

# Mellanox Infiniband Foundations



- Founded in 1999
- Actively markets and promotes InfiniBand from an industry perspective through public relations engagements, developer conferences and workshops
- Steering Committee Members:



# InfiniBand is a Switch Fabric Architecture

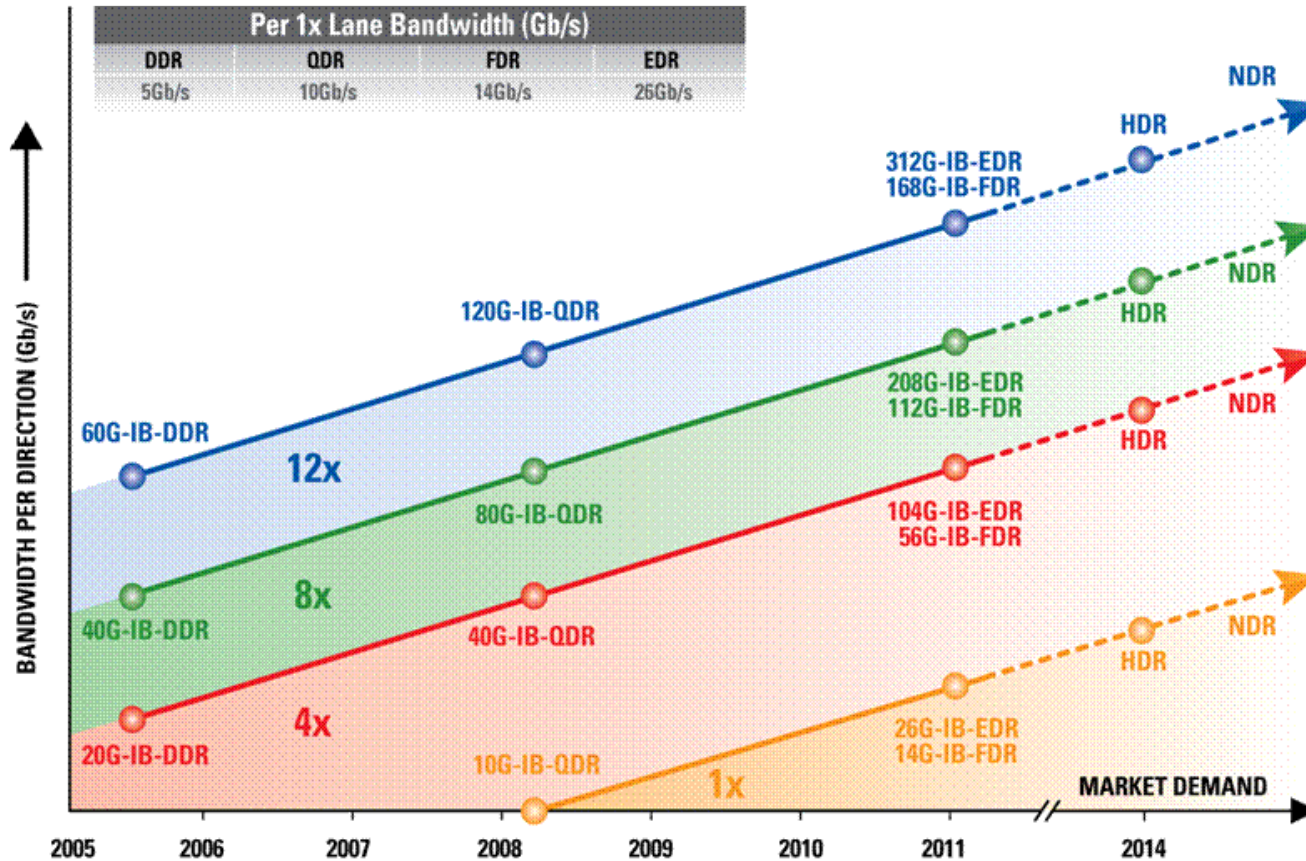


- ▶ Interconnect technology connecting CPUs and I/O
- ▶ Super high performance
  - High bandwidth (starting at 10Gbps and up to 60Gbps) – *Lots of head room!*
  - Low latency – Fast application response across the cluster.
  - Low CPU Utilization with RDMA (Remote Direct Memory Access) – Unlike Ethernet, communication bypasses the OS and the CPU' s.
- ▶ Increased application performance
- ▶ Single port solution for all LAN, SAN, and application communication
- ▶ High reliability Subnet Manger with redundancy
- ▶ InfiniBand is a technology that was designed for large scale grids and clusters



**First industry standard high speed interconnect!**

# InfiniBand Roadmap



SDR - Single Data Rate  
 DDR - Double Data Rate  
 QDR - Quad Data Rate  
 FDR - Fourteen Data Rate  
 EDR - Enhanced Data Rate  
 HDR - High Data Rate  
 NDR - Next Data Rate

- InfiniBand software is developed under OpenFabrics Open Source Alliance

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

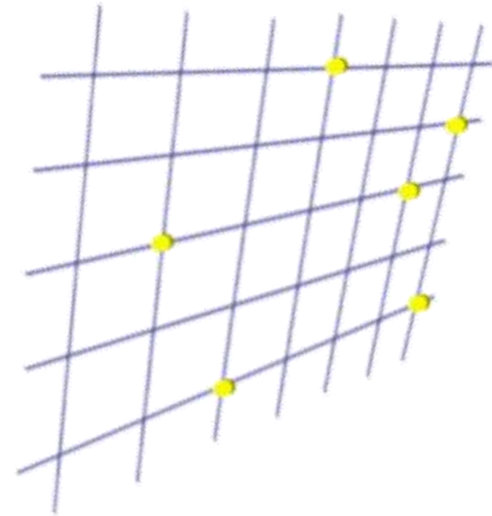
- InfiniBand standard is developed by the InfiniBand Trade

<http://www.infinibandta.org/home>



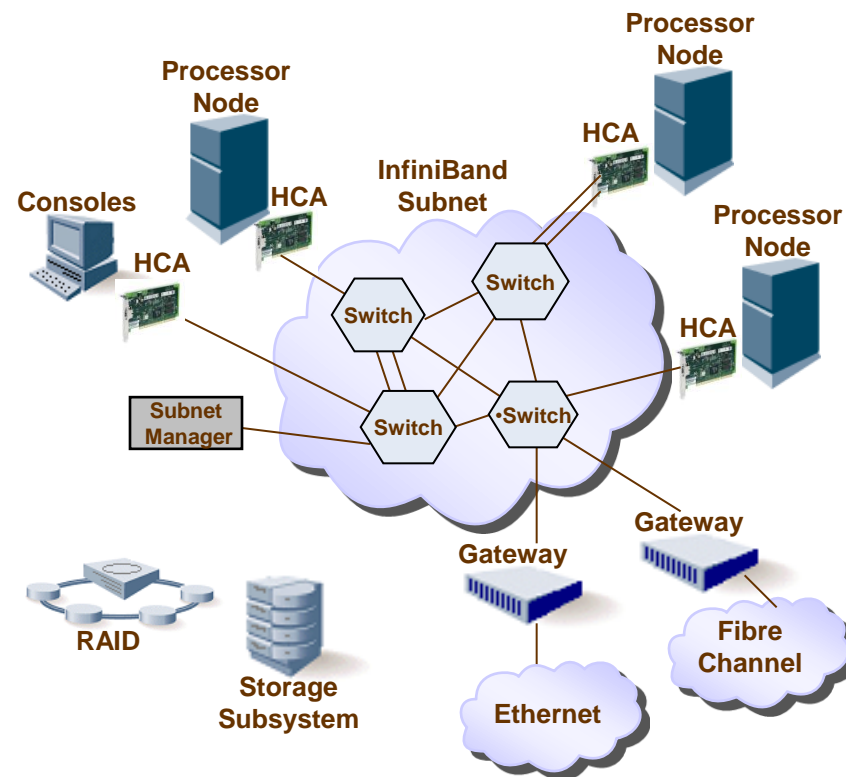


- **Use off-the-shelf components**
- **Based on open standards**
- **Manageable**
- **Scalable**
- **Provide access to storage systems and external networks**



- Industry standard defined by the InfiniBand Trade Association
  - Originated in 1999
- InfiniBand™ specification defines an input/output architecture used to interconnect servers, communications infrastructure equipment, storage and embedded systems
- InfiniBand is a pervasive, low-latency, high-bandwidth interconnect which requires low processing overhead and is ideal to carry multiple traffic types (clustering, communications, storage, management) over a single connection.
- As a mature and field-proven technology, InfiniBand is used in thousands of data centers, high-performance compute clusters and embedded applications that scale from small scale to large scale

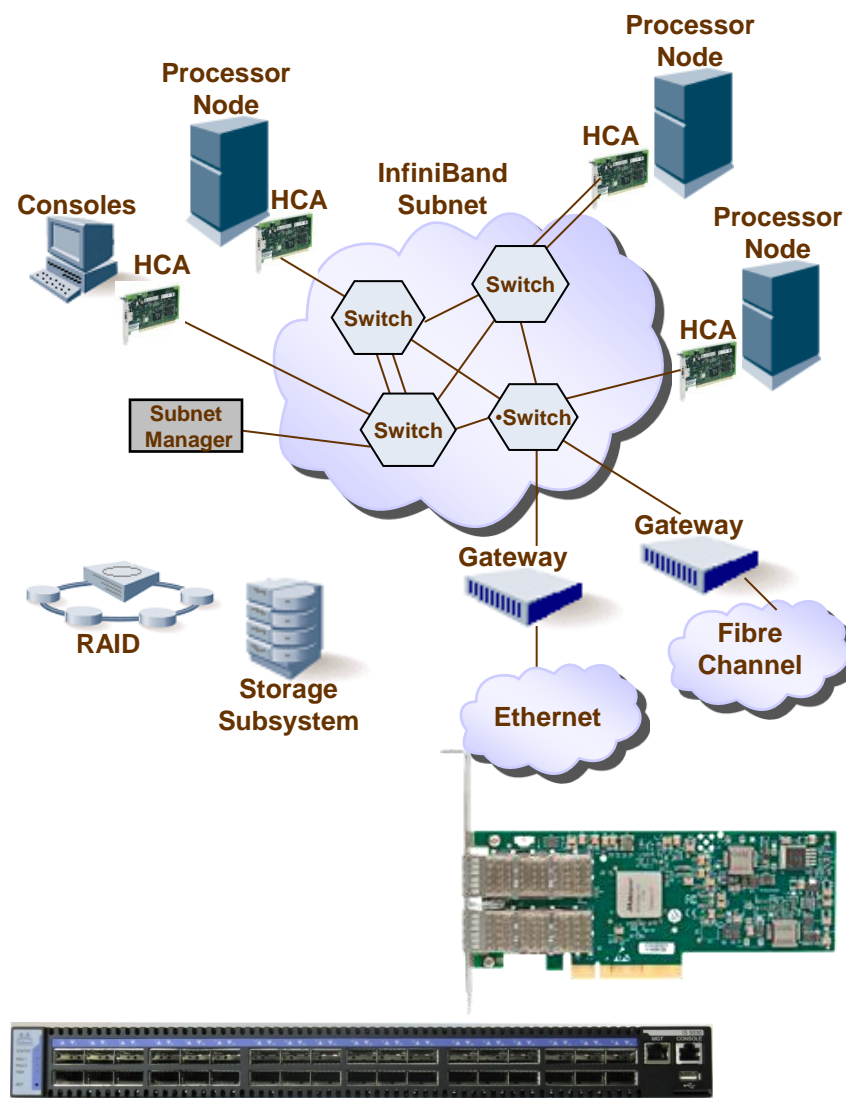
- Industry standard defined by the InfiniBand Trade Association
- Defines System Area Network architecture
  - Comprehensive specification: from physical to applications
- Architecture supports
  - Host Channel Adapters (HCA)
  - Switches
  - Routers
- Facilitated HW design for
  - Low latency / high bandwidth
  - Transport offload





# InfiniBand Components Overview

- 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
  - A device that transports packets between different IBA subnets
- Bridge
  - InfiniBand to Ethernet



- **Equivalent to a NIC (Ethernet)**
  - GUID (Global Unique ID = MAC)
- **Converts PCI to InfiniBand**
- **CPU offload of transport operations**
- **End-to-end QoS and congestion control**
- **Communicate via Queue Pairs (QPs)**
- **HCA Options:**
  - **Single**      **Data Rate**      **2.5GB/S \* 4 = 10**
  - **Double**     **Data Rate**      **5 GB/S \* 4 = 20**
  - **Quadruple** **Data Rate**    **10GB/S \* 4 = 40**
  
  - **Fourteen**   **Data Rate**      **14 Gb/s \* 4 = 56**
  - **FDR 10**                      **10Gb/s \*4 = 40**



# HCA Physical Address Global Unique Identifier (GUID)

Host Channel Adapters (HCA's) & all Switches require GUID & LID addresses

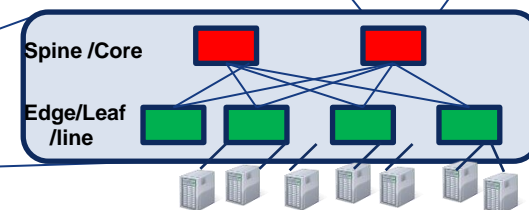
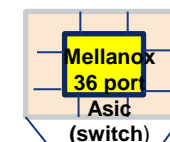
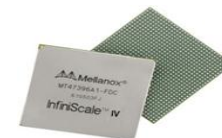
## ■ GUID - 64 bit

Global Unique Identifier “Like a Ethernet MAC address”

- Assigned by IB Vendor
- Persistent through reboots

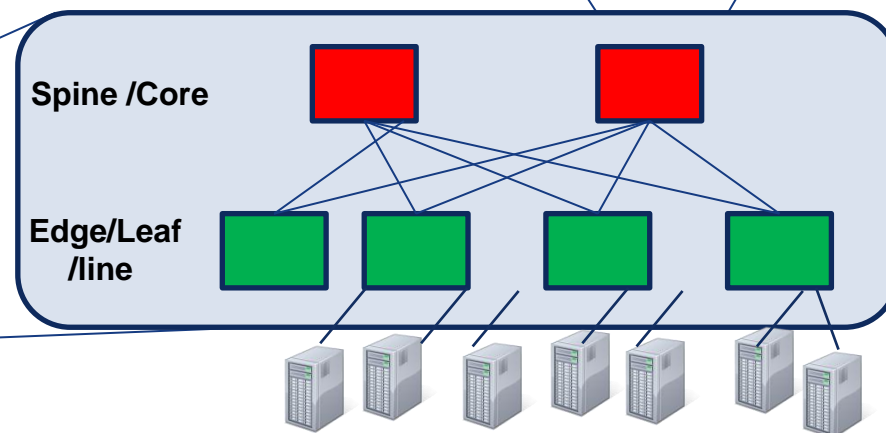
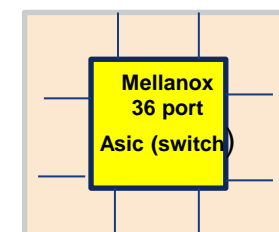
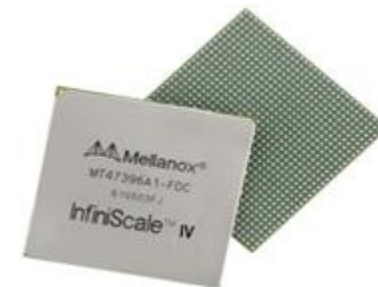
## 3 Types of Guid's per ASIC

- Node = Is meant to identify the HCA as a entity
- Port = Identifies the Port as a port
- System = Allows to combine multiple GUIDS creating one entity



# The IB Fabric Basic Building Block

- A single 36 ports IB switch chip , is the Basic Block for every IB switch Module
- We create A multiple ports switching Module using Multiple chips
- In this Example we create 72 ports switch , using 6 identical chips
  - 4 chips will function as **lines**
  - 2 chips will function as core **core**



# IB Fabric L2 Switching Addressing

## Local Identifier (LID)

Host Channel Adapters (HCA's) & Switches all require GUID & LID addresses

- **LID - 16 bit**

Local Identifier “Like a dynamic IP address”

- Assigned by the SM when port becomes active
- Not Persistent through reboots
- Address ranges

0x0000 = reserved

0x0001 = 0xBFFF = Unicast

0xc001 = 0xFFFE = Multicast

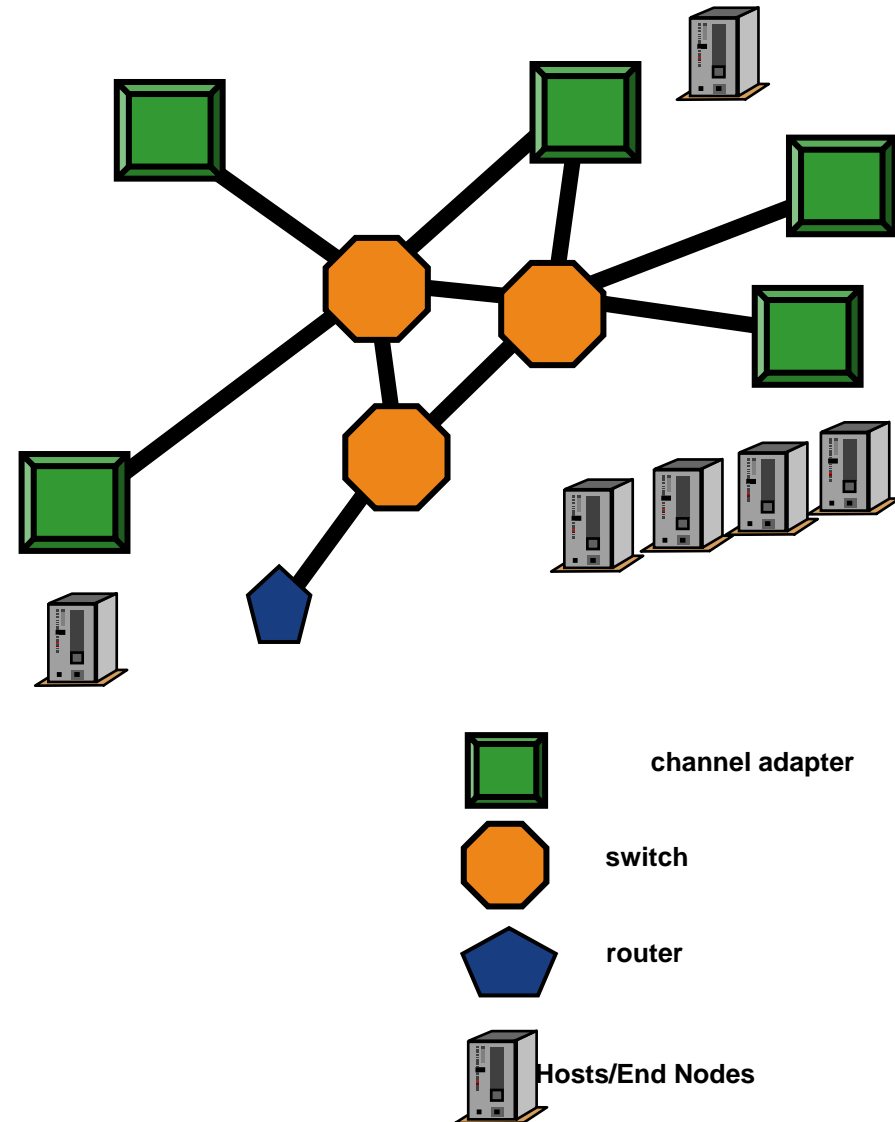
0xFFFF = Reserved for special use



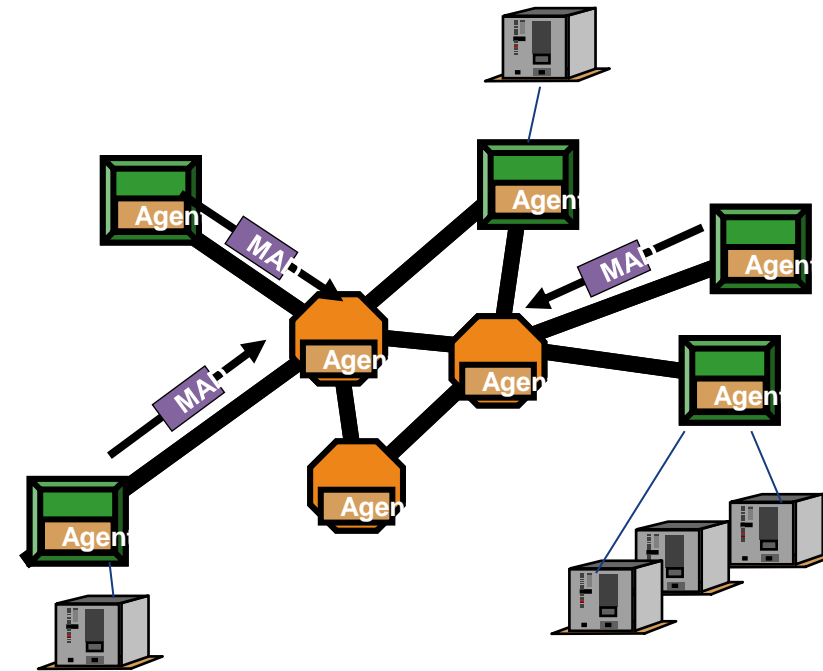


# Refresh - What is a Fabric ?

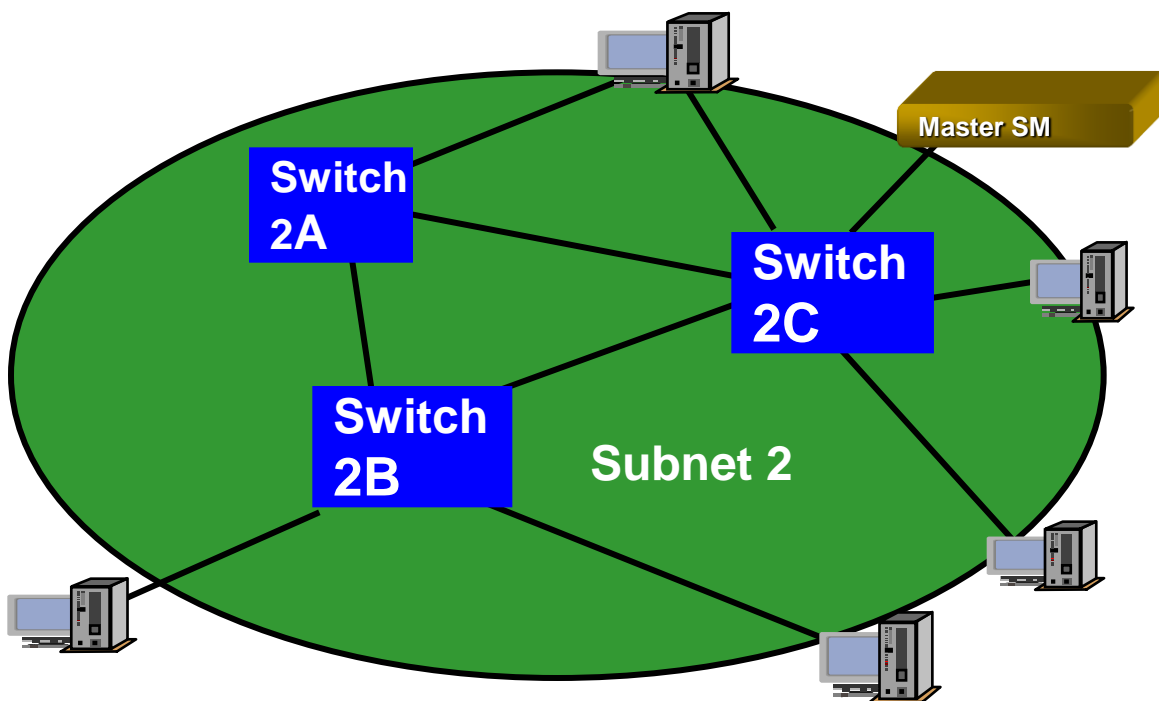
- Switching fabric is the combination of hardware and software that moves data coming in to a network node out by the correct port (door) to the next node in the network.
- Switching fabric includes the switching units (individual boxes) in a node, the integrated circuits that they contain, and the programming that allows switching paths to be controlled. The switching fabric is independent of the bus technology and



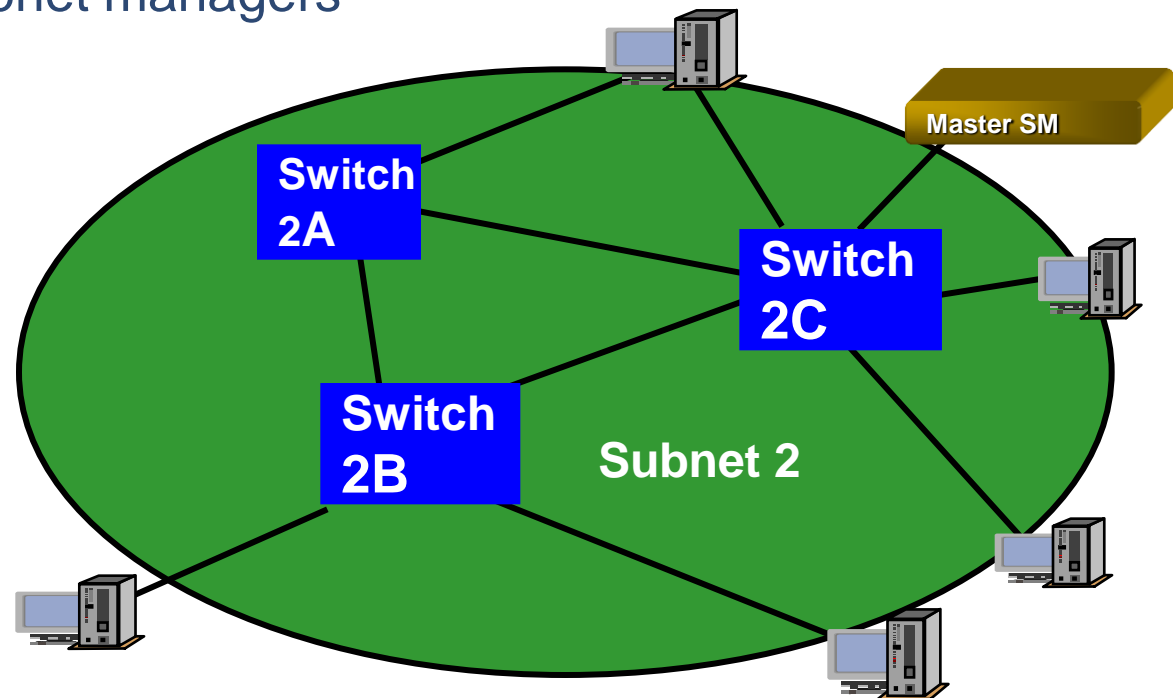
- Node: any managed entity - Endnode, switch, router
- Manager: active entity; sources commands and queries. There are few managers.
- Agent: passive (mostly) entity, responds to managers (but can source traps). Many agents.
- Management Datagram (MAD): standard message format for manager-agent communication. Carried in an unreliable datagram (UD).
- ✓ All data formats & actions are defined solely in terms of MAD content. Implementation not defined: hardware, firmware, software, whatever...



- Subnet = HCAs and interconnected through switches
- Each subnet has its own LID space
- Each subnet has at least one SM and exactly one (logical) Master SM

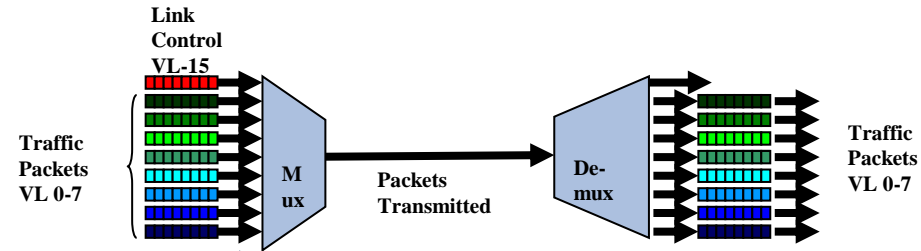


- Initialization and configuration of the subnet elements
- Establishing paths through the subnet
- Fault isolation
- Continue these activities during topology changes
- Prevent unauthorized subnet managers



## IB Port Basic Identifiers

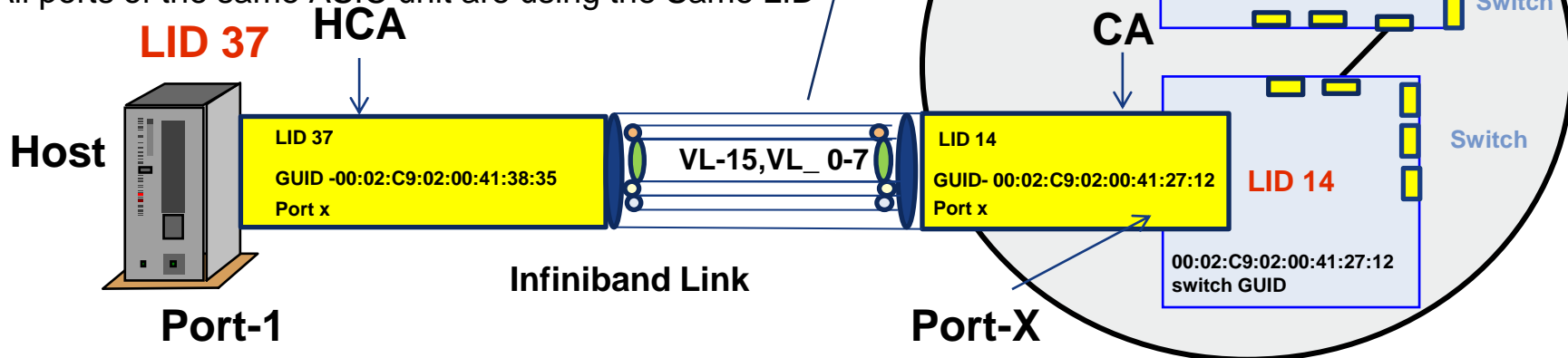
- Port number
- Host Channel Adapter – HCA (IB “NIC”)
- Global universal id – GUID 64 bit ( like mac )  
ex. 00:02:C9:02:00:41:38:30
  - Each 36 ports “basic “ switch has its own switch & system GUID
  - All ports belong to the same “basic “ switch will share the switch GUID



- Local Identifier - LID
- Virtual Lane –VL Used to separate different Bandwidth & Qos using same physical port

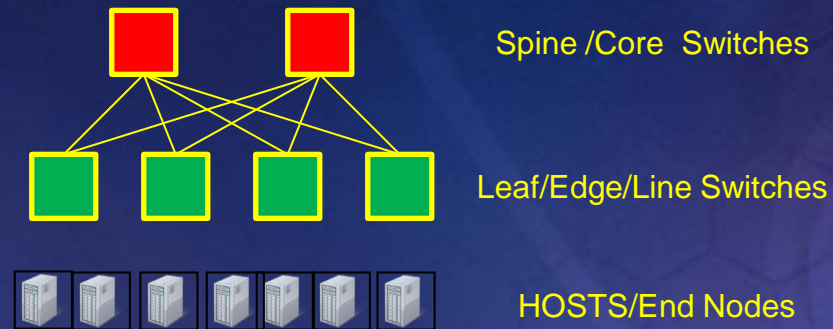
### ■ LID

- Local Identifier that is assigned to any IB device by the SM and used for packets “ routing “ within an IB fabric .
- All ports of the same ASIC unit are using the Same LID

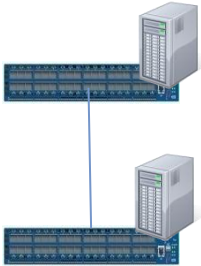




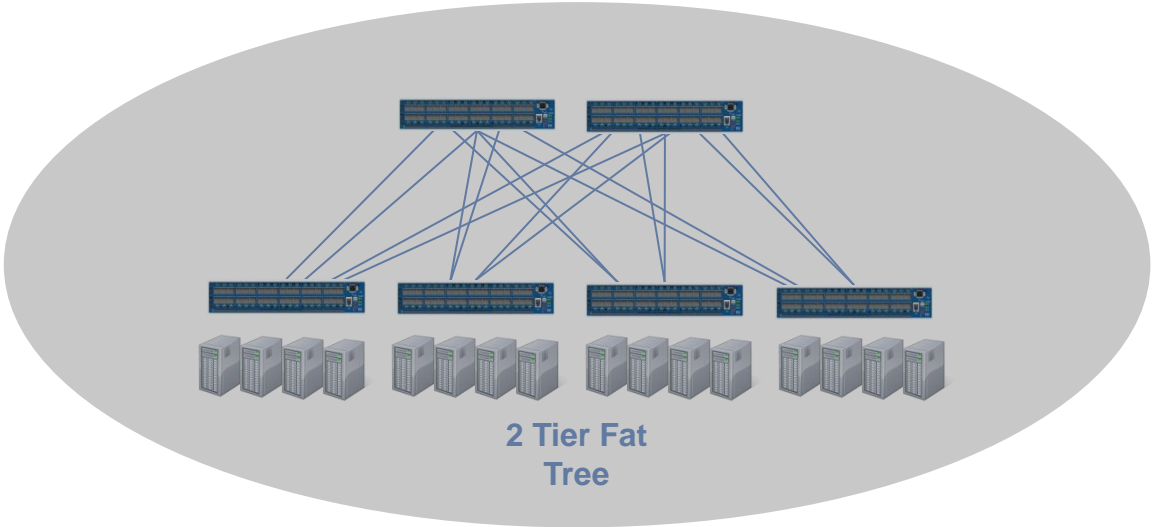
# InfiniBand Fabric Topologies



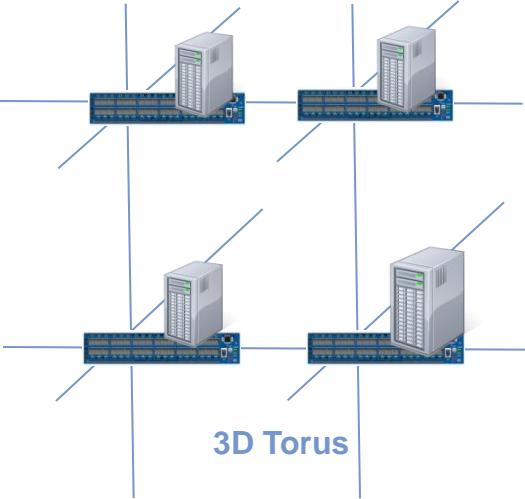
# InfiniBand Fabric commonly used Topologies



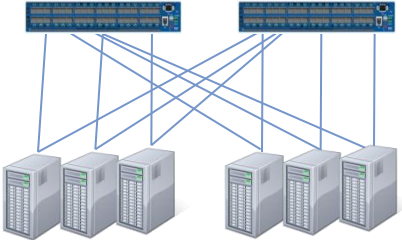
Back to Back



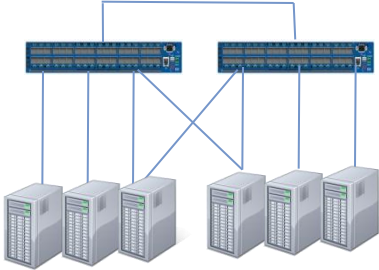
2 Tier Fat Tree



3D Torus



Dual Star

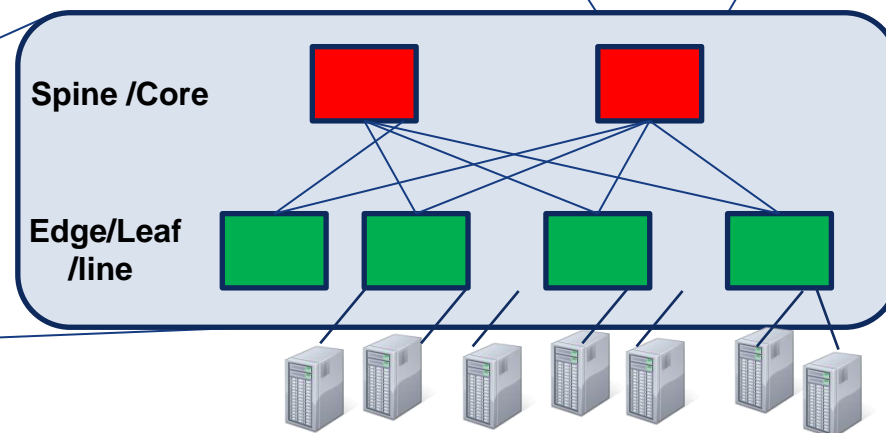
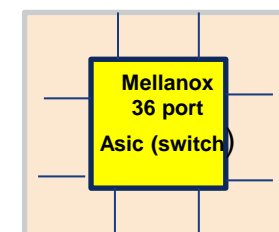
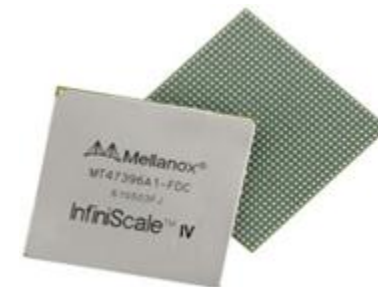


Hybrid

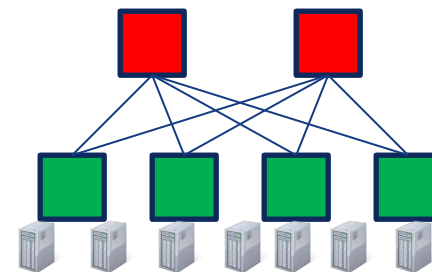
**Modular switches are based on fat tree architecture**

# The IB Fabric Basic Building Block

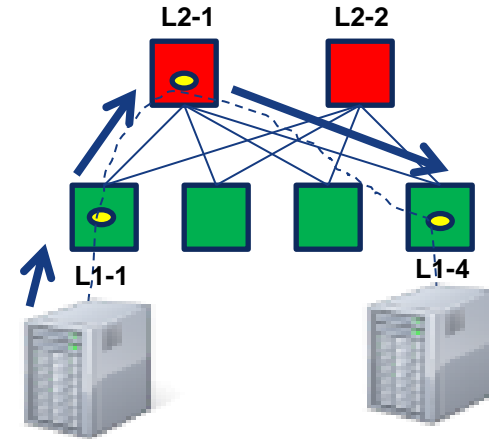
- A single 36 ports IB switch chip , is the Basic Block for every IB switch Module
- We create A multiple ports switching Module using Multiple chips
- In this Example we create 72 ports switch , using 6 identical chips
  - 4 chips will function as **lines**
  - 2 chips will function as core **core**



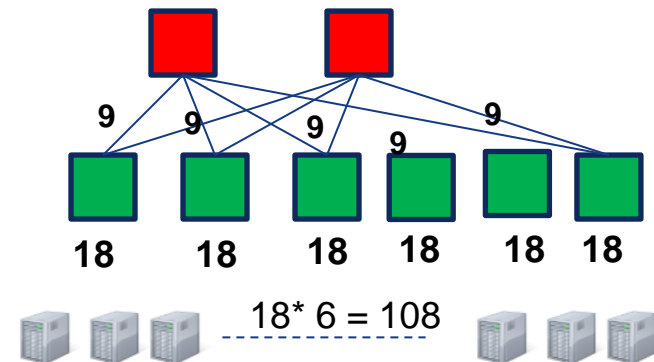
- Pyramid Shape Topology
- The switches at the Top of the Pyramid are called Spines/Core
  - The Core/Spine switches are Interconnected to the Other switch Environments
- The switches at the Bottom of the Pyramid are called Leafs/Lines
  - The Leaf/Lines/Edge are connected to the Fabric Nodes/Hosts
- In A NON Blocking CLOS Fabric there are **Equal Number** of External and internal connections
- External connections :
  - The connections Between the Core and the Line switches
- Internal Connections
  - The Connected of Hosts to the Line Switches
- In a non Blocking Fabric there is always a Balanced Bidirectional Bandwidth
- In Case the Number of Internal Connections is Higher we have Blocking Configuration



- The Topology detailed here is called CLOS 3
- The path between source to Destination includes 3 HOPS
- Example a session between A to B
  - One Hop from A to switch L1-1
  - Next Hop from switch L1-1 to switch L2-1
  - Last Hop from L2-1 to L1-4

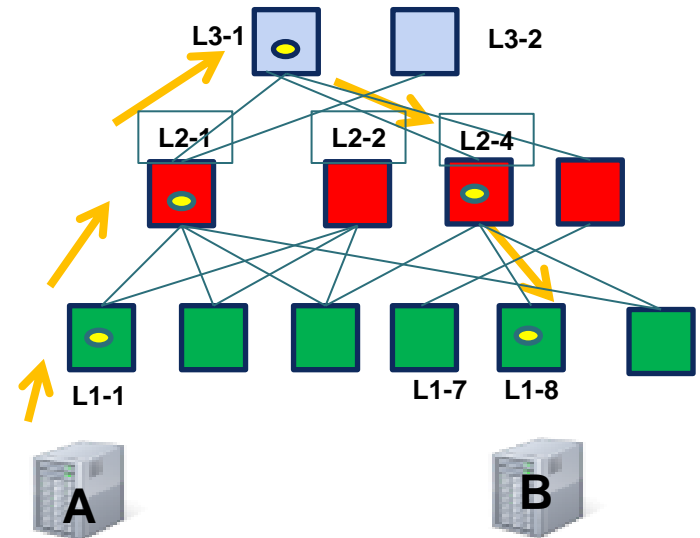


- In this Example we can see 108 Non blocked Fabric
  - 108 Hosts are connected to the Line switches
  - 108 Links connect between the Line Switches To the Core witches to enable Non Blocking Interconnection of the Line switches



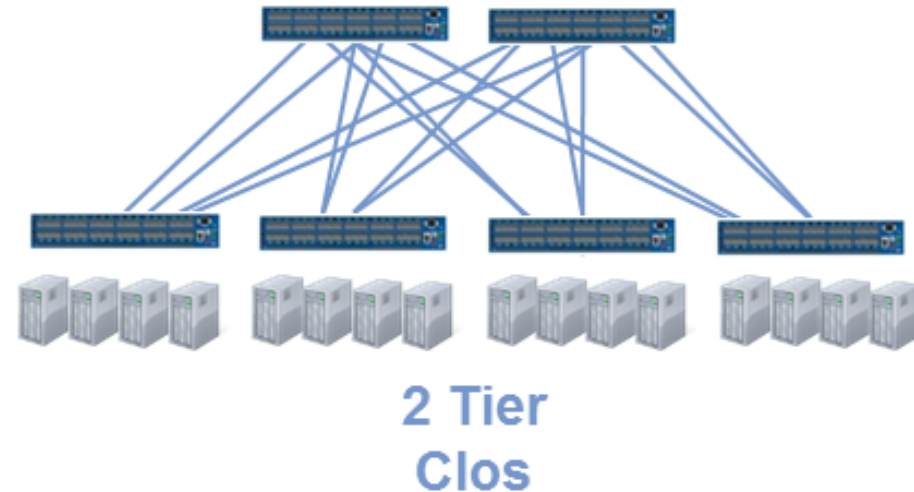


- The Topology detailed here is called CLOS 5
- The path between source to Destination includes 5 HOPS
- Example - a session between A to B
  1. One Hop from A to switch L1-1
  2. Next Hop from switch L1-1 to switch L2-1
  3. Next Hop from L2-1 to L3-1
  4. Next Hop from L3-1 to L2-4
  5. Next Hop from L2-4 to L1-8

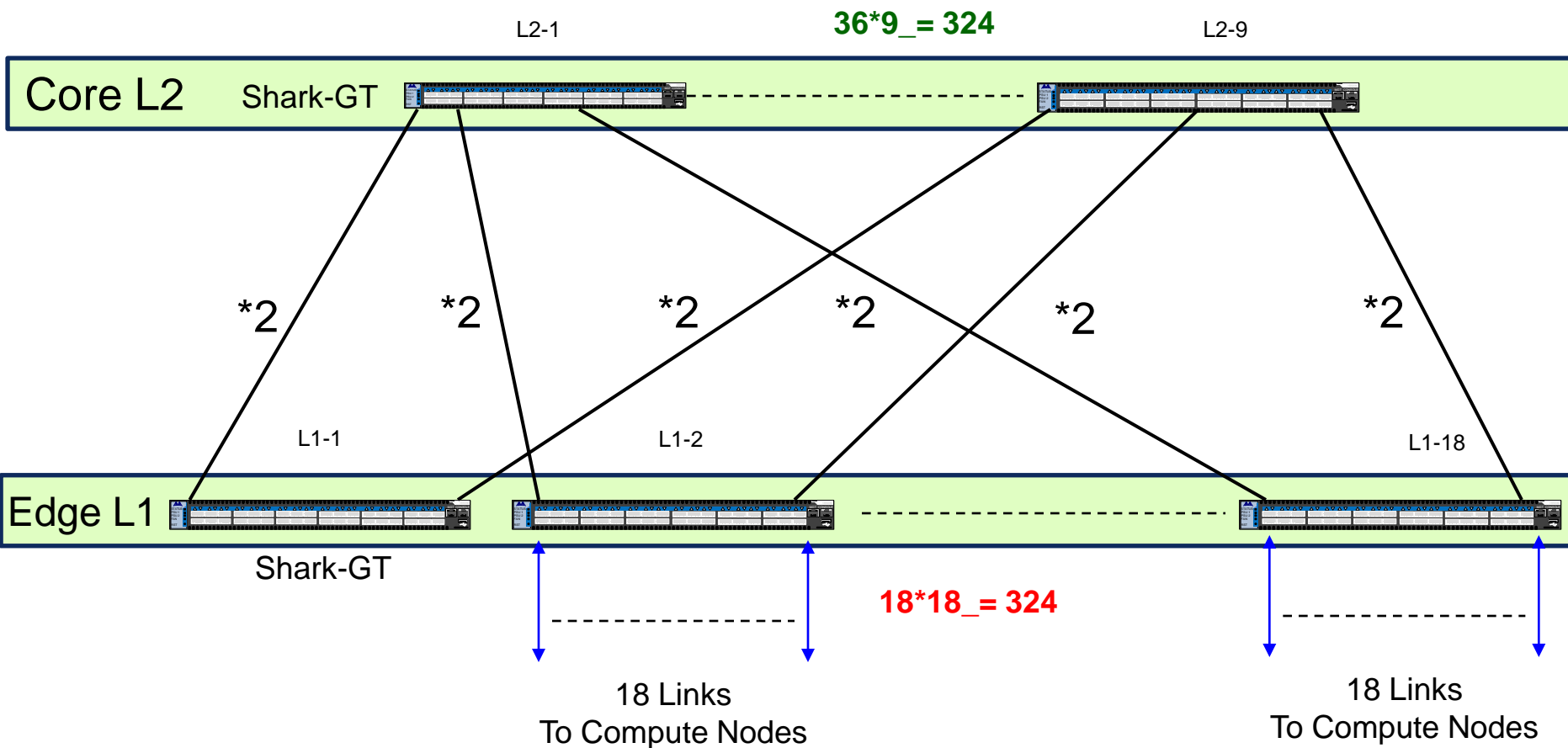


- In most cases use same amount of links to connect L1 switch to all L2
- Port used in all L2 switches for the above connectivity should be identical
- It is advised to have nodes connected only to L1
- To be able to use FAT tree routing algorithm (ftree) fabric should be symmetrical and all switches in L1 should be populated
- Cluster configuration tool can be found on the Mellanox website:

<http://www.mellanox.com/clusterconfig/>

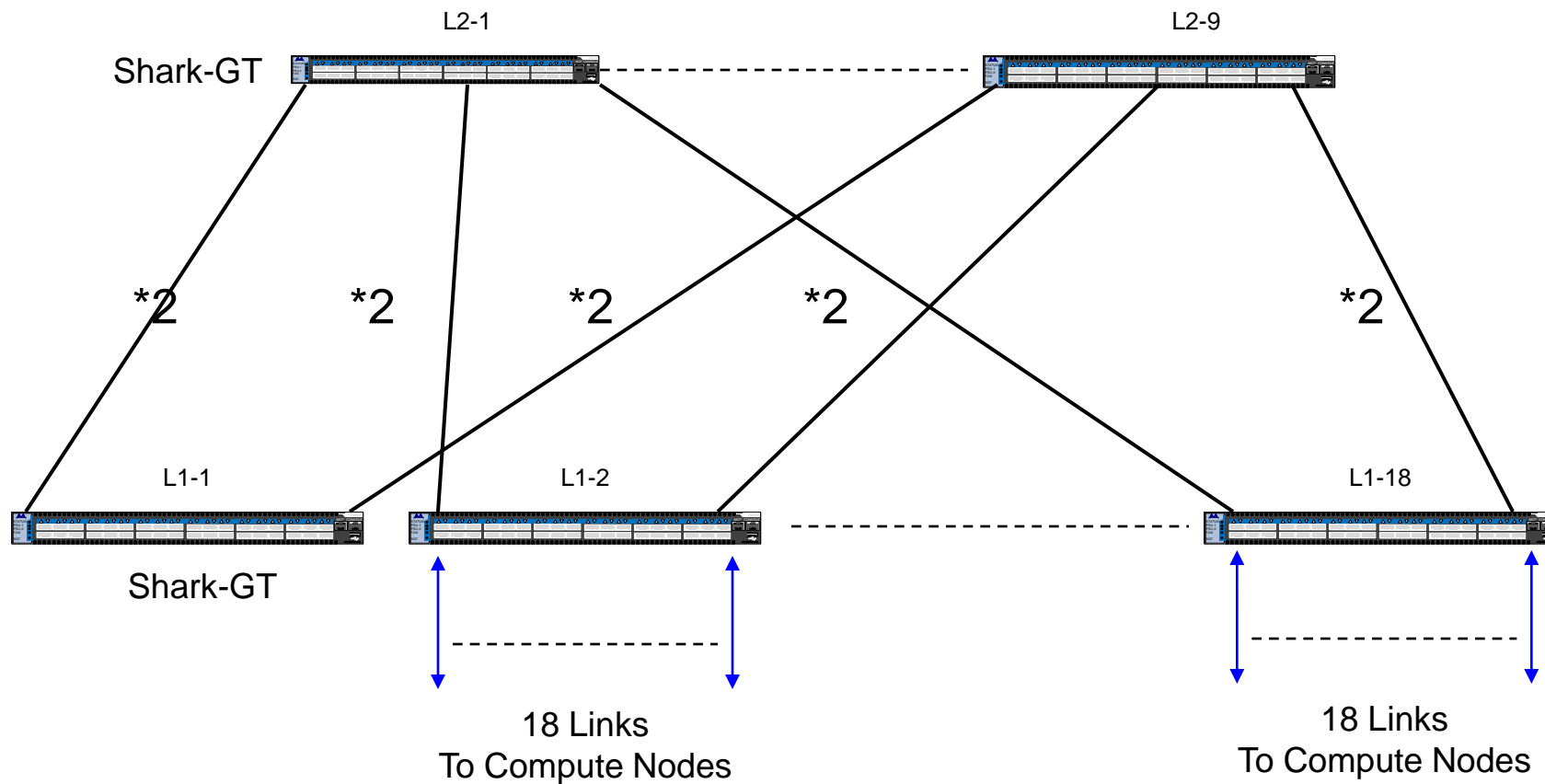


# 324 ports Switch internal Fabric topology



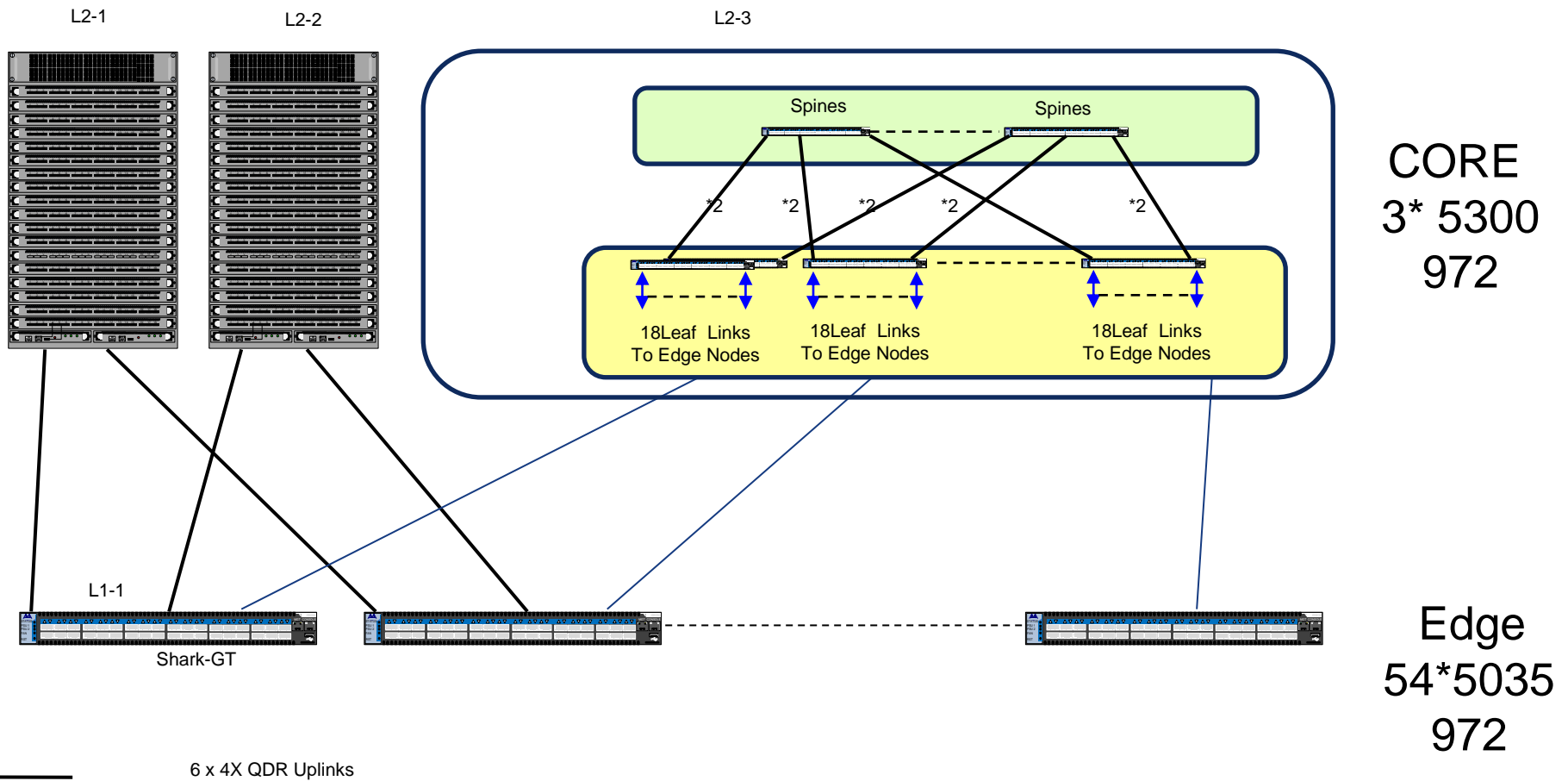
- 2 x 4X QDR Uplinks
- 1 x 4X QDR Uplinks

# 324 Node Full CBB using Shark-GTs



- 2 x 4X QDR Uplinks
- 1 x 4X QDR Uplinks

# How do we create 972 HOSTS Fabric with 5300 Boxes ?





1. Physical subnet establishment
2. Subnet discovery
3. Information gathering
4. LID Assignment
5. Path Establishment
6. Port configuration
7. Switch configuration
8. Subnet activation

Physical Fabric Establish

Subnet Discovery

Information Gathering

Lid Assignment

Path Establishment

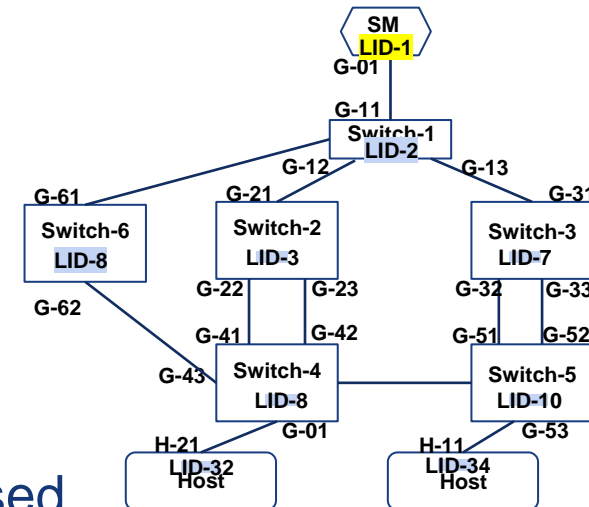
Port Configuration

Switch Configuration

Subnet Activation

- **Every subnet must have at least one**
  - Manages all elements in the IB fabric
  - Discover subnet topology
  - Assign LIDs to devices
  - Calculate and program switch chip forwarding tables (LFT pathing)
  - Monitor changes in subnet
- **Implemented anywhere in the fabric**
  - Node, Switch, Specialized device
- **No more than one active SM allowed**
  - 1 Active (Master) and remaining are Standby (HA)

1. The **SM wakes up** and starts the Fabric Discovery process
2. The SM starts “ **conversation** “ with every node , over the infiniband link it is connected to . in this stage the **discovery stage** , the SM collects :
  - Switch Information
  - Host Information
3. Any switch which is already discovered , will be used as a gate for the SM , for further discovery of all this switch links and the switches it is connected to, known also as its neighbors .
4. The SM management dialog :
  - Uses the SMP - **S**ubnet **M**anager **P**ackets
  - All management packets are handled via **V**irtual **L**ane -15



Physical Fabric  
Establish

Subnet Discovery

Information  
Gathering

Lid Assignment

Path Establishment

Port Configuration

Switch Configuration

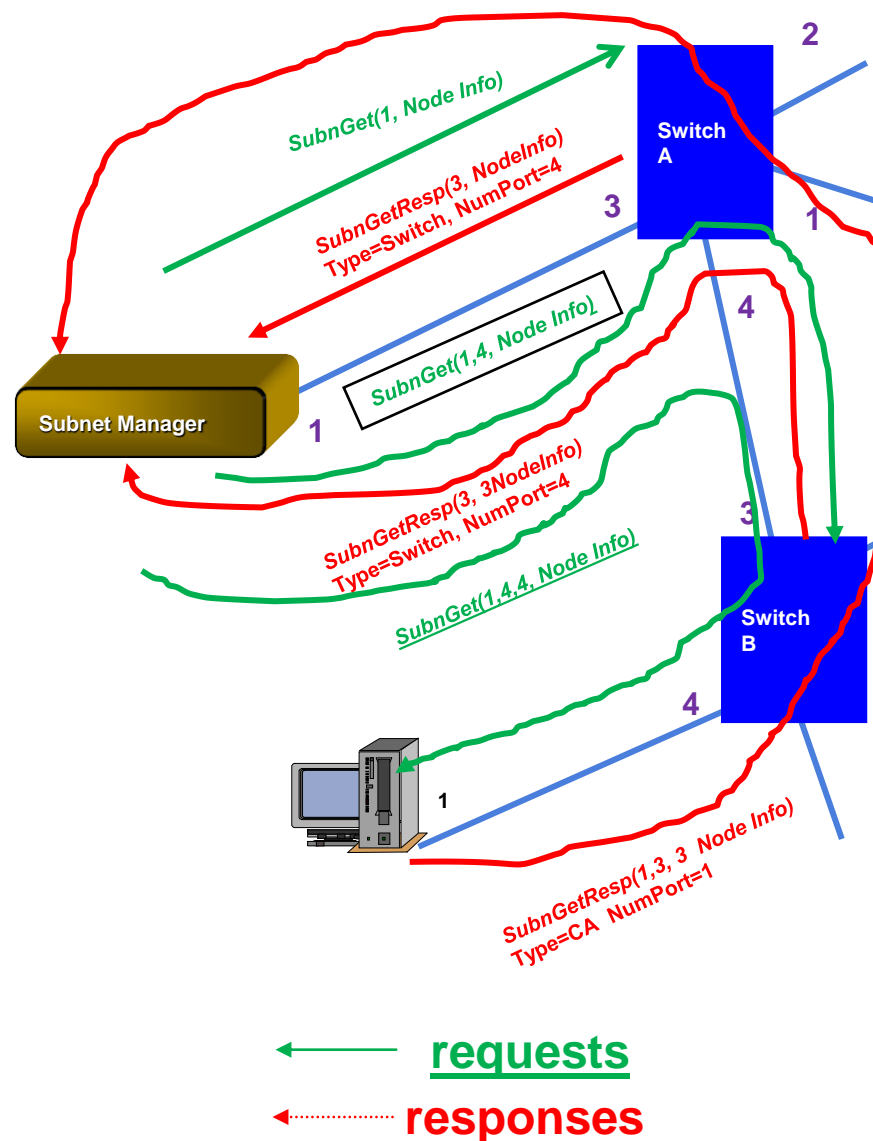
Subnet Activation

# Subnet Discovery managed by the SM

❖ SM requests like devices responses will include :

- ❑ Node Info
- ❑ Ports info

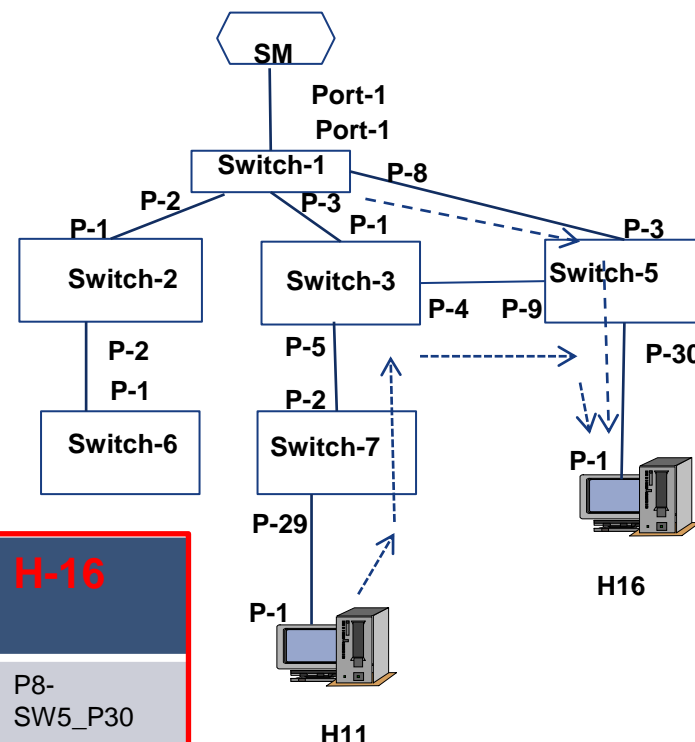
1. SM – I am requesting info via port number 1
2. Switch A – I am responding via port number 3 , I have an Active port Number 4
3. SM – I am requesting info via :
  - my port 1- next switch (A) port 4
4. SWITCH B – I am a switch responding via my port 3 via next switch (A) port 3
  - I have a live port port 4
5. SM – I am requesting info via :
  - my port 1- next switch (A) port 4 ,next switch (B) port 4
6. Host – I am a CA , responding via my port 1 , next switch (B) port 3 , next switch ( A ) port 3



- What information components are included under PORT INFO and Node INFO :
  1. Group Parameters
    - Type
    - NumPorts
    - GUID
    - Partition table size
  2. Group Parameters
    - Forwarding Database size
    - MTU
    - Width
    - VLs
  3. Group Parameters
    - IsSM,
    - IsM\_KeyinNVRAM, ...

# Fabric Direct route Information Gathering

- Building the direct routing table from & to each one of the fabric elements
- Each segment in a path, is identified by its *PORT NUMBER & GUID*
- The table content is saved in the SM LMX table

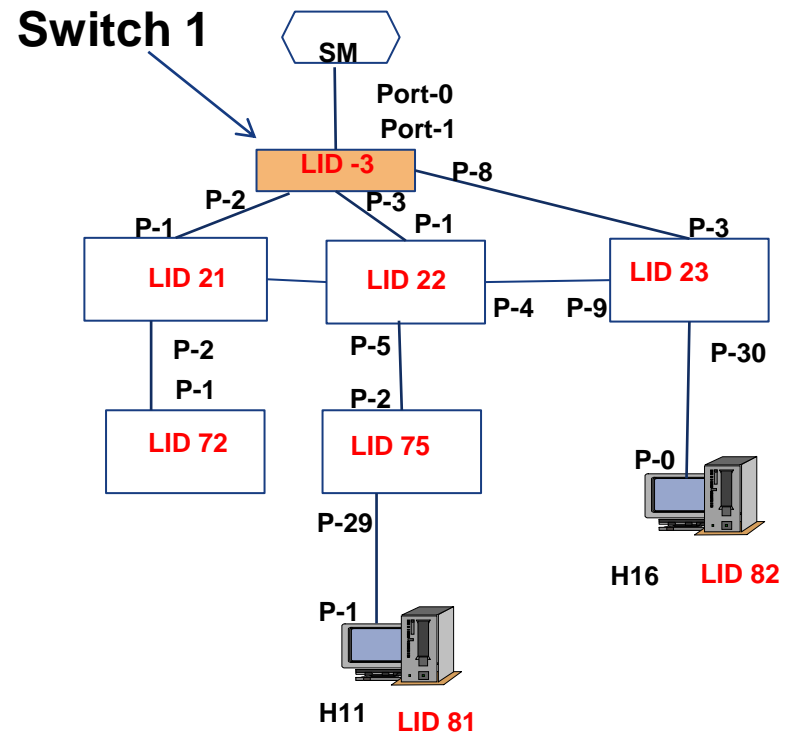


	SW-2	SW-6	SW-3	SW-7	SW-5	H-11	H-16
SW-1	P2	P2-SW2-P2	P3	P3-SW3-P5	P8	P3-SW3_P5-SW7_P29	P8-SW5_P30
SW-1						P8-SW5_P9-SW3_P5	
H11	P1-SW7_P2-SW3_P4-SW5_P30	P1-SW7_P2-SW3_P1-SW1_P2-SW2-P2	P1-SW7_P2-	P1-	P1-SW7_P2-SW3_P4-		P1-SW7_P2-SW3_P4-SW5_P30



# LID Assignment

- After the SM finished gathering all Fabric information , including direct route tables , it assigns a LID to each one of the NODES
- The LID is used as the Main identifier source& destination address for Infiniband packet switching
- The LID is assigned to a Device level Rather than a port Level
- Each port than will be identified by the combination of LID + Port Number



Physical Fabric  
Establish

Subnet Discovery

Information Gathering

**Lid Assignment**

Path Establishment

Port Configuration

Switch Configuration

Subnet Activation

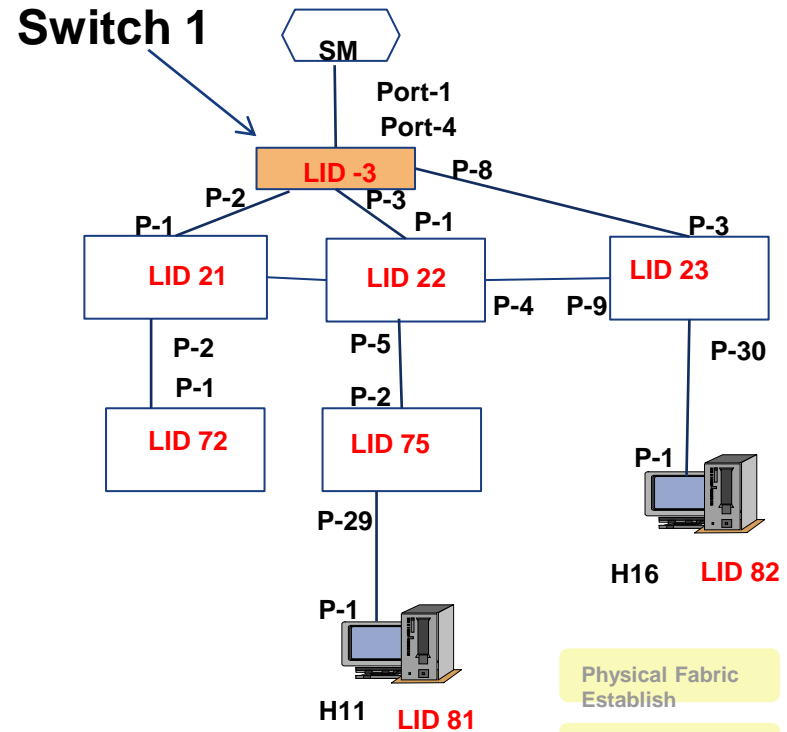
- **The SA is typically an extension of the SM**
- **A passive entity that provides a database of :**
  - Subnet topology
  - Device types
  - Device characteristics
- **Responds to queries**
  - Paths between HCAs
  - Event notification
  - Persistent information
  - Switch forwarding tables
- **Used to keep multiple SMs in sync**

- **The SA is typically an extension of the SM**
- **A passive entity that provides a database of :**
  - Subnet topology
  - Device types
  - Device characteristics
- **Responds to queries**
  - Paths between HCAs
  - Event notification
  - Persistent information
  - Switch forwarding tables
- **Used to keep multiple SMs in sync**

# Linear Forwarding Table Establishment (Path Establishment)



- After the SM finished gathering all Fabric information, including direct route tables, it assigns a LID to each one of the NODES
- At this stage the LMX table will be populated with the relevant routes option to each one of the nodes
- The output of the LMX will provide the Best Route to Reach a DLID. That Result Will be based on Shortest Path First (SPF)



LMX Switch\_1

LFT Switch\_1

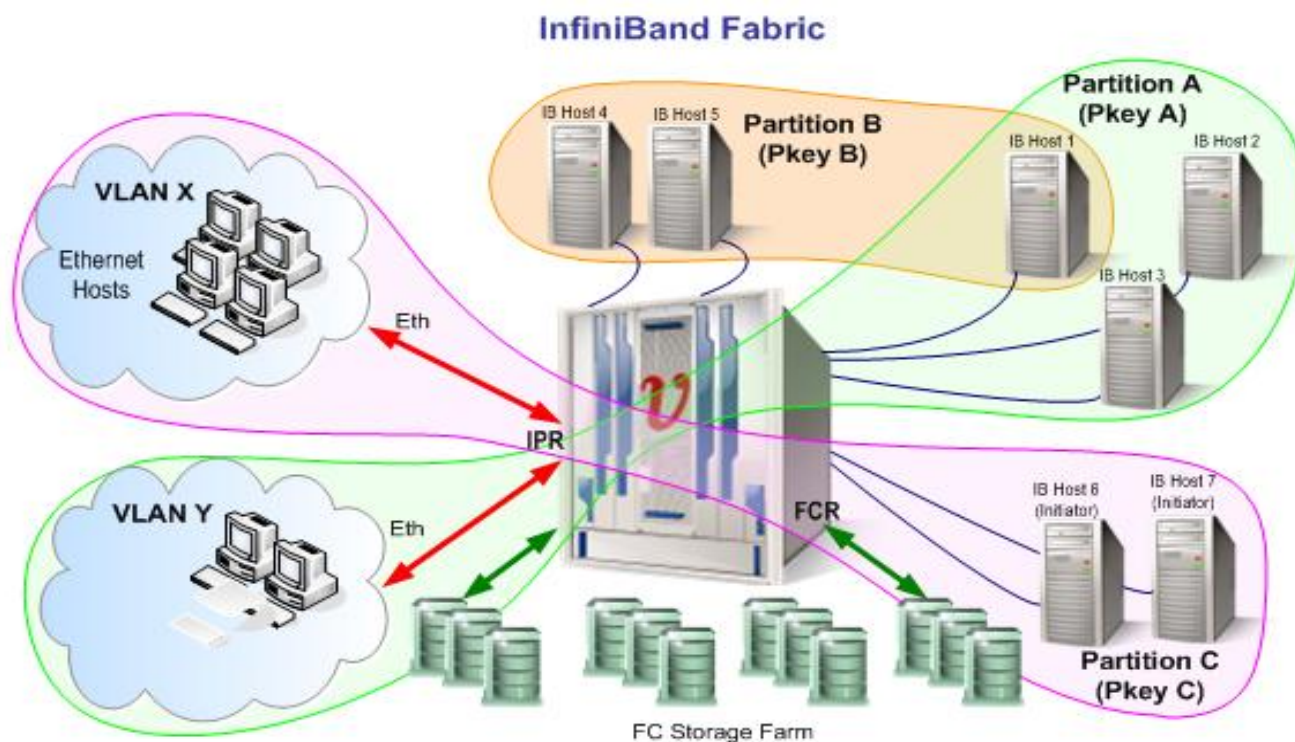
PORT \ D-LID	2	3	8	Min Hop s
21	1	2	3	1
22	2	1	2	1
23	3	2	1	1
75	3	2	3	2
81	4	3	4	3
82	4	3	2	2

The Dest. LID	Best Route / exit port
21	2
22	3
23	8
75	3
81	3
82	8

- Physical Fabric Establish
- Subnet Discovery
- Information Gathering
- Lid Assignment
- Path Establishment**
- Port Configuration
- Switch Configuration
- Subnet Activation

# Partitioning - Pkey to VLAN mapping

- Define up to 64 partitions in a single 10G/4036E
- Partition by mapping port and Ethernet VLAN to InfiniBand PKEY



- Using (partially) direct routed *SubnSet(PortInfo)* , the Subnet Master sets:
  - LID/LMC
  - MasterSM-LID and MasterSM-SL
  - P\_Keys
  - VLs
  - MTU
  - Rate
  - SLtoVL
  - VL arbitration

Physical Fabric  
Establish

Subnet  
Discovery

Information  
Gathering

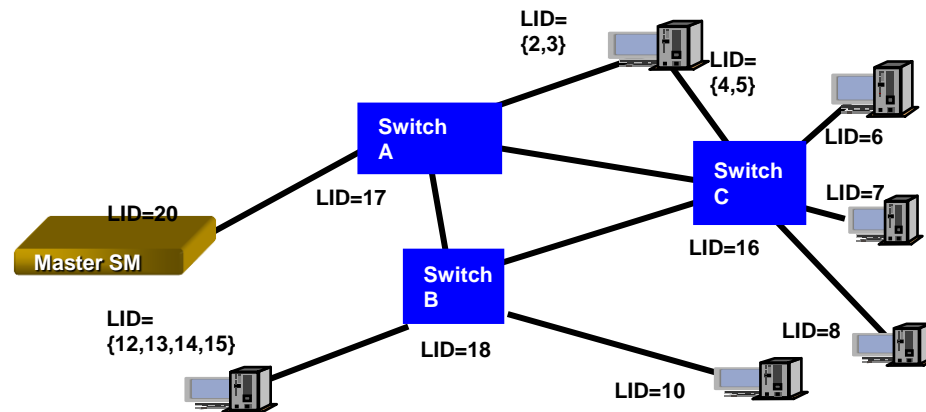
Lid Assignment

Path  
Establishment

Port  
Configuration

Switch  
Configuration

Subnet  
Activation





- Using the topology info, the Master “programs” the paths through the subnet by configuring the switches with
  - Unicast Forwarding Table: DLID → output-port
  - Multicast Forwarding Table: DLID → port-mask
  - SLtoVLMap: {SL, in-port, out-port} → VL
  - VL Arbitration tables
  - Optional P\_Key tables for P\_Key enforcement by switches

Physical Fabric  
Establish

Subnet  
Discovery

Information  
Gathering

Lid Assignment

Path  
Establishment

Port  
Configuration

**Switch  
Configuration**

Subnet  
Activation

# Subnet Activation

- The Master sends to ALL ports *SubnSet(PortInfo): PortState = **Armed*** (were INITIALIZE)
- All the ports change to **Active** state by:
  - The Master sending *SubnSet(PortInfo):PortState= **Active***
  - Any data packet sent to an **Armed** port causes the port state to change to **Active**
  - Data packets cause all the ports on a switch to change to **Active**
- When SM has sent **Active** to all ports, the subnet is operational.
- SA must be operational as soon as First port→**Active**

Physical Fabric  
Establish

Subnet  
Discovery

Information  
Gathering

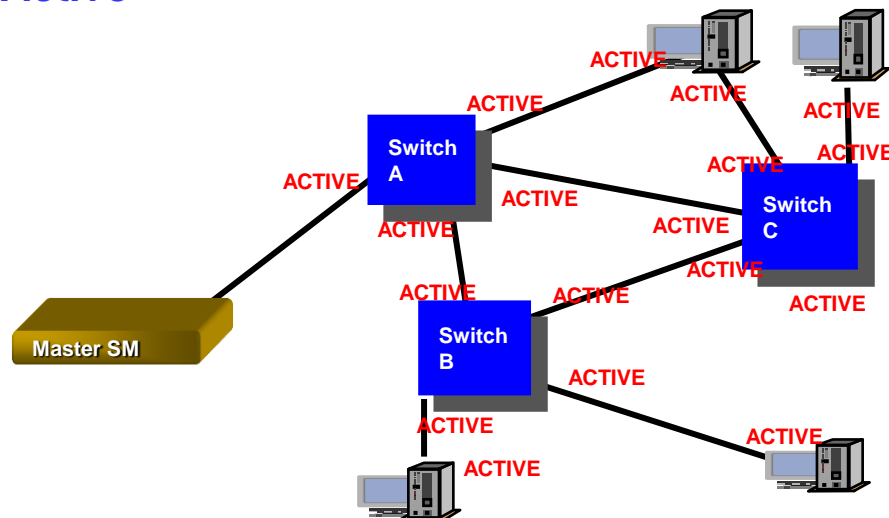
Lid Assignment

Path  
Establishment

Port  
Configuration

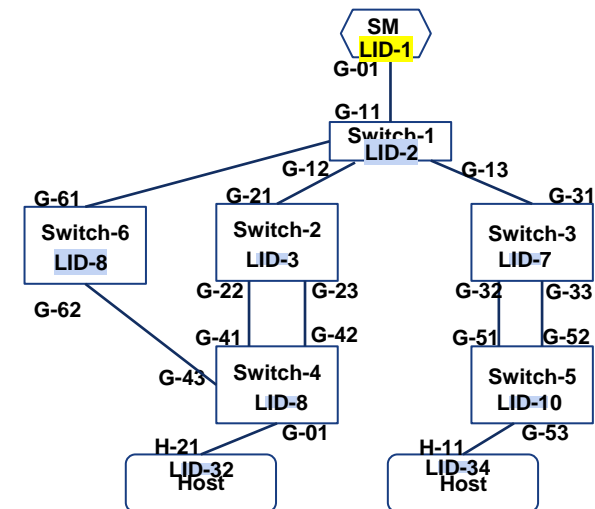
Switch  
Configuration

**Subnet  
Activation**



# What happens in the Fabric in case of Topology change ??

- **Fabric** Topology change may be caused by :
  - Switches added or removed
  - Links added , removed , fall , Recover
  - Operation administration or Maintenance activity
- A topology change will trigger **SM SWEEP** :
  - Every status change of a Port/Link will cause a Trap that is sent By the switch to the SM over VL-15
  - A change in the Topology triggers the SM to start a process in which Every node in the Fabric will have to report its node and ports status
  - The LMX will rebuilt although in many cases most of its data including direct routes will remain the same
  - If there is no need LID of the Nodes will not be changed
  - Traffic packets that are not physically impacted by link or a switch failure will not be affected
- In order to avoid frequent unnecessary change of tables routing and updated flooding of the FLTs , modification occurs only following a status change.



- Subnet Manager Failover & Handover
  - Scope: an InfiniBand subnet
  - SM implements the standard “SMInfo” protocol
  - The “SMInfo” protocol defines failover and handover between subnet managers in a given IB subnet
  - ActCount Increments each time a SM performs a SMP or other mgt activities. Used as a heartbeat indicator for standby SM’ s.
  - The SM with the higher priority / lower GUID is elected as master  
All other SM's are standby – polling the master activity

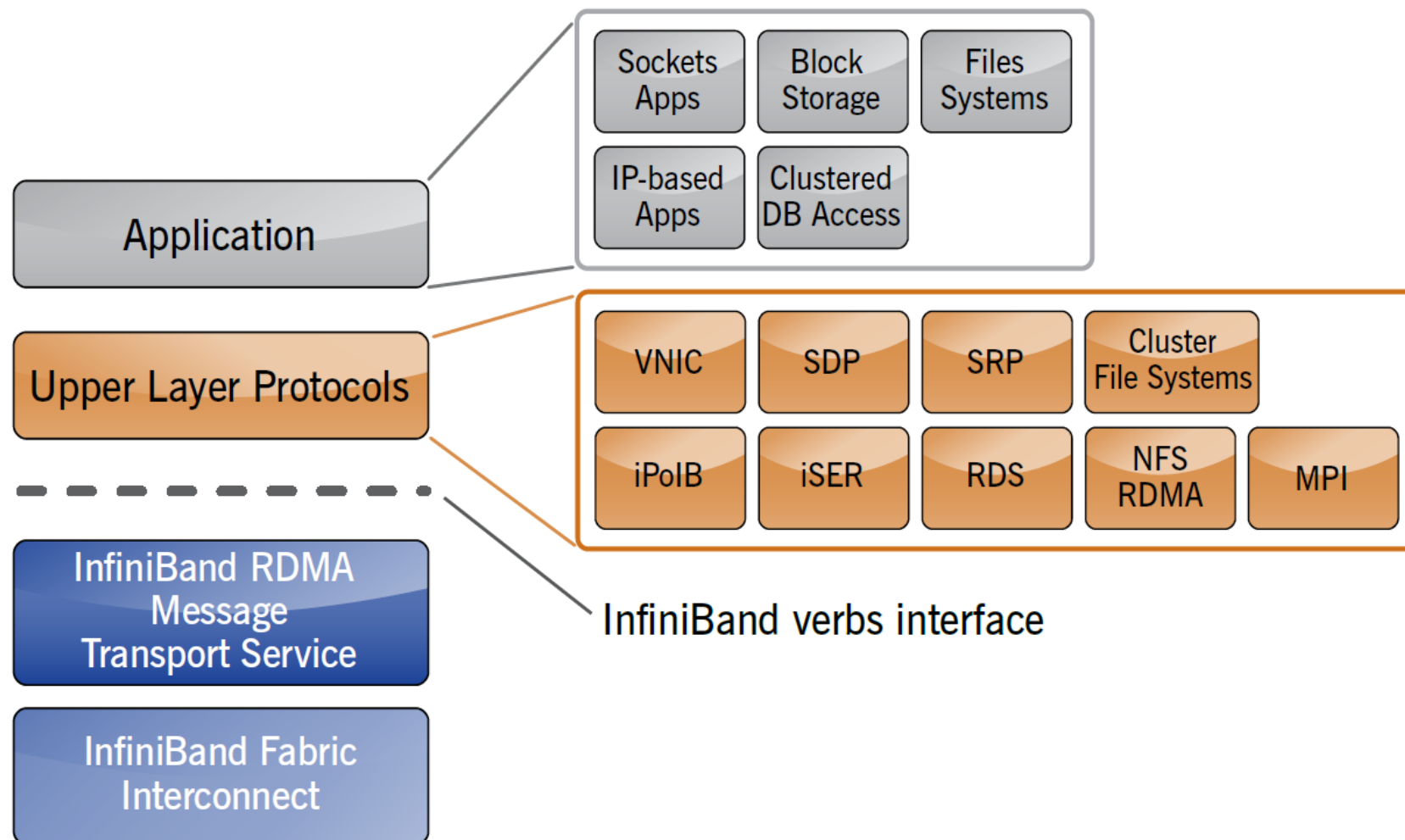
- OpenSM (osm) is an Infiniband compliant subnet manger.
- Included in Linux Open Fabrics Enterprise Distribution.
- Ability to run several instance of osm on the cluster in a Master/Slave(s) configuration for redundancy.
- Partitions (p-key) support
- QoS support
- Enhanced routing algorithms:
  - Min-hop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

# IB Fabric Protocol Layers

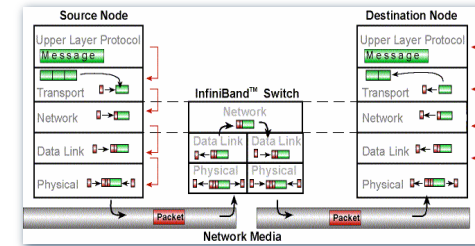




# Infiniband Protocol Layers



- **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



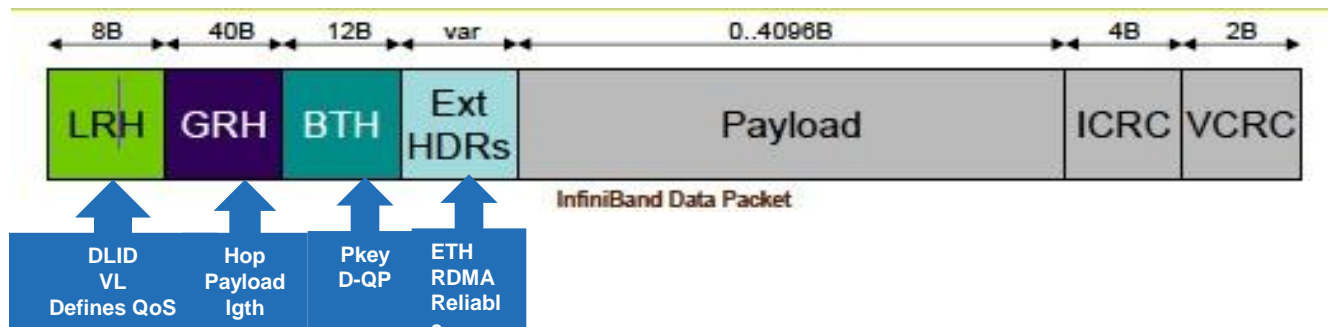
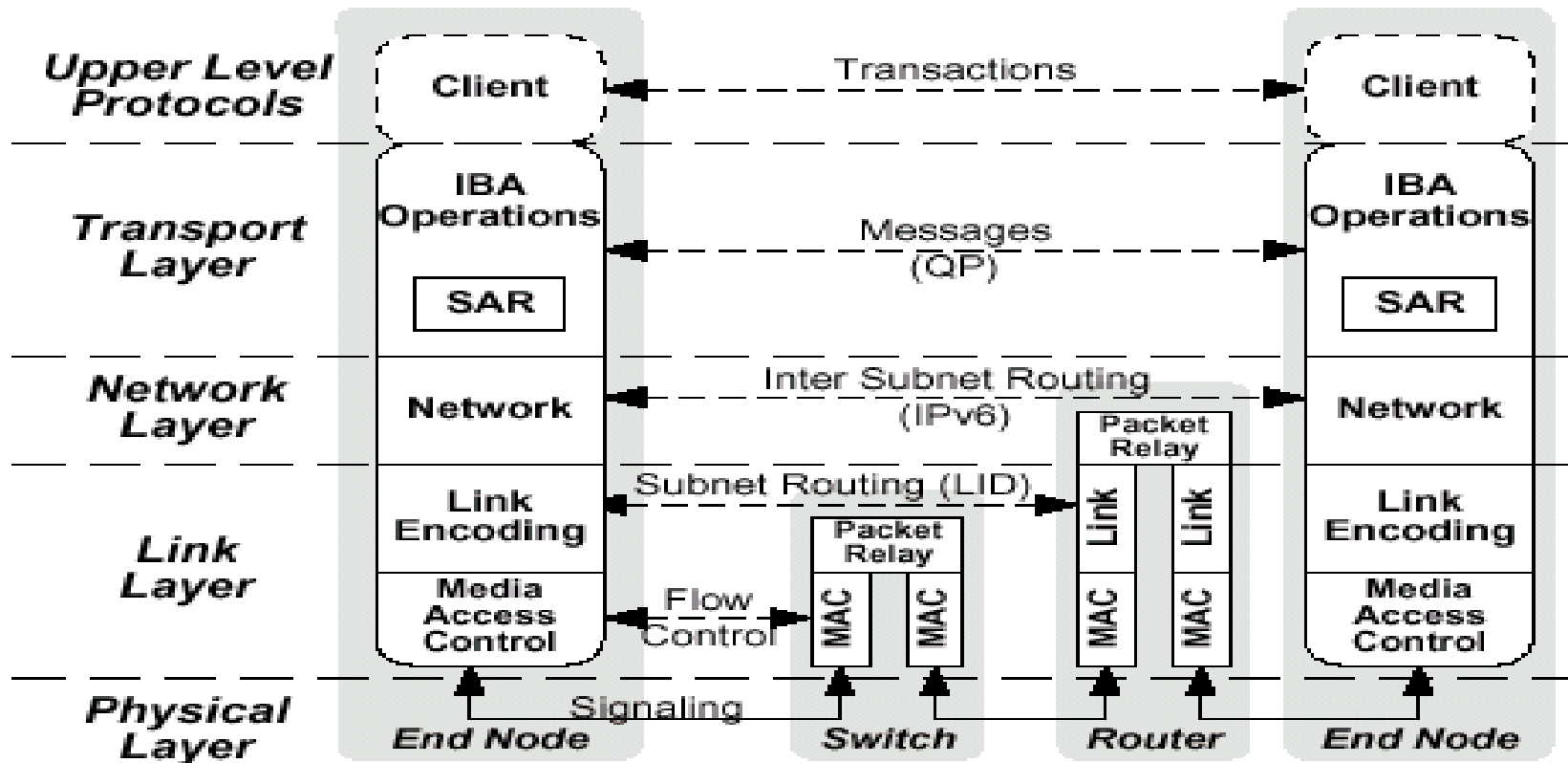
- **Transport:**
  - Delivers packets to the appropriate Queue Pair; Message Assembly/De-assembly, access rights, etc.

- **Network:**
  - How packets are routed between Different Partitions /subnets

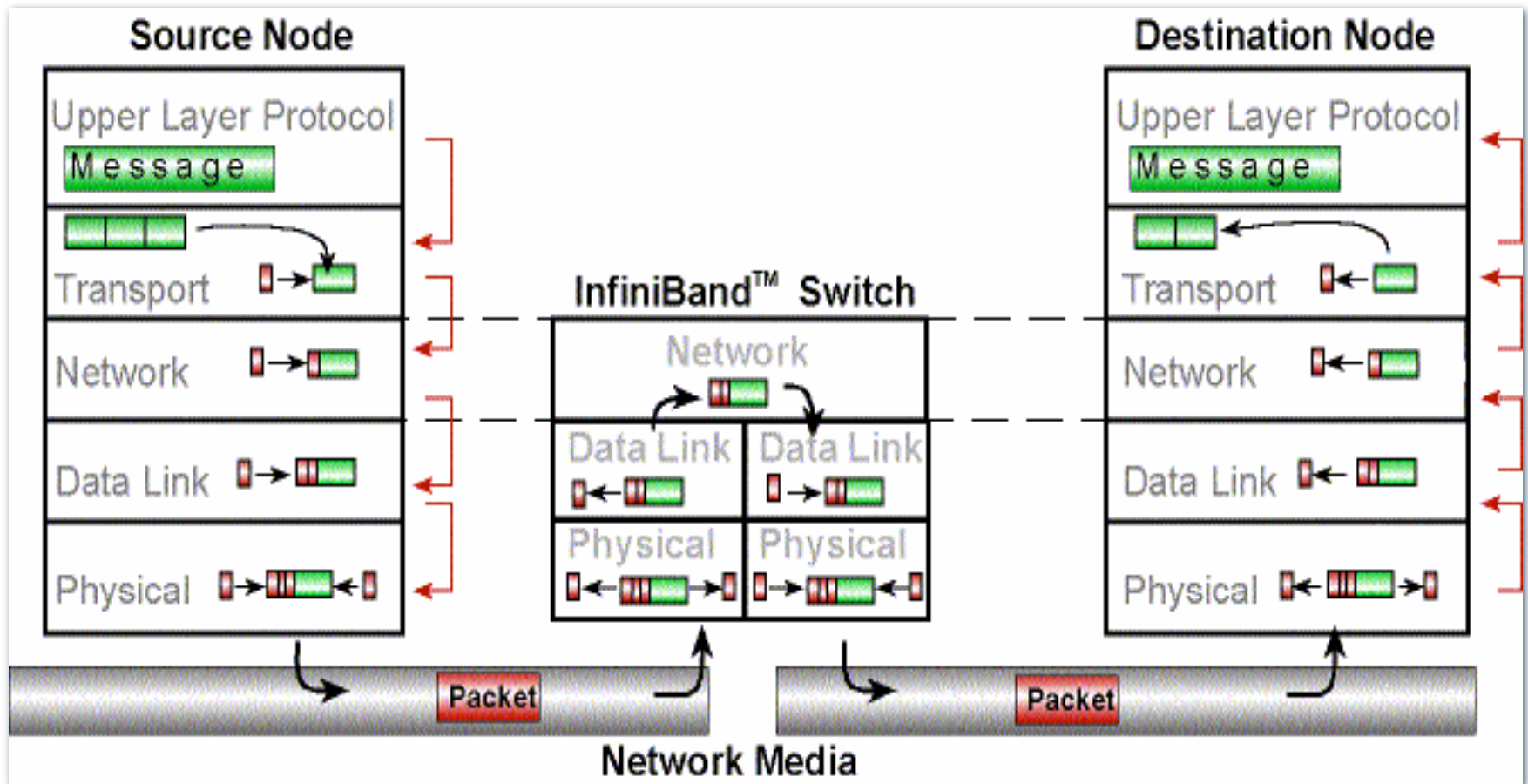
- **Data Link (Symbols and framing):**
  - Flow control (credit-based); How packets are routed , from Source to Destination on the same Partition Subnet

- **Physical:**
  - Signal levels and Frequency; Media; Connectors

# Distributed Computing using IB



- Extended headers:**
- Reliable Datagram ETH (4B)
  - Datagram ETH (8B)
  - RDMA ETH (16B)
  - Atomic ETH (28B)
  - ACK ETH (4B)
  - Atomic ACK ETH (8B)
  - Immediate Data ETH (4B)
  - Invalidate ETH (4B)



- InfiniBand is a lossless fabric.
- Maximum Bit Error Rate (BER) allowed by the IB spec is  $10e-12$ .
- The physical layer should guaranty affective signaling to meet this BER requirement
- The physical layer specifies how :
  - Bits iare placed on the wire to form symbols
  - Defines the symbols used for framing (i.e., start of packet & end of packet), data symbols,
  - Fill between packets (Idles).
  - Specifies the signaling protocol as to what constitutes a validly formed packet

- InfiniBand uses serial stream of bits for data transfer

- **Link Speed**

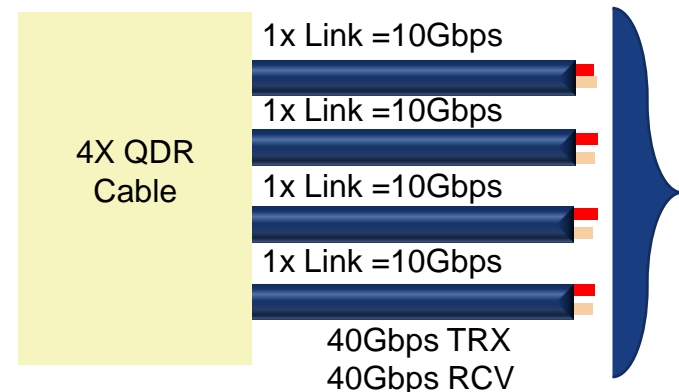
- Single Data Rate (SDR) - 2.5Gb/s per lane (10Gb/s for 4x)
- Double Data Rate (DDR) - 5Gb/s per lane (20Gb/s for 4x)
- Quad Data Rate (QDR) - 10Gb/s per lane (40Gb/s for 4x)
- Fourteen Data Rate (FDR) - 14Gb/s per lane (56Gb/s for 4x)
- Enhanced Data rate (EDR) - 25Gb/s per lane (100Gb/s for 4x)

- **Link width**

- 1x – One differential pair per Tx/Rx
- 4x – Four differential pairs per Tx/Rx
- 12x - Twelve differential pairs per Tx and per Rx

- **Link rate**

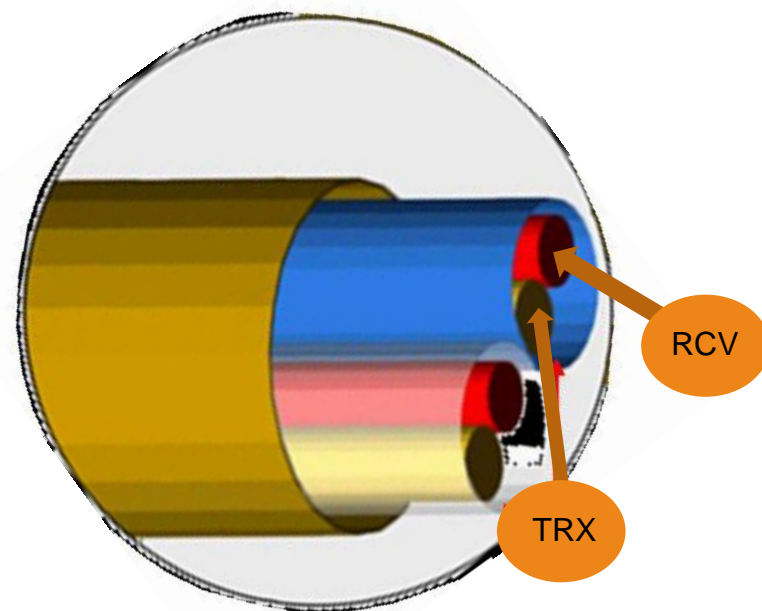
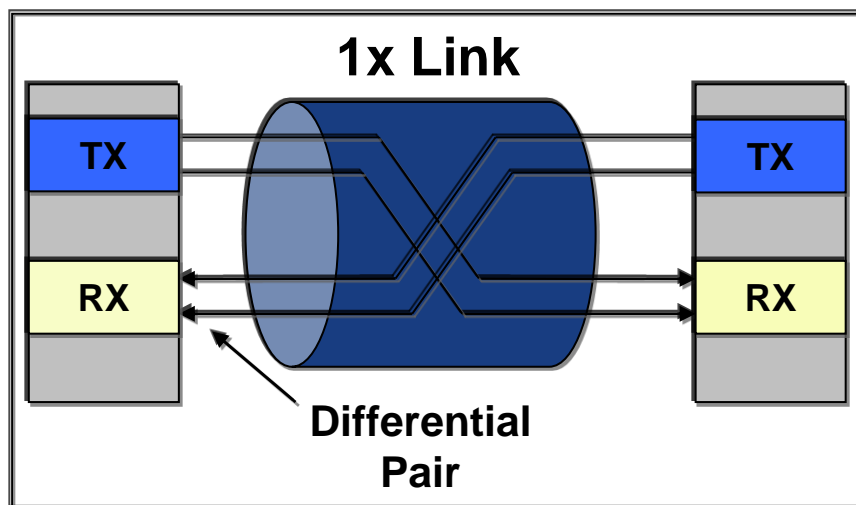
- Multiplication of the link width and link speed
- Most common shipping today is 4x ports





- 1X Link is the basic building block

- Differential pair of conductors for RX
- Differential pair of conductors for TX
- Link Rate per type
  - Timed at 2.5 GHz with SDR
  - Doubled to 5GHz with DDR
  - Quad to 10GHz with QDR



## ■ Media types

- Printed Circuit Board : several inches
- Copper: 20m SDR, 10m DDR, 7m QDR
- Fiber: 300m SDR, 150m DDR, 100/300m QDR

## ■ 64/66 encoding on FDR links

- Encoding makes it possible to send digital High Speed signals to a Longer Distance
- x actual data bits are sent on the line by y bits
- $64/66 * 56 = 54.6\text{Gbps}$

## ■ 8/10 bit encoding (SDR, DDR, and QDR)

- x/y Line efficiency ( example  $80\% * 40 = 32\text{Gbps}$  )

## ■ Industry standard components

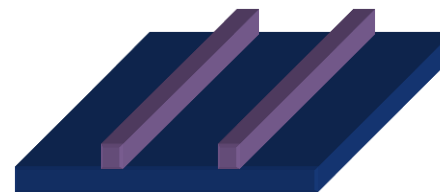
- Copper cables / Connectors
- Optical cables
- Backplane connectors



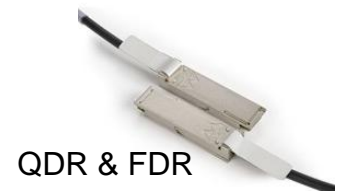
4X CX4



4x CX4 Fiber

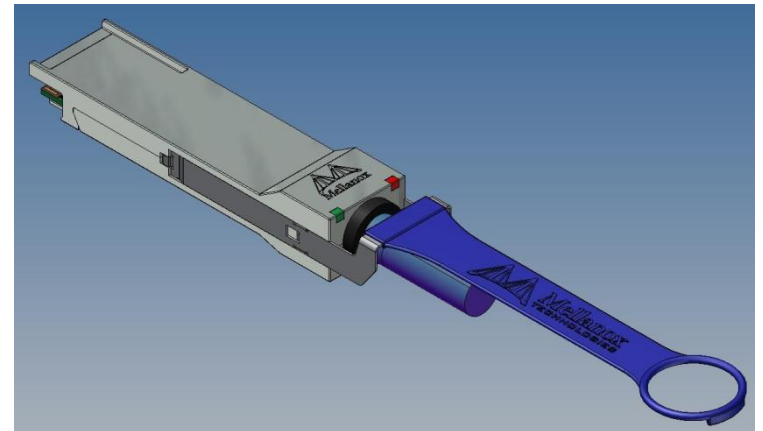
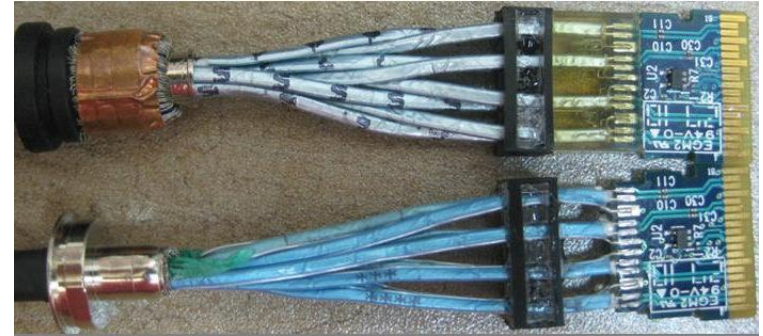


FR4 PCB



4X QSFP Copper

- Connector – Mellanox PCB design
  - Leverage board design and H/S expertise
- Improved connector and housing
  - Servers cage compliancy
  - Additional functionality
  - Built-in LEDs
- Active Copper Cables
  - QDR, 40GigE, FDR10
  - Lower power
- Cable optimization for length, performance and cost

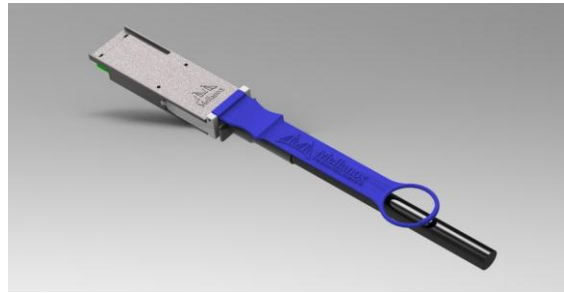


- Mellanox cables are rebranded from a cable vendor
  - Mellanox cables are manufactured by Mellanox

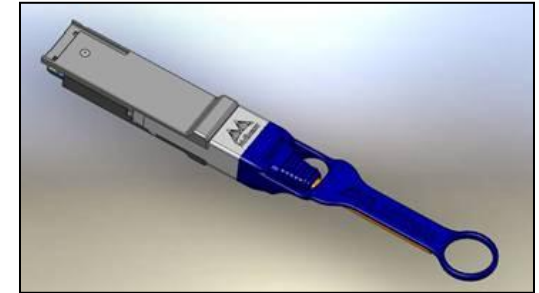
## Passive Copper Cables



## Active Copper Cables

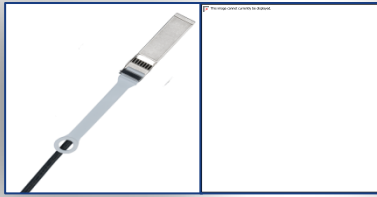


## Active Optical Cables



- Our vendor can sell the same cables
  - No other vendor is allowed to sell Mellanox cables
- Mellanox cables use a different assembly procedure
- Mellanox cables are tested with unique test suite
- Vendors' "Finished Goods" fail Mellanox dedicated testing

# Cable Portfolio



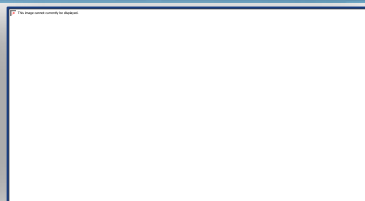
	QSFP-QSFP	SFP+ - SFP+	CX4 – CX4	CXP - CXP
Applications / Date Rates	DDR/QDR, 40GE and 10GE	10GE	DDR, 10GE	DDR/QDR
Passive Copper	Up to 8m	Up to 7m	Up to 8m	Up to 7m
Active Copper	Up to 12m	N/A	Up to 15m	N/A
Active Optical (fiber)	Regular SKUs up to 300m. Options available up to 4Km.	Use 10GBASE-xR Optical Modules	Up to 100m	Up to 50m
Optical Modules	SR4 based, MTP/MPO connector	10GBase-SR 10GBase-LR	N/A	N/A

**Notes:**

- CX4 connectors also known as MicroGiGaCN
- QSFP and QSFP+ are used here synonymously, all cables adhere to QSFP+ spec, SFF-8436.



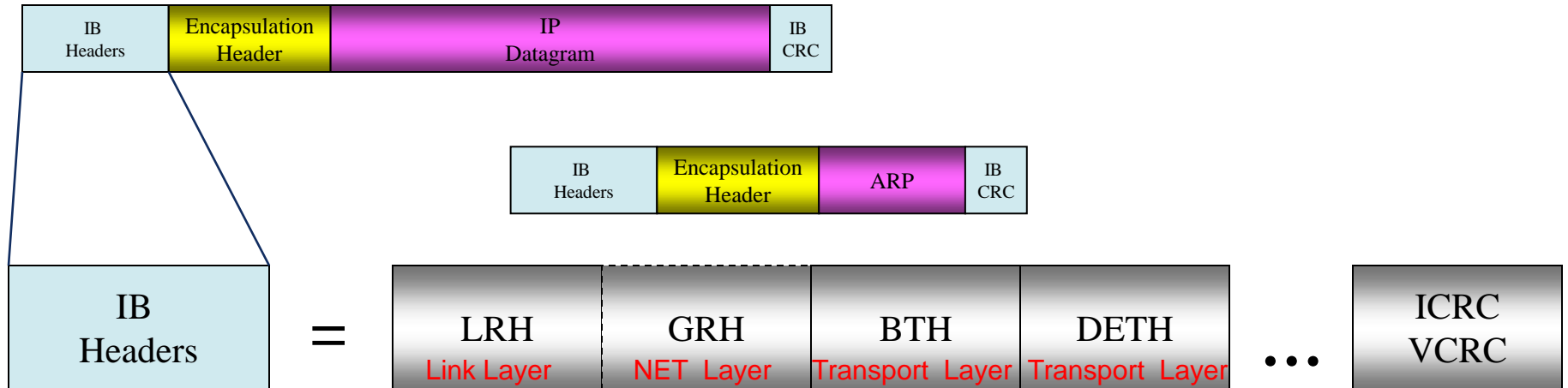
# Cable Portfolio – Hybrid Cables



	QSFP-CX4	QSFP+ - SFP+	QSA	Tri -QSFP - CXP
Applications / Date Rates	DDR, 10GE	10GE	10GE	DDR/QDR
Passive Copper	Up to 5m	Up to 7m	Use SFP+ DA cable	Up to 6m
Active Optical (fiber)	Up to 100m	Use QSA and SFP+ Optical Module	Use SFP+ Optical Module	Up to 50m

**Notes:**

- CX4 connectors also known as MicroGiGaCN
- QSFP and QSFP+ are used here synonymously, all cables adhere to QSFP+ spec, SFF-8436.



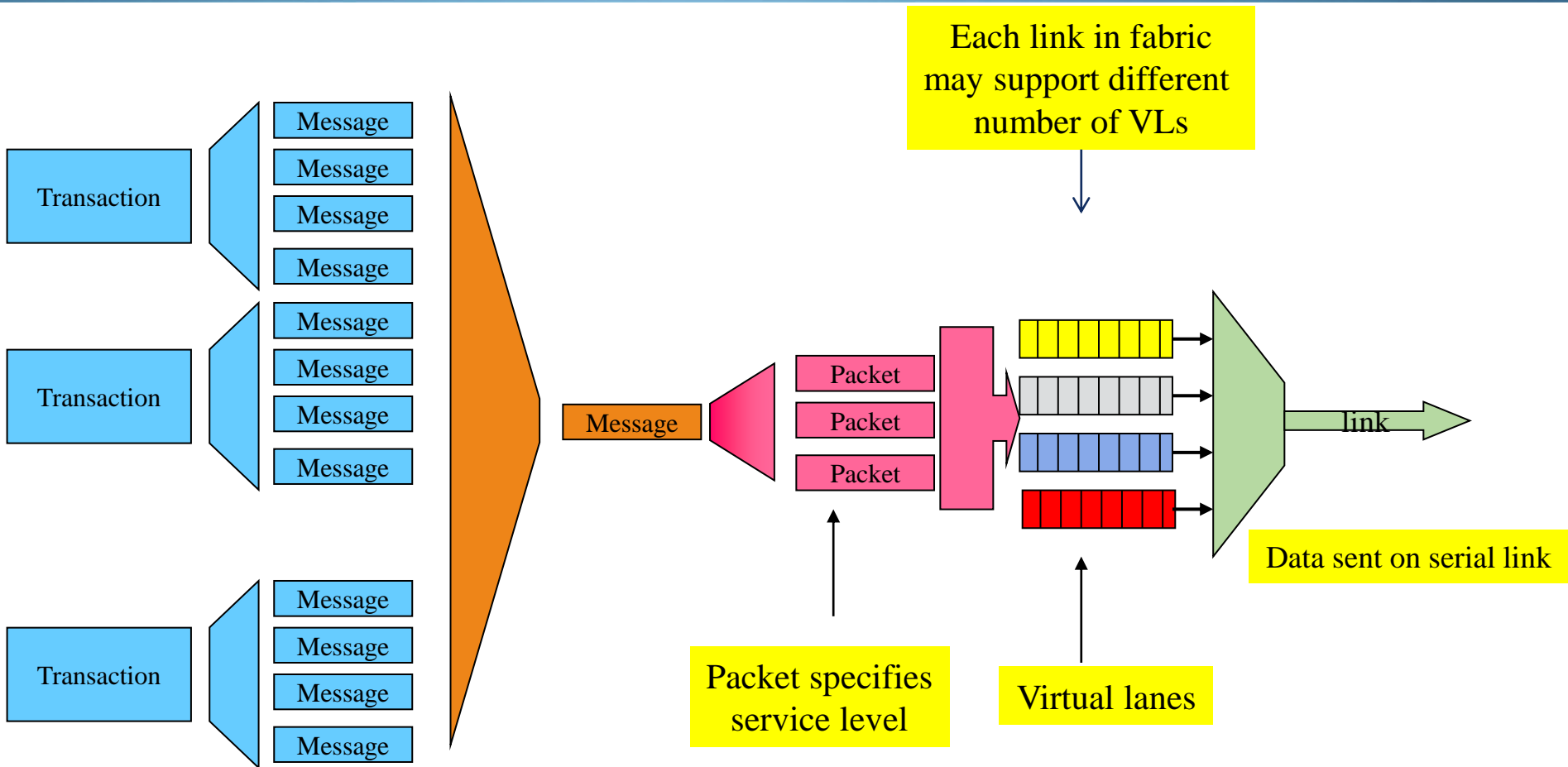
**LRH**: Local Routing Header – Includes LIDs, SL, etc

**BTH**: Base Transport Header – includes opcode, destination QP, partition, etc.

All Layers



# Link Layer Priority Implementation SL to VL Mapping



**LRH:** Local Routing Header – Includes LIDs, SL, etc

- 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)

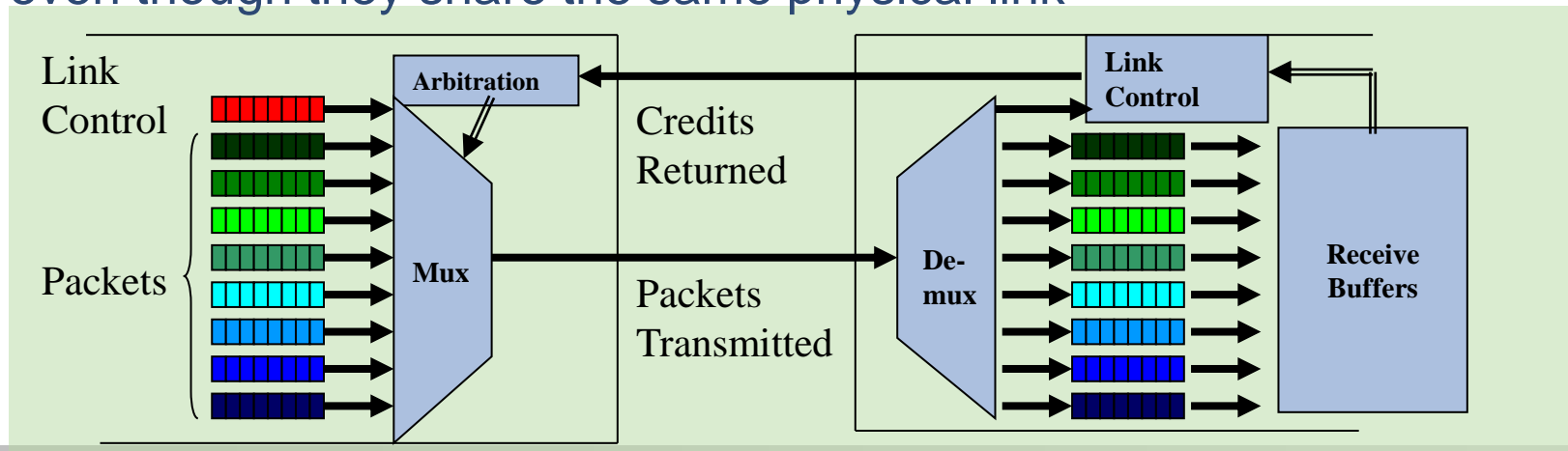
## ■ 16 Service Levels (SLs)

- 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

- 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



# Mellanox Switches Packet Flow QOS Management

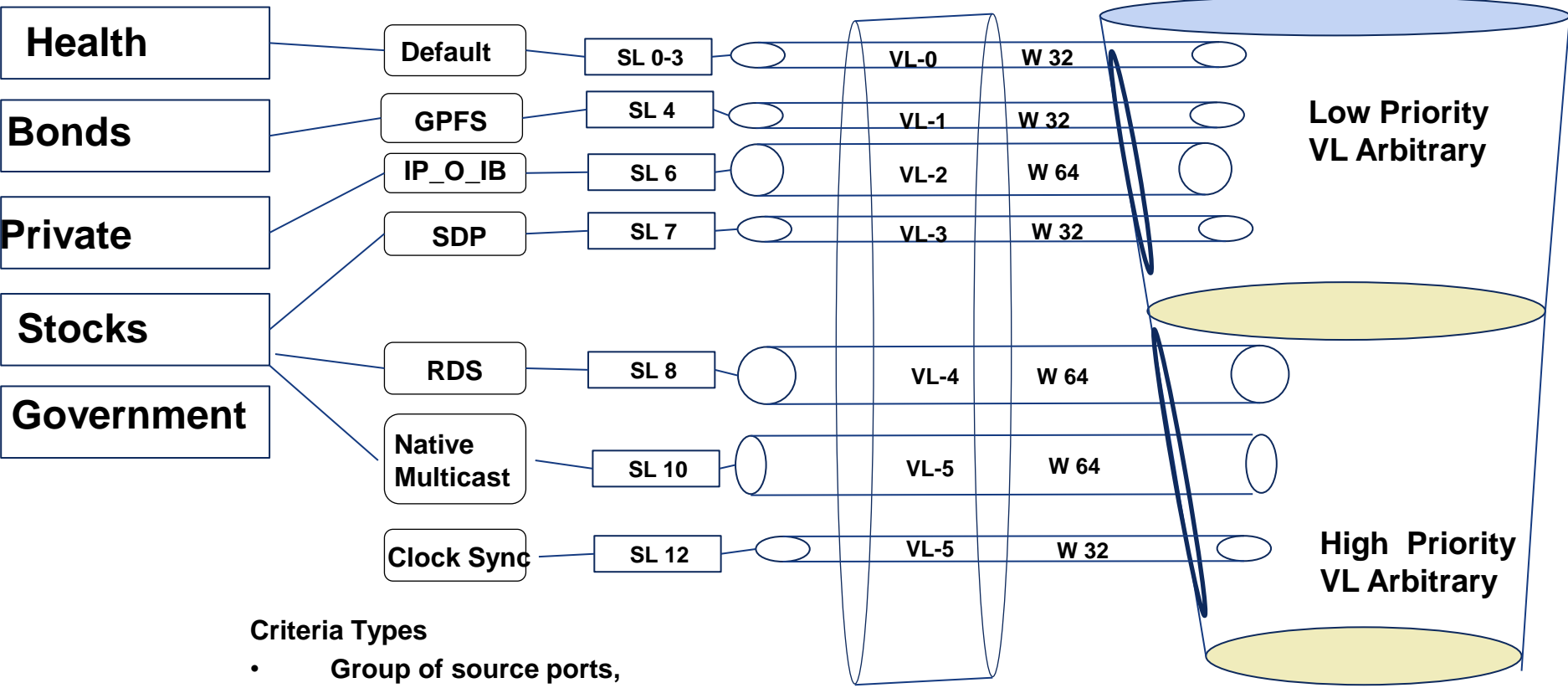


Fabric Nodes  
Users

Packets  
Criteria  
Categorized

Service  
level

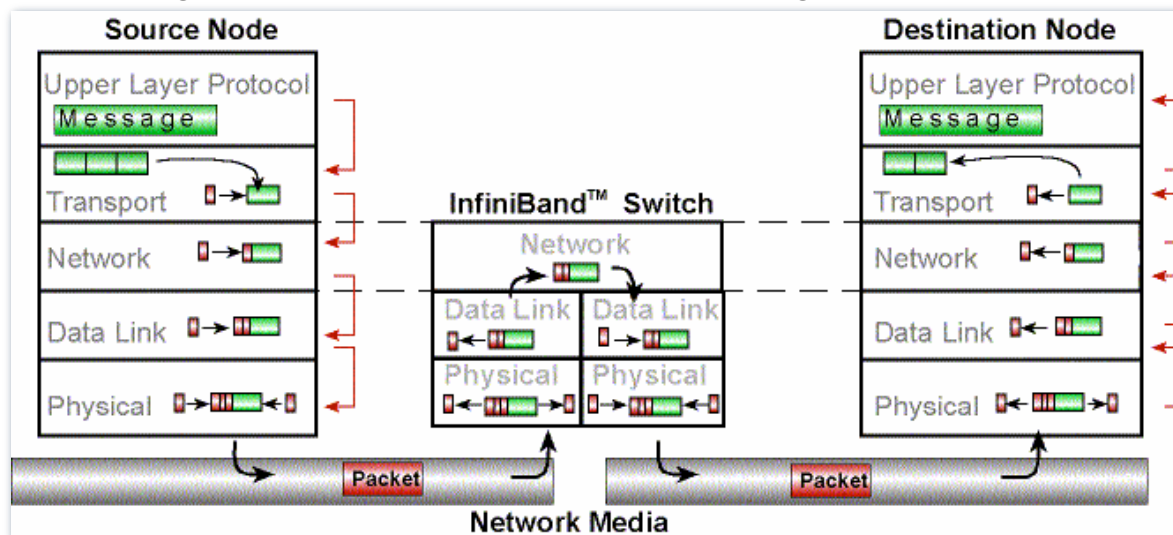
Virtual Lanes  
over Physical Link

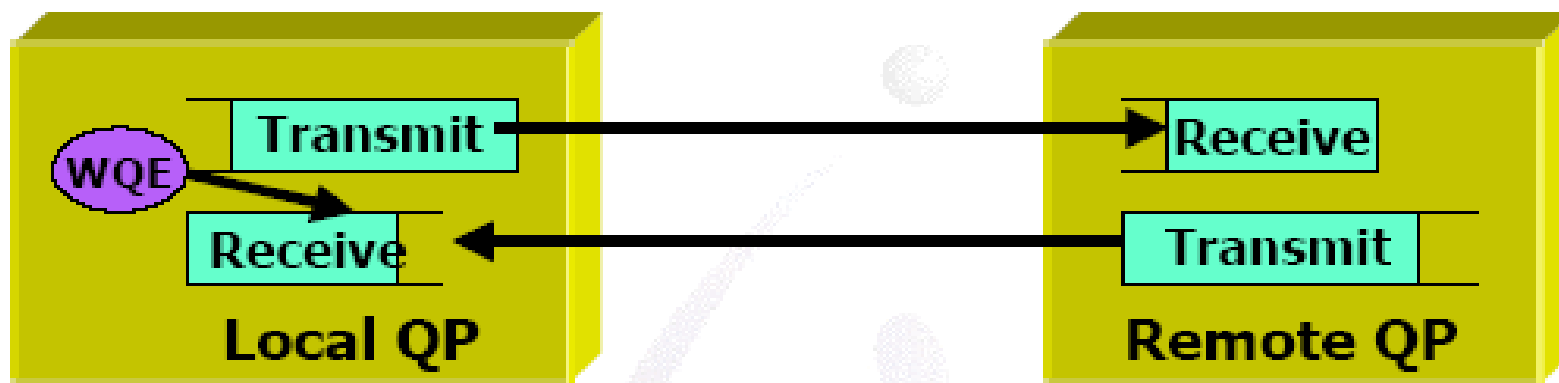


### Criteria Types

- Group of source ports,
- Groups of destination ports,
- Partitions,
- QOS classes
- Application Service ID's

- The network and link protocols deliver a packet to the desired destination.
- The transport Layer
  - Segmenting Messages data payload coming from the Upper Layer , into multiple packets that will suit Valid MTU size
  - Delivers the packet to the proper Queue Pair (assigned to a specific session )
  - Instructs the QP how to process the packet's data. ( Work Request Element )
  - Reassembles the Packets arriving from the Other side into Messages





- QPs are in pairs (Send/Receive)
- Every active connection / Session will be assigned with Individual Working Que Pair
- 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)



## ■ Data transfer

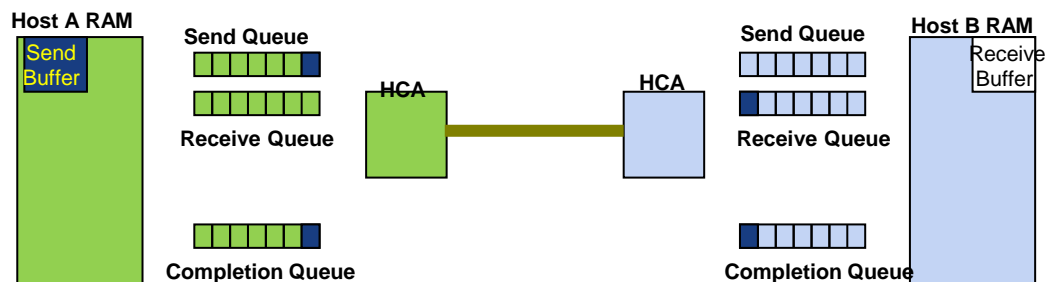
- Send work request
  - Local gather – remote write
  - Remote memory read
  - Atomic remote operation
- Receive work request
  - Scatter received data to local buffer(s)

## ■ 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

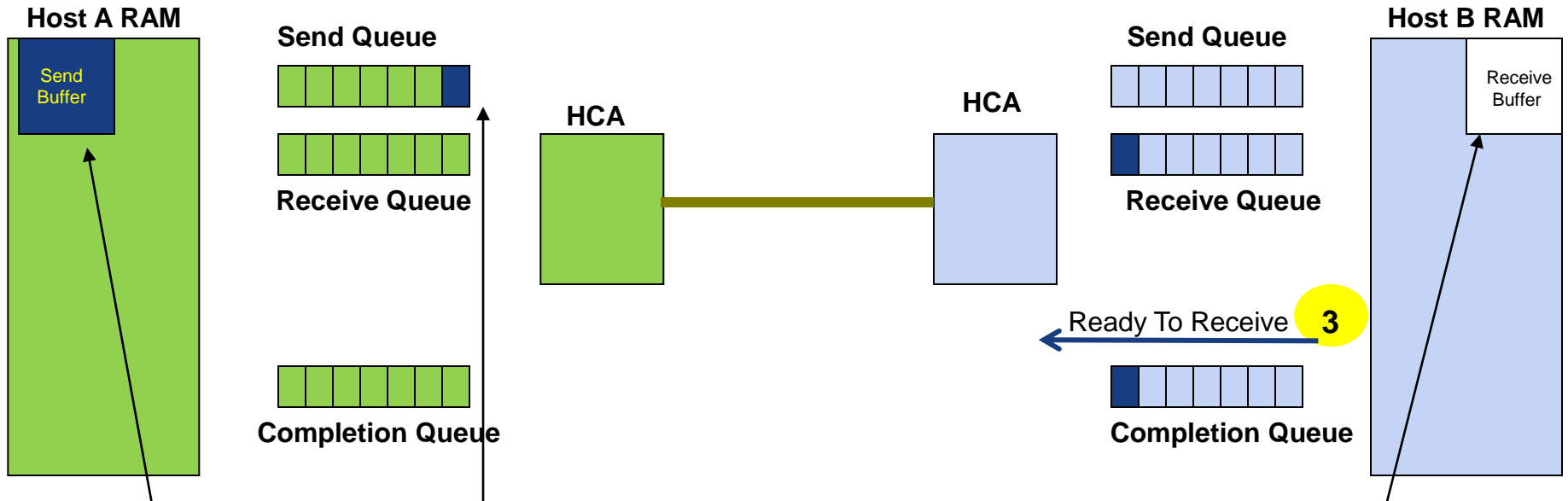
- Memory registration/mapping
- Open/close connection (QP)



# Transport Layer – **Send** operation example

- 4**
- HCA then Executes the send Request,
  - read the buffer of the Host Ram
  - and send to remote side (HCA)

- 5**
- When the packet arrives to the HCA
  - It Executes the **receive WQE Commands**
  - Place the buffer **CONTENT** in the appropriate location
  - And Generate a Completion Que



- 2**
- The send side allocates a send buffer on the **User Space Virtual Memory** register it with the HCA,
  - place a send Request On the send que

- 1**
- The Receive side Application allocates receive buffer on the **User Space Virtual Memory** register it with the HCA,
  - And place a receive **Work Request** on the Receive QUE

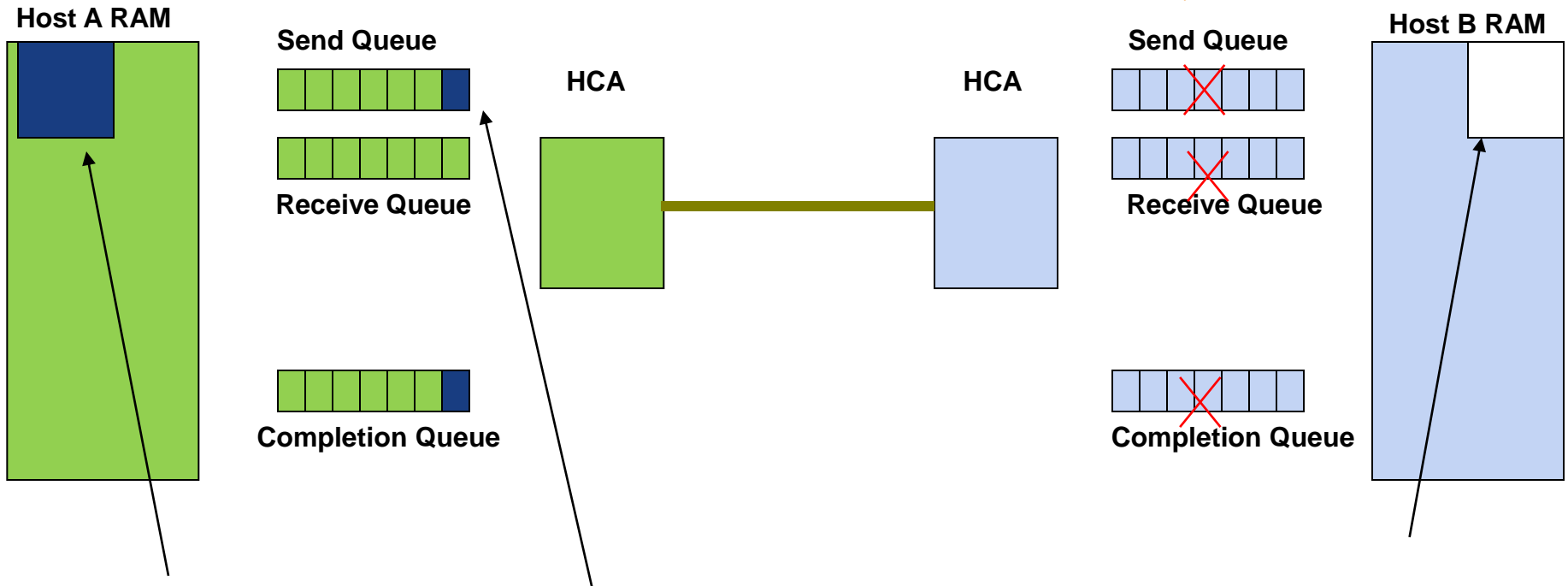
# Transport Layer – RDMA Write Example

3

- HCA then Executes the send Request commands
- Reads the buffer and send to remote side
- send completion is generated

4

- When the packet arrives to the HCA
- It checks the address and memory keys
- And write to Host memory directly
- No use of HCA QUES



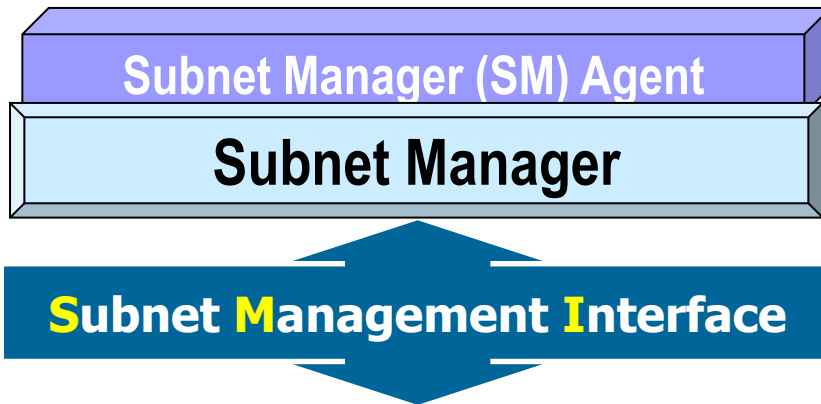
2

- The send side allocates a send buffer on the **User Space Virtual Memory** register it with the HCA,
- place a send Request On the send queue with the remote side's virtual address and the Remote Permission key

1

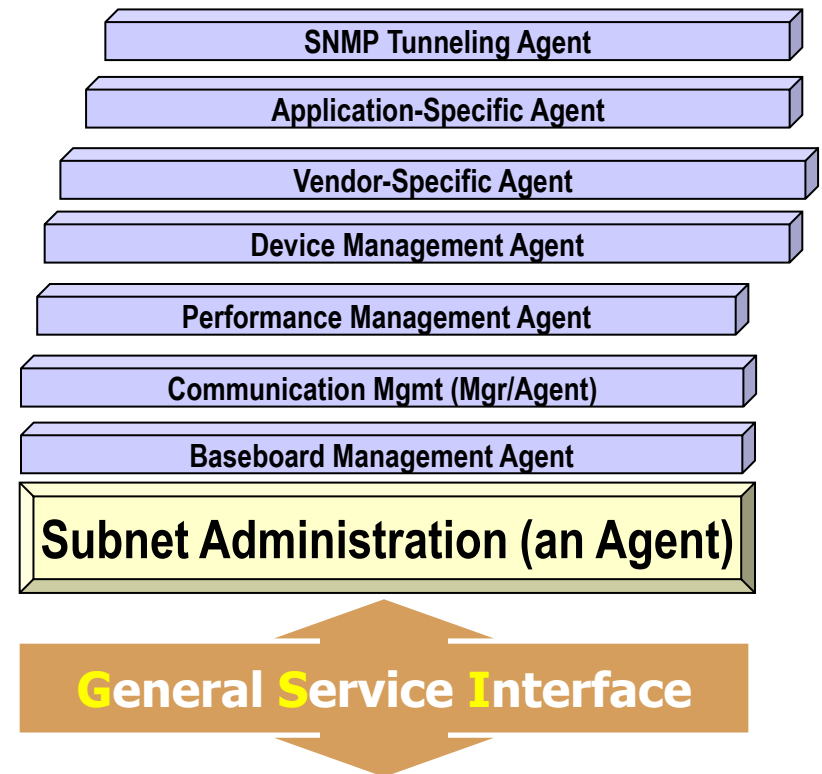
- Application performs memory Registration
- And passes address and keys to remote side
- No HCA Receive que is assigned

## Pure InfiniBand Management



QP0 (virtualized per port)  
Always uses VL15  
MADs called **SMPs** – LID or Direct-Routed  
No Flow Control

## Other Management Features



QP1 (virtualized per port)  
**Uses any VL except 15**  
MADs called **GMPs** - LID-Routed  
Subject to Flow Control

- **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)**
  - Power Management
  - Hot plug in and removal of modules
  - Monitoring of environmental parameters

- There are **7** management packet types (256 byte per packet)

Fatal

Urgent

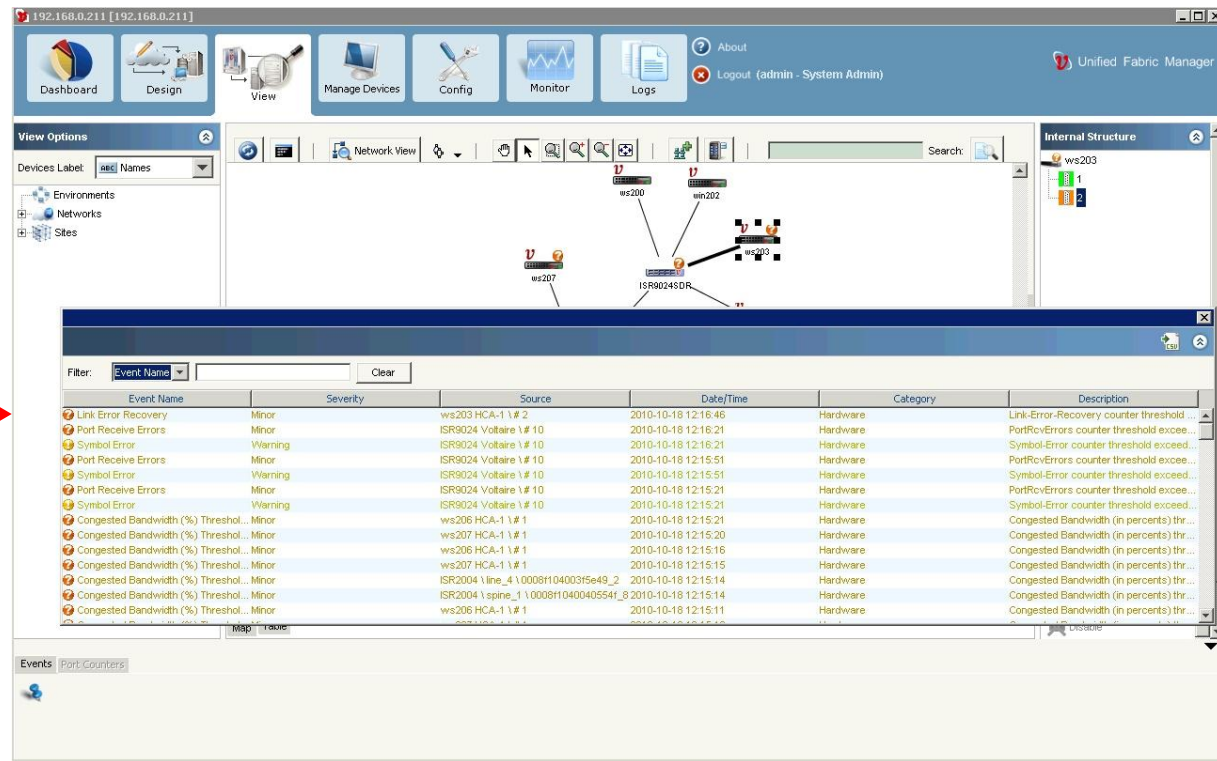
Security

Reserved

Informational

Empty Notice

Subnet Management



The screenshot shows the Unified Fabric Manager interface. The top navigation bar includes Dashboard, Design, View, Manage Devices, Config, Monitor, and Logs. The main area displays a network topology with nodes labeled ws200, ws202, ws203, and ws207 connected to a central node labeled ISR9024SDR. An event log window is open, showing a table of events.

Event Name	Severity	Source	Date/Time	Category	Description
Link Error Recovery	Minor	ws203 HCA-1 \# 2	2010-10-18 12:18:46	Hardware	Link-Error-Recovery counter threshold exceeded
Port Receive Errors	Minor	ISR9024 Voltaire 1 \# 10	2010-10-18 12:16:21	Hardware	PortRecvErrors counter threshold exceeded
Symbol Error	Warning	ISR9024 Voltaire 1 \# 10	2010-10-18 12:16:21	Hardware	Symbol-Error counter threshold exceeded
Port Receive Errors	Minor	ISR9024 Voltaire 1 \# 10	2010-10-18 12:15:51	Hardware	PortRecvErrors counter threshold exceeded
Symbol Error	Warning	ISR9024 Voltaire 1 \# 10	2010-10-18 12:15:51	Hardware	Symbol-Error counter threshold exceeded
Port Receive Errors	Minor	ISR9024 Voltaire 1 \# 10	2010-10-18 12:15:21	Hardware	PortRecvErrors counter threshold exceeded
Symbol Error	Warning	ISR9024 Voltaire 1 \# 10	2010-10-18 12:15:21	Hardware	Symbol-Error counter threshold exceeded
Congested Bandwidth (%) Threshold...	Minor	ws206 HCA-1 \# 1	2010-10-18 12:15:21	Hardware	Congested Bandwidth (in percents) threshold exceeded
Congested Bandwidth (%) Threshold...	Minor	ws207 HCA-1 \# 1	2010-10-18 12:15:20	Hardware	Congested Bandwidth (in percents) threshold exceeded
Congested Bandwidth (%) Threshold...	Minor	ws206 HCA-1 \# 1	2010-10-18 12:15:16	Hardware	Congested Bandwidth (in percents) threshold exceeded
Congested Bandwidth (%) Threshold...	Minor	ws207 HCA-1 \# 1	2010-10-18 12:15:15	Hardware	Congested Bandwidth (in percents) threshold exceeded
Congested Bandwidth (%) Threshold...	Minor	ISR2004 \line_4 \10008110400315e49_2	2010-10-18 12:15:14	Hardware	Congested Bandwidth (in percents) threshold exceeded
Congested Bandwidth (%) Threshold...	Minor	ISR2004 \spine_1 \1000811040040554f_8	2010-10-18 12:15:14	Hardware	Congested Bandwidth (in percents) threshold exceeded
Congested Bandwidth (%) Threshold...	Minor	ws206 HCA-1 \# 1	2010-10-18 12:15:11	Hardware	Congested Bandwidth (in percents) threshold exceeded

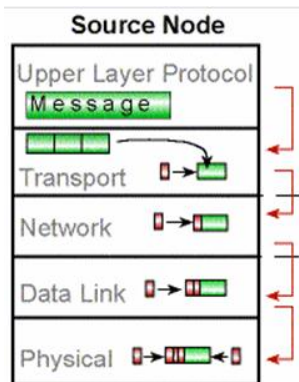
# InfiniBand Upper Layer Protocols



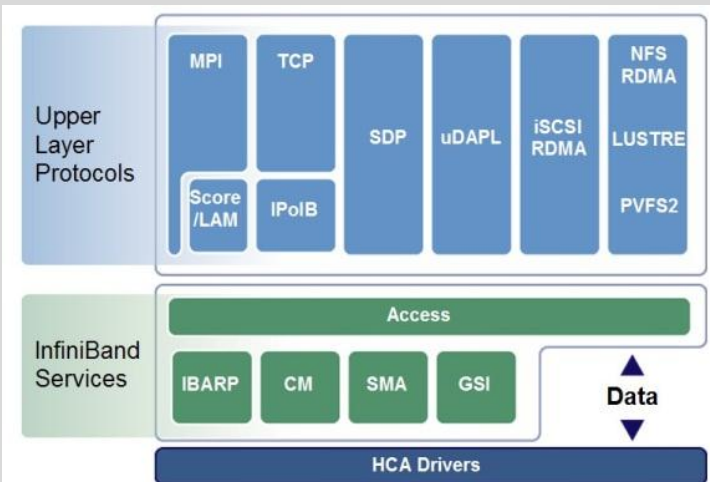


## ■ Communication Protocols & Interfaces

- IPoIB – IP over InfiniBand
  - Forwards IP traffic (TCP, UDP and IGMP)
- SDP – Sockets Direct Protocol
  - RDMA Off-load Socket Protocol (low CPU utilization)
- UDAPL – User level Direct Access Provider Library
  - Enables full kernel by-pass and use of native IB transport
- MPI – Message Passing Interface Library interface for **distributed/parallel computing**



## InfiniBand Processes within the Host KERNEL



- There are multiple drivers, existing in kernel and user space, involved in a connection. See Figure 2a.
- To explain it simply, much of the connection setup work, goes through the kernel driver, as speed is not a critical concern in that area.
- The user space drivers are involved in function calls such as `ibv post send` and `ibv post recv`.
- Instead of going through kernel space, they interact directly with the hardware by writing to a segment of mapped memory.
- Avoiding kernel traps is one way to decrease the overall latency of each operation.

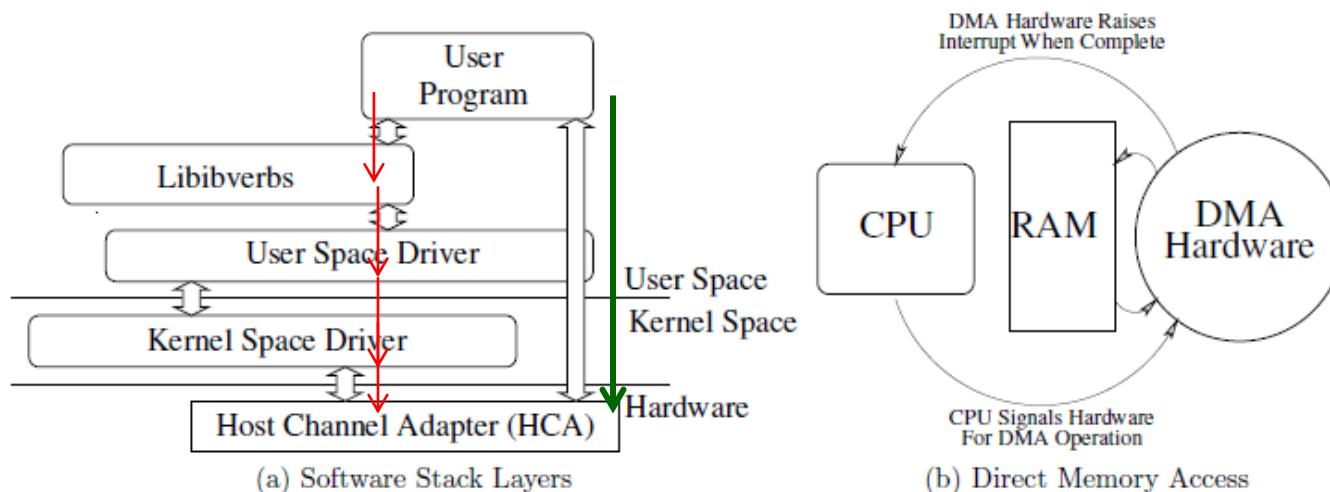
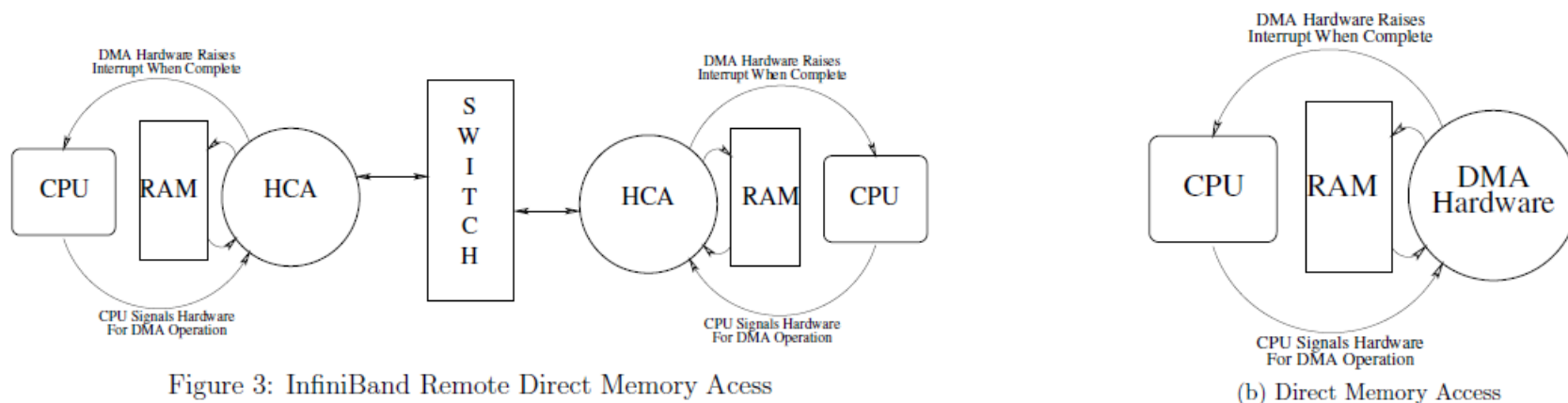


Figure 2: Layers and DMA

- Main key concepts in InfiniBand is Remote Direct Memory Access (RDMA).
- It allows a node to directly access the memory of another node on the subnet, without involving the remote CPU or software layers.
- Remember the key concepts of Direct Memory Access (DMA) as illustrated by Figure 2b.
- In the DMA, the CPU sends a command to the hardware to begin a DMA operation.
- When the operation finishes, the DMA hardware raises an interrupt with the CPU, signaling completion.



- The RDMA concept used in InfiniBand is similar to DMA, except with two nodes accessing each other's memory;
- One node is the sender and one is the receiver.
- Figure 2 illustrates an InfiBand connection. In this case the DMA Hardware is the Host Channel Adapter (HCA), and the two HCAs are connected, through a switch, to each other.
- The HCA is InfiniBand's version of a network card;
- it is the hardware local to each node that facilitates communications.
- This allows an HCA on one node to use another HCA to perform DMA operations on a remote node.

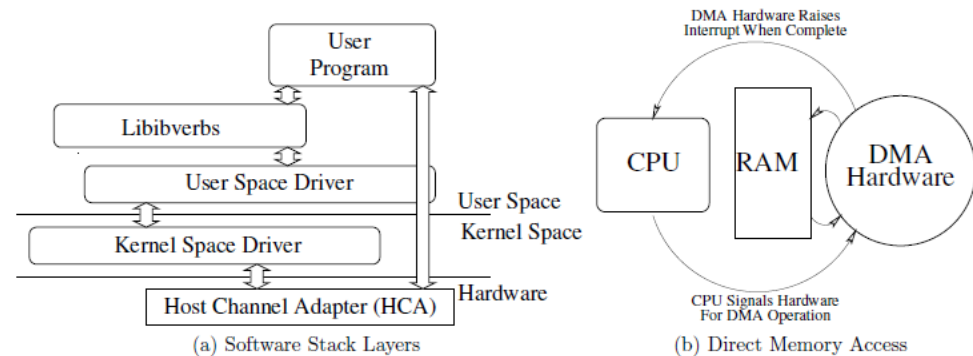


Figure 2: Layers and DMA

# Mellanox

## Host Channel Adapters (HCA)

This session has partially been discussed before  
in IB Basics – Filter accordingly

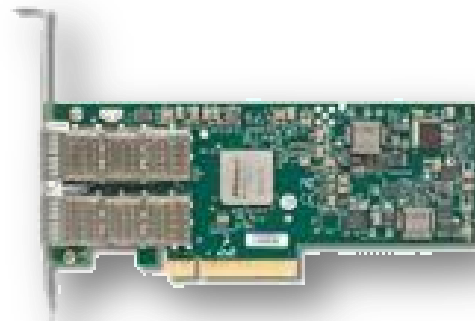
The logo for Mellanox ConnectX-3. It features the word "Connect" in a black serif font, followed by "X" in a large, stylized, orange and grey font, and "3" in a black serif font. A horizontal line with a dot at each end passes through the "X" and "3".

ConnectX-3



# ConnectX-3 InfiniBand Differentiation

- 1 $\mu$ s MPI ping latency
- Up to 56Gb/s InfiniBand or 40 Gigabit Ethernet per port
- PCI Express 3.0 (up to 8GT/s)
- CPU offload of transport operations
- Application offload
- GPU communication acceleration
- Precision Clock Synchronization
- End-to-end QoS and congestion control
- Hardware-based I/O virtualization
- Dynamic power management
- Fibre Channel encapsulation (FCoIB or FCoE)
- Ethernet encapsulation (EoIB)



Virtualization



HPC



Database



Cloud Computing



# Ethernet & IB Differences





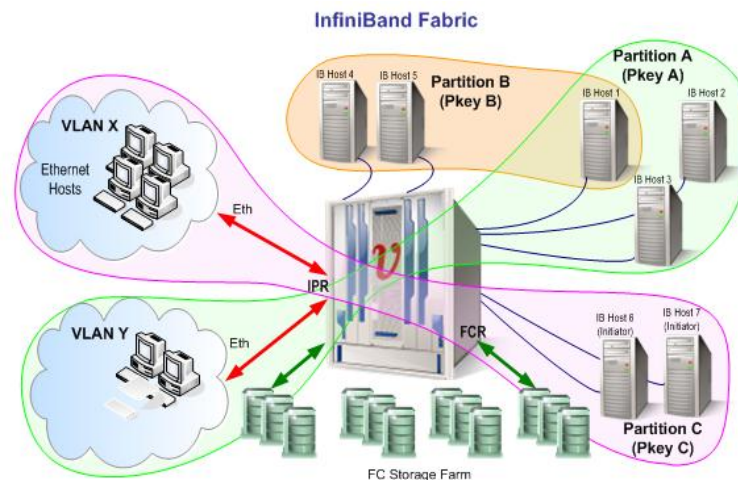
- Infiniband advantage is **very low latency**
  - IB is fast forward cut through only 2 frames of buffers in the port
  
- **Infiniband advantage Lossless Protocol Fabric**
  - Infiniband is a **credit based protocol** VS pause flow protocol
  
- **Reliability**
  - Ethernet has one CRC and **IB protocol has CRC on each layer**
  
- Infiniband designed for **Scalability with low administration cost**
  
- **The Fabric Subnet Manager does it all**
  - No need for spanning tree definitions (avoided by the SM algorithm)
  - No need for Link Aggregation Definitions Trunk Groups – Port Channels-LAGS
  - No need for routing definitions
  - There is no need for specific Network Packets Tagging
    - PKey will always be tag by default

- **Off loading HOST CPU resources** Using Remote **D**irect **M**emory **A**ccess
- **Extremely Cheaper cost per-port**
  - ~ app 100\$ VS 300 \$
- **No Forward Data Base Aging**
  - LID to Port Mapping (Like mac FDB aging )
  - A route map line is removed from the cache , only if a real change is identified and updated by the SM .  
no flooding , less cpu & memory tasks

# Fabric Partitioning

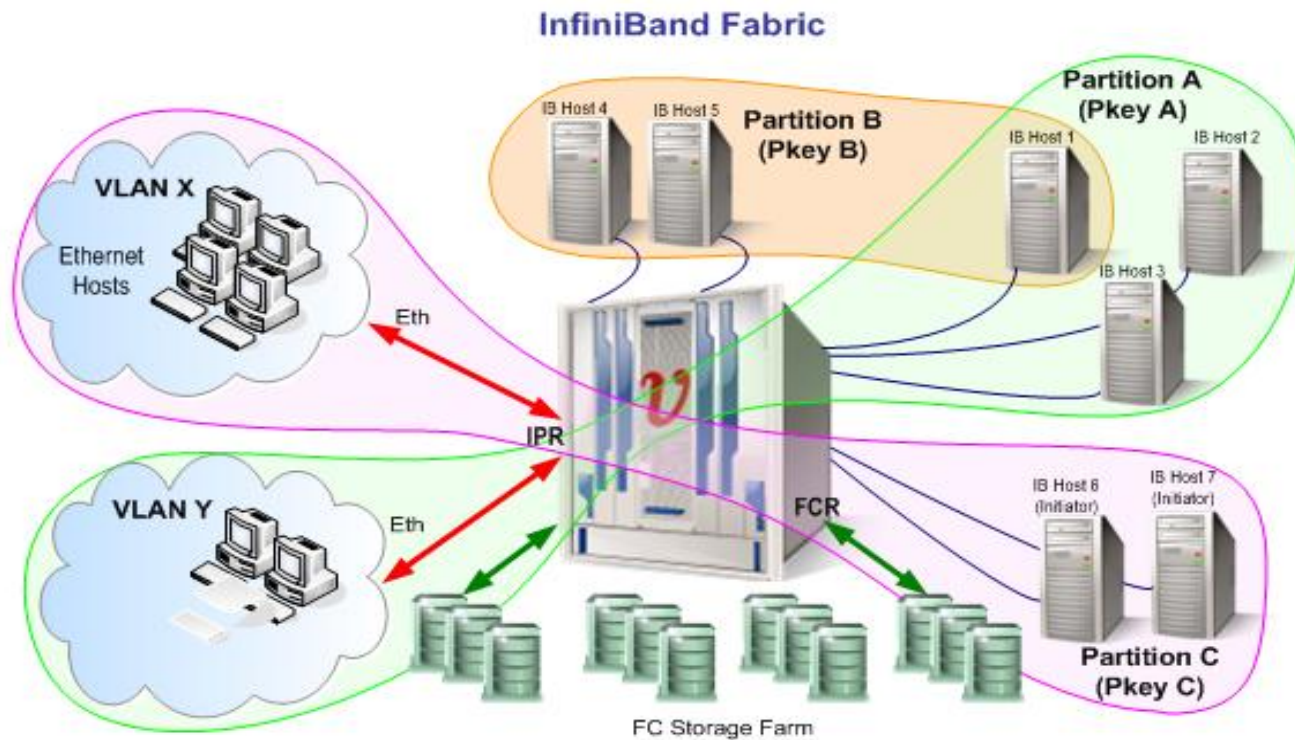


- Defines up to 64 Networks (partitions )
- Assign Server ports (Port GUID) to one or more networks
- Configure IPR interfaces enabling connectivity between Ethernet VLAN's and IB partitions.
- Provides a level of security with in the fabric. Full or Limited membership.
- Resources can be members of multiple networks.



# Partitioning - Pkey to VLAN mapping

- Define up to 64 partitions in a single 10G/4036E
- Partition by mapping port and Ethernet VLAN to InfiniBand PKEY



**Thank You**  
**[www.mellanox.com](http://www.mellanox.com)**

