DHCP Failover Protocol

Motivación

- En las redes grandes el servicio DHCP es crítico.
- Es recomendable que varios servidores trabajen de forma **sicronizada** proporcionando **redundancia** por si uno de ellos falla.
- El **DHCP Failover Protocol** permite:
    - Que ambos servidores mantengan una **base de datos** de asignaciones **consistente**.
    - **Balanceo de carga** en la asignación de configuraciones.

Configuración en ISC-DHCP

Configuraremos dos servidores **ISC-DHCP** para que utilicen este protocolo y se sincronicen.

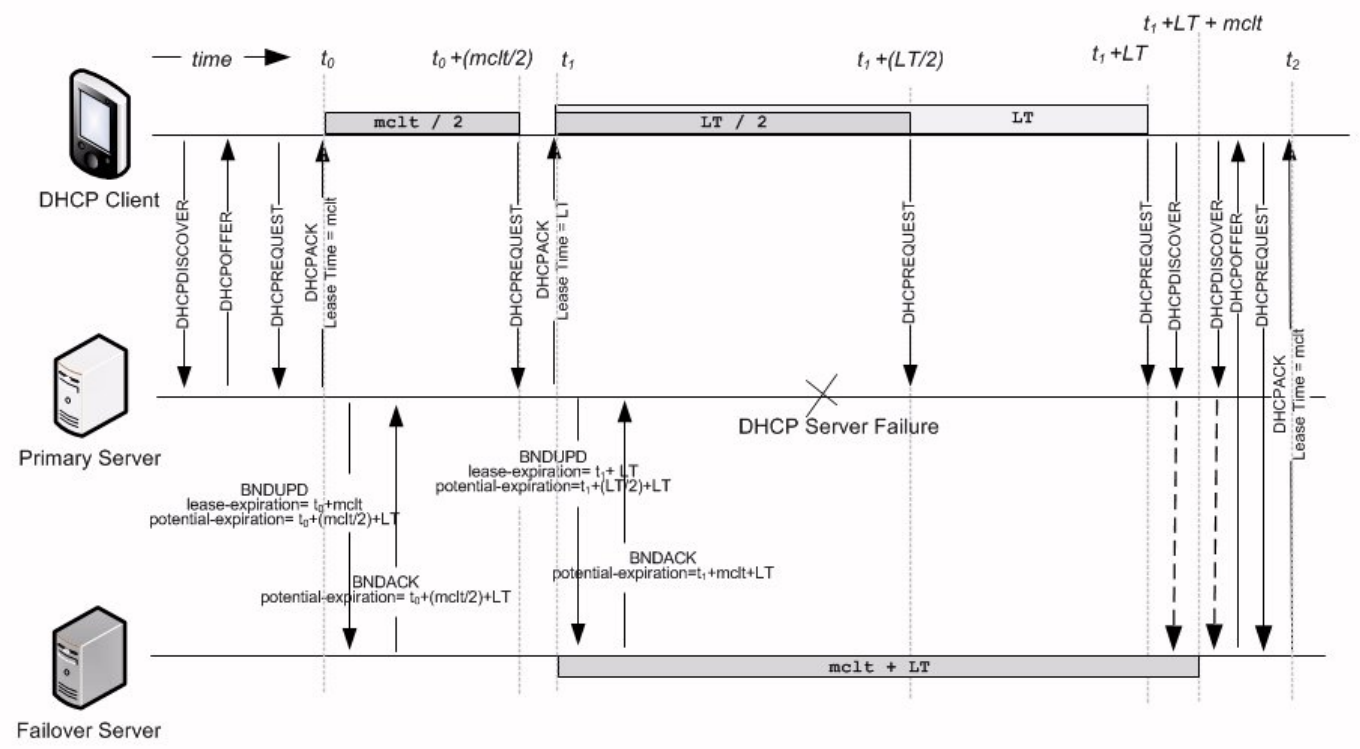
La configuración del protocolo **DHCP Failover** comienza con la declaración **failover peer** en cada uno de los dos servidores y se establecen los **parámetros de la relación** en los que se especifican cómo contactar con el servidor compañero. También se deben asociar los **pools** (o grupos) de direcciones que los servidores vayan a compartir.

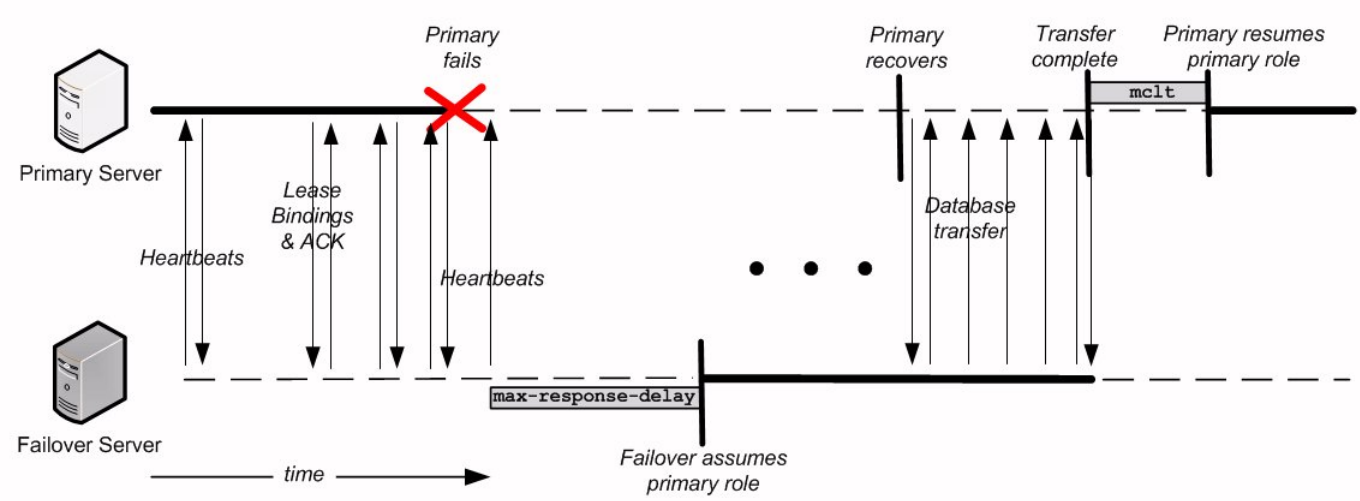
```
failover peer "nombre" {
    estamentos;
}
```

Estamentos

- **primary**: Especifica que el servidor actuará como primario (obligatorio).
- **secondary**: Especifica que el servidor actuará como secundario. No hay diferencia en cuanto al funcionamiento pero sí se configuran de forma distinta (obligatorio).
- **address direccion-IP**: Especifica la dirección IP del servidor que se está definiendo. Puede ser una IP numérica o un nombre (obligatorio).
- **port numero-puerto**: Especifica el puerto por el que escuchará el servidor los mensajes de sincronización de su compañero (opcional, por defecto **647**).
- **peer address direccion-IP**: Especifica la dirección IP del compañero (obligatorio).
- **peer port numero-puerto**: Especifica el puerto por el que escucha el servidor compañero los mensajes del protocolo failover (opcional, por defecto **647**).
- **max-unacked-updates número**: Especifica el número de mensajes de sincronización **BNDUPD** que pueden ser enviados desde uno de los servidores sin recibir el mensaje de aceptación **BDNACK** del otro servidor (obligatorio).
- **max-response-delay segundos**: Especifica el número de segundos que el servidor debe esperar sin recibir mensajes para concluir que la conexión con su pareja ha caído.
    - Debe de ser un tiempo razonable para que un corte accidental de unos pocos segundos no se interprete prematuramente como la caída de conexión, ya que posteriormente implicará la sincronización de ambos servidores (obligatorio).

- **mclt segundos**: Indica el tiempo por el que uno de los servidores extiende las concesiones de IP del servidor caído. También es el tiempo que el servidor caído estará sin servir IP cuando se recupere, dando tiempo a que caduquen las IP renovadas por el servidor que lo ha sustituido.
    - Esta declaración es obligatoria y **sólo se especifica en la configuración del servidor maestro**, que es quien lo comunicará al servidor esclavo.





Estas dos imágenes clafican el funcionamiento de **mclt** *y* **max-response-delay**.

- **split valor**: Siver para indicar que habrá balanceo entre los dos servidores DHCP, es decir, que se repartirán el trabajo de asignación (obligatorio).

    - Esta declaración solo se especifica en el **servidor maestro** y el valor es un número entre 0 y 255: 0 -> 0%, 128 -> 50%, 255 -> 100%.
        - El valor 128 (50%) es el más utilizado habitualmente.
        - El valor 255 (100%) indica que solo el servidor primario asignará IP y el secundario se utilizará como una copia de seguridad que el primario irá actualizando.

        - En este caso, el secundario solo se activaría si detectase que la conexión con el primario ha caído.

- **load balance max seconds segundos**: El efecto de este parámetro se produce cuando uno de los servidores responde a los mensajes de sincronización (está por lo tanto activo) pero no responde a los mensajes de solicitud de IP de los clientes porque **su parte del pool se ha agotado**.
    - Los clientes DHCP incluyen en la solicitud de IP el número de segundos que llevan haciendo la solicitud sin respuesta y, cuando los segundos son iguales o superiores a los especificados en el **load balance max seconds** entonces el otro servidor, que en principio no debería de atender la petición ya que recae en el ámbito del compañero, sí la atiende y la hace suya.

Ejemplo de configuración

Dos servidores ISC DHCP compartiendo al 50% un mismo pool:

Servidor DHCP maestro

```
authoritative;
ddns-update-style none;

failover peer "FAILOVER" {
  primary;
  address 192.168.1.2;
  port 647;
  peer address 192.168.1.3;
  peer port 647;
  max-unacked-updates 10;
  max-response-delay 30;
  load balance max seconds 3;
  mclt 1800;
  split 128;
}

subnet 192.168.1.0 netmask 255.255.255.0 {
  option broadcast-address 192.168.1.255;
  option routers 192.168.1.1;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
  pool {
    failover peer "FAILOVER";
    max-lease-time 3600;
    range 192.168.1.50 192.168.1.199;
  }
}
```

Servidor DHCP secundario

```
authoritative;
ddns-update-style none;
```

```
failover peer "FAILOVER" {
  secondary;
  address 192.168.1.3;
  port 647;
  peer address 192.168.1.2;
  peer port 647;
  max-unacked-updates 10;
  max-response-delay 30;
  load balance max seconds 3;
}

subnet 192.168.1.0 netmask 255.255.255.0 {
  option broadcast-address 192.168.1.255;
  option routers 192.168.1.1;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
  pool {
    failover peer "FAILOVER";
    max-lease-time 3600;
    range 192.168.1.50 192.168.1.199;
  }
}
```

Implementación

Clonación de máquinas virtuales

Partiendo de una máquina virtual con ISC-DHCP instalado, clonamos la máquina apagada en dos máquinas enlazadas con los nombres: - Servidor ISC DHCP Maestro - Servidor ISC DHCP Esclavo

Configuración del servidor maestro

Arrancamos el Servidor ISC DHCP Maestro.

Nos aseguramos que en VirtualBox esté en "Red Interna".

Debe tener una IP estática. Si no la tiene, debemos dársela creando un archivo de configuración para **netplan**.

```
sudo nano /etc/netplan/01-cfg-static-ip.yaml
```

Abrimos el fichero de configuración de IP y escribimos:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
     dhcp4: no
```

```
      addresses: [192.168.100.2/24]
      gateway4: 192.168.100.1
      nameservers:
        addresses: [8.8.8.8,8.8.4.4]
```

Salvamos y aplicamos los cambios:

```
sudo netplan apply
```

Netplan va a utilizar nuestro fichero **01-cfg-static-ip.yaml** para darnos una IP estática porque en orden alfabético está antes que el fichero que trae por defecto: **01-network-manager-all.yaml**.

Al escribir el comando:

```
ip a
```

Deberíamos ver que nuestra interfaz de red tiene la dirección estática **192.168.100.2**.

```
ubuntu@ubuntu2004:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 08:00:27:51:aa:86 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.2/24 brd 192.168.100.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe51:aa86/64 scope link tentative
       valid_lft forever preferred_lft forever
```

Primero debemos asegurarnos de que esté sirviendo configuraciones por su interfaz **enp0s3**, para ello revisamos el archivo **/etc/default/isc-dhcp-server**.

```
ubuntu@ubuntu2004:~$ cat /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf
```

```
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
#   Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#   Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s3"
INTERFACESv6=""
```

Debemos fijarnos en la línea **INTERFACESv4="enp0s3"**. Es ahí donde declaramos que nuestro servidor escuche peticiones por esa interfaz.

Y ahora nos toca actualizar su configuración en el archivo **/etc/dhcp/dhcpd.conf**:

```
authoritative;
ddns-update-style none;

failover peer "FAILOVER" {
  primary;
  address 192.168.100.2;
  port 647;
  peer address 192.168.100.3;
  peer port 647;
  max-unacked-updates 10;
  max-response-delay 30;
  load balance max seconds 3;
  mclt 1800;
  split 128;
}

subnet 192.168.100.0 netmask 255.255.255.0 {
  option broadcast-address 192.168.100.255;
  option routers 192.168.100.1;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
  pool {
    failover peer "FAILOVER";
    max-lease-time 3600;
    range 192.168.100.50 192.168.100.199;
  }
}
```

Configuración del servidor esclavo

Deberíamos reiniciar el servicio para que se apliquen los cambios, sin embargo vamos primero a configurar también el servidor esclavo:

- Arrancamos de la misma manera la máquina virtual esclavo que creamos.
- Nos aseguramos de que está en red interna y tiene como IP estática: 192.168.100.3. Si no la tiene, seguimos los pasos anteriores para configurarla.
- Nos aseguramos de que esté sirviendo concesiones a través de su interfaz **enp0s3**.

Tras estas configuraciones y verificaciones, editamos su archivo de configuración principal, el **/etc/dhcp/dhcpd.conf**:

```
authoritative;
ddns-update-style none;

failover peer "FAILOVER" {
  secondary;
  address 192.168.100.3;
  port 647;
  peer address 192.168.100.2;
  peer port 647;
  max-unacked-updates 10;
  max-response-delay 30;
  load balance max seconds 3;
}

subnet 192.168.100.0 netmask 255.255.255.0 {
  option broadcast-address 192.168.100.255;
  option routers 192.168.100.1;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
  pool {
    failover peer "FAILOVER";
    max-lease-time 3600;
    range 192.168.100.50 192.168.100.199;
  }
}
```

Comprobación de la fecha y hora

Para que el servicio funcione correctamente, ambos servidores deben compartir fecha y hora. Al deribar de la misma máquina virtual, la hora de ambas máquinas será la misma.

Podemos comprobarlo con:

```
date
```

La sincronización de fecha y hora se puede conseguir también con un servidor **NTP** (Network Time Protocol) o algún mencanismo alternativo.

Si la diferencia horaria entre servidores es mayor que un minuto, el proceso de configuración del protocolo failover se parará e indicará que la hora de los servidores tiene que estar sincronizada.

Probando la configuración

Reiniciamos los servidores en ambas máquinas para que carguen la nueva configuración y comiencen a servir concesiones en el rango de IPs: **192.168.100.50-199**.

```
sudo service isc-dhcp-server restart
```

Ahora podemos comprobar que conceden direcciones a clientes y que sus bases de datos están en sincronía. Para ello, creamos una nueva máquina virtual de Ubuntu y vamos a conectarla a la red interna.

Si en la consola del cliente escribimos:

```
sudo dhclient -r
```

El cliente liberará la concesión otorgada, de tener una.

Y si escribimos:

```
sudo dhclient
```

El cliente solicitará una nueva configuración.

Si se incluye la opción *-v*, de *verbose*, el programa cliente escribirá lo que va haciendo en la consola.

Es posible que no haga falta escribir estos comandos si la máquina virtual tiene activado el **DHCP**.

Debemos comprobar que nuestro cliente tiene una IP dentro del rango establecido:

```
ubuntu@ubuntu2004:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 08:00:27:ce:bf:8c brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.124/24 brd 192.168.100.255 scope global dynamic noprefixroute
enp0s3
       valid_lft 1736sec preferred_lft 1736sec
    inet6 fe80::4f7f:2b1f:b2dc:36f1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Si escribimos el comando **dhcp-lease-list** en las dos máquinas virtuales de servidores, veríamos que las dos nos muestran la concesión dada al cliente:

```
ubuntu@ubuntu2004:~$ dhcp-lease-list
To get manufacturer names please download
http://standards.ieee.org/regauth/oui/oui.txt to /usr/local/etc/oui.txt
Reading leases from /var/lib/dhcp/dhcpd.leases
MAC                IP              hostname       valid until         manufacturer
==================================================================================
=============
08:00:27:ce:bf:8c  192.168.100.124 -NA-           2022-10-04 12:31:56 -NA-
```

Por lo tanto, están compartiendo una base de datos de concesiones.

Ahora podemos parar el servicio del servidor principal o maestro con:

```
sudo service isc-dhcp-server stop
```

Y liberar y obtener configuración desde el cliente, y ver que el servidor secundario es quien nos ha atendido.

```
ubuntu@ubuntu2004:~$ sudo dhclient -r
ubuntu@ubuntu2004:~$ sudo dhclient
ubuntu@ubuntu2004:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 08:00:27:ce:bf:8c brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.124/24 brd 192.168.100.255 scope global dynamic noprefixroute
enp0s3
       valid_lft 1377sec preferred_lft 1377sec
    inet 192.168.100.123/24 brd 192.168.100.255 scope global secondary dynamic
enp0s3
       valid_lft 1799sec preferred_lft 1799sec
    inet6 fe80::4f7f:2b1f:b2dc:36f1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Analizando el tráfico

Podemos observar mejor este proceso con **Wireshark**. Así que vamos a instalar la herramienta en la máquina cliente.

Como nuestro cliente está en Red Interna, no va a tener conectividad hacia el exterior, por lo que debemos de cambiar temporalmente su red a NAT o a Puente (Bridge) en VirtualBox y solicitar una configuración dhcp en el caso de que nuestro equipo no lo haga automáticamente:

```
sudo dhclient
```

Actualizamos repos e instalamos el Wireshark.

```
sudo apt-get update -y & sudo apt-get install wireshark -y
```

En la instalación gráfica del wireshark, permitimos a usuarios que no son administradores, capturar paquetes. Si nos hemos confundido y no hemos dado permiso, podemos remediarlo añadiendo nuestro usuario al grupo de los usuarios autorizados:

```
sudo usermod -aG wireshark $(whoami)
```

Reiniciamos la máquina virtual cliente:

```
sudo reboot
```

Y al iniciarse de nuevo Ubuntu, encontraremos Wireshark en el menú de las aplicaciones, pero vamos abrirlo desde la terminal con el comando:

```
sudo wireshark
```

Es importante correr wireshark como super usuario porque si no, no tendremos acceso a nuestras interfaces y no podremos elegir la **enp0s3** para capturar el tráfico.

Nos establecemos de nuevo en **red interna** en virtualbox y en el Wireshark establecemos el filtro **dhcp**.

Si escribimos en la terminal:

```
sudo dhclient -r
sudo dhclient
```

Podremos capturar en Wireshark el proceso DORA (Discover, Offer, Request, ACK) y ver que quien nos está sirviendo es el servidor secundario: **192.168.100.3**.

Ahora vamos a parar el servidor secundario:

```
sudo service isc-dhcp-server stop
```

Y a iniciar el servidor DHCP maestro:

```
sudo service isc-dhcp-server start
```

En la consola del cliente escribimos los comandos para liberar la concesión:

```
sudo dhclient -r
```

En el Wireshark nos aparecerá un mensaje del protoclo **ICMP** diciendo **Destination Unreachable** ya que nuestro cliente envió un **DHCP RELEASE** al servidor secundario que ahora no se encuentra activo.

Y renovamos la concesión:

```
sudo dhclient
```

Y veremos en el Wireshark que quien nos atiende es el servidor primario.

Fuentes

- FP en red.
- Microsoft - DHCP Failover Architecture