Administración de redes TCP/IP, IPMP y túneles IP en Oracle® Solaris 11.2

Referencia: E53800
Julio de 2014

ORACLE

Contenido

Uso de esta documentación

- **Descripción general**: describe tareas para administrar redes TCP/IP, IPMP y túneles IP en el sistema operativo Oracle Solaris.
- **Destinatarios**: administradores de sistemas.
- **Conocimientos necesarios**: experiencia avanzada en la administración de red, incluida la administración de redes TCP/IP y funciones avanzadas de red como IPMP y túneles IP.

Biblioteca de documentación del producto

En la biblioteca de documentación (http://www.oracle.com/pls/topic/lookup?ctx=E56339), se incluye información de última hora y problemas conocidos para este producto.

Acceso a My Oracle Support

Los clientes de Oracle tienen acceso a soporte electrónico por medio de My Oracle Support. Para obtener más información, visite http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info o, si tiene alguna discapacidad auditiva, visite http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs.

Feedback

Envíenos comentarios acerca de esta documentación mediante http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CAPÍTULO 1

Administración de redes TCP/IP

En este capítulo se describe cómo administrar el protocolo TCP / IP en sistemas que tienen instalado Oracle Solaris. Las tareas en este capítulo dan por sentado que se dispone de una red TCP/IP operativa en su sitio, ya sea IPv4 solamente o IPv4/IPv6 de doble pila.

Para obtener información sobre la planificación del desarrollo de su red, consulte “Planificación de la implementación de red en Oracle Solaris 11.2 ”.

Para conocer las tareas de configuración de red, consulte “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.

Para obtener información general sobre la resolución de problemas de redes TCP/IP, consulte “Resolución de problemas de administración de redes en Oracle Solaris 11.2 ”.

Este capítulo se divide en los siguientes apartados:

Personalización de las propiedades de TCP/IP

El comando `ipadm` se utiliza para configurar la mayoría de las propiedades de TCP/IP, también conocidas como *ajustables*. El comando `ipadm` reemplaza el comando `ndd` como herramienta

principal para definir los valores ajustables. Para obtener más información acerca de estos cambios, consulte "Comparación del comando ndd con el comando ipadm" de "Transición de Oracle Solaris 10 a Oracle Solaris 11.2 ".

Las propiedades de TCP/IP pueden basarse en la interfaz o ser globales y las propiedades se pueden aplicar a una interfaz específica o globalmente a todas las interfaces dentro de una zona. Las propiedades globales también pueden tener diferentes valores en diferentes zonas no globales. Para obtener una lista completa de las propiedades de protocolo admitidas, consulte la página del comando man `ipadm`(1M).

Normalmente, los valores predeterminados del protocolo TCP/IP son suficientes para que la red funcione. Sin embargo, si estos valores son insuficientes para su topología de red en concreto, puede personalizar las propiedades mediante los siguientes tres subcomandos de `ipadm`:

- `ipadm show-prop -p` *property protocol*: muestra las propiedades de un protocolo y sus valores actuales. Si no utiliza la opción `-p` *property*, se muestran todas las propiedades del protocolo. Si no especifica un protocolo, se muestran todas las propiedades de todos los protocolos.
- `ipadm set-prop -p` *property=value protocol*: asigna un valor a la propiedad del protocolo.
  - Para asignar varios valores a la propiedad de un protocolo, utilice la siguiente sintaxis:

    `ipadm set-prop [-t] -p` *property=value*`[,...] protocol`
  - Para eliminar un valor de un conjunto de valores de una propiedad determinada, utilice el cualificador `-=`:

    `ipadm set-prop -p` *property*`-=`*value2*
- Restablezca una propiedad de protocolo específica para sus valores predeterminados de la siguiente manera:

  `ipadm reset-prop -p` *property protocol*

Para obtener información sobre la personalización de las propiedades de interfaz IP, consulte "Personalización de las propiedades y direcciones de la interfaz IP" de "Configuración y administración de componentes de red en Oracle Solaris 11.2 ".

Para obtener información sobre la personalización de las propiedades de la dirección IP, consulte "Personalización de las propiedades de dirección IP" de "Configuración y administración de componentes de red en Oracle Solaris 11.2 ".

Activación del reenvío de paquetes de forma global

Al activar el reenvío en una interfaz IP con el comando `ipadm set-ifprop`, sólo se activa para dicha interfaz, y el reenvío en todas las demás interfaces permanece sin cambios. La configuración del reenvío de paquetes en una propiedad de interfaz IP Individual permite implementar esta función de manera selectiva en interfaces específicas del sistema. Para

obtener más información, consulte “Activación de reenvío de paquetes” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.

Para activar el reenvío de paquetes en un sistema completo, independientemente del número de interfaces IP, utilice la propiedad `protocol`. La propiedad `forwarding` es la propiedad de IP global que se utiliza para gestionar el reenvío, que es el mismo nombre de propiedad que se utiliza para gestionar el reenvío en interfaces IP individuales. Dado que puede activar el reenvío para los protocolos de IPv4 o IPv6 (o ambos), debe gestionar cada protocolo individualmente.

Por ejemplo, debe activar el reenvío de paquetes para todo el tráfico IPv4 e IPv6 en el sistema de la siguiente manera:

```
# ipadm show-prop -p forwarding ip
PROTO   PROPERTY     PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4    forwarding   rw     off       --           off       on,off
ipv6    forwarding   rw     off       --           off       on,off

# ipadm set-prop -p forwarding=on ipv4
# ipadm set-prop -p forwarding=on ipv6

# ipadm show-prop -p forwarding ip
PROTO   PROPERTY     PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
ipv4    forwarding   rw     on        on           off       on,off
ipv6    forwarding   rw     on        on           off       on,off
```

Nota - La propiedad `forwarding` de protocolos o interfaces IP no es exclusiva. Puede establecer la propiedad para la interfaz y el protocolo al mismo tiempo. Por ejemplo, puede activar el reenvío de paquetes de forma global en el protocolo y, a continuación, personalizar el reenvío de paquetes de cada interfaz IP del sistema. Por lo tanto, aunque esté activado globalmente, aún pueden gestionar el reenvío de paquetes de manera selectiva sobre la base de una interfaz.

Administración de selección de direcciones predeterminadas

Oracle Solaris permite que una misma interfaz tenga varias direcciones IP. Por ejemplo, las tecnologías como IPMP permiten que varias tarjetas de interfaz de red (NIC) se conecten a la misma capa de enlace IP. Ese vínculo puede tener una o varias direcciones IP. Además, las interfaces en sistemas compatibles con IPv6 disponen de una dirección IPv6 local de vínculo, como mínimo una dirección de enrutamiento IPv6 y una dirección IPv4 para al menos una interfaz.

Cuando el sistema inicia una transacción, una aplicación realiza una llamada al socket `getaddrinfo`. El socket `getaddrinfo` descubre la posible dirección que está en uso en el sistema de destino. El núcleo da prioridad a esta lista a fin de buscar el destino más idóneo para

el paquete. Este proceso se denomina *ordenación de direcciones de destino*. A continuación, el núcleo de Oracle Solaris selecciona el formato correspondiente para la dirección de origen, a partir de la dirección de destino más apropiada para el paquete. Este proceso se denomina *selección de direcciones*. Para obtener más información sobre la ordenación de direcciones de destino, consulte la página del comando man `getaddrinfo`(3SOCKET).

Los sistemas IPv4 y de doble pila IPv4/IPv6 deben realizar una selección de direcciones predeterminadas. En la mayoría de los casos, no hace falta cambiar los mecanismos de selección de direcciones predeterminadas. Sin embargo, quizá deba cambiar la prioridad de los formatos de direcciones para poder admitir IPMP o preferir los formatos de direcciones 6to4, por ejemplo.

Descripción del archivo de configuración `/etc/inet/ipaddrsel.conf`

El archivo `/etc/inet/ipaddrsel.conf` contiene la tabla de directrices de selección de direcciones predeterminadas de IPv6. Al instalar Oracle Solaris activado para IPv6, este archivo incluye el contenido que se muestra en la Tabla 1-1, “Tabla de directrices de selección de direcciones IPv6”.

El contenido de `/etc/inet/ipaddrsel.conf` se puede editar. Ahora bien, en la mayoría de los casos no es conveniente modificarlo. Si hace falta realizar cambios, consulte el procedimiento Cómo administrar la tabla de directrices de selección de direcciones IPv6 [14]. Para obtener más información sobre `ippaddrsel.conf`, consulte “Motivos para modificar la tabla de políticas de selección de direcciones IPv6” [13] y la página del comando man `ipaddrsel.conf`(4).

Descripción del comando `ipaddrsel`

El comando `ipaddrsel` permite modificar la tabla de directrices de selección de direcciones predeterminadas de IPv6.

El núcleo de Oracle Solaris utiliza la tabla de políticas de selección de direcciones IPv6 predeterminadas para ordenar direcciones de destino y seleccionar direcciones de origen en un encabezado de paquetes de IPv6. El archivo `/etc/inet/ipaddrsel.conf` contiene la tabla de políticas.

En la tabla siguiente se enumeran los formatos de direcciones predeterminadas y las correspondientes prioridades en la tabla de directrices. Puede encontrar información técnica para la selección de direcciones IPv6 en la página del comando man `inet6`(7P).

TABLA 1-1 Tabla de directrices de selección de direcciones IPv6

Prefijo	Prioridad	Definición
`::1/128`	50	Bucle de retorno
`::/0`	40	Valor predeterminado
`2002::/16`	30	6to4
`::/96`	20	Compatible con IPv4
`::ffff:0:0/96`	10	IPv4

En esta tabla, los prefijos de IPv6 (`::1/128` y `::/0`) tienen prioridad sobre las direcciones 6to4 (`2002::/16`) y las direcciones IPv4 (`::/96` y `::ffff:0:0/96`). Así pues, de forma predeterminada, el núcleo selecciona la dirección IPv6 global de la interfaz para paquetes que se dirigen a otro destino de IPv6. La dirección IPv4 de la interfaz tiene una prioridad inferior, sobre todo en cuanto a paquetes que se dirigen a un destino de IPv6. A partir de la dirección IPv6 de origen seleccionada, el núcleo también utiliza el formato de IPv6 para la dirección de destino.

Motivos para modificar la tabla de políticas de selección de direcciones IPv6

En la mayoría de los casos, no se necesita cambiar la tabla de directrices de selección de direcciones predeterminadas de IPv6. Para administrar la tabla de políticas, se utiliza el comando `ipaddrsel`.

La tabla de políticas podría modificarse por alguno de los motivos siguientes:

- Si el sistema tiene una interfaz que se emplea para un túnel de 6to4, puede asignar mayor prioridad a las direcciones 6to4.
- Si desea utilizar una determinada dirección de origen sólo para comunicarse con una determinada dirección de destino, puede agregar dichas direcciones a la tabla de directrices. A continuación, mediante el comando `ipadm`, marque estas direcciones en función de sus preferencias. Para obtener más información, consulte la página del comando man `ipadm`(1M).
- Si quiere otorgar más prioridad a las direcciones IPv4 respecto a las de IPv6, la prioridad de `::ffff:0:0/96` puede cambiarse por un número superior.
- Si debe asignar mayor prioridad a direcciones descartadas, tales direcciones se pueden incorporar a la tabla de directrices. Por ejemplo, las direcciones locales de sitio ahora se descartan en IPv6. Estas direcciones tienen el prefijo `fec0::/10`. La tabla de políticas se puede modificar para asignar mayor prioridad a las direcciones locales de sitio.

Para obtener detalles sobre el comando `ipaddrsel`, consulte la página del comando man `ipaddrsel`(1M).

Cómo administrar la tabla de directrices de selección de direcciones IPv6

A continuación se describe el procedimiento para modificar la tabla de políticas de selección de direcciones. Para obtener información conceptual sobre la selección de direcciones IPv6 predeterminadas, consulte Descripción del comando ipaddrsel.

Atención - No cambie la tabla de políticas de selección de direcciones IPv6 excepto por los motivos que se proporcionan en el siguiente procedimiento. Si lo hace, una tabla de políticas mal configurada puede ocasionar problemas en la red. Además, asegúrese de guardar una copia de seguridad de la tabla de políticas, como se muestra en este procedimiento.

1. **Conviértase en un administrador.**

2. **Revise la tabla de políticas de selección de direcciones IPv6 actual.**

```
# ipaddrsel
# Prefix                  Precedence Label
::1/128                           50 Loopback
::/0                              40 Default
2002::/16                         30 6to4
::/96                             20 IPv4_Compatible
::ffff:0.0.0.0/96                 10 IPv4
```

3. **Efectúe una copia de seguridad de la tabla de directrices de direcciones predeterminadas.**

```
# cp /etc/inet/ipaddrsel.conf /etc/inet/ipaddrsel.conf.orig
```

4. **Personalice el archivo `/etc/inet/ipaddrsel.conf`.**

```
# pfedit /etc/inet/ipaddrsel.conf
```

Utilice la sintaxis siguiente para las entradas del archivo `/etc/inet/ipaddrsel`:

prefix/prefix-length precedence label [# comment]

Consulte Ejemplo 1-1, “Modificación de la tabla de políticas de selección de direcciones IPv6 predeterminadas” para obtener ejemplos de algunas de las modificaciones habituales que podría realizar.

5. **Cargue en el núcleo la tabla de directrices modificada.**

```
# ipaddrsel -f /etc/inet/ipaddrsel.conf
```

6. **Si la tabla de directrices modificada presenta problemas, restaure la tabla predeterminada de directrices de selección de direcciones IPv6.**

```
# ipaddrsel -d
```

ejemplo 1-1 Modificación de la tabla de políticas de selección de direcciones IPv6 predeterminadas

A continuación, se muestran varias de las modificaciones habituales que podría querer aplicar a la tabla de políticas:

- Asignar la máxima prioridad a las direcciones 6to4.

```
2002::/16                       50 6to4
::1/128                         45 Loopback
```

  El formato de dirección 6to4 ahora tiene la prioridad más alta: 50. Bucle, que anteriormente presentaba una prioridad de 50, ahora presenta una prioridad de 45. Los demás formatos de direcciones siguen igual.
- Designar una dirección de origen concreta que se deba utilizar en las comunicaciones con una determinada dirección de destino.

```
::1/128                         50 Loopback
2001:1111:1111::1/128           40 ClientNet
2001:2222:2222::/48             40 ClientNet
::/0                            40 Default
```

  Esta entrada en concreto es útil para los host que cuentan sólo con una interfaz física. En este caso, `2001:1111:1111::1/128` se prefiere como dirección de origen de todos los paquetes cuyo destino previsto es la red `2001:2222:2222::/48`. La prioridad 40 otorga una posición preferente a la dirección de origen `2001:1111:1111::1/128` en relación con los demás formatos de direcciones configurados para la interfaz.
- Favorecer direcciones IPv4 respecto a direcciones IPv6.

```
::ffff:0.0.0.0/96               60 IPv4
::1/128                         50 Loopback
.
.
```

  El formato de IPv4 `::ffff:0.0.0.0/96` ha cambiado su prioridad predeterminada de 10 a 60, la prioridad máxima de la tabla.

▼ Cómo modificar la tabla de selección de direcciones IPv6 sólo para la sesión actual

Si edita el archivo `/etc/inet/ipaddrsel.conf`, las modificaciones que efectúe se mantendrán después de cada reinicio del sistema. Si quiere que la tabla de políticas modificada exista únicamente en la sesión actual, siga el siguiente procedimiento.

1. **Conviértase en un administrador.**

2. **Copie el contenido del archivo `/etc/inet/ipaddrsel` a** *nombre de archivo***, donde** *filename* **representa cualquier nombre que usted elija.**

   ```
   # cp /etc/inet/ipaddrsel filename
   ```

3. **Utilice el comando `pfedit` para editar la tabla de políticas de** *filename* **a su conveniencia.**

4. **Cargue en el núcleo la tabla de directrices modificada.**

   ```
   # ipaddrsel -f filename
   ```

   El núcleo emplea la nueva tabla de directrices hasta que se vuelva a iniciar el sistema.

Administración de servicios de capa de transporte

Los siguientes protocolos de capa de transporte son parte estándar de Oracle Solaris: protocolo de control de transmisión (TCP), protocolo de transmisión para el control de flujo (SCTP) y protocolo de datagramas de usuario (UDP). Estos protocolos normalmente no requieren ninguna intervención para ejecutarse correctamente. Sin embargo, las circunstancias de su sitio podrían requerir el registro o la modificación de los servicios que ejecutan los protocolos de capa de transporte. En este caso, debe modificar los perfiles de los servicios con los comandos de la utilidad de gestión de servicios (SMF).

El daemon `inetd` se encarga de iniciar los servicios estándar de Internet cuando se inicia un sistema. Estos servicios incluyen aplicaciones que utilizan TCP, SCTP o UDP como protocolo de capa de transporte. Puede modificar los servicios de Internet existentes o agregar servicios nuevos con los comandos SMF.

Las operaciones que requieren protocolos de capa de transporte incluyen lo siguiente:

- Configuración de un puerto con privilegios
- Implementación del control de congestión de tráfico
- Registro de todas las conexiones TCP entrantes
- Agregar servicios que ejecutan un protocolo de capa de transporte, utilizando SCTP a modo de ejemplo
- Configurar la función de envoltorios TCP para el control de acceso
- Cambio del tamaño de la memoria intermedia de recepción de TCP

Para obtener más información, consulte “Modificación de los servicios controlados por inetd” de “Gestión de los servicios del sistema en Oracle Solaris 11.2 ” y la página del comando man `inetd`(1M).

Configuración de un puerto con privilegios

En protocolos de transporte como TCP, UDP y SCTP, los puertos 1-1023 son puertos con privilegios de manera predeterminada. Para vincular a un puerto con privilegios, un proceso se debe ejecutar con permisos root. Los puertos superiores a 1023 son puertos sin privilegios de manera predeterminada. Puede utilizar el comando ipadm para ampliar el rango de puertos con privilegios, o puede marcar puertos específicos en el rango sin privilegios como puertos con privilegios.

Para gestionar el rango de puertos con privilegios, puede personalizar las siguientes propiedades de protocolo de transporte:

`smallest_nonpriv_port` Especifica un valor que indica el comienzo del rango de números de puerto sin privilegios, que son los puertos a los que los usuarios comunes pueden vincularse. Puede definir puertos individuales dentro del rango sin privilegios como puertos con privilegios. Utilice el comando `ipadm show-prop` para mostrar los valores de la propiedad.

`extra_priv_ports` Especifica qué puertos fuera del rango con privilegios también son con privilegios. Utilice el comando `ipadm set-prop` para especificar los puertos que desee restringir. Se pueden asignar varios valores a esta propiedad.

Por ejemplo, suponga que desea establecer los puertos TCP 3001 y 3050 como puertos con privilegios con acceso restringido sólo al rol root. La propiedad smallest_nonpriv_port indica que 1024 es el número de puerto más bajo para un puerto sin privilegios. Por lo tanto, puede cambiar los puertos 3001 y 3050 designados a puertos con privilegios de la siguiente manera:

```
# ipadm show-prop -p smallest_nonpriv_port tcp
PROTO PROPERTY               PERM   CURRENT   PERSISTENT   DEFAULT   POSSIBLE
tcp   smallest_nonpriv_port   rw     1024      --           1024      1024-32768

# ipadm show-prop -p extra_priv_ports tcp
PROTO   PROPERTY           PERM   CURRENT     PERSISTENT   DEFAULT     POSSIBLE
tcp     extra_priv_ports   rw     2049,4045   --           2049,4045   1-65535

# ipadm set-prop -p extra_priv_ports+=3001 tcp
# ipadm set-prop -p extra_priv_ports+=3050 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO   PROPERTY           PERM   CURRENT     PERSISTENT   DEFAULT     POSSIBLE
tcp     extra_priv_ports   rw     2049,4045   3001,3050    2049,4045   1-65535
                                  3001,3050
```

Debe eliminar un puerto con privilegios, por ejemplo, 4045, de la siguiente manera:

```
# ipadm set-prop -p extra_priv_ports-=4045 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO   PROPERTY           PERM   CURRENT     PERSISTENT   DEFAULT     POSSIBLE
```

```
tcp     extra_priv_ports   rw     2049,3001   3001,3050    2049,4045   1-65535
                                  3050
```

Implementación del control de congestión de tráfico

La congestión de la red generalmente se produce cuando hay desbordamientos de la memoria intermedia del enrutador, cuando los nodos envían más paquetes de los que la red puede recibir. Existen diversos algoritmos que evitan la congestión del tráfico mediante el establecimiento de controles en los sistemas de envío. Estos algoritmos se admiten en Oracle Solaris y pueden agregarse con facilidad o incorporarse directamente en el sistema operativo, como se muestra en la siguiente tabla que describe los algoritmos incorporados.

Algoritmo	Nombre de Oracle Solaris	Descripción
NewReno	`newreno`	Algoritmo predeterminado en Oracle Solaris. Este mecanismo de control incluye la elusión de la congestión, el retardo en el inicio y el intervalo de congestión de un remitente.
HighSpeed	`highspeed`	Una de las modificaciones más sencillas y más conocidas de NewReno para redes de alta velocidad.
CUBIC	`cubic`	Actualmente, el algoritmo predeterminado en Linux 2.6. Cambia la fase de elusión de congestión del aumento de intervalo lineal a una función cúbica.
Vegas	`vegas`	Un algoritmo basado en retraso clásico que intenta predecir la congestión sin desencadenar la pérdida real de paquetes.

El control de congestión se activa mediante el establecimiento de las siguientes propiedades TCP relacionadas con el control. Aunque estas propiedades se muestran para TCP, el mecanismo de control que se activa mediante estas propiedades también se aplica al tráfico SCTP.

`cong_enabled` Contiene una lista de algoritmos, separados por comas, que funcionan actualmente en el sistema. Puede agregar o eliminar algoritmos para activar sólo los algoritmos que desea utilizar. Esta propiedad puede tener varios valores. Por lo tanto, debe usar el cualificador `+=` o `-=`, según el cambio que desee realizar.

`cong_default` Utilizado de manera predeterminada cuando las aplicaciones no especifican los algoritmos explícitamente en las opciones de socket. El valor de la propiedad `cong_default` se aplica a zonas globales y no globales.

En el ejemplo siguiente se muestra cómo agregar un algoritmo para el control de congestión al protocolo:

```
# ipadm set-prop -p cong_enabled+=algorithm tcp
```

Elimine un algoritmo de la siguiente manera:

```
# ipadm set-prop -p cong_enabled-=algorithm tcp
```

Reemplace el algoritmo predeterminado de la siguiente manera:

```
# ipadm set-prop -p cong_default=algorithm tcp
```

Nota - No se siguen reglas de secuencia para agregar o eliminar algoritmos. Puede eliminar un algoritmo antes de agregar otros algoritmos para una propiedad. Sin embargo, la propiedad `cong_default` siempre debe tener un algoritmo definido.

En el ejemplo siguiente se muestra cómo implementar el control de congestión. En este ejemplo, el algoritmo predeterminado para el protocolo TCP se cambia de `newreno` a `cubic`. A continuación, el algoritmo `vegas` se elimina de la lista de algoritmos activados.

```
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY              PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp   extra_priv_ports      rw   2049,4045    --           2049,4045    1-65535

# ipadm show-prop -p cong_default,cong_enabled tcp
PROTO PROPERTY              PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp   cong_default          rw   newreno      --           newreno      newreno,cubic,
                                                                       highspeed,vegas
tcp   cong_enabled          rw   newreno,cubic, newreno,cubic, newreno  newreno,cubic,
                                 highspeed,   highspeed,                highspeed,vegas
                                 vegas        vegas

# ipadm set-prop -p cong_enabled-=vegas tcp
# ipadm set-prop -p cong_default=cubic tcp

# ipadm show-prop -p cong_default,cong_enabled tcp
PROTO PROPERTY              PERM CURRENT      PERSISTENT   DEFAULT      POSSIBLE
tcp   cong_default          rw   cubic        cubic        newreno      newreno,cubic,
                                                                       highspeed
tcp   cong_enabled          rw   newreno,cubic, newreno,cubic, newreno  newreno,cubic,
                                 highspeed    highspeed                highspeed,vegas
```

Registro de las direcciones IP de todas las conexiones TCP entrantes

Las direcciones IP para todas las conexiones TCP entrantes se registran mediante el servicio `syslog`, en la utilidad o nivel `daemon.notice`. La información se encuentra en `/var/adm/`

`messages`, a menos que haya cambiado la configuración de archivos local `syslog.conf`. Tenga en cuenta que cambiar la configuración de archivos `syslog.conf` puede afectar dónde se almacena esta información. Para obtener más información, consulte la página del comando man `syslog.conf`(4).

Configure el seguimiento TCP como `TRUE` (activado) para todos los servicios gestionados por el daemon `inetd` de la siguiente manera:

```
# inetadm -M tcp_trace=TRUE
```

Agregación de servicios que utilizan el protocolo SCTP

El protocolo de protocolo de transmisión para el control de flujo (SCTP) proporciona servicios a los protocolos de capa de aplicación de modo similar a TCP. Sin embargo, SCTP permite las comunicaciones entre dos sistemas, que pueden ser (uno o ambos) de host múltiple. La conexión SCTP se denomina *asociación*. En una asociación, una aplicación divide los datos que se transmitirán en uno o más flujos de mensajes, también denominados *múltiples flujos*. Una conexión SCTP puede realizarse en los puntos finales con varias direcciones IP, lo cual es especialmente importante en las aplicaciones de telefonía. Las posibilidades que ofrece el host múltiple de SCTP constituyen una consideración de seguridad si el sitio utiliza filtro IP o IPsec. En la página del comando man `sctp`(7P) se describen algunas de estas consideraciones.

▼ Cómo agregar servicios que utilicen el protocolo SCTP

De manera predeterminada, SCTP se incluye en Oracle Solaris y no requiere ninguna configuración adicional. Sin embargo, es posible que tenga que configurar de modo explícito determinados servicios de capa de la aplicación para que utilicen SCTP. Algunos ejemplos son aplicaciones de `echo` y `discard`. El procedimiento siguiente describe cómo agregar un servicio echo que utilice un socket de estilo uno a uno SCTP. Puede utilizar el mismo procedimiento para agregar servicios para TCP y UDP.

La tarea siguiente describe cómo agregar un servicio SCTP `inet` que administre el daemon `inetd` al repositorio SMF. Luego, la tarea describe cómo utilizar los comandos SMF para agregar el servicio.

Antes de empezar Antes de llevar a cabo el procedimiento siguiente, cree un archivo de manifiesto para el servicio. Por ejemplo, este procedimiento utiliza un manifiesto para el servicio `echo` que se denomina `echo.sctp.xml`.

1. **Inicie sesión en el sistema local con una cuenta de usuario con privilegios de escritura para los archivos del sistema.**

2. **Agregue una definición para el nuevo servicio en el archivo `/etc/services` mediante el comando `pfedit`.**

   Consulte la página del comando man `pfedit`(1M).

   Utilice la siguiente sintaxis para la definición del servicio:

   service-name port/protocol aliases

3. **Cambie al directorio donde se encuentra el manifiesto del servicio y, a continuación, importe el manifiesto del servicio.**

   ```
   # cd dir-name
   # svccfg import service-manifest-name
   ```

   Por ejemplo, debe agregar un nuevo servicio SCTP `echo` con el manifiesto `echo.sctp.xml` que se encuentra en el directorio `service.dir` de la siguiente manera:

   ```
   # cd service.dir
   # svccfg import echo.sctp.xml
   ```

4. **Compruebe que se haya agregado el manifiesto del servicio.**

   ```
   # svcs FMRI
   ```

   Para el argumento *FMRI*, utilice el Fault Managed Resource Identifier (FMRI) del manifiesto del servicio.

5. **Enumere las propiedades del servicio para determinar si debe realizar modificaciones.**

   ```
   # inetadm -l FMRI
   ```

6. **Active el nuevo servicio.**

   ```
   # inetadm -e FMRI
   ```

7. **Compruebe que el servicio esté activado.**

ejemplo 1-2 Cómo agregar un servicio que utilice el protocolo de transporte SCTP

El siguiente ejemplo muestra los comandos para utilizar y las entradas de archivo que son necesarios para que el servicio `echo` utilice SCTP.

```
# cat /etc/services
.
.
echo            7/tcp
echo            7/udp
echo             7/sctp
```

```
# cd service.dir

# svccfg import echo.sctp.xml

# svcs network/echo*
STATE          STIME    FMRI
disabled       15:46:44 svc:/network/echo:dgram
disabled       15:46:44 svc:/network/echo:stream
disabled       16:17:00 svc:/network/echo:sctp_stream

# inetadm -l svc:/network/echo:sctp_stream
SCOPE    NAME=VALUE
         name="echo"
         endpoint_type="stream"
         proto="sctp"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/lib/inet/in.echod -s"
         user="root"
default  bind_addr=""
default  bind_fail_max=-1
default  bind_fail_interval=-1
default  max_con_rate=-1
default  max_copies=-1
default  con_rate_offline=-1
default  failrate_cnt=40
default  failrate_interval=60
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE

# inetadm -e svc:/network/echo:sctp_stream

# inetadm | grep echo
disabled  disabled       svc:/network/echo:stream
disabled  disabled       svc:/network/echo:dgram
enabled   online         svc:/network/echo:sctp_stream
```

Configuración de envoltorios TCP

Los envoltorios TCP incorporan una medida de seguridad para los daemons de servicio al permanecer entre el daemon y las solicitudes de servicio entrantes. Los envoltorios TCP registran los intentos de conexión correctos e incorrectos. Asimismo, los envoltorios TCP pueden proporcionar control de acceso, y permitir o denegar la conexión en función del lugar donde se origine la solicitud. Puede utilizar envoltorios TCP para proteger los daemons como Secure Shell (SSH), Telnet y el Protocolo de transferencia de archivos (FTP). La aplicación `sendmail` también puede utilizar envoltorios TCP, como se describe en “Compatibilidad con envoltorios TCP de la versión 8.12 de sendmail” de “Gestión de servicios de sendmail en Oracle Solaris 11.2 ”.

▼ Cómo utilizar los envoltorios TCP para controlar el acceso a los servicios TCP

El programa `tcpd` implementa *envoltorios TCP*.

1. **Asuma el rol `root`.**
2. **Active los envoltorios TCP.**

```
# inetadm -M tcp_wrappers=TRUE
```

3. **Configure la política de control de acceso de los envoltorios TCP.**

Para obtener instrucciones, consulte la página del comando man `hosts_access`(3), que se puede encontrar en el directorio `/usr/sfw/man`.

Cambio del tamaño de la memoria intermedia de recepción de TCP

El tamaño de la memoria intermedia de recepción TCP se establece mediante la propiedad TCP `recv_buf`, que de manera predeterminada es de 128 KB. Sin embargo, las aplicaciones no utilizan el ancho de banda de manera uniforme. Por lo tanto, es posible que la latencia de conexión requiera el cambio del tamaño predeterminado. Por ejemplo, si se utiliza la función Secure Shell de Oracle Solaris, se genera una sobrecarga en el uso del ancho de banda debido a los procesos de cifrado y la suma de comprobación adicionales que se realizan en la secuencia de datos. Por lo tanto, es posible que sea necesario aumentar el tamaño de la memoria intermedia. Del mismo modo, para que las aplicaciones que realizan transferencias masivas puedan usar el ancho de banda eficientemente, también es necesario el mismo ajuste de tamaño de memoria intermedia.

Puede calcular el tamaño de memoria intermedia de recepción correcto que se debe utilizar mediante el cálculo del producto de retraso de ancho de banda (BDP). Para calcular el BDP, multiplique el ancho de banda disponible por el valor de la latencia de conexión.

Utilice el comando `ping -s` *host* para obtener el valor de latencia de conexión.

El tamaño de memoria intermedia de recepción adecuado se aproxima al valor del producto de retraso de ancho de banda. Sin embargo, el uso del ancho de banda también depende de una serie de condiciones. Una infraestructura compartida o el número de aplicaciones y usuarios que compiten por el uso del ancho de banda pueden cambiar ese cálculo.

Cambie el valor del tamaño de la memoria intermedia de la siguiente manera:

```
# ipadm set-prop -p recv_buf=value tcp
```

En el siguiente ejemplo se muestra cómo aumentar el tamaño de la memoria intermedia a 164 KB:

```
# ipadm show-prop -p recv_buf tcp
PROTO PROPERTY   PERM CURRENT   PERSISTENT   DEFAULT  POSSIBLE
tcp   recv_buf   rw   128000       --        128000   2048-1048576

# ipadm set-prop -p recv_buf=164000 tcp

# ipadm show-prop -p recv_buf tcp
PROTO PROPERTY   PERM CURRENT   PERSISTENT   DEFAULT  POSSIBLE
tcp   recv_buf   rw   164000    164000       128000   2048-1048576
```

No se prefiere ningún valor establecido para el tamaño de la memoria intermedia, porque el tamaño preferido varía según las circunstancias. Tenga en cuenta los siguientes ejemplos cuando se establecen distintos valores para el BDP en cada red con condiciones específicas:

Red de área local (LAN) típica de 1 Gbps, donde 128 KB es el valor predeterminado del tamaño de la memoria intermedia:	`BDP = 128 MBps * 0.001 s = 128 kB`
Red de área extensa (WAN) teórica de 1 Gbps con latencia de 100 ms:	`BDP = 128 MBps * 0.1 s = 12.8 MB`
Enlace de Europa a Estados Unidos (ancho de banda medido por `uperf`):	`BDP = 2.6 MBps * 0.175 = 470 kB`

Si no puede calcular el BDP, utilice las siguientes directrices:

- Para transferencias masivas mediante una LAN, el valor predeterminado del tamaño de la memoria intermedia (128 KB) es suficiente.
- Para la mayoría de las implementaciones de WAN, el tamaño de la memoria intermedia de recepción debe estar dentro del rango de 2 MB.

Atención - Aumentar el tamaño de la memoria intermedia de recepción de TCP aumenta el espacio de memoria de muchas aplicaciones de red.

Supervisión del estado de la red con el comando netstat

Se utiliza el comando netstat para generar visualizaciones que muestran el estado de la red y estadísticas de protocolo. Puede visualizar el estado de los puntos finales de UDP, SCTP y TCP en formato de tabla, así como visualizar información sobre la tabla de enrutamiento e información de interfaces.

El comando netstat muestra varios tipos de datos de red, según la opción de línea de comandos que se haya especificado. La información que se muestra es especialmente útil para la administración del sistema. Las opciones que se utilizan para algunas de las tareas realizadas comúnmente se describen en este capítulo. Para obtener información completa, consulte la página del comando man netstat(1M).

Esta sección incluye los siguientes temas:

Filtrado de la salida netstat por tipo de dirección

Se puede utilizar el comando netstat para ver el estado de redes IPv4 e IPv6. Puede elegir la información de protocolo que se visualizará; para ello, establezca el valor de DEFAULT_IP en el archivo /etc/default/inet_type o recurra a la opción -f. Al establecer el valor de DEFAULT_IP permanentemente, puede asegurarse de que el comando netstat muestre sólo información de IPv4. Para reemplazar esta configuración, utilice la opción -f desde la línea de comandos. Para obtener más información acerca del archivo inet_type, consulte la página de comando man inet_type(4).

También puede utilizar la opción -f para especificar los argumentos inet, inet6, unix (para sockets de dominio Unix utilizados para la comunicación interna) o sdp (Protocolo de descripción del socket).

Visualización del estado de sockets

Utilice el comando netstat para ver el estado de los sockets en un host local. Puede especificar varias opciones de comandos netstat como usuario común.

Visualice el estado de sockets de conexión de la siguiente forma:

```
% netstat
```

Visualice el estado de todos los sockets, incluidos los sockets del listener que no están conectados, de la siguiente manera:

```
% netstat -a
```

EJEMPLO 1-3 Visualización de sockets conectados

La salida del comando `netstat` muestra estadísticas exhaustivas. En el ejemplo siguiente se muestra cómo limitar la salida a sockets IPv4 únicamente.

```
% netstat -f inet

TCP: IPv4
   Local Address        Remote Address    Swind Send-Q Rwind Recv-Q    State
-------------------- -------------------- ----- ------ ----- ------ -----------
system-1.ssh          remote.38474     128872      0 128872      0 ESTABLISHED
system2.40721         remote.ldap      49232       0 128872      0 ESTABLISHED
```

Visualización de estadísticas por protocolo

La opción `netstat -s` muestra estadísticas de los protocolos UDP, TCP, SCTP, protocolo de mensajes de control de Internet (ICMP) e IP.

Visualice el estado del protocolo de la siguiente manera:

```
% netstat -s
```

Utilice la opción `-P` para filtrar la salida del comando `netstat` por protocolo. Esta opción no está limitada a los protocolos de transporte.

Puede especificar los siguientes valores con esta opción:

- `icmp`
- `icmpv6`
- `igmp`
- `ipv6tcp`
- `rawip`
- `sctp`
- `tcp`
- `udp`

Por ejemplo, debe filtrar la salida `netstat` por el protocolo UDP de la siguiente manera:

```
% netstat -s -P udp
```

```
UDP     udpInDatagrams      =  3704     udpInErrors         =     0
        udpOutDatagrams     =  3703     udpOutErrors        =     0
```

EJEMPLO 1-4 Visualización del estado del protocolo TCP

En el ejemplo siguiente se muestra cómo visualizar los resultados para el protocolo TCP especificando la opción -P.

```
% netstat -P tcp

TCP: IPv4
Local Address     Remote Address    Swind Send-Q  Rwind Recv-Q      State
----------------- ------------------- ----- ------  ----- ------     -------
lhost-1.login      abc.def.local.Sun.COM.980 49640      0     49640    0 ESTABLISHED
lhost.login        ghi.jkl.local.Sun.COM.1020 49640     1     49640    0 ESTABLISHED
remhost.1014       mno.pqr.remote.Sun.COM.nfsd 49640    0     49640    0 TIME_WAIT

TCP: IPv6
Local Address    Remote Address        Swind Send-Q Rwind Recv-Q   State If
---------------- --------------------- ------ ----- ------ ------------ -----
localhost.38983   localhost.32777       49152      0 49152      0 ESTABLISHED
localhost.32777   localhost.38983       49152      0 49152      0 ESTABLISHED
localhost.38986   localhost.38980       49152      0 49152      0 ESTABLISHED
```

Visualización del estado de interfaces de red

Utilice la opción i del comando netstat para ver el estado de las interfaces de red que se configuran en el sistema local. Esta opción permite determinar la cantidad de paquetes que transmite un sistema y que recibe cada red.

Visualice el estado de las interfaces que están en una red de la siguiente manera:

```
% netstat -i
```

En el ejemplo siguiente se muestra cómo utilizar el comando netstat con una opción de filtrado para limitar la salida de una interfaz específica:

```
% netstat -i -I net0 -f inet
  Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
  net0  1500 abc.oracle.com abc.oracle.com 231001 0     55856  0     0      0
```

EJEMPLO 1-5 Visualización del estado de interfaces de red

En el ejemplo siguiente se muestra el estado de un flujo de paquetes IPv4 e IPv6 a través de las interfaces del host. Por ejemplo, la cantidad de paquetes de entrada (Ipkts) que aparece en un servidor puede aumentar cada vez que un cliente intenta iniciar, mientras que la cantidad de paquetes de salida (Opkts) no se modifica. De esta salida puede inferirse que el servidor está reconociendo los paquetes de solicitud de inicio del cliente. Sin embargo, parece que el servidor

no sabe responder. Esta confusión podría deberse a una dirección incorrecta en la base de datos hosts o ethers.

Si el recuento de paquetes de entrada permanece invariable, el equipo no ve los paquetes. De este resultado puede inferirse otra clase de error, posiblemente un problema de hardware.

```
Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 loopback      localhost      142    0     142    0     0      0
net0  1500 host58        host58        1106302 0     52419  0     0      0

Name  Mtu  Net/Dest      Address                   Ipkts  Ierrs Opkts  Oerrs Collis
lo0   8252 localhost     localhost                  142    0     142    0     0
net0  1500 fe80::a00:20ff:feb9:4c54/10 fe80::a00:20ff:feb9:4c54 1106305 0 52422 0  0
```

Visualización de información de usuario y procesos

El comando netstat incluye una nueva opción -u. Esta opción proporciona información sobre usuarios y procesos del sistema que utilizan puertos específicos. Utilice el comando netstat -u para ver el usuario, el ID del proceso y el programa que ha creado el punto final de red o el programa que actualmente controla el punto final de red.

Para producir una mejor alineación de la salida del comando netstat, puede especificar la opción -n con la opción -u. Para obtener más información y ejemplos detallados, consulte la página del comando man netstat(1M).

Visualización del estado de rutas conocidas

Utilice la opción -r con el comando netstat para ver la tabla de enrutamiento para un host local. Esta tabla muestra el estado de todas las rutas conocidas para un host.

```
% netstat -r
```

EJEMPLO 1-6 Visualización de la salida de tabla de enrutamiento con el comando netstat

El siguiente ejemplo muestra la salida del comando netadm list -x.

```
Routing Table: IPv4
Destination           Gateway           Flags  Ref   Use   Interface
-------------------- -------------------- ----- ----- ------ ---------
host15               myhost               U         1  31059  net0
10.0.0.14            myhost               U         1      0  net0
default              distantrouter        UG        1      2  net0
localhost            localhost            UH        42019361  lo0

Routing Table: IPv6
```

```
Destination/Mask      Gateway                           Flags Ref  Use   If
--------------------- --------------------------          ----- --- ------ -----
2002:0a00:3010:2::/64 2002:0a00:3010:2:1b2b:3c4c:5e6e:abcd U     1   0      net0:1
fe80::/10             fe80::1a2b:3c4d:5e6f:12a2           U     1   23     net0
ff00::/8              fe80::1a2b:3c4d:5e6f:12a2           U     1   0      net0
default               fe80::1a2b:3c4d:5e6f:12a2           UG    1   0      net0
localhost             localhost                           UH    9   21832  lo0
```

La siguiente tabla describe la información que se muestran en la salida del comando `netstat -r`.

Parámetro	Descripción
`Destination` (Destino)	Indica el host que es el punto final de destino de la ruta. La tabla de enrutamiento IPv6 muestra el prefijo de un punto final de túnel 6to4 (`2002:0a00:3010:2::/64`) como punto final de destino de la ruta.
`Destination/Mask`	
`Gateway` (Puerta de enlace)	Especifica el portal que se usa para enviar paquetes.
`Flags (Indicadores)`	Indica el estado actual de la ruta. El indicador `U` especifica que la ruta está activa. El indicador `G` especifica que la ruta es a un portal.
`Usar`	Muestra la cantidad de paquetes enviados.
`Interfaz`	Indica la interfaz concreta del host local que es el punto final de origen de la transmisión.

Visualización del estado de red adicional con el comando `netstat`

Puede utilizar el comando `netstat` con otras opciones de línea de comandos para mostrar el estado de red adicional, más allá de lo que se documenta en este capítulo. Hay diez formas de comando y cada una produce una tabla de estadísticas diferente sobre las distintas partes del subsistema de red.

Algunas de las estadísticas adicionales que puede obtener incluyen lo siguiente:

`netstat -g`	Muestra los miembros de grupo de multidifusión
`netstat -P`	Muestra las tablas de red a soporte: asignaciones ARP y NDP
`netstat -m`	Muestra las estadísticas de memoria de STREAMS
`netstat -M`	Muestra la tabla de enrutamiento de multidifusión
`netstat -D`	Muestra el estado de permisos DHCP
`netstat -d`	Muestra la tabla caché de destino

Para obtener información completa, consulte la página del comando man `netstat`(1M).

Realización de la administración TCP y UDP con la utilidad netcat

Utilice la utilidad netcat (nc) para realizar una serie de tareas asociadas con la administración de TCP o UDP. Puede utilizar el comando para redes tanto IPv4 como IPv6.

Puede realizar las siguientes tareas con la utilidad netcat:

- Conexiones TCP abiertas
- Enviar paquetes UDP
- Recibir en puertos TCP y UDP arbitrarios
- Realizar la exploración de puertos

A diferencia del comando telnet, donde cada mensaje de error se emite por separado en la salida estándar, los mensajes de error generados por secuencias de comandos nc se combinan en un error estándar, método más eficaz.

La utilidad netcat admite una nueva opción -M, que le permite especificar las propiedades del acuerdo de nivel de servicios (SLA). Al especificar las propiedades adecuadas junto con la opción -M, se crea un flujo de MAC para el socket. Por ejemplo, puede utilizar la opción -M de la siguiente manera:

```
% nc -M maxbw=50M host.example.com 7777
% nc -l -M priority=high,inherit=on 2222
```

Como se muestra en el ejemplo anterior, la opción -M se puede utilizar para especificar una lista separada por comas de pares *name=value* de propiedades de SLA.

Algunos métodos de instalación no instalan el depósito de paquetes de software netcat de manera predeterminada. Compruebe si el depósito de paquetes está instalado en el sistema de la siguiente manera:

```
% pkg list network/netcat
```

Si el depósito de paquetes no está instalado, instálelo de la siguiente manera:

```
% pfexec pkg install pkg:/network/netcat
```

Para obtener información completa, consulte la página del comando man netcat(1).

Administración y registro de la visualización del estado de la red

Las tareas siguientes describen cómo comprobar el estado de la red mediante comandos de red perfectamente conocidos.

▼ Cómo controlar la salida de visualización de comandos relacionados con IP

Puede controlar la salida del comando `netstat` para visualizar información de IPv4 únicamente, o información de IPv4 y de IPv6.

1. **Cree el archivo `/etc/default/inet_type`.**

2. **Agregue una de las entradas siguientes al archivo `/etc/default/inet_type`, según lo que necesite la red:**

   - **Para visualizar únicamente información de IPv4, establezca la siguiente variable:**

     ```
     DEFAULT_IP=IP_VERSION4
     ```

   - **Para visualizar información tanto de IPv4 como de IPv6, defina una de las siguientes variables:**

     ```
     DEFAULT_IP=BOTH

     DEFAULT_IP=IP_VERSION6
     ```

Para obtener más información, consulte la página del comando man `inet_type`(4).

Nota - La opción `-f` del comando `netstat` sustituye los valores establecidos en el archivo `inet_type`.

---

ejemplo 1-7 Control de la salida para ver información de IPv4 e IPv6

El ejemplo siguiente muestra la salida al especificar la variable `DEFAULT_IP=BOTH` o `DEFAULT_IP=IP_VERSION6` en el archivo `inet_type`:

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
net0: flags=100001004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4,PHYSRUNNING> mtu 1500 index
 603
```

```
        inet 10.46.86.54 netmask ffffff00 broadcast 10.46.8.255
        ether 0:1e:68:49:98:80
lo0: flags=2002000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
        inet6 ::1/128
net0: flags=120002000841<UP,RUNNING,MULTICAST,IPv6,PHYSRUNNING> mtu 1500 index 603
        inet6 fe80::21e:68ff:fe49:9880/10
        ether 0:1e:68:49:98:80
net0:1: flags=120002080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6,PHYSRUNNING> mtu 1500 index 603
        inet6 2001:b400:414:6059:21e:68ff:fe49:9880/64
```

Si se especifica la variable `DEFAULT_IP=IP_VERSION4` en el archivo `inet_type`, debe ver la siguiente salida:

```
# ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
net0: flags=100001004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4,PHYSRUNNING> mtu 1500 index
 603
        inet 10.46.86.54 netmask ffffff00 broadcast 10.46.8.255
        ether 0:1e:68:49:98:80
```

Registro de acciones del daemon de enrutamiento de IPv4

Si tiene la impresión de que el daemon de enrutamiento de IPv4, `routed`, funciona de modo incorrecto, inicie un log que efectúe el seguimiento de la actividad del daemon. El log incluye todas las transferencias de paquetes al iniciarse el daemon `routed`.

Cree un archivo log que efectúe el seguimiento de acciones del daemon de enrutamiento de la siguiente forma:

```
# /usr/sbin/in.routed /var/log-file-name
```

Atención - En una red que esté ocupada, este comando puede generar salida casi continua.

El ejemplo siguiente muestra el comienzo del registro que se crea al realizar el procedimiento “Registro de acciones del daemon de enrutamiento de IPv4” [32].

```
-- 2003/11/18 16:47:00.000000 --
Tracing actions started
RCVBUF=61440
Add interface lo0  #1   127.0.0.1      -->127.0.0.1/32
<UP|LOOPBACK|RUNNING|MULTICAST|IPv4> <PASSIVE>
Add interface net0 #2   10.10.48.112    -->10.10.48.0/25
<UP|BROADCAST|RUNNING|MULTICAST|IPv4>
turn on RIP
Add    10.0.0.0        -->10.10.48.112      metric=0  net0  <NET_SYN>
Add    10.10.48.85/25  -->10.10.48.112      metric=0  net0  <IF|NOPROP>
```

▼ Cómo efectuar el seguimiento de las actividades del daemon de descubrimiento cercano de IPv6

Si tiene la impresión de que el daemon `in.ndpd` funciona de modo incorrecto, inicie un log que efectúe el seguimiento de la actividad del daemon. Dicho seguimiento se refleja en la salida estándar hasta su conclusión. En el seguimiento figuran todas las transferencias de paquetes al iniciarse el daemon `in.ndpd`.

1. **Inicie el seguimiento del daemon `in.ndpd`.**

   ```
   # /usr/lib/inet/in.ndpd -t
   ```

2. **Concluya el seguimiento a su conveniencia. Para ello, presione la teclas Control +C.**

ejemplo 1-8 Seguimiento del daemon `in.ndpd`

En la salida siguiente se muestra el inicio de un seguimiento del daemon `in.ndpd`.

```
# /usr/lib/inet/in.ndpd -t
Nov 18 17:27:28 Sending solicitation to  ff02::2 (16 bytes) on net0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:b9:4c:54>
Nov 18 17:27:28 Received valid advert from fe80::a00:20ff:fee9:2d27 (88 bytes) on net0
Nov 18 17:27:28         Max hop limit: 0
Nov 18 17:27:28         Managed address configuration: Not set
Nov 18 17:27:28         Other configuration flag: Not set
Nov 18 17:27:28         Router lifetime: 1800
Nov 18 17:27:28         Reachable timer: 0
Nov 18 17:27:28         Reachable retrans timer: 0
Nov 18 17:27:28         Source LLA: len 6 <08:00:20:e9:2d:27>
Nov 18 17:27:28         Prefix: 2001:08db:3c4d:1::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
Nov 18 17:27:28         Prefix: 2002:0a00:3010:2::/64
Nov 18 17:27:28                 On link flag:Set
Nov 18 17:27:28                 Auto addrconf flag:Set
Nov 18 17:27:28                 Valid time: 2592000
Nov 18 17:27:28                 Preferred time: 604800
```

Sondeo de hosts remotos con el comando `ping`

Puede utilizar el comando `ping` para determinar si el sistema se puede comunicar con un host remoto. Al ejecutar el comando `ping`, el protocolo ICMP envía al host un determinado

datagrama para solicitar una respuesta. ICMP es el protocolo que se ocupa de los errores en las redes TCP/IP. Al utilizar el comando `ping`, puede determinar si se pueden intercambiar paquetes IP entre su host y un host remoto especificado.

La siguiente es la sintaxis básica del comando `ping`:

`/usr/sbin/ping` *host* **`[`***timeout***`]`**

donde *host* corresponde al nombre del host remoto. El argumento *timeout* opcional indica el tiempo (en segundos) para que el comando `ping` siga intentando contactar con el host remoto. El valor predeterminado es de 20 segundos. Para obtener más información, consulte la página del comando man `ping`(1M).

Modificaciones del comando `ping` para admitir IPv6

El comando `ping` puede utilizar protocolos IPv4 e IPv6 para sondear hosts de destino. La selección de protocolo depende de las direcciones que devuelve el servidor de nombres en relación con el host de destino específico. De forma predeterminada, si el servidor de nombres devuelve una dirección IPv6 para el host de destino, el comando `ping` utiliza el protocolo IPv6. Si el servidor devuelve sólo una dirección IPv4, el comando `ping` emplea el protocolo IPv4. Si desea anular este comportamiento, utilice la opción `-A` para indicar el protocolo que debe usarse.

Para obtener información detallada, consulte la página del comando man `ping`(1M).

Determinación de si un host remoto está accesible

Para determinar si un host remoto está accesible, utilice el comando `ping` de la siguiente manera:

`%` **`ping`** *hostname*

Si el host acepta transmisiones ICMP, aparece el siguiente mensaje:

hostname `is alive`

Este mensaje indica que el host ha respondido a la solicitud de ICMP. Sin embargo, si el host está inactivo, no puede recibir los paquetes ICMP o bien, si existe un problema de enrutamiento entre su host y el host remoto, recibe la siguiente respuesta del comando `ping`:

`no answer from` *hostname*

Determinación de si los paquetes entre su host y un host remoto se están perdiendo

La pérdida de paquetes puede provocar que las conexiones de red se ralenticen porque se dedica tiempo adicional a retransmitir datos perdidos. Utilice la opción -s del comando ping para determinar si los paquetes entre su host y un host remoto se están perdiendo:

```
% ping -s hostname
```

En el siguiente ejemplo, el comando ping -s *hostname* envía constantemente paquetes al host especificado hasta que se envía un carácter de interrupción o finaliza el tiempo de espera:

```
% ping -s host1.domain8
PING host1.domain8 : 56 data bytes
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=0. time=1.67 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=1. time=1.02 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=2. time=0.986 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=3. time=0.921 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=4. time=1.16 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.00 ms
64 bytes from host1.domain8.COM (172.16.83.64): icmp_seq=5. time=1.980 ms

^C

----host1.domain8  PING Statistics----
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms)  min/avg/max/stddev = 0.921/1.11/1.67/0.26
```

La estadística de pérdida de paquetes indica si el host ha descartado paquetes. Si el comando ping indica que se produjo una pérdida de paquetes, compruebe el estado de la red mediante los comandos ipadm y netstat. Consulte “Supervisión de direcciones e interfaces IP” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ” y “Supervisión del estado de la red con el comando netstat” [25] para obtener más información.

Visualización de información de enrutamiento con el comando traceroute

El comando traceroute efectúa el seguimiento de la ruta que sigue un paquete de IP en dirección a un sistema remoto. El comando traceroute se usa para descubrir cualquier error de configuración de enrutamiento y errores de ruta de enrutamiento. Si no se puede conectar con un determinado host, el comando traceroute sirve para comprobar la ruta que sigue el paquete hasta el host remoto y detectar los errores que pudiera haber.

Asimismo, el comando traceroute muestra el tiempo de ida y vuelta en cada portal de la ruta del host de destino. Esta información resulta útil para analizar dónde hay tráfico de red lento entre dos host.

Para obtener más información sobre el comando traceroute, consulte la página del comando man traceroute(1M).

Esta sección incluye los siguientes temas:

Modificaciones del comando traceroute para admitir IPv6

El comando traceroute efectúa el seguimiento de las enrutamiento IPv4 e IPv6 de un determinado host. En una perspectiva de protocolos, el comando traceroute utiliza el mismo algoritmo que ping. Si desea anular este comportamiento, utilice la opción de línea de comandos -A. Asimismo, puede efectuar el seguimiento de cada ruta en cada dirección de un host con varias direcciones permanentes mediante la opción de línea de comandos -a.

Detección de la ruta de un host remoto

Utilice el comando traceroute de la siguiente manera para descubrir la ruta de un sistema remoto:

```
% traceroute destination-hostname
```

La salida siguiente del comando traceroute muestra la ruta de siete saltos de un paquete que va del sistema local nearhost al sistema remoto farhost. La salida también muestra los intervalos de tiempo que emplea el paquete en atravesar cada salto.

```
% traceroute farhost
traceroute to farhost (172.16.64.39), 30 hops max, 40 byte packets
1  frbldg7c-86 (172.16.86.1)  1.516 ms  1.283 ms  1.362 ms
2  bldg1a-001 (172.16.1.211)  2.277 ms  1.773 ms  2.186 ms
3  bldg4-bldg1 (172.16.4.42)  1.978 ms  1.986 ms  13.996 ms
4  bldg6-bldg4 (172.16.4.49)  2.655 ms  3.042 ms  2.344 ms
5  ferbldg11a-001 (172.16.1.236)  2.636 ms  3.432 ms  3.830 ms
6  frbldg12b-153 (172.16.153.72)  3.452 ms  3.146 ms  2.962 ms
7  farhost (172.16.64.39)  3.430 ms  3.312 ms  3.451 ms
```

Seguimiento de todo el enrutamiento

Utilice el comando `traceroute` con la opción `-a` en el sistema local para realizar un seguimiento de todo el enrutamiento:

```
% traceroute -a host-name
```

El siguiente ejemplo muestra todo el enrutamiento de un host de doble pila:

```
% traceroute -a v6host
traceroute: Warning: Multiple interfaces found; using 2001:db8:4a3a:1:56:a0:a8 @ net0:2
 traceroute to v6host (2001:db8:4a3b:5:102:a00:fe79:19b0),30 hops max, 60 byte packets
1 v6-rout86 (2001:db8:4a3b:1:56:a00:fe1f:59a1) 35.534 ms 56.998 ms *
2 2001:db8::255:0:c0a8:717 32.659 ms 39.444 ms *
3  farhost (2001:db8:4a3b:2:103:a00:fe9a:ce7b)  401.518 ms  7.143 ms *
4  distant (2001:db8:4a3b:3:100:a00:fe7c:cf35)  113.034 ms  7.949 ms *
5  v6host (2001:db8:4a3b:5:102:a00:fe79:19b0)  66.111 ms *  36.965 ms *

traceroute to v6host  (192.168.10.75),30 hops max,40 byte packets
1  v6-rout86 (172.16.86.1)  4.360 ms  3.452 ms  3.479 ms
2  flrmpj17u (172.16.17.131)  4.062 ms  3.848 ms  3.505 ms
3  farhost (10.0.0.23)  4.773 ms *  4.294 ms
4  distant (192.168.10.104)  5.128 ms  5.362 ms *
5  v6host  (192.168.15.85)  7.298 ms  5.444 ms *
```

Análisis del tráfico de red con analizadores TShark y Wireshar

TShark es un analizador de tráfico de red de línea de comando que permite capturar datos de paquetes desde una red activa o paquetes de lectura de un *archivo de capturas* guardado anteriormente, mediante la impresión de un formulario decodificado de dichos paquetes a la salida estándar o mediante la escritura de los paquetes en un archivo. Sin otras opciones, TShark funciona de forma similar al comando `tcpdump` y también utiliza el mismo formato de archivo de captura activa, `libpcap`. Además, TShark es capaz de detectar, leer y escribir los mismos archivos de captura que los que son compatibles con Wireshark.

Wireshark es un analizador de protocolo de red de interfaz gráfica de usuario (GUI) de terceros que se utiliza para el volcado y el análisis del tráfico de red interactivo. Similar al comando `snoop`, puede utilizar Wireshark para examinar los datos de paquete en una red activa o desde un archivo de capturas guardado anteriormente. De manera predeterminada, Wireshark utiliza el formato `libpcap` para la captura de archivos, que también es utilizado por la utilidad `tcpdump` y otras herramientas similares. Una ventaja clave de utilizar Wireshark es que es capaz de leer e importar varios otros formatos de archivo además del formato `libpcap`.

Tanto TShark como Wireshark proporcionan varias funciones únicas, incluidas las siguientes:

- Son capaces de montar todos los paquetes en una conversación TCP y mostrar los datos de esa conversación en ASCII, EBCDIC o en formato hexadecimal
- Contienen más campos que se pueden filtrar que cualquier otro analizador de protocolo de red
- Utilizan una sintaxis que es mucho más rica que otros analizadores de protocolo de red para la creación de filtros

Para utilizar TShark y Wireshark en su sistema Oracle Solaris, en primer lugar, compruebe que los paquetes de software estén instalados y, si es necesario, instálelos de la siguiente manera:

```
# pkg install tshark
```

```
# pkg install wireshark
```

Para obtener más información, consulte las páginas del comando man `tshark`(1) and `wireshark`(1).

Consulte también la documentación en http://www.wireshark.org/.

Control de transferencias de paquetes con el comando `snoop`

Se puede utilizar el comando `snoop` para supervisar el tráfico de red. El comando `snoop` captura paquetes de red y muestra el contenido en el formato que usted especifique. Los paquetes se pueden visualizar nada más recibirse o se pueden guardar en un archivo. Si el comando `snoop` escribe en un archivo intermedio, es improbable que haya pérdidas de paquete en situaciones de seguimiento ocupado. El comando `snoop` se utiliza entonces para interpretar el archivo.

Para capturar paquetes en y desde la interfaz predeterminada en modo promiscuo, se debe adquirir la función de gestión de redes o el rol `root`. En el formato resumido, el comando `snoop` sólo muestra los datos relativos al protocolo de nivel más alto. Por ejemplo, un paquete de NFS muestra únicamente información de NFS. Se suprime la información subyacente de la llamada a procedimiento remoto (RPC), UDP, IP y Ethernet; sin embargo, se puede visualizar en caso de utilizar cualquiera de las opciones detalladas.

Utilice el comando `snoop` con frecuencia y buen criterio para familiarizarse con el comportamiento normal del sistema. Para obtener más información sobre el comando `snoop`, consulte la página del comando man `snoop`(1M).

Esta sección incluye los siguientes temas:

- Cómo comprobar paquetes entre un cliente y un servidor IPv4 [41]
- “Supervisión del tráfico de red IPv6” [41]
- “Supervisión de paquetes mediante dispositivos de capa IP” [42]

Modificaciones del comando `snoop` para admitir IPv6

El comando `snoop` puede capturar paquetes tanto IPv4 como IPv6. Este comando puede mostrar encabezados de IPv6, encabezados de extensiones de IPv6, encabezados de ICMPv6 y datos de protocolo de descubrimiento de vecinos (ND). De manera predeterminada, el comando `snoop` muestra paquetes de IPv4 e IPv6. Si especifica la palabra clave de protocolo `ip` o `ip6`, el comando muestra sólo paquetes de IPv4 o IPv6. La opción para filtrar IPv6 permite filtrar en todos los paquetes, tanto de IPv4 como IPv6, y mostrar únicamente los paquetes de IPv6. Consulte la página del comando man “Supervisión del tráfico de red IPv6” [41] and the `snoop`(1M).

▼ Cómo comprobar paquetes de todas las interfaces

1. **Imprima la información relativa a las interfaces conectadas al sistema.**

   ```
   # ipadm show-if
   ```

   El comando `snoop` suele utilizar el primer dispositivo que no es de bucle de retorno, que, en general, es la interfaz de red principal.

2. **Comience la captura de paquetes escribiendo `snoop` sin argumentos.**

   ```
   # snoop
   ```

3. **Para detener el proceso, presione Control+C.**

ejemplo 1-9 Visualización de la salida de `snoop` básica

El ejemplo siguiente muestra la salida del comando `snoop` básica para un host de doble pila.

```
# snoop
Using device /dev/net (promiscuous mode)
router5.local.com -> router5.local.com ARP R 10.0.0.13, router5.local.com is
0:10:7b:31:37:80
router5.local.com -> BROADCAST     TFTP Read "network-confg" (octet)
myhost -> DNSserver.local.com      DNS C 192.168.10.10.in-addr.arpa. Internet PTR ?
DNSserver.local.com  myhost        DNS R 192.168.10.10.in-addr.arpa. Internet PTR
niserve2.
```

```
.
.
.
fe80::a00:20ff:febb:e09 -> ff02::9 RIPng R (5 destinations)
```

En la salida anterior, los paquetes que se capturan muestran una respuesta y consulta DNS, así como paquetes de protocolo de resolución de direcciones (ARP) periódicos desde el enrutador local y anuncios de la dirección local de enlace IPv6 para el daemon `in.ripngd`.

▼ Cómo comprobar paquetes de un grupo IPMP

En Oracle Solaris 10, si desea capturar paquetes de un grupo IPMP, debe ejecutar manualmente el comando `snoop` en cada una de las interfaces del grupo IPMP. En la versión Oracle Solaris 11, puede capturar paquetes de todas las interfaces de un grupo IPMP mediante el comando `snoop` con la opción `-I`. Luego, la salida se consolida en un único flujo de salida.

El comando `snoop` suele utilizar el primer dispositivo que no es de bucle de retorno, que, en general, es la interfaz de red principal. Sin embargo, cuando se utiliza la opción `-I`, el comando `snoop` utiliza interfaces IP (desde `/dev/ipnet/*`), como muestra el comando `ipadm show-if`. También puede utilizar esta opción para activar la curiosidad del tráfico en bucle de retorno y otras construcciones de sólo IP. La opción `-d` utiliza interfaces de enlace de datos, que se muestran en la salida del comando `dladm show-link`.

1. **Imprima la información relativa a las interfaces conectadas al sistema.**

   ```
   # ipadm show-if
   ```

2. **Comience la captura de paquetes para el grupo IPMP especificado.**

   ```
   # snoop -I ipmp-group
   ```

3. **Para detener el proceso, presione Control+C.**

▼ Cómo capturar la salida del comando `snoop` en un archivo

1. **Capture una sesión de `snoop` en un archivo. Por ejemplo:**

   ```
   # snoop -o /tmp/cap
   Using device /dev/eri (promiscuous mode)
   30 snoop: 30 packets captured
   ```

   En el ejemplo anterior, se han capturado 30 paquetes en un archivo que se denomina `/tmp/cap`. El archivo se puede ubicar en cualquier directorio que disponga de suficiente espacio en

disco. La cantidad de paquetes capturados se muestra en la línea de comandos, y permite pulsar Control+C para cancelar en cualquier momento.

El comando `snoop` crea una evidente carga de red en la máquina host que puede distorsionar el resultado. Para ver el resultado real, ejecute el comando `snoop` desde otro sistema.

2. **Inspeccione el archivo de capturas de la salida del comando `snoop`.**

```
# snoop -i filename
```

ejemplo 1-10 Visualización de capturas de la salida de `snoop`

La salida siguiente muestra una captura que se puede recibir como salida del comando `snoop -i`.

```
# snoop -i /tmp/cap
1   0.00000 fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375
ICMPv6 Neighbor advertisement
...
10  0.91493    10.0.0.40 -> (broadcast)  ARP C Who is 10.0.0.40, 10.0.0.40 ?
34  0.43690 nearserver.here.com  -> 224.0.1.1  IP  D=224.0.1.1 S=10.0.0.40 LEN=28,
ID=47453, TO =0x0, TTL=1
35  0.00034  10.0.0.40 -> 224.0.1.1    IP  D=224.0.1.1 S=10.0.0.40 LEN=28, ID=57376,
TOS=0x0, TTL=47
```

▼ Cómo comprobar paquetes entre un cliente y un servidor IPv4

1. **Establezca un sistema `snoop` fuera de un hub (o puerto reflejado en un conmutador) conectado al cliente o al servidor.**

   El tercer sistema (sistema `snoop`) comprueba todo el tráfico involucrado, de manera que el seguimiento de `snoop` refleja lo que sucede realmente en la conexión.

2. **Escriba el comando `snoop` con opciones adecuadas y guarde la salida que se genere en un archivo.**

3. **Inspeccione e interprete la salida.**

   Consulte RFC 1761, Snoop Version 2 Packet Capture File Format (http://www.rfc-editor.org/rfc/rfc1761.txt) para obtener detalles del archivo de capturas `snoop`.

Supervisión del tráfico de red IPv6

Utilice el comando `snoop` de la siguiente manera para capturar paquetes IPv6:

```
# snoop ip6
```

El ejemplo siguiente muestra la salida típica que puede aparecer al ejecutar el comando `snoop ip6` en un nodo.

```
# snoop ip6
fe80::a00:20ff:fecd:4374 -> ff02::1:ffe9:2d27 ICMPv6 Neighbor solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
solicitation
fe80::a00:20ff:fee9:2d27 -> fe80::a00:20ff:fecd:4375 ICMPv6 Neighbor
solicitation
fe80::a00:20ff:febb:e09 -> ff02::9      RIPng R (11 destinations)
fe80::a00:20ff:fee9:2d27 -> ff02::1:ffcd:4375 ICMPv6 Neighbor solicitation
```

Para obtener más información sobre el comando `snoop`, consulte la página de comando man snoop(1M).

Supervisión de paquetes mediante dispositivos de capa IP

Los dispositivos de capa IP se agregan en Oracle Solaris para mejorar la observabilidad IP. Estos dispositivos ofrecen acceso a todos los paquetes con direcciones que están asociadas con la interfaz de red del sistema. Las direcciones incluyen direcciones locales y direcciones que están alojadas en interfaces que no son de bucle de retorno o interfaces lógicas. El tráfico observable puede incluir tanto direcciones IPv4 como direcciones IPv6. Por lo tanto, se puede supervisar todo el tráfico destinado al sistema. El tráfico puede incluir tráfico IP en bucle de retorno, paquetes de máquinas remotas, paquetes que se envían desde el sistema o todo el tráfico reenviado.

Con los dispositivos de capa IP, un administrador de una zona global de Oracle Solaris puede supervisar el tráfico entre zonas y dentro de una zona. Un administrador de una zona no global también puede observar el tráfico que envía y recibe esa zona.

Para supervisar tráfico en la capa IP, utilice el comando `snoop` con `-I` más nueva. Esta opción especifica que el comando utilice los dispositivos de capa IP nuevos, en lugar del dispositivo subyacente de capa de enlace, para visualizar los datos de tráfico.

▼ Cómo comprobar paquetes en la capa IP

1. **(Opcional) Si es necesario, imprima la información relativa a las interfaces conectadas al sistema.**

   ```
   # ipadm show-if
   ```

2. **Capture el tráfico IP en una interfaz específica.**

```
# snoop -I interface [-V | -v]
```

Métodos de comprobación de paquetes

Todos los siguientes ejemplos se basan en esta configuración del sistema:

```
# ipadm show-addr

ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           dhcp     ok           10.153.123.225/24
lo0/v6            static   ok           ::1/128
net0/v6           addrconf ok           fe80::214:4fff:2731:b1a9/10
net0/v6           addrconf ok           2001:0db8:212:60bb:214:4fff:2731:b1a9/64
net0/v6           addrconf ok           2001:0db8:56::214:4fff:2731:b1a9/64
```

Suponga que dos zonas, `sandbox` y `toybox`, están utilizando las siguientes direcciones IP:

- `sandbox` – `172.0.0.3`
- `toybox` – `172.0.0.1`

Puede utilizar el comando `snoop -I` en las distintas interfaces del sistema. La información de paquetes que se visualiza depende de si usted es administrador de la zona global o de la zona no global.

EJEMPLO 1-11 Observación del tráfico en la interfaz en bucle de retorno

El ejemplo siguiente muestra la salida del comando `snoop` para la interfaz de bucle de retorno.

```
# snoop -I lo0
Using device ipnet/lo0 (promiscuous mode)
localhost -> localhost    ICMP Echo request (ID: 5550 Sequence number: 0)
localhost -> localhost    ICMP Echo reply (ID: 5550 Sequence number: 0)
```

Para generar una salida detallada, utilice la opción `-v`.

```
# snoop -v -I lo0
Using device ipnet/lo0 (promiscuous mode)
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 1 arrived at 10:40:33.68506
IPNET:  Packet size = 108 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 4
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 0
IPNET:
IP:   ----- IP Header -----
```

```
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
...
```

La compatibilidad para la observación de paquetes en la capa IP implementa un encabezado `ipnet` nuevo que precede a los paquetes que se están observando. Se indican los ID de origen y de destino. El ID `0` indica que el tráfico se genera en la zona global.

EJEMPLO 1-12 Observación del flujo de paquetes en el dispositivo `net0` en las zonas locales

El ejemplo siguiente muestra el tráfico que se produce en las distintas zonas que están en el sistema. Puede ver todos los paquetes que están asociados con las direcciones IP `net0`, incluidos los paquetes que se transfieren localmente a otras zonas. Si genera una salida detallada, también puede ver las zonas que forman parte del flujo de paquetes.

```
# snoop -I net0
Using device ipnet/net0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=62117 Syn Seq=195630514 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=62117 S=22 Syn Ack=195630515 Seq=195794440 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=62117 Ack=195794441 Seq=195630515 Len=0 Win=49152
sandbox -> toybox TCP D=62117 S=22 Push Ack=195630515 Seq=195794441 Len=20 Win=491

# snoop -I net0 -v port 22
IPNET:  ----- IPNET Header -----
IPNET:
IPNET:  Packet 5 arrived at 15:16:50.85262
IPNET:  Packet size = 64 bytes
IPNET:  dli_version = 1
IPNET:  dli_type = 0
IPNET:  dli_srczone = 0
IPNET:  dli_dstzone = 1
IPNET:
IP:   ----- IP Header -----
IP:
IP:   Version = 4
IP:   Header length = 20 bytes
IP:   Type of service = 0x00
IP:         xxx. .... = 0 (precedence)
IP:         ...0 .... = normal delay
IP:         .... 0... = normal throughput
IP:         .... .0.. = normal reliability
IP:         .... ..0. = not ECN capable transport
IP:         .... ...0 = no ECN congestion experienced
IP:   Total length = 40 bytes
IP:   Identification = 22629
IP:   Flags = 0x4
IP:         .1.. .... = do not fragment
IP:         ..0. .... = last fragment
IP:   Fragment offset = 0 bytes
IP:   Time to live = 64 seconds/hops
IP:   Protocol = 6 (TCP)
```

```
IP:   Header checksum = 0000
IP:   Source address = 172.0.0.1, 172.0.0.1
IP:   Destination address = 172.0.0.3, 172.0.0.3
IP:   No options
IP:
TCP:  ----- TCP Header -----
TCP:
TCP:  Source port = 46919
TCP:  Destination port = 22
TCP:  Sequence number = 3295338550
TCP:  Acknowledgement number = 3295417957
TCP:  Data offset = 20 bytes
TCP:  Flags = 0x10
TCP:        0... .... = No ECN congestion window reduced
TCP:        .0.. .... = No ECN echo
TCP:        ..0. .... = No urgent pointer
TCP:        ...1 .... = Acknowledgement
TCP         .... 0... = No push
TCP         .... .0.. = No reset
TCP:        .... ..0. = No Syn
TCP:        .... ...0 = No Fin
TCP:  Window = 49152
TCP:  Checksum = 0x0014
TCP:  Urgent pointer = 0
TCP:  No options
TCP:
```

En la salida anterior, el encabezado `ipnet` indica que el paquete proviene de la zona global (ID `0`) a `sandbox` (ID `1`).

EJEMPLO 1-13 Observación del tráfico de red mediante la identificación de una zona

El siguiente ejemplo muestra cómo observar el tráfico de red mediante la identificación de la zona, que es extremadamente útil en los sistemas que tienen varias zonas. En la actualidad, únicamente se puede identificar la zona con el ID de zona. No se admite el uso del comando `snoop` con nombres de zona.

```
# snoop -I hme0 sandboxsnoop -I net0 sandbox
Using device ipnet/hme0 (promiscuous mode)
toybox -> sandbox TCP D=22 S=61658 Syn Seq=374055417 Len=0 Win=49152 Options=<mss
sandbox -> toybox TCP D=61658 S=22 Syn Ack=374055418 Seq=374124525 Len=0 Win=49152
toybox -> sandbox TCP D=22 S=61658 Ack=374124526 Seq=374055418 Len=0 Win=49152
```

Observación del tráfico de red con los comandos `ipstat` y `tcpstat`

Se incluyeron dos nuevos comandos para observar varios tipos de tráfico de red en un servidor en esta versión: `ipstat` y `tcpstat`.

El comando ipstat se utiliza para recopilar y registrar estadísticas sobre el tráfico IP en un servidor basado en el modo de salida y el orden de clasificación seleccionados que se especifican en la sintaxis de comando. Este comando permite observar el tráfico de red en la capa IP, agregado en el origen, el destino, el protocolo de nivel superior y la interfaz. Utilice este comando cuando desee observar la cantidad de tráfico entre un servidor y otros servidores.

El comando tcpstat se utiliza para recopilar y registrar estadísticas sobre el tráfico TCP y UDP en un servidor basado en el modo de salida y el orden de clasificación seleccionados que se especifican en la sintaxis de comando. Este comando permite observar el tráfico de red en la capa de transporte, específicamente para TCP y UDP. Además de las direcciones IP de origen y destino, puede observar los puertos TCP o UDP de origen y destino, el PID del proceso que envía o recibe tráfico y el nombre de la zona en la que se está ejecutando el proceso.

A continuación, incluimos algunas de las formas en que puede utilizar el comando tcpstat:

- Identifique el mayor origen de tráfico TCP y UDP en un servidor.
- Examine el tráfico que se está generado por un proceso concreto.
- Examine el tráfico que se generada a partir de una zona concreta.
- Determine qué proceso está enlazado a un puerto local.

Nota - La lista anterior no es exhaustiva. Hay otras formas en que puede utilizar el comando tcpstat. Para obtener más información, consulte la página del comando man tcpstat(1M).

Para utilizar los comandos ipstat y tcpstat, se requiere uno de los siguientes privilegios:

- Asumir el rol root.
- Tener el privilegio dtrace_kernel asignado explícitamente
- Tener asignado el perfil de derechos de la gestión de red o la observabilidad de red

Los siguientes ejemplos muestran diferentes formas en las que puede utilizar estos dos comandos para observar el tráfico de red. Para obtener más información, consulte las páginas del comando man tcpstat(1M) and ipstat(1M).

El ejemplo siguiente muestra la salida del comando ipstat cuando se ejecuta con la opción -c. Utilice la opción -c para imprimir informes nuevos después de informes anteriores, sin sobrescribir el informe anterior. El número 3 en este ejemplo especifica el intervalo para mostrar datos, que es el mismo que si el comando se ha llamado como ipstat 3.

```
# ipstat -c 3
SOURCE                  DEST                    PROTO    INT       BYTES
zucchini                antares                 TCP      net0      72.0
zucchini                antares                 SCTP     net0      64.0
antares                 zucchini                SCTP     net0      56.0
amadeus.foo.example.com 10.6.54.255             UDP      net0      40.0
antares                 zucchini                TCP      net0      40.0
zucchini                antares                 UDP      net0      16.0
antares                 zucchini                UDP      net0      16.0
```

```
Total: bytes in: 192.0  bytes out: 112.0
```

En comparación, el ejemplo siguiente muestra la salida del comando `tcpstat` cuando se utiliza con la opción `-c`:

```
# tcpstat -c 3
ZONE         PID PROTO  SADDR             SPORT DADDR             DPORT   BYTES
global    100680 UDP    antares           62763 agamemnon          1023   76.0
global    100680 UDP    antares             775 agamemnon          1023   38.0
global    100680 UDP    antares             776 agamemnon          1023   37.0
global    100680 UDP    agamemnon          1023 antares           62763   26.0
global    104289 UDP    zucchini          48655 antares            6767   16.0
global    104289 UDP    clytemnestra      51823 antares            6767   16.0
global    104289 UDP    antares            6767 zucchini          48655   16.0
global    104289 UDP    antares            6767 clytemnestra      51823   16.0
global    100680 UDP    agamemnon          1023 antares             776   13.0
global    100680 UDP    agamemnon          1023 antares             775   13.0
global    104288 TCP    zucchini          33547 antares            6868    8.0
global    104288 TCP    clytemnestra      49601 antares            6868    8.0
global    104288 TCP    antares            6868 zucchini          33547    8.0
global    104288 TCP    antares            6868 clytemnestra      49601    8.0
Total: bytes in: 101.0  bytes out: 200.0
```

Los siguientes ejemplos adicionales muestran otras formas en las que puede observar el tráfico en la red mediante los comandos `ipstat` y `tcpstat`.

EJEMPLO 1-14 Observación de los cinco flujos de tráfico IP más activos con el comando `ipstat`

En el ejemplo siguiente se muestran los cinco flujos de tráfico IP más activos. La opción `-l` *nlines* especifica cuántas líneas de datos utilizar para la salida por informe.

```
# ipstat -l 5
SOURCE                    DEST                      PROTO    IFNAME     BYTES
charybdis.foo.example.com  achilles.exampl           UDP      net0        6.6K
eratosthenes.example.com   aeneas.example.c          TCP      tun0        6.1K
achilles.exampl            charybdis.foo.example.com  UDP      net0       964.0
aeneas.example.c           eratosthenes.example.com   TCP      tun0       563.0
odysseus.example.          255.255.255.255           UDP      net0        66.0
Total: bytes in: 12.6K bytes out:  2.2K
```

EJEMPLO 1-15 Visualización de una indicación de hora mediante el comando `ipstat`

El siguiente ejemplo informa del tráfico IP superior con un registro de hora en formato de fecha estándar (`-d d`). Puede especificar que el registro de hora se imprima en segundos u hora de UNIX (`-d u`). El intervalo está definido en 10 segundos.

```
# ipstat -d d -c 10
Monday, March 26, 2012 08:34:07 PM EDT
SOURCE                    DEST                      PROTO    IFNAME     BYTES
charybdis.foo.example.com  achilles.exampl           UDP      net0       15.1K
eratosthenes.example.com   aeneas.example.c          TCP      tun0       13.9K
```

```
achilles.exampl            charybdis.foo.example.com  UDP      net0        2.4K
aeneas.example.c           eratosthenes.example.com   TCP      tun0        1.5K
odysseus.example.          255.255.255.255            UDP      net0       66.0
cassiopeia.foo.example.com aeneas.example.c           TCP      tun0       29.0
aeneas.example.c           cassiopeia.foo.example.com TCP      tun0       20.0
Total: bytes in: 29.1K bytes out:  3.8K
```

EJEMPLO 1-16 Observación de los cinco flujos de tráfico más activos con el comando `tcpstat`

En el ejemplo siguiente se muestran los cinco flujos de tráfico TCP más activos para un servidor:

```
# tcpstat -l 5
ZONE            PID PROTO  SADDR            SPORT DADDR            DPORT   BYTES
global        28919 TCP    achilles.exampl  65398 aristotle.exampl   443   33.0
zone1          6940 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
zone1          6940 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
global         8350 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
global         8350 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
Total: bytes in: 16.0  bytes out: 49.0
```

EJEMPLO 1-17 Visualización de información de registro de hora con el comando `tcpstat`

En el ejemplo siguiente, el comando `tcpstat` se utiliza para mostrar información de registro de hora para el tráfico de red TCP en un servidor en formato de fecha estándar:

```
# tcpstat -d d -c 10
Saturday, March 31, 2012 07:48:05 AM EDT
ZONE            PID PROTO  SADDR            SPORT DADDR            DPORT   BYTES
global         2372 TCP    penelope.example 58094 polyphemus.examp    80   37.0
zone1          6940 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
zone1          6940 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
global         8350 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
global         8350 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
Total: bytes in: 16.0  bytes out: 53.0
```

♦ ♦ ♦ CAPÍTULO 2

Acerca de la administración de IPMP

Rutas múltiples de red IP (IPMP, IP Network Multipathing) es una tecnología de capa 3 (L3) que permite agrupar varias interfaces IP en una única interfaz lógica. Con funciones tales como la detección de fallos, failover de acceso transparente y la expansión de carga de paquetes, IPMP mejora el rendimiento de la red, ya que garantiza que la red esté siempre disponible en el sistema.

Este capítulo se divide en los siguientes apartados:

Nota - En este capítulo y en Capítulo 3, Administración de IPMP, toda referencia al término *interfaz* significa específicamente *interfaz IP*. A menos que se indique explícitamente un uso diferente del término, como una tarjeta de interfaz de red (NIC), el término siempre hace referencia a la interfaz que se configura en la capa IP.

Novedades de IPMP

Las siguientes funciones son nuevas o se han cambiado en esta versión.

Cambios en la configuración de IPMP

En esta versión, IPMP funciona de forma diferente que en Oracle Solaris 10. Un cambio importante es que las interfaces IP ahora se agrupan en una interfaz IP *virtual*, por ejemplo `ipmp0`. La interfaz IP virtual sirve a todas las direcciones IP de datos, mientras que las direcciones de prueba que se utilizan para la detección de fallos basada en sondeos se asignan a una interfaz subyacente, como `net0`. Para obtener más información, consulte "Cómo funciona IPMP" [55].

Consulte el siguiente flujo de trabajo general al realizar la transición de la configuración de IPMP existente al nuevo modelo de IPMP:

1. Asegúrese de que el perfil `DefaultFixed` esté activado en el sistema antes de configurar IPMP. Consulte “Activación y desactivación de perfiles” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.
2. Asegúrese de que las direcciones MAC en sistemas basados en SPARC sean únicas. Consulte “Cómo asegurarse de que la dirección MAC de cada interfaz sea única” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.
3. Utilice el comando `dladm` para configurar enlaces de datos. Para utilizar los mismos dispositivos de red físicos en su configuración de IPMP, primero deberá identificar los enlaces de datos que están asociados a cada instancia de dispositivo:

```
# dladm show-phys
LINK              MEDIA               STATE      SPEED  DUPLEX    DEVICE
net1              Ethernet            unknown    0      unknown   bge1
net0              Ethernet            up         1000   full      bge0
net2              Ethernet            unknown    1000   full      e1000g0
net3              Ethernet            unknown    1000   full      e1000g1
```

4. Si ya utilizó `e1000g0` y `e1000g1` para configuración de IPMP, ahora debe utilizar `net2` y `net3`. Tenga en cuenta que los enlaces de datos se pueden usar en enlaces físicos y también en agregaciones, VLAN, VNIC, etc. Para obtener más información, consulte “Visualización de los enlaces de datos de un sistema” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.
5. Utilice el comando `ipadm` para realizar las siguientes tareas:
   - Configurar capa de red.
   - Crear interfaces IP.
   - Agregar la interfaz IP a un grupo IPMP.
   - Agregar direcciones de datos a un grupo IPMP.

Para obtener más información sobre los cambios de comandos de la administración de la red en esta versión, consulte “Cambios de comandos de administración de red” de “Transición de Oracle Solaris 10 a Oracle Solaris 11.2 ”.

Compatibilidad de IPMP en Oracle Solaris

IPMP ofrece la siguiente compatibilidad:

- IPMP permite configurar varias interfaces IP en un mismo grupo que se denomina grupo IPMP. En su conjunto, el grupo IPMP con su varias interfaces IP subyacentes se representa como una sola *interfaz IPMP*. Esta interfaz se trata como cualquier otra interfaz en la capa IP de la pila de red. Todas las tareas administrativas, tablas de enrutamiento, tablas de protocolo de resolución de direcciones (ARP), reglas de cortafuegos y otros procedimientos relacionados con IP funcionan con un grupo IPMP haciendo referencia a la interfaz IPMP.

- El sistema gestiona la distribución de direcciones de datos entre interfaces activas subyacentes. Cuando se crea el grupo IPMP, las direcciones de datos pertenecen a la interfaz IPMP como una *agrupación de direcciones*. El núcleo automáticamente y aleatoriamente vincula las direcciones de datos a las interfaces activas del grupo.
- Principalmente, utilice el comando `ipmpstat` para obtener información sobre los grupos IPMP. Este comando proporciona información acerca de todos los aspectos de la configuración de IPMP, como las interfaces IP subyacentes del grupo, las direcciones de datos y las pruebas, los tipos de detección de fallos en uso y las interfaces que tuvieron errores. Las funciones del comando `ipmpstat`, las opciones que puede utilizar y la salida que cada opción genera se describen en “Supervisión de información de IPMP” [89].
- Puede asignar una interfaz IPMP un nombre personalizado para identificar el grupo IPMP más fácilmente. Consulte “Configuración de grupos IPMP” [69].

Beneficios de usar IPMP

Distintos factores pueden hacer que una interfaz se vuelva inutilizable; por ejemplo, que se produzca un fallo de la interfaz o que las interfaces se desconecten por tareas de mantenimiento. Sin IPMP, no se puede contactar al sistema mediante ninguna de las direcciones IP asociadas con la interfaz no utilizable. Además, las conexiones existentes que utilizan esas direcciones IP se interrumpen.

Con IPMP, una o más interfaces IP se pueden configurar en un *grupo IPMP*. El grupo funciona como una interfaz IP con direcciones de datos para enviar o recibir tráfico de la red. Si una interfaz subyacente del grupo falla, las direcciones de datos se redistribuyen entre las restantes interfaces activas subyacentes del grupo. Por lo tanto, el grupo mantiene conectividad de red a pesar de un fallo de la interfaz. Con IPMP, la conectividad de red siempre está disponible, siempre que un mínimo de una interfaz pueda ser utilizado por el grupo.

IPMP mejora el rendimiento global de la red al expandir automáticamente el tráfico de la red saliente por un conjunto de interfaces del grupo IPMP. Este proceso se denomina *expansión de carga saliente*. El sistema también indirectamente controla la expansión de carga entrante realizando una selección de direcciones de origen para los paquetes cuya dirección IP de origen no fue especificada por la aplicación. Sin embargo, si una aplicación ha elegido de manera explícita una dirección de IP de origen, el sistema no cambia la dirección de origen.

Tenga en cuenta la siguiente información importante sobre las políticas que aplica IPMP para la expansión de carga entrante y saliente:

- Para las direcciones IP en enlaces, IPMP selecciona aleatoriamente una única interfaz IP activa para acceder a esa dirección IP. En caso de que haya varias conexiones diferentes para una determinada dirección IP en enlaces, todas esas conexiones utilizarán la misma interfaz IP saliente. Además, si la interfaz IP cambia a lo largo del tiempo, el cambio afecta a todas las conexiones a esa dirección IP.
- Para las direcciones IP fuera de enlaces, IPMP selecciona aleatoriamente una única interfaz IP para acceder a la dirección IP del enrutador IP en enlaces a través del cual se accederá a

la dirección IP fuera de enlaces. Esta política significa, de hecho, que todas las direcciones IP fuera de enlaces para un determinado grupo IPMP utilizan una única interfaz IP.

Nota - Las políticas actuales de expansión de carga entrante y saliente están sujetas a cambios.

Las agregaciones de enlaces realizan funciones similares a IPMP para mejorar el rendimiento y disponibilidad de la red. Para comparar estas dos tecnologías, consulte Apéndice A, “Agregaciones de enlaces e IPMP: comparación de funciones” de “Gestión de enlaces de datos de red en Oracle Solaris 11.2 ”.

Reglas de uso de IPMP

La configuración del grupo IPMP se determina según su configuración específica del sistema.

Tenga en cuenta las siguientes reglas para la configuración de IPMP:

1. Varias interfaces IP que están en la misma LAN deben configurarse en un grupo IPMP. Un LAN hace referencia básicamente a una variedad de configuraciones de redes locales, incluidas las VLAN y las redes locales inalámbricas y cableadas con nodos que pertenezcan al mismo dominio de difusión de capa de enlace.

Nota - No se admiten varios grupos IPMP en el mismo dominio de emisión de capa de enlace (L2). Normalmente, un dominio de difusión L2 se asigna a una subred específica. Por lo tanto, debe configurar sólo un grupo IPMP por subred. Tenga en cuenta también que se aplican algunas excepciones a esta regla, por ejemplo, en el caso de que sean proporcionadas por ciertos sistemas de ingeniería de Oracle. Para obtener aclaraciones, póngase en contacto con el representante de asistencia de Oracle.

---

2. Las interfaces IP subyacentes de un grupo IPMP no deben abarcar diferentes LAN.

   Por ejemplo, supongamos que un sistema con tres interfaces está conectado a dos LAN separadas. Dos interfaces IP se conectan a una LAN, mientras que una interfaz IP única se conecta a la otra LAN. En este caso, las dos interfaces IP que se conectan a la primera LAN deben estar configuradas como un grupo IPMP, tal y como exige la primera regla. De conformidad con la segunda regla, la interfaz IP única que se conecta a la segunda LAN no puede convertirse en miembro de ese grupo IPMP. No es necesaria ninguna configuración IPMP de la interfaz IP única. Sin embargo, puede configurar la interfaz única en un grupo IPMP para supervisar la disponibilidad de la interfaz. La configuración de IPMP de interfaz única se trata en profundidad en “Tipos de configuraciones de interfaces IPMP” [54].

   Tenga en cuenta otro caso en que el enlace a la primera LAN consta de tres interfaces IP mientras que el otro enlace consta de dos interfaces. Esta configuración requiere la configuración de dos grupos IPMP: un grupo de tres interfaces que se conecta a la primera LAN y un grupo de dos interfaces que se conecta a la segunda.

3. Todas las interfaces del mismo grupo deben tener configurados los mismos módulos STREAMS en el mismo orden. Al planificar un grupo IPMP, en primer lugar, compruebe el orden de los módulos STREAMS en todas las interfaces del posible grupo IPMP, a continuación, inserte los módulos de cada interfaz en el orden estándar para el grupo IPMP. Para imprimir una lista de los módulos STREAMS, use el comando `ifconfig` *interface* `modlist`. Por ejemplo, la siguiente es la salida de `ifconfig` para una interfaz `net0`:

```
# ifconfig net0 modlist
0 arp
1 ip
2 e1000g
```

Como se muestra en la salida anterior, las interfaces existen normalmente como controladores de red directamente debajo del módulo IP. Estas interfaces no deberían requerir ninguna configuración adicional. Sin embargo, determinadas tecnologías se insertan como módulos STREAMS entre el módulo IP y el controlador de red. Si un módulo STREAMS tiene estado, puede producirse un comportamiento inesperado en failover, aunque se inserte el mismo módulo en todas las interfaces de un grupo. Sin embargo, puede utilizar módulos STREAMS sin estado, siempre y cuando los inserte en el mismo orden en todas las interfaces del grupo IPMP.

El ejemplo siguiente muestra el comando que puede utilizar para enviar los módulos de cada interfaz en el orden estándar para el grupo IPMP:

```
# ifconfig net0 modinsert vpnmod@3
```

Para obtener instrucciones paso a paso sobre la planificación de un grupo IPMP, consulte Cómo planificar un grupo IPMP [70].

Componentes IPMP

A continuación, se mencionan los componentes de software de IPMP:

- **Daemon de rutas múltiples** (`in.mpathd`): detecta fallos y efectúa reparaciones. El daemon realiza una detección de fallos basada en enlaces y una detección de fallos basada en sondeos si las direcciones de prueba se configuran para las interfaces subyacentes. Según el tipo de método de detección de fallos que se emplea, el daemon establece los indicadores adecuados en la interfaz o los borra para indicar si la interfaz ha fallado o se ha reparado. De manera opcional, también puede configurar el daemon para supervisar la disponibilidad de todas las interfaces, incluidas aquellas que no se han configurado para que pertenezcan a un grupo IPMP. Para obtener una descripción de detección de fallos, consulte “Detección de fallos en IPMP” [62].

  El daemon `in.mpathd` controla también la designación de interfaces activas del grupo IPMP. El daemon intenta mantener el mismo número de interfaces activas que se configuró originalmente cuando se creó el grupo IPMP. Por lo tanto, `in.mpathd` activa o desactiva interfaces subyacentes según sea necesario para mantener la coherencia con la política

configurada del administrador. Para obtener más información sobre el modo en que el daemon `in.mpathd` gestiona la activación de interfaces subyacentes, consulte “Cómo funciona IPMP” [55]. Para obtener más información sobre el daemon, consulte la página del comando man `in.mpathd`(1M).

- **Módulo de núcleo IP**: gestiona la expansión de carga saliente mediante la distribución del conjunto de direcciones de datos IP disponibles en el grupo IPMP por medio del conjunto de interfaces IP subyacentes disponibles en el grupo. El módulo también realiza una selección de direcciones de origen para gestionar la expansión de carga entrante. Ambos roles del módulo mejoran el rendimiento del tráfico de red.
- **Archivo de configuración de IPMP** (`/etc/default/mpathd`): define el comportamiento el daemon.

  Personalice el archivo para establecer los siguientes parámetros:

  - Interfaces de destino que se deben sondear cuando se ejecuta la detección de fallos basada en sondeos
  - Tiempo disponible para sondear un destino a fin de detectar fallos
  - Estado con el que se marca una interfaz que tenía fallos una vez que se ha reparado
  - Ámbito de interfaces IP por supervisar a fin de determinar si se incluyen interfaces IP en el sistema que no estén configuradas para pertenecer a los grupos IPMP

  Para obtener más información sobre cómo modificar el archivo de configuración, consulte Cómo configurar el comportamiento del daemon IPMP [87].
- **Comando** `ipmpstat`: proporciona distintos tipos de información sobre el estado de IPMP en su totalidad. La herramienta también muestra otra información sobre las interfaces IP subyacentes para cada grupo IPMP y también datos y direcciones de prueba que se han configurado para el grupo. Para obtener más información sobre este comando, consulte “Supervisión de información de IPMP” [89] y la página del comando man `ipmpstat`(1M).

Tipos de configuraciones de interfaces IPMP

Una configuración de IPMP típica consta de dos o más interfaces físicas en el mismo sistema que están conectadas a la misma LAN.

Estas interfaces pueden pertenecer a un grupo IPMP en cualquiera de las siguientes configuraciones:

- **Configuración activa/activa**: un grupo IPMP en el que todas las interfaces subyacentes están activas. Una *interfaz activa* es una interfaz IP que está actualmente disponible para ser utilizada por el grupo IPMP.

Nota - De manera predeterminada, una interfaz subyacente se vuelve activa cuando configura la interfaz para que sea parte de un grupo IPMP.

- **Configuración activa/en espera**: un grupo IPMP en el que al menos una interfaz está configurada administrativamente como una *interfaz en espera*. Aunque está inactiva, la interfaz IP en espera es supervisada por el daemon de rutas múltiples para realizar un seguimiento de la disponibilidad de la interfaz, en función de cómo está configurada la interfaz. Si la notificación de fallos por enlaces es admitida por la interfaz, se utiliza la detección de fallos basada en enlaces. Si la interfaz está configurada con una dirección de prueba, también se utiliza la detección de fallos basada en sondeos. Si una interfaz activa falla, la interfaz en espera se implementa automáticamente según sea necesario. Puede configurar tantas interfaces en espera como necesite para un grupo IPMP.

También puede configurar una única interfaz en su propio grupo IPMP. El grupo IPMP de interfaz única se comporta del mismo modo que un grupo IPMP con varias interfaces. Sin embargo, esta configuración de IPMP no proporciona alta disponibilidad para el tráfico de la red. Si la interfaz subyacente falla, el sistema pierde toda capacidad de enviar o recibir tráfico. La finalidad de configurar un grupo de IPMP de una sola interfaz es la de supervisar la disponibilidad de la interfaz mediante la detección de fallos. Mediante la configuración de una dirección de prueba en la interfaz, el daemon de rutas múltiples puede realizar un seguimiento de la interfaz mediante la detección de fallos basada en sondeos.

Normalmente, una configuración de grupo IPMP de una sola interfaz se utiliza junto con otras tecnologías que tienen capacidades más amplias de failover, tales como el software de Oracle Solaris Cluster. El sistema puede seguir supervisando el estado de la interfaz subyacente, pero el software de Oracle Solaris Cluster proporciona la funcionalidad para garantizar la disponibilidad de la red cuando se produce un fallo. Para obtener más información sobre el software de Oracle Solaris Cluster, consulte “Oracle Solaris Cluster Concepts Guide ”.

Un grupo IPMP sin interfaces subyacentes también puede existir, como un grupo cuyas interfaces subyacentes se han eliminado. El grupo IPMP no se destruye, pero el grupo no se puede utilizar para enviar ni recibir tráfico. Cuando las interfaces IP subyacentes se ponen en línea para el grupo, las direcciones de datos de la interfaz IPMP se asignan a estas interfaces, y el sistema reanuda el alojamiento de tráfico de la red.

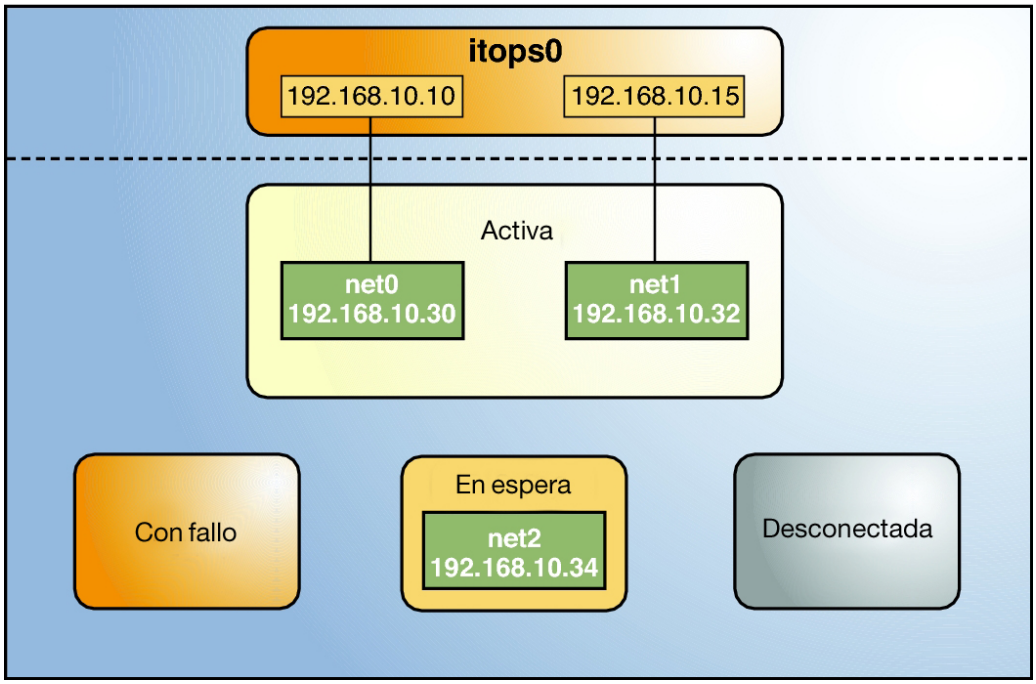
Cómo funciona IPMP

IPMP mantiene la disponibilidad de la red mediante un intento de mantener la misma cantidad de interfaces activas y en espera que había en la configuración original cuando se creó el grupo IPMP.

La detección de fallos IPMP puede estar basada en enlaces, en sondeos o en ambos para determinar la disponibilidad de una interfaz IP subyacente específica del grupo. Si IPMP determina que una interfaz subyacente ha fallado, esa interfaz se marca como con fallos y ya no es utilizable. La dirección IP de datos asociada con la interfaz con fallos se redistribuye a otra interfaz en funcionamiento del grupo. Si está disponible, una interfaz en espera también se implementa para mantener el número original de interfaces activas.

Considere un grupo IPMP de tres interfaces, `itops0`, con una configuración activa/en espera como se ilustra en la siguiente figura.

FIGURA 2-1 Configuración activa/en espera de IPMP



El grupo IPMP `itops0` se configura del siguiente modo:

- Dos direcciones de datos se asignan al grupo: `192.168.10.10` y `192.168.10.15`.
- Dos interfaces subyacentes se configuran como interfaces activas y se asignan nombres de enlace flexibles: `net0` y `net1`.
- El grupo posee una sola interfaz en espera, también con un nombre de enlace flexible: `net2`.
- Se utiliza la detección de fallos basada en sondeos y, por consiguiente, las interfaces activas y en espera se configuran con direcciones de prueba, de la siguiente manera:
  - `net0`: `192.168.10.30`
  - `net1`: `192.168.10.32`
  - `net2`: `192.168.10.34`

Nota - Las áreas Activa, Fuera de línea, En espera y Con fallo en Figura 2-1, “Configuración activa/en espera de IPMP”, Figura 2-2, “Fallo de la interfaz en IPMP”, Figura 2-3, “Fallo de interfaz en espera en IPMP” y Figura 2-4, “Proceso de recuperación de IPMP” indican el estado de las interfaces subyacentes solamente, no el de ubicaciones físicas. Ningún movimiento físico de interfaces o direcciones, ni ninguna transferencia de interfaces IP, se produce en esta implementación de IPMP. Las áreas sirven solamente para mostrar cómo una interfaz subyacente cambia de estado como resultado de un fallo o reparación.

Puede usar el comando `ipmpstat` con diferentes opciones para mostrar tipos determinados de información sobre grupos IPMP existentes. Para ver ejemplos adicionales, consulte “Supervisión de información de IPMP” [89].

El siguiente comando muestra información sobre la configuración IPMP en Figura 2-1, “Configuración activa/en espera de IPMP”:

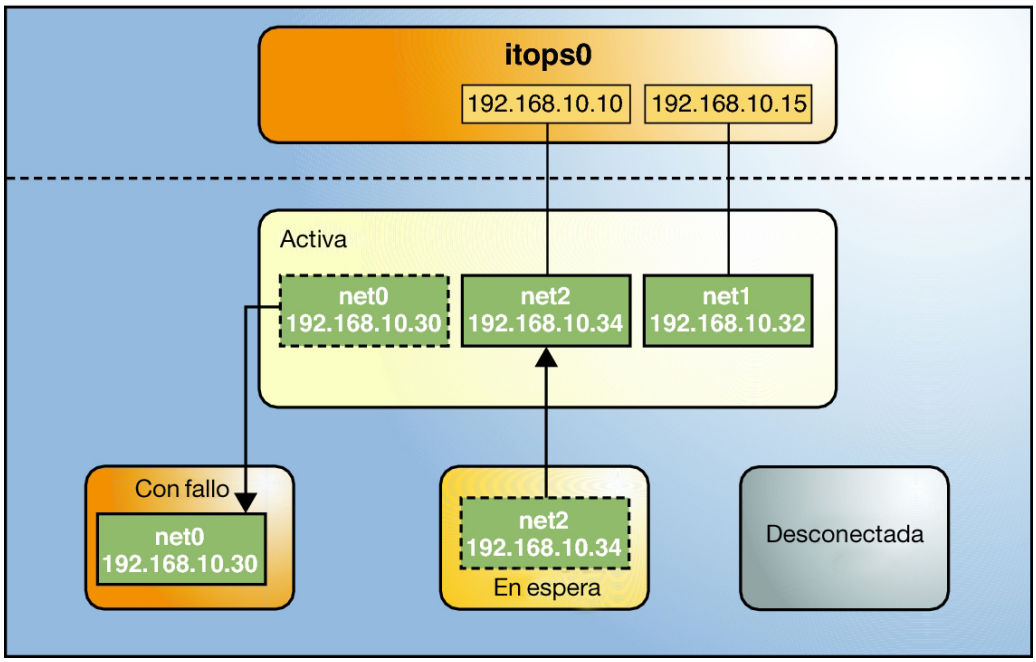
```
# ipmpstat -g
GROUP     GROUPNAME     STATE     FDT        INTERFACES
itops0    itops0        ok        10.00s     net1 net0 (net2)
```

Debe ver información sobre las interfaces subyacentes del grupo de la siguiente manera:

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS      LINK        PROBE     STATE
net0        yes        itops0    -------    up          ok        ok
net1        yes        itops0    --mb---    up          ok        ok
net2        no         itops0    is-----    up          ok        ok
```

IPMP mantiene la disponibilidad de la red administrando las interfaces subyacentes para mantener el número original de interfaces activas. Por lo tanto, si `net0` falla, se implementa `net2` para garantizar que el grupo IPMP siga teniendo dos interfaces activas. La activación de `net2` se muestra en la siguiente figura.

FIGURA 2-2 Fallo de la interfaz en IPMP



Nota - La asignación uno a uno de direcciones de datos a interfaces activas en la Figura 2-2, "Fallo de la interfaz en IPMP" solo sirve para simplificar la ilustración. El módulo de núcleo IP puede asignar direcciones de datos aleatoriamente sin que sea necesario adherirse a una relación de uno a uno entre direcciones de datos e interfaces.

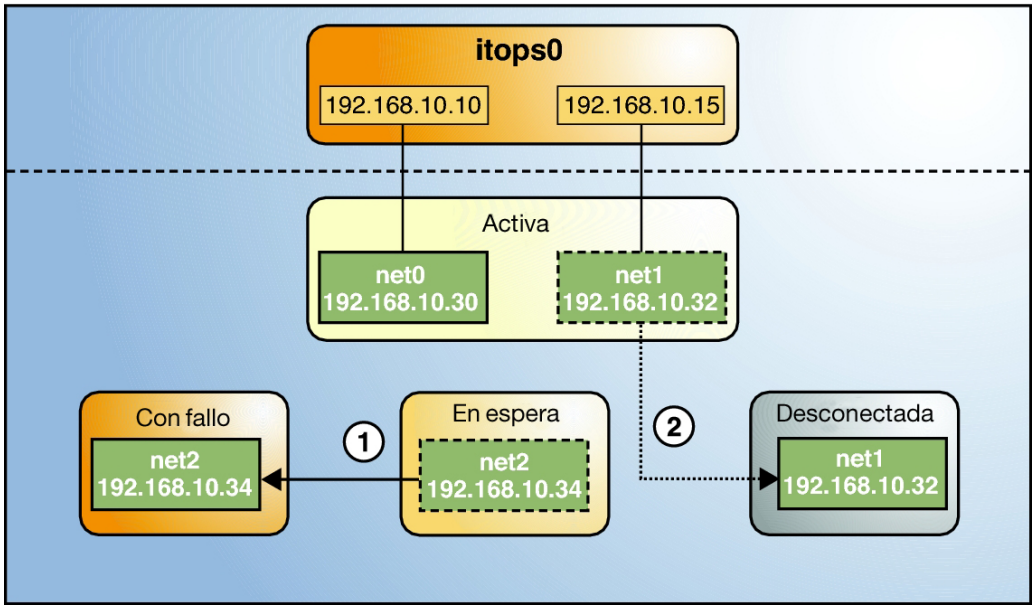
El comando `ipmpstat` muestra la información en la figura como se indica a continuación:

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS       LINK        PROBE     STATE
net0        no         itops0    -------     up          failed    failed
net1        yes        itops0    --mb---     up          ok        ok
net2        yes        itops0    -s-----     up          ok        ok
```

Una vez que `net0` se ha reparado, vuelve a su estado como una interfaz activa. A su vez, `net2` vuelve a su estado en espera original.

Otro escenario de fallo se muestra en la Figura 2-3, "Fallo de interfaz en espera en IPMP", donde la interfaz en espera `net2` falla (1). Más tarde, el administrador desconecta una interfaz activa, `net1` (2). El resultado es que el grupo IPMP se deja con una única interfaz en funcionamiento, `net0`.

FIGURA 2-3 Fallo de interfaz en espera en IPMP

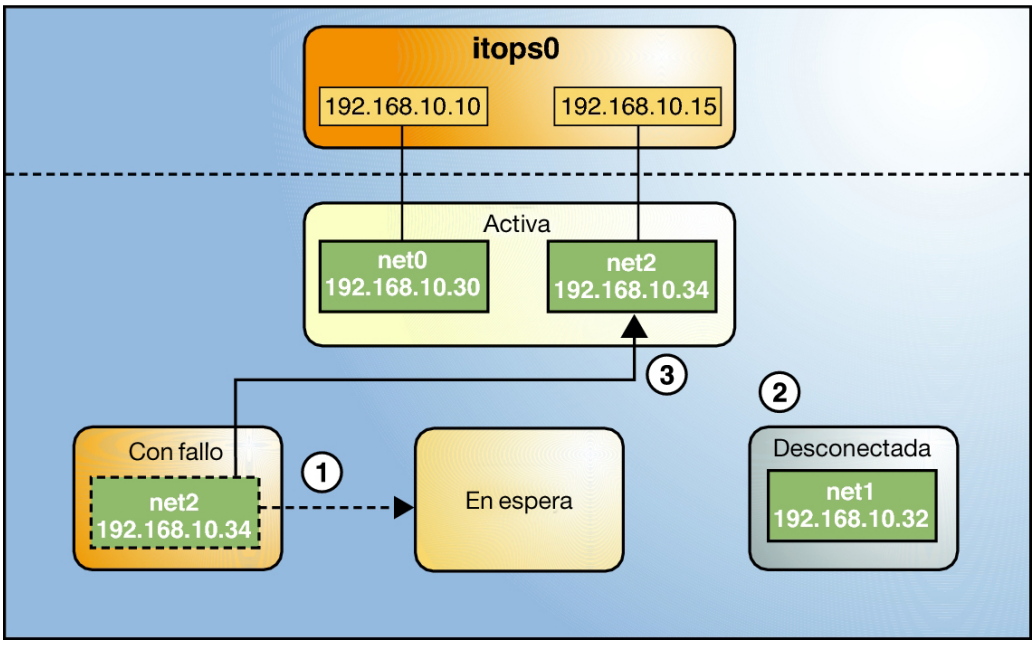


El comando `ipmpstat` muestra la información en la figura como se indica a continuación:

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS       LINK        PROBE     STATE
net0        yes        itops0    -------     up          ok        ok
net1        no         itops0    --mb-d-     up          ok        offline
net2        no         itops0    is-----     up          failed    failed
```

Para este fallo en particular, el proceso de recuperación después de que una interfaz se repara de manera diferente. El proceso de restauración depende del número original de interfaces activas del grupo IPMP en comparación con la configuración después de la reparación. La siguiente figura representa el proceso de recuperación.

FIGURA 2-4 Proceso de recuperación de IPMP



En la Figura 2-4, “Proceso de recuperación de IPMP”, cuando se repara `net2`, vuelve a su estado original como una interfaz en espera (1). Sin embargo, el grupo IPMP no refleja el número original de dos interfaces activas, ya que `net1` sigue estando sin conexión (2). Por lo tanto, IPMP implementa `net2` como una interfaz activa en su lugar (3).

La utilidad `ipmpstat` mostrará el escenario de IPMP posterior a la reparación, de la siguiente manera:

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS       LINK        PROBE     STATE
net0        yes        itops0    -------     up          ok        ok
net1        no         itops0    --mb-d-     up          ok        offline
net2        yes        itops0    -s-----     up          ok        ok
```

Una secuencia de restauración similar se produce si el fallo implica una interfaz activa que también está configurada en modo `FAILBACK=no`, donde una interfaz activa con fallos no se revierte automáticamente al estado activo después de la reparación. Supongamos que `net0` en la Figura 2-2, “Fallo de la interfaz en IPMP” se configura en modo `FAILBACK=no`. En ese modo, una interfaz `net0` reparada se convierte en una interfaz en espera, incluso si originalmente era una interfaz activa. La interfaz `net2` permanece activa para mantener el número original de dos interfaces activas del grupo IPMP.

El comando `ipmpstat` muestra la información de recuperación, como se indica a continuación:

```
# ipmpstat -i
INTERFACE   ACTIVE     GROUP     FLAGS      LINK        PROBE     STATE
net0        no         itops0    i------    up          ok        ok
net1        yes        itops0    --mb---    up          ok        ok
net2        yes        itops0    -s-----    up          ok        ok
```

Para obtener más información sobre este tipo de configuración, consulte “Modo FAILBACK=no” [66].

Direcciones IPMP

Puede configurar la detección de fallos IPMP en redes IPv4 y en redes IPv4 e IPv6 de pila doble. Las interfaces que configura con IPMP admiten dos tipos de direcciones: direcciones de datos y direcciones de prueba. Las direcciones IP residen en la interfaz IPMP (el grupo) *únicamente* y se especifican como direcciones de datos. Las direcciones de prueba son direcciones IP que residen en las interfaces subyacentes.

Direcciones de datos

Las direcciones de datos son direcciones IPv4 e IPv6 convencionales que se asignan a una interfaz IP dinámicamente en el momento del inicio mediante el servidor DHCP o manualmente mediante el comando `ipadm`. Las direcciones de datos se asignan a la interfaz IPMP (o grupo) *únicamente*. El tráfico de paquetes IPv4 estándar y el tráfico de paquetes IPv6 (si es aplicable), se consideran *tráfico de datos*. El flujo de tráfico de datos utiliza las direcciones de datos que se encuentran alojadas en la interfaz IPMP y fluyen por las interfaces activas de ese grupo IPMP.

Direcciones de prueba

Las *direcciones de prueba* son direcciones específicas IPMP que utiliza el daemon `in.mpathd` para realizar la detección de fallos basada en sondeos y las reparaciones. Las direcciones de prueba también se pueden asignar dinámicamente mediante el servidor DHCP o manualmente mediante el comando `ipadm`. Asigne direcciones de prueba para las interfaces subyacentes del grupo IPMP. Cuando una interfaz subyacente falla, la dirección de prueba de la interfaz continúa siendo utilizada por el daemon `in.mpathd` para la detección de fallos basada en sondeos para comprobar la reparación subsecuente de la interfaz.

Nota - Configure direcciones de prueba si se va a utilizar la detección de fallos basada en sondeos. De lo contrario, puede activar el sondeo transitivo para detectar fallos sin utilizar direcciones de prueba. Para obtener más información sobre la detección de fallos basada en sondeos con o sin direcciones de prueba, consulte "Detección de fallos basada en sondeos" [63].

En implementaciones de IPMP anteriores, las direcciones de prueba debían estar marcadas como `DEPRECATED` para evitar que se utilizaran aplicaciones especialmente durante fallos de la interfaz. En la implementación actual, las direcciones de prueba residen en las interfaces subyacentes. Por lo tanto, estas direcciones ya no pueden ser utilizadas accidentalmente por aplicaciones que son independientes de IPMP. Sin embargo, para asegurarse de que estas direcciones no se tendrán en cuenta como un posible origen para los paquetes de datos, el sistema marca de manera automática las direcciones con el indicador `NOFAILOVER` y también con `DEPRECATED`.

Puede utilizar cualquier dirección IPv4 de la subred como dirección de prueba. Dado que las direcciones IPv4 son un recurso limitado para muchos sitios, puede utilizar direcciones privadas RFC 1918 no enrutables como direcciones de prueba. Observe que el daemon `in.mpathd` intercambia sólo sondeos ICMP con otros hosts que se encuentran en la misma subred que la dirección de prueba. Si utiliza las direcciones de prueba RFC 1918, debe configurar otros sistemas, preferiblemente enrutadores, en la red con direcciones en la subred RFC 1918 pertinente. De este modo, el daemon `in.mpathd` podrá intercambiar correctamente los sondeos con los sistemas de destino. Para obtener más información acerca de direcciones privadas RFC 1918, consulte RFC 1918, Address Allocation for Private Internets (http://www.rfc-editor.org/rfc/rfc1918.txt).

La única dirección de prueba IPv6 válida es la dirección local de enlace de una interfaz física. No necesita una dirección IPv6 aparte para que cumpla la función de dirección de prueba IPMP. La dirección local de enlace IPv6 se basa en la dirección de control de acceso de medios (MAC) de la interfaz. Las direcciones locales de enlace se configuran automáticamente cuando la interfaz se activa para IPv6 durante el inicio o cuando la interfaz se configura manualmente mediante `ipadm`.

Cuando un grupo IPMP tiene conectadas direcciones tanto IPv4 como IPv6 en todas las interfaces del grupo, no es necesario configurar direcciones de IPv4 aparte. El daemon `in.mpathd` puede utilizar las direcciones locales de enlace IPv6 como direcciones de prueba.

Detección de fallos en IPMP

Para garantizar la disponibilidad continua de la red para enviar o recibir tráfico, IPMP realiza una detección de fallos en las interfaces IP subyacentes del grupo IPMP. Las interfaces con fallos siguen sin poder utilizarse hasta que se hayan reparado. Las interfaces activas restantes siguen funcionando mientras que las interfaces en espera se implementan según sea necesario.

El daemon `in.mpathd` controla los siguientes tipos de detección de fallos:

- Dos tipos de detección de fallos basada en sondeos:
  - No se configuran direcciones de prueba (sondeo transitivo).
  - Se configuran las direcciones de prueba.
- Detección de fallos basada en enlaces, si la admite el controlador de la NIC.

Detección de fallos basada en sondeos

La detección de fallos basada en sondeos consiste en utilizar los sondeos ICMP para comprobar si una interfaz ha fallado. La implementación de este método de detección de fallos depende de si las direcciones de prueba se utilizan o no.

Detección de fallos basada en sondeos utilizando direcciones de prueba

Este método de detección de fallos implica enviar y recibir mensajes de sondeo de ICMP que utilizan direcciones de prueba. Estos mensajes, también denominados *tráfico de sondeo* o *tráfico de prueba*, pasan por la interfaz y se dirigen a uno o más sistemas de destino de la misma red local. El daemon `in.mpathd` sondea todos los destinos por separado mediante todas las interfaces que se han configurado para la detección de fallos basada en sondeos. Si no hay ninguna respuesta a cinco sondeos consecutivos en una interfaz específica, el daemon `in.mpathd` considera que la interfaz ha fallado. La velocidad de sondeo depende del *tiempo de detección de fallos* (*FDT, Failure Detection Time*). El valor predeterminado del tiempo de detección de fallos es de 10 segundos. Sin embargo, puede ajustar el FDT en el archivo de configuración IPMP. Para instrucciones, consulte Cómo configurar el comportamiento del daemon IPMP [87].

Para optimizar la detección de fallos basada en sondeos, debe establecer varios sistemas de destino para recibir los sondeos del daemon `in.mpathd`. Teniendo múltiples sistemas de destino, puede determinar mejor la naturaleza de un fallo informado. Por ejemplo, la ausencia de una respuesta del único sistema de destino definido puede indicar un fallo en el sistema de destino o en una de las interfaces del grupo IPMP. Por el contrario, si sólo un sistema entre varios sistemas de destino no responde a un sondeo, es probable que el fallo se encuentre en el sistema de destino en lugar de en el grupo IPMP en sí.

El daemon `in.mpathd` determina qué sistemas de destino se deben sondear dinámicamente. En primer lugar, el daemon busca la tabla de enrutamiento para sistemas de destino en la misma subred de las direcciones de prueba que están asociadas a las interfaces del grupo IPMP. Si estos destinos se encuentran, el daemon los utiliza como destinos para el sondeo. Si no se

encuentran sistemas de destino en la misma subred, el daemon envía paquetes de multidifusión para sondear hosts vecinos en el enlace. Un paquete multidifusión se envía a la dirección de multidifusión de todos los hosts, `224.0.0.1` en IPv4 y `ff02::1` en IPv6, para determinar qué hosts se utilizarán como sistemas de destino. Los primeros hosts que responden a los paquetes de eco se eligen como destinos para los sondeos. Si el daemon no encuentra enrutadores ni hosts que respondan a los sondeos de multidifusión, el daemon no puede detectar los fallos basados en sondeos. En este caso, el comando `ipmpstat -i` informará el estado de sondeo como `unknown`.

Puede utilizar las rutas host para configurar explícitamente una lista de sistemas de destino para utilizar con el comando `in.mpathd`. Para obtener instrucciones, consulte “Configuración de detección de fallos basada en sondeos” [84].

Detección de fallos basada en sondeos sin utilizar direcciones de prueba

Sin direcciones de prueba, este método se implementa con dos tipos de sondeos:

- **Sondeos ICMP**

  Las interfaces activas del grupo IPMP envían los sondeos ICMP para sondear destinos definidos en la tabla de enrutamiento. Una interfaz *activa* es la interfaz subyacente que puede recibir los paquetes IP entrantes dirigidos a la dirección de capa de enlace (L2) de la interfaz. El sondeo ICMP utiliza la dirección de datos como dirección de origen del sondeo. Si el sondeo ICMP alcanza su destino y obtiene una respuesta del mismo, la interfaz activa está en funcionamiento.
- **Sondeos transitivos**

  Las interfaces alternativas del grupo IPMP envían sondeos transitivos para sondear la interfaz activa. Una interfaz alternativa es una interfaz que no recibe activamente paquetes IP entrantes.

  Por ejemplo, considere un grupo IPMP que consta de cuatro interfaces subyacentes y una dirección de datos. En esta configuración, los paquetes salientes pueden utilizar todas las interfaces subyacentes. Sin embargo, los paquetes entrantes sólo se pueden recibir mediante la interfaz a la que la dirección de datos está vinculada. Las tres interfaces subyacentes restantes que no pueden recibir paquetes entrantes son las *interfaces alternativas*.

  Si una interfaz alternativa puede enviar correctamente un sondeo para una interfaz activa y recibir una respuesta, la interfaz activa está en funcionamiento y, por ende, también lo está la interfaz alternativa que ha enviado el sondeo.

Nota - En Oracle Solaris, la detección de fallos basada en sondeos funciona con las direcciones de prueba. Para seleccionar la detección de fallos basada en sondeos sin direcciones de prueba, debe activar manualmente el sondeo transitivo. Para obtener instrucciones, consulte “Selección de un método de detección de fallos” [85].

Fallo de grupo

Los *fallos de grupo* se producen cuando todas las interfaces de un grupo IPMP fallan al mismo tiempo. En este caso, ninguna interfaz es utilizable. Además, cuando todos los sistemas de destino fallan al mismo tiempo y la detección de fallos basada en sondeos está activada, el daemon `in.mpathd` vacía todos sus sistemas de destino y sondeos actuales para los nuevos sistemas de destino.

En un grupo IPMP que no tiene direcciones de prueba, se designa una sola interfaz que pueda sondear la interfaz activa como encargado de los *sondeos*. Esta interfaz designada tiene establecido el indicador `FAILED` y también el indicador `PROBER`. La dirección de datos está vinculada con esta interfaz, lo cual permite a la interfaz continuar con el sondeo del destino para detectar la recuperación.

Detección de fallos basada en enlaces

La detección de fallos basada en enlaces siempre está activada, cuando la interfaz admite este tipo de detección de fallos.

Para determinar si la interfaz de otro proveedor admite la detección de fallos basada en enlaces, utilice el comando `ipmpstat -i`. Si la salida de una interfaz dada incluye un estado `unknown` para su columna `LINK`, dicha interfaz no admite detección de fallos basada en enlaces. Consulte la documentación del fabricante para obtener más información específica sobre el dispositivo.

Los controladores de red que admiten la detección de fallos basada en enlaces supervisan el estado de enlace de la interfaz y notifican al subsistema de red si dicho estado cambia. Cuando se notifica un cambio, el subsistema de red establece o borra el indicador `RUNNING` para dicha interfaz, según sea preciso. Si el daemon `in.mpathd` detecta que el indicador `RUNNING` de la interfaz se ha borrado, el daemon hace fallar inmediatamente a la interfaz.

Detección de fallos y función del grupo anónimo

IPMP admite la detección de fallos en un grupo anónimo. De manera predeterminada, IPMP supervisa el estado sólo de interfaces que pertenecen a grupos IPMP. Sin embargo, el daemon IPMP se puede configurar también para realizar el seguimiento del estado de las interfaces que no pertenecen a ningún grupo IPMP. Por lo tanto, estas interfaces se consideran parte de un *grupo anónimo*. Al emitir el comando `ipmpstat -g`, el grupo anónimo se muestra como guiones dobles (`--`). En los grupos anónimos, las interfaces tendrían direcciones de datos que también funcionan como direcciones de prueba. Debido a que estas interfaces no pertenecen a un grupo IPMP denominado, estas direcciones son visibles a las aplicaciones. Para activar el

seguimiento de interfaces que no forman parte de un grupo IPMP, consulte Cómo configurar el comportamiento del daemon IPMP [87].

Detección de reparaciones de interfaces físicas

El *tiempo de detección de reparaciones* es el doble del tiempo de detección de fallos. El tiempo predeterminado para la detección de fallos es de 10 segundos. En consecuencia, el tiempo predeterminado de detección de reparaciones es de 20 segundos. Después de que una interfaz con fallos se ha marcado con el indicador `RUNNING` de nuevo y el método de detección de fallos ha detectado que la interfaz se ha reparado, el daemon `in.mpathd` borra el indicador `FAILED` de la interfaz. La interfaz reparada se vuelve a implementar en función del número de interfaces activas que el administrador haya definido originalmente.

Cuando una interfaz subyacente falla y se utiliza la detección de fallos basada en sondeos, el daemon `in.mpathd` sigue sondeando, ya sea mediante un encargado de sondeos cuando no se configuró ninguna dirección de prueba o mediante direcciones de prueba de la interfaz.

Durante una reparación de interfaz, cómo continúa el modo de recuperación depende de cómo la interfaz con fallos se configuró originalmente, de la siguiente manera:

- Si la interfaz con fallos era originalmente una interfaz activa, la interfaz reparada vuelve a su estado activo original. La interfaz en espera que funcionaba como un reemplazo durante el fallo se vuelve a cambiar al estado en espera si hay suficientes interfaces que están activas para el grupo IPMP según las definiciones del administrador del sistema.

Nota - Una excepción se da cuando la interfaz activa reparada también está configurada con el modo `FAILBACK=no`. Para obtener más información, consulte “Modo `FAILBACK=no`” [66].

- Si la interfaz con fallos era originalmente una interfaz en espera, la interfaz reparada vuelve a su estado en espera original, siempre que el grupo IPMP refleje el número original de interfaces activas. De lo contrario, la interfaz en espera se convierte en una interfaz activa.

Para una presentación gráfica de cómo funciona IPMP durante el fallo y la reparación de la interfaz, consulte “Cómo funciona IPMP” [55].

Modo `FAILBACK=no`

De manera predeterminada, las interfaces activas que han presentado fallos y se han reparado vuelven automáticamente a convertirse en interfaces activas en el grupo IPMP. Este comportamiento es controlado por el valor del parámetro `FAILBACK` en el archivo de configuración del daemon `in.mpathd`. Sin embargo, es posible que incluso una interrupción

insignificante que se genera a medida que las direcciones de datos son reasignadas a interfaces reparadas no sea aceptable. En ese caso, puede que prefiera permitir que una interfaz en espera activada continúe como una interfaz activa. IPMP permite reemplazar el comportamiento predeterminado para evitar que una interfaz se active automáticamente después de su reparación. Estas interfaces deben estar configuradas en el modo `FAILBACK=no`. Para conocer procedimientos relacionados, consulte Cómo configurar el comportamiento del daemon IPMP [87].

Cuando una interfaz activa en el modo `FAILBACK=no` falla y, consecuentemente, se repara, el daemon `in.mpathd` restaura la configuración IPMP de la siguiente manera:

- El daemon conserva el estado `INACTIVE` de la interfaz, siempre que el grupo IPMP refleje la configuración original de interfaces activas.
- Si la configuración IPMP en el momento de la reparación no refleja la configuración original del grupo de interfaces activas, la interfaz reparada se vuelve a desplegar como una interfaz activa, a pesar del estado `FAILBACK=no`.

Nota - El modo `FAILBACK=NO` está configurado para todo el grupo IPMP, en lugar de como un parámetro ajustable por interfaz.

IPMP y reconfiguración dinámica

La función de reconfiguración dinámica (DR, Dynamic Reconfiguration) de Oracle Solaris permite volver a configurar el hardware del sistema, como las interfaces, mientras el sistema está en ejecución. DR sólo se puede utilizar en los sistemas que admiten esta función. En sistemas que admiten la DR, IPMP se integra en la estructura del gestor de coordinación de reconfiguración (RCM, Reconfiguration Coordination Manager). Por lo tanto, puede conectar, desconectar o volver a conectar de manera segura las NIC y RCM gestiona la reconfiguración dinámica de los componentes del sistema. Por ejemplo, puede conectar, asociar y agregar nuevas interfaces a grupos IPMP existentes. Una vez que estas interfaces están configuradas, están inmediatamente disponibles para que IPMP las utilice.

Primero se revisan todas las solicitudes de desconexión de NIC a fin de garantizar que se preserve la conectividad. Por ejemplo, de manera predeterminada, no puede desconectar una NIC que no se encuentre en un grupo IPMP. Tampoco puede desconectar una NIC que contenga las únicas interfaces en funcionamiento de un grupo IPMP. Sin embargo, si debe eliminar el componente del sistema, puede modificar este comportamiento con la opción `-f` del comando `cfgadm`, tal como se explica en la página del comando man `cfgadm`(1M).

Si las comprobaciones son correctas, el daemon `in.mpathd` establece el indicador `OFFLINE` para la interfaz. Todas las direcciones de prueba de las interfaces se desconfiguran. A continuación, la NIC se desconecta del sistema.

Si falla alguno de estos pasos, o falla la DR de otro hardware del mismo componente del sistema, solo se restablece la configuración persistente. En este caso, se registra el siguiente mensaje de error:

```
"IP: persistent configuration is restored for <ifname>"
```

De lo contrario, la solicitud de desconexión se completará correctamente. Puede eliminar el componente del sistema, y las conexiones existentes no se interrumpen.

Nota - Al reemplazar las NIC, asegúrese de que las tarjetas de remplazo sean del mismo tipo, como Ethernet. Después de que la NIC se reemplaza, las configuraciones de la interfaz IP persistente se aplican a dicha NIC.

♦ ♦ ♦ CAPÍTULO 3

Administración de IPMP

En este capítulo se describe cómo administrar grupos de interfaces con IPMP en la versión de Oracle Solaris. Las tareas que se encuentran en este capítulo describen cómo configurar IPMP mediante el comando `ipadm`, que reemplaza el comando `ifconfig` que se utiliza para configurar IPMP en Oracle Solaris 10. Para obtener más información sobre cómo se asignan entre sí estos dos comandos, consulte “Comparación del comando ifconfig con el comando ipadm” de “Transición de Oracle Solaris 10 a Oracle Solaris 11.2 ”.

Para obtener una explicación detallada de los cambios en el modelo conceptual de IPMP, consulte “Novedades de IPMP” [49].

Este capítulo se divide en los siguientes apartados:

Configuración de grupos IPMP

La siguiente información describe cómo planificar y configurar los grupos IPMP. La descripción general de Capítulo 2, Acerca de la administración de IPMP explica la implementación de un grupo IPMP como interfaz. En este capítulo, los términos *grupo IPMP* e *interfaz IPMP* se utilizan indistintamente.

Esta sección contiene las siguientes tareas:

▼ Cómo planificar un grupo IPMP

El siguiente procedimiento incluye las tareas de planificación requeridas y la información que se debe obtener antes de configurar un grupo IPMP. No es necesario que realice estas tareas en orden secuencial.

Su configuración de IPMP depende de los requisitos de la red para manejar el tipo de tráfico que se aloja en el sistema. IPMP reparte paquetes de red salientes entre las interfaces del grupo IPMP y, por lo tanto, mejora el rendimiento de la red. Sin embargo, para una determinada conexión TCP, el tráfico entrante normalmente sólo sigue una ruta física a fin de minimizar el riesgo de procesar paquetes fuera de servicio.

Por lo tanto, si la red maneja un gran volumen de tráfico saliente, la configuración de un gran número de interfaces en un grupo IPMP puede mejorar el rendimiento de la red. Si en su lugar, el sistema aloja mucho tráfico entrante, tener un gran número de interfaces en el grupo no necesariamente mejora el rendimiento al repartir la carga del tráfico. Sin embargo, tener más interfaces subyacentes ayuda a garantizar la disponibilidad de la red durante fallos de la interfaz.

Nota - Debe configurar sólo un grupo IPMP para cada subred o dominio de emisión L2. Para obtener más información, consulte “Reglas de uso de IPMP” [52].

---

1. **Determine la configuración general de IPMP que se ajuste a sus necesidades.**

   Consulte la información en el resumen de tareas de este procedimiento para obtener ayuda para determinar qué configuración de IPMP utilizar.

2. **(SPARC solamente) Compruebe que cada interfaz del grupo tenga una dirección MAC exclusiva.**

   Para configurar una dirección MAC exclusiva para cada interfaz en el sistema, consulte “Cómo asegurarse de que la dirección MAC de cada interfaz sea única” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.

3. **Asegúrese de que se inserte y configure el mismo conjunto de módulos STREAMS en todas las interfaces del grupo IPMP.**

   Para obtener instrucciones y la sintaxis del comando para utilizar, consulte “Reglas de uso de IPMP” [52].

4. **Utilice el mismo formato de direcciones IP en todas las interfaces del grupo IPMP.**

   Si una interfaz está configurada para IPv4, debe configurar todas las interfaces del grupo IPMP para IPv4. Del mismo modo, si agrega direcciones IPv6 a una interfaz, debe configurar todas las interfaces del grupo IPMP para que admitan IPv6.

5. **Determine el tipo de detección de fallos que desea implementar.**

Por ejemplo, si desea implementar detección de fallos basada en sondeos, debe configurar las direcciones de prueba de las interfaces subyacentes. Consulte “Detección de fallos en IPMP” [62].

6. **Asegúrese de que todas las interfaces del grupo IPMP estén conectadas a la misma red local.**

   Por ejemplo, puede configurar los conmutadores Ethernet en la misma subred IP en un grupo IPMP. Puede configurar cualquier número de interfaces en un grupo IPMP.

   Nota - También puede configurar un grupo IPMP de interfaz única; por ejemplo, si el sistema tiene una única interfaz física. Consulte “Tipos de configuraciones de interfaces IPMP” [54].

7. **Asegúrese de que el grupo IPMP no contenga interfaces con diferentes tipos de medios de red.**

   Las interfaces que están agrupadas deben tener el mismo tipo de interfaz. Por ejemplo, no puede combinar interfaces Ethernet y Token Ring en un mismo grupo IPMP. Tampoco puede combinar una interfaz de bus Token con las interfaces de modalidad de transferencia asíncrona (ATM) del mismo grupo IPMP.

8. **En el caso de IPMP con interfaces ATM, configure dichas interfaces en modo de emulación de LAN.**

   IPMP no se admite para las interfaces que utilicen la solución clásica de IP sobre tecnología ATM, como se define en RFC 1577 (http://www.rfc-editor.org/rfc/rfc1577.txt) y RFC 2225 (http://www.rfc-editor.org/rfc/rfc2225.txt).

▼ Cómo configurar un grupo IPMP que use DHCP

Puede configurar un grupo IPMP con varias interfaces con interfaces activas/activas o activas/en espera. Consulte “Tipos de configuraciones de interfaces IPMP” [54]. En el siguiente procedimiento, se describen los pasos para configurar un grupo IPMP activo/en espera con el DHCP.

Antes de empezar Antes de realizar el siguiente procedimiento, realice las siguientes acciones:

- Asegúrese de que las interfaces IP que estarán en el eventual grupo IPMP hayan sido configuradas correctamente a través de los enlaces de datos de red del sistema. Para conocer los procedimientos, consulte “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”. Puede crear una interfaz IPMP incluso si no ha creado las interfaces IP subyacentes. Sin embargo, sin crear interfaces IP subyacentes, las configuraciones posteriores de la interfaz IPMP fallarán.
- Además, si está utilizando un sistema basado en SPARC, debe configurar una dirección MAC exclusiva para cada interfaz. Consulte “Cómo asegurarse de que la dirección MAC

de cada interfaz sea única" de "Configuración y administración de componentes de red en Oracle Solaris 11.2 ".

- Por último, si está usando el DHCP, asegúrese de que las interfaces subyacentes cuenten con *concesiones infinitas*. De lo contrario, si falla un grupo IPMP, las direcciones de prueba caducarán, y el daemon `in.mpathd` desactivará la detección de fallos basada en sondeos, y se usará la detección de fallos basada en enlaces. Si la detección de fallos basada en enlaces detecta que la interfaz está funcionando, el daemon puede informar erróneamente que la interfaz ha sido reparada. Para obtener más información sobre la configuración de DHCP, consulte "Uso de DHCP en Oracle Solaris 11.2 ".

1. **Asuma el rol `root`.**

2. **Cree una interfaz IPMP.**

   ```
   # ipadm create-ipmp ipmp-interface
   ```

   Donde *ipmp-interface* especifica el nombre de la interfaz IPMP. Puede asignar cualquier nombre significativo a la interfaz IPMP. Al igual que con cualquier interfaz IP, el nombre está formado por una cadena y un número; por ejemplo `ipmp0`.

3. **Cree las interfaces IP subyacentes si todavía no existen.**

   ```
   # ipadm create-ip under-interface
   ```

   Donde *under-interface* hace referencia a la interfaz IP que agregará al grupo IPMP.

4. **Agregue las interfaces IP subyacentes que contendrán las direcciones de prueba para el grupo IPMP.**

   ```
   # ipadm add-ipmp -i under-interface1 [-i under-interface2 ...] ipmp-interface
   ```

   Puede agregar tantas interfaces IP para el grupo IPMP como interfaces haya disponibles en el sistema.

5. **Especifique que DHCP configure y gestione las direcciones de datos en la interfaz IPMP.**

   ```
   # ipadm create-addr -T dhcp ipmp-interface
   ```

   El paso anterior asocia la dirección proporcionada por el servidor DHCP con un objeto de dirección. El objeto de dirección identifica de manera exclusiva la dirección IP con el formato *interface*/*address-type*; por ejemplo, `ipmp0/v4`. Para obtener más información sobre el objeto de dirección, consulte "Cómo configurar una interfaz IPv4" de "Configuración y administración de componentes de red en Oracle Solaris 11.2 ".

6. **Si utiliza la detección de fallos basada en sondeos con direcciones de prueba, especifique que DHCP gestione las direcciones de prueba de las interfaces subyacentes.**

```
# ipadm create-addr -T dhcp under-interface
```

El objeto de dirección que se crea automáticamente en paso 6 utiliza el formato *under-interface*/*address-type*; por ejemplo, `net0/v4`.

7. **(Opcional) Repita el paso 6 para cada interfaz subyacente del grupo IPMP.**

ejemplo 3-1 Configuración de un grupo IPMP con el DHCP

En el siguiente ejemplo se muestra la configuración de un grupo IPMP con interfaz activa-en espera con el DHCP según la siguiente situación:

- Tres interfaces subyacentes `net0`, `net1` y `net2` están configuradas en un grupo IPMP.
- La interfaz IPMP `ipmp0` tiene el mismo nombre que el grupo IPMP.
- `net2` es la interfaz en espera designada.
- A todas las interfaces subyacentes se les asignan direcciones de prueba.

La interfaz IPMP se crea por primera vez.

```
# ipadm create-ipmp ipmp0
```

Las interfaces IP subyacentes se crean y se agregan a la interfaz IPMP.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2

# ipadm add-ipmp -i net0 -i net1 -i net2 ipmp0
```

Las direcciones IP gestionadas por DHCP se asignan a la interfaz IPMP. Las direcciones IP asignadas a la interfaz IPMP son las direcciones de datos. En este ejemplo, la interfaz IPMP tiene dos direcciones de datos.

```
# ipadm create-addr -T dhcp ipmp0
ipadm: ipmp0/v4
# ipadm create-addr -T dhcp ipmp0
ipadm: ipmp0/v4a
```

Luego, las direcciones IP gestionadas por DHCP se asignan a las interfaces IP subyacentes del grupo IPMP. Las direcciones IP asignadas a las interfaces subyacentes son direcciones de prueba que se utilizan para la detección de fallos basada en sondeos.

```
# ipadm create-addr -T dhcp net0
ipadm: net0/v4
# ipadm create-addr -T dhcp net1
ipadm: net1/v4
# ipadm create-addr -T dhcp net2
ipadm net2/v4
```

Por último, la interfaz `net2` se configura como una interfaz en espera.

```
# ipadm set-ifprop -p standby=on net2
```

▼ Cómo configurar un grupo IPMP activo/activo

En el siguiente procedimiento, se describe cómo configurar manualmente un grupo IPMP activo-activo. En este procedimiento, los pasos 1 a 4 describen cómo configurar un grupo IPMP activo-activo basado en enlaces. En el paso 5, se describe cómo realizar la configuración basada en enlaces, basada en sondeos.

Antes de empezar Asegúrese de que las interfaces IP que estarán en el eventual grupo IPMP estén configuradas correctamente a través de los enlaces de datos de red del sistema. Para obtener instrucciones, consulte “Cómo configurar una interfaz IPv4” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”. Puede crear una interfaz IPMP incluso si no existen interfaces IP subyacentes. Sin embargo, las configuraciones posteriores en la interfaz IPMP fallarán.

Además, si está utilizando un sistema basado en SPARC, configure una dirección MAC exclusiva para cada interfaz. Consulte “Cómo asegurarse de que la dirección MAC de cada interfaz sea única” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.

1. **Asuma el rol `root`.**

2. **Cree una interfaz IPMP.**

   ```
   # ipadm create-ipmp ipmp-interface
   ```

   Donde *ipmp-interface* especifica el nombre de la interfaz IPMP. Puede asignar cualquier nombre significativo a la interfaz IPMP. Al igual que con cualquier interfaz IP, el nombre está formado por una cadena y un número; por ejemplo `ipmp0`.

3. **Agregue las interfaces IP subyacentes al grupo.**

   ```
   # ipadm add-ipmp -i under-interface1 [-i underinterface2 ...] ipmp-interface
   ```

   Donde *under-interface* hace referencia a la interfaz subyacente del grupo IPMP. Puede agregar tantas interfaces IP como haya disponibles en el sistema.

   Nota - En un entorno de doble pila, si se coloca una instancia IPv4 de una interfaz en un grupo específico, automáticamente se coloca la instancia IPv6 en el mismo grupo.

4. **Agregue las direcciones de datos a la interfaz IPMP.**

```
# ipadm create-addr -a address ipmp-interface
```

Donde *address* puede utilizar la notación CIDR.

Nota - Solo se necesita la dirección DNS del nombre del grupo IPMP o la dirección IP.

5. **Si utiliza la detección de fallos basada en sondeos con direcciones de prueba, agregue las direcciones de prueba a las interfaces subyacentes.**

```
# ipadm create-addr -a address under-interface
```

Donde *address* puede utilizar la notación CIDR. Todas las direcciones IP de prueba de un grupo IPMP deben pertenecer a una única subred IP y, por lo tanto, deben usar el mismo prefijo de red.

▼ Cómo configurar un grupo IPMP activo/en espera

En el procedimiento siguiente, se describe cómo configurar un grupo IPMP en el que una interfaz se mantiene como una interfaz en espera. Esta interfaz se despliega sólo cuando la interfaz activa del grupo falla.

Para obtener una descripción general sobre las interfaces en espera, consulte “Tipos de configuraciones de interfaces IPMP” [54].

1. **Asuma el rol `root`.**

2. **Cree una interfaz IPMP.**

```
# ipadm create-ipmp ipmp-interface
```

Donde *ipmp-interface* especifica el nombre de la interfaz IPMP.

3. **Agregue las interfaces IP subyacentes al grupo.**

```
# ipadm add-ipmp -i under-interface1 [-i underinterface2 ...] ipmp-interface
```

Donde *under-interface* hace referencia a la interfaz subyacente del grupo IPMP. Puede agregar tantas interfaces IP como haya disponibles en el sistema.

Nota - En un entorno de doble pila, si se coloca una instancia IPv4 de una interfaz en un grupo IPMP específico, automáticamente se coloca la instancia IPv6 en el mismo grupo.

4. **Agregue las direcciones de datos a la interfaz IPMP.**

```
# ipadm create-addr -a address ipmp-interface
```

Donde *address* puede utilizar la notación CIDR.

5. **Si utiliza la detección de fallos basada en sondeos con direcciones de prueba, agregue las direcciones de prueba a las interfaces subyacentes.**

   # ipadm create-addr -a *address under-interface*

   Donde *address* puede utilizar la notación CIDR. Todas las direcciones IP de prueba de un grupo IPMP deben pertenecer a una única subred IP y, por lo tanto, deben usar el mismo prefijo de red.

6. **Configure una de las interfaces subyacentes como interfaz en espera.**

   # ipadm set-ifprop -p standby=on -m ip *under-interface*

ejemplo 3-2 Configuración de un grupo IPMP de interfaz activa-en espera

El siguiente ejemplo muestra cómo crear una configuración IPMP activa/en espera.

En primer lugar, se crea la interfaz IPMP.

```
# ipadm create-ipmp ipmp0
```

Luego, se crean las interfaces IP subyacentes y se agregan a la interfaz IPMP.

```
# ipadm create-ip net0
# ipadm create-ip net1
# ipadm create-ip net2

# ipadm add-ipmp -i net0 -i net1 -i net2 ipmp0
```

A continuación, las direcciones IP se asignan a la interfaz IPMP. Las direcciones IP asignadas a la interfaz IPMP son las direcciones de datos. En este ejemplo, la interfaz IPMP tiene dos direcciones de datos.

```
# ipadm create-addr -a 192.168.10.10/24 ipmp0
ipadm: ipmp0/v4
# ipadm create-addr -a 192.168.10.15/24 ipmp0
ipadm: ipmp0/v4a
```

La dirección IP de este ejemplo incluye la propiedad `prefixlen`, que se expresa como un número decimal. La parte `prefixlen` de la dirección IP especifica el número de bits contiguos que están más a la izquierda de la dirección que comprende la máscara de red IPv4 o el prefijo de IPv6 de la dirección. El resto de los bits de orden inferior definen la parte del host de la dirección. Cuando la propiedad `prefixlen` se convierte en una representación de texto de la dirección, la dirección contiene 1 para las posiciones de bits que se van a utilizar para la parte de la red y 0 para la parte del host. Esta propiedad no se admite en el tipo de objeto de dirección `dhcp`. Para obtener más información, consulte la página del comando man `ipadm`(1M).

Las direcciones IP se asignan a las interfaces IP subyacentes del grupo IPMP. Las direcciones IP asignadas a las interfaces subyacentes son direcciones de prueba que se utilizan para la detección de fallos basada en sondeos.

```
# ipadm create-addr -a 192.168.10.30/24 net0
ipadm: net0/v4
# ipadm create-addr -a 192.168.10.32/24 net1
ipadm: net1/v4
# ipadm create-addr -a 192.168.10.34/24 net2
ipadm: net2/v4
```

Por último, la interfaz `net2` se configura como una interfaz en espera.

```
# ipadm set-ifprop -p standby=on net2
```

El administrador puede ver la configuración IPMP mediante el comando `ipmpstat .`

```
# ipmpstat -g
GROUP     GROUPNAME   STATE      FDT        INTERFACES
ipmp0     ipmp0       ok         10.00s     net0 net1 (net2)

# ipmpstat -t
INTERFACE  MODE     TESTADDR        TARGETS
net0       routes   192.168.10.30   192.168.10.1
net1       routes   192.168.10.32   192.168.10.1
net2       routes   192.168.10.34   192.168.10.5
```

Mantenimiento de enrutamiento durante la implementación de IPMP

Al configurar un grupo IPMP, la interfaz IPMP hereda las direcciones IP de sus interfaces subyacentes para utilizarlos como direcciones de datos. Las interfaces subyacentes, luego, reciben la dirección IP `0.0.0.0`. Por lo tanto, las rutas que se definen mediante las interfaces IP se pierden si estas interfaces se agregan posteriormente a un grupo IPMP.

La pérdida de enrutamiento al configurar IPMP comúnmente implica a la ruta predeterminada y se produce junto con una instalación de Oracle Solaris. Durante la instalación, se le solicita que defina una ruta predeterminada, para la que se utiliza una interfaz que está en el sistema, como la interfaz principal. Posteriormente, se configura un grupo IPMP con la misma interfaz en la que definió la ruta predeterminada. Después de la configuración de IPMP, el sistema ya no puede enrutar paquetes de red porque la dirección de la interfaz se transfirió a la interfaz de IPMP.

Para asegurarse de que se conserve la ruta predeterminada mientras usa IPMP, la ruta debe estar definida sin especificar la interfaz. De esta forma, cualquier interfaz, incluida la interfaz

de IPMP, se puede utilizar para el enrutamiento. Por lo tanto, el sistema puede continuar para enrutando el tráfico.

Nota - En la siguiente tarea, se utiliza la interfaz principal como ejemplo en el que se define la ruta predeterminada. Sin embargo, este tipo de caso de pérdida de enrutamiento se aplica a cualquier interfaz que se utiliza para enrutamiento que, posteriormente, se convierte en parte de un grupo IPMP.

▼ Cómo mantener la ruta predeterminada mientras se utiliza IPMP

En el siguiente procedimiento se describe cómo mantener la ruta predeterminada al configurar IPMP.

1. **Inicie sesión en el sistema mediante una consola.**

   Debe utilizar la consola para realizar este procedimiento. Si utiliza el comando `ssh` o `telnet` para iniciar la sesión, se perderá la conexión cuando realice los siguientes pasos.

2. **(Opcional) Muestre las rutas que están definidas actualmente en la tabla de enrutamiento.**

   ```
   # netstat -nr
   ```

3. **Suprima la ruta que está vinculada a la interfaz específica.**

   ```
   # route -p delete default gateway-address -ifp interface
   ```

4. **Agregue la ruta sin especificar una interfaz.**

   ```
   # route -p add default gateway-address
   ```

5. **(Opcional) Muestre las rutas redefinidas en la tabla de enrutamiento.**

   ```
   # netstat -nr
   ```

6. **(Opcional) Si la información no cambió, reinicie el servicio de enrutamiento y, a continuación, vuelva a comprobar la información de la tabla de enrutamiento para asegurarse de que las rutas se hayan redefinido en forma correcta.**

   ```
   # svcadm restart routing-setup
   ```

ejemplo 3-3 Definición de rutas para IPMP

En este ejemplo se supone que la ruta predeterminada se definió para `net0` durante la instalación.

```
# netstat -nr
Routing Table: IPv4
Destination     Gateway       Flags    Ref     Use       Interface
-------------  ------------  --------  -----  -----------  --------
default        10.153.125.1     UG      107    176682262   net0
10.153.125.0   10.153.125.222   U        22    137738792   net0

# route -p delete default 10.153.125.1 -ifp net0
# route -p add default 10.153.125.1

# netstat -nr
Routing Table: IPv4
Destination     Gateway       Flags    Ref     Use       Interface
-------------  ------------  --------  -----  -----------  --------
default        10.153.125.1     UG      107    176682262
10.153.125.0   10.153.125.222   U        22    137738792   net0
```

Administración de IPMP

En esta sección, se incluyen los siguientes procedimientos para mantener el grupo IPMP que ha creado en el sistema:

- Cómo agregar una interfaz a un grupo IPMP [79]
- Cómo eliminar una interfaz de un grupo IPMP [80]
- Cómo agregar direcciones IP a un grupo IPMP [80]
- Cómo suprimir direcciones IP de una interfaz IPMP [81]
- Cómo mover una interfaz de un grupo IPMP a otro grupo IPMP [82]
- Cómo suprimir un grupo IPMP [83]

▼ Cómo agregar una interfaz a un grupo IPMP

Antes de empezar Asegúrese de que la interfaz que agregue al grupo cumple todos los requisitos necesarios. Para obtener una lista de requisitos, consulte Cómo planificar un grupo IPMP [70].

1. **Asuma el rol `root`.**

2. **Si la interfaz IP subyacente aún no existe, créela.**

   ```
   # ipadm create-ip under-interface
   ```

3. **Agregue la interfaz IP al grupo IPMP.**

   ```
   # ipadm add-ipmp -i under-interface ipmp-interface
   ```

Donde *ipmp-interface* hace referencia al grupo IPMP al que desea agregar la interfaz subyacente.

ejemplo 3-4 Agregación de una interfaz a un grupo IPMP

En el ejemplo siguiente se muestra cómo agregar la interfaz `net4` al grupo IPMP `ipmp0`.

```
# ipadm create-ip net4
# ipadm add-ipmp -i net4 ipmp0
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT       INTERFACES
ipmp0   ipmp0      ok         10.00s    net0 net1 net4
```

▼ Cómo eliminar una interfaz de un grupo IPMP

1. **Asuma el rol `root`.**
2. **Elimine una o más interfaces del grupo IPMP.**

   `# ipadm remove-ipmp -i` *under-interface*`[ -i` *under-interface* `...]` *ipmp-interface*

   Donde *under-interface* hace referencia a una interfaz IP que se elimina del grupo IPMP e *ipmp-interface* hace referencia al grupo IPMP del que desee quitar interfaces subyacentes.
   Puede eliminar tantas interfaces subyacentes en un comando único según sea necesario. La eliminación de todas las interfaces subyacentes no suprime la interfaz IPMP. En su lugar, existe como interfaz o grupo IPMP vacío.

ejemplo 3-5 Eliminación de una interfaz de un grupo IPMP

En el ejemplo siguiente se muestra cómo eliminar la interfaz `net4` del grupo IPMP `ipmp0`.

```
# ipadm remove-ipmp net4 ipmp0
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT       INTERFACES
ipmp0   ipmp0      ok         10.00s    net0 net1
```

▼ Cómo agregar direcciones IP a un grupo IPMP

Para agregar direcciones IP a un grupo IPMP, utilice el comando `ipadm create-addr`. Para la configuración IPMP, una dirección IP puede ser una dirección de datos o una dirección de prueba. Una dirección de datos se agrega a una interfaz IPMP, mientras que una dirección de prueba se agrega a una interfaz subyacente de la interfaz IPMP. En el siguiente procedimiento, se describe cómo agregar direcciones IP que son direcciones de prueba o direcciones de datos.

1. **Asuma el rol `root`.**

2. **Agregue las direcciones IP a un grupo IPMP.**

   - **Agregue direcciones de datos a un grupo IPMP de la siguiente manera:**

     `# ipadm create-addr -a` *address ipmp-interface*

     Un objeto de dirección se asigna automáticamente a la dirección IP que se acaba de crear. Un objeto de dirección es un identificador único de la dirección IP. El nombre del objeto de dirección utiliza la convención de denominación de *interface*/*random-string*. Por lo tanto, los objetos de dirección de las direcciones de datos incluirán la interfaz IPMP en sus nombres.

   - **Agregue direcciones de prueba a una interfaz subyacente del grupo IPMP de la siguiente manera:**

     `# ipadm create-addr -a` *address under-interface*

     Un objeto de dirección se asigna automáticamente a la dirección IP que se acaba de crear. Un objeto de dirección es un identificador único de la dirección IP. El nombre del objeto de dirección utiliza la convención de denominación de *interface*/*random-string*. Por lo tanto, los objetos de dirección de las direcciones de prueba incluirán la interfaz subyacente en sus nombres.

▼ Cómo suprimir direcciones IP de una interfaz IPMP

Para suprimir direcciones IP de un grupo IPMP, utilice el comando `ipadm delete-addr`. Para la configuración IPMP, las direcciones de datos se alojan en la interfaz IPMP y las direcciones de prueba se alojan en las interfaces subyacentes. En el siguiente procedimiento, se muestra cómo eliminar direcciones IP que sean direcciones de datos o direcciones de prueba.

1. **Asuma el rol `root`.**

2. **Determine qué direcciones IP desea eliminar.**

   - **Muestre una lista de direcciones de datos de la siguiente manera:**

     `# ipadm show-addr` *ipmp-interface*

   - **Muestre una lista de direcciones de prueba de la siguiente manera:**

     `# ipadm show-addr`

Las direcciones de prueba se identifican mediante objetos de dirección cuyos nombres incluyen las interfaces subyacentes donde se configuran las direcciones.

3. **Elimine las direcciones IP de un grupo IPMP.**

   - **Elimine direcciones de datos de la siguiente manera:**

     `# ipadm delete-addr` *addrobj*

     Donde *addrobj* debe incluir el nombre de la interfaz IPMP. Si el objeto de dirección que ha escrito no incluye el nombre de la interfaz IPMP, la dirección que se va a suprimir no es una dirección de datos.

   - **Elimine direcciones de prueba de la siguiente manera:**

     `# ipadm delete-addr` *addrobj*

     Donde *addrobj* debe incluir el nombre de la interfaz subyacente correcta para suprimir la dirección de prueba correcta.

ejemplo 3-6 Eliminación de una dirección de prueba de una interfaz

En el siguiente ejemplo, se usa la configuración del grupo IPMP activo/en espera `ipmp0` que se muestra en Ejemplo 3-2, “Configuración de un grupo IPMP de interfaz activa-en espera”. Este ejemplo elimina la dirección de prueba de la interfaz subyacente `net1`.

```
# ipadm show-addr net1
ADDROBJ           TYPE     STATE     ADDR
net1/v4           static   ok        192.168.10.30

# ipadm delete-addr net1/v4
```

▼ Cómo mover una interfaz de un grupo IPMP a otro grupo IPMP

Puede colocar una interfaz en un grupo IPMP nuevo cuando la interfaz pertenece a un grupo IPMP existente. No es necesario eliminar la interfaz del grupo IPMP actual. Cuando coloca la interfaz en un grupo nuevo, se elimina automáticamente de cualquier grupo IPMP existente.

1. **Asuma el rol `root`.**

2. **Mueva la interfaz a un grupo IPMP nuevo.**

   `# ipadm add-ipmp -i` *under-interface ipmp-interface*

Donde *under-interface* hace referencia a la interfaz subyacente que desea mover y *ipmp-interface* hace referencia a la interfaz IPMP a la que desea mover la interfaz subyacente.

ejemplo 3-7 Cómo mover una interfaz a otro grupo IPMP

En el siguiente ejemplo, las interfaces subyacentes del grupo IPMP son `net0` , `net1` y `net2`. Este ejemplo muestra cómo eliminar la interfaz `net0` del grupo IPMP `ipmp0` y, luego, coloca `net0` del grupo IPMP `cs-link1`.

```
# ipadm add-ipmp -i net0 ca-link1
```

▼ Cómo suprimir un grupo IPMP

Utilice el siguiente procedimiento si ya no necesita un grupo IPMP específico.

1. **Asuma el rol `root`.**
2. **Identifique el grupo IPMP y las interfaces IP subyacentes que se van a suprimir.**

   ```
   # ipmpstat -g
   ```

3. **Elimine todas las interfaces IP que pertenecen actualmente al grupo IPMP.**

   `# ipadm remove-ipmp -i` *under-interface*`[, -i` *under-interface*`, ...]` *ipmp-interface*

   Donde *under-interface* hace referencia a la interfaz subyacente que desea suprimir y *ipmp-interface* hace referencia a la interfaz IPMP de la que desea suprimir la interfaz subyacente.

   Nota - Para suprimir correctamente una interfaz IPMP, no debe existir ninguna interfaz IP como parte del grupo IPMP.

4. **Suprima la interfaz IPMP.**

   `# ipadm delete-ipmp` *ipmp-interface*

   Después de suprimir la interfaz IPMP, cualquier dirección IP que esté asociada con la interfaz también se suprime del sistema.

ejemplo 3-8 Supresión de una interfaz IPMP

En el siguiente ejemplo, se suprime la interfaz `ipmp0` con las interfaces IP subyacentes `net0` y `net1`.

```
# ipmpstat -g
GROUP   GROUPNAME   STATE      FDT        INTERFACES
```

```
ipmp0   ipmp0      ok         10.00s     net0 net1

# ipadm remove-ipmp -i net0 -i net1 ipmp0

# ipadm delete-ipmp ipmp0
```

Configuración de detección de fallos basada en sondeos

Esta sección incluye los siguientes temas:

Acerca de la detección de fallos basada en sondeos

La detección de fallos basada en sondeos implica el uso de sistemas de destino, tal como se describe en "Detección de fallos basada en sondeos" [63]. Para identificar destinos para la detección de fallos basada en sondeos, el daemon `in.mpathd` funciona en dos modos: *modo de destino de enrutador* o *modo de destino de multidifusión*. En el modo de destino de enrutador, el daemon sondea los destinos definidos en la tabla de enrutamiento. Si no se han definido destinos, el daemon funciona en modo de destino de multidifusión, en el que se envían paquetes de multidifusión para sondear los hosts vecinos en la LAN.

Preferiblemente, debe configurar los sistemas de destino para que el daemon `in.mpathd` realice el sondeo. Para algunos grupos IPMP, el enrutador predeterminado es suficiente como destino. Sin embargo, para algunos grupos IPMP, quizá desee configurar destinos específicos para la detección de fallos basada en sondeos. Para especificar los destinos, configure las rutas de host en la tabla de enrutamiento como destinos de sondeo. Cualquier ruta host configurada en la tabla de enrutamiento aparece enumerada antes del enrutador predeterminado. IPMP utiliza rutas host definidas explícitamente para la selección de destino. Por lo tanto, debe configurar rutas host para configurar destinos específicos de sondeo en lugar de utilizar el enrutador predeterminado.

Para configurar las rutas host en la tabla de enrutamiento, utilice el comando `route`. Puede utilizar la opción `-p` con este comando para agregar rutas persistentes. Por ejemplo, `route -p add` agrega una ruta que permanecerá en la tabla de enrutamiento incluso después de reiniciar el sistema. Por lo tanto, la opción `-p` permite agregar rutas persistentes sin necesidad de secuencias

de comandos especiales para volver a crear estas rutas con cada inicio del sistema. Para utilizar de forma óptima la detección de fallos basada en sondeos, asegúrese de configurar varios destinos para recibir sondeos.

El comando `route` funciona en rutas IPv4 e IPv6; el valor predeterminado son las rutas IPv4. Si la opción `-inet6` se utiliza inmediatamente después del comando `route`, las operaciones se llevan a cabo en rutas IPv6.

El procedimiento descrito en Cómo especificar manualmente los sistemas de destino para la detección de fallos basada en sondeos [86] muestra la sintaxis exacta para usar para agregar rutas persistentes a los objetivos para la detección de fallos basada en sondeos. Para obtener más información sobre las opciones que se pueden utilizar con el comando `route`, consulte la página del comando man `route`(1M) y “Mantenimiento de enrutamiento durante la implementación de IPMP” [77].

Requisitos para seleccionar destinos para la detección de fallos basada en sondeos

Consulte los siguientes requisitos para determinar qué hosts de su red podrían servir como buenos destinos:

- Asegúrese de que los posibles destinos estén disponibles y de que se estén ejecutando. Haga una lista de sus direcciones IP.
- Asegúrese de que las interfaces de destino se encuentren en la misma red que el grupo IPMP que está configurando.
- La máscara de red y las direcciones de difusión de los sistemas de destino deben ser las mismas que las direcciones del grupo IPMP.
- El host de destino debe poder responder a las solicitudes de ICMP desde la interfaz que utiliza la detección de fallos basada en sondeos.

Selección de un método de detección de fallos

De manera predeterminada, la detección de fallos basada en sondeos funciona mediante direcciones de prueba. Si el controlador NIC es compatible con ella, la detección de fallos basada en enlaces también se activa automáticamente.

No puede desactivar la detección de fallos basada en enlaces si este método es compatible con el controlador NIC. Sin embargo, puede seleccionar qué tipo de detección de fallos basada en sondeos implementar.

Antes de seleccionar un método de detección basada en sondeos, asegúrese de que los destinos de sondeo cumplan los requisitos que se enumeran en “Requisitos para seleccionar destinos para la detección de fallos basada en sondeos” [85].

Para utilizar únicamente sondeo transitivo, realice lo siguiente:

1. Active la propiedad IPMP `transitive-probing` con los comandos SMF.

   ```
   # svccfg -s svc:/network/ipmp setprop config/transitive-probing=true
   # svcadm refresh  svc:/network/ipmp:default
   ```

   Para obtener más información sobre la configuración de esta propiedad, consulte la página del comando man `in.mpathd`(1M).
2. Elimine cualquier dirección de prueba existente que se haya configurado para el grupo IPMP.

   ```
   # ipadm delete-addr address  addrobj
   ```

   Donde *addrobj* debe hacer referencia a un interfaz subyacente que aloja una dirección de prueba.

Para utilizar únicamente direcciones de prueba para el sondeo de fallos, realice lo siguiente:

1. Si es necesario, desactive el sondeo transitivo mediante comandos SMF.

   ```
   # svccfg -s svc:/network/ipmp setprop config/transitive-probing=false
   # svcadm refresh  svc:/network/ipmp:default
   ```
2. Asigne direcciones de prueba para las interfaces subyacentes del grupo IPMP.

   ```
   # ipadm create-addr -a address  under-interface
   ```

   Donde *address* puede estar en notación CIDR y *under-interface* es una interfaz subyacente del grupo IPMP.

▼ Cómo especificar manualmente los sistemas de destino para la detección de fallos basada en sondeos

En el siguiente procedimiento se describe cómo agregar destinos específicos si utiliza las direcciones de prueba para implementar la detección de fallos basada en sondeos.

Antes de empezar Asegúrese de que los destinos de sondeo cumplan los requisitos descritos en “Requisitos para seleccionar destinos para la detección de fallos basada en sondeos” [85].

1. **Inicie sesión con su cuenta de usuario en el sistema en el que va a configurar la detección de fallos basada en sondeos.**
2. **Agregue una ruta al host particular que se utilizará como destino en la detección de fallos basada en sondeos.**

```
% route -p add -host destination-IP gateway-IP -static
```

Donde *destination-IP* y *gateway-IP* son direcciones IPv4 del host que se utilizará como un destino. Por ejemplo, para especificar el sistema de destino escribiría `192.168.10.137`, que se encuentra en la misma subred que las interfaces del grupo IPMP `ipmp0`:

```
% route -p add -host 192.168.10.137 192.168.10.137 -static
```

Esta nueva ruta se configurará automáticamente cada vez que se reinicie el sistema. Si desea definir sólo una ruta temporal para un sistema de destino para la detección de fallos basada en sondeos, no utilice la opción `-p`.

3. **Agregue rutas a los host adicionales de la red para utilizar como sistemas de destino.**

▼ Cómo configurar el comportamiento del daemon IPMP

Utilice el archivo de configuración IPMP `/etc/default/mpathd` para configurar los siguientes parámetros del sistema para los grupos IPMP:

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

1. **Asuma el rol `root`.**

2. **Edite el archivo `/etc/default/mpathd`.**

```
# pfedit /etc/default/mpathd
```

Consulte la página del comando man `pfedit`(1M) para obtener instrucciones.

Cambie el valor predeterminado de uno o más de los tres parámetros siguientes:

- Escriba el nuevo valor para el parámetro `FAILURE_DETECTION_TIME` de la siguiente manera:

  ```
  FAILURE_DETECTION_TIME=n
  ```

  Donde *n* es el tiempo en segundos para que los sondeos ICMP detecten si se ha producido un fallo de la interfaz. El valor predeterminado es de 10 segundos.
- Escriba el nuevo valor para el parámetro `FAILBACK` de la siguiente manera:

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

`yes` — Es el comportamiento predeterminado de failback de IPMP. Cuando se detecta la reparación de una interfaz fallida, el acceso de red recupera la interfaz reparada, tal como se describe en “Detección de reparaciones de interfaces físicas” [66].

`no` — Indica que el tráfico de datos no regresa a una interfaz reparada. Cuando se detecta la reparación de una interfaz fallida, se configura el indicador `INACTIVE` para dicha interfaz. Este indicador significa que la interfaz no se va a utilizar para el tráfico de datos. La interfaz se puede seguir utilizando para el tráfico de sondeos.

Por ejemplo, suponga que el grupo IPMP `ipmp0` consta de dos interfaces: `net0` y `net1`. En el archivo `/etc/default/mpathd`, está definido el parámetro `FAILBACK=no`. Si `net0` falla, se marca como `FAILED` y se vuelve no utilizable. Tras la reparación, la interfaz se marca como `INACTIVE` y permanece no utilizable por el valor `FAILBACK=no`.

Si `net1` falla y solamente `net0` está en el estado `INACTIVE`, se borra el indicador `INACTIVE` de `net0`, y la interfaz se vuelve utilizable. Si el grupo de IPMP tiene otras interfaces que también están en el estado `INACTIVE`, cualquiera de estas interfaces `INACTIVE`, y no necesariamente `net0`, se pueden borrar y convertir en utilizables cuando `net1`.

- Escriba el nuevo valor para el parámetro `TRACK_INTERFACES_ONLY_WITH_GROUPS` de la siguiente manera:

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

`yes` — Es el comportamiento predeterminado de IPMP. Este valor hace que IPMP omita las interfaces de red que no estén configuradas en un grupo IPMP.

`no` — Define la detección de fallos y la reparación para *todas* las interfaces de red, independientemente de si están configuradas en un grupo IPMP. Sin embargo, cuando se detecta un fallo o reparación en una interfaz que no está configurada en un grupo IPMP, no se desencadena ninguna acción en IPMP para mantener la funciones de red de dicha interfaz. Por tanto, el valor `no` sólo resulta útil para comunicar fallos y no mejora directamente la disponibilidad de la red.

Para obtener más información sobre este parámetro y la función de grupo anónimo, consulte “Detección de fallos y función del grupo anónimo” [65].

3. **Reinicie el daemon `in.mpathd`.**

```
# pkill -HUP in.mpathd
```

El daemon se reinicia con los nuevos valores de los parámetros en vigor.

Supervisión de información de IPMP

Los siguientes ejemplos muestran cómo utilizar el comando `ipmpstat` para supervisar diferentes aspectos de los grupos IPMP que están en el sistema. Puede observar el estado de un grupo IPMP como un todo o puede observar sus interfaces IP subyacentes. También puede verificar la configuración de los datos y las direcciones de prueba para un grupo IPMP. También puede utilizar el mismo comando para obtener información sobre la detección de fallos. Para obtener más información, consulte la página del comando man `ipmpstat`(1M).

Cuando utiliza el comando `ipmpstat`, de manera predeterminada, se muestran los campos más significativos que entran en 80 columnas. En la salida, se muestran todos los campos que son específicos para la opción que se utilizan con el comando `ipmpstat`, excepto en el caso en el que se utiliza `ipmpstat` con la opción `-p`.

De manera predeterminada, los nombres de host se muestran en la salida en lugar de en las direcciones IP numéricas, siempre que haya nombres de host. Para listar las direcciones IP numéricas en la salida, utilice la opción `-n` junto con otras opciones para mostrar información específica del grupo IPMP.

Nota - En los siguientes procedimientos, el uso del comando `ipmpstat` no requiere privilegios de administrador del sistema, a menos que se especifique lo contrario.

Utilice el comando `ipmpstat` con las opciones siguientes para mostrar la información deseada:

`-g` Muestra información sobre los grupos IPMP en el sistema. Consulte Ejemplo 3-9, “Obtención de información del grupo IPMP”.

`-a` Permite visualizar las direcciones de datos que se configuran para los grupos IPMP. Consulte Ejemplo 3-10, “Obtención de información de dirección de datos IPMP”.

`-i` Muestra información acerca de interfaces IP que están relacionadas con la configuración IPMP. Consulte Ejemplo 3-11, “Obtención de información sobre interfaces IP subyacentes de un grupo IPMP”.

`-t` Muestra información sobre los sistemas de destino que se utilizan para detectar los fallos. Esta opción también muestra las direcciones de prueba

que se utilizan en el grupo IPMP. Consulte Ejemplo 3-12, “Obtención de información de destino de sondeo IPMP”.

`-p` Muestra información sobre los sondeos utilizados para la detección de fallos. Consulte Ejemplo 3-13, “Observación de sondeos IPMP”.

Los siguientes ejemplos muestran cómo visualizar información adicional sobre la configuración IPMP del sistema mediante el comando `ipmpstat`.

EJEMPLO 3-9 Obtención de información del grupo IPMP

La opción `-g` muestra el estado de los diferentes grupos IPMP en el sistema, incluidos los estados de sus interfaces subyacentes. Si está activada la detección de fallos basada en sondeos para un grupo específico, el comando también incluye el tiempo de detección de fallos para ese grupo.

```
% ipmpstat -g
GROUP   GROUPNAME   STATE      FDT        INTERFACES
ipmp0   ipmp0       ok         10.00s     net0 net1
acctg1  acctg1      failed     --         [net3 net4]
field2  field2      degraded   20.00s     net2 net5 (net7) [net6]
```

Los campos de salida proporcionan la siguiente información:

`GROUP` Especifica el nombre de la interfaz IPMP. Para un grupo anónimo, este campo aparece vacío. Para obtener más información sobre los grupos anónimos, consulte la página del comando man `in.mpathd`(1M).

`GROUPNAME` Especifica el nombre del grupo IPMP. En el caso de un grupo anónimo, este campo está vacío.

`ESTADO` Indica el estado actual de un grupo IPMP, que puede ser uno de los siguientes:

- `ok`: indica que pueden usarse todas las interfaces subyacentes del grupo IPMP.
- `degraded`: indica que algunas de las interfaces subyacentes del grupo son no utilizables.
- `failed`: indica que todas las interfaces del grupo son no utilizables.

`FDT` Especifica el tiempo de detección de fallos, si está activada la detección de fallos. Si la detección de fallos está desactivada, este campo estará vacío.

`INTERFACES` Especifica las interfaces subyacentes que pertenecen al grupo IPMP. En este campo, primero se muestran las interfaces activas, luego las

interfaces inactivas y, por último, las interfaces no utilizables. La manera en que se muestra una interfaz indica su estado:

- *interface* (sin paréntesis ni corchetes): indica una interfaz activa. Las interfaces activas son las que están siendo usadas por el sistema para enviar o recibir tráfico de datos.
- *(interface)* (con paréntesis): indica una interfaz inactiva, pero en funcionamiento. La interfaz no está en uso, según como lo define la política de administración.
- *[interface]* (con corchetes): indica que la interfaz es no utilizable porque ha fallado o se ha puesto fuera de línea.

EJEMPLO 3-10 Obtención de información de dirección de datos IPMP

La opción `-a` muestra las direcciones de datos y el grupo IPMP al que pertenece cada dirección. La información que se muestra también incluye qué direcciones están disponibles para su uso, en función de si las direcciones han sido activadas y desactivadas mediante el comando `ipadm [up-addr/down-addr]`. También puede determinar en qué interfaz entrante o saliente se puede utilizar una dirección.

```
% ipmpstat -an
ADDRESS         STATE    GROUP      INBOUND     OUTBOUND
192.168.10.10   up       ipmp0         net0    net0 net1
192.168.10.15   up       ipmp0         net1    net0 net1
192.0.0.100     up       acctg1        --          --
192.0.0.101     up       acctg1        --          --
192.168.10.31   up       field2        net2    net2 net7
192.168.10.32   up       field2        net7    net2 net7
192.168.10.33   down     field2        --          --
```

Los campos de salida proporcionan la siguiente información:

`ADDRESS` Especifica el nombre del host o la dirección de datos, si se utiliza la opción `-n` junto con la opción `-a`.

`ESTADO` Indica si la dirección de la interfaz IPMP es `up` y, por lo tanto, es utilizable, o `down` y, por lo tanto, es no utilizable.

`GROUP` Especifica la interfaz IPMP que contiene una dirección de datos específica. En Oracle Solaris, por lo general, el nombre del grupo IPMP es la interfaz IPMP.

`INBOUND` Identifica la interfaz que recibe los paquetes para una dirección determinada. La información de campo puede cambiar en función de eventos externos. Por ejemplo, si una dirección de datos está caída o si no quedan interfaces IP activas en el grupo IPMP, este campo aparecerá vacío. El campo vacío indica que el sistema no acepta paquetes IP que están destinados a la dirección indicada.

`OUTBOUND` Identifica la interfaz que envía los paquetes que utilizan una dirección dada como dirección de origen. Al igual que con el campo `INBOUND`, la información del campo `OUTBOUND` también puede cambiar en función de eventos externos. Un campo vacío indica que el sistema no envía paquetes con la dirección de origen dada. El campo podría estar vacío, ya sea porque la dirección está caída o porque no quedan interfaces IP activas en el grupo.

EJEMPLO 3-11 Obtención de información sobre interfaces IP subyacentes de un grupo IPMP

La opción `-i` muestra información sobre un grupo IPMP de interfaces IP subyacentes.

```
% ipmpstat -i
INTERFACE   ACTIVE   GROUP      FLAGS      LINK       PROBE      STATE
net0        yes      ipmp0      --mb---    up         ok         ok
net1        yes      ipmp0      -------    up         disabled   ok
net3        no       acctg1     -------    unknown    disabled   offline
net4        no       acctg1     is-----    down       unknown    failed
net2        yes      field2     --mb---    unknown    ok         ok
net6        no       field2     -i-----    up         ok         ok
net5        no       filed2     -------    up         failed     failed
net7        yes      field2     --mb---    up         ok         ok
```

Los campos de salida proporcionan la siguiente información:

`INTERFACE` Especifica cada interfaz subyacente de cada grupo IPMP.

`ACTIVE` Indica si la interfaz está en funcionamiento y en uso (`yes`) o no (`no`).

`GROUP` Especifica el nombre de la interfaz IPMP. Para grupos anónimos, este campo aparece vacío. Para obtener más información sobre los grupos anónimos, consulte la página del comando man `in.mpathd`(1M).

`FLAGS` Indica el estado de cada interfaz subyacente, que puede ser uno de los siguientes o cualquier combinación de ellos:

- `b`: indica que la interfaz fue designada por el sistema para recibir el tráfico de difusión para el grupo IPMP.
- `d`: indica que la interfaz está caída y que, por lo tanto no es utilizable.
- `h`: indica que la interfaz comparte una dirección de hardware físico duplicada con otra interfaz y que ha sido puesta fuera de línea. El indicador `h` significa que la interfaz es no utilizable.
- `i`: marca que se configura el indicador `INACTIVE` para la interfaz. Por lo tanto, la interfaz no se utiliza para enviar o recibir tráfico de datos.
- `m`: indica que la interfaz fue designada por el sistema para enviar y recibir tráfico de multidifusión IPv4 para el grupo IPMP.

- `M`: indica que la interfaz fue designada por el sistema para enviar y recibir tráfico de multidifusión IPv6 para el grupo IPMP.
- `s`: indica que la interfaz está configurada para ser una interfaz en espera.

`LINK` Indica el estado de la detección de fallos basada en enlaces, que es uno de los siguientes:

- `up` o `down`: indican la disponibilidad o la no disponibilidad de un enlace.
- `unknown`: indica que el controlador no admite la notificación que indica si un enlace está `up` o `down` y, por lo tanto, no detecta los cambios de estado del enlace.

`PROBE` Especifica el estado de la detección de fallos basada en sondeos para las interfaces que se configuraron con una dirección de prueba, de la siguiente manera:

- `ok`: indica que la sonda funciona y está activa.
- `failed`: indica que la detección de fallos basada en sondeos ha detectado que la interfaz no funciona.
- `unknown`: indica que no se pudieron encontrar destinos de sondeo adecuados y que, por lo tanto, no se enviaron sondeos.
- `disabled`: indica que no hay una dirección de prueba de IPMP configurada en la interfaz. Por lo tanto, la detección de fallos basada en sondeos está desactivada.

`ESTADO` Especifica el estado general de la interfaz, de la siguiente manera:

- `ok`: indica que la interfaz está en línea y que funciona normalmente según la configuración de los métodos de detección de fallos.
- `failed`: indica que la interfaz no funciona, ya sea porque el enlace de la interfaz está caído o porque la detección basada en sondeos determinó que la interfaz no puede enviar ni recibir tráfico.
- `offline`: indica que la interfaz no está disponible para su uso. Normalmente, la interfaz se cambia a sin conexión en las siguientes circunstancias:
  - La interfaz se está probando.
  - Se está realizando la reconfiguración dinámica.
  - La interfaz comparte una dirección de hardware duplicada con otra interfaz.
- `unknown`: indica que el estado de la interfaz IPMP no se puede determinar porque no se pueden encontrar destinos de sondeo para la detección de fallos basada en sondeos.

EJEMPLO 3-12 Obtención de información de destino de sondeo IPMP

La opción -t identifica los destinos de sondeo asociados a cada interfaz IP en un grupo IPMP. La salida en el siguiente ejemplo muestra una configuración IPMP que utiliza direcciones de prueba para la detección de fallos basada en sondeos.

```
% ipmpstat -nt
INTERFACE   MODE          TESTADDR        TARGETS
net0        routes        192.168.85.30   192.168.85.1 192.168.85.3
net1        disabled      --              --
net3        disabled      --              --
net4        routes        192.1.2.200      192.1.2.1
net2        multicast     192.168.10.200   192.168.10.1 192.168.10.2
net6        multicast     192.168.10.201   192.168.10.2 192.168.10.1
net5        multicast     192.168.10.202   192.168.10.1 192.168.10.2
net7        multicast     192.168.10.203   192.168.10.1 192.168.10.2
```

La siguiente salida muestra una configuración IPMP que utiliza el sondeo transitivo o la detección de fallos basada en sondeos sin direcciones de prueba.

```
% ipmpstat -nt
INTERFACE   MODE         TESTADDR         TARGETS
net3        transitive   <net1>           <net1> <net2> <net3>
net2        transitive   <net1>           <net1> <net2> <net3>
net1        routes       172.16.30.100    172.16.30.1
```

Los campos de salida proporcionan la siguiente información:

`INTERFACE` Especifica cada interfaz subyacente de cada grupo IPMP.

`MODE` Especifica el método para obtener los destinos de sondeo.

- `routes`: indica que se usa la tabla de enrutamiento del sistema para encontrar destinos de sondeo.
- `mcast`: indica que se usan sondas ICMP multidifusión para encontrar destinos de sondeo.
- `disabled`: indica que se desactivó la detección de fallos basada en sondeos para la interfaz.
- `transitive`: indica que se usa sondeo transitivo para la detección de fallos, como se muestra en el segundo ejemplo. Tenga en cuenta que no puede implantar la detección de fallos basada en sondeos si utiliza simultáneamente sondeos transitivos y direcciones de prueba. Si no desea utilizar las direcciones de prueba, debe activar el sondeo transitivo. Si no desea utilizar el sondeo transitivo, debe configurar direcciones de prueba. Para obtener una descripción general, consulte "Detección de fallos basada en sondeos" [63].

`TESTADDR` Especifica el nombre del host o, si la opción -n se utiliza junto con la opción -t, la dirección IP asignada a la interfaz para enviar y recibir sondeos.

Si se utiliza sondeo transitivo, entonces los nombres de interfaz hacen referencia a las interfaces IP subyacentes que no se usan activamente para recibir datos. Los nombres también indican que se envían sondeos de prueba transitivos con la dirección de origen de estas interfaces especificadas. Para las interfaces IP subyacentes activas que reciben datos, una dirección IP que se muestra indica la dirección de origen de los sondeos ICMP salientes.

Nota - Si una interfaz IP está configurada con la dirección de prueba IPv4 y la IPv6, la información de destino de prueba se muestra por separado para cada dirección de prueba.

`TARGETS` Muestra los destinos de sondeo actuales en una lista separada por espacios. Los destinos de sondeo se muestran como nombres de host o direcciones IP. Si la opción `-n` se utiliza con la opción `-t`, se muestran las direcciones IP.

EJEMPLO 3-13 Observación de sondeos IPMP

La opción `-p` le permite observar los sondeos en curso. Al utilizar esta opción con el comando `ipmpstat`, se muestra continuamente información sobre la actividad del sondeo hasta que finalice el comando presionando Control+C. Debe asumir el rol de usuario `root` o tener los privilegios adecuados para ejecutar este comando.

El siguiente es un ejemplo de una configuración IPMP que utiliza las direcciones de prueba para la detección de fallos basada en sondeos.

```
# ipmpstat -pn
TIME    INTERFACE   PROBE     NETRTT   RTT      RTTAVG     TARGET
0.11s   net0        589       0.51ms   0.76ms   0.76ms     192.168.85.1
0.17s   net4        612       --       --       --         192.1.2.1
0.25s   net2        602       0.61ms   1.10ms   1.10ms     192.168.10.1
0.26s   net6        602       --       --       --         192.168.10.2
0.25s   net5        601       0.62ms   1.20ms   1.00ms     192.168.10.1
0.26s   net7        603       0.79ms   1.11ms   1.10ms     192.168.10.1
1.66s   net4        613       --       --       --         192.1.2.1
1.70s   net0        603       0.63ms   1.10ms   1.10ms     192.168.85.3
^C
```

El siguiente es un ejemplo de una configuración IPMP que utiliza el sondeo transitivo o la detección de fallos basada en sondeos sin direcciones de prueba.

```
# ipmpstat -pn
TIME    INTERFACE   PROBE      NETRTT   RTT      RTTAVG     TARGET
1.39S   net4        t28        1.05ms   1.06ms   1.15ms     <net1>
1.39s   net1        i29        1.00ms   1.42ms   1.48ms     172.16.30.1
^C
```

Los campos de salida proporcionan la siguiente información:

`TIME`	Especifica el tiempo de envío de un sondeo en relación con la fecha de emisión del comando `ipmpstat`. Si un sondeo se inició antes de `ipmpstat`, entonces, la hora se muestra con un valor negativo, en relación a cuándo se emitió el comando.
`INTERFACE`	Especifica la interfaz en la que se envió el sondeo.
`PROBE`	Especifica el identificador que representa el sondeo. Si se utiliza sondeo transitivo para la detección de fallos, el identificador tiene un prefijo `t` en el caso de los sondeos transitivos o un prefijo `i` en el caso de los sondeos ICMP.
`NETRTT`	Especifica el total de tiempo de ida y vuelta en la red del sondeo, en milisegundos. `NETRTT` cubre el tiempo entre el momento en que el módulo IP envía la sonda y el momento en que el módulo IP recibe los paquetes `ack` desde el destino. Si el daemon `in.mpathd` ha determinado que el sondeo está perdido, el campo aparece vacío.
`RTT`	Especifica el total de tiempo de ida y vuelta del sondeo, en milisegundos. `RTT` abarca el tiempo entre el momento en que el daemon `in.mpathd` ejecuta el código para enviar el sondeo y el momento en que el daemon termina de procesar los paquetes `ack` desde el destino. Si el daemon ha determinado que el sondeo está perdido, el campo aparece vacío. Los picos que se producen en `RTT` que no están presentes en `NETRTT` podrían indicar que el sistema local está sobrecargado.
`RTTAVG`	Especifica el tiempo promedio de ida y vuelta del sondeo por la interfaz entre el sistema local y el destino. El tiempo de ida y vuelta promedio ayuda a identificar los destinos lentos. Si los datos no son suficientes para calcular el promedio, este campo estará vacío.
`TARGET`	Especifica el nombre del host. O, si se utiliza la opción `-n` junto con la opción `-p`, especifica la dirección de destino a la que el sondeo se envía.

Personalización de la salida del comando `ipmpstat`

La opción `-o` permite personalizar la salida del comando `ipmpstat`. Utilice esta opción junto con opciones mencionadas anteriormente de `ipmpstat` para seleccionar qué campos específicos se van a mostrar de todos los campos que la opción principal muestra habitualmente.

Por ejemplo, la opción `-g` proporciona la siguiente información:

- Grupo IPMP

- Nombre de grupo IPMP
- Estado del grupo
- Tiempo de detección de fallos
- Interfaces subyacentes del grupo IPMP

Supongamos que desea que se muestre únicamente el estado de los grupos IPMP en el sistema. Debe combinar las opciones `-o` y `-g`, y especificar los campos `groupname` y `state`, como se muestra en el siguiente ejemplo:

```
% ipmpstat -g -o groupname,state
GROUPNAME  STATE
ipmp0      ok
accgt1     failed
field2     degraded
```

Para mostrar todos los campos del comando `ipmpstat` para un tipo específico de información, incluya la opción `-o` con el argumento `all`.

Uso del comando `ipmpstat` en secuencias de comandos

La opción `-o` es útil cuando se ejecuta el comando `ipmpstat` desde una secuencia de comandos o mediante un alias de comando, especialmente si también desea generar una salida de análisis automático.

Para generar información de análisis automático, se combinan las opciones `-P` y `-o` con una de las otras opciones `ipmpstat` principales, junto con los campos específicos que desee mostrar.

La salida de análisis automático difiere de la salida normal de las siguientes maneras:

- Se omiten los encabezados de columna.
- Los campos se separan con dos puntos (`:`).
- Los campos con valores vacíos están justamente vacíos, en lugar de estar rellenos con el guión doble (`--`).
- Cuando se solicitan varios campos, si un campo contiene dos puntos (`:`) o una barra diagonal inversa (`\`). Puede omitir o excluir estos caracteres anteponiéndoles una barra diagonal inversa (`\`).

Para usar correctamente el comando `ipmpstat -P`, tenga en cuenta las siguientes reglas:

- Utilice la opción `-o` *option field* junto con la opción `-P`. Separe varios campos de opciones con comas.
- Nunca utilice la opción `-o all` con la opción `-P`.

Atención - Si ignora alguna de estas reglas, `ipmpstat -P` falla.

El ejemplo siguiente muestra la sintaxis correcta para utilizar la opción `-P`:

```
% ipmpstat -P -o -g groupname,fdt,interfaces
ipmp0:10.00s:net0 net1
acctg1::[net3 net4]
field2:20.00s:net2 net7 (net5) [net6]
```

El nombre del grupo, el tiempo de detección de fallos y las interfaces subyacentes son campos de información de grupos. Por lo tanto, puede utilizar las opciones `-o` y `-g` junto con la opción `-P`.

La opción `-P` está diseñada para ser utilizada en las secuencias de comandos. En el ejemplo siguiente se muestra cómo ejecutar el comando `ipmpstat` desde una secuencia de comandos. La secuencia de comandos muestra el tiempo de detección de fallos para un grupo IPMP.

```
getfdt() {
ipmpstat -gP -o group,fdt | while IFS=: read group fdt; do
[[ "$group" = "$1" ]] && { echo "$fdt"; return; }
done
}
```

♦ ♦ ♦ CAPÍTULO 4

Acerca de la administración de túneles IP

En este capítulo se proporciona una descripción general de la administración de túneles IP en Oracle Solaris. Para obtener información relacionada con tareas, consulte Capítulo 5, Administración de túneles IP.

Este capítulo se divide en los siguientes apartados:

Novedades de la administración de túneles IP

Se revisó la administración de túneles IP para que sea coherente con el nuevo modelo que se utiliza para la administración de enlaces de datos de red en Oracle Solaris 11. En esta versión, puede crear y configurar túneles IP mediante el comando `dladm`. Los túneles también pueden utilizar otras funciones de enlace de datos que se admiten en esta versión. Por ejemplo, la compatibilidad con nombres elegidos administrativamente permite asignar nombres más significativos a los túneles. Para obtener más información, consulte la página del comando man `dladm`(1M).

Resumen de la función Túnel IP

Los túneles IP (también denominados simplemente túneles en este manual) proporcionan un medio para transportar paquetes de datos entre dominios cuando el protocolo en esos dominios no es compatible con redes intermediarias. Por ejemplo, las redes IPv6 requieren una manera de comunicarse más allá de sus límites en un entorno donde la mayoría de las redes utilizan el protocolo IPv4. Esta comunicación es posible gracias al uso de túneles. Los túneles IP proporcionan un enlace virtual entre dos nodos a los que se puede acceder mediante IP. De esta forma, el enlace se puede utilizar para transportar paquetes IPv6 en redes IPv4 para permitir la comunicación IPv6 entre los dos sitios IPv6.

Tipos de túneles

La creación de túneles implica la encapsulación de un paquete IP dentro de otro paquete. Esta encapsulación permite que el paquete llegue a destino a través de redes intermediarias que no admiten el protocolo del paquete. Los túneles varían según el tipo de encapsulación de paquetes que se utiliza.

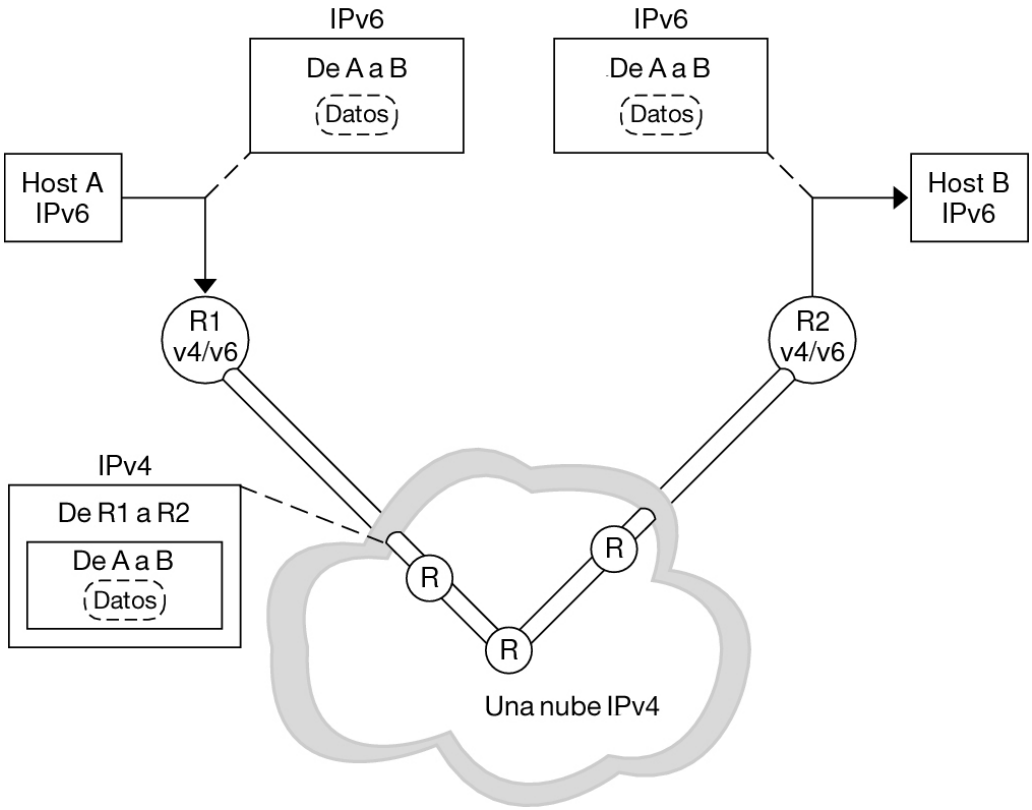
En Oracle Solaris, se admiten los siguientes tipos de paquetes:

- **túneles IPv4**: Los paquetes IPv4 se encapsulan en un encabezado IPv4 y se envían a un destino IPv4 de unidifusión preconfigurado. Para indicar más específicamente los paquetes que pasan por el túnel, los túneles IPv4 también se denominan *IPv4 en túneles IPv4* o *IPv6 en túneles IPv4*.
- **túneles 6to4**: Los paquetes IPv6 se encapsulan en un encabezado IPv4 y se envían a un destino IPv4 que se determina automáticamente según cada paquete. Esta determinación se basa en un algoritmo definido en el *protocolo 6to4*.
- **túneles IPv6**: Los paquetes IPv6 se encapsulan en un encabezado IPv4 y se envían a un destino IPv4 que se determina automáticamente según cada paquete. La determinación se basa en un algoritmo definido en el protocolo 6to4.

Túneles en los entornos de red IPv6 e IPv4 combinados

Muchos sitios tienen dominios IPv6 que posiblemente necesiten comunicarse con otros dominios IPv6 atravesando redes IPv4 durante las primeras fases de la implementación de IPv6. La siguiente figura ilustra el mecanismo de colocación en túneles (indicado por una "R" en la figura) entre dos hosts IPv6 a través de enrutadores IPv4.

FIGURA 4-1 Mecanismo de creación de túneles IPv6



En la figura anterior, el túnel está compuesto por dos enrutadores configurados con un enlace de punto a punto virtual entre los dos enrutadores en la red IPv4.

Un paquete IPv6 está encapsulado dentro de un paquete IPv4. El enrutador de límite de la red IPv6 configura un túnel de extremo a extremo a través de varias redes IPv4 hasta el enrutador de límite de la red IPv6 de destino. El paquete es transportado por el túnel hasta el enrutador de límite de destino, donde se desencapsula. A continuación, el enrutador reenvía el paquete IPv6 separado al nodo de destino.

Túneles 6to4

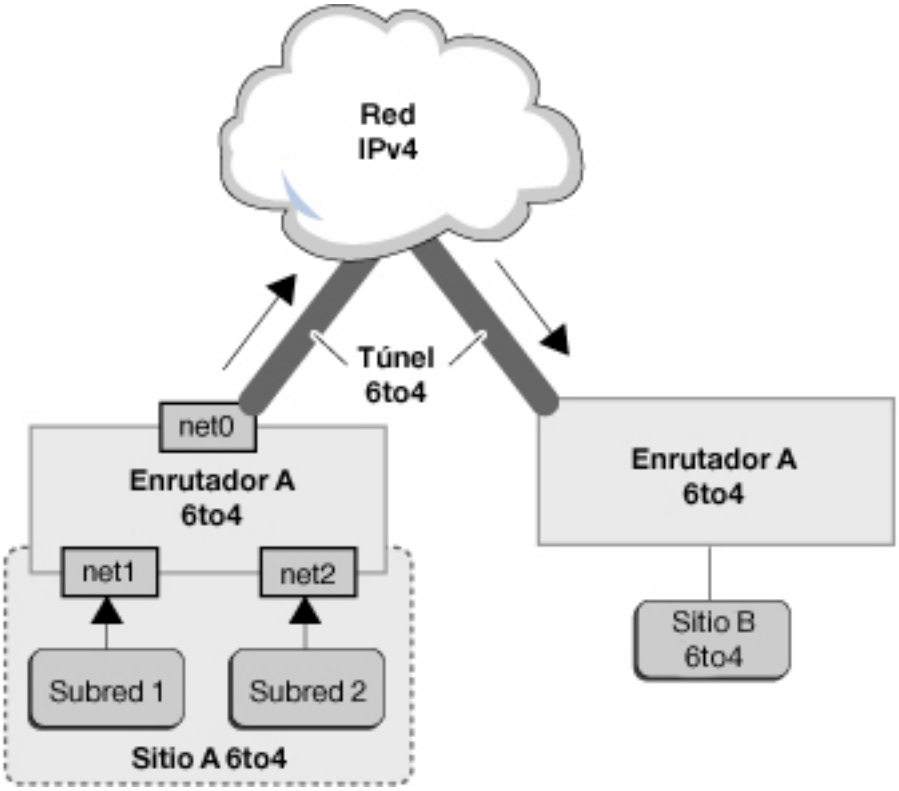
Oracle Solaris incluye túneles 6to4 como método provisional para realizar la transición de direcciones IPv4 a IPv6. Los *túneles 6to4* permiten que los sitios IPv6 aislados se comuniquen a través de un túnel automático en una red IPv4 que no admite IPv6. Para utilizar túneles 6to4,

primero debe configurar un enrutador de límite de sistema en la red IPv6 como un punto final del túnel automático 6to4. Después, el enrutador 6to4 puede participar en un túnel hasta otra ubicación 6to4, o, si es necesario, hasta un ubicación IPv6 nativa, no 6to4.

Configuración de un túnel 6to4

Un túnel 6to4 proporciona conectividad IPv6 a todas las ubicaciones 6to4 en cualquier parte. Asimismo, el túnel ejerce como enlace con todas las ubicaciones IPv6, incluida Internet IPv6 nativa, siempre que el enrutador se configure para reenviar a un enrutador de repetición. La figura siguiente ilustra la forma en que un túnel 6to4 proporciona esta clase de conectividad entre sitios 6to4.

FIGURA 4-2 Túnel entre dos ubicaciones 6to4



En la figura, se muestran dos redes 6to4 aisladas: sitio A y sitio B. Cada sitio tiene configurado un enrutador con una conexión externa a una red IPv4. Un túnel 6to4 en la red IPv4 proporciona una conexión para vincular ubicaciones 6to4.

Antes de que una ubicación IPv6 pueda convertirse en 6to4, debe configurar al menos una interfaz de enrutador para que admite 6to4. Esta interfaz debe proporcionar la conexión externa a la red IPv4. En la figura anterior, la interfaz `net0` del enrutador de límite de sistema encaminador A conecta la ubicación de sitio A con la red IPv4. La dirección configurada en `net0` debe ser única globalmente. Debe configurar la interfaz `net0` con una dirección IPv4 antes de poder configurar una interfaz de túnel para admitir 6to4 en el enrutador.

En la figura, sitio A 6to4 está compuesto por dos subredes conectadas a las interfaces `net1` y `net2` en el enrutador A. Todos los hosts IPv6 de la subredes del sitio A se reconfiguran automáticamente con direcciones derivadas de 6to4 al recibir el anuncio del enrutador A.

La ubicación de sitio B es otra ubicación 6to4 aislada. Para recibir correctamente tráfico de la ubicación de sitio A, se debe configurar un enrutador de límite en la ubicación sitio B para admitir 6to4. De no ser así, los paquetes que recibe el enrutador de sitio A no se reconocen y se descartan.

Comando `6to4relay`

El establecimiento de túneles de 6to4 permite las comunicaciones entre sitios de 6to4 que están aislados. Sin embargo, para transferir paquetes con un sitio de IPv6 nativo que no sea de 6to4, el enrutador de 6to4 debe establecer un túnel con un enrutador de relé de 6to4. Así, el *enrutador de relé de 6to4* reenvía los paquetes de 6to4 a la red IPv6 y, en última instancia, al sitio de IPv6 nativo. Si el sitio activado para 6to4 debe intercambiar datos con sitio de IPv6 nativo, utilice el comando `6to4relay` para activar el túnel correspondiente.

Nota - Como el uso de enrutadores de relé no es seguro, en Oracle Solaris de manera predeterminada se desactiva el establecimiento de túneles con un enrutador de relé. Antes de implementar esta situación hipotética, debe tener muy en cuenta los problemas que comporta crear un túnel con un enrutador de relé de 6to4. Para obtener más información, consulte “Consideraciones para activar túneles hasta un enrutador de reenvío 6to4” [104]. Si decide activar la compatibilidad con enrutadores de relé 6to4, consulte Cómo crear y configurar un túnel IP [110] para conocer los procedimientos relacionados.

Para obtener más información, consulte la página del comando man `6to4`(7M).

Flujo de paquetes a través del túnel 6to4

Esta sección describe el flujo de paquetes entre un hosts en una ubicación 6to4 y un host en una ubicación 6to4 remota. Esta situación hipotética utiliza la topología de la Figura 4-2, “Túnel entre dos ubicaciones 6to4”. En el ejemplo se considera que los enrutadores y hosts 6to4 ya están configurados.

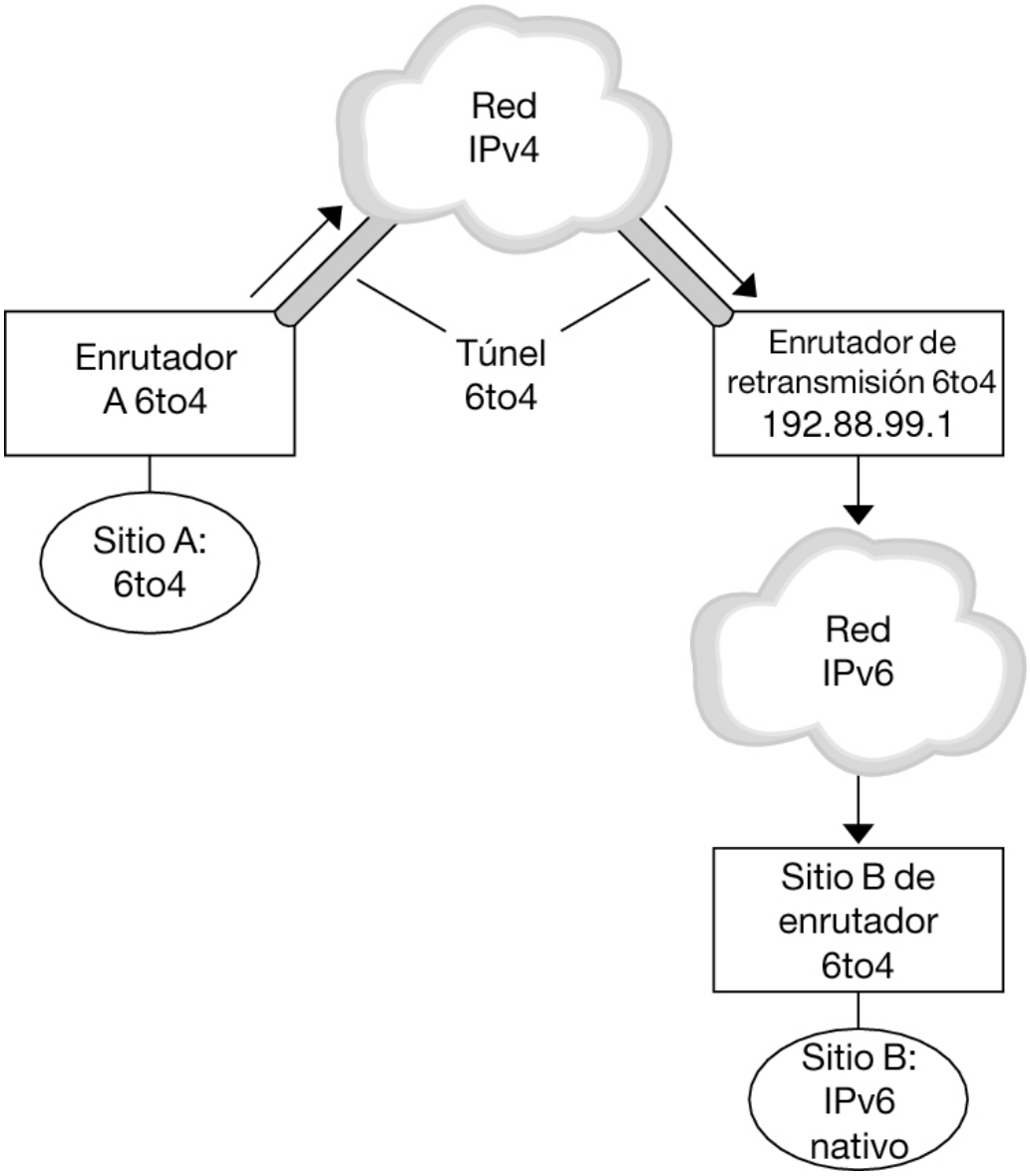
El flujo de paquete es el siguiente:

1. Un host en la subred 1 de la ubicación de sitio A de 6to4 envía una transmisión, con un host de la ubicación sitio B de 6to4 como destino. El encabezado de cada paquete tiene una dirección de origen derivada de 6to4 y una dirección de destino derivada de 6to4.
2. El enrutador de la ubicación sitio A encapsula cada paquete 6to4 dentro de un encabezado IPv4. En este proceso, el enrutador establece la dirección IPv4 de destino del encabezado de encapsulado en la dirección de enrutador de la ubicación de sitio B. En cada paquete de IPv6 que pasa por la interfaz de túnel, la dirección de destino de IPv6 también contiene la dirección de destino de IPv4. De este modo, el enrutador puede determinar la dirección IPv4 de destino que se establece en el encabezado de encapsulado. Después, el enrutador utiliza procedimientos estándar IPv4 para reenviar los paquetes a través de la red IPv4.
3. Cualquier enrutador IPv4 que encuentren los paquetes en su camino utilizará la dirección de destino IPv4 del paquete para reenviarlo. Esta dirección es la dirección IPv4 globalmente única de la interfaz del encaminador B, que también funciona como pseudointerfaz 6to4.
4. Los paquetes de sitio A llegan al encaminador B, que desencapsula los paquetes IPv6 del encabezado IPv4.
5. A continuación, el encaminador B utiliza la dirección de destino del paquete IPv6 para reenviar los paquetes al receptor en el sitio B.

Consideraciones para activar túneles hasta un enrutador de reenvío 6to4

Los enrutadores de reenvío 6to4 funcionan como puntos finales para túneles desde enrutadores 6to4 que necesitan comunicarse con redes IPv6 nativas, no 6to4. Los enrutadores de reenvío son básicamente puentes entre la ubicación 6to4 y ubicaciones IPv6 nativas. Debido a que esta solución puede llegar a ser muy insegura, de manera predeterminada, Oracle Solaris no tiene la admisión de enrutadores 6to4 activada. No obstante, si es necesario establecer un túnel de este tipo en su ubicación, puede utilizar el comando `6to4relay` para activar la creación de túneles, como se muestra en la siguiente figura.

FIGURA 4-3 Túnel desde una ubicación 6to4 hasta un enrutador de reenvío 6to4



En Figura 4-3, “Túnel desde una ubicación 6to4 hasta un enrutador de reenvío 6to4”, el sitio A 6to4 necesita comunicarse con un nodo en el sitio B IPv6 nativo. En la figura, se muestra la ruta de tráfico del sitio A al túnel 6to4 a través de una red IPv4. Los puntos finales del túnel son el

encaminador A de 6to4 y un enrutador de reenvío 6to4. Más allá del enrutador de reenvío 6to4 se encuentra la red IPv6, a la que está conectada la ubicación de sitio B IPv6.

Flujo de paquetes entre una ubicación 6to4 y una ubicación IPv6 nativa

En esta sección se describe el flujo de paquetes desde una ubicación 6to4 hasta una ubicación IPv6 nativa. Esta situación hipotética utiliza la topología de la Figura 4-3, “Túnel desde una ubicación 6to4 hasta un enrutador de reenvío 6to4”.

El flujo de paquete es el siguiente:

1. Un host en el sitio A 6to4 envía una transmisión que especifica como destino un host en el sitio B IPv6 nativo. El encabezado de cada paquete tiene una dirección derivada de 6to4 como dirección de origen. La dirección de destino es una dirección IPv6 estándar.
2. El enrutador 6to4 de la ubicación de sitio A encapsula cada paquete dentro de un encabezado IPv4, que tiene la dirección IPv4 del enrutador de reenvío 6to4 como destino. El enrutador 6to4 utiliza procedimientos IPv4 estándar para reenviar el paquete a través de la red IPv4. Cualquier enrutador IPv4 que encuentren los paquetes en su camino los reenviará al enrutador de reenvío 6to4.
3. El enrutador de reenvío 6to4 de difusión por proximidad más cercano físicamente a la ubicación de sitio A recibe los paquetes destinados al grupo de difusión por proximidad `192.88.99.1`.

Nota - Los enrutadores de reenvío 6to4 que forman parte del grupo de difusión por proximidad de enrutador de reenvío 6to4 tienen la dirección IP `192.88.99.1`. Esta dirección de difusión por proximidad es la dirección predeterminada de enrutadores de reenvío 6to4. Si necesita utilizar un enrutador de reenvío 6to4 específico, puede anular la dirección predeterminada y especificar la dirección IPv4 del enrutador.

---

4. El enrutador de reenvío desencapsula el encabezado IPv4 de los paquetes 6to4 y, de este modo, revela la dirección de destino IPv6 nativa.
5. A continuación, el enrutador de relé envía los paquetes que ahora son de sólo IPv6 a la red IPv6, donde, en última instancia, un enrutador del sitio B recupera los paquetes. Luego, el enrutador reenvía los paquetes al nodo IPv6 de destino.

Acerca de la implementación de túneles IP

Para implementar adecuadamente los túneles IP, debe realizar dos tareas principales. En primer lugar, cree el enlace de túnel. Luego, configure una interfaz IP en el túnel. Los siguientes son los requisitos para crear túneles y sus correspondientes interfaces IP.

Requisitos para crear túneles IP

Para crear túneles IP correctamente, debe tener cumplir los siguientes requisitos:

- Si utiliza nombres de host en lugar de direcciones IP literales, estos nombres deben remitir a direcciones IP válidas compatibles con el tipo de túnel.
- El túnel IPv4 o IPv6 que cree no debe compartir la misma dirección de origen ni la misma dirección de destino con otro túnel configurado.
- El túnel IPv4 o IPv6 que cree no debe compartir la misma dirección de origen con un túnel 6to4 existente.
- Si crea un túnel 6to4, ese túnel no debe compartir la misma dirección de origen con otro túnel configurado.

Para obtener más información, consulte “Planificación para el uso de túneles en la red” de “Planificación de la implementación de red en Oracle Solaris 11.2 ”.

Requisitos para túneles IP e interfaces IP

Cada tipo de túnel tiene requisitos de direcciones IP específicos para la interfaz IP que se configure en el túnel. Estos requisitos se resumen en la tabla siguiente.

TABLA 4-1 Requisitos de túneles e interfaces IP

Tipo de túnel	Interfaz IP permitida en el túnel	Requisito de interfaz IP
Túnel IPv4	Interfaz IPv4	Las direcciones locales y remotas se especifican manualmente.
	Interfaz IPv6	Las direcciones locales y remotas de enlace local se configuran automáticamente al emitir el comando `ipadm create-addr -T addrconf`. Consulte la página del comando man `ipadm`(1M).
Túnel IPv6	Interfaz IPv4	Las direcciones locales y remotas se especifican manualmente.
	Interfaz IPv6	Las direcciones locales y remotas de enlace local se configuran automáticamente al emitir el comando `ipadm create-addr -T addrconf`. Consulte la página del comando man `ipadm`(1M).
Túnel 6to4	Interfaz IPv6 únicamente	La dirección IPv6 predeterminada se selecciona automáticamente al ejecutar el comando `ipadm create-ip`. Consulte la página del comando man `ipadm`(1M).

Puede sustituir las direcciones de enlace local configuradas automáticamente para las interfaces IPv6 en túneles IPv4 o IPv6 mediante la especificación de un `interface-id` local y remoto con el comando `ipadm create-addr -T addrconf`.

♦ ♦ ♦ CAPÍTULO 5

Administración de túneles IP

En este capítulo se describen las tareas para administrar los túneles IP en Oracle Solaris. Para obtener una descripción general de la administración de túneles IP en Oracle Solaris, consulte Capítulo 4, Acerca de la administración de túneles IP.

Este capítulo se divide en los siguientes apartados:

Administración de túneles IP en Oracle Solaris

En esta versión de Oracle Solaris, la administración de túneles está separada de la configuración de la interfaz IP. Usted administra el aspecto del enlace de datos de túneles IP con el comando `dladm` y el aspecto IP de la configuración (incluidos aquellos para los túneles IP) con el comando `ipadm`.

Se utilizan los siguientes subcomandos `dladm` para configurar túneles IP:

- `create-iptun`
- `modify-iptun`
- `show-iptun`
- `delete-iptun`
- `set-linkprop`

Para obtener más información, consulte la página del comando man `dladm`(1M).

Nota - La administración de túneles IP está estrechamente relacionada con la configuración de IPsec. Por ejemplo, las redes privadas virtuales (VPN) IPsec constituyen uno de los principales usos de la creación de túneles IP. Para obtener más información sobre la seguridad de red en Oracle Solaris, consulte Capítulo 6, "Acerca de la arquitectura de seguridad IP" de "Protección de la red en Oracle Solaris 11.2 ". Para configurar IPsec, consulte Capítulo 7, "Configuración de IPsec" de "Protección de la red en Oracle Solaris 11.2 ".

Administración de túneles IP

Esta sección incluye los siguientes temas:

▼ Cómo crear y configurar un túnel IP

1. **Asuma el rol `root`.**

2. **Cree el túnel.**

```
# dladm create-iptun [-t] -T type -a [local|remote]=addr,... tunnel-link
```

`-t`	Crea un túnel temporal. De manera predeterminada, el comando crea un túnel persistente. Si desea configurar una interfaz IP persistente en el túnel, debe crear un túnel persistente y no utilizar la opción `-t`.
`-T` *type*	Especifica el tipo de túnel que desea crear. Este argumento es necesario para crear todos los tipos de túneles.
`-a [local|remote]=`*address*,...	Especifica los nombres de host o las direcciones IP literales que corresponden a la dirección local y a la dirección de túnel remota. Las direcciones deben ser válidas y ya deben estar creadas en el sistema. Según el tipo de túnel, debe especificar una sola dirección o ambas direcciones (locales y remotas). Si especifica direcciones locales y remotas, debe separarlas con una coma. ■ Los túneles `IPv4` requieren direcciones IPv4 locales y remotas para funcionar. ■ Los túneles `IPv6` requieren direcciones IPv6 locales y remotas para funcionar. ■ Los túneles `6to4` requieren una dirección IPv4 local para funcionar.

Nota - Para configuraciones de enlace de datos de túneles IP, si está utilizando nombres de host para las direcciones, estos nombres de host se guardan en el almacenamiento de la configuración. Durante un inicio posterior del sistema, si el nombre remite a direcciones IP distintas de las direcciones IP utilizadas cuando se creó el túnel, el túnel adquiere una nueva configuración.

tunnel-link Especifica el enlace de túnel IP. Al admitir nombres significativos en una administración de enlace de red en esta versión, los nombres de los túneles ya no se restringen al tipo de túnel que se está creando. En su lugar, puede asignar cualquier nombre elegido administrativamente a un túnel. Los nombres de túneles está formados por una cadena y el número de punto físico de conexión (PPA), por ejemplo, *mitúnel0*. Para conocer las reglas que rigen la asignación de nombres significativos, consulte "Reglas para nombres de enlaces válidos" de "Configuración y administración de componentes de red en Oracle Solaris 11.2 ".

3. (Opcional) Configure valores para el límite de salto o el límite de encapsulación.

```
# dladm set-linkprop -p [hoplimit=value] [encaplimit=value] tunnel-link
```

`hoplimit` Especifica el límite de salto de la interfaz de túnel para la creación de túneles en IPv6. El valor de *hoplimit* es equivalente al campo de tiempo de vida (TTL) de IPv4 para la creación de túneles en IPv4.

`encaplimit` Especifica el número de niveles de creación de túneles anidados permitidos para un paquete. Esta opción se aplica únicamente a túneles IPv6.

Los valores establecidos para las propiedades `hoplimit` y `encaplimit` deben estar dentro de rangos aceptables. Las propiedades `hoplimit` y `encaplimit` son propiedades de enlace de túnel. Por lo tanto, estas propiedades se administran con los mismos subcomandos `dladm` que otras propiedades de enlace. Los subcomandos que utiliza son `dladm set-linkprop`, `dladm reset-linkprop` y `dladm show-linkprop`.

4. Cree una interfaz IP en el túnel.

```
# ipadm create-ip tunnel-interface
```

Donde *interfaz_túnel* utiliza el mismo nombre que el enlace de túnel.

5. Asigne direcciones IP locales y remotas a la interfaz de túnel.

```
# ipadm create-addr [-t] -a local=address,remote=address interface
```

donde *interface* especifica la interfaz del túnel.

Para obtener más información, consulte la página del comando man `ipadm`(1M) y "Configuración y administración de componentes de red en Oracle Solaris 11.2 ".

6. **(Opcional) Verifique el estado de la configuración de la interfaz IP del túnel.**

```
# ipadm show-addr interface
```

ejemplo 5-1 Creación de una interfaz IPv6 en un túnel IPv4

El siguiente ejemplo muestra cómo crear un túnel IPv6 a través de IPv4 persistente.

```
# dladm create-iptun -T ipv4 -a local=192.0.2.23,remote=203.0.113.14 private0
# dladm set-linkprop -p hoplimit=200 private0
# ipadm create-ip private0
# ipadm create-addr -T addrconf private0
private0/v6
# ipadm show-addr private0/
ADDROBJ           TYPE     STATE        ADDR
private0/v6       addrconf ok fe80::c000:217->fe80::cb00:710e
```

Para agregar direcciones alternativas, use la misma sintaxis. Por ejemplo, puede agregar una dirección global de la siguiente manera:

```
# ipadm create-addr -a local=2001:db8:4728::1,remote=2001:db8:4728::2 private0
private0/v6a
# ipadm show-addr private0/
ADDROBJ           TYPE     STATE        ADDR
private0/v6       addrconf ok fe80::c000:217->fe80::cb00:710e
private0/v6a      static   ok 2001:db8:4728::1->2001:db8:4728::2
```

Tenga en cuenta que el prefijo `2001:db8` para las direcciones IPv6 es un prefijo IPv6 especial que se utiliza específicamente para ejemplos de documentación.

ejemplo 5-2 Creación de una interfaz IPv4 en un túnel IPv4

El siguiente ejemplo muestra cómo crear un túnel IPv4 a través de IPv4 persistente.

```
# dladm create-iptun -T ipv4 -a local=192.0.2.23,remote=203.0.113.14 vpn0
# ipadm create-ip vpn0
# ipadm create-addr -a local=10.0.0.1,remote=10.0.0.2 vpn0
vpn0/v4
# ipadm show-addr vpn0/
ADDROBJ           TYPE     STATE        ADDR
vpn0/v4           static   ok 10.0.0.1->10.0.0.2
```

Puede configurar, además, una política IPsec para proporcionar conexiones seguras para los paquetes que pasan por este túnel. Para obtener información, consulte Capítulo 7, "Configuración de IPsec" de "Protección de la red en Oracle Solaris 11.2 ".

ejemplo 5-3 Creación de una interfaz IPv6 en un túnel IPv6

El siguiente ejemplo muestra cómo crear un túnel IPv6 a través de IPv6 persistente.

```
# dladm create-iptun -T ipv6 -a local=2001:db8:feed::1234,remote=2001:db8:beef::4321 tun0
# ipadm create-ip tun0
# ipadm create-addr -T addrconf tun0
tun0/v6
# ipadm show-addr tun0/
ADDROBJ           TYPE     STATE        ADDR
tun0/v6           addrconf ok fe80::1234->fe80::4321
```

Par agregar direcciones, por ejemplo, una dirección global o direcciones locales y remotas alternativas, utilice el comando `ipadm` de la siguiente manera:

```
# ipadm create-addr -a local=2001:db8:cafe::1,remote=2001:db8:cafe::2 tun0
tun0/v6a
# ipadm show-addr tun0/
ADDROBJ           TYPE     STATE        ADDR
tun0/v6           addrconf ok fe80::1234->fe80::4321
tun0/v6a          static   ok 2001:db8:cafe::1->2001:db8:cafe::2
```

▼ Cómo configurar un túnel 6to4

Al configurar túneles 6to4, un enrutador 6to4 debe actuar como enrutador IPv6 para los nodos de los sitios 6to4 de la red. Por lo tanto, al configurar un enrutador 6to4, también debe configurar ese enrutador como enrutador IPv6 en las interfaces físicas. Para obtener más información sobre cómo configurar un host de Oracle Solaris como enrutador, consulte “Configuración de un enrutador IPv6” de “Configuración del sistema Oracle Solaris 11.2 como enrutador o equilibrador de carga ”.

1. **Cree un túnel 6to4.**

```
# dladm create-iptun -T 6to4 -a local=address  tunnel-link
```

-a local=*address* — Especifica la dirección local del túnel, que ya debe existir en el sistema para ser una dirección válida.

tunnel-link — Especifica el enlace de túnel IP. Al admitir nombres significativos en una administración de enlace de red, los nombres de los túneles ya no se restringen al tipo de túnel que se está creando. En su lugar, puede asignar a un túnel cualquier nombre elegido administrativamente. Los nombres de túneles está formados por una cadena y el número de PPA, por ejemplo, *mitúnel0*. Para obtener más información, consulte “Reglas para nombres de enlaces válidos” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.

2. **Cree la interfaz IP del túnel.**

```
# ipadm create-ip tunnel-interface
```

Donde *interfaz_túnel* utiliza el mismo nombre que el enlace de túnel.

3. **(Opcional) Agregue direcciones IPv6 alternativas para el uso del túnel.**

4. **Edite el archivo `/etc/inet/ndpd.conf`.**

```
# pfedit /etc/inet/ndpd.conf
```

5. **Anuncie el enrutamiento 6to4 mediante la agregación de las siguientes dos líneas al archivo.**

```
if subnet-interface AdvSendAdvertisements 1
IPv6-address subnet-interface
```

donde la primera línea especifica la subred que recibe el anuncio y la *subnet-interface* hace referencia al enlace al que está conectada la subred. La dirección IPv6 de la segunda línea debe tener el prefijo 6to4 `2000` que se utiliza para direcciones IPv6 en túneles 6to4.

Para obtener información detallada sobre el archivo `ndpd.conf`, consulte la página del comando man `ndpd.conf`(4).

6. **Active el reenvío de IPv6.**

```
# ipadm set-prop -p forwarding=on ipv6
```

7. **Elija una de las siguientes opciones:**

   - **Reinicie el enrutador.**
   - **Emita un `sighup` al daemon `/etc/inet/in.ndpd` para que empiece a enviar anuncios de enrutador.**

     Los nodos IPv6 de cada subred que recibirá el prefijo 6to4 se autoconfiguran con las nuevas direcciones derivadas 6to4.

8. **Agregue las nuevas direcciones derivadas 6to4 de los nodos al servicio de nombre utilizado en la ubicación 6to4.**

Para obtener instrucciones, consulte Capítulo 4, “Administración de servicios de nombres y directorios en un cliente de Oracle Solaris” de “Configuración y administración de componentes de red en Oracle Solaris 11.2 ”.

ejemplo 5-4 Creación de un túnel 6to4

El siguiente ejemplo muestra cómo se crea un túnel 6to4. Tenga en cuenta que únicamente las interfaces IPv6 se pueden configurar en túneles 6to4. En este ejemplo, la interfaz de subred es `net0` a la que hace referencia `/etc/inet/ndpd.conf`.

```
# dladm create-iptun -T 6to4 -a local=192.0.2.23 tun0
# ipadm create-ip tun0
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           dhcp     ok           192.0.2.23/24
lo0/v6            static   ok           ::1/128
tun0/v6           static   ok           2002:c000:217::1/16

# ipadm create-addr -T addrconf net0
net0/v6
# ipadm create-addr -a 2002:c000:217:cafe::1 net0
net0/v6a
# ipadm show-addr
ADDROBJ           TYPE     STATE        ADDR
lo0/v4            static   ok           127.0.0.1/8
net0/v4           dhcp     ok           192.0.2.23/24
lo0/v6            static   ok           ::1/128
net0/v6           addrconf ok     fe80::214:4fff:fef9:b1a9/10
net0/v6a          static   ok           2002:c000:217:cafe::1/64
tun0/v6           static   ok           2002:c000:217::1/16

# vi /etc/inet/ndpd.conf
if net0 AdvSendAdvertisements on
prefix 2002:c000:217:cafe::0/64 net0

# ipadm set-prop -p forwarding=on ipv6
```

Tenga en cuenta que para los túneles 6to4, el prefijo para la dirección IPv6 es `2002`.

▼ Cómo activar un túnel 6to4 hasta un enrutador de reenvío 6to4

Atención - Por problemas graves de seguridad, Oracle Solaris tiene desactivada la compatibilidad con enrutadores de reenvío 6to4 de manera predeterminada. Consulte "Cuestiones de seguridad al transmitir datos mediante túnel a un enrutador de reenvío 6to4" de "Resolución de problemas de administración de redes en Oracle Solaris 11.2 " para obtener detalles.

Antes de empezar

Antes de activar un túnel 6to4 hasta un enrutador de reenvío 6to4, debe haber realizado las siguientes tareas:

- Configure un enrutador 6to4 en el sitio. Consulte Cómo crear y configurar un túnel IP [110].
- Revise los problemas de seguridad relacionados con el establecimiento de un túnel hasta un enrutador de reenvío 6to4.

1. **Active un túnel hasta el enrutador de reenvío 6to4 utilizando uno de los siguientes métodos:**

   - **Activar un túnel a un enrutador de reenvío 6to4 de difusión por proximidad.**

     ```
     # 6to4relay -e
     ```

     La opción `-e` establece un túnel entre el enrutador 6to4 y un enrutador de reenvío 6to4 de difusión por proximidad. Los enrutadores de reenvío 6to4 de difusión por proximidad tienen la dirección IPv4 `192.88.99.1`. El enrutador de reenvío de difusión por proximidad que se encuentre más cerca físicamente de su ubicación pasa a ser el punto final del túnel 6to4. Este enrutador de reenvío gestiona el reenvío de paquetes entre su ubicación 6to4 y una ubicación IPv6 nativa.

     Para obtener más información, consulte RFC 3068, "An Anycast Prefix for 6to4 Relay Routers" (http://www.rfc-editor.org/rfc/rfc3068.txt).

   - **Active un túnel hasta un enrutador de reenvío 6to4 específico.**

     ```
     # 6to4relay -e -a relay-router-address
     ```

     La opción `-a` indica que a continuación se especifica una dirección de un enrutador determinado. Reemplace *dirección_enrutador_reenvío* con la dirección IPv4 del enrutador de reenvío 6to4 específico con el que quiera establecer un túnel.

   El túnel hasta el enrutador de reenvío 6to4 permanece activo hasta que se elimine la pseudointerfaz de túnel 6to4.

2. **Suprima el túnel hasta el enrutador de reenvío 6to4 cuando ya no sea necesario.**

   ```
   # 6to4relay -d
   ```

3. **(Opcional) Haga que el túnel hasta el enrutador de reenvío 6to4 se mantenga al reiniciar.**

   Es posible que en su ubicación sea necesario restablecer el túnel hasta el enrutador de reenvío 6to4 cada vez que se reinicia en enrutador 6to4. Para ello, debe hacer lo siguiente:

   a. **Edite el archivo `/etc/default/inetinit`.**

      ```
      # pfedit /etc/default/inetinit
      ```

      La línea que se debe modificar se encuentra al final del archivo.

   b. **Cambie el valor `NO` en la línea `ACCEPT6TO4RELAY=NO` a `YES`.**

   c. **(Opcional) Cree un túnel a un enrutador de reenvío 6to4 específico que se mantenga al reiniciar.**

En el parámetro `RELAY6TO4ADDR`, cambie la dirección `192.88.99.1` por la dirección IPv4 del enrutador de reenvío 6to4 que quiera usar.

ejemplo 5-5 Obtención de información de estado sobre la compatibilidad con enrutador de reenvío 6to4

Utilice el comando `6to4relay` para averiguar si la compatibilidad con enrutadores de reenvío 6to4 está activada. El siguiente ejemplo muestra la salida cuando la compatibilidad con enrutadores de reenvío 6to4 está desactivada, que es la opción predeterminada en Oracle Solaris.

```
# 6to4relay
6to4relay: 6to4 Relay Router communication support is disabled.
```

Cuando la compatibilidad con enrutadores de reenvío 6to4 está activada, se muestra la siguiente salida:

```
# 6to4relay
6to4relay: 6to4 Relay Router communication support is enabled.
IPv4 remote address of Relay Router=192.88.99.1
```

Modificación de la configuración de un túnel IP

Para modificar la configuración de un túnel, se utiliza la siguiente sintaxis de comando:

```
# dladm modify-iptun -a [local|remote]=addr,... tunnel-link
```

No puede modificar el tipo de un túnel existente. Por lo tanto, la opción `-T` *type* no se permite para este comando. Únicamente pueden modificarse los parámetros de túnel siguientes:

`-a [local|remote]=`*address*,...
: Especifica los nombres de host o las direcciones IP literales que corresponden a la dirección local y a la dirección de túnel remota. Según el tipo de túnel, debe especificar una sola dirección o ambas direcciones (locales y remotas). Si especifica direcciones locales y remotas, debe separarlas con una coma.

  - Los túneles `IPv4` requieren direcciones IPv4 locales y remotas para funcionar.
  - Los túneles `IPv6` requieren direcciones IPv6 locales y remotas para funcionar.
  - Los túneles `6to4` requieren una dirección IPv4 local para funcionar.

  Para configuraciones de enlace de datos de túneles IP, si está utilizando nombres de host para las direcciones, estos nombres de host se guardan en el almacenamiento de la configuración. Durante un inicio posterior del sistema, si el nombre remite a direcciones IP distintas de las direcciones IP utilizadas cuando se creó el túnel, el túnel adquiere una nueva configuración.

Si está cambiando las direcciones locales y remotas del túnel, asegúrese de que estas direcciones sean coherentes con el tipo de túnel que está modificando.

- Para cambiar el nombre del enlace de túnel, utilice el comando `dladm rename-link` en lugar del comando `modify-iptun` de la siguiente manera:

  `# dladm rename-link` *old-tunnel-link new-tunnel-link*

- Para cambiar las propiedades del túnel, como `hoplimit` o `encaplimit`, utilice el comando `dladm set-linkprop` en lugar del comando `modify-iptun`.

EJEMPLO 5-6 Modificación de la dirección y las propiedades de un túnel

El ejemplo siguiente consta de dos procedimientos. En primer lugar, las direcciones locales y remotas del túnel IPv4 `vpn0` se cambian temporalmente. Cuando el sistema se reinicia más adelante, el túnel vuelve a utilizar las direcciones originales. El segundo comando muestra cómo cambiar el valor `hoplimit` de `vpn0` a 60.

```
# dladm modify-iptun -t -a local=10.8.48.149,remote=192.168.2.3 vpn0

# dladm set-linkprop -p hoplimit=60 vpn0
```

Visualización de la configuración de un túnel IP

Para ver la configuración de un túnel, se utiliza la siguiente sintaxis de comando:

```
# dladm show-iptun [-p] -o fields [tunnel-link]
```

`-p` — Muestra la información en un formato que la máquina puede analizar. Este argumento es opcional.

`-o` *fields* — Muestra campos seleccionados que proporcionan información de un túnel específico.

tunnel-link — Especifica el túnel cuya información de configuración desea visualizar. Este argumento es opcional. Si omite el nombre del túnel, el comando muestra la información sobre todos los túneles del sistema.

EJEMPLO 5-7 Visualización de información sobre todos los túneles

En el ejemplo siguiente, sólo existe un túnel en el sistema.

```
# dladm show-iptun
LINK     TYPE     FLAGS     LOCAL          REMOTE
tun0     6to4     --       192.168.35.10   --
```

```
vpn0     ipv4     --       10.8.48.149     192.168.2.3
```

EJEMPLO 5-8 Visualización de campos seleccionados en un formato que la máquina puede analizar

En el siguiente ejemplo, únicamente se muestran campos específicos con información del túnel.

```
# dladm show-iptun -p -o link,type,local
tun0:6to4:192.168.35.10
vpn0:ipv4:10.8.48.149
```

Visualización de las propiedades de un túnel IP

Visualización de las propiedades del enlace de túnel mediante la siguiente sintaxis:

```
# dladm show-linkprop [-c] [-o fields] [tunnel-link]
```

-c	Muestra la información en un formato que la máquina puede analizar. Este argumento es opcional.
-o *fields*	Muestra campos seleccionados que proporcionan información sobre las propiedades del enlace.
tunnel-link	Especifica el túnel cuyas propiedades desea visualizar. Este argumento es opcional. Si omite el nombre del túnel, el comando muestra la información sobre todos los túneles del sistema.

EJEMPLO 5-9 Visualización de las propiedades de un túnel

En el ejemplo siguiente se muestra cómo visualizar todas las propiedades del enlace de un túnel.

```
# dladm show-linkprop iptun0
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
iptun0   autopush        rw   --           --           --        --
iptun0   zone            rw   --           --           --        --
iptun0   state           r-   up           up           up        up,down
iptun0   mtu             rw   1480         1480         1480      1280-1480
iptun0   maxbw           rw   --           --           --        --
iptun0   cpus            rw   --           --           --        --
iptun0   rxfanout        rw   --           8            8         --
iptun0   pool            rw   --           --           --        --
iptun0   priority        rw   medium       medium       medium    low,medium,
                                                                  high
iptun0   hoplimit        rw   64           64           64        1-255
iptun0   protection      rw   --           --           --        mac-nospoof,
                                                                  restricted,
                                                                  ip-nospoof,
                                                                  dhcp-nospoof
iptun0   allowed-ips     rw   --           --           --        --
```

```
iptun0   allowed-dhcp-cids rw --          --          --        --
iptun0   rxrings         rw   --          --          --        --
iptun0   txrings         rw   --          --          --        --
iptun0   txringsavail    r-   0           0           --        --
iptun0   rxringsavail    r-   0           0           --        --
iptun0   rxhwclntavail   r-   0           0           --        --
iptun0   txhwclntavail   r-   0           0           --        --
```

▼ Cómo suprimir un túnel IP

1. **Desconecte la interfaz IP configurada en el túnel según el tipo de interfaz.**

   ```
   # ipadm delete-ip tunnel-link
   ```

   Nota - Para suprimir correctamente un túnel, no puede conectarse en el túnel ninguna interfaz IP existente.

2. **Suprima el túnel IP.**

   ```
   # dladm delete-iptun tunnel-link
   ```

   La única opción para este comando es `-t`, que suprime el túnel temporalmente. Al reiniciar el sistema, se restaura el túnel.

ejemplo 5-10 Supresión de un túnel IPv6 configurado con una interfaz IPv6

En el ejemplo siguiente, se suprime permanentemente un túnel persistente.

```
# ipadm delete-ip ip6.tun0
# dladm delete-iptun ip6.tun0
```

índice

D

E

F

G

H

I

M

N

O

P

R

S

T

U