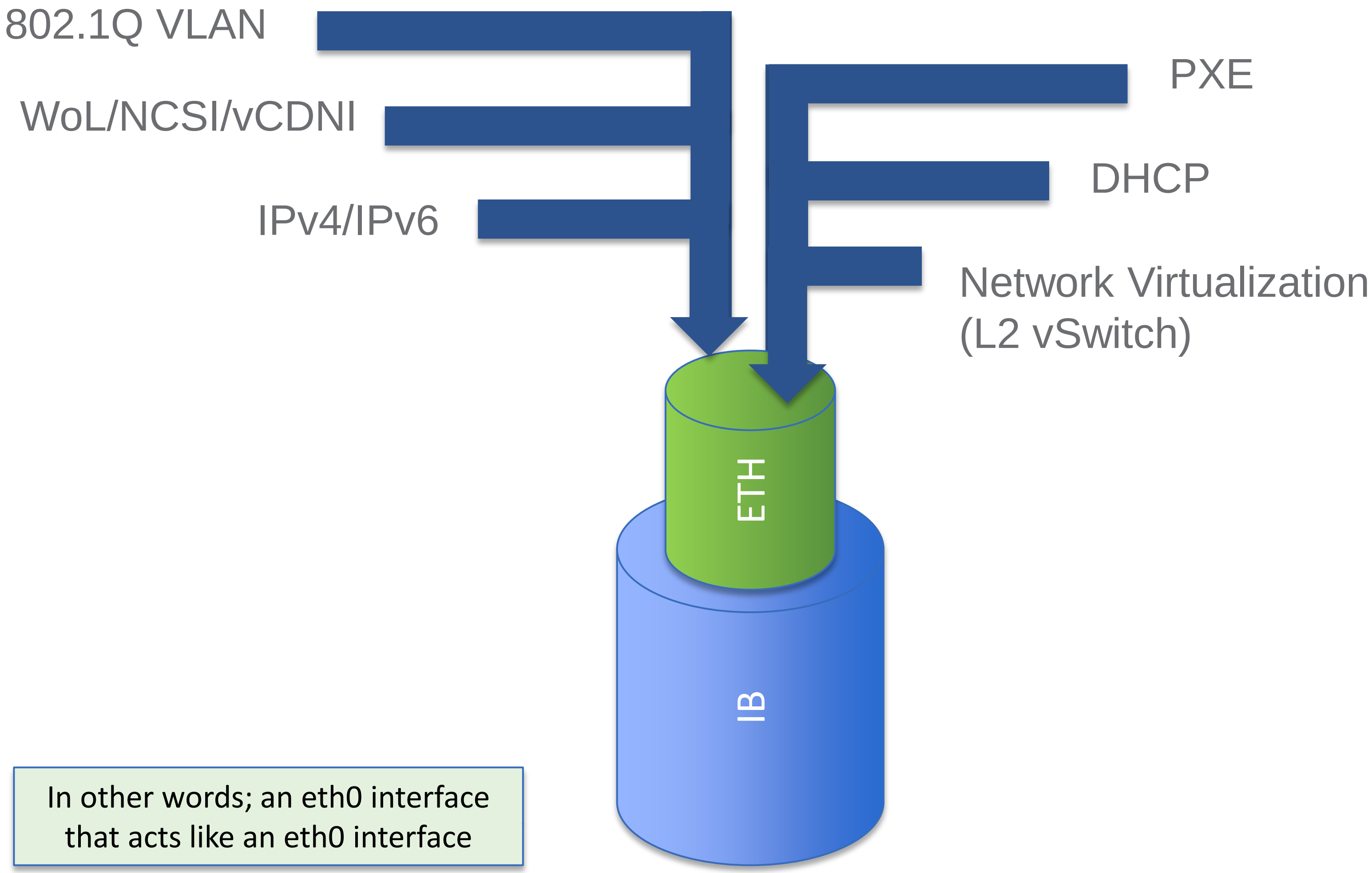
OPENFABRICS ALLIANCE

Ethernet over InfiniBand

Ali Ayoub, Mellanox Technologies

April 2013

Goal



In other words; an eth0 interface that acts like an eth0 interface

Goal

- Seamless Support for Ethernet Services over InfiniBand Network
  - IP and non-IP Applications
  - Virtualization (vSwitch)
  - 802.1Q
- Seamless Ethernet Management
  - DHCP, PXE, etc.
  - Load Balancing & High Availability
    - Unmodified Bonding/Teaming driver support
- Protocol may be distributed
  - Doesn’t rely on central software/hardware manager
- Simple bridging between EoIB and Ethernet

What’s New

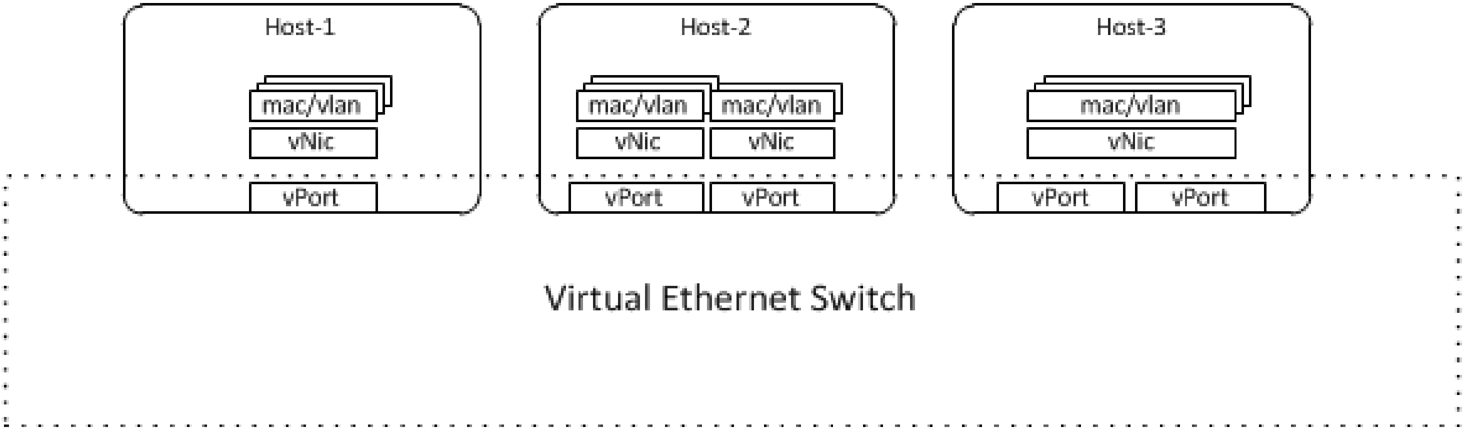
	Ethernet	EoIB	IPoIB
Ethernet Header	➢Present	➢Present	➢Not Present
Compatibility with L2-based apps	➢Seamless	➢Seamless	➢Not Supported ➢Needs special handling when using eIPoIB
MAC Setting	➢Any	➢Any	➢Limited: based on QPN and GID
MAC Length	➢6 bytes	➢6 bytes	➢20 bytes
Migration	➢Transparent to the netdev driver	➢Transparent to the netdev driver	➢Requires special handling
MTU	➢9K	➢Limited by IB mtu: 4K (in UD)	➢Limited by IB mtu: 4K (in UD)
VLAN ID	➢Any	➢Any	➢IPoIB: Not Supported ➢eIPoIB: Mapped to PKEY (1..128 only, cannot exceed PKEY range)

Model

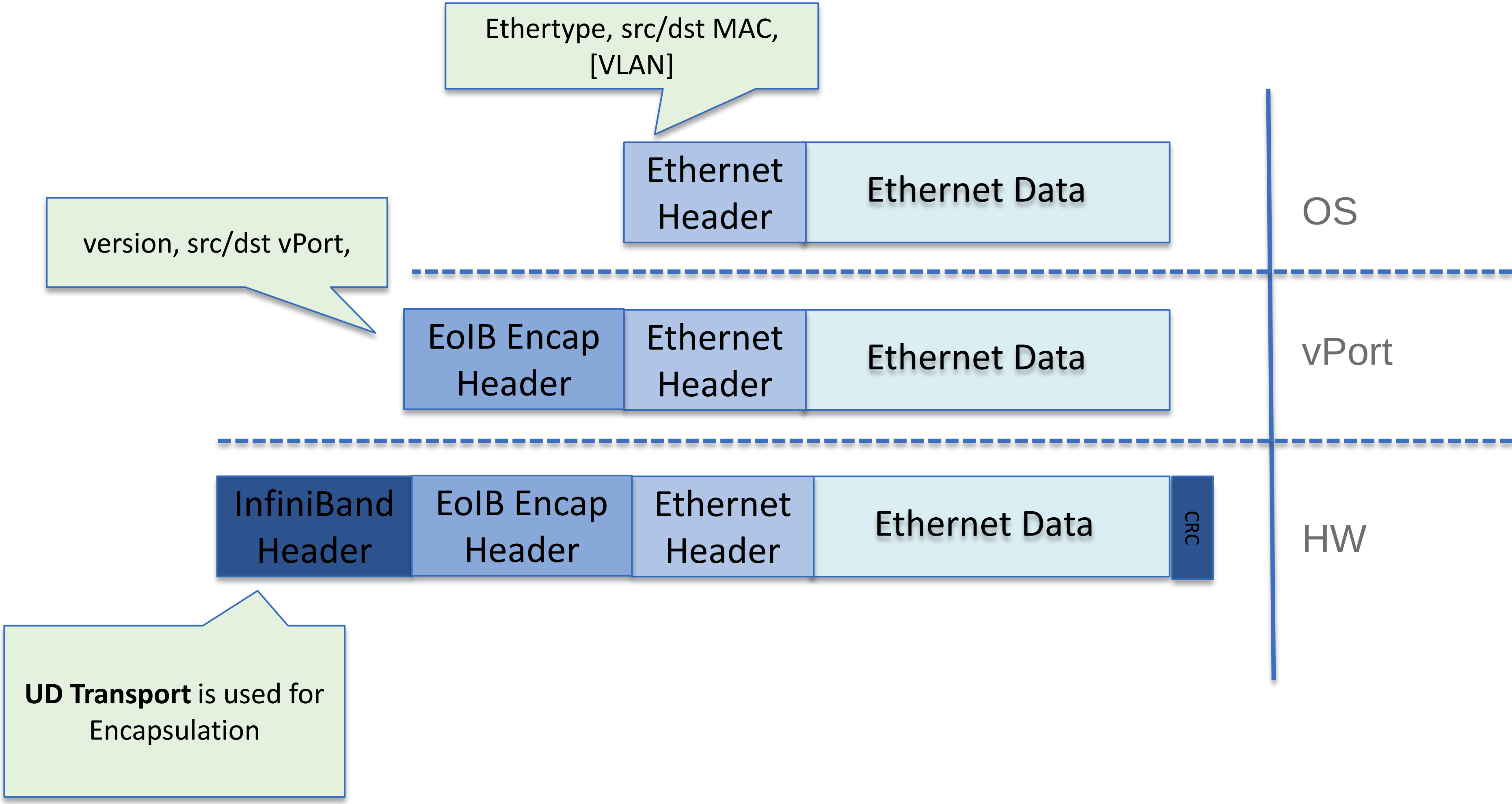
- **Ethernet Overlay Network on top of InfiniBand Underlying Network (UD Transport)**
- InfiniBand Network as a “giant” Virtual Ethernet Switch (VES)
- End points may have one or more Virtual Ports (vPort) connected to the VES
- A Virtual NIC (vNIC) represents the Ethernet Interface within the end-point, connected directly to the vPort
- A Gateway (GW) can be implemented the same way as a host with multiple pNIC/vNIC instances

Model

- VES is distributed; each vPort holds a Forwarding Database (FDB) table.
- Optionally, a VES manager can be used to push the FDB table to the end points
- A Gateway (GW) can be implemented the same as a host with multiple pNIC/vNIC instances



Packet Format



UD Transport is used for Encapsulation

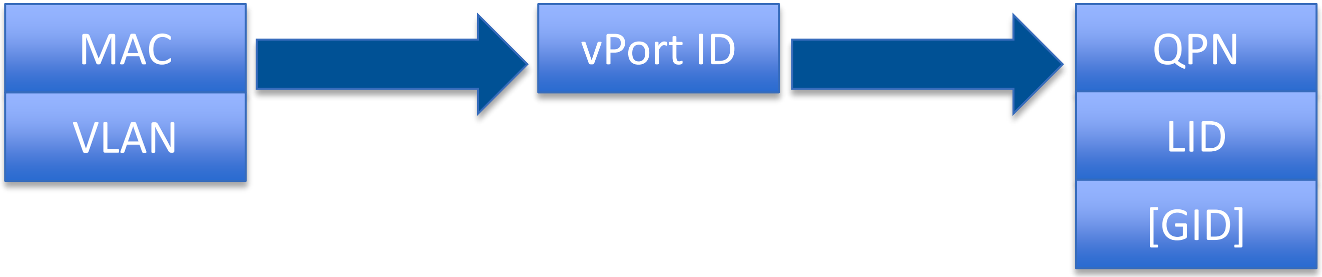
Address Resolution

- What’s New:
  - Ethernet Link Layer (MAC) is decoupled from the underlying InfiniBand network
    - Allows using any MAC address; a must for virtualization models where the hypervisor is responsible for VM’s MAC setting
  - EoIB is decoupled from ARP/NDP protocols
    - No dependency on the OS address resolution and Control Plane
    - Allows EoIB to have its own Control Plane and carry information/notifications not available in ARP/NDP
  - Learning

Address Resolution

- How it works:
  - Each end-point holds a Forwarding Database (FDB) table
  - The FDB is used to map the Ethernet packet based on MAC/VLAN to the corresponding InfiniBand Address Handle
  - FDB is updated based on ingress traffic learning as well as EoIB Control Plane
  - If mapping is missing, the packet is flooded (distributed mode)
    - Similar to VXLAN approach

- Egress Packet Flow:



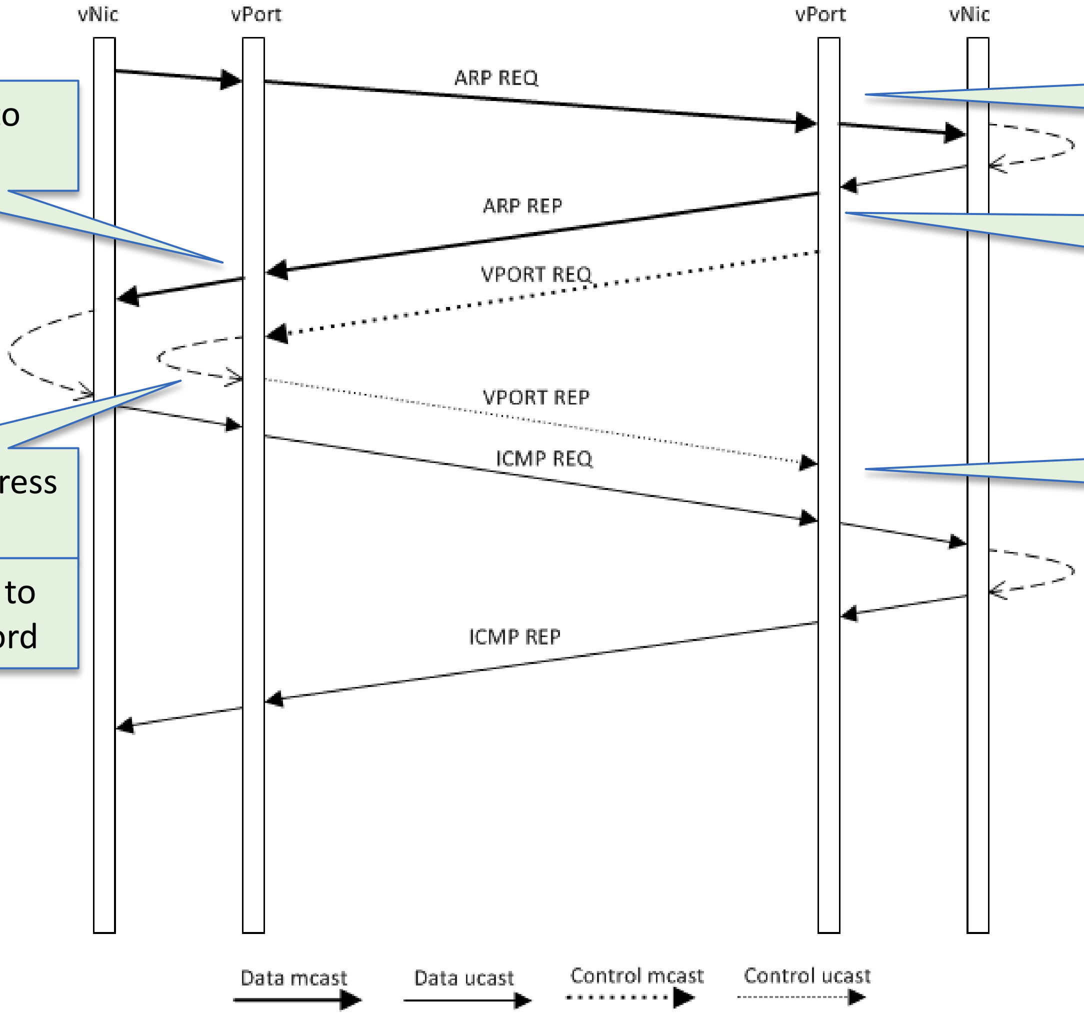
FDB

- Construction:
  - Learn incoming traffic to map MAC/VLAN to a vPort
    - Same approach as physical Switch learning
  - Use EoIB Control Plane (vPort Request/Reply) to map vPort to IB Address
  - SA query is sent out to get the PathRecord based on the IB Address
- Scheme

Overlay Address		Underlying Address	Physical Address		
MAC	VLAN	vPort ID	QPN	LID	[GID]
			QPN	LID	[GID]
MAC	VLAN	vPort ID	QPN	LID	[GID]
MAC	VLAN	vPort ID	QPN	LID	[GID]

For Link Aggregation

Ping Example



Learn MAC/VLAN to vPort mapping

Learn vPort to IB address mapping

SA Query is issued* to obtain the PathRecord

Learn MAC/VLAN to vPort mapping

FDB is incomplete; flooding is used (in distributed mode)

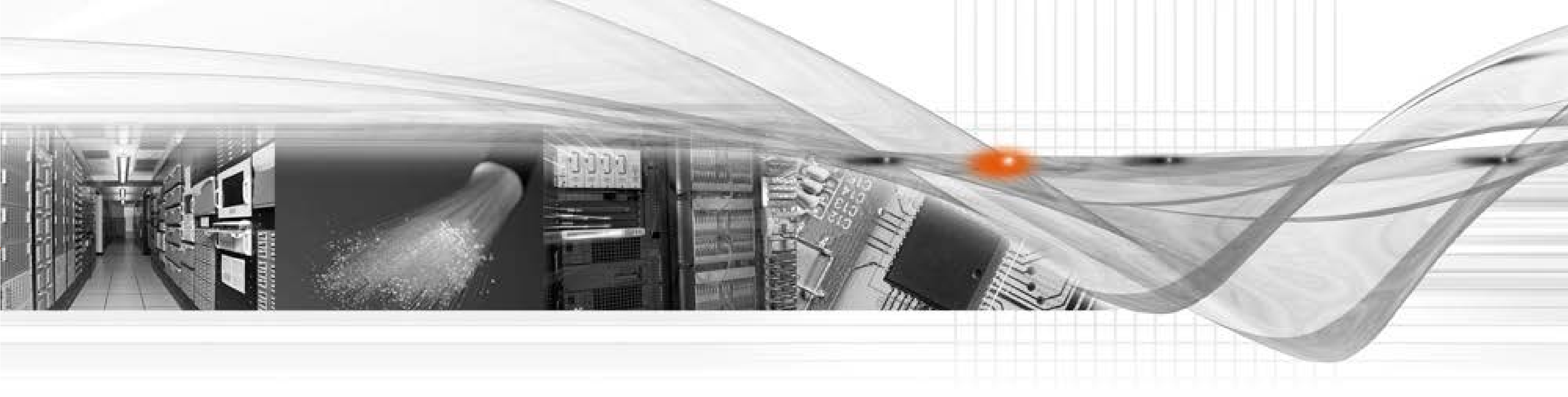
Learn vPort to IB address mapping

SA Query is issued* to obtain the PathRecord

* Not shown in the diagram

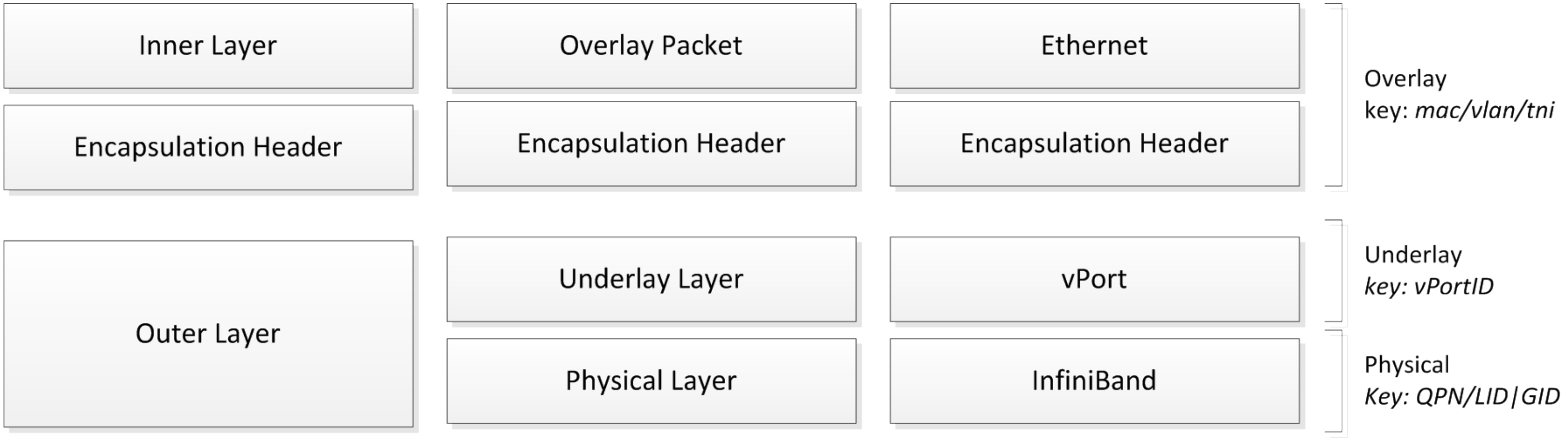
Thank You

OPENFABRICS ALLIANCE



Backup

Layers



Inner Layer

Encapsulation Header

Outer Layer

Overlay Packet

Encapsulation Header

Underlay Layer

Physical Layer

Ethernet

Encapsulation Header

vPort

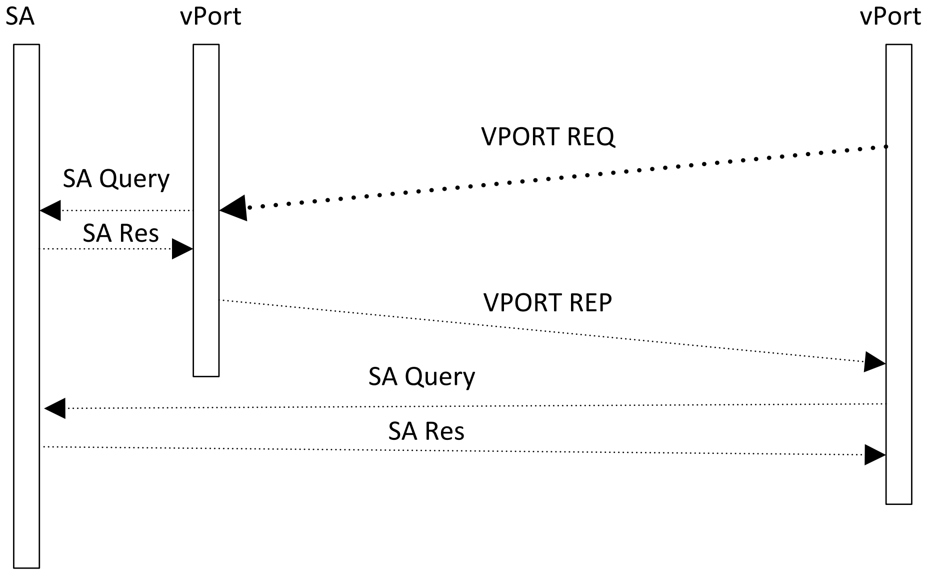
InfiniBand

Overlay
key: *mac/vlan/tni*

Underlay
key: vPortID

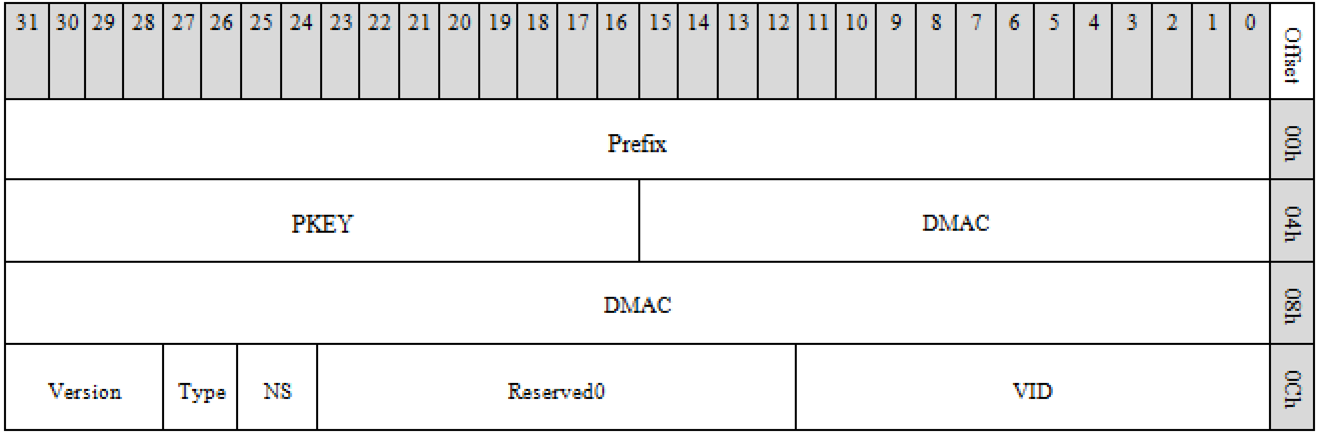
Physical
Key: QPN/LID|GID

SA Query


SA
vPort
vPort
VPORT REQ
SA Query
SA Res
VPORT REP
SA Query
SA Res

Multicast

Table 19: Multicast GID Layout



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Offset
Prefix																																00h
PKEY																DMAC																04h
DMAC																																08h
Version				Type		NS		Reserved0												VID												0Ch

VES Instances

- Each PKEY defines a VES instance
- VES can serve multiple VLANs
  - VLAN and PKEY are decoupled
  - The administrator can limit the use of specific VLAN group for each VES instance for higher security