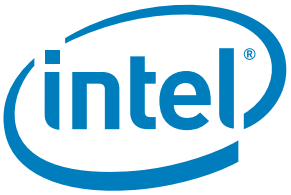


Intel® 82575EB Gigabit Ethernet Controller Manageability

LAN Access Division

324631-003
Revision: 2.11
January 2011

Revisions

Date	Revision	Description
4/2006	0.26	
2/2006	0.25	
6/2006	0.5	
7/2007	1.0	
10/2007	1.1	• Added Section 4.5.2.1.
12/3/2010	2.0	• Format updated.
12/13/2010	2.1	• Supported feature set updated. ASF references removed (not supported). • Structure updated. SMBus section elevated.
1/28/2011	2.11	• Updated product references to match brand string.

Contents

1.0 Introduction

Network management is an increasingly important requirement in today's networked computer environment. Software-based management applications provide the ability to administer systems while the operating system is functioning in a normal power state (not in a pre-boot state or powered-down state). The Intel® System Management Bus (SMBus) Interface and the Network Controller - Sideband Interface (NC-SI) for the 82575EB fills the management void that exists when the operating system is not running or fully functional.

This is accomplished by providing a mechanism by which manageability network traffic can be routed to and from a management controller. The 82575EB provides two different and mutually exclusive bus interfaces for manageability traffic. The first is the Intel® proprietary SMBus interface; several generations of Intel® Ethernet controllers have provided this same interface that operates at speeds of up to 1 MHz.

The second interface is NC-SI, which is a new industry standard interface created by the DMTF specifically for routing manageability traffic to and from a management controller. The NC-SI interface operates at 100 Mb/s full-duplex speeds.

1.1 Scope

This document describes the supported management interfaces and hardware configurations for platform system management. It describes the interfaces to an external Baseboard Management Controller (BMC), the partitioning of platform manageability among system components, and the functionality provided by the 82575EB in each of the platform configurations.

1.2 Number Conventions

Unless otherwise specified, numbers are represented as follows:

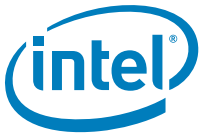
- Hexadecimal numbers are identified by an "0x" suffix on the number (0x2A, 0x12).
- Binary numbers are identified by a "b" suffix on the number (0011b). Values for SMBus transactions in diagrams are listed in binary without the "b" or in hexadecimal without the "0x".

Any other numbers without a suffix are intended as decimal numbers.

1.3 Acronyms

Following are a list of acronyms that are used throughout this document.

Acronym	Definition
ACK	Acknowledge.
ARA	SMBus Alert Response Address.
ARP	Address Resolution Protocol.
BMC	Baseboard Manageability Controller. The general name for an external TCO controller, relevant only in TCO mode.
CSR	Control and Status Register. Usually refers to a hardware register.
DHCP	Dynamic Host Configuration Protocol. A TCP/IP protocol that enables a client to receive a temporary IP address over the network from a remote server.
FW	Firmware. Also known as embedded software.
HW	Hardware.
IEEE	Institute of Electrical and Electronics Engineers.
IP	Internet Protocol. The protocol within TCP/IP that governs the breakup and reassembly of data messages into packets and the packet routing within the network.
IP Address	The 4-byte or 16-byte address that designates the Ethernet controller within the IP communication protocol. This address is dynamic and can be updated frequently during runtime.
IPMI	Intelligent Platform Management Interface Specification.
LAN	Local Area Network. Also known as the Ethernet.
MAC Address	The 6-byte address that designates Ethernet controller within the Ethernet protocol. This address is constant and unique per Ethernet controller.
NA	Not Applicable.
NACK	Not Acknowledged.
Acronym	Definition.
NC-SI	Network Controller Sideband Interface. New DMTF industry standard sideband interface.
NIC	Network Interface Card. Generic name for a Ethernet controller that resides on a Printed Circuit Board (PCB).
OS	Operating System. Usually designates the PC system's software.
PEC	The SMBus checksum signature, sent at the end of an SMBus packet. An SMBus device can be configured either to require or not require this signature.
PET	Platform Event Trap.
PT	Pass-Through. Also known as TCO mode.
PSA	SMBus Persistent Slave Address device. In the SMBus 2.0 specification, this designates an SMBus device whose address is stored in non-volatile memory.
RMCP	Remote Management and Control Protocol.
RSP	RMCP Security Extensions Protocol.
SA	Security Association.
SMBus	System Management Bus.
SNMP	Simple Network Management Protocol.
SW	Software.
TCO	Total Cost of Ownership.
TBD	To Be Defined.



1.4 Reference Documents

Document Name	Version	Owner	Location
SMBus Specification	2.0	SBS Forum	http://www.smbus.org/
I^2C Specification	2.1	Phillips Semiconductors	http://www.philipslogic.com/
NC-SI Specification	1.0	DMTF	http://www.dmtf.org/apps/org/workgroup/pos_sb/

2.0 Pass-Through (PT) Functionality

Pass-Through (PT) is the term used when referring to the process of sending and receiving Ethernet traffic over the sideband interface. The 82575EB has the ability to route Ethernet traffic to the host operating system as well as the ability to send Ethernet traffic over the sideband interface to an external BMC.

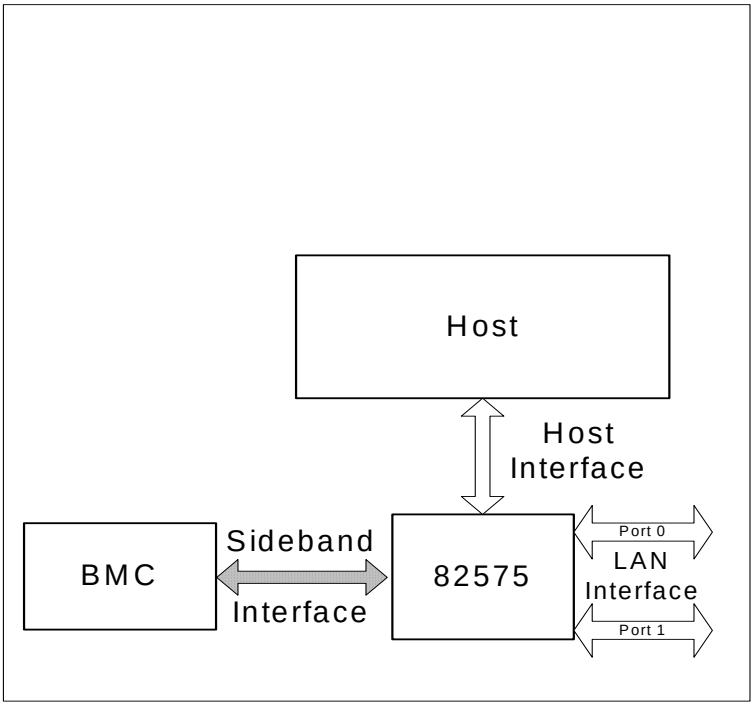
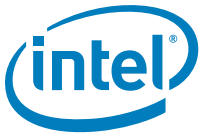


Figure 1. Sideband Interface

The sideband interface provides a mechanism by which the 82575EB can be shared between the host and the BMC. By providing this sideband interface, the BMC can communicate with the LAN without requiring a dedicated Ethernet controller to do so.

The 82575EB Ethernet controller supports two sideband interfaces:

- SMBus
- NC-SI



The usable bandwidth for either direction is up to 400 Kb/s when using the SMBus interface and 100 Mb/s for the NC-SI interface.

Note that only one mode of sideband can be active at any given time. This configuration is done via an EEPROM setting.

2.1 Components of a Sideband Interface

There are two components to a sideband interface:

- Physical Layer - The electrical layer that transfers data
- Logical Layer - the agreed upon protocol that is used for communications

The BMC and the 82575EB must be in alignment for both of these components. For example, the NC-SI physical interface is based on the RMII interface. However, there are some differences at the physical level (detailed in the NC-SI specification) and the protocol layer is completely different.

2.2 SMBus Pass-Through Interface

SMBus is the system management bus defined by Intel® Corporation in 1995. It is used in personal computers and servers for low-speed system management communications. The SMBus interface is one of two pass-through interfaces available in the 82575EB Ethernet controller.

This section describes how the SMBus interface in the 82575EB operates in pass-through mode.

2.3 General

The SMBus sideband interface includes the standard SMBus commands used for assigning a slave address and gathering device information as well as Intel® proprietary commands used specifically for the pass-through interface.

2.4 Pass-Through Capabilities

This section details the specific manageability capabilities the 82575EB provides while in SMBus mode.

The pass-through traffic is carried by the sideband interface as described in Section 2.0.

When operating in SMBus mode, in addition to exposing a communication channel to the LAN for the BMC, the 82575EB provides the following manageability services to the BMC:

- ARP handling - The 82575EB can be programmed to auto-ARP replying for ARP request packets to reduce the traffic over the BMC interconnect. See Section 2.5.2 for details.
- Teaming and fail-over - The 82575EB can be configured to one of several teaming and fail-over configurations (See Section 3.3.1):
  - No-teaming - The 82575EB dual LAN ports act independently of each other and no fail-over is provided by the 82575. The BMC is responsible for teaming and fail-over.

— Teaming - The 82575EB is configured to provide fail-over capabilities, such that manageability traffic is routed to an active port if any of the ports fail. Several modes of operation are provided.

Note: These services are not available in NC-SI mode.

2.4.1 Packet Filtering

Since the host OS and the BMC both use the 82575EB Ethernet controller to send and receive Ethernet traffic, there needs to be a mechanism by which incoming Ethernet packets can be identified as those that should be sent to the BMC rather than the host OS.

In order to determine the types of traffic that is forwarded to the BMC over the sideband interface, the 82575EB supports a manageability receive filtering mechanism. This mechanism is used to decide if a received packet should be forwarded to the BMC or to the host.

The following is a list of the filtering capabilities available for the SMBus interface with the 82575:

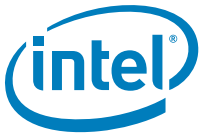
- RMCP/RMCP+ ports
- Flexible UDP/TCP port filters
- 128-byte flexible filters
- VLAN
- IPv4 address
- IPv6 address
- MAC address filters

Each of these are discussed in detail later in this document.

2.5 Pass-Through Multi-Port Modes

Pass-through configuration depends on the way the LAN ports are configured. If the LAN ports are configured as two different channels (non-teaming mode), then the 82575EB is presented on the manageability link as two different devices. For example, via two different SMB addresses on which each device is connected to a different LAN port. In this mode (the same as in the LAN channels), there is no logical connection between the two devices. In this mode the fail-over between the two LAN ports is done by the external BMC (by sending/receiving packets through different devices). The status report to the BMC, ARP handling, DHCP, and other pass-through functionality are unique for each port.

When the LAN ports are configured to work as one LAN channel (teaming mode), the 82575EB presents itself on the SMBus as one device (one SMB address). In this mode, the external BMC is not aware that there are two LAN ports. The 82575EB decides how to route the packet that it receives from the LAN according to the fail-over algorithm. The status report to the BMC and other pass-through configuration are common to both ports.



2.5.1 Automatic Ethernet ARP Operation

Automatic Ethernet ARP parameters are loaded from the EEPROM when the 82575EB is powered up or configured through the sideband management interface. The following parameters should be configured in order to enable ARP operation:

- ARP auto-reply enabled
- ARP IP address (to filter ARP packets)
- ARP MAC addresses (for ARP responses)

These are all configurable over the sideband interface using the advanced version of the Receive Enable command (see Section 3.2.1.3) or via the EEPROM.

When an ARP request packet is received on the wire, and ARP auto-reply is enabled, the 82575EB checks the targeted IP address (after the packet has passed L2 checks and ARP checks). If the targeted IP matches the IP the 82575EB was configured to, than it replies with an ARP response. The 82575EB responds to ARP request targeted to the ARP IP address with the ARP MAC address configured to it. In case that there is no match, the 82575EB silently discards the packets. If the 82575EB is not configured to do auto-ARP response it forwards the ARP packets to the BMC.

When the external BMC uses the same IP and MAC address of the OS, the ARP operation should be coordinated with the OS operation. In this mode, this is the external BMC's responsibility and the 82575EB stops/starts doing automatic ARP response as configured.

2.5.2 Manageability Receive Filtering

This section describes the manageability receive packet filtering flow when in SMBus mode. The description applies to a capability of each of the 82575EB LAN ports. Packets that are received by the 82575EB can be discarded, sent to the host memory, sent to the external BMC, or sent to both BMC and host memory.

2.5.2.1 Overview and General Structure

There are two modes of receive manageability filtering:

1. Receive Filtering - In this mode only certain types of packets are directed to the manageability block. The BMC should set the *RCV_TCO_EN* bit together with the specific packet type bits in the manageability filtering registers.

Note: The *RCV_ALL* bit must be cleared.

2. Receive All - all receive packets are routed to the BMC in this mode. It is enabled by setting the *RCV_TCO_EN* bit (which enables packets to be routed to the BMC) and the *RCV_ALL* bit (which routes all packets to the BMC) in the MANC register. RCV_ALL is only used for debug because it blocks all host traffic.

The default mode is that every packet that is directed to the BMC is not directed to host memory. The BMC enables the 82575EB to direct certain manageability packets also to host memory by setting the *EN_MNG2HOST* bit in the MANC register. It then needs to configure the 82575EB to send manageability packets to the host according to their type by setting the corresponding bits in the MANC2H register.

The BMC can control the types of packets that it receives by programming the receive manageability filters. Following is the list of filters that are accessible to the BMC:

Filters	Functionality	When Reset?
Filters Enable	General configuration of the manageability filters	Internal_Power_On_Reset and FW Reset
Manageability to Host	Enables routing of manageability packets to host	Internal_Power_On_Reset and FW Reset
Manageability Decision Filters [6:0]	Configuration of manageability decision filters	Internal_Power_On_Reset and FW Reset
MAC Address [3:0]	Four unicast MAC manageability addresses	Internal_Power_On_Reset
VLAN Filters [7:0]	Eight VLAN tag values	Internal_Power_On_Reset
UDP/TCP Port Filters [15:0]	16 destination port values	Internal_Power_On_Reset
Flexible 128 bytes TCO Filters [3:0]	Length values for four flex TCO filters	Internal_Power_On_Reset
IPv4 and IPv6 Address Filters [3:0]	IP address for manageability filtering	Internal_Power_On_Reset

All these filters are reset only on Internal_Power_On_Reset. Register filters that enable filters or functionality are also reset on FW reset. Also, these registers can be loaded from the EEPROM following a reset.

The high-level structure of manageability filtering is done in three steps:

1. Packets are filtered by L2 criteria (MAC address, unicast/multicast/broadcast).
2. Packets are then filtered by VLAN if a VLAN tag is present.
3. Packets are filtered by the manageability filters (port, IP, flex, other), as configured in the decision filters.

Some general rules:

- Fragmented packets are passed to manageability but not parsed beyond the IP header.
- Packets with L2 errors (CRC, alignment, other) are never forwarded to manageability.

Note: If the BMC uses a dedicated MAC address/VLAN tag, it should take care not to use L3/L4 decision filtering on top of it. Otherwise all the packets with the manageability MAC address/VLAN tag filtered out at L3/L4 are forwarded to the host.

The following sections describe each of these stages in detail.

2.5.2.2 L2 Layer Filtering

Figure 2 shows the manageability L2 filtering. A packet passes successfully through L2 filtering if any of the following conditions are met:

- It is a unicast packet and promiscuous unicast filtering is enabled from host.
- It is a multicast packet and promiscuous multicast filtering is enabled from host.
- It is a unicast packet and it matches one of the unicast MAC filters (host or manageability).
- It is a multicast packet and it matches one of the multicast filters.
- It is a broadcast packet. Note that in this case, the packet does not go through VLAN filtering (VLAN filtering is assumed to match).

Promiscuous unicast mode - Promiscuous unicast mode can be set/cleared only by the LAN device driver (not by the BMC), and it is usually used when the LAN device is used as a sniffer.

Promiscuous multicast mode - Promiscuous multicast is used in LAN devices that are used as a sniffer, and is controlled only by the LAN device driver. This bit can also be used by a BMC requiring forwarding of all multicasts.

Unicast filtering - The entire MAC address is checked against the 16 host unicast addresses and four management unicast addresses (if enabled). The 16 host unicast addresses are controlled by the LAN device driver (the BMC can not change them). The other four addresses are dedicated to management functions and are only accessed by the BMC.

The BMC configures manageability unicast filtering via update manageability receive filtering (see Section 3.2.1.6).

Multicast filtering - only 12 bits out of the packet's destination MAC address are compared against the multicast entries. These entries can be configured only by the LAN device driver and cannot be controlled by the BMC.

Note: Refer to the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual* for more information.

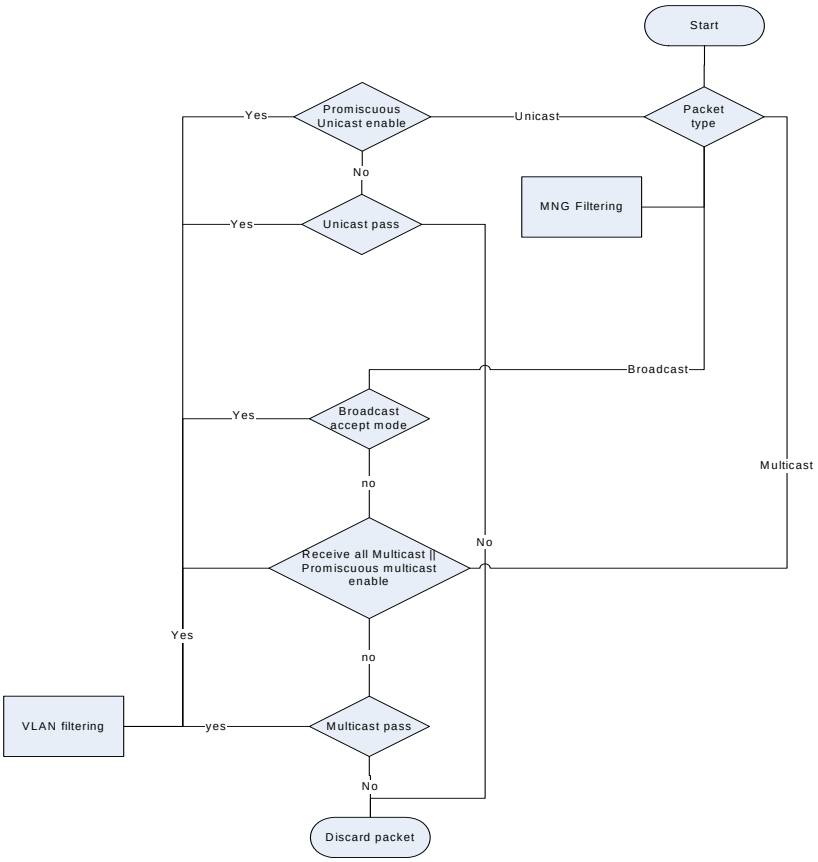


Figure 2. L2 Packet Filtering (Receive)

2.5.2.2.1 VLAN Filtering

Figure 3 shows the manageability VLAN filtering. A receive packet that passed L2 layer filtering successfully is then subjected to VLAN header filtering:

- If the packet does not have a VLAN header and the 82575EB is not configured to receive only VLAN packets, it passes to the next filtering stage (manageability filtering).
- If the packet has a VLAN header and it passes a valid manageability VLAN filter and the 82575EB is configured to receive VLAN packets, it passes to the next filtering stage (manageability filtering).
- If the packet has a VLAN header, the 82575EB manageability is configured to receive VLAN packets, and it matches an enabled host VLAN filter, the packet is forwarded to the next filtering stage (manageability filtering).
- Otherwise, the packet is dropped.

The BMC configures the 82575EB with up to eight different manageability VLAN IDs (VIDs) via the Management VLAN Tag Filters (see Section 2.5.2.2.1).

Note: Refer to the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual* for more details.

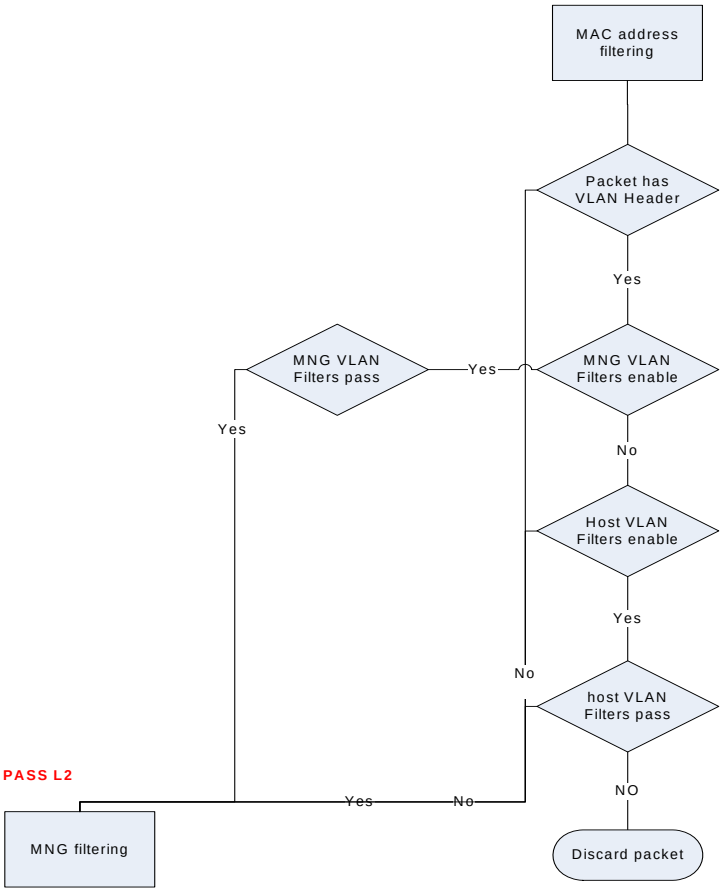


Figure 3. Manageability VLAN Header Filtering (Receive)

2.5.2.3 Manageability Decision Filtering

The manageability decision filtering stage combines some of the checks done at the previous stages with additional L3/L4 checks into a final decision whether to route a packet to the BMC. This section describes the additional filters done at layers L3 &L4, followed by the final filtering rules.

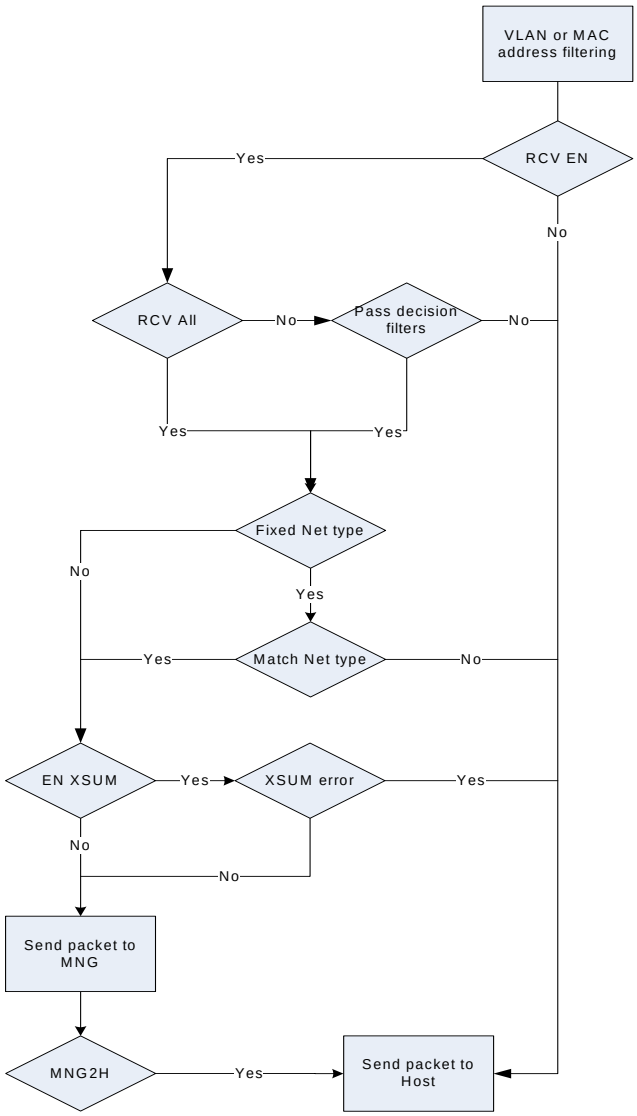


Figure 4. Manageability Filtering (Receive)

2.5.2.3.1 L3 & L4 Filters

ARP Filtering

The 82575EB supports filtering of both ARP request packets (initiated externally) and ARP responses (to requests initiated by the BMC or the 82575).

Neighbor Discovery Filtering

The 82575EB supports filtering of neighbor discovery packets. Neighbor discovery uses the IPv6 destination address filters defined in the MIPAF registers (for instance, all enabled IPv6 addresses are matched for neighbor discovery).

Port 0x298/0x26F Filtering

The 82575EB supports filtering by fixed destination ports numbers, port 0x26F and port 0x298.

Flex Port Filtering

The 82575EB implements 16 flex destination port filters. The 82575EB directs packets that their L4 destination port and matches the value of the respective word in the MFUTP registers. The BMC must insure that only valid entries are enabled in the decision filters.

Flex TCO Filters

The 82575EB provides four flex TCO filters. Each filter looks for a pattern match within the first 128 bytes of the packet. The BMC configures the pattern to match into the FTFT table, and the length in the last two entries of the FFLT table. The BMC must insure that only valid entries are enabled in the decision filters.

IP Address Filtering

The 82575EB supports filtering by IP address through IPv4 and IPv6 address filters, dedicated to manageability. Two modes are possible, depending on the value of the MANC.EN_IPv4_FILTER bit:

- EN_IPv4_FILTER = 0b: The 82575EB provides four IPv6 address filters.
- EN_IPv4_FILTER = 1b: The 82575EB provides three IPv6 address filters and 4 IPv4 address filters.
- The MFVAL register indicates which of the IP address filters are valid (for instance, contains a valid entry and should be used for comparison).

Checksum Filter

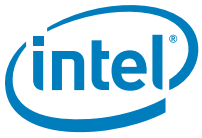
If bit MANC.EN_XSUM_FILTER is set, the 82575EB directs packets to the BMC only if they pass (if exists) L3/L4 checksum, in addition to matching the previously mentioned filters. Enabling this filter makes it so the BMC does not need to do the L3/L4 checksum verification, as a packet that fails this filter will never be sent to the BMC.

To enable the checksum filter, the BMC uses the Update Management Receive Filter Parameters command (see Section 3.2.1.6) with the parameter of 0x1 to configure the MANC register, setting the *EN_XSUM_FILTER* bit (bit 23). Refer to the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual* for MANC register details.

2.5.2.4 Manageability Decision Filters

The manageability decision filters are a set of eight filters with the same structure. The filtering rule for each decision filter is programmed by the BMC and defines which of the filters (L2, VLAN, manageability) participate in the decision. A packet that passes at least one rule is directed to manageability and possibly to the host.

The inputs to each decision filter are:



- Packet passed a valid management L2 unicast address filter
- Packet is a broadcast packet
- Packet has a VLAN header and it passed a valid manageability VLAN filter
- Packet matched one of the valid IPv4 or IPv6 manageability address filters
- Packet passed ARP filtering (request or response)
- Packet passed neighbor discovery filtering
- Packet passed 0x298/0x26F port filter
- Packet passed a valid flex port filter
- Packet passed a valid flex TCO filter
- Packet is multicast

The structure of each of the decision filters is shown in Figure 3 4. A boxed number indicates that the input is conditioned on a mask bit defined in the MDEF register for this rule. The decision filter rules are as follows:

- At least one bit must be set in a MDEF register. If all bits are cleared (such as MDEF=0x0000), then the decision filter is disabled and ignored.
- All enabled AND filters must match for the decision filter to match. An AND filter not enabled in the MDEF register is ignored.
- If no OR filter is enabled in the MDEF register, the OR filters are ignored in the decision (such as the filter might still match).
- If at least one OR filter is enabled in the MDEF register, then at least one of the enabled OR filters must match for the decision filter to match.

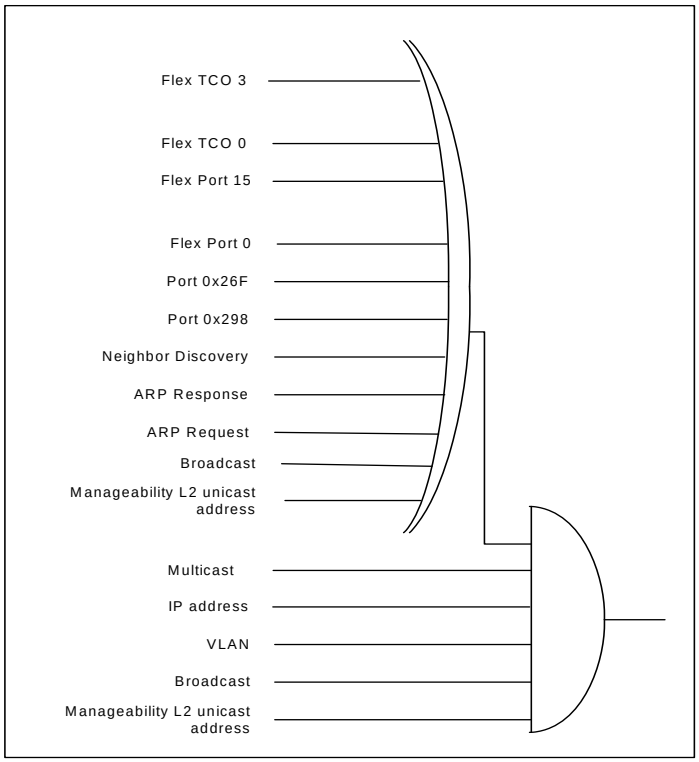


Figure 5. Manageability Decision Filters

A decision filter (for any of the seven filters) defines which of the inputs is enabled as part of the rule. The BMC programs a 32-bit rule with the settings as listed in Table 1. A set bit enables its corresponding filter to participate in the filtering decision.

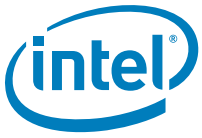


Table 1. Assignment of Decision Filters

Filter	And / OR Input	Mask Bits in MDEF[7:0]
L2 unicast address	AND	0
Broadcast	AND	1
Manageability VLAN	AND	2
IP address	AND	3
L2 unicast address	OR	4
Broadcast	OR	5
Multicast	AND	6
ARP request	OR	7
ARP response	OR	8
Neighbor discovery	OR	9
Port 0x298	OR	10
Port 0x26F	OR	11
Flex port 15:0	OR	27:12
Flex TCO 3:0	OR	31:28

The default mode is that every packet that is directed to the BMC is not directed to host memory. The BMC can also enable the 82575EB to direct certain manageability packets to host memory by setting the *EN_MNG2HOST bit* in the MANC register and then configuring the 82575EB to send manageability packets to the host according to their type by setting the corresponding bits in the manageability to host filter (one bit per each of the seven decision rules).

The Mng2Host register has the following structure:

Bits	Description	Default
0	Decision Filter 0	Determines if packets that have passed decision filter 0 are also forwarded to the host OS.
1	Decision Filter 1	Determines if packets that have passed decision filter 1 are also forwarded to the host OS.
2	Decision Filter 2	Determines if packets that have passed decision filter 2 are also forwarded to the host OS.
3	Decision Filter 3	Determines if packets that have passed decision filter 3 are also forwarded to the host OS.
4	Decision Filter 4	Determines if packets that have passed decision filter 4 are also forwarded to the host OS.
5	Unicast and Mixed	Determines if broadcast packets are also forwarded to the host OS.
6	Global Multicast	Determines if unicast packets are also forwarded to the host OS.
7	Broadcast	Determines if multicast packets are also forwarded to the host OS.

All manageability filters are controlled by the BMC only and not by the LAN device driver.

The BMC enables these filters by issuing the Update Management Receive Filter Parameters command (see Section 3.2.1.6) with the parameter of 0x60.

2.6 SMBus Transactions

This section gives a brief overview of the SMBus protocol.

Following is an example for a format of a typical SMBus transaction:

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0010	0	[Data Dependent]	0	

The top row of the table identifies the bit length of the field in a decimal bit count. The middle row (bordered) identifies the name of the fields used in the transaction. The last row appears only with some transactions, and lists the value expected for the corresponding field. This value can be either hexadecimal or binary.

The shaded fields are fields that are driven by the slave of the transaction. The un-shaded fields are fields that are driven by the master of the transaction. The SMBus controller is a master for some transactions and a slave for others. The differences are identified in this document.

Shorthand field names are listed in Table 2 and are fully defined in the SMBus specification:

Table 2. Shorthand Field Name

Field Name	Definition
S	SMBus START Symbol
P	SMBus STOP Symbol
PEC	Packet Error Code
A	ACK (Acknowledge)
N	NACK (Not Acknowledge)
Rd	Read Operation (Read Value = 1b)
Wr	Write Operation (Write Value = 0b)

2.6.1 SMBus Addressing

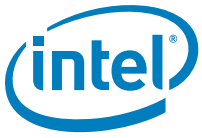
The SMBus addresses that the 82575EB responds to depends on the LAN mode (teaming/non-teaming). If the LAN is in teaming mode (fail-over), the 82575EB is presented over the SMBus as one device and has one SMBus address. While in non-teaming mode in the LAN ports, the SMBus is presented as two SMBus devices on the SMBus (two SMBus addresses). In dual-address mode, all pass-through functionality is duplicated on the SMBus address, where each SMBus address is connected to a different LAN port. Note that it is not allowed to configure both ports to the same SMBus address. When a LAN function is disabled, the corresponding SMBus address is not presented to the BMC (see Section 3.3.1).

The SMBus addressing mode is defined through the SMBus *Addressing Mode* bit in the EEPROM. The SMBus addresses are set in SMBus address 0 and SMBus address 1 in the EEPROM. Note that if single-address mode is set, only the SMBus address 0 field is valid.

The SMBus addresses (enabled from the EEPROM) can be re-assigned using the SMBus ARP protocol.

In addition to the SMBus address values, all parameters of the SMBus (SMBus channel selection, address mode, and address enable) can be set only through EEPROM configuration. Note that the EEPROM is read at 82575EB power up and resets.

All SMBus addresses should be in Network Byte Order (NBO); MSB first.



2.6.2 SMBus ARP Functionality

The 82575EB supports the SMBus ARP protocol as defined in the SMBus 2.0 specification. The 82575EB is a persistent slave address device so its SMBus address is valid after power-up and loaded from the EEPROM. The 82575EB supports all SMBus ARP commands defined in the SMBus specification both general and directed.

Note: The SMBus ARP capability can be disabled through the EEPROM.

2.6.2.1 SMBus ARP Flow

SMBus ARP flow is based on the status of two flags:

- AV (Address Valid): This flag is set when the 82575EB has a valid SMBus address.
- AR (Address Resolved): This flag is set when the 82575EB SMBus address is resolved (SMBus address was assigned by the SMBus ARP process).

Note: These flags are internal 82575EB flags and are not exposed to external SMBus devices.

Since the 82575EB is a Persistent SMBus Address (PSA) device, the AV flag is always set, while the AR flag is cleared after power up until the SMBus ARP process completes. Since AV is always set, the 82575EB always has a valid SMBus address.

When the SMBus master needs to start an SMBus ARP process, it resets (in terms of ARP functionality) all devices on the SMBus by issuing either Prepare to ARP or Reset Device commands. When the 82575EB accepts one of these commands, it clears its AR flag (if set from previous SMBus ARP process), but not its AV flag (The current SMBus address remains valid until the end of the SMBus ARP process).

Clearing the AR flag means that the 82575EB responds to the following SMBus ARP transactions that are issued by the master. The SMBus master issues a Get UDID command (general or directed) to identify the devices on the SMBus. The 82575EB always responds to the Directed command and to the General command only if its AR flag is not set. After the Get UDID, The master assigns the 82575EB SMBus address by issuing an Assign Address command. The 82575EB checks whether the UDID matches its own UDID and if it matches, it switches its SMBus address to the address assigned by the command (byte 17). After accepting the Assign Address command, the AR flag is set and from this point (as long as the AR flag is set), the 82575EB does not respond to the Get UDID General command. Note that all other commands are processed even if the AR flag is set. The 82575EB stores the SMBus address that was assigned in the SMBus ARP process in the EEPROM, so at the next power up, it returns to its assigned SMBus address.

SMBus ARP Flow shows the 82575EB SMBus ARP flow.

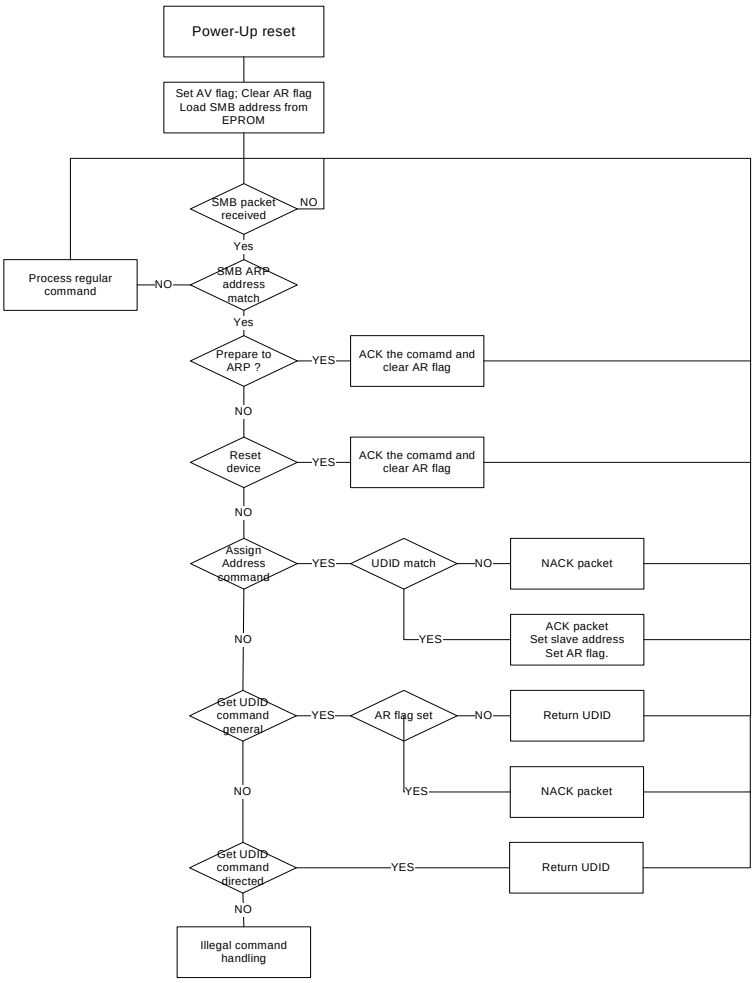
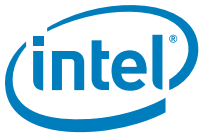


Figure 6. SMBus ARP Flow

2.6.2.2 SMBus ARP UDID Content

The UDID provides a mechanism to isolate each device for the purpose of address assignment. Each device has a unique identifier. The 128-bit number is comprised of the following fields:

1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes	4 Bytes
Device Capabilities	Version/ Revision	Vendor ID	Device ID	Interface	Subsystem Vendor ID	Subsystem Device ID	Vendor Specific ID
See Below	See Below	0x8086	0x10AA	0x0004	0x0000	0x0000	See Below
MSB							LSB



Where:

Vendor ID:	The device manufacturer's ID as assigned by the SBS Implementers' Forum or the PCI SIG. Constant value: 0x8086
Device ID:	The device ID as assigned by the device manufacturer (identified by the Vendor ID field). Constant value: 0x10AA
Interface:	Identifies the protocol layer interfaces supported over the SMBus connection by the device. In this case, SMBus Version 2.0 Constant value: 0x0004
Subsystem Fields:	These fields are not supported and return zeros.

Device Capabilities: Dynamic and Persistent Address, *PEC Support* bit:

7	6	5	4	3	2	1	0
Address Type		Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	PEC Supported
0b	1b	0b	0b	0b	0b	0b	0b
MSB							LSB

Version/Revision: UDID Version 1, Silicon Revision:

7	6	5	4	3	2	1	0
Reserved (0)	Reserved (0)	UDID Version			Silicon Revision ID		
0b	0b	001b			See Below		
MSB							LSB

Silicon Revision ID:

Silicon Version	Revision ID
A0	000b
A1	001b
A2	010b

Vendor Specific ID: Four LSB bytes of the device Ethernet MAC Address. The device Ethernet address is taken from words 0x2:0x0 in the EEPROM. Note that in the 82575EB there are two MAC addresses (one for each port). Bit 0 of the port 1 MAC address has the inverted value of bit 0 from the EEPROM.

1 Byte	1 Byte	1 Byte	1 Byte
MAC Address, Byte 3	MAC Address, Byte 2	MAC Address, Byte 1	MAC Address, Byte 0
MSB			LSB

2.6.2.3 SMBus ARP in Dual/Single Mode

The 82575EB operates in either single SMBus address mode or in dual SMBus address mode. These modes reflect its SMBus ARP behavior.

While operating in single mode, the 82575EB presents itself on the SMBus as one device and only responds to SMBus ARP as one device. In this case, the 82575's SMBus address is SMBus address 0 as defined in the EEPROM SMBus ARP address word. The 82575EB has only one AR and AV flag. The vendor ID, the MAC address of the LAN's port, is taken from the port 0 address.

In dual mode, the 82575EB responds as two SMBus devices having two sets of AR/AV flags (one for each port). The 82575EB responds twice to the SMBus ARP master, once each for each port. Both SMBus addresses are taken from the SMBus ARP address word of the EEPROM. Note that the Unique Device Identifier (UDID) is different between the two ports in the version ID field, which represents the MAC address and is different between the two ports. It is recommended that the 82575EB first respond as port 0, and only when the address is assigned, to start responding as port 1 to the Get UDID command.

2.6.3 Concurrent SMBus Transactions

In single-address mode, concurrent SMBus transactions (receive, transmit and configuration read/write) are allowed without limitation. Transmit fragments can be sent between receive fragments and configuration Read/Write commands can also issue between receive and transmit fragments.

In dual-address mode, the same rules apply to concurrent traffic between the two addresses supported by the 82575.

Note: Packets can only be transmitted from one port/device at a given time. As a result, the BMC must finish sent packets (send a last fragment command) from one port before starting the transmission for the other port.

2.7 SMBus Notification Methods

The 82575EB supports three methods of notifying the BMC that it has information that needs to be read by the BMC:

- SMBus alert
- Asynchronous notify
- Direct receive

The notification method that is used by the 82575EB can be configured from the SMBus using the Receive Enable command. This default method is set by the EEPROM in the pass-through init field.

The following events cause the 82575EB to send a notification event to the BMC:

- Receiving a LAN packet that is designated to the BMC.
- Receiving a Request Status command from the BMC initiates a status response.
- Status change has occurred and the 82575EB is configured to notify the external BMC at one of the status changes.
- Change in any in the Status Data 1 bits of the Read Status command.

There can be cases where the BMC is hung and therefore not responding to the SMBus notification. The 82575EB has a time-out value (defined in the EEPROM) to avoid hanging while waiting for the notification response. If the BMC does not respond until the time out expires, the notification is de-asserted and all pending data is silently discarded.

Note that the SMBus notification time-out value can only be set in the EEPROM, the BMC cannot modify this value.

2.7.1 SMBus Alert and Alert Response Method

The SMBus Alert# (SMBALERT_N) signal is an additional SMBus signal that acts as an asynchronous interrupt signal to an external SMBus master. The 82575EB asserts this signal each time it has a message that it needs the BMC to read and if the chosen notification method is the SMBus alert method. Note that the SMBus alert method is an open-drain signal which means that other devices besides the 82575EB can be connected on the same alert pin. As a result, the BMC needs a mechanism to distinguish between the alert sources.

The BMC can respond to the alert, by issuing an ARA Cycle command, to detect the alert source device. The 82575EB responds to the ARA cycle with its own SMBus slave address (if it was the SMBus alert source) and de-asserts the alert when the ARA cycle is completes. Following the ARA cycle, the BMC issues a read command to retrieve the 82575EB message.

Some BMCs do not implement the ARA cycle transaction. These BMCs respond to an alert by issuing a Read command to the 82575EB (0xC0/0xD0 or 0xDE). The 82575EB always responds to a Read command, even if it is not the source of the notification. The default response is a status transaction. If the 82575EB is the source of the SMBus Alert, it replies the read transaction and then de-asserts the alert after the command byte of the read transaction.

Note: In SMBus Alert mode, the SMBALERT_N pin is used for notification. In dual-address mode, both devices generate alerts on events that are independent of each other.

The ARA cycle is a SMBus Receive Byte transaction to SMBus address 0x18. Note that the ARA transaction does not support PEC. The alert response address transaction format is shown in Table 3.

Table 3. SMBus ARA Cycle Format

1	7	1	1	8	1	1
S	Alert Response Address	Rd	A	Slave Device Address	A	P
	0001 100	0	0		1	

2.7.2 Asynchronous Notify Method

When configured using the asynchronous notify method, the 82575EB acts as a SMBus master and notifies the BMC by issuing a modified form of the write word transaction. The asynchronous notify transaction SMBus address and data payload is configured using the Receive Enable command or using the EEPROM defaults. Note that the asynchronous notify is not protected by a PEC byte.

1	7	1	1	7	1	1	
S	Target Address	Wr	A	Sending Device Address		A	. . .
	BMC Slave Address	0	0	MNG Slave SMBus Address	0	0	

8	1	8	1	1
Data Byte Low	A	Data Byte High	A	P
Interface	0	Alert Value	0	

The target address and data byte low/high is taken from the Receive Enable command or EEPROM configuration.

2.7.3 Direct Receive Method

If configured, the 82575EB has the capability to send a message it needs to transfer to the external BMC as a master over the SMBus instead of alerting the BMC and waiting for it to read the message.

The message format is shown below. Note that the command that is used is the same command that is used by the external BMC in the Block Read command. The opcode that the 82575EB puts in the data is also the same as it put in the Block Read command of the same functionality. The rules for the *F* and *L* flags (bits) are also the same as in the Block Read command.

1	7	1	1	1	1	6	1	
S	Target Address	Wr	A	F	L	Command	A	. . .
	BMC Slave Address	0	0	First Flag	Last Flag	Receive TCO Command 01 0000b	0	

8	1	8	1		1	8	1	1
Byte Count	A	Data Byte 1	A	. . .	A	Data Byte N	A	P
N	0		0		0		0	

2.8 1 MHz SMBus Support

SMBus specification defines the maximum frequency of the SMBus as 100 KHz. The 82575EB SMBus can be activated up to frequency of 1 MHz. When operating at 1 MHz, few of the SMBus specification parameters are violated. The regular SMBus can be activated in two modes: slow, which meets the SMBus specification requirements (can be activated up to 400 KHz without violating hold and setup time) and fast, which can be operated up to 1 MHz, but does not meet the SMBus specification.

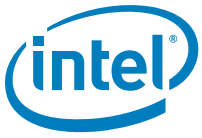
This configuration is only available via an EEPROM setting.

The EEPROM Pass-Through SMBus Connection field defines the SMBus mode (slow SMBus/fast SMBus). Note that SMBus frequency support can be defined only from the EEPROM.

The slow SMBus DC parameters are defined in the SMBus 2.0 specification. The DC parameters of the 1 MHz SMBus are described in the datasheet.

2.9 Receive TCO Flow

The 82575EB is used as a channel for receiving packets from the network link and passing them to the external BMC. The BMC configures the 82575EB to pass these specific packets to the BMC. Once a full packet is received from the link and identified as a manageability packet that should be transferred to the BMC, the 82575EB starts the receive TCO flow to the BMC.



The 82575EB uses the SMBus notification method to notify the BMC that it has data to deliver. Since the packet size might be larger than the maximum SMBus fragment size, the packet is divided into fragments, where the 82575EB uses the maximum fragment size allowed in each fragment (configured via the EEPROM). The last fragment of the packet transfer is always the status of the packet. As a result, the packet is transferred in at least two fragments. The data of the packet is transferred as part of the receive TCO LAN packet transaction.

When SMBus alert is selected as the BMC notification method, the 82575EB notifies the BMC on each fragment of a multi fragment packet. When asynchronous notify is selected as the BMC notification method, the 82575EB notifies the BMC only on the first fragment of a received packet. It is the BMC's responsibility to read the full packet including all the fragments.

Any timeout on the SMBus notification results in discarding the entire packet. Any NACK by the BMC causes the fragment to be re-transmitted to the BMC on the next Receive Packet command.

The maximum size of the received packet is limited by the 82575EB hardware to 1536 bytes. Packets larger then 1536 bytes are silently discarded. Any packet smaller than 1536 bytes is processed by the 82575.

2.10 Transmit TCO Flow

The 82575EB is used as the channel for transmitting packets from the external BMC to the network link. The network packet is transferred from the BMC over the SMBus and then, when fully received by the 82575, is transmitted over the network link.

In dual-address mode, each SMBus address is connected to a different LAN port. When a packet is received during a SMBus transaction using SMBus address #0, it is transmitted to the network using LAN port #0 and is transmitted through LAN port #1, if received on SMBus address #1. In single address mode, the transmitted port is chosen according to the fail-over algorithm.

The 82575EB supports packets up to an Ethernet packet length of 1536 bytes. Since SMBus transactions can only be up to 240 bytes in length, packets might need to be transferred over the SMBus in more than one fragment. This is achieved using the *F* and *L* bits in the command number of the transmit TCO packet Block Write command. When the *F* bit is set, it is the first fragment of the packet. When the *L* bit is set, it is the last fragment of the packet. When both bits are set, the entire packet is in one fragment. The packet is sent over the network link, only after all its fragments are received correctly over the SMBus. The maximum SMBus fragment size is defined within the EEPROM and cannot be changed by the BMC.

If the packet sent by the BMC is larger than 1536 bytes, than the packet is silently discarded by the 82575. The minimum packet length defined by the 802.3 spec is 64 bytes. The 82575EB pads packets that are less than 64 bytes to meet the specification requirements (there is no need for the external BMC to pad packets less than 64 bytes). If the packet sent by the BMC is larger than 1536 bytes the 82575EB silently discards the packet.

The 82575EB calculates the L2 CRC on the transmitted packet and adds its four bytes at the end of the packet. Any other packet field (such as XSUM) must be calculated and inserted by the BMC (the 82575EB does not change any field in the transmitted packet, other than adding padding and CRC bytes).

If the network link is down when the 82575EB has received the last fragment of the packet from the BMC, it silently discards the packet. Note that any link down event during the transfer of any packet over the SMBus does not stop the operation since the 82575EB waits for the last fragment to end to see whether the network link is up again.

2.10.1 Transmit Errors in Sequence Handling

Once a packet is transferred over the SMBus from the BMC to the 82575, the *F* and *L* flags should follow specific rules. The *F* flag defines that this is the first fragment of the packet; the *L* flag defines that the transaction contains the last fragment of the packet.

Flag options during transmit packet transactions lists the different flag options in transmit packet transactions:

Table 4. Flag Options During Transmit Packet Transactions

Previous	Current	Action / Notes
Last	First	Accept both.
Last	Not First	Error for the current transaction. Current transaction is discarded and an abort status is asserted.
Not Last	First	Error in previous transaction. Previous transaction (until previous First) is discarded. Current packet is processed. No abort status is asserted.
Not Last	Not First	Process the current transaction.

Note: Since every other Block Write command in TCO protocol has both *F* and *L* flags off, they cause flushing any pending transmit fragments that were previously received. When running the TCO transmit flow, no other Block Write transactions are allowed in between the fragments.

2.10.2 TCO Command Aborted Flow

The 82575EB indicates to the BMC an error or an abort condition by setting the TCO Abort bit in the general status. The 82575EB might also be configured to send a notification to the BMC (see Section 3.2.1.3.3).

Following is a list of possible error and abort conditions:

- Any error in the SMBus protocol (NACK, SMBus timeouts, etc.).
- Any error in compatibility between required protocols to specific functionality (for example, RX Enable command with a byte count not equal to 1/14, as defined in the command specification).
- If the 82575EB does not have space to store the transmitted packet from the BMC (in its internal buffer space) before sending it to the link, the packet is discarded and the external BMC is notified via the *Abort* bit.
- Error in the *F*/*L* bit sequence during multi-fragment transactions.
- An internal reset to the 82575's firmware.

3.0 82575EB SMBus Transactions

3.1 SMBus ARP Transactions

Note: All SMBus ARP transactions include the PEC byte.

3.1.1 Prepare to ARP

This command clears the *Address Resolved* flag (set to false). It does not affect the status or validity of the dynamic SMBus address and is used to inform all devices that the ARP master is starting the ARP process:

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0001	0	[Data Dependent Value]	0	

3.1.2 Reset Device (General)

This command clears the *Address Resolved* flag (set to false). It does not affect the status or validity of the dynamic SMBus address.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0010	0	[Data Dependent Value]	0	

3.1.3 Reset Device (Directed)

The Command field is NACKed if bits 7:1 do not match the current 82575EB SMBus address. This command clears the *Address Resolved* flag (set to false) and does not affect the status or validity of the dynamic SMBus address.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	Targeted Slave Address | 0	0	[Data Dependent Value]	0	

3.1.4 Assign Address

This command assigns the 82575EB SMBus address. The address and command bytes are always acknowledged.

The transaction is aborted (NACKed) immediately if any of the UDID bytes is different from the 82575EB UDID bytes. If successful, the manageability system internally updates the SMBus address. This command also sets the *Address Resolved* flag (set to true).

1	7	1	1	8	1	8	1	
S	Slave Address	Wr	A	Command	A	Byte Count	A	•••
	1100 001	0	0	0000 0100	0	0001 0001	0	

8	1	8	1	8	1	8	1	
Data 1	A	Data 2	A	Data 3	A	Data 4	A	•••
UDID Byte 15 (MSB)	0	UDID Byte 14	0	UDID Byte 13	0	UDID Byte 12	0	

8	1	8	1	8	1	8	1	
Data 5	A	Data 6	A	Data 7	A	Data 8	A	•••
UDID Byte 11	0	UDID Byte 10	0	UDID Byte 9	0	UDID Byte 8	0	

8	1	8	1	8	1	
Data 9	A	Data 10	A	Data 11	A	•••
UDID Byte 7	0	UDID Byte 6	0	UDID Byte 5	0	

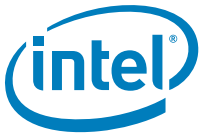
8	1	8	1	8	1	8	1	
Data 12	A	Data 13	A	Data 14	A	Data 15	A	•••
UDID Byte 4	0	UDID Byte 3	0	UDID Byte 2	0	UDID Byte 1	0	

8	1	8	1	8	1	1
Data 16	A	Data 17	A	PEC	A	P
UDID Byte 0 (LSB)	0	Assigned Address	0	[Data Dependent Value]	0	

3.1.5 Get UDID (General and Directed)

The general get UDID SMBus transaction supports a constant command value of 0x03 and in directed, supports a Dynamic command value equal to the dynamic SMBus address.

If the SMBus address has been resolved (*Address Resolved* flag set to true), the manageability system does not acknowledge (NACK) this transaction. If its a General command, the manageability system always acknowledges (ACKs) as a directed transaction.



This command does not affect the status or validity of the dynamic SMBus address or the *Address Resolved* flag.

S	Slave Address	Wr	A	Command	A	S	•••
	1100 001	0	0	See Below	0		

7	1	1	8	1	
Slave Address	Rd	A	Byte Count	A	•••
1100 001	1	0	0001 0001	0	

8	1	8	1	8	1	8	1	
Data 1	A	Data 2	A	Data 3	A	Data 4	A	•••
UDID Byte 15 (MSB)	0	UDID Byte 14	0	UDID Byte 13	0	UDID Byte 12	0	

8	1	8	1	8	1	8	1	
Data 5	A	Data 6	A	Data 7	A	Data 8	A	•••
UDID Byte 11	0	UDID Byte 10	0	UDID Byte 9	0	UDID Byte 8	0	

8	1	8	1	8	1	
Data 9	A	Data 10	A	Data 11	A	•••
UDID Byte 7	0	UDID Byte 6	0	UDID Byte 5	0	

8	1	8	1	8	1	8	1	
Data 12	A	Data 13	A	Data 14	A	Data 15	A	•••
UDID Byte 4	0	UDID Byte 3	0	UDID Byte 2	0	UDID Byte 1	0	

8	1	8	1	8	1	1
Data 16	A	Data 17	A	PEC	~Ã	P
UDID Byte 0 (LSB)	0	Device Slave Address	0	[Data Dependent Value]	1	

Note: The Get UDID command depends on whether or not this is a Directed or General command.

The General Get UDID SMBus transaction supports a constant command value of 0x03.

The Directed Get UDID SMBus transaction supports a Dynamic command value equal to the dynamic SMBus address with the LSB bit set.

Bit 0 (LSB) of Data byte 17 is always 1b.

3.2 SMBus Pass-Through Transactions

This section details all of the commands (both read and write) that the 82575EB SMBus interface supports for pass-through.

3.2.1 Write Transactions

This section details the commands that the BMC can send to the 82575EB over the SMBus interface. SMBus write transactions table lists the different SMBus write transactions supported by the 82575.

TCO Command	Transaction	Command	Fragmentation	Section
Transmit Packet	Block Write	First: 0x84 Middle: 0x04 Last: 0x44	Multiple	3.2.1.1
Transmit Packet	Block Write	Single: 0xC4	Single	3.2.1.1
Request Status	Block Write	Single: 0xDD	Single	3.2.1.2
Receive Enable	Block Write	Single: 0xCA	Single	3.2.1.3
Force TCO	Block Write	Single: 0xCF	Single	3.2.1.4
Management Control	Block Write	Single: 0xC1	Single	3.2.1.5
Update MNG RCV Filter Parameters	Block Write	Single: 0xCC	Single	3.2.1.6

3.2.1.1 Transmit Packet Command

Note: If the overall packet length is greater than 1536 bytes, the packet is silently discarded by the 82575.

3.2.1.2 Request Status Command

An external BMC can initiate a request to read the 82575EB manageability status by sending a Request Status command. When received, the 82575EB initiates a notification to an external BMC (when status is ready), after which, an external BMC is able to read the status by issuing this command. The format is as follows:

Function	Command	Byte Count	Data 1
Request Status	0xDD	1	0

3.2.1.3 Receive Enable Command

The Receive Enable command is a single fragment command used to configure the 82575. This command has two formats: short, 1-byte legacy format (providing backward compatibility with previous components) and long, 14-byte advanced format (allowing greater configuration capabilities). The Receive Enable command format is as follows:

Function	CMD	Byte Count	Data 1	Data 2	...	Data 7	Data 8	...	Data 11	Data 12	Data 13	Data 14
Legacy Receive Enable	0xCA	1	Receive Control Byte	-	...	-	-	...	-	-	-	-
Advanced Receive Enable		14 (0x0E)		MAC Addr LSB		MAC Addr MSB	IP Addr LSB		IP Addr MSB	BMC SMBus Addr	I/F Data Byte	Alert Value Byte

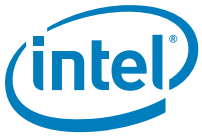


Table 5. Receive Control Byte (Data Byte)

Field	Bit(s)	Description
RCV_EN	0	Receive TCO Enable. 0b: Disable receive TCO packets. 1b: Enable Receive TCO packets. Setting this bit enables all manageability receive filtering operations. Enabling specific filters is done via the EEPROM or through special configuration commands. **Note:** When the RCV_EN bit is cleared, all receive TCO functionality is disabled, not just the packets that are directed to the BMC (also auto ARP packets).
RCV_ALL	1	Receive All Enable. 0b: Disable receiving all packets. 1b: Enable receiving all packets. Forwards all packets received over the wire that passed L2 filtering to the external BMC. This flag has no effect if bit 0 (Enable TCO packets) is disabled.
EN_STA	2	Enable Status Reporting. 0b: Disable status reporting. 1b: Enable status reporting.
EN_ARP_RES	3	Enable ARP Response. 0b: Disable the 82575EB ARP response. The 82575EB treats ARP packets as any other packet, for example, packet is forwarded to the BMC if it passed other (non-ARP) filtering. 1b: Enable the 82575EB ARP response. The 82575EB automatically responds to all received ARP requests that match its IP address. Note that setting this bit does not change the Rx filtering settings. Appropriate Rx filtering to enable ARP request packets to reach the BMC should be set by the BMC or by the EEPROM. The BMC IP address is provided as part of the Receive Enable message (bytes 8-11). If a short version of the command is used, the 82575EB uses IP address configured in the most recent long version of the command in which the EN_ARP_RES bit was set. If no such previous long command exists, then the 82575EB uses the IP address configured in the EEPROM as ARP Response IPv4 Address in the pass-through LAN configuration structure. If the *CBDM* bit is set, the 82575EB uses the BMC dedicated MAC address in ARP response packets. If the CBDM bit is not set, the BMC uses the host MAC address.

NM	5:4	Notification Method. Define the notification method the 82575EB uses. 00b: SMBUS Alert. 01b: Asynchronous Notify. 10b: Direct Receive. 11b: Not Supported. **Note:** In dual SMBus address mode, both SMBus addresses **must** be configured with the same notification method.
Reserved	6	Reserved. Must be set to 1b.
CBDM	7	Configure the BMC Dedicated MAC Address. **Note:** This bit should be set to 0b when the *RCV_EN* bit (bit 0) is not set. 0b: The 82575EB shares the MAC address for MNG traffic with the host MAC address specified in EEPROM words 2h:0h. 1b: The 82575EB uses the BMC dedicated MAC address as a filter for incoming receive packets and as the sender address in ARP response packets. The BMC MAC address is set in bytes 7:2 in this command. If the short version of the command is used, the 82575EB uses the MAC address configured in the most recent long version of the command in which the CBDM bit was set. If no such previous long command exists, then the 82572 uses the MAC address configured in the EEPROM as MAC address 3 (MMAL/H3) in the pass-through LAN configuration structure. When the Dedicated MAC address feature is activated, the 82575EB uses the following registers to filter in all the traffic addressed to the BMC MAC. The BMC should not modify these registers as follows: Manageability Decision Filter - MDEF7 (and corresponding bit 7 in Management Control To Host Register - MANC2H), Manageability MAC Address Low - MMAL3 and Manageability MAC Address High - MMAH3 (and corresponding bit 3 of Manageability Filters Valid - MFVAL).

3.2.1.3.1 Management MAC Address (Data Bytes 7:2)

Ignored if the *CBDM* bit is not set. This MAC address is used to configure the dedicated MAC address. In addition, it is used in the ARP response packet when the EN_ARP_RES bit is set. This MAC address is also used when *CBDM* bit is set in subsequent short versions of this command.

3.2.1.3.2 Management IP Address (Data Bytes 11:8)

This IP address is used to filter ARP request packets.

3.2.1.3.3 Asynchronous Notification SMBus Address (Data Byte 12)

This address is used for the asynchronous notification SMBus transaction and for direct receive.

3.2.1.3.4 Interface Data (Data Byte 13)

Interface data byte used in asynchronous notification.

3.2.1.3.5 Alert Value Data (Data Byte 14)

Alert Value data byte used in asynchronous notification.

3.2.1.4 Force TCO Command

This command causes the 82575EB to perform a TCO reset, if Force TCO reset is enabled in the EEPROM. The force TCO reset clears the data path (Rx/Tx) of the 82575EB to enable the BMC to transmit/receive packets through the 82575. Note that in single -address mode, both ports are reset when the command is issued. In dual-address mode, force TCO reset is asserted only to the port related to the SMBus address the command was issued to. This command should only be used when the BMC is unable to transmit receive and suspects that the 82575EB is inoperable. This command also causes the LAN device driver to unload. It is recommended to perform a system restart to resume normal operation.

The 82575EB considers the Force TCO command as an indication that the operating system is hung and clears the *DRV_LOAD* flag. The Force TCO Reset command format is as follows:

Function	Command	Byte Count	Data 1
Force TCO Reset	0xCF	1	TCO Mode

Where TCO Mode is:

Field	Bit(s)	Description
DO_TCO_RST	0	Perform TCO Reset. 0b: Do nothing. 1b: Perform TCO reset.
Reserved	7:1	Reserved (set to 0x00).

3.2.1.5 Management Control

This command is used to set generic manageability parameters. The parameters list is shown in Management Control Command Parameters/Content. The command is 0xC1 which states that it is a Management Control command. The first data byte is the parameter number and the data after words (length and content) are parameter specific as shown in Management Control Command Parameters/ Content.

Note: If the parameter that the BMC sets is not supported by the 82575. The 82575EB does not NACK the transaction. After the transaction ends, the 82575EB discards the data and asserts a transaction abort status.

The Management Control command format is as follows:

Function	Command	Byte Count	Data 1	Data 2	...	Data N
Management Control	0xC1	N	Parameter Number	Parameter Dependent		

Table 6. Management Control Command Parameters/Content

Parameter	#	Parameter Data
Keep PHY Link Up	0x00	A single byte parameter: Data 2: Bit 0: Set to indicate that the PHY link for this port should be kept up throughout system resets. This is useful when the server is reset and the BMC needs to keep connectivity for a manageability session. Bit [7:1] Reserved. 0b: Disabled. 1b: Enabled.

3.2.1.6 Update Management Receive Filter Parameters

This command is used to set the manageability receive filters parameters. The command is 0xCC. The first data byte is the parameter number and the data that follows (length and content) are parameter specific as listed in management RCV filter parameters.

Note: If the parameter that the BMC sets is not supported by the 82575, then the 82575EB does not NACK the transaction. After the transaction ends, the 82575EB discards the data and asserts a transaction abort status.

The update management RCV receive filter parameters command format is as follows:

Function	Command	Byte Count	Data 1	Data 2	...	Data N
Update Manageability Filter Parameters	0xCC	N	Parameter Number	Parameter Dependent		

Management RCV filter parameters lists the different parameters and their content.

Table 7. Management RCV Filter Parameters

Parameter	Number	Parameter Data
Filters Enables	0x1	Defines the generic filters configuration. The structure of this parameter is four bytes as the MANC register shown in the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual*. **Note:** The general filter enable is in the Receive Enable command that enables receive filtering.
Management-to-Host Configuration	0xA	This parameter defines which of the packet types identified as manageability packets in the receive path are directed to the host memory. Refer to the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual* for more details. Data 5:2 = MNG2H register bits.
Fail-Over Configuration	0xB	Fail-over register configuration. The bytes of this parameter are loaded into the Fail-Over Configuration register. See Section 3.3.3 for more information. Data 2:5 = Fail-Over Configuration register (bit 0 of Data 2 is bit 0 of the configuration register.).
Flex Filter 0 Enable Mask and Length	0x10	Flex Filter 0 Mask. Data 17:2 = Mask. Bit 0 in data 2 is the first bit of the mask. Data 19:18 = Reserved. Should be set to 00b. Date 20 = Flexible filter length.
Flex Filter 0 Data	0x11	Data 2 = Group of flex filter's bytes: 0x0 = bytes 0-29 0x1 = bytes 30-59 0x2 = bytes 60-89 0x3 = bytes 90-119 0x4 = bytes 120-127 Data 3:32 = Flex filter data bytes. Data 3 is LSB. Group's length is not a mandatory 30 bytes; it might vary according to filter's length and must NOT be padded by zeros.
Flex Filter 1 Enable Mask and Length	0x20	Same as parameter 0x10 but for filter 1.
Flex Filter 1 Data	0x21	Same as parameter 0x11 but for filter 1.
Flex Filter 2 Enable Mask and Length	0x30	Same as parameter 0x10 but for filter 2.
Flex Filter 2 Data	0x31	Same as parameter 0x11 but for filter 2.
Flex Filter 3 Enable Mask and Length	0x40	Same as parameter 0x10 but for filter 3.
Flex Filter 3 Data	0x41	Same as parameter 0x11 but for filter 3.
Filters Valid	0x60	Four bytes to determine which of the 82575EB filter registers contain valid data. Loaded into the MFVAL0 and MFVAL1 registers. Should be updated after the contents of a filter register are updated. Data 2: MSB of MFVAL. ... Data 5: LSB of MFVAL.
Decision Filters	0x61	Five bytes are required to load the manageability decision filters (MDEF). Data 2: Decision filter number. Data 3: MSB of MDEF register for this decision filter. ... Data 6: LSB of MDEF register for this decision filter.

VLAN Filters	0x62	Three bytes are required to load the VLAN tag filters. Data 2: VLAN filter number. Data 3: MSB of VLAN filter. Data 4: LSB of VLAN filter.
Flex Port Filters	0x63	Three bytes are required to load the manageability flex port filters. Data 2: Flex port filter number. Data 3: MSB of flex port filter. Data 4: LSB of flex port filter.
IPv4 Filters	0x64	Five bytes are required to load the IPv4 address filter. Data 2: IPv4 address filter number (3:0). Data 3: MSB of IPv4 address filter. ... Data 6: LSB of IPv4 address filter.
IPv6 Filters	0x65	17 bytes are required to load the IPv6 address filter. Data 2: IPv6 address filter number (3:0). Data 3: MSB of IPv6 address filter. ... Data 18: LSB of IPv6 address filter.
MAC Filters	0x66	Seven bytes are required to load the MAC address filters. Data 2: MAC address filters pair number (3:0). Data 3: MSB of MAC address. ... Data 8: LSB of MAC address.

3.2.2 Read Transactions (82575EB to BMC)

This section details the pass-through read transactions that the BMC can send to the 82575EB over the SMBus.

SMBus read transactions lists the different SMBus read transactions supported by the 82575. All the read transactions are compatible with SMBus read block protocol format.

Table 8. SMBus Read Transactions

TCO Command	Transaction	Command	Opcode	Fragments	Section
Receive TCO Packet	Block Read	0xD0 or 0xC0	First: 0x90 Middle: 0x10 Last[1]: 0x50	Multiple	3.2.2.1
Read Status	Block Read	0xD0 or 0xC0 or 0xDE	Single: 0xDD	Single	3.2.2.2
Get System MAC Address	Block Read	0xD4	Single: 0xD4	Single	3.2.2.3
Read Configuration	Block Read	0xD2	Single: 0xD2	Single	3.2.2.4
Read Management Parameters	Block Read	0xD1	Single: 0xD1	Single	3.2.2.5
Read Management RCV Filter Parameters	Block Read	0xCD	Single: 0xCD	Single	3.2.2.6
Read Receive Enable Configuration	Block Read	0xDA	Single: 0xDA	Single	3.2.2.7

1. The last fragment of the receive TCO packet is the packet status.

0xC0 or 0xD0 commands are used for more than one payload. If BMC issues these read commands, and the 82575EB has no pending data to transfer, it always returns as default opcode 0xDD with the 82575EB status and does not NACK the transaction.

3.2.2.1 Receive TCO LAN Packet Transaction

The BMC uses this command to read packets received on the LAN and its status. When the 82575EB has a packet to deliver to the BMC, it asserts the SMBus notification for the BMC to read the data (or direct receive). Upon receiving notification of the arrival of a LAN receive packet, the BMC begins issuing a Receive TCO packet command using the block read protocol.

A packet can be transmitted to the BMC in at least two fragments (at least one for the packet data and one for the packet status). As a result, BMC should follow the *F* and *L* bit of the op-code.

The op-code can have these values:

- 0x90 - First Fragment
- 0x10 - Middle Fragment
- When the opcode is 0x50, this indicates the last fragment of the packet, which contains packet status.

If a notification timeout is defined (in the EEPROM) and the BMC does not finish reading the whole packet within the timeout period, since the packet has arrived, the packet is silently discarded.

Following is the receive TCO packet format and the data format returned from the 82575.

Function	Command
Receive TCO Packet	0xC0 or 0xD0

Function	Byte Count	Data 1 (Op-Code)	Data 2	...	Data N
Receive TCO First Fragment	N	0x90	Packet Data Byte	...	Packet Data Byte
Receive TCO Middle Fragment	N	0x10	Packet Data Byte		
Receive TCO Last Fragment		0x50	Packet Data Byte		

3.2.2.1.1 Receive TCO LAN Status Payload Transaction

This transaction is the last transaction that the 82575EB issues when a packet received from the LAN is transferred to the BMC. The transaction contains the status of the received packet.

The format of the status transaction is as follows:

Function	Byte Count	Data 1 (Op-Code)	Data 2 – Data 17 (Status Data)
Receive TCO Long Status	17 (0x11)	0x50	See Below

The status is 16 bytes where byte 0 (bits 7:0) is set in Data 2 of the status and byte 15 in Data 17 of the status.

TCO LAN packet status data lists the content of the status data.

Table 9. TCO LAN Packet Status Data

Name	Bits	Description
Packet Length	13:0	Packet length including CRC, only 14 LSB bits.
Reserved	24:14	Reserved.
CRC	25	CRC Insert (CRC insertion is needed).
Reserved	28:26	Reserved.
VEXT	29	Additional VLAN present in packet.
VP	30	VLAN Stripped (VLAN TAG insertion is needed).
Reserved	33:31	Reserved.
Flow	34	TX/RX Packet (Packet Direction (0b = Rx, 1b = Tx).
LAN	35	LAN number.
Reserved	39:36	Reserved.
Reserved	47:40	Reserved.
VLAN	63:48	The two bytes of the VLAN header tag.
Error	71:64	See Error Status Information.
Status	79:72	See Status Info.
Reserved	87:80	Reserved.
MNG Status	127:88	This field should be ignored if Receive TCO is not enabled (see Management Status).

Bit descriptions of each field in can be found in the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual*.

Table 10. Error Status Information

Field	Bits	Description
RXE	7	RX Data Error
IPE	6	IPv4 Checksum Error
TCPE	5	TCP/UDP Checksum Error
CXE	4	Carrier Extension Error
Rsv	3	Reserved
SEQ	2	Sequence Error
SE	1	Symbol Error
CE	0	CRC Error or Alignment Error

Table 11. Status Info

Field	Bits	Description
UDPV	7	Checksum field is valid and contains checksum of UDP fragment header
IPIDV	6	IP Identification Valid
CRC32V	5	CRC 32 valid bit indicates that the CRC32 check was done and a valid result was found
Reserved	4	Reserved
IPCS	3	IPv4 Checksum Calculated on Packet
TCPCS	2	TCP Checksum Calculated on Packet
UDPCS	1	UDP Checksum Calculated on Packet
Reserved	0	Reserved

Table 12. Management Status

Name	Bits	Description
Pass RMCP 0x026F	0	Set when the UDP/TCP port of the manageability packet is 0x26F.
Pass RMCP 0x0298	1	Set when the UDP/TCP port of the manageability packet is 0x298.
Pass MNG Broadcast	2	Set when the manageability packet is a broadcast packet.
Pass MNG Neighbor	3	Set when the manageability packet neighbor discovery packet.
Pass ARP Request/ARP Response	4	Set when the manageability packet is ARP response/request packet.
Reserved	7:5	Reserved.
Pass MNG VLAN Filter Index	10:8	
MNG VLAN Address Match	11	Set when the manageability packet match one of the MNG VLAN filters.
Unicast Address Index	14:12	Match any of the four unicast MAC address.
Unicast Address Match	15	Match any of the four unicast MAC address.
L4 port Filter Index	22:16	Indicate the flex filter number.
L4 port Match	23	Match any of the UDP/TCP port filters.
Flex TCO Filter Index	26:24	If bit 27 is set, this field indicates which TCO filter was matched.
Flex TCO Filter Match	27	Set if a flexible filter matched.
IP Address Index	29:28	IP filter number. (IPv4 or IPv6).
IP Address Match	30	Match any of the IP address filters.
IPv4/IPv6 Match	31	IPv4 match or IPv6 match. This bit is valid only if the bit 30 (IP match bit) or bit 4 (ARP match bit) are set.
Decision Filter Match	39:32	Match decision filter.

3.2.2.2 Read Status Command

The BMC should use this command after receiving a notification from the 82575EB (such as SMBus Alert). The 82575EB also sends a notification to the BMC in either of the following two cases:

- The BMC asserts a request for reading the 82575EB status.
- The 82575EB detects a change in one of the Status Data 1 bits (and was set to send status to the BMC on status change) in the Receive Enable command.

Note: Commands 0xC0/0xD0 are for backward compatibility and can be used for other payloads. The 82575EB defines these commands in the opcode as well as which payload this transaction is. When the 0XDE command is set, the 82575EB always returns opcode 0XDD with the 82575EB status. The BMC reads the event causing the notification, using the Read Status command as follows:

The 82575EB response to one of the commands (0xC0 or 0xD0) in a given time as defined in the SMBus Notification Timeout and Flags word in the EEPROM.

Function	Command
Read Status	0XC0 or 0XD0 or 0XDE

Function	Byte Count	Data 1 (Op-Code)	Data 2 (Status Data 1)	Data 3 (Status Data 2)
Receive TCO Partial Status	3	0XDD	See Below	

Status Data Byte 1 lists the status data byte 1 parameters.

Table 13. Status Data Byte 1

Bit	Name	Description
7	Reserved	Reserved.
6	TCO Command Aborted	1b = A TCO command abort event occurred since the last read status cycle. 0b = A TCO command abort event did not occur since the last read status cycle.
5	Link Status Indication	0b = LAN link down. 1b = LAN link up[1].
4	PHY Link Forced Up	Contains the value of the *PHY_Link_Up* bit. When set, indicates that the PHY link is configured to keep the link up.
3	Initialization Indication	0b = An EEPROM reload event has not occurred since the last Read Status cycle. 1b = An EEPROM reload event has occurred since the last Read Status cycle[2].
2	Reserved	Reserved.
1:0	Power State	00b = Dr state. 01b = D0u state. 10b = D0 state. 11b = D3 state[3].

1. When the 82575EB is operating in teaming mode, and presented as one SMBus device, the link indication is 0b only when both links (on both ports) are down. If one of the LANs is disabled, its link is considered to be down.
2. This indication is asserted when the 82575EB manageability block reloads the EEPROM and its internal database is updated to the EEPROM default values. This is an indication that the external BMC should reconfigure the 82575, if other values other than the EEPROM default should be configured.
3. In single-address mode, the 82575EB reports the highest power-state modes in both devices. The "D" state is marked in this order: D0, D0u, Dr, and D3.

Status data byte 2 is used by the BMC to indicate whether the LAN device driver is alive and running.

The LAN device driver valid indication is a bit set by the LAN device driver during initialization; the bit is cleared when the LAN device driver enters a Dx state or is cleared by the hardware on a PCI reset.

Bits 2 and 1 indicate that the LAN device driver is stuck. Bit 2 indicates whether the interrupt line of the LAN function is asserted. Bit 1 indicates whether the LAN device driver dealt with the interrupt line before the last Read Status cycle. Table 14 lists status data byte 2.

Table 14. Status Data Byte 2

Bit	Name	Description
5	Reserved	Reserved.
4	Reserved	Reserved.
3	Driver Valid Indication	0b = LAN driver is not alive. 1b = LAN driver is alive.
2	Interrupt Pending Indication	1b = LAN interrupt line is asserted. 0b = LAN interrupt line is not asserted.
1	ICR Register Read/Write	1b = ICR register was read since the last read status cycle. 0b = ICR register was not read since the last read status cycle. Reading the ICR indicates that the driver has dealt with the interrupt that was asserted.
0	Reserved	Reserved

Note: When the 82575EB is in teaming mode, the bits listed in Status Data Byte 2 represent both cores:

- The LAN device driver alive indication is set if one of the LAN device drivers is alive.
- The LAN interrupt is considered asserted if one of the interrupt lines is asserted.
- The ICR is considered read if one of the ICRs was read (LAN 0 or LAN 1).

Status Data Byte 2 (bits 2 and 1) lists the possible values of bits 2 and 1 and what the BMC can assume from the bits:

Table 15. Status Data Byte 2 (Bits 2 and 1)

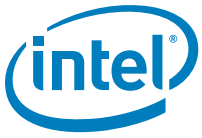
Previous	Current	Description
Don't Care	00b	Interrupt is not pending (OK).
00b	01b	New interrupt is asserted (OK).
10b	01b	New interrupt is asserted (OK).
11b	01b	Interrupt is waiting for reading (OK).
01b	01b	Interrupt is waiting for reading by the driver for more than one read cycle (not OK). Possible drive hang state.
Don't Care	11b	Previous interrupt was read and current interrupt is pending (OK).
Don't Care	10b	Interrupt is not pending (OK).

Note: The BMC reads should consider the time it takes for the LAN device driver to deal with the interrupt (in µs). Note that excessive reads by the BMC can give false indications.

3.2.2.3 Get System MAC Address

The Get System MAC Address returns the system MAC address over to the SMBus. This command is a single-fragment Read Block transaction that returns the following data:

Note: This command returns the MAC address configured in EEEPROM offset 0.



The format is as follows:

Function	Command
Get System MAC Address	0xD4

Data returned from the 82575:

Function	Byte Count	Data 1 (Op-Code)	Data 2	...	Data 7
Get System MAC Address	7	0xD4	MAC Address MSB	...	MAC Address LSB

3.2.2.4 Read Configuration

This command can be used by the BMC to read the CSR's from the manageability firmware.

In order to read a manageability CSR, the BMC executes two SMBus transactions. The first transaction is a block write that sets the CSR that the BMC needs to read. The second transaction is a block read that reads the CSR value.

The block write transaction is as follows:

Function	Command	Byte Count	Data 1	Data 2	Data 3
Write Configuration	0xC6	3	CSR Number MSB	...	CSR Number LSB

Following the block write, the BMC issues a block read that reads the parameter that was set in the Block Write command:

Function	Command
Read Configuration	0xD2

For example, if the BMC wants to read the MANC register, it would issue a block write with command of 0xC6, length of 3 and parameters of 0x00, 0x58, 0x20.

3.2.2.5 Read Management Parameters

In order to read the management parameters, the BMC executes two SMBus transactions. The first transaction is a block write that sets the parameter that the BMC needs to read. The second transaction is a block read that reads the parameter.

The block write transaction is as follows:

Function	Command	Byte Count	Data 1
Management Control Request	0xC1	1	Parameter Number

Following the block write, the BMC issues a block read that reads the parameter that was set in the Block Write command:

Function	Command
Read Management Parameter	0xD1

Data returned from the 82575:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data N
Read Management Parameter	N	0xD1	Parameter Number	Parameter Dependent		

The returned data is in the same format of the Management Control command.

Note that it might be that the parameter that is returned is not the parameter requested by the BMC. The BMC should verify the parameter number (default parameter to be returned is 0x10).

If the parameter number is 0xFF, it means that the data that the 82575EB should supply is not ready yet. The BMC should retry the read transaction.

It is the BMC responsibility to follow the previous procedure. In the case where the BMC sends a Block Read command, which is not preceded by a Block Write command with byte count = 1b, the 82575EB sets the parameter number in the read block transaction to be 0xFE.

3.2.2.6 Read Management Receive Filter Parameters

In order to read the Management RCV filter parameters, the BMC should execute two SMBus transactions. The first transaction is a block write that sets the parameter that the BMC wants to read. The second transaction is block read that read the parameter.

Following is the block write transaction:

Function	Command	Byte Count	Data 1	Data 2
Update MNG RCV Filter Parameters	0xCC	1 or 2	Parameter Number	Parameter Data

Following the block write, the BMC should issue a block read that reads the parameter that was set in the Block Write command as follows:

Parameter	#	Parameter Data
Filters Enable	0x01	None
MNG2H Configuration	0x0A	None
Fail-Over Configuration	0x0B	None
Flex Filter 0 Enable Mask and Length	0x10	None

Flex Filter 0 Data	0x11	Data 2: Group of Flex Filter's Bytes: 0x0 = bytes 0-29 0x1 = bytes 30-59 0x2 = bytes 60-89 0x3 = bytes 90-119 0x4 = bytes 120-127
Flex Filter 1 Enable Mask and Length	0x20	None
Flex Filter 1 Data	0x21	Same as parameter 0x11 but for filter 1.
Flex Filter 2 Enable Mask and Length	0x30	None
Flex Filter 2 Data	0x31	Same as parameter 0x11 but for filter 2.
Flex Filter 3 Enable Mask and Length	0x40	None
Flex Filter 3 Data	0x41	Same as parameter 0x11 but for filter 3.
Filters Valid	0x60	None
Decision Filters	0x61	One byte to define the accessed manageability decision filter (MDEF) Data 2 – Decision Filter number
VLAN Filters	0x62	One byte to define the accessed VLAN tag filter (MAVTV) Data 2 – VLAN Filter number
Flex Ports Filters	0x63	One byte to define the accessed manageability flex port filter (MFUTP). Data 2 – Flex Port Filter number
IPv4 Filter	0x64	One byte to define the accessed IPv4 address filter (MIPAF) Data 2 – IPv4 address filter number
Parameter	**#**	**Parameter Data**
IPv6 Filters	0x65	One byte to define the accessed IPv6 address filter (MIPAF) Data 2 – IPv6 address filter number
MAC Filters	0x66	One byte to define the accessed MAC address filters pair (MMAL, MMAH) Data 2 – MAC address filters pair number (0-3)

Following the block write, the BMC issues a block read that reads the parameter that was set in the Block Write command:

Function	Command
Request MNG RCV Filter Parameters	0xCD

Data returned from the 82575.

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data N
Read MNG RCV Filter Parameters	N	0xCD	Parameter Number	Parameter Dependent		

The returned data is the same format as the Update command.

Note that it might be that the parameter that is returned is not the parameter requested by the BMC. The BMC should verify the parameter number (default parameter to be returned is 0x1).

Note: If the parameter number is 0xFF then this indicates that the data the 82575EB should supply is not ready yet and the BMC should retry the read transaction.

It is the BMC responsibility to follow the procedure previously defined. In the case where the BMC sends a Block Read command that is not preceded by a Block Write command with bytecount = 1b, the 82575EB sets the parameter number in the read block transaction as 0xFE.

3.2.2.7 Read Receive Enable Configuration

The BMC uses this command to read the receive configuration data. This data can be configured when using Receive Enable command or through the EEPROM.

Read Receive Enable Configuration command format (SMBus Read Block) is as follows:

Function	Command
Read Receive Enable	0xDA

Data returned from the 82575:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data 8	Data 9	...	Data 12	Data 13	Data 14	Data 15
Read Receive Enable	15 (0x0F)	0xDA	Receive Control Byte	MAC Addr LSB	...	MAC Addr MSB	IP Addr LSB	...	IP Addr MSB	BMC SMBus Addr	I/F Data Byte	Alert Value Byte

3.3 LAN Fail-Over in LAN Teaming Mode

Manageability fail-over is the ability in a dual-port network device (the 82575) to detect that the LAN connection on the manageability enabled port is lost, and to enable the other port in the device to receive/transmit manageability packets. When the 82575EB operates in teaming mode, the OS and the external BMC consider the 82575EB as one logical network device. The decision to determine which of the 82575EB ports to use is done internally in the 82575EB (or in the ANS driver in case of the regular receive/transmit traffic). This section deals with fail-over in teaming mode only. In non-teaming mode, the external BMC should consider the 82575's network ports as two different network devices, and is solely responsible for the fail-over mechanism.

3.3.1 Fail-Over Functionality

In teaming mode, the 82575EB mirrors both the network ports into a single SMBus slave device. The 82575EB automatically handles the configurations of both network ports. Thus, for configurations, receiving and transmitting the BMC should consider both ports as a single entity.

When the currently active port for transmission becomes unavailable (for instance, the link is down), the 82575EB automatically tries to switch the packet transmission to the other port. Thus, as long as one of the ports is valid, the BMC has a valid link indication for the SMBus slave.

3.3.1.1 Transmit Functionality

In order to transmit a packet, the BMC should issue the appropriate SMBus packet transmission commands to the 82575. The 82575EB will then automatically choose the transmission port.

3.3.1.2 Receive Functionality

When the 82575EB receives a packet from any of the teamed ports, it will notifies and forwards the packet to the BMC.

Note: As both ports might be active (for instance, with a valid link), packets might be received on the currently non-active port. To avoid this, fail-over should be used only in a switched network.

3.3.1.3 Port Switching (Fail-Over)

While in teaming mode, transmit traffic is always transmitted by the 82575EB through only one of the ports at any given time. The 82575EB might switch the traffic transmission between ports under any of the following conditions:

1. The current transmitting port link is not available.
2. The preferred primary port is enabled and becomes available for transmission.

3.3.1.4 Device Driver Interactions

When the LAN device driver is present, the decision to switch between the two ports is done by the device driver. When the device driver is absent, this decision is done internally by the 82575.

Note: When the device driver releases teaming mode, such as when the system state changes, the 82575EB reconfigures the LAN ports to teaming mode. The 82575EB accomplishes this by re-setting the MAC address of the two ports to be the teaming address in order to re-start teaming. This is followed by transmitting gratuitous ARP packets to notify the network of teaming mode re-setting.

3.3.2 Fail-Over Configuration

Fail-over operation is configured through the fail-over register, as described in Table 16.

The BMC should configure this register after every initialization indication from the 82575EB (such as after every the 82575EB firmware reset). The BMC needs to use the Update Management Receive Filters command, with parameter 0x0A, detailed in Section 3.2.1.6.

The different configurations available to the BMC are detailed in this section.

Note: In teaming mode, both ports should be configured with the same receive manageability filters parameters (EEPROM sections for port 0 and port 1 should be identical).

3.3.2.1 Preferred Primary Port

The BMC might choose one of the network ports (LAN0 or LAN1) as a preferred primary port for packet transmission. The 82575EB uses the preferred primary port as the transmission port each time the link for that port is valid. For example, the 82575EB always switches back to the preferred primary port when available.

3.3.2.2 Gratuitous ARPs

In order to notify the link partner that a port switching has occurred, the 82575EB can be configured to automatically send gratuitous ARPs. These gratuitous ARPs cause the link partner to update its ARP tables to reflect the change.

The BMC might enable/disable gratuitous ARPs, configure the number of gratuitous ARPs, or the interval between them by modifying the fail-over register.

3.3.2.3 Link Down Timeout

The BMC can control the timeout for a link to be considered invalid. The 82575EB waits on this timeout before attempting to switch from an inactive port.

3.3.3 Fail-Over Register

This register is loaded at power up from the EEPROM (see the *82575EB Software Developer's Manual* for more information). The BMC can change the contents of the fail-over register using the Update Management Receive Filters command, with parameter 0x0A, detailed in Section 3.2.1.6.

Table 16 lists the register different bits:

Table 16. Fail-Over Register

Bits	Field	Initial Value	Read / Write	Description
0	RMP0EN	0x1	RO	RCV MNG port 0 Enable. When this bit is set, it reports that management traffic will be received from port 0.
1	RMP1EN	0x1	RO	RCV MNG port 1 Enable. When this bit is set, it reports that management traffic will be received from port 1.
2	MXP	0x0	RO	MNG XMT Port. 0b - reports that management traffic should be transmitted through port 0. 1b – reports that MNG traffic should be transmitted through port 1.
3	PRPP	0x0	RW	Preferred Primary Port. 0b – Port 0 is the preferred primary port. 1b – Port 1 is the preferred primary port.
4	PRPPE	0x0	RW	Preferred primary port enables.
5	Reserved	0x0	RO	Reserved
6	RGAEN	0x0		Repeated Gratuitous ARP Enable. If this bit is set, the 82575EB sends a configurable number of gratuitous ARP packets (GAC bits of this register) using configurable interval (GATI bits of this register) after the following events: • System move to Dx. • Fail-over event initiated the 82575.
8:7	Reserved	0x0	RO	Reserved
9	TFOENODX	0x0	RW	Teaming Fail-Over Enable on Dx. Enable fail-over mechanism. Bits 3:8 are valid only if this bit is set.
10:11	Reserved	0x0	RO	Reserved
12:15	GAC	0x0	RW	Gratuitous ARP Counter. Indicates the number of gratuitous ARP that should be sent after a fail-over event and after move to Dx. The value of 0b means that there is no limit on the gratuitous ARP packets to be sent.
16:23	LDFOT	0x0	RW	Link down Fail-Over Time. Defines the time (in seconds) the link should be down before doing a fail-over to the second port. This is also the time that the primary link should be up (after it was down) before the 82575EB will fail-over back to the primary port.
24:31	GATI	0x0	RW	Gratuitous ARP Transmission Interval. Defines the interval in seconds before retransmission of gratuitous ARP packets.

3.4 SMBus Troubleshooting Guide

This section outlines the most common issues found while working with pass-through using the SMBus sideband interface.

3.4.1 TCO Alert Line Stays Asserted After a Power Cycle

After the 82575EB resets both of its ports indicates a status change. If the BMC only reads status from one port (slave address) the other one will continue to assert the TCO alert line.

Ideally, the BMC should use the ARA transaction (see Section 3.1) to determine which slave asserted the TCO alert. Many customers only wish to use one port for manageability thus using ARA might not be optimal.

An alternate to using ARA is to configure one of the ports to not report status and to set its SMBus timeout period. In this case, the SMBus timeout period determines how long a port asserts the TCO alert line awaiting a status read from a BMC; by default this value is zero, which indicates an infinite timeout.

The SMBus configuration section of the EEPROM has a SMBus Notification Timeout (ms) field that can be set to a recommended value of 0xFF (for this issue). Note that this timeout value is for both slave addresses. Along with setting the SMBus Notification Timeout to 0xFF, it is recommended that the second port be configured in the EEPROM to disable status alerting. This is accomplished by having the *Enable Status Reporting* bit set to 0b for the desired port in the LAN configuration section of the EEPROM.

The last solution for this issue is to have the BMC hard-code the slave addresses to always read from both ports. As with the previous solution, it is also recommend that the second port have status reporting disabled.

3.4.2 SMBus Commands are Always NACK'd by the 82575

There are several reasons why all commands sent to the 82575EB from a BMC could be NACK'd. The following are the most common:

- Invalid EEPROM Image - The image itself might be invalid, or it could be a valid image; however, it is not a pass-through image, as such SMBus connectivity is disabled.
- The BMC is not using the correct SMBus address - Many BMC vendors hard-code the SMBus address(es) into their firmware. If the incorrect values are hard-coded, the 82575EB does not respond.
  - The SMBus address(es) can also be dynamically set using the SMBus ARP mechanism.
- The BMC is using the incorrect SMBus interface - The EEPROM might be configured to use one physical SMBus port; however, the BMC is physically connected to a different one.
- Bus Interference - the bus connecting the BMC and the 82575EB might be unstable.

3.4.3 SMBus Clock Speed is 16.6666 KHz

This can happen when the SMBus connecting the BMC and the 82575EB is also tied into another device (such as a ICH6) that has a maximum clock speed of 16.6666 KHz. The solution is to not connect the SMBus between the 82575EB and the BMC to this device.

3.4.4 A Network Based Host Application is not Receiving any Network Packets

Reports have been received about an application not receiving any network packets. The application in question was NFS under Linux. The problem was that the application was using the RMPC/RMCP+ IANA reserved port 0x26F (623), and the system was also configured for a shared MAC and IP address with the OS and BMC.

The management control to host configuration, in this situation, was setup not to send RMCP traffic to the OS (this is typically the correct configuration). This means that no traffic send to port 623 was being routed.

The solution in this case is to configure the problematic application NOT to use the reserved port 0x26F.

3.4.5 Status Registers

If the EEPROM image is configured correctly, the physical connections are valid, and problems still exist, use LANConf or other utilities/drivers to check the appropriate 82575EB status registers for other indications.

3.4.5.1 Firmware Semaphore Register (FWSM, 0x5B54)

This register (described in the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual*) provides a way to find out if the firmware on the 82575EB is functioning properly and if so, in what mode.

Check the error indication bits (24:19), if they are anything other than zero, then the firmware is not going to be fully functional, if at all.

The most common errors are:

- EEPROM checksum errors - these can be caused by a number of things:
  - Mismatch in 82575EB stepping and EEPROM image version (old EEPROM image on a new 82575)
  - EEPROM part too small (recommended minimum size for manageability is 32 Kb)
  - Old utility was used to update the EEPROM (always make sure to have the latest versions)
- Invalid Firmware Mode (0x08)

If bits 3:1 of the register indicate a firmware mode that is reserved, this error condition can be reset.

Always make note of the firmware mode, bits 3:1. In nearly all cases, this value should be set to 010b for pass-through mode to an external BMC.

The firmware valid bit (15) should be set to 1b to indicate that the firmware is up and running. If it is not set to 1b, then an error code should be indicated in bits 24:19.

The reset count bits (18:16) indicate how many times the internal firmware on the 82575EB has been reset. This value should be a one (the firmware was reset at power up). If the value is greater than one then there are issues somewhere. Note that this counter goes from 0-7 and wraps around.

3.4.5.2 Management Control Register (MANC 0x5820)

This register is described in detail in the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual*.

This register indicates which filters are enabled. It is possible to configure all of the filters yet not enable them, in which case, no management traffic is routed to the BMC. Or, the BMC might be receiving undesired traffic, such as ARP requests when the 82575EB was configured to do automatic ARP responses.

Check this register if getting unwanted traffic or if packets aren't getting sent to the BMC.

Bit 17 (*Receive TCO Packets Enable*) must also be set in order for any packets are sent to a BMC. Note that it doesn't matter what the other enabled filters are, if this one is off, no packets are sent to the BMC.

Bit 21 (*Enable Management-to-Host*) enables or disables the various filters that also allow manageability traffic (all those that pass the filters in the 82575) to optionally be passed to the OS.

3.4.5.3 Management Control To Host Register (MANC2H 0x5860)

This register is described in detail in the *Intel® 82575EB Gigabit Ethernet Controller Software Developer's Manual*.

The 82575EB has a large number of filtering mechanisms by which network traffic can be directed to a BMC. Traffic sent to the BMC is typically not sent to the host OS as well. For example, the OS is not interested in RMCP/RMCP+ traffic. However, the OS might be interested in ARP requests. If the 82575EB is configured for automatic ARP requests, this means that a filter for ARP requests has been configured and enabled, if the ARP request matches that of the BMC, then that ARP request is not sent to the OS unless specifically configured to do so. This is what the MANC2H register shows, which manageability filters to also pass the data up to the OS as well as the BMC.

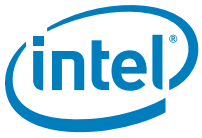
Using the ARP request example; typically it is desirable to allow the OS to receive these, as such, bit 7 (*ARP Request*) should be set.

3.4.6 Unable to Transmit Packets from the BMC

If the BMC has been transmitting and receiving data without issue for a period of time and then begins to receive NACKs from the 82575EB when it attempts to write a packet, the problem is most likely due to the fact that the buffers internal to the 82575EB are full of data that has been received from the network; however, has yet to be read by the BMC.

Being an embedded device, the 82575EB has limited buffers, which it shares for receiving and transmitting data. If a BMC does not keep the incoming data read, the 82575EB can be filled up, which does not allow the BMC to transmit anymore data, resulting in NACKs.

If this situation occurs, the recommended solution is to have the BMC issue a Receive Enable command to disable anymore incoming data, go read all the data from the 82575EB and then use the Receive Enable command to enable incoming data once again.



3.4.7 SMBus Fragment Size

The SMBus specification indicates a maximum SMBus transaction size of 32 bytes. Most of the data passed between the 82575EB and the BMC over the SMBus is RMCP/RMCP+ traffic, which by its very nature (UDP traffic) is significantly larger than 32 bytes in length, thus requiring multiple SMBus transactions to move a packet from the 82575EB to the BMC or to send a packet from the BMC to the 82575.

Recognizing this bottleneck, the 82575EB can handle up to 240 bytes of data within a single transaction. This is a configurable setting within the EEPROM.

The default value in the EEPROM images is 32, per the SMBus specification. If performance is an issue, it is recommended that you increase this size.

During the initialization phase, the firmware within the 82575EB allocates buffers based upon the SMBus fragment size setting within the EEPROM. The 82575EB firmware has a finite amount of RAM for its use, as such the larger the SMBus fragment size, the fewer buffers it can allocate. As such, the BMC implementation must take care to send data over the SMBus in an efficient way.

For example, the 82575EB firmware has 3 KB of RAM it can use for buffering SMBus fragments. If the SMBus fragment size is 32 bytes then the firmware could allocate 96 buffers of size 32 bytes each. As a result, the BMC could then send a large packet of data (such as KVM) that is 800 bytes in size in 25 fragments of size 32 bytes apiece.

However, this might not be the most efficient way because the BMC must break the 800 bytes of data into 25 fragments and send each one at a time.

If the SMBus fragment size is changed to 240 bytes, the 82575EB firmware can create 12 buffers of 240 bytes each to receive SMBus fragments. The BMC can now send that same 800 bytes of KVM data in only four fragments, which is much more efficient.

The problem of changing the SMBus fragment size in the EEPROM is if the BMC does not also reflect this change. If a programmer changes the SMBus fragment size in the 82575EB to 240 bytes and then wants to send 800 bytes of KVM data, the BMC can still only send the data in 32 byte fragments. As a result, the firmware runs out of memory.

This is because the 82575EB firmware created the 12 buffers of 240 bytes each for fragments, however the BMC is only sending fragments of size 32 bytes. This results in a memory waste of 208 bytes per fragment in this case, and when the BMC attempts to send more than 12 fragments in a single transaction, the 82575EB NACKs the SMBus transaction due to not enough memory to store the KVM data.

In summary, if a programmer increases the size of the SMBus fragment size in the EEPROM, which is recommended for efficiency purposes, take care to ensure that the BMC implementation reflects this change and uses that fragment size to its fullest when sending SMBus fragments.

3.4.8 Enable XSum Filtering

If XSum filtering is enabled, the BMC does not need to perform the task of checking this checksum for incoming packets. Only packets that have a valid XSum is passed to the BMC, all others are silently discarded.

This is a way to offload some work from the BMC.

3.4.9 Still Having Problems?

If problems still exist, contact your field representative. Before contacting, be prepared to provide the following:

- The contents of status registers:
  - 0x5820
  - 0x5860
  - 0x5B54
- A SMBus trace if possible
- A dump of the EEPROM image
  - This should be taken from the actual 82575, rather than the EEPROM image provided by Intel. Parts of the EEPROM image are changed after writing, such as the physical EEPROM size. This information could be key in helping assist in solving an issue.

4.0 NC-SI Interface

The Network Controller Sideband Interface (NC-SI) is a DMTF industry standard protocol for the sideband interface. NC-SI uses a modified version of the industry standard RMII interface for the physical layer as well as defining a new logical layer.

The NC-SI specification can be found at the DMTF website at:

http://www.dmtf.org/apps/org/worgroup/pos_sb/

4.1 Overview

4.1.1 Terminology

The terminology in this document is taken directly from the NC-SI specification and is as follows:

Term	Definition
Frame Versus Packet	Frame is used in reference to Ethernet, whereas packet is used everywhere else.
External Network Interface	The interface of the network controller that provides connectivity to the external network infrastructure (port).
Internal Host Interface	The interface of the network controller that provides connectivity to the host OS running on the platform.
Management Controller (MC)	An intelligent entity comprising of HW/FW/SW, that resides within a platform and is responsible for some or all management functions associated with the platform (BMC, service processor, etc.).
Network Controller (NC)	The component within a system that is responsible for providing connectivity to the external Ethernet networked world.
Remote Media	The capability to allow remote media devices to appear as if they were attached locally to the host.
Network Controller Sideband Interface	The interface of the network controller that provides connectivity to a management controller. It can be shorten to sideband interface as appropriate in the context.
Interface	This refers to the entire physical interface, such as both the transmit and receive interface between the management controller and the network controller.
Integrated Controller	The term integrated controller refers to a network controller device that supports two or more channels for NC-SI that share a common NC-SI physical interface. For example, a network controller that has two or more physical network ports and a single NC-SI bus connection.
Multi-Drop	Multi-drop commonly refers to the case where multiple physical communication devices share an electrically common bus and a single device acts as the master of the bus and communicates with multiple slave or target devices. In NC-SI, a management controller serves the role as the master, and the network controllers are the target devices.

Point-to-Point	Point-to-point commonly refers to the case where only two physical communication devices are interconnected via a physical communication medium. The devices might be in a master/slave relationship, or could be peers. In NC-SI, point-to-point operation refers to the situation where only a single management controller and single network controller package are used on the bus in a master/slave relationship where the management controller is the master.
Channel	The control logic and data paths supporting NC-SI pass-through operation on a single network interface (port). A network controller that has multiple network interface ports can support an equivalent number of NC-SI channels.
Package	One or more NC-SI channels in a network controller that share a common set of electrical buffers and common buffer control for the NC-SI bus. Typically, there will be a single, logical NC-SI package for a single physical network controller package (chip or module). However, the specification allows a single physical chip or module to hold multiple NC-SI logical packages.
Control Traffic/Messages/Packets	Command, response and notification packets transmitted between MC and NCs for the purpose of managing NC-SI.
Pass-Through Traffic/Messages/Packets	Non-control packets passed between the external network and the MC through the NC.
Channel Arbitration	Refer to operations where more than one of the network controller channels can be enabled to transmit pass-through packets to the MC at the same time, where arbitration of access to the RXD, CRS_DV, and RX_ER signal lines is accomplished either by software of hardware means.
Logically Enabled/Disabled NC	Refers to the state of the network controller wherein pass-through traffic is able/ unable to flow through the sideband interface to and from the management controller, as a result of issuing Enable/Disable Channel command.
NC RX	Defined as the direction of ingress traffic on the external network controller interface
NC TX	Defined as the direction of egress traffic on the external network controller interface
NC-SI RX	Defined as the direction of ingress traffic on the sideband enhanced NC-SI Interface with respect to the network controller.
NC-SI TX	Defined as the direction of egress traffic on the sideband enhanced NC-SI Interface with respect to the network controller.

4.1.2 System Topology

In NC-SI each physical endpoint (NC package) can have several logical slaves (NC channels).

NC-SI defines that one management controller and up to four network controller packages can be connected to the same NC-SI link.

Figure 7 shows an example topology for a single MC and a single NC package. In this example the NC package has two NC channels.

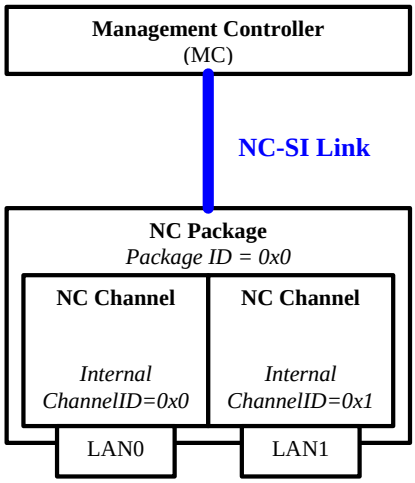


Figure 7. Single NC Package, Two NC Channels

Figure 8 shows an example topology for a single MC and two NC packages. In this example, one NC package has two NC channels and the other has only one NC channel.

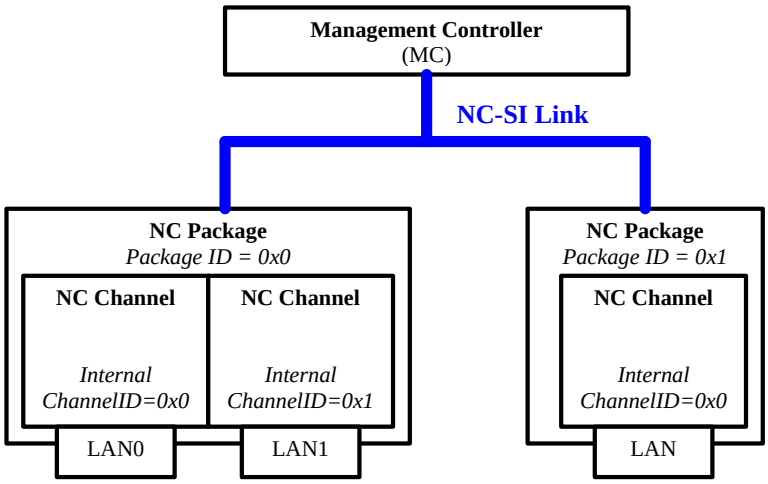


Figure 8. Two NC Packages (Left, with Two NC Channels and Right, with One NC Channel)

Scenarios in which the NC-SI lines are shard by multiple NCs (as shown in Figure 8) mandate an arbitration mechanism. The arbitration mechanism is described in Section 4.5.1.

4.1.3 Data Transport

Since NC-SI uses the RMII transport layer, data is transferred in the form of Ethernet frames.

NC-SI defines two types of frames transmitted on the NC-SI interface:

1. Control frames:
   a. Frames used to configure and control the interface.
   b. Control frames are identified by a unique EtherType in their L2 header.
2. Pass-through frames:
   a. The actual LAN pass-through frames transferred from/to the MC.
   b. Pass-through frames are identified as not being a control frame.
   c. Pass-through frames are attributed to a specific NC channel by their source MAC address (as configured in the NC by the MC).

4.1.3.1 Control Frames

NC-SI control frames are identified by a unique NC-SI EtherType (0x88F8).

Control frames are used in a single-threaded operation, meaning commands are generated only by the MC and can only be sent one at a time. Each command from the MC is followed by a single response from the NC (command-response flow), after which the MC is allowed to send a new command.

The only exception to the command-response flow is the Asynchronous Event Notification (AEN). These control frames are sent unsolicited from the NC to the MC.

Note: AEN functionality by the NC must be disabled by default, until activated by the MC using the Enable AEN commands.

In order to be considered a valid command, the control frame must:

1. Comply with the NC-SI header format.
2. Be targeted to a valid channel in the package via the *Package ID* and *Channel ID* fields.

For example, to target a NC channel with package ID of 0x2 and internal channel ID of 0x5, The MC must set the channel ID inside the control frame to 0x45.

Note: Channel ID is composed of three bits of package ID and five bits of internal channel ID.

1. Contain a correct payload checksum (if used).
2. Meet any other condition defined by NC-SI.

Note: There are also commands (such as select package) targeted to the package as a whole. These commands must use an internal channel ID of 0x1F.

For more details, refer to the NC-SI specification.

4.1.3.2 NC-SI Frames Receive Flow

Figure 9 shows the overall flow for frames received on the NC from the MC.

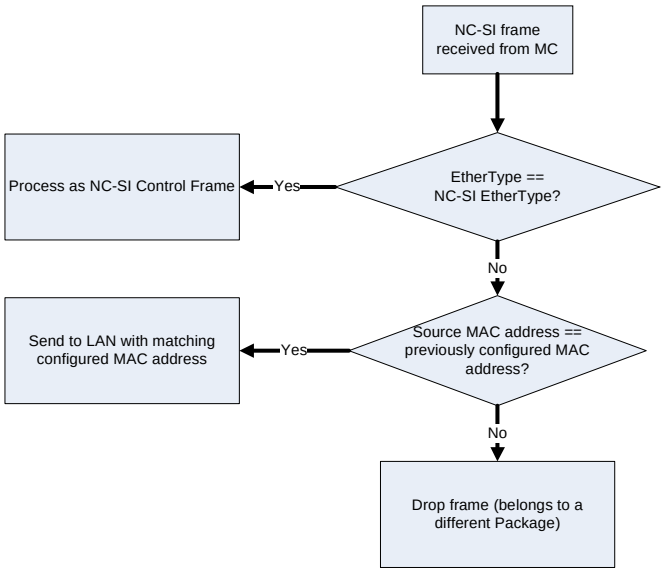


Figure 9. NC-SI Frames Receive Flow for the NC

4.2 NC-SI Support

4.2.1 Supported Features

The 82575EB supports the all the mandatory features of the NC-SI specification. Table 17 lists the supported commands.

Table 17. Supported NC-SI commands

Command	Supported?
Clear Initial State	Yes
Get Version ID	Yes
Get Parameters	Yes
Get Controller Packet Statistics	Yes, partially
Get Link Status	Yes
Enable Channel	Yes
Disable Channel	Yes
Reset Channel	Yes
Enable VLAN	Yes. The 82575EB does not support filtering of User priority/CFI Bits of VLAN
Disable VLAN	Yes
Enable Broadcast	Yes
Disable Broadcast	Yes
Set MAC Address	Yes
Clear MAC Address	Yes

Table 17. Supported NC-SI commands

Get NC-SI Statistics	Yes, partially
Enable NC-SI Flow-Control	No
Disable NC-SI Flow-Control	No
Set Link Command	Yes
Enable Global Multicast Filter	Yes
Disable Global Multicast Filter	Yes
Get Capabilities	Yes
Set VLAN Filters	Yes
AEN Enable	Yes
Get Pass-Through Statistics	Yes, partially
Select Package	Yes
Deselect Package	Yes
Enable Channel Network TX	Yes
Disable Channel Network TX	Yes
OEM Command	Yes

Table 18 lists the different capabilities and parameters that are advertised by the 82575EB (per each NC-SI channel):

Table 18. NC-SI Capabilities Advertisement

Feature	Capability	Details
Capabilities Flags	Hardware arbitration	Supported
	OS presence	Supported
	Network controller to management controller flow control support	Not supported
	Management Controller to Network Controller Flow Control Support	Not supported
	All multicast addresses support	Supported
Broadcast Filtering	ARP	Supported
	DHCP client	Supported
	DHCP Server	Supported
	NetBIOS	Supported
Multicast Filtering	IPv6 neighbor advertisement	Supported
	IPv6 router advertisement	Supported
	DHCPv6 relay and server multicast	Supported
Buffering Capabilities	LAN receive buffer size	8Kbytes
AEN Support		All AENs are supported
MAC Filtering	Unicast filters	0
	Multicast filters	0
	Mixed filters	2
VLAN Filtering	VLAN filters	8
VLAN Mode Support	Supported modes	Modes 1 and 3
Channel Count	Number of channels in the package	2 channels
Version ID	NC-SI version	1.0
	FW name	The 82575EB ROM Verified
	FW version	0x20D0002
	PCI DID	0x10A7
	PCI VID	0x8086
	Manufacturer ID (IANA)	0x157 (Intel)

Table 19 lists the optional features supported:

Table 19. Optional NC-SI Features Support

Feature	Supported	Details
AENs	Yes	**Note:** The driver state AEN might be emitted up to 15 seconds after actual driver change.
Get Controller Packet Statistics Command	No	
Get NC-SI Statistics Command	Yes, partially	Supports the following counters: 1-4, 7.
Get NC-SI Pass-Through Statistics Command	Yes, partially	Supports the following counters: 2 Support the following counters only when the OS is down: 1, 6, 7.
VLAN Modes	Yes, partially	Supports only modes 1, 3.
MAC Address Filters	Yes	Supports two MAC addresses as mixed per port.
Channel Count	Yes	Supports two channels.
VLAN Filters	Yes	Supports eight VLAN filters per port.
Broadcast Filters	Yes	Supports the following filters: • ARP. • DHCP. • NetBIOS.
Multicast Filters	Yes	Supports the following filters: • IPv6 neighbor advertisement. • IPv6 router advertisement. • DHCPv6 relay and server multicast.
NC-SI Flow Control Command	No	Does not support NC-SI flow-control.
HW Arbitration	No	Does not support NC-SI HW arbitration.
Allow Link Down	No	Does not support shutting down the link when disabled.

4.2.2 NC-SI Mode - Intel Specific Commands

In addition to the regular NC-SI commands, the following Intel vendor specific commands are supported. The purpose of these commands is to provide a means for the Baseboard Management Controller (BMC) to access some of the Intel-specific features present in the 82575.

4.2.2.1 Overview

The following features are available via the NC-SI OEM specific command:

- Receive filters:
  - Packet Addition Decision Filters 0x0...0x4
  - Packet Reduction Decision Filters 0x5...0x7
  - MNG2HOST register (controls the forwarding of manageability packets to the host)
  - Flex 128 filters 0x0...0x3
  - Flex TCP/UDP port filters 0x0...0xA
  - IPv4/IPv6 filters
- Get System MAC Address - This command allows the MC to retrieve the system MAC address used by the NC. This MAC address can be used for a shared MAC address mode.

- Keep Phy Link Up (*Veto* bit) Enable/Disable - This feature enables the BMC to block Phy reset, which might cause session loss.
- TCO Reset - Allows the MC to reset the 82575.
- Checksum offloading - Offloads IP/UDP/TCP checksum checking from the MC.

These commands are designed to be compliant with their corresponding SMBus commands (if existing).

All of the commands are based on a single DMTF defined NC-SI command, known as OEM Command. This command is as follows.

4.2.2.1.1 OEM Command (0x50)

The OEM command can be used by the MC to request the sideband interface to provide vendor-specific information. The Vendor Enterprise Number (VEN) is the unique MIB/SNMP private enterprise number assigned by IANA per organization. Vendors are free to define their own internal data structures in the vendor data fields

.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..	Intel Command Number	Optional Data		

Figure 10. OEM Command Packet Format

4.2.2.1.2 OEM Response (0xD0)

Below is the vendor specific format for commands, as defined by NC-SI.

	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 (Intel 0x157)			
24..27	Intel Command Number	Optional Return Data		

Figure 11. OEM Response Packet Format

4.2.2.1.3 OEM Specific Command Response Reason Codes

Response Code		Reason Code	
Value	Description	Value	Description
0x1	Command Failed	0x5081	Invalid Intel Command Number
0x1	Command Failed	0x5082	Invalid Intel Command Parameter Number
0x1	Command Failed	0x5085	Internal Network Controller Error
0x1	Command Failed	0x5086	Invalid Vendor Enterprise Code

Table 20. Commands Summary

Intel Command	Parameter	Command Name
0x00	0x00	Set IP Filters Control
0x01	0x00	Get IP Filters Control
0x02	0x0A	Set Manageability to Host
	0x10	Set Flexible 128 Filter 0 Mask and Length
	0x11	Set Flexible 128 Filter 0 Data
	0x20	Set Flexible 128 Filter 1 Mask and Length
	0x21	Set Flexible 128 Filter 1 Data
	0x30	Set Flexible 128 Filter 2 Mask and Length
	0x31	Set Flexible 128 Filter 2 Data
	0x40	Set Flexible 128 Filter 3 Mask and Length
	0x41	Set Flexible 128 Filter 3 Data
	0x61	Set Packet Addition Filters
	0x63	Set Flex TCP/UDP Port Filters
	0x64	Set Flex IPv4 Address Filters
	0x65	Set Flex IPv6 Address Filters
0x3	0x0A	Get Manageability to Host
	0x10	Get Flexible 128 Filter 0 Mask and Length
	0x11	Get Flexible 128 Filter 0 Data
	0x20	Get Flexible 128 Filter 1 Mask and Length
	0x21	Get Flexible 128 Filter 1 Data
	0x30	Get Flexible 128 Filter 2 Mask and Length
	0x31	Get Flexible 128 Filter 2 Data
	0x40	Get Flexible 128 Filter 3 Mask and Length
	0x41	Get Flexible 128 Filter 3 Data
	0x61	Get Packet Addition Filters
	0x63	Get Flex TCP/UDP Port Filters
	0x64	Get Flex IPv4 Address Filters
	0x65	Get Flex IPv6 Address Filters
0x04	0x00	Set Unicast Packet Reduction
	0x01	Set Multicast Packet Reduction
	0x02	Set Broadcast Packet Reduction
0x05	0x00	Get Unicast Packet Reduction
	0x01	Get Multicast Packet Reduction
	0x02	Get Broadcast Packet Reduction
0x06	N/A	Get System MAC Address
0x20	N/A	Set Intel Management Control
0x21	N/A	Get Intel Management Control
0x22	N/A	Perform TCO Reset
0x23	N/A	Enable IP/UDP/TCP Checksum Offloading
0x24	N/A	Disable IP/UDP/TCP Checksum Offloading

4.2.2.2 Proprietary Commands Format

4.2.2.2.1 Set Intel Filters Control Command (Intel Command 0x00)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x00	Filter Control Index		

4.2.2.2.2 Set Intel Filters Control Response Format (Intel Command 0x00)

	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 (Intel 0x157)			
24..27	0x00	Filter Control Index		

4.2.2.3 Set Intel Filters Control - IP Filters Control Command (Intel Command 0x00, Filter Control Index 0x00)

This command controls different aspects of the Intel filters.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x00	0x00	IP Filters control (3-2)	
24..27	IP Filters Control (1-0)			

Where “IP Filters Control” has the following format:

Bit #	Name	Description	Default Value
0	IPv4/IPv6 Mode	IPv6 (0b): There are zero IPv4 filters and four IPv6 filters IPv4 (1b): There are four IPv4 filters and three IPv6 filters	1b
1..15	Reserved		

16	IPv4 Filter 0 Valid	Indicates if the IPv4 address configured in IPv4 address 0 is valid.	0b **Note:** The network controller automatically sets this bit to 1b if the Set Intel Filter – IPv4 Filter Command is used for filter zero.
17	IPv4 Filter 1 Valid	Indicates if the IPv4 address configured in IPv4 address 1 is valid.	0b **Note:** The network controller automatically sets this bit to 1b if the Set Intel Filter – IPv4 Filter Command is used for filter one.
18	IPv4 Filter 2 Valid	Indicates if the IPv4 address configured in IPv4 address 2 is valid.	0b **Note:** The network controller automatically sets this bit to 1b if the Set Intel Filter – IPv4 Filter Command is used for filter two.
19	IPv4 Filter 3 Valid	Indicates if the IPv4 address configured in IPv4 address 3 is valid.	0b **Note:** The network controller automatically sets this bit to 1b if the Set Intel Filter – IPv4 Filter Command is used for filter three.
20..23	Reserved		
24	IPv6 Filter 0 Valid	Indicates if the IPv6 address configured in IPv6 address 0 is valid.	0b **Note:** The network controller automatically sets this bit to 1b if the Set Intel Filter – IPv6 Filter Command is used for filter zero.
25	IPv6 Filter 1 Valid	Indicates if the IPv6 address configured in IPv6 address 1 is valid.	0b **Note:** The network controller automatically sets this bit to 1b if the Set Intel Filter – IPv6 Filter Command is used for filter one.
26	IPv6 Filter 2 Valid	Indicates if the IPv6 address configured in IPv6 address 2 is valid.	0b **Note:** The network controller automatically sets this bit to 1b if the Set Intel Filter – IPv6 Filter Command is used for filter two.
27	IPv6 Filter 3 Valid	Indicates if the IPv6 address configured in IPv6 address 3 is valid.	0b **Note:** The network controller automatically sets this bit to 1b if the Set Intel Filter – IPv6 Filter Command is used for filter three.
28..31	Reserved		

4.2.2.3.1 Set Intel Filters Control - IP Filters Control Response (Intel Command 0x00, Filter Control Index 0x00)

	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 (Intel 0x157)			
24..27	0x00	0x00		

4.2.2.4 Get Intel Filters Control Command (Intel Command 0x01)

4.2.2.4.1 Get Intel Filters Control - IP Filters Control Command (Intel Command 0x01, Filter Control Index 0x00)

This command controls different aspects of the Intel filters.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x01	0x00		

4.2.2.4.2 Get Intel Filters Control - IP Filters Control Response (Intel Command 0x01, Filter Control Index 0x00)

	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 (Intel 0x157)			
24..27	0x01	0x00	IP Filters Control (3-2)	
28..29	IP Filters Control (1-0)			

4.2.2.5 Set Intel Filters Formats

4.2.2.5.1 Set Intel Filters Command (Intel Command 0x02)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x02	Parameter Number	Filters Data (optional)	

4.2.2.5.2 Set Intel Filters Response (Intel Command 0x02)

	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 (Intel 0x157)			
24..	0x02	Filter Control Index	Return Data (Optional)	

4.2.2.5.3 Set Intel Filters - Manageability to Host Command (Intel Command 0x02, Filter Parameter 0x0A)

This command sets the Mng2Host register. The Mng2Host register controls whether pass-through packets destined to the BMC are also be forwarded to the host OS.

The Mng2Host register has the following structure:

Bits	Description	Default
0	Decision Filter 0	Determines if packets that have passed Decision Filter 0 is also forwarded to the host OS.
1	Decision Filter 1	Determines if packets that have passed Decision Filter 1 is also forwarded to the host OS.
2	Decision Filter 2	Determines if packets that have passed Decision Filter 2 is also forwarded to the host OS.
3	Decision Filter 3	Determines if packets that have passed Decision Filter 3 is also forwarded to the host OS.
4	Decision Filter 4	Determines if packets that have passed Decision Filter 4 is also forwarded to the host OS.
5	Unicast and Mixed	Determines if broadcast packets are also forwarded to the host OS.
6	Global Multicast	Determines if unicast and mixed packets are also forwarded to the host OS.
7	Broadcast	Determines if global multicast packets are also forwarded to the host OS.

	Bits			
Bytes	31..24	23..16	15..08	07..00
1.1 00..15	1.2 NC-SI Header			
1.3 16..19	1.4 Manufacturer ID (Intel 0x157)			
1.5 20..23	1.6 0x02	1.7 0x0A	1.8 Manageability to Host (3-2)	
1.9 24..25	1.10 Manageability to Host (1-0)		1.11	

4.2.2.5.4 Set Intel Filters - Manageability to Host Response (Intel Command 0x02, Filter Parameter 0x0A)

	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 (Intel 0x157)			
24..	0x02	0x0A		

4.2.2.5.5 Set Intel Filters - Flex Filter 0 Enable Mask and Length Command (Intel Command 0x02, Filter Parameter 0x10/0x20/0x30/0x40)

The following command sets the Intel flex filters mask and length. Use filter parameters 0x10/0x20/ 0x30/0x40 for flexible filters 0/1/2/3 accordingly.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x10/ 0x20/ 0x30/ 0x40	Mask Byte 1	Mask Byte 2
24..27	..	..	..	..
28...31	..	..	..	..
32...35	..	..	..	..
35...37	..	Mask Byte 16	Reserved	Reserved
38	Flexible Filter Length			

4.2.2.5.6 Set Intel Filters - Flex Filter 0 Enable Mask and Length Response (Intel Command 0x02, Filter Parameter 0x10/0x20/0x30/0x40)

	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 (Intel 0x157)			
24..25	0x02	0x10/ 0x20/ 0x30/ 0x40		

4.2.2.5.7 Set Intel Filters - Flex Filter 0 Data Command (Intel Command 0x02, Filter Parameter 0x11/0x21/0x31/0x41)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..	0x02	0x11/ 0x21/ 0x31/ 0x41	Filter Data Group	Filter Data 1
	..	Filter Data N		

4.2.2.5.8 Set Intel Filters - Flex Filter 0 Data Response (Intel Command 0x02, Filter Parameter 0x11/0x21/0x31/0x41)

	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 (Intel 0x157)			
24..25	0x02	0x11/ 0x21/ 0x31/ 0x41		

4.2.2.5.9 Set Intel Filters - Packet Addition Decision Filter Command (Intel Command 0x02, Filter Parameter 0x61)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x61	Decision Filter (3-2)	
24..25	Decision Filter (1-0)			

Filter index range: 0x0..0x4.

Bit #	Name	Description
0	Unicast (AND)	If set, packets must match a unicast filter.
1	Broadcast (AND)	If set, packets must match the broadcast filter.
2	VLAN (AND)	If set, packets must match a VLAN filter.
3	IP Address (AND)	If set, packets must match an IP filter.
4	Unicast (OR)	If set, packets must match a unicast filter or a different OR filter.
5	Broadcast	If set, packets must match the broadcast filter or a different OR filter.
6	Multicast (AND)	If set, packets must match the multicast filter.
7	ARP Request (OR)	If set, packets must match the ARP request filter or a different OR filter.
8	ARP Response (OR)	If set, packets must also match the ARP response filter or a different OR filter.
9	Neighbor Discovery (OR)	If set, packets must also match the neighbor discovery filter or a different OR filter.
10	Port 0x298 (OR)	If set, packets must also match a fixed TCP/UDP port 0x298 filter or a different OR filter.
11	Port 0x26F (OR)	If set, packets must also match a fixed TCP/UDP port 0x26F filter or a different OR filter.
12	Flex port 0 (OR)	If set, packets must also match the TCP/UDP port filter 0 or a different OR filter.
13	Flex port 1 (OR)	If set, packets must also match the TCP/UDP port filter 1 or a different OR filter.
14	Flex port 2 (OR)	If set, packets must also match the TCP/UDP port filter 2 or a different OR filter.
15	Flex port 3 (OR)	If set, packets must also match the TCP/UDP port filter 3 or a different OR filter.
16	Flex port 4 (OR)	If set, packets must also match the TCP/UDP port filter 4 or a different OR filter.
17	Flex port 5 (OR)	If set, packets must also match the TCP/UDP port filter 5 or a different OR filter.
18	Flex port 6 (OR)	If set, packets must also match the TCP/UDP port filter 6 or a different OR filter.
19	Flex port 7 (OR)	If set, packets must also match the TCP/UDP port filter 7 or a different OR filter.

20	Flex port 8 (OR)	If set, packets must also match the TCP/UDP port filter 8 or a different OR filter.
21	Flex port 9 (OR)	If set, packets must also match the TCP/UDP port filter 9 or a different OR filter.
22	Flex port 10 (OR)	If set, packets must also match the TCP/UDP port filter 10 or a different OR filter.
23	Flex port 11 (OR)	If set, packets must also match the TCP/UDP port filter 11 or a different OR filter.
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	
28	Flex TCO 0 (OR)	If set, packets must also match the Flex 128 TCO filter 0 or a different OR filter.
29	Flex TCO 1 (OR)	If set, packets must also match the Flex 128 TCO filter 1 or a different OR filter.
30	Flex TCO 2 (OR)	If set, packets must also match the Flex 128 TCO filter 2 or a different OR filter.
31	Flex TCO 3 (OR)	If set, packets must also match the Flex 128 TCO filter 3 or a different OR filter.

The filtering is divided into two decisions:

- Bits 0, 1, 2, 3, and 6 work in an AND manner; they all must be true in order for a packet to pass (if any were set).
- Bits 5 and 7-31 work in an OR manner; at least one of them must be true for a packet to pass (if any were set).

See Figure 5 for more details on the decision filters.

Note: These filter settings operate according to the VLAN mode, as configured according to the DMTF NC-SI specification. After disabling packet reduction filters, the MC must re-set the VLAN mode using the Set VLAN command.

4.2.2.5.10 Set Intel Filters - Packet Addition Decision Filter Response (Intel Command 0x02, Filter Parameter 0x61)

	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 (Intel 0x157)			
24..25	0x02	0x61		

4.2.2.5.11 Set Intel Filters - Flex TCP/UDP Port Filter Command (Intel Command 0x02, Filter Parameter 0x63)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x63	TCP/UDP Port	

Filter index range: 0x0..0xA.

4.2.2.5.12 Set Intel Filters - Flex TCP/UDP Port Filter Response (Intel Command 0x02, Filter Parameter 0x63)

	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 (Intel 0x157)			
24..25	0x02	0x63		

4.2.2.5.13 Set Intel Filters - IPv4 Filter Command (Intel Command 0x02, Filter Parameter 0x64)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x64	IPv4 Address (3-2)	
24..25	IPv4 Address (3-2)			

Note: The filters index range can vary according to the IPv4/IPv6 mode setting in the Filters Control command.

IPv4 Mode: Filter index range: 0x0..0x3.

IPv6 Mode: No IPv4 Filters.

4.2.2.5.14 Set Intel Filters - IPv4 Filter Response (Intel Command 0x02, Filter Parameter 0x64)

	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 (Intel 0x157)			
24..25	0x02	0x64		

4.2.2.5.15 Set Intel Filters - IPv6 Filter Command (Intel Command 0x02, Filter Parameter 0x65)

Bits / Bytes	31..24	23..16	15..08	07..00
00..03	**MC ID**	***0x01***	**Reserved**	**IID**
04..07	***0x50***	**ID**	Resvd.	***0x09***
08..11	Reserved			
12..15	Reserved			
16..19	**0x8086**		**0x02**	**0x65**
20...23	IPv6 Filter Index	IPv6 Address (MSB, Byte 15)	..	..
24...27	..	..	..	..
28...31	..	..	..	..
32...35	..	..	..	..
36	IPv6 Address (LSB, Byte 0)			

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x02	0x65	IPv6 Address (MSB, byte 15)	..
24..27	..	..	..	..
28..31	..	..	..	..
32..35	..	..	..	..
36..37	..	IPv6 Address (LSB, byte 0)		

Note: The filters index range can vary according to the IPv4/IPv6 mode setting in the Filters Control command.

IPv4 Mode: Filter index range: 0x0..0x2.

IPv6 Mode: Filter index range: 0x0..0x3.

4.2.2.5.16 Set Intel Filters - IPv6 Filter Response (Intel Command 0x02, Filter Parameter 0x65)

	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 (Intel 0x157)			
24..25	0x02	0x65		

4.2.2.6 Get Intel Filters Formats

4.2.2.6.1 Get Intel Filters Command (Intel Command 0x03)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x03	Parameter Number		

4.2.2.6.2 Get Intel Filters Response (Intel Command 0x03)

	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 (Intel 0x157)			
24..25	0x03	Parameter Number	Optional Return Data	

4.2.2.6.3 Get Intel Filters - Manageability to Host Command (Intel Command 0x03, Filter Parameter 0x0A)

This command retrieves the Mng2Host register. The Mng2Host register controls whether pass-through packets destined to the BMC are also be forwarded to the host OS.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x03	0x0A		

4.2.2.6.4 Get Intel Filters - Manageability to Host Response (Intel Command 0x03, Filter Parameter 0x0A)

The Mng2Host register has the following structure:

Bits	Description	Default
0	Decision Filter 0	Determines if packets that have passed decision filter 0 are also forwarded to the host OS.
1	Decision Filter 1	Determines if packets that have passed decision filter 1 are also forwarded to the host OS.
2	Decision Filter 2	Determines if packets that have passed decision filter 2 are also forwarded to the host OS.
3	Decision Filter 3	Determines if packets that have passed decision filter 3 are also forwarded to the host OS.
4	Decision Filter 4	Determines if packets that have passed decision filter 4 are also forwarded to the host OS.
5	Unicast and Mixed	Determines if broadcast packets are also forwarded to the host OS.
6	Global Multicast	Determines if unicast packets are also forwarded to the host OS.
7	Broadcast	Determines if multicast packets are also forwarded to the host OS.

	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 (Intel 0x157)			
24..27	0x03	0x0A	Manageability to Host (3-2)	
28..29	Manageability to Host (1-0)			

4.2.2.6.5 Get Intel Filters - Flex Filter 0 Enable Mask and Length Command (Intel Command 0x03, Filter Parameter 0x10/0x20/0x30/0x40)

The following command retrieves the Intel flex filters mask and length. Use filter parameters 0x10/ 0x20/0x30/0x40 for flexible filters 0/1/2/3 accordingly.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x03	0x10/ 0x20/ 0x30/ 0x40		

4.2.2.6.6 Get Intel Filters - Flex Filter 0 Enable Mask and Length Response (Intel Command 0x03, Filter Parameter 0x10/0x20/0x30/0x40)

	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 (Intel 0x157)			
24..27	0x03	0x10/ 0x20/ 0x30/ 0x40	Mask Byte 1	Mask Byte 2
28..31	..	..	..	..
32..35	..	..	..	..
36..39	..	..	..	..
40..43	..	Mask Byte 16	Reserved	Reserved
44	Flexible Filter Length			

4.2.2.6.7 Get Intel Filters - Flex Filter 0 Data Command (Intel Command 0x03, Filter Parameter 0x11/0x21/0x31/0x41)

The following command retrieves the Intel flex filters data. Use filter parameters 0x11/0x21/0x31/0x41 for flexible filters 0/1/2/3 accordingly.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x03	0x11/ 0x21/ 0x31/ 0x41		

4.2.2.6.8 Get Intel Filters - Flex Filter 0 Data Response (Intel Command 0x03, Filter Parameter 0x11)

	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 (Intel 0x157)			
24..	0x03	0x11/ 0x21/ 0x31/ 0x41	Filter Group Number	Filter Data 1
	..	Filter Data N		

4.2.2.6.9 Get Intel Filters - Packet Addition Decision Filter Command (Intel Command 0x03, Filter Parameter 0x61)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x03	0x61		

Filter index range: 0x0..0x4.

4.2.2.6.10 Get Intel Filters - Packet Addition Decision Filter Response (Intel Command 0x03, Filter Parameter 0x0A)

	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 (Intel 0x157)			
24..27	0x03	0x61	Decision Filter (3-2)	
28..29	Decision Filter (1-0)			

4.2.2.6.11 Get Intel Filters - Flex TCP/UDP Port Filter Command (Intel Command 0x03, Filter Parameter 0x63)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x63	TCP/UDP Filter Index	

Filter index range: 0x0..0xA.

4.2.2.6.12 Get Intel Filters - Flex TCP/UDP Port Filter Response (Intel Command 0x03, Filter Parameter 0x63)

	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 (Intel 0x157)			
24..27	0x03	0x63	TCP/UDP Filter Index	TCP/UDP Port (1)
28	TCP/UDP Port (0)			

Filter index range: 0x0..0xA.

4.2.2.6.13 Get Intel Filters - IPv4 Filter Command (Intel Command 0x03, Filter Parameter 0x64)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x64	IPv4 Filter Index	

Note: The filters index range can vary according to the IPv4/IPv6 mode setting in the Filters Control command.

IPv4 Mode: Filter index range: 0x0..0x3.

IPv6 Mode: No IPv4 filters.

4.2.2.6.14 Get Intel Filters - IPv4 Filter Response (Intel Command 0x03, Filter Parameter 0x64)

	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 (Intel 0x157)			
24..27	0x03	0x64	IPv4 Filter Index	IPv4 Address (3)
28..29	IPv4 Address (2-0)			

4.2.2.6.15 Get Intel Filters - IPv6 Filter Command (Intel Command 0x03, Filter Parameter 0x65)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x03	0x65	IPv6 Filter Index	

Note: The filters index range can vary according to the IPv4/IPv6 mode setting in the Filters Control command

IPv4 Mode: Filter index range: 0x0..0x2.

IPv6 Mode: Filter index range: 0x0..0x3.

4.2.2.6.16 Get Intel Filters - IPv6 Filter Response (Intel Command 0x03, Filter parameter 0x65)

	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 (Intel 0x157)			
24..27	0x03	0x65	IPv6 Filter Index	IPv6 Address (MSB, Byte 16)
28..31	..	..	..	..
32..35	..	..	..	..
36..39	..	..	..	..
40..42	..	..	IPv6 Address (LSB, Byte 0)	

4.2.2.7 Set Intel Packet Reduction Filters Formats

4.2.2.7.1 Set Intel Packet Reduction Filters Command (Intel Command 0x04)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	Packet Reduction Index		

4.2.2.7.2 Set Intel Packet Reduction Filters Response (Intel Command 0x04)

	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 (Intel 0x157)			
24..	0x04	Packet Reduction Index	Optional Return Data	

4.2.2.7.3 Set Unicast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x00)

This command causes the NC to filter packets that have passed due to the unicast filter (MAC address filters, as specified in the DMTF NC-SI). Note that unicast filtering might be affected by other filters, as specified in the DMTF NC-SI.

The filtering of these packets are done such that the MC might add a logical condition that a packet must match, or it must be discarded.

Note: Packets that might have been blocked can still pass due to other decision filters.

In order to disable unicast packet reduction, the MC should set all reduction filters to 0b. Following such a setting the NC must forward, to the MC, all packets that have passed the unicast filters (MAC address filtering) as specified in the DMTF NC-SI.

The *Unicast Packet Reduction* field has the following structure:

Bit #	Name	Description
0	Reserved	
1	Reserved	
2	Reserved	
3	IP Address	If set, all unicast packets must also match an IP filter.
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	
8	ARP Response	If set, all unicast packets must also match the ARP response filter (any of the active filters).
9	Reserved	
10	Port 0x298	If set, all unicast packets must also match a fixed TCP/UDP port 0x298 filter.
11	Port 0x26F	If set, all unicast packets must also match a fixed TCP/UDP port 0x26F filter.
12	Flex port 0	If set, all unicast packets must also match the TCP/UDP port filter 0.
13	Flex port 1	If set, all unicast packets must also match the TCP/UDP port filter 1.
14	Flex port 2	If set, all unicast packets must also match the TCP/UDP port filter 2.
15	Flex port 3	If set, all unicast packets must also match the TCP/UDP port filter 3.
16	Flex port 4	If set, all unicast packets must also match the TCP/UDP port filter 4.
17	Flex port 5	If set, all unicast packets must also match the TCP/UDP port filter 5.
18	Flex port 6	If set, all unicast packets must also match the TCP/UDP port filter 6.
19	Flex port 7	If set, all unicast packets must also match the TCP/UDP port filter 7.
20	Flex port 8	If set, all unicast packets must also match the TCP/UDP port filter 8.
21	Flex port 9	If set, all unicast packets must also match the TCP/UDP port filter 9.
22	Flex port 10	If set, all unicast packets must also match the TCP/UDP port filter 10.
23	Reserved	
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	
28	Flex TCO 0	If set, all unicast packets must also match the flex 128 TCO filter 0.

29	Flex TCO 1	If set, all unicast packets must also match the flex 128 TCO filter 1.
30	Flex TCO 2	If set, all unicast packets must also match the flex 128 TCO filter 2.
31	Flex TCO 3	If set, all unicast packets must also match the flex 128 TCO filter 3.

The filtering is divided into two decisions:

- Bit 3 works in an AND manner; it must be true in order for a packet to pass (if was set).
- Bits 8-31 work in an OR manner; at least one of them must be true for a packet to pass (if any were set).

See Figure 5 for more details on the decision filters.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x00	Unicast Packet Reduction (3-2)	
24..25	Unicast Packet Reduction (1-0)			

4.2.2.7.4 Set Unicast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x00)

	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 (Intel 0x157)			
24..25	0x04	0x00		

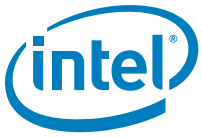
4.2.2.7.5 Set Multicast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x01)

This command causes the NC to filter packets that have passed due to the multicast filter (MAC address filters, as specified in the DMTF NC-SI).

The filtering of these packets are done such that the MC might add a logical condition that a packet must match, or it must be discarded.

Note: Packets that might have been blocked can still pass due to other decision filters.

In order to disable mulitcast packet reduction, the MC should set all reduction filters to 0b. Following such a setting, the NC must forward, to the MC, all packets that have passed the multicast filters (global multicast filtering) as specified in the DMTF NC-SI.



The *Multicast Packet Reduction* field has the following structure:

Bit #	Name	Description
0	Reserved	Reserved.
1	Reserved	
2	Reserved	
3	IP Address	If set, all multicast packets must also match an IP filter.
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	
8	ARP Response	If set, all multicast packets must also match the ARP response filter (any of the active filters).
9	Reserved	
10	Port 0x298	If set, all multicast packets must also match a fixed TCP/UDP port 0x298 filter.
11	Port 0x26F	If set, all multicast packets must also match a fixed TCP/UDP port 0x26F filter.
12	Flex port 0	If set, all multicast packets must also match the TCP/UDP port filter 0.
13	Flex port 1	If set, all multicast packets must also match the TCP/UDP port filter 1.
14	Flex port 2	If set, all multicast packets must also match the TCP/UDP port filter 2.
15	Flex port 3	If set, all multicast packets must also match the TCP/UDP port filter 3.
16	Flex port 4	If set, all multicast packets must also match the TCP/UDP port filter 4.
17	Flex port 5	If set, all multicast packets must also match the TCP/UDP port filter 5.
18	Flex port 6	If set, all multicast packets must also match the TCP/UDP port filter 6.
19	Flex port 7	If set, all multicast packets must also match the TCP/UDP port filter 7.
20	Flex port 8	If set, all multicast packets must also match the TCP/UDP port filter 8.
21	Flex port 9	If set, all multicast packets must also match the TCP/UDP port filter 9.
22	Flex port 10	If set, all multicast packets must also match the TCP/UDP port filter 10.
23	Reserved	
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	
28	Flex TCO 0	If set, all multicast packets must also match the flex 128 TCO filter 0.
29	Flex TCO 0	If set, all multicast packets must also match the flex 128 TCO filter 1.
30	Flex TCO 0	If set, all multicast packets must also match the flex 128 TCO filter 2.
31	Flex TCO 0	If set, all multicast packets must also match the flex 128 TCO filter 3.

The filtering is divided into two decisions:

Bit 3 works in an AND manner; it must be true in order for a packet to pass (if was set).

Bits 4, 5, and 7-31 work in an OR manner; at least one of them must be true for a packet to pass (if any were set).

See Figure 5 for more details on the decision filters.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x01	Multicast Packet Reduction (3-2)	
24..25	Multicast Packet Reduction (1-0)			

4.2.2.7.6 Set Multicast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x01)

	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 (Intel 0x157)			
24..25	0x04	0x01		

4.2.2.7.7 Set Broadcast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x02)

This command causes the NC to filter packets that have passed due to the broadcast filter (MAC address filters, as specified in the DMTF NC-SI).

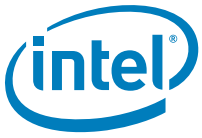
The filtering of these packets are done such that the MC might add a logical condition that a packet must match, or it must be discarded.

Note: Packets that might have been blocked can still pass due to other decision filters.

In order to disable broadcast packet reduction, the MC should set all reduction filters to 0b. Following such a setting, the NC must forward, to the MC, all packets that have passed the broadcast filters as specified in the DMTF NC-SI.

The *Broadcast Packet Reduction* field has the following structure:

Bit #	Name	Description
0	Reserved	Reserved.
1	Reserved	
2	Reserved	
3	IP Address	If set, all broadcast packets must also match an IP filter.
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	



8	ARP Response	If set, all broadcast packets must also match the ARP response filter (any of the active filters).
9	Reserved	
10	Port 0x298	If set, all broadcast packets must also match a fixed TCP/UDP port 0x298 filter.
11	Port 0x26F	If set, all broadcast packets must also match a fixed TCP/UDP port 0x26F filter.
12	Flex port 0	If set, all broadcast packets must also match the TCP/UDP port filter 0.
13	Flex port 1	If set, all broadcast packets must also match the TCP/UDP port filter 1.
14	Flex port 2	If set, all broadcast packets must also match the TCP/UDP port filter 2.
15	Flex port 3	If set, all broadcast packets must also match the TCP/UDP port filter 3.
16	Flex port 4	If set, all broadcast packets must also match the TCP/UDP port filter 4.
17	Flex port 5	If set, all broadcast packets must also match the TCP/UDP port filter 5.
18	Flex port 6	If set, all broadcast packets must also match the TCP/UDP port filter 6.
19	Flex port 7	If set, all broadcast packets must also match the TCP/UDP port filter 7.
20	Flex port 8	If set, all broadcast packets must also match the TCP/UDP port filter 8.
21	Flex port 9	If set, all broadcast packets must also match the TCP/UDP port filter 9.
22	Flex port 10	If set, all broadcast packets must also match the TCP/UDP port filter 10.
23	Reserved	
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	
28	Flex TCO 0	If set, all broadcast packets must also match the flex 128 TCO filter 0.
29	Flex TCO 0	If set, all broadcast packets must also match the flex 128 TCO filter 1.
30	Flex TCO 0	If set, all broadcast packets must also match the flex 128 TCO filter 2.
31	Flex TCO 0	If set, all broadcast packets must also match the flex 128 TCO filter 3.

The filtering is divided into two decisions:

Bit 3 works in an AND manner; it must be true in order for a packet to pass (if was set).

Bits 4, 5, and 7-31 work in an OR manner; at least one of them must be true for a packet to pass (if any were set).

See Figure 5 for more details on the decision filters.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x02	Broadcast Packet Reduction (3-2)	
24..25	Broadcast Packet Reduction (1-0)			

4.2.2.7.8 Set Broadcast Packet Reduction Response (Intel Command 0x08)

	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 (Intel 0x157)			
24..25	0x04	0x02		

4.2.2.8 Get Intel Packet Reduction Filters Formats

4.2.2.8.1 Get Intel Packet Reduction Filters Command (Intel Command 0x05)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	Reduction Filter Index		

4.2.2.8.2 Set Intel Packet Reduction Filters Response (Intel Command 0x05)

	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 (Intel 0x157)			
24..	0x05	Reduction Filter Index	Optional Return Data	

4.2.2.8.3 Get Unicast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x00)

This command causes the NC to disable any packet reductions for unicast address filtering.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x00		

4.2.2.8.4 Get Unicast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x00)

	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 (Intel 0x157)			
24..27	0x05	0x00	Unicast Packet Reduction (3-2)	
28..29	Unicast Packet Reduction (1-0)			

4.2.2.8.5 Get Multicast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x01)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x01		

4.2.2.8.6 Get Multicast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x01)

	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 (Intel 0x157)			
24..27	0x05	0x00	Multicast Packet Reduction (3-2)	
28..29	Multicast Packet Reduction (1-0)			

4.2.2.8.7 Get Broadcast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x02)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x05	0x02		

4.2.2.8.8 Get Broadcast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x02)

	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 (Intel 0x157)			
24..27	0x05	0x00	Broadcast Packet Reduction (3-2)	
28..29	Broadcast Packet Reduction (1-0)			

4.2.2.9 System MAC Address

4.2.2.9.1 Get System MAC Address Command (Intel Command 0x06)

In order to support a system configuration that requires the NC to hold the MAC address for the MC (such as shared MAC address mode), the following command is provided to enable the MC to query the NC for a valid MAC address.

The NC must return the system MAC addresses. The MC should use the returned MAC addressing as a shared MAC address by setting it using the Set MAC Address command as defined in NC-SI 1.0.

It is also recommended that the MC use packet reduction and Manageability-to-Host command to set the proper filtering method.

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20	0x06			

4.2.2.9.2 Get System MAC Address Response (Intel Command 0x06)

	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 (Intel 0x157)			
24..27	0x06	MAC Address		
28..30	MAC Address			

4.2.2.10 Set Intel Management Control Formats

4.2.2.10.1 Set Intel Management Control Command (Intel Command 0x20)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..22	0x20	0x00	Intel Management Control 1	

Where:

Intel Management Control 1 is as follows:

Bit #	Default value	Description
0	0b	Enable Critical Session Mode (Keep Phy Link Up and Veto Bit) 0b - Disabled 1b - Enabled When critical session mode is enabled, the following behaviors are disabled: • The PHY is not reset on PE_RST# and PCIe* resets (in-band and link drop). Other reset events are not affected - Internal_Power_On_Reset, device disable, Force TCO, and PHY reset by software. • The PHY does not change its power state. As a result link speed does not change. • The device does not initiate configuration of the PHY to avoid losing link.
1...7	0x0	Reserved

4.2.2.10.2 Set Intel Management Control Response (Intel Command 0x20)

	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 (Intel 0x157)			
24..25	0x20	0x00		

4.2.2.11 Get Intel Management Control Formats

4.2.2.11.1 Get Intel Management Control Command (Intel Command 0x21)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..21	0x20	0x00		

Where:

Intel Management Control 1 is as described in Section 4.2.2.10.2.

4.2.2.11.2 Get Intel Management Control Response (Intel Command 0x21)

	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 (Intel 0x157)			
24..26	0x21	0x00	Intel Management Control 1	

4.2.2.12 TCO Reset

This command causes the NC to perform TCO reset, if force TCO reset is enabled in the EEPROM.

If the BMC has detected that the OS is hung and has blocked the Rx/Tx path, the force TCO reset clears the data-path (Rx/Tx) of the NC to enable the BMC to transmit/receive packets through the NC.

When this command is issued to a channel in a package, it applies only to the specific channel.

After successfully performing the command, the NC considers the Force TCO command as an indication that the OS is hung and clears the DRV_LOAD flag (disable the LAN device driver).

4.2.2.12.1 Perform Intel TCO Reset Command (Intel Command 0x22)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20	0x22			

4.2.2.12.2 Perform Intel TCO Reset Response (Intel Command 0x22)

	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 (Intel 0x157)			
24..26	0x22			

4.2.2.13 Checksum Offloading

This command enables the checksum offloading filters in the NC.

When enabled, these filters block any packets that did not pass IP, UDP and TCP checksums from being forwarded to the MC.

4.2.2.13.1 Enable Checksum Offloading Command (Intel Command 0x23)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20	0x23			

4.2.2.13.2 Enable Checksum Offloading Response (Intel Command 0x23)

	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 (Intel 0x157)			
24..26	0x23			

4.2.2.13.3 Disable Checksum Offloading Command (Intel Command 0x24)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20	0x24			

4.2.2.13.4 Disable Checksum Offloading Response (Intel Command 0x24)

	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 (Intel 0x157)			
24..26	0x24			

4.3 Basic NC-SI Workflows

4.3.1 Package States

A NC package can be in one of the following two states:

1. Selected - In this state, the package is allowed to use the NC-SI lines, meaning the NC package might send data to the BMC.
2. De-selected - In this state, the package is not allowed to use the NC-SI lines, meaning, the NC package cannot send data to the BMC.

Also note that the MC must select no more than one NC package at any given time.

Package selection can be accomplished in one of two methods:

1. Select Package command - this command explicitly selects the NC package.
2. Any other command targeted to a channel in the package also implicitly selects that NC package.

Package de-select can be accomplished only by issuing the De-Select Package command.

Note: The MC should always issue the Select Package command as the first command to the package before issuing channel-specific commands.

For further details on package selection, refer to the NC-SI specification.

4.3.2 Channel States

A NC channel can be in one of the following states:

1. Initial State - In this state, the channel only accepts the Clear Initial State command (the package also accepts the Select Package and De-Select Package commands).
2. Active state - This is the normal operational mode. All commands are accepted.

For normal operation mode, the MC should always send the Clear Initial State command as the first command to the channel.

4.3.3 Discovery

After interface power-up, the MC should perform a discovery process to discover the NCs that are connected to it.

This process should include an algorithm similar to the following:

1. For package_id=0x0 to MAX_PACKAGE_ID
   a. Issue Select Package command to package ID package_id
   b. If received a response then

      For internal_channel_id = 0x0 to MAX_INTERNAL_CHANNEL_ID

      Issue a Clear Initial State command for package_id | internal_channel_id (the combination of package_id and internal_channel_id to create the channel ID).

      If a response was received then

      Consider internal_channel_id as a valid channel for the package_id package

      The MC can now optionally discover channel capabilities and version ID for the channel

      Else (If not a response was not received, then issue a Clear Initial State command three times.

      Issue a De-Select Package command to the package (and continue to the next package).
   c. Else (If a response was not received, issue a Select Packet command three times.

4.3.4 Configurations

This section details different configurations that should be performed by the MC.

It is considered a good practice that the MC does not consider any configuration valid unless the MC has explicitly configured it after every reset (entry into the initial state).

As a result, it is recommended that the MC re-configure everything at power-up and channel/package resets.

4.3.4.1 NC Capabilities Advertisement

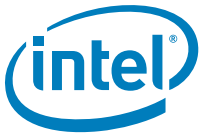
NC-SI defines the Get Capabilities command. It is recommended that the MC use this command and verify that the capabilities match its requirements before performing any configurations.

For example, the MC should verify that the NC supports a specific AEN before enabling it.

4.3.4.2 Receive Filtering

In order to receive traffic, the MC must configure the NC with receive filtering rules. These rules are checked on every packet received on the LAN interface (such as from the network). Only if the rules matched, will the packet be forwarded to the MC.

4.3.4.2.1 MAC Address Filtering



NC-SI defines three types of MAC address filters: unicast, multicast and broadcast. To be received (not dropped) a packet must match at least one of these filters.

Note: The MC should set one MAC address using the Set MAC Address command and enable broadcast and global multicast filtering.

Unicast/Exact Match (Set MAC Address Command)

This filter filters on specific 48-bit MAC addresses. The MC must configure this filter with a dedicated MAC address.

Note: The NC might expose three types of unicast/exact match filters (such as MAC filters that match on the entire 48 bits of the MAC address): unicast, multicast and mixed. The 82575EB exposes two mixed filters, which might be used both for unicast and multicast filtering. The MC should use one mixed filter for its MAC address.

Refer to NC-SI specification - Set MAC Address for further details.

Broadcast (Enable/Disable Broadcast Filter Command)

NC-SI defines a broadcast filtering mechanism which has the following states:

1. Enabled - All broadcast traffic is blocked (not forwarded) to the MC, except for specific filters (such as ARP request, DHCP, and NetBIOS).
2. Disabled - All broadcast traffic is forwarded to the MC, with no exceptions.

Note: Refer to NC-SI specification Enable/Disable Broadcast Filter command.

Global Multicast (Enable/Disable Global Multicast Filter)

NC-SI defines a multicast filtering mechanism which has the following states:

1. Enabled - All multicast traffic is blocked (not forwarded) to the MC.
2. Disabled - All multicast traffic is forwarded to the MC, with no exceptions.

The recommended operational mode is Enabled, with specific filters set.

Note: Not all multicast filtering modes are necessarily supported.

Refer to NC-SI specification Enable/Disable Global Multicast Filter command for further details.

4.3.4.2.2 VLAN

NC-SI defines the following VLAN work modes:

Mode	Command and Name	Descriptions
Disabled	Disable VLAN command	In this mode, no VLAN frames are received.
Enabled #1	Enable VLAN command with VLAN only	In this mode, only packets that matched a VLAN filter are forwarded to the MC.
Enabled #2	Enable VLAN command with VLAN only + non-VLAN	In this mode, packets from mode 1 + non-VLAN packets are forwarded.
Enabled #3	Enable VLAN command with Any-VLAN + non-VLAN	In this mode, packets are forwarded regardless of their VLAN state.

Refer to NC-SI specification - Enable VLAN command for further details.

The 82575EB only supports modes #1 and #3.

Recommendation:

1. Modes:
   a. If VLAN is not required - use the disabled mode.
   b. If VLAN is required - use the enabled #1 mode.
2. If enabling VLAN, The MC should also set the active VLAN ID filters using the NC-SI Set VLAN Filter command prior to setting the VLAN mode.

4.3.5 Pass-Through Traffic States

The MC has independent, separate controls for enablement states of the receive (from LAN) and of the transmit (to LAN) pass-through paths.

4.3.5.1 Channel Enable

This mode controls the state of the receive path:

1. Disabled: The channel does not pass any traffic from the network to the MC.
2. Enabled: The channel passes any traffic from the network (that matched the configured filters) to the MC.

Note: This state also affects AENs: AENs is only sent in the enabled state.

The default state is disabled.

It is recommended that the MC complete all filtering configuration before enabling the channel.

4.3.5.2 Network Transmit Enable

This mode controls the state of the transmit path:

1. Disabled - the channel does not pass any traffic from the MC to the network.
2. Enabled - the channel passes any traffic from the MC (that matched the source MAC address filters) to the network.

Note: The default state is disabled.

The NC filters pass-through packets according to their source MAC address. The NC tries to match that source MAC address to one of the MAC addresses configured by the Set MAC Address command. As a result, the MC should enable network transmit only after configuring the MAC address.

It is recommended that the MC complete all filtering configuration (especially MAC addresses) before enabling the network transmit.

This feature can be used for fail-over scenarios. See Section 4.5.3.

4.3.6 Asynchronous Event Notifications

The asynchronous event notifications are unsolicited messages sent from the NC to the MC to report status changes (such as link change, OS state change, etc.).

Recommendations:

- The MC firmware designer should use AENs. To do so, the designer must take into account the possibility that a NC-SI response frame (such as a frame with the NC-SI EtherType), arrives out-of-context (not immediately after a command, but rather after an out-of-context AEN).
- To enable AENs, the MC should first query which AENs are supported, using the Get Capabilities command, then enable desired AEN(s) using the Enable AEN command, and only then enable the channel using the Enable Channel command.

4.3.7 Querying Active Parameters

The MC can use the Get Parameters command to query the current status of the operational parameters.

4.4 Resets

In NC-SI there are two types of resets defined:

1. Synchronous entry into the initial state.
2. Asynchronous entry into the initial state.

Recommendations:

- It is very important that the MC firmware designer keep in mind that following any type of reset, all configurations are considered as lost and thus the MC must re-configure everything.
- As an asynchronous entry into the initial state might not be reported and/or explicitly noticed, the MC should periodically poll the NC with NC-SI commands (such as Get Version ID, Get Parameters, etc.) to verify that the channel is not in the initial state. Should the NC channel respond to the command with a Clear Initial State Command Expected reason code - The MC should consider the channel (and most probably the entire NC package) as if it underwent a (possibly unexpected) reset event. Thus, the MC should re-configure the NC. See the NC-SI specification section on Detecting Pass-through Traffic Interruption.
- The Intel recommended polling interval is 2-3 seconds.

For exact details on the resets, refer to NC-SI specification.

4.5 Advanced Workflows

4.5.1 Multi-NC Arbitration

As described in Section 4.1.2, in a multi-NC environment, there is a need to arbitrate the NC-SI lines.

Figure 12 shows the system topology of such an environment.

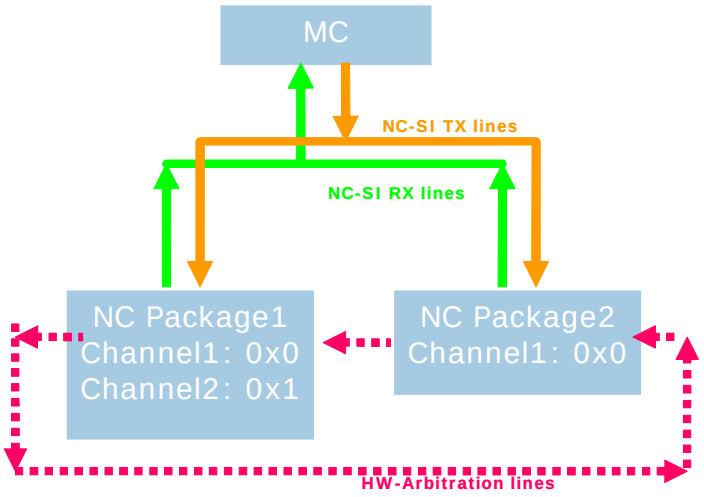


Figure 12. Multi-NC Environment

As shown in Figure 12, the NC-SI Rx lines are shared between the NCs.

Thus, to enable sharing of the NC-SI Rx lines, NC-SI has defined an arbitration scheme.

The arbitration scheme mandates that only one NC package can use the NC-SI Rx lines at any given time. The NC package that is allowed to use these lines is defined as selected. All the other NC packages are de-selected.

NC-SI has defined two mechanisms for the arbitration scheme:

1. Package Selection by the MC. In this mechanism, the MC is responsible for arbitrating between the packages by issuing NC-SI commands (Select/De-Select Package). The MC is responsible for having only one package selected at any given time.
2. HW Arbitration. In this mechanism, two additional pins on each NC package are used to synchronize the NC package. Each NC package has an ARB_IN and ARB_OUT line and these lines are used to transfer Tokens. A NC package that has a token is considered selected.

Note: Hardware arbitration is enabled by default after interface power-up.

The 82575EB does not support hardware arbitration.

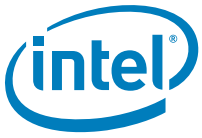
For further details, refer to section 4 in the NC-SI specification.

4.5.1.1 Package Selection Sequence Example

Following is an example work flow for a MC and occurs after the discovery, initialization, and configuration.

Assuming the MC needs to share the NC-SI bus between packages the MC should:

1. Define a time-slot for each device.
2. Discover, initialize, and configure all the NC packages and channels.



3. Issue a De-Select Package command to all the channels.
4. Set active_package to 0x0 (or the lowest existing package ID).
5. At the beginning of each time slot the MC should:
   a. Issue a De-Select Package to the active_package. The MC must then wait for a response and then an additional timeout for the package to become de-selected (200 µs). See the NC-SI specification table 10 - parameter NC Deselect to Hi-Z Interval.
   b. Find the next available package (typically active_package = active_package + 1).
   c. Issue a Select Package command to active_package.

4.5.2 External Link Control

The MC can use the NC-SI Set Link command to control the external interface link settings. This command enables the MC to set the auto-negotiation, link speed, duplex, and other parameters.

This command is only available when the host OS is not present. Indicating the host OS status can be obtained via the Get Link Status command and/or Host OS Status Change AEN command.

Recommendation:

- Unless explicitly needed, it is not recommended to use this feature. The NC-SI Set Link command does not expose all the possible link settings and/or features. This might cause issues under different scenarios. Even if decided to use this feature, it is recommended to use it only if the link is down (trust the 82575EB until proven otherwise).
- It is recommended that the MC first query the link status using the Get Link Status command. The MC should then use this data as a basis and change only the needed parameters when issuing the Set Link command.

For further details, refer to the NC-SI specification.

4.5.2.1 Set Link While LAN PCIe Functionality is Disabled

In cases where a LAN device is used solely for manageability and its LAN PCIe function is disabled, using the NC-SI Set Link command while advertising multiple speeds and enabling auto-negotiation results in the lowest possible speed chosen.

To enable link of higher a speed, the MC should not advertise speeds that are below the desired link speed, as the lowest advertised link speed is chosen.

When the LAN device is only used for manageability and the link speed advertisement is configured by the MC, changes in the power state of the LAN device is not effected and the link speed is not re-negotiated by the LAN device.

4.5.3 Multiple Channels (Fail-Over)

In order to support a fail-over scenario, it is required from the MC to operate two or more channels. These channels might or might not be in the same package.

The key element of a fault-tolerance fail-over scenario is having two (or more) channels identifying to the switch with the same MAC address, but only one of them being active at any given time (such as switching the MAC address between channels).

To accomplish this, NC-SI provides the following:

1. Enable Network Tx command. This command enables shutting off the network transmit path of a specific channel. This enables the MC to configure all the participating channels with the same MAC address but only enable one of them.
2. Link Status Change AEN or Get Link Status command.

4.5.3.1 Fail-Over Algorithm Example

Following is a sample workflow for a fail-over scenario for the 82575EB dual-port GbE controller (one package and two channels).

1. MC initializes and configures both channels after power-up. However, the MC uses the same MAC address for both of the channels.
2. The MC queries the link status of all the participating channels. The MC should continuously monitor the link status of these channels. This can be accomplished by listening to AENs (if used) and/or periodically polling using the Get Link Status command.
3. The MC then only enables channel 0 for network transmission.
4. The MC then issues a gratuitous ARP (or any other packet with its source MAC address) to the network. This packet informs the switch that this specific MAC address is registered to channel 0's specific LAN port.
5. The MC begins normal workflow.
6. Should the MC receive an indication (AEN or polling) that the link status for the active channel (channel 0) has changed, the MC should:
   a. Disable channel0 for Network Transmission.
   b. Check if a different channel is available (link is up).

      If found:

      Enable network Tx for that specific channel.

      Issue a gratuitous ARP (or any other packet with its source MAC address) to the network. This packet informs the switch that this specific MAC address is registered to channel 0's specific LAN port.

      Resume normal workflow.

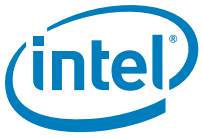
      If not found, report the error and continue polling until a valid channel is found.

Note: The above algorithm can be generalized such that the start-up and normal workflow are the same.

In addition, the MC might need to use a specific channel (such as channel 0). In this case, the MC should switch the network transmit to that specific channel as soon as that channel becomes valid (link is up).

Recommendations:

- It is recommended to wait a link-down-tolerance timeout before a channel is considered invalid. For example, a link re-negotiation might take a few seconds (normally 2 to 3 or might be up to 9). Thus, the link must be re-established after a short time.
- Typically, this timeout is recommended to be three seconds.
- Even when enabling and using AENs, it is still recommended to periodically poll the link status, as dropped AENs might not be detected.



4.5.4 Statistics

The MC might use the statistics commands as defined in NC-SI. These counters are meant mostly for debug purposes and are not all supported.

The statistics are divided into three commands:

1. Controller statistics - These are statistics on the primary interface (to the host OS). See the NC-SI specification for details.
2. NC-SI statistics - These are statistics on the NC-SI control frames (such as commands, responses, AENs, etc.). See the NC-SI specification for details.
3. NC-SI pass-through statistics - These are statistics on the NC-SI pass-through frames. See the NC-SI specification for details.

§ §

NOTE: ***This page intentionally left blank.***