InfiniBand For HPC Overview

HPC Advisory Council
Switzerland Workshop
March 21-23, 2011

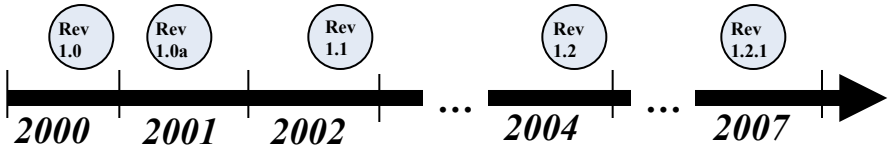
Erez Cohen - Sr. Director of Field Engineering

Mellanox TECHNOLOGIES

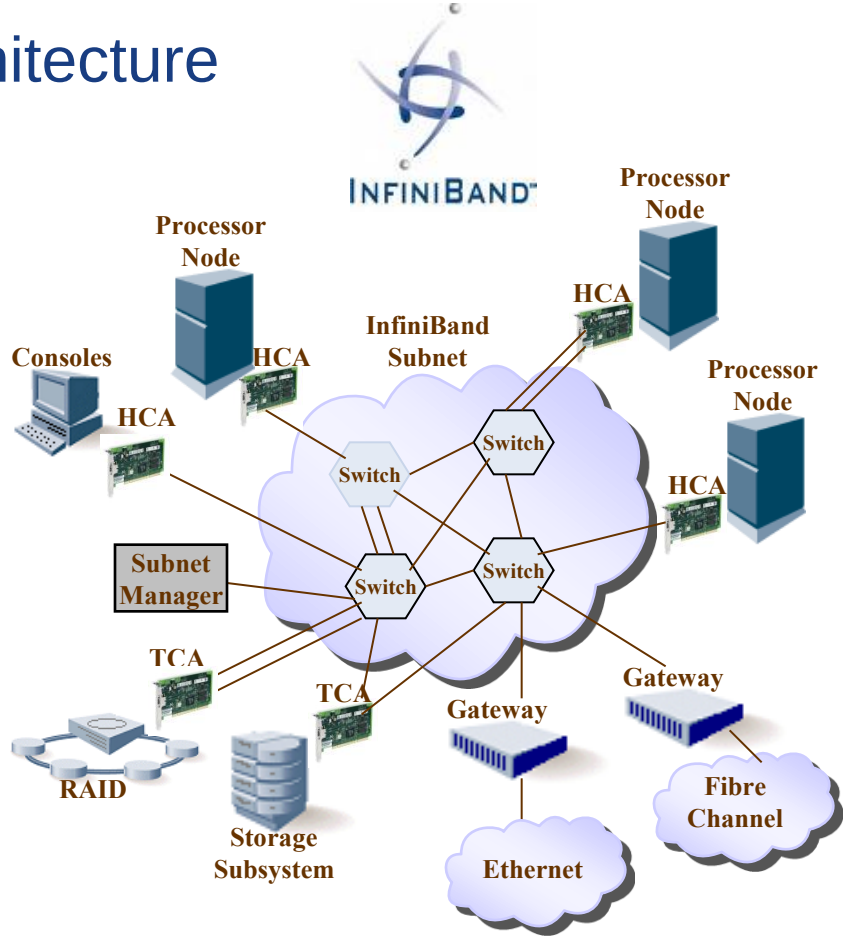
InfiniBand Overview

The InfiniBand Architecture

- 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)
  - Target Channel Adapters (TCA)
  - Switches
  - Routers
- Facilitated HW design for
  - Low latency / high bandwidth
  - Transport offload

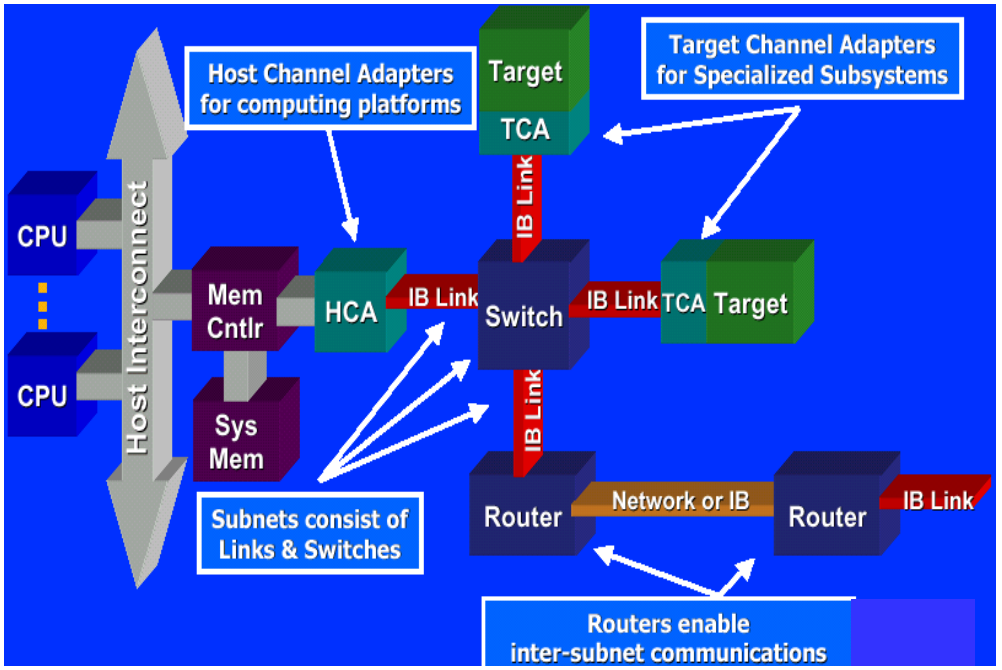


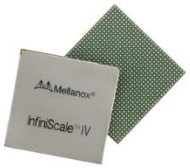
InfiniBand Feature Highlights

- Serial High Bandwidth Links
  - 56 Gb/s HCA links
  - Up to 120Gb/s switch-switch links
- Ultra low latency
  - Under 1 us application to application
- Reliable, lossless, self-managing fabric
  - Link level flow control
  - Congestion control to prevent HOL blocking
- Full CPU Offload
  - Hardware Based Transport Protocol
  - Reliable Transport
  - Kernel Bypass (User level applications get direct access to hardware)
- Memory exposed to remote node access
  - RDMA-read and RDMA-write
- Quality Of Service
  - Independent I/O channels at the adapter level
  - Virtual Lanes at the link level
- Cluster Scalability/flexibility
  - Up to 48K nodes in subnet, up to 2^{128} in network
  - Parallel routes between end nodes
  - Multiple cluster topologies possible
- Simplified Cluster Management
  - Centralized route manager
  - In-band diagnostics and upgrades

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
  - A device that transports packets between IBA subnets

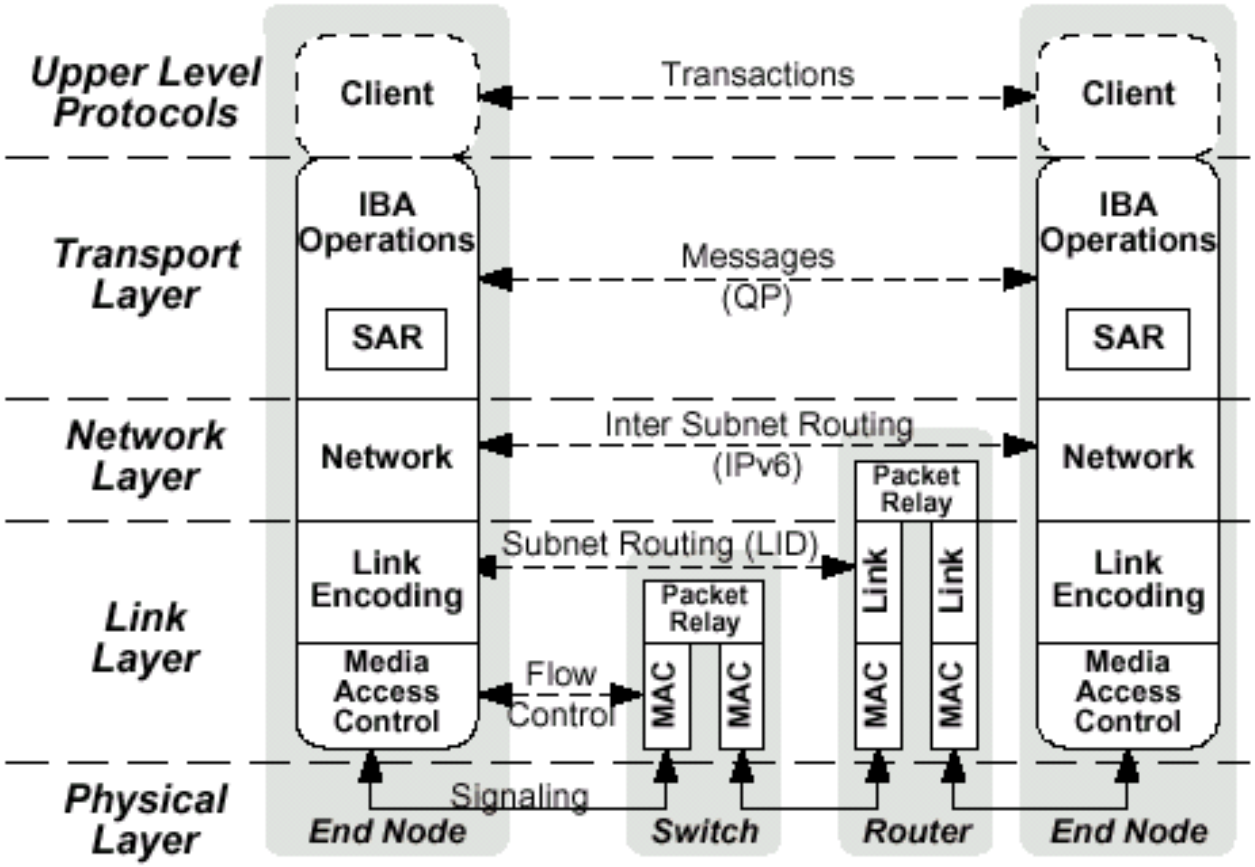




IB Architecture Layers

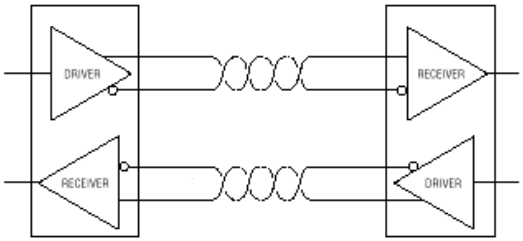
- **Physical**
  - Signal levels and Frequency; Media; Connectors
- **Link**
  - Symbols and framing; Flow control (credit-based); How packets are routed from Source to Destination
- **Network**
  - How packets are routed between subnets
- **Transport**
  - Delivers packets to the appropriate Queue Pair; 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 – Link Rate

- InfiniBand uses serial stream of bits for data transfer
- Link width
  - 1x – One differential pair per Tx/Rx
    - Not used today
  - 4x – Four differential pairs per Tx/Rx
    - Used on all Mellanox HCA, switch and cables
  - 12x - Twelve differential pairs per Tx and per Rx
    - Limited use



- Link Speed
  - Single Data Rate (SDR) – 2.5 Gb/s signaling (10-Gb/s for 4x)
  - Double Data Rate (DDR) – 5 Gb/s signaling (20-Gb/s for 4x)
  - Quad Data Rate (QDR) - 10 Gb/s signaling (40-Gb/s for 4x)
  - FDR - 14Gb/s signaling (56-Gb/s for 4x). 64/66 Encoding
  - EDR (25-Gb/lane) coming in near future
- Link rate
  - Multiplication of the link width and link speed
  - Most common shipping today is 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 for SDR, DDR and QDR
- 64/66 bit encoding for FDR
- Industry standard components
  - Copper cables / Connectors
  - Optical cables
  - Backplane connectors

4X QSFP

4x QSFP Fiber

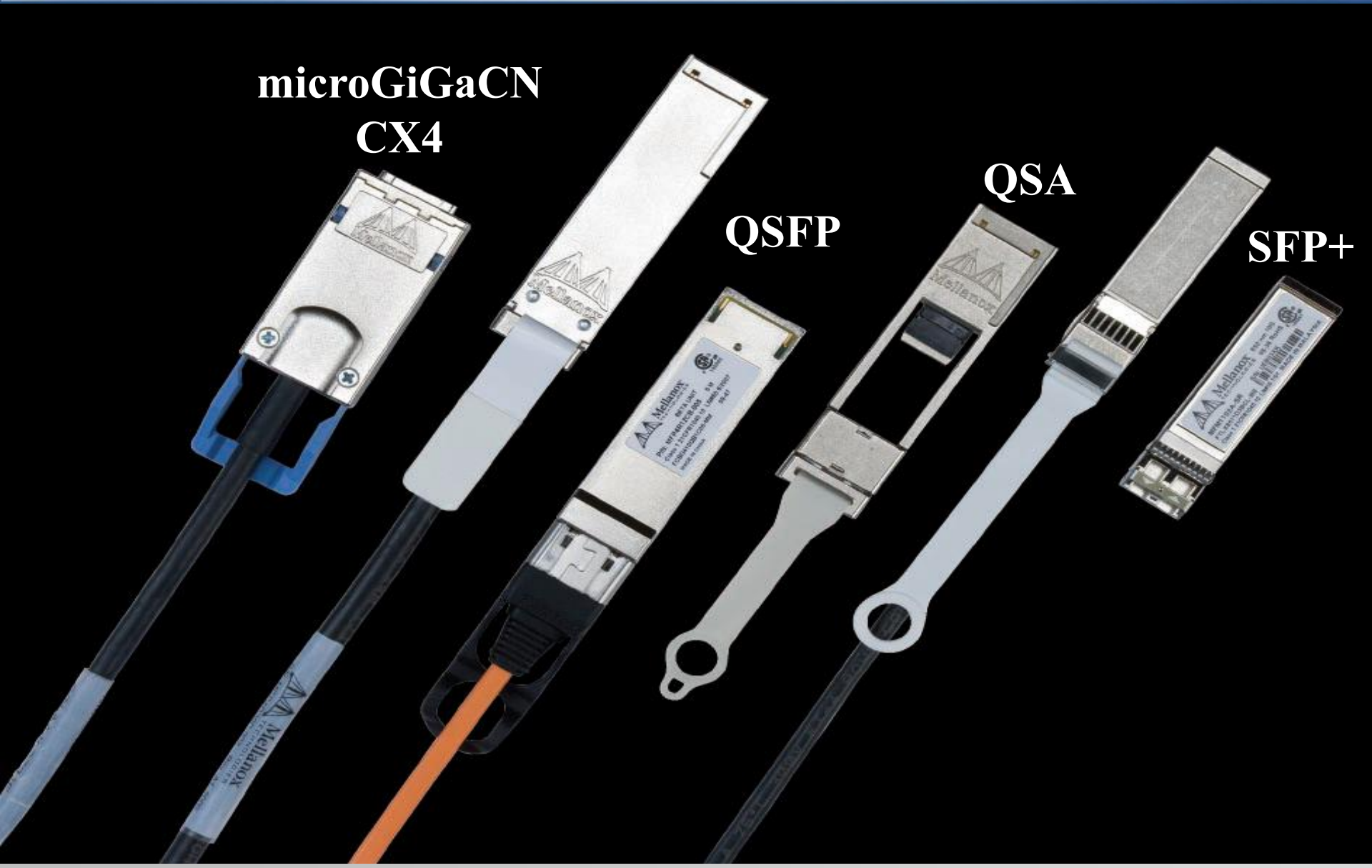
12X Cable

4x CX4 Fiber

4X CX4

FR4 PCB

Cable Interconnect



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

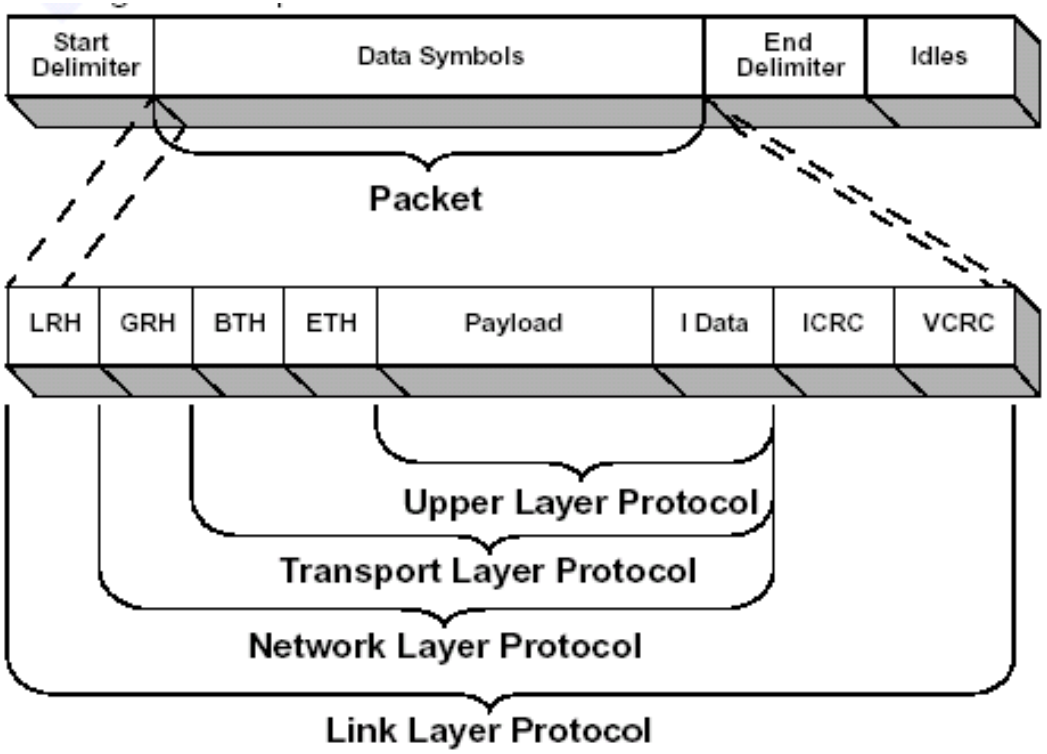


Figure 27 IBA Data Packet Format

Link Layer: Payload Size

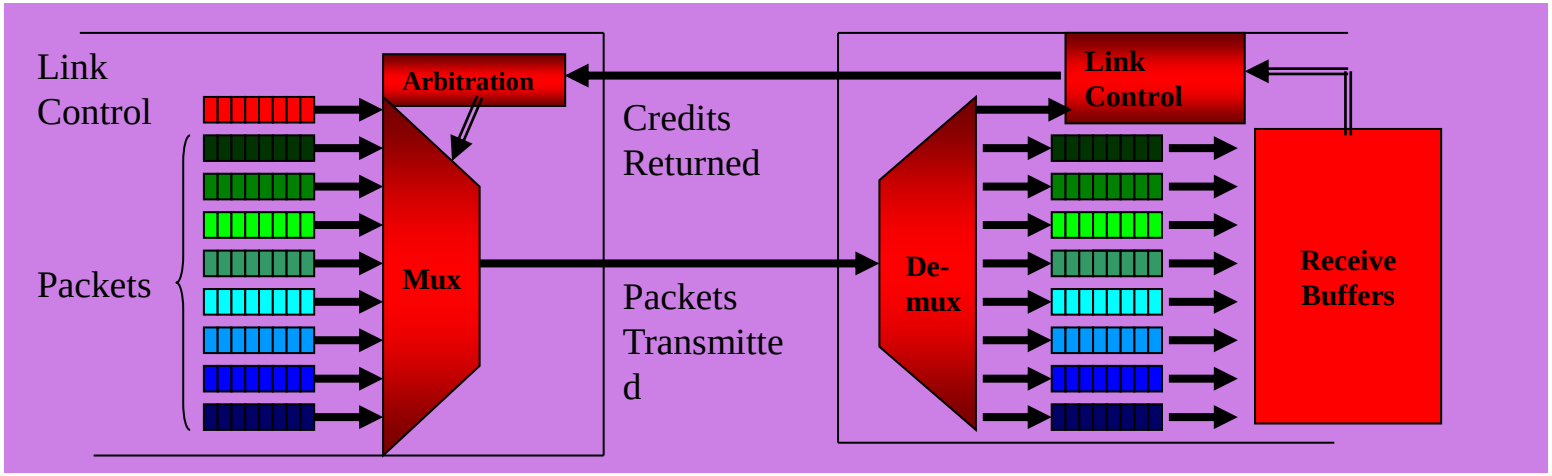
- Maximum Transfer Unit (MTU)
  - MTU allowed from 256 Bytes to 4K Bytes (Message sizes much larger).
  - Only packets smaller than or equal to the MTU are transmitted
  - Large MTU is more efficient (less overhead)
  - Small MTU gives less jitter
  - Small MTU preferable since segmentation/reassembly 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



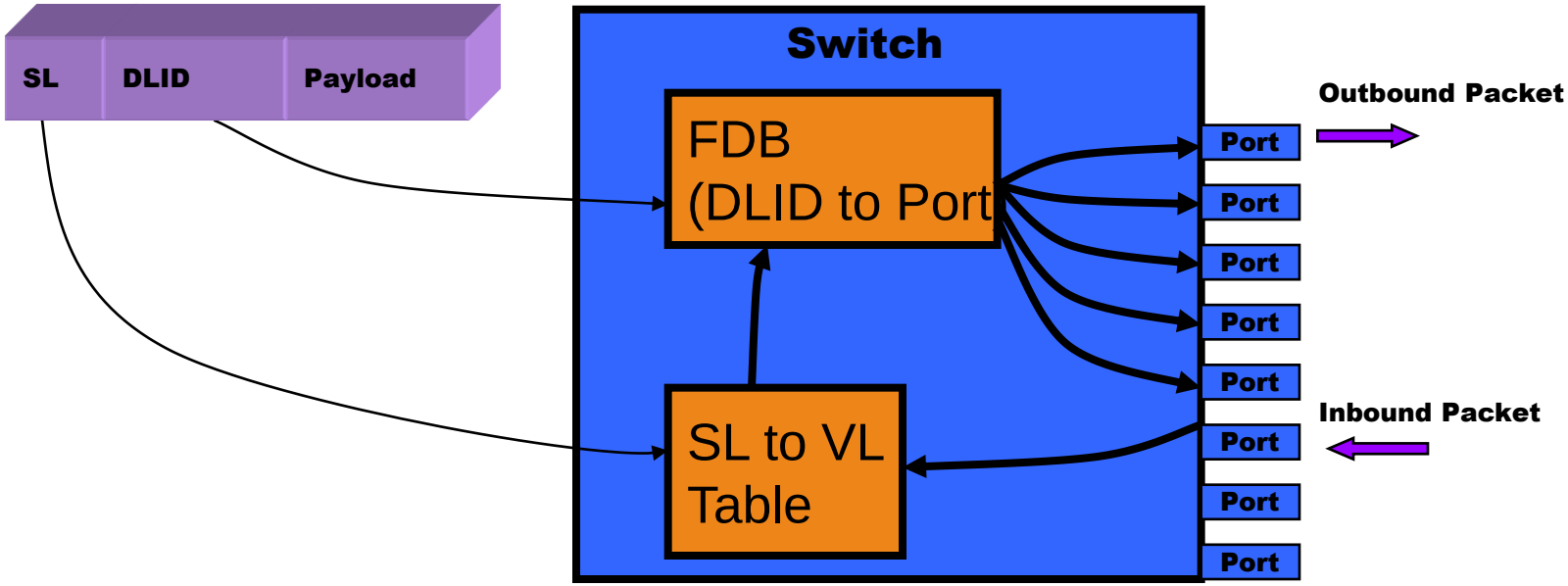
Link Layer: Addressing

- 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: Queue Pairs

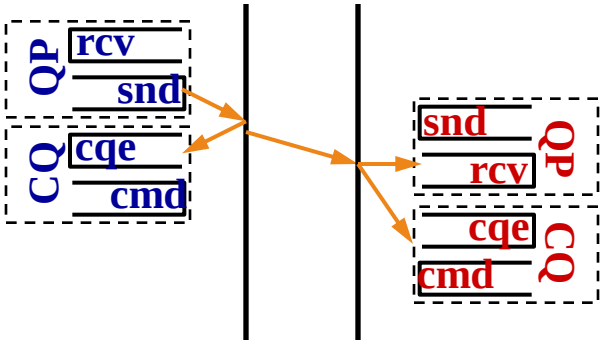
- QPs are in pairs (Send/Receive)
- Work Queue is the consumer/producer interface to the fabric
  - The Consumer/producer initiates a Work Queue Element (WQE)
  - The Channel Adapter executes the work request
  - The Channel Adapter notifies on completion or errors by writing a Completion Queue Element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

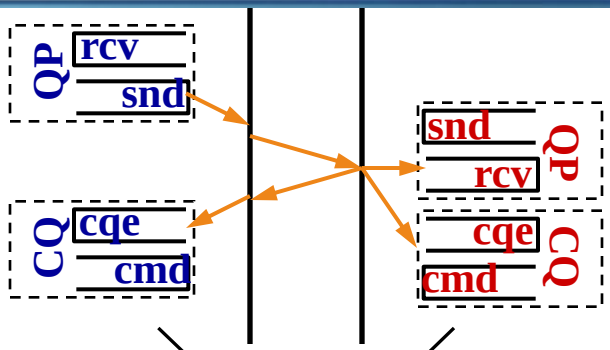
Transport Layer: Types Transfer Operations

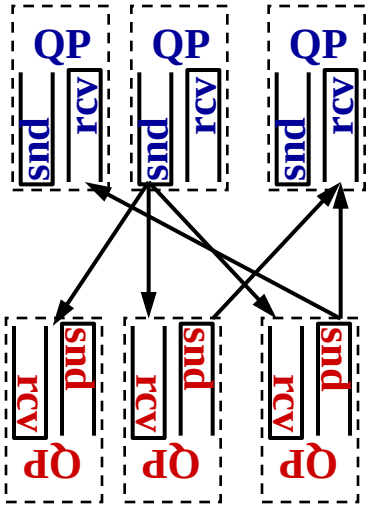
- SEND
  - Read message from HCA local system memory
  - Transfers data to Responder HCA Receive Queue logic
  - Does not specify where the data will be written in remote memory
  - Immediate Data option available
- RDMA Read
  - Responder HCA reads its local memory and returns it to the Requesting HCA
  - 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

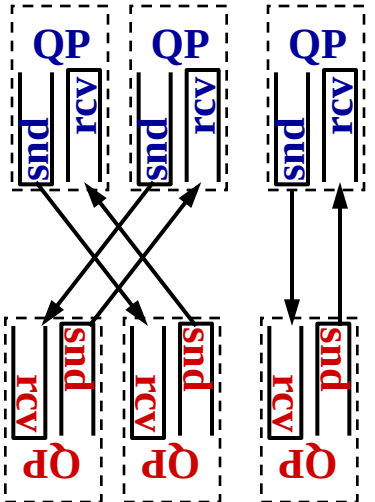
Transport Services

	Unreliable	Reliable
Non-connected	UD	RD XRC
Connected	UC	RC





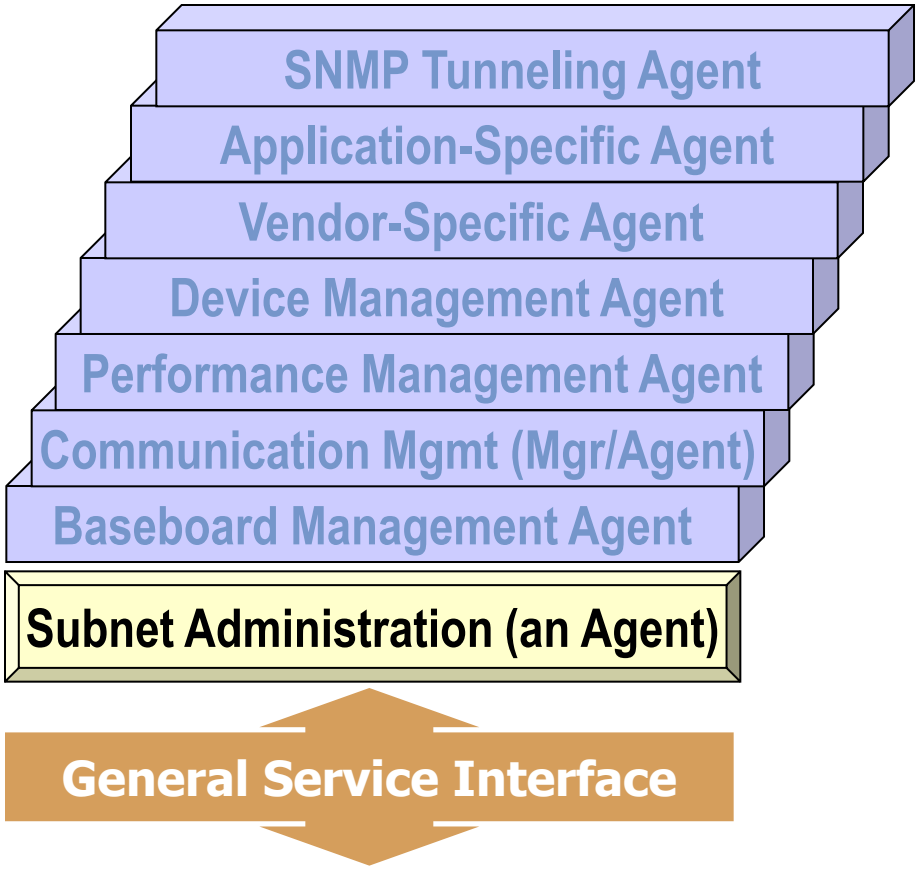




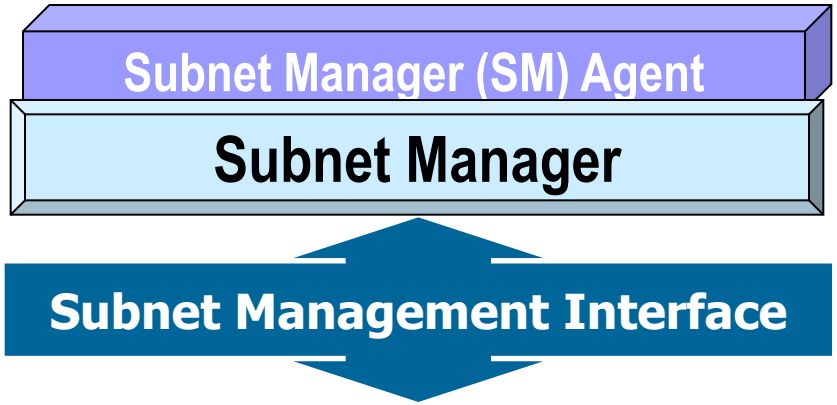
Verbs

- 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



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



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

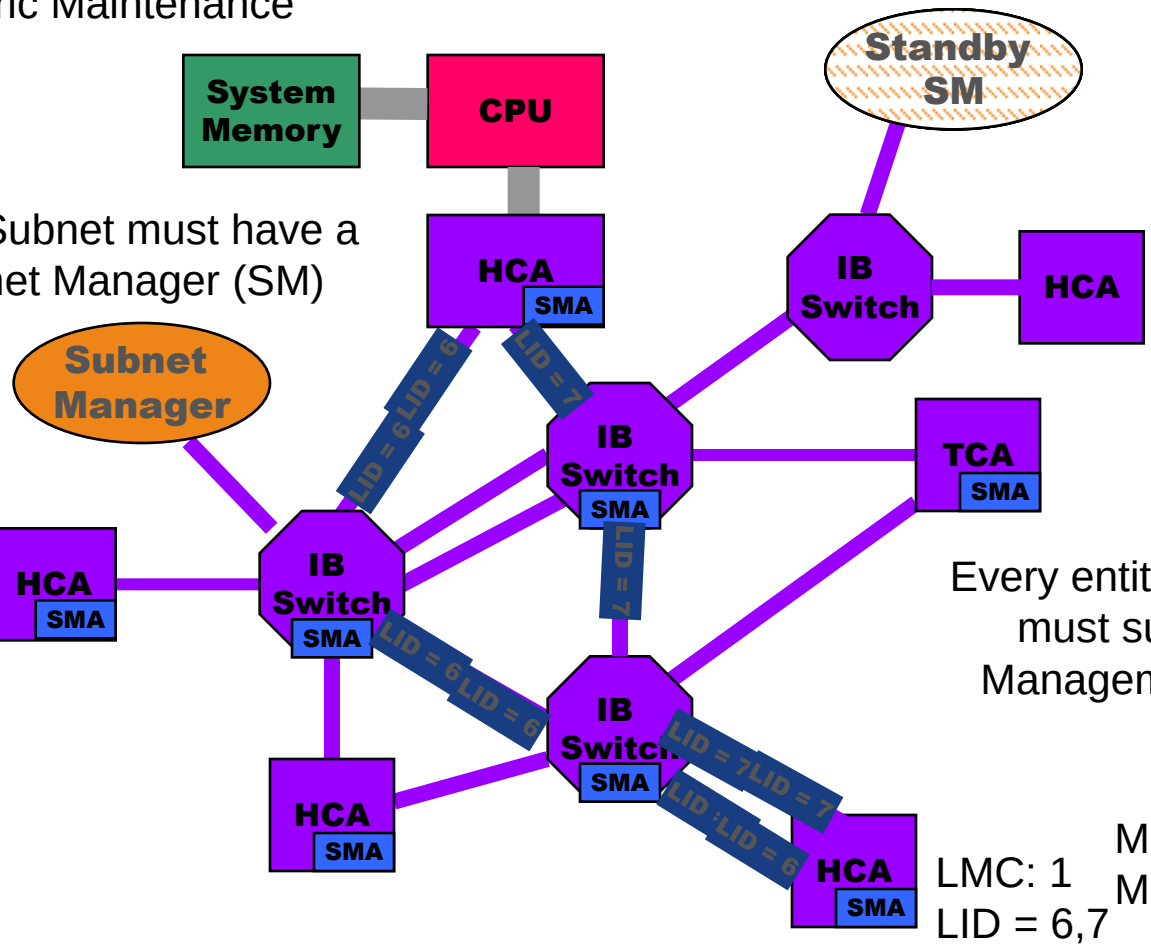
Subnet Management

Topology Discovery
FDB Initialization
Fabric Maintenance

Initialization uses
Directed Route MADs:

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
LID = 6,7

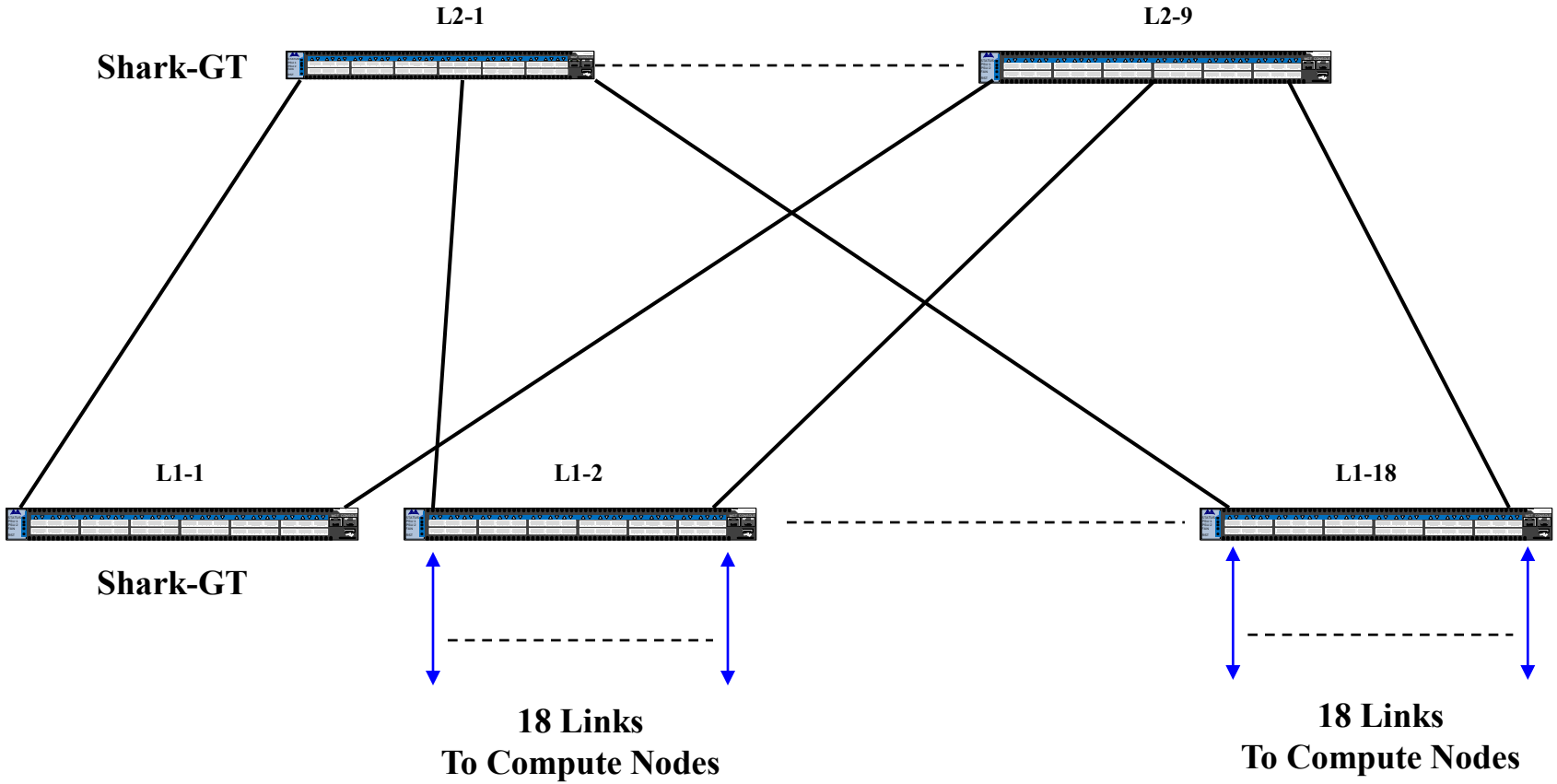
Multipathing: LMC Supports Multiple LIDS

InfiniBand Cluster Topologies

Cluster Topologies

- Two topologies are mainly in use for large clusters
  - Fat-Tree (most popular topology in HPC)
  - 3D Torus
- Fat-tree characteristics:
  - Use same BW for all links (or close BW)
  - Many times use same number of ports for all switches
  - Many configurations are possible
  - But they are all only “Rearrangeably Non Blocking”
    - For any permutation of src/dst pairs exists non-blocking routing
- Main issues with fabric design
  - Is the SM capable of routing the fabric?
  - Does it generate credit loops?
  - Are the paths evenly distributed?

324 Node Full FAT Tree using MTS5030 (max 648 ports)



2 x 4X QDR Uplinks

1 x 4X QDR Uplinks

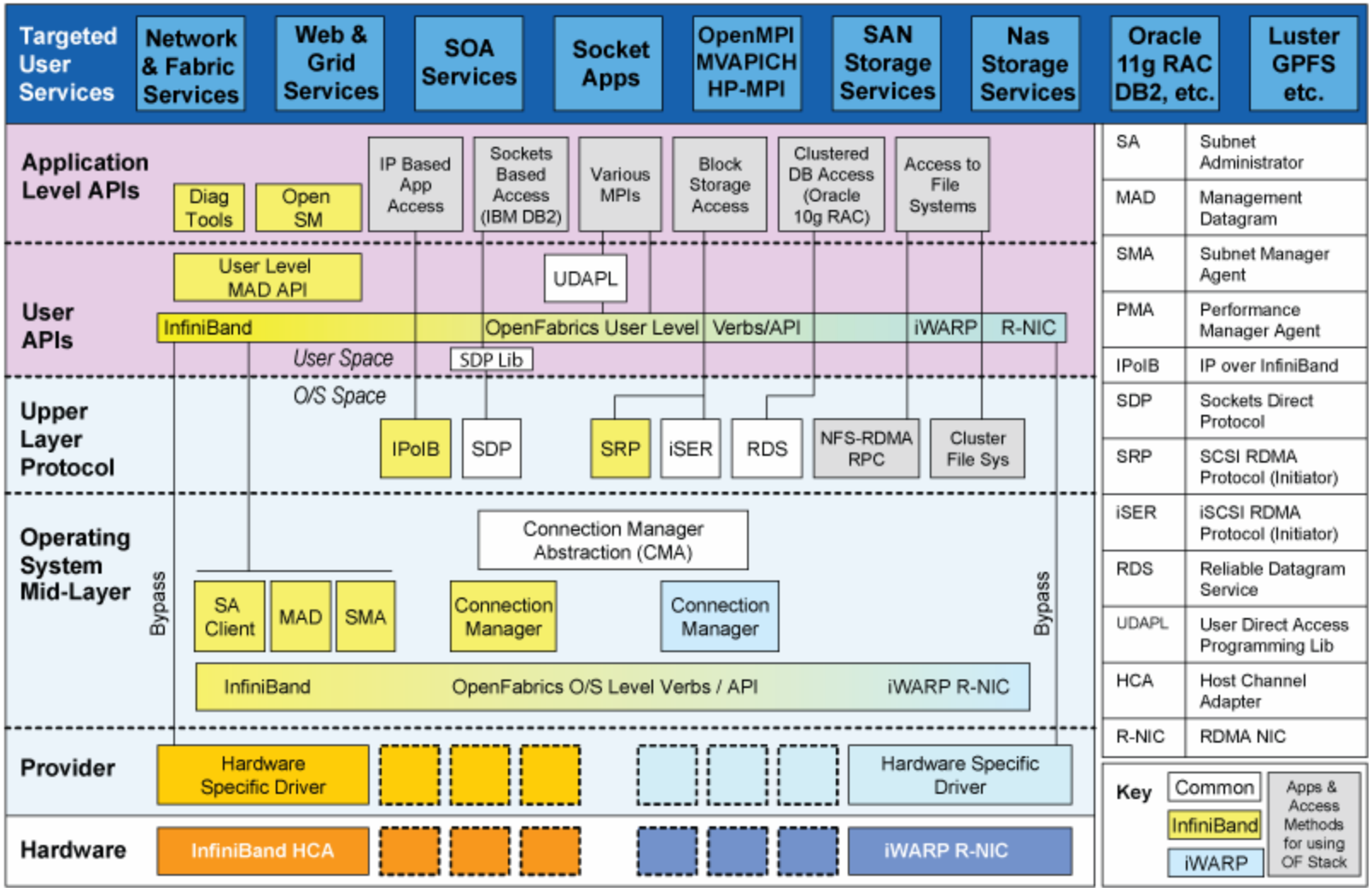
InfiniBand Linux SW Stack

MLNX_OFED

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
- Mellanox OFED is a single Virtual Protocol Interconnect (VPI) software stack based on the OFED stack
  - Operates across all Mellanox network adapters
  - Supports:
    - SDR, DDR, QDR and FDR InfiniBand
    - 10Gb/s Ethernet (10GigE)
    - Fiber Channel over Ethernet (FCoE)
    - 2.5 or 5.0 GT/s PCI Express 2.0

The SW stack


Targeted User Services
Network & Fabric Services
Web & Grid Services
SOA Services
Socket Apps
OpenMPI MVAPICH HP-MPI
SAN Storage Services
Nas Storage Services
Oracle 11g RAC DB2, etc.
Luster GPFS etc.
Application Level APIs
Diag Tools
Open SM
IP Based App Access
Sockets Based Access (IBM DB2)
Various MPIs
Block Storage Access
Clustered DB Access (Oracle 10g RAC)
Access to File Systems
User APIs
User Level MAD API
UDAPL
InfiniBand
OpenFabrics User Level Verbs/API
iWARP
R-NIC
User Space
SDP Lib
O/S Space
Upper Layer Protocol
IPoIB
SDP
SRP
iSER
RDS
NFS-RDMA RPC
Cluster File Sys
Operating System Mid-Layer
Bypass
Connection Manager Abstraction (CMA)
SA Client
MAD
SMA
Connection Manager
Connection Manager
Bypass
InfiniBand
OpenFabrics O/S Level Verbs / API
iWARP R-NIC
Provider
Hardware Specific Driver
Hardware Specific Driver
Hardware
InfiniBand HCA
iWARP R-NIC
SA
Subnet Administrator
MAD
Management Datagram
SMA
Subnet Manager Agent
PMA
Performance Manager Agent
IPoIB
IP over InfiniBand
SDP
Sockets Direct Protocol
SRP
SCSI RDMA Protocol (Initiator)
iSER
iSCSI RDMA Protocol (Initiator)
RDS
Reliable Datagram Service
UDAPL
User Direct Access Programming Lib
HCA
Host Channel Adapter
R-NIC
RDMA NIC
Key
Common
InfiniBand
iWARP
Apps & Access Methods for using OF Stack

MLNX_OFED Installation

- Pre-built RPM install.
  - 1. mount -o rw,loop MLNX_OFED_LINUX-*.iso /mnt
  - 2. cd /mnt
  - 3. ./mlnxofedinstall
- Building RPMs for un-supported kernels.
  - 1. mount -o rw,loop MLNX_OFED_LINUX-*.iso /mnt
  - 2. cd /mnt/src
  - 3. cp OFED-*.tgz /root (this is the original OFED distribution tarball)
  - 4. tar zxvf OFED-*.tgz
  - 5. cd OFED-*
  - 6. copy ofed.conf to OFED-* directory
  - 7. ./install.pl -c ofed.conf

OpenSM - Features

- 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
- Congestion Control
- Adaptive Routing
- 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

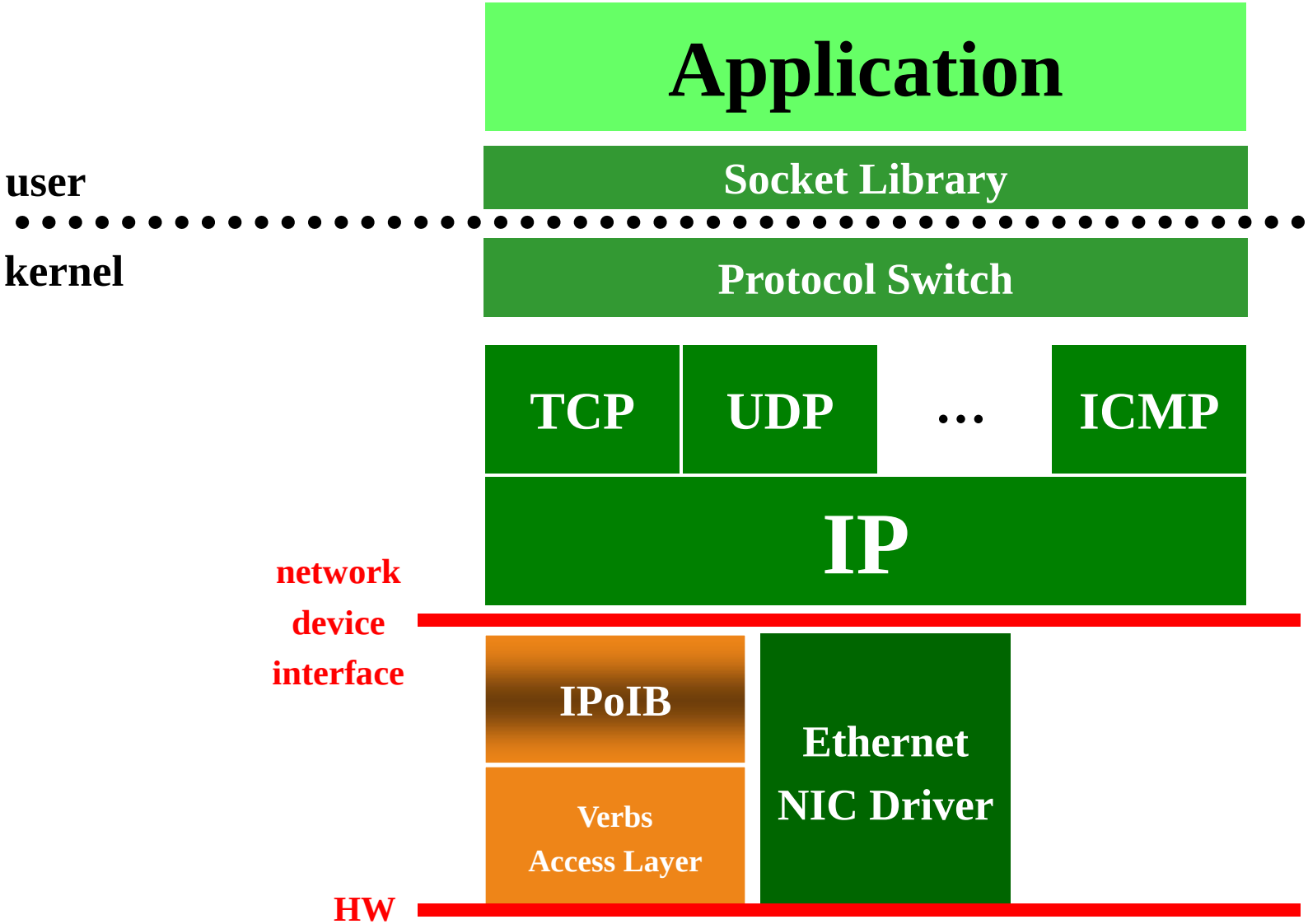
```
# ONBOOT
#  To start OpenSM automatically set ONBOOT=yes
ONBOOT=yes
```

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

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



MPI

- A message passing interface
- Used for point to point communication
  - MPI_I/SEND, MPI_I/RECV
- 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)

Hands On

InfiniBand For HPC – Hands On

- Set up
  - 2 servers with ConnectX HCA running SLES 11
  - 8 port QDR IB switch based on InfiniScale 4 switch silicon
- Steps
  - Identify OFED package
  - Install OFED package
  - Configure IPoIB interface
  - Run OpenSM
  - Check HCA status
  - Test IPoIB (ping)
  - Run MPI test without IB
  - Run BW and Latency tests over IB

Thank You

www.mellanox.com