IP Network Multipathing Administration Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 816–5249–10
December 2002

Adobe PostScript

020912@4660

Contents

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™ 5.9 operating system, and you have set up any networking software that you plan to use. The SunOS 5.9 operating system is part of the Solaris product family, which includes many features, such as the Solaris Common Desktop Environment (CDE). The SunOS 5.9 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 9 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 addresses. This chapter also provides other useful procedures related to IP Network Multipathing.

Glossary provides definitions of key IP Network Multipathing terms.

Accessing Sun Documentation Online

The docs.sun.com[SM] 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 `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.

TABLE P–1 Typographic Conventions *(Continued)*

Typeface or Symbol	Meaning	Example
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* 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	`#`

CHAPTER 1

IP Network Multipathing (Overview)

IP network multipathing provides failover when you have multiple network interface cards that are connected to the same IP link, for example, Ethernet. IP network multipathing also provides load spreading.

This chapter contains the following information:

Introduction

IP network multipathing provides your system with the following capabilities:

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

To use IP network multipathing, you must have an alternate adapter connected to the same IP link. Consequently, if a failure occurs in the network adapter, 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 that are connected to the same IP link, you achieve increased traffic throughput. The traffic is spread 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. Also, you can 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 automatic switching (*failover*) of the network access to an alternate network adapter. Failure detection can only occur when 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. The network access is switched back automatically (*failback*) to an alternate network adapter. Repair detection assumes that you have enabled failbacks. See "Detecting Physical Interface Repairs" on page 14 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 that use multiple connections.

Communication Failures

Communication failures can occur in the following ways:

1. Transmit or 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 or 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 compose 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 following IPv4 or IPv6. Examples include Ethernets, simple or bridged, or ATM networks. One or more IPv4 subnet numbers or prefixes are assigned to an IP link. A subnet number or prefix cannot 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 *Network adapter* in this document refers to a "Single Point of Attachment."
Physical interface group	The set of physical interfaces on a system that is connected to the same link. The set is identified by assigning the same (non-null) character string name to all the physical interfaces in the group.
Physical interface group name	A name that is 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 a layer 3 device no longer works.

TABLE 1–1 IP Network Multipathing Components *(Continued)*

Component	Description
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 that is detected to have been repaired.
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`(1M)
- `ip`(7P)

The `in.mpathd` daemon detects failures. Then, the daemon 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. The Internet protocol, which implements the ioctl, does the network access failover transparently and automatically.

Caution – Do not use Alternate Pathing while using IP network multipathing on the same set of NICs. Likewise, you should not use IP network multipathing while you are 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 can detect interface failure and repair by two methods. In the first method, the daemon sends and receives ICMP echo probes through the interface. In the second method, the daemon monitors the RUNNING flag on the interface. The link state on some models of network interface cards is reflected by the

RUNNING flag. Consequently, when the link fails, the failure is detected much sooner. An interface is considered to have failed if either of the previous two methods indicate failure. An interface is considered repaired only if both methods indicate that the interface is repaired.

The `in.mpathd` daemon sends ICMP echo probes to the targets that are connected to the link on all the interfaces. The interfaces must belong to a group to detect failures and repair. After you add an interface to a multipathing group, assign a test address. Then, 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 that you perform to configure test address and groups.

Because `in.mpathd` determines which targets to probe dynamically, you cannot configure the targets. Routers that are 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 that is 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 no replies are made to five consecutive probes, `in.mpathd` considers the interface to have 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. Failback occurs after a repair is detected. The actual time to detect an interface failure can take from 20 seconds to a few minutes. The time depends on the system and network load.

The failure detection time only applies to the ICMP echo probe method of detecting failures. If link failure clears the RUNNING flag for an interface, the `in.mpathd` daemon responds immediately to the change in the flag status.

After a failure is detected, failover of all network access occurs 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 instance of failure detection. The startup script `/etc/init.d/network` detects these types of failure. Error messages that are similar to the following messages are displayed:

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

Note – In this special instance of failure detection, only static IP addresses are moved to a different physical interface. The addresses must be specified in the host name file. The physical interface must be in the same multipathing group.

This type of failure can be automatically repaired by a failback. The RCM DR Post-attach feature for IP network multipathing automates the DR attachment of a NIC. When a NIC is DR attached, the interface is plumbed and configured. If the interface was removed prior to a reboot, the IP multipathing Reboot-safe feature recovers the IP address. The IP address is transferred to the replaced NIC. The replaced NIC is added to the original IP multipathing interface group. See "How to Recover a Physical Interface That Was Not Present at System Boot" on page 38.

Detecting Physical Interface Repairs

The `in.mpathd` daemon considers an interface is repaired if the daemon receives responses to 10 consecutive probe packets. Also, the RUNNING flag must be set on the interface.

When an interface fails, all addresses are moved to another functional interface in the group. `in.mpathd` needs an address for probing so that `in.mpathd` can detect repairs. Consequently, you must configure a test IP address that cannot move during the failover. Moreover, you should not allow a normal application to use this test address, because the failover of network access cannot occur for these addresses. "How to Configure a Multipathing Interface Group With Two Interfaces" on page 28 describes the steps that you perform. If `in.mpathd` detects a repair, failback of all network access occurs to the repaired interface.

As noted in "Detecting Physical Interface Failures" on page 12, automatic failback is 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 38.

Group Failures

A group failure is when all the network interface cards appear to fail at the same time. `in.mpathd` does not do any failovers for a group failure. Also, no failover occurs when all the targets fail at the same time. In this instance, `in.mpathd` flushes all of its current targets and discovers new targets. See "Detecting Physical Interface Failures" on page 12.

Note – Group failures were previously known as link failures.

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, you can configure the same Ethernet switch or the same IP subnet under the same multipathing group. If you have just one physical interface, refer to "Administering Multipathing Groups With a Single Physical Interface" on page 22.

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. Network access switches from a failed adapter to a working adapter in the multipathing group when a failure is detected. 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 addresses. To achieve unique MAC addresses, set the 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 asynchronous transfer mode (ATM).
7. When you use IP network multipathing with an ATM, 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 that do not come with factory-set unique MAC addresses, you can manually configure a MAC address for each adapter as a workaround. For example, use the `ifconfig ether` command in a startup script file.

Note – The MAC addresses that are configured manually cannot be maintained across system reboot. You are responsible for choosing unique MAC addresses. IP network multipathing might behave unpredictably if the MAC addresses 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. The `ifconfig` command places both the IPv4 and IPv6 instances 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, do not create two similar group names. If one of the group names contains a space, these group names look the same in the status display. However, the group names are different. This difference 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, which is 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.

You can place an interface in a new group when the interface is already part of some multipathing group. You do not need to remove the interface from any existing group. By placing the interface in a new group, the interface is automatically removed from any existing group. See "How to Move an Interface From an Existing Group to a Different Group" on page 35.

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 the parameter with `hme0`, but not with `hme0:1`.

You must connect all the interfaces in the multipathing group to the same IP link. When an interface fails, the failover operation moves all the IP addresses from the interface that has failed to a functional interface in the group. The functional interface must be connected to the same IP link. Consequently, routers can continue to rout packets to the addresses that have been switched to the functional interface.

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, the test address is not chosen for failover. Only `in.mpathd` uses test addresses. Normal applications should not use this address. This address is not chosen for 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 the concepts of test address configuration 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. 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 that are 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 that you perform for an IPv4 test address.

For example, if you wanted to do the following configuration:

- Add a new logical interface with an address of `19.16.85.21`
- Have the netmask and broadcast address set to the default value
- Configure the interface with a test address

Type the following command:

```
# 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. 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, because link-local addresses are tied to the physical interface. Thus, you do not need a separate IP address in the IPv6 situation. 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 that 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 by 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 fail, because test addresses do not fail over 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`

IP does not pick a deprecated 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, and NIS+. In IPv6, link-local addresses are used as test addresses. In IPv6, link-local addresses are not normally placed in the name service tables.

Autoconfigured IPv6 addresses are not preserved across system reboot. If you require that IP addresses be preserved across reboot, then applications should use static IP addresses.

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

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

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

For example, if you want 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 that you perform to configure the IPv4 hostname file.

For IPv6 setup, add a line to the /etc/hostname6.*interface* file by 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
addif 1080::56:a00:20ff:feb9:19fa up
```

"How to Configure a Multipathing Interface Group With Two Interfaces" on page 28 shows the steps that 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 a standby interface. The standby interface is not used unless some other interface in the group fails. A standby interface has an IFF_INACTIVE flag when the interface is not hosting any failover IP address. Consequently, when an active interface fails, the standby interface is always chosen for failover. After the standby interface is chosen, the IFF_INACTIVE flag is cleared on the standby interface. From that instant, the active standby interface is treated the same as other active interfaces. Some failures might not choose a standby interface. Instead, some failures might choose an active interface that hosts fewer IP addresses than the standby interface.

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 ensure 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 by using the standby interface address, the standby interface is subsequently used for incoming packets.
- Applications can bind to an address by using bind or IP_ADD_MEMBERSHIP. Applications that bind to the address that is hosted on the standby interface might continue to generate traffic by using the standby interface.

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

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

"How to Configure a Multipathing Group With One of the Interfaces a Standby Interface" on page 31 shows the steps that you perform.

The in.mpathd daemon sends probes on the standby interface after 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 this address fails. If a standby interface already has addresses other than test addresses, automatic failover of these addresses occurs to a different interface in the group. Only the test address remains, if a test address exists. Avoid configuring 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 options 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 that are required by your configuration. See the ifconfig(1M) man page for descriptions.

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

For example, if you want 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 With One of the Interfaces a Standby Interface" on page 31 shows the steps that you perform.

You can clear a standby interface by 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, failover is not possible. However, you can sill use a multipathing group for failure detection on that NIC.

In the case of a single adapter in a multipathing group, you do not have to configure a dedicated test IP address for failure detections. You can use a single IP address for sending data and detecting failure. If you choose to configure an IFF_NOFAILOVER address for a single adapter in a multipathing group, `in.mpathd` sends probe packets by using that address. Otherwise, `in.mpathd` picks a data address to send probe packets. Unlike the multiple physical interface instance, you should not mark a single physical interface as deprecated.

For single adapter groups, use the following syntax to configure an IPv4 address:

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

You can also use the following syntax:

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

For IPv6, use the following syntax:

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

You can also use the following syntax:

```
# ifconfig interface-name inet6 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 that is being probed has failed or the NIC has failed. The target can be probed through only one physical interface. If only one default router is on the subnet, turn off multipathing if a single physical interface is in the group. If a separate IPv4 and IPv6 default router exists, or multiple default routers exist, more than one target needs to be probed. Hence, you can safely 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, if hme1 is part of the same group. The removal of hme1 from the group causes in.mpathd to return all the failover addresses 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 the interface needs to be unplumbed, you should remove the interface from the group first. Then, ensure that the interface has all the original IP addresses configured on the interface. 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 24 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 are failed over. These addresses, that are hosted on that NIC, are automatically failed over to another NIC in the same IP network multipathing group. The test addresses are brought down. Then, the NIC is unplumbed.

With the IP network multipathing reboot-safe feature, the static IP addresses are hosted automatically on an alternate interface. The interface must be within the same IP network multipathing group. The static IP addresses are the addresses in the /etc/hostname.* file that are associated with the missing card. However, these addresses are returned to the original interface automatically when the original interface is inserted back into the system at a later time.

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. The in.mpathd multipathing daemon also detects failures and repairs by monitoring the RUNNING flag on each interface in the group. When an interface is part of a group, and has a test address, the daemon starts sending out probes to determine failures on that interface. If the daemon does not receive any replies to five consecutive probes, the interface is considered to have failed. Or, if the RUNNING flag is not set, the interface is considered to have 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 determines that the interface has failed. The daemon performs a failover of the network access from the failed interface to another functional interface in the group. If a standby interface is configured, the interface 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. The following two examples also show 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 ffffff00 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 ffffff00 broadcast 129.146.85.255
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
     inet 19.16.85.20 netmask ffffff00 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 ffffff00 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 ffffff00 broadcast 129.146.85.255
```

```
hme1: flags=9000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 19.16.85.20 netmask ffffff00 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 ffffff00 broadcast 129.146.85.255
hme1:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 6
        inet 19.16.85.19 netmask ffffff00 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 that are 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` occurred, a dummy address `0.0.0.0` was created on `hme0`. The dummy address is removed when a subsequent failback 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` occurred. In IPv6, multicast memberships are associated with interface indexes. Multicast memberships also fail over from `hme0` to `hme1`. All the addresses that `in.ndpd` configures also move. This action is not shown in the examples.

The `in.mpathd` daemon continues to probe through the failed NIC, `hme0`. After the daemon receives 10 consecutive replies for a default failure detection time of 10 seconds, the daemon determines that the interface is repaired. Then, the daemon invokes the failback. After failback, the original configuration is reestablished.

See `in.mpathd`(1M) man page for a description of all error messages that are 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 40 shows the steps that 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. The message indicates that the time cannot be met. The daemon also prints the time that the daemon can meet currently. If the response comes back correctly, `in.mpathd` meets the failure detection time that is provided in this file.

Failback

After a failover, failback occurs when the failed interface is repaired. However, `in.mpathd` does not fail back the interface if `FAILBACK` is set to `no`.

As noted in "Detecting Physical Interface Failures" on page 12, automatic failback is 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 38.

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.

CHAPTER 2

Administering Network Multipathing (Task)

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.

This chapter contains the following information:

Configuring Multipathing Interface Groups

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

"Grouping Physical Interfaces" on page 16 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 `ifconfig` command, the `group` parameter, `-failover` option, the `deprecated` option, and the `/etc/hostname.`*interface* 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 `ifconfig` command, the `group` parameter, standby parameter, `-failover` option, and the `/etc/hostname.`*interface* file	"How to Configure a Multipathing Group With One of the Interfaces a Standby Interface" on page 31
Displaying the group to which a physical interface belongs	Use the `ifconfig` 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 `ifconfig` command and the interface name	"How to Add an Interface To a Group" on page 34
Removing an interface from a group	Use the `ifconfig` 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 `ifconfig` command and the `group` parameter	"How to Move an Interface From an Existing Group to a Different Group" on page 35

▼ 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.**
   - 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, if you wanted 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 ffffff00 broadcast 19.16.85.255
```

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

For example, if you want 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
```

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

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

Note – Physical interfaces with IPv6 addresses are implicitly placed in the same multipathing group as the IPv4 addresses. This result occurs by placing the physical interface with IPv4 addresses into a multipathing group. You might have 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. **Do this step only if you want to preserve the configuration across reboots. To preserve the configuration across reboots, do the following substeps.**

   - 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 to be invoked in the current session, do steps 1, 2, and 3.

   For example, if you want 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
   ```

   - 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 to be 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 With One of the Interfaces a Standby Interface

The examples that are used in this procedure assume that `hme1` is 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 the test address on all physical interfaces by 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 that are required by your configuration. See the `ifconfig`(1M) man page for descriptions.

For example, if you want 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 ffffff00 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 occurs on this standby interface, the `INACTIVE` flag is cleared.

3. **Do this step only if you want to preserve the configuration across reboots. To preserve the configuration across reboots, do the following substeps.**

   - 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 to be invoked in the current session, do steps 1 and 2.

For example, if you want 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
```

- 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 to be 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 ffffff00 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 ffffff00 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 23 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 the interface from any existing group.

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

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

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

Replacing a Physical Interface That Has Failed, or DR-detaching or DR-attaching a Physical Interface

The steps in this section pertain to only IP layers that are configured by using `ifconfig`(1M). Layers before or after the IP layer, such as ATM or other services, require specific manual steps if the layers are not automated. The specific steps are used to unconfigure during pre-detach and configure after post-attach. See the layers and applications documentation for instructions on how to handle the failure and DR scenarios.

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 by 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 ffffff00 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 19 for details on how to configure test addresses by 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 DR Configuration 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 DR Configuration Guide*, **or** *Sun Fire 880 Dynamic Reconfiguration User's 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 test address that was 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 configuration triggers the in.mpathd daemon to resume probing. As a result of this probing, in.mpathd detects the repair. Consequently, in.mpathd causes the original IP address to fail back from hme1.

See "Configuring Test Addresses" on page 17 for more details about how to configure test addresses.

Note – The failback of IP addresses during the recovery of a failed physical interface requires as much as three minutes. This time might vary. The time depends on network traffic. The time also depends on the determination of the stability of the incoming interface to failback failed over interfaces by the in.mpathd daemon.

Recovering a Physical Interface That Was Not Present at System Boot

The steps in this section pertain to only IP layers that are configured by using ifconfig(1M). Layers before or after the IP layer, such as ATM or other services, require specific manual steps if the layers are not automated. The specific steps are used to unconfigure during pre-detach and configure after post-attach. See the layers and applications documentation for instructions on how to handle the failure and DR scenarios.

Recovery after a DR operation for a NIC that is part of the IO board on a Sun Fire platform is automatic. If the NIC is a PCI device, the recovery is also automatic. Consequently, the following steps are not required for a NIC that is coming back as part of a DR operation. For more information on the Sun Fire x800 and Sun Fire 15000, see the cfgadm_sbd(1M) man page. The physical interface is recovered to the configuration that is specified in the /etc/hostname.*interface* file. See "Configuring Multipathing Interface Groups" on page 27 for details on how to configure interfaces to preserve the configuration across reboots.

Note – On Sun Fire legacy (Exx00) systems, DR detaches are still subject to manual procedures. However, DR attaches are automated.

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. The procedure also assumes that hme0 was not present at system boot.

Note – The failback of IP addresses during the recovery of a failed physical interface last three minutes. This time might vary. The time depends on network traffic. The time also depends on the determination of the stability of the incoming interface to fail back failed-over interfaces by the in.mpathd daemon.

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

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

   See the syslog(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 interface)
   ```

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

   Refer to the cfgadm(1M) man page, *Sun Enterprise 10000 DR Configuration Guide*, or *Sun Enterprise 6x00, 5x00, 4x00, and 3x00 Systems 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 that are 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 that are marked with the `failover` parameter, or those addresses that are not marked with the `-failover` parameter. You do not need to unplumb IP addresses that are 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 type 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 25 for a description of these parameters.

▼ How to Configure the Multipathing Configuration File

1. **Become superuser.**

2. **Edit the** `/etc/default/mpathd`**. Change the default value of one or more of the three parameters by using one or more of the following substeps.**

   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.**

   ```
   # pkill -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/?p=/doc/805-4368.

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 *Network adapter* 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

L

M

N

P

R

S

T