Document Identifier: DSP0222

Date: 2020-08-04

Version: 1.2.0b

Network Controller Sideband Interface (NC-SI) Specification

Information for Work-in-Progress version:

IMPORTANT: This document is not a standard. It does not necessarily reflect the views of the DMTF or its members. Because this document is a Work in Progress, this document may still change, perhaps profoundly and without notice. This document is available for public review and comment until superseded.

Provide any comments through the DMTF Feedback Portal:
http://www.dmtf.org/standards/feedback

Supersedes: 1.1.0

Document Class: Normative

Document Status: Work in Progress

Document Language: en-US

CONTENTS

Figures

Tables

Foreword

The *Network Controller Sideband Interface (NC-SI) Specification* (DSP0222) was prepared by the PMCI Working Group.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Acknowledgments

The DMTF acknowledges the following individuals for their contributions to this document:

Editors:

- Hemal Shah – Broadcom Inc
- Bob Stevens – Dell Inc.

Contributors:

- Patrick Caporale - Lenovo
- Phil Chidester – Dell
- Yuval Itkin – Mellanox Technologies
- Ira Kalman – Intel Corporation
- Patrick Kutch – Intel Corporation
- Eliel Louzoun – Intel Corporation
- Patrick Schoeller – Hewlett-Packard Company
- Tom Slaight – Intel Corporation

Introduction

In out-of-band management environments, the interface between the out-of-band Management Controller and the Network Controller is critical. This interface is responsible for supporting communication between the Management Controller and external management applications. Currently there are multiple such proprietary interfaces in the industry, leading to inconsistencies in implementation of out-of-band management.

The goal of this specification is to define an interoperable sideband communication interface standard to enable the exchange of management data between the Management Controller and Network Controller. The Sideband Interface is intended to provide network access for the Management Controller, and the Management Controller is expected to perform all the required network functions.

This specification defines the protocol and commands necessary for the operation of the sideband communication interface. This specification also defines physical and electrical characteristics of a sideband binding interface that is a variant of RMII targeted specifically for sideband communication traffic.

The specification is primarily intended for architects and engineers involved in the development of network interface components and Management Controllers that will be used in providing out-of-band management.

Network Controller Sideband Interface (NC-SI) Specification

1 Scope

This specification defines the functionality and behavior of the Sideband Interface responsible for connecting the Network Controller (including Ethernet, Fibre Channel and InfiniBand controllers) to the Management Controller. It also outlines the behavioral model of the network traffic destined for the Management Controller from the Network Controller.

This specification defines the following two aspects of the Network Controller Sideband Interface (NC-SI):

- behavior of the interface, which include its operational states as well as the states of the associated components
- the payloads and commands of the communication protocol supported over the interface

The scope of this specification is limited to addressing only a single Management Controller communicating with one or more Network Controllers.

This specification also defines the following aspects of a 3.3V RMII-Based Transport (RBT) based physical medium:

- transport binding for NC-SI over RBT
- electrical and timing requirements for the RBT
- an optional hardware arbitration mechanism for RBT

Only the topics that may affect the behavior of the Network Controller or Management Controller, as it pertains to the Sideband Interface operations, are discussed in this specification.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

DMTF DSP0261, *NC-SI over MCTP Binding Specification 1.2*
http://www.dmtf.org/standards/published_documents/DSP0261_1.0.pdf
http://www.dmtf.org/standards/published_documents/DSP0261_1.2.pdf

https://www.dmtf.org/sites/default/files/standards/documents/DSP0261_1.2.2.pdf

DMTF DSP0240, Platform Level Data Model (PLDM) Base Specification 1.0.0

https://www.dmtf.org/sites/default/files/standards/documents/DSP0240_1.0.0.pdf

IEEE 802.3, *802.3™ IEEE Standard for Information technology— Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications,* December 2012, http://www.ieee.org/portal/site

IEEE 802.1Q, *IEEE 802.1Q-2005 IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks*, http://www.ieee.org/portal/site. This standard defines the operation of

562 Virtual LAN (VLAN) Bridges that permit the definition, operation and administration of Virtual LAN
563 topologies within a Bridged LAN infrastructure.

564 IETF RFC2131, *Dynamic Host Configuration Protocol* (DHCP), March 1997,
565 http://www.ietf.org/rfc/rfc2131.txt

566 IETF RFC2373, *IP Version 6 Addressing Architecture*, July 1998, http://www.ietf.org/rfc/rfc2373.txt

567 IETF RFC2461, *Neighbor Discovery for IP Version 6 (IPv6)*, December 1998,
568 http://www.ietf.org/rfc/rfc2461.txt

569 IETF RFC2464, *Transmission of IPv6 Packets over Ethernet Networks*, December 1998,
570 http://www.ietf.org/rfc/rfc2464.txt

571 IETF RFC3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, July 2003,
572 http://www.ietf.org/rfc/rfc3315.txt

573 IETF, RFC4122, *A Universally Unique Identifier (UUID) URN Namespace*, July 2005
574 http://datatracker.ietf.org/doc/rfc4122/

575 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards,*
576 http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456&objAction=browse&sort=subtype

577 Reduced Media Independent Interface (RMII) Consortium, *RMII Specification*, revision 1.2, March 20,
578 1998, http://ebook.pldworld.com/_eBook/-Telecommunications,Networks-/TCPIP/RMII/rmii_rev12.pdf

579 InfiniBand™ Architecture Specification

580 https://www.infinibandta.org/ibta-specification/

581 Fibre Channel

582 ?

583 3 Terms and definitions

584 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms
585 are defined in this clause.

586 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"),
587 "may", "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
588 in ISO/IEC Directives, Part 2, Clause 7. The terms in parentheses are alternatives for the preceding term,
589 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
590 ISO/IEC Directives, Part 2, Clause 7 specifies additional alternatives. Occurrences of such additional
591 alternatives shall be interpreted in their normal English meaning.

592 The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as
593 described in ISO/IEC Directives, Part 2, Clause 6.

594 The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC
595 Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do
596 not contain normative content. Notes and examples are always informative elements.

597 The terms defined in DSP0004, DSP0223, and DSP1001 apply to this document. The following additional
598 terms are used in this document.

3.1 Requirement term definitions

This clause defines key phrases and words that denote requirement levels in this specification.

3.1.1
conditional

indicates that an item is required under specified conditions

3.1.2
deprecated

indicates that an element or profile behavior has been outdated by newer constructs

3.1.3
mandatory

indicates that an item is required under all conditions

3.1.4
may

indicates that an item is truly optional

NOTE An implementation that does not include a particular option shall be prepared to interoperate with another implementation that does include the option, although perhaps with reduced functionality. An implementation that does include a particular option shall be prepared to interoperate with another implementation that does not include the option (except for the feature that the option provides).

3.1.5
may not

indicates flexibility of choice with no implied preference

3.1.6
not recommended

indicates that valid reasons may exist in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and carefully weighed before implementing any behavior described with this label

3.1.7
obsolete

indicates that an item was defined in prior specifications but has been removed from this specification

3.1.8
optional

indicates that an item is not mandatory, conditional, or prohibited

3.1.9
recommended

indicates that valid reasons may exist in particular circumstances to ignore a particular item, but the full implications should be understood and carefully weighed before choosing a different course

3.1.10
required

indicates that the item is an absolute requirement of the specification

3.1.11
shall
indicates that the item is an absolute requirement of the specification

3.1.12
shall not
indicates that the item is an absolute prohibition of the specification

3.1.13
should
indicates that valid reasons may exist in particular circumstances to ignore a particular item, but the full implications should be understood and carefully weighed before choosing a different course

3.1.14
should not
indicates that valid reasons may exist in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and carefully weighed before implementing any behavior described with this label

3.2 NC-SI term definitions

For the purposes of this document, the following terms and definitions apply.

3.2.1
Frame
a data packet of fixed or variable length that has been encoded for digital transmission over a node-to-node link

Frame is used in references to IEEE 802.3 Frames. *Packet* is used in all other references.

3.2.2
Packet
a formatted block of information carried by a computer network

Frame is used in references to IEEE 802.3 Frames. *Packet* is used in all other references.

3.2.3
External Network Interface
the interface of the Network Controller that provides connectivity to the external network infrastructure; also known as *port*

3.2.4
Internal Host Interface
the interface of the Network Controller that provides connectivity to the host operating system running on the platform

3.2.5
Management Controller
an intelligent entity composed of hardware/firmware/software that resides within a platform and is responsible for some or all of the management functions associated with the platform; also known as BMC and Service Processor

3.2.6
Network Controller

the component within a system that is responsible for providing connectivity to an external Ethernet , Fibre Channel or InfiniBand network

3.2.7
Remote Media

a manageability feature that enables remote media devices to appear as if they are attached locally to the host

3.2.8
Network Controller Sideband Interface

NC-SI

The RBT interface of the Network Controller that provides network connectivity to a Management Controller; also shown as *Sideband Interface, RBT* or *NC-SI* as appropriate in the context

3.2.9
Integrated Controller

a Network Controller device that supports two or more channels for the NC-SI that share a common NC-SI physical interface (for example, a Network Controller that has two or more physical network ports and a single NC-SI bus connection)

3.2.10
Multi-drop

refers to the situation in which multiple physical communication devices share an electrically common bus and a single device acts as the master of the bus and communicates with multiple “slave” or “target” devices

Related to NC-SI, a Management Controller serves the role of the master, and the Network Controllers are the target devices.

3.2.11
Point-to-Point

refers to the situation in which only a single Management Controller and single Network Controller package are used on the bus in a master/slave relationship, where the Management Controller is the master

3.2.12
Channel

refers to the logical representation of a physical network port in a Network Controller that supports Control traffic and may support Pass-through traffic

A Network Controller that has multiple network interface ports can support an equivalent number of NC-SI channels.

3.2.13
Package

one or more NC-SI channels in a Network Controller that share a common set of electrical buffers and common electrical buffer controls for the NC-SI bus

Typically, a single, logical NC-SI package exists for a single physical Network Controller package (chip or module). However, this specification allows a single physical chip or module to hold multiple NC-SI logical packages.

720 **3.2.14**
721 **Control traffic**
722 **Control Packets**
723 command, response, and asynchronous event notification packets transmitted between the Management
724 Controller and Network Controllers for the purpose of managing the NC and NC-SI

725 **3.2.15**
726 **Command**
727 Control Packet sent by the Management Controller to the Network Controller to request the Network
728 Controller to perform an action, and/or return data

729 **3.2.16**
730 **Response**
731 Control Packet sent by the Network Controller to the Management Controller as a positive
732 acknowledgement of a command received from the Management Controller, and to provide the execution
733 outcome of the command, as well as to return any required data

734 **3.2.17**
735 **Asynchronous Event Notification**
736 Control Packet sent by the Network Controller to the Management Controller as an explicit notification of
737 the occurrence of an event of interest to the Management Controller

738 **3.2.18**
739 **Pass-through traffic**
740 **Pass-through packets**
741 network packets passed between the external network and the Management Controller through the
742 Network Controller

743 **3.2.19**
744 **RBT**
745 **RMII Based Transport**

746 Electrical and timing specification for a 3.3V physical medium that is derived from RMII.

747 3.3 Numbers and number bases

748 Hexadecimal numbers are written with a “0x” prefix (for example, `0xFFF` and `0x80`). Binary numbers are
749 written with a lowercase “b” suffix (for example, `1001b` and `10b`). Hexadecimal and binary numbers are
750 formatted in the `Courier New` font.

751 3.4 Reserved fields

752 Unless otherwise specified, reserved fields are reserved for future use and should be written as zeros and
753 ignored when read.

754 4 Acronyms and abbreviations

755 The following symbols and abbreviations are used in this document.

756 **4.1**
757 **AC**
758 alternating current

4.2
AEN
Asynchronous Event Notification

4.3
BMC
Baseboard Management Controller (often used interchangeably with MC)

4.4
CRC
cyclic redundancy check

4.5
CRS_DV
a physical NC-SI signal used to indicate Carrier Sense/Received Data Valid

4.6
DC
direct current

4.7
DHCP
Dynamic Host Configuration Protocol

4.8
FCS
Frame Check Sequence

4.9
MC
Management Controller

4.10
NC
Network Controller

4.11
NC-SI
Network Controller Sideband Interface

4.12
NC-SI RX
the direction of traffic on RBT from the Network Controller to the Management Controller

4.13
NC-SI TX
the direction of traffic RBT to the Network Controller from the Management Controller

4.14
RMII
Reduced Media Independent Interface

798 **4.15**
799 **RX**
800 Receive

801 **4.16**
802 **RXD**
803 physical NC-SI signals used to transmit data from the Network Controller to the Management Controller

804 **4.17**
805 **RX_ER**
806 a physical NC-SI signal used to indicate a Receive Error

807 **4.18**
808 **SerDes**
809 serializer/deserializer; an integrated circuit (IC or chip) transceiver that converts parallel data to serial data
810 and vice-versa. This is used to support interfaces such as 1000Base-X and others.

811 **4.19**
812 **TX**
813 Transmit

814 **4.20**
815 **TXD**
816 physical NC-SI signals used to transmit data from the Management Controller to the Network Controller

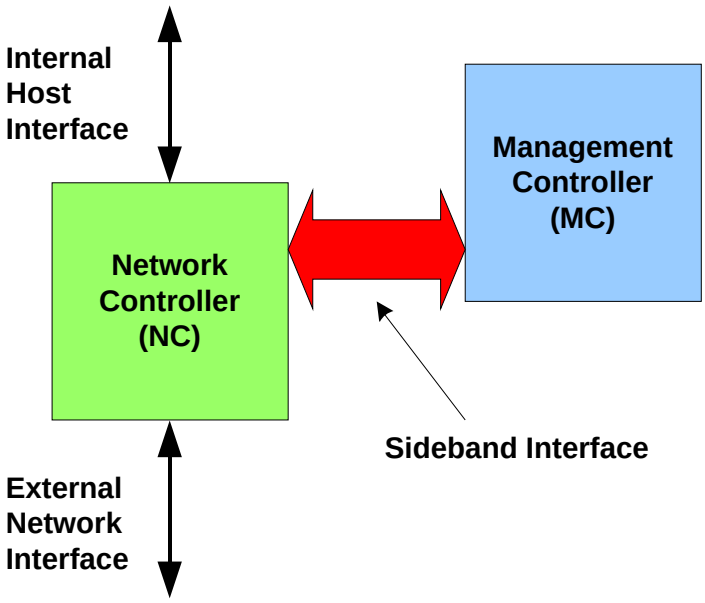
817 **4.21**
818 **VLAN**
819 Virtual LAN

820 5 NC-SI overview

821 With the increasing emphasis on out-of-band manageability and functionality, such as Remote Media
822 (R-Media) and Remote Keyboard-Video-Mouse (R-KVM), the need for defining an industry standard
823 Network Controller Sideband Interface (NC-SI) has become clear. This specification enables a common
824 interface definition between different Management Controller and Network Controller vendors. This
825 specification addresses not only the electrical and protocol specifications, but also the system-level
826 behaviors for the Network Controller and the Management Controller related to the NC-SI.

827 The NC-SI is defined as the interface (protocol, messages, and medium) between a Management
828 Controller and one or multiple Network Controllers. This interface, referred to as a Sideband Interface in
829 Figure 1, is responsible for providing external network connectivity for the Management Controller while
830 also allowing the external network interface to be shared with traffic to and from the host.

831 The specification of how the NC-SI protocol and messages are implemented over a particular physical
832 medium is referred to as a transport binding. This document, DSP0222, includes the definition of the
833 transport binding, electrical, framing, and timing specifications for a physical interface called RBT
834 (RMII-based Transport). Electrically, RBT, as described in clause 10, is similar to the Reduced Media
835 Independent Interface™ (RMII) – hence the name. Transport bindings for NC-SI over other media and
836 transport protocols are defined through external transport binding specifications, such as DSP0261, the
837 *NC-SI over MCTP Transport Binding Specification*. That specification defines the Get Supported Media
838 command (0x54) which is used to discover if the NC supports operation over multiple media. This
839 command may be issued on any NC-SI transport including RBT.



840

841 **Figure 1 – NC-SI functional block diagram**

842 NC-SI traffic flow is illustrated in Figure 2. Two classes of packet data can be delivered over the Sideband
843 Interface:

844 • "Pass-through" packets that are transferred between the Management Controller and the
845 external network

846 • "Control" packets that are transferred between the Management Controller and Network
847 Controllers for control or configuration functionality

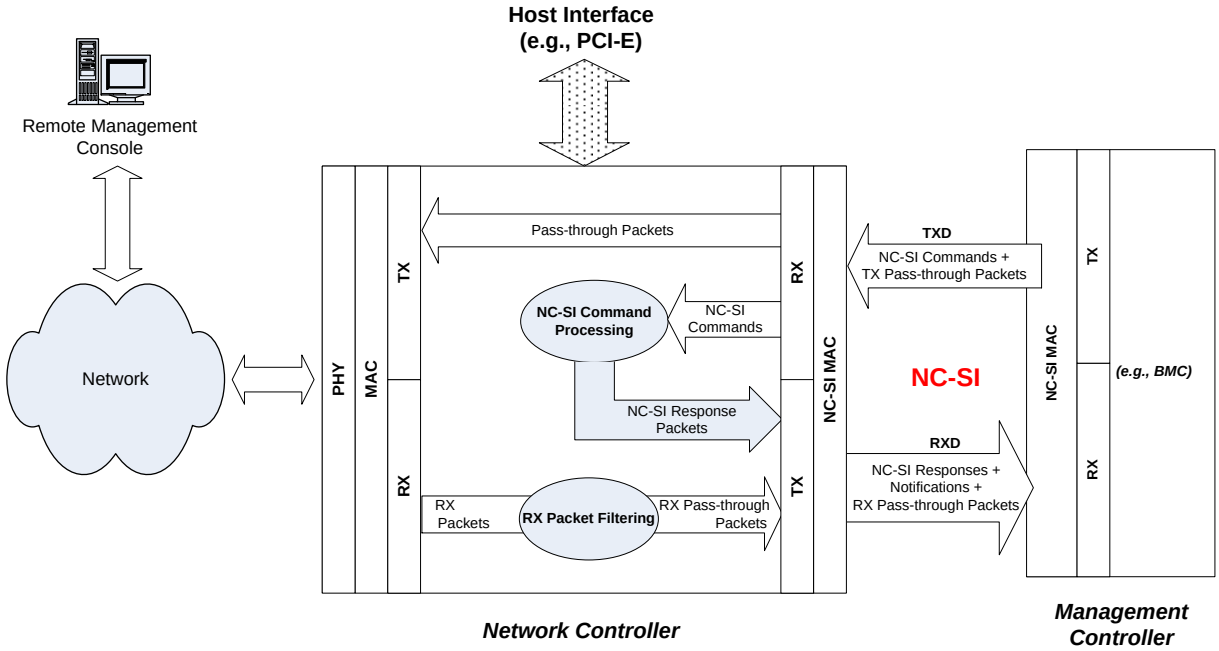


Figure 2 – NC-SI RBT traffic flow diagram

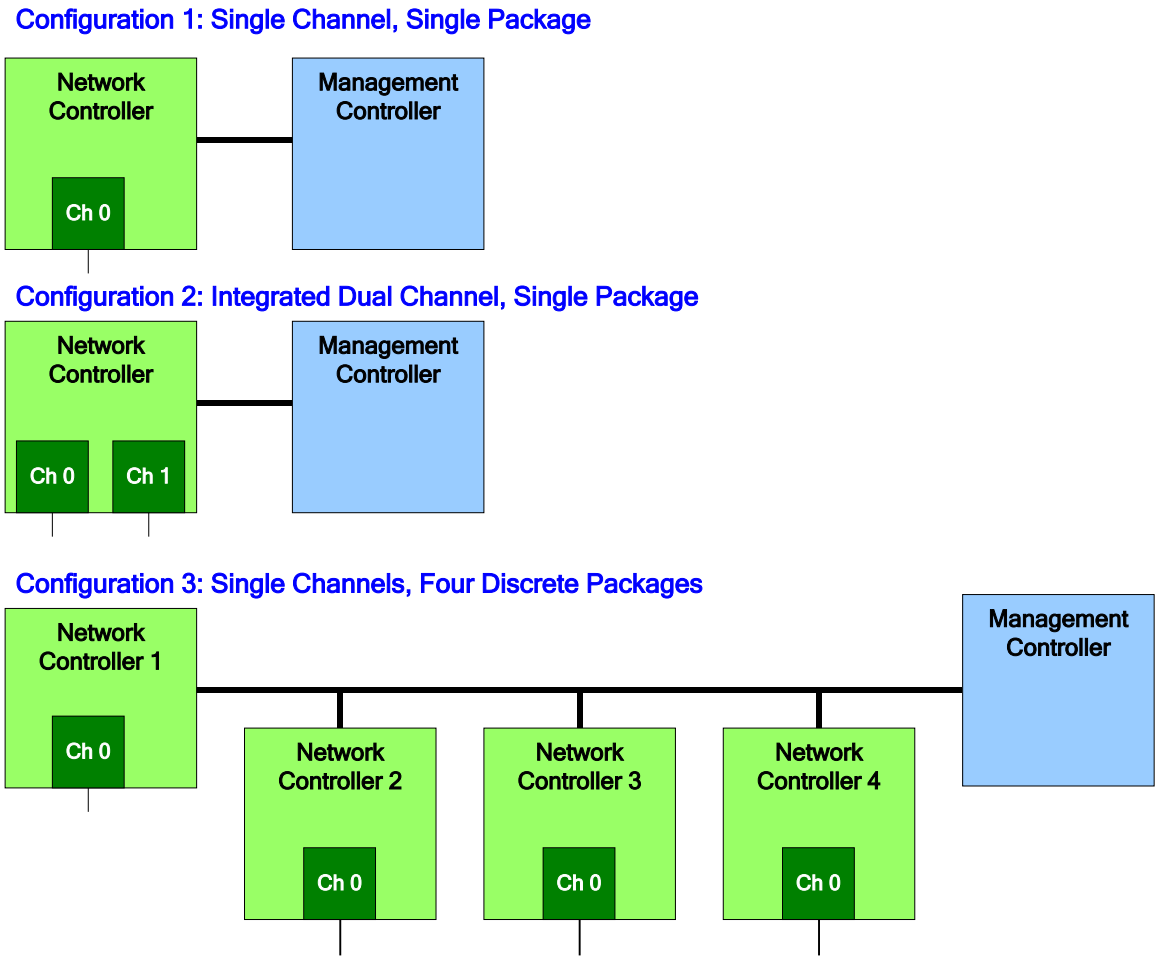
The NC-SI is intended to operate independently from the in-band activities of the Network Controller. As such, the Sideband Interface is not specified to be accessible through the host interface of the Network Controller. From the external world, this interface should behave and operate like a standard Ethernet Interface.

5.1 Defined topologies

The topologies supported under this specification apply to the case in which a single Management Controller is actively communicating with one or more Network Controllers on the Sideband Interface. The RBT electrical specification is targeted to directly support up to four physical Network Controller packages. The protocol specification allows up to eight Network Controller packages, with up to 31 channels per package.

Figure 3 illustrates some examples of Network Controller configurations supported by the NC-SI in the current release:

- Configuration 1 shows a Management Controller connecting to a single Network Controller with a single external network connection.
- Configuration 2 shows a Management Controller connecting to a Network Controller package that supports two NC-SI channel connections.
- Configuration 3 shows a Management Controller connecting to four discrete Network Controllers.



868

869 **Figure 3 – Example topologies supported by the NC-SI**

870 5.2 Single and integrated Network Controller implementations

871 This clause illustrates the general relationship between channels, packages, receive buffers, and bus
872 buffers for different controller implementations.

873 An integrated controller is a Network Controller that connects to the NC-SI and provides NC-SI support for
874 two or more network connections. A single controller is a controller that supports only a single NC-SI
875 channel.

876 For the *NC-SI Specification*, an integrated controller can be logically implemented in one of three basic
877 ways, as illustrated in Figure 4. Although only two channels are shown in the illustration, an integrated
878 controller implementation can provide more than two channels. The example channel and package
879 numbers (for example, channel 0, pkg 0) refer to the Internal Channel and Package ID subfields of the
880 Channel ID. For more information, see 6.2.9.

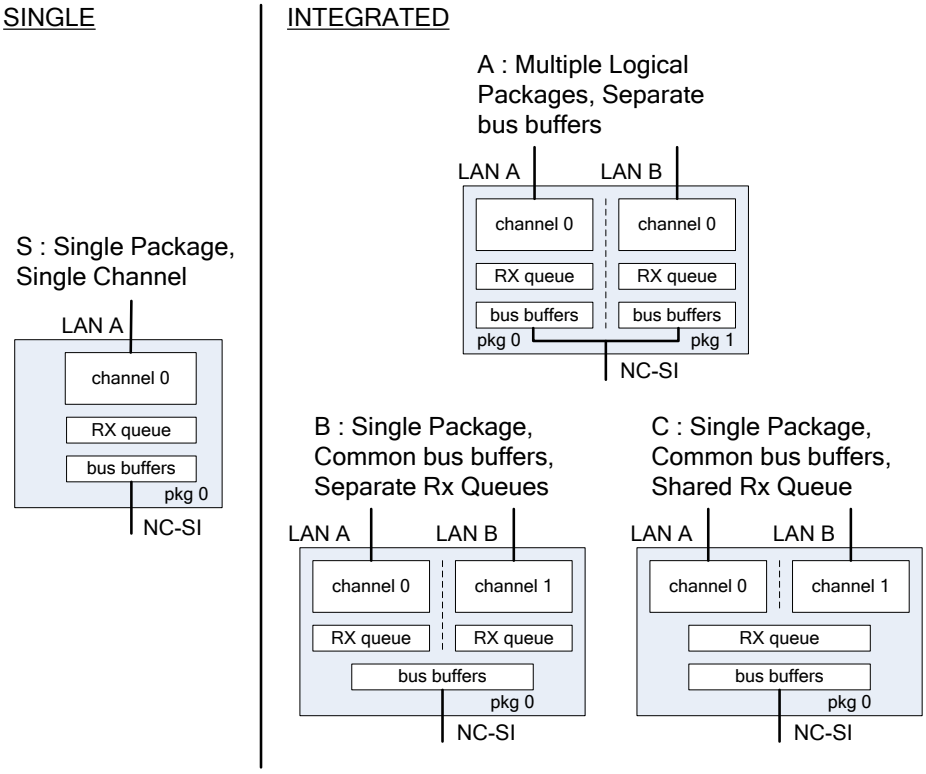


Figure 4 – Network Controller integration options

Packages that include multiple channels are required to handle internal arbitration between those channels and the Sideband Interface. The mechanism by which this occurs is vendor specific and not specified in this document. This internal arbitration is always active by default. No NC-SI commands are defined for enabling or disabling internal arbitration between channels.

The following classifications refer to a logical definition. The different implementations are distinguished by their *behavior* with respect to the NC-SI bus and command operation. The actual physical and internal implementation can vary from the simple diagrams. For example, an implementation can act as if it has separate RX queues without having physically separated memory blocks for implementing those queues.

- **S: Single Package, Single Channel**

  This implementation has a single NC-SI interface providing NC-SI support for a single LAN port, all contained within a package or module that has a single connection to the NC-SI physical bus.

- **A: Multiple Logical Packages, Separate Bus Buffers**

  This implementation acts like two physically separate Network Controllers that happen to share a common overall physical container. Electrically, they behave as if they have separate electrical buffers connecting to the NC-SI bus. This behavior may be accomplished by means of a passive internal bus or by separate physical pins coming from the overall package. From the point of view of the Management Controller and the NC-SI command operation, this implementation behaves as if the logical controllers were implemented as physically separate controllers.

  This type of implementation may or may not include internal hardware arbitration between the two logical Network Controller packages. If hardware arbitration is provided external to the package, it shall meet the requirements for hardware arbitration described later in this specification. (For more information, see 7.2.)

- **B: Single Package, Common Bus Buffers, Separate RX Queues**

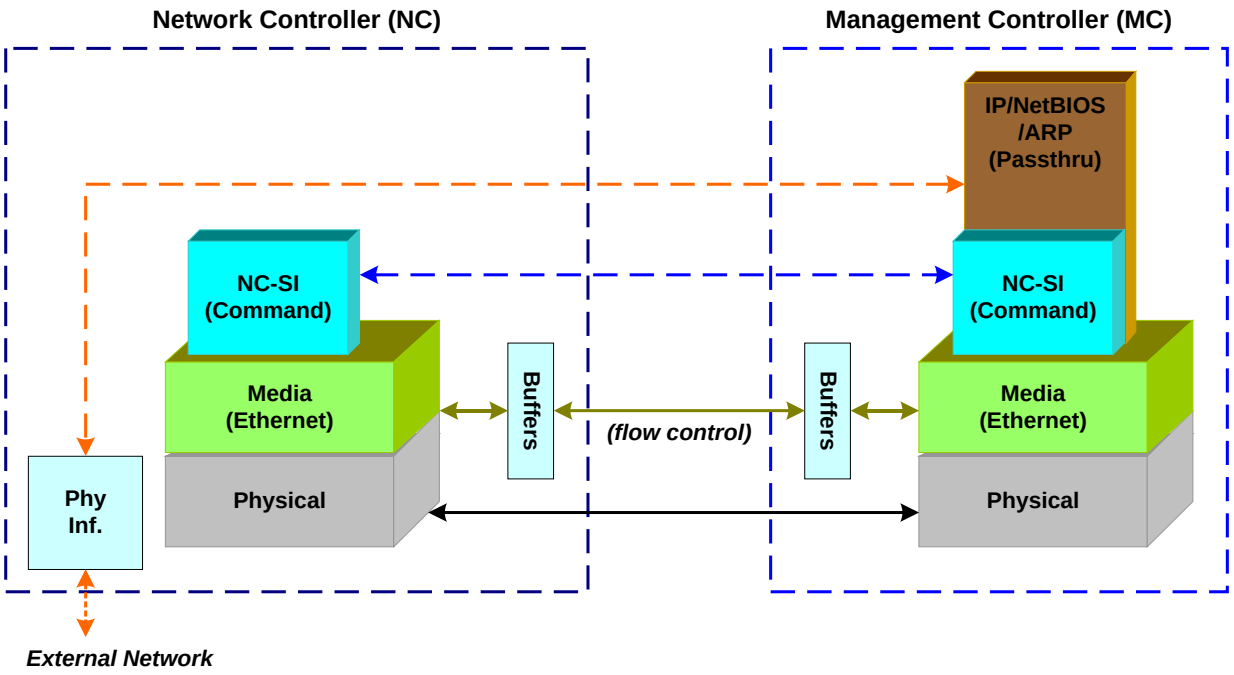
  In this implementation, the two internal NC-SI channels share a common set of electrical bus buffers. A single Deselect Package command will deselect the entire package. The Channel Enable and Channel Disable commands to each channel control whether the channel can transmit Pass-through and AEN packets through the NC-SI interface. The Channel Enable command also determines whether the packets to be transmitted through the NC-SI interface will be queued up in an RX Queue for the channel while the channel is disabled or while the package is deselected. Because each channel has its own RX Queue, this queuing can be configured for each channel independently.

- **C: Single Package, Common Bus Buffers, Shared RX Queue**

  This implementation is the same as described in the preceding implementation, except that the channels share a common RX Queue for holding Pass-through packets to be transmitted through the NC-SI interface. This queue may or may not also queue up AEN or Response packets.

5.3 Transport stack

The overall transport stack of the NC-SI is illustrated in Figure 5. The lowest level is the physical-level interface (for example, RBT), and the media-level interface is based on Ethernet. Above these interfaces are the two data-level protocols that are supported by the *NC-SI Specification*: NC-SI Command Protocol and the Network Data Protocol (for example, ARP, IP, DHCP, and NetBIOS) associated with Pass-through traffic. Both of these protocols are independent from binding to the underlying physical interface. This specification only defines the binding for NC-SI over RBT.



928

929 **Figure 5 – NC-SI transport stack**

930 This document defines the necessary NC-SI command set and interface specification that allows the
931 appropriate configuration of the Network Controller parameters and operation to enable network traffic to
932 flow to and from external networks to the Management Controller. As shown in Figure 5, the scope of the
933 NC-SI Command Protocol is limited to the internal interface between the Network Controller and the
934 Management Controller.

935 ## 5.4 Transport protocol

936 A simple transport protocol is used to track the reliable reception of command packets. The transport
937 protocol is based upon a command/response paradigm and involves the use of unique Instance IDs (IIDs)
938 in the packet headers to allow responses received to be matched to previously transmitted commands.
939 The Management Controller is the generator of command packets sent to the Sideband Interface of one
940 or more Network Controllers in the system, and it receives response packets from them. A response
941 packet is expected to be received for every command packet successfully sent.

942 The transport protocol described here shall apply only to command and response packets sent between
943 the Management Controller and the Network Controller.

944 ## 5.5 Byte and bit ordering for transmission

945 Unless otherwise specified, the bytes for a multi-byte numeric field are transmitted most significant byte
946 first and bits within a byte are transmitted most significant bit first.

947 # 6 Operational behaviors

948 This clause describes the NC-SI operating states and typical system-level operation of the NC-SI.

6.1 Typical operational model

This clause describes the typical system-level operation of the NC-SI components.

The following tasks are associated with Management Controller use of the NC-SI:

- **Initial configuration**

  When the NC-SI interface is first powered up, the Management Controller needs to discover and configure NC-SI devices in order to enable pass-through operation. This task includes setting parameters such as MAC addresses, configuring Layer 2 filtering, setting Channel enables, and so on.

- **General Controller configuration and monitoring**

  The Management Controller may also configure and monitor aspects of Controller operation.

- **Pass-through**

  The Management Controller handles transmitting and receiving Pass-through packets using the NC-SI. Pass-through packets can be delivered to and received from the network through the NC-SI based on the Network Controller's NC-SI configuration.

- **Asynchronous event handling**

  In certain situations, a status change in the Network Controller, such as a Link State change, can generate an asynchronous event on the Sideband Interface. These event notifications are sent to the Management Controller where they are processed as appropriate.

- **Error handling**

  The Management Controller handles errors that may occur during operation or configuration. For example, a Network Controller may have an internal state change that causes it to enter a state in which it requires a level of reconfiguration (this condition is called the "Initial State," described in more detail in 6.2.4); or a data glitch on the NC-SI could have caused an NC-SI command to be dropped by the Network Controller, requiring the Management Controller to retry the command.

6.2 State definitions

This clause describes NC-SI operating states.

6.2.1 General

Table 1 describes states related to whether and when the Network Controller is ready to handle NC-SI command packets, when it is allowed to transmit packets through the NC-SI interface, and when it has entered a state where it is expecting configuration by the Management Controller.

980 **Table 1 – NC-SI operating state descriptions**

State	Applies to	Description
Interface Power Down	Package	The NC-SI is in the power down state.
Interface Power Up	Package	The NC-SI is in the power up state, as defined in clause 10.
Package Selected (also referred to as the Selected state)	Package	A Selected package is allowed to turn on its electrical buffers and transmit through the NC-SI interface.
Package Deselected (also referred to as the Deselected state)	Package	A Deselected package is not allowed to turn on its electrical buffers and transmit through the NC-SI interface.
Hardware Arbitration Enabled	Package	When hardware arbitration is enabled, the package is allowed to transmit through the NC-SI interface only when it is Selected and has the TOKEN op-code.
Hardware Arbitration Disabled	Package	When hardware arbitration is disabled, the package is allowed to transmit through the NC-SI interface anytime that it is Selected, regardless of whether it has the TOKEN op-code.
Package Ready	Package	In the Package Ready state, the package is able to accept and respond to NC-SI commands for the package and be Selected.
Package Not Ready	Package	The Package Not Ready state is a transient state in which the package does not accept package-specific commands.
Channel Ready	Channel	In the Channel Ready state, a channel within the package is able to accept channel-specific NC-SI commands that are addressed to its Channel ID (Package ID + Internal Channel ID).
Channel Not Ready	Channel	The Channel Not Ready state is a transient state in which the channel does not accept channel-specific commands.
Initial State	Channel	In the Initial State, the channel is able to accept and respond to NC-SI commands, and one or more configuration settings for the channel need to be set or restored by the Management Controller (that is, the channel has not yet been initialized, or has encountered a condition where one or more settings have been lost and shall be restored). Refer to 6.2.4 for more information.
Channel Enabled	Channel	This is a sub-state of the Channel Ready state. When a channel is enabled, the channel is allowed to transmit unrequested packets (that is, packets that are not command responses—for example, AEN and Pass-through packets) through the NC-SI interface whenever the package is Selected.
Channel Disabled	Channel	This is a sub-state of the Channel Ready state. When a channel is disabled, the channel is not allowed to transmit unrequested packets (that is, packets that are not command responses—for example, AEN and Pass-through packets) through the NC-SI interface.

981 ### 6.2.2 NC-SI power states

982 Only two power states are defined for the NC-SI:

983 - **NC-SI Interface Power Down state**

984 In this state, the NC-SI Physical interface and the associated receive and transmit buffers in all
985 devices on the NC-SI (that is, the NC-SI interfaces on the Network Controllers and Management
986 Controller) are not powered up.

- **NC-SI Power Up state**

  In this state, the NC-SI Physical interface and the associated receive and transmit buffers in all devices on the NC-SI (that is, the Network Controller and Management Controller) are powered up. Note: NC transmit I/O buffers should not be enabled in this state. The Network Controller is expected to transition to the Initial State within T4 seconds after the Power Up state is entered.

6.2.3 Package Ready state

A Network Controller in the Package Ready state shall be able to respond to any NC-SI commands that are directed to the ID for the overall package (versus being directed to a particular channel within the package). Package-specific commands are identified by a particular set of Channel ID values delivered in the command header (see 6.2.9).

6.2.4 Initial State

The Initial State for a channel corresponds to a condition in which the Sideband Interface is powered up and is able to accept NC-SI commands, and the channel has one or more configuration settings that need to be set or restored by the Management Controller. Unless default configuration settings are explicitly defined in this specification, the default values are implementation specific. The MC should not make any assumptions on any configuration settings that are not defined in this specification. Because this state may be entered at any time, the Initial State shall be acknowledged with a Clear Initial State command for the Initial State to be exited. This requirement helps to ensure that the Management Controller does not continue operating the interface unaware that the NC-SI configuration had autonomously changed in the Network Controller.

An NC-SI channel in the Initial State shall:

- be able to respond to NC-SI commands that are directed to the Channel ID for the particular channel (see 6.2.9)
- respond to all non-OEM command packets that are directed to the channel with a Response Packet that contains a Response Code of “Command Failed” and a Reason Code of “Initialization Required”

NOTE This requirement does not apply to commands that are directed to the overall package, such as the Select Package and Deselect Package commands.

- place the channel into the Disabled state
- set hardware arbitration (if supported) to “enabled” on Interface Power Up only; otherwise, the setting that was in effect before entry into the Initial State shall be preserved (that is, the hardware arbitration enable/disable configuration is preserved across entries into the Initial State)
- set the enabled/disabled settings for the individual MAC and VLAN filters (typically set using the Set MAC Address,Set VLAN Filter, and Enable VLAN commands) to “disabled”

NOTE It is recommended that global multicast and broadcast filters are “disabled” in the Initial State. This means that all multicast and broadcast traffic is forwarded to the MC in the Initial State. An implementation may not have the global multicast or broadcast filters in “disabled” state in the Initial State. In this case, the MC may need to explicitly set global multicast and/or broadcast filters prior to enabling receiving pass-through traffic from the NC-SI channel.

- reset the counters defined in the Get NC-SI Statistics command and the Get NC-SI Pass-Through Statistics command to `0x0`
- disable transmission of Pass-through packets onto the network

NOTE Upon entry into the Initial State, the Channel Network TX setting is also set to “disabled”.

- clear any record of prior command instances received upon entry into the Initial State (that is, assume that the first command received after entering the Initial State is a new command and not a retried command, regardless of any Instance ID that it may have received before entering the Initial State)
- disable transmission of AENs

Otherwise, there is no requirement that other NC-SI configuration settings be set, retained, or restored to particular values in the Initial State.

The Initial State is a NC-SI configuration state and therefore places no requirements on the NC's network link state.

6.2.5 NC-SI Initial State recovery

As described in 6.2.4, a channel in the Initial State shall receive the Clear Initial State command before other commands can be executed. This requirement ensures that if the Initial State is entered asynchronously, the Management Controller is made aware that one or more NC-SI settings may have changed without its involvement, and blocks the Management Controller from issuing additional commands under that condition. Until the channel receives the Clear Initial State command, the Management Controller shall respond to any other received command (except the Select Package and Deselect Package commands) with a Command Failed response code and Interface Initialization Required reason code to indicate that the Clear Initial State command shall be sent. See response and reason code definitions in 8.2.4.1.

NOTE Package commands (for example, Select Package and Deselect Package) are always accepted and responded to normally regardless of whether the Channel is in the Initial State.

If the Management Controller, at any time, receives the response indicating that the Clear Initial State command is expected, it may interpret this response to mean that default settings have been restored for the channel (per the Initial State specification), and that one or more channel settings may need to be restored by the Management Controller.

6.2.6 State transition diagram

Figure 6 illustrates the general relationship between the package- and channel-related states described in Table 1 and the actions that cause transitions between the states. Each bubble in Figure 6 represents a particular combination of states as defined in Table 1.

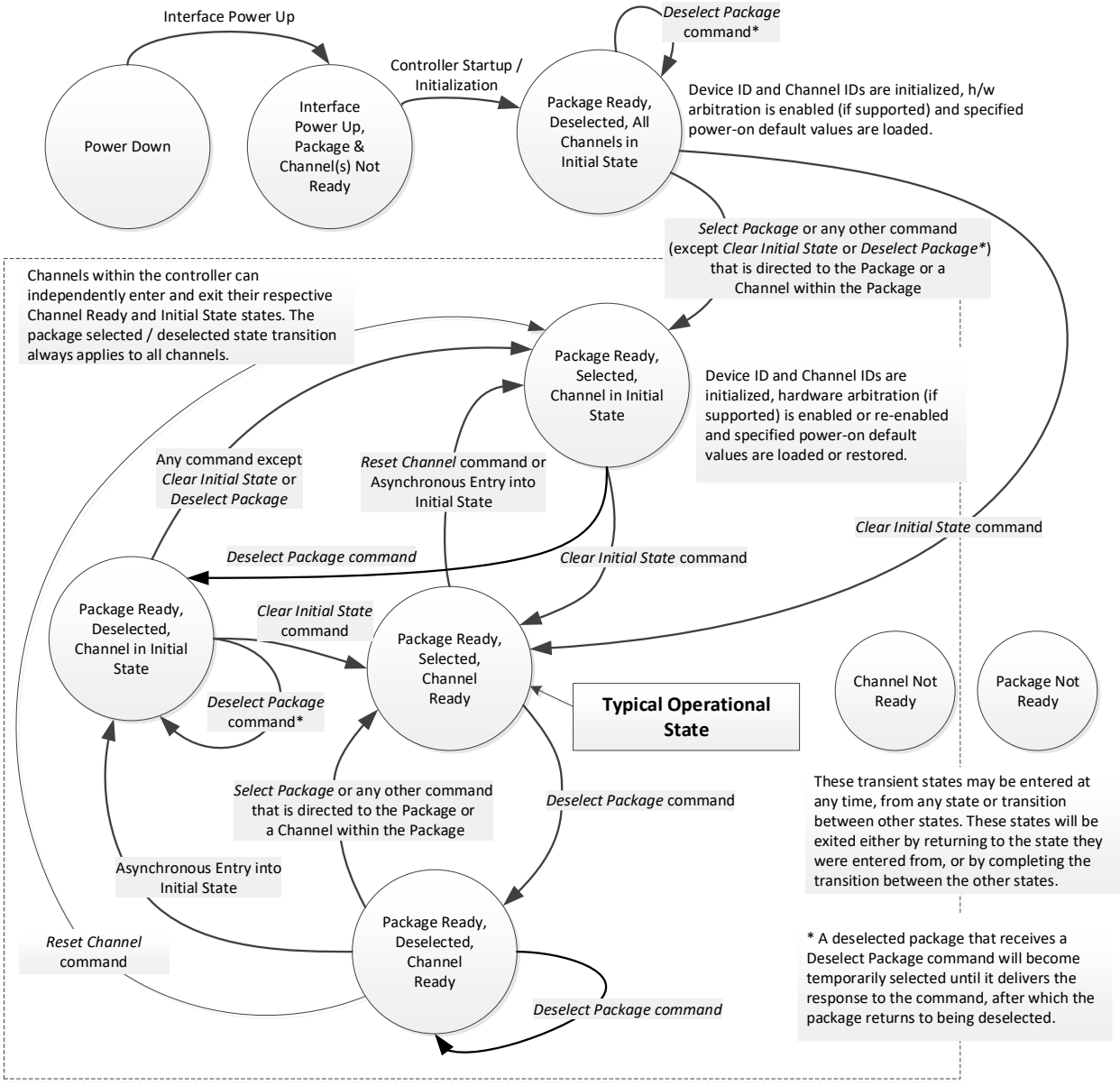


Figure 6 – NC-SI package/channel operational state diagram

6.2.7 State diagram for NC-SI operation with hardware arbitration

Figure 7 shows NC-SI operation in the hardware arbitration mode of operation. This is a sub-set of the general NC-SI operational state diagram (Figure 6) and has been included to illustrate the simplified sequence of package selection when this optional capability is used.

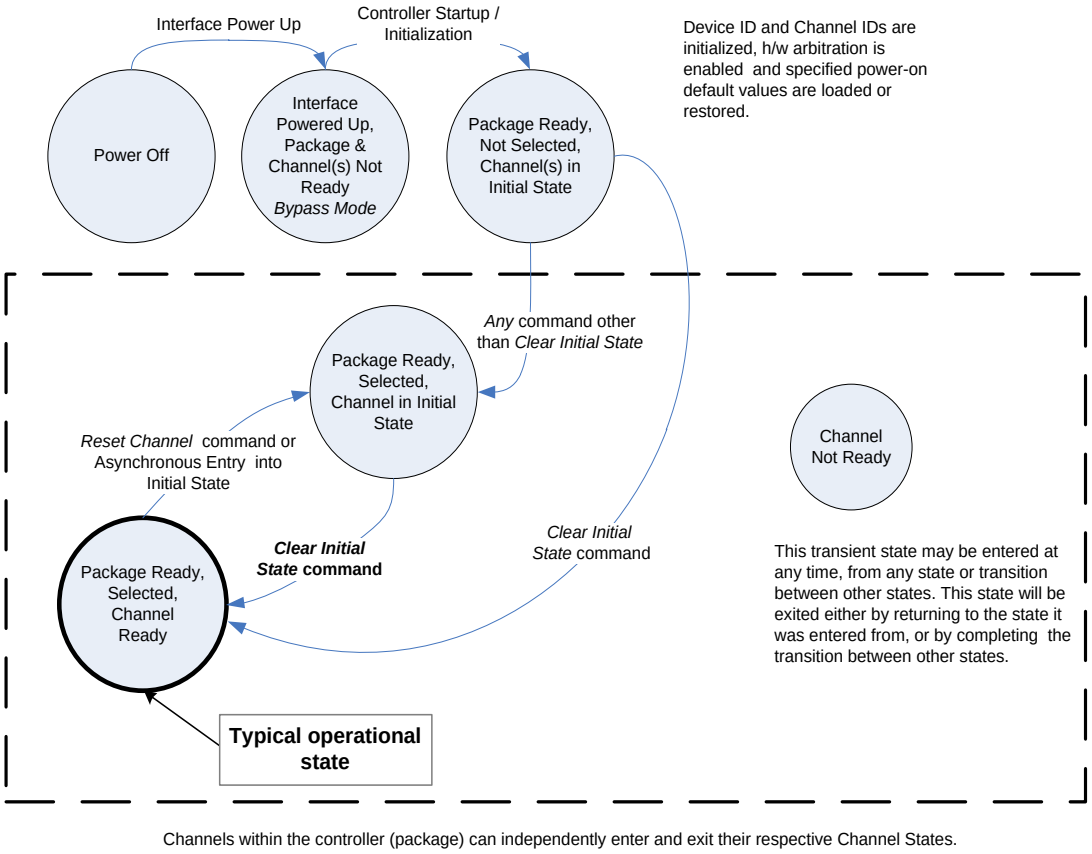


Figure 7 – NC-SI operational state diagram for hardware arbitration operation

While Select and Deselect package commands are not shown in Figure 7, these commands can be used with the HW arbitration and will behave as specified in this specification.

Select and Deselect package commands can work together with HW arbitration. If HW arbitration is enabled, a package needs both the HW arbitration token and to be selected in order to transmit on the NC-SI. If either the package is deselected or the package does not have HW arbitration token, then the package is not allowed to transmit on the NC-SI.

6.2.8 Resets

Two types of Reset events are defined for NC-SI Channels:

- Asynchronous Entry into Initial State
- Synchronous Reset

NOTE Resets that do not affect NC-SI operation are outside the scope of this specification.

6.2.8.1 Asynchronous entry into Initial State

An Asynchronous Reset event is defined as an event that results in a Channel asynchronously entering the Initial State. This event could occur as a consequence of powering up, a System Reset, a Driver Reset, an Internal Firmware error, loss of Configuration errors, Internal hardware errors, and so on. Additionally, it is recommended that any event in the NC that causes a total or partial loss of configuration should be interpreted as a Asynchronous Reset event

Unless otherwise specified, NC-SI configuration settings beyond those required by the Initial State may or may not be preserved following asynchronous entry into the Initial State, depending on the Network Controller implementation.

There is no explicit definition of a Reset for an entire package. However, it is possible that an Asynchronous Reset condition may cause an Asynchronous Entry into the Initial State for all Channels in a package simultaneously.

6.2.8.2 Synchronous Reset

A Synchronous Reset event on the NC-SI is defined as a Reset Channel command issued by a Management Controller to a Channel. Upon the receipt of this command, the Network Controller places the Channel into the Initial State.

Unless otherwise specified, NC-SI configuration settings beyond those required by the Initial State may or may not be preserved following a Synchronous Reset, depending on the Network Controller implementation.

6.2.9 Network Controller Channel ID

Each channel in the Network Controller shall be physically assigned a Network Controller Channel ID that will be used by the Management Controller to specify with which Network Controller channel, of possibly many, it is trying to communicate. The Network Controller Channel ID shall be physically assignable (configured) at system-integration time based on the following specification.

It is the system integrator’s or system designer’s responsibility to correctly assign and provide these identifier values in single- and multi-port Network Controller configurations, and to ensure that Channel IDs do not conflict between devices sharing a common NC-SI interconnect.

The Channel ID field comprises two subfields, Package ID and Internal Channel ID, as described in Table 2.

Table 2 – Channel ID format

Bits	Field Name	Description
[7..5]	Package ID	The Package ID is required to be common across all channels within a single Network Controller that share a common NC-SI physical interconnect. The system integrator will typically configure the Package IDs starting from 0 and increasing sequentially for each physical Network Controller. The Network Controller shall allow the least significant two bits of this field to be configurable by the system integrator, with the most significant bit of this field = `0b`. An implementation is allowed to have all 3 bits configurable.
[4..0]	Internal Channel ID	The Network Controller shall support Internal Channel IDs that are numbered starting from 0 and increasing sequentially for each channel supported by the Network Controller that is accessible by the Management Controller through the NC-SI using NC-SI commands. An implementation is allowed to support additional configuration options for the Internal Channel ID as long as the required numbering can be configured. An Internal Channel ID value of `0x1F` applies to the entire Package.

Channel IDs shall be completely decoded. Aliasing between values is not allowed (that is, the Network Controller is not allowed to have multiple IDs select the same channel on a given Sideband Interface).

Once configured, the settings of the Package ID and Internal Channel ID values shall be retained in a non-volatile manner. That is, they shall be retained across power-downs of the Sideband Interface and shall not be required to be restored by the Management Controller for NC-SI operation. This specification does not define the mechanism for configuring or retaining the Package ID or the Internal Channel ID (if configurable). Some implementations may use pins on the Network Controller for configuring the IDs, other implementations may use non-volatile storage logic such as electrically-erasable memory or FLASH, while others may use a combination of pins and non-volatile storage logic.

6.2.10 Configuration-related settings

This clause presents an overview of the different settings that the Management Controller may need to configure for NC-SI operation.

6.2.10.1 Package-specific operation

Only three NC-SI configuration settings are package-specific:

- the enable/disable settings for hardware arbitration
- NC-SI flow control
- Package-related AENs

There may also be NC configuration settings that are controlled by NC-SI Commands addressed to the package. These commands specify this requirement in their command description.

Hardware arbitration is enabled or disabled through a parameter that is delivered using the Select Package command. If hardware arbitration is enabled on all Network Controller packages on the NC-SI, more than one package can be in the Selected state simultaneously. Otherwise, only one package is allowed to be in the Selected state at a time in order to prevent electrical buffer conflicts (buffer fights) that can occur from more than one package being allowed to drive the bus.

NC-SI flow control is enabled or disabled using the Set NC-SI Flow Control command. The flow control setting applies to all channels in the package.

Package-specific commands should only be allowed and executed when the Channel ID field is set to 0x1F.

6.2.10.2 Channel-specific operation

Channel-specific commands should only be allowed to be executed when the Channel ID field is set to a value other than `0x1F`. Channel-specific commands with Invalid Channel IDs should not be allowed or executed. The recommended command response is Command Failed, Invalid Parameter.

Table 3 shows the major categories of configuration settings that control channel operation when a channel is in the Channel Ready state.

Table 3 – Channel Ready state configuration settings

Setting/Configuration Category	Description
"Channel Enable" settings	The Enable Channel and Disable Channel commands are used to control whether the channel is allowed to asynchronously transmit unrequested packets (AEN and Pass-through packets) through the NC-SI interface whenever the package is Selected. Note that channels are always allowed to transmit responses to commands sent to the channel.
Pass-through Transmit Enable settings	The Enable Channel Network TX command is used to enable the channel to transmit any Pass-through packets that it receives through the NC-SI onto the network, provided that the source MAC address in those packets matches the Network Controller settings. Correspondingly, the Disable Channel Network TX command is used to direct the controller not to transmit Pass-through packets that it receives onto the network.
AEN Enable settings	The AEN Enable command is used to enable and disable the generation of the different AENs supported by the Network Controller.
MAC Address Filter settings and control	The Set MAC Address, Enable Broadcast Filter, and Enable Global Multicast Filter commands are used to configure the filters for unicast, broadcast, and multicast addresses that the controller uses in conjunction with the VLAN Filter settings for filtering incoming Pass-through packets.
VLAN Filter settings and control	The Set VLAN Filter command is used to configure VLAN Filters that the controller uses in conjunction with the MAC Address Filters for filtering incoming Pass-through packets. The Enable VLAN and Disable VLAN commands are used to configure VLAN filtering modes and enable or disable whether VLAN filtering is used.

6.2.11 Transmitting Pass-through packets from the Management Controller

Packets not recognized as command packets (that is, packets without the NC-SI Ethertype) that are received on the Network Controller's NC-SI interface shall be assumed to be Pass-through packets provided that the source MAC Address matches one of the unicast MAC addresses settings (as configured by the Set MAC Address command) for the channel in the Network Controller, and will be forwarded for transmission to the corresponding external network interface if Channel Network TX is enabled.

6.2.12 Receiving Pass-through packets for the Management Controller

The Management Controller has control over and responsibility for configuring packet-filtering options, such as whether broadcast, multicast, or VLAN packets are accepted. Depending on the filter configurations, after the channel has been enabled, any packet that the Network Controller receives for the Management Controller shall be forwarded to the Management Controller through the NC-SI interface.

6.2.13 Startup sequence examples

The following clauses show possible startup sequences that may be used by the Management Controller to start NC-SI operation. Depending upon the specific configuration of each system, there are many possible variations of startup sequences that may be used, and these examples are intended for reference only.

6.2.13.1 Typical nonhardware arbitration specific startup sequence

The following sequence is provided as an example of one way a Management Controller can start up NC-SI operation. This sequence assumes that the Management Controller has no prior knowledge of how many Network Controllers are present on RBT, or what capabilities those controllers support. Note that this is not the only possible startup sequence. Alternative sequences can also be used to start up NC-SI operation. Some steps may be skipped if the Management Controller has prior knowledge of the Network Controller capabilities, such as whether Network Controllers are already connected and enabled for hardware arbitration.

1) **Power up**

   The NC-SI is powered up (refer to 10.2.7 for the specification of this condition). The Network Controller packages are provided a Network Controller Power Up Ready Interval during which they can perform internal firmware startup and initialization to prepare their NC-SI to accept commands. The Management Controller first waits for the maximum Network Controller Power Up Ready Interval to expire (refer to Table 218). At this point, all the Network Controller packages and channels should be ready to accept commands through the NC-SI. (The Management Controller may also start sending commands before the Network Controller Power Up Ready Interval expires, but will have to handle the case that Network Controller devices may be in a state in which they are unable to accept or respond to commands.)

2) **Discover package**

   The Management Controller issues a Select Package command starting with the lowest Package ID (see 8.4.5 for more information). Because the Management Controller is assumed to have no prior knowledge of whether the Network Controller is enabled for hardware arbitration, the Select Package command is issued with the Hardware Arbitration parameter set to ‘disable’.

   If the Management Controller receives a response within the specified response time, it can record that it detected a package at that ID. If the Management Controller does not receive a response, it is recommended that the Management Controller retry sending the command. Three total tries is typical. (This same retry process should be used when sending all commands to the Network Controller and will be left out of the descriptions in the following steps.) If the retries fail, the Management Controller can assume that no Network Controller is at that Package ID and can immediately repeat this step 2) for the next Package ID in the sequence.

3) **Discover and get capabilities for each channel in the package**

   The Management Controller can now discover how many channels are supported in the Network Controller package and their capabilities. To do this, the Management Controller issues

the Clear Initial State command starting from the lowest Internal Channel ID (which selects a given channel within a package). If it receives a response, the Management Controller can then use the Get Version ID command to determine NC-SI specification compatibility, and the Get Capabilities command to collect information about the capabilities of the channel. The Management Controller can then repeat this step until the full number of internal channels has been discovered. (The Get Capabilities command includes a value that indicates the number of channels supported within the given package.)

NOTE The *NC-SI Specification* requires Network Controllers to be configurable to have their Internal Channel IDs be sequential starting from 0. If it is known that the Network Controller is configured this way, the Management Controller needs only to iterate sequentially starting from Internal Channel ID = 0 up to the number of channels reported in the first Get Capabilities response.

The Management Controller should temporarily retain the information from the Get Capabilities command, including the information that reports whether the overall package supports hardware arbitration. This information is used in later steps.

4) **Repeat steps 2 and 3 for remaining packages**

The Management Controller repeats steps 2) and 3) until it has gone through all the Package IDs.

IMPORTANT: Because hardware arbitration has not been enabled yet, the Management Controller shall issue a Deselect Package command to the present Package ID before issuing the Select Package command to the next Package ID. If hardware arbitration is not being used, only one package can be in the Selected state at a time. Otherwise, hardware electrical buffer conflicts (buffer fights) will occur between packages.

5) **Initialize each channel in the package**

Based on the number of packages and channels that were discovered, their capabilities, and the desired use of Pass-through communication, the Management Controller can initialize the settings for each channel. This process includes the following general steps for each package:

a) Issue the Select Package command.

b) For each channel in the package, depending on controller capabilities, perform the following actions. Refer to individual command descriptions for more information.

- Use the Set MAC Address command to configure which unicast and multicast addresses are used for routing Pass-through packets to and from the Management Controller.
- Use the Enable Broadcast Filter command to configure whether incoming broadcast Pass-through packets are accepted or rejected.
- Use the Enable Global Multicast Filter command to configure how incoming multicast Pass-through packets are handled based on settings from the Set MAC Address command.
- Use the Set VLAN Filter and Enable VLAN Filters commands to configure how incoming Pass-through packets with VLAN Tags are handled.
- Use the Set NC-SI Flow Control command (if supported) to configure how Ethernet Pause Frames are used for flow control on the NC-SI. Note: Set NC-SI Flow Control is a package command and only needs to be issued once.
- Use the AEN Enable command to configure what types of AEN packets the channel should send out on the NC-SI.
- Use the Enable Channel Network TX command to configure whether the channel is enabled to deliver Pass-through packets from the NC-SI to the network (based on the

MAC address settings) or is disabled from delivering any Pass-through packets to the network.

c) Issue the Deselect Package command.

6) **Start Pass-through packet and AEN operation on the channels**

The channels should now have been initialized with the appropriate parameters for Pass-through packet reception and AEN operation. Pass-through operation can be started by issuing the Enable Channel command to each channel that is to be enabled for delivering Pass-through packets or generating AENs through the NC-SI interface.

NOTE If hardware arbitration is not operational and it is necessary to switch operation over to another package, a Deselect Package command shall be issued to the presently selected package before a different package can be selected. Deselecting a package blocks all output from the package. Therefore, it is not necessary to issue Disable Channel commands before selecting another package. There is no restriction on enabling multiple channels *within* a package.

6.2.13.2 Hardware arbitration-specific startup sequence

This clause applies when multiple NCs are used by the MC. This clause only applies to the NC-SI over RBT binding.

The following is an example of the steps that a Management Controller may perform to start up NC-SI operation when Hardware Arbitration is specifically known to be used, present, and enabled on all Network Controllers. This example startup sequence assumes a high level of integration where the Management Controller knows the Network Controllers support and default to the use of Hardware Arbitration on startup but does not have prior knowledge of how many Network Controllers are present on RBT, or the full set of capabilities those controllers support, so discovery is still required.

Although other startup examples may show a specific ordering of steps for the process of discovering, configuring and enabling channels, the Management Controller has almost total flexibility in choosing how these steps are performed once a channel in a package is discovered. In the end, it would be just as valid for a Management Controller to follow a breadth-first approach to discovery steps as it would be to follow a depth-first approach where each channel that is discovered is fully initialized and enabled before moving to the next.

1) **Power up**

No change from other startup scenarios.

2) **Discovery**

The process of discovery consists of identifying the number of packages that are available, the number of channels that are available in each package, and for each channel, the capabilities that are provided for Management Controller use. Because, in this startup scenario, the Management Controller knows Hardware Arbitration is used, it is not required to use the ***Select Package*** and ***Deselect Package*** commands for discovery but may elect to just use the ***Clear Initial State*** command for this purpose instead.

In this startup scenario, Packages and Channels are discovered by sending the ***Clear Initial State*** command starting with the lowest Package ID and Channel ID, then waiting for, and recording, the response event as previously described. Internal channel IDs are required to be numbered sequentially starting with 0, so when the Management Controller does not receive a response to repeated attempts at discovery, it knows this means no additional channels exist in the current package. If this happens when the internal channel ID is 0, the Management Controller knows a package is not available at the current package ID, and it continues with the next package ID in sequence. If the Management Controller receives a response to the ***Clear***

Initial State command, it records that the channel and package are available, and continues discovery.

During discovery, the Management Controller should interrogate the capabilities of each channel found to be available in each package by sending the ***Get Capabilities*** command appropriate package and channel ID values. However, it does not matter whether this is done as the very next step in the discovery process or performed for each channel after all packages and channels have been discovered, just as long as the Management Controller does interrogate each channel.

3) **Configure each channel and enable pass-through**

Once the existence of all packages and channels, and the capabilities of each channel, have been discovered and recorded, the Management Controller shall initialize and enable each channel as needed for use. The details of these steps remain essentially the same as have been previously stated, except to note that there are no restrictions on how they are performed. What this means is that the MC may perform these steps in any order across the channels in each package as it sees fit. The MC may fully initialize and enable each channel in each package one at a time or perform the same step on each channel in sequence before moving on to the next, or in a different order. The specific order of steps is not dictated by this specification.

6.2.13.3 Summary of scheme for the MC without prior knowledge of hardware arbitration

The following scheme describes the case when the MC does not have a priori knowledge of the hardware arbitration support across multiple NCs.

1. For each available NC,

    a. The MC checks whether a device supports the HW arbitration, using "**Get Capabilities**" commands (this implicitly selects the package).

    b. The MC issues "**Deselect Package**" for the NC (needed as at this stage we do not know whether all the devices support HW arbitration).

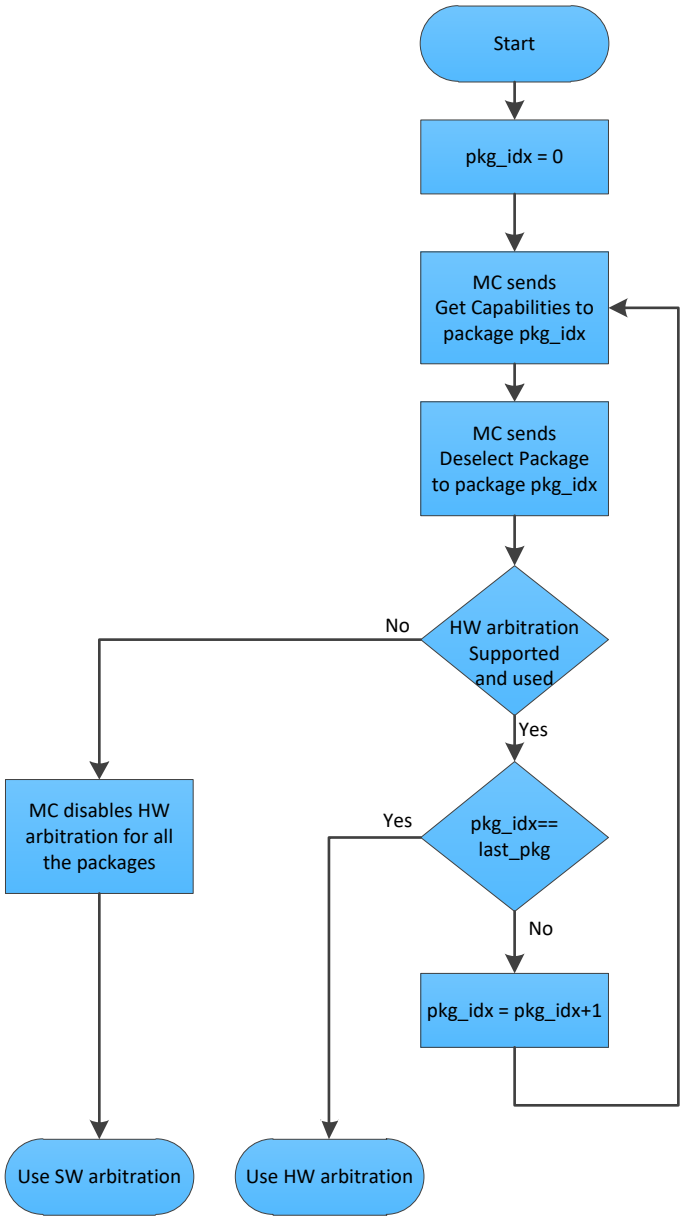
2. If (all NCs support HW arbitration and the HW arbitration is used by all NCs), then

    the MC assumes that HW arbitration is active because according to clause 6.2.4 "set hardware arbitration (if supported) to *enabled* on Interface Power Up only", and the MC can "Select" any number of packages at the same time.

    Otherwise (at least one NC reports that HW arbitration is not supported, or at least one NC reports that HW arbitration is not used, or at least one NC cannot report its support level)

    The HW arbitration is **not** active, and the MC can "Select" only single package at the any time.

    The MC configures each and every NC to disable HW arbitration, using the "***Select Package***" command.



1324

1325 **Figure 8 – MC steps when the MC does not have prior knowledge of hardware arbitration**

6.3 NC-SI traffic types

Two types of traffic are defined by NC-SI, based on the network fabric type: Pass-through traffic and Control traffic.

- Pass-through traffic consists of packets that are transferred between the external network interface and the Management Controller using the Sideband Interface.
- Control traffic consists of commands (requests) and responses that support the inventory, configuration and control of the Network Controller, the Sideband Interface and Pass-through operation of the Network Controller, and AENs that support reporting various events to the Management Controller.

6.3.1 Command protocol

Commands are provided to allow a Management Controller to initialize, control, and regulate Management Controller packet flow across the sideband interface, configure channel filtering, and to interrogate the operational status of the Network Controller. As interface master, the Management Controller is the initiator of all commands, and the Network Controller responds to commands, but may also generated AENs if enabled..

6.3.1.1 Instance IDs

The command protocol uses a packet field called the Instance ID (IID). IID numbers are 8-bit values that shall range from `0x01` to `0xFF`. IIDs are used to uniquely identify instances of a command, to improve the robustness of matching responses to commands, and to differentiate between new and retried commands. The Network Controller that receives a command handles the IID in the following ways:

- It returns the IID value from the command in the corresponding response.
- If the IID is the same as the IID for the previous command, it recognizes the command as a 'retried' command rather than as a new instance of the command. It is expected that the 'retried' command contains the same command type value in the Control Packet Type field. The NC behavior when a 'retried' command type does not match the original command type is outside the scope of this specification.
- If a retried command is received, the Network Controller shall return the previous response. Depending on the command, the Network Controller can accomplish this either by holding the previous response data so that it can be returned, or, if re-executing the command has no side effects (that is, the command is idempotent*)*, by re-executing the command operation and returning that response.
- If the command IID is the same as the IID for the previous command, and the Poll Indication is set, the NC recognizes the command as a 'polling' command rather than as a new instance of the command. When polling, the MC is expected to use the command type value of the original command in the Control Packet Type field. If there was no command in progress, the NC shall fail the 'polling' command and respond with an error.. When the NC fails the 'polling' command, the outcome of the original command is indeterminate and is outside the scope of this specification.
- If a command with Poll Indication set is received and the original command has been completed, then the Network Controller shall return the response of the completed command. If it is still processing the command, it shall return a "Delayed Response" reason code and optionally recommend a next polling time interval.
- When an IID value is received that is different from the one for the previous command, the Network Controller executes the command as a new command.

- When the NC-SI Channel first enters the Initial State, it clears any record of any prior requests. That is, it assumes that the first command after entering the Initial State is a new command and not a retried command, regardless of any IID that it may have received before entering the Initial State.

Thus, for single-threaded operation with idempotent commands, a responding Network Controller can simply execute the command and return the IID in the response that it received in the command. If it is necessary to not execute a retried command, the responding controller can use the IID to identify the retried command and return the response that was delivered for the original command.

The Management Controller that generates a command handles the IID in the following ways:

- The IID changes for each new instance of a command.
- If a command needs to be retried, the Management Controller uses the same value for the IID that it used for the initial command.
- The Management Controller can optionally elect to use the IID as a way to provide additional confirmation that the response is being returned for a particular command.

Because an AEN is not a response, an AEN always uses a value of `0x00` for its IID.

NOTE The Instance ID mechanism can be readily extended in the future to support multiple controllers and multiple outstanding commands. This extension would require having the responder track the IID on a per command and per requesting controller basis. For example, a retried command would be identified if the IID and command matched the IID and command for a prior command for the given originating controller's ID. That is, a match is made with the command, originating controller, and IID fields rather than on the IID field alone. A requester that generates multiple outstanding commands would correspondingly need to track responses based on both command and IID in order to match a given response with a given command. IIDs need to be unique for the number of different commands that can be concurrently outstanding.

6.3.1.2 Single-threaded operation

The Network Controller is required to support NC-SI commands only in a single-threaded manner. That is, the Network Controller is required to support processing only one command at a time and is not required to accept additional commands until after it has sent the response to the previous one.

Therefore, the Management Controller should issue NC-SI commands in a single-threaded manner. That is, the Management Controller should have only one command outstanding to a given Network Controller package at a time. Upon sending an NC-SI command packet, and before sending a subsequent command, the Management Controller should wait for the corresponding response packet to be received or a command timeout event to occur before attempting to send another command. For the full descriptions of command timeout, see 6.9.2.1.

6.3.1.3 Responses

The Network Controller shall process and acknowledge each validly formatted command received at the NC-SI interface by formatting and sending a valid response packet to the Management Controller through the NC-SI interface.

To allow the Management Controller to match responses to commands, the Network Controller shall copy the IID number of the Command into the Instance ID field of the corresponding response packet.

To allow for retransmission and error recovery, the Network Controller may re-execute the last command or maintain a copy of the response packet most recently transmitted to the Management Controller through its sideband interface. This "previous" response packet shall be updated every time a new response packet is transmitted to the Management Controller by replacing it with the one just sent.

The Network Controller response shall return a "Command Unsupported" response code with an "Unknown Command Type" reason code for any command (standard or OEM) that the Network Controller does not support or recognize.

6.3.1.4 Response and post-response processing

Typically, a Network Controller completes a requested operation before sending the response. In some situations, however, it may be useful for the controller to be allowed to queue up the requested operation and send the response assuming that the operation will complete correctly (for example, when the controller is requested to change link configuration). The following provisions support this process:

- A Network Controller is allowed to send a response before performing the requested action if the command is expected to complete normally and all parameters that are required to be returned with the response are provided.
- Temporal ordering of requested operations shall be preserved. For example, if one command updates a configuration parameter value and a following command reads back that parameter, the operation requested first shall complete so that the following operation returns the updated parameter.
- Under typical operation of the Network Controller, responses should be delivered within the Normal Execution Interval (T5) (see Table 218).
- Unless otherwise specified, all requested operations shall complete within the Asynchronous Reset/Asynchronous Not Ready interval (T6) following the response.
- If the Network Controller channel determines that the requested operation or configuration change has not been completed correctly after sending the response, the channel shall enter the Initial State.
- If the command response is dependent on the execution of the command and the command response cannot be provided within Normal Execution Interval (T5), then a “Delayed Response” response code may be returned. In this case, the MC can poll the command later with the “Poll Indication” set to retrieve the response. The decision on when the MC polls again can be based on one of the following criteria:
  - A fixed delay. In this case a delay greater than T5 is recommended.
  - If provided, based on the “recommended next polling time” in the original response
  - If AEN is enabled, based on reception of a “Delayed Response Ready AEN”
  - When using delayed responses, the NC shall complete the command processing within T14 sec.

6.3.1.5 NC-SI traffic ordering

This specification does not require any ordering between AENs, NC-SI responses, and NC-SI Pass-through packets. Specific transport binding specifications may require ordering between AENs, NC-SI responses, and NC-SI Pass-through packets.

6.4 Link configuration and control

The Network Controller provides commands to allow the Management Controller to specify the auto-negotiation, link speed, duplex settings, FEC algorithm, link training, Serdes lane configuration, and so on to be used on the network interface. For more information, see 8.4.21.

NOTE The Management Controller should make link configuration changes only when the host network driver is absent or non-operational.

6.4.1 Link Status

The Network Controller provides a Get Link Status command to allow the Management Controller to interrogate the configuration and operational status of the primary Ethernet links. The Management Controller may issue the Get Link Status command regardless of OS operational status.

6.5 Frame filtering for Pass-through mode

The Network Controller provides the option of configuring various types of filtering mechanisms for the purpose of controlling the delivery of received Ethernet frames to the Management Controller. These options include VLAN Tag filter, L2 address filters, MAC address support, and limited frame filtering using L3, L4 protocol header fields. All frames that pass frame filtering are forwarded to the Management Controller over the Sideband Interface.

6.5.1 Multicast filtering

The Network Controller may provide commands to allow the Management Controller to enable and disable global filtering of all multicast packets. The Network Controller may optionally provide one or more individual multicast filters, as well as DHCP v6, IPv6 Neighbor Advertisement, IPv6 Router Advertisement, IPv6 Neighbor Solicitation, and IPv6 MLD filters.

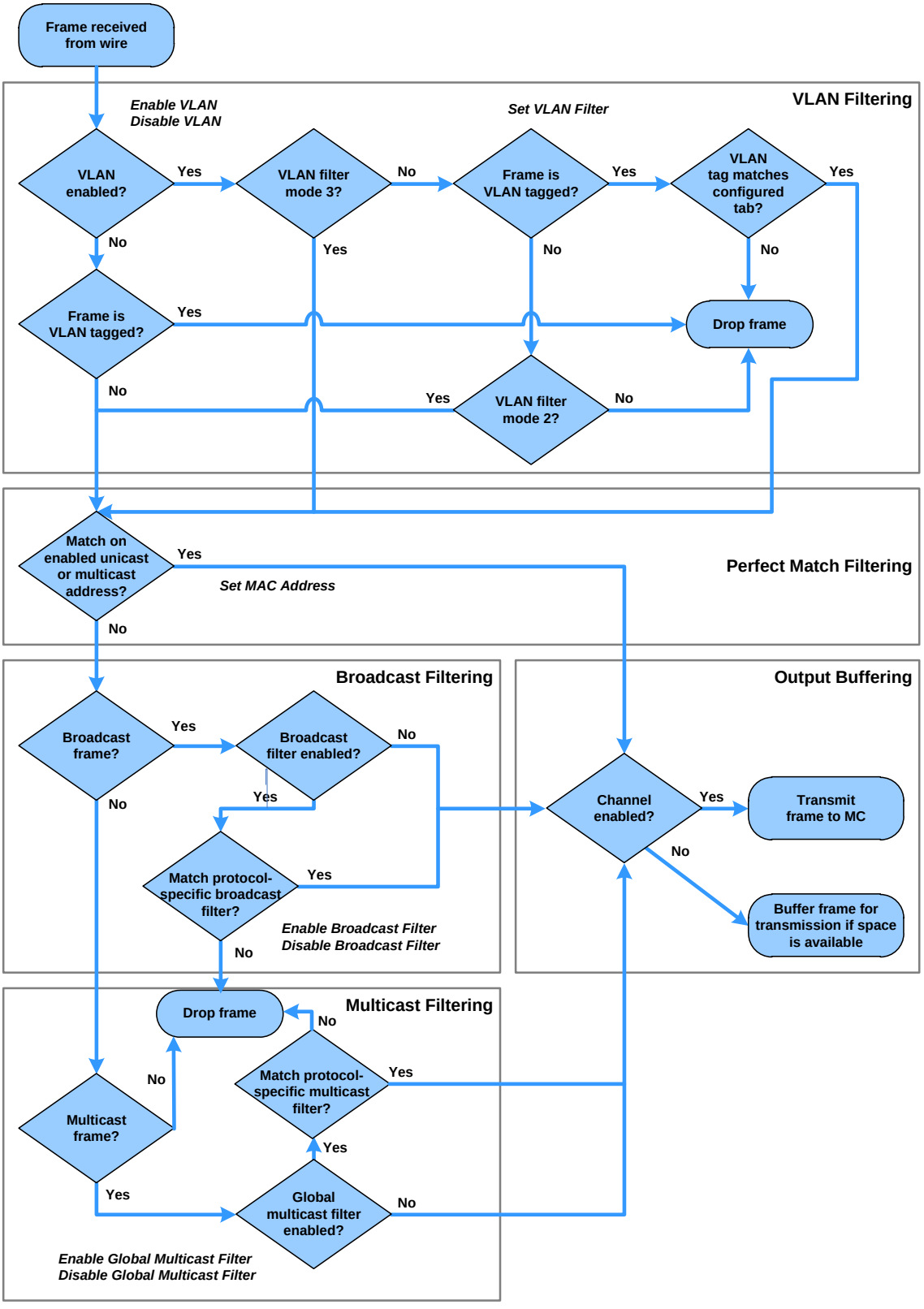
6.5.2 Broadcast filtering

The Network Controller provides commands to allow the Management Controller to enable and disable forwarding of Broadcast and ARP packets. The Network Controller may optionally support selective forwarding of broadcast packets for specific protocols, such as DHCP and NetBIOS.

6.5.3 VLAN filtering

The Network Controller provides commands to allow the Management Controller to enable and disable VLAN filtering, configure one or more VLAN Filters, and to configure VLAN filtering modes.

Figure 9 illustrates the flow of frame filtering. Italicized text in the figure is used to identify NC-SI command names.



1479

1480 **Figure 9 – NC-SI packet filtering flowchart**

6.6 Output buffering behavior

There are times when the NC is not allowed to transmit Pass-through, AEN, or Control Packets onto the Sideband Interface.

The NC should buffer Pass-through frames to be transmitted to the MC under any of the following conditions:

- The package is deselected.
- For a channel within a package while that channel is disabled.
- When the hardware arbitration is enabled and the NC does not have the token to transmit frames to the MC.

The NC may buffer AENs to the MC under any of the above conditions.

Control Packets (responses) are buffered when hardware arbitration is enabled and the NC does not have the token to transmit frames to the MC.

Additionally, while an NC-SI channel is in the initial state, previously received Pass-through frames and AENs may or may not be buffered. This behavior is outside the scope of this specification.

6.7 NC-SI flow control

The Network Controller may provide commands to enable flow control on the Sideband Interface between the Network Controller and the Management Controller. The NC-SI flow control behavior follows the PAUSE frame behavior as defined in the IEEE 802.3 specification. Flow control is configured using the Set NC-SI Flow command (see 8.4.41).

6.8 Asynchronous Event Notification

Asynchronous Event Notification (AEN) packets enable the Network Controller to deliver unsolicited notifications to the Management Controller when certain status changes that could impact interface operation occur in the Network Controller. Because the NC-SI is a small part of the larger Network Controller, its operation can be affected by a variety of events that occur in the Network Controller. These events include link status changes, OS driver loads and unloads, and chip resets. This feature defines a set of notification packets that operate outside of the established command-response mechanism.

Control over the generation of the AEN packets is achieved by control bits in the AEN Enable command. Each type of notification is optional and can be independently enabled by the Management Controller.

AENs are not acknowledged, and there is no protection against the possible loss of an AEN packet. Each defined event has its own AEN packet. Because the AEN packets are generated asynchronously by the Network Controller, they cannot implement some of the features of the other Control Packets. AEN packets leverage the general packet format of Control Packets.

- The originating Network Controller channel shall fill in its Channel ID (Ch. ID) field in the command header to identify the source of notification.
- The IID field in an AEN shall be set to `0x00` to differentiate it from a response or command packet.
- The Network Controller shall copy the AEN MC ID field from the AEN Enable command into the MC ID field in every AEN sent to the Management Controller.

6.9 Error handling

This clause describes the error-handling methods that are supported over the NC-SI. Two types of error-handling methods are defined:

- Synchronous Error Handling
- Errors that trigger Asynchronous Entry into the Initial State

Synchronous Error Handling occurs when an Error (non-zero) Response/Reason Code is received in response to a command issued by the Management Controller. For information about response and reason codes, see 8.2.4.1.

Asynchronous Entry into the Initial State Error Handling occurs when the Network Controller asynchronously enters the Initial State because of an error condition that affects NC-SI configuration or a failure of a command that was already responded to. For more information, see 6.2.8.1.

6.9.1 Transport errors

Transport error handling includes the dropping of command packets. Data packet errors are out of the scope of this specification.

6.9.1.1 Dropped Control Packets

The Network Controller shall drop Control Packets received on the NC-SI interface only under the following conditions:

- The packet has an invalid Frame Check Sequence (FCS) value.
- Frame length does not meet IEEE 802.3 requirements (except for OEM commands, where accepting larger packets may be allowed as a vendor-specific option).
- The packet checksum (if provided) is invalid.
- The NC-SI Channel ID value in the packet does not match the expected value.
- The Network Controller does not have resources available to accept the packet.
- The Network Controller receives a command packet with an incorrect header revision.

The Network Controller may also drop Control Packets if an event that triggers Asynchronous Entry into the Initial State causes packets to be dropped during the transition.

6.9.2 Missing responses

There are typical scenarios in which the Management Controller may not receive the response to a command:

- The Network Controller dropped the command and thus never sent the response.
- The response was dropped by the Management Controller (for example, because of a CRC error in the response packet).
- The Network Controller is in the process of being reset or is disabled.

The Management Controller can detect a missing response packet as the occurrence of an NC-SI command timeout event.

6.9.2.1 Command timeout

The Management Controller can detect missing responses by implementing a command timeout interval. The timeout value chosen by the Management Controller shall not be less than Normal Execution Interval, T5. Upon detecting a timeout condition, the Management Controller should not make assumptions on the state of the unacknowledged command (for example, the command was dropped or the response was dropped), but should retransmit (retry) the previous command using the same IID it used in the initial command.

The Management Controller should try a command at least three times before assuming an error condition in the Network Controller.

It is possible that a Network Controller could send a response to the original command at the same time a retried command is being delivered. Under this condition, the Management Controller could get more than one response to the same command. Thus, the Management Controller should be capable of determining that it has received a second instance of a previous response packet. Dropped commands may be detected by the Management Controller as a timeout event waiting for the response.

6.9.2.2 Handling dropped commands or missing responses

To recover from dropped commands or missing responses, the Management Controller can retransmit the unacknowledged command packet using the same IID that it used for the initial command.

The Network Controller shall be capable of reprocessing retransmitted (retried) commands without error or undesirable side effects. The Network Controller can determine that the command has been retransmitted by verifying that the IID is unchanged from the previous command.

6.9.3 Detecting Pass-through traffic interruption

The Network Controller might asynchronously enter the Initial State because of a reset or other event. In this case, the Network Controller stops transmitting Pass-through traffic on the RXD lines. Similarly, Pass-through traffic sent to the Network Controller may be dropped. If the Management Controller is not in the state of sending or receiving Pass-through traffic, it may not notice this condition. Thus the Management Controller should periodically issue a command to the Network Controller to test whether the Network Controller has entered the Initial State. How often this testing should be done is a choice of the Management Controller.

7 Arbitration in configurations with multiple Network Controller packages

This clause applies to NC-SI over RBT only. More than one Network Controller package on an NC-SI over RBT can be enabled for transmitting packets to the Management Controller. This specification defines two mechanisms to accomplish Network Controller package arbitration operations. One mechanism uses software commands provided by the Network Controller for the Management Controller to control whose turn it is to transmit traffic. The other mechanism uses hardware arbitration to share the single RBT bus. Implementations are required to support command-based Device Selection operation; the hardware arbitration method is optional.

7.1 General

Figure 10 is a simplified block diagram of the Sideband Interface being used in a multi-drop configuration. The RMII (upon which NC-SI is based) was originally designed for use as a point-to-point interconnect. Accordingly, only one party can transmit data onto the bus at any given time. There is no arbitration protocol intrinsic in the RMII to support managing multiple transmitters.

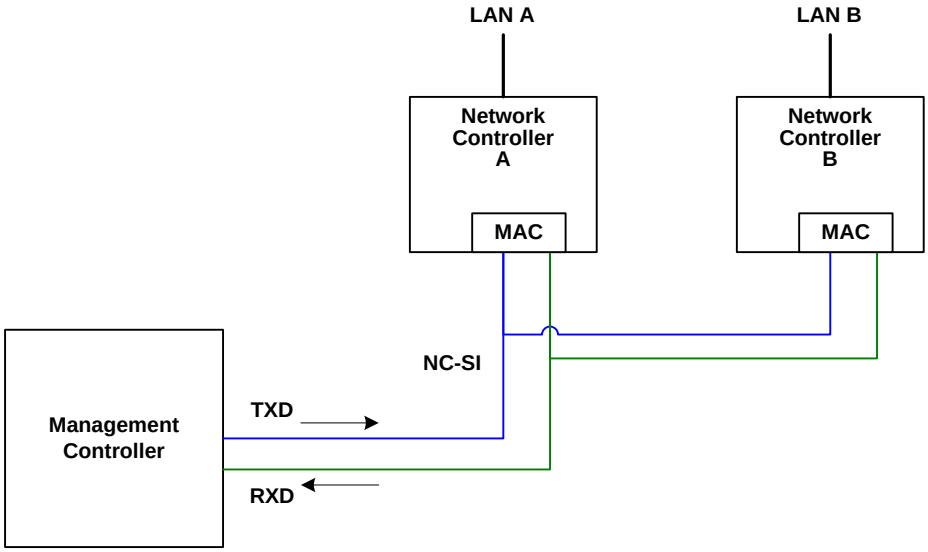


Figure 10 – Basic multi-drop block diagram

However, it is possible for multiple Network Controllers on the interface to be able to simultaneously *receive* traffic from the Management Controller that is being transmitted on the RBT TXD lines. The Network Controllers can receive commands from the Management Controller without having to arbitrate for the bus. This facilitates the Management Controller in delivering commands for setup and configuration of arbitration.

Arbitration allows multiple Network Controller packages that are attached to the interface to be enabled to share the RXD lines to deliver packets to the Management Controller.

This operation is summarized as follows:

- Only one Network Controller at a time can transmit packets on the RXD lines of the interface.
- Network Controllers can accept commands for configuring and controlling arbitration for the RXD lines.

7.2 Hardware arbitration

To prevent two or more NC-SI packages from transmitting at the same time, a hardware-based arbitration scheme was devised to allow only one Network Controller package to drive the RX lines of the shared interface at any given time. This scheme uses a mechanism of passing messages (op-codes) between Network Controller packages to coordinate when a controller is allowed to transmit through the RBT interface.

7.2.1 General

Three conceptual modes of hardware arbitration exist: arbitration master assignment, normal operation, and bypass. After a package is initialized and has its Channel IDs assigned, it enters the arbitration master assignment mode. This mode assigns one package the role of an Arbitration Master (ARB_Master) that is responsible for initially generating a TOKEN op-code that is required for the normal operating mode. In the normal operating mode, the TOKEN op-code is passed from one package to the next in the ring. The package is allowed to use the shared RXD signals and transmit if the package has received the TOKEN op-code and has a packet to send.

Bypass mode allows hardware arbitration op-codes to pass through a Network Controller package before it is initialized. Bypass mode shall be in effect while hardware arbitration is disabled. Bypass mode shall be exited and arbitration master assignment mode shall be entered when the hardware arbitration becomes enabled or re-enabled.

Hardware-based arbitration requires two additional pins (ARB_IN and ARB_OUT*)* on the Network Controller. The ARB_OUT pin of one package is connected to the ARB_IN pin of the next package to form a ring configuration, as illustrated in Figure 11. The timing requirements for hardware arbitration are designed to accommodate a maximum of four Network Controller packages. If the implementation consists of a single Network Controller package, the ARB_OUT pin may be connected to the ARB_IN pin on the same package, or may be left disconnected, in which case hardware arbitration should be disabled by using the Select Package command. This specification optionally supports reporting of Hardware arbitration implementation status and hardware arbitration status using the **Get Capabilities** command.

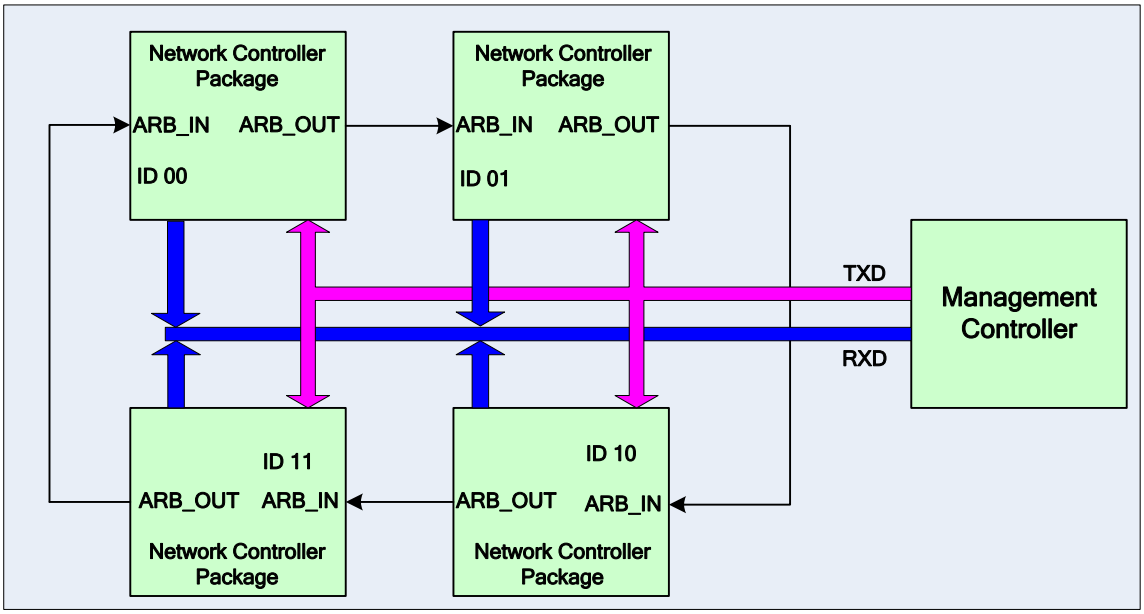


Figure 11 – Multiple Network Controllers in a ring format

Each Network Controller package sends out pulses on the ARB_OUT pin to create a series of symbols that form op-codes (commands) between Network Controllers. Each pulse is one clock wide and synchronized to REF_CLK. The hardware arbitration data bits follow the same timing specifications used for the TXD and RXD data bits (see 10.2.6). The pulses are di-bit encoded to ensure that symbols are correctly decoded. The symbols have the values shown in Table 4.

While clause 7.2.2.1 allows for op-code to be truncated, it is recommended that the transmission of current op-code on ARB_OUT be completed if the HW arbitration mode is changed in the middle of an op-code transfer (or in the middle of a symbol).

1646 **Table 4 – Hardware arbitration di-bit encoding**

Symbol Name	Encoded Value
Esync	11b
Ezero	00b
Eone	01b
Illegal symbol	10b

1647 ### 7.2.2 Hardware arbitration op-codes

1648 The hardware-based arbitration feature has five defined op-codes: IDLE, TOKEN, FLUSH, XON, and
1649 XOFF. Each op-code starts with an Esync symbol and is followed by either E_{one} or E_{zero} symbols. The
1650 legal op-codes are listed in Table 5.

1651 **Table 5 – Hardware arbitration op-code format**

Op-Code	Format
IDLE	E_{sync} E_{zero} E_{zero} (110000b)
TOKEN	E_{sync} E_{one} E_{zero} (110100b)
FLUSH	E_{sync} E_{one} E_{one} E_{zero} E(Package_ID[2:0]) E_{zero} (11010100xxxxxx00b)
XOFF	E_{sync} E_{zero} E_{one} E_{zero} E_{zero} E_{zero} (1100010000000b)
XON	E_{sync} E_{zero} E_{one} E_{one} E_{zero} E(Package_ID[2:0]) E_{zero} (1100010100uuuuuu00b)

1652 #### 7.2.2.1 Detecting truncated op-codes

1653 A truncated op-code is detected when the number of clocks between E_{sync}s is less than the number of bits
1654 required for the op-code. Note that any additional bits clocked in after a legitimate op-code is detected do
1655 not indicate an error condition and are ignored until the next E_{sync}.

1656 #### 7.2.2.2 Handling truncated or illegal op-codes

1657 When a Network Controller receives a truncated or illegal op-code, it should discard it.

1658 #### 7.2.2.3 Relationship of op-codes processing and driving the RX data lines

1659 A Network Controller package shall take no more than T9 REF_CLK times after receiving the last bit of
1660 the op-code to decode the incoming op-code and start generating the outgoing op-code. This time limit
1661 allows for decoding and processing of the incoming op-code under the condition that an outgoing op-code
1662 transmission is already in progress.

1663 A package that has received a TOKEN and has packet data to transmit shall turn on its buffer and begin
1664 transmitting the packet data within T11 REF_CLK times of receiving the TOKEN, as illustrated in
1665 Figure 12. The package shall disable the RXD buffers before the last clock of the transmitted TOKEN.

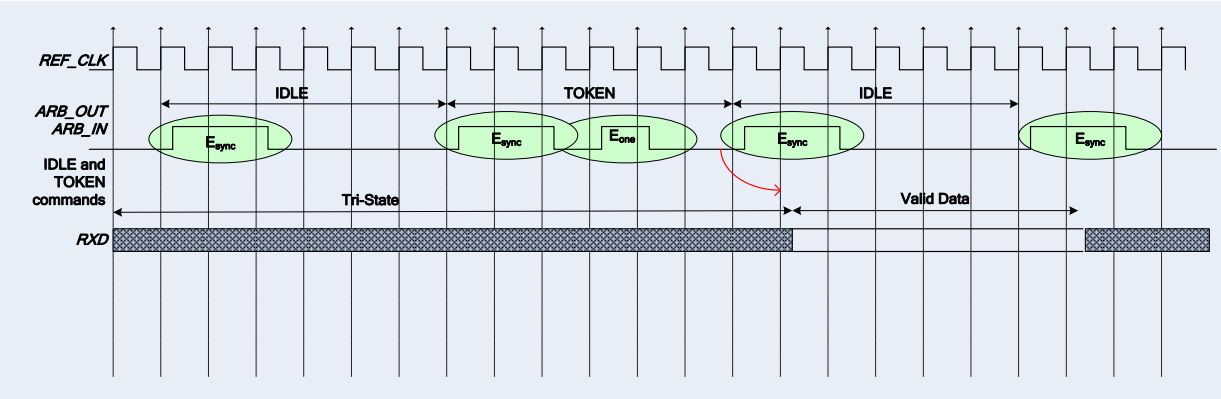


Figure 12 – Op-code to RXD relationship

7.2.3 Op-code operations

This clause describes the behavior associated with the five defined op-codes.

7.2.3.1 TOKEN op-code

When a TOKEN op-code is received, the Network Controller package may drive the RXD signals to send only one of the following items: a Pass-through packet, a command response, or an AEN. One IEEE 802.3 PAUSE frame (XON or XOFF) may also be sent either before or after one of the previous packets, or on its own. While the Network Controller package is transmitting the data on the RXD signals of the interface, it shall generate IDLE op-codes on its ARB_OUT pin. Once a package completes its transmission, if any, it shall generate and send the TOKEN on its ARB_OUT pin.

7.2.3.2 IDLE op-code

A package that has no other op-code to send shall continuously generate IDLE op-codes. Typically, a received IDLE op-code indicates that the TOKEN is currently at another package in the ring. This op-code is also used in the ARB_Master assignment process (for details, see 7.2.5). An Idle op-code typically will also be generated when the package is transmitting on RBT

7.2.3.3 FLUSH op-code

A FLUSH op-code is used to establish an Arbitration Master for the ring when the package enters the Package Ready state or when the TOKEN is not received within the specified timeout, T8. This op-code is further explained in 7.2.5.

If the package receives a FLUSH op-code while it is in the middle of transmitting a packet onto NC-SI, it shall generate IDLE op-codes until the transmission is complete and then process the FLUSH op-code as described.

7.2.3.4 Flow Control op-codes

The XON and XOFF op-codes are used to manage the generation of IEEE 802.3 PAUSE frames on the RBT interface. If the Network Controller supports flow control and flow control is enabled, the XOFF and XON op-codes behave as described in this clause. If the Network Controller does not support flow control or if flow control is not enabled, the Network Controller shall pass the op-codes to the next package.

There may be a configuration where some NCs support flow control and others do not. In this configuration, an NC sending an XOFF op-code may see the XOFF packet emission delayed by two or

more full size Pass-through packets, one for each package not supporting XOFF when it gets the token, and one for the next package supporting XOFF before sending the XOFF packet. The NC is not required to provide buffering to prevent packet loss in this configuration. No drop behavior should be expected by an MC only if all NCs have flow control enabled.

NOTE There is a maximum amount of time that the Network Controller may maintain a PAUSE. For more information, see 8.4.41.

7.2.3.4.1 XOFF op-code

A Network Controller package that becomes congested while receiving packets from the NC-SI shall perform the following actions:

- If it does not have a TOKEN, it sends the XOFF op-code to the next package.

  NOTE If it has the TOKEN and has not previously sent an XOFF frame for this instance of congestion, it shall send a single XOFF frame (PAUSE frame with a pause time of `0xFFFF`) and will not generate an XOFF op-code.

- A package may also regenerate an XOFF frame or op-code if it is still congested and determines that the present PAUSE frame is about to expire.

When a package on the ring receives an XOFF op-code, it shall perform one of the following actions:

- If it does not have a TOKEN op-code, it passes the XOFF op-code to the next package in the ring.
- If it has the TOKEN, it shall send an XOFF frame (PAUSE frame with a pause time of `0xFFFF`) and will not regenerate the XOFF op-code. If it receives another XOFF op-code while sending the XOFF frame or a regular network packet, it discards the received XOFF op-code.

7.2.3.4.2 XON op-code

XON frames (PAUSE frame with a pause time of `0x0000`) are used to signal to the Management Controller that the Network Controller packages are no longer congested and that normal traffic flow can resume. XON op-codes are used between the packages to coordinate XON frame generation. The package ID is included in this op-code to provide a mechanism to verify that every package is not congested before sending an XON frame to the Management Controller.

The XON op-code behaves as follows:

- When a package is no longer congested, it generates an XON op-code with its own Package ID. This puts the package into the ‘waiting for its own XON’ state.
- A package that receives the XON op-code takes one of the following actions:
  - If it is congested, it replaces the received XON op-code with the IDLE op-code. This action causes the XON op-code to be discarded. Eventually, the congested package generates its own XON op-code when it exits the congested state.
  - If the package is not congested and is not waiting for the XON op-code with own Package ID, it forwards the received XON op-code to the next package in the ring.

    NOTE If the received XON op-code contains the package’s own Package ID, the op-code should be discarded.

  - If the package is not congested and is waiting for its own XON op-code, it performs one of the following actions:
    - If it receives an XON op-code with a Package ID that is higher than its own, it replaces the XON op-code with its own Package ID.

- If it receives an XON op-code with a Package ID lower than its own, it passes that XON op-code to the next package and it exits the 'waiting for its own XON' state.
- If it receives an XON op-code with the Package ID equal to its own, it sends an XON frame on the NC-SI when it receives the TOKEN op-code and exits the 'waiting for its own XON' state.

NOTE More than one XON op-code with the same Package ID may be received while waiting for the TOKEN and while sending the XON frame. These additional XON op-codes should be discarded.

- If a package originates an XON op-code but receives an XOFF op-code, it terminates its XON request so that it does not output an XON frame when it receives the TOKEN.

NOTE This behavior should not occur because the Management Controller will be in the Pause state at this point.

- A package that generated an XON op-code may receive its own XON op-code back while it has the TOKEN op-code. In this case, it may send a regular packet (Pass-through, command response, or AEN) to the Management Controller (if it has one to send), an XON frame, or both.

7.2.4 Bypass mode

When the Network Controller package is in bypass mode, data received on the ARB_IN pin is redirected to the ARB_OUT pin within the specified clock delay. This way, arbitration can continue between other devices in the ring.

A package in bypass mode shall take no more than T10 REF_CLK times to forward data from the ARB_IN pin to the ARB_OUT pin. The transition in and out of bypass mode may result in a truncated op-code.

A Network Controller package enters into bypass mode immediately upon power up and transitions out of this mode after the Network Controller completes its startup/initialization sequence.

7.2.5 Hardware arbitration startup

Hardware arbitration startup works as follows:

1) All the packages shall be in bypass mode within T_{pwrz} seconds of NC-SI power up.
2) As each package is initialized, it shall continuously generate FLUSH op-codes with its own Package ID.
3) The package then participates in the ARB_MSTR assignment process described in the following clause.

7.2.6 ARB_MSTR assignment

ARB_MSTR assignment works as follows:

1) When a package receives a FLUSH op-code with a Package ID numerically smaller than its own, it shall forward on the received FLUSH op-code. If the received FLUSH op-code's Package ID is numerically larger than the local Package ID, the package shall continue to send its FLUSH op-code with its own Package ID. When a package receives a FLUSH op-code with its own Package ID, it becomes the master of the ring (ARB_MSTR).
2) The ARB_MSTR shall then send out IDLE op-codes until it receives an IDLE op-code.
3) Upon receiving the IDLE op-code, the ARB_MSTR shall be considered to be in possession of the TOKEN op-code (see 7.2.3.1).

NOTE If the package receives a FLUSH op-code while it is in the middle of transmitting a packet onto NC-SI, it shall generate IDLE op-codes until the transmission is complete and then process the FLUSH op-code as described.

7.2.7 Token timeout mechanism

Each Network Controller package that supports hardware-based arbitration control shall implement a timeout mechanism in case the TOKEN op-code is not received. When a package has a packet to send, it starts its timer. If it does not receive a TOKEN prior to the TOKEN timeout, the package shall send a FLUSH op-code. This restarts the arbitration process.

The timer may be programmable depending on the number of packages in the ring. The timeout value is designed to accommodate up to four packages, each sending the largest packet (1536 bytes) plus possible XON or XOFF frame transmission and op-code processing time. The timeout shall be no fewer than T8 cycles of the REF_CLK.

7.2.8 Timing considerations

The ARB_OUT and ARB_IN pins shall follow the timing specifications outlined in Clause 10.

To improve the efficiency of the multi-drop NC-SI, TOKEN op-code generation may overlap the Inter Packet Gap (IPG) defined by the 802.3 specification, as shown in Figure 13. The TOKEN op-code shall be sent no earlier than the last T13 REF_CLK cycles of the IPG.

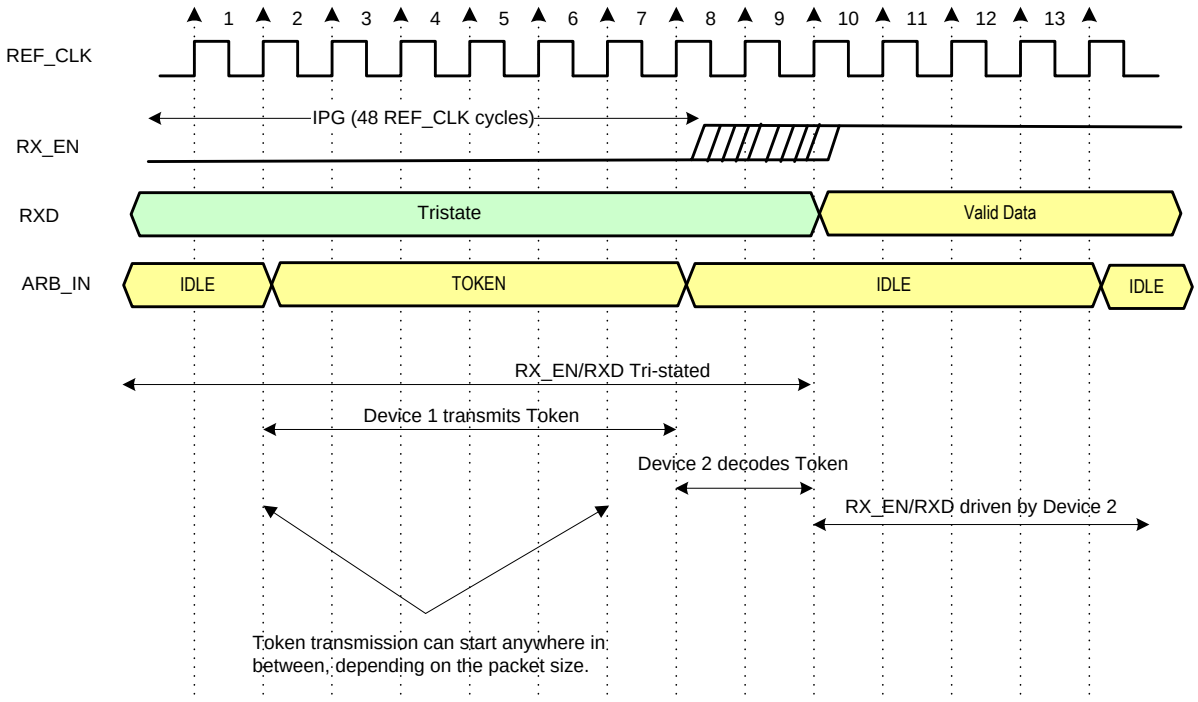


Figure 13 – Example TOKEN to transmit relationship

7.2.9 Example hardware arbitration state machine

The state machine diagram shown in Figure 14 is provided as a guideline to help illustrate the startup process and op-code operations described in the preceding clauses.

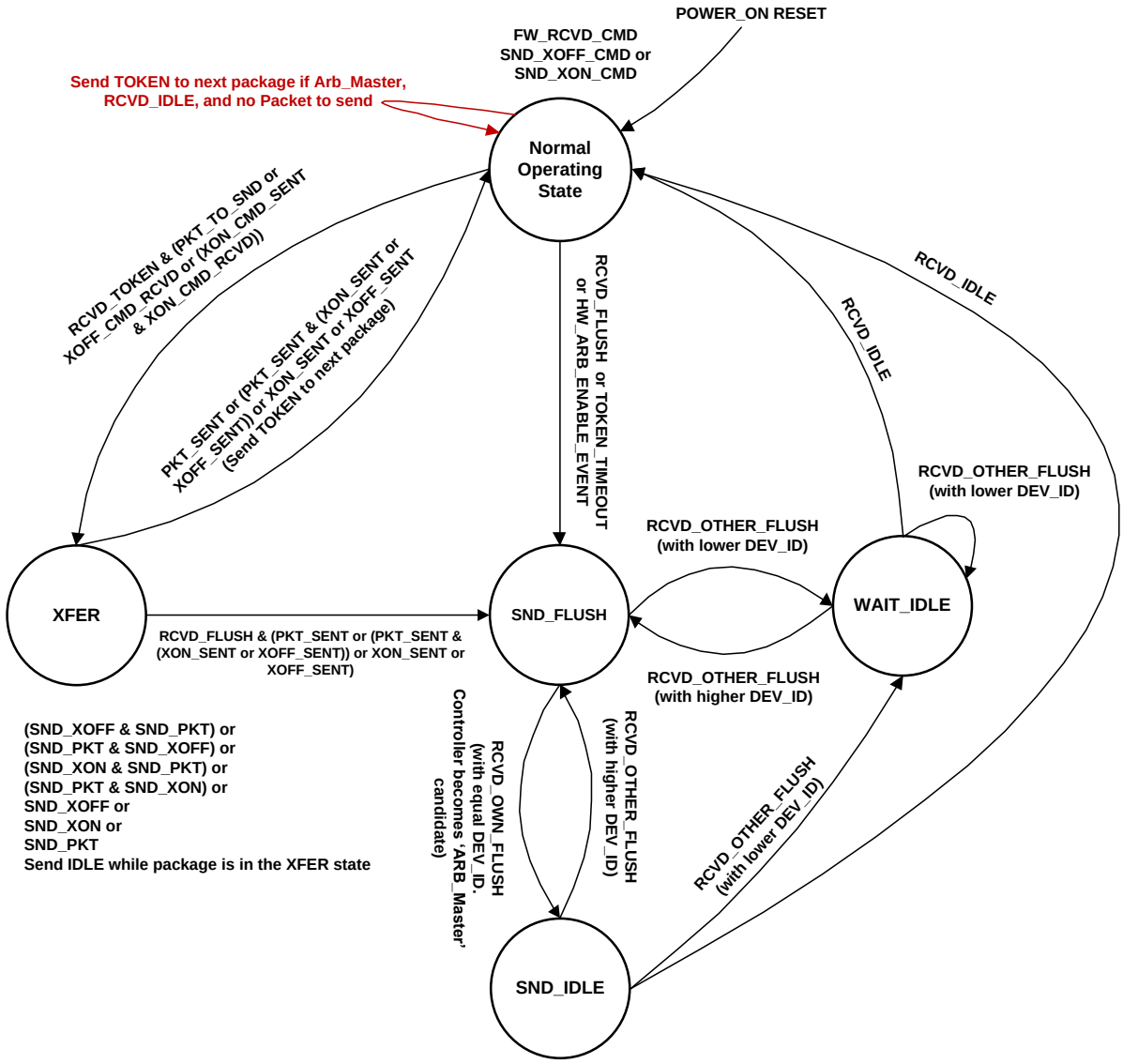


Figure 14 – Hardware arbitration state machine

1803 The states and events shown in Figure 14 are described in Table 6 and Table 7, respectively.

1804 **Table 6 – Hardware arbitration states**

State	Action
Normal Operating State	This state is the normal operating state for hardware arbitration. The following actions happen in this state: • FW_RCVD_CMD: Forward received command. As op-codes are received and acted upon, the resulting op-code is sent to the next package. For example, the TOKEN op-code is received and no packet data is available to send, so the TOKEN op-code is sent to the next package in the ring. • SND_XOFF_CMD: Send the XOFF op-code to the next package. This action happens when the specific conditions are met as described in 7.2.3. • SND_XON_CMD: Send the XON op-code to the next package. This action happens when the specific conditions are met as described in 7.2.3. • If the Network Controller is ARB_Master, it generates the TOKEN op-code upon receiving an IDLE op-code at the end of the FLUSH process. • The RXD lines will be in a high-impedance condition in this state.
XFER	In this state, data is sent on the RXD lines. This data will be a Pass-through packet, response packet, XON (Pause Off) packet, XOFF (Pause On) packet, or AEN. (An XON or XOFF packet can be sent in addition to a Pass-through packet, response packet, or AEN.) IDLE op-codes are sent to the next package while the device is in the XFER state. The following actions happen in this state: • SND_XON: Transmit an XON frame (Pause Off) to the Management Controller. • SND_XOFF: Transmit an XOFF frame (Pause On) to the Management Controller. • SND_PKT: Transmit a Pass-through packet, response packet, or AEN to the Management Controller. • The TOKEN op-code is sent to the next package upon completion of the transfer.
SND_FLUSH	This state is the entry point for determining the ARB_Master among the packages. In this state, the FLUSH op-code is continuously sent. This state is exited upon receiving a FLUSH op-code that has a DEV_ID that is equal to or lower than the package's own DEV_ID.
SND_IDLE	This is the final state for determining the ARB_Master, entered when a device's own FLUSH op-code is received. In this state, the IDLE op-code is continuously sent.
WAIT_IDLE	This state is entered when a FLUSH command is received from another package with a lower Device ID. When an IDLE op-code is received, the ARB_Master has been determined and the device transitions to the Normal Operating State.

1805 **Table 7 – Hardware arbitration events**

Event	Description
RCVD_TOKEN	A TOKEN op-code was received or the arbitration was just completed and won by this package.
RCVD_IDLE	An IDLE op-code was received.
XOFF_SENT	The Pause On frame was sent on the RXD interface.
XON_SENT	The Pause Off frame was sent on the RXD interface.
PKT_TO_SND	The Network Controller package has a Pass-through packet, command response packet, XON (Pause Off) frame, XOFF (Pause On) frame, or AEN to send.
XON_CMD_RCVD	A package received an XON op-code with its own Package ID.
XOFF_CMD_RCVD	An XOFF op-code was received.
XON_CMD_SENT	A package sent an XON op-code with its own Package ID.
RCVD_FLUSH	A FLUSH op-code was received.
TOKEN_TIMEOUT	The timeout limit expired while waiting for a TOKEN op-code.
HW_ARB_ENABLE_EVENT	This event begins ARB_MSTR assignment. This event occurs just after the Network Controller package initializes or when hardware arbitration is re-enabled through the Select Package command.
RCVD_OTHER_FLUSH	A package received a FLUSH op-code with a Package ID other than its own.
RCVD_OWN_FLUSH	A package received a FLUSH op-code with a Package ID equal to its own.

1806 ## 7.3 Command-based arbitration

1807 If hardware arbitration is not being used, the **Select Package** and **Deselect Package** commands shall be
1808 used to control which Network Controller package has the ability to transmit on the RXD lines. Because
1809 only one Network Controller package is allowed to transmit on the RXD lines, the Management Controller
1810 shall only have one package in the selected state at any given time. For more information, see 8.4.5 and
1811 8.4.7.

1812 # 8 Packet definitions

1813 This clause presents the formats of NC-SI packets and their relationship to frames used to transmit and
1814 receive those packets on RBT interface.

1815 ## 8.1 NC-SI packet encapsulation

1816 The RBT interface is an Ethernet interface adhering to the standard IEEE 802.3 Ethernet frame format.
1817 Whether or not the Network Controller accepts runt packets is unspecified.

1818 As shown in Figure 15, this L2, or data link layer, frame format encapsulates all NC-SI packets, including
1819 Pass-through, command, and response packets, as the L2 frame payload data by adding a 14-byte
1820 header to the front of the data and appending a 4-byte Frame Check Sequence (FCS) to the end.

1821 NC-SI Control Packets shall not include any VLAN tags. NC-SI Pass-through may include 802.1Q VLAN
1822 tag.

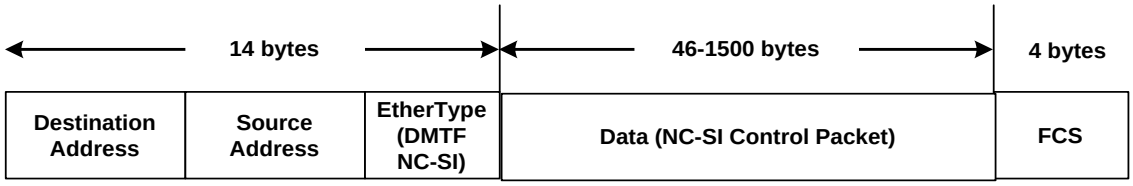


Figure 15 – Ethernet frame encapsulation of NC-SI packet data without VLAN tag

8.1.1 Ethernet frame header

The Management Controller shall format the 14-byte Ethernet frame header so that when it is received, it shall be formatted in the big-endian byte order shown in Table 8.

Channels shall accept Pass-through packets that meet the IEEE 802.3 frame requirements.

Table 8 – Ethernet Header Format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	DA_5= 0xFF	DA_4= 0xFF	DA_3= 0xFF	DA_2= 0xFF
04..07	DA_1= 0xFF	DA_0= 0xFF	SA_5	SA_4
08..11	SA_3	SA_2	SA_1	SA_0
12..13	EtherType = 0x88F8 (DMTF NC-SI)			

8.1.1.1 Destination Address (DA)

Bytes 0–5 of the header represent bytes 5–0 of the Ethernet Destination Address field of an L2 header.

The channel is not assigned a specific MAC address and the contents of this field are not interpreted as a MAC address by the Management Controller or the Network Controller. However, the DA field in all NC-SI Control Packets shall be set to the broadcast address (FF:FF:FF:FF:FF:FF) for consistency.

If the Network Controller receives a Control Packet with a Destination Address other than FF:FF:FF:FF:FF:FF, the Network Controller may elect to accept the packet, drop it, or return a response packet with an error response/reason code.

8.1.1.2 Source Address (SA)

Bytes 6–11 of the header represent bytes 5–0 of the Ethernet Source Address field of the Ethernet header. The contents of this field may be set to any value. The Network Controller may use FF:FF:FF:FF:FF:FF as the source address for NC-SI Control Packets that it generates.

8.1.1.3 EtherType

The final two bytes of the header, bytes 12..13, represent bytes 1..0 of the EtherType field of the Ethernet header. For NC-SI Control Packets, this field shall be set to a fixed value of 0x88F8 as assigned to the NC-SI by the IEEE. This value allows NC-SI Control Packets to be differentiated from other packets in the overall packet stream.

8.1.2 Frame Check Sequence

The Frame Check Sequence (FCS) shall be added at the end of the frame to provide detection of corruption of the frame. Any frame with an invalid FCS shall be discarded.

8.1.3 Data length

NC-SI Commands, Responses, and AENs do not carry any VLAN tag. NC-SI Commands, Responses and AENs shall have a payload data length between 46 and 1500 octets (bytes). This is in compliance with the 802.3 specification. This means that the length of Ethernet frame shown in Figure 15 is between 64 octets (for a payload of 46 octets) and 1518 octets (for a payload with 1500 octets).

Pass-through packets also follow the 802.3 specification. The maximum payload size is 1500 octets; the minimum payload size shall be 42 octets when 802.1Q (VLAN) tag is present and 46 octets when the 802.1Q tag is not present. The Layer-2 Ethernet frame for a 802.1Q tagged frame shall be between 64 octets (for a payload of 42 octets) and 1522 octets (for a payload with 1500 octets). For Pass-through packets that are not 802.1Q tagged, the minimum Layer-2 Ethernet frame size is 64 octets (for a payload of 46 octets) and the maximum Layer-2 Ethernet frame size is 1518 octets (for a payload with 1500 octets).

8.2 Control Packet data structure

Each NC-SI Control Packet is made up of a 16-byte packet header and a payload section whose length is specific to the packet type.

8.2.1 Control Packet header

The 16-byte Control Packet header is used in command, response, and AEN packets, and contains data values intended to allow the packet to be identified, validated, and processed. The packet header is in big-endian byte order, as shown in Table 9.

Table 9 – Control Packet header format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	MC ID	Header Revision	Reserved	IID
04..07	Control Packet Type	Ch. ID	Flags	Payload Length
08..11	Reserved			
12..15	Reserved			

8.2.1.1 Management Controller ID

In Control Packets, this 1-byte field identifies the Management Controller issuing the packet. For this version of the specification, Management Controllers should set this field to `0x00` (zero). This implies that only one management controller is supported for accessing the NC via NC-SI at any given time, Network Controllers responding to command packets should copy the Management Controller ID field from the command packet header into the response packet header. For AEN packets, this field should be copied from the parameter that was set using the AEN Enable command.

8.2.1.2 Header revision

This 1-byte field identifies the version of the Control Packet header in use by the sender. For this version of the specification, the header revision is `0x01`.

8.2.1.3 Instance ID (IID)

This 1-byte field contains the IID of the command and associated response. The Network Controller can use it to differentiate retried commands from new instances of commands. The Management Controller can use this value to match a received response to the previously sent command. For more information, see 6.3.1.1.

8.2.1.4 Control Packet type

This 1-byte field contains the Identifier that is used to identify specific commands and responses, and to differentiate AENs from responses. Each NC-SI command is assigned a unique 7-bit command type value in the range `0x00..0x60`. The proper response type for each command type is formed by setting the most significant bit (bit 7) in the original 1-byte command value. This allows for a one-to-one correspondence between 96 unique response types and 96 unique command types.

8.2.1.5 Channel ID

This 1-byte field contains the Network Controller Channel Identifier. The Management Controller shall set this value to specify the package and internal channel ID for which the command is intended.

In a multi-drop configuration, all commands are received by all NC-SI Network Controllers present in the configuration. The Channel ID is used by each receiving Network Controller to determine if it is the intended recipient of the command. In Responses and AENs, this field carries the ID of the channel from which the response of AEN was issued.

8.2.1.6 Payload length

This 12-bit field contains the length, in bytes, of any payload data present in the command or response frame following the NC-SI packet header. This value does not include the length of the NC-SI Control Packet Header, the checksum value, or any padding that might be present.

8.2.1.7 Flags

Bit 0: Poll Indication: If this bit is set, it indicates that this command instance is polling on a previously sent command that was responded with a “Delayed Response” response code. This bit is relevant only for commands and not for responses or AENs.

Bits 7:1: Reserved

8.2.1.8 Reserved

These fields are reserved for future use and should be written as zeros and ignored when read.

8.2.2 Control Packet payload

The NC-SI packet payload may contain zero or more defined data values depending on whether the packet is a command or response packet, and on the specific type. The NC-SI packet payload is always formatted in big-endian byte order, as shown in Table 10.

1915 **Table 10 – Generic example of Control Packet payload**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	$Data0_3$	$Data0_2$	$Data0_1$	$Data0_0$
04..07	$Data1_7$	$Data1_6$	$Data1_5$	$Data1_4$
08..11	$Data1_3$	$Data1_2$	$Data1_1$	$Data1_0$
..				
…	$DataN\text{-}1_4$	$DataN\text{-}1_3$	$DataN\text{-}1_2$	$DataN\text{-}1_1$
…	$DataN\text{-}1_0$	Payload Pad (as required)		
…	2s Complement Checksum Compensation			
…	Ethernet Packet Pad (as required)			

1916 ### 8.2.2.1 Data

1917 As shown in Table 10, the bytes following the NC-SI packet header may contain payload data fields of
1918 varying sizes, and which may be aligned or require padding. In the case where data is defined in the
1919 payload, all data-field byte layouts (Data0–Data-1) shall use big-endian byte ordering with the most
1920 significant byte of the field in the lowest addressed byte position (that is, coming first).

1921 ### 8.2.2.2 Payload pad

1922 If the payload is present and does not end on a 32-bit boundary, one to three padding bytes equal to
1923 `0x00` shall be present to align the checksum field to a 32-bit boundary.

1924 ### 8.2.2.3 2’s Complement checksum compensation

1925 This 4-byte field contains the 32-bit checksum compensation value that may be included in each
1926 command and response packet by the sender of the packet. When it is implemented, the checksum
1927 compensation shall be computed as the 2’s complement of the checksum, which shall be computed as
1928 the 32-bit unsigned sum of the NC-SI packet header and NC-SI packet payload interpreted as a series of
1929 16-bit unsigned integer values. A packet receiver supporting packet checksum verification shall use the
1930 checksum compensation value to verify packet data integrity by computing the 32-bit checksum described
1931 above, adding to it the checksum compensation value from the packet, and verifying that the result is 0.

1932 Verification of non-zero NC-SI packet checksum values is optional. An implementation may elect to
1933 generate the checksums and may elect to verify checksums that it receives. The checksum field is
1934 generated and handled according to the following rules:

1935 • A checksum field value of all zeros specifies that a header checksum is not being provided for
1936 the NC-SI Control Packet, and that the checksum field value shall be ignored when processing
1937 the packet.

1938 • If the originator of an NC-SI Control Packet is not generating a checksum, the originator shall
1939 use a value of all zeros for the header checksum field.

1940 • If a non-zero checksum field is generated for an NC-SI Control Packet, that header checksum
1941 field value shall be calculated using the specified algorithm.

1942 • All receivers of NC-SI Control Packets shall accept packets with all zeros as the checksum
1943 value (provided that other fields and the CRC are correct).

- The receiver of an NC-SI Control Packet may reject (silently discard) a packet that has an incorrect non-zero checksum.
- The receiver of an NC-SI Control Packet may ignore any non-zero checksums that it receives and accept the packet, even if the checksum value is incorrect (that is, an implementation is not required to verify the checksum field).
- A controller that generates checksums is not required to verify checksums that it receives.
- A controller that verifies checksums is not required to generate checksums for NC-SI Control Packets that it originates.

8.2.2.4 Ethernet packet pad

Per IEEE 802.3, all Ethernet frames shall be at least 64 bytes in length, from the DA through and including FCS. For NC-SI packets, this requirement applies to the Ethernet header and payload, which includes the NC-SI Control Packet header and payload. Most NC-SI Control Packets are less than the minimum Ethernet frame payload size of 46 bytes in length and require padding to comply with IEEE 802.3.

8.2.3 Command packet Payload

Command packets have no common fixed payload format.

8.2.4 Response packet payload

Unlike command packets that do not necessarily contain payload data, all response packets carry at least a 4-byte payload. This default payload carries the response codes and reason codes (described in 8.2.4.1) that provide status on the outcome of processing the originating command packet, and is present in all response packet payload definitions.

The default payload occupies bytes `00..03` of the response packet payload, with any additional response-packet-specific payload defined to follow starting on the next word. All response packet payload fields are defined with big-endian byte ordering, as shown in Table 11.

Table 11 – Generic example of Response packet payload format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	Response Code		Reason Code	
..	...	...	...	...
…	DataN-1_4	DataN-1_3	DataN-1_2	DataN-1_1
…	DataN-1_0	Word Pad (as required)		
…	2s Complement Checksum Compensation			
…	Ethernet Packet Pad (as required)			

8.2.4.1 Response Packet in case of Delayed Response Code

If a response includes a “Delayed Response” Code, then the response does not contain the payload of the original response, The Delayed Response shall contain a payload of a single word (uint16) including the recommended next polling time in milliseconds. If no polling time estimate is available, then the recommended next polling time shall be set to `0x0000`.

1974 **Table 12 – Generic example of Delayed Response packet payload**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	Response Code = 0x004		Reason Code = 0x0000	
04..07	Reserved		Next Polling time	
08…11	2s Complement Checksum Compensation			
…	Ethernet Packet Pad (as required)			

1975

1976 ### 8.2.5 Response codes and reason codes

1977 Response codes and reason codes are status values that are returned in the responses to NC-SI
1978 commands. The response code values provide a general categorization of the status being returned. The
1979 reason code values provide additional detail related to a particular response code.

1980 #### 8.2.5.1 General

1981 Response codes and reason codes are divided into numeric ranges that distinguish whether the values
1982 represent standard codes that are defined in this specification or are vendor/OEM-specific values that are
1983 defined by the vendor of the controller.

1984 The response code is a 2-byte field where values from `0x00` through `0x7F` are reserved for definition by
1985 this specification. Values from `0x80` through `0xFF` are vendor/OEM-specific codes that are defined by the
1986 vendor of the controller.

1987 The reason code is a 2-byte field. The ranges of values are defined in Table 13.

1988 **Table 13 – Reason code ranges**

MS-byte	LS-byte	Description
`00h`	`0x00–0x7F`	Standard generic reason codes This range of values for the lower byte is used for reason codes that are not specific to a particular command but can be used as reason codes in responses for any command. The values in this range are reserved for definition by this specification.
	`0x80–0xFF`	Vendor/OEM generic reason codes This range of values for the lower byte is used for reason codes that are not specific to a particular command but can be used as reason codes in responses for any command. Values in this range are defined by the vendor of the controller.
Command Number Note: This means that Command	`0x00–0x7F`	Standard command-specific reason codes This range of values for the lower byte is used for reason codes that are specific to a particular command. The upper byte holds the value of the command for which the reason code is defined. The values in this range are reserved for definition by this specification.

MS-byte	LS-byte	Description
Number 00 cannot have any command-specific reason codes.	0x80–0xFF	Vendor/OEM command-specific reason codes This range of values for the lower byte is used for reason codes that are specific to a particular command. The upper byte holds the value of the command for which the reason code is defined. Values in this range are defined by the vendor of the controller.

8.2.5.2 Response code and reason code values

The standard response code values are defined in Table 14, and the standard reason code values are defined in Table 15. Command-specific values, if any, are defined in the clauses that describe the response data for the command. Unless otherwise specified, the standard reason codes may be used in combination with any response code. There are scenarios where multiple combinations of response and reason code values are valid. Unless otherwise specified, an implementation may return any valid combination of response and reason code values for the condition.

Table 14 – Standard response code values

Value	Description	Comment
0x0000	Command Completed	Returned for a successful command completion. When this response code is returned, the reason code shall be 0x0000 as described in Table 15 unless a more informative reason code is appropriate such as 0x0E08 meaning MAC Address is zero .
0x0001	Command Failed	Returned to report that a valid command could not be processed or failed to complete correctly
0x0002	Command Unavailable	Returned to report that a command is temporarily unavailable for execution because the controller is in a transient state, busy condition, or in need of external intervention.
0x0003	Command Unsupported	Returned to report that a command is not supported by the implementation. The reason code “Unknown / Unsupported Command Type should be returned along with this response code for all unsupported commands.
0x0004	Delayed Response	Returned to report that the command was accepted, and the NC started to handle it, but it cannot respond within T5 seconds with a final answer. When this response code is provided, the reason code shall be 0x0000
0x8000–0xFFFF	Vendor/OEM-specific	Response codes defined by the vendor of the controller

Table 15 – Standard Reason Code Values

Value	Description	Comment
0x0000	No Error/No Reason Code	When used with the Command Completed response code, indicates that the command completed normally. Otherwise this value indicates that no additional reason code information is being provided.
0x0001	Interface Initialization Required	Returned for all commands except Select/Deselect Package commands when the channel is in the Initial State, until the channel receives a Clear Initial State command

Value	Description	Comment
0x0002	Parameter Is Invalid, Unsupported, or Out-of-Range	Returned when a received parameter value is outside of the acceptable values for that parameter
0x0003	Channel Not Ready	Returned when the channel is in a transient state in which it is unable to process commands normally
0x0004	Package Not Ready	Returned when the package and channels within the package are in a transient state in which normal command processing cannot be done
0x0005	Invalid payload length	Returned when the payload length in the command is incorrect for the given command
0x0006	Information not available	Returned when the channel is unable to provide response data to a valid supported command.
0x0007	Intervention Required	May be returned for all commands, except for Select and Deselect Package, when the Package is not ready and requires intervention to restore its operational state. When this code is returned, the NC does not check if the command is otherwise valid and the defined response is not returned.
0x7FFF	Unknown / Unsupported Command Type	Returned when the command type is unknown or unsupported. This reason code shall only be used when the response code is 0x0003 (Command Unsupported) as described in Table 14.
0x8000–0xFFFF	OEM Reason Code	Vendor-specific reason code defined by the vendor of the controller

8.2.6 AEN packet format

AEN packets shall follow the general packet format of Control Packets, with the IID field set to 0 because, by definition, the Management Controller does not send a response packet to acknowledge an AEN packet. The Control Packet Type field shall have the value 0xFF. The originating Network Controller shall fill in the Channel ID (Ch. ID) field with its own ID to identify itself as the source of notification. Currently, three AEN types are defined in the AEN Type field. Table 16 represents the general AEN packet format.

Table 16 – AEN packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	MC ID = 0x0	0x01	Reserved	IID = 0x0
04..07	Control Packet Type = 0xFF	Originating Ch. ID	Reserved	Payload Length
08..11	Reserved			
12..15	Reserved			
16..19	Reserved			AEN Type
20..23	OPTIONAL AEN Data			
24..27	Checksum			

8.2.7 AEN packet data structure

The AEN type field (8-bit) has the values shown in Table 17.

Table 17 – AEN types

Value	AEN Type
0x0	Link Status Change
0x1	Configuration Required
0x2	Host NC Driver Status Change
0x3	Delayed Response Ready
0x4..0x6F	Reserved
0x70..0x7F	Transport-specific AENs
0x80..0xFF	OEM-specific AENs

8.2.8 OEM AEN packet format

OEM AEN packets shall conform to the format below……

Table 18 – OEM AEN packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	MC ID = 0x0	0x01	Reserved	IID = 0x0
04..07	Control Packet Type = 0xFF	Originating Ch. ID	Reserved	Payload Length
08..11	Reserved			
12..15	Reserved			
16..19	Reserved			AEN Type
20..23	Manufacturer ID (IANA)			
24..27	OPTIONAL AEN Data			
28..31	Checksum			

8.3 Control Packet type definitions

Command packet types are in the range of 0x00 to 0x7F. **Error! Reference source not found.** describes each command, its corresponding response, and the type value for each. **Error! Reference source not found.** includes commands addressed to either a package or a channel. The commands addressed to a package are highlighted with gray background. PLDM and OEM-specific commands carried over NC-SI may be package specific or channel specific or both.

Mandatory (M), Optional (O), and Conditional (C) refer to command support requirements for the Network Controller.

Table 19 – Command and response types

2020

Command Type	Command Name	Description	Response Type	Cmd/Fabric Support Requirement		
				E	IB	FC
0x00	Clear Initial State	Used by the Management Controller to acknowledge that the Network Controller is in the Initial State	0x80	M	M	M
0x01	Select Package	Used to explicitly select a controller package to transmit packets through the NC-SI interface	0x81	M	M	M
0x02	Deselect Package	Used to explicitly instruct the controller package to stop transmitting packets through the NC-SI interface	0x82	M	M	M
0x03	Enable Channel	Used to enable the NC-SI channel and to cause the forwarding of bidirectional Management Controller packets to start	0x83	M	M	M
0x04	Disable Channel	Used to disable the NC-SI channel and to cause the forwarding of bidirectional Management Controller packets to cease	0x84	M	M	M
0x05	Reset Channel	Used to synchronously put the Network Controller back to the Initial State	0x85	M	M	M
0x06	Enable Channel Network TX	Used to explicitly enable the channel to transmit Pass-through packets onto the network	0x86	M	O	N/A
0x07	Disable Channel Network TX	Used to explicitly disable the channel from transmitting Pass-through packets onto the network	0x87	M	O	N/A
0x08	AEN Enable	Used to control generating AENs	0x88	C	C	C
0x09	Set Link	Used during OS absence to force link settings, or to return to auto-negotiation mode	0x89	M	N/A	N/A
0x0A	Get Link Status	Used to get current link status information	0x8A	M	N/A	N/A
0x0B	Set VLAN Filter	Used to program VLAN IDs for VLAN filtering	0x8B	M	O	N/A
0x0C	Enable VLAN	Used to enable VLAN filtering of Management Controller RX packets	0x8C	M	O	N/A

0x0D	Disable VLAN	Used to disable VLAN filtering	0x8D	M	O	N/A
0x0E	Set MAC Address	Used to configure and enable unicast and multicast MAC address filters	0x8E	M	O	N/A
0x10	Enable Broadcast Filter	Used to enable selective broadcast packet filtering	0x90	M	O	N/A
0x11	Disable Broadcast Filter	Used to disable all broadcast packet filtering, and to enable the forwarding of all broadcast packets	0x91	M	O	N/A
0x12	Enable Global Multicast Filter	Used to enable selective multicast packet filtering	0x92	C	O	N/A
0x13	Disable Global Multicast Filter	Used to disable all multicast packet filtering, and to enable forwarding of all multicast packets	0x93	C	O	N/A
0x14	Set NC-SI Flow Control	Used to configure IEEE 802.3 flow control on RBT	0x94	O	O	N/A
0x15	Get Version ID	Used to get controller-related version information	0x95	M	M	M
0x16	Get Capabilities	Used to get optional functions supported by the NC-SI	0x96	M		
0x17	Get Parameters	Used to get configuration parameter values currently in effect on the controller	0x97	M		
0x18	Get Controller Packet Statistics	Used to get current packet statistics for the Ethernet Controller	0x98	O		
0x19	Get NC-SI Statistics	Used to request the packet statistics specific to the NC-SI	0x99	O		
0x1A	Get NC-SI Pass-through Statistics	Used to request NC-SI Pass-through packet statistics	0x9A	O		
0x1B	Get Package Status	Used to get current status of the package.	0x9B	O		
x						

2021

2022

Command Type	Command Name	Description	Response Type	Command Support Requirement	
0x13	Disable Global Multicast Filter	Used to disable all multicast packet filtering, and to enable forwarding of all multicast packets	0x93	C	
0x14	Set NC-SI Flow Control	Used to configure IEEE 802.3 flow control on the NC-SI	0x94	O	
0x15	Get Version ID	Used to get controller-related version information	0x95	M	
0x16	Get Capabilities	Used to get optional functions supported by the NC-SI	0x96	M	
0x17	Get Parameters	Used to get configuration parameter values currently in effect on the controller	0x97	M	
0x18	Get Controller Packet Statistics	Used to get current packet statistics for the Ethernet Controller	0x98	O	
0x19	Get NC-SI Statistics	Used to request the packet statistics specific to the NC-SI	0x99	O	
0x1A	Get NC-SI Pass-through Statistics	Used to request NC-SI Pass-through packet statistics	0x9A	O	
0x1B	Get Package Status	Used to get current status of the package.	0x9B	O	
New	Get PF Assignment	Used to request Function assignment information			
New	Set PF Assignment	Used to configure and enable Functions			
New	Get NC Capabilities and Settings	Used to request device configuration information and capabilities			
New	Set NC Configuration	Used to configure device interfaces			
New	Get iSCSI Offload Statistics	Used to get iSCSI Offload packet statistics			
New	Get Port Configuration	Used to request port configuration information			
New	Set Port Configuration	Used to configure operational characteristics of the port			
New	Get ASIC Temperature	Used to request silicon temperature information			
New	Get Ambient Temperature	Used to request ambient temperature information			
New	Get SFF Module Temp	Used to request termperature information from the attached SFF transceiver			
New					

Command Type	Command Name	Description	Response Type	Command Support Requirement	
New					
New	Get Partition Configuration	Used to request partition configuration information			
New	Set Partition Configuration	Used to configure partition operational characteristics			
New	Get Boot Config	Used to request boot protocol configuration information			
New	Set Boot Config	Used to configure			
0x50	OEM Command	Used to request vendor-specific data	0xD0	O	
0x51	PLDM	Used for PLDM request over NC-SI over RBT	0xD1	O	
0x52	Get Package UUID	Returns a universally unique identifier (UUID) for the package	0xD2	O	
0x51–0x60	Reserved for Transport Protocol Oriented Commands	Used to define transport protocol oriented commands (e.g., PLDM over NC-SI/RBT)	0xD1–0xE0	O	
0x51	Reserved				
0x52	Get Package UUID	Returns a universally unique identifier (UUID) for the package	0xD2	O	
0x53	PLDM	Used for PLDM request over NC-SI over RBT	0xD3	O	
0x54	Get Supported Media	See MCTP DSP0261 for full definition This command may be used on any transport	0xD4		
0x55	Transport Specific AEN Enable	See MCTP DSP0261 for full definition			
New	Get FC Link Status	Used to request link information from the			
New	Get FC Statistics				
New	Get InfiniBand Link Status				
New	Get IB Statistics				
New	Get MC MAC Address				
Key: M = Mandatory (required) O = Optional C = Conditional (see command description)					

8.4 Command and response packet formats

This clause describes the format for each of the NC-SI commands and corresponding responses.

The corresponding response packet format shall be mandatory when a given command is supported.

8.4.1 NC-SI command frame format

Table 20 illustrates the NC-SI frame format that shall be accepted by the Network Controller.

Table 20 – Example of complete minimum-sized NC-SI command packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	0xFF	0xFF	0xFF	0xFF
04..07	0xFF	0xFF	0xXX	0xXX
08..11	0xXX	0xXX	0xXX	0xXX
12..15	0x88F8		MC ID	Header Revision
16..19	Reserved	IID	Command Type	Ch. ID
20..23	Reserved	Payload Length	Reserved	
24..27	Reserved		Reserved	
28..31	Reserved		Checksum (3..2)	
32..35	Checksum (1..0)		Pad	
36..39	Pad			
40..43	Pad			
44..47	Pad			
48..51	Pad			
52..55	Pad			
56..59	Pad			
60..63	FCS			

8.4.2 NC-SI response packet format

Table 21 illustrates the NC-SI response packet format that shall be transmitted by the Network Controller.

Table 21 – Example of complete minimum-sized NC-SI response packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..03	0xFF	0xFF	0xFF	0xFF
04..07	0xFF	0xFF	0xFF	0xFF
08..11	0xFF	0xFF	0xFF	0xFF
12..15	0x88F8		MC ID	Header Revision

16..19	Reserved	IID	Response Type	Ch. ID
20..23	Reserved	Payload Length	Reserved	
24..27	Reserved		Reserved	
28..31	Reserved		Response Code	
32..35	Reason Code		Checksum (3..2)	
36..39	Checksum (1..0)		Pad	
40..43	Pad			
44..47	Pad			
48..51	Pad			
52..55	Pad			
56..59	Pad			
60..63	FCS			

8.4.3 Clear Initial State command (0x00)

The Clear Initial State command provides the mechanism for the Management Controller to acknowledge that it considers a channel to be in the Initial State (typically because the Management Controller received an “Interface Initialization Required” reason code) and to direct the Network Controller to start accepting commands for initializing or recovering the NC-SI operation. When in the Initial State, the Network Controller shall return the “Interface Initialization Required” reason code for all channel commands until it receives the Clear Initial State command.

If the channel is in the Initial State when it receives the Clear Initial State command, the command shall cause the Network Controller to stop returning the “Interface Initialization Required” reason code. The channel shall also treat any subsequently received instance ID numbers as IDs for new command instances, not retries.

If the channel is not in the Initial State when it receives this command, it shall treat any subsequently received instance ID numbers as IDs for new command instances, not retries.

Table 22 illustrates the packet format of the Clear Initial State command.

Table 22 – Clear Initial State command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.4 Clear Initial State response (0x80)

Currently no command-specific reason code is identified for this response (see Table 23).

2049 **Table 23 – Clear Initial State response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

2050 ### 8.4.5 Select Package command (0x01)

2051 A package is considered to be “selected” when its NC-SI output buffers are allowed to transmit packets
2052 through the NC-SI interface. Conversely, a package is “deselected” when it is not allowed to transmit
2053 packets through the NC-SI interface.

2054 The Select Package command provides a way for a Management Controller to explicitly take a package
2055 out of the deselected state and to control whether hardware arbitration is enabled for the package.
2056 (Similarly, the Deselect Package command allows a Management Controller to explicitly deselect a
2057 package.)

2058 The NC-SI package in the Network Controller shall also become selected if the package receives any
2059 other NC-SI command that is directed to the package or to a channel within the package.

2060 The Select Package command is addressed to the package, rather than to a particular channel (that is,
2061 the command is sent with a Channel ID where the Package ID subfield matches the ID of the intended
2062 package and the Internal Channel ID subfield is set to 0x1F).

2063 More than one package can be in the selected state simultaneously if hardware arbitration is used
2064 between the selected packages and is active. The hardware arbitration logic ensures that buffer conflicts
2065 will not occur between selected packages.

2066 If hardware arbitration is not active or is not used for a given package, only one package shall be selected
2067 at a time. To switch between packages, the Deselect Package command is used by the Management
2068 Controller to put the presently selected package into the deselected state before another package is
2069 selected.

2070

2071 A package shall stay in the selected state until it receives a Deselect Package command, unless an
2072 internal condition causes all internal channels to enter the Initial State.

2073 A package that is not using hardware arbitration may leave its output buffers enabled for the time that it is
2074 selected, or it may place its output buffers into the high-impedance state between transmitting packets
2075 through the NC-SI interface. (Temporarily placing the output buffers into the high-impedance state is not
2076 the same as entering the deselected state.)

2077 For Type A integrated controllers: Because the bus buffers are separately controlled, a separate Select
2078 Package command needs to be sent to each Package ID in the controller that is to be enabled to transmit
2079 through the NC-SI interface. If the internal packages do not support hardware arbitration, only one
2080 package shall be selected at a time; otherwise, a bus conflict will occur.

2081 For Type S single channel, and Types B and C integrated controllers: A single set of bus buffers exists for
2082 the package. Sending a Select Package command selects the entire package and enables all channels
2083 within the package to transmit through the NC-SI interface. (Whether a particular channel in a selected

2084 package starts transmitting Pass-through and AEN packets depends on whether that channel was
2085 enabled or disabled using the Enable or Disable Channel commands and whether the package may have
2086 had packets queued up for transmission.)

2087 Table 24 illustrates the packet format of the Select Package command. Table 25 illustrates the disable
2088 byte for hardware arbitration.

2089 **Table 24 – Select Package command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Reserved			Features Control
20..23	Checksum			
24..45	Pad			

2090 **Table 25 – Features Control byte**

Bits	Description
0	`0b` = Hardware arbitration between packages is enabled. `1b` = Disable hardware arbitration. Disabling hardware arbitration causes the package's arbitration logic to enter or remain in bypass mode. In the case that the Network Controller does not support hardware arbitration, this bit is ignored; the Network Controller shall not return an error if the Select Package command can otherwise be successfully processed.
1	Delayed Response Enable: `0b` = NC is not allowed to use the "Delayed Response" response code `1b` = NC is allowed to use the "Delayed Response" response code
7..2	Reserved

2091 ### 8.4.6 Select package response (`0x81`)

2092 Currently no command-specific reason code is identified for this response (see Table 26).

2093 **Table 26 – Select package response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.7 Deselect Package command (0x02)

The Deselect Package command directs the controller package to stop transmitting packets through the NC-SI interface and to place the output buffers for the package into the high-impedance state.

The Deselect Package command is addressed to the package, rather than to a particular channel (that is, the command is sent with a Channel ID where the Package ID subfield matches the ID of the intended package and the Internal Channel ID subfield is set to 0x1F).

The controller package enters the deselected state after it has transmitted the response to the Deselect Package command and placed its buffers into the high-impedance state. The controller shall place its outputs into the high-impedance state within the Package Deselect to Hi-Z Interval (T1). (This interval gives the controller being deselected time to turn off its electrical output buffers after sending the response to the Deselect Package command.)

If hardware arbitration is not supported or used, the Management Controller should wait for the Package Deselect to Hi-Z Interval (T1) to expire before selecting another controller.

For Type A integrated controllers: Because the bus buffers are separately controlled, putting the overall controller package into the high-impedance state requires sending separate Deselect Package commands to each Package ID in the overall package.

For Type S single channel, and Types B and C integrated controllers: A single set of bus buffers exists for the package. Sending a Deselect Package command deselects the entire NC-SI package and prevents all channels within the package from transmitting through the NC-SI interface.

Table 27 illustrates the packet format of the Deselect Package command.

Table 27 – Deselect Package command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.8 Deselect Package response (0x82)

The Network Controller shall always put the package into the deselected state after sending a Deselect Package Response.

No command-specific reason code is identified for this response (see Table 28).

Table 28 – Deselect Package response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.9 Enable Channel command (0x03)

The Enable Channel command shall enable the Network Controller to allow transmission of Pass-through and AEN packets to the Management Controller through the NC-SI.

Table 29 illustrates the packet format of the Enable Channel command.

Table 29 – Enable Channel command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.10 Enable Channel response (0x83)

No command-specific reason code is identified for this response (see Table 30).

Table 30 – Enable Channel response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.11 Disable Channel command (0x04)

The Disable Channel command allows the Management Controller to disable the flow of packets, including Pass-through and AEN, to the Management Controller.

A Network Controller implementation is not required to flush pending packets from its RX Queues when a channel becomes disabled. If queuing is subsequently disabled for a channel, it is possible that a number of packets from the disabled channel could still be pending in the RX Queues. These packets may continue to be transmitted through the NC-SI interface until the RX Queues are emptied of those packets. The Management Controller should be aware that it may receive a number of packets from the channel before receiving the response to the Disable Channel command.

The 1-bit Allow Link Down (ALD) field can be used by the Management Controller to indicate that the link corresponding to the specified channel is not required after the channel is disabled. The Network Controller is allowed to take down the external network physical link if no other functionality (for example, host OS or WoL [Wake-on-LAN]) is active.

Possible values for the 1-bit ALD field are as follows:

- 0b = Keep link up (establish and/or keep a link established) while channel is disabled
- 1b = Allow link to be taken down while channel is disabled

Table 31 illustrates the packet format of the Disable Channel command.

Table 31 – Disable Channel command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Reserved			ALD
20..23	Checksum			
24..45	Pad			

NOTE It is currently unspecified whether this command will cause the Network Controller to cease the pass through of traffic from the Management Controller to the network, or if this can only be done using the Disable Channel Network TX command.

8.4.12 Disable Channel response (0x84)

No command-specific reason code is identified for this response (see Table 32).

Table 32 – Disable Channel response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.13 Reset Channel command (0x05)

The Reset Channel command allows the Management Controller to put the channel into the Initial State. Packet transmission is not required to stop until the Reset Channel response has been sent. Thus, the Management Controller should be aware that it may receive a number of packets from the channel before receiving the response to the Reset Channel command.

Table 33 illustrates the packet format of the Reset Channel command.

Table 33 – Reset Channel command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Reserved			
20..23	Checksum			
24..45	Pad			

8.4.14 Reset Channel response (0x85)

Currently no command-specific reason code is identified for this response (see Table 34).

2161 **Table 34 – Reset Channel response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

2162 ### 8.4.15 Enable Channel Network TX command (0x06)

2163 The Enable Channel Network TX command shall enable the channel to transmit Pass-through packets
2164 onto the network. After network transmission is enabled, this setting shall remain enabled until a Disable
2165 Channel Network TX command is received, or the channel enters the Initial State.

2166 The intention of this command is to control which Network Controller ports are allowed to transmit to the
2167 external network. The Network Controller compares the source MAC address in outgoing Pass-through
2168 packets to the unicast MAC address(es) configured using the Set MAC Address command. If a match
2169 exists, the packet is transmitted to the network.

2170 Table 35 illustrates the packet format of the Enable Channel Network TX command.

2171 **Table 35 – Enable Channel Network TX command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

2172

2173 ### 8.4.16 Enable Channel Network TX response (0x86)

2174 No command-specific reason code is identified for this response (see Table 36).

2175 **Table 36 – Enable Channel Network TX response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.17 Disable Channel Network TX command (0x07)

The Disable Channel Network TX command disables the channel from transmitting Pass-through packets onto the network. After network transmission is disabled, it shall remain disabled until an Enable Channel Network TX command is received.

Table 37 illustrates the packet format of the Disable Channel Network TX command.

Table 37 – Disable Channel Network TX command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..23	Pad			

8.4.18 Disable Channel Network TX response (0x87)

The NC-SI shall, in the absence of a checksum error or identifier mismatch, always accept the Disable Channel Network TX command and send a response.

Currently no command-specific reason code is identified for this response (see Table 38).

Table 38 – Disable Channel Network TX response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.19 AEN Enable command (0x08)

Network Controller implementations shall support this command on the condition that the Network Controller generates one or more standard AENs. The AEN Enable command enables and disables the different standard AENs supported by the Network Controller. The Network Controller shall copy the AEN

MC ID field from the AEN Enable command into the MC ID field in every subsequent AEN sent to the Management Controller.

For more information, see 8.5 ("AEN packet formats") and 8.2.1.1 ("Management Controller ID").

Control of transport-specific AENs is outside the scope of this specification and should be defined by the particular transport binding specifications.

Table 39 illustrates the packet format of the AEN Enable command.

Table 39 – AEN Enable command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Reserved			AEN MC ID
20..23	AEN Control			
24..27	Checksum			
28..45	Pad			

The AEN Control field has the format shown in Table 40.

Table 40 – Format of AEN control

Bit Position	Field Description	Value Description
0	Link Status Change AEN control	0b = Disable Link Status Change AEN 1b = Enable Link Status Change AEN
1	Configuration Required AEN control	0b = Disable Configuration Required AEN 1b = Enable Configuration Required AEN
2	Host NC Driver Status Change AEN control	0b = Disable Host NC Driver Status Change AEN 1b = Enable Host NC Driver Status Change AEN
3	Delayed Response Ready AEN control	0b = Disable Delayed Response Ready AEN 1b = Enable Delayed Response Ready AEN
4	Transceiver Event AEN Control	0b = Disable Transceiver Event AEN 1b = Enable Transceiver Event AEN
15..5	Reserved	Reserved
31..16	OEM-specific AEN control	OEM-specific control

8.4.20 AEN Enable response (0x88)

Currently no command-specific reason code is identified for this response (see Table 41).

Table 41 – AEN Enable response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.21 Set Link command (0x09)

The Set Link command may be used by the Management Controller to configure the external network interface associated with the channel by using the provided settings. Upon receiving this command, while the host NC driver is not operational, the channel shall attempt to set the link to the configuration specified by the parameters. Upon successful completion of this command, link settings specified in the command should be used by the network controller as long as the host NC driver does not overwrite the link settings.

In the absence of an operational host NC driver, the NC should attempt to make the requested link state change even if it requires the NC to drop the current link. The channel shall send a response packet to the Management Controller within the required response time. However, the requested link state changes may take an unspecified amount of time to complete.

The actual link settings are controlled by the host NC driver when it is operational. When the host NC driver is operational, link settings specified by the MC using the Set Link command may be overwritten by the host NC driver. The link settings are not restored by the NC if the host NC driver becomes non-operational.

Table 42 illustrates the packet format of the Set Link command.

Table 42 – Set Link command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Link Settings			
20..23	OEM Link Settings			
24..27	Checksum			
28..45	iPad			

Table 43 and Table 44 describe the Set Link bit definitions. Refer to IEEE 802.3 for definitions of Auto Negotiation, Duplex Setting, Pause Capability, and Asymmetric Pause Capability.

Table 43 – Set Link bit definitions

Bit Position	Field Description	Value Description
00	Auto Negotiation	1b = enable 0b = disable
01..07	Link Speed Selection More than one speed can be selected when Auto Negotiation is set to 'enable'. If Auto Negotiation is not used, the channel attempts to force the link to the specified setting (in this case, if the setting is not supported or if multiple speeds are enabled, a Command Failed response code and Parameter Is Invalid, Unsupported, or Out-of-Range reason code shall be returned). NOTE Additional link speeds are defined below.	Bit 01: 1b = enable 10 Mbps
		Bit 02: 1b = enable 100 Mbps
		Bit 03: 1b = enable 1000 Mbps (1 Gbps)
		Bit 04: 1b = enable 10 Gbps
		Bit 05: 1b = enable 20 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0)
		Bit 06: 1b = enable 25 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0)
		Bit 07: 1b = enable 40 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0)
08..09	Duplex Setting (separate duplex setting bits) More than one duplex setting can be selected when Auto Negotiation is set to 'enable'. If Auto Negotiation is not used, the channel attempts to force the link to the specified setting (in this case, if the setting is not supported or if multiple settings are enabled, a Command Failed response code and Parameter Is Invalid, Unsupported, or Out-of-Range reason code shall be returned. If multiple settings are enabled, a Command Failed response code and Set Link Speed Conflict reason code shall be returned)."	Bit 08: 1b = enable half-duplex
		Bit 09: 1b = enable full-duplex
10	Pause Capability If Auto Negotiation is not used, the channel should apply pause settings assuming the partner supports the same capability.	1b = disable 0b = enable
11	Asymmetric Pause Capability If Auto Negotiation is not used, the channel should apply asymmetric pause settings assuming the partner supports the same capability.	1b = enable 0b = disable
12	OEM Link Settings Field Valid (see Table 44)	1b = enable 0b = disable

13..19	Additional Link Speeds (see Link Speed Selection)	Bit 13: 1b = enable 50 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) Bit 14: 1b = enable 100 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) Bit 15: 1b = enable 2.5 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) Bit 16: 1b = enable 5 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) Bit 17: 1b = enable 200 Gbps (optional for NC-SI 1.2, Reserved for NC-SI 1.1/1.0) Bit 18: 1b = enable 400 Gbps (optional for NC-SI 1.2, Reserved for NC-SI 1.1/1.0) Bit 19: 1b = enable 800 Gbps (optional for NC-SI 1.2, Reserved for NC-SI 1.1/1.0)
20..21	Reserved	0
22..23	Modulation Scheme	Bit 22: 1b = NRZ (optional for NC-SI 1.2, Reserved for NC-SI 1.1/1.0) Bit 23: 1b = PAM-4 (optional for NC-SI 1.2, Reserved for NC-SI 1.1/1.0) Bit 23-22 Values: 00 – Use default 01 – Enable NRZ 10 – Enable PAM-4 11 – Enable NRZ and PAM-4
24..27	Forward Error Correction (FEC) Algorithm	Bit 24: 1b = BASE-R FEC (Firecode) (optional for NC-SI 1.2, Reserved for NC-SI 1.1/1.0) Bit 25: 1b = RS-FEC (Reed Solomon) (optional for NC-SI 1.2, Reserved for NC-SI 1.1/1.0) Bit 26..27 Reserved If all bits are set to 0, then no FEC algorithm shall be selected
28	Energy Efficient Ethernet (EEE)	1b = enable 0b = disable
29	Link Training (LT)	1b = enable 0b = disable
30	Parallel Detect An auto-negotiation link partner's mechanism to establish links with non-negotiation, fixed-speed linked partners.	1b = enable 0b = disable

31	Reserved	0

2223 **Table 44 – OEM Set Link bit definitions**

Bit Position	Field Description	Value Description
00..31	OEM Link Settings	Vendor specified

2224 ### 8.4.22 Set Link Response (0x89)

2225 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Set Link
2226 command and send a response (see Table 45). In the presence of an operational Host NC driver, the NC
2227 should not attempt to make link state changes and should send a response with reason code 0x1 (Set
2228 Link Host OS/ Driver Conflict).

2229 If the Auto Negotiation field is set, the NC should ignore Link Speed Selection and Duplex Setting fields
2230 that are not supported by the NC.

2231 **Table 45 – Set Link response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

2232 Table 46 describes the reason codes that are specific to the Set Link command. Returning the following
2233 command-specific codes is recommended, conditional upon Network Controller support for the related
2234 capabilities.

2235 **Table 46 – Set Link command-specific reason codes**

Value	Description	Comment
0x0901	Set Link Host OS/ Driver Conflict	Returned when the Set Link command is received when the Host NC driver is operational
0x0902	Set Link Media Conflict	Returned when Set Link command parameters conflict with the media type (for example, Fiber Media)
0x0903	Set Link Parameter Conflict	Returned when Set Link parameters conflict with each other (for example, 1000 Mbps HD with copper media)
0x0904	Set Link Power Mode Conflict	Returned when Set Link parameters conflict with current low-power levels by exceeding capability
0x0905	Set Link Speed Conflict	Returned when Set Link parameters attempt to force more than one speed at the same time when autoneg is disabled
0x0906	Link Command Failed-Hardware Access Error	Returned when PHY R/W access fails to complete normally while executing the Set Link or Get Link Status command

Value	Description	Comment
0x0907	Set Link Serdes Conflict	Returned when Set Link parameters attempt to force an unsupported Serdes configuration
0x0908	Set Link FEC Conflict	Returned when Set Link parameters attempt to force an unsupported FEC algorithm
0x0909	Set Link EEE Conflict	Returned when Set Link parameters attempt to force an unsupported EEE configuration
0x090A	Set Link LT Conflict	Returned when Set Link parameters attempt to force an unsupported link training configuration
0x090B	Set Link Parallel Detection Conflict	Returned when Set Link parameters attempt to force an unsupported parallel detection configuration

8.4.23 Get Link Status command (0x0A)

The Get Link Status command allows the Management Controller to query the channel for potential link status and error conditions (see Table 47).

Table 47 – Get Link Status command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.24 Get Link Status response (0x8A)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get Link Status command and send a response (see Table 48).

Table 48 – Get Link Status response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Link Status			
24..27	Other Indications			
28..31	OEM Link Status			
32..35	Checksum			
36..45	Pad			

2244 Table 49 describes the Link Status bit definitions.

2245 **Table 49 – Link Status field bit definitions**

Bit Position	Field Description	Value Description
00	Link Flag	0b = Link is down 1b = Link is up This field is mandatory. NOTE If the IEEE 802.3az (EEE) is enabled on the link, Low Power Idle (LPI) state shall not be interpreted as “Link is down”.
04..01	Speed and duplex	0x0 = Auto-negotiate not complete [per IEEE 802.3], or SerDes Flag = 1b, or no Highest Common Denominator (HCD) from the following options (0x1 through 0xF) was found. 0x1 = 10BASE-T half-duplex 0x2 = 10BASE-T full-duplex 0x3 = 100BASE-TX half-duplex 0x4 = 100BASE-T4 0x5 = 100BASE-TX full-duplex 0x6 = 1000BASE-T half-duplex 0x7 = 1000BASE-T full-duplex 0x8 = 10G-BASE-T support or 10 Gbps 0x9 = 20 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) 0xA = 25 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) 0xB = 40 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) 0xC = 50 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) 0xD = 100 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) 0xE = 2.5 Gbps (optional for NC-SI 1.1, Reserved for NC-SI 1.0) 0xF = Use values defined in Enhanced Speed and Duplex field starting at bit 24 (optional for NC-SI 1.1, Reserved for NC-SI 1.0) When SerDes Flag = 0b, the value may reflect forced link setting. NOTE For the physical medium and/or speed/duplex not listed above, the closest speed and duplex option may be reported by the NC. This field should not be used to infer any media type information.
05	Auto Negotiate Flag	1b = Auto-negotiation is enabled. This field always returns 0b if auto-negotiation is not supported, or not enabled. This field is mandatory if supported by the controller.
06	Auto Negotiate Complete	1b = Auto-negotiation has completed. This includes if auto-negotiation was completed using Parallel Detection. Always returns 0b if auto-negotiation is not supported or is not enabled. This field is mandatory if the Auto Negotiate Flag is supported.

Bit Position	Field Description	Value Description
07	Parallel Detection Flag	1b = Link partner did not support auto-negotiation and parallel detection was used to get link. This field contains 0b if Parallel Detection was not used to obtain link.
08	Reserved	None
09	Link Partner Advertised Speed and Duplex 1000TFD	1b = Link Partner is 1000BASE-T full-duplex capable. Valid when: SerDes Flag = 0b Auto-Negotiate Flag = 1b Auto-Negotiate Complete = 1b This field is mandatory.
10	Link Partner Advertised Speed and Duplex 1000THD	1b = Link Partner is 1000BASE-T half-duplex capable. Valid when: SerDes Flag = 0b Auto-Negotiate Flag = 1b Auto-Negotiate Complete = 1b This field is mandatory.
11	Link Partner Advertised Speed 100T4	1b = Link Partner is 100BASE-T4 capable. Valid when: SerDes Flag = 0b Auto-Negotiate Flag = 1b Auto-Negotiate Complete = 1b This field is mandatory.
12	Link Partner Advertised Speed and Duplex 100TXFD	1b = Link Partner is 100BASE-TX full-duplex capable. Valid when: SerDes Flag = 0b Auto-Negotiate Flag = 1b Auto-Negotiate Complete = 1b This field is mandatory.
13	Link Partner Advertised Speed and Duplex 100TXHD	1b = Link Partner is 100BASE-TX half-duplex capable. Valid when: SerDes Flag = 0b Auto-Negotiate Flag = 1b Auto-Negotiate Complete = 1b This field is mandatory.

Bit Position	Field Description	Value Description
14	Link Partner Advertised Speed and Duplex 10TFD	1b = Link Partner is 10BASE-T full-duplex capable. Valid when: SerDes Flag = 0b Auto-Negotiate Flag = 1b Auto-Negotiate Complete = 1b This field is mandatory.
15	Link Partner Advertised Speed and Duplex 10THD	1b = Link Partner is 10BASE-T half-duplex capable. Valid when: SerDes Flag = 0b Auto-Negotiate Flag = 1b Auto-Negotiate Complete = 1b This field is mandatory.
16	TX Flow Control Flag	0b = Transmission of Pause frames by the NC onto the external network interface is disabled. 1b = Transmission of Pause frames by the NC onto the external network interface is enabled. This field is mandatory.
17	RX Flow Control Flag	0b = Reception of Pause frames by the NC from the external network interface is disabled. 1b = Reception of Pause frames by the NC from the external network interface is enabled. This field is mandatory.
19..18	Link Partner Advertised Flow Control	00b = Link partner is not pause capable. 01b = Link partner supports symmetric pause. 10b = Link partner supports asymmetric pause toward link partner. 11b = Link partner supports both symmetric and asymmetric pause. Valid when: SerDes Flag = 0b Auto-Negotiate = 1b Auto-Negotiate Complete = 1b This field is mandatory.

Bit Position	Field Description	Value Description
20	SerDes Link	SerDes status (See 4.18.) 0b = SerDes is not used 1b = SerDes is used This field is mandatory. NOTE This bit should not be set if the SerDes is used to connect to an external PHY that connects to the network. This bit should be set if the SerDes interface is used as a direct attach interface to connect.
21	OEM Link Speed Valid	0b = OEM link settings are invalid. 1b = OEM link settings are valid.
22..23	Modulation Scheme	00b = Reserved 01b = NRZ is used. 10b = PAM-4 is used. 11b = Reserved This field is optional for NC-SI 1.2, reserved for NC-SI 1.1/1.0.
31..24	Extended Speed and duplex	Optional for NC-SI 1.2/1.1, Reserved for NC-SI 1.0 0x0 = Auto-negotiate not complete [per IEEE 802.3], or SerDes Flag = 1b, or no highest common denominator speed from the following options (0x01 through 0x0F) was found. 0x01 = 10BASE-T half-duplex 0x02 = 10BASE-T full-duplex 0x03 = 100BASE-TX half-duplex 0x04 = 100BASE-T4 0x05 = 100BASE-TX full-duplex 0x06 = 1000BASE-T half-duplex 0x07 = 1000BASE-T full-duplex 0x08 = 10G-BASE-T support or 10 Gbps 0x09 = 20 Gbps 0x0A = 25 Gbps 0x0B = 40 Gbps 0x0C = 50 Gbps 0x0D = 100 Gbps 0x0E = 2.5 Gbps 0x10 = 1 Gbps (for non Base-T) 0x11 = 200 Gbps 0x12 = 400 Gbps 0x13 = 800 Gbps 0x14-0xFF = Reserved When SerDes Flag = 0b, the value may reflect forced link setting. NOTE For the physical medium and/or speed/duplex not listed above, the closest speed and duplex option may be reported by the NC. This field should not be used to infer any media type information.

Table 50 describes the Other Indications field bit definitions.

Table 50 – Other Indications field bit definitions

Bits	Description	Values
00	Host NC Driver Status Indication	0b = The Network Controller driver for the host external network interface associated with this channel is not operational (not running), unknown, or not supported. 1b = The Network Controller driver for the host external network interface associated with this channel is being reported as operational (running). This bit always returns 0b if the Host NC Driver Status Indication is not supported.
01	Energy Efficient Ethernet (EEE)	1b = enabled 0b = disabled
02	Link Training (LT)	1b = enabled 0b = disabled
03	Parallel Detect	1b = enabled 0b = disabled
04..31	Reserved	None

Table 51 describes the OEM Link Status field bit definitions.

Table 51 – OEM Link Status field bit definitions (optional)

Bits	Description	Values
00..31	OEM Link Status	OEM specific

Table 52 describes the reason code that is specific to the Get Link Status command.

Table 52 – Get Link Status command-specific reason code

Value	Description	Comment
0x0A06	Link Command Failed-Hardware Access Error	Returned when PHY R/W access fails to complete normally while executing the Set Link or Get Link Status command

8.4.25 Set VLAN Filter command (0x0B)

The Set VLAN Filter command is used by the Management Controller to program one or more VLAN IDs that are used for VLAN filtering.

Incoming packets that match both a VLAN ID filter and a MAC address filter are forwarded to the Management Controller. Other packets may be dropped based on the VLAN filtering mode per the Enable VLAN command.

The quantity of each filter type that is supported by the channel can be discovered by means of the Get Capabilities command. Up to 15 filters can be supported per channel. A Network Controller implementation shall support at least one VLAN filter per channel.

To configure a VLAN filter, the Management Controller issues a Set VLAN Filter command with the Filter Selector field indicating which filter is to be configured, the VLAN ID field set to the VLAN TAG values to be used by the filter, and the Enable field set to either enable or disable the selected filter.

The VLAN-related fields are specified per IEEE 802.1q. When VLAN Tagging is used, the packet includes a Tag Protocol Identifier (TPID) field and VLAN Tag fields, as shown in Table 53.

Table 53 – IEEE 802.1q VLAN Fields

Field	Size	Description
TPI	2 bytes	Tag Protocol Identifier = 8100h
VLAN TAG – user priority	3 bits	User Priority (typical value = 000b)
VLAN TAG – CFI	1 bit	Canonical Format Indicator = 0b
VLAN TAG – VLAN ID	12 bits	Zeros = no VLAN

When checking VLAN field values, the Network Controller shall match against the enabled VLAN Tag Filter values that were configured with the Set VLAN Filter command. The Network Controller shall also match on the TPI value of 8100h, as specified by IEEE 802.1q. Matching against the User Priority/CFI bits is optional. An implementation may elect to ignore the setting of those fields.

Table 54 illustrates the packet format of the Set VLAN Filter command.

Table 54 – Set VLAN Filter command packet format

	Bits								
Bytes	**31..24**		**23..16**		**15..08**		**07..00**		
00..15	NC-SI Control Packet Header								
16..19	Reserved				User Priority/CFI		VLAN ID		
20..23	Reserved				Filter Selector		Reserved		E
24..27	Checksum								
28..45	Pad								

Table 55 provides possible settings for the Filter Selector field. Table 56 provides possible settings for the Enable (E) field.

Table 55 – Possible Settings for Filter Selector field (8-bit field)

Value	Description
1	Settings for VLAN filter number 1
2	Settings for VLAN filter number 2
..	
N	Settings for VLAN filter number *N*

2276 **Table 56 – Possible Settings for Enable (E) field (1-bit field)**

Value	Description
0b	Disable this VLAN filter
1b	Enable this VLAN filter

2277 ### 8.4.26 Set VLAN Filter response (0x8B)

2278 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Set
2279 VLAN Filter command and send a response (see Table 57).

2280 **Table 57 – Set VLAN Filter response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

2281 Table 58 describes the reason code that is specific to the Set VLAN Filter command.

2282 **Table 58 – Set VLAN Filter command-specific reason code**

Value	Description	Comment
0x0B07	VLAN Tag Is Invalid	Returned when the VLAN ID is invalid (VLAN ID = 0)

2283 ### 8.4.27 Enable VLAN command (0x0C)

2284 The Enable VLAN command may be used by the Management Controller to enable the channel to accept
2285 VLAN-tagged packets from the network for NC-SI Pass-through operation (see Table 59).

2286 **Table 59 – Enable VLAN command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Reserved			Mode #
20..23	Checksum			
24..45	Pad			

2287 Table 60 describes the modes for the Enable VLAN command.

2288 **Table 60 – VLAN Enable modes**

Mode	#	O/M	Description
Reserved	0x00	N/A	Reserved
VLAN only	0x01	M	Only VLAN-tagged packets that match the enabled VLAN Filter settings (and also match the MAC Address Filtering configuration) are accepted. Non-VLAN-tagged packets are not accepted.
VLAN + non-VLAN	0x02	O	VLAN-tagged packets that match the enabled VLAN Filter settings (and also match the MAC Address Filtering configuration) are accepted. Non-VLAN-tagged packets (that also match the MAC Address Filtering configuration) are also accepted.
Any VLAN + non-VLAN	0x03	O	Any VLAN-tagged packets that also match the MAC Address Filtering configuration are accepted, regardless of the VLAN Filter settings. Non-VLAN-tagged packets (that also match the MAC Address Filtering configuration) are also accepted.
Reserved	0x04 – 0xFF	N/A	Reserved

2289 ### 8.4.28 Enable VLAN response (0x8C)

2290 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Enable
2291 VLAN command and send a response.

2292 Currently no command-specific reason code is identified for this response (see Table 61).

2293 **Table 61 – Enable VLAN response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

2294 ### 8.4.29 Disable VLAN command (0x0D)

2295 The Disable VLAN command may be used by the Management Controller to disable VLAN filtering. In the
2296 disabled state, only non-VLAN-tagged packets (that also match the MAC Address Filtering configuration)
2297 are accepted. VLAN-tagged packets are not accepted.

2298 Table 62 illustrates the packet format of the Disable VLAN command.

2299 **Table 62 – Disable VLAN command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

2300 ### 8.4.30 Disable VLAN response (0x8D)

2301 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Disable
2302 VLAN command and send a response.

2303 Currently no command-specific reason code is identified for this response (see Table 63).

2304 **Table 63 – Disable VLAN response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

2305 ### 8.4.31 Set MAC Address command (0x0E)

2306 The Set MAC Address command is used by the Management Controller to program the channel's unicast
2307 or multicast MAC address filters.

2308 The channel supports one or more "perfect match" MAC address filters that are used to selectively
2309 forward inbound frames to the Management Controller. Assuming that a packet passes any VLAN filtering
2310 that may be active, it will be forwarded to the Management Controller if its 48-bit destination MAC address
2311 exactly matches an active MAC address filter.

2312 MAC address filters may be configured as unicast or multicast addresses, depending on the capability of
2313 the channel. The channel may implement three distinct types of filter:

2314 - **Unicast filters** support exact matching on 48-bit unicast MAC addresses (AT = 0x0 only).
2315 - **Multicast filters** support exact matching on 48-bit multicast MAC addresses (AT = 0x1 only).
2316 - **Mixed filters** support matching on both unicast and multicast MAC addresses. (AT=0x0 or
2317 AT=0x1)

2318 The number of each type of filter that is supported by the channel can be discovered by means of the Get
2319 Capabilities command. The channel shall support at least one unicast address filter or one mixed filter, so
2320 that at least one unicast MAC address filter may be configured on the channel. Support for any
2321 combination of unicast, multicast, or mixed filters beyond this basic requirement is vendor specific. The
2322 total number of all filters shall be less than or equal to 8.

To configure an address filter, the Management Controller issues a Set MAC Address command with the Address Type field indicating the type of address to be programmed (unicast or multicast) and the MAC Address Num field indicating the specific filter to be programmed.

Filters are addressed using a 1-based index ordered over the unicast, multicast, and mixed filters reported by means of the Get Capabilities command. For example, if the interface reports four unicast filters, two multicast filters, and two mixed filters, then MAC Address numbers 1 through 4 refer to the interface's unicast filters, 5 and 6 refer to the multicast filters, and 7 and 8 refer to the mixed filters. Similarly, if the interface reports two unicast filters, no multicast filters, and six mixed filters, then MAC address numbers 1 and 2 refer to the unicast filters, and 3 through 8 refer to the mixed filters.

The filter type of the filter to be programmed (unicast, multicast, or mixed) shall be compatible with the Address Type being programmed. For example, programming a mixed filter to a unicast address is allowed, but programming a multicast filter to a unicast address is an error.

The Enable field determines whether the indicated filter is to be enabled or disabled. When a filter is programmed to be enabled, the filter is loaded with the 48-bit MAC address in the MAC Address field of the command, and the channel enables forwarding of frames that match the configured address. If the specified filter was already enabled, it is updated with the new address provided.

When a filter is programmed to be disabled, the contents of the MAC Address field are ignored. Any previous MAC address programmed in the filter is discarded and the channel no longer uses this filter in its packet-forwarding function.

Only unicast MAC addresses, specified with AT set to 0x0, should be used in source MAC address checking and for determining the NC-SI channel for Pass-through transmit traffic.

Table 64 illustrates the packet format of the Set MAC Address command.

Table 64 – Set MAC Address command packet format

	Bits					
Bytes	**31..24**	**23..16**	**15..08**	**07..00**		
00..15	NC-SI Control Packet Header					
16..19	MAC Address byte 5	MAC Address byte 4	MAC Address byte 3	MAC Address byte 2		
20..23	MAC Address byte 1	MAC Address byte 0	MAC Address Num	AT	Rsvd	E
24..27	Checksum					
28..45	Pad					

NOTE AT = Address Type, E = Enable.

Table 65 provides possible settings for the MAC Address Number field. Table 66 provides possible settings for the Address Type (AT) field. Table 67 provides possible settings for the Enable (E) field.

Table 65 – Possible settings for MAC Address Number (8-bit field)

Value	Description
0x01	Configure MAC address filter number 1
0x02	Configure MAC address filter number 2
..	

Value	Description
N	Configure MAC address filter number *N*

Table 66 – Possible settings for Address Type (3-bit field)

Value	Description
0x0	Unicast MAC address
0x1	Multicast MAC address
0x2–0x7	Reserved

Table 67 – Possible settings for Enable Field (1-bit field)

Value	Description
0b	Disable this MAC address filter
1b	Enable this MAC address filter

8.4.32 Set MAC Address response (0x8E)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Set MAC Address command and send a response (see Table 68).

Table 68 – Set MAC Address response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

Table 69 describes the reason code that is specific to the Set MAC Address command.

Table 69 – Set MAC Address command-specific reason code

Value	Description	Comment
0x0E08	MAC Address Is Zero	Returned when the Set MAC Address command is received with the MAC address set to 0

8.4.33 Enable Broadcast Filter command (0x10)

The Enable Broadcast Filter command allows the Management Controller to control the forwarding of broadcast frames to the Management Controller. The channel, upon receiving and processing this command, shall filter all received broadcast frames based on the broadcast packet filtering settings specified in the payload. If no broadcast packet types are specified for forwarding, all broadcast packets shall be filtered out.

2363 The Broadcast Packet Filter Settings field is used to specify those protocol-specific broadcast filters that
2364 should be activated. The channel indicates which broadcast filters it supports in the Broadcast Filter
2365 Capabilities field of the Get Capabilities Response frame defined in 8.4.46.

2366 Table 70 illustrates the packet format of the Enable Broadcast Filter command.

2367 **Table 70 – Enable Broadcast Filter command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Broadcast Packet Filter Settings			
20..23	Checksum			
24..45	Pad			

2368 Table 71 describes the Broadcast Packet Filter Settings field bit definitions.

2369 **Table 71 – Broadcast Packet Filter Settings field**

Bit Position	Field Description	Value Description
0	ARP Packets	`1b` = Forward this packet type to the Management Controller. `0b` = Filter out this packet type. For the purposes of this specification, an ARP broadcast packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to the layer 2 broadcast address (`FF:FF:FF:FF:FF:FF`). • The EtherType field set to `0x0806`. This field is mandatory.
1	DHCP Client Packets	`1b` = Forward this packet type to the Management Controller. `0b` = Filter out this packet type. For the purposes of this filter, a DHCP client broadcast packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to the layer 2 broadcast address (`FF:FF:FF:FF:FF:FF`). • The EtherType field is set to `0x0800` (IPv4). • The IP header's Protocol field is set to 17 (UDP). • The UDP destination port number is set to 68. This field is optional. If unsupported, broadcast DHCP client packets will be blocked when broadcast filtering is enabled. The value shall be set to 0 if unsupported.

Bit Position	Field Description	Value Description
2	DHCP Server Packets	1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type. For the purposes of this filter, a DHCP server broadcast packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to the layer 2 broadcast address (FF:FF:FF:FF:FF:FF). • The EtherType field is set to 0x0800 (IPv4). • The IP header's Protocol field is set to 17 (UDP). • The UDP destination port number is set to 67. This field is optional. If unsupported, broadcast DHCP packets will be blocked when broadcast filtering is enabled. The value shall be set to 0b if unsupported.
3	NetBIOS Packets	1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type. For the purposes of this filter, NetBIOS broadcast packets are defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to the layer 2 broadcast address (FF:FF:FF:FF:FF:FF). • The EtherType field is set to 0x0800 (IPv4). • The IP header's Protocol field is set to 17 (UDP). • The UDP destination port number is set to 137 for NetBIOS Name Service or 138 for NetBIOS Datagram Service, per the assignment of IANA well-known ports. This field is optional. If unsupported, broadcast NetBIOS packets will be blocked when broadcast filtering is enabled. The value shall be set to 0b if unsupported.
4..31	Reserved	None

8.4.34 Enable Broadcast Filter response (0x90)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Enable Broadcast Filter command and send a response.

Currently no command-specific reason code is identified for this response (see Table 72).

Table 72 – Enable Broadcast Filter response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.35 Disable Broadcast Filter command (0x11)

The Disable Broadcast Filter command may be used by the Management Controller to disable the broadcast filter feature and enable the reception of all broadcast frames. Upon processing this command, the channel shall discontinue the filtering of received broadcast frames.

Table 73 illustrates the packet format of the Disable Broadcast Filter command.

Table 73 – Disable Broadcast Filter command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.36 Disable Broadcast Filter response (0x91)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Disable Broadcast Filter command and send a response.

Currently no command-specific reason code is identified for this response (see Table 74).

Table 74 – Disable Broadcast Filter response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.37 Enable Global Multicast Filter command (0x12)

The Enable Global Multicast Filter command is used to activate global filtering of multicast frames with optional filtering of specific multicast protocols. Upon receiving and processing this command, the channel shall only deliver multicast frames that match specific multicast MAC addresses enabled for Pass through using this command or the Set MAC Address command.

The Multicast Packet Filter Settings field is used to specify optional, protocol-specific multicast filters that should be activated. The channel indicates which optional multicast filters it supports in the Multicast Filter Capabilities field of the Get Capabilities Response frame defined in 8.4.46. The Management Controller should not set bits in the Multicast Packet Filter Settings field that are not indicated as supported in the Multicast Filter Capabilities field.

Neighbor Solicitation messages are sent to a Solicited Node multicast address that is derived from the target node's IPv6 address. This command may be used to enable forwarding of solicited node multicasts.

2399 The IPv6 neighbor solicitation filter, as defined in this command, may not be supported by the Network
2400 Controller. In this case, the Management Controller may configure a multicast or mixed MAC address
2401 filter for the specific Solicited Node multicast address using the Set MAC Address command to enable
2402 forwarding of Solicited Node multicasts.

2403 This command shall be implemented if the channel implementation supports accepting all multicast
2404 addresses. An implementation that does not support accepting all multicast addresses shall not
2405 implement these commands. Pass-through packets with multicast addresses can still be accepted
2406 depending on multicast address filter support provided by the Set MAC Address command. Multicast filter
2407 entries that are set to be enabled in the Set MAC Address command are accepted; all others are rejected.
2408 Table 75 illustrates the packet format of the Enable Global Multicast Filter command. Unsupported fields
2409 should be treated as reserved fields unless otherwise specified.

2410

2411 **Table 75 – Enable Global Multicast Filter command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Multicast Packet Filter Settings			
20..23	Checksum			
24..45	Pad			

2412 Table 76 describes the bit definitions for the Multicast Packet Filter Settings field.

2413 **Table 76 – Bit Definitions for Multicast Packet Filter Settings field**

Bit Position	Field Description	Value Description
0	IPv6 Neighbor Advertisement	`1b` = Forward this packet type to the Management Controller. `0b` = Filter out this packet type. For the purposes of this specification, an IPv6 Neighbor Advertisement multicast packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to a layer 2 multicast address of the form `33:33:00:00:00:01`. This address corresponds to the All_Nodes (`FF02::1`) multicast address. • The EtherType field is set to `0x86DD` (IPv6). • The IPv6 header's Next Header field is set to 58 (ICMPv6). • The ICMPv6 header's Message Type field is set to the following value: 136 – Neighbor Advertisement. This field is optional.

Bit Position	Field Description	Value Description
1	IPv6 Router Advertisement	1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type. For the purposes of this specification, an IPv6 Router Advertisement multicast packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to a layer 2 multicast address of the form 33:33:00:00:00:01. This corresponds to the All_Nodes multicast address, FF02::1. • The EtherType field is set to 0x86DD (IPv6). • The IPv6 header's Next Header field is set to 58 (ICMPv6). • The ICMPv6 header's Message Type field is set to 134. This field is optional.
2	DHCPv6 relay and server multicast	1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type. For the purposes of this filter, a DHCPv6 multicast packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to the layer 2 multicast address 33:33:00:01:00:02 or 33:33:00:01:00:03. These correspond to the IPv6 multicast addresses FF02::1:2 (All_DHCP_Relay_Agents_and_Servers) and FF05::1:3 (All_DHCP_Servers). • The EtherType field is set to 0x86DD (IPv6). • The IPv6 header's Next Header field is set to 17 (UDP). • The UDP destination port number is set to 547. This field is optional.
3	DHCPv6 multicasts from server to clients listening on well-known UDP ports	1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type. For the purposes of this filter, a DHCPv6 multicast packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to the layer 2 multicast address 33:33:00:01:00:02. These correspond to the IPv6 multicast addresses FF02::1:2 (All_DHCP_Relay_Agents_and_Servers). • The EtherType field is set to 0x86DD (IPv6). • The IPv6 header's Next Header field is set to 17 (UDP). • The UDP destination port number is set to 546. This field is optional.

Bit Position	Field Description	Value Description
4	IPv6 MLD	1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type. For the purposes of this specification, an IPv6 MLD packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to a layer 2 multicast address of the form 33:33:00:00:00:01. This address corresponds to the All_Nodes (FF02::1) multicast address. • The EtherType field is set to 0x86DD (IPv6). • The IPv6 header's Next Header field is set to 58 (ICMPv6). • The ICMPv6 header's Message Type field is set to one of the following values: 130 (Multicast Listener Query), 131 (Multicast Listener Report), 132 (Multicast Listener Done) This field is optional.
5	IPv6 Neighbor Solicitation	1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type. For the purposes of this specification, an IPv6 MLD packet is defined to be any packet that meets all of the following requirements: • The destination MAC address field is set to a layer 2 multicast address of the form 33:33:FF:XX:XX:XX. This address corresponds to the Solicited Note multicast address where the last three bytes of the destination MAC address are ignored for this filter. • The EtherType field is set to 0x86DD (IPv6). • The IPv6 header's Next Header field is set to 58 (ICMPv6). • The ICMPv6 header's Message Type field is set to one of the following values: 135 This field is optional. IMPLEMENTATION NOTE Enabling of this filter results in receiving all IPv6 neighbor solicitation traffic on this channel. If IPv6 neighbor solicitation traffic for a specific multicast address is of interest, then it is recommended that the MC uses a multicast address filter (configured for the multicast address using the Set MAC Address command) instead of this filter.
31..6	Reserved	None

8.4.38 Enable Global Multicast Filter response (0x92)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Enable Global Multicast Filter command and send a response.

Currently no command-specific reason code is identified for this response (see Table 77).

2418 **Table 77 – Enable Global Multicast Filter response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

2419 ## 8.4.39 Disable Global Multicast Filter command (0x13)

2420 The Disable Global Multicast Filter command is used to disable global filtering of multicast frames. Upon
2421 receiving and processing this command, and regardless of the current state of multicast filtering, the
2422 channel shall forward all multicast frames to the Management Controller.

2423 This command shall be implemented on the condition that the channel implementation supports accepting
2424 all multicast addresses. An implementation that does not support accepting all multicast addresses shall
2425 not implement these commands. Pass-through packets with multicast addresses can still be accepted
2426 depending on multicast address filter support provided by the Set MAC Address command. Packets with
2427 destination addresses matching multicast filter entries that are set to enabled in the Set MAC Address
2428 command are accepted; all others are rejected.

2429 Table 78 illustrates the packet format of the Disable Global Multicast Filter command.

2430 **Table 78 – Disable Global Multicast Filter command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

2431 ## 8.4.40 Disable Global Multicast Filter response (0x93)

2432 In the absence of any errors, the channel shall process and respond to the Disable Global Multicast Filter
2433 command by sending the response packet shown in Table 79.

2434 Currently no command-specific reason code is identified for this response.

2435 **Table 79 – Disable Global Multicast Filter response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.41 Set NC-SI Flow Control command (0x14)

The Set NC-SI Flow Control command allows the Management Controller to configure IEEE 802.3 pause packet flow control on the NC-SI.

The Set NC-SI Flow Control command is addressed to the package, rather than to a particular channel (that is, the command is sent with a Channel ID where the Package ID subfield matches the ID of the intended package and the Internal Channel ID subfield is set to 0x1F).

When enabled for flow control, a channel may direct the package to generate and renew 802.3x (XOFF) PAUSE Frames for a maximum interval of T12 for a single congestion condition. If the congestion condition remains in place after a second T12 interval expires, the congested channel shall enter the Initial State and remove its XOFF request to the package. Note that some implementations may have shared buffering arrangements where all channels within the package become congested simultaneously. Also note that if channels become congested independently, the package may not immediately go into the XON state after T12 if other channels within the package are still requesting XOFF.

The setting of IEEE 802.3 pause packet flow control on the NC-SI is independent from any arbitration scheme, if any is used.

Table 80 illustrates the packet format of the Set NC-SI Flow Control command.

Table 80 – Set NC-SI Flow Control command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Reserved			Flow Control Enable
20..23	Checksum			
24..45	Pad			

Table 81 describes the values for the Flow Control Enable field.

Table 81 – Values for the Flow Control Enable field (8-bit field)

Value	Description
0x0	Disables NC-SI flow control
0x1	Enables Network Controller to Management Controller flow control frames (Network Controller generates flow control frames) This field is optional.
0x2	Enables Management Controller to Network Controller flow control frames (Network Controller accepts flow control frames) This field is optional.
0x3	Enables bi-directional flow control frames This field is optional.
0x4..0xFF	Reserved

8.4.42 Set NC-SI Flow Control response (0x94)

The package shall, in the absence of a checksum error or identifier mismatch, always accept the Set NC-SI Flow Control command and send a response (see Table 82).

Table 82 – Set NC-SI Flow Control response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

Table 83 describes the reason code that is specific to the Set NC-SI Flow Control command.

Table 83 – Set NC-SI Flow Control command-specific reason code

Value	Description	Comment
0x1409	Independent transmit and receive enable/disable control is not supported	Returned when the implementation requires that both transmit and receive flow control be enabled and disabled simultaneously

8.4.43 Get Version ID command (0x15)

The Get Version ID command may be used by the Management Controller to request the channel to provide the controller and firmware type and version strings listed in the response payload description.

Table 84 illustrates the packet format of the Get Version ID command.

Table 84 – Get Version ID command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.44 Get Version ID Response (0x95)

The channel shall, in the absence of an error, always accept the Get Version ID command and send the response packet shown in Table 85. Currently no command-specific reason code is identified for this response.

Note: When multiple Physical Functions are enabled on the channel, the PCI ID that is returned shall be that of the lowest numbered Function on the channel.

Table 85 – Get Version ID response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	NC-SI Version			
	Major	Minor	Update	Alpha1
24..27	reserved	reserved	reserved	Alpha2
28..31	Firmware Name String (11-08)			
32..35	Firmware Name String (07-04)			
36..39	Firmware Name String (03-00)			
40..43	Firmware Version			
	MS-byte (3)	Byte (2)	Byte (1)	LS-byte (0)
44..47	PCI DID		PCI VID	
48..51	PCI SSID		PCI SVID	
52..55	Manufacturer ID (IANA)			
56..59	Checksum			

8.4.44.1 NC-SI Version encoding

The NC-SI Version field holds the version number of the NC-SI specification with which the controller is compatible. The version field shall be encoded as follows:

- The ‘major’, ‘minor’, and ‘update’ bytes are BCD-encoded, and each byte holds two BCD digits.
- The ‘alpha’ byte holds an optional alphanumeric character extension that is encoded using the ISO/IEC 8859-1 Character Set.
- The semantics of these fields follow the semantics specified in DSP4014.
- The value `0x00` in the Alpha1 or Alpha2 fields means that the corresponding alpha field is not used. The Alpha1 field shall be used first.
- The value `0xF` in the most-significant nibble of a BCD-encoded value indicates that the most-significant nibble should be ignored and the overall field treated as a single digit value.
- A value of `0xFF` in the update field indicates that the entire field is not present. `0xFF` is not allowed as a value for the major or minor fields.

EXAMPLE: Version 3.7.10a → `0xF3F7106100`
Version 10.01.7 → `0x1001F70000`
Version 3.1 → `0xF3F1FF0000`
Version 1.0a → `0xF1F0FF4100`
Version 1.0ab → `0xF1F0FF4142 (Alpha1 = 0x41, Alpha2 = 0x42)`

8.4.44.2 Firmware Name encoding

The Firmware Name String shall be encoded using the ISO/IEC 8859-1 Character Set. Strings are left-justified where the leftmost character of the string occupies the most-significant byte position of the

Firmware Name String field, and characters are populated starting from that byte position. The string is null terminated if the string is smaller than the field size. That is, the delimiter value, 0x00, follows the last character of the string if the string occupies fewer bytes than the size of the field allows. A delimiter is not required if the string occupies the full size of the field. Bytes following the delimiter (if any) should be ignored and can be any value.

8.4.44.3 Firmware Version encoding

To facilitate a common way of representing and displaying firmware version numbers across different vendors, each byte is hexadecimal encoded where each byte in the field holds two hexadecimal digits. The Firmware Version field shall be encoded as follows. The bytes are collected into a single 32-bit field where each byte represents a different 'point number' of the overall version. The selection of values that represent a particular version of firmware is specific to the Network Controller vendor.

Software displaying these numbers should not suppress leading zeros, which should help avoid user confusion in interpreting the numbers. For example, consider the two values 0x05 and 0x31. Numerically, the byte 0x31 is greater that 0x05, but if leading zeros were incorrectly suppressed, the two displayed values would be ".5" and ".31", respectively, and a user would generally interpret 0.5 as representing a greater value than 0.31 instead of 0.05 being smaller than 0.31. Similarly, if leading zeros were incorrectly suppressed, the value 0x01 and 0x10 would be displayed as 0.1 and 0.10, which could potentially be misinterpreted as representing the same version instead of 0.01 and 0.10 versions.

EXAMPLE: 0x00030217 → Version 00.03.02.17
0x010100A0 → Version 01.01.00.A0

8.4.44.4 PCI ID fields

These fields (PCI DID, PCI VID, PCI SSID, PCI SVID) hold the PCI ID information for the Network Controller when the Network Controller incorporates a PCI or PCI Express™ interface that provides a host network interface connection that is shared with the NC-SI connection to the network.

If this field is not used, the values shall all be set to zeros (0000h). Otherwise, the fields shall hold the PCI ID information for the host interface as defined by the version of the PCI/PCI Express™ specification to which the device's interface was designed.

If multiple partitions are enabled on the channel, the values should represent the PCI ID of the lowest Function number assigned to the channel by the Set PF Assignment command (0x28).

8.4.44.5 Manufacturer ID (IANA) field

The Manufacturer ID holds the IANA Enterprise Number for the manufacturer of the Network Controller as a 32-bit binary number. If the field is unused, the value shall be set to 0xFFFFFFFF.

8.4.45 Get Capabilities command (0x16)

The Get Capabilities command is used to discover additional optional functions supported by the channel, such as the number of unicast/multicast addresses supported, the amount of buffering in bytes available for packets bound for the Management Controller, and so on.

Table 86 illustrates the packet format for the Get Capabilities command.

Table 86 – Get Capabilities command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.46 Get Capabilities response (0x96)

In the absence of any errors, the channel shall process and respond to the Get Capabilities Command and send the response packet shown in Table 87. Currently no command-specific reason code is identified for this response.

Table 87 – Get Capabilities response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Capabilities Flags			
24..27	Broadcast Packet Filter Capabilities			
28..31	Multicast Packet Filter Capabilities			
32..35	Buffering Capability			
36..39	AEN Control Support			
40..43	VLAN Filter Count	Mixed Filter Count	Multicast Filter Count	Unicast Filter Count
44..47	Reserved		VLAN Mode Support	Channel Count
48..51	Checksum			

8.4.46.1 Capabilities Flags field

The Capabilities Flags field indicates which optional features of this specification the channel supports, as described in Table 88.

2540 **Table 88 – Capabilities Flags bit definitions**

Bit Position	Field Description	Value Description
0	Hardware Arbitration Capability	0b = Hardware arbitration capability is not supported by the package. 1b = Hardware arbitration capability is supported by the package.
1	Host NC Driver Status	0b = Host NC Driver Indication status is not supported. 1b = Host NC Driver Indication status is supported. See Table 50 for the definition of Host NC Driver Indication Status.
2	Network Controller to Management Controller Flow Control Support	0b = Network Controller to Management Controller flow control is not supported. 1b = Network Controller to Management Controller flow control is supported.
3	Management Controller to Network Controller Flow Control Support	0b = Management Controller to Network Controller flow control is not supported. 1b = Management Controller to Network Controller flow control is supported.
4	All multicast addresses support	0b = The channel cannot accept all multicast addresses. The channel does not support enable/disable global multicast commands. 1b = The channel can accept all multicast addresses. The channel supports enable/disable global multicast commands.
6..5	Hardware Arbitration Implementation Status	00b = Unknown 01b = Hardware arbitration capability is not implemented for the package on the given system. 10b = Hardware arbitration capability is implemented for the package on the given system. 11b = Reserved.
7..31	Reserved	Reserved

2541 ### 8.4.46.2 Broadcast Packet Filter Capabilities field

2542 The Broadcast Packet Filter Capabilities field defines the optional broadcast packet filtering capabilities
2543 that the channel supports. The bit definitions for this field correspond directly with the bit definitions for the
2544 Broadcast Packet Filter Settings field defined for the Enable Broadcast Filter command in Table 71. A bit
2545 set to 1 indicates that the channel supports the filter associated with that bit position; otherwise, the
2546 channel does not support that filter.

2547 ### 8.4.46.3 Multicast Packet Filter Capabilities field

2548 The Multicast Packet Filter Capabilities field defines the optional multicast packet filtering capabilities that
2549 the channel supports. The bit definitions for this field correspond directly with the bit definitions for the
2550 Multicast Packet Filter Settings field defined for the Enable Global Multicast Filter command in Table 76.
2551 A bit set to 1 indicates that the channel supports the filter associated with that bit position; otherwise, the
2552 channel does not support that filter.

8.4.46.4 Buffering Capability field

The Buffering Capability field defines the amount of buffering in bytes that the channel provides for inbound packets destined for the Management Controller. The Management Controller may make use of this value in software-based Device Selection implementations to determine the relative time for which a specific channel may be disabled before it is likely to start dropping packets. A value of 0 indicates that the amount of buffering is unspecified.

8.4.46.5 AEN Control Support field

The AEN Control Support field indicates various standard AENs supported by the implementation. The format of the field is shown in Table 40.

8.4.46.6 VLAN Filter Count field

The VLAN Filter Count field indicates the number of VLAN filters, up to 15, that the channel supports, as defined by the Set VLAN Filter command.

8.4.46.7 Mixed, Multicast, and Unicast Filter Count fields

The Mixed Filter Count field indicates the number of mixed address filters that the channel supports. A mixed address filter can be used to filter on specific unicast or multicast MAC addresses.

The Multicast Filter Count field indicates the number of multicast MAC address filters that the channel supports.

The Unicast Filter Count field indicates the number of unicast MAC address filters that the channel supports.

The channel is required to support at least one unicast or mixed filter, such that at least one unicast MAC address can be configured on the interface. The total number of unicast, multicast, and mixed filters shall not exceed 8.

8.4.46.8 VLAN Mode Support field

The VLAN Mode Support field indicates various modes supported by the implementation. The format of field is defined in Table 89.

Table 89 – VLAN Mode Support bit definitions

Bit Position	Field Description	Value Description
0	VLAN only	1 = VLAN shall be supported in the implementation.
1	VLAN + non-VLAN	0 = Filtering ‘VLAN + non-VLAN’ traffic is not supported in the implementation. 1 = Filtering ‘VLAN + non-VLAN’ traffic is supported in the implementation.
2	Any VLAN + non-VLAN	0 = Filtering ‘Any VLAN + non-VLAN’ traffic is not supported in the implementation. 1 = Filtering ‘Any VLAN + non-VLAN’ traffic is supported in the implementation.
3..7	Reserved	0

8.4.46.9 Channel Count field

The Channel Count field indicates the number of channels supported by the Network Controller.

8.4.47 Get Parameters command (0x17)

The Get Parameters command can be used by the Management Controller to request that the channel send the Management Controller a copy of all of the currently stored parameter settings that have been put into effect by the Management Controller, plus “other” Host/Channel parameter values that may be added to the Get Parameters Response Payload.

Table 90 illustrates the packet format for the Get Parameters command.

Table 90 – Get Parameters command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.48 Get Parameters response (0x97)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get Parameters command and send a response. As shown in Table 91, each parameter shall return the value that was set by the Management Controller. If the parameter is not supported, 0 is returned. Currently no command-specific reason code is identified for this response.

The payload length of this response packet will vary according to how many MAC address filters or VLAN filters the channel supports. All supported MAC addresses are returned at the end of the packet, without any intervening padding between MAC addresses.

MAC addresses are returned in the following order: unicast filtered addresses first, followed by multicast filtered addresses, followed by mixed filtered addresses, with the number of each corresponding to those reported through the Get Capabilities command. For example, if the interface reports four unicast filters, two multicast filters, and two mixed filters, then MAC addresses 1 through 4 are those currently configured through the interface’s unicast filters, MAC addresses 5 and 6 are those configured through the multicast filters, and 7 and 8 are those configured through the mixed filters. Similarly, if the interface reports two unicast filters, no multicast filters, and six mixed filters, then MAC addresses 1 and 2 are those currently configured through the unicast filters, and 3 through 8 are those configured through the mixed filters.

2605 **Table 91 – Get Parameters response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	MAC Address Count	Reserved		MAC Address Flags
24..27	VLAN Tag Count	Reserved	VLAN Tag Flags	
28..31	Link Settings			
32..35	Broadcast Packet Filter Settings			
36..39	Configuration Flags			
40..43	VLAN Mode	Flow Control Enable	Reserved	
44..47	AEN Control			
48..51	MAC Address 1 byte 5	MAC Address 1 byte 4	MAC Address 1 byte 3	MAC Address 1 byte 2
52..55[a]	MAC Address 1 byte 1	MAC Address 1 byte 0	MAC Address 2 byte 5	MAC Address 2 byte 4
56..59	MAC Address 2 byte 3	MAC Address 2 byte 2	MAC Address 2 byte 1	MAC Address 2 byte 0
variable	…			
	VLAN Tag 1		VLAN Tag 2	
	…			
	…		Pad (if needed)	
	Checksum			

[a] Variable fields can start at this byte offset.

2606 Table 92 lists the parameters for which values are returned in this response packet.

2607 **Table 92 – Get Parameters data definition**

Parameter Field Name	Description
MAC Address Count	The number of MAC addresses supported by the channel
MAC Address Flags	The enable/disable state for each supported MAC address See Table 93.
VLAN Tag Count	The number of VLAN Tags supported by the channel
VLAN Tag Flags	The enable/disable state for each supported VLAN Tag See Table 94.
Link Settings	The 32-bit Link Settings value as defined in the Set Link command. See Table 46.
Broadcast Packet Filter Settings	The current 32-bit Broadcast Packet Filter Settings value

Parameter Field Name	Description
Configuration Flags	See Table 95.
VLAN Mode	See Table 60.
Flow Control Enable	See Table 81.
AEN Control	See Table 40.
MAC Address 1..8	The current contents of up to eight 6-byte MAC address filter values.
VLAN Tag 1..15	The current contents of up to 15 16-bit VLAN Tag filter values
NOTE The contents of the various configuration value fields, such as MAC Address, VLAN Tags, Link Settings, and Broadcast Packet Filter Settings, shall be considered valid only when the corresponding configuration bit is set (Enabled) in the Configuration Flags field.	

2608 The format of the MAC Address Flags field is defined in Table 93.

2609 **Table 93 – MAC Address Flags bit definitions**

Bit Position	Field Description	Value Description
0	MAC address 1 status	0b = Default or unsupported or disabled 1b = Enabled
1	MAC address 2 status, or Reserved	0b = Default or unsupported or disabled 1b = Enabled
2	MAC address 3 status, or Reserved	0b = Default or unsupported or disabled 1b = Enabled
…	…	…
7	MAC address 8 status, or Reserved	0b = Default or unsupported or disabled 1b = Enabled

2610 The format of the VLAN Tag Flags field is defined in Table 94.

2611 **Table 94 – VLAN Tag Flags bit definitions**

Bit Position	Field Description	Value Description
0	VLAN Tag 1 status	0b = Default or unsupported or disabled 1b = Enabled
1	VLAN Tag 2 status, or Reserved	0b = Default or unsupported or disabled 1b = Enabled
2	VLAN Tag 3 status, or Reserved	0b = Default or unsupported or disabled 1b = Enabled
…	…	…
14	VLAN Tag 15 status, or Reserved	0b = Default or unsupported or disabled 1b = Enabled

2612 The format of the Configuration Flags field is defined in Table 95.

2613 **Table 95 – Configuration Flags bit definitions**

Bit Position	Field Description	Value Description
0	Broadcast Packet Filter status	0b = Disabled 1b = Enabled
1	Channel Enabled	0b = Disabled 1b = Enabled
2	Channel Network TX Enabled	0b = Disabled 1b = Enabled
3	Global Multicast Packet Filter Status	0b = Disabled 1b = Enabled
4..31	Reserved	Reserved

2614 ### 8.4.49 Get Controller Packet Statistics command (0x18)

2615 The Get Controller Packet Statistics command may be used by the Management Controller to request a
2616 copy of the aggregated Ethernet packet statistics that the channel maintains for its external interface to
2617 the LAN network. The statistics are an aggregation of statistics for both the host side traffic and the NC-SI
2618 Pass-through traffic.

2619 **Table 96 – Get Controller Packet Statistics command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

2620 ### 8.4.50 Get Controller Packet Statistics response (0x98)

2621 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get
2622 Controller Packet Statistics command and send the response packet shown in Table 97.

2623 The Get Controller Packet Statistics Response frame contains a set of Ethernet statistics counters that
2624 monitor the LAN traffic in the Network Controller. Implementation of the counters listed in Table 98 is

2625 optional. The Network Controller shall return any unsupported counter with a value of `0xFFFFFFFF` for
2626 32-bit counters and 0xFFFFFFFFFFFFFFFF for 64-bit counters.

2627 **Table 97 – Get Controller Packet Statistics response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Counters Cleared From Last Read (MS Bits)			
24..27	Counters Cleared From Last Read (LS Bits)			
28..35	Total Bytes Received			
36..43	Total Bytes Transmitted			
44..51	Total Unicast Packets Received			
52..59	Total Multicast Packets Received			
60..67	Total Broadcast Packets Received			
68..75	Total Unicast Packets Transmitted			
76..83	Total Multicast Packets Transmitted			
84..91	Total Broadcast Packets Transmitted			
92..95	FCS Receive Errors			
96..99	Alignment Errors			
100..103	False Carrier Detections			
104..107	Runt Packets Received			
108..111	Jabber Packets Received			
112..115	Pause XON Frames Received			
116..119	Pause XOFF Frames Received			
120..123	Pause XON Frames Transmitted			
124..127	Pause XOFF Frames Transmitted			
128..131	Single Collision Transmit Frames			
132..135	Multiple Collision Transmit Frames			
136..139	Late Collision Frames			
140..143	Excessive Collision Frames			
144..147	Control Frames Received For version 1.2, this counter may include Priority flow control packets			
148..151	64-Byte Frames Received			
152..155	65–127 Byte Frames Received			
156..159	128–255 Byte Frames Received			
160..163	256–511 Byte Frames Received			
164..167	512–1023 Byte Frames Received			

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
168..171	1024–1522 Byte Frames Received			
172..175	1523–9022 Byte Frames Received			
176..179	64-Byte Frames Transmitted			
180..183	65–127 Byte Frames Transmitted			
184..187	128–255 Byte Frames Transmitted			
188..191	256–511 Byte Frames Transmitted			
192..195	512–1023 Byte Frames Transmitted			
196..199	1024–1522 Byte Frames Transmitted			
200..203	1523–9022 Byte Frames Transmitted			
204..211	Valid Bytes Received			
212..215	Error Runt Packets Received			
216..219	Error Jabber Packets Received			
220..223	Checksum			

2628 **Table 98 – Get Controller Packet Statistics counters**

Counter Number	Name	Meaning
0	Total Bytes Received	Counts the number of bytes received
1	Total Bytes Transmitted	Counts the number of bytes transmitted
2	Total Unicast Packets Received	Counts the number of good (FCS valid) packets received that passed L2 filtering by a specific MAC address
3	Total Multicast Packets Received	Counts the number of good (FCS valid) multicast packets received
4	Total Broadcast Packets Received	Counts the number of good (FCS valid) broadcast packets received
5	Total Unicast Packets Transmitted	Counts the number of good (FCS valid) packets transmitted that passed L2 filtering by a specific MAC address
6	Total Multicast Packets Transmitted	Counts the number of good (FCS valid) multicast packets transmitted
7	Total Broadcast Packets Transmitted	Counts the number of good (FCS valid) broadcast packets transmitted
8	FCS Receive Errors	Counts the number of receive packets with FCS errors
9	Alignment Errors	Counts the number of receive packets with alignment errors
10	False Carrier Detections	Counts the false carrier errors reported by the PHY

Counter Number	Name	Meaning
11	Runt Packets Received	Counts the number of received frames that passed address filtering, were less than minimum size (64 bytes from <Destination Address> through <FCS>, inclusively), and had a valid FCS
12	Jabber Packets Received	Counts the number of received frames that passed address filtering, were greater than the maximum size, and had a valid FCS
13	Pause XON Frames Received	Counts the number of XON packets received from the network
14	Pause XOFF Frames Received	Counts the number of XOFF packets received from the network
15	Pause XOFF Frames Transmitted	Counts the number of XON packets transmitted to the network
16	Pause XOFF Frames Transmitted	Counts the number of XOFF packets transmitted to the network
17	Single Collision Transmit Frames	Counts the number of times that a successfully transmitted packet encountered a single collision
18	Multiple Collision Transmit Frames	Counts the number of times that a transmitted packet encountered more than one collision but fewer than 16
19	Late Collision Frames	Counts the number of collisions that occurred after one slot time (defined by IEEE 802.3)
20	Excessive Collision Frames	Counts the number of times that 16 or more collisions occurred on a single transmit packet
21	Control Frames Received	Counts the number of MAC control frames received that are *not* XON or XOFF flow control frames
22	64 Byte Frames Received	Counts the number of good packets received that are exactly 64 bytes (from <Destination Address> through <FCS>, inclusively) in length
23	65–127 Byte Frames Received	Counts the number of good packets received that are 65–127 bytes (from <Destination Address> through <FCS>, inclusively) in length
24	128–255 Byte Frames Received	Counts the number of good packets received that are 128–255 bytes (from <Destination Address> through <FCS>, inclusively) in length
25	256–511 Byte Frames Received	Counts the number of good packets received that are 256–511 bytes (from <Destination Address> through <FCS>, inclusively) in length
26	512–1023 Byte Frames Received	Counts the number of good packets received that are 512–1023 bytes (from <Destination Address> through <FCS>, inclusively) in length
27	1024–1522 Byte Frames Received	Counts the number of good packets received that are 1024–1522 bytes (from <Destination Address> through <FCS>, inclusively) in length
28	1523–9022 Byte Frames Received	Counts the number of received frames that passed address filtering and were greater than 1523 bytes in length

Counter Number	Name	Meaning
29	64 Byte Frames Transmitted	Counts the number of good packets transmitted that are exactly 64 bytes (from <Destination Address> through <FCS>, inclusively) in length
30	65–127 Byte Frames Transmitted	Counts the number of good packets transmitted that are 65–127 bytes (from <Destination Address> through <FCS>, inclusively) in length
31	128–255 Byte Frames Transmitted	Counts the number of good packets transmitted that are 128–255 bytes (from <Destination Address> through <FCS>, inclusively) in length
32	256–511 Byte Frames Transmitted	Counts the number of good packets transmitted that are 256–511 bytes (from <Destination Address> through <FCS>, inclusively) in length
33	512–1023 Byte Frames Transmitted	Counts the number of good packets transmitted that are 512–1023 bytes (from <Destination Address> through <FCS>, inclusively) in length
34	1024–1522 Byte Frames Transmitted	Counts the number of good packets transmitted that are 1024–1522 bytes (from <Destination Address> through <FCS>, inclusively) in length
35	1523–9022 Byte Frames Transmitted	Counts the number of transmitted frames that passed address filtering and were greater than 1523 in length
36	Valid Bytes Received	Counts the bytes received in all packets that did not manifest any type of error
37	Error Runt Packets Received	Counts the number of invalid frames that were less than the minimum size (64 bytes from <Destination Address> through <FCS>, inclusively)
38	Error Jabber Packets Received	Counts Jabber packets, which are defined as packets that exceed the programmed MTU size *and* have a bad FCS value

The Network Controller shall also indicate in the Counters Cleared from Last Read fields whether the corresponding field has been cleared by means other than NC-SI (possibly by the host) since it was last read by means of the NC-SI. Counting shall resume from 0 after a counter has been cleared. The Counters Cleared from Last Read fields format is shown in Table 99.

Currently no command-specific reason code is identified for this response.

Table 99 – Counters Cleared from Last Read Fields format

Field	Bits	Mapped to Counter Numbers
MS Bits	0..6	32..38
	7..31	Reserved
LS Bits	0..31	0..31

IMPLEMENTATION NOTE The Get Controller Packet Statistics response contains the following counters related to flow control: Pause XON Frames Received, Pause XOFF Frames Received, Pause XON Frames Transmitted, and Pause XOFF Frames Transmitted. An implementation may or may not include Priority-Based Flow Control (PFC) packets in these counters.

8.4.51 Get NC-SI Statistics command (0x19)

In addition to the packet statistics accumulated on the LAN network interface, the channel separately accumulates a variety of NC-SI specific packet statistics for the channel. The Get NC-SI Statistics command may be used by the Management Controller to request that the channel send a copy of all current NC-SI packet statistic values for the channel. The implementation may or may not include statistics for commands that are directed to the package.

Table 100 illustrates the packet format of the Get NC-SI Statistics command.

Table 100 – Get NC-SI Statistics command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.52 Get NC-SI Statistics response (0x99)

In the absence of any error, the channel shall process and respond to the Get NC-SI Statistics command by sending the response packet and payload shown in Table 101.

Table 101 – Get NC-SI Statistics response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	NC-SI Commands Received			
24..27	NC-SI Control Packets Dropped			
28..31	NC-SI Command Type Errors			
32..35	NC-SI Command Checksum Errors			
36..39	NC-SI Receive Packets			
40..43	NC-SI Transmit Packets			
44..47	AENs Sent			
48..51	Checksum			

The Get NC-SI Statistics Response frame contains a set of statistics counters that monitor the NC-SI traffic in the Network Controller. Counters that are supported shall be reset to `0x0` when entering into the Initial State and after being read. Implementation of the counters shown in Table 102 is optional. The Network Controller shall return any unsupported counter with a value of `0xFFFFFFFF`. Counters may wraparound or stop if they reach `0xFFFFFFFE`. It is vendor specific how NC-SI commands that are sent to the package ID are included in the NC-SI statistics.

Currently no command-specific reason code is identified for this response.

Table 102 – Get NC-SI Statistics counters

Counter Number	Name	Meaning
1	NC-SI Commands Received	For packets that are not dropped, this field returns the number of NC-SI Control Packets received and identified as NC-SI commands.
2	NC-SI Control Packets Dropped	Counts the number of NC-SI Control Packets that were received and dropped (Packets with correct FCS and EtherType, but are dropped for one of the other reasons listed in 6.9.1.1). NC-SI Control Packets that were dropped because the channel ID was not valid may not be included in this statistics counter.
3	NC-SI Unsupported Commands Received	Counts the number of NC-SI command packets that were received, but are not supported. (Network controller responded to the command with a Command Unsupported response code).
4	NC-SI Command Checksum Errors	Counts the number of NC-SI Control Packets that were received but dropped because of an invalid checksum (if checksum is provided and checksum validation is supported by the channel)
5	NC-SI Receive Packets	Counts the total number of NC-SI Control Packets received. This count is the sum of NC-SI Commands Received and NC-SI Control Packets Dropped.
6	NC-SI Transmit Packets	Counts the total number of NC-SI Control Packets transmitted to the Management Controller. This count is the sum of NC-SI responses sent and AENs sent.
7	AENs Sent	Counts the total number of AEN packets transmitted to the Management Controller

8.4.53 Get NC-SI Pass-through Statistics command (0x1A)

The Get NC-SI Pass-through Statistics command may be used by the Management Controller to request that the channel send a copy of all current NC-SI Pass-through packet statistic values.

Table 103 illustrates the packet format of the Get NC-SI Pass-through Statistics command.

Table 103 – Get NC-SI Pass-through Statistics command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.54 Get NC-SI Pass-through Statistics response (0x9A)

In the absence of any error, the channel shall process and respond to the Get NC-SI Pass-through Statistics command by sending the response packet and payload shown in Table 104.

Table 104 – Get NC-SI Pass-through Statistics response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..27	Pass-through TX Packets Received on NC-SI Interface (Management Controller to Network Controller)			
28..31	Pass-through TX Packets Dropped			
32..35	Pass-through TX Packet Channel State Errors			
36..39	Pass-through TX Packet Undersized Errors			
40..43	Pass-through TX Packet Oversized Errors			
44..47	Pass-through RX Packets Received on LAN Interface			
48..51	Total Pass-through RX Packets Dropped			
52..55	Pass-through RX Packet Channel State Errors			
56..59	Pass-through RX Packet Undersized Errors			
60..63	Pass-through RX Packet Oversized Errors			
64..67	Checksum			

The Get NC-SI Statistics Response frame contains a set of statistics counters that monitor the NC-SI Pass-through traffic in the Network Controller. Supported counters shall be reset to 0x0 when entering into the Initial State and after being read. Implementation of the counters shown in Table 105 is optional. The Network Controller shall return any unsupported counter with a value of 0xFFFFFFFF for 32-bit

2672 counters and 0xFFFFFFFFFFFFFFFF for 64-bit counters. Counters may wraparound or stop if they reach
2673 `0xFFFFFFFE` for 32-bit counters and 0xFFFFFFFFFFFFFFFE for 64-bit counters..

2674 **Table 105 – Get NC-SI Pass-through Statistics counters**

Counter Number	Name	Meaning
1	Total Pass-through TX Packets Received (Management Controller to Channel)	Counts the number of Pass-through packets forwarded by the channel to the LAN
2	Total Pass-through TX Packets Dropped (Management Controller to Channel)	Counts the number of Pass-through packets from the Management Controller that were dropped by the Network Controller
3	Pass-through TX Packet Channel State Errors (Management Controller to Channel)	Counts the number of egress management packets (Management Controller to Network Controller) that were dropped because the channel was in the disabled state when the packet was received
4	Pass-through TX Packet Undersized Errors (Management Controller to Channel)	Counts the number of Pass-through packets from the Management Controller that were undersized (under 64 bytes, including FCS)
5	Pass-through TX Packet Oversized Errors (Management Controller to Channel)	Counts the number of Pass-through packets from the Management Controller that were oversized (over 1522 bytes, including FCS)
6	Total Pass-through RX Packets Received On the LAN Interface (LAN to Channel)	Counts the number of Pass-through packets that were received on the LAN interface of the channel. This counter does not necessarily count the number of packets that were transmitted to the Management Controller, because some of the packets might have been dropped due to RX queue overflow.
7	Total Pass-through RX Packets Dropped (LAN to Channel)	Counts the number of Pass-through packets that were received on the LAN interface of the channel but were dropped and not transmitted to the Management Controller
8	Pass-through RX Packet Channel State Errors (LAN to Channel)	Counts the number of ingress management packets (channel to Management Controller) that were dropped because the channel was in the disabled state when the packet was received. The NC may also count packets that were dropped because the package was in the deselected state.
9	Pass-through RX Packet Undersized Errors (LAN to Channel)	Counts the number of Pass-through packets from the LAN that were undersized (under 64 bytes, including FCS)
10	Pass-through RX Packet Oversized Errors (LAN to Channel)	Counts the number of Pass-through packets from the LAN that were oversized (over 1522 bytes, including FCS)

2675 Currently no command-specific reason code is identified for this response.

2676 8.4.55 Get Package Status command (0x1B)

2677 The Get Package Status command provides a way for a Management Controller to explicitly query the
2678 status of a package. The Get Package Status command is addressed to the package, rather than to a

2679 particular channel (that is, the command is sent with a Channel ID where the Package ID subfield
2680 matches the ID of the intended package and the Internal Channel ID subfield is set to `0x1F`).

2681 Table 106 illustrates the packet format of the Get Package Status command.

2682 **Table 106 – Get Package Status packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
20..23	Checksum			
24..45	Pad			

2683 ### 8.4.56 Get Package Status response (0x9B)

2684 Currently no command-specific reason code is identified for this response (see Table 26).

2685 **Table 107 – Get Package Status response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Package Status			
24..27	Checksum			
28..45	Pad			

2686 **Table 108 – Package Status field bit definitions**

Bit Position	Field Description	Value Description
0	Hardware Arbitration Status	`0b` = Hardware arbitration is non-operational (inactive) or unsupported. NOTE This means that hardware arbitration tokens are not flowing through this NC. `1b` = Hardware arbitration is supported, active, and implemented for the package on the given system.
31..1	Reserved	Reserved

2687

2688 8.4.57 Get NC Capabilities and Settings command (0x25)

2689 The Get NC Capabilities and Settings command is sent only as a package command. It is used to
2690 discover the supported architectural and currently configured (active) parameters of the NC.

2691 Table 109 illustrates the packet format for the Get NC Capabilities and Settings command.

2692 **Table 109 – Get NC Capabilities and Settings command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

2693 8.4.58 Get NC Capabilities and Settings response (0xA5)

2694 In the absence of any errors, the package shall process and respond to the Get NC Capabilities and
2695 Settings Command and send the response packet shown in Table 110.

2696 Currently no command-specific reason code is identified for this response.

2697 **Table 110 – Get NC Capabilities and Settings response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Max Ports	Enabled Ports	Max Busses	Enabled Busses
24..27	Max PFs	Enabled PFs	MAX VFs	
28..31	Fabrics	Enabled Fabrics	Other Caps	
32..35				
36..39				
40..43				
44..47	Reserved			
48..51	Checksum			

2698 8.4.58.1 Max Ports field

2699 The Max Ports field indicates the maximum number of network ports that can be supported by the
2700 implementation (uint8).

2701

8.4.58.2 Enabled Ports field

The Enabled Ports field indicates the current number of network ports that are currently configured (uint8).

8.4.58.3 Max Busses field

The Max Busses field indicates the maximum number of PCI busses that can be supported by the implementation (uint8).

8.4.58.4 Enabled Busses field

The Enabled Ports field indicates the current number of PCI busses that are currently configured (uint8).

8.4.58.5 Max PFs field

The Max PFs field indicates the maximum number of PCI Physical Functions that can be supported by the implementation (uint8).

8.4.58.6 Enabled PFs field

The Enabled PFs field indicates the current number of PCI Physical Functions that are currently configured (uint8).

8.4.58.7 Max VFs field

The Max VFs field indicates the maximum number of PCI Virtual Functions that can be supported by the implementation (uint8).

8.4.58.8 Fabrics field

The Fabrics field indicates the network fabrics that can be supported by the implementation.

Table 111 – Fabrics field bit definitions

Bit Position	Field Description	Value Description
0	Ethernet	0b0 = Ethernet Fabric is not supported 0b1 = Ethernet Fabric is supported
1	Fibre Channel	0b0 = Fibre Channel Fabric is not supported 0b1 = Fibre Channel Fabric is supported
2	InfiniBand	0b0 = InfiniBand Fabric is not supported 0b1 = InfiniBand Fabric is supported

Bit Position	Field Description	Value Description
3..7	Reserved	Reserved

8.4.58.9 Enabled Fabrics field

The Enabled Fabrics field indicates the currently configured fabrics.

Table 112 – Enabled Fabrics field bit definitions

Bit Position	Field Description	Value Description
0	Ethernet	0b0 = Ethernet Fabric is not enabled 0b1 = Ethernet Fabric is enabled
1	Fibre Channel	0b0 = Fibre Channel Fabric is not enabled 0b1 = Fibre Channel Fabric is enabled
2	InfiniBand	0b0 = InfiniBand Fabric is not enabled 0b1 = InfiniBand Fabric is enabled
3..7	Reserved	Reserved

8.4.58.10 Other Caps field

The Other Capabilities field indicates which features of this specification the channel supports, as described in Table 113.

Table 113 – Capabilities Flags bit definitions

Bit Position	Field Description	Value Description
0..31	Reserved	Reserved

8.4.59 Set NC Configuration command (0x26)

The Set NC Configuration command allows the Management Controller to configure the number of active Physical functions and PCI (host) and network interfaces, where allowed (generally if the reported max value of the respective entity is greater than one). If the implementation or controller architecture does not allow any configuration of these parameters, this command shall not be implemented.

The values configured by this command are held by the controller and only take effect at the next PCI reset.

The Set NC Configuration command is addressed to the package, rather than to a particular channel (that is, the command is sent with a Channel ID where the Package ID subfield matches the ID of the intended package and the Internal Channel ID subfield is set to 0x1F).

Table 114 illustrates the packet format of the Set NC Configuration command.

Table 114 – Set NC Configuration command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Enable Ports	Enable Busses	Enable PFs	
20..23	Checksum			
24..45	Pad			

8.4.60 Set NC Configuration response (0xA6)

The package shall, in the absence of a checksum error or identifier mismatch, always accept the Set NC Configuration command and send a response (see Table 115).

Table 115 – Set NC Configuration response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

8.4.61 Get PF Assignment command (0x27)

The Get PF Assignment command is a Package command that allows the Management controller to receive the list of PCI Physical Functions (partitions) currently assigned to channels (network ports) in the

2758 package, their enablement state and conditionally what PCI bus they are assigned to if the NC supports
2759 multiple host interfaces.

2760 See the Set PF Assignment command description for additional information.

2761 Table 116 – Get PF Assignment Command Packet Formatillustrates the packet format of the Get PF
2762 Assignment Command.

2763 **Table 116 – Get PF Assignment Command Packet Format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Checksum (3..2)		Checksum (1..0)	
20..45	Pad			

2764 ## 8.4.62 Get PF Assignment Response (0xA7)

2765 In the absence of any errors, the channel shall process and respond to the Get PF Assignment Command
2766 and send the response packet shown in the table below.

2767 Note: Braces {} denote fields that depend on device capabilities.

2768 **Table 117 – Get PF Assignment Response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Channel 0 Function Assignment bitmap			
24..27	{Channel 1 Function Assignment bitmap}			
	…			
	{Channel *c*-1 Function Assignment bitmap}			
	Freaky Floating Function Feature - Function Port Association			
	Function Enablement bitmap			
	{ PCI Bus 0 Function Assignment bitmap}			
	{ PCI Bus 1 Function Assignment bitmap}			
	…			
	{ PCI Bus *b*-1 Function Assignment bitmap}			
	Checksum (3..2)		Checksum (1..0)	
	Pad			

2769

8.4.62.1 Channel c Function Assignment bitmap fields

The number of Channel Function Assignment bitmaps returned in the response is equal to ‘c’, the number returned in the Get NC Capabilities and Settings Command Enabled Ports field. The Channel c Function Assignment bitmaps are 32-bit fields in which each bit position corresponds to a PCI physical function in the NC on the specified channel. If the physical function is assigned to the c^{th} channel (port), even if it not currently enabled, the bit value shall be set to 0b1; otherwise, the bit is set to 0.

Table 118 – Channel c Function Assignment bitmap field

Bit Position	Field Description	Value Description
0	F0 status	0b = F0 is not assigned on the channel (port). 1b = F0 is assigned on the channel (port).
1	F1 status	0b = F1 is not assigned on the channel (port). 1b = F1 is assigned on the channel (port).
…	…	…
15	F15 status	0b = F15 is not assigned on the channel (port). 1b = F15 is assigned on the channel (port)

8.4.62.2 Function Port Association bitmap field

The Function Assignment bitmap is a 32-bit field in which each bit position corresponds to a physical function in the device.

Table 119 – Function Port Association bitmap field

Bit Position	Field Description	Value Description
0	F0 association	0b = F0 is fixed to the specified channel (port). 1b = F0 may be assigned to any channel (port).
1	F1 association	0b = F1 is fixed to the specified channel (port). 1b = F1 may be assigned to any channel (port).
…	…	…
15	F15 association	0b = F15 is fixed to the specified channel (port). 1b = F15 may be assigned to any channel (port).

8.4.62.3 Function Enablement bitmap field

The Function Assignment bitmap is a 32-bit field in which each bit position corresponds to a physical function in the NC. The number of functions shown as enabled in this field shall be equal to the number shown in the Get/Set NC Configuration command. A function may be assigned to a PCI bus and be enabled and not be assigned to a channel in some implementations.

2787 **Table 120 – Function Enablement bitmap field**

Bit Position	Field Description	Value Description
0	F0 status	0b = F0 is not enabled 1b = F0 is enabled
1	F1 status	0b = F1 is not enabled. 1b = F1 is enabled.
…	…	…
31	F31 status	0b = F31 is not enabled. 1b = F31 is enabled

2788 #### 8.4.62.4 PCI Bus b Assignment bitmap field

2789 The number of PCI Bus Assignment bitmaps returned in the response is equal to ‘b’, the number returned
2790 in the Get NC Capabilities and Settings Command Enabled Busses field. The PCI Bus b Assignment
2791 bitmaps are 32-bit fields in which each bit position corresponds to a physical function in the NC on the
2792 specified host bus. If the physical function is assigned to the b^{th} bus, even if it not currently enabled, the
2793 bit value shall be set to 0b1, otherwise the bit is set to 0.

2794 **Table 121 – PCI Bus b Assignment bitmap field**

Bit Position	Field Description	Value Description
0	F0 status	0b = F0 is not assigned on the specified PCI Bus. 1b = F0 is assigned on the specified PCI Bus.
1	F1 status	0b = F1 is not assigned on the specified PCI Bus. 1b = F1 is assigned on the specified PCI Bus.
…	…	…
31	F15 status	0b = F31 is not assigned on the specified PCI Bus. 1b = F31 is assigned on the specified PCI Bus

2795 ### 8.4.62.5 Calculation of Partition ID

2796 When multiple functions are assigned to a channel, they are addressed by a value called the Partition ID.
2797 The Partition ID is created by taking the set of Functions that are assigned to a channel and assigning
2798 each an index value starting with the lowest numbered Function. A Function assigned to a channel has a
2799 Partition ID even if it is not enabled. Partition numbering starts at 1. For example, if F2 and F6 are
2800 assigned to channel 3, but only F2 is enabled, then F2 has Partition ID = 1 and F6 has Partition ID = 2 on
2801 that channel.

8.4.63 Set PF Assignment command (0x28)

The Set PF Assignment command is a Package command that allows the Management controller to enable, disable, and assign PCI Physical Functions (partitions) in the controller to the channels (network ports), and, if applicable, to different PCI buses in multi-home or multi-host configurations.

The format of the command payload is dependent on the numbers of Physical Functions, Channels and PCI Buses supported by the controller:

1) The number of Function Assignments bitmap fields shall be determined by the value (c) of the Channel Count field in the Get Capabilities response.

4) The number of Physical Functions allowed to be configured in the Function Assignment and Enablement bitmap fields shall be determined by the value of the <Physical Function Count> field in the <Get NC Capabilities and Settings command> response. Assignment in all bitmaps starts at bit 0 and continues sequentially for the number of Functions supported. To support various implementation architectures, the definition of assignment/enablement rules is beyond the scope of this specification.

5) If the value (b) of the <PCI Bus Count> field in the <Get Device Capabilities and Settings command> response is greater than 1, the Controller shall also include that number of PCI Bus Function Assignment bitmap fields in the command. Controllers that do not support multiple PCI interfaces shall not implement PCI Bus Host Function Assignment bitmap fields. PCI Bus 0 shall be used if the Controller is configured for single bus operation.

The values configured by this command are held by the controller and only take effect at the next PCI reset.

Table 122 illustrates the packet format of the Set PF Assignment Command.

Table 122 – Set PF Assignment Command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Channel 0 Function Assignment bitmap			
	{Channel 1 Function Assignment bitmap}			
	…			
	{Channel *c*-1 Function Assignment bitmap}			
	Function Enablement bitmap			
	{ PCI Bus 0 Function Assignment bitmap}			
	{ PCI Bus 1 Function Assignment bitmap}			
	…			
	{ PCI Bus *b*-1 Function Assignment bitmap}			
F20..23	Checksum (3..2)		Checksum (1..0)	
24..27	Pad			

8.4.63.1 Channel Function Assignment bitmap field

The Channel Function Assignment bitmap is a 32-bit field in which each bit position corresponds to a physical function in the device. If the physical function is assigned to the channel (port), even if it not

currently enabled, the bit value shall be set to 0b1. This allows for a partition ID to be assigned and partition commands to be sent to the function even if it is not enabled.

Table 123 – Channel Function Assignment bitmap field

Bit Position	Field Description	Value Description
0	F0 status	0b = F0 is not assigned on the channel (port). 1b = F0 is assigned on the channel (port).
1	F1 status	0b = F1 is not assigned on the channel (port). 1b = F1 is assigned on the channel (port).
…	…	…
15	F15 status	0b = F15 is not assigned on the channel (port). 1b = F15 is assigned on the channel (port)

8.4.63.2 Function Enablement bitmap field

The Function Assignment bitmap is a 32-bit field in which each bit position corresponds to a physical function in the device.

Table 124 – Function Enablement bitmap field

Bit Position	Field Description	Value Description
0	F0 status	0b = F0 is not enabled on the specified channel (port). 1b = F0 is enabled on the specified channel (port).
1	F1 status	0b = F1 is not enabled on the specified channel (port). 1b = F1 is enabled on the specified channel (port).
…	…	…
15	F15 status	0b = F15 is not enabled on the specified channel (port). 1b = F15 is enabled on the specified channel (port)

8.4.63.3 PCI Bus Assignment bitmap field

The PCI Bus Assignment bitmap is a 32-bit field in which each bit position corresponds to a physical function in the device.

Table 125 – PCI Bus Assignment bitmap field

Bit Position	Field Description	Value Description
0	F0 status	0b = F0 is not assigned on the specified PCI Bus. 1b = F0 is assigned on the specified PCI Bus.
1	F1 status	0b = F1 is not assigned on the specified PCI Bus. 1b = F1 is assigned on the specified PCI Bus.

Bit Position	Field Description	Value Description
…	…	…
15	F15 status	0b = F15 is not assigned on the specified PCI Bus. 1b = F15 is assigned on the specified PCI Bus

8.4.64 Set PF Assignment Response (0xA8)

In the absence of any errors, the channel shall process and respond to the Get PF Assignment Command and send the response packet shown in Table 126.

Table 126 – Set PF Assignment Response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
24..27	Checksum (3..2)		Checksum (1..0)	
36..39	Pad			

8.4.65 Get Port Configuration command (0x29)

The Get Port Configuration command is used to discover additional optional functions supported by the channel, such as the number of unicast/multicast addresses supported, the amount of buffering in bytes available for packets bound for the Management Controller, and so on.

Table 127 illustrates the packet format for the Get Port Configuration command.

Table 127 – Get Port Configuration command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.66 Get Port Configuration response (0xA9)

In the absence of any errors, the channel shall process and respond to the Get Port Configuration Command and send the response packet shown in Table 128.

Currently no command-specific reason code is identified for this response.

2855 **Table 128 – Get Port Configuration response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Fabric Type	QoS Type	Max MTU?	Media Type
24..27	bits		Reserved	# Enabled Partitions
28..31	P1 Max TX BW	P1 Min TX BW	P2 Max TX BW	P2 Min TX BW
32..35	…			
36..39				
48..51	Checksum			

2856 ### 8.4.66.1 Media Type field

2857 The Media Type field indicates the physical interface type used on the port implementation and if that port
2858 supports one or more than one NC-SI channels, as described in **Error! Reference source not found.**.

2859 NOTE An implementation that implements a SFF cage interface into which a RJ-45 transceiver is plugged shall
2860 return 'SFF cage' as the media type.

2861 **Table 129 – Media Type bit definitions**

Bit Position	Field Description	Value Description
0	Backplane	`0b` = The port does not have a backplane interface `1b` = The port has a backplane interface
1	Base-T (RJ-45 style)	`0b` = The port does not have a Base-T interface `1b` = The port has a Base-T (RJ-45 style) interface
2	SFF cage	`0b` = The port does not have a SFF style interface `1b` = The port has a SFF style interface
3..6	Reserved	Reserved
7	Shared Interface	0b = The port is dedicated to one NC-SI channel 1b = The port is shared between multiple channels

2862

2863 ### 8.4.66.2 bits field

2864 The bits field indicates which features of this specification the channel supports, as described in Table
2865 130.

Table 130 – bits field definitions

Bit Position	Field Description	Value Description
0	TBD	
1		
2		
3		
4		
6..5		
7..31	Reserved	Reserved

8.4.66.3 P(n) Max TX BW Fields

These fields contain the Maximum TX bandwidth allocation of the n^{th} enabled partition expressed in % of the physical port link speed.

8.4.66.4 P(n) Min TX BW Fields

These fields contain the Minimum TX bandwidth allocation of the n^{th} enabled partition expressed in % of the physical port link speed.

8.4.67 Set Port Configuration command (0x2A)

The Set Port Configuration command allows the Management Controller to configure characteristics of the port.

Table 131 illustrates the packet format of the Set Port Configuration command.

Table 131 – Set Port Configuration command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Fabric Type	QoS Type?	MTU?	Media Type
	bits			# Partitions
	P1 Max TX BW	P1 Min TX BW	P2 Max TX BW	P2 Min TX BW
	…			
20..23	Checksum			
24..45	Pad			

8.4.67.1 Fabric Type field

The Fabric Type field indicates which personality types are currently enabled on the partition, as described in Table 132.

2881 **Table 132 – Fabric Type bit definitions**

Bit Position	Field Description	Value Description
0	Ethernet	0b = Ethernet operation is not enabled 1b = Ethernet operation is enabled
1	Fibre Channel	0b = Fibre Channel operation is not enabled 1b = Fibre Channel operation is enabled
2	InfiniBand Status	0b = InfiniBand operation is not enabled 1b = InfiniBand operation is enabled
3..7	Reserved	Reserved

2882 ### 8.4.67.2 QoS Type field

2883 The QoS Type field indicates which QoSypes are currently enabled on the partition, as described in Table
2884 133.

2885 **Table 133 – QoS Type bit definitions**

Bit Position	Field Description	Value Description
0	TBD	
1		
2		
3		
4		
5		
6..7	Reserved	Reserved

2886 ### 8.4.67.3 MTU field

2887 The MTU field is used to configure the maximum allowed MTU size on the port.

2888 Table 134 describes the values for the bits field.

2889 **Table 134 – Values for the bits field (8-bit field)**

Value	Description
0x0	TBD
0x1	
0x2	
0x3	
0x4..0xFF	

8.4.67.4 # Partitions

The Number of Partitions field indicates the number of Functions that have been assigned to the channel/port in the Set PF Assignment command. Each assigned partition must be allocated min and max TX bandwidth when enabled.

The initial value is generally expected to be one partition enabled per port and if modified, the new value should persist across system boot and power cycles.

8.4.67.5 P(n) Max TX BW fields

These fields contain the Maximum TX bandwidth allocation of the n^{th} enabled partition expressed in % of the physical port link speed. Oversubscription of partition maximum bandwidth is allowed. The field value is an integer ranging from 0 to $100_{10,}$ expressed as a hexadecimal value.

8.4.67.6 P(n) Min TX BW field

These fields contain the Minimum TX bandwidth allocation of the n^{th} enabled partition expressed in % of the physical port link speed. This is interpreted as committed bandwidth to the partition and as such the Min TX BW fields of all enabled partitions on the port must sum to 100%. The field value is an integer ranging from 0 to $100_{10,}$ expressed as a hexadecimal value.

8.4.68 Set Port Configuration response (0xAA)

The package shall, in the absence of a checksum error or identifier mismatch, always accept the Set Port Configuration command and send a response (see Table 135).

Table 135 – Set Port Configuration response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

Table 136 describes the reason code that is specific to the Set Port Configuration command.

Table 136 – Set Port Configuration command-specific reason code

Value	Description	Comment
0x1409		Returned when the implementation requires that both

8.4.69 Get Partition Configuration command (0x2B)

The Get Partition Configuration command is used to discover additional optional functions supported by the channel, such as the number of unicast/multicast addresses supported, the amount of buffering in bytes available for packets bound for the Management Controller, and so on.

Table 137 illustrates the packet format for the Get Partition Configuration command.

Table 137 – Get Partition Configuration command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Partition ID	Reserved		
20..23	Checksum			
24..45	Pad			

8.4.69.1 Partition ID field

The Partition ID field is the identifier for the function on the channel as defined in clause 8.4.63

8.4.70 Get Partition Configuration response (0xAB)

In the absence of any errors, the channel shall process and respond to the Get Partition Configuration Command and send the response packet shown in Table 138.

Currently no command-specific reason code is identified for this response.

Table 138 – Get Partition Configuration response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Personality Cfg	Personality Spt	Configuration Flags	
24..27	Max TX BW	Min TX BW	Advertised VF Count	
28..31	PCI DID		PCI VID	
32..35	PCI SSID		PCI SVID	
36..39	FCoE Cfg	Address Count	Address TLVs	
40..43				
44..47	Reserved			
48..51	Checksum			

8.4.70.1 Personality Cfg field

The Personality Configured field indicates which personality type(s) are currently enabled on the partition, as described in Table 139.

Note: Some implementations may support multiple personalities being simultaneously enabled.

Table 139 – Personality Cfg bit definitions

Bit Position	Field Description	Value Description
0	Ethernet Status	0b = Ethernet operation is not enabled 1b = Ethernet operation is enabled
1	Fibre Channel Status	0b = Fibre Channel operation is not enabled 1b = Fibre Channel operation is enabled
2	Fibre Channel over Ethernet Status	0b = Fibre Channel over Ethernet operation is not enabled 1b = Fibre Channel over Ethernet operation is enabled
3	InfiniBand Status	0b = InfiniBand operation is not enabled 1b = InfiniBand operation is enabled
4	iSCSI Offload Status	0b = iSCSI Offload operation is not enabled 1b = iSCSI Offload operation is enabled
5	RDMA Status	0b = RDMA operation is not enabled 1b = RDMA operation is enabled
6..7	Reserved	Reserved

8.4.70.2 Personality Spt field

The Personality Supported field indicates which personality types the partition supports, as described in Table 140.

Table 140 – Personality Spt bit definitions

Bit Position	Field Description	Value Description
0	Ethernet Support	0b = Ethernet operation is not supported 1b = Ethernet operation is supported
1	Fibre Channel Support	0b = Fibre Channel operation is not supported 1b = Fibre Channel operation is supported
2	Fibre Channel over Ethernet Support	0b = Fibre Channel over Ethernet operation isn't supported 1b = Fibre Channel over Ethernet operation is supported
3	InfiniBand Support	0b = InfiniBand operation is not supported 1b = InfiniBand operation is supported
4	iSCSI Offload Support	0b = iSCSI Offload operation is not supported 1b = iSCSI Offload operation is supported
5	RDMA Support	0b = RDMA operation is not supported 1b = RDMA operation is supported
6..7	Reserved	Reserved

8.4.70.3 Configuration Flags field

The Configuration Flags field indicates which optional features of this specification the channel supports, as described in Table 141.

Table 141 – Configuration Flags bit definitions

Bit Position	Field Description	Value Description
0	Host Driver Status	0b = When reporting is supported, Host driver is not present 1b = When reporting is supported, Host driver is present
1	Host Driver Status Reporting	0b = Host Driver status is not supported. 1b = Host Driver status is supported.
2	Partition Link Status	0b = When reporting is supported, Partition Link is down 1b = When reporting is supported, Partition Link is up
3	Partition Link Status Reporting	0b = Partition Link Status is not supported. 1b = Partition Link Status is supported.
4	Boot Status	0b = The partition is not configured for boot. 1b = The partition is configured for boot.
5	Bootable	0b = The partition supports boot 1b = The partition does not support boot
7..31	Reserved	Reserved

8.4.70.4 Partition Link fields

This fields describe the ability of a partition to support traffic when the partition is assigned to a PCI bus and NC-SI channel and either its associated physical port link is up or the implementation supports internal communication between partitions when the physical port link is down.

8.4.70.5 Max TX BW field

This field contains the Maximum TX bandwidth allocation of the partition expressed in % of the physical port link speed. The % value ranges from 0 to 100 represented in hexadecimal.

8.4.70.6 Min TX BW field

This field contains the Minimum TX bandwidth allocation of the partition expressed in % of the physical port link speed. This is interpreted as committed bandwidth to the partition and as such the Min TX BW fields of all enabled partitions on the port must sum to 100%. The % value ranges from 0 to 100 represented in hexadecimal.

8.4.70.7 Advertised VF Count field

The Advertised VF Count field indicates the number of Virtual Functions that are advertised by the partition's PF.

2955 #### 8.4.70.8 FC/FCoE stuff

2956 This field contains nothing right now.

2957 #### 8.4.70.9 Address Count field

2958 This field indicates the number of permanent and virtual addresses reported by the partition.

2959

2960 #### 8.4.70.10 Address TLVs

2961 These TLVs show the permanently programmed and current addresses being used by the partition.

2962

2963

2964 **Table 142 – Address Type-Length Field Bit Definitions**

Bit Position	Field Description	Value Description
7..0	Address Type	The following type encodings shall be used to indicate the address values that are permanently assigned to the partition. The response shall include all types whether or not that mode of operation is active, or the partition is enabled: 0x0 = Reserved 0x1 = Ethernet MAC 0x2 = iSCSI Offload (Ethernet MAC) 0x3 = Fibre Channel World Wide Node Name 0x4 = Fibre Channel World Wide Port Name 0x5 = FCoE-FIP MAC 0x6 = InfiniBand Node GUID 0x7 = InfiniBand Port GUID 0x8 = InfiniBand VPort/LID The following type encodings shall be used to indicate all address values that are currently in use by the partition based on configured mode of operation. These may be the permanent address or a programmatically assigned address. : : 0xF1 = Ethernet MAC 0xF2 = iSCSI Offload (Ethernet MAC) 0xF3 = Fibre Channel World Wide Node Name 0xF4 = Fibre Channel World Wide Port Name 0xF5 = FCoE-FIP MAC 0xF6 = InfiniBand Node GUID 0xF7 = InfiniBand Port GUID 0xF8 = InfiniBand VPort/LID all others = Reserved
15..8	Address Length	The length indicates the number of bytes used in the address

2965 ### 8.4.71 Set Partition Configuration command (0x2C)

2966 The Set Partition Configuration command allows the Management Controller to configure various settings
2967 of the partition including virtual addresses, VF allocation and other parameters.

2968 The Set Partition Configuration command is addressed to the channel with the Partition ID field set to the
2969 index/ordinal of the target PF on the channel.

Table 143 illustrates the packet format of the Set Partition Configuration command.

Table 143 – Set Partition Configuration command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Partition ID	Personality Cfg	VF Count	
	Config flags	FCoE Cfg	Address Count	Address TLV
20..23	Reserved Checksum			
24..45	Pad			

8.4.71.1 Personality Cfg field

The Personality Configuration field indicates which personality type(s) shall be enabled on the partition, as described in Table 144. Any attempt to enable a personality not shown as supported in clause 8.4.70.2 shall be cause the command to fail. In some implementations it may be appropriate to select more than one personality at a time, for instance Ethernet and RDMA.

Table 144 – Personality Cfg bit definitions

Bit Position	Field Description	Value Description
0	Ethernet Status	`0b` = Ethernet operation is not enabled `1b` = Ethernet operation is enabled
1	Fibre Channel Status	`0b` = Fibre Channel operation is not enabled `1b` = Fibre Channel operation is enabled
2	Fibre Channel over Ethernet Status	`0b` = Fibre Channel over Ethernet operation is not enabled `1b` = Fibre Channel over Ethernet operation is enabled
3	InfiniBand Status	`0b` = InfiniBand operation is not enabled `1b` = InfiniBand operation is enabled
4	iSCSI Offload Status	`0b` = iSCSI Offload operation is not enabled `1b` = iSCSI Offload operation is enabled
5	RDMA Status	`0b` = RDMA operation is not enabled `1b` = RDMA operation is enabled
6..7	Reserved	Reserved

8.4.71.2 VF Count

The VF Count field contains the number of VFs advertised in PCI Configuration Space by the partition.

8.4.71.3 Config flags

Table 145 describes the values for the Config flags field.

Table 145 – Values for the Config flags field (8-bit field)

Value	Description
0x0	TBD
0x1	
0x2	
0x3	
0x4..0xFF	Reserved

8.4.71.4 FCoE Config field

The FCoE Config field contains FCoE configuration details. For partitions that do not support FCoE, it shall be written as 0x00.

Table 146 – FCoE Configuration field

Bit Position	Field Description	Value Description
7..0	Address Type	TBD all others = Reserved
15..8	Address Length	The length indicates the number of bytes used in the address

8.4.71.5 Address Count field

The Address Count field contains the number of partition virtual addresses to be configured as specified in the Address TLV field.

2990 ### 8.4.71.6 Address TLV

2991 **Table 147 – Address Type-Length field bit definitions**

Bit Position	Field Description	Value Description
7..0	Address Type	The expedited use of the addresses specified herein is to override the permanent or factory network address to be used by the partition based on configured mode of operation. To return to using the permanent address, supply either an address of 0 or the permanent address in this field. : : 0xF1 = Ethernet MAC 0xF2 = iSCSI Offload (Ethernet MAC) 0xF3 = Fibre Channel World Wide Node Name 0xF4 = Fibre Channel World Wide Port Name 0xF5 = FCoE-FIP MAC 0xF6 = InfiniBand Node GUID 0xF7 = InfiniBand Port GUID 0xF8 = InfiniBand VPort/LID all others = Reserved
15..8	Address Length	The length indicates the number of bytes used in the address

2992 ### 8.4.72 Set Partition Configuration response (0xAC)

2993 The package shall, in the absence of a checksum error or identifier mismatch, always accept the Set
2994 Partition Configuration command and send a response (see Table 148).

2995 **Table 148 – Set Partition Configuration response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Checksum			
24..45	Pad			

2996 Table 149 describes the reason code that is specific to the Set Partition Configuration command.

2997 **Table 149 – Set Partition Configuration command-specific reason code**

Value	Description	Comment
0x	TBD	

8.4.73 Get Boot Config Command (0x2D)

The Get Boot Config Command allows the Management Controller to query for the Boot Initiator settings of a given Boot Protocol type configured on the channel/PF/partition and stored in the NVRAM of the controller.

If the command is sent to a destination that exists but that does not support the specified Boot Protocol type, the command execution shall fail with a response indicating an unsupported command.

Table 150 illustrates the packet format of the Get Boot Config command.

Table 150 – Get Boot Config command packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Reserved			
20..23				Protocol Type
20..23	Checksum (3..2)		Checksum (1..0)	
24..45	Pad			

8.4.73.1 Protocol Type field

The Protocol Type field specifies the boot protocol for which configuration data is requested.

Table 151 – Protocol Type field

Bit Position	Field Description	Value Description
7..0	Boot Protocol Type	`0x0` = PXE `0x1` = iSCSI `0x2` = FCoE `0x3` = FC `0x4` = NVMeoFC `0x??-0xFF` = Reserved

8.4.74 Get Boot Config Response (0xAD)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get Boot Config command and send a response.

The Get Boot Config Response frame contains the currently stored settings for the specified Boot Protocol type contained in the controller's NVRAM that the channel/PF/partition will use in a boot operation done locally by the adapter. Settings that the Controller supports but does not have a value for (e.g., have no initial or current value) should be included in the Response and have a length of 0.

All attribute values returned by this command shall be in unterminated ASCII string format.

Table 152 illustrates the packet format of the Get Boot Config Response.

Table 152 – Get Boot Config Response packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23			Protocol Type	Number of TLVs
28..	Type-Length Field #1		Value Field #1	
…	Type-Length Field #2		Value Field #2	
…	…			
….	Checksum (3..2)		Checksum (1..0)	

8.4.74.1 Protocol Type field

The Protocol Type field specifies the boot protocol for which boot attributes are being returned..

Table 153 – Protocol Type field

Bit Position	Field Description	Value Description
7..0	Boot Protocol Type	0x0 = PXE 0x1 = iSCSI 0x2 = FCoE 0x3 = FC 0x4 = NVMeoFC 0x??-0xFF = Reserved

8.4.74.2 Boot Protocol Type-Length-Value fields

The set of boot attributes (one of the following 4 tables) that correspond to the specified Protocol Type in the Command are returned as TLVs in the Response.

3026 **Table 154 – PXE Boot Protocol Type-Length field**

Bit Position	Field Description	Value Description
7..0	Attribute Name/Type	0x0 = VLAN ID 0x1 = VLAN enable 0x2 = 0x??-0xFF = Reserved
15..8	Length	
	Attribute Value	Value data

3027

3028 **Table 155 – iSCSI Boot Protocol Type-Length field**

Bit Position	Field Description	Value Description
7..0	Attribute Name/Type	0x0 = IscsiInitiatorIPAddrType 0x1 = IscsiInitiatorAddr 0x2 = IscsiInitiatorName 0x3 = IscsiInitiatorSubnet 0x4 = IscsiInitiatorSubnetPrefix 0x5 = IscsiInitiatorGateway 0x6 = IscsiInitiatorFirstDNS 0x7 = IscsiInitiatorSecondDNS 0x10 = ConnectFirstTgt 0x11 = FirstTgtIpAddress 0x12 = FirstTgtTcpPort 0x13 = FirstTgtBootLun 0x14 = FirstTgtIscsiName 0x15 = FirstTgtChapId 0x16 = FirstTgtChapPwd 0x17 = FirstTgtVLANEnable *bool 0x18 = FirstTgtVLAN 0x20 = ConnectSecondTgt 0x21 = SecondTgtIpAddress 0x22 = SecondTgtTcpPort 0x23 = SecondTgtBootLun 0x24 = SecondTgtIscsiName 0x25 = SecondTgtChapId 0x26 = SecondTgtChapPwd 0x27 = SecondTgtVLANEnable *bool 0x28 = SecondTgtVLAN 0x??-0xFF = Reserved
15..8	Length	
	Attribute Value	Value data

3029 **Table 156 – Get FC Boot Protocol Type-Length field**

Bit Position	Field Description	Value Description
7..0	Attribute Name/Type	0x0 = FCInitiatorBootSelection 0x1 = FirstFCTargetWWPN 0x2 = FirstFCTargetLUN 0x3 = SecondFCTargetWWPN 0x4 = SecondFCTargetLUN 0x5-0xF = Reserved
15..8	Length	
	Attribute Value	Value data

3030 **Table 157 – FCoE Boot Protocol Type-Length field**

Bit Position	Field Description	Value Description
7..0	Attribute Name/Type	0x0 = FCoEInitiatorBootSelection 0x1 = FirstFCoEWWPNTarget 0x2 = FirstFCoEBootTargetLUN 0x3 = FirstFCoEFCFVLANID 0x4 = FCoETgTBoot 0x5-0xF = Reserved
15..8	Length	
	Attribute Value	Value data

8.4.75 Set Boot Config command (0x2E)

The Set Boot Config command allows the Management Controller to send to the channel/PF/partition the Boot settings to be used by the channel/PF/partition in conducting boot operations of the specified type.

The Network Controller shall apply the attribute values in the order received in this command (e.g., TLV1 before TLV2, etc.) so that any dependency relationships are maintained.

See the Get Boot Config Command for the definition of the **command** fields.

All string values specified in this command shall be in unterminated ASCII string format.

A NC that does not support or is not in partitioning mode shall have the Partition ID field programed as 0x00.

A TLV length value of 0 indicates the clearing of the current value of the attribute to null or no value.

A maximum of 32 TLVs may be sent in any one instance of the Set Boot Config command.

If the command is sent to a destination that exists but that does not support the specified Boot Protocol type, the command execution shall fail with a response indicating an unsupported command.

Table 158 – Set Boot Config command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Partition ID		Protocol Type	Number of TLVs
24..	Type-Length Field #1.		Value Field #1.	
….	Type-Length Field #2		Value Field #2	
….	**….**			
….	Checksum (3..2)		Checksum (1..0)	
….	Pad			

8.4.76 Set Boot Config Response (0xAE)

In the absence of any errors, the channel shall process and respond to the Set Boot Config command and send the response packet shown in Table 159.

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Set Boot Config command and send a response.

If the command is sent to a destination that exists but that does not support the specified Boot Protocol type, the command response shall indicate an unsupported command.

If there are errors in any of the TLVs included in the Set command, the entire command is deemed to fail, and no configuration changes are to be made by the controller. The TLV Error Reporting field shall be used to provide individual status reporting on the TLVs received.

3055 **Table 159 – Set Boot Config Response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	TLV Error Reporting			
28..31	Checksum (3..2)		Checksum (1..0)	
32..45	Pad			

3056 ### 8.4.76.1 TLV Error Reporting field

3057 The TLV Error Reporting field is a bit-map indicating which TLVs were processed successfully and which
3058 were not in the incoming Set command. The bit order corresponds to the order of TLVs in the incoming
3059 Set command. There is a 1:1 correspondence between incoming TLVs and the active bits in this field. If
3060 fewer than 32 TLVs are transmitted, the bits corresponding to the unsent TLVs shall be set to 0.

3061 **Table 160 – TLV Error Reporting field**

Bit Position	Field Description	Value Description
0	TLV0 status	`0b` = 0 No error detected in TLV0 `0b` = 1 Error detected in TLV0
1	TLV1 status	`1b` = 0 No error detected in TLV1 or TLV1 not present `1b` = 1 Error detected in TLV1
		`0x??-0xFF` = Reserved

3062

3063 ### 8.4.77 Get Partition Statistics command (`0x2F`)

3064 The Get Partition Statistics command is used to retrieve network statistics relevant to the partition from
3065 the NC. For example, the MC should only request Ethernet statistics from a partition configured for
3066 Ethernet operation. The defined responses are customized for each personality type.

3067 Implementation of this command is conditional and is required only for NCs that support partitioning.
3068 Implementation of each response type is conditional based on the NC supporting the specified type of
3069 operation on the partition.

3070 The NC shall return in the response a value of 0xFFFFFFFF for unsupported 32-bit counters and
3071 0xFFFFFFFFFFFFFFFF for unsupported 64-bit counters.

3072 As the intent of the command is to retrieve live statistics from enabled partitions, if the command is sent to
3073 a Partition ID that doesn't exist in the current configuration or if the Stats type does not match the
3074 configured personality of the partition, the command shall fail with the Parameter is Invalid reason code.

3075

3076 **Table 157 – Get Partition Statistics command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Partition ID	Reserved		Stats Type
20..23	Checksum			
24..45	Pad			

3077 ### 8.4.77.1 Stats Type field

3078 The Stats Type field is the identifier for the type of statistics to be queried

3079 **Table 158 – Stats Type Field**

3080

Bit Position	Field Description	Value Description
7..0	Stats Type	0x01 = Ethernet 0x02 = iSCSI 0x04 = FCoE 0x08 = RDMA 0x10 = IB All others = Reserved

3081

3082 ## 8.4.78 Get Partition Statistics response for Ethernet (0xAF)

3083 In the absence of any errors, the channel shall process and respond to the Get Partition Statistics
3084 Command and send the response packet shown below when the Stats Type indicates Ethernet.

3085 Currently no command-specific reason code is identified for this response.

3086 **Table 161– Get Partition Statistics (Ethernet) response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Stats Type	Counter Sizes	Counters Cleared	
24..27	Total Bytes Received (upper)			
28..31	Total Bytes Received (lower)			
32..35	Total Bytes Transmitted (upper)			
36..39	Total Bytes Transmitted (lower)			
40..43	Total Unicast Packets Received			
44..47	Total Multicast Packets Received			
48..51	Total Broadcast Packets Received			
52..55	Total Unicast Packets Transmitted			
56..59	Total Multicast Packets Transmitted			
60..63	Total Broadcast Packets Transmitted			
64..67	Total Unicast Bytes Received (upper)			
68..71	Total Unicast Bytes Received (lower)			
72..75	Total Multicast Bytes Received (upper)			
76..79	Total Multicast Bytes Received (lower)			
80..83	Total Broadcast Bytes Received (upper)			
84..87	Total Broadcast Bytes Received (lower)			
88..91	Total Unicast Bytes Transmitted (upper)			
92..95	Total Unicast Bytes Transmitted (lower)			
96..99	Total Multicast Bytes Transmitted (upper)			
100..103	Total Multicast Bytes Transmitted (lower)			
104..107	Total Broadcast Bytes Transmitted (upper)			
108..111	Total Broadcast Bytes Transmitted (lower)			
112..115	Checksum			

3087

3088 **8.4.78.1 Counter Sizes field**

3089 The NC shall indicate in the Counter Sizes field whether the implementation uses 32-bit counters or 64-bit
3090 counters in those counter fields above that are defined as 64-bit.

3091 **Table 160 – Counter Sizes field format**

Bit Position	Field Description	Value Description
0	Total Bytes Received	0b = 32-bit 1b = 64-bit
1	Total Bytes Transmitted	0b = 32-bit 1b = 64-bit
2	Total Unicast Bytes Received	0b = 32-bit 1b = 64-bit
3	Total Multicast Bytes Received	0b = 32-bit 1b = 64-bit
4	Total Broadcast Bytes Received	0b = 32-bit 1b = 64-bit
5	Total Unicast Bytes Transmitted	0b = 32-bit 1b = 64-bit
6	Total Multicast Bytes Transmitted	0b = 32-bit 1b = 64-bit
7	Total Broadcast Bytes Transmitted	0b = 32-bit 1b = 64-bit

3092

3093 #### 8.4.78.2 Counters Cleared from Last Read field

3094 The NC shall indicate in the Counters Cleared from Last Read field whether the corresponding fields have
3095 been cleared since they were last read over NC-SI.

3096 **Table 162 – Counters Cleared from Last Read field format**

Bit Position	Field Description	Value Description
0	Total Bytes Received	0b = Not Cleared 1b = Cleared
1	Total Bytes Transmitted	0b = Not Cleared 1b = Cleared
2	Total Unicast Packets Received	0b = Not Cleared 1b = Cleared
3	Total Multicast Packets Received	0b = Not Cleared 1b = Cleared
4	Total Broadcast Packets Received	0b = Not Cleared 1b = Cleared
5	Total Unicast Packets Transmitted	0b = Not Cleared 1b = Cleared
6	Total Multicast Packets Transmitted	0b = Not Cleared 1b = Cleared

Bit Position	Field Description	Value Description
7	Total Broadcast Packets Transmitted	0b = Not Cleared 1b = Cleared
8	Total Unicast Bytes Received	0b = Not Cleared 1b = Cleared
9	Total Multicast Bytes Received	0b = Not Cleared 1b = Cleared
10	Total Broadcast Bytes Received	0b = Not Cleared 1b = Cleared
11	Total Unicast Bytes Transmitted	0b = Not Cleared 1b = Cleared
12	Total Multicast Bytes Transmitted	0b = Not Cleared 1b = Cleared
13	Total Broadcast Bytes Transmitted	0b = Not Cleared 1b = Cleared
15..14	Reserved	

8.4.79 Get Partition Statistics response for FCoE (0xAF)

In the absence of any errors, the channel shall process and respond to the Get Partition Statistics Command and send the response packet shown below when the Stats Type indicates FCoE.

Currently no command-specific reason code is identified for this response.

Table 163 – Get Partition Statistics (FCoE) response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Stats Type	Reserved	Counters Cleared	
24..27	Total FCoE Bytes Received			
28..31	Total FCoE Bytes Received			
32..35	Total FCoE Packets Transmitted			
36..39	Total FCoE Packets Transmitted			
40..43	Total FCoE Packets Received			
44..47	Total FCoE Packets Transmitted			
48..51	Checksum			

3104 #### 8.4.79.1 Counters Cleared from Last Read

3105 The NC shall indicate in the Counters Cleared from Last Read field whether the corresponding fields have
3106 been cleared since they were last read over NC-SI.

3107 **Table 164 – Counters Cleared from Last Read field format**

Bit Position	Field Description	Value Description
0	Total FCoE Bytes Received	0b = Not Cleared 1b = Cleared
1	Total FCoE Packets Transmitted	0b = Not Cleared 1b = Cleared
2	Total FCoE Packets Received	0b = Not Cleared 1b = Cleared
3	Total FCoE Packets Transmitted	0b = Not Cleared 1b = Cleared
15..4	Reserved	

3108

3109 ### 8.4.80 Get Partition Statistics response for iSCSI (0xAF)

3110 In the absence of any errors, the channel shall process and respond to the Get Partition Statistics
3111 Command and send the response packet shown below when the Stats Type indicates iSCSI.

3112 Currently no command-specific reason code is identified for this response.

3113 **Table 165 – Get Partition Statistics (iSCSI) response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Stats Type	Reserved	Counters Cleared	
24..27	Total iSCSI Offload Bytes Received (upper?)			
28..31	Total iSCSI Offload Bytes Received (lower?)			
32..35	Total iSCSI Offload Bytes Transmitted (upper?)			
36..39	Total iSCSI Offload Bytes Transmitted (lower?)			
40..43	Total iSCSI Offload PDUs Received			
44..47	Total iSCSI Offload PDUs Transmitted			
48..51	Checksum			

3114

3115 #### 8.4.80.1 Counters Cleared from Last Read

3116 The NC shall indicate in the Counters Cleared from Last Read field whether the corresponding fields have
3117 been cleared since they were last read over NC-SI.

3118 **Table 166 – Counters Cleared from Last Read field format**

Bit Position	Field Description	Value Description
0	Total iSCSI Offload Bytes Received	0b = Not Cleared 1b = Cleared
1	Total iSCSI Offload Bytes Transmitted	0b = Not Cleared 1b = Cleared
2	Total iSCSI Offload PDUs Received	0b = Not Cleared 1b = Cleared
3	Total iSCSI Offload PDUs Transmitted	0b = Not Cleared 1b = Cleared
15..4	Reserved	

3119

3120 ### 8.4.81 Get Partition Statistics response for InfiniBand (0xAF)

3121 In the absence of any errors, the channel shall process and respond to the Get Partition Statistics
3122 Command and send the response packet shown below when the Stats Type indicates InfiniBand.

3123 Currently no command-specific reason code is identified for this response.

3124 **Table 1676 – Get Partition Statistics (IB) response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Stats Type	Reserved	Counters Cleared	
24..27	Total Unicast Packets Received			
28..31	Total Multicast Packets Received			
32..35	Total Unicast Packets Transmitted			
36..39	Total Multicast Packets Transmitted			
40..43	Total Unicast Bytes Received			
44..47	Total Multicast Bytes Received			
48..51	Total Unicast Bytes Transmitted			
52..55	Total Multicast Bytes Transmitted			
56..59	Checksum			

3125

3126 #### 8.4.81.1 Counters Cleared from Last Read

3127 The NC shall indicate in the Counters Cleared from Last Read field whether the corresponding fields have
3128 been cleared since they were last read over NC-SI.

3129 **Table 168 – Counters Cleared from Last Read field format**

Bit Position	Field Description	Value Description
0	Total Unicast Packets Received	0b = Not Cleared 1b = Cleared
1	Total Multicast Packets Received	0b = Not Cleared 1b = Cleared
2	Total Unicast Packets Transmitted	0b = Not Cleared 1b = Cleared
3	Total Multicast Packets Transmitted	0b = Not Cleared 1b = Cleared
4	Total Unicast Bytes Received	0b = Not Cleared 1b = Cleared
5	Total Multicast Bytes Received	0b = Not Cleared 1b = Cleared
6	Total Unicast Bytes Transmitted	0b = Not Cleared 1b = Cleared
7	Total Multicast Bytes Transmitted	0b = Not Cleared 1b = Cleared
15..8	Reserved	

3130

3131 ### 8.4.82 Get Partition Statistics response for RDMA (0xAF)

3132 In the absence of any errors, the channel shall process and respond to the Get Partition Statistics
3133 Command and send the response packet shown below when the Stats Type indicates RDMA.

3134 Currently no command-specific reason code is identified for this response.

3135 **Table 169 – Get Partition Statistics (RDMA) response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Stats Type	Reserved	Counters Cleared	
24..27	Total RDMA Bytes Received (upper)			
28..31	Total RDMA Bytes Received (lower)			
32..35	Total RDMA Bytes Transmitted (upper)			
36..39	Total RDMA Bytes Transmitted (lower)			
40..43	Total RDMA Packets Received (upper)			
44..47	Total RDMA Packets Received (lower)			
48..51	Total RDMA Packets Transmitted (upper)			
52..55	Total RDMA Packets Transmitted (lower)			
56..59	Total Read Request Packets Transmitted (upper)			
60..63	Total Read Request Packets Transmitted (lower)			
64..67	Total Send Packets Transmitted (upper)			
68..71	Total Send Packets Transmitted (lower)			
72..75	Total Write Packets Transmitted (upper)			
76..79	Total Write Packets Transmitted (lower)			
80..83	Checksum			

3136

3137 **8.4.82.1 Counter Sizes**

3138 The NC shall indicate in the Counter Sizes field whether the implementation uses 32-bit counters or 64-bit
3139 counters in those counter fields above that are defined as 64-bit.

3140 **Table 170 – Counter Sizes field format**

Bit Position	Field Description	Value Description
0	Total RDMA Bytes Received	`0b` = 32-bit `1b` = 64-bit
1	Total RDMA Bytes Transmitted	`0b` = 32-bit `1b` = 64-bit
2	Total RDMA Packets Received	`0b` = 32-bit `1b` = 64-bit
3	Total RDMA Packets Transmitted	`0b` = 32-bit `1b` = 64-bit

Bit Position	Field Description	Value Description
4	Total Read Request Packets Transmitted	0b = 32-bit 1b = 64-bit
5	Total Send Packets Transmitted	0b = 32-bit 1b = 64-bit
6	Total Write Packets Transmitted	0b = 32-bit 1b = 64-bit
7	Reserved	

3141

3142

3143 #### 8.4.82.2 Counters Cleared from Last Read

3144 The NC shall indicate in the Counters Cleared from Last Read field whether the corresponding fields have
3145 been cleared since they were last read over NC-SI.

3146 **Table 171 – Counters Cleared from Last Read field format**

Bit Position	Field Description	Value Description
0	Total RDMA Bytes Received	0b = Not Cleared 1b = Cleared
1	Total RDMA Bytes Transmitted	0b = Not Cleared 1b = Cleared
2	Total RDMA Packets Received	0b = Not Cleared 1b = Cleared
3	Total RDMA Packets Transmitted	0b = Not Cleared 1b = Cleared
4	Total Read Request Packets Transmitted	0b = Not Cleared 1b = Cleared
5	Total Send Packets Transmitted	0b = Not Cleared 1b = Cleared
6	Total Write Packets Transmitted	0b = Not Cleared 1b = Cleared
15..7	Reserved	

3147

3148

3149 ### 8.4.83 Get Partition Statistics Response for Fibre Channel (0xAF)

3150 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get
3151 Partition Statistics command and send a response when the Stats Type indicates FC..

Table 172 illustrates the packet format of the Get FC Statistics Response.

Table 172 – Get Partition Statistics (FC) Response packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Reserved	Reserved	Counters Cleared from Last Read	
24..27	Total FC Frames Received			
28..31	Total FC Frames Transmitted			
32..35	Receive KB Count			
36..39	Transmit KB Count			
40..43	FC Sequences Received			
44..47	FC Sequences Transmitted			
48..51	Link Failures			
52..55	Loss of Signal			
56..59	Invalid CRCs			
60..63	Checksum (3..2)		Checksum (1..0)	

8.4.83.1 Counters Cleared from Last Read field

The FC Controller shall also indicate in the Counters Cleared from Last Read field whether the corresponding fields has been cleared since it was last read via NC-SI. The Counters Cleared from Last Read fields should have the format shown in Table 173.

Table 173 – Counters Cleared from Last Read fields format

Bit Position	Field Description	Value Description
0	Total FC Frames Received	0b = Not Cleared 1b = Cleared
1	Total FC Frames Transmitted	0b = Not Cleared 1b = Cleared
2	Receive KB Count	0b = Not Cleared 1b = Cleared
3	Transmit KB Count	0b = Not Cleared 1b = Cleared
4	FC Sequences Received	0b = Not Cleared 1b = Cleared
5	FC Sequences Transmitted	0b = Not Cleared 1b = Cleared
6	Link Failures	0b = Not Cleared 1b = Cleared

Bit Position	Field Description	Value Description
7	Loss of Signal	0b = Not Cleared 1b = Cleared
8	Invalid CRCs	0b = Not Cleared 1b = Cleared
15..9	Reserved	

3159 **8.4.83.2 FC Statistics Counter definitions**

3160 **Table 174 – FC Statistics**

Name	Meaning
Total FC Frames Received	Counts the number of FC frames received by the port
Total FC Frames Transmitted	Counts the number of FC frames transmitted by the port
Receive KB Count	Counts the number of kilobytes transmitted by the port
Transmit KB Count	Counts the number of kilobytes transmitted by the port
FC Sequences Received	Counts the number of FC sequences received by the port
FC Sequences Transmitted	Counts the number of FC sequences transmitted by the port
Link Failures	Counts the number of times the link has failed.
Loss of Signal	Counts the number of times the signal was lost.
Invalid CRCs	Counts the number of CRC errors detected.

3161

8.4.84 Get FC Link Status command (0x31)

The Get FC Link Status command allows the Management Controller to query the channel for potential link status and error conditions (see Table 175).

Implementation of this command is conditional and is required only for controllers supporting native Fibre Channel.

Table 175 – Get FC Link Status command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Reserved	Reserved	Reserved	Reserved
20..23	Checksum (3..2)		Checksum (1..0)	
24..27	Pad			

8.4.85 Get FC Link Status Response (0xB1)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get FC Link Status command and send a response (see Table 176).

Table 176 – Get FC Link Status Response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Reserved	Reserved	OS Driver Status	FC Link Status
24..27	Checksum			
28..31	Pad			

Table 177 describes the FC Link Status field bit definitions.

Table 177 – FC Link Status field bit definitions

Bit Position	Field Description	Value Description
0	Link Flag	`0b` = Link is down `1b` = Link is up

Bit Position	Field Description	Value Description
4..1	Link Speed	0x0 = No link speed established 0x1 = FC2 0x2 = FC4 0x3 = FC8 0x4 = FC16 0x5 = FC32 0x6 = FC64 0x7 = FC128
7..5	Reserved	None

Table 178 describes the OS Driver Status field bit definitions.

Table 178 – OS Driver Status field bit definitions

Bits	Description	Values
1..0	Host Driver Status Indication	0x0 = Fibre Channel Host driver state feature is not implemented. 0x1 = Fibre Channel Host driver state is not operational 0x2 = Fibre Channel Host driver state is operational 0x3 = Reserved
7..2	Reserved	None

Table 179 describes the reason code that is specific to the Get FC Link Status command.

Table 179 – Get FC Link Status Command-Specific Reason Code

Value	Description	Comment
0x800C	Link Command Failed-Hardware Access Error	Returned when PHY R/W access fails while executing the Get FC Link Status command

8.4.86 Get InfiniBand Link Status command (0x38)

The Get InfiniBand Link Status command allows the Management Controller to query the channel for the IB Statistics.

Implementation of this command is conditional and is required only for controllers supporting InfiniBand.

Table 180 illustrates the packet format of the InfiniBand Link Status command.

3184 **Table 180 – Get InfiniBand Link Status command**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Checksum (3..2)		Checksum (1..0)	
20..45	Pad			

3185 ## 8.4.87 Get InfiniBand Link Status Response (0xB8)

3186 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get
3187 InfiniBand Link Status command and send a response.

3188 The Get InfiniBand Link Status Response frame reports link width, logical and physical link state and the
3189 supported and configured link speed of the port.

3190 Table 181 illustrates the packet format of the Get InfiniBand Link Status Response.

3191 **Table 181 – Get InfiniBand Link Status Response packet**

	Bits				
Bytes	**31..24**	**23..16**	**15..08**	**07..00**	
00..15	NC-SI Header				
16..19	Response Code		Reason Code		
28..31	IB Link Active Width	IB Link Supported Width	Link Type	Phys State	Log State
32..35	Reserved	IB Link Active Speed	Reserved	IB Link Supported Speed	
44..47	Checksum (3..2)		Checksum (1..0)		

3192 **Table 182 – InfiniBand Link Status definitions**

Name	Direction	Description
IB Link Active Width	TX	When Link Type is InfiniBand and physical link is up, this field reflects the active link width. Otherwise this field returns 0b. Bit 0 – 1b = 1X Bit 1 - 1b = 2X Bit 2 - 1b = 4X Bit 3 - 1b = 8X Bits 7:4 Reserved

Name	Direction	Description
IB Link Supported Width	RX	When Link Type is InfiniBand, this field reflects the supported link widths. When Link Type is Ethernet, this field returns 0. Bit 0 - 1b = 1X Bit 1 - 1b = 2X Bit 2 - 1b = 4X Bit 3 - 1b = 8X Bits 7:4 Reserved
Link Type	TX	Reflects the configured link type. Bit 0 - 0b = Ethernet 1b = InfiniBand
Phys State	RX	The physical link state as specified in IB spec (PortInfoPortPhysicalState) 0x0 = Used when Link Type is Ethernet 0x1 = Sleep 0x2 = Polling 0x3 = Disabled 0x4 = PortConfigurationTraining 0x5 = LinkUp 0x6 = LinkErrorRecovery 0x7 = PhyTest
Logical Port State	TX	The logical port state of the physical port as specified in IB spec (PortInfo.PortState) 0x0: Used when Link Type is Ethernet 0x1: Down 0x2: Init 0x3: Arm 0x4: Active
IB Link Active Speed	TX	When Link Type is InfiniBand and the physical link is up, this field reflects the active link speed. Otherwise this field returns `0x00`. Bit 0 – `1b` = SDR Bit 1 - `1b` = DDR Bit 2 - `1b` = QDR Bit 3 - `1b` = FDR10 Bit 4 - `1b` = FDR Bit 5 - `1b` = EDR Bit 6 - `1b` = HDR Bit 7 - `1b` = NDR

Name	Direction	Description
IB Link Supported Speed	RX	When Link Type is InfiniBand, this field reflects the supported link speeds. When Link Type is Ethernet this field returns `0x00`. Bit 0 - `1b` = SDR Bit 1 - `1b` = DDR Bit 2 - `1b` = QDR Bit 3 - `1b` = FDR10 Bit 4 - `1b` = FDR Bit 5 - `1b` = EDR Bit 6 - `1b` = HDR Bit 7 - `1b` = NDR

8.4.88 Get IB Statistics command (0x39)

The Get IB Statistics command allows the Management Controller to query the channel for the IB Statistics.

Implementation of this command is conditional and is required only for controllers supporting InfiniBand.

Table 183 illustrates the packet format of the Get IB Statistics Command.

Table 183 – Get IB Statistics Command

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Checksum (3..2)		Checksum (1..0)	
20..45	Pad			

8.4.89 Get IB Statistics Response (0xB9)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get IB Statistics command and send a response.

The Get IB Statistics Response frame reports a set of IB statistics from the channel. A value of `0xFFFFFFFF shall be used` for any unsupported counter.

All counters are reset on Controller reset or power-cycle only.

Table 184 illustrates the packet format of the Get IB Statistics Response.

3207 **Table 184 – Get IB Statistics Response packet**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	PortXmitData			
24..27	PortRcvData			
28..31	PortXmitPkts			
32..35	PortRcvPkts			
36..39	PortXmitWait			
40..43	PortXmitDiscard			
44..47	SymbolErrorCounter			
48..51	LinkErrorRecoveryCounter			
52..55	LinkDownedCounter			
56..59	PortRcvErrors			
60..63	PortRcvRemotePhysicalErrors			
64..67	PortRcvSwitchRelayErrors			
68..71	LocalLinkIntegrityErrors			
72..75	ExcessiveBufferOverrun			
76..79	VL15Dropped			
80..83	Checksum (3..2)		Checksum (1..0)	

3208 **Table 185 – IB Statistics Counter definitions**

Name	Direction	Description
PortXmitData	TX	Total number of data octets, divided by 4 (lanes), transmitted on all VLs.
PortRcvData	RX	Total number of data octets, divided by 4 (lanes), received on all VLs.
PortXmitPkts	TX	Total number of packets transmitted on all VLs from this port. This may include packets with errors.
PortRcvPkts	RX	Total number of packets (this may include packets containing Errors.
PortXmitWait	TX	Number of ticks during which the port had data to transmit but no data was sent during the entire tick (either because of insufficient credits or because of lack of arbitration).
PortXmitDiscard	TX	Total number of outbound packets discarded by the port because the port is down or congested.
SymbolErrorCounter	RX	Total number of minor link errors detected on one or more physical lanes.
LinkErrorRecoveryCounter	RX	Total number of times the Port Training state machine has successfully completed the link error recovery process.

Name	Direction	Description
LinkDownedCounter	RX	Total number of times the Port Training state machine has failed the link error recovery process and downed the link.
PortRcvErrors	RX	Total number of packets containing an error that were received on the port.
PortRcvRemotePhysicalErrors	RX	Total number of packets marked with the EBP delimiter received on the port.
PortRcvSwitchRelayErrors	RX	Total number of packets received on the port that were discarded because they could not be forwarded by the switch relay.
LocalLinkIntegrityErrors	RX	Number of times that the count of local physical errors exceeded the threshold specified by LocalPhyErrors.
ExcessiveBufferOverrun	RX	Number of times that OverrunErrors consecutive flow control update periods occurred, each having at least one overrun error.
VL15Dropped	RX	Number of incoming VL15 packets dropped due to resource limitations (e.g., lack of buffers) of the port.

8.4.90 Get ASIC Temperature (0x48)

The Get ASIC Temperature command allows the Management controller to query for temperature values from the Controller's on-chip thermal sensor(s) or alternately from attached devices.

The Get ASIC Temperature command is defined as both a package level command and a channel command. This means the command can be either addressed to the package (that is, the command is sent with a Channel ID set to `0x1F`) or addressed to a specific channel in the package.

When sent as a package command, the internal temperature of the controller is returned. If the controller has multiple internal temperature sensors, the highest measured temperature with respect to its threshold shall be returned.

In cases where there are other devices connected to the controller that can also report silicon temperature via the controller (such as one or more external PHYs), then the channel version of the command is used and the response contains the temperature data and threshold from the external device on that channel. Multiple sensor implementations in the external device shall be handled as described above.

Table 186 illustrates the packet format of the Get ASIC Temperature Command.

Table 186 – Get ASIC Temperature Command packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Reserved			
20..23	Checksum (3..2)		Checksum (1..0)	
24..45	Pad			

8.4.91 Get ASIC Temperature Response (0xC8)

The package shall, in the absence of a checksum error or identifier mismatch, always accept the Get ASIC Temperature Command and send a response.

Table 187 illustrates the packet format of the Get ASIC Temperature Response.

Table 187 – Get ASIC Temperature Response packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Reserved	Reserved	Maximum temperature	Current temperature
24..27	Checksum (3..2)		Checksum (1..0)	
32..45	Pad			

8.4.91.1 Maximum Temperature Value

This value is the maximum T-Diode temperature limit in degrees Celsius at which the controller can operate at full load for its rated service lifetime. The value should be derated to take measurement tolerance into account. The value shall be reported as a hexadecimal integer number.

8.4.91.2 Current Temperature Value

This value is the current real-time temperature of the chip in degrees Celsius. The value shall be reported as a hexadecimal integer number.

8.4.92 Get Ambient Temperature (0x49)

The Get Ambient Temperature command allows the Management controller to query for temperature values from ambient temperature sensor(s) attached to the Controller.

The Get Ambient Temperature command is defined as a package command.

Controllers that do not support ambient temperature sensors should not implement this command.

Table 188 illustrates the packet format of the Get Ambient Temperature command.

Table 188 – Get Ambient Temperature command packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Reserved			
20..23	Checksum (3..2)		Checksum (1..0)	
24..45	Pad			

8.4.93 Get Ambient Temperature Response (0xC9)

The Package shall, in the absence of a checksum error or identifier mismatch, always accept the Get Ambient Temperature Command and send a response.

Table 189 illustrates the packet format of the Get Ambient Temperature Response.

Table 189 – Get Ambient Temperature Response packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Temperature3 value	Temperature2 Value	Temperature1 Value	Number of sensors
24..27	Checksum (3..2)		Checksum (1..0)	
32..45	Pad			

8.4.93.1 Temperature Value

This value (zero or more as specified by the Number of sensors field) is the real time ambient temperature reported in degrees Celsius. The value shall be reported as a hexadecimal integer number.

8.4.94 Get SFF Module Temperature (0x4A)

The Get SFF Module Temperature command allows the Management controller to query for the real time temperature value and thresholds of the (optical) transceiver attached to the channel. Implementations that do not support either fixed optics, or an SFF-like cage that supports pluggable transceivers that can provide temperature information, such as a Base-T Ethernet adapter, should not implement this command.

Table 190 illustrates the packet format of the Get SFF Module Temperature Command.

Table 190 – Get SFF Module Temperature Command Packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Reserved			
20..23	Checksum (3..2)		Checksum (1..0)	
24..45	Pad			

8.4.95 Get SFF Module Temperature Response (0xCA)

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get SFF Module Temperature command and send a response.

The Get SFF Module Temperature Response frame contains the current temperature of the attached module and the high side temperature thresholds.

Definitions and interpretation of the data fields in the response are defined in the relevant SFF or MSA specification (e.g., SFF-8472, SFF-8436, SFF-8636, etc.) for the transceiver. 16-bit values are encoded as one contiguous entity with the most significant bit in bit 15 (or 31) and least significant bit in bit 0 (or 16) in the response packet. The Controller is not expected to modify the data read from the transceiver.

In cases where the transceiver supports more than one channel, each channel shall provide a response when queried.

The reason code - *Information not available* - shall be used if the transceiver is not present, does not provide temperature data or if the command is issued before the transceiver has not yet achieved power up state.

Table 191 illustrates the packet format of the Get SFF Module Temperature Response.

Table 191 – Get SFF Module Temperature Response packet

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Temp High Alarm Threshold		Temp High Warning Threshold	
24..27	Temperature Value		Reserved	
28..31	Checksum (3..2)		Checksum (1..0)	

8.4.96 OEM command (0x50)

The OEM command may be used by the Management Controller to request that the channel provide vendor-specific information. The Vendor Enterprise Number is the unique MIB/SNMP Private Enterprise number assigned by IANA per organization. Vendors are free to define their own internal data structures in the vendor data fields.

Table 192 illustrates the packet format of the OEM command.

Table 192 – OEM command packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Manufacturer ID (IANA)			
20…	Vendor-Data NOTE: The optional checksum is unspecified for the OEM command. OEMs supporting checksum validation for NC-SI commands may include the checksum in the OEM specific payload for the command and response.			

8.4.97 OEM response (0xD0)

The channel shall return the “Unknown Command Type” reason code for any unrecognized enterprise number, using the packet format shown in Table 193. If the command is valid, the response, if any, is allowed to be vendor-specific. The 0x8000 range is recommended for vendor-specific code.

Currently no command-specific reason code is identified for this response.

3290 **Table 193 – OEM response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (IANA)			
24…	Return Data (Optional) NOTE The optional checksum is unspecified for the OEM command. OEMs supporting checksum validation for NC-SI commands may include the checksum in the OEM specific payload for the command and response.			

3291 ## 8.4.98 PLDM Request (0x51)

3292 The PLDM Request Packet may be used by the Management Controller to send PLDM commands over
3293 NC-SI/RBT. This command may be targeted at the entire package or a specific channel.

3294 Table 194 illustrates the packet format of the PLDM Request Packet over NC-SI/RBT.

3295 **Table 194 – PLDM Request packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	PLDM Message Common Fields			
20..	PLDM Message Payload (zero or more bytes) + Payload Pad (see 8.2.2.2)			
..	Checksum			
..	Ethernet Packet Pad (optional – See 8.2.2.4)			

3296 Refer to the PLDM Base specification (DSP0240) for details on the PLDM Request Messages.

3297 ## 8.4.99 PLDM Response (0xD1)

3298 The PLDM Response Packet may be used by the Network Controller to send PLDM responses over NC-
3299 SI/RBT. The package shall, in the absence of a checksum error or identifier mismatch, always accept the
3300 PLDM Request Command and send a response.

3301 **Table 195 – PLDM Response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..23	PLDM Message Common Fields			PLDM Completion Code
24..	PLDM Message Payload (zero or more bytes) + Payload Pad (see 8.2.2.2)			

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
..	Checksum			
..	Ethernet Packet Pad (optional – See 8.2.2.4)			

Refer to the PLDM Base specification (DSP0240) for details on the PLDM Response Messages.

Note that the NC-SI PLDM Response (`0xD1`) response/reason codes are only used to report the support, success, or failure of the PLDM Request command (`0x51`) at the NC-SI over RBT messaging layer. The PLDM Completion Code is used for determining the success or failure of the encapsulated PLDM Commands at the PLDM messaging layer.

8.4.100 Query Pending NC PLDM Request (0x56)

The Query Pending NC PLDM Request may be used by the Management Controller to read the status of pending PLDM commands which the NC needs to send to the MC. Only one PLDM request can be handled at any time. When multiple requests are pending in the NC, each will be handled independently and the order at which requests are provided to the MC is decided by the NC.

NC which supports PLDM over RBT, where the NC has to send PLDM commands to the MC, shall support this command. It is expected that the MC will use PLDM Request command `0x51` to query the supported PLDM commands, before using Query Pending NC PLDM Request command.

Table 196 – Query Pending NC PLDM Request packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Checksum			
20..45	Pad			

8.4.101 Query Pending NC PLDM Request Response (0xD6)

Currently no command-specific reason code is identified for this response (see Table 197).

3319 **Table 197 – Query Pending NC PLDM Request Response Packet Format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..	PLDM Message Common Fields			PLDM Message Payload
	PLDM Message Payload + Payload Pad (zero or more bytes)			
	Checksum			
	Pad			

3320

3321 **Table 198 – Query Pending NC PLDM Request Response parameters**

Name	Meaning
PLDM Message Common fields	Optional, included only when there is a pending request
PLDM Message Payload	Optional, included only when there is a pending request

3322 ### 8.4.102 Send NC PLDM Reply (0x57)

3323 The Reply Pending PLDM command may be used by the Management Controller to provide the PLDM
3324 command response to previously read PLDM command from the NC that requires a response (Rq = 1, D
3325 = 0 in PLDM Message Common Fields). The response to this command further provides indication to the
3326 MC regarding additional pending PLDM NC commands.

3327 **Table 199 – Send NC PLDM Reply packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	PLDM Message Common Fields			PLDM Completion Code
20..	PLDM Message Payload (zero or more bytes) + Payload Pad			
	Checksum			
	Pad			

3328

3329 ### 8.4.103 Send NC PLDM Reply Response (0xD7)

3330 Currently no command-specific reason code is identified for this response (see Table 200).

3331 **Table 200 – Reply NC PLDM Response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Reserved			Flags
24..27	Checksum			
28..45	Pad			

3332

3333 **Table 201 – Reply NC PLDM Response parameters**

Name	Meaning
Flags bit 0 – Pending request	`0b` – No additional pending PLDM command from NC to MC `1b` – The NC has additional pending PLDM command to the MC
Flags bits 7:1 - Reserved	Reserved, always return 0.

3334 ### 8.4.104 Pending PLDM request AEN and associated enablement commands

3335 An optional medium specific AEN is defined. This AEN allows the NC to notify the MC regarding a
3336 pending PLDM command that the NC has to send to the MC.

3337 As a transport specific AEN, this AEN is enabled using the transport specific AEN enable command and
3338 is controlled by bit 1 in Transport Specific AENs enable field.

3339 The AEN Type for this AEN shall be `0x71` and is described below.

3340 ### 8.4.105 Transport Specific AEN Enable command (0x55)

3341 Network Controller implementations shall support this command on the condition that the Network
3342 Controller generates one or more transport specific AENs defined in this specification or other NC-SI
3343 bindings such as DSP0261. The AEN Enable command enables and disables the different transport
3344 specific AENs supported by the Network Controller. The Network Controller shall copy the AEN MC ID
3345 field from the AEN Enable command into the MC ID field in every subsequent AEN sent to the
3346 Management Controller as defined in AEN Enable command

3347 Table 202 illustrates the packet format of the Enable Transport Specific AENs command.

3348 **Table 202 – Transport Specific AENs Enable command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Reserved		Transport Specific AENs enable	
20..23	Checksum			
24..45	Pad			

3349 **Table 203 – Transport Specific AENs enable field format**

Bit Position	Field Name	Value Description
0	Medium Change AEN Control (`0x70`)	`0b` = Disable Medium Change AEN `1b` = Enable Medium Change AEN Relevant only for NC-SI/MCTP
1	Pending PLDM Request AEN (`0x71`)	`0b` = Disable Pending PLDM Request AEN `1b` = Enable Pending PLDM Request AEN Relevant only for PLDM over NC-SI control over RBT
2..15	Reserved For future AEN	Reserved

3350

3351 ### 8.4.106 Transport Specific AENs Enable Response (0xD5)

3352 In the absence of any error, the package shall process and respond to the Transport Specific AENs
3353 Enable command by sending the response packet and payload shown in Table 204.

3354 **Table 204 – Transport Specific AENs Enable Response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Checksum			
…	Pad			

3355

3356 ### 8.4.107 Pending PLDM Request AEN

3357 The Pending PLDM Request AEN is used to alert the MC that there is a pending PLDM request for the
3358 MC in the NC. This AEN allows for the MC to poll for pending PLDM request on the NC at a lower rate.

3359 This AEN should be sent if there is a new pending PLDM command that is available in the NC designated
3360 to the MC, which was not reported to the MC through **Send NC PLDM Reply Response (**`0xD7`**)**. A
3361 Pending PLDM Request AEN should not be sent from the time the NC recognizes an incoming **Query**

3362 **Pending NC PLDM Request** (0x56) until the NC sends **Send NC PLDM Reply Response (**0xD7**)** for the
3363 PLDM request.

3364 **Table 205 – Pending PLDM Request AEN format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Reserved			AEN Type = 0x71
20..23	Checksum			
24..45	Pad			

3365 ### 8.4.108 Get MC MAC Address command (0x??)

3366 A network controller may provision MAC addresses for Out-Of-Band (OOB) management traffic. These
3367 MAC addresses are not visible to the host(s). Get MC MAC Address is used to discover MAC addresses
3368 provisioned on the network controller for the MC. Get MC MAC Address is a channel-specific command.
3369 For multiport devices, it is expected that the MC queries provisioned MC MAC Addresses on each
3370 channel individually.

3371 Table 206 illustrates the packet format of the Get MC Address Command over NC-SI/RBT.

3372 **Table 206 – Get MC MAC Address command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Checksum			
20..45	Pad			

3373 ### 8.4.109 Get MC MAC Address response (0x??)

3374 In the response of Get MC MAC Address command, the network controller provides the information about
3375 the provisioned MAC addresses for the MC on that channel. The NC shall, in the absence of an error,
3376 always accept the Get MC MAC Address command and send the response packet shown in Table 207.
3377 Currently no command-specific reason code is identified for this response.

3378 **Table 207 – Get MC MAC Address response packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Address Count	Reserved		
Variable	Addr 1 Byte 5	Addr 1 Byte 4	Addr 1 Byte 3	Addr 1 Byte 2
	Addr 1 Byte 1	Addr 1 Byte 0	Addr 2 Byte 5	Addr 2 Byte 4
	…			
	…		Pad (if needed)	

3379 #### 8.4.109.1 Address Count

3380 This field shall be set to the number of MC MAC addresses provisioned on the channel.

3381 #### 8.4.109.2 Reserved

3382 This field shall be set to 0 by the network controller and shall be ignored by the management controller.

3383 #### 8.4.109.3 Addr i Byte j

3384 This field shall be set to the value of jth byte ($1 \le j \le 6$) of ith provisioned MC MAC address.

3385 #### 8.4.109.4 Pad

3386 If the number of MC MAC addresses is an odd number, then 2 bytes of the Pad field shall be present at
3387 the end of the payload to align the payload on a 32-bit boundary. If present, each byte of the Pad field
3388 shall be set to `0x00`.

3389 If the number of MC MAC addresses is an even number, then 0 bytes of Pad shall be present.

3390 ### 8.4.110 Get Package UUID command (`0x52`)

3391 The Get Package UUID command may be used by the Management Controller to query Universally
3392 Unique Identifier (UUID), also referred to as a globally unique ID (GUID), of the Network Controller over
3393 NC-SI/RBT. This command is targeted at the entire package. This command can be used by the MC to
3394 correlate endpoints used on different NC-SI transports (e.g. RBT, MCTP).

3395 Table 208 illustrates the packet format of the Get Package UUID Command over NC-SI/RBT.

3396 **Table 208 – Get Package UUID command packet format**

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Checksum			
20..45	Pad			

8.4.111 Get Package UUID response (0xD2)

The package shall, in the absence of an error, always accept the Get Package UUID command and send the response packet shown in Table 209. Currently no command-specific reason code is identified for this response.

Table 209 – Get Package UUID response packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Control Packet Header			
16..19	Response Code		Reason Code	
20..35	UUID bytes 1:16, respectively			
36..39	Checksum			
40..45	Pad			

The individual fields within the UUID are stored most-significant byte (MSB) first per the convention described in RFC4122. RFC4122 specifies four different versions of UUID formats and generation algorithms suitable for use for a UUID. These are version 1 (0001b) "time based", and three "name-based" versions: version 3 (0011b) "MD5 hash", version 4 (0100b) "Pseudo-random", and version 5 "SHA1 hash". The version 1 format is recommended. However, versions 3, 4, or 5 formats are also allowed. See Table 210 for the UUID format version 1.

Table 210 – UUID Format

Field	UUID Byte	MSB
time low	1	MSB
	2	
	3	
	4	
time mid	5	MSB
	6	
time high and version	7	MSB
	8	
clock seq and reserved	9	MSB
	10	
node	11	MSB
	12	
	13	
	14	
	15	
	16	

8.5 AEN packet formats

This clause defines the formats for the different types of AEN packets. For a list of the AEN types, see Table 17.

8.5.1 Link Status Change AEN

The Link Status Change AEN indicates to the Management Controller any changes in the channel's external interface link status.

This AEN should be sent if any change occurred in the link status (that is, the actual link mode was changed). The Link Status and OEM Link Status fields reproduce the bit definitions defined in the Get Link Status Response Packet (see Table 49).

Table 211 illustrates the packet format of the Link Status Change AEN.

Table 211 – Link Status Change AEN packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	AEN Header			
16..19	Reserved			AEN Type = 0x00
20..23	Link Status			
24.27	OEM Link Status			
28..31	Checksum			

8.5.2 Configuration Required AEN

The Configuration Required AEN indicates to the Management Controller that the channel is transitioning into the Initial State. (This AEN is not sent if the channel enters the Initial State because of a Reset Channel command.)

NOTE This AEN may not be generated in some situations in which the channel goes into the Initial State. For example, some types of hardware resets may not accommodate generating the AEN.

Table 212 illustrates the packet format of the Configuration Required AEN.

Table 212 – Configuration Required AEN packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	AEN Header			
16..19	Reserved			AEN Type = `0x01`
20..23	Checksum			

8.5.3 Host Network Controller Driver Status Change AEN

This AEN indicates a change of the Host Network Controller Driver Status. Table 213 illustrates the packet format of the AEN.

Table 213 – Host Network Controller Driver Status Change AEN packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	AEN Header			
16..19	Reserved			AEN Type = 0x02
20..23	Host Network Controller Driver Status			
24..27	Checksum			

The Host Network Controller Driver Status field has the format shown in Table 214.

Table 214 – Host Network Controller Driver Status format

Bit Position	Name	Description
0	Host Network Controller Driver Status	0b = The Network Controller driver for the host external network interface associated with this channel is not operational (not running). 1b = The Network Controller driver for the host external network interface associated with this channel is being reported as operational (running).
1..31	Reserved	Reserved

8.5.4 Delayed Response Ready AEN

This AEN indicates the response to a delayed command is ready. Table 215 illustrates the packet format of the AEN.

Note: This AEN does not deliver the delayed command response, it must be retrieved separately.

Table 215 – Delayed Response Ready AEN packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	AEN Header			
16..19	Reserved			AEN Type = 0x03
20..23	Original Command Type	Original Command IID	Padding	
24..27	Checksum			

The Original Command Type includes the Control Packet Type field of the completed command and the Original Command IID includes the IID field of the original command.

8.5.5 Transceiver Event AEN

This indicates to the Management Controller that a change in presence status or a thermal threshold in the SFF-compliant Transceiver attached to the channel has occurred. Since some SFF cages have multiple TX and RX lanes, it is possible that multiple NC-SI channels are handled by a single transceiver module or copper cable assembly. Only one instance of the Transceiver Event AEN sent to one of the channels involved is required to enable reporting for all channels. The NC may|should|shall send the Transceiver Event AEN on all affected channels if one or more alerts are triggered.

Table 216 illustrates the packet format of the AEN.

Table 216 – Transceiver Event AEN packet format

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	AEN Header			
16..19	Reserved			AEN Type = `0x04`
20..23	Transceiver Event List			
24..27	Checksum			

The Transceiver Event List field has the format shown in Table 217.

Table 217 – Transceiver Event List format

Bit Position	Name	Description
0	Transceiver Presence Change	0b = No change in presence detected 1b = The Transceiver was either removed or inserted. The insertion event reporting shall occur only after the Transceiver has completed its initialization stage
1	Low Temp Warning	0b = no alert 1b = The Transceiver's low temperature warning threshold has been exceeded
2	High Temp Warning	0b = no alert 1b = The Transceiver's high temperature warning threshold has been exceeded
3	Low Temp Alarm	0b = no alert 1b = The Transceiver's low temperature alarm threshold has been exceeded
4	High Temp Alarm	0b = no alert 1b = The Transceiver's high temperature alarm threshold has been exceeded

Bit Position	Name	Description
5	Low Voltage Warning	0b = no alert 1b = The Transceiver's low voltage warning threshold has been exceeded
6	High Voltage Warning	0b = no alert 1b = The Transceiver's high voltage warning threshold has been exceeded
7	Low Voltage Alarm	0b = no alert 1b = The Transceiver's low voltage alarm threshold has been exceeded
8	High Voltage Alarm	0b = no alert 1b = The Transceiver's high voltage alarm threshold has been exceeded
	4 x RX Power Levels	TBD
	4 x TX Power Levels	TBD
	4 x TX Bias Levels	TBD
tbd	Reserved	Reserved

9 Packet-based and op-code timing

Table 218 presents the timing specifications for a variety of packet-to-electrical-buffer interactions, inter-packet timings, and op-code processing requirements. The following timing parameters shall apply to NC-SI over RBT binding defined in this specification.

Table 218 – NC-SI packet-based and op-code timing parameters

Name	Symbol	Value	Description
Package Deselect to Hi-Z Interval	T1	200 µs, max	Maximum time interval from when a Network Controller completes transmitting the response to a Deselect Package command to when the Network Controller outputs are in the high-impedance state Measured from the rising edge of the first clock that follows the last bit of the packet to when the output is in the high-impedance state as defined in clause 10
Package Output to Data	T2	2 clocks, min	Minimum time interval after powering up the output drivers before a Network Controller starts transmitting a packet through the NC-SI interface Measured from the rising edge of the first clock of the packet
Network Controller Power Up Ready Interval	T4	2 s, max	Time interval from when the NC-SI on a Network Controller is powered up to when the Network Controller is able to respond to commands over the NC-SI Measured from when V_{ref} becomes available
Normal Execution Interval	T5	50 ms, max	Maximum time interval from when a controller receives a command to when it delivers a response to that command, unless otherwise specified Measured from the rising edge of the first clock following the last bit of the command packet to the rising edge of the clock for the first bit of the response packet
Asynchronous Reset Interval	T6	2 s, max	Interval during which a controller may not recognize or respond to commands or handle Passthru traffic due to an Asynchronous Reset event. See clause 6.2.8 For a Management Controller, this means that a Network Controller could become unresponsive for up to T6 seconds if an Asynchronous Reset event occurs. This is not an error condition. The Management Controller retry behavior should be designed to accommodate this possibility.
Synchronous Reset Interval	T7	2 s, max	Interval during which a controller may not recognize or respond to commands or handle Passthru traffic due to a Synchronous Reset event. See clause 6.2.8 Measured from the rising edge of the first clock following the last bit of the Reset Channel response packet
Token Timeout	T8	32,000 REF_CLK min	Number of REF_CLKs before timing out while waiting for a TOKEN to be received

Name	Symbol	Value	Description
Op-Code Processing	T9	32 REF_CLK max	Number of REF_CLKs after receiving an op-code on ARB_IN to decode the op-code and generate the next op-code on ARB_OUT Measured from the falling edge of the last bit of the op-code received on ARB_IN to the rising edge of the next op-code on ARB_OUT
Op-Code Bypass Delay	T10	32 REF_CLK max	Number of REF_CLK delays between a bit received on ARB_IN and the corresponding bit passed on to ARB_OUT while in Bypass Mode Measured from the falling edge of the last bit of the op-code received on ARB_IN to the rising edge of the next op-code on ARB_OUT
TOKEN to RXD	T11	T2 min, 32 REF_CLK max	Number of REF_CLKs after receiving TOKEN to when packet data is driven onto the RXD lines Measured from the falling edge of the last bit of the op-code received on ARB_IN to the rising edge of the next op-code on ARB_OUT
Max XOFF Renewal Interval	T12	50,331,648 REF_CLK max	Maximum time period (3 XOFF Frame timer cycles) during which a channel within a package is allowed to request and renew a single XOFF condition after requesting the initial XOFF
IPG to TOKEN Op-code Overlap	T13	6 REF_CLK max	Maximum number of REF_CLKs that the beginning of TOKEN transmission can precede the end of the Inter Packet Gap. For more information, see 7.2.8.
Delayed Execution Interval	T14	4 s, max	Maximum time interval from when a controller receives a command to when it delivers a response to that command, including all responses with “Delayed Response” code Measured from the rising edge of the first clock following the last bit of the command packet to the rising edge of the clock for “Delayed Response Ready” AEN if enabled or to the moment the NC is internally ready with a response for a polling command.
NOTE If hardware arbitration is in effect, the hardware arbitration output buffer enable/disable timing specifications take precedence.			

10 RBT Electrical specification

This clause provides background information about the NC-SI RBT specification, describes the RBT topology, and defines the electrical, timing, signal behavior, and power-up characteristics for the RBT physical interface.

10.1 Topologies

The electrical specification defines the RBT electrical characteristics for one management processor and one to four Network Controller packages in a bussed “multi-drop” arrangement. The actual number of devices that can be supported may differ based on the trace characteristics and routing used to interconnect devices in an implementation.

Figure 16 shows an example topology.

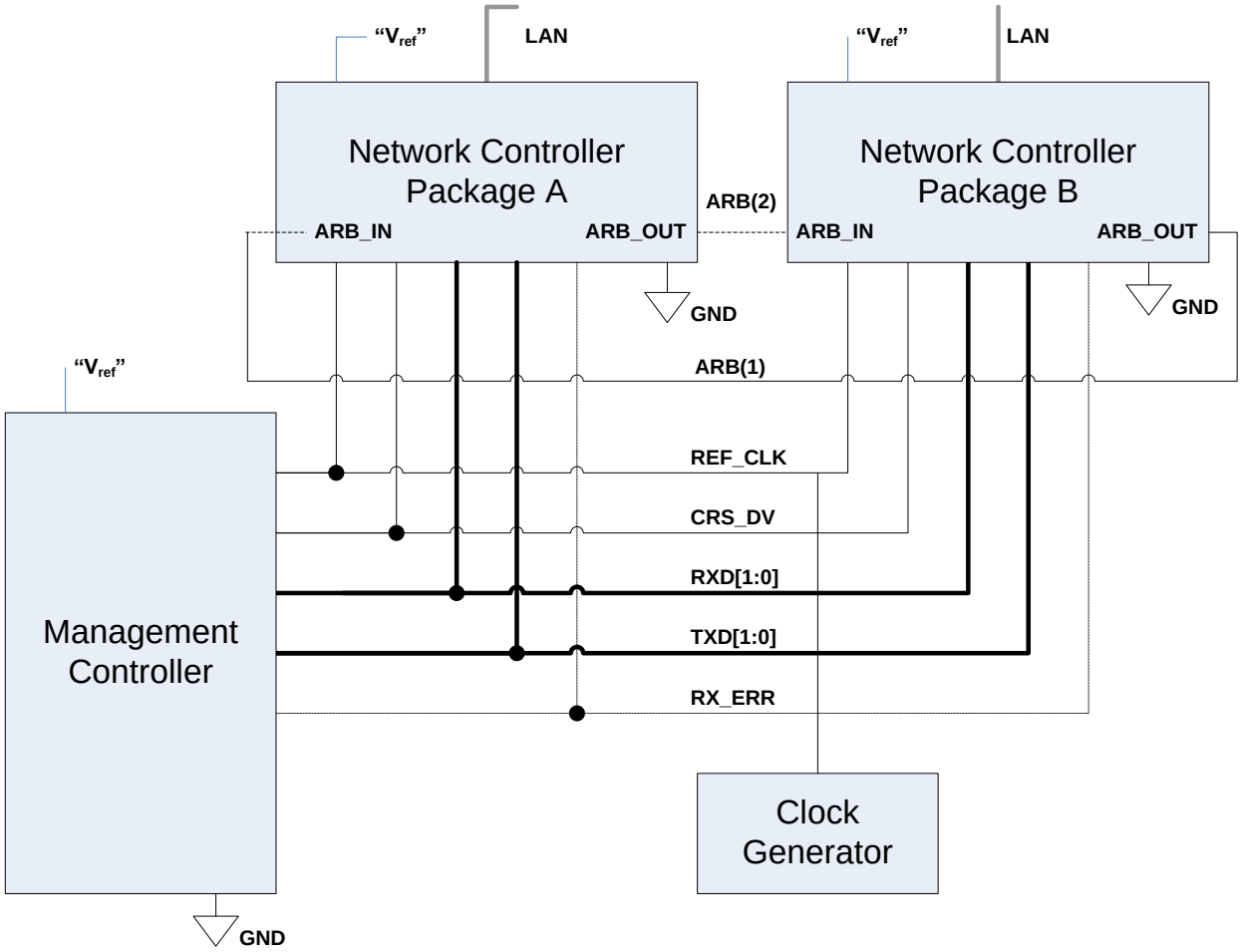


Figure 16 – Example NC-SI RBT signal interconnect topology

10.2 Electrical and signal characteristics and requirements

This clause defines the electrical, timing, signal behavior, and power-up characteristics for the NC-SI RBT physical interface.

10.2.1 Companion specifications

Implementations of the physical interface and signaling for RBT shall meet the specifications in RMII and IEEE 802.3, except where those requirements differ or are extended with specifications provided in this document, in which case the specifications in this document shall take precedence.

10.2.2 Full-duplex operation

The RBT is specified only for full-duplex operation. Half-duplex operation is not covered by this specification.

10.2.3 Signals

Table 219 lists the signals that make up the RBT physical interface.

Unless otherwise specified, the high level of a RBT signal corresponds to its asserted state, and the low level represents the de-asserted state. For data bits, the high level represents a binary '1' and the low level a binary '0'.

Table 219 – Physical RBT signals

Signal Name	Direction (with respect to the Network Controller)	Direction (with respect to the Management Controller MAC)	Use	Mandatory or Optional
REF_CLK [a]	Input	Input	Clock reference for receive, transmit, and control interface	M
CRS_DV [b]	Output	Input	Carrier Sense/Receive Data Valid	M
RXD[1:0]	Output	Input	Receive data	M
TX_EN	Input	Output	Transmit enable	M
TXD[1:0]	Input	Output	Transmit data	M
RX_ER	Output	Input	Receive error	O
ARB_IN	Input [c]	N/A	Network Controller hardware arbitration Input	O [c]
ARB_OUT	Output [c]	N/A	Network Controller hardware arbitration Output	O [c]

[a] A device may provide an additional option to allow it to be configured as the source of REF_CLK, in which case the device is not required to provide a separate REF_CLK input line, but it can use REF_CLK input pin as an output. The selected configuration shall be in effect at NC power up and remain in effect while the NC is powered up.

[b] In the *RMII Specification*, the MII Carrier Sense signal, CRS, was combined with RX_DV to form the CRS_DV signal. When RBT is using its specified full-duplex operation, the CRS aspect of the signal is not required; therefore, the signal shall provide only the functionality of RX_DV as defined in IEEE 802.3. (This is equivalent to the CRS_DV signal states in *RMII Specification* when a carrier is constantly present.) The Carrier Sense aspect of the CRS_DV signal is not typically applicable to RBT because it does not typically detect an actual carrier (unlike an actual PHY). However, the Network Controller should emulate a carrier-present status on CRS_DV per IEEE 802.3 in order to support Management Controller MACs that may require a carrier-present status for operation.

[c] If hardware arbitration is implemented, the Network Controller package shall provide both ARB_IN and ARB_OUT connections. In some implementations, ARB_IN may be required to be tied to a logic high or low level if it is not used.

10.2.4 High-impedance control

Shared RBT operation requires Network Controller devices to be able to set their outputs (RXD[1:0], CRS_DV, and, if implemented, RX_ER) into a high-impedance state either upon receipt of a command being received, or, if hardware-based arbitration is enabled as a result of hardware-based arbitration. A pull-down resistor should be provided on high impedance signals to prevent them from floating and keep their C_{load} value when not driven.

Network Controllers shall leave their RBT outputs in the high-impedance state on interface power up and shall not drive them until the package is selected. For additional information about Network Controller packages, see 8.4.5.

For RBT output signals in this specification, unless otherwise specified, the high-impedance state is defined as the state in which the signal leakage meets the I_z specification provided in 10.2.5.

10.2.5 DC characteristics

This clause defines the DC characteristics of the RBT physical interface.

10.2.5.1 Signal levels

CMOS 3.3 V signal levels are used for this specification.

The following characteristics apply to DC signals:

- Unless otherwise specified, DC signal levels and V_{ref} are measured relative to Ground (GND) at the respective device providing the interface, as shown in Figure 17.
- Input specifications refer to the signals that a device shall accept for its input signals, as measured at the device.
- Output specifications refer to signal specifications that a device shall emit for its output signals, as measured at the device.

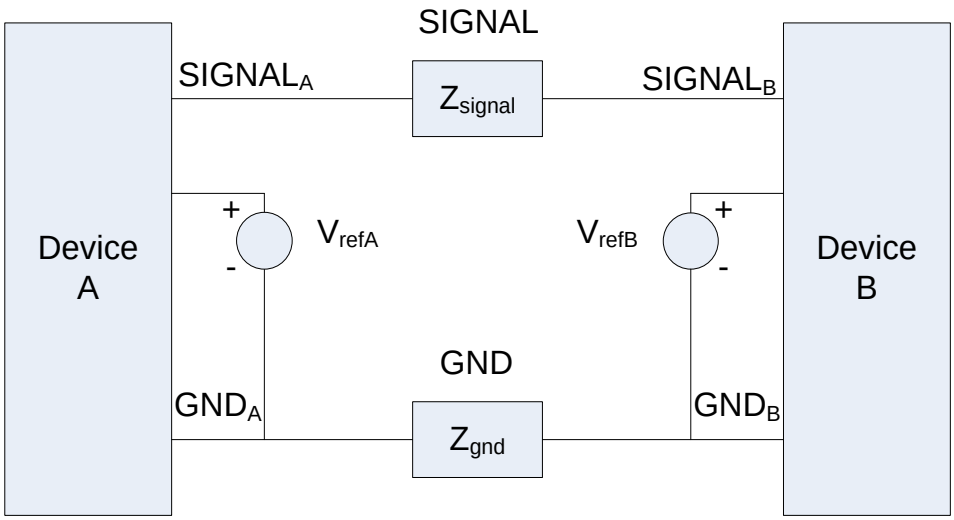


Figure 17 – DC measurements

3510 Table 220 provides DC specifications.

3511 **Table 220 – DC specifications**

Parameter	Symbol	Conditions	Minimum	Typical	Maximum	Units
IO reference voltage	V_{ref} [a]		3.0	3.3	3.6	V
Signal voltage range	V_{abs}		-0.300		3.765	V
Input low voltage	V_{il}				0.8	V
Input high voltage	V_{ih}		2.0			V
Input high current	I_{ih}	$V_{in} = V_{ref} = V_{ref,max}$	0		200	µA
Input low current	I_{il}	$V_{in} = 0$ V	-20		0	µA
Output low voltage	V_{ol}	$I_{ol} = 4$ mA, V_{ref} = min	0		400	mV
Output high voltage	V_{oh}	$I_{oh} = -4$ mA, V_{ref} = min	2.4		V_{ref}	V
Clock midpoint reference level	V_{ckm}				1.4	V
Leakage current for output signals in high-impedance state	I_z	$0 \le V_{in} \le V_{ref}$ at $V_{ref} = V_{ref,max}$	-20		20	µA

[a] V_{ref} = Bus high reference level (typically the NC-SI logic supply voltage). This parameter replaces the term *supply voltage* because actual devices may have internal mechanisms that determine the operating reference for RBT that are different from the devices' overall power supply inputs.

V_{ref} is a reference point that is used for measuring parameters (such as overshoot and undershoot) and for determining limits on signal levels that are generated by a device. To facilitate system implementations, a device shall provide a mechanism (for example, a power supply pin, internal programmable reference, or reference level pin) to allow V_{ref} to be set to within 20 mV of any point in the specified V_{ref} range. This approach enables a system integrator to establish an interoperable V_{ref} level for devices on RBT.

3512 ### 10.2.6 AC characteristics

3513 This clause defines the AC characteristics of the RBT physical interface.

3514 #### 10.2.6.1 Rise and fall time measurement

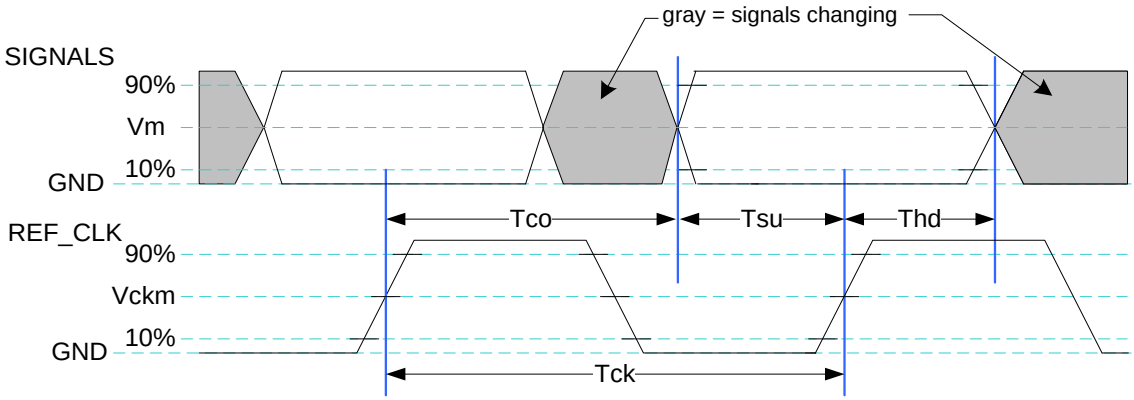
3515 Rise and fall time are measured between points that cross 10% and 90% of V_{ref} (see Table 220). The
3516 middle points (50% of V_{ref}) are marked as V_{ckm} and V_m for clock and data, respectively.

3517 #### 10.2.6.2 REF_CLK measuring points

3518 In Figure 18, REF_CLK duty cycle measurements are made from V_{ckm} to V_{ckm}. Clock skew T_{skew} is
3519 measured from V_{ckm} to V_{ckm} of two RBT devices and represents maximum clock skew between any two
3520 devices in the system.

3521 #### 10.2.6.3 Data, control, and status signal measuring points

3522 In Figure 18, all timing measurements are made between V_{ckm} and V_m. T_{co} is measured with a capacitive
3523 load between 10 pF and 50 pF. Propagation delay T_{prop} is measured from V_m on the transmitter to V_m on
3524 the receiver.



3525

3526 **Figure 18 – AC measurements**

3527 Table 221 provides AC specifications.

3528 **Table 221 – AC specifications**

Parameter	Symbol	Minimum	Typical	Maximum	Units
REF_CLK Frequency			50	50+100 ppm	MHz
REF_CLK Duty Cycle		35		65	%
Clock-to-out [a] (10 pF ≤ c_{load} ≤ 50 pF)	T_{co}	2.5		12.5	ns
Skew between clocks	T_{skew}			1.5	ns
TXD[1:0], TX_EN, RXD[1:0], CRS_DV, RX_ER, and ARB_IN data setup to REF_CLK rising edge	T_{su}	3			ns
TXD[1:0], TX_EN, RXD[1:0], CRS_DV, RX_ER, and ARB_OUT data hold from REF_CLK rising edge	T_{hd}	1			ns
Signal Rise/Fall Time	T_r/T_f	0.5		6	ns
REF_CLK Rise/Fall Time	T_{ckr}/T_{ckf}	0.5		3.5	ns
Interface Power-Up High-Impedance Interval	T_{pwrz}	2			µs
Power Up Transient Interval (recommendation)	T_{pwrt}			100	ns
Power Up Transient Level (recommendation)	V_{pwrt}	-200		200	mV
REF_CLK Startup Interval	$T_{clkstrt}$			100	ms

[a] This timing relates to the output pins, while T_{su} and T_{hd} relate to timing at the input pins.

10.2.6.4 Timing calculation (informative)

This clause presents the relationships between the timing parameters and how they are used to calculate setup and hold time margins.

10.2.6.4.1 Setup calculation

$$T_{su} \leq T_{clk} - (T_{skew} + T_{co} + T_{prop})$$

10.2.6.4.2 Hold calculation

$$T_{hd} \leq T_{co} - T_{skew} + T_{prop}$$

10.2.6.5 Overshoot specification

Devices shall accept signal overshoot within the ranges specified in Figure 19, measured at the device, without malfunctioning.

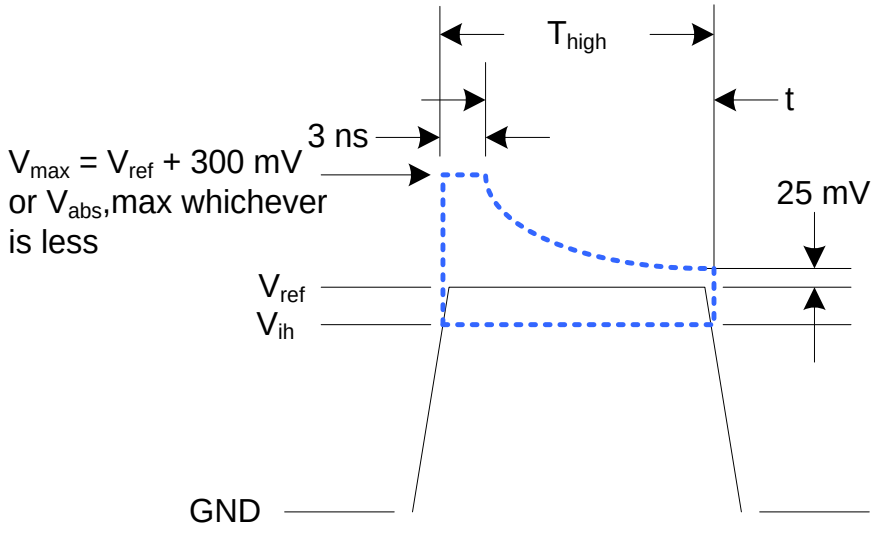


Figure 19 – Overshoot measurement

The signal may overshoot up to the specified V_{max} for the first 3 ns following the transition above V_{ih}. Following that interval is an exponential decay envelope equal to the following:

$$V_{ref} + V_{os} * e^{[- K * ([t - 3\ ns] / T_d)]}$$

Where, for t = 3 to 10 ns:

t = 0 corresponds to the leading crossing of V_{ih}, going high.

V_{ref} is the bus high reference voltage (see 10.2.5).

V_{abs},max is the maximum allowed signal voltage level (see 10.2.5).

$V_{os} = V_{max} - V_{ref}$

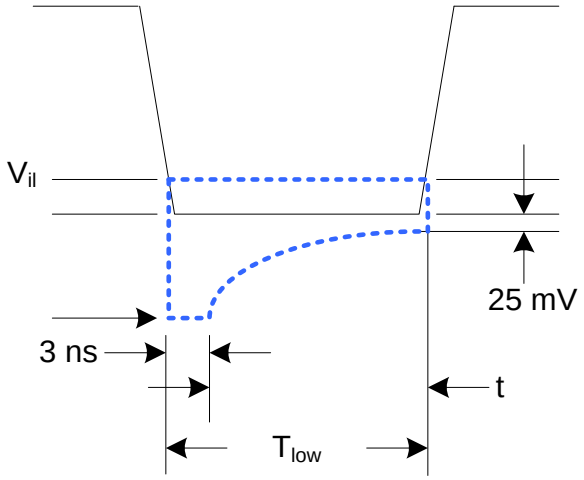
3549 $K = l_n(25\ mV/V_{os})$

3550 $T_d = 7\ ns$

3551 For t > 10 ns, the V_{ref} + 25 mV limit holds flat until the conclusion of T_{high}.

3552 ### 10.2.6.6 Undershoot specification

3553 Devices are required to accept signal undershoot within the ranges specified in Figure 20, measured at
3554 the device, without malfunctioning.



3555

3556 **Figure 20 – Undershoot measurement**

3557 The signal is allowed to undershoot up to the specified V_{abs},min for the first 3 ns following the transition
3558 above V_{il}. Following that interval is an exponential envelope equal to the following:

3559 $* ([t - 3\ ns]/T_d)]$

3560 Where, for t = 3 to 10 ns:

3561 t = 0 corresponds to the leading crossing of V_{il}, going low.

3562 V_{abs},min is the minimum allowed signal voltage level (see 10.2.5).

3563 $K = l_n(25\ mV/V_{os})$

3564 $T_d = 7\ ns$

3565 For t > 7 ns, the GND – 25 mV limit holds flat until the conclusion of T_{low}.

3566 ### 10.2.7 Interface power-up

3567 To prevent signals from back-powering unpowered devices, it is necessary to specify a time interval
3568 during which signals are not to be driven until devices sharing the interface have had time to power up.
3569 To facilitate system implementation, the start of this interval shall be synchronized by an external signal
3570 across devices.

10.2.7.1 Power-up control mechanisms

The device that provides the interface shall provide one or more of the following mechanisms to enable the system integrator to synchronize interface power-up among devices on the interface:

- **Device power supply pin**

  The device has a power supply pin that the system integrator can use to control power-up of the interface. The device shall hold its outputs in a high-impedance state (current < I_z) for at least T_{pwrz} seconds after the power supply has initially reached its operating level (where the power supply operating level is specified by the device manufacturer).

- **Device reset pin or another similar signal**

  The device has a reset pin or other signal that the system integrator can use to control the power-up of the interface. This signal shall be able to be driven asserted during interface power-up and de-asserted afterward. The device shall hold its outputs in a high-impedance state (current < I_z) for at least T_{pwrz} seconds after the signal has been de-asserted, other than as described in 10.2.7.2. It is highly recommended that a single signal be used; however, an implementation is allowed to use a combination of signals if required. Logic levels for the signals are as specified by the device manufacturer.

- **REF_CLK detection**

  The device can elect to detect the presence of an active REF_CLK and use that for determining whether NC-SI power up has occurred. It is recommended that the device should count at least 100 clocks and continue to hold its outputs in a high-impedance state (current < I_z) for at least T_{pwrz} seconds more (Informational: 100 clocks at 50 MHz is 2 us).

10.2.7.2 Power-up transients

It is possible that a device may briefly drive its outputs while the interface or device is first receiving power, due to ramping of the power supply and design of its I/O buffers. It is recommended that devices be designed so that such transients, if present, are less than V_{pwrt} and last for no more than T_{pwrt}.

10.2.8 REF_CLK startup

REF_CLK shall start up, run, and meet all associated AC and DC specifications within $T_{clkstrt}$ seconds of interface power up.

10.3 Implementation guidance

This specification does not define implementation requirements due to the wide variation in architectures, devices and materials used. Following good engineering practices are a key part of a successful NC-SI implementation:

- Care must be taken in placement and layout
- Do a complete signal integrity analysis including determining what, if any, termination is required
- Minimize stubs
- Have uniform clock trace lengths
- Minimize noise on high-impedance0 signals

3608

ANNEX A
(normative)

Extending the model

This annex explains how the model can be extended to include vendor-specific content.

Commands extension

A Network Controller vendor may implement extensions and expose them using the OEM command, as described in 0.

Design considerations

This clause describes certain design considerations for vendors of Management Controllers.

PHY support

Although not a requirement of this specification, a Management Controller vendor may want to consider designing an NC-SI in such a manner that it could also be configured for use with a conventional RMII PHY. This would enable the vendor's controller to also be used in applications where a direct, non-shared network connection is available or preferred for manageability.

Multiple Management Controllers support

Currently, there is no requirement for Management Controllers to be able to put their TXD output lines and other output lines into a high-impedance state, because the present definition assumes only one Management Controller on the bus. However, component vendors may want to consider providing such control capabilities in their devices to support possible future system topologies where more than one Management Controller shares the bus to enable functions such as Management Controller fail-over or to enable topologies where more than one Management Controller can do NC-SI communications on the bus. If a vendor elects to make such provision, it is recommended that the TXD line and the remaining output lines be independently and dynamically switched between a high-impedance state and re-enabled under firmware control.

ANNEX B
(informative)

Relationship to RMII Specification

Differences with the *RMII Specification*

The following list presents key differences and clarifications between the *NC-SI Specification* and sections in the *RMII Specification*. (Section numbers refer to the *RMII Specification*.)

- General: Where specifications from IEEE 802.3 apply, this specification uses the version specified in clause 2, rather than the earlier IEEE 802.3u version that is referenced by RMII.
- Section 1.0:
  - The *NC-SI Specification* requires 100 Mbps support, but it does not specify a required minimum. (10 Mbps support is not required by NC-SI.)
  - Item 4. (Signals may or may not be considered to be TTL. NC-SI is not 5-V tolerant.)
- Section 2.0:
  - Comment: NC-SI chip-to-chip includes considerations for multi-drop and allows for non-PCB implementations and connectors (that is, not strictly point-to-point).
- Section 3.0:
  - Note/Advisory: The NC-SI clock is provided externally. An implementation can have REF_CLK provided by one of the devices on the bus or by a separate device.
- Section 5.0:
  - For NC-SI, the term *PHY* is replaced by *Network Controller*.
- Table 1:
  - The information in Table 1 in the *RMII Specification* is superseded by tables in this specification.
- Section 5.1, paragraph 2:
  - The *NC-SI Specification* allows 100 ppm. This supersedes the *RMII Specification*, which allows 50 ppm.
- Section 5.1, paragraph 3:
  - The NC-SI inherits the same requirements. The NC-SI MTU is required only to support Ethernet MTU with VLAN, as defined in the IEEE 802.3 version listed in clause 2.
- Section 5.1 paragraph 4:
  - The *RMII Specification* states: "During a false carrier event, CRS_DV shall remain asserted for the duration of carrier activity." This statement is not applicable to full-duplex operation of the NC-SI. CRS_DV from the Network Controller is used only as a data valid (DV) signal. Because the Carrier Sense aspect of CRS_DV is not used for full-duplex operation of the NC-SI, the Network Controller would not generate false carrier events for the NC-SI. However, it is recommended that the MAC in the Management Controller be able to correctly detect and handle these patterns if they occur, as this would be part of enabling the Management Controller MAC to also be able to work with an RMII PHY.

- Section 5.2:
  - The NC-SI does not specify a 10 Mbps mode. The Carrier Sense aspect of CRS_DV is not used for full-duplex operation of NC-SI.
- Section 5.3.1:
  - While the NC-SI does not specify Carrier Sense usage of CRS_DV, it is recommended that a Management Controller allow for CRS_DV toggling, in which CRS_DV toggles at 1/2 clock frequency, and that Management Controller MACs tolerate this and realign bit boundaries correctly in order to be able to work with an RMII PHY also.
- Section 5.3.2:
  - There is no 10 Mbps mode specified for the NC-SI.
- Section 5.3.3:
  - Generally, there is no expectation that the Network Controller will generate these error conditions for the NC-SI; however, the MAC in the Management Controller should be able to correctly detect and handle these patterns if they occur.
- Section 5.3.3:
  - The NC-SI does not specify or require support for RMII Registers.
- Section 5.5.2:
  - Ignore (N/A) text regarding 10 Mbps mode. The NC-SI does not specify or require interface operation in 10 Mbps mode.
- Section 5.6:
  - The Network Controller will not generate collision patterns for the specified full-duplex operation of the NC-SI; however, the MAC in the Management Controller should be able to detect and handle these patterns if they occur in order to be able to work with an RMII PHY also.
- Section 5.7:
  - NC-SI uses the IEEE 802.3 version listed in clause 2 instead of 802.3u as a reference.
- Section 5.8:
  - Loopback operation is not specified for the NC-SI.
- Section 7.0:
  - The NC-SI electrical specifications (clause 10) take precedence. (For example, section 7.4.1 in the *RMII Specification* for capacitance is superseded by *NC-SI Specification* 25 pF and 50 pF target specifications.)
- Section 8.0:
  - NC-SI uses the IEEE 802.3 version listed in clause 2 as a reference, instead of 802.3u.

ANNEX C
(informative)

Change log

Version	Date	Description
1.0.0	2009-07-21	
1.0.1	2013-01-24	DMTF Standard release
1.1.0	2015-09-23	DMTF Standard release
1.2.0b	2020-08-04	DMTF Work in Progress release

Bibliography

IANA, Internet Assigned Numbers Authority (www.iana.org). A body that manages and organizes numbers associated with various Internet protocols.

DMTF DSP4014, *DMTF Process for Working Bodies 2.2*, August 2015, http://www.dmtf.org/sites/default/files/standards/documents/DSP4014_2.2.0.pdf