Linux: Network Commands in Linux

Below you will find an overview of the most important network commands with descriptions of their use and useful parameters.

—

ping

Checking host availability - Sends ICMP Echo Request packets to the target host and waits for a response. It is used to test the network connection and measure response time.

Useful parameters:

- `-c number` - number of packets to be sent (Linux, macOS)
- `-W timeout` - response timeout in milliseconds
- `-i interval` - packet sending interval in seconds
- `-s size` - size of the data packet (default 56 bytes)

Example: `ping -c 4 -i 0.5 8.8.8.8`

—

ifconfig

Display of network interfaces configuration - shows details of all active network interfaces (IP, mask, MAC, statistics). On newer systems replaced by `ip`.

Useful parameters:

- `ifconfig interface_name` - information about a specific interface
- `ifconfig -a` - displays all interfaces (active and inactive)
- `ifconfig interface_name up/down` - enable/disable interface
- `ifconfig interface_name IP netmask MASK` - setting the IP address

Example: `ifconfig eth0 192.168.1.100 netmask 255.255.255.0`

—

ip a (ip addr)

Modern IP configuration display - replacement `ifconfig` offering more detailed information

about IP addresses, routers and interfaces. Part of the iproute2 package.

Useful parameters:

- `ip address show interface_name` - information about a specific interface
- `ip addr add IP/MASKA dev name_interface` - add IP address
- `ip addr del IP/MASKA dev interface_name` - remove IP address
- `ip link set interface_name up/down` - enable/disable interface

Example: `ip addr add 192.168.1.50/24 dev eth0`

—

route

Viewing and managing the routing table - shows how packets are routed to different networks. Allows you to add, delete and modify routes.

Useful parameters:

- `route -n` - display the routing table in numerical format
- `route add default gw IP` - set default gateway
- `route add -net netmask MASK gw IP` - add route to network
- `route del -net NETWORK netmask MASK` - route delete
- `route del default gw IP` - delete default gateway

Example: `route add default gw 192.168.1.1`

—

ip route

Modern routing management - A newer replacement for the `route` with more intuitive syntax, part of the iproute2 package.

Useful parameters:

- `ip route show` - show all routes
- `ip route add NETWORK/MASK via IP` - add route
- `ip route del NETWORK/MASK via IP` - deleting route
- `ip route add default via IP dev INTERFACE` - set default gateway
- `ip route get IP` - check which route the packet will go through to the address

Example: `ip route add 192.168.2.0/24 via 192.168.1.254`

—

arp

Viewing and managing the ARP table - maps IP addresses to MAC (physical) addresses on the local network. Arp Resolution Protocol is used to recognise hardware addresses.

Useful parameters:

- `arp -a` - display the entire ARP table
- `arp -n` - display ARP table in numerical format
- `arp -d IP` - removing an entry from the ARP table
- `arp -s IP MAC` - adding a static ARP entry
- `arp -i interface_name` - display ARP only for a specific interface

Example: `arp -a -n`

—

ip neigh

Modern neighbour table management (ARP) - a newer replacement for the `arp` with richer capabilities, part of the iproute2 package.

Useful parameters:

- `ip neigh show` - show neigh table
- `ip neigh add IP lladdr MAC dev INTERFACE` - add
- `ip neigh del IP dev INTERFACE` - deleting entry
- `ip neigh flush all` - flush all table
- `ip neigh show dev name_interface` - show for a specific interface

Example: `ip neigh show dev eth0`

—

traceroute

Tracking the path of packets to the destination - shows each router (hop) on the path from your computer to the destination host, along with the response time. Very useful for diagnosing network problems.

Useful parameters:

- `-m number_hops` - maximum number of hops (default 30)
- `-w timeout` - response timeout in seconds
- `-q number` - number of questions sent to each hop (default 3)
- `-n` - display of IP addresses without resolving DNS names
- `-p port` - destination port (for TCP traceroute)

Example: `traceroute -m 15 -n 8.8.8.8`

—

mtr

Interactive route tracking with continuous monitoring - combination `ping` i `traceroute` displaying real-time statistics. Tool for advanced network diagnostics.

Useful parameters:

- `-c number` - number of packets to be sent (end after sending)
- `-r` - non-interactive report
- `-n` - displaying only IP addresses
- `-w` - increased column width
- `-s size` - packet size

Example: `mtr -c 50 -n 8.8.8.8`

—

netstat

Display of network statistics and active connections - Shows open ports, active connections, protocol statistics and routing. Can monitor network activity in real time.

Useful parameters:

- `-tuln` - display of open TCP/UDP ports in numerical format
- `-a` - all connections (both listening and established)
- `-p` - display the process (PID) responsible for the connection
- `-s` - statistics for each protocol
- `-r` - routing table (equivalent to `route -n`)

Example: `netstat -tuln`

—

ss

Modern replacement for netstat - Faster display of network socket information. Part of the iproute2 package, preferred in newer distributions.

Useful parameters:

- `-tuln` - open TCP/UDP ports in numerical format
- `-tap` - all connections with process information
- `-s` - network statistics
- `-i` - interface information
- `-l` - only listening sockets

Example: `ss -tuln`

—

dig

DNS queries for domain records - A tool for advanced DNS queries to check A, AAAA, MX, NS and other records.

Useful parameters:

- `dig domain` - A record request
- `dig domain MX` - email server query
- `dig domain NS` - query for name servers
- `+short` - shortened response format
- `@name-server_DNS` - request for a specific DNS server

Example: `dig example.com +short`

—

nslookup

A simpler version of the DNS query - Domain name recognition and reverse recognition tool (IP per domain). More accessible than `dig`.

Useful parameters:

- `nslookup domain` - domain IP recognition
- `nslookup IP` - reverse recognition (IP to domain)

- `nslookup domain @server_DNS` - query of a specific DNS server
- interactive mode - entering the `nslookup` without arguments

Example: `nslookup google.com 8.8.8.8`

—

host

Recognition of names and IP addresses - A simple tool to check IP/domain mappings. More minimalistic than `dig` or `nslookup`.

Useful parameters:

- `host domain` - IP address recognition
- `host IP` - reverse domain recognition
- `host domain server_DNS` - query specific server
- `-t type` - query for specific record type (A, MX, NS etc.)

Example: `host google.com`

—

curl / wget

HTTP/HTTPS download of web content and files - `curl` displays the response, `wget` downloads files to disk. Tools for testing HTTP connections and downloads.

Useful parameters (curl):

- `-I` - fetch HTTP headers only
- `-X method` - selecting an HTTP method (GET, POST, PUT).
- `-d data` - send POST data
- `-H „header”` - adding a custom header
- `-o file` - writing response to file

Example: `curl -I https://example.com`

—

tcpdump

Network packet capture and analysis - Tool for detailed analysis of network traffic at packet level. Requires administrator rights.

Useful parameters:

- `-i interface` - capture on a specific interface
- `-n` - displaying IP addresses instead of names
- `host IP` - filtering packets of a specific host
- `port PORT` - filtering a specific port
- `-w file.pcap` - save packets to file

Example: `tcpdump -i eth0 -n host 8.8.8.8`

—

Quick reference table

Command	Application	Modern replacement
`ping`	Host availability test	-
`ifconfig`	Interface configuration	`ip a` / `ip link`
`route`	Routing management	`ip route`
`arp`	ARP/MAC table	`ip neigh`
`netstat`	Network statistics	`ss`
`traceroute`	Packet traceroute	`mtr`
`dig/nslookup`	DNS queries	-

—

This tutorial should give you a solid foundation for working with networking in Linux. Remember that most of these commands require Internet or local network access, and some (like `tcpdump` or configuration modification) require the privileges of `sudo`.