NSN ANPI1-A/NIRT3-A

Embedded Software Specification

Authors: Dave Lyons, Prakash Siva, Nilan Naidoo, Vlad Rakocevic
Email: dave.lyons@radisys.com, prakash.siva@radisys.com, nilan.naidoo@radisys.com, vlad.rakocevic@radisys.com
Owner: Dave Lyons

002-xxxxx-0000
Revision 0.80
Feb 10, 2015

Note: To verify signoff, see the on-line repository.

	Name	Title/Department
Approvers:	Jeff Moll	PLM
	Vlad Rakocevic	S/W Manager
	Andrew Alleman	VP Engineering
Reviewers:	Vlad Rakocevic	Sr. Software Manager
	Nilan Naidoo	Sr. Software Architect
	Prakash Siva	Sr. Software Architect
	Alan Anderson	Sr. Software Architect

Preface

On-line Document Repositories

- Approved versions of document are stored in Agile under the Radisys document number
- Interim released versions of document are stored in AB5 collaboration site under Shared Documents section.

Revision History

All revisions that start with zero (i.e., 0.5, 0.65, etc.) are preliminary drafts for internal reviews.

Table 1. Revision history

No.	Date	Author	Description
0.02	21 Nov 2012	Dave Lyons	Laying out document and beginning to add content.
0.03	28 Nov 2012	Dave Lyons	Added requirements matrix in appendix.
0.04	29 Nov 2012	Dave Lyons	Additional requirements info and some document formatting changes.
0.10	17 Dec 2012	Dave Lyons	Added port tables and functionality descriptions for additional requirements.
0.50	27 Dec 2012	Dave Lyons/Nilan Naidoo	Unit computer and IPMI content added. Additional references added to requirements matrix. Readied for draft review by NSN.
0.51	31 Dec 2012	Prakash Siva	Added NPU related section 3.4
0.52	10 Jan 2013	Dave Lyons	Added content to give overview to functionality already implemented in standard Radisys products.
0.53	17 Jan 2013	Dave Lyons	Minor corrections/clarifications.
0.54	18 Jan 2013	Dave Lyons/Nilan Naidoo	Additional content/changes to address NSN review feedback.
0.55	22 Jan 2013	Vlad Rakocevic	Updates on OS Requirements
0.56	30 Jan 2013	Dave Lyons	Updates to address NSN feedback
0.60	02 Feb 2013	Vlad Rakocevic	Addressed NSN questions for HPM and OS requirements
0.80	14 Feb 2013	Dave Lyons, Vlad Rakocevic	Addressed NSN comments from review of 0.60 version.
0.81	19 Feb 2013	Dave Lyons	Formatting changes.

Notational Conventions

Contents

Figures

Tables

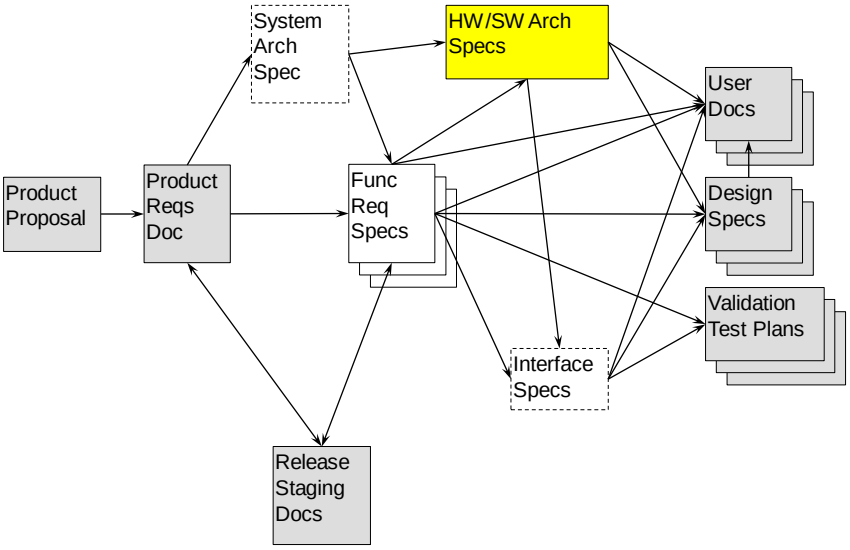
1. Overview

1.1 Document Overview

This document provides the detailed functional requirements for the NSN ANPI1-A/NIRT3-A based on NSN requirements and HPRS documentation. Some additional agreements on particular requirements and functionality were reached during face-to-face meeting and electronic exchanges, and the outcomes are captured in later revisions of NSN documentation .

Figure 1 shows the relative positioning of the FRS in the overall product requirements process flow.

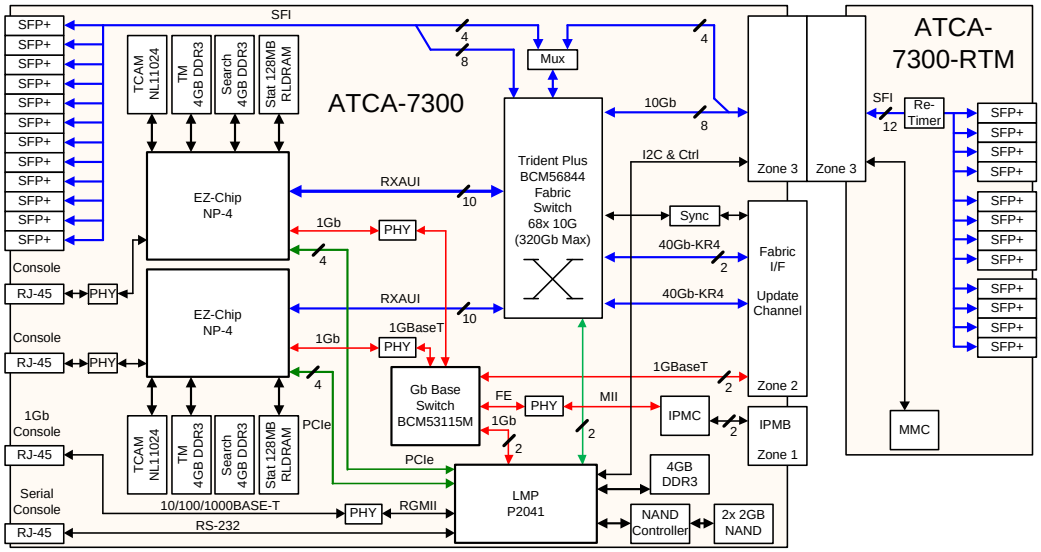
Figure 1. Product Specification Document Flow



1.2 Product Overview

The Nokia Siemens Networks (NSN) ANPI1-A is a blade with two NP-4 NPU processors connected by a Broadcom 565844 Trident 40G Ethernet switch. This blade is part of NSN's AB5 program. The blade hardware and software concept are described in [7] in the References section.

Figure 2. ANPI1-A Hardware Block Diagram



1.3 Terminology

Term	Definition
ANPI1-A	Radisys-developed dual NP-4 processor blade.

1.4 Acronyms

AMC	Advanced Mezzanine Card.
ATCA	Advanced Telecom Computing Architecture
BITS	Building Integrated Timing Source
BOOTP	BOOTstrap Protocol
CAE	Content Aware Engine
CLI	Command Line Interface
DC	Direct current.
DFA	Deterministic Finite Automata
DFT	Design For Test
DFM	Design for manufacturing
DHCP	Dynamic Host Configuration Protocol
EAU	Early Availability Unit
EJTAG	Enhanced JTAG
GARP	Generic Attribute Registration Protocol
GbE	Gigabit Ethernet
GE	Gigabit Ethernet
GMRP	GARP Multicast Registration Protocol
GP	General Purpose
GVRP	GARP VLAN Registration Protocol
HFA	Hyper Finite Automata
I^2C	Inter-Integrated Circuit (Bus).
I/O	Input/output.
IC	Integrated circuit.

ICMP	Integrated Control Message Protocol
IPMB	IPMI Bus.
JTAG	Joint Test Action Group.
LED	Light Emitting Diode.
LMP	Local Management Processor.
MMC	Module Management Controller.
NEBS	Network Equipment Building Standard.
NSP	Network Service Processor.
NV	Non-volatile.
PCB	Printed circuit board.
PCI	Peripheral Component Interconnect
PCIe	PCI-Express
PCI -X	Peripheral Component Interconnect Extended
PICMG	PCI Industrial Computer Manufacturers Group.
P/N	Part number.
PPM40-RM	Packet Processing Module – 20 Gb
RL	Reduced Latency
RoHS	Restriction of Hazardous Substances
RPL	Recommended parts list
RTM	Rear transition module.
SCTP	Simple Control Transport Protocol
SFP	Small Form-factor Pluggable
ShMC	Shelf Management Controller
ShMS	Shelf Management Server
SMC	Satellite Management Controller
SMPMC	System Manager PMC
SNMP	Simple Network Management Protocol
SMT	Surface mount technology.
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TSEC	Triple Speed Ethernet Controller
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
VLP	Very low profile

1.5 Issues List

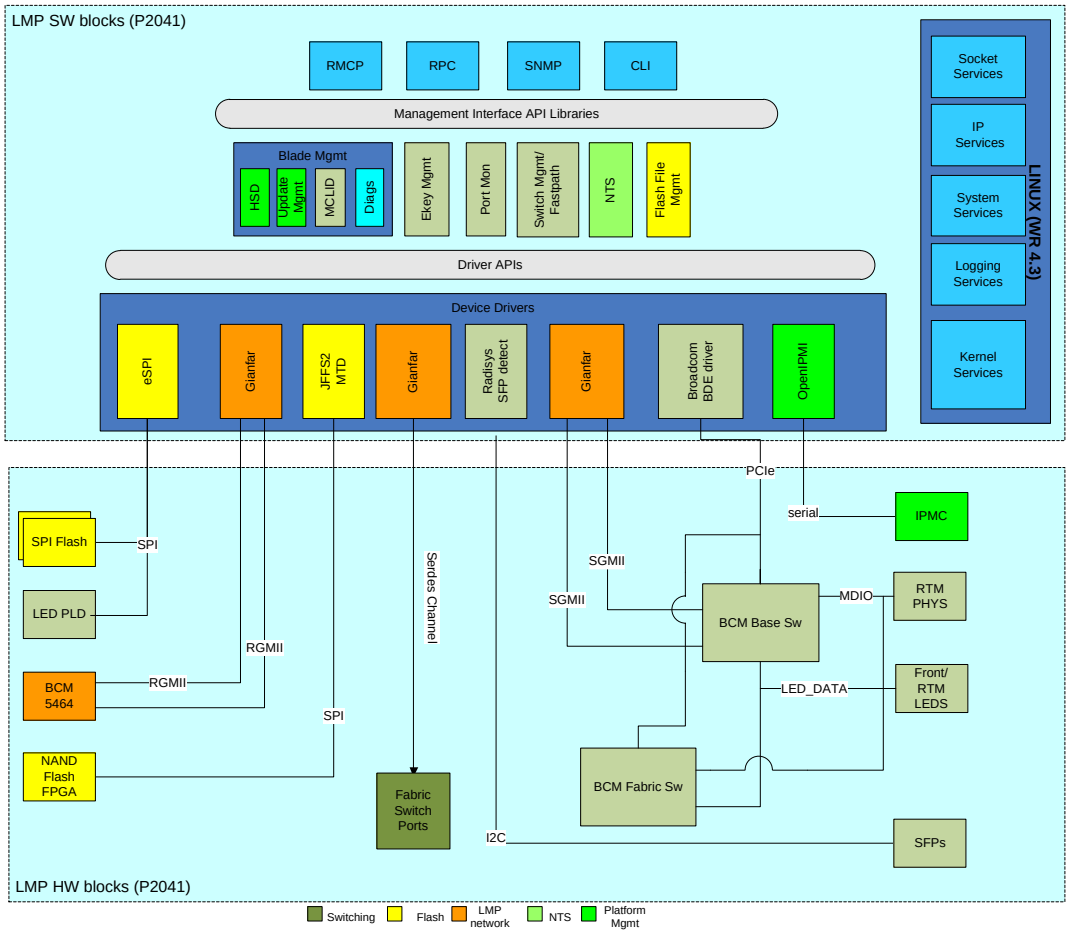
This section provides a list of issues that need to be resolved in order to complete the document. Additional are scattered throughout the document, flagged by "**ISSUE:**" or formatted as questions.

1. [Issue # N.] RESOLUTION ➔ Resolution of Issue #N.

2. Software Design

The following diagram gives a high-level, block diagram of the software used on the ANPI1-A and the hardware blocks controlled by the various device drivers.

Figure 3. ANPI1-A Software Block Diagram (Placeholder)

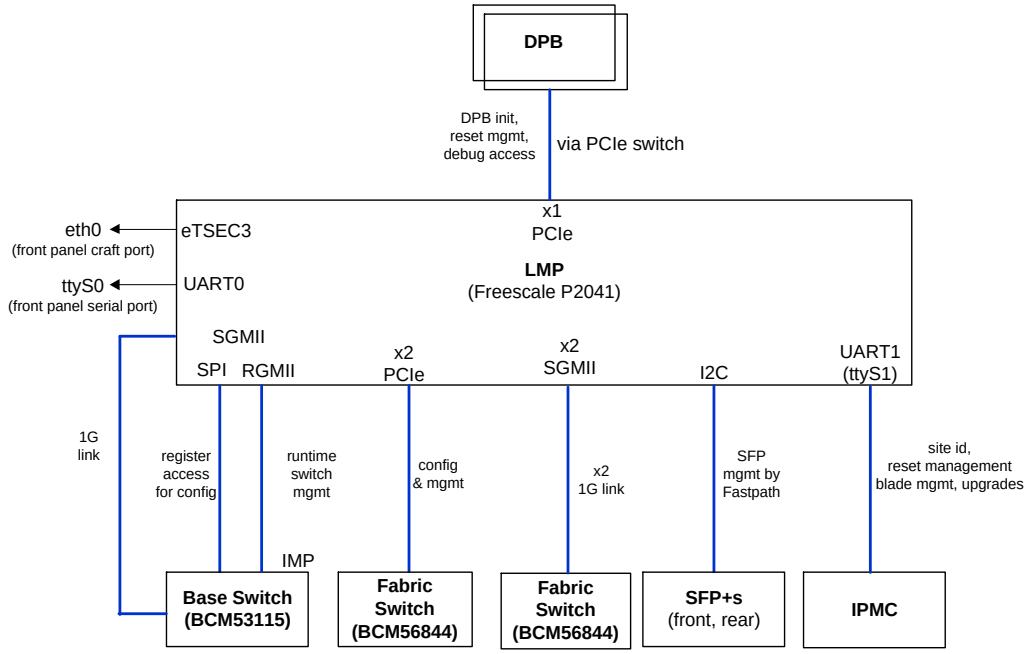


2.1 Internal Structure

2.1.1 LMP Physical Management Interfaces

Figure 4 shows the management connectivity for the various interfaces available to the ATCA-7240 LMP for configuring, managing and monitoring the HW. Debug interfaces like the craft port and serial port are also shown.

Figure 4. LMP debug and management connectivity



As shown above, the base switch has dual management interfaces from the LMP - SPI and the Ethernet based IMP. The former is used for initial configuration of the switch registers while the latter is capable of packet transfer necessary for runtime switch management. There is also an SGMII port for carrying normal Ethernet traffic.

There are two Ethernet links from the LMP to the Trident+ fabric switch for carrying application specific traffic.

The PCIe switch has to be initialized by the LMP and is used for EZ-Chip configuration and management.

Figure 5 shows the management connectivity available to the EZ-Chip blocks.

Figure 5. DPB management and debug connectivity

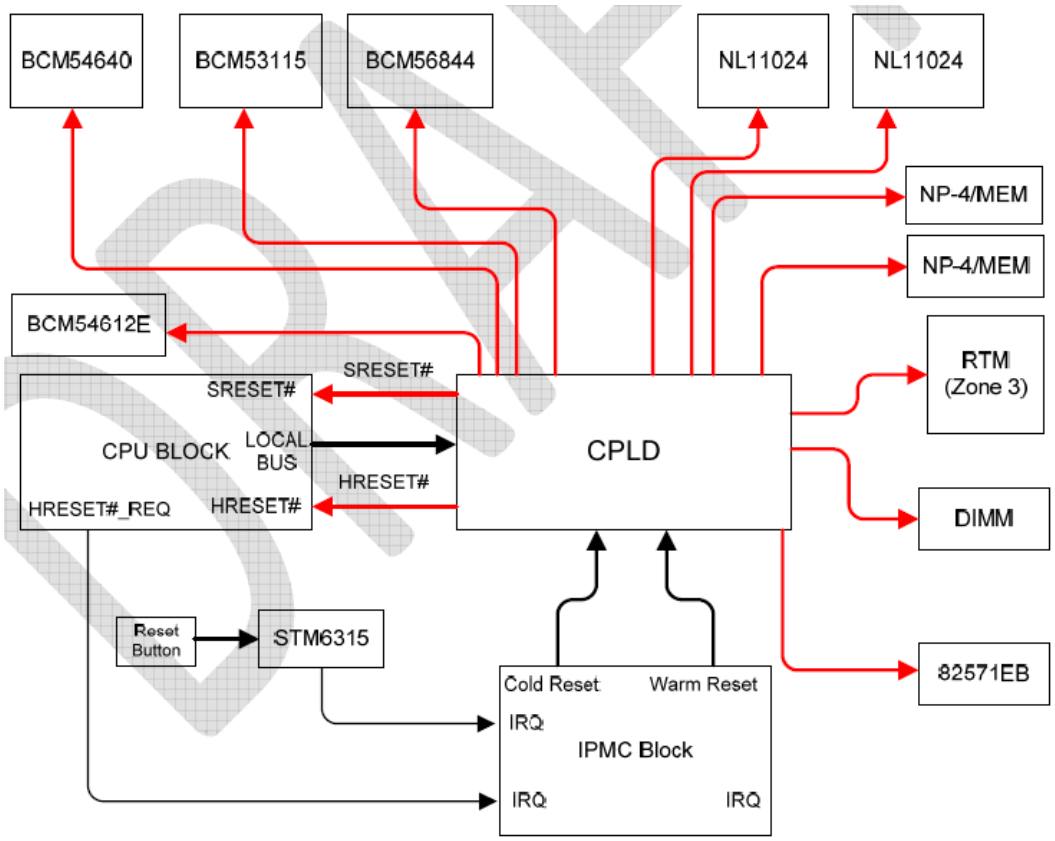


Table 2 lists the network interfaces available to the LMP.

Table 2. LMP network interfaces

Interface/ Device Name	Connectivity/usage
pci0	Packet interface between P2041 and BCM56844 switch. Implemented using PCIe bus. Reserved for internal use.
eth0	10/100/1000 Mbps mgmt & debug port. This is RJ45 connectivity to front panel (eTSEC3): By default, the ANPI1-A uses DHCP to configure this interface's address. An eth0:0 sub-interface with an IP address of 10.1.<chassis>.<slot>/16 and an eth0:1 sub-interface with an IP address of 10.0.0.1/24 are also created by default.
eth1	1 Gbps interface to the Base switch. By default, the ANPI1-A uses DHCP to configure this interface's address. (eTSEC1)
eth2	1 Gbps Ethernet connectivity to base switch Inband Management Port (IMP) (eTSEC2). Used for internal switch management.
eth3	1Gbps Ethernet connectivity to fabric switch port. By default, the ANPI1-A uses DHCP to configure this interface's address.
eth4	Additional 1Gbps Ethernet link to fabric switch port. By default, this link will not use DHCP.
ez0	PCI point-to-point interface between the LMP and EZ-Chip #1

Interface/ Device Name	Connectivity/usage
ez1	PCI point-to-point interface between the LMP and EZ-Chip #2
ttyS0	Serial interface serves both front panel & RTM (Linux console device). Either port can be used for console or debug.
ttyS1	Serial interface to the IPMC processor.

2.1.2 Base and Fabric Ethernet Interfaces

This chapter captures the default initial state of the switch ports (base in Table 3; fabric in Table 4). All rear/front fabric switch ports are disabled by default. The initial switch states are captured in configuration file, rc.soc. This file can be modified by user. The file intent is to be an initial script for bcm.user.

2.1.2.1 Port Mapping for Base and Fabric

This section defines the details of the base interfaces of the ANPI1-A. The Ethernet Interface Seed is taken from an EEPROM on the blade. The seed + 0 interface is eth0 (refer to Table 2).

Table 3. ANPI1-A Base Ports and MAC addresses

ANPI1-A Port Description	BCM53115M HW Port Name	Switch Logical Port	Interface Speed (MBps)	Interface Type	Ethernet Interface Seed +	CLI Port Number	ifIndex
Backplane Z2 Base Ch-1	TRD{4}	4	10/100/1000	Copper	5	0/1	1006
Backplane Z2 Base Ch-2	TRD{3}	3	10/100/1000	Copper	6	0/2	1007
NP-4 1	TRD{0}	0	1000	Copper	7	2/1	1008
NP-4 2	TRD{1}	1	1000	Copper	8	3/1	1009
LMP eTSEC2	GMII	5	10/100	RGMII	9	4/1	1010
IPMC SOL	TRD{2}	2	10/100/1000	Copper	10	4/2	1011
LMP eTSEC1	IMP	CPU		RGMII	1	N/A	N/A

The table below shows the details of the fabric interfaces of the ANPI1-A. Although the ANPI1-A has 12 front panel ports, only the 1/1 through 1/8 (the top 8 ports) are configured to connect to the Trident+ switch and pass traffic. All 12 RTM ports (5/1 – 5/12) are configured to connect to the Trident+ switch and pass traffic.

Table 4. Trident+ Logical Port Mapping

PPM40 RM Port Description	BCM56846 HW Port Name	Switch Logical Port	Interface Speed (MBps)	Interface Type	Ethernet Interface Seed +	CLI Port Number	ifIndex
Backplane Z2 Fabric Ch-1	XGXS7[0:3]/ XGXS7[0]	1	40000/10000	40G-KR4/ 10G-KR	12	0/1	2001
Backplane Z2 Fabric Ch-2	XGXS6[0:3]/ XGXS6[0]	2	40000/10000	40G-KR4/ 10G-KR	13	0/2	2002
Front SFP+ 1	XGXS16[0]	3	10000/1000	10G-SFI	14	1/1	2009
Front SFP+ 2	XGXS16[1]		10000/1000	10G-SFI	15	1/2	2010
Front SFP+ 3	XGXS16[2]	5	10000/1000	10G-SFI	16	1/3	2011
Front SFP+ 4	XGXS16[3]	6	10000/1000	10G-SFI	17	1/4	2012
Front SFP+ 5	XGXS17[0]	7	10000/1000	10G-SFI	18	1/5	2013
Front SFP+ 6	XGXS17[1]	8	10000/1000	10G-SFI	19	1/6	2014
Front SFP+ 7	XGXS17[2]	9	10000/1000	10G-SFI	20	1/7	2015
Front SFP+ 8	XGXS17[3]	10	10000/1000	10G-SFI	21	1/8	2016
DPB1 Port 1	XGXS13[2:3]	15	10000	RXAUI	26	2/1	2025
DPB1 Port 2	XGXS13[0:1]	16	10000	RXAUI	27	2/2	2026
DPB1 Port 3	XGXS12[2:3]	17	10000	RXAUI	28	2/3	2027
DPB1 Port 4	XGXS12[0:1]	18	10000	RXAUI	29	2/4	2028
DPB1 Port 5	XGXS11[2:3]	19	10000	RXAUI	30	2/5	2029
DPB1 Port 6	XGXS11[0:1]	20	10000	RXAUI	31	2/6	2030
DPB1 Port 7	XGXS10[2:3]	21	10000	RXAUI	32	2/7	2031
DPB1 Port 8	XGXS10[0:1]	22	10000	RXAUI	33	2/8	2032

Table 4. Trident+ Logical Port Mapping

PPM40 RM Port Description	BCM56846 HW Port Name	Switch Logical Port	Interface Speed (MBps)	Interface Type	Ethernet Interface Seed +	CLI Port Number	ifIndex
DPB1 Port 9	XGXS8[2:3]	23	10000	RXAUI	34	2/9	2034
DPB1 Port 10	XGXS8[0:1]	24	10000	RXAUI	35	2/10	2035
DPB2 Port 1	XGXS5[2:3]	25	10000	RXAUI	36	3/1	2041
DPB2 Port 2	XGXS5[0:1]	26	10000	RXAUI	37	3/2	2042
DPB2 Port 3	XGXS3[2:3]	27	10000	RXAUI	38	3/3	2043
DPB2 Port 4	XGXS3[0:1]	28	10000	RXAUI	39	3/4	2044
DPB2 Port 5	XGXS2[0:1]	29	10000	RXAUI	40	3/5	2045
DPB2 Port 6	XGXS2[2:3]	30	10000	RXAUI	41	3/6	2046
DPB2 Port 7	XGXS1[2:3]	31	10000	RXAUI	42	3/7	2047
DPB2 Port 8	XGXS1[0:1]	32	10000	RXAUI	43	3/8	2048
DPB2 Port 9	XGXS0[2:3]	33	10000	RXAUI	44	3/9	2049
DPB2 Port 10	XGXS0[0:1]	34	10000	RXAUI	45	3/10	2050
LMP Fabric Port 0	XGXS4[0]	35	1000	SGMII	46	4/1	2056
LMP Fabric Port 1	XGXS4[1]	36	10000	SGMII	47	4/2	2057
Update Channel 1	XGXS4[2]	37	10000	10G-KR	48	4/3	2058
Update Channel 2	XGXS4[3]	38	10000	10G-KR	49	4/4	2059
RTM Port 1	XGXS14[0]	39	10000/1000	10G-SFI	50	5/1	2064
RTM Port 2	XGXS14[1]	40	10000/1000	10G-SFI	51	5/2	2065
RTM Port 3	XGXS14[2]	41	10000/1000	10G-SFI	52	5/3	2066
RTM Port 4	XGXS14[3]	42	10000/1000	10G-SFI	53	5/4	2067
RTM Port 5	XGXS9[0]	43	10000/1000	10G-SFI	54	5/5	2068
RTM Port 6	XGXS9[1]	44	10000/1000	10G-SFI	55	5/6	2069

Table 4. Trident+ Logical Port Mapping

PPM40 RM Port Description	BCM56846 HW Port Name	Switch Logical Port	Interface Speed (MBps)	Interface Type	Ethernet Interface Seed +	CLI Port Number	ifIndex
RTM Port 7	XGXS9[2]	45	10000/1000	10G-SFI	56	5/7	2070
RTM Port 8	XGXS9[3]	46	10000/1000	10G-SFI	57	5/8	2071
RTM Port 9	XGXS15[3]	47	10000/1000	10G-SFI	58	5/9	2072
RTM Port 10	XGXS15[2]	48	10000/1000	10G-SFI	59	5/10	2073
RTM Port 11	XGXS15[1]	49	10000/1000	10G-SFI	60	5/11	2074
RTM Port 12	XGXS15[0]	50	10000/1000	10G-SFI	61	5/12	2075

2.1.2.2 Port Filtering Configuration for Base and Fabric

The default port-filtering configurations for the base and fabric interfaces are shown in Table 5 and Table 6 below.

Table 5. Port Filtering Setup for Base Interfaces

Port Description	Ports/Port Group	Port-filter Egress Set	Comments
Zone 2 Backplane Ports	0/1, 0/2	2/1, 3/1, 4/1	Zone 2 ports send to NP-4s and LMP
NP-4 #1 Ports	2/1	0/x, 3/1, 4/1	NP-4 #1 sends to backplane, NP-4 #2 and LMP
NP-4 #2 Ports	3/1	0/x, 2/1, 4/1	NP-4 #2 sends to backplane, NP-4 #1 and LMP
LMP Port	4/1	0/x, 2/1, 3/1	LMP sends to backplane, NP-4 #1 and NP-4 #2

Table 6. Port Filtering Setup for Fabric Interfaces

Port Description	Ports/Port Group	Port-filter Egress Set	Comments
Zone 2 Backplane Ports	0/1, 0/2	2/x, 3/x, 4/x	Doesn't allow backplane ports to send traffic to front panel/RTM ports
Front Panel Ports	1/x	2/x, 3/x	Front panel sends traffic to both NP-4s
NP-4 #1 Ports	2/x	0/x, 1/1–1/4, 3/x, 4/x, 5/1–5/6	NP-4 #1 sends traffic to backplane as well as front panel, NP-4 #2, LMP and RTM ports
NP-4 #2 Ports	3/x	0/x, 1/5-1/8, 2/x, 4/x, 5/7-5/12	NP-4 #2 sends traffic to backplane as well as front panel, NP-4 #1, LMP and RTM ports
LMP Ports	4/x	0/x, 2/x, 3/x	LMP sends traffic to backplane and NP-4 #s
RTM Ports	5/x	2/x, 3/x	RTM sends traffic to both NP-4s

Port Description	Ports/Port Group	Port-filter Egress Set	Comments
LAG Ports	6/x	None	Port channel port filtering must be configured

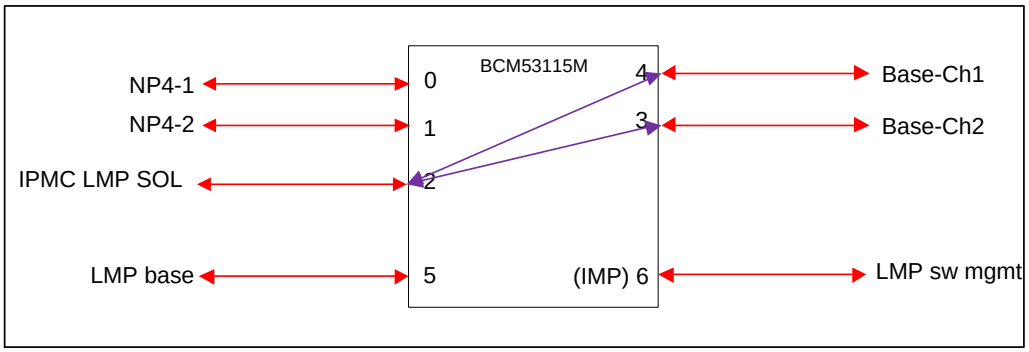
2.1.2.3 Default VLAN Membership

The default configuration for all base and fabric ports is to participate in VLAN 1 and to assign VLAN 1 to incoming traffic. VLAN tags are not added to any egress traffic by default.

2.1.2.4 Initial Base Switch Configuration to support SOL

Serial Over LAN feature is available on ANPI-1 blade. The settings of the IPM Controller should be in place so that the proper IP encapsulation is used for capturing traffic from the serial port. "**Set LAN Configuration Parameters**" IPMI command will be used to provide LAN parameters to IPMC for its LAN channel. The LAN parameters will be saved in a persistent storage (persistent over blade reinsertion). Once a SOL connection is established, the saved LAN parameters will be used to construct the IP header for the serial link (UART) payload encapsulation. The base switch configuration is set very early during U-Boot initialization to enable the IPMC LAN port and both base channel interfaces so the serial console is available during the INIT_R process shown in 3.3.1.4. The base switch egress port assignments shall be set as shown in FIGURE 6.

Figure 6 Base switch SOL connections during boot stage



The base switch port enable and egress assignments are never changed during the DYNAMIC boot phase when selecting BI0 or BI1 ports for DHCP. SOL connectivity shall be maintained until U-Boot has passed control to the Linux kernel.

Figure 7. Base switch SOL connections during DYNAMIC boot from BI0

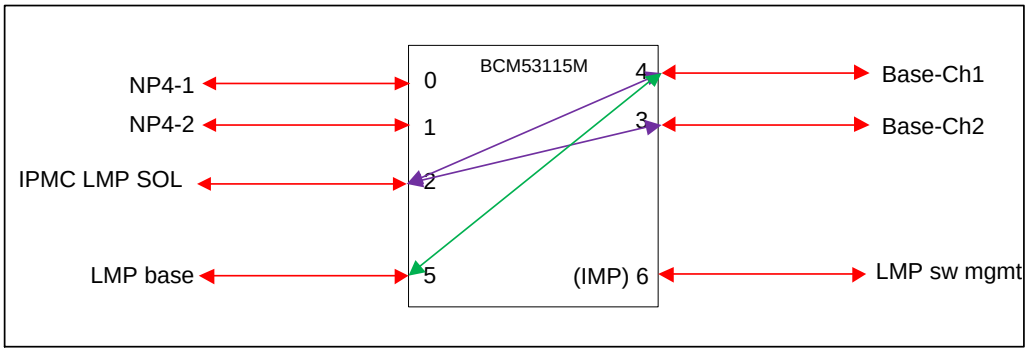
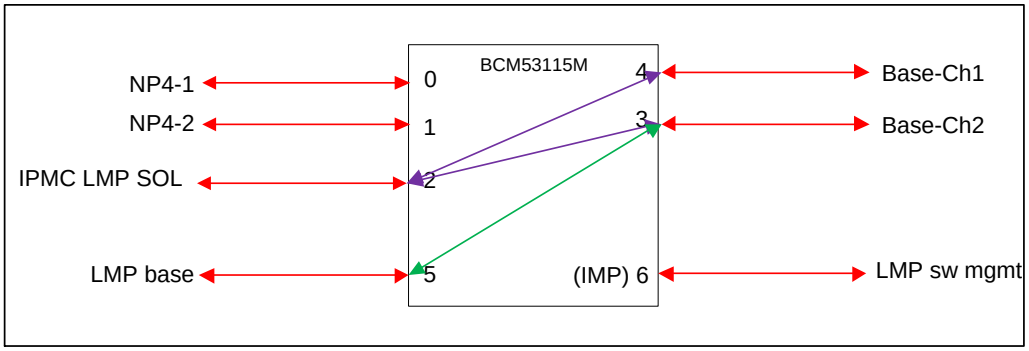


Figure 8. Base switch SOL connections during DYMANIC boot from BI1



The SOL encapsulated traffic from the switch port connected to IPMC will flow to the base port BI0 or BI1 based on mac address learning, in similar fashion supported in assets of AB3 program.

After booting, SOL connectivity can be established by setting a base switch configuration that enables the BI and SOL ports and sets the egress port filters to enable egress between these ports.

Table 7. Base switch egress settings for SOL

Port	Egress Port List (added for SOL)
0/1	4/2
0/2	4/2
4/2	0/1, 0/2

2.1.3 Software Interfaces

2.1.3.1 Command Line Interface

The standard Radisys Command Line Interface is used as the basis of the CLI for the ANPI1-A. This CLI is documented in [1]. Changes/additions to this CLI needed to meet NSN requirements for the ANPI1-A and a summary of some commands are given in Section 3 below.

Configuration changes made in the Radisys CLI take effect immediately. It is not necessary to restart the blade for them to take effect.

2.1.3.2 SNMP Interface

The standard Radisys SNMP interface is used as the basis of the ANPI1-A SNMP implementation. Radisys' standard version of SNMP is documented in Chapter 7 of [2]. Changes/additions to the SNMP interface are given in Section 3 below.

Configuration changes made via the SNMP interface take effect immediately. It is not necessary to restart the blade for them to take effect.

2.1.3.3 SSH Support

The ANPI1-A supports the SSH protocol as a standard interface and is based on the open source code included in the Wind River Linux 4.3 package included in the blade's software.

2.2 Usage Scenarios

This section describes the typical use case for the ANPI1-A envisioned by NSN.

2.2.1 Traffic Flow Through the ANPI1-A

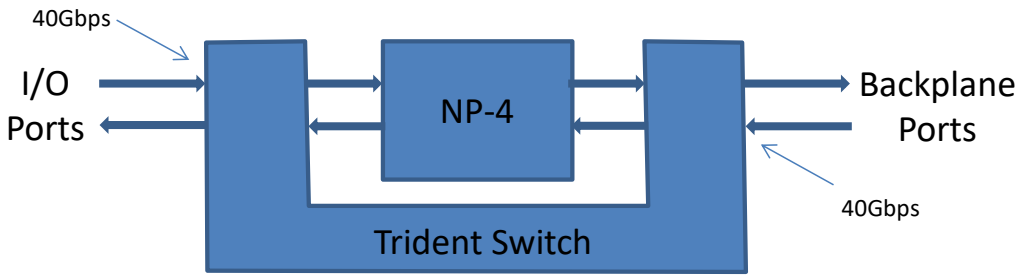
The typical traffic flow through the ANPI1-A is shown below in Figure 9. Traffic destined for one of the NP-4 processors flows into the blade through six I/O ports (typically on the RTM) in the ingress direction and comes from the 40Gbps backplane link in the egress direction. The actual throughput in both directions, however, is 40Gbps. Each NP-4 is connected to the fabric switch by ten 10Gbps links. This allows the net 80Gbps of traffic (the sum of the ingress and egress traffic) to be carried.

The total amount of bandwidth flowing through the Trident switch then is given by:

- 2x40Gbps Ingress (40Gbps for each NP-4)
- 2x40Gbps Egress
- 4x40Gbps Switch to/from NP-4

The above traffic results in a total of 320Gbps of traffic.

Figure 6. Traffic Flow Through ANPI1-A



3. Component Level Design

3.1 Platform Specific Functionality

3.1.1 Interconnection

There is no ANPI1-A specific implementation needed to meet these requirements.

3.1.2 Hardware Platform Management

3.1.2.1 Hardware Platform Management Functions

Requirements are covered in section 3.3.1.

3.1.3 Software Management and Upgrade

The ANPI1 software product consists of three major software components:

- LMP eSW – this consists of:
  - the U-Boot image,
  - Linux Images

    that are programmed on the SPI Flash (UBoot) and NAND flash (Linux). They could be programmed independently or together. FUMI interface provides independent way of flashing these images.
- IPMC eSW – this consists of the IPMC Application and IPMC Boot images that are programmed on the IPMC Flash.
- ANPI1 FRU Data – this consists of the FRU image file that is programmed on the FRU data storage attached to the IPMC.

The eSW for each of these entities is delivered as a separate tgz packages that are compliant to the Embedded SW Delivery Format Description for HW Platforms Specification [11]. The package will consist of the binary images and the Image Info File (.iif) file. The ANPI1 FRU data package will also include a fru_update configuration file that will enable the package version information to be written to the FRU Information.

The ANPI1 has two identical SPI Flash devices for the U-Boot images and two identical NAND devices for storage of Linux Kernel, Root File System, DTB and System Configuration. The layout of the devices and commands used to read/write the images is shown in Table 8. Each of the read-only partitions will also contain a checksum and version info that is accessible by the upgrade tools. Each of the flash partitions from both banks will be mounted when the Linux starts up and be accessible by the user. This will enable the user to access the running and standby banks from the running OS to read/write the versions and checksum info of all of the banks.

It should be noted that user has an option to modify the U-Boot env variables and to save them. Once they are saved, they persist not only over the LMP reset, but also over re-flash of the U-Boot images. In addition, a user can restore factory defaults of the U-Boot env variables since they are stored within the U-Boot image itself.

Table 8. LMP Boot Image Storage

Device	#	Name	Content	Size	Commands
Nand0 (1GB)	0	dtbA	DTB image	4MB	Nand
	1	kernelA	kernel image	8MB	Nand
	2	rfsA	RFS image	100MB	Nand
	-	-	Undefined	912MB	-
Nand1 (1GB)	0	dtbB	DTB image	4MB	Nand
	1	kernelB	kernel image	8MB	Nand
	2	rfsB	RFS image	100MB	Nand
	-	-	Undefined	912MB	-
SPI-flash0 (1MB)	0	u-boot	u-boot image	640KB	sf getenv, saveenv
		u-boot-env	Env. For u-boot	128KB	Sf
		(undefined)	-	256KB	Sf
SPI-flash1 (1MB)	1	u-boot	u-boot image	640KB	Sf
		u-boot-env	Env. For u-boot	128KB	Sf
		u-boot-env	Env. For u-boot	256KB	Sf

The nbm-upgrade utility is provided to perform local upgrades of the LMP eSW package from the Linux command prompt. Note that it is possible to upgrade any above mentioned programmable from any version to any other without the need to do intermediate upgrades using “force” option [AST873]. The usage of the nbm-upgrade utility is shown below:

```
Usage:
nbm-upgrade (primary|secondary|all) (u-boot|kernel/dtb/rfs) <Filename>
nbm-upgrade show
```

Detailed examples and usage of the tool will be provided in the user documentation. The LMP and NP-4 operation is not affected during the time the flash devices are being updated and verified. Before commencing with the upgrade, the nbm-upgrade tool verifies the checksum of the upgrade image in the bundle. If the checksum is not valid, an error message is displayed to the user and the upgrade is aborted. nbm-upgrade will verify the checksum of the image once it is written to the device and alert the user if the checksum fails. nbm-upgrade shall not perform any check on the file name NSN IIF file name or on the contents of the IIF file. The user can then try the command again. nbm-upgrade will also update the version information in the FRU data after the image is written correctly to the device. The version info will be stored in the corresponding “OEM SW and FW Version Information Record” defined in [16].Of course, the LMP will need to be reset for the newly upgraded images to take effect. nbm-upgrade does not automatically activate the firmware after it is programmed in the flash. This can be done as separate step later to enable all of the programmed components to be activated at the same time.

The upgrade of the IPMC eSW package is described in section 3.5.3.4. The LMP and NP-4 operation is not affected during the time the IPMC firmware is updated, verified and activated.

The remote upgrade of the eSW components is done through the HPI FUMI interface. The FUMI upgrade process is described in section 3.5.4.2.

The eSW Bundle delivered to NSN will contain binary versions of the all upgrade tools (i.e. nbm-upgrade, ipmitool, etc) that have been compiled and tested to run on the WR BSP on the LMP.

The tools will be used during the upgrade process to upgrade the images in the bundle. All upgrade tools will not force the upgrade of any component – the tool will enable the user to select which component needs to be upgraded. The upgrade tools will enable the upgrade of a component from any version to any other version (including the case when same version on the target and in the bundle).

eSW components should be backward compatible to previous (older) eSW versions in terms of existing settings, other eSW components, SW and HW interfaces. Therefore the activation of a new eSW version to a unit shall maintain the functionalities and settings of the previously running version. In no case after eSW upgrade shall the unit become unresponsive or fail to resume its previous functionality. Changes to the backward compatibility rules are acceptable prior to the NSN P6 milestone. After, P6 milestone every attempt should be made to ensure that newer versions of eSW are backward compatible with previously released versions to NSN. If Radisys is forced to break backward compatibility rules after P6 is achieved, then this should be negotiated with NSN to determine the best course of action to simplify the upgrade procedure. Backward compatibility will also be maintained with the upgrade tools. Older versions of the tools should support the upgrade of newer version so of the eSW bundle. If, for any reason, it is necessary to modify the eSW upgrade tools in a way that the user interface changes, such modifications shall be communicated and agreed with NSN beforehand. Radisys will not update the upgrade tools (after P6) without first getting agreement from NSN. With each delivery to NSN of a new eSW drop a release note shall be provided which lists all modifications and provides detailed statements on backward compatibility. In addition, each eSW bundle will also contain installation instructions for that bundle as well as instructions to install and upgrade the FRU data. The ANPI1 User Guide shall contain eSW installation and upgrading instructions.

The hardware version of the ANPI1 should be retrieved from the IPMC. The hardware version should be updated whenever the board hardware is updated. The upgrade tools should check the hardware version and detect any incompatibilities between the eSW and the hardware version of the ANPI1 and report it to the user.

The upgrade of all banks will take less than 15minutes to complete. This includes the erasing of flash and reprogramming of the flash and verification. It will be possible to upgrade the LMP firmware and IPMC firmware simultaneously so that the upgrade time is minimized.

3.1.4 Boot Performance

This section covers the compliance to AST505.

In agreement with NSN, the ANPI1-A will go from FRU activation to Login prompt in approx. 60 seconds. See current timing and improvements to be made in following table.

power-on
first console msg.

u-boot prompt

STATE	Details	current		after improvements(plan)		Remarks
		time(Sec.)	time(accumul.)	time(Sec.)	time(accumul.)	
IPMC & CPLD	Power Sequencing	1.5	1.5	1.5	1.5	
u-boot	init Devices	3.95	5.45	3.95	5.45	
	POST	3.7	9.15	0	5.45	

		Reading OS from NAND	25.77	34.92	4.5	**9.95**	current RFS size: 89MiB nand read speed: 4.7MiB/sec
linux prompt	**linux**	linux kernel starting-up	14.55	49.47	11	20.95	
		RC scripts	50.77	100.24	47	**67.95**	Broadcom SW init time: 38.39

3.2 General CPU Technology

There is no specific functionality implementation needed to meet the requirements for General CPU Technology.

3.3 Unit Computer

3.3.1 Boot

3.3.1.1 Overview

The primary purpose of U-Boot is to initialize the ANPI1 module hardware and jump to the Linux operating system. Secondarily, U-Boot provides the user with an environment suitable for debugging, changing of the flash contents, scripting, and downloading of more complex applications via the use of a serial console and an Ethernet connection.

U-Boot is comprised of several major components:

- Hardware initialization that initializes the modules memory controllers to access Flash and DRAM, and enables the console port.
- Debug and scripting capabilities allowing an interactive user to create and change environment variables, read and write memory and registers, and perform some basic functionality testing.
- Flash programming capabilities to allow saving and restoring of environment variables, reflash of U-Boot, Linux, JFFS2, and filesystem images.
- Tools for downloading and uploading images via TFTP, serial XMODEM, and serial binary.
- Advanced debugging capabilities allowing the user to download and execute specially packaged binary images. These images are in ELF format, and are specially manipulated using tools included in the U-Boot distribution.
- Scripting commands for verifying kernel image integrity and jumping into the kernel image.

The "U-Boot" Universal Boot loader project provides firmware with full source code under GPL. This is the preliminary boot loader for the P2041. It is used as a base for the board bring-up sequence.

The boot loader loads from flash. After the core initialization is complete, it pauses a configurable amount of time for operator input to halt the boot-up sequence. After entering the halt sequence the operator must enter a command within 10 minutes or the boot-up process continues by running

the default boot command. This typically involves loading the Linux operating system located on flash.

The default boot command is stored as an environment variable within U-Boot. All environment variables are stored in non-volatile flash memory. The "factory defaults" are included as a part of UBoot image and can be restored via command at the UBoot shell. [OS12].

When paused the bootloader has available a wide range of diagnostic and validation functions. Entering "help" at the command line causes the list of commands that are available to be displayed. There are a set of run-time diagnostics and hardware verification test code that may be executed post operating system boot-up.

Boot redundancy is supported by using two copies, one each in primary and secondary flash banks. If one image is corrupted, the LMP will boot from the other copy.

The bootloader supports loading of Linux BSP from Local flash as well as from a network source.

A user can interrupt the LMP boot sequence while in UBoot either via directly connected serial console or SOL connection. The UBoot commands and env variable settings are exercisable via SOL connection as well.

3.3.1.2 Dual Bootstrap Mechanism

Each module in the ANPI1 includes two identical SPI Flash memory devices for storage of U-Boot images and two identical NAND flash devices for storage of Linux images. The Local Management Processor (LMP) of the module may boot from either SPI device. If the boot process fails using the selected device, the IPMC will force the LMP to retry the boot process, using the secondary Flash memory device. A detailed description of this process is given below. By default, the Primary Flash Device (Flash device 0) is used to boot the module.

The redundant flash devices are provided to mitigate the following potential failures.

- Physical failure of flash device in U-Boot binary image. Such a failure could be caused by a hardware fault to the primary physical flash device, by a programming failure during reflash, or by an erroneous write to the binary flash image.
- Failure of the Linux kernel image integrity test. Prior to jumping into the Linux kernel, U-Boot performs a checksum test of the Linux kernel binary image. If the image fails this test, the LMP requests that the IPMC reset the payload and boot from the secondary flash device.
- Failure during an update process. If module power is interrupted during a reflash operation, the U-Boot or Linux image could be corrupted. This would be detected and recovery would proceed as described above.

It is important to note that the dual bootstrap devices do not mitigate failures caused by application failures or elongated boot times. Those failures are mitigated via the BMC watchdog described in 3.3 below.

3.3.1.3 U-boot Flash Update Functionality

This section documents the implementation of CETH141.

The U-Boot package included with the ANPI1-A supports the following commands for erasing and updating the flash memory. This approach to updating the flash can be used when there is no functional software in the flash part.

Table 9. U-boot Commands for Erasing/Programming Flash

CLI Commands	Comments
cp.[bwl]	Copies data from memory into flash
md.[bwl]	Dumps contents of flash
Erase	Erases flash

3.3.1.4 Boot Sequence

The general ANPI1 booting sequence is shown in Figure 10 . The functions executed in each step are described in Table 10. During the execution of the boot sequence, all errors printed to the console and to the SEL as progress events. In addition, the errors will be logged to a reserved area of RAM. The memory location and the format of the data in this area will be provided in the ANPI user documentation.

Figure 7. Booting Sequence

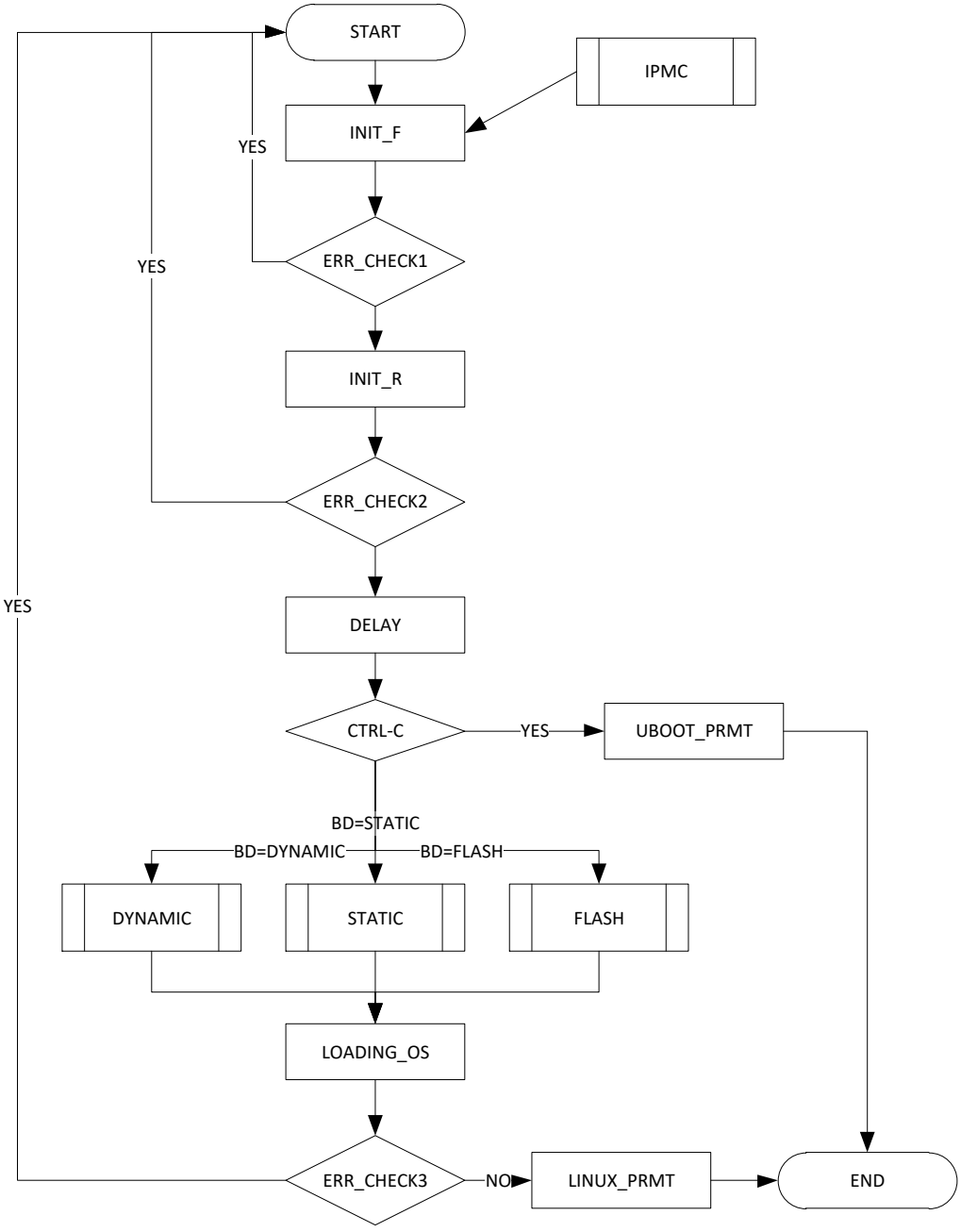


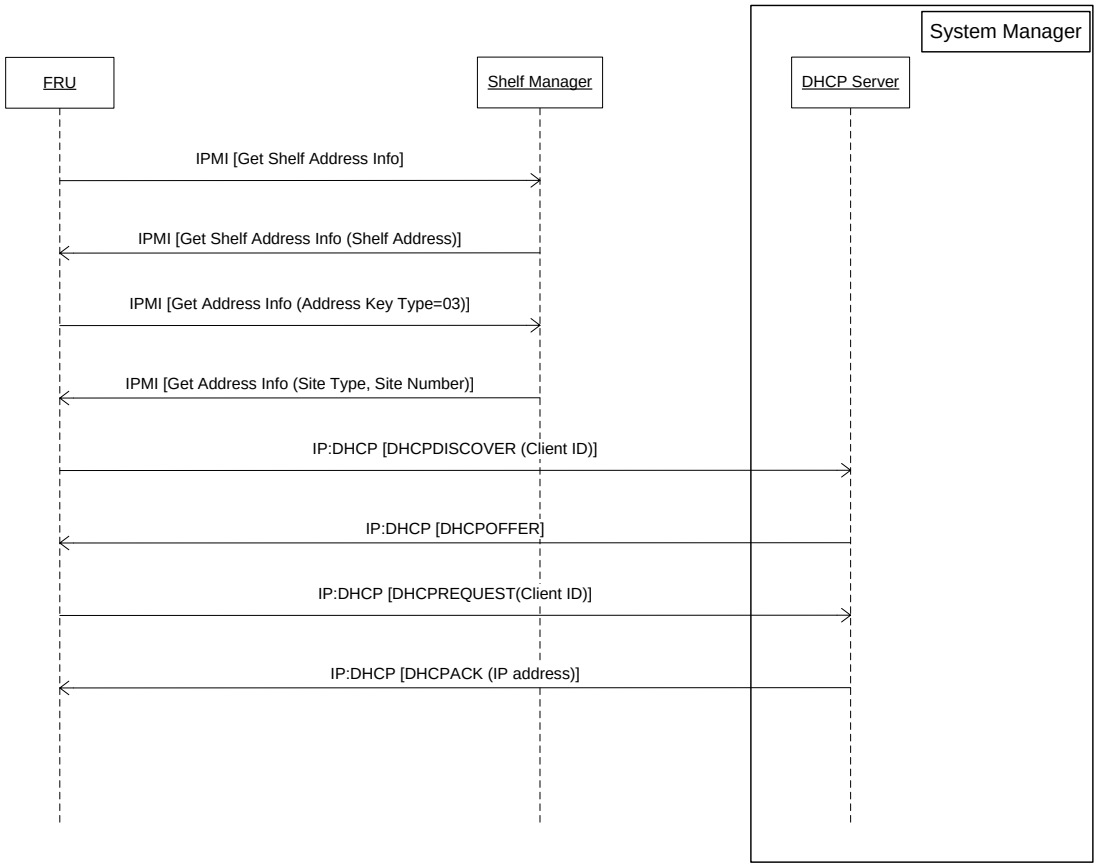
Table 10. Boot Process Description

Process	Function	Description
INIT_F	LAW setting TLB setting UART initialization DRAM initialization Relocation DRAM test	Preprocessing of U-boot before running on the DRAM The IPMC selects which u-boot image to boot from two redundant SPI flash memories.
ERR_CHECK1	Error checking during INIT_F	Two types of errors are inspected, one with U-boot recognizable error (e.g. DRAM test error during INIT_F) and another one with uboot unrecognizable error (e.g. broken u-boot image in the SPI FLASH). U-boot unrecognizable errors are identified by the IPMC CFD watchdog. U-boot should disable CFD watchdog if it concludes that there's no error identified.
INIT_R	BMC watchdog setting DEVICE initialization POST execution	Base switch is initialized and configured for SOL. PHY device access interface is provided.
ERR_CHECK2	Error checking during INIT_R	Two types of errors are inspected, one with U-boot recognizable error (e.g. POST test error during INIT_F) and another one with u-boot unrecognizable error (e.g. program hanging due to a bug. The BMC watchdog is enabled by early INIT_R procedure. The IPMC afterwards, restart u-boot unless it receives the watchdog disable command within a specified time window.
DELAY	Delay for n Seconds and poll for console input	Add a delay to enable the user to enter CTRL-C to enter the u-boot console. If the timer expires without any user input then, the u-boot reads an environment variable to determine the boot device to use to boot the LMP. There are three possible boot devices listed in Table 12.
DYNAMIC	Boot Device = Dynamic Network boot	U-Boot executes the following algorithm to perform network boot. U-Boot configures the local switch so that it has connection to BI0 network. U-Boot performs "Get Shelf Address Info" and "Get Address Info" IPMI commands to obtain location information for the option code 61. The format of ANPI1 DHCP Option 61 is shown in 3.3.1.5 U-boot sends DHCPDISCOVER to the System Manager via selected Base Interface If there is no reply from the server U-Boot will modify local switch configuration so that connection to BI1 is established.

Process	Function	Description
		Step 4 is repeated until valid DHCPOFFER is received from either BI network Client ID contains: Plug-In Unit identifier coded for the ANPI1 blade Blade location in the cabinet and shelf System Manager provides IP address, “next-server” and “filename” parameters in the DHCPACK. (Note: in some cases option codes could be used instead of those parameters. Server may provide also other additional options to the client.) U-Boot checks if it has received valid filename and next-server address. U-Boot fetches the file from the System Manager using TFTP U-boot checks that file has not corrupted during transfer and executes it. When executed the script downloads all the content needed for the blade net boot. [The message exchange between u-boot and Shelf Manager, and u-boot and system manager is shown inFigure 11]
STATIC	Boot Device = Static Network Boot	U-Boot uses static IP addressing (i.e. configured through console and saved in env variables) for TFTP server. The files to load from the server is also configured and saved in env variables.
FLASH	Boot Device = Local Flash	U-Boot selects Linux images from active NAND flash device
UBOOT_PRMT	U-boot command prompt	Disables the BMC watchdog.
LOADING_OS	DTB image loading Linux kernel image loading Root File System image loading	Image integrity check is performance by Checksum and image file type inspection.
ERR_CHECK3	Error checking during LOADING_OS execution	Two types of errors are inspected, one with U-boot recognizable error (e.g. kernel image checksum error) and another one with U-boot unrecognizable error (e.g. linux-kernel panic). The BMC watchdog is enabled by early INIT_R procedure. The IPMC afterwards, restart u-boot unless it receives the watchdog disable command within a specified time window.
LINUX_PRMT	Linux command prompt	Disables the BMC watchdog.

Process	Function	Description
IPMC	CFD Watchdog execution BMC Watchdog execution	Upon the Watchdog timeout, the system is restarted with an option to switch SPI FLASH0/1. If the boot flash is corrupt or missing, the CFD watchdog will expire and switch the boot to alternate bank.

Figure 8. Dynamic Boot Sequence

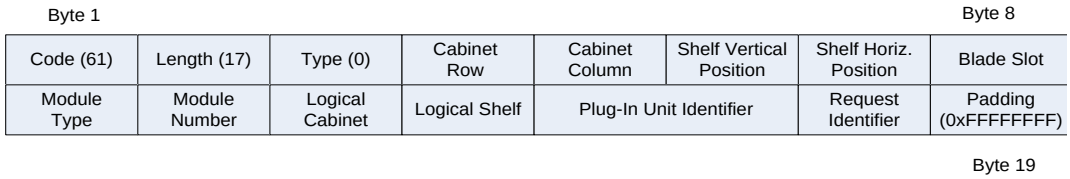


The behavior for the network boot will not require any interaction while in UBoot. If the "Dynamic Network boot" is set as the default boot device, or it is the next selection in the boot device order, the base switch port will be configured automatically to enable the correct datapath for reaching out to the right Server handling the Linux boot images. For the DHCP servers that can be reached via BI0 or BI1 interfaces, the switch will be configured so that UBoot executable code will attempt to reach first server via BI0 (via configuring base switch ingress port mapping accordingly allowing traffic only from the LMP port to BI0 port),and then after programmed timeout, via BI1 (via now configuring base switch ingress port mapping accordingly allowing traffic only from the LMP port to BI1 port).

3.3.1.5 DHCP Option 61 Format

Following figure describes the format of NSN specific DHCP Option 61

Figure 9. Format of NSN specific DHCP Option 61



The Length of the whole Option 61 is 19 bytes. The meaning of each field of the Option 61 is in Table 11. It also tells how certain information can be derived to be filled into the fields.

Table 11. Content of NSN Specific Option 61

Byte	Field	Description	Type	Range	Default value	Data Source
1	Code	Option code	Decimal	61	61	Fixed
2	Length	Length of the Option code	Decimal	17	17	Fixed
3	Type	Type of the Code	Decimal	0	0	Fixed
4	Cabinet Row	Cabinet row location in the central office. Number is one of the coordinates used to indicate where the cabinet is installed in the central office.	Hex	00..FEh	FFh	Shelf Address Offset 2
5	Cabinet Column	Cabinet column in the central office. Number is the second coordinate used to indicate where the cabinet is installed in the central office.	ASCII	A…Z	FFh	Shelf Address Offset 3
6	Shelf Vertical Position	Vertical position of the shelf in the cabinet	Hex	00..FEh	FFh	Shelf Address Offset 0
7	Shelf Horizontal Position	Horizontal position of the shelf in the cabinet.	Hex	00..FEh	FFh	Shelf Address Offset 1
8	Blade Slot	Contains the slot of the blade in the shelf. In case of AMC carrier this contains the slot of the carrier.	Hex	00..FFh	FFh	From Get Address Info command response.
9	Module Type	Contains Site Type value as defined in PICMG 3.0	Hex	00..FFh	07h	Received in Get Address Info response.
10	Module Number	Contains Site Number as defined in PICMG 3.0. In case of AMC contains Site ID value as defined in AMC.0.	Hex	01..1Ah	FFh	Does not apply to ANPI1. Set to default.
11	Logical Cabinet	Defines the logical cabinet number	Hex	00..FFh	FFh	Shelf Address Offset 4
12	Logical Shelf	Defines the logical shelf number in the cabinet	Hex	00..FFh	FFh	Shelf Address Offset 5

Byte	Field	Description	Type	Range	Default value	Data Source
13-14	Plug-in Unit Identifier	Used to identify the type of the FRU.	Hex	0000…FFFFh	FFFFh	Local FRU Information (3).
15	Request Identifier	Indicates the processor or interface inside the module that requests an IP address.	Hex	00..FEh	FFh	Set to 0x00 for LMP
16-19	Padding	Padding field to be compliant with Option 61 in PXE Boot.	Hex	FFFFFFFFh	FFFFFFFFh	

3.3.1.6 Boot Devices and Configurable Options

The ANPI1 implementation of booting the LMP to an operating system is based on software entities called boot devices. Boot devices are software entities that define where an operating system will be found and how it will be retrieved. For ANPI1 the following boot device types are supported.

Table 12. Boot Device Types

Default Boot Order	Boot Device	Description
1	Local Flash	This boot device attempts to load the “active” image from flash. Radisys blades have two flash banks where the currently selected flash bank is called the “active” flash bank.
2	Dynamic Network	This boot device dynamically retrieves and loads an image based on a DHCP request and response. The image name, server IP address, and board IP address are provided by the DHCP server. The image is retrieved using the TFTP protocol.
3	Static Network	This boot process statically retrieves and loads an image based on statically defined U-Boot environment variables. These variables include the server IP address, local IP address, and image name. The image is retrieved using the TFTP protocol.

Boot devices are managed using an environment variable that represents a list of boot device types. The order and selection of boot devices can be changed. When U-Boot needs to boot an image, it will step through each boot device and attempt to boot from that device. The number of network retries can be configured. Ethernet devices (i.e. eTSEC0, eTSEC1, bmd0, etc) are managed using environment variables.

Several elements of the U-Boot Boot Device process are configurable to provide customers with more flexibility in their implementation. All configurable options are managed as U-Boot environment variables. The environment variables are stored persistently in a separate section of the U-Boot Flash memory.

Currently, environment variables can be changed by accessing U-Boot through the serial console, or by using the Linux applications “fw_printenv” and “fw_setenv”. These applications are included as part of Radisys Linux distributions.

The following table describes all of the boot process configuration options.

Table 13. Boot Process Configuration Options

Configuration Option	Variable	Factory Default
Set the Board Type (**note:** *This field is set by U-Boot code and should not be changed.*)	boardtype	<blade-specific> (i.e. ANPI1)
Sets the number of DHCP, ARP, or TFTP retries that should be performed before moving on to the next boot device. One iteration includes all of the Ethernet devices in the “ethlist” variable.	netretrymax	3 (Set to zero for infinite loops.)
If all boot devices fail, this flag indicates whether or not U-Boot should step through the devices again or if it should go to the prompt. This could be used to try to network boot forever.	bootdevforever	y
Sets whether or not the board should switch to the secondary flash bank if a boot image is found to be invalid.	noswitchflash	n
Indicates the board’s boot device list. The list is separated by colons. For example, the list could be “bootlist=dynamic:static”. Three options are supported: “dynamic”, “static”, and “flash”.	bootlist	dynamic
Indicates the board’s Ethernet list. The list is separated by colons. For example, the list could be “ethlist=bmd0:eTSEC1”. The Ethernet list will vary based on board type, but can include the TSEC devices (i.e. eTSECx) and the Broadcom devices (i.e. bmdx) if supported.	ethlist	bmd0
Indicates which ports to use on Ethernet device bmd0. This setting only applies to sending packets. The order that the ports are used depends on the order of the list.	bmd0portlist	bi0:bi1
Sets the static local IP address	staticipaddr	10.0.0.1
Sets the static server IP address	staticserverip	10.0.0.2
Sets the static image name to retrieve	staticbootfile	<blade-specific> (i.e. atca1200.bin)
Sets the local MAC address for the board. The MAC address is assigned to each interface when the interface is used to send packets.	ethaddr	<varies> (i.e. 00:E0:0C:00:00:FD)
The current number of unsuccessful boots in a row	bootcount	<varies> This option is automatically updated on every boot to reflect the current count.

Table 13. Boot Process Configuration Options

Configuration Option	Variable	Factory Default
The limit of unsuccessful boots in a row before the alternate boot command will be used.	bootlimit	<unset> This option is not set by default. By default the board loads from local flash memory, and on a boot failure the IPMC will switch the flash bank.
The alternate boot command to use in case of repeated boot failures.	altbootcmd	<unset> This option is not set by default. By default the board loads from local flash memory, and on a boot failure the IPMC will switch the flash bank. After this, UBoot will use the primary boot command from that flash bank.

3.3.1.7 POST

The Power On Self Test (POST) is performed during bootloader booting up. The POST includes the major devices on the board. The POST result is printed out on the monitoring port at the end of the boot loader booting. These post results are also logged to the SEL as events against the POST Sensor described in 3.5.3.1. Table 14 summarizes the devices, the testing and the codes.

Table 14. ANPI1 POST Codes

No.	POST device ID	Description
1	CONFIG_SYS_POST_MEMORY	LMP DDR DRAM memory testing. The specified memory region is tested with Read/Write.
2	CONFIG_SYS_POST_IPMC	Sample IPM commands are executed to see if it is operational.
3	CONFIG_SYS_POST_SPI_EEPROM	spi flash probe is executed to check if SPI eeprom is operational.
4	CONFIG_SYS_POST_MGMT_PHY	The PHY access through MII is performed to check if it is operational.
5	CONFIG_SYS_POST_RTC	RTC device is verified for time of the day operation.
6	CONFIG_SYS_POST_EEPROM	The EEPROM is checked by I2C probe to see if it is operational.
7	CONFIG_SYS_POST_NAND_FPGA	The NAND fpga is checked by read/write operation to check if it is operational.
8	CONFIG_SYS_POST_NAND_FLASH	The post of NAND flash will pass if init process passed.

No.	POST device ID	Description
9	CONFIG_SYS_POST_CPLD	The CPLD is checked by read/write operation to check if it is operational.
10	CONFIG_SYS_POST_SWITCH_BASE	The SWITCH (base) access through SPI is performed to check if it is operational.
11	CONFIG_SYS_POST_SWITCH_FAB	The SWITCH (fabric) access through PCI is performed to check if it is operational.
12	CONFIG_SYS_POST_POWER_CTRL	The Power controller is checked by I2C probe to see if it is operational.
13	CONFIG_SYS_POST_SYNC_IN	The SYNC (in) is checked by I2C probe to see if it is operational.
14	CONFIG_SYS_POST_NP_TOP	The NP (TOP) access through PCI is performed to check if it is operational.
15	CONFIG_SYS_POST_NP_BOT	The NP (BOT) access through PCI is performed to check if it is operational.
16	CONFIG_SYS_POST_I2C_MUX_0	The I2C Muxes are checked by I2C probe to see if it is operational.
17	CONFIG_SYS_POST_I2C_MUX_1	The I2C Muxes are checked by I2C probe to see if it is operational.
18	CONFIG_SYS_POST_I2C_MUX_2	The I2C Muxes are checked by I2C probe to see if it is operational.
19	CONFIG_SYS_POST_IO2I2C_INS	IO buffer device is verified by probing.
20	CONFIG_SYS_POST_IO2I2C_TXF	IO buffer device is verified by probing.
21	CONFIG_SYS_POST_IO2I2C_LOS	IO buffer device is verified by probing.
22	CONFIG_SYS_POST_IO2I2C_TXDIS	IO buffer device is verified by probing.

3.3.2 Unit Computer Operating System – Linux support for LMP P2041

The Linux image is based on Wind River PNE-LE 4.3. A BSP will be provided which compiles most of the software provided in the binary image. The software image contains the Fastpath software which is a proprietary software package from Broadcom. Radisys does not have the rights to distribute the source code for the Fastpath software [OS82] [OS117]. The base module supports JFFS2, NFS & RAM file systems (tmpfs). The JFFS2 partition reserved for binaries, libraries and default config is 100MB (minimum). NTP package is included with Radisys WR Linux and is used to synchronize to its clock [OS99].

Radisys will distribute eSW in a format for embedded blades LMP (more like a black box use case), not in the format of Wind River development environment. The Linux image from Radisys contains proprietary software that cannot be distributed under the GPL. [OS32]. All Linux kernel modifications and device driver source, with the exception of the Broadcom BDE driver, will be released under GPL. The Broadcom BDE driver source will be released under a proprietary license, and according to the existing agreement with NSN.

Access to Linux shell is provided via the serial port and the telnet server and/or SSH daemon [OS100]. The Linux can be accessed via top level CLI. All configuration and monitoring utilities can be run from Linux shell script, except switch configuration we must be done via Radisys switch CLI [OS98].

General management tools are mostly command line tools that meet the CLI requirements. Not all BSP commands are available through SNMP. [OS88]

The default tool for managing Linux system logs is syslog-ng. Syslog-ng is configured to log information to a local file by default, but has capability to be configured to log to a remote system, if desired. Users should consult the syslog-ng documentation for how to configure this operation [OS101][OS56].

All of the logs on the blade are self-limiting except. Radisys added a configuration file to the blade that configures logrotate to monitor the syslog file. It limits the size of the file to 500k. When the file reaches the 500k limit, it backs up the log file to a compressed backup copy. It will keep a maximum of 5 backups before deleting the oldest copy. A crontab entry is added to the ANPI-1 software that will check on the file size every 5 minutes. If the limit has been reached, the log will be rotated.

The Linux image will include device drivers which support all hardware blocks and system devices of the blade [OS19][OS21]. These drivers will be automatically loaded by initialization scripts upon booting of the Linux OS [OS22].

The Linux image will use freely available open-source drivers and interfaces, including kernel drivers and patches, except where explicitly stated otherwise (see a note for OS82 ANPI1-A Requirements Matrix). For example, the BDE driver used by switchdrvr is not GPL, and will taint the kernel [OS85].

The Ethernet drivers used for configuring LMP Ethernet port will be open source and will provide all available functionality through standard APIs and interfaces. [OS25] The open source drivers expose all interface as standard network devices.[OS57] The open source drivers provide only one device per port [OS58].The setting for registers for Ethernet devoices are exposed by the open source drivers and can be used by Radisys customers for tuning the settings. This is available through open source tool “ethtool”. [OS59]

All software components (kernel, kernel modules, userspace components) contains an explicit version number.[OS43] All device drivers and the Linux kernel have a version number stamped into the binaries [OS44].

Many of the packages included in the Linux BSP are not compiled by Radisys, and are taken as binaries straight from WindRiver. In order to track the version info of deliver tools and binaries, all utilities provided on the blade, as well as device drivers and the Linux kernel, have their version information stored in “/etc/versions”.[OS87]

The Linux image will support the kdump mechanism.

The Radisys Linux board support package does not include symbols to allow debugging with kgdb [OS55]. However, Radisys can provide Linux kernel image with debug symbols on upn customer requests, if a field issue arises.

3.3.2.1 LMP eSW image component

The LMP software binary package includes:

- U-Boot bootloader
- Linux BSP including device drivers
- Blade Management Software
- Datapath Management Software
- Utilities

The ANPI1 blade can be managed through APIs as well as through a CLI. The management API is defined in Radisys proprietary common blade management software spec and it can be available upon request.

The CLI allows management of the following ANPI1 subsystems:

- Blade health & logging
- FASTPATH CLI for Ethernet switches
- DPB control (Details in later sections)
- Network Timing
- Software Image
- System services
- SNMP
- File system management

3.3.2.2 Software Upgrade

The P2041 LMP image will be fully upgradeable using FUMI support through HPI. FUMI is able to selectively upgrade components only as needed, and is able to update backup banks while running from the active bank. The SW version of each eSW component will be retrievable from the Linux OS generated from the BSP. [OS33]

The upgrade of ANPI-1 software components process for a single eSW component is activated with a single command.

3.4 Data Plane Processing and Transport

3.4.1 NPU Specific Requirements

This section covers the design of the implementation for requirement DPP81, which covers a range of NPU requirements. NP-4 core clock is set to 365Mhz, the maximum core clock frequency based on the NP-4 data sheet. DPP101, which is a sub-section under DPP81 requires Radisys to provide boot configuration in NPSL format. This configuration is expected to be used for test purposes only. NSN OS, platform layer is responsible for final (complete) configuration of the NP-4, including both the microcode and TM sections. The remainder of this section describes the current mechanism used by Radisys to configure the NP-4. The NP-4 is currently configured using the following steps:

a. Manual load of Ezlkm_linux_ppc.ko driver module

b. Execution of stand-alone NP-4 application – /usr/np-4/Ezware_linux_ppc

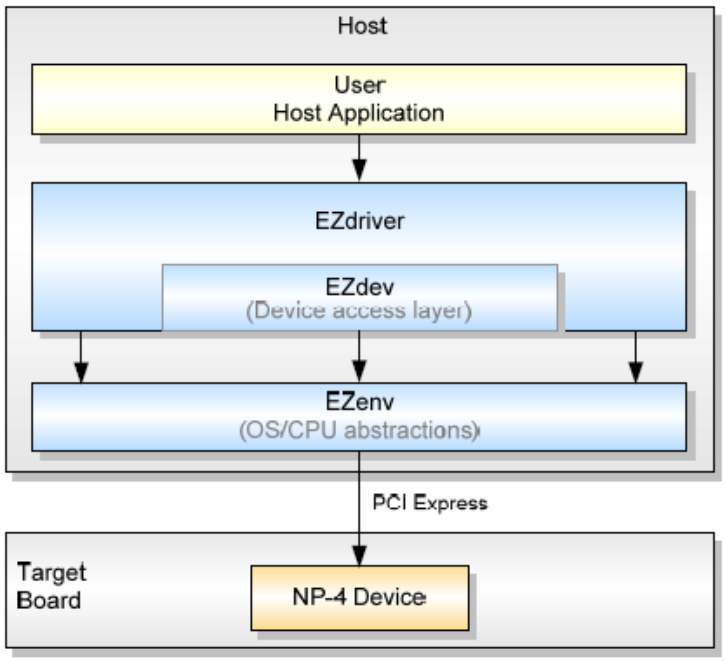
c. Loading and execution of an ASCII NPSL script using the NP-4 specific CLI presented by the stand-alone application. The NPSL scripts supplied by Radisys is for performing I&V testing.

The NPSL script supplied by Radisys is to be used for test purposes. We expect the custom application that implements the L3/MPLS forwarding along with the required traffic management to be supplied by NSN.

3.4.2 EZDriver operation

The following figure describes the operation of the EZDriver library, which provides an API between the host software (running on LMP) and the NP-4 network processors.

Figure 10. EZDriver operation



Application developers are expected to use this API library for communication between host software and NPUs. API includes:

- Configuration of NP-4 device TOPs
- Configuration of network interfaces
- Configuration of traffic manager
- Creating and maintaining of lookup structures
- TX/RX of frames to/from the NPUs

3.4.3 NPU Reset support

HW supports independent reset lines to the NPU that are controlled by a CPLD. A change will be implemented in CPLD to support individual resets to NPU. Also need to be added an OEM command in IPMC to support the individual resets to NPU.

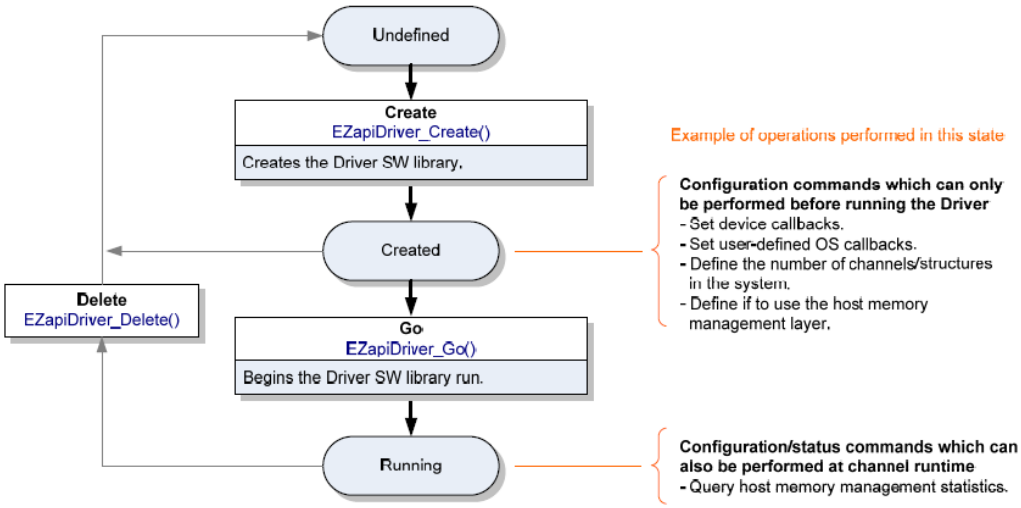
ANPI-1 software is supporting the following cold resets of the LMP and NPUs:

1. Joined cold reset of the LMP and NP-4s
2. Independent cold reset of the LMP only
3. Independent cold reset of each of the NP-4s

3.4.4 Driver/Channel Group APIs

As part of driver initialization, the set of driver API calls (represented by EZapi_Driver_Create, EZapi_Driver_Go etc) are invoked in the following sequence. The complete list of driver group API calls is available in section 2 of the EZDriver (versin #8.46) reference manual.

Figure 11. Sequence of Driver states/API

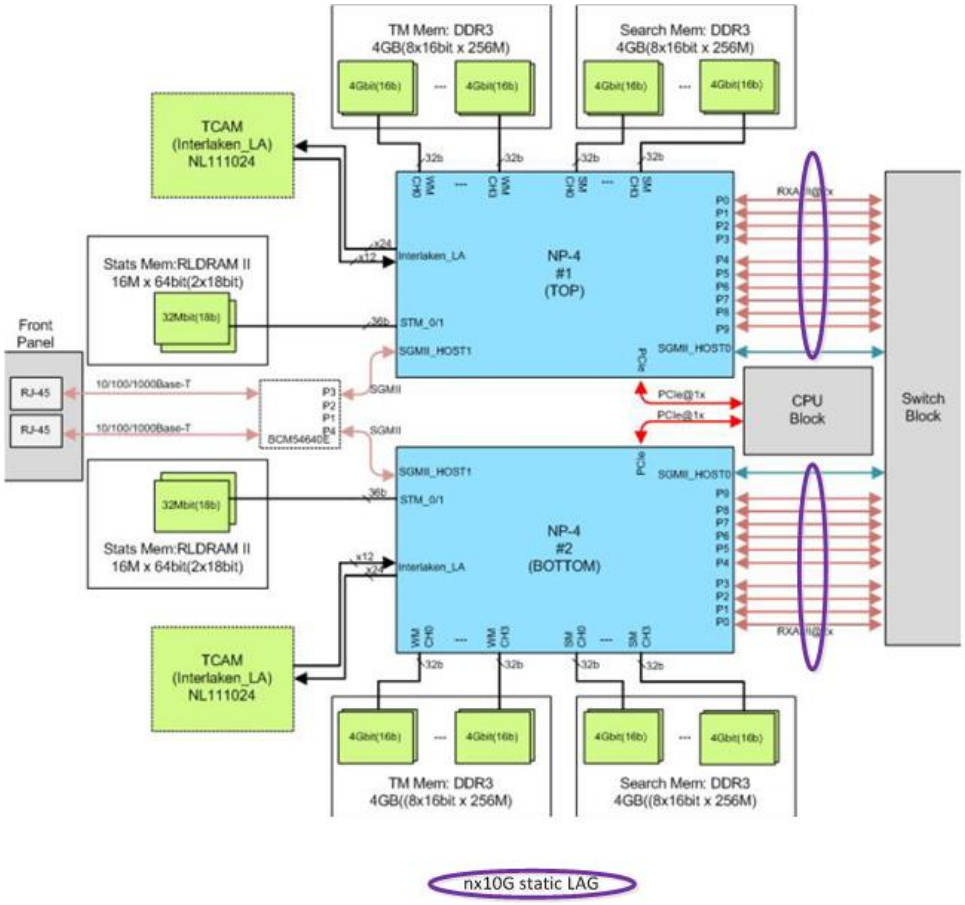


Each NPU is considered to be a channel by EZDriver and there exists a channel object representing each NP-4 in the system. Channel group API routines enable initializing the channel, initial loading microcode to TOPS , configuration of network interfaces and message queue sizes. Examples of channel group APIs include EZapiChannel_Initialize, EZapiChannel_Finalize, EZapiChannel_Go as described in section 3 of the EZDriver reference manual.

3.4.5 Link Aggregation Group between Switch and NPUs

DPP329 covers the use of the static 10x10G LAGs between the NPU and Trident+ switch. Two such LAGs are created to support the 2 NP-4s as shown in the figure below.

Figure 12. Static LAGs between NP-4 and Trident+ switch



Since the Trident+ switch does not perform native parsing of GRE keys or GTP-TEID fields, the LAGs are expected to use the packet's source, destination IP addresses as hash fields to select the physical link to transmit the packet. LAGs are created using the following example sequence of commands for 2 ports per LAG. These commands are similar to those used to create static LAGs on AB3 blades.

```
config
interface 1/1
addport 4/1
exit
interface 1/2
addport 4/1
exit

interface 1/3
addport 4/2
exit
interface 1/4
addport 4/2
exit
exit
```

Use the show port-channel all command to list the results.

```
show port-channel all

        Port-
Log.    Channel         Adm.          Mbr     Device/       Port    Port
Intf     Name    Link   Mode   Type  Ports    Timeout      Speed   Active
-----  -------  -----  ----  -----  -----  ------------   ------  -------
4/1    lag_1    Down    En.   Dyn.   1/1   actor/short     Auto    False
                                            partner/short
                                     1/2   actor/short     Auto    False
                                            partner/short
4/2    lag_2    Down    En.   Dyn.   1/3   actor/short     Auto    False
                                            partner/short
                                     1/4   actor/short     Auto    False
                                            partner/short
```

The hashing mode is expected to be set using the command:

port-channel load-balance <load-balancing_option> {<slot/port> | all}

where the load-balancing_option chosen is 6. This mode utilizes the Source/Destination IP and Source/Destination TCP/UDP Port fields of the packets as inputs to the LAG hash function. See section 3.6.3.11 for more information on port-channel (LAG) commands.

3.5 Hardware Platform Management

3.5.1 Hardware Platform Management Standards

The ANPI1 is equipped with the ATCA compliant chassis management capabilities which allow the blade to be accessible by a Shelf Manager for monitoring and controlling the blade.

The Intelligent Management Controller (IPMC) sits at the center of the blade management architecture. The IPMC enables Hot-swap, E-keying, Power management, Environment/Device monitoring and management with or without the ATCA Shelf Manager. The IPMC is initialized as soon as the dedicated power is supplied, and works completely independent of the LMP. Since the

dedicated power is separate from the main power for other devices, the IPMC can work when the power to the other devices are not supplied.

The IPMC has embedded flash memory, RAM, and System Event Log memory and multiple I2C bus interfaces for interconnection to other devices. The firmware for the IPMC is saved into the flash. The serial port connected to the IPMC enables access to all management information and commands.

The Field Replacement Unit (FRU) information is saved into the EEPROM, which is connected to the IPMC through an I2C interface. Two I2C buses are provided as a redundant connection for the IPMB bus to the backplane to communicate with the Shelf Manager. One I2C bus is reserved for the RTM MMC connection. The remaining I2C bus is used to monitor the sensors, the power modules, and to read from or write to the EEPROM in master-only mode. During monitoring, when one of the monitored values goes over the threshold, the IPMC reports to the Shelf Manager for the event.

When a fault is detected on one of the redundant IPMB buses, the IPMC isolates the troublesome bus and re-establishes the communication path again on the new IPMB bus.

The IPMC is compliant to the Intelligent Platform Management Bus Communications Protocol Specification v1.0. The IPMC is based on IPMI version 1.5 specification and is also compliant to parts of IPMI version 2.0 (i.e. Serial over LAN and RMCP+)

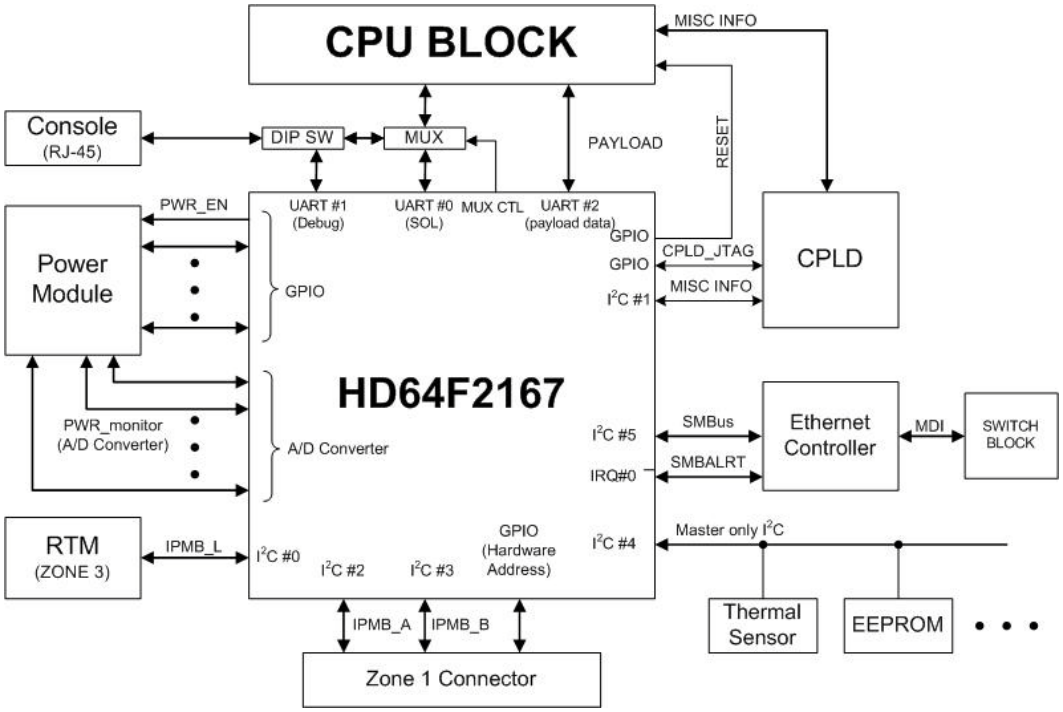
Note: The ANPI-1 FRU is not compliant to IPMI Platform Event Trap Format Specification v1.0.

3.5.2 Hardware Platform Management Functions

3.5.2.1 IPMC

Shown in Figure 16 is the structure of the IPMC block. The IPMC is HD64F2166, a 16-bit Single-Chip microcomputer from Renesas. The IPMC and the CPLD form the important functional block for the ATCA blade management task

Figure 13. IPMC HW Block Diagrams.



The IPMC block interconnections are summarized:

- Serial Over LAN (SOL): The HD64F2166 connects to the GB82571EB by SMBUS and the GB82571 is connected to the base switch. The LMP P2041 serial console connects to either the front panel RJ-45 console port or to the SOL port (UART #0), but not at the same time.
- Payload Interface: By UART #1 port, the IMPC communicates with the LMP.
- RTM Management: The IPMC communicates with the RTM MMC through the hot-swap buffer and IPMB-L for FRU information and Command exchange.
- IPMB-A/B Interface: The IPMC communicates with the Shelf Manager via the redundant IPMB-A and IPMB-B interfaces on ZONE 1 connector through the hot-swap buffer.
- Blade Power Sequence: All the power rail enable/disables are controlled by the IPMC GPIO.
- Power Monitoring: All the power rails are monitored by the IPMC and any violation on the level is reported to the Shelf Manager. When a violation is observed, the IPMC shutdowns the FRU.
- FRU Information: The EEPROM that contains the FRU information, Log, alarm information, etc is connected to the IPMC via the master only I2C.
- Sensor: Various sensors including thermal and power are connected to the IPMC via the master only I2C. The IPMC monitors sensor information.

3.5.2.2 IPMI Commands

3.5.2.2.1 Network Function Codes

The table below gives the values for each of the Network Function codes of the IPMI commands. The column on the right gives the value of the 6bit/2bit combined NetFn/LUN0 byte for use over the serial interface.

Table 15. NetFn Values

NetFn	Value	NetFn / LUN0
App	06h	18h
Firmware	08h	20h
S/E	04h	10h
Storage	0Ah	28h
Transport	0Ch	30h
PICMG	2Ch	B0h
OEM Group (Radisys IANA*)	2Eh (F1h 10h 00h)	B8h
OEM1	30h	C0h

* IANA bytes are sent as the first three data bytes of OEM Group IPMI request messages to identify the manufacturer that defined the command.

3.5.2.2.2 Standard IPMI/ATCA Commands

The table below enumerates the standard IPMI/ATCA messages that are supported by the IPMC firmware. These commands are defined in the IPMI and ATCA specifications. Additional commands may be added as necessary.

As mentioned in chapter 3.5.1 The IPMC is compliant to the Intelligent Platform Management Bus Communications Protocol Specification v1.0. The IPMC is based on IPMI version 1.5 specification and is also compliant to parts of IPMI version 2.0 (i.e. Serial over LAN and RMCP+), including Table 29-6, Event Request Message Event Data Field Contentsv1.0.

Table 16. Supported Standard IPMI and ATCA Commands

Message Commands	NetFn (NetFn / LUN0)	Cmd
Get Device ID	App - 06h (18h)	01h
Cold Reset	App - 06h (18h)	02h
Warm Reset	App - 06h (18h)	03h
Get Self-Test Results	App - 06h (18h)	04h
Set ACPI Power State	App - 06h (18h)	06h
Get ACPI Power State	App - 06h (18h)	07h
Get Device GUID	App – 06h (18h)	08h
Reset Watchdog Timer	App - 06h (18h)	22h
Set Watchdog Timer	App - 06h (18h)	24h
Get Watchdog Timer	App - 06h (18h)	25h
Set BMC Global Enables	App - 06h (18h)	2Eh
Get BMC Global Enables	App - 06h (18h)	2Fh

Table 16. Supported Standard IPMI and ATCA Commands

Message Commands	NetFn (NetFn / LUN0)	Cmd
Clear Message Flags	App - 06h (18h)	30h
Get Message Flags	App - 06h (18h)	31h
Get Message	App - 06h (18h)	33h
Send Message	App - 06h (18h)	34h
Get Channel Authentication Capabilities	App - 06h (18h)	38h
Get Session Challenge	App - 06h (18h)	39h
Activate Session	App - 06h (18h)	3Ah
Set Session Privilege Level	App - 06h (18h)	3Bh
Close Session	App - 06h (18h)	3Ch
Get Session Info	App - 06h (18h)	3Dh
Set Channel Access	App - 06h (18h)	40h
Get Channel Access	App - 06h (18h)	41h
Get Channel Info	App - 06h (18h)	42h
Set User Access	App - 06h (18h)	43h
Get User Access	App - 06h (18h)	44h
Set User Name	App - 06h (18h)	45h
Get User Name	App - 06h (18h)	46h
Set User Password	App - 06h (18h)	47h
Activate Payload	App - 06h (18h)	48h
Deactivate Payload	App - 06h (18h)	49h
Get Payload Activation Status	App - 06h (18h)	4Ah
Get Payload Instance Info	App - 06h (18h)	4Bh
Set User Payload Access	App - 06h (18h)	4Ch
Get User Payload Access	App - 06h (18h)	4Dh
Get Channel Payload Support	App - 06h (18h)	4Eh
Get Channel Payload Version	App - 06h (18h)	4Fh
Get Channel OEM Payload Info	App - 06h (18h)	50h
Get Channel Cipher Suites	App - 06h (18h)	54h
Suspend/Resume Payload Encryption	App - 06h (18h)	55h
Set Channel Security Keys	App - 06h (18h)	56h
Get System Info	App - 06h (18h)	59h
Set Event Receiver	S/E - 04h (10h)	00h
Get Event Receiver	S/E - 04h (10h)	01h
Platform Event (a.k.a “Event Message”)	S/E - 04h (10h)	02h
Get Device SDR Info	S/E - 04h (10h)	20h
Get Device SDR	S/E - 04h (10h)	21h

Table 16. Supported Standard IPMI and ATCA Commands

Message Commands	NetFn (NetFn / LUN0)	Cmd
Reserve Device SDR Repository	S/E - 04h (10h)	22h
Set Sensor Hysteresis	S/E - 04h (10h)	24h
Get Sensor Hysteresis	S/E - 04h (10h)	25h
Set Sensor Thresholds	S/E - 04h (10h)	26h
Get Sensor Thresholds	S/E - 04h (10h)	27h
Set Sensor Event Enable	S/E - 04h (10h)	28h
Get Sensor Event Enable	S/E - 04h (10h)	29h
Re-arm Sensor Events	S/E - 04h (10h)	2Ah
Get Sensor Event Status	S/E – 04h (10h)	2Bh
Get Sensor Reading	S/E - 04h (10h)	2Dh
Get FRU Inventory Area Info	Storage - 0Ah (28h)	10h
Read FRU Data	Storage - 0Ah (28h)	11h
Write FRU Data	Storage - 0Ah (28h)	12h
Get SDR Repository Info	Storage - 0Ah (28h)	20h
Reserve SDR Repository	Storage - 0Ah (28h)	22h
Get SDR	Storage - 0Ah (28h)	23h
Enter SDR Repository Update Mode	Storage - 0Ah (28h)	2Ah
Exit SDR Repository Update Mode	Storage - 0Ah (28h)	2Bh
Get SEL Info	Storage - 0Ah (28h)	40h
Get SEL Allocation Info	Storage - 0Ah (28h)	41h
Reserve SEL	Storage - 0Ah (28h)	42h
Get SEL Entry	Storage - 0Ah (28h)	43h
Add SEL Entry	Storage - 0Ah (28h)	44h
Partial Add SEL Entry	Storage - 0Ah (28h)	45h
Delete SEL Entry	Storage - 0Ah (28h)	46h
Clear SEL	Storage - 0Ah (28h)	47h
Get SEL Time	Storage - 0Ah (28h)	48h
Set SEL Time	Storage - 0Ah (28h)	49h
Set LAN Configuration Parameters	Transport - 0Ch (30h)	01h
Get LAN Configuration Parameters	Transport - 0Ch (30h)	02h
Get IP/UDP/RMCP Statistics	Transport - 0Ch (30h)	04h
Set SOL Configuration Parameters	Transport - 0Ch (30h)	21h
Get SOL Configuration Parameters	Transport - 0Ch (30h)	22h
Get PICMG Properties	PICMG - 2Ch (B0h)	00h
Get Address Info	PICMG - 2Ch (B0h)	01h
FRU Control	PICMG - 2Ch (B0h)	04h

Table 16. Supported Standard IPMI and ATCA Commands

Message Commands	NetFn (NetFn / LUN0)	Cmd
Get FRU LED Properties	PICMG - 2Ch (B0h)	05h
Get LED Color Capabilities	PICMG - 2Ch (B0h)	06h
Set FRU LED State	PICMG - 2Ch (B0h)	07h
Get FRU LED State	PICMG - 2Ch (B0h)	08h
Set IPMB State	PICMG - 2Ch (B0h)	09h
Set FRU Activation Policy	PICMG - 2Ch (B0h)	0Ah
Get FRU Activation Policy	PICMG - 2Ch (B0h)	0Bh
Set FRU Activation	PICMG - 2Ch (B0h)	0Ch
Get Device Locator Record ID	PICMG - 2Ch (B0h)	0Dh
Set Port State	PICMG - 2Ch (B0h)	0Eh
Get Port State	PICMG - 2Ch (B0h)	0Fh
Compute Power Properties	PICMG - 2Ch (B0h)	10h
Set Power Level	PICMG - 2Ch (B0h)	11h
Get Power Level	PICMG - 2Ch (B0h)	12h
Get Fan Speed Properties	PICMG - 2Ch (B0h)	14h
Set Fan Level	PICMG - 2Ch (B0h)	15h
Get Fan Level	PICMG - 2Ch (B0h)	16h
Bused Resource	PICMG - 2Ch (B0h)	17h
Get IPMB Link Info	PICMG - 2Ch (B0h)	18h
Set AMC Port State	PICMG – 2Ch (B0h)	19h
Get AMC Port State	PICMG – 2Ch (B0h)	1Ah
FRU Control Capabilities	PICMG - 2Ch (B0h)	1Eh
Get Telco Alarm Capability	PICMG - 2Ch (B0h)	29h
Set Telco Alarm State	PICMG - 2Ch (B0h)	2Ah
Get Telco Alarm State	PICMG - 2Ch (B0h)	2Bh
Get Target Upgrade Capabilities	PICMG - 2Ch (B0h)	2Eh
Get Component Properties	PICMG - 2Ch (B0h)	2Fh
Abort Firmware Upgrade	PICMG - 2Ch (B0h)	30h
Initiate Upgrade Action	PICMG - 2Ch (B0h)	31h
Upload Firmware Block	PICMG - 2Ch (B0h)	32h
Finish Firmware Upload	PICMG - 2Ch (B0h)	33h
Get Upgrade Status	PICMG - 2Ch (B0h)	34h
Activate Firmware	PICMG - 2Ch (B0h)	35h
Query Self Test Results	PICMG - 2Ch (B0h)	36h
Query Rollback Status	PICMG - 2Ch (B0h)	37h
Initiate Manual Rollback	PICMG - 2Ch (B0h)	38h

Table 16. Supported Standard IPMI and ATCA Commands

Message Commands	NetFn (NetFn / LUN0)	Cmd
Switch Active Boot Flash	OEM1 – 30h (C0h)	A9h

3.5.3 Unit IPMI Controller

3.5.3.1 Managed Sensors

The IPMC monitors various on-board sensors to determine the health status of the board and to take appropriate actions in the event of a catastrophic failure, such as lighting LEDs and generating events. For information regarding the placement of the temperature sensors on the board, refer to the *ANPI1 Hardware Specification.* The thresholds are based on the voltage and temperature requirements of the devices present.

Table 17. ANPI1 Managed Sensors

#	Name	Type	Reading Type	Normal Reading	Notes
0	Hot-Swap	ATCA FRU Hotswap	Sensor-specific	0x00-0x07	Return M0 to M7 ATCA hotswap states
1	RTM Hot-Swap	ATCA FRU Hotswap	Sensor-specific	0x00-0x07	Return M0 to M7 ATCA hotswap states for the RTM
2	Version Change	Version Change	Sensor-specific	0x07	Software or F/W Change detected with associated Entity was successful.
3	IPMB	IPMB Link	Sensor-specific	0x00-0x03	0x00 – IPMB-A disabled, IPMB-B disabled 0x01 – IPMB-A enabled, IPMB-B disabled 0x02 – IPMB-A disabled, IPMB-B enabled 0x03 – IPMB-A enabled, IPMB-B enabled
4	Watch-Dog Timer	Watchdog 2	Sensor-specific	0x00-0x02	0x00 – Timer expired, status only 0x01 – Hard Reset 0x02 – Power Down
5	12V	Voltage	Threshold	12.00	This sensor measures voltage in Volts Default Thresholds LNR: 6.029 / LC: 10.843 / LNC: 11.423 / UNC: 12.641 / UC: 13.221 / UNR: 14.845
6	3.3V	Voltage	Threshold	3.3	This sensor measures voltage in Volts Default Thresholds LNR: 2.010 / LC: 2.974 / LNC: 3.147 / UNC: NA / UC: 3.464 / UNR: 3.666
7	2.5V	Voltage	Threshold	2.5	This sensor measures voltage in Volts Default Thresholds LNR: 1.509 / LC: 2.389 / LNC: NA / UNC: NA / UC: 2.627 / UNR: 2.960
8	1.8VF	Voltage	Threshold	1.8	This sensor measures voltage in Volts Default Thresholds

#	Name	Type	Reading Type	Normal Reading	Notes					
					LNR	LC	LNC	UNC	UC	UNR
					1.000	1.705	1.715	1.891	1.901	2.499
9	1.8VS	Voltage	Threshold	1.8	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					1.000	1.705	1.715	1.891	1.901	2.499
10	1.5VF	Voltage	Threshold	1.5	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					1.000	1.431	NA	NA	1.578	2.009
11	1.5VS	Voltage	Threshold	1.5	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					1.000	1.431	NA	NA	1.578	2.009
12	1.2V	Voltage	Threshold	1.2	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.500	1.147	NA	1.264	1.323	2.009
13	LMP Temp	Temperature	Threshold	25	This sensor measures temperature in degrees					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					NA	0.0	10.0	85.0	95.0	105.0
14	NPA Temp	Temperature	Threshold	25	This sensor measures temperature in degrees					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					NA	-10.0	0.0	79.0	89.0	99.0
15	CAMA Temp	Temperature	Threshold	25	This sensor measures temperature in degrees					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					NA	0.0	10.0	102.0	112.0	122.0
16	NPB Temp	Temperature	Threshold	25	This sensor measures temperature in degrees					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					NA	0.0	10.0	97.0	107.0	117.0
17	CAMB Temp	Temperature	Threshold	25	This sensor measures temperature in degrees					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					NA	0.0	10.0	103.0	113.0	123.0
18	SWT Temp	Temperature	Threshold	25	This sensor measures temperature in degrees					
					Default Thresholds					

#	Name	Type	Reading Type	Normal Reading	Notes					
					LNR	LC	LNC	UNC	UC	UNR
					NA	0.0	10.0	100.0	110.0	120.0
19	Reset	OEM reserved	OEM Discrete	0 or 1	0			1		
					No reset has occurred			Reset occurred		
20	Boot Flash	OEM reserved	Digital	0 or 1	0			1		
					Flash bank 0 has been selected			Flash bank 1 has been selected		
21	FRU Control	OEM reserved	OEM Discrete	0x00-0x03	FRU Control commands					
					0	1	2	3		
					Command deasserted	Cold reset requested	Graceful reboot requested	Diagnostic Interrupt requested		
22	-48V A	Voltage	Threshold	48	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					30.027	36.221	40.133	60.019	72.081	77.292
23	-48V B	Voltage	Threshold	48	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					30.027	36.221	40.133	60.019	72.081	77.292
24	-48V Current	Current	Threshold	4	This sensor measures current in Amps					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					NA	NA	NA	NA	NA	9.071
25	-48V Temp	Temperature	Threshold	25	This sensor measures temperature in degrees					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					NA	-10.8	-0.95	89.67	99.52	109.37
26	P1V	Voltage	Threshold	1	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.500	0.956	NA	NA	1.052	2.004
27	1VNPA	Voltage	Threshold	1	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.500	0.956	NA	NA	1.052	2.004
28	1VNPB	Voltage	Threshold	1	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR

#	Name	Type	Reading Type	Normal Reading	Notes					
					0.500	0.956	NA	NA	1.052	2.004
29	1VSWT	Voltage	Threshold	1	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.500	0.972	NA	NA	1.036	2.004
30	AVSSWT	Voltage	Threshold	1	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.500	0.876	NA	NA	1.036	2.004
31	1VCAMA	Voltage	Threshold	1	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.500	0.956	NA	NA	1.052	2.004
32	1VCAMB	Voltage	Threshold	1	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.500	0.956	NA	NA	1.052	2.004
33	VTTA	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
34	VTTB	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
35	VTTC	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
36	VTTD	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
37	VTTE	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
38	VTTF	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR

#	Name	Type	Reading Type	Normal Reading	Notes					
					0.300	0.700	NA	NA	0.780	2.004
39	VTTG	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
40	VTTH	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
41	VTTI	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
42	VTTJ	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
43	VTTK	Voltage	Threshold	0.75	This sensor measures voltage in Volts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					0.300	0.700	NA	NA	0.780	2.004
44	Base Channel 1	OEM reserved	OEM Discrete	0 or 1	0			1		
					Base channel 1 disabled			Base channel 1 enabled		
45	Base Channel 2	OEM reserved	OEM Discrete	0x00-0x01	0			1		
					Base channel 2 disabled			Base channel 2 enabled		
46	Fabric Channel 1	OEM reserved	OEM Discrete	0x00-0x04	0	1	2	3	4	
					Fabric channel 1 disabled	Fabric channel 1 enabled 1GBASE-BX	Fabric channel 1 enabled 10GBASE-BX4	Fabric channel 1 enabled 10GBASE-KR	Fabric channel 1 enabled 40GBASE-KR4	
47	Fabric Channel 2	OEM reserved	OEM Discrete	0x00-0x04	0	1	2	3	4	
					Fabric channel 2 disabled	Fabric channel 2 enabled 1GBASE-BX	Fabric channel 2 enabled 10GBASE-BX4	Fabric channel 2 enabled 10GBASE-KR	Fabric channel 2 enabled 40GBASE-KR4	

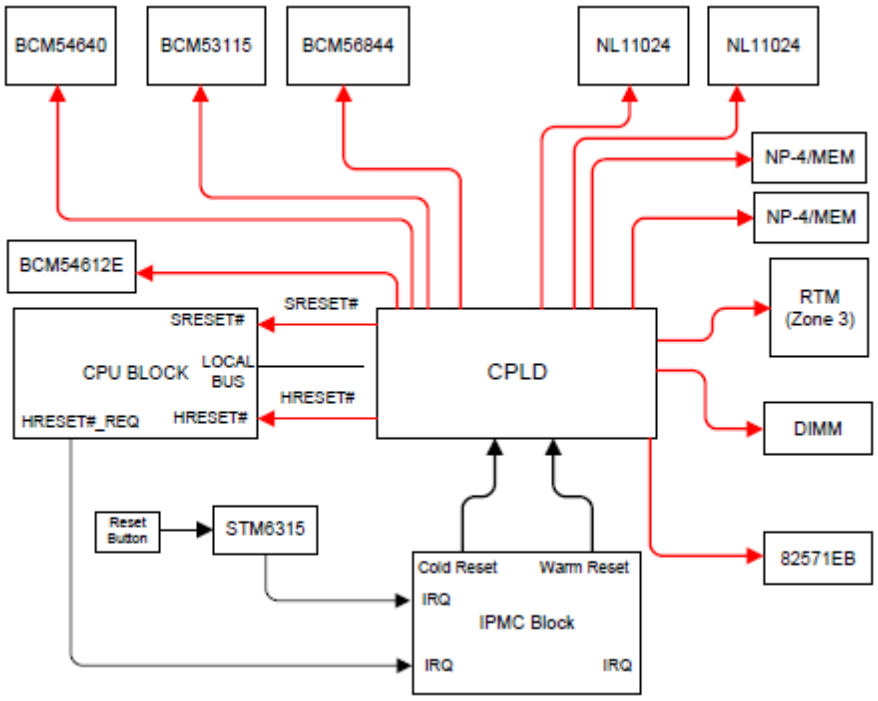
#	Name	Type	Reading Type	Normal Reading	Notes					
48	Update Channel	OEM reserved	OEM Discrete	0 or 1	0			1		
					Update channel disabled			Update channel enabled		
49	Bused Resource	OEM reserved	OEM Discrete	0 or 1	Bused Resource Control command					
					Shelf Manager to Board					
					0			1		
					Command deasserted			Command requested		
50	Power	OEM reserved	Threshold	25	This sensor measures power in watts					
					Default Thresholds					
					LNR	LC	LNC	UNC	UC	UNR
					NA	NA	NA	NA	NA	250.89
51	Eject Latch Closed	Slot or Connector	Digital	0 or 1	0			1		
					Eject latch is open Fault Status asserted			Eject latch is closed Identify Status asserted		
52	RTM Present	Slot or Connector	Digital	0 or 1	0			1		
					RTM is not present			RTM is present		
53	Power Fail	Power Supply	Digital	0 or 1	0			1		
					Power is healthy			Power is not healthy		
54	-48V Absent A	Power Supply	Digital	0 or 1	0			1		
					Power Supply A Presence detected			Power Supply A Failure detected		
55	-48V Absent B	Power Supply	Digital	0 or 1	0			1		
					Power Supply B Presence detected			Power Supply B Failure detected		
56	Fuse Fault	Power Supply	Digital	0 or 1	0			1		
					Fuse OK			Fuse fault detected		
57	En Fuse Fault A	Power Supply	Digital	0 or 1	0			1		
					Fuse OK			Fuse fault detected		
58	En Fuse Fault B	Power Supply	Digital	0 or 1	0			1		
					Fuse OK			Fuse fault detected		

#	Name	Type	Reading Type	Normal Reading	Notes
59	Memory	Memory	Sensor-specific	N/A	
60	Sys FW Progress	System Firmware Progress	Sensor-specific	N/A	FW progress event codes

3.5.3.2 Reset Handling

The blade level reset signals are controlled by the IPMC. Figure below illustrates the device reset signals and their interconnections. The IPMC forwards the command either initiated by the LMP application or the IPMI application to the CPLD and the CPLD decodes the command and sends individual reset to the devices accordingly. Refer to the IPMC and the CPLD pin descriptions for the details.

Figure 14. CPLD Connections



The IPMC supports the reset procedures described in Table 18.

Table 18. Reset Handling

Priority	Reset Source	IPMI Command (NetFn/Cmd)	Description
--	IPMC Cold Reset	Cold Reset (0x06/0x02)	Performs a cold reset of the IPMC. The state of the payload reset signals are maintained by the CPLD during the IPMC reset sequence.
1	IPMC Warm reset	Warm Reset (0x06/0x03)	Performs a cold reset of the IPMC. The state of the payload reset signals are maintained by the CPLD during the IPMC reset sequence.
2	Cold Reset	FRU Control (Cold Reset, FRU #0) (0x2c/0x04)	Performs cold reset of the payload. The IPMC asserts HRESET# and SRESET# signals to LMP and reset signals to all of the other components except the NP-4 blocks and RTM.
3	Warm Reset	FRU Control (Cold Reset, FRU #0) (0x2c/0x04)	Performs warm reset of the payload. The IPMC asserts SRESET# to LMP.
4	Graceful Reboot	FRU Control (Graceful Reboot, FRU #0) (0x2c/0x04)	Performs a graceful reboot of the LMP. The IPMC forwards the “Graceful Reboot Command” to LMP over the payload interface (UART) and logs an event to the SEL. The LMP completes shutdown process and sends Completion Message to IPMC via the payload interface (UART). The IPMC then performs a cold reset of payload (i.e. LMP and associated devices, excluding NP4 and RTM)
5	Diagnostic Interrupt	FRU Control (Graceful Reboot, FRU #0) (0x2c/0x04)	Performs a Diagnostic Interrupt to the LMP. The IPMC sends a “Diagnostic Reboot Command” to LMP over the payload interface (UART). The LMP completes shutdown process and sends Completion Message to IPMC via the payload interface (UART).
8	NP4 Reset	OEM IPMI command	Performs a cold reset of an NP4. It will reset the CPU and all TOPs and internal memories.
10	Front Panel Reset Button		Performs LMP cold reset. See description under #2.

3.5.3.2.1 OEM Boot Flash Sensor

The OEM Boot Flash Sensor included on ANPI1 is used to indicate when the LMP boot flash select signal changes state and to indicate why the signal has changed state. The OEM Boot Flash

Sensor's reading indicates which bank the IPMC has currently selected. The format of the event data generated from the OEM Boot Flash Sensor is described in the table below.

Table 19. OEM Boot Flash Sensor – Event Data Format

Event Data	Data Field
1	[7:6] - 00b = Unspecified in byte 2 [5:4] - 10b = OEM code in byte 3 [3:0] - 00h = Flash bank 0 has been selected 01h = Flash bank 1 has been selected
2	Not used. Set to 00h.
3	The cause of reset: [7:0] - 00h = CFD Timer was initialized 01h = CFD Timeout has occurred 02h = Flash bank switch was externally initiated. 03h = No Change

OEM IPMI command is used to change the boot flash externally. It can be executed through payload interface, IPMB and IOL. The sequence of the events is:

1. OEM command is requested externally.
2. IPMC Firmware stores the boot flash number to a Non-volatile memory.
3. If reset Command is requested, the IPMC firmware selects the boot flash and updates the sensor event with reset cause. UBoot from SPI Flash and rootfs from NAND flash stay in synch, i.e. the banks form SPI and NAND flash are switched together.

3.5.3.2.2 OEM Payload Reset Sensor

The OEM Payload Reset Sensor is used to indicate the cause of payload resets that occur during the standard operation of the blade. Each time a payload reset occurs, an event containing the cause of the reset (or resets depending on how many resets were recorded by the reset sensor before an event could be generated by the IPMC) is generated by the blade.

The logic for the Reset Sensor is located on the IPMC FPGA, where all reset sources are physically routed. The IPMC FPGA monitors each reset source and toggles an interrupt connected to the H8 IPMC whenever a payload reset is detected. The IPMC learns the reason for the reset by reading a status register on the FPGA. In the event that multiple resets are generated at the same time (e.g. the front panel push button is pressed at the same time a FRU Control cold reset is issued), the IPMC prioritizes the reset sources for the sake of listing the event data. This prioritization is not related in any way to the hardware functionality.

The following table describes the mappings between the reset source and the associated sensor reading/event data, as well as their respective priorities. The type field is included in the table for informational purposes only. For the Sensor Reading/Event Data fields, if bit 7 is set to 01b then the reset source was a cold reset, and if bit 7 is 00b then the reset is a warm reset.

The format of the event data generated from the Reset Sensor is described in the table below.

Table 20. OEM Payload Reset Sensor – Event Data Format

Event Data	Data Field
1	[7:6] - 10b = OEM code in byte 2 [5:4] - 10b = OEM code in byte 3 [3:0] - 00h = No reset has occurred 01h = Reset occurred
2	The highest priority reset source
3	The second highest priority reset source. If only one reset occurs, this value is set to 0x00.

Table 21. OEM Payload Reset Sensor – Readings, Event Data, and Priorities

Priority	Reset Source	Type	Sensor Reading/ Event Data
--	No Previous Reset	N/A	0x00
1	Power-on Reset	Cold	0x01
2	CFD Watchdog	Cold	0x02
3	BMC Watchdog (Cold Reset)	Cold	0x03
4	PQ3 HRESET_REQ# assertion	Cold	0x04
5	LMP CKSTP_OUT0# or LMP CKSTP_OUT1# assertion	Cold	0x05
7	IPMI command – FRU Control (Cold Reset) or Set Control State (Payload Reset)	Cold	0x06
8	NP4-1 or NP4-2 reset	Cold	0x0F
9	BMC Watchdog (Warm Reset)	Warm	0x81
10	Front Panel Reset Button	Warm	0x82
11	RTM Reset Button	Warm	0x83
12	IPMI command – FRU control (Warm Reset)	Warm	0x84

3.5.3.3 ANPI FRU Data

The ANPI FRU data will be formatted according to the NSN specification. The FRU data area will be writable using the IPMI FRU Write commands. The FRU data will be provided in a binary format.

3.5.3.4 IPMI Controller SW Upgrade

The IPMC (for the front blade) and the MMC (for the RTM) firmware supports a reliable filed upgrade procedure compatible with the PICMG HPM.1 specification. The salient features of the IPMC, MMC upgrade procedures are as follows:

- The upgrade can be performed either over the Payload Interface and the LAN interface or over IPMB-0.

- The upgrade procedure is performed while the ANPI1 is online and operating normally.
- The upgrades of the firmware component are reliable. A failure in the download (error or interruption) does not disturb the ANPI1's ability to continue using the "old" firmware or its ability to restart the download process. The upgrades of the Boot Loader components are not reliable and may render the IPMC non-functional in the case of an incomplete upgrade.
- The upgrade is reversible. The firmware automatically reverts back to the previous firmware if there is a problem when first running the new code. It can be reverted manually using the HPM.1-defined Manual Rollback command.
- FRU data records are not lost or impacted during embedded SW upgrade

The IPMC has two upgradeable components:

- IPMC Firmware – Component Id 0
- IPMC Boot Loader – Component Id 1

The MMC has two upgradeable components:

- MMC Firmware – Component Id 0
- MMC Boot Loader – Component Id 1

IPMC and MMC FWs with component ID 0 exist in Active/Backup for upgrade reliability. Whereas the IPMC Boot Loader and the MMC Boot Loader only exists in Active.

3.5.3.5 Unit IPMI Controller Resets

The IPMC supports both cold and warm resets. The reset of the IPMC does not affect Payload, state of I/O control signals, affecting it nor with a Payload Reset nor with a Payload Power Cycle. The IPMC Cold Reset has the following behavior:

Info initialized by Cold Reset

- IPMC Firmware resets all internal data/states
- internal message rings
- IPMB message sequence number

Info not initialized by Cold Reset

- FRU State : Front , RTM
- Activation/Deactivation Locked bit : Front, RTM
- Power Enable signals : Front, RTM
- Payload Reset Signal : Front, RTM
- Blue LED state
- BMC Watchdog timer state.

The IPMC Warm Reset has the following behavior

Info initialized by Warm Reset

- IPMC Firmware resets all internal data/states
- internal message rings
- IPMB message sequence number

Info not initialized by Warm Reset

- FRU State : Front , RTM
- Activation/Deactivation Locked bit : Front, RTM
- Power Enable signals : Front, RTM
- Payload Reset Signal : Front, RTM
- Blue LED state
- BMC Watchdog timer state.
- Sensor thresholds/hysteresis
- Sensor event masks
- Sensor events.

3.5.3.6 Serial over LAN and IPMI over LAN Support

The IPMC implements IPMI over LAN (IOL) and Serial-over-LAN (SOL) (compliant to IPMI v. 2.0 specification with support for RCMP+, including support for straight password authentication for the RMCP+ sessions) via an Intel 82571EI Gigabit Ethernet Controller which has one full-integrated Gigabit Ethernet Media Access Control (MAC) and physical layer (PHY) port. The 82571EI on the board uses SGMII interface to connect to the base switch. The interface with the IPMC is the System Management Bus (SMB), where three signals SMB_ALERT, SMB_SCL (SMBCLK0), and SMB_SDA (SMBD0) are connected to the IPMC. The IPMC uses IPMI Channel 0x5 for the LAN channel () over the sideband interface to the 82571. The IPMC LAN channel is accessible through either base interface of the base switch on the ANPI1. The IPMC LAN channel can be configured to support IEEE 802.1q VLAN headers for IPMI over IP sessions on IEEE 802.3 Ethernet.

Note that only LMP processor is accessible via SOL. SOL support provides same services as the console that is attached locally.

The IPMC responds independently to a connection request from an IPMI 2.0 compliant counterpart.

3.5.3.7 BMC Watchdog Support Timer

The IPMI 1.5 specification defines the “BMC Watchdog Timer” as a software-based timer that can be used by applications that require a timeout function. ANPI1 IPMC implements the BMC Watchdog Timer interface. The behavior of the action is also configurable based on the IPMI 1.5 specification.

By default, the IPMC has this watchdog timer disabled. It is the responsibility of the U-Boot or Linux to program the BMC payload watchdog timeout value, and to enable the watchdog. From the U-Boot prompt the BMC payload watchdog may be configurable over a range of 0 – 600 seconds. After complete booting, the OS is responsible to continue strobing this watchdog timer. Native Linux Watchdog daemon is used to strobe BMC watchdog. Enabling and disabling of the BMC watchdog are supported using generic Linux kernel watchdog driver.

As pre-timeout interrupt support is defined by the IPMI specification as optional, the ANPI1 support for this feature is detailed here. The pre-timeout interrupts supported by the ANPI1 are NMI and the messaging interrupt.

When the BMC Watchdog Timer is set to “Hard Reset” for action to take upon timeout, the IPMC generates either a warm reset or a cold reset. The type of reset generated by the IPMC is set by the U-Boot or SMS using the “Set WDT Reset Type” IPMI OEM request.

3.5.4 HPI Support

ANPI-1 supports original Radisys HPI implementation (HPI Core Services in the diagram below) which is compliant with HPI specification B03.02 and implements all requested functionalities. The ANPI-1 blade will be model as a HPI AdvancedTCA® FRU Resource and will support the following capabilities:

SAHPI_CAPABILITY_RESOURCE

SAHPI_CAPABILITY_FRU

SAHPI_CAPABILITY_POWER

SAHPI_CAPABILITY_RESET

SAHPI_CAPABILITY_RDR

SAHPI_CAPABILITY_SENSOR

SAHPI_CAPABILITY_CONTROL

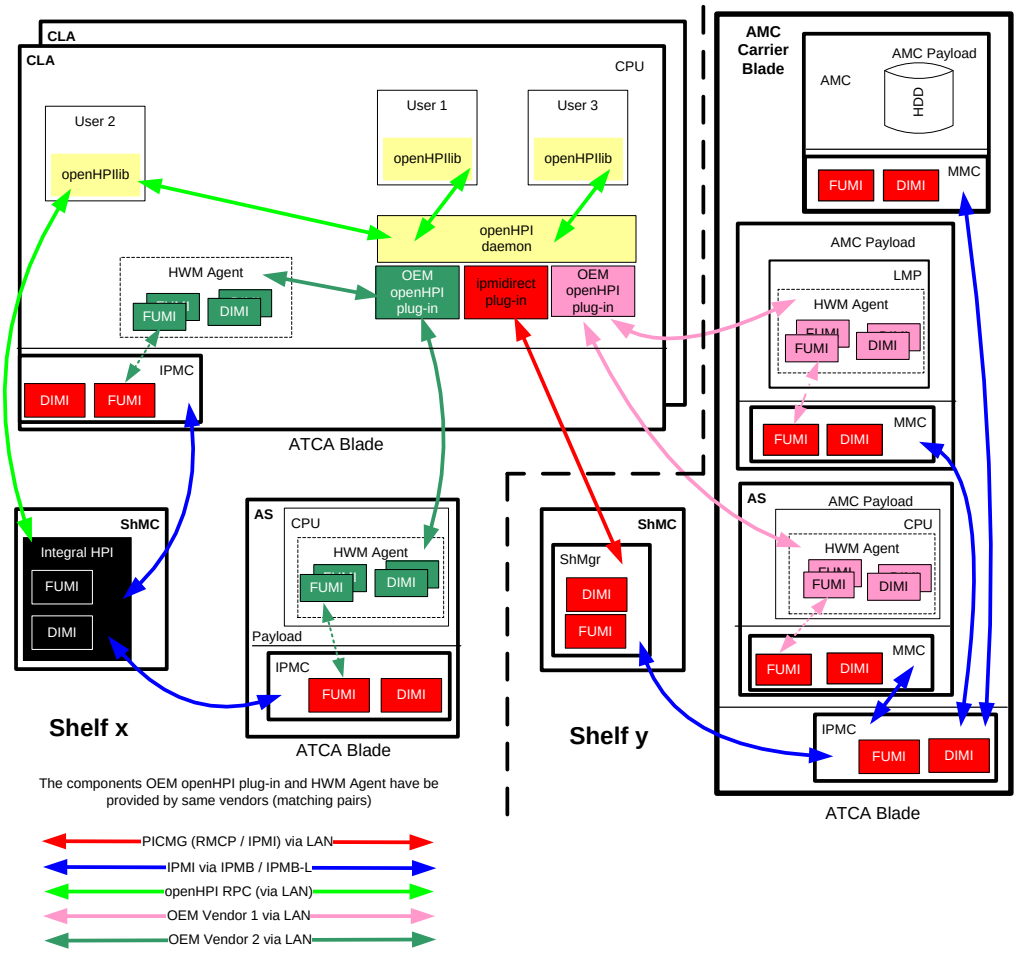
SAHPI_CAPABILITY_MANAGED_HOTSWAP

SAHPI_CAPABILITY_INVENTORY_DATA

SAHPI_CAPABILITY_EVENT_LOG

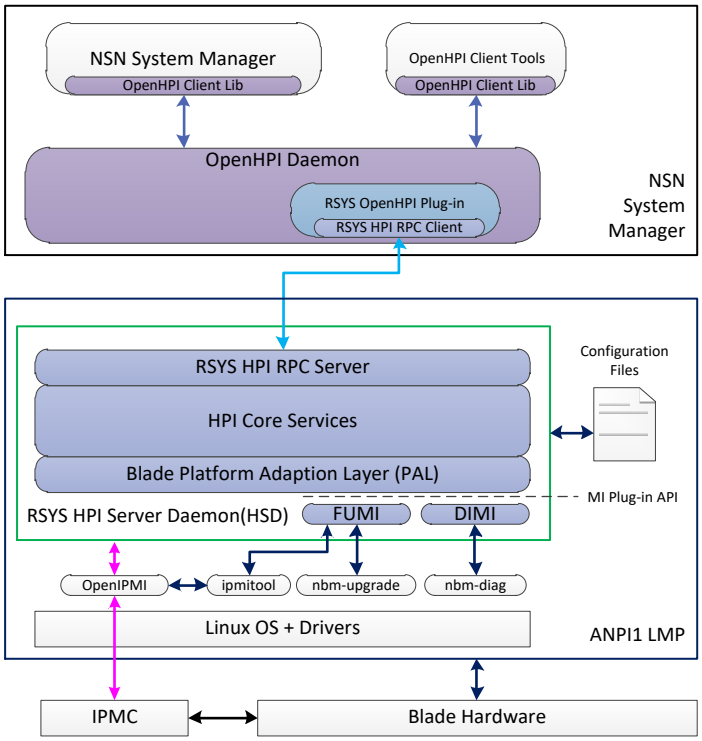
NSN requires the OpenHPI Daemon to run on the System Manager node. Each vendor is expected to add an OpenHPI plug-in that provides access to the blades HPI Management instruments (i.e. FUMI, DIMI, Sensors, Controls, etc.). The NSN architecture is shown in Figure 17and described in detail in [14].

Figure 15. NSN HPI Reference Architecture



The Radisys Implementation to support the NSN reference architecture is shown in Figure 16.

Figure 16. ANPI1 HPI Architecture



A new RSYS OpenHPI Plug-in (RSYSOHPI) module to implement the functionality of the OEM HPI Plug-in shown in Figure 15 will be developed to meet requirements for ANPI1. The RSYSOHPI will encapsulate the Radisys HPI Client Library (RHCL). The RSYSOHPI will not modify the OpenHPI daemon or its associated plugins. The RSYSOHPI will not prevent the usage of any other plugins. The Radisys Blade HPI Server Daemon (BHSD) which runs on the LMP of the ANPI1 implements the functionality of the ipmidirect plug-in and the HWM agent shown in Figure 17.

The RSYSOHPI uses the RHCL to access the BHSD through a RPC interface (RMCP over UDP socket). The BHSD creates two HPI resources: one for the ANPI1 and one for the NIRT3. BHSD interacts with the IPMC through the OpenIPMI driver and maps the sensors, inventory data and PICMG defined controls to corresponding instrumentation in the ANPI1 and NIRT3 HPI resources. BHSD implements FUMIs which use the command line upgrade tools to perform the upgrades. BHSD implements DIMIs which use the command line diagnostics tools to perform the diagnostics on the blade. The ANPI1 will also support the option where the OpenHPI daemon is running locally on the LMP as shown in 16. This will enable the OpenHPI client tools to be run locally.

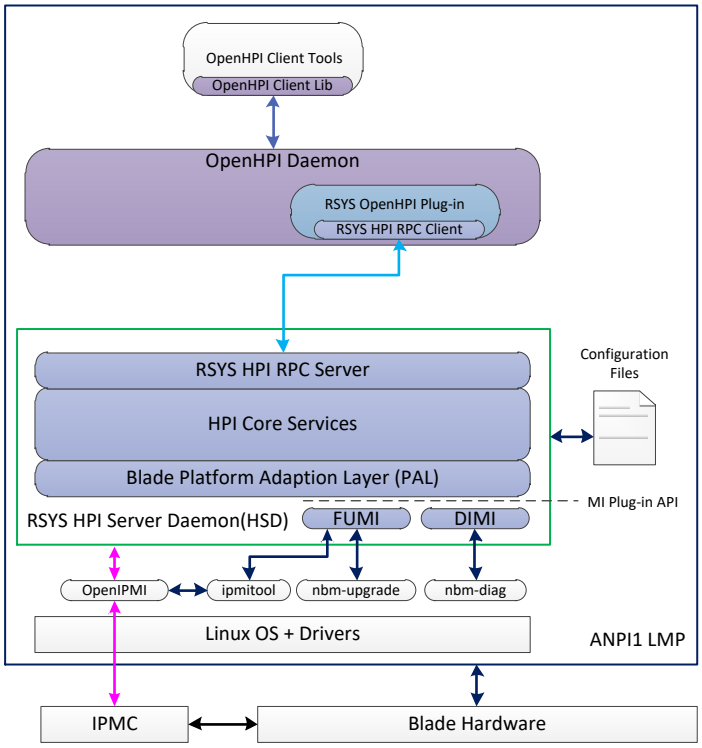
One RSYSOHPI plugin is capable of handling connection to multiple ANPI1-A blades. There is no limitation regarding the number of supported ANPI1 blades. Radisys restricted the RSYS HPI RPC server to 32 simultaneously opened HPI sessions, which could be increased up to 255. The HPI plug-in will be using the HPI Client Library and there is no restrictions regarding the number of the HPI plug-ins instances.

Only configuration required for the functioning of RSYS Open HPI plugin is the IP address of the ANPI1 and the entity path prefix. There is no communication between the BHSD servers on different ANPI1. Assumption is that all access is through the OpenHPI daemon. In this sense, each ANPI1 does not know the existence of other ANPI1s.

The RPC link between the Radisys OpenHPI Plugin and BHSD is UDP based and the health of the session is monitored via a “keep alive” ping. If the connection is down for more than the configured time (currently 60 seconds), the session will be dropped and new session should be initiated.

It should be noted that OpenHpi daemon, including the plug-in of the ANP1-A can be started at any time regardless of the state of the ANPI1-A and the agent. OpenHpi daemon, including the plug-in of the ANPI1-A don't need to reset regardless of the state of the ANPI1-A and the agent. If the underlying UDP socket is interrupted for whatever reason, the BHSD and the HCL would still hold onto valid HPI session ID and re-establish the UDP socket communication automatically.

Figure 17. OpenHPI Daemon on ANPI1



The main FRU representing the ANP-1 blade in Radisys HPI implementation is model with the following HPI capabilities:

3.5.4.1 Radisys OpenHPI Plug-in (RSYSOHPI)

RSYSOHPI will be implemented as a plug-in library for OpenHPI version 3.0.0. The RSYSOPHI will encapsulate the Radisys HPI Client Library and will support simultaneous sessions to multiple ANPI1 blades. The configuration for each blade instance will be provided as an entry in the /etc/openhpi/openhpi.conf file. The configuration parameters offered by the plug-in is shown below:

handler librsysophi {

entity_root = "{SYSTEM_BOARD,n}"

host = xxx.yyy.zzz.bbb

}

The "host" field specifies the IP address of the target ANPI1 blade to be managed by the daemon.

RSYSOHPI shall support the APIs listed in Table 22.

Table 22. RSYSOHPI APIs

Req. #	HPI Function API (acc. SAI-HPI-B.03.02)	SW Plat. Priority	RSYSOHPI Plug-in
	General		
R1-1	saHpiversionGet()	High	N/A
R1-2	saHpiInitialize()	High	N/A
R1-3	saHpiFinalize()	High	N/A
	Session Management		
R2-1	saHpiSessionOpen()	High	Yes
R2-2	saHpiSessionClose()	High	Yes
R2-3	saHpiDiscover()	High	Yes
	Domain Discovery		
R3-1	saHpiDomainInfoGet()	High	Yes
R3-2	saHpiDrtEntryGet()	High	Yes
R3-3	saHpiDomainTagSet()	Low	No
	Resource Presence Table		
R4-1	saHpiRptEntryGet()	High	Yes
R4-2	saHpiRptEntryGetByResouceId()	High	Yes
R4-3	saHpiResourceSeveritySet()	Medium	Yes
R4-4	saHpiResourceTagSet()	Low	No
R4-5	saHpiMyEntityPathGet()	High	No
R4-6	saHpiGetIdByEntityPath()	High	Yes
R4-7	saHpiGetChildEntityPath()	Medium	Yes
R4-8	saHpiResourceFailedRemove()	High	Yes
	Event Log Management		
R5-1	saHpiEventLogInfoget()	High	Yes
R5-2	saHpiEventLogCapabilitiesGet()	Medium	Yes
R5-3	saHpiEventLogEntryGet()	High	Yes

R5-4	saHpiEventLogEntryAdd()	Low	No
R5-5	saHpiEventLogClear()	High	Yes
R5-6	saHpiEventLogTimeGet()	High	Yes
R5-7	saHpiEventLogTimeSet()	High	Yes
R5-8	saHpiEventLogStateGet()	High	Yes
R5-9	saHpiEventLogStateSet()	High	Yes
R5-10	saHpiEventLogOverflowReset()	High	Yes
	Events		
R6-1	saHpiSubscribe()	High	Yes
R6-2	saHpiUnsubsribe()	High	Yes
R6-3	saHpiEventGet()	High	Yes
R6-4	saHpiEventAdd()	Low	No
	Domain Alarm Table		
R7-1	saHpiAlarmGetNext()	Low	No
R7-2	saHpiAlarmGet()	Low	No
R7-3	saHpiAlarmAcknowledge()	Low	No
R7-4	saHpiAlarmAdd()	Low	No
R7-5	saHpiAlarmDelete()	Low	No
	RDR Management		
R8-1	saHpiRdrGet()	High	Yes
R8-2	saHpiRdrGetByInstrumentId()	High	Yes
R8-3	saHpiRdrUpdateCountGet()	High	Yes
	Sensors		
R9-1	saHpiSensorReadingGet()	High	Yes
R9-2	saHpiSensorTresholdsGet()	Medium	Yes
R9-3	saHpiSensorTresholdsSet()	Medium	Yes
R9-4	saHpiSensorTypeGet()	High	Yes
R9-5	saHpiSensorEnableGet()	Medium	Yes
R9-6	saHpiSensorEnableSet()	Medium	Yes
R9-7	saHpiSensorEventEnableGet()	Medium	Yes
R9-8	saHpiSensorEventEnableSet()	Medium	Yes
R9-9	saHpiSensorEventMasksGet()	Medium	Yes
R9-10	saHpiSensorEventmasksSet()	Medium	Yes

	Controls		
R10-1	saHpiControlTypeGet()	High	Yes
R10-2	saHpiControlGet()	High	Yes
R10-3	saHpiControlSet()	High	Yes
	Inventory Data Repositories		
R11-1	saHpiIdrInfoGet()	High	N/A
R11-2	saHpiIdrAreaHeaderGet()	High	N/A
R11-3	saHpiIdrAreaAdd()	Low	N/A
R11-4	saHpiIdrAreaAddById()	Low	N/A
R11-5	saHpiIdrAreaDelete()	Low	N/A
R11-6	saHpiIdrFieldGet()	High	N/A
R11-7	saHpiIdrFieldAdd()	Low	N/A
R11-8	saHpiIdrFieldAddById()	Low	N/A
R11-9	saHpiIdrFieldSet()	Low	N/A
R11-10	saHpiIdrFieldDelete()	Low	N/A
	Watchdog timers		
R12-1	saHpiWatchdogTimerGet()	Medium	N/A
R12-2	saHpiWatchdogTimerSet()	Medium	N/A
R12-3	saHpiWatchdogTimerReset()	Medium	N/A
	Annunciators		
R13-1	saHpiAnnunciatorGetNext()	Medium	N/A
R13-2	saHpiAnnunciatorGet()	Medium	N/A
R13-3	saHpiAnnunciatorAcknowledge()	Medium	N/A
R13-4	saHpiAnnunciatorAdd()	Medium	N/A
R13-5	saHpiAnnunciatorDelete()	Medium	N/A
R13-6	saHpiAnnunciatorModeGet()	Medium	N/A
R13-7	saHpiAnnunciatorModeSet()	Medium	N/A
	Hot Swap		
R16-1	saHpiHotSwapPolicyCancel()	High	N/A
R16-2	saHpiresourceActiveSet()	High	N/A
R16-3	saHpiResourceInactiveSet()	High	N/A
R16-4	saHpiAutoInsertTimeoutGet()	Medium	N/A
R16-5	saHpiAutoInsertTimeoutSet()	Medium	N/A
R16-6	saHpiAutoExtractTimeoutGet()	Medium	N/A

R16-7	saHpiAutoExtractTimeoutSet()	Medium	N/A
R16-8	saHpiHotSwapStateGet()	High	N/A
R16-9	saHpiHotSwapActionRequest()	High	N/A
	Reset Management		
R18-1	saHpiResourceResetStateGet()	High	N/A
R18-2	saHpiResourceResetStateSet()	High	N/A
	Power Management		
R19-1	saHpiResourcePowerStateGet()	High	N/A
R19-2	saHpiResourcePowerStateSet()	High	N/A

3.5.4.2 FUMI

RSYSOHPI shall support the FUMI APIs listed in Table 25. The FUMI interface supports TFTP as one of transfer protocols. The FUMI plug-in module in the BHSD will implement FUMIs listed in Table 23 on ANPI1 resource, and FUMIs listed in Table 24 on NITR3 resource.

Table 23. ANPI1-A FUMIs

FUMI Name	Entity Path Suffix	Rollback Support	Notes
system-os	SAHPI_ENT_OPERATING_SYSTEM.0	Yes	This FUMI maps to the Linux Kernel, dtb and RFS. All three images will be updated at once through this FUMI
system-boot	SAHPI_ENT_BIOS.1	Yes	This FUMI maps to the U-Boot component
ipmc-app	SAHPI_ENT_IPMC.0, SAHPI_ENT_MC_FIRMWARE.1	Yes	
ipmc-boot	SAHPI_ENT_IPMC.0, SAHPI_ENT_MC_FIRMWARE.0	No	
fru-data	SAHPI_ENT_SYSTEM_INVENTORY_DEVICE.0	No	

It should be noted that the ANPI1 blade stores two versions of the IPMI run-time code, a.k.a. application, and one for the boot code (16K size). Boot code is responsible for loading the active image. The boot code is very mature and in a way simple, and hasn't changed for many years. There are no two or two copies of the boot code. Therefore, there is no rollback capability for ipmc-boot and mmc-boot images.

ANPI1 software has two FUMI instruments for system-os and system-boot. A user will be able to upgrade only system-boot if it wants to.

Table 24. NITR3-A FUMIs

FUMI Name	Entity Path Suffix	Rollback Support	Notes
mmc-app	SAHPI_ENT_IPMC.0, SAHPI_ENT_MC_FIRMWARE.1	Yes	
mmc-boot	SAHPI_ENT_IPMC.0 , SAHPI_EN T_MC_FIRMWARE.0	No	
fru-data	SAHPI_ENT_SYSTEM_INVENTORY_DEVICE.0	No	

Table 25. RSYSOHPI FUMI APIs

Req. #	HPI Function API (acc. SAI-HPI-B.03.02)	SW Plat. Priority	Plug -in	API Behavior
	FUMI			
R15-1	saHpiFumiSpecInfoGet()	High	A	As defined by HPI-B03.02
R15-2	saHpiFumiServiceImpactGet()	High	A	The list of affected entities shall be set to resources that will be impacted by upgrading this entity. The impacted entities are confined to the entities on the resource.
R15-3	saHpiFumiSourceSet()	High	A	Sets the SourceUri of the source image file. This URI will typically point to a local file that was installed by the bundle FUMI.

R15-4	saHpiFumiSourceInfoValidateStart()	High	A	Starts the validation of the image file. If the SourceUri indicates a remote location, then the image file is first copied to the local file system. The validation of the image will entail checking the integrity of the file, checking the compatibility with existing hardware. The FUMI Library will call the associated Firmware Tool to perform the validation and compatibility checks.
R15-5	saHpiFumiSourceInfoGet()	High	A	The SourceInfo structure shall return the version of the source image in the version fields. The Description field shall contain the system release number of the bundle that provided this image. The FUMI Library will call the associated Firmware Tool to get the version info from the file.
R15-6	saHpiFumiSourceComponentInfoGet()	High	A	Returns SA_ERR_HPI_CAPABILITY
R15-7	saHpiFumiTargetInfoGet()	High	A	Returns the information of the installed image that is currently active on the hardware module. The FUMI Library will call the associated Firmware Tool to get the version info from the target device.
R15-8	saHpiFumiTargetComponentInfoGet()	High	A	Returns SA_ERR_HPI_CAPABILITY
R15-9	saHpiFumiLogicalTargetInfoGet()	High	A	Returns the information of the rollback image if it is supported by the software entity represented by the FUMI. The FUMI Library will call the associated Firmware Tool to get the rollback version info from the target device.
R15-10	saHpiFumiLogicalTargetComponentInfoGet()	High	A	Returns *SA_ERR_HPI_CAPABILITY*

R15-11	saHpiFumiBackupStart()	High	A	Returns *SA_ERR_HPI_CAPABILITY*
R15-12	saHpiFumiBankBootOrderSet()	High	A	Returns *SA_ERR_HPI_CAPABILITY*
R15-13	saHpiFumiBankCopyStart()	High	A	Returns *SA_ERR_HPI_CAPABILITY*
R15-14	saHpiFumiInstallStart()	High	A	Starts the installation of the image over rollback image on the device. If the rollback is not supported by the entity then the image is installed over the main image. The FUMI Library will call the associated Firmware Tool to program the device with the new image
R15-15	saHpiFumiUpgradeStatusGet()	High	A	Returns the status as defined by HPI-B03.02
R15-16	saHpiFumiTargetVerifyStart()	High	A	Returns *SA_ERR_HPI_CAPABILITY*
R15-17	saHpiFumiTargetVerifyMainStart()	High	A	Returns *SA_ERR_HPI_CAPABILITY*
R15-18	saHpiFumiUpgradeCancel()	High	A	As defined by HPI-B03.02
R15-19	saHpiFumiAutoRollbackDisableGet()	High	A	Returns *SA_ERR_HPI_CAPABILITY*
R15-20	saHpiFumiAutoRollbackDisableSet()	High	A	Returns *SA_ERR_HPI_CAPABILITY*
R15-21	saHpiFumiRollbackStart()	High	A	Reverses the roles of the active and rollback images installed on the target. If the entity does not support rollback then this function shall return *SA_ERR_HPI_CAPABILITY*. The FUMI Library will call the associated Firmware Tool to perform the rollback on the target device.
R15-22	saHpiFumiActivateStart()	High	A	Starts the activation of the newly installed image. The FUMI Library will call the associated Firmware Tool to perform the steps to activate the newly installed image.

R15-23	saHpiFumiCleanup()	High	A	As defined by HPI-B03.02

3.5.4.3 DIMI

RSYSOHPI shall support the DIMI APIs listed in Table 26. The DIMI plug-in module in the BHSD will implement DIMIs listed in Table 27 on ANPI1 resource, and DIMIs listed in Table 28 on NITR3 resource.

Table 26. RSYSOHPI DIMI APIs

Req. #	HPI Function API (acc. SAI-HPI-B.03.02)	SW Plat. Priority	Plug-in
	DIMI		
R14-1	saHpiDimiInfoGet()	High	A
R14-2	saHpiDimiTestInfoGet()	High	A
R14-3	saHpiDimiTestReadinessGet()	High	A
R14-4	saHpiDimiTestStart()	High	A
R14-5	saHpiDimiTestCancel()	High	A
R14-6	saHpiDimiTestStatusGet()	High	A
R14-7	saHpiDimiTestResultsGet()	High	A

Table 27. ANPI1-A DIMIs

DIMI Name	Entity Path Suffix	Notes
unit-computer	SAHPI_ENT_PROCESSOR.0	This DIMI implements the tests that listed in section 3.5.5.3.
switch	SAHPI_ENT_SWITCH.0	This DIMI implements the tests that are listed in section 3.5.5.1
npu	SAHPI_ENT_NPU.0,	This DIMI implements the tests that are listed in section 3.5.5.2
ipmc	SAHPI_ENT_IPMC.0,	This DIMI implements the tests that are listed in section **Error! Reference source not found.**

Each NPU entity will be model as an HPI MI. It will be possible to trigger a particular test via DIMI to be executed on a particular NPU.

Table 28. NITR3-A DIMIs

FUMI Name	Entity Path Suffix	Notes
mmc	SAHPI_ENT_IPMC.0,	This DIMI implements one test that maps to the MMC Get Self-test results

3.5.5 Diagnostics

The nbm_diag utility performs diagnostics on the ANPI1 board hardware resources. The same utility is usable a serial-over-LAN connection. The diagnostic implementation does allow for the user to explicitly cancel a diagnostic test through any supported interface. However, the diagnostic CLI and underlying tool does not allow for the user to start overlapping diagnostic tests or start the same test concurrently from different interfaces.

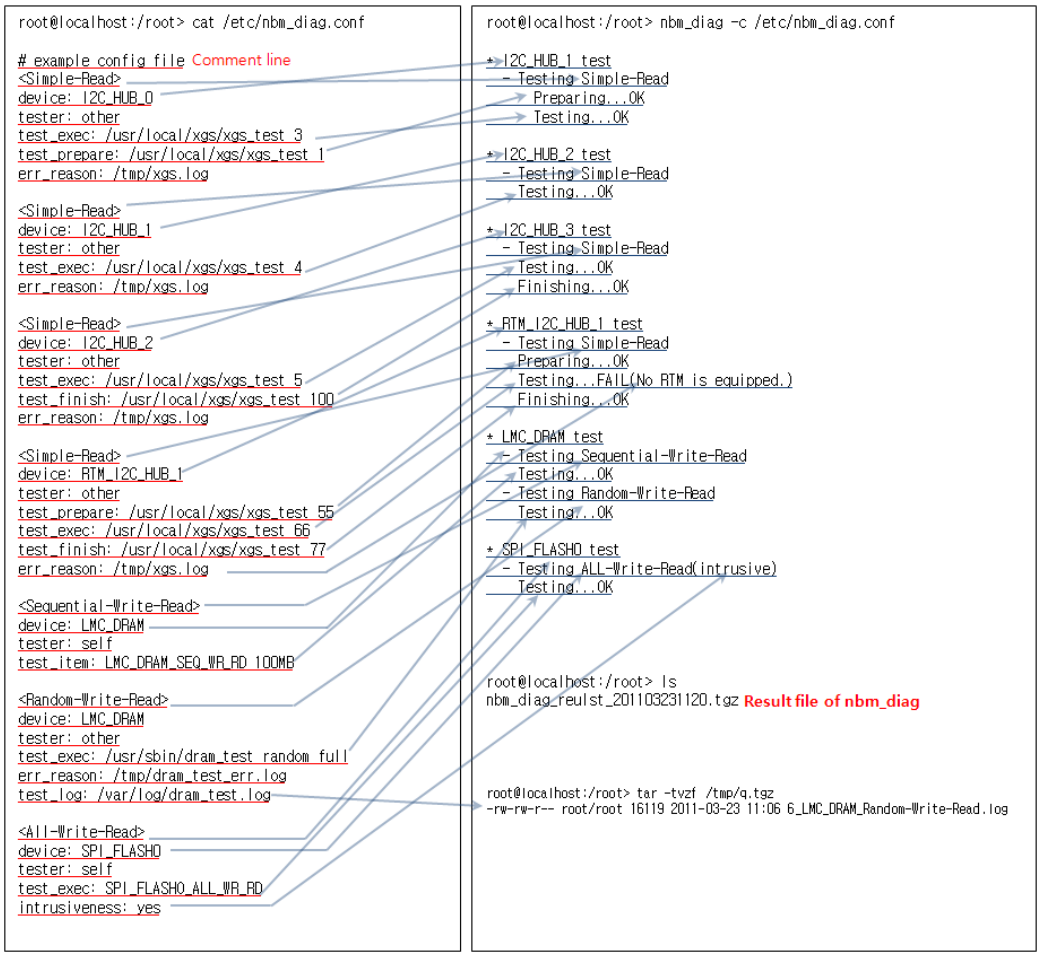
The list of diagnostics tests are specified in a configuration file that is read by nbm_diag. The directives in the configuration file are shown in Table 29.

Table 29. Nbm_diag Configuration File Directives

Directive	Function	Remark
#	Start character of comment line	
<$STRING>	Defines a Test item.	The directives after this one until next “<>” or “[]” become the attribute of the Test Item.
[$STRING]	Defines a Test module.	The directives after this one until next “<>” or “[]” become the attribute of the Test module.
device: $STRING	Defines a device to test.	
extern: (yes:no)	Defines the Test program: either *nbm_diag* or an external program.	Default: no
intrusiveness: (yes|no)	Defines intrusiveness.	Default: no
test_item: $STRING	Defines nbm_diag built-in test item.	Usable only with “tester: self”. nbm_diag built-in test item list is checked by “nbm_diag –l”.
test_exec: $LINE	Defines an execution file and options for the test.	Usable only with “tester: other”
test_prepare: $LINE	Defines an execution file and options for test preparation before the test.	Usable only with “tester: other”
test_finish: $LINE	Defines an execution file and options to recover the status after the test.	Usable only with “tester: other”
test_log: $STRING	Defines a log file when the Test program generates one.	Log file may be used for the test result data.
test_result: $STRING	Defines a result file of external program including an err log when the Test program generates one.	Usable only with “tester: other” See the syntax for test result Without this, return value shall be the result of the test. default test_result file is /tmp/nbm_diag_result.txt

Figure 20 shows an example configuration file and an expected result on the monitor for the hardware resources diagnostics including I2C_HUB_0, I2C_HUB_1, I2C_HUB_2 device, LMC_DRAM, SPI_FLASH0.

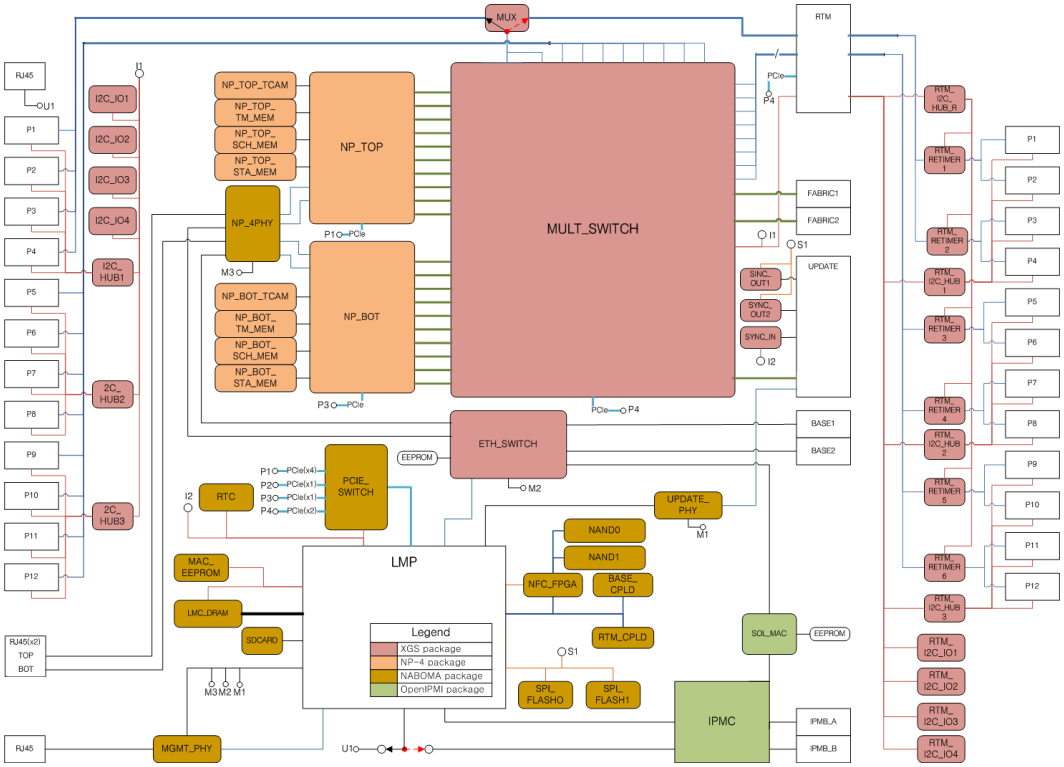
Figure 18. Example configuration and execution for Diagnostics



The devices that are available for diagnostics are illustrated in Figure 21along with corresponding diagnostic packages. The diagnostics are organized into 4 major packages:

- XGS Package – provides tests for Ethernet Switches and physical interfaces
- NP-4 Package – provides tests for the NP-4 NPU and associated memory devices
- NABOMA Package – provides tests for the LMP subsystem and associated board devices
- OpenIPMI Package – provides tests for the IPMC and IPMC interfaces

Figure 19. Devices and their diagnostic packages



3.5.5.1 XGS Package

TBD

3.5.5.2 NP-4 Package

Table 30. NP-4 Package Diagnostic Tests

No	Device	Description
1	Search Memory	Memory Read/Write Test using EZapiPrm_BstStartTest/CheckTest in EZ API Library (Return: Success or Fail)
2	Traffic Manager Memory	Memory Read/Write Test using EZapiPrm_BstStartTest/CheckTest in EZ API Library Check (Return: Success or Fail)
3	Statistics Memory	Memory Read/Write Test using EZapiPrm_BstStartTest/CheckTest in EZ API Library (Return: Success or Fail)
4	TCAM	Common Status Register Check via TCAM MDC/MDIO interface (Return: Register Status)

3.5.5.3 NABOMA Package

Table 31. NABOMA Package Diagnostic Tests

Device	Test ID	Intru.[1]	Procedure
NP_4PHY	NP_4PHY_ READ_OUI	No	1. Read OUI register. 2. If the value read is static OUI, the test passes. 3. Repeat 1 and 2 for all the ports (4 ports) for different PHY addresses.
	NP_4PHY_ RDWR_REG	YES	1. Read and store LED register values.(backup) 2. Write an arbitrary value to LED register. 3. Read back the LED register. 4. Test passes if the value is the same. 5. Restore the backup value at step 1 to the register. (Restore) 6. Repeat 1~5 for all the ports (4 ports) for different PHY addresses.
UPDATE_PHY	UPDATE_PHY_ READ_OUI	No	1. Read OUI register. 2. If the value read is static OUI, the test passes.
	UPDATE_PHY _ RDWR_REG	YES	1. Read and store LED register values.(backup) 2. Write an arbitrary value to LED register. 3. Read back the LED register. 4. Test passes if the value is the same. 5. Restore the backup value at step 1 to the register. (Restore)
MGMT_PHY	MGMT_PHY_ READ_OUI	No	1. Read OUI register. 2. If the value read is static OUI, the test passes.
	MGMT_PHY _ RDWR_REG	YES	1. Read and store LED register values.(backup) 2. Write an arbitrary value to LED register. 3. Read back the LED register. 4. Test passes if the value is the same. 5. Restore the backup value at step 1 to the register. (Restore)
RTC	RTC_ I2C_PROBE	No	1. Read RTC I2C address 0 register. 2. Test passes if a response is received.
	RTC_	No	1. Read RTC and store the value.

[1] Intrusiveness. It may differ per target board application. The value here is a default value for the test. nbm_diag.conf may have different values.

	CHECK_TIME		2. Wait another 3 sec using the system clock. 3. Read RTC again and see the difference. 4. Test passes if the difference is 3~4 sec.
	RTC_ RDWR_REG	Yes	1. Read and store date register values.(backup) 2. Write an arbitrary value to date register. 3. Read back the date register. 4. Test passes if the value is the same. 5. Restore the backup value at step 1 to the register. (Restore)
PCIE_SWITCH	PCIE_SWITCH PCI_PROBE	No	1. See if PCIe switch is recognized after PCIe configuration cycle. 2. Test passes if the switch is recognized.
MAC_EEPROM	MAC_EEPROM_ I2C_PROBE	No	1. Read MAC_EEPROM I2C address 0 register. 2. Test passes if a response is received.
	MAC_EEPROM_ RDWR_REG	YES	1. Read and store entire MAC_EEPROM data.(backup) 2. Write predefined data to the entire MAC_EEPROM range. 3. Read back the data. 4. Test passes if two are identical. 5. Restore the backup value at step 1 to the EEPROM.(restore)
LMC_DRAM	LMC_DRAM_ SEQ_RDWR	No	1. Find the free memory size in the system. 2. Perform malloc for (free memory size*0.8). 3. Write incremental numbers in entire alloc memory. 4. Read the numbers in the memory. 5. Test passes if two are identical.
	LMC_DRAM_ SEQ_RDWR $SIZE	YES	Same procedure as LMC_DRAM_SEQ_RDWR except at step 2. In step 2, malloc is for a predefined size.
SDCARD	SDCARD_ PROBE	No	1. Test passes if SDCARD is present.
	SDCARD_ SEQ_RDWR	YES	1. Mount SDCARD. 2. Get SDCARD disk utilization (DU). 3. Generate a file on RAMDISK (TEST_FILE) and write random data for (DU size*0.8) size. 4. Copy TEST_FILE to SDCARD mount point. 5. Unmount SDCARD. 6. Mount SDCARD. 7. Compare RAMDISK TEST_FILE and SDCARD TEST_FILE.

			8. Remove SDCARD TEST_FILE. 9. Unmount SDCARD. 10. Test passes if no problem was indicated during 1~9.
	SDCARD_ SEQ_RDWR $SIZE	YES	Identical to SDCARD_SEQ_RDWR procedure except at step 2 and 3, TEST_FILE is fixed at predefined size.
NAND*n*	NAND*n*_ PROBE	No	1. Test passes if NAND*n* device is recognized. (NAND device shows up at mtd map). *n* = 0 or 1.
	NAND*n*_ SEQ_RDWR	YES	1. Copy NAND*n* usr1 region into RAMDISK. (backup) 2. Generate TEST_FILE on RAMDISK at the size of NAND0 usr1 region. 3. Copy TEST_FILE to NAND*n* usr1 region. 4. Copy NAND*n* usr1 region into RAMDISK. (TEST_FILE2) 5. Compare TEST_FILE1 and TEST_FILE2 contents. 6. Restore the backup to NAND*n* usr1 region.(restore) 7. Test passes if no problem was indicated 1~6. ※ Implemented with MTD tool
	NAND0_ SEQ_RDWR $NAME	YES	Identical to NAND*n*_SEQ_RDWR procedure except that user defined name region is used instead of usr1.
NFC_FPGA			TBD.
BASE_CPLD	BASE_CPLD_ RDWR_REG	NO	1. Write predefined value to test register. 2. Read back test register. 3. Test passes if two are identical.
RTM_CPLD			N/A
SPI_FLASH*n*	SPI_FLASH*n*_ PROBE	No	1. Test passes if SPI_FLASH*n* device is recognized. (The device shows up at mtd map). *n* = 0 or 1.
	SPI_FLASH*n*_ SEQ_RDWR	YES	1. Copy SPI_FLASH*n* entire region into RAMDISK. (backup) 2. Generate TEST_FILE on RAMDISK at the size of SPI_FLASH*n* entire region. 3. Copy TEST_FILE to SPI_FLASH*n* region. 4. Copy SPI_FLASH*n* region into RAMDISK. (TEST_FILE2) 5. Compare TEST_FILE1 and TEST_FILE2 contents. 6. Restore the backup to SPI_FLASH*n* region.(restore) 7. Test passes if no problem was indicated 1~6.

			※ Implemented with MTD tool
	SPI_FLASH0_ SEQ_RDWR $NAME	YES	Identical to SPI_FLASH0_SEQ_RDWR procedure except that user defined name region is used instead of entire region.

3.6 Ethernet Requirements

3.6.1 Physical Requirements

3.6.1.1 Default Autonegotiation Settings

This section covers implementation of CETH22.

The ANPI1-A will have autonegotiation enabled by default on all interfaces as is standard for Radisys blades including an Ethernet switch.

3.6.1.2 Switch Diagnostics

This section covers the design and implementation of requirement CETH108.

Switch diagnostics functionality is still being determined.

3.6.1.3 Management of Energy Efficient Ethernet Features

This section covers the design and implementation of requirement CETH224.

The Ethernet CLI for the fabric switch allows enable/disable of the IEEE 802.3az (a.k.a. Energy Efficient Ethernet or EEE) functionality on a port-by-port or global basis. The syntax for EEE commands is given below.

Table 32. CLI Commands for Energy Efficient Ethernet Feature

CLI Commands	Comments
config [interface <slot/port>] [no] dot3az	This command configures the interface (or all interfaces) to use IEEE 802.3az functionality for energy efficiency.
show dot3az [<slot/port> | all]	Shows enable/disable state for EEE on interface(s)
config no monitor session <session-id>	Disables monitor session

While no explicit method is given for showing the number of times that EEE has been enabled for a particular interface, it is possible, using lower level Broadcom diagnostic shell commands, to see the related internal counters.

The default configuration for the dot3az feature will be disabled.

The outputs for these internal commands are sent to the /var/run/switchdrvr/fabric/switchdrvr.log file. To see the output for the RX_EEE_LPI_EVENT_COUNTER registers, use the command:

bcmdiag “get rx_eee_lpi_event_counter”

To see the output for the RX_EEE_LPI_DURATION_COUNTER registers, use the command:

bcmdiag “get rx_eee_lpi_duration_counter”

3.6.2 General SW Requirements for Switches

3.6.2.1 Verification of Configuration File

This section covers the design and implementation of requirement CETH30.

This functionality was originally implemented for the AHUB3 blade and is being merged into the ANPI1-A codebase.

Whenever the blade configuration file is loaded into the blade via TFTP/SFTP the syntax of the file is checked for correctness before being applied. If any syntax is found to be incorrect, the configuration file is not applied and the previous configuration is maintained.

The MCLI "copy" command is used to upload configuration into the ANPI1-A. Details of this command can be found below.

Syntax: copy <source> <destination>

Depending on which options you select, this command:

- saves and then uploads the specified configuration file to the designated URL or
- reloads the specified configuration file so it becomes the running configuration.

Options:

<source> Specifies the location of the file being copied.

<destination> Specifies where to copy the file. Possible values for <source> and <destination> are:

Possible values	Entering this value...
nvram:startup-config	Reloads the configuration file to RAM and makes it the current configuration. Saves the current running configuration to permanent storage on the module or to another system. If saved to permanent storage, the saved configuration is reloaded when the module is rebooted.
system:running-config	Loads the configuration to RAM, making it the current running configuration.
`tftp:`<*url*>	Specifies a valid path reachable on the network using tftp.
`sftp:`<*url*>	Specifies a valid path reachable on the network using sftp.

3.6.3 Switching Requirements

3.6.3.1 Support for VLANs

This section covers the design and implementation of requirements CETH35, CETH36, CETH37 and CETH43.

The ANPI1-A supports VLANs on both the base and fabric interconnects. The network interfaces can be configured to belong to any VLAN between 1 and 4093.

The ANPI1-A supports VLAN operation per 802.1Q 2003, including support for dynamic registration of VLANs per GVRP.

VLANs can be used for forwarding broadcast, multicast, or unicast frames, or any combination thereof.

Port membership within each configured VLAN is selected by configuring the "port participation" for each port/VLAN combination. Ports can be explicitly included within a VLAN's member set or explicitly excluded from its member set. Alternatively, the port membership can be controlled dynamically by GVRP by first setting the port participation to "auto". In this case, each port is considered to not be in the VLAN's member set unless GVRP registers the VLAN on the port.

A default VLAN using VLAN ID 1 always exists. Other VLANs with arbitrary VLAN ID values in the range 2 through 4093 can be created statically using the CLI commands or SNMP objects, or they can be created dynamically using GVRP.

VLANs that were dynamically created via GVRP can be later made static.

Changes to VLAN configuration take effect immediately (there is no need to reset the LMP or the switching chip).

The CLI commands listed in the following table are used to view and modify the VLAN and GVRP configuration:

Table 33. CLI Commands for VLANs and GVRP

CLI Commands	Comments
[no] network mgmt_vlan <1-4093>	Configures the management Virtual LAN (VLAN) ID of the switch. This ID is used for accessing the switch's network interface.
vlan database [no] protocol group <groupid> <1-4093>	Associates the specified VLAN to the specified protocol-based VLAN group. Use the system-assigned numeric ID to identify the VLAN group. The VLAN group may only be associated with one VLAN at a time, but the association can be changed.
config interface <slot/port> [no] protocol vlan group <groupid>	Adds a physical interface to the specified protocol-based VLAN group.
config [no] protocol vlan group all <groupid>	Adds/removes all physical interfaces to/from the protocol-based VLAN group
vlan database [no] vlan <1-4093>	Create a VLAN instance
config interface <slot/port> [no] vlan acceptframe {admituntaggedonly | vlanonly | all}	Sets the frame acceptance mode per port
vlan database vlan association mac <macaddr> <1-4093>	Associates a source MAC address with a VLAN.
config interface <slot/port> [no] vlan ingressfilter	Enables/disables ingress filtering
vlan database [no] vlan makestatic <1-4093>	Changes a dynamic VLAN to a static VLAN
vlan database [no] vlan name <2-4093> <name>	Changes the name of a VLAN (to a blank string for the "no" case)
config interface <slot/port> [no] vlan participation <exclude/include/auto> <1-4093>	Configures VLAN participation for a port
config [no] vlan participation all {exclude | include | auto} <1-4093>	Configures VLAN participation for all ports

Table 33. CLI Commands for VLANs and GVRP

CLI Commands	Comments
config [no] vlan port acceptframe all {vlanonly | all}	Sets the frame acceptance mode for all interfaces. The VLAN tagged frames are forwarded in accordance with the IEEE 802.1Q VLAN Specification.
config [no] vlan port ingressfilter all	Enables/Disables ingress filtering for all ports
config [no] vlan port pvid all <1-4093>	Configures for Port VLAN ID for all ports
config [no] vlan port tagging all <1-4093>	Configures VLAN tagging for all ports for the specifies VLAN ID
vlan database [no] vlan association mac <macaddr> <1-4093>	Adds/removes a MAC address association
vlan database [no] vlan association subnet <ipaddr> <net mask> <1-4093>	Adds/removes an IP-subnet association
config [no] protocol vlan group all <groupid>	Adds/removes all physical interfaces to/from the protocol-based VLAN group
config vlan port priority all <0-7>	Configures the 802.1 port priority assigned for frames received on any switch port interface.
config interface vlan priority <0-7>	Configures the default 802.1p port priority assigned for untagged frames received on a specific interface.
config interface <slot/port> [no] vlan protocol group <groupid>	Adds/removes the specified port to/from the protocol-based VLAN group
config [no] vlan protocol group add protocol <groupid> <protocol>	Adds/Removes the protocol to/from the protocol-based VLAN
config [no] vlan protocol group remove <groupid>	Removes the protocol-based VLAN group
config interface <slot/port> [no] vlan pvid <1-4093>	Changes the port VLAN ID (PVID) per interface. The PVID is the VLAN ID assigned to untagged frames and priority-tagged frames received on the interface.
config interface <slot/port> [no] vlan tagging <vlan-list>	Enables egress tagging for this VLAN on the interface. Egress tagging preserves the outermost VLAN tag header on the VLAN's frames as they egress the interface.

The details of commands supported for VLAN configuration can be found in section 39 of [1].

3.6.3.2 Support for Clearing Switch Configuration

This section covers the implementation of requirement CETH251.

The switching software on the ANPI1-A provides the "clear config" command, as shown in the table below, to clear all switch configuration back to factory defaults.

Table 34. CLI Command for Clearing to Factory Defaults

CLI Commands	Comments
clear config	Resets the configuration to the factory defaults without powering off the switch. When you issue this command, a prompt appears to confirm that the reset should proceed. When you enter y, you automatically reset the switch

3.6.3.3 Support for Double VLAN Tagging (Q in Q)

This section covers the implementation of CETH238.

The ANPI1-A supports the use of Double VLAN Tagging (aka Double .Q, Q-in-Q and IEEE 802.1Q Tunneling). Double VLAN Tagging allows an additional VLAN tag to be added to frames that are already VLAN-tagged or to untagged frames. The EtherType field used to identify a Double VLAN-tagged packet can be configured as the IEEE 802.1Q standard value (0x8100), as the standard service provider value (0x88A8) or as a custom value (e.g., 0x9100). As with standard IEEE 802.1Q tagging, the VLAN ID (known as the customer ID) for a Double VLAN-tagged frame can be assigned based on the interface on which a frame is received. The legal range for customer IDs is 0 to 4095.

The figures below illustrate how VLAN tags are added when Double VLAN Tagging is enabled.

Figure 22 shows the receipt of an untagged frame. The switch adds a tag to this frame which uses the configured Double VLAN EtherType, in this case, 0x9100 and the VLAN ID assigned to the port (shown as PVLAN). When the frame egresses a port that does not have egress tagging enabled (this kind of port is often referred to as an Access Port), the VLAN tag is removed. When the frame egresses on a port that has egress tagging enabled for PVLAN (often referred to as Trunk Ports or Uplink Ports), the VLAN tag remains on the frame. The frame will only egress ports that are configured as participating in PVLAN.

When Double VLAN Tagging is enabled, VLAN tags with EtherTypes other than that configured with the "dvlan-tunnel ethertype" command are ignored for purposes of forwarding the frame. These tags will not be removed from the frame on egress regardless of which VLANs the egress port is a tagged member. Frames which are untagged at egress will be tagged using the EtherType configured with the "dvlan-tunnel ethertype" command. This is the case regardless of what mechanism is used to add the tag (e.g., protocol-based tagging, IP-subnet based tagging, etc.).

Figure 20. Receipt of Untagged Frame

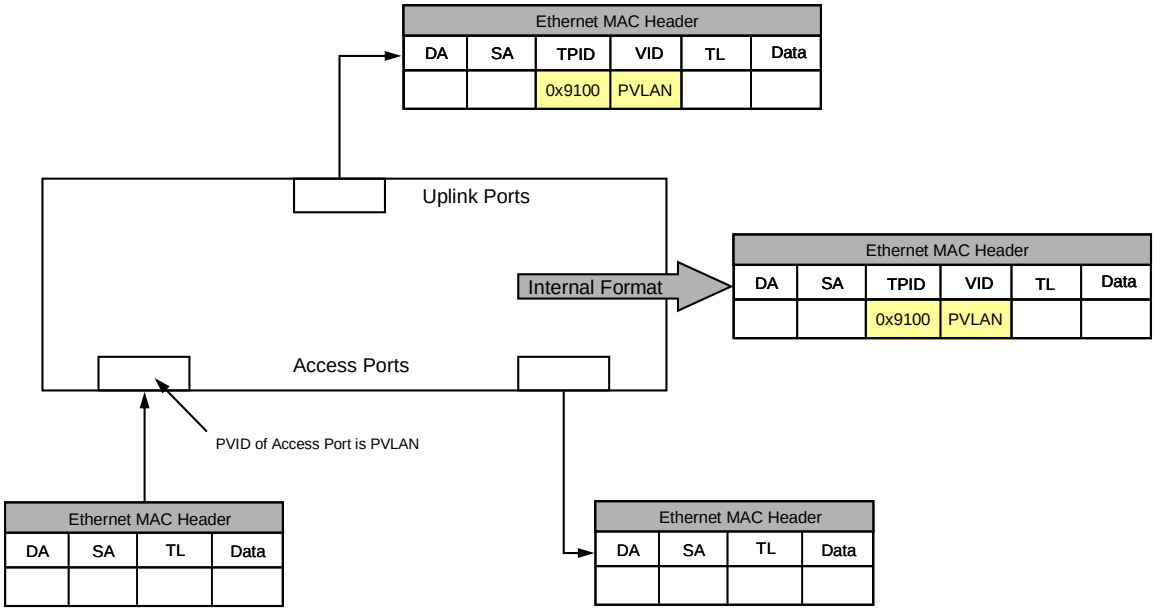


Figure 23 shows the receipt of a frame that is tagged with an IEEE 802.1Q VLAN Tag (EtherType is 0x8100). The switch adds an outer VLAN tag to this frame which uses the configured Double VLAN EtherType (0x9100 in this case), and the VLAN ID assigned to the port (shown as PVLAN). When the frame egresses a port that does not have egress tagging enabled, the outer VLAN tag is removed. When the frame egresses on a port that has egress tagging enabled for PVLAN, the outer VLAN tag remains on the frame. The frame will only egress ports that are configured as participating in PVLAN.

Figure 21. Receipt of IEEE 802.1Q Tagged Frame

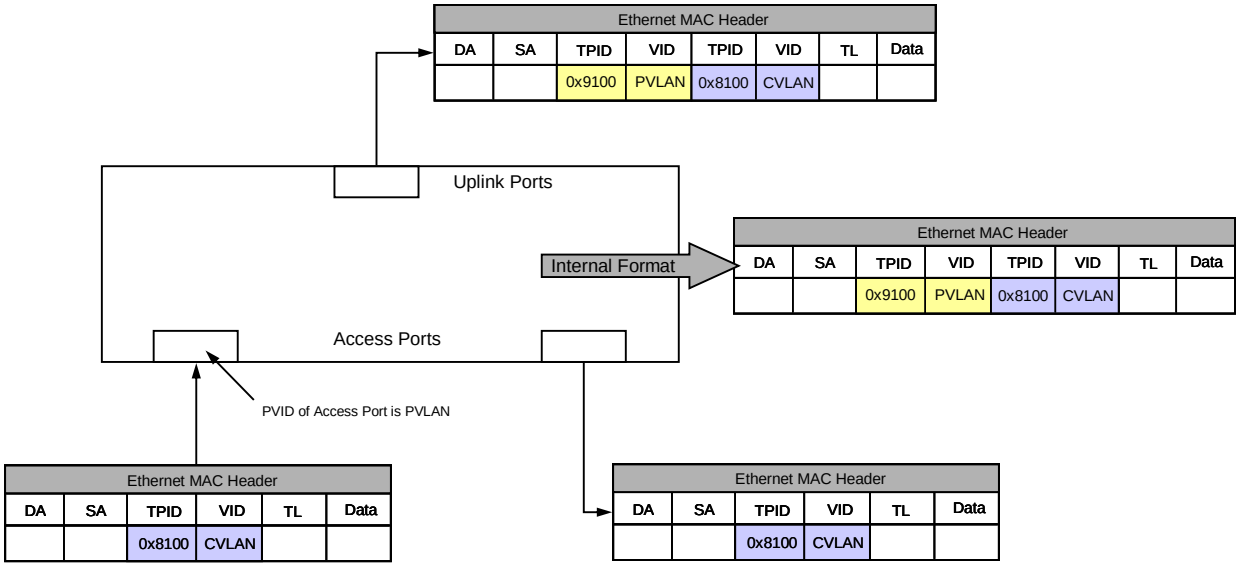
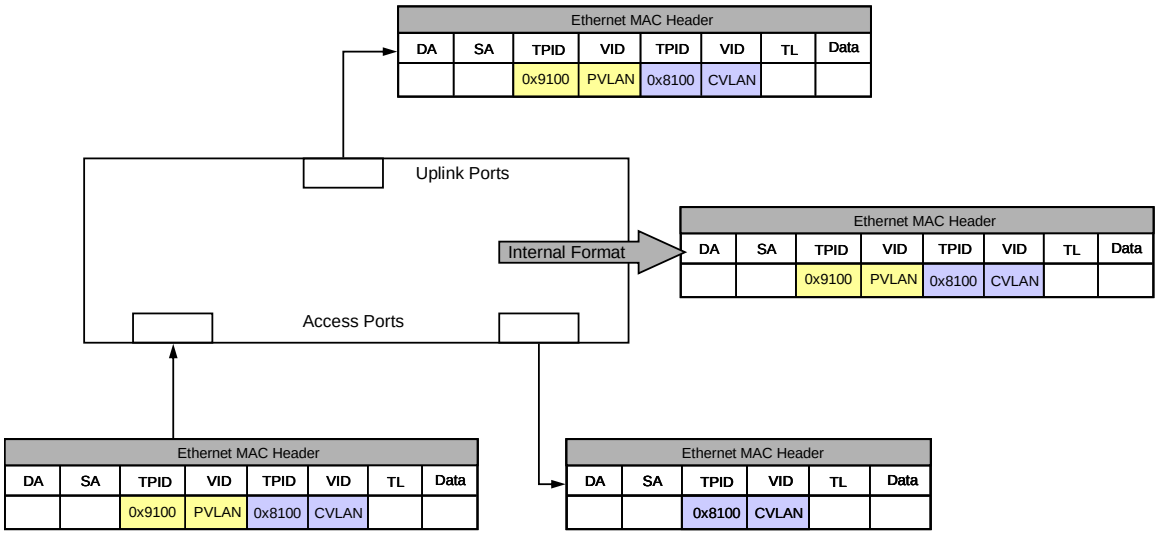


Figure 24 shows the receipt of a frame that already has a VLAN tag with the EtherType configured for Double VLAN tagging. The received frame will retain the outer VLAN tag while it travels through the switch. The frame will be forwarded to all ports that are members of PVLAN. For ports that are configured for egress tagging of the PVLAN, the frame will egress with both VLAN tags. For ports that are not configured for egress tagging of PVLAN, the frame will egress with only the original VLAN tag.

Figure 22. Receipt of Double VLAN-tagged Frame



The following CLI commands are used to configure Q-in-Q on the ANPI1-A.

Table 35. CLI Command for Clearing to Factory Defaults

CLI Commands	Comments
config [interface <slot/port>] dvlan-tunnel ethertype {802.1Q | custom [0-65535] | vman}	Configures the ethertype for all interfaces (Global Config) or a specific interface (Interface Config).
config [no] dvlan-tunnel ethertype default-tpid {802.1Q | custom [0-65535] | vman} [default-tpid]	Creates a new tag protocol identifier (TPID) and associates it with the next available TPID register. If no TPID registers are available, the system returns an error to the user.
config interface <slot/port> [no] mode dvlan-tunnel	Enables double VLAN tunneling on the specified interface. Interfaces enabled for double VLAN tunneling are service provider ports. Interfaces that do not have double VLAN tunneling enabled are customer ports.

3.6.3.4 Support for Management Traffic to the LMP

This section covers the design and implementation of requirement CETH44.

3.6.3.4.1 History of Implementation for CETH44

The Radisys-developed AHUB3 blade includes an implementation of the CETH44 requirement. This implementation is to be leveraged for the ANPI1-A. The following text describes the details of what is implemented and differences from the AHUB3 implementation.

3.6.3.4.2 Functionality Provided by Implementation

- Ability to specify a particular VLAN which is dedicated to carrying management traffic to/from the management processor on the ANPI1-A.
- Ability to specify unique VLAN IDs for carrying management traffic in both base and fabric domains.
- Ability to configure LAN-ports to belong to the management VLANs.
- Ability to assign a default priority (i.e., assign a CoS queue) for management traffic.
- Dissociation of Linux services (telnet, ssh, etc.) from all interfaces other than the management interfaces.

For the purposes of this requirement, management traffic is considered all traffic ingressing and/or assigned to the management VLAN and all traffic egressing the management interface of the ANPI1-A.

3.6.3.4.3 Implementation Details[2]

The AHUB3 hub blade used interfaces built into the Ethernet switch (also known as pseudo-interfaces) to carry management traffic. This was due to the fact that there weren't enough Ethernet ports on the fabric switch to spare one for the LMP interface. The ANPI1-A doesn't share this issue and so provides physical Ethernet interfaces to carry management traffic. For this reason, the handling of the management VLAN is somewhat different between the two blades. Where the "network mgmt._vlan" command was used to configure the management VLAN on the AHUB3, standard VLAN configuration commands are used to do the same on the ANPI1-A. For example, to configure VLAN 10 as the base network management VLAN, the following commands would be used:

```
vlan database
vlan 10
exit
configure
interface 4/1
vlan participation include 10
vlan participation exclude 1
vlan pvid 10
```

The "vlan pvid" command ensures that traffic received by the switch from the LMP is assigned to the management VLAN. The "vlan participation exclude" command ensures that traffic on VLAN 1 (the default management VLAN) is no longer sent to the LMP.

---

[2] This section assumes that a direct Ethernet link exists between the P2041 LMP and the Trident+ switch

Any egress Ethernet interfaces for management traffic must belong to the management VLAN. Management traffic received by Ethernet interfaces must either be tagged with the management VLAN, be assigned to the management VLAN via special assignment mechanisms (port-based, subnet-based, etc.), using the PVID for an interface, using ACL commands or using policy commands.

Note: Two VLAN assignment mechanisms, subnet-based VLANs and IP ACLs are not supported by the ANPI1-A base.

The ANPI1-A will use a management VLAN of 1 by default.

Finally, when the "mgmt-binding service strict" command is configured, the binding of services/daemons in the Linux running on the LMP will be restricted to IP addresses configured on the eth0 (service port), eth1 (base), and eth3 (fabric) interfaces. When new IP addresses are assigned to these interfaces via DHCP or the MCLI, the services are reconfigured as necessary to bind to those new addresses. Because of this, standard IP addresses otherwise supported by the ANPI1-A software (e.g., 10.1.<chassis>.<slot> on eth3:0) will no longer be usable for management operations. Only the addresses assigned by DHCP or the MCLI will be usable when the strict management binding is in force.

As part of this, default routes assigned to an interface via DHCP options (i.e., options 3 or 33) will be taken into use in the routing table associated with the interface on which the DHCP offer is received. Routing changes on their own, however, won't cause the re-binding of services to new IP addresses. So, if a change is made to the default route for an interface (eth0, eth1, eth3), that does not cause a re-binding of services.

If a particular service only allows one IP address to be bound to it, then only the IP address assigned statically (via MCLI) to the base management interface (eth3) would be used.

3.6.3.5 MCLI Commands for CETH44

3.6.3.5.1 Service Address Binding

This MCLI command will support enabling binding of management services on the ANPI1-A to the specific IP addresses assigned to the management interfaces (eth0, eth1, eth3) as opposed to the default case where service addresses are a wild card allowing access to the services from any management processor IP interface. The open case, which is the default, allows services to be accessed via any IP interface on the management processor. The strict binding case means that only IP addresses assigned to management interfaces via DHCP or MCLI commands will be bound to services. The default for this command is the open setting.

```
blade-mgmt config mgmt-binding service { open | strict }
```

3.6.3.5.2 Default Route Interface

In order to support the use of multiple default routes on the ANPI1-A, Policy Based Routing is brought into use. The basic approach is to use the source IP address of outbound traffic to determine which routing table to use. If the source IP address matches the address configured on serviceport (eth0) and then if there is a default route configured for the serviceport, it will be used. If the source IP address matches the IP address on the base interface (eth3), the default route for the base will be used, etc. However, when a connection is initiated from the LMP, there is no source IP address assigned so there is no defined way to choose which default route would be used if needed.

An MCLI command will support selecting which management interface should be used when a networking connection is initiated from the ANPI1-A LMP to an IP address that does not have a specific route. This command allows either the serviceport (eth0), base (eth1) or fabric (eth3) to be chosen. The default setting for this command is to use the base interface.

```
blade-mgmt config mgmt-binding default-route
{ serviceport | base | fabric }
```

3.6.3.5.3 Show Management Binding

A MCLI command will support showing the current address bound for management interfaces. Sources for IP addresses include the MCLI interface configuration commands as well as DHCP. This will also show the management binding default route if is defined, and if the management binding service is enabled.

```
blade-mgmt show mgmt-binding
```

Example output:

```
blade-mgmt show mgmt-binding

Management Binding

 Binding      : { open | strict }

 Default Route: { serviceport | base | fabric }

 Interfaces

   Serviceport : <operational IP>

   Base        : <operational IP>
   Fabric      : <operational IP>
```

3.6.3.6 Support for Port Mirroring

This section covers the design of the implementation for requirement CETH45.

The ANPI1-A supports port mirroring on any single pair of the interconnect processor's switch ports. One or more source interfaces are selected and a destination port is also selected. The destination port cannot be one of the source ports. In addition, the mirrored port(s) and the destination port should be of the same link speed, to accommodate full-bandwidth traffic, although this is not required.

> **Note:** Port mirroring is not supported across interconnects (i.e., base interconnect interfaces cannot be mirrored to fabric interconnect interfaces and vice versa.)

The various Ethernet interfaces on the LMP can NOT be selected as mirrored or probe ports.

The CLI commands listed in the following table are used to view and modify the port mirroring configuration:

Table 36. CLI Commands for Port Mirroring

CLI Commands	Comments
show monitor session <session –id>	Displays configuration of port mirroring
config monitor session <session-id> source interface <slot/port> destination interface <slot/port>	Configures port mirroring session
config no monitor session <session-id>	Disables monitor session

Port mirroring on the base and fabric interconnects is supported as described in section 32 of [1].

3.6.3.7 Support for VLAN Mirroring

This section covers the design of the implementation for requirement CETH229.

This functionality is supported using “class-map” functionality of the FASTPATH software package. An example of how to mirror VLAN 10 from interfaces 1/1 and 1/2 to interface 1/3 is given below. An overview of the matching/policy commands can be found in section 3.6.5.5 and details of the commands used in this example can be found in chapter 50 of [1].

```
vlan database
vlan 10
exit
config
class-map match-all vlan-class ipv4
match vlan 10
exit
policy-map vlan-policy in
class vlan-class
mirror 1/3
exit
exit
interface 1/1
no shutdown
vlan participation include 10
vlan tagging 10
service-policy in vlan-policy
exit
interface 1/2
no shutdown
vlan participation include 10
vlan tagging 10
service-policy in vlan-policy
exit
interface 1/3
no shutdown
exit
```

3.6.3.8 Lossless Handling of Traffic at 99% Line Rate

This section covers the configuration of switch parameters to meet requirements CETH46 and CETH47. These requirements are met only by the fabric switch.

This requirement is tested by sending 10Gbps multicast and broadcast into one interface on the fabric and ensuring that all traffic is received on all 10Gbps egress ports when 99% line rate is used with different packet sizes up to the jumbo frame size of 9212 bytes.

3.6.3.9 Number of Multicast Groups Addresses Supported by Switch

This section covers requirement CETH52.

Only 128 multicast groups are supported for the entire base switch on the ANPI1-A.

3.6.3.10 Support for Jumbo Frames

This section covers the design and implementation for requirement CETH48 and CETH49.

The configuration of maximum packet size is done with the "mtu" command as described in section 31 of [1]. The standard Radisys default had previously been 9212 but will be 1518 (when there is no VLAN tag) for the ANPI1-A.

The range of values for the mtu command on the fabric switch is 1518 – 9216.

The mtu command is summarized below.

Table 37. CLI Command for Controlling Max Frame Size

CLI Commands	Comments
config interface <slot/port> [no] mtu <mtu size>	Configures the maximum transmission unit.

3.6.3.11 Support for Link Aggregation

This section covers the design of the implementation for requirements CETH53, CETH90 and CETH91.

The ANPI1-A supports the Link Aggregation protocol per IEEE 802.1AX (formerly IEEE 802.3 clause 43).

The ANPI1-A will support up to 10 interfaces being assigned to a link aggregation group.

Each link aggregation of switch ports is called a LAG trunk (or simply a trunk) or a port-channel, and individual switch ports may be administratively added and removed from a trunk independently of all other switch ports. Once configured, a trunk behaves like any other switch port. From the standpoint of network management, each trunk is given its own logical slot/port number for use in the CLI commands, and its own ifIndex for use in SNMP objects.

The Link Aggregation Control Protocol (LACP) is used to control dynamic membership of switch ports in a LAG trunk. When LACP is enabled on individual switch ports, and it detects multiple links available at both ends, then the member switch ports on both of the trunks will behave as a single aggregated switch port. Otherwise, the member switch ports will behave like individual switch ports, although they will continue to participate in LACP, if enabled. A LAG trunk may be configured to be dynamic or static.

The Ethernet switch chip is responsible for dynamically distributing frames over individual LAG trunk member links, and collecting the frames received over them. The FASTPATH management stack software is responsible for implementing the LACP and Marker protocols.

The summary of Link aggregation commands is given in the table below.

Table 38. CLI Commands for Link Aggregation Configuration

CLI Commands	Comments
config interface <slot/port> addport <logical slot/port>	Add port to LAG
config interface <slot/port> deleteport <logical slot/port>	Remove port from LAG
config interface <slot/port> lacp actor admin key <key>	Set the administrative value of the key for the link aggregation control protocol (LACP) actor.

Table 38. CLI Commands for Link Aggregation Configuration

CLI Commands	Comments
config interface <slot/port> lacp actor admin state <state>	Sets the administrative value for the actor state. This value is transmitted by the actor in link aggregation control protocol data units (LACPDUs).
config interface <slot/port> lacp actor port priority <priority>	Sets the value of the priority assigned to the aggregation port.
config interface <slot/port> lacp actor system priority <priority>	Sets the value of the priority associated with the LACP actor's system ID.
config interface <slot/port> lacp admin key <key>	Sets the administrative value of the key for the port-channel.
config interface <slot/port> lacp collector max-delay <delay>	Sets the port-channel collector max-delay.
config interface <slot/port> lacp partner admin key <key>	Sets the administrative value of the key for the protocol partner.
config interface <slot/port> lacp partner admin state <state>	Sets the administrative value of the actor state for the protocol partner.
config interface <slot/port> lacp partner port id <port-id>	Sets the value of the LACP partner port ID.
config interface <slot/port> lacp partner port priority <priority>	Sets the value of the priority assigned to the LACP partner port.
config interface <slot/port> lacp partner system id <system-id>	Specifies a 6-octet MAC Address to represent the system ID of the aggregation port's protocol partner.
config interface <slot/port> lacp partner system priority <priority>	Sets the priority associated with the partner's system ID.
config port-channel <name>	Configures a new port-channel (LAG) and generates a logical slot/port number for the port- channel.
config port-channel adminmode all	Enables all port-channels. To enable a single port-channel, use the no shutdown command.
config port-channel linktrap {<logical slot/port> | all}	Enables the link trap notifications for the port-channel (LAG).

Table 38. CLI Commands for Link Aggregation Configuration

CLI Commands	Comments
config port-channel load-balance <load-balancing_option>{<slot/port> | all}	Selects the load-balancing option used on a port-channel (LAG). Traffic is balanced on a port-channel by selecting one of the links in the channel over which to transmit specific packets. The link is a binary pattern created from selected fields in a packet.
config port-channel name {<logical slot/port> | all} <name>	Defines a name for the port-channel (LAG).
config port-channel system priority <priority>	Sets the port-channel system priority.
config interface <slot/port> port-channel static	Enables the static mode on a port-channel (LAG) interface. By default, the static mode for a new port-channel is disabled, which means the port-channel is dynamic. However, if the maximum number of allowable dynamic port-channels are already present in the system, the static mode for a new port-channel is enabled, which means the port-channel is static. You can use this command only on port-channel interfaces.
config interface <slot/port> port lacpmode	Enables Link Aggregation Control Protocol (LACP) on a port.
config port lacpmode enable all	Enables Link Aggregation Control Protocol (LACP) on all ports.
config [interface <slot/port>] port lacptimeout {actor | partner} {long | short}	Sets the timeout on a physical interface.

The possible load balancing options for “port-channel load-balance” (as required by CETH90 and CETH91) are:

Possible value	Description
1	Source MAC, VLAN, EtherType, and incoming port associated with the packet
2	Destination MAC, VLAN, EtherType, and incoming port associated with the packet
3	Source/Destination MAC, VLAN, EtherType, and incoming port associated with the packet
4	Source IP and Source TCP/UDP fields of the packet
5	Destination IP and Destination TCP/UDP Port fields of the packet
6	Source/Destination IP and Source/Destination TCP/UDP Port fields of the packet

Details of the link aggregation commands can be found in section 58 of [1]. The functionality defined there allows all six modes described in the ANPI1-A requirements [8].

3.6.3.12 Support for Flow Control

This section covers the implementation of requirement CETH64.

Each switch port supports flow control per clauses 31 and 32 of IEEE 802.3. The interconnects support the behaviors specified in IEEE 802.3x for receipt and generation of pause frames. The specified behaviors upon receiving pause frames are always enabled. However, the specified behaviors for generating pause frames can be enabled on a per-port basis, and apply only for ports configured for full-duplex operation.

The ANPI1-A internal buffer configuration is setup so that when flow control is enabled then there is no loss due to buffer overflow.

The following command is used to enable/disable flow control on the ANPI1-A interfaces.

Table 39. CLI Command for Controlling Flow Control

CLI Commands	Comments
config [nterface <slot/port>] storm-control flowcontrol	Enables 802.3x flow control for the switch globally on all full-duplex mode ports via the Global Config mode or for a specific port via the Interface Config mode. ***Note:*** 802.3x flow control works by pausing a port when the port becomes oversubscribed and dropping all traffic for small bursts of time during the congestion condition. This can lead to high-priority and network control traffic loss.
config storm-control flowcontrol all	Same as above only that it enables flow control for all interfaces.

3.6.3.13 Buffer Allocation for Prevention of Traffic Loss

This section covers the design of the implementation for requirement CETH65.

The switch buffering configuration of the ANPI1-A allows for maximum sized Ethernet frames destined for the same egress port to ingress on 12 ports simultaneously so that backpressure will prevent the loss of any of the frames.

3.6.3.14 Updating Configuration on Operational ANPI1-A

This section covers the implementation for requirement CETH131.

A CLI command is provided that can load a saved configuration from a TFTP server and apply it to the operational ANPI1-A. The syntax for this command is:

copy tftp://(ip-path) system:running-config

See section 3.6.2.1 for more information on this command.

3.6.3.15 Updating Persistent Configuration

This section covers the implementation for requirement CETH123.

The following CLI command is provided for making the current switch configuration persistent so that is applied upon blade restart.

Table 40. CLI Command for Saving Current Configuration Persistently

CLI Commands	Comments
copy system:running-config nvram:startup-config	Copies current switch configuration to non-volatile memory so that it is used to configure switch on restart.

3.6.3.16 Reset of Blade

This section covers the implementation of CETH126.

The following MCLI commands are used to reset the blade (either a cold or warm reset as described).

Table 41. CLI Command for Resetting Blade

CLI Commands	Comments
reset cold	Performs a cold reset of the module. A cold reset is performed by the IPMC on the module and it causes all memory to be cleared. If enabled, the appropriate SNMP trap will be generated as part of the reset operation.
reset warm	Performs a warm reset of the module, which is equivalent to cold reset. A warm reset is performed by the LMP on the module and only a portion of the memory is cleared. If enabled, the appropriate SNMP trap will be generated as part of the reset operation.

3.6.3.17 Support for CLI Command Logging

This section covers the implementation of CETH127 and CETH130.

The ANPI1-A provides support for logging all CLI commands to a configuration history file. The log file is **/var/log/mcli/cli-history.**log. The CLI commands for controlling the logging function are shown in Table 42.

Table 42. CLI Commands for CLI Command Logging

CLI Commands	Comments
blade-mgmt config [no] cli-logging	Enable/disable logging of CLI commands
blade-mgmt show cli-logging	Displays whether or not CLI command logging is enabled.

3.6.3.18 Support for Long Passwords

This section documents the implementation of CETH135.

The ANPI1-A uses the WR 4.3 password mechanism (passwd utility, etc.). This mechanism supports passwords longer than 16 characters. The practical maximum for the ANPI1-A is 72 characters.

3.6.4 Supported L3+ Protocols

3.6.4.1 Retrieving Boot File Name from DHCP ACK Message

This section covers the design of the implementation for requirement CETH94.

A custom version of the /etc/dhclient-exit-hooks script is used to check for the specification of a TFTP server by the system's DHCP server. The **$tftp_server_name** and **$bootfile_name** variables, if supplied to the dhclient-exit-hooks script, are used to download the update script which is subsequently run to perform the needed updates.

3.6.4.2 Support for Standard Linux Remote Login/File Transfer

This section covers the implementation for requirements CETH166, CETH168 and CETH169 as well as part of CETH132.

The WindRiver 4.3 Linux used in the ANPI1-A implementation includes support for telnet and SSH servers which provide remote access to the blade.

By default, the only created user name for the blade will be "root" and there will be no password by default. The Linux "passwd" command can be used to specify a password for any user account on the ANPI1-A so that the services described here are password protected.

The ANPI1-A will also provide an SFTP client so that the SFTP protocol can be used to transfer files to/from the blade.

3.6.4.3 Support for SNMP Including Get/Set/Get Bulk Operations, SNMP Traps and SNMP Trap Logs

This section covers the design of the implementation for requirement CETH70, CETH73 and CETH74, CETH75 and CETH140.

The ANPI1-A supports standard SNMP MIBs as shown in the table below. The Get/Set/Get Bulk, Traps and Trap Logs are supported as part SNMPv2 functionality.

Table 43 lists the MIB modules supported by the SNMP agent. For each MIB module, the table also references the associated Request for Comments (RFC) document if applicable, identifies whether the MIB module is supported by the master agent and/or one of its subagents, and provides support details for the MIB module.

Table 43. MIB Module Support

MIB	RFC document	Supported by	Support details
BRIDGE-MIB	RFC 1493	*switchdrvr*	*BRIDGE-MIB* on page 119
DISMAN-EVENT-MIB	RFC 2981	*snmpd*	No major exceptions
EtherLike-MIB	RFC 2665	*switchdrvr*	*EtherLike-MIB* on page 119
FASTPATH-DENIALOFSERVICE	n/a	*switchdrvr*	No major exceptions
FASTPATH-QOS-COS-	n/a	*switchdrvr*	*FASTPATH-QOS-COS-MIB*
FASTPATH-QOS-	n/a	*switchdrvr*	*FASTPATH-QOS-DIFFSERV-PRIVATE-MIB* on page 120

HPI-B0101-MIB	n/a, SA Forum Web site	*hpiSubagent* on the ATCA-2210 and the ATCA-23xx series only	*HPI-B0101-MIB* on page 120
IEEE8023-LAG-MIB	n/a, IEEE Web site	*snmpd* and *switchdrvr*	*IEEE8023-LAG-MIB* on page 120
IF-MIB	RFC 2863	*snmpd* and *switchdrvr*	*IF-MIB* on page 120
IP-MIB	RFC 2011	*snmpd*	*IP-MIB* on page 120
LLDP-MIB	n/a, IEEE Web site	*switchdrvr*	*LLDP-MIB* on page 121
NET-SNMP-AGENT-MIB	n/a, Net-SNMP Web site	*snmpd*	*NET-SNMP-AGENT-MIB* on page 121
NOTIFICATION-LOG-MIB	RFC 3014	*snmpd*	No major exceptions
P-BRIDGE-MIB	RFC 2674	*switchdrvr*	*P-BRIDGE-MIB* on page 121
Q-BRIDGE-MIB	RFC 2674	*switchdrvr*	*Q-BRIDGE-MIB* on page 121
RADISYS-BLADE-MGMT-	n/a	*mclid*	No major exceptions
RADISYS-EXT-BRIDGE-	n/a	*switchdrvr*	No major exceptions
RADISYS-EXT-EtherLike-	n/a	*switchdrvr*	No major exceptions
RADISYS-EXT-IEEE8023-	n/a	*switchdrvr*	No major exceptions
RADISYS-EXT-IF-MIB	n/a	*switchdrvr*	No major exceptions
RADISYS-IGMP-SNOOPING-MIB	n/a	*switchdrvr*	No major exceptions
RADISYS-NTS-ATCA-MIB	n/a	*mclid*	*RADISYS-NTS-ATCA-MIB*
RFC1213-MIB	RFC 1213	*snmpd*	*RFC1213-MIB* on page 127
RMON-MIB	RFC 2819	*switchdrvr*	*RMON-MIB* on page 127
SNMP-COMMUNITY-MIB	RFC 3584	*snmpd*	*SNMP-COMMUNITY-MIB* on
SNMP-FRAMEWORK-MIB	RFC 3411	*snmpd*	No major exceptions
SNMP-MPD-MIB	RFC 3412	*snmpd*	No major exceptions
SNMP-NOTIFICATION-	RFC 3413	*snmpd*	No major exceptions
SNMP-TARGET-MIB	RFC 3413	*snmpd*	No major exceptions
SNMP-USER-BASED-SM-	RFC 3414	*snmpd*	No major exceptions
SNMPv2-MIB	RFC 3418	*snmpd*	*SNMPv2-MIB* on page 127
SNMP-VIEW-BASED-	RFC 3415	*snmpd*	No major exceptions
TCP-MIB	RFC 2012	*snmpd*	*TCP-MIB* on page 127
UDP-MIB	RFC 2013	*snmpd*	No major exceptions

Notifications are supported. Before notifications are generated:

- Destinations must be configured to receive trap or inform notifications.
- The SNMP agent must be enabled
- Notification generation must be enabled for the notifications you want to receive. You can configure notification generation using CLI commands, SNMP objects, or an API.

The following notifications can be enabled using the command “snmp-trap <parameter> from the CLI.

Table 44. CLI snmp-trap parameters

snmp-trap parameter	Notification
all	All traps list below
coldstart	coldStart
config-file	bladeConfigLoadFailure
drop-packet	bladeDropPacketError
frame-errors	bladeCrcError, bladeOversizeFrameError, bladeRuntError
linkstate	linkDown, linkup
lmp-overload	bladeLmpOverload
nts-bused-resource	ntsAtcaBusedResourceCommand
nts-device-status	ntsAtcaDeviceStatusChange
restart	nsNotifyRestart
security	bladeLoginFailure
signalchange	reDot3PortSignalStatus
traplogsfull	bladeTrapLogsFull

Further details of SNMP support can be found in Chapter 7 of [2].

3.6.4.4 Support for Storm Control for Unicast/Multicast/Broadcast

This section describes the implementation of CETH142.

The commands provided for storm control are shown below.

Table 45. CLI Commands for Storm Control

CLI Commands	Comments
config [interface <slot/port>] [no] storm-control broadcast	Enables/disables broadcast storm recovery mode for all interfaces (Global Config) or a specific interface (Interface Config). In this mode, traffic drops if the rate of incoming L2 broadcast traffic is more than the configured threshold.
config [interface <slot/port>] [no] storm-control broadcast rate <0-14880000>	Specifies a broadcast storm recovery threshold in packets per second (pps) for all interfaces (Global Config)or for a specific interface (Interface Config). The rate of incoming broadcast traffic is limited to the configured threshold. The “no” form of the command resets the default rate of 0.

config [interface <slot/port>] [no] storm-control broadcast threshold <0-100>	Configures the broadcast storm recovery threshold as a percentage of port speed for all interfaces (Global Config) or a specific interface (Interface Config). The rate of incoming broadcast traffic is limited to the configured threshold. The “no” form of the command resets the default value of 5.
config [interface <slot/port>] [no] storm-control flood	Enables/disables flood storm recovery mode for all interfaces (Global Config) or a specific interface (Interface Config). In this mode, traffic drops if the rate of incoming L2 flood traffic is more than the configured threshold.
config [interface <slot/port>] [no] storm-control flood rate <0-14880000>	Specifies a flood storm recovery threshold in packets per second (pps) for all interfaces (Global Config)or for a specific interface (Interface Config). The rate of incoming flood traffic is limited to the configured threshold. The “no” form of the command resets the default rate of 0.
config [interface <slot/port>] [no] storm-control flood threshold <0-100>	Configures the flood storm recovery threshold as a percentage of port speed for all interfaces (Global Config) or a specific interface (Interface Config). The rate of incoming flood traffic is limited to the configured threshold. The “no” form of the command resets the default value of 5.
config [interface <slot/port>] [no] storm-control multicast	Enables/disables multicast storm recovery mode for all interfaces (Global Config) or a specific interface (Interface Config). In this mode, traffic drops if the rate of incoming L2 multicast traffic is more than the configured threshold.
config [interface <slot/port>] [no] storm-control multicast rate <0-14880000>	Specifies a multicast storm recovery threshold in packets per second (pps) for all interfaces (Global Config)or for a specific interface (Interface Config). The rate of incoming multicast traffic is limited to the configured threshold. The “no” form of the command resets the default rate of 0.
config [interface <slot/port>] [no] storm-control multicast threshold <0-100>	Configures the multicast storm recovery threshold as a percentage of port speed for all interfaces (Global Config) or a specific interface (Interface Config). The rate of incoming multicast traffic is limited to the configured threshold. The “no” form of the command resets the default value of 5.

3.6.4.5 Support for TFTP Client

This section covers the design of the implementation for requirement CETH76.

A TFTP client is provided for the ANPI1-A as a standard part of the Wind River Linux 4.3. The open-source Linux version of the client is used.

3.6.4.6 Recovery of Self-Test Results via SNMP

This section covers the design of the implementation for requirement CETH173.

The MIB definitions, etc., needed to meet this requirement will be developed by Radisys and reviewed with NSN to ensure the needed functionality is implemented.

3.6.4.7 Support for Extended Capabilities in Access Control Lists

This section covers the design of the implementation for requirement CETH174.

ACL functionality is not supported on the base switch.

ACLs are supported on both the IP and MAC levels.

A summary of the fundamental ACL commands is given in the table below.

Table 46. CLI Commands for Access Control Lists

CLI Commands	Comments
config access-list <1-99> [deny | permit] [every | <srcip> <srcmask>] [assign-queue <0-3>] [log][mirror <slot/port>] [redirect <slot/port>][time-range <name>]	Creates an IP Access Control List (ACL) identified by the access list number for either standard or extended ACLs, and assigns it matching conditions.
config [interface <slot/port>] ip access-group [<acl-number> | <name>] [in | vlan <vlan-id> in] [sequence <seq-number>]	Attaches an IP Access Control List (ACL) to an interface or associates it with a VLAN ID in the direction specified.
config ip access-list <name>	Creates an extended IP Access Control List (ACL), consisting of classification fields defined for the IP header of an IPv4 frame, and changes the CLI to IPv4 Access-list Config mode.
config ip access-list <name> {permit | deny} {<protocol> | every } <src-ip> <src-mask> [eq <src-port-key>] <dest-ip> <dest-mask> [eq <dest-port-key>] [precedence <precedence> | tos <tos> <tos-mask> | dscp <description>] [log] [assign-queue <queue-id>] [{mirror | redirect} <slot/port>]	Creates a new rule for the current IP access list and appends the rule to the list of configured rules for the IP access list. (An implicit *deny all* IP rule always terminates the access list.) A rule either permits or denies traffic, according to the specified classification fields.
config ipv6 access-list <name>	Creates an extended IP Access Control List (ACL), consisting of classification fields defined for the IP header of an IPv4 frame, and changes the CLI to IPv6 Access-list Config mode.

Table 46. CLI Commands for Access Control Lists

CLI Commands	Comments
config ipv6 access-list <name> {permit | deny} { <protocol> | every } [log] [assign-queue <queue-id>] [{mirror | redirect} <slot/port>]	Creates a new rule for the current IPv6 access list and appends the rule to the list of configured rules for the IP access list. (An implicit *deny all* IPv6 rule always terminates the access list.) A rule either permits or denies traffic, according to the specified classification fields.
config mac access-list extended <name>	Creates a MAC Access Control List (ACL) consisting of classification fields defined for the Layer 2 header of an Ethernet frame.
config [interface <slot/port>] mac access-group <name> [vlan <vlan-id>] [<direction>] [sequence <seq-number>]	Attaches a MAC Access Control List (ACL) to an interface, or associates it with a VLAN ID, in the specified direction for a specific interface (Interface Config) or for all interfaces (Global Config). In Interface Config mode, this command is available only on platforms that support independent per-port class of service queue configuration.
config mac access-list extended <name> {permit | deny} { [<src-mac> | any] <srcmacmask> } {[<dest-mac> | any] <dstmacmask>} {[assign-queue | cos | log | mirror | redirect | vlan] <dstmacmask> [queue-id][<0x06000-0xffff> | <ethertypekey>]}	Creates a new rule for the current MAC access list and appends the rule to the list of configured rules for the MAC access list. A rule either permits or denies traffic, according to the specified classification fields. An implicit *deny all* MAC rule always terminates the access list. The “cos” qualifier is used to match the 802.1p CoS bits in the VLAN tag.

For the fabric switch, both IP and MAC-based ACL commands are described in detail in sections 51 and 52 of [1]. For the IP-based ACLs, additional functionality is added to match ICMP type/class data.

The FASTPATH permit/deny command for IPv4/IPv6 ACLs has additional qualifiers added to meet the CETH174 requirement. These additional qualifiers are:

icmp-type <type number>

icmp-class <class number>

These fields will only be configurable if the ICMP protocol is also specified in the permit/deny criteria.

In face-to-face meetings in December, 2012, it was explained by NSN that the full functionality defined in CETH174 is only required on the NP-4 parts and so the capabilities described above will be all that is supported by the fabric switch.

Radisys will test ACL functionality using its normal test suite.

3.6.5 Quality of Service (QoS)

3.6.5.1 Limitations on Support of QoS Priority Queues

This section covers the design of the implementation for requirement CETH232, CETH79 and CETH200.

Only 6 QoS queues are supported on the base switch. Of these six, two are dedicated for hardware-specific functions leaving only four for normal QoS purposes. The eight QoS classes supported by FASTPATH are mapped to these four queues as follows:

Table 47. Packet Priority to Queue Mapping

Packet Priority	BCM53115 Egress Queue
0	0
1	1
2	2
3	3
4	2
5	3
6	3
7	3

The CLI commands included to configure mapping is shown in the table below.

Table 48. CLI Commands for Priority to Class of Service Mapping

CLI Commands	Comments
config classofservice dot1p-mapping <user priority> <traffic-class>	Maps an 802.1p priority to an internal traffic class
config classofservice ip-dscp-mapping <ipdscp> <trafficclass>	Maps an IP DSCP value to an internal traffic class.
config [interface <slot/port>] classofservice trust {ip-dscp | ip-precedence | untrusted}	Changes the class of service trust mode of an interface from the default mode of dot1p to either the IP DSCP or the IP Precedence service trust mode, or used to set the interface mode to untrusted.
config [interface <slot/port>] cos-queue strict <queue-id-0> [<queue-id-2> ... <queue-id-7>]	Activates the strict priority scheduler mode for each specified queue on the interface (or all of the interfaces when the command is executed at the global configuration level). Activating strict scheduling sets the weight for the queue to 0.

Details of commands for mapping of 802.1p/802.1Q priorities to switch chip priorities (for both the base and fabric switches) can be found in section 42 of [1].

3.6.5.2 Limitations on Support for Deficit Round Robin

This section covers the design of the implementation for requirements CETH80, CETH81, and CETH86.

The Deficit Round Robin scheduling mode is not supported on the base switch.

On the fabric switch, the Weighted Round Robin scheduling is used by default. The type of scheduler used is configured using the following commands (also supported on the AHUB3).

Table 49. CLI Command for Controlling Max Frame Size

CLI Commands	Comments
config [interface <slot/port>] [no] cos-queue strict <queue id 0> [<queue id>]	This command configures the interface (or all interfaces) to use Deficit Round Robin scheduling.
config [interface <slot/port>] cos-queue strict <queue id 0> [<queue id>]	This command configures the specified queues on the interface (or all interfaces) to use strict priority scheduling.
config [interface <slot/port>] cos-queue weight <w0> <w1> … <w7>	This command configures the weight assigned to each of the 8 CoS queues.
config [interface <slot/port>] cos-queue wrr	This command configures the interface (or all interfaces) to use Weighted Round Robin scheduling.
config [interface <slot/port>] cos-queue drr	This command configures the interface (or all interfaces) to use Deficit Round Robin scheduling.

3.6.5.3 Implementation of Traffic Shaping Functionality

This section covers the implementation for requirement CETH88.

The following table describes the traffic-shape command implemented on the ANPI1-A.

Table 50. CLI Command for Traffic Shaping

CLI Commands	Comments
config [interface <slot/port>] traffic-shape <bw>	Sets the maximum transmission bandwidth limit for all interfaces or for a single interface. Traffic shaping functionality is supported on a per-port basis and has the effect of smoothing temporary traffic bursts over time so that the transmitted traffic rate is bounded. Traffic shaping can also be done at the CoS queue level.

The default value for traffic shaping is 100%.

3.6.5.4 Limitations on Queue Buffer Memory Allocation

This section covers the design of the implementation for requirement CETH82.

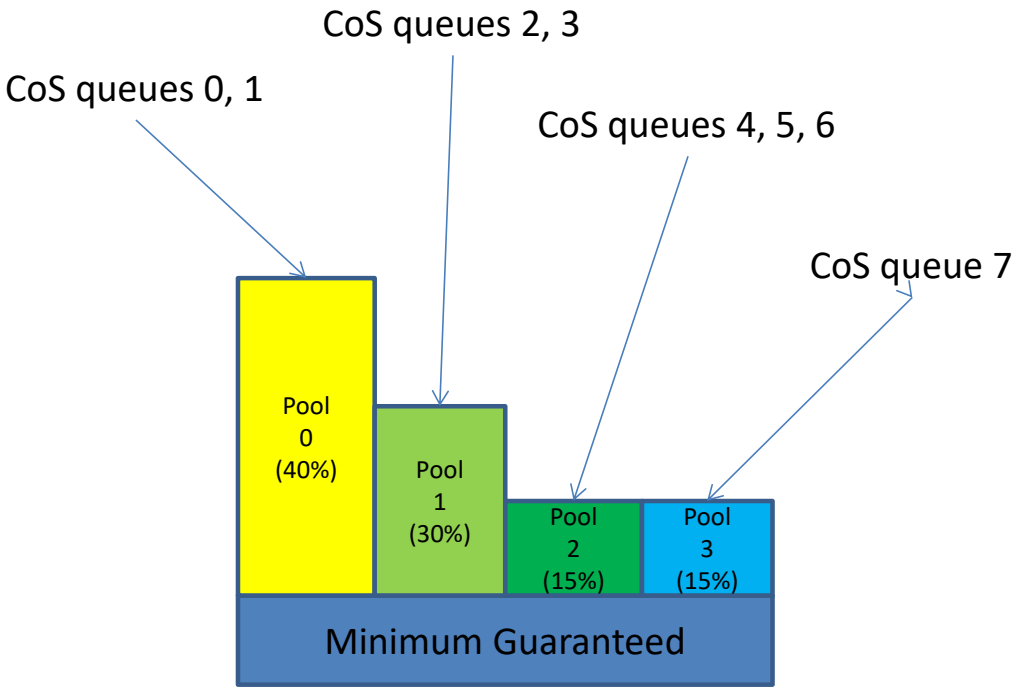
The buffer memory for egress traffic on each of the fabric interfaces can be configured. Four buffer pools are available and the size of the pools relative to the total amount of buffer space available can be configured. The buffer pool to use for each class of service on each interface can be configured. The proposed CLI commands for this functionality are shown in Table 51.

As an example of the use of these commands, consider the case where the following commands were used to configure the pool sizes and the queue-to-pool mapping for an interface.

config
cos-queue egress pool-size 40 30 15 15
interface 1/1
cos-queue queue-egress-pool 0 0 1 1 2 2 2 3

Figure 25 below shows the resulting sizing of the memory pools and the mapping of CoS queues for interface 1/1 to each of the pools.

Figure 23. CoS Queue and Memory Pool Configuration



Inside the Trident+ switch, the configuration of the pool sizes are defined using the OP_BUFFER_SHARED_LIMIT_CELL registers. The CoS queue to buffer pool mapping is defined using the OP_UC_PORT_CONFIG1_CELL registers for each port. Multicast queues are not configured by this command and use service pool 0.

Table 51. CLI Commands for Pool and Queue Configuration

CLI Commands	Comments
config cos-queue egress-pool-size <pool 0 %> <pool 1 %> <pool 2 %> <pool 3 %>	This command sets the relative size of each of four shared egress buffer pools. All the queues of each egress port are assigned to one of the four pools using the queue-egress-pool command described below. The default values are 100% for pool 0 and 0% for the three remaining pools. The sum of the four pool sizes must be less than or equal to 100.
Config [interface <slot/port>] cos-queue queue-egress-pool <queue 0> <queue 1> … <queue 7>	This command configures which pool is used for each of the 8 egress class of service queues. The default is pool 0 for all eight queues.

3.6.5.5 Limitations on Packet Marking

This section covers the design of the implementation for requirement CETH252.

Packet priority marking is not available on the base switch of the ANPI1-A.

All of the required marking capabilities are supported on the fabric with the exception of the MPLS EXP bits. MPLS EXP bits are expected to be assigned by the microcode based software executing on the EZChip.

The ANPI1-A supports the use of Differentiated Services as defined in RFC 2474, RFC 2475, RFC 2597, and RFC 2598. Frames received on an interface can be classified based on the following:

- Diffserv Codepoint (IPv4 Type of Service field or IPv6 Traffic Class octet)
- IP Protocol (ICMP, IGMP, TCP, UDP, Any)
- Source or Destination MAC Address
- Source or Destination IPv4 Address
- Source or Destination Layer 4 Port Number
- VLAN ID

Once a frame is assigned to a Diffserv traffic class, the policy associated with the class is used to determine conformance and perform policing. Policing of a traffic class can be configured using the FASTPATH “police-simple” command. This command allows a data rate and burst size to specified. Two sets of actions are configurable based on whether the traffic conforms to or violates the specified rate.

The possible actions per frame include:

- Drop the frame
- Mark the frame using the Diffserv Code Point field based on the applicable policy, and transmit the marked frame
- Mark the frame using priority field in the 802.1p header
- Transmit the frame as is

The following table contains a summary of the available commands.

Table 52. CLI Commands for DiffServ

CLI Commands	Comments
Class Commands	
[no] diffserv	Enables and disables diffserv function
class-map match-all <class-map-name>	Creates a DiffServ class
no class-map <class-map-name>	Eliminates an existing DiffServ class
class-map rename <class-map-name> <new-class-map-name>	Changes the name of a DiffServ class
match ethertype [<keyword> | custom <0x0600 – 0xFFFF>]	Adds a match condition based on ethertype to Diffserv class.
match any	Causes DiffServ class to match all packets
[no] match class-map <refclassname>	Adds a set of match conditions from another class to a DiffServ class.
match cos <0-7>	Adds a match condition for the Class of Server value specified
match destination-address mac <macaddr> <macmask>	Adds a match condition for the destination MAC address of a packet.
match dstip <ipaddr> <ipmask>	Adds a match condition based on the specified IPv4 destination address
match dstl4port { <portkey> | <0-65535> }	Adds a match based on the destination Layer 4 port. <portkey> can be *domain, echo, ftp, ftpdata, http, smtp, snmp, telnet, tftp, or www.*
match ip dscp <dscpval>	Adds a match based on the IPv4 DiffServ Code Point field in the packet
match ip precedence <0 – 7>	Adds a match condition based on the value of the IPv4 precedence field
match ip tos <tosbits> <tosmask>	Adds a match condition based on all eight bits of the Service Type octet
match protocol { <protocol-name> | <0-255> }	Adds a match condition based on the value of the IPv4 protocol field
match source-address mac <macaddr> <macmask>	Adds a match condition for the source MAC address of a packet.
match srcip <ipaddr> <ipmask>	Adds a match condition based on the specified IPv4 source address
match srcl4port { <portkey> | <0-65535> }	Adds a match based on the source Layer 4 port. <portkey> can be *domain, echo, ftp, ftpdata, http, smtp, snmp, telnet, tftp, or www.*
match vlan <1 – 4095>	Adds a match based on the VLAN ID field of the packet.
Policy Commands	
assign-queue <queueid>	Modifies the queue-id to which a packet is assigned
Drop	Specifies that packets be dropped
redirect <slot/port>	Redirect packets to a specified interface

conform-color <class-map-name>	Enable color-aware traffic policing and define the conform-color class map
[no] class <classname>	Associates a DiffServ class with the policy
mark cos <0 – 7>	Mark all packets with the specified class of service value
mark ip-dscp <dscpval>	Mark all packets with the specified DiffServ Code Point value
mark ip-precedence <0 – 7>	Mark al packets with the specified IPv4 precedence value
police-simple { <1-4294967295> <1-128> conform-action { drop | set-prec-transmit <0 – 7> | set-dscp-transmit <0-63> | set-cos-transmit <0 – 7> | transmit } [violate action {drop | set-prec-transmit <0 – 7> | set-dscp-transmit <0 – 63> | set-cos-transmit <0 – 7> | transmit }] }	Sets traffic policing style
[no] policy-map <policyname> in	Establishes (or eliminates) a DiffServ policy
policy-map rename <policyname> <newpolicyname>	Renames a DiffServ policy
Service Commands	
[no] service-policy in <policymapname>	Associates a DiffServ policy with an interface
Show Commands	
show class-map <classname>	Displays configuration information for a DiffServ class
show diffserv	Displays the general status for DiffServ
show policy-map <policyname>	Displays configuration information for the given policy map
show diffserv service <slot/port> in	Displays policy service information for the specified interface
show diffserv service brief	Displays DiffServ information for all interfaces in the system
show policy-map interface <slot/port> in	Displays policy-oriented statistics for the specified interface
show service-policy in	Displays a summary of policy-oriented statistics information for all interfaces

The details of these commands are given in chapter 50 of [1].

3.6.6 Factory Defaults

3.6.6.1 Reverting to Factory Defaults

This section covers the implementation for requirement CETH195.

The factory default configuration for the ANPI1 is kept in non-volatile storage either as a configuration file or via defaults built into a binary that is kept in non-volatile memory.

The blade can be returned to factory default settings using the command “erase nvram:startup-config”. The syntax for the erase command is given in the table below.

Table 53. CLI Command for Reverting to Default Configuration

CLI Commands	Comments
erase nvram:startup-config	Deletes the saved MCLI and Ethernet switching configuration files from the startup. Upon restart, the module's components will use the factory default configuration.

3.6.6.2 Factory Default Configuration for Networking Protocols/Services

This section covers the implementation for requirement CETH823.

The factory default configuration for the switch is as follows:

- All ports are enabled
- Autonegotiation is enabled on all ports supporting autonegotiation
- All ports belongs to default VLAN 1 (PVID=1). VLAN tagging is disabled for VLAN 1.
- Link aggregation is disabled.
- LACP is disabled
- GVRP is disabled
- LLDP is disabled
- RSTP is disabled, all backplane interfaces configured as edge ports.
- Management traffic is allowed

The factory default configuration for the unit computer is as follows:

- DHCP is enabled
- CLI time-out timer is disabled
- NTP client enabled

3.6.6.3 API for SFP Handling

This section describes the design and implementation for requirement EXIF224.

- Get the received power of SFP plugged into the port

```
RSYS_Esw2SFPRxPowerGet(
                       RSYS_Handle_t handle,
                       uint32_t      interfaceId,
                       uint32_t      *rxPower
                       )
```

Parameters

handle (IN) Specifies the remote RPC server,

interfaceId (IN) The interface ID of input port,

rxPower (OUT) The received power.

- Get the received power measurement type of SFP which is plugged into the port

```
RSYS_Esw2SFPRxPowerTypeGet(
                    RSYS_Handle_t handle,
                    uint32_t      interfaceId,
                    uint32_t      *rxPowerType
                    )
```

Parameters

handle (IN) Specifies the remote RPC server,

interfaceId (IN) The interface ID of input port,

rxPowerType (OUT) The received power type.

- Get the SFP presence detection

```
RSYS_Esw2SFPPresenceGet(
                    RSYS_Handle_t  handle,
                    uint32_t       interfaceId,
                    RSYS_Boolean_t isPresence
                    )
```

Parameters

handle (IN) Specifies the remote RPC server,

interfaceId (IN) The interface ID of input port,

isPresence (OUT) Is the SFP present.

- Get the name of the interface type used for the SFP

```
RSYS_Esw2SFPInterfaceTypeNameGet(
                    RSYS_Handle_t  handle,
                    uint32_t       interfaceId,
                    char           *typeName
                    )
```

Parameters

handle (IN) Specifies the remote RPC server,

interfaceId (IN) The interface ID of input port,

typeName (OUT) Interface type used for SFP (SGMII, SerDes, SFI, QSFP).

- Get the name of the SFP vendor

```
RSYS_Esw2SFPVendorNameGet(
                    RSYS_Handle_t  handle,
                    uint32_t       interfaceId,
                    char           *name
                    )
```

Parameters

handle (IN) Specifies the remote RPC server,

interfaceId (IN) The interface ID of input port,
name (OUT) The vendor name.

- Get the TX-fault value of the SFP

```
RSYS_Esw2SFPTXFaultGet(
                    RSYS_Handle_t  handle,
                    uint32_t       interfaceId,
                    RSYS_Boolean_t *txFault
                    )
```

Parameters

handle (IN) Specifies the remote RPC server,

interfaceId (IN) The interface ID of input port,

txFault (OUT) txFault is present (or not).

- Get the SFP LOS value of the SFP

```
RSYS_Esw2SFPLOSGet(
                    RSYS_Handle_t  handle,
                    uint32_t       interfaceId,
                    uint32_t       *los
                    )
```

Parameters

handle (IN) Specifies the remote RPC server,

interfaceId (IN) The interface ID of input port,

los (OUT) The LOS value.

3.7 External Interfaces and Module Interfaces

3.7.1 General Interface Requirements

3.7.2 External Ethernet Interface

3.7.3 Serial Ports

3.7.3.1 Serial Port Default Speed

This section covers the implementation for requirement EXIF180.

The default baud rate for the ANPI1-A serial ports will be 38.4Kbps.

3.8 Operating System

3.8.1 General for OS

This section covers the design of the implementation of operating system requirements in the OS9 group.

3.8.1.1 Linux Version Used on ANPI1-A

This section covers requirement OS91.

The ANPI1-A embedded software will be based on Wind River Linux version 4.3.

3.8.2 Board Support Package

This section covers the design of the implementation for requirement OS17.

3.8.2.1 Link State Change Notification Mechanism

This section covers the API functions supplied to provide a user application the ability to detect a link down event within 50ms as required by OS26.

A source code library will be provided that includes two functions which implement the needed mechanism. These two functions are:

- rsys_PortMoinitorThreadInit(void)
- rsys_PortMonitorCallbackRegister(void (*linkChangeCallback)(void))

When there is a change to the link status of any port, the callback function(s) registered by the rsys_PortMonitorCallbackRegister function will be called and appropriate action can be taken by the user application (using the ESWAPI2 API) to determine which port(s) have changed state and then perform any needed action.

4. Embedded Software Image release package

The embedded software released to NSN will contain the eSW package as described in 3.3.2.1. In addition the released eSW package will contain the accompanying documentation.

Radisys build and release process dictates the existence of change (build) logs with corresponding version numbers. This documentation will be delivered as part of eSW releases. [OS36]

Radisys will deliver this eSW Software specification which outlines and describes all non-trivial design choices and approaches. [OS37]

Radisys will deliver user level specifications and reference manuals that describes ANPI-1 product and how to configure, build and make. This document will be part of each eSW release. [OS38]

Radisys will deliver a separate document indicating all the SW packages within delivered BSP with corresponding licenses. [OS39]

Radisys will deliver “the list of test cases the vendor executes on the BSP (test plan) and the relevant test report” [OS 40].

A. References

A.1. Related Documents

A.1.1. Radisys Documents

1. *ATCA Command Line Interface Reference*. Radisys Corporation, September, 2012
2. *ATCA Software Guide,* Radisys Corporation, September, 2012
3. *ANPI1-A Hardware Specification.* Radisys Corporation, December, 2012
4. *ANPI1-A Test Plan.* Radisys Corporation, December, 2012
5. *ATCA-7240 Packet Processing Module,* Radisys Corporation, October 2012
6. *Ethernet Switching API, Version 2*,Radisys Corporation, September 2012

A.1.2. Other Documents

7. *ANPI1-A / NIRT3-A HW and SW Concept Document*. Nokia Siemens Networks, September 26, 2012
8. *AB5-RFQ_ANPI1-A_NIRT3-A_HPRS_v4_COMBINED_WORKING_FINAL.xlsx*. Nokia Siemens Networks, October 17, 2012
9. *BCM53115M Multiport Gigabit Ethernet Switch Data Sheet,* Broadcom Corporation, March 18, 2010
10. *BCM56842/BCM56844/BCM56846 Theory of Operation,* Broadcom Corporation, October 6, 2010
11. *Embedded SW Delivery Format Description for HW Platforms*, Nokia Siemens Networks, Version 3.0, May 7, 2012
12. *FRU Start Up in ATCA HW Platform*, Nokia Siemens Networks, Version 2.0, September 26, 2012 ATCA HW Platform
13. *ATCA Hardware Platform Management SW Specification*, Nokia Siemens Network, Version 8.0, June 09, 2010
14. *HPI FUMI – DIMI Implementation Guide*, Nokia Siemens Network, Version 1.1, February 18, 2011
15. *ATCA HW Diagnostics Specification*, Nokia Siemens Network, Version 1.0, February 24, 2011
16. *ATCA Blade HW Platform FRU Information Specification*, Nokia Siemens Network, Version 6.6, April 10, 2012

A.1.3. Industry-Standard References

17. *IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications,* IEEE, December 26, 2008

B. ANPI1-A Requirements Matrix

This section lists applicable product requirements and identifies where this specification addresses each requirement.

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST613	FRU shall have redundant Fabric Interface supporting 40GBase-KR4.	Refer to [3]	
AST269	FRU shall have redundant Base Interface.	Refer to [3]	
AST681	Base interface links shall be of type 10/100/1000Base-T/Tx-T.	Refer to [3]	
AST718	Unit shall provide segregation between Base interface and Fabric interface networks.	Refer to [3]	
AST719	Unit shall provide segregation of external ports from Base interface and Fabric interface networks.	Refer to [3]	
AST729	Unit shall provide segregation between the two sides of Base interface.	Refer to [3]	
AST730	Unit shall provide segregation between the two sides of Fabric interface.	Refer to [3]	
AST892	Automatic switchover to the "backup" eSW version shall be signaled by means of an IPMI event.	3.5.3.2.1	
AST624	The unit boot-up sequence shall follow the principles specified in the "General FRU start-up" document.	3.3.1.4	
AST740	DHCP DISCOVERY shall be repeated until valid OFFER and ACK are received from the server	3.3.1.4	See description of DYNAMIC boot in Table 10
AST501	Vendor shall provide a description that describes what embedded SW are upgradeable (remotely or locally) in the board.	3.1.3	
AST528	Vendor shall provide necessary SW upgrade tools and documentation.	3.1.3	A brief description of upgrade tools is provided. Detailed instructions will be provided in User Manual for the tools
AST527	SW packages delivered by the vendor shall apply a format specified by NSN .	3.1.3	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST875	The eSW tools/functions provided by the supplier shall NOT perform any kind of integrity check based either on the naming or on the contents of the IIF file defined in NSN's "Embedded SW Delivery Format in ATCA HW Platform" document. The document defines in what format the supplier must deliver the eSW components of a unit to NSN, but the conventions and rules therein defined are meant to be used solely by NSN. All the supplier must do is to deliver the eSW components following the rules defined by the document in question and leave the checks against the document exclusively to NSN.	3.1.3	
AST618	The eSW bundle that the vendor delivers to NSN shall contain the eSW images and relevant upgrade tools in a way that they can be used to upgrade all the updatable SW components without the need of additional SW	3.1.3	
AST409	It shall be possible to upgrade all embedded SW of the FRU remotely from the System Management function	3.5.4.2	Remote upgrade is performed using FUMI.
AST607	It shall be possible to perform the embedded SW upgrade over the Base Interface.	3.5.4	
AST410	Critical SW (e.g. BIOS, Boot, embedded Linux with BSP, file system, configuration files and embedded applications) shall be redundant with two copies (active and spare copy) in non-volatile memory	3.1.3	U-Boot, Linux and IPMC Application have redundant copies
AST462	Version information of the embedded SW shall be filled by the vendor to the local FRU Information (FRU Information specification is delivered from NSN).	3.1.3	
AST411	Embedded SW upgrade procedure shall not interfere FRU's normal operation during the upgrade	3.1.3	
AST413	If the upgrade procedure fails, it shall not interfere FRU's normal operation during the upgrade	3.1.3	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST414	If the upgraded SW found as faulty in the boot phase, automatic switch over to the spare copy shall be done	3.3.1.4	
AST463	After the successful upgrade the Embedded SW version information shall be updated to the local FRU Information.	3.1.3	
AST733	It shall be possible to read the eSW version within the unit from Linux using CLI and API	3.1.3 3.5.4.2.	The version are read using the CLI tools (nbm-upgrade and ipmitool), and HPI FUMI provides the API to read versions
AST758	For each embedded SW/FW component which has a spare copy for redundancy purposes (e.g. stored on two separate flash banks) it shall be possible to activate the eSW/FW spare copy by means of an IPMI command. The selected copy shall become the active copy on next blade boot. In other words, it shall be possible to select via IPMI command which of the eSW copies (stored on e.g. flash banks) shall become the active one at next blade boot.	3.5.2.2.2	Radisys “Switch Active Boot Flash” command is used to switch U-Boot and Linux banks, and HPM.1 “Initiate Manual Rollback” is used to switch the IPMC App banks.
AST870	For units with redundant flash banks the eSW versions that the user can install on the two banks shall be independent of each other, i.e. the user can install the same eSW version on both banks or two different eSW versions without limitations.	3.1.3	The upgrade process does not have any restrictions on which versions can be installed on each bank
AST871	For units with redundant flash banks the user shall be able to access the configuration data of both the active bank and the backup bank i.e., whichever bank is active at any moment, it shall be possible to access (read and write) the configuration data of the other bank without having to do a switchover.	3.1.3	Both Active and Standby banks are mounted as partitions that is accessible from Linux OS that is currently running on the blade.

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST872	The eSW and the eSW upgrade tools provided by the supplier shall not force the upgrading of any of the eSW components of the unit. Rationale is: if the O.S. is loaded over the network the user shall not be forced to upgrade the O.S. on the unit flash bank. This means that the user must be enabled to select what eSW components shall be upgraded during an upgrade run.	3.1.3	
AST873	It shall always be possible to upgrade a unit from any version to any other without the need to do intermediate upgrades.	3.1.3	
AST874	The eSW upgrade tools provided by the supplier shall check that the file containing the SW to be loaded onto the unit bank is not corrupted; should the file be found to be corrupted the eSW upgrade tool shall report an error.	3.1.3	
AST732	It shall be possible to upgrade all embedded software to the unit in a time less than 15 minutes.	3.1.3	
AST735	It shall be possible to access (read/write) the "non running" flash bank from the running O.S.	3.1.3	
AST878	For all the eSW components stored in multiple copies for redundancy purposes it shall be possible to read eSW version from both active and redundant banks.	3.1.3	
AST885	For all the eSW components stored in multiple copies for redundancy purposes it shall be possible to select which bank is used to load the eSW	3.5.2.2.2	See description for AST758
AST886	In case of redundant eSW located in multiple banks, bank switchover shall be detectable immediately via NSN defined OEM IPMI sensor and event shall be logged into SEL.	3.5.3.2.1	Radisys has OEM sensor for this purpose.
AST757	An ATCA Blade FRU containing embedded SW/FW components shall support an HPI FUMI interface, per correct HPI spec, for the upgrading of all the embedded SW/FW components	3.5.4.2	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST671	A FRU with subsidiary FRUs, such as a front board with AMC bays and a RTM shall support FUMI functionality for embedded SW on the subsidiary FRUs as well as for itself	3.5.4.2	
AST672	FUMI operations shall be done via logical banks	3.5.4.2	All FUMIs on ANPI1 and NIRT3 use logical banks
AST674	The FUMI interface shall be visible on the Base Interface	3.5.4	
AST675	The FUMI interface shall be compatible with an OpenHPI client	3.5.4	
AST676	All embedded SW version information shall be accessible through the FUMI interface	3.5.4.2	
AST677	In the context of eSW/FW upgrade the download/reflashing and the activation of the new eSW/FW shall be two separate phases. The download/reflashing shall not affect the functioning of the unit and shall not require unit reboot	3.1.3	
AST883	It shall be possible to activate all eSW within FRU at the same time.	3.1.3	
AST887	eSW downgrade where the existing eSW component is changed with smaller version number eSW, is considered as upgrade within these requirements.	3.1.3	The tools support upgrade from any version to any version, even the case when target is higher version than bundle image.
AST679	The eSW upgrade process for a single eSW component shall be activated with a single command	3.1.3	
AST680	The FUMI interface shall support TFTP as the eSW transfer protocol	3.5.4.2	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST734	eSW components should be backward compatible to previous (older) eSW versions in terms of existing settings, other eSW components, SW and HW interfaces. This means that activating a new eSW version to a unit shall maintain the functionalities and settings of the previously running version. In no case after eSW upgrade shall the unit become unresponsive or fail to resume its previous functionality.	3.1.3	
AST867	Changes in the eSW that cause backward incompatibility shall be accepted only in HW/ eSW development phase (i.e. before HW product P6). After P6 declaration, any changes that cause backward incompatibility need to be negotiated with and approved by NSN and, if agreed, they shall be clearly documented in the eSW release notes. If the supplier includes (per previous agreement with NSN) any changes in any eSW component which cause backward incompatibility in the HW product data (e.g. BIOS settings, UBoot environment variables and the like), the supplier shall provide NSN with conversion SW to automatically convert the data from the old format to the new. The conversion SW might be embedded in the eSW or it might be an external tool runnable under Linux: it is up to the supplier how to implement it. The tool shall be delivered to NSN together with the eSW drop that causes the backward incompatibility. After P6 changes that violate the backward compatibility shall always be considered exceptions.	3.1.3	
AST736	Upgrading to a new eSW revision must not rely on use of a specific (new) revision of the upgrade tool. If, for any reason, it is necessary to modify the eSW upgrade tools in a way that the user interface changes, such modifications shall be communicated and agreed with NSN beforehand	3.1.3	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST739	eSW upgrade tool shall check that the image has been successfully written to non-volatile memory and shall report anomalies during upgrade	3.1.3	
AST737	With each delivery to NSN of a new eSW drop a release note shall be provided which lists all modifications and provides detailed statements on backwards compatibility	3.1.3	
AST888	Each eSW delivery to NSN shall contain installation instructions.	3.1.3	
AST889	An ATCA Blade FRU User guide shall contain eSW installation and upgrading instructions.	3.1.3	
AST738	eSW upgrade tools and version inquiry tools shall be scriptable	3.1.3	The tools provided are Linux command line applications that are called from a script.
AST876	Supplier shall provide scriptable Linux tools that support upgrade of all eSW components of a FRU and allow upgrade granularity down to individual eSW components.	3.1.3	
AST772	HW change that causes backwards incompatible eSW change shall be detectable by means of a Linux tool from interchangeability code (i.e. by reading FRU information)	3.1.3	
AST465	The board vendor shall provide necessary and detailed documented APIs for the user space applications written by Nokia Siemens Networks.	Refer to [6]	
AST236	The unit shall be fully operational when the operating voltage range is -39.0 V DC to -72.0 V DC.	Refer to [3]	
AST237	The voltage range +75 V to -39 V and -72 V to -75 V shall be non-damaging.	Refer to [3]	
AST238	The unit undervoltage shutdown shall be done when the operating voltage is between -32V and -36V.	Refer to [3]	
AST851	Shutdown of payload shall be used and shall occur when voltage at unit input is at -37 +/- 0.5 V.	Refer to [3]	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST240	The undervoltage condition must persist for at least 100ms before shutdown of payload is allowed.	Refer to [3]	
AST241	Unit overvoltage shutdown shall be done when the operating voltage is 1.5V higher than maximum operating voltage specified for the unit.	Refer to [3]	
AST822	The power dissipation of the unit (single wide front blade) shall not exceed 255 W.	Refer to [3]	
AST354	The power dissipation of the unit (single wide RTM) shall not exceed 25 W.	Refer to [3]	
AST247	Redundancy of -48V power feed at the unit shall be verifiable.	Refer to [3]	
AST248	It shall be possible to detect any blown fuse on the unit.	Refer to [3]	
AST252	The holdup time of the unit shall be 5 ms minimum during voltage cut-off, when the initial voltage is higher or equal than -48V, and the FRU consumption is at maximum.	Refer to [3]	
AST860	Ripple on the on-board supplies shall be not more than 120 mV pp for 12 V, and not more than 50 mV pp for 5 V and 3.3 V.	Refer to [3]	
AST440	The unit shall implement operating voltage and current measurement, with values available to Shelf manager through IPMI and further to System manager.	3.5.3.1	
AST691	Internal power supplies of the unit shall be short-circuit proof, i.e. normal operation is restored after the removal of the short-circuit.	Refer to [3]	
AST257	The unit mechanics and connectors shall tolerate min. 250 insertions / extractions.	Refer to [3]	
AST686	Unit shall be compliant with NSN Industrial design requirement specification, latest version.	Refer to [3]	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST687	Pin contact phasing or some other efficient means shall be used to guarantee that the unit can be safely replaced without contact arching and latch-up risk.	Refer to [3]	
AST688	The unit shall be provided with locking mechanism to allow locking to its designated position e.g. by using thumb screws.	Refer to [3]	
AST852	It shall be possible to lock the connectors of any cables to avoid accidential disconnection of cables.	Refer to [3]	
AST853	Cable connectors shall be polarized connectors with a latching mechanism.	Refer to [3]	
AST258	Face plate EMC gaskets shall withstand 500 insertion/extraction cycles without degradation.	Refer to [3]	
AST360	NSN's preferred scheme: Face plates of front and rear blades are 1.0 - 1.2 mm thick sheet metal. Sheet metal material is cold rolled carbon steel or stainless steel or equivalent with RoHS compliant clear chromating surface treatment. Attached on the sheet metal there is 0.5 mm thick, face plate wide straight polycarbonate cover sheet, which hides the corner roundings of folded sheet metal.	Refer to [3]	
AST361	NSN's preferred scheme: Front panel polycarbonate sheet is Lexan or equivalent. It shall be mat finished/grained (martio). Background color is PMS 877 (NET silver). Color for texts and symbols are PMS Black CVC (NET black).	Refer to [3]	
AST359	NSN's preferred scheme: There is no AdvancedTCA logo, no AdvancedMC logo, no vendor logo, no NSN logo.	Refer to [3]	
AST362	In NSN's preferred scheme: Primary color for handles, retention screws and other possible parts relevant for the human operator is black.	Refer to [3]	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST363	NSN's preferred scheme: Font is Arial. Font size on FRU front panels is 6 - 9 pt. For readability, the minimum font size is 5.5 pt.	Refer to [3]	
AST747	NSN's preferred scheme: Handle for front blades and RTMs is Southco ATCA handle of the 500 series or equivalent Schroff model.	Refer to [3]	
AST748	NSN's preferred scheme: Hot swap switch is placed to work with the lower handle.	Refer to [3]	
AST857	The product identification related marking shall be according to latest version of "NSN Product Marking" document.	Refer to [3]	
AST420	The form factor shall be an ATCA single-wide blade for node slot.	Refer to [3]	
AST493	Empty transceiver cages shall be fitted with dust and EMI covers.	Refer to [3]	
AST502	Code for keying receptacle K1 shall be 11.	Refer to [3]	
AST503	Code for keying receptable rK1 shall be 00.	Refer to [3]	
AST621	State the K2/A2 keying set for the unit.	Refer to [3]	
AST264	The unit (single wide front blade and RTM) shall cope with the ambient as follows: Tamb Front board RTM 25°C 25 m3/h 3 m3/h 45°C 50 m3/h 6 m3/h 55°C 50 m3/h 6 m3/h	Refer to [3]	
AST858	The unit (front blade) shall present an air flow impedance of min. 45 MFI.	Refer to [3]	
AST859	The unit (RTM) shall present an air flow impedance of min. 1100 MFI.	Refer to [3]	
AST820	Unit shall not issue thermal events for temperature at shelf inlet below +45°C. For temperature at shelf inlet above +45°C UNC thermal events are allowed and for temperature above +55°C UC or UNR thermal events are allowed.	Refer to [3]	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST445	Unit shall be designed so that it is fully operational when temperature at Shelf inlet air is +42°C and one fan has failed. Unit can issue UNC temperature event but UC events are not allowable.	Refer to [3]	
AST205	The unit shall be fully operational without system clocks CLK1 and CLK2.	Refer to [3]	
AST453	The unit shall comply to PICMG 3.0 R3.0.	Refer to [3]	
AST454	The unit shall comply to PICMG 3.1 R2.0.	Refer to [3]	
AST461	Each FRU shall, as a part of ATCA platform, meet the safety and environmental requirements set in the Regulatory and Customer Compliance Requirements.	Refer to [3]	
AST690	Unit shall be fully RoHS compliant.	Refer to [3]	
AST755	Printed circuit board finish can be any RoHS compliant finish except for immersion silver (ImAg).	Refer to [3]	
AST756	Printed circuit board finish should be preferably immersion tin (ImSn).	Refer to [3]	
AST467	Unit (blade) failure rate shall not exceed 2 kFIT.	Refer to [3]	
AST468	Unit (AMC, or RTM) failure rate shall not exceed 1 kFIT.	Refer to [3]	
AST717	Supplier shall state the fit value of the unit.	Refer to [3]	
AST890	Vendor shall perform protocol robustness testing for all active protocols implemented in the product, and provide results of the testing.	Refer to [4]	
AST891	Vendor shall execute vulnerability scan to detect security holes in the product, and provide results of the testing.	Refer to [4]	
AST897	Vendor shall execute Port scan testing to detect security risks related to open TCP/UDP/SCTP ports in the product, and provide results of the testing.	Refer to [4]	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
AST898	Tool versions, test suites and plugins used in robustness testing, vulnerability scan and port scan must be either latest released version or not older than 30 days at the time of testing.	Refer to [4]	
AST499	Unit shall be shipped without any pluggable transceivers.	Refer to [3]	
AST645	Supplier shall indicate if any hw parts are under export restriction, and if there is a replacement part for avoiding the restriction.	Refer to [3]	
AST647	Supplier shall state if any software is under export restriction, and if there is a replacement software for avoiding the restriction.	N/A	Radisys will provide export complaint software to NSN. Radisys owns a license of Black Duck and will run the final software image through it to verify no export violation. The offending code will be stripped and an alternative solution will be provided.
AST505	Unit shall boot up in a time of 30 s counted from the moment it receives power to payload.	Section 3.1.4	
CPT305	It shall be possible to supply the unit without DRAM modules (DIMMs) if requested by NSN.	N/A	This is an operation/mfg requirement.
UNC214	Unit computer shall have access to both BI networks, including onboard application processors	Refer to [3]	
UNC123	Unit Computer bootloader shall be based on U-boot.	3.3.1	
UNC124	A U-boot and default U-boot configuration shall be provided	3.3.1	
UNC122	There shall be two identical copies of Boot firmware in the Flash memory.	3.3.1.2	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
UNC163	If DHCP ACK message includes boot file name option and next server, the embedded software shall fetch file pointed by that path and execute it. Rationale: this hook can be used for example for configuring the embedded software	3.3.1.5	
UNC125	Unit Computer shall support cold reset	3.5.3.2	
UNC126	Bootloader on unit computer shall configure and feed the external watchdog until OS is loaded.	3.3.1.4	
UNC128	It shall be possible to store all boot loader configurations in non volatile memory	3.3.1.6	
UNC129	Bootloader shall support OS loading from boot flash	3.3.1.6	
UNC130	Netboot shall be supported	3.3.1.6	The AB3 equivalent implementation of Netboot will be implemented on the ANPI1-A
UNC162	The ordering of the boot devices shall be changeable.	3.3.1.6	The AB3 equivalent implementation of Netboot will be implemented on the ANPI1-A
UNC131	DHCP and static configuration shall be supported for netboot	3.3.1.5	The AB3 equivalent implementation of Netboot will be implemented on the ANPI1-A
UNC221	Unit computer shall load the OS from network by default	3.3.1.6	The AB3 equivalent implementation of Netboot will be implemented on the ANPI1-A
UNC133	All errors detected during boot phase after UART is available shall be reported to console and to the memory location that is possible to analyze afterwards.	3.3.1.4	
UNC134	All errors detected during boot phase shall be logged to SEL.	3.3.1.4	
UNC136	Each boot flash shall contain bootloader, related environment variables, OS and applications	3.1.3	Refer to the “LMP Boot Image Storage” table

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
UNC137	Either boot flash shall be able to set active. The other software set is in standby role.	3.3.1.2	
UNC138	IPMI Controller shall have the responsibility for selecting active flash.	3.5.3.2.1	IPMC provides “Set Active Boot Flash” Command
UNC139	Boot flash shall contain checksum and version information for bootloader, environment variables and OS segments.	3.1.3	
UNC140	A checksum mismatch shall result in switching to backup boot flash.	3.3.1.4	
UNC141	A completely corrupt or missing bootloader shall result in switching to backup boot flash.	3.3.1.4	
UNC142	Reflashing (overwriting) boot flash shall be supported from Unit Computer.	3.1.3	Reflashing of boot flash and Linux from LMP as described in this section
UNC143	It shall be possible to reflash either boot flash without disturbing the normal operation.	3.1.3	Normal operation is not disturbed
UNC144	Reflashing bootloader shall not require immediate reset.	3.1.3	
UNC204	Watchdog shall prevent the processor hanging, e.g. because it is stuck in some infinite loop. Consequent actions of watchdog timeout expiration shall be described in the product documentation	3.3.1.4	
UNC242	For units with redundant flash banks whose bootloader is UBoot the possibility is required that the unit perform an automatic bank switchover in such conditions that would otherwise lead the unit to end up in an endless reset loop. Such possibility ought to be achieved by means of the UBoot variables "bootcount", "bootlimit" and "altbootcmd", which shall be configurable by the user.	3.3.1.1 Table 12	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
UNC146	The Power-On Self Test (POST) performs initialization and basic tests of the processor core and unit computer block components on which the processor depends. All peripherals of the unit computer block shall be tested.	3.3.1.7	
UNC220	Critical POST errors shall be logged and retrievable via IPMI (SEL). Non critical POST errors should be logged into SEL as well	3.3.1.7	
UNC147	Bootloader shall provide a set of tools for downloading and executing programs on the unit computer block and tools for manipulating the unit computer's environment. It shall be used for both product development (debug support) and for end product deployment	3.3.1.6	
UNC148	A Command Line Interface shall be activated (Bootloader breaks normal execution and enters console prompt) in the following cases: - After completion of POST when the startup script is disabled (default case). - When a specified key combination is pressed during POST execution (after UART ports have been initiated and tested) i.e user intervention is done during POST. In this case, the current phase of POST is completed first and then Bootloader enters the console prompt, waiting for user input. - When startup script is completed and there was no command executing operating software. - When the specified key combination was pressed during the execution of the startup script (currently executed command is finished first).	3.3.1.4	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
UNC149	The Bootloader console shall support at least: - Execution of download utilities: The user can download a file into a selected memory area using X/Y modem (via serial port) or TFTP (via Ethernet), and can manually define IP addresses necessary for TFTP transfer. - Execution of flash tools: The user can write a selected area of memory into Flash. Usually it will be a file previously downloaded into RAM. The image can also be loaded from Flash to RAM or can be deleted from Flash. - Definition of configuration: The user can define configuration parameters which include local IP addresses, default server's IP address or content of startup script. The configuration is stored in a dedicated sector of Flash. - Code execution: The user can execute the code located in selected address of RAM or Flash. - Diagnostics: It should be possible to run single tests or complete diagnostics for the entire unit computer block. Some tests can be executed with parameters that allow the user to modify test execution. Some tests can be repeated a specified number of times or run until Ctrl-C is pressed.	3.3.1.1	
UNC150	A startup script shall be executed on startup.	3.3.1.1	
UNC151	The startup script shall be stored in Flash together with other configration parameters.	3.3.1.1	U-Boot supports scripting that can be assigned to an env variable

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
UNC152	The startup script shall include a command to start operating software. The operating software will automatically take over control of the processor (it doesn't return to Boot Monitor).	3.3.1.1	
UNC153	The startup script configuration information format and content shall be approved by NSN. The startup script shall have NSN approved default values.	3.3.1.1	
UNC215	It should be possible to run all configuration and monitoring utilities from a script.		Radisys provides CLI, and API interfaces well as a plethora of executable and binaries which are all callable from the scripts.
UNC217	U-boot shall be able run U-boot script-images"	3.3.1.1	
UNC183	U-boot/BIOS must be VLAN-aware.	Refer to ADSP1 manual	NSN and Radisys have agreed that the functionality implemented for the ADSP1 blade would satisfy this requirement
UNC216	UBoot shall treat VLAN-A and VLAN-B as sub-options for netboot, which is one of the boot sources. When trying the boot from LAN Uboot shall try first to boot from VLAN-A, then from VLAN-B and then, if both are unsuccessful, move to the next boot source in a round-robin way until a valid boot source is found or watchdog bites.	3.3.1.4	NSN and Radisys have agreed that the functionality implemented for the ADSP1 blade would satisfy this requirement
UNC64	Documentation for Flash updating interface shall be provided in order to facilitate the use of proprietary updating SW for NSN proprietary OS.	Refer to chapter 4 of [2]	
UNC203	There shall be the possibility to restore the UBoot environment to the default factory values	Refer to chapter 16 of [2]	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
UNC218	All tunable u-boot settings shall have Nokia Siemens Networks approved default value	Refer to ADSP1 manual	NSN and Radisys have agreed that the functionality implemented for the ADSP1 blade would satisfy this requirement
UNC219	NSN unit name (e.g. AMPP2-A, ADSP2-A, SCNAM-C) shall be visible in an environment variable on u-boot	Refer to ADSP1 manual	NSN and Radisys have agreed that the functionality implemented for the ADSP1 blade would satisfy this requirement
UNC158	Minimum requirement for NSN SW build storage is 128MBytes of FLASH memory	Refer to [3]	
UNC207	Unit computer shall have at least 3072MB of RAM available for NSN application	Refer to [3]	
UNC161	Unit computer RAM shall be ECC capable.	Refer to [3]	
UNC224	Unit computer type shall be Freescale P2040 @ 1,2 GHz or equivalent/better from performance point of view.	Refer to [3]	
DPP83	The blade shall be based on dual EZChip NP-4 processors	Refer to [3]	
DPP91	Approved version of blade shall be based on 400MHz version of NP-4	Refer to [3]	Maximum frequency supported for NP-4 is 365MHz which is what is used on ANPI1-A.
DPP101	The vendor shall provide NP4 boot configuration in NP Script Language (NPsl) format. The boot configuration shall include all necessary register settings for physical buses/interfaces towards other HW devices (such as on-board switch, possible MAC devices etc).	3.4.1	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
DPP316	Each NPU shall presented as their own FRU.	Not compliant	The NPUs are not presented as their own FRU. The LMP and NPUs are treated as one FRU. Reviewed and agreed in 7-Nov FTF meeting.
DPP317	It shall be possible to reset packet processors individually, so that it wont´t affect rest of the board including other NPU and unit computer.	3.4.3	HW supports independent reset lines to the NPU that are controlled by a CPLD. A change will be implemented in CPLD to support individual resets to NPU. Also need to be added an OEM command in IPMC to support the individual resets to NPU
DPP318	It shall be possible to power down and power up NPU's individually	Not compliant	Hardware does not support individual power control to each NPU. Reviewed and agreed in 7-Nov FTF meeting.
DPP88	NP-4 SDRAM Search memory capacity shall be 4GBytes / NPU	Refer to [3]	
DPP89	NP-4 SDRAM TM memory capacity shall be 8GBytes / NPU	Refer to [3]	
DPP93	NP-4 shall support TCAM offload for L3 and L4 ACL lookup (IPv4 and IPv6)	N/A	The NPU units are complaint with this request. Different RTM with installed TCAM is required.
DPP94	TCAM shall support 1500 records with 160b keys or up to 3000 records with 32b-64b keys	Table 6 NL11K Datasheet Rev 2.2	
DPP95	TCAM shall support up to 40M decisions per second	Section 10.1 of NP-4 specification version 8.46	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
DPP329	Interfaces between switch and NPU shall support link aggregation based on GTP TEID and GRE key criteria.	Future	The method used to implement this functionality will be discussed by NSN and Radisys.
DPP92	Throughput of the blade shall be 80Gbps	Refer to [3]	
DPP330	NPU shall have access to both BI networks	Refer to [3]	
HPM280	The FRU shall be compliant to IPMI Platform Event Trap Format Specification v1.0	3.5.1	IPMC does not support Platform Event Traps (PET), and not planned to be added. Reviewed and agreed in 7-Nov FTF meeting.
HPM277	The FRU shall be compliant to Intelligent Platform Management Bus Communications Protocol Specification v1.0	3.5.1	
HPM279	Get Sensor Event Status Command as defined into Intelligent Platform Management Interface Specification Second Generation V2.0 (35.13 Get Sensor Event Status Command) shall be supported	3.5.2.2.2	
HPM278	Event Request Message Event Data Field Contents shall be mandatory for all Event Messages according to Intelligent Platform Management Interface Specification Second Generation v2.0 - Table 29-6, Event Request Message Event Data Field Contentsv1.0	3.5.2.2.2	
HPM5	The Hardware Platform Management shall be based on the IPMI version 1.5.	3.5.1	
HPM6	The HW Management shall be prepared to support and comply with IPMI version 2.0.	3.5.1	
HPM291	The FRU shall have System Firmware Progress (formerly POST Error) sensor type as described into IPMI specification v2.0	3.5.3.1	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
HPM292	The FRU shall have Memory sensor type as described into IPMI specification v2.0	3.5.3.1	
HPM12	It shall be possible to access, in read/write mode, the FRU Information Area Storage for any managed FRU, via standard IPMI Commands, via IPMB-0/IPMB-L Interface and Payload Interface if the interface is applicable.	3.5.2.2.2	
HPM151	The unit shall comply to i2c (smbus) specification	3.5.2.1	
HPM13	Unit IPMI Controller shall supervise/monitor all unit voltages	3.5.3.1	
HPM14	All payload resets shall be monitored by the IPMC.	3.5.3.2	
HPM15	All payload resets shall be monitored by the MMC.	N/A	
HPM16	An IPMI event message shall be generated about any exception in the supervised parameters.	3.5.3.1	
HPM17	Alarm levels for the temperature and voltage sensors shall be set according to document ATCA Blade HW Design Guide: Onboard voltage and thermal sensors.	3.5.2.2.2	
HPM18	Unit IPMI Controller shall be capable of issuing Cold Reset to the Payload according to the PICMG 3.0.	3.5.3.2	
HPM19	Unit IPMI Controller shall be capable of issuing a Diagnostic Interrupt to the Payload according to the PICMG 3.0.	3.5.3.2	
HPM20	Unit IPMI Controller shall be capable of issuing a Graceful shutdown to the Payload according to the PICMG 3.0.	3.5.3.2	"Implemented using a message to the payload over SMI between IPMC and LMP. Reviewed and agreed in 7-Nov FTF meeting."
HPM21	The unit shall support controlled hot swap.	3.5.3.1	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
HPM22	The unit shall continue to operate even if a shelf manager is unavailable.	3.5.2.1	The IPMC will not power down the payload if the Shelf Manager is removed.
HPM23	NSN specific data fields are required for the FRU EEPROM per NSN supplied format definition.	3.5.3.3	
HPM205	FRU data area shall be fully writebale	3.5.3.3	
HPM206	FRU data shall be provided in binary format (.bin)	3.5.3.3	
HPM207	FRU data shall not be lost during eSW upgrade	3.5.3.4	
HPM260	FRU data shall be defined according to "NSN ATCA Blade FRU information Specification" documents.	3.5.3.3	
HPM25	Unit IPMI Controller shall support firmware upgrade according to HPM.1.	3.5.3.4	
HPM26	Unit IPMI Controller shall support firmware Upgrade via local ATCA Payload Interface.	3.5.3.4	
HPM27	Unit IPMI Controller shall support firmware Upgrade via local IPMB-0/IPMB-L interface.	3.5.3.4	
HPM29	Reset of unit IPMI controller shall not affect its controlled/managed unit Payload, state of I/O control signals, affecting it nor with a Payload Reset nor with a Payload Power Cycle.	3.5.3.2	
HPM31	The unit shall support Serial over LAN (SOL) according to IPMI version 2.0.	3.5.3.6	
HPM32	The unit shall support IPMI over LAN (IOL).	3.5.3.6	
HPM33	The unit shall support straight password authentication for the RMCP+ sessions.	3.5.3.6	
HPM34	The unit shall support SOL RMCP+ session over both Base Interfaces.	3.5.3.6	
HPM35	The unit shall support IOL RMCP session over both Base Interfaces.	3.5.3.6	
HPM36	The unit shall support segregation of debug traffic (SOL) via user configurable tagged VLAN.	3.5.3.6	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
HPM37	The unit shall respond independently to a connection request from an IPMI 2.0 compliant counterpart.	3.5.3.6	
HPM38	SOL implementation shall provide IPMI commands for setting and getting all related parameters to/from the target.	Table 16	
HPM39	SOL implementation shall provide same services as the console that is attached locally.	3.5.3.6	
HPM220	The watchdog shall support configurable pretimeout .	3.5.3.7	
HPM221	The hardware unit shall implement a configurable watchdogs to monitor all on-board processors.	3.5.3.7	
HPM222	If a watchdog is not fed during configurable amount of time, the watchdog shall trigger an action. The default action shall be Hard reset	3.5.3.7	
HPM223	It shall be possible to enable and disable watchdog from operating system using generic Linux kernel watchdog driver.	3.5.3.7	
HPM224	It shall be possible to read and write watchdog timeout from operating system using generic Linux kernel watchdog driver.	3.5.3.7	
HPM225	It shall be possible to read and write watchdog pretimeout from operating system using generic Linux kernel watchdog driver.	3.5.3.7	
HPM281	The FRU shall support persistent event log that can be retrieved via HPI	3.5.4	
HPM284	FRU shall be compliant to NSN HPI FUMI - DIMI Implementation guide	3.5.4	
HPM285	The vendor shall provide the FUMI - DIMI test report based on NSN HPI FUMI - DIMI Test Plan	3.5.4	
HPM273	The FRU shall have SAHPI_CAPABILITY_EVENT_LOG	3.5.4	
HPM286	The FRU shall have SAHPI_CAPABILITY_RESET	3.5.4	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
HPM271	Payload processor shall be able to control local watchdog timer via payload interface	Section 16 of [1]	The MCLI “config service watchdog” command is used to enable and disable the local watchdog.
HPM270	HPI Sensor Type SAHPI_SYSTEM_ACPI_POWER_STATE shall be supported	3.5.4	
HPM267	Unit IPMI Controller shall be capable of issuing Get Device GUID command according to the IPMI spec 2.0	3.5.2.2.2	
HPM266	The FRU shall have SAHPI_CAPABILITY_SENSOR	3.5.4	
HPM265	The FRU shall be compliant to SAI-HPI-B.03.02	3.5.4	
HPM264	The FRU shall have SAHPI_CAPABILITY_POWER	3.5.4	
HPM261	The FRU shall be compliant to SAIM-HPI-B.03.02-xTCA	3.5.4	
HPM290	Logical FRUs within physical FRU, for example Octeons sub blocks, shall have the same FRU HPI capabilities as the physical FRU	3.5.4	Individual resources for NP-4 are not supported. There will be one resource for Blade and one for RTM
HPM41	Board vendor shall provide well documented interface for HPI according to SAI-HPI-B.03.02.	3.5.4	
HPM42	All hardware management commands shall be supported via OpenHPI.	3.5.4	
HPM43	The FRU shall include a mechanism that enables the upgrade of all firmwares of the FRU using the mechanisms of HPI FUMI.	3.5.4	
HPM142	The interface for diagnostics shall be HPI DIMI as specified in SAI-HPI-B.03.02. All boards and AMCs containing a management processor shall support a HPI DIMI interface	3.5.4	
HPM143	Boards that support RTMs shall support diagnostic tests for the RTM as well	3.5.4	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
HPM144	The FRU shall support both online diagnostics and offline tests. The offline diagnostic tests shall be marked as DEGRADING in the DIMI resource descriptor record. In the CLI, offline tests shall be clearly marked as such.	3.5.4	
HPM145	The supplier shall provide an OpenHPI plugin for their DIMI interface.	3.5.4	
HPM146	Same diagnostic tests shall be supported via CLI and HPI DIMI.	3.5.4	
HPM147	The test results shall be provided in human-readable plaintext format, both via the interface used to start the test and saved to a file within the FRU.	3.5.4	
HPM148	The FRU shall make use of the logging and error monitoring features provided by the HW and offer the user visibility to the monitoring via CLI commands and/or DIMI.	3.5.4	
HPM149	The DIMI test parameters names and test names shall be as self-explanatory as possible.	3.5.4	
HPM161	The Diagnostic CLI should be usable through a serial-over-LAN connection.	3.5.5	
HPM162	The diagnostic implementation shall allow for the user to explicitly cancel a diagnostic test through any supported interface.	3.5.5	
HPM163	The diagnostic CLI shall not allow for the user to start overlapping diagnostic tests or start the same test concurrently from different interfaces.	3.5.5	
HPM165	A diagnostic test shall not last longer than 5 minutes.	3.5.5	
HPM237	Diagnostics implementation shall support CLI as the system management interface.	3.5.5	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
HPM238	FRUs with or without payload processor shall support IPM Controller Diagnostics Initiator Architecture	Not complaint	
CETH22	All 10/100/1000Base-T/TX/T Ethernet ports shall support autonegotiation as stated in IEEE 802.3 standard. By default shall autonegotiation be enabled on all 10/100/1000Base-T/TX/T interfaces.	Section 3.6.1.1	
CETH23	Autonegotiation with 10/100/1000Base-T/TX/T ports shall be implemented so that if both ends advertise flow control capability shall it be enabled unless explicitly disabled by configuration action.	Refer to [3]	
CETH25	The switch shall support 16000 MAC addresses.	Refer to [3]	
CETH26	The total buffer memory in the switch shall be at least 2 MBytes.	Refer to [3]	
CETH27	Bit error rate of any internal link in the network element shall be better than 10exp-12.	Refer to [3]	
CETH104	The switch component shall support store and forward as well as cut through operating modes.	Refer to [10]	Broadcom 56844 supports both store and forward and cut through modes.
CETH838	The switch shall support 200 MAC addresses.	Refer to [3]	
CETH837	The total buffer memory in the switch shall be at least 128kBytes.	Refer to [3]	
CETH108	The switch shall implement at least the following diagnostics: Verifying the validity of contents of all flash and prom memory Verifying the functionality of the Unit Computer Verifying the functionality of the switch matrix chip-set Verifying the functionality of all R/W-memory Verifying the functionality of all LAN-ports (to be performed with local loop-back)	Section 3.6.1.2	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
CETH109	The Ethernet frame loss rate shall be less than 10exp-9.	Refer to [3]	
CETH122	The switch shall work as a L2/L3 switch from the beginning, including VLAN support in environments where there exist both IPv4 and IPv6 traffic.	Section 3.6.3.1	L3 routing is not enabled in the ANPI1-A per discussion with NSN on the use case for the blade.
CETH224	The hardware shall support Energy Efficient Ethernet features introduced in the IEEE802.3az.	Section 3.6.1.3	
CETH29	The switch shall implement a CLI-based configuration interface, which can be used to configure any of the configurable parameters.	Refer to [1]	Summary of some CLI commands relevant to particular functionality are given in this document but complete details are given in the referenced document.
CETH30	The switch shall have a utility (CLI command), which can be used to upload and download a file, which contains all configuration data. The file shall be in a human readable format, i.e. ASCII-text.	Section 3.6.2.1	
CETH32	Management of configuration data shall be implemented so that when changes are done, either through CLI or through SNMP, the effects of new commands are added into the existing configuration.	Section 2.1.3.1, Section 2.1.3.2	
CETH35	The ethernet management SW shall support Virtual LAN's.	Section 3.6.3.1	
CETH251	The FRU shall support restoring the factory default configuration by a single command.	Section 3.6.3.2	
CETH238	The Ethernet management SW shall support Q in Q.	Section 3.6.3.3	
CETH36	When PVID is set by a command, the PVID shall be taken into use immediately without a delay.	Section 3.6.3.1	
CETH37	It shall be possible to change any of the VLAN-parameters without resetting the ethernet management computer.	Section 3.6.3.1	
CETH40	Tag based IEEE 802.1p frame priority shall be supported.	Section 3.6.5.1	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
CETH43	The ethernet management sw shall support setting of the default VLAN-ID and priority per port (to be used when receiving untagged frames).	Section 3.6.3.1	
CETH44	It shall be possible to configure which LAN-ports, VLAN-ID and priority are to be used when receiving and sending management messages.	Section 3.6.3.4	
CETH45	The switch shall support port mirroring.	Section 3.6.3.6	
CETH229	The switch shall support VLAN mirroring.	Section 3.6.3.7	
CETH46	The switch matrix shall be capable of handling 99% of the full wire-speed traffic with 0% drop rate in a situation, where the offered traffic is broadcast from a single port.	Refer to [4] and also section 3.6.3.8	
CETH47	The switch matrix shall be capable of handling 99% of the full wire-speed traffic with 0% drop rate in a situation, where the offered traffic is multicast from a single port and is targeted to all other ports.	Refer to [4] and also section 3.6.3.8	
CETH48	Jumbo frames up to 9216 bytes (=9*1024) shall be supported.	Section 3.6.3.10	
CETH49	The maximum size of the jumbo frames shall be configurable.	Section 3.6.3.10	
CETH52	The switch shall support at least 32 group addresses per port.	Section 3.6.3.9	
CETH53	Link Aggregation (LAG) shall be supported in the ethernet switch management SW.	Section 3.6.3.11	
CETH90	Link aggregation load sharing method shall support: - source mac address - destination mac address - source & destination mac address	Section 3.6.3.11	
CETH91	Link aggregation load sharing method shall support: - source IP address - destination IP address - source & destination IP address	Section 3.6.3.11	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
CETH64	The switch shall support flow control according to IEEE 802.3x standard.	Section 3.6.3.12	
CETH65	The buffering shall be implemented in a manner, which in co-operation with 802.3x flow control guarantees that no frames are lost due to buffer overflow.	3.6.3.13	Taken from requirements spreadsheet "Requires further review of traffic expectations with NSN… e.g., what are the max number of input ports from which traffic could egress on an output port?" Need to get further info to design for this.
CETH123	After the configuration has been updated, either by management interface or by a configuration file, this configuration (stored in non-volatile memory) shall replace the factory defaults.	Section 3.6.3.15	
CETH126	It shall be possible to reset the whole unit i.e. switch matrix and the unit computer through a CLI command.	Section 3.6.3.16	
CETH129	After new SW image is transferred into switch unit computer shall its consistency be checked by using CRC.	Section 3.1.3	
CETH130	The switch shall implement a Configuration-history feature, which records onto a non-volatile memory all commands given with CLI and affecting configuration.	Section 3.6.3.17	
CETH131	It shall be possible to load and execute a new or updated configuration file without resetting the switch.	Section 3.6.3.14	
CETH132	All management connections to the switch (through SNMP, telnet/SSH, RS232 serial port or 10/100Base-T/TX) shall be protected by a password.	Refer to Chapter 7 of [2] and section 3.6.4.2	The standard SNMP password mechanism is used for the ANPI1-A
CETH135	It shall be possible to use passwords, which are up to 16 characters long.	Section 3.6.3.18	
CETH138	There shall be a possibility to erase all the flash (or other non-volatile memory) contents thoroughly so that also the passwords will be erased.	Section 3.1.3	Note last paragraph of section which states that flash part is erased as part of programming.

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
CETH140	When the switch detects that a previously detected hw or sw failure has been corrected (or disappears), shall the switch write a log/SEL entry and inform the supervision and maintenance SW by an SNMP trap.	Section 3.6.4.3	List of supported traps is given in referenced section. Radisys will supply only existing SNMP functionality.
CETH141	It shall be possible to update software even in the situation where there's no functional software on the flash.	Section 3.3.1.3	
CETH142	It shall be possible to suppress unicast/multicast/broadcast Ethernet storm coming from a network interface.	Section 3.6.4.4	
CETH68	The ethernet switch management SW shall support Dynamic Host Configuration Protocol (DHCP).	Refer to chapter 6 of [2]	The standard Linux DHCP support is included on the ANPI1-A
CETH94	If the DHCP ACK message includes boot file name option and next server, shall the embedded software fetch file pointed by that path and execute it.	Section 3.6.4.1	
CETH70	The management of the switch shall be based on open protocols, namely SNMP. It shall be possible to configure any of the configurable parameters by using SNMP. In other words, it must be possible to do a GET, SET and GET BULK operation to any parameter, which can be modified through the CLI.	Section 3.6.4.3	
CETH73	It shall be possible to activate all supported SNMP traps by a single CLI command.	See description of snmp-trap command in section 3.6.4.3	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
CETH74	The switch shall implement at least the following SNMP traps: - Cold start (to be sent after power-up, watchdog reset) - Active (i.e. forwarding) link has changed physical state from link-up to link down or vice versa - Redundant (i.e. blocked or discarding) link has changed physical state from link-up to link down or vice versa - Unit Computer overload. Threshold must be configurable (default 80%), integration time shall be in the order of 30..60 seconds) - High rate of input-errors(CRC, alignment and bad symbols) on active or redundant link. Threshold must be configurable (default 1%), integration time shall be in the order of 1..60 seconds - High rate of runt or oversize frames. Threshold must be configurable (default 1%), integration time shall be in the order of 1..60 seconds - Loading of configuration file failed or the loaded file has been found to be corrupted -Loading of operating system image file has been found to be corrupted (MD5 error) - Security violation (failed login, unauthorized SNMP query) - Notification (trap) when self-test has been completed. Each sent trap shall specify which port exactly the trap relates to.	Section 3.6.4.3	
CETH75	All traps shall be stored to a Trap log.	Section 3.6.4.3	
CETH76	The switch shall support TFTP client.	Section 3.6.4.5	
CETH150	The ethernet switch management SW shall support Address Resolution Protocol (ARP).	Refer to chapter 71 of [1]	Standard Linux ARP support is included.

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
CETH153	The ethernet switch management SW shall support Internet Control Message Protocol (ICMP).	Refer to chapter 30 of [1]	Standard Linux ICMP support is included.
CETH166	The ethernet switch management SW shall support telnet.	Section 3.6.4.2	Standard Linux telnet support is included.
CETH168	The ethernet switch management SW shall support Secure Shell (SSH).	Section 3.6.4.2	Standard Linux SSH support is included.
CETH169	The ethernet switch management SW shall support SSH file transfer protocol (SFTP).	Section 3.6.4.2	Standard Linux SFTP support is included.
CETH170	The ethernet switch management SW shall support Simple Network Management Protocol (SNMP).	Section 3.6.4.3	
CETH171	It shall be possible to configure the switch so that instead of SNMP traps are inform requests sent.	Section 3.6.4.3	
CETH172	It shall be possible to configure the IP-address for the SNMP manager (trap host) trough SNMP. Result shall be identical as compared to the situation where the trap host address is set through CLI.	Section 3.6.4.3	
CETH173	It shall be possible to query the latest self-test result by SNMP. The result shall specify any executed sub-test and the corresponding result.	Section 3.6.4.3	
CETH174	The switch shall support standard and extended Layer 2 and Layer 3 Access Control Lists (ACL). At least the following selectors shall be supported: - source IP address / subnet / network - destination IP address / subnet /network - protocol type (udp, tcp, sctp, icmp, icmp6, igmp, esp, ah, ..) - port (udp, tcp, sctp) - ICMP type, ICMP type/code, ICMP message - Source/Destination MAC address	Section 3.6.4.7	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
CETH216	The switch shall implement at least following MIBs: -MIB-II, -BRIDGE MIB (RFC-1493) -Possible vendor specific private MIB -Target MIB -RMON MIB (Group 1,2,3,9,x)	Section 3.6.4.3	
CETH232	The switch matrix of the switch shall support eight priority queues per port. The priority mechanism shall be as described in the IEEE 802.1 standards.	Section 3.6.5.1	
CETH79	The switch matrix of the switch shall support eight priority queues per port. The priority mechanism shall be as described in the IEEE 802.1 standards.	Section 3.6.5.1	
CETH80	The switch matrix shall support strict priority queuing (as described in the IEEE 802.1 standards).	Section 3.6.5.2	
CETH81	The switch matrix shall support WRR (Weighted Round Robin).	Section 3.6.5.2	
CETH86	The switch matrix shall support DRR (Deficit Round Robin).	Section 3.6.5.2	
CETH88	The switch shall support rate-shaping in its egress ports.	Section 3.6.5.3	
CETH82	Buffer memory allocation shall be configurable per transmit queue.	Section 3.6.5.4	
CETH200	The QoS queue used shall be determined based on the configured user priority bits in the VLAN tag of the incoming Ethernet frame (as described in IEEE 802.1p/q/d) +DSCP/ToS.	Section 3.6.5.1	
CETH252	The switch management SW shall support packet marking by DSCP, VLAN prority, MPLS EXP bit as well as IP precedence fields.	Section 3.6.5.5	
CETH195	The switch shall store the factory default configuration in a non-volatile memory, which is used when the board is powered up for the first time.	Section 3.6.6.1	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
CETH823	The factory default configuration for the switch shall be as follows: All ports are enabled Autonegotiation is enabled on all ports supporting autonegotiation All ports belongs to default VLAN1 (PVID=1). Link aggregation is disabled. LACP is disabled GVRP is disabled LLDP is disabled RSTP is disabled, all backplane interfaces configured as edge ports. Management traffic is allowed The factory default configuration for the unit computer shall be as follows: DHCP is enabled CLI time-out timer is disabled NTP client enabled	Section 3.6.6.2	
EXIF223	Supplier shall state for which pluggable transceivers the external ports are going to be tested by supplier.	Refer to [4]	
EXIF224	Unit shall support pluggable transceiver management; Nonvolatile memory information of transceivers must be available for NSN API. Also TX-fault, LOS, and other supported alarms shall be visible to NSN API	Section 3.6.6.3	
EXIF217	Unit shall support optionally 2 x 40GBASE-LR4 interface according IEEE802.3ba and additional 2 x 1000Base interfaces with SFP sockets.	Refer to [3]	
EXIF221	Unit shall support 12 x 10GBase interfaces with SFP+ connector	Refer to [3]	
EXIF220	External payload ports shall be located on RTM	Refer to [3]	
EXIF19	The serial interfaces shall be implemented with RJ-45 connectors.	Refer to [3]	
EXIF211	The pinout of the serial interface RJ-45 connectors shall be according to Cisco pinout.	Refer to [3]	
EXIF215	There shall be at least one RS-232 serial port connection to the Unit Computer processor.	Refer to [3]	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
EXIF180	Serial port default speed shall be 38.4 kbps.	Section 3.7.3.1	
OS91	Unit shall be equipped with WindRiver PNE Linux Operating System version 4 (WR4) ported on the specific CPU and environment	Section 3.8.1.1	
OS81	The vendor shall provide binary Board Support Package.	3.3.2.1	The board support package is supplied in an update bundle and can be applied to the blade through FUMI.
OS12	The startup script shall be stored in Flash together with other configuration parameters.	3.3.2	The startup script is stored in a subdirectory of /etc. The /etc directory, which includes all configuration files generated by the CLI, is bind mounted for persistence. The configuration file to the JFFS because of the bind mount settings.
OS98	It should be possible to run all configuration and monitoring utilities from a shell script.	3.3.2	Configuration and monitoring is done through the CLI. It is possible to pass commands to the CLI from a shell script through a utility provided on the blade.
OS99	The unit shall synchronize its clock using NTP	3.3.2	The carrier supports the Linux standard Network Time Protocol (NTP) server, which allows it to provide time of day services to other installed modules or systems. The NTP server is disabled by default. This can be enabled through the CLI.

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
OS100	The Unit Computer shall be accessible via ssh	3.3.2	SSH can be enabled through the services menu of the CLI. A password must be added to the user on the blade to allow SSH connections.
OS101	The Unit Computer shall be able to redirect its syslog to a remote host using syslog	3.3.2	The module supports the standard Linux syslog-ng service. This service collects and controls the output of messages from the software components on the module. The output of messages can be enabled or disabled, and can be directed to a remote server or to a local file. By default, the module-wide syslog-ng service is disabled. Logging for all software components is enabled by default, but messages from components are ignored unless the module-wide service is enabled. The module-wide syslog-ng configuration also determines the destination for the messages from components.
OS117	A BSP for WindRiver Linux PNE Release 4.3 shall be provided	3.3.2	
OS19	The BSP shall fully support the HW with all the HW configuration options.		
OS21	The BSP shall support access to and configuration of all system devices that support such access		
OS22	Each driver shall have an initialization routine inserting it into the OS device list		

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
OS82	The vendor shall provide sources Board Support Package with tools for building, if it's not open source	3.3.2	We will distribute most of the source code. The full list of exceptions to source code is listed in section 3.3.2.
OS83	Vendor shall provide the Board Support Package so it is physically separated from: 1) the development tools tree 2) the working directory 3) the building directory	N/A	
OS25	Ethernet ports shall be configurable through standard Linux kernel API	3.3.2	
OS26	Link down notification shall be given within 50 ms after Ethernet device has noticed link status change	Section 3.8.2.1	
OS29	The BSP shall use standard interfaces and tools provided by baseline operating system and exceptions and their rationale must be documented by the vendor.	3.3.2	
OS85	Drivers included in the Board Support Packages shall not taint the kernel	N/A	The BDE driver used by switchdrvr is not GPL, and will taint the kernel.
OS32	The board support package shall be released under GPL.	N/A	Because we include Fastpath in the image, we cannot release it under GPL.
OS33	The board support package shall contain tools to programmatically identify (resolve version identifiers) and update all firmware elements used on the board.		The SW version of each eSW component wil be retrievable from the Linux OS generated from the BSP.
OS34	The BSP components that are stored in the redundant flash shall be upgradeable in-field without affecting the normal operation	3.3.2.2	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
OS88	BSP shall provide general management for accessing the FRU both via SNMP and via CLI.	N/A	Most of the tools are command line tools that meet the CLI requirements. Not all BSP commands are available through SNMP. **Reviewed and agreed in 7-Nov FTF meeting.**
OS36	The board support package shall include ChangeLog file where release dates and version numbers are explicitly mentioned. Version number shall also be mentioned in the package's filename (for example <board>-wr-bsp-1.3.5.tar.bz2)	N/A	This requirement is not part of this spec. Radisys build and release process dictates the existence of change (build) logs with corresponding version numbers.
OS37	The board support package shall be delivered with documentation that describes any non-trivial design choices made.	N/A	This requirement is not part of this eSW spec. Radisys will provide both Technical Pub and Engineering documentation to accompany non-trivial design or operation choices.
OS38	The board support package shall be delivered with documentation that describes the steps that are needed to configure, build and take the operating system into use.	N/A	This requirement is not part of this spec. Radisys will provide requested documentation.
OS39	The board support package shall be delivered with a listing that explicitly documents all licenses used.	N/A	Radisys will deliver a separate document indicating all the SW packages within delivered BSP with corresponding licenses.

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
OS40	The board support package shall be delivered with a test suite.	N/A	As agreed before, Radisys will deliver at least “the list of test cases the vendor executes on the BSP (test plan) and the relevant test report”.
OS41	The vendor shall provide the list of test cases that cover the BSP functional areas	Refer to [4]	
OS42	The board support package shall be delivered with a list of test cases (test plan) and a description of testing (test report).	Refer to [4]	
OS43	Any software components (kernel, kernel modules, userspace components) provided by the board support package shall contain an explicit version number.	N/A	Same response as for OS40
OS44	Linux kernel and all loadable kernel modules shall have static string "BSP_VERSION_INFO=x.x.x" which has BSP version number.	3.3.2	Most of the packages included in our BSP are not even compiled by us, and are taken as binaries straight from WindRiver. We have an alternative tracking method, which I have included in the documentation below.
OS45	Linux kernel should have "BSP_VERSION_INFO" to be listed in "/proc/".	3.3.2	Same comment as for OS44
OS87	Linux user space tools included in BSP should have BSP_VERSION_INFO=x.x.x embedded as static strings.	3.3.2	Same comment as for OS44
OS49	The board support package shall support the boot mechanism used by NSN products (currently u-boot on Octeon/PPC, BIOS on x86).	Table 12 3.3.1	
OS50	The board support package shall support booting from all supported devices as well as network boot.	Table 12 3.3.1	

Table 54. Requirements Matrix

Cust. #	Requirement	Section Addressed	Comments
OS54	The board support package shall contain mechanism for collection of data to be used in postmortem-analysis. Preferred method is kdump-mechanism or other mechanism from bendor which is agreed with NSN.	3.3.2	
OS55	The board support package shall be debuggable with kgdb (or equivalent kernel debugging tool) over a serial and/or ethernet connection.	3.3.2	
OS56	The board support package shall propagate (error codes) and report (log entries) driver-level errors via standard interfaces.	3.3.2	
OS57	All interfaces shall be visible in Linux kernel as standard network devices.	3.3.2	
OS58	There shall be only one device per port.	3.3.2	
OS59	Device specific register settings (e.g. in PHY and MAC) shall be tunable by NSN The APIs etc required to implement the above must be documented		Radisys will provide API for specific register settings and accompany the implementation with an adequate engineering level documentation.
OS60	The board support package shall contain tools to access status data provided by board self-test / offline diagnostics.	3.5.5	
OS104	The BSP shall include FUMI agent and corresponding HPI plug-in	3.5.4	