



1  
2  
3  
4

**Document Number: DSP0222**

**Date: 2009-07-21**

**Version: 1.0.0**

5 **Network Controller Sideband Interface (NC-SI)**  
6 **Specification**

7 **Document Type: Specification**  
8 **Document Status: DMTF Standard**  
9 **Document Language: E**

10 Copyright Notice

11 Copyright © 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

12 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
13 management and interoperability. Members and non-members may reproduce DMTF specifications and  
14 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to  
15 time, the particular version and release date should always be noted.

16 Implementation of certain elements of this standard or proposed standard may be subject to third party  
17 patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations  
18 to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose,  
19 or identify any or all such third party patent right, owners or claimants, nor for any incomplete or  
20 inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to  
21 any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize,  
22 disclose, or identify any such third party patent rights, or for such party's reliance on the standard or  
23 incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any  
24 party implementing such standard, whether such implementation is foreseeable or not, nor to any patent  
25 owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is  
26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  
27 implementing the standard from any and all claims of infringement by a patent owner for such  
28 implementations.

29 For information about patents held by third-parties which have notified the DMTF that, in their opinion,  
30 such patent may relate to or impact implementations of DMTF standards, visit  
31 <http://www.dmtf.org/about/policies/disclosures.php>.

32

## CONTENTS

34	Foreword .....	7
35	Introduction .....	8
36	1 Scope .....	9
37	2 Normative References.....	9
38	3 Terms and Definitions .....	10
39	3.1 Requirement Term Definitions .....	10
40	3.2 NC-SI Term Definitions.....	11
41	3.3 Numbers and Number Bases .....	13
42	3.4 Reserved Fields.....	13
43	4 Acronyms and Abbreviations .....	14
44	5 NC-SI Overview.....	15
45	5.1 Defined Topologies.....	17
46	5.2 Single and Integrated Network Controller Implementations .....	18
47	5.3 Transport Stack.....	20
48	5.4 Transport Protocol .....	21
49	6 Operational Behaviors.....	21
50	6.1 Typical Operational Model.....	21
51	6.2 State Definitions.....	22
52	6.3 NC-SI Traffic Types .....	33
53	6.4 Link Configuration and Control .....	35
54	6.5 Frame Filtering for Pass-through Mode.....	35
55	6.6 NC-SI Flow Control.....	37
56	6.7 Asynchronous Event Notification .....	37
57	6.8 Error Handling.....	37
58	7 Arbitration in Configurations with Multiple Network Controller Packages .....	39
59	7.1 General .....	39
60	7.2 Hardware Arbitration.....	40
61	7.3 Command-based Arbitration.....	49
62	8 Packet Definitions.....	49
63	8.1 NC-SI Packet Encapsulation .....	49
64	8.2 Control Packet Data Structure .....	51
65	8.3 Control Packet Type Definitions .....	57
66	8.4 Command and Response Packet Formats.....	58
67	8.5 AEN Packet Formats .....	106
68	9 Packet-Based and Op-Code Timing .....	108
69	10 Electrical Specification .....	109
70	10.1 Topologies .....	109
71	10.2 Electrical and Signal Characteristics and Requirements.....	110
72	ANNEX A (normative) Extending the Model .....	118
73	ANNEX B (informative) Relationship to RMII Specification .....	119
74	ANNEX C (informative) Change Log .....	121
75	Bibliography .....	122
76		

## 77 Figures

78	Figure 1 – NC-SI Functional Block Diagram .....	16
79	Figure 2 – NC-SI Traffic Flow Diagram .....	17
80	Figure 3 – Example Topologies Supported by the NC-SI .....	18
81	Figure 4 – Network Controller Integration Options .....	19
82	Figure 5 – NC-SI Transport Stack.....	20
83	Figure 6 – NC-SI Operational State Diagram .....	25
84	Figure 7 – NC-SI Operational State Diagram for Hardware Arbitration Operation .....	26
85	Figure 8 – NC-SI Packet Filtering Flowchart.....	36
86	Figure 9 – Basic Multi-Drop Block Diagram .....	39
87	Figure 10 – Multiple Network Controllers in a Ring Format .....	41
88	Figure 11 – Op-Code to RXD Relationship .....	42
89	Figure 12 – Example TOKEN to Transmit Relationship .....	46
90	Figure 13 – Hardware Arbitration State Machine.....	47
91	Figure 14 – Ethernet Frame Encapsulation of NC-SI Packet Data .....	50
92	Figure 15 – Example NC-SI Signal Interconnect Topology .....	110
93	Figure 16 – DC Measurements .....	112
94	Figure 17 – AC Measurements .....	114
95	Figure 18 – Overshoot Measurement .....	115
96	Figure 19 – Undershoot Measurement .....	116
97		

## 98 Tables

99	Table 1 – NC-SI Operating State Descriptions .....	22
100	Table 2 – Channel ID Format.....	27
101	Table 3 – Channel Ready State Configuration Settings .....	28
102	Table 4 – Hardware Arbitration Di-bit Encoding.....	41
103	Table 5 – Hardware Arbitration Op-Code Format.....	42
104	Table 6 – Hardware Arbitration States.....	48
105	Table 7 – Hardware Arbitration Events .....	49
106	Table 8 – Ethernet Header Format .....	50
107	Table 9 – Control Packet Header Format .....	51
108	Table 10 – Generic Example of Control Packet Payload.....	52
109	Table 11 – Generic Example of Response Packet Payload Format .....	54
110	Table 12 – Reason Code Ranges.....	54
111	Table 13 – Standard Response Code Values.....	55
112	Table 14 – Standard Reason Code Values .....	55
113	Table 15 – AEN Packet Format .....	56
114	Table 16 – AEN Types .....	56
115	Table 17 – Command and Response Types .....	57
116	Table 18 – Example of Complete Minimum-Sized NC-SI Command Packet.....	58
117	Table 19 – Example of Complete Minimum-Sized NC-SI Response Packet .....	59
118	Table 20 – Clear Initial State Command Packet Format .....	60
119	Table 21 – Clear Initial State Response Packet Format.....	60
120	Table 22 – Select Package Command Packet Format.....	61

121 Table 23 – Hardware Arbitration Disable Byte..... 62

122 Table 24 – Select Package Response Packet Format ..... 62

123 Table 25 – Deselect Package Command Packet Format ..... 63

124 Table 26 – Deselect Package Response Packet Format ..... 63

125 Table 27 – Enable Channel Command Packet Format ..... 63

126 Table 28 – Enable Channel Response Packet Format..... 64

127 Table 29 – Disable Channel Command Packet Format ..... 64

128 Table 30 – Disable Channel Response Packet Format..... 65

129 Table 31 – Reset Channel Command Packet Format ..... 65

130 Table 32 – Reset Channel Response Packet Format ..... 65

131 Table 33 – Enable Channel Network TX Command Packet Format ..... 66

132 Table 34 – Enable Channel Network TX Response Packet Format..... 66

133 Table 35 – Disable Channel Network TX Command Packet Format ..... 67

134 Table 36 – Disable Channel Network TX Response Packet Format..... 67

135 Table 37 – AEN Enable Command Packet Format ..... 67

136 Table 38 – Format of AEN Control..... 68

137 Table 39 – AEN Enable Response Packet Format..... 68

138 Table 40 – Set Link Command Packet Format..... 69

139 Table 41 – Set Link Bit Definitions ..... 69

140 Table 42 – OEM Set Link Bit Definitions..... 70

141 Table 43 – Set Link Response Packet Format ..... 70

142 Table 44 – Set Link Command-Specific Reason Codes..... 70

143 Table 45 – Get Link Status Command Packet Format ..... 71

144 Table 46 – Get Link Status Response Packet Format..... 71

145 Table 47 – Link Status Field Bit Definitions ..... 71

146 Table 48 – Other Indications Field Bit Definitions ..... 74

147 Table 49 – OEM Link Status Field Bit Definitions (Optional) ..... 74

148 Table 50 – Get Link Status Command-Specific Reason Code..... 75

149 Table 51 – IEEE 802.1q VLAN Fields..... 75

150 Table 52 – Set VLAN Filter Command Packet Format ..... 76

151 Table 53 – Possible Settings for Filter Selector Field (8-Bit Field) ..... 76

152 Table 54 – Possible Settings for Enable (E) Field (1-Bit Field)..... 76

153 Table 55 – Set VLAN Filter Response Packet Format ..... 76

154 Table 56 – Set VLAN Filter Command-Specific Reason Code..... 77

155 Table 57 – Enable VLAN Command Packet Format ..... 77

156 Table 58 – VLAN Enable Modes..... 77

157 Table 59 – Enable VLAN Response Packet Format..... 78

158 Table 60 – Disable VLAN Command Packet Format ..... 78

159 Table 61 – Disable VLAN Response Packet Format..... 78

160 Table 62 – Set MAC Address Command Packet Format ..... 80

161 Table 63 – Possible Settings for MAC Address Number (8-Bit Field) ..... 80

162 Table 64 – Possible Settings for Address Type (3-Bit Field) ..... 80

163 Table 65 – Possible Settings for Enable Field (1-Bit Field) ..... 80

164 Table 66 – Set MAC Address Response Packet Format..... 81

165 Table 67 – Set MAC Address Command-Specific Reason Code..... 81

166 Table 68 – Enable Broadcast Filter Command Packet Format ..... 81

167 Table 69 – Broadcast Packet Filter Settings Field..... 82

168 Table 70 – Enable Broadcast Filter Response Packet Format..... 83

169	Table 71 – Disable Broadcast Filter Command Packet Format.....	84
170	Table 72 – Disable Broadcast Filter Response Packet Format.....	84
171	Table 73 – Enable Global Multicast Filter Command Packet Format.....	85
172	Table 74 – Bit Definitions for Multicast Packet Filter Settings Field.....	85
173	Table 75 – Enable Global Multicast Filter Response Packet Format .....	86
174	Table 76 – Disable Global Multicast Filter Command Packet Format .....	87
175	Table 77 – Disable Global Multicast Filter Response Packet Format .....	87
176	Table 78 – Set NC-SI Flow Control Command Packet Format .....	88
177	Table 79 – Values for the Flow Control Enable Field (8-Bit Field).....	88
178	Table 80 – Set NC-SI Flow Control Response Packet Format.....	88
179	Table 81 – Set NC-SI Flow Control Command-Specific Reason Code.....	89
180	Table 82 – Get Version ID Command Packet Format .....	89
181	Table 83 – Get Version ID Response Packet Format.....	89
182	Table 84 – Get Capabilities Command Packet Format .....	91
183	Table 85 – Get Capabilities Response Packet Format.....	92
184	Table 86 – Capabilities Flags Bit Definitions.....	92
185	Table 87 – VLAN Mode Support Bit Definitions.....	94
186	Table 88 – Get Parameters Command Packet Format .....	94
187	Table 89 – Get Parameters Response Packet Format.....	95
188	Table 90 – Get Parameters Data Definition .....	95
189	Table 91 – MAC Address Flags Bit Definitions.....	96
190	Table 92 – VLAN Tag Flags Bit Definitions.....	96
191	Table 93 – Configuration Flags Bit Definitions.....	97
192	Table 94 – Get Controller Packet Statistics Command Packet Format.....	97
193	Table 95 – Get Controller Packet Statistics Response Packet Format .....	98
194	Table 96 – Get Controller Packet Statistics Counter Numbers.....	99
195	Table 97 – Counters Cleared from Last Read Fields Format.....	102
196	Table 98 – Get NC-SI Statistics Command Packet Format.....	102
197	Table 99 – Get NC-SI Statistics Response Packet Format .....	102
198	Table 100 – Get NC-SI Statistics Response Counters .....	103
199	Table 101 – Get NC-SI Pass-through Statistics Command Packet Format .....	103
200	Table 102 – Get NC-SI Pass-through Statistics Response Packet Format.....	104
201	Table 103 – Get NC-SI Statistics Response.....	104
202	Table 104 – OEM Command Packet Format.....	105
203	Table 105 – OEM Response Packet Format .....	106
204	Table 106 – Link Status Change AEN Packet Format.....	106
205	Table 107 – Configuration Required AEN Packet Format .....	107
206	Table 108 – Host Network Controller Driver Status Change AEN Packet Format .....	107
207	Table 109 – Host Network Controller Driver Status Format .....	107
208	Table 110 – NC-SI Packet-Based and Op-Code Timing Parameters .....	108
209	Table 111 – Physical NC-SI Signals .....	111
210	Table 112 – DC Specifications.....	113
211	Table 113 – AC Specifications.....	114
212		

213

## Foreword

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

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

218

## Introduction

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

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

228 The specification defines the following primary characteristics of the interface:

- 229 • the physical and electrical characteristics of the interface
- 230 • the protocol and commands necessary for the operation of the interface

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



# 234 Network Controller Sideband Interface (NC-SI) Specification

## 235 1 Scope

236 This specification defines the functionality and behavior of the Sideband Interface responsible for  
237 connecting the Network Controller to the Management Controller. It also outlines the behavioral model of  
238 the network traffic destined for the Management Controller from the Network Controller.

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

- 241 • behavior of the interface, which include its operational states as well as the states of the  
242 associated components
- 243 • the electrical characteristics of the interface
- 244 • the payloads and commands of the communication protocol supported over the interface

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

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

## 249 2 Normative References

250 The following referenced documents are indispensable for the application of this document. For dated  
251 references, only the edition cited applies. For undated references, the latest edition of the referenced  
252 document (including any amendments) applies.

253 IEEE 802.3, *802.3™ IEEE Standard for Information technology— Part 3: Carrier sense multiple access*  
254 *with collision detection (CSMA/CD) access method and physical layer specifications*, December 2005,  
255 <http://www.ieee.org/portal/site>

256 IEEE 802.1Q, *IEEE 802.1Q-2005 IEEE Standard for Local and Metropolitan Area Networks—Virtual*  
257 *Bridged Local Area Networks*, <http://www.ieee.org/portal/site>. This standard defines the operation of  
258 Virtual LAN (VLAN) Bridges that permit the definition, operation and administration of Virtual LAN  
259 topologies within a Bridged LAN infrastructure.

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

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

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

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

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

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

271 Reduced Media Independent Interface (RMII) Consortium, *RMII Specification*, revision 1.2, March 20,  
272 1998, [http://www.national.com/appinfo/networks/files/rmii\\_1\\_2.pdf](http://www.national.com/appinfo/networks/files/rmii_1_2.pdf)

## 273 **3 Terms and Definitions**

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

### 275 **3.1 Requirement Term Definitions**

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

#### 277 **3.1**

##### 278 **conditional**

279 indicates that an item is required under specified conditions

#### 280 **3.2**

##### 281 **deprecated**

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

#### 283 **3.3**

##### 284 **mandatory**

285 indicates that an item is required under all conditions

#### 286 **3.4**

##### 287 **may**

288 indicates that an item is truly optional

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

#### 293 **3.5**

##### 294 **may not**

295 indicates flexibility of choice with no implied preference

#### 296 **3.6**

##### 297 **not recommended**

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

#### 301 **3.7**

##### 302 **obsolete**

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

#### 304 **3.8**

##### 305 **optional**

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

#### 307 **3.9**

##### 308 **recommended**

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

- 311 **3.10**  
312 **required**  
313 indicates that the item is an absolute requirement of the specification
- 314 **3.11**  
315 **shall**  
316 indicates that the item is an absolute requirement of the specification
- 317 **3.12**  
318 **shall not**  
319 indicates that the item is an absolute prohibition of the specification
- 320 **3.13**  
321 **should**  
322 indicates that valid reasons may exist in particular circumstances to ignore a particular item, but the full  
323 implications should be understood and carefully weighed before choosing a different course
- 324 **3.14**  
325 **should not**  
326 indicates that valid reasons may exist in particular circumstances when the particular behavior is  
327 acceptable or even useful, but the full implications should be understood and carefully weighed before  
328 implementing any behavior described with this label

## 329 **3.2 NC-SI Term Definitions**

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

### 331 **3.2.1**

#### 332 **Frame**

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

335 *Frame* is used in references to [IEEE 802.3 Frames](#). *Packet* is used in all other references.

### 336 **3.2.2**

#### 337 **Packet**

338 a formatted block of information carried by a computer network

339 *Frame* is used in references to [IEEE 802.3 Frames](#). *Packet* is used in all other references.

### 340 **3.2.3**

#### 341 **External Network Interface**

342 the interface of the Network Controller that provides connectivity to the external network infrastructure;  
343 also known as *port*

### 344 **3.2.4**

#### 345 **Internal Host Interface**

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

- 348 **3.2.5**  
349 **Management Controller**  
350 an intelligent entity composed of hardware/firmware/software that resides within a platform and is  
351 responsible for some or all of the management functions associated with the platform; also known as  
352 BMC and Service Processor
- 353 **3.2.6**  
354 **Network Controller**  
355 the component within a system that is responsible for providing connectivity to an external Ethernet  
356 network
- 357 **3.2.7**  
358 **Remote Media**  
359 a manageability feature that enables remote media devices to appear as if they are attached locally to the  
360 host
- 361 **3.2.8**  
362 **Network Controller Sideband Interface**  
363 **NC-SI**  
364 the interface of the Network Controller that provides network connectivity to a Management Controller;  
365 also shown as *Sideband Interface* or *NC-SI* as appropriate in the context
- 366 **3.2.9**  
367 **Integrated Controller**  
368 a Network Controller device that supports two or more channels for the NC-SI that share a common  
369 NC-SI physical interface (for example, a Network Controller that has two or more physical network ports  
370 and a single NC-SI bus connection)
- 371 **3.2.10**  
372 **Multi-drop**  
373 refers to the situation in which multiple physical communication devices share an electrically common bus  
374 and a single device acts as the master of the bus and communicates with multiple “slave” or “target”  
375 devices  
  
376 Related to NC-SI, a Management Controller serves the role of the master, and the Network Controllers  
377 are the target devices.
- 378 **3.2.11**  
379 **Point-to-Point**  
380 refers to the situation in which only a single Management Controller and single Network Controller  
381 package are used on the bus in a master/slave relationship, where the Management Controller is the  
382 master
- 383 **3.2.12**  
384 **Channel**  
385 the control logic and data paths that support NC-SI Pass-through operations through a single network  
386 interface (port)
- 387 A Network Controller that has multiple network interface ports can support an equivalent number of NC-SI  
388 channels.

389 **3.2.13**390 **Package**

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

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

396 **3.2.14**397 **Control traffic**398 **Control packets**

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

401 **3.2.15**402 **Command**

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

405 **3.2.16**406 **Response**

407 control packet sent by the Network Controller to the Management Controller as a positive  
408 acknowledgement of a command received from the Management Controller, and to provide the execution  
409 outcome of the command, as well as to return any required data

410 **3.2.17**411 **Asynchronous event notification**

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

414 **3.2.18**415 **Pass-through traffic**416 **Pass-through packets**

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

419 **3.3 Numbers and Number Bases**

420 Hexadecimal numbers are written with a "0x" prefix (for example, 0xFFFF and 0x80). Binary numbers are  
421 written with a lowercase *b* suffix (for example, 1001b and 10b). Hexadecimal and binary numbers are  
422 formatted in the *Courier New* font.

423 **3.4 Reserved Fields**

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

## 426 **4 Acronyms and Abbreviations**

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

428 **4.1**

429 **AC**

430 alternating current

431 **4.2**

432 **AEN**

433 Asynchronous Event Notification

434 **4.3**

435 **BMC**

436 Baseboard Management Controller (often used interchangeably with MC)

437 **4.4**

438 **CRC**

439 cyclic redundancy check

440 **4.5**

441 **CRS\_DV**

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

443 **4.6**

444 **DC**

445 direct current

446 **4.7**

447 **DHCP**

448 Dynamic Host Configuration Protocol

449 **4.8**

450 **FCS**

451 Frame Check Sequence

452 **4.9**

453 **MC**

454 Management Controller

455 **4.10**

456 **NC**

457 Network Controller

458 **4.11**

459 **NC-SI**

460 Network Controller Sideband Interface

461 **4.12**

462 **NC-SI RX**

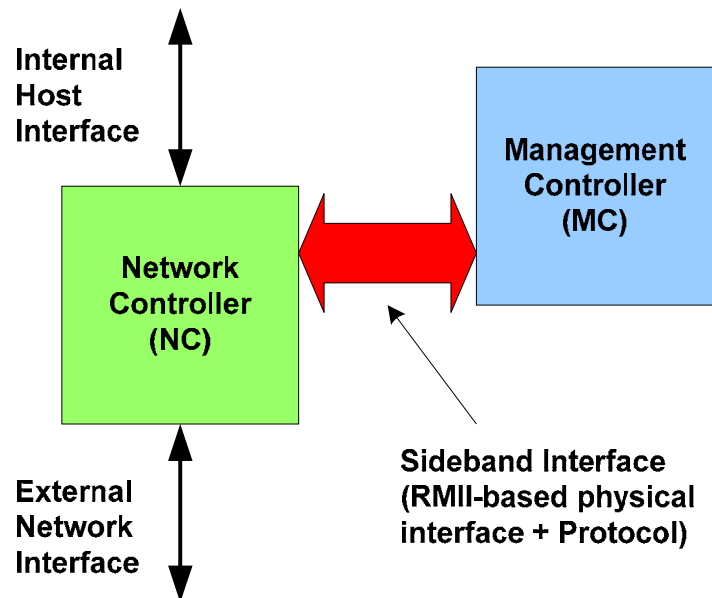
463 the direction of traffic on the NC-SI from the Network Controller to the Management Controller

- 464 **4.13**  
465 **NC-SI TX**  
466 the direction of traffic on the NC-SI to the Network Controller from the Management Controller
- 467 **4.14**  
468 **RMII**  
469 Reduced Media Independent Interface
- 470 **4.15**  
471 **RX**  
472 Receive
- 473 **4.16**  
474 **RXD**  
475 physical NC-SI signals used to transmit data from the Network Controller to the Management Controller
- 476 **4.17**  
477 **RX\_ER**  
478 a physical NC-SI signal used to indicate a Receive Error
- 479 **4.18**  
480 **SerDes**  
481 serializer/deserializer; an integrated circuit (IC or chip) transceiver that converts parallel data to serial data  
482 and vice-versa. This is used to support interfaces such as 1000Base-X and others.
- 483 **4.19**  
484 **TX**  
485 Transmit
- 486 **4.20**  
487 **TXD**  
488 physical NC-SI signals used to transmit data from the Management Controller to the Network Controller
- 489 **4.21**  
490 **VLAN**  
491 Virtual LAN

## 492 **5 NC-SI Overview**

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

499 The NC-SI is defined as the interface between a Management Controller and one or multiple Network  
500 Controllers. This interface, depicted in Figure 1, is responsible for providing external network connectivity  
501 for the Management Controller. Electrically, it is similar to the Reduced Media Independent Interface™  
502 (RMII) operating in full-duplex, as described in Clause 10.



503

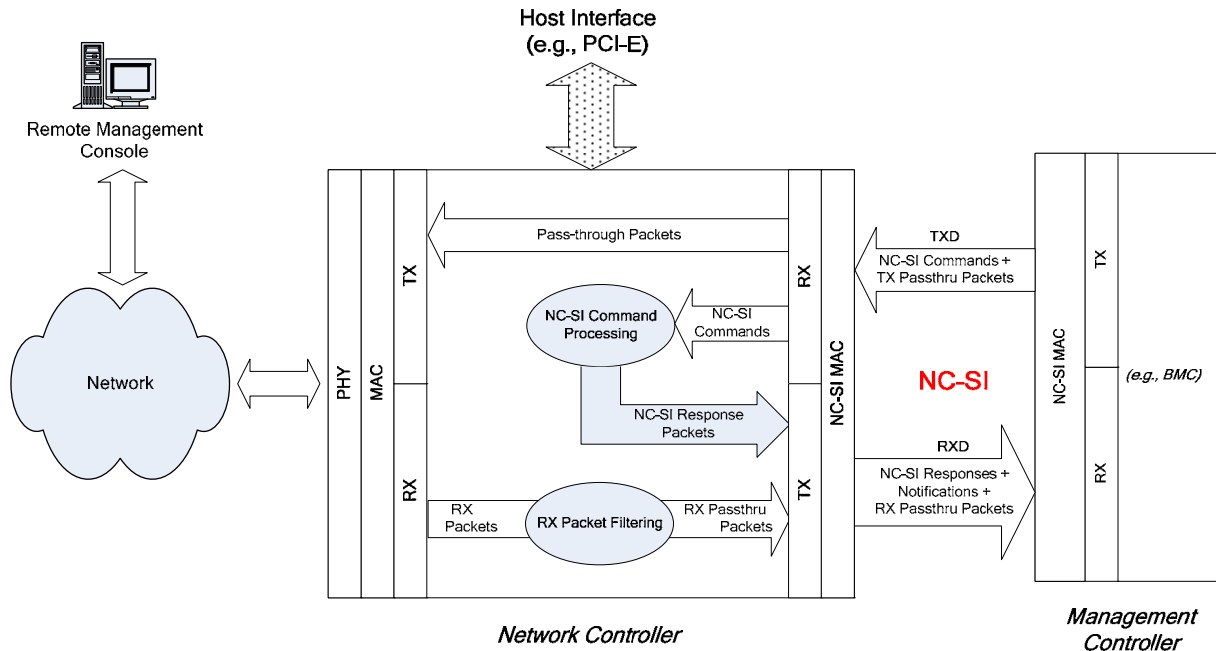
504

**Figure 1 – NC-SI Functional Block Diagram**

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

- 507
- 508
- “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
- 509
- 510





511

512

Figure 2 – NC-SI Traffic Flow Diagram

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

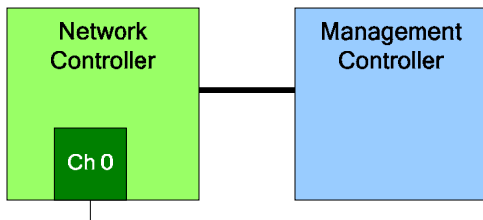
## 517 5.1 Defined Topologies

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

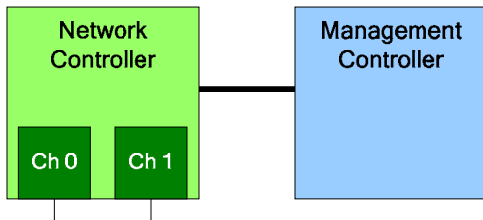
522 Figure 3 illustrates some examples of Network Controller configurations supported by the NC-SI in the  
 523 current release:

- 524 • Configuration 1 shows a Management Controller connecting to a single Network Controller with  
 525 a single external network connection.
- 526 • Configuration 2 shows a Management Controller connecting to a Network Controller package  
 527 that supports two NC-SI channels connections.
- 528 • Configuration 3 shows a Management Controller connecting to four discrete Network  
 529 Controllers.

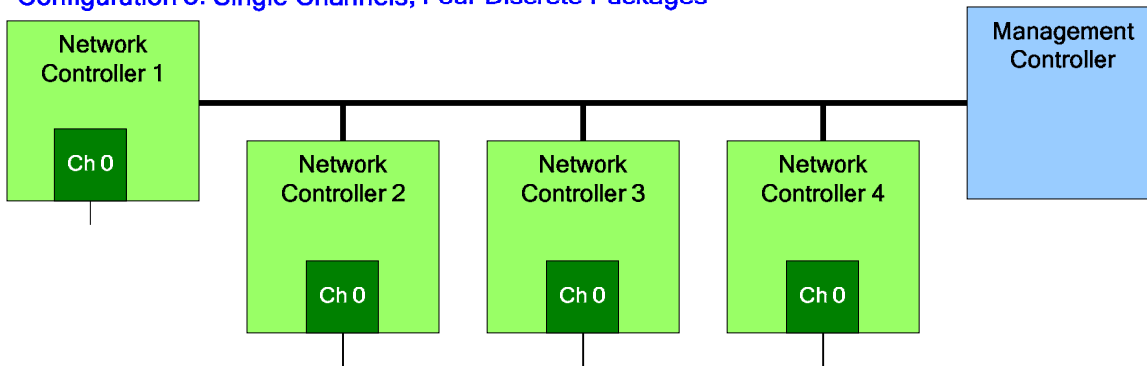
Configuration 1: Single Channel, Single Package



Configuration 2: Integrated Dual Channel, Single Package



Configuration 3: Single Channels, Four Discrete Packages



530

531

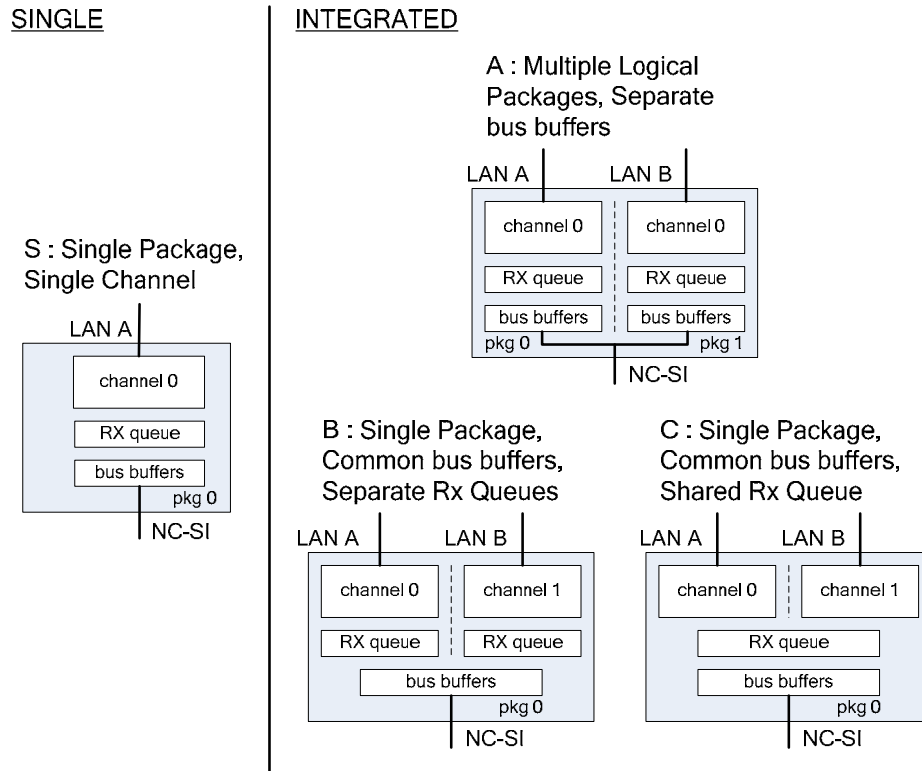
Figure 3 – Example Topologies Supported by the NC-SI

## 532 5.2 Single and Integrated Network Controller Implementations

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

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

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



543

544

**Figure 4 – Network Controller Integration Options**

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

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

553 • **S: Single Package, Single Channel**

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

557 • **A: Multiple Logical Packages, Separate Bus Buffers**

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

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

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

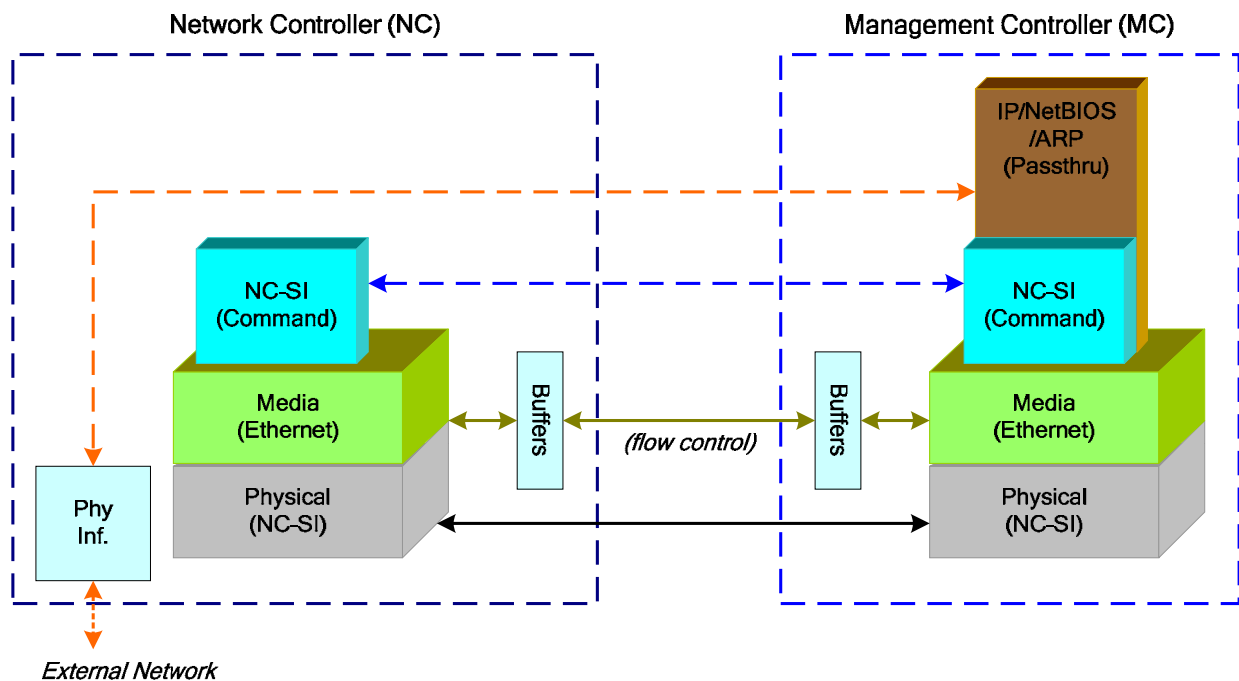
570 In this implementation, the two internal NC-SI channels share a common set of electrical bus  
 571 buffers. A single Deselect Package command will deselect the entire package. The Channel  
 572 Enable and Channel Disable commands to each channel control whether the channel can  
 573 transmit Pass-through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

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

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

583 **5.3 Transport Stack**

584 The overall transport stack of the NC-SI is illustrated in Figure 5. The lowest, physical-level interface is  
 585 based on RMI, and the media-level interface is based on Ethernet. Above these interfaces are the two  
 586 data-level protocols that are supported by the *NC-SI Specification*: NC-SI Command Protocol and the  
 587 Network Data Protocol (for example, ARP, IP, DHCP, and NetBIOS) associated with Pass-through traffic.



588 External Network

589 **Figure 5 – NC-SI Transport Stack**

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

## 595 **5.4 Transport Protocol**

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

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

## 604 **6 Operational Behaviors**

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

### 606 **6.1 Typical Operational Model**

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

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

- 609 • **Initial Configuration**

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

- 614 • **Pass-through**

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

- 618 • **Asynchronous Event Handling**

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

- 622 • **Error Handling**

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

629 **6.2 State Definitions**

630 This clause describes NC-SI operating states.

631 **6.2.1 General**

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

635 **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.

## 6.2.2 NC-SI Power States

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

- **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.

## 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 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. 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 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

NOTE: It shall not transmit AENs or Pass-through packets through the NC-SI interface.

- 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 and Set VLAN Filter commands) to "disabled"

- reset the counters defined in the Get NC-SI Statistics command and the Get NC-SI Pass-Through Statistics command to 0x0

- 677 • disable transmission of Pass-through packets onto the network
  - 678 NOTE: Upon entry into the Initial State, the Channel Network TX setting is also set to “disabled”.
  - 679 • reset the counters defined in the Get NC-SI Statistics command and the Get NC-SI Pass-  
680 Through Statistics command to 0x0 when entering into the Initial State
  - 681 • clear any record of prior command instances received upon entry into the Initial State (that is,  
682 assume that the first command received after entering the Initial State is a new command and  
683 not a retried command, regardless of any Instance ID that it may have received before entering  
684 the Initial State)
- 685 Otherwise, there is no requirement that other NC-SI configuration settings be set, retained, or restored to  
686 particular values in the Initial State.

### 687 **6.2.5 NC-SI Initial State Recovery**

688 As described in 6.2.4, a channel in the Initial State shall receive the Clear Initial State command before  
689 other commands can be executed. This requirement ensures that if the Initial State is entered  
690 asynchronously, the Management Controller is made aware that one or more NC-SI settings may have  
691 changed without its involvement, and blocks the Management Controller from issuing additional  
692 commands under that condition. Until the channel receives the Clear Initial State command, the  
693 Management Controller shall respond to any other received command (except the Select Package and  
694 Deselect Package commands) with a Command Failed response code and Interface Initialization  
695 Required reason code to indicate that the Clear Initial State command shall be sent. See response and  
696 reason code definitions in 8.2.5.

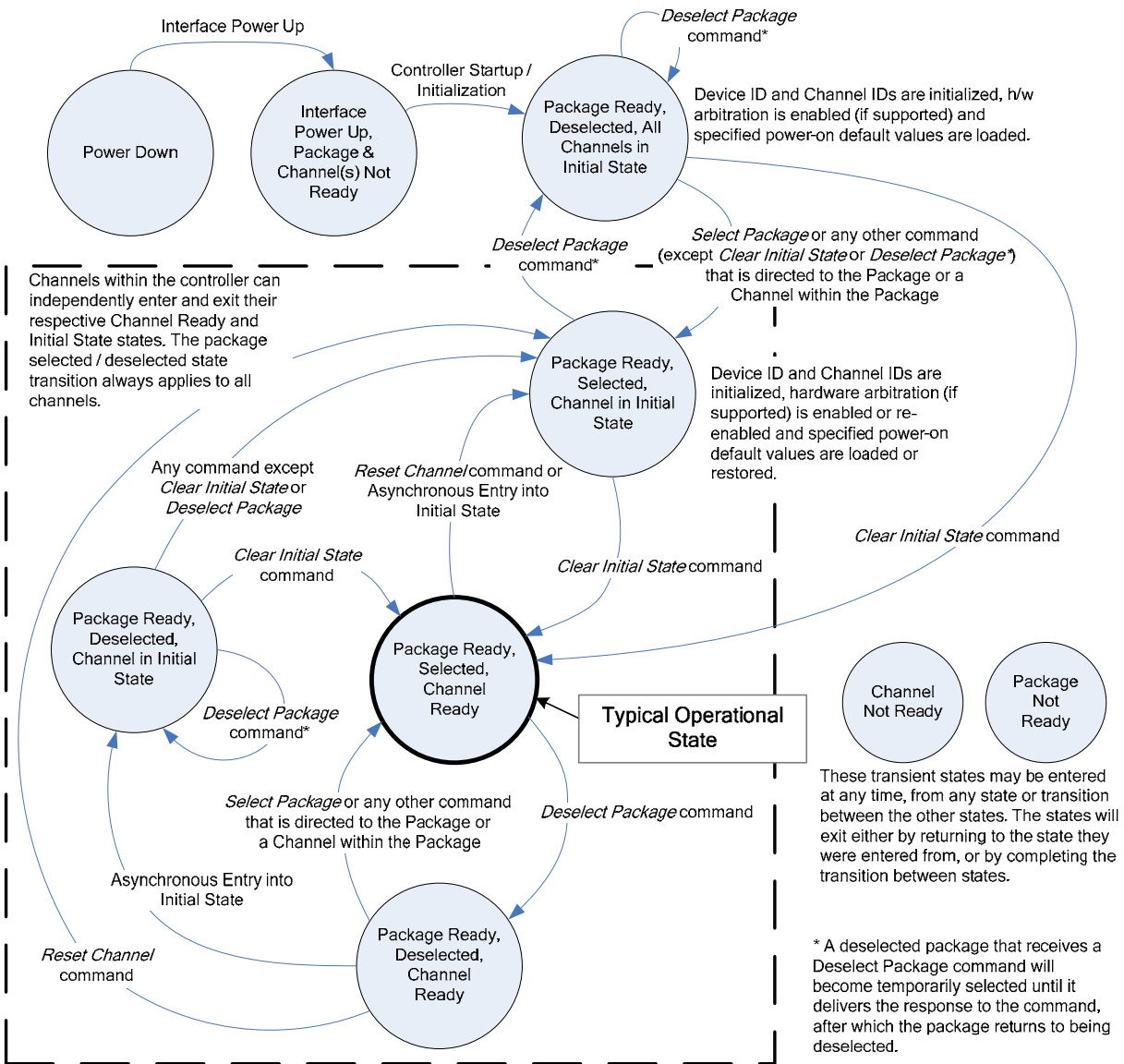
697 NOTE: Package commands (for example, Select Package and Deselect Package) are always accepted and  
698 responded to normally regardless of whether the Channel is in the Initial State.

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



703 **6.2.6 State Transition Diagram**

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



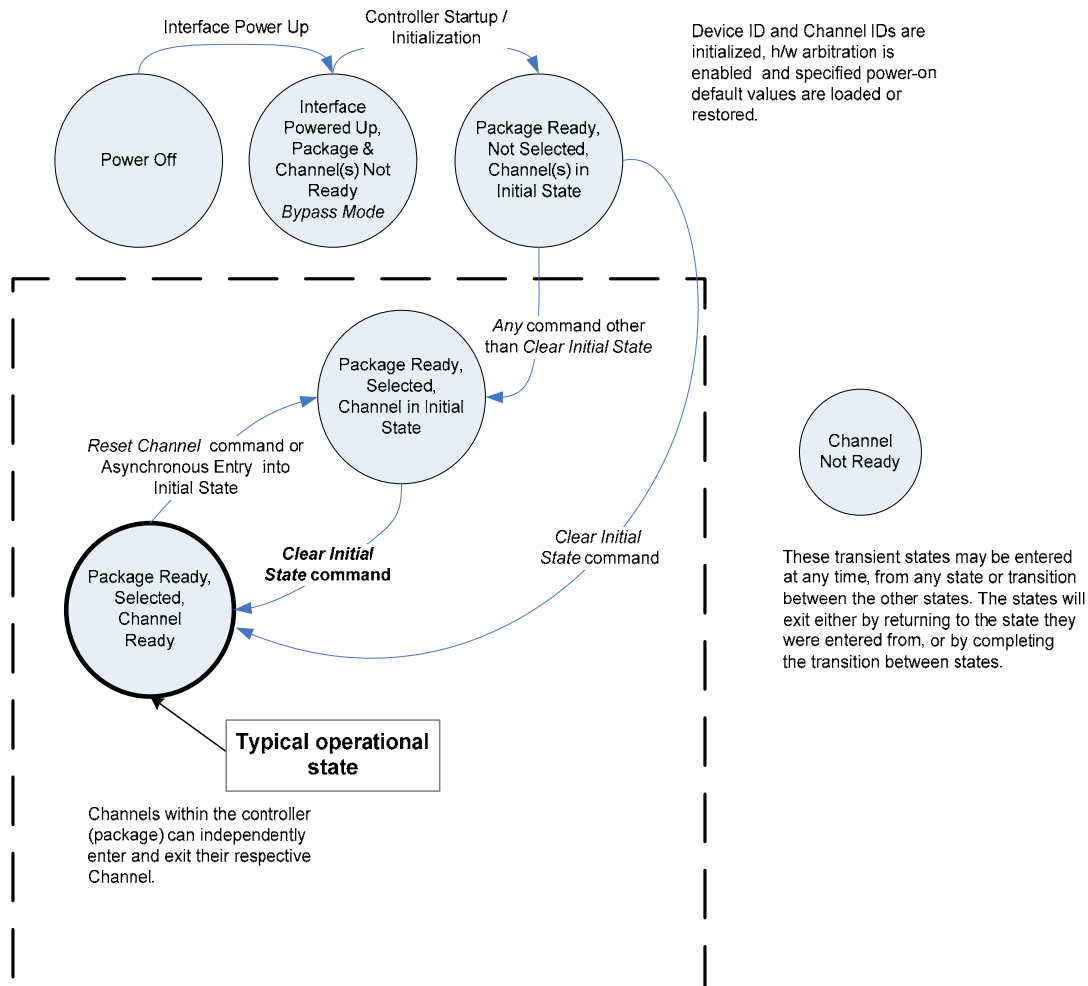
707

708

**Figure 6 – NC-SI Operational State Diagram**

709 **6.2.7 State Diagram for NC-SI Operation with Hardware Arbitration**

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



713  
 714 **Figure 7 – NC-SI Operational State Diagram for Hardware Arbitration Operation**

715 **6.2.8 Resets**

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

- 717 • Asynchronous Entry into Initial State
- 718 • Synchronous Reset

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

720 **6.2.8.1 Asynchronous Entry into Initial State**

721 An Asynchronous Reset event is defined as an event that results in a Channel asynchronously entering  
 722 the Initial State. This event could occur as a consequence of powering up, a System Reset, a Driver  
 723 Reset, an Internal Firmware error, loss of Configuration errors, Internal hardware errors, and so on.

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

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

730 **6.2.8.2 Synchronous Reset**

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

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

737 **6.2.9 Network Controller Channel ID**

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

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

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

747

**Table 2 – Channel ID Format**

Bits	Field Name	Description
[7..5]	Package ID	<p>The Package ID is required to be common across all channels within a single Network Controller that share a common NC-SI physical interconnect.</p> <p>The system integrator will typically configure the Package IDs starting from 0 and increasing sequentially for each physical Network Controller.</p> <p>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.</p>
[4..0]	Internal Channel ID	<p>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.</p> <p>An implementation is allowed to support additional configuration options for the Internal Channel ID as long as the required numbering can be configured.</p> <p>An Internal Channel ID value of 0x1F applies to the entire Package.</p>

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

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

## 757 6.2.10 Configuration-Related Settings

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

### 760 6.2.10.1 Package-Specific Operation

761 Only two configuration settings are package-specific:

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

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

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

### 771 6.2.10.2 Channel-Specific Operation

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

774 **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.

Setting/Configuration Category	Description
MAC Address Filter settings and control	The Set MAC Address command and Enable Global Multicast commands are used to configure the MAC Address Filter for unicast 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.

### 775 **6.2.11 Transmitting Pass-through Packets from the Management Controller**

776 Packets not recognized as command packets (that is, packets without the NC-SI Ethertype) that are  
 777 received on the Network Controller's NC-SI interface shall be assumed to be Pass-through packets  
 778 provided that they meet the source MAC Address setting for the channel in the Network Controller, and  
 779 will be forwarded for transmission to the corresponding external network interface if Channel Network TX  
 780 is enabled.

### 781 **6.2.12 Receiving Pass-through Packets for the Management Controller**

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

### 787 **6.2.13 Startup Sequence Examples**

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

#### 792 **6.2.13.1 Typical Non Hardware Arbitration Specific Startup Sequence**

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

#### 800 **1) Power Up**

801 The NC-SI is powered up (refer to 10.2.7 for the specification of this condition). The Network  
 802 Controller packages are provided a Device Ready Interval during which they can perform  
 803 internal firmware startup and initialization to prepare their NC-SI to accept commands. The  
 804 Management Controller first waits for the maximum Device Ready Interval to expire (refer to  
 805 Table 110). At this point, all the Network Controller packages and channels should be ready to  
 806 accept commands through the NC-SI. (The Management Controller may also start sending  
 807 commands before the Device Ready Interval expires, but will have to handle the case that  
 808 Network Controller devices may be in a state in which they are unable to accept or respond to  
 809 commands.)

**810 2) Discover Package**

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

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

**824 3) Discover and Get Capabilities for Each Channel in the Package**

825 The Management Controller can now discover how many channels are supported in the  
826 Network Controller package and their capabilities. To do this, the Management Controller issues  
827 the Clear Initial State command starting from the lowest Internal Channel ID (which selects a  
828 given channel within a package). If it receives a response, the Management Controller can then  
829 use the Get Version ID command to determine NC-SI specification compatibility, and the Get  
830 Capabilities command to collect information about the capabilities of the channel. The  
831 Management Controller can then repeat this step until the full number of internal channels has  
832 been discovered. (The Get Capabilities command includes a value that indicates the number of  
833 channels supported within the given package.)

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

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

**841 4) Repeat Steps 2 and 3 for Remaining Packages**

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

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

**849 5) Initialize Each Channel in the Package**

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

- 853 a) Issue the Select Package command.
- 854 b) For each channel in the package, depending on controller capabilities, perform the  
855 following actions. Refer to individual command descriptions for more information.

- 856           • Use the Set MAC Address command to configure which unicast and multicast  
857 addresses are used for routing Pass-through packets to and from the Management  
858 Controller.
  - 859           • Use the Enable Broadcast Filter command to configure whether incoming broadcast  
860 Pass-through packets are accepted or rejected.
  - 861           • Use the Enable Global Multicast Filter command to configure how incoming multicast  
862 Pass-through packets are handled based on settings from the Set MAC Address  
863 command.
  - 864           • Use the Set VLAN Filter and Enable VLAN Filters commands to configure how  
865 incoming Pass-through packets with VLAN Tags are handled.
  - 866           • Use the Set NC-SI Flow Control command to configure how Ethernet Pause Frames  
867 are used for flow control on the NC-SI.
  - 868           • Use the AEN Enable command to configure what types of AEN packets the channel  
869 should send out on the NC-SI.
  - 870           • Use the Enable Channel Network TX command to configure whether the channel is  
871 enabled to deliver Pass-through packets from the NC-SI to the network (based on the  
872 MAC address settings) or is disabled from delivering any Pass-through packets to the  
873 network.
- 874       c) Issue the Deselect Package command.

#### 875       6) **Enable Hardware Arbitration for the Packages**

876       If only a single Network Controller package is discovered, the Management Controller does not  
877 need to enable hardware arbitration if the controller hardware supports it. In fact, the  
878 Management Controller may always elect to disable hardware arbitration, because then it does  
879 not need to be concerned with whether the implementation provided a 'loop back' of the  
880 hardware arbitration 'ARB\_OUT' signal to the controller to the 'ARB\_IN' signal.

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

887       NOTE: There is no status to indicate whether hardware arbitration is hooked up and operating correctly.  
888 The Management Controller shall have prior knowledge that the implementation routes the hardware  
889 arbitration signals between the packages.

#### 890       7) **Start Pass-through Packet and AEN Operation on the Channels**

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

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

### 900 6.2.13.2 Hardware Arbitration Specific Startup Sequence

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

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

#### 913 1) **Power Up**

914 No change from other startup scenarios.

#### 915 2) **Discovery**

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

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

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

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

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



## 948 6.3 NC-SI Traffic Types

949 Two types of traffic are carried on the NC-SI: Pass-through traffic and Control traffic.

- 950 • Pass-through traffic consists of packets that are transferred between the external network  
951 interface and the Management Controller using the NC-SI.
- 952 • Control traffic consists of commands (requests) and responses that support the configuration  
953 and control of the NC-SI and Pass-through operation of the Network Controller, and AENs that  
954 support reporting various events to the Management Controller..

### 955 6.3.1 Command Protocol

956 Commands are provided to allow a Management Controller to initialize, control, and regulate  
957 Management Controller packet flow across the NC-SI, configure channel filtering, and to interrogate the  
958 operational status of the Network Controller. As interface master, the Management Controller is the  
959 initiator of all commands, and the Network Controller responds to commands.

#### 960 6.3.1.1 Instance IDs

961 The command protocol uses a packet field called the Instance ID (IID). IID numbers are 8-bit values that  
962 shall range from 0x01 to 0xFF. IIDs are used to identify new instances of a command, to improve the  
963 robustness of matching responses to commands, and to differentiate between new and retried  
964 commands.

965 The Network Controller that receives a command handles the IID in the following ways:

- 966 • It returns the IID value from the command in the corresponding response.
- 967 • If the IID is the same as the IID for the previous command, it recognizes the command as a  
968 'retried' command rather than as a new instance of the command.
- 969 • If a retried command is received, the Network Controller shall return the previous response.  
970 Depending on the command, the Network Controller can accomplish this either by holding the  
971 previous response data so that it can be returned, or, if re-executing the command has no side  
972 effects (that is, the command is idempotent), by re-executing the command operation and  
973 returning that response.
- 974 • When an IID value is received that is different from the one for the previous command, the  
975 Network Controller executes the command as a new command.
- 976 • When the Network Controller first enters the Initial State, it clears any record of any prior  
977 requests. That is, it assumes that the first command after entering the Initial State is a new  
978 command and not a retried command, regardless of any IID that it may have received before  
979 entering the Initial State.

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

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

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

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

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

### 999 6.3.1.2 Single-Threaded Operation

1000 The Network Controller is required to support NC-SI commands only in a single-threaded manner. That is,  
1001 the Network Controller is required to support processing only one command at a time, and is not required  
1002 to accept additional commands until after it has sent the response to the previous one.

1003 Therefore, the Management Controller should issue NC-SI commands in a single-threaded manner. That  
1004 is, the Management Controller should have only one command outstanding to a given Network Controller  
1005 package at a time. Upon sending an NC-SI command packet, and before sending a subsequent  
1006 command, the Management Controller should wait for the corresponding response packet to be received  
1007 or a command timeout event to occur before attempting to send another command. For the full  
1008 descriptions of command timeout, see 6.8.2.1.

### 1009 6.3.1.3 Responses

1010 The Network Controller shall process and acknowledge each validly formatted command received at the  
1011 NC-SI interface by formatting and sending a valid response packet to the Management Controller through  
1012 the NC-SI interface.

1013 To allow the Management Controller to match responses to commands, the Network Controller shall copy  
1014 the IID number of the Command into the Instance ID field of the corresponding response packet.

1015 To allow for retransmission and error recovery, the Network Controller may re-execute the last command  
1016 or maintain a copy of the response packet most recently transmitted to the Management Controller  
1017 through its NC-SI interface. This "previous" response packet shall be updated every time a new response  
1018 packet is transmitted to the Management Controller by replacing it with the one just sent.

1019 The Network Controller response shall return a "Command Unsupported" response code with an  
1020 "Unknown Command Type" reason code for any command (standard or OEM) that the Network Controller  
1021 does not support or recognize.

### 1022 6.3.1.4 Response and Post-Response Processing

1023 Typically, a Network Controller completes a requested operation before sending the response. In some  
1024 situations, however, it may be useful for the controller to be allowed to queue up the requested operation  
1025 and send the response assuming that the operation will complete correctly (for example, when the  
1026 controller is requested to change link configuration). The following provisions support this process:

- 1027 • A Network Controller is allowed to send a response before performing the requested action if  
1028 the command is expected to complete normally and all parameters that are required to be  
1029 returned with the response are provided.
- 1030 • Temporal ordering of requested operations shall be preserved. For example, if one command  
1031 updates a configuration parameter value and a following command reads back that parameter,  
1032 the operation requested first shall complete so that the following operation returns the updated  
1033 parameter.

- 1034       • Under typical operation of the Network Controller, responses should be delivered within the  
1035       Normal Execution Interval (T5) (see Table 110).
- 1036       • Unless otherwise specified, all requested operations shall complete within the Asynchronous  
1037       Reset/Asynchronous Not Ready interval (T6) following the response.
- 1038       • If the Network Controller channel determines that the requested operation or configuration  
1039       change has not been completed correctly after sending the response, the channel shall enter  
1040       the Initial State.

## 1041   **6.4 Link Configuration and Control**

1042   The Network Controller provides commands to allow the Management Controller to specify the auto-  
1043   negotiation, link speed, duplex settings, and so on to be used on the network interface. For more  
1044   information, see 8.4.21.

1045   NOTE:   The Management Controller should make link configuration changes only when the operating system (OS)  
1046   is absent.

### 1047   **6.4.1 Link Status**

1048   The Network Controller provides a Get Link Status command to allow the Management Controller to  
1049   interrogate the configuration and operational status of the primary Ethernet links. The Management  
1050   Controller may issue the Get Link Status command regardless of OS operational status.

## 1051   **6.5 Frame Filtering for Pass-through Mode**

1052   The Network Controller provides the option of configuring various types of filtering mechanisms for the  
1053   purpose of controlling the delivery of received Ethernet frames to the Management Controller. These  
1054   options include VLAN Tag filter, L2 address filters, MAC address support, and limited frame filtering using  
1055   L3, L4 protocol header fields. All frames that pass frame filtering are forwarded to the Management  
1056   Controller over the NC-SI.

### 1057   **6.5.1 Multicast Filtering**

1058   The Network Controller may provide commands to allow the Management Controller to enable and  
1059   disable global filtering of all multicast packets. The Network Controller may optionally provide one or more  
1060   individual multicast filters, as well as DHCP v6, IPv6 Neighbor Advertisement, and IPv6 Router  
1061   Advertisement filters.

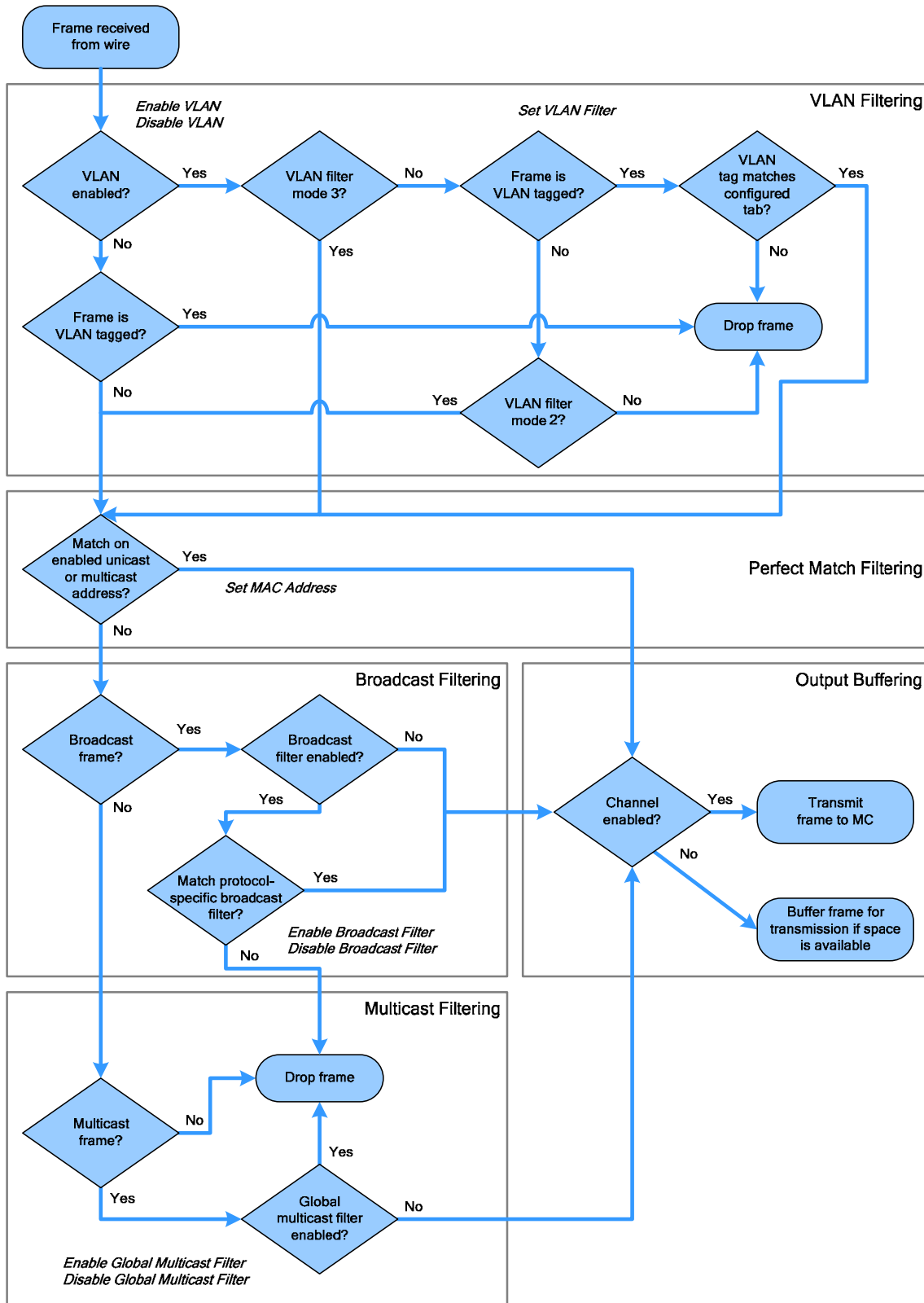
### 1062   **6.5.2 Broadcast Filtering**

1063   The Network Controller provides commands to allow the Management Controller to enable and disable  
1064   forwarding of Broadcast and ARP packets. The Network Controller may optionally support selective  
1065   forwarding of broadcast packets for specific protocols, such as DHCP and NetBIOS.

### 1066   **6.5.3 VLAN Filtering**

1067   The Network Controller provides commands to allow the Management Controller to enable and disable  
1068   VLAN filtering, configure one or more VLAN Filters, and to configure VLAN filtering modes.

1069   Figure 8 illustrates the flow of frame filtering. Italicized text in the figure is used to identify NC-SI  
1070   command names.



1071

1072

Figure 8 – NC-SI Packet Filtering Flowchart

## 1073 **6.6 NC-SI Flow Control**

1074 The Network Controller may provide commands to enable flow control on the NC-SI between the Network  
1075 Controller and the Management Controller. The NC-SI flow control behavior follows the PAUSE frame  
1076 behavior as defined in the [IEEE 802.3 specification](#). Flow control is configured using the Set NC-SI Flow  
1077 Command (see 8.4.41).

## 1078 **6.7 Asynchronous Event Notification**

1079 Asynchronous Event Notification (AEN) packets enable the Network Controller to deliver unsolicited  
1080 notifications to the Management Controller when certain status changes that could impact interface  
1081 operation occur in the Network Controller. Because the NC-SI is a small part of the larger Network  
1082 Controller, its operation can be affected by a variety of events that occur in the Network Controller. These  
1083 events include link status changes, OS driver loads and unloads, and chip resets. This feature defines a  
1084 set of notification packets that operate outside of the established command-response mechanism.

1085 Control over the generation of the AEN packets is achieved by control bits in the AEN Enable command.  
1086 Each type of notification is optional and can be independently enabled by the Management Controller.

1087 AENs are not acknowledged, and there is no protection against the possible loss of an AEN packet.

1088 Each defined event has its own AEN packet. Because the AEN packets are generated asynchronously by  
1089 the Network Controller, they cannot implement some of the features of the other Control packets. AEN  
1090 packets leverage the general packet format of Control packets.

- 1091 • The originating Network Controller channel shall fill in its Channel ID (Ch. ID) field in the  
1092 command header to identify the source of notification.
- 1093 • The IID field in an AEN shall be set to 0x00 to differentiate it from a response or command  
1094 packet.
- 1095 • The Network Controller shall copy the AEN MC ID field from the AEN Enable command into the  
1096 MC ID field in every AEN sent to the Management Controller.

## 1097 **6.8 Error Handling**

1098 This clause describes the error-handling methods that are supported over the NC-SI. Two types of error-  
1099 handling methods are defined:

- 1100 • Synchronous Error Handling
- 1101 • Errors that trigger Asynchronous Entry into the Initial State

1102 Synchronous Error Handling occurs when an Error (non-zero) Response/Reason Code is received in  
1103 response to a command issued by the Management Controller. For information about response and  
1104 reason codes, see 8.2.5.

1105 Asynchronous Entry into the Initial State Error Handling occurs when the Network Controller  
1106 asynchronously enters the Initial State because of an error condition that affects NC-SI configuration or a  
1107 failure of a command that was already responded to. For more information, see 6.2.8.1.

### 1108 **6.8.1 Transport Errors**

1109 Transport error handling includes the dropping of command packets. Data packet errors are out of the  
1110 scope of this specification.

### 1111 6.8.1.1 Dropped Control Packets

1112 The Network Controller shall drop command packets received on the NC-SI interface only under the  
1113 following conditions:

- 1114 • The packet has an invalid Frame Check Sequence (FCS) value.
- 1115 • Frame length does not meet [IEEE 802.3](#) requirements (except for OEM commands, where  
1116 accepting larger packets may be allowed as a vendor-specific option).
- 1117 • The packet checksum (if provided) is invalid.
- 1118 • The NC-SI Channel ID value in the packet does not match the expected value.
- 1119 • The Network Controller is congested and cannot accept the packet.
- 1120 • The Network Controller receives a command packet with an incorrect header revision.

1121 The Network Controller may also drop command packets if an event that triggers Asynchronous Entry into  
1122 the Initial State causes packets to be dropped during the transition.

### 1123 6.8.2 Missing Responses

1124 There are two typical scenarios in which the Management Controller may not receive the response to a  
1125 command:

- 1126 • The Network Controller dropped the command and thus never sent the response.
- 1127 • The response was dropped by the Management Controller (for example, because of a CRC  
1128 error in the response packet).

1129 The Management Controller can detect a missing response packet as the occurrence of an NC-SI  
1130 command timeout event.

#### 1131 6.8.2.1 Command Timeout

1132 The Management Controller can detect missing responses by implementing a command timeout interval.  
1133 The timeout value chosen by the Management Controller shall not be less than Normal Execution  
1134 Interval, T5. Upon detecting a timeout condition, the Management Controller should not make  
1135 assumptions on the state of the unacknowledged command (for example, the command was dropped or  
1136 the response was dropped), but should retransmit (retry) the previous command using the same IID it  
1137 used in the initial command.

1138 The Management Controller should try a command at least three times before assuming an error  
1139 condition in the Network Controller.

1140 It is possible that a Network Controller could send a response to the original command at the same time a  
1141 retried command is being delivered. Under this condition, the Management Controller could get more than  
1142 one response to the same command. Thus, the Management Controller should be capable of determining  
1143 that it has received a second instance of a previous response packet. Dropped commands may be  
1144 detected by the Management Controller as a timeout event waiting for the response.

#### 1145 6.8.2.2 Handling Dropped Commands or Missing Responses

1146 To recover from dropped commands or missing responses, the Management Controller can retransmit  
1147 the unacknowledged command packet using the same IID that it used for the initial command.

1148 The Network Controller shall be capable of reprocessing retransmitted (retried) commands without error  
1149 or undesirable side effects. The Network Controller can determine that the command has been  
1150 retransmitted by verifying that the IID is unchanged from the previous command.

### 1151 6.8.3 Detecting Pass-through Traffic Interruption

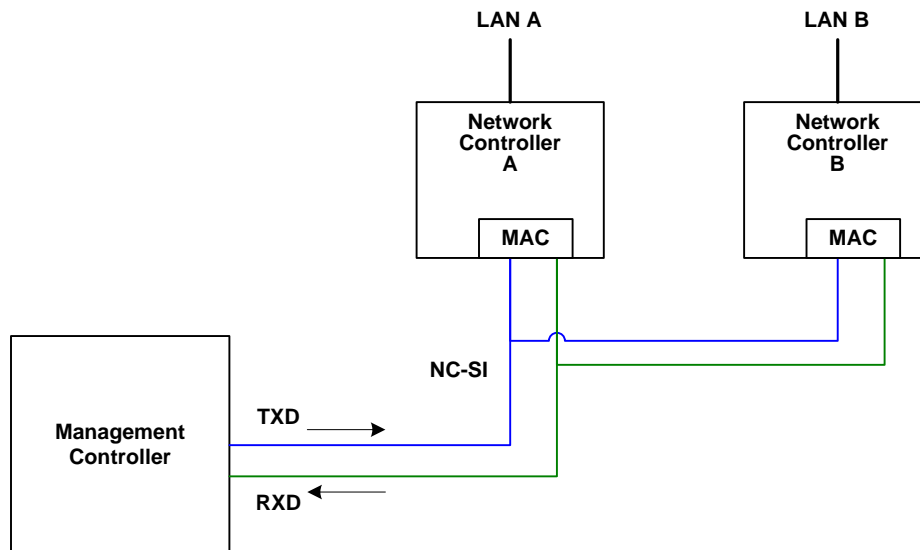
1152 The Network Controller might asynchronously enter the Initial State because of a reset or other event. In  
 1153 this case, the Network Controller stops transmitting Pass-through traffic on the RXD lines. Similarly, Pass-  
 1154 through traffic sent to the Network Controller may be dropped. If the Management Controller is not in the  
 1155 state of sending or receiving Pass-through traffic, it may not notice this condition. Thus the Management  
 1156 Controller should periodically issue a command to the Network Controller to test whether the Network  
 1157 Controller has entered the Initial State. How often this testing should be done is a choice of the  
 1158 Management Controller.

## 1159 7 Arbitration in Configurations with Multiple Network Controller 1160 Packages

1161 More than one Network Controller package on an NC-SI can be enabled for transmitting packets to the  
 1162 Management Controller. This specification defines two mechanisms to accomplish Network Controller  
 1163 package arbitration operations. One mechanism uses software commands provided by the Network  
 1164 Controller for the Management Controller to control whose turn it is to transmit traffic. The other  
 1165 mechanism uses hardware arbitration to share the single NC-SI bus. Implementations are required to  
 1166 support command-based Device Selection operation; the hardware arbitration method is optional.

### 1167 7.1 General

1168 Figure 9 is a simplified block diagram of the Sideband Interface being used in a multi-drop configuration.  
 1169 The RMII (upon which NC-SI is based) was originally designed for use as a point-to-point interconnect.  
 1170 Accordingly, only one party can transmit data onto the bus at any given time. There is no arbitration  
 1171 protocol intrinsic in the RMII to support managing multiple transmitters.



1172

1173

Figure 9 – Basic Multi-Drop Block Diagram

1174 However, it is possible for multiple Network Controllers on the interface to be able to simultaneously  
1175 *receive* traffic from the Management Controller that is being transmitted on the NC-SI TXD lines. The  
1176 Network Controllers can receive commands from the Management Controller without having to arbitrate  
1177 for the bus. This facilitates the Management Controller in delivering commands for setup and  
1178 configuration of arbitration.

1179 Arbitration allows multiple Network Controller packages that are attached to the interface to be enabled to  
1180 share the RXD lines to deliver packets to the Management Controller.

1181 This operation is summarized as follows:

- 1182 • Only one Network Controller at a time can transmit packets on the RXD lines of the interface.
- 1183 • Network Controllers can accept commands for configuring and controlling arbitration for the  
1184 RXD lines.

## 1185 7.2 Hardware Arbitration

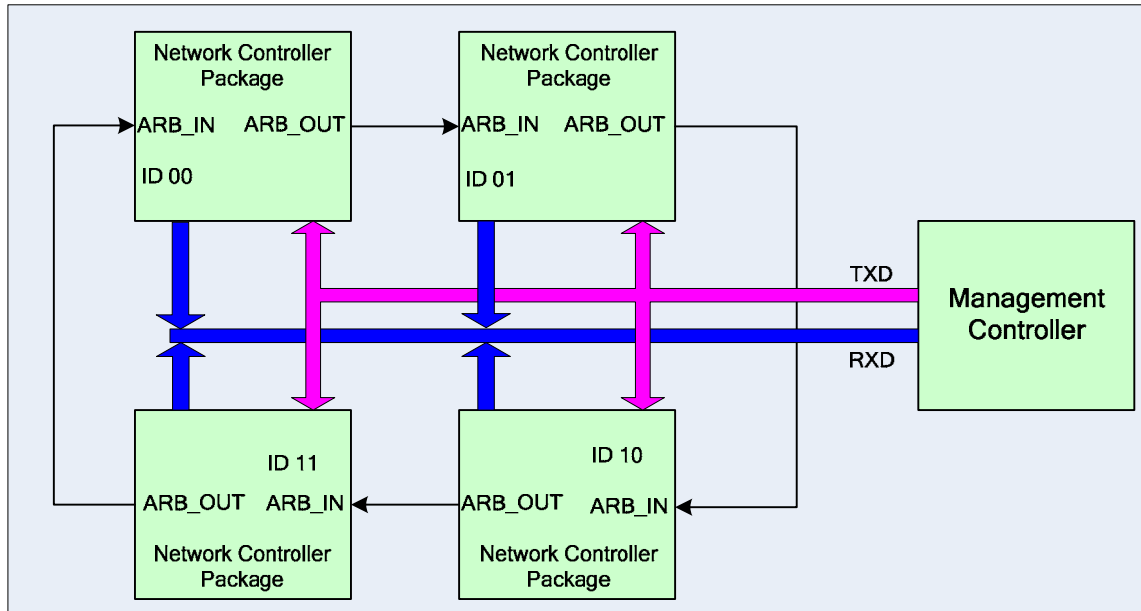
1186 To prevent two or more NC-SI packages from transmitting at the same time, a hardware-based arbitration  
1187 scheme was devised to allow only one Network Controller package to drive the RX lines of the shared  
1188 interface at any given time. This scheme uses a mechanism of passing messages (op-codes) between  
1189 Network Controller packages to coordinate when a controller is allowed to transmit through the NC-SI  
1190 interface.

### 1191 7.2.1 General

1192 Three conceptual modes of hardware arbitration exist: arbitration master assignment, normal operation,  
1193 and bypass. After a package is initialized and has its Channel IDs assigned, it enters the arbitration  
1194 master assignment mode. This mode assigns one package the role of an Arbitration Master  
1195 (ARB\_Master) that is responsible for initially generating a TOKEN op-code that is required for the normal  
1196 operating mode. In the normal operating mode, the TOKEN op-code is passed from one package to the  
1197 next in the ring. The package is allowed to use the shared RXD signals and transmit if the package has  
1198 received the TOKEN op-code and has a packet to send. Bypass mode allows hardware arbitration op-  
1199 codes to pass through a Network Controller package before it is initialized.

1200 Hardware-based arbitration requires two additional pins (ARB\_IN and ARB\_OUT) on the Network  
1201 Controller. The ARB\_OUT pin of one package is connected to the ARB\_IN pin of the next package to  
1202 form a ring configuration, as illustrated in Figure 10. The timing requirements for hardware arbitration are  
1203 designed to accommodate a maximum of four Network Controller packages. If the implementation  
1204 consists of a single Network Controller package, the ARB\_OUT pin may be connected to the ARB\_IN pin  
1205 on the same package, or may be left disconnected, in which case hardware arbitration should be disabled  
1206 by using the Select Package command.





1207

1208

**Figure 10 – Multiple Network Controllers in a Ring Format**

1209

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.

1210

1211

1212

1213

1214

**Table 4 – Hardware Arbitration Di-bit Encoding**

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

1215

**7.2.2 Hardware Arbitration Op-Codes**

1216

The hardware-based arbitration feature has five defined op-codes: IDLE, TOKEN, FLUSH, XON, and XOFF. Each op-code starts with an Esync symbol and is followed by either E<sub>one</sub> or E<sub>zero</sub> symbols. The legal op-codes are listed in Table 5.

1217

1218

1219

**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	$E_{sync} E_{one} E_{one} E_{zero} E(\text{Package\_ID}[2:0]) E_{zero}$ (11010100xxxxx00b)
XOFF	$E_{sync} E_{zero} E_{one} E_{zero} E_{zero} E_{zero}$ (110001000000b)
XON	$E_{sync} E_{zero} E_{one} E_{one} E_{zero} E(\text{Package\_ID}[2:0]) E_{zero}$ (1100010100uuuuuu00b)

1220 **7.2.2.1 Detecting Truncated Op-Codes**

1221 A truncated op-code is detected when the number of clocks between  $E_{sync}$ s is less than the number of bits  
 1222 required for the op-code. Note that any additional bits clocked in after a legitimate op-code is detected do  
 1223 not indicate an error condition and are ignored until the next  $E_{sync}$ .

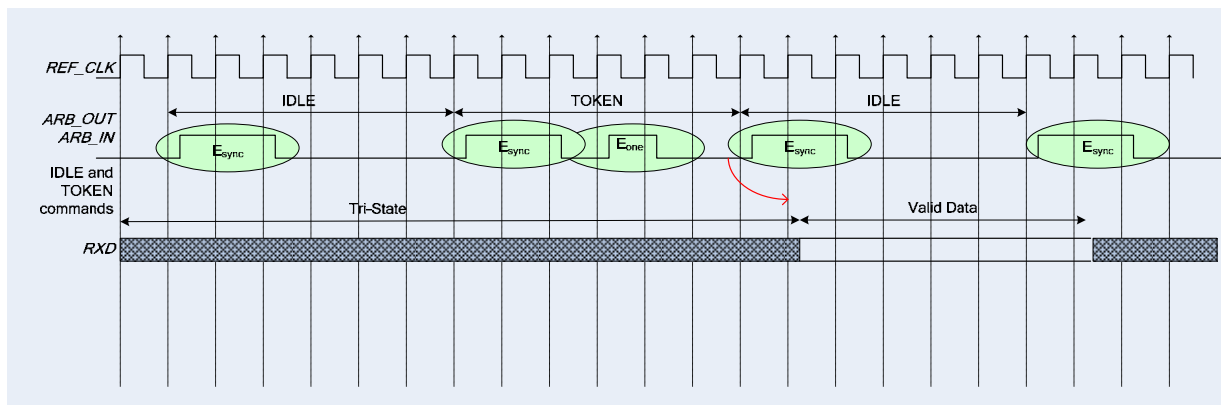
1224 **7.2.2.2 Handling Truncated or Illegal Op-Codes**

1225 When a Network Controller receives a truncated or illegal op-code, it should discard it.

1226 **7.2.2.3 Relationship of Op-Codes Processing and Driving the RX Data Lines**

1227 A Network Controller package shall take no more than T9 REF\_CLK times after receiving the last bit of  
 1228 the op-code to decode the incoming op-code and start generating the outgoing op-code. This time limit  
 1229 allows for decoding and processing of the incoming op-code under the condition that an outgoing op-code  
 1230 transmission is already in progress.

1231 A package that has received a TOKEN and has packet data to transmit shall turn on its buffer and begin  
 1232 transmitting the packet data within T11 REF\_CLK times of receiving the TOKEN, as illustrated in  
 1233 Figure 11. The package shall disable the RXD buffers before the last clock of the transmitted TOKEN.



1234

1235

**Figure 11 – Op-Code to RXD Relationship**

### 1236 7.2.3 Op-Code Operations

1237 This clause describes the behavior associated with the five defined op-codes.

#### 1238 7.2.3.1 TOKEN Op-Code

1239 When a TOKEN op-code is received, the Network Controller package may drive the RXD signals to send  
1240 only one of the following items: a Pass-through packet, a command response, or an AEN. One [IEEE](#)  
1241 [802.3](#) PAUSE frame (XON or XOFF) may also be sent either before or after one of the previous packets,  
1242 or on its own. While the Network Controller package is transmitting the data on the RXD signals of the  
1243 interface, it shall generate IDLE op-codes on its ARB\_OUT pin. Once a package completes its  
1244 transmission, if any, it shall generate and send the TOKEN on its ARB\_OUT pin.

#### 1245 7.2.3.2 IDLE Op-Code

1246 A package that has no other op-code to send shall continuously generate IDLE op-codes. Typically, a  
1247 received IDLE op-code indicates that the TOKEN is currently at another package in the ring. This op-code  
1248 is also used in the ARB\_Master assignment process (for details, see 7.2.5).

#### 1249 7.2.3.3 FLUSH Op-Code

1250 A FLUSH op-code is used to establish an Arbitration Master for the ring when the package enters the  
1251 Package Ready state or when the TOKEN is not received within the specified timeout, T8. This op-code  
1252 is further explained in 7.2.5.

1253 If the package receives a FLUSH op-code while it is in the middle of transmitting a packet onto NC-SI, it  
1254 shall generate IDLE op-codes until the transmission is complete and then process the FLUSH op-code as  
1255 described.

#### 1256 7.2.3.4 Flow Control Op-Codes

1257 The XON and XOFF op-codes are used to manage the generation of [IEEE 802.3](#) PAUSE frames on the  
1258 NC-SI. If the Network Controller supports flow control and flow control is enabled, the XOFF and XON  
1259 op-codes behave as described in this clause. If the Network Controller does not support flow control or if  
1260 flow control is not enabled, the Network Controller shall pass the op-codes to the next package.

1261 Note: There is a maximum amount of time that the Network Controller may maintain a PAUSE. For more  
1262 information, see 8.4.41.

##### 1263 7.2.3.4.1 XOFF Op-Code

1264 A Network Controller package that becomes congested while receiving packets from the NC-SI shall  
1265 perform the following actions:

- 1266 • If it does not have a TOKEN, it sends the XOFF op-code to the next package.
- 1267 • If it has the TOKEN and has not previously sent an XOFF frame for this instance of congestion,  
1268 it shall send a single XOFF frame (PAUSE frame with a pause time of 0xFFFF) and will not  
1269 generate an XOFF op-code.
- 1270 • A package may also regenerate an XOFF frame or op-code if it is still congested and  
1271 determines that the present PAUSE frame is about to expire.

1272 When a package on the ring receives an XOFF op-code, it shall perform one of the following actions:

- 1273 • If it does not have a TOKEN op-code, it passes the XOFF op-code to the next package in the  
1274 ring.
- 1275 • If it has the TOKEN, it shall send an XOFF frame (PAUSE frame with a pause time of 0xFFFF)  
1276 and will not regenerate the XOFF op-code. If it receives another XOFF op-code while sending  
1277 the XOFF frame or a regular network packet, it discards the received XOFF op-code.

#### 1278 7.2.3.4.2 XON Op-Code

1279 XON frames (PAUSE frame with a pause time of 0x0000) are used to signal to the Management  
1280 Controller that the Network Controller packages are no longer congested and that normal traffic flow can  
1281 resume. XON op-codes are used between the packages to coordinate XON frame generation. The  
1282 package ID is included in this op-code to provide a mechanism to verify that every package is not  
1283 congested before sending an XON frame to the Management Controller.

1284 The XON op-code behaves as follows:

- 1285 • When a package is no longer congested, it generates an XON op-code with its own Package  
1286 ID. This puts the package into the 'waiting for its own XON' state.
- 1287 • A package that receives the XON op-code takes one of the following actions:
  - 1288 – If it is congested, it replaces the received XON op-code with the IDLE op-code. This action  
1289 causes the XON op-code to be discarded. Eventually, the congested package generates  
1290 its own XON op-code when it exits the congested state.
  - 1291 – If the package is not congested and is not waiting for the XON op-code with own Package  
1292 ID, it forwards the received XON op-code to the next package in the ring.
  - 1293 NOTE: If the received XON op-code contains the package's own Package ID, the op-code should  
1294 be discarded.
  - 1295 – If the package is not congested and is waiting for its own XON op-code, it performs one of  
1296 the following actions:
    - 1297 • If it receives an XON op-code with a Package ID that is higher than its own, it replaces  
1298 the XON op-code with its own Package ID.
    - 1299 • If it receives an XON op-code with a Package ID lower than its own, it passes that  
1300 XON op-code to the next package and it exits the 'waiting for its own XON' state.
    - 1301 • If it receives an XON op-code with the Package ID equal to its own, it sends an XON  
1302 frame on the NC-SI when it receives the TOKEN op-code and exits the 'waiting for its  
1303 own XON' state.
  - 1304 NOTE: More than one XON op-code with the same Package ID may be received while waiting for  
1305 the TOKEN and while sending the XON frame. These additional XON op-codes should be discarded.
- 1306 • If a package originates an XON op-code but receives an XOFF op-code, it terminates its XON  
1307 request so that it does not output an XON frame when it receives the TOKEN.
- 1308 NOTE: This behavior should not occur because the Management Controller will be in the Pause state  
1309 at this point.
- 1310 • A package that generated an XON op-code may receive its own XON op-code back while it has  
1311 the TOKEN op-code. In this case, it may send a regular packet (Pass-through, command  
1312 response, or AEN) to the Management Controller (if it has one to send), an XON frame, or both.

## 1313 7.2.4 Bypass Mode

1314 When the Network Controller package is in bypass mode, data received on the ARB\_IN pin is redirected  
1315 to the ARB\_OUT pin within the specified clock delay. This way, arbitration can continue between other  
1316 devices in the ring.

1317 A package in bypass mode shall take no more than T10 REF\_CLK times to forward data from the  
1318 ARB\_IN pin to the ARB\_OUT pin. The transition in and out of bypass mode may result in a truncated  
1319 op-code.

1320 A Network Controller package enters into bypass mode immediately upon power up and transitions out of  
1321 this mode after the Network Controller completes its startup/initialization sequence.

## 1322 7.2.5 Hardware Arbitration Startup

1323 Hardware arbitration startup works as follows:

- 1324 1) All the packages shall be in bypass mode within  $T_{pwrz}$  seconds of NC-SI power up.
- 1325 2) As each package is initialized, it shall continuously generate FLUSH op-codes with its own  
1326 Package ID.
- 1327 3) The package then participates in the ARB\_MSTR assignment process described in the  
1328 following clause.

## 1329 7.2.6 ARB\_MSTR Assignment

1330 ARB\_MSTR assignment works as follows:

- 1331 1) When a package receives a FLUSH op-code with a Package ID numerically smaller than its  
1332 own, it shall forward on the received FLUSH op-code. If the received FLUSH op-code's  
1333 Package ID is numerically larger than the local Package ID, the package shall continue to send  
1334 its FLUSH op-code with its own Package ID. When a package receives a FLUSH op-code with  
1335 its own Package ID, it becomes the master of the ring (ARB\_MSTR).
- 1336 2) The ARB\_MSTR shall then send out IDLE op-codes until it receives an IDLE op-code.
- 1337 3) Upon receiving the IDLE op-code, the ARB\_MSTR shall be considered to be in possession of  
1338 the TOKEN op-code (see 7.2.3.1).

1339 NOTE: If the package receives a FLUSH op-code while it is in the middle of transmitting a packet onto  
1340 NC-SI, it shall generate IDLE op-codes until the transmission is complete and then process the FLUSH op-  
1341 code as described.

## 1342 7.2.7 Token Timeout Mechanism

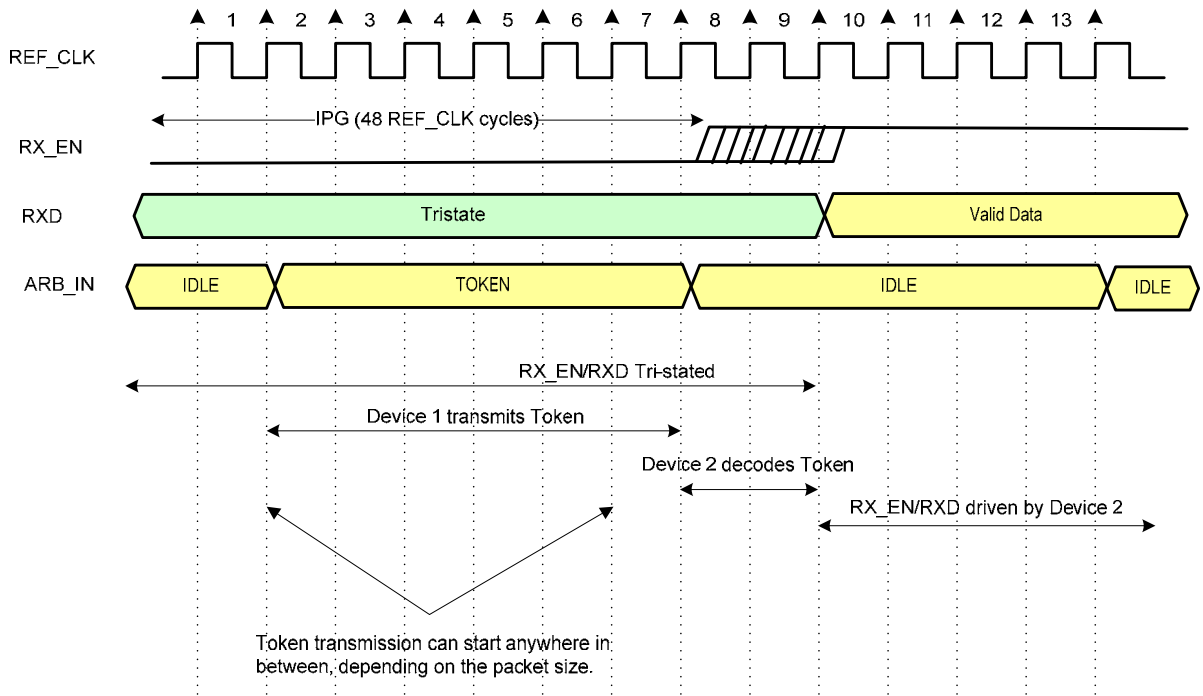
1343 Each Network Controller package that supports hardware-based arbitration control shall implement a  
1344 timeout mechanism in case the TOKEN op-code is not received. When a package has a packet to send, it  
1345 starts its timer. If it does not receive a TOKEN prior to the TOKEN timeout, the package shall send a  
1346 FLUSH op-code. This restarts the arbitration process.

1347 The timer may be programmable depending on the number of packages in the ring. The timeout value is  
1348 designed to accommodate up to four packages, each sending the largest packet (1536 bytes) plus  
1349 possible XON or XOFF frame transmission and op-code processing time. The timeout shall be no fewer  
1350 than T8 cycles of the REF\_CLK.

1351 **7.2.8 Timing Considerations**

1352 The ARB\_OUT and ARB\_IN pins shall follow the timing specifications outlined in Clause 10.

1353 To improve the efficiency of the multi-drop NC-SI, TOKEN op-code generation may overlap the Inter  
 1354 Packet Gap (IPG) defined by the [802.3](#) specification, as shown in Figure 12. The TOKEN op-code shall  
 1355 be sent no earlier than the last T13 REF\_CLK cycles of the IPG.



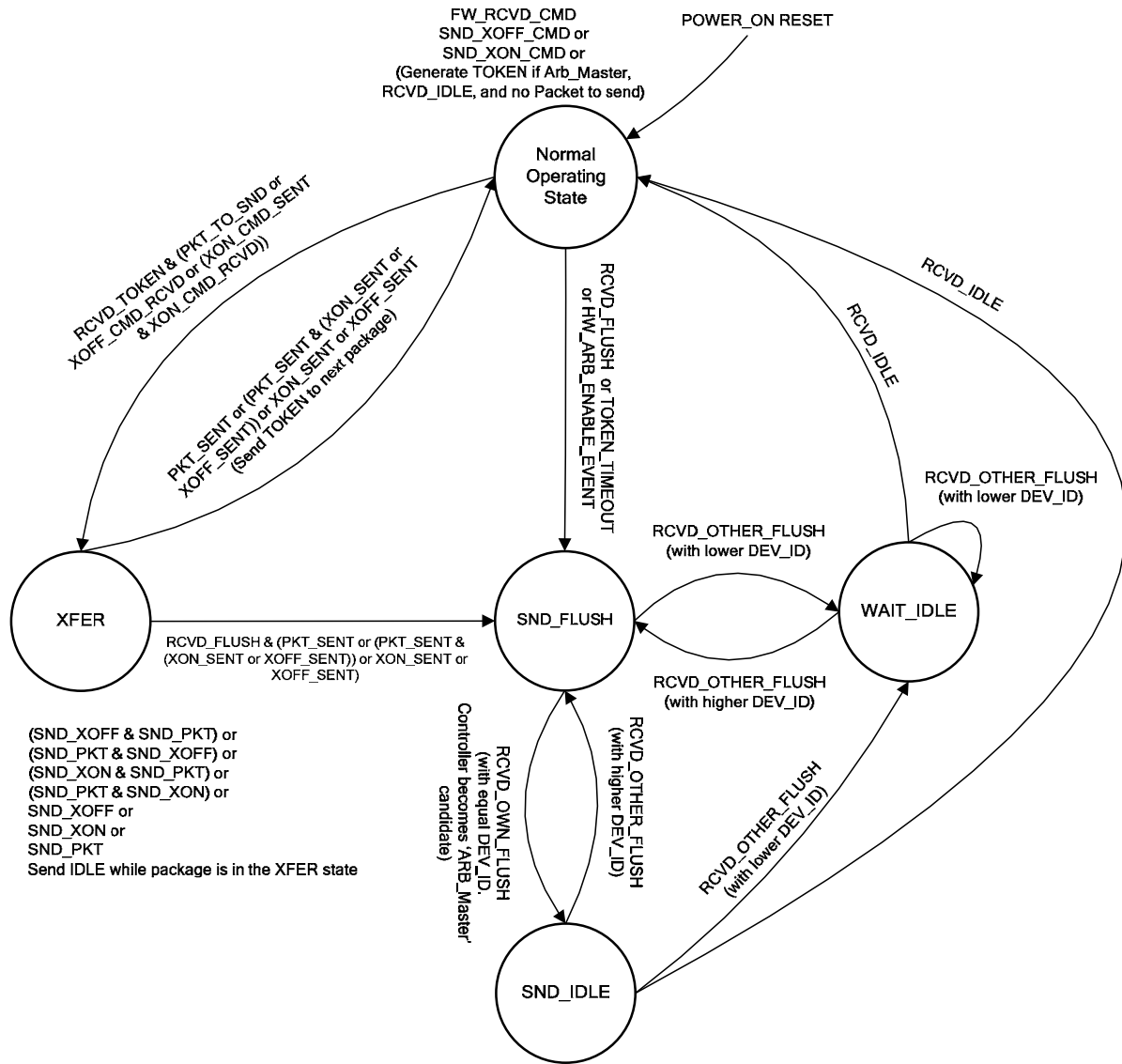
1356

1357

**Figure 12 – Example TOKEN to Transmit Relationship**

1358 **7.2.9 Example Hardware Arbitration State Machine**

1359 The state machine diagram shown in Figure 13 is provided as a guideline to help illustrate the startup  
 1360 process and op-code operations described in the preceding clauses. Where Figure 13 may vary from the  
 1361 preceding specifications, the preceding specifications shall take precedence.



1362  
 1363 **Figure 13 – Hardware Arbitration State Machine**

1364 The states and events shown in Figure 13 are described in Table 6 and Table 7, respectively.

1365 **Table 6 – Hardware Arbitration States**

State	Action
Normal Operating State	<p>This state is the normal operating state for hardware arbitration. The following actions happen in this state:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• 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.</li> <li>• 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.</li> <li>• The RXD lines will be in a high-impedance condition in this state.</li> </ul>
XFER	<p>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.</p> <p>The following actions happen in this state:</p> <ul style="list-style-type: none"> <li>• SND_XON: Transmit an XON frame (Pause Off) to the Management Controller.</li> <li>• SND_XOFF: Transmit an XOFF frame (Pause On) to the Management Controller.</li> <li>• SND_PKT: Transmit a Pass-through packet, response packet, or AEN to the Management Controller.</li> <li>• The TOKEN op-code is sent to the next package upon completion of the transfer.</li> </ul>
SND_FLUSH	<p>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 the package's own DEV_ID.</p>
SND_IDLE	<p>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.</p>
WAIT_IDLE	<p>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.</p>



1366

Table 7 – Hardware Arbitration Events

Event	Description
RCVD_TOKEN	A TOKEN op-code was received.
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.

### 1367 7.3 Command-based Arbitration

1368 If hardware arbitration is not being used, the Select Package and Deselect Package commands can be  
 1369 used to control which Network Controller package has the ability to transmit on the RXD lines. Because  
 1370 only one Network Controller package is allowed to transmit on the RXD lines, the Management Controller  
 1371 shall only have one package in the selected state at any given time. For more information, see 8.4.5 and  
 1372 8.4.7.

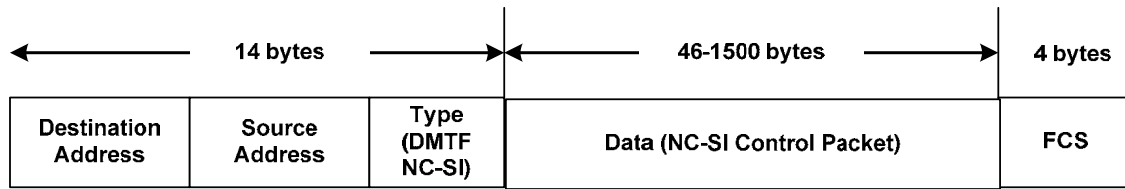
## 1373 8 Packet Definitions

1374 This clause presents the formats of NC-SI packets and their relationship to frames used to transmit and  
 1375 receive those packets on NC-SI.

### 1376 8.1 NC-SI Packet Encapsulation

1377 The NC-SI is an Ethernet interface adhering to the standard [IEEE 802.3](#) Ethernet frame format. Whether  
 1378 or not the Network Controller accepts runt packets is unspecified.

1379 As shown in Figure 14, this L2, or data link layer, frame format encapsulates all NC-SI packets, including  
 1380 command and response packets, as the L2 frame payload data by adding a 14-byte header to the front of  
 1381 the data and appending a 4-byte Frame Check Sequence (FCS) to the end.



1382

1383

Figure 14 – Ethernet Frame Encapsulation of NC-SI Packet Data

1384 **8.1.1 Ethernet Frame Header**

1385 The Management Controller shall format the 14-byte Ethernet frame header so that when it is received, it  
 1386 shall be formatted in the big-endian byte order shown in Table 8.

1387 Channels shall accept Pass-through packets that meet the [IEEE 802.3](#) frame requirements.

1388

Table 8 – Ethernet Header Format

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..03	DA <sub>5</sub> = 0xFF	DA <sub>4</sub> = 0xFF	DA <sub>3</sub> = 0xFF	DA <sub>2</sub> = 0xFF
04..07	DA <sub>1</sub> = 0xFF	DA <sub>0</sub> = 0xFF	SA <sub>5</sub>	SA <sub>4</sub>
08..11	SA <sub>3</sub>	SA <sub>2</sub>	SA <sub>1</sub>	SA <sub>0</sub>
12..13	EtherType = 0x88F8 (DMTF NC-SI)			

1389 **8.1.1.1 Destination Address (DA)**

1390 Bytes 0–5 of the header represent bytes 5–0 of the Ethernet Destination Address field of an L2 header.

1391 The channel is not assigned a specific MAC address and the contents of this field are not interpreted as a  
 1392 MAC address by the Management Controller or the Network Controller. However, the DA field in all NC-  
 1393 SI control packets shall be set to the broadcast address (FF:FF:FF:FF:FF:FF) for consistency.

1394 If the Network Controller receives a control packet with a Destination Address other than  
 1395 FF:FF:FF:FF:FF:FF, the Network Controller may elect to accept the packet, drop it, or return a response  
 1396 packet with an error response/reason code.

1397 **8.1.1.2 Source Address (SA)**

1398 Bytes 6–11 of the header represent bytes 5–0 of the Ethernet Source Address field of the Ethernet  
 1399 header. The contents of this field may be set to any value. The Network Controller may use  
 1400 FF:FF:FF:FF:FF:FF as the source address for NC-SI Control packets that it generates.

1401 **8.1.1.3 EtherType**

1402 The final two bytes of the header, bytes 12..13, represent bytes 1..0 of the EtherType field of the Ethernet  
 1403 header. For NC-SI Control packets, this field shall be set to a fixed value of 0x88F8 as assigned to the  
 1404 NC-SI by the IEEE. This value allows NC-SI Control packets to be differentiated from other packets in the  
 1405 overall packet stream.

### 1406 8.1.2 Frame Check Sequence

1407 The Frame Check Sequence (FCS) shall be added at the end of the frame to provide detection of  
1408 corruption of the frame. Any frame with an invalid FCS shall be discarded.

## 1409 8.2 Control Packet Data Structure

1410 Each NC-SI Control packet is made up of a 16-byte packet header and a payload section whose length is  
1411 specific to the packet type.

### 1412 8.2.1 Control Packet Header

1413 The 16-byte control packet header is used in command, response, and AEN packets, and contains data  
1414 values intended to allow the packet to be identified, validated, and processed. The packet header is in  
1415 big-endian byte order, as shown in Table 9.

1416 **Table 9 – Control Packet Header Format**

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

#### 1417 8.2.1.1 Management Controller ID

1418 In Control packets, this 1-byte field identifies the Management Controller issuing the packet. For this  
1419 version of the specification, Management Controllers should set this field to 0x00 (zero). Network  
1420 Controllers responding to command packets should copy the Management Controller ID field from the  
1421 command packet header into the response packet header. For AEN packets, this field should be copied  
1422 from the parameter that was set using the AEN Enable command.

#### 1423 8.2.1.2 Header Revision

1424 This 1-byte field identifies the version of the Control packet header in use by the sender. For this version  
1425 of the specification, the header revision is 0x01.

#### 1426 8.2.1.3 Instance ID (IID)

1427 This 1-byte field contains the IID of the command and associated response. The Network Controller can  
1428 use it to differentiate retried commands from new instances of commands. The Management Controller  
1429 can use this value to match a received response to the previously sent command. For more information,  
1430 see 6.3.1.1.

#### 1431 8.2.1.4 Control Packet Type

1432 This 1-byte field contains the Identifier that is used to identify specific commands and responses, and to  
1433 differentiate AENs from responses. Each NC-SI command is assigned a unique 7-bit command type  
1434 value in the range 0x00 . . 0x7F. The proper response type for each command type is formed by setting  
1435 the most significant bit (bit 7) in the original 1-byte command value. This allows for a one-to-one  
1436 correspondence between 128 unique response types and 128 unique command types.

1437 **8.2.1.5 Channel ID**

1438 This 1-byte field contains the Network Controller Channel Identifier. The Management Controller shall set  
1439 this value to specify the package and internal channel ID for which the command is intended.

1440 In a multi-drop configuration, all commands are received by all NC-SI Network Controllers present in the  
1441 configuration. The Channel ID is used by each receiving Network Controller to determine if it is the  
1442 intended recipient of the command. In Responses and AENs, this field carries the ID of the channel from  
1443 which the response of AEN was issued.

1444 **8.2.1.6 Payload Length**

1445 This 12-bit field contains the length, in bytes, of any payload data present in the command or response  
1446 frame following the NC-SI packet header. This value does not include the length of the NC-SI header, the  
1447 checksum value, or any padding that might be present.

1448 **8.2.1.7 Reserved**

1449 These fields are reserved for future use and should be written as zeros and ignored when read.

1450 **8.2.2 Control Packet Payload**

1451 The NC-SI packet payload may contain zero or more defined data values depending on whether the  
1452 packet is a command or response packet, and on the specific type. The NC-SI packet payload is always  
1453 formatted in big-endian byte order, as shown in Table 10.

1454 **Table 10 – Generic Example of Control Packet Payload**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..03	Data0 <sub>3</sub>	Data0 <sub>2</sub>	Data0 <sub>1</sub>	Data0 <sub>0</sub>
04..07	Data1 <sub>7</sub>	Data1 <sub>6</sub>	Data1 <sub>5</sub>	Data1 <sub>4</sub>
08..11	Data1 <sub>3</sub>	Data1 <sub>2</sub>	Data1 <sub>1</sub>	Data1 <sub>0</sub>
..				
...	DataN-1 <sub>4</sub>	DataN-1 <sub>3</sub>	DataN-1 <sub>2</sub>	DataN-1 <sub>1</sub>
...	DataN-1 <sub>0</sub>	Payload Pad (as required)		
...	2s Complement Checksum Compensation			
...	Ethernet Packet Pad (as required)			

1455 **8.2.2.1 Data**

1456 As shown in Table 10, the bytes following the NC-SI packet header may contain payload data fields of  
1457 varying sizes, and which may be aligned or require padding. In the case where data is defined in the  
1458 payload, all data-field byte layouts (Data0–Data-1) shall use big-endian byte ordering with the most  
1459 significant byte of the field in the lowest addressed byte position (that is, coming first).

1460 **8.2.2.2 Payload Pad**

1461 If the payload is present and does not end on a 32-bit boundary, one to three padding bytes equal to  
1462 0x00 shall be present to align the checksum field to a 32-bit boundary.

### 1463 8.2.2.3 2's Complement Checksum Compensation

1464 This 4-byte field contains the 32-bit checksum compensation value that may be included in each  
1465 command and response packet by the sender of the packet. When it is implemented, the checksum  
1466 compensation shall be computed as the 2's complement of the checksum, which shall be computed as  
1467 the 32-bit unsigned sum of the NC-SI packet header and NC-SI packet payload interpreted as a series of  
1468 16-bit unsigned integer values. A packet receiver supporting packet checksum verification shall use the  
1469 checksum compensation value to verify packet data integrity by computing the 32-bit checksum described  
1470 above, adding to it the checksum compensation value from the packet, and verifying that the result is 0.

1471 Verification of non-zero NC-SI packet checksum values is optional. An implementation may elect to  
1472 generate the checksums and may elect to verify checksums that it receives. The checksum field is  
1473 generated and handled according to the following rules:

- 1474 • A checksum field value of all zeros specifies that a header checksum is not being provided for  
1475 the NC-SI Control packet, and that the checksum field value shall be ignored when processing  
1476 the packet.
- 1477 • If the originator of an NC-SI Control packet is not generating a checksum, the originator shall  
1478 use a value of all zeros for the header checksum field.
- 1479 • If a non-zero checksum field is generated for an NC-SI Control packet, that header checksum  
1480 field value shall be calculated using the specified algorithm.
- 1481 • All receivers of NC-SI Control packets shall accept packets with all zeros as the checksum  
1482 value (provided that other fields and the CRC are correct).
- 1483 • The receiver of an NC-SI Control packet may reject (silently discard) a packet that has an  
1484 incorrect non-zero checksum.
- 1485 • The receiver of an NC-SI Control packet may ignore any non-zero checksums that it receives  
1486 and accept the packet, even if the checksum value is incorrect (that is, an implementation is not  
1487 required to verify the checksum field).
- 1488 • A controller that generates checksums is not required to verify checksums that it receives.
- 1489 • A controller that verifies checksums is not required to generate checksums for NC-SI Control  
1490 packets that it originates.

### 1491 8.2.2.4 Ethernet Packet Pad

1492 Per [IEEE 802.3](#), all Ethernet frames shall be at least 64 bytes in length, from the DA through and  
1493 including FCS. For NC-SI packets, this requirement applies to the Ethernet header and payload, which  
1494 includes the NC-SI Control packet header and payload. Most NC-SI Control packets are less than the  
1495 minimum Ethernet frame payload size of 46 bytes in length and require padding to comply with  
1496 [IEEE 802.3](#).

### 1497 8.2.3 Command Packet Payload

1498 Command packets have no common fixed payload format.

### 1499 8.2.4 Response Packet Payload

1500 Unlike command packets that do not necessarily contain payload data, all response packets carry at least  
1501 a 4-byte payload. This default payload carries the response codes and reason codes (described in 8.2.5)  
1502 that provide status on the outcome of processing the originating command packet, and is present in all  
1503 response packet payload definitions.

1504 The default payload occupies bytes 00 . . 03 of the response packet payload, with any additional  
 1505 response-packet-specific payload defined to follow starting on the next word. All response packet payload  
 1506 fields are defined with big-endian byte ordering, as shown in Table 11.

1507 **Table 11 – Generic Example of Response Packet Payload Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..03	Response Code		Reason Code	
..	...	...	...	...
...	DataN-1 <sub>4</sub>	DataN-1 <sub>3</sub>	DataN-1 <sub>2</sub>	DataN-1 <sub>1</sub>
...	DataN-1 <sub>0</sub>	Word Pad (as required)		
...	2s Complement Checksum Compensation			
...	Ethernet Packet Pad (as required)			

1508 **8.2.5 Response Codes and Reason Codes**

1509 Response codes and reason codes are status values that are returned in the responses to NC-SI  
 1510 commands. The response code values provide a general categorization of the status being returned. The  
 1511 reason code values provide additional detail related to a particular response code.

1512 **8.2.5.1 General**

1513 Response codes and reason codes are divided into numeric ranges that distinguish whether the values  
 1514 represent standard codes that are defined in this specification or are vendor/OEM-specific values that are  
 1515 defined by the vendor of the controller.

1516 The response code is a 2-byte field where values from 0x00 through 0x7F are reserved for definition by  
 1517 this specification. Values from 0x80 through 0xFF are vendor/OEM-specific codes that are defined by the  
 1518 vendor of the controller.

1519 The reason code is a 2-byte field. The ranges of values are defined in Table 12.

1520 **Table 12 – 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.

MS-byte	LS-byte	Description
Command Number	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.
	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.

1521 **8.2.5.2 Response Code and Reason Code Values**

1522 The standard response code values are defined in Table 13, and the standard reason code values are  
1523 defined in Table 14. Command-specific values, if any, are defined in the clauses that describe the  
1524 response data for the command. Unless otherwise specified, the standard reason codes may be used in  
1525 combination with any response code.

1526 **Table 13 – Standard Response Code Values**

Value	Description	Comment
0x0	Command Completed	Returned for a successful command completion
0x1	Command Failed	Returned to report that a valid command could not be processed or failed to complete correctly
0x2	Command Unavailable	Returned to report that a command is temporarily unavailable for execution because the controller is in a transient state or busy condition
0x3	Command Unsupported	Returned to report that a command is not supported by the implementation
0x8000–0xFFFF	Vendor/OEM-specific	Response codes defined by the vendor of the controller

1527 **Table 14 – 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 when the channel is in the Initial State, until the channel receives a Clear Initial State command
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	May be returned when the channel is in a transient state in which it is unable to process commands normally

Value	Description	Comment
0x0004	Package Not Ready	May be 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	The payload length in the command is incorrect for the given command
0x7FFF	Unknown / Unsupported Command Type	Returned when the command type is unknown or unsupported
0x8000-0xFFFF	OEM Reason Code	Vendor-specific reason code defined by the vendor of the controller

1528 **8.2.6 AEN Packet Format**

1529 AEN packets shall follow the general packet format of Control packets, with the IID field set to 0 because,  
 1530 by definition, the Management Controller does not send a response packet to acknowledge an AEN  
 1531 packet. The Control Packet Type field shall have the value 0xFF. The originating Network Controller shall  
 1532 fill in the Channel ID (Ch. ID) field with its own ID to identify itself as the source of notification. Currently,  
 1533 three AEN types are defined in the AEN Type field. Table 15 represents the general AEN packet format.

1534 **Table 15 – AEN Packet Format**

Bytes	Bits			
	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 = 0x4
08..11	Reserved			
12..15	Reserved			
16..19	Reserved			AEN Type
20..23	OPTIONAL AEN Data			
24..27	Checksum			

1535 **8.2.7 AEN Packet Data Structure**

1536 The AEN Type field (8-bit) has the values shown in Table 16.

1537 **Table 16 – AEN Types**

Value	AEN Type
0x0	Link Status Change
0x1	Configuration Required
0x2	Host NC Driver Status Change
0x3..0x7F	Reserved
0x80..0xFF	OEM-specific AENs



1538 **8.3 Control Packet Type Definitions**

1539 Command packet types are in the range of 0x00 to 0x7F. Table 17 describes each command, its  
 1540 corresponding response, and the type value for each.

1541 Mandatory (M), Optional (O), and Conditional (C) refer to command support requirements for the Network  
 1542 Controller.

1543 **Table 17 – Command and Response Types**

Command Type	Command Name	Description	Response Type	Command Support Requirement
0x00	Clear Initial State	Used by the Management Controller to acknowledge that the Network Controller is in the Initial State	0x80	M
0x01	Select Package	Used to explicitly select a controller package to transmit packets through the NC-SI interface	0x81	M
0x02	Deselect Package	Used to explicitly instruct the controller package to stop transmitting packets through the NC-SI interface	0x82	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
0x04	Disable Channel	Used to disable the NC-SI channel and to cause the forwarding of bidirectional Management Controller packets to cease	0x84	M
0x05	Reset Channel	Used to synchronously put the Network Controller back to the Initial State	0x85	M
0x06	Enable Channel Network TX	Used to explicitly enable the channel to transmit Pass-through packets onto the network	0x86	M
0x07	Disable Channel Network TX	Used to explicitly disable the channel from transmitting Pass-through packets onto the network	0x87	M
0x08	AEN Enable	Used to control generating AENs	0x88	C
0x09	Set Link	Used during OS absence to force link settings, or to return to auto-negotiation mode	0x89	M
0x0A	Get Link Status	Used to get current link status information	0x8A	M
0x0B	Set VLAN Filter	Used to program VLAN IDs for VLAN filtering	0x8B	M
0x0C	Enable VLAN	Used to enable VLAN filtering of Management Controller RX packets	0x8C	M
0x0D	Disable VLAN	Used to disable VLAN filtering	0x8D	M
0x0E	Set MAC Address	Used to configure and enable unicast and multicast MAC address filters	0x8E	M
0x10	Enable Broadcast Filtering	Used to enable full or selective broadcast packet filtering	0x90	M

Command Type	Command Name	Description	Response Type	Command Support Requirement
0x11	Disable Broadcast Filtering	Used to disable all broadcast packet filtering, and to enable the forwarding of broadcast packets	0x91	M
0x12	Enable Global Multicast Filtering	Used to disable forwarding of all multicast packets to the Management Controller	0x92	C
0x13	Disable Global Multicast Filtering	Used to enable forwarding of all multicast packets to the Management Controller	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
0x50	OEM Command	Used to request vendor-specific data	0xD0	O
Key: M = Mandatory (required) O = Optional C = Conditional (see command description)				

## 1544 8.4 Command and Response Packet Formats

1545 This clause describes the format for each of the NC-SI Commands and corresponding responses.

1546 The corresponding response packet format shall be mandatory when a given command is supported.

### 1547 8.4.1 NC-SI Command Frame Format

1548 Table 18 illustrates the NC-SI frame format that shall be accepted by the Network Controller.

1549 **Table 18 – Example of Complete Minimum-Sized NC-SI Command Packet**

Bytes	Bits			
	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

Bytes	Bits			
	31..24	23..16	15..08	07..00
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			

1550 **8.4.2 NC-SI Response Packet Format**

1551 Table 19 illustrates the NC-SI response packet format that shall be transmitted by the Network Controller.

1552 **Table 19 – Example of Complete Minimum-Sized NC-SI Response Packet**

Bytes	Bits			
	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			

1553 **8.4.3 Clear Initial State Command (0x00)**

1554 The Clear Initial State command provides the mechanism for the Management Controller to acknowledge  
 1555 that it considers a channel to be in the Initial State (typically because the Management Controller received  
 1556 an “Interface Initialization Required” reason code) and to direct the Network Controller to start accepting  
 1557 commands for initializing or recovering the NC-SI operation. When in the Initial State, the Network  
 1558 Controller shall return the “Interface Initialization Required” reason code for all commands until it receives  
 1559 the Clear Initial State command.

1560 If the channel is in the Initial State when it receives the Clear Initial State command, the command shall  
 1561 cause the Network Controller to stop returning the “Interface Initialization Required” reason code. The  
 1562 channel shall also treat any subsequently received instance ID numbers as IDs for new command  
 1563 instances, not retries.

1564 If the channel is not in the Initial State when it receives this command, it shall treat any subsequently  
 1565 received instance ID numbers as IDs for new command instances, not retries.

1566 Table 20 illustrates the packet format of the Clear Initial State command.

1567 **Table 20 – Clear Initial State Command Packet Format**

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

1568 **8.4.4 Clear Initial State Response (0x80)**

1569 Currently no command-specific reason code is identified for this response (see Table 21).

1570 **Table 21 – Clear Initial State 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	Checksum			
24..45	Pad			

1571 **8.4.5 Select Package Command (0x01)**

1572 A package is considered to be “selected” when its NC-SI output buffers are allowed to transmit packets  
 1573 through the NC-SI interface. Conversely, a package is “deselected” when it is not allowed to transmit  
 1574 packets through the NC-SI interface.

1575 The Select Package command provides a way for a Management Controller to explicitly take a package  
 1576 out of the deselected state and to control whether hardware arbitration is enabled for the package.  
 1577 (Similarly, the Deselect Package command allows a Management Controller to explicitly deselect a  
 1578 package.)

- 1579 The NC-SI package in the Network Controller shall also become selected if the package receives any
- 1580 other NC-SI command that is directed to the package or to a channel within the package.
  
- 1581 The Select Package command is addressed to the package, rather than to a particular channel (that is,
- 1582 the command is sent with a Channel ID where the Package ID subfield matches the ID of the intended
- 1583 package and the Internal Channel ID subfield is set to 0x1F).
  
- 1584 More than one package can be in the selected state simultaneously if hardware arbitration is used
- 1585 between the selected packages and is active. The hardware arbitration logic ensures that buffer conflicts
- 1586 will not occur between selected packages.
  
- 1587 If hardware arbitration is not active or is not used for a given package, only one package shall be selected
- 1588 at a time. To switch between packages, the Deselect Package command shall be issued to put the
- 1589 presently selected package into the deselected state before another package is selected.
  
- 1590 A package shall also become selected if it receives any command that is directed to the package or to a
- 1591 channel within the package.
  
- 1592 A package shall stay in the selected state until it receives a Deselect Package command, unless an
- 1593 internal condition causes all internal channels to enter the Initial State.
  
- 1594 A package that is not using hardware arbitration may leave its output buffers enabled for the time that it is
- 1595 selected, or it may place its output buffers into the high-impedance state between transmitting packets
- 1596 through the NC-SI interface. (Temporarily placing the output buffers into the high-impedance state is not
- 1597 the same as entering the deselected state.)
  
- 1598 For Type A integrated controllers: Because the bus buffers are separately controlled, a separate Select
- 1599 Package command needs to be sent to each Package ID in the controller that is to be enabled to transmit
- 1600 through the NC-SI interface. If the internal packages do not support hardware arbitration, only one
- 1601 package shall be selected at a time; otherwise, a bus conflict will occur.
  
- 1602 For Type S single channel, and Types B and C integrated controllers: A single set of bus buffers exists for
- 1603 the package. Sending a Select Package command selects the entire package and enables all channels
- 1604 within the package to transmit through the NC-SI interface. (Whether a particular channel in a selected
- 1605 package starts transmitting Pass-through and AEN packets depends on whether that channel was
- 1606 enabled or disabled using the Enable or Disable Channel commands and whether the package may have
- 1607 had packets queued up for transmission.)
  
- 1608 Table 22 illustrates the packet format of the Select Package command. Table 23 illustrates the disable
- 1609 byte for hardware arbitration.

1610

**Table 22 – Select Package Command Packet Format**

	Bits			
Bytes	31..24	23..16	15..08	07..00
00..15	NC-SI Header			
16..19	Reserved			Hardware Arbitration Disable
20..23	Checksum			
24..45	Pad			

1611

**Table 23 – Hardware Arbitration Disable Byte**

Bits	Description
7..1	Reserved
0	<p>0b = Hardware arbitration between packages is enabled.</p> <p>1b = Disable hardware arbitration. Disabling hardware arbitration causes the package's arbitration logic to enter or remain in bypass mode.</p> <p>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.</p>

**1612 8.4.6 Select Package Response (0x81)**

1613 Currently no command-specific reason code is identified for this response (see Table 24).

1614

**Table 24 – Select Package Response Packet Format**

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

**1615 8.4.7 Deselect Package Command (0x02)**

1616 The Deselect Package command directs the controller package to stop transmitting packets through the  
1617 NC-SI interface and to place the output buffers for the package into the high-impedance state.

1618 The Deselect Package command is addressed to the package, rather than to a particular channel (that is,  
1619 the command is sent with a Channel ID where the Package ID subfield matches the ID of the intended  
1620 package and the Internal Channel ID subfield is set to 0x1F).

1621 The controller package enters the deselected state after it has transmitted the response to the Deselect  
1622 Package command and placed its buffers into the high-impedance state. The controller shall place its  
1623 outputs into the high-impedance state within the Package Deselect to Hi-Z Interval (T1). (This interval  
1624 gives the controller being deselected time to turn off its electrical output buffers after sending the  
1625 response to the Deselect Package command.)

1626 It is recommended that a Network Controller should become deselected if it receives any command traffic  
1627 directed to a different package ID as this suggests the Management Controller is attempting to  
1628 communicate with another device.

1629 If hardware arbitration is not supported or used, the Management Controller should wait for the Package  
1630 Deselect to Hi-Z Interval (T1) to expire before selecting another controller.

1631 For Type A integrated controllers: Because the bus buffers are separately controlled, putting the overall  
1632 controller package into the high-impedance state requires sending separate Deselect Package  
1633 commands to each Package ID in the overall package.

1634 For Type S single channel, and Types B and C integrated controllers: A single set of bus buffers exists for  
 1635 the package. Sending a Deselect Package command deselects the entire NC-SI package and prevents  
 1636 all channels within the package from transmitting through the NC-SI interface.

1637 Table 25 illustrates the packet format of the Deselect Package command.

1638 **Table 25 – Deselect Package Command Packet Format**

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

1639 **8.4.8 Deselect Package Response (0x82)**

1640 The Network Controller shall always put the package into the deselected state after sending a Deselect  
 1641 Package Response.

1642 No command-specific reason code is identified for this response (see Table 26).

1643 **Table 26 – Deselect Package 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	Checksum			
24..45	Pad			

1644 **8.4.9 Enable Channel Command (0x03)**

1645 The Enable Channel command allows the Management Controller to begin the flow of Network Controller  
 1646 packets, including Pass-through and AEN, through the NC-SI.

1647 Table 27 illustrates the packet format of the Enable Channel command.

1648 **Table 27 – Enable Channel Command Packet Format**

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

1649 NOTE: It is currently unspecified whether the Enable Channel command by itself will cause the Network Controller to  
 1650 perform pass through from the Management Controller to the network, or if this can be enabled only by the Enable  
 1651 Channel Network TX command.

1652 **8.4.10 Enable Channel Response (0x83)**

1653 No command-specific reason code is identified for this response (see Table 28).

1654 **Table 28 – Enable Channel Response Packet Format**

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

1655 **8.4.11 Disable Channel Command (0x04)**

1656 The Disable Channel command allows the Management Controller to disable the flow of packets,  
1657 including Pass-through and AEN, to the Management Controller.

1658 A Network Controller implementation is not required to flush pending packets from its RX Queues when a  
1659 channel becomes disabled. If queuing is subsequently disabled for a channel, it is possible that a number  
1660 of packets from the disabled channel could still be pending in the RX Queues. These packets may  
1661 continue to be transmitted through the NC-SI interface until the RX Queues are emptied of those packets.  
1662 The Management Controller should be aware that it may receive a number of packets from the channel  
1663 before receiving the response to the Disable Channel command.

1664 The 1-bit Allow Link Down (ALD) field can be used by the Management Controller to indicate that the  
1665 specified channel will not be required to handle Pass-through traffic while disabled. The Network  
1666 Controller is allowed to take down the external network physical link if no other functionality (for example,  
1667 host OS or WoL [Wake-on-LAN]) is active.

1668 Possible values for the 1-bit ALD field are as follows:

- 1669
- 0b = Keep link up for Pass-through management traffic
  - 1b = Allow link to be taken down
- 1670

1671 Table 29 illustrates the packet format of the Disable Channel command.

1672 **Table 29 – Disable Channel Command Packet Format**

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

1673 NOTE: It is currently unspecified whether this command will cause the Network Controller to cease the pass through  
1674 of traffic from the Management Controller to the network, or if this can only be done using the Disable Channel  
1675 Network TX command.



1676 **8.4.12 Disable Channel Response (0x84)**

1677 No command-specific reason code is identified for this response (see Table 30).

1678 **Table 30 – Disable Channel Response Packet Format**

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

1679 **8.4.13 Reset Channel Command (0x05)**

1680 The Reset Channel command allows the Management Controller to put the channel into the Initial State.  
 1681 Packet transmission is not required to stop until the Reset Channel response has been sent. Thus, the  
 1682 Management Controller should be aware that it may receive a number of packets from the channel before  
 1683 receiving the response to the Reset Channel command.

1684 Table 31 illustrates the packet format of the Reset Channel command.

1685 **Table 31 – Reset Channel Command Packet Format**

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

1686 **8.4.14 Reset Channel Response (0x85)**

1687 Currently no command-specific reason code is identified for this response (see Table 32).

1688 **Table 32 – Reset Channel Response Packet Format**

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

1689 **8.4.15 Enable Channel Network TX Command (0x06)**

1690 The Enable Channel Network TX command enables the channel to transmit Pass-through packets onto  
 1691 the network. After network transmission is enabled, this setting shall remain enabled until a Disable  
 1692 Channel Network TX command is received or the channel enters the Initial State.

1693 The intention of this command is to control which Network Controller ports are allowed to transmit to the  
 1694 external network. The Network Controller compares the MAC address in outgoing Pass-through packets  
 1695 to the MAC addresses set using the Set MAC Address command. If a match exists, the packet is  
 1696 transmitted to the network.

1697 Table 33 illustrates the packet format of the Enable Channel Network TX command.

1698 **Table 33 – Enable Channel Network TX Command Packet Format**

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

1699 NOTE: It is currently unspecified whether any dependencies exist between the Enable Channel and the Enable  
 1700 Channel Network TX commands, and whether the Enable Channel Network TX command, if sent before the Enable  
 1701 Channel command has been sent, should cause the Network Controller to immediately start forwarding pass thru  
 1702 packets received from the Management Controller to the network.

1703 **8.4.16 Enable Channel Network TX Response (0x86)**

1704 No command-specific reason code is identified for this response (see Table 34).

1705 **Table 34 – Enable Channel Network TX 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	Checksum			
24..45	Pad			

1706 **8.4.17 Disable Channel Network TX Command (0x07)**

1707 The Disable Channel Network TX command disables the channel from transmitting Pass-through packets  
 1708 onto the network. After network transmission is disabled, it shall remain disabled until an Enable Channel  
 1709 Network TX command is received.

1710 Table 35 illustrates the packet format of the Disable Channel Network TX command.

1711 **Table 35 – Disable Channel Network TX Command Packet Format**

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

1712 **8.4.18 Disable Channel Network TX Response (0x87)**

1713 The NC-SI shall, in the absence of a checksum error or identifier mismatch, always accept the Disable  
1714 Channel Network TX command and send a response.

1715 Currently no command-specific reason code is identified for this response (see Table 36).

1716 **Table 36 – Disable Channel Network TX 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	Checksum			
24..45	Pad			

1717 **8.4.19 AEN Enable Command (0x08)**

1718 Network Controller implementations shall support this command on the condition that the Network  
1719 Controller generates one or more standard AENs. The AEN Enable command enables and disables the  
1720 different standard AENs supported by the Network Controller. The Network Controller shall copy the AEN  
1721 MC ID field from the AEN Enable command into the MC ID field in every subsequent AEN sent to the  
1722 Management Controller.

1723 For more information, see 8.5 ("AEN Packet Formats") and 8.2.1.1 ("Management Controller ID").

1724 Table 37 illustrates the packet format of the AEN Enable command.

1725 **Table 37 – AEN Enable Command Packet Format**

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

1726 The AEN Control field has the format shown in Table 38.

1727 **Table 38 – 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..15	Reserved	Reserved
16..31	OEM-specific AEN control	OEM-specific control

#### 1728 8.4.20 AEN Enable Response (0x88)

1729 Currently no command-specific reason code is identified for this response (see Table 39).

1730 **Table 39 – AEN Enable Response Packet Format**

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

#### 1731 8.4.21 Set Link Command (0x09)

1732 The Set Link command may be used by the Management Controller to configure the external network  
 1733 interface associated with the channel by using the provided settings. Upon receiving this command, the  
 1734 channel shall attempt to force the link to the configuration specified by the parameters. The channel shall  
 1735 send a response packet to the Management Controller within the required response time. However, the  
 1736 requested link state changes may take an unspecified amount of time to complete.

1737 Table 40 illustrates the packet format of the Set Link command.

1738 **Table 40 – Set Link Command Packet Format**

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

1739 Table 41 and Table 42 describe the Set Link bit definitions. Refer to [IEEE 802.3](#) for definitions of Auto  
 1740 Negotiation, Duplex Setting, Pause Capability, and Asymmetric Pause Capability.

1741 **Table 41 – 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).	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
		Bits 05..07: RESERVED
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).	Bit 08: 1b = enable half-duplex
		Bit 09: 1b = enable full-duplex
10	Pause Capability	1b = disable 0b = enable
11	Asymmetric Pause Capability	1b = enable 0b = disable
12	OEM Link Settings Field Valid (see Table 42)	1b = enable 0b = disable
13..31	Reserved	0

1742

Table 42 – OEM Set Link Bit Definitions

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

1743

**8.4.22 Set Link Response (0x89)**

1744

The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Set Link command and send a response (see Table 43).

1745

1746

Table 43 – Set Link Response Packet Format

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

1747

Table 44 describes the reason codes that are specific to the Set Link command. Returning the following command-specific codes is recommended, conditional upon Network Controller support for the related capabilities.

1748

1749

1750

Table 44 – Set Link Command-Specific Reason Codes

Value	Description	Comment
0x1	Set Link Host OS/ Driver Conflict	Returned when the Set Link command is received when the Host NC driver is operational
0x2	Set Link Media Conflict	Returned when Set Link command parameters conflict with the media type (for example, Fiber Media)
0x3	Set Link Parameter Conflict	Returned when Set Link parameters conflict with each other (for example, 1000 Mbps HD with copper media)
0x4	Set Link Power Mode Conflict	Returned when Set Link parameters conflict with current low-power levels by exceeding capability
0x5	Set Link Speed Conflict	Returned when Set Link parameters attempt to force more than one speed at the same time
0x6	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

1751 **8.4.23 Get Link Status Command (0x0A)**

1752 The Get Link Status command allows the Management Controller to query the channel for potential link  
 1753 status and error conditions (see Table 45).

1754 **Table 45 – Get Link Status Command Packet Format**

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

1755 **8.4.24 Get Link Status Response (0x8A)**

1756 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get Link  
 1757 Status command and send a response (see Table 46).

1758 **Table 46 – Get Link Status Response Packet Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..15	NC-SI 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			

1759 Table 47 describes the Link Status bit definitions.

1760 **Table 47 – 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.

Bit Position	Field Description	Value Description
01..04	Speed and duplex	<p>0x0 = Auto-negotiate not complete [per <a href="#">IEEE 802.3</a>], SerDes Flag = 1b, or no Highest Common Denominator (HCD) from the following options (0x1 through 0x8) was found.</p> <p>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  0x9 – 0xf = RESERVED</p> <p>Except when SerDes = 1b, the value may reflect forced link setting.</p>
05	Auto Negotiate Flag	<p>1b = Auto-negotiation is enabled.</p> <p>This field always returns 0b if auto-negotiation is not supported.</p> <p>This field is mandatory if supported by the controller.</p>
06	Auto Negotiate Complete	<p>1b = Auto-negotiation has completed.</p> <p>This includes if auto-negotiation was completed using Parallel Detection. Always returns 0b if auto-negotiation is not supported or is not enabled.</p> <p>This field is mandatory if the Auto Negotiate Flag is supported.</p>
07	Parallel Detection Flag	<p>1b = Link partner did not support auto-negotiation and parallel detection was used to get link.</p> <p>This field contains 0b if Parallel Detection was not used to obtain link.</p>
08	Reserved	None
09	Link Partner Advertised Speed and Duplex 1000TFD	<p>1b = Link Partner is 1000BASE-T full-duplex capable.</p> <p>Valid when:</p> <p>SerDes Flag = 0b  Auto-Negotiate Flag = 1b  Auto-Negotiate Complete = 1b</p> <p>This field is mandatory.</p>
10	Link Partner Advertised Speed and Duplex 1000THD	<p>1b = Link Partner is 1000BASE-T half-duplex capable.</p> <p>Valid when:</p> <p>SerDes Flag = 0b  Auto-Negotiate Flag = 1b  Auto-Negotiate Complete = 1b</p> <p>This field is mandatory.</p>



Bit Position	Field Description	Value Description
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.
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	1b = Pause Flow Control enabled for TX on the external network interface. This field is mandatory.
17	RX Flow Control Flag	1b = Pause Flow Control enabled for RX on the external network interface. This field is mandatory.

Bit Position	Field Description	Value Description
18..19	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.
20	SerDes Link	SerDes status (see 4.18") 0b = SerDes not used 1b = SerDes used This field is mandatory.
21	OEM Link Speed Valid	0b = OEM link settings are invalid. 1b = OEM link settings are valid.
22..31	Reserved	0

1761 Table 48 describes the Other Indications field bit definitions.

1762 **Table 48 – 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). 1b = The Network Controller driver for the host external network interface associated with this channel is being reported as operational (running). This field is optional, and always returns 0b if the option is not supported.
01..31	Reserved	None

1763 Table 49 describes the OEM Link Status field bit definitions.

1764 **Table 49 – OEM Link Status Field Bit Definitions (Optional)**

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

1765 Table 50 describes the reason code that is specific to the Get Link Status command.

1766 **Table 50 – Get Link Status Command-Specific Reason Code**

Value	Description	Comment
0x6	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

#### 1767 **8.4.25 Set VLAN Filter Command (0x0B)**

1768 The Set VLAN Filter command is used by the Management Controller to program one or more VLAN IDs  
1769 that are used for VLAN filtering.

1770 Incoming packets that match both a VLAN ID filter and a MAC address filter are forwarded to the  
1771 Management Controller. Other packets may be dropped based on the VLAN filtering mode per the Enable  
1772 VLAN command.

1773 The quantity of each filter type that is supported by the channel can be discovered by means of the Get  
1774 Capabilities command. Up to 15 filters can be supported per channel. A Network Controller  
1775 implementation shall support at least one VLAN filter per channel.

1776 To configure a VLAN filter, the Management Controller issues a Set VLAN Filter command with the Filter  
1777 Selector field indicating which filter is to be configured, the VLAN ID field set to the VLAN TAG values to  
1778 be used by the filter, and the Enable field set to either enable or disable the selected filter.

1779 The VLAN-related fields are specified per [IEEE 802.1q](#). When VLAN Tagging is used, the packet includes  
1780 a Tag Protocol Identifier (TPID) field and VLAN Tag fields, as shown in Table 51.

1781 **Table 51 – IEEE 802.1q VLAN Fields**

Field	Size	Description
TPID	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

1782 When checking VLAN field values, the Network Controller shall match against the enabled VLAN Tag  
1783 Filter values that were configured with the Set VLAN Filter command. The Network Controller shall also  
1784 match on the TPID value of 8100h, as specified by [IEEE 802.1q](#). Matching against the User Priority/CFI  
1785 bits is optional. An implementation may elect to ignore the setting of those fields.

1786 Table 52 illustrates the packet format of the Set VLAN Filter command.

1787 **Table 52 – Set VLAN Filter Command Packet Format**

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

1788 Table 53 provides possible settings for the Filter Selector field. Table 54 provides possible settings for the  
 1789 Enable (E) field.

1790 **Table 53 – 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 <i>N</i>

1791 **Table 54 – Possible Settings for Enable (E) Field (1-Bit Field)**

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

1792 **8.4.26 Set VLAN Filter Response (0x8B)**

1793 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Set  
 1794 VLAN Filter command and send a response (see Table 55).

1795 **Table 55 – Set VLAN Filter Response Packet Format**

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

1796 Table 56 describes the reason code that is specific to the Set VLAN Filter command.

1797 **Table 56 – Set VLAN Filter Command-Specific Reason Code**

Value	Description	Comment
0x7	VLAN Tag Is Invalid	Returned when the VLAN ID is invalid (VLAN ID = 0)

1798 **8.4.27 Enable VLAN Command (0x0C)**

1799 The Enable VLAN command may be used by the Management Controller to enable the channel to accept  
1800 VLAN-tagged packets from the network for NC-SI Pass-through operation (see Table 57).

1801 **Table 57 – Enable VLAN Command Packet Format**

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

1802 Table 58 describes the modes for the Enable VLAN command.

1803 **Table 58 – 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

1804 **8.4.28 Enable VLAN Response (0x8C)**

1805 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Enable  
1806 VLAN command and send a response.

1807 Currently no command-specific reason code is identified for this response (see Table 59).

1808 **Table 59 – Enable VLAN 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	Checksum			
24..45	Pad			

1809 **8.4.29 Disable VLAN Command (0x0D)**

1810 The Disable VLAN command may be used by the Management Controller to disable VLAN filtering. In the  
1811 disabled state, only non-VLAN-tagged packets (that also match the MAC Address Filtering configuration)  
1812 are accepted. VLAN-tagged packets are not accepted.

1813 Table 60 illustrates the packet format of the Disable VLAN command.

1814 **Table 60 – Disable VLAN Command Packet Format**

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

1815 **8.4.30 Disable VLAN Response (0x8D)**

1816 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Disable  
1817 VLAN command and send a response.

1818 Currently no command-specific reason code is identified for this response (see Table 61).

1819 **Table 61 – Disable VLAN 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	Checksum			
24..45	Pad			

### 1820 8.4.31 Set MAC Address Command (0x0E)

1821 The Set MAC Address command is used by the Management Controller to program the channel's unicast  
1822 or multicast MAC address filters.

1823 The channel supports one or more "perfect match" MAC address filters that are used to selectively  
1824 forward inbound frames to the Management Controller. Assuming that a packet passes any VLAN filtering  
1825 that may be active, it will be forwarded to the Management Controller if its 48-bit destination MAC address  
1826 exactly matches an active MAC address filter.

1827 MAC address filters may be configured as unicast or multicast addresses, depending on the capability of  
1828 the channel. The channel may implement three distinct types of filter:

- 1829 • **Unicast filters** support exact matching on 48-bit unicast MAC addresses.
- 1830 • **Multicast filters** support exact matching on 48-bit multicast MAC addresses.
- 1831 • **Mixed filters** support exact matching on both unicast and multicast MAC addresses.

1832 The number of each type of filter that is supported by the channel can be discovered by means of the Get  
1833 Capabilities command. The channel shall support at least one unicast address filter or one mixed filter, so  
1834 that at least one unicast MAC address filter may be configured on the channel. Support for any  
1835 combination of unicast, multicast, or mixed filters beyond this basic requirement is vendor specific. The  
1836 total number of all filters shall be less than or equal to 8.

1837 To configure an address filter, the Management Controller issues a Set MAC Address command with the  
1838 Address Type field indicating the type of address to be programmed (unicast or multicast) and the MAC  
1839 Address Num field indicating the specific filter to be programmed.

1840 Filters are addressed using a 1-based index ordered over the unicast, multicast, and mixed filters  
1841 reported by means of the Get Capabilities command. For example, if the interface reports four unicast  
1842 filters, two multicast filters, and two mixed filters, then MAC Address numbers 1 through 4 refer to the  
1843 interface's unicast filters, 5 and 6 refer to the multicast filters, and 7 and 8 refer to the mixed filters.  
1844 Similarly, if the interface reports two unicast filters, no multicast filters, and six mixed filters, then MAC  
1845 address numbers 1 and 2 refer to the unicast filters, and 3 through 8 refer to the mixed filters.

1846 The filter type of the filter to be programmed (unicast, multicast, or mixed) shall be compatible with the  
1847 Address Type being programmed. For example, programming a mixed filter to a unicast address is  
1848 allowed, but programming a multicast filter to a unicast address is an error.

1849 The Enable field determines whether the indicated filter is to be enabled or disabled. When a filter is  
1850 programmed to be enabled, the filter is loaded with the 48-bit MAC address in the MAC Address field of  
1851 the command, and the channel enables forwarding of frames that match the configured address. If the  
1852 specified filter was already enabled, it is updated with the new address provided.

1853 When a filter is programmed to be disabled, the contents of the MAC Address field are ignored. Any  
1854 previous MAC address programmed in the filter is discarded and the channel no longer uses this filter in  
1855 its packet-forwarding function.

1856 Table 62 illustrates the packet format of the Set MAC Address command.

1857 **Table 62 – Set MAC Address Command Packet Format**

		Bits					
Bytes		31..24	23..16	15..08	07..00		
00..15	NC-SI 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.							

1858 Table 63 provides possible settings for the MAC Address Number field. Table 64 provides possible  
 1859 settings for the Address Type (AT) field. Table 65 provides possible settings for the Enable (E) field.

1860 **Table 63 – 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
..	
N	Configure MAC address filter number <i>N</i>

1861 **Table 64 – Possible Settings for Address Type (3-Bit Field)**

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

1862 **Table 65 – Possible Settings for Enable Field (1-Bit Field)**

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



1863 **8.4.32 Set MAC Address Response (0x8E)**

1864 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Set MAC  
1865 Address command and send a response (see Table 66).

1866 **Table 66 – Set MAC Address Response Packet Format**

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

1867 Table 67 describes the reason code that is specific to the Set MAC Address command.

1868 **Table 67 – Set MAC Address Command-Specific Reason Code**

Value	Description	Comment
0x8	MAC Address Is Zero	Returned when the Set MAC Address command is received with the MAC address set to 0

1869 **8.4.33 Enable Broadcast Filter Command (0x10)**

1870 The Enable Broadcast Filter command allows the Management Controller to control the forwarding of  
1871 broadcast frames to the Management Controller. The channel, upon receiving and processing this  
1872 command, shall filter all received broadcast frames based on the broadcast packet filtering settings  
1873 specified in the payload. If no broadcast packet types are specified for forwarding, all broadcast packets  
1874 shall be filtered out.

1875 The Broadcast Packet Filter Settings field is used to specify those protocol-specific broadcast filters that  
1876 should be activated. The channel indicates which broadcast filters it supports in the Broadcast Filter  
1877 Capabilities field of the Get Capabilities Response frame defined in 8.4.46.

1878 Table 68 illustrates the packet format of the Enable Broadcast Filter command.

1879 **Table 68 – Enable Broadcast Filter Command Packet Format**

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

1880 Table 69 describes the Broadcast Packet Filter Settings field bit definitions.

1881

**Table 69 – Broadcast Packet Filter Settings Field**

Bit Position	Field Description	Value Description
0	ARP Packets	<p>1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type.</p> <p>For the purposes of this specification, an ARP broadcast packet is defined to be any packet that meets all of the following requirements:</p> <ul style="list-style-type: none"> <li>• The destination MAC address field is set to the layer 2 broadcast address (FF:FF:FF:FF:FF:FF).</li> <li>• The EtherType field set to 0x0806.</li> </ul> <p>This field is mandatory.</p>
1	DHCP Client Packets	<p>1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type.</p> <p>For the purposes of this filter, a DHCP client broadcast packet is defined to be any packet that meets all of the following requirements:</p> <ul style="list-style-type: none"> <li>• The destination MAC address field is set to the layer 2 broadcast address (FF:FF:FF:FF:FF:FF).</li> <li>• The EtherType field is set to 0x0800 (IPv4).</li> <li>• The IP header's Protocol field is set to 17 (UDP).</li> <li>• The UDP destination port number is set to 68.</li> </ul> <p>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.</p>
2	DHCP Server Packets	<p>1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type.</p> <p>For the purposes of this filter, a DHCP server broadcast packet is defined to be any packet that meets all of the following requirements:</p> <ul style="list-style-type: none"> <li>• The destination MAC address field is set to the layer 2 broadcast address (FF:FF:FF:FF:FF:FF).</li> <li>• The EtherType field is set to 0x0800 (IPv4).</li> <li>• The IP header's Protocol field is set to 17 (UDP).</li> <li>• The UDP destination port number is set to 67.</li> </ul> <p>This field is optional. If unsupported, broadcast DHCP packets will be blocked when broadcast filtering is enabled. The value shall be set to 0 if unsupported.</p>

Bit Position	Field Description	Value Description
3	NetBIOS Packets	<p>1b = Forward this packet type to the Management Controller.                      0b = Filter out this packet type.</p> <p>For the purposes of this filter, NetBIOS broadcast packets are defined to be any packet that meets all of the following requirements:</p> <ul style="list-style-type: none"> <li>• The destination MAC address field is set to the layer 2 broadcast address (FF:FF:FF:FF:FF:FF).</li> <li>• The EtherType field is set to 0x0800 (IPv4).</li> <li>• The IP header's Protocol field is set to 17 (UDP).</li> <li>• 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.</li> </ul> <p>This field is optional. If unsupported, broadcast NetBIOS packets will be blocked when broadcast filtering is enabled. The value shall be set to 0 if unsupported.</p>
4..31	Reserved	None

1882 **8.4.34 Enable Broadcast Filter Response (0x90)**

1883 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Enable  
 1884 Broadcast Filter command and send a response.

1885 Currently no command-specific reason code is identified for this response (see Table 70).

1886 **Table 70 – Enable Broadcast Filter Response Packet Format**

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

1887 **8.4.35 Disable Broadcast Filter Command (0x11)**

1888 The Disable Broadcast Filter command may be used by the Management Controller to disable the  
 1889 broadcast filter feature and enable the reception of all broadcast frames. Upon processing this command,  
 1890 the channel shall discontinue the filtering of received broadcast frames.

1891 Table 71 illustrates the packet format of the Disable Broadcast Filter command.

1892 **Table 71 – Disable Broadcast Filter Command Packet Format**

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

### 1893 8.4.36 Disable Broadcast Filter Response (0x91)

1894 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Disable  
1895 Broadcast Filter command and send a response.

1896 Currently no command-specific reason code is identified for this response (see Table 72).

1897 **Table 72 – Disable Broadcast Filter 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	Checksum			
24..45	Pad			

### 1898 8.4.37 Enable Global Multicast Filter Command (0x12)

1899 The Enable Global Multicast Filter command is used to activate global filtering of multicast frames with  
1900 optional filtering of specific multicast protocols. Upon receiving and processing this command, the  
1901 channel shall deliver to the Management Controller only multicast frames that match protocol-specific  
1902 multicast filters enabled using this command or specific, multicast addresses that have been configured  
1903 and enabled using the Set MAC Address command.

1904 The Multicast Packet Filter Settings field is used to specify optional, protocol-specific multicast filters that  
1905 should be activated. The channel indicates which optional multicast filters it supports in the Multicast Filter  
1906 Capabilities field of the Get Capabilities Response frame defined in 8.4.46. The Management Controller  
1907 should not set bits in the Multicast Packet Filter Settings field that are not indicated as supported in the  
1908 Multicast Filter Capabilities field.

1909 IPv6 Neighbor Solicitation messages are not covered by the currently defined multicast filters. When  
1910 multicast, Neighbor Solicitation messages are sent to a Solicited Node multicast address that is derived  
1911 from the target node's IPv6 address. To enable forwarding of Solicited Node multicasts when global  
1912 multicast filtering is active, the Management Controller would configure a multicast or mixed MAC address  
1913 filter for the specific Solicited Node multicast address required, using the Set MAC Address command.

1914 This command shall be implemented if the channel implementation supports accepting all multicast  
1915 addresses. An implementation that does not support accepting all multicast addresses shall not  
1916 implement these commands. Pass-through packets with multicast addresses can still be accepted

1917 depending on multicast address filter support provided by the Set MAC Address command. Multicast filter  
 1918 entries that are set to enabled in the Set MAC Address command are accepted; all others are rejected.

1919 Table 73 illustrates the packet format of the Enable Global Multicast Filter command.

1920 **Table 73 – Enable Global Multicast Filter Command Packet Format**

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

1921 Table 74 describes the bit definitions for the Multicast Packet Filter Settings field.

1922 **Table 74 – Bit Definitions for Multicast Packet Filter Settings Field**

Bit Position	Field Description	Value Description
0	IPv6 Neighbor Advertisement	<p>1b = Forward this packet type to the Management Controller.                      0b = Filter out this packet type.</p> <p>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:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• The EtherType field is set to 0x86DD (IPv6).</li> <li>• The IPv6 header's Next Header field is set to 58 (ICMPv6).</li> <li>• The ICMPv6 header's Message Type field is set to the following value: 136 – Neighbor Advertisement.</li> </ul> <p>This field is optional.</p>
1	IPv6 Router Advertisement	<p>1b = Forward this packet type to the Management Controller.                      0b = Filter out this packet type.</p> <p>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:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• The EtherType field is set to 0x86DD (IPv6).</li> <li>• The IPv6 header's Next Header field is set to 58 (ICMPv6).</li> <li>• The ICMPv6 header's Message Type field is set to 134.</li> </ul> <p>This field is optional.</p>

Bit Position	Field Description	Value Description
2	DHCPv6 relay and server multicast	<p>1b = Forward this packet type to the Management Controller. 0b = Filter out this packet type.</p> <p>For the purposes of this filter, a DHCPv6 multicast packet is defined to be any packet that meets all of the following requirements:</p> <ul style="list-style-type: none"> <li>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).</li> <li>The EtherType field is set to 0x86DD (IPv6).</li> <li>The IPv6 header's Next Header field is set to 17 (UDP).</li> <li>The UDP destination port number is set to 547.</li> </ul> <p>This field is optional. If unsupported, multicast DHCP packets will be blocked when multicast filtering is enabled, unless they are matched by an address filter configured using the Set MAC Address command. The value shall be set to 0 if unsupported.</p>
3..31	Reserved	None

#### 1923 8.4.38 Enable Global Multicast Filter Response (0x92)

1924 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Enable  
1925 Global Multicast Filter command and send a response.

1926 Currently no command-specific reason code is identified for this response (see Table 75).

1927 **Table 75 – Enable Global Multicast Filter Response Packet Format**

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

#### 1928 8.4.39 Disable Global Multicast Filter Command (0x13)

1929 The Disable Global Multicast Filter command is used to disable global filtering of multicast frames. Upon  
1930 receiving and processing this command, and regardless of the current state of multicast filtering, the  
1931 channel shall forward all multicast frames to the Management Controller.

1932 This command shall be implemented on the condition that the channel implementation supports accepting  
1933 all multicast addresses. An implementation that does not support accepting all multicast addresses shall  
1934 not implement these commands. Pass-through packets with multicast addresses can still be accepted  
1935 depending on multicast address filter support provided by the Set MAC Address command. Packets with  
1936 destination addresses matching multicast filter entries that are set to enabled in the Set MAC Address  
1937 command are accepted; all others are rejected.

1938 Table 76 illustrates the packet format of the Disable Global Multicast Filter command.

1939 **Table 76 – Disable Global Multicast Filter Command Packet Format**

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

#### 1940 **8.4.40 Disable Global Multicast Filter Response (0x93)**

1941 In the absence of any errors, the channel shall process and respond to the Disable Global Multicast Filter  
1942 command by sending the response packet shown in Table 77.

1943 Currently no command-specific reason code is identified for this response.

1944 **Table 77 – Disable Global Multicast Filter 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	Checksum			
24..45	Pad			

#### 1945 **8.4.41 Set NC-SI Flow Control Command (0x14)**

1946 The Set NC-SI Flow Control command allows the Management Controller to configure [IEEE 802.3](#) pause  
1947 packet flow control on the NC-SI.

1948 The Set NC-SI Flow Control command is addressed to the package, rather than to a particular channel  
1949 (that is, the command is sent with a Channel ID where the Package ID subfield matches the ID of the  
1950 intended package and the Internal Channel ID subfield is set to 0x1F).

1951 When enabled for flow control, a channel may direct the package to generate and renew 802.3x (XOFF)  
1952 PAUSE Frames for a maximum interval of T12 for a single congestion condition. If the congestion  
1953 condition remains in place after a second T12 interval expires, the congested channel shall enter the  
1954 Initial State and remove its XOFF request to the package. Note that some implementations may have  
1955 shared buffering arrangements where all channels within the package become congested simultaneously.  
1956 Also note that if channels become congested independently, the package may not immediately go into  
1957 the XON state after T12 if other channels within the package are still requesting XOFF.

1958 Table 78 illustrates the packet format of the Set NC-SI Flow Control command.

1959 **Table 78 – Set NC-SI Flow Control Command Packet Format**

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

1960 Table 79 describes the values for the Flow Control Enable field.

1961 **Table 79 – 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

1962 **8.4.42 Set NC-SI Flow Control Response (0x94)**

1963 The package shall, in the absence of a checksum error or identifier mismatch, always accept the Set  
1964 NC-SI Flow Control command and send a response (see Table 80).

1965 **Table 80 – Set NC-SI Flow Control 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	Checksum				
24..45	Pad				



1966 Table 81 describes the reason code that is specific to the Set NC-SI Flow Control command.

1967 **Table 81 – Set NC-SI Flow Control Command-Specific Reason Code**

Value	Description	Comment
0x9	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

1968 **8.4.43 Get Version ID Command (0x15)**

1969 The Get Version ID command may be used by the Management Controller to request the channel to  
 1970 provide the controller and firmware type and version strings listed in the response payload description.

1971 Table 82 illustrates the packet format of the Get Version ID command.

1972 **Table 82 – Get Version ID Command Packet Format**

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

1973 **8.4.44 Get Version ID Response (0x95)**

1974 The channel shall, in the absence of an error, always accept the Get Version ID command and send the  
 1975 response packet shown in Table 83. Currently no command-specific reason code is identified for this  
 1976 response.

1977 **Table 83 – Get Version ID Response Packet Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..15	NC-SI 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)

Bytes	Bits			
	31..24	23..16	15..08	07..00
44..47	PCI DID		PCI VID	
48..51	PCI SSID		PCI SVID	
52..55	Manufacturer ID (IANA)			
56..59	Checksum			

#### 1978 8.4.44.1 NC-SI Version Encoding

1979 The NC-SI Version field holds the version number of the NC-SI specification with which the controller is  
1980 compatible. The version field shall be encoded as follows:

- 1981 • The 'major', 'minor', and 'update' bytes are BCD-encoded, and each byte holds two BCD digits.
- 1982 • The 'alpha' byte holds an optional alphanumeric character extension that is encoded using the  
1983 ISO/IEC 8859-1 Character Set.
- 1984 • The semantics of these fields follow the semantics specified in [DSP4004](#).
- 1985 • The value 0x00 in the Alpha1 or Alpha2 fields means that the corresponding alpha field is not  
1986 used. The Alpha1 field shall be used first.
- 1987 • The value 0xF in the most-significant nibble of a BCD-encoded value indicates that the most-  
1988 significant nibble should be ignored and the overall field treated as a single digit value.
- 1989 • A value of 0xFF in the update field indicates that the entire field is not present. 0xFF is not  
1990 allowed as a value for the major or minor fields.

1991 EXAMPLE: Version 3.7.10a → 0xF3F7104100  
1992 Version 10.01.7 → 0x1001F70000  
1993 Version 3.1 → 0xF3F1FF0000  
1994 Version 1.0a → 0xF1F0FF4100  
1995 Version 1.0ab → 0xF1F0FF4142 (Alpha1 = 0x41, Alpha2 = 0x42)

#### 1996 8.4.44.2 Firmware Name Encoding

1997 The Firmware Name String shall be encoded using the ISO/IEC 8859-1 Character Set. Strings are left-  
1998 justified where the leftmost character of the string occupies the most-significant byte position of the  
1999 Firmware Name String field, and characters are populated starting from that byte position. The string is  
2000 null terminated if the string is smaller than the field size. That is, the delimiter value, 0x00, follows the last  
2001 character of the string if the string occupies fewer bytes than the size of the field allows. A delimiter is not  
2002 required if the string occupies the full size of the field. Bytes following the delimiter (if any) should be  
2003 ignored and can be any value.

#### 2004 8.4.44.3 Firmware Version Encoding

2005 To facilitate a common way of representing and displaying firmware version numbers across different  
2006 vendors, each byte is hexadecimal encoded where each byte in the field holds two hexadecimal digits.  
2007 The Firmware Version field shall be encoded as follows. The bytes are collected into a single 32-bit field  
2008 where each byte represents a different 'point number' of the overall version. The selection of values that  
2009 represent a particular version of firmware is specific to the Network Controller vendor.

2010 Software displaying these numbers should not suppress leading zeros, which should help avoid user  
 2011 confusion in interpreting the numbers. For example, consider the two values 0x05 and 0x31.  
 2012 Numerically, the byte 0x31 is greater than 0x05, but if leading zeros were suppressed, the two displayed  
 2013 values would be "0.5" and "0.31", respectively, and a user would generally interpret 0.5 as representing a  
 2014 greater value than 0.31. Similarly, if leading zeros were suppressed, the value 0x01 and 0x10 would be  
 2015 displayed as 0.1 and 0.10, which could potentially be misinterpreted as representing the same version.

2016 EXAMPLE: 0x00030217 → Version 00.03.02.17  
 2017 0x010100A0 → Version 01.01.00.A0

2018 **8.4.44.4 PCI ID Fields**

2019 These fields (PCI DID, PCI VID, PCI SSID, PCI SVID) hold the PCI ID information for the Network  
 2020 Controller when the Network Controller incorporates a PCI or PCI Express™ interface that provides a  
 2021 host network interface connection that is shared with the NC-SI connection to the network.

2022 If this field is not used, the values shall all be set to zeros (0000h). Otherwise, the fields shall hold the  
 2023 PCI ID information for the host interface as defined by the version of the PCI/PCI Express™ specification  
 2024 to which the device’s interface was designed.

2025 **8.4.44.5 Manufacturer ID (IANA) Field**

2026 The Manufacturer ID holds the [IANA Enterprise Number](#) for the manufacturer of the Network Controller as  
 2027 a 32-bit binary number. If the field is unused, the value shall be set to 0xFFFFFFFF.

2028 **8.4.45 Get Capabilities Command (0x16)**

2029 The Get Capabilities command is used to discover additional optional functions supported by the channel,  
 2030 such as the number of unicast/multicast addresses supported, the amount of buffering in bytes available  
 2031 for packets bound for the Management Controller, and so on.

2032 Table 84 illustrates the packet format for the Get Capabilities command.

2033 **Table 84 – Get Capabilities Command Packet Format**

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

2034 **8.4.46 Get Capabilities Response (0x96)**

2035 In the absence of any errors, the channel shall process and respond to the Get Capabilities Command  
 2036 and send the response packet shown in Table 85. Currently no command-specific reason code is  
 2037 identified for this response.

2038

**Table 85 – Get Capabilities Response Packet Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..15	NC-SI 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			

2039 **8.4.46.1 Capabilities Flags Field**

2040 The Capabilities Flags field indicates which optional features of this specification the channel supports, as  
 2041 described in Table 86.

2042 **Table 86 – Capabilities Flags Bit Definitions**

Bit Position	Field Description	Value Description
0	Hardware Arbitration	0b = Hardware arbitration is not supported by the package. 1b = Hardware arbitration is supported by the package.
1	OS Presence	0b = OS Presence status is not supported. 1b = OS Presence status is supported.
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.
5..31	Reserved	Reserved

**2043 8.4.46.2 Broadcast Packet Filter Capabilities Field**

2044 The Broadcast Packet Filter Capabilities field defines the optional broadcast packet filtering capabilities  
2045 that the channel supports. The bit definitions for this field correspond directly with the bit definitions for the  
2046 Broadcast Packet Filter Settings field defined for the Enable Broadcast Filter command in Table 69. A bit  
2047 set to 1 indicates that the channel supports the filter associated with that bit position; otherwise, the  
2048 channel does not support that filter.

**2049 8.4.46.3 Multicast Packet Filter Capabilities Field**

2050 The Multicast Packet Filter Capabilities field defines the optional multicast packet filtering capabilities that  
2051 the channel supports. The bit definitions for this field correspond directly with the bit definitions for the  
2052 Multicast Packet Filter Settings field defined for the Enable Global Multicast Filter command in Table 74.  
2053 A bit set to 1 indicates that the channel supports the filter associated with that bit position; otherwise, the  
2054 channel does not support that filter.

**2055 8.4.46.4 Buffering Capability Field**

2056 The Buffering Capability field defines the amount of buffering in bytes that the channel provides for  
2057 inbound packets destined for the Management Controller. The Management Controller may make use of  
2058 this value in software-based Device Selection implementations to determine the relative time for which a  
2059 specific channel may be disabled before it is likely to start dropping packets. A value of 0 indicates that  
2060 the amount of buffering is unspecified.

**2061 8.4.46.5 AEN Control Support Field**

2062 The AEN Control Support field indicates various standard AENs supported by the implementation. The  
2063 format of the field is shown in Table 38.

**2064 8.4.46.6 VLAN Filter Count Field**

2065 The VLAN Filter Count field indicates the number of VLAN filters, up to 15, that the channel supports, as  
2066 defined by the Set VLAN Filter command.

**2067 8.4.46.7 Mixed, Multicast, and Unicast Filter Count Fields**

2068 The Mixed Filter Count field indicates the number of mixed address filters that the channel supports. A  
2069 mixed address filter can be used to filter on specific unicast or multicast MAC addresses.

2070 The Multicast Filter Count field indicates the number of multicast MAC address filters that the channel  
2071 supports.

2072 The Unicast Filter Count field indicates the number of unicast MAC address filters that the channel  
2073 supports.

2074 The channel is required to support at least one unicast or mixed filter, such that at least one unicast MAC  
2075 address can be configured on the interface. The total number of unicast, multicast, and mixed filters shall  
2076 not exceed 8.

**2077 8.4.46.8 VLAN Mode Support Field**

2078 The VLAN Mode Support field indicates various modes supported by the implementation. The format of  
2079 field is defined in Table 87.

2080

**Table 87 – 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

**2081 8.4.46.9 Channel Count Field**

2082 The Channel Count field indicates the number of channels supported by the Network Controller.

**2083 8.4.47 Get Parameters Command (0x17)**

2084 The Get Parameters command can be used by the Management Controller to request that the channel  
2085 send the Management Controller a copy of all of the currently stored parameter settings that have been  
2086 put into effect by the Management Controller, plus "other" Host/Channel parameter values that may be  
2087 added to the Get Parameters Response Payload.

2088 Table 88 illustrates the packet format for the Get Parameters command.

2089

**Table 88 – Get Parameters Command Packet Format**

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

**2090 8.4.48 Get Parameters Response (0x97)**

2091 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get  
2092 Parameters command and send a response. As shown in Table 89, each parameter shall return the value  
2093 that was set by the Management Controller. If the parameter is not supported, 0 is returned. Currently no  
2094 command-specific reason code is identified for this response.

2095 The payload length of this response packet will vary according to how many MAC address filters or VLAN  
2096 filters the channel supports. All supported MAC addresses are returned at the end of the packet, without  
2097 any intervening padding between MAC addresses.

2098 MAC addresses are returned in the following order: unicast filtered addresses first, followed by multicast  
2099 filtered addresses, followed by mixed filtered addresses, with the number of each corresponding to those  
2100 reported through the Get Capabilities command. For example, if the interface reports four unicast filters,  
2101 two multicast filters, and two mixed filters, then MAC addresses 1 through 4 are those currently

2102 configured through the interface’s unicast filters, MAC addresses 5 and 6 are those configured through  
 2103 the multicast filters, and 7 and 8 are those configured through the mixed filters. Similarly, if the interface  
 2104 reports two unicast filters, no multicast filters, and six mixed filters, then MAC addresses 1 and 2 are  
 2105 those currently configured through the unicast filters, and 3 through 8 are those configured through the  
 2106 mixed filters.

2107 **Table 89 – Get Parameters Response Packet Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..15	NC-SI 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 <sup>a</sup>	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			

<sup>a</sup> Variable fields can start at this byte offset.

2108 Table 90 lists the parameters for which values are returned in this response packet.

2109 **Table 90 – 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 91.
VLAN Tag Count	The number of VLAN Tags supported by the channel

Parameter Field Name	Description
VLAN Tag Flags	The enable/disable state for each supported VLAN Tag See Table 92.
Link Settings	The 32-bit Link Settings value as defined in the Set Link command
Broadcast Packet Filter Settings	The current 32-bit Broadcast Packet Filter Settings value
Configuration Flags	See Table 93.
VLAN Mode	See Table 58.
Flow Control Enable	See Table 79.
AEN Control	See Table 38.
MAC Address 1..24	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.	

2110 The format of the MAC Address Flags field is defined in Table 91.

2111 **Table 91 – 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

2112 The format of the VLAN Tag Flags field is defined in Table 92.

2113 **Table 92 – 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



2114 The format of the Configuration Flags field is defined in Table 93.

2115 **Table 93 – 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

2116 **8.4.49 Get Controller Packet Statistics Command (0x18)**

2117 The Get Controller Packet Statistics command may be used by the Management Controller to request a  
 2118 copy of the aggregated packet statistics that the channel maintains for its external interface to the LAN  
 2119 network. The statistics are an aggregation of statistics for both the host side traffic and the NC-SI Pass-  
 2120 through traffic.

2121 **Table 94 – Get Controller Packet Statistics Command Packet Format**

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

2122 **8.4.50 Get Controller Packet Statistics Response (0x98)**

2123 The channel shall, in the absence of a checksum error or identifier mismatch, always accept the Get  
2124 Controller Packet Statistics command and send the response packet shown in Table 95.

2125 The Get Controller Packet Statistics Response frame contains a set of statistics counters that monitor the  
2126 LAN traffic in the Network Controller. Implementation of the counters listed in Table 96 is optional. The  
2127 Network Controller shall return any unsupported counter with a value of 0xFFFFFFFF.

2128 **Table 95 – Get Controller Packet Statistics Response Packet Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..15	NC-SI 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			
88..91	FCS Receive Errors			
92..95	Alignment Errors			
96..99	False Carrier Detections			
100..103	Runt Packets Received			
104..107	Jabber Packets Received			
108..111	Pause XON Frames Received			
112..115	Pause XOFF Frames Received			
116..119	Pause XON Frames Transmitted			
120..123	Pause XOFF Frames Transmitted			
124..127	Single Collision Transmit Frames			
128..131	Multiple Collision Transmit Frames			
132..135	Late Collision Frames			
136..139	Excessive Collision Frames			
140..143	Control Frames Received			
144..147	64-Byte Frames Received			
148..151	65–127 Byte Frames Received			

Bytes	Bits			
	31..24	23..16	15..08	07..00
152..155	128–255 Byte Frames Received			
156..159	256–511 Byte Frames Received			
160..163	512–1023 Byte Frames Received			
164..167	1024–1522 Byte Frames Received			
168..171	1523–9022 Byte Frames Received			
172..175	64-Byte Frames Transmitted			
176..179	65–127 Byte Frames Transmitted			
180..183	128–255 Byte Frames Transmitted			
184..187	256–511 Byte Frames Transmitted			
188..191	512–1023 Byte Frames Transmitted			
192..195	1024–1522 Byte Frames Transmitted			
196..199	1523–9022 Byte Frames Transmitted			
200..207	Valid Bytes Received			
208..211	Error Runt Packets Received			
212..215	Error Jabber Packets Received			
216..219	Checksum			

2129

**Table 96 – Get Controller Packet Statistics Counter Numbers**

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

Counter Number	Name	Meaning
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
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 bad 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 <a href="#">IEEE 802.3</a> )
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 <i>not</i> 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

Counter Number	Name	Meaning
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
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 <i>and</i> have a bad FCS value

2130 The Network Controller shall also indicate in the Counters Cleared from Last Read fields whether the  
 2131 corresponding field has been cleared (possibly by the host) since it was last read by means of the NC-SI.  
 2132 The Counters Cleared from Last Read fields have the format shown in Table 97.

2133 Currently no command-specific reason code is identified for this response.

2134

**Table 97 – 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

2135

**8.4.51 Get NC-SI Statistics Command (0x19)**

2136 In addition to the packet statistics accumulated on the LAN network interface, the channel separately  
 2137 accumulates a variety of NC-SI specific packet statistics for the channel. The Get NC-SI Statistics  
 2138 command may be used by the Management Controller to request that the channel send a copy of all  
 2139 current NC-SI packet statistic values for the channel. The implementation may or may not include  
 2140 statistics for commands that are directed to the package.

2141 Table 98 illustrates the packet format of the Get NC-SI Statistics command.

2142

**Table 98 – Get NC-SI Statistics Command Packet Format**

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

2143

**8.4.52 Get NC-SI Statistics Response (0x99)**

2144 In the absence of any error, the channel shall process and respond to the Get NC-SI Statistics command  
 2145 by sending the response packet and payload shown in Table 99.

2146

**Table 99 – Get NC-SI Statistics Response Packet Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..15	NC-SI 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			

2147 The Get NC-SI Statistics Response frame contains a set of statistics counters that monitor the NC-SI  
 2148 traffic in the Network Controller. Counters that are supported shall be reset to 0x0 when entering into the  
 2149 Initial State and after being read. Implementation of the counters shown in Table 100 is optional. The  
 2150 Network Controller shall return any unsupported counter with a value of 0xFFFFFFFF. Counters may  
 2151 wraparound or stop if they reach 0xFFFFFFFFE.

2152 Currently no command-specific reason code is identified for this response.

2153 **Table 100 – Get NC-SI Statistics Response Counters**

Counter Number	Name	Meaning
1	NC-SI Commands Received	Counts the number of NC-SI frames received and identified as valid NC-SI commands (commands that generate a response packet)
2	NC-SI Control Packets Dropped	Counts the number of NC-SI Control packets that were received and dropped
3	NC-SI Command Type Errors	Counts the number of NC-SI commands that had a Command Unsupported response code
4	NC-SI Command Checksum Errors	Counts the number of NC-SI commands that had a checksum invalid error (if checksum is supported by the channel)
5	NC-SI Receive Packets	Counts the total number of NC-SI Control packets received
6	NC-SI Transmit Packets	Counts the total number of NC-SI Control packets transmitted to the Management Controller
7	AENs Sent	Counts the total number of AEN packets transmitted to the Management Controller

2154 **8.4.53 Get NC-SI Pass-through Statistics Command (0x1A)**

2155 The Get NC-SI Pass-through Statistics command may be used by the Management Controller to request  
 2156 that the channel send a copy of all current NC-SI Pass-through packet statistic values.

2157 Table 101 illustrates the packet format of the Get NC-SI Pass-through Statistics command.

2158 **Table 101 – Get NC-SI Pass-through Statistics Command Packet Format**

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

2159 **8.4.54 Get NC-SI Pass-through Statistics Response (0x9A)**

2160 In the absence of any error, the channel shall process and respond to the Get NC-SI Pass-through  
2161 Statistics command by sending the response packet and payload shown in Table 102.

2162 **Table 102 – Get NC-SI Pass-through Statistics Response Packet Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..15	NC-SI 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			

2163 The Get NC-SI Statistics Response frame contains a set of statistics counters that monitor the NC-SI  
2164 Pass-through traffic in the Network Controller. Supported counters shall be reset to 0x0 when entering  
2165 into the Initial State and after being read. Implementation of the counters shown in Table 103 is optional.  
2166 The Network Controller shall return any unsupported counter with a value of 0xFFFFFFFF. Counters may  
2167 wraparound or stop if they reach 0xFFFFFFFFE.

2168 **Table 103 – Get NC-SI Statistics Response**

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)



Counter Number	Name	Meaning
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
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)

2169 Currently no command-specific reason code is identified for this response.

2170 **8.4.55 OEM Command (0x50)**

2171 The OEM command may be used by the Management Controller to request that the channel provide  
 2172 vendor-specific information. The [Vendor Enterprise Number](#) is the unique MIB/SNMP Private Enterprise  
 2173 number assigned by IANA per organization. Vendors are free to define their own internal data structures  
 2174 in the vendor data fields.

2175 Table 104 illustrates the packet format of the OEM command.

2176 **Table 104 – OEM Command Packet Format**

Bytes	Bits			
	31..24	23..16	15..08	07..00
00..15	NC-SI 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.			

2177 **8.4.56 OEM Response (0xD0)**

2178 The channel shall return the “Unknown Command Type” reason code for any unrecognized enterprise  
 2179 number, using the packet format shown in Table 105. If the command is valid, the response, if any, is  
 2180 allowed to be vendor-specific. The 0x8000 range is recommended for vendor-specific code.

2181 Currently no command-specific reason code is identified for this response.

2182 **Table 105 – OEM 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	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.				

2183 **8.5 AEN Packet Formats**

2184 This clause defines the formats for the different types of AEN packets. For a list of the AEN types, see  
 2185 Table 16.

2186 **8.5.1 Link Status Change AEN**

2187 The Link Status Change AEN indicates to the Management Controller any changes in the channel's  
 2188 external interface link status.

2189 This AEN should be sent if any change occurred in the link status (that is, the actual link mode was  
 2190 changed). The Link Status and OEM Link Status fields reproduce the bit definitions defined in the Get  
 2191 Link Status Response Packet (see Table 47).

2192 Table 106 illustrates the packet format of the Link Status Change AEN.

2193 **Table 106 – 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				

2194 **8.5.2 Configuration Required AEN**

2195 The Configuration Required AEN indicates to the Management Controller that the channel is transitioning  
 2196 into the Initial State. (This AEN is not sent if the channel enters the Initial State because of a Reset  
 2197 Channel command.)

2198 NOTE: This AEN may not be generated in some situations in which the Network Controller goes into the Initial  
 2199 State. For example, some types of hardware resets may not accommodate generating the AEN.

2200 Table 107 illustrates the packet format of the Configuration Required AEN.

2201 **Table 107 – 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			

2202 **8.5.3 Host Network Controller Driver Status Change AEN**

2203 This AEN indicates a change of the Host Network Controller Driver Status. Table 108 illustrates the  
 2204 packet format of the AEN.

2205 **Table 108 – 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			

2206 The Host Network Controller Driver Status field has the format shown in Table 109.

2207 **Table 109 – 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

2208 **9 Packet-Based and Op-Code Timing**

2209 Table 110 presents the timing specifications for a variety of packet-to-electrical-buffer interactions, inter-  
 2210 packet timings, and op-code processing requirements.

2211 **Table 110 – NC-SI Packet-Based and Op-Code Timing Parameters**

Name	Symbol	Value	Description
Package Deselect to Hi-Z Interval	T1	200 $\mu$ s, 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 (that is, when it enters the Initial State)  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 is allowed to not recognize or respond to commands due to an Asynchronous Reset event  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 requests due to a Synchronous Reset event  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.
NOTE: If hardware arbitration is in effect, the hardware arbitration output buffer enable/disable timing specifications take precedence.			

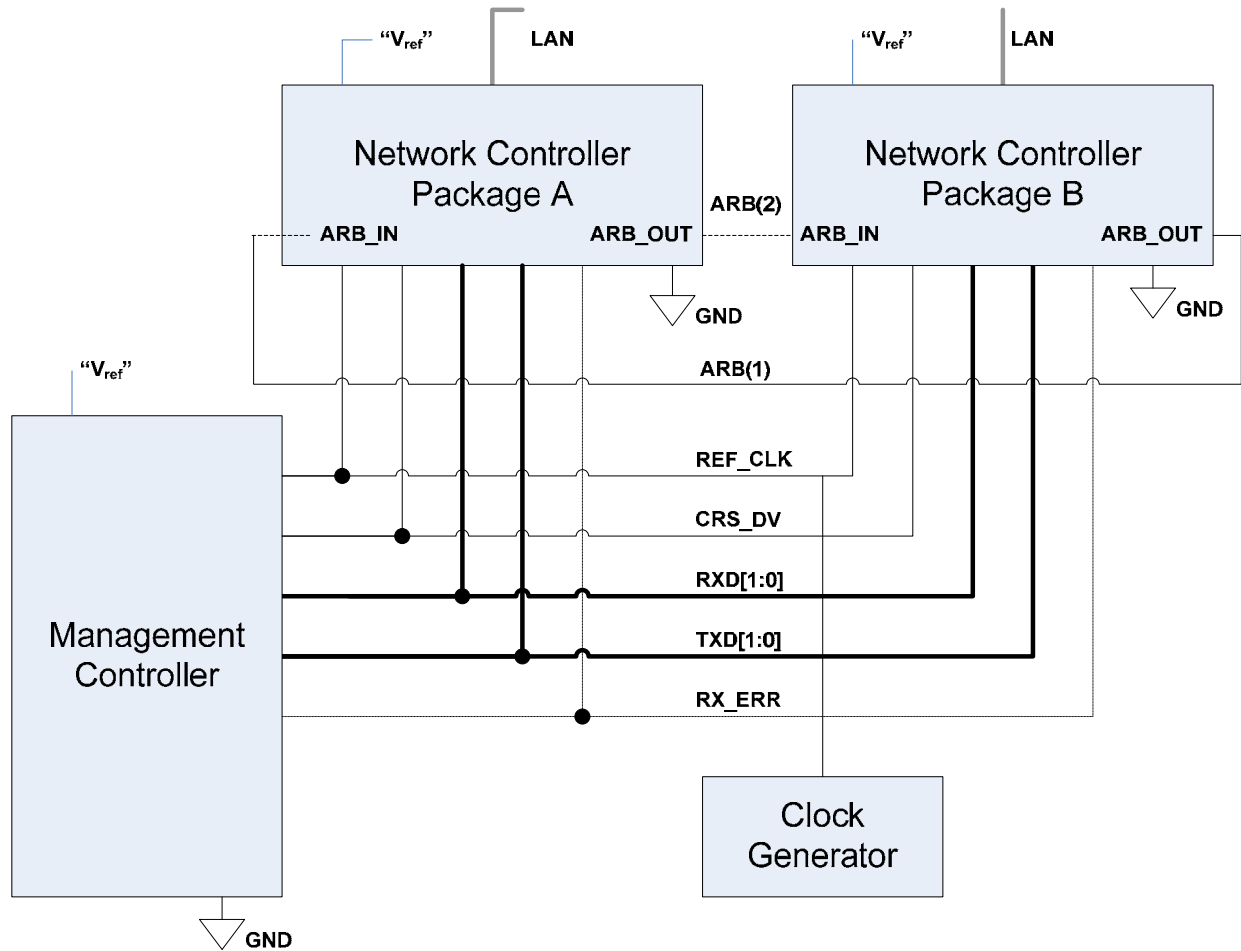
## 2212 10 Electrical Specification

2213 This clause provides background information about the NC-SI specification, describes the NC-SI  
2214 topology, and defines the electrical, timing, signal behavior, and power-up characteristics for the NC-SI  
2215 physical interface.

### 2216 10.1 Topologies

2217 The electrical specification defines the NC-SI electrical characteristics for one management processor  
2218 and one to four Network Controller packages in a bussed “multi-drop” arrangement. The actual number of  
2219 devices that can be supported may differ based on the trace characteristics and routing used to  
2220 interconnect devices in an implementation.

2221 Figure 15 shows an example topology.



2222

2223

Figure 15 – Example NC-SI Signal Interconnect Topology

2224 **10.2 Electrical and Signal Characteristics and Requirements**

2225 This clause defines the electrical, timing, signal behavior, and power-up characteristics for the NC-SI  
 2226 physical interface.

2227 **10.2.1 Companion Specifications**

2228 Implementations of the physical interface and signaling for the NC-SI shall meet the specifications in [RMII](#)  
 2229 and [IEEE 802.3](#), except where those requirements differ or are extended with specifications provided in  
 2230 this document, in which case the specifications in this document shall take precedence.

2231 **10.2.2 Full-Duplex Operation**

2232 The NC-SI is specified only for full-duplex operation. Half-duplex operation is not covered by this  
 2233 specification.

2234 **10.2.3 Signals**

2235 Table 111 lists the signals that make up the NC-SI physical interface.

2236 Unless otherwise specified, the high level of an NC-SI signal corresponds to its asserted state, and the  
 2237 low level represents the de-asserted state. For data bits, the high level represents a binary '1' and the low  
 2238 level a binary '0'.

2239 **Table 111 – Physical NC-SI Signals**

Signal Name	Direction (with respect to the Network Controller)	Direction (with respect to the Management Controller MAC)	Use	Mandatory or Optional
REF_CLK <sup>[a]</sup>	Input	Input	Clock reference for receive, transmit, and control interface	M
CRS_DV <sup>[b]</sup>	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 <sup>[c]</sup>	N/A	Network Controller hardware arbitration Input	O <sup>[c]</sup>
ARB_OUT	Output <sup>[c]</sup>	N/A	Network Controller hardware arbitration Output	O <sup>[c]</sup>

<sup>[a]</sup> 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-SI power up and remain in effect while the NC-SI is powered up.

<sup>[b]</sup> In the [RMII Specification](#), the MII Carrier Sense signal, CRS, was combined with RX\_DV to form the CRS\_DV signal. When the NC-SI 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 the NC-SI 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.

<sup>[c]</sup> 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.

2240 **10.2.4 High-Impedance Control**

2241 Shared NC-SI operation requires Network Controller devices to be able to set their NC-SI outputs  
 2242 (RXD[1:0], CRS\_DV, and, if implemented, RX\_ER) into a high-impedance state either upon receipt of a  
 2243 command received through NC-SI, or, if hardware-based arbitration is in effect, as a result of hardware-  
 2244 based arbitration.

2245 Network Controller packages shall leave their NC-SI outputs in the high-impedance state on interface  
 2246 power up and shall not drive their NC-SI outputs until selected. For additional information about Network  
 2247 Controller packages, see 8.4.5.

2248 For NC-SI output signals in this specification, unless otherwise specified, the high-impedance state is  
 2249 defined as the state in which the signal leakage meets the I<sub>z</sub> specification provided in 10.2.5.

2250 **10.2.5 DC Characteristics**

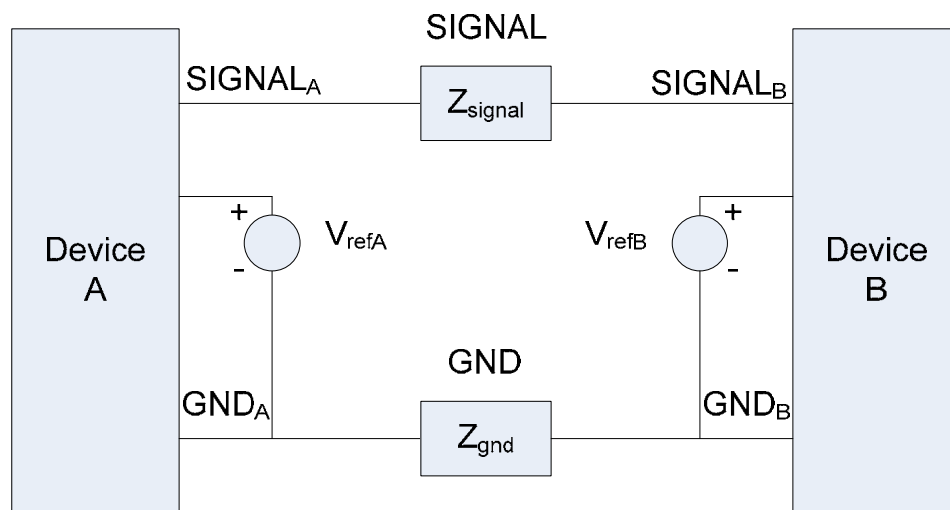
2251 This clause defines the DC characteristics of the NC-SI physical interface.

2252 **10.2.5.1 Signal Levels**

2253 CMOS 3.3 V signal levels are used for this specification.

2254 The following characteristics apply to DC signals:

- 2255 • Unless otherwise specified, DC signal levels and  $V_{\text{ref}}$  are measured relative to Ground (GND) at  
2256 the respective device providing the interface, as shown in Figure 16.
- 2257 • Input specifications refer to the signals that a device shall accept for its input signals, as  
2258 measured at the device.
- 2259 • Output specifications refer to signal specifications that a device shall emit for its output signals,  
2260 as measured at the device.



2261

2262

Figure 16 – DC Measurements



2263 Table 112 provides DC specifications.

2264 **Table 112 – 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	$\mu A$
Input low current	$I_{il}$	$V_{in} = 0 V$	-20		0	$\mu A$
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	$I_z$	$0 \leq V_{in} \leq V_{ref}$ at $V_{ref} = V_{ref,max}$	-20		20	$\mu A$

<sup>[a]</sup>  $V_{ref}$  = 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 the NC-SI 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. In order 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 the NC-SI.

## 2265 10.2.6 AC Characteristics

2266 This clause defines the AC characteristics of the NC-SI physical interface.

### 2267 10.2.6.1 Rise and Fall Time Measurement

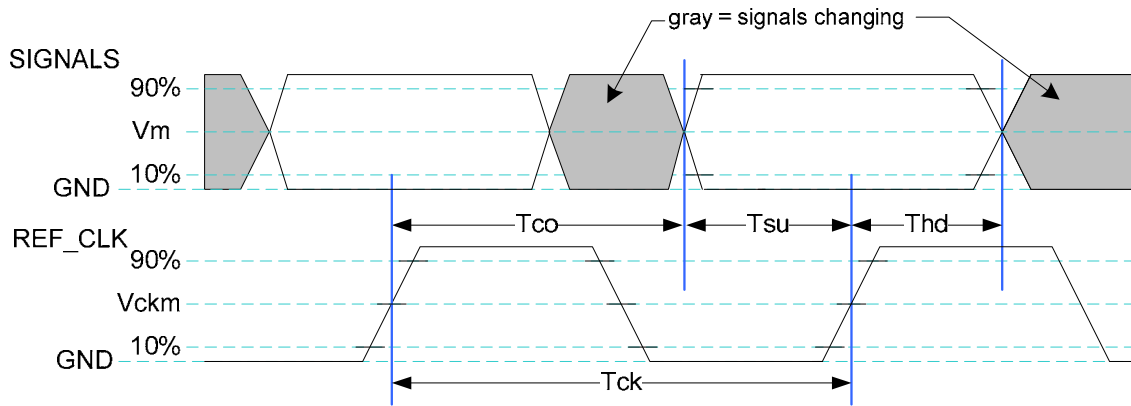
2268 Rise and fall time are measured between points that cross 10% and 90% of  $V_{ref}$  (see Table 112). The  
2269 middle points (50% of  $V_{ref}$ ) are marked as  $V_{ckm}$  and  $V_m$  for clock and data, respectively.

### 2270 10.2.6.2 REF\_CLK Measuring Points

2271 In Figure 17, REF\_CLK duty cycle measurements are made from  $V_{ckm}$  to  $V_{ckm}$ . Clock skew  $T_{skew}$  is  
2272 measured from  $V_{ckm}$  to  $V_{ckm}$  of two NC-SI devices and represents maximum clock skew between any two  
2273 devices in the system.

### 2274 10.2.6.3 Data, Control, and Status Signal Measuring Points

2275 In Figure 17, all timing measurements are made between  $V_{ckm}$  and  $V_m$ .  $T_{co}$  is measured with a capacitive  
2276 load between 10 pF and 50 pF. Propagation delay  $T_{prop}$  is measured from  $V_m$  on the transmitter to  $V_m$  on  
2277 the receiver.



2278

2279

Figure 17 – AC Measurements

2280 Table 113 provides AC specifications.

2281

Table 113 – 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 <sup>[a]</sup> (10 pF ≤ C <sub>load</sub> ≤ 50 pF)	T <sub>co</sub>	2.5		12.5	ns
Skew between clocks	T <sub>skew</sub>			1.5	ns
TXD[1:0], TX_EN, RXD[1:0], CRS_DV, RX_ER Data Setup to REF_CLK rising edge	T <sub>su</sub>	3			ns
TXD[1:0], TX_EN, RXD[1:0], CRS_DV, RX_ER data hold from REF_CLK rising edge	T <sub>hd</sub>	1			ns
Signal Rise/Fall Time	T <sub>r</sub> /T <sub>f</sub>	0.5		6	ns
REF_CLK Rise/Fall Time	T <sub>ckr</sub> /T <sub>ckf</sub>	0.5		3.5	ns
Interface Power-Up High-Impedance Interval	T <sub>pwrtz</sub>	2			μs
Power Up Transient Interval (recommendation)	T <sub>pwrt</sub>			100	ns
Power Up Transient Level (recommendation)	V <sub>pwrt</sub>	-200		200	mV
Interface Power-Up Output Enable Interval	T <sub>pwre</sub>			10	ms
EXT_CLK Startup Interval	T <sub>clkstrt</sub>			100	ms

<sup>[a]</sup> This timing relates to the output pins, while T<sub>su</sub> and T<sub>hd</sub> relate to timing at the input pins.

2282 **10.2.6.4 Timing Calculation (Informative)**

2283 This clause presents the relationships between the timing parameters and how they are used to calculate  
 2284 setup and hold time margins.

2285 **10.2.6.4.1 Setup Calculation**

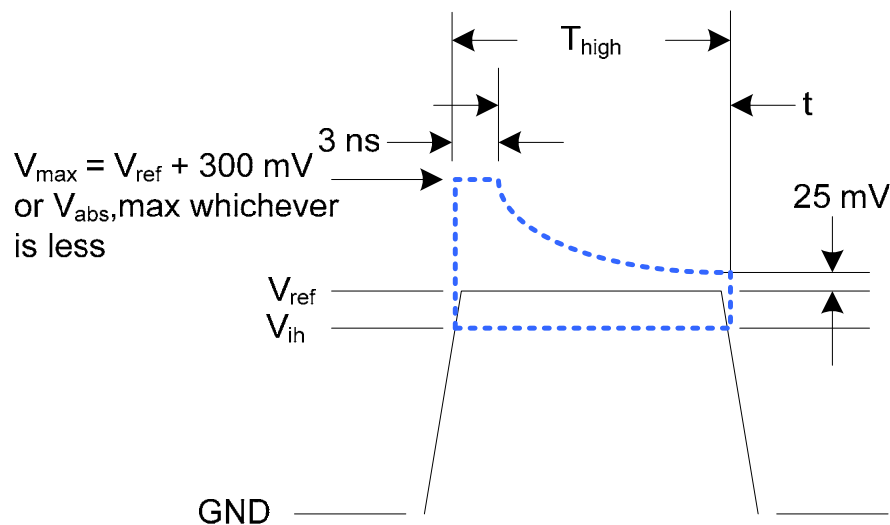
$$2286 \quad T_{su} \leq T_{clk} - (T_{skew} + T_{co} + T_{prop})$$

2287 **10.2.6.4.2 Hold Calculation**

$$2288 \quad T_{hd} \leq T_{co} - T_{skew} + T_{prop}$$

2289 **10.2.6.5 Overshoot Specification**

2290 Devices shall accept signal overshoot within the ranges specified in Figure 18, measured at the device,  
 2291 without malfunctioning.



2292

2293 **Figure 18 – Overshoot Measurement**

2294 The signal is allowed to overshoot up to the specified  $V_{max}$  for the first 3 ns following the transition above  
 2295  $V_{ih}$ . Following that interval is an exponential decay envelope equal to the following:

$$2296 \quad V_{ref} + V_{os} * e^{[-K * ([t - 3 \text{ ns}] / T_d)]}$$

2297 Where, for  $t = 3$  to 10 ns:

2298  $t = 0$  corresponds to the leading crossing of  $V_{ih}$ , going high.

2299  $V_{ref}$  is the bus high reference voltage (see 10.2.5).

2300  $V_{abs,max}$  is the maximum allowed signal voltage level (see 10.2.5).

$$2301 \quad V_{os} = V_{max} - V_{ref}$$

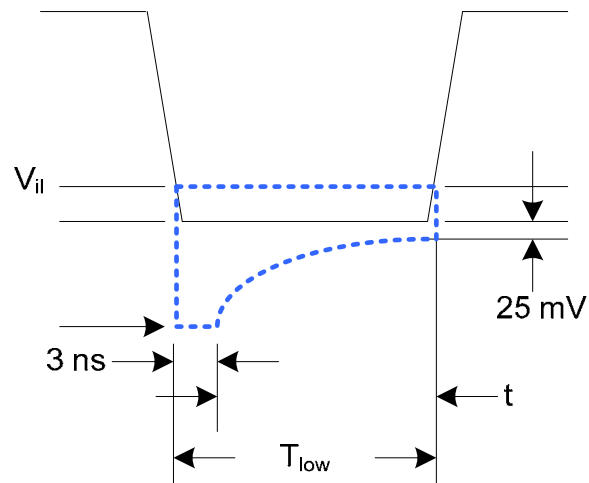
$$2302 \quad K = I_n(25 \text{ mV} / V_{os})$$

$$2303 \quad T_d = 7 \text{ ns}$$

2304 For  $t > 10$  ns, the  $V_{ref} + 25 \text{ mV}$  limit holds flat until the conclusion of  $T_{high}$ .

2305 **10.2.6.6 Undershoot Specification**

2306 Devices are required to accept signal undershoot within the ranges specified in Figure 19, measured at  
 2307 the device, without malfunctioning.



2308

2309

**Figure 19 – Undershoot Measurement**

2310 The signal is allowed to undershoot up to the specified  $V_{abs,min}$  for the first 3 ns following the transition  
 2311 above  $V_{ii}$ . Following that interval is an exponential envelope equal to the following:

$$2312 \quad * ([t - 3 \text{ ns}] / T_d)$$

2313 Where, for  $t = 3$  to 10 ns:

2314  $t = 0$  corresponds to the leading crossing of  $V_{ii}$ , going low.

2315  $V_{abs,min}$  is the minimum allowed signal voltage level (see 10.2.5).

$$2316 \quad K = I_n(25 \text{ mV} / V_{os})$$

$$2317 \quad T_d = 7 \text{ ns}$$

2318 For  $t > 7$  ns, the GND – 25 mV limit holds flat until the conclusion of  $T_{low}$ .

2319 **10.2.7 Interface Power-Up**

2320 To prevent signals from back-powering unpowered devices, it is necessary to specify a time interval  
 2321 during which signals are not to be driven until devices sharing the interface have had time to power up.  
 2322 To facilitate system implementation, the start of this interval shall be synchronized by an external signal  
 2323 across devices.

2324 **10.2.7.1 Power-Up Control Mechanisms**

2325 The device that provides the interface shall provide one or more of the following mechanisms to enable  
 2326 the system integrator to synchronize interface power-up among devices on the interface:

- 2327 • **Device Power Supply Pin**

2328 The device has a power supply pin that the system integrator can use to control power-up of the  
 2329 interface. The device shall hold its outputs in a high-impedance state (current  $< I_z$ ) for at least

2330  $T_{\text{pwrz}}$  seconds after the power supply has initially reached its operating level (where the power  
2331 supply operating level is specified by the device manufacturer).

2332 • **Device Reset Pin or Other Similar Signal**

2333 The device has a reset pin or other signal that the system integrator can use to control the  
2334 power-up of the interface. This signal shall be able to be driven asserted during interface power-  
2335 up and de-asserted afterward. The device shall hold its outputs in a high-impedance state  
2336 (current  $< I_z$ ) for at least  $T_{\text{pwrz}}$  seconds after the signal has been de-asserted, other than as  
2337 described in 10.2.7.2. It is highly recommended that a single signal be used; however, an  
2338 implementation is allowed to use a combination of signals if required. Logic levels for the signals  
2339 are as specified by the device manufacturer.

2340 • **REF\_CLK Detection**

2341 The device can elect to detect the presence of an active REF\_CLK and use that for determining  
2342 whether NC-SI power up has occurred. It is recommended that the device should count at least  
2343 100 clocks and continue to hold its outputs in a high-impedance state (current  $< I_z$ ) for at least  
2344  $T_{\text{pwrz}}$  seconds more (Informational: 100 clocks at 50 MHz is 2 us).

2345 **10.2.7.2 Power-Up Transients**

2346 It is possible that a device may briefly drive its outputs while the interface or device is first receiving  
2347 power, due to ramping of the power supply and design of its I/O buffers. It is recommended that devices  
2348 be designed so that such transients, if present, are less than  $V_{\text{pwrt}}$  and last for no more than  $T_{\text{pwrt}}$ .

2349 **10.2.8 REF\_CLK Startup**

2350 REF\_CLK shall start up, run, and meet all associated AC and DC specifications within  $T_{\text{clkstrt}}$  seconds of  
2351 interface power up.

## ANNEX A (normative)

### Extending the Model

2352  
2353  
2354  
2355

2356 This annex explains how the model can be extended to include vendor-specific content.

#### 2357 **A.1 Commands Extension**

2358 A Network Controller vendor may implement extensions and expose them using the OEM command, as  
2359 described in 8.4.55.

#### 2360 **A.2 Design Considerations**

2361 This clause describes certain design considerations for vendors of Management Controllers.

##### 2362 **A.2.1 PHY Support**

2363 Although not a requirement of this specification, a Management Controller vendor may want to consider  
2364 designing an NC-SI in such a manner that it could also be configured for use with a conventional RMII  
2365 PHY. This would enable the vendor's controller to also be used in applications where a direct, non-shared  
2366 network connection is available or preferred for manageability.

##### 2367 **A.2.2 Multiple Management Controllers Support**

2368 Currently, there is no requirement for Management Controllers to be able to put their TXD output lines  
2369 and other output lines into a high-impedance state, because the present definition assumes only one  
2370 Management Controller on the bus. However, component vendors may want to consider providing such  
2371 control capabilities in their devices to support possible future system topologies where more than one  
2372 Management Controller shares the bus to enable functions such as Management Controller fail-over or to  
2373 enable topologies where more than one Management Controller can do NC-SI communications on the  
2374 bus. If a vendor elects to make such provision, it is recommended that the TXD line and the remaining  
2375 output lines be independently and dynamically switched between a high-impedance state and re-enabled  
2376 under firmware control.

2377

## ANNEX B (informative)

### Relationship to RMI Specification

#### 2382 B.1 Differences with the *RMI Specification*

2383 The following list presents key differences and clarifications between the *NC-SI Specification* and  
2384 sections in the [RMI Specification](#). (Section numbers refer to the [RMI Specification](#).)

- 2385 • General: Where specifications from [IEEE 802.3](#) apply, this specification uses the version  
2386 specified in clause 2, rather than the earlier IEEE 802.3u version that is referenced by [RMI](#).
- 2387 • Section 1.0:
  - 2388 – The *NC-SI Specification* requires 100 Mbps support, but it does not specify a required  
2389 minimum. (10 Mbps support is not required by NC-SI.)
  - 2390 – Item 4. (Signals may or may not be considered to be TTL. NC-SI is not 5-V tolerant.)
- 2391 • Section 2.0:
  - 2392 – Comment: NC-SI chip-to-chip includes considerations for multi-drop and allows for non-  
2393 PCB implementations and connectors (that is, not strictly point-to-point).
- 2394 • Section 3.0:
  - 2395 – Note/Advisory: The NC-SI clock is provided externally. An implementation can have  
2396 REF\_CLK provided by one of the devices on the bus or by a separate device.
- 2397 • Section 5.0:
  - 2398 – For NC-SI, the term *PHY* is replaced by *Network Controller*.
- 2399 • Table 1:
  - 2400 – The information in Table 1 in the [RMI Specification](#) is superseded by tables in this  
2401 specification.
- 2402 • Section 5.1, paragraph 2:
  - 2403 – The *NC-SI Specification* allows 100 ppm. This supersedes the [RMI Specification](#), which  
2404 allows 50 ppm.
- 2405 • Section 5.1, paragraph 3:
  - 2406 – The NC-SI inherits the same requirements. The NC-SI MTU is required only to support  
2407 Ethernet MTU with VLAN, as defined in the [IEEE 802.3](#) version listed in clause 2.
- 2408 • Section 5.1 paragraph 4:
  - 2409 – The [RMI Specification](#) states: "During a false carrier event, CRS\_DV shall remain asserted  
2410 for the duration of carrier activity." This statement is not applicable to full-duplex operation  
2411 of the NC-SI. CRS\_DV from the Network Controller is used only as a data valid (DV)  
2412 signal. Because the Carrier Sense aspect of CRS\_DV is not used for full-duplex operation  
2413 of the NC-SI, the Network Controller would not generate false carrier events for the NC-SI.  
2414 However, it is recommended that the MAC in the Management Controller be able to  
2415 correctly detect and handle these patterns if they occur, as this would be part of enabling  
2416 the Management Controller MAC to also be able to work with an RMI PHY.

- 2417 • Section 5.2:
  - 2418 – The NC-SI does not specify a 10 Mbps mode. The Carrier Sense aspect of CRS\_DV is not
  - 2419 used for full-duplex operation of NC-SI.
- 2420 • Section 5.3.1:
  - 2421 – While the NC-SI does not specify Carrier Sense usage of CRS\_DV, it is recommended that
  - 2422 a Management Controller allow for CRS\_DV toggling, in which CRS\_DV toggles at 1/2
  - 2423 clock frequency, and that Management Controller MACs tolerate this and realign bit
  - 2424 boundaries correctly in order to be able to work with an RMII PHY also.
- 2425 • Section 5.3.2:
  - 2426 – There is no 10 Mbps mode specified for the NC-SI.
- 2427 • Section 5.3.3:
  - 2428 – Generally there is no expectation that the Network Controller will generate these error
  - 2429 conditions for the NC-SI; however, the MAC in the Management Controller should be able
  - 2430 to correctly detect and handle these patterns if they occur.
- 2431 • Section 5.3.3:
  - 2432 – The NC-SI does not specify or require support for RMII Registers.
- 2433 • Section 5.5.2:
  - 2434 – Ignore (N/A) text regarding 10 Mbps mode. The NC-SI does not specify or require interface
  - 2435 operation in 10 Mbps mode.
- 2436 • Section 5.6:
  - 2437 – The Network Controller will not generate collision patterns for the specified full-duplex
  - 2438 operation of the NC-SI; however, the MAC in the Management Controller should be able to
  - 2439 detect and handle these patterns if they occur in order to be able to work with an RMII PHY
  - 2440 also.
- 2441 • Section 5.7:
  - 2442 – NC-SI uses the [IEEE 802.3](#) version listed in clause 2 instead of 802.3u as a reference.
- 2443 • Section 5.8:
  - 2444 – Loopback operation is not specified for the NC-SI.
- 2445 • Section 7.0:
  - 2446 – The NC-SI electrical specifications (clause 10) take precedence. (For example, section
  - 2447 7.4.1 in the [RMII Specification](#) for capacitance is superseded by *NC-SI Specification* 25 pF
  - 2448 and 50 pF target specifications.)
- 2449 • Section 8.0:
  - 2450 – 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	Editor	Description
1.0.0	07/21/2009		DMTF Standard Release

2456

## Bibliography

- 2457 IANA, Internet Assigned Numbers Authority ([www.iana.org](http://www.iana.org)). A body that manages and organizes  
2458 numbers associated with various Internet protocols.
- 2459 DMTF [DSP4004](#), *DMTF Release Process*, January 2007,  
2460 [http://www.dmtf.org/standards/published\\_documents/DSP4004\\_2.1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP4004_2.1.0.pdf)