

Intel® Sideband Technology

AN OVERVIEW OF THE INTEL SERVER MANAGEABILITY INTERFACES

Intel® LAN Access Division

321786-002EN

Revision 1.10

July 2009

Legal

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel products are not intended for use in medical, life saving, life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

This manual may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This manual as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Chips, Core Inside, Dialogic, EtherExpress, ETOX, FlashFile, i386, i486, i960, iCOMP, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, MMX logo, Optimizer logo, OverDrive, Paragon, PDCharm, Pentium, Pentium II Xeon, Pentium III Xeon, Performance at Your Command, Pentium Inside, skool, Sound Mark, The Computer Inside., The Journey Inside, VTune, Xeon, Xeon Inside and Xircor are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

*Other names and brands may be claimed as the property of others.

Copyright © 2009, Intel Corporation

Revisions

Date	Revision	Description
February 10, 2009	0.9	1st Draft
February 19, 2009	0.95	Added configuration examples..
February 24, 2009	0.96	Added MDEF content
March 16, 2009	0.97	Cleaned examples.
March 17, 2009	0.98	Changed EC to NC
May 5, 2009	1.00	First external release.
July 9, 2009	1.10	Corrected figure that wasn't displaying. Figure 3-5.

Contents

1	Introduction	9
1.1	Management Controller	9
1.2	Remote Monitoring	10
1.2.1	Standards	10
1.2.1.1	IPMI	10
1.2.1.2	WS-Management	11
1.2.1.3	SMASH	11
1.3	Consolidated Ethernet Connectivity	11
2	Sideband Interface	13
2.1	Components of a Sideband Interface	13
2.1.1	Sideband Physical Layer	14
2.1.2	Sideband Protocol Layer	14
2.1.3	Filtering Mechanism	14
2.2	Evolution of the Intel Sideband Interface	15
2.2.1	Pass-Through – First Generation	15
2.2.1.1	Limitations	15
2.2.1.1.1	Gratuitous ARPs	16
2.2.1.1.2	Shared MAC Address	16
2.2.1.1.3	Management Traffic Sent to Host	16
2.2.2	Advanced Pass-Through – Second Generation	16
2.2.2.1	Dedicated MAC Address Support	17
2.2.2.2	Broadcast Filtering	17
2.2.2.3	IP Address Filtering	17
2.2.2.4	RMCP Port Filtering	17
2.2.2.5	Definable Port Filtering	17
2.2.2.6	Flexible Filtering	17
2.2.2.7	Automatic responses to ARP requests	17
2.2.2.8	VLAN Filtering	18
2.2.2.9	XSUM Filtering	18
2.2.3	Super Pass-Through – Third Generation	18
2.2.3.1	Serial Over LAN	18
2.2.3.2	IDE Redirection (Floppy & CD Redirection)	18
2.2.3.3	RMCP+ Session Establishment	18
2.2.3.4	Limitations	19
2.2.4	Fast Management Link– Third Generation	19
2.2.5	Pass-Through – Fourth Generation	20
2.2.6	NC-SI – Fourth Generation	20

3	SMBus Sideband Interface	21
3.1	Overview	21
3.2	SMBus Physical Layer	21
3.2.1	SMBus Clock	21
3.2.2	SMBus Data	22
3.2.3	SMBAlert#	22
3.3	SMBus Protocol Layer	22
3.3.1	Addressing	23
3.3.2	SMBus Notification Methods	24
3.3.2.1	SMBus Alert and Alert Response Method	24
3.3.2.2	Asynchronous Notify Method	25
3.3.2.3	Direct Receive Method	25
3.3.3	SMBus Notification Timeout	26
3.4	Dual-Port Ethernet Controller	26
3.4.1	Addressing	26
3.4.2	Filtering & Configuration	26
3.4.3	Alert Notification	27
3.4.4	Failover Support	27
3.4.4.1	Transmit Functionality	27
3.4.4.2	Receive Functionality	27
3.4.4.3	Port Switching (fail-over)	27
3.4.4.4	Driver Interactions	28
3.4.4.5	Fail-Over Configuration	28
3.4.4.5.1	Preferred Primary Port	28
3.4.4.5.2	164BGratuitous ARPs	28
3.4.4.5.3	165BLink Down Timeout	28
3.5	SMBus Interface Filtering Overview	28
3.5.1	General Filtering Flow	29
3.5.2	Management To Host Filter	30
3.5.3	Shared L2 Filtering	31
3.5.3.1	VLAN Filtering – Incoming Packets	32
3.5.3.2	VLAN Filtering – Outgoing Packets	33
3.5.4	Filtering Algorithm	34
3.5.5	Filtering Configuration	36
3.5.5.1	Old Mechanism	36
3.5.5.2	133BNew Mechanism	36
3.5.5.2.1	Manageability Decision Filters (MDEF)	38
3.5.5.2.2	Management to Host	41
3.5.5.2.3	168BMDEF Filters and Interaction with the Receive Enable Command	42
3.6	Configuration Examples	43

3.6.1	Example 1 – Shared MAC, RMCP Only Ports	43
3.6.1.1	Example 1 - Old Method	44
3.6.1.2	Example 1 – New Method	44
3.6.2	Example 2 – Dedicated MAC, Auto ARP Response and RMCP Port Filtering	45
3.6.2.1	Example 2 - Old Method	46
3.6.2.2	Example 2 – New Method	47
3.6.3	Example 3 – Dedicated MAC & IP Address	49
3.6.3.1	Example 3 - Old Method	49
3.6.3.2	Example 3 – New Method	50
3.6.4	Example 4 – Dedicated MAC and VLAN Tag	53
3.6.4.1	Example 4 - Old Method	53
3.6.4.2	Example 4 – New Method	54
3.7	Sending and Receiving SMBus Packets	56
3.7.1	Receive TCO Packet Command	56
3.7.2	Transmit TCO Packet Command	57
4	NC-SI Interface	59
4.1	Overview	59
4.1.1	Terminology	59
4.1.1.1	Package	59
4.1.1.2	Channel	59
4.1.1.3	AEN	59
4.1.2	Package Options	60
4.2	Physical Layer	60
4.2.1	REF_CK Source	62
4.2.2	Multi-drop Arbitration	62
4.2.2.1	Command-base Arbitration (Software Arbitration)	63
4.2.2.2	Hardware Arbitration	63
4.3	Protocol Layer	63
4.3.1	Overview	64
4.3.2	Traffic Types	64
4.3.2.1	Control Packets	64
4.3.2.2	AEN Packets	64
4.3.3	Addressing	64
4.3.4	Transmit Flow Overview	65
4.3.5	Receive Flow Overview	66
4.4	Differences Between NC-SI and RMII	66
4.5	Basic NC-SI Work-flows	66
4.5.1	Package States	66
4.5.2	Channel States	67
4.5.3	Discovery	67

4.5.4	Configurations.....	68
4.5.4.1	NC Capabilities Advertisement	68
4.5.4.2	Receive Filtering	68
4.5.4.2.1	MAC Address Filtering	68
4.5.4.3	VLAN.....	69
4.5.5	Pass-Through Traffic States.....	69
4.5.5.1	Channel Enable.....	69
4.5.5.2	Network Transmit Enable	70
4.5.6	Asynchronous Event Notifications.....	70
4.5.7	Querying Active Parameters	70
4.5.8	Resets	70
4.6	Advanced Workflows	71
4.6.1	Multi-NC Arbitration	71
4.6.1.1	Example Package Selection Sequence	72
4.6.2	External Link Control.....	72
4.6.3	Multiple Channels (Fail-Over)	73
4.6.3.1	Example Fail-Over Algorithm.....	73
4.6.4	Statistics	74
4.7	OEM Extensions	74
4.7.1	Get System MAC	74
4.7.2	TCO Reset	74
4.7.3	Keep PHY Link Up	74
4.7.4	Checksum Offloading	74
4.7.5	Additional Filtering.....	75
5	Troubleshooting Recommendations	76
5.1	General Troubleshooting	76
5.1.1	Remote Management Connection Dropped After Power Action	76
5.1.2	Byte Order.....	76
5.2	SMBus Troubleshooting.....	76
5.2.1	SMBus Alert Line Stays Asserted After Power Cycle.....	76
5.2.2	SMBus Commands are Always NACK'd by the Intel Ethernet Controller.....	77
5.2.3	Slow SMBus Clock Speed	77
5.2.4	Network Based Host Application Not Receiving Network Packets	78
5.3	NC-SI Troubleshooting.....	78
5.3.1	Verify Electrical Connections.....	78
5.3.2	NC-SI Control Packets	78
5.3.3	Ensure NC-SI Mode.....	78
5.4	Recommendations.....	78
5.4.1	General Recommendations.....	78
5.4.1.1	Default Configuration.....	78
5.4.2	SMBus Recommendations	79

5.4.2.1	Dedicated SMBus for Manageability.....	79
5.4.2.2	SMBus Fragment Size	79
5.4.2.3	Enable XSUM Filtering.....	79
5.4.3	NC-SI Recommendations	79
5.4.3.1	Use Hardware Arbitration	79
6	Manageability Registers	80
6.1	Manageability Control Register (MANC)	80
6.2	Management To Host (MNG2Host)	81
6.3	Manageability Decision Filters (MDEF & MDEF_EXT)	82
7	Additional Documentation	85

1 Introduction

This document provides an overview for the manageability side-band interface available in Intel Ethernet Controllers for servers. The side band interface provides a mechanism by which dedicated management controllers (MCs) can send and receive Ethernet data.

1.1 Management Controller

In nearly all modern servers, there is a processor dedicated to managing the server. Such dedicated processors are called by many names. Among these are: Management Controllers (MC), Management Controllers (MC)¹ and IPMI Management Controllers (IPMC). This document consistently uses the MC term.

No matter what name is used, the MC's purpose is to monitor the health of the server. This is almost always done by monitoring sensors. These include temperature, voltage, and fan sensors, to name just a few.

In addition to sensors, there are logs which record key activities (such as power actions, events defined as sensor events, and opening the chassis). Using sensors and the logs, it is possible to do predictive failure. For example, if a fan over a period of time becomes slower, the issue may be an indication that the fan is failing.

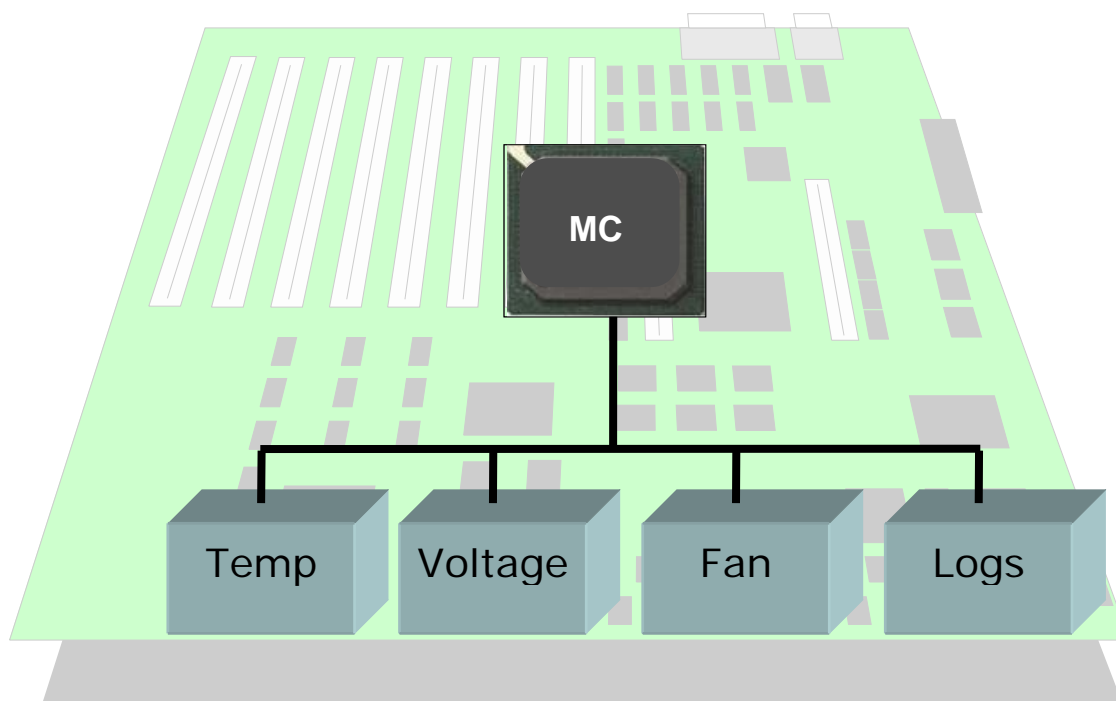


Figure 1-1. Management Controller

¹ Also referred to as BMCs (baseboard management controllers)

MCs can come with the ability to perform power actions (power up, power down and reset) and can usually perform autonomous tasks based upon rules. For example, if the temperature sensor on a processor crosses a critical threshold level, an MC can usually turn off the system in order to prevent system from being damaged.

More advanced MCs provide features such as embedded web interfaces, KVM redirection, Remote Media (USB, CD-ROM), FTP, remote updates of BIOS and more.

1.2 Remote Monitoring

While MCs are always available to the server OS through a host interface, the real strength of server management is the ability to access MCs remotely.

Consider a typical datacenter – it may have hundreds of servers running. In such an environment, it is not realistic to go to each system and manually access individual MCs from local OSs. In addition, a particular server may be in a failing state (making access to the OS impossible) or powered down.

For the above reasons, it is desirable to access MCs remotely via an Ethernet connection. Utilizing the Ethernet connectivity, MCs can be accessed utilizing a defined protocol to perform a multitude of tasks.

1.2.1 Standards

This section provides a short overview of the more common remote monitoring standards.

1.2.1.1 IPMI

The Intelligent Platform Management Interface (IPMI) specification defines a set of common interfaces to a computer system which system administrators use to monitor system health and manage the system. Dell, HP, Intel Corporation and NEC Corporation announced IPMI v1.0 on 1998-09-16, v1.5 on 2001-03-01, and v2.0 on 2004-02-14.

IPMI operates independently of the OS and allows administrators to manage a system remotely even without an OS, system management software, and even if the monitored system is powered off (along as it is connected to a power source). IPMI can also function after an OS has started, offering enhanced features when used with system management software.

IPMI prescribes the structure and format of interfaces. Detailed implementations may vary. An implementation of IPMI (version 1.5 and later) can send out alerts via a direct serial connection, a local area network (LAN) or a serial over LAN (SOL) connection to a remote client. System administrators then use IPMI messaging to query platform status, to review hardware logs or to issue other requests from a remote console through the same connections.

The IPMI standard also defines an alert mechanism for the system to send a simple network management protocol (SNMP) platform event trap (PET).

A typical IPMI implementation consists of a main controller called the Management Controller (MC) and satellite controllers. The satellite controllers within the same chassis connect to the MC using a system interface called IPMB (Intelligent Platform Management Bus/Bridge) — an enhanced implementation of I²C (Inter-Integrated Circuit).

MCs connect to satellite controllers or other MCs in another chassis via IPMC (Intelligent

Platform Management Chassis) bus/bridge.

A Field Replaceable Unit (FRU) holds the inventory (such as vendor id, manufacturer etc.) of potentially replaceable devices. A Sensor Data Records (SDR) repository provides the properties of the individual sensors present on the board. For example, the board may contain sensors for temperature, fan speed, and voltage.²

1.2.1.2 WS-Management

WS-Management is a specification of a SOAP-based (Simple Object Access Protocol) protocol for the management of servers, devices, applications and more. The specification is based on DMTF open standards and Internet standards for Web Services and was published in March, 2005 by a group of companies, including AMD, Dell, Intel, Microsoft, Sun Microsystems and others.

WS-Management provides a common way for systems to access and exchange management information across an IT infrastructure. The specification is rich, supporting much more than get/set of simple variables. It is closer to WBEM or Netconf than to SNMP.

A mapping of the DMTF-originated Common Information Model (CIM) into WS-Management is also defined.³

1.2.1.3 SMASH

The Systems Management Architecture for Server Hardware (SMASH) is a suite of specifications that deliver industry standard protocols to increase productivity of the management of a data center.

The SMASH Command Line Protocol (SM CLP) specification provides an interface to servers independent of machine state, OS, OS state, system topology or access method. It is a standard method for local and remote management of server hardware using out-of-band communication.

SMASH is being developed by the Distributed Management Task Force (DMTF) Server Management Working Group (SMWG).⁴

1.3 Consolidated Ethernet Connectivity

Some of the first servers with remote manageability used a dedicated Network Controller (NC). While this provided a connectivity mechanism, it came at a cost of resources.

A more elegant solution has been provided using a mechanism that passes manageability traffic to and from MCs using LOM (LAN on Motherboard; built into the server motherboard). This is accomplished when a Network Controller (NC) examines incoming traffic and, based upon a filtering, determines if the packet is destined for the OS or for an MC.

² http://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface

³ <http://en.wikipedia.org/wiki/WS-Management>

⁴ http://en.wikipedia.org/wiki/Systems_Management_Architecture_for_Server_Hardware

The filtering mechanism and the physical connection between the Ethernet Controller and the MC are called by several names (including the Total Cost of Ownership (TCO) port, management link, pass-through interface and the Sideband Interface). In this document, this technology layer is called the Sideband Interface.

The Sideband interface is the focus of this document, specifically as it pertains to the features and capabilities of Intel Server based Ethernet Controllers.

2 Sideband Interface

In order to provide a cost-effective mechanism which allowed MCs Ethernet connectivity, manufacturers added a new interface to the Ethernet Controllers. This interface was designed to be simple and used only for manageability traffic. See Figure 2-1.

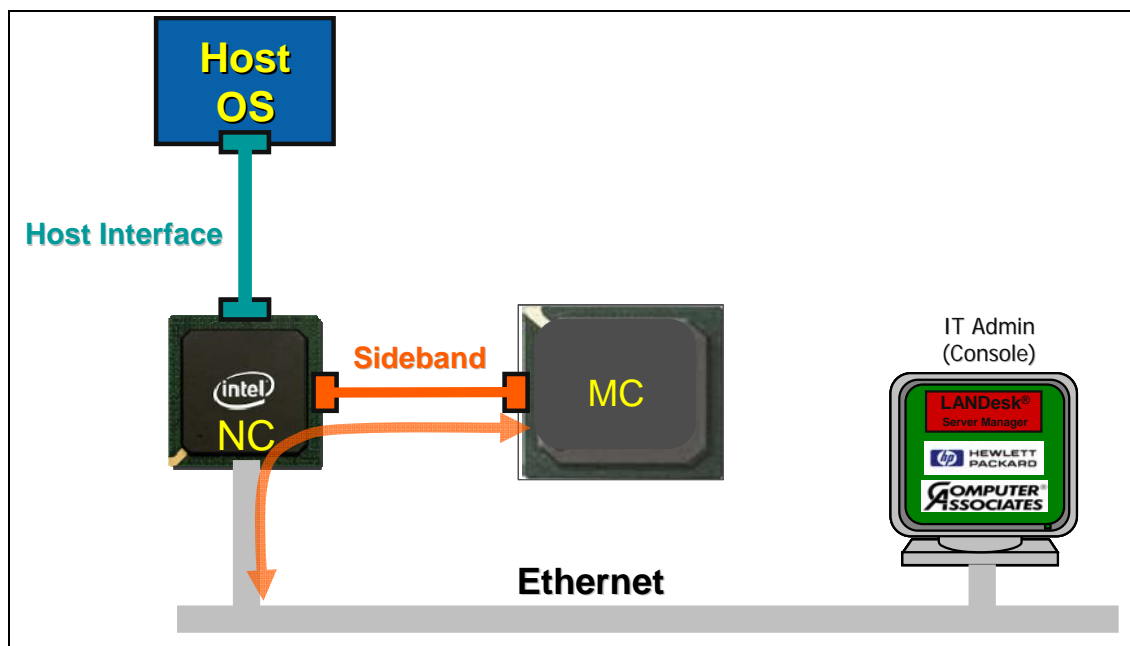


Figure 2-1. Sideband Interface to MC

The first implementation of the Sideband Interface on Intel Ethernet Controllers was simple, with limited filtering capabilities and speed. The latest generation has the ability to run at 100 Mb/s Full-Duplex speeds and a highly configurable filtering mechanism.

Details are provided later.

2.1 Components of a Sideband Interface

The Sideband interface consists of two key ingredients:

The Physical Layer

- The physical electrical connection between the NC and the MC

The Protocol Layer

- The agreed upon communication protocol between the NC and the MC

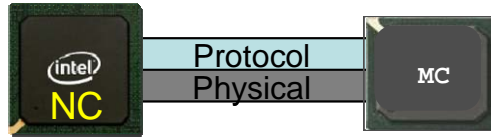


Figure 2-2. Sideband Interface

2.1.1 Sideband Physical Layer

The physical layer portion of a Sideband Interface consists of the electrical connection between the Ethernet Controller and the MC. The electrical and timing requirements differ from one Sideband Interface to another. Make sure to consult the product Datasheet for specifics.

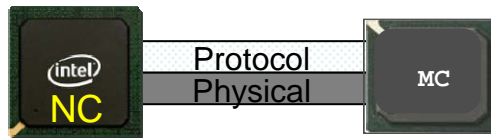


Figure 2-3. Sideband Physical Layer

2.1.2 Sideband Protocol Layer

The protocol layer is an agreed upon 'language' that the NC and the use MC 'speak' to each other. It defines the form in which Ethernet Packets will be sent and received and also how the MC can sent configuration commands to the NC.

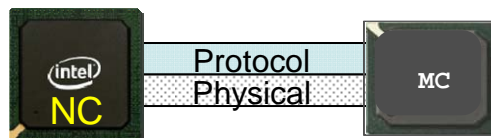


Figure 2-4. Sideband Protocol Layer

2.1.3 Filtering Mechanism

When an Ethernet packet comes into the NC, it is examined to determine if it is a packet that should be sent to the host or to the MC. This decision is based upon filters that are configurable by the MC over the Sideband Interface.

2.2 Evolution of the Intel Sideband Interface

The Sideband Interface available in Intel Ethernet Controllers for servers has progressed from a relatively slow interface with limited filtering capabilities to a robust very fast interface for the MC to send and receive Ethernet Traffic.

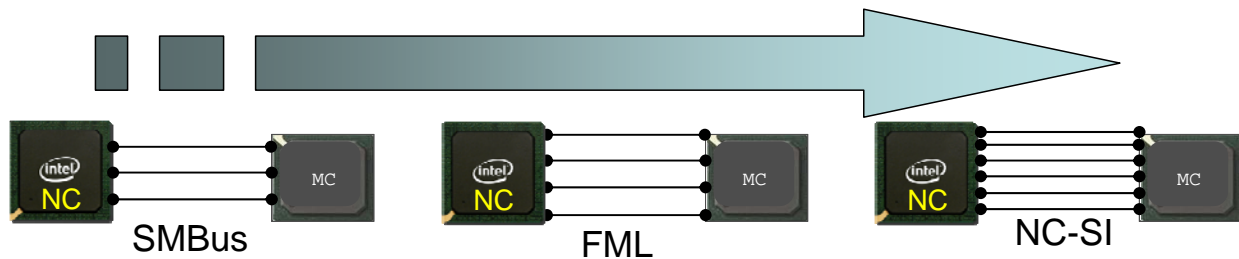


Figure 2-5. Intel Sideband Interface Evolution

2.2.1 Pass-Through – First Generation

The first generation of the Sideband interface was designed for IPMI traffic only. The IPMI 1.5 specification indicates that RMCP (IPMI over LAN packets) traffic should be sent to UDP port 26Fh. Port 26Fh is an IANA reserved number specifically for RMCP Ethernet traffic.

This generation Sideband interface was a very simple, with the physical layer using the SMBus specification and the protocol layer being an Intel proprietary specification detailed in the specifications for the Ethernet Controller.

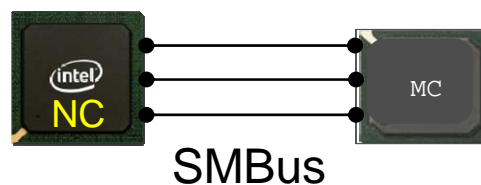


Figure 2-6. SMBus Interface

The filtering mechanism was also extremely simple – only Ethernet packets that had a destination UDP port of 26Fh were sent to the MC. These packets were also sent to the host network as well.

2.2.1.1 Limitations

While the first implementation of the sideband interface allowed a MC to send and receive Ethernet traffic, it was limited in its capabilities. This section details some limitations.

2.2.1.1.1 Gratuitous ARPs

With this interface, the MC was able to transmit any selected packets; however, it could only receive packets destined for UDP port 26Fh. This resulted in the MC having to transmit gratuitous ARPs at a regular interval.

Usually in Ethernet topology, a device will send an ARP request in order to associate a destination MAC address with an IP address. When an ARP request is received by a device, if that device 'owns' the IP address within the request, it will send an ARP response back with its MAC address.

In the 1st generation of the Intel Sideband interface, broadcast packets were not sent to the MC – only packets with a destination UDP port address of 26Fh were passed to the MC. As such, in order to allow other devices on the Ethernet to know the MC's MAC address, the MC had to send gratuitous ARPs at a regular interval – usually every 2 seconds.

While this does not generate a significant amount of Ethernet traffic from a single server – consider a data center with hundreds or even thousands of servers. Then bandwidth becomes problem.

2.2.1.1.2 Shared MAC Address

Today, many MCs have a dedicated MAC address. This allows them to receive all Ethernet traffic with that MAC address as the destination. First generation Intel Sideband interfaces shared their MAC addresses with the host.

2.2.1.1.3 Management Traffic Sent to Host

A side-effect of the way the manageability traffic filtering was done with a first generation device was that packets sent to the MC were also sent to the host.

Since the host almost never had an application listening on port 26Fh (that's an IANA reserved port number for RMCP traffic), the host networking stack on some OSs would send back a 'port unreachable' error to the remote console.

This resulted in the situation where the remote console would send a packet, the MC would respond, and the server host would respond with an error. Management software would have to be programmed to recognize this possibility and handle it appropriately.

2.2.2 Advanced Pass-Through – Second Generation

The Second Generation Sideband Interface is backwards compatible with the First Generation. It uses SMBus for the physical layer and still supports many of the original configuration commands.

The next iteration of the Sideband Interface, sometimes referred to as 'Advanced Pass-Through' in Intel documentation, provided a much richer set of capabilities. These include:

- Dedicated MAC address support
- Broadcast Filtering
- IP Address Filtering
- RMCP Port Filtering
- Flexible Port Filtering
- Flexible Filtering
- Automatic responses to ARP requests

2.2.2.1 Dedicated MAC Address Support

This feature provides the MC with the ability to have its own MAC address. The MC issues a command over the SMBus interface to the NC, indicating that it has its own MAC address and that all packets with that MAC address should be sent to the MC.

2.2.2.2 Broadcast Filtering

Broadcast filtering enabled the MC to receive broadcast packets. In addition, there are filters for specific types of broadcast packets:

- ARP Requests
- ARP Responses
- Network Neighborhood Discovery packets

2.2.2.3 IP Address Filtering

The MC can request to the NC that only directed (non-broadcast) packets matching a specific IP address be sent to the MC. The MC can perform its own DHCP actions or use a statically assigned IP address.

2.2.2.4 RMCP Port Filtering

There are two specific RMCP ports, 26Fh and 298h. Both are registers with IANA (Internet Assigned Number Authority) for RMCP support. Port 26Fh is for standard RMCP traffic while 298h is defined as the Secure Secondary RMCP port.

There is a filter for each of these ports.

2.2.2.5 Definable Port Filtering

In addition to defined RMCP ports, the MC can also define filters for ports of its own choosing.

For example, an MC could support IPMI and an embedded web browser. It could enable RMCP filtering and define flexible port filters for HTTP and HTTPS ports.

2.2.2.6 Flexible Filtering

This filter enables the MC to define a 128-bit filter and corresponding filter mask. Only incoming packets matching the filter would be passed to the MC from the NC.

2.2.2.7 Automatic responses to ARP requests

When a dedicated MAC address is used by the MC, this allows the MC to instruct the NC to automatically respond to ARP requests for the MCs MAC address with a specific IP address. This relieves the MC of having to perform the task; it automatically be done by the NC.

2.2.2.8 VLAN Filtering

The MC is able to configure VLAN tags which the NC will use to filter incoming packets.

2.2.2.9 XSUM Filtering

The NC examines all incoming packets and drops all packets with errors - except for checksum errors.

The MC can validate all incoming packets for checksum errors or it can enable the XSUM filter, which performs checksum validation within the NC.

If the filter is enabled, the NC detects errors; if an error is detected, it drops the packet.

2.2.3 Super Pass-Through – Third Generation

Super Pass-Through was the term given to a feature set that was only available for a single generation of Intel Ethernet Controllers for servers. Super Pass-Through has the same features as Advance Pass Through with the added ability to be able to perform autonomous Serial over LAN (SOL) and IDE Redirection (IDE-R).

These features are only available on the Intel ESB2 and 82571 GbE controllers

2.2.3.1 Serial Over LAN

SOL is a protocol that redirects a local serial connection (COM port) over the network. Its primary use is to redirect a text screen and keyboard to a remote console. The SOL protocol is encapsulated inside an RMCP+ session, supporting both authentication and encryption. The SOL protocol is part of the IPMI 2.0 specification (see the IPMI site for details).

The NC presents a 'virtual' COM port to the PCI bus, the system BIOS then uses this virtual COM port to redirect all text output and receive keystrokes. The NC is able to autonomously establish a RMCP+ (IPMI 2.0) session with a remote console and perform SOL with text data sent to it from the BIOS.

2.2.3.2 IDE Redirection (Floppy & CD Redirection)

The Intel ESB2 and the 82571 Ethernet controllers have the ability to perform autonomous SOL. In addition they can also be used to achieve remote simulation of a local IDE floppy & CD ROM drive.

The controller can remotely control a system by redirecting that system's IDE controller to another remote system using the network. The remote system can have an IDE CDROM and/or IDE floppy device attached (usually attached to the management terminal window on a remote control system); or the remote system might contain files that have a full copy of a CDROM or a floppy (ISO files).

While this capability does exist, it is not currently supported.

2.2.3.3 RMCP+ Session Establishment

Both SOL and IDE-R work using RMCP+ sessions (connections). There are two ways in which such a session can be established.

By the MC

By the NC

If the MC establishes the RMCP+ session (as it would any other IPMI 2.0 session), it can

'hand-off' or 'give' that session over to the NC once the remote console issues the command to begin the SOL or IDE-R traffic. This 'hand-off' is accomplished using a number of commands available to the MC over the Sideband Interface. Once the session is owned by the NC, all traffic on that session is sent only to the NC, none is sent to the MC.

The NC is also capable of establishing the RMCP+ session. It must be configured with necessary pieces of information such as passwords, usernames, etc. This information can come from EEPROM settings, or the MC can provide this information over the Sideband Interface.

2.2.3.4 Limitations

Only two NCs (ESB2 and 82571) support Super Pass Through correctly. Only the SOL functionality is fully supported.

The IDE-R feature works correctly, however it does so using an Intel proprietary protocol over RMCP+ and requires a now unsupported Windows based management client to work correctly.

The Serial Over LAN functionality is fully IPMI 2.0 compliant and will work with most IPMI 2.0 SOL utilities with limitations. When the NC is performing SOL, it only 'understands' the commands to establish a RMCP+ session and those commands pertaining directly to SOL. If the remote console sends a different command, such as 'Get Chassis Status', the NC will silently drop this command. If remote software relies on such a non-SOL specific command as a check to ensure the session is still active, the remote software may fail.

In addition, if the NC is the one configured for establishing the RMCP+ session, no traffic will ever be sent over the sideband interface to the MC – the fact is that there does not even need to be a MC present for this functionality to work. This may be desirable for some customers, such as those in the ATCA environment, who want to be able to perform SOL to a blade, but not have any other network traffic go to the MC (as they are managed over a different blade interface).

In general, the SOL and IDE-R functionality was an interesting concept, however since SOL is now part of the IPMI 2.0 specification and many MCs already implement some sort of IDE Redirection, these features usually have not been worth the challenges required to integrate them with the MC.

2.2.4 Fast Management Link– Third Generation

The Sideband Interface technology up until this point was all done over the SMBus interface. It is also fairly slow. It is too slow for SOL or embedded web server applications.

As a potential solution, Intel came up with the Fast Management Link (FML). FML is only available on two Intel Ethernet Controller, the ESB2 and the 82571 GbE controllers.

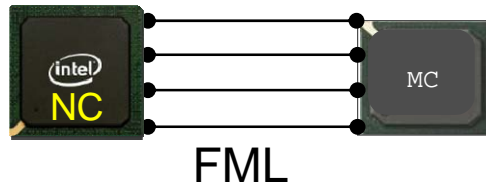


Figure 2-7. Fast Management Link

FML took SMBus electricals and modified them. Instead of a shared Send/Receive line as in SMBus, there is a Send and a Receive line, making FML a full-duplex interface. In addition, the clock speed was increased (8 MHz; as opposed to SMBus running at 10-400 KHz). The protocol layer is exactly the same as SMBus.

2.2.5 Pass-Through – Fourth Generation

Current Intel controllers for servers still have SMBus sideband. This interface has been expanded to include a larger number of filters and has implemented a simpler mechanism for defining and enabling the filters.

In general, all the capabilities that were present in Advanced Pass-Through are still available in this interface. Only the way the filters are enabled has changed. The new filtering mechanism provides a number of configurable filter sets, where each set can have multiple filters (such as RMCP ports, VLAN tags, MAC address etc).

There are new capabilities as well (such as filtering on Ether Type and the ability to control link during a reset).

2.2.6 NC-SI – Fourth Generation

The latest generation of Intel Ethernet Controllers for servers implements the industry standard DMTF Network Controller Sideband Interface (NC-SI). Intel was involved in the development of this specification and has worked with a number of hardware and firmware vendors to ensure that the interface works as expected.

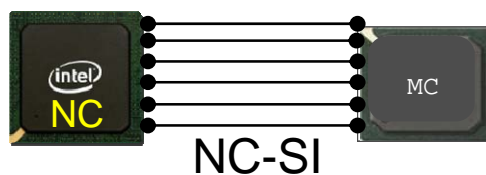


Figure 2-8. NC-SI Interface

NC-SI is much faster than SMBus; it runs at 100Mb/s full-duplex, transferring entire Ethernet frames to and from the MC - making it a highly robust and capable interface.

NC-SI allows for up to four physical Ethernet Controllers to share the NC-SI lines. Each NC can have multiple ports.

3 SMBus Sideband Interface

All Intel Ethernet Controllers with a manageability Sideband interface support the SMBus Sideband interface. Some NC's also support the DMTF defined NC-SI specification as well.

This section provides an overview of the SMBus sideband interface, covering the electrical requirements as well as information regarding the protocol or commands.

See the SMBus specification at: <http://smbus.org/specs/smbus20.pdf> .

3.1 Overview

The System Management Bus (abbreviated to SMBus or SMB) is a simple two-wire bus, derived from I²C and used for communication with low-bandwidth devices on a motherboard that might include temperature, fan, or voltage sensors; and lid switches.

SMBus was defined by Intel in 1995. It carries clock, data, and instructions and is based on Philips I²C serial bus protocol. Its clock frequency range is 10 kHz to 100 kHz. (some Intel Ethernet Controllers extend this to 400 kHz.) Voltage levels and timings are more strictly defined than those of I²C, but devices belonging to the two systems (I²C and SMBus) are often successfully mixed on the same bus.

SMBus is mostly a subset of I²C. The SMBus has an extra optional shared interrupt signal called SMBALERT#, which can be used by slaves to tell the master to ask its slaves about events of interest.

3.2 SMBus Physical Layer

The physical layer of a Sideband Interface consists of the electrical requirements for the lines connecting the MC and the NC.

I²C and SMBus use only two bidirectional open-drain lines, Serial Data (SDA) and Serial Clock (SCL), pulled up with resistors. SMBus adds the optional SMBALERT# line which is open drain as well, requiring a pull up resistor. Typical voltages used are +5 V or +3.3 V although systems with other, higher or lower, voltages are permitted.

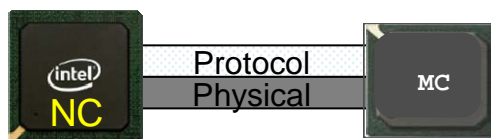


Figure 3-1. SMBus Physical Layer

3.2.1 SMBus Clock

The SMBus clock (SCL) is provided by the SMBus master, which in the case of manageability is the MC.

The default clock speed is 100 KHz; some of the NC's have support for a faster clock of 1 MHz. On those Intel Ethernet controllers supporting the faster clock speed, the configuration

for the use of this faster clock speed is done within the EEPROM image.

3.2.2 SMBus Data

SMBus Data (SDA) can be driven by either the MC or the NC. In general, after the Sideband Interface is configured, the MC usually only issues the 'Read Data' command and the NC in turn will write either status or an incoming Ethernet frame.

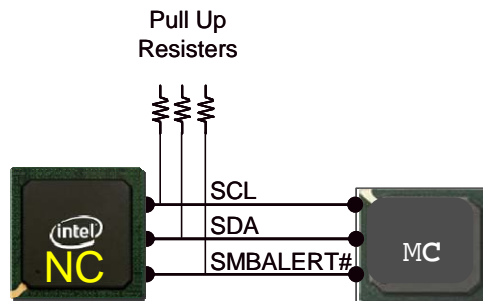


Figure 3-2. SMBus Physical Layer

3.2.3 SMBAlert#

In general, the SMBus for the Sideband Interface is defined so that there is a master and a slave. Only the master may initiate a transaction. This means that the master (the MC) would be required to poll the NC for data, which is of course not a very efficient mechanism. The SMBALERT# line is provided to allow the slave device (the NC) to signal the MC that it needs attention. The MC should monitor this line, when it is asserted (pulled low), the MC should go find out why the NC asserted the alert line.

3.3 SMBus Protocol Layer

The Protocol Layer of a Sideband Interface describes the agreed upon commands that the MC and NC use for communication. For the SMBus Sideband Interface, there are a number of basic SMBus defined commands and requirements that are built upon for the use as a Sideband Interface.

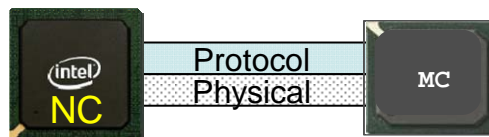


Figure 3-3. SMBus Protocol Layer

Table 3-1 lists the format of a typical SMBus transaction:

Table 3-1. Typical SMBus Transaction

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P

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 below, and are fully defined in the SMBus specification:

Table 3-2. Shorthand Field Names

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)

3.3.1 Addressing

The SMBus protocol is designed to have a single master (the MC) and multiple slaves. As such, each slave device requires a 1-byte address that is unique to that SMBus. While the SMBus specification is designed for multiple slaves, Intel highly recommends that for the purposes of a Sideband Interface, that the only devices on the SMBus are the MC and the Intel NC.

The SMBus address that the Intel NC responds is defined within the EEPROM for the NC. The Intel Ethernet Controllers all support the SMBus Address Resolution Protocol (ARP) functionality, which allows a MC to re-assign the address the NC responds to. By far, the majority of MC firmware vendors usually hard-code the NC slave addresses based upon the SMBus slave addresses in the EEPROM.

3.3.2 SMBus Notification Methods

The Intel NC Sideband Interface supports three methods of notifying the MC that it has information that needs to be read by the MC when using the SMBus:

- SMBus Alert
- Asynchronous Notify
- Direct Receive

The notification method used by the Intel NC 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. Please refer to the EEPROM Guide for a specific NC more information.

The following events cause the Intel NC to send a notification event to the MC:

- Receiving a LAN packet that is designated to the MC.
- Receiving a Request Status command from the MC initiates a status response.
- Status change has occurred and the NC is configured to notify the MC of the status changes.

While the MC firmware and hardware can choose from any of the supported notification methods, the simplest and most efficient (and highly recommended) method is to use the SMBus Alert mechanism. With this mechanism, the MC can act only as a master, with the other two mechanisms the MC must also be able to act as a SMBus slave.

3.3.2.1 SMBus Alert and Alert Response Method

The SMBus Alert# signal is an additional SMBus signal that acts as an asynchronous interrupt signal to an external SMBus master. The Intel NC asserts this signal each time it has a message that it needs the MC 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 Intel NC can be connected on the same alert pin. As a result, the MC needs a mechanism to distinguish between the alert sources.

The MC can respond to the alert by issuing an ARA cycle that detects the alert source device. The Intel NC responds to an ARA cycle (if it was the SMBus alert source) and de-asserts the alert when the ARA cycle is completes. Following the ARA cycle, the MC issues a read command to retrieve the NC message.

The ARA (Alert Response Address) mechanism is defined within the SMBus specification. Some MCs do not implement the ARA cycle transaction. These MCs respond to an alert by issuing a Read command to the Intel NC (C0h/D0h or DEh). The NC 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 Intel NC 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. The ARA cycle is a SMBus Receive Byte transaction to SMBus Address 18h. Note that the ARA transaction does not support PEC. The Alert Response Address transaction format is shown in the figure below:

Table 3-3. 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	

3.3.2.2 Asynchronous Notify Method

When configured using the Asynchronous Notify method, the Intel NC acts as a SMBus master and notifies the MC 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.

Table 3-4. Asynchronous Notify Command Format

1	7	1	1	7	1	1	
S	Target Address	Wr	A	Sending Device Address		A	...
	MC 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.

3.3.2.3 Direct Receive Method

If configured, the Intel NC has the capability to send a message it needs to transfer to the MC as a master over the SMBus instead of alerting the MC 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 MC in the Block Read command. The opcode that the NC 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.

Table 3-5. Direct Receive Transaction Format

1	7	1	1	1	1	6	1	
S	Target Address	W r	A	F	L	Command	A	...
	MC 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	

3.3.3 SMBus Notification Timeout

There are cases where the MC is hung and not responding to SMBus notification. The Intel NC has a time-out value defined in the EEPROM to avoid hanging while waiting for the notification response. If the MC does not respond when the timeout expires, notification is de-asserted.

Note: The SMBus notification time-out value can only be set in EEPROM and the MC can not modify this value. Please refer to the product Datasheet for details.

3.4 Dual-Port Ethernet Controller

The majority of Intel Ethernet Controllers supporting the Sideband Interface are dual-port devices. Each port is capable of sending and receiving Sideband traffic.

3.4.1 Addressing

Each port has its own SMBus slave address. The default address is stored in EEPROM; the MC can use the SMBus ARP (Address Resolution Protocol) to dynamically assign slave addresses to each port.

3.4.2 Filtering & Configuration

Each port is configured independently. One port could be configured to handle only RMCP traffic using a shared MAC address, while a second one could be configured with a dedicated MAC and IP address capable of HTTP and FTP traffic.

When the MC sends the configuration commands, the Slave Address field of the message

indicates its target port.

3.4.3 Alert Notification

The SMBALERT# notification mechanism uses a shared alert line. This means that if either port asserts the alert line, the MC must determine which one raised it and service the correct requester. This can be done a couple of ways. The MC can simply issue a read to each port (SMBus Slave), or it can use the SMBus ARA (Alert Response Address) cycle to determine which port asserted the alert line.

There are times when both ports will assert the alert line. In such cases, the MC must go read from each port (SMBus Slave) in order for the alert line to be de-asserted.

Alternatively, the alert can also de-asserted with the SMBus timeout period has elapsed. See Section 3.3.3.

3.4.4 Failover Support

Intel dual-port Ethernet Controllers supporting the Sideband Interface provide a failover mechanism that can be configured to failover from one port to another in the event of a port failure.

Note: Automatic failover of the Sideband Interface is only valid when there is no driver.

In teaming mode, the Intel NC mirrors both the network ports to a single SMBus slave device. The NC will automatically handle configurations of both network ports. Thus, for configurations, receiving & transmitting the MC should consider both ports as a single entity.

When the currently active port for transmission becomes unavailable (for example, the link is down), the Intel NC will automatically try to switch the packet transmission to the other port. Thus, as long as one of the ports is valid, the MC will have a valid link indication for the SMBus slave.

3.4.4.1 Transmit Functionality

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

3.4.4.2 Receive Functionality

When the Intel NC receives a packet from any of the teamed ports, it will notify and forward the packet to the MC.

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

3.4.4.3 Port Switching (fail-over)

While in teaming mode, transmit traffic is always transmitted by the Intel NC through only one of the ports at any given time. The NC may 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.

3.4.4.4 Driver Interactions

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

Note: When the driver releases teaming mode (for example, when system state changes), the Intel NC reconfigures the LAN ports to teaming mode. The NC 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 transmission of gratuitous ARP packets to notify the network of teaming mode re-setting.

3.4.4.5 Fail-Over Configuration

Fail-over operation is configured through the fail-over register, as described in the documentation for specific Intel NCs.

The MC should configure this register after every initialization indication from the NC (i.e. after every the NC firmware reset). The MC needs to use Update Management Receive Filters command.

Note: In teaming mode both ports should be configured with the same receive manageability filters parameters.

3.4.4.5.1 Preferred Primary Port

The MC may choose one of the network ports (LAN0 or LAN1) as a preferred primary port for packet transmission. The Intel NC will use the preferred primary port as the transmission port whenever the link for that port is valid.

The NC will always switch back to the preferred primary port when available.

3.4.4.5.2 16BGratuitous ARPs

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

3.4.4.5.3 165BLink Down Timeout

The MC can control the timeout for a link to be considered invalid. The Intel NC will wait for this timeout period before attempting to switch from an inactive port.

3.5 SMBus Interface Filtering Overview

This section provides an overview of how the filtering mechanism determines which incoming frames are routed to the MC instead of the host OS.

There are a number of possible filters that can be used in combination:

- MAC Address Filter (Shared or Dedicated)
- Broadcast Packet Filter
- VLAN Filter

- IPv4 or IPv6 Address Filter
- Multicast Packet Filter
- ARP Filter (Request or Response)
- Neighborhood Discovery Filter
- RMCP Port (298h/26Fh) Filter
- Flexible Port Filter
- Flexible 128-Byte Filter
- XSUM Filter
- Ether Type Filter

Note: Not all of these filters are available on all Intel Ethernet Controllers. Please refer to product documentation for details.

3.5.1 General Filtering Flow

From a high-level, the filtering mechanism is simple. As an Ethernet Packet comes into the Intel Ethernet Controller, it is examined and compared to a set of filters. If the packet matches one or more of the filters, it is sent to the MC. If the packet does not match any of the filters, it is sent to the host OS.

While the number, types and configuration mechanism for the filters has evolved through the years, Figure 3-4 provides an overview of the basic filtering mechanism. A packet comes into the NC, is examined and compared against a number of filters, then sent to the MC or the host based on the results of the filter comparisons.

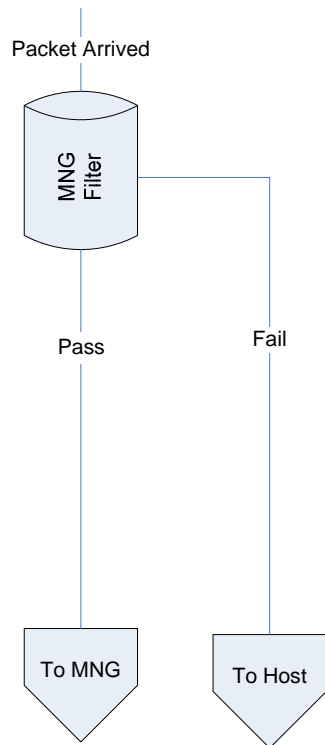


Figure 3-4. General Filtering Flow

3.5.2 Management To Host Filter

The general rule is that if an incoming Ethernet packet is determined to be a manageability packet, it is sent only to the MC.

There are times, however, when it is a requirement that Ethernet traffic be sent to both the MC and Host OS. Examples of this include:

- ARP Requests
- General Broadcast Packets
- Multicast Packets

ARP Requests are the most important and for many implementations, the packets that must be sent to both the MC and the Host OS.

ARP requests are broadcast when a network application needs to determine the MAC address associated with an IP address. If the MC enables a filter to receive ARP requests, so that it may respond to them, then by default the host OS will not receive any ARP requests. Doing this can effectively prevent network applications from communicating with the host OS –not usually a desired result.

There is an additional filter, called the Manageability To Host filter. This filter allows the MC to specify that some of the Ethernet traffic it receives should also be sent to the host OS. Most Intel Datasheets call this filter MNG2HOST.

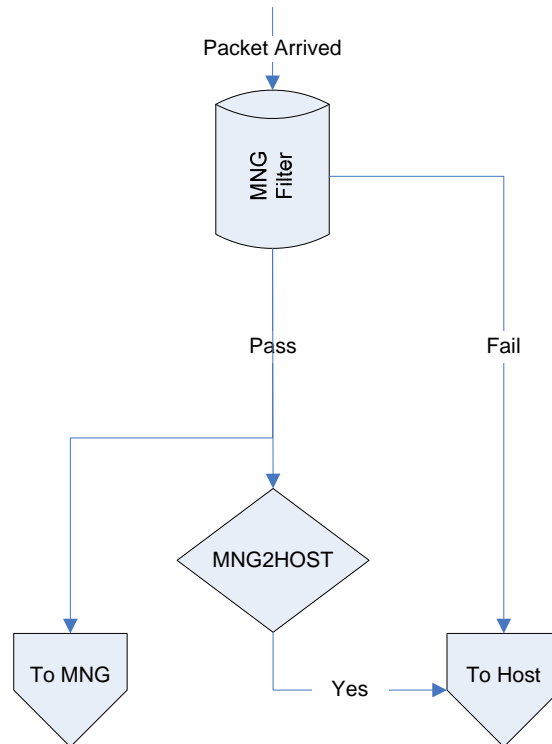


Figure 3-5. General Filtering - with MNG2HOST

After an incoming packet has passed all of the Manageability filtering, it is compared against the Manageability to Host Filter. If the packet matches filters setup by the MC for traffic that should go to both the MC and the Host, it is then sent to both locations.

3.5.3 Shared L2 Filtering

Layer 2 filtering includes:

- MAC Address
- Unicast
- Multicast
- Broadcast
- VLAN tag

The L2 filtering mechanism in the NC is currently shared with host traffic for Intel Ethernet Controllers. This means that when a packet comes in, it is compared against L2 filters for both the host and the MC. If there is a match, the packet is then sent on to the next level of filtering.

This occurs if the MC configures either a shared or dedicated MAC address for manageability.

3.5.3.1 VLAN Filtering – Incoming Packets

Intel Ethernet Controllers have the ability to filter on VLAN tags; this capability is available to the host OS driver as well as the MC. When VLAN filtering is enabled for the host, the Intel Ethernet Controller will strip the VLAN tag from the incoming Ethernet frame.

Since the Layer 2 filtering mechanism is common for the host and manageability, when VLAN filtering is enabled for the host and the MC, incoming Ethernet packets destined for the MC will also have the VLAN tag stripped.

If the host driver is not configured to do VLAN filtering or if there is no OS driver loaded (such as a pre-boot state), then the Ethernet frames will be passed to the MC with the VLAN tags intact.

When the Ethernet packets destined for the MC have the VLAN tag stripped, the MC is notified of this with indications within the received packet status.

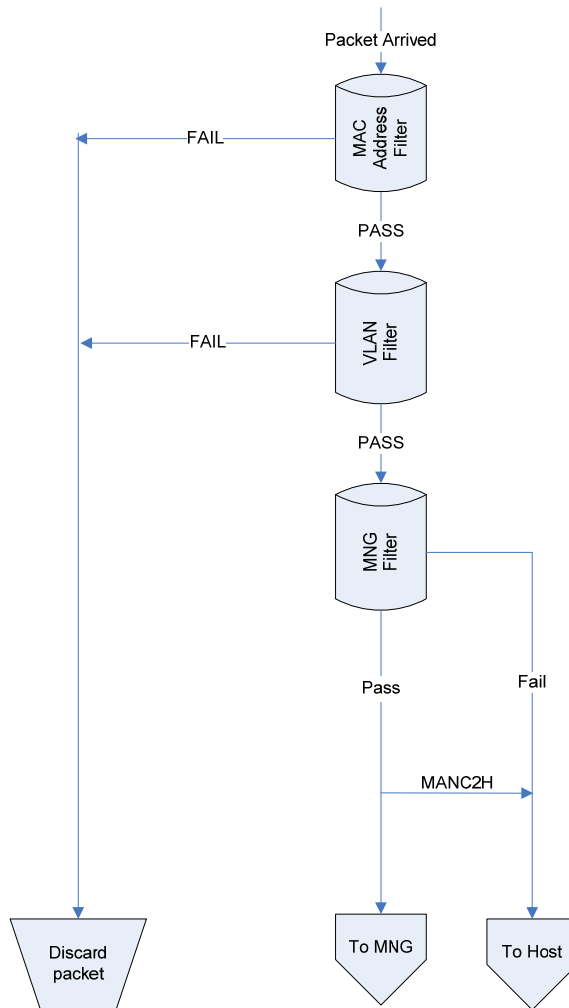


Figure 3-6. Packet Filtering

3.5.3.2 VLAN Filtering – Outgoing Packets

Under some circumstances, incoming manageability Ethernet Packets can have a VLAN tag stripped by the filtering hardware within the NC; the reverse is not true. Intel Ethernet Controllers do not provide the ability to insert VLAN tags in outgoing packets.

The MC is responsible for creating the contents of outgoing packets, including VLAN tags if required.

The only exception to this is for those Intel Ethernet Controllers that support the Super Pass-Through mode. Even on such controllers, the VLAN tag is inserted for outbound traffic when the NC itself generates the traffic in the case of Autonomous Serial Over LAN and IDE Redirection.

3.5.4 Filtering Algorithm

There are several stages of filtering.

The first is described in Figure 3-6. An incoming packet is compared against one or more MAC addresses (some Intel NC's allow the configuration of multiple MAC addresses for manageability); then VLAN tagging is examined.

If the incoming packet passes both the MAC and VLAN filters, it is passes to the next level of filtering, described in Figure 3-7.

The first check at the next level is to see if the interface has been enabled to receive traffic (this is done with the Receive Enable command). If it has been enabled, then a check is done to see if specific filtering has been enabled or if 'receive all' has been enabled.

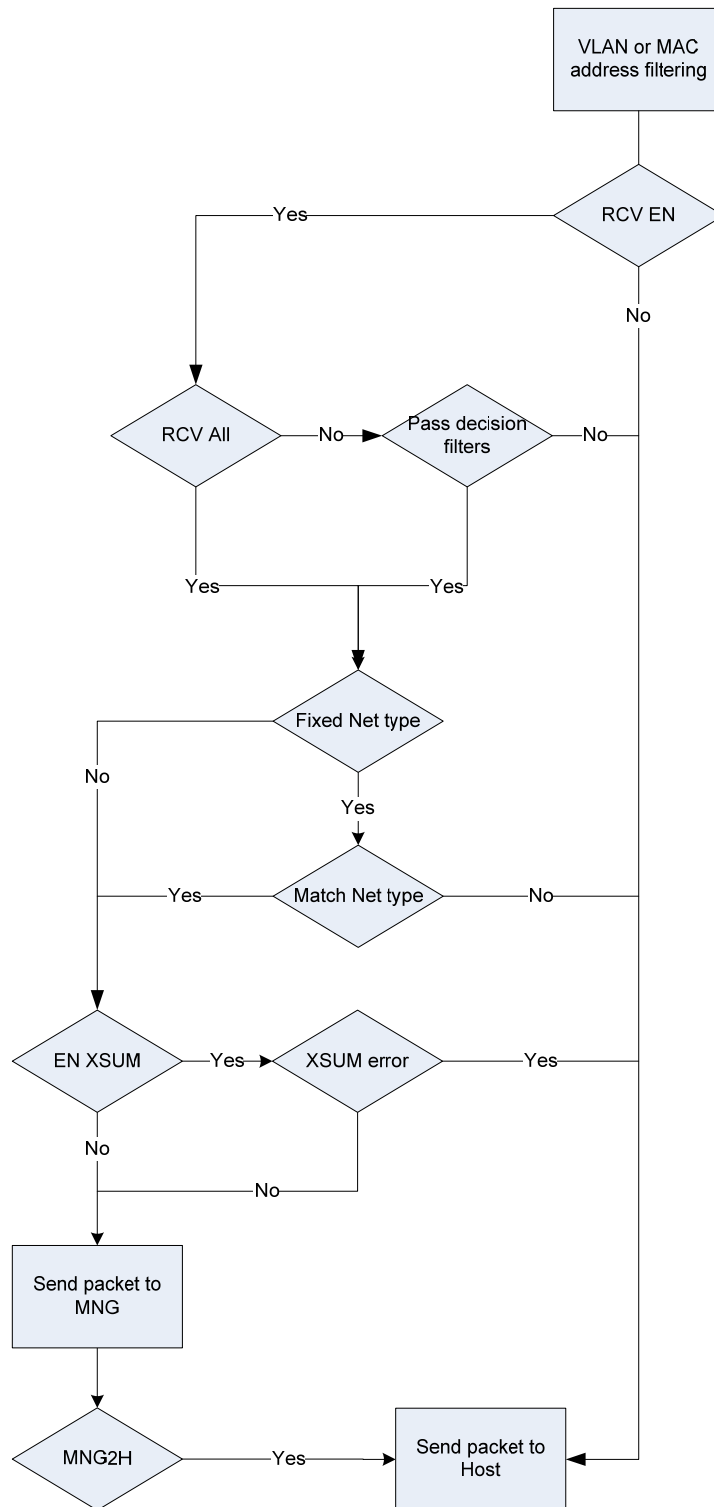


Figure 3-7. Receive Filtering

If Receive All has been enabled, all traffic passing the L2 filtering is sent to the MC. This is usually not desirable; however, it is an option.

If Receive All has not been enabled, then the incoming packet is compared against the manageability filters. After passing these filters, additional filters such as Ether Type and XSUM are used.

If the incoming packet has successfully passed all filters, it is then sent to the MC. Then the Management To Host filter is examined and if the packet matches this filter, the packet is also sent to the host.

3.5.5 Filtering Configuration

This section provides a general overview of how the MC can configure manageability filtering.

There are two basic mechanisms. The first is used in old generations of Intel Ethernet Controllers; the second is used in new generations.

3.5.5.1 Old Mechanism

The mechanism provided a way to configure and enable all types of filters (such RMCP port, ARP Requests and dedicated MAC address).

The default EEPROM image had common defaults already configured, including RMCP ports (26Fh & 298h). The MC issues a Receive Enable command with bits to enable filtering; it also provided optional bits to enable automatic ARP responses and dedicated MAC addresses.

More advanced filtering capabilities, such as Flex and VLAN filters, were configured and enabled using the Updated Management Received Filter Parameters command.

Intel Ethernet Controllers that use this method are:

- Intel 82573 Ethernet Controller
- Intel ESB2 Ethernet Controller
- Intel 82571 Ethernet Controller

3.5.5.2 133B New Mechanism

With the addition of NC-SI capability to the Intel Ethernet controllers, there are now two Sideband interfaces. As a result, methods for configuring manageability filters changed.

The new organizational method revolves around Manageability Decision Filters (MDEFs; now eight). Each MDEF has bits that enable or disable specific filters (such as L2 address [MAC] filtering, VLAN filters, ARP requests etc).

Inputs to each 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 is a multicast packet.
- Packet passed ARP filtering (request or response).
- Packet passed neighbor solicitation filtering.
- Packet passed 0x298/0x26F port filter.
- Packet passed a valid flex Port filter.
- Packet passed a valid flex TCO filter.

Each MDEF consists of bits that enable and disable various filters. Some have the option to be OR'd into the filter, some are a logical AND, and others can be either.

Table 3-6. Manageability Decision Filter

Filter	AND/OR Input	Mask Bits in MDEF
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

For this discussion, see Table 3-6.

If the MC wished to configure filtering for:

- Dedicated MAC address

- Port 0x26F

- ARP Request

Then bits 0 (MAC Address), 7 (ARP Request) and 11 (Port 0x26F) would need to be set. These bits are set using the Update Manageability Filter Parameters command, with Parameter Number 61h. The command itself then takes two additional parameters, the decision filter number (0-7) and the value of the MDEF. Writing a non-zero value for MDEF enables that MDEF.

In the example above, the simpler filters of ARP Requests and Port 0x26F are enabled by setting the associated bit within the MDEF. Configuring and enabling the dedicated MAC address filters requires additional steps.

To enable a MAC filter, the Update Manageability Filter Parameters command is also used, this time with a parameter number of 0x66, followed by the MAC Filter number (refer to the product specific documentation to find out how many are supported), followed by the MAC

address.

Since there are multiple MAC addresses that can be configured for filtering, the MC must then indicate which MAC address filters it wishes to use. This is accomplished again by using the Update Manageability Filter Parameters command, with a parameter of 0x60 to update the Filters Valid settings.

The Filters Valid setting is a bitmask of various filters including MAC, VLAN, IPv4 and IPv6. Each of which has the possibility of multiple values. Refer to product specific Datasheets for details as different products have a different number of possible filters.

Intel Ethernet Controllers that utilize this newer configuration mechanism are:

- Intel® 82574 Ethernet Controller
- Intel® 82575 Ethernet Controller
- Intel® 82576 Ethernet Controller
- Intel® 82598 Ethernet Controller

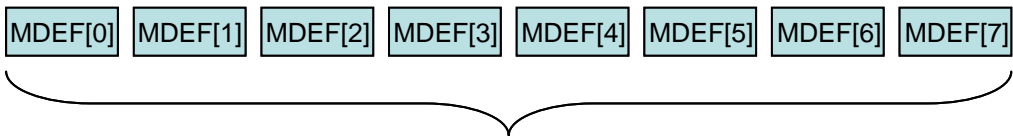
All new Intel Ethernet controllers will use this new configuration mechanism.

3.5.5.2.1 Manageability Decision Filters (MDEF)

This section provides more information about MDEFs.

There are eight separate MDEF filters, each with a number of possible filters to enable (see Table 3-7). Filters are configured independently using the Update Manageability Filters Parameters command, with parameter number 61h followed by the MDEF filter number to configure (0-7).

Table 3-7. Logical View of MDEF Filters



Filter		Manageability Decision Filter (MDEF)							
		0	1	2	3	4	5	6	7
L2 Unicast Address	AND								
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

After all filters have been configured, the eight MDEF filters are logically combined to form a matrix against which incoming packets are compared.

Figure 18 shows how MDEF filters are examined. An incoming packet is compared each filters until a match is found (a valid filter match) or the packet is dropped because it did not match any of the filters.

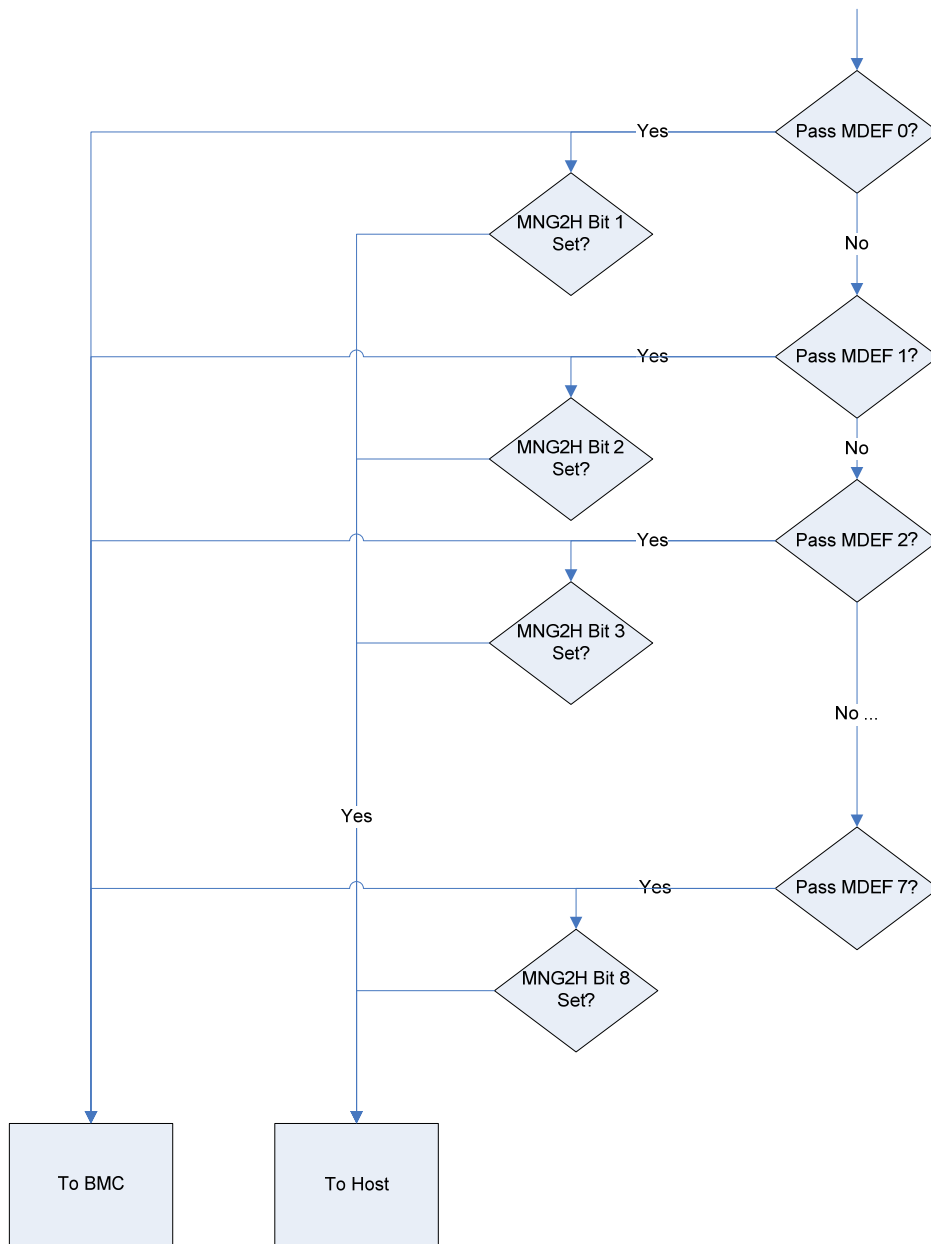


Figure 3-8 Logical flow of MDEF Filter Checking

Care must be taken when configuring and organizing the MDEF Filters. For example, if the MC desires a dedicated MAC and IP address, this should be configured using the same MDEF Filter.

See Table 3-8.

Table 3-8. MDEF Filters

Filter		Manageability Decision Filter (MDEF)							
		0	1	2	3	4	5	6	7
L2 Unicast Address	AND		x						
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND	x							
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

MDEF[0] is configured with an IP address filter; MDEF[1] is configured with a MAC address filter. IP and MAC addresses are configured in different locations.

With the above filter configuration, it would be possible for the MC to receive a packet with a correct MAC address, yet have an IP address that was not configured in the filter.

This is because the first filter compared will be the IP address filter from MDEF[0]; this would fail because the destination IP address of the incoming packet does not match what the MC configured.

The next filter checked is MDEF[1], where the destination MAC address of the packet matches that the MC configured, so the packet is sent to the MC.

The solution is to ensure that the IP and MAC address filters are in the same MDEF[].

See Table 3-9.

Table 3-9. MDEF Filters

Filter		Manageability Decision Filter (MDEF)							
		0	1	2	3	4	5	6	7
L2 Unicast Address	AND		x						
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND		x						
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

3.5.5.2.2 Management to Host

With the updated configuration mechanism, the Management to Host configuration no longer consists of specifying a specific filter (such as ARP Requests). Rather, the configuration indicates which (if any) of the MDEF filters should be used for Management to Host traffic.

As an example, consider a desired configuration where there is a Dedicated MAC address and the MC only wants to receive ARP Requests and RMCP Traffic (UDP Port 26Fh). In addition, the Management to Host configuration should ensure ARP Requests are sent to the host. See Table 3-10

Table 3-10. Example (Incorrect) MDEF Configuration

Filter		Manageability Decision Filter (MDEF)							
		0	1	2	3	4	5	6	7
L2 Unicast Address	AND	x							
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR	x							
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR		x						
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

MDEF[0] is configured so that a dedicated MAC is required, as is the ability to filter ARP Requests and RMCP traffic. In this configuration, the MC will only receive ARP requests and RMCP traffic as desired.

If Management to Host is left un-configured, the Host OS will not be able to receive ARP Requests. The logical thing to do is to configure the Management to host with a value of 1 (indicating MDEF[0] should be used).

Doing this however allows ALL traffic that passed MDEF[0] to also go to the host, meaning that RMCP traffic destined for the MC would also be sent to the host – not the desired filtering configuration.

Table 3-11 shows a slightly different configuration.

Table 3-11. Example (Correct) MDEF Configuration

Filter		Manageability Decision Filter (MDEF)							
		0	1	2	3	4	5	6	7
L2 Unicast Address	AND	x							
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR		x						
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR	x							
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

MDEF[0] still has the dedicated MAC address and RMCP Port 26Fh filtering. The ARP Request filtering has been moved to MDEF[1]. It did not have to be MDEF[1]; it could have been any of the other MDEF filters except MDEF[0]).

With the filters configured in this way, if the Management to Host filter is configured with a 02 (to indicated MDEF[1]), then all traffic that passes MDEF[1] is sent to both the MC and the Host.

This now provides the desired filtering. The MC receives only ARP Requests and RMCP Traffic with a specific Destination MAC address, while the host also receives ARP Requests.

3.5.5.2.3 168BMDEF Filters and Interaction with the Receive Enable Command

The Advanced version of the Receive Enable command allows the MC to specify a MAC and IP address for Automatic ARP Response. Refer to product-documentation to ensure support for this feature.

When the Advanced version of the Receive Enable command is used and the Dedicated MAC address bit is set within this command, Manageability firmware within the Intel Ethernet Controller will automatically configure MDEF[7] with dedicated MAC address support.

See Table 3-12.

Table 3-12. MDEF[7] Automatically Configured

Filter		Manageability Decision Filter (MDEF)							
		0	1	2	3	4	5	6	7
L2 Unicast Address	AND								x
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR		x						
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR	x							
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

Revisiting the previous example, if the MC uses the Advanced version of the Receive Enable command to configure the MAC Address (highly recommended, as it saves the MC several steps), then the MC can configure MDEF[0] for RMCP traffic, MDEF[1] for ARP Requests and Management to Host with a value of 02 (for MDEF[1]).

MDEF[7] is automatically configured for the MC by the NC.

3.6 Configuration Examples

This section provides an overview and sample configuration settings for commonly used filtering configurations. Three examples are presented, using the older mechanism and the updated method for configuration.

The examples are in pseudo code format, with the name of the SMBus command followed by parameters and an explanation. Here is a sample:

Receive Enable[00]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering.

These examples will be general, refer to product specific documentation for more detail.

3.6.1 Example 1 – Shared MAC, RMCP Only Ports

This example is the most basic configuration. The MAC address filtering is shared with the host operating system and only traffic directed the RMCP ports (26Fh & 298h) are filtered. For this example, the MC must issue gratuitous ARPs because no filter is enabled to pass ARP requests to the MC.

3.6.1.1 Example 1 - Old Method

Step 1: - Disable existing filtering

Receive Enable[00]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering:

Receive Enable settings 00h:

Bit 0 [0] – Disable Receiving of packets

Step 2: - Configure RMCP Port filtering

Update Manageability Filter Parameters [01, 00000300]

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register:

MANC value of 00000300h:

Bit 8 [1] – port 26Fh

Bit 9 [1] – port 298h

Step 3: - Enable Filtering

Receive Enable[05]

Using the simple form of the Receive Enable command:

Receive Enable settings 05h:

Bit 0 [1] – Enable Receiving of packets

Bit 2 [1] – Enable status reporting (such as link lost)

Bit 5:4 [00] – Notification method = SMB Alert

Bit 7 [0] – Use shared MAC

3.6.1.2 Example 1 – New Method

Step 1: - Disable existing filtering

Receive Enable[40]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering:

Receive Enable Control 40h:

Bit 0 [0] – Disable Receiving of packets

Bit 6 [1] - Reserved, must be set to 1

Step 2: - Configure MDEF[0]

Update Manageability Filter Parameters [61, 0, 00000C00]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update MDEF[0], as indicated by the 2nd parameter (0).

MDEF[0] value of 00000C00h:

Bit 10 [1] – port 298h

Bit 11 [1] – port 26Fh

Step 3: - Enable Filtering

Receive Enable[45]

Using the simple form of the Receive Enable command:

Receive Enable Control 45h:

Bit 0 [1] – Enable Receiving of packets

Bit 2 [1] – Enable status reporting (such as link lost)

Bit 5:4 [00] – Notification method = SMB Alert

Bit 6 [1] – Reserved, must be set to 1

Bit 7 [0] – Use shared MAC

Table 3-13. Example 1 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		0	1	2	3	4	5	6	7
L2 Unicast Address	AND								
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR	x							
Port 0x26F	OR	x							
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

3.6.2 Example 2 – Dedicated MAC, Auto ARP Response and RMCP Port Filtering

This example shows a common configuration; the MC has a dedicated MAC and IP address. Automatic ARP responses will be enabled as well as RMCP port filtering. By enabling Automatic ARP responses the MC is not required to send the gratuitous ARPs as it did in Example 1. Since ARP requests are now filtered, in order for the host to receive the ARP requests, the Manageability to Host filter will be configured to send the ARP requests to the host as well.

For demonstration purposes, the dedicated MAC address will be calculated by reading the System MAC address and adding 1 to it, assume the System MAC is AABBCDDED. The IP address for this example are 1.2.3.4.

Additionally, the XSUM filtering is enabled.

Note that not all Intel Ethernet Controllers support automatic ARP responses. Refer to product specific documentation.

3.6.2.1 Example 2 - Old Method

Step 1: - Disable existing filtering

Receive Enable[00]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering:

Receive Enable Control 00h:

Bit 0 [0] – Disable Receiving of packets

Step 2: - Read System MAC Address

Get System MAC Address []

Reads the System MAC address. Assume returned AABBCDDED for this example.

Step 3: - Configure RMCP Port filtering

Update Manageability Filter Parameters [01, 00A00300]

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register.

MANC Register 00A00300:

Bit 8 [1] – port 26Fh

Bit 9 [1] – port 298h

Bit 21 [1] – Management to Host Filter enable

Bit 23 [1] – XSUM Filter enable

Step 4: - Configure the Management to Host Filter

Update Manageability Filter Parameters [0A, 00000080]

Use the Update Manageability Filter Parameters command to update the Management Control to Host (MANC2H) Register.

MANC2H Register 00000080:

Bit 7 [1] – ARP Requests

This allows ARP requests to be passed to both manageability and to the host.

Step 5: - Enable Filtering

Receive Enable [8D, AABBCDDED, 01020304, 00, 00, 00]

Using the advanced version Receive Enable command, the first parameter:

Receive Enable Control 8Dh:

- Bit 0 [1] – Enable Receiving of packets
- Bit 2 [1] – Enable status reporting (such as link lost)
- Bit 3 [1] – Enable Automatic ARP Responses
- Bit 5:4 [00] – Notification method = SMB Alert
- Bit 7 [1] – Use Dedicated MAC

Second parameter is the MAC address (AABBCCDDED).

Third Parameter is the IP address(01020304).

The last three parameters are zero when the [notification method](#) is SMB Alert.

3.6.2.2 Example 2 – New Method

Step 1: - Disable existing filtering

Receive Enable[40]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering:

Receive Enable Control 40h:

- Bit 0 [0] – Disable Receiving of packets
- Bit 6 [1] - Reserved, must be set to 1

Step 2: - Read System MAC Address

Get System MAC Address []

Reads the System MAC address. Assume returned AABBCCDDED for this example.

Step 3: - Enable MNG2Host & XSUM Filters

Update Manageability Filter Parameters [01, 00A00000]

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register.

MANC Register 00A00000h:

- Bit 21 [1] - MNG2Host Filter Enable
- Bit 23 [1] – XSUM Filter enable

Note that some of the following configuration steps manipulate the MANC register indirectly; this command sets all bits except XSUM to 0. It is important to either do this step before the others or to read the value of the MANC and then write it back with only bit 32 changed. Also note that the XSUM enable bit may differ between Ethernet Controllers. Refer to product specific documentation.

Step 4: - Configure MDEF[0]

Update Manageability Filter Parameters [61, 0, 00000C00]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This updates MDEF[0], as indicated by the second parameter (0).

MDEF value of 00000C00h:

Bit 10 [1] – port 298h

Bit 11 [1] – port 26Fh

Step 5: - Configure MDEF[1]

Update Manageability Filter Parameters [61, 1, 00000080]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This updates MDEF[1], as indicated by the second parameter (1).

MDEF value of 00000080:

Bit 7 [7] – ARP Requests

When Enabling Automatic ARP responses, the ARP requests still goes into the manageability filtering system and as such needs to be designated as also needing to be sent to the host. For this reason, a separate MDEF is created with only ARP request filtering enabled.

Refer to the next step for more details.

Step 6: - Configure the Management to Host Filter

Update Manageability Filter Parameters [0A, 00000002]

Use the Update Manageability Filter Parameters command to update the Management Control to Host (MANC2H) Register.

MANC2H Register 00000002:

Bit 2 [1] – Enable MDEF[1] traffic to go to Host as well

This allows ARP requests to be passed to both manageability and to the host.

Specify a separate MDEF filter for ARP requests.

If ARP requests had been added to MDEF[0] and then MDEF[0] specified in Management to Host configuration, then not only would ARP requests be sent to the MC and host, RMCP traffic (ports 26Fh and 298h) would also be sent both places.

The MANC2H Filter is configured in this step and enabled in step 3.

Step 7: - Enable Filtering

Receive Enable [CD, AABCCDDEE, 01020304, 00, 00, 00]

Using the advanced version Receive Enable command, the first parameter:

Receive Enable Control CDh:

Bit 0 [1] – Enable Receiving of packets

Bit 2 [1] – Enable status reporting (such as link lost)

Bit 3 [1] – Enable Automatic ARP Responses

Bit 5:4 [00] – Notification method = SMB Alert

Bit 6 [1] - Reserved, must be set to 1

Bit 7 [1] – Use Dedicated MAC

Second parameter is the MAC address (AABCCDDEE).

Third Parameter is the IP address (01020304).

The last three parameters are zero when the notification method is SMB Alert.

Table 3-14. Example 2 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		0	1	2	3	4	5	6	7
L2 Unicast Address	AND								x
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR		x						
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR	x							
Port 0x26F	OR	x							
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

3.6.3 Example 3 – Dedicated MAC & IP Address

This example provides the MC with a dedicated MAC and IP address and allows it to receive ARP requests. The MC is then responsible for responding to ARP requests.

For demonstration purposes, the dedicated MAC address are calculated by reading the System MAC address and adding 1 to it. Assume the System MAC is AABBCDDDED. The IP address for this example is 1.2.3.4. The Receive Enable command is used to configure the MAC address filter.

In order for the MC to be able to receive ARP Requests, it needs to specify a filter and that filter needs to be included in the Manageability To Host filtering so that the host OS may also receive ARP Requests.

Note that the Legacy Method does not support IP address filtering; the MC must provide that filtering.

3.6.3.1 Example 3 - Old Method

Note that in the older Intel Ethernet Controllers, there is no configurable IP address filter. If the MC wishes to support more than one IP address, it must respond to ARP requests and filter incoming packets accordingly.

Step 1: - Disable existing filtering

Receive Enable[00]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering:

Bit 0 [0] – Disable Receiving of packets

Step 2: - Enable ARP Request and Mng2Host Filters

Update Manageability Filter Parameters [01, 00202000]

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register.

MANC Register 00202000:

Bit 13 [1] - ARP Request Filter

Bit 21 [1] - Enable Management To Host Filter

Step 3: - Configure the Management to Host Filter

Update Manageability Filter Parameters [0A, 00000080]

Use the Update Manageability Filter Parameters command to update the Management Control to Host (MANC2H) Register.

MANC2H Register 00000080:

Bit 7 [1] – ARP Requests

This allows ARP requests to be passed to both manageability and to the host.

Step 4: - Read System MAC Address

Get System MAC Address []

Reads the System MAC address. Assume returned AABCCDDED for this example.

Step 5: - Enable Filtering

Receive Enable [85, AABCCDDEE, 00000000, 00, 00, 00]

Using the advanced version Receive Enable command, the first parameter:

Receive Control byte 85h:

Bit 0 [1] – Enable Receiving of packets

Bit 2 [1] – Enable status reporting (such as link lost)

Bit 5:4 [00] – Notification method = SMB Alert

Bit 7 [1] – Use Dedicated MAC

Second parameter is the MAC address: AABCCDDEE.

Third Parameter is the IP address and is not used when not enabling Auto ARP response.

The last three parameters are zero when the notification method is SMB Alert.

3.6.3.2 Example 3 – New Method

Step 1: - Disable existing filtering

Receive Enable[40]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering:

Receive Enable Control 40h:

- Bit 0 [0] – Disable Receiving of packets
- Bit 6 [1] – Reserved, must be set to 1

Step 2: - Read System MAC Address

Get System MAC Address []

Reads the System MAC address. Assume returned AABCCDDED for this example.

Step 3: - Configure IP Address Filter

Update Manageability Filter Parameters [64, 00, 01020304]

Use the Update Manageability Filter Parameters to configure an IPv4 filter.

The first parameter (64h) specifies that we are configuring an IPv4 filter.

The second parameter (00h) indicates which IPv4 filter is being configured, in this case filter 0.

The third parameter is the IP address – 1.2.3.4.

Step 4: - Configure MAC Address Filter

Update Manageability Filter Parameters [66, 00, AABCCDDEE]

Use the Update Manageability Filter Parameters to configure a MAC Address filter.

The first parameter (66h) specifies that we are configuring a MAC Address filter.

The second parameter (00h) indicates which MAC Address filter is being configured, in this case filter 0.

The third parameter is the MAC Address - AABCCDDEE

Step 5: - Configure MDEF[0] for IP and MAC Filtering

Update Manageability Filter Parameters [61, 0, 00000009]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update MDEF[0], as indicated by the second parameter (0).

MDEF value of 00000009:

- Bit 1 [1] – MAC Address Filtering
- Bit 3 [1] – IP Address Filtering

Step 6: - Configure MDEF[1]

Update Manageability Filter Parameters [61, 1, 00000080]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update MDEF[1], as indicated by the second parameter (1).

MDEF value of 00000080:

Bit 7 [7] – ARP Requests

When filtering ARP requests the requests go into the manageability filtering system and as such need to be designated as also needing to be sent to the host. For this reason a separate MDEF is created with only ARP request filtering enabled.

Step 7: - Configure the Management to Host Filter

Update Manageability Filter Parameters [0A, 00000002]

Use the Update Manageability Filter Parameters command to update the Management Control to Host (MANC2H) Register.

MANC2H Register 00000002:

Bit 2 [1] – Enable MDEF[1] traffic to go to Host as well

Step 8: - Enable MNG2Host Filter

Update Manageability Filter Parameters [01, 00200000]

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register.

MANC Register 00200000h:

Bit 21 [1] - MNG2Host Filter Enable

This enables the MANC2H filter configured in step 7.

Step 9: - Enable Filtering

Receive Enable [45]

Using the simple form of the Receive Enable command,:

Receive Enable Control 45h:

Bit 0 [1] – Enable Receiving of packets

Bit 2 [1] – Enable status reporting (such as link lost)

Bit 5:4 [00] – Notification method = SMB Alert

Bit 6 [1] - Reserved, must be set to 1

The Resulting MDEF filters are as follows:

Table 3-15. Example 3 MDEF Results

Filter		Manageability Decision Filter (MDEF)							
		0	1	2	3	4	5	6	7
L2 Unicast Address	AND	x							
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND	x							
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR		x						
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

3.6.4 Example 4 – Dedicated MAC and VLAN Tag

This example shows an alternate configuration; the MC has a dedicated MAC and IP address, along with a VLAN tag of 32h will be required for traffic to be sent to the MC. This means that all traffic with VLAN a matching tag will be sent to the MC.

For demonstration purposes, the dedicated MAC address will be calculated by reading the System MAC address and adding 1 to it, assume the System MAC is AABBCDDDED. The IP address for this example will be 1.2.3.4 and the VLAN tag will be 0032h.

It is assumed the host will not be using the same VLAN tag as the MC. If they were to share the same VLAN tag then additional filtering would need to be configured to allow VLAN tagged non-unicast (such as ARP requests) to be sent to the host as well as the MC using the [Manageability to Host filter](#) capability.

Additionally, the [XSUM filtering](#) will be enabled.

3.6.4.1 Example 4 - Old Method

Step 1: - Disable existing filtering

```
Receive Enable[00]
```

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering:

Bit 0 [0] – Disable Receiving of packets

Step 2: - Configure XSUM Filter

```
Update Manageability Filter Parameters [01, 00800000]
```

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register.

MANC Register 00800000:

Bit 23 [1] – XSUM Filter enable

For this example no other settings within the MANC are required as dedicated MAC and VLAN filtering are configured with different commands.

Step 3: - Read System MAC Address

Get System MAC Address []

Reads the System MAC address. Assume returned AABBCCDDED for this example.

Step 4: - Configure the VLAN Filter 0

Update Manageability Filter Parameters [06, 8, 0, 0032]

Use the Update Manageability Filter Parameters command to configure the Management VLAN Filter 0 (parameter 06h).

The second, third and fourth parameters enable the VLAN filter and set the VLAN ID to 0032h.

Step 5: - Enable Filtering

Receive Enable [85, AABBCCDDEE, 01020304, 00, 00, 00]

Using the advanced version Receive Enable command, the first parameter:

Receive Control byte 85h:

Bit 0 [1] – Enable Receiving of packets

Bit 2 [1] – Enable status reporting (such as link lost)

Bit 5:4 [00] – Notification method = SMB Alert

Bit 7 [1] – Use Dedicated MAC

Second parameter is the MAC address: AABBCCDDEE.

Third Parameter is the IP address: 01020304.

The last three parameters are zero when the [notification method](#) is SMB Alert.

3.6.4.2 Example 4 – New Method

Step 1: - Disable existing filtering

Receive Enable[40]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the MC by disabling filtering:

Receive Enable Control 40h:

Bit 0 [0] – Disable Receiving of packets

Bit 6 [1] – Reserved, must be set to 1

Step 2: - Read System MAC Address

Get System MAC Address []

Reads the System MAC address. Assume returned AABBCCDDED for this example.

Step 3: - Configure XSUM Filter

Update Manageability Filter Parameters [01, 00800000]

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register.

MANC Register 00800000h:

Bit 23 [1] – XSUM Filter enable

Note that some of the following configuration steps manipulate the MANC register indirectly, this command sets all bits except XSUM to 0. It is important to either do this step before the others, or to read the value of the MANC and then write it back with only bit 32 changed. Also note that the XSUM enable bit may differ between Ethernet Controllers, refer to product specific documentation.

Step 4: - Configure VLAN 0 Filter

Update Manageability Filter Parameters [62, 0, 0032]

Use the Update Manageability Filter Parameters command to configure VLAN filters. Parameter 62h indicates update to VLAN Filter, the second parameter indicates which VLAN filter (0 in this case), the last parameter is the VLAN ID (0032h).

Step 5: - Configure MDEF[0]

Update Manageability Filter Parameters [61, 0, 00000040]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update MDEF[0], as indicated by the second parameter (0).

MDEF value of 00000040:

Bit 2 [1] – VLAN AND

Step 6: - Enable Filtering

Receive Enable [A5, AABBCDDEE, 01020304, 00, 00, 00]

Using the advanced version Receive Enable command, the first parameter:

Receive Enable Control A5h:

- Bit 0 [1] – Enable Receiving of packets
- Bit 2 [1] – Enable status reporting (such as link lost)
- Bit 5:4 [00] – Notification method = SMB Alert
- Bit 6 [1] – Reserved, must be set to 1
- Bit 7 [1] – Use Dedicated MAC

Second parameter is the MAC address: AABBCDDEE.

Third Parameter is the IP address: 01020304.

The last three parameters are zero when the [notification method](#) is SMB Alert.

Table 3-16. Example 4 MDEF Results

Filter		Manageability Decision Filter (MDEF)							
		0	1	2	3	4	5	6	7
L2 Unicast Address	AND								x
Broadcast	AND								
Manageability VLAN	AND	x							
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

3.7 Sending and Receiving SMBus Packets

The SMBus specification specifies a maximum packet size of 32 bytes. Most Ethernet packets are significantly larger than this limitation.

For packets that are larger than the 32 byte packet limit size, the packets are sent in fragments, 32 bytes at a time. Large packets will be broken in to many packets.

The Intel NC will automatically break up large incoming packets into fragments and send them to the MC. Likewise, the MC must transmit large packets in multiple fragments to the NC, which in turn will combine the fragments into a single packet.

Some Intel Ethernet Controllers provide the ability to increase the maximum size of the SMBus packet up to 240 bytes. This is a configuration option within the EEPROM image and cannot be modified by the MC.

Refer to product specific documentation for more information on increasing the SMBus packet fragment size and for the packet transmit and receive SMBus commands.

3.7.1 Receive TCO Packet Command

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

A packet is delivered in more than one SMBus fragment (a minimum of two: one for the packet and the other for status). Also, MC should follow the *First* and *Last* bit.

The opcode can have these values:

90h - First Fragment

10h - Middle Fragment.

When the opcode is 50h, this indicates the last fragment of the packet, which contains packet status.

Following is the Receive TCO Packet format and the data format returned from the NC.

Table 3-17. Receive TCO Packet Format

Function	Command
Receive TCO Packet	C0h or D0h

Function	Byte Count	Data 1 (Op-Code)	Data 2	...	Data N
Receive TCO First Fragment	N	90	Packet Data Byte	...	Packet Data Byte
Receive TCO Middle Fragment	N	10	Packet Data Byte	...	Packet Data Byte
Receive TCO Last Fragment		50			

Most received packets will be longer than a single SMBus fragment, as such the MC must issue the Read TCO Packet command repeatedly until the OpCode (Data byte 1) is a 50h. The MC will then re-assemble the entire Ethernet Packet and process it.

The last fragment received (the one with an OpCode of 50h) does not contain any data belonging to the received Ethernet Packet, it contains status information regarding the packet just read by the MC. The format of this status information varies from Ethernet Controller to Ethernet Controller, refer to product specific documentation for details.

3.7.2 Transmit TCO Packet Command

When the MC wishes to transmit an Ethernet packet, it does so using the Transmit TCO command. There are different commands depending upon what part of the packet is being sent to the NC.

Most packets will be larger than the SMBus fragment size, as such it must be transmitted to the NC one fragment at a time. There can be only one Transmit First Fragment' and

'Transmit Last Fragment' command issues, with multiple 'Transmit Middle Fragment' commands issues.

Upon completion the NC will assemble the packet and transmit it.

The Transmit Packet command format is as follows:

Table 3-18. Transmit Packet Command Format

Function	Command	Byte Count	Data 1	...	Data N
Transmit First Fragment	84h	N	Packet Data MSB	...	Packet Data LSB
Transmit Middle Fragment	04h	N	Packet Data MSB	...	Packet Data LSB
Transmit Last Fragment	44h	N	Packet Data MSB	...	Packet Data LSB

If the overall packet length is greater than 1536 bytes, the packet is silently discarded by the NC.

4 NC-SI Interface

This section discusses the industry standard DMTF defined NC-SI interface available in the current generation of Intel Ethernet Controllers for Servers.

4.1 Overview

While the SMBus interface for sideband communication between a MC and the NC controller is simple, the protocol layer is made up of proprietary commands. This places additional work on the MC firmware writers wishing to support more than one vendor's sideband interface. In addition, the SMBus interface is not very fast, making it unsuitable for high bandwidth uses such as Remote Media (R-Media) and Remote Keyboard-Video-Mouse (R-KVM) functionality.

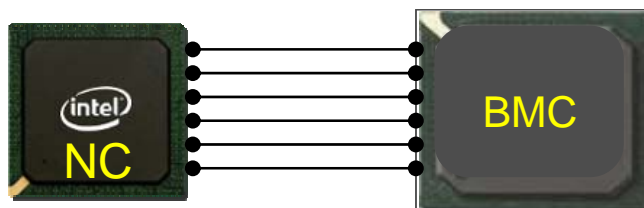


Figure 4-1. NC-SI Interface

Recognizing the need for increased speed and an industry standard, the DMTF created the Network Controller Sideband Interface (NC-SI) specification. This specification details the electrical requirements for the physical connection between the MC and the Network Controller as well as the protocol requirements.

Intel was an active participant in the DMTF workgroup that defined the NC-SI specification.

4.1.1 Terminology

The NC-SI specification introduces several new terms that have not previously been used in regards to the Intel Sideband interface.

4.1.1.1 Package

A package is another term for an Network Controller (NC). There can be up to four distinct packages connected to a MC on a shared NC-SI connection.

4.1.1.2 Channel

A channel is another name for a port on a package (Network Controller). Each package can have 31 different channels (ports) that support the NC-SI interface.

4.1.1.3 AEN

This is an acronym for Asynchronous Event Notification. An AEN is a message that the NC

can send (if configured by the MC to do so) to the MC to notify it of specific events, such as a link lost event for example.

4.1.2 Package Options

A package (NC) can have up to 31 channels (ports) and there can be up to 4 packages. This provides a larger number of possible combinations on physical connectivity. Examples of which are show in three figures below.

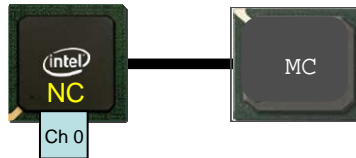


Figure 4-2. Single Package, Single Channel

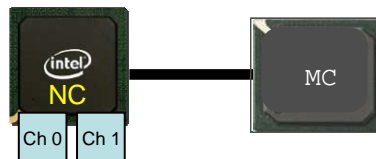


Figure 4-3. Single Package, Dual Channel

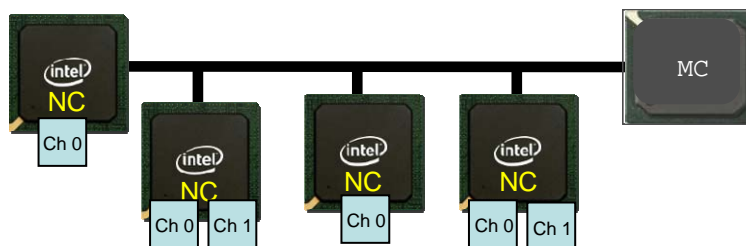


Figure 4-4. Four Separate Packages, varied Channels

Internal to each package there is also a number of different options, such as dedicated verses shared buffers for example. Refer to the NC-SI specification for more information on these possibilities. For the purposes of how an NC configures filters and is physically connected to an NC, the internal implementation of a NC makes little or no difference.

4.2 Physical Layer

This section discusses the physical connection between the MC and the NC for the NC-SI interface.

When the DMTF NS-SI workgroup began the process of defining an industry standard sideband interface, it was decided to start with the existing Reduced Media Independent Interface (RMII) standard interface. The RMII specification can be downloaded from http://www.national.com/appinfo/networks/files/rmii_1_2.pdf.

RMII is a point to point standard for passing entire Ethernet Frames – having NC-SI based upon this provides a considerable improvement over the SMBus interface where packets are broken up into fragments and passed to the MC.

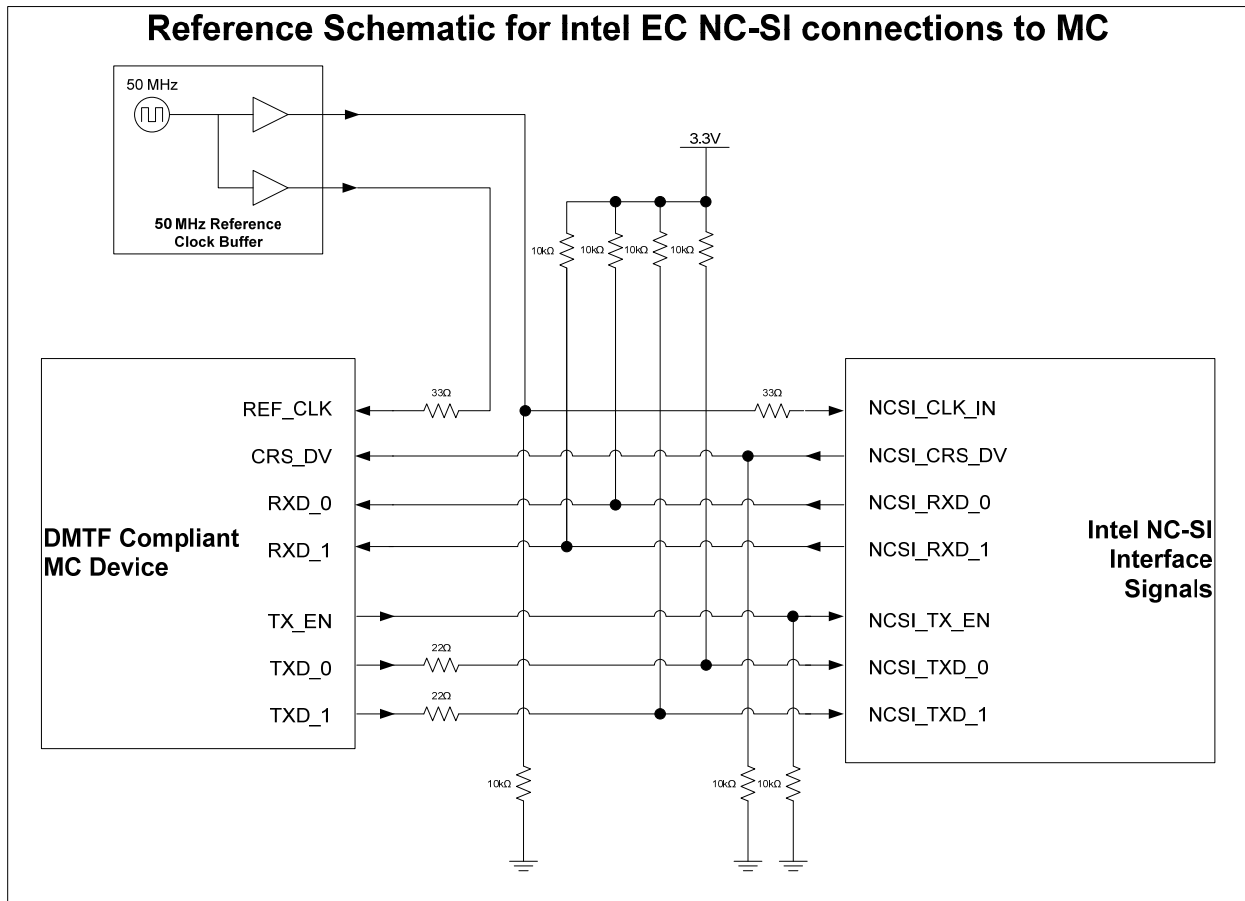


Figure 4-5. Sample NC-SI Reference Schematic

There are some electrical differences between NC-SI and RMII which are detailed in the NC-SI specification that should be noted.

Table 4-1. NC-SI Pins

Pin Name	Description	I/O to NC	I/O to MC	Mandatory or Optional
REF_CLK	Clock reference for receive, transmit, and control interface	Input	Input	Mandatory
CRS_DV	Carrier Sense/Receive Data Valid	Output	Input	Mandatory
RXD[1:0]	Receive data	Output	Input	Mandatory
TX_EN	Transmit enable	Input	Output	Mandatory
TXD[1:0]	Transmit data	Input	Output	Mandatory
RX_ER	Receive error	Output	Input	Optional
ARB_IN	Network Controller hardware arbitration Input	Input	N/A	Optional
ARB_OUT	Network Controller hardware arbitration Output	Output	N/A	Optional

The NC-Si interface includes includes six mandatory pins and three optional pins. Table 11 details the NC-SI pins and Figure 4-5 shows a sample schematic.

4.2.1 REF_CLK Source

On Intel® Ethernet Controllers supporting NC-SI, the reference clock can come from either an external source, or from the Ethernet Controller itself using the NCSI_CLK_OUT pin.

The configuration of whether the reference clock comes from an external source or from the Intel Ethernet Controller is determined by a setting within the EEPROM image of the Intel NC.

4.2.2 Multi-drop Arbitration

When there is more than one NC on the NC-SI connection, all NC's share the transmit, receive and clock pins. When the MC transmits data, all NC's (and therefore channels) receive this data, the data is examined and if the source MAC address of the packet matches the MAC address a channel was configured for, it transmits the packet. If it does not match, the packet is simply ignored.

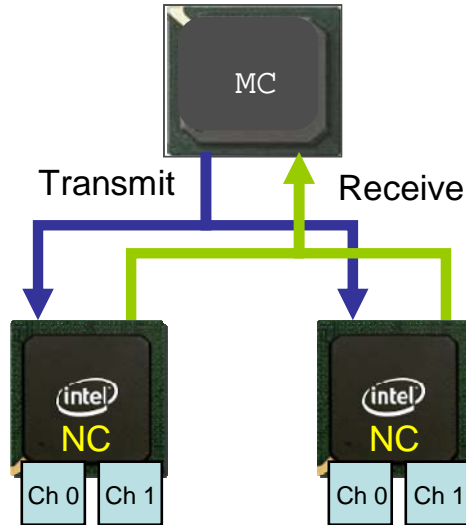


Figure 4-6. Multi-drop Arbitration

This situation is for when the MC transmits, where there is a one to many relationship (1 MC to many NC). When a NC needs to transmit data to the MC there must be some sort of arbitration mechanism so that only one NC transmits data to the MC at a time, otherwise it would be possible for packet corruption.

There are two mechanisms for this, Software Arbitration and Hardware Arbitration.

4.2.2.1 Command-based Arbitration (Software Arbitration)

Software arbitration relies on the MC to actively select an NC so that it may transmit data (if available) to the MC. The MC does this by issuing the Select Package command to select a Package and the Deselect Package command to deselect it.

4.2.2.2 Hardware Arbitration

With software arbitration, the MC selects an NC so that it has the opportunity to transmit data from to the MC. This places additional responsibilities on the MC, it must implement some sort of round robin type of scheduler to accomplish this task.

Another option that requires no actions on the part of the NC is to use hardware arbitration, specifically the optional arbitration pins ARB_IN and ARB_OUT. By connecting these pins (from NC to NC) the individual NC's decide amongst themselves which one can transmit data to the MC.

Refer to products specific documentation, as not all Intel Ethernet controllers supporting NC-SI support hardware arbitration.

4.3 Protocol Layer

This section discusses the protocol layer of the NC-SI interface.

4.3.1 Overview

In order to utilize the NC-SI interface for sending and receive Ethernet packets, the MC must configure the NC-SI interface. This is accomplished by the MC sending Control Packets to the NC. The NC can also send notification of events to the MC using AEN packets.

The MC must issue Control Packets to configure and enable a specific channel (port) on a package (NC). Configuration can be as simple as specifying a MAC address to filter on to much more robust and sophisticated configuration that includes link speed negotiation, VLAN configuration, traffic filters etc.

4.3.2 Traffic Types

The NC-SI interface is used to transmit full Ethernet Frames between the NC and the MC, this includes not only the manageability traffic, but the Control and AEN packets as well, they are fully formed Ethernet packets.

Control and AEN packets are uniquely identifiable because they have an EtherType of 88F8h that is a registered IANA number specifically for NC-SI usage.

4.3.2.1 Control Packets

Control packets are sent from the MC to the NC. Control packets include the ability to configure and enable filtering, set link negotiation policy, read statistics etc.

Refer to the DMTF NC-SI specification for details.

4.3.2.2 AEN Packets

Asynchronous Event Notification packets unsolicited notifications from the Ethernet Controller to the Management Controller. Some possible reasons for an AEN could be link status changes, and the OS driver being loaded or unloaded.

The MC configures which, if any, types of AEN it wishes to receive. Refer to the DMTF NC-SI specification for details.

4.3.3 Addressing

There needs to be a mechanism to direct configuration commands (Control Packets) to a specific port on a specific NC. The DMTF defined a mechanism by which each package (NC) within a system must have a unique 3 bit number associated with it. Additionally, each channel (port) on a given package must have a 5 bit ID associated with it.

Table 4-2. Channel ID

Bits	Field
[7..5]	Package ID
[4..0]	Internal Channel ID

In this way, a single 8 bit value can be used to uniquely identify a specific channel in a system, even if there are multiple packages. If a Control packet is sent with a Channel ID of 1Fh, the command applies to the entire package.

The Package ID is configurable within the EEPROM image on Intel Ethernet Controllers. The

Internal Channel ID is not configurable and is associated with the port number on the NC, such that port 0 has the Internal Channel ID of 0, port 1 has Internal Channel ID 1 and so on.

4.3.4 Transmit Flow Overview

When the MC transmits a packet, whether it is a [Control Packet](#) or a pass-through packet, all NC's receive that packet as they share the NC-SI transmit & receive lines. There must be a mechanism for the different NC's to determine if the packet is for it, or not.

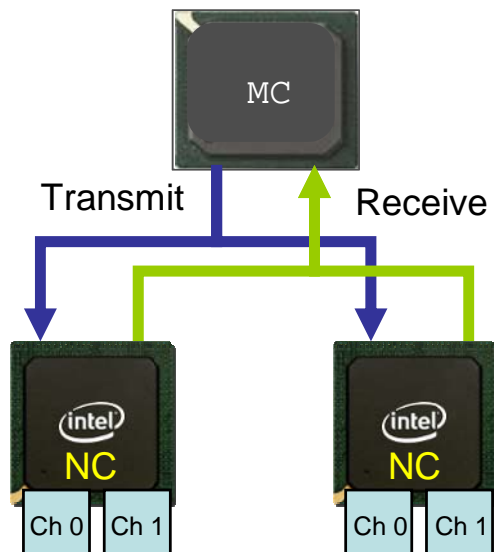


Figure 4-7. NC-SI

When the MC is configuring an NC, it sends [Control Packets](#), which are valid Ethernet packets with an EtherType of 88F8h. Part of the header in the control packet identifies which Package and Channel the Control Packet is targeting. If the Package ID within the Control Packet does not match the Package ID of the NC receiving the packet, it is ignored.

Manageability packets are 'normal' Ethernet packets that can be anything from RMCP to HTTP packets. If there are say two NC's, each with two ports on them, the mechanism for determining which port should transmit the packet is based upon the source MAC address of the packet being transmitted.

During configuration, the MC configures each channel on each package with a MAC address. Every NC on the NC-SI bus will receive the manageability packet being transmitted. When received by an NC, the packet is examined and the source MAC address compared to the one configured for the channels on that Package by the NC. If there is a match, then the packet is transmitted by the Channel with the matching MAC address.

4.3.5 Receive Flow Overview

When an Ethernet packet is received by a channel/port, the destination MAC address is compared against the manageability MAC address that the MC configured for that channel. If there is a match, then the packet is sent to the next level of filtering, if there is no match, it is sent to the host side filtering path.

4.4 Differences Between NC-SI and RMII

The NC-SI physical connection between the MC and the NC is very similar however not exactly the same as that defined in the RMII standard. One difference is that RMII is a point to point standard, while NC-SI added the ability to have up to 8 different physical NC's sharing the connection to the MC.

In general, the physical layer for NC-SI is very close to that of RMII. NC-SI extended the RMII physical requirements from a point to point standard to a point to multi-drop capable implementation. The NC-SI specification allows for a single MC to be connected to up to 4 physical Ethernet Controllers.

Refer to the NC-SI specification for a detailed list of differences.

In general, many customers have found that many MCs supporting the RMII electrical specification have sufficient tolerances as to also support NC-SI. Intel does not validate specific MCs for NC-SI electrical compatibility.

4.5 Basic NC-SI Work-flows

4.5.1 Package States

The information and recommendation in this section is only of significance when using software arbitration.

An NC Package may be in one of the following 2 states:

1. **Selected** – In this state the package is allowed to use the NC-SI lines, meaning the NC Package may send data to the MC
2. **Deselected** – In this state the package is not allowed to use the NC-SI lines, meaning, the NC Package may not send data to the MC.

Also note that the MC must "Select" no more than one NC Package at any given time.

Package Selection may be accomplished in one of 2 methods:

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

Package Deselect may be accomplished only by issuing the "Deselect Package" command.

Summary & recommendation: 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 please refer to the NC-SI specification..

4.5.2 Channel States

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

1. Initial State – In this state the channel will only accept “Clear Initial State” command (the package will also accept the “Select Package” & “Deselect Package” commands).
2. Active” state – This is the normal operational mode. All commands are accepted.

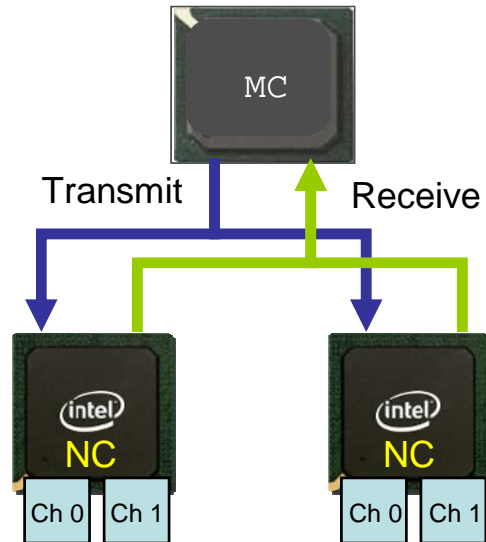


Figure 4-8. NC-SI Channels

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

4.5.3 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:

```
For package_id=0x0 to MAX_PACKAGE_ID
{
  Issue “Select Package” command to Package ID package_id
  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 received a response then
      {
        Consider internal_channel_id as a valid channel for the package_id
      }
    }
  }
}
Package
```

The MC may now optionally also discover channel capabilities & version Id

```

for the channel
    }
    Else (If not received a response to "Clear Initial State"), retry step for
three times
    }
    Issue a "Deselect Package" command to the package (and continue to the next
package)
    }
    Else (If not received a response to "Select Package"), retry step three times
}

```

4.5.4 Configurations

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

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

Thus, it is recommended that the MC re-configure everything on power-up and channel/package resets.

4.5.4.1 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.

Example: The MC should verify that the NC supports a specific AEN before enabling it.

4.5.4.2 Receive Filtering

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

4.5.4.2.1 MAC Address Filtering

NC-SI defines 3 types of MAC address filters: Unicast, Multicast & broadcast. To be received (i.e. not dropped) a packet must match at least one of these filters.

Recommendation: The MC should set 1 MAC address using the "Set MAC Address" command and Enable Broadcast & Global Multicast filtering.

Unicast/Exact match (Set MAC Address command)

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

Note: The NC may expose 3 types of Unicast/Exact match filters (i.e. MAC filters that match on the entire 48 bits of the MAC address): Unicast, Multicast & Mixed.

Please refer to NC-SI – "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 (i.e. ARP request, DHCP, NetBIOS).

2. Disabled – All Broadcast traffic is **forwarded** to the MC, with no exceptions. The recommended operational mode is “Enabled”, with specific filters set. Please refer to NC-SI “Enable/Disable Broadcast Filter” command.

Global Multicast (Enable/Disable Global Multicast Filter)

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

3. Enabled – All Multicast traffic is **blocked (not forwarded)** to the MC.
4. Disabled – All Multicast traffic is **forwarded** to the MC, with no exceptions.

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

Note that not all multicast filtering modes are necessarily supported.

Please refer to NC-SI “Enable/Disable Global Multicast Filter” command for further details.

4.5.4.3 VLAN

NC-SI defines the following VLAN work-modes:

Table 4-3. VLAN Work Modes

Mode	Command & 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.

Please refer to NC-SI – “Enable VLAN” command for further details on this command and products specific documentation for support of the VLAN modes on Intel Ethernet Controllers.

Recommendation:

1. Modes:
 - If VLAN is not required – use the “Disabled” mode.
 - 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.

4.5.5 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.

4.5.5.1 Channel Enable

This mode controls the state of the Receive path:

1. **Disabled:** The channel will not pass any traffic from the network to the MC.

2. **Enabled:** The channel will pass any traffic from the network (that matched the configured filters) to the MC.

Note:

This state also affects AENs: AENs will only be sent in “Enabled” state.
The default state is “Disabled”.

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

4.5.5.2 Network Transmit Enable

This mode controls the state of the Transmit path:

1. **Disabled:** The channel will not pass any traffic from the MC to the Network.
2. **Enabled:** The channel will pass 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. Thus, the MC should enable network transmit only after configuring the MAC address.

Recommendation:

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

4.5.6 Asynchronous Event Notifications

The Asynchronous Event Notifications are unsolicited messages sent from the NC to the MC to report status changes (e.g. Link change, OS state change).

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 (e.g. a frame with the NC-SI EtherType), will arrive out-of-context (e.g. 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.

4.5.7 Querying Active Parameters

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

4.5.8 Resets

In NC-SI there are 2 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 keeps 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 may not be reported and/or explicitly noticed, the MC should periodically poll the NC with NC-SI commands (i.e. "Get Version ID", "Get Parameters"...) 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.

The Intel recommended polling interval is 2-3 seconds.

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

4.6 Advanced Workflows

4.6.1 Multi-NC Arbitration

In a multi-NC environment, there is a need to arbitrate the NC-SI lines.

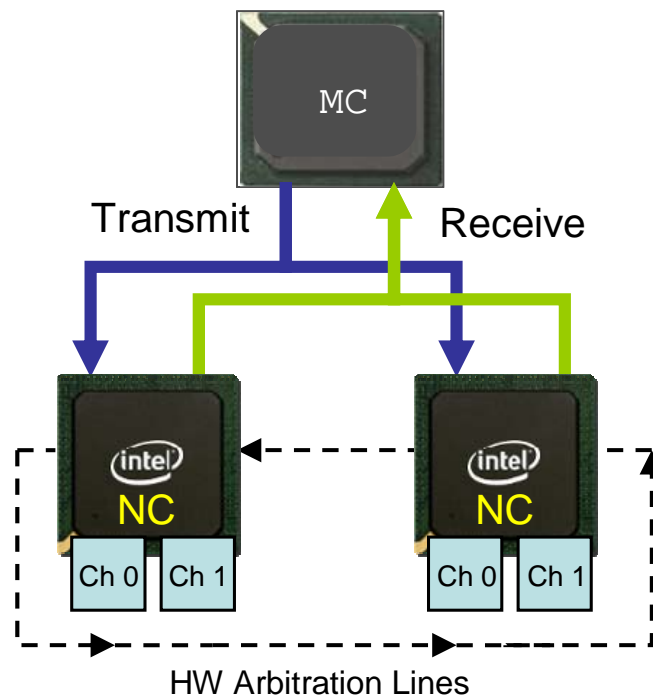


Figure 4-9. Multi-EC Environment

The NC-SI RX lines are shared between the Network Controllers. Thus, to allow sharing of the NC-SI RX lines NC-SI has defined an arbitration scheme.

The arbitration scheme mandates that only one NC Package may 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 "Deselected".

NC-SI has defined 2 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/Deselect Package"). The MC is responsible for having only one package Selected at any given time.
2. "HW Arbitration". In this mechanism 2 additional pins on each NC Package are used to synchronize the NC Package. Each NC Package has a ARB_IN & ARB_OUT line and these lines are used to transfer Tokens. An NC Package that has a TOKEN is considered Selected.

Refer to Section 4 of the NC-SI specification.

The Intel recommendation is to utilize HW arbitration if available, for a multiple NC environment.

4.6.1.1 Example Package Selection Sequence

The following details an example work-flow for a MC and occurs **after** the discovery, initialization & configuration. It is assumed that HW Arbitration is NOT supported/used for this example.

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

1. Define a time-slot for each device. See below for details.
2. Discover, initialize & configure all the NC Packages & Channels.
3. Issue a "Deselect 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:
 - Issue a "Deselect Package" to the active_package. The MC **must** then wait for a response and then an additional timeout for the Package to become Deselected
 - Find the next available package (typically active_package = active_package + 1.
 - Issue a "Select Package" to active_package

4.6.2 External Link Control

The MC may use the NC-SI "Set Link" command control the external interface link settings. This command allows 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**. Indication of the Host OS status can be obtained via "Get Link Status" and/or "Host OS Status Change" AEN.

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 Network Controller until proven otherwise").

It is recommended that the MC first queries 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 for details.

4.6.3 Multiple Channels (Fail-Over)

In order to support a fail-over scenario, it is required from the MC to operate 2 or more channels. These channels may or may not be in the same package.

The key element of a fault-tolerance fail-over scenario is having 2 (or more) channels identifying to the switch with the same MAC address, but only one of them being active at any given time (i.e. switching the MAC address between channels).

To accomplish this, NC-SI provides the following:

1. "Enable Network TX" command. This command allows 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

4.6.3.1 Example Fail-Over Algorithm

Following is a sample work-flow for a fail-over scenario, for an Intel dual-port adapter (1 Package, 2 channels)

1. MC initializes & 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 channel0's specific LAN port.
5. The MC begins normal work flow.
6. Should the MC receive an indication (AEN or polling) that the link status for the active channel (channel0) has changed, the MC should:
Disable channel0 for Network Transmission.
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 channel0's specific LAN port.
Resume normal work-flow

}

If not found: Report error and continue polling until a valid channel is found.

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

In addition, the MC might prefer to use a specific channel (e.g. 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 2-3, might even be up to 9). Thus, the link is re-established after a short time.

For Intel/LAD this timeout is recommended to be 3 seconds.

Even when enabling & using AENs it is still recommended to periodically poll the link status, as dropped AENs might not be detected.

4.6.4 Statistics

The MC may use the statistics commands as defined in NC-SI. These counters are meant mostly for debug purposes and may not all be supported on Intel Ethernet Controllers, please refer to product specific documentation for details.

The statistics are divided to 3 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 (e.g. commands, responses, AENs...). 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.

4.7 OEM Extensions

The NC-SI specification allows for extensibility by OEM's. Intel has added significant capabilities using this mechanism. Some of those capabilities are described in this document. Refer to product documentation for details

4.7.1 Get System MAC

This command allows the MC to retrieve the System MAC address used by the NC. This MAC address may be used by the MC for its purposes, such as possibly adding an offset to it and using it as a dedicated MAC address for a NC-SI channel.

4.7.2 TCO Reset

Allows the Management Controller to reset the Ethernet Controller.

4.7.3 Keep PHY Link Up

This allows the MC to block a PHY reset that could possibly lead to loss of session between the MC and a remote console.

4.7.4 Checksum Offloading

Offloads IP/UDP/TCP checksum checking of received packets from the Management Controller.

4.7.5 Additional Filtering

Intel Ethernet Controllers support the same filtering capabilities described in the SMBus section, including:

- Management 2 Host Configuration

- Flex 128 filters

- Flex TCP/UDP port filters

- IPv4/IPv6 filters

5 Troubleshooting Recommendations

5.1 General Troubleshooting

5.1.1 Remote Management Connection Dropped After Power Action

There are reports of times when a server is reset or powered down when the remote management connection is lost and cannot be re-established for up to a minute – by which time a the session has timed out and must be re-established.

Investigation into this issues has shown that this is a switch configuration. Under specific switch configurations, the switch receives a link-down notification when the server performs a power action. The switch 'assumes' the device on the port is no longer capable of handling traffic and it removes it from its internal list of active devices. It can take up to a minute for the switch to 're-learn' that there is a device on that port desiring connectivity.

During this timeout period most secure communication protocols (such as RMCP/IPMI) will timeout, causing a loss of connectivity.

Intel has added the ability to keep the PHY link up during some system actions such as a reset. This is configurable using either SMBus or NC-SI and is also available as an EEPROM configuration option.

5.1.2 Byte Order

Take care to ensure the correct Byte Order when configuring various filters. For example, when configuring VLAN filters, the Byte Order is very important. If say a VLAN tag of 0123h was desired and the incorrect order was used within the configuration command, the actual filtering may be on a tag of 1201h.

This would result in the command (whether it be over SMBus or NC-SI) will be accepted without error and all packets from the remote console (sending with a VLAN tag of 0123h) being silently dropped.

5.2 SMBus Troubleshooting

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

5.2.1 SMBus Alert Line Stays Asserted After Power Cycle

After the Intel Ethernet controller resets both ports on the controller will indicate a status

change. If the MC only reads status from one port (slave address) the other one will continue to assert the TCO alert line.

Ideally, the MC should use the SMBus ARA transaction to determine which slave asserted the SMBus Alert line. Many customers only wish to use one port for manageability thus using ARA may 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 will assert that SMBus Alert line awaiting a status read from a MC; by default this value may be 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 FFh (for this issue). Note that this timeout value is for both slave addresses. Along with setting the SMBus Notification Timeout to FFh, 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 0 for the desired port in the LAN Configuration section of the EEPROM.

The last solution for this issue is to have the MC hard-code the slave addresses and simply always read from both ports.

5.2.2 SMBus Commands are Always NACK'd by the Intel Ethernet Controller

There are many reasons why all commands sent to the controller from a MC could be NACK'd. The following are the most common:

Invalid EEPROM Image – The image itself may be invalid, or it could be a valid image however it is not a Pass-Through image, as such SMBus connectivity will be disabled.

The MC is not using the correct SMBus address – Many MC firmware vendors hard-code the SMBus address(es) into their firmware. If the incorrect values are hard-coded, the controller will not respond. Refer to the product specific EEPROM map to determine the SMBus address(es).

The SMBus address(es) can also be dynamically set using the SMBus ARP mechanism. The MC is using the incorrect

SMBus interface – The EEPROM may be configured to use one physical SMBus port, however the MC is physically connected to a different one.

Bus Interference – the bus connecting the MC and the controller may be instable.

5.2.3 Slow SMBus Clock Speed

This can happen when the SMBus connecting the MC and the controller is also tied into another device (such as a ICH6) that has a maximum clock speed of 16.6666KHz. The SMBus connection between the MC and the NC is recommended to be dedicated, and not connected to any other device.

5.2.4 Network Based Host Application Not Receiving 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 26Fh (623), and the system was also configured for a shared MAC and IP address with the Operating System and MC.

The management control to host configuration (see Section 3.5.2) in this situation was setup not to send RMCP traffic to the Operating System – this is in general 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 26Fh.

5.3 NC-SI Troubleshooting

5.3.1 Verify Electrical Connections

The NC-SI interface is a relatively high-speed interface and as such has strict requirements for clock strength, maximum bus length etc. Ensure that the layout meets these requirements.

As part of the support model for Intel Ethernet Controllers, Intel offers schematic and layout reviews.

5.3.2 NC-SI Control Packets

Take care to ensure that the MC sends complete and valid Ethernet packets when it sends Control Packets. Otherwise, packets will be silently discarded.

5.3.3 Ensure NC-SI Mode

The two manageability interfaces (SMBus and NC-SI) are mutually exclusive and must be configured within the EEPROM image. There are different EEPROM images for SMBus and NC-SI modes.

5.4 Recommendations

5.4.1 General Recommendations

5.4.1.1 Default Configuration

It is recommended that the MC not make any assumptions regarding default filter

configurations coming from an EEPROM image. While it is possible to configure an EEPROM for many defaults, there is no guarantee that the next generation of NC will have the same default configuration.

5.4.2 SMBus Recommendations

5.4.2.1 Dedicated SMBus for Manageability

When using the SMBus interface for manageability, Intel recommends that system designers allow for a dedicated SMBus. Intel Ethernet Controllers have been tested and will work when there are multiple devices on the SMBus. However, given the bandwidth required for passing Ethernet packets back and forth, a dedicated SMBus is preferred.

5.4.2.2 SMBus Fragment Size

The SMBus specification indicates a maximum SMBus transaction size of 32 bytes. Most of the data passed between the Ethernet controller and the MC over the SMBus is RMCP/RMCP+ traffic, which by its very nature (UDP traffic) will be significantly larger than 32 bytes in length. This ensures multiple SMBus transactions will be required to move a packet from the Intel Ethernet controller to the MC or to send a packet from the MC to the Intel Ethernet controller.

Recognizing the bottleneck, an Intel Ethernet controller can handle up to 240 bytes of data within a single transaction. This is a configurable setting within EEPROM. Refer to the EEPROM setting for your specific Intel Ethernet controller.

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

5.4.2.3 Enable XSUM Filtering

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

This is an easy way to offload some work from the MC.

5.4.3 NC-SI Recommendations

5.4.3.1 Use Hardware Arbitration

If the design calls for more than one physical Ethernet controller, and if the NC's support Hardware Arbitration, it is strongly recommended that it be enabled and used. Hardware Arbitration relieves the MC of having to provide a scheduler type mechanism that allows the NC's to transmit data to the MC.

6 Manageability Registers

This section details common manageability registers for several Intel Ethernet Controllers.

6.1 Manageability Control Register (MANC)

Table 6-1. MANC Register

Manageability Control Register							
Bit	82573	82571	ESB2	82575	82598	82574	82576
0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
2	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
3	EN_FLEXPOR0	EN_FLEXPOR T0	EN_FLEXPOR 0	Reserved	Reserved	Reserved	Reserved
4	EN_FLEXPOR1	EN_FLEXPOR T1	EN_FLEXPOR 1	Reserved	Reserved	Reserved	Reserved
5	EN_FLEXPOR2	EN_FLEXPOR T2	EN_FLEXPOR 2	Reserved	Reserved	Reserved	Reserved
6	EN_FLEXTCO1	EN_FLEXTCO 1	EN_FLEXTCO1	Reserved	Reserved	Reserved	Reserved
7	EN_FLEXTCO0	EN_FLEXTCO 0	EN_FLEXTCO0	Reserved	Reserved	Reserved	Reserved
8	RMCP_EN	RMCP_EN	RMCP_EN	Reserved	Reserved	Reserved	Reserved
9	0298_EN	0298_EN	0298_EN	Reserved	Reserved	Reserved	Reserved
10	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
11	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
12	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
13	ARP_REQ_EN	ARP_REQ_E N	ARP_REQ_EN	Reserved	Reserved	Reserved	Reserved
14	NEIGHBORHOOD	NEIGHBORH OOD	NEIGHBORHO OD	Reserved	Reserved	Reserved	Reserved
15	ARP_RES_EN	ARP_RES_EN	ARP_RES_EN	Reserved	Reserved	Reserved	Reserved
16	Reserved	EN_TCO_RES ET	EN_TCO_RESE T	TCO_RST_O CC	Reserved	TCO_RST_OCC	TCO_RST_OCC
17	Reserved	RCV_TCO_E N	RCV_TCO_EN	RCV_TCO_E N	RCV_TCO_E N	RCV_TCO_EN	RCV_TCO_EN

18	Reserved	KEEP_PHY_LNK	KEEP_PHY_LNK	KEEP_PHY_LNK	Reserved	KEEP_PHY_LNK	KEEP_PHY_LNK
19	Reserved	RCV_ALL	RCV_ALL	RCV_ALL	RCV_ALL	RCV_ALL	RCV_ALL
20	Reserved	MAC_16_EN	MAC_16_EN	EN_MCAST	Reserved	EN_MCAST	EN_MCAST
21	MNG2Host	MNG2Host	MNG2Host	MNG2Host	MNG2Host	MNG2Host	MNG2Host
22	EN_IP_ADDR	EN_IP_ADDR	EN_IP_ADDR	Reserved	Reserved	Reserved	BYPASS_VLAN
23	EN_XSUM	EN_XSUM	EN_XSUM	EN_XSUM	EN_XSUM	EN_XSUM	EN_XSUM
24	Broadcast_En	Broadcast_En	Broadcast_En	EN_IPV4_FLT	EN_IPV4_FLT	Reserved	EN_IPV4_FLT
25	Reserved	Reserved	Reserved	FIXED_NETTYPE	Reserved	FIXED_NETTYPE	FIXED_NETTYPE
26	Reserved	Reserved	Reserved	NET_TYPE	Reserved	NET_TYPE	NET_TYPE
27	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	EN_LINK_SEC
28	Reserved	Reserved	Reserved	Reserved	Reserved	DIS_IP_ADDR_P	Reserved
29	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
30	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
31	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

6.2 Management To Host (MNG2Host)

This section shows the Management to Host filter configuration register bit values for several Intel Ethernet controllers.

Table 6-2. Management To Host Register

MNG2Host Register (5860h)							
Bit	82573	82571	ESB2	82575	82598	82574	82576
0	Flex Port 0	Flex Port 0	Flex Port 0	MDEF 0	MDEF 0	MDEF 0	MDEF 0
1	Flex Port 1	Flex Port 1	Flex Port 1	MDEF 1	MDEF 1	MDEF 1	MDEF 1
2	Flex Port 2	Flex Port 2	Flex Port 2	MDEF 2	MDEF 2	MDEF 2	MDEF 2
3	Flex TCO 0	Flex TCO 0	Flex TCO 0	MDEF 3	MDEF 3	MDEF 3	MDEF 3
4	Flex TCO 1	Flex TCO 1	Flex TCO 1	MDEF 4	MDEF 4	MDEF 4	MDEF 4
5	Port 26Fh	Port 26Fh	Port 26Fh	MDEF 5	MDEF 5	MDEF 5	MDEF 5
6	Port 298h	Port 298h	Port 298h	MDEF 6	MDEF 6	MDEF 6	MDEF 6
7	ARP_REQ	ARP_REQ	ARP_REQ	MDEF 7	MDEF 7	MDEF 7	MDEF 7
8	ARP_RES	ARP_RES	ARP_RES	Reserved	Reserved	Reserved	Reserved
9	Broadcast	Broadcast	Broadcast	Reserved	Reserved	Reserved	Reserved
10	Neighbor	Neighbor	Neighbor	Reserved	Reserved	Reserved	Reserved

11	VLAN 0	VLAN 0	VLAN 0	Reserved	Reserved	Reserved	Reserved
12	VLAN 1	VLAN 1	VLAN 1	Reserved	Reserved	Reserved	Reserved
13	VLAN 2	VLAN 2	VLAN 2	Reserved	Reserved	Reserved	Reserved
14	VLAN 3	VLAN 3	VLAN 3	Reserved	Reserved	Reserved	Reserved
15	Ded MAC	Ded MAC	Ded MAC	Reserved	Reserved	Reserved	Reserved
16	Reserved	VLAN 4	VLAN 4	Reserved	Reserved	Reserved	Reserved
17	Reserved	VLAN 5	VLAN 5	Reserved	Reserved	Reserved	Reserved
18	Reserved	VLAN 6	VLAN 6	Reserved	Reserved	Reserved	Reserved
19	Reserved	VLAN 7	VLAN 7	Reserved	Reserved	Reserved	Reserved
20	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
21	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
22	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
23	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
24	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
25	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
26	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
27	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
28	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
29	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
30	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
31	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

6.3 Manageability Decision Filters (MDEF & MDEF_EXT)

This section shows the MDEF and MDEF_EXT filter configuration register bit values for several Intel Ethernet Controllers.

Table 6-3. MDEF Register

MDEF (5890)							
Bit	82573	82571	ESB2	82575	82598	82574	82576
0	N/A	N/A	N/A	Unicast AND	Unicast AND	Unicast AND	Unicast AND
1	N/A	N/A	N/A	Broadcast AND	Broadcast AND	Broadcast AND	Broadcast AND
2	N/A	N/A	N/A	VLAN AND	VLAN AND	VLAN AND	VLAN AND
3	N/A	N/A	N/A	IP Address	IP Address	IP Address	IP Address
4	N/A	N/A	N/A	Unicast OR	Unicast OR	Unicast OR	Unicast OR
5	N/A	N/A	N/A	Broadcast OR	Broadcast OR	Broadcast OR	Broadcast OR
6	N/A	N/A	N/A	Multicast AND	Multicast AND	Multicast AND	Multicast AND
7	N/A	N/A	N/A	ARP Request	ARP Request	ARP Request	ARP Request
8	N/A	N/A	N/A	ARP Response	ARP Response	ARP Response	ARP Response
9	N/A	N/A	N/A	Neighborhood Dsc	Neighborhood Dsc	Neighborhood Dsc	Neighborhood Dsc
10	N/A	N/A	N/A	Port 298h	Port 298h	Port 298h	Port 298h

11	N/A	N/A	N/A	Port 26Fh	Port 26Fh	Port 26Fh	Port 26Fh
12	N/A	N/A	N/A	Flex Port	Flex Port	Flex Port	Flex Port
13	N/A	N/A	N/A				
14	N/A	N/A	N/A				
15	N/A	N/A	N/A				
16	N/A	N/A	N/A			Reserved	
17	N/A	N/A	N/A			Reserved	
18	N/A	N/A	N/A			Reserved	
19	N/A	N/A	N/A			Reserved	
20	N/A	N/A	N/A			Reserved	
21	N/A	N/A	N/A			Reserved	
22	N/A	N/A	N/A			Reserved	
23	N/A	N/A	N/A			Reserved	
24	N/A	N/A	N/A			Reserved	
25	N/A	N/A	N/A			Reserved	
26	N/A	N/A	N/A			Reserved	
27	N/A	N/A	N/A	Reserved			
28	N/A	N/A	N/A	Flex TCO	Flex TCO	Flex TCO	Flex TCO
29	N/A	N/A	N/A			Reserved	
30	N/A	N/A	N/A			Reserved	
31	N/A	N/A	N/A			Reserved	

Table 6-4. MDEF_EXT Register

MDEF_EXT							
Bit	82573	82571	ESB2	82575	82598	82574	82576
0	N/A	N/A	N/A	N/A	N/A	N/A	L2 EtherType AND
1	N/A	N/A	N/A	N/A	N/A	N/A	
2	N/A	N/A	N/A	N/A	N/A	N/A	
3	N/A	N/A	N/A	N/A	N/A	N/A	
4	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
5	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
6	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
7	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
8	N/A	N/A	N/A	N/A	N/A	N/A	L2 EtherType OR
9	N/A	N/A	N/A	N/A	N/A	N/A	
10	N/A	N/A	N/A	N/A	N/A	N/A	
11	N/A	N/A	N/A	N/A	N/A	N/A	
12	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
13	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
14	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
15	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
16	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
17	N/A	N/A	N/A	N/A	N/A	N/A	Reserved

18	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
19	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
20	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
21	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
22	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
23	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
24	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
25	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
26	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
27	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
28	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
29	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
30	N/A	N/A	N/A	N/A	N/A	N/A	Reserved
31	N/A	N/A	N/A	N/A	N/A	N/A	Reserved

7 Additional Documentation

There are a number of specifications used for manageability, links to some are listed here:

SMBus Specification: <http://smbus.org/specs/smbus20.pdf>

I2C Specification: <http://smbus.org/specs/smbus20.pdf>

NC-SI Specification: http://www.dmtf.org/standards/published_documents/DSP0222.pdf

IPMI Specification: <http://www.intel.com/design/servers/ipmi/spec.htm>

Intel® Ethernet Controller Documentation:

<http://support.intel.com/support/network/adapter/index.htm>

NOTE: This page intentionally left blank.