InfiniBand Architecture Overview

David A. Deming
Solution Technology

Topics

- Documentation and Trade Association
- IB architectural components
- IB protocol layers
- IB enhancements
- Physical layer
- Link layer
- Network layer
- Transport layer
- ULP
- Software Interface
- Management

InfiniBand specifications – volumes

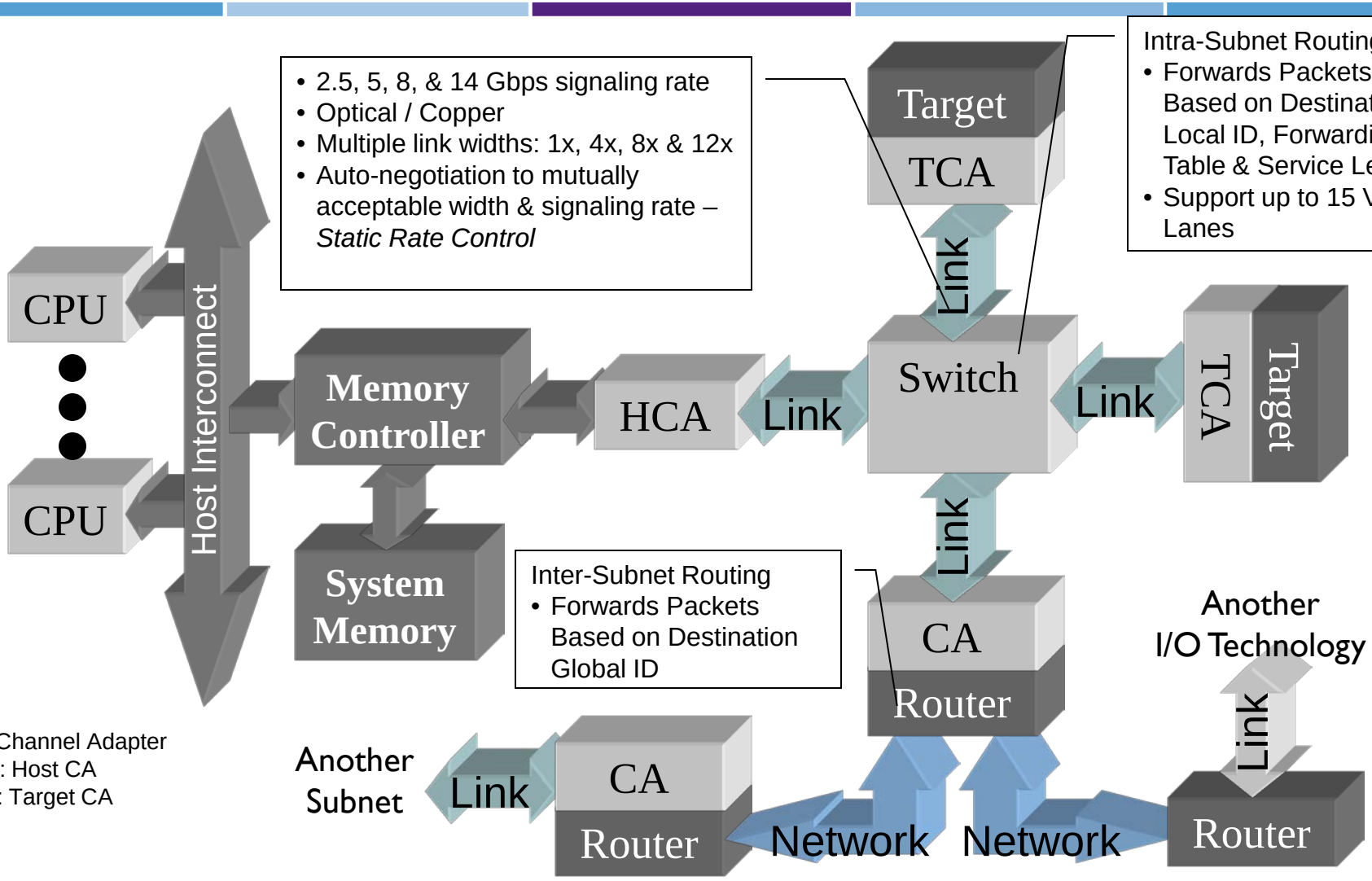
- Volume 1 - 1727 pages – November 2007 Release 1.2.1
  - Specifies the core IBA
  - It provides normative information required for IBA operation for switches, host channel adapters for processor nodes, target channel adapters for I/O devices, routers, and management functions.
- Volume 2 - 833 pages – October 2006
  - Specifies electrical and mechanical configurations for a number of different physical media and signaling rates, mechanical form factors, and physical chassis management requirements.
- Volume 3 – was changed to Annexes
  - Specifies policies, recommended practices and examples for applying IBA
  - 1 through 13 included in the 1.2.1 specification
  - 14, 15, & 16 not included in 1.2.1 specification

Annex's

#	Title	Date	Description
11	RDMA IP CM Service	September 2006	Defines the RDMA IP CM Service which provides support for a socket-like connection model for RDMA-aware ULPs
12	Support for iSCSI Extensions for RDMA (iSER)	September 2006	Adds the RDMA data transfer capability to iSCSI by layering iSCSI on top of an RDMA-Capable Protocol
13	Quality of Service (QoS)	November 2007	Defines the framework for managing Quality of Service in an InfiniBand fabric
14	Extended Reliable Connected (XRC) Transport Service	March 2009	Defines protocol that allows significant savings in the number of QPs required to establish all-to-all process connectivity in large clusters
15	Hierarchy Information	July 2009	Defines the framework for discovering the hierarchy of an InfiniBand fabric
16	RDMA over Converged Ethernet (RoCE)	April 2010	Defines how IBA transport protocol can be sent across L2 Ethernet infrastructure

IB architectural components

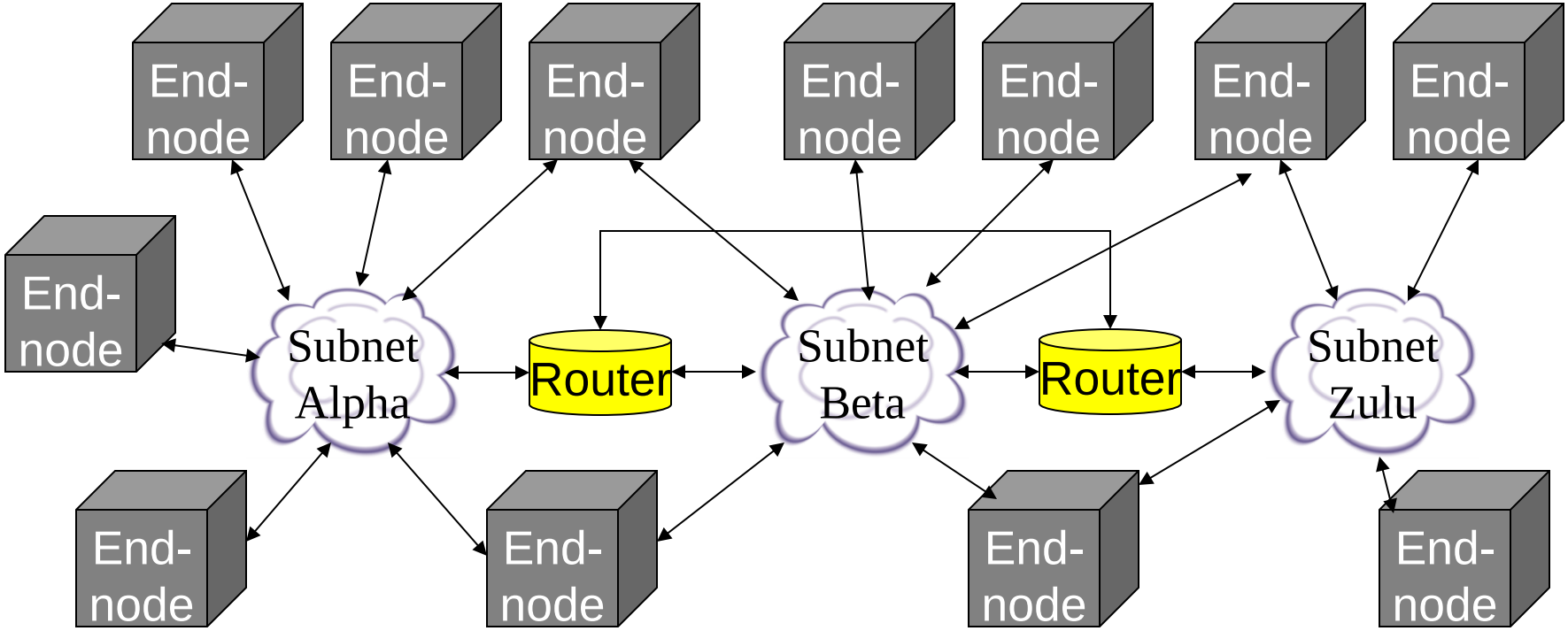
InfiniBand Architecture Model



System Area Network

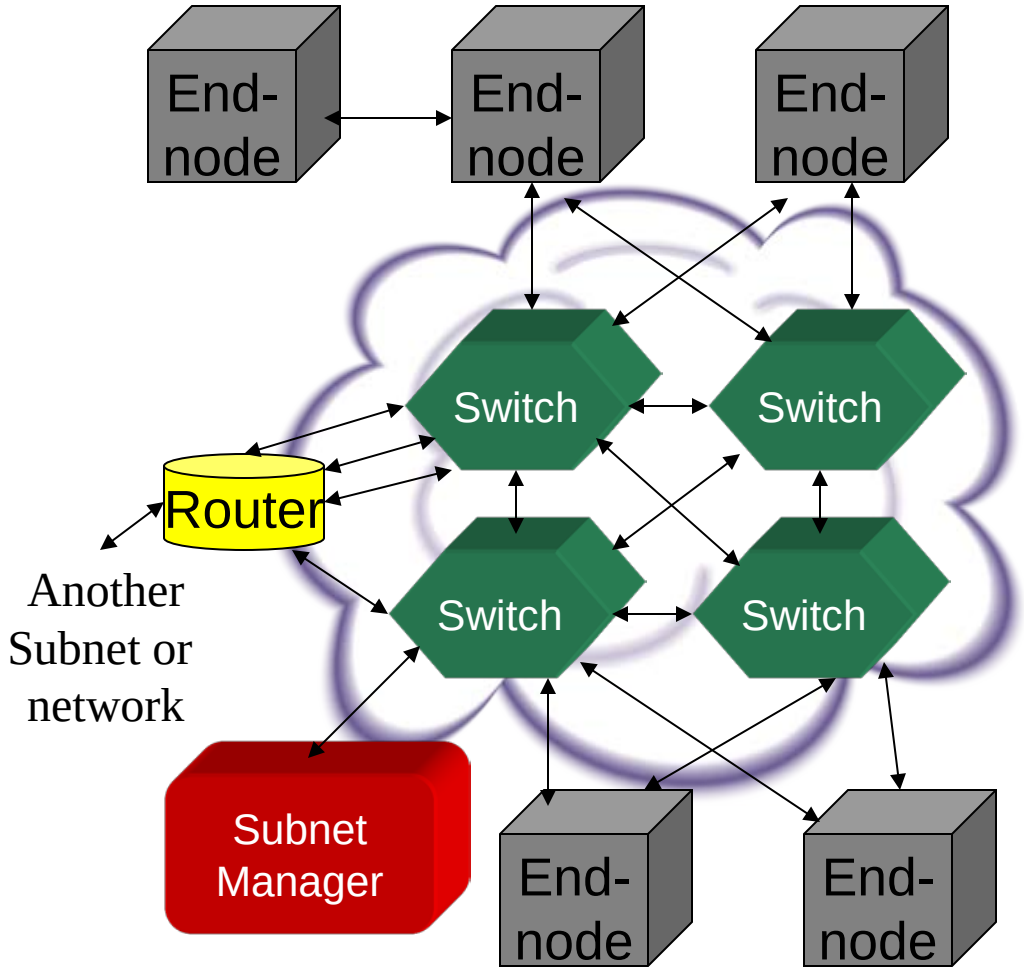
- The IBA defines a System Area Network (SAN) for connecting
  - multiple independent processor platforms (i.e., host processor nodes),
  - I/O platforms
  - I/O devices
- The IBA SAN is a communications and management infrastructure supporting
  - I/O communications
  - inter-processor communications (IPC) for one or more computer systems

IB Network and Subnets



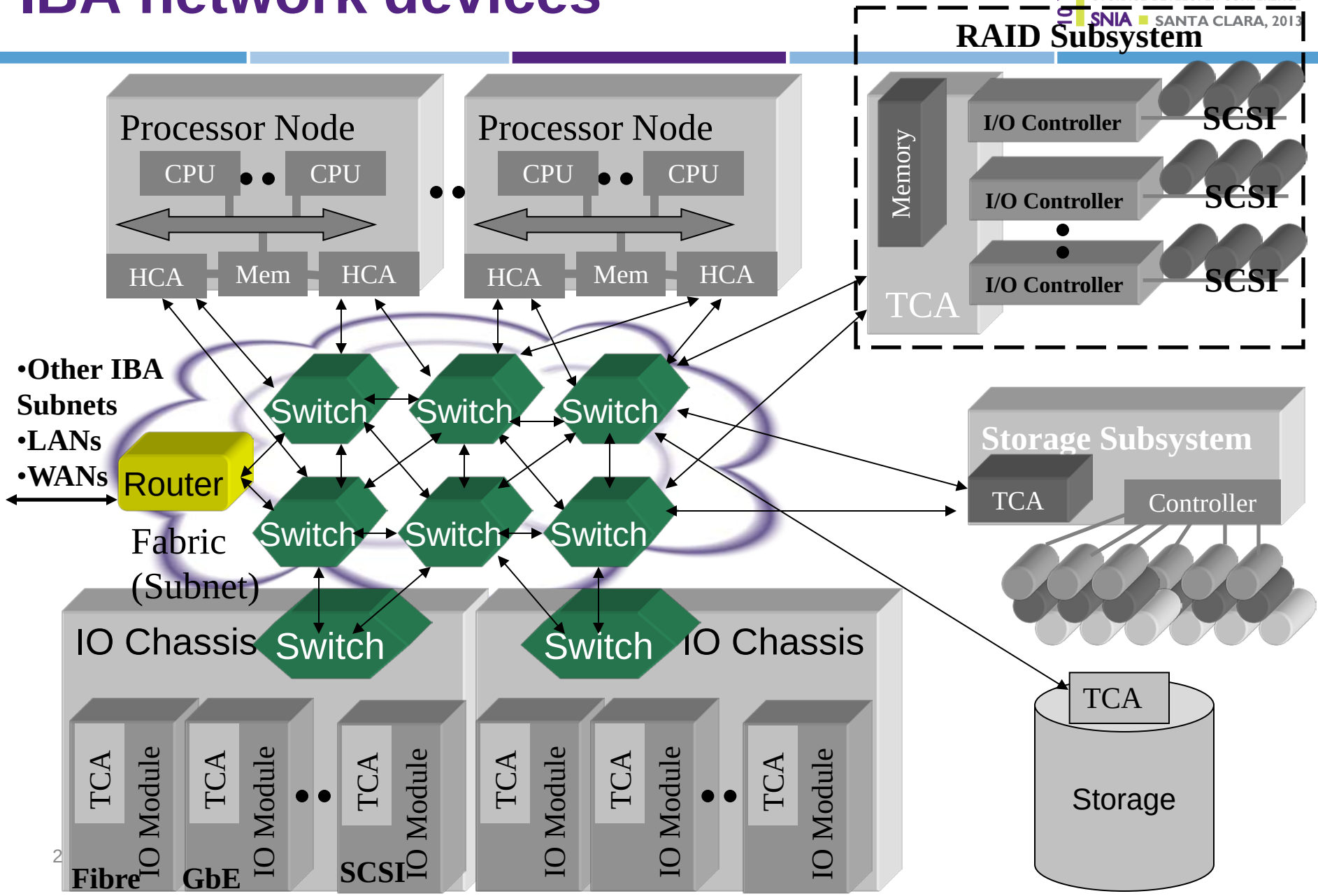
- The IBA network is subdivided into subnets that are interconnected by routers.
- Endnodes can attach to a single subnet or multiple subnets.

Subnet Components



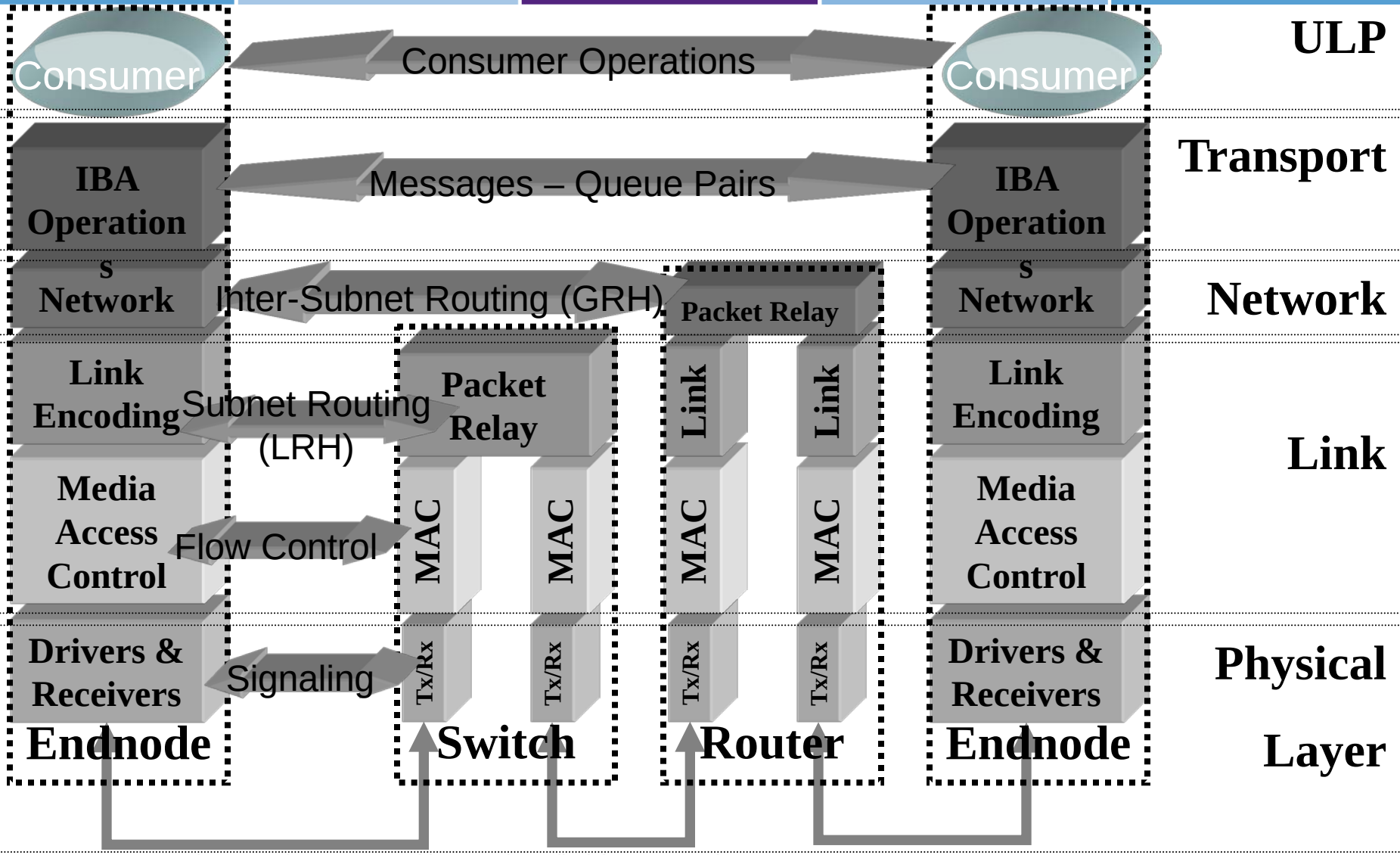
- An IBA subnet is composed of endnodes, switches, routers, and subnet managers.
- Each IBT device may attach to a single switch or multiple switches and/or directly with each other.
- Multiple links can exist between any two IBT devices.

IBA network devices



Protocols and IBA protocol layers

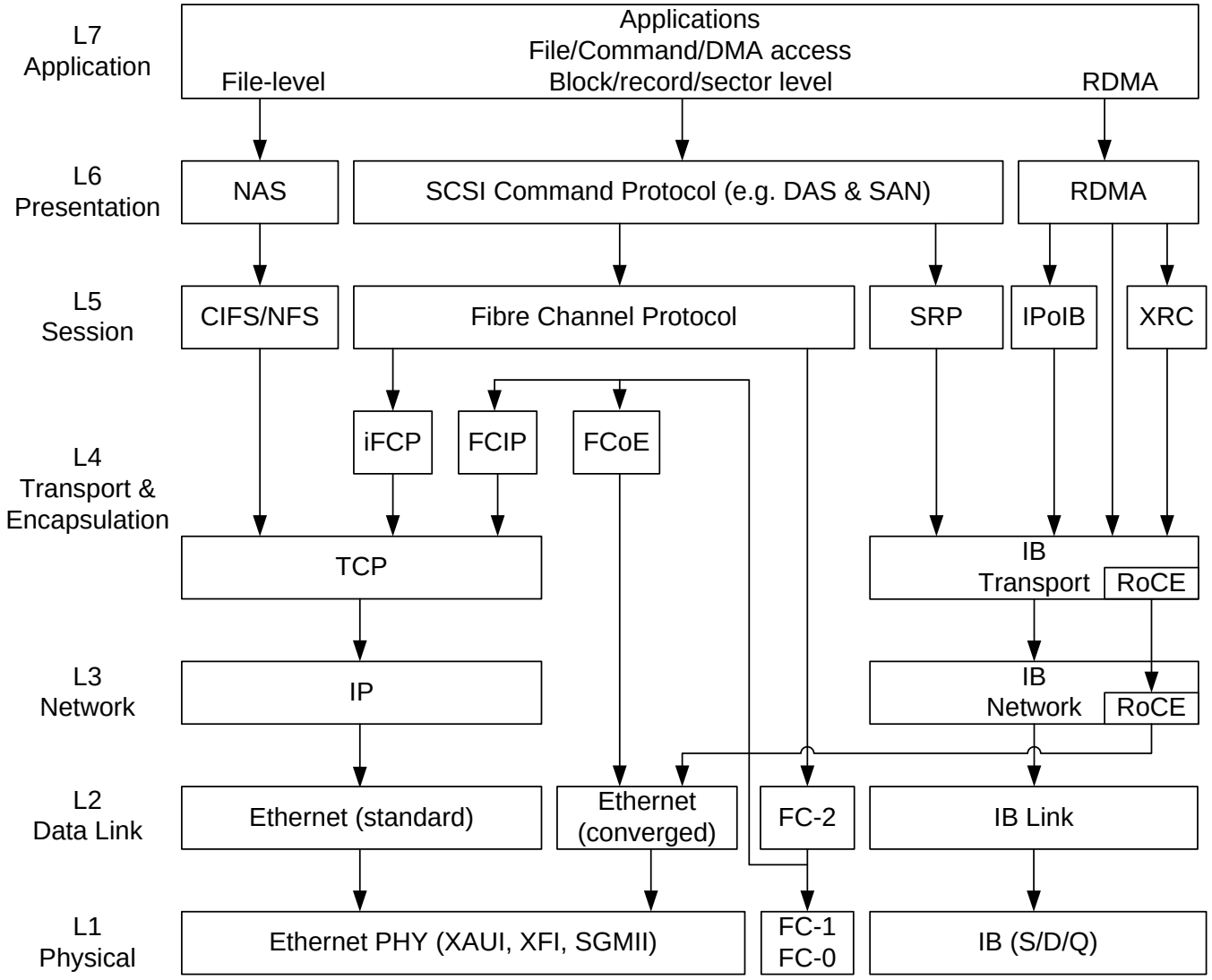
IB Protocol Layers



IB protocol stack

- ULP (Upper Layer Protocol)
  - consists of any consumer application e.g. storage, server, database, etc...
- Transport layer
  - provides transport services and functions
  - determines header and packet formats
  - provides reliability mechanisms, error detection, and error handling
- Network layer
  - provides addressing characteristics
  - defines routers and packet relay between subnets
  - partitioning and multicast capabilities
- Link layer
  - link states, link and data packet checks
  - virtual lanes, flow control and static rate injection
- Physical layer
  - provides link and physical interface including encoding/decoding
  - order set definition, packet framing, link initialization

Protocol Stack Comparison

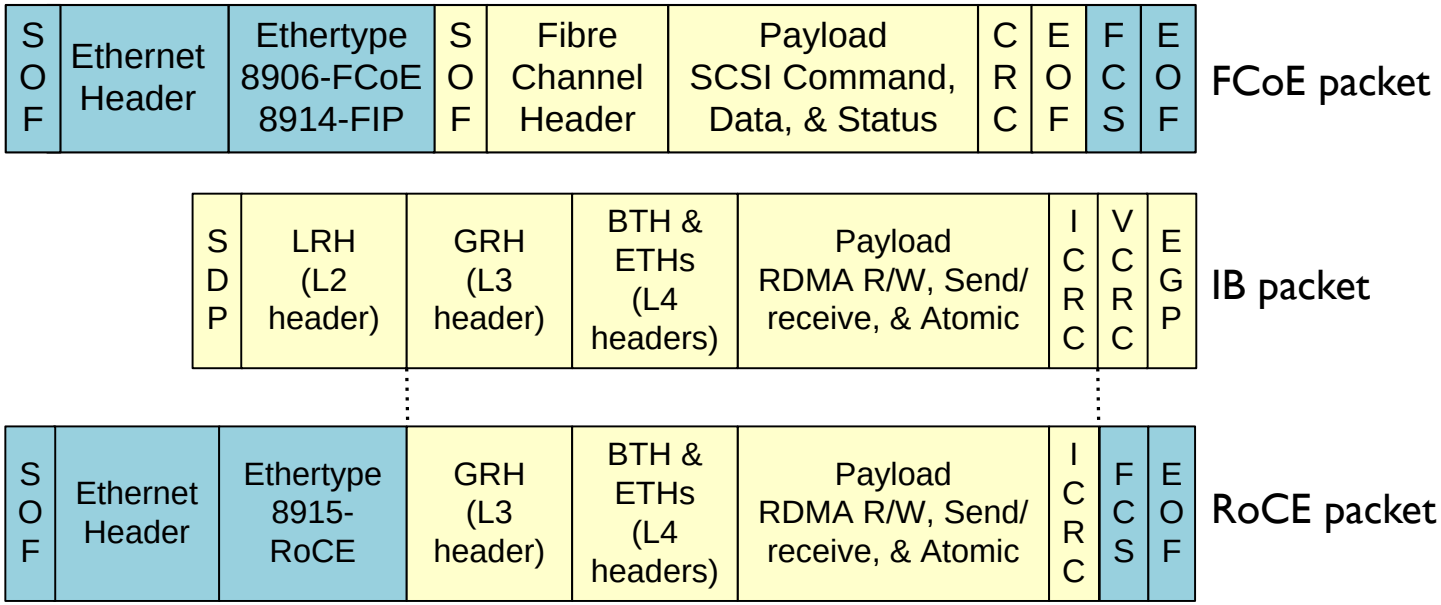


IB enhancements
RoCE, XRC, and FDR

3.7 RDMA over Converged Ethernet (RoCE)

Protocol allows IB to be sent across CEE infrastructures.

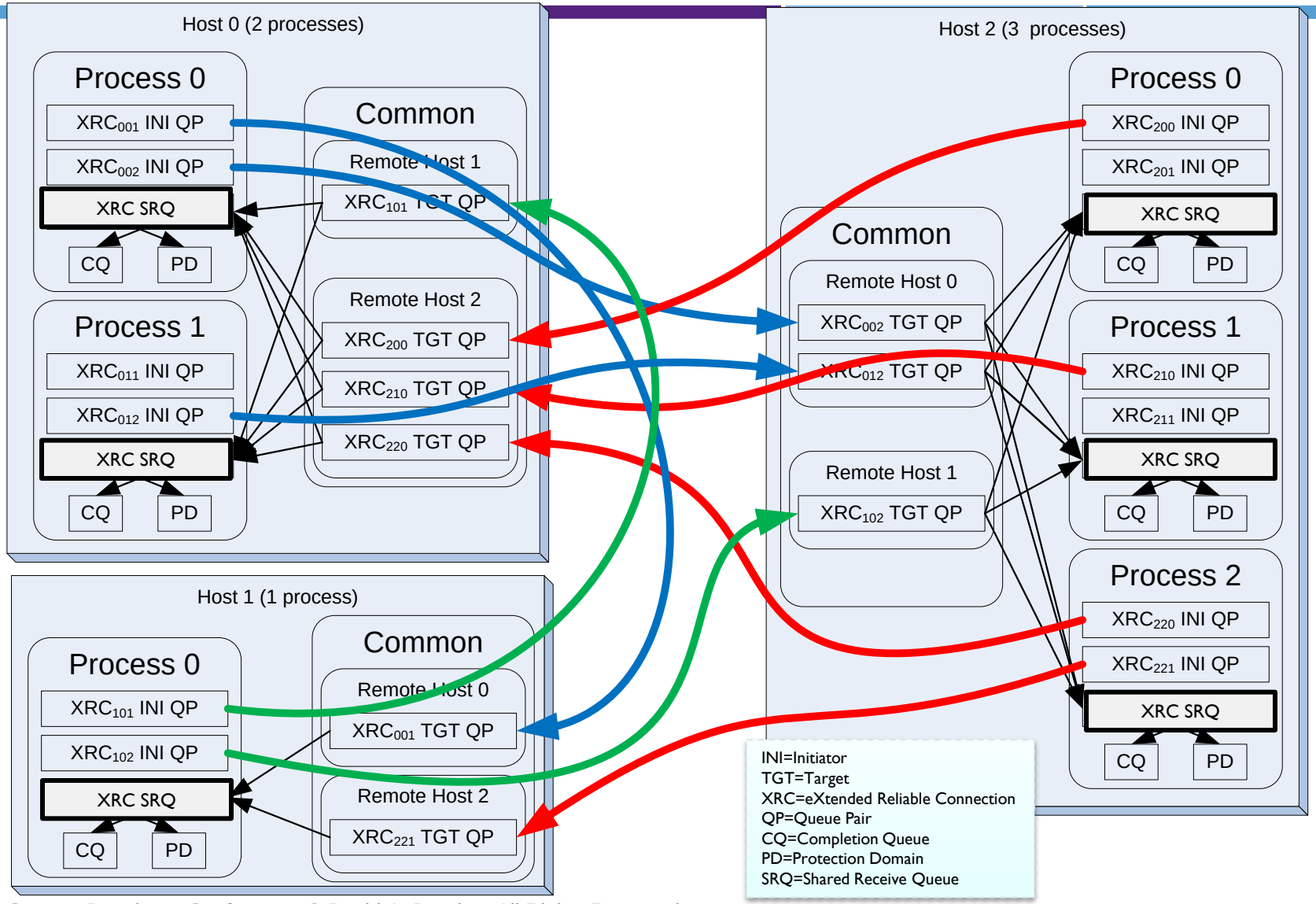
- The concept is identical to FCoE
  - Encapsulate FC frame in Ethernet frame
- Making Ethernet "lossless" with IEEE reliability mechanisms
  - Priority-based flow control (802.1Qbb)
  - Congestion notification (802.1Qau)
  - Enhanced transmission selection (802.1Qaz)



Extended Reliable Connected Services (XRC)

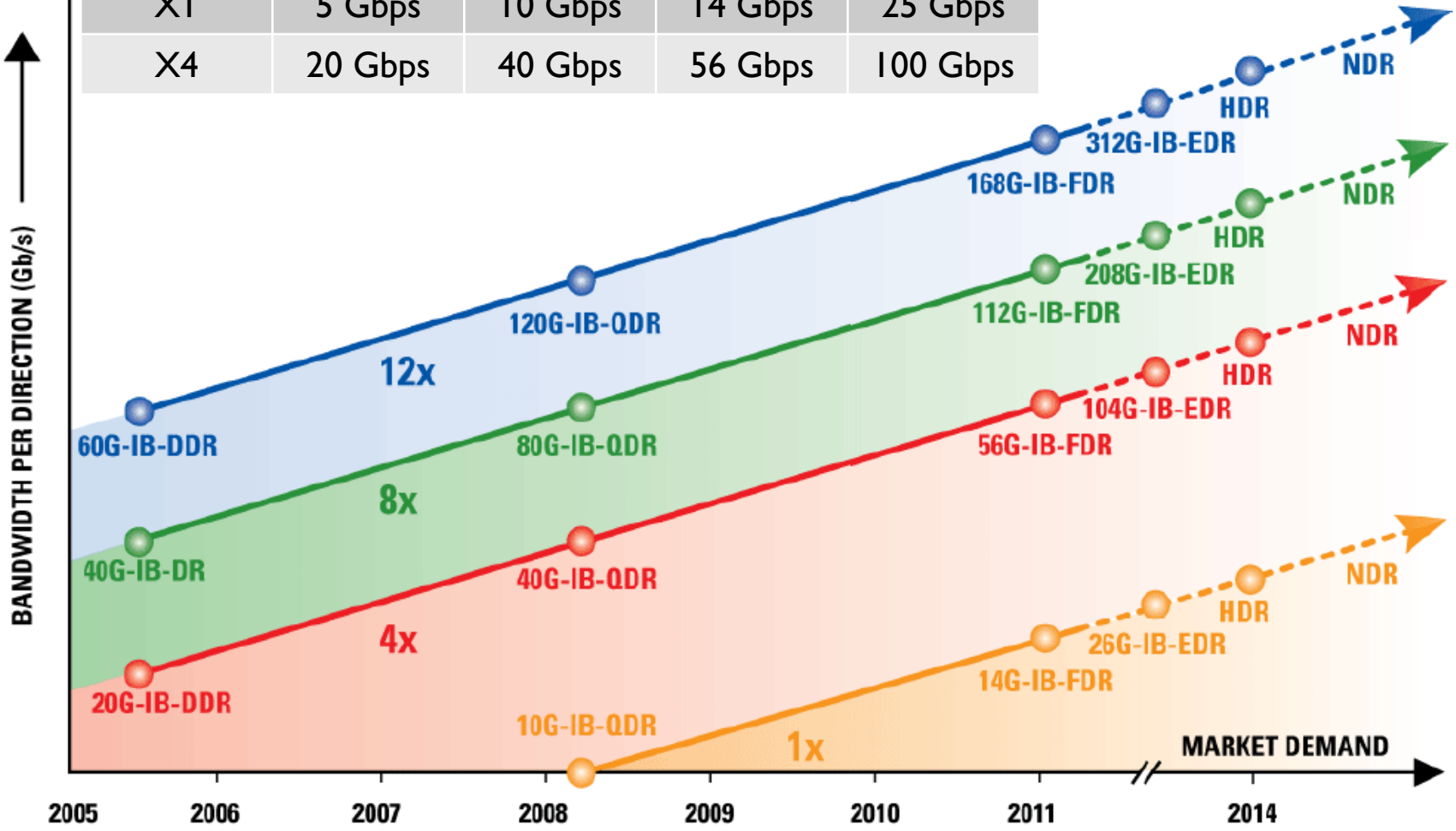
- XRC allows significant savings in the number of QPs required to establish all-to-all process connectivity in large clusters.
  - The established trend in multicore processors will result in a direct increase in the number of processes that run on each endnode of an IB connected cluster.
  - Multi core node systems are very common today with roadmaps showing even more cores per node in the not so distant future.

XRC Model



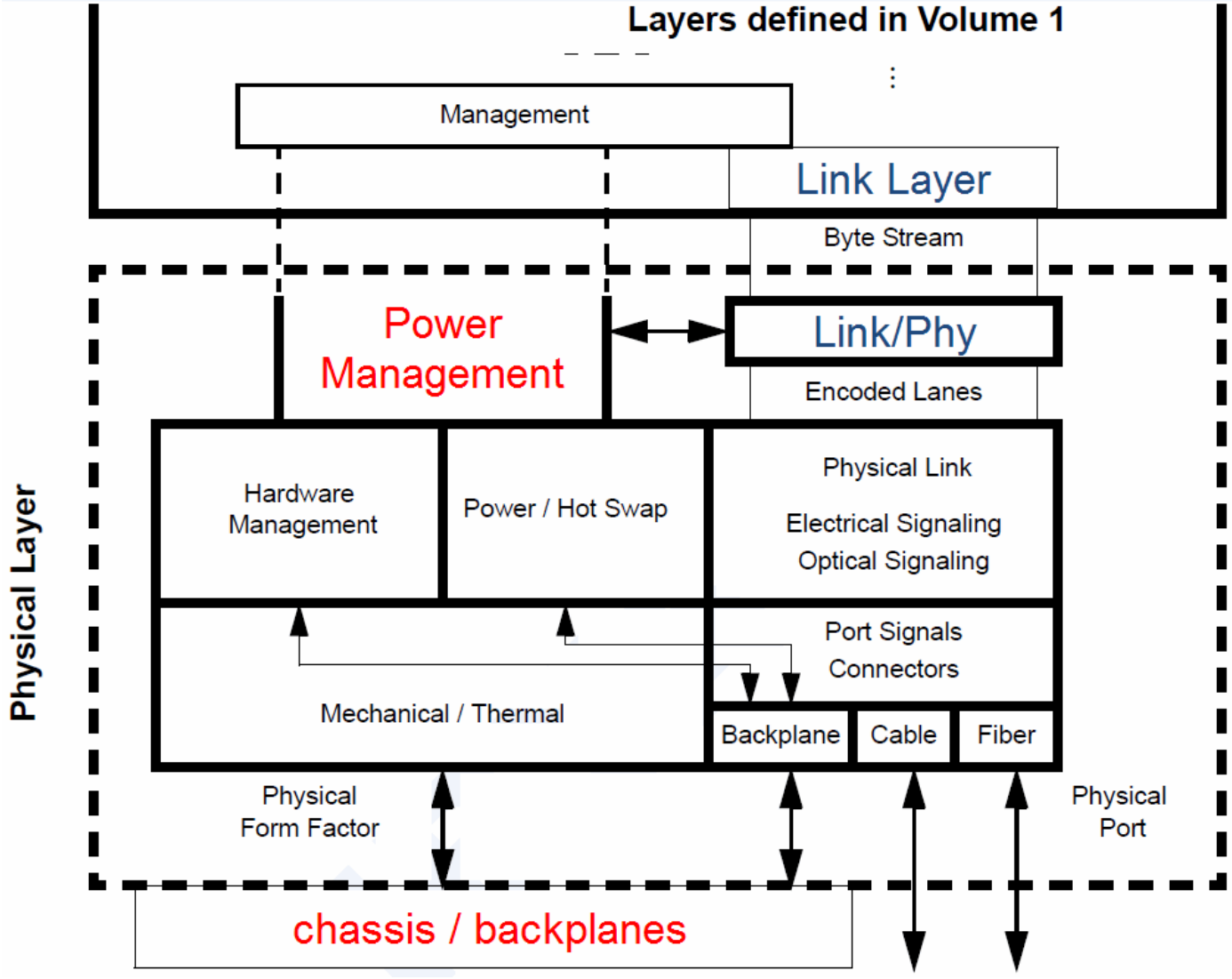
IB roadmap

Lanes	DDR	QDR	FDR	EDR
X1	5 Gbps	10 Gbps	14 Gbps	25 Gbps
X4	20 Gbps	40 Gbps	56 Gbps	100 Gbps



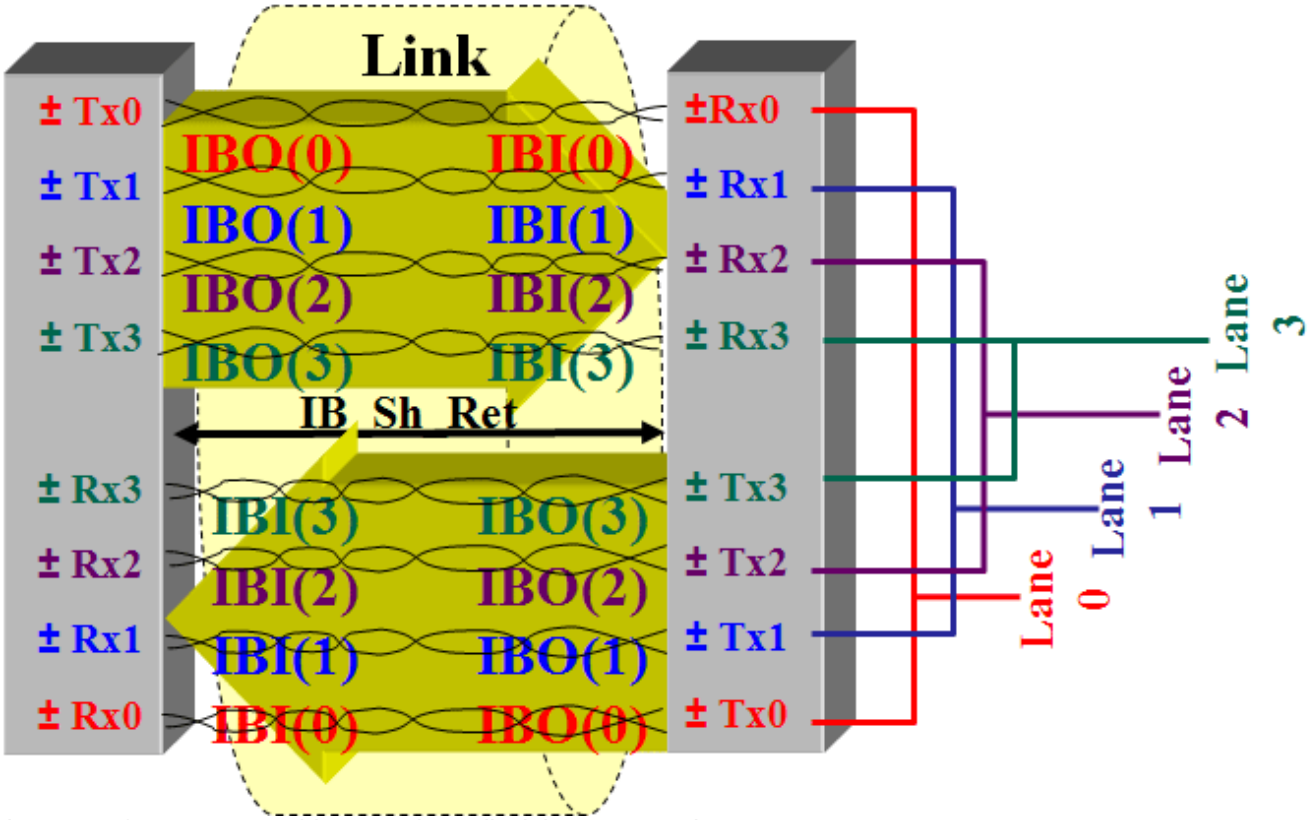
Physical layer

Physical layer structure



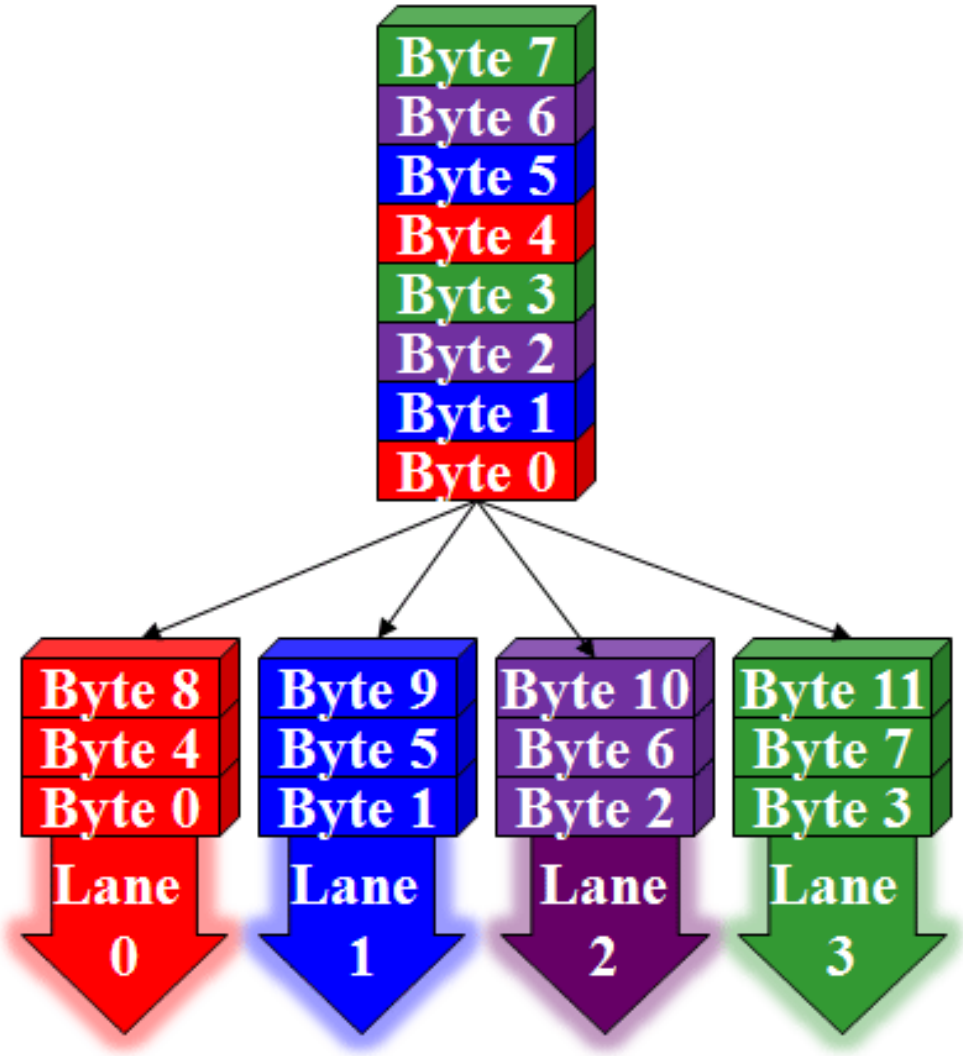
Physical Links

- Links can be implemented in both copper and optical cables
- Copper links consist of differential signals which means it takes four copper wires to make a single link,
  - 2 wires for each signal (i.e. ±Tx0 & ±Rx0).



Byte stripping

InfiniBand, PCIe, and Fibre Channel's XAUI interfaces perform byte stripping.



Physical Layer Attributes

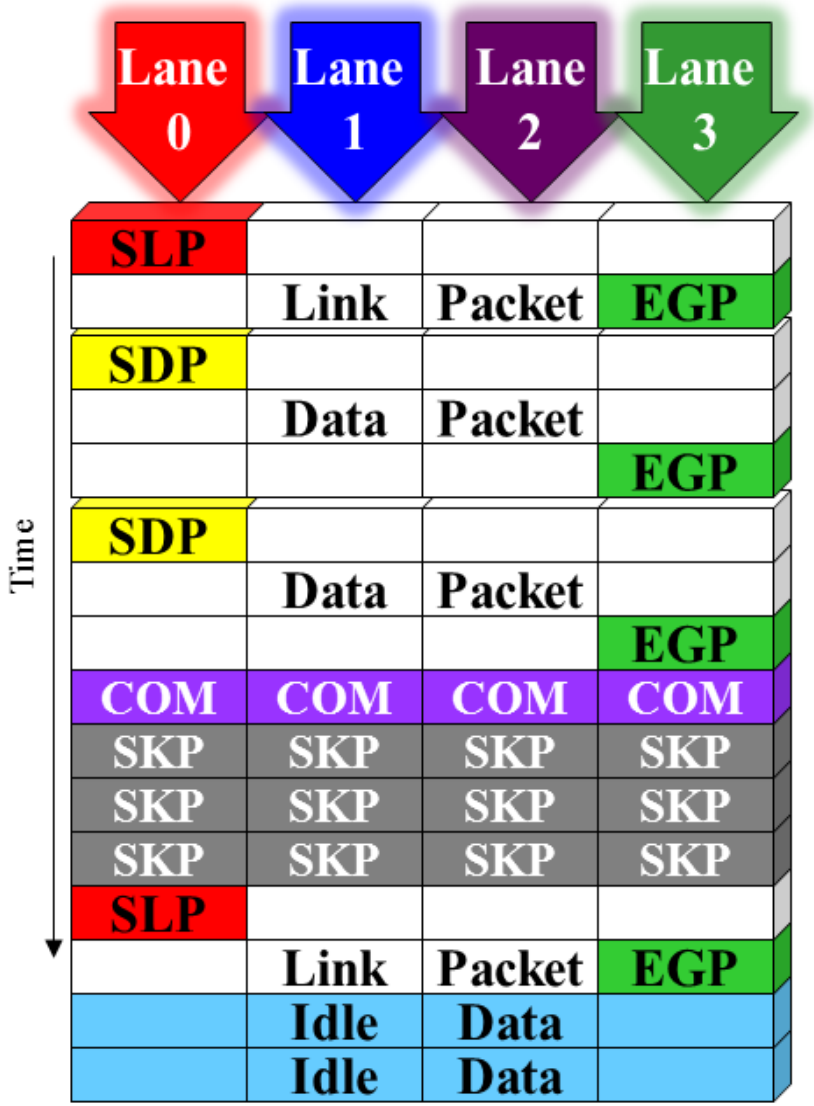
- The physical layer specifies
  - how bits are placed on the wire to form symbols
  - defines the symbols
    - used for framing (i.e., start of packet & end of packet),
    - data symbols,
    - and fill between packets (Idles).
- Specifies the signaling protocol that constitutes a validly formed packet for instance,
  - symbol encoding,
  - proper alignment of framing symbols,
  - no invalid or non-data symbols between start and end delimiters,
  - no disparity errors,
  - synchronization method, etc.

Link Control Signals

Symbol	Encoding	Description
COM	**K28.5**	**Comma, character boundary alignment symbol.**
SDP	**K27.7**	**Start of Data Packet Delimiter**
SLP	**K28.2**	**Start of Link Packet Delimiter**
EGP	**K29.7**	**End of Good Packet Delimiter**
EBP	**K30.7**	**End of Bad Packet Delimiter**
PAD	**K23.7**	**Packet padding symbol**
SKP	**K28.0**	**Skip symbol**

Lane ordering

- The start of packet delimiters (SDP & SLP) shall be transmitted in lane zero only.
- The end of packet delimiters (EGP & EBP) shall be transmitted in lane three only.
- SKIP ordered-sets shall be transmitted on all (4) lanes simultaneously.
- When the idle data is placed on the link, the idle data pattern shall be inserted on all (4) lanes simultaneously.
- The pseudo-random idle data for each lane should start a different point in the sequence.



Link layer

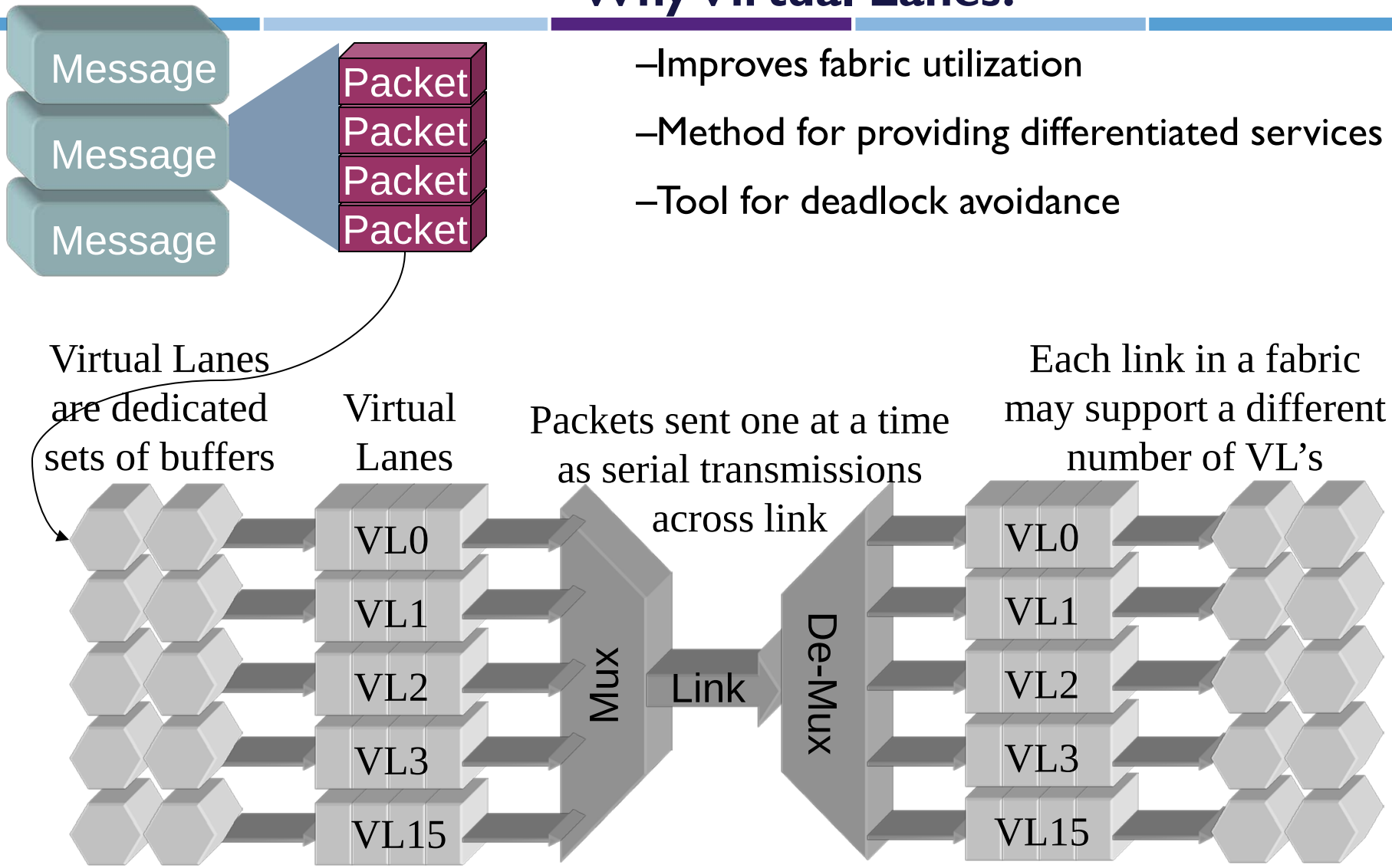
Link Layer Functions

- Responsible for the reception and transmission of packets
  - Transmit packets
    - Control flow
    - Arbitration between virtual lanes
  - Receive packets
    - Error checking
- Link Initialization and Control
- Provide virtual lanes (VLs) to support multiple logical flows over a single link
- Link/Phy Interface
  - Physical layer contains all medium dependent functions:
    - initialization of physical layer signaling (e.g. training sequences)
    - speed and link width negotiation
    - data and control signal coding
  - Link and higher layers are independent of physical layer details
    - simplifies specification of future physical layer technologies

Virtual Lane

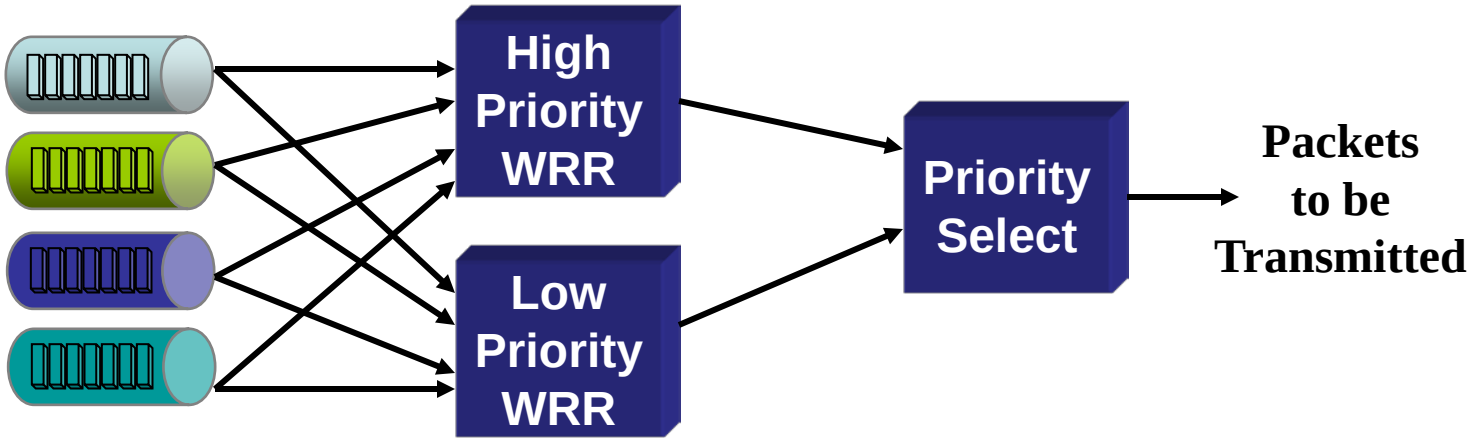
•Why Virtual Lanes?

- –Improves fabric utilization
- –Method for providing differentiated services
- –Tool for deadlock avoidance



VL Arbitration

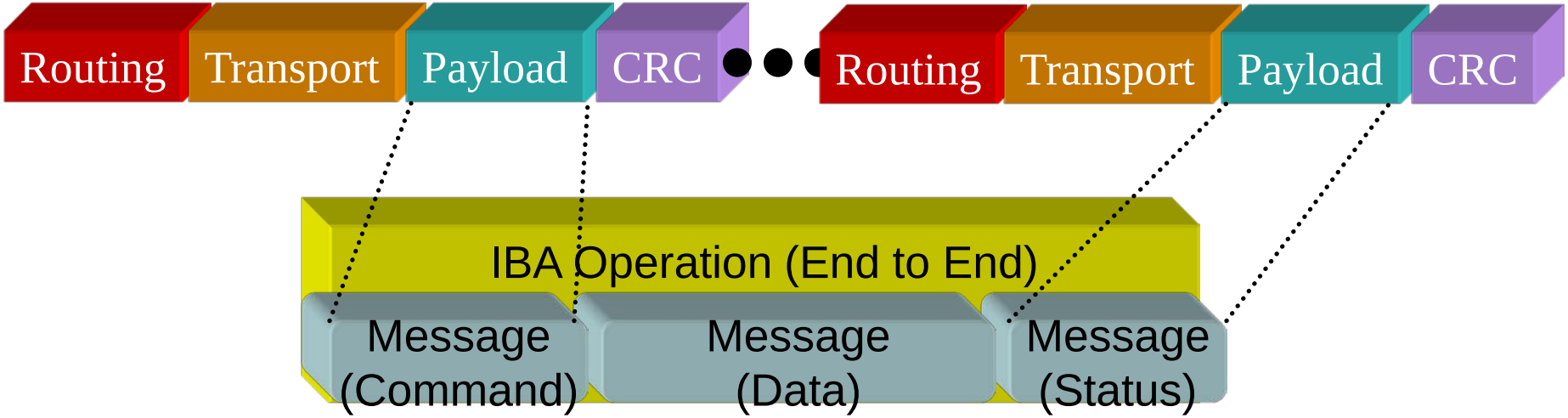
- Each output port selects packets from virtual lanes based on the pre-programmed VL arbitration policy:



- Packets within a VL are transmitted in FIFO order

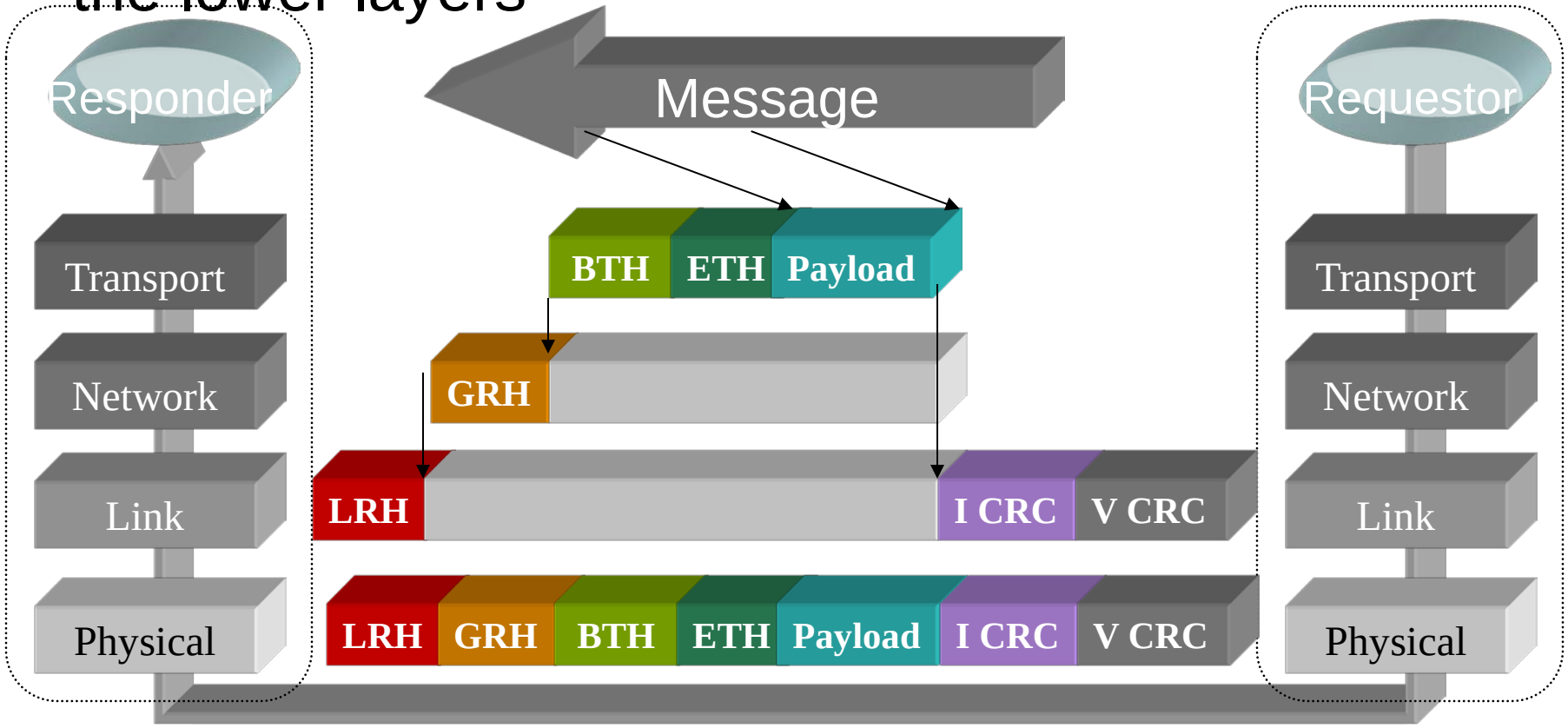
Data Packets

- Data Packets
  - are packets that convey IBA operations
  - consist of a number of different headers, which might or might not be present



Packet header formation

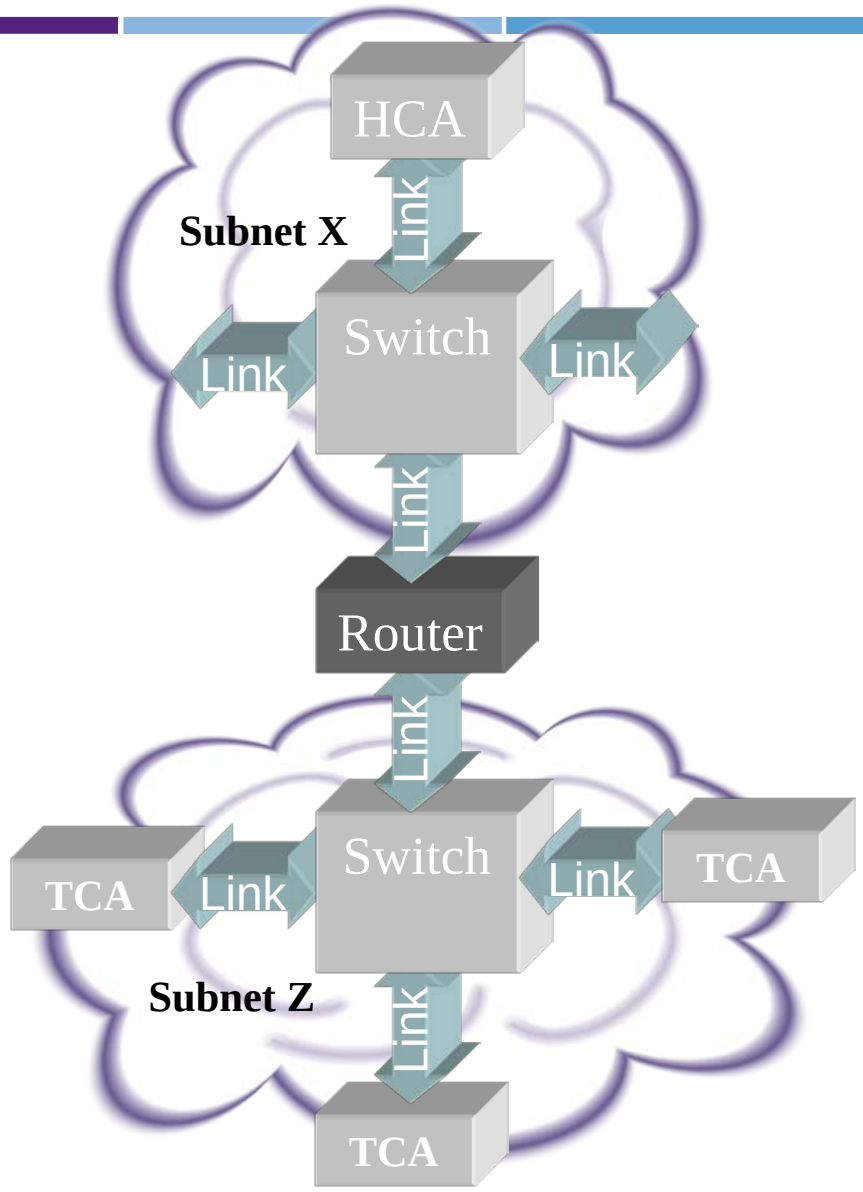
- Transport packet interface is encapsulated by the lower layers



Network layer

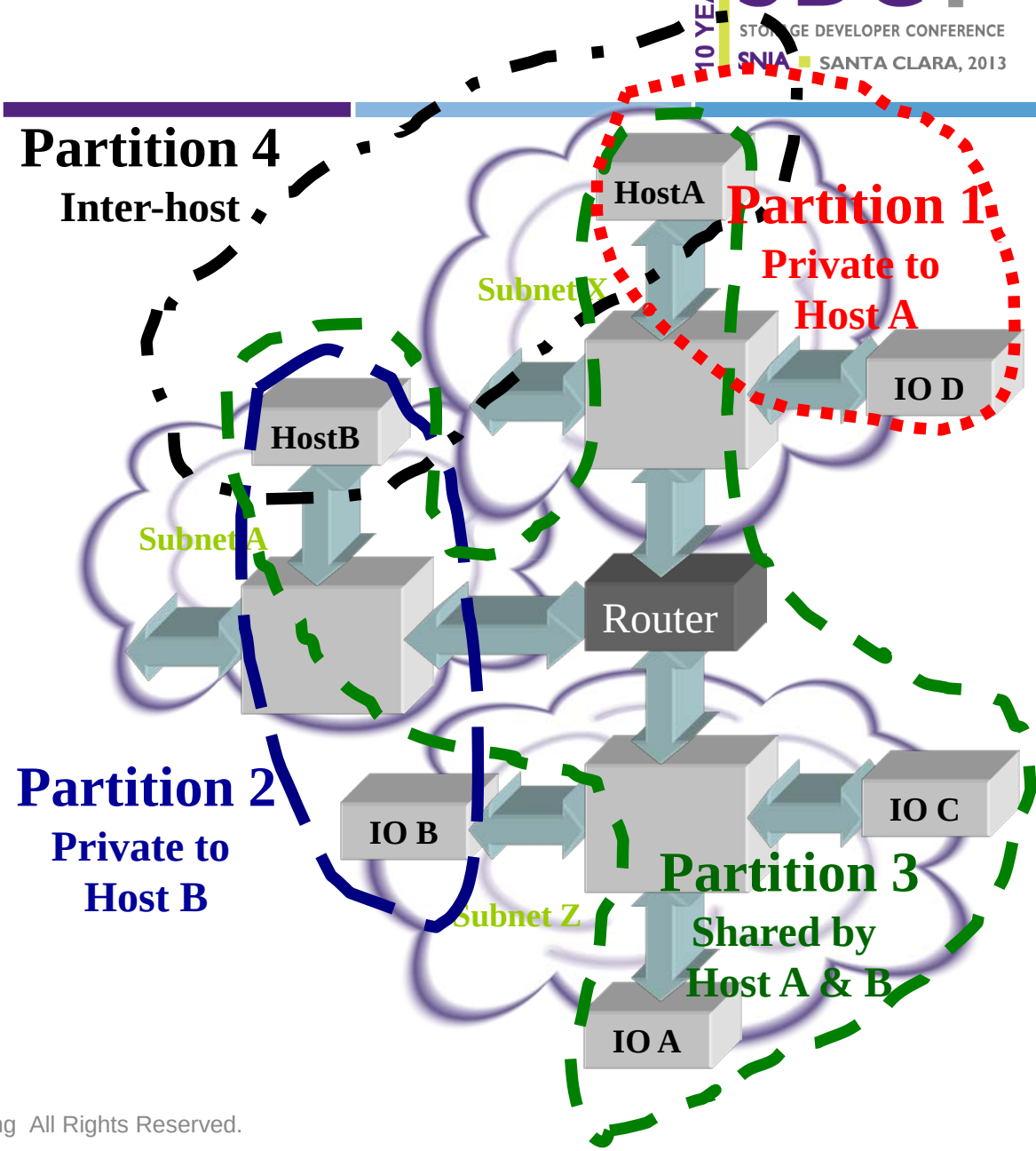
Addressing Hierarchy

- Two level hierarchy
  - Allows for scaling, administrative scope, and deployment flexibility
- Lower level - Subnet
  - Packets are forwarded throughout the subnet utilizing switches
  - Commonly referred to as “switching”
- Higher level – Routing
  - subnets are interconnected using routers
  - the process of forwarding a packet from one subnet to another is referred to as routing
- Nothing new, networks have been switching & routing since the stone age



Partition

- Partition endnodes into logical groups based on a given criteria
  - Conceptually similar to VLANs
  - Facilitates I/O sharing
- Partitions can span subnets
  - Partitioning is managed by a Partition Manager (PM) per subnet
- 16-bit P_Key transmitted within the Base Transport Header
  - Limited members can't talk to one another but communication is allowed between any other combination of membership types.



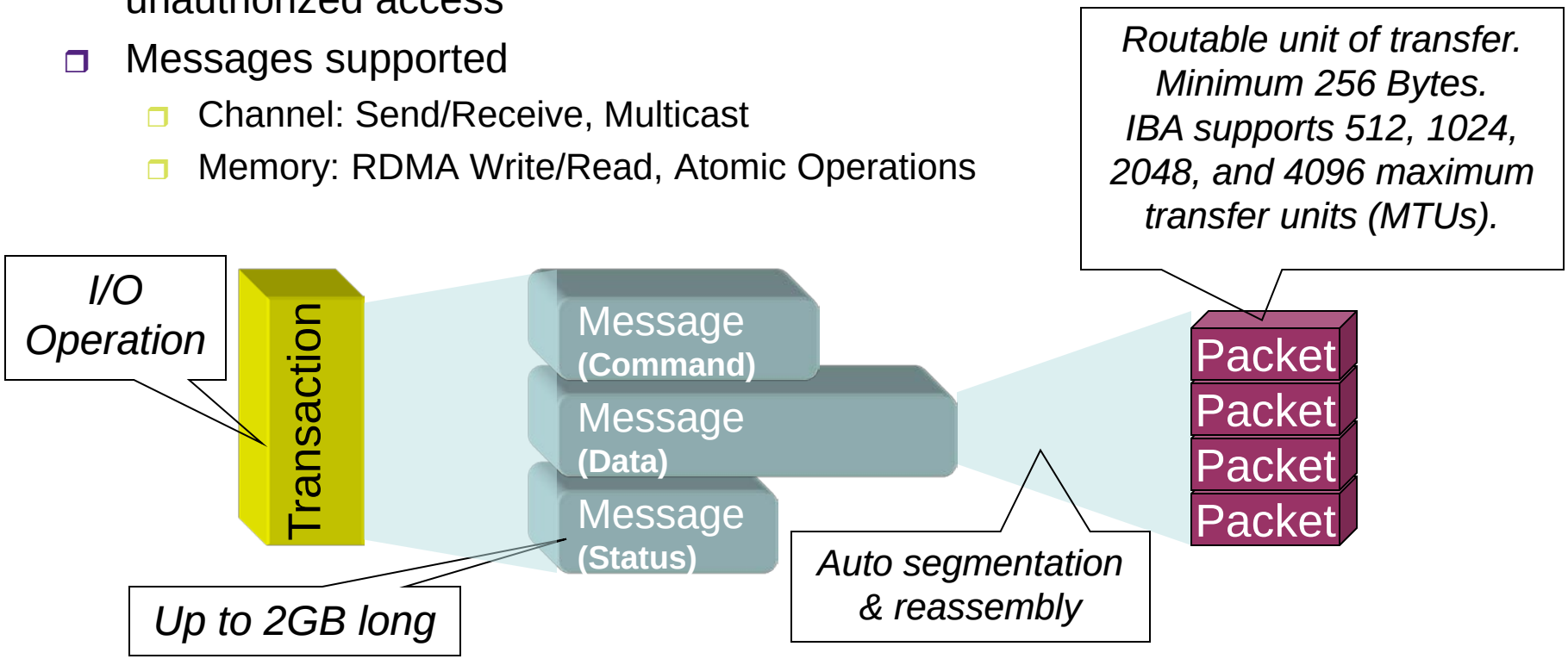
Transport Layer

Transport Layer

- 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
  - 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.

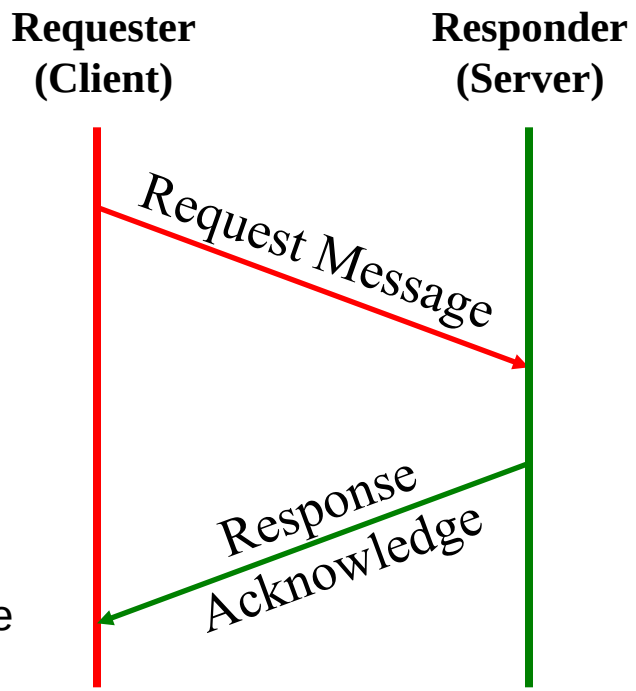
Messages

- Applications, adapters, drivers, devices, etc, execute transactions in logical units called messages
- Messages are moved via the transport layer
- H/W based memory/resource protection is required to prevent unauthorized access
- Messages supported
  - Channel: Send/Receive, Multicast
  - Memory: RDMA Write/Read, Atomic Operations



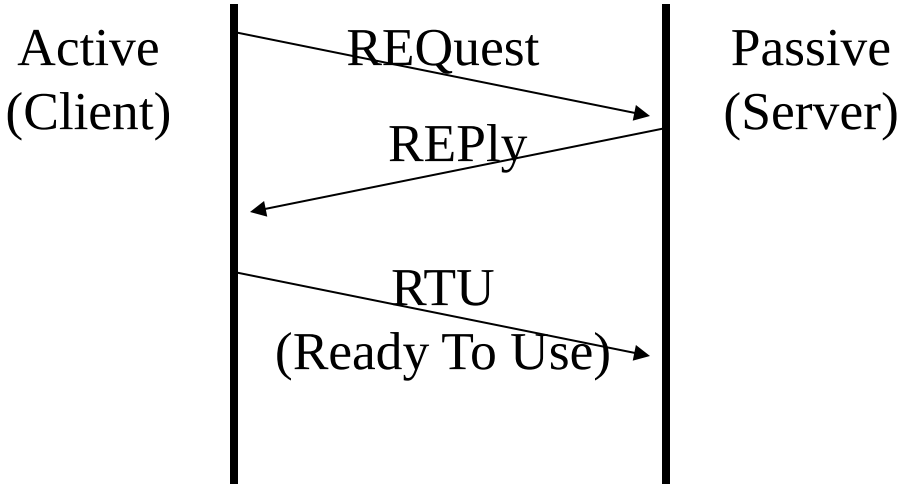
IB Operation

- An IBA operation is defined to include a request message and, for reliable services, its corresponding response.
  - Thus, the request message is generated by a requester, and a response, if one exists, is generated by the responder.
- A request message consists of one or more IBA packets.
  - The packets of a request message are called request packets.
- A response consists of exactly one packet.
  - A response is also called an acknowledge.
  - The response packet acknowledges receipt of one or more request packets.
  - The response may acknowledge the receipt of packets that comprise anywhere from a portion of a request message to multiple request messages.



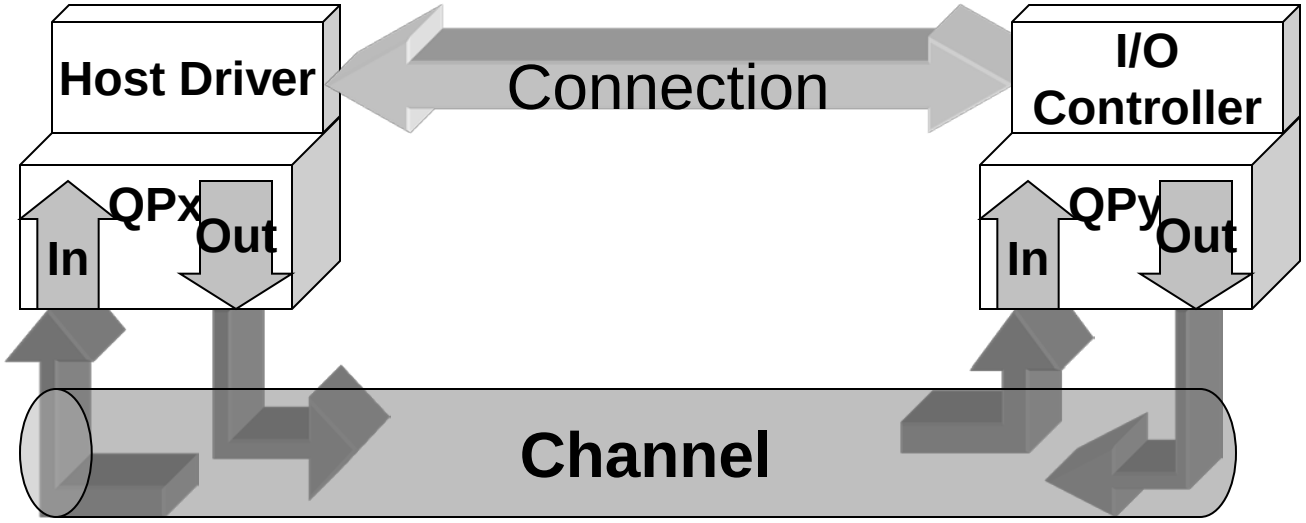
Connection Protocol

- Both sides need to agree, and to know that they've agreed
- Uses a simple three-message exchange



Connections

- A connection is a logical association or work request between requester and responder consumer entities.
- Establishes a channel
  - An association of Queue Pairs
- Transport Services include:
  - Reliable Connection and Unreliable Connection



Transport functions

Transport Functions and Services

Transport Functions	Transport Services				
	RC	RD	UC	UD	RAW
SEND	**Supported**	**Supported**	**Supported**	**Supported**	NA
RDMA READ	**Supported**	**Supported**	Not Supported	Not Supported	NA
RDMA WRITE	**Supported**	**Supported**	**Supported**	Not Supported	NA
ATOMIC	Optional	Optional	Not Supported	Not Supported	NA
RESYNC	Not Supported	Not Supported	**Supported**	Not Supported	Not Supported

There are five types of operations (transport functions) that can be posted to the work queues:

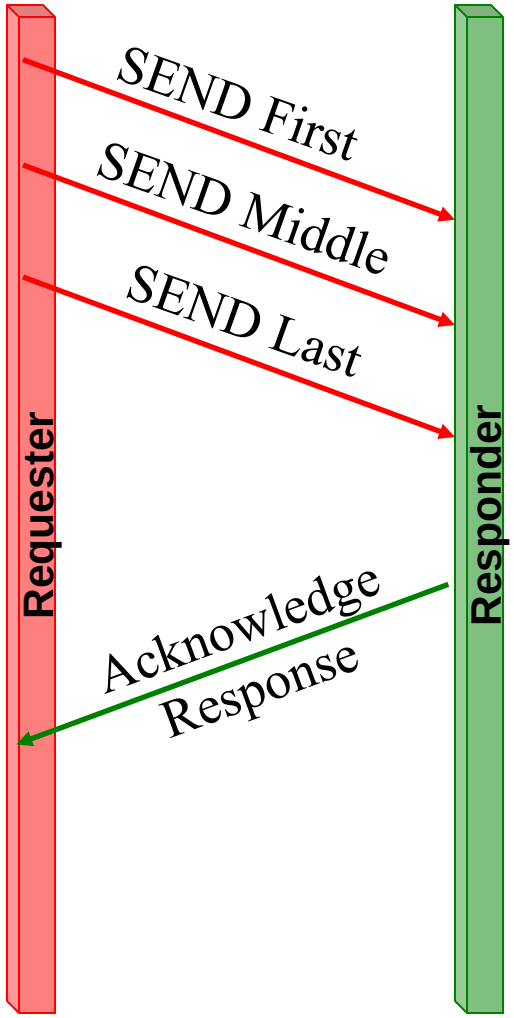
- SEND
- RDMA Write
- RDMA Read
- ATOMIC
- Memory Binding

There are five service levels (transport services) that may be used for each operation:

- **Reliable Connection (RC)**
- **Reliable Datagram (RD)**
- **Unreliable Connection (UC)**
- **Unreliable Datagram (UD)**
- **Raw Datagram (RAW)**

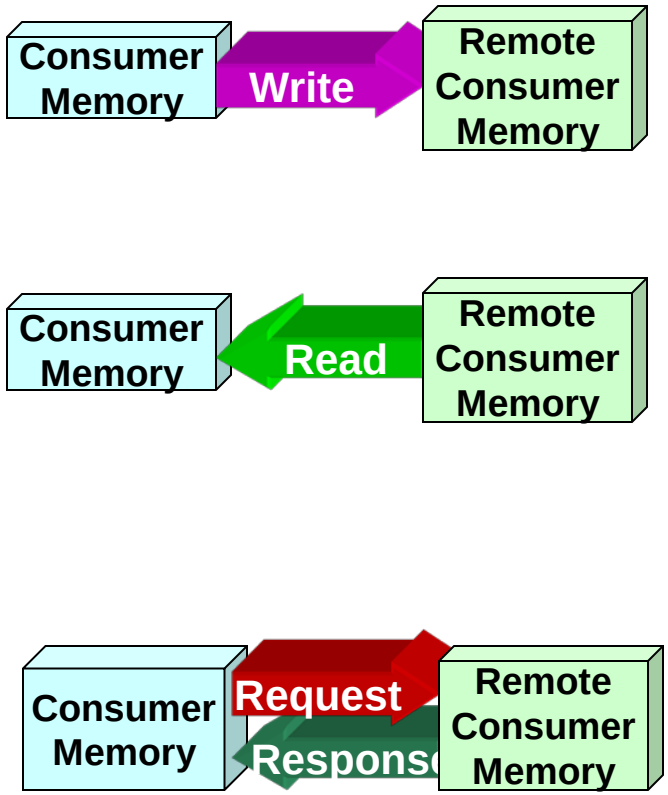
Send Operation Characteristics

- The SEND Operation is sometimes referred to as a *Push* operation or as having *channel* semantics because it moves data much like a mainframe I/O channel.
  - the data is tagged with a discriminator (for IBA the discriminator is the destination LID and QP number)
  - the destination chooses where to place the data based on the discriminator
- A SEND Operation moves a single message.
  - For the RC, RD, and UC transport services this message may be longer than a single packet.
  - A message may range in size from zero bytes to 2^{31} bytes (2 GB).
- With a SEND operation the initiator of the data transfer pushes data to the remote QP.
  - The initiator doesn't know where the data is going on the remote node.
  - The remote node's Channel Adapter places the data into the next available receive buffer for that QP.



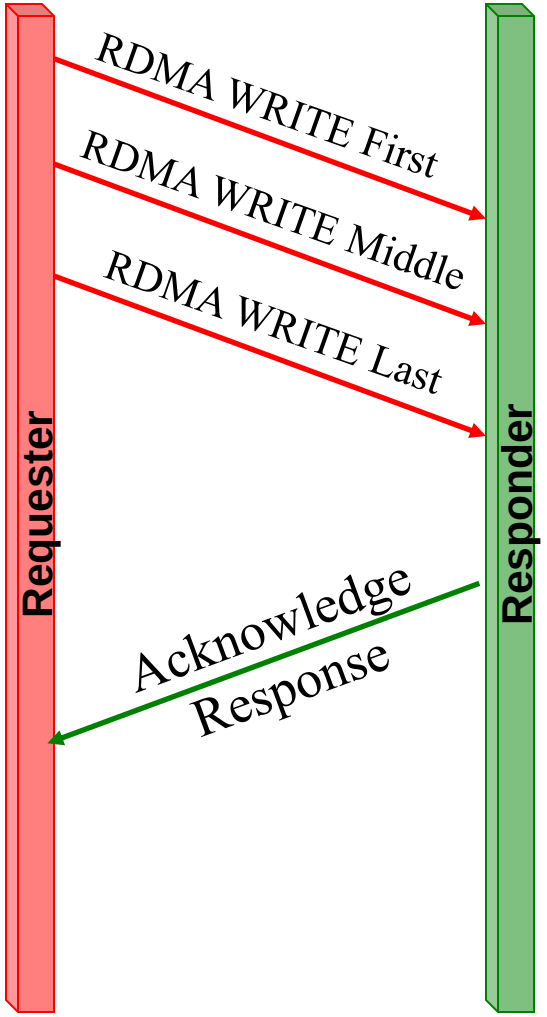
RDMA and Atomic Operations

- Remote Direct Memory Addressing and Atomic operations allow virtual address access to a remote consumers memory
- RDMA-WRITE
  - Stipulates that the hardware is to transfer data from the consumer's memory to the ***remote*** consumer's memory
- RDMA-READ
  - Stipulates that the hardware is to transfer data from the ***remote*** consumer's memory to the consumer's memory
- Atomic
  - Stipulates that the hardware is to perform a read of a remote 64-bit memory location
  - Target returns the value read
  - Conditionally modifies/replaces the remote memory contents by writing an updated value to the same location



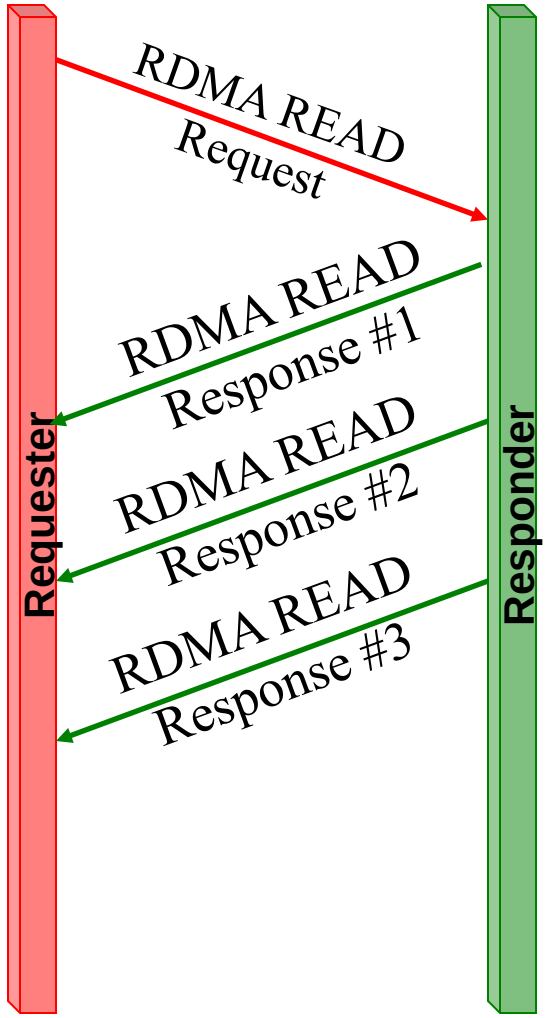
RDMA WRITE Operations

- The RDMA WRITE Operation is used by the requesting node to write into the virtual address space of a destination node.
- The message may be between zero and 2 ^{31} bytes (inclusive) and is written to a contiguous range of the destination QP's virtual address space (not necessarily a contiguous range of physical memory).
  - For an HCA requester performing RDMA WRITE operations, the length of an RDMA WRITE message, as reflected in the RETH:DMALen field, shall be between zero and 2 **31** bytes (inclusive).



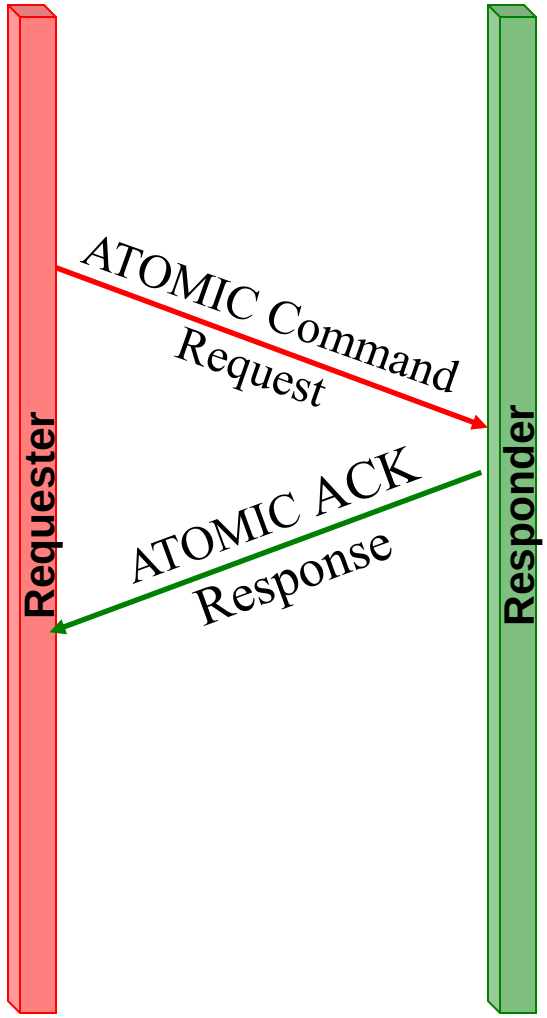
RDMA READ Operation

- RDMA READ Operations are similar to RDMA WRITE Operations.
  - They allow the requesting node to read a virtually contiguous block of memory on a remote node.
  - The responding node first allows the requesting node permission to access its memory.
    - The responder passes to the requestor a virtual address, length, and R_Key to use in the RDMA READ request packet.
- A single RDMA READ request can read from zero to 2^{31} bytes (inclusive) of data.
- When responding to an RDMA READ request,
  - if the requested data size is greater than the PMTU,
    - the responder shall segment the data into PMTU size data segments for transmission as multiple RDMA READ Response packets.
    - The data is reassembled in the requesting node's memory.



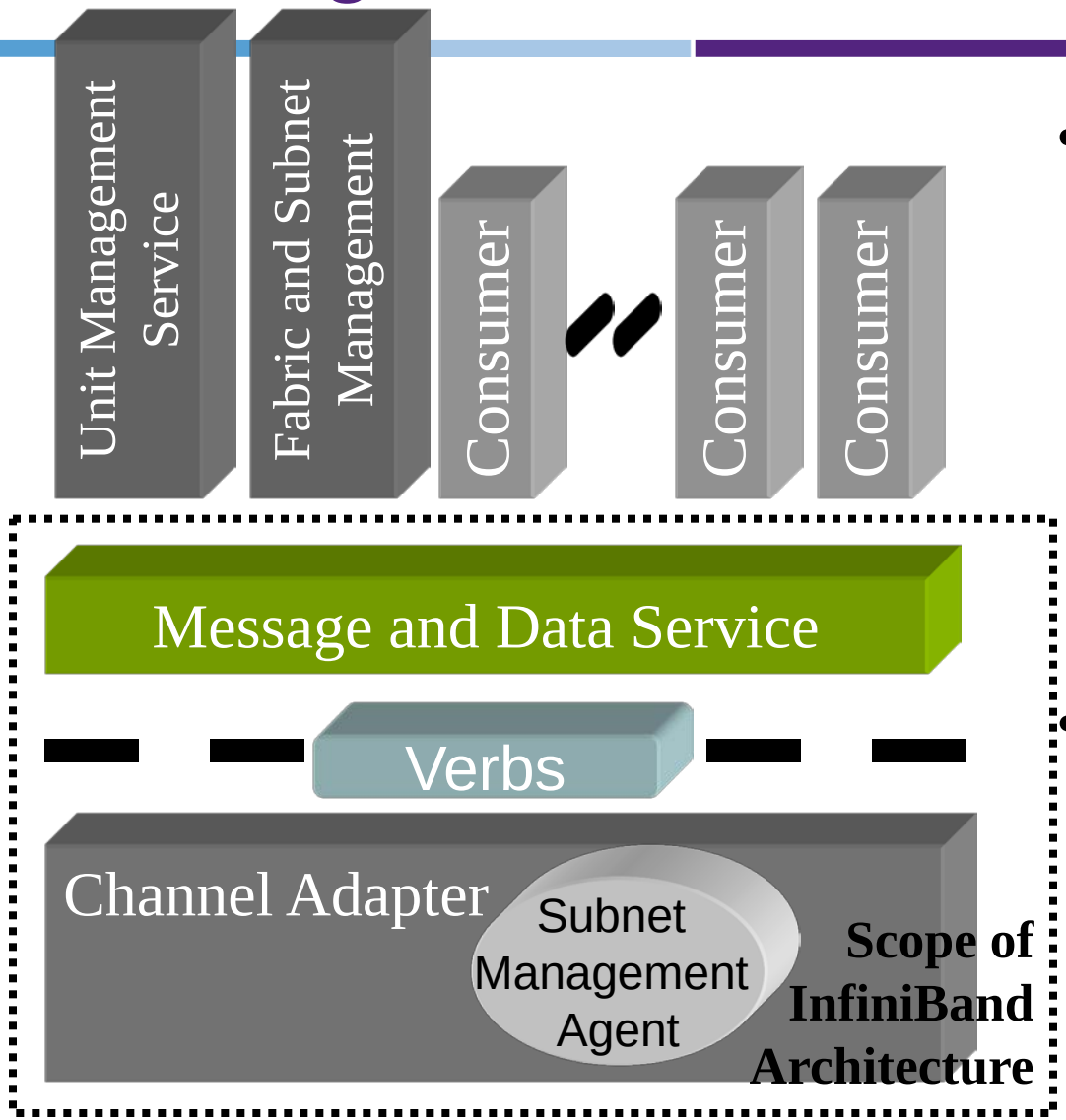
ATOMIC Operations

- ATOMIC Operations execute a 64-bit operation at a specified address on a remote node.
  - The operations
    - atomically read, modify and write the destination address
    - and guarantee that operations on this address by other QPs on the same CA do not occur between the read and the write.
- The scope of the atomicity guarantee may optionally extend to other CPUs and HCAs.
- ATOMIC Operations use the same remote memory addressing mechanism as RDMA READs and WRITEs.
  - The virtual address specified in the request packet is in the address space of the remote QP that the ATOMIC Operation has targeted.



Upper Level Protocols

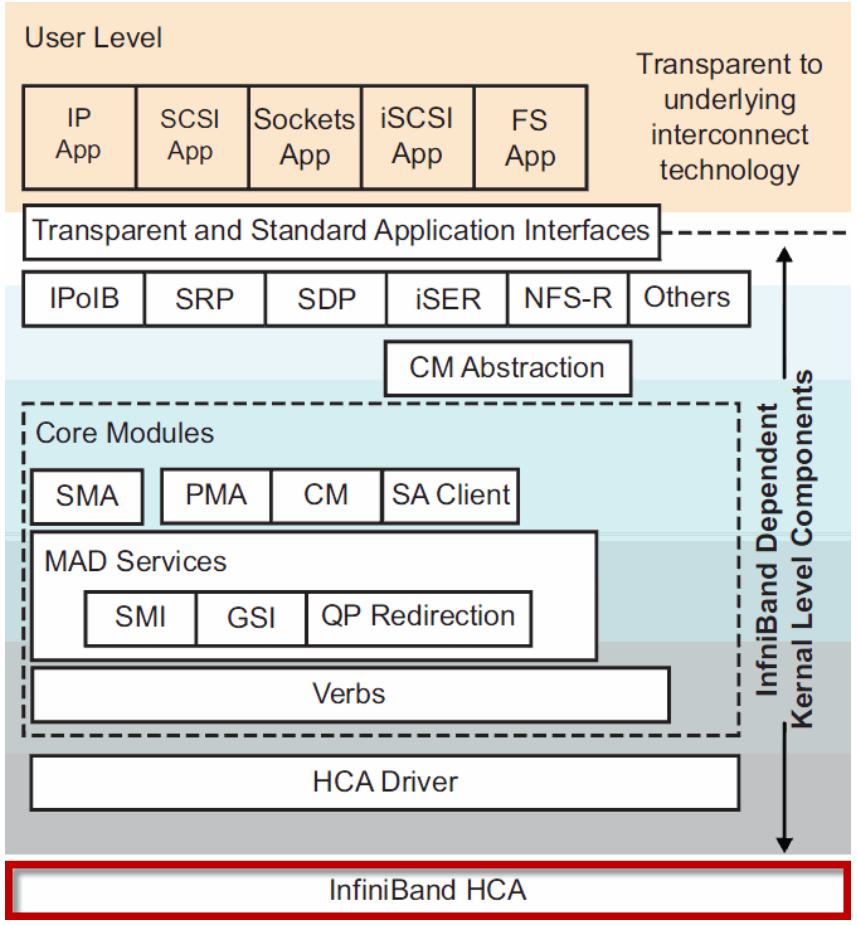
ULP Diagram



- IBA is architected as a first order network
  - Defines the host behavior (IBA verbs)
  - Defines memory operation so that the channel adapter can be located as close to the memory complex as possible
- IBA provides channel semantics
  - Send and receive
  - Direct memory access
  - Prevents access by non-participating consumers

Linux InfiniBand software architecture

- The software consists of:
  - a set of kernel modules and protocols.
  - associated user-mode shared libraries as well which are not shown in the figure.
- Applications at the user level
  - stay transparent to the underlying interconnect technology
- Application developers
  - need to know to enable their IP, SCSI, iSCSI, sockets or
  - file system based applications
  - to operate over InfiniBand



Software architecture details

The kernel code divides logically into three layers:

1. the HCA driver(s),
2. the core InfiniBand modules,
3. and the upper level protocols.

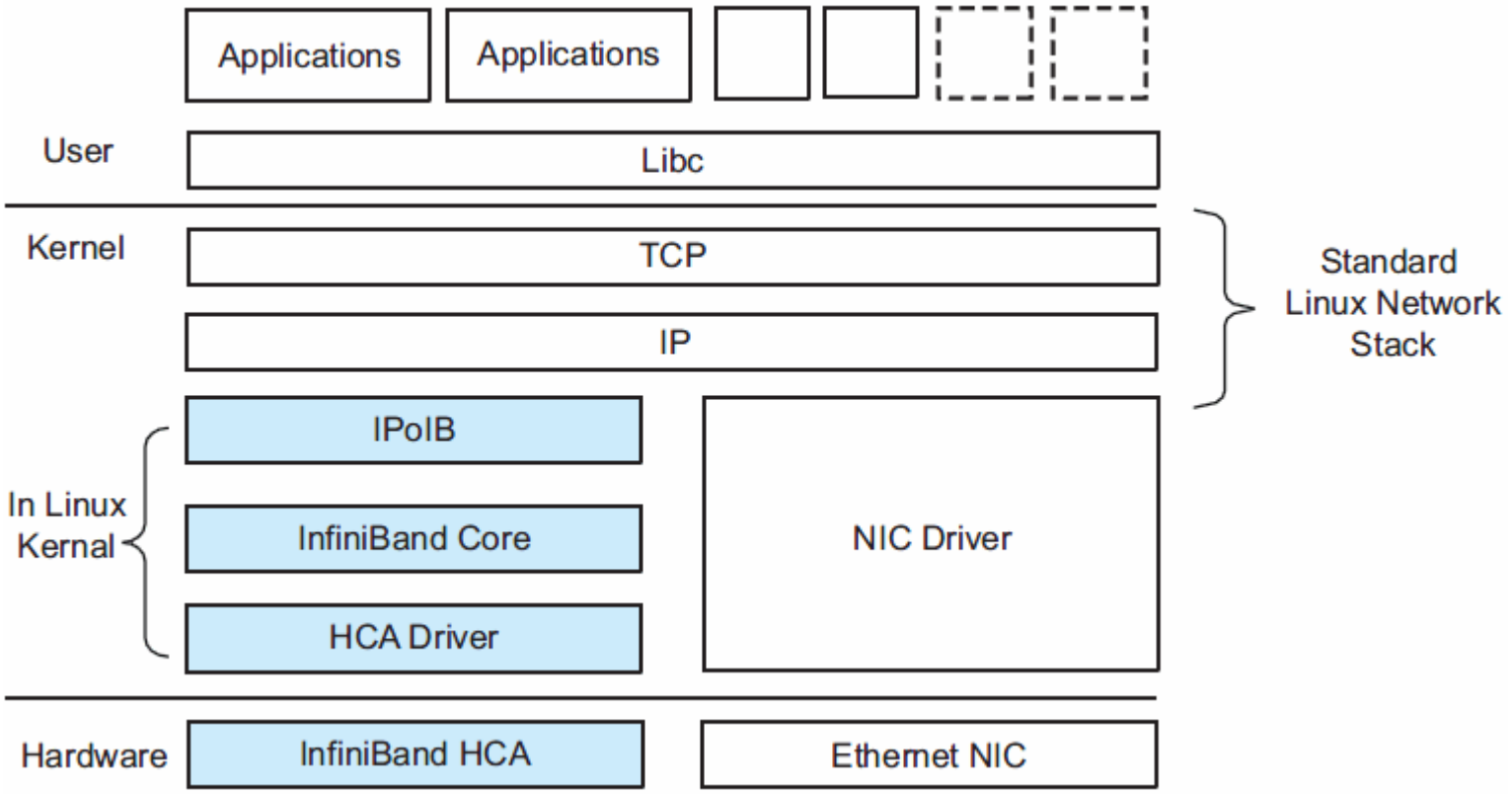
- Core InfiniBand modules comprise the kernel level mid-layer for InfiniBand devices.
- The mid-layer
  - allows access to multiple HCA NICs and
  - provides a common set of shared services.
- These include the following services:
  - User-level Access Modules
  - The user-level access modules implement the necessary mechanisms to allow access to InfiniBand hardware from user-mode applications.

Mid-layer functions

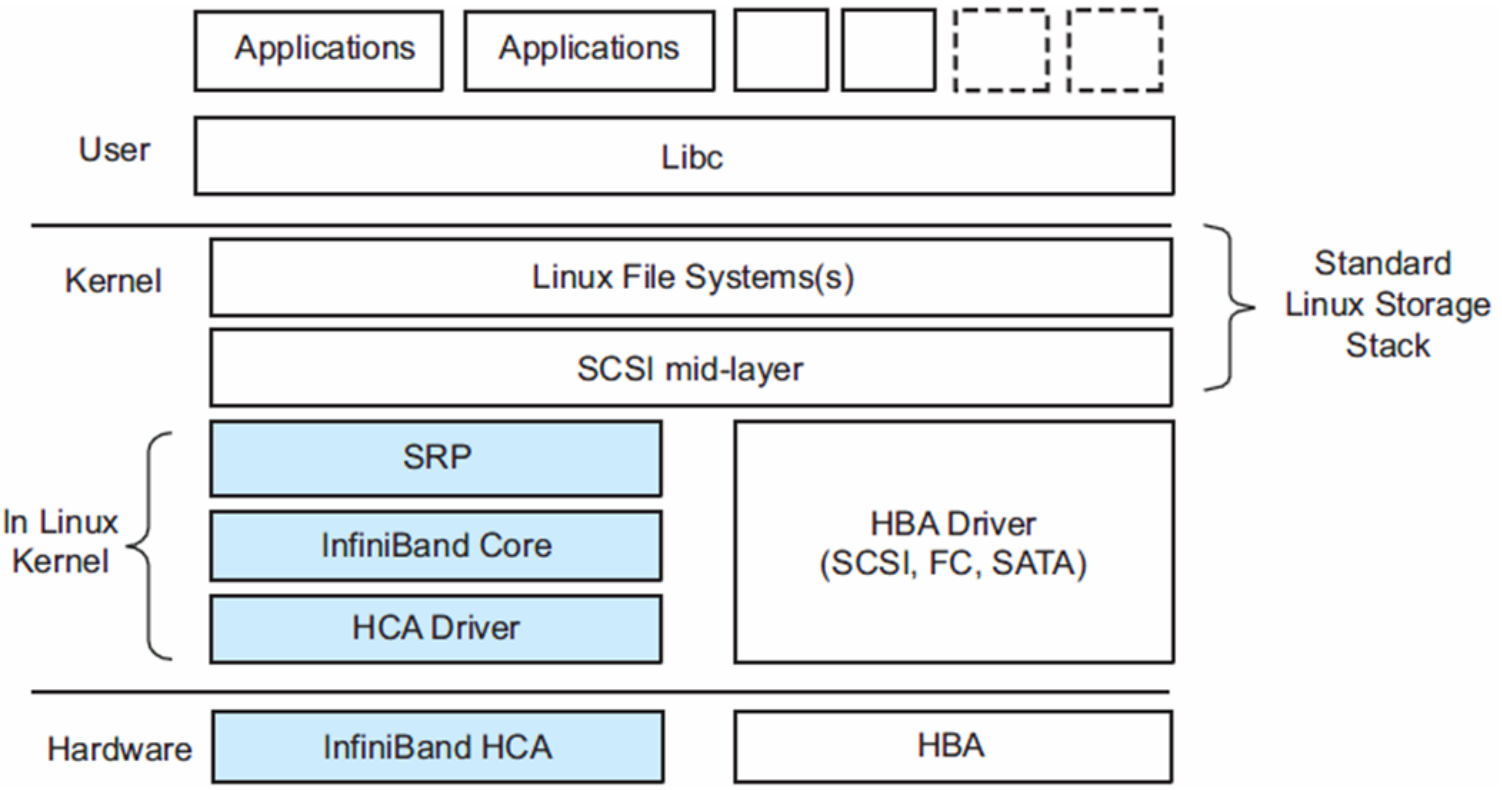
- Communications Manager (CM)
  - CM provides the services needed to allow clients to establish connections
- Subnet Administrator (SA) Client
  - SA client provides functions that allow clients to communicate with the subnet administrator
  - The SA contains important information, such as path records, that are needed to establish connections
- Subnet Management Agent (SMA)
  - SMA responds to subnet management packets that allow the subnet manager to query and configure the devices on each host
- Performance Management Agent (PMA)
  - PMA responds to management packets that allow retrieval of the hardware performance counters

- **General Services Interface (GSI)**
  - GSI allows clients to send and receive management packets on special QP 1.
- **MAD services**
  - Management Datagram (MAD) services
  - a set of interfaces that allow clients to access the special InfiniBand queue pairs (QP), 0 and 1
- **Queue pair (QP) redirection**
  - allows an upper level management protocol that would normally share access to special QP 1 to redirect that traffic to a dedicated QP.
  - This is done for upper level management protocols that are bandwidth intensive.
- **Subnet Management Interface (SMI)**
  - SMI allows clients to send and receive packets on special QP 0.
  - This is typically used by the subnet manager (SM).

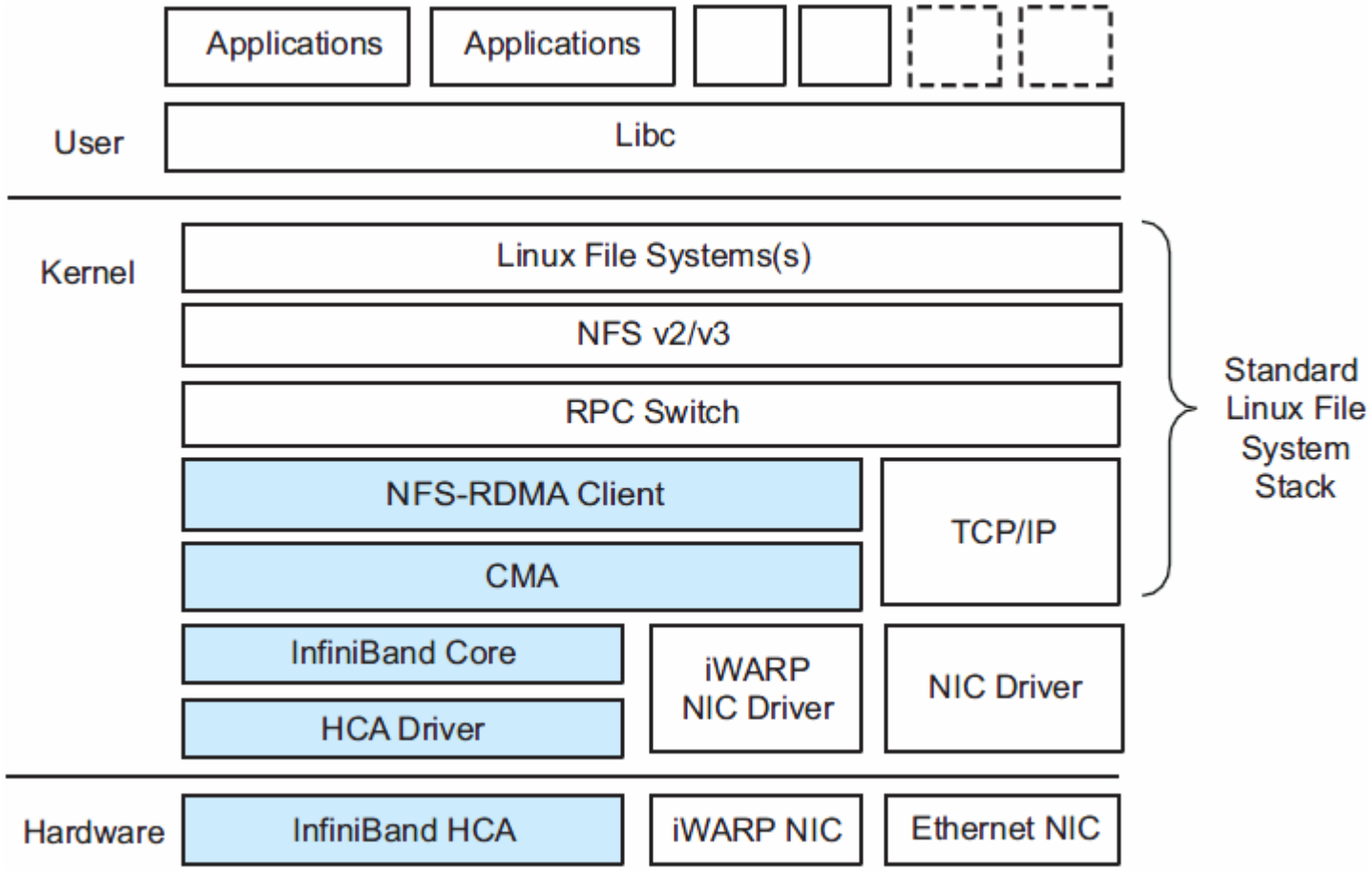
IP application



SCSI RDMA Protocol (SRP)



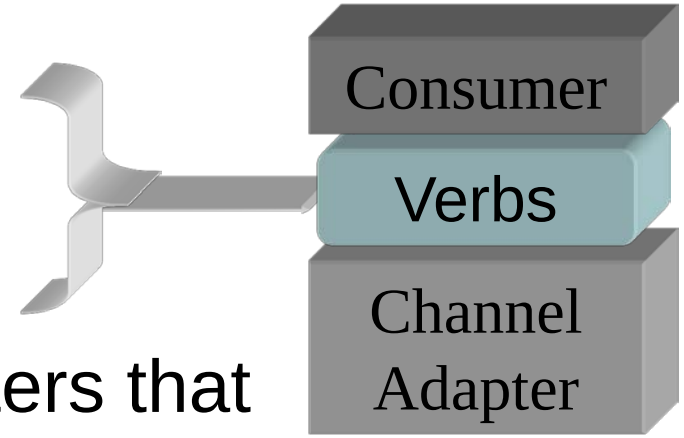
Network File System (NFS) over RDMA



IB verbs

IB Verbs

- Verbs describe the functions necessary to:
  - Configure
  - Manage
  - Operate a channel adapter (CA)
- Identifies appropriate parameters that need to be included for each particular function
- Verbs are not an API (application programming interface) nor a Hardware Abstraction Layer (HAL)
  - provide the framework for the OSV to specify the API



Verb Concepts

- Host Channel Adapter (HCA)
  - Represents local channel interface (CI)
  - Also known as plain old channel adapter (CA) or even a Target Channel Adapter (TCA) if on receiving end
- Queue Pair (QP):
  - Represents communications endpoint, like a socket
  - Consists of a SEND Queue and a RECEIVE Queue
- Work Request Element (WQE): requests communications operation
- Completion Queue (CQ): provides completed operation status
- Memory Regions: system memory is “registered” to allow access to local and remote channel adapters to source/sink communications data

IB defined verbs

HCA verbs

Open HCA
Query HCA
Modify HCA Attributes Access Violation Counters
Close HCA
Allocate Protection Domain
Deallocate Protection Domain
Allocate Reliable Datagram Domain RD Service
Deallocate Reliable Datagram Domain RD Service

HCA resource verbs

Create Address Handle
Modify Address Handle
Query Address Handle
Destroy Address Handle

Queue Pair verbs

Create Queue Pair
Modify Queue Pair
Query Queue Pair
Destroy Queue Pair
Get Special QP
Create Completion Queue
Query Completion Queue
Resize Completion Queue
Destroy Completion Queue
Create EE Context RD Service
Modify EE Context Attributes RD Service
Query EE Context RD Service
Destroy EE Context RD Service

Memory Management verbs

Register Memory Region
Register Physical Memory Region
Query Memory Region
Deregister Memory Region
Reregister Memory Region
Reregister Physical Memory Region
Register Shared Memory Region
Allocate Memory Window
Query Memory Window
Bind Memory Window
Deallocate Memory Window

Multicast verbs

Attach QP to Multicast Group UD Multicast Service
Detach QP from Multicast Group UD Multicast Service

Processing verbs

Post Send Request
Post Receive Request

Completion Queue verbs

Poll for Completion
Request Completion Notification

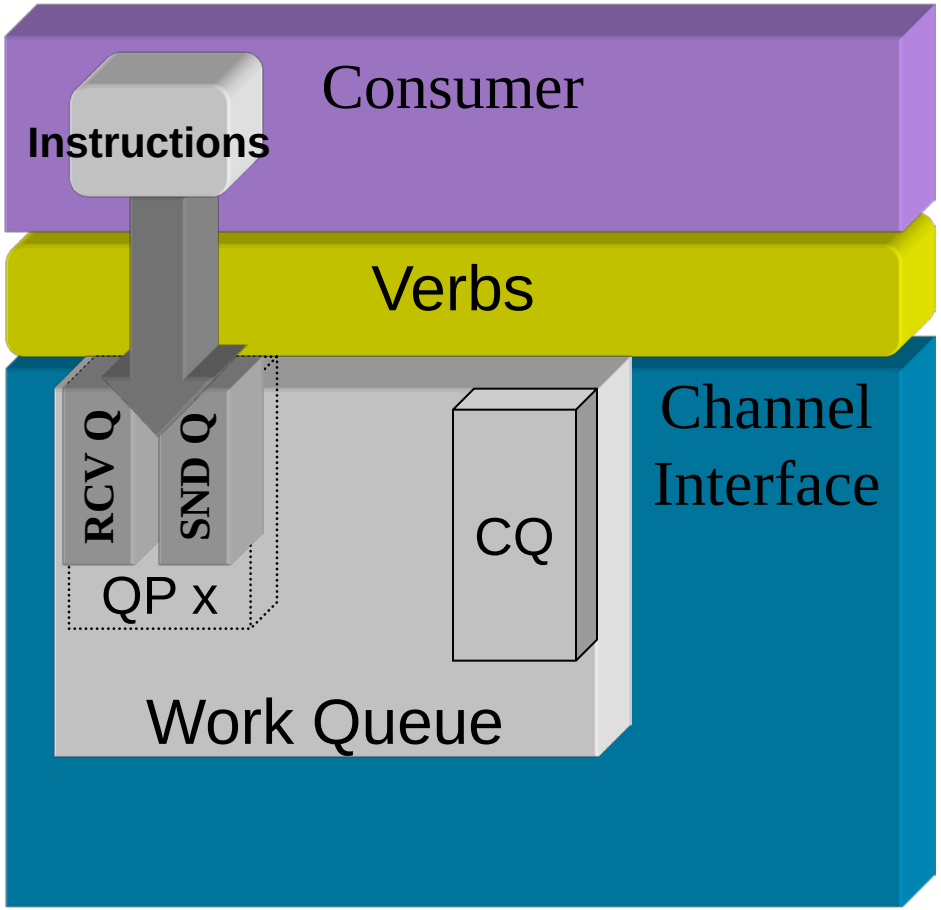
Event handling

Set Completion Event Handler
Set Asynchronous Event Handler

Software transport interface

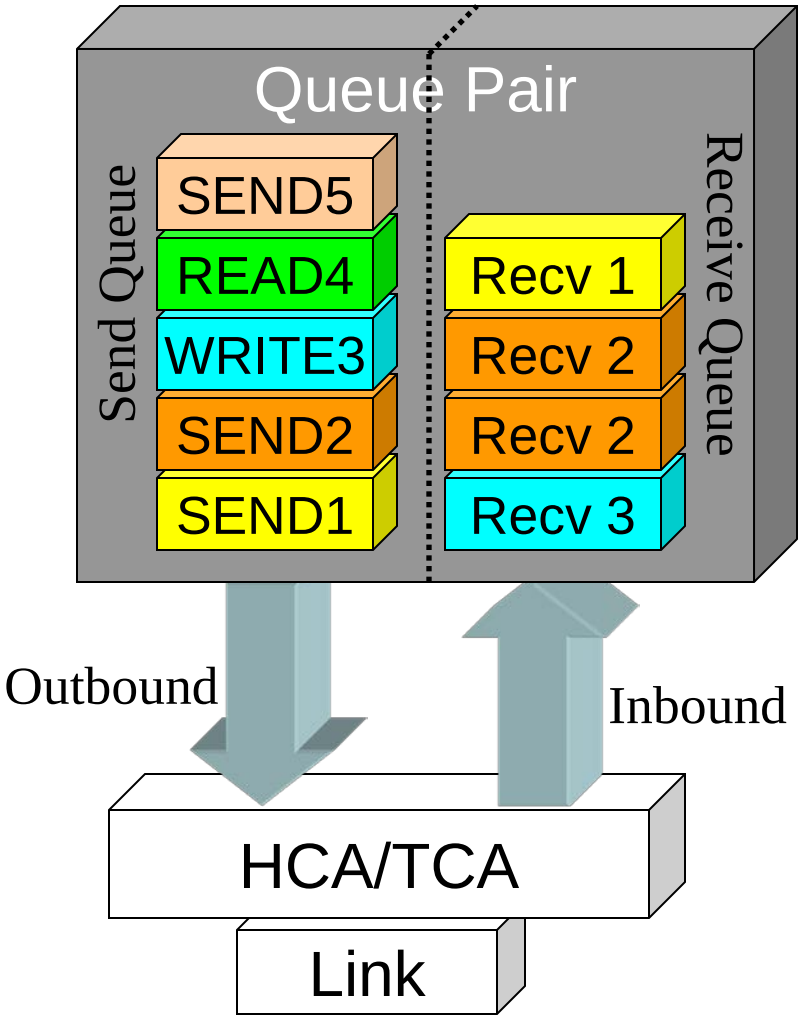
Work Queue

- The foundation of the IBA operation is the ability of a consumer to queue up a set of instructions that the hardware executes
- This facility is referred to as a “Work Queue”.
- A work queue is made up of:
  - A Queue Pair (QP)
    - Send Queue (SND Q)
    - Receive Queue (RCV Q)
  - A Completion Queue (CQ)
    - are required for communication of completion status
    - All QP’s are linked to a CQ



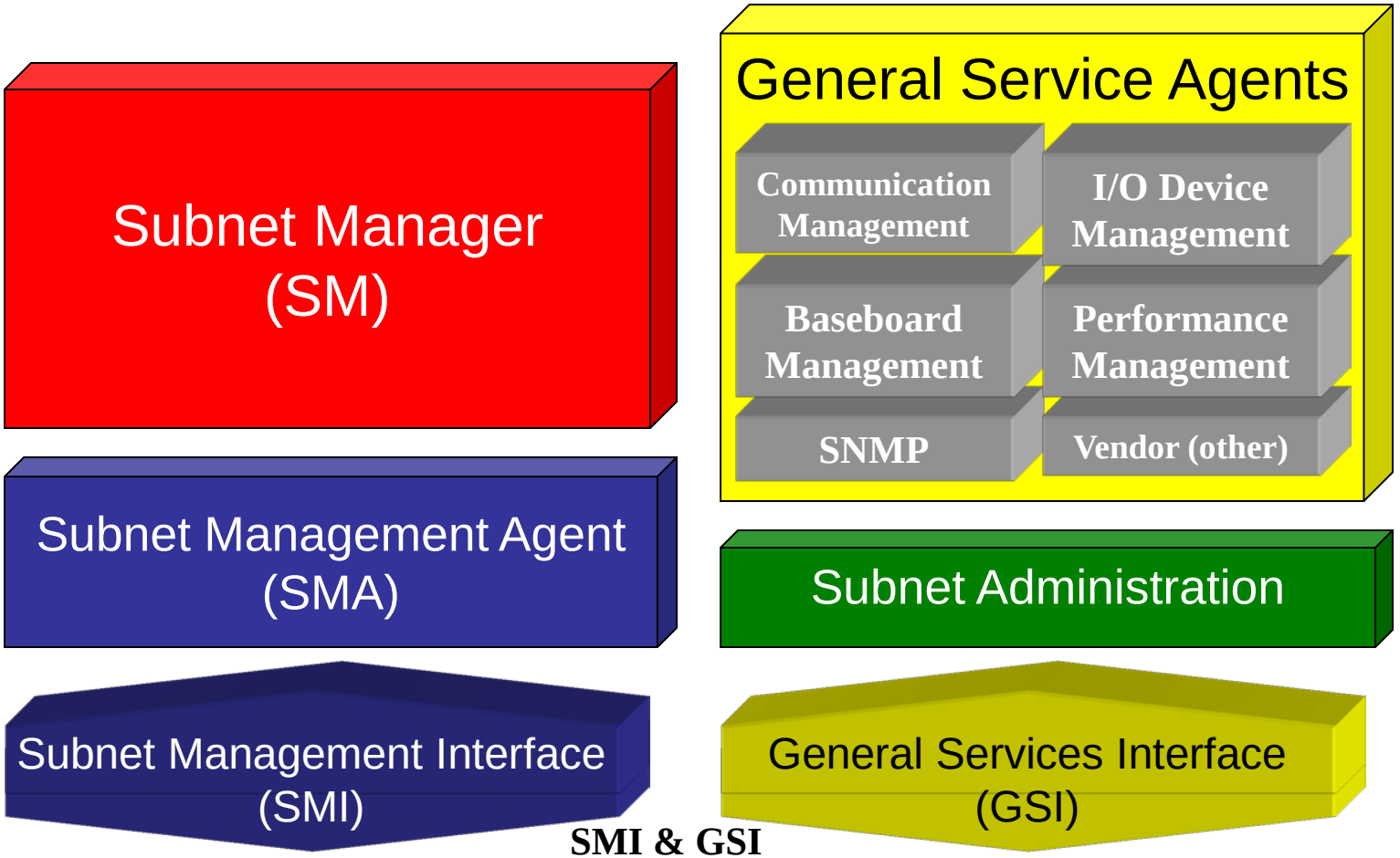
Queue pair

- Work Queues utilize communication entities known as Queue pairs
  - Queue Pairs are the basic mechanism for inter-endpoint communications
- Dual-simplex model
- Work is scheduled by posting the request to the outbound queue
- Poster is notified when requests complete
- QP Verbs:
  - Create, Modify, Query, and Destroy



IB Management

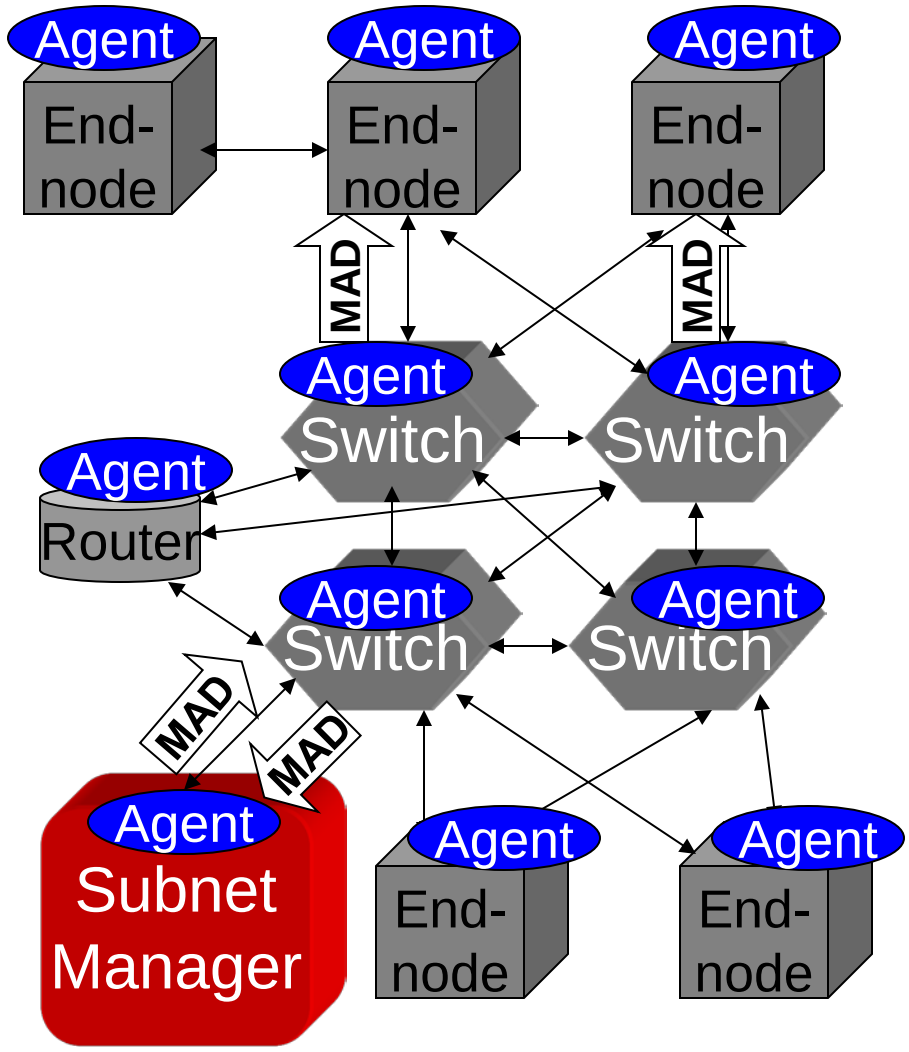
Management Model



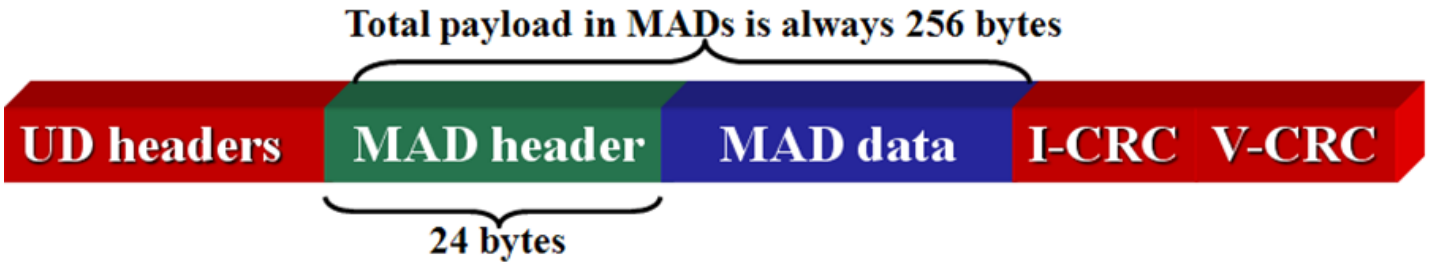
SMI & GSI
Are abstract interfaces not protocol layers

Management Model Concepts

- Node: any managed entity including endnodes, switches, and routers
- Manager: Active entity – sources commands and queries
- Agent: typically passive entity – responds to managers but can source Traps
- MAD: management Datagram – standard format for all communications between managers and agents

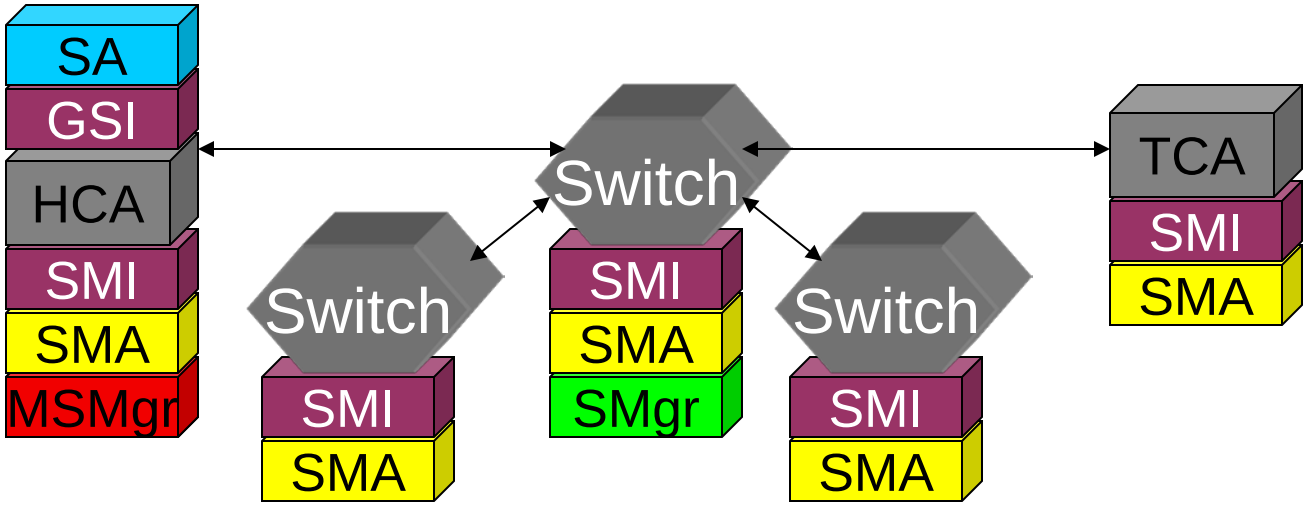


MAD (management datagram)



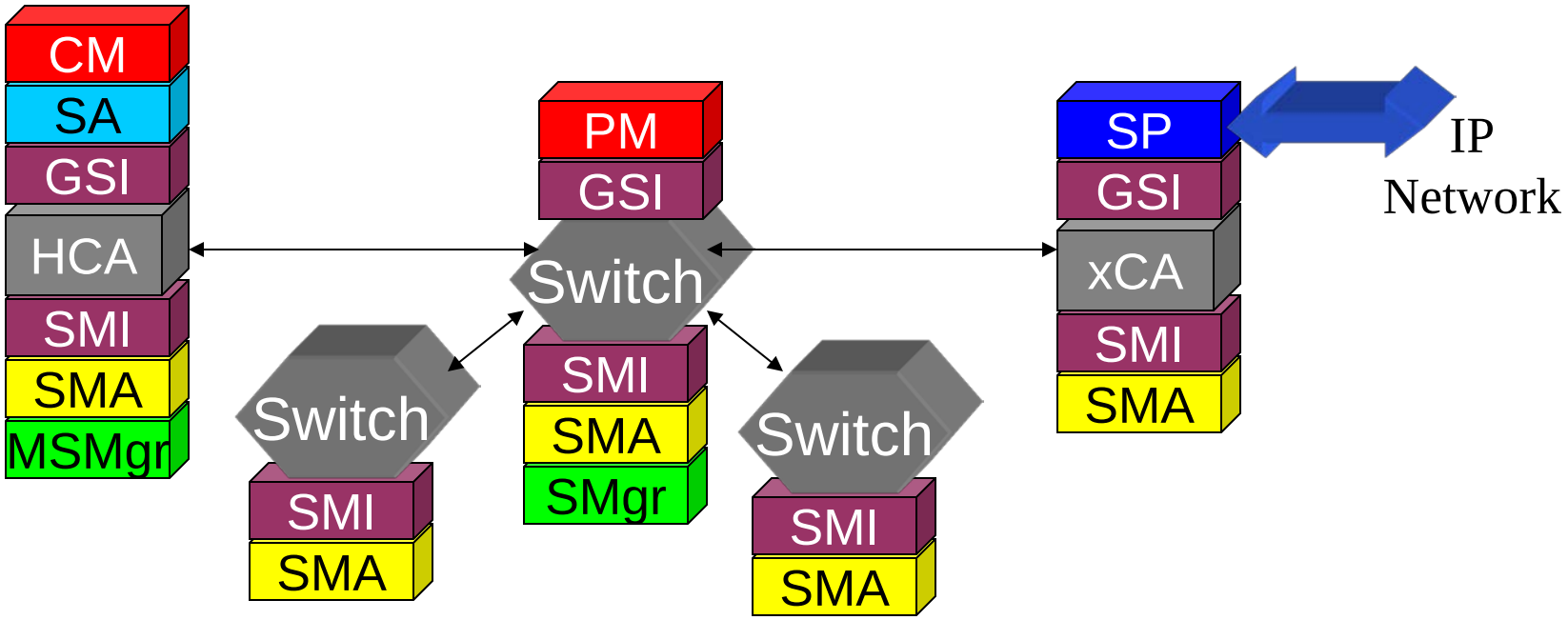
	Byte n+3								Byte n+2								Byte n+1								Byte n							
bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
bytes																																
0	BaseVersion								ManagementClass								ClassVersion								R	Method						
4	Status																ClassSpecific															
8	TransactionID																															
12																																
16	AttributeID																R=reserved															
20	AttributeModifier																															
24 to 255	(MSB) MAD Data (LSB)																															

Subnet Management Agents



- There can be multiple nodes that may be Subnet Management capable
- Exactly one node is selected as Master SM (MSMgr)
  - Controls fabric routing
  - All endnodes confer with SA for connection establishment, topology info, path MTU, etc.
  - Any other SM goes to standby mode
  - Master SM selection routine is outside scope of IBA

Adding General Service Agents



- Additional Agents can become active when needed
  - Communication Management (CM)
  - SNMP Proxy (SP)
  - Performance Management (PM)
  - etc…

InfiniBand Architecture
End