Oracle Linux

Configuring Multipath TCP

G40597-01
September 2025

Oracle Linux Configuring Multipath TCP,

G40597-01

Contents

Preface

Multipath TCP (MPTCP) is an advanced network protocol that lets a single TCP connection use several network paths simultaneously. This can improve network performance, reliability, and resilience. This guide describes how to enable, configure, monitor, and disable Multipath TCP in Oracle Linux.

> ⓘ **Note**
>
> MPTCP is available in Oracle Linux release 10 and later.

1 About Multipath TCP

Multipath TCP (MPTCP) lets a client and server send data over several network interfaces using a single TCP connection. This is useful for systems with different network connections such as wired and wireless links, or servers configured for high availability.

Benefits of MPTCP include:

- Increased connection reliability by using several redundant paths.
- Improved data throughput by using all available network interfaces.
- Automatic network failover so that applications remain connected.

Typical use cases for MPTCP include:

- Multi homed server environments, in which a server is connected to two or more separate networks or network interfaces. MPTCP lets the server communicate over several network paths simultaneously.
- Mobile clients transitioning between Wi-Fi and cellular networks.
- Critical application deployments that benefit from path redundancy.

For more information about MPTCP, see RFC 8684.

2

Enabling and Configuring MPTCP

Multipath TCP is available in Oracle Linux versions based on UEK8 or later kernels.

You can use MPTCP both on-premises and in Oracle Cloud Infrastructure when you control both ends of the TCP connection and have more than one independent connection (for example, Wi-Fi plus cellular, two internet providers, or a server with several network cards). MPTCP can't help if traffic goes through services that end the connection (many load balancers/proxies) or with only a single connection. You need to enable Multipath TCP at both ends of the connection and test it.

Verifying MPTCP Availability

To verify that the system provides MPTCP, run the following command:

```
grep MPTCP /boot/config-$(uname -r)
```

If the system provides MPTCP, the output includes the following:

```
CONFIG_MPTCP=y
CONFIG_INET_MPTCP_DIAG=y
CONFIG_MPTCP_IPV6=y
```

Enabling MPTCP

1. Enable MPTCP for the current session by running the following command:

   ```
   sudo sysctl -w net.mptcp.enabled=1
   ```

2. Persist the change across system reboots.

   To ensure that MPTCP remains enabled after the system reboots, run the following command to add `net.mptcp.enabled=1` to the `/etc/sysctl.conf` file:

   ```
   echo "net.mptcp.enabled=1" | sudo tee -a /etc/sysctl.conf
   ```

3. Reload the `sysctl` configuration to apply all persistent settings.

   ```
   sudo sysctl -p
   ```

4. Verify that MPTCP is enabled:

   ```
   sysctl net.mptcp.enabled
   ```

   The expected output is:

   ```
   net.mptcp.enabled = 1
   ```

Managing Subflow Limits

When using MPTCP on Oracle Linux, administrators control how traffic is distributed across paths by setting MPTCP runtime limits. Subflows are parallel TCP streams within a single MPTCP connection and managing their number helps optimize resources, predictability, and compliance.

By default, MPTCP typically starts with one subflow. Extra subflows are added only when both endpoints can use MPTCP and extra addresses are available. You can manage subflow limits system-wide per network namespace by using the `ip mptcp` command:

- Show current limits:

  ```
  ip mptcp limits show
  ```

- Set the maximum number of subflows (*N*) per connection:

  ```
  sudo ip mptcp limits set subflow N
  ```

- Control how many extra paths (`ADD_ADDR`) a connection accepts from a peer, where *N* is the number of paths:

  ```
  sudo ip mptcp limits set add_addr_accepted N
  ```

- Advertise a specific local address and enable more subflows:

  ```
  sudo ip mptcp endpoint add ip_address signal subflow
  ```

- List which local addresses are advertised to peers:

  ```
  ip mptcp endpoint show
  ```

- Stop advertising an address identified in the output of the `ip mptcp endpoint show` command:

  ```
  sudo ip mptcp endpoint del id address_id
  ```

These settings apply at runtime. To persist across reboots, apply them at startup (for example, by using a `systemd` oneshot service that runs the `ip mptcp limits` commands).

For more information on the `ip mptcp` command, see the `ip-mptcp(8)` manual page.

Configuring a Permanent MPTCP Connection

This task shows how to configure a permanent MPTCP connection between two systems.

Before you begin, ensure the following:

- Oracle Linux has been configured to enable MPTCP.
- Two Oracle Linux systems have more than one interface. For example: `eth0` and `wlan0`.
- The names of network interfaces and their assigned IP addresses have been changed to match the environment.

- A user with administrator privileges has been configured on both systems. For more information, see Oracle Linux 10: Setting Up System Users and Authentication.

This task uses `nmcli` to enable MPTCP with subflows across different interfaces (Ethernet and Wi-Fi), between two Oracle Linux 10 systems (referred to as System A and System B).

The following IP ranges are assigned to System A and System B:

System A:

- Ethernet: `192.0.2.10/24 (eth0)`
- Wi-Fi: `198.51.100.10/24 (wlan0)`

System B:

- Ethernet: `192.0.2.20/24 (eth0)`
- Wi-Fi: `198.51.100.20/24 (wlan0)`

These systems are connected to both networks.

> **Note**
>
> All steps must be completed on both systems.

1. To ensure both interfaces are configured and activated, run the following command to verify that the interfaces are present:

   ```
   sudo nmcli connection show
   ```

   If the `eth0` and `wlan0` connections are missing or down, bring them up with the following commands:

   ```
   sudo nmcli connection up eth0
   ```

   ```
   sudo nmcli connection up wlan0
   ```

2. (Optional) If you're not using DHCP, set static IP addresses for each interface.

   For example, on System A, assign static IP addresses as follows:

   ```
   sudo nmcli connection modify eth0 ipv4.addresses 192.0.2.10/24 ipv4.method
   manual
   ```

   ```
   sudo nmcli connection modify wlan0 ipv4.addresses 198.51.100.10/24
   ipv4.method manual
   ```

On System B:

```
sudo nmcli connection modify eth0 ipv4.addresses 192.0.2.20/24 ipv4.method
manual
```

```
sudo nmcli connection modify wlan0 ipv4.addresses 198.51.100.20/2
ipv4.method manual
```

For more information on assigning static IP addresses in Oracle Linux, see Oracle Linux 10: Setting Up Networking With NetworkManager.

3. Verify the interfaces.

   Run `ip addr show` to confirm that both interfaces are up and have the correct IP addresses:

   ```
   ip addr show eth0
   ```

   ```
   ip addr show wlan
   ```

4. If MPTCP isn't already enabled, enable it:

   ```
   sudo sysctl -w net.mptcp.enabled=1
   ```

5. To increase the subflow limits on both systems, configure MPTCP to use both interfaces:

   ```
   sudo ip mptcp limits set subflow 2
   ```

6. Test the connectivity between the two systems.

   For example, use `curl` to test that a web page served by System B can be accessed by System A:

   ```
   curl http://192.0.2.20/
   <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
   <html>
   <head>
      <title>Directory listing for /</title>
   </head>
   <body>
      <h2>Directory listing for /</h2>
      <hr>
   ...
   ```

   For more information about subflow verification, see the steps described in Testing MPTCP Subflows Using mptcpize.

When you configure network interfaces by using the `nmcli` command, you can save MPTCP-related system settings in the `/etc/sysctl.conf` file so that these changes persist between reboots. For more information, see Enabling and Configuring MPTCP.

3

Testing MPTCP Subflows Using mptcpize

The `mptcpize` utility can be used to confirm that MPTCP is working and using different subflows between hosts. The `mptcpize` utility lets existing network-aware programs use MPTCP by loading a special library before executing the program.

Installing mptcpize

On Oracle Linux release 10, `mptcpize` is provided by the `mptcpd` package. To install `mptcpd`, run the following command:

```
sudo dnf install mptcpd
```

Verify installation by running the following command and reviewing the help:

```
mptcpize --help
```

Using mptcpize to Test MPTCP

This example shows how to test an HTTP connection between two systems (System A and System B).

On System B, start an HTTP server. Use any available server, for example, using Python:

```
python3 -m http.server 80
```

The output is similar to the following:

```
Serving HTTP on 0.0.0.0 port 80
```

On System A, use `curl` to connect to System B using `mptcpize`. For example:

```
mptcpize run curl http://192.0.2.20/
```

The response is an HTML page containing a directory listing:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Directory listing for /</title>
</head>
<body>
<h2>Directory listing for /</h2>
<hr>
...
```

Note

You can use `mptcpize` with many other network tools, such as `wget` and `iperf3`.

Checking Subflow Usage

After running the test connection, you can verify that different subflows are active by running the following command on either host and receiving results similar to those shown:

```
ss --mptcp
```

In the following command output, the field `mptcp_subflows:2` indicates that two subflows are established:

```
State      Recv-Q Send-Q        Local Address:Port       Peer Address:Port
ESTAB      0      0             192.0.2.10:50412         192.0.2.20:80
users:(("curl",pid=20748,fd=3)) mptcp_subflows:2
```

4

Monitoring and Troubleshooting MPTCP

After enabling and configuring MPTCP, use the following checks to monitor active multipath connections and troubleshoot any issues.

Checking if MPTCP is Enabled

To check if MPTCP is enabled, run the following command:

```
sysctl net.mptcp.enabled
```

If MPTCP is enabled, the output is as follows:

```
net.mptcp.enabled = 1
```

Viewing Active MPTCP Connections

Run the following command to display all active MPTCP connections.

```
ss --mptcp
```

The output is similar to the following:

```
State      Recv-Q Send-Q        Local Address:Port       Peer Address:Port
ESTAB      0      0             192.0.2.10:50412         192.0.2.20:80
users:(("curl",pid=20748,fd=3)) mptcp_subflows:2
```

Checking Interface Configuration

List all network devices:

```
ip link show
```

The output is similar to the following:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
```

Review MPTCP-capable addresses:

```
ip addr show eth0
```

```
ip addr show wlan0
```

Use `nmcli` to check and manage connections:

```
nmcli device status
```

The output is similar to the following:

```
DEVICE   TYPE      STATE      CONNECTION
eth0     ethernet  connected  eth0
wlan0    wifi      connected  wlan0
lo       loopback  unmanaged  --
```

> ✓ **Tip**
>
> You can also configure interfaces using `nmtui` (the text-based NetworkManager UI):
>
> 1. Run `nmtui`.
> 2. Select **Edit a connection**.
> 3. Edit the required connection and ensure it's active and up.

Checking Kernel Logs

Check kernel messages for MPTCP events or errors:

```
dmesg | grep -i mptcp
```

The output is similar to the following

```
[13948.311111] mptcp: subflow established
[13948.311222] mptcp: additional subflow established
```

Or, review the system journal:

```
sudo journalctl -k | grep -i mptcp
```

The output is similar to the following:

```
date and time host kernel: mptcp: subflow established
date and time host kernel: mptcp: additional subflow established
```

5

Disabling MPTCP

To disable MPTCP:

1. Disable MPTCP at runtime, by running the following command:

   ```
   sudo sysctl -w net.mptcp.enabled=0
   ```

   The output shows that MPTCP is disabled:

   ```
   net.mptcp.enabled = 0
   ```

2. (Optional) If you added `net.mptcp.enabled=1` to `/etc/sysctl.conf`, remove or comment out the line.
3. Reload the `sysctl` settings, by running the following command:

   ```
   sudo sysctl --system
   ```

4. Confirm that MPTCP remains disabled, by running the following command:

   ```
   sysctl net.mptcp.enabled
   ```

   If MPTCP has been successfully disabled, the command returns the following:

   ```
   net.mptcp.enabled = 0
   ```