Command line tools for networking

```
Module Code: ELEE1157

Module Name: Network Routing Management

Credits: 15

Module Leader: Seb Blair BEng(H) PGCAP MIET MIHEEM FHEA
```

Download as a PDF

1. Network Configuration and Interface Management

- `ip`: Configure IP addresses, routes, and manage network interfaces.
- `ifconfig`: View and configure network interfaces (deprecated on some systems, replaced by ip).
- `netsh`: Configure and manage network settings on Windows.
- `nmcli`: Control NetworkManager and configure network connections on Linux.
- `iwd/wpa/iw/connman`: Network management tools
- `route`: Manage IP routing tables.

2. Connection Testing and Diagnostics

- `ping`: Test reachability of hosts and measure round-trip time.
- `traceroute` / `tracert` (Windows): Trace the path packets take to a destination.
- `mtr`: Combines ping and traceroute for continuous network diagnostics.
- `telnet`: Test connectivity and basic communication with TCP ports.
- `nc` (Netcat): Send and receive data over TCP/UDP; useful for testing ports.
- `curl` / `wget`: Retrieve data from URLs, test HTTP/HTTPS connections.

3. Network Analysis and Troubleshooting

- `arp`: Display or manipulate the ARP cache (used to map IPs to MAC addresses).
- `tcpdump`: Capture and analyze packets on a network interface.
- `Wireshark` (CLI: `tshark`): Network protocol analyzer for in-depth packet analysis.
- `ss`: Display socket statistics and details for active connections.
- `nmap`: Network discovery and security auditing, includes port scanning.
- `nslookup` / dig: Query DNS servers for information about hostnames and IPs.
- `host`: Simple tool for DNS lookups.

4. Performance Monitoring and Statistics

- `netstat`: View network connections, routing tables, interface stats, and more.
- `iftop`: Monitor bandwidth usage on a specific interface.
- `nload`: Visualize network traffic in real-time.
- `bmon`: Bandwidth monitor and rate estimator.
- `iperf3`: Measure network bandwidth between two hosts.
- `vnstat`: Network traffic monitor and logger.

5. Network File Transfer and Communication

- `scp`: Securely copy files between hosts over SSH.
- `sftp`: Secure File Transfer Protocol, similar to ftp but encrypted with SSH.
- `rsync`: Sync files and directories locally or across networks efficiently.
- `ftp`: Transfer files using the File Transfer Protocol (less secure than SFTP).
- `tftp`: Transfer files over Trivial File Transfer Protocol (often used in PXE environments).

6. Firewall and Security Management

- `ufw`: Simple command-line interface for managing firewall on Linux.
- `iptables` / nftables: Configure firewall rules and manage packet filtering.
- `firewalld`: A service to manage firewall on Linux, often used with firewall-cmd.

7. VPN and Tunnel Management

- `openvpn`: Connect to OpenVPN-compatible VPNs.
- `ssh`: Secure Shell for encrypted connections and tunneling.
- `sshd`: SSH daemon, runs on servers to allow SSH access.
- `stunnel`: Provides TLS encryption for arbitrary TCP connections.
- `ipsec` / `strongSwan`: Manage IPsec VPN connections.

8. nmcli

`nmcli` is the command-line tool for managing network connections with NetworkManager. It can handle both wired and wireless connections.

```
# List all connections
nmcli connection show

# Connect to a Wi-Fi network
nmcli device wifi connect "SSID" password "password"

# Disconnect a connection
nmcli connection down id "ConnectionName"

# Display device status
nmcli device status
```

9. iwd (Internet Wireless Daemon)

`iwd` is a lightweight Wi-Fi management daemon developed by Intel, offering WPA2 and WPA3 support.

```
# Start interactive mode to manage Wi-Fi connections
iwctl

# Inside iwctl:
# List available Wi-Fi networks
> station wlan0 get-networks

# Connect to a Wi-Fi network
> station wlan0 connect "SSID"

# Disconnect from a network
> station wlan0 disconnect
```

10. wpa_supplicant

`wpa_supplicant` is a Wi-Fi management daemon often used to connect to WPA and WPA2 protected networks.

```
# Start wpa_supplicant with a configuration file
wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf

# Connect interactively using wpa_cli
wpa_cli -i wlan0

# Within wpa_cli:
# Connect to a network by SSID and passphrase
> add_network
> set_network 0 ssid ""SSID""
> set_network 0 psk ""password""
> enable_network 0
```