



**International
Standard**

ISO/IEC 24079

Information technology — Network Controller Sideband Interface (NC- SI) specifications collection

*Technologies de l'information — Collection de spécifications
pour l'interface entre contrôleur réseau et contrôleur de gestion
système (NC-SI)*

**First edition
2024-07**

Reference number
ISO/IEC 24079:2024(en)

Full document details: ISO IEC 24079-2024

 **Documents**
© ISO/IEC 2024 Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by DMTF [as DMFT Network Controller Sideband Interface (NC-SI) Specifications Collection] and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Contents

Foreword..... xi

Introduction xiii

1 Scope 1

2 Normative references..... 2

3 Terms and definitions..... 3

 3.1 Requirement term definitions..... 3

 3.2 NC-SI term definitions 4

 3.3 Numbers and number bases 6

 3.4 Reserved fields..... 6

4 Symbols and abbreviated terms 6

5 Network Controller Sideband Interface (NC-SI) Specification (DSP0222) 8

 5.1 NC-SI overview..... 8

 5.2 Operational behaviors 14

 5.3 Arbitration in configurations with multiple Network Controller packages 35

 5.4 Packet definitions 45

 5.5 Packet-based and op-code timing..... 105

 5.6 RBT Electrical specification 107

6 Network Controller Sideband Interface (NC-SI) Collection (DSP0261)..... 114

 6.1 NC-SI over MCTP overview 114

 6.2 NC-SI over MCTP specific considerations 116

 6.3 Supported NC-SI commands..... 126

 6.4 Message types 128

 6.5 NC-SI support specific to MCTP transport 132

 6.6 Packet-Based Timing Specific to MCTP Binding 135

ANNEX A (normative) Extending the model..... 136

 A.1 Commands extension..... 136

 A.2 Design considerations 136

ANNEX B (informative) Relationship to RMII Specification 137

ANNEX C (informative) Notation and conventions 139

Bibliography 140

Figures

Figure 1 – NC-SI functional block diagram.....	9
Figure 2 – NC-SI traffic flow diagram.....	10
Figure 3 – Example topologies supported by the NC-SI	11
Figure 4 – Network Controller integration options	12
Figure 5 – NC-SI transport stack	13
Figure 6 – NC-SI operational state diagram.....	18
Figure 7 – NC-SI operational state diagram for hardware arbitration operation	19
Figure 8 – MC steps when the MC does not have prior knowledge of hardware arbitration	27
Figure 9 – NC-SI packet filtering flowchart.....	32
Figure 10 – Basic multi-drop block diagram	36
Figure 11 – Multiple Network Controllers in a ring format	37
Figure 12 – Op-code to RXD relationship	39
Figure 13 – Example TOKEN to transmit relationship	42
Figure 14 – Hardware arbitration state machine	43
Figure 15 – Ethernet frame encapsulation of NC-SI packet data without VLAN tag.....	45
Figure 16 – Example NC-SI signal interconnect topology.....	107
Figure 17 – DC measurements.....	109
Figure 18 – AC measurements.....	110
Figure 19 – Overshoot measurement.....	112
Figure 20 – Undershoot measurement	113
Figure 21 – NC-SI over RBT traffic flow diagram	115
Figure 22 – NC-SI over MCTP traffic flow diagram	116
Figure 23 – Single MCTP EID to multiple NC-SI channels mapping	117
Figure 24 – Multiple MCTP EIDs to multiple NC-SI channels mapping.....	118
Figure 25 – Multiple MCTP transport bindings example.....	121

Tables

Table 1 – Component documents	xi
Table 2 – NC-SI operating state descriptions.....	15
Table 3 – Channel ID format	20
Table 4 – Channel Ready state configuration settings	21
Table 5 – Hardware arbitration di-bit encoding	38
Table 6 – Hardware arbitration op-code format	38
Table 7 – Hardware arbitration states.....	44
Table 8 – Hardware arbitration events.....	44
Table 9 – Ethernet header format	46
Table 10 – Control packet header format	47
Table 11 – Generic example of control packet payload	48
Table 12 – Generic example of response packet payload format	50
Table 13 – Reason code ranges	51
Table 14 – Standard response code values	51
Table 15 – Standard reason code values	52
Table 16 – AEN packet format	52
Table 17 – AEN types	53
Table 18 – Command and response types	53
Table 19 – Example of complete minimum-sized NC-SI command packet	55
Table 20 – Example of complete minimum-sized NC-SI response packet.....	55
Table 21 – Clear Initial State command packet format	56
Table 22 – Clear Initial State response packet format.....	56
Table 23 – Select Package command packet format.....	57
Table 24 – Hardware arbitration disable byte.....	58
Table 25 – Select Package response packet format	58
Table 26 – Deselect Package command packet format.....	59
Table 27 – Deselect Package response packet format	59
Table 28 – Enable Channel command packet format.....	59
Table 29 – Enable Channel response packet format	60
Table 30 – Disable Channel command packet format.....	60
Table 31 – Disable Channel response packet format	61
Table 32 – Reset Channel command packet format	61
Table 33 – Reset Channel response packet format	61

Table 34 – Enable Channel Network TX command packet format	62
Table 35 – Enable Channel Network TX response packet format	62
Table 36 – Disable Channel Network TX command packet format	62
Table 37 – Disable Channel Network TX response packet format	63
Table 38 – AEN Enable command packet format	63
Table 39 – Format of AEN control	64
Table 40 – AEN Enable response packet format	64
Table 41 – Set Link command packet format	65
Table 42 – Set Link bit definitions	65
Table 43 – OEM Set Link bit definitions	66
Table 44 – Set Link response packet format	66
Table 45 – Set Link command-specific reason codes	66
Table 46 – Get Link Status command packet format	67
Table 47 – Get Link Status response packet format	67
Table 48 – Link Status field bit definitions	67
Table 49 – Other Indications field bit definitions	71
Table 50 – OEM Link Status field bit definitions (optional)	71
Table 51 – Get Link Status command-specific reason code	71
Table 52 – IEEE 802.1q VLAN fields	72
Table 53 – Set VLAN Filter command packet format	72
Table 54 – Possible settings for Filter Selector field (8-bit field)	72
Table 55 – Possible settings for Enable (E) field (1-bit field)	72
Table 56 – Set VLAN Filter response packet format	73
Table 57 – Set VLAN Filter command-specific reason code	73
Table 58 – Enable VLAN command packet format	73
Table 59 – VLAN Enable modes	73
Table 60 – Enable VLAN response packet format	74
Table 61 – Disable VLAN command packet format	74
Table 62 – Disable VLAN response packet format	75
Table 63 – Set MAC Address command packet format	76
Table 64 – Possible settings for MAC Address Number (8-bit field)	76
Table 65 – Possible settings for Address Type (3-bit field)	76
Table 66 – Possible settings for Enable Field (1-bit field)	77
Table 67 – Set MAC Address response packet format	77

Table 68 – Set MAC Address command-specific reason code.....	77
Table 69 – Enable Broadcast Filter command packet format	77
Table 70 – Broadcast Packet Filter Settings field.....	78
Table 71 – Enable Broadcast Filter response packet format	79
Table 72 – Disable Broadcast Filter command packet format	79
Table 73 – Disable Broadcast Filter response packet format	79
Table 74 – Enable Global Multicast Filter command packet format	80
Table 75 – Bit definitions for Multicast Packet Filter Settings field.....	81
Table 76 – Enable Global Multicast Filter response packet format.....	82
Table 77 – Disable Global Multicast Filter command packet format	83
Table 78 – Disable Global Multicast Filter response packet format.....	83
Table 79 – Set NC-SI Flow Control command packet format	84
Table 80 – Values for the Flow Control Enable field (8-bit field).....	84
Table 81 – Set NC-SI Flow Control response packet format.....	85
Table 82 – Set NC-SI Flow Control command-specific reason code	85
Table 83 – Get Version ID command packet format	85
Table 84 – Get Version ID response packet format.....	85
Table 85 – Get Capabilities command packet format	87
Table 86 – Get Capabilities response packet format.....	88
Table 87 – Capabilities Flags bit definitions.....	88
Table 88 – VLAN Mode Support bit definitions	90
Table 89 – Get Parameters command packet format	90
Table 90 – Get Parameters response packet format.....	91
Table 91 – Get Parameters data definition	92
Table 92 – MAC Address Flags bit definitions	92
Table 93 – VLAN Tag Flags bit definitions.....	92
Table 94 – Configuration Flags bit definitions	93
Table 95 – Get Controller Packet Statistics command packet format.....	93
Table 96 – Get Controller Packet Statistics response packet format	94
Table 97 – Get Controller Packet Statistics counters	95
Table 98 – Counters Cleared from Last Read Fields format	96
Table 99 – Get NC-SI Statistics command packet format.....	97
Table 100 – Get NC-SI Statistics response packet format	97
Table 101 – Get NC-SI Statistics counters.....	98

Table 102 – Get NC-SI Pass-through Statistics command packet format.....	98
Table 103 – Get NC-SI Pass-through Statistics response packet format	99
Table 104 – Get NC-SI Pass-through Statistics counters	99
Table 105 – Get Package Status packet format.....	100
Table 106 – Get Package Status response packet format	101
Table 107 – Package Status field bit definitions	101
Table 108 – OEM command packet format.....	101
Table 109 – OEM response packet format	102
Table 110 – PLDM Request packet format	102
Table 111 – PLDM Response packet format.....	102
Table 112 – Get Package UUID command packet format	103
Table 113 – Get Package UUID response packet format.....	103
Table 114 – UUID format	104
Table 115 – Link Status Change AEN packet format.....	104
Table 116 – Configuration Required AEN packet format	105
Table 117 – Host Network Controller Driver Status Change AEN packet format.....	105
Table 118 – Host Network Controller Driver Status format	105
Table 119 – NC-SI packet-based and op-code timing parameters	106
Table 120 – Physical NC-SI signals	108
Table 121 – DC specifications	109
Table 122 – AC specifications	111
Table 123 – MCTP Message types for NC-SI over MCTP	116
Table 124 – Ordering rules	120
Table 125 – Supported NC-SI commands	126
Table 126 – NC-SI messages encapsulation.....	129
Table 127 – MCTP Transport Header fields	129
Table 128 – MCTP Specific Message Header field	129
Table 129 – Ethernet messages encapsulation	130
Table 130 – MCTP Transport Header fields	131
Table 131 – MCTP Specific Message Header field.....	131
Table 132 – Get Supported Media Command packet format	132
Table 133 – Get Supported Media Response packet format	132
Table 134 – Get Supported Media Response media descriptors format	133
Table 135 – Transport Specific AENs Enable Command packet format	133

Table 136 – Transport Specific AENs Enable field format 134

Table 137 – Transport-specific AENs Enable Response packet format 134

Table 138 – Medium change AEN format.....134

Table 139 – NC-SI Timing Parameters Specific to MCTP Binding 135

Foreword

The *Network Controller Sideband Interface (NC-SI) Specification* 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.

Component documents

Table 1 lists the component documents for this specification:

Table 1 – Component documents

Document number	Document title	Version
DSP0222	Network Controller Sideband Interface (NC-SI) Specification (DSP0222)	1.1.1
DSP0261	Network Controller Sideband Interface (NC-SI) Collection (DSP0261)	1.2.3

Acknowledgments

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

Editors:

- Eliel Louzoun – Intel Corporation
- Hemal Shah – Broadcom Corporation
- Bob Stevens – Dell
- Tom Slaight – Intel Corporation

Contributors:

- Alan Berenbaum – SMSC
- Patrick Caporale – Lenovo
- Philip Chidester – Dell Technologies
- Kelly Couch – Intel Corporation
- Benzi Friedman – Intel Corporation
- Yuval Itkin – Mellanox Technologies and NVIDIA Corporation
- Ira Kalman – Intel Corporation
- Ed Klodnicki – IBM
- Joe Kozlowski – Dell Technologies
- Patrick Kutch – Intel Corporation
- John Leung – Intel Corporation
- Eliel Louzoun – Intel Corporation
- Patrick Schoeller – Hewlett-Packard Company and Intel Corporation

- Hemal Shah – Broadcom Inc.
- Tom Slight – Intel Corporation
- Bob Stevens – Dell Technologies

Introduction

Network Controller Sideband Interface (NC-SI) Specification (DSP0222)

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.

NC-SI over MCTP Binding Specification (DSP0261)

The *NC-SI over MCTP Binding Specification* defines new MCTP messages used to convey NC-SI Control packets and Ethernet traffic over MCTP to allow NC-SI Pass-through traffic over MCTP. This specification is based on the [DSP0222 1.1](#) specification and uses the same NC-SI Control packet definitions.

Document conventions

Typographical conventions

The following typographical conventions are used in this document:

- Document titles are marked in *italics*.
- Important terms that are used for the first time are marked in *italics*.
- Terms include a link to the term definition in Terms and definitions, enabling easy navigation to the term definition.
- ABNF rules are in `monospaced font`.

ABNF usage conventions

Format definitions in this document are specified using ABNF (see [RFC5234](#)), with the following deviations:

- Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the definition in [RFC5234](#) that interprets literal strings as case-insensitive US-ASCII characters.

Reserved and unassigned values

Unless otherwise specified, any reserved, unspecified, or unassigned values in enumerations or other numeric ranges are reserved for future definition by the DMTF.

Unless otherwise specified, numeric or bit fields that are designated as reserved shall be written as 0 (zero) and ignored when read.

Byte ordering

Unless otherwise specified, byte ordering of multibyte numeric fields or bit fields is "Big Endian" (that is, the lower byte offset holds the most significant byte, and higher offsets hold lesser significant bytes).

Other conventions

See ANNEX C
(informative)

Notation and conventions for other conventions.

Information technology — Network Controller Sideband Interface (NC-SI) Specifications Collection

1 Scope

Network Controller Sideband Interface (NC-SI) Specification (DSP0222)

This specification defines the functionality and behavior of the Sideband Interface responsible for connecting the Network Controller 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 RMT 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.

NC-SI over MCTP Binding Specification (DSP0261)

This specification defines the bindings between NC-SI protocol elements and MCTP elements in order for NC-SI Control and Pass-Through traffic to be transported using MCTP.

Portions of this specification rely on information and definitions from other specifications, which clause 2 identifies. Two of these references are particularly relevant:

- DMTF [DSP0222](#), *Network Controller Sideband Interface (NC-SI) Specification*, provides the NC-SI base control that is to be bound over MCTP by this specification.
- DMTF [DSP0236](#), *Management Component Transport Protocol (MCTP) Base Specification*, defines the MCTP transport on which the NC-SI Control and Pass-through packets are to be conveyed.

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.

ACPI, *Advanced Configuration and Power Interface Specification Revision 4.0a*, April 5, 2010,
<http://www.acpi.info/DOWNLOADS/ACPIspec40a.pdf>

DMTF DSP0004, *CIM Infrastructure Specification 3.0*,
http://www.dmtf.org/standards/published_documents/DSP0004_3.0.pdf

DMTF DSP0222, *Network Controller Sideband Interface (NC-SI) Specification 1.1*,
http://www.dmtf.org/sites/default/files/standards/documents/DSP0222_1.1.0.pdf

DMTF DSP0223, *Generic Operations 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification 1.3*,
http://www.dmtf.org/standards/published_documents/DSP0236_1.3.pdf

DMTF DSP0237, *Management Component Transport Protocol (MCTP) SMBus/I2C Transport Binding Specification 1.1*, http://www.dmtf.org/standards/published_documents/DSP0237_1.1.pdf

DMTF DSP0238, *Management Component Transport Protocol (MCTP) PCIe VDM Transport Binding Specification 1.0*, http://www.dmtf.org/standards/published_documents/DSP0238_1.0.pdf

DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes 1.4*,
http://www.dmtf.org/standards/published_documents/DSP0239_1.4.pdf

DMTF DSP0240, *Platform Level Data Model (PLDM) Base 5 Specification 1.1.0*,
https://www.dmtf.org/sites/default/files/standards/documents/DSP0240_1.1.0.pdf

DMTF DSP0261, *NC-SI over MCTP Binding Specification 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0261_1.0.pdf

DMTF DSP1001, *Management Profile Specification Usage Guide 1.2*,
http://www.dmtf.org/standards/published_documents/DSP1001_1.2.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 2005,
<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 Virtual LAN (VLAN) Bridges that permit the definition, operation and administration of Virtual LAN topologies within a Bridged LAN infrastructure.

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

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

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

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

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

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

IETF RFC5234, *ABNF: Augmented BNF for Syntax Specifications*, January 2008, <http://tools.ietf.org/html/rfc5234>

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

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

3 Terms and definitions

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

In this document, some terms have a specific meaning beyond the normal English meaning. This clause defines those terms.

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

The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as described in [ISO/IEC Directives, Part 2](#), Clause 6.

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

The terms that [DSP0004](#), [DSP0223](#), [DSP0236](#), and [DSP1001](#) apply to this document. The following additional 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

deprecated

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

3.1.2

obsolete

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

3.2 NC-SI term definitions

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

3.2.1

Asynchronous Event Notification

control packet sent by the Network Controller to the Management Controller as an explicit notification of the occurrence of an event of interest to the Management Controller

3.2.2

channel

the control logic and data paths that support NC-SI Pass-through operations through a single network interface (port)

Note 1 to entry: A Network Controller that has multiple network interface ports can support an equivalent number of NC-SI channels.

3.2.3

command

control packet sent by the Management Controller to the Network Controller to request the Network Controller to perform an action, and/or return data

3.2.4

control traffic

control packets

command, response, and asynchronous event notification packets transmitted between the Management Controller and Network Controllers for the purpose of managing the NC-SI

3.2.5

external network interface

the interface of the Network Controller that provides connectivity to the external network infrastructure

Note 1 to entry: also known as *port*.

3.2.6

frame

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

Note 1 to entry: *Frame* is used in references to [IEEE 802.3 Frames](#). *Packet* is used in all other references.

3.2.7

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)

3.2.8

internal host interface

the interface of the Network Controller that provides connectivity to the host operating system running on the platform

3.2.9

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

3.2.10

multi-drop

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

Note 1 to entry: Related to NC-SI, a Management Controller serves the role of the master, and the Network Controllers are the target devices.

3.2.11

Network Controller

the component within a system that is responsible for providing connectivity to an external Ethernet network

3.2.12

Network Controller Sideband Interface

NC-SI

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

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

Note 1 to entry: 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.

3.2.14

packet

a formatted block of information carried by a computer network

Note 1 to entry: *Frame* is used in references to [IEEE 802.3 Frames](#). *Packet* is used in all other references.

3.2.15

pass-through traffic

pass-through packets

network packets passed between the external network and the Management Controller through the Network Controller

3.2.16

point-to-point

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

3.2.17

RBT

RMII-Based Transport

electrical and timing specification for a 3.3V physical medium that is derived from [RMII](#)

3.2.18

remote media

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

3.2.19

response

control packet sent by the Network Controller to the Management Controller as a positive acknowledgment of a command received from the Management Controller, and to provide the processing outcome of the command, as well as to return any required data

3.2.20

System Power States

S0 and Sx

S0 represents an active system

Sx represents system power states S1 – S5, which reflects various levels of inactivity of a system

Note 1 to entry: The definition of the power states is as [ACPI](#) defines.

3.3 Numbers and number bases

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

3.4 Reserved fields

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

4 Symbols and abbreviated terms

The abbreviations that [DSP0004](#), [DSP0223](#), [DSP0236](#), and [DSP1001](#) define apply to this document. The following additional abbreviations are used in this document.

4.1

AC

alternating current

4.2

ACPI

Advanced Configuration and Power Interface

4.3

AEN

Asynchronous Event Notification

4.4

BMC

Baseboard Management Controller (often used interchangeably with MC)

4.5

CRC

cyclic redundancy check

4.6

CRS_DV

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

4.7

DC

direct current

4.8

DHCP

Dynamic Host Configuration Protocol

4.9

EEE

Energy Efficient Ethernet

4.10

FCS

Frame Check Sequence

4.11

IANA

Internet Assigned Numbers Authority

4.12

MC

Management Controller

4.13

MCTP

Management Component Transport Protocol

4.14

NC

Network Controller

4.15

NC-SI

Network Controller Sideband Interface

4.16

NC-SI RX

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

4.17

NC-SI TX

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

4.18

RID

PCIe Requester ID (Bus/Device/Function)

4.19

RMII

Reduced Media Independent Interface

4.20

RX

Receive

4.21

RXD

physical NC-SI signals used to transmit data from the Network Controller to the Management Controller

4.22

RX_ER

a physical NC-SI signal used to indicate a Receive Error

4.23

SerDes

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

4.24

TX

Transmit

4.25

TXD

physical NC-SI signals used to transmit data from the Management Controller to the Network Controller

4.26

VLAN

Virtual LAN

5 Network Controller Sideband Interface (NC-SI) Specification (DSP0222)

5.1 NC-SI overview

5.1.1 General

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

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

The specification of how the NC-SI protocol and messages are implemented over a particular physical medium is referred to as a transport binding. This document, DSP0222, includes the definition of the transport binding, electrical, framing, and timing specifications for a physical interface called RBT (RMII-based Transport). Electrically, RBT, as clause 5.6 describes, is similar to the Reduced Media Independent Interface™ (RMII)—see ANNEX B (informative)

Relationship to RMII Specification. Transport bindings for NC-SI over other media and transport protocols are defined through external transport binding specifications, such as [DSP0261](#), the *NC-SI over MCTP Transport Binding Specification*.

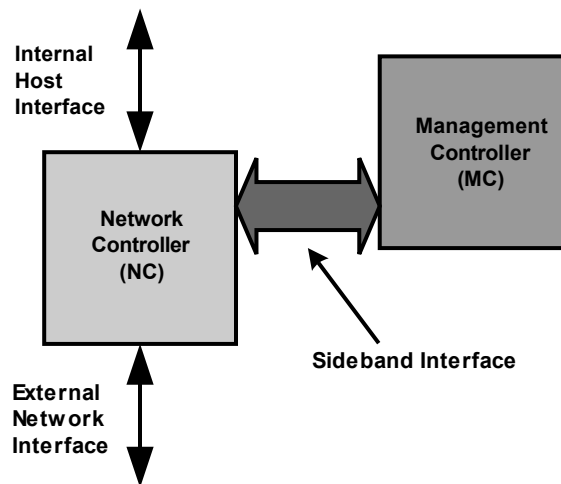


Figure 1 – NC-SI functional block diagram

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

- “Pass-through” packets that are transferred between the Management Controller and the external network
- “Control” packets that are transferred between the Management Controller and Network Controllers for control or configuration functionality. This specification defines a number of NC-SI commands and responses as well as a mechanism to customize and extend functionality via OEM commands—see ANNEX A (normative)

Extending the model.

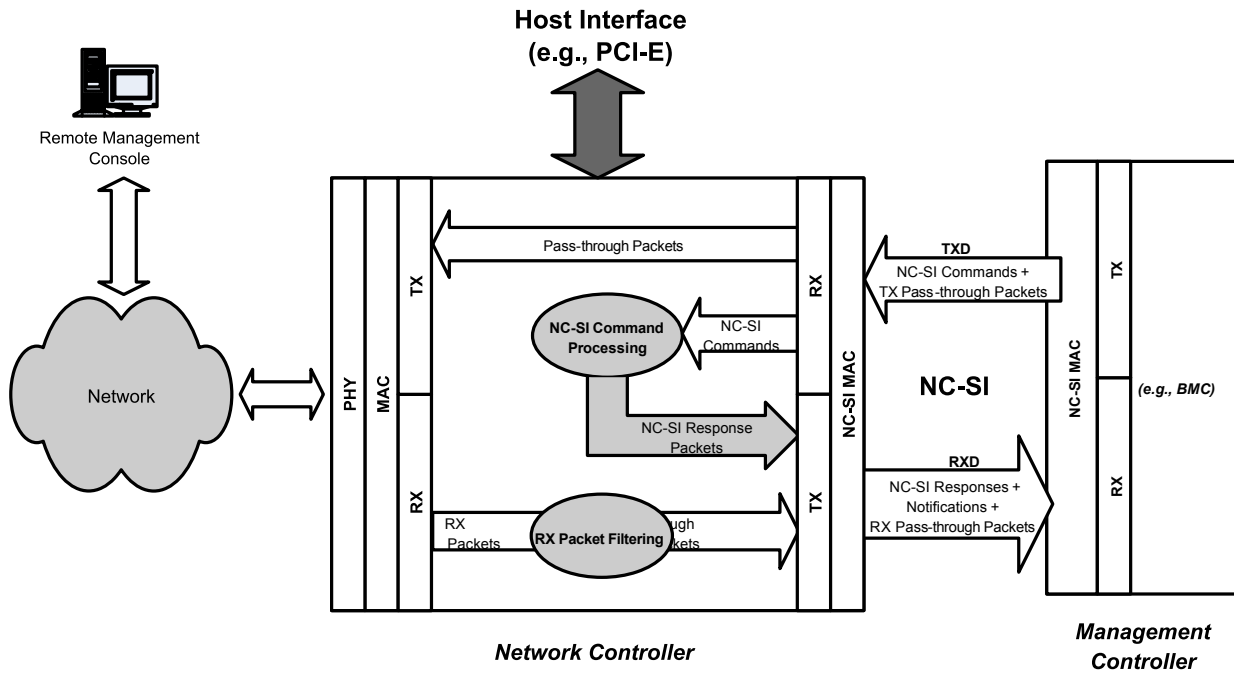


Figure 2 – NC-SI traffic flow diagram

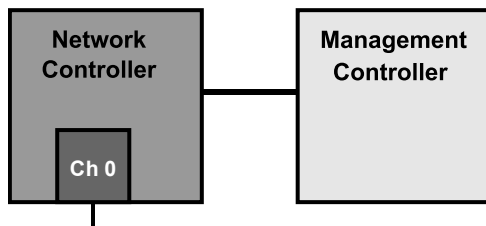
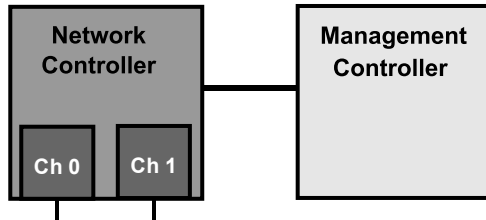
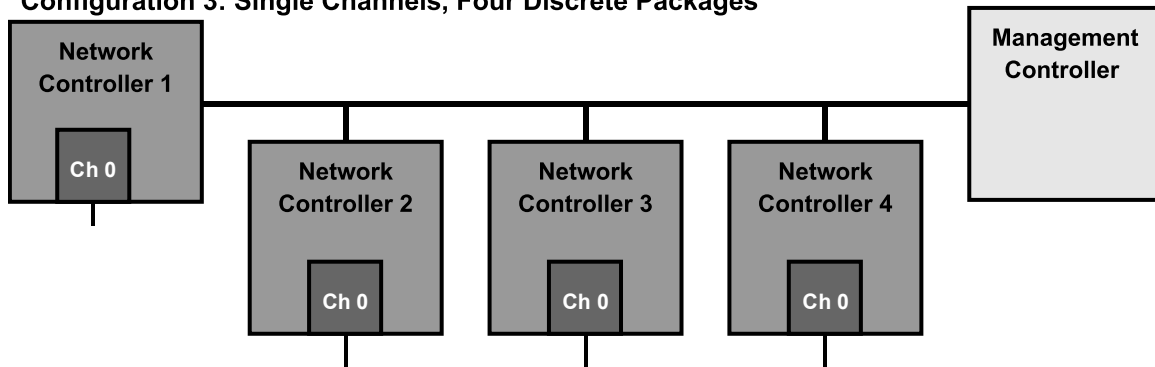
NC-SI is intended to operate independently from the in-band activities of the Network Controller. As such, the Sideband Interface is not specified to be accessible through the host interface of the Network Controller. From the external world, this interface should behave and operate like a standard Ethernet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

5.1.2 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 over NC-SI RBT. The 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 channels connections.
- Configuration 3 shows a Management Controller connecting to four discrete Network Controllers.

Configuration 1: Single Channel, Single Package**Configuration 2: Integrated Dual Channel, Single Package****Configuration 3: Single Channels, Four Discrete Packages****Figure 3 – Example topologies supported by the NC-SI**

5.1.3 Single and integrated Network Controller implementations

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

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

For the *NC-SI Specification*, an integrated controller can be logically implemented in one of three basic ways, as illustrated in Figure 4. Although only two channels are shown in the illustration, an integrated controller implementation can provide more than two channels. The example channel and package numbers (for example, channel 0, pkg 0) refer to the Internal Channel and Package ID subfields of the Channel ID. For more information, see 5.2.2.9.

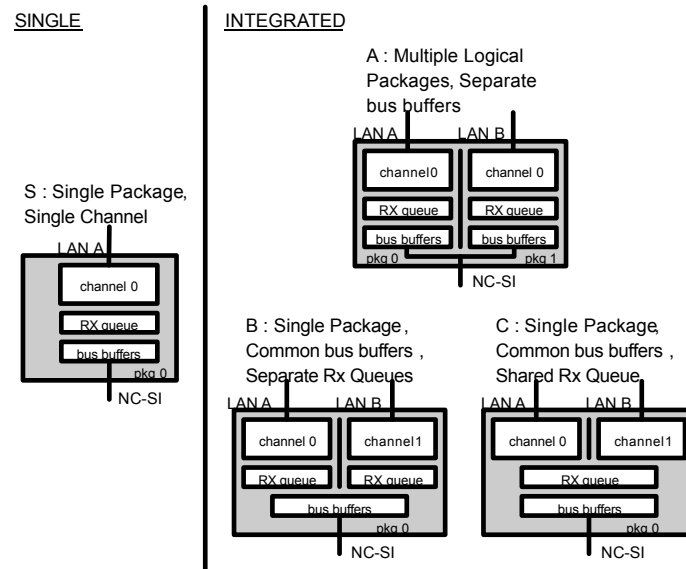


Figure 4 – Network Controller integration options

Packages that include multiple channels are required to handle internal arbitration between those channels and the NC-SI. 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 might 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 could 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 5.3.3.)

- **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 could also queue up AEN or Response packets.

5.1.4 Transport stack

Figure 5 illustrates the overall transport stack of the NC-SI. 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.

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

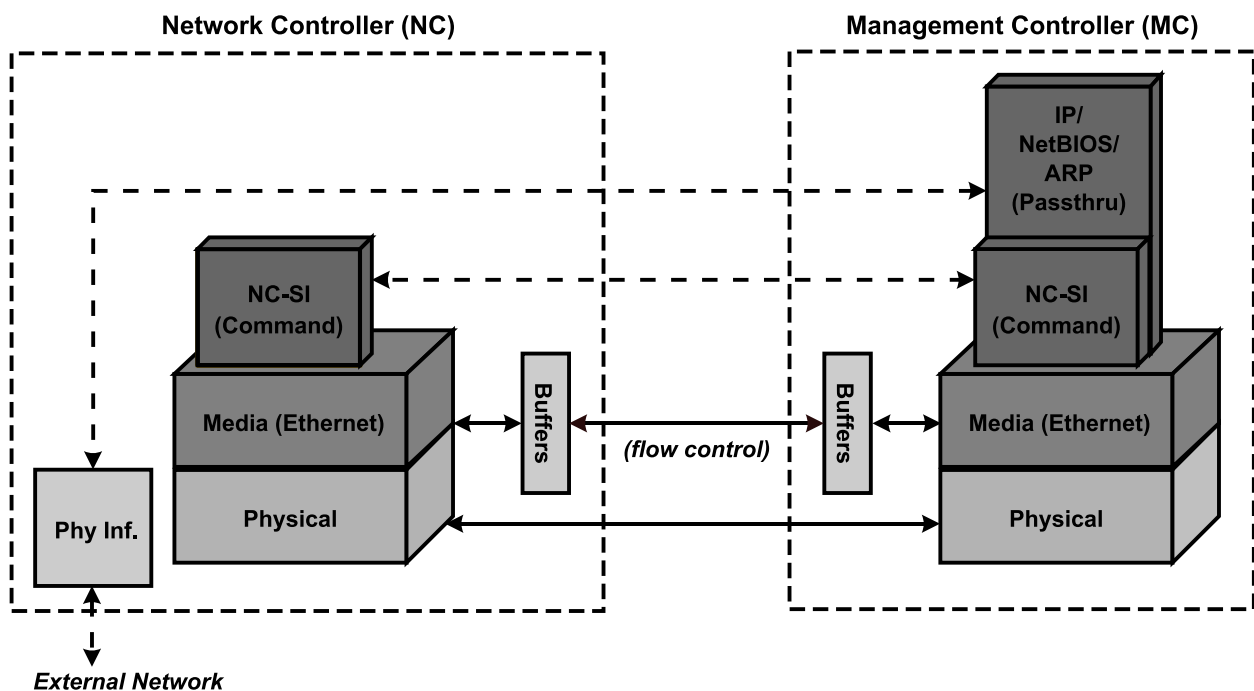


Figure 5 – NC-SI transport stack

5.1.5 Transport protocol

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

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

5.1.6 Byte and bit ordering for transmission

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

5.2 Operational behaviors

5.2.1 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.

- **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 could occur during operation or configuration. For example, a Network Controller might 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 5.2.2.4); 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.

5.2.2 State definitions

5.2.2.1 General

Table 2 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.

Table 2 – 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 clause 5.6 defines.
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). See 5.2.2.4 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.

5.2.2.2 NC-SI power states

The NC-SI defines only two power states:

- **NC-SI Interface Power Down 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 NC-SI interfaces on the Network Controllers and Management 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. The Network Controller is expected to transition to the Initial State within T4 seconds after the Power Up state is entered.

5.2.2.3 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 5.2.2.9).

5.2.2.4 Initial State

The Initial State for a channel corresponds to a condition in which the NC-SI 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 this specification explicitly defines the default configuration settings, the default values are implementation specific. The MC should not make any assumptions on any configuration settings that this specification does not define. Because this state may be entered at any time, the Initial State shall be acknowledged with a Clear Initial State command in order 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 5.2.2.9)
- 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. If the implementation does not have the global multicast or broadcast filters in “disabled” state in the Initial State, the MC might 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.

5.2.2.5 NC-SI Initial State recovery

As 5.2.2.4 describes, a channel in the Initial State shall receive the Clear Initial State command before other commands can be run.

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 5.4.2.5.

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 should 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 needs to be restored by the Management Controller.

5.2.2.6 State transition diagram

Figure 6 illustrates the general relationship between the package- and channel-related states described in Table 2 and the actions that cause transitions between the states. Each bubble in Figure 6 represents a particular combination of states as Table 2 defines.

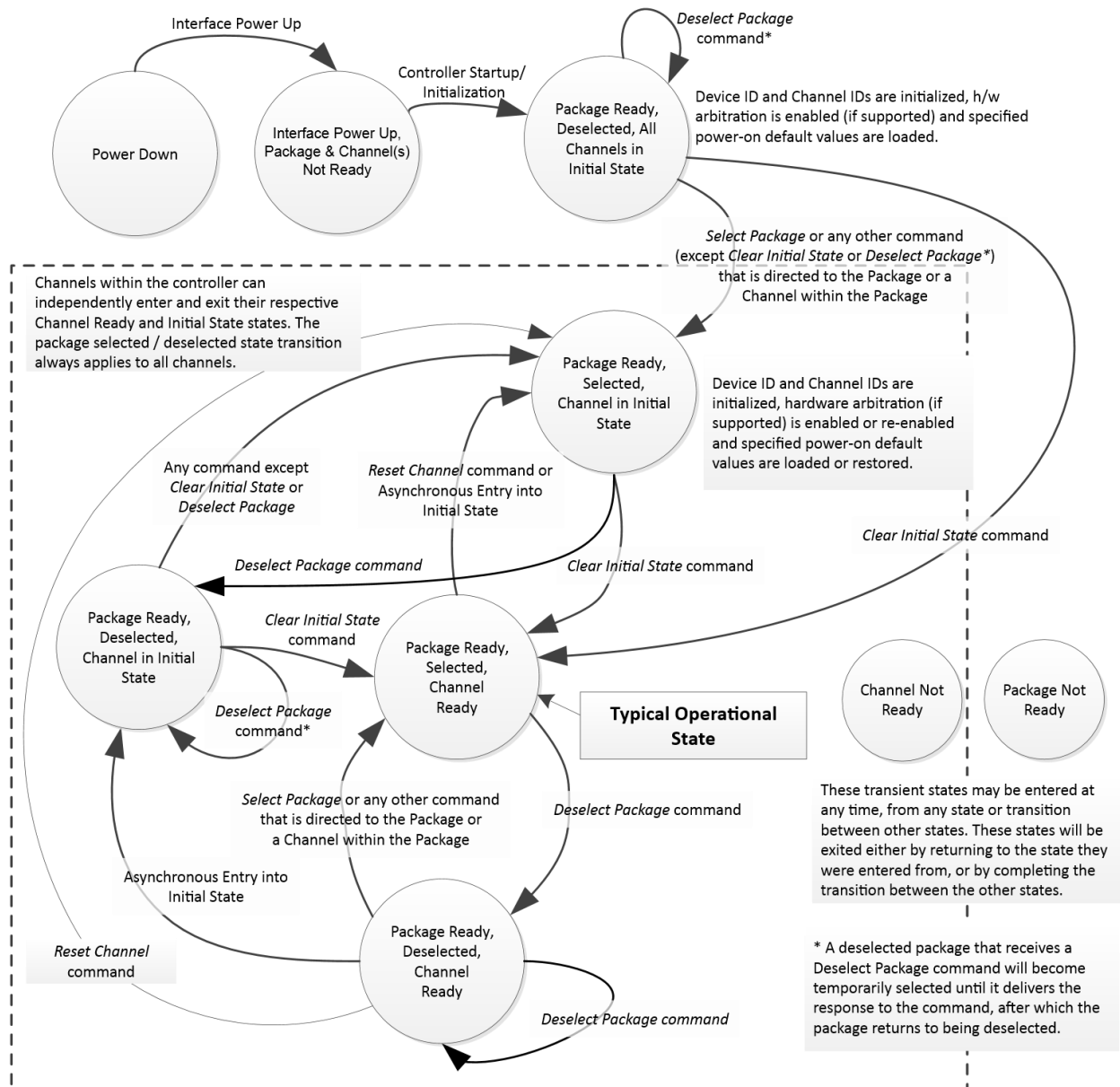


Figure 6 – NC-SI operational state diagram

5.2.2.7 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.

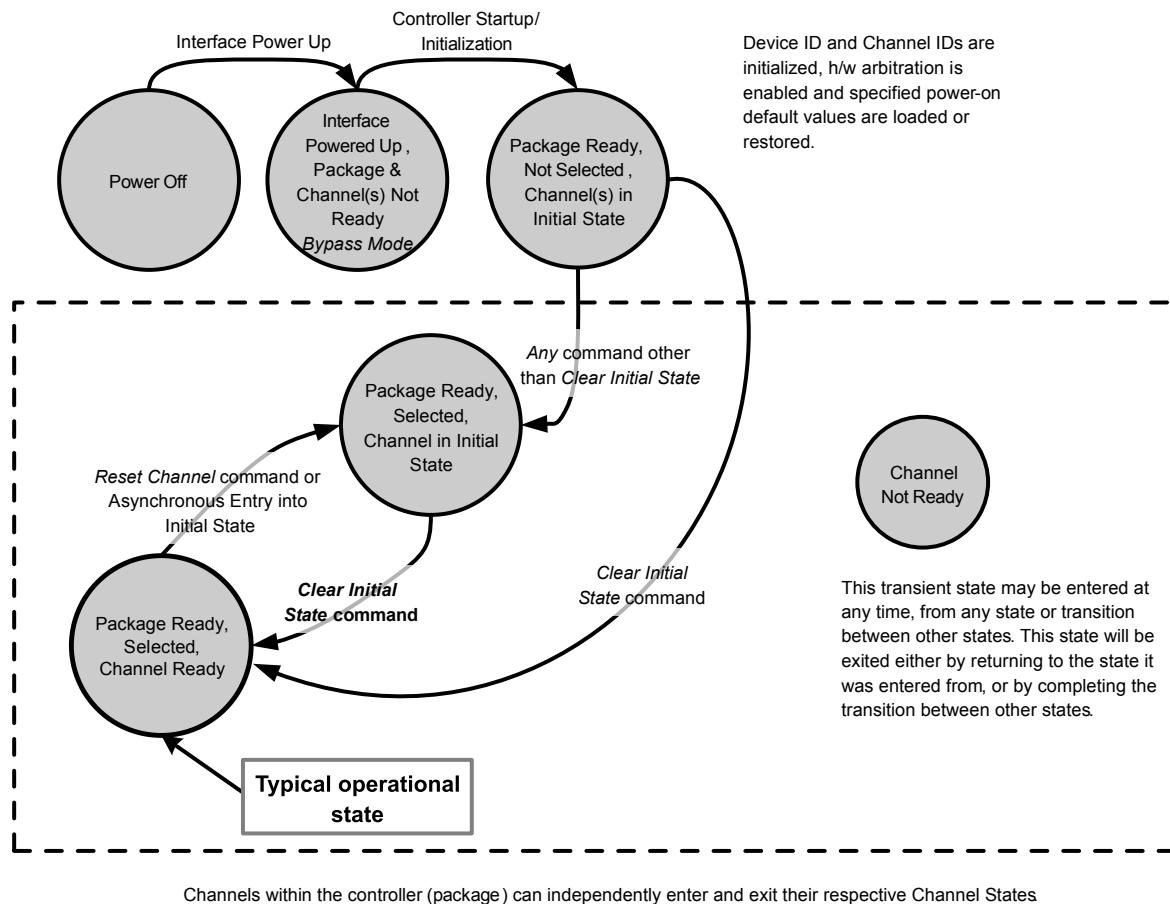


Figure 7 – NC-SI operational state diagram for hardware arbitration operation

While Figure 7 does not show Select and Deselect package commands, 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.

5.2.2.8 Resets

5.2.2.8.1 Asynchronous entry into Initial State

An Asynchronous Reset event is 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.

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.

5.2.2.8.2 Synchronous Reset

A Synchronous Reset event on the NC-SI is a Reset Channel command issued by a Management Controller to a Channel. Upon the receipt of this command, the Network Controller shall place 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.

5.2.2.8.3 Other Resets

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

5.2.2.9 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 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 Table 3 describes.

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 NC-SI).

Table 3 – 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 three 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 Pass-through 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.

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 NC-SI 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.

5.2.2.10 Configuration-related settings

5.2.2.10.1 Package-specific operation

Only two configuration settings are package-specific:

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

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 $0 \times 1F$.

5.2.2.10.2 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 $0 \times 1F$. Channel-specific commands with Invalid Channel IDs should not be allowed or executed.

Table 4 shows the major categories of configuration settings that control channel operation when a channel is in the Channel Ready state.

Table 4 – 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.

Setting/Configuration category	Description
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.

5.2.2.11 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.

5.2.2.12 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.

5.2.2.13 Startup sequence examples

5.2.2.13.1 Overview

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.

5.2.2.13.2 Typical non hardware 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 hooked to its NC-SI, or what capabilities those controllers support. Note that this is not the only possible 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 (see 5.6.2.7 for the specification of this condition). The Network Controller packages are provided a Device 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 Device Ready Interval to expire (see Table 119).

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 Device 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 5.4.4.5 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 to configure how Ethernet Pause Frames are used for flow control on the NC-SI.
- 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. Enable hardware arbitration for the packages

If only a single Network Controller package is discovered, the Management Controller does not need to enable hardware arbitration if the controller hardware supports it. In fact, the Management Controller may always elect to disable hardware arbitration, because then it does not need to be concerned with whether the implementation provided a 'loop back' of the hardware arbitration 'ARB_OUT' signal to the controller to the 'ARB_IN' signal.

If multiple packages are detected, and each package has reported that it supports hardware arbitration, then the hardware arbitration operation can be enabled by issuing a Select Package command, with the Hardware Arbitration parameter for the command set to 'enabled', to each package. Because hardware arbitration enables multiple packages to be selected simultaneously, sending Deselect Package commands is not necessary when hardware arbitration is being used.

NOTE There is no mandatory status to indicate whether hardware arbitration is hooked up and operating correctly. In that case, the Management Controller needs to have prior knowledge that the implementation routes the hardware arbitration signals between the packages.

7. 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.

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.

5.2.2.13.3 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 interfaced to the NC-SI, 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 actually 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.

5.2.2.13.4 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.

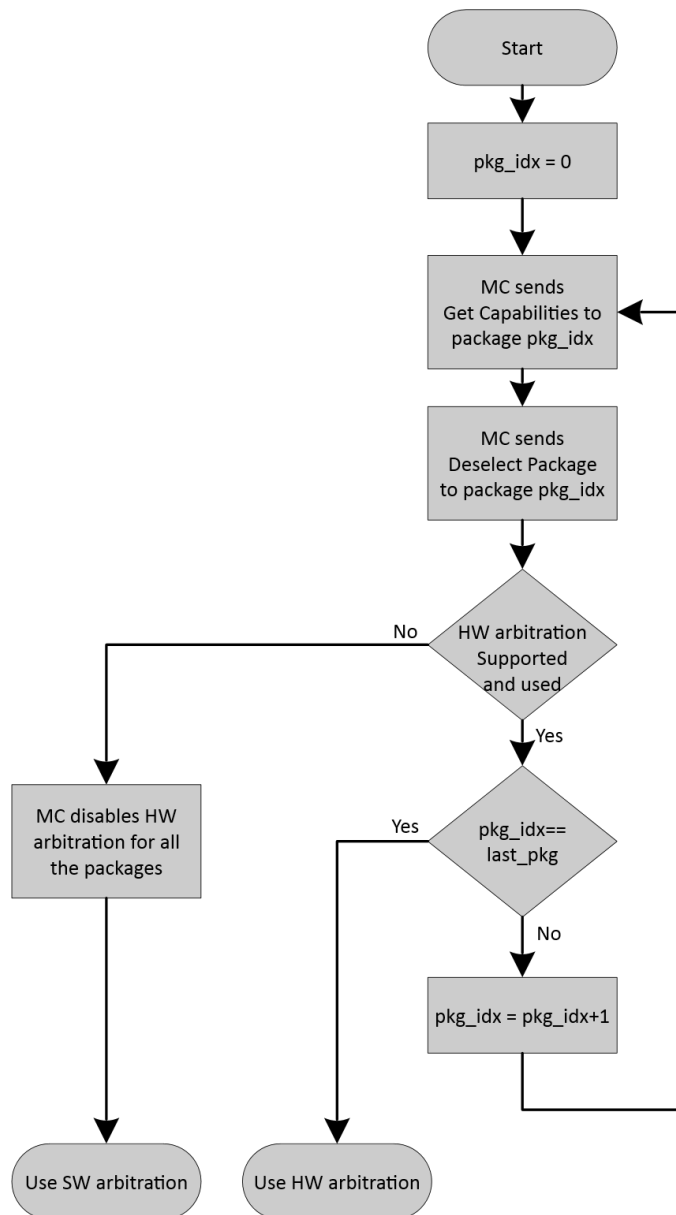


Figure 8 – MC steps when the MC does not have prior knowledge of hardware arbitration

5.2.3 NC-SI traffic types

5.2.3.1 Overview

Two types of traffic are carried on the NC-SI: 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 NC-SI.
- Control traffic consists of commands (requests) and responses that support the configuration and control of the NC-SI and Pass-through operation of the Network Controller, and AENs that support reporting various events to the Management Controller.

5.2.3.2 Command protocol

5.2.3.2.1 Overview

Commands are provided to allow a Management Controller to initialize, control, and regulate Management Controller packet flow across the NC-SI, 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.

5.2.3.2.2 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.
- 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.

5.2.3.2.3 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 5.2.9.3.2.

5.2.3.2.4 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 NC-SI 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.

5.2.3.2.5 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 119).

- 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.

5.2.3.2.6 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.

5.2.4 Link configuration and control

5.2.4.1 Link Configuration

The Network Controller provides commands to allow the Management Controller to specify the auto-negotiation, link speed, duplex settings, and so on to be used on the network interface. For more information, see 5.4.4.21. The Management Controller should make link configuration changes only when the host network driver is absent or non-operational.

5.2.4.2 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.

5.2.5 Frame filtering for Pass-through mode

5.2.5.1 Overview

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 NC-SI. See [RFC2373](#), [RFC2461](#), and [RFC3315](#) for IPv6-related definitions.

5.2.5.2 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.

5.2.5.3 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 (see [RFC2131](#)) and NetBIOS.

5.2.5.4 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.

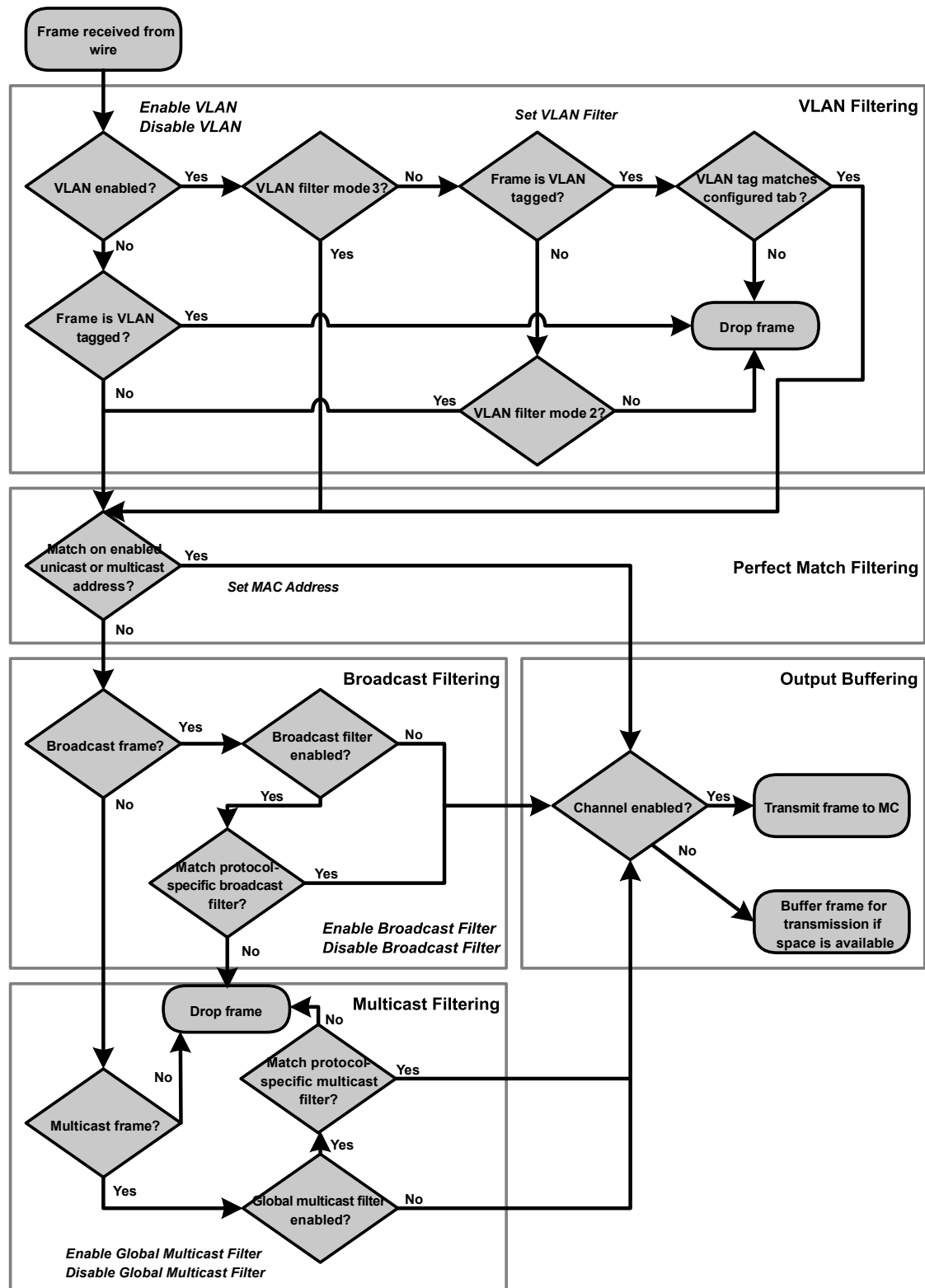


Figure 9 – NC-SI packet filtering flowchart

5.2.6 Output buffering behavior

There are times when the NC is not allowed to transmit Pass-through, AEN, or control packets onto the NC-SI.

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.

5.2.7 NC-SI flow control

The Network Controller may provide commands to enable flow control on the NC-SI between the Network Controller and the Management Controller. The NC-SI flow control behavior follows the PAUSE frame behavior as the [IEEE 802.3 specification](#) defines. Flow control is configured using the Set NC-SI Flow command (see 5.4.4.41).

5.2.8 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.