



IP Network Multipathing Administration Guide

Alpha

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part No: 816-0850-10
July 2001

Sun Proprietary/Confidential: Need to Know

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303-4900 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



040210@7940



Sun Proprietary/Confidential: Need to Know

Contents

Preface	5
1 Overview of IP Network Multipathing	9
Introduction	9
IP Network Multipathing Features	10
Communication Failures	10
IP Network Multipathing Components	11
Solaris Network Multipathing	12
Detecting Physical Interface Failures	12
Detecting Physical Interface Repairs	13
Link Failures	13
Administering Multipathing Groups with Multiple Physical Interfaces	14
Grouping Physical Interfaces	15
Configuring Test Addresses	16
Using the <code>hostname</code> File to Configure Groups and Test Addresses	18
Configuring Standby Interfaces	19
Administering Multipathing Groups With a Single Physical Interface	20
Removing Network Adapters From Multipathing Groups	21
Detached Network Adapters	22
Multipathing Daemon	22
Multipathing Configuration File	24
Failure Detection Time	24
Failback	25
Track Interfaces Only With Groups Option	25

2	Deploying Network Multipathing	27
	Configuring Multipathing Interface Groups	27
	Configuring Multipathing Interface Groups—Task Map	28
	▼ How to Configure a Multipathing Interface Group with Two Interfaces	28
	▼ How to Configure a Multipathing Group Where One of the Interfaces is a Standby Interface	31
	▼ How to Display the Group to Which a Physical Interface Belongs	33
	▼ How to Add an Interface To a Group	34
	▼ How to Remove an Interface From a Group	34
	▼ How to Move an Interface From an Existing Group to a Different Group	34
	Replacing a Physical Interface That Has Failed	35
	▼ How to Remove a Physical Interface That Has Failed	35
	▼ How to Replace a Physical Interface That Has Failed	36
	Recovering a Physical Interface That Was Not Present at System Boot	36
	▼ How to Recover a Physical Interface That Was Not Present at System Boot	37
	Configuring the Multipathing Configuration File	38
	▼ How to Configure the Multipathing Configuration File	38
	Glossary	41
	Index	43

Preface

The *IP Network Multipathing Administration Guide* provides information about configuring and managing the IP Network Multipathing framework installed in your Solaris™ Operating Environment. This book assumes that you have already installed the SunOS™ 5.8 operating system, and you have set up any networking software that you plan to use. The SunOS 5.8 operating system is part of the Solaris product family, which includes many features, such as the Solaris Common Desktop Environment (CDE). The SunOS 5.8 operating system is compliant with AT&T's System V, Release 4 operating system.

Note – The Solaris operating environment runs on two types of hardware, or platforms: SPARC™ and IA. The Solaris operating environment runs on both 64-bit and 32-bit address spaces. The information in this document pertains to both platforms and address spaces unless called out in a special chapter, section, note, bullet, figure, table, example, or code example.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Solaris 8 release. To use this book, you should have one to two years of UNIX system administration experience. Attending UNIX® system administration training courses might be helpful.

How This Book Is Organized

Chapter 1 provides an overview of IP Network Multipathing and also describes conceptual information about the Solaris deployment of IP Network Multipathing.

Chapter 2 describes how to configure the various IP Network Multipathing parameters for creating interface groups and test address. This chapter also provides other useful procedures related to IP Network Multipathing.

Glossary provides definitions of key IP Network Multipathing terms.

Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at <http://www1.fatbrain.com/documentation/sun>.

Accessing Sun Documentation Online

The docs.sun.comSM Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is <http://docs.sun.com>.

What Typographic Conventions Mean

The following table describes the typographic changes used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. machine_name% you have mail.
AaBbCc123	What you type, contrasted with on-screen computer output	machine_name% su Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type rm <i>filename</i> .
<i>AaBbCc123</i>	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You must be <i>root</i> to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	\$
Bourne shell and Korn shell superuser prompt	#

Overview of IP Network Multipathing

IP Network Multipathing provides both load spreading and failover when you have multiple network interface cards connected to the same IP link (for example, Ethernet).

- “Introduction” on page 9
- “IP Network Multipathing Features” on page 10
- “Communication Failures” on page 10
- “IP Network Multipathing Components” on page 11
- “Solaris Network Multipathing” on page 12
- “Administering Multipathing Groups with Multiple Physical Interfaces” on page 14
- “Administering Multipathing Groups With a Single Physical Interface” on page 20
- “Removing Network Adapters From Multipathing Groups” on page 21
- “Detached Network Adapters” on page 22
- “Multipathing Daemon” on page 22
- “Multipathing Configuration File” on page 24

Introduction

IP Network Multipathing provides your system with the following capabilities:

- Recovery from single-point failures with network adapters
- Increased traffic throughput

If a failure occurs in the network adapter, and if you have an alternate adapter connected to the same IP link, the system switches all the network accesses automatically from the failed adapter to the alternate adapter. This process ensures uninterrupted access to the network. Also, when you have multiple network adapters connected to the same IP link, you achieve increased traffic throughput by spreading the traffic across multiple network adapters.

Note – Other IP related documents, such as RFC 2460, use the term *link* instead of *IP link*. This document uses the term *IP link* to avoid confusion with IEEE 802. In IEEE 802, *link* refers to a single wire from an Ethernet NIC to an Ethernet switch.)

See IP link definition in the Glossary or refer to Table 1-1.

IP Network Multipathing Features

The Solaris implementation of IP Network Multipathing provides the following features:

- **Failure Detection**—Ability to detect when a network adapter has failed and automatically switching (*failover*) the network access to an alternate network adapter. This assumes that you have configured an alternate network adapter. See “Detecting Physical Interface Failures” on page 12 for more information.
- **Repair Detection**—Ability to detect when a network adapter that failed previously has been repaired and automatically switching back (*failback*) the network access to an alternate network adapter. This assumes that you have enabled failbacks. See “Detecting Physical Interface Repairs” on page 13 for more information.
- **Outbound Load Spreading**—Outbound network packets are spread across multiple network adapters without affecting the ordering of packets in order to achieve higher throughput. Load spreading occurs only when the network traffic is flowing to multiple destinations using multiple connections.

Communication Failures

Communication failures can occur in the following ways:

1. Transmit/receive path of the NIC can stop transmitting packets.
2. Attachment of the NIC to the link is down.
3. Port on the switch does not transmit/receive packets.
4. Physical interface in a group not present at system boot.
5. Host on the other end is not responding or the router that is forwarding the packets is not responding.

The Solaris implementation of IP Network Multipathing addresses the first four types of communication failures.

IP Network Multipathing Components

The following table identifies and describes the components that make up IP Network Multipathing.

TABLE 1-1 IP Network Multipathing Components

Component	Description
IP Link	A communication facility or medium over which nodes can communicate at the link layer. The link layer is the layer immediately below IPv4/IPv6. Examples include Ethernets (simple or bridged) or ATM networks. One or more IPv4 subnet numbers/prefixes are assigned to an IP link. A subnet number/prefix can not be assigned to more than one IP link. In ATM LANE, an IP link is a single emulated LAN. When using ARP, the scope of the ARP protocol is a single IP link.
Network Interface Card (NIC)	Network adapter which is either internal or a separate card that serves as an interface to a link.
Physical interface	A node's attachment to a link. This attachment is often implemented as a device driver plus a network adapter. Some network adapters can have multiple points of attachment, for example, qfe. The usage of <i>Network adapter</i> in this document refers to a "Single Point of Attachment."
Physical interface group	The set of physical interfaces on a system that are connected to the same link. They are identified by assigning the same (non-null) character string name to all the physical interfaces in the group.
Physical interface group name	A name assigned to a physical interface that identifies the group. The name is local to a system. Multiple physical interfaces, sharing the same group name, form a physical interface group.
Failure detection	The process of detecting when a NIC or the path from the NIC to some layer 3 device no longer works.
Repair detection	The process of detecting when a NIC or the path from the NIC to some layer 3 device starts operating correctly after a failure.
Failover	The process of switching network access from a failed interface to a good physical interface. Network access includes IPv4 unicast, multicast, and broadcast traffic, as well as IPv6 unicast and multicast traffic.
Failback	The process of switching back network access to an interface detected as having been repaired.

TABLE 1-1 IP Network Multipathing Components (Continued)

Component	Description
Standby Interface	A physical interface that is not used to carry data traffic unless some other physical interface in the group has failed.

Solaris Network Multipathing

The following components implement Solaris network multipathing:

- Multipathing daemon – `in.mpathd`
- IP

The `in.mpathd` daemon detects failures and implements various policies for failover and failback. After `in.mpathd` detects a failure or repair, `in.mpathd` sends an `ioctl` to do the failover or failback. IP, which implements the `ioctl`, does the network access failover transparently and automatically.



Caution – You should not use Alternate Pathing while using IP Network Multipathing on the same set of NICs. Likewise, you should not use IP Network Multipathing while using Alternate Pathing. You can use Alternate Pathing and IP Network Multipathing at the same time on different sets of NICs.

Detecting Physical Interface Failures

The `in.mpathd` daemon sends ICMP echo probes to the targets connected to the link on all the interfaces that belong to a group to detect failures and repair. After you add an interface to a multipathing group and assign a test address, the daemon sends probes to detect failures on all the interfaces of the multipathing group. “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28 describes the steps you perform to configure test address and groups.

Because `in.mpathd` determines what targets to probe dynamically, you cannot configure the targets. Routers connected to the link are chosen as targets for probing. If no routers exist on the link, arbitrary hosts on the link are chosen. A multicast packet sent to the “all hosts” multicast address (224.0.0.1 in IPv4 and ff02::1 in IPv6) determines the arbitrary hosts. The first few hosts that respond to the echo packets are chosen as targets for probing. If `in.mpathd` cannot find routers or hosts that responded to ICMP echo packets, `in.mpathd` cannot detect failures.

To ensure that each NIC in the group functions properly, `in.mpathd` probes all the targets separately through all the interfaces in the multipathing group. If there are no replies to five consecutive probes, `in.mpathd` considers the interface as having failed.

The probing rate depends on the failure detection time (FDT). The default value for failure detection time is 10 seconds. The `in.mpathd(1M)` man page describes how to change the failure detection time. For a failure detection time of 10 seconds, the probing rate is approximately one probe every two seconds.

If five consecutive probes fail, `in.mpathd` considers the interface as having failed. After a failure is detected, failover of all network access takes place from the failed interface to another functional interface in the group. If you have configured a standby interface, `in.mpathd` chooses the standby interface for failover of IP addresses, broadcasts, and multicast memberships. If you have not configured a standby interface, `in.mpathd` chooses the interface with the least number of IP addresses.

Physical interfaces in the same group that are not present at system boot represent a special case of failure detection. The startup script `/etc/init.d/network` detects these types of failure. This type of failure displays error messages similar to the following:

```
moving addresses from failed IPv4 interfaces: hme0 (moved to hme1)
moving addresses from failed IPv6 interfaces: hme0 (moved to hme1)
```

Currently, this type of failure cannot be automatically repaired by a failback. See “How to Recover a Physical Interface That Was Not Present at System Boot” on page 37.

Detecting Physical Interface Repairs

To detect if a physical interface has been repaired, `in.mpathd` sends probes through the failed NIC. When an interface fails, all addresses are moved to another functional interface in the group. Because `in.mpathd` needs an address for probing so that it can detect repairs, you must configure a test IP address that will not move during the failover. Moreover, you should not allow a normal application to use this test address, because the failover of network access will not take place for these addresses. “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28 describes the steps that you perform. `in.mpathd` considers the interface repaired if it receives responses to 10 consecutive probe packets. Then, if `in.mpathd` detects a failure, failback of all network access takes place to the repaired interface.

As noted in “Detecting Physical Interface Failures” on page 12, automatic failback is not supported for physical interfaces that are not present at system boot. See “How to Recover a Physical Interface That Was Not Present at System Boot” on page 37.

Link Failures

If all the NICs appear to fail at the same time, it is considered a link failure and `in.mpathd` does not do any failovers. This is also true when all the targets fail at the same time. In this case, `in.mpathd` flushes all of its current targets and discovers new targets (see “Detecting Physical Interface Failures” on page 12).

Administering Multipathing Groups with Multiple Physical Interfaces

This section describes how you enable IP Network Multipathing. To use the IP Network Multipathing feature, you should have more than one physical interface connected to the same IP link. For example, the same Ethernet switch or the same IP subnet, configured under the same multipathing group, works. If you have just one physical interface, refer to “Administering Multipathing Groups With a Single Physical Interface” on page 20.

Multipathing groups are identified by non-null names. For example, math-link, bio-link, and chem-link, make good names. The names typically represent where these groups are connected. When failure is detected in one of the network adapters in the multipathing group, all network access is failed over from the failed adapter to the good adapter in the group. The failover of network access includes IPv4 unicast, broadcast, and multicast traffic, as well as IPv6 unicast and multicast traffic. For IP network multipathing to function properly, the following conditions must exist for the network adapters that are part of the same multipathing group:

1. You must push and configure the same set of STREAMS modules on all network adapters in the multipathing group.
2. If you have plumbed IPv4 on one network adapter, then you must plumb IPv4 on all network adapters in the multipathing group.
3. If you have plumbed IPv6 on one network adapter, then you must plumb IPv6 on all network adapters in the multipathing group.
4. All Ethernet network adapters in the system should have unique MAC address in the case of ethernets. This is achieved by setting local-mac-address? to true in the openboot PROM for SPARC platforms. Nothing needs to be done for IA (x86) platforms.
5. All network adapters of the multipathing group must be connected to the same IP link.
6. The multipathing group should not contain dissimilar interfaces. The interfaces that are grouped together should be of the same interface type that is defined in `/usr/include/net/if_types.h`. For example, you cannot combine Ethernet with Token ring, and you cannot combine a Token bus with ATM (asynchronous transfer mode).
7. When you use IP network multipathing with ATMs, you must configure the ATM for LAN emulation (multipathing over classical IP instances is not currently supported).

Note – The fourth condition concerns all interfaces in the system, not just those belonging to the multipathing group.

For the adapters which do not come with factory set unique mac addresses, you can manually configure a mac address for each adapter as a work around. For example, use the `ifconfig ether` command in a start script file.

Note – The mac addresses configured manually will not be maintained across system reboot. You are responsible for choosing unique mac addresses. IP Network Multipathing might behave unpredictably if the mac address of adapters are not unique.

Grouping Physical Interfaces

You use the `ifconfig` command to configure groups. This command uses a new `group` parameter that requires a group name and places both the IPv4 and IPv6 instance of the interface in that group. The `group` parameter has the following syntax:

```
ifconfig interface-name group group-name
```

Note – Avoid using spaces in group names. The `ifconfig` status display does not show spaces. Consequently, if you have created two similar group names, but one of them contains a space, these group names look alike in the status display. However, they are different group names. This might be confusing.

Placing the IPv4 instance under a particular group automatically places the IPv6 instance under the same group. Also, you can place a second interface, connected to the same subnet, in the same group by using the same command (see “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28).

You can remove an interface from a multipathing group by using a null string with the `group` sub-command (see “How to Remove an Interface From a Group” on page 34).

To place an interface in a new group when it is already part of some multipathing group, you do not need to remove it from any existing group. Placing the interface in a new group automatically removes it from any existing group (see “How to Move an Interface From an Existing Group to a Different Group” on page 34).

You can have any number of network adapters that you can configure in the same multipathing group. You cannot use the `group` parameter with logical interfaces. For example, you can use it with `hme0`, but not with `hme0:1`.

You must connect all the interfaces in the multipathing group to the same IP link, because when an interface fails, the failover operation moves all the IP addresses from the failed interface to an interface in the group that is functional. For routers to continue routing packets to the addresses that have been switched to the functional interface, the functional interface must be connected to the same IP link.

Configuring Test Addresses

You must configure all physical interfaces of a multipathing group with a test address. You need test addresses to detect failures and repairs. If a test address is not configured, it is not chosen for failover. Only `in.mpathd` uses test addresses. Normal applications should not use this address. This address will not failover when the interface fails. In IPv4, you should configure the test address in such a way that normal applications do not use the test address (See “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28).

This section describes test address configuration concepts for the following Internet protocols:

- IPv4
- IPv6

IPv4 Test Addresses

The `in.mpathd` multipathing daemon requires a test IP address for detecting failures and repairs. You must use a routeable address for this IP address. That is, the subnet prefix of the address must be known to any routers present on the link. You use the `ifconfig` command's new `-failover` option to configure a test address. Use the following syntax to configure a test address:

```
# ifconfig interface-name addif ip-address <other-parameters> -failover up
```

For `<other-parameters>`, use the parameters required by your configuration. See the `ifconfig(1M)` man page for descriptions. “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28 shows the steps you perform for an IPv4 test address.

For example, to add a new logical interface with an address of `19.16.85.21`, the netmask and broadcast address set to the default value, and also configure the interface with a test address, type the following:

```
# ifconfig hme0 addif 19.16.85.21 netmask + broadcast + -failover up
```

Note – You must mark an IPv4 test address as deprecated to prevent applications from using the test address (see “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28).

Use `failover` without the dash to turn on the failover attribute of the address.

Note – All test IP addresses in a multipathing group must use the same network prefix. That is, the test IP addresses must belong to a single IP subnet.

IPv6 Test Addresses

To configure an IPv6 test address, you use the link-local address itself, because link-local addresses are tied to the physical interface. Thus, you do not need a separate IP address in the IPv6 case. For IPv6, the `-failover` option has the following syntax:

```
# ifconfig interface-name inet6 -failover
```

“How to Configure a Multipathing Interface Group with Two Interfaces” on page 28 shows the steps you perform for an IPv6 test address.

When a multipathing group has both IPv4 and IPv6 plumbed on all the group’s interfaces, you might not need a separate IPv4 test address. The `in.mpathd` daemon can probe the interfaces using an IPv6 link-local address. IPv6 link-local addresses are created when IPv6 is plumbed.

Use `failover` without the dash to turn on the failover attribute of the address.

Note – The only valid IPv6 test address is the link-local address.

Preventing Applications From Using Test Addresses

After you have configured a test address, you need to ensure that this address is not used by normal applications. If you let applications use the test address, applications will fail, because test addresses do not failover during the failover operation. To ensure that IP does not pick the test address for normal applications, you mark the test address deprecated by using the `ifconfig` command. This parameter has the following syntax:

```
ifconfig interface-name deprecated
```

After you mark the address as deprecated, IP does not pick this address as a source address for any communication, unless the applications explicitly bind to the address. Only `in.mpathd` explicitly binds to such an address (see “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28).

Because link-local addresses are not present in the name service (DNS, NIS, and NIS+), applications do not use link-local addresses for communication. Consequently, you do not need to mark IPv6 test addresses as deprecated.

Note – You must not mark IPv6 link-local addresses as deprecated.

Use the `-deprecated` option to turn off the deprecated attribute of the address.

Note – IPv4 test addresses should not be placed in the name service tables (DNS/NIS/NIS+). In IPv6, link-local addresses are used as test addresses and are not normally placed in the name service tables.

Using the `hostname` File to Configure Groups and Test Addresses

You can use the `/etc/hostname.interface` files to configure multipathing groups and test address. To configure a multipathing group using the `/etc/hostname.interface` file, you can add a line to the file using the following syntax:

```
interface-address <parameters> group group-name up \  
addif logical-interface-address <parameters> up
```

For example, to create the group `test` with the following configuration:

- Physical interface `hme0` with address `19.16.85.19`
- A logical interface address of `19.16.85.21`
- With `deprecated` and `-failover` set
- Sets the netmask and broadcast address to the default value

You add the following line to the `/etc/hostname.hme0` file:

```
19.16.85.19 netmask + broadcast + group test up \  
addif 19.16.85.21 deprecated -failover netmask + broadcast + up
```

“How to Configure a Multipathing Interface Group with Two Interfaces” on page 28 shows the steps you perform to configure the IPv4 `hostname` file.

For IPv6 setup, add a line to the `/etc/hostname6.interface` file using the following syntax:

```
<parameter> group group-name up
```

For example, to create a test group for hme0 with an IPv6 test address, add the following line to the `/etc/hostname6.hme0` file:

```
-failover group test up
```

“How to Configure a Multipathing Interface Group with Two Interfaces” on page 28 shows the steps you perform to configure the IPv6 `hostname6` file.

Configuring Standby Interfaces

You can configure multipathing groups with standby interfaces. As the name implies, the interface is considered as standby and is not used unless some other interface in the group fails.

The standby interface is not used to send normal data packets. Consequently, limited traffic flows on a standby interface. You must configure standby interfaces with a test address to insure that probes are sent to determine if the interface is functional. If you do not configure standby interfaces with a test address, the interface is not chosen for failovers when another interface in the group fails. A standby interface might carry traffic under the following conditions:

- If another host on the network communicates with a host using the standby interface address, the standby interface is subsequently used for incoming packets.
- Applications binding (either using `bind` or using `IP_ADD_MEMBERSHIP`) to the address hosted on the standby interface might continue to generate traffic using the standby interface.

Thus, the system does not normally select a standby interface (except for probes), unless it is explicitly chosen by an application. If some interface in the group fails, all network access is failed over to the standby interface. To configure a standby interface, you use the `ifconfig` command's new `standby` parameter using the following syntax:

```
# ifconfig interface-name standby group group-name
```

“How to Configure a Multipathing Group Where One of the Interfaces is a Standby Interface” on page 31 shows the steps you perform.

The `in.mpathd` daemon sends probes on the standby interface once a test address is configured on the standby interface. You should configure only test addresses on a standby interface. If any other address is added on the standby, the addition of these addresses will fail. If standby is marked on an interface which already has addresses other than test addresses, automatic failover of these addresses will take place to a different interface in the group, leaving behind only the test address, if there is one. It is advisable not to configure non-test address on a standby interface.

You need to mark the address as a test address by using the `ifconfig` command's `deprecated` and `-failover` option before setting `standby` or setting up.

To configure a test address on a standby interface, use the following syntax:

```
# ifconfig interface-name plumb ip-address <other-parameters> deprecated -failover standby up
```

For `<other-parameters>`, use the parameters required by your configuration. See the `ifconfig(1M)` man page for descriptions.

Note – Standby interfaces are not used for failover if there is no test address configured on that interface.

For example, to create a test address with the following configuration:

- Physical interface `hme2` as a standby interface
- Address of `19.16.85.22`
- With `deprecated` and `-failover` set
- Sets the netmask and broadcast address to the default value

You type the following command line:

```
# ifconfig hme2 plumb 19.16.85.22 netmask + broadcast + deprecated -failover standby up
```

Note – The interface is marked as a standby interface only after the address is marked as a `NOFAILOVER` address.

“How to Configure a Multipathing Group Where One of the Interfaces is a Standby Interface” on page 31 shows the steps you perform.

You can clear a standby interface using the following syntax:

```
# ifconfig interface-name -standby
```

Administering Multipathing Groups With a Single Physical Interface

When you have only one network adapter in the multipathing group, you can configure the network adapter to detect failures on that NIC alone.

Because failovers cannot occur with only one NIC in the group, you do not need a separate test address on each of the physical interface in the group. You can configure the normal address as an IFF_NOFAILOVER address, which is sufficient for the daemon to send out probes on that interface. Unlike the multiple physical interface case, you do not have to mark it deprecated. Because there is only one interface in the group, addresses can not failover to a different interface. Hence, using this address does not cause any application failures.

Use the following syntax to configure the interface's IPv4 address as a NOFAILOVER:

```
# ifconfig interface-name -failover group group-name
```

For IPv6, use the following syntax:

```
# ifconfig interface-name inet6 -failover group group-name
```

When the daemon detects failures, the interface is marked and logged appropriately on the console.

Note – You cannot verify whether the target being probed has failed or the NIC has failed, because there is only one physical interface through which the target can be probed. If there is only one default router on the subnet, you should turn off multipathing when there is a single physical interface in the group. If a separate IPv4 and IPv6 default router exists (or multiple default routers exist), there is at least more than one target to probe. Hence, it is safe to turn on multipathing.

Removing Network Adapters From Multipathing Groups

When you execute the `ifconfig` command's `group` parameter with a null string, the interface is removed from the existing group (see "How to Remove an Interface From a Group" on page 34). Be careful when removing interfaces from a group. If some other interface in the multipathing group failed, a failover could have happened earlier. For example, if `hme0` failed previously, all addresses are failed over to `hme1` (assuming `hme1` is part of the same group). Removing `hme1` from the group causes `in.mpathd` to move all the failover addresses back to some other interface in the group. If no other interfaces are functioning in the group, failover might not restore all the network accesses.

Similarly, when an interface is part of the group and it needs to be unplumbed, you should remove it from the group first, and then ensure that it has all the original IP addresses configured on it. The `in.mpathd` daemon tries to restore the original

configuration of an interface that is removed from the group. You need to ensure that the configuration is restored before unplumbing the interface. Refer to “Multipathing Daemon” on page 22 to see how interfaces look before and after a failover.

Detached Network Adapters

Dynamic Reconfiguration (DR) uses IP Network Multipathing to decommission a specific network device without impacting existing IP users. Before a NIC is DR-detached (off-lined), all failover IP addresses hosted on that NIC are automatically failed over to another NIC in the same IP Network Multipathing group. The test addresses are brought down and the NIC unplumbed.

Also, with the IP Network Multipathing reboot-safe feature, the static IP addresses in the `/etc/hostname.*` file that are associated with the missing card are hosted automatically on an alternate interface within the same IP Network Multipathing group. However, these addresses will not be moved back to the original interface automatically if the original interface is inserted back into the system at a later time.

Note – NIC attaches and re-attaches require manual intervention.

Multipathing Daemon

The `in.mpathd` multipathing daemon detects failures and repairs by sending out probes on all the interfaces that are part of a group. When an interface is part of a group and has a test address, the daemon starts sending out probes for determining failures on that interface. If there are no replies to five consecutive probes, the interface is considered as having failed. The probing rate depends on the failure detection time. By default, failure detection time is 10 seconds. Thus, the probing rate is one probe every two seconds. To avoid synchronization in the network, probing is not periodic. If five consecutive probes fail, `in.mpathd` considers the interface as failed and performs a failover of the network access from the failed interface to another interface in the group that is functioning properly. If a standby interface is configured, it is chosen for failover of the IP addresses, and broadcasts and multicast memberships. If no standby interface exists, the interface with the least number of IP addresses is chosen. Refer to the man page `in.mpathd(1M)` for more information.

The following two examples show a typical configuration and how the configuration automatically changes when an interface fails. When the `hme0` interface fails, notice that all addresses move from `hme0` to `hme1`.

EXAMPLE 1-1 Interface Configuration Before an Interface Failure

```
hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 19.16.85.19 netmask ffffffff0 broadcast 19.16.85.255
      groupname test
hme0:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500
      index 2 inet 19.16.85.21 netmask ffffffff0 broadcast 129.146.85.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 19.16.85.20 netmask ffffffff0 broadcast 19.16.85.255
      groupname test
hme1:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500
      index 2 inet 19.16.85.22 netmask ffffffff0 broadcast 129.146.85.255
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:19fa/10
      groupname test
hme1: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:1bfc/10
      groupname test
```

EXAMPLE 1-2 Interface Configuration After an Interface Failure

```
hme0: flags=19000842<BROADCAST,RUNNING,MULTICAST,IPv4,NOFAILOVER,FAILED> mtu 0 index 2
      inet 0.0.0.0 netmask 0
      groupname test
hme0:1: flags=19040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,FAILED>
      mtu 1500 index 2 inet 19.16.85.21 netmask ffffffff0 broadcast 129.146.85.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 19.16.85.20 netmask ffffffff0 broadcast 19.16.85.255
      groupname test
hme1:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500
      index 2 inet 19.16.85.22 netmask ffffffff0 broadcast 129.146.85.255
hme1:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 6
      inet 19.16.85.19 netmask ffffffff0 broadcast 19.16.18.255
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER,FAILED> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:19fa/10
      groupname test
hme1: flags=a000841<UP,RUNNING,MULTICAST,IPv6,NOFAILOVER> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:1bfc/10
      groupname test
```

You can see that the FAILED flag is set on hme0 to indicate that hme0 has failed. You can also see that hme1:2 is now created. hme1:2 was originally hme0. The address 19.16.85.19 then becomes accessible through hme1. Multicast memberships associated with 19.16.85.19 can still receive packets, but now through hme1. When the failover of address 19.16.85.19 from hme0 to hme1 took place, a dummy address 0.0.0.0 was created on hme0. The dummy address is removed when a subsequent fallback takes place. The dummy address is created so that hme0 can still be accessed. hme0:1 cannot exist without hme0.

Similarly, failover of the IPv6 address from hme0 to hme1 took place. In IPv6, multicast memberships are associated with interface indexes. They also failover from hme0 to hme1. All the addresses that in.ndpd configures also move (this is not shown in the examples).

The `in.mpathd` daemon continues to probe through the failed NIC, `hme0`. After it receives 10 consecutive replies (for a default failure detection time of 10 seconds), it considers the interface repaired and invokes the failback. After failback, the original configuration is re-established.

See `in.mpathd(1M)` man page for a description of all error messages logged on the console during failures and repairs.

Multipathing Configuration File

The `in.mpathd` daemon uses the settings in the `/etc/default/mpathd` configuration file to invoke multipathing. Changes to this file are read by `in.mpathd` at startup and on `SIGHUP`. This file contains the following default settings and information:

```
#
# Time taken by mpathd to detect a NIC failure in ms. The minimum time
# that can be specified is 100 ms.
#
FAILURE_DETECTION_TIME=10000
#

# Failback is enabled by default. To disable failback turn off this option
#
FAILBACK=yes
#

# By default only interfaces configured as part of multipathing groups
# are tracked. Turn off this option to track all network interfaces
# on the system
#
TRACK_INTERFACES_ONLY_WITH_GROUPS=yes
```

“How to Configure the Multipathing Configuration File” on page 38 shows the steps you perform to configure the `/etc/default/mpathd` configuration file.

Failure Detection Time

You can set a lower value of failure detection time. Sometimes these values might not be achieved if the load on the network is too high. Then `in.mpathd` prints a message on the console, indicating that the time cannot be met. It also prints the time that it can meet currently. If the response comes back correctly, `in.mpathd` meets the failure detection time provided in this file.

Failback

After a failover, failbacks take place when the failed interface is repaired. However, `in.mpathd` does not failback the interface if `FAILBACK` is set to `no`.

As noted in “Detecting Physical Interface Failures” on page 12, automatic failback is not supported for physical interfaces that are not present at system boot. See “How to Recover a Physical Interface That Was Not Present at System Boot” on page 37.

Track Interfaces Only With Groups Option

By turning off this option, `in.mpathd` tracks all interfaces in the system. When a failure is detected, an appropriate message is logged on the console. For this option to function properly, Ethernet addresses on all the interfaces must be unique.

Deploying Network Multipathing

This chapter provides procedures for creating and working with an interface group, configuring test addresses, configuring the `hostname` file, and configuring the multipathing configuration file.

- “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28
- “How to Configure a Multipathing Group Where One of the Interfaces is a Standby Interface” on page 31
- “How to Display the Group to Which a Physical Interface Belongs” on page 33
- “How to Add an Interface To a Group” on page 34
- “How to Remove an Interface From a Group” on page 34
- “How to Move an Interface From an Existing Group to a Different Group” on page 34
- “How to Remove a Physical Interface That Has Failed” on page 35
- “How to Replace a Physical Interface That Has Failed” on page 36
- “How to Recover a Physical Interface That Was Not Present at System Boot” on page 37
- “How to Configure the Multipathing Configuration File” on page 38

Configuring Multipathing Interface Groups

This section provides procedures for configuring multipathing interface groups. It also describes how to make an interface a hot standby interface.

“Grouping Physical Interfaces” on page 15 provides additional information.

Configuring Multipathing Interface Groups—Task Map

TABLE 2-1 Configuring Multipathing Interface Groups—Task Map

Task	Description	For Instructions, Go to ..
Configuring a multipathing interface group with two interfaces.	Use the <code>ifconfig</code> command, the <code>group</code> parameter, <code>-failover</code> option, the deprecated option, and the <code>/etc/hostname.interface</code> file	“How to Configure a Multipathing Interface Group with Two Interfaces” on page 28
Configuring a multipathing group where one of the interfaces is a standby interface	Use the <code>ifconfig</code> command, the <code>group</code> parameter, <code>standby</code> parameter, <code>-failover</code> option, and the <code>/etc/hostname.interface</code> file	“How to Configure a Multipathing Group Where One of the Interfaces is a Standby Interface” on page 31
Displaying the group to which a physical interface belongs	Use the <code>ifconfig</code> command and the interface name	“How to Display the Group to Which a Physical Interface Belongs” on page 33
Adding an interface to a group	Use the <code>ifconfig</code> command and the interface name	“How to Add an Interface To a Group” on page 34
Removing an interface from a group	Use the <code>ifconfig</code> command and a null string to disable IP network multipathing	“How to Remove an Interface From a Group” on page 34
Moving an interface from an existing group to a different group	Use the <code>ifconfig</code> command and the <code>group</code> parameter	“How to Move an Interface From an Existing Group to a Different Group” on page 34

▼ How to Configure a Multipathing Interface Group with Two Interfaces

1. **Become superuser.**
2. **Place each physical interface into a multipathing group by typing the following command.**

```
# ifconfig interface-name group group-name
```

For example, to place `hme0` and `hme1` under group `test`, you type the following commands:

```
# ifconfig hme0 group test
# ifconfig hme1 group test
```
3. **Configure a test address for all the physical interfaces.**
 - a. **For an IPv4 test address, type the following command.**

Note – This step assumes that you have already configured your physical interfaces' addresses.

```
# ifconfig interface-name addif ip-address <parameters> -failover deprecated up
```

For example, to configure a test address on hme0 with the following configuration:

- Address set to 19.16.85.21
- Netmask and broadcast address set to the default value
- -failover and deprecated options set

You type the following command:

```
# ifconfig hme0 addif 19.16.85.21 netmask + broadcast + -failover deprecated up
```

You can check the configuration by typing the following:

```
# ifconfig hme0:1
hme0:1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER> mtu 1500
index 2 inet 19.16.85.21 netmask ffffffff broadcast 19.16.85.255
```

Note – You must mark an IPv4 test address as deprecated to prevent applications from using the test address.

To configure a test address on hme1 with the following configuration:

- Address set to 19.16.85.22
- Netmask and broadcast address set to the default value
- -failover and deprecated options set

Type the following command:

```
# ifconfig hme1 addif 19.16.85.22 netmask + broadcast + -failover deprecated up
```

b. For an IPv6 test address, type the following command.

```
# ifconfig interface-name inet6 -failover
```

Note – Because you have already placed the physical interfaces with IPv4 addresses into a multipathing group, physical interfaces with IPv6 addresses are also implicitly placed in the same multipathing group. If you had placed physical interfaces with IPv6 addresses into a multipathing group first, then physical interfaces with IPv4 addresses would have been also implicitly placed in the same multipathing group.

For example, to configure hme0 with an IPv6 test address, you type the following command:

```
# ifconfig hme0 inet6 -failover
```

You can check the configuration by typing the following:

```
# ifconfig hme0 inet6
  hme0: flags=a000841<UP, RUNNING, MULTICAST, IPv6, NOFAILOVER> mtu 1500 index 2
        inet6 fe80::a00:20ff:feb9:17fa/10
        groupname test
```

Note – You do not need to mark an IPv6 test address as deprecated to prevent applications from using the test address.

For the second interface, hme1, type the following command:

```
# ifconfig hme1 inet6 -failover
```

4. (You do this step only if you want to preserve the configuration across reboots.) To preserve the configuration across reboots, do the following substeps.

- a. For IPv4, edit the `/etc/hostname.interface` file and add the following line.

```
interface-address <parameters> group group-name up \  
  addif logical-interface -failover deprecated <parameters> up
```

Note – This test IP address is configured only on the next reboot. If you want the configuration invoked in the current session, do steps 1, 2, and 3.

For example, to create a group `test` with the following configuration for hme0:

- Physical interface hme0 with address 19.16.85.19
- A logical interface address of 19.16.85.21
- With `deprecated` and `-failover` set
- Sets the netmask and broadcast address to the default value

You add the following line to the `/etc/hostname.hme0` file:

```
19.16.85.19 netmask + broadcast + group test up \  
  addif 19.16.85.21 deprecated -failover netmask + broadcast + up
```

Similarly, to place hme1 under the same group `test` and configure a test address, type the following command:

```
19.16.85.20 netmask + broadcast + group test up \  
  addif 19.16.85.22 deprecated -failover netmask + broadcast + up
```

- b. For IPv6, edit the `/etc/hostname6.interface` file and add the following line.

```
-failover group group-name up
```

Note – This test IP address is configured only on the next reboot. If you want the configuration invoked in the current session, do steps 1, 2, and 3.

For example, to create a test group for hme0 with an IPv6 address, add the following line to the `/etc/hostname6.hme0` file:

```
-failover group test up
```

Similarly, to place hme1 under the same group test and configure a test address, add the following line to the `/etc/hostname6.hme1` file:

```
-failover group test up
```

Note – To add more interfaces to the multipathing group, repeat steps 1 through 3. New interfaces can be added to an existing group on a live system. However, changes are lost across reboots.

▼ How to Configure a Multipathing Group Where One of the Interfaces is a Standby Interface

The examples used in this procedure assume that hme1 will be configured as the standby interface.

Note – A standby interface has only a test address.

1. Do steps 1 and 2 in “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28.
2. Configure test address on all physical interfaces using the following substeps.
 - a. For the non-standby interface, such as hme0, do step 3 in “How to Configure a Multipathing Interface Group with Two Interfaces” on page 28.
 - b. For a standby interface, configure a test address by typing the following command.

Note – A standby interface can have only a test address. A standby interface cannot have any other IP address.

```
# ifconfig interface-name plumb ip-address <other-parameters> deprecated -failover standby up
```

Note – You must set the `-failover` option before the `standby` option and the `standby` option before `up`.

For <other-parameters>, use the parameters required by your configuration. See the `ifconfig(1M)` man page for descriptions.

For example, to create a test address with the following configuration:

- Physical interface `hme1` as a standby interface
- Address of `19.16.85.22`
- With `deprecated` and `-failover` set
- Sets the netmask and broadcast address to the default value

You type the following command:

```
# ifconfig hme1 plumb 19.16.85.22 netmask + broadcast + deprecated -failover standby up
```

You can check the results by typing the following:

```
# ifconfig hme1
flags=69040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,STANDBY,INACTIVE>
mtu 1500 index 4 inet 19.16.85.22 netmask ffffffff broadcast 19.16.85.255
groupname test
```

For IPv6, to create a test address, type the following command:

```
ifconfig hme1 plumb -failover standby up
```

The `INACTIVE` flag indicates that this interface is not used for any outbound packets. When a failover takes place to this standby interface, the `INACTIVE` flag is cleared.

3. (You do this step only if you want to preserve the configuration across reboots.) To preserve the configuration across reboots, do the following substeps.

a. For IPv4, edit the `/etc/hostname.interface` file and add the following line.

```
interface-address <parameters> group group-name up \  
addif logical-interface-failover deprecated <parameters> up
```

Note – This test IP address is configured only on the next reboot. If you want the configuration invoked in the current session, do steps 1 and 2.

For example, to create a group `test` with the following configuration for `hme0`:

- Physical interface `hme0` with address `19.16.85.19`
- A logical interface address of `19.16.85.21`
- With `deprecated` and `-failover` set
- Sets the netmask and broadcast address to the default value

You add the following line to the `/etc/hostname.hme0` file:


```
19.16.85.19 netmask + broadcast + group test up \  
    addif 19.16.85.21 deprecated -failover netmask + broadcast + up
```

Similarly, to place the standby interface hme1 under the same group test and configure a test address, type the following command:

```
19.16.85.22 netmask + broadcast + deprecated + group test -failover standby up
```

b. For IPv6, edit the `/etc/hostname6.interface` file and add the following line.

```
-failover group group-name up
```

Note – This test IP address is configured only on the next reboot. If you want the configuration invoked in the current session, do steps 1 and 2.

For example, to create a test group for hme0 with an IPv6 address, add the following line to the `/etc/hostname6.hme0` file:

```
-failover group test up
```

Similarly, to place the standby interface hme1 under the same group test and configure a test address, add the following line to the `/etc/hostname6.hme1` file:

```
-failover group test standby up
```

▼ How to Display the Group to Which a Physical Interface Belongs

1. Become superuser.

2. On a command line, type the following command.

```
# ifconfig interface-name
```

For example, to display the group name for hme0, you type the following command:

```
# ifconfig hme0  
hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2  
    inet 19.16.85.19 netmask ffffffff broadcast 19.16.85.255  
    groupname test
```

To display the group name for only the IPv6 instance, you type the following command:

```
# ifconfig hme0 inet6  
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2  
    inet6 fe80::a00:20ff:feb9:19fa/10  
    groupname test
```

▼ How to Add an Interface To a Group

1. Become superuser.
2. On a command line, type the following command.

```
# ifconfig interface-name group group-name
```

For example, to add hme0 to the group test, you type the following command:

```
# ifconfig hme0 group test
```

▼ How to Remove an Interface From a Group

1. Become superuser.
2. On a command line, type the following command.

```
# ifconfig interface-name group ""
```

The quotation marks indicate a null string.

For example, to remove hme0 from the group test, you type the following command:

```
# ifconfig hme0 group ""
# ifconfig hme0
hme0: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 19.16.85.19 netmask ffffffff broadcast 19.16.85.255
# ifconfig hme0 inet6
hme0: flags=a000841<UP,RUNNING,MULTICAST,IPv6> mtu 1500 index 2
      inet6 fe80::a00:20ff:feb9:19fa/10
```

“Removing Network Adapters From Multipathing Groups” on page 21 provides additional information.

▼ How to Move an Interface From an Existing Group to a Different Group

1. Become superuser.
2. On a command line, type the following command.

```
# ifconfig interface-name group group-name
```

Note – Placing the interface in a new group automatically removes it from any existing group.

For example, to remove hme0 from group test and place it in group cs-link, you type the following:

```
# ifconfig hme0 group cs-link
```

This removes the interface from any existing group and then puts the interface in the group cs-link.

Replacing a Physical Interface That Has Failed

You must do the following manual steps before replacing a physical interface that has failed. The following procedures use physical interfaces hme0 and hme1 as example interfaces. The procedures assume that both interfaces are in a multipathing group and that hme0 has failed. The procedures also assume that the logical interface hme0:1 has the test address.

Note – These procedures assume that you are replacing the failed interface with the same physical interface name (for example, hme0 with hme0).

▼ How to Remove a Physical Interface That Has Failed

Note – You can skip step 1 if the test address is plumbed using the /etc/hostname.hme0 file.

1. Retrieve the test address configuration by typing the following command.

```
# ifconfig hme0:1

hme0:1:
flags=9040842<BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3
inet 129.146.233.250 netmask ffffffff broadcast 129.146.233.255
```

You need this information to replumb the test address when replacing the physical interface.

See “Using the hostname File to Configure Groups and Test Addresses” on page 18 for details on how to configure test addresses using the hostname file.

2. Refer to the `cfgadm(1M)` man page, *Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide*, or *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* for a description of how to remove the physical interface.

▼ How to Replace a Physical Interface That Has Failed

1. Refer to the `cfgadm(1M)` man page, *Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems Dynamic Reconfiguration User's Guide*, or *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* for a description of how to replace the physical interface.
2. Plumb in and bring up the test address by typing the following command.

```
# ifconfig hme0 <test address configuration>
```

Note – The test address configuration is the same as that configured in the `/etc/hostname.hme0` file. Using the previous procedure, the test configuration is the same configuration that is displayed in step 1.

This triggers the `in.mpathd` daemon to resume probing. As a result of this probing, `in.mpathd` will detect the repair. Consequently, `in.mpathd` causes the original IP address to failback from `hme1`.

See “Configuring Test Addresses” on page 16 for more details about how to configure test addresses.

Recovering a Physical Interface That Was Not Present at System Boot

You must do the following manual steps before recovering a physical interface that was not present at system boot. The following procedure uses physical interfaces `hme0` and `hme1` as example interfaces. The procedure assumes that both interfaces are in a multipathing group and that `hme0` was not present at system boot.

▼ How to Recover a Physical Interface That Was Not Present at System Boot

1. **Retrieve the failed network information from the console log failure error message.**

See the `syslogd(3C)` man page. The error message might be similar to the following message:

```
moving addresses from failed IPv4 interfaces:hme1 (moved to hme0)
```

The error message might also be similar to the following message:

```
moving addresses from failed IPv4 interfaces:hme1 (couldn't move, no alternative interfa
```

2. **Attach the physical interface to the system.**

Refer to the `cfgadm(1M)` man page, Sun Enterprise 10000, 6x00, 5x00, 4x00 or 3x00 Dynamic Reconfiguration User's Guide for a description of how to replace the physical interface.

3. **Refer to the message content from step 1. If the addresses could not be moved, go to step 5. If the addresses were moved, do step 4.**

4. **Unplumb the logical interfaces configured as part of the failover process by doing the following substeps.**

- a. **Look at the contents of the file `/etc/hostname.<moved_from_interface>` to see what logical interfaces were configured as part of the failover process.**

- b. **Unplumb each failover IP address by typing the following command:**

```
# ifconfig moved_to_interface removeif moved_ip_address
```

Note – Failover addresses are those addresses marked with the `failover` parameter, or those addresses not marked with the `-failover` parameter. You do not need to unplumb IP addresses marked `-failover`.

For example, assume that the contents of the `/etc/hostname.hme0` file contained the following lines:

```
inet 1.2.3.4 -failover up group one
addif 1.2.3.5 failover up
addif 1.2.3.6 failover up
```

Then, to unplumb each failover IP address, you would type the following commands:

```
# ifconfig hme0 removeif 1.2.3.5
# ifconfig hme0 removeif 1.2.3.6
```

5. **Reconfigure the IPv4 information for the replaced physical interface by typing the following command for each interface that was removed.**

```
# ifconfig removed_from_NIC <parameters>
```

By using the example in step 4, you would enter the following commands:

```
# ifconfig hme1 inet plumb
# ifconfig hme1 inet 1.2.3.4 -failover up group one
# ifconfig hme1 addif 1.2.3.5 failover up
# ifconfig hme1 addif 1.2.3.6 failover up
```

Configuring the Multipathing Configuration File

The multipathing `/etc/default/mpathd` configuration file contains three parameters that you can adjust for your configuration requirements:

- `FAILURE_DETECTION_TIME`
- `FAILBACK`
- `TRACK_INTERFACES_ONLY_WITH_GROUPS`

See “Multipathing Configuration File” on page 24 for a description of these parameters.

▼ How to Configure the Multipathing Configuration File

1. **Become superuser.**
2. **Edit the `/etc/default/mpathd` and change the default value of one or more of the three parameters using one or more of the following sub-steps.**

- a. **Type the new value for the `FAILURE_DETECTION_TIME` parameter.**

```
FAILURE_DETECTION_TIME=n
```

- b. **Type the new value for the `FAILBACK` parameter.**

```
FAILBACK=[yes | no]
```

- c. **Type the new value for the `TRACK_INTERFACES_ONLY_WITH_GROUPS` parameter.**

```
TRACK_INTERFACES_ONLY_WITH_GROUPS=[yes | no]
```

3. On a command line, type the following command.

```
# kill -HUP in.mpathd
```


Glossary

This glossary contains only definitions of new terms found in this book and are not in the Global Glossary. For definitions of other terms, see the Global Glossary at <http://docs.sun.com:80/ab2/coll.417.1/GLOBALGLOSS/@Ab2TocView>.

failback	The process of switching back network access to an interface detected as having been repaired.
failover	The process of switching network access from a failed interface to a good physical interface. Network access includes IPv4 unicast, multicast, and broadcast traffic, as well as IPv6 unicast and multicast traffic.
failure detection	The process of detecting when a NIC or the path from the NIC to some layer 3 device starts operating correctly after a failure.
IP link	A communication facility or medium over which nodes can communicate at the link layer. The link layer is the layer immediately below IPv4/IPv6. Examples include Ethernets (simple or bridged) or ATM networks. One or more IPv4 subnet numbers/prefixes are assigned to an IP link. A subnet number/prefix can not be assigned to more than one IP link. In ATM LANE, an IP link is a single emulated LAN. When using ARP, the scope of the ARP protocol is a single IP link.
Network Interface Card (NIC)	Network adapter that is either internal or a separate card that serves as an interface to a link.
physical interface	A node's attachment to a link. This attachment is often implemented as a device driver plus a network adapter. Some network adapters can have multiple points of attachment, for example, qfe. The usage of <i>Network adapter</i> in this document refers to a "Single Point of Attachment."
physical interface group	The set of physical interfaces on a system that are connected to the same link. They are identified by assigning the same (non-null) character string name to all the physical interfaces in the group.

physical interface group name	A name assigned to a physical interface that identifies the group. The name is local to a system. Multiple physical interfaces, sharing the same group name, form a physical interface group.
repair detection	The process of detecting when a NIC or the path from the NIC to some layer 3 device starts operating correctly after a failure.
standby	A physical interface that is not used to carry data traffic unless some other physical interface has failed.

Index

A

ATM, multipathing, 14

C

communication failures, 10
components, multipathing, 11

D

deprecated attribute, ifconfig command, 17
deprecated parameter, IPv4 test address, 29
detecting physical interface failure, 12
detecting physical interface repairs, 13
dynamic reconfiguration, multipathing, 22

E

/etc/default/mpathd file, 24, 38
Ethernet, multipathing, 14

F

failback, 10, 11, 25
failover, 10, 11, 14
 examples, 22
 standby interface, 19
failover option, ifconfig command, 16
failover success conditions, 14

failure detection

 definition, 10
 multipathing, 11
 test addresses, 16

failure detection time, 24

 multipathing, 22

failures, communication, 10

G

group names, multipathing, 14

group parameter

 ifconfig command, 15, 21, 28, 35
 tracking interfaces, 25

grouping physical interfaces, multipathing, 15

H

hostname file

 configuring groups and test addresses, 18
 multipathing, 30, 32

I

ifconfig command

 deprecated attribute, 17
 displaying multipathing group, 33
 failover option, 16
 group parameter, 15, 21, 28, 35
 multipathing groups, 15

- ifconfig command (Continued)
 - standby parameter, 19, 32
 - test parameter, 28
- ifconfig ether command, multipathing, 15
- in.mpathd daemon, 12
 - failback, 25
 - failure detection time, 24
 - multipathing, 22
 - probing rate, 22
 - probing targets, 12
 - standby interface, 19
- IP link, multipathing, 11
- IP network multipathing, *See* multipathing
- IPv4 test address
 - configuring, 16, 28
 - deprecated parameter, 29
- IPv6 link-local address, multipathing, 17
- IPv6 test address
 - configuring, 17, 29

L

- link failures, multipathing, 13
- link-local address
 - IPv6 test address, 18
 - multipathing, 18
- load spreading, definition, 10

M

- mac addresses, multipathing, 15
- multipathing
 - adding an interface from a group, 34
 - ATM, 14
 - components, 11
 - configuration file, 24
 - configuring a group with a hot standby interface, 31
 - configuring a standby interface, 31
 - configuring configuration file, 38
 - configuring interface group, 27
 - configuring IPv6 test address, 29
 - configuring test addresses, 16
 - creating a test group, 30
 - detached network adapters, 22
 - display group name, 33

- multipathing (Continued)
 - display groups, 33
 - DR-detached, 22
 - dynamic reconfiguration, 22
 - enabling, 14
 - Ethernet, 14
 - failure detection, 10, 11
 - failure detection time, 24
 - features, 10
 - group names, 14
 - groups with multiple interfaces, 14
 - hostname file, 18, 30, 32
 - ifconfig command, 15
 - ifconfig ether command, 15
 - IP link, 11
 - IPv4
 - creating a test group, 32
 - placing a standby interface in a group, 33
 - IPv6
 - creating a test group, 33
 - placing a standby interface in a group, 33
 - link failures, 13
 - link-local address, 18
 - load spreading, 10
 - mac addresses, 15
 - moving interfaces from groups, 34
 - network interface, 11
 - physical interface, 11
 - physical interface group, 11
 - physical interface group name, 11
 - placing an interface in a test group, 31
 - preserving configuration across reboots, 30, 32
 - reboot-safe, 22
 - recovering a physical interface not present at system boot, 36
 - removing adapters from groups, 21
 - removing an interface from a group, 34
 - removing failed physical interface, 35
 - repair detection, 10, 11
 - replacing failed physical interface, 35, 36
 - standby interface, 12
 - standby interface and group, 19
 - Token ring, 14
 - tracking interfaces, 25
 - when to turn off, 21
- multipathing daemon, 22

multipathing groups, administering with a single interface, 20
multipathing interface group, configuring with two interfaces, 28

N

/net/if_types.h file, 14
network interface, multipathing, 11

P

physical interface, multipathing, 11
physical interface failures, detecting, 12
physical interface group, multipathing, 11
physical interface group name, multipathing, 11
physical interface repairs, detecting, 13
probing targets, in.mpathd daemon, 12

R

reboot-safe, multipathing, 22
repair detection
 definition, 10
 multipathing, 11
 test addresses, 16

S

standby interface
 clearing, 20
 configuring, 19
 configuring a multipathing group, 31
 configuring test address on, 32
 multipathing, 12
 test addresses, 19
standby parameter
 ifconfig command, 19, 32

T

test address
 configuring, 16
 configuring on a standby interface, 32
 IPv4 and IPv6, 16
 standby interface, 19
test addresses, preventing applications from using, 17
test parameter, ifconfig command, 28
Token ring, multipathing, 14

