

# InfiniBand Architecture Overview



CONFIDENTIAL

- In the end of this section you will be able to
  - List the major InfiniBand components
  - List the 5 main layers of InfiniBand architecture
  - Understand each layer responsibilities
  - Identify the main mechanisms/features of each layer
  - Understand InfiniBand management model
  - Understand the role and operation of Subnet Manager
  - Get familiar with common cluster topologies

# InfiniBand Technical Overview

## ■ What is InfiniBand?

- InfiniBand is an open standard, interconnect protocol developed by the InfiniBand® Trade Association:  
<http://www.infinibandta.org/home>
- First InfiniBand specification was released in 2000

## ■ What does the specification includes?

- The specification is very comprehensive
- From physical to applications

## ■ InfiniBand SW is developed under OpenFabrics Open source Alliance

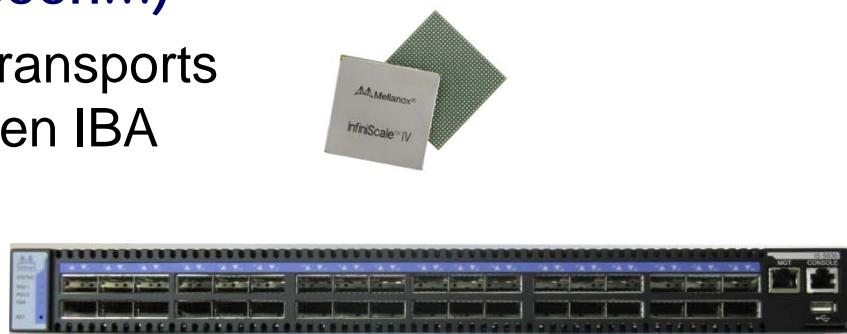
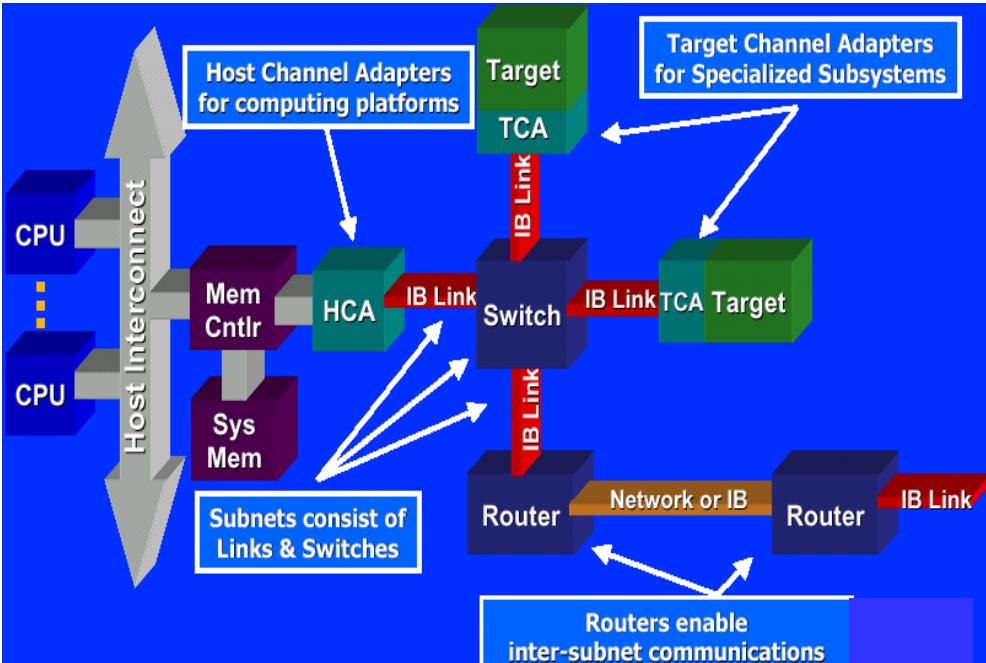
- <http://www.openfabrics.org/index.html>

# Infiniband Feature Highlights

- **Serial High Bandwidth Links**
  - 10Gb/s to 40Gb/s HCA links
  - Up to 120Gb/s switch-switch
- **Ultra low latency**
  - Under 1 us
- **Reliable, lossless, self-managing fabric**
  - Link level flow control
  - Congestion control
- **Full CPU Offload**
  - Hardware Based Transport Protocol
  - Reliable Transport
  - Kernel Bypass
- **Memory exposed to remote node**
  - RDMA-read and RDMA-write
- **Quality Of Service**
  - I/O channels at the adapter level
  - Virtual Lanes at the link level
- **Scalability/flexibility**
  - Up to 48K nodes in subnet, up to  $2^{128}$  in network

# InfiniBand Components

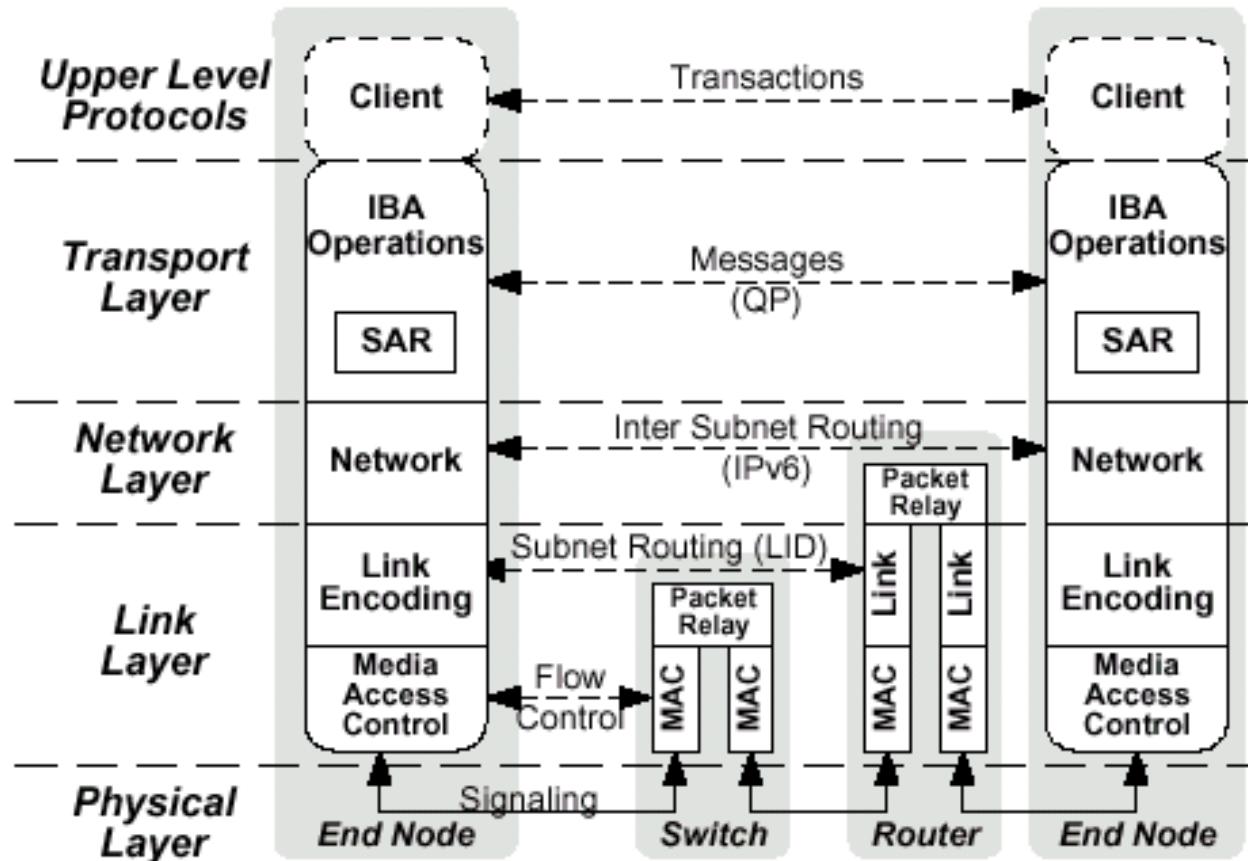
- **Host Channel Adapter (HCA)**
  - Device that terminates an IB link and executes transport-level functions and support the verbs interface
  
- **Switch**
  - A device that routes packets from one link to another of the same IB Subnet
  
- **Router (coming soon...)**
  - A device that transports packets between IBA subnets



# IB Architecture Layers

- Physical
  - Signal levels and Frequency; Media; Connectors
- Link
  - Symbols and framing; Flow control (credit-based); How packets are routed from Source to Destination
- Network:
  - How packets are routed between subnets
- Transport:
  - Delivers packets to the appropriate Queue Pair; Message Assembly/De-assembly, access rights, etc.
- Software Transport Verbs and Upper Layer Protocols
  - Interface between application programs and hardware.
  - Allows support of legacy protocols such as TCP/IP
  - Defines methodology for management functions

# InfiniBand Layered Architecture



# Physical Layer - Responsibilities

- The physical layer specifies how bits are placed on the wire to form symbols and defines the symbols used for framing (i.e., start of packet & end of packet), data symbols, and fill between packets (Idles). It specifies the signaling protocol as to what constitutes a validly formed packet
- InfiniBand is a lossless fabric. Maximum Bit Error Rate (BER) allowed by the IB spec is 10e-12. The physical layer should guarantee effective signaling to meet this BER requirement

# Physical Layer – Link Rate

- InfiniBand uses serial stream of bits to transfer data
- Link width
  - 1x – One differential pair per Tx and per Rx
  - 4x – Four differential pairs per Tx and per Rx
  - 12x - Twelve differential pairs per Tx and per Rx
- Link Speed
  - Single Dada Rate (SDR) – 2.5 GHz signaling (2.5Gb/s for 1x)
  - Doable Data Rate (DDR) – 5 GHz signaling (5Gb/s for 1x)
  - Quad Data rate (QDR) - 10 GHz signaling (10Gb/s for 1x)
- Link rate
  - Multiplication of the link width and link speed
  - Most common 4x QDR (40Gb/s)

# Physical Layer Cont'

## ■ Media types

- PCB: several inches
- Copper: 20m SDR, 10m DDR, 7m QDR
- Fiber: 300m SDR, 150m DDR, 100/300m QDR
- CAT6 Twisted Pair in future.

## ■ 8 to 10 bit encoding

## ■ Industry standard components

- Copper cables / Connectors
- Optical cables
- Backplane connectors



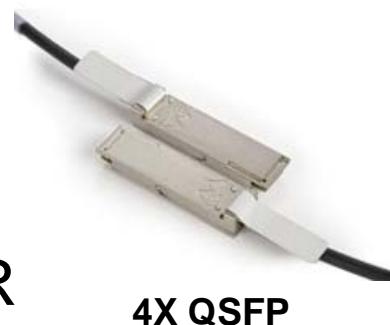
12X Cable



4x CX4 Fiber



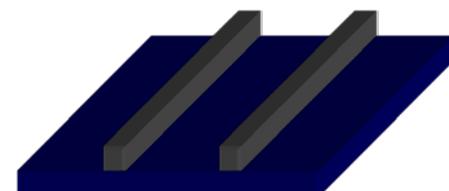
4X CX4



4X QSFP



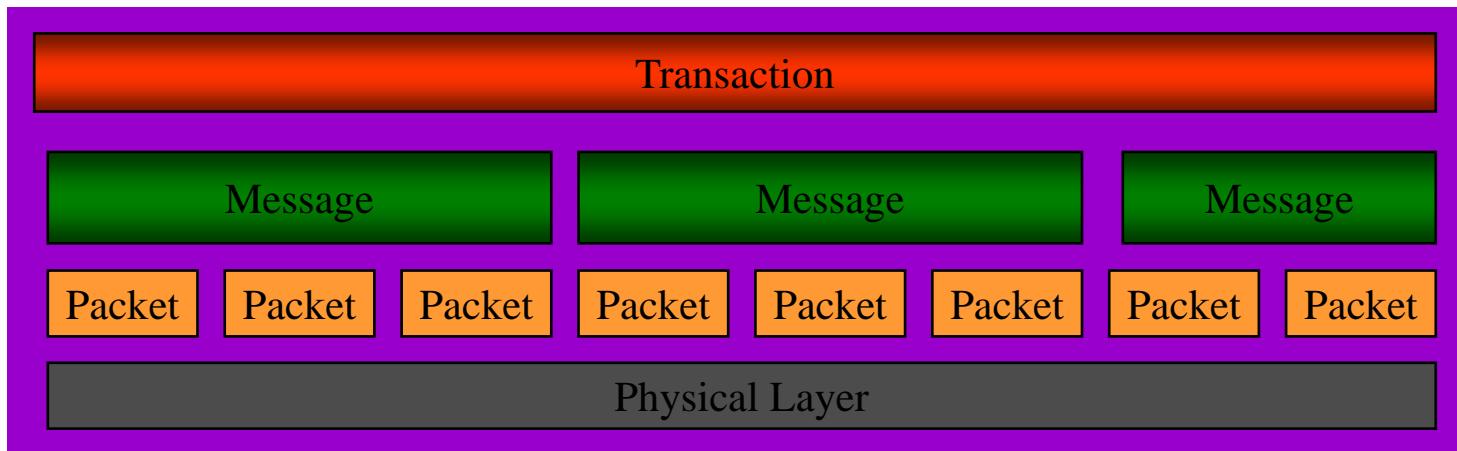
4x QSFP Fiber



FR4 PCB

# Link Layer - Responsibilities

- The link layer describes the packet format and protocols for packet operation, e.g. flow control and how packets are routed within a subnet between the source and destination



# Link Layer: Packets

- **Packets are routable end-to-end fabric unit of transfer**

- Link management packets: train and maintain link operation
- Data packets
  - Send
  - Read
  - Write
  - Acks

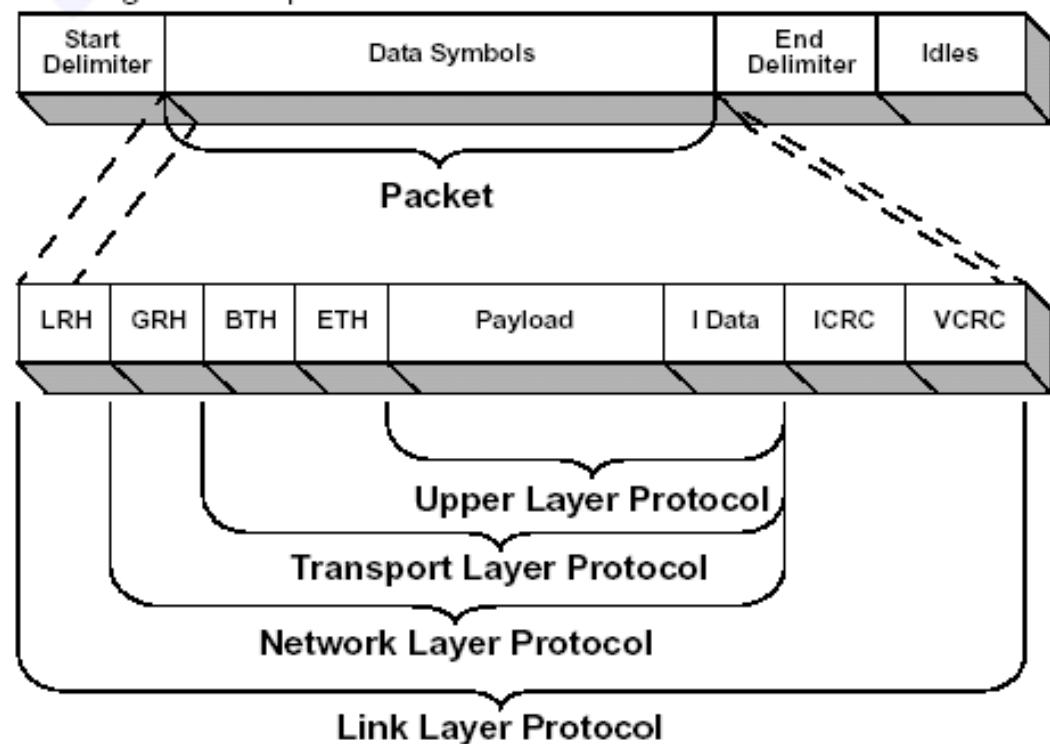


Figure 27 IBA Data Packet Format

# Link Layer: Payload Size

## ■ Maximum Transfer Unit (MTU)

- MTU allowed from 256 Bytes to 4K Bytes (Message sizes much larger).
- Only packets smaller than or equal to the MTU are transmitted
- Large MTU is more efficient (less overhead)
- Small MTU gives less jitter
- Small MTU preferable since segmentation/reassembly performed by hardware in the HCA.
- Routing between end nodes utilizes the smallest MTU of any link in the path (Path MTU)

# Link Layer: Virtual Lanes (Quality of Service)

## ■ 16 Service Levels (SLs)

- A field in the Local Routing Header (LRH) of an InfiniBand packet
- Defines the requested QoS

## ■ Virtual Lanes (VLs)

- A mechanism for creating multiple channels within a single physical link.
- Each VL:
  - Is associated with a set of Tx/Rx buffers in a port
  - Has separate flow-control
- A configurable Arbiter control the Tx priority of each VL
- Each SL is mapped to a VL
- IB Spec allows a total of 16 VLs (15 for Data & 1 for Management)
  - Minimum of 1 Data and 1 Management required on all links
  - Switch ports and HCAs may each support a different number of VLs
- VL 15 is a management VL and is not a subject for flow control

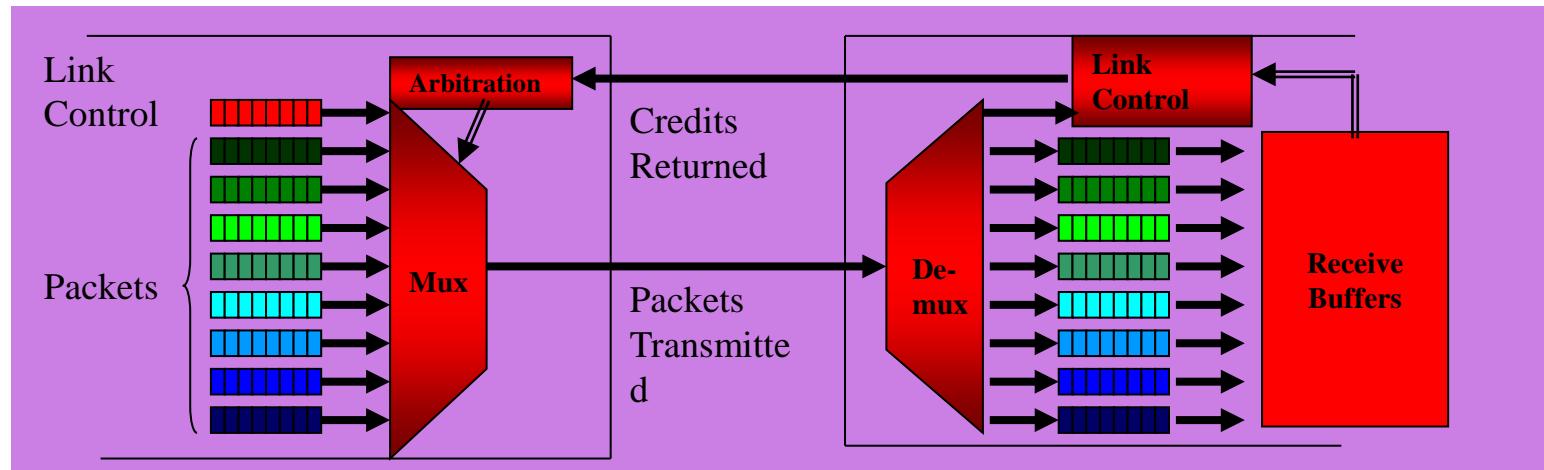
# Link Layer: Flow Control

## ■ Credit-based link-level flow control

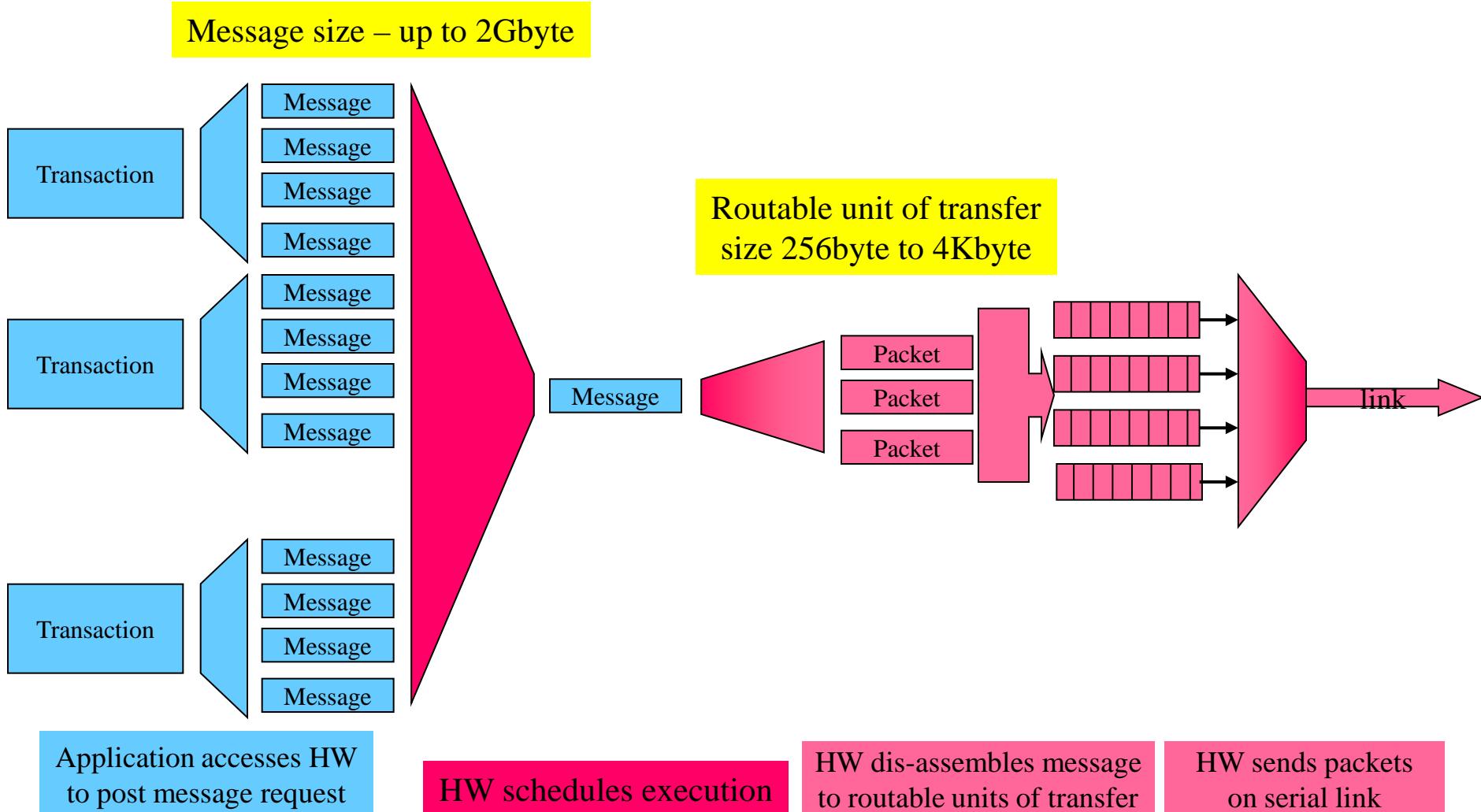
- Link Flow control assures NO packet loss within fabric even in the presence of congestion
- Link Receivers grant packet receive buffer space credits per Virtual Lane
- Flow control credits are issued in 64 byte units

## ■ Separate flow control per Virtual Lanes provides:

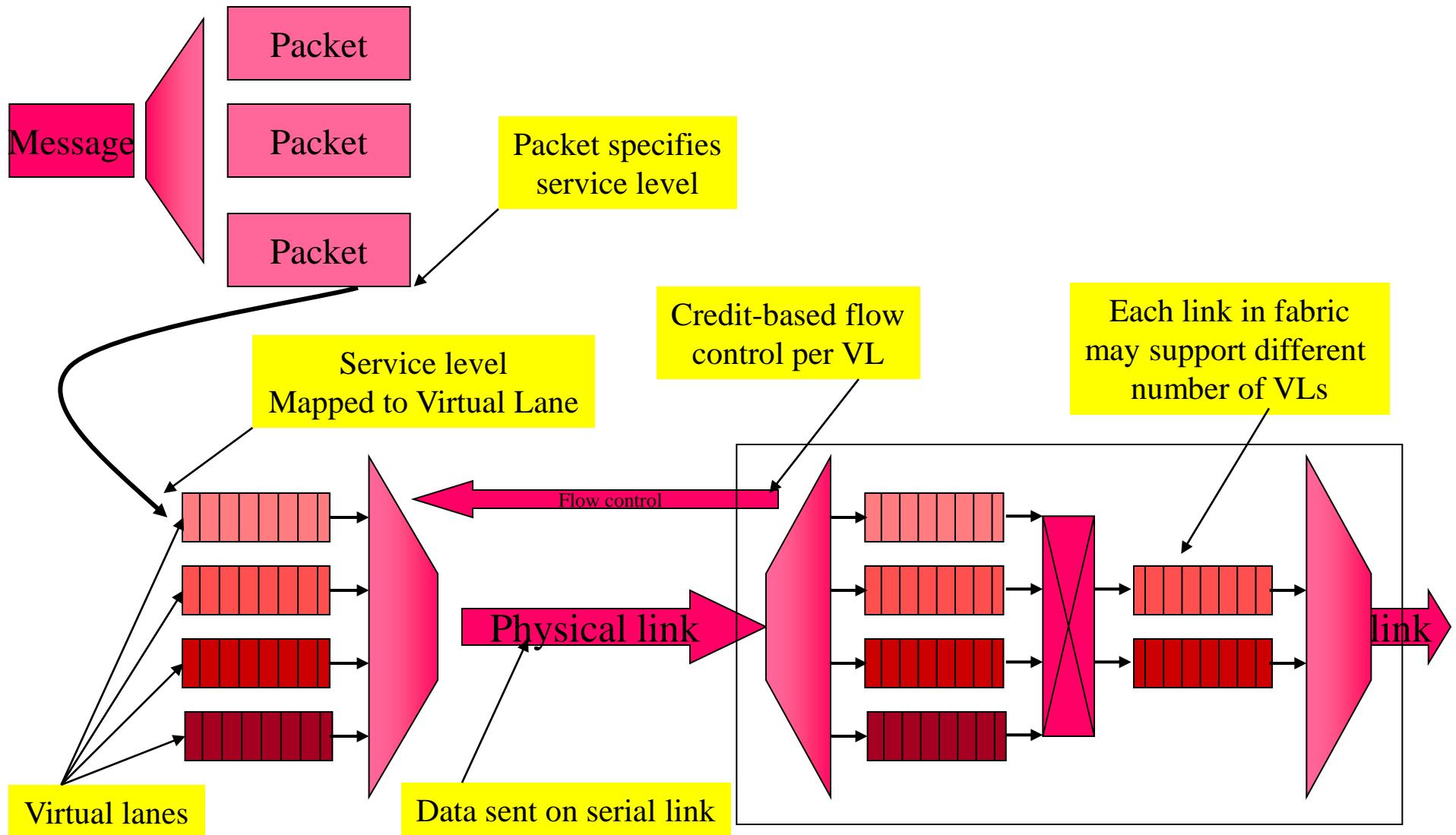
- Alleviation of head-of-line blocking
- Virtual Fabrics – Congestion and latency on one VL does not impact traffic with guaranteed QOS on another VL even though they share the same physical link



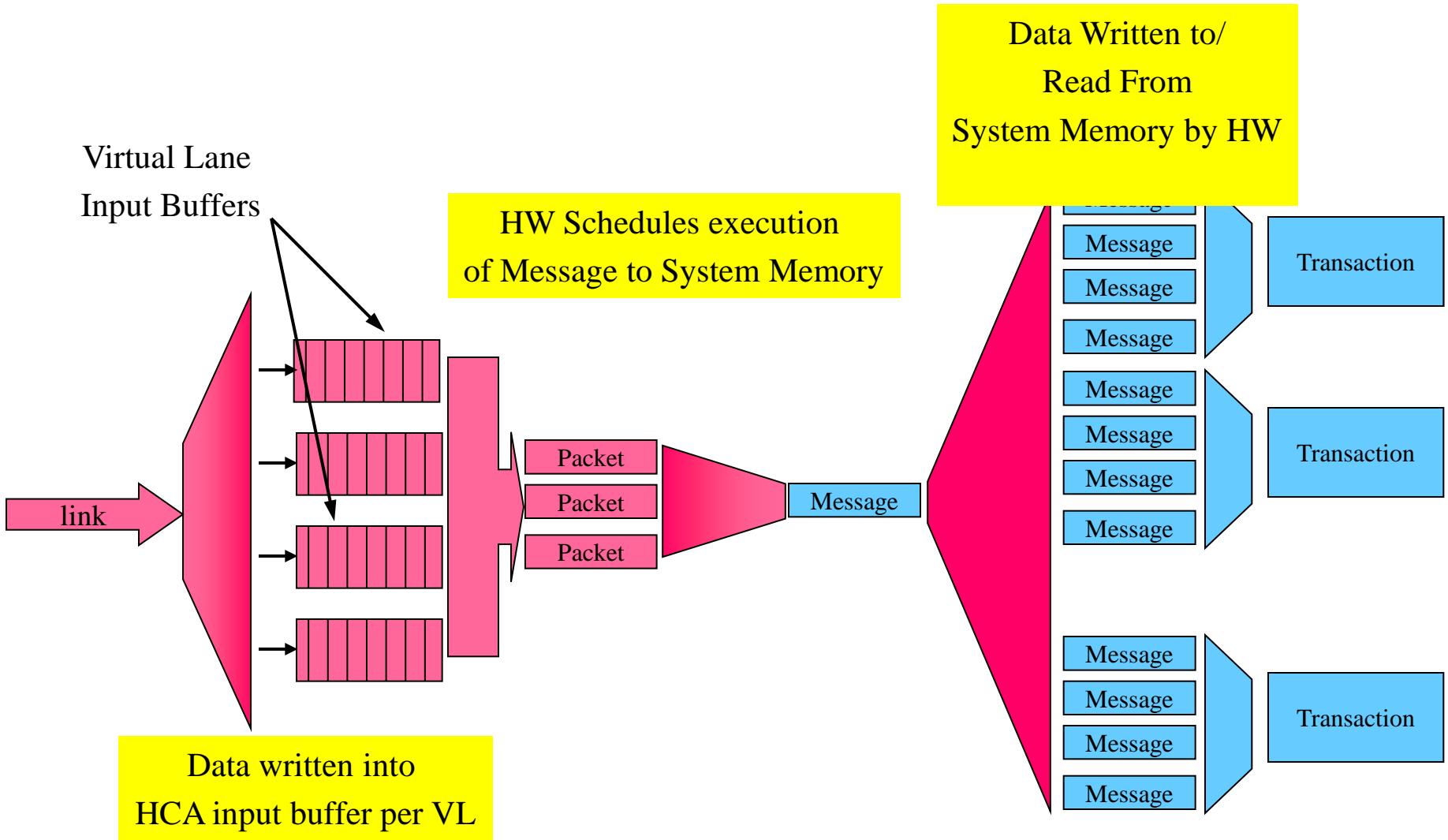
# Link Layer: Example



# Link Layer: Example



# Link Layer: Example



# Link Layer: Addressing

## ■ Local ID (LID)

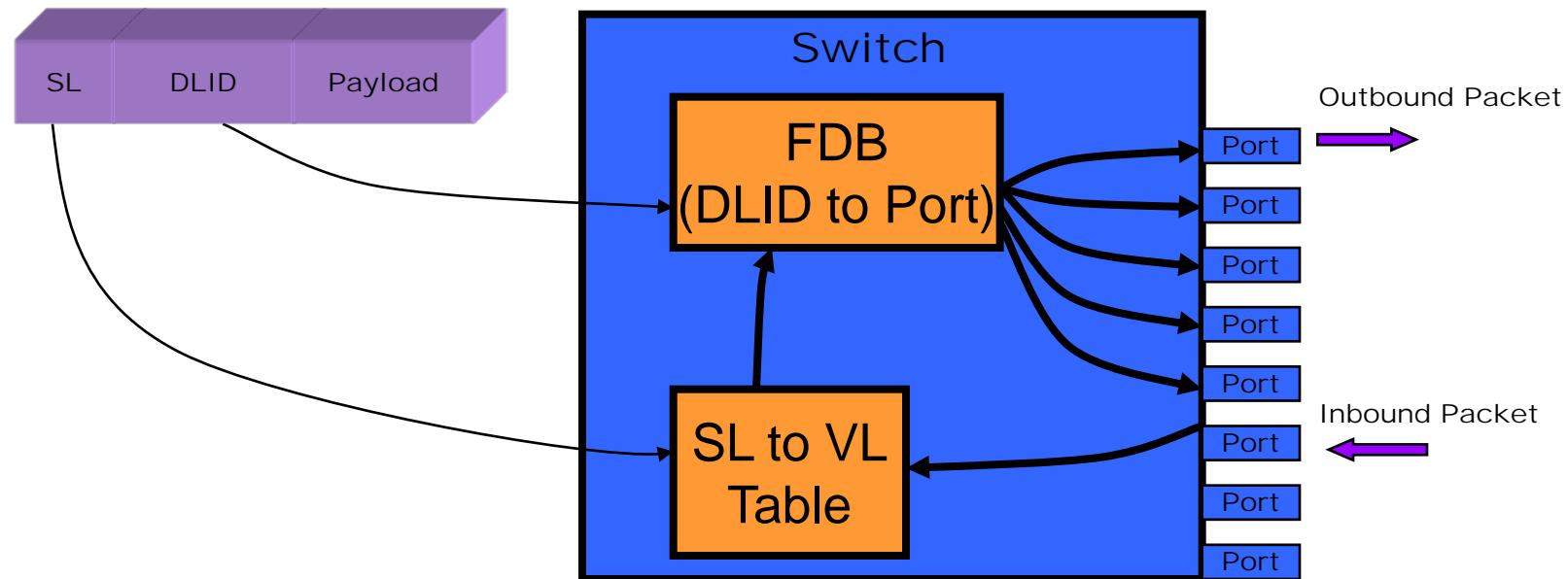
- 16 bit field in the Local Routing Header (LRH) of all IB packets
- Used to rout packet in an InfiniBand subnet
- Each subnet may contain up to:
  - 48K unicast addresses
  - 16K multicast addresses

## ■ Assigned by Subnet Manager at initialization and topology changes

# Layer 2 Forwarding

- Switches use FDB (Forwarding Database)
  - Based on DLID and SL a packet is sent to the correct output port.

Multicast Destinations supported!!



# Network Layer

## ■ Responsibility

- The network layer describes the protocol for routing a packet between subnets

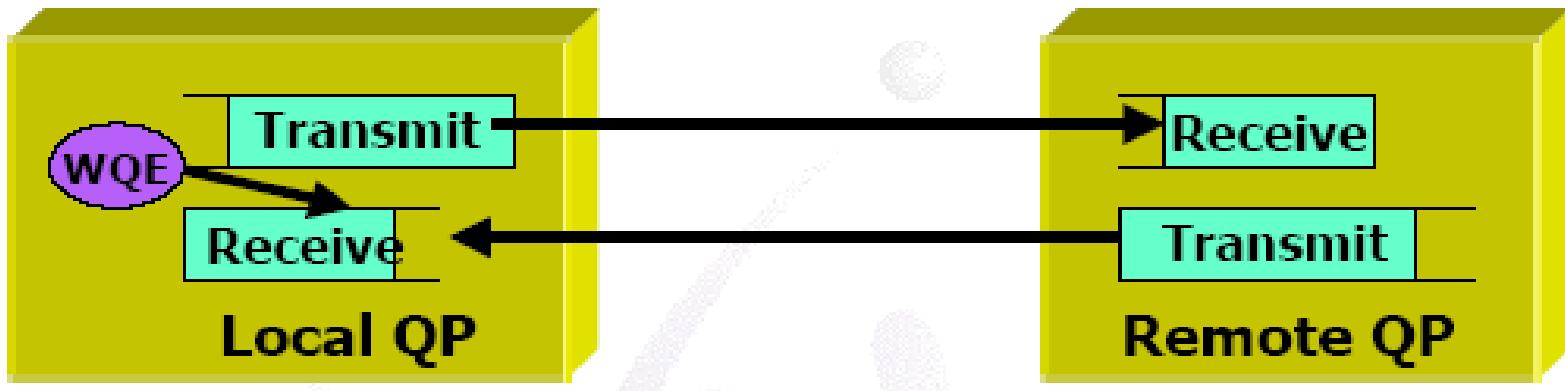
## ■ Globally Unique ID (GUID)

- A 64 bit field in the Global Routing Header (GRH) used to route packets between different IB subnets
- Every node must have a GUID
- IPv6 type header

# Transport Layer – Responsibilities

- The network and link protocols deliver a packet to the desired destination. The transport portion of the packet delivers the packet to the proper QP and instructs the QP how to process the packet's data.
- The transport layer is responsible for segmenting an operation into multiple packets when the message's data payload is greater than the *maximum transfer unit (MTU) of the path*. *The QP on the receiving end* reassembles the data into the specified data buffer in its memory

# Transport Layer: Queue Pairs



- QPs are in pairs (Send/Receive)
- Work Queue is the consumer/producer interface to the fabric
  - The Consumer/producer initiates a Work Queue Element (WQE)
  - The Channel Adapter executes the work request
  - The Channel Adapter notifies on completion or errors by writing a Completion Queue Element (CQE) to a Completion Queue (CQ)

# Transport Layer: Work Request Element

- **Data transfer**
  - Send work request
    - Local gather – remote write
    - Remote memory read
    - Atomic remote operation
  - Receive work request
    - Scatter received data to local buffer(s)
- **Memory management operations**
  - Bind memory window
    - Open part of local memory for remote access
  - Send & remote invalidate
    - Close remote window after operations' completion
- **Control operations**
  - Memory registration/mapping
  - Open/close connection (QP)

# Transport Layer: Types Transfer Operations

## ■ SEND

- Read message from HCA local system memory
- Transfers data to Responder HCA Receive Queue logic
- Does not specify where the data will be written in remote memory
- Immediate Data option available

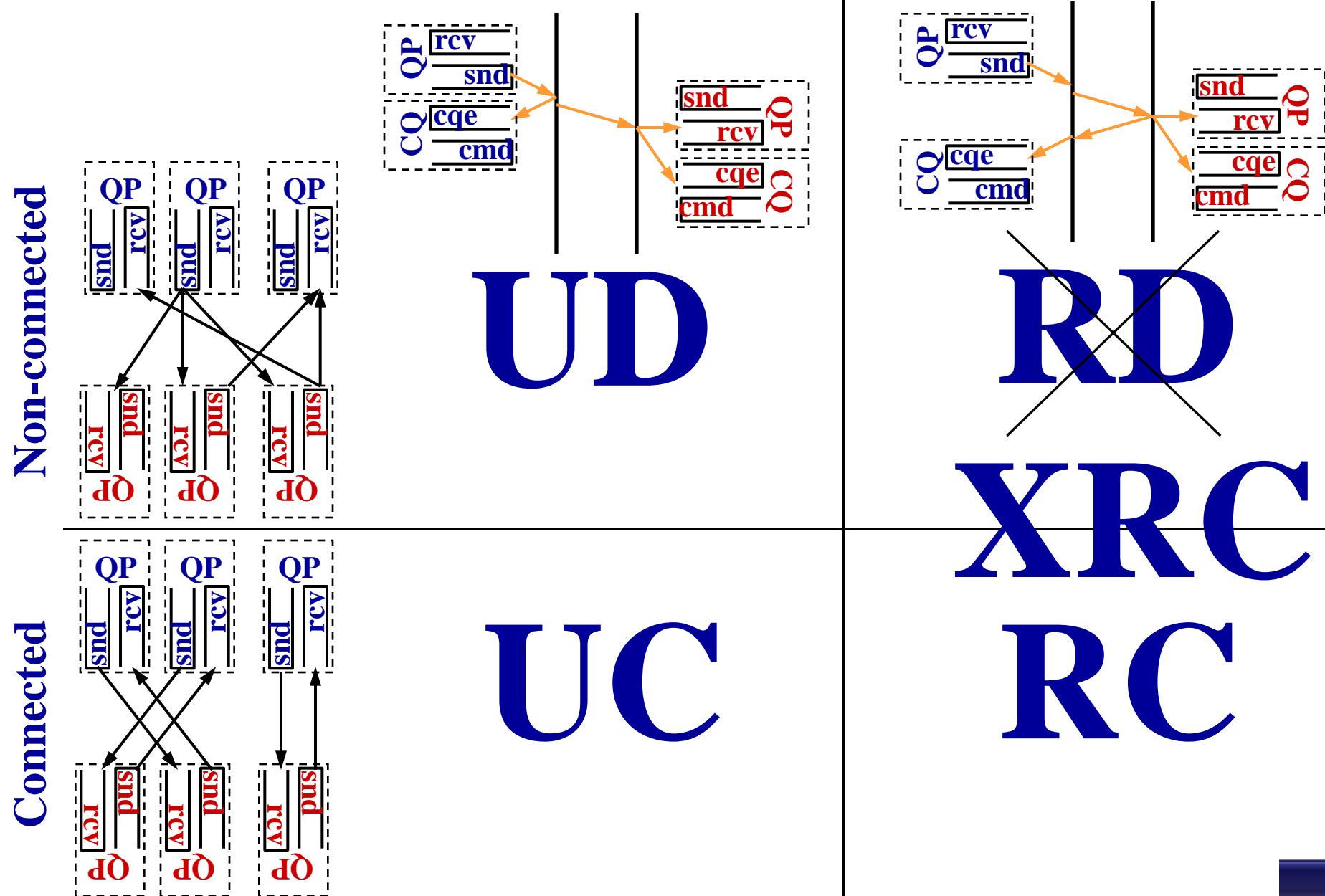
## ■ RDMA Read

- Responder HCA reads its local memory and returns it to the Requesting HCA
- Requires remote memory access rights, memory start address, and message length

## ■ RDMA Write

- Requester HCA sends data to be written into the Responder HCA's system memory
- Requires remote memory access rights, memory start address, and message length

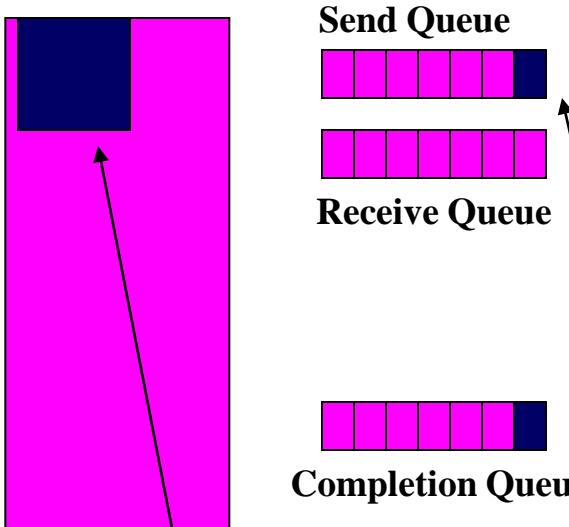
# Transport Services



# Transport Layer: Send operation example

HCA then consume the WQE,  
read the buffer and send to remote side  
send completion is generated

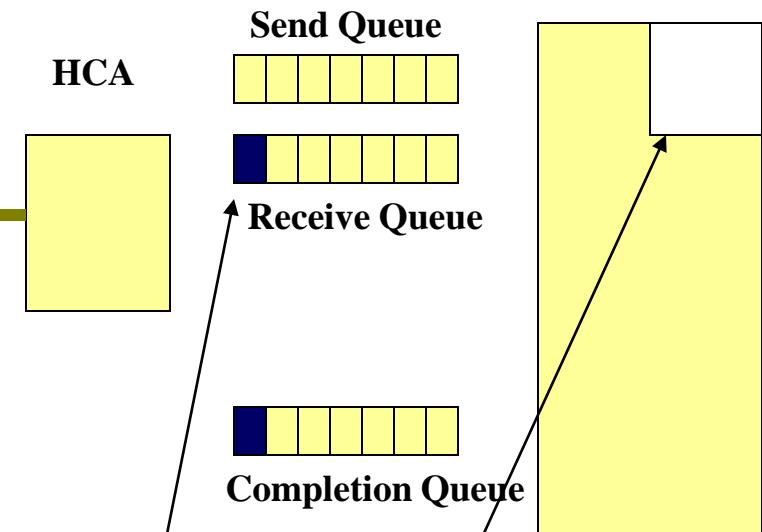
Host A RAM



The send side allocate a send buffer  
register it with the HCA, place a send WQE  
and ring a doorbell

When the packet arrives to the HCA  
It consumes a receive WQE, place  
the buffer in the appropriate location  
and generate a completion

Host B RAM

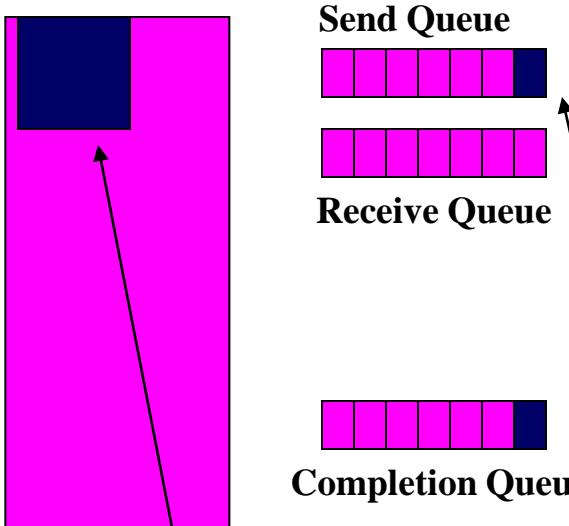


Application allocate receive buffer  
and place a receive WQE

# Transport Layer: RDMA Write Example

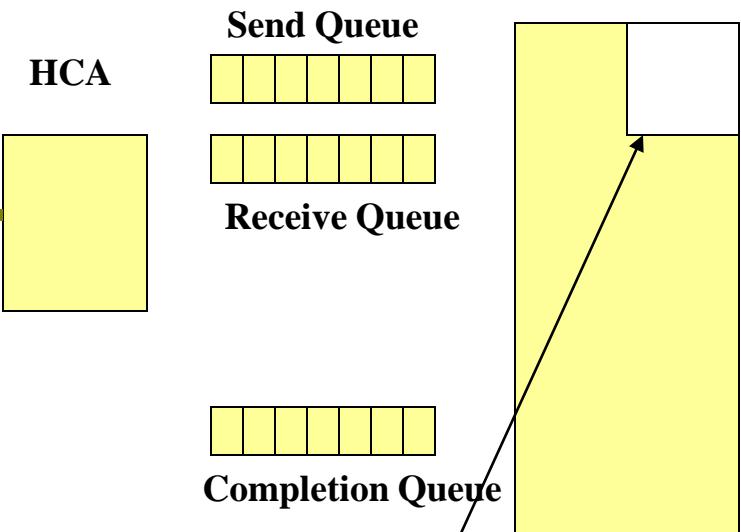
HCA then consume the WQE,  
read the buffer and send to remote side  
send completion is generated

Host A RAM



When the packet arrives to the HCA  
It checks the address and memory  
keys and write to memory directly

Host B RAM



The send side allocate a send buffer  
register it with the HCA, place a send WQE  
with the remote side's virtual address  
  
and ring a doorbell

Application allocate receive buffer  
and pass address and keys to  
remote side

# Transport Layer: Retransmissions

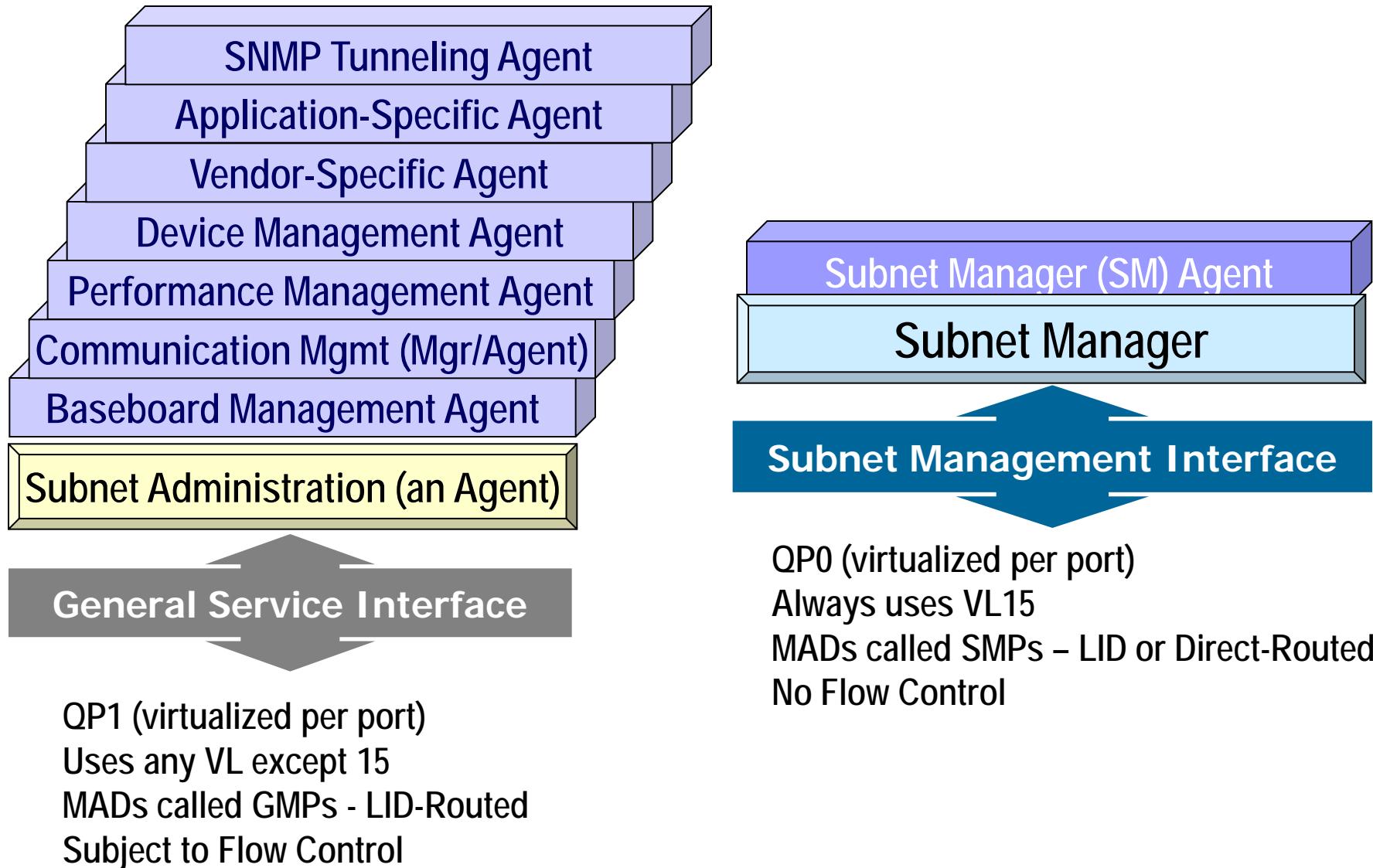
- For reliable transport services (RC, XRC) QPs maintain the flow of packets and retransmit in case a packet was dropped
- Each packet has a Packet Serial Number (PSN) that is used by the receiver identify lost packets
- The receiver will send ACKs if packets arrive in order and NACKs otherwise
- The send QP maintain a timer to catch cases where packets did not arrive to the receive QP or ACK was lost
- Retransmission is considered a “bad flow” which reduce performance or may break a connection

- Verbs are the SW interface to the HCA and the IB fabric
- Verbs are not API but rather allow flexibility in the API implementation while defining the framework
- Some verbs for example
  - Open/Query/Close HCA
  - Create Queue Pair
  - Query Completion Queue
  - Post send Request
  - Post Receive Request
- Upper Layer Protocols (ULPs) are application writing over the verbs interface that bridge between standard interfaces like TCP/IP to IB to allow running legacy application intact

# Management Model

- IBA management defines a common management infrastructure for
  - Subnet Management - provides methods for a subnet manager to discover and configure IBA devices and manage the fabric
  - General management services
    - Subnet administration - provides nodes with information gathered by the SM and provides a registrar for nodes to register general services they provide
    - Communication establishment & connection management between end nodes
    - Performance management - monitors and reports well-defined performance counters
    - And more...

# Management Model



# Management Model – Packets

- Management is done using Management Datagram (MAD) packets
  - SMP – Subnet Manager MADs
  - GMP – General Management MADs

bytes	bits 31-24	bits 23-16	bits 15-8	bits 7-0				
0	BaseVersion	MgmtClass	ClassVersion	R	Method			
4	Status		ClassSpecific					
8	TransactionID							
12								
16	AttributeID		Reserved					
20	AttributeModifier							
24	Data							
...								
252								

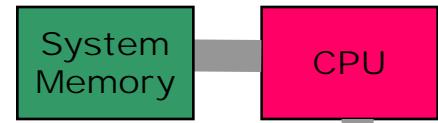
Figure 145 MAD Base Format

# Subnet Management

Topology Discovery  
FDB Initialization  
Fabric Maintenance

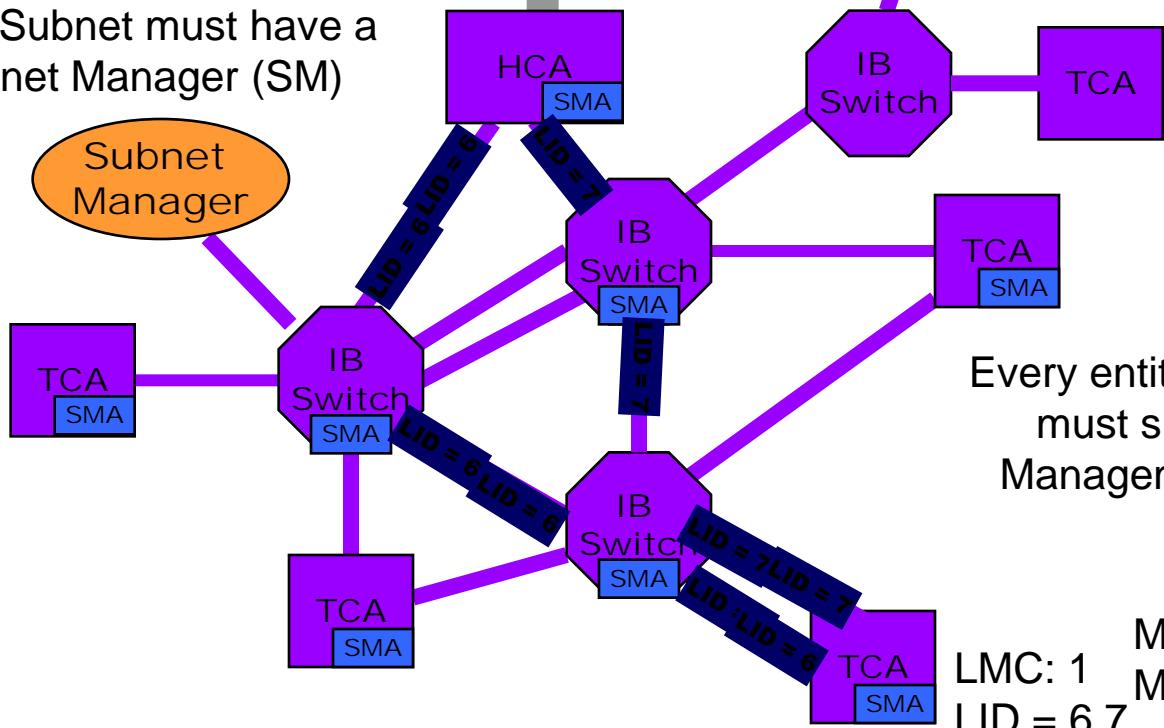
Initialization uses  
Directed Route MADs:

LID Route	Directed Route Vector	LID Route
-----------	-----------------------	-----------



MADs use unreliable  
datagrams

Each Subnet must have a  
Subnet Manager (SM)



Every entity (CA, SW, Router)  
must support a Subnet  
Management Agent (SMA)

LMC: 1      Multipathing: LMC Supports  
LID = 6,7      Multiple LIDS

# Other management entities

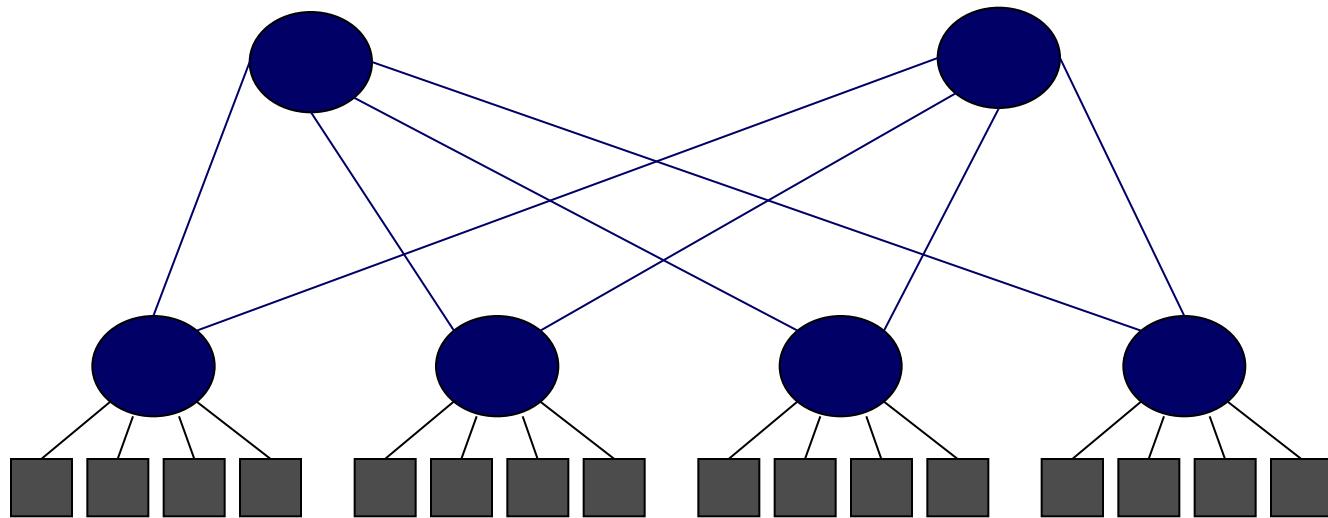
- **Connection Manager (CM)**
  - Establishes connection between end-nodes
- **Performance Management (PM)**
  - Performance Counters
    - Saturating counters
  - Sampling Mechanism
    - Counter works during programmed time period
- **Baseboard Management (BSM)**
  - Access Vital Product Data (VPD)
  - Bridge to/from IBML devices
    - Power Management
    - Hot plug in and removal of modules
    - Monitoring of environmental parameters

# Topologies

- There are several common topologies for an IB fabric
  - Fat Tree – Most popular. A tree where the HCA are the leaf of the tree and that allow full bisectional Bandwidth (BW) between pair of nodes
  - Mash – each node is connected to 4 other nodes: positive and negative X and Y axis
  - 3D mash – Each node is connected to 6 other nodes: positive and negative X, Y and Z axis
  - 2D/3D torus – The ends of the 2D/3D mashes are connected

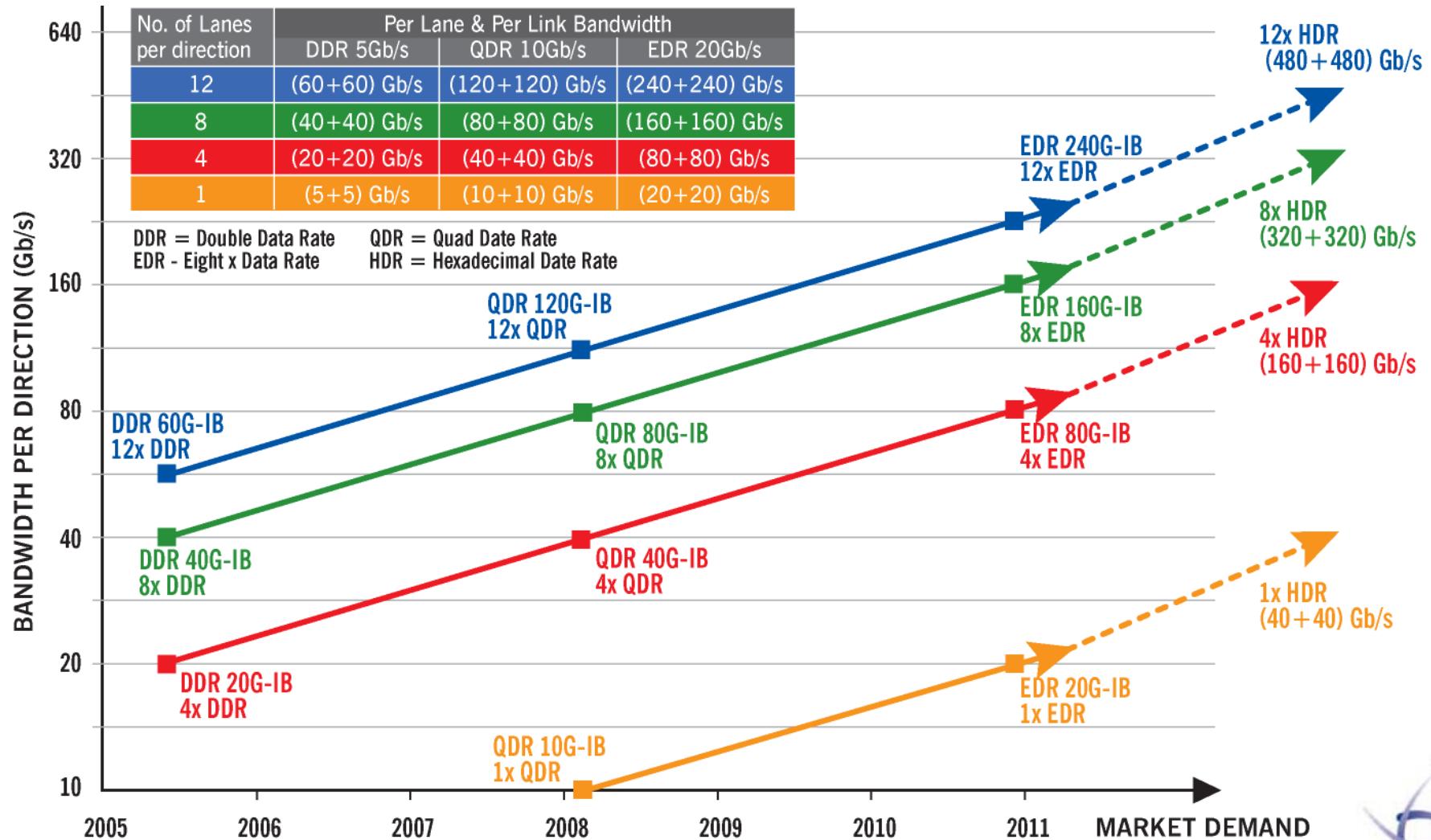
# Topologies – Fat Tree Example

## Full Fat Tree / Full CBB



## Half Fat Tree / Half CBB

# InfiniBand Link Speed Roadmap



# Questions

1. What is the difference between an HCA and a switch?
2. What layers does the InfiniBand specification define?
3. How many wires will be used for a 4x QDR link?

What is the data rate?

What is the effective data rate?

4. What is the maximum packet size in IB?
5. Will InfiniBand fabric drop packets?

If so on which case and what may be the implications?

## 6. What are VLs used for?

How many VLs are there?

Are they all have the same behavior?

## 7. What is LID and what is it used for?

## 8. What is a QP and what is it used for?

## 9. What type of transport services does InfiniBand supports and how reliability is realized?

## 10. What is the role of the Subnet Manager?

Can a cluster run without it?

# Mellanox InfiniBand Products

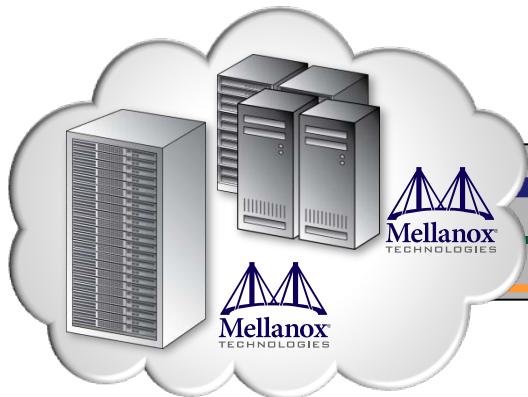


CONFIDENTIAL

# End-to-End Data Center Connectivity



## Server / Compute



## Switch / Gateway



## Storage Front / Back-End

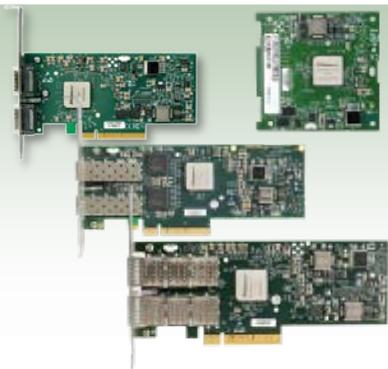


## Mellanox Solutions

### ICs



### Adapters



### Switches/Gateway



### Cables



# HCA Silicon Features



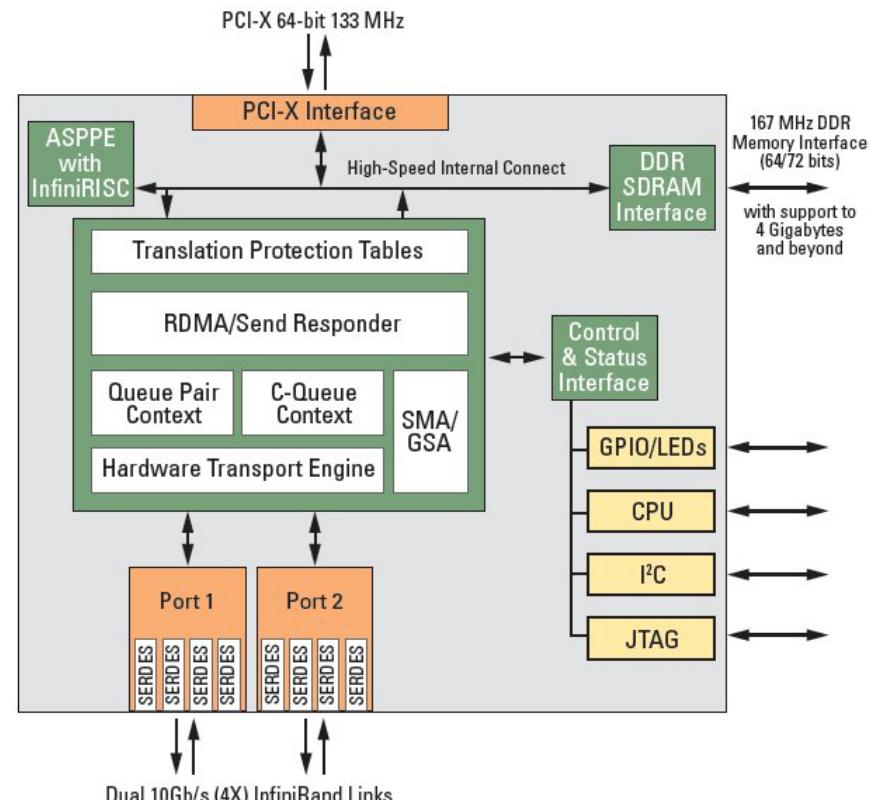
CONFIDENTIAL

# InfiniBand HCA Silicon and Cards



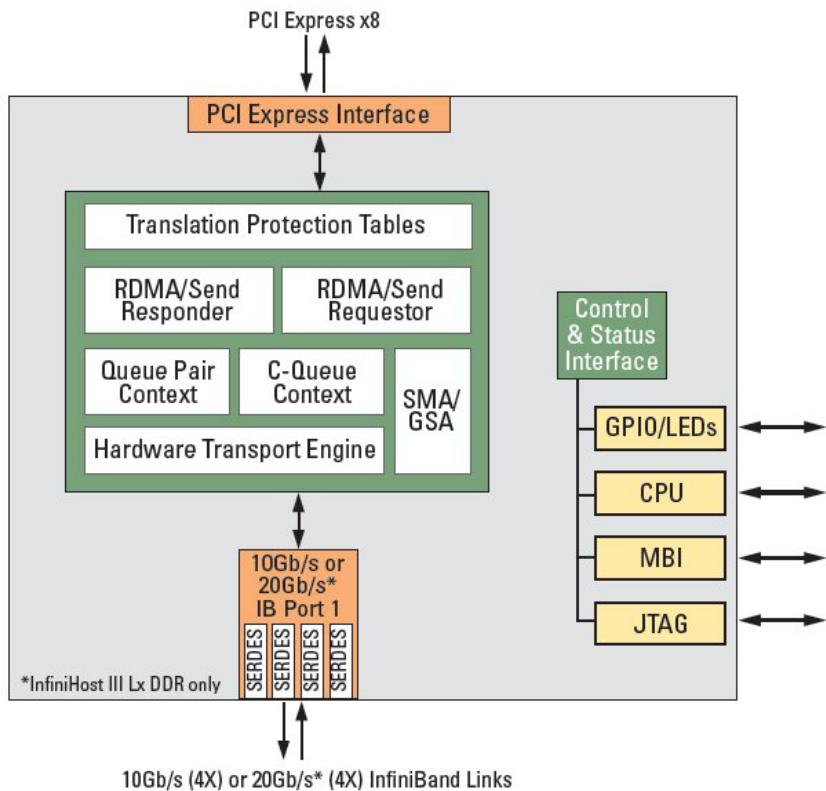
Family	InfiniHost	InfiniHost III Lx	InfiniHost III Ex	ConnectX IB
<b># IB Ports</b>	2 * 10Gb/s	1 * 10,20Gb/s	2 * 10,20Gb/s	1,2 * 10,20,40Gb/s
<b>Max Host Interface</b>	PCI-X	PCIe 1.1 x8	PCIe 1.1 x8	PCIe 2.0 x8 2.5,5GT/s
<b>Max Uni-BW</b>	750MB/s	1500MB/s	1500MB/s	3400MB/s
<b>Latency</b>	4.0 µs	2.91 µs	2.35 µs	0.9 µs
<b>Typ. IC Power</b>	10W One port 10Gb/s	3.5W One port 20Gb/s	10W Both ports 20Gb/s	9.7W Both ports 40Gb/s, PCIe G2
<b>Package (mm)</b>	35x35	16x16	27x27	21x21
<b>RoHS Compliance</b>	R5	R5 R6 IC available	R5 R6 IC available	R5 R6 IC available
<b>China RoHS</b>	Yes	Yes	Yes	Yes

- IBTA v1.2 Compatible
- Dual 10Gb/s Infiniband 4X Ports
- Latency 4µs
- MAX Uni-BW: 750MB/s
- Externally Attached DDR memory
  - Up to 4GB
  - 64 bit addressing support
- 8 Data VLs + Management VL (#15)
- MTU size – up to 2K Bytes
- Support for 2GB Messages
- PCI-X interface – 8Gb/s



InfiniHost Block Diagram

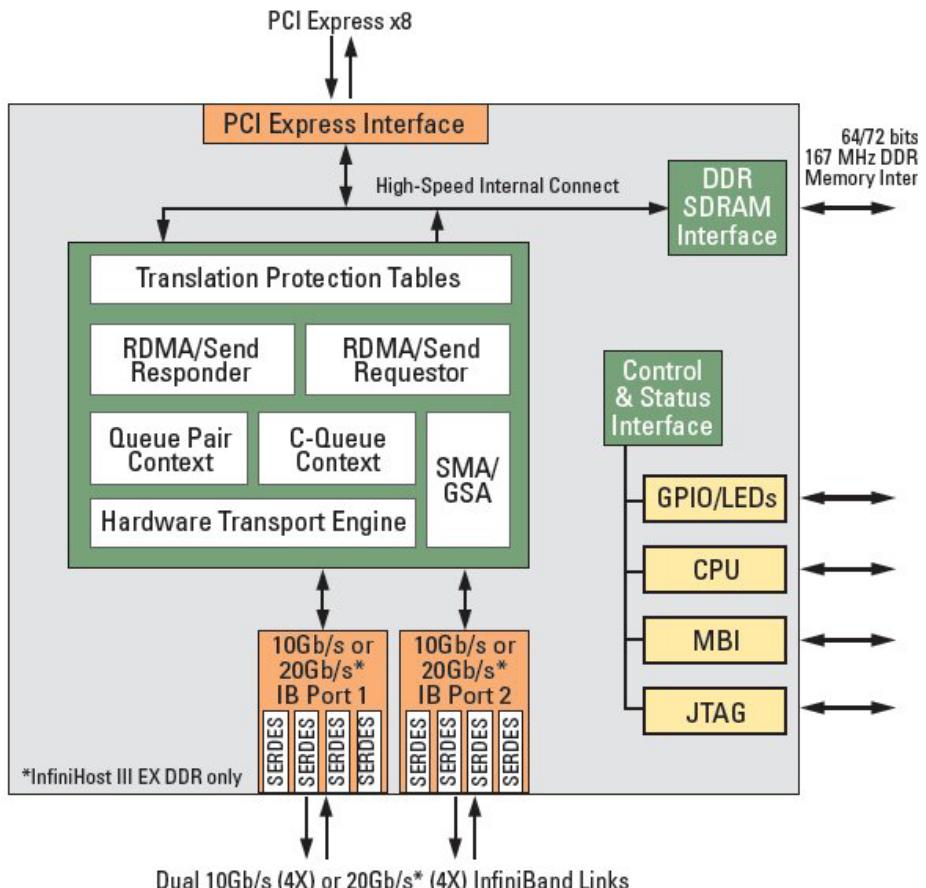
- IBTA v1.2 Compatible
- Single 10Gb/s or 20 Gbp/s Port
- Latency 2.9μs
- MAX Uni-BW:1500MB/s
- 4 Data VLs + Management VL (#15)
- MTU size – up to 2K Bytes
- Support for 2GB Messages
- PCIe 1.1 x8 interface
- Support MSI-X interrupts



*InfiniHost III Lx Block Diagram*

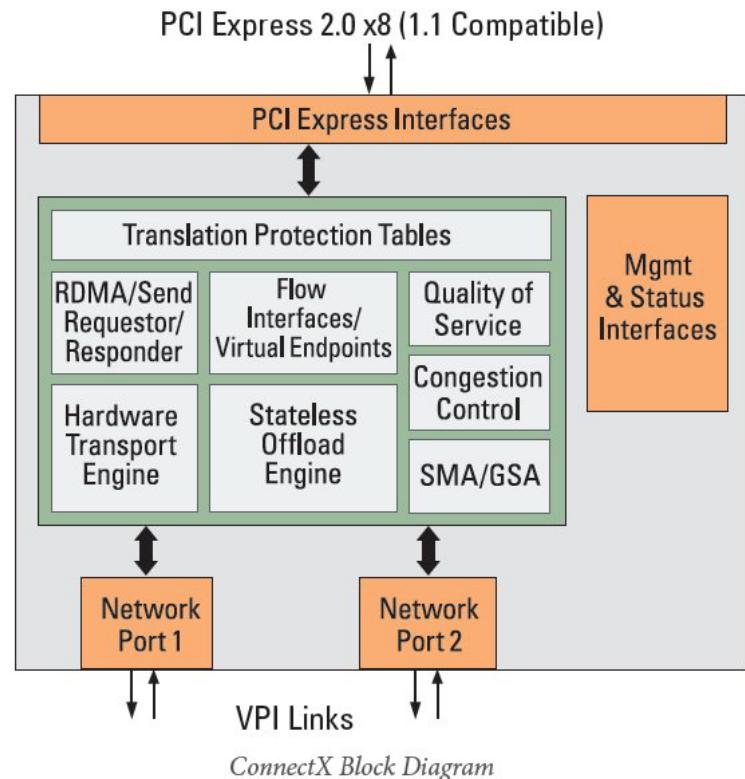
# InfiniHost III Ex

- IBTA v1.2 Compatible
- Dual 10Gb/s or 20 Gbp/s Ports
- Latency 2.35µs
- MAX Uni-BW:1500MB/s
- 8 Data VLs + Management VL (#15)
- MTU size – up to 2K Bytes
- Multicast support
- Support for 2GB Messages
- PCIe 1.1 x8 interface
- Support MSI-X interrupts



*InfiniHost III Ex Block Diagram*

- VPI (Virtual Protocol Interconnect)
- Support InfiniBand and 10GigE
- IBTA v1.2.1 Compatible
- Auto detect 10, 20, 40Gbps InfiniBand or 10GigE per Port
- 8 Data VLs + Management VL (#15)
- MTU size – up to 4K Bytes
- End to End QoS and Congestion Control
- Hardware based I/O Virtualization
- TCP/UDP/IP Stateless Offload
- Fiber Channel Encapsulation (FCoIB or FCoE)
- PCIe 2.0 x8
  - Up to 5GT/s
- Latency 0.9µs
- MAX Uni-BW:3400MB/s



## ConnectX - 2

- Drop-in replacement for ConnectX based devices
- Additional/improved features include
  - Low power
  - IB - Collective Operations Offload
  - Enhanced QoS and Congestion Control
  - SR-IOV virtualization
  - 40 Gbps Ethernet
  - Full HW offload for T11 FCoE

# HCA Cards



CONFIDENTIAL

# IB Adapter Card variations

## ■ Variations:

- Bracket – Short/Tall
- Connectors
  - CX4
  - QSFP
- Speed
  - SDR
  - DDR
  - QDR
- Silicon
  - From InfiniHost to ConnectX-2
- Host Interface
  - PCI-X to PCIe 2 x8



Short Bracket



Tall Bracket

# ConnectX-2 Cards

## ConnectX-2 VP

10/20/40Gb/s InfiniBand  
10 Gigabit Ethernet



QSFP  
SFP+



## ConnectX-2 IB

10/20/40Gb/s InfiniBand



QSFP



## ConnectX-2 EN/ENt

10 Gigabit Ethernet

CX4



SFP+



\*Single-port and OEM-branded Mezzanine cards available

- Fabric consolidation – QSFP and SFP+ connectors
  - 10, 20, 40Gb/s InfiniBand and 10Gig Ethernet
  - Lower TCO (Purchase Cost/ Power/ Service)
  - Saves PCIe slot
- Highest Networking and Storage Performance
  - InfiniBand and LLE
  - TCP/UDP/IP Acceleration
  - FCoE / FCoIB
- Uses
  - IB for IPC, EN for storage
  - EN now, IB in the future

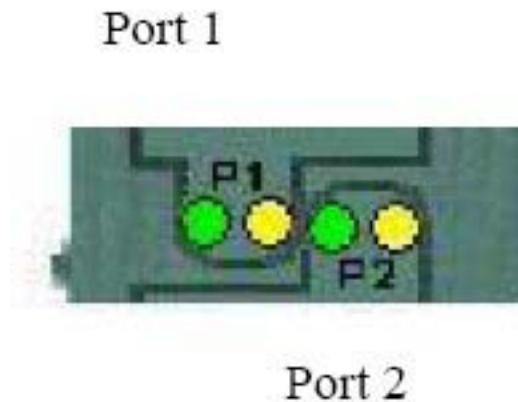


# Card Installation

- Cards are Standard PCI-X or PCIe
- Please Consult Server documentation for instructions
- Copper InfiniBand cables should be carefully attached or detached while maintaining reasonable bend radios

# Port Numbering and LEDs

- When two ports, refer to the picture
- For CX4 Connector LED arrangement is shown in the right picture
- LEDs behavior:
  - Green – Physical link
    - Constant on Good Physical Link
    - Blinking indicates a problem
  - Yellow – Logical link, Data Activity
    - Constant – Logical link up. No data transfer
    - Blinking indicates data transfer



# Switch Silicon Features



CONFIDENTIAL

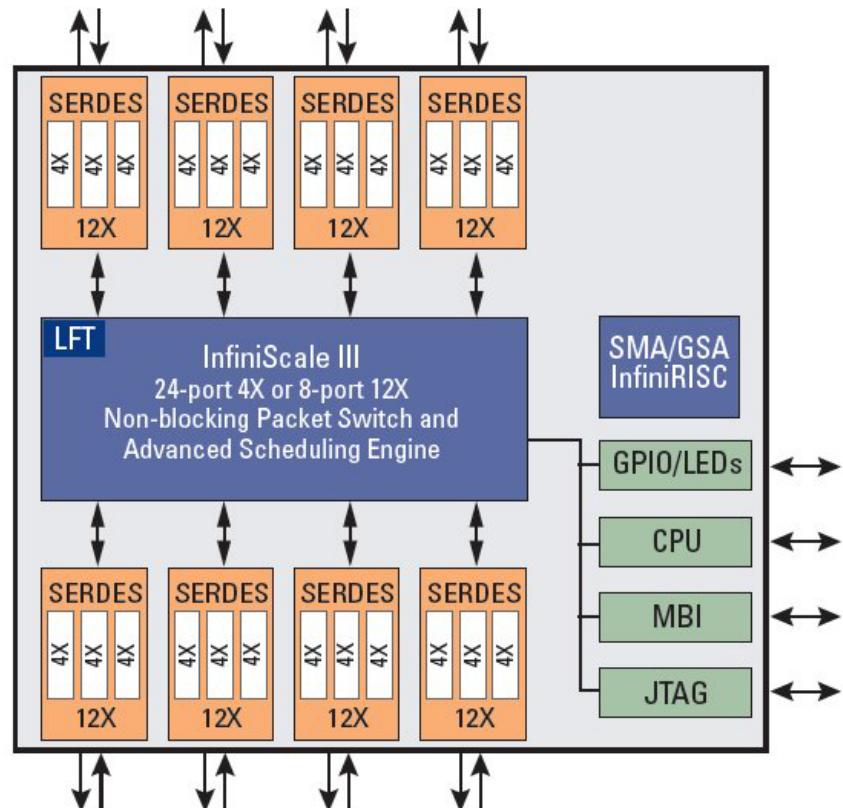
# InfiniBand Switch Silicon



Family	InfiniScale™	InfiniScale™ III	InfiniScale™ IV
# IB Ports	8 (4X) * 10Gb/s	24 (4X) or 8 (12X) * 10, 20Gb/s	36 (4X) or 12 (12X) * 20, 40Gb/s
Ball to Ball Latency	240 ns	200, 140 ns	120, 100 ns
Switching Capacity	160 Gb/s	960 Gb/s	2880 Gb/s
CPU Interface	PCI 2.2 or MPC860 (slave only)	MPC860 (master and slave)	PCIe 2.0 x4
Typ. Power (W)	18	25 (SDR), 30 (DDR)	74 (DDR), 85 (QDR)
Package (mm)	40x40	40x40	45x45
RoHS Compliance	R5	R5 R6 IC available	R5 R6 IC available

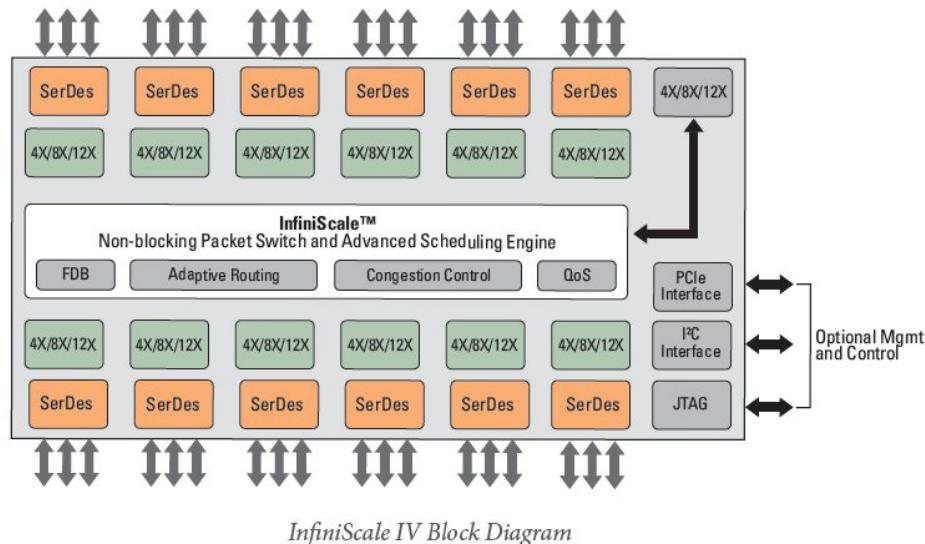
# InfiniScale III

- IBTA v1.2 support
- 24 10 or 20 Gb/s IB 4x ports
- Or 8 30 or 60Gb/s IB 12x ports
- 480Gb/s (SDR) Or 960Gb/s (DDR) switching bandwidth
- Auto negotiation of Port Link Speed
- Programmable Port Mirroring
- Multicast – up to 1K entries
- HW CRC checking and generation



*InfiniScale III Block Diagram*

- IBTA v1.2 support
- 36 port 40Gb/s
- Flexible Port Configuration
  - 4x, 8x, 12x
  - 20 or 40Gb/s Per 4x Port
- 2.88 Tb/s switching capability
- IBTA compliant auto negotiation
- Programmable Port Mirroring
- Multicast – up to 1K entries
- Adaptive Routing
- Congestion Control



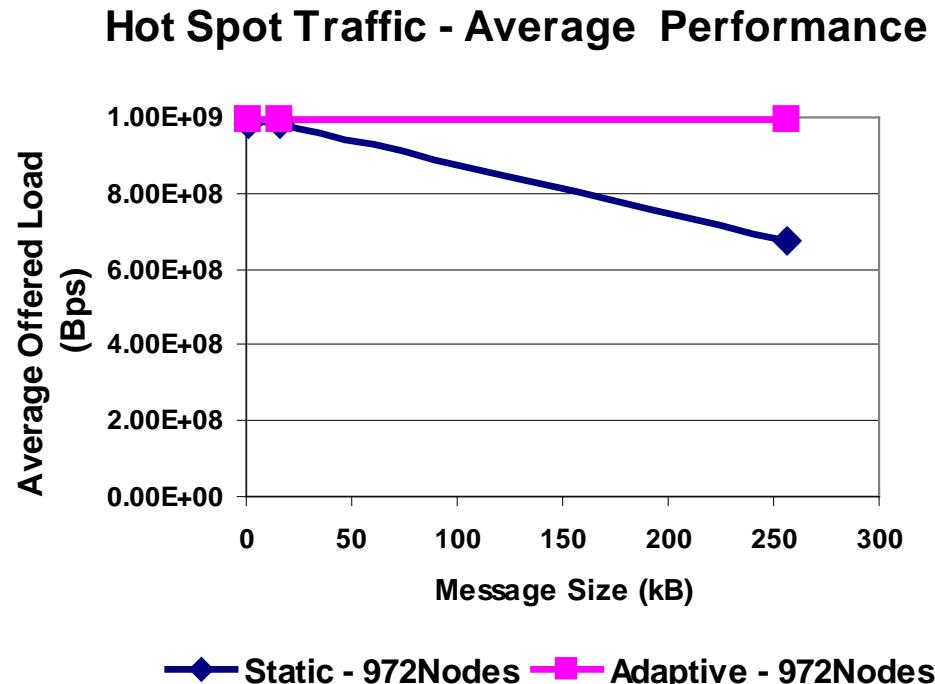
# Superior Scaling

- Fewer switch hops needed, dramatically reduces latency
  - Compared with InfiniScale III DDR latency 140ns

Port range	Tiers		Switch Hops	
	InfiniScale III	InfiniScale IV	InfiniScale III	InfiniScale IV
1 to 24	1	1	1	1
<b>25 to 36</b>	<b>2</b>	<b>1</b>	<b>3</b>	<b>1</b>
37 to 288	2	2	3	3
<b>289 to 648</b>	<b>3</b>	<b>2</b>	<b>5</b>	<b>3</b>
649 to 3,456	3	3	5	5
<b>3,457 to 11,664</b>	<b>4</b>	<b>3</b>	<b>7</b>	<b>5</b>

# InfiniBand Adaptive Routing

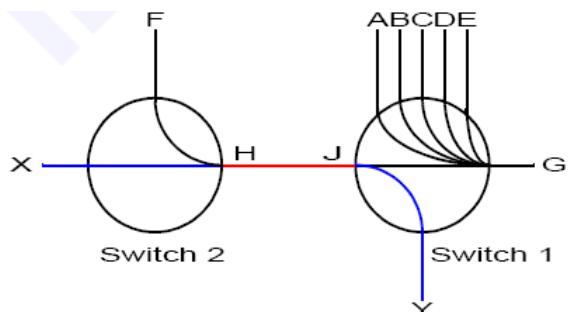
- Maximizes “One to One” random traffic network efficiency
- Dynamically re-routes traffic to alleviate congested ports
  
- Fast path modifications
- No overhead throughput
- Several algorithms for maximum flexibility
  - Randomly select a port
  - Randomly select a port out of N least busy ports
  - Use least busy port
  - Use preferred “static” port if free



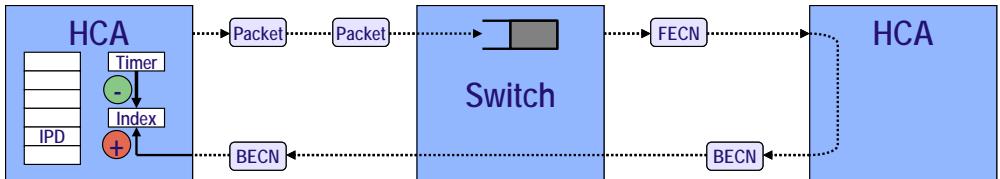
Simulation model (Mellanox):  
972 nodes cases, Hot Spot traffic

# Hardware Congestion Control

- Congestion spots → catastrophic loss of throughput
  - Old techniques are not adequate today

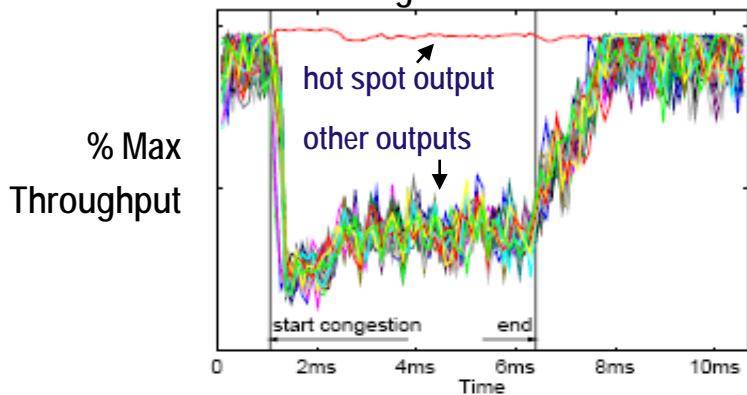


- InfiniBand HW congestion control
  - No a priori network assumptions needed
  - Automatic hot spots discovery
  - Data traffics adjustments
  - No bandwidth oscillation or other stability side effects
  - SM receives notices of congestion
- Ensures maximum effective bandwidth

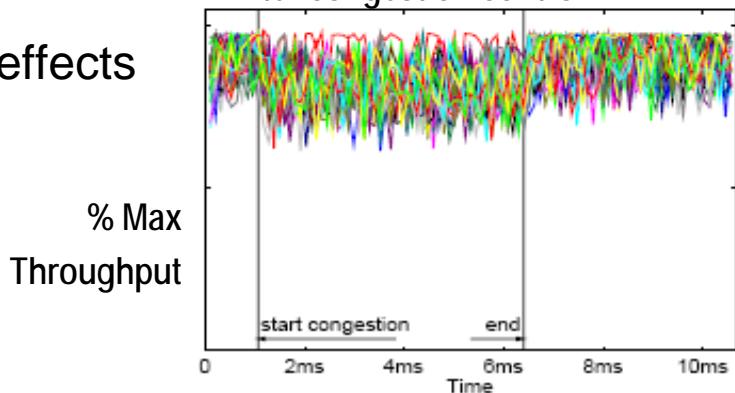


## Simulation results

32-port 3 stage fat-tree network  
High input load, large hot spot degree  
Before congestion control



## After congestion control



"Solving Hot Spot Contention Using InfiniBand Architecture Congestion Control  
IBM Research; IBM Systems and Technology Group; Technical University of Valencia,  
Spain"

# Port Mirroring

- Enables sophisticated traffic monitoring
- Copies or redirects packets to a monitor port
  - Port based mirroring
    - All received packets, all transmitted packets, or both
  - Filter based mirroring
    - Exact match on selected fields
    - Hash matching using Bloom Filter

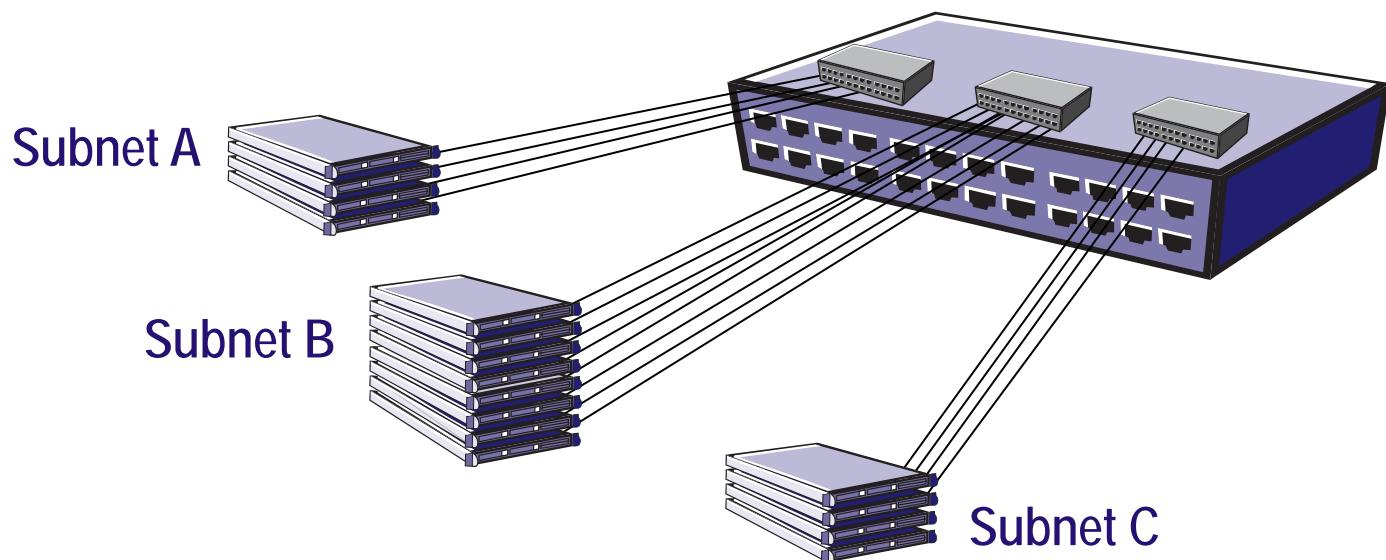
# Multiple Subnet Partitioning

## ■ Enables utility computing

- Virtually partition cluster to suit individual clients' needs
- Secure segregation of each client's network traffic

## ■ Up to 6 independent subnets

- Flexible assignment of ports to subnet
- Dynamic re-configuration



# Switch Systems



CONFIDENTIAL

# InfiniScale III Systems

- Offered as Production Development Kit (PDK)
- 24 Port 4X 1U
- SDR or DDR Variants
- Power consumption:
  - 25W for SDR
  - 34W for DDR



# Mellanox IS4 Systems Family



	EDGE SWITCHES				DIRECTOR SWITCHES			
	IS5025	IS5030	IS5035	MTS3600	IS5100	IS5200	MTS3610	IS5600
<b>Ports</b>	36	36	36	36	108	216	324	648
<b>Switching Capacity</b>	2.88Tb/s	2.88Tb/s	2.88Tb/s	2.88Tb/s	8.64Tb/s	17.28Tb/s	25.9Tb/s	51.8Tb/s
<b>Performance</b>	Non-blocking	Non-blocking	Non-blocking	Non-blocking	Non-blocking	Non-blocking	Non-blocking	Non-blocking
<b>Spine Modules</b>	—	—	—	3	3	6	9	18
<b>Leaf Module (Max)</b>	—	—	—	6	6	12	18	36
<b>Management Modules</b>	None	1-Fixed	1-Fixed	1-Fixed	2	2	2	2
<b>BridgeX Module</b>	—	—	—	Yes	Yes	—	Yes	—
<b>Management Ports</b>	0	1	2	1	2	2	2	2
<b>Installation Kit</b>	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<b>Console Cables</b>	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
<b>PSU Redundancy</b>	Optional	Optional	Optional	Optional	Optional	Optional	Optional	Optional
<b>Fan Redundancy</b>	Optional	Built In	Built In	Built In	Optional	Optional	Built In	Optional
<b>Management</b>	Unmanaged	Lightly Managed	Managed	Managed	Managed	Managed	Managed	Managed

# IS50XX Switch Systems

## ■ 1U 36 Ports systems

- 2.88TB switching



## ■ IS5025

- Unmanaged
- Host Subnet Manager

## ■ IS5030

- Chassis Management
- Fabric Management for small clusters (up to 108)



## ■ IS5035

- Fully Managed
- Fabric Management for large clusters

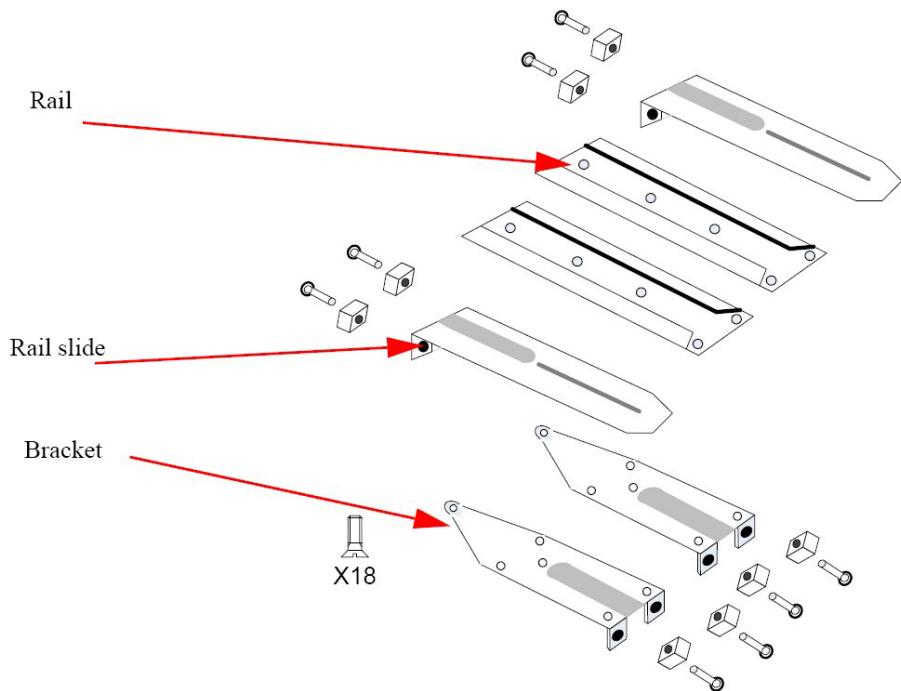


# Accelerating QDR Deployment

# 1U Physical installation

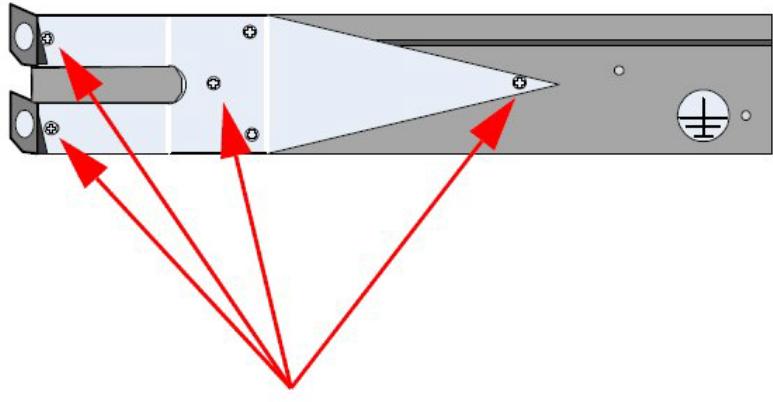
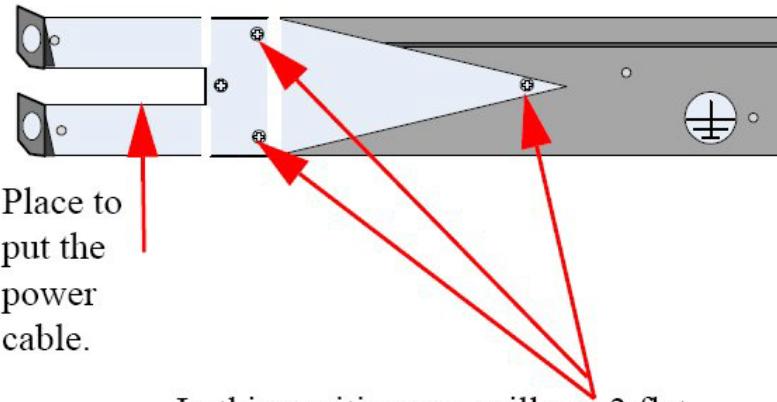
- MIS000079 Installation kit
  - **Can only go into a 19" rack whose vertical supports are between 380mm and 500mm apart.**
  - **Includes the iDataPlex rack.**
- Rack deeper than 500mm:
  - **Order the switch with standard depth.**
  - **Or order the MIS000083 installation kit.**

# 1U installation kit



- Use ESD mat and strap
- Select location of connectors (front or back)

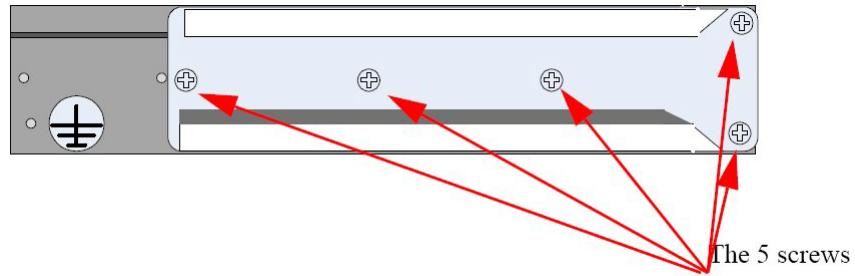
# 1U brackets



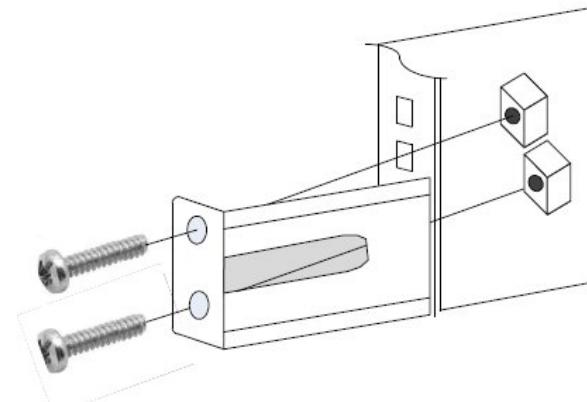
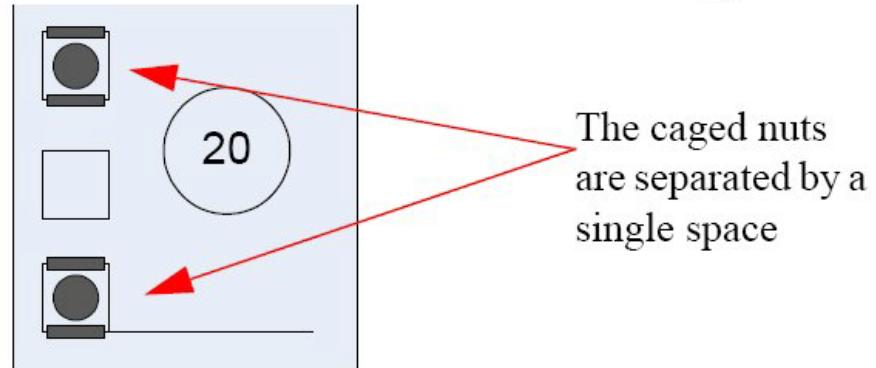
- Depending on your location selection, attach brackets to switch
- Side with bracket will be aligned to vertical rack support

# 1U – Rail and final installation

- Screw rail onto switch



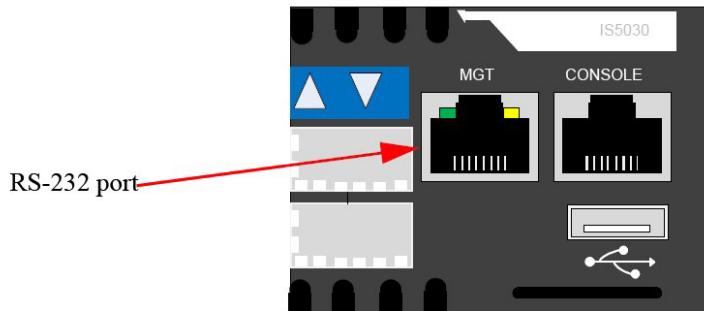
- Clip 4 caged nuts into holes
- Check both sides are in same position number on the rack
- Clip 4 more caged nuts into the holes for brackets
- Install Rail slides
  - If power cable on this side, feed in the slot
- Slide switch, screw into nuts



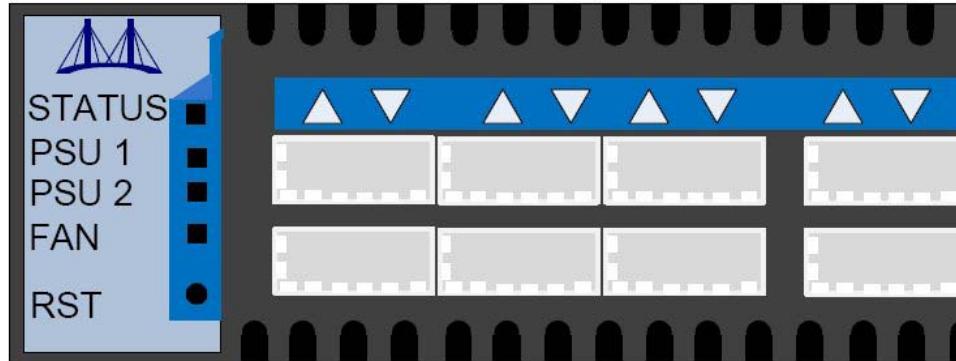
# IS50XX Quick Setup

## ■ Initial Configuration

- Connect to RS232 port (9600,8,1,n,n)
- Login user: admin, password: admin
- Follow the configuration wizard to define:
  - Hostname
  - Management IP (DHCP or static)
  - Admin password
- When done (and saved) CLI and Fabric IT will be available



## ■ Please refer to Fabric IT training



- **Status**
  - Green when has power
  - Red – indicates an error. Turn off and contact support
- **PSU 1, PSU 2**
  - Green – when has power
  - PSU 2 will be off if not installed
- **FAN**
  - Green – normal behavior
  - Yellow – turn off soon (in 2 minutes) to analyze
  - Red – turn off immediately – troubleshoot the fan
- **RST button**
  - Resets the Switch to Factory Defaults

# MT36XX Switch Systems

## ■ Scalable switch architecture

- DDR (20Gb/s) and QDR (40Gb/s)
- Latency as low as 100ns
- Adaptive routing, congestion management, QoS
- Multiple subnets, mirroring

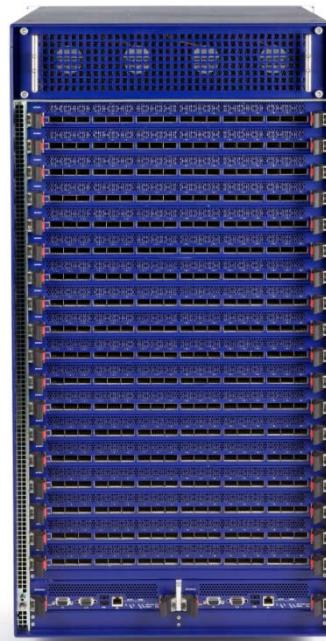


## ■ MTS3600

- 1U 36 port QSFP
- Up to 2.88Tb/s switching capacity

## ■ MTS3610

- 324 QDR ports
- 19U, 18 leaf cards with 18 ports each
- Dual management boards
- Up to 25.9Tb/s switching capacity



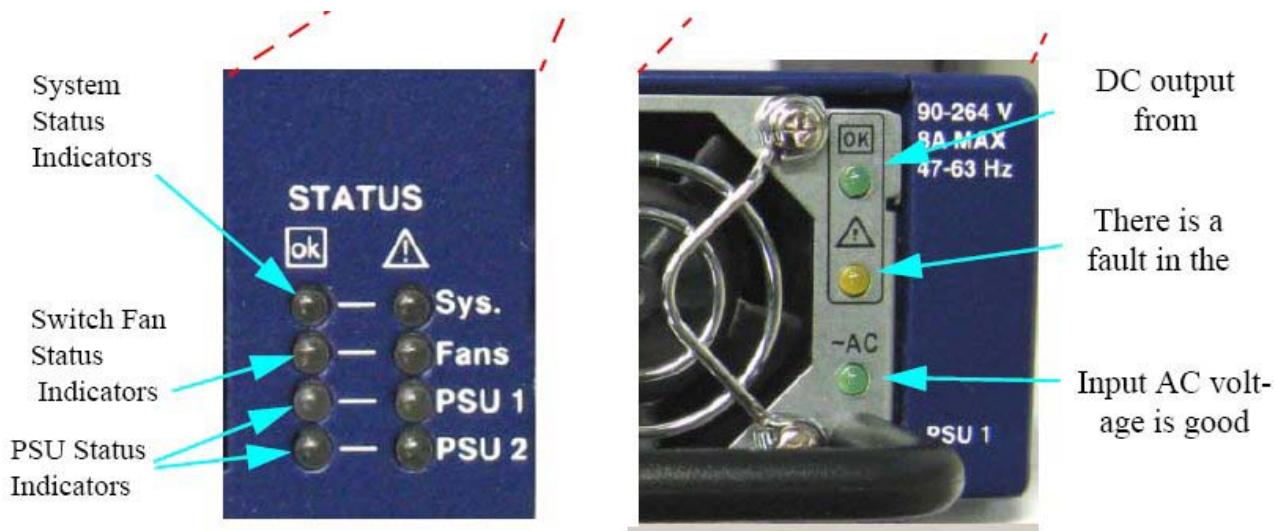
## Accelerating QDR Deployment

# MTS3600 Power side Panel



- Power Side Panel:
  - PSU
  - I2C, Console
  - Management Ethernet
  - USB
  - Status LEDs

# MTS3600 LEDs and Status



## ■ PSU LEDs:

- AC – lit when input voltage is between 90 and 264 Volts
- Warning Sign (yellow) – lit when there is a fault in the power supply
- OK – lit when output from the PSU is +12VDC

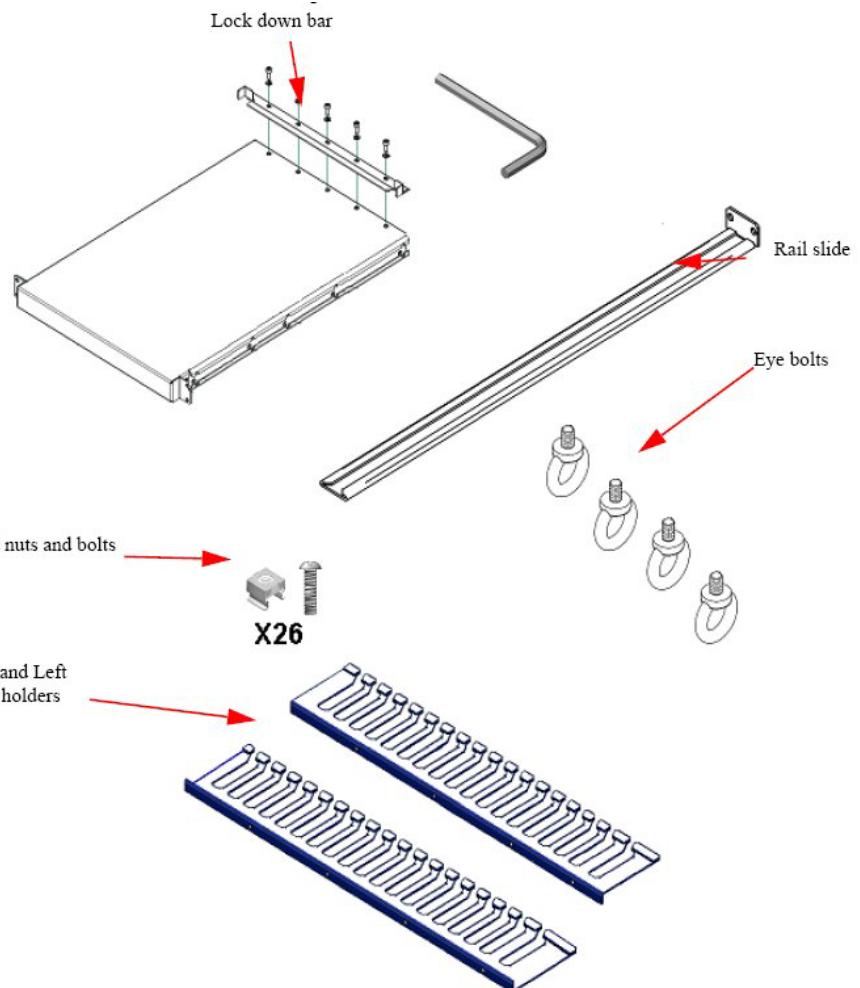
## ■ Status LEDs:

- OK – Green – system/fan/PSU is up and running
- Warning – Yellow – Fault in the system
- Off – No Power

- Package contents:
  - 1 Chassis
  - 1-18 leaf modules
  - 1 leaf fan module
  - 1 spine fan module
  - 9 spine modules
  - 1-2 management modules
  - Power cables, PSU, RJ45 to DB9 cable
- The equipment is heavy! Make sure proper manpower and equipment are used for transporting
- Follow the ESD guidelines in the User Manual

# Chassis installation kit

- Remember to use ESD strap
- Connect wrist strap to chassis ESD connector



# Shelf installation

- Place the Chassis as low as possible
- Insert caged nuts to chosen location
- Screw Rail into Rack
- Connect Shelf to Rack
- Tighten all bolts

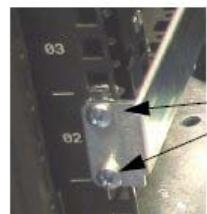
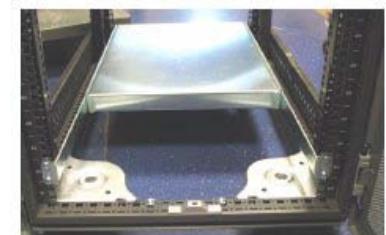
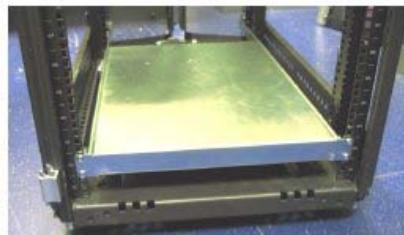
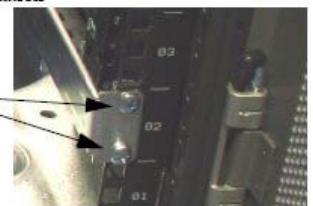


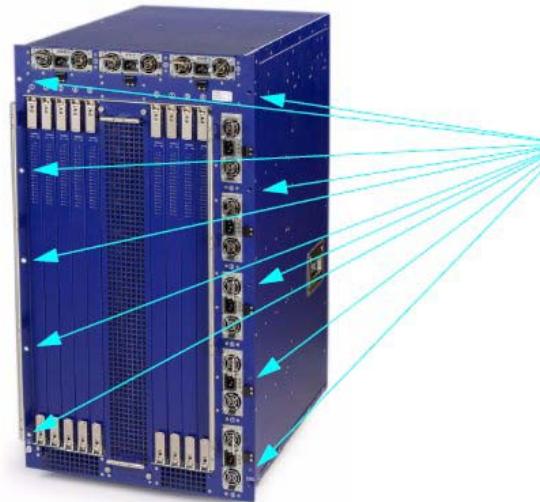
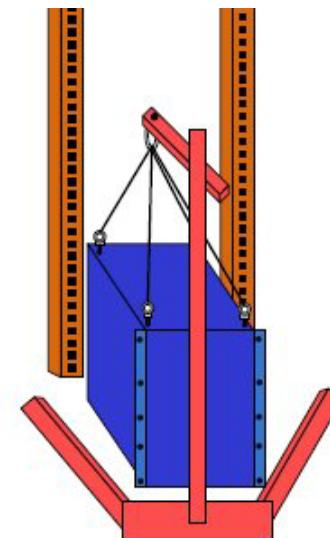
Figure 5: Rail Slide Installation

The caged nuts  
are separated by a  
single space.



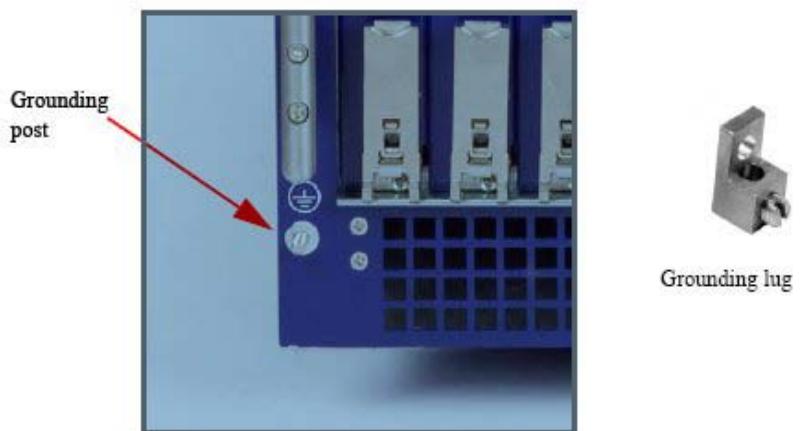
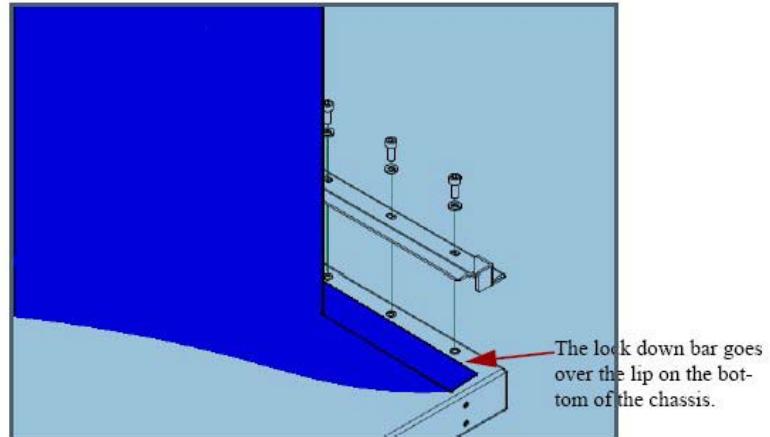
# Chassis insertion

- Screw eye bolts to 4 corners of the top of the chassis
- Connect eye bolts to mechanical lifting device
- Raise Chassis 2cm (1") above shelf
- Place the chassis onto the shelf
- Attach chassis to vertical support using 10 caged nuts



# Chassis - final

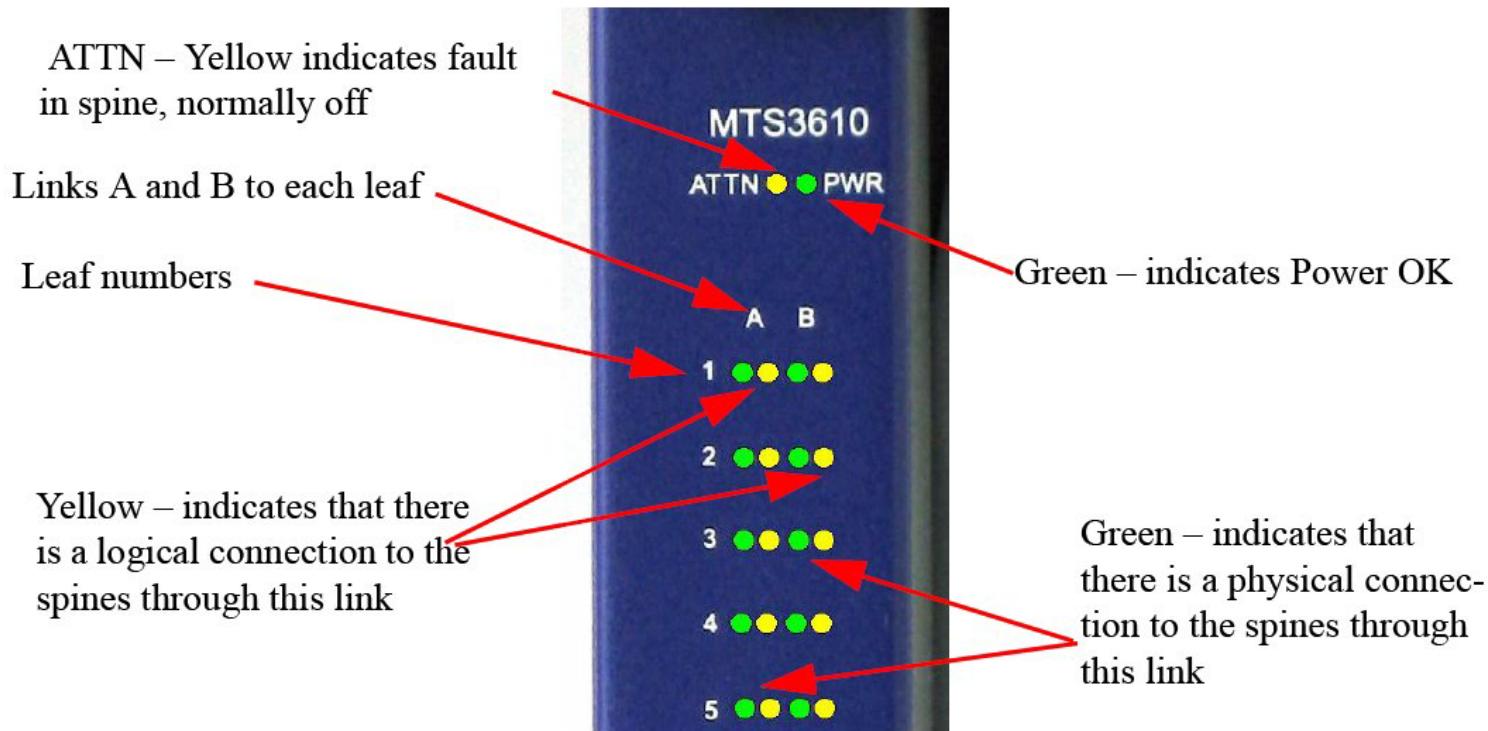
- Place and Screw Lock down bar over lip of chassis
- Connect a valid ground to grounding post
- Install cable holder



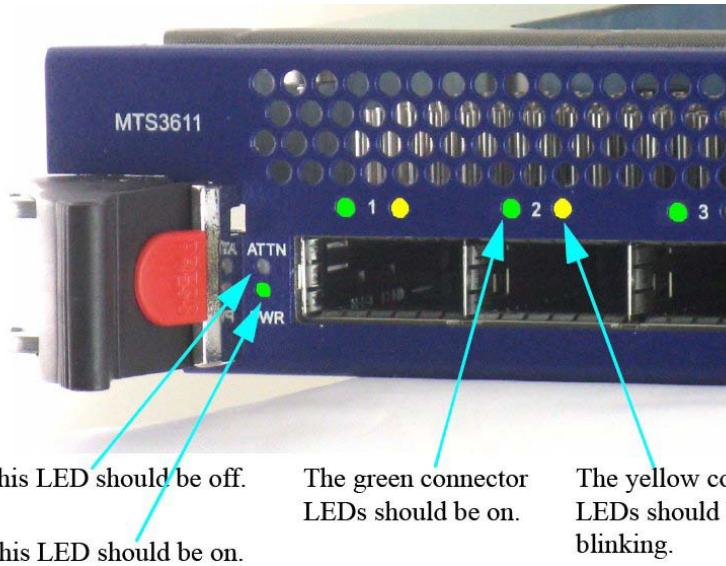
# PSU requirement and Status

- 6 PSU are required for fully populated platform
  - 2 additional PSUs provide failover protection
- Verify PSU LEDs are Green
- Status LEDs on all Management modules are Green
  - Troubleshoot if there is a yellow status on one of the modules

# Spine Module



- **ATTN LEDs should be off**
  - If yellow, troubleshoot



- **ATTN LED should be off**
  - If yellow, troubleshoot

# Setup Guide



RS-232 port

- Connect to the RS-232 port as shown above
- Follow setup steps identical to all Switch Systems
- Refer to Fabric IT for Chassis and Fabric Management

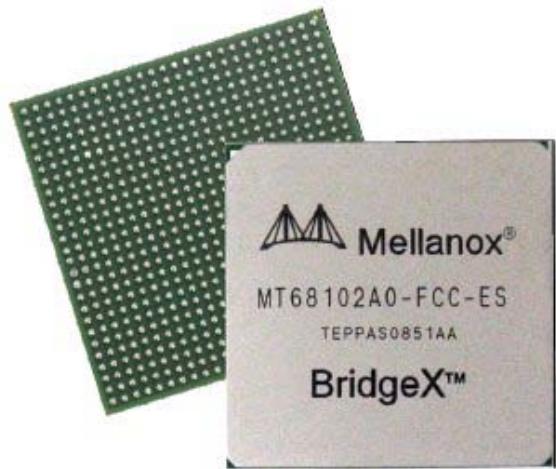
# Gateway Silicon and Systems



CONFIDENTIAL

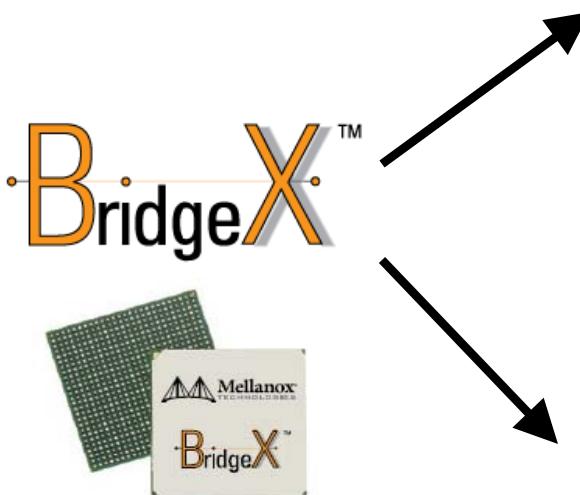
# BX Silicon

- Single chip Solution for IO consolidation
- 2 Infiniband or 6 10GigE uplink Ports
- 6 10GigE Downlink Ports or 8 2/4/8 FC Downlink Ports
- 1024 Virtual NICs per Ethernet Port
- 1024 Virtual HBAs per FC Port
- 8K MAC, VLAN addresses
- 8K WWN addresses
- Interoperable with IB, Ethernet, FC
- Interfaces
  - PCIe
  - Flash memory
  - I2C
  - GPIO
  - MDIO
  - LEDs

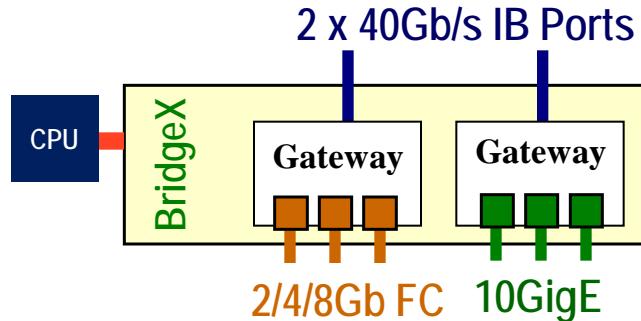


# Introducing BridgeX™ Product Family

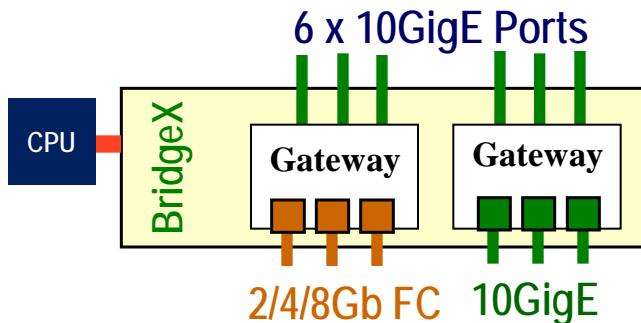
- InfiniBand to Ethernet & FC gateway



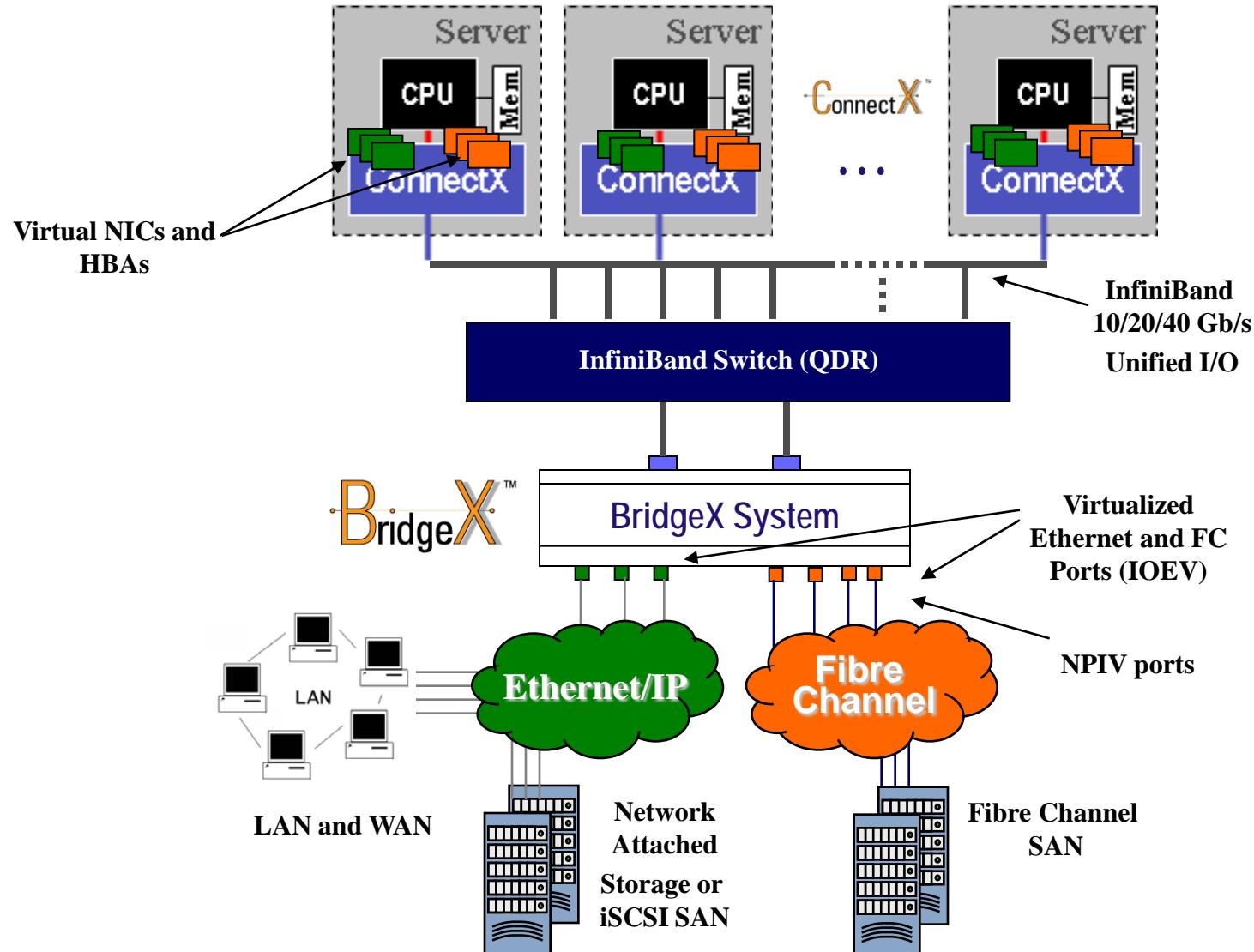
- Ethernet to Ethernet & FC gateway



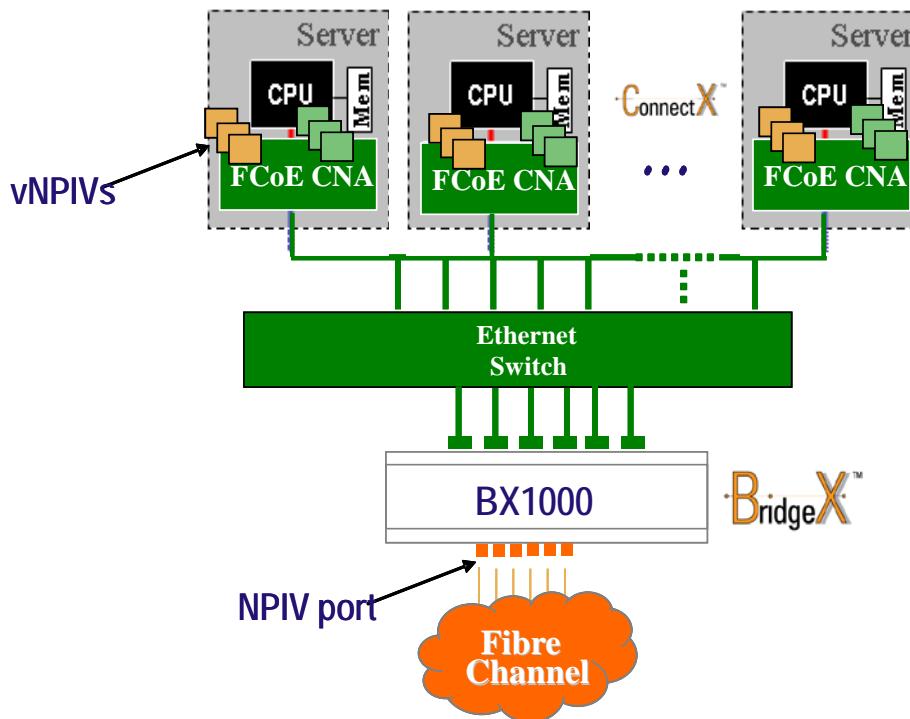
Integrated PHYs  
XAUI, XFI/SFP+, 10GBASE-KR  
Supports 802.3ap including KR, KX4, KX



# BridgeX System Deployment Scenario: FCoIB



# BridgeX – NPIV model



- **N-port ID virtualization (NPIV)**
  - A FC standard used by FC HBAs
  - Multiple N port IDs share a single physical N port
  - Similar to having multiple vNICs to one physical port
- **Virtual NPIVs**
  - Instantiated in the hosts
- **BridgeX and EN switch are invisible**
  - Hosts see FC-SAN cloud
  - FC-SAN sees the hosts

- BX4010: 1U with 1 BridgeX device
  - Uplink Ports: 2 x CX4
  - Downlink ports: SFP+ configurable as 10GbE or 2/4/8G FC
  - Flexibility in port configuration – EN or FC
- Dual hot-swappable redundant power supplies
- Replaceable fan drawer
- Embedded management
- CX4 to QSFP Hybrid Cables



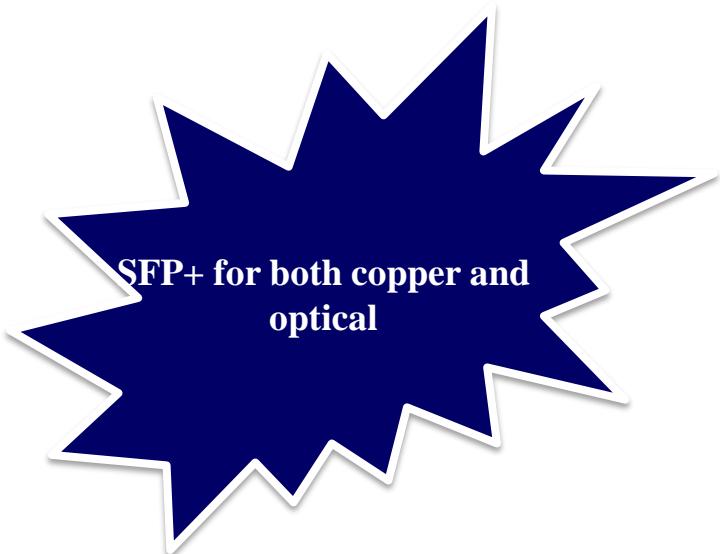
# BridgeX Interoperability

## ■ 10 Gigabit Ethernet Switches

- Cisco Nexus 5020
- Cisco Cat6K
- Arista 24 / 48 port switches
- HP ProCurve
- Juniper EX series
- Blade Networks
- Dell

## ■ Fibre Channel Switches

- Cisco MDS Series
- Brocade



SFP+ for both copper and optical

# Access and configuration

## ■ The Gateway can be accessed using:

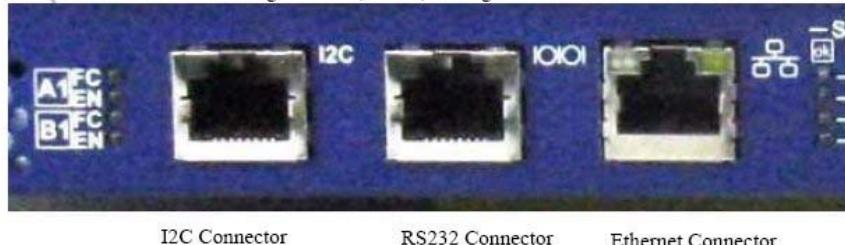
- Serial Port
- SSH
- Telnet

## ■ Serial Port access

- Cable HAR000028
  - 9600,8,1,n,n
- Cable HAR000034
  - 19200,8,1,n,n,
- User: admin
- Password: password

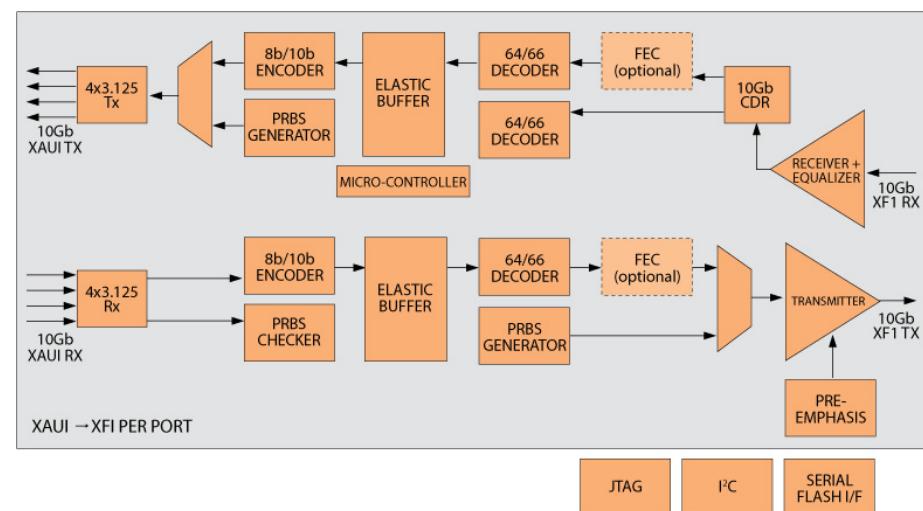
## ■ IP Access

- Initial IP configuration: 172.22.2.2
- SSH or Telnet
- User: admin
- Password: password



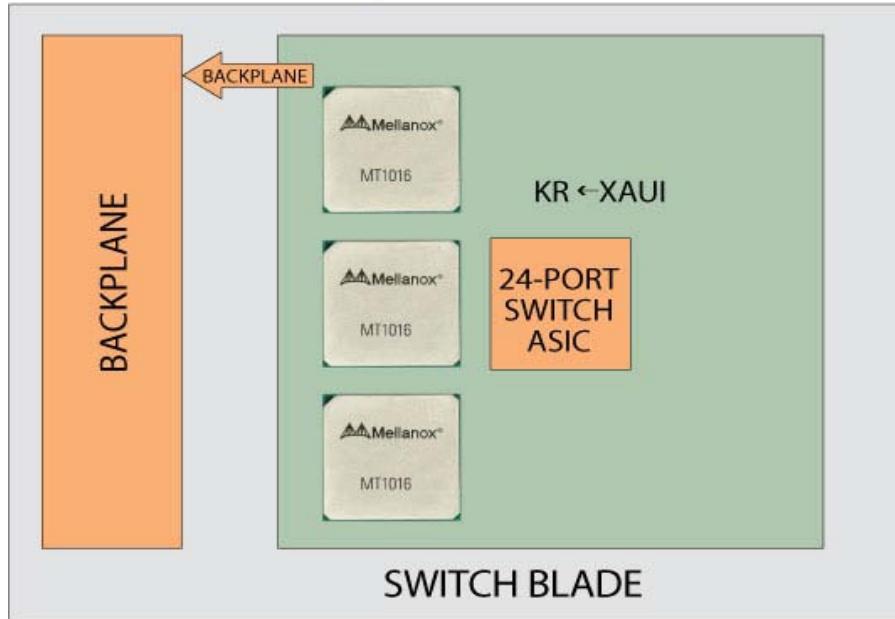
# MT1016 – Mellanox 10GigE PHY Device

- Flexible device supporting
  - XAUI to XFI/SFI
  - XAUI to 10GBASE-KR
  - 10GBASE-KR to XFI/SFI
- High density PHY - 6 ports
- Lowest latency – 80ns
- Small real-estate, 31x31 HFCBGA
- Support for IEEE802.3ap
- Auto-negotiation to 1G and 10G
- Support for receive equalizer
- Support for pre-emphasis
- Supports optional FEC encoder / decoder
- Complete 1G and 10G PCS layers
- Supports internal loopbacks
- XAUI to XFI – 1.95W / port

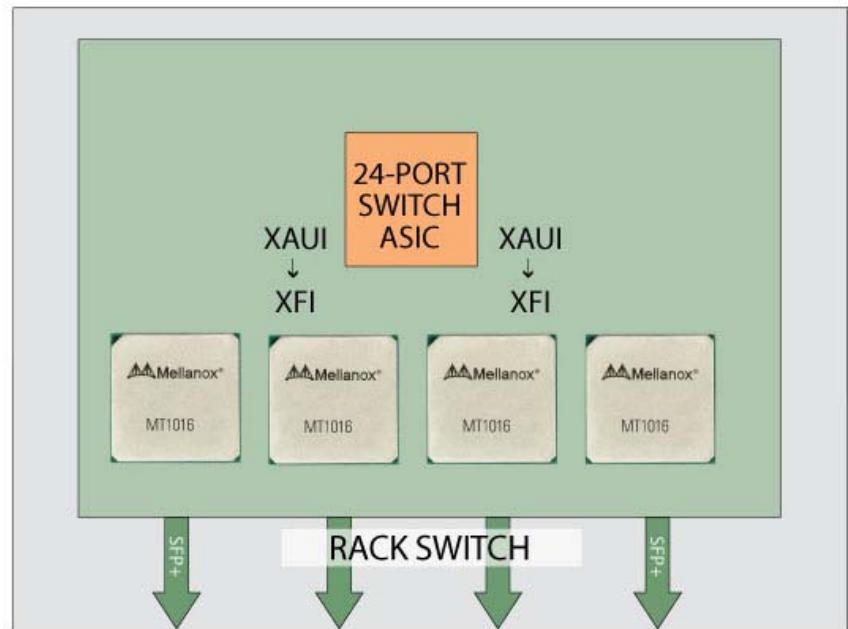


# MT1016 10GigE PHY Applications

3 MT1016 10GigE PHYs for a 16/18 Port Backplane Solution



4 MT1016 10GigE PHYs for a 24 Port Rack Switch Solution



# Cables



CONFIDENTIAL

# Mellanox Cable Product Line

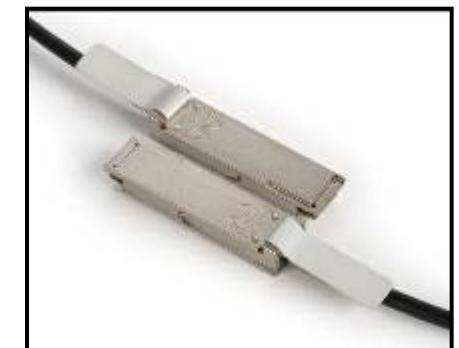
- Provide high-quality cost effective cables to interconnect MLNX HCA and switch product offerings

- Best price/performance
  - Superior signal integrity at each length
  - Very low Bit Error Rate (BER)
- End-to-end validation on Mellanox HCAs and switch silicon
  - Proven server and storage interconnect solution
- Serial numbers on each end
  - Eases installation



- A full range of passive/active copper and fiber cables

- 10GBASE-CX4/CR
- InfiniBand SDR/DDR/QDR



# Cable Characteristics

## ■ Copper

- QSFP and CX4 Connectors
- Maximum reach:
  - 8 meter for 24 AWG, 20Gb/s CX4
  - 7 meter for QSFP 26AWG, 40Gb/s
- Available Lengths: 0.5, 1, 2, 3, 4, 5, 6, 7, 8

## ■ Fiber

- QSFP
- 40Gb/s Maximum Bandwidth
- Lengths: 5, 10, 20, 30

# Questions



CONFIDENTIAL

# Questions

## ■ Adapters and Silicon

- Which Connectors are available for Ethernet Adapters?
- What is the Max Speed of ConnectX based Adapters?
- What is the Host Speed (PCI) for each of the HCA silicon?

## ■ Switch Systems

- What is the Max Switching capability of IS4?
- Name and explain differences between IS50XX systems.
- How many external Ports are available per Leaf module?

## ■ BridgeX (BX)

- Draw a network diagram which uses BX to bridge between IB and Ethernet
- How many Downlink Ethernet ports are available on BX?
- In an FCoIB mode, how many FC ports are available?

# InfiniBand Linux SW Stack

MLNX\_OFED

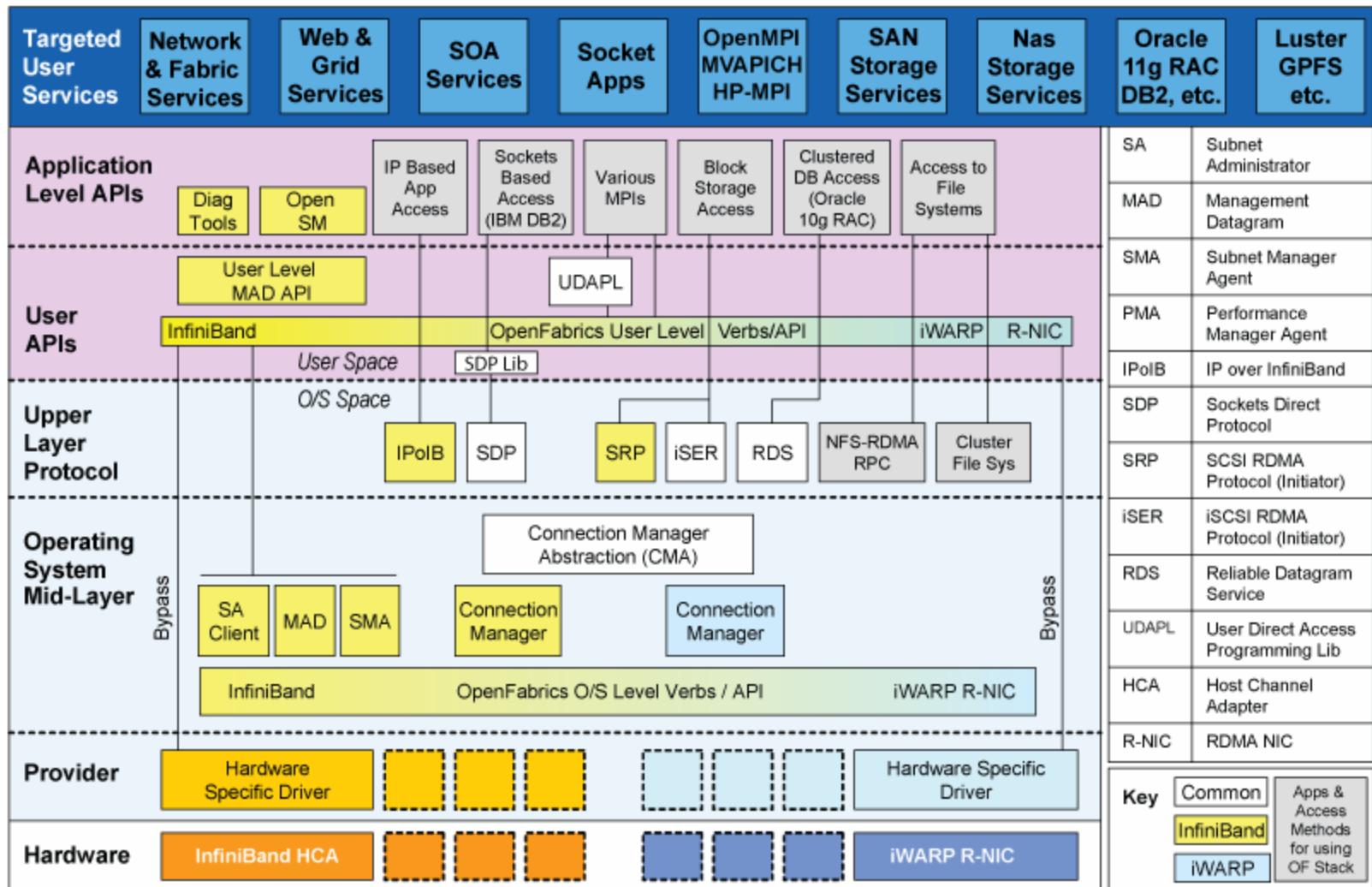


CONFIDENTIAL

# OpenFabrics Enterprise Distribution (OFED)

- Open Fabrics Enterprise Distribution (OFED) is a complete SW stack for RDMA capable devices.
- Contains low level drivers, core, Upper Layer Protocols (ULPs), Tools and documents
- Available on OpenFabrics.org or as a Mellanox supported package at:
  - [http://www.mellanox.com/content/pages.php?pg=products\\_dyn&product\\_family=26&menu\\_section=34](http://www.mellanox.com/content/pages.php?pg=products_dyn&product_family=26&menu_section=34)
- Mellanox OFED is a single Virtual Protocol Internconnect (VPI) software stack based on the OFED stack
  - Operates across all Mellanox network adapters
  - Supports:
    - 10, 20 and 40Gb/s InfiniBand (SDR, DDR and QDR IB)
    - 10Gb/s Ethernet (10GigE)
    - Fibre Channel over Ethernet (FCoE)
    - 2.5 or 5.0 GT/s PCI Express 2.0

# The SW stack



- Mellanox OFED is delivered as an ISO image.
- The ISO image contains both source code and binary RPMs for selected Linux distributions.
- It also contains installation scripts called `mlnxofedinstall`. The `install` script performs the necessary steps to accomplish the following:
  - Discovers the currently installed kernel
  - Uninstalls any IB stacks that are part of the standard operating system distribution or other commercial IB stacks
  - Installs the Mellanox OFED binary RPMs if they are available for the current kernel
  - Identifies the currently installed IB HCA and perform the required firmware updates

# MLNX\_OFED Installation

## ■ Pre-built RPM install.

- 1. mount -o rw,loop MLNX\_OFED\_LINUX-1.4-rhel5.3.iso /mnt
- 2. cd /mnt
- 3. ./mlnxofedinstall

## ■ Building RPMs for un-supported kernels.

- 1. mount -o rw,loop MLNX\_OFED\_LINUX-1.4-rhel5.3.iso /mnt
- 2. cd /mnt/src
- 3. cp OFED-1.4.tgz /root (this is the original OFED distribution tarball)
- 4. tar zxvf OFED-1.4.tgz
- 5. cd OFED-1.4
- 6. copy ofed.conf to OFED-1.4 directory
- 7. ./install.pl -c ofed.conf

# Configuration

## ■ Loading and Unloading the IB stack

- /etc/infiniband/openib.conf controls boot time configuration

```
# Start HCA driver upon boot  
ONBOOT=yes
```

```
# Load IPoIB  
IPOIB_LOAD=yes
```

- Manually start and stop the stack once the node has booted
  - /etc/init.d/openibd start|stop|restart|status

# OpenSM Subnet Manager



CONFIDENTIAL

# OpenSM - Features

- OpenSM (osm) is an Infiniband compliant subnet manager.
- Included in Linux Open Fabrics Enterprise Distribution.
- Ability to run several instance of osm on the cluster in a Master/Slave(s) configuration for redundancy.
- Partitions (p-key) support
- QoS support
- Enhanced routing algorithms:
  - Min-hop
  - Up-down
  - Fat-tree
  - LASH
  - DOR

# Running OpenSm

## ■ Command line

- Default (no parameters)
  - Scans and initializes the IB fabric and will occasionally sweep for changes
- opensm –h for usage flags
  - E.g. to start with up-down routing: opensm --routing\_engine updn
- Run is logged to two files:
  - /var/log/messages – opensm messages, registers only general major events
  - /var/log/opensm.log - details of reported errors.

## ■ Start on Boot

- As a daemon:
  - /etc/init.d/opensmd start|stop|restart|status
  - /etc/opensm.conf for default parameters

```
# ONBOOT
# To start OpenSM automatically set ONBOOT=yes
ONBOOT=yes
```

## ■ SM detection

- /etc/init.d/opensd status
  - Shows opensm runtime status on a machine
- sminfo
  - Shows master and standby subnets running on the cluster

# OpenSM Command Line parameters

## ■ A few important command line parameters:

- c, --cache-options. Write out a list of all tunable OpenSM parameters, including their current values from the command line as well as defaults for others, into the file /var/cache/opensm. This file can then be modified to change OSM parameters, such as HOQ (Head of Queue timer).
- g, --guid This option specifies the local port GUID value with which OpenSM should bind. OpenSM may be bound to 1 port at a time. This option is used if the SM needs to bind to Port 2 of an HCA.
- R, --routing\_engine This option chooses routing engine instead of Min Hop algorithm (default). Supported engines: updn, file, ftree, lash
- x, --honor\_guid2lid. This option forces OpenSM to honor the guid2lid file, when it comes out of Standby state, if such file exists under /var/cache/opensm
- V This option sets the maximum verbosity level and forces log flushing.

# Routing Algorithms

- **Min Hop algorithm (DEFAULT)**
  - Based on the minimum hops to each node where the path length is optimized.
- **UPDN unicast routing algorithm**
  - Based on the minimum hops to each node, but it is constrained to ranking rules. This algorithm should be chosen if the subnet is not a pure Fat Tree, and a deadlock may occur due to a loop in the subnet.
    - Root GUID list file can be specified using the –a option
- **Fat Tree unicast routing algorithm**
  - This algorithm optimizes routing for a congestion-free “shift” communication pattern. It should be chosen if a subnet is a symmetrical Fat Tree of various types, not just a K-ary-N-Tree: non-constant K, not fully staffed, and for any CBB ratio. Similar to UPDN, Fat Tree routing is constrained to ranking rules.
    - Root GUID list file can be specified using the –a option
- **Addition algorithms**
  - LASH - Uses InfiniBand virtual layers (SL) to provide deadlock-free shortest-path routing.
  - DOR. This provides deadlock free routes for hypercube and mesh clusters
  - Table Based. A file method which can load routes from a table.

# OFED Tools



CONFIDENTIAL

## Single Node

ibv\_devinfo

ibstat

lbportstate

ibroute

smpquery

perfquery

## SRC/DST Pair

lbdiaqpath

ibtracert

ibv\_rc\_pingpong

ibv\_srq\_pingpong

ibv\_ud\_pingpong

ib\_send\_bw

ib\_write\_bw

## Network

lbdiaqnet

ibnetdiscover

ibhosts

lbswitches

saquery

sminfo

smpdump

# Node Based Tools



CONFIDENTIAL

# Determine if driver is loaded

- **/etc/init.d/openibd status**
  - HCA driver is loaded
  - Configured devices
    - Ib0
    - Ib1
    - OFED modules are loaded
      - ib\_ipoib
      - ib\_mthca
      - ib\_core
      - ib\_srp

# Determine modules that are loaded

## ■ **lsmod**

- ib\_core
- ib\_mthca
- ib\_mad
- ib\_sa
- ib\_cm
- ib\_uverbs
- ib\_srp
- ib\_ipoib

## ■ **modinfo 'module name'**

- List all parameters accepted by the module
- Module parameter can be added to /etc/modprobe.conf

## ■ **ibstat**

- displays basic information obtained from the local IB driver.
- Normal output includes Firmware version, GUIDS, LID, SMLID, port state, link width active, and port physical state.
- Has options to list CAs and/or Ports.

## ■ **ibv\_devinfo**

- Reports similar information to ibstat
- Also includes PSID and an extended verbose mode (-v).

## ■ **/sys/class/infiniband**

- File system which reports driver and other ULP information.
  - e.g. [root@ibd001 /]# cat /sys/class/infiniband/mlx4\_0/board\_id  
MT\_04A0110002

# HCA Firmware management

## ■ Determine HCA firmware version

- /usr/bin/ibv\_devinfo
- /usr/bin/mstflint -d mlx4\_0 v
- /usr/bin/mstflint -d 07:00.0 q

## ■ Burn new HCA firmware

- /usr/bin/mstflint [switches] <command> [parameters...]
- /usr/bin/mstflint -d mlx4\_0 -i fw.bin b

# Switch Firmware management

- Determine IS4 firmware version
  - /usr/bin/flint –d lid-6 q
- Burn new IS4 firmware
  - /usr/bin/flint –d lid-6 –i fw.img b

---

**Note:** Mellanox FW Tools (MFT) package that contains flint tool can be found at:  
[http://www.mellanox.com/content/pages.php?pg=firmware\\_HCA\\_FW\\_update](http://www.mellanox.com/content/pages.php?pg=firmware_HCA_FW_update)

---

# Node management utilities

- **perfquery**
  - Obtains and/or clears the basic performance and error counters from the specified node
  - Can be used to check port counters of any port in the cluster using ‘perfquery <lid> <port number>’
- **ibportstate**
  - Query, change state (i.e. disable), or speed of Port
    - ibportstate 38 1 query
- **ibroute**
  - Dumps routes within a switch
- **smpquery**
  - Dump SMP query parameters, including:
    - nodeinfo, nodedesc, switchinfo, pkeys, sl2vl, vlarb, guids

# Performance tests

## ■ Run performance tests

- /usr/bin/ib\_write\_bw
- /usr/bin/ib\_write\_lat
- /usr/bin/ib\_read\_bw
- /usr/bin/ib\_read\_lat
- /usr/bin/ib\_send\_bw
- /usr/bin/ib\_send\_lat

## ■ Usage

- Server: <test name> <options>
- Client: <test name> <options> <server IP address>

---

**Note:** Same options must be passed to both server and client. Use –h for all options.

---

# Collecting debug information

## ■ Collect debug information if driver load fails

- **mstregdump**
  - Internal register dump is produced on standard output
  - Store it in file for analysis in Mellanox
  - Examples
    - `mstregdump 13:00.0 > dumpfile_1.txt`
    - `mstregdump mthca > dumpfile_2.txt`
- **mstvpd mthca0**
- **/var/log/messages**
  - `tail -n 500 /var/log/messages > messages_1.txt`
  - `dmesg > dmesg_1.txt`

# Cluster Based Tools



CONFIDENTIAL

- Open source Linux tools
- pdsh allows to run same command on multiple machines
  - Example
    - ‘pdsh -w ibc0[01-10] ls’ will run ls command on ibc001 through ibc010
- dshbak formats output of pdsh into more readable form
  - -c flag will make nodes with identical output be grouped in one listing
  - Example
    - pdsh -w ibd0[02-32] ‘ibstat | grep State’ | dshbak -c
      - ibd[002-032]
      - -----
      - State: Initializing
      - State: Down

# Cluster utilities

## ■ ibswitches

- Lists all switches in cluster

## ■ ibhosts

- Lists all HCAs in cluster

## ■ ibtracert

- Shows path between two lids

```
[root@ibd001 mft-2.5.0]# ibtracert -G 0x0002c90300001481 0x0002c90300001489
From ca {0x0002c90300001480} portnum 1 lid 12-12 "ibd017 HCA-1"
[1] -> switch port {0x000b8cffff002772}[5] lid 39-39 "MT47396 Infiniscale-III Mellanox Technologies"
[6] -> ca port {0x0002c90300001489}[1] lid 15-15 "ibd012 HCA-1"
To ca {0x0002c90300001488} portnum 1 lid 15-15 "ibd012 HCA-1"
```

# Cluster utilities - ibdiagnet / ibdiagpath

## ■ Integrated diagnostic tools

- Queries cluster topology and indicates any port errors, link width, or link speed mismatch.
- Automates calls to many “low level” operations

## ■ Easy to use

- Similar flags, logs and reports for both tools
- Report using meaningful names when topology file is provided

# ibdiagnet - Optional flags

- **-i <dev-index> -p <port-num>**
  - Device index (0..N) and port number connected to the network
- **-o <out-dir>**
  - Directory to output the reports to
- **-lw <1x|4x|12x> -ls <2.5|5|10>**
  - Link speed and width checked on every port on the network
- **-pm -pc**
  - Perform error counters extensive check or clear counters respectively
- **-r**
  - Extensive additional checks performed.
- **-P**
  - Sets threshold for error levels. Also checks for errors of counters based on absolute value of the error counter. When not using –P flag, error thresholds are only triggered based on how many errors were incremented DURING the ibdiagnet run.
- **-c**
  - Packets to be sent on each link for error level checking
- **-h -V -v**
  - Help, Verbosity and Revision flags respectively

# Ibdiagnet usage

- Ibdiagnet is particularly useful in finding misconfigured links (speed/width, topology mismatches, and marginal link/cable issues).
- Typical usage:
  - Clear all port counters using ‘ibdiagnet –pc’
  - Stress the cluster
  - Check cluster using ‘ibdiagnet –lw 4x –ls 5 –P all=1’
    - Checks for link speed, link width, and port error counters greater than 1

```

root@mtlab32:~#
-----[REDACTED]-----
-I- PM Counters Info
-I-
-I- No illegal PM counters values were found
-----[REDACTED]
-I- Links With links width != 4x (as set by -lw option)
-I-
-I- No unmatched Links (with width != 4x) were found
-----[REDACTED]
-I- Links With links speed != 5 (as set by -ls option)
-I-
-I- No unmatched Links (with speed != 5) were found
-----[REDACTED]
-I- Fabric Partitions Report (see ibdiagnet.pkey for a full hosts list)
-I-
-I- PKey:0x7fff Hosts:2 full:2 partial:0
-----[REDACTED]
-I- IPoIB Subnets Check
-I-
-I- Subnet: IPv4 PKey:0x7fff QKey:0x000000bib MTU:2048Byte rate:10Gbps SL:0x00
-W- Suboptimal rate for group. Lowest member rate:20Gbps > group-rate:10Gbps
-----[REDACTED]
-I- Bad Links Info
-I- No bad link were found
-----[REDACTED]
-I- Stages Status Report:
STAGE           Errors  Warnings
Bad GUIDs/LIDs Check    0       0
Link State Active Check  0       0
Performance Counters Report 0       0
Specific Link Width Check 0       0
Specific Link Speed Check 0       0
Partitions Check        0       0
IPoIB Subnets Check     0       1
-----[REDACTED]
Please see /tmp/ibdiagnet.log for complete log
-----[REDACTED]
-I- Done. Run time was 1 seconds.
[root@mtlab32 ~]# 

```

# Cluster utilities - ibnetdiscover

- Reports a complete topology of cluster
- Shows all interconnect connections reporting:
  - Port LIDs
  - Port GUIDs
  - Host names
  - Link Speed
- GUID to name file can be used for more readable topology in regards to switch devices

# Cluster utilities - ibnetdiscover

- Simple usage is: `ibnetdiscover --node-name-map <guid to name file>`

```

root@mtilab32:~#
root@mtilab32:~#
root@mtilab32:~#
root@mtilab32:~#
root@mtilab32:~#
root@mtilab32:~#
root@mtilab32:~#
root@mtilab32:~# ibnetdiscover --node-name-map node_name_map
#
# Topology file: generated on Mon May 25 11:57:29 2009
#
# Max of 2 hops discovered
# Initiated from node 0002c90300000148 port 0002c90300000149

vendid=0x2c9
devid=0xbd36
sysimguid=0x2c9020040525b
switchguid=0x2c90200405258(2c90200405258)
Switch 36 "S-0002c90200405258"          # "SWITCH-1" enhanced port 0 lid 8 lmc 0
[18]   "H-0002c9030000057c"[1](2c9030000057d)      # "mtilab31 HCA-1" lid 17 4xDDR
[32]   "H-0002c90300000148"[1](2c90300000149)      # "mtilab32 HCA-1" lid 13 4xDDR

vendid=0x2c9
devid=0x634a
sysimguid=0x2c9030000057f
caguid=0x2c9030000057c
Ca      2 "H-0002c9030000057c"          # "mtilab31 HCA-1"
[1](2c9030000057d)      "S-0002c90200405258"[18]           # lid 17 lmc 0 "SWITCH-1" lid 8 4xDDR

vendid=0x2c9
devid=0x634a
sysimguid=0x2c9030000014b
caguid=0x2c90300000148
Ca      2 "H-0002c90300000148"          # "mtilab32 HCA-1"
[1](2c90300000149)      "S-0002c90200405258"[32]           # lid 13 lmc 0 "SWITCH-1" lid 8 4xDDR
[root@mtilab32:~#
[root@mtilab32:~#
[root@mtilab32:~#
[root@mtilab32:~#

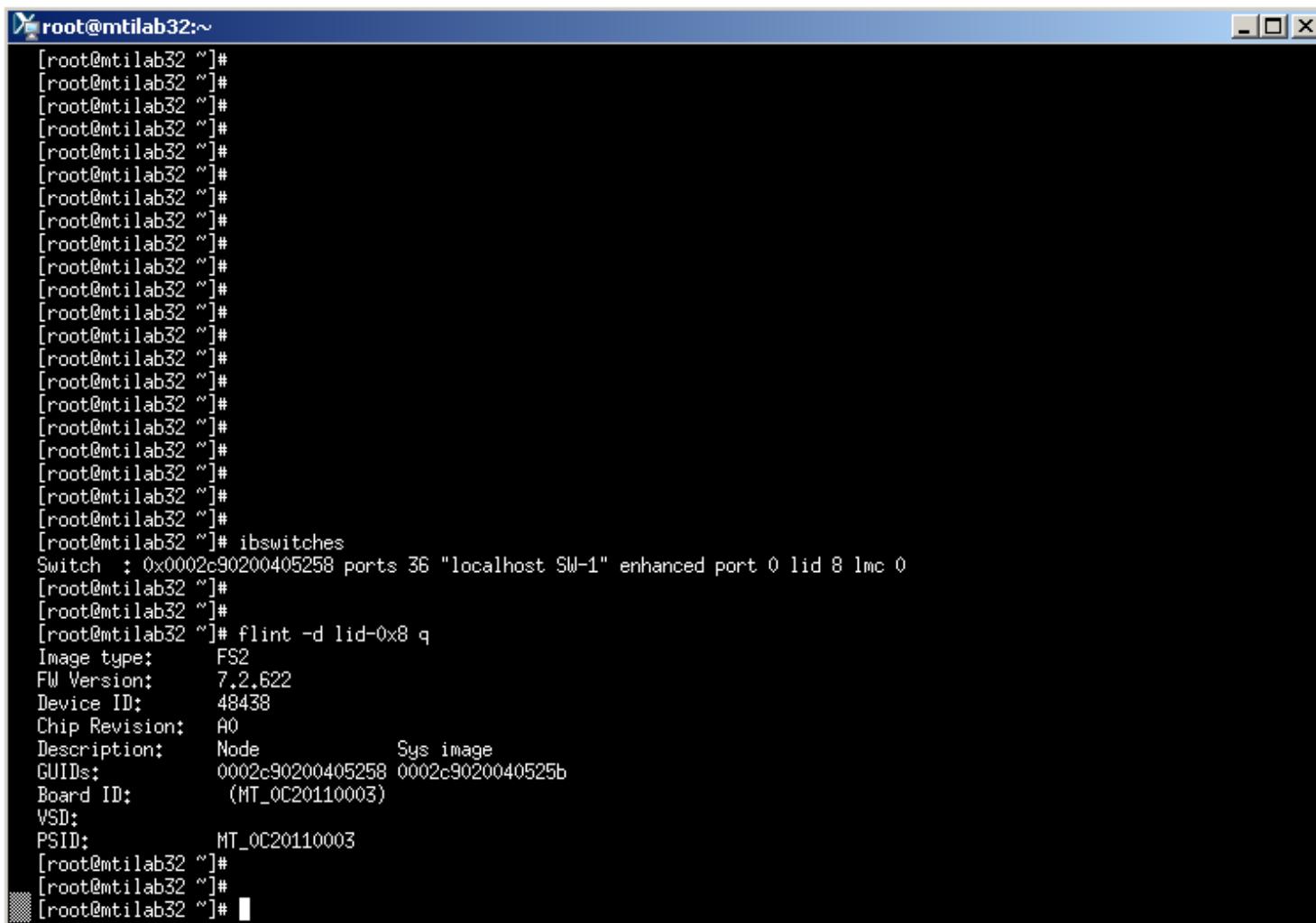
```

# Error counter review

- **SymbolErrors**
  - **Total number of minor link errors.** Usually an 8b/10b error due to a bit error
- **Link Recovers**
  - **Total number of times the Port Training state machine has successfully completed the link error recovery process.**
- **LinkDowned**
  - **Total number of times the Port Training state machine has failed the link error recovery process and downed the link.**
- **RcvErrors**
  - **Total number of packets containing an error that were receive on the port.** Usually due to a CRC error caused by a bit error within the packet.
- **RcvSwRelayErrors**
  - **Total number of packets received on the port that were discarded because they could not be forwarded by the switch relay.** This counter should typically be ignored since Anafa-II has a bug that counts these when it gets a multicast packet on a port where that port also belongs to the multicast group of the packet.
- **XmtDiscards**
  - **Total number of outbound packets discarded by the port because the port is down or congested.** Usually due to the output port HQQ lifetime being exceeded.
- **VL15Dropped**
  - **Number of incoming VL15 packets dropped due to resource limitations (e.g., lack of buffers) in the port**
- **XmtData,RcvData**
  - **Total number of 32-bit data words transmitted and received on the port.**
- **XmtPkts,RcvPkts**
  - **Total number of data packets transmitted and received on the port.**

# Switch firmware update example

- Determine LID using ibswitches, or ibnetdiscover.



```
[root@mtilab32 ~]# 
[root@mtilab32 ~]# ibswitches
Switch : 0x0002c90200405258 ports 36 "localhost SW-1" enhanced port 0 lid 8 lmc 0
[root@mtilab32 ~]#
[root@mtilab32 ~]#
[root@mtilab32 ~]# flint -d lid=0x8 q
Image type: FS2
FW Version: 7.2.622
Device ID: 48438
Chip Revision: A0
Description: Node      Sys image
GUIDs: 0002c90200405258 0002c9020040525b
Board ID: (MT_0C20110003)
VSD:
PSID: MT_0C20110003
[root@mtilab32 ~]#
[root@mtilab32 ~]#
[root@mtilab32 ~]#
```

# Alternate Switch update example

- All Infiniband devices mapped into /dev/mst space using 'mst ib add'
- Devices can be updated using proper /dev/mst device (shown using 'mst status'). Can also be used to update HCA devices.

```

root@mtlab32:~#
Connection to mtilab33 closed.
[root@mtlab32 ~]#
[root@mtlab32 ~]#
[root@mtlab32 ~]#
[root@mtlab32 ~]#
[root@mtlab32 ~]# mst status
MST modules:
-----
      MST PCI module loaded
      MST PCI configuration module loaded
      MST Calibre (I2C) module is not loaded

MST devices:
-----
/dev/mst/mt25418_pciconf0      - PCI configuration cycles access.
                                bus:dev.fn=04:00.0 addr.reg=88 data.reg=92
                                Chip revision is: A0
/dev/mst/mt25418_pci_cr0      - PCI direct access.
                                bus:dev.fn=04:00.0 bar=0xfcfc00000 size=0x100000
                                Chip revision is: A0
/dev/mst/mt25418_pci_msix0    - PCI direct access.
                                bus:dev.fn=04:00.0 bar=0x00000000 size=0x0
/dev/mst/mt25418_pci_uar0      - PCI direct access.
                                bus:dev.fn=04:00.0 bar=0xfa000000 size=0x2000000

Inband devices:
-----
/dev/mst/CA_MT25418_mtilab31_HCA-1_lid-0x0011
/dev/mst/CA_MT25418_mtilab32_HCA-1_lid-0x000D
/dev/mst/SW_MT48438_lid-0x0008
[root@mtlab32 ~]#
[root@mtlab32 ~]#
[root@mtlab32 ~]# flint -d /dev/mst/SW_MT48438_lid-0x0008 q
Image type: FS2
FW Version: 7.2.622
Device ID: 48438
Chip Revision: A0
Description: Node          Sys image
GUIDs:      0002c90200405258 0002c9020040525b
Board ID:   (MT_0C20110003)
VSD:
PSID:      MT_0C20110003
[root@mtlab32 ~]#

```

# IPoIB

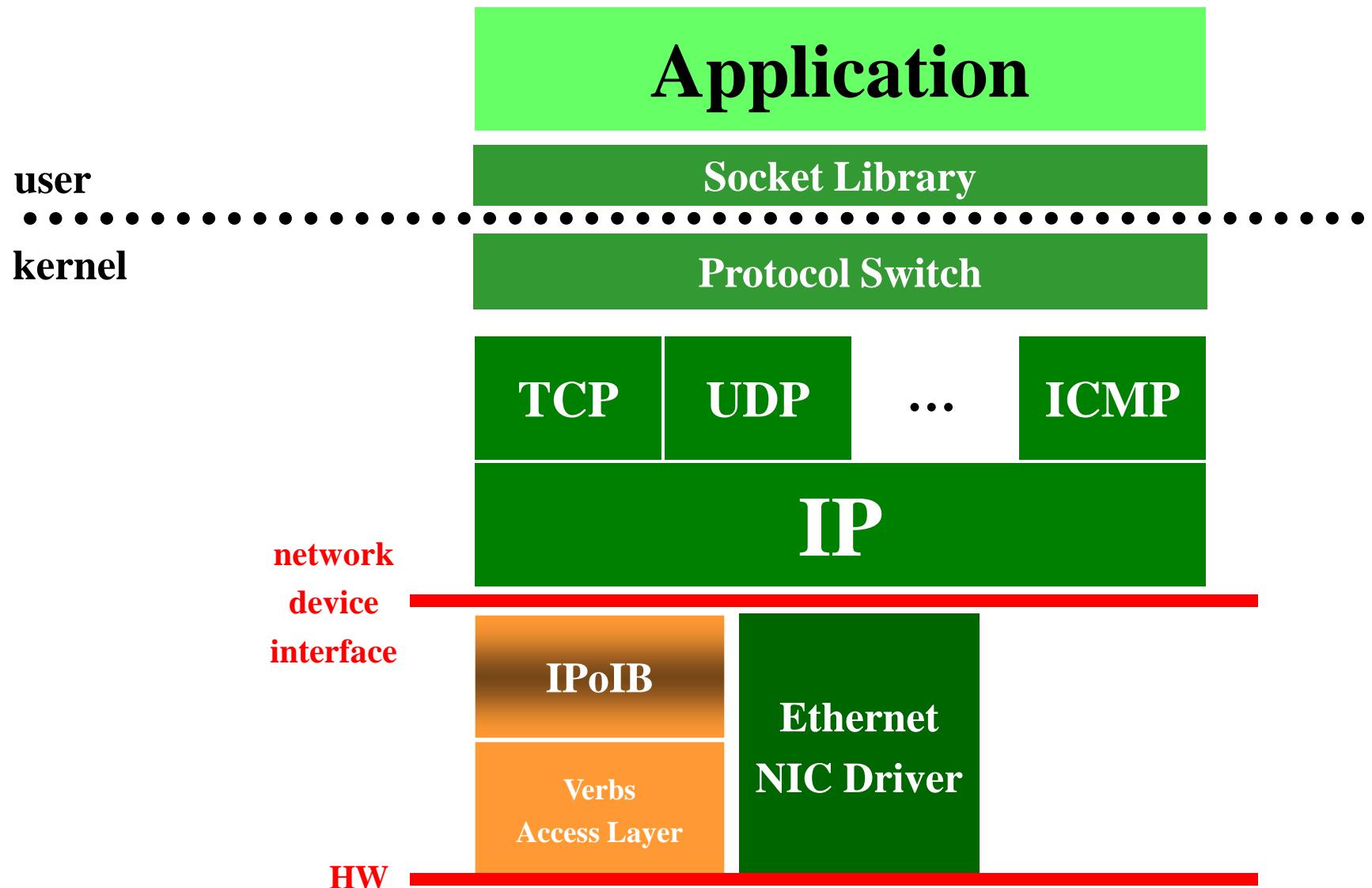


CONFIDENTIAL

# IPoIB in a Nut Shell

- Encapsulation of IP packets over IB
- Uses IB as “layer two” for IP
  - Supports both UD service (up to 2KB MTU) and RC service (connected mode, up to 64KB MTU).
- IPv4, IPv6, ARP and DHCP support
- Multicast support
- VLANs support
- Benefits:
  - Transparency to the legacy applications
  - Allows leveraging of existing management infrastructure
- Specification state: IETF Draft

# IPoIB in Generic Protocol Stack



# IPoIB Building Blocks

- Two modes: UD or CM (`/sys/class/net/ib*/mode`)
  - UD uses UD QP
    - Unreliable
    - Each destination described using AV
    - IPoIB MTU constrained by IB MTU
  - CM uses RC QP
    - Allows for large MTU
    - Better performance
- Destination is described by:
  - GID of destination port
  - Destination QP
  - GID + QP used as MAC address
- Uses multicast tree for address resolution
- Uses SA to get path record for node

# ARP.. How it works over Ethernet

- Assuming “IPx”=querying host IP, “IPy”=target host IP
- “IPx” send broadcast query (ARP) content:
  - I’m “IPx” and want to know who is IP=“IPy”
- All receiving nodes will compare “IPy” to their node IP
- If node IP matches “IPy” then:
  - Send unicast message to IPx saying I’m “IPy” and my MAC address is “MACy”
  - Node with IPy will also cache MAC of IPx already embedded in query
- Node IPx will store MACy address of IPy
- Next time node IPx needs to send packet it will use MACy

# ARP .... How is it works with IPoIB

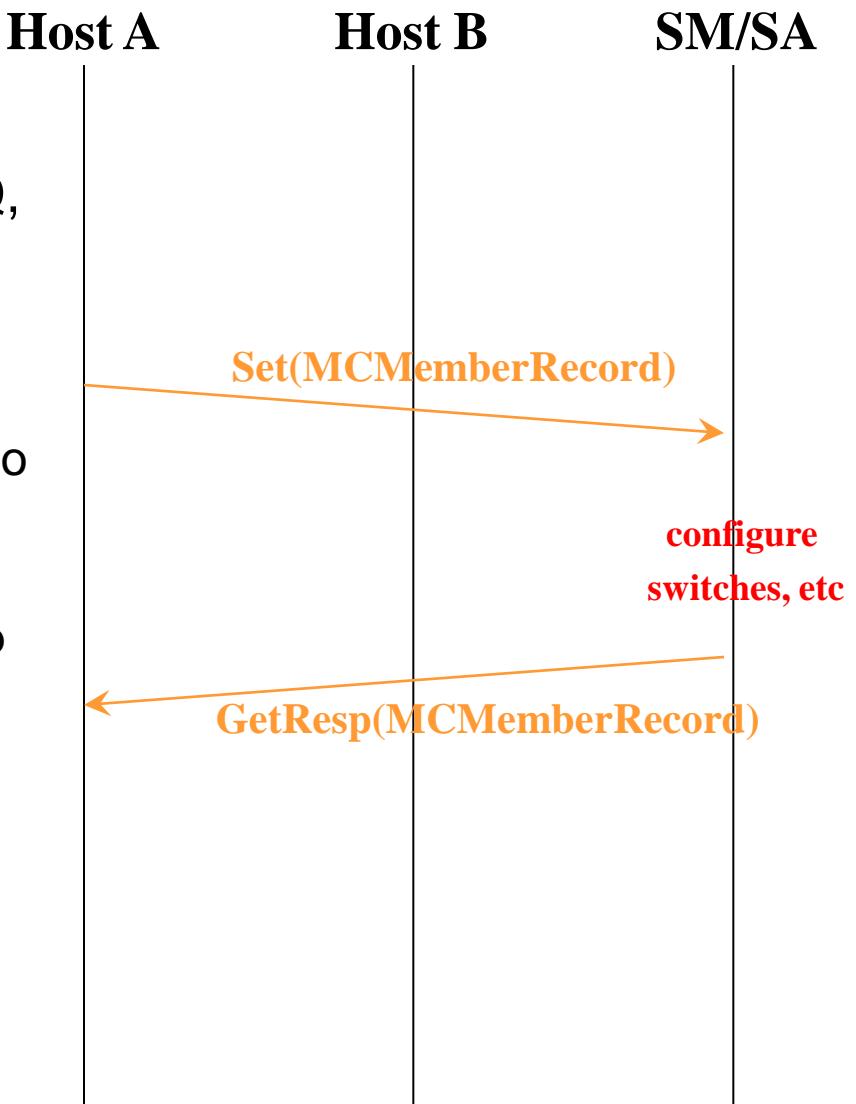
- “IPx” send multicast query (ARP) content:
  - I’m “IPx” and want to know who is IP=“IPy”
- All receiving nodes will compare “IPy” to their node IP
- If node IP matches “IPy” then:
  - Send unicast message to IPx saying I’m “IPy” and my MAC (QP+GID) address is “MACy”
  - Node with IPy will also cache MAC of IPx already embedded in query
- Node IPx will store MACy address of IPy
- Next time node IPx needs to send packet it will use MACy
- but.....

- IPoIB MAC is not routable
- LID is needed to send IPoIB packet to destination node
- Querying node needs to retrieve LID for MACy
- So.....:
  - Once arp reply is received
  - Sends SA query for port GID
  - Until SA query is replied queue outgoing packet
  - Once SA query response is received send queued packets to remote node
  - Cache SA entry for future use

# IPoIB – A Day in a Life

## ■ Setup

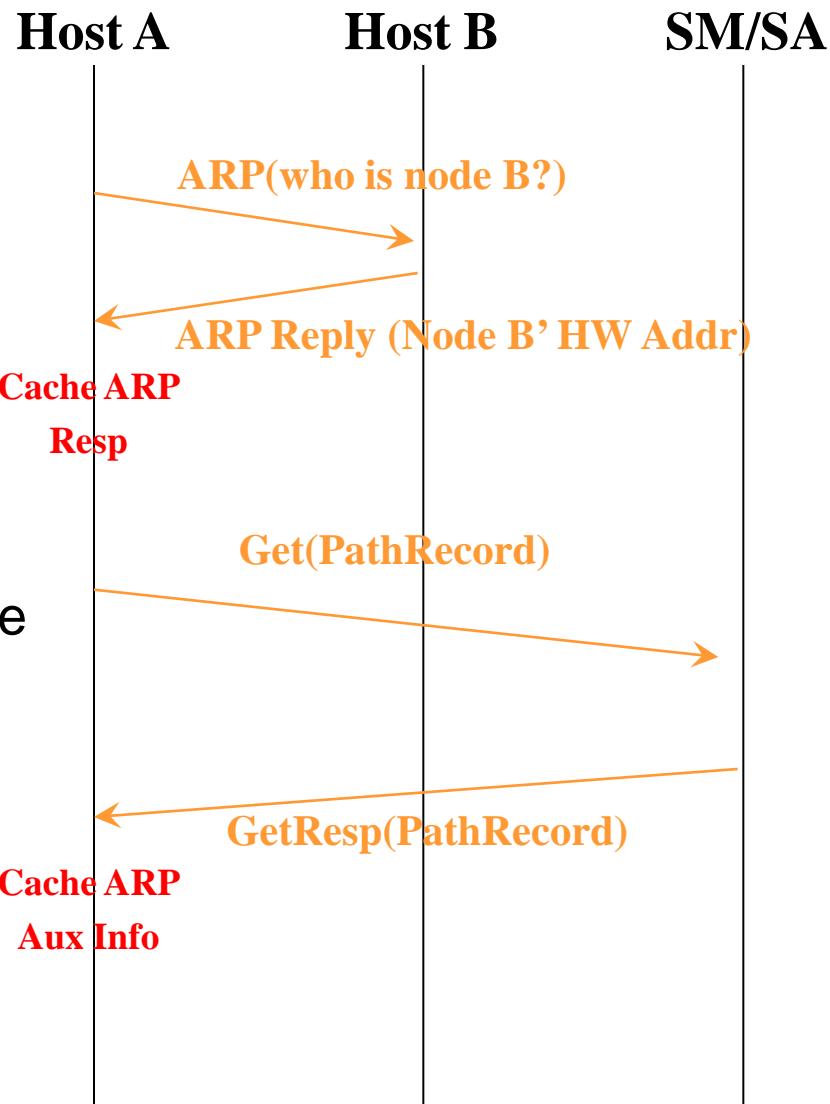
- Creation of IB resources: QP, CQ, PD, etc
- Setup of broadcast group
  - Send a Set(MCMemberRecord) to the SA to join/create the group
  - Wait for acknowledgement
  - Attach IPoIB QP to the MC group



# IPoIB – A Day in a Life

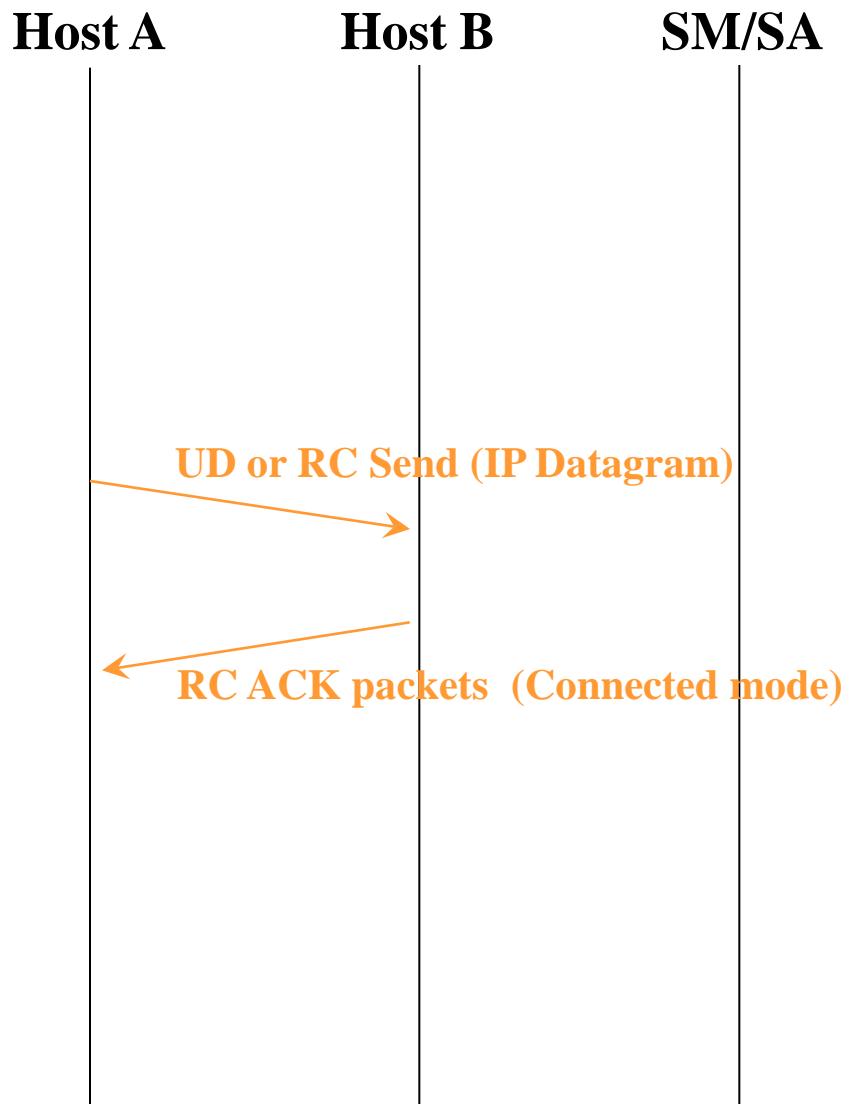
## ■ Address resolution

- Send ARP packet (broadcast)
- Get ARP reply
- Query SA with PathRecord
- Get PathRecord
- Create and cache AddressHandle (performance)



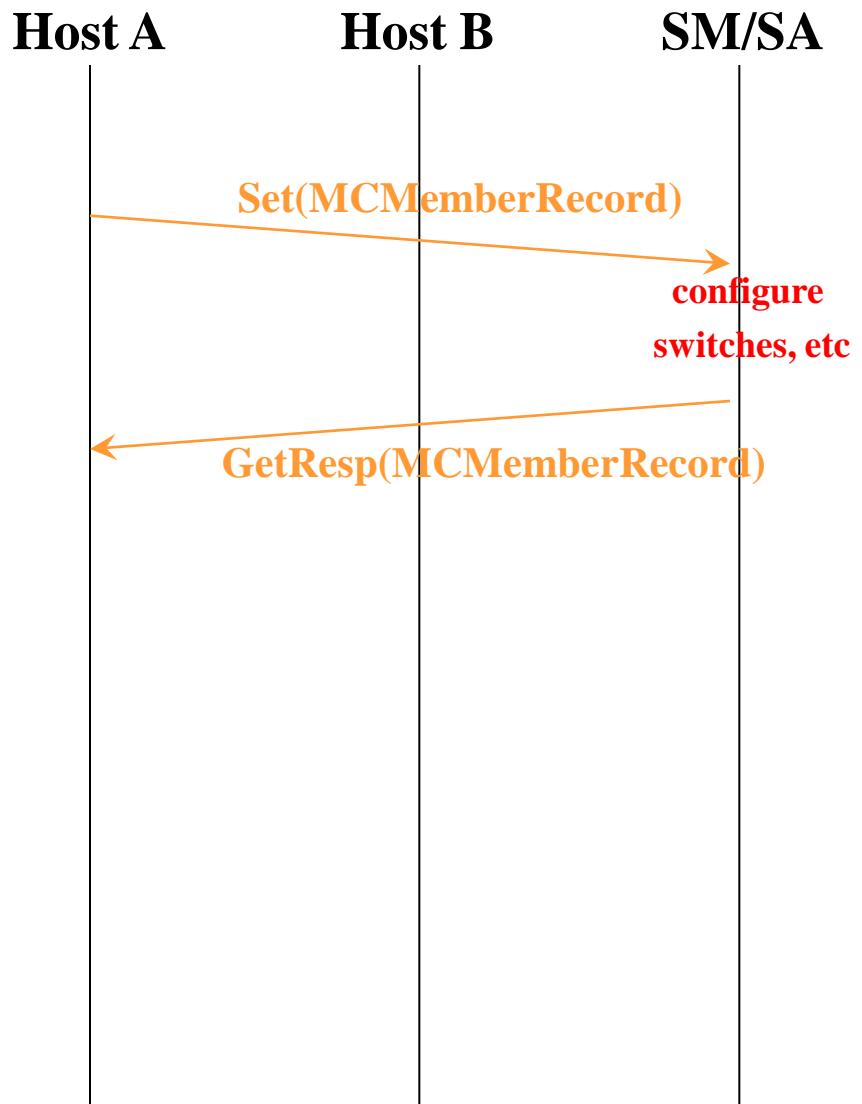
## ■ Data flow

- Add IPoIB encapsulation header
- Post send WR to the QP



## ■ Teardown

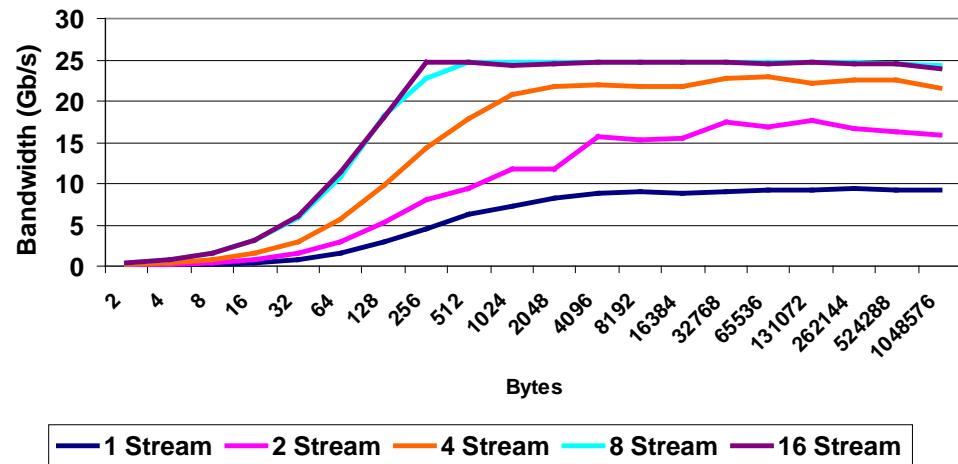
- Unregister from MC and Broadcast groups
- Cleanup of IB resources



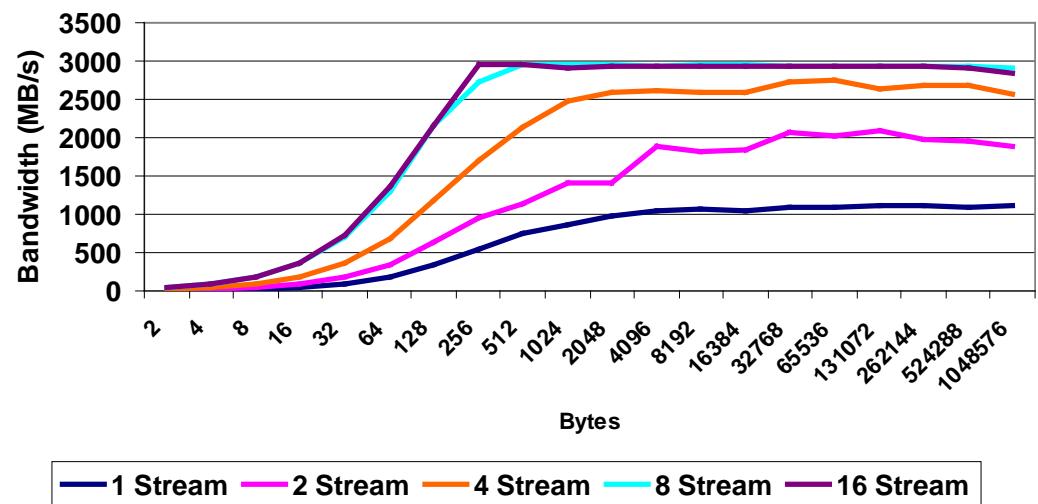
# IPoIB-CM ConnectX Performance - IB QDR PCIe Gen2



IPoIB-CM ConnectX IB QDR PCIe Gen2



IPoIB-CM ConnectX IB QDR PCIe Gen2



# IPoIB Mode Settings

## ■ IPoIB runs in two modes

- Datagram mode using UD transport type
- Connected mode using RC transport type

## ■ Default mode is Connected Mode

- This can be changed by editing /etc/infiniband/openib.conf and setting 'SET\_IPOIB\_CM=no'.
- After changing the mode, you need to restart the driver by running:
  - **/etc/init.d/openibd restart**
- To check the current mode used for out-going connections, enter:
  - **cat /sys/class/net/ib<n>/mode**

# IPoIB Configuration

- Requires assigning an IP address and a subnet mask to each HCA port (like any other network adapter)
- The first port on the first HCA in the host is called interface ib0, the second port is called ib1, and so on.
- Configuration can be based on DHCP or on a static configuration
  - Modify /etc/sysconfig/network-scripts/ifcfg-ib0:

```
DEVICE=ib0
BOOTPROTO=static
IPADDR=10.10.0.1
NETMASK=255.255.255.0
NETWORK=10.10.0.0
BROADCAST=10.10.0.255
ONBOOT=yes
```
  - ifconfig ib0 10.10.0.1 up

# MPI



CONFIDENTIAL

- A message passing interface
- Used for point to point communication
  - MPI\_ISEND, MPI\_IRecv
- Used for collective operations:
  - MPI\_AlltoAll, MPI\_Reduce, MPI\_barrier
- Other primitives
  - MPI\_Wait, MPI\_Walltime
- MPI Ranks are IDs assigned to each process
- MPI Communication Groups are subdivisions a job node used for collectives
- Three MPI stacks are included in this release of OFED:
  - MVAPICH 1.1.0
  - Open MPI 1.2.8
- This presentation will concentrate on MVAPICH-1.1.0

# MPI Example

```
01: MPI_Init(&argc,&argv);
02: MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
03: MPI_Comm_rank(MPI_COMM_WORLD,&myid);
04:
05: MPI_Barrier(MPI_COMM_WORLD);
06:
07: if(myid==0)
08:   printf("Passed first barrier\n");
09:
10: srand(myid*1234);
11: x = rand();
12:
13: printf("I'm rank %d and my x is 0x%08x\n",myid, x);
14:
15: MPI_Barrier(MPI_COMM_WORLD);
16:
17: MPI_Bcast(&x,1,MPI_INT,0,MPI_COMM_WORLD);
18:
19: if(myid == 1)
20:   printf("My id is rank 1 and I got 0x%08x from rank 0\n", x);
21:
22: if(myid == 2)
23:   printf("My id is rank 2 and I got 0x%08x from rank 1\n", x);
24:
25: MPI_Finalize();
```

# Compiling

- mpicc is used to compiling mpi applications
- mpicc is equivalent to gcc
- mpicc includes all the gcc flags needed for compilation
  - Head files paths
  - Libraries paths
- To see real compilation flag run: mpicc -v
- MPI application can be shared or dynamic

# Launching MPI jobs using mpirun\_rsh

## ■ Prerequisites for Running MPI:

- The mpirun\_rsh launcher program requires automatic login (i.e., password-less) onto the remote machines.
- Must also have an /etc/hosts file to specify the IP addresses of all machines that MPI jobs will run on.
- Make sure there is no loopback node specified (i.e. 127.0.0.1) in the /etc/hosts file or jobs may not launch properly.
- Details on this procedure can be found in Mellanox OFED User's manual

## ■ Basic format:

- `mpirun_rsh -np procs node1 node2 node3 BINARY`

## ■ Other flags:

-show: show only

-paramfile: environment variables

-hostfile: list of host

-ENV=VAL (i.e. VIADEV\_RENDEZVOUS\_THRESHOLD=8000)

# Launching MPI jobs using mpirun\_rsh (cont...)

- `mpirun_rsh -show -np 3 mtilab32 mtilab33 mtilab33 ./dcest:`
- command: `/usr/bin/ssh mtilab32 cd /home/rabin/tmp; /usr/bin/env MPIRUN_MPDU=0 MPIRUN_HOST=mtilab32.mti.mtl.com MPIRUN_PORT=33111 MPIRUN_PROCESSES='mtilab32:mtilab33:mtilab33:' MPIRUN_RANK=0 MPIRUN_NPROCS=3 MPIRUN_ID=26974 DISPLAY=localhost:12.0 ./dcest`
- command: `/usr/bin/ssh mtilab33 cd /home/rabin/tmp; /usr/bin/env MPIRUN_MPDU=0 MPIRUN_HOST=mtilab32.mti.mtl.com MPIRUN_PORT=33111 MPIRUN_PROCESSES='mtilab32:mtilab33:mtilab33:' MPIRUN_RANK=1 MPIRUN_NPROCS=3 MPIRUN_ID=26974 DISPLAY=localhost:12.0 ./dcest`
- command: `/usr/bin/ssh mtilab33 cd /home/rabin/tmp; /usr/bin/env MPIRUN_MPDU=0 MPIRUN_HOST=mtilab32.mti.mtl.com MPIRUN_PORT=33111 MPIRUN_PROCESSES='mtilab32:mtilab33:mtilab33:' MPIRUN_RANK=2 MPIRUN_NPROCS=3 MPIRUN_ID=26974 DISPLAY=localhost:12.0 ./dcest`

# MVAPICH Internals

- The basic data transfer unit is a vbuf
- vbuf are generally used for small messages ~<12k (configurable)
- A vbuf always requires a memory copy from user buffer to the mvapich layer and vice versa
- vbufs are also used internally
  - Use for implementation to implementation info
  - E.g RDMA addresses
- vbufs are transferred between node using:
  - Fast RDMA Path
  - Eager Mode (Send/Recv)

# Fast RDMA Path

- Fastest way (lowest latency) for transfer of small messages (vbufs)
- Optimized for latency
  - Doesn't require completion
  - Based on RDMA Write
  - Doesn't require synchronization
  - If message is small then post inline is used
- Algorithm:
  - Each connection size has two arrays of vbufs (virtually contiguous)
    - Send Array
    - Receive Array
  - When small message is sent and vbuf is available from array, data is copied from user buffer to vbuf entry in array.
  - RDMA write is sent to remote node vbuf array
  - Remote node constantly polls vbuf receive array
  - If new vbuf is user buffer data is copied to user buffer
  - Progress engine sends credits of array to remote side
    - Piggybacked to other vbuf transfers
    - Using dedicated vbufs
- Environment Variables:
  - Number of vbufs in array per connection is controlled: VIADEV\_NUM\_RDMA\_BUFFER
  - Size of each vbuf: VBUF\_TOTAL\_SIZE

# Eager Mode

- Simple send/receive buffers
- Used for vbuf transfers
- Used once vbufs are exhausted
- WQE will point to vbuf buffers
  - Different vbuf pool than fast path
- Eager mode is transparent to user

# Rendezvous mode (zero copy)

- Used for large messages
- Used when certain threshold is reached
  - Control through VIADEV\_RENDEZVOUS\_THRESHOLD
- Zero copy transfers
- Uses vbuf for flow control transfers
  - Used to send rdma address of user space buffers
  - Used to send completions of transfers
- User buffer registration
  - User buffers are registered on demand
  - User buffers are not deregistered but placed in cache
  - If user reuses buffer for new transfer region is reused
  - If user buffer freed buffer is not de-registered
  - OS will not free buffer if user calls “free”
    - Pages still registered in driver
    - This is called lazy de-registration
    - Only when lazy de-registration is called buffers will be freed

# Cheat Sheet

- All binaries are under **MPIHOME/bin**
  - Default /usr/mpi/gcc/mvapich-1.1.0/bin/
- **mpirun\_rsh -np num\_proc node1 node2 ... BINARY PARAMS**
  - debug: open gdb (need display set)
  - show: show what mpi does
  - hostfile: node list
- **mpicc -v:** shows commands
- Environment Variables:
  - VIADEV\_DEVICE=device name (def=InfiniHost0)
  - VIADEV\_DEFAULT\_MTU=mtu size (def=1024)
  - VIADEV\_DEFAULT\_SERVICE\_LEVEL=sl to use in QP
  - VIADEV\_DEFAULT\_TIME\_OUT=QP timeout
  - VIADEV\_DEFAULT\_RETRY\_COUNT=RC retry count
  - VIADEV\_NUM\_RDMA\_BUFFER=fast path array size (def=32 0=disabled)

# SDP – Sockets Direct Protocol

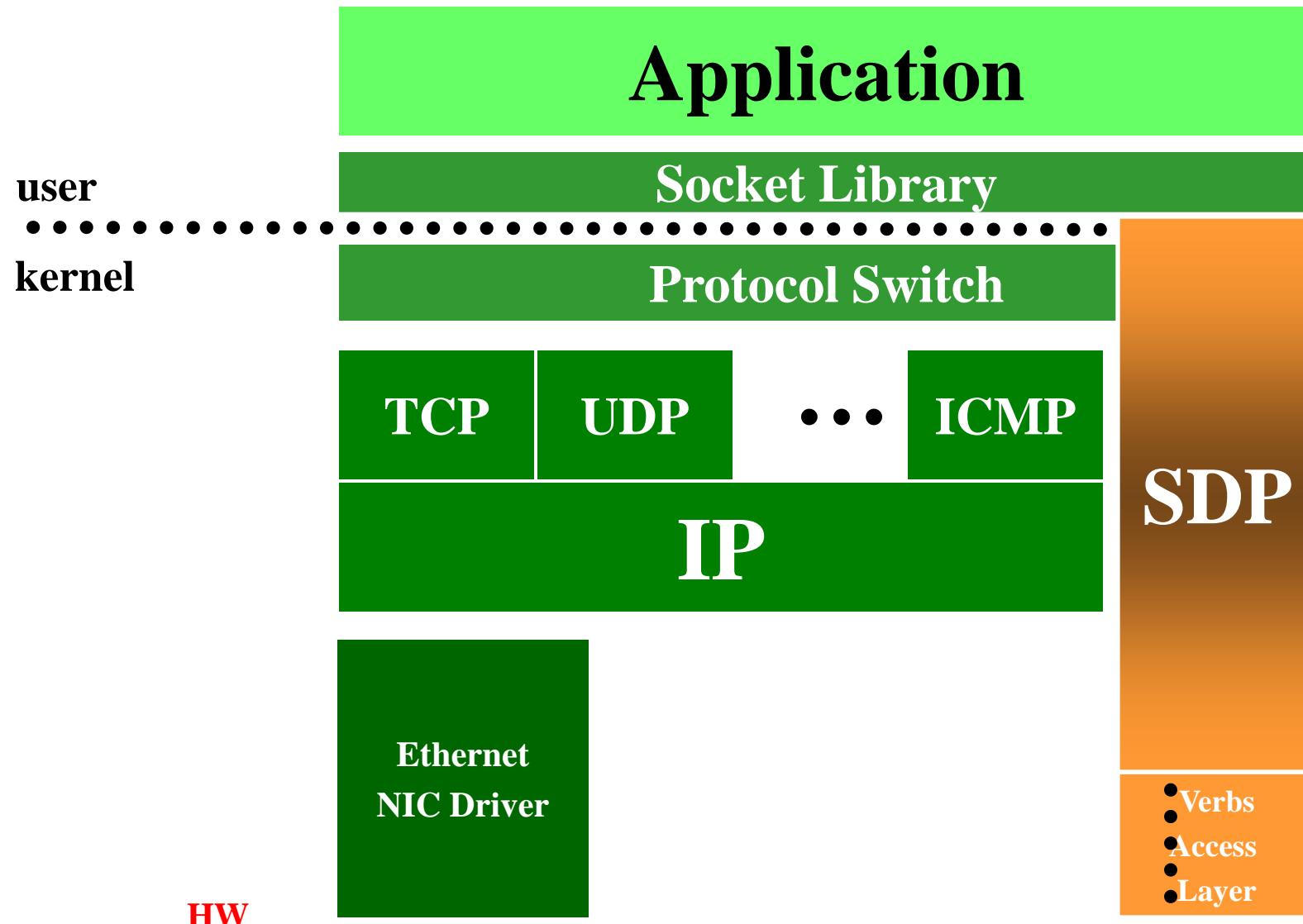


CONFIDENTIAL

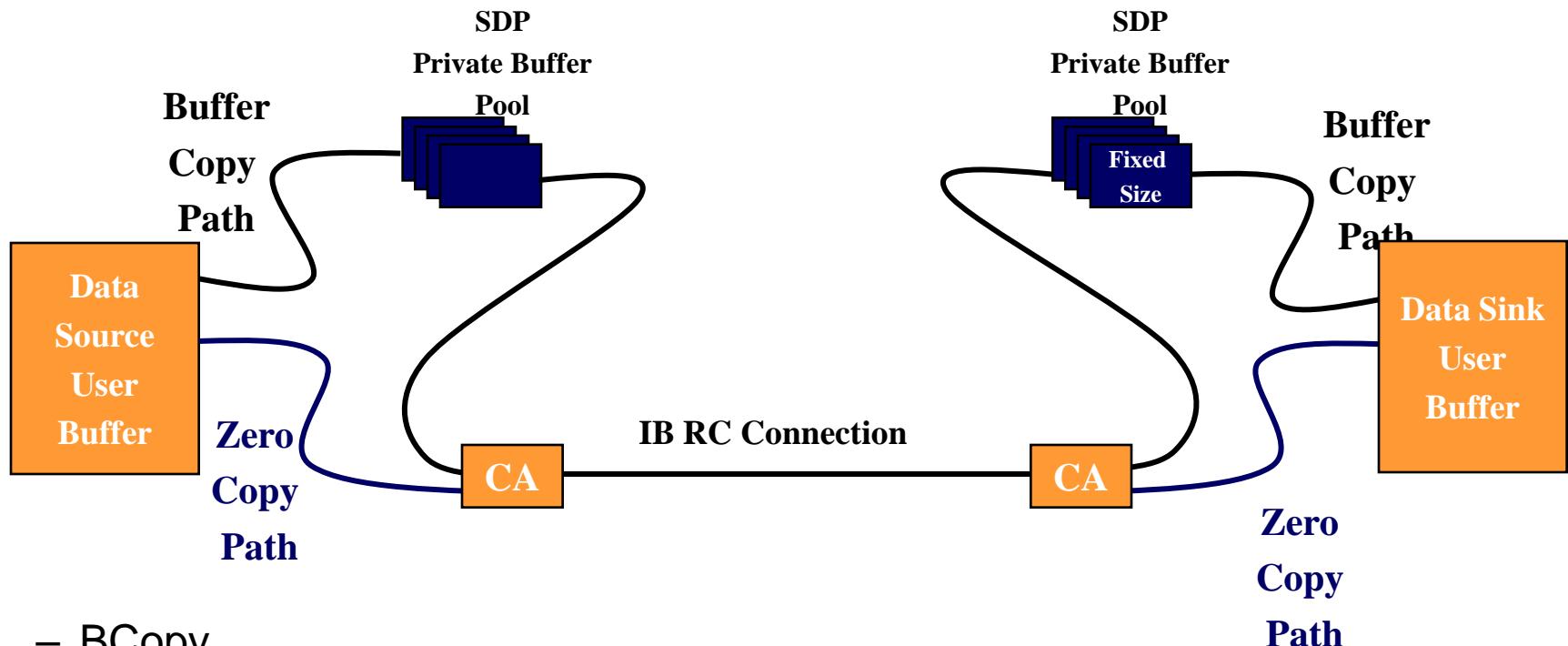
# SDP in a Nut Shell

- An InfiniBand byte-stream transport protocol that provides TCP stream semantics.
- Capable of utilizing InfiniBand's advanced protocol offload capabilities, SDP can provide lower latency, higher bandwidth, and lower CPU utilization than IPoIB running some sockets-based applications.
- Composed of a kernel module that implements the SDP as a new address-family/protocol-family, and a library that is used for replacing the TCP address family with SDP according to a policy.

# SDP in Generic Protocol Stack (User)



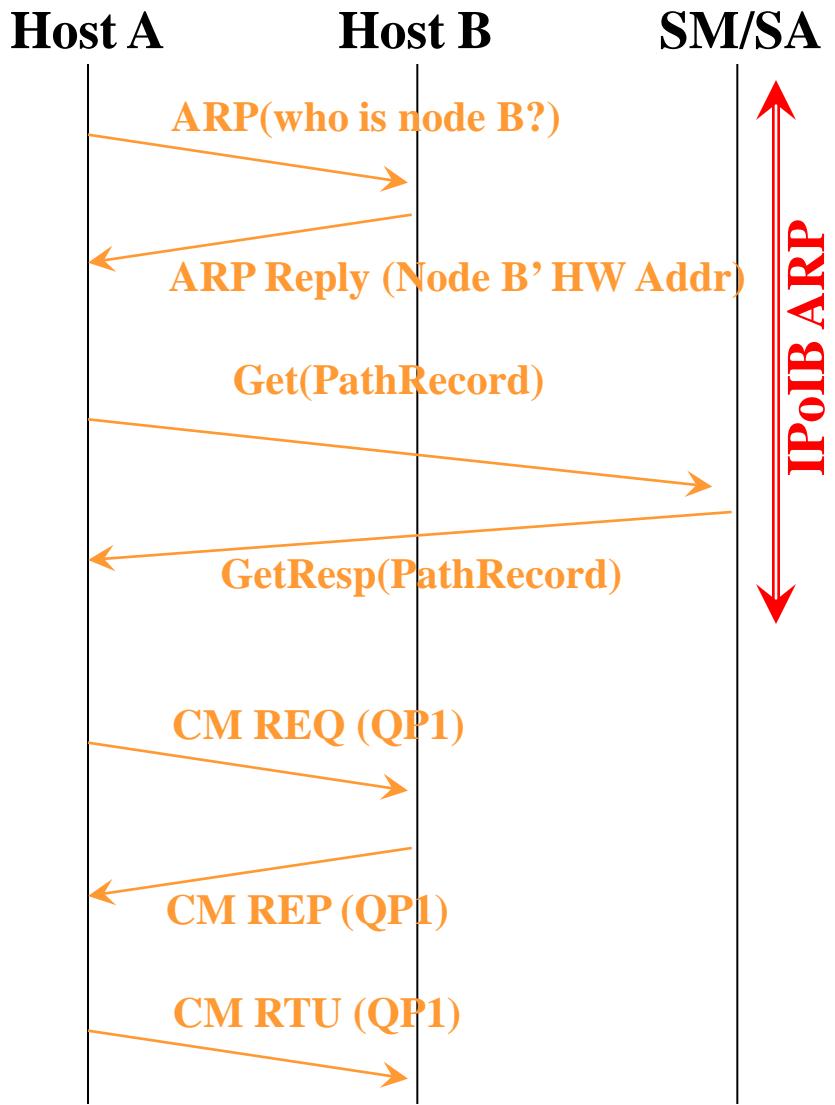
# SDP Buffering Model



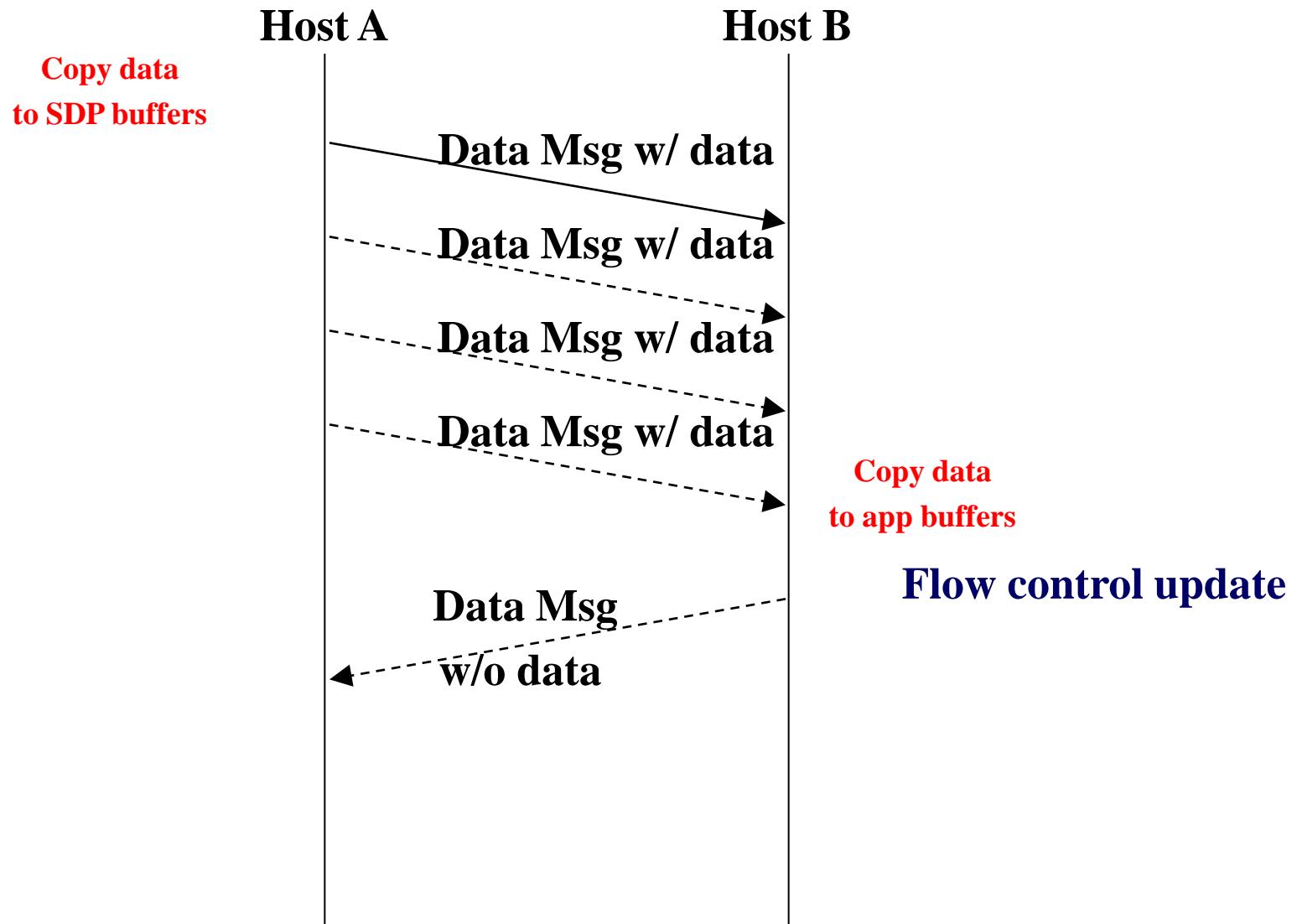
- BCopy
  - Short Transfer
  - Application needs buffering (e.g. async)
- ZCopy
  - Large data buffers
- BZCopy
  - Uses Zero copy path on Transmit side

# Connection Setup

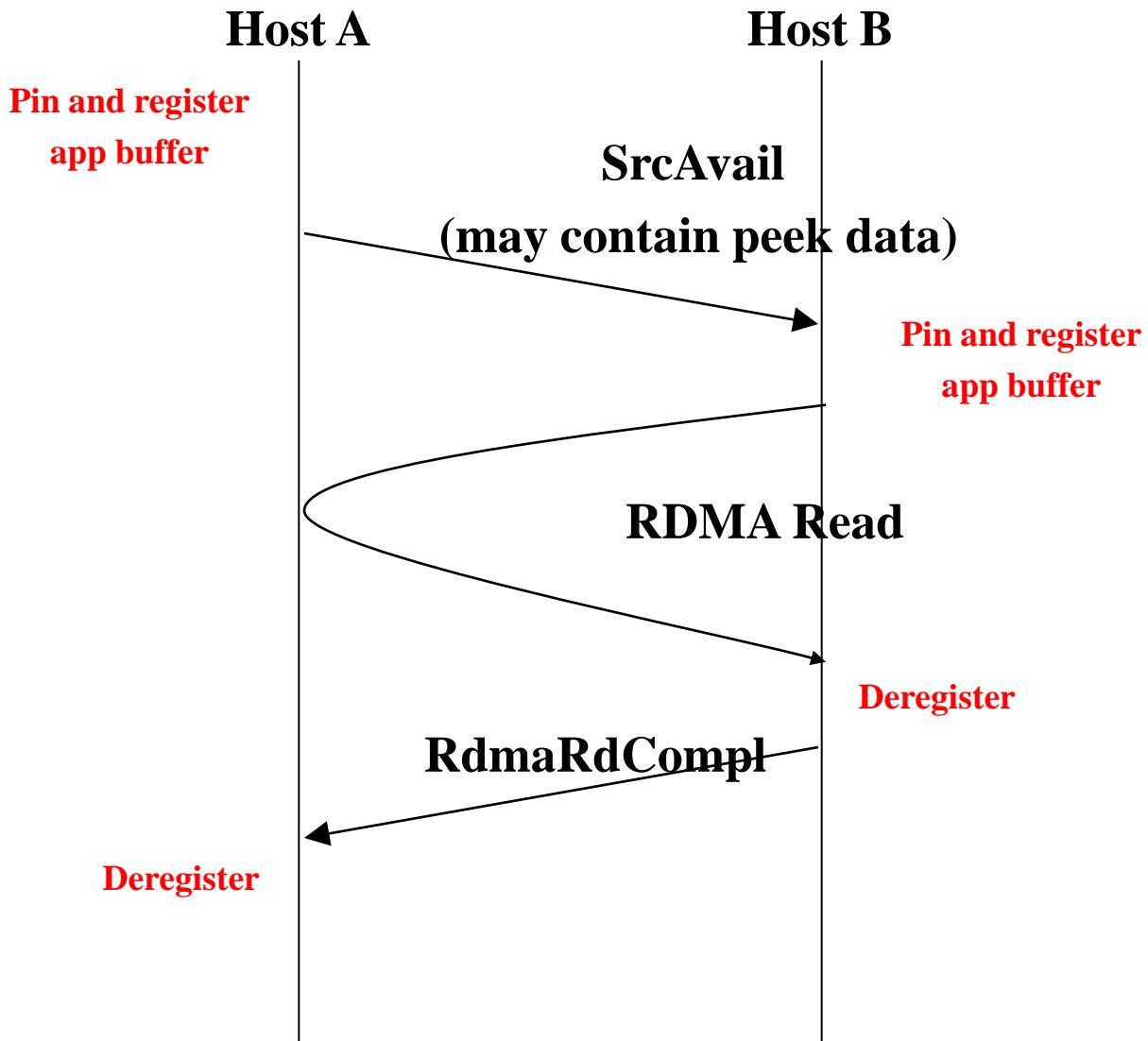
- Address resolution
  - Send ARP packet (broadcast)
  - Get ARP reply
  
- Query SA with PathRecord
- Get PathRecord
  
- CM Connect (3 way handshake)
  - Send REQ with Hello message in private data
  - Receive REP with HelloACK
  - Send RTU



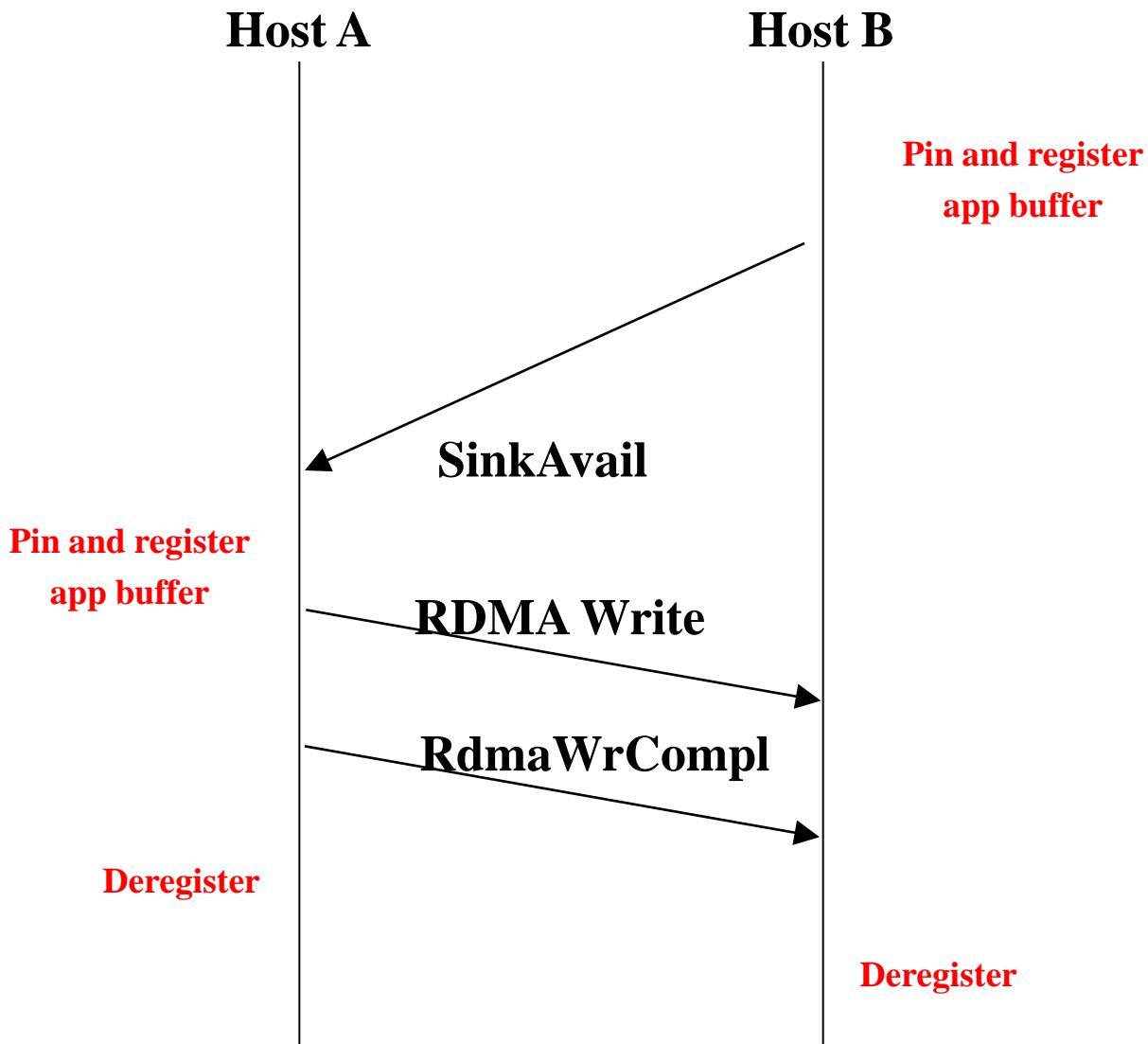
# BCopy Data Transfer



# Read ZCopy Data Transfer

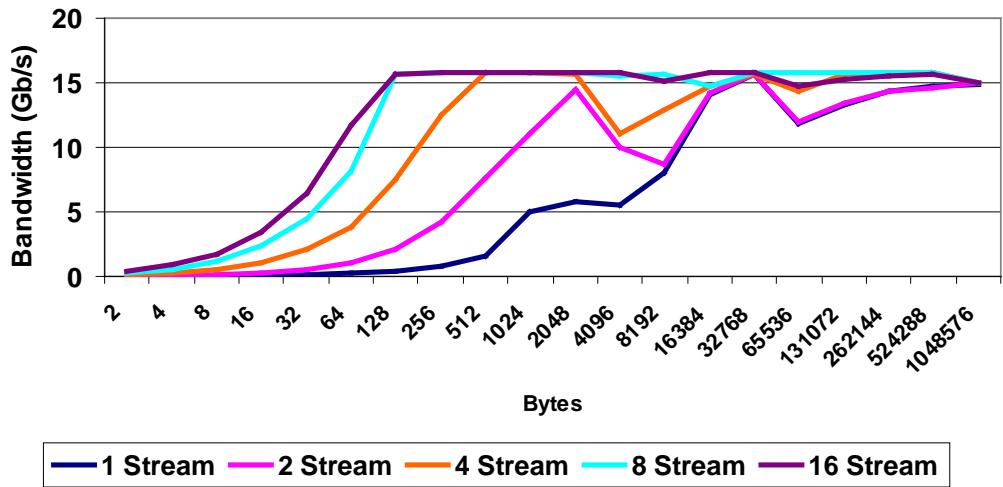


# Write ZCopy Data Transfer

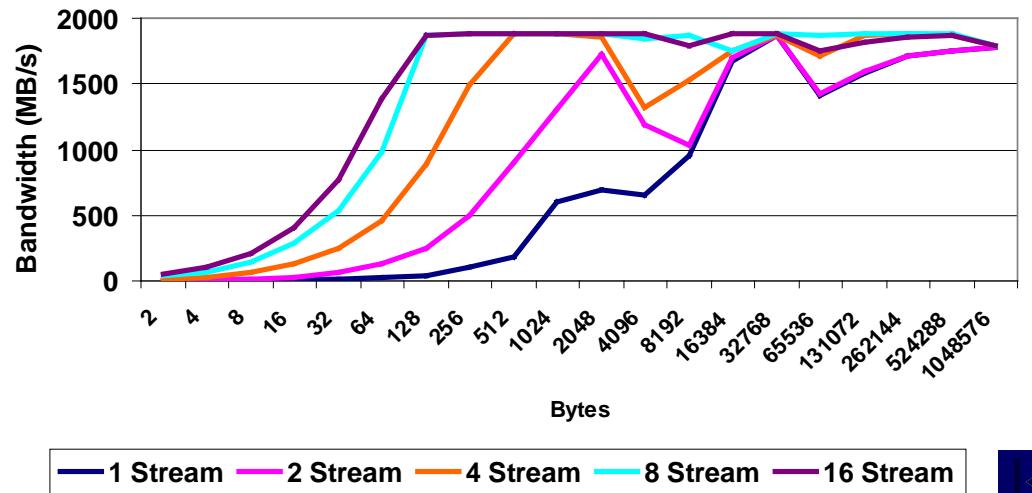


# SDP BCOPY IB DDR PCIe Gen2

SDP Bcopy ConnectX IB DDR PCIe Gen2



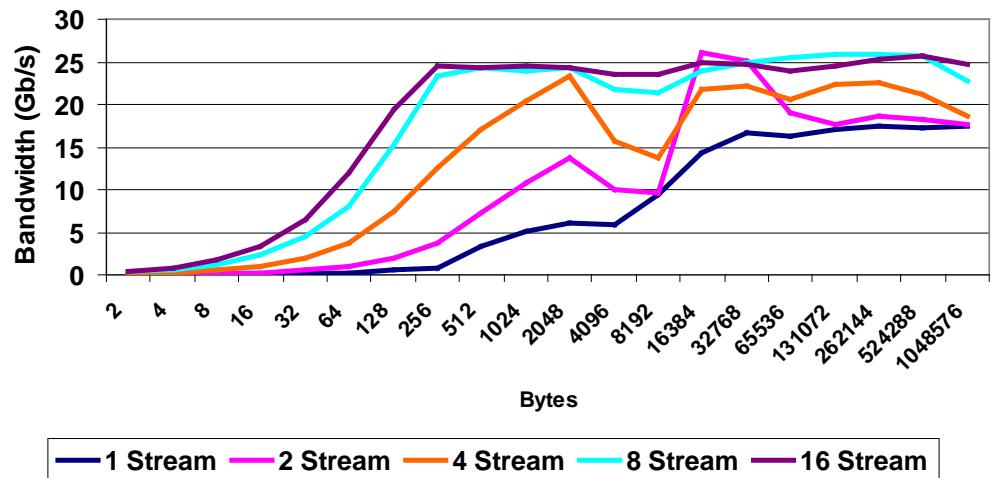
SDP Bcopy ConnectX IB DDR PCIe Gen2



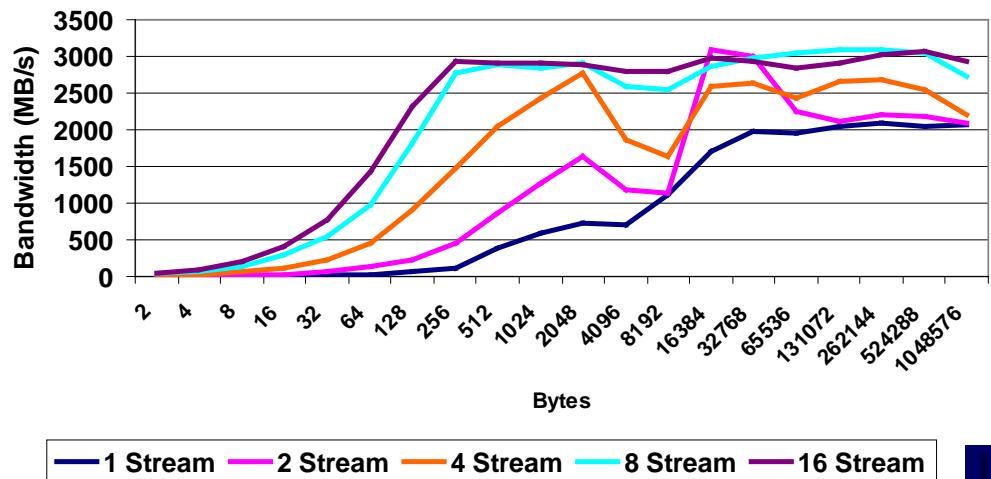
# SDP BCOPY IB QDR PCIe Gen2



SDP Bcopy ConnectX IB QDR PCIe Gen2

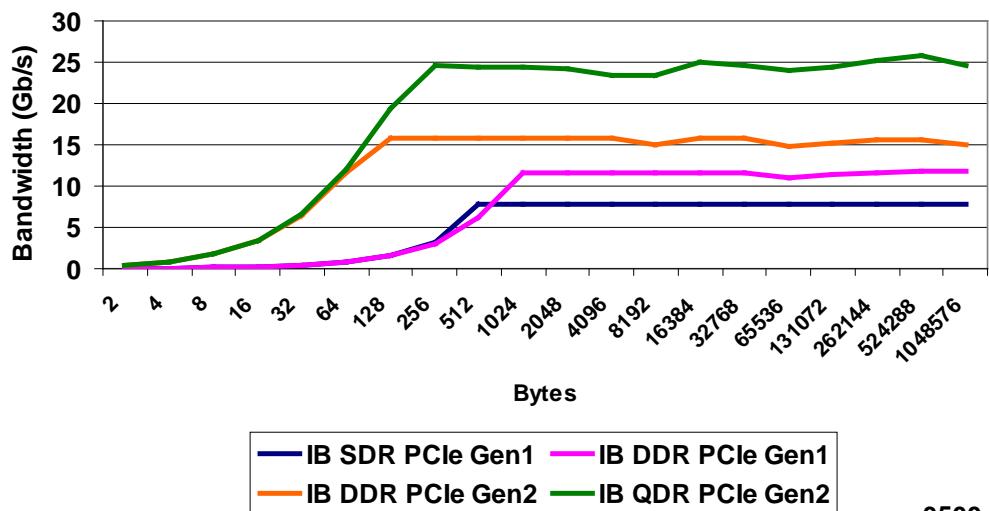


SDP Bcopy ConnectX IB QDR PCIe Gen2

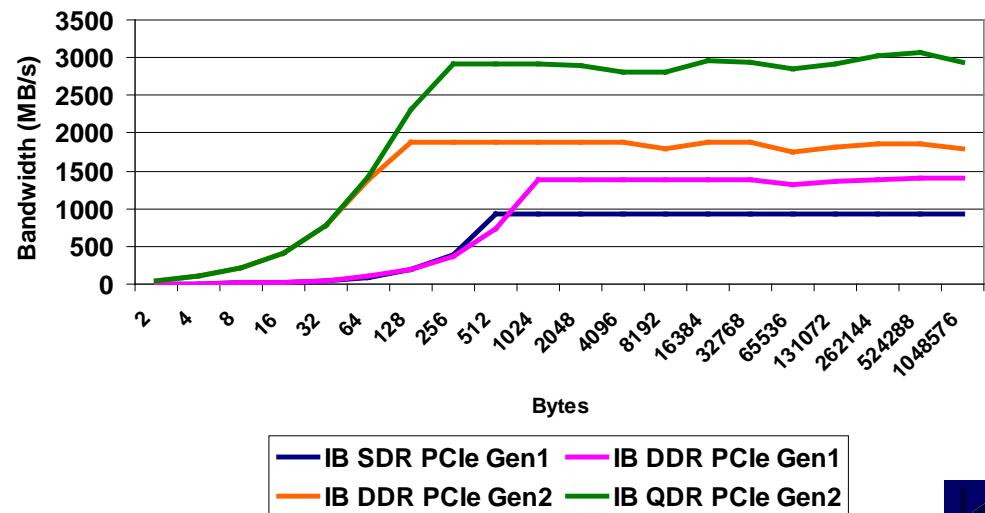


# SDP BCopy ConnectX IB Bandwidth

SDP Bcopy ConnectX IB - SDR, DDR PCIe Gen1, DDR PCIe Gen2, QDR PCIe Gen2



SDP Bcopy ConnectX IB - SDR, DDR PCIe Gen1, DDR PCIe Gen2, QDR PCIe Gen2



# SDP libdsp.so Library

- Dynamically linked library used for replacing the TCP address family with SDP according to a policy.
- 'Hijacks' socket calls and replaces the address family
- Library acts as a user-land socket switch

# SDP libsdp.so Library

## ■ Active Side

- `socket()`
  - Create two sockets, one TCP and one SDP
- `bind()`
- `connect()`
  - Address based decision whether to take SDP or TCP
  - The other socket is closed and the connecting socket is moved to the original file descriptor

## ■ Passive Side

- `socket()`
  - Create two sockets, one TCP and one SDP
- `bind()`
- `listen()`
  - Address based decision whether to take SDP or TCP
  - The other socket is closed and the connecting socket is moved to the original file descriptor
- `accept()`
  - Uses socket that has been decided upon at `listen()`

# SDP in OFED Overview

- Linux TCP Socket implementation
  - Uses standard API
  - Socket type: STREAM
  - New socket family: AF\_INET\_SDP (set to 26)
- Implemented as a kernel module `ib_sdp`
- Implements BCopy and BZCopy operation (Zcopy in upcoming release)

# Configuring SDP

## ■ Loading kernel module

- Automatic (on boot):
  - Edit /etc/infiniband/openib.conf:

SDP\_LOAD=yes

- Restart openibd

- Manual

```
modprobe ib_sdp <_use_zcopy=[0|1] _src_zthresh=[value]>
```

## ■ Change/create kernel application

- Should use AF\_INET\_SDP STREAM sockets
- Include sdp\_inet.h

# Usage – User Level configuration

## ■ Using dynamically loaded libsdp library

- Must set the following environment variables:

```
export LD_PRELOAD=/usr/[lib|lib64]/libsdp.so
export LIBSDP_CONFIG_FILE=/etc/libsdp.conf
```

- Or... Inside the command line

```
env LD_PRELOAD='stack_prefix'/[lib|lib64]/libsdp.so
LIBSDP_CONFIG_FILE='stack_prefix'./etc/libsdp.conf <program>
```

## ■ Simplest usage

- All sockets from AF\_INET family of type STREAM will be converted to SDP

```
export SIMPLE_LIBSDP=1
```

## ■ For more finite control use libsdp.conf

# Usage – User Level configuration (cont.)

## ■ Configure /etc/libdsp.conf

- Substitute particular socket connections by SDP
- Match vs match\_both directives
- Matching according to program name

[match|match\_both] program <regular expr.>

- Matching according to IP address

– on source

[match|match\_both] listen <tcp\_port>

Where tcp\_port is

<ip\_addr>[ /<prefix\_length> ][ :<start\_port>[-<end\_port>] ]

– on destination

match destination <tcp\_port>

# Usage – User Level configuration (cont.)

## ■ Running ssh, scp over SDP

- In `libsdp.conf`:

```
match_both listen *:22
```

- On the server side

```
/etc/init.d/sshd stop
env LD_PRELOAD=/usr/lib64/libsdp.so
LIBSDP_CONFIG_FILE=/u/etc/libsdp.conf /etc/init.d/sshd start
```

- On the client side

```
LD_PRELOAD=/usr//lib64/libsdp.so
LIBSDP_CONFIG_FILE=/etc/libsdp.conf scp <file> <user>@<IPoIB
addr>:<dir>
```

# Debug and monitoring

- Make sure `ib_sdp` module is loaded using:
  - `lsmod | grep sdp`
- To determine if a particular application is actually going over SDP use:
  - `sdpnetstat -S`

# SRP – SCSI RDMA Protocol

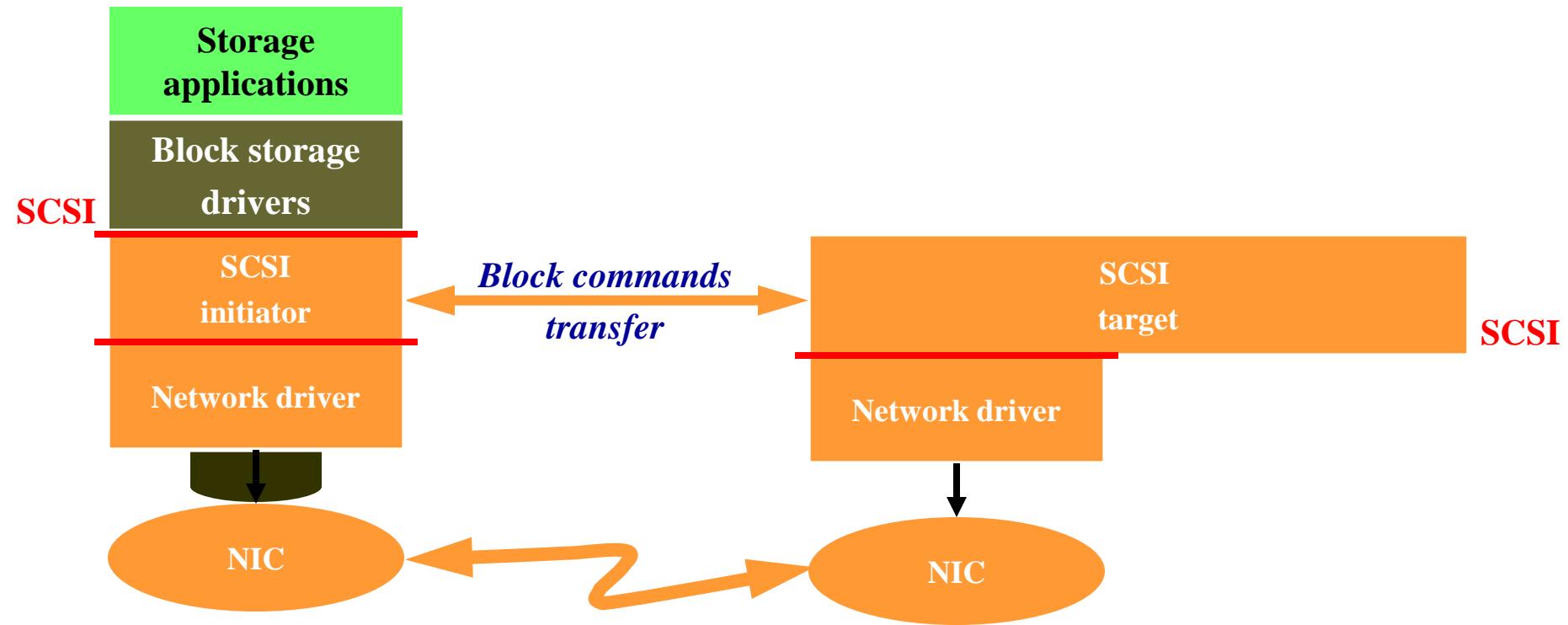


CONFIDENTIAL

# SRP in a Nut Shell

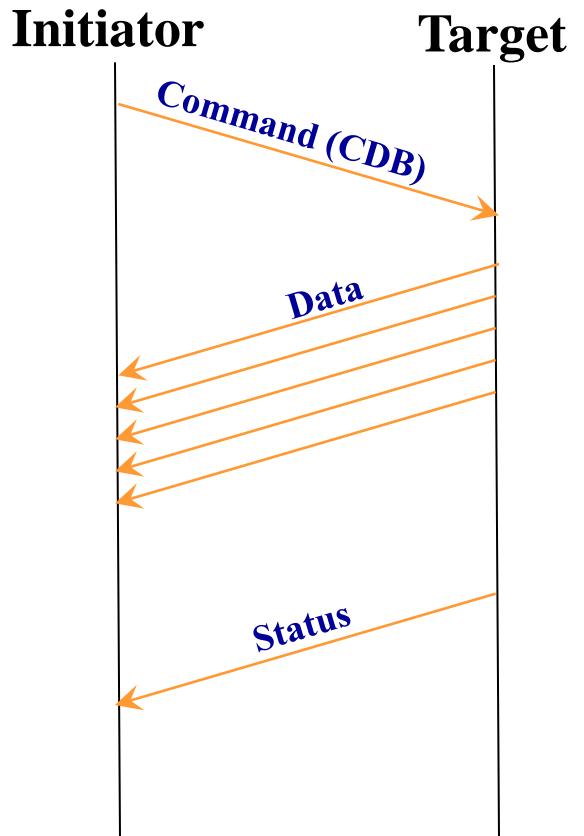
- **Maintain local disk access semantics**
  - Plugs to the bottom of SCSI mid-layer
  - Delivers same functionality as Fiber Channel
  - Provides all hooks for storage network management
    - Requires in-network agents and SW
- **Benefits – protocol offload**
  - Enable RDMA optimized transfers
  - Protocol offload (SAR, retransmission, ack, etc)
- **SRP defines the wire protocol**

# SCSI – from local to network storage



# SCSI I/O Operation

- Initiator sends command to target
  - CDB with transfer attributes
  
- Target transfers data
  
- Status update
  - Success/Failure of operation
  - Busy
  - Not ready
  - Task Set Full
  - Error condition for another task



# SRP connection setup

## ■ Discovery

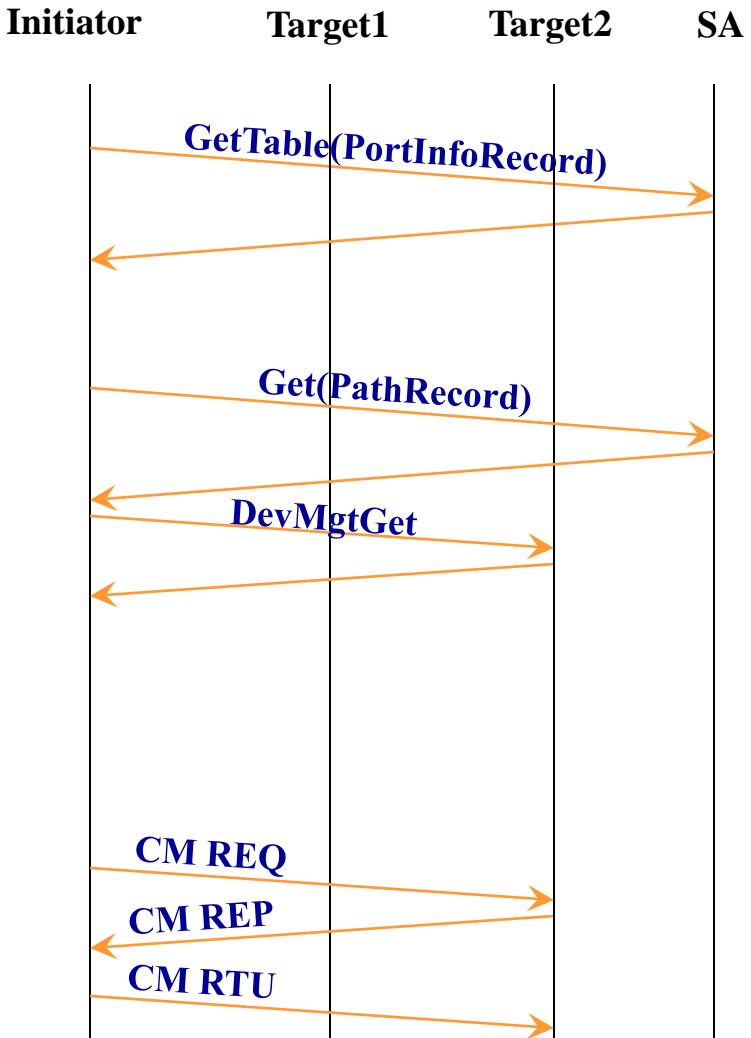
- Query SA for port info data
- Check if a port has DM bit set

## ■ For each IOUNIT found

- Get path record
- Query DM agent
  - IOU Info – how many IO controllers
  - IOControllerProfile – IOC properties
    - which protocol, etc.
  - ServiceEntries - to get the service ID

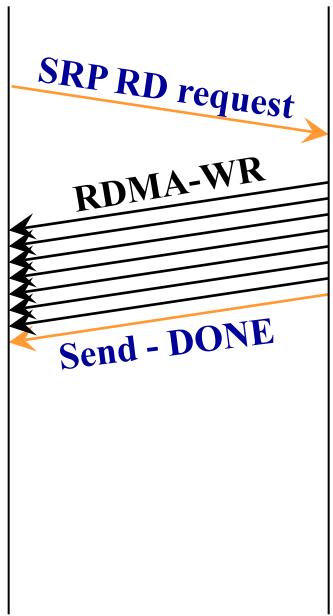
## ■ Login

- 3-way CM connect



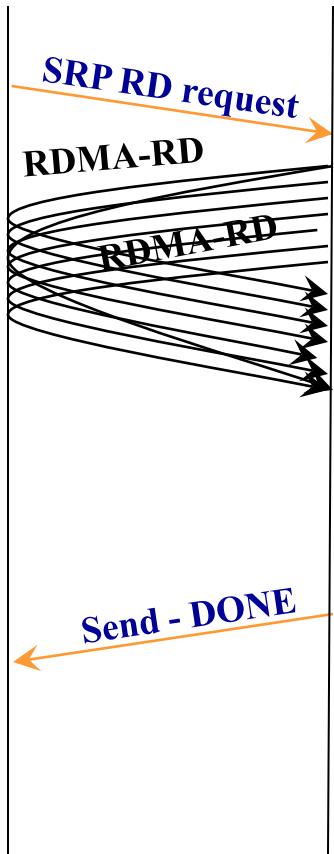
# Disk read

Initiator      Target



# Disk write

Initiator      Target



# Loading SRP Initiator and Target Discovery

- Manual Load: `modprobe ib_srp`
  - Module parameter `srp_sg_tablesize` – max number of scatter/gather entries per I/O – default is 12
- Automatic Load: modify `/etc/infiniband/openib.conf` with `SRP_LOAD=yes`
- Discovering targets
  - `ibsrpdm -c -d /dev/infiniband/umadXX`
    - umad0: port 1 of first HCA in the system (mthca0 or mlx4\_0)
    - umad1: port 2 of first HCA in the system
    - umad2: port 1 of second HCA in the system
    - ...

Example-> `ibsrpdm -c -d /dev/infiniband/umad3`

```
id_ext=0002c9020023130c,ioc_guid=0002c9020023130c,dgid=fe80000000000000
000002c9020023130d,pkey=ffff,service_id=0002c9020023130c
id_ext=0002c9020023193c,ioc_guid=0002c9020023193c,dgid=fe80000000000000
000002c9020023193d,pkey=ffff,service_id=0002c9020023193c
id_ext=0002c9020023187c,ioc_guid=0002c9020023187c,dgid=fe80000000000000
000002c9020023187d,pkey=ffff,service_id=0002c9020023187c
id_ext=0002c9020021d6f8,ioc_guid=0002c9020021d6f8,dgid=fe80000000000000
000002c9020021d6f9,pkey=ffff,service_id=0002c9020021d6f8
```

# Manual Establishing a Connection

- `echo *target login info* > /sys/class/infiniband_srp/srp-mthca[hca#]-[port#]/add_target`
  - Default target login info string:  
`id_ext=[value],ioc_guid=[value],dgid=[target port GID],pkey=ffff,service_id=[value]`
  - Other optional parameters can be in the target login info string
    - `max_cmd_per_lun=[value]` (default is 63)
    - `max_sect=[value]` (default is 512)
    - `io_class=[value]` (default is 0x100 as in rev16A of the srp specification.  
For rev10 srp target the `io_class` value is 0xff00)
    - `initiator_ext`: enabling multiple paths to same target(s)

# SRP Tools - ibsrpdm

## ■ Used to:

- Detect targets on the fabric reachable by the Initiator
- Output target attributes in a format suitable for use in the above “echo” command.
  - To detect all targets run: ibsrpdm
  - To generate output suitable for echo command run: ibsrpdm -c
    - » Sample output:

```
id_ext=200400A0B81146A1,ioc_guid=0002c90200402bd4,
dgid=fe8000000000000000000002c90200402bd5,pkey=ffff,
service_id=200400a0b81146a1
```
    - Next you can copy paste this output into the “echo” command to establish the connection

- **srp\_daemon is based on ibsrpdm and extends its functionalities.**
  - Establish connection to target without manual issuing the \*echo <target login info>\* command
  - Continue running in the background, detecting new targets and establishing connections to targets (in daemon mode)
  - Enable High Availability operation (working together with Device-Mapper Multipath)
  - Have a configuration file (including/excluding targets to connect to)

## ■ srp\_daemon commands equivalent to ibsrpdm

- srp\_daemon –a –o (same as \*ibsrpdm\*)
- srp\_daemon –c –a –o (same as \*ibsrpdm –c\*)

## ■ srp\_daemon extensions

- To discover target from HCA name and port number:  
`srp_daemon –c –a –o –i <mthca0> -p <port#>`
- To discover target and establish connections to them, just add the **\*-e\*** option and remove the **\*-a\*** option to the above commands
- Configuration file /etc/srp\_daemon.conf. Use –f option to provide a different configuration file. You can set values for optional parameters(ie. max\_cmd\_per\_lun, max\_sect...)

# SRP Tools - SRP Daemon

- Run `srp_daemon` in **\*daemon\*** mode
  - `run_srp_daemon -e -c -n -i <hca_name> -p <port#>` → execute srp daemon as a daemon on specific port of a HCA. Please make sure to run only one instance of `run_srp_daemon` per port
  - `srp_daemon.sh` → execute `run_srp_daemon` on all ports of all HCAs in the system. You can look at `srp_daemon` log file in `/var/log/srp_daemon.log`
- Run `srp_daemon` automatically
  - Edit `/etc/infiniband/openib.conf` and turn on `SRPH_A_ENABLE=yes`

# Verifying SRP installation correctness

- “lsscsi” or “fdisk –l” will show the current scsi disk(s) in the system ie. /dev/sda
- Manual loading the SRP module and login to targets
- “lsscsi” or “fdisk –l” will show the new scsi disk(s) in the system ie. /dev/sdb, /dev/sdc,...
- Running some raw “dd”, xdd,... to new block devices ie.  
\*dd if=/dev/sdb of=/dev/null bs=64k count=2000\*
- Creating/mounting file-system
  - fdisk /dev/sdb (to create partitions)
  - mkfs –t ext3 /dev/sdb1
  - mount /dev/sdb1 /test\_srp

# SRP High Availability

- Using Device-Mapper (DM) multipath and srp\_daemon
- There are several connections between an initiator host and target through different ports/HCAs of both host and target
- DM multipath is responsible for identifying paths to the same target and fail-over between paths
- When a path (say from port1) to a target fails, the ib\_srp module starts an error recovery process.

# SRP High Availability

- To turn on and run DM multipath automatically
  - For RHEL4, RHEL5:
    - Edit /etc/multipath.conf to comment out the devnode\_blacklist (rhel4) or the blacklist (rhel5)
    - chkconfig multipathd on
  - For SLES10
    - chkconfig boot.multipathd on
    - Chkconfig multipathd on
- To manually run DM
  - modprobe dm-multipath
  - multipath -v 3 -l → list all luns with paths
  - multipath -m
- Access the srp luns/disks on /dev/mapper

# Exercises



CONFIDENTIAL

## Questions:

1. What is the difference between OFED and MLNX\_OFED?
2. What is the purpose of Subnet Manager in InfiniBand?
3. Which subnet manager comes standard with OFED?
4. What OFED utility used to update FW on HCA cards?
5. What OFED utility used to find link errors?

6. What are the 2 modes IPoIB runs on? What are the advantages of one over the other?
7. What is the IPoIB interfaces called on a dual port HCA card?
8. What is the difference between SDP and IPoIB?
9. What is MPI? What is it used for?
10. What ULP used to run SCSI storage commands over IB?

# Lab Exercise



CONFIDENTIAL

## Exercise #1 – Basic checks

- 1. Check that all nodes have MLX\_OFED install**
  - 1. Install latest MLX\_OFED if missing**
- 2. Which nodes do not have the driver up and running?**
- 3. Are all cards in the cluster the same card type?**
- 4. All port 1 links should be Active. Is this the case?**
- 5. Verify that all HCAs are running 2.6.000 firmware.**

## Exercise #2 – Update firmware

1. Upgrade firmware on all down rev nodes to 2.6.000.

## Exercise #3 – Driver checks

1. Are all machines running OFED-1.4?
2. What is the module parameter that would have to be set to 0 in /etc/modprobe.conf to disable MSI-X interrupts for mlx4 driver?

## Exercise #4 – Subnet manager checks

1. Determine which nodes are running Master and any Standby Subnet managers.
2. Turn off Master SM.
3. Verify that a Standby SM has come on line.
4. Configure your designated node to load OSM automatically on boot-up.

## Exercise #5 – Link diagnostics

1. Clear all port counters in the fabric
2. Run ibdiagnet across the complete cluster to verify that it is running 4x/QDR and the links are error free.

## Exercise #6 – Performance Tests

1. Run `ib_send_bw` between two nodes. What uni-directional bandwidth is achieved? What bi-directional bandwidth
2. Run `ib_write_lat` between two nodes. What latency is achieved?

## Exercise #7 – Switch Queries

1. How many switch devices are in the cluster?
2. What is the firmware version of one of them?

## Exercise #8 – MPI Tests

1. Reset all port counters within cluster.
2. Run Pallas benchmark between two nodes, two processes per node.
3. Check port counters on the complete cluster for any link errors.
4. Are any port\_xmit\_discard counters greater than 0?

# FabricIT



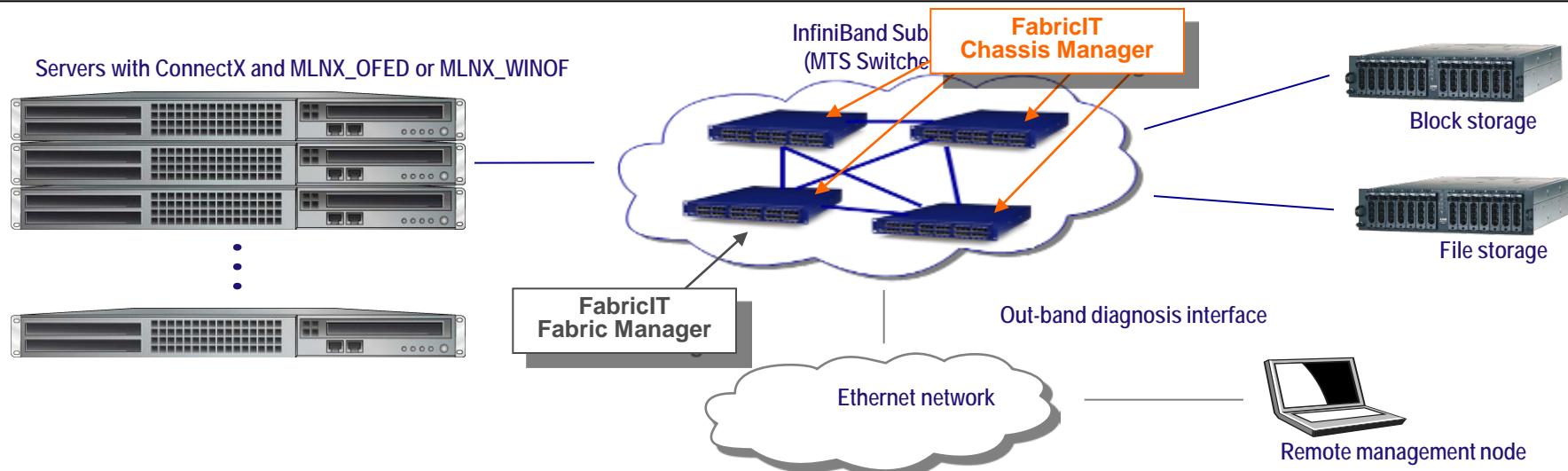
CONFIDENTIAL

# FabricIT Management Software Packages

- **Chassis Management** – ships with all switch systems that have CPU Modules
  - System monitoring
  - RS232 console, 10/100/1000 Eth, IPoIB management
  - CLI / Web Interface / SNMP communication protocols
- **Fabric Management** – FabricIT-EFM
  - Subnet management, cluster diagnostics
  - IPoIB, CLI / Web Interface / SNMP communication protocols

# Embedded Fabric Management Solution

- Switch fabric and chassis management accessed from remote node
- MTS3600/3610 with FabricIT Fabric Manager and chassis management
  - Subnet Manger with fabric diagnostics
  - Hardware monitoring, error and event logging/notification
  - One or two per network
- MTS3600/3610 with chassis management
  - Hardware monitoring, error and event logging/notification
  - All other switches in the fabric



# Embedded Chassis Management Solution



## ■ Hardware monitoring

- Monitor and configure system parameters
- CPU / Memory / File System resources
- Port management
- Power supply management
- LED status
- Voltage, temperature status
- System reset

The screenshot shows a software interface titled "Ports Information & Configuration". At the top, there's a grid of 48 ports labeled 1 through 48. Below the grid, there are two main sections: "Port Info" and "Port Counters".

**Port Info:**

- Port Number: 20
- Logical port state: Active
- Physical port state: LinkUp
- Current line rate: 40.0 Gbps
- Supported speeds: 2.5 / 5.0 / 10.0 Gbps
- Speed: 10.0 Gbps
- Supported widths: x1, x4
- Width: x4
- Max supported MTU: 4096
- MTU: 4096
- VL capabilities: VL0-7
- Operational VLS: VL0-7

**Port Counters:**

- RX bytes: 0
- RX packets: 0
- RX errors: 0
- Symbol errors: 0
- VL15 dropped packets: 0
- TX bytes: 0
- TX packets: 0
- TX wait: 0
- TX discarded packets: 0

## ■ Error and Event Logs on the Switch

## ■ SNMP support

- Get, Traps
- Standard MIBs

## ■ Easy to use communication protocols

- CLI & Web interface
- Secure login and access with ACLs (Telnet/SSH and Secure HTTP)
  - Authentication And Authorization (AAA) : RADIUS, TACACS+
- IPoIB

# Embedded Fabric Management v1.0



- **Fabric Subnet Manager**
  - Subnet Manager and Subnet Administrator
  - Fabric initialization
  - Routing algorithm
  - Execution on boot-up or manually
  - Error logs and Debug Information
- **Advanced features**
  - QoS manager
  - Fabric Inspector cluster management
- **Fabric Inspector**
  - SM status, location, route checks
  - Duplicate GUID/LID's checks
  - Simple and intuitive interface for bring-up and maintenance
- **Additional Mellanox Tools**
  - Switch device Information
  - Switch Firmware upgrades
  - Port status
  - Error logs and Debug Information
- **Easy to use communication protocols**
  - CLI & Web Interface
  - Secure login and access with ACLs (Telnet/SSH and Secure HTTP)
    - Authentication And Authorization (AAA) : RADIUS, TACACS+
  - SNMP Agent
    - 3rd Party management (IBM Tivoli, HP OpenView, packet sniffer) tool interface
  - IPoIB

Mellanox FabricIT IS5035 Management Console

Host: mtilab225  
User: admin [logout]

Mellanox TECHNOLOGIES

Summary	Summary
Temperature	Date and Time: 2000/01/21 07:22:34
Power Supplies	Hostname: mtilab225
Fans	Uptime: 5d 4h 45m 27s
CPU Load	Version: EFM_PPC_M460EX 1.0.2-rc6 2009-07-22 18:38:15 ppc
Memory	Model: ppc
Network	Host ID: 767975f187d7
Logs	System memory: 45 MB used / 1982 MB free / 2027 MB total
	CPU load averages: 0.00 / 0.02 / 0.04

Save Changes? Save

© 2009 Mellanox Technologies, Inc.

# Manager User Interfaces

Mellanox FabricIT MTS3610 Management Console

Host: switch-112082  
User: admin [Logout](#)



SETUP SYSTEM SECURITY PORTS FABRIC MGNT FABRIC INSPECTR STATUS

**Summary**

Summary	
Temperature	Date and Time
Power Supplies	2009/05/21 21:01:46
Fans	Hostname
CPU Load	switch-112082
Memory	Uptime
Network	3h 29m 58.380s
Logs	Version
	EFM_PPC 1.0.0 2009-05-21 13:41:05 ppc
	Model
	ppc
	Host ID
	ecaa8794f376
	System memory
	46 MB used / 458 MB free / 504 MB total
	CPU load averages
	0.51 / 0.61 / 0.62

Unsaved changes [Save](#)

## Web Interface

## Familiar CLI



```
- PuTTY -
```

```

debug          Debugging commands
demo          Set demo constant
echo           Set echo daemon configuration
email          Configure email and event notification via email
exit           Leave configuration mode
file           Manipulate files on disk
ftp-server     Configure FTP server settings
help           View description of the interactive help system
hostname       Set the system's hostname
image          Manipulate system software images
interface      Configure network interfaces
tb8 (config) # interface
ether1 ether2 lo
tb8 (config) # interface
ether1 ether2 lo
tb8 (config) # interface ether1
alias          comment      ip          speed
bond           dhcp         mtu        zeroconf
bridge-group   duplex       shutdown
tb8 (config) # interface ether1
alias          comment      ip          speed
bond           dhcp         mtu        zeroconf
bridge-group   duplex       shutdown
tb8 (config) #

```

# Chassis Initialization



CONFIDENTIAL

# Initial Switch Configuration

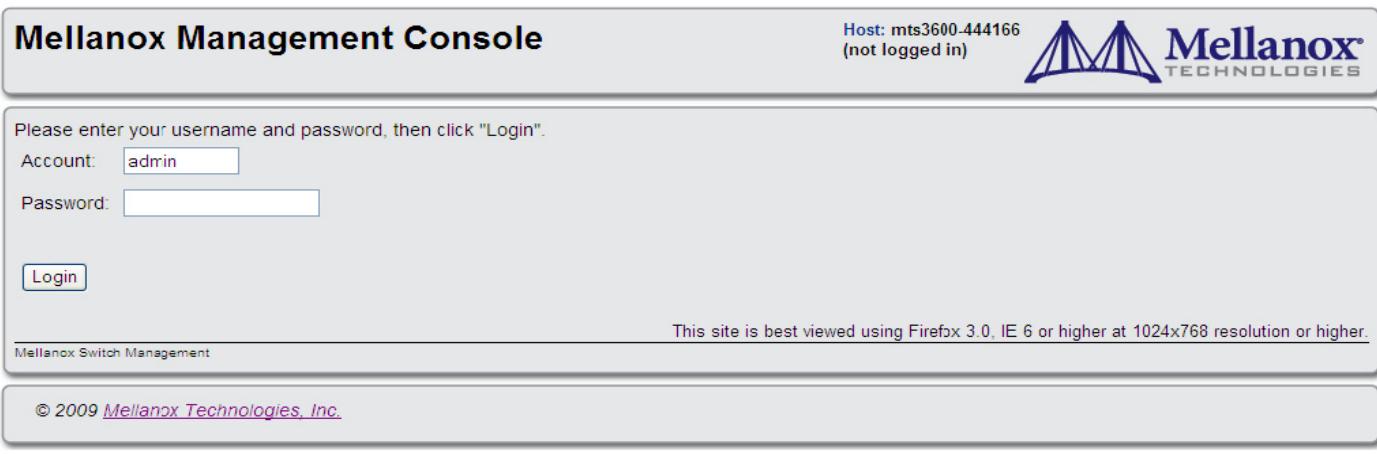
- To change initial switch configuration at first power-on is through RS-232 port of Switch Management Module (RS-232 cable).
- No default IP address is available at this stage. Steps to run Initial Installation:
  - Connect RS-232 cable to management module
  - Configure a serial terminal program (i.e. HyperTerminal) with default serial parameters found in UM
  - Login as admin (password admin)
  - The Mellanox Configuration Wizard will be entered at this point by default.
    - Walk through list of prompts that need to be answered
    - Configures IP address, hostname, passwords, etc
  - Check that eth0 IP address is configured the way you have specified
    - hostname > enable
    - hostname # show interface eth0
- To enter wizard from command line use:
  - hostname # configuration jump-start

# Starting SSH connection to Switch

- Once initial configuration is completed it is possible to access switch through Ethernet port. This will allow CLI and GUI interface to management software.
- Steps to establish connection with an SSH connection once eth0 is configured:
  - Connect Ethernet cable into Ethernet port of Switch Management Module.
  - From a remote machine start an ssh shell to the switch using the command:
    - ssh -l admin 192.168.10 (IP address assigned to eth0)
    - Configures IP address, hostname, passwords, etc
  - Any support CLI command can be entered now

# Starting Web GUI connection to Switch

- Once initial configuration is completed it is possible to interface to the switch through a Web GUI
- Steps to interface with Web GUI:
  - Connect Ethernet cable into Ethernet port of Switch Management Module.
  - Start a Web browser – Internet Explorer 7.0 or Mozilla Firefox 3.0.
    - **Note** Make sure the screen resolution is set to 1024\*768 or higher.
    - Enter URL of [http://<switch\\_eth0\\_IP\\_address>](http://<switch_eth0_IP_address>)
  - Login window for switch will appear in browser



The screenshot shows the 'Mellanox Management Console' login page. At the top, it displays the host information 'Host: mts3600-444166 (not logged in)' and the Mellanox Technologies logo. Below this, there is a form for entering a username and password. The 'Account:' field contains 'admin' and the 'Password:' field is empty. A 'Login' button is located below the password field. A note at the bottom of the form area states, 'This site is best viewed using Firefox 3.0, IE 6 or higher at 1024x768 resolution or higher.' At the very bottom of the page, there is a copyright notice: '© 2009 Mellanox Technologies, Inc.'

# GUI/CLI Interface Overview



CONFIDENTIAL

# Web GUI - Basics

- Home page of the WebGUI has several tabs to click on.

- Status. Default page at login. Includes several status information sub-tabs (system status, uptime, logs, etc).
- Setup. All enclosure setup functions, including network interface setup, SNMP setup, logs/alerts, time/date, etc.
- System. Includes component inventory and status, power management, and image management.
- Security. Includes setting of security features such as passwords, user levels, authentication, etc.
- Ports. Infiniband port control/status.
- Fabric Management. Includes management of fabric subnet.
- Fabric Inspector. Cluster-wide diagnostics.

**Mellanox FabricIT MTS3610 Management Console**

Host: switch-112082  
User: admin ([Logout](#))

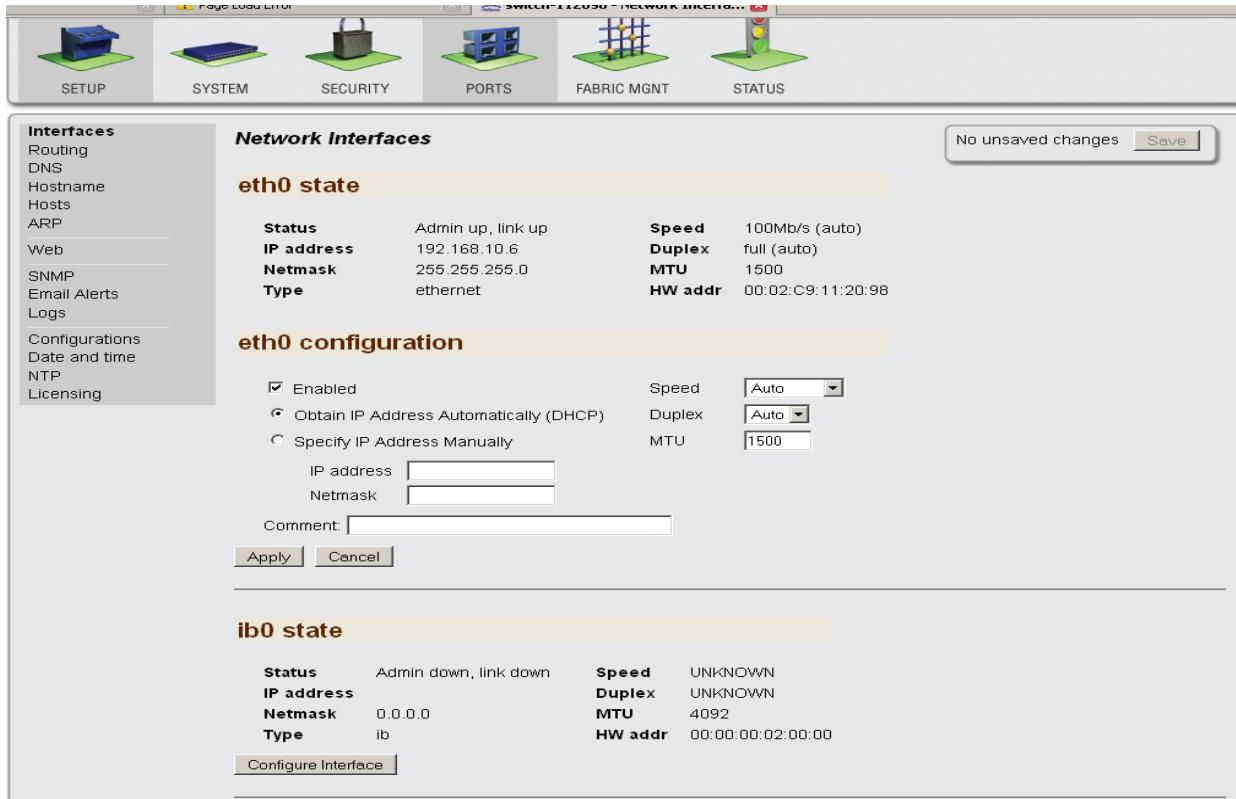


[!\[\]\(dc9000f579511626611ed666ae914fd7\_img.jpg\) SETUP](#)   [!\[\]\(18ac0670c49eff8d4771291498fb68b4\_img.jpg\) SYSTEM](#)   [!\[\]\(67b6f8c3fec1bb64e11553bf347c02b1\_img.jpg\) SECURITY](#)   [!\[\]\(302b225d1d9944a7ae5e127bfc439fc0\_img.jpg\) PORTS](#)   [!\[\]\(9f65b2b3e6f20e7b2efc0557e8179846\_img.jpg\) FABRIC MNGT](#)   [!\[\]\(f92b639e0109056c18e4ac46f4763a1b\_img.jpg\) FABRIC INSPECTR](#)   [!\[\]\(d39188d60d610e91bbe7373b25321881\_img.jpg\) STATUS](#)

<b>Summary</b>		<b>Summary</b>			
Temperature	2009/05/21 21:01:46	Date and Time	2009/05/21 21:01:46	Unsaved changes <input type="button" value="Save"/>	
Power Supplies	switch-112082	Hostname	switch-112082		
Fans	3h 29m 58.380s	Uptime	3h 29m 58.380s		
CPU Load	EFM_PPC 1.0.0 2009-05-21 13:41:05 ppc	Version	EFM_PPC 1.0.0 2009-05-21 13:41:05 ppc		
Memory	ppc	Model	ppc		
Network	ecaa8794f376	Host ID	ecaa8794f376		
Logs	46 MB used / 458 MB free / 504 MB total	System memory	46 MB used / 458 MB free / 504 MB total		
	CPU load averages	0.51 / 0.61 / 0.62	CPU load averages	0.51 / 0.61 / 0.62	

# GUI – Network Interface Setup

- Network interface setup is through the Setup->Interfaces tab.
- Both GigE (eth0) and IPoIB (ib0) are setup on this page.
- Network configuration can be static or through DHCP



The screenshot shows the 'Network Interfaces' configuration page. The left sidebar lists various setup tabs: SETUP, SYSTEM, SECURITY, PORTS, FABRIC MGNT, and STATUS. The 'Interfaces' section is expanded, showing sub-options like Routing, DNS, Hostname, Hosts, ARP, Web, SNMP, Email Alerts, Logs, Configurations, Date and time, NTP, and Licensing.

**eth0 state:**

Status	Admin up, link up	Speed	100Mb/s (auto)
IP address	192.168.10.6	Duplex	full (auto)
Netmask	255.255.255.0	MTU	1500
Type	ethernet	HW addr	00:02:C9:11:20:98

**eth0 configuration:**

Enabled:  Obtain IP Address Automatically (DHCP):  Specify IP Address Manually:

IP address		Speed	Auto
Netmask		Duplex	Auto
		MTU	1500

Comment:

Buttons: Apply, Cancel

---

**ib0 state:**

Status	Admin down, link down	Speed	UNKNOWN
IP address	0.0.0.0	Duplex	UNKNOWN
Netmask		MTU	4092
Type	ib	HW addr	00:00:00:02:00:00

Buttons: Configure Interface

# GUI – Default gateway setup

- Default Gateway setup is through the Setup->Routing tab.

**Mellanox FabricIT MTS3610 Management Console**

Host: switch-112098  
User: admin [Logout](#)

**SETUP** **SYSTEM** **SECURITY** **PORTS** **FABRIC MGMT** **STATUS**

**IP Routing**

No unsaved changes [Save](#)

**Default Gateway**

Default gateway

[Set Default Gateway](#)

**Static and Dynamic Routes**

Destination	Mask	Gateway	Interface	Active	Static
default	0.0.0.0	192.168.10.1	eth0	yes	no
192.168.10.0	255.255.255.0	0.0.0.0	eth0	yes	no

[Remove Selected](#)

**Add Static Route**

Destination   
Netmask   
Gateway

[Add Route](#)

© 2009 [Mellanox Technologies, Inc.](#)

# GUI – Date/Time setup

- Date and Time setup are through the Setup->Data/Time tab.

Mellanox FabricIT MTS3610 Management Console      Host: switch-112098  
User: admin [Logout](#)       **Mellanox**  
TECHNOLOGIES

SETUP SYSTEM SECURITY PORTS FABRIC MGNT STATUS

Interfaces Routing DNS Hostname Hosts ARP Web SNMP Email Alerts Logs Configurations **Date and time** NTP Licensing

**Date and Time**

Date  (YYYY/MM/DD)  
Time  (HH:MM:SS)  
Time Zone

(NTP time synchronization is [enabled](#).)

No unsaved changes

© 2009 [Mellanox Technologies, Inc.](#)

# CLI - Basics

- The CLI (Command Line Interface) is modeled on popular industry standard command line interfaces.
- Context sensitive help at any time by pressing ‘?’ on the command line
  - Shows a list of choices for the word you are on
  - For instance, typing ‘stats ?’ returns all options for the stats command:

```
#> stats ?  
alarm      Configure alarms based on sampled or computed statistics  
chd       Configure computed historical data points  
clear-all  Clear data for all samples and CHDs, and status for all alarms  
export     Export statistics to a file  
sample    Configure sampled statistics
```

- Helpful key shortcuts:
  - TAB- Finishes a partial command
  - Ctrl-A- Moves the cursor to the beginning of the current line
  - Ctrl-U- Erases a line
  - Up Arrow - Allows user to scroll forward through former commands.
  - Down Arrow - Allows user to scroll backward through former commands.

# CLI - User Modes

- The CLI can be in one of 3 modes. Each of these modes makes available a group of commands for execution.
  - Standard Mode
    - CLI launched into Standard mode
    - Most restrictive. Users cannot directly affect the system or change any configuration in this mode.
  - Enable Mode
    - Offer commands to view all state information, take actions like rebooting the system, but it does not allow any configuration to be changed.
    - Entered from Standard mode by running ‘enable’
  - Configure Mode
    - Configure mode is allowed only for user accounts with ‘admin’ permissions
    - Full unrestricted set of commands to view anything, take any action, or change any configuration.
    - Entered from enable mode running ‘configure terminal’.

# CLI - User Modes (cont.)

- Prompt begins with hostname of system followed by indicator for mode that user is in. For example:
  - switch-1 > (Standard mode)
  - switch-1 # (Enable mode)
  - switch-1 (config) # (Config mode)
- The following session shows how to move between command modes:

```
switch-1 >                                // Start in Standard mode
switch-1 > enable   // Move to Enable mode
switch-1 #                      // In Enable mode
switch-1 # configure terminal // Move to Config mode
switch-1 (config) #  // In Config mode
switch-1 (config) # exit // Exit Config mode
switch-1 #                      // Back in Enable mode
switch-1 # disable   // Exit Enable mode
switch-1 >                                // In Standard mode
```

# CLI – Special Command Forms

## ■ 'show' command

- Can be used in any User mode to show system configurations or statistics
- Follow show by ? to get a list of show specific keyword commands
  - e.g. to show current switch image version

```
switch-1 > show version
Product name: EFM_PPC
Product release: <version>
Build ID: <id>
Build date: 2009-05-13 16:26:35
```

## ■ 'no' command

- Provides negations of several Config mode commands
- Can be used to disable a function or to cancel certain command parameters or options
- To re-enable, re-enter the command without the 'no' keyword
  - e.g. disable auto-logout

```
switch-1 (config) # no cli session auto-login
```
  - e.g. to re-enable auto-login for 15 minutes

```
switch-1 (config) # cli session auto-login 15
```

# CLI – Network Interface Setup

- Network Interface commands define the IP address and attributes of the network interfaces of the chassis
  - To set the IP address
    - switch-1 (config) interface eth0 10.2.2.10 255.255.0.0
  - To disable DHCP on the interface:
    - switch-1 (config) no interface eth0 dhcp
  - To display information about the interface
    - switch-1 (config) show interfaces eth0
  - To set hostname:
    - switch-1 (config) hostname <hostname>
  - To set the default gateway
    - switch-1 (config) ip default-gateway <next hop IP address or Interface> [<Interface>]

# Chassis Management



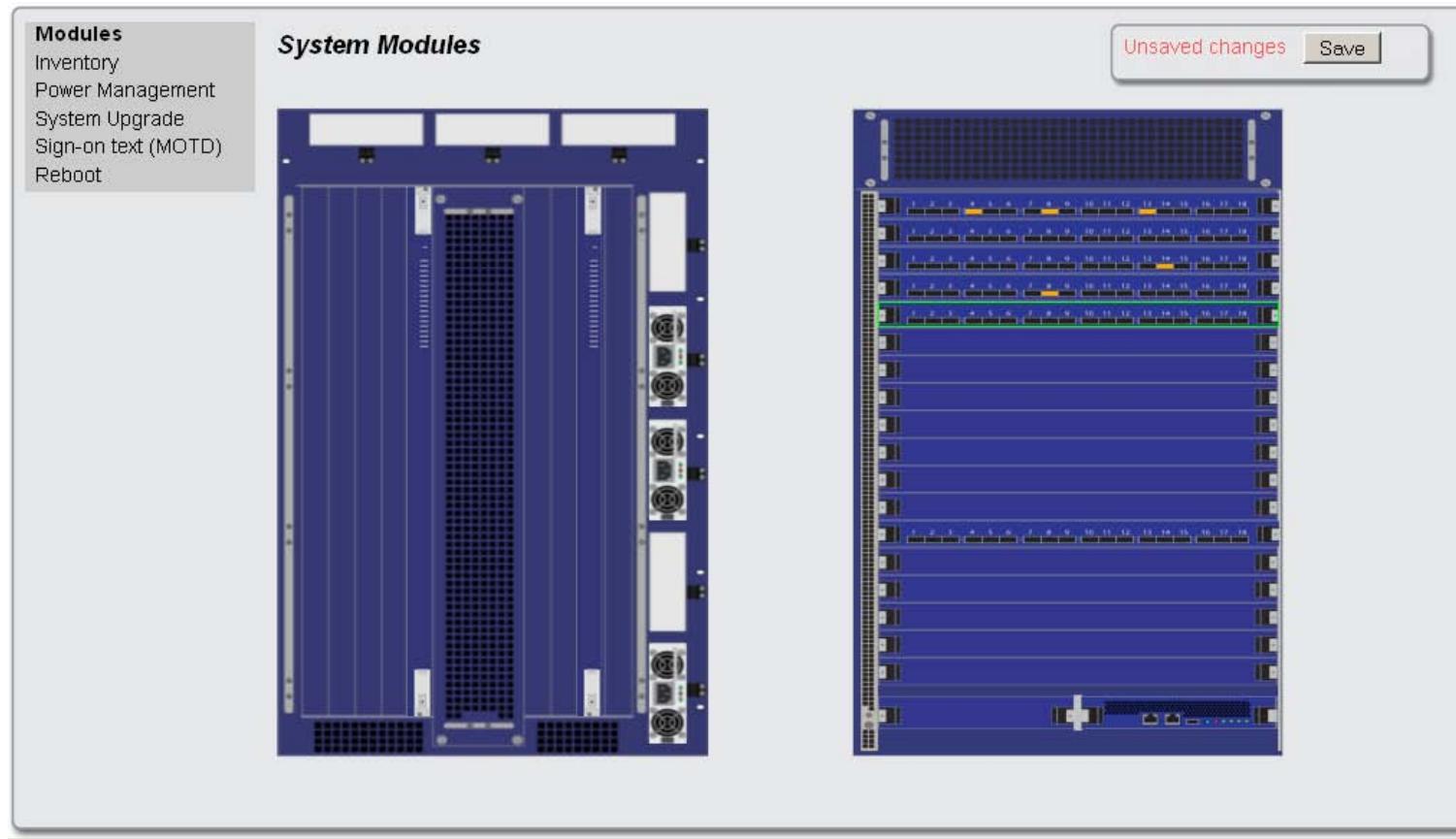
CONFIDENTIAL

# What is Chassis Management

- Chassis Management Interfaces in CLI and GUI provide a way to obtain following information:
  - Monitor and configure system parameters
  - CPU / Memory / File System resources
  - Port management
  - Power supply management
  - LED status
  - Voltage status
  - Fan status
  - System reset

# GUI - System Monitoring

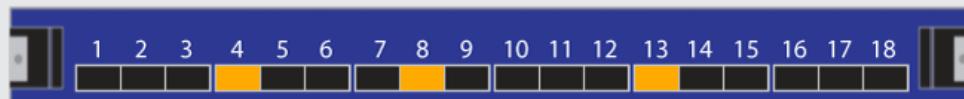
- Overall top level system view of switch chassis components.
- Hierarchical view. ‘Click on’ various components to push down into component for more detailed info. ‘Hovering’ over a component will display component in pop-up window.



# GUI - System Monitoring

- Push down into various components of the system for full details and environmental conditions of the selected component.

## Leaf #1 Status:



## Leaf part information:

Type:	MTS3610_LEAF
S/N:	MT0920X01239
P/N:	MTS3611QC

## Leaf Temperatures:

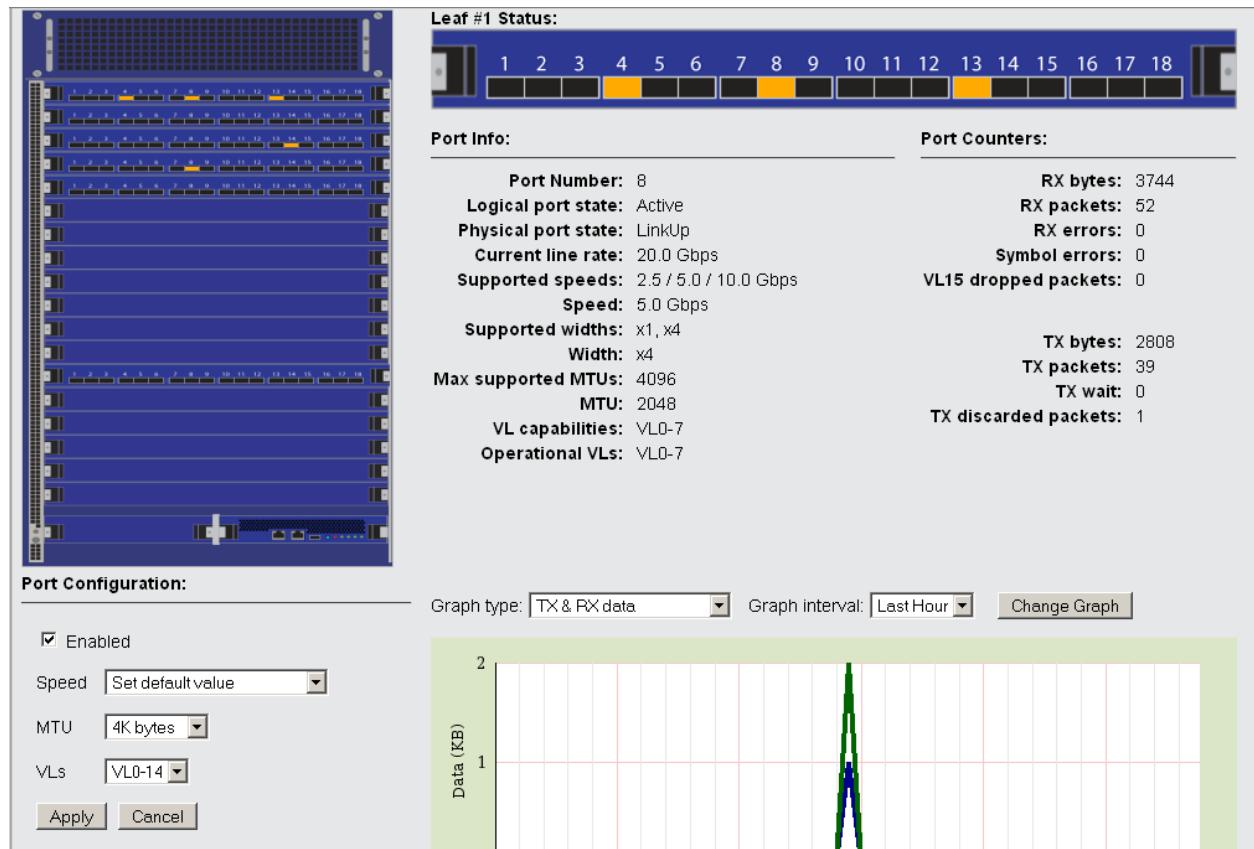
Module	Sensor	Temperature (Celsius)	Status
L01	BOARD_MONITOR	19	OK
L01	IS4_AMBIENT_TEMP	24.5	OK
L01	IS4_PRIM	31	OK
L01	PS_AMBIENT_TEMP	21	OK

## Leaf Voltage:

Module	Sensor	Reg	Expected Voltage	Actual Voltage	Status
L01	BOARD_MONITOR	V1	2.50	2.49	OK
L01	BOARD_MONITOR	V2	2.50	2.49	OK
L01	BOARD_MONITOR	V3	3.30	3.31	OK
L01	BOARD_MONITOR	V4	2.50	2.49	OK
L01	BOARD_MONITOR	V5	1.80	1.77	OK
L01	BOARD_MONITOR	V6	3.30	3.30	OK
L01	BOARD_MONITOR	V7	1.20	1.20	OK

# GUI – Port Monitoring

- Port information is Provided through the Ports Button the main page.
- Port information includes port attributes and port counters.
- This page also contains a port counter histogram



# GUI – System Environmental Monitoring

- Fan tray status and power unit status can be seen from top System view
- Pushing down into the various components gives full details.

**Fans Status:**

Module	Fan	Speed (RPM)	Status
FAN_LEAF	F1	8320.49	OK
FAN_LEAF	F2	8346.21	OK
FAN_LEAF	F3	8307.69	OK
FAN_LEAF	F4	8463.95	OK
FAN_LEAF	F5	8503.94	OK
FAN_LEAF	F6	8450.70	OK
FAN_LEAF	F7	8450.70	OK



**Power Supply #8 Status:**



Module	Sensor	Power (Watts)	Voltage	Current (Amp)	Status
PS8	PS_MONITOR	254.75	48.29	5.28	OK

# GUI – System Environmental Monitoring

- System->Power Management tab provides system level power supply status.

**Power Management**

No unsaved changes

**Current Power Supplies Status:**

Module	Sensor	Power (Watts)	Voltage	Current (Amp)	Status
PS1	PS_MONITOR	-	-	-	NOT PRESENT
PS2	PS_MONITOR	-	-	-	NOT PRESENT
PS3	PS_MONITOR	250.03	48.29	5.18	OK
PS4	PS_MONITOR	268.21	48.05	5.37	OK
PS5	PS_MONITOR	254.75	48.29	5.28	OK
PS6	PS_MONITOR	334.95	48.29	6.94	OK
PS7	PS_MONITOR	261.62	47.82	5.47	OK
PS8	PS_MONITOR	262.90	48.05	5.47	OK

**Total Power summary:**

**Total power used:** 1622.46 Watts

**Total power capacity:** 8064.00 Watts

**Total power available:** 6441.54 Watts

# GUI – System FRU

- System->Inventory tab provides system component FRU information.

**System Inventory**

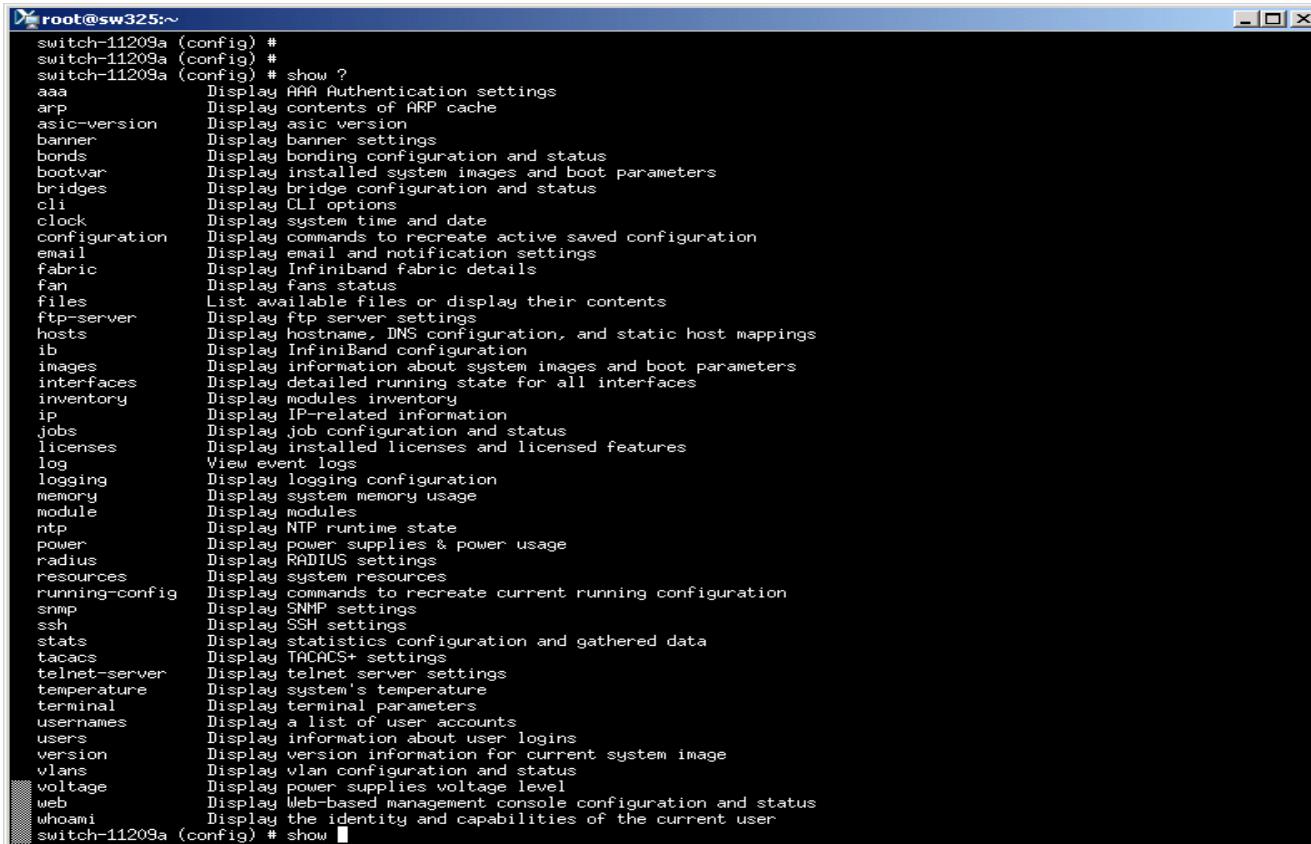
No unsaved changes

**Modules Information:**

Module	Type	S/N	P/N
MGMT	MTS3610_MGMT	N/A	N/A
PS3	FPS1000	N/A	N/A
PS4	FPS1000	N/A	N/A
PS5	FPS1000	N/A	N/A
PS6	FPS1000	N/A	N/A
PS7	FPS1000	N/A	N/A
PS8	FPS1000	N/A	N/A
FAN LEAF	MTS3610_LEAF_FAN	N/A	N/A
L01	MTS3610_LEAF	SN-/L01/VPD	PN-/L01/VPD
L02	MTS3610_LEAF	SN-/L02/VPD	PN-/L02/VPD
L03	MTS3610_LEAF	SN-/L03/VPD	PN-/L03/VPD
L04	MTS3610_LEAF	SN-/L04/VPD	PN-/L04/VPD
L05	MTS3610_LEAF	SN-/L05/VPD	PN-/L05/VPD
L06	MTS3610_LEAF	SN-/L06/VPD	PN-/L06/VPD
L07	MTS3610_LEAF	SN-/L07/VPD	PN-/L07/VPD
L08	MTS3610_LEAF	SN-/L08/VPD	PN-/L08/VPD
L09	MTS3610_LEAF	SN-/L09/VPD	PN-/L09/VPD
L10	MTS3610_LEAF	SN-/L10/VPD	PN-/L10/VPD

# CLI - System Monitoring

- Chassis management commands are used to check status of fans, obtain module temperate, display switch system configuration, etc. and are accessed through show commands
- Reminder: ‘show ?’ will give a list of all possible command options.



```

root@sw325:~#
switch-11209a (config) #
switch-11209a (config) #
switch-11209a (config) # show ?
aaa                  Display AAA Authentication settings
arp                  Display contents of ARP cache
asic-version         Display asic version
banner               Display banner settings
bonds                Display bonding configuration and status
bootvar              Display installed system images and boot parameters
bridges              Display bridge configuration and status
cli                  Display CLI options
clock                Display system time and date
configuration        Display commands to recreate active saved configuration
email                Display email and notification settings
fabric               Display Infiniband fabric details
fan                  Display fans status
files                List available files or display their contents
ftp-server           Display ftp server settings
hosts                Display hostname, DNS configuration, and static host mappings
ib                  Display InfiniBand configuration
images               Display information about system images and boot parameters
interfaces           Display detailed running state for all interfaces
inventory            Display modules inventory
ip                  Display IP-related information
jobs                Display job configuration and status
licenses             Display installed licenses and licensed features
log                 View event logs
logging              Display logging configuration
memory              Display system memory usage
module               Display modules
ntp                 Display NTP runtime state
power               Display power supplies & power usage
radius               Display RADIUS settings
resources            Display system resources
running-config       Display commands to recreate current running configuration
snmp                Display SNMP settings
ssh                 Display SSH settings
stats               Display statistics configuration and gathered data
tacacs              Display TACACS+ settings
telnet-server        Display telnet server settings
temperature          Display system's temperature
terminal             Display terminal parameters
usernames            Display a list of user accounts
users               Display information about user logins
version              Display version information for current system image
vlans               Display vlan configuration and status
voltage              Display power supplies voltage level
web                 Display Web-based management console configuration and status
whoami              Display the identity and capabilities of the current user
switch-11209a (config) # show ?

```

# CLI - System Monitoring

## ■ Some useful chassis management cli commands:

- ‘show fans’ - Display system fan status
- ‘show module’ – Display list of installed modules
- ‘show inventory’ – FRU information for all installed modules
- ‘show power’ – Display main power supplies information and power usage
- ‘show temperature’ – Display system’s temperature
- ‘show voltage’ – Display all power supplies voltage levels
- ‘show stats alarm temperature’ - Display temperature alarm thresholds and current temperature measurements.
- ‘show ib ports’ – Display the state of all IB ports. Can be chassis wide, card wide, or specific port.
- ‘show resources’ - Display the system resources: memory size and utilization, CPU(s) and utilization, etc.
- ‘show fabric pm’ – Display fabric diagnostic information on all ports in the fabric.

# Updating the Software Image

- Switch software image contains kernel, management software modules, and all switch device firmware.
- New image is copied to the switch via scp, or selecting image file via browsing facility in GUI.
- Once image is copied onto switch, this ‘new image’ can be installed and selected to be the bootable image
- After system reboot, new image is loaded.

# GUI - Updating Software Image

- Update Software (including device firmware) through System->System Upgrade tab.
- Select installation file, and then click on ‘Install Image’ to download the new image.
- Click on ‘Switch Boot Partition’ to make new image the active one.

Modules  
Inventory  
Power Management  
**System Upgrade**  
Sign-on text (MOTD)  
Reboot

**System Upgrades and Imaging**

No unsaved changes

**Installed Images**

**Partition 1 (currently booted) (to boot next)**  
EFM\_PPC 1.0.0 2009-05-19 18:59:53 ppc

**Partition 2**  
EFM\_PPC 1.0.0 2009-05-20 03:15:28 ppc

**Install New Image to Partition 2**

Install from URL:   
 Install via scp (pseudo-URL format: `scp://username@hostname/path/image.img`):  
 URL:   
 Password:

Install from local file:    
 (Progress tracking begins after file is uploaded)

View image upgrade progress

To activate a newly-installed software image, please [reboot](#) the system.

# CLI - Updating Software Image

- To upgrade FabricIT software on your system from the CLI, perform the following steps:

- Copy the new software image

```
switch-1 (config) # image fetch  
scp://<user>@192.168.10.125/var/www/html/<image_name>
```

- Display the available images

```
switch-1 (config) #show images
```

Images available to be installed:

```
new_image.img EFM <new ver> 2009-05-13 16:52:50
```

Installed images:

```
Partition 1: EFM <old ver> 2009-05-13 03:46:25
```

```
Partition 2: EFM <new ver> 2009-05-13 03:46:25
```

```
Last boot partition: 1
```

```
Next boot partition: 1
```

- Install the new image

```
switch-1 (config) # image install <image_name>
```

# CLI - Updating Software Image (cont)

- Make the new image active (next boot will use the new image)

```
switch-1 (config) # image boot next
```

- Display the available images

```
switch-1 (config) # show images
```

Images available to be installed:

new\_image.img EFM <new ver> 2009-05-13 16:52:50

Installed images:

Partition 1: EFM <old ver> 2009-05-13 03:46:25

Partition 2: EFM <new ver> 2009-05-13 16:52:50

Last boot partition: 1

**Next boot partition: 2**

# CLI Updating Switch Firmware

- Firmware updates to the switch devices in the system are through CLI only.
- A firmware image can be updated across multiple devices (i.e. all Mammoth Spine cards can be update simultaneously).
- Firmware is updated in-band if possible, and through i2c bus if the in-band link is not available.
- Steps to updating firmware:
  - fetch images: image fetch <url>
  - burn images: image install-is4-fw
- Example:

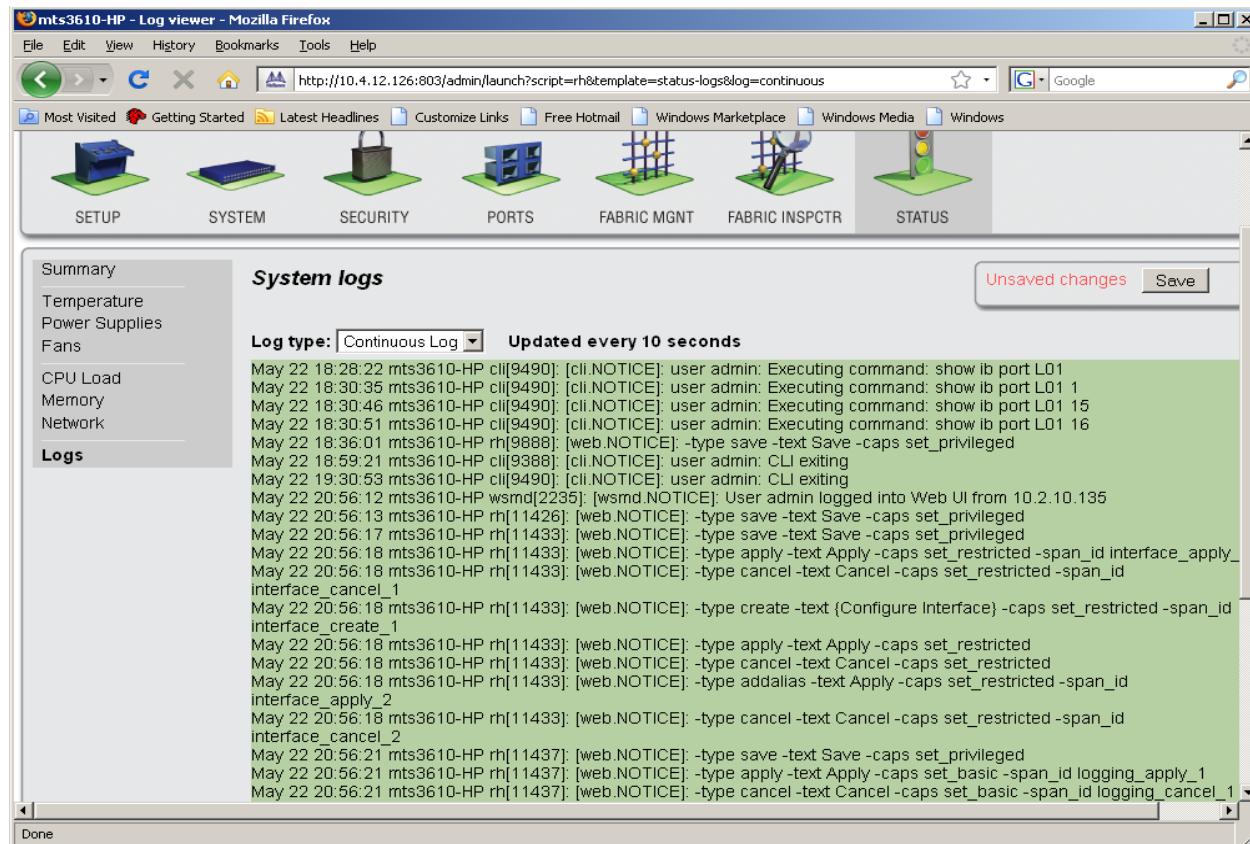
```
switch-1 (config) # image fetch
```

```
http://192.168.10.125/firmware/MTS3610QSC-SPINE.bin
```

```
switch-1 (config) # image install-is4-fw SPINES MTS3610QSC-SPINE.bin
```

# Logs

- System logs are setup in the WebGUI through the ‘Setup->Logs’ tab. Setup can include type of log level to filter, log depth, remote sink, etc.
- Can setup syslog to dump the log to external server.
- System logs are viewed through the ‘Status->Logs’ tab.



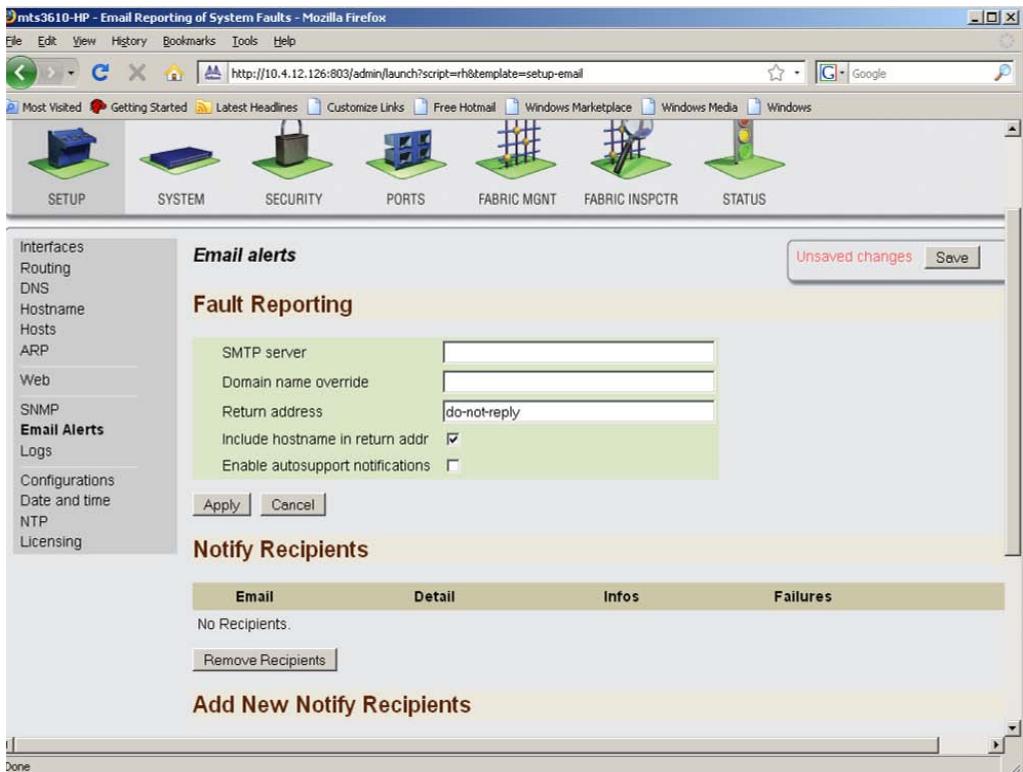
# Event Notification

## ■ Email Alerts

- Email alerts setup done through ‘Setup->Email Alerts’ tab.
- Possible to add email server information, recipients, type of alerts, etc
- Enable/disable for info and on failures.

## ■ SNMP

- SNMP traps setup through ‘Setup->SNMP’ tab
- **Supported traps listed in User’s Manual. Includes items such as Link up/down, CPU load too high, process crashed, etc.**



# Fabric Management



CONFIDENTIAL

# Fabric Management - Features

- FabricIT Fabric Management is based on OpenSM and is an Infiniband compliant subnet manger.
- Must have purchased EFM (Embedded Fabric Manager) piece to use this feature.
- Ability to run several instances of FabricIT SM on the cluster in a Master/Slave(s) configuration for redundancy.
- Partitions (p-key) support
- QOS support
- Enhanced routing algorithm support:
  - Min-hop
  - Up-down
  - Fat-tree
  - LASH and DOR
  - Table based

# Licensing

- First things first.....a EFM license must be installed on the system to use Fabric Management.
- EFM license is purchased separate. License key will be downloaded by customer from license website (work in progress).
- Licenses are added under the 'Setup-Licensing' page.

Screenshot of the 'Setup-Licensing' page interface:

System Serial Number		
MT1234X12345		
Installed Licenses		
License		
<input type="checkbox"/>	Key	LK2-EFM_CONFIG-5G22-85HG-L0KA-L64R-36D2-Q64R-36D1-M88A-60UA-RTQV-GG
	Feature	EFM_CONFIG
	Valid	yes
	Chassis serial number	MT1234X12345
	Active	yes
<input type="checkbox"/>	Key	LK2-RESTRICTED_CMDS-88A0-RFD7-W4CF-Y
	Feature	RESTRICTED_CMDS
	Valid	yes
	Active	yes
<input type="checkbox"/>	Key	LK2-EFM_CONFIG-88A7-29T3-5MDU-P
	Feature	EFM_CONFIG
	Valid	yes
	Active	yes

**Add New License(s)**

Please enter one or more licenses, each on a separate line.

# GUI - Fabric Management

- Infiniband Fabric Management GUI is used to manage the Subnet Manager of the fabric
- Base SM features found in ‘Fabric Mgmt->Base SM’ buttons.
- Allows admin to enable/disable and set priority of the SM.

Mellanox FabricIT Management Console      Host: switch-11208c      User: admin ([Logout](#))



SETUP    SYSTEM    SECURITY    PORTS    FABRIC MGNT    FABRIC INSPECTR    STATUS

**Summary**

**Base SM**

Advanced SM    Expert SM

**Basic SM configuration**

Enabled    Priority: 0 - lowest ▾

No unsaved changes    [Save](#)

[Apply](#)    [cancel](#)

© 2009 [Mellanox Technologies, Inc.](#)

# GUI - Fabric Management (cont)

- Advanced configuration options are possible through ‘Fabric Mgmt->Advanced SM’ buttons
- Entries include number of LIDS/port, number of VLs, timeouts, etc.

**Mellanox MTS3610 Management Console**

Host: mammoth1  
User: admin ([Logout](#))

**Mellanox**  
TECHNOLOGIES

SETUP SYSTEM SECURITY PORTS FABRIC MGNT STATUS

**Advanced Subnet Manager configuration**

**Basic configuration for different size fabrics**

Unsaved changes

1 LID per-port	<input type="button" value="Imc"/>	<input type="checkbox"/> Imc_esp0
5	<input type="button" value="max_op_vls"/>	About 1 second <input type="button" value="subnet_timeout"/>
200	<input type="button" value="transaction_timeout"/>	10000 <input type="button" value="max_msg_fifo_timeout"/>
4	<input type="button" value="polling_retry_number"/>	10000 <input type="button" value="sminfo_polling_timeout"/>
4	<input type="button" value="max_wire_smps"/>	10 <input type="button" value="sweep_interval (in seconds)"/>

# CLI - Fabric Management

- InfiniBand Subnet Manager (**ib sm**) commands are used to manage the Subnet Manager service running on the switch
- All fabric/subnet management commands are in ‘ib->sm’ submenu.
- ‘ib sm’ options can all be included in a single command line or entered separately.

- ‘show ib sm’ gives SM status
- ‘show ib sm’ shows all possible SM attributes to query

```

root@sw325:~#
switch-11209a (config) #
switch-11209a (config) # show ib sm
disable
switch-11209a (config) # show ib sm ?

<cr>
accum-log-file      Show if SM overwrites or appends the current log file
babbling-policy     Show if SM is allowed to disable babbling ports
connect-roots       Display special IBRA compliant multi-stage switch directive
console            Show internal console support
console-port        Display telnet socket for internal 'socket' or 'loopback' console
daemon             Show if SM automatically starts in the background
enable-quirks      Show if SM applies HW workarounds and some high risk features
exit-on-fatal       Show if SM exits normally after a fatal error
force-link-speed    Display maximum time packet can remain queued in a switch
force-log-flush     Show if SM forces a flush after every log write
guid2lid-cache      Display if SM is allowed to use cached guid to lid mapping data
honor-partitions   Display subnet partition enforcement policy
hoq-lifetime       Maximum switch-to-switch head of transmit queue lifetime
ignore-other-sm    Display dangerous SM control variable to ignore election
ipv6-nsm           Show if IPv6 SNM group joins are consolidated
leafhoq-lifetime   Maximum head of switch-to-CA/RTR transmit queue lifetime
leafvl-stalls      Display count of sequentially dropped frames before stall state
lmc                Display subnet LMC (number LIDs/port)
lmc-esp0            Display LMC enabled/disabled for enhanced switch port 0
log-flags           Display log flags
log-max-size       Show maximum size of SM log file
m-key              Display m_key value (0=default, not use)
max-op-vls          Display maximum number of VLS supported by this subnet
max-ports           Show how many CA ports SM can manage
max-reply-time     Display maximum time SM will wait for reply
max-wire-smps      Display SM maximum concurrent mgmt packets
mkey-lease          Display m_key period in seconds
msfifo-timeout     Display maximum time SA query will wait before BUSY returned on new queries
multicast          Show if SM is supporting multicast on the fabric
no-client-rereg    Show if client reregistration requests are processed
overrun-trigger    Display count of local buffer overrun errors for trap 130
packet-life-time   Display maximum time packet can remain queued in a switch
phy-err-trigger    Display count of local link integrity errors for trap 129
polling-retries    Display number of missed polls before active SM considered dead
reassign-lids      Display subnet lid reassignment policy
routing-engines   Display ordered list of routing engines
sa-key             Display sa_key
single-thread      Show if SM is allowed to use more than one thread
sm-inactive         Show if SM starts in inactive (no SM/SA function) mode
sm-key              Display sm_key value
sm-priority         Priority of SM on this node (0=lowest, 15=highest)
sminfo-poll-time   Display maximum time SM will wait between polls of active SM
subnet-prefix       Display subnet_prefix value
subnet-timeout     Display PortInfo:SubnetTimeOut and maximum trap frequency
sweep-interval     Display SM sweep_interval; 0=disabled
sweep-on-trap      Display SM requirement to always use heavy sweep after trap
use-heavy-sweeps   Display SM requirement to always use heavy sweeps
vl-stalls          Display count of sequentially dropped frames before stall state
switch-11209a (config) # show ib sm

```



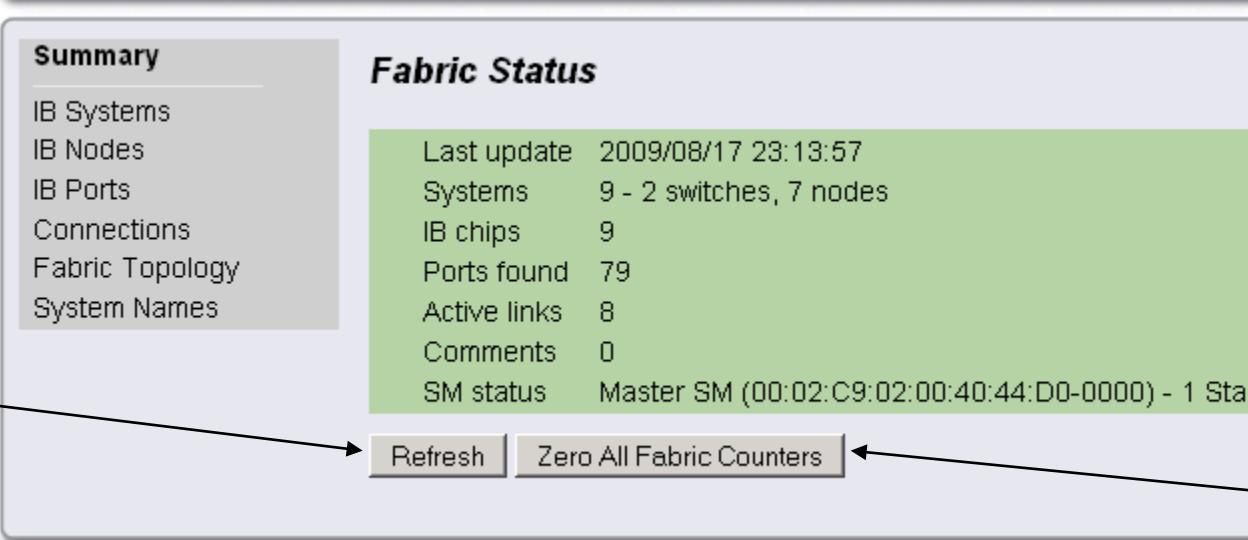
# Fabric Inspector



CONFIDENTIAL

# Fabric Inspector - Features

- FabricIT Fabric Inspector GUI provides simple interface to monitor and debug cluster.
- Includes advanced filtering techniques to quickly isolate problem areas.
- All data is based on the last sweep of cluster. In current version of FabricIT sweeps are kicked off manually.
- Inspector main page:



**Sweep fabric**

**Fabric Status**

Last update	2009/08/17 23:13:57
Systems	9 - 2 switches, 7 nodes
IB chips	9
Ports found	79
Active links	8
Comments	0
SM status	Master SM (00:02:C9:02:00:40:44:D0-0000) - 1 Sta

**Reset**

**Counters**

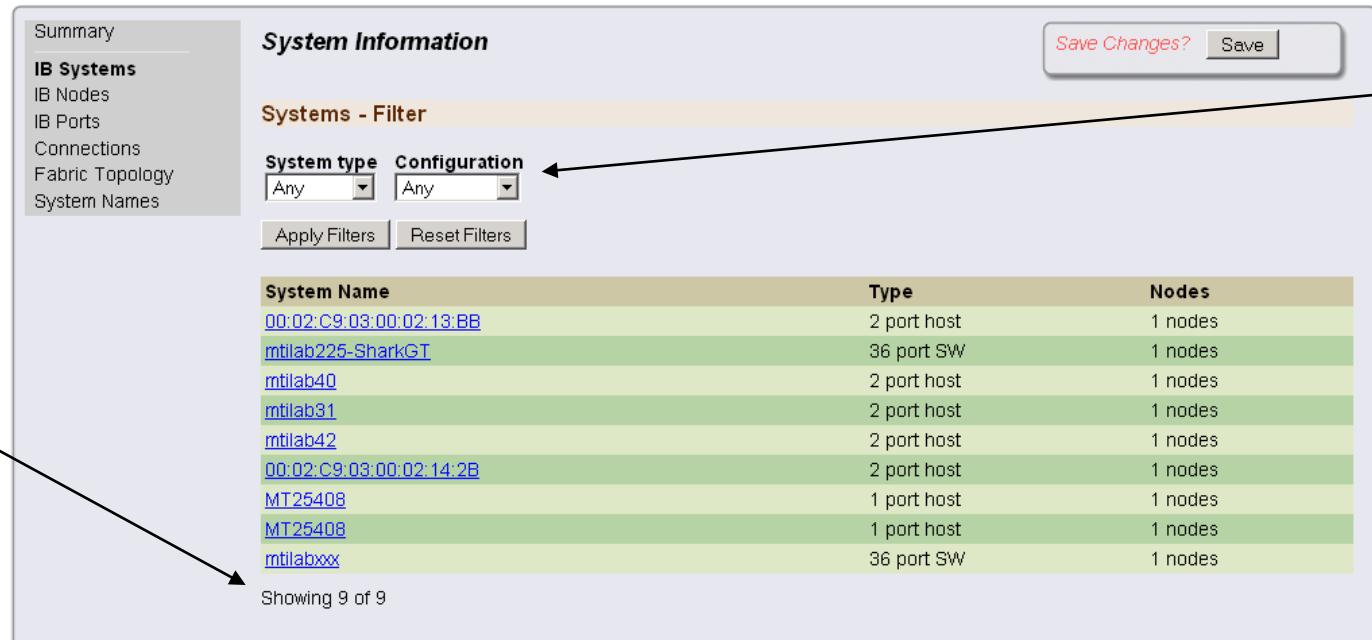
Refresh   Zero All Fabric Counters

# Fabric Inspector – Systems Page

- Systems Page shows all Infiniband Systems (switches and hosts) in the cluster.
- Each switch system is treated as one system. This means a Mammoth will show up once (not 27 times!!).
- System Names are used if they have them. If not, the GUID is used.
- System Names page provides a way to assign names to systems (more on this later).

**9 Systems in cluster**

**Filters**



System Name	Type	Nodes
<a href="#">00:02:C9:03:00:02:13:BB</a>	2 port host	1 nodes
<a href="#">mtilab225-SharkGT</a>	36 port SW	1 nodes
<a href="#">mtilab40</a>	2 port host	1 nodes
<a href="#">mtilab31</a>	2 port host	1 nodes
<a href="#">mtilab42</a>	2 port host	1 nodes
<a href="#">00:02:C9:03:00:02:14:2B</a>	2 port host	1 nodes
<a href="#">MT25408</a>	1 port host	1 nodes
<a href="#">MT25408</a>	1 port host	1 nodes
<a href="#">mtilabxx</a>	36 port SW	1 nodes

Showing 9 of 9

# Fabric Inspector – Nodes Page

- Nodes Page shows all Infiniband Devices (switches and HCAs) in the cluster.
- Each device in a switch system is displayed. This means a Mammoth will show a maximum of 27 Nodes.
- Filters are provided to show only HCAs, only Leafs, Spines, etc.

**Summary**

**IB Systems**

**IB Nodes** (Selected)

IB Ports

Connections

Fabric Topology

System Names

**Node (IB chip) Information**

**Save Changes? Save**

**Nodes - Filter**

**Node Type** Any **System Role** Any

**Apply Filters** **Reset Filters**

System Name	Node	Type	Role	Ports	Description
00:00:2D:90:00:14:3D:17	<a href="#">00:00:2D:90:00:14:3D:14</a>	CA	Host	2	PCI 25418:713
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:D1:28</a>	SW	spine	36	MTS3610 S01:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:D1:38</a>	SW	spine	36	MTS3610 S05:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:CD:E8</a>	SW	leaf	36	MTS3610 L15:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:CD:F0</a>	SW	leaf	36	MTS3610 L18:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:D1:60</a>	SW	spine	36	MTS3610 S06:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:CE:A8</a>	SW	leaf	36	MTS3610 L17:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:5B:C8</a>	SW	leaf	36	MTS3610 L14:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:CE:18</a>	SW	leaf	36	MTS3610 L03:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:CE:30</a>	SW	leaf	36	MTS3610 L08:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:CE:60</a>	SW	leaf	36	MTS3610 L02:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:D0:38</a>	SW	spine	36	MTS3610 S09:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:CE:98</a>	SW	leaf	36	MTS3610 L01:0/U1
00:02:C9:09:99:09:99:98	<a href="#">00:02:C9:02:00:40:5B:98</a>	SW	leaf	36	MTS3610 L16:0/U1

Showing 14 of 14

**Location in switch**



# Fabric Inspector – Ports Page

- Ports Page shows all Infiniband Ports in the cluster.
- Ability to filter out ports types (i.e. internal for external switch system ports) and port rates (link speed and width).
- Ability to filter on packet count levels and error levels.

Setup      System      Security      Ports      Fabric Mgmt      Fabric Inspctr      Status

**Port Information**

**IB Ports**

**Ports - Filter**

Port Traffic	Port Type	Port State
Any <input type="button" value=""/> < packets	Any <input type="button" value=""/> <	Link up <input type="button" value=""/> <

Port Errors	Port Rate	SM Status
Symbol errors > threshold <input type="button" value=""/> < 2 errors	Any <input type="button" value=""/> <	Any <input type="button" value=""/> <

**Show only ports with Symbol errors**

**Apply Filters**    **Reset Filters**

System Name	Type	Port	Description	State	Rate
mtilabxxx	SW	<a href="#">00:02:C9:02:00:40:47:B8-0027</a>	Port 27	Link Up	40 Gbps

Showing 1 of 81

# Fabric Inspector – Connections Page

- Connections Page shows all link pairs in the cluster.
- Ability to filter out link types (i.e. switch-switch, switch-HCA) and link rates.
- Port description today is via GUID. Will add system names in next release.

Setup      System      Security      Ports      Fabric Mgmt      Fabric Inspctr      Status

**Connection Information**

**Connections - Filter**

Connection Type: Switch to switch: Any      Connection Rate: Any

Apply Filters      Reset Filters

Port	Port	Description
00:02:C9:02:00:40:44:D0-0018	00:02:C9:02:00:40:47:B8-0027	Active 4X @ 10 Gbps mtu=4096 VL0 - VL7

Show only switch switch connection

Showing 1 of 8

Display Connection Details

# Fabric Inspector – System Names Page

- System Names page is used to equate GUIDs to System Names.
- Cluster can be scanned in, and then naming relationship can easily be assigned.
- Scanned cluster data uses hostnames if assigned, and the system GUID if no hostname is defined.

**System-name assignments**

System Name to GUID Mapping					
<input type="button" value="Save Changes?"/> <input type="button" value="Save"/>					
System Name Definitions					
<input type="checkbox"/> System Name	Type	Ports	System GUID	Description	Static
<input type="checkbox"/> mtilab14	CA	NA	00:02:C9:02:00:23:12:C7	mtilab14 HCA-2	yes
<input type="checkbox"/> MT25408	CA	1	00:02:C9:02:00:25:99:07	MT25408 ConnectX Mellanox Technologies	yes
<input type="checkbox"/> mtilab225-SharkGT	SW	36	00:02:C9:02:00:40:44:D3	mtilab250 SW-1	yes
<input type="checkbox"/> mtilabxxx	SW	36	00:02:C9:02:00:40:47:BB	Infiniscale-IV Mellanox Technologies	yes
<input type="checkbox"/> mtilab228 - Shark	SW	NA	00:02:C9:02:00:40:52:5B	Infiniscale-IV Mellanox Technologies	yes
<input type="checkbox"/> mtilab40	CA	2	00:02:C9:03:00:00:02:6B	mtilab40 HCA-1	yes
<input type="checkbox"/> mtilab42	CA	2	00:02:C9:03:00:00:03:1F	mtilab42 HCA-1	yes
<input type="checkbox"/> mtilab31	CA	2	00:02:C9:03:00:00:05:7F	mtilab31 HCA-1	yes
<input type="checkbox"/>	CA	2	00:02:C9:03:00:02:13:BB	HCA-1	yes
<input type="checkbox"/>	CA	2	00:02:C9:03:00:02:14:2B	HCA-1	yes
<input type="checkbox"/> mtilab14	CA	NA	00:02:C9:03:00:03:2D:47	mtilab14 HCA-1	yes
<input type="checkbox"/> mtilab13	CA	NA	00:02:C9:03:00:03:36:FB	mtilab13 HCA-1	yes
<input type="checkbox"/> MT25408	CA	1	00:02:C9:03:00:11:04:17	MT25408 ConnectX Mellanox Technologies	yes

**Save changes**

**Load from cluster**

# CLI - Fabric Inspector

- Equivalent of all shown GUI operations can be done through the CLI.
- The root commands to display the meta-data, or variables that show up on the GUI summary screen is 'show ib fabric monitor'
  - show ib fabric monitor unique-GUIDs
    - Display the total number of unique system, node, and port GUIDs
  - show ib fabric monitor snapshot-time
    - Display date/time of the active topology data set
  - show ib fabric monitor warnings
    - Display the number of errors/warnings in the snapshot
  - show ib fabric monitor active-links
    - Display the number of active connections
  - show ib fabric monitor active-ports
    - Display the number of ports that are LINK\_UP
  - show ib fabric monitor nodes
    - Display the number of IB chips in the fabric
  - show ib fabric monitor systems
    - Display the number of systems in fabric.
  - show ib fabric monitor host-ports
    - Display the number of active HCA ports

# CLI - Fabric Inspector (cont.)

- Command 'ib fabric refresh' will sweep the fabric and update cluster information.
- Commands that deal with systems (unique system image GUIDs)
  - show ib fabric system <##:#:#:#:#:#:#:#:#:#> {ports | nodes}
    - Display details on system with GUID given. If 'ports' or 'nodes' display one line list of ports or chips. You are able to use a nodename instead of ##:#:#.... as well.
  - show ib fabric sys {type {switch | host | router | unknown}} | {config {multi-chip | single-chip | MTS3600 | MTS3610}}
    - Show list of systems that pass filters. You can use both, either, or none.
- Commands that deal with nodes (unique node GUIDs)
  - show ib fabric node ##:#:#:#:#:#:#:#:#:# {ports}
    - Display details about the node with the given GUID. If 'ports' is added, display a one line list of ports.
  - show ib fabric nodes {type {switch | host | router | unknown}} | {role {multi-chip | single-chip | leaf | spine | MTS3600 | MTS3610}}
    - Show list of nodes that pass filters. You may use both, either, or none.

# CLI - Fabric Inspector (cont.)

- Command that shows messages (errors or warning about a fabric snapshot).
  - show ib fabric messages
    - Display errors and warnings about a fabric snapshot.
- Command that shows connections
  - show ib fabric connections {type {<options>}} {attrib {<options>}} {details}
    - Display filtered list of connections. Use '?' to see the various <options>. The details flag will do 3 lines per-connection of details.
- Command that shows ports
  - show ib fabric ports {type {<options>}} {attrib {<options>}} {details}
    - Display filtered list of ports. Use '?' to see the various <options>.

# Cluster Bring-up with FabricIT



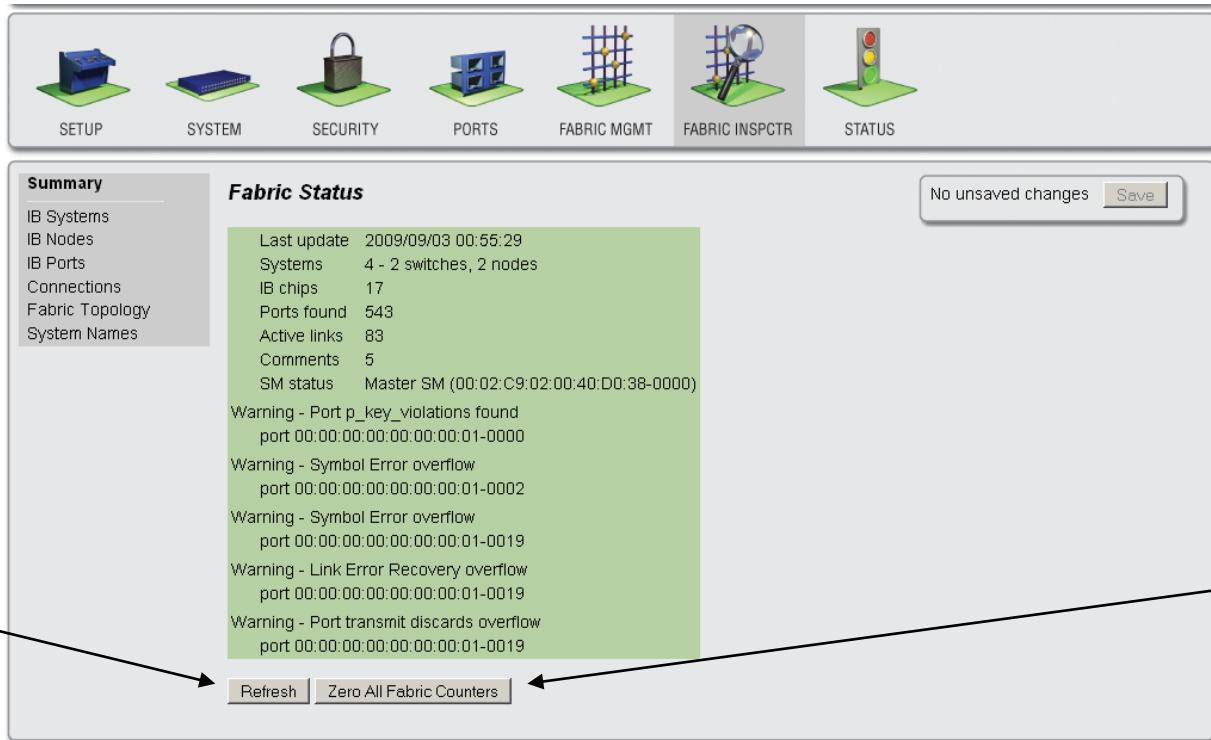
CONFIDENTIAL

# Cluster Bring-up Steps

- Steps to verify cluster is running free of physical errors, such as bad cable connections, is an important step in verifying proper operation of the cluster.
- FabricIT has a number of useful utilities to aid in this. The following steps outline a methodology for this and will concentrate on the following steps:
  - 1. Verify that cluster connectivity.
  - 2. Run initial diagnostics and verify that the fabric is error free in a static idle state.
  - 3. Run stress traffic to assure all links in fabric are properly stressed with heavy data usage.
  - 4. Run diagnostics on traffic under this stressed state.
    - In general steps 3 and 4 can be an iterative process where heavy traffic is run on the cluster, or on a subset of the cluster and problem areas are identified and fixed, until the fabric is running error-free.

# Step 1: Verify Cluster Connectivity

- The first step is to verify the proper connectivity of the cluster and to make sure that all of the links in the cluster are running with the proper rate.
- Use Fabric Inspector Utilities in FabricIT for this task:
  - Step 1. Enter the Fabric Inspector page and scan the fabric by clicking on the Refresh tab.



The screenshot shows the Fabric Status page of the FabricIT interface. The top navigation bar includes icons for SETUP, SYSTEM, SECURITY, PORTS, FABRIC MGMT, FABRIC INSPCTR (which is selected), and STATUS. On the left, a sidebar titled "Summary" lists options: IB Systems, IB Nodes, IB Ports, Connections, Fabric Topology, and System Names. The main content area displays "Fabric Status" with the following information:

Last update	2009/09/03 00:55:29
Systems	4 - 2 switches, 2 nodes
IB chips	17
Ports found	543
Active links	83
Comments	5
SM status	Master SM (00:02:C9:02:00:40:D0:38-0000)

Below this, a green box contains several warning messages:

- Warning - Port p\_key\_violations found port 00:00:00:00:00:00:01-0000
- Warning - Symbol Error overflow port 00:00:00:00:00:00:01-0002
- Warning - Symbol Error overflow port 00:00:00:00:00:00:01-0019
- Warning - Link Error Recovery overflow port 00:00:00:00:00:00:01-0019
- Warning - Port transmit discards overflow port 00:00:00:00:00:00:01-0019

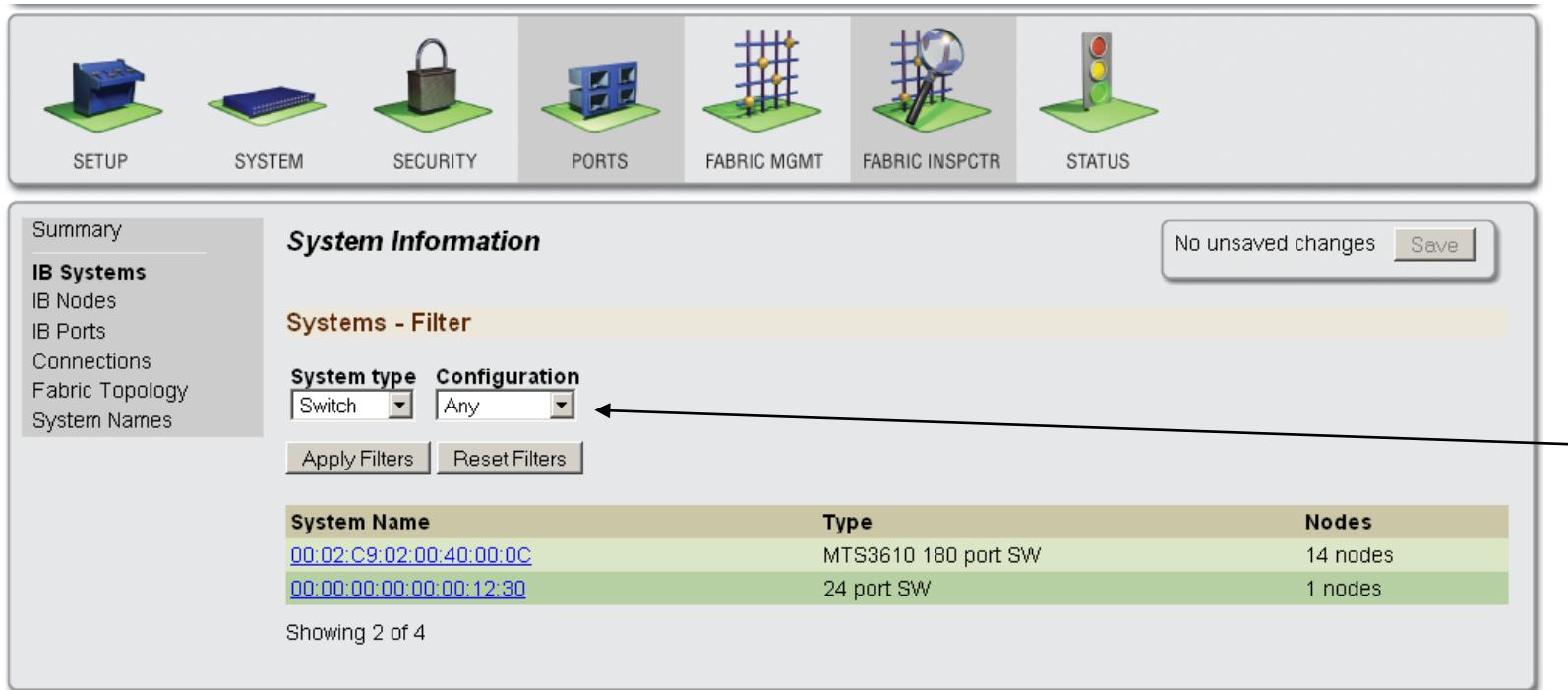
At the bottom of the page are two buttons: "Refresh" and "Zero All Fabric Counters". A callout box points to the "Scan fabric" button with the text "Scan fabric" and another callout box points to the "Zero All Fabric Counters" button with the text "Clear counters".

# Step 1: Verify Cluster Connectivity (cont.)

- Once scanned, a high-level status is displayed in the window, which includes:
  - the number of Systems (including switches and end-nodes), number of separate Infiniband devices
  - number of ports
  - number of Active links (an Active link means the link is enabled to transport user data)
- Zero All Fabric Counters tab resets all port counters across all nodes on the cluster.
  - Should be used if end-nodes have been reset, or if cables are being moved around.

# Step 2: Check Cluster Components Present

- Next step is to use Fabric Inspector IB Systems page to make sure all of the switch systems and end-nodes are detected and on-line.
- Fabric Inspector includes powerful filtering techniques which allow the administrator to quickly narrow relevant information necessary for cluster debug.
- One simple technique shows only Switch Systems for checking that all are present.



The screenshot shows the Mellanox Fabric Inspector interface. At the top, there is a navigation bar with icons for SETUP, SYSTEM, SECURITY, PORTS, FABRIC MGMT, FABRIC INSPECTOR, and STATUS. Below the navigation bar, on the left, is a sidebar with links for Summary, IB Systems, IB Nodes, IB Ports, Connections, Fabric Topology, and System Names. The main content area is titled "System Information" and contains a "Systems - Filter" section. This section includes dropdown menus for "System type" (set to "Switch") and "Configuration" (set to "Any"), along with "Apply Filters" and "Reset Filters" buttons. To the right of the filter section is a message "No unsaved changes" with a "Save" button. The main table displays two entries:

System Name	Type	Nodes
<a href="#">00:02:C9:02:00:40:00:0C</a>	MTS3610 180 port SW	14 nodes
<a href="#">00:00:00:00:00:12:30</a>	24 port SW	1 nodes

At the bottom left of the main content area, it says "Showing 2 of 4".

**Filters**

# System Names Utility

- FabrILT includes ability to show all systems with their system name instead of System GUIDS.
- GUID to System Names is done through the System Names page.
- First populate table by reading in all names information from cluster, and then modify names for usability.

Setup      System      Security      Ports      Fabric Mgmt      Fabric Inspect      Status

**System Name to GUID Mapping**

**System Name Definitions**

<input checked="" type="checkbox"/>	System Name	Type	Ports	System GUID	Description	Static
<input checked="" type="checkbox"/>	MT47396	SW	24	00:00:00:00:00:00:12:30	MT47396 Infiniscale-III Mellanox Technologies	no
<input checked="" type="checkbox"/>	sw006	CA	1	00:00:00:00:00:12:30:03	sw006 HCA-1	no
<input checked="" type="checkbox"/>	sw005	CA	2	00:02:C9:00:01:00:D0:50	sw005 HCA-1	no
<input checked="" type="checkbox"/>	switch-112478	SW	180	00:02:C9:02:00:40:00:0C	switch-112478	no

**Save Changes? Save**

**System Names**

**Import Names**

Accept Selected | Remove Selected | Import Fabric SysNames | Remove Imported SysNames

Add additional system name to GUID mapping

GUID:   
System Name:

Add/Rename Entry

## Step 3: Verify Cluster Ports Status

- Once all systems are verified to be present, next logical step is to make sure that all of the ports connected to other end-points are Up and at the proper link width and speed expected.
- This is done through the Fabric Inspector IB Ports page.
- Use filters:
  - The first filter should be to check that all of the ports are Up that are expected to be up.
  - Verify all links are expected link width (usually 4x) and the proper speed (DDR for 20Gb/s or QDR for 40 Gb/s). This can be done by using the Port Rate filter.
  - If a port that is supposed to be Up is not, or if the rate of the port is not as expected, please check that both ends of the link are running, or replace/re-seat the cable and re-test. Remember, whenever some status in the cluster has changed, like changing the cable for instance, the Fabric Inspector must be refreshed as was done in Step 1 of this section.

# Step 3: Verify Cluster Ports Status (cont)

- If port is not Up, or if the rate of the port is not as expected, please check that both ends of the link are running, or replace/re-seat the cable and re-test.
- **IMPORTANT REMINDER:** whenever some status in the cluster has changed, like changing the cable for instance, the Fabric Inspector must be refreshed as was done in Step 1 of this section.

**Mellanox FabricIT MTS3610 Management Console**

Host: switch-112478 User: admin [Logout](#)

**Mellanox TECHNOLOGIES**

SETUP SYSTEM SECURITY PORTS FABRIC MGMT FABRIC INSPECTR STATUS

**Port Information**

**Ports - Filter**

Port Traffic: Any packets   Port Type: Switch: External   Port State: Link up   SM Status: Any

Port Errors: Any errors   Port Speed: Any   Port Width: Any

Sort order: none

Save Changes? Save

Check for ports with Errors

Only Ports with Link Up

System Name	Type	Port	Description	State	Rate
switch-112478	SW	00:02:C9:02:00:40:CE:30-0015	L08-15	Link Up	10 Gbps
MT47396	SW	00:00:00:00:00:00:01-0001	Port 1	Link Up	10 Gbps
MT47396	SW	00:00:00:00:00:00:01-0002	Port 2	Link Up	10 Gbps
MT47396	SW	00:00:00:00:00:00:01-0019	Port 19	Link Up	10 Gbps

Showing 4 of 545

## Step 4: Verify Links run Error Free (cont.)

- Next verify all ports counters are error free with no excessive bit errors under heavy stress traffic.
- Run MPI across a subset of nodes. Use benchmark that has collective operations, such as Intel MPI Benchmark (formally known as Pallas).
- It is recommended to run this for an hour to properly stress the cluster. Steps are:
  - Reset all of the port counters
  - Run MPI benchmarks
  - Once benchmark completes, rescan the fabric and check for symbol errors.
  - Correct any errors that are found by reseating cables, and/or swapping out problem cables or hardware.
    - *Hint: To isolate problems change one end of the cable and see if the problem follows the cable or stays with the port.*
  - Run above steps iteratively until reaching an acceptable number of errors across the fabric.

# FabricIT Questions



CONFIDENTIAL

# Questions

1. What is the difference between FabricIT Chassis Manager and Embedded Fabric Manager (EFM)?
2. Which if any of the above two modules require a purchased license to enable?
3. What key is used to obtain help from the CLI command?
4. What command is used from the cli to see the IP address of eth0 Ethernet interface?
5. Which CLI command is used to show the FRU information of all modules in the system?
6. Which CLI command is used to show the temperature of a module in the system?
7. What are the steps to upgrading the software of FabricIT? This should be in general terms and applies to the CLI or WebGUI.

# FabricIT Questions (cont.)

8. Which main WebGUI tab is used to control the Subnet Manager that is part of EFM?
9. The customers Fabric Management and Fabric Inspector tabs are grayed out and cannot be accessed. What is the most likely cause of this?
10. From the WebGUI how do you clear out all port counters in the cluster?

# FabricIT Hands-on Exercises



CONFIDENTIAL

# FabricIT Hands-On Exercises

- 1. Log into the chassis from a Linux shell.**
  1. Determine that you can go to the 'configure terminal' sub-menu in the CLI
  2. Get a list of commands available in this menu
  3. Show the status of Infiniband Port 5 from this menu
- 2. From the CLI read the voltage of the power supply units on your switch chassis.**
- 3. From the CLI determine the version of firmware running on the devices in your chassis.**
- 4. Log into FabricIT WebGUI.**
  1. How long has your system been up and running?
  2. What is the version of FabricIT running on your system?
  3. Are there any licenses installed on your system
  4. Is the ib0 IPoIB interface configured on your system. If not, configure this and make sure you can ping into FabricIT from an external interface over the Infiniband subnet.
- 5. Determining SM usage**
  1. which nodes are running the SM in your fabric and which SM is Master.
  2. Turn off any host based SMs
  3. Enable the SM within FabricIT Give it a priority of 15.
  4. Verify FabricIT SM is not Master.

# FabricIT Hands-On Exercises

6. Using Fabric Inspector, determine how many switch devies and HCA devices reside in the cluster.
7. Using the Main Ports page determine how many Active links are part of this switch.
8. Using the Fabric Inspector Ports page determine the same information. What are some important differences between the Main Ports page and the Fabric Inspector Ports page?
9. Check that all ports in the fabric are 4x? What speed are the ports?
10. Run an MPI Pallas benchmark across all HCA devices connected in the fabric.
  1. Clear the counters before the run.
  2. After the run, how many packets have been received on the ports that were part of the job?
  3. Are there any symbol errors on any of the ports? (Did you refresh the Fabric Inspector database before checking for errors?)

# Thank You

[www.mellanox.com](http://www.mellanox.com)

