DPU Sideband Management

May 2024

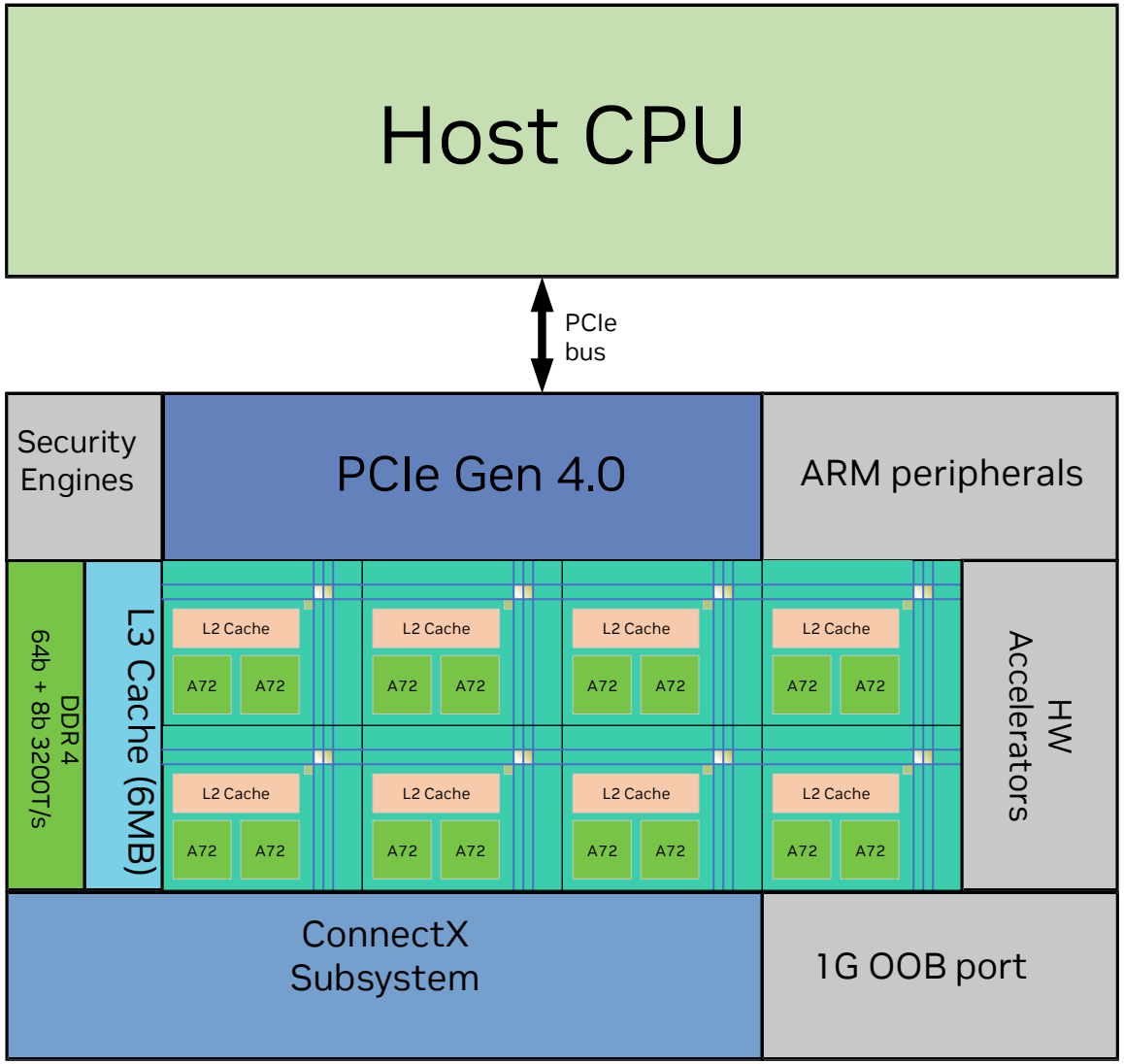
Agenda

- DPU overview
- DPU Terminology & Management Interface
- PXE
- RSHIM
- NCSI Commands & DPU Boot Sequence
- Multiple DPU Management
- DPU BMC

DPU Overview

DPU OVERVIEW

DPU context in a server system



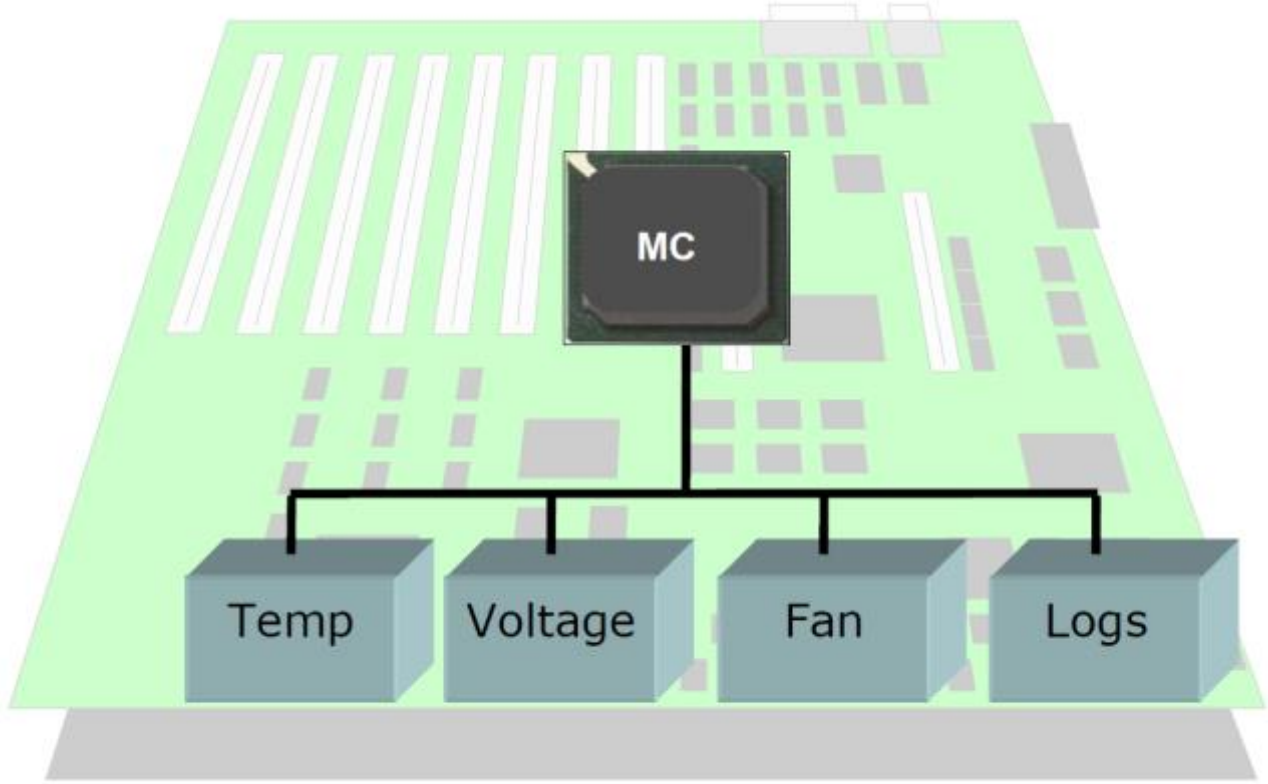
Platform BMC

NIC BMC

SIDEBAND MANAGEMENT

Host Management

- BMC – Board Management Controller
- What does it control:
  - Monitoring
  - SW/FW versions
  - Boot protocol
  - Recovery
  - Generic non-OS configuration
- Who controls the BMC:
  - Chassis Management
  - OEM specific code
  - Remote management tool

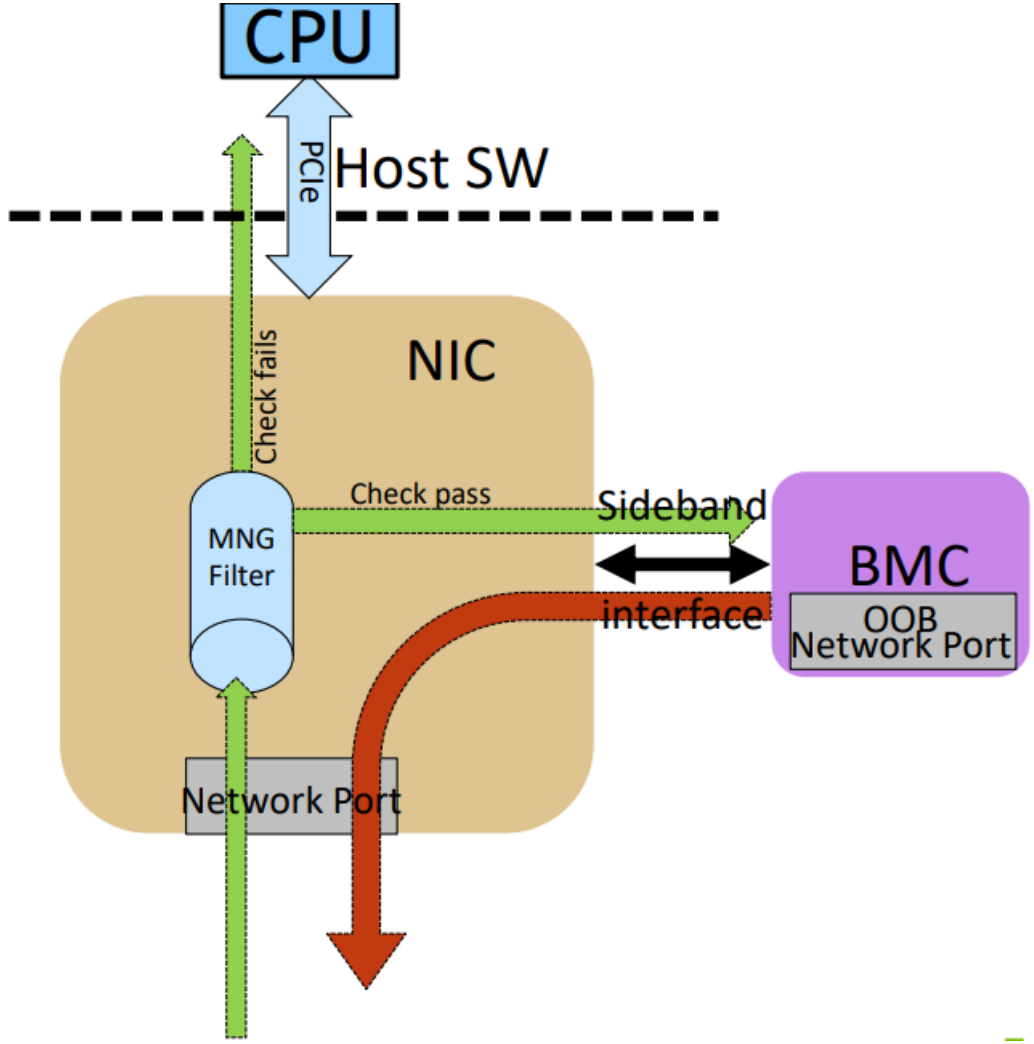


DPU Sideband Terminology & Management Interface

TERMINOLOGY

Terminology

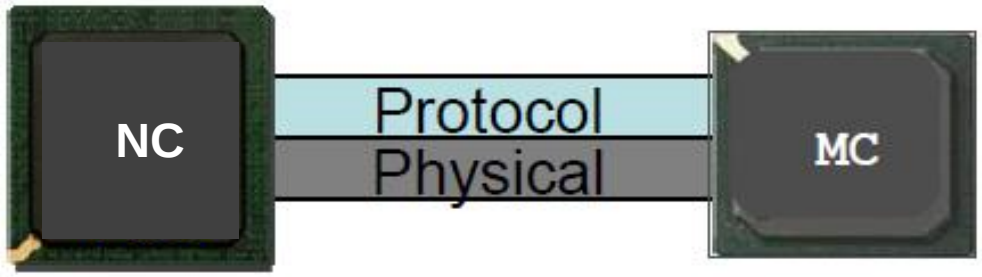
- In-Band
  - An operation that is performed through the primary interface. This includes mainly the PCIe interface from the host side and the network ports for network-based operations
- Sideband
  - A non-primary interface such as I2C, RBT, UART or USB. MCTP over PCIe is also considered a sideband interface.
- Out of band
  - An operation that is performed through a non-primary interface. This includes the OOB management network as well as using any of the sideband interfaces
- Passthrough management traffic
  - A BMC management network traffic that is using the network port for the connection to the data network and connects the BMC using any of the sideband interfaces



SIDEBAND PROTOCOL

Sideband Protocol

- The Physical Layer- The physical electrical connection between the NC and the MC
  - Electrical connection between NC and BMC
  - Electrical and timing differ between sidebands
  - Example of Physical layer options –
    - **SMBus**, **RMII, PCIe**
- The Protocol Layer - The agreed upon communication protocol between the NC and the MC
  - Agreed upon “language” between NC and BMC
  - Defines the Ethernet packet encapsulation
  - Defines the configuration command format
  - Example of Protocol layer options –
    - **MCTP**, **NC-SI, PLDM, IPMI**



PHYSICAL LAYER

Management Interfaces for NIC

- SMBus
  - 2 wire interconnect – Data and Clock
  - Bi-directional traffic
  - Based on block wire operations
  - MCTP over SMBus - DMTF DSP0237 spec
- RBT (RMII Based Transport)
  - 100Mbps Ethernet connection
  - Support flow control
  - NCSI over RBT - DMTF DSP0222 spec
- PCIe
  - PCIe VDM Transport
  - MCTP over PCIe - DMTF DSP0238 spec

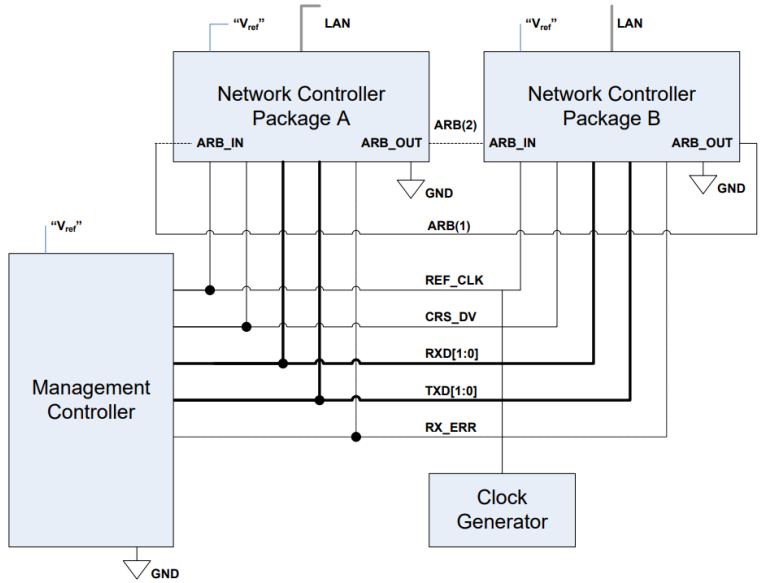


Figure 16 – Example NC-SI signal interconnect topology

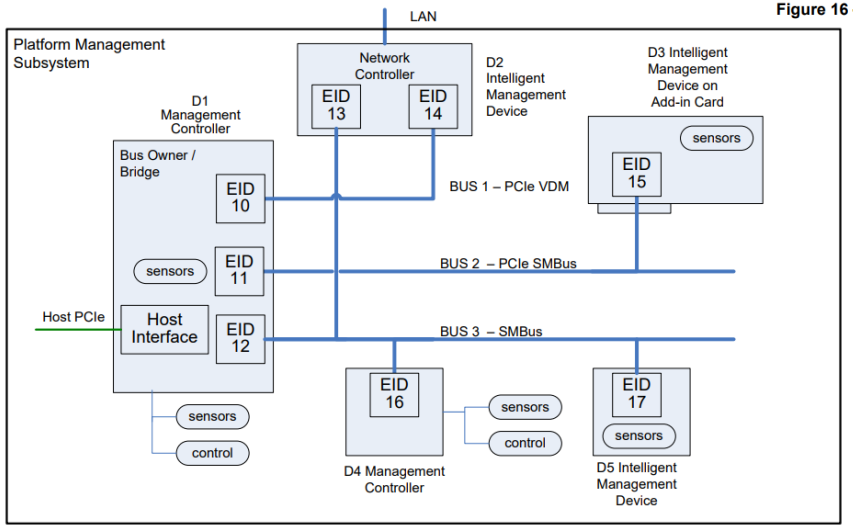
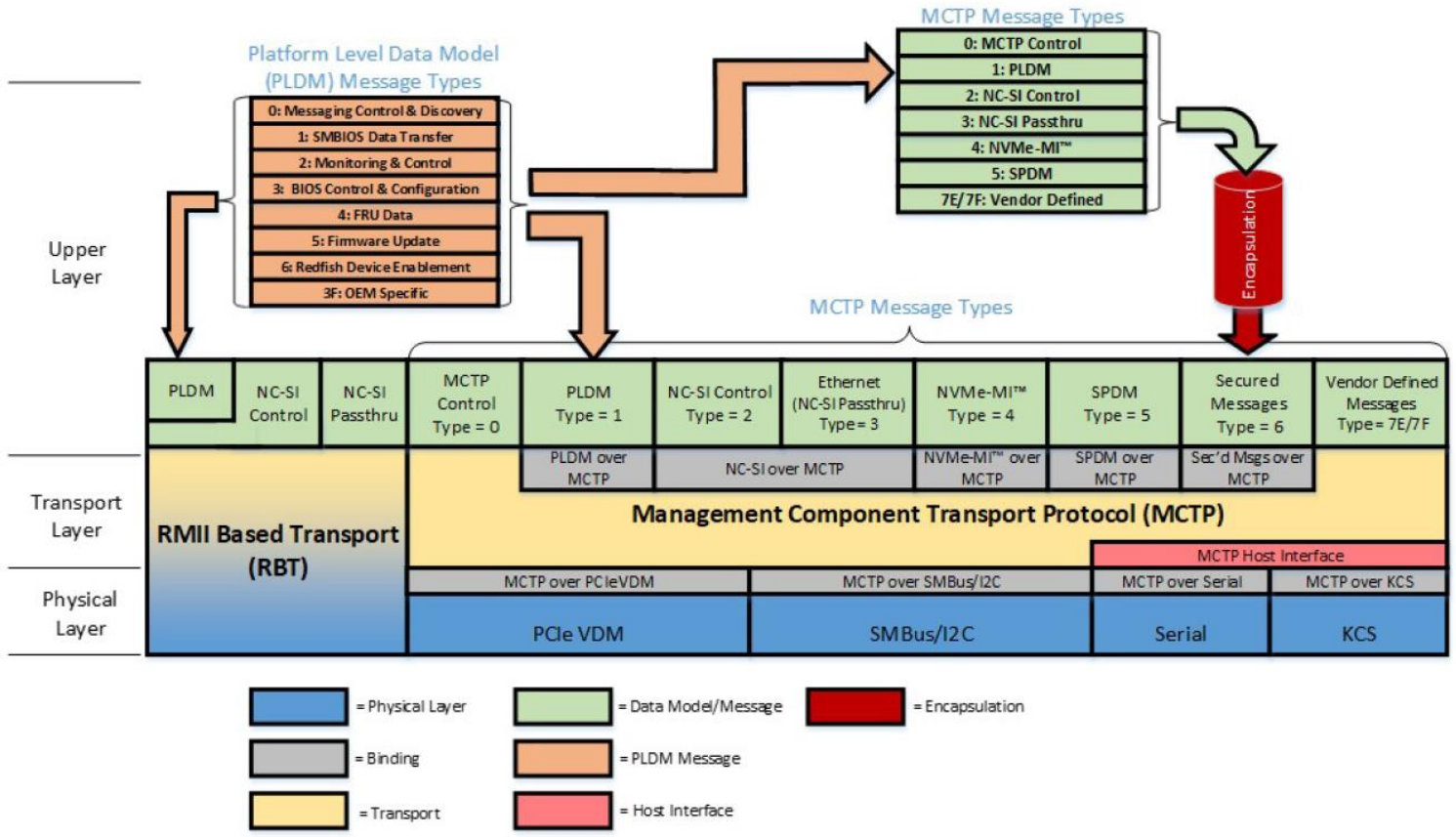


Figure 3 – MCTP Topology

PROTOCOL LAYER

MCTP - Management Component Transport Protocol

- Support communications between different intelligent hardware components.
- Providing protocol for monitoring and control.
- Independent of the underlying physical bus properties.



DPU MANAGEMENT SETTINGS

ARM subsystem management methods

- Security settings – e.g. UEFI secure boot enablement, keys-database update/reset
- Boot settings – e.g. boot sources enablement, boot order
- OS settings
- OS/FW Update
- Network parameters configuration
- Expose/hide rshim interface
- CPU & OS monitoring
  - Health
  - Boot progress
  - Sensors (Temperature, Clock-speed….)

NIC management methods

- BMC interface configuration
- BMC Trust establishment
- Network ports settings
- BMC to ARM communication
- FW Update
- Monitoring and control (Temp, power, link speed, Health, ports power/temp)
- ARM OS state
- ARM Reset
- NIC/Full-chip reset

Supported BMC modes

BlueField devices may be managed by:

1. Platform BMC
2. NIC BMC – an onboard BMC that is integrated on the DPU card
3. Both 1 & 2 at the same time

DPU MANAGEMENT INTERFACES

ARM subsystem

- Rshim – an internal module which is used to allow to reset and boot the ARM subsystem. Rshim interface is accessible by:
  - USB
  - PCIe (using a dedicated PF on the PCIe bus)
- OOB port – used for remote management of the ARM cores as well as to enable the ARM to do network boot over the management network
- USB device interface
  - Rshim interface to ARM cores
  - Network interface between the ARM and the BMC
- UART – console interface and event logging during boot time
- I2C – IPMI interface between ARM and the NIC BMC
- PCIe – Rshim interface (when enabled)

MANAGEMENT INTERFACES

NIC subsystem

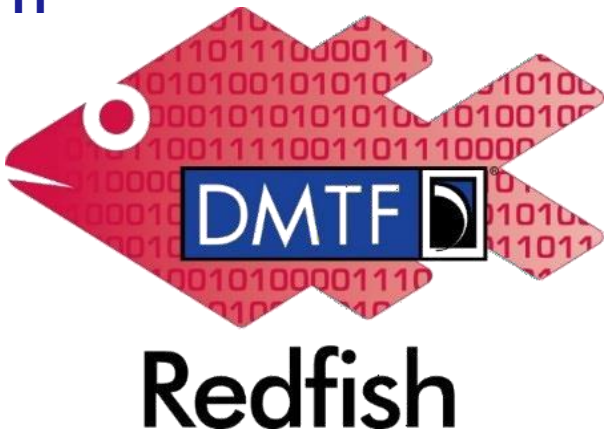
- SMBus – used to connect the NIC to the platform BMC
- RBT – used to connect the NIC to the platform BMC
- PCIe – used for host-based management as well as for MCTP over PCIe

MANAGEMENT PROTOCOLS

- PLDM – a class of management protocols. DPUs use PLDM for
  1. Monitoring and control (PLDM Type 2)
  2. FRU information reading from the device (PLDM Type 4)
  3. FW Update (PLDM Type 5)
- NC-SI basic NIC management, including
  1. BMC configuration and monitoring of NIC parameters
     - OEM extensions are used as a method to provide extra functionality to manage the DPUs
  2. BMC communication through the NIC ports, using a sideband interface (typically RBT or PCIe)
- IPMI – A legacy management protocol. On the DPU cards it is used between the DPU and the NIC BMC
- Redfish – Restful interface for remote client to enable management of a platform.
- Redfish Host interface – a method to manage a CPU from a BMC using Redfish semantics. The CPU is always a Client and the BMC acts as the server

WHAT IS REDFISH?

- **Industry Standard Software Defined Management for Converged, Hybrid IT**
  - HTTPS in JSON format based on OData v4
  - Schema-backed but human-readable
  - Equally usable by Apps, GUIs and Scripts
  - Extensible, Secure, Interoperable
- **Version 1 focused on Servers**
  - A secure, multi-node capable replacement for IPMI-over-LAN
  - Represent full server category: Rackmount, Blades, HPC, Racks, Future
  - Intended to meet OCP Remote Machine Management requirement
- **Expand scope over time to rest of IT infrastructure**
  - Additional features coming out approximately every 4 months



SYSTEM LEVEL CONSIDERATIONS (1/2)

Dependency between platform state and DPU state

- A DPU is a complex device which operates within a server
  - Being an add-on device mandates the DPU to be ready for service when the system starts
  - Being a server that loads a full operating system requires a long time

- Therefore
  - The server must be able to track the progress of the DPU OS loading - this is possible using a Mellanox OEM command "***Get Smart NIC OS State***"

- Similarly, when restarting the embedded ARM OS, the system must be aware and must not fail
  - Generic services (like Virt-IO, NVMe, NVMf) must be suspended for the duration of the OS restart
    - The OS must be aware of the ARM reset event which can be done by:
      - Hot-removal of associated PFs
      - Using PCIe events like AER/DPC can be used instead of hot-removal
      - If there is no way to allow in-service ARM OS reset, the ARM subsystem must wait for the next host system boot to reset
- Before the host shuts down it must assure that the ARM OS gracefully shuts down (can be done using Mellanox OEM command "***Get Smart NIC OS State***")
  - Failing to comply with this requirement can potentially corrupt the ARM OS file system on the DPU card
  - Signaling the DPU to request graceful shutdown mandates platform customization by the platform developer

- All the DPU specific functions are available using NC-SI Mellanox OEM command

SYSTEM LEVEL CONSIDERATIONS (2/2)

Dependency between ARM state and NIC state

- The NIC is the owner of the PCIe interface (including the PCIe switch)
  - Whenever the NIC resets, the PCIe connection will be affected, and the ARM cannot use the PCIe fabric during this time
- The ARM is the owner of the on-board DRAM
  - Whenever the ARM subsystem resets, the content of the DRAM is lost. Therefore, if the NIC contexts memory (ICM) uses the ARM DRAM, once the ARM is reset, the host communication through the NIC will be affected by the loss of the ICM memory

PROVISIONING BFB IMAGE

- A new BFB image can be pushed to the DPU through rshim or to be fetched by the DPU as part of an HTTPS/PXE boot
- When the DPU is operational, exposing/hiding rshim from the external host is configurable
  - Can be performed from a trusted BMC using Mellanox OEM command
  - Can be performed from ARM OS using Mellanox tools
  - Can be configured via Redfish host interface from the platform/NIC BMC
- When the DPU is in recovery mode, rshim is exposed on both USB and PCIe to allow device recovery
  - The DPU is in recovery mode when both ARM and the NIC are operating in recovery mode

PROVISIONING NIC FW

- When the DPU is fully operational a new FW image can be pushed to the NIC from ARM/x86 OS or to be programmed as part of the ARM/x86 UEFI boot
- When the DPU is in recovery mode a new FW can be provisioned through PCIe only
  - The DPU is in recovery mode when both ARM and the NIC are operating in recovery mode
- When the NIC is in recovery mode (livefish) and ARM is operational, a new FW can be provisioned either internally by the ARM cores or from the host through PCIe

SUMMARY POINT

1. Use DPU cards with a NIC BMC to simplify the integration to existing platforms
2. Using NVIDIA NC-SI OEM commands & Redfish Host Interface allows the platform BMC to configure the DPU
3. Enabling RBT connection, mandates using harness between NCSI connector and the system
4. Using the OOB port on the DPU enables remote OOB management of the DPU but requires additional 1Gb connection
5. When controlling the DPU from the platform BMC, the NIC BMC should be placed into “field mode”

RSHIM

What is Rshim driver

Interfaces

- A special driver called RShim must be installed and run to expose the various BlueField management interfaces on the host OS.
- Physical channel for Rshim device
  - USB – Note: USB interface on BF3 card is not for rshim
  - PCIe
- When the Rshim driver runs properly on the host side, expose two devices
  - A sysfs device - `/dev/rshim0/*`
  - A virtual ethernet interface - `tmfifo_net0`

```
root@sz-intel-gen5:~# ls /dev/rshim0/
boot  console  misc  rshim
```

```
root@sz-intel-gen5:~# ip a | grep tmfifo
9: tmfifo_net0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
```

- If Multiple DPUs may connect to the same host machine and start rshim driver correctly, each board will have
  - A sysfs device - `/dev/rshim<N>`
  - A virtual ethernet interface - `tmfifo_net<N>`
  - Note, BlueField DPUs arrive with the following factory default configurations for tmfifo_net0.
    - MAC - `00:1a:ca:ff:ff:01`
    - IP - `192.168.100.2`
  - if you are working with more than one DPU, you must change the default MAC and IP addresses.
- If customer wish to block rshim interface in some scenarios
  - Enable Zero-trust mode, rshim interface is blocked.
  - Modes of Operation - NVIDIA Docs

Functions of Rshim Interfaces

Functions

- Sysfs device
  - Rshim console - /dev/rshim0/console
    - # `sudo screen /dev/rshim0/console 115200`
  - Rshim log buffer - /dev/rshim0/misc, display different level
    - e.g.
    - `# echo "DISPLAY_LEVEL 1" > /dev/rshim0/misc`
    - `# cat /dev/rshim0/misc`
  - Rshim boot device that load BFB image - /dev/rshim0/boot
    - `# cat <BlueField-OS>.bfb > /dev/rshim0/boot`
    - `# bfb-install --bfb <BlueField-OS>.bfb --config bf.cfg --rshim rshim0`

```
root@sz-intel-gen5:~# echo "DISPLAY_LEVEL 0" > /dev/rshim0/misc &&  cat /dev/rshim0/misc
DISPLAY_LEVEL   0 (0:basic, 1:advanced, 2:log)
BOOT_MODE       1 (0:rshim, 1:emmc, 2:emmc-boot-swap)
BOOT_TIMEOUT    150 (seconds)
DROP_MODE       0 (0:normal, 1:drop)
SW_RESET        0 (1: reset)
DEV_NAME        pcie-0000:03:00.3
DEV_INFO        BlueField-3(Rev 1)
OPN_STR         N/A
UP_TIME         220(s)
root@sz-intel-gen5:~# echo "DISPLAY_LEVEL 1" > /dev/rshim0/misc &&  cat /dev/rshim0/misc
DISPLAY_LEVEL   1 (0:basic, 1:advanced, 2:log)
BOOT_MODE       1 (0:rshim, 1:emmc, 2:emmc-boot-swap)
BOOT_TIMEOUT    150 (seconds)
DROP_MODE       0 (0:normal, 1:drop)
SW_RESET        0 (1: reset)
DEV_NAME        pcie-0000:03:00.3
DEV_INFO        BlueField-3(Rev 1)
OPN_STR         N/A
UP_TIME         224(s)
BOOT_RESET_SKIP 0 (1: skip)
PEER_MAC        00:00:00:00:00:00 (rw)
PXE_ID          0x00000000 (rw)
VLAN_ID         0 0 (rw)
```

- Virtual ethernet device
  - peer-to-peer tunnel connection between the host and the DPU OS
  - The DPU OS also configures a similar device
  - DPU OS's BFB images are customized to configure the DPU side of this connection with a preset IP of 192.168.100.2/30

```
root@sz-intel-gen5:~# ip a | grep tmfifo_net0
9: tmfifo_net0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
root@sz-intel-gen5:~# ip addr add dev tmfifo_net0 192.168.100.1/30
root@sz-intel-gen5:~# ping 192.168.100.2
```

Install Rshim Driver on Host

Install

- Deploying BlueField Software Using BFB from Host - NVIDIA Docs
- Verify rshim devices before install driver
  - Run “lspci” or “lsusb”

```
root@sz-intel-gen5:~# lspci | grep -i mell | grep -i management
03:00.3 DMA controller: Mellanox Technologies MT43244 BlueField-3 SoC Management Interface (rev 01)
13:00.2 DMA controller: Mellanox Technologies MT43244 BlueField-3 SoC Management Interface (rev 01)
```

- Install rshim driver by doca-runtime package
  - `host# sudo rpm -Uvh doca-host-repo-rhel<version>.x86_64.rpm`
  - `host# sudo yum makecache`
  - `host# sudo yum install doca-runtime`
- Start rshim service
  - `host# sudo systemctl enable rshim`
  - `host# sudo systemctl start rshim`
  - `host# sudo systemctl status rshim`
  - `host# sudo systemctl restart rshim`

Rshim Troubleshooting and How-Tos

Troubleshooting

- RShim Troubleshooting and How-Tos - NVIDIA Docs

Another backend already attached

RShim driver not loading

- RShim driver not loading on DPU with integrated BMC
  - RShim driver not loading on host
  - RShim driver not loading on BMC
- RShim driver not loading on host on DPU without integrated BMC

Change ownership of RShim from NIC BMC to host

How to support multiple DPUs on the host

BFB installation monitoring

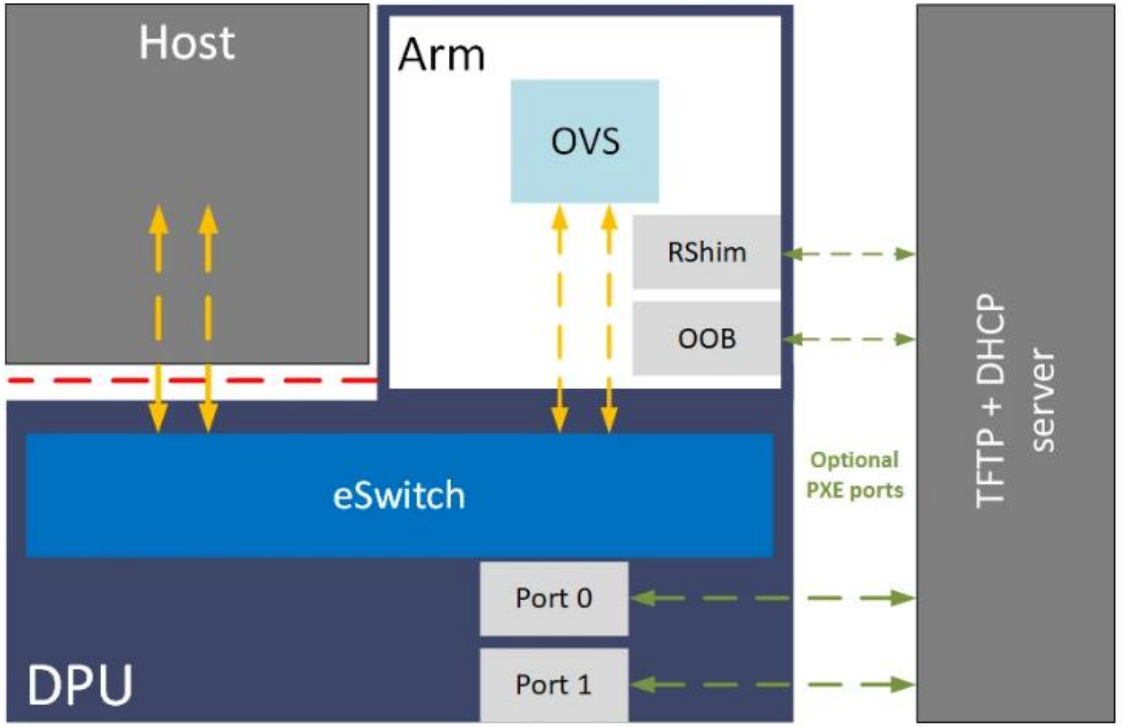
PXE

PXE OS Installation

Host or ARM

- PXE install host OS
  - Same PXE host OS installation with traditional NIC
  - If fail, need check
    - ARM OS is up
    - Virtual switch on DPU is up

- PXE install ARM OS
  - Deploying BlueField Software Using BFB with PXE - NVIDIA Docs
  - PXE installation is not supported with BlueField-3 NIC mode.
  - Not recommended in no-real deployment scenarios.
  - BFB Deployment through Rshim is much easier and popular.



NCSI Commands & DPU boot sequence

NCSI COMMAND

Native & NVIDIA Mellanox OEM command

- NCSI Native Spec - DSP0222
  - Universal spec for Network Controllers from various vendors
  - Frequently used commands for NVIDIA DPU/NIC
    - 0x0E Set MAC Address
    - 0x0A Get Link Status – Eth NIC
    - 0x15 Get Version ID - FW version
    - 0x16 Get Capabilities – VLan/Port count etc.

Command Type	Command Name	Description	Response Type	Command Support Requirement
0x05	Reset Channel	Used to synchronously put the Network Controller back to the Initial State	0x85	M
0x06	Enable Channel Network TX	Used to explicitly enable the channel to transmit Pass-through packets onto the network	0x86	M
0x07	Disable Channel Network TX	Used to explicitly disable the channel from transmitting Pass-through packets onto the network	0x87	M
0x08	AEN Enable	Used to control generating AENs	0x88	C
0x09	Set Link	Used during OS absence to force link settings, or to return to auto-negotiation mode	0x89	M
0x0A	Get Link Status	Used to get current link status information	0x8A	M
0x0B	Set VLAN Filter	Used to program VLAN IDs for VLAN filtering	0x8B	M
0x0C	Enable VLAN	Used to enable VLAN filtering of Management Controller RX packets	0x8C	M
0x0D	Disable VLAN	Used to disable VLAN filtering	0x8D	M
0x0E	Set MAC Address	Used to configure and enable unicast and multicast MAC address filters	0x8E	M
0x10	Enable Broadcast Filter	Used to enable selective broadcast packet filtering	0x90	M
0x11	Disable Broadcast Filter	Used to disable all broadcast packet filtering, and to enable the forwarding of all broadcast packets	0x91	M
0x12	Enable Global Multicast Filter	Used to enable selective multicast packet filtering	0x92	C
0x13	Disable Global Multicast Filter	Used to disable all multicast packet filtering, and to enable forwarding of all multicast packets	0x93	C
0x14	Set NC-SI Flow Control	Used to configure IEEE 802.3 flow control on the NC-SI	0x94	O
0x15	Get Version ID	Used to get controller-related version information	0x95	M
0x16	Get Capabilities	Used to get optional functions supported by the NC-SI	0x96	M
0x17	Get Parameters	Used to get configuration parameter values currently in effect on the controller	0x97	M
0x18	Get Controller Packet Statistics	Used to get current packet statistics for the Ethernet Controller	0x98	O
0x19	Get NC-SI Statistics	Used to request the packet statistics specific to the NC-SI	0x99	O
0x1A	Get NC-SI Pass-through Statistics	Used to request NC-SI Pass-through packet statistics	0x9A	O

- NVIDIA Mellanox OEM Command - PID: 1093216
  - NVIDIA DPU/NIC specific with rich features
  - Frequently used commands
    - Get Temperature
    - Get SmartNIC Mode / OS state
    - Get Link Status – IB
    - Get PF MAC Address

Table 3 - Mellanox OEM Specific Commands

Mellanox Command ID	Parameter	Command Description	Section
0x0	0x0	Get PF MAC Address	1.2.1
	0x1	Get FCoE Configuration	1.2.2
	0x2	Get PXE Configuration	1.2.3
	0x3	Get Multi-PF Capabilities	1.2.4
	0x4	Get SR-IOV Configuration	1.2.5
	0x5	Get Enhanced Tagging Configuration	1.2.6
	0x6	Get iSCSI Configuration	1.2.7
	0xA	Get Addresses Groups Count	1.2.8
	0x1A	Get Addresses	1.2.9
	0x1B	Get Allocated Management Address	1.2.10
	0x1C	Get Safe Mode Configuration	1.2.11
	0x20	Get Driver Information	1.2.12
	0x21	Get Cable Information	1.2.13
	0x22	Get Card VPD Information	1.2.14
	0x23	Get Card TLV Information	1.2.15
	0x24	Query Hosts	1.2.16
	0x25	Get Chassis Rate Limiting	1.2.17
	0x26	Get Rate Limiting	1.2.18
	0x27	Get Port ID	1.2.19
	0x28	Get Self Recovery Setting	1.2.20
	0x29	Get Interface Info	1.2.21
	0x2A	Get Device ID	1.2.22
	0x2B	Get Port ECC Counters	1.2.23
	0x2E	Get LLDP NB	1.2.24
	0x2F	Get Log Information	1.2.25
	0x31	Get Network Debug Info	1.2.26

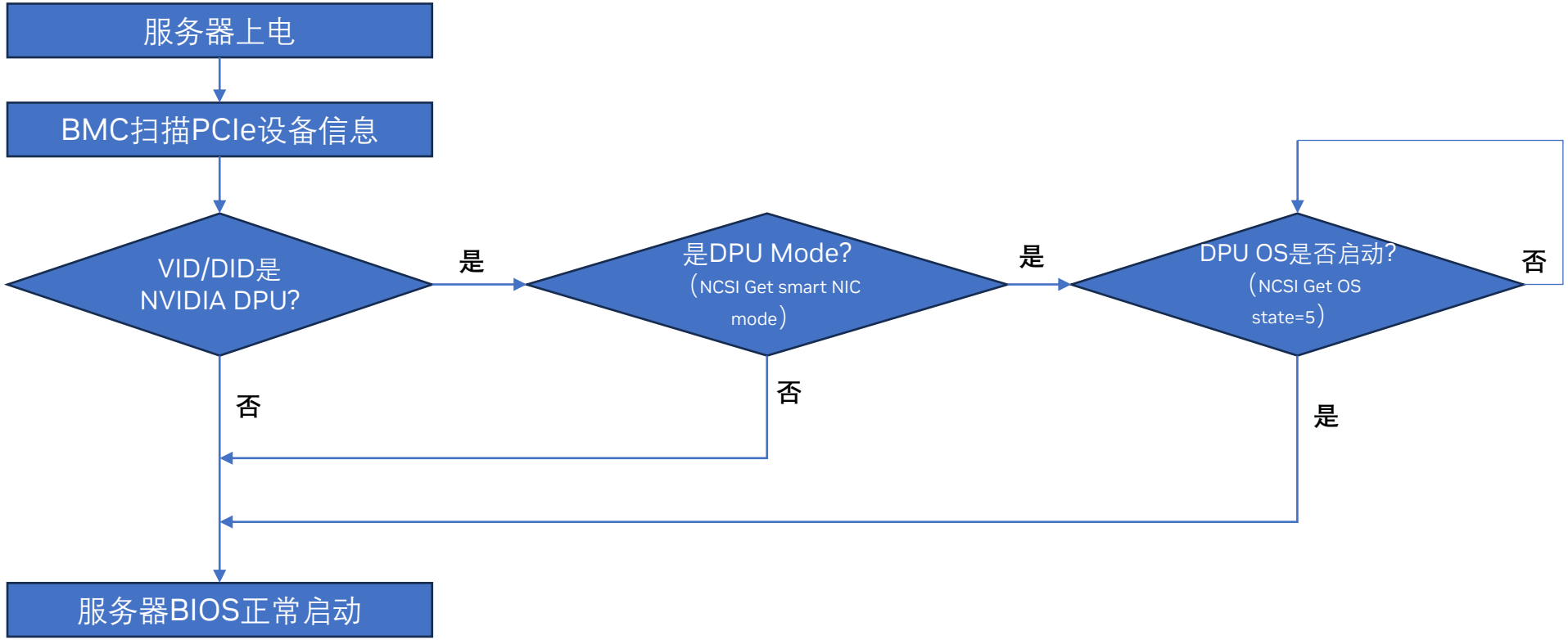
Table 3 - Mellanox OEM Specific Commands

Mellanox Command ID	Parameter	Command Description	Section
0x13	0x0	Get Port LED control	1.2.71
	0x1	Get Temperature Controls	1.2.72
	0x2	Get Temperature	1.2.73
	0x3	Get Register	1.2.74
	0x4	Get Mellanox AEN Controls	1.2.75
	0x5	Get Mellanox Link Status	1.2.76
	0x6	Get Electrical Sensors Count	1.2.77
	0x7	Get Electrical Sensor	1.2.78
	0x8	Get Electrical Sensors	1.2.79
	0x9	Get System Thermal Sensors Count	1.2.80
	0xB	Get PCIe Parameters	1.2.81
	0xC	Get Chip Registers	1.2.82
	0x11	Get Module Serial Data	1.2.83
	0x12	Get PHY Serial Data	1.2.84
	0x13	Get Challenge	1.2.85
	0x14	Get Debug Mode Info	1.2.86
0x14	0x0	Mellanox Indirect NC-SI	1.2.87
0x14 - 0xFF	Reserved	Reserved	

NCSI COMMAND - HOST BOOT AFTER DPU

Host CPU and DPU boot sequence control

- DPU S5 boot sequence
  - If OEM server doesn't support sustainable DPU power supply under S5 state, make sure DPU boot up/OS ready earlier than host CPU BIOS load



NCSI COMMAND - HOST BOOT AFTER DPU

Host CPU and DPU boot sequence control

- DPU OS state check on Host
  - BF2
    - mcra B:D.F 0xfb004
  - BF3
    - mcra B:D.F 0xf1930

Table 330 - Get Smart NIC OS State Response

Field	Bytes	Offset in NC-SI Command	Description
OS_State	1	31	Embedded CPU OS state 0 - Reset/Boot-ROM 1 - BL2 2 - BL31 3 - UEFI 4 - OS starting 5 - OS is running 6 - Low-Power standby Other - reserved

```
root@      l:~# mst status -v
MST modules:
------------
    MST PCI module is not loaded
    MST PCI configuration module loaded
PCI devices:
------------
DEVICE_TYPE             MST                           PCI       RDMA           NET
BlueField2(rev:1)       /dev/mst/mt41686_pciconf0     8e:00.0   mlx5_0         net-enp142s0f0np0

root@      :~# mcra 8e:00.0 0xfb004
0x00000005
```

```
root@l-csi      i:~# mst status -v
MST modules:
------------
    MST PCI module is not loaded
    MST PCI configuration module loaded
PCI devices:
------------
DEVICE_TYPE             MST                           PCI       RDMA           NET
BlueField3(rev:1)       /dev/mst/mt41692_pciconf0.1   17:00.1   mlx5_1         net-ens2f1np1

BlueField3(rev:1)       /dev/mst/mt41692_pciconf0     17:00.0   mlx5_0         net-ens2f0np0


root@l-csi      i:~# mcra 17:00.0  0xf1930
0x00000005
```

Multiple DPU Management

BlueField-3 Management Controller

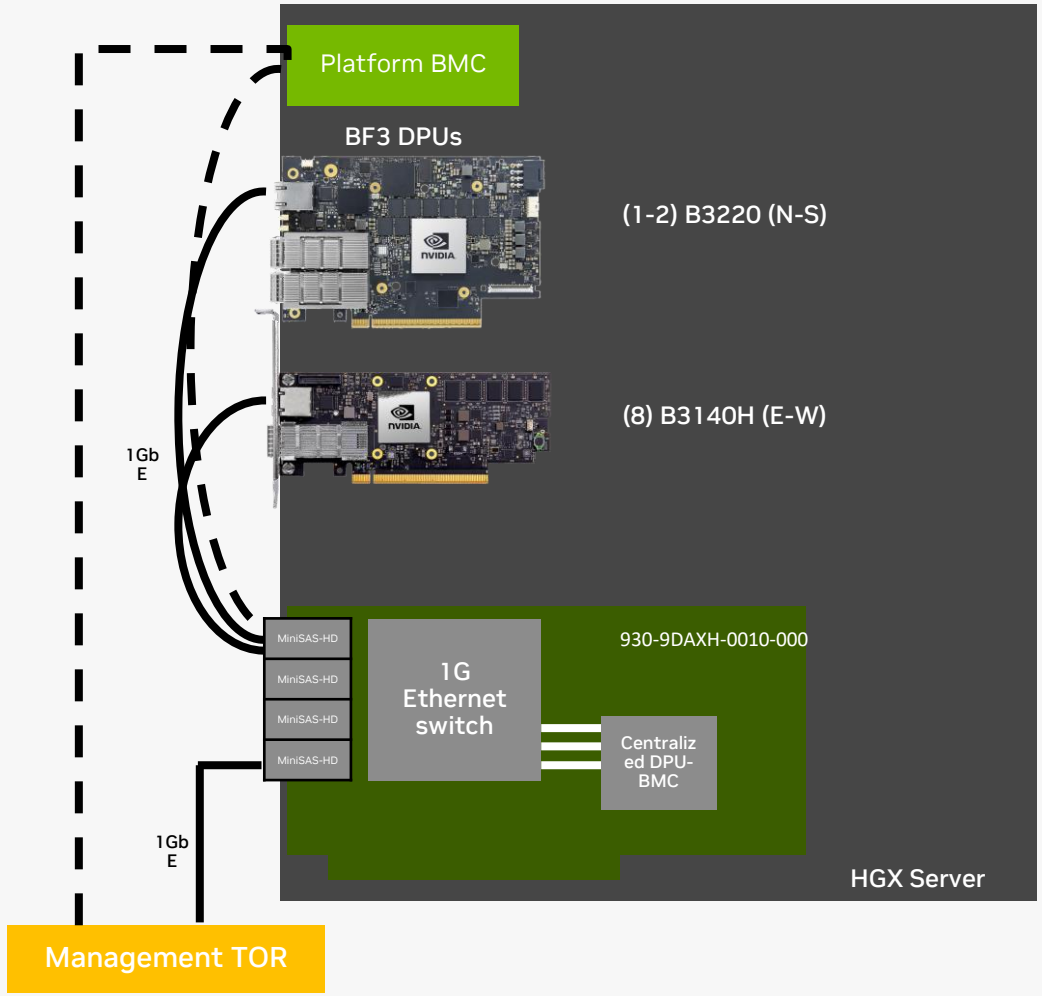
1GbE-Based Solution

Specifications

Network Speed	1GbE
Interface Type	12 x RJ45
Cables length	70 cm (fits 6U server connectivity)
Host Interface	Gen4 x8
Max Power	<25W
Thermal solution	Passive
Form factor	HHHL

Ordering Part Number

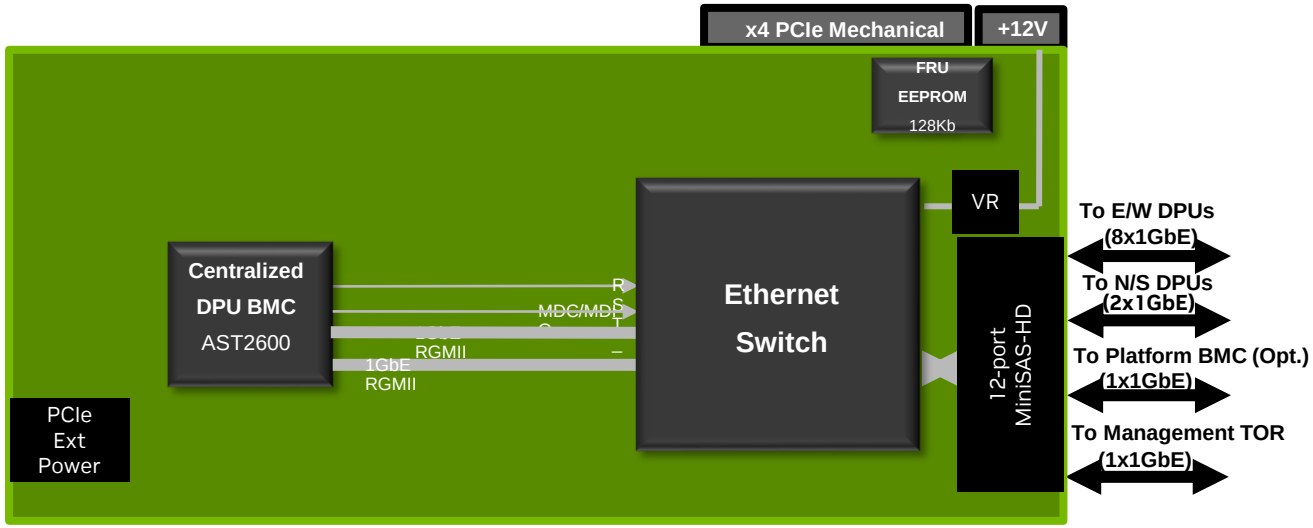
BlueField Management	930-9DAXH-0010-000



BlueField-3 Management Controller

Power-Efficient, Low-Profile Design

- PCIe HHHL form factor
- Est. power consumption <10W
- Redundant power supply options
  - Power supply via Golden-Fingers
  - Power supply via PCIe 8-pin Ext power connector (Enables Power-On option)
- Centralized BF3 BMC – identical circuitry to BF3 BMC
  - Full BF3 management through centralized BF3 BMC
  - Tunneling platform BMC through ethernet switch
- MiniSAS-HD cables:
  - MiniSAS-HD to 3x 1GbE connectivity
  - Up to 4 cables connection
  - Cables to be provided by NVIDIA
- To be included in NV-Cert/NV-Qual



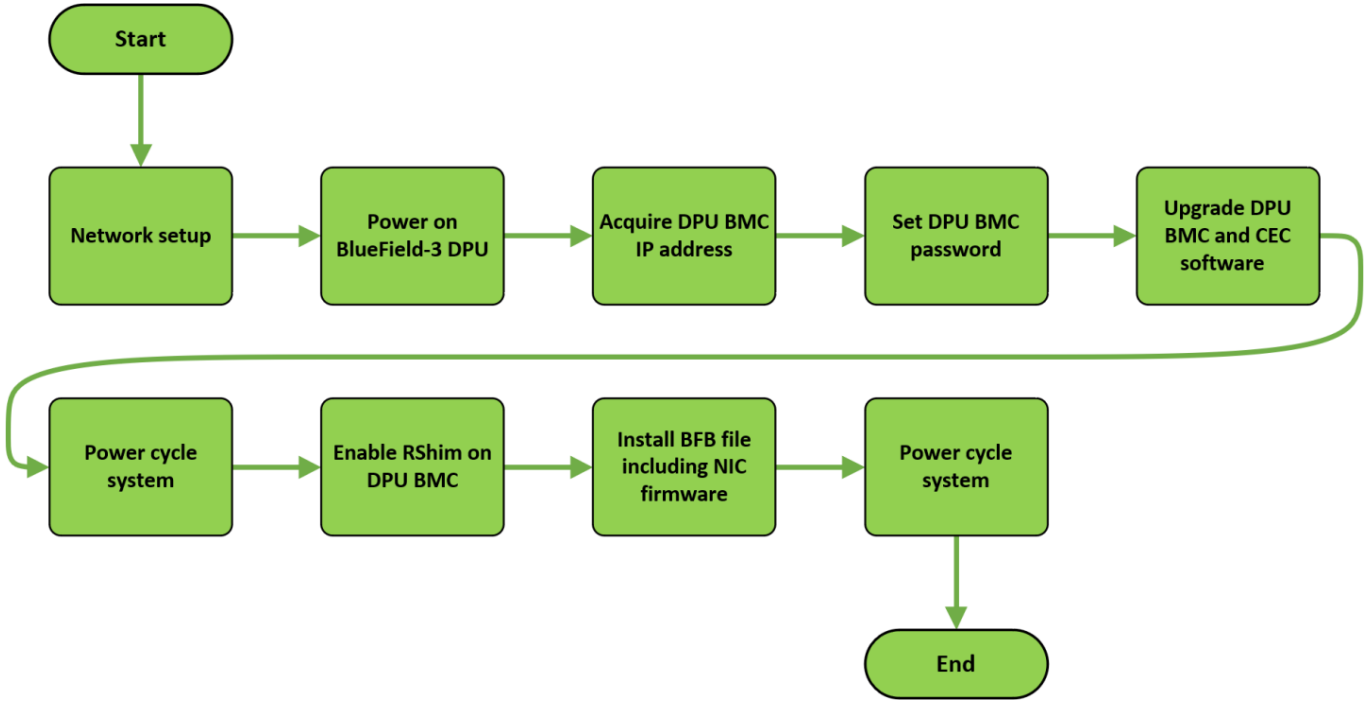
[PID#1116232 BlueField-3 Platforms Management Board User Manual]

DPU BMC

DPU BMC

Supported features

- IPMI 2.0 based features
  - System management – power state control
  - Environmental monitoring – Voltage/Power/Current
  - SOL (Serial over Lan)
  - Event management
- Redfish DMTF Standard - DSP0266
  - Restful API
- Image update
  - DPU bootstream/BFB
  - BMC image update
- BMC OS system is OpenBMC based, Web UI not supported



DPU BMC

Login & initial settings

- SSH/Serial
  - BMC OOB IP is DHCP by default - Get IP through IPMI tool
  - Login account: root/0penBmc（密码首字母为数字0)
  - Password change is required upon the first login

```
dpu-bmc login: root
Password: <Type default password>
You are obliged to immediately change your password (mandatory for administrators).
Changing the root password.
Current password: <Retype the default password>
New password: <Type the new password according to the above rules>
Retype the new password: <Retype the new password>
```

```
ubuntu@l-csi                :mgmt:~$ sudo ipmitool lan print 1
Set in Progress         : Set Complete
Auth Type Support       :
Auth Type Enable        : Callback :
                        : User     :
                        : Operator :
                        : Admin    :
                        : OEM      :
IP Address Source       : DHCP Address
IP Address              : 10.7.157.97
Subnet Mask             : 255.255.252.0
MAC Address             : 94:6d:ae:4e:99:a3
Default Gateway IP      : 10.7.156.1
Default Gateway MAC     : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
RMCP+ Cipher Suites     : 17
Cipher Suite Priv Max   : aaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
Bad Password Threshold  : Not Available
```

- Redfish – change default password
  - curl -k -u root:0penBmc -H "Content-Type: application/json" -X PATCH https://<bmc_ip>/redfish/v1/AccountService/Accounts/root -d '{"Password" : "<password>"}'

```
}ubuntu@l-csi                :~$curl -k -u root:'Nvidia_12345!' -H "Content-Type: application/json" -X PATCH  https://10.7.157.97/redfish/v1/AccountService/Accounts/root -d  '{"Password" : "Nvidia_Test1!"}'
{
  "@Message.ExtendedInfo": [
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The request completed successfully.",
      "MessageArgs": [],
      "MessageId": "Base.1.13.0.Success",
      "MessageSeverity": "OK",
      "Resolution": "None"
    }
  ]
}ubuntu@l-csi                :~$
```

DPU BMC

Change IP & User password

- Static IP/DHCP
  - Change IP through IPMItool
    - ipmitool lan set

```
ubuntu@l-csi-              :mgmt:~$ sudo ipmitool lan set 1 ipsrc static
ubuntu@l-csi-              :mgmt:~$ sudo ipmitool lan set 1 ipaddr 10.7.157.98
Setting LAN IP Address to 10.7.157.98
ubuntu@l-csi-              :mgmt:~$ sudo ipmitool lan set 1 defgw ipaddr 10.7.156.1
Setting LAN Default Gateway IP to 10.7.156.1
ubuntu@l-csi-              :mgmt:~$ sudo ipmitool lan set 1 netmask 255.255.252.0
Setting LAN Subnet Mask to 255.255.252.0
ubuntu@l-csi-              :mgmt:~$
```

```
ubuntu@l-csi-              :mgmt:~$ sudo ipmitool lan set 1 ipsrc dhcp
ubuntu@l-csi-              :mgmt:~$ sudo ipmitool lan print 1
Set in Progress         : Set Complete
Auth Type Support       :
Auth Type Enable        : Callback :
                        : User     :
                        : Operator :
                        : Admin    :
                        : OEM      :
IP Address Source       : DHCP Address
IP Address              : 10.7.157.97
Subnet Mask             : 255.255.252.0
MAC Address             : 94:6d:ae:4e:99:a3
Default Gateway IP      : 10.7.156.1
Default Gateway MAC     : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
RMCP+ Cipher Suites     : 17
Cipher Suite Priv Max   : aaaaaaaaaaaaaaa
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
Bad Password Threshold  : Not Available
```

- User Management
  - User creation/change password/user privilege
    - ipmitool user

```
ubuntu@l-csi-              :mgmt:~$ sudo ipmitool user list 1
ID  Name             Callin  Link Auth  IPMI Msg   Channel Priv Limit
1   root             false   true       true       ADMINISTRATOR
2                    true    false      false      NO ACCESS
3                    true    false      false      NO ACCESS
4                    true    false      false      NO ACCESS
5                    true    false      false      NO ACCESS
6                    true    false      false      NO ACCESS
7                    true    false      false      NO ACCESS
8                    true    false      false      NO ACCESS
9                    true    false      false      NO ACCESS
10                   true    false      false      NO ACCESS
11                   true    false      false      NO ACCESS
12                   true    false      false      NO ACCESS
13                   true    false      false      NO ACCESS
14                   true    false      false      NO ACCESS
15                   true    false      false      NO ACCESS
ubuntu@l-csi-              :mgmt:~$ sudo ipmitool user set password 1 Nvidia_Test2!
Set User Password command successful (user 1)
```

DPU BMC

Factory Reset / Reboot

- IPMI
  - Factory Reset
    - ipmitool raw 0x32 0x66
  - Reboot
    - ipmitool mc reset cold

```
ubuntu@l-csi[redacted]:mgmt:~$ sudo ipmitool mc reset cold
Sent cold reset command to MC
```

- Redfish
  - Factory Reset
    - curl -k -u root:"<PASSWORD>" -H "Content-Type: application/json" -X POST https:/<bmc_ip>/redfish/v1/Managers/Bluefield_BMC/Actions/Manager.ResetToDefaults -d '{"ResetToDefaultsType": "ResetAll"}'
  - Reboot
    - curl -k -u root:'<password>' -H "Content-Type: application/json" -X POST -d '{"ResetType": "GracefulRestart"}' https://<bmc_ip>/redfish/v1/Managers/Bluefield_BMC/Actions/Manager.Reset

```
ubuntu@l-csi[redacted]:~$ curl -k -u root:'Nvidia_12345!' -H "Content-Type: application/octet-stream" -X POST -d '{"ResetType": "GracefulRestart"}' https://10.7.156.108/redfish/v1/Managers/Bluefield_BMC/Actions/Manager.Reset
{
  "@Message.ExtendedInfo": [
    {
      "@odata.type": "#Message.v1_1_1.Message",
      "Message": "The request completed successfully.",
      "MessageArgs": [],
      "MessageId": "Base.1.15.0.Success",
      "MessageSeverity": "OK",
      "Resolution": "None"
    }
  ]
}
```

DPU BMC

Update BMC FW

- Redfish
  - Check current running FW
    - curl -k -u root:'<password>' -X GET https://<bmc_ip>/redfish/v1/UpdateService/FirmwareInventory/BMC_Firmware | jq -r ' .Version'

```
ubuntu@l-csi            :~$ curl -k -u root:'Nvidia_12345!'  -X GET  https://10.7.157.97/redfish/v1/UpdateService/FirmwareInventory/BMC_Firmware | jq -r ' .Version'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   497  100   497    0     0   2939      0 --:--:-- --:--:-- --:--:--  2958
bf-23.01-2-0-g0c7367a1e9.1673458885.7647724
```

  - Program FW
    - curl -k -u root:'<password>' -H "Content-Type: application/octet-stream" -X POST -T /home/bf3-bmc-24.01-5_opn.fwpkg https://<bmc_ip>/redfish/v1/UpdateService
    - curl -k -u root:'<password>' -X GET https://<bmc_ip>/redfish/v1/TaskService/Tasks/<task-id> | jq -r ' .PercentComplete'

```
ubuntu@l-csi-           :~$ curl -k -u root:'Nvidia_12345!' -H "Content-Type: application/octet-stream" -X POST -T /home/ubuntu/bf3-bmc-24.01-5_opn.fwpkg https://10.7.157.97/redfish/v1/UpdateService/update
{
  "@odata.id": "/redfish/v1/TaskService/Tasks/0",
  "@odata.type": "#Task.v1_4_3.Task",
  "Id": "0",
  "TaskState": "Running",
  "TaskStatus": "OK"
}
ubuntu@l-csi-bf3-200g-03:mgmt:~$curl -k -u root:'Nvidia_12345!' -X GET https://10.7.157.97/redfish/v1/TaskService/Tasks/0 | jq -r ' .PercentComplete'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2966  100  2966    0     0  20107      0 --:--:-- --:--:-- --:--:-- 20040
100
```

  - Power cycle DPU
    - curl -k -u root:'<password>' -H "Content-Type: application/json" -X POST -d '{"ResetType": "GracefulRestart"}' https://<bmc_ip>/redfish/v1/Managers/Bluefield_BMC/Actions/Manager.Reset

DPU BMC

NIC management

- Redfish
  - DPU mode Check/Setting
    - curl -k -u root:'<password>' -X GET https://<bmc_ip>/redfish/v1/Systems/Bluefield/Oem/Nvidia

```
ubuntu@l-csi-[redacted]:~$ curl -k -u root:'Nvidia_12345!' -X GET https://10.7.156.108/redfish/v1/Systems/Bluefield/Oem/Nvidia
{
  "Actions": {
    "#Mode.Set": {
      "Parameters": [
        {
          "AllowableValues": [
            "NicMode",
            "DpuMode"
          ],
          "DataType": "String",
          "Name": "Mode",
          "Required": true
        }
      ],
      "target": "/redfish/v1/Systems/Bluefield/Oem/Nvidia/Actions/Mode.Set"
    }
  },
  "Mode": "DpuMode"
}
```

  - Host Rshim Enable/Disable
    - curl -k -u root:'<password>' -H "Content-Type: application/json" -X POST -d '{"HostRshim":"Enabled"}' https://<bmc_ip>/redfish/v1/Systems/Bluefield/Oem/Nvidia/Actions/HostRshim.Set

- IPMI
  - DPU mode Check/Setting
  - Host Rshim Enable/Disable
  - DPU SmartNIC OS State

Changing Operation Mode

netfunc	cmd	data	Description
0x32	0x9D	0x1	Change to DPU mode
0x32	0x9D	0x0	Change to NIC mode

Enable/Disable RShim from Host

netfunc	cmd	data	Description
0x32	0x9F	0x1	Enable RShim from host
0x32	0x9F	0x0	Disable RShim from host

DPU BMC

Monitoring

- Redfish
  - DPU SoC temperature/Port temperature/Voltage
    - curl -k -u root:'<password>' -H 'Content-Type: application/json' -X GET https://<bmc_ip>/redfish/v1/Chassis/Card1/Sensors

Sensor Name	Sensor Type	Source	Description
p0_link	Discrete	IPMB	Uplink port 0 link status • 0x100 – connection OK • 0x200 – connection error
p1_link	Discrete	IPMB	Uplink port 1 link status • 0x100 – connection OK • 0x200 – connection error
bluefield_temp	Temperature	IPMB	Bluefield DPU Temperature
p0_temp	Temperature	IPMB	Uplink port 0 SFP temperature
p1_temp	Temperature	IPMB	Uplink port 1 SFP temperature
1V_BMC	Voltage	BMC ADC	
1_2V_BMC	Voltage	BMC ADC	
1_8V	Voltage	BMC ADC	
1_8V_BMC	Voltage	BMC ADC	
2_5V	Voltage	BMC ADC	

- IPMI
  - DPU SoC temperature/Port temperature/Voltage
    - ipmitool -C 17 -I lanplus -H <bmc_ip> -U root -P <password> sdr list

```
ubuntu@l-csi-          :~$ sudo ipmitool -C 17 -I lanplus -H 10.7.156.108 -U root -P Nvidia_12345!  sdr list
p0_link          | 0x00              | ok
p1_link          | 0x00              | ok
bluefield_temp   | no reading        | ns
p0_temp          | no reading        | ns
p1_temp          | no reading        | ns
1V_BMC           | 0.99 Volts        | ok
1_2V_BMC         | 1.19 Volts        | ok
1_8V             | 1.78 Volts        | ok
1_8V_BMC         | 1.80 Volts        | ok
2_5V             | 2.49 Volts        | ok
3_3V             | 3.35 Volts        | ok
3_3V_RGM         | 3.39 Volts        | ok
5V               | 4.96 Volts        | ok
12V_ATX          | 12.26 Volts       | ok
12V_PCIe         | 12.32 Volts       | ok
DVDD             | 0.87 Volts        | ok
HVDD             | 1.20 Volts        | ok
VDD              | 0.78 Volts        | ok
VDDQ             | 1.10 Volts        | ok
VDD_CPU_L        | 0.94 Volts        | ok
VDD_CPU_R        | 0.94 Volts        | ok
```

REFERENCED DOCUMENTS FROM DMTF

Spec name	Document link
NC-SI	DSP0222
MCTP Base spec	DSP0236
MCTP over SMBus	DSP0237
MCTP over PCIe	DSP0238
PLDM Base spec	DSP0240
PLDM for monitoring and control	DSP0248
PLDM for NIC Modeling	DSP2054
PLDM for FRU	DSP0257
PLDM for FW Update	DSP0267
Redfish Device Enablement	DSP0218
Redfish	DSP0266
Redfish Host Interface	DSP0270
Redfish schema	DSP8010

REFERENCED NVIDIA PUBLIC DOCUMENTS

Spec name	Document link
NVIDIA Mellanox NC-SI OEM commands	PID: 1093216 NC-SI OEM Commands Application Note
NVIDIA BlueField-2 DPU User Guide	BF2 DPU User Guide
NVIDIA BlueField-3 DPU User Guide	BF3 DPU User Guide
NVIDIA BlueField BMC Software User Manual	NVIDIA BlueField BMC Software User Manual
NVIDIA BlueField DPU Management & Provision	NVIDIA BlueField Management & Provision
NVIDIA DOCA Documentation v2.6	DOCA Documentation v2.6


NVIDIA