

## Unknown Title

---

ChatGPT●

Tú dijiste:

como estableces más de una ruta estática: network: version: 2 ethernet: ens3: # match: # macaddress: 0c:3a:40:aa:00:00 # set-name: ens3 # dhcp4: false addresses: - 192.168.1.17/24 routes: - to: 0.0.0.0/0 via: 192.168.1.2 nameservers: addresses: - 172.16.200.1 - 8.8.8.8

ChatGPT Plus

Perfecto 🧙 estás usando **Netplan**, y la clave está en el bloque `routes` :: puedes definir **varias rutas estáticas** simplemente añadiendo más entradas dentro de esa lista.

Aquí te muestro un ejemplo **correctamente indentado y funcional**, basado en tu configuración:

yaml

```
network:
  version: 2
  ethernet:
    ens3:
      dhcp4: false
      addresses:
```

```

- 192.168.1.17/24
routes:
- to: 0.0.0.0/0          # Ruta por defecto
  via: 192.168.1.2
- to: 10.0.0.0/24        # Segunda ruta estática
  via: 192.168.1.1
- to: 172.16.0.0/16      # Tercera ruta estática
  via: 192.168.1.3
nameservers:
addresses:
- 172.16.200.1
- 8.8.8.8

```

#### ♦ Puntos importantes:

- Cada ruta se define con `- to:` y `via:`.
- Todas las rutas van bajo `routes:` (con la misma indentación).
- Si quieres rutas solo para ciertas subredes, cambia la dirección en `to:` (no pongas `0.0.0.0/0`, que es la ruta por defecto).
- Asegúrate de mantener la indentación con **espacios**, no **tabulaciones**, o Netplan fallará.

Después de editar el archivo (por ejemplo `/etc/netplan/01-netcfg.yaml`), aplica los cambios con:

bash

```
sudo netplan apply
```

¿Quieres que te muestre cómo poner rutas **balanceadas o de respaldo** (por ejemplo, con `metric:`)?

Tú dijiste:

```

Comprueba si esto esta bien? ((DHCP-PRIM)) failover peer "FAILOVER" { primary; address
192.168.1.17; port 647; peer address 192.168.1.27; peer port 647; max-unacked-updates 10; max-
response-delay 30; load balance max seconds 3; mclt 1800; split 255; } ((DHCP-SECOND)) failover peer
"FAILOVER" { secondary; address 192.168.1.27; port 647; peer address 192.168.1.17; peer port 647;
max-unacked-updates 10; max-response-delay 30; load balance max seconds 3; mclt 1800; }

```

ChatGPT Plus

✅ Tu configuración de **failover DHCP (ISC DHCPD)** está **casi perfecta**, pero hay un **pequeño ajuste importante** para que funcione de forma totalmente correcta y simétrica.

Vamos a revisar cada parte 📌

## ■ Servidor Primario (DHCP-PRIM) — correcto con un detalle:

bash

```
failover peer "FAILOVER" {
    primary;
    address 192.168.1.17;
    port 647;
    peer address 192.168.1.27;
    peer port 647;
    max-unacked-updates 10;
    max-response-delay 30;
    load balance max seconds 3;
    mclt 1800;
    split 255;
}
```

### ✓ Comentarios:

- `primary;` → correcto.
- `mclt 1800;` (30 min) → obligatorio en el primario y **solo en el primario**.
- `split 255;` → significa que **todo el tráfico inicial lo maneja el primario**, lo cual es válido si quieres que el secundario solo actúe en caso de fallo.
  - Si quieres balancear, puedes usar `split 128;` (mitad y mitad).

---

## ■ Servidor Secundario (DHCP-SECOND) — pequeño detalle:

bash

```
failover peer "FAILOVER" {
    secondary;
    address 192.168.1.27;
    port 647;
    peer address 192.168.1.17;
    peer port 647;
    max-unacked-updates 10;
    max-response-delay 30;
    load balance max seconds 3;
    mclt 1800;    # ✗ Esto no debe ir en el secundario
}
```

### ⊘ Error menor:

En el **secundario**, no debe aparecer la línea `mclt`.

Ese parámetro solo se define en el **primario**.

---

## ✅ Configuración corregida (perfectamente funcional)

### DHCP-PRIM:

bash

```
failover peer "FAILOVER" {
    primary;
    address 192.168.1.17;
    port 647;
    peer address 192.168.1.27;
    peer port 647;
    max-unacked-updates 10;
    max-response-delay 30;
    load balance max seconds 3;
    mclt 1800;
    split 255;
}
```

### DHCP-SECOND:

bash

```
failover peer "FAILOVER" {
    secondary;
    address 192.168.1.27;
    port 647;
    peer address 192.168.1.17;
    peer port 647;
    max-unacked-updates 10;
    max-response-delay 30;
    load balance max seconds 3;
}
```

---

## 💡 Recomendación final

En tus bloques de **subnet**, recuerda incluir el `failover peer "FAILOVER";` dentro de los `pool { ... }` así:

bash

```
pool {
    failover peer "FAILOVER";
```

```
range 192.168.1.50 192.168.1.100;  
}
```

¿Quieres que te muestre también cómo verificar que ambos servidores sincronizan correctamente (comandos y logs de estado)?

Tú dijiste:

Dime que significa cada uno: failover peer "FAILOVER" { primary; address 192.168.1.17; port 647; peer address 192.168.1.27; peer port 647; max-unacked-updates 10; max-response-delay 30; load balance max seconds 3; mclt 1800; split 255; }

ChatGPT Plus

Perfecto 🙌

Te explico **línea por línea** qué significa cada parámetro del bloque failover peer en **ISC DHCPD**. Esto define cómo dos servidores DHCP cooperan (primario y secundario) para compartir cargas y mantener alta disponibilidad.

## 🧩 Desglose completo

bash

```
failover peer "FAILOVER" {  
    primary;  
    address 192.168.1.17;  
    port 647;  
    peer address 192.168.1.27;  
    peer port 647;  
    max-unacked-updates 10;  
    max-response-delay 30;  
    load balance max seconds 3;  
    mclt 1800;  
    split 255;  
}
```

### ■ failover peer "FAILOVER"

- **Qué hace:**

Define un bloque de configuración de failover (alta disponibilidad) entre dos servidores DHCP.

- "FAILOVER" es simplemente un **nombre de la pareja**; debe ser idéntico en ambos servidores.

Ejemplo: failover peer "FAILOVER"; luego se usa dentro de los pool { ... }.

---

### **primary;**

- Indica que **este servidor es el primario**.
  - Solo **uno** de los dos puede tener primary;, el otro tendrá secondary;.
  - El **primario** es el que inicializa el estado y posee autoridad cuando ambos se sincronizan por primera vez.
- 

### **address 192.168.1.17;**

- Es la **IP local** del servidor (la que usa para comunicarse con su par).
  - Debe ser **alcanzable desde el otro servidor**.
- 

### **port 647;**

- Es el **puerto TCP** en el que este servidor **escucha** las conexiones de failover.
  - Por defecto, el estándar de DHCP failover usa el **puerto 647**.
- 

### **peer address 192.168.1.27;**

- IP del **otro servidor DHCP** (el compañero).
  - Es a donde este servidor se conectará para intercambiar estado y concesiones (leases).
- 

### **peer port 647;**

- Puerto TCP en el que **el otro servidor** escucha el protocolo de failover.  
(Normalmente también el **647**).
- 

### **max-unacked-updates 10;**

- Número máximo de **actualizaciones de concesiones** (leases) que este servidor puede enviar **sin recibir confirmación** del otro.
  - Evita que se acumulen demasiados cambios en un servidor sin que el otro los registre.  
👉 Si llega a 10 sin respuesta, suspende el envío hasta sincronizar.
- 

### **max-response-delay 30;**

- Tiempo máximo (en segundos) que un servidor esperará una **respuesta del otro** antes de considerarlo “no disponible”.
- Si pasan más de 30 s sin respuesta → entra en modo **partner-down** (activa failover).

---

### **load balance max seconds 3;**

- Define cuántos segundos de diferencia en el tiempo de respuesta se toleran antes de decidir cuál servidor atiende una solicitud DHCP.
- En pocas palabras: controla **el reparto de carga** entre ambos servidores.
  - Cuanto menor el número → más balanceo.
  - Si uno responde lento, el otro toma la petición.

---

### **mclt 1800;**

- **Maximum Client Lead Time** (Tiempo máximo de concesión adelantada).
- Solo se configura en el **primario**.
- Significa: el secundario puede **extender las concesiones existentes** hasta 1800 s (30 min) si el primario está caído.
- Garantiza que los clientes no pierdan IPs durante una caída temporal del primario.

---

### **split 255;**

- Controla cómo se reparte la **asignación inicial de clientes nuevos**:
  - Rango 0–255 → porcentaje de carga para el primario.
  - `split 128;` → 50/50.
  - `split 255;` → el **primario maneja todas las nuevas solicitudes** (el secundario solo entra en acción si el primario falla).
- Ideal si quieres un **modo activo/pasivo** (no balanceado).

---

## **Resumen rápido**

Parámetro	Rol	Explicación breve
primary	Identifica al servidor principal	Inicializa y controla el failover
address	IP local	Desde donde se comunica
peer address	IP del compañero	Con quién sincroniza

Parámetro	Rol	Explicación breve
port / peer port	TCP 647	Puerto de comunicación failover
max-unacked-updates	10	Cuántos leases se envían sin confirmación
max-response-delay	30 s	Tiempo de espera antes de considerar al otro caído
load balance max seconds	3 s	Límite de tiempo para repartir carga
mclt	1800 s	Tiempo máximo de renovación sin el primario
split	0–255	Porcentaje de balanceo entre servidores

¿Quieres que te muestre un **esquema visual** (tipo diagrama) de cómo se comunican ambos servidores y cómo actúa cada parámetro en el failover?

ChatGPT puede cometer errores. Considera verificar la información importante. Ver [preferencias de cookies](#).