

Data ONTAP™ 6.4

System Administration

Network Management Guide

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089 USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: <http://www.netapp.com>

Part number 210-00321
January 2003

Copyright and trademark information

Copyright information

Copyright © 1994–2003 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

No part of this book covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Portions of this product are derived from the Berkeley Net2 release and the 4.4-Lite-2 release, which are copyrighted and publicly distributed by The Regents of the University of California.

Copyright © 1980–1995 The Regents of the University of California. All rights reserved.

Portions of this product are derived from NetBSD, which is copyrighted by Carnegie Mellon University.

Copyright © 1994, 1995 Carnegie Mellon University. All rights reserved. Author Chris G. Demetriou.

Permission to use, copy, modify, and distribute this software and its documentation is hereby granted, provided that both the copyright notice and its permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS “AS IS” CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Software derived from copyrighted material of The Regents of the University of California and Carnegie Mellon University is subject to the following license and disclaimer:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notices, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER

IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Software derived from copyrighted material of Network Appliance, Inc. is subject to the following license and disclaimer:

Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use and purchase of this product do not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp and the Network Appliance design are registered trademarks of Network Appliance, Inc. in the United States, Canada, and the European Union. Network Appliance is a registered trademark of Network Appliance, Inc. in Monaco and a trademark of Network Appliance, Inc. in the United States and Canada. FAServer is a registered trademark of Network Appliance, Inc. in the United States and the European Union. NetCache is a registered trademark of Network Appliance, Inc. in the European Union and Japan, and a trademark of Network Appliance, Inc. in the United States. SnapCopy is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States. WAFL is a registered trademark of Network Appliance, Inc. in the United States, the European Union, and Canada. FilerView, SecureShare, SnapManager, SnapMirror and SnapRestore are registered trademarks of Network Appliance, Inc. in the United States. Data ONTAP is a trademark of Network Appliance, Inc. in the United States and Canada. Snapshot is a trademark of Network Appliance, Inc. in the United States and the European Union. NetApp—the Network Appliance Company is a registered trademark of Network Appliance, Inc. in the United States and other countries. ApplianceWatch, BareMetal, Center-to-Edge, ContentDirector, DataFabric, MultiStore, NearStore, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapDrive, SnapVault, Vfiler, and Web Filer are trademarks of Network Appliance, Inc. in the United States.

Network Appliance is a licensee of the CompactFlash and CF Logo trademarks.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Table of Contents

	Preface	.ix
Chapter 1	Network Interface Configuration	1
	Understanding filer network interfaces	2
	Understanding jumbo frames	5
	Using the ifconfig command	7
	Changing an interface address	8
	Configuring an alias for an interface	10
	Configuring a media type on an Ethernet interface.	12
	Configuring flow control	15
	Setting the MTU on an interface	17
	Changing the interface status	18
	Viewing interface configuration	19
	Displaying network interface statistics	21
Chapter 2	ATM Configuration	23
	About ATM and ATM LANE	24
	Preparing the ATM adapter for LANE	29
	Verifying that the ATM adapter is installed and functioning	31
	Verifying a working connection to the ATM network	32
	Verifying that the UNI is operational	33
	Configuring the LANE Configuration Server address	35
	Configuring the ATM adapter for an Emulated LAN	38
	Adding an Emulated LAN to the ATM adapter	39
	Configuring the logical Ethernet interface	41
	Deleting an Emulated LAN from an ATM adapter	42
	Checking and completing the Emulated LAN configuration.	43
	Verifying the communications link	44
	Checking the configuration settings	45
	Checking the other elements of the Emulated LAN	46
	Modifying load balancing and failover	48
	Saving the ATM configuration commands in the /etc/rc file	50
	Saving the host and IP address data in the /etc/hosts file	51

	Understanding FORE/IP over SPANS	52
	Managing FORE/IP and PVCs	53
	Establishing FORE/IP PVCs on the filer.	54
	Displaying information about a FORE/IP PVC	56
	Displaying the FORE/IP configuration	57
	Changing the ATM adaptation layer for FORE/IP and SPANS	59
	Deleting a FORE/IP PVC	60
Chapter 3	Network Routing Configuration	61
	About routing in Data ONTAP.	62
	Enabling or disabling routed	65
	Displaying the routing table	66
	Disabling fast path	67
	Modifying the routing table.	68
	Diagnosing network problems	71
Chapter 4	Host Name Resolution.	73
	Maintenance of host information	74
	Using the /etc/hosts file to maintain host information	75
	Using DNS to maintain host information.	78
	Using NIS to maintain host information	83
	Using LDAP to authenticate multiprotocol file access.	87
	Changing the host name search order	92
Chapter 5	Filer Monitoring Using SNMP	95
	Understanding SNMP implementation in Data ONTAP.	96
	Contents of the Network Appliance custom MIB	97
	SNMP options and commands supported by Data ONTAP	100
	Understanding traps in Data ONTAP.	104
	Defining or modifying a trap	105
	SNMP parameters	107

Chapter 6	Vif Configuration	.113
	Understanding vifs	.114
	Types of vifs	.116
	Vif management	.119
	Creating a single-mode vif	.121
	Changing an active interface in a single-mode vif	.123
	Creating a multi-mode vif	.124
	Adding interfaces to a vif	.126
	Deleting an interface from a vif	.127
	Displaying the status of a vif	.128
	Displaying statistics of a vif	.132
	Destroying a vif	.133
	Second-level vifs	.134
	Understanding second-level vifs on a single filer	.135
	Creating a second-level vif on a single filer	.136
	Understanding second-level vifs in a cluster.	.138
	Creating a second-level vif in a cluster	.140
 Chapter 7	 Virtual LAN (VLAN) Configuration.	 .143
	Understanding VLANs	.144
	VLANs in Data ONTAP	.148
	Managing VLANs on a filer	.150
	Creating and configuring a VLAN on a filer.	.151
	Adding an interface to a VLAN	.154
	Deleting a VLAN.	.155
	Modifying a VLAN interface	.157
	Viewing VLAN statistics.	.158
 Chapter 8	 Internet Protocol Security Configuration	 .161
	Understanding IPsec	.162
	Setting up IPsec.	.163
	Managing security policies	.165
	Viewing security associations	.171
 Appendix A	 Network Interface Statistics	 .173
	Fast Ethernet interface statistics	.174

Znynx 10Base-T/100Base-TX Ethernet interface statistics 177

Ethernet Controller II, III, and IV interface statistics 180

Gigabit Ethernet interface statistics. 184

FDDI interface statistics 186

ATM interface statistics 190

Appendix B

Improving filer performance 191

Index 193

Preface

About this guide

This guide describes how to configure and manage network interfaces, virtual network interfaces (vifs), virtual local area networks (VLANs), and routing on Network Appliance™ filers that run Data ONTAP™ 6.4 software and are Serving Data By Design™. It covers all filer models; however, some filer models do not support all of the networking interfaces. See the hardware guide for your filer to identify which interfaces are supported on your filer.

Audience

This guide is for system administrators who are familiar with operating systems that run on the filer's clients, such as UNIX®, Windows 95™, Windows NT®, and Windows® 2000. It also assumes that you are familiar with how the NFS, CIFS, DAFS, and HTTP protocols are used for file sharing or transfers. This guide doesn't cover basic system or network topics, such as IP addressing, routing, and network topology; it emphasizes the characteristics of the NetApp® filer.

Terminology and conventions

When the term “filer” is used, the information applies to all Network Appliance filer models, including the 900 series storage appliances and R100 series NearStore™ storage appliances.

You can enter Data ONTAP commands either on the system console or from any client computer that can access the filer through a Telnet session.

In examples that illustrate commands executed on a UNIX workstation, this guide uses the command syntax of SunOS 4.1.x. The command syntax and output might differ, depending on your version of UNIX.

This guide uses the term “type” to mean pressing one or more keys on the keyboard. It uses the term “enter” to mean pressing one or more keys and then pressing the Enter key.

Keyboard conventions

When describing key combinations, this guide uses the hyphen (-) to separate individual keys. For example, “Ctrl-D” means pressing the “Control” and “D” keys simultaneously. Also, this guide uses the term “Enter” to refer to the key that generates a carriage return, although the key is named “Return” on some keyboards.

Typographic conventions

The following table describes typographic conventions used in this guide.

Convention	Type of information
<i>Italic font</i>	Words or characters that require special attention. Placeholders for information you must supply. For example, if the guide says to enter the <code>arp -d hostname</code> command, you enter the characters “arp -d” followed by the actual name of the host. Book titles in cross-references.
Monospaced font	Command and daemon names. Information displayed on the system console or other computer monitors. The contents of files.
Bold monospaced font	Words or characters you type. What you type is always shown in lowercase letters, unless you must type it in uppercase letters.

Special messages

This guide contains special messages that are described as follows:

Note

A note contains important information that helps you install or operate the system efficiently.

Caution

A caution contains instructions that you must follow to avoid damage to the equipment, a system crash, or loss of data.

WARNING

A warning contains instructions that you must follow to avoid personal injury.

About this chapter

This chapter discusses the types of interfaces supported on the filer; how the interfaces are named; the `ifconfig` command, and how you can use this command to configure the network interfaces on your filer; and the `ifstat` command, and how you can use this command to obtain detailed statistics on various interfaces supported on your filer.

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“Understanding filer network interfaces”](#) on page 2
- ◆ [“Understanding jumbo frames”](#) on page 5
- ◆ [“Using the ifconfig command”](#) on page 7
- ◆ [“Changing an interface address”](#) on page 8
- ◆ [“Configuring an alias for an interface”](#) on page 10
- ◆ [“Configuring a media type on an Ethernet interface”](#) on page 12
- ◆ [“Configuring flow control”](#) on page 15
- ◆ [“Setting the MTU on an interface”](#) on page 17
- ◆ [“Changing the interface status”](#) on page 18
- ◆ [“Viewing interface configuration”](#) on page 19
- ◆ [“Displaying network interface statistics”](#) on page 21

Understanding filer network interfaces

Types of interfaces the filer supports

The filer supports the following interface types:

- ◆ Ethernet and quad-port Ethernet
- ◆ Gigabit Ethernet
- ◆ Fiber Distributed Data Interface (FDDI)
- ◆ Asynchronous Transfer Mode (ATM)—Emulated LAN and FORE/IP
- ◆ Virtual—vifs, virtual LAN (VLAN), vh

Note

Data ONTAP imposes a limit of 128 interfaces (including physical, vifs, VLAN, vh, and loopback interfaces) per filer.

How interfaces are numbered

For network adapters, the interface numbers are assigned based on the slot in which the interface adapter is installed.

VLAN interfaces are displayed in the *slot_number-vlan_id* format, where *slot_number* is the slot in which the network adapter is installed and *vlan_id* is the identifier of the VLAN configured on the interface. For example, e8-2, e8-3, and e8-4 are three VLAN interfaces for VLANs 2, 3, and 4, configured on interface e8.

You can assign numbers for vifs and the emulated LAN interfaces.

How multiple ports are identified

Some Ethernet adapters support four ports. These adapters are referred to as quad-port interfaces. Data ONTAP uses a letter to refer to each port on a quad-port interface. The following table shows the relationship of port numbers to letters.

Port number	Letter
1	a
2	b
3	c

Port number	Letter
4	d

Interface naming conventions

The following table shows how interfaces are named, and how interface names are combined with interface slot numbers and port letters (if applicable) or VLAN identifier (if applicable) to make their names unique.

Interface type	Letter used in name	Examples of names
Ethernet (single) and Gigabit Ethernet	e	e0 e1
Ethernet (quad-port)	e	e0a e0b e0c e0d e1a e1b
vifs	Any user-specified string.	web_vif proxy_vif
VLAN	e	e8-2 e8-3
FDDI	f	f0 f1
ATM	a (used only to configure clusters)	a0 a1
ATM—Emulated LAN	e1 (default)	e10
ATM—Fore IP	fa	fa0

How Data ONTAP creates host names

The first time you run the `setup` program, Data ONTAP creates a host name for each installed interface by appending the interface name to the host name of the filer.

Examples of host names

Example 1: A filer named *toaster* with a single Ethernet interface in slot 0 and a quad-port Ethernet interface in slot 1 has the host names given in the following table.

Interface	Host name
Single-port Ethernet interface in slot 0	toaster-e0
Quad-port Ethernet interface in slot 1	toaster-e1a toaster-e1b toaster-e1c toaster-e1d

Example 2: A filer named *toaster* with an FDDI interface in slot 1 has the host name toaster-f1.

Understanding jumbo frames

What jumbo frames are

Jumbo frames are packets that are longer than the standard Ethernet (IEEE 802.3) frame size of 1,518 bytes. The frame size definition for jumbo frames is vendor-specific because jumbo frames are not part of the IEEE standard. The most commonly used jumbo frame sizes are 9,018 bytes and higher.

Advantage of jumbo frames

Because the data is transferred in longer frames than standard Ethernet, jumbo frames reduce the CPU overhead by reducing the number of frames processed.

Limitation of jumbo frames

On the filer, jumbo frames are supported on all copper or fiber Gigabit Ethernet Controllers and interfaces numbered II or higher. The interfaces must be operating at 1,000 Mbps.

Frame size and MTU size definitions

The two commonly used terms to define the size of a jumbo frame are frame size and Maximum Transmission Unit (MTU) size. The definitions of these terms are as follows:

Frame size: The frame size of an Ethernet frame is the sum of the Ethernet header (14 bytes), the payload (IP datagram, up to 9,000 bytes), and the Frame Check Sequence (FCS) field (4 bytes).

MTU size: The MTU size specifies the maximum number of bytes of data (the payload) that can be encapsulated in an Ethernet frame. For example, the MTU size of an Ethernet frame (RFC 894) is 1,500 bytes.

Filer frame size and MTU size definition

The default frame size defined for a jumbo frame on the filer is 9,018 bytes, and the maximum supported size is 9,210 bytes. These sizes correspond to MTU sizes of 9,000 and 9,192, respectively.

Ways to set up jumbo frames on the filer

You can set up jumbo frames on the filer in the following two ways:

- ◆ `setup` command—The `setup` command prompts you for jumbo frames if you have an interface that supports jumbo frames on your filer.
- ◆ `ifconfig` command—If your system is already running Data ONTAP, you can enable jumbo frames by setting the MTU size on an interface. For information about how to set up the MTU size, see [“Setting the MTU on an interface”](#) on page 17.

Network infrastructure requirements

Before enabling jumbo frames on the filer, the following network infrastructure requirements (as appropriate) must be satisfied:

- ◆ The switch ports must also have jumbo frames enabled.
- ◆ If the filer and the client are on different subnets, the next-hop router must be configured for jumbo frames.

Client configuration recommendations

You should consider the following client configuration recommendations to gain performance benefits of jumbo frames:

- ◆ You must configure jumbo frames on the client. To find out how to configure jumbo frames on your client, see the network adapter documentation for your client.
- ◆ You should enlarge the client’s TCP window size. The minimum value for the client’s window size should be $2 * (\text{MTU size} - 40)$, and the maximum value can be the highest value your system allows. Typically, the maximum value you can set for your client’s TCP window is 65535.

If the filer is configured to support jumbo frames and the client is not, the communication between the filer and client occurs at the client’s frame size.

- ◆ You must ensure that the User Datagram Protocol (UDP) clients are configured with the same MTU size as the filer.

The UDP clients do not communicate their MTU. Therefore, the filer and the client should be configured with the same MTU size, or the receiver might not receive packets being sent.

- ◆ You must consider the per-route MTU if the filer and the client are on different subnets.

If the filer and the client (both configured to use jumbo frames) are on different subnets and an intermediate subnet does not support jumbo frames, the advantages of using jumbo frames are lost. In the case of TCP clients, jumbo frames are not used at all. In the case of UDP clients, IP fragmentation occurs and leads to performance degradation.

Using the ifconfig command

About the ifconfig command

The `ifconfig` command assigns an address to a network interface and configures interface parameters such as `netmask`, `mediatype`, and `mtusize`.

How the ifconfig command works

An `ifconfig` command is included in the `/etc/rc` file of the root volume for each filer interface that you specify an IP address for during the filer setup. After the filer has been set up, the `ifconfig` commands in the `/etc/rc` file are used to configure the interfaces on consequent filer reboots.

How to use the ifconfig command

You can use the `ifconfig` command at the command line of the filer to change any of the interface parameters when the filer is operating. However, such changes are not automatically included in the `/etc/rc` file. If you want the configuration changes you make with the `ifconfig` command to be persistent after a reboot, include the `ifconfig` commands in the `/etc/rc` file.

The ifconfig command syntax

The `ifconfig` command syntax is as follows:

```
ifconfig [-a | [interface  
[[alias | -alias] address] [up | down]  
[netmask mask] [broadcast address] [mtusize size]  
[mediatype { tp | tp-fd | 100tx | 100tx-fd | 1000fx | auto }]  
[flowcontrol { none | receive | send | full }]  
[trusted | untrusted] [ wins | -wins ]  
[[partner { address | interface }} | [-partner]]]
```

For additional information

The `ifconfig` option `partner` is discussed in the *Storage Management Guide*. For detailed information about the `ifconfig` command and all the other options available with this command, see the `na_ifconfig(1)` man page on the filer.

Changing an interface address

What you can change

You can use the `ifconfig` command to change the following interface parameters:

- ◆ IP address
- ◆ Network mask
- ◆ Broadcast address

Changing the IP address

To change the IP address of an interface, complete the following step.

Note—
You must configure the network mask of the interface when configuring the IP address if the network mask is not classfull (Class A, B, C, or D). By default, a classfull network mask is applied.

Step	Action
1	Enter the following command: <pre>ifconfig interface address</pre> <i>interface</i> is the name of the interface of which you want to configure the address. <i>address</i> is the new IP address for this interface. Example: You configure a quad-port Ethernet interface e3a with an IP address 192.168.25.10 with the following command: <pre>ifconfig e3a 192.168.25.10</pre>

Changing the network mask

To change the network mask of an interface, complete the following step.

Step	Action
1	Enter the following command: <pre>ifconfig interface netmask mask</pre> <i>interface</i> is the name of a specific interface. <i>mask</i> is the new network mask for this interface. Example: You configure a 24-bit mask for the interface e3a configured in the previous example with the following command: <pre>ifconfig e3a netmask 255.255.255.0</pre>

Changing the broadcast address

To change the broadcast address of an interface, complete the following step.

Step	Action
1	Enter the following command: <pre>ifconfig interface broadcast address</pre> <i>interface</i> is the name of a specific interface. <i>address</i> is the new broadcast address to be used for this interface. Example: You set a broadcast address of 192.168.25.250 for the network 192.168.25.10 with subnet mask 255.255.255.0, configured in an earlier example, with the following command: <pre>ifconfig e3a broadcast 192.168.25.250</pre>

Configuring an alias for an interface

Available options

There are two alias options available for the `ifconfig` command:

- ◆ `alias`—Enables you to establish an additional IP address for an interface.
- ◆ `-alias`—Enables you to remove an aliased IP address from an interface.

Using the `alias` option

The `alias` option is useful when you are changing the IP address of an interface to a new address, but also want to keep accepting packets addressed to the old IP address.

You can use the `alias` option at the command line. However, the IP address configured using the `alias` option at the command line is lost if the filer reboots. If you want to make your changes persistent across reboots, include these changes in the `/etc/rc` file of the root volume.

You cannot set up an IP address and an alias for an interface in the same `ifconfig` command; you must configure the IP address for the interface before setting up the alias.

Using the `-alias` option

The `-alias` option is useful when you want to stop using the old IP address configured on an interface but do not want to reboot the filer.

Configuring an alias

To configure an alias for an interface, complete the following step.

Step	Action
1	Enter the following command: <pre>ifconfig interface alias address [netmask mask]</pre> <i>interface</i> is the interface you want to set up an alias for. <i>address</i> is the alias IP address of the interface. <i>mask</i> is the subnet mask. Example: In this example, the interface e0 (already configured with an IP address of 172.28.50.21) is set up with an alias IP address of 172.28.50.30. <pre>ifconfig e0 alias 172.28.50.30 netmask 255.255.255.0</pre>

Configuring a media type on an Ethernet interface

About media types

You can use an `ifconfig` command option called `mediatype` to configure the speed and duplex or specify autonegotiation for an Ethernet interface. The media types available for the filer interfaces are described in the following table.

Note

For 10Base-T and 100Base-T interfaces, the `mediatype` option of the interface and its link partner (the interface on the other end of the connection) must be the same, that is, both interfaces must be configured either for speed and duplex or for autonegotiation. Otherwise, a duplex mismatch will occur, which can lead to poor performance.

Media type	Description
tp	10Base-T, half-duplex
tp-fd	10Base-T, full-duplex
100tx	100Base-T, half-duplex
100tx-fd	100Base-T, full-duplex
auto	Autonegotiate speed, duplex, and flow control. See “ How media type auto works ” on page 13.
1000fx	1000Base-SX, full-duplex (Gigabit Ethernet Controller)
auto-1000fx	Autonegotiate (Gigabit Ethernet Controller)

Understanding media type settings for a fiber Gigabit Ethernet interface

The media type setting for a fiber Gigabit Ethernet interface depends on the type of Gigabit Ethernet controller installed on your filer. You can use the `sysconfig` command to identify the type of controller on your filer.

If the `sysconfig` command output indicates that you have a Gigabit Ethernet Controller *number*, where *number* is a roman numeral, you do not specify the media type. The interface only supports full-duplex. In addition, by default, the interface detects whether the link partner autonegotiates. Based on the outcome of autonegotiation, the operational flow control setting is determined.

If the `sysconfig` command output indicates that you have a Gigabit Ethernet Controller (without the roman numeral), you must ensure that the filer interface and its link partner have the same media type configured (that is, autonegotiation enabled or disabled); otherwise, the Gigabit Ethernet Controller does not work. The media type `auto-1000fx` enables autonegotiation and `1000fx` disables autonegotiation on the Gigabit Ethernet Controller.

Because the Gigabit Ethernet Controller only supports full-duplex, it does not need to be configured on the interface.

How media type auto works

The behavior of media type `auto` is determined by the type of network adapter installed on your filer. The following table lists the autonegotiated parameters and the possible values for each type of network adapter.

Note

You can use the `ifstat` command to determine the speed, duplex, and flow control settings that get negotiated between the interface of the filer and the link partner. For more information about the `ifstat` command and the information it displays, see [“Displaying network interface statistics”](#) on page 21.

Network adapter	Autonegotiated parameters (autonegotiated values)
10Base-T/100Base-T	Speed and duplex (half or full)
100Base-T/1000Base-T	Speed. If speed is 100 Mbps, duplex (half or full) and flow control are negotiated. If speed is 1000 Mbps, flow control is negotiated.
10Base-T/100Base-T/ 1000Base-T	Speed. If speed is 10 Mbps, duplex (half or full) is negotiated. If speed is 100 Mbps, duplex and flow control are negotiated. If speed is 1000 Mbps, flow control is negotiated.
Gigabit Ethernet	Flow control.

Configuring media types

To configure a media type on an Ethernet interface on a filer, complete the following step.

Step	Action
1	Enter the following command, choosing an appropriate media type option: <pre>ifconfig interface mediatype { tp tp-fd 100tx 100tx-fd 1000fx auto-1000fx auto }</pre> <i>interface</i> is the name of an interface. Example: You configure an interface e2 as a 100Base-T full-duplex interface with the following command: <pre>ifconfig e2 mediatype 100tx-fd</pre>

Configuring flow control

Interfaces you can configure flow control on

You can configure flow control for Gigabit Ethernet interfaces and for 100Base-T/1000Base-T and 10Base-T/100Base-T/1000Base-T interfaces.

What the flowcontrol option is used for

The `flowcontrol` option of the `ifconfig` command sets the flow control of the abovementioned interfaces. This *configured* flow control setting is advertised during autonegotiation. If autonegotiation succeeds, the *operational* flow control setting is determined based on the negotiated speed and the value advertised by the other device. If autonegotiation fails, the configured flow control setting is used.

How flow control works

Flow control manages the flow of frames between two directly connected link partners. This is done by using packets called Pause frames. For example, a link partner A sends a Pause On frame to link partner B when its receive buffers are nearly full. The link partner B suspends transmission until it receives a Pause Off frame from the link partner A or a specified timeout threshold is reached. Thus, flow control can reduce or eliminate dropped packets due to overrun.

Flow control types

The following table describes the types you can specify for the `flowcontrol` option.

Flow control type	Description
None	No flow control
Receive	Only receive flow control frames
Send	Only send flow control frames
Full	Send and receive flow control frames

**Commands to
configure and view
flow control**

You can use the `ifconfig` command to configure and display the configured flow control setting and the `ifstat` command to view the operational flow control setting. If you do not specify the `flowcontrol` option with the `ifconfig` command, then the configured flow control setting defaults to full.

**Configuring flow
control**

To configure the flow control of an interface, complete the following step.

Step	Action
1	Enter the following command, choosing one of the flow control options: <pre>ifconfig interface flowcontrol { none receive send full }</pre> <i>interface</i> is the specific interface name. Example: You set flow control type <code>none</code> for an interface <code>e8</code> with the following command: <pre>ifconfig e8 flowcontrol none</pre>

Setting the MTU on an interface

About the mtusize option

The `mtusize` option specifies the MTU to use for transmission between your host and the filer. You can use this option to set the jumbo frame size on any copper or fiber Gigabit Ethernet Contoller numbered II or higher.

Note _____
Jumbo frame support requires an interface that is operating at 1,000 Mbps.

You can also use the `mediatype` option to set the frame size on an FDDI interface when the network infrastructure does not support the default FDDI frame size. The MTU size does not include the media header or FCS (checksum).

Setting the MTU

To set the MTU for an interface, complete the following step.

Step	Action
1	Enter the following command: <pre>ifconfig interface mtusize size</pre> <i>interface</i> is the specific name of the interface to be changed. <i>size</i> is the size of MTU this interface should have. Example: You specify an MTU size of 9000 for a Gigabit Ethernet interface e8 with the following command: <pre>ifconfig e8 mtusize 9000</pre>

Changing the interface status

Reasons to change the interface status

You might have to change the status of an interface for one of the following reasons:

- ◆ Installing a new interface
- ◆ Upgrading an interface
- ◆ Troubleshooting network connectivity issues
- ◆ Disabling a failed interface

Changing the interface status

To change the status of an interface to up or down, complete the following step.

Step	Action
1	Enter the following command, choosing the appropriate option: ifconfig interface {up down} <i>interface</i> specifies the name of the interface you want to change the status of.

Viewing interface configuration

Viewable interface configurations

You can view the current configuration of a specific interface or all interfaces that exist on a filer.

Viewing the configuration of a specific interface

To view the configuration of a specific interface, complete the following step.

Step	Action
1	Enter the following command: ifconfig interface <i>interface</i> is the name of the specific network interface you want to view configuration information about.

Viewing the configuration of all interfaces

To view the configuration of all interfaces, complete the following step.

Step	Action
1	Enter the following command: ifconfig -a The current configuration of all network interfaces on the filer is displayed.

Example

The following example displays the configuration of all network interfaces on the filer toaster:

```
toaster> ifconfig -a
e8: flags=48043<UP,BROADCAST,RUNNING,MULTICAST> mtu 9000
    inet 172.28.50.2 netmask 0xffffffff broadcast 172.28.50.255
    ether 00:d0:b7:60:04:a3 (auto-1000sx-fd-down) flowcontrol full
e1: flags=8042<BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 00:c0:95:f9:d2:6c (tp-down)
e0: flags=848043<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.28.50.21 netmask 0xffffffff broadcast 172.28.50.255
```

```
ether 00:a0:98:00:3d:28 (100tx-fd-up)
lo:  flags=948049<UP,LOOPBACK,RUNNING,MULTICAST,TCPCSUM> mtu 9188
    inet 127.0.0.1 netmask 0xff000000 broadcast 127.0.0.1
```

In this example, the *e8* interface is a Gigabit Ethernet interface with MTU size of 9000 (**mtu 9000**). The interface is configured to be Up (<**UP**,BROADCAST,RUNNING,MULTICAST>). However, the current status of the interface is Down (auto-1000sx-fd-**down**).

Displaying network interface statistics

Commands for displaying network interface statistics

You use the `ifstat` and the `netstat` commands to display network interface statistics. The `ifstat` command displays statistics maintained by the networking code, network adapter, and the network driver. For more information about the `netstat` command, see the `na_netstat(1)` man page.

About the ifstat command

You use the `ifstat` command to determine the media type setting that is negotiated between the interface of the filer and the link partner.

The output of the `ifstat` command might contain many fields, because different types of interfaces — for example, Ethernet, Gigabit Ethernet, FDDI, and ATM — have different statistics. For details of the statistics displayed for each network interface, see Appendix A, “[Network Interface Statistics](#),” on page 173.

Displaying interface statistics with the ifstat command

Displayed statistics are gathered from the time of the last reboot or from the last time you cleared them.

Note _____
For cluster systems, the `ifstat` command reflects the filer that is executing the commands, not the partner.

To display or clear interface statistics, complete the following step.

Step	Action
1	Enter the following command: <code>ifstat [-z] -a interface</code> -z zeros (clears) the statistics. -a lists statistics for all the filer’s interfaces. <i>interface</i> indicates the interface for which you want to display or clear statistics.

For more information about the `ifstat` command, see the `na_ifstat(1)` man page.

About this chapter

This chapter outlines the features and concepts of Asynchronous Transfer Mode (ATM) and describes the key components of an Emulated LAN, including the LAN Emulation (LANE) Client, LANE Server, LANE Configuration Server, and Broadcast and Unknown Server (BUS). It also describes how to use FORE/IP over Simple Protocol for ATM Network Signaling (SPANS).

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“About ATM and ATM LANE”](#) on page 24
- ◆ [“Preparing the ATM adapter for LANE”](#) on page 29
- ◆ [“Configuring the ATM adapter for an Emulated LAN”](#) on page 38
- ◆ [“Checking and completing the Emulated LAN configuration”](#) on page 43
- ◆ [“Understanding FORE/IP over SPANS”](#) on page 52
- ◆ [“Managing FORE/IP and PVCs”](#) on page 53

About ATM and ATM LANE

What ATM is

ATM is a network technology that combines the features of cell-switching and multiplexing to offer reliable and efficient network services. ATM provides an interface between the network and devices such as workstations and routers. The asynchronous nature of ATM means that bandwidth is made available on demand instead of allocating slots of transmission time to network devices, as it is in a synchronous system employing Time-Division Multiplexing (TDM).

ATM employs fixed-sized cells of 53 bytes each as the basic unit of transmission. Each cell consists of a 5-octet header, identifying the source of the transmission and other information, and a 48-octet payload containing the user data and headers for higher-level protocols. This architecture permits text, voice, graphics, and video to share the same network without any one source dominating network bandwidth.

ATM employs a star topology with an ATM switch acting as the hub of the network. All devices are connected directly to this hub, making network configuration and troubleshooting more straightforward, as well as offering dedicated bandwidth to the central switch.

Ways to use ATM on a filer

You can use ATM in two ways on a filer:

- ◆ ATM LANE, which provides the services of an Ethernet LAN to higher-level network application software
- ◆ FORE/IP over Permanent Virtual Connection (PVC) or Switched Virtual Connection (SVC), using Simple Protocol for ATM Network Signaling (SPANS) to establish the SVCs

The filer can simultaneously support FORE/IP and LANE over User-Network Interface (UNI) 3.0 or 3.1 on the same physical interface.

Differences between LANs and ATM

In addition to the issue of connection-based versus connectionless service, LANs differ from ATM in the following ways:

- ◆ The shared medium approach of LANs makes them ideal for broadcast and multicast messages.
- ◆ The Media Access Control (MAC) addresses used to identify a network interface of a LAN are typically based on the manufacturer's serial numbers. This means that the address is constant, independent of the network to which the interface is connected.

About LANE

Many organizations use a LAN for their internal data communications. Examples of these LANs include Ethernet/IEEE 802.3 and IEEE 802.5 (Token Ring). However, LANs typically offer a connectionless service, while ATM is always connection-oriented. This means that to use LAN-based applications using ATM, some form of LANE is required.

Benefits provided by LANE

LANE is an ATM service that offers the following benefits:

- ◆ You can run LAN-based application software on an ATM network.
- ◆ You can interconnect ATM networks to conventional LANs with existing bridging methods.

This permits applications running on ATM-connected end systems to interoperate with those running on traditional LAN-based devices. These LAN-based end systems can also communicate with each other across the ATM network.
- ◆ You can run more than one Emulated LAN on the same ATM network, with each Emulated LAN independent of the others.

About ATM cause codes

Data ONTAP displays cause code strings when ATM connections for LANE Configuration Server, LANE Server, Broadcast and Unknown Server, or LAN Emulation Client normally or abnormally terminate. They describe the reason for the connection termination or rejection. For more information about these cause codes, see the ATM Forum's UNI 3.0 and 3.1 specifications.

What an Emulated LAN is

An Emulated LAN comprises a group of ATM-attached devices, logically analogous to a group of LAN stations attached to an Ethernet/802.3 LAN segment. You can configure several Emulated LANs within an ATM network, and membership in an Emulated LAN is independent of where the end station is physically connected. The end station can belong to multiple Emulated LANs.

Components of an Emulated LAN

An Emulated LAN consists of the following components:

- ◆ A single LANE Service, which itself consists of a LANE Server and a BUS. Each of these components is discussed in more detail later in this chapter.
- ◆ A set of LANE Clients.
A LANE Client communicates with other LANE Clients and with the LANE Service using Virtual Channel Connections (VCCs) in an ATM SVC environment.

What a LANE Client is

The LANE Client is part of an ATM end station or a MAC bridge. It performs data forwarding as well as address resolution, among other control functions. The LANE Client supplies higher-level software with an Ethernet/IEEE 802.3 MAC layer interface that enables LAN-based application software to communicate over ATM networks just as it would over a traditional LAN.

How LANE Clients communicate

LANE Clients communicate with other clients using the LANE Service and represent users by their MAC addresses. A LANE Client employs separate VCCs for data and control communication, including LE_ARP requests for address resolution. User data intended for another end station is encapsulated in IEEE 802.3 frames.

What LANE Service is

The LANE Service, consisting of a LANE Server, BUS, and LANE Configuration Client Server, can be implemented as part of one or more end systems or as part of the ATM switch. When you implement the service in a distributed fashion over multiple devices, benefits include parallel operation as well as better error recovery through redundancy.

Within the LANE Service, the LANE Server is responsible for coordinating the control functions, while the LANE Configuration Server serves network clients by supplying Emulated LAN configuration information. The BUS forwards broadcast and multicast frames and handles unresolved unicast frames.

What a LANE Server does

The LANE Server performs control services within an Emulated LAN, including registering MAC addresses and resolving these addresses to corresponding ATM addresses when requested. LANE Clients can also request the LANE Server to resolve a route descriptor to an ATM address. The LANE Server responds to the client or forwards the request to other clients, which might be in a better position to service the query.

The LANE Server also coordinates the process of a LANE Client joining an Emulated LAN. There is a single LANE Server per Emulated LAN, and each LANE Server has a unique ATM address.

How a LANE Configuration Server functions in an administrative domain

The LANE Configuration Server maintains information concerning all the Emulated LANs in an administrative domain, and supplies the LANE Client with the ATM address for the LANE Server in the domain. Before joining an Emulated LAN, the LANE Client first exchanges configuration information with the LANE Configuration Server.

Upon successfully communicating with the LANE Configuration Server, the LANE Client receives a list of Emulated LANs that are available to join. There is a single LANE Configuration Server per administrative domain for Emulated LANs within a domain.

What a BUS does

The BUS accepts and processes data sent by a LANE Client to the broadcast MAC address 'FFFFFFFFFFFF'. The BUS also handles all multicast messages, as well as initial unicast frames sent by a LANE Client before the ATM address has been resolved.

The BUS thereby offers services that emulate the shared medium capabilities typical of a LAN. The BUS does this by serializing the frames and retransmitting them to the appropriate LANE Clients within the Emulated LAN.

Although there might be multiple BUSes defined within an Emulated LAN, each LANE Client is associated with only a single BUS per Emulated LAN.

UNI load balancing

The User-Network Interface (UNI), which serves as an interface point between ATM end systems and the ATM switch, supports both automatic adapter failover and load balancing across multiple adapters connected to the same physical ATM switch. This means that the UNI signaling module automatically detects which adapters are connected to a single physical network and places all adapters connected to that network in a failover group.

Both incoming and outgoing connections are directed to the least used adapter in a load balancing group. If an adapter in a load balancing group fails, the connections on that adapter are automatically transferred to another adapter.

UNI load balancing and adapter failover do not require any configuration. However, you can statically configure or disable UNI load balancing.

How LANE handles addressing and address resolution

LANs use a MAC address to designate the source and destination addresses for end stations. For LANE to function transparently, it must offer similar functionality. In practical terms, this means that each LANE Client has a MAC address, and when more than one LANE Client uses the same network interface, each LANE Client is assigned a different MAC address.

When the LANE Client needs to send data to another MAC address, it must first resolve that address to an ATM address, thus enabling it to establish a data-direct VCC to that LANE Client. To do so, it sends an LE_ARP_REQUEST to the LANE Server. The LANE Server can either respond to this request or forward it to other LANE Clients. If the specified MAC address is known anywhere on the Emulated LAN, the originating LANE Client gets an LE_ARP_RESPONSE frame containing the corresponding ATM address.

LANE standards supported in this release

This release of Data ONTAP supports the following features and standards:

- ◆ ATM Forum LANE Version 1.0 LANE Client Support
- ◆ UNI 3.0 and 3.1
- ◆ Integrated Local Management Interface (ILMI) Address Registration
- ◆ ILMI MIB extensions for LANE

The software works with the FORE OC3 ATM network interface and on Network Appliance F520, F630, F720, F740, and F760 filers. The software provides Ethernet LANE services, with the capability to configure multiple Emulated LANs on each available network interface.

The current release does not support ATM LANE 2.0, Multiprotocol Over ATM (MPOA), Classical IP (CLIP), or Token Ring LANE services.

Preparing the ATM adapter for LANE

Preparing for ATM LANE

Before the ATM adapter can communicate using ATM LANE, you need to ensure that the ATM adapter is installed correctly and that it can communicate with the network. This section describes the steps you take to enable the ATM adapter in the filer to communicate using ATM LANE.

Prerequisites for configuring ATM adapters

Before you start configuring the ATM adapters in the filer, ensure that you meet the prerequisites in the following table.

Prerequisite	Explanation
Complete the normal setup procedure for your filer, run automatically when you first install the filer, or run the <code>setup</code> command for an existing installation.	You need an ATM switch with one or more Emulated LANs already configured on the switch (with the corresponding configurations for the LANE Configuration Server, LANE Server, and BUS).
Know the LANE Configuration Server address for the Emulated LAN that you want the filer to join.	In most cases, the LANE Configuration Server has been configured to use the “well-known” address; however, this might be different at your site.
If your site has multiple Emulated LANs, know the ATM address of the LANE Configuration Server for each Emulated LAN you want a client to join.	You can configure each ATM adapter in the filer to communicate over multiple Emulated LANs on the network.

Note

If you need more information about creating an Emulated LAN or configuring the LANE Configuration Server, LANE Server, and BUS, see the documentation that came with your switch.

**For detailed
information**

The following sections discuss how to configure the ATM adapter for LANE.

- ◆ “[Verifying that the ATM adapter is installed and functioning](#)” on page 31
- ◆ “[Verifying a working connection to the ATM network](#)” on page 32
- ◆ “[Verifying that the UNI is operational](#)” on page 33
- ◆ “[Configuring the LANE Configuration Server address](#)” on page 35

Preparing the ATM adapter for LANE

Verifying that the ATM adapter is installed and functioning

Verifying that the adapter works

To verify that the ATM adapter is functioning, complete the following step.

Step	Action
1	Enter the following command: atm adinfo You can perform the <code>atm adinfo</code> command again at any time to determine the correct unit number.

Example of the atm adinfo command

Sample output from the `adinfo` command is as follows:

```
atm adinfo
a1:unit 1:      PCA-200E Media=OC3-MM-SC HW=2.0.1 FW=4.2.0
Serial=4201600 Slot=1
                MAC=00:20:48:40:1C:80
a2:unit 2:      PCA-200E Media=OC3-MM-SC HW=2.0.1 FW=4.2.0
Serial=4201621 Slot=2
                MAC=00:20:48:40:1C:95
a3:unit 3:      PCA-200E Media=OC3-MM-SC HW=2.0.1 FW=4.2.0
Serial=4201955 Slot=3
                MAC=00:20:48:40:1D:E3
```

Interpreting the output

You should see lines for each ATM adapter in your filer that is functioning properly. The presence of the lines indicate that the adapter has passed its self-test procedure and that the filer initialized the adapter. The `adinfo` command also displays the device name for each of the installed adapters, as well as the unit number, at the beginning of each line.

The unit number uniquely identifies the ATM adapter in the filer, and there is a one-to-one mapping between the device names and unit numbers. The device name consists of the prefix ‘a’ followed by the physical slot number. The unit number is the slot number.

Preparing the ATM adapter for LANE

Verifying a working connection to the ATM network

Verifying that the connection works

To check that the ATM adapter in your filer is properly connected using the appropriate cabling, complete the following step.

Step	Action
1	Enter the following command: atm adstat -d <i>device</i> <i>device</i> is the name of the ATM adapter whose connection you want to check.

Example of the atm adstat -d command

The following command displays statistics about the ATM adapter in slot 1 of your filer:

```
atm adstat -d a1
```

Sample output from this command is as follows:

```
Device statistics:
      Buffer Allocation Failures
      Type 1           Type 2
      Small   Large   Small   Large   Receive Queue Full   Carrier
      0         0       0       0       0                   0       ON
```

Interpreting the output

The Carrier column should indicate ON. If it does not, your cabling is incorrectly connected or faulty, or your ATM network is malfunctioning or misconfigured.

Note_____

If you need information about connecting the cabling to your filer’s ATM adapter, see the appropriate section in the hardware guide that came with your filer.

Preparing the ATM adapter for LANE

Verifying that the UNI is operational

Checking whether the UNI is operational

The filer ATM address is automatically registered with the switch; therefore, you only use the `uniconfig` command to display configuration parameters, check the UNI version number, and ensure that the UNI is operational.

To check whether the UNI is operational, complete the following step.

Step	Action
1	Enter the following command: <pre>atm uniconfig show [-unit unit]</pre> <i>unit</i> is the UNI you want to check. Note_____ If you do not specify the unit number, the UNI information for all ATM adapters in the filer is displayed. _____

Example of the atm uniconfig show command

Abbreviated sample output (showing unit 3 only) from the `atm uniconfig show` command is as follows:

```
atm uniconfig show
UNI parameters for unit 3
=====
VPI/VCI                : 0/5
AAL type                : 5
QoS                    : UBR
UNI configured version  : 3.1
UNI operating version   : 3.1
SSCOP operational state : operational
Primary ATM address     :
47.0005.80.ffe100.0000.f21a.4d19.002048401de3.00
UNI failover configuration
```

=====

Status: dynamic

Groups: (0) (1*) (2)

Interpreting the output

The following items should enable you to verify that the ATM interface is operational:

- ◆ The UNI configured version and UNI operating version value each should be 3.1.
- ◆ The SSCOP operational state should indicate that the UNI is operational. If you see inoperational instead, the ATM card is improperly connected to the network or the switch is improperly configured.
- ◆ The Primary ATM address should be a valid ATM address for your network. If you see an address consisting entirely of zeros, the ATM card is improperly connected to the network or there might be a configuration problem.

Configuring the LANE Configuration Server address

About configuring the LANE Configuration Server address

For the filer to participate in an Emulated LAN, you must configure the ATM adapter with the address of the LANE Configuration Server. The LANE Client joins an Emulated LAN by first exchanging configuration information with the LANE Configuration Server. The LANE Configuration Server then supplies the client with the ATM address for the LANE Server.

Knowing the LANE Configuration Server address, the system can now determine all existing Emulated LANs, as well as the ATM address of the LANE Server. However, the LAN Type remains unknown until you configure the adapter to join the Emulated LAN, which is discussed in [“Configuring the ATM adapter for an Emulated LAN”](#) on page 38.

Configuring the LANE Configuration Server address

To configure the LANE Configuration Server address for your ATM adapter, complete the following steps.

Step	Action
1	Enter the following command to set the LANE Configuration Server address: <pre>atm elconfig set -lecs <i>ATM_address</i> -wellknown -manual -unit <i>unit_number</i></pre> <i>ATM_address</i> is the ATM address of the LANE Configuration Server on the network. -wellknown indicates that the well-known ATM address will be used. -manual places the host in manual configuration mode; configuration information is not retrieved from the LANE Configuration Server. <i>unit_number</i> is the unit designator for the ATM adapter in the filer. Note_____ You do not have to specify the unit number if only one ATM adapter is installed in a filer. _____

Step	Action
2	Enter the following command to check that the adapter is communicating with the LANE Configuration Server: atm elconfig show -all
3	Study the output. If a separate line fails to appear for each configured Emulated LAN, grouped by adapter, verify that each Emulated LAN is configured properly for the ATM switch. For more information, see your switch vendor's documentation.

Example of the atm elconfig set command with a well-known address

The following command sets the LANE Configuration Server address to the well-known ATM address for the adapter in slot 2:

```
atm elconfig set -lecs -wellknown -unit 2
```

Example of the atm elconfig set command without a well-known address

If the LANE Configuration Server on your network does not use the well-known address, specify the LANE Client Server ATM address in place of wellknown, as shown in the following command:

```
atm elconfig set -lecs
47.0079.00.000000.0000.0000.0000.00a03e000001.00 -unit 2
```

Example of the atm elconfig show command

Abbreviated sample output (showing Emulated LANs available through the LANE Configuration Server for Adapter 2 only) from the atm elconfig show -all command is as follows:

```
ELANs on Adapter 2
```

```
=====
```

```
LECS (current): 47.0079.00.000000.0000.0000.0000.00a03e000001.00
```

```
ELAN          LAN Type      LES ATM Address
```

```
=====
```

```
eighteenKMTU   Unknown
```

```
47.0005.80.ffe100.0000.f21a.4d19.0020481a4d19.14
```

```
=> default          Unknown
47.0005.80.ffe100.0000.f21a.4d19.0020481a4d19.f0

      nineKMTU       Unknown
47.0005.80.ffe100.0000.f21a.4d19.0020481a4d19.11
```

Interpreting the output

Each Emulated LAN on the network should appear in the output, and the LAN Type and LANE Server ATM addresses should appear as expected. There should be a separate line for each configured Emulated LAN on the network, grouped and arranged for each ATM adapter that is installed in the filer.

An arrow to the left of an Emulated LAN signifies that the ATM adapter has been configured to operate on that Emulated LAN. For information about configuring adapters to operate on an Emulated LAN, see “[Adding an Emulated LAN to the ATM adapter](#)” on page 39.

The LANE Server ATM address should appear valid. If it does, you know that the ATM adapter in the filer is communicating properly with the switch. If the LANE Server ATM address is all zeros, it might mean that the cable connection is not working or something else is improperly configured at the switch.

The LANE Configuration Server ATM address should match the address that you specified earlier in “[Configuring the LANE Configuration Server address](#)” on page 35.

Configuring the ATM adapter for an Emulated LAN

About the configuration

You must configure the ATM adapter to enable it to operate on one or more Emulated LANs.

For detailed information

The following sections discuss the actions you take to configure an ATM adapter for an Emulated LAN:

- ◆ [“Adding an Emulated LAN to the ATM adapter”](#) on page 39
- ◆ [“Configuring the logical Ethernet interface”](#) on page 41
- ◆ [“Deleting an Emulated LAN from an ATM adapter”](#) on page 42

Configuring the ATM adapter for an Emulated LAN

Adding an Emulated LAN to the ATM adapter

Prerequisite

Before you configure the ATM adapter to operate on an Emulated LAN, you must configure the LANE Configuration Server ATM address for each adapter, as described in “[Configuring the LANE Configuration Server address](#)” on page 35. Also, the Emulated LAN you specify must already have been configured at the switch.

Adding an Emulated LAN to the adapter

To add the Emulated LAN to the adapter, complete the following steps.

Step	Action
1	Enter the following command: <pre>atm elconfig add ELAN -if interface -les ATM_address -type ethernet -unit unit_number</pre> <i>ELAN</i> is the Emulated LAN that you want the adapter to join. <i>interface</i> is the logical network interface. The <i>-les</i> flag is for joining Emulated LANs whose configuration information is not returned by a LANE Configuration Server. Note You only use the <i>-les</i> flag when the <i>-manual</i> flag is set in the <code>atm elconfig set</code> command. Do not use the <i>-les</i> flag if the LANE Configuration Server address is set to <code>wellknown</code>. <i>ATM_address</i> is the LANE Server ATM address. <i>unit_number</i> is the unit designator for the ATM adapter in the filer. Note If there is only a single ATM adapter in the filer, you do not need to specify the unit number in the command. The <code>atm elconfig</code> command sets it automatically.
2	To add more than one Emulated LAN to an adapter, repeat Step 1.

Example of the atm elconfig add command

The following command adds the adapter unit 2 to the nineKMTU Emulated LAN of type Ethernet, using interface el1:

```
atm elconfig add nineKMTU -if el1 -type ethernet -unit 2
```

Interpreting the output

This example assumes that the nineKMTU Emulated LAN already exists on the switch.

The el1 interface refers to a logical interface, thereby enabling you to configure more than one logical interface for the same physical ATM adapter. This means that you can use the `atm elconfig add` command repeatedly to configure the filer to communicate over multiple Emulated LANs using a single physical ATM adapter. Only Ethernet emulated networks are supported.

Configuring the ATM adapter for an Emulated LAN

Configuring the logical Ethernet interface

Configuring the interface

After the ATM adapter joins an Emulated LAN, you need to assign an IP address to the (logical) network interface and configure additional parameters.

To configure a logical Ethernet interface, complete the following steps.

Step	Action
1	Enter the following command: <pre>ifconfig interface address netmask mask up</pre> <i>interface</i> is the name of the logical network interface. <i>address</i> is the IP address associated with the interface. <i>mask</i> is the network mask that is selected according to the class of the IP address. For more information about the netmask parameter and the <code>ifconfig</code> command, see the <i>Command Reference</i>. Example: The following command configures the <code>el0</code> logical Ethernet network interface, assigning a corresponding IP address and netmask: <pre>ifconfig el0 172.20.12.19 netmask 255.255.252.0 up</pre>
2	To configure more than one logical Ethernet interface, repeat Step 1.

Configuring the ATM adapter for an Emulated LAN

Deleting an Emulated LAN from an ATM adapter

Deleting an Emulated LAN from an adapter

To delete an Emulated LAN from an ATM adapter, complete the following steps.

Step	Action
1	Enter the following command to configure the interface that you want to delete down: <pre>ifconfig interface down</pre> <i>interface</i> is the name of the logical network interface that you want to delete. Example: The following command marks the el1 logical network interface as down: <pre>ifconfig el1 down</pre>
2	Enter the following command: <pre>atm elconfig delete ELAN -unit unit_number</pre> <i>ELAN</i> is the ELAN that you want to delete from the adapter. <i>unit_number</i> is the unit designator for the ATM adapter in the filer. Example: The following command deletes the nineKMTU Emulated LAN for the adapter designated by unit 2: <pre>atm elconfig delete nineKMTU -unit 2</pre>

Checking and completing the Emulated LAN configuration

About completing the configuration

After you configure the ATM adapter for an Emulated LAN, you should verify your configuration to ensure that it is correct.

For detailed information

The following sections discuss the actions you take to check and complete the Emulated LAN configuration:

- ◆ [“Verifying the communications link”](#) on page 44
- ◆ [“Checking the configuration settings”](#) on page 45
- ◆ [“Checking the other elements of the Emulated LAN”](#) on page 46
- ◆ [“Modifying load balancing and failover”](#) on page 48
- ◆ [“Saving the ATM configuration commands in the /etc/rc file”](#) on page 50
- ◆ [“Saving the host and IP address data in the /etc/hosts file”](#) on page 51

Verifying the communications link

Verifying the communications link

After you add an Emulated LAN to an adapter and configure the interface, you need to check to ensure that the filer can communicate with other clients through the Emulated LAN. The easiest way to do this is to ping another LANE Client on the Emulated LAN to ensure that information is traveling out through the ATM adapter and back again.

To ping a LANE Client, or any other client, complete the following step.

Step	Action
1	Enter the following command: ping host <i>host</i> is the IP address of the computer to which you want to send an ICMP ECHO_REQUEST datagram.

Example

The following command sends the datagram to host 204.125.14.45, and waits for a response:

```
ping 204.125.14.45
```

If the host responds, ping prints “*host* is alive.” Otherwise, ping resends the ECHO_REQUEST once a second. If the host does not respond after 20 seconds, ping prints the following output:

```
no answer from host.
```

Checking and completing the Emulated LAN configuration

Checking the configuration settings

Verifying adapter configurations

After you verify the communication link, you should check the state of the adapters in the filer to ensure that the configuration is correct.

To check the configuration settings, complete the following steps.

Step	Action
1	Enter the following command: atm elconfig show -all
2	Verify there is an arrow preceding the Emulated LAN name, which indicates that the adapter has joined the Emulated LAN.
3	Check that the LAN Type is Ethernet.
4	Check that the LANE Server ATM address is a valid ATM address. If the address is all zeros, it indicates a configuration problem at the switch.

Example of the atm elconfig show -all command

Abbreviated sample output (showing Emulated LANs on Adapter 2 only) from this command is as follows:

```
atm elconfig show -all
ELANs on Adapter 2
=====
LECS (current): 47.0079.00.000000.0000.0000.00a03e000001.00
      ELAN           LAN Type      LES ATM Address
      ====           =====      =====
      eighteenKMTU   Ethernet
47.0005.80.ffe100.0000.f21a.4d19.0020481a4d19.14
=> default           Unknown
47.0005.80.ffe100.0000.f21a.4d19.0020481a4d19.f0
=> nineKMTU          Ethernet
47.0005.80.ffe100.0000.f21a.4d19.0020481a4d19.11
```

Checking and completing the Emulated LAN configuration

Checking the other elements of the Emulated LAN

Checking the other elements You should also check the other elements of the Emulated LAN to ensure they are configured and operating correctly.

To check the other elements of the Emulated LAN, complete the following steps.

Step	Action
1	Enter the following command: atm elconfig show -configured
2	Check the output to verify that the LANE Server, LANE Configuration Server, and BUS ATM addresses are all valid, and that the state of the Emulated LAN is operational.
3	Use the <code>netstat -i</code> command to check the MTU of each Emulated LAN to verify that the setting matches the configuration created on the switch.

Example of the atm elconfig show -configured command Abbreviated sample output (showing information related to the eighteenKMTU Emulated LAN on adapter 2 only) from the `elconfig show -configured` command is as follows:

```
atm elconfig show -configured
ELAN Name                : eighteenKMTU
Interface                : el1
Configured Unit          : 2
MAC Address              : 00:20:48:08:12:c3
LEC Address              :
47.0005.80.ffe100.0000.f20f.6d4c.0020480812c3.00
LECS Address             :
c5.0079.00.000000.00000000000000a03e000001.00
Configuration Direct VCC : unit=2 vpi/vci=0/279
LES Address              :
47.0005.80.ffe100.0000.f21a.4d19.0020481a4d19.f0
```

```
Control Direct VCC      : unit=2 vpi/vci=0/280
Control Distribute VCC  : unit=2 vpi/vci=0/281
BUS Address             :
47.0005.80.ffe100.0000.f21a.4d19.0020481a4d19.f0
Multicast Send VCC      : unit=2 vpi/vci=0/282
Multicast Forward VCC   : unit=2 vpi/vci=0/283
LAN Type                : Ethernet/IEEE 802.3
Maximum Frame Size      : 1516
State                   : operational
MPOA                    : disabled
LECID                   : 13
```

Modifying load balancing and failover

About load balancing

Load balancing enables incoming and outgoing traffic to be spread across ATM adapters in a group.

By default, all ATM adapters are dynamically assigned to load-balancing groups. In this case, no configuration is necessary to activate the load balancing and failover features. However, you can override the default by changing the adapters to static mode so groups can be manually configured, for instance, when a failover group contains adapters on different switches, resulting in reduced network performance.

Requirements for load balancing

Load balancing does not depend on any nonstandard extensions to the UNI. However, the switch must support the following:

- ◆ Registering the same ATM address on multiple ports
- ◆ Registering multiple ATM addresses on a single port

If the switch does not support these features, load balancing and failover are automatically disabled.

Modifying load balancing groups

To modify load-balancing groups, complete the following step.

Step	Action
1	Enter the following command: <pre>atm uniconfig set failover [-state off static dynamic] [-group unit ...]</pre> -state dynamic is the default. Use -state off to disable load balancing and failover, -state static to put UNI load-balancing in a static mode, and -group unit... to specify the load-balancing group membership. Note The parameters to the -group option of the atm uniconfig set failover command specify the ATM adapters (units) that should be logically assigned to a load-balancing and failover group. If you specify a unit that has already been assigned to another group, the unit is automatically removed from the original group before being assigned to the new group.

Example of the atm uniconfig set failover command

The following example demonstrates how you disable load balancing and failover:

```
atm uniconfig set failover -state off
```

Checking and completing the Emulated LAN configuration

Saving the ATM configuration commands in the /etc/rc file

Saving the configuration commands

By saving the ATM configuration information in the /etc/rc file, you avoid having to reconfigure the adapters manually each time the filer is restarted.

To save the configuration commands in the /etc/rc file for automatic execution at boot time, complete the following steps.

Step	Action
1	Mount the root file system and add the configuration commands to the /etc/rc file using a text editor, such as vi.
2	Save your changes.

Sample /etc/rc file with ATM configuration commands

Following is a sample portion of an /etc/rc file containing configuration commands for three ATM adapters:

```
# unit 1
elconfig set -lecs -wellknown -unit 1
elconfig add default -if el0 -type ethernet -unit 1
ifconfig el0 172.20.12.19 netmask 255.255.252.0 up
# unit 2
elconfig set -lecs -wellknown -unit 2
elconfig add nineKMTU -if el1 -type ethernet -unit 2
ifconfig el1 201.201.201.219 netmask 255.255.255.0 up
# unit 3
elconfig set -lecs -wellknown -unit 3
elconfig add eighteenKMTU -if el2 -type ethernet -unit 3
ifconfig el2 201.201.210.219 netmask 255.255.255.0 up
elconfig add nineKMTU -if el3 -type ethernet -unit 3
ifconfig el3 201.201.210.220 netmask 255.255.255.0 up
elconfig wait
```

Saving the host and IP address data in the /etc/hosts file

Saving the host and IP address

To save the host and IP address information of the Emulated LAN configuration in the /etc/hosts file, complete the following steps.

Step	Action
1	Mount the root file system and add the host and IP address to the /etc/hosts file using a text editor, such as vi.
2	Save your changes.

Sample /etc/hosts file entry

Following is a sample portion of the /etc/hosts file containing the host and IP address information for the ATM adapters:

```
172.20.12.19      myfiler-e10
201.201.201.219   myfiler-e11
201.201.210.219   myfiler-e12
201.201.210.220   myfiler-e13
```

Additional information for filers with only ATM adapters

If you have a filer that has only ATM adapters, the /etc/hosts file must contain an entry for the filer’s host name.

The host name is not displayed as part of the command prompt until you add the host name in one IP entry and reboot the filer. On filers that include other types of network interfaces, the installation setup procedure automatically adds the host name entry to the /etc/hosts file.

Example

Following is the first line in a sample /etc/hosts file:

```
172.20.12.19      myfiler   myfiler-e10
```

Understanding FORE/IP over SPANS

Version supported	Data ONTAP currently supports only FORE/IP 5.3. For more information about the differences between FORE/IP 5.3 and older versions, see the FORE documentation.
Types of SPANS connections ATM supports	ATM supports two types of SPANS connections: ◆ SVC ◆ PVC
When SVCs get assigned	Data ONTAP dynamically assigns SVCs when interoperating with ATM hosts and with switches that support the FORE/IP SPANS protocols.
When to use PVCs	You use ATM PVCs to interoperate with ATM hosts and with switches that do not support FORE/IP SPANS. For example, if you are not using a FORE systems switch, PVCs can connect FORE equipment at each end through non-FORE switches.
How FORE/IP interfaces allow communication	For each physical ATM interface, Data ONTAP creates a FORE/IP interface, called fa, at boot time. The fa interface supports FORE/IP on top of SPANS signaling. FORE/IP allows communication as follows: ◆ Using AAL4 or AAL5 ATM adaptation layer types with no encapsulation ◆ Using a broadcast Address Resolution Protocol (ARP) for SPANS address resolution ◆ Using direct communication of all hosts on a physical ATM network without the use of IP routers

Note

Data ONTAP does not support FORE/IP load balancing or failover options.

Managing FORE/IP and PVCs

About establishing and deleting PVCs

PVCs are static; for each destination, you must establish (attach the IP layer to) a PVC explicitly and delete (detach the IP layer from) the PVC explicitly.

For each destination that needs to establish a PVC with the filer, you must establish an outgoing PVC and an incoming PVC in three places:

- ◆ On the filer
- ◆ On the destination ATM host
- ◆ On all interconnecting ATM switches

For detailed information

The following sections describe the actions involved in managing FORE/IP PVCs:

- ◆ [“Establishing FORE/IP PVCs on the filer”](#) on page 54
- ◆ [“Displaying information about a FORE/IP PVC”](#) on page 56
- ◆ [“Displaying the FORE/IP configuration”](#) on page 57
- ◆ [“Changing the ATM adaptation layer for FORE/IP and SPANS”](#) on page 59
- ◆ [“Deleting a FORE/IP PVC”](#) on page 60

What this section does not discuss

The following topics are not discussed in this section:

- ◆ Establishing a FORE/IP PVC on the remote ATM host
Set up a FORE/IP PVC on the remote ATM host according to the documentation for that host.
- ◆ Establishing a FORE/IP PVC on interconnecting ATM switches
On the interconnecting ATM switches, assign virtual channels corresponding to the VPI and VCI entries made on the filer and the remote ATM host according to the documentation for those switches.

Establishing FORE/IP PVCs on the filer

Process for establishing FORE/IP PVCs

The process for establishing FORE/IP PVCs on a filer includes the following tasks:

- ◆ Establishing an outgoing FORE/IP PVC
- ◆ Establishing an incoming FORE/IP PVC

FORE/IP PVC configuration variables

When establishing an outgoing or incoming FORE/IP PVC, replace the following variables with their respective values in the command line.

Variable	Description
<i>hostname</i>	Name or IP address of the remote host.
<i>iface</i>	Name of the ATM interface. This is usually <i>fan</i> , where <i>n</i> is a number.
<i>vpi</i>	VPI (virtual path identifier); this must be 0.
<i>vci</i>	VCI (virtual channel identifier); this number must have the following properties: ◆ It must not be in use on your filer. ◆ It must be less than 1,024. ◆ It must obey the limits of the destination host and interconnecting devices.
<i>aal</i>	ATM adaptation layer type. It must be 4 or 5, and should be the AAL type supported by the destination host, which is typically 5. The default is 5. Note _____AAL4 is not supported on ForeRunner HE622 (OC-12) adapters. _____

Variable	Description
<i>encap</i>	Encapsulation type. Specify one of the following encapsulation types: ◆ null (no encapsulation; this is the default) ◆ llc_routed (IEEE LLC encapsulation for routed PDUs) ◆ llc_bridged_8023 (IEEE LLC encapsulation for Ethernet/802.3 bridged PDUs) It should be the same as the encapsulation type used by the destination host. If the encapsulation type is llc_bridged_8023, you must include <i>addr</i>.
<i>addr</i>	Six-byte colon-separated destination MAC address.

Establishing an outgoing FORE/IP PVC on the filer

To establish an outgoing FORE/IP PVC on the filer, complete the following step.

Step	Action
1	Enter the following command: <pre>atm atmarp -s <i>hostname</i> <i>pvc</i> <i>iface</i> <i>vpi</i> <i>vci</i> [<i>aal</i> [<i>encap</i> [<i>addr</i>]]]</pre>

Establishing an incoming FORE/IP PVC on the filer

To establish an incoming FORE/IP PVC on the filer, complete the following step.

Step	Action
1	Enter the following command: <pre>atm atmarp -l <i>pvc</i> <i>iface</i> <i>vpi</i> <i>vci</i> [<i>aal</i> [<i>encap</i>]]</pre>

Displaying information about a FORE/IP PVC

Displaying information about FORE/IP PVCs

Data ONTAP enables you to display address resolution information for incoming and outgoing PVCs so that you can verify the current settings.

To display information about all FORE/IP PVCs and other interfaces on a host, complete the following step.

Step	Action
1	Enter the following command: atm atmarp [<i>hostname</i> -a] <i>hostname</i> is the name or IP address of a specific remote host; use the -a flag to display information about all the current FORE/IP PVCs.

Example

If you use the -a flag, a display similar to the following appears:

```
atm atmarp -a

iface=a5 switch.port=f21a2420.56 vpi.vci=0.114 aal=5
encapsulation=NULL
iface=a5 switch.port=f21a2420.25 vpi.vci=0.113 aal=5
encapsulation=NULL
```

Managing FORE/IP and PVCs

Displaying the FORE/IP configuration

**FORE/IP
information
displayed**

You can display the FORE/IP configuration information to verify the current ATM adapter settings. The following types of information are displayed:

- ◆ Fore/IP parameters
- ◆ Connectionless VC parameters
- ◆ SPANS signaling VC parameters

**Displaying FORE/IP
configuration
information**

To display the current FORE/IP configuration information on an ATM adapter, complete the following step.

Step	Action
1	Enter the following command: atm atmconfig device <i>device</i> is the ATM adapter name.

**Sample atm
atmconfig
command output**

The following is sample output from the atm atmconfig command:

```
atm atmconfig fa0
FORE IP parameters for fa0
=====
MTU:                9188
SVC peak rate:      (unlimited)

Connectionless VC parameters
=====
VPI/VCI:            0/14
AAL:                5
peak rate:          (unlimited)
```

```
SPANS signaling VC parameters
=====
VPI/VCI:      0/15
AAL:          5
peak rate:    (unlimited)
```

Changing the ATM adaptation layer for FORE/IP and SPANS

When to change the AAL

You can change the FORE/IP ATM adaptation layer, for instance, when you install an OC-12 adapter and you need to change the AAL from 4 to 5.

See “[FORE/IP PVC configuration variables](#)” on page 54 for a description of command variables.

Changing the FORE/IP AAL

To change the FORE/IP AAL, complete the following step.

Step	Action
1	Enter the following command: <pre>atm atmconfig -c vpi vci aal device</pre> <i>device</i> is the ATM adapter name; use the -c switch to display information for FORE/IP. Example: atm atmconfig -c 0 14 5 fa3

Changing the SPANS AAL

To change the SPANS AAL, complete the following step.

Step	Action
1	Enter the following command: <pre>atm atmconfig -s vpi vci aal device</pre> <i>device</i> is the ATM adapter name; use the -s switch to display information for SPANS. Example: atm atmconfig -s 0 15 5 fa3

Managing FORE/IP and PVCs

Deleting a FORE/IP PVC

Deleting an outgoing FORE/IP PVC

To delete an outgoing FORE/IP PVC entry, complete the following step.

Step	Action
1	Enter the following command: atm atmarp -d <i>hostname</i> <i>hostname</i> is the name or IP address of the remote host.

Deleting an incoming FORE/IP PVC

To delete an incoming FORE/IP PVC for a remote host, complete the following step.

Step	Action
1	Enter the following command: atm atmarp -x <i>iface vpi vci</i> <i>iface</i> is the name of the interface. <i>vpi</i> and <i>vci</i> are the values of the VPI and VCI of the FORE/IP PVC to be deleted for the specific interface.

About this chapter

This chapter discusses how Data ONTAP manages routing, how it handles different types of packet requests, and how you can modify the routing table and diagnose network problems.

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“About routing in Data ONTAP”](#) on page 62
- ◆ [“Enabling or disabling routed”](#) on page 65
- ◆ [“Displaying the routing table”](#) on page 66
- ◆ [“Disabling fast path”](#) on page 67
- ◆ [“Modifying the routing table”](#) on page 68
- ◆ [“Diagnosing network problems”](#) on page 71

About routing in Data ONTAP

About Data ONTAP routing

Although the filer can have multiple network interfaces, it does not function as a router. The Data ONTAP software does not route packets between the interfaces of the filer on behalf of other network hosts; however, it can route its own packets.

To route Network File System (NFS) over User Datagram Protocol (UDP) and all TCP traffic, Data ONTAP uses a mechanism called fast path. For all other IP traffic Data ONTAP uses the information available in the kernel routing table. The kernel routing table is maintained by a daemon called `routed`.

Note

Data ONTAP always uses conventional IP routing table lookups for routing all traffic on a FORE/IP Asynchronous Transfer mode (ATM) interface.

What fast path is

Fast path is an alternate routing mechanism available in Data ONTAP. Instead of using the routing table of the filer to route, this mechanism uses the source Media Access Control (MAC) address of the incoming packet as the destination MAC address of the outgoing packet for NFS-over-UDP and TCP traffic transmitted from the filer. Using this mechanism provides the following advantages:

- ◆ Load balancing between multiple filer interfaces on the same subnet
The load balancing is achieved by sending responses on the same interface of the filer as incoming requests.
- ◆ Increasing filer performance
The increase in filer performance is achieved by skipping routing table lookups.

Fast path is enabled automatically on the filer; however, you can disable it.

NFS-over-UDP: The NFS-over-UDP traffic uses fast path only when sending a reply to a request. The reply packet is sent out on the same interface that the request packet came in on. For example, a filer named `toaster` uses the `toaster-e1` interface to send reply packets in response to NFS-over-UDP requests received on the `toaster-e1` interface.

TCP: Because data is acknowledged as part of the TCP protocol, Data ONTAP can use fast path on every TCP packet transmitted except the very first SYN packet (if Data ONTAP initiates a connection). For fast path, the interface used to transmit a packet is the same interface the last packet was received on.

For TCP connections, Data ONTAP automatically turns off fast path if it detects that using fast path in a network setup is not optimal.

**Effect of fast path:
Telnet works, but
ping fails**

If fast path is enabled and the `routed` daemon on a filer stops working, you might notice that telnet sessions to a filer can still be established from a non-local subnet, even though you cannot ping the filer interfaces. This happens because the fast path mechanism skips routing table lookups. As a result, the routing table is not used to respond to any NFS-over-UDP or TCP connection requests. Therefore, Telnet requests (TCP) succeed, while ping requests (ICMP) fail.

**Effect of fast path
on asymmetric
routing**

If fast path is enabled on a filer in an asymmetric network, the destination MAC address of the response packet will be that of the router that forwarded the incoming packet. However, in asymmetric networks the router forwarding packets to the filer is not the one forwarding the packets that the filer sends back. Therefore, you might have to disable fast path.

**The routed
implementation in
Data ONTAP**

To manage the routing table, Data ONTAP invokes the `routed` daemon at boot time. The `routed` implementation in Data ONTAP is similar to the conventional UNIX implementation. It supports RIPv1 (RFC 1058), RIPv2 (RFC 1723), and the ICMP Router Discovery Protocol (RFC 1256).

Generally, `routed` acts as a silent router and never broadcasts its routing table. However, you can set `routed` to generate RIP response messages that advertise a host route to the primary interface of the filer.

For more information about `routed`, see the `na_routed(1)` man page.

**Default route setup
in Data ONTAP**

Data ONTAP uses the default route entry to route to destinations that it does not explicitly know about in its routing table. You can set the default route in Data ONTAP either during the initial setup or later by modifying the `/etc/rc` file.

If you are upgrading your filer to Data ONTAP 6.0 or later and currently use the `/etc/dgateways` file to set a default route, Network Appliance recommends that you use the `/etc/rc` file, router discovery, or RIP instead. The `/etc/dgateways` file was deprecated in Data ONTAP 6.0.

Example: The following sample `/etc/rc` file shows the `route add` command used to add a default route.

```
hostname tpubs-f720
ifconfig e0 172.28.50.21 netmask 255.255.255.0 mediatype 100tx-fd
route add default 172.28.50.1 1
routed on
```

Enabling or disabling routed

About enabling or disabling routed

You can turn the `routed` daemon on or off.

You can edit the `/etc/rc` file in the root volume to keep the `routed` daemon turned off across filer reboots. However, Network Appliance recommends that you always keep `routed` on, even if you do not want Data ONTAP to make routing decisions based on routing updates. Turning `routed` off could cause unexpected routing behavior in Data ONTAP.

Turning routed on or off

To turn the `routed` daemon on or off, complete the following step.

Step	Action
1	Enter the following command: <code>routed {on off}</code> Use <code>on</code> to enable routing or <code>off</code> to disable routing.

Displaying the routing table

About the routing table

The routing table contains the current routes that have been established and are currently in use. The routing table also contains a default route that is used if Data ONTAP cannot find an explicit route for a particular destination in its routing table.

Displaying the routing table

To display the Data ONTAP routing table, complete the following step.

Step	Action
1	Enter the following command: netstat -nr

Example of displaying a routing table

The following example displays a routing table:

```
toaster> netstat -nr
Routing tables

Internet:
Destination      Gateway           Flags           Refs      Use  Interface
default          172.28.50.1      UGS             5         5860  e0
127.0.0.1        127.0.0.1       UH              1          262  lo
172.28           link#1          UC              0           0  e0
172.28.50.1      0:e0:52:1:dd:66 UHL             1           0  e0
172.28.50.3      8:0:20:9b:37:e6 UHL             0           4  e0
172.28.50.18     8:0:20:94:1c:ce UHL             0           0  e0
172.28.50.255    ff:ff:ff:ff:ff:ff UHL             0        3903  e0
172.28.255.255   ff:ff:ff:ff:ff:ff UHL             1        1733  e0
```

Note _____
For more information about `netstat`, see the `na_netstat(1)` man page.

Disabling fast path

When to disable fast path

You might need to disable fast path on your filer if you have asymmetric routing in your network. For more information, see “[Effect of fast path on asymmetric routing](#)” on page 63.

You can use the `ip.fastpath.enable` option to enable and disable fast path on your filer. The default value for this option is `on`, which enables fast path.

Note

You can use the `-x` option with the `netstat` command to see if the fast path mechanism is enabled for a specific connection.

Disabling fast path

To disable fast path on your filer, complete the following step.

Step	Action
1	Enter the following command: <code>options ip.fastpath.enable off</code>

Modifying the routing table

About modifying the routing table

The `route` command enables you to add or delete a specific route for a host or a network in the routing table.

Cluster considerations

In takeover mode, Data ONTAP maintains the routing table on each filer in the cluster. You can make changes to the routing table of the takeover filer, or you can make changes to the routing table on the filer in partner mode using the `route` command. However, the changes you make in partner mode are lost after a giveback.

Modifying the routing table

To modify the routing table, complete the following step.

Step	Action
1	Enter the following command: <code>route [-fn] add delete [host net] destination [gateway [metric]]</code> Use the following table for explanations about the options and parameters used in the command.

Option/Parameter	Description
-f	Removes all gateway entries in the routing table. If this is used in conjunction with one of the commands, <code>route</code> removes the entries before performing the command.
-n	Prints the IP address (numeric) of a host or a network rather than the host name or network name.
add delete	Adds or deletes the specific route for the given destination in the routing table.

Option/Parameter	Description
<code>host net</code>	Routes to a particular host are distinguished from those to a network by interpreting the IP address associated with the destination. The optional keywords <code>host</code> and <code>net</code> force the destination to be interpreted as a host or a network, respectively. If the <code>host</code> and <code>net</code> keywords are not specified and the destination IP address (specified in the dot decimal notation) has a host part of 0, or if the destination is the symbolic name of a network, the route is assumed to be to a network; otherwise, it is assumed to be to a host.
<i>destination</i>	Specifies the destination for which the route is being added or deleted. The destination can be a specific host or a network. The netmask for a route to a network is implicitly derived from the class of the network. To override this, the destination for a network route can have <i>/bits</i> or <i>&mask</i> after it, where <i>bits</i> is the number of high-order bits to be set in the netmask, and <i>mask</i> is the netmask (either as a number that defaults to decimal, preceded with 0x for hexadecimal or with 0 for octal, or as an IP address in dot decimal notation). All symbolic names specified for a destination are looked up as a host name in the <code>/etc/hosts</code> database. If the lookup fails, the name is looked up as a network name in the <code>/etc/networks</code> database. The keyword <code>default</code> is also a valid destination. The default destination is used if there is no specific host or network route.
<i>gateway</i>	Specifies the next-hop gateway to which packets should be forwarded for the corresponding destination.

Option/Parameter	Description
<i>metric</i>	The <code>metric</code> option indicates the number of hops to the destination. The <code>metric</code> option is required for the <code>add</code> command; it must be zero if the destination is on a directly attached network, and nonzero if the route is through one or more gateways. A <code>metric</code> of zero implies that the route does not refer to a gateway, but rather to one of the machine's interfaces. Destinations matching such a route are sent out on the interface identified by the gateway address.

Diagnosing network problems

About diagnosing network problems

The `netdiag` command automatically and easily diagnoses network problems.

This command analyzes the statistics continuously gathered by Data ONTAP and performs diagnostic tests to identify and report problems related to the physical, network, or transport layers. If any problems are found, the command output also suggests remedial actions.

Syntax of the netdiag command

The `netdiag` command syntax is as follows:

```
netdiag [-s|v|d] [-nbat] [-p [-I interface]]
```

For information about all the options available with the `netdiag` command, see the `na_netdiag(1)` man page.

Diagnosing transport layer problems

To diagnose transport layer problems in your filer, complete the following step.

Step	Action
1	Enter the following command: <pre>netdiag -t</pre> Example: A filer whose TCP window size is smaller than the recommended value displays the following output when the above command is run: <pre>Performing transport layer diagnostics..... The TCP receive window advertised by CIFS client 10.10.10.10 is 8760. This is less than the recommended value of 32768 bytes. You should increase the TCP receive buffer size for CIFS on the client. Press enter to continue.</pre>

About this chapter

This chapter discusses how you can use the Data ONTAP configuration files, Domain Name System (DNS), and Network Information Service (NIS) to resolve host names. It also describes how to use Lightweight Directory Access Protocol (LDAP) to authenticate multiprotocol file access.

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“Maintenance of host information”](#) on page 74
- ◆ [“Using the /etc/hosts file to maintain host information”](#) on page 75
- ◆ [“Using DNS to maintain host information”](#) on page 78
- ◆ [“Using NIS to maintain host information”](#) on page 83
- ◆ [“Using LDAP to authenticate multiprotocol file access”](#) on page 87
- ◆ [“Changing the host name search order”](#) on page 92

Maintenance of host information

Ways to maintain host information

Host information can be maintained in one or all of the following ways in Data ONTAP:

- ◆ In the `/etc/hosts` file on the filer's default volume. For detailed information, see [“Using the `/etc/hosts` file to maintain host information](#)” on page 75.
- ◆ On a Domain Name System (DNS) server. For detailed information, [“Using DNS to maintain host information](#)” on page 78.
- ◆ On a Network Information Service (NIS) server. For detailed information, see [“Using NIS to maintain host information](#)” on page 83.

Search order for host information

If you use more than one of the above ways to maintain host information, they are used in the order determined by the `/etc/nsswitch.conf` file. For detailed information about this file, see [“Changing the host name search order](#)” on page 92.

Using the /etc/hosts file to maintain host information

About the /etc/hosts file

Data ONTAP uses the /etc/hosts file to resolve host names to IP addresses, including host names used in any of the following files:

- ◆ /etc/rc
- ◆ /etc/syslog.conf
- ◆ /etc/exports
- ◆ /etc/netgroup
- ◆ /etc/hosts.equiv

You must ensure that the /etc/hosts file is kept up-to-date.

When Data ONTAP is first installed, the /etc/hosts file is automatically created. If you update the file, you do not need to reboot the filer—the changes to the file take effect immediately.

Note

The /etc/hosts file only resolves the host names for the filer it is configured on. This file cannot be used by other systems for name resolution.

Ways to add entries to the /etc/hosts file

You can add IP address-hostname entries in the /etc/hosts file in the following two ways:

- ◆ Manually
You might want to add entries manually if the number of entries is small. For more information, see “[Updating the /etc/hosts file manually](#)” on page 76.
- ◆ Using the NIS makefile master
If the number of entries is large and you have access to an NIS makefile master, you might want to use the makefile master to create the /etc/hosts file. This method prevents errors that could be introduced in the manual creation process. For details, see “[Creating /etc/hosts from the NIS master](#)” on page 76.

Updating the /etc/hosts file manually

To update the /etc/hosts file manually, complete the following steps.

Step	Action
1	From a workstation that has access to the filer's root volume, open the /etc/hosts file using a text editor.
2	Edit the file to your needs. The format of the file is as follows: <i>IP address Host-name aliases</i>
3	Save the file.

Example: The following shows how the entries might look in the /etc/hosts file on a filer:

```
192.16.3.145  toaster_black toaster
192.16.4.155  toaster_red
192.16.5.165  toaster_white
192.16.6.175  toaster_blue
```

In the first line the filer host name itself is used as an alias for the first network interface.

Creating /etc/hosts from the NIS master

To modify the makefile for the NIS master to create a hosts file and copy it to the /etc directory on the filer's default volume, complete the following steps.

Step	Action
1	On the NIS server, open the NIS makefile with an editor.
2	Locate the section for hosts.time.

Step	Action
3	Add the following lines at the end of the hosts.time section, replacing <i>toaster 1</i>, <i>toaster2</i>, and so on with names of your filers: <pre>@mntdir=/tmp/netapp_etc_mnt_\$\$\$;\ if [! -d \$mntdir]; then rm -f \$mntdir; \ mkdir \$mntdir; fi;\ for filer in toaster1 toaster2 toaster3 ; do \ mount \$\$filer:/etc \$mntdir;\ mv \$mntdir/hosts \$mntdir/hosts.bak;\ cp /etc/hosts \$mntdir/hosts;\ umount \$mntdir;\ done;\ rmdir \$mntdir</pre>
4	Save the NIS makefile. The /etc/hosts file on the filer is updated whenever the NIS makefile is run.

Using DNS to maintain host information

Advantage of using DNS

DNS enables you to maintain host information centrally. As a result, you do not have to update the `/etc/hosts` file every time you add a new host to the network. If you have several filers on your network, maintaining host information centrally saves you from updating the `/etc/hosts` file on each filer every time you add or delete a host.

About configuring DNS

You can configure the filer to use one or more DNS servers either during the setup procedure or later using the Data ONTAP options at the command line.

If you configure DNS later, you need to

- ◆ Create and edit the `/etc/resolv.conf` file in the filer's root volume.
- ◆ Specify the domain name of the filer using the `dns.domainname` option.
- ◆ Enable DNS on the filer using the `dns.enable` option.

Note

If you want to use DNS as your primary method of host name resolution, specify it ahead of other methods in the hosts map in the `/etc/nsswitch.conf` file. For information about how to edit the `nsswitch.conf` file, see [“Changing the host name search order”](#) on page 92.

Creating or editing /etc/resolv.conf

To create or edit the /etc/resolv.conf file, complete the following step.

Step	Action	
1	If...	Then...
	You are creating the /etc/resolv.conf file	Using a text editor, create the /etc/resolv.conf file in the root volume. The file can consist of up to three lines, each specifying a name server host in the following format: <code>nameserver ip_address</code> Example: <code>nameserver 192.9.200.10</code> <code>nameserver 192.9.200.20</code> <code>nameserver 192.9.200.30</code>
	You are editing the /etc/resolv.conf file	From a workstation that has access to the filer's root volume, open and edit the /etc/resolv.conf file using a text editor.
		Edit the files to your needs.
		Save the file.

Specifying the DNS domain name

To specify the DNS domain name, complete the following step.

Note

You can also use this procedure to change the domain name of the filer.

Step	Action
1	Enter the following command: <code>options dns.domainname domain</code> <i>domain</i> is the new domain name; for example, the domain name of a filer <i>toaster.netapp.com</i> is <i>netapp.com</i> .

Disabling or enabling DNS

To disable or enable DNS, complete the following step.

Step	Action
1	Enter the following command: options dns.enable {off on} Use <i>off</i> to disable DNS or <i>on</i> to enable DNS.

DNS is disabled by default.

What DNS name caching does

DNS name caching enables the DNS name resolver to speed up the process by which it converts host names into IP addresses. DNS name caching stores DNS requests by caching them so that they are easy to find the next time. Name caching improves DNS performance in the case of name server failure as well as reducing the time it takes for cluster takeover and giveback.

Disabling or enabling DNS name caching

To disable or enable DNS name caching, complete the following step.

Caution

Disabling DNS name caching clears the DNS name cache.

Step	Action
1	Enter the following command: options dns.cache.enable {on off} Use <i>on</i> to enable DNS name caching or <i>off</i> to disable DNS name caching.

DNS name caching is enabled by default.

Flushing the DNS cache

Entries in the DNS cache have a set expiration. If an entry that has expired is needed again, the filer contacts the DNS server to get an updated entry. However, if a DNS entry changes before it has expired, you must flush the DNS cache to force the filer to get the new DNS record.

If some of your DNS records change often, you should make sure that your DNS server transmits them with a low Time to Live (TTL). (You set this in the DNS server.) You can also disable DNS caching on the filer via the `dns.cache.enable` option, but this may have an adverse performance impact.

To flush the DNS cache, complete the following step.

Step	Action
1	Enter the following command: dns flush

Displaying DNS information

You can display the following types of DNS information:

- ◆ Status of the DNS resolver
- ◆ List of DNS servers configured in the `/etc/resolv.conf`
- ◆ State of each DNS server
- ◆ Timestamp when the DNS server was last polled
- ◆ Average round-trip time of a DNS query
- ◆ Total number of DNS queries made
- ◆ Number of failed DNS queries
- ◆ Default domain configured on the filer
- ◆ List of other domains that will be used with unqualified names for name lookup

To display DNS information, complete the following step.

Step	Action
1	Enter the following command: dns info

Example:

```
>dns info
DNS is enabled
DNS caching is enabled

5 cache hits
8 cache misses
6 cache entries
```

0 expired entries
0 cache replacements

IP Address	State	Last Polled	Avg RTT	Calls	Errs
172.19.2.2	UP	Fri Sep 13 19:15:10 GMT 2002	430	12	0
172.19.2.16	??		0	0	0
172.19.3.11	??		0	0	0

Default domain: lab.netapp.com
Search domains: lab.netapp.com netapp.com

Using NIS to maintain host information

Advantage of using NIS

Like DNS, NIS enables you to centrally maintain host information, so you don't have to update the `/etc/hosts` file on each filer in your network.

NIS also enables you to maintain user information. For more information, see the *Storage Management Guide*.

About configuring NIS

You can configure the filer to use one or more NIS servers either during the setup procedure or later using the Data ONTAP options at the command line.

If you configure NIS later, you need to

- ◆ Specify the NIS server to which the filer should bind.
- ◆ Specify the domain name of the filer using the `nis.domainname` option.
- ◆ Enable NIS on the filer using the `nis.enable` option.

Note

If you want to use NIS as your primary method of host name resolution, specify it ahead of other methods in the hosts map in the `/etc/nsswitch.conf` file. For information about how to edit the `nsswitch.conf` file, see [“Changing the host name search order”](#) on page 92.

Specifying NIS servers to bind to

You can specify an ordered list of NIS servers to which you want the filer to bind. The list should begin with the closest NIS server and end with the furthest one.

To specify an ordered list of NIS servers you want the filer to bind to, complete the following step.

Note

You can specify NIS servers by IP address or host name. If host names are used, make sure each host name along with its IP address is listed in the `/etc/hosts` file. Otherwise, the binding with host name will fail.

Step	Action
1	Enter the following command to specify the NIS servers and their order: <pre>options nis.servers ip_address,server_name,*</pre> The asterisk (*) specifies that broadcast is used to bind to NIS servers if the servers in the list are not responding. This is the default. If you do not specify broadcasting (that is, no asterisk) and none of the listed servers are responding, NIS services are disrupted until one of the preferred servers responds. Note Do not use the broadcast feature if you want secured NIS services.

Example of specifying NIS servers to bind to: The following lists two servers and uses the broadcast default:

```
options nis.servers 172.15.16.1,nisserver-1,*
```

The filer first tries to bind to 172.15.16.1. If the binding fails, the filer tries to bind to nisserver-1. If this binding fails also, the filer binds to any server that responds to the broadcast. While bound to the NIS server that responded to the broadcast, the filer continues to poll the preferred servers. As soon as one of the preferred servers is found, the filer binds to the preferred server.

Specifying the NIS domain name

To specify the NIS domain name, complete the following step.

Step	Action
1	Enter the following command: options nis.domainname domain <i>domain</i> is the domain name of the filer; for example, the domain name of a filer <i>toaster.netapp.com</i> is <i>netapp.com</i> .

Enabling or disabling NIS

To enable or disable NIS on your filer, complete the following step.

Step	Action
1	Enter the following command: options nis.enable {on off} Use <i>on</i> to enable and <i>off</i> to disable NIS.

Displaying NIS information

You can display the following types of NIS information:

- ◆ NIS domain name
- ◆ Last time the local group cache was updated
- ◆ The following information about each NIS server that was polled by the filer:
 - ❖ IP address of the NIS server
 - ❖ Type of NIS server
 - ❖ State of the NIS server
 - ❖ Whether the filer is bound to the NIS server
 - ❖ Time of polling

To display NIS information, complete the following step.

Step	Action
1	Enter the following command: nis info

Displaying the NIS server name

To display the NIS server name to which the filer binds to currently, complete the following step.

Step	Action
1	Enter the following command: <code>ypwhich</code>

Using LDAP to authenticate multiprotocol file access

About using LDAP

LDAP enables multiprotocol users to be authenticated for accessing UNIX files on the filer. If you store your UNIX user database on an LDAP server, you can configure your filer to look up user information in the LDAP database rather than using the `/etc/passwd` file or NIS.

About configuring LDAP

You can configure the filer to use one or more LDAP servers only by using Data ONTAP options at the command line.

To configure LDAP, you need to do the following:

- ◆ Specify the LDAP base distinguished name using the `ldap.base` option.
- ◆ Specify the LDAP servers to use for LDAP queries using the `ldap.servers` option.
- ◆ Specify the preferred LDAP servers, if any, using the `ldap.servers.preferred` option.
- ◆ Enable LDAP on the filer using the `ldap.enable` option.
- ◆ Edit the password line of the `/etc/nsswitch.conf` file for LDAP.

In addition, you might need to do the following:

- ◆ If anonymous authentication does not work in your environment, specify the administrative user name to use for queries to look up user IDs (UIDs) and group IDs (GIDs) using the `ldap.name` option.
- ◆ If there is a password for the administrative user, specify the administrative password using the `ldap.passwd` option.
- ◆ If your LDAP server has been set up to use a port for LDAP queries other than the default port 389, specify the LDAP port using the `ldap.port` option.
- ◆ If root does not appear in the `/etc/passwd` file, make an entry in the LDAP database for root.

Types of LDAP servers supported

Data ONTAP supports the following types of LDAP servers:

- ◆ Netscape®
- ◆ iPlanet
- ◆ OpenLDAP

Enabling RFC 2307 RFC 2307 must be enabled on the LDAP servers to be used for LDAP queries. Netscape and iPlanet servers have RFC 2307 enabled by default.

If you are using an OpenLDAP server, you must enable RFC 2307 by including the following line in the slapd.conf file:

```
include .servers/slapd/schema/nis.schema
```

Specifying the LDAP base distinguished name

The LDAP base distinguished name is the name of the LDAP tree in which user information is stored.

To specify the LDAP base distinguished name to use to look up user names, complete the following step.

Step	Action
1	Enter the following command: options ldap.base <i>name</i> <i>name</i> is the base distinguished name. Use quotes around names with embedded spaces. Example: options ldap.base "o=network appliance, c=us"

Specifying LDAP servers

To specify the LDAP servers to be used for LDAP queries, complete the following step.

Step	Action
1	Enter the following command: options ldap.servers "<i>name</i> [<i>name</i>...]" <i>name</i> is the name of an LDAP server. You can enter multiple server names using a space-separated list enclosed in quotes. Example: options ldap.servers "server1 server2"

Specifying preferred servers

You might want to specify LDAP servers that are on faster links as the preferred servers, if any of your LDAP servers are on a WAN link or are considered slower or less reliable.

To specify the LDAP servers that you want the filer to check first, complete the following step.

Step	Action
1	Enter the following command: <pre>options ldap.servers.preferred "name [name...]"</pre> <i>name</i> is the name of a preferred LDAP server. You can enter multiple server names using a space-separated list enclosed in quotes. Example: <pre>options ldap.servers.preferred "server1 server2"</pre>

Enabling or disabling LDAP

To enable or disable LDAP on your filer, complete the following step.

Step	Action
1	Enter the following command: <pre>options ldap.enable {on off}</pre> Use <code>on</code> to enable and <code>off</code> to disable LDAP.

Editing the /etc/nsswitch.conf file for LDAP

To edit the `/etc/nsswitch.conf` file for LDAP, complete the following steps.

Step	Action
1	Open the <code>/etc/nsswitch.conf</code> file on the filer for editing. Note_____ If the <code>/etc/nsswitch.conf</code> file does not exist, see “Changing the host name search order” on page 92 for instructions on how to create the file. _____

Step	Action
2	Enter the following at the beginning of the password line: ldap Note— You can optionally add <code>files</code> and <code>nis</code> to the password line, but they must be entered after <code>ldap</code>. Example: <code>passwd: ldap files nis</code>
3	Save the file.

Specifying the admin user name

If anonymous authentication does not work in your environment, you need to specify an administrative user name.

To specify the user name to be used for administrative queries to look up UIDs and GIDs, complete the following step.

Step	Action
1	Enter the following command: options ldap.name <i>name</i> <i>name</i> is the name to be used for administrative queries. The best practice is for this to be the name of a user with read-only access to the LDAP database. Use quotes around names with embedded spaces. Example: <code>options ldap.name "cn=root, o=network appliance, c=us"</code>

Specifying the admin password

To specify a password for the administrative user, complete the following step.

Step	Action
1	Enter the following command: options ldap.passwd <i>password</i> <i>password</i> is the password for the administrative user. The password is displayed as a series of asterisks (***** for example).

Specifying the LDAP port

You might need to specify the port to use for LDAP queries, if the LDAP server has been set up to use a port other than the default for LDAP, port 389.

To specify the LDAP port, complete the following step.

Step	Action
1	Enter the following command: options ldap.port <i>N</i> <i>N</i> is the LDAP port number.

Changing the host name search order

How the host name search order is determined

If you use more than one method for host name resolution, you must specify the order in which each name resolution service is used. This order is specified in the `/etc/nsswitch.conf` file in the filer's root volume.

The default `/etc/nsswitch.conf` file

If the `/etc/nsswitch.conf` file does not exist in the filer's root volume, Data ONTAP creates a default `nsswitch.conf` file when NIS is enabled on your filer. The contents of the default file are as follows:

```
hosts: files nis dns
passwd: files nis
netgroup: files nis
group: files nis
shadow: files nis
```

Note
Only the `hosts` entry in the `/etc/nsswitch.conf` file pertains to host name resolution. For information about other entries, see the *Storage Management Guide*.

By default, the host information is searched in the following order:

- ◆ `/etc/hosts` file
- ◆ NIS
- ◆ DNS

Changing the search order

To change the order in which Data ONTAP searches for host information, complete the following steps.

Step	Action
1	If an <code>/etc/nsswitch.conf</code> file does not exist in the filer's root volume, create it.

Step	Action
2	Edit the file and enter each line in the following format: <i>map: service ...</i> <i>map</i> for the host name resolution service is <i>hosts</i> . <i>service</i> is one or more of the following: <i>files</i> , <i>dns</i> , <i>nis</i> .
3	Save the file.

About this chapter

This chapter describes how Data ONTAP supports SNMP on the filer and how you can use SNMP to monitor your filer.

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“Understanding SNMP implementation in Data ONTAP”](#) on page 96
- ◆ [“Contents of the Network Appliance custom MIB”](#) on page 97
- ◆ [“SNMP options and commands supported by Data ONTAP”](#) on page 100
- ◆ [“Understanding traps in Data ONTAP”](#) on page 104
- ◆ [“Defining or modifying a trap”](#) on page 105
- ◆ [“SNMP parameters”](#) on page 107

Understanding SNMP implementation in Data ONTAP

SNMP process

If SNMP is enabled in Data ONTAP, SNMP managers might query the SNMP agent for information (specified in the Network Appliance custom MIB or the MIB-II specification) from time to time. In response, the SNMP agent gathers information and forwards it to the SNMP managers using the SNMP protocol. The SNMP agent also generates traps whenever specific events occur and sends these traps to the SNMP managers. The SNMP managers might then carry out actions based on information received in the trap notifications.

SNMP agent and MIB groups supported

For diagnostic and other network management services, Data ONTAP supports an SNMP version 1 compatible agent. This agent provides support for the MIB-II specification and a Network Appliance custom MIB. The following MIB-II groups are supported:

- ◆ System
- ◆ Interfaces
- ◆ Address translation
- ◆ IP
- ◆ ICMP
- ◆ TCP
- ◆ UDP
- ◆ SNMP

Note

Transmission and EGP MIB-II groups are not supported.

The Network Appliance custom MIB is discussed in “[Contents of the Network Appliance custom MIB](#)” on page 97.

SNMP limitations in Data ONTAP

Data ONTAP imposes the following SNMP limitations:

- ◆ Up to eight communities are allowed.
- ◆ Trap notifications are sent to a maximum of eight network management stations.

Contents of the Network Appliance custom MIB

About the Network Appliance custom MIB

The Network Appliance custom MIB provides detailed information about many aspects of filer operation. You can get the Network Appliance custom MIB from the following sources:

- ◆ After you have installed the Data ONTAP software, in the `/etc/mib/netapp.mib` file
- ◆ The NOW Web page at <http://now.netapp.com>

A short cross-reference of OIDs and short names is available in the `/etc/mib/traps.dat` file. You can find the latest `netapp.mib` and `traps.dat` files online on the NOW site at <http://mktg-web.netapp.com/products/software/systemmanagement/MIB/SNMPinfo.htm>

The Network Appliance custom MIB was verified using `smilint` from the `libsmi` tool version 0.4.0.

Network Appliance custom MIB groups

The top-level groups in the Network Appliance custom MIB that are relevant to the filer are described in the following table.

Group name	Contents
product	Product information, such as the software version and system ID.
sysStat	System-level statistics, such as CPU uptime and idle time, total number of kilobytes transmitted and received on all network interfaces, cluster takeover status, hardware temperature, and power supply status. Note_____ If the filer is not licensed for cluster setup, a value indicating no cluster license is returned. _____

Group name	Contents
nfs	Statistics such as those displayed by the <code>nfsstat</code> command, including statistics for each client if per-client statistics are enabled. The per-client statistics are indexed by client IP addresses.
quota	Information related to disk quotas, including the output of the <code>quota report</code> command.
filesystem	Information related to the file system, including the equivalent of the <code>maxfiles</code> and <code>df</code> commands, overall mirror status, number of plexes in a file system, and some of the information from the Snapshot™ <code>snap list</code> command.
raid	RAID configuration and plex-specific information, such as plex id, plex status, and parent volume.
cifs	Statistics such as those displayed by the <code>cifs stat</code> command.
snapmirror	SnapMirror™ statistics, such as status, number of active backups and restores, and number of bytes read and written by SnapMirror.
ndmp	Information related to NDMP sessions, including the equivalent of the <code>ndmpd status</code> command, such as currently active sessions and number of backups and restores that succeeded.
fabric	Information about the SAN fabric.
dafs	Information about DAFS.
vi	Information about vi status.
backup	Information about dump and restore activities.
vfiler	Information about vfilers.
blocks	Information about block transfer activities.
nfscache	Information about nfscache.

Group name	Contents
snapvault	SnapVault™ statistics, such as status, number of active backups and restores, and number of bytes read and written by SnapVault.

SNMP options and commands supported by Data ONTAP

SNMP options supported by Data ONTAP

Data ONTAP supports the following SNMP options:

- ◆ `snmp.enable`—Enables (with value `on`) or disables (with value `off`) SNMP in Data ONTAP.
For more information about this option, see the `na_options(1)` man page.
- ◆ `snmp.access`—Restricts SNMP access to a filer.
You can restrict SNMP access on a host or interface basis. For details about using this option, see the `na_protocolaccess(8)` man page.

Types of SNMP commands supported by Data ONTAP

Data ONTAP supports the following types of SNMP commands:

- ◆ Configuration—Configuration commands set or change values of SNMP variables.
- ◆ Trap—Trap commands define or change the values of user-defined traps.

Guidelines for using SNMP commands

The following guidelines apply to all SNMP commands in Data ONTAP:

- ◆ Commands entered at the command line are persistent across reboots.
- ◆ If you specify one or more variables for an option of the SNMP commands, the value of that option is set or changed. However, if no variables are specified, the current value of that option variable is returned.

Configuration commands

The following table lists all the SNMP configuration commands available in Data ONTAP.

Command	Description
<code>snmp</code>	Displays the current values of all SNMP options, such as <code>init</code> , <code>community</code> , <code>contact</code> , and <code>traphost</code> .

Command	Description
<code>snmp community [add ro <i>community</i>]</code>	With the option: Adds a community with the specified access control type. Without the option: Displays the current list of communities. Default value: The default community for the SNMP agent in Data ONTAP is public and the default access mode is ro.
<code>snmp community [delete {all ro <i>community</i>}]</code>	With the option: Deletes one or all communities with the specified access control type. Without the option: Displays the current list of communities.
<code>snmp contact [<i>contact</i>]</code>	With the option: Sets the contact name for the filer. You must enclose the <i>contact</i> string in single quotes (' ') if the string contains spaces. Without the option: Displays the current contact name set in Data ONTAP.
<code>snmp location [<i>location</i>]</code>	With the option: Sets the location of the filer. You must enclose the <i>location</i> string in single quotes (' ') if the string contains spaces. Without the option: Displays the current location set in Data ONTAP.

Trap commands

The following table lists all the SNMP trap commands available in Data ONTAP.

Command	Description
<code>snmp authtrap [0 1]</code>	With the option: Enables (with value 1) or disables (with value 0) SNMP agent authentication failure traps. These traps are generated when the agent receives queries with the wrong community string. The traps are sent to all hosts specified with the <code>snmp traphost</code> command. Without the option: Displays the current value of <code>authtrap</code> set in Data ONTAP.
<code>snmp traphost [{add delete} {hostname ipaddress}]</code>	With the option: Adds or deletes SNMP hosts that receive traps from Data ONTAP. Without the option: Displays the current traphosts set in Data ONTAP.
<code>snmp traps [enable disable reset delete] [trapname]</code>	With the option: Enables, disables, resets, or deletes the <i>trapname</i> trap. If you do not specify the <i>trapname</i> trap, all traps defined so far are acted on. Traps are disabled by default. You must enable a trap before it can be triggered with the <code>snmp traps</code> command. Without the option: Displays the list of all traps set in Data ONTAP.
<code>snmp traps walk prefix</code>	Lists all traps that have names beginning with <i>prefix</i> .

Command	Description
<code>snmp traps load <i>filename</i></code>	Loads a set of traps from a text file. The <i>filename</i> file contains a list of traps without the <code>snmp traps</code> command preceding each trap. If the specified filename is defaults, traps are read from the <code>/etc/defaults/traps</code> file.
<code>snmp traps <i>trapname.parm</i> <i>value</i></code>	Defines or changes a user-defined trap.
<code>snmp init [0 1]</code>	With the option: Enables (with value 1) or disables (with value 0) built-in traps and the traps defined using the <code>snmp traps</code> command. A coldStart trap is sent by the agent when the traps are enabled. Without the option: Displays the current value of <code>snmp init</code> in Data ONTAP. Default value: By default, SNMP traps are disabled in Data ONTAP; the system runs as the equivalent of <code>snmp init 0</code>.

Example of typical SNMP commands

The following example shows a typical set of SNMP commands.

```
snmp contact 'jdoe@abc.com 415-555-1212'
snmp location 'ABC corporation, engineering lab'
snmp community add ro private
snmp traphost add snmp-mgr1
snmp init 1
```

Understanding traps in Data ONTAP

About traps

You set traps to inspect the value of MIB variables periodically. Whenever the value of a MIB variable meets the conditions you specify, a trap is sent to the network management stations on the traphost list. The traphost list specifies the network management stations that receive the trap information.

You can set traps on any numeric variable in the MIB. For example, you can set a trap to monitor the fans on a filer and have the SNMP application on your network management station show a flashing message on your console when a fan has ceased operation.

Traps are persistent. After you set a trap, it exists across reboots until you remove it or modify it.

Types of traps in Data ONTAP

There are two types of traps in Data ONTAP:

- ◆ Built-in—Built-in traps are predefined in Data ONTAP and are automatically sent to the network management stations on the traphost list if an event occurs. These traps are based on
 - ❖ RFC 1213, such as coldStart, linkDown, linkUp, and authenticationFailure
 - ❖ Specific traps defined in the Network Appliance custom MIB, such as diskFailedShutdown, cpuTooBusy, and volumeNearlyFull
- ◆ User-defined—User-defined traps exist only after they are defined by a series of `snmp traps` commands.

Where to get further information

See the following RFCs for more information:

- ◆ RFC 1157—Defines and describes SNMP.
- ◆ RFC 1213—Defines and describes the SNMP MIB-II specification.
- ◆ RFC 1155—Defines and describes the structure and identification of management information for TCP/IP-based internets.
- ◆ RFC 1215—Defines the convention for defining traps for use with SNMP.
- ◆ RFC 1212—Gives concise MIB definitions.

Defining or modifying a trap

Defining or modifying a trap

To define or modify a trap, complete the following step.

Step	Action
1	Enter the following command: <pre>snmp traps trapname.parm value</pre> <i>trapname</i> is the name of the user-defined trap you want to create or change. <i>parm</i> is one of the following parameters. For more information about these parameters, see “SNMP parameters” on page 107. ◆ var ◆ trap-id ◆ trigger ◆ edge-1 and edge-2 ◆ edge-1-direction and edge-2-direction ◆ interval ◆ interval-offset ◆ rate-interval ◆ backoff-calculator ◆ backoff-step ◆ backoff-multiplier ◆ priority ◆ message <i>value</i> is the value you assign to <i>trapname.parm</i>.

Example of trap definitions

The following example sets a cifstotalops trap [1] that is evaluated every 10 seconds [2]. The value received from the previous evaluation and the current value are used to calculate the number of CIFS operations per hour [3]. If the number exceeds one million [4], the trap fires and continues to fire every hour [5] until the total number of CIFS operations drops below one million.

```
snmp traps cifstotalops.var snmp.1.3.6.1.4.1.789.1.7.3.1.1.1.0 [1]
snmp traps cifstotalops.trigger level-trigger
```

```
snmp traps cifstotalops.edge-1 1000000 [4]
snmp traps cifstotalops.interval 10 [2]
snmp traps cifstotalops.backoff-calculator step-backoff [5]
snmp traps cifstotalops.backoff-step 3590 [5]
snmp traps cifstotalops.rate-interval 3600 [3]
snmp traps cifstotalops.priority alert
snmp traps cifstotalops.message snmp.1.3.6.1.4.1.789.1.7.3.1.1.1.0
```

SNMP parameters

SNMP parameters

The `snmp traps` command supports the following parameters:

- ◆ `var`
- ◆ `trap-id`
- ◆ `trigger`
- ◆ `edge-1` and `edge-2`
- ◆ `edge-1-direction` and `edge-2-direction`
- ◆ `interval`
- ◆ `interval-offset`
- ◆ `rate-interval`
- ◆ `backoff-calculator`
- ◆ `backoff-step`
- ◆ `backoff-multiplier`
- ◆ `priority`
- ◆ `message`

The var parameter

The `var` parameter associates a user-defined trap name (specified by the *trapname* variable in the `snmp traps` command) with a specific MIB object. The MIB object is specified in the *value* field of the `snmp traps` command. It must be of the form `snmp.oid`, where *oid* is an Object Identifier (OID).

Note

A short list of the Network Appliance MIB OIDs is available in the `traps.dat` file. The file is located in the `/etc/mib` directory on your filer. For more information about a particular OID, see the MIB.

The trap-id parameter

The `trap-id` parameter tags a trap identifier (`trap-id`) to the trap being sent. Each trap defined in the Network Appliance custom MIB has its own identifier. For each user-defined trap the trap manager uses the `trap-id` associated with the trap's priority.

The trigger parameter

The `trigger` parameter specifies the type of triggers that you can set for a trap. If a trap is triggered, data about the event that caused the trigger is sent to the network management stations. You can specify the following values for the `trigger` parameter in the *value* variable field of the `snmp traps` command:

- ◆ `single-edge-trigger`—Fires a trap and sends data when the value of the trap's MIB variable crosses an edge (a value that you specify) for the first time.
- ◆ `double-edge-trigger`—Fires a trap and sends data when either of two edges is crossed. A double-edge-trigger enables you to set two edges, each with its own direction.
- ◆ `level-trigger`—Fires a trap and sends data whenever the trap's value crosses a specified edge value.
- ◆ `change-trigger`—Keeps track of the last value received from the trap. If the current value differs from the previously received value, the trap is triggered.
- ◆ `always-trigger`—Enables a trap to always trigger at the specified evaluation interval (specified by the `interval` parameter discussed later in this section). For example, a trap can trigger every 24 hours for the agent to send the total number of CIFS operations to an SNMP manager.

The edge-1 and edge-2 parameters

The `edge-1` and `edge-2` parameters of a trap specify the threshold values that are compared during trap evaluation to determine whether to fire a trap and send data.

The `edge-1` parameter specifies the value for the edge in a single-edge-triggered trap or the first edge in a double-edge-triggered trap. The default value for the `edge-1` parameter in the *value* variable field of the `snmp traps` command is `MAXINT`.

The `edge-2` parameter specifies the value for the second edge in a double-edge-triggered trap. The default value for the `edge-2` parameter in the *value* variable field of the `snmp traps` command is 0.

The edge-1-direction and edge-2-direction parameters

The `edge-1-direction` and `edge-2-direction` parameters let you set or change the direction that is used to evaluate a trap. The edge-triggered traps only send data when the edge is crossed in either the up or down direction. The default values for the `edge-1-direction` and the `edge-2-direction` parameters in the *value* variable field of the `snmp traps` command are

- ◆ `edge-1-direction—up`
- ◆ `edge-2-direction—down`

Note

The direction values are printed on the same line as the edge value when you run the `snmp traps` command.

The interval parameter

The `interval` parameter is the time, in seconds, between evaluations of a trap. A trap can only send data as often as it is evaluated, even if the edge values are exceeded sooner. The default value for the `interval` parameter in the *value* variable field of the `snmp traps` command is 3600.

Note

The `interval` value for the Network Appliance predefined traps is 60, or one minute.

The interval-offset parameter

The `interval-offset` parameter is the amount of time, in seconds, until the first trap evaluation. The default value for the `interval-offset` parameter in the *value* variable field of the `snmp traps` command is 0. You can set it to a nonzero value to prevent too many traps from being evaluated at once (at system startup, for example).

The rate-interval parameter

The `rate-interval` parameter specifies the time, in seconds, in which the change in value of a trap's variable (rate of change) is expressed. If the `rate-interval` value is set for a trap, the samples of data obtained at the interval points (set using the `interval` parameter) for a trap variable are used to calculate the rate of change. If the calculated value exceeds the value set for the `edge-1` or `edge-2` parameter, the trap is fired.

For example, to obtain the number of CIFS operations per hour, you specify a `rate-interval` of 3600. If `rate-interval` is set to 0, no sampling at interval points occurs and trap evaluation proceeds as with any other kind of trap. The default value for the `rate-interval` parameter in the *value* variable field of the `snmp traps` command is 0.

The backoff-calculator parameter

The `backoff-calculator` parameter enables you to change the trap evaluation interval for a trap after a trap fires. After a trap fires and sends data, you might not want it to be evaluated so often. For instance, you might want to know within

a minute of when a file system is full, but only want to be notified every hour thereafter that it is still full. The `backoff-calculator` parameter can take the following values in the *value* variable field:

- ◆ `step-backoff`
- ◆ `exponential-backoff`
- ◆ `no-backoff`

The default value for the `backoff-calculator` parameter in the *value* variable field of the `snmp traps` command is `no-backoff`.

The backoff-step parameter

The `backoff-step` parameter specifies the number of seconds by which the trap evaluation interval is increased. If a trap interval is 10 and its backoff-step is 3590, the trap is evaluated every 10 seconds until it fires the first time and sends data, and once an hour thereafter. The default value for the `backoff-step` parameter in the *value* variable field of the `snmp traps` command is 0.

The backoff-multiplier parameter

The `backoff-multiplier` parameter specifies the value by which to multiply a trap's evaluation interval each time it fires. If you set `backoff-calculator` to `exponential-backoff` and `backoff-multiplier` to 2, the interval doubles each time the trap fires. The default value for the `backoff-multiplier` parameter in the *value* variable field of the `snmp traps` command is 1.

The priority parameter

The `priority` parameter sets the priority of a trap. If several traps are scheduled to fire at the same time, you can use the `priority` parameter to decide which trap is serviced first. The possible values for the `priority` parameter in the *value* variable field of the `snmp traps` command, from highest to lowest, are as follows:

- ◆ `emergency`
- ◆ `alert`
- ◆ `critical`
- ◆ `error`
- ◆ `warning`
- ◆ `notification`
- ◆ `informational`
- ◆ `debug`

The default value for the `priority` parameter in the *value* variable field of the `snmp traps` command is `notification`.

The message parameter

The `message` parameter specifies a message that goes out with a trap. The message can be a string of text or of the form `snmp.oid`. If an OID is specified, the result of evaluating that OID is sent. If the `message` parameter is not specified, the default value sent when a trap fires is a string that shows the OID value that triggered the trap. For example, the following string is sent to the network management stations for the trap `cpuUpTime` if the `message` parameter is not set:

```
cpuUpTime == 10562288.priority == notification
```

Note

If the message is a string that includes spaces, you must enclose the string in quotation marks (“ ”).

About this chapter This chapter discusses vifs (a feature that implements link aggregation on the filer) and how you can manage various types of vifs on the filer.

Topics in this chapter This chapter discusses the following topics:

- ◆ “[Understanding vifs](#)” on page 114
- ◆ “[Types of vifs](#)” on page 116
- ◆ “[Vif management](#)” on page 119
- ◆ “[Second-level vifs](#)” on page 134

Understanding vifs

About vifs

Vifs are a feature in Data ONTAP that implements link aggregation on a filer. This feature provides a mechanism to group together multiple network interfaces (links) into one logical interface (aggregate). After being created, a vif is indistinguishable from a physical network interface.

Vifs are also referred to by the following terms by different vendors:

- ◆ Virtual aggregations
- ◆ Link aggregations
- ◆ Trunks
- ◆ EtherChannel

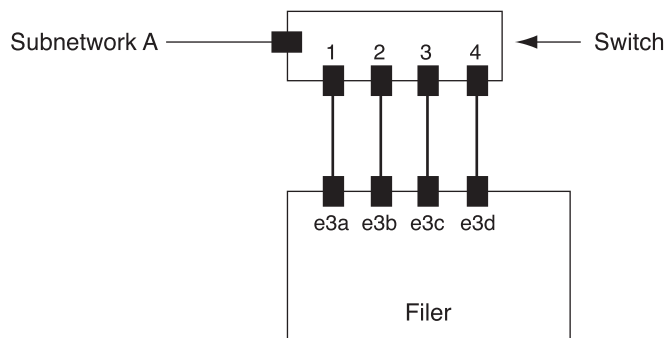
Advantages of vifs

Vifs provide several advantages over individual network interfaces, such as

- ◆ Higher throughput because multiple interfaces work as one interface.
- ◆ Fault tolerance; if one interface in a vif goes down, the filer can stay connected to the network using the other interfaces.
- ◆ Protection against a switch becoming a single point of failure.

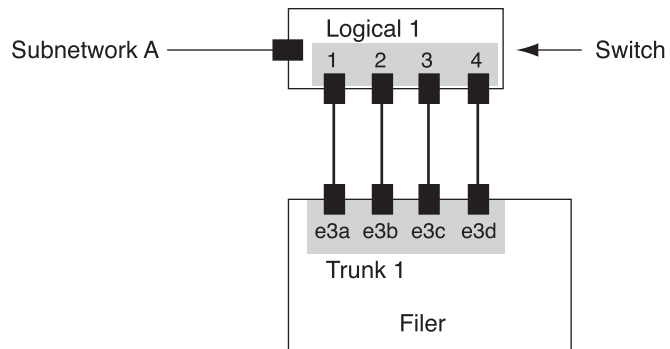
Filer interfaces before grouping into a vif

The following figure shows four separate filer interfaces, e3a, e3b, e3c, and e3d, before grouping into a vif.



Filer interfaces after grouping into a vif

The following figure shows the four filer interfaces grouped into a single vif called Trunk1.



Types of vifs

Kinds of vifs on a filer

There are two kinds of vifs:

- ◆ Single-mode
- ◆ Multi-mode

Single-mode vif operation

In a single-mode vif, only one of the interfaces in the vif is active. The other interfaces are on standby, ready to take over if the active interface fails. Failure means that the link status of the interface is down, which signals that the interface has lost connection with the switch.

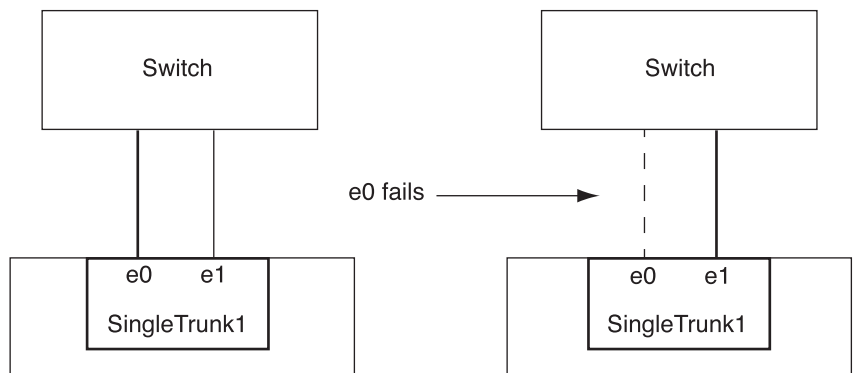
There can be more than one interface on standby in a single-mode vif. If an active interface fails, the filer randomly picks one of the standby interfaces to be the next active link.

All interfaces in a single-mode vif share a common Media Access Control (MAC) address.

Note

A single-mode vif does not require any additional switch configuration or a switch that supports link aggregation.

Example: In the following figure, e0 and e1 are part of the SingleTrunk1 single-mode vif. The active interface, e0, fails. The standby e1 interface takes over and maintains the connection to the switch.



Multi-mode vif operation

In a multi-mode vif, all interfaces in the vif are active and share a common MAC address. This logical aggregation of interfaces provides higher throughput than a single-mode vif. Multi-mode vifs can recover from a failure of up to $(n-1)$ interfaces, where n is the total number of interfaces that form the vif.

Note

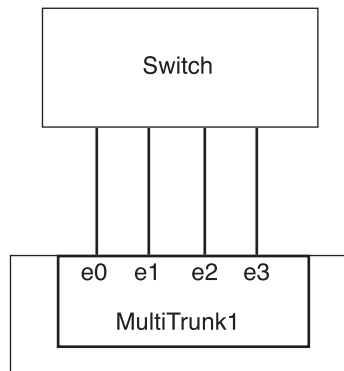
The multi-mode vif implementation in Data ONTAP is in compliance with IEEE 802.3ad (static).

A multi-mode vif requires a switch that supports link aggregation over multiple switch ports. The switch is configured so that all ports to which links of a vif are connected are part of a single logical port. For information about configuring the switch, see your switch vendor's documentation.

Some switches might not support link aggregation of ports configured for jumbo frames. For more information, see your switch vendor's documentation.

Several load-balancing options are available to distribute traffic among the interfaces of a multi-mode vif. The load-balancing schemes are discussed in detail in [“Load balancing in multi-mode vifs”](#) on page 118.

Example: In the following figure, e0, e1, e2, and e3 are part of the MultiTrunk1 multi-mode vif. All four interfaces in the MultiTrunk1 multi-mode vif are active.



If any three of the interfaces fail one by one or simultaneously, the filer still stays connected to the network.

Load balancing in multi-mode vifs

To ensure that all interfaces of a multi-mode vif are equally utilized for outgoing traffic, the following load-balancing methods are available:

- ◆ IP-address based
- ◆ MAC-address based
- ◆ Round robin

The load-balancing method to use for a multi-mode vif can be specified only when the vif is created. If no method is specified, the IP-address based load balancing method is used.

IP-address and MAC-address based: In both of these methods, the last byte of the source and destination address (IP address and MAC address) is used to determine the interface to use for the outgoing frame. The following formula is used:

((source_address XOR destination_address) % number_of_links)

If the result of this formula maps to an interface that is not in the UP link-state, the next active interface is used.

For example, a vif consisting of eight physical interfaces is created with the IP-address based load-balancing method. It is configured with IP address 10.0.0.10. Based on the above formula, an IP frame going through this vif to the destination IP address 172.26.15.224 will use interface #2, provided that this interface is in the UP link-state.

Note

You should not select the MAC-address based load-balancing method when creating vifs on a filer that connects directly to a router. In such a setup, for every outgoing IP frame, the destination MAC address will be the MAC address of the router. As a result, only one interface of the vif will be used.

Round robin: Unlike the IP-address and MAC-address load-balancing methods, this method provides true load balancing. This method may cause out-of-order packet delivery and retransmissions due to overruns.

This method of load balancing is recommended for clients connected in a back-to-back configuration with a filer.

Vif management

About managing vifs

You manage vifs on the filer with the `vif` command. This command enables you to create, add interfaces to, delete interfaces from, display status and statistics of, and destroy a vif.

Guidelines for creating and configuring vifs on a filer

The following guidelines apply to creating and configuring vifs on a filer:

- ◆ You can group up to 16 physical Ethernet interfaces on a filer to obtain a vif. Filer network interfaces that are part of a vif do not have to be on the same network adapter, but Network Appliance recommends that all network interfaces be full-duplex.
- ◆ You cannot include a virtual LAN (VLAN) interface in a vif.
- ◆ The interfaces that form a vif must have the same Maximum Transmission Unit (MTU) size.

You can use the `ifconfig` command to configure the MTU size on the interfaces of a vif. You need to configure the MTU size only if you are enabling jumbo frames on the interfaces. For more information about jumbo frames, see “[Understanding jumbo frames](#)” on page 5.

- ◆ You can include any copper or fiber Gigabit Ethernet Controller numbered II or higher, or any 10Base-T/100Base-TX Ethernet Controller.

Note

Do not mix interfaces of different speeds or media in the same multi-mode vif.

- ◆ Some switches might not support multi-mode link aggregation of ports configured for jumbo frames. For more information, see your switch vendor’s documentation.

The vif command syntax

The `vif` command syntax is as follows:

```
vif create [single|multi] vif_name -b [rr|mac|ip] [interface_list]
vif {favor|nofavor} interface
vif add vif_name interface_list
vif delete vif_name interface
vif destroy vif_name
```

```
vif status [vif_name]
vif stat vif_name [interval]
```

For detailed information about the `vif` command and all the options available with this command, see the `na_vif(1)` man page.

Persistence of the vif command

The following `vif` commands are not persistent if used at the command line; however, you can put any of these commands in the `/etc/rc` file to make them persistent across reboots:

- ◆ `vif create`
- ◆ `vif add`
- ◆ `vif delete`
- ◆ `vif destroy`
- ◆ `vif favor`
- ◆ `vif nofavor`

For detailed information

For detailed information about how to perform specific tasks using the `vif` command, see the following topics:

- ◆ [“Creating a single-mode vif”](#) on page 121
- ◆ [“Changing an active interface in a single-mode vif”](#) on page 123
- ◆ [“Creating a multi-mode vif”](#) on page 124
- ◆ [“Adding interfaces to a vif”](#) on page 126
- ◆ [“Deleting an interface from a vif”](#) on page 127
- ◆ [“Displaying the status of a vif”](#) on page 128
- ◆ [“Displaying statistics of a vif”](#) on page 132
- ◆ [“Destroying a vif”](#) on page 133

Creating a single-mode vif

About creating a single-mode vif

This procedure enables you to create a vif in which only one interface is active at a time and the others are ready to take over in case the active interface fails. If you want a specific interface in a vif to be active, you need to specify that interface as preferred. For more information, see [“Changing an active interface in a single-mode vif”](#) on page 123.

Prerequisites

You need the following information to create a single-mode vif:

- ◆ A case-sensitive name for the vif that meets the following criteria:
 - ❖ It must begin with a letter.
 - ❖ It must not contain any spaces.
 - ❖ It must not contain more than 15 characters.
 - ❖ It must not already be in use for a vif.
- ◆ A list of the interfaces you want to combine into the vif.

Creating a single-mode vif

To create a vif in which only one interface is active at a time, complete the following steps.

Note_____

The operation performed using the `vif create` command is not persistent across reboots unless the command is added to the `/etc/rc` file.

Step	Action
1	Enter the following command: <pre>vif create single vif_name [interface_list]</pre> <i>vif_name</i> is the name of the vif. <i>interface_list</i> is a list of the interfaces you want the vif to consist of. Note_____ You must ensure that all interfaces to be included in the vif are configured down. You can use the <code>ifconfig</code> command to configure an interface down. Example: You can create a single-mode vif with the following command: <pre>vif create single SingleTrunk1 e0 e1</pre>
2	Enter the following command to configure an address on the vif: <pre>ifconfig vif_name address</pre> <i>vif_name</i> is the name of the vif. <i>address</i> is the IP address for this interface. Example: You can configure an IP address of 10.120.5.74 and a netmask of 255.255.255.0 on the single-mode vif <code>SingleTrunk1</code>, created in the previous step, with the following command: <pre>ifconfig SingleTrunk1 10.120.5.74 netmask 255.255.255.0</pre>

Changing an active interface in a single-mode vif

About changing an active interface

When you create a single-mode vif, an interface is selected randomly to be the active interface. However, if you want to specify another interface as active, you can use the `vif favor` command to override the random selection. This new active interface is also known as a preferred interface. There can only be one active interface in a single-mode vif.

Reason for changing an active interface

You might want to change the active interface to another interface if you add a new interface to the vif and want this new interface to be active.

Changing an active interface

To change the active interface in a single-mode vif, complete the following step.

Note _____
The operation performed using the `vif favor` command is not persistent across reboots unless the command is added to the `/etc/rc` file.

Step	Action
1	Enter the following command: <code>vif favor interface</code> <i>interface</i> is the name of the interface you want to be active. Example: You can specify the interface <code>e1</code> to be preferred with the following command: <code>vif favor e1</code>

Creating a multi-mode vif

About creating a multi-mode vif

This procedure enables you to create a multi-mode vif on a filer. By default, the IP-address based load-balancing method is used for a multi-mode vif. However, you can select another method while creating the vif. Once a load-balancing method has been assigned to a vif, it cannot be changed.

Note

You should not select the MAC-address based load-balancing method when creating vifs on a filer that connects directly to a router. In such a setup, for every outgoing IP frame, the destination MAC address will be the MAC address of the router. As a result, only one interface of the vif will be used.

For more information about load-balancing methods available for multi-mode vifs, see [“Load balancing in multi-mode vifs”](#) on page 118.

Prerequisites

You need the following items to create a multi-mode vif:

- ◆ A switch that supports link aggregation over multiple port connections, configured according to your switch vendor’s instructions.
- ◆ A case-sensitive name for the vif that meets the following criteria:
 - ❖ It must begin with a letter.
 - ❖ It must not contain a space.
 - ❖ It must not contain more than 15 characters.
 - ❖ It must not already be in use for a vif.
- ◆ A list of the interfaces you want the vif to consist of.

Creating a multi-mode vif

To create a vif in which all interfaces are active at once, complete the following steps.

Note—The operation performed using the `vif create` command is not persistent across reboots unless the command is added to the `/etc/rc` file.

Step	Action
1	Enter the following command: <pre>vif create multi vif_name -b {rr mac ip} [interface_list]</pre> <code>-b</code> specifies the type of load balancing method. ◆ <code>rr</code>—Round robin ◆ <code>mac</code>—MAC-address based ◆ <code>ip</code>—IP-address based (default) <i>vif_name</i> is the name of the vif. <i>interface_list</i> is a list of the interfaces that make up the vif. Note—You must ensure that all interfaces to be included in the vif are configured down. You can use the <code>ifconfig</code> command to configure an interface down. Example: You can create a multi-mode vif, comprising interfaces <code>e0</code>, <code>e1</code>, <code>e2</code>, and <code>e3</code> and using MAC-based load balancing, with the following command: <pre>vif create multi MultiTrunk1 -b mac e0 e1 e2 e3</pre>
2	Enter the following command to configure an address on the vif: <pre>ifconfig vif_name address</pre> <i>vif_name</i> is the name of the vif. <i>address</i> is the IP address for this interface. Example: You can configure an IP address of <code>10.120.5.75</code> and a netmask of <code>255.255.255.0</code> on the multi-mode vif <code>MultiTrunk1</code>, created in the previous step, with the following command: <pre>ifconfig MultiTrunk1 10.120.5.75 netmask 255.255.255.0</pre>

Adding interfaces to a vif

About adding interfaces

This procedure enables you to add one or more interfaces to a vif. You can add physical interfaces to a vif any time after you create it.

Requirement before adding interfaces

You must configure additional ports on the switch where the new interfaces will be connected. For information about configuring the switch, see your switch vendor’s documentation.

Adding interfaces to a vif

To add one or more interfaces to a vif, complete the following step.

Note

The operation performed using the `vif add` command is not persistent across reboots unless the command is added to the `/etc/rc` file.

Step	Action
1	Enter the following command: vif add vif_name interface_list <i>vif_name</i> is the name of a previously configured vif. <i>interface_list</i> is a list of the interfaces you want to add to the vif. Example: You can add the interface e4 to the multi-mode vif MultiTrunk1 with the following command: vif add MultiTrunk1 e4

Deleting an interface from a vif

About deleting an interface from a vif

This procedure enables you to delete an interface from a vif. The vif must be configured down before you delete its interface.

Deleting interface of a vif

To delete an interface of a vif, complete the following steps.

Note_____

The operation performed using the `vif delete` command is not persistent across reboots unless the command is added to the `/etc/rc` file.

Step	Action
1	Bring the vif down by entering the following command: ifconfig vif_name down <i>vif_name</i> is the name of the vif you want to bring down.
2	Enter the following command: vif delete vif_name interface <i>vif_name</i> is the name of a vif. <i>interface</i> is the interface of the vif you want to delete. Example: You can delete the interface e4 from a multi-mode vif MultiTrunk1 with the following commands: ifconfig MultiTrunk1 down vif delete MultiTrunk1 e4

Displaying the status of a vif

Displaying vif status

You can display the current status of a specified vif or all single-mode and multi-mode vifs on a filer.

To display the status of a vif, complete the following step.

Step	Action
1	Enter the following command: vif status [vif_name] <i>vif_name</i> is the name of the vif whose status you want to display. If you don't specify the vif name, the status of all vifs is displayed.

Example of displaying vif status

The following example displays the status of vif1 on a filer called toaster:

```
toaster> vif status vif1
default: transmit 'IP Load balancing', VIF Type 'multi_mode', fail
'log'
vif1: 1 link, transmit 'none', VIF Type 'single_mode' fail
'default'

VIF Status      Up      Addr_set
up:
e10: state up, since 05Oct2001 17:17:15 (05:23:05)
    mediatype: auto-1000t-fd-up
    flags: enabled
    input packets 20, input bytes 1280
    output packets 2, output bytes 84
    up indications 1, broken indications 0
    drops (if) 0, drops (link) 0
    indication: up at boot
           consecutive 3, transitions 1
down:
e5: state down, since 05Oct2001 17:17:03 (05:22:00)
    mediatype: auto-unknown-cfg_down
    flags: disabled
    input packets 0, input bytes 0
```

```

output packets 0, output bytes 0
up indications 0, broken indications 0
drops (if) 0, drops (link) 0
indication: down at boot
consecutive 4, transitions 0

```

The following table describes the console output.

Field Name	Sub-field Name	Description	Value
default		Indicates the default values for fields such as transmit, VIF Type, and fail. These values apply if no values are specified for these fields when a vif is created.	
	transmit	Indicates the default load-balancing method.	IP Load balancing
	VIF Type	Indicates the default vif type.	multi_mode
	fail	Indicates the default location where the errors will be logged.	log (that is, system log)

Field Name	Sub-field Name	Description	Value
vif1		Indicates that the data following this field pertains to the vif vif1.	
	transmit	Indicates the load-balancing method used.	none (load-balancing methods are used for multi-mode vifs only)
	VIF type	Indicates the type of vif1.	single_mode
	fail	Indicates the location where errors will be logged for vif1.	default (system log)
	VIF Status	Indicates the current status of vif1.	Up
	Addr_set	Indicates that a MAC address has been configured for vif1 and all of its interfaces.	
	up	Indicates that the interface following this sub-field is up. In this example, because vif1 is a single-mode vif, e10 is up while e5 is down.	

Field Name	Sub-field Name	Description	Value
	state	Indicates the current link-state of the interface.	up
	since	Indicates the date, time, and number of hours since the interface has been up.	05Oct2001 17:17:15 (05:23:05)
	flags	Indicates that the interface is enabled to send and receive data (enabled).	enabled
	consecutive	Indicates the number of consecutively received Up or Broken indications from the switch and link interaction.	3
	transitions	Indicates the number of indications received that caused a state transition from Up to Broken or Down to Up.	1
	down	Indicates that the interface following this sub-field is down. In this example, e5 is the standby interface for the single-mode vif, vif1.	

For more information about the `vif status` command, see the `na_vif(1)` man page.

Displaying statistics of a vif

Displaying vif statistics

You display statistics dynamically for a specific vif or for all vifs.

To display statistics, complete the following step.

Step	Action
1	Enter the following command: <pre>vif stat [vif_name] [interval]</pre> <i>vif_name</i> is the name of the vif. If you don't specify a vif, the status of all vifs is displayed. <i>interval</i> is the interval, in seconds. The default is one second.

Example of displaying vif statistics

The following example displays output of the `vif stat` command:

```
vif stat vif0

vif (trunk) vif0
  e3a                e3b
Pkts In Pkts Out      Pkts In Pkts Out
8637076 47801540      158    159
1617    9588         0      0
1009    5928         0      0
1269    7506         0      0
1293    7632         0      0
920     5388         0      0
1098    6462         0      0
2212   13176         0      0
1315    7776         0      0
```

The first row of the output shows the total number of packets received and sent until the time the `vif stat` command was run, and the following rows show the total number of packets received and sent per second thereafter.

Destroying a vif

About destroying a vif

You destroy a vif when you no longer need it or when you want to use the interfaces that form the vif for other purposes. After you complete this procedure, the links in the vif act individually rather than as an aggregate.

Destroying a vif

To destroy a vif, complete the following steps.

Note_____

The operation performed using the `vif destroy` command is not persistent across reboots. If you want to destroy a vif permanently, make sure that the `vif create` commands corresponding to this vif do not exist in the `/etc/rc` file.

Step	Action
1	Configure the vif down by entering the following command: <code>ifconfig vif_name down</code> <i>vif_name</i> is the name of the vif you want to bring down.
2	Enter the following command: <code>vif destroy vif_name</code> <i>vif_name</i> is the name of the vif to destroy.

Second-level vifs

About second-level vifs

You group multiple multi-mode vifs to obtain a second layer of vif called the second-level vif.

Second-level vifs enable you to provide a standby multi-mode vif in case the primary multi-mode vif fails. You can use second-level vifs on a single filer or in a cluster.

For detailed information

For detailed information about second-level vifs and how to create them on a single filer and in a cluster, see the following topics:

- ◆ [“Understanding second-level vifs on a single filer”](#) on page 135
- ◆ [“Creating a second-level vif on a single filer”](#) on page 136
- ◆ [“Understanding second-level vifs in a cluster”](#) on page 138
- ◆ [“Creating a second-level vif in a cluster”](#) on page 140

Understanding second-level vifs on a single filer

About second-level vifs on a single filer

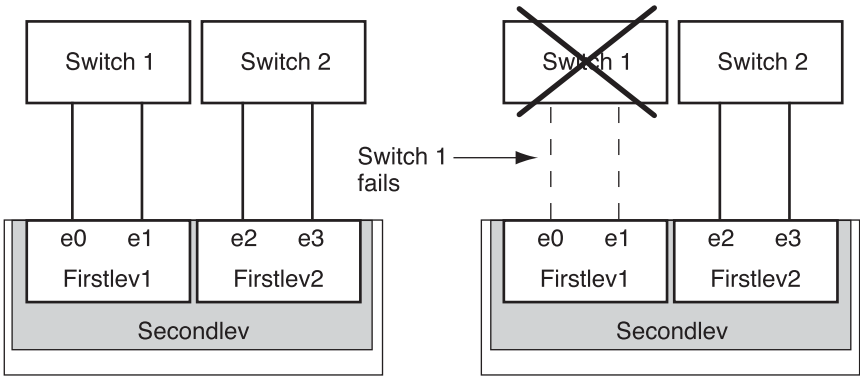
You use a second-level vif on a single filer to provide a standby multi-mode vif in case the primary vif fails. You can provide additional redundancy by using two switches configured for multiple-port connections and four or more interfaces on the filer.

Example of a second-level vif on a single filer

You can set up a filer with two two-link multi-mode vifs. Each vif is connected to a different switch capable of link aggregation over multiple ports. Next, you can set up a second-level single-mode vif that contains both of the multi-mode vifs.

When you configure the second-level vif using the `vif create` command, only one of the two multi-mode vifs is brought up as the active link. If all the underlying interfaces in the active vif fail, the second-level vif activates the link corresponding to the other vif.

In the following illustration, Secondlev is the single-mode second-level vif comprising the Firstlev1 and Firstlev2 vifs. Firstlev1 is initially the active interface; if Switch 1 drops both links, Switch 2 and Firstlev2 take over and maintain the connection to the network.



Creating a second-level vif on a single filer

Assumptions made in this procedure

The following procedure assumes that you want to create a second-level vif, called *vif_name*, on a single filer with two multi-mode vifs, called *vif_name1* and *vif_name2*. The *vif_name1* vif is composed of two physical interfaces, *if1* and *if2*, and *vif_name2* is composed of two physical interfaces, *if3* and *if4*.

By default, IP-based load balancing will be used for the multi-mode vifs created in this procedure.

Prerequisites

You need the following items to create a second-level vif:

- ◆ A switch that supports link aggregation over multiple port connections, configured according to your switch vendor's instructions.
- ◆ A case-sensitive name for each vif that meets the following criteria:
 - ❖ It must begin with a letter.
 - ❖ It must not contain a space.
 - ❖ It must not contain more than 15 characters.
 - ❖ It must not already be in use for a vif.
- ◆ A list of the interfaces you want the vif to consist of.

Creating a second-level vif on a single filer

To create a second-level vif on a single filer, complete the following steps.

Note

The operation performed using the `vif create` command is not persistent across reboots unless the command is added to the `/etc/rc` file.

Step	Action
1	Enter the following commands to create two multi-mode interfaces: <pre>vif create multi -b {rr mac ip} vif_name1 if1 if2 vif create multi -b {rr mac ip} vif_name2 if3 if4</pre> -b specifies the type of load-balancing method. ◆ rr—Round robin ◆ mac—MAC-address based ◆ ip—IP-address based (default) Note_____ You must ensure that all interfaces to be included in the vif are configured down. You can use the <code>ifconfig</code> command to configure an interface down. _____
2	Enter the following command to create a single-mode interface from the multi-mode interfaces: <pre>vif create single vif_name vif_name1 vif_name2</pre>

Example of creating a second-level vif on a single filer

The following commands create the second-level vif shown in [“Example of a second-level vif on a single filer”](#) on page 135. In this example, IP-based load-balancing is used for the multi-mode vifs.

```
vif create multi Firstlev1 e0 e1
vif create multi Firstlev2 e2 e3
vif create single Secondlev Firstlev1 Firstlev2
```

Understanding second-level vifs in a cluster

Advantage of second-level vifs in a cluster

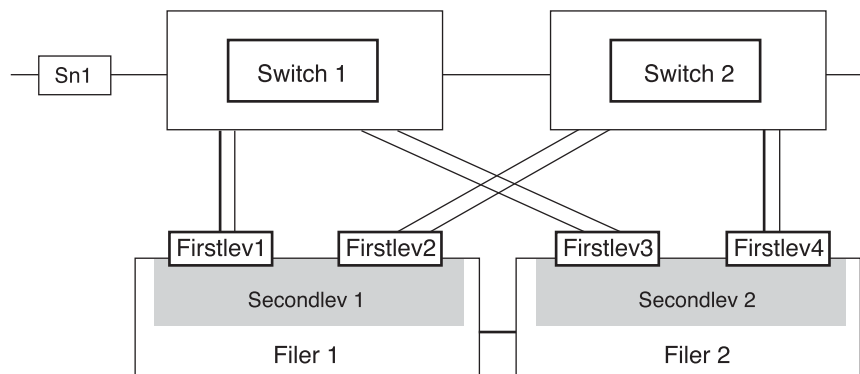
In a cluster configuration, you can access data from both filers even if one of the filers in the cluster fails. In a second-level vif connected in a single-mode configuration, you can maintain connectivity to the filer even if one of the switches fails. Thus, by using the two configurations together, you can achieve a fully redundant filer connectivity architecture.

Normal cluster operation with second-level vifs

The following figure shows second-level vifs in a cluster. When both filers are in operation, the following connections exist:

- ◆ Firstlev1 in Secondlev 1 connects Filer 1 to the network through Switch 1.
- ◆ Firstlev2 in Secondlev 1 connects Filer 1 to Switch 2.
- ◆ Firstlev 4 in Secondlev 2 connects Filer 2 to the network through Switch 2.
- ◆ Firstlev3 in Secondlev 2 connects Filer2 to Switch 1.

Firstlev2 and Firstlev3 are in standby mode.

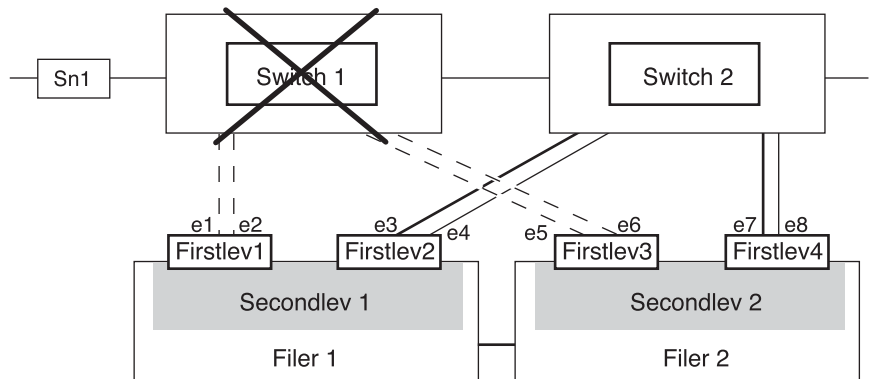


Switch failure in a cluster with second-level vifs

If one of the switches fails, the following happens:

- ◆ If Switch 1 fails, Firstlev2 and Firstlev4 maintain the connection for their filers through Switch 2.
- ◆ If Switch 2 fails, Firstlev1 and Firstlev3 maintain the connection for their filers through Switch 1.

In the following figure, Switch 1 fails in a cluster. Firstlev2 takes over the MAC address of Firstlev1 and maintains the connectivity through Switch 2.



Creating a second-level vif in a cluster

Assumptions made in this procedure

The following procedure assumes that you want to create two second-level vifs, *secondlev1* and *secondlev2*, on clustered filers, Filer 1 and Filer 2. Filer 1 and Filer 2 are configured as shown in the following table.

Filer	Multi-mode vifs	Interfaces
Filer 1	vif_name1	if1 if2
	vif_name2	if3 if4
Filer 2	vif_name3	if5 if6
	vif_name4	if7 if8

Creating a second-level vif in a cluster

To create a second-level vif in a cluster, complete the following steps.

Note—The operation performed using the `vif create` command is not persistent across reboots unless the command is added to the `/etc/rc` file.

Step	Action
1	Enter the following commands on Filer 1 to create two multi-mode vifs: <pre>vif create multi -b {rr mac ip} vif_name1 if1 if2 vif create multi -b {rr mac ip} vif_name2 if3 if4</pre> -b specifies the type of load-balancing method. ◆ rr—Round robin ◆ mac—MAC-address based ◆ ip—IP-address based (default) Note You must ensure that all interfaces to be included in the vif are configured to be down. You can use the <code>ifconfig</code> command to configure an interface.
2	Enter the following command on Filer 1 to create a second-level interface from the multi-mode vifs: <pre>vif create single secondlev1 vif_name1 vif_name2</pre>
3	Enter the following commands on Filer 2 to create two multi-mode vifs: <pre>vif create multi -b {rr mac ip} vif_name3 if5 if6 vif create multi -b {rr mac ip} vif_name4 if7 if8</pre>
4	Enter the following command on Filer 2 to create a second-level interface from the multi-mode vifs: <pre>vif create single secondlev2 vif_name3 vif_name4</pre>
5	Enter the following command on Filer 1 to configure the second-level vifs for takeover: <pre>ifconfig secondlev1 partner secondlev2</pre> Note In this command, <i>secondlev1</i> and <i>secondlev2</i> (arguments to the partner option) should be the interface names and not the interface IP addresses.

Step	Action
6	Enter the following command on Filer 2 to configure the second-level vifs for takeover: <pre>ifconfig secondlev2 partner secondlev1</pre> Note _____ In this command, <i>secondlev1</i> and <i>secondlev2</i> (arguments to the partner option) should be the interface names and not the interface IP addresses.

Example of creating a second-level vif in a cluster

The following commands create the second-level vif in the cluster shown in “[Normal cluster operation with second-level vifs](#)” on page 138. In this example, IP-based load balancing is used for the multi-mode vifs.

On Filer 1:

```
vif create multi Firstlev1 e1 e2
vif create multi Firstlev2 e3 e4
vif create single Secondlev1 Firstlev1 Firstlev2
```

On Filer 2:

```
vif create multi Firstlev3 e5 e6
vif create multi Firstlev4 e7 e8
vif create single Secondlev2 Firstlev3 Firstlev4
```

On Filer 1:

```
ifconfig Secondlev1 partner Secondlev2
```

On Filer 2:

```
ifconfig Secondlev2 partner Secondlev1
```

About this chapter

This chapter discusses the concepts underlying virtual local area networks (VLANs) and VLAN tagging, how VLANs are implemented in Data ONTAP, and how you manage VLANs on a filer.

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“Understanding VLANs”](#) on page 144
- ◆ [“VLANs in Data ONTAP”](#) on page 148
- ◆ [“Managing VLANs on a filer”](#) on page 150

Understanding VLANs

What a VLAN is

A VLAN is a logical network segment that can span multiple physical network segments. The end-stations belonging to a VLAN are related by function or application. For example, end-stations might be grouped by departments, such as engineering and accounting, or by projects, such as release1 and release2. Because physical proximity of the end-stations is not essential in a VLAN, you can disperse the end-stations geographically and still contain the broadcast domain in a switched network.

About VLAN membership

An end-station must become a member of a VLAN before it can share the broadcast domain with other end-stations on that VLAN. The switch ports can be configured to belong to one or more VLANs (static registration), or end-stations can register their VLAN membership dynamically, with VLAN-aware switches.

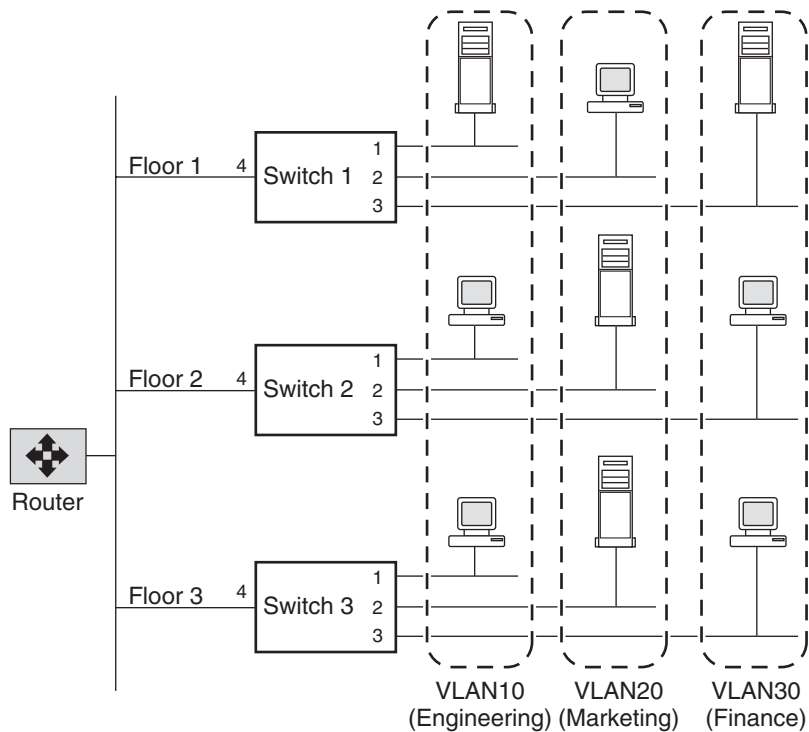
VLAN membership can be based on one of the following:

- ◆ Switch ports
- ◆ End-station MAC addresses
- ◆ Protocol

In Data ONTAP, VLAN membership is *port-based*, or based on switch ports. With port-based VLANs, ports on the same or different switches can be grouped to create a VLAN. As a result, multiple VLANs can exist on a single switch.

How VLAN membership affects communication

Any broadcast or multicast packets originating from a member of a VLAN will be flooded only among the members of that VLAN. Communication *between* VLANs, however, must go through a router. The following figure illustrates how communication occurs between geographically dispersed VLAN members.



In this figure, VLAN 10 (Engineering), VLAN 20 (Marketing), and VLAN 30 (Finance) span three floors of a building. If a member of VLAN 10 on Floor 1 wants to communicate with a member of VLAN 10 on Floor 3, the communication occurs without going through the router, and packet flooding is limited to port 1 of Switch 2 and Switch 3 even if the destination MAC address to Switch 2 and Switch 3 is not known.

What GVRP is

GARP VLAN Registration Protocol (GVRP) uses the Generic Attribute Registration Protocol (GARP) to allow end-stations on a network to *dynamically* register their VLAN membership with GVRP-aware switches. Similarly, these switches dynamically register with other GVRP-aware switches on the network, thus creating a VLAN topology across the network.

Because GVRP provides dynamic registration of VLAN membership, members can be added or removed from a VLAN on the fly, saving the overhead of maintaining static VLAN configuration on switch ports. Additionally, VLAN membership information stays current, limiting the broadcast domain of a VLAN only to the active members of that VLAN.

For more information about GVRP and GARP, see IEEE 802.1Q and IEEE 802.1p (incorporated in 802.1D:1998 edition).

What a VLAN tag is

A VLAN tag is a unique identifier that indicates the VLAN to which a frame belongs. Generally, a VLAN tag is included in the header of every frame sent by an end-station on a VLAN.

How VLAN tagging works

On receiving a tagged frame, the switch inspects the frame header, and based on the VLAN tag, identifies the VLAN. The switch then forwards the frame to the destination in the identified VLAN. If the destination MAC address is unknown, the switch limits flooding of the frame to ports that belong to the identified VLAN.

For example, in the previous figure, if a member of VLAN 10 on Floor 1 sends a frame for a member of VLAN 10 on Floor 2, Switch 1 inspects the frame header for the VLAN tag (to determine the VLAN) and the destination MAC address. Because the destination MAC address is not known to Switch 1, the switch forwards the frame to all other ports that belong to VLAN 10, that is, port 4 of Switch 2 and Switch 3. Similarly, Switch 2 and Switch 3 inspect the frame header. If the destination MAC address on VLAN 10 is known to either switch, that switch forwards the frame to the destination. The end-station on Floor 2 thereby receives the frame.

Advantages of VLANs

VLANs provide the following advantages:

- ◆ **Ease of administration**

VLANs enable logical grouping of end-stations that are physically dispersed on a network. When users on a VLAN move to a new physical location but continue to perform the same job function, the end-stations of those users do not need to be reconfigured. Similarly, if users change their job function, they need not physically move: changing the VLAN membership of the end-stations to that of the new team makes the users' end-stations local to the resources of the new team.

- ◆ **Confinement of broadcast domains**
VLANs reduce the need to have routers deployed on a network to contain broadcast traffic. Flooding of a packet is limited to the switch ports that belong to a VLAN.
- ◆ **Reduction in network traffic**
As a result of confinement of broadcast domains on a network, traffic on the network is significantly reduced.
- ◆ **Enforcement of security policies**
By confining the broadcast domains, end-stations on a VLAN can be isolated from listening to or receiving broadcasts not intended for them. Moreover, if a router is not connected between the VLANs, the end-stations of a VLAN cannot communicate with the end-stations of the other VLANs.

Prerequisites for setting up VLANs

The following requirements must be satisfied before you set up VLANs in a network:

- ◆ The switches deployed in the network either must comply with IEEE 802.1Q standards or must have a vendor-specific implementation of VLANs.
- ◆ For an end-station to support multiple VLANs, it must be able to dynamically register (using GVRP) or must be statically configured to belong to one or more VLANs.

If an end-station cannot register or cannot be configured to belong to a VLAN, the end-station can belong only to one VLAN. This VLAN is configured on the switch port to which the end-station connects. The frames sent on this switch port are untagged.

VLANs in Data ONTAP

GVRP configuration for VLAN interfaces

By default, GVRP is disabled on all VLAN interfaces in Data ONTAP; however, you can enable it.

After you enable GVRP on an interface, the VLAN interface informs the connecting switch about the VLANs it will support. This information (dynamic registration) is updated periodically thereafter. This information is also sent every time an interface comes up after being down or if there is a change in the VLAN configuration of the interface.

Guidelines for setting up VLANs in Data ONTAP

VLANs in Data ONTAP are implemented in compliance with the IEEE 802.1Q standard. Additionally, you must follow the following guidelines while setting up VLANs in Data ONTAP:

- ◆ You cannot set up VLANs using the `setup` procedure. You must use the command line or the FilerView® interface to create, change, or destroy VLANs.
- ◆ You must add the commands to create VLANs on a filer to the `/etc/rc` file to make the VLANs persistent across reboots.
- ◆ You can create any number of VLANs on a NIC (supporting IEEE 802.1Q) on a filer; however, Data ONTAP imposes a limit of 128 interfaces (including physical, vif, vlan, vh, and loopback interfaces) per filer.
- ◆ You can create VLANs on physical interfaces as well as vifs. For more information about vifs, see Chapter 6, “[Vif Configuration](#),” on page 113.
- ◆ You can use VLANs to support packets of different Maximum Transmission Unit (MTU) sizes on the same network interface. If a network interface is a member of multiple VLANs, different MTU sizes can be specified for individual VLANs.
- ◆ You can assign an identification number in the range of 1 to 4,094 to a VLAN.
- ◆ You must ensure that the interface on a filer is also a member of its partner’s VLANs in a cluster failover pair.
- ◆ You cannot configure any parameters except `mediatype` for the physical network interface configured to handle VLANs.

Limitations of setting up VLANs in Data ONTAP

The following are the limitations of setting up VLANs on your filer:

- ◆ The 10Base-T/100Base-T network interface card (NIC) displayed as FastEnet-10/100 Ethernet Controller in the `sysconfig -v` output is the only Fast Ethernet NIC that supports VLANs on a filer.
- ◆ The Gigabit NICs that support VLANs on a filer include all copper or fiber Gigabit Ethernet Controllers numbered II or higher. You can use the `sysconfig` command to determine if you have the appropriate Gigabit NICs installed on your filer.

Reverting to earlier versions of Data ONTAP

Reverting to Data ONTAP 6.1 or 6.1.x: If your filer is a member of a VLAN and you need to revert to Data ONTAP 6.1 or 6.1.x, you must ensure that the `ifconfig` commands in the `/etc/rc` file do not contain the `-g GVRP` flag or the `vlan modify` command.

Reverting to a version earlier than Data ONTAP 6.1: If your filer is a member of a VLAN and you need to revert to a version earlier than Data ONTAP 6.1, you must make sure that the `ifconfig` commands in the `/etc/rc` file do not contain any VLAN configuration information.

Managing VLANs on a filer

Command for managing VLANs on a filer

You manage VLANs on a filer using the `vlan` command. This command allows you to create, add interfaces to, delete, and display statistics of a VLAN.

The `vlan` command syntax

The `vlan` command syntax is as follows:

```
vlan create [-g {on|off}] ifname vlanid_list
vlan add ifname vlanid_list
vlan delete -q ifname [vlanid_list]
vlan modify -g {on|off} ifname
vlan stat ifname [vlanid_list]
```

For detailed information about the `vlan` command, see the `na_vlan(1)` man page.

Persistence of the `vlan` commands

The VLANs created or changed using the `vlan` command are not persistent across reboots unless the `vlan` commands are added to the `/etc/rc` file.

For detailed information

For detailed information on how to perform specific tasks using the `vlan` command, see the following topics:

- ◆ [“Creating and configuring a VLAN on a filer”](#) on page 151
- ◆ [“Adding an interface to a VLAN”](#) on page 154
- ◆ [“Deleting a VLAN”](#) on page 155
- ◆ [“Modifying a VLAN interface”](#) on page 157
- ◆ [“Viewing VLAN statistics”](#) on page 158

Creating and configuring a VLAN on a filer

Commands for creating and configuring a VLAN

Creating and configuring a VLAN involves two commands: the `vlan create` command and the `ifconfig` command.

The `vlan create` command creates a VLAN interface, includes that interface in one or more VLAN groups as specified by the VLAN identifier, enables VLAN tagging, and enables (optionally) GVRP on that interface.

The `ifconfig` command enables you to configure the VLAN interface created by the `vlan` command.

About enabling and disabling GVRP on VLAN interfaces

By default, GVRP is disabled on VLAN interfaces created using the `vlan create` command; however, you can enable it with the `-g` flag available with the command.

If you enable GVRP on an interface that is configured down, the state of the interface and all associated VLAN interfaces is automatically configured up. This state change occurs so that the interface can start sending VLAN registration frames to register its VLAN membership with the switch.

Creating a VLAN on a filer

To create a VLAN on a filer, complete the following step.

Note— You must be familiar with “[Guidelines for setting up VLANs in Data ONTAP](#)” on page 148 before proceeding with the following procedure.

Step	Action
1	Enter the following command: <pre>vlan create [-g {on off}] ifname vlanid</pre> -g enables (on) or disables (off) GVRP on an interface. By default, GVRP is disabled on the interface. <i>ifname</i> is the name of the network interface. <i>vlanid</i> is the VLAN identifier to which the <i>ifname</i> interface belongs. You can include a list of VLAN identifiers. Result: A VLAN interface with the name <i>ifname-vlanid</i> is created. Note— VLANs created using the <code>vlan create</code> command are not persistent across reboots unless the <code>vlan</code> commands are added to the <code>/etc/rc</code> file.

Example of creating a VLAN interface

You can create VLANs with identifiers 10, 20, and 30 on interface e4 of a filer using the following command:

```
vlan create e4 10 20 30
```

As a result, VLAN interfaces e4-10, e4-20, and e4-30 are created. The `ifconfig` command output displays e4 as a VLAN interface as follows:

```
e2: flags=80008042<BROADCAST,RUNNING,MULTICAST, VLAN> mtu 1500
```

Configuring an interface in a VLAN

Using the `ifconfig` command, you can configure all the parameters for a VLAN interface that you can for a physical interface. The parameters you can configure are

- ◆ IP address
- ◆ Network mask
- ◆ Interface status

- ◆ Media type
- ◆ MTU size
- ◆ Flow control
- ◆ Partner

For detailed information about the `ifconfig` command, see Chapter 1, “[Network Interface Configuration](#),” on page 1.

To configure the IP address and network mask for a VLAN interface, complete the following step.

Step	Action
1	Enter the following command: <pre>ifconfig ifname-vlanid address netmask mask</pre> <i>ifname-vlanid</i> is the VLAN interface name. <i>address</i> is the IP address for this interface. <i>mask</i> is the network mask for this interface. Example: You can configure a VLAN interface e4-10, created in the previous example, using the following command: <pre>ifconfig e4-10 172.25.66.11 netmask 255.255.255.0</pre>

Managing VLANs on a filer

Adding an interface to a VLAN

Command for adding an interface to a VLAN

If a physical interface does not belong to any VLAN, you use the `vlan create` command to make the interface a member of one or more VLANs. However, if the interface is already a member of a VLAN, you must use the `vlan add` command to add the interface to subsequent VLANs.

Like the `vlan create` command, the `vlan add` command creates a VLAN interface that must be configured using the `ifconfig` command.

Adding an interface to a VLAN

To add an interface to a VLAN, complete the following step.

Step	Action
1	Enter the following command: <pre>vlan add ifname vlanid</pre> <i>ifname</i> is the name of the network interface. <i>vlanid</i> is the VLAN identifier to which the <i>ifname</i> interface belongs. You can include a list of VLAN identifiers. Result: A VLAN interface with the name <i>ifname-vlanid</i> is created. Note_____ VLANs created using the <code>vlan add</code> commands are not persistent across reboots unless the <code>vlan</code> commands are added to the <code>/etc/rc</code> file.

Example of adding an interface to a VLAN

You can add VLANs with identifiers 40 and 50 on interface e4 of a filer using the following command:

```
vlan add e4 40 50
```

As a result, VLAN interfaces e4-40 and e4-50 are created.

Deleting a VLAN

Command for deleting a VLAN

The `vlan delete` command is used to delete the VLANs on an interface. You can delete either a specific VLAN or all VLANs associated with that interface. If all VLANs for an interface are deleted, the interface is available to be configured as a regular physical interface.

Deleting a VLAN

To delete a VLAN on a filer, complete the following step.

Note—
By default, the `vlan delete` command prompts you to confirm the deletion. If you do not want to receive this prompt, use the `-q` flag. This action invokes *quiet mode*, which causes the operation to complete without prompting.

Step	Action	
1	If you want to delete...	Then...
	All VLANs	Enter the following command: vlan delete [-q] ifname <i>ifname</i> is the name of the network interface.
	Example: You delete all VLANs configured on interface e4 with the following command: vlan delete e4	

Step	Action	
	If you want to delete...	Then...
	A specific VLAN	Enter the following command: <pre>vlan delete [-q] ifname vlanid</pre> <i>ifname</i> is the name of the network interface. <i>vlanid</i> is the VLAN identifier to which the <i>ifname</i> interface belongs. You can include a list of VLAN identifiers.
	Example: You delete VLAN e4-3 with the following command: <pre>vlan delete e4 3</pre>	

Managing VLANs on a filer

Modifying a VLAN interface

Command for
modifying a VLAN

The `vlan modify` command enables or disables GVRP on all the interfaces of a network adapter. That is, you can enable GVRP on network adapter e8 of a filer but not the VLAN interface e8-2. Once you enable GVRP on a network adapter, it is enabled on all associated VLAN interfaces.

Modifying a VLAN
interface

To enable or disable GVRP on VLAN interfaces, complete the following step.

Step	Action
1	Enter the following command: <pre>vlan modify -g {on off} ifname</pre> -g enables (on) or disables (off) GVRP. <i>ifname</i> is the name of the network adapter. Note_____ VLANs modified using the <code>vlan modify</code> command are not persistent across reboots unless the <code>vlan</code> commands are added to the <code>/etc/rc</code> file. _____

Managing VLANs on a filer

Viewing VLAN statistics

Command for displaying VLAN statistics

The `vlan stat` command is used to display the statistics of network interfaces configured in VLANs on a filer. In addition to displaying the frames received and transmitted on an interface, this command displays the number of frames that were rejected because the frames did not belong to any of the VLAN groups to which the interface belongs.

Viewing VLAN statistics

To view VLAN statistics on a filer, complete the following step.

Step	Action	
1	If you want to view...	Then...
	Statistics of all VLANs configured on a network interface	Enter the following command: vlan stat ifname <i>ifname</i> is the name of the network interface.
	Statistics of a specific VLAN configured on a network interface	Enter the following command: vlan stat ifname vlanid <i>ifname</i> is the name of the network interface. <i>vlanid</i> is the VLAN identifier to which the <i>ifname</i> interface belongs. You can include a list of VLAN identifiers.

Example of the vlan stat command

The following example displays the statistics of all VLANs on the filer *toaster*:

```
toaster> vlan stat e4

Vlan Physical Interface: e4 (5 hours, 50 minutes, 38 seconds) --
Vlan IDs: 3,5
GVRP: enabled
RECEIVE STATISTICS
  Total frames:      0 | Total bytes: 0 | Multi/broadcast:    0
  Untag drops:      0 | Vlan tag drops:    0
TRANSMIT STATISTICS
  Total frames:      8 | Total bytes:    368

Vlan Interface: e4-3 (0 hours, 20 minutes, 45 seconds) --
ID: 3
MAC Address: 00:90:27:5c:58:14

RECEIVE STATISTICS
  Total frames:      0 | Total bytes: 0 | Multi/broadcast:    0
TRANSMIT STATISTICS
  Total frames:      0 | Total bytes: 0 | Multi/broadcast:    0
  Queue overflows:    0

Vlan Interface: e4-5 (0 hours, 0 minutes, 7 seconds) --
ID: 5
MAC Address: 00:90:27:5c:58:14

RECEIVE STATISTICS
  Total frames:      0 | Total bytes: 0 | Multi/broadcast:    0
TRANSMIT STATISTICS
  Total frames:      0 | Total bytes: 0 | Multi/broadcast:    0
  Queue overflows:    0
```


About this chapter

This chapter explains how to set up and manage the Internet Protocol Security (IPsec) suite of protocols to secure information within a network. IPsec protects data being transferred between hosts and files by

- ◆ Creating virtual private networks (VPNs) in which only users who have been authenticated can participate
- ◆ Periodically changing encryption and authentication keys to protect data from accidental or deliberate disclosure

Topics in this chapter

This chapter discusses the following topics:

- ◆ [“Understanding IPsec”](#) on page 162
- ◆ [“Setting up IPsec”](#) on page 163
- ◆ [“Managing security policies”](#) on page 165
- ◆ [“Viewing security associations”](#) on page 171

Understanding IPsec

What IPsec is

IPsec is a security protocol that protects data being transmitted between filers and clients from unauthorized disclosure.

IPsec enables authentication and encryption of data in transition between a filer and its CIFS client running Windows 2000 or higher. Using IPsec, you can add policies on the filer that

- ◆ Configure encryption and authentication algorithms between the filer and client. Policies can be configured from the filer to the client and from the client to the filer over a range of IP addresses and ports.
- ◆ Negotiate a security association (SA) between the two end-stations that is used for secure data exchanges between the filer and the client.

IPsec in a cluster configuration

The IPsec protocol, by its nature, does not work well in a failover environment, that is, when one filer in a cluster configuration must takeover the other filer. Reduce the security association lifetime to a minimum value to optimize IPsec operation in a cluster configuration. This minimizes the time clients use to destroy their security associations and negotiate new ones with the filer that took over.

Note

You set the security association lifetime value on clients.

Setting up IPsec

Preparing to use IPsec

Before you can use IPsec, you must do the following:

- ◆ Create a preshared key and store it on the filer and the client that will be sharing the secure connection. The preshared key is an ASCII text string that is used to authenticate the client and filer when IPsec negotiations occur.
- ◆ Enable IPsec functionality on the filer.

Creating the preshared key

To create and store the preshared key on the filer, complete the following steps.

Step	Action
1	Create a psk.txt file in the etc directory.
2	Create an ASCII text key that you will use for authenticating client and filer.
3	In the psk.txt file, add a line using the following format: <i>ip_address</i> tab <i>key</i> <i>ip_address</i> is the IP address of the filer. <i>key</i> is the preshared key you created. Example: 172.25.102.81 ag8key

You will need to entering the preshared key on the client when you configure a policy using the Windows user interface.

Enabling or disabling IPsec

To enable or disable IPsec on a filer, complete the following step.

Step	Action
1	Enter the following command: options ip.ipsec.enable on off on enables IPsec. off disables IPsec.

Managing security policies

What a security policy is

A security policy is a collection of information that determines how security is handled in a transfer of information. A security policy consists of security associations (SAs) that specify the source and destination, the authentication header, the direction, and the security (encryption/decryption) algorithm.

Setting up a security policy

To set up a security policy, complete the following step.

Step	Action
1	Enter the following command: <pre>ipsec policy add -s <i>src_ip</i> -t <i>dest_ip</i> -p {esp ah} [-e {des 3des null}] [-a {sha1 md5}] -d {in out} [-m] [-f <i>ip_protocol</i>] [-l {restrict permit}]</pre> <i>-s src_ip</i> specifies the address of the source of the secure communication as an IP address or IP address range. The address can optionally specify a TCP/UDP port. The address can have one of the following forms: - address - address/prefixlen - address/[port] - address/prefixlen[port] The <i>prefixlen</i> and <i>port</i> specified must be decimal numbers. The square brackets around the port specification are required. <i>-t dest_ip</i> specifies the address of the destination of the secure communication in the same format as the source. <i>-p {esp ah}</i> specifies the security protocol, either encapsulated security payload (ESP), based on RFC 2405, or the authentication header (AH), based on RFC 2402.

Step	Action
	<code>-e {des 3des null}</code> specifies an encryption algorithm— the Data Encryption Standard (<code>des</code>), the triple Data Encryption Standard (<code>3des</code>), or no encryption (<code>null</code>). If you specify <code>-p esp</code> for protocol, it is optional to specify an encryption algorithm. If an algorithm is not specified, the best algorithm is selected based on peer capabilities. <code>-a {sha1 md5 null}</code> specifies an authentication algorithm, either <code>sha1</code> (a 128-bit key) , <code>md5</code> (a 160-bit key), or no algorithm (<code>null</code>). If you specify <code>-p esp</code> for protocol, it is optional to specify an authentication algorithm. If you specify <code>-p ah</code> for protocol, you must specify an authentication algorithm. If you do not specify the authentication algorithm, no authentication takes place. <code>-d {in out}</code> specifies the direction (from the filer) of the data traffic to be affected. <code>-m</code> disables creation of a mirror policy, the same one you are specifying with this command but in the opposite direction. <code>-f</code> specifies an upper-layer protocol using a numeric protocol number, for example, 1 for ICMP or 6 for TCP, from the <code>/etc/protocols</code> file. <code>-l</code> specifies a level of permission, either <code>restrict</code> or <code>permit</code>. The <code>restrict</code> argument enables data traffic only if a valid security association (SA) is obtained. The <code>permit</code> argument causes Data ONTAP to search for an SA. If an SA is present, Data ONTAP uses that SA to send outgoing data. Otherwise, Data ONTAP tries to create a new SA and then sends the current packet of data out unencrypted.

Note

Ensure that policies match on the filer and client that are negotiating the secure connection.

Example: `ipsec policy add -s 10.56.19.172/24[139] -p esp -e des -a ah -d in`

Deleting a security policy

To delete a security policy from a filer, complete the following step.

Step	Action
1	Enter the following command: <pre>ipsecc policy delete all -i index [[-s src_ip -t dst_ip] -d {in out} [-m]]</pre> -i <i>index</i> specifies an index in the security policy database (SPD). -s <i>src_ip</i> specifies the address of the source of the secure communication as an IP address or IP address range. -t <i>dst_ip</i> specifies the address of the destination of the secure communication in the same format as the source. -d {in out} specifies the direction (from the filer) of the data traffic to be affected. -m disables deletion of a mirror policy, the same one you are specifying with this command but in the opposite direction.

Displaying existing security policies

To display security policies, complete the following step.

Step	Action
1	Enter the following command: <code>ipsec policy show [-s <i>src_ip</i>] [-t <i>dst_ip</i>] [-f <i>ip_protocol</i>] -d {in out} [-p {esp ah}]</code> <code>-s <i>src_ip</i></code> specifies the address of the source of the secure communication as an IP address or IP address range. <code>-t <i>dst_ip</i></code> specifies the address of the destination of the secure communication in the same format as the source. <code>-f</code> specifies an upper-layer protocol using a numeric protocol number, for example, <code>-f 6</code> specifies the TCP protocol for the <code>/etc/protocols</code> file. <code>-d {in out}</code> specifies the direction (from the filer) of the data traffic to be affected. <code>-p {esp ah}</code> specifies the security protocol, either encapsulated security payload (ESP), based on RFC 2405, or the authentication header (AH), based on RFC 2402.

Example:

The following example displays security policy information for the device that has a source IP address of 10.56.19.172:

```
ipsec policy show -s 10.56.19.172

Index IPAddress      /prefix/port/protocol Dir/Policy Alg/SecLevel
-----
1      10.56.19.172 / 0/    [any  ]/any -> in /IPSEC  esp/Default
```

How to display IPsec statistics

To display statistics about how IPsec is working, complete the following step.

Step	Action
1	Enter the following command: <pre>ipsec stats [-z]</pre> -z resets the statistics counter. This command displays the following statistics: ◆ Total number of IPsec packets processed inbound and outbound ◆ Total number of AH packets processed inbound and outbound ◆ Total number of ESP packets processed ◆ Total number of AH and ESP processing failures ◆ Total number of AH and ESP replay window failures and successes ◆ Transmit and receive violations, including ❖ Improper or missing policies ❖ Improper or missing SAs ❖ Number of successful and failed Internet Key Exchange (IKE) exchanges

Example:

The following example shows the statistics provided by the `ipsec stats` command.

```
filer1*> ipsec stats
ipsec:
148460138 inbound packets processed successfully
0 inbound packets violated process security policy
983 inbound packets with no SA available
0 invalid inbound packets
0 inbound packets failed due to insufficient memory
0 inbound packets failed getting SPI
0 inbound packets failed on AH replay check
0 inbound packets failed on ESP replay check
143929988 inbound packets considered authentic
0 inbound packets failed on authentication
ESP input packets
    des    : 3886739
    3des   : 140043249
```

```

AH input packets
    md5      : 4530150
134002232 outbound packets processed successfully
0 outbound packets violated process security policy
0 outbound packets with no SP available
11 outbound packets with no SA available
0 invalid outbound packets
0 outbound packets failed due to insufficient memory
0 outbound packets with no route
ESP output packets
    des      : 4571170
    3des     : 124667606
AH output packets
    md5      : 4763456
ike:
IKE input packets
    Identity Protection      : 107
    Informational            : 3682
    Quick                    : 7310
IKE output packets
    Identity Protection      : 108
    Informational            : 10
    Quick                    : 3663

```

Viewing security associations

What security associations are

A security association is a collection of information that determines how security is handled in a transfer of information. Each association lasts for the time assigned by the server.

Displaying security associations

To view the currently active security associations on a filer, complete the following step.

Step	Action
1	Enter the following command: <pre>ipsec sa show [<i>spi</i> [-s <i>src_ip</i> -t <i>dst_ip</i> -p {<i>esp</i> <i>ah</i>}] [-v]</pre> <i>spi</i> specifies the security parameter index. -s <i>src_ip</i> specifies the address of the source of the secure communication as an IP address or IP address range. -t <i>dst_ip</i> specifies the address of the destination of the secure communication in the same format specified for the source. -p specifies the protocol to be used, either <i>esp</i> (encapsulating security payload, based on RFC 2405) or <i>ah</i> (authentication header, based on RFC 2402) -v specifies the verbose (full) output.

Example:

The following example displays security association information for the device that has a source IP address of 10.56.19.172:

```
ipsec sa show 1 -s 10.56.19.172 -p esp
```

Alg/State/Spi	Current Bytes/CreateTime	SrcIPAddr->DstIPAddr

esp/M/0001388	0/20 Aug 2002 17:28:19	10.56.19.172->10.56.19.173

The values for state are:

M Mature and active

D Dead

d Dying

L Larval (uninitiated)

About this appendix This appendix describes the statistics displayed by the `ifstat` command for the network interfaces supported by Data ONTAP.

Topics in this appendix

This appendix discusses statistics for the following interfaces:

- ◆ [“Fast Ethernet interface statistics”](#) on page 174
- ◆ [“Znynx 10Base-T/100Base-TX Ethernet interface statistics”](#) on page 177
- ◆ [“Ethernet Controller II, III, and IV interface statistics”](#) on page 180
- ◆ [“Gigabit Ethernet interface statistics”](#) on page 184
- ◆ [“FDDI interface statistics”](#) on page 186
- ◆ [“ATM interface statistics”](#) on page 190

Fast Ethernet interface statistics

RECEIVE section statistics

The following table describes the statistics in the RECEIVE section of the `ifstat` command output when you use the command on a Fast Ethernet interface, such as an X1001C or X1012C card.

Statistic	Meaning
Frames/second	Rate of received frames per second.
Bytes/second	Rate of received bytes per second.
Errors/minute	Rate of errors (which led to frames being lost) per minute.
Total frames	Total frames that are received on the interface.
Total bytes	Total bytes that are received on the interface.
Total errors	Total errors that occur on the interface.
Multi/broadcast	Total number of multicast or broadcast packets received.
CRC errors	Number of CRC errors that occurred on the received packets due mainly to duplex mismatches.
Alignment errors	Number of frames that are both misaligned and contain CRC errors.
Non-primary u/c	Number of Ethernet frames received for the partner's MAC address after a failover in a cluster configuration.
No buffers	Number of times the driver was unable to get a buffer from its buffer pool because the pool was empty.
Tag drop	Number of tagged frames dropped on an interface that is not configured to support VLAN tagging and receives tagged frames.
Vlan tag drop	Number of tagged frames received and dropped that do not match the tags configured on an interface configured to support VLAN tagging.

Statistic	Meaning
Vlan untag drop	Number of untagged frames dropped on an interface that is configured to be part of a VLAN.
List overflow	Number of frames dropped due to the unavailability of receive resources.
Bus overruns	Number of frames lost due to receive FIFO overflows.
Runt frames	Number of runt frames received.
Long frames	Number of long frames received that exceeded the maximum Ethernet-specified size of 1,518 bytes.
Flow controls	Number of flow control frames received.

TRANSMIT section statistics

The following table describes the statistics in the TRANSMIT section of the `ifstat` command output when you use the command on a Fast Ethernet interface.

Statistic	Meaning
Frames/second	Rate of transmitted frames per second.
Bytes/second	Rate of transmitted bytes per second.
Errors/minute	Rate of errors (which led to frames being lost) per minute.
Total frames	Total frames that are transmitted on the interface.
Total bytes	Total bytes that are transmitted on the interface.
Total errors	Total errors that occur on the interface.
Multi/broadcast	Total number of multicast or broadcast packets transmitted.
Queue overflows	Total number of frames dropped due to software queue overflow.
Max collisions	Total number of frames that were not transmitted because they encountered the maximum number of allowed collisions.

Statistic	Meaning
Late collisions	Number of frames that were not transmitted because they encountered a collision outside the collision window.
Bus underruns	Number of times the transmitter aborted the frame to be transmitted because data arrived late from memory. These packets are retransmitted later.
Lost carriers	Number of frames that were transmitted by the device despite the deassertion of CRS during transmission.
Deferred	Number of frames that were deferred before transmission due to activity on the link.
Single collision	Number of transmitted frames that encountered one and only one collision.
Multiple collision	Number of transmitted frames that encountered more than one collision, but fewer than the maximum allowed collisions.
Flow controls	Number of flow control frames transmitted.

LINK INFO section statistics

The following table describes the statistics in the LINK INFO section of the `ifstat` command output when you use the command on a Fast Ethernet interface.

Statistic	Meaning
Current state	The state of the link. It can be up, down, or enabling.
Up to downs	Number of times the link toggled between LINK_UP and LINK_DOWN states.
Speed	Current negotiated speed.
Duplex	Duplex of the link negotiated or set.
Flow control	Negotiated value of flow control if the interface is autonegotiable; otherwise, it is the configured setting.

Znyx 10Base-T/100Base-TX Ethernet interface statistics

RECEIVE section statistics

The following table describes the statistics in the RECEIVE section of the `ifstat` command output when you use the command on a Znyx 10Base-T/100Base-TX Ethernet interface, such as an X1001, X1001B, X1002, X1003, X1012, or X1012B card.

Statistic	Meaning
Packets	Number of packets received on the interface.
Bytes	Number of bytes received on the interface.
Total errors	Number of errors during Ethernet frame reception, including all kinds of receive errors.
No buffers	Number of received packets dropped due to the unavailability of buffers.
Length errors	Number of frames truncated due to the shortage of receive descriptors.
Runt frames	Number of runt frames.
Long frames	Number of frames received that exceeded the maximum Ethernet-specified size of 1,518 bytes.
CRC errors	Number of CRC errors that occurred on the received frames.
H/w overflow	Number of frames discarded because of receive FIFO overflow.
List overflow	Number of frames dropped due to the unavailability of descriptors.
Rst frame drops	Number of frames discarded due to the resetting of the receive process.
Collisions	Number of frames received that were damaged by late collisions.

TRANSMIT section statistics

The following table describes the statistics in the TRANSMIT section of the `ifstat` command output when you use the command on a Znyx 10Base-T/100Base-TX Ethernet interface.

Statistic	Meaning
Packets	Number of packets attempted to be transmitted.
Bytes	Number of bytes attempted to be transmitted.
Total errors	Number of hardware errors encountered while attempting to transmit.
Collisions	Number of collisions that occurred while transmitting frames.
Late collisions	Number of collisions aborted due to a late collision. This usually indicates a duplex mismatch.
Excess coll	Number of times transmission was aborted due to excessive collisions.
Queue overflow	Number of frames dropped because the software queue was full.
No carrier	Number of times the carrier signal was not present during transmission.
H/w underflow	Number of times the transmitter aborted the message because data arrived late from memory.
Defer	Number of times transmission had to be deferred.
Link failure	Number of times the link failed.
Time out	Number of times the transmit jabber timer expired.
List underflow	Number of frames that had to be dropped due to the unavailability of descriptors.
Loss of carrier	Number of times the carrier was lost.
No buffers	Number of times buffers were unavailable.

DEVICE section statistics

The following table describes the statistics in the DEVICE section of the `ifstat` command output when you use the command on a Znyx 10Base-T/100Base-TX Ethernet interface.

Statistic	Meaning
Resets	Number of times the Ethernet device was reset.

LINK INFO section statistics

The following table describes the statistics in the LINK INFO section of the `ifstat` command output when you use the command on a Znyx 10Base-T/100Base-TX Ethernet interface.

Statistic	Meaning
Auto	Autonegotiation state.
Mediatype	Media type, such as twisted pair.
Link Partner	The Autonegotiation capability of the remote end. It is unknown if Autonegotiation is disabled.
Link State	Link status.

Ethernet Controller II, III, and IV interface statistics

RECEIVE section statistics

The following table describes the statistics in the RECEIVE section of the `ifstat` command output when you use the command on Gigabit Ethernet Controllers II, III, IV, and V, both fiber and copper, as well as the onboard 10 Base-T/100 Base-TX Ethernet Controller IV.

Statistic	Meaning
Frames/second	Rate of received frames per second.
Bytes/second	Rate of received bytes per second.
Errors/minute	Rate of errors (which led to frames being lost) per minute.
Total frames	Total frames that are received on the interface.
Total bytes	Total bytes that are received on the interface.
Total errors	Total errors that occur on the interface.
Multi/broadcast	Total number of multicast or broadcast packets received.
Alignment errors	Number of frames that are both misaligned and contain CRC errors.
Non-primary u/c	Number of Ethernet frames received for the partner's MAC address after a failover in a cluster configuration.
Tag drop	Number of tagged frames dropped on an interface that is not configured to support VLAN tagging.
Vlan tag drop	Number of tagged frames dropped that do not match the VLAN tags configured on the interface.
Vlan untag drop	Number of untagged frames dropped on an interface that is configured to be part of a VLAN.
CRC errors	Packets with bad CRC.
Bad length	Total number of received packets with a bad length. These are frames counted as undersize, fragment, oversize, or jabber.

Statistic	Meaning
Runt frames	Number of received frames that were less than the minimum size (64 bytes) and had a valid CRC.
Fragment	Number of received frames that were less than the minimum size and had a bad CRC.
Long frames	Number of received frames that were greater than the maximum size and had a valid CRC.
Jabber	Number of received frames that were greater than the maximum size and had a bad CRC.
Bus overruns	Number of times the adapter's receive FIFO overflowed and a packet was dropped. This occurs when the bus is very busy and the adapter cannot transfer data into host memory. This might also occur when the filer CPU is very busy and cannot process the received packets fast enough.
Queue overflows	Number of frames dropped on receive due to the driver receive queue overflowing.
No buffer	Number of times the driver could not allocate a buffer and a packet was dropped. This might happen when the filer is very busy. If the count increases continually, it might indicate that a software component is not returning buffers.
Xon	Number of XON frames received when receive or full flow control is enabled.
Xoff	Number of XOFF frames received when receive or full flow control is enabled.
Jumbo	Number of good packets received that were larger than the standard Ethernet packet size when jumbo frames are enabled.
Reset	Number of times the driver reset the NIC because the NIC was in a bad state.
Reset1	Number of times the driver reset the NIC because the NIC was in a bad state.

TRANSMIT section statistics

The following table describes the statistics in the TRANSMIT section of the `ifstat` command output when you use the command on Gigabit Ethernet Controllers II, III, IV, and V, both fiber and copper, as well as the onboard 10 Base-T/100 Base-TX Ethernet Controller IV.

Statistic	Meaning
Frames/second	Rate of transmitted frames per second.
Bytes/second	Rate of transmitted bytes per second.
Errors/minute	Rate of errors (which led to frames being lost) per minute.
Total frames	Total frames that are transmitted on the interface.
Total bytes	Total bytes that are transmitted on the interface.
Total errors	Total errors that occur on the interface.
Multi/broadcast	Total number of multicast or broadcast packets transmitted.
Queue overflow	Number of outgoing packets dropped because the driver's queue was full. It might indicate a system problem.
Max collisions	Number of frames that were not transmitted because they encountered the maximum number of allowed collisions. Only valid in half-duplex mode.
Single collision	Number of frames that encountered exactly one collision. Only valid in half-duplex mode.
Multi collisions	Number of frames that encountered more than one collision, but less than the maximum allowed. Only valid in half-duplex mode.
Late collisions	Number of collisions that occurred outside the collision window. Only valid in half-duplex mode.
Xon	Number of XON frames transmitted when send or full flow control is enabled.
Xoff	Number of XOFF frames transmitted when send or full flow control is enabled.

Statistic	Meaning
Timeout	Number of times the adapter's transmitter hung and the adapter had to be reset. This can happen when the cable is pulled and the transmitter cannot transmit a packet. The adapter is reset to reclaim packet buffers.
Jumbo	Number of packets transmitted that were larger than the standard Ethernet packet size when jumbo frames are enabled.

LINK INFO section statistics

The following table describes the statistics in the LINK INFO section of the `ifstat` command output when you use the command on Gigabit Ethernet Controller II and III interfaces and 100 Base-T/1000 Base-T Ethernet III interfaces.

Statistic	Meaning
Current state	Current state of the interface: ◆ up or down—The state of the link. ◆ <code>cfg_down</code>—The interface is configured down. ◆ <code>enabling</code>—The interface is coming up.
Up to downs	Number of times the link toggled between up and down.
Auto	Operational state of autonegotiation: ◆ on—Autonegotiation is enabled and succeeded. ◆ off—Autonegotiation failed. This happens when the device to which the interface is connected has disabled autonegotiation or is incompatible with the interface. This may also indicate that the interface is down.
Speed	Speed of link negotiated or set.
Duplex	Duplex of the link negotiated or set.
Flow control	The operations flow control setting. For information on how the operational flow control setting is determined, see Chapter 1, “ Network Interface Configuration ,” on page 1.

Gigabit Ethernet interface statistics

RECEIVE section statistics

The following table describes the statistics in the RECEIVE section of the `ifstat` command output when you use the command on a Gigabit Ethernet interface.

Statistic	Meaning
Packets	Number of packets received on the interface.
Bytes	Number of bytes received on the interface.
Errors	Number of errors during Ethernet frame reception, including all kinds of receive errors.
Queue full	Number of received packets dropped due to the unavailability of buffers.
Unicast packets	Number of unicast packets received.
Runt frames	Number of runt frames received.
No buffers	Number of times the driver failed to get buffers from the system for its receive descriptor ring.

TRANSMIT section statistics

The following table describes the statistics in the TRANSMIT section of the `ifstat` command output when you use the command on a Gigabit Ethernet interface.

Statistic	Meaning
Packets	Number of packets attempted to be transmitted.
Bytes	Number of bytes attempted to be transmitted.
Errors	Number of hardware errors encountered while attempting to transmit.
Collisions	Number of collisions that occurred while transmitting frames.
Unicast packets	Number of unicast packets transmitted.

DEVICE section statistics

The following table describes the statistics in the DEVICE section of the `ifstat` command output when you use the command on a Gigabit Ethernet interface.

Statistic	Meaning
Received errors	Number of errors encountered during reception by the interface.
Transmit errors	Number of errors encountered during transmission by the interface.
Collisions	Number of collisions encountered during transmission by the interface.

FDDI interface statistics

RECEIVE section statistics

The following table describes the statistics in the RECEIVE section of the `ifstat` command output when you use the command on an FDDI interface.

Statistic	Meaning
Packets	Number of packets received on the interface.
Bytes	Number of bytes received on the interface.
Total Errors	Number of errors during reception, including all kinds of receive errors.
Queue full	Number of received packets dropped due to the unavailability of buffers.
H/w faults	Number of hardware faults.
SA matches	Number of SA matches.
DA nonmatches	Number of DA nonmatches.
PDQ aborts	Number of PDQ aborts.
Invalid lengths	Number of invalid lengths.
Fragments	Number of fragments.
Format errs	Number of frames received that contained an odd number of symbols; that is, the number of nonintegral octets.
MAC resets	Number of MAC resets.
CRC Errors	Number of frames received with a bad frame check sequence.
Status Errors	Number of frames received that had the E (error detected) indicator set or a missing E indicator.
Long Frames	Number of frames received that were larger than the maximum legal size.
Short Frames	Number of short frames received.
Copied buffers	Number of copied buffers.

Statistic	Meaning
No buffers	Number of times there were insufficient buffers.
LLC errors	Number of LLC errors.
Blk check errs	Number of block check errors.
Length errors	Number of length errors.
Overruns	Number of times that the interface received a frame loss indication from the hardware on the interface.

TRANSMIT section statistics

The following table describes the statistics in the TRANSMIT section of the ifstat command output when you use the command on an FDDI interface.

Statistic	Meaning
Packets	Number of packets attempted to be transmitted.
Bytes	Number of bytes attempted to be transmitted.
Errors	Number of hardware errors encountered while attempting to transmit.
Queue full	Number of received packets dropped due to unavailability of buffers.
No buffers	Number of times buffers were not available.
No buf resets	Number of times the process was reset due to the unavailability of buffers.
No descriptors	Number of times descriptors were unavailable.
I/f down	Number of times the interface was down.
Underruns	Number of times an underrun occurred because the interface's transmit FIFO became empty due to insufficient memory. That is, the device tried to transmit but could not get data into memory fast enough.
Failures	Number of frames that were transmitted with an error.

DEVICE section statistics

The following table describes the statistics in the DEVICE section of the `ifstat` command output when you use the command on an FDDI interface.

Statistic	Meaning
Interrupts	Number of times the FDDI device interrupted the host.
Spurious ints	Number of spurious interrupts received from the device.
Resets	Number of times the device was reset due to unrecoverable hardware faults.
Non-existing mem	Number of times the device was asked to address nonexistent memory.
Parity errs	Number of parity errors detected on the interface.
Host parity errs	Number of parity errors detected on the host.
Spurious Unsols	Number of unknown unsolicited error events reported by this link entity.
Ring inits in	Number of ring claim processes initiated by this link entity.
Ring beacon out	Number of ring beacon processes initiated by this link entity.
Ring beacon in	Number of ring beacon processes recognized by this link entity.
Dupl. address	Number of times the duplicate address test failed.
Dupl. token	Number of times this link entity detected a duplicate token.
Ring purge errs	Number of times the MAC PurgeError Event bit was set.
FCI strip errs	Number of times a Frame Content Independent Strip operation was terminated.
Trace init	Number of PC traces initiated by this station.
Selftest failure	Number of failed self-tests during device resets.
PC trace rcvd	Number of PC traces received by this station.

Statistic	Meaning
Config changes	Number of configuration changes.
LEM rejects	Number of Link Error Monitor (LEM) rejects.
Elastic buf errs	Number of Elasticity Buffer and Link Monitor (ELM) Chip EBUFF_ERROR events.
Link test reject	Number of link test rejects.

ATM interface statistics

RECEIVE section statistics

The following table describes the statistics in the RECEIVE section of the `ifstat` command output when you use the command on an ATM interface.

Statistic	Meaning
Packets	Number of packets received on the interface.
Bytes	Number of bytes received on the interface.
Errors	Number of errors during reception, including all kinds of receive errors.
Queue full	Number of packets dropped because they could not be put in the transmit queue.
Collisions	Ignore this field.

TRANSMIT section statistics

The following table describes the statistics in the TRANSMIT section of the `ifstat` command output when you use the command on an ATM interface.

Statistic	Meaning
Packets	Number of packets attempted to be transmitted.
Bytes	Number of bytes attempted to be transmitted.
Errors	Number of hardware errors encountered while attempting to transmit.

About this appendix This appendix describes configuration procedures that might improve your filer's performance.

Balance NFS traffic on network interfaces Attach multiple interfaces on the filer to the same physical network to balance network traffic among different interfaces. For example, if two Ethernet interfaces on the filer named toaster are attached to the same network where four NFS clients reside, specify in the `/etc/fstab` file on client1 and client2 that these clients mount from toaster-0:/home. Specify in the `/etc/fstab` file on client3 and client4 that these clients mount from toaster-1:/home. This scheme can balance the traffic among interfaces if all clients generate about the same amount of traffic.

The filer always responds to an NFS request by sending its reply on the interface on which the request was received.

Correct duplex mismatches on a 10Base-T or 100Base-T Ethernet networks On 10Base-T or 100Base-T Ethernet networks, the speed and duplex settings for the interfaces at both ends of a link must match exactly. Use the `ifconfig interface` command to check the duplex setting of the filer's interface. If the setting is set to autonegotiate, the `ifconfig` command displays a setting that begins with `auto` (for example, `auto-100tx-fd-up`). Otherwise, the `ifconfig` command displays the setting (for example, `100tx-fd-up`).

Note

If one end of the link is set to autonegotiate, the other end must also be set to autonegotiate; otherwise, a mismatch might occur. You can determine the negotiated setting with the `ifstat` command.

Make optimal use of Slot 8 This method of improving filer performance applies to F700 series filers only. The fastest available PCI slot in an F700 series filer that is not clustered is Slot 8. If your filer uses a 64-bit network interface card, put it in Slot 8 to take advantage of the speed. If the filer does not use a 64-bit network interface card, put the main FC-AL adapter in this slot.

Upgrade to a faster network interface

You can increase filer performance by upgrading to a faster network interface. The following lists network interfaces from the fastest to the slowest:

- ◆ Gigabit Ethernet interfaces
- ◆ ATM OC-12 interfaces
- ◆ Gigabit Ethernet interface
- ◆ ATM OC-3 interface
- ◆ Fast Ethernet 100Base-T interface
- ◆ F700 series onboard 100Base-T interfaces
- ◆ Znyx 100Base-T and FDDI interfaces
- ◆ 10Base-T interface

Determine when to use UDP or TCP transport

The following are guidelines to determine when you should use the UDP transport or the TCP transport to improve filer performance:

- ◆ Use the TCP transport over a WAN network.
- ◆ Use the UDP transport over a LAN network.
- ◆ Use the TCP transport if you are using the UDP transport and you experience packet loss, especially during periods of heavy write traffic.

You can specify the transport using the `options nfs.tcp.enable` command.

Adjust flow control for Gigabit Ethernet II on F700 series filers

If you are using the Gigabit Ethernet II interface on an F700 series filer that is connected to a switch, you can adjust the flow control to improve maximum transmit performance. The default flow control setting `full` should be changed to `send` or `none`. Most switches rarely send flow control frames, so disabling `receive` should not cause problems. This does not apply to other Gigabit Ethernet interfaces or other platforms.

Note

Do not perform this tuning when the interface is connected back-to-back with another host machine. In this situation, the flow control setting should be set to `full` on both sides to avoid overruns and poor performance.

Index

Symbols

/etc/hosts file

creation of 76

resolving host names with 75

updating of 76

/etc/rc file, default route 63

/etc/resolv.conf file, creating 79

A

adapter failover, automatic 27

aliases

configuring (ifconfig) 11

for an interface 10

ATM and LANs, bridging between 25

ATM commands

atm adinfo (verifies adapter operation) 31

atm adstat (verifies connection works) 32

atm atmarp (deletes incoming FORE/IP PVCs)
60

atm atmarp (deletes outgoing FORE/IP PVCs)
60

atm atmarp (displays FORE/IP
PVC address resolution) 56

atm atmarp (establishes incoming FORE/IP
PVCs) 55

atm atmarp (establishes outgoing FORE/IP
PVCs) 55

atm atmconfig (changes ATM AAL) 59

atm atmconfig (changes SPANS AAL) 59

atm atmconfig (displays configuration data) 57

atm elconfig add (adds emulated LAN) 39

atm elconfig delete (deletes emulated LAN
from adapter) 42

atm elconfig set (configures LANE
configuration server) 35

atm elconfig show (verifies adapter
configurations) 45

atm elconfig show (verifies elements of
emulated LAN) 46

atm uniconfig set failover (modifies load
balancing groups) 49

atm uniconfig show (verifies UNI operation)
33

ATM protocol

automatic adapter failover of 27

bridging between ATM and LANs 25

BUS, description of 27

cause codes 25

cells, description of 24

checking UNI operation (atm uniconfig show)
33

configuring logical Ethernet interface
(ifconfig) 41

deleting emulated LAN from adapter (atm
elconfig delete) 42

description of 24

differences between LANs and 25

emulated LANs

adding (atm elconfig add) 39

components of 26

deleting from adapter (atm elconfig
delete) 42

description of 26

saving host and IP address in 51

verifying adapter configurations (atm
elconfig show) 45

verifying communications (ping) 44

verifying elements of (atm elconfig show)
46

establishing incoming FORE/IP PVCs (atm
atmarp) 55

establishing outgoing FORE/IP PVCs (atm
atmarp) 55

FORE/IP

changing ATM AAL (atm atmconfig) 59

deleting incoming PVCs (atm atmarp) 60

deleting outgoing PVCs (atm atmarp) 60

description of 24

displaying configuration data (atm
atmconfig) 57

over SPANS, description of 52

PVCs, description of 54

PVCs, displaying address resolution (atm

atmarp) 56

LANE

- Clients, description of 26
- configuration server, configuring (atm elconfig set) 35
- configuration server, description of 27
- description of 24, 25, 26
- handling addressing and resolution of 28
- preparing ATM adapter to use 29
- Server, description of 27
- standards supported 28

load balancing 27, 48

- description of 48
- modifying load balancing groups (atm uniconfig set failover) 49

UNI 27

PVCs

- and SVCs, description of 52
- description of 53

saving configuration commands 50

saving host and IP address 51

SPANS

- changing the SPANS AAL (atm atmconfig) 59

UNI (User-Network Interface), description of 27

VCCs (Virtual Channel Connections), component of emulated LANs 26

verifying adapter configurations in (atm elconfig show) 45

verifying adapter operation (atm adinfo) 31

verifying connection works (atm adstat) 32

verifying elements of emulated LAN (atm elconfig show) 46

ways to use 24

automatic adapter failover 27

B

Before 6

bridging between ATM and LANs 25

broadcast address, changing (ifconfig) 9

BUS, within an emulated LAN 27

C

cause codes, ATM 25

configuration

- of aliases 11

- of LANE configuration server 35

- of logical Ethernet interface 41

- of media types 14

D

Data ONTAP

- default route 63

- NetApp custom MIB, description of 97

- routing, description of 62

- SNMP protocol

 - commands supported 100

 - configuration commands 100

 - implementation, description of 96

 - location of 97

 - trap commands 102

DNS

- changing domain name (options dns.domainname) 79

- configuring 79

- enabling and disabling (options dns.enable) 80

DNS commands

- options dns.domainname (changes domain name) 79

- options dns.enable (enables and disables DNS) 80

domain names, changing of 79

E

Emulated LANs

- adding (atm elconfig add) 39

- and a LANE Client 27

- ATM BUS, description of 27

- components of 26

- deleting from adapter 42

- description of 26

- saving host and IP address in 51

- verifying communications (ping) 44

- verifying elements of 46

Ethernet interfaces, media types 12

F

failover

- modifying load-balancing groups 49
- of adapter 27

fast path mechanism, description of 62

filer, improving performance in 191

flow control on Gigabit Ethernet

- configuring 16
- description of 15

FORE/IP

- changing ATM AAL (atm atmconfig) 59
- displaying configuration data (atm atmconfig) 57

over SPANS, description of 52

PVCs

- deleting incoming (atm atmarp) 60
- deleting outgoing (atm atmarp) 60
- displaying address resolution (atm atmarp) 56
- establishing incoming (atm atmarp) 55
- establishing outgoing (atm atmarp) 55
- establishment of 54

frame size, definition of 5

G

Gigabit Ethernet

- description of 12
- flow control
 - configuring 16
 - description of 15

Gigabit Ethernet interfaces, media types 12

H

host names

- for interfaces, description of 3
- resolving 75

I

ifconfig command

- description of 7
- ifconfig (changes broadcast address) 9

ifconfig (changes interface status) 18

ifconfig (changes IP address) 8

ifconfig (changes network mask) 9

ifconfig (configures alias for an interface) 11

ifconfig (configures media type) 14

ifconfig (views configuration of interface) 19

what can be changed 8

interface command

ifconfig (destroys virtual interface) 133

interfaces

alias

- configuring (ifconfig) 11
- description of 10

broadcast address, changing (ifconfig) 9

current configuration views 19

description of 2

Gigabit Ethernet

- description of 12
- flow control 15, 16

host name creation, description of 3

ifconfig command 7, 8

IP address, changing (ifconfig) 8

jumbo frame and MTU size definition 5

jumbo frames

- and MTU size 5
- client-size recommendations 6
- description of 5
- infrastructure requirements 6
- ways to set up 6

media types on Ethernet 12

media types, configuring (ifconfig) 14

MTU size, setting 17

mtusize option, description of 17

multiple ports, description of 2

naming conventions 3

network mask, changing (ifconfig) 9

numbering of 2

physical, adding (vif add) 126

status of, changing 18

supported for VLAN setup 149

types of 2

viewing current configuration (ifconfig) 19

IP addresses, changing (ifconfig) 8

J

jumbo frames

- and MTU size 5
- client-size recommendations 6
- description of 5
- infrastructure requirements 6
- ways to set up 6

L

LANE

- and Emulated LAN configuration information 26
- Client, description of 26
- configuration server, configuring 35
- configuration server, description of 27
- description of 25
- description of service 26
- handling addressing and resolution of 28
- preparing ATM adapter to use 29
- Server, description of 27
- service, description of 25
- standards supported 28

LANs, bridging between ATM and 25

LDAP

- editing /etc/nsswitch.conf file 89
 - enabling and disabling (options ldap.enable) 89
 - servers 87
 - specify base distinguished name (options ldap.base) 88
 - specify preferred servers (options ldap.servers.preferred) 88
 - specify servers (options ldap.servers) 88
 - specifying admin password (options ldap.passwd) 91
 - specifying admin user name (options ldap.name) 90
 - specifying LDAP port (options ldap.port) 91
- ### LDAP commands
- options ldap.base (specifies base distinguished name) 88
 - options ldap.enable (enables and disables LDAP) 89
 - options ldap.name (specifies admin user name)

90

- options ldap.passwd (specifies admin password) 91
- options ldap.port (specifies LDAP port) 91
- options ldap.servers (specifies servers) 88
- options ldap.servers.preferred (specifies preferred servers) 88

M

media types, how auto works 13

MIB (NetApp custom), location of 97

MTU size

- definition of 5
- mtusize option 17
- setting the 17

multi-mode virtual interface, creating (vif create multi) 125, 136

multiple ports on interfaces, description of 2

N

name resolution, NIS and DNS configuration files 73

NetApp custom MIB, location of 97

network commands

- ifconfig 7
- ifconfig (configures logical Ethernet interface) 41
- ifstat and netstat 21

network interfaces

- filer supported 2
- statistics, displaying (ifstat) 21
- virtual 2

network mask, changing (ifconfig) 9

network, VLAN 144

NFS protocol over-UDP routing, description of 62

NIS

- changing NIS domain names (options nis.domainname) 85
- displaying information (nis info) 81, 85
- displaying server name (ypwhich) 86
- specifying servers to bind to (options nis.servers) 83

NIS commands

- nis info (displays NIS information) 81, 85

- options nis.domainname (changes NIS domain name) 85
- options nis.servers (binds NIS servers) 83
- ypwich (displays NIS server name) 86

P

- packets, jumbo frames 5
- physical interfaces, adding (vif add) 126
- PVCs
 - deleting incoming FORE/IP (atm atmarp) 60
 - deleting outgoing FORE/IP (atm atmarp) 60
 - description of 53
 - displaying address resolution (atm atmarp) 56
 - establishing incoming FORE/IP (atm atmarp) 55
 - establishing outgoing FORE/IP (atm atmarp) 55

R

- routing
 - default route 63
 - description of 62
 - fast path mechanism 62
 - NFS-over-UDP, description of 62
 - routed daemon 62
 - table
 - description of 66
 - displaying (netstat) 66
 - modification of 68
 - modifying (route) 68
 - TCP, description of 62
 - turning on or off (routed) 65
- routing commands
 - nr (displays routing table) 66
 - route (modifies routing table) 68
 - routed (turns routing on or off) 65
- routing, TCP 62

S

- search order, changing (nsswitch.conf file) 92
- second-level interfaces
 - creating in a cluster 140
 - creating in a single filer 136
 - in a cluster, description of 138

- in single filer, description of 135
- single-mode virtual interface
 - creating (vif create single) 122
 - specifying preferred (vif favor) 123
- SNMP commands
 - snmp configuration 100
 - snmp traps 102
- SNMP protocol
 - agent and groups supported 96
 - commands supported by Data ONTAP 100
 - Data ONTAP implementation, description of 96
 - MIB specifications implemented 96
 - NetApp custom MIB, description of 97
 - trap commands 102
 - traps
 - defining (snmp traps) 105
 - description of 104
 - modifying (snmp traps) 105
 - parameters supported 109
 - types of 104
- SPANS, changing the AAL (atm atmconfig) 59
- statistics
 - DEVICE statistics on Fast Ethernet card, description of 179
 - DEVICE statistics on FDDI card, description of 188
 - DEVICE statistics on Gigabit Ethernet card, description of 185
 - displaying interface (ifstat) 21
 - LINK INFO statistics on Fast Ethernet card, description of 176
 - LINK INFO statistics on Gigabit Ethernet Controller II, description of 183
 - LINK INFO statistics on Znyx card, description of 179
 - RECEIVE statistics on 100/1000 Ethernet Controller III, description of 182, 183
 - RECEIVE statistics on 100Base-T/1000Base-T Ethernet Controller III, description of 180
 - RECEIVE statistics on ATM card, description of 190
 - RECEIVE statistics on Fast Ethernet card, description of 174

RECEIVE statistics on FDDI card, description of 186

RECEIVE statistics on Gigabit Ethernet card, description of 184

RECEIVE statistics on Gigabit Ethernet Controller II, description of 180

RECEIVE statistics on Gigabit Ethernet Controller III, description of 180, 182, 183

RECEIVE statistics on Znyx card, description of 177

TRANSMIT statistics on ATM card, description of 190

TRANSMIT statistics on Fast Ethernet card, description of 175

TRANSMIT statistics on FDDI card, description of 187

TRANSMIT statistics on Gigabit Ethernet card, description of 184

TRANSMIT statistics on Gigabit Ethernet Controller II, description of 182

TRANSMIT statistics on Znyx card, description of 178

stats

ifstat command, description of 21

stats commands

ifstat (displays interface statistics) 21

IPsec stats (displays IPsec statistics) 169

SVCs and PVCs, description of 52

T

TCP routing 62

This 126

trap commands 102

traps

defining or modifying (snmp traps) 105

description of 104

parameters supported 109

types of 104

trunking interfaces

description of 114

guidelines for 119

U

UNI, description of 27

user authentication, NIS and DNS configuration files 73

V

vifs

adding physical interfaces (vif add) 126

destroying virtual interface (ifconfig) 133

displaying statistics of virtual interface (vif stat) 132

displaying status of virtual interface (vif status) 128

multi-mode interface

creating (vif create multi) 124, 125, 136

multi-mode operation 117

preferred single-mode interfaces

specifying (vif favor) 123

second-level interfaces

creating in a cluster 140

creating in a single filer 136

description of 134

in a cluster 138

in single filer 135

single-mode interface

creating (vif create single) 122

specifying preferred (vif favor) 123

single-mode operation 116

trunking interfaces, description of 114

trunking, guidelines for 119

vif command, description of 119

virtual interfaces

kinds of 116, 117

management of (vif command) 119

vifs commands

vif add (adds physical interface) 126

vif create multi (creates multi-mode interface) 125

vif create single (creates single-mode interface) 122

vif favor (specifies preferred interface) 123

vif stat (displays statistics of virtual interface) 132

vif status (displays status of virtual interface)

- 128
- virtual local area network. *See* VLAN
- virtual network interfaces. *See* vifs
- VLAN
 - adding an interface to 154
 - advantages of 146
 - configuring on a filer 152
 - considerations for reverting Data ONTAP
 - version 149
 - creating on a filer 151
 - definition 144
 - deleting on a filer 155
 - display statistics of 158
 - guidelines for setting up 148
 - how tagging works 146
 - ifconfig command 152
 - interfaces supported 149
 - members, communication between 144
 - membership 144

- persistence across reboots 148
 - port-based 144
 - setup limitations 149
 - setup requirements 147
 - statistics, viewing 158
 - tag 146
 - vlan command 150
- VLAN commands
 - add 154
 - create 151
 - delete 155
 - persistence of 150
 - stat 158
 - syntax of 150

Y

- You 124

