



NC-SI 1.2 Topics- Work-In-Progress

Version 0.10
September 13, 2017



Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change without notice. The standard specifications remain the normative reference for all information.
- For additional information, see the Distributed Management Task Force (DMTF) website.



Proposed Topics for NC-SI 1.2

- **Versioning**
- **Expanding NC-SI to other types of I/O controllers**
 - Fibre Channel
- **Multiple Partitions/PFs per Port**
- **Bandwidth Allocation**
- **Representation of Host function types (networking, storage) and Traffic Types**
- **Ethernet-based storage**
- **Storage Boot Configuration**
- **SR-IOV**
- **LLDP/DCBx/Energy Efficient Ethernet (EEE)/QoS**
- **Virtual Addressing**
- **Ethernet Flow Control**
- **Additional capability reporting**
- **Enhanced Port Information**
- **Additional AENs – e.g. package AENs**
- **WoL**
 - Awareness of Host state / preOS / boot / OS
- PLDM enhancements
- RBT – MCTP Bridging nvmemi
- Mantis



▪ Versioning

- Some of the proposed changes are most easily addressed by adopting a change in the Control Packet Header
 - The header revision field can support this differentiation
- Other changes are easily handled by just adding commands
- Desire to minimize impact on NCs that don't support advanced features
- Transport agnostic, not just about RBT
- PFs as subset of both channel cmds and pkg cmds?
- Backward compatibility likely important with current users of NC-SI
- Version discovery (command?)
- What is mandatory



▪ **Expanding NC-SI to other types of I/O controllers**

- Need method of tailoring NC-SI to other types of I/O controllers
 - Scope of support
 - Fibre Channel
 - IB
- **NC-SI Control**
 - Expansion of existing controls
 - Addition of New controls
- Pass-through definition on new fabrics subject to further discussion
- Capability reporting
- Command tailoring
 - Context sensitive
 - Numeric vs. bit mapped fields



▪ **Multiple Partitions/PFs per Port**

- Need to distinguish items owned by port and those owned by PFs
- Delink such that channel commands own port-related command/responses and PFs own Function-related command/responses
- Need to understand PF to Port mapping
 - Rules to reduce impact on 'simple' controllers with 1:1 PF to Port mapping
- Definition of a Function 'address'



- **TX Bandwidth Allocation (PF only)**
- Allocation of a Port's TX Bandwidth to all attached PFs
- Default settings
- How do we handle changes in one Function's allocation

- Absolute vs. relative vs. weighted (%)
- Impact on enabling/disabling PFs on existing port BW allocations
 - Who is controlling this OS or sideband
- How to handle excursions above available BW
- How to handle changes in link speed
- May be several different implementations to handle
- How does these settings relate to DCB ETS/TCs
- Both Min and Max BW allocations need to be handled



▪ Representation of Host function types

- Representation of Host function types (networking, storage) and Traffic Types





▪ Ethernet-based storage

- Capability reporting
- Configuration
- Dependency handling wrt other resource users



▪ **Storage Boot Configuration**

- **UEFI only or UEFI and legacy OpROM?**
 - Does any UEFI configuration make sense or should just direct to BIOS?
- **iSCSI HBA mode and iBFT mode?**
- **IPv4 & IPv6**
- **iSCSI, FCoE, and FC**
- **What should be done to check for errors in boot config?**
- **Boot timing is a question**
- **Do boot failures get reported back? How?**
- **How should persistency of settings be handled**
 - Discover persistency settings/capabilities
 - Persistency attribute for settings
- **Impact on BIOS handling UEFI boot (BIOS owns parameters, not controller)**



▪ SR-IOV

▪ Support of VFs

- What do we want to do with VFs??
- Query only?
- Controller may not know of VFs/VMs hiding behind software switch – default queues w/ permissive mode

▪ Addressing of VFs

▪ Advertisement of VFs

- Allocation of VFs from resource pool
- What types of topologies are supported, granularities

▪ Enablement of VFs

▪ How should we deal with e-switches?

- Not always easy for FW to know
- Move to partitioning/NPar, don't deal with vSwitches

Do VFs just share in the parent's BW allocation?

Profile for VF resource allocation & properties is it possible to standardize diverse architectures?



▪ **LLDP/DCBx/Energy Efficient Ethernet (EEE)/QoS**

- LLDP is the advertisement/negotiation mechanism between the port and its link partner
- Incoming LLDP packets need to simply be duplicated and sent to FW and host
- Outgoing LLDP packets more of a problem
 - Extension of SetLink, easy for pre-OS
- What is the relationship of the MC to the OS in handling LLDP TLVs
- Persistent storage of some TLVs
 - Chassis information?
 - Passing of information to host if LLDP ownership is transitioned to host



Virtual Addressing

- Timing of Host Virtual Address application and removal
- Persistence
 - Situations where the VA is the only MAC
- Number and perhaps distribution of MACs
- How to handle MAC provisioning of the BMC
 - In a multi-host environment



▪ Ethernet Flow Control

- This is port-level 802.3x pause frame-based flow control
- This is NOT PFC
- Why - Ownership of port's flow control unclear when multiple PFs are instantiated. OS driver typically controls 802.3x
- TX, RX, TX & RX
- May want to modify link when host driver(s) is up
- Possible FC ownership by LLDP
- How do we handle things that are not known to controller FW?
 - Jumbo frames/buffer sizes/etc.



■ Additional capability reporting

- Additional capability reporting
 - PCIe link margining/reporting
- Devices other than Ethernet
 - FC – just focus on controller, not the fabric and beyond
 - IB – Yuval
 - Storage – probably not
- How to discover/advertise ‘optional’ features
 - PLDM model = Get Supported Commands
 - Extend existing NC-SI Capability command
 - Eliel’s Mantis proposal
 - OEM commands



▪ Enhanced Port Information

- Enhanced Port Information
- Get Link extensibility
- How to handle new fabrics
- 25GbE autoneg
- Focus on link establishment and not beyond



▪ Additional AENs

- AENs for packages
 - Init issues (impacting/including AEN enablement)
 - Health
 - Temperature
 - Need for reset (also a cmd to initiate reset)
 - Controlled entry into Initial State (notification to MC)
 - Estimated 'offline time'
- AENs for PFs/VFs?
- Boot process? Not much visibility in UEFI environment, so leave in host?



▪ WoL

- OOB WoL (Magic Packet only) port-based, non-virtual address?
 - What MAC is guaranteed?
 - Permanent or alt MAC applied persistently in a prior event
 - Port based
- WoL (Magic Packet)
 - Permanent or alt MAC applied persistently in a prior event
 - Applies to all assigned MACs on port
- ACPI pattern- based (Get only?)
 - Needed? Contention with host (handle unknown vs. none set)
 - MAC requirements – Permanent and Virtual?
 - Subset of interesting packet definitions?



▪ Awareness of Host State

- Pre-OS
- Boot
- OS
- Arbitration of configuration responsibilities (host vs. BMC)



- **PLDM enhancements**

- OOB





▪ RBT – MCTP Bridging

- OOB





▪ Mantis





Brainstorming

Multi-host - including boot

Single BMC

Single BMC – multiple logical BMC

Multiple BMC

peer-peer vs. hierarchy

NC-SI over Ethernet fabric

changes to PCI CEM?



BACKUP





Topics

- Break up of spec
- Shared NC, Multiple MCs
- Network Partitions
- SR-IOV
- Host Configuration
- LLDP/DCBx/Energy Efficient Ethernet (EEE)/QoS
- Optics information for Ethernet NC
- NC diagnostics – can be covered as PLDM for diagnostics
- Lower voltage for RMII
- Additional physical interfaces for sideband communications
- Additional AENs – e.g. package AENs
- Representation of Host function types (networking, HPC, storage) and Traffic Types
- Ethernet based storage
- Expanding NC-SI to other types of I/O controllers
- RDMA
- Command format
- Versioning
- Candidate topics for NC-SI 1.2