

POLITECNICO DI TORINO



**Politecnico  
di Torino**

# **Linux and Mininet command reference**

*“Computer network design and control”* module of  
Communication and network systems

Academic Year 2025/26

Andrea Bianco, Paolo Giaccone,  
Alessandro Cornacchia, Matteo Sacchetto

Version: October 28, 2025

© 2025-26

# Contents

Commands	2
Notation	2
Mininet commands	2
Overview	2
General	2
nodes	2
links	3
net	3
help	3
xterm	3
Network	3
ping/pingall	3
iperf [NOT RECOMMENDED - See iperf3 instead]	3
Linux commands	4
Overview	4
General	4
ls	4
cd	5
cat	5
gedit	5
man	5
stress-ng	5
Network	6
ifconfig	6
ping	6
iperf3	6
ovs-ofctl	7
add-flow	7
dump-flows	8
del-flows	8
Usage in mininet	8
tshark	8

# Commands

This document aims to be a reference guide for the commands you will use in the **Labs on SDN**.

## Notation

The following notation conventions are used throughout this document:

- [ ... ]: squared brackets are used to specify **optional** parameters
- < ... >: angular brackets are used to specify **required** parameters

## Mininet commands

The following commands are available directly from the **Mininet CLI**. All commands must be executed within the Mininet CLI.

```
mininet> net
```

Commands can also be executed on a specific node by prefixing the command with the node name.

```
mininet> h1 ping h2
```

## Overview

### General

- `nodes`: display all nodes in the current network topology (hosts, switches, etc.)
- `links`: display the connections between nodes
- `net`: display a general overview of the network topology (combines `nodes` and `links`)
- `help`: display help documentation for a given command
- `xterm`: open a terminal on one or more nodes

### Network

- `ping`: test connectivity between hosts
- `iperf`: measure network bandwidth

## General

### `nodes`

Displays network nodes (hosts, switches, etc.)

```
mininet> nodes
```

## links

Displays the connections between nodes

```
mininet> links
```

## net

Displays the network topology (nodes and links)

```
mininet> net
```

## help

Displays documentation related to Mininet and its commands.

```
mininet> help [command]
```

If [command] is omitted, it will show the general Mininet help, listing all available commands with a basic overview of them. Otherwise, it will display the help documentation for the specified command.

## xterm

Allows you to open a terminal on the given node/nodes

```
mininet> xterm [nodes...]
```

## Network

### ping/pingall

Allows you to ping a series of hosts

```
mininet> ping <hosts...>
```

or all of them

```
mininet> pingall
```

For further details, refer to the [ping](#) command documentation in the Linux section.

### iperf [NOT RECOMMENDED - See [iperf3](#) instead]

Allows you to test the bandwidth between a pair of hosts

```
mininet> iperf <host1> <host2>
```

While Mininet provides `iperf` as a built-in command, we do not recommend using it due to known bugs. To test bandwidth between hosts, open one `xterm` terminal for each host and manually run [iperf3](#).

# Linux commands

The following commands are available in any **Linux shell**. They can be executed directly from a Linux terminal, or from within the Mininet CLI by prefixing each command with **sh**.

```
mininet> sh man iperf3
```

## Overview

### General

- `ls`: list directory contents
- `cd`: change directory
- `cat`: print file contents
- `gedit`: open the text editor
- `sudo`: run a command as administrator
- `man`: display the manual for a command (press `q` to exit)

### Network

- `ifconfig`: display network interface configuration (see also the `ip` command)
- `ping`: test connectivity to a host
- `iperf3`: measure network bandwidth
- `ovs-ofctl`: configure OpenFlow switches (Open vSwitch OpenFlow Control)
- `tshark`: capture and analyze network traffic

## General

### ls

List directory contents.

```
$ ls [flags...] [files or directories...]
```

### Useful flags

- `-l`: display detailed information (permissions, owner, group, size, etc.)
- `-a`: show hidden files
- `-h`: use human-readable sizes (KB, MB, GB, etc.)
- `-t`: sort by modification time (newest first)
- `-r`: reverse sort order

If no file or directory is specified, it lists the content of the **current directory** (denoted as `.`).

## cd

Change to the specified directory

```
$ cd [dir]
```

If no directory is specified, changes to the current user's home directory (`/home/$USER`, denoted as `~`, e.g., `/home/netlab`). To change to the parent directory, use `..`.

```
$ cd ..
```

In Linux, the directory separator is the forward slash (`/`).

```
$ cd dir/subdir
```

## cat

Print the contents of a file.

```
$ cat <file>
```

## gedit

Open the text editor, optionally editing a specified file.

```
$ gedit [file]
```

## man

Display the manual for a command.

```
$ man <cmd>
```

Use the following keys to navigate the manual:

- `↑↓`: navigate up and down
- `/`: search forward for text
- `?`: search backward for text
- `n`: go to next search result
- `N`: go to previous search result
- `q`: exit the manual
- `h`: open help

## stress-ng

Stress test system components (CPU, RAM, disk, etc.)

```
$ stress-ng [options...]
```

### Useful flags

- `-c, --cpu <N>`: stress the CPU with N workers (use `--cpu-method <method>` to specify the stress method)

- `--cpu-method <method>`: specify the CPU stress method. By default, all stress methods run sequentially, but a specific one can be selected. Refer to the `stress-ng` manual for available methods
- `--io <N>`: stress disk I/O with N workers
- `-t`, `--timeout <T>`: run the stress test for T seconds (runs until CTRL+C if not specified)
- `-m`, `--vm <N>`: stress RAM with N workers

## Network

### ifconfig

Display network interface configuration.

```
$ ifconfig [interface]
```

Without parameters, it displays all network interfaces and their configurations. To view details for a specific interface, specify its name.

**NOTE:** `ifconfig` is part of the deprecated `net-tools` package. The modern `ip` command is preferred in current Linux systems.

To display the same information using the `ip` command:

```
$ ip -s addr
```

### ping

Send ICMP echo requests to a network device.

```
$ ping [flags...] <host>
```

#### Useful flags

- `-i <seconds>`: set interval between packets in seconds (values below 0.2 require `sudo`)
- `-c <count>`: send a specified number of packets
- `-D`: print timestamps
- `-s <bytes>`: set packet size in bytes
- `-t <ttl>`: set TTL (Time To Live)

### iperf3

Measure network bandwidth between hosts. This is an improved version of `iperf`. It uses a client-server architecture, requiring one host as server and one as client.

#### Server

```
$ iperf3 -s
```

## Client

```
$ iperf3 -c [flags...] <host>
```

## Useful flags

- -u: use UDP instead of TCP
- -b <bandwidth [kmgKMG | pps]>: target bandwidth (bits/s, K/M/G multiples, or pps (packets per second))
- -l <size[KM]>: packet size in bytes (K/M multiples supported)
- -t <duration>: test duration in seconds

## ovs-ofctl

OpenFlow control utility for Open vSwitch (Open Virtual Switch). Configure OpenFlow switches created by Mininet.

```
$ ovs-ofctl [sub-command] [options...]
```

The sub-commands used in the labs are the following:

- add-flow: add a new flow rule to the switch
- dump-flows: display current flow rules
- del-flows: delete flow rules from the switch

## add-flow

Configure the switch to perform an action on traffic matching specified criteria.

```
$ ovs-ofctl add-flow <switch> <matcher>, <action>
```

## Useful matchers:

- in\_port=<switch\_port\_number>: (**port based matcher**) match traffic received on a specific switch port
- dl\_type=<type>, nw\_dest=<ip/netmask>: (**destination IP based matcher**) match traffic by destination IP and data link type
  - Standard <type> values:
    - \* 0x0800: IPv4
    - \* 0x86dd: IPv6
  - Note: Only IPv4 is used in these labs
- dl\_type=0x0800, nw\_proto=<proto>, nw\_dest=<ip/netmask>: (**destination IP + protocol based matcher**) match traffic by destination IP and protocol
  - Standard <proto> values:
    - \* 1: ICMP
    - \* 6: TCP
    - \* 17: UDP

- `<shorthand>, nw_dest=<ip/netmask>`: shorthand notation combining type and protocol

– Available shorthands:

- \* `ip`: equivalent to `dl_type=0x0800`
- \* `icmp`: equivalent to `dl_type=0x0800, nw_proto=1`
- \* `tcp`: equivalent to `dl_type=0x0800, nw_proto=6`
- \* `udp`: equivalent to `dl_type=0x0800, nw_proto=17`

## Action

Actions typically route traffic through a specific switch port using the syntax `actions=output:<port>`, where `<port>` is the port number.

## dump-flows

Display the flow rules configured on a switch. The switch name is assigned by Mininet (e.g., `s1`).

```
$ ovs-ofctl dump-flows <switch>
```

## del-flows

Delete flow rules based on a matcher. Available matchers are described in the [add-flow](#) section. If no matcher is specified, it deletes all rules on the switch.

```
$ ovs-ofctl del-flows <switch> [matcher]
```

## Usage in mininet

It is recommended to use the above command from within the Mininet CLI by prefixing it with `sh`.

```
mininet> sh ovs-ofctl dump-flows s1
```

## tshark

Capture and analyze network traffic.

```
$ tshark [flags..]
```

## Useful flags

- `-i <interface>`: select capture interface (e.g., `s1-eth1`)
  - Use [ifconfig](#) to list available interface names
- `-q`: suppress continuous packet count during capture (display final count only)
- `-z <options>`: collect and display statistics
  - `ip_srcdst,tree`: calculate IPv4 statistics with source and destination addresses grouped together
  - `conv, <type>`: display conversation table for the capture
    - \* `eth`: Ethernet addresses

- \* `ip`: IP addresses
- \* `tcp`: TCP/IP socket pairs (IPv4 and IPv6)
- \* `udp`: UDP/IP socket pairs (IPv4 and IPv6)