Subnet Management (SM) and Subnet Adminstration (SA) in OpenFabrics

OPENFABRICS ALLIANCE

Hal Rosenstock

4/19/13

Mellanox® TECHNOLOGIES

Agenda

- **IB Management Architecture Overview**
- **IB Management Tools Overview**
- **OpenSM**

IB Management Architecture Overview

IB Management Architecture Overview Agenda

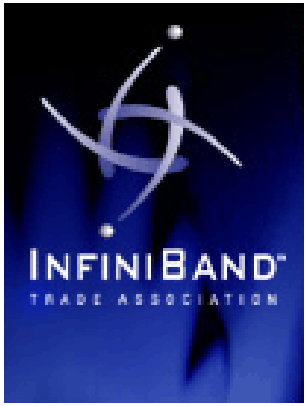
- InfiniBand Trade Association (IBTA)
- IB Management Architecture and IBTA
- What is a subnet ?
- Subnet Model
- Basic Management Concepts
- Management Model
- Objectives of Subnet Management
- Path Information
- SM Maintenance of Subnet

IB Management Architecture Overview Agenda

- Subnet Administration (SA) Information
- SA Functions
- SM/SA Architecture
- Relationship between SM and SA
- Performance Management
- Congestion Management
- Quality of Service

InfiniBand Trade Association (IBTA)

- **Founded in 1999**
- **Actively markets and promotes InfiniBand from an industry perspective through public relations engagements, developer conferences and workshops**
- **Steering Committee (SC) Members**
  - **Cray, HP, IBM, Intel, Mellanox, Oracle**



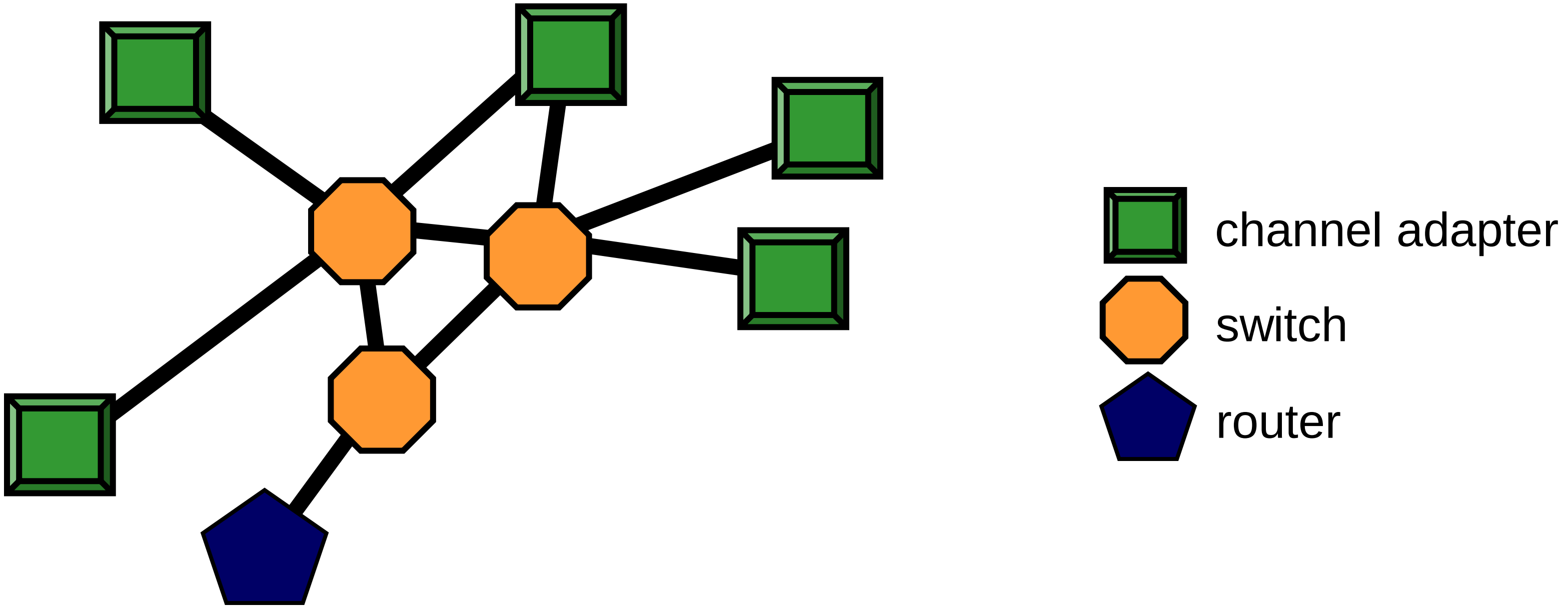
IB Management Architecture and IBTA

- **InfiniBand Architecture (IBA) is specified by InfiniBand Trade Association (IBTA)**
  - Currently at version 1.2.1 – issued November 2007
  - In two volumes (1 & 2)
- **IB Management is specified in chapters 13-16 in volume 1**
  - Chapter 13: Management Model
  - Chapter 14: Subnet Management
  - Chapter 15: Subnet Administration
  - Chapter 16: General Services
- **Also, various annexes of interest in volume 1**
  - Annex A10: Congestion Control
  - Annex A13: Quality of Service
  - Hierarchy Annex

IB Management Architecture

- **IBA has evolved beyond 1.2.1**
  - **Errata (primarily MgtWG and LWG)**
  - **1.3 Volume 2 – Extended Link Speeds (EWG)**
- **Released specs and errata are available as non member**
  - **http://www.infinibandta.org/content/pages.php?pg=technology_download**

What is a subnet ?



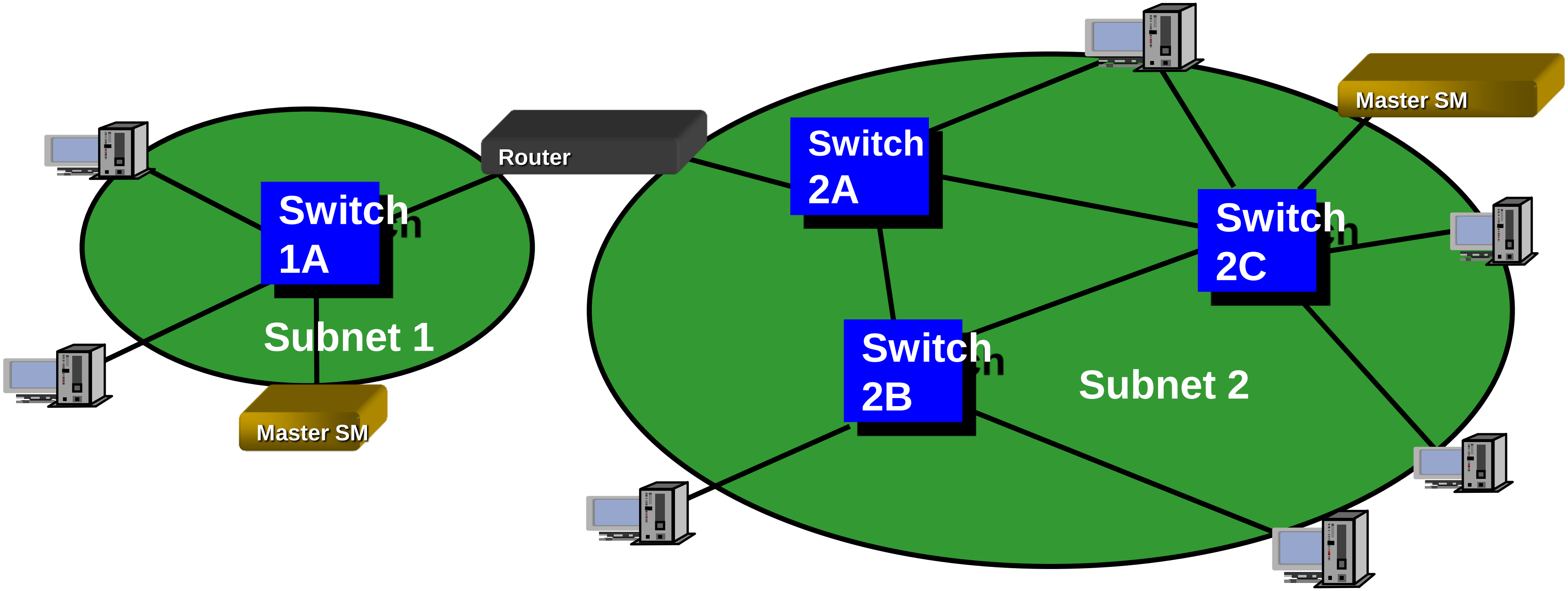
🟩 channel adapter

⯃ switch

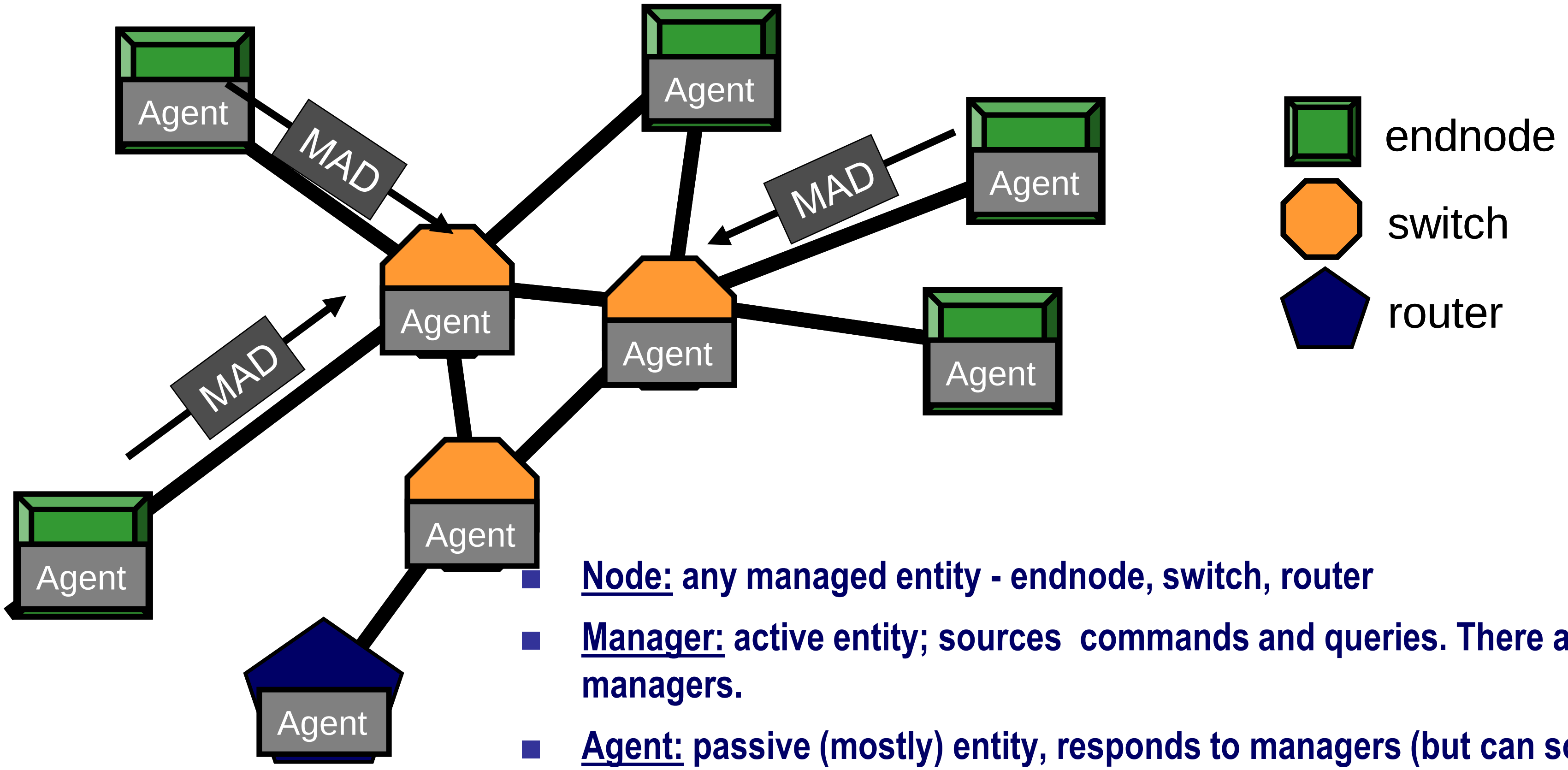
⬟ router

Subnet Model

- Subnet = HCAs and TCAs interconnected through switches
- Each subnet has its own LID space
- Each subnet has at least one SM and exactly one (logical) Master SM
  - after initialization, mastership could be a distributed function
- Fabric = subnets interconnected through routers



Basic Management Concepts

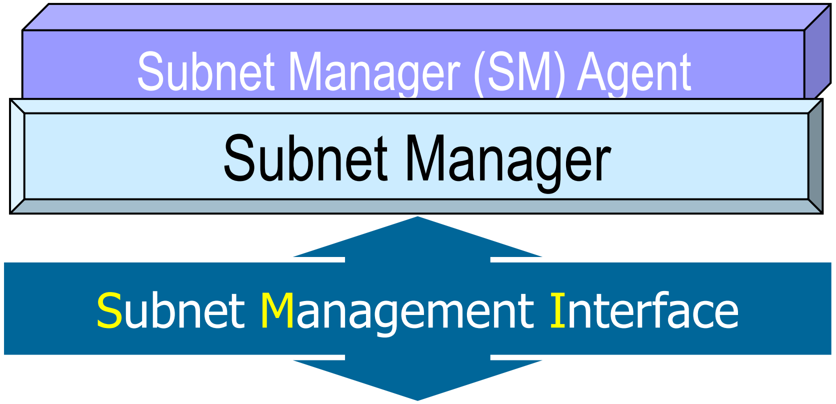


- **Node:** any managed entity - endnode, switch, router
- **Manager:** active entity; sources commands and queries. There are few managers.
- **Agent:** passive (mostly) entity, responds to managers (but can source traps). Many agents.
- **Management Datagram (MAD):** standard message format for manager–agent 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...**

Management Model

Pure InfiniBand Management



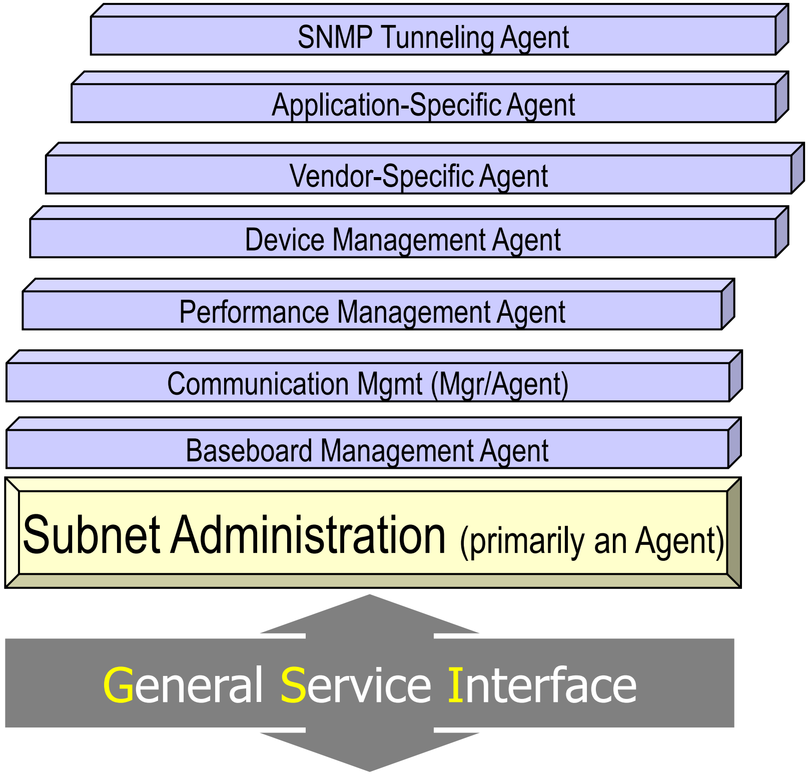
Subnet Manager (SM) Agent

Subnet Manager

Subnet Management Interface

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

Other Management Features



SNMP Tunneling Agent

Application-Specific Agent

Vendor-Specific Agent

Device Management Agent

Performance Management Agent

Communication Mgmt (Mgr/Agent)

Baseboard Management Agent

Subnet Administration (primarily an Agent)

General Service Interface

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

SM Class Attributes

- **Node attributes:**
  - NodeInfo (type, numPorts, version, GUIDs, deviceID…)
  - NodeDescription
- **Port attributes:**
  - PortInfo (M_Key, LIDs, state, capabilities, VLs, width, speed, MTU, Master…)
  - GUIDInfo
  - SLtoVLMapping
  - VLArbitration
  - Partition table

- Switch attributes:
  - SwitchInfo
  - LinearFDB
  - RandomFDB
  - MulticastFDB
- SM attribute:
  - SMInfo
- Router attribute:
  - RouterInfo (TBD)

Objectives of Subnet Management

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

Path Information

- **The Subnet Manager establishes / defines paths through the subnet**
  - does so using SMPs that set switch forwarding tables, LIDs, etc.
- **Subnet Administration responds to path resolution requests, using GMPs**
- **Path Record returned by SA contains:**
  - Local header: DLID, SLID, SL
  - Global header: DGID, SGID, (TClass, FlowLabel, HopLimit)
  - Properties: MTU, Rate, Latency, P_Key

SM Initialization of Subnet

- Physical subnet establishment
- Subnet discovery
- Information gathering
- Path determination
- Port configuration
- Switch configuration
- Subnet activation

SM Maintenance of Subnet

- Topology changes
- SM Actions
- SM state machine
- Determination of the Master
- Mastership handover
- Mastership failover
- Handling topology changes

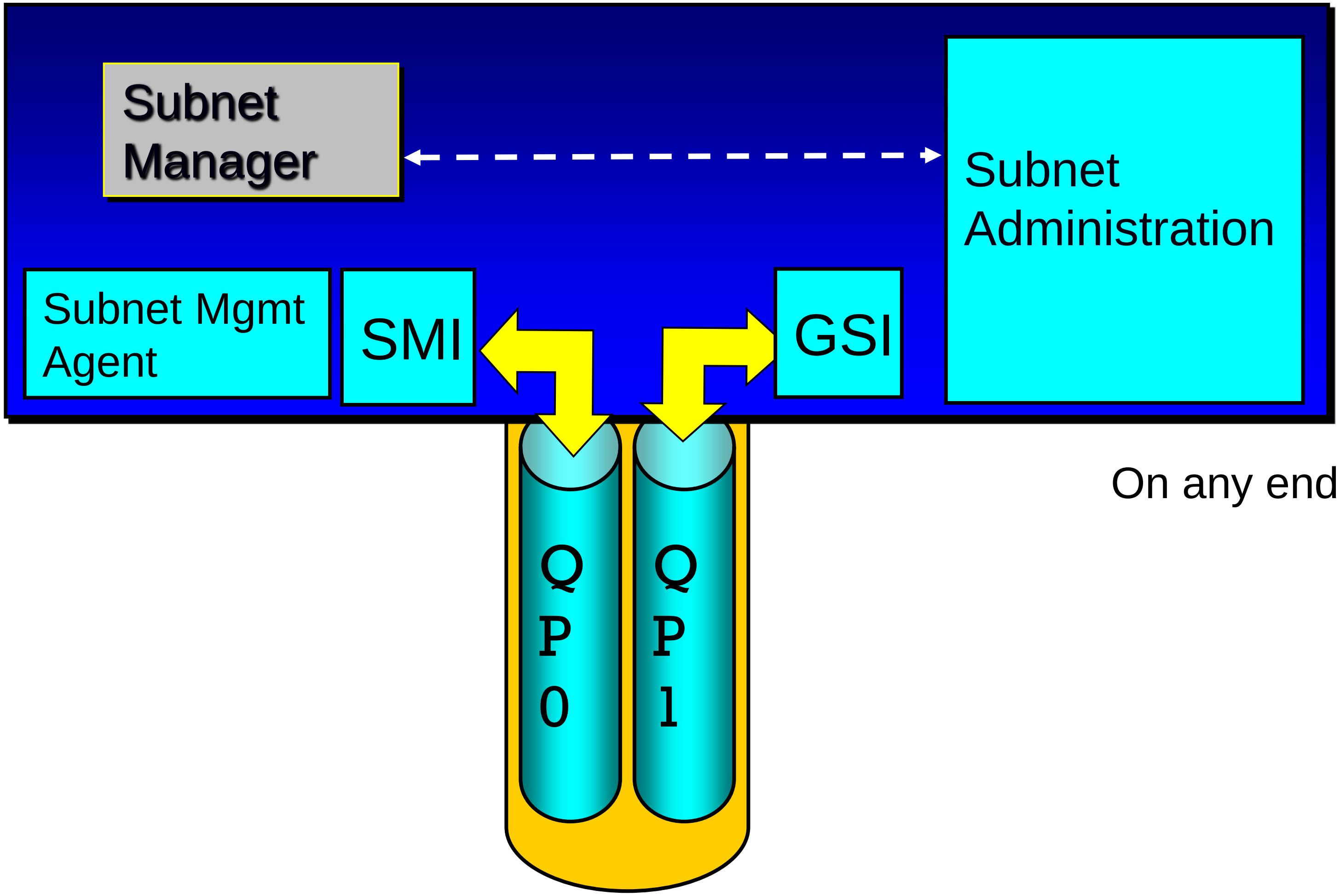
Subnet Administration (SA) Information

- **SA provides access to & storage of three kinds of information:**
  - info that endnodes need to operate on the subnet
    - examples: paths, multicast info, services
  - info that is non-algorithmic, typically entered by a network administrator
    - examples: partitioning information, SL to VL mappings, etc.
    - required for level 3 interoperability (move to different manager)
  - info that may be useful to, e.g., standby SMs
    - example: network topology, GUIDs of nodes, etc.
    - this is ***optional***

SA Functions

- **The SM and SA cooperate to provide this information**
- **To provide that information, SA has two major functions:**
  - A query subsystem to identify type of information sent and received
  - An event reporting subsystem to forward SM traps as Report() MADs to subscribers
- **All subnets must have an SA**

SM/SA Architecture



On any end port

Relationship between SA and SM

- **The SA is part of the SM**
  - Effectively, it's the SM talking in "normal" packets to clients on the subnet
  - "Tightly" coupled to SM in IBA
- **Discussed separately for convenience of description only**
- **SM-SA communication is "vendor" specific**
  - SA can be on a node different from SM
    - – Redirection could be used to accomplish this
- **If SM is master, the (or a) related SA must also be master**
- **If an SM ceases to be master, a (or the) related SA must also cease to be master**

(watch out when SA & SM are on different nodes)

SA Attribute Records

State Records

- NodeRecord
- SwitchRecord*
- PortInfoRecord
- PartitionRecord
- SLToVLMappingTableRecord
- LinearForwardingTableRecord*
- RandomForwardingTableRecord*
- MulticastForwardingTableRecord*
- SmInfoRecord*
- LinkRecord*
- GuidInfoRecord*
- PathRecord
- MultiPathRecord
- InformInfoRecord
- Notice* (Traps and Notices) (not record)

Subscription Records

- ServiceRecord (Service Advertisement)
- MCMemberRecord*
- InformInfo (not record)

* = optional

Performance Management

- **Purpose:**
  - Allows retrieval of performance and error statistics from InfiniBand™ components
  - Provides a means of sampling a specified quantity over a specified interval
  - IBA specifies only the PerfMgt Agent; manager is “beyond the spec”

Congestion Management

- **Specified in Annex A10 (updated in LWG errata 8/19/10)**
- **Avoid/eliminate congestion spreading**
- **Limit flow injection rate at ports which are root cause of congestion**

Congestion Management

- **Accomplished via FECN/BECN mechanism**
  - Formerly reserved bits in BTH header
  - CN (congestion notification) opcode is 0b'10000000
    - Transparent to version 1.1 or earlier switches
  - B (BECN): 0 indicates that no congestion was encountered; 1 indicates that the packet indicated by this header was subject to forward congestion. The B bit is set in an ACK on CN BTH.
  - F (FECN): 0 indicates that it probably did not go through a point of congestion; 1 indicates that the packet went through a point of congestion.

Quality of Service

- **Specified in Annex A13 (updated in LWG errata 8/19/10)**
- **QoS scheme providing different *relative* classes of service**
- **Requester specifies priority by specifying the type of priority management requested via a 2 bit QoSClass.Type and 8 bit QoSClass.Priority field in [Multi]PathRecord**
  - Currently, the only type of priority management defined is DiffServ compatible
  - All other values for the QoSClass.Type are reserved
- **Requester can also specify ServiceID field in PathRecord/MultiPathRecord**
  - Also used in subsequent CM REQ messages

IB Management Tools Overview

IB Management Tools Overview Agenda

- **OpenFabrics**
  - OpenFabrics Stack
  - OpenFabrics Software
- **OpenSM**

OpenFabrics

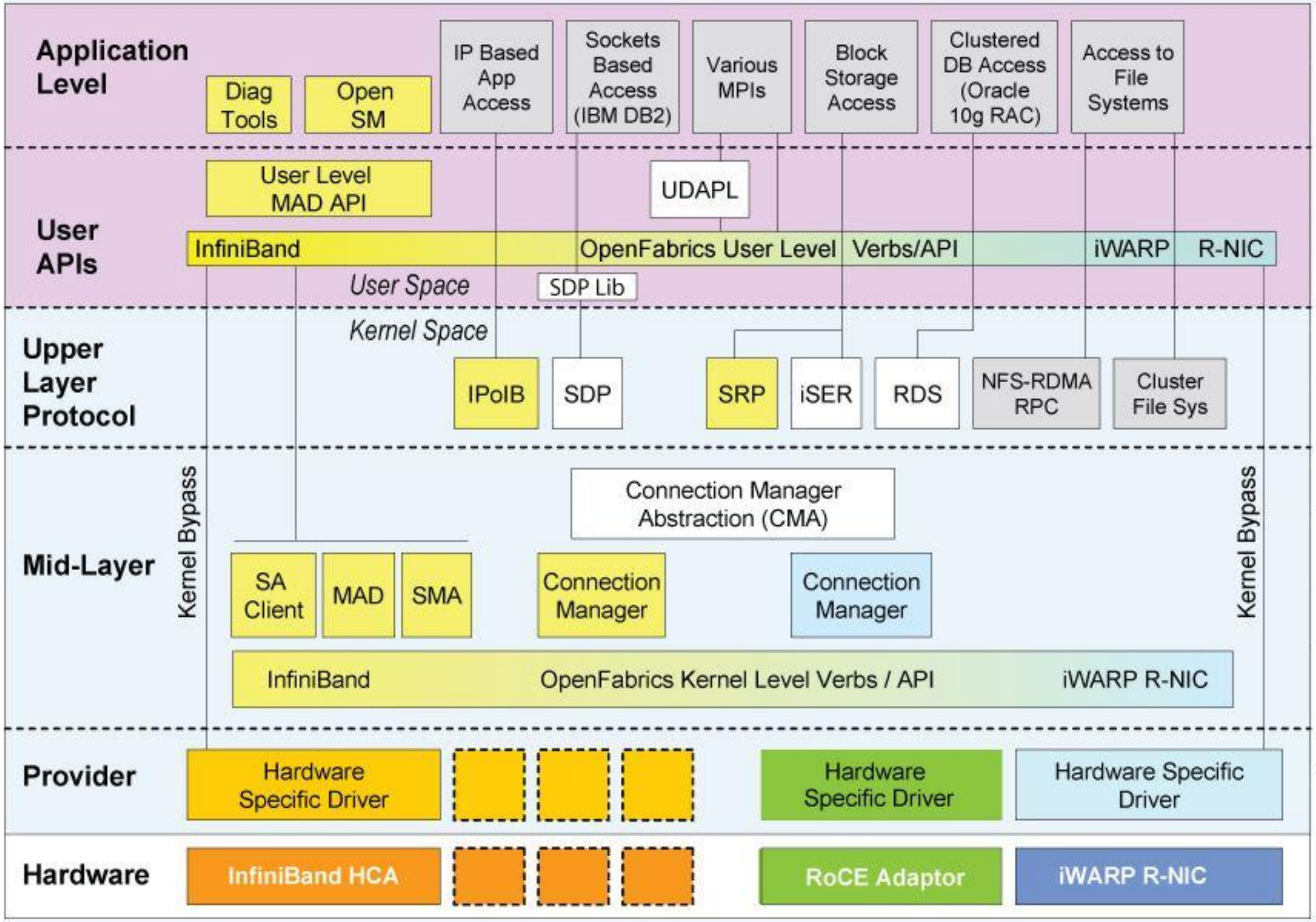
- **Open source InfiniBand software is developed under OpenFabrics Open Source Alliance**

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

  - Primarily targeted at Linux and Windows
    - Other OS ports have been done

OPENFABRICS ALLIANCE

OpenFabrics Stack



Application Level

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

Kernel Space

Upper Layer Protocol

IPoIB

SDP

SRP

iSER

RDS

NFS-RDMA RPC

Cluster File Sys

Mid-Layer

Kernel Bypass

Connection Manager Abstraction (CMA)

SA Client

MAD

SMA

Connection Manager

Connection Manager

Kernel Bypass

InfiniBand OpenFabrics Kernel Level Verbs / API iWARP R-NIC

Provider

Hardware Specific Driver

Hardware Specific Driver

Hardware Specific Driver

Hardware

InfiniBand HCA

RoCE Adaptor

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

🟩 RoCE

🟨 InfiniBand

🟦 iWARP

Apps & Access Methods for using OF Stack

OpenFabrics Software

- **Mellanox is current maintainer for OpenSM and libibumad**
  - I am current maintainer for OpenSM, libibumad, and ibsim
  - I was former maintainer for infiniband-diags and libibmad
- **Git trees' location is http://git.openfabrics.org/git/**
  - ~halr for opensm, libibumad, and ibsim
  - ~iraweiny for infiniband-diags and libibmad
- **Released tar balls location is http://www.openfabrics.org/downloads/management**
  - libibumad, libibmad, opensm, infiniband-diags, ibsim
- **Software is dual licensed**
  - BSD or GPL v2

OpenSM

OpenSM Agenda

- **What is OpenSM ?**
- **OpenSM Porting Layers**
- **OpenSM "History"**
- **OpenSM Gen2 Layering**
- **Introducing OpenSM**
- **Starting OpenSM**
- **OpenSM Command Line Options**
- **opensm.conf Parameters**
- **Logging**
- **Console**

OpenSM Agenda

- **Stages of Operation**
- **Partition Management**
- **Quality of Service (QoS)**
- **Routing**
  - Credit Loops/Deadlock
  - Unicast Routing Stages
  - Unicast Routing Algorithms
    - Up/Down Routing
  - Multicast Routing

OpenSM Agenda

- **SM High Availability**
- **Some Notable Features**
- **Windows OpenSM (and diagnostics)**
- **Performance Manager**
- **Congestion Manager**

What is OpenSM ?

- **InfiniBand compliant Subnet Manager and Administration**
- **Also contains an (optional) performance manager**
- **Now also contains an (optional) (experimental) congestion manager**
- **Approximately 2-3 releases/year**
  - Latest release is 3.3.16 (February 2013)

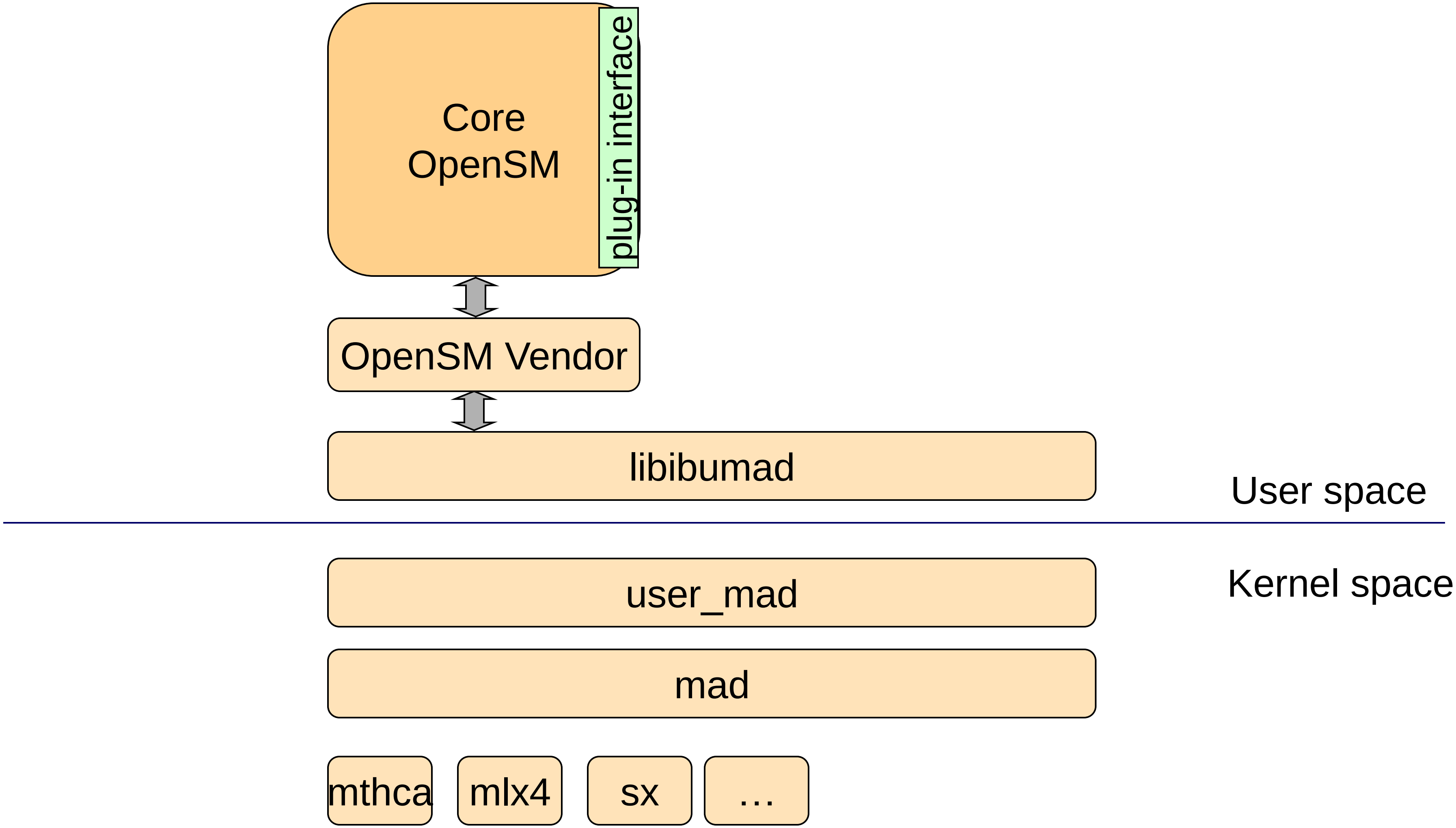
OpenSM Porting Layers

- **Component Library (complib)**
  - OS portability/abstraction layer
    - threads, timers, locking, events, various data structures
- **Vendor Library (libvendor)**
  - IB hardware portability layer
    - osm_vendor_ibumad
    - osm_vendor_ibal
    - Vendor layer for simulator (ibmgtsim not ibsim)
      - osm_vendor_mlx

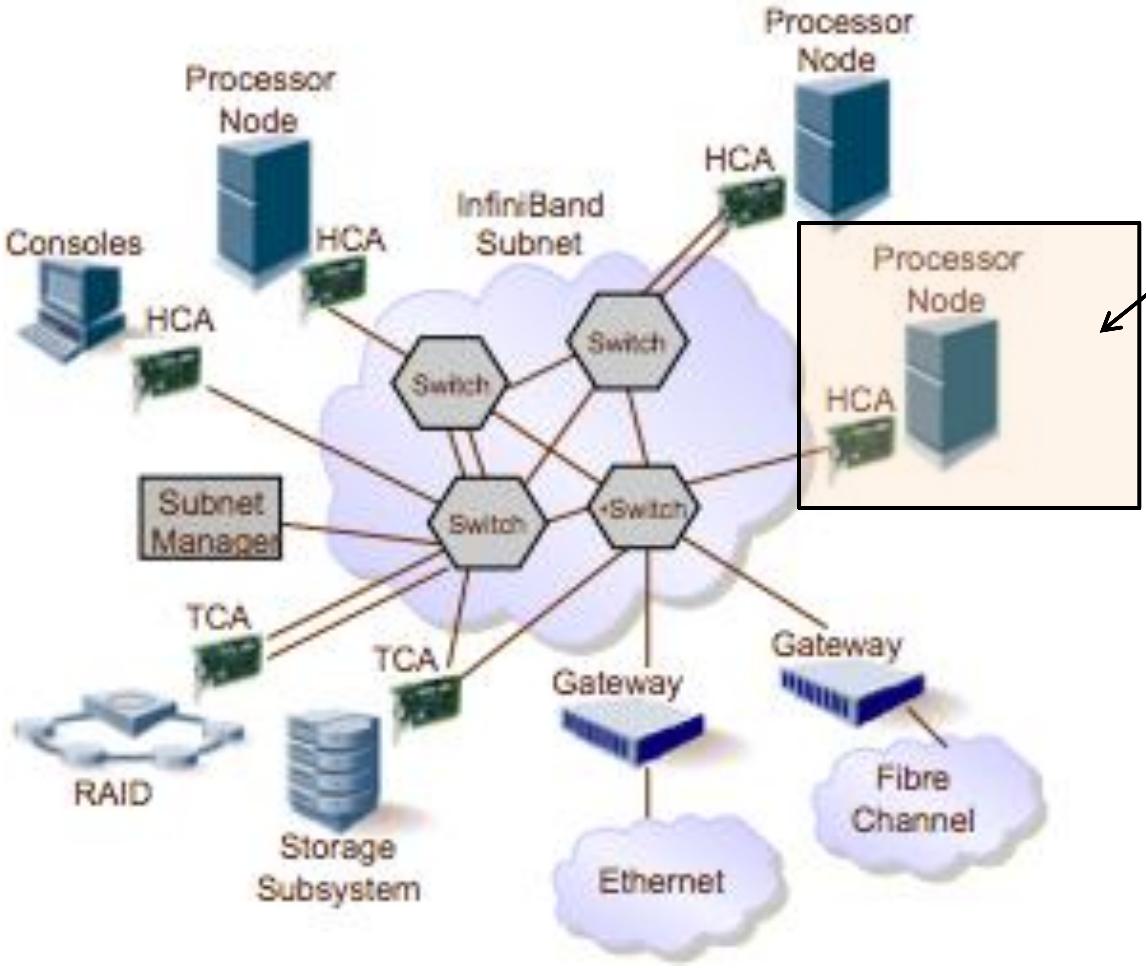
OpenSM “History”

- **Timeline**
  - Gen 1 (IBAL based vendor layer)
    - Intel – 2002 to mid 2003
    - Mellanox – mid 2003 to end 2004
  - Gen 2 (UMAD based vendor layer)
    - Voltaire – 2005 to 2010
      - PathForward “seeded” OpenIB/OpenFabrics
    - Now Mellanox again - 2011

OpenSM Gen2 Layering



Introducing OpenSM



Load OpenSM on a node/server

By default, in Linux, the **opensm** run is logged to two files: **/var/log/messages** and /var/log/opensm.log and in Windows, \Windows\Temp\osm.syslog and \Windows\Temp\osm.log

Starting OpenSM

- **Invoked via command line options and various configuration files (even more options)**
  - See extensive man page
  - Default options file
    - – Sample from OpenSM 3.3.11
- **Maintains log file**

Running OpenSM

- **Command line**
  - Default (no parameters)
    - Scans and initializes the IB fabric and periodically sweeps 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
    - Windows
      - \Windows\Temp\osm.syslog - opensm messages, registers only general major events
      - \Windows\Temp\osm.log – details of reported errors
    - Linux
      - /var/log/messages – opensm messages, registers only general major events
      - /var/log/opensm.log - details of reported errors
  - *Installed as service in Windows*
  - */etc/init.d/opensm start (Linux)*

If opensm has started correctly you should see SUBNET UP messages in the opensm logfile (/var/log/opensm.<PORTID>.log).

Running OpenSM

Start on Boot

- As a Linux 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) SMs running on the subnet
- saquery –s
  - Shows SMs on the subnet

OpenSM Command Line Parameters

- **A few important command line parameters:**

  -c, --create-config OpenSM will dump its configuration to the specified file and exit. This is a way to generate OpenSM configuration file template.

  -g, --guid This option specifies the local port GUID value with which OpenSM should bind. OpenSM may be bound to 1 port at a time. This option is used if the SM needs to bind to Port 2 of an HCA.

  -R, --routing_engine This option chooses routing engine instead of Min Hop algorithm (default). Supported engines: updn, dnup, file, ftree, lash, dor, torus-2QoS

  -x, --honor_guid2lid. This option forces OpenSM to honor the guid2lid file, when it comes out of Standby state, if such file exists under \Windows\Temp (in Windows) and /var/cache/opensm (in Linux)

  -V This option sets the maximum verbosity level and forces log flushing

opensm command line options

Command	Description
version	Prints OpenSM version and exits
guid	Specifies the local port GUID
lmc	The # of LIDs per port (power of 2)
priority	Specifies SMs priority master = highest
smkey	This will effect SM authentication (64 bits)
reassign_lids	Causes SM to reassign Lid’s to all end nodes
routing_engine	Chooses routing algorithm (default minhop)
no_default_routing	Prevents SM from falling back to default routing
do_mesh_analysis	Enables additional analysis for lash algorithm
lash_start_vl	Sets VL to use Lash (default 0)
sm_sl	Sets SL to use SM/SA (default 0)
connect_roots	Forces routing engines to connect between root
ucast_cache	Prevents ucast routing changes per heavy sweep

opensm command line options (Cont'd)

Command	Description
lid_matrix_file	Specifies the name of the lid matrix dump file
lfts_file	Specifies the name of the lft file to be loaded
sadb_file	Specifies the SA dump file
root_guid_file	Sets the root nodes GUID (updn or fat tree)
cn_guid_file	Sets compute nodes guids for Fat Tree
io_guid_file	Sets I/O nodes guids for Fat Tree
max_reverse_hops	Sets max hops the wrong way
ids_guid_file	Name of map file with set of Id's instead GUIDs
guid_routing_order_file	Sets order port GUIDs for Min-Hop and UpDn
torus_config	Defines the file name for extra config info needed
once	Causes SM to configure subnet once then exits
sweep <interval>	Specifies number of seconds between subnet sweeps
timeout (milliseconds)	Specifies transaction timeouts
retries <number> (def 3)	Specifies the number of retries used per transaction
maxsmps <number>	Specifies the number of VL15 SMP MADs per wire
console	Activates the SM console (default off)
ignore-guids	Defines set of ports guid to be ignored by link load algorithm

opensm command line options (Cont'd)

Command	Description
hop_weights_file	Provides the means to define weighting factor per port
dimn_ports_file	Provides the means to define mapping between ports
honor_guid2lid (default false)	Forces SM to honor the guid2lid file (when coming out of standby state)
log_file	Defines the log to given file name (default /var/log/opensm.log)
log_limit	Defines the max file size in MB
erase_log_file	Causes deletion of file if previously exists (default is accumulative)
Pconfig (default partitions.conf)	Defines the optional partition configuration file
no_part_enforced	Disables partition enforcement on switch ext ports
ar	Enables Adaptive Routing Manager (ARM) on SM
ar_config_file	Specifies optional Adaptive Routing config file
qos	Enables QoS setup
qos_policy_file	Defines optional QoS policy file (default qos-policy.conf)
stay_on_fatal	Will cause SM not to exit on fatal initialization
daemon	Will run SM in the background
inactive	Start SM in inactive rather than normal init state
prefix_routes_file	Specifies how SA responds to path record queries

opensm.conf Parameters

- ***opensm.conf*** **provides full access to all OpenSM internal options which control various aspects of its operation. The available options are listed in the following table:**

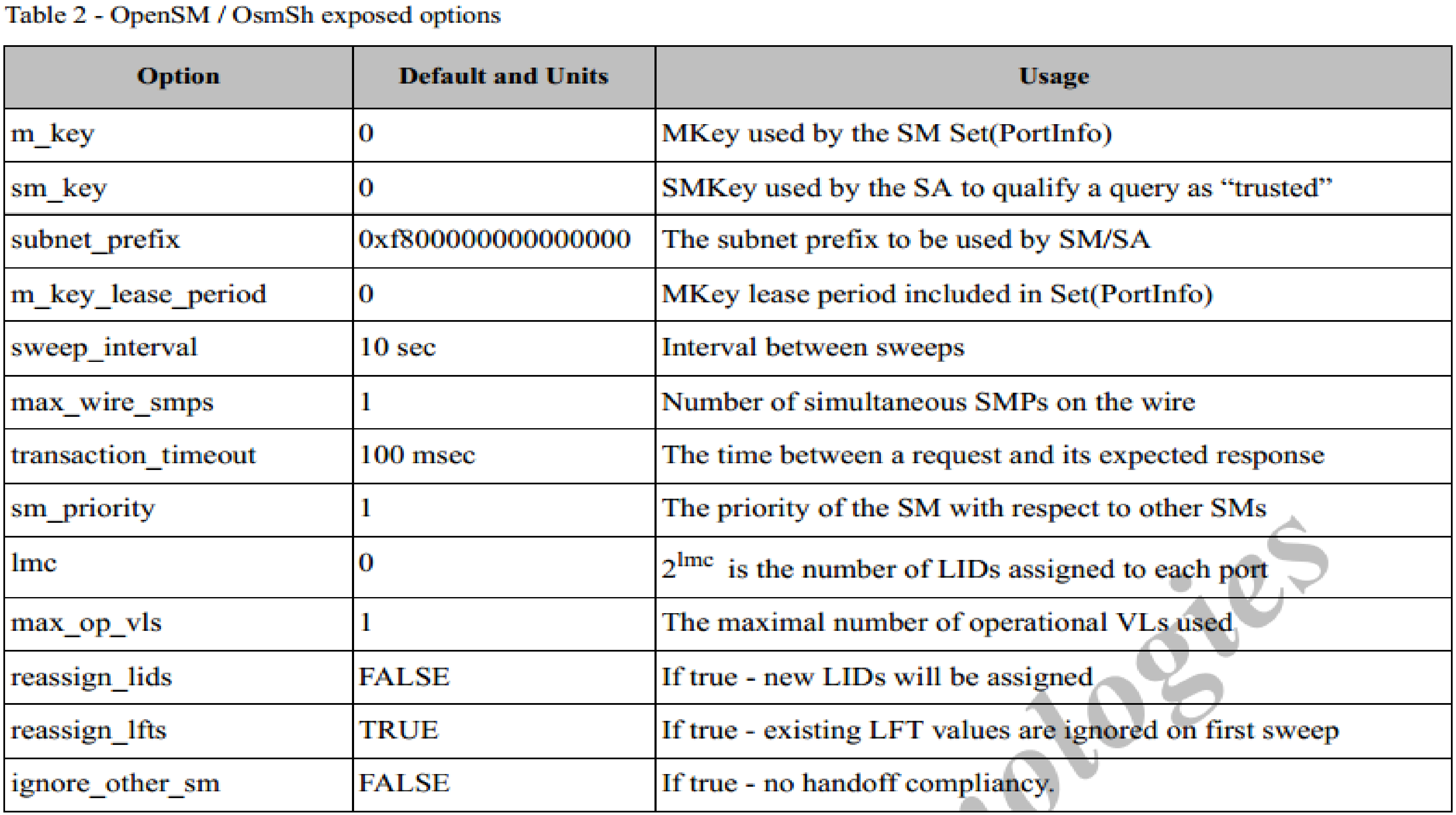
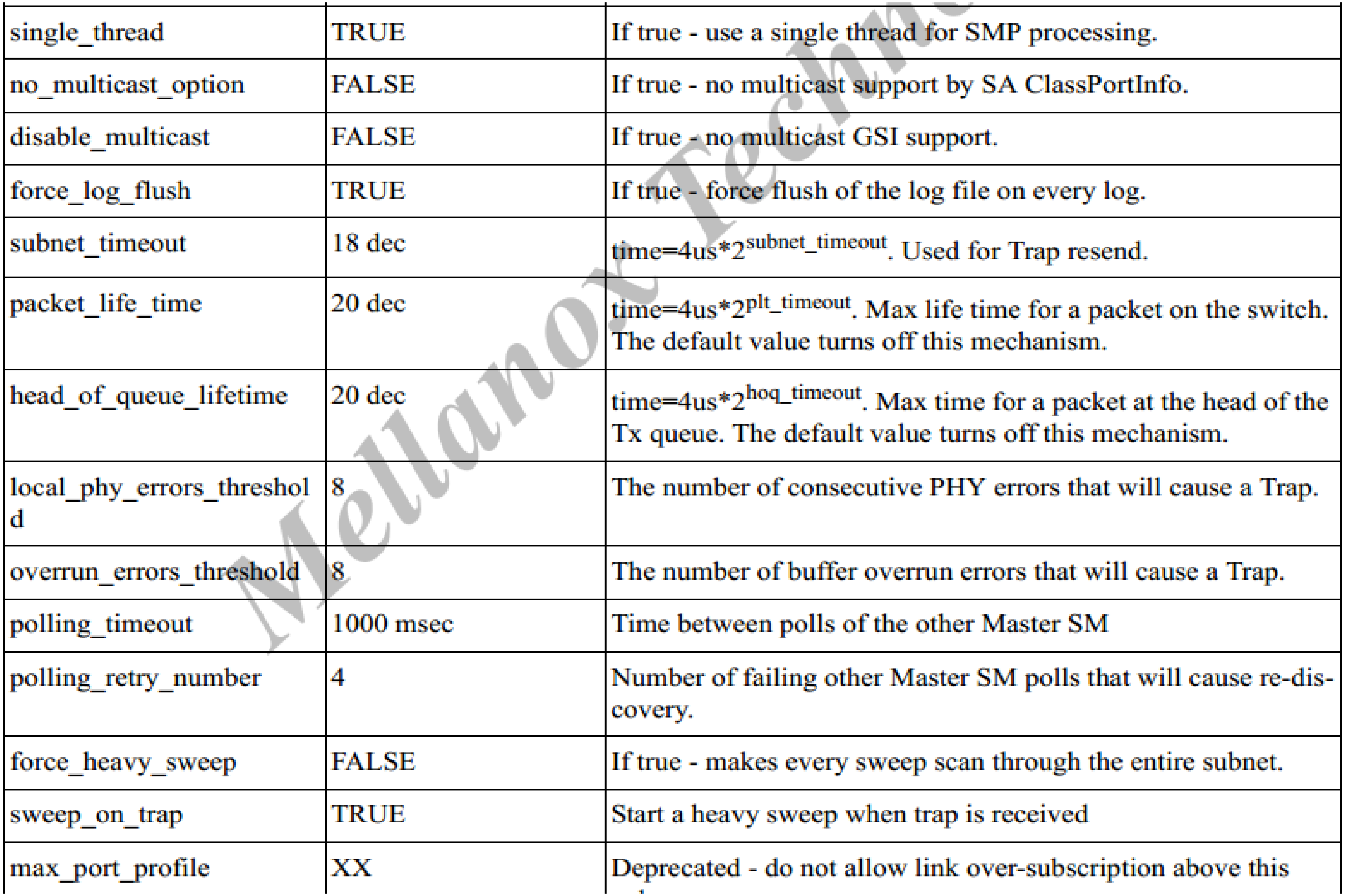


Table 2 - OpenSM / OsmSh exposed options

Option	Default and Units	Usage
m_key	0	MKey used by the SM Set(PortInfo)
sm_key	0	SMKey used by the SA to qualify a query as “trusted”
subnet_prefix	0xf800000000000000	The subnet prefix to be used by SM/SA
m_key_lease_period	0	MKey lease period included in Set(PortInfo)
sweep_interval	10 sec	Interval between sweeps
max_wire_smps	1	Number of simultaneous SMPs on the wire
transaction_timeout	100 msec	The time between a request and its expected response
sm_priority	1	The priority of the SM with respect to other SMs
lmc	0	2^{lmc} is the number of LIDs assigned to each port
max_op_vls	1	The maximal number of operational VLs used
reassign_lids	FALSE	If true - new LIDs will be assigned
reassign_lfts	TRUE	If true - existing LFT values are ignored on first sweep
ignore_other_sm	FALSE	If true - no handoff compliancy.

opensm.conf Parameters



single_thread	TRUE	If true - use a single thread for SMP processing.
no_multicast_option	FALSE	If true - no multicast support by SA ClassPortInfo.
disable_multicast	FALSE	If true - no multicast GSI support.
force_log_flush	TRUE	If true - force flush of the log file on every log.
subnet_timeout	18 dec	time=4us*$2^{subnet\_timeout}$. Used for Trap resend.
packet_life_time	20 dec	time=4us*$2^{plt\_timeout}$. Max life time for a packet on the switch. The default value turns off this mechanism.
head_of_queue_lifetime	20 dec	time=4us*$2^{hoq\_timeout}$. Max time for a packet at the head of the Tx queue. The default value turns off this mechanism.
local_phy_errors_threshold	8	The number of consecutive PHY errors that will cause a Trap.
overrun_errors_threshold	8	The number of buffer overrun errors that will cause a Trap.
polling_timeout	1000 msec	Time between polls of the other Master SM
polling_retry_number	4	Number of failing other Master SM polls that will cause re-discovery.
force_heavy_sweep	FALSE	If true - makes every sweep scan through the entire subnet.
sweep_on_trap	TRUE	Start a heavy sweep when trap is received
max_port_profile	XX	Deprecated - do not allow link over-subscription above this

Logging

- **Levels control the amount of information logged**
  - Error (error messages)
  - Info (basic messages, low volume)
  - Verbose (interesting stuff, moderate volume)
  - Debug (diagnostic, high volume)
  - Functions (function entry/exit, very high volume)
  - Frames (dumps all SMP and GMP frames)
  - Routing (dump FDB routing information)
  - Without –D option, OpenSM defaults to ERROR + INFO
- **Other logging features**
  - Limit log file size
  - Log rotation
  - Erase log file (at startup)
  - Now (at OpenSM 3.3.15): per module logging

Console

- **Optional**
  - Local
  - Socket
    - telnet
      - Can limit to loopback IP address
    - SSL not yet supported
- **Commands**
  - loglevel
  - priority
  - resweep
  - reroute
  - sweep

Console

- **More commands**
  - status
  - logflush
  - querylid
  - portstatus
  - switchbalance
  - lidbalance
  - dump_conf
  - update_desc
  - version
  - perfmgr
  - dump_portguid

Stages of Operation

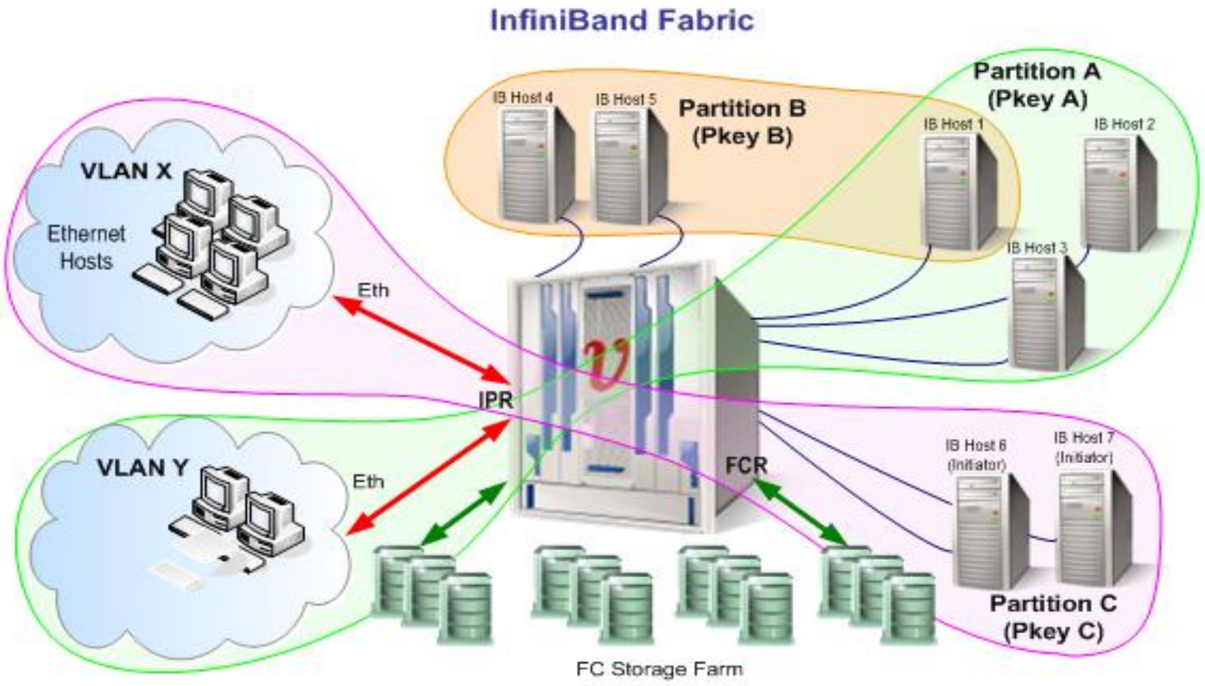
- **Discovery (Heavy sweep)**
- **If master, configuration**
  - PKey/QoS setup
  - SM LID configuration
  - LID configuration
  - Switch configuration
    - Unicast
    - Multicast
  - Link/Port configuration
    - Set state to Armed and then Active
  - Subnet is up if all sets worked
    - Otherwise, another heavy sweep is invoked

Stages of Operation

- **Once subnet up, then light sweeping**
  - Poll SwitchInfo for PortStateChange
    - If topology change, trigger heavy sweep
- **Traps can also indicate significant change triggering heavy sweep**
  - Link state change (trap 128)

Partition Management

- Define partitions
- Assign end ports (Port GUID) to one or more partitions
- Provide level of access control within the fabric
  - Full or Limited membership
- End ports can be members of multiple partitions



Partition Management

- **If enabled, partition enforcement is set on leaf links**
  - Leaf link is between switch external port and xCA (or router) port
- **Partition syntax**
  - Full/limited
  - Also, ipoib on partition preconfigures the IPoIB broadcast group
    - Rate, SL, MTU
  - Can now also preconfigure multicast groups
    - New feature at OpenSM 3.3.15
  - See man page section on PARTITION CONFIGURATION or ***doc/partition-config.txt*** for syntax

IB Partitioning - Membership / security mechanism

- **Partial Membership**
  - Partial members are able to communicate only with *full* members
  - Partial can talk only with full, full can talk with all (either full or partial)
- **Some "Common" Use Cases**
  - IB Storage (File/Block) target: avoid initiator nodes to communicate over the I/O subnet
  - Network management with IPoIB avoiding managed nodes over the managed IP subnet
    Subnet Manager port is a Full member, all other ports are Partial members

Full / partial membership in practice

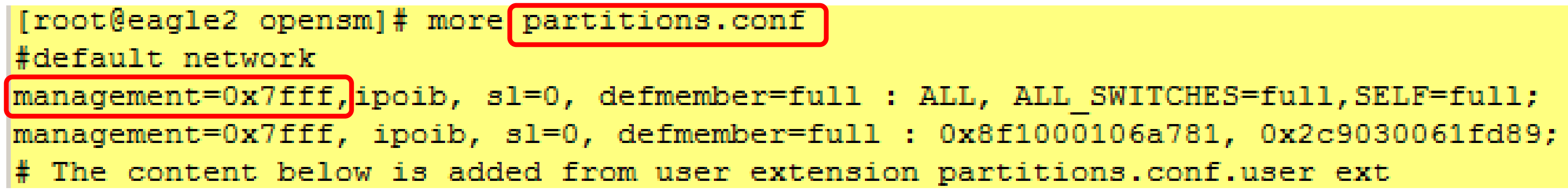
- **PKEY is a 16 bit integer**
  - MSb defines the nature of Membership: is it Full or Partial member ?
  - 15 other bits comprise the partition
- **Example: for the partition 0x0003**
  - for **Full** members: 0x **8**003
  - for **Partial** members: 0x **0**003
- **IPoIB uses full membership**
  - Note: the child interface name has the partition (low 15 bits) but uses full membership

Partitioning – Configuring OpenSM

- **OpenSM enables the configuration of partitions (PKeys) in an InfiniBand fabric.**
- **By default, OpenSM searches for the partitions configuration file under the name *[/usr/etc/opensm/partitions.conf]()***
- **To change this filename, you can use *opensm* with the '--Pconfig' or '-P' flags.**

Partitioning – Configuring OpenSM

- The default partition is created by OpenSM unconditionally even when a partition configuration file does not exist or cannot be accessed
- The default partition has a P_Key value of *0x7fff*
(The partition range is 0x0001 – 0x7fff)
- The port which “runs” OpenSM is assigned full membership in the default partition. All other end ports are assigned full or partial membership.



```
[root@eagle2 opensm]# more partitions.conf
#default network
management=0x7fff,ipoib, sl=0, defmember=full : ALL, ALL_SWITCHES=full,SELF=full;
management=0x7fff, ipoib, sl=0, defmember=full : 0x8f1000106a781, 0x2c9030061fd89;
# The content below is added from user extension partitions.conf.user ext
```

Partitioning – Configuring OpenSM

File format (partitions.conf):

<Partition Definition>:<PortGUIDs list> ;

<Partition Definition>

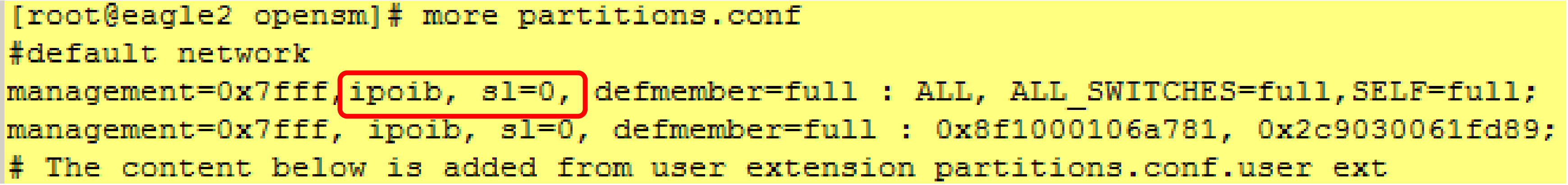
[PartitionName][=PKey] [,flag[=value]] [,defmember=full/limited]

PartitionName string, will be used with logging. When omitted empty string will be used.

Pkey P_Key value for this partition. Only low 15 bits will be used. When omitted will be autogenerated.

flag used to indicate IPoIB capability of this partition.

Defmember Specifies default membership for port GUID list (Default is limited).



```
[root@eagle2 opensm]# more partitions.conf
#default network
management=0x7fff,ipoib, sl=0, defmember=full : ALL, ALL_SWITCHES=full,SELF=full;
management=0x7fff, ipoib, sl=0, defmember=full : 0x8f1000106a781, 0x2c9030061fd89;
# The content below is added from user extension partitions.conf.user ext
```

Partitioning – Configuring OpenSM

Optional Flags cont

ipoib - indicates that this partition may be used for IPoIB, as result IPoIB capable MC group will be created.

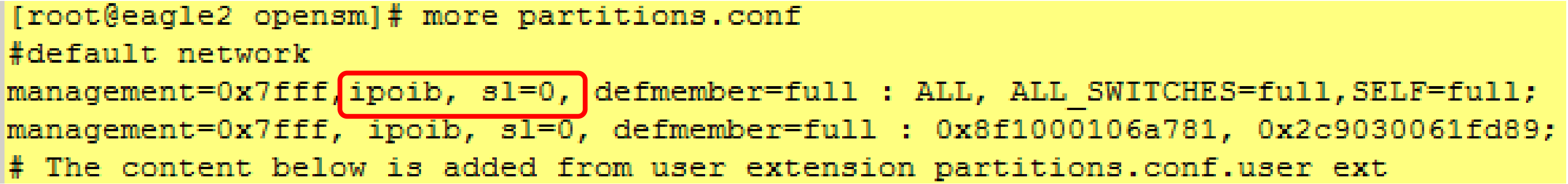
rate=<val> - specifies rate for this IPoIB MC group (default is 3 (10GBps))

mtu=<val> - specifies MTU for this IPoIB MC group (default is 4 (2048))

sl=<val> - specifies SL for this IPoIB MC group (default is 0)

scope=<val> - specifies scope for this IPoIB MC group (default is 2 (link local)). Multiple scope settings are permitted for a partition.

- **Note: Values for rate, mtu, and scope should be specified as defined in the IBTA specification (for example, mtu=4 for 2048)**

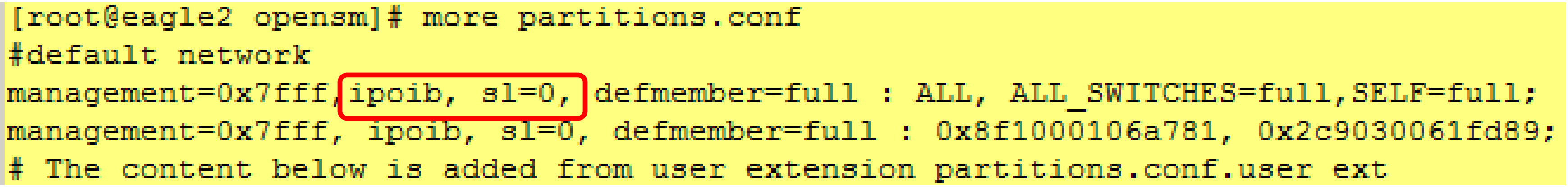


```
[root@eagle2 opensm]# more partitions.conf
#default network
management=0x7fff,ipoib, sl=0, defmember=full : ALL, ALL_SWITCHES=full,SELF=full;
management=0x7fff, ipoib, sl=0, defmember=full : 0x8f1000106a781, 0x2c9030061fd89;
# The content below is added from user extension partitions.conf.user ext
```

Partitioning – Configuring OpenSM

- <PortGUIDs list>
- PortGUID - GUID of partition member end port. Hexadecimal numbers should start with 0x, decimal numbers are accepted too
  Full or limited - Indicates full or limited membership for this port. When omitted (or unrecognized) *limited* membership is assumed
- There are two useful keywords for PortGUID definition:
- - 'ALL' means all end ports in this subnet
  - 'ALL_CAS' means all Channel Adapter end ports in this subnet
  - 'ALL_SWITCHES' means all Switch end ports in this subnet
  - 'ALL_ROUTERS' means all Router end ports in this subnet
  - 'SELF' means the subnet manager port

- Empty list means no ports in this partition



```
[root@eagle2 opensm]# more partitions.conf
#default network
management=0x7fff,ipoib, sl=0, defmember=full : ALL, ALL_SWITCHES=full,SELF=full;
management=0x7fff, ipoib, sl=0, defmember=full : 0x8f1000106a781, 0x2c9030061fd89;
# The content below is added from user extension partitions.conf.user ext
```

Quality of Service (QoS) in OpenSM

- **When Quality of Service (QoS) in OpenSM is enabled (using the '-Q' or '--qos' flags or qos TRUE option), OpenSM looks for a QoS Policy file under */usr/local/etc/opensm/qos-policy.conf***
- **During fabric initialization and at every heavy sweep, OpenSM parses the QoS policy, applies its settings to the discovered fabric elements, and enforces the provided policy on SA client path requests**
- **The overall flow for SA client path requests is as follows:**
  - The SA request is matched against the defined matching rules such that the QoS Level definition is found
  - Given the QoS Level, a path(s) search is performed with the given restrictions imposed by that level

There are two ways to define QoS policy:

- **Simple**
- **Advanced policy file syntax**
  - Provides the administrator various ways to match a PathRecord/MultiPathRecord (PR/MPR) request
  - Provides the administrator QoS constraints on the requested PR/MPR
  - Enables the administrator to match PR/MPR requests by various ULPs and applications running on top of these ULPs

Quality of Service (QoS) Types

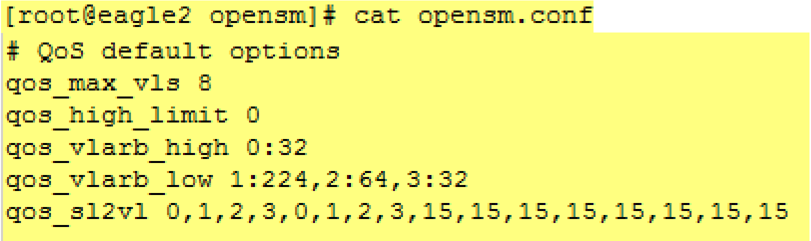
- **Primitive (Simple)**
  - See ***doc/qos-config.txt***
- **Policy (Advanced)**
  - Per QoS Annex A13
    - – See ***doc/QoS_management_in_OpenSM.txt***

Simple QoS Configuration

- **In opensm configuration/options file**
- **Provides for coarse configuration of SL2VL Mapping and VLArbitration tables**
  - Max VLs, VL High Limit, VLArb High/Low, and SL2VL configuration
  - Options for All ports, CA ports, Router ports, Switch port 0 ports, Switch external ports

Simple QoS Configuration Examples

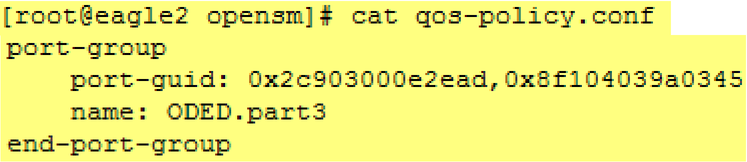
- qos TRUE
- qos_max_vls 8
- qos_high_limit 4
- qos_vlarb_high 0:64,1:64,2:64,3:64,4:64,5:64,6:64,7:64
- qos_vlarb_low 0:4,1:4,2:4,3:4,4:4,5:4,6:4,7:4
- qos_sl2vl 0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7



```
[root@eagle2 opensm]# cat opensm.conf
# QoS default options
qos_max_vls 8
qos_high_limit 0
qos_vlarb_high 0:32
qos_vlarb_low 1:224,2:64,3:32
qos_sl2vl 0,1,2,3,0,1,2,3,15,15,15,15,15,15,15,15
```

Advanced QOS Policy File Syntax

- **The QoS policy file has the following sections:**

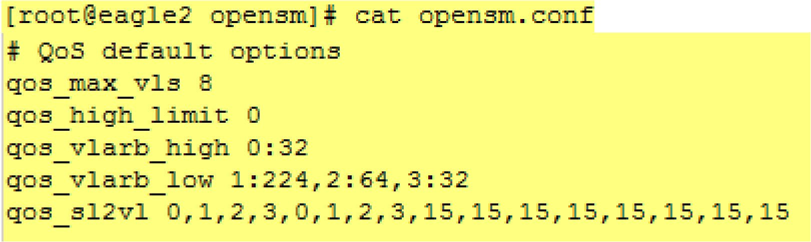


```
[root@eagle2 opensm]# cat qos-policy.conf
port-group
     port-guid: 0x2c903000e2ead,0x8f104039a0345
     name: ODED.part3
end-port-group
```

1. **Port Groups (denoted by port-groups)**
   This section defines zero or more port groups that can be referred later by matching rules (see below). Port group lists ports by:
   a. **Port GUID**
   b. **Port name** - is a combination of Node Description and IB port number
   c. **Pkey** - means that all the ports in the subnet that belong to partition with a given PKey belong to this port group
   d. **Partition name** - means that all the ports in the subnet that belong to partition with a given name belong to this port group
   e. **Node type** - supported node types are: CA, SWITCH, ROUTER, ALL, and SELF (SM port).

2. **QoS Setup (denoted by qos-setup)**
   This section describes how to set up SL2VL and VL Arbitration tables on various nodes in the fabric. However, this is not supported.

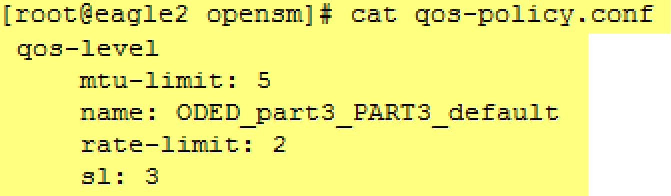
   SL2VL and VLArb tables should be configured in the OpenSM options file
   (default location - *_/var/cache/opensm/opensm.conf_*)



```
[root@eagle2 opensm]# cat opensm.conf
# QoS default options
qos_max_vls 8
qos_high_limit 0
qos_vlarb_high 0:32
qos_vlarb_low 1:224,2:64,3:32
qos_sl2vl 0,1,2,3,0,1,2,3,15,15,15,15,15,15,15,15
```

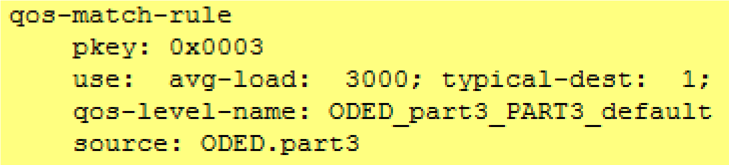
Advanced QOS Policy File Syntax

3. **QoS Levels (denoted by qos-levels)**
   Each QoS Level defines Service Level (SL) and a few optional fields:
   a. MTU limit
   b. Rate limit
   c. PKey
   d. Packet lifetime



```
[root@eagle2 opensm]# cat qos-policy.conf
 qos-level
     mtu-limit: 5
     name: ODED_part3_PART3_default
     rate-limit: 2
     sl: 3
```

4. **When a path search is performed, it is done with regards to restriction that these QoS Level parameters impose. One QoS level that is mandatory to define is a DEFAULT QoS level. It is applied to a PR/MPR query that does not match any existing match rule. Similar to any other QoS Level, it can also be explicitly referred by any match rule.**

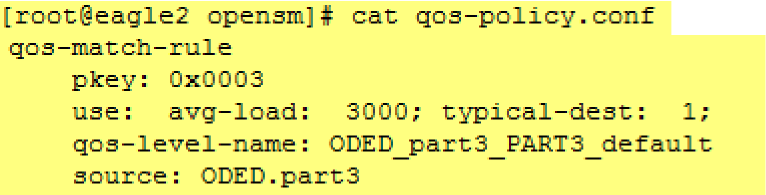


```
qos-match-rule
     pkey: 0x0003
     use:  avg-load:  3000; typical-dest:  1;
     qos-level-name: ODED_part3_PART3_default
     source: ODED.part3
```

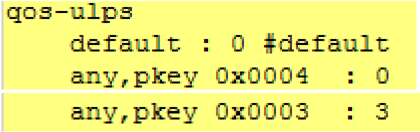
Advanced QOS Policy File Syntax

4. **QoS Matching Rules (denoted by qos-match-rules)**
   Each PathRecord/MultiPathRecord query that OpenSM receives is matched against the set of matching rules.
   Rules are scanned in order of appearance in the QoS policy file such as the first match takes precedence.
   Each rule has a name of QOS level that will be applied to the matching query.

5. **A default QOS level is applied to a query that did not match any rule, Queries can be matched by:**
   a. Source port group (whether a source port is a member of a specified group)
   b. Destination port group (same as above, only for destination port)
   c. PKey
   d. QoS class
   e. Service ID



```
[root@eagle2 opensm]# cat qos-policy.conf
qos-match-rule
    pkey: 0x0003
    use:  avg-load:  3000; typical-dest:  1;
    qos-level-name: ODED_part3_PART3_default
    source: ODED.part3
```



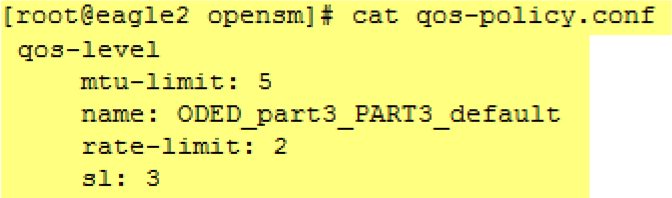
```
qos-ulps
    default  : 0 #default
    any,pkey 0x0004   : 0
    any,pkey 0x0003   : 3
```

Advanced QoS Policy File Syntax

- **QoS policy definition can also consist of single qos-ulps section**
  - Has a list of matching rules and their QoS Level
  - A matching rule has only one criteria
  - Rule goal is to match a certain ULP
  - QOS Level has only one constraint - Service Level (SL)

Advanced QoS Policy File Syntax/Example

- **As mentioned earlier, any section of the policy file is optional, and the only mandatory part of the policy file is a default QoS Level.**

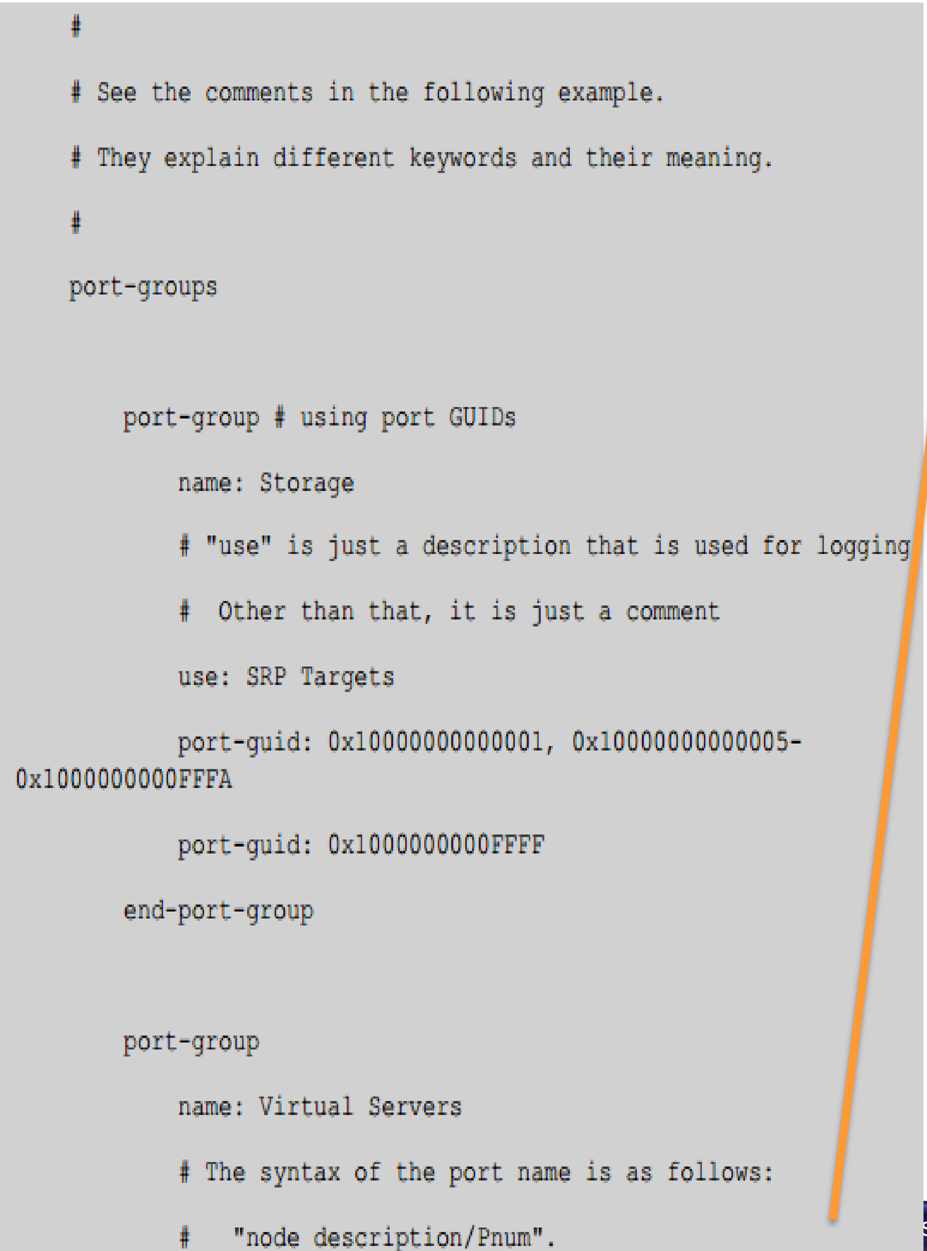


```
[root@eagle2 opensm]# cat qos-policy.conf
 qos-level
     mtu-limit: 5
     name: ODED_part3_PART3_default
     rate-limit: 2
     sl: 3
```

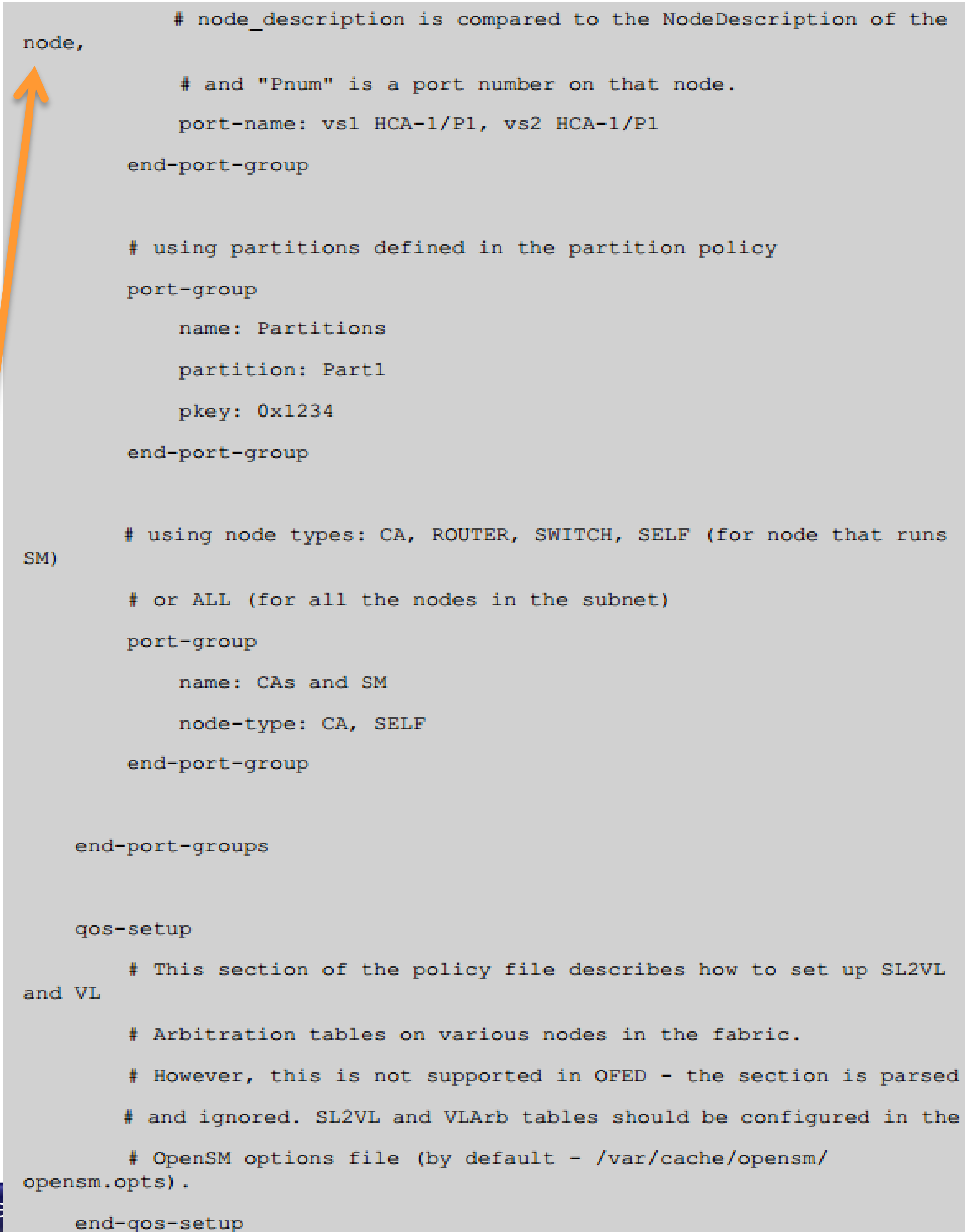
- **Port groups section is missing because there are no match rules, which means that port groups are not referred to anywhere, so there is no need defining them. Also, since this policy file doesn't have any matching rules, PR/MPR query will not match any rule, and OpenSM will enforce default QoS level. Essentially, the above example is equivalent to not having a QoS policy file at all.**

- **The example on the next page shows all the possible options and keywords in the policy file and their syntax.**

Advanced QoS Policy File Example

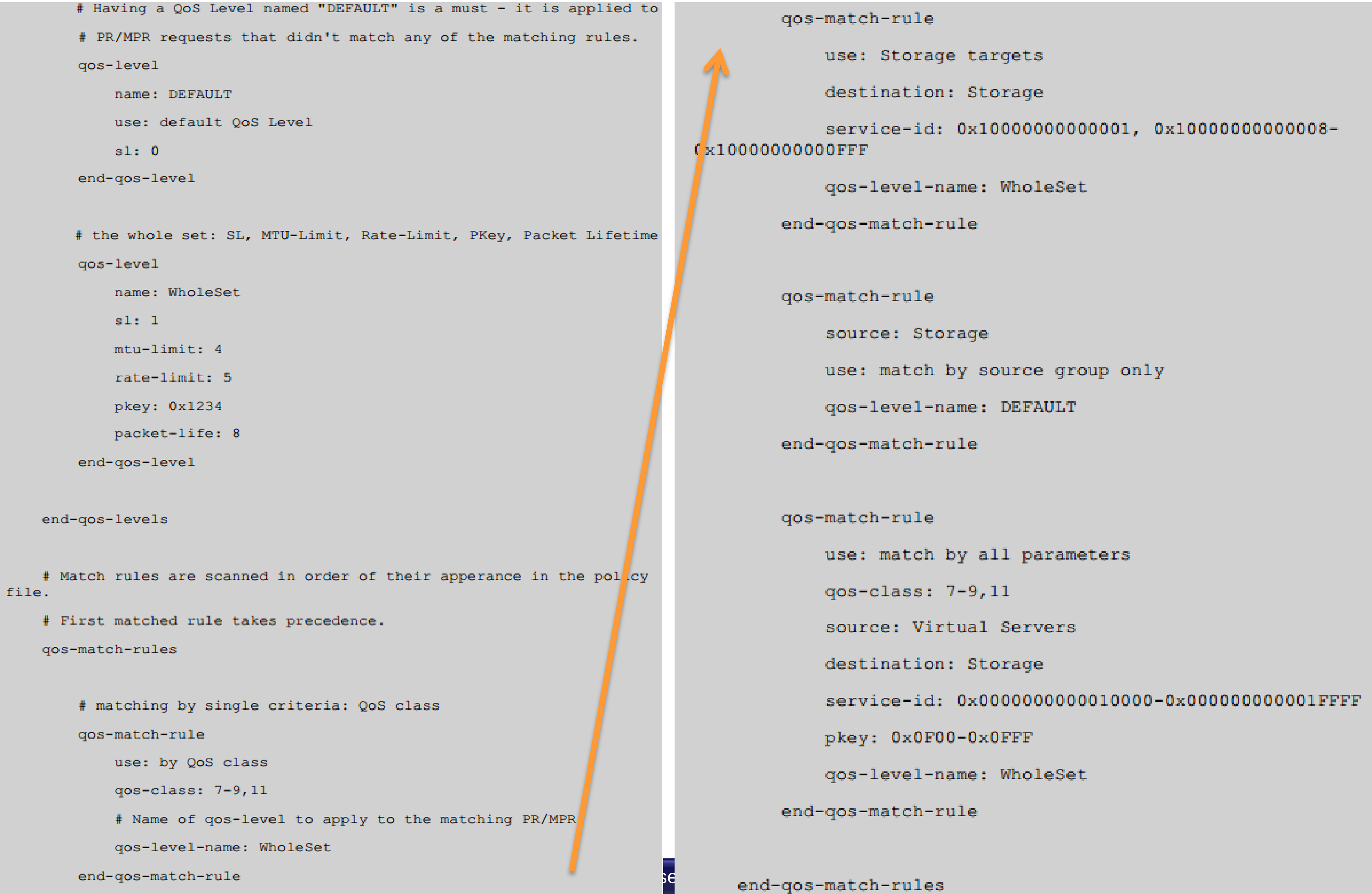


```
    #

    # See the comments in the following example.

    # They explain different keywords and their meaning.

    #

    port-groups


        port-group # using port GUIDs

            name: Storage

            # "use" is just a description that is used for logging

            #  Other than that, it is just a comment

            use: SRP Targets

            port-guid: 0x10000000000001, 0x10000000000005-
0x1000000000FFFA

            port-guid: 0x1000000000FFFF

        end-port-group


        port-group

            name: Virtual Servers

            # The syntax of the port name is as follows:

            #    "node description/Pnum".
```



```
                # node_description is compared to the NodeDescription of the
node,

                # and "Pnum" is a port number on that node.

                port-name: vsl HCA-1/P1, vs2 HCA-1/P1

            end-port-group


            # using partitions defined in the partition policy

            port-group

                name: Partitions

                partition: Part1

                pkey: 0x1234

            end-port-group


            # using node types: CA, ROUTER, SWITCH, SELF (for node that runs
SM)

            # or ALL (for all the nodes in the subnet)

            port-group

                name: CAs and SM

                node-type: CA, SELF

            end-port-group


        end-port-groups


        qos-setup

            # This section of the policy file describes how to set up SL2VL
and VL

            # Arbitration tables on various nodes in the fabric.

            # However, this is not supported in OFED - the section is parsed

            # and ignored. SL2VL and VLArb tables should be configured in the

            # OpenSM options file (by default - /var/cache/opensm/
opensm.opts).

        end-qos-setup
```

Advanced QoS Policy File Example



```
        # Having a QoS Level named "DEFAULT" is a must - it is applied to
        # PR/MPR requests that didn't match any of the matching rules.
        qos-level
            name: DEFAULT
            use: default QoS Level
            sl: 0
        end-qos-level

        # the whole set: SL, MTU-Limit, Rate-Limit, PKey, Packet Lifetime
        qos-level
            name: WholeSet
            sl: 1
            mtu-limit: 4
            rate-limit: 5
            pkey: 0x1234
            packet-life: 8
        end-qos-level

    end-qos-levels

    # Match rules are scanned in order of their apperance in the policy
file.
    # First matched rule takes precedence.
    qos-match-rules

        # matching by single criteria: QoS class
        qos-match-rule
            use: by QoS class
            qos-class: 7-9,11
            # Name of qos-level to apply to the matching PR/MPR
            qos-level-name: WholeSet
        end-qos-match-rule
```

```
        qos-match-rule
            use: Storage targets
            destination: Storage
            service-id: 0x10000000000001, 0x10000000000008-
0x10000000000FFF
            qos-level-name: WholeSet
        end-qos-match-rule

        qos-match-rule
            source: Storage
            use: match by source group only
            qos-level-name: DEFAULT
        end-qos-match-rule

        qos-match-rule
            use: match by all parameters
            qos-class: 7-9,11
            source: Virtual Servers
            destination: Storage
            service-id: 0x0000000000010000-0x000000000001FFFF
            pkey: 0x0F00-0x0FFF
            qos-level-name: WholeSet
        end-qos-match-rule

    end-qos-match-rules
```

Routing

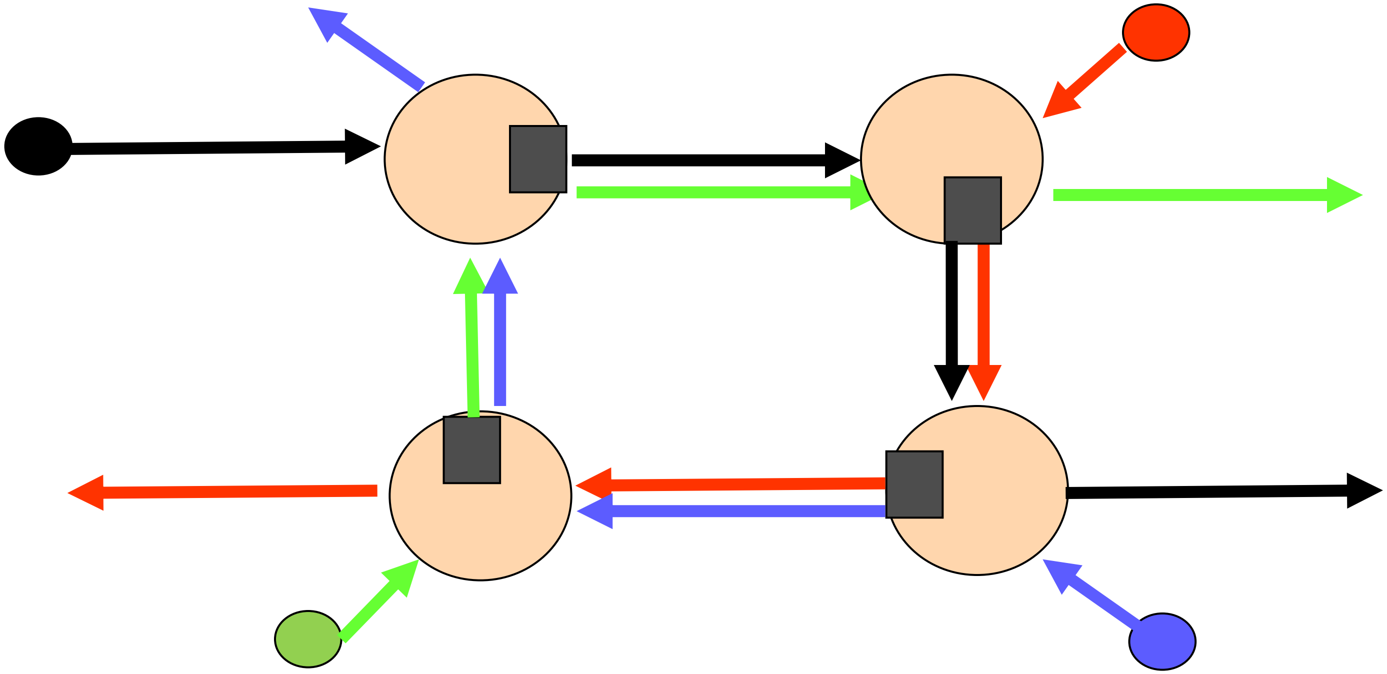
- **Layer 2 but still *routing* term still used**
- **Topology defines physical connectivity**
- **Routing defines paths through topology**
- **Unicast routing**
  - See ROUTING section in opensm man page or ***doc/current-routing.txt***
- **Multicast routing**

IBA Routing

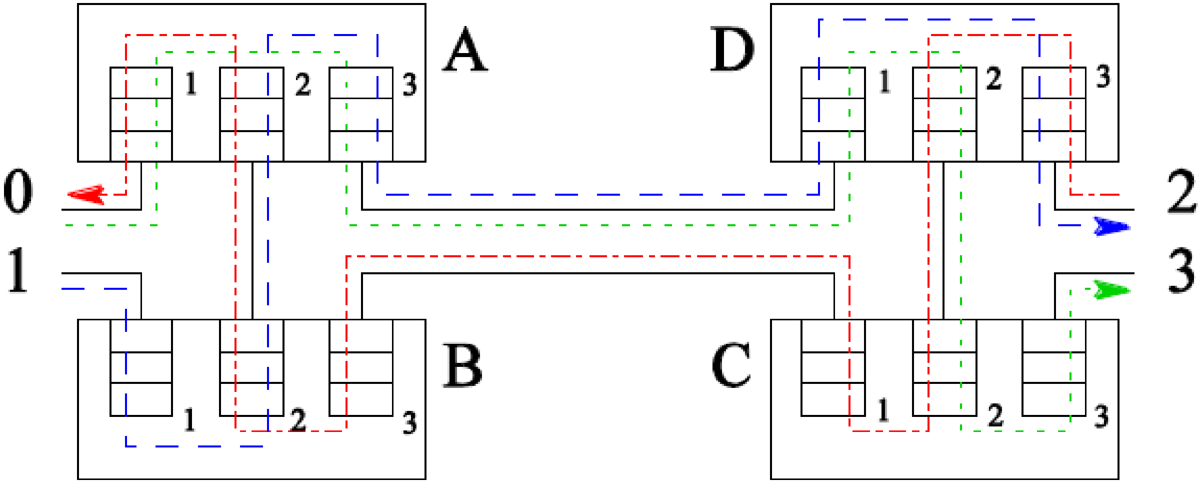
- **IBA mandates deadlock free routing**
- **C14-62.1.2:** When establishing the contents of switch forwarding tables and SL to VL maps, the subnet manager **shall** ensure that no cyclic flow control dependencies exist in the fabric.
- Cyclic dependencies in flow control can cause deadlock and subsequent failure of an IBA fabric. There are a number of routing methods that may be employed to prevent these dependencies. These include pruned and fat tree structures, dimension order routing in meshes and hyper cubes, and use of multiple virtual lanes to break the flow control cycle in routing loops. While IBA does not specify a particular routing method, whatever method is utilized must ensure deadlock-free operation.

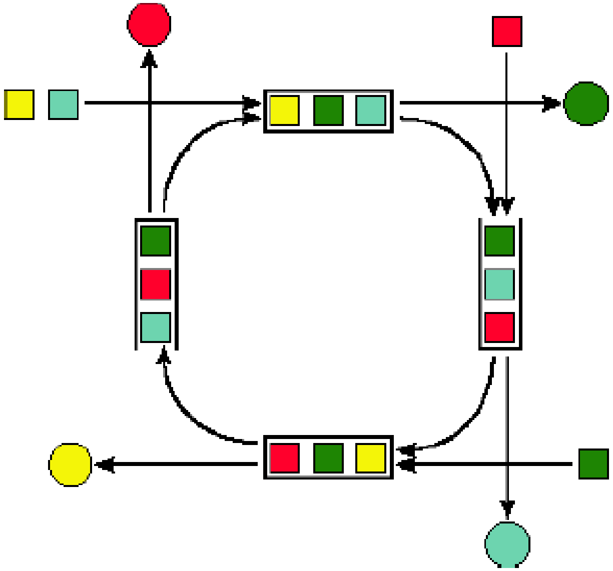
Credit Loops and Deadlock

- “loss less fabric” = “link level flow control” = packet not sent if there is no receive buffer for it
- If traffic to DST-1 waits on traffic for DST-2 which in turn depends on traffic to DST-3 which depends on DST-1 we have a dependency loop and the fabric deadlocks



Credit Loops and Deadlock





Unicast Routing Stages

- **Build LID matrices**
  - For each switch, LID matrix is a table of LIDs, port number, and number of hops (using that port number)
    - SL too for QoS based algorithms
  - If there are several possible output ports (“default” policy):
    - the one with minimum number of routes is chosen
    - if there are several with minimum number of routes, then the one with lowest port number is chosen
- **Build unicast forwarding tables**

Summary: Unicast Engines in OpenSM

	+ Pros	- Cons
MinHop	+any topology +LMC support +Fast	-Not loop-free
UpDown	+any topology +LMC support +loop-free	-works only in topology where there are root or roots are manually set -SM must be a leaf -roots are hot spots -No communication in the spine
FatTree	+optimal performance for fat-tree +loop-free	-only “almost” fat-tree -LMC is not supported
LASH	+any topology (without roots) +outperforms UpDown +loop-free	-VLs are “wasted” -LMC is not supported
DOR	+loop-free	-unbalanced or under-utilized fabric

Unicast Routing Algorithms

- **Min hop**
- **Up/down (MLNX)**
- **Down/up (PNL)**
- **Fat tree (MLNX originated, enhanced by Bull)**
- **DOR (Dimension ordered routing) (SGI)**
  - Meshes & hypercubes

Unicast Routing Algorithms

- **VL Based Routing**
  - LASH/mesh (Simula/SystemFabricWorks)
    - SL changes on topology changes
  - torus2qos for 2D/3D torus (Sandia)
    - SL is constant despite many topology changes
  - SSSP/DFSSSP (Dresden University of Technology)

Unicast Routing Algorithms

- **Other**
  - ucast cache (MLNX)
    - prevents routing recalculation (which is a heavy task in a large cluster) when there was no topology change detected during the heavy sweep, or when the topology change does not require new routing calculation, e.g. when one or more CAs/RTRs/leaf switches going down, or one or more of these nodes coming back after being down.
    - Useful for gpxe installations
  - file save/restore
    - Used for experimentation with routing algorithms
    - Doesn’t handle topology changes

Up/Down Routing

Activation through OpenSM

- Use '-R updn' option (instead of old '-u') to activate the UPDN algorithm.
- Use `-a <guid_list_file>' for adding an UPDN guid file that contains the root nodes for ranking.
- If the `-a' option is not used, OpenSM uses its auto-detect root nodes algorithm.
- Notes on the guid list file:
  - 1. A valid guid file specifies one guid in each line. Lines with an invalid format will be discarded.
  - 2. The user should specify the root switch guids. However, it is also possible to specify CA guids; OpenSM will use the guid of the switch (if it exists) that connects the CA to the subnet as a root node.

Up/Down Routing

Activation through OpenSM

- -X, --guid_routing_order_file <file name>
  - Set the order port guids will be routed for the MinHop and Up/Down routing algorithms to the guids provided in the given file (one to a line).
- -m, --ids_guid_file <path to file>
  - Name of the map file with set of the IDs which will be used by Up/Down routing algorithm instead of node GUIDs (format: <guid> <id> per line).

Up/Down Routing

- **Activation through OpenSM**
  - Other options
    - Scatter ports
      - Use --scatter-ports
        - » This option randomizes port selection in routing.
    - Port shifting
      - Use –port-shifting
        - » This option enables a feature called port shifting. In some fabrics, particularly cluster environments, routes commonly align and congest with other routes due to algorithmically unchanging traffic patterns. This routing option will "shift“ routing around in an attempt to alleviate this problem.

Up/Down Routing

- **Activation through OpenSM**
  - Other options
    - Ucast cache
      - Use –A or --ucast_cache
        - » Prevents routing recalculation in simple cases
        - » This option enables unicast routing cache and prevents routing recalculation (which is a heavy task in a large cluster) when there was no topology change detected during the heavy sweep, or when the topology change does not require new routing calculation, e.g. when one or more CAs/RTRs/leaf switches going down, or one or more of these nodes coming back after being down. A very common case that is handled by the unicast routing cache is host reboot, which otherwise would cause two full routing recalculations: one when the host goes down, and the other when the host comes back online.

Up/Down Routing

Activation through OpenSM

- Other options
  - – Connect roots
    - Use –z or –connect_roots
      - » This option enforces routing engines (up/down and fat-tree) to make connectivity between root switches and in this way to be fully IBA compliant. In many cases this can violate "pure" deadlock free algorithm, so use it carefully.

Multicast Routing

- **Addressing**
  - MGIDs and MLIDs
    - Limited MLIDs supported in switches
      - consolidate_ipv6_snm_req
        - IPv6 Solicited Node Multicast on single MLID
- **Spanning tree per MLID**
  - Root determination
    - First (by GUID order) switch with the minimal maximal distance to all the group members
      - Root can move
- **Routing is triggered by MC joins/leaves in ULPs or applications**
  - IPoIB

SM High Availability

- **Failover/handover**
- **Client reregistration mechanism**
- **Need to be able to migrate the following during OpenSM failover / handover:**
  1. OpenSM configuration
  2. GUID-2-LID assignment
  3. Full Multicast group membership
  4. ServiceRecord registrations
  5. InformInfo registrations
- **Relying on Client ReRegister support by ULPs/application, the list becomes shorter:**
  1. OpenSM configuration
  2. GUID-2-LID assignment
  3. MGID-2-MLID assignment

SM High Availability

- **Current solution is hybrid**
  - Replicating the basic OpenSM and fabric configuration
    - OpenSM configuration
    - GUID-2-LID assignment
  - Replicating all the SA clients' info
    - Full Multicast group membership
    - ServiceRecord registrations
    - InformInfo registrations replicating all the SA clients info
  - Requiring SA Clients to ReRegister on OpenSM handover/failover
    - Replicating doesn't lock SA so need to cover the possible holes

Some Notable Features

- **FDR and FDR-10 support (OpenSM 3.3.11 – Aug 2011)**
  - FDR (and EDR) are IBTA standards
  - FDR-10 is MLNX proprietary
- **SRIOV support (OpenSM 3.3.14 – May 2012)**
  - Additional GUIDs for virtual machines
  - Minor impact on partition manager
- **SA scalability (future)**
  - Distributed SA
  - A step towards Exascale

Windows OpenSM and Diagnostics

- **Ported from Upstream Linux Version**
  - Now based on OpenSM 3.3.13
  - Switched from use of IBAL vendor layer to UMAD vendor layer so consistent with Linux vendor layer
    - libibumad ported to Windows
    - Also, most of infiniband-diags and libibmad ported to Windows
      - Mainly missing shell script based infiniband-diags tools

Performance Manager

- **Fabric discovery**
- **Polls each port's PortCounters periodically**
  - Recently enhanced for optional PortCountersExtended
    - 64 bit counters for data
- **Performs data reduction on counters**
- **Logs performance data and indicates interesting events**
- **See *doc/perf-manager-arch.txt* and *doc/performance-manager-HOWTO.txt***

Congestion Manager

- **Recently added (in OpenSM 3.3.15 released August 2012)**
- **Currently *experimental* status**
  - Default mode is disabled
- **Currently lacking congestion log monitoring and SwitchPortCongestionSetting attribute support**
  - Also, CongestionKeyInfo and trap logging

Thank You

OPENFABRICS ALLIANCE

Mellanox®
TECHNOLOGIES