# 🧠 DevOps-Focused Computer Networking Notes

## 🔁 1. OSI Model – The Blueprint of Networking

The **OSI (Open Systems Interconnection) Model** is a **conceptual framework** used to understand network interactions in 7 layers.

mathematica

CopyEdit

Application – Layer 7

Presentation – Layer 6

Session – Layer 5

Transport – Layer 4

Network – Layer 3

Data Link – Layer 2

Physical – Layer 1

💡 Mnemonic: **All People Seem To Need Data Processing**

## 📚 2. All Layers in Detail

### 🟣 Layer 7 – Application

* **Closest to user**
* Interfaces with apps (like browsers, curl, Postman)
* Examples:
  + HTTP, HTTPS
  + DNS
  + SMTP (email)
  + FTP, SSH

### 🔵 Layer 6 – Presentation

* **Data translator** for application and network
* Handles:
  + **Encryption/Decryption** (TLS, SSL)
  + **Compression** (ZIP, MPEG)
  + **Encoding/Decoding** (ASCII, JPEG)

Example: TLS encrypts data before it goes to Transport Layer.

### 🟢 Layer 5 – Session

* **Establishes, manages, and terminates sessions**
* Manages:
  + Dialog control (who sends when)
  + Synchronization
* Protocols: NetBIOS, RPC

In SSH, session layer ensures you can keep multiple terminals open to one server.

### 🟡 Layer 4 – Transport

* **End-to-end delivery of data**
* Key Concepts:
  + **TCP** (reliable, ordered, error-checked)
  + **UDP** (faster, connectionless)
  + Ports (80, 443, 22, etc.)
  + Segmentation, flow control, error control

DevOps Relevance: Understanding TCP vs UDP helps in debugging Kubernetes/Container/Cloud networking.

### 🟠 Layer 3 – Network

* **Routes packets**
* Adds IP addresses (source + destination)
* Protocols:
  + IP (IPv4, IPv6)
  + ICMP (ping)
  + ARP (IP → MAC mapping)
  + OSPF, BGP (routing protocols)

DevOps Use: You’ll work with IPs, CIDRs, Subnets, DNS all the time.

### 🔴 Layer 2 – Data Link

* **Frame creation and MAC addressing**
* Handles error detection with checksums
* Switches operate here
* Protocols:
  + Ethernet
  + PPP
  + VLAN tagging (802.1Q)

### ⚫ Layer 1 – Physical

* **Actual physical media**: cables, fiber, NICs
* Deals with:
  + Voltage, bitstream, modulation
  + Connectors (RJ45), topology

Important when troubleshooting server racks, cable setups in on-prem DevOps.

## 🔌 3. Ports and Protocols – Ultimate Cheat Sheet

### 🔥 Common Ports (You MUST Know)

| Protocol | Port | Purpose |
| --- | --- | --- |
| HTTP | 80 | Web (non-secure) |
| HTTPS | 443 | Secure web (TLS) |
| SSH | 22 | Secure shell (remote login) |
| FTP | 20/21 | File transfer |
| DNS | 53 | Domain resolution |
| SMTP | 25 | Email sending |
| POP3 | 110 | Email retrieval |
| IMAP | 143 | Email sync |
| Telnet | 23 | Remote login (insecure) |
| SNMP | 161 | Monitoring devices |
| RDP | 3389 | Remote Desktop |
| MySQL | 3306 | Database |
| PostgreSQL | 5432 | Database |
| Redis | 6379 | Cache DB |
| MongoDB | 27017 | NoSQL DB |

### 🧩 TCP vs UDP Comparison

| Feature | TCP (Transmission Control) | UDP (User Datagram) |
| --- | --- | --- |
| Reliable | ✅ Yes | ❌ No |
| Ordered | ✅ Yes | ❌ No |
| Speed | Slower | Faster |
| Use Cases | HTTP, SSH, FTP, SMTP | DNS, VoIP, video/gaming |
| Connection | Stateful | Stateless |

In DevOps, **TCP is used in reliable services like SSH/HTTP**, **UDP for speed-first stuff like DNS/Kubernetes health checks**.

### 🌐 Important Protocols by Layer

| OSI Layer | Protocols |
| --- | --- |
| Application | HTTP, HTTPS, FTP, SMTP, SSH, DNS, SNMP |
| Presentation | TLS, SSL, JPEG, ASCII, MPEG |
| Session | NetBIOS, RPC |
| Transport | TCP, UDP |
| Network | IP, ICMP, ARP, BGP, OSPF |
| Data Link | Ethernet, PPP, MAC, VLAN |
| Physical | Ethernet cables, RJ45, Fiber optics |

## 🛠️ 4. DevOps Tools & Networking Concepts

* **Ping**: Tests connectivity using ICMP
* **Traceroute**: Shows route packets take (network hops)
* **netstat**: Show ports/services listening
* **iptables/firewalld**: Firewall config in Linux
* **cURL / wget**: HTTP requests (test APIs)
* **nmap**: Network scanner (security, open ports)
* **DNS Lookup Tools**: dig, nslookup
* **Hosts File**: Override DNS locally (/etc/hosts)
* **CIDR Notation**: IP block notation (192.168.0.0/24)
* **Subnetting**: Splitting network into parts
* **VPNs & Proxies**: For tunneling and obfuscation
* **Load Balancers**: Distribute traffic
* **Reverse Proxy (e.g., Nginx)**: Handles incoming requests

## 🧠 5. Must-Know Networking Commands (Linux CLI)

bash

CopyEdit

ping google.com # Test connection

curl ifconfig.me # Public IP

ip a # Show IP addresses

netstat -tulnp # Open ports

ss -tuln # Modern netstat alternative

traceroute google.com # Route path

nslookup openai.com # DNS resolution

dig openai.com # Advanced DNS lookup

tcpdump -i eth0 port 80 # Packet capture (like Wireshark)

nmap -p 1-65535 localhost # Full port scan

## 📘 Recommended Resources

1. **FreeCodeCamp: Networking for Beginners**
2. **Practical Networking** (<https://www.practicalnetworking.net/>)
3. **Wireshark** – Learn packet analysis
4. **Hack The Box / TryHackMe** – For real-world labs
5. **Cisco Packet Tracer** – Visual OSI simulations

created by chatgpt to quickly brush up cn topics   
can also checkout

https://youtu.be/IPvYjXCsTg8?si=9Xf-sX8PjYXuVMqm