

[Redes] Comandos: tracepath

Función

- `tracepath` es una herramienta de diagnóstico de red en Linux que rastrea la ruta desde nuestro equipo hasta un host remoto, similar a `traceroute`, pero integrando también información sobre la MTU (Unidad de Transmisión Máxima) y la latencia en cada salto.
- No requiere permisos especiales (a diferencia de algunas versiones de `traceroute`) y suele usarse para depurar problemas de red y de MTU.
- Muestra la secuencia de enrutadores (saltos) por los que pasan los paquetes hacia un destino (IP o nombre de dominio).
- Incluye el tiempo de ida-y-vuelta (RTT) en cada salto y la MTU efectiva en el camino, útil para detectar cuellos de botella o problemas de fragmentación.

Sintaxis

- En la mayoría de distribuciones de Linux, la sintaxis es muy simple:

```
tracepath [opciones] <destino>
```

donde:

1. `<destino>` es una IP (`8.8.8.8`) o un nombre de host (`google.com`).
2. Las opciones habituales son:

Opción	Ejemplo	Descripción resumida
<code>-n`</code>	<code>tracepath -n google.com`</code>	No resuelve nombres DNS; muestra solo IPs.
<code>-b`</code>	<code>tracepath -b google.com`</code>	Muestra tanto IPs como nombres de host.
<code>-l <MTU>`</code>	<code>tracepath -l 1400 google.com`</code>	Establece el tamaño inicial del paquete (MTU).
<code>-h <n>`</code>	<code>tracepath -h 30 google.com`</code>	Máximo número de saltos (hops).

Para ver todas las opciones, ejecutar:

```
tracepath --help
```

o

```
man tracepath
```

Ejemplos prácticos

- **1. Rastreo básico a un dominio**

```
tracepath google.com
```

Muestra la ruta hacia Google incluyendo número de salto, IP, RTT y MTU en cada enlace.

- **2. Sin resolución DNS (solo IPs)**

```
tracepath -n google.com
```

Ahorra tiempo si hay problemas de DNS.

- **3. Con tamaño de paquete limitado (MTU)**

```
tracert -l 1400 google.com
```

Útil para detectar si algún enlace rompe la conexión por una MTU más baja.

• 4. Limitar número máximo de saltos

```
tracert -h 10 google.com
```

Detiene el rastreo a 10 saltos aunque no haya llegado al destino.

Ejemplo típico explicado

Un ejemplo típico de salida de `tracert` en Linux es algo como:

```
tracert google.com
 1?: [LOCALHOST]          pmtu 1500
 1:  _gateway              1.234ms
 1:  _gateway              0.987ms
 2:  10.236.32.1           3.456ms
 3:  no reply
 4:  172.16.100.1         5.678ms
    pmtu 1492
 ...
10:  142.250.179.174     24.789ms
    reached
Resume: pmtu 1492 hops 10 back 10
```

Vamos a desglosar campo por campo para poder interpretarlo en nuestro entorno de red.

1. Línea inicial y PMTU local

```
1?: [LOCALHOST]          pmtu 1500
```

1. `1?:` → primer “salto” (hop 1), marcado como `?` porque aún no se ha visto respuesta ICMP válida.
2. `[LOCALHOST]` → indica que el primer salto es desde tu propia máquina; no se ha identificado todavía el primer enrutador.
3. `pmtu 1500` → Path MTU detectado hasta este punto es 1500 bytes (MTU típico de Ethernet).

- **Utilidad:** Nos dice el tamaño de paquete máximo que podemos usar en nuestro segmento local sin fragmentación; si vemos valores bajos (1492, 1480, etc.) suele indicar PPPoE, VLAN, o túneles.

2. Salto con IP y nombre (`_gateway`)

```
1:  _gateway              1.352ms
1:  _gateway              1.426ms
```

1. `1:` → número de salto.
2. `_gateway` → nombre del primer enrutador (normalmente tu router de acceso a la LAN; puede ser `192.168.1.1`, `10.0.0.1`, etc.).
3. `1.352ms`, `1.426ms` → RTT (round-trip time) para ese salto; `tracert` suele hacer varias pruebas por hop.

- Interpretación en nuestra red:
 1. Un salto `1:` con tiempos bajos (< 2 ms) indica que nuestra LAN está sana.
 2. Si son muy altos o aparecen `no reply` en el primer salto, revisar switch, router o incluso ARP.

3. Saltos intermedios

```
2: 10.236.32.1 3.764ms
3: no reply
4: 172.16.100.1 5.678ms
```

1. `2:`, `3:`, `4:` → segundos, terceros, cuartos enrutadores en la ruta.
2. `10.236.32.1 / 172.160.100.1` → IP del enrutador o dispositivo de la red.
3. `3: no reply` → el enrutador en el salto 3 no responde a ICMP (firewall bloqueando, política de seguridad, etc.).
4. `5.678ms` → RTT acumulado hasta ese nodo.

- Para un entorno como el nuestro:
 1. Si saltos intermedios aparecen como `no reply` pero el destino final se alcanza, suele ser normal (ISP/proveedor censura ICMP).
 2. Si desde cierto salto todo es `no reply` y luego se corta, probablemente ahí estamos tocando un enclave donde no hay retorno ICMP o hay micro-cortes.

4. Cambio de MTU (`pmtu`)

```
pmtu 1492
```

1. Línea extra que aparece cuando el camino detecta un cambio de MTU.
- En nuestro entorno:
 1. Si vemos `pmtu 1492` en un salto, es muy típico de PPPoE (MTU 1500 – 8 bytes de cabecera PPPoE).
 2. Si más adelante vemos `pmtu` menor aún (1480, 1400, etc.), puede haber túneles GRE/IPsec, VPNs o enlaces más restringidos.

5. Último salto y “reached”

```
10: 142.250.179.174 24.789ms
reached
```

1. `10:` → décimo salto en la ruta.
 2. `142.250.179.174` → IP del host destino (por ejemplo, servidor de Google).
 3. `24.789ms` → RTT total hasta el destino.
 4. `reached` → indica que el destino responde correctamente.
- En un uso diario:
 1. Si vemos `reached` pero con RTT muy alto (cientos de ms), buscar enlace congestionado o ruta larga.
 2. Si no vemos `reached`, sino que al final se corta con `no reply` o `TOO LONG`, el destino o algún tramo intermedio no está respondiendo o hay bloqueo de tráfico.

6. Línea de resumen final

```
Resume: pmtu 1492 hops 10 back 10
```

1. `pmtu 1492` → MTU efectivo del camino completo.
 2. `hops 10` → número total de saltos ida hasta el destino.
 3. `back 10` → número de saltos de vuelta (en general igual a `hops` en rutas simétricas).
- Para nosotros como administrador:
 1. Si `hops` es muy alto en una red interna, revisar diseño de ruteo (circuitos, políticas BGP/IGP, etc.).
 2. Si vemos `pmtu` muy bajo en nuestra red interna, investigar si hay túneles innecesarios, PPPoE mal configurado o VLANs que estén “truncando” MTU.

Cómo utilizarlo en nuestro entorno

- Usa `tracepath` para:
 1. Verificar MTU real en la ruta hacia un servidor remoto (por ejemplo, VPN, túnel GRE, o ISP).
 2. Asegurarnos de que no hay cuellos de botella de MTU que provoquen fragmentación o drops silenciosos.
 3. Combinarlo con `ping -M do -s 1472` (Ethernet) para confirmar el tamaño máximo de payload sin fragmentar.

Enlaces

- Material obtenido en parte del artículo escrito por S3L3N1TY para Hacker's Land.
- [¿Qué es el comando Tracert o Traceroute y cómo utilizarlo?](#)
- [Tracert o Traceroute: qué es, cómo funciona o cómo se utiliza](#)
- [Comando tracpath de Linux con Ejemplos Prácticos](#)
- [Tracepath y traceroute \(Linux\)](#)
- [Baúl de Linux: 8.4.2.- Hacer una traza de una ruta](#)

From:
<https://www.euloxio.myds.me/dokuwiki/> - **Euloxio wiki**

Permanent link:
<https://www.euloxio.myds.me/dokuwiki/doku.php/doc:tec:net:cmd:tracpath:inicio?rev=1775726494>

Last update: **2026/04/09 11:21**

