Networking Essentials

Session 11

Objectives:

- Understand key networking tools in Linux
- Test connectivity using common commands
- Transfer files over the network
- Use SSH to access remote systems
- Review IP addressing concepts (IPv4/IPv6)
- Gain basic troubleshooting skills
- Understand DHCP and manual configuration tools

Networking

Why Networking?

- Enables communication: sharing files, accessing the internet, using cloud resources
- Essential for remote work and system administration
- Underpins security: firewalls, open ports, etc.

Almost everything today depends on network connections—whether you're sending emails, coding a web app, or logging into a server on the other side of the world. Even for data science, remote data storage and compute clusters require networking.

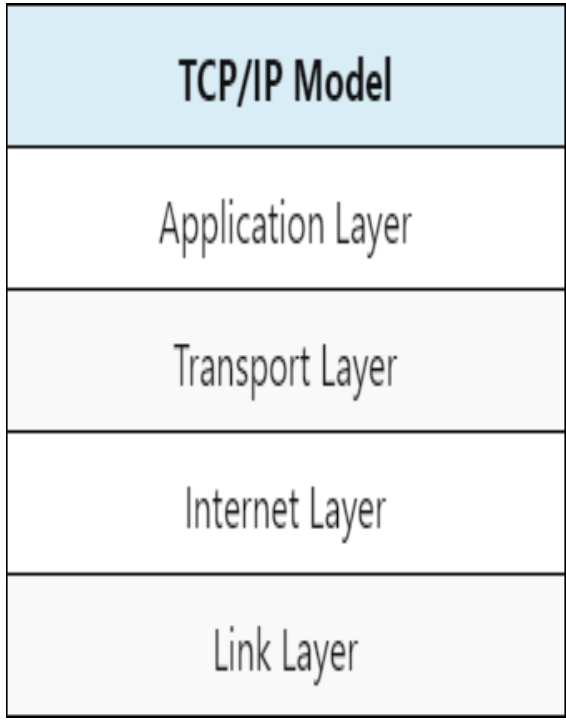
TCP/IP Overview

- TCP/IP is the foundational protocol suite for modern networking.
- It ensures end-to-end communication, routing, and application-level services.
- It is structured into four abstraction layers:

TCP/IP Overview

- TCP/IP is the foundational protocol suite for modern networking.
- It ensures end-to-end communication, routing, and application-level services.
- It is structured into four abstraction layers:

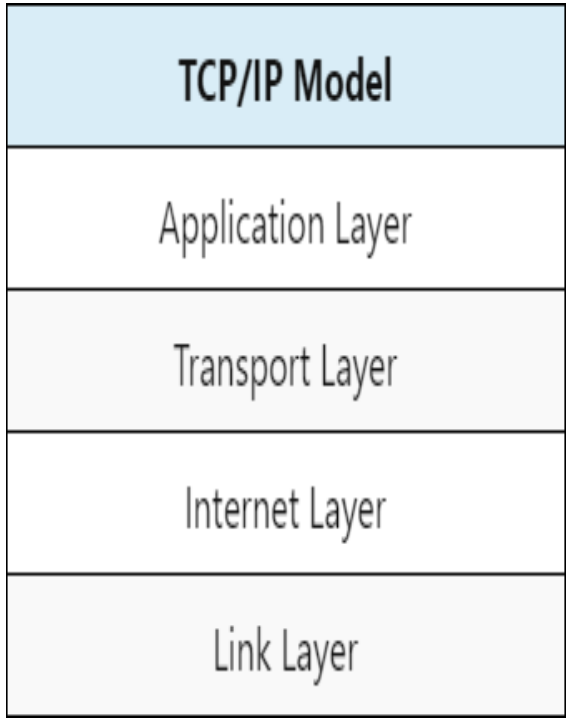
  - Application Layer:
    - Provides services to applications like web browsers, email clients, terminals.
    - Examples of protocols: HTTP, HTTPS, FTP, SSH, DNS, SMTP.
    - **is what you interact with: web browsers, SSH clients, email, etc**



TCP/IP Overview

- TCP/IP is the foundational protocol suite for modern networking.
- It ensures end-to-end communication, routing, and application-level services.
- It is structured into four abstraction layers:

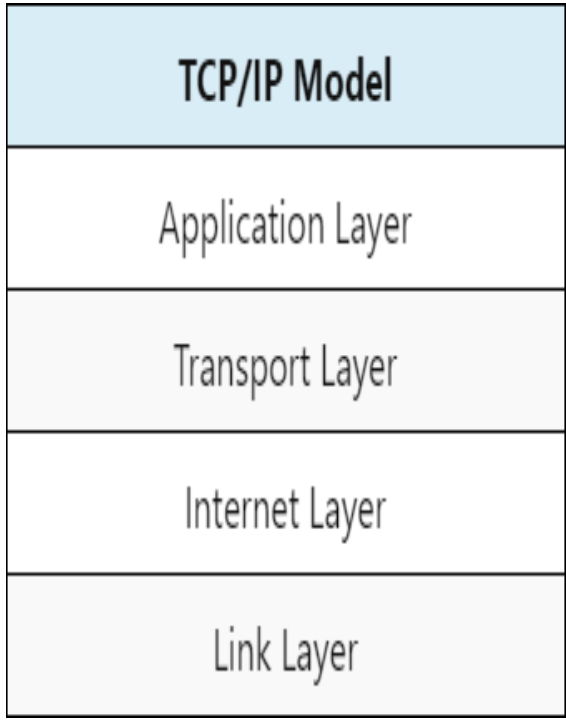
  - Transport Layer:
    - Ensures communication between host systems.
    - TCP: reliable, connection-oriented; ensures ordered delivery.
    - UDP: lightweight, connectionless; no delivery guarantees.
    - **delivers your data: either reliably (TCP) or fast but with less control (UDP).**



TCP/IP Overview

- TCP/IP is the foundational protocol suite for modern networking.
- It ensures end-to-end communication, routing, and application-level services.
- It is structured into four abstraction layers:

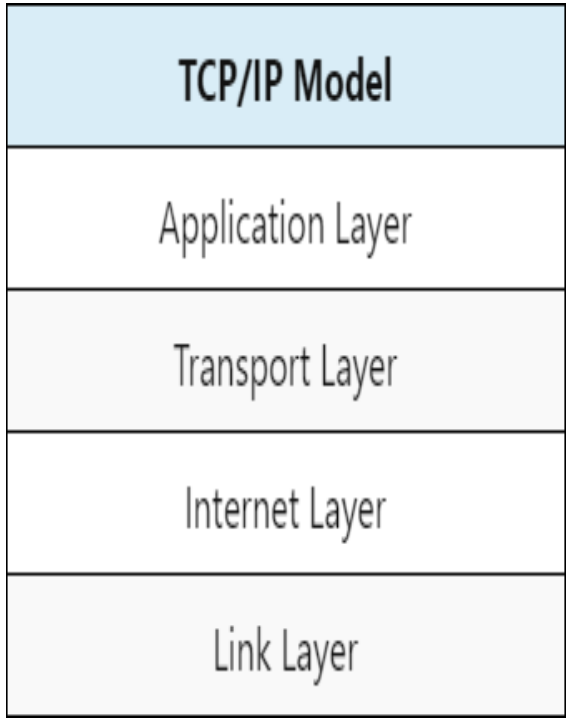
  - Internet Layer:
    - Responsible for logical addressing and routing across multiple networks.
    - The IP protocol routes packets from source to destination.
    - Also includes ICMP (used for diagnostics like ping, traceroute).
    - Packets are identified and routed by source/destination IP addresses.
    - **finds the route to your destination with IP addresses.**



TCP/IP Overview

- TCP/IP is the foundational protocol suite for modern networking.
- It ensures end-to-end communication, routing, and application-level services.
- It is structured into four abstraction layers:

  - Link Layer:
    - Handles communication with the physical network.
    - Defines how data is encoded for a specific type of link (e.g., Ethernet, Wi-Fi).
    - Responsible for MAC addresses, frame transmission, and error checking.
    - This layered model allows interoperability, modularity, and reliable communication over diverse networks.
    - **moves the data through cables or WiFi, using things like MAC addresses.**



Think of it as the postal system:
•The letter (data) goes in an envelope (TCP/UDP),
•gets an address (IP),
•and travels via different roads and delivery people (Ethernet/WiFi).

Real-World Protocol Examples

Layer	Example Protocols	Example Use
Application	HTTP, SSH, FTP, DNS, SMTP	Web, remote login
Transport	TCP (reliable), UDP (fast)	File transfer, VoIP
Internet	IP, ICMP	Routing, ping
Link	Ethernet, WiFi	LAN, wireless

Browsing the web uses HTTP at the Application layer, which uses TCP to ensure your page loads correctly. DNS lets you use domain names instead of remembering IP addresses. SSH allows secure remote login.

What Is a Network?

- A network is a collection of connected devices that share resources and data.
- Networks allow communication between computers and services, locally or globally.
- Two major types:
  - LAN (Local Area Network) – e.g., your home Wi-Fi
  - WAN (Wide Area Network) – e.g., the Internet
- Requires protocols to establish rules (e.g., TCP/IP)

A network is just a collection of devices connected together. At home, that's your phone, laptop, and maybe a smart TV—using WiFi (a LAN). The Internet connects billions of devices together into a WAN. But for these devices to "talk," they have to agree on the language and rules. That's the job of protocols like TCP/IP.

What is a IP adress?

- An IP address is a unique identifier for a device on a network.
- IPv4: 32 bits, written in dotted decimal (e.g., 192.168.1.1)
- IPv6: 128 bits, written in hexadecimal (e.g., 2001:0db8:85a3::8a2e:0370:7334)

IPv4 is what we still mostly use today. It's easier to read, but its address space is running out. IPv6 is the modern replacement, already widely used in smartphones, servers

- IP addresses can be statically assigned (manually configured by an admin), or dynamically assigned using a protocol like DHCP.
  - Static IP: **Useful for servers, printers, or devices requiring predictable addresses**.
  - Dynamic IP (via DHCP): Automatically assigned on demand; suitable for most client machines. For example, your laptop gets a different IP when you reconnect to WiFi.

- The IP address includes:
  - Network ID (defines which network)
  - Host ID (defines which device on the network)

Having a look at your IP

- In a terminal type ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
    valid_lft forever preferred_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:bf:7b:b6 brd ff:ff:ff:ff:ff:ff
inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
    valid_lft 85436sec preferred_lft 85436sec
inet6 fd17:625c:f037:2:678f:aae7:5d8c:bcdd/64 scope global temporary dynamic
    valid_lft 86285sec preferred_lft 14285sec
inet6 fd17:625c:f037:2:a00:27ff:febf:7bb6/64 scope global dynamic mngtmpaddr
    valid_lft 86285sec preferred_lft 14285sec
inet6 fe80::a00:27ff:febf:7bb6/64 scope link
    valid_lft forever preferred_lft forever

Having a look at your IP

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
    valid_lft forever preferred_lft forever

- The lo interface (loopback) has IPv4 127.0.0.1, **Used for internal communication within the same machine**
- /8: means that the first 8 bits are the network part, covering all addresses from 127.0.0.0 to 127.255.255.255.

- Inet6::1/128 is the loopback IPv6 address, similar to 127.0.0.1 in IPv4

- **IPv4 address: Always 127.0.0.1 (or in the range 127.0.0.0/8).**

- Typical use: When software on your machine wants to talk to itself (e.g., a web server tested locally with localhost).

Having a look at your IP

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:bf:7b:b6 brd ff:ff:ff:ff:ff:ff
inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
    valid_lft 85436sec preferred_lft 85436sec
inet6 fd17:625c:f037:2:678f:aae7:5d8c:bcdd/64 scope global temporary dynamic
    valid_lft 86285sec preferred_lft 14285sec
inet6 fd17:625c:f037:2:a00:27ff:febf:7bb6/64 scope global dynamic mngtmpaddr
    valid_lft 86285sec preferred_lft 14285sec
inet6 fe80::a00:27ff:febf:7bb6/64 scope link
    valid_lft forever preferred_lft forever

- 2: enp0s3: — This is the second listed interface, named enp0s3. **Connects your computer to a physical network (LAN, Internet, etc.)**
- link/ether 08:00:27:bf:7b:b6 — This is the MAC address of the interface (a unique identifier assigned to a network interface card)
- inet 10.0.2.15/24 — IPv4 address assigned via DHCP.
- scope global — Global IPv4 address assigned dynamically (DHCP)
- valid_lft, preferred_lft — DHCP lease durations: time left before renewal required.
- inet6 fd17:... scope global temporary dynamic — A temporary IPv6 address for privacy, assigned dynamically.
- inet6 fd17:... scope global dynamic mngtmpaddr — A stable, managed global IPv6 address.
- inet6 fe80::... scope link — Link-local IPv6 address, only reachable on the local network

Public vs Private IP Addresses

- **Private IP:**
  - used inside LANs
  - not accessible directly from the internet
  - IPv4: 192.168.x.x, 10.x.x.x, 172.16.x.x – 172.31.x.x

- **Public IP**:
  - Routable
  - assigned by ISP
  - visible online NAT (Network Address Translation)

Type:

- ip a → show Private IP
- curl ifconfig.me → show your public IP visible by website

curl stands for **Client URL:**

- used to transfer data from or to a server using supported protocols
- curl ifconfig.me sends an HTTP request to a remote web service that responds with your public IP address.

DNS

What is DNS?

DNS

What is DNS?

- DNS stands for Domain Name System.
- It is the Internet’s "phonebook" — it translates human-readable names like google.com into machine-readable IP addresses like 142.250.184.110.
- Without DNS, you would have to remember and type IP addresses instead of domain names.
- Your system queries a DNS to resolve names.(e.g. google: 8.8.8.8)

Try
ping google.com
ping 8.8.8.8

How IPs Are Assigned – DHCP

- **DHCP = Dynamic Host Configuration Protocol**

- Automatically assigns IP address, subnet mask, gateway, and DNS settings to clients.
- **NO need for manual IP configuration.**
- Works in four steps:
  - DHCPDISCOVER – client broadcasts to find DHCP server
  - DHCPOFFER – server responds with available IP and settings
  - DHCPREQUEST – client accepts the offer
  - DHCPACK – server confirms the lease

- Common scenarios: connecting a laptop to Wi-Fi, booting a VM on NAT
- Indicators of DHCP usage:
  - “ip a” shows scope global dynamic
  - valid_lft, preferred_lft timers on IP lease

DHCP: Automatic IP Assignment

On modern Linux systems, there are two main ways to manage network interfaces:

1. **Netplan (For static config on servers):**
   - Declarative YAML files (e.g., /etc/netplan/ 01-network-manager-all.yaml) Example:

     network: version: 2
     ethernets: enp0s3:
     dhcp4: true
   - **Apply changes with**: sudo netplan apply

2. **NetworkManager GUI tool or CLI via nmcli, nmtui**
   - Common on Ubuntu Desktop, Fedora, etc.
   - Configuration is managed internally or via /etc/NetworkManager
   - Interfaces will not appear in netplan YAML if managed by NetworkManager only.

- Search and open the YAML files:

  network: version: 2
  renderer: NetworkManager

Using nmcli

nmcli is the command-line tool for NetworkManager

Allows you to view or configure network settings without a GUI

- **Commands**
  - nmcli device show # Show interface details (IP, DNS, etc.)
  - nmcli con show # List all saved connections
  - nmcli device status # Connection state of all devices
  - nmcli con up id <name> # Activate a connection
  - nmcli con edit # Interactive conf

Ex: nmcli device show enp0s3

Expected output:

•Interface name and MAC address
•IP addresses (IPv4/IPv6)
•DNS servers
•DHCP status

Ports

- A port is a logical **communication endpoint on a device**.
- Think of an IP address as a street address, and a port as the apartment number.
- **Ports let one machine run multiple services but only one process can be bound to a port at a time.**

Port numbers range from 0 to 65535:

- 0–1023: Well-known ports (require root privileges to use)
- 1024–49151: Registered/user ports
- 49152–65535: Dynamic/private ports

Common ports:
22 → SSH
80 → HTTP
443 → HTTPS
53 → DNS
25 → SMTP (email)

Tools to list open ports:

- ss -tuln
- lsof -i → see which processes use which ports

Basic Troubleshooting
Steps if networking fails

Is the interface up?

- Run: ip a
- Look for state UP on enp0s3 or similar
- If DOWN: sudo ip link set enp0s3 up

Do you have an IP address?

- Check: ip a
- If no inet line → request IP with: sudo dhclient -v

Can you reach your gateway?

- View default route: ip r
- Test: ping 192.168.1.1 (or whatever your gateway is)

Can you resolve domain names? (DNS)

- Try: ping google.com
- If that fails but ping to 8.8.8.8 works → DNS problem
- Check: cat /etc/resolv.conf

Use traceroute to find the failure point

- traceroute google.com or tracepath google.com

Are services listening?

- ss -tuln → shows open ports
- lsof -i → shows which apps use the ports

Exemple Why can’t I SSH into a server?:

Is the interface up?

- Run: ip a
- What to check:
  - Is your main interface (enp0s3, eth0, wlan0, etc.) UP and does it have an IP address?
    - If not:
      - You may not be connected to the network. Check cables, or enable WiFi.

Can You Reach the Server?

- ping <server_ip>
- What to check:
  - Do you get replies, or does it time out?
    - If ping fails:
    - The server may be down, the IP may be wrong, or there’s a network problem (firewall, routing)

Is DNS Working? If you use a hostname instead of an IP address, check if DNS is working.

- ping <server_hostname>
- What to check:
  - If ping <IP> works, but ping <hostname> fails, you have a DNS problem.

Exemple Why can’t I SSH into a server?:

Is SSH Service Running on the Server?

- sudo systemctl status ss
- ss -tuln | grep :22
- What to check:
  - Is sshd running? Is port 22 open?
    - If not:
      - Start the SSH service: sudo systemctl start ssh

Is a Firewall Blocking SSH??

- sudo ufw status
- sudo firewall-cmd --list-all
- What to check:
  - Is port 22 allowed?

Are You Using the Right Port?

- ssh -p <port> user@server

Check Your Credentials

- Are you using the correct username and password or SSH key?

Exemple Why can’t I SSH into a server?:

Step	What to Check	Command(s)
Network Connection	Interface up & has IP	ip a
Server Reachable	Can ping server IP	ping <server_ip>
DNS Works	Hostname resolves	ping <hostname>
SSH Running	Service up, port 22 open	systemctl status ssh`ss -tuln
Firewall	Port 22 allowed	ufw status or firewall-cmd
Right Port	Correct SSH port	ssh -p <port> …
Credentials	Correct user & key permissions	Check username, chmod 600
SSH Verbose	See debug output	ssh -v …