Sonoma Workshop

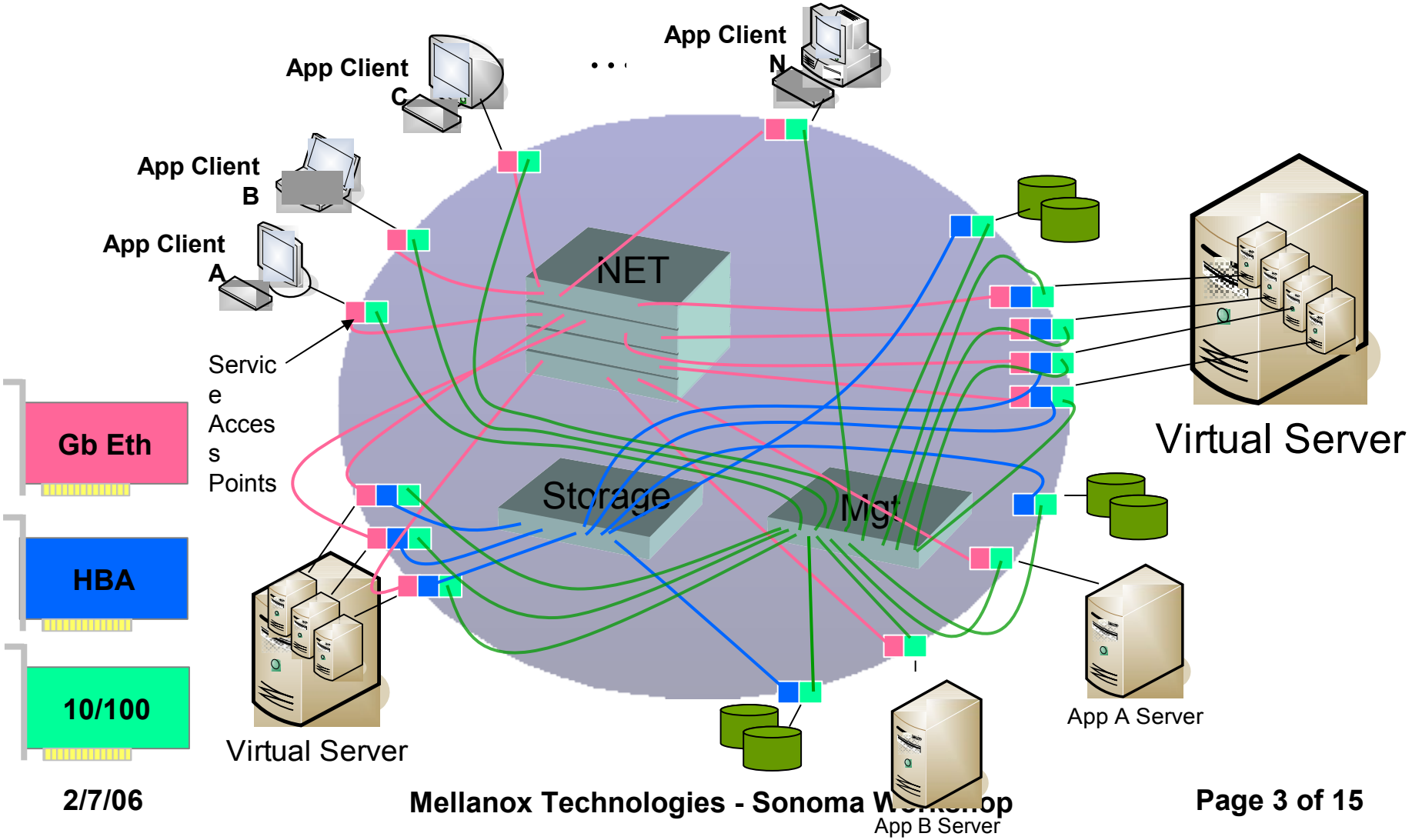
Quality of Service in InfiniBand Networks

Eitan Zahavi, Mellanox Technologies

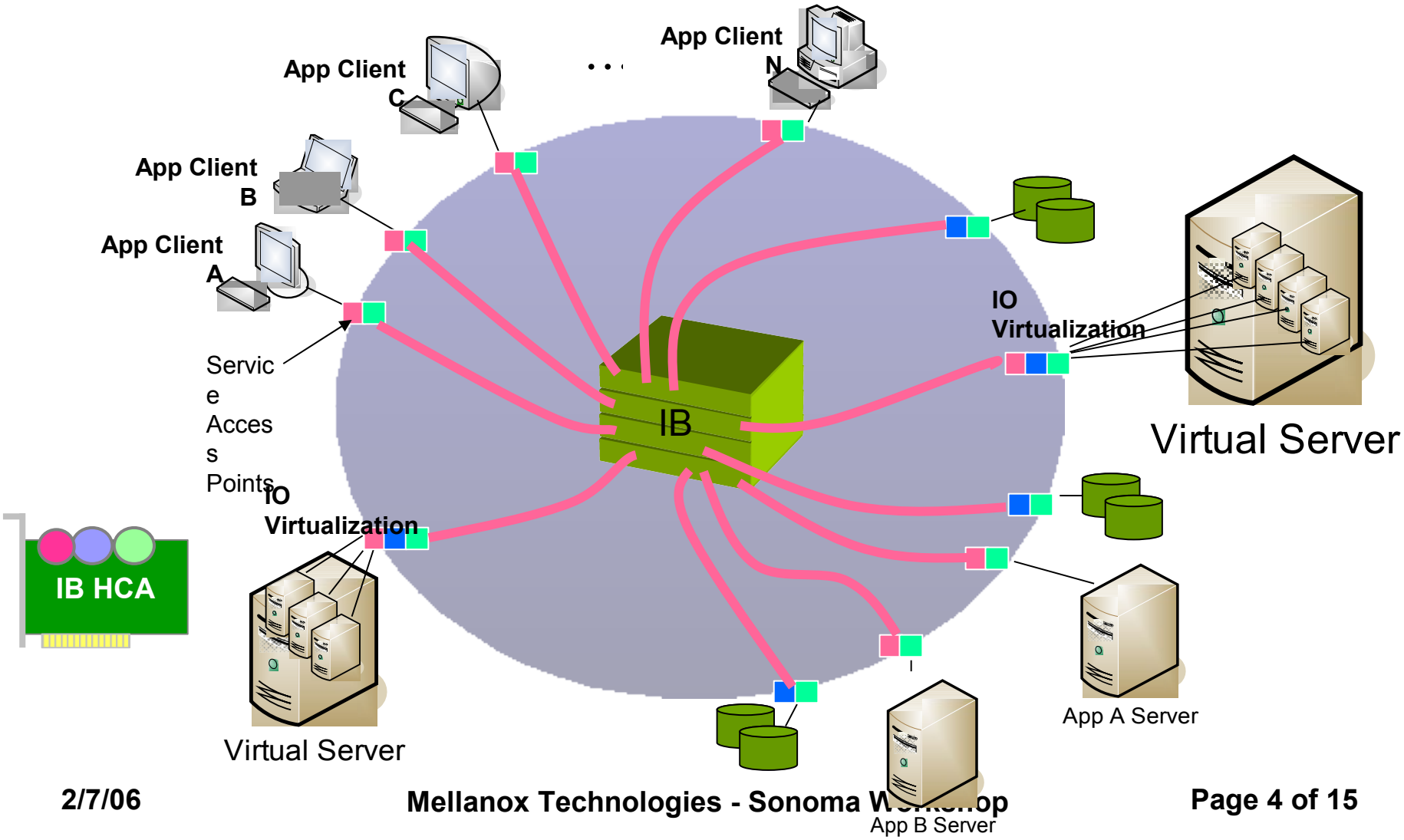
Agenda

- IO Consolidation
- IB mechanisms for QoS
- IB QoS demo
- QoS Proposal
  - Three policy levels
  - Implementation

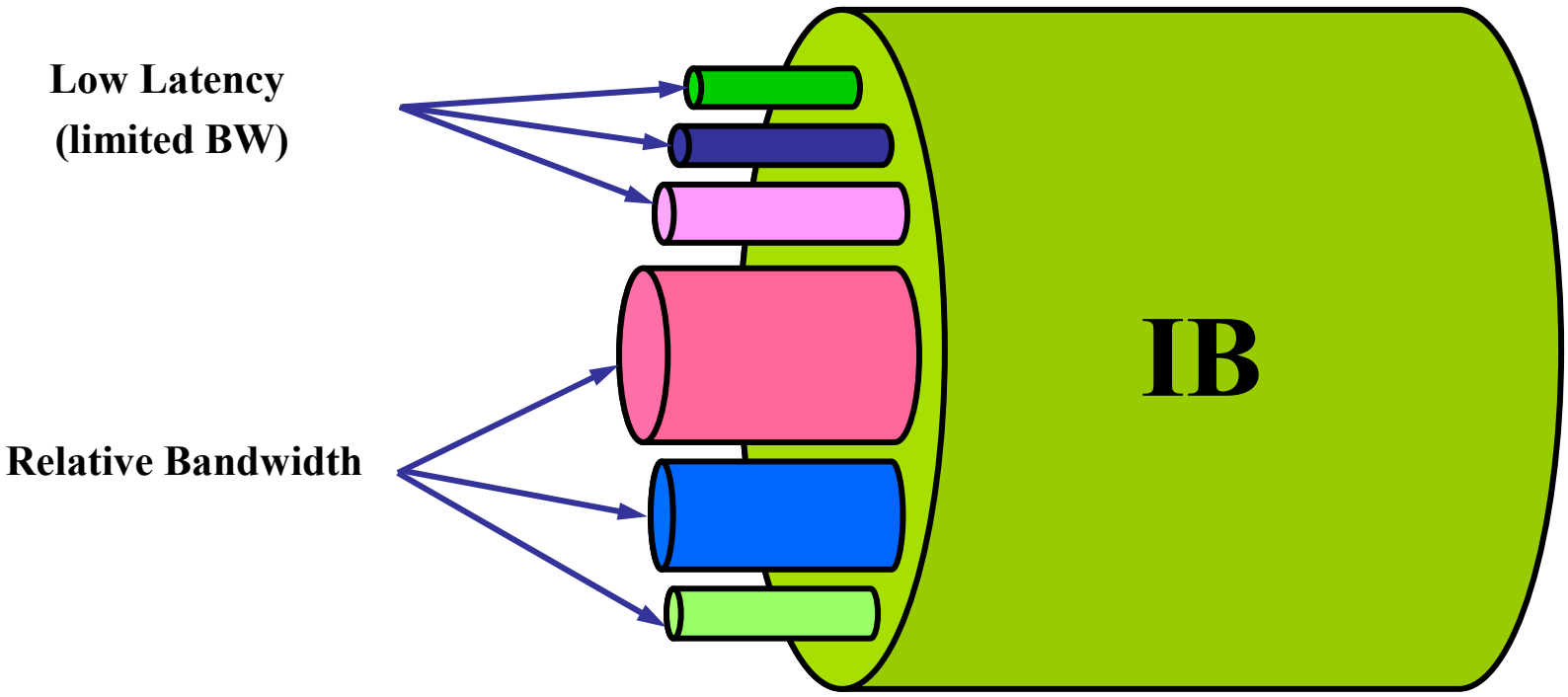
Data Center → Many Wires



IO Consolidation → One Wire



IO Virtualization

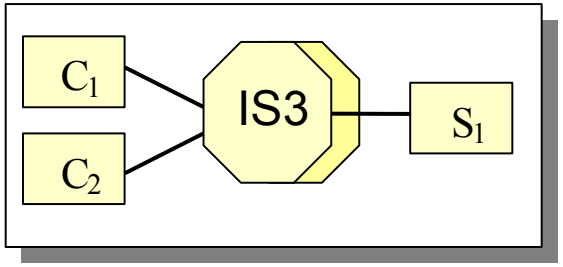


IB QoS Hardware Features

- Virtual Lanes
  - Up to 16 packet queues
- SL
  - Packet Class of Service (CoS) mark ala 802.1p or DSCP
  - Marking performed at the network edge
- SL2VL
  - Aggregating traffic by SL to queue
- VL Arbitration
  - High and Low priority WRR shapers
- Multipathing
  - Allow different paths to same target
- Congestion Control Inter-Packet Delay
  - Fine tuning of injection rate per VL or connection

QoS Demo Setup

- Three machines one/two switches
- Use SDP traffic
  - Map TCP Port to SL (through /proc)
  - Zero copy
  - Application is a multi threaded (4) echo client/server
- QoS Configuration
  - Setup by dedicated utility
  - Dynamically controlled High/Low relative bandwidth settings



QoS Level	QL0	QL1	QL2	QL3	QL4	QL5
IB SL	SL0	SL1	SL2	SL3	SL4	SL5
Arbitration Priority	Low	High	Low	High	Low	High
HCA egress VL	VL 0	VL 1	VL 2	VL 3	VL 4	VL 5
HCA Relative BW	***HRBW 0***	***HRBW 1***	***HRBW 2***	***HRBW 3***	***HRBW 4***	***HRBW 5***
SW egress VL	VL 0	VL 1	VL 2	VL 3	VL 4	VL 5
SW Relative BW	***SRBW 0***	***SRBW 1***	***SRBW 2***	***SRBW 3***	***SRBW 4***	***SRBW 5***

Dynamically Configurable

QoS Demo Results

- Total BW / 4x link
  - 916MBytes/Sec
- QoS Dynamic Range / BW Ratio
  - Three parallel clients on each machine
  - Up to 6.8:1 for clients running on same server
  - Up to16:1 for clients running on different servers
- Low latency SLs
  - Up to 3:1 BW for the low weight higher priority flow vs. higher weight SL
- Isolation
  - Stopping one VL does not affect the other

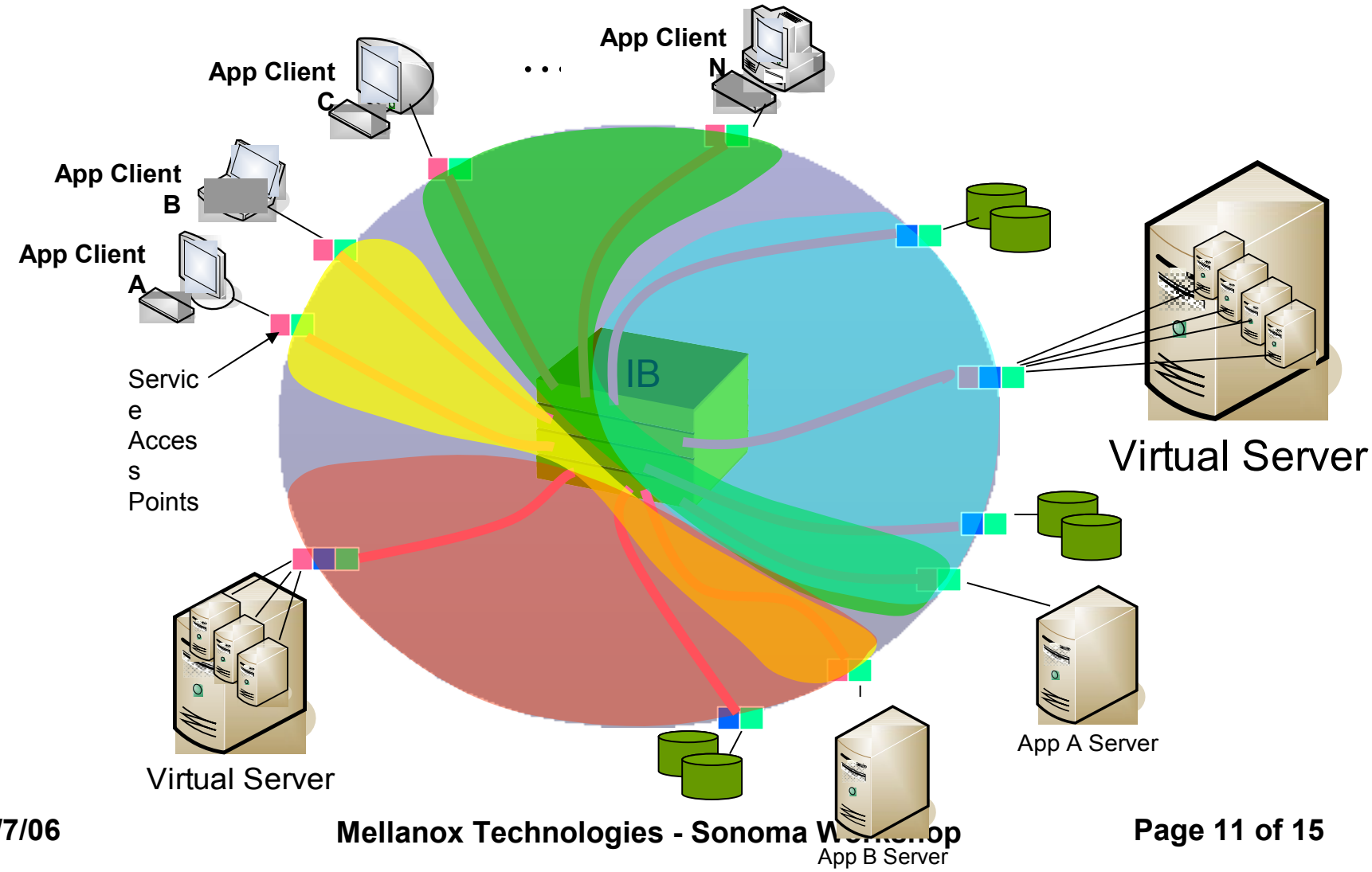
IBTA LWG - QoS Annex

- Several years of effort to specify how QoS should be done in IB
- Draft Annex exists
- Many open questions regarding:
  - Defining mechanisms for bandwidth reservation?
  - Where policy is defined?
  - How it is enforced?
- Our proposal
  - Relies only on existing IB spec
  - Follows the existing Annex and discussions
  - Provides a first implementation

Proposed QoS Policy

- QoS Level (QL) should define
  - Fabric setup: SL2VL, VLArb, LMC, LFT
  - PathRecord: SL, path-bits, MTU, rate
- Three levels of policy refinement
  - Subnet/Partition
    - All member share same QL
  - End-to-End and CoS
    - Going from A to B on a specific SL
  - Service Based
    - Negotiated during connection establishment

Subnet/Partition Level Policy



End-to-End Level Policy

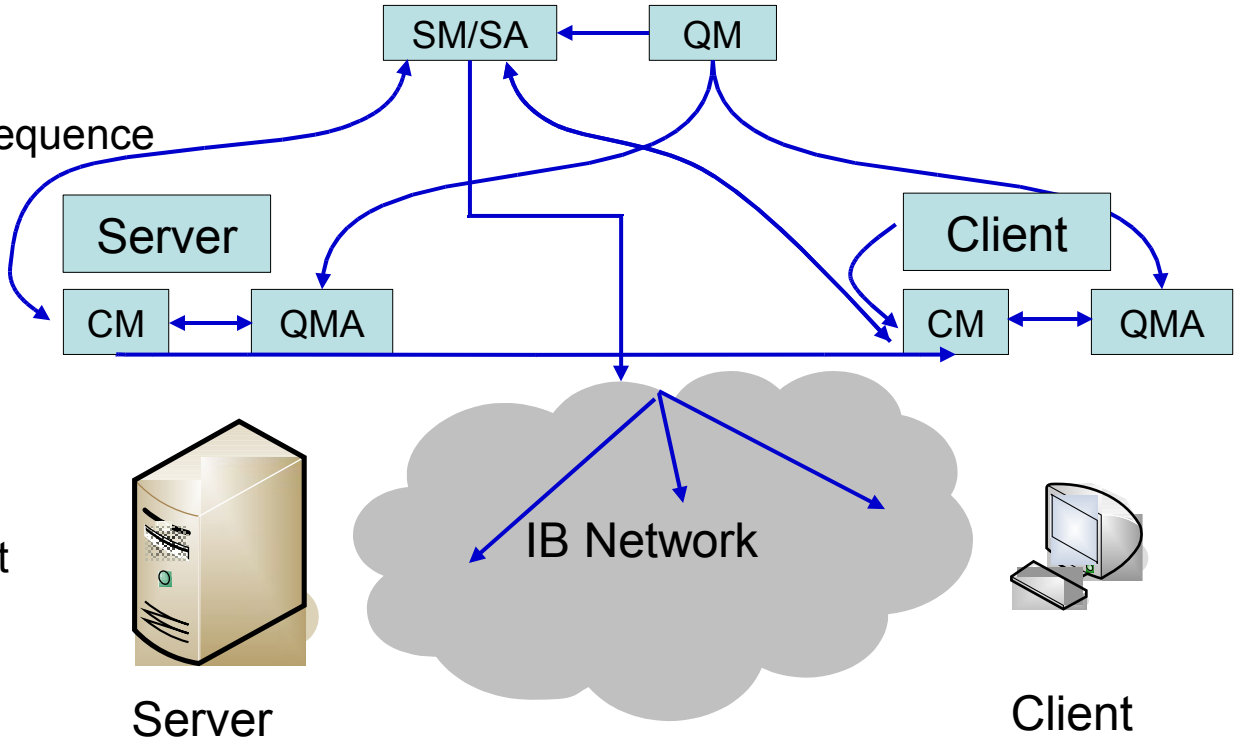
- Diff-Serv Code Point (DSCP)
  - Commonly used to mark a flow quality of service
  - Known and used by many existing applications
  - In IB can map to SL and be carried in each packet header
- Define QoS Level by source, destination and DSCP
- Policy Examples
  - All communication into storage server
  - All communication into Virtual Server
  - Specific application can request Path Record for low latency (by specifying DSCP)
  - Access Cluster Storage lock manager with high priority
- QL Provided by the SA
  - PathRecord requests carry Source, Destination and DSCP
  - PathRecord response carry the QL params

Service Based

- Services follow the “Client Server” model
- QL negotiated during connection establishment
- Negotiation may be implemented as
  - Just a simple /proc file
  - Communication with a QoS Manager (a server by itself)
  - Communication with a local QoS Management Agent
- Support for Server side override for client side QL
  - Important in case the client cheats or not QoS aware
  - The server might know better

Software Blocks and Flow

1. QM sets SM policy
2. SM configures fabric PHB
3. QM configures QMA
4. Client starts connection sequence
5. CM_{client} finds the server
6. CM_{client} negotiates QL
7. CM_{client} PathRecord
8. CM_{serv} negotiates QL
9. CM_{serv} opt. overrides client PathRecord
10. Connection Established



SM/SA Enhancements

- SM/SA Support QoS Policy
  - Let the QM define QLs
    - In terms of SL and path bits (MTU and IPD)
    - Define their relative BW and latency (low, high)
  - Define QL per partition
  - Enable {SRC,DST,TClass} mapping to QL
  - QL based routing optionally:
    - Preserve All-to-All links for QL
    - Dynamically reroute when path-bits are used
- Distributed SA cache
  - Use DSCP as caching key

CMA Enhancements

- Map IP to partition
  - Use that P_Key in the PathRecord query
- Add interface to provide requested DSCP
  - If specified use it as the PathRecord TClass field
- Add plugin interface for QoS Agent
  - The agent should provide back DSCP
- Enable Server PathRecord override
  - PathRecord should be included in the REP

Sonoma Workshop

Thank You

Special thanks to Yaron Haviv (Voltaire)
for his support, discussions and ideas