SESSION 1 & SESSION 2 – COMPUTER NETWORKING (PG-DCS)

1. INTERNetworking

1.1 Definition

Internetworking is the process of **interconnecting multiple independent networks** (LANs, MANs, WANs) using networking devices and protocols so that they function as **one logical network**.

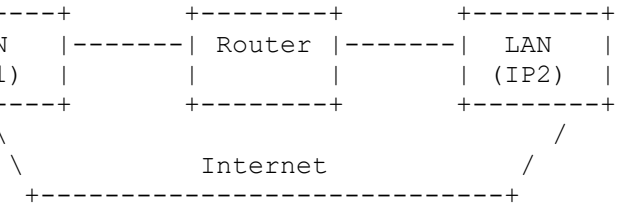
1.2 Purpose & Need

- Enable **communication across heterogeneous networks**
- Share **resources** (servers, printers, internet)
- Provide **scalability**
- Support **enterprise & global connectivity**
- Enable **fault tolerance & redundancy**

1.3 History

- Originated with **ARPANET (1969)**
- Development of **TCP/IP (1970s–80s)**
- Commercial internet expansion in the **1990s**
- Today: Cloud, IoT, SDN, 5G

1.4 Internetworking Architecture



1.5 Key Components

- **Routers** – Connect different networks
- **Switches** – Connect devices within LAN
- **Gateways** – Protocol translation
- **Firewalls** – Security control
- **Links** – Wired / Wireless

1.6 Internetworking Devices Comparison

Device	Layer	Purpose
Hub	L1	Broadcast signal
Switch	L2	MAC-based forwarding
Router	L3	IP-based routing
Gateway	L4–L7	Protocol conversion
Firewall	L3–L7	Traffic filtering

1.7 Internetworking Protocols

- **IP** – Logical addressing
- **ICMP** – Error & diagnostics
- **ARP** – IP to MAC resolution
- **RIP / OSPF / BGP** – Routing
- **NAT** – Address translation

1.8 Types of Internetworking

1. **LAN-to-LAN**
2. **LAN-to-WAN**
3. **WAN-to-WAN**
4. **Internet-based Internetworking**

1.9 Real-World Example

- Corporate branch offices connected via **MPLS VPN**

- Home network connecting to ISP via router
- Cloud VPC peering

1.10 Advantages & Disadvantages

☑ Advantages

- Resource sharing
- Centralized control
- Scalability
- Cost efficiency

❌ Disadvantages

- Complexity
- Security risks
- Latency
- Troubleshooting difficulty

1.11 Exam Points

- Router works at **Layer 3**
- Internetworking enables **heterogeneous networks**
- TCP/IP is the backbone protocol suite

1.12 Interview Q&A

Q: What is internetworking?
A: Connecting multiple networks using routers and protocols so they operate as one.

---

2. OSI MODEL (Open Systems Interconnection)

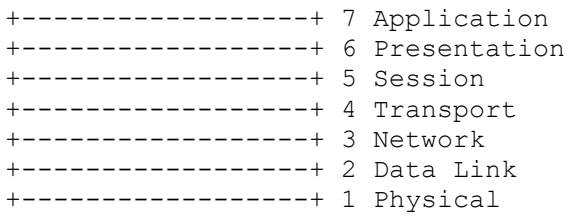
2.1 Definition

The **OSI Model** is a **7-layer conceptual framework** developed by **ISO** to standardize network communication.

2.2 Purpose

- Standardization
- Vendor interoperability
- Troubleshooting
- Network design clarity

2.3 OSI Architecture Diagram



2.4 Detailed Layer-Wise Explanation (ASSIGNMENT)

◇ Layer 7 – Application

- User interface to network
- Protocols: HTTP, FTP, SMTP, DNS
- Example: Web browser

◇ Layer 6 – Presentation

- Encryption / Decryption
- Compression
- Encoding
- Example: SSL/TLS

◇ Layer 5 – Session

- Session establishment
- Session maintenance
- Session termination
- Example: NetBIOS

◇ Layer 4 – Transport

- End-to-end delivery
- Flow & error control
- Protocols:
    - **TCP** – Reliable
    - **UDP** – Fast, unreliable

◇ Layer 3 – Network

- Logical addressing
- Routing
- Protocols: IP, ICMP, RIP, OSPF
- Devices: Router

◇ Layer 2 – Data Link

- Framing
- MAC addressing
- Error detection
- Protocols: Ethernet, ARP
- Devices: Switch

◇ Layer 1 – Physical

- Transmission of bits
- Cables, connectors, voltages
- Devices: Hub, Repeater

2.5 OSI Mnemonic

All **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing

---

2.6 Exam & Interview Tips

- Encryption → Layer 6
- Routing → Layer 3
- MAC address → Layer 2

---

3. ETHERNET

3.1 Definition

Ethernet is a **LAN technology** that defines **wiring, signaling, and frame formats**.

3.2 IEEE Standard

- **IEEE 802.3**

3.3 Ethernet Architecture

```
PC ——— Switch ——— Server
        |
     Router
```

3.4 Ethernet Frame Format

```
| Preamble | Dest MAC | Src MAC | Type | Data | FCS |
```

3.5 Ethernet Types

Standard	Speed
Ethernet	10 Mbps
Fast Ethernet	100 Mbps
Gigabit Ethernet	1 Gbps
10G Ethernet	10 Gbps

3.6 Ethernet Characteristics

- CSMA/CD (legacy)
- MAC addressing
- Full duplex
- Reliable LAN delivery

3.7 Real-World Use

- Office LAN
- Data centers
- Campus networks

3.8 Advantages & Limitations

☑ Advantages

- Low cost
- Scalable
- High speed

❌ Limitations

- Distance limitation
- Broadcast traffic

---

4. WIRELESS NETWORKING

4.1 Definition

Wireless networking allows **data transmission without physical cables** using **radio waves**.

4.2 IEEE Standards (Wi-Fi)

Standard	Speed
802.11b	11 Mbps
802.11g	54 Mbps
802.11n	600 Mbps
802.11ac	>1 Gbps
802.11ax (Wi-Fi 6)	High efficiency

4.3 Wireless Architecture

```
Laptop )))) Access Point )))) Internet
Phone  ))))
```

4.4 Wireless Components

- Access Point
- Wireless NIC
- Antenna
- Authentication server

4.5 Security Protocols

- WEP (weak)
- WPA
- WPA2
- WPA3 (strongest)

4.6 Wireless Networking Types

- Infrastructure mode
- Ad-hoc mode
- Mesh networking

4.7 Advantages & Disadvantages

☑ Advantages

- Mobility
- Easy installation
- Cost-effective

❌ Disadvantages

- Security risks
- Interference
- Lower speed than wired

📌 ASSIGNMENT TOPICS (DETAILED)

A. Difference Between UTP & STP

Feature	UTP	STP
Shielding	No	Yes
EMI Protection	Low	High
Cost	Low	High
Installation	Easy	Complex
Usage	LAN	Industrial

B. Categories of Network Cables

Category	Speed
Cat3	10 Mbps
Cat5	100 Mbps

Category	Speed
Cat5e	1 Gbps
Cat6	10 Gbps
Cat6a	10 Gbps
Cat7	40 Gbps

C. Meaning of “e” in CAT5e

“e” = Enhanced

- Improved specifications
- Reduced crosstalk
- Supports **Gigabit Ethernet**

D. OSI Model – Exam-Oriented Summary

- 7 layers
- Layered communication
- Each layer has defined responsibility
- Used for **design & troubleshooting**

🔍 Troubleshooting Using OSI

Issue	Layer
Cable unplugged	L1
MAC conflict	L2
IP unreachable	L3
Slow transmission	L4
Application error	L7

🎯 Mini Case Study

Company Network

- Wired Ethernet LAN
- Wireless access for employees
- Router connects to ISP
- Firewall for security
- OSI model used for troubleshooting

🎓 FINAL EXAM KEYWORDS

- Internetworking
- OSI Layers
- Ethernet (IEEE 802.3)
- Wireless (802.11)
- UTP vs STP
- Cat5e Enhanced

SESSION 3 – NETWORK PROTOCOLS (PG-DCS)

---

1. INTERNET PROTOCOL (IP)

1.1 Definition

Internet Protocol (IP) is a **network-layer (Layer 3) protocol** responsible for **logical addressing, packetization, and routing of data packets** across interconnected networks.

1.2 Purpose & Need of IP

- Provide **unique identification** to devices (IP address)
- Enable **routing across multiple networks**
- Support **scalability of the Internet**
- Allow **heterogeneous network communication**

1.3 History of IP

- Developed under **ARPANET**
- Part of **TCP/IP protocol suite**
- IPv4 standardized in **1981**
- IPv6 introduced to overcome address exhaustion

1.4 IP Architecture (Logical View)

```
Application Data
      ↓
Transport Segment
      ↓
IP Packet
      ↓
Frame
      ↓
Bits on wire
```

1.5 Functions of Internet Protocol

1. Logical addressing
2. Packet encapsulation
3. Routing & forwarding
4. Fragmentation & reassembly
5. Best-effort delivery (connectionless)

1.6 IP Addressing Concepts (ASSIGNMENT – DETAILED)

1.6.1 IP Address

An **IP address** is a **unique numerical identifier** assigned to a network interface.

Example:

```
IPv4: 192.168.1.10
IPv6: 2001:db8::1
```

1.6.2 IPv4 Address Structure (32-bit)

```
| Network Part | Host Part |
```

Binary Representation:

```
192.168.1.10
11000000.10101000.00000001.00001010
```

1.6.3 Classes of IPv4 Addresses

Class	Range	Default Mask	Usage
A	1 – 126	/8	Large networks
B	128 – 191	/16	Medium networks
C	192 – 223	/24	Small networks
D	224 – 239	Multicast	Streaming
E	240 – 255	Reserved	Research

1.6.4 Private vs Public IP

Type	Range
Private	10.0.0.0/8
	172.16.0.0/12
	192.168.0.0/16
Public	ISP assigned

1.6.5 Subnet Mask

Defines **network and host portion**.

Example:

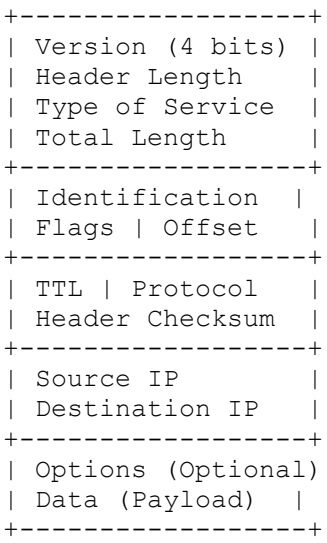
```
IP: 192.168.1.10
Mask: 255.255.255.0 (/24)
```

1.6.6 CIDR Notation

```
192.168.1.0/24
```

- Improves routing efficiency
- Supports VLSM

1.7 IPv4 Packet Structure (ASSIGNMENT)



1.8 Important IP Header Fields

- **TTL** – Prevents infinite loops
- **Protocol** – Indicates TCP/UDP
- **Checksum** – Error detection
- **Source/Destination IP**

1.9 IPv6 Overview

- 128-bit addressing
- Hexadecimal notation
- No broadcast (uses multicast)
- Built-in security (IPSec)

1.10 Advantages & Limitations of IP

☑ Advantages

- Scalable
- Interoperable
- Simple routing

❌ Limitations

- No reliability
- No error recovery
- No congestion control

1.11 Real-World IP Use Cases

- Internet browsing
- Cloud networking
- VPN tunnels
- Mobile networks

1.12 Exam Focus (IP)

- IP is **connectionless**
- Works at **Network layer**
- IPv4 = 32-bit, IPv6 = 128-bit

1.13 Interview Q&A (IP)

Q: Why is IP unreliable?
A: It provides best-effort delivery without acknowledgments or retransmission.

---

2. TCP/IP MODEL

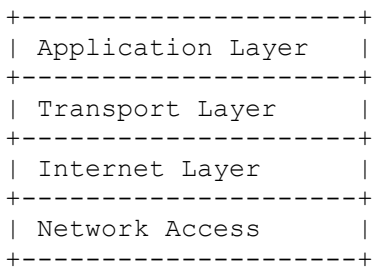
2.1 Definition

The **TCP/IP Model** is a **practical 4-layer networking model** that governs **Internet communication**.

2.2 Purpose of TCP/IP Model

- Enable **end-to-end communication**
- Provide **interoperability**
- Support **internet scalability**
- Serve as **real-world implementation model**

2.3 TCP/IP Model Layers (ASSIGNMENT)



2.4 Layer-Wise Detailed Explanation

◇ 1. Application Layer

- Combines OSI Layers 5, 6, 7
- Provides services to applications

Protocols:

- HTTP / HTTPS
- FTP
- SMTP / POP3 / IMAP
- DNS
- SNMP

◇ 2. Transport Layer

- End-to-end communication
- Flow control
- Error control

Protocols:

- TCP
- UDP

◇ 3. Internet Layer

- Logical addressing
- Routing

Protocols:

- IP
- ICMP
- ARP
- RARP

◇ 4. Network Access Layer

- Physical transmission
- Framing

Technologies:

- Ethernet
- Wi-Fi
- ARP
- MAC addressing

2.5 OSI vs TCP/IP Model (Comparison Table)

OSI	TCP/IP
7 layers	4 layers
Conceptual	Practical
ISO model	DARPA model
Not implemented directly	Implemented

2.6 Protocol Workflow (End-to-End)

```
Client Application
   ↓ HTTP
Transport (TCP)
   ↓ Segment
Internet (IP)
   ↓ Packet
Network (Ethernet)
   ↓ Frame
Transmission
```

---

3. TCP vs UDP (ASSIGNMENT – DETAILED)

3.1 Transmission Control Protocol (TCP)

Characteristics

- Connection-oriented
- Reliable
- Ordered delivery
- Congestion control

TCP 3-Way Handshake

```
Client → SYN → Server
Client ← SYN-ACK ← Server
Client → ACK → Server
```

3.2 User Datagram Protocol (UDP)

Characteristics

- Connectionless
- Unreliable
- Faster
- No flow control

3.3 TCP vs UDP Comparison Table

Feature	TCP	UDP
Connection	Yes	No
Reliability	High	Low
Speed	Slower	Faster
Header Size	20 bytes	8 bytes
Use Case	Web, Email	Streaming, VoIP

3.4 Real-World Examples

- **TCP**: HTTP, HTTPS, FTP
- **UDP**: DNS, Video streaming, Online games

3.5 Exam & Interview Focus (TCP/UDP)

- TCP ensures reliability
- UDP is used where speed matters
- DNS uses UDP (mostly)

3.6 Mini Case Study

Video Streaming Platform

- Control messages → TCP
- Media streaming → UDP
- IP handles routing
- TCP/IP model governs communication

3.7 Troubleshooting Using TCP/IP Layers

Problem	Layer
No IP address	Internet
Website not loading	Application
Packet loss	Transport
Cable issue	Network Access

3.8 Key Exam Keywords

- Best-effort delivery
- Logical addressing
- Encapsulation
- TCP handshake
- UDP datagrams

3.9 Quick Revision Points

- IP → Layer 3 (OSI), Internet layer (TCP/IP)
- TCP/IP has **4 layers**
- TCP reliable, UDP fast
- IP packet has TTL & checksum

3.10 Interview Rapid Q&A

Q: Why TCP/IP model is preferred over OSI?
A: It is practical, simpler, and implemented in real networks.

Q: Which layer does IP belong to in TCP/IP?
A: Internet layer.

SESSION 4 & 5 – IP SUBNETTING & VLSM

PART A: IP SUBNETTING

---

1. IP SUBNETTING – INTRODUCTION

1.1 Definition

IP Subnetting is the process of **dividing a large IP network into smaller logical sub-networks (subnets)** by borrowing bits from the host portion.

1.2 Why Subnetting is Required

- Efficient IP address utilization
- Reduced broadcast traffic
- Improved network security
- Easier network management
- Hierarchical network design

1.3 Subnetting Terminology

Term	Meaning
Network Address	Identifies subnet
Broadcast Address	Last address in subnet
Host Addresses	Assignable IPs
Subnet Mask	Defines network/host bits
CIDR	Classless addressing

---

2. IPv4 ADDRESS STRUCTURE (RECAP)

```
IPv4 = 32 bits
| Network bits | Host bits |
```

Example:

```
192.168.1.0/24
Network = 24 bits
Host = 8 bits
```

---

3. CLASSFUL SUBNETTING (EXAM IMPORTANT)

Class	Default Mask	Hosts
A	/8	16 million
B	/16	65,534
C	/24	254

---

4. STEP-BY-STEP SUBNETTING METHOD (CORE)

Step 1: Identify Class

Example:

```
192.168.1.0 → Class C
```

Step 2: Identify Default Mask

```
Class C → /24
```

Step 3: Identify Required Subnets / Hosts

Step 4: Borrow Host Bits

Formula:

```
2^n ≥ Required Subnets
```

Step 5: Calculate New Subnet Mask

Step 6: Calculate:

- Network address
- First host
- Last host
- Broadcast address

---

5. BINARY SUBNETTING FUNDAMENTALS

5.1 Binary Place Values

```
128 64 32 16 8 4 2 1
```

5.2 Subnet Mask Calculation Example

Borrow 3 bits:

```
11111111.11111111.11111111.11100000
= 255.255.255.224 (/27)
```

---

6. NETWORK & HOST PORTION IDENTIFICATION (ASSIGNMENT)

Example 1:

IP: `192.168.5.85/24`

```
Subnet Mask: 255.255.255.0
Network Part: 192.168.5
Host Part: 85
```

- ✔ **Network:** 192.168.5.0
- ✔ **Broadcast:** 192.168.5.255
- ✔ **Host Range:** 192.168.5.1 – 192.168.5.254

Example 2:

IP: `10.128.240.50/30`

```
/30 → Block size = 4
```

Subnets:

```
10.128.240.48
10.128.240.52
```

- ✔ **Network:** 10.128.240.48
- ✔ **Broadcast:** 10.128.240.51
- ✔ **Usable Hosts:** 10.128.240.49, 10.128.240.50

---

7. SUBNET DESIGN PROBLEMS (FIXED LENGTH)

Problem:

Subnet `192.168.1.0/24` into **4 equal subnets**

```
2^2 = 4 subnets
Borrow 2 bits → /26
```

Subnet	Network	Broadcast
1	192.168.1.0	.63
2	192.168.1.64	.127
3	192.168.1.128	.191
4	192.168.1.192	.255

---

8. LIMITATIONS OF FLSM

- Wastes IP addresses
- Not scalable
- Inefficient for real-world networks

PART B: VARIABLE LENGTH SUBNET MASKING (VLSM)

---

9. VLSM – INTRODUCTION

9.1 Definition

VLSM allows **different subnet masks within the same network**, optimizing IP usage.

9.2 Why VLSM?

- Efficient address allocation
- ISP and enterprise usage

- Supports hierarchical routing
- Reduces IP wastage

---

10. VLSM DESIGN RULES (VERY IMPORTANT)

1. Sort subnets in **descending order of host requirement**
2. Allocate largest subnet first
3. Move sequentially
4. Never overlap subnets

---

11. VLSM CALCULATION FORMULA

```
Hosts needed → Find nearest 2^n
Subnet size = 2^n
Subnet mask = 32 − n
```

---

12. VLSM BASED NETWORK DIVISION

Given Network:

```
192.168.1.0/24
```

Requirements:

Subnet	Hosts
Subnet 1	52
Subnet 2	28
Subnet 3	15
Subnet 4	5

Step 1: Sort by Host Size

```
52 → 28 → 15 → 5
```

Step 2: Calculate Required Blocks

Hosts	Block	Mask
52	64	/26
28	32	/27
15	16	/28
5	8	/29

Step 3: Allocate Subnets

Subnet 1 (52 Hosts)

```
Network: 192.168.1.0/26
Usable: .1 – .62
Broadcast: .63
```

Subnet 2 (28 Hosts)

```
Network: 192.168.1.64/27
Usable: .65 – .94
Broadcast: .95
```

Subnet 3 (15 Hosts)

```
Network: 192.168.1.96/28
Usable: .97 – .110
Broadcast: .111
```

Subnet 4 (5 Hosts)

```
Network: 192.168.1.112/29
Usable: .113 – .118
Broadcast: .119
```

---

13. VLSM DIAGRAM (ASCII)

```
192.168.1.0/24
│
├── /26 (52 hosts)
├── /27 (28 hosts)
├── /28 (15 hosts)
├── /29 (5 hosts)
```

---

14. REAL-WORLD ISP EXAMPLE

- ISP receives `/20`

- Allocates:
  - Enterprise → /22
  - Medium → /24
  - Small → /27
- Achieves **efficient address utilization**

---

15. SUBNETTING TROUBLESHOOTING

Problem	Reason
IP conflict	Overlapping subnets
No connectivity	Wrong mask
Broadcast storm	Large subnet

---

16. EXAM-FOCUSED NUMERICAL SHORTCUTS

CIDR	Hosts
/30	2
/29	6
/28	14
/27	30
/26	62
/25	126

---

17. COMMON MISTAKES (INTERVIEW ALERT 🚨)

- Forgetting network & broadcast
- Wrong block size
- Not sorting VLSM requirements
- Mixing FLSM & VLSM

---

18. SUBNETTING vs VLSM (COMPARISON)

Feature	Subnetting	VLSM
Mask	Same	Different
Efficiency	Low	High

Feature	Subnetting	VLSM
Usage	Legacy	Modern

---

19. QUICK REVISION POINTS

- Subnetting divides networks
- VLSM saves IPs
- Binary is the key
- Always allocate largest subnet first

---

20. INTERVIEW Q&A

Q: Why VLSM is better than FLSM?
A: It prevents IP wastage by using variable masks.

Q: What is /30 used for?
A: Point-to-point links.

---

21. MINI CASE STUDY

Enterprise Network

- HQ: /26
- Branch: /27
- Remote office: /29
- Implemented using VLSM

🎯 FINAL EXAM KEYWORDS

- CIDR
- Subnet mask
- Block size
- Network address
- Broadcast address
- VLSM hierarchy

SESSION 4 & 5 – IP SUBNETTING & VLSM

PART A: IP SUBNETTING

---

1. IP SUBNETTING – INTRODUCTION

1.1 Definition

IP Subnetting is the process of **dividing a large IP network into smaller logical sub-networks (subnets)** by borrowing bits from the host portion.

1.2 Why Subnetting is Required

- Efficient IP address utilization
- Reduced broadcast traffic
- Improved network security
- Easier network management
- Hierarchical network design

1.3 Subnetting Terminology

Term	Meaning
Network Address	Identifies subnet
Broadcast Address	Last address in subnet
Host Addresses	Assignable IPs
Subnet Mask	Defines network/host bits
CIDR	Classless addressing

---

2. IPv4 ADDRESS STRUCTURE (RECAP)

```
IPv4 = 32 bits
| Network bits | Host bits |
```

Example:

```
192.168.1.0/24
Network = 24 bits
Host = 8 bits
```

---

3. CLASSFUL SUBNETTING (EXAM IMPORTANT)

Class	Default Mask	Hosts
A	/8	16 million
B	/16	65,534
C	/24	254

---

4. STEP-BY-STEP SUBNETTING METHOD (CORE)

Step 1: Identify Class

Example:

```
192.168.1.0 → Class C
```

Step 2: Identify Default Mask

```
Class C → /24
```

Step 3: Identify Required Subnets / Hosts

Step 4: Borrow Host Bits

Formula:

```
2^n ≥ Required Subnets
```

Step 5: Calculate New Subnet Mask

Step 6: Calculate:

- Network address
- First host
- Last host
- Broadcast address

---

5. BINARY SUBNETTING FUNDAMENTALS

5.1 Binary Place Values

```
128 64 32 16 8 4 2 1
```

5.2 Subnet Mask Calculation Example

Borrow 3 bits:

```
11111111.11111111.11111111.11100000
= 255.255.255.224 (/27)
```

---

6. NETWORK & HOST PORTION IDENTIFICATION (ASSIGNMENT)

Example 1:

IP: `192.168.5.85/24`

```
Subnet Mask: 255.255.255.0
Network Part: 192.168.5
Host Part: 85
```

- ✔ **Network:** 192.168.5.0
- ✔ **Broadcast:** 192.168.5.255
- ✔ **Host Range:** 192.168.5.1 – 192.168.5.254

Example 2:

IP: `10.128.240.50/30`

```
/30 → Block size = 4
```

Subnets:

```
10.128.240.48
10.128.240.52
```

- ✔ **Network:** 10.128.240.48
- ✔ **Broadcast:** 10.128.240.51
- ✔ **Usable Hosts:** 10.128.240.49, 10.128.240.50

---

7. SUBNET DESIGN PROBLEMS (FIXED LENGTH)

Problem:

Subnet `192.168.1.0/24` into **4 equal subnets**

```
2^2 = 4 subnets
Borrow 2 bits → /26
```

Subnet	Network	Broadcast
1	192.168.1.0	.63
2	192.168.1.64	.127
3	192.168.1.128	.191
4	192.168.1.192	.255

---

8. LIMITATIONS OF FLSM

- Wastes IP addresses
- Not scalable
- Inefficient for real-world networks

PART B: VARIABLE LENGTH SUBNET MASKING (VLSM)

---

9. VLSM – INTRODUCTION

9.1 Definition

VLSM allows **different subnet masks within the same network**, optimizing IP usage.

9.2 Why VLSM?

- Efficient address allocation
- ISP and enterprise usage
- Supports hierarchical routing
- Reduces IP wastage

---

10. VLSM DESIGN RULES (VERY IMPORTANT)

1. Sort subnets in **descending order of host requirement**
2. Allocate largest subnet first
3. Move sequentially
4. Never overlap subnets

---

11. VLSM CALCULATION FORMULA

```
Hosts needed → Find nearest 2^n
Subnet size = 2^n
Subnet mask = 32 − n
```

---

12. VLSM BASED NETWORK DIVISION (ASSIGNMENT – FULLY SOLVED)

Given Network:

```
192.168.1.0/24
```

Requirements:

Subnet	Hosts
Subnet 1	52
Subnet 2	28
Subnet 3	15
Subnet 4	5

Step 1: Sort by Host Size

```
52 → 28 → 15 → 5
```

Step 2: Calculate Required Blocks

Hosts	Block	Mask
52	64	/26
28	32	/27
15	16	/28
5	8	/29

Step 3: Allocate Subnets

Subnet 1 (52 Hosts)

```
Network: 192.168.1.0/26
Usable: .1 – .62
Broadcast: .63
```

Subnet 2 (28 Hosts)

```
Network: 192.168.1.64/27
Usable: .65 – .94
Broadcast: .95
```

Subnet 3 (15 Hosts)

```
Network: 192.168.1.96/28
Usable: .97 – .110
Broadcast: .111
```

Subnet 4 (5 Hosts)

```
Network: 192.168.1.112/29
Usable: .113 – .118
Broadcast: .119
```

---

13. VLSM DIAGRAM (ASCII)

```
192.168.1.0/24
│
├── /26 (52 hosts)
├── /27 (28 hosts)
├── /28 (15 hosts)
├── /29 (5 hosts)
```

---

14. REAL-WORLD ISP EXAMPLE

- ISP receives `/20`
- Allocates:
  - Enterprise → /22
  - Medium → /24
  - Small → /27
- Achieves **efficient address utilization**

---

15. SUBNETTING TROUBLESHOOTING

Problem	Reason
IP conflict	Overlapping subnets
No connectivity	Wrong mask
Broadcast storm	Large subnet

---

16. EXAM-FOCUSED NUMERICAL SHORTCUTS

CIDR	Hosts
/30	2
/29	6
/28	14
/27	30
/26	62
/25	126

---

17. COMMON MISTAKES (INTERVIEW ALERT 🚨)

- Forgetting network & broadcast
- Wrong block size
- Not sorting VLSM requirements
- Mixing FLSM & VLSM

---

18. SUBNETTING vs VLSM (COMPARISON)

Feature	Subnetting	VLSM
Mask	Same	Different
Efficiency	Low	High
Usage	Legacy	Modern

---

19. QUICK REVISION POINTS

- Subnetting divides networks
- VLSM saves IPs
- Binary is the key
- Always allocate largest subnet first

---

20. INTERVIEW Q&A

Q: Why VLSM is better than FLSM?
A: It prevents IP wastage by using variable masks.

Q: What is /30 used for?
A: Point-to-point links.

---

21. MINI CASE STUDY

Enterprise Network

- HQ: /26

- Branch: /27
- Remote office: /29
- Implemented using VLSM

🎯 FINAL EXAM KEYWORDS

- CIDR
- Subnet mask
- Block size
- Network address
- Broadcast address
- VLSM hierarchy

🟫 SESSION 7 – MANAGING AN INTERNETWORKING ROUTER

---

1. INTRODUCTION TO ROUTER MANAGEMENT

1.1 Definition

Managing an Internetworking Router involves **monitoring, configuring, securing, maintaining, and troubleshooting routers** to ensure **reliable communication across interconnected networks**.

1.2 Objectives of Router Management

- Ensure continuous network availability
- Control routing and forwarding
- Secure management access
- Optimize performance
- Rapid fault detection and recovery

1.3 Router Management Plan (Industry View)

- Initial configuration
- Access control
- Configuration backup
- Monitoring & logging
- Change management
- Disaster recovery

---

2. ROUTER BOOTING PROCESS (LAB + EXAM CORE)

2.1 Router Booting – Definition

The **router booting process** is the **sequence of steps executed when a router is powered on** to load the IOS and configuration.

2.2 Router Boot Sequence (Step-by-Step)

```
Power ON
   ↓
POST (Power-On Self Test)
   ↓
Bootstrap Program (ROM)
   ↓
Locate IOS Image
   ↓
Load IOS into RAM
   ↓
Load startup-config from NVRAM
   ↓
Create running-config in RAM
```

2.3 Memory Involvement During Boot

Memory	Role
ROM	POST, bootstrap
Flash	Stores IOS
RAM	Running IOS + config
NVRAM	Startup config

2.4 Boot Failure Scenarios (LAB IMPORTANT)

- IOS missing → **ROMMON mode**
- Corrupt IOS → TFTP recovery
- No startup-config → Setup mode

2.5 Exam Focus

- Bootstrap program resides in **ROM**
- IOS executes from **RAM**
- Startup config stored in **NVRAM**

---

3. CONFIGURATION REGISTERS (CRITICAL TOPIC)

3.1 Definition

A **configuration register** is a **16-bit value** that controls **router boot behavior and password recovery options**.

3.2 Common Configuration Register Values

Register	Meaning
`0x2102`	Normal boot (default)
`0x2142`	Ignore startup-config
`0x2100`	Boot to ROMMON

3.3 Password Recovery Workflow (LAB SCENARIO)

```
Power cycle router
↓
Break sequence
↓
ROMMON mode
↓
Change register to 0x2142
↓
Boot router
```

```
↓
Access without password
↓
Restore config & reset register
```

3.4 Command to View Register

```
show version
```

3.5 Exam & Interview Tip

- **0x2142** is used for **password recovery**

---

4. IOS IMAGE HANDLING (OPERATIONS CORE)

4.1 IOS Image – Definition

An **IOS image** is the **binary operating system file** that provides router functionality.

4.2 IOS Image Storage Locations

- Flash (primary)
- TFTP server
- USB (newer routers)
- ROM (limited IOS)

4.3 Viewing IOS Image

```
show flash
show version
```

4.4 IOS Backup to TFTP (LAB COMMANDS)

```
copy flash tftp
```

4.5 IOS Restore from TFTP

```
copy tftp flash
```

4.6 Boot System Command

```
boot system flash:c1900-universalk9.bin
```

4.7 IOS Upgrade Best Practices

- Verify compatibility
- Backup existing IOS
- Ensure sufficient flash space
- Verify MD5 checksum

4.8 Exam Focus

- IOS runs from **RAM**
- Flash stores **persistent IOS image**

---

5. TELNET ACCESS TO ROUTER

5.1 Telnet – Definition

Telnet is a **remote management protocol** that provides **CLI access over TCP/IP**.

5.2 Telnet Characteristics

- TCP port **23**
- Plain-text (insecure)
- Used for legacy networks

5.3 Telnet Configuration (LAB ALIGNED)

```
line vty 0 4
password cisco
login
transport input telnet
```

5.4 Testing Telnet

```
telnet 192.168.1.1
```

5.5 Telnet Limitations (SECURITY ALERT 🚨)

- No encryption
- Vulnerable to sniffing
- Replaced by SSH

5.6 Exam Point

- Telnet works on **VTY lines**

---

6. HOSTNAME RESOLUTION (NETWORK MANAGEMENT)

6.1 Definition

Hostname resolution allows **human-friendly names** to be mapped to **IP addresses**.

6.2 Local Hostname Resolution (Router)

```
ip host R1 192.168.1.1
ip host R2 192.168.1.2
```

6.3 Enabling DNS Lookup

```
ip domain-lookup
```

6.4 DNS Server Configuration

```
ip name-server 8.8.8.8
```

6.5 Disabling DNS Lookup (LAB TIP)

```
no ip domain-lookup
```

6.6 Exam & Interview Focus

- Router uses DNS if domain lookup enabled
- Incorrect commands may cause delays due to DNS lookup

---

7. DEBUGGING CONCEPTS (ADVANCED OPERATIONS)

7.1 Debugging – Definition

Debugging provides **real-time diagnostic information** about router operations.

7.2 Debug vs Show Commands

Feature	show	debug
Output	Static	Real-time
CPU load	Low	High
Usage	Monitoring	Troubleshooting

7.3 Common Debug Commands

```
debug ip routing
debug ip packet
debug interface
```

7.4 Stopping Debugging

```
undebug all
```

7.5 Debugging Best Practices (EXAM ALERT 🚨)

- Use during low traffic
- Disable immediately after use
- Prefer `show` first

---

8. ROUTER MANAGEMENT SECURITY CONSIDERATIONS

8.1 Secure Access

- Use SSH instead of Telnet
- Strong passwords
- Enable secret

8.2 Configuration Protection

- Backup configs
- Role-based CLI
- Access Control Lists

8.3 Logging & Monitoring

- Syslog
- SNMP
- NTP for time sync

---

9. TROUBLESHOOTING INTERNETWORKING ROUTERS

9.1 Common Issues

Problem	Cause
Router not booting	IOS missing
Remote login fails	VTY misconfig
Slow response	Debug enabled
Name resolution delay	DNS lookup

9.2 Systematic Troubleshooting Steps

1. Identify problem
2. Check physical & IP
3. Verify config
4. Test connectivity
5. Apply fix
6. Verify & document

---

10. REAL-WORLD ENTERPRISE SCENARIO

Branch Router Management

- IOS stored in flash
- Backup via TFTP
- Telnet disabled, SSH enabled
- Hostname resolution via DNS
- Debug used for routing issue

---

11. EXAM-ORIENTED QUICK REVISION

- Boot sequence = POST → Bootstrap → IOS → Config
- Default config register = **0x2102**
- Telnet = TCP 23 (insecure)
- Debug is CPU intensive
- IOS stored in flash

---

12. INTERVIEW Q&A

Q: Why is `no ip domain-lookup` used?
A: To prevent delays caused by DNS lookup on mistyped commands.

Q: Difference between show and debug?
A: Show gives snapshot; debug gives real-time data.

Q: Which register ignores startup config?
A: `0x2142`

SESSION 8 – ROUTING CONCEPTS & ROUTING PROTOCOLS

---

1. INTRODUCTION TO ROUTING

1.1 What is Routing?

Routing is the process of **selecting the best path** for forwarding packets from a **source network to a destination network** using **routers**.

1.2 Why Routing is Required

- Enables **inter-network communication**
- Supports **scalability** (LAN → WAN → Internet)
- Allows **path selection & redundancy**
- Enables **fault tolerance**

1.3 Routing vs Switching (Quick Recall)

Feature	Routing	Switching
OSI Layer	Layer 3	Layer 2
Address Used	IP Address	MAC Address
Device	Router	Switch
Scope	Inter-network	Intra-network

1.4 Routing Table – Core Concept

A **routing table** is a data structure stored in router memory that contains:

- Destination network
- Next hop
- Metric
- Routing protocol
- Outgoing interface

```
Destination   Next Hop      Metric   Interface
192.168.2.0   10.0.0.2        1       G0/0
```

---

2. STATIC ROUTING

2.1 Definition

Static routing is a routing method where **routes are manually configured by an administrator** and do not change unless manually modified.

2.2 Characteristics of Static Routing

- Manually configured
- No routing updates
- Low CPU & bandwidth usage
- Not adaptive to topology changes

2.3 Static Route Configuration (Cisco IOS)

```
ip route <destination-network> <subnet-mask> <next-hop-ip>
```

Example:

```
ip route 192.168.2.0 255.255.255.0 10.0.0.2
```

2.4 Types of Static Routes

2.4.1 Standard Static Route

- Points to a next-hop IP

2.4.2 Default Route

Used when no specific route matches.

```
ip route 0.0.0.0 0.0.0.0 10.0.0.1
```

2.4.3 Floating Static Route

- Backup route
- Higher Administrative Distance

```
ip route 192.168.2.0 255.255.255.0 10.0.0.3 200
```

2.5 Advantages of Static Routing

- Simple
- Predictable paths
- Secure (no advertisements)
- Ideal for small networks

2.6 Disadvantages of Static Routing

- Poor scalability
- Manual maintenance
- No automatic failover
- Error-prone in large networks

2.7 Real-World Use Cases

- Stub networks
- Default route to ISP
- Small branch offices

2.8 Exam Points (Static Routing)

- No routing protocol used
- Administrative Distance = **1**
- Default route = `0.0.0.0/0`

---

3. DYNAMIC ROUTING

3.1 Definition

Dynamic routing uses **routing protocols** to automatically **discover, learn, update, and maintain routes** based on network topology.

3.2 Characteristics of Dynamic Routing

- Automatic route learning
- Adapts to failures
- Uses routing updates
- Consumes CPU & bandwidth

3.3 Dynamic Routing Workflow

```
Router starts
   ↓
Routing protocol enabled
   ↓
Routers exchange updates
   ↓
Routing table populated
   ↓
Best path selected
```

3.4 Advantages of Dynamic Routing

- Scalable
- Automatic failover

- Easier management
- Suitable for large networks

3.5 Disadvantages of Dynamic Routing

- Complex configuration
- Consumes bandwidth
- Slower convergence (some protocols)

3.6 Real-World Use Cases

- Enterprise networks
- ISP backbones
- Data centers

3.7 Exam Focus

- Dynamic routing uses **routing protocols**
- Routing table updates are automatic

---

4. STATIC vs DYNAMIC ROUTING

Feature	Static Routing	Dynamic Routing
Configuration	Manual	Automatic
Scalability	Low	High
Adaptability	No	Yes
Overhead	Minimal	Moderate
Protocol	Not required	Required
Best For	Small networks	Large networks

---

5. ROUTING PROTOCOLS – OVERVIEW

5.1 Definition

A **routing protocol** is a set of rules that routers use to **exchange routing information** and determine **best paths**.

5.2 Routing Protocol Functions

- Network discovery
- Route advertisement
- Path selection
- Failure detection
- Convergence

---

6. ROUTING PROTOCOL CLASSIFICATION (VERY IMPORTANT)

6.1 Based on Routing Domain

6.1.1 IGP (Interior Gateway Protocol)

Used **within an autonomous system**.

Examples:

- RIP
- OSPF
- EIGRP
- IS-IS

6.1.2 EGP (Exterior Gateway Protocol)

Used **between autonomous systems**.

Example:

- **BGP**

6.2 Based on Algorithm Type

6.2.1 Distance Vector Protocols

- Share full routing table
- Periodic updates
- Simple but slower

Examples:

- RIP
- IGRP

6.2.2 Link State Protocols

- Share link-state information
- Fast convergence
- Complex

Examples:

- OSPF
- IS-IS

6.2.3 Hybrid Protocols

- Combine features of both

Example:

- **EIGRP**

6.3 Based on Classfulness

Type	Protocols
Classful	RIP v1, IGRP
Classless	RIP v2, OSPF, EIGRP, BGP

---

7. COMMON ROUTING PROTOCOLS (SUMMARY)

7.1 RIP (Routing Information Protocol)

- Distance Vector
- Metric: Hop count
- Max hops: 15
- Slow convergence

7.2 OSPF (Open Shortest Path First)

- Link State
- Metric: Cost (bandwidth-based)
- Fast convergence
- Supports areas

7.3 EIGRP (Enhanced IGRP)

- Hybrid (Cisco proprietary)
- Metric: Bandwidth, delay
- Fast convergence

7.4 BGP (Border Gateway Protocol)

- Path Vector
- Internet backbone
- Policy-based routing

---

8. METRICS IN ROUTING (ASSIGNMENT)

8.1 What is a Metric?

A **metric** is a value used by routing protocols to **determine the best path**.

8.2 Common Metrics Used

Protocol	Metric
RIP	Hop Count
OSPF	Cost
EIGRP	Bandwidth, Delay
BGP	Path attributes

8.3 Metric Example

- Route A: 2 hops
- Route B: 3 hops
  ✔ Route A selected (RIP)

8.4 Exam Point

- Lower metric = better path

---

9. CONVERGENCE (ASSIGNMENT – IMPORTANT)

9.1 Definition

Convergence is the time taken by routers to **reach a consistent routing table state after a topology change**.

9.2 Convergence Process

1. Failure detected
2. Routing updates exchanged
3. Routing tables recalculated
4. Network stabilizes

9.3 Convergence Speed Comparison

Protocol	Convergence
RIP	Slow
OSPF	Fast
EIGRP	Very Fast
BGP	Slow (policy-based)

9.4 Importance of Fast Convergence

- Reduces downtime
- Prevents packet loss
- Critical for real-time applications

---

10. ROUTING LOOP & COUNT-TO-INFINITY (EXAM ALERT 🚨)

10.1 Routing Loop

Occurs when packets circulate endlessly between routers.

10.2 Count-to-Infinity Problem

Seen in distance vector protocols when routers slowly increment hop count.

10.3 Loop Prevention Techniques

- Split Horizon
- Route Poisoning
- Hold-down Timers

---

11. REAL-WORLD CASE STUDY

Enterprise Network

- Static routes for branch default route
- OSPF within core network
- EIGRP between distribution routers
- Fast convergence ensures high availability

---

12. TROUBLESHOOTING ROUTING ISSUES

Problem	Cause
No route	Missing config
Wrong path	Metric issue
Slow recovery	Slow convergence
Routing loop	Misconfig

---

13. EXAM-ORIENTED QUICK REVISION

- Routing works at **Layer 3**
- Static routing = manual
- Dynamic routing = protocol-based
- Metrics decide best path
- Convergence = network stability time

---

14. INTERVIEW Q&A

Q: Why is dynamic routing preferred in large networks?
A: It automatically adapts to topology changes.

Q: Which routing protocol is used on the Internet?
A: BGP.

Q: Which protocol has hop count as metric?
A: RIP.

---

15. MINI CASE STUDY

ISP Network

- BGP between ISPs
- OSPF inside ISP core
- Static routes for customer edge
- Policy-based routing using metrics

🎯 FINAL EXAM KEYWORDS

- Routing table
- Static route
- Dynamic routing
- IGP / EGP
- Metric
- Convergence

🟫 SESSION 9 & 10 – ROUTING PROTOCOL IMPLEMENTATION

PART 1: ROUTING PROTOCOL IMPLEMENTATION (FOUNDATION)

---

1. What is Routing Protocol Implementation?

Definition

Routing Protocol Implementation is the process of **configuring, operating, and maintaining routing protocols on routers** so they can:

- Discover networks
- Exchange routing information
- Select best paths
- Update routing tables dynamically

---

2. Routing Table – INTERNAL STRUCTURE

A **routing table** is maintained in **RAM** and contains:

Field	Description
Destination Network	Target network
Subnet Mask	Network size
Next Hop	Where to forward
Metric	Path cost
Administrative Distance	Trust level
Protocol	Route source

Example:

```
O 192.168.2.0/24 [110/20] via 10.0.0.2, G0/0
```

---

3. Route Selection Logic (EXAM CORE)

Routers select routes based on:

1. **Longest Prefix Match**
2. **Lowest Administrative Distance (AD)**
3. **Lowest Metric**

---

4. Administrative Distance (VERY IMPORTANT)

Route Source	AD
Connected	0

Route Source	AD
Static	1
EIGRP (internal)	90
OSPF	110
RIP	120
EIGRP (external)	170

👉 **Lower AD = More Trusted**

---

5. Routing Protocol Algorithms – OVERVIEW

Protocol Type	Algorithm
Distance Vector	Bellman-Ford
Link State	Dijkstra (SPF)
Hybrid	DUAL

PART 2: RIP (Routing Information Protocol)

---

6. RIP – INTRODUCTION

Definition

RIP is a **distance-vector routing protocol** that uses **hop count** as its metric.

---

7. RIP CHARACTERISTICS

Feature	Value
Algorithm	Bellman-Ford
Metric	Hop Count
Max Hops	15
Update Interval	30 seconds
AD	120
Transport	UDP

Feature	Value
Port	520

---

8. RIP VERSIONS

RIP v1

- Classful
- No subnet mask
- Broadcast updates

RIP v2

- Classless
- Supports VLSM & CIDR
- Multicast (224.0.0.9)
- Authentication support

---

9. RIP ROUTING TABLE ENTRY

```
R 192.168.2.0/24 [120/2] via 10.0.0.2
```

---

10. RIP CONFIGURATION LOGIC (CISCO IOS)

```
router rip
version 2
network 192.168.1.0
no auto-summary
```

---

11. RIP LIMITATIONS (EXAM ALERT 🚨)

- Slow convergence
- Hop count limit
- Not scalable
- Routing loops possible

---

12. RIP LOOP PREVENTION

- Split Horizon
- Route Poisoning
- Hold-down Timers

PART 3: IGRP (Interior Gateway Routing Protocol)

---

13. IGRP – INTRODUCTION

Definition

IGRP is a **Cisco proprietary distance-vector routing protocol**, designed to overcome RIP limitations.

---

14. IGRP CHARACTERISTICS

Feature	Value
Algorithm	Distance Vector
Metric	Bandwidth, Delay
Max Hops	255
Update Interval	90 seconds
AD	100
Classful	Yes

---

15. IGRP METRIC FORMULA (EXAM IMPORTANT)

```
Metric = Bandwidth + Delay
```

(Load & Reliability optional)

---

16. IGRP LIMITATIONS

- Classful only
- No VLSM

- Obsolete (replaced by EIGRP)

---

17. IGRP STATUS

⚠ **Deprecated** – Not supported in modern IOS
(Still asked in exams for comparison)

PART 4: EIGRP (Enhanced IGRP)

---

18. EIGRP – INTRODUCTION

Definition

EIGRP is a **Cisco proprietary advanced distance-vector (hybrid) routing protocol** that provides **fast convergence and scalability**.

---

19. EIGRP KEY FEATURES

Feature	Value
Algorithm	DUAL
Metric	Bandwidth, Delay
AD	90 (internal)
Transport	RTP
Multicast	224.0.0.10
Classless	Yes

---

20. DUAL ALGORITHM (CORE EXAM TOPIC)

DUAL = Diffusing Update Algorithm

Functions:

- Loop-free routing

- Fast convergence
- Backup routes (Feasible Successor)

---

21. EIGRP ROUTE TYPES

Route Type	Meaning
Successor	Best route
Feasible Successor	Backup route

---

22. EIGRP METRIC CALCULATION

```
Metric = (Bandwidth + Delay) × 256
```

(Default K-values: K1 & K3)

---

23. EIGRP CONFIGURATION LOGIC

```
router eigrp 100
network 192.168.1.0
no auto-summary
```

---

24. EIGRP ROUTING TABLE ENTRY

```
D 192.168.2.0/24 [90/30720] via 10.0.0.2
```

---

25. EIGRP ADVANTAGES

- Fast convergence
- Low bandwidth usage
- Scalable
- Loop-free

---

26. EIGRP LIMITATIONS

- Cisco proprietary (historically)
- More complex than RIP

PART 5: OSPF (Open Shortest Path First)

---

27. OSPF – INTRODUCTION

Definition

OSPF is a **link-state, open-standard routing protocol** used in **large enterprise and ISP networks**.

---

28. OSPF CHARACTERISTICS

Feature	Value
Algorithm	Dijkstra (SPF)
Metric	Cost
AD	110
Updates	Triggered
Transport	IP (Protocol 89)
Classless	Yes

---

29. OSPF METRIC (COST)

```
Cost = Reference Bandwidth / Interface Bandwidth
```

(Default reference = 100 Mbps)

---

30. OSPF AREAS (ARCHITECTURE CORE)

```
Area 0 (Backbone)
   |
   ├── Area 1
   ├── Area 2
```

- Area 0 is mandatory
- Reduces routing overhead
- Improves scalability

---

31. OSPF ROUTER TYPES

Router Type	Description
Internal	Same area
Backbone	Area 0
ABR	Area Border Router
ASBR	External routes

---

32. OSPF LSA TYPES (EXAM FAVORITE)

LSA	Purpose
Type 1	Router LSA
Type 2	Network LSA
Type 3	Summary LSA
Type 5	External LSA

---

33. OSPF CONFIGURATION LOGIC

```
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
```

---

34. OSPF ROUTING TABLE ENTRY

```
O 192.168.2.0/24 [110/20] via 10.0.0.2
```

---

35. OSPF ADVANTAGES

- Fast convergence
- Scalable
- Vendor-neutral
- Efficient updates

---

36. OSPF DISADVANTAGES

- Complex configuration
- Higher CPU usage
- Requires planning

PART 6: COMPARISON TABLES (EXAM ESSENTIAL)

---

37. RIP vs IGRP vs EIGRP vs OSPF

Feature	RIP	IGRP	EIGRP	OSPF
Type	Distance Vector	Distance Vector	Hybrid	Link State
Metric	Hop Count	BW + Delay	BW + Delay	Cost
Convergence	Slow	Slow	Fast	Fast
Scalability	Low	Medium	High	Very High
Classless	v2 only	No	Yes	Yes
Standard	Open	Cisco	Cisco	Open

---

38. STATIC vs DYNAMIC vs ADVANCED ROUTING

Aspect	Static	RIP	OSPF
Automation	No	Yes	Yes
Speed	Fast	Slow	Fast
Complexity	Low	Low	High

PART 7: REAL-WORLD CASE STUDY

Enterprise Network

- RIP used in small legacy segments
- EIGRP used in Cisco-only campus
- OSPF used in multi-vendor core
- Static routes for ISP edge

PART 8: TROUBLESHOOTING ROUTING PROTOCOLS

Issue	Cause
No adjacency	Network mismatch
Routes missing	Auto-summary
Slow convergence	RIP
Wrong path	Metric misconfig

PART 9: EXAM-ORIENTED QUICK REVISION

- RIP max hops = **15**
- EIGRP AD = **90**
- OSPF algorithm = **Dijkstra**
- IGRP is **obsolete**
- OSPF Area 0 is backbone

PART 10: INTERVIEW Q&A

Q: Which protocol converges fastest?
A: EIGRP (very fast), OSPF (fast)

Q: Why OSPF preferred in large networks?
A: Scalability and fast convergence.

Q: Which protocol uses DUAL?
A: EIGRP.

🎯 FINAL EXAM KEYWORDS

- Bellman-Ford
- DUAL
- Dijkstra
- LSA
- Metric
- Administrative Distance

SESSION 11 – LAYER 2 SWITCHING & SPANNING TREE PROTOCOL

PART A: LAYER 2 SWITCHING

---

1. INTRODUCTION TO LAYER 2 SWITCHING

1.1 Definition

Layer 2 Switching is the process of **forwarding Ethernet frames based on MAC addresses** using **Data Link Layer (OSI Layer 2)** logic.

1.2 Purpose of Layer 2 Switching

- Fast packet forwarding
- Efficient LAN communication
- Reduced collisions
- Support for VLANs & redundancy

1.3 Switching vs Routing (Quick Recall)

Feature	Switching	Routing
OSI Layer	Layer 2	Layer 3
Address Used	MAC	IP
Device	Switch	Router
Scope	LAN	Inter-network

---

2. SWITCH ARCHITECTURE & COMPONENTS

2.1 Switch Internal Components

```
+-------------------------+
| MAC Address Table (CAM) |
+-------------------------+
| Switching Fabric        |
+-------------------------+
| Interfaces / Ports      |
+-------------------------+
```

2.2 MAC Address Table (CAM Table)

Stores:

- Source MAC address
- Port number
- VLAN ID
- Timestamp

Example:

```
MAC Address      Port
AA:BB:CC:11:22  Fa0/1
```

---

3. SWITCHING OPERATIONS (CORE LOGIC)

3.1 Frame Forwarding Process

1. Frame arrives on port
2. Source MAC learned
3. Destination MAC checked
4. Frame forwarded or flooded

3.2 MAC Address Learning

- Switch learns MAC from **source address**
- Entries age out (default 300 seconds)

3.3 Frame Handling Scenarios

Scenario	Action
Known Unicast	Forward
Unknown Unicast	Flood
Broadcast	Flood
Multicast	Flood (unless controlled)

---

4. SWITCHING METHODS

4.1 Store-and-Forward

- Entire frame received
- CRC checked
- High reliability

4.2 Cut-Through

- Forwards after reading destination MAC
- Low latency
- No error checking

4.3 Fragment-Free

- Reads first 64 bytes
- Balance of speed & reliability

---

5. COLLISION DOMAINS & BROADCAST DOMAINS

Device	Collision Domain	Broadcast Domain
Hub	1	1
Switch	One per port	1
Router	One per interface	Multiple

---

6. LAYER 2 LOOP PROBLEM (WHY STP IS NEEDED)

6.1 Layer 2 Loop – Definition

A **Layer 2 loop** occurs when **redundant links cause frames to circulate endlessly**.

6.2 Problems Caused by Loops

- Broadcast storms
- MAC table instability
- Duplicate frames
- Network outage

```
Switch A ---- Switch B
   |             |
   +---- Switch C+
```

PART B: SPANNING TREE PROTOCOL (STP)

---

7. STP – DEFINITION (THEORY ASSIGNMENT)

7.1 What is STP?

Spanning Tree Protocol (STP) is a **Layer 2 loop prevention protocol** that **logically blocks redundant paths** while maintaining **physical redundancy**.

7.2 Standard

- **IEEE 802.1D**

---

8. WHY STP IS REQUIRED (EXAM CORE)

- Prevents loops
- Ensures single active path
- Provides redundancy backup
- Protects Layer 2 networks

---

9. STP WORKING PRINCIPLE (STEP-BY-STEP)

9.1 STP Uses BPDU

BPDU – Bridge Protocol Data Unit

- Exchanged between switches
- Used to elect topology

9.2 STP Election Process

1. Root Bridge Election
2. Root Port Selection
3. Designated Port Selection
4. Blocking redundant ports

---

10. ROOT BRIDGE ELECTION (VERY IMPORTANT)

10.1 Bridge ID (BID)

```
Bridge ID = Priority + MAC Address
```

10.2 STP Priority Values (ASSIGNMENT)

- Default priority: **32768**
- Range: **0 – 65535**
- Lower value wins

Example:

```
Switch A: Priority 32768
Switch B: Priority 24576 → ROOT
```

---

11. STP PORT ROLES

Role	Description
Root Port	Best path to root
Designated Port	Forwarding port per segment
Blocking Port	Prevents loop

---

12. STP PORT STATES

State	Function
Blocking	No traffic
Listening	BPDU exchange
Learning	MAC learning
Forwarding	Data transfer
Disabled	Admin down

---

13. STP TIMERS (EXAM FAVORITE)

Timer	Default
Hello	2 sec
Forward Delay	15 sec
Max Age	20 sec

---

14. TYPES OF STP (THEORY ASSIGNMENT)

14.1 STP (802.1D)

- Original
- Slow convergence

14.2 RSTP (802.1w)

- Rapid STP
- Faster convergence

14.3 MSTP (802.1s)

- Multiple STP
- VLAN grouping

14.4 PVST+ (Cisco)

- Per-VLAN STP

14.5 Rapid PVST+

- Cisco implementation of RSTP

---

15. STP COST CALCULATION

Speed	Cost
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

---

16. STP CONFIGURATION LOGIC (CONCEPTUAL)

```
spanning-tree vlan 1 priority 24576
```

---

17. DISABLING STP – RISKS (THEORY ASSIGNMENT)

17.1 Can STP be Disabled?

Yes, but **NOT recommended**.

17.2 Risks of Disabling STP 🚨

- Broadcast storms
- MAC table instability
- Network collapse
- CPU overload
- Packet duplication

17.3 When STP May Be Disabled (Controlled)

- Lab environments
- Single-switch networks
- No redundant links

---

18. STP FAILURE & TOPOLOGY CHANGE

- Link failure detected
- BPDU recalculation
- New root path selected
- Convergence occurs

---

19. REAL-WORLD ENTERPRISE CASE STUDY

Campus Network

- Multiple access switches
- Redundant uplinks
- RSTP enabled
- Core switch set as root using priority
- Zero broadcast storms

---

20. TROUBLESHOOTING STP ISSUES

Problem	Cause
Loop	STP disabled
Wrong root	Priority misconfig
Slow convergence	Using 802.1D
Port blocking	STP design

---

21. EXAM-ORIENTED QUICK REVISION

- STP prevents Layer 2 loops
- Root bridge = lowest BID
- Default priority = 32768
- RSTP faster than STP
- Disabling STP is dangerous

---

22. INTERVIEW Q&A

Q: Why is STP required in switches but not routers?
A: Routers operate at Layer 3 and do not forward broadcasts.

Q: What happens if STP is disabled?
A: Network loops cause broadcast storms.

Q: Which STP version converges fastest?
A: RSTP (802.1w).

---

23. MINI CASE STUDY

Data Center Switch Loop

- Redundant fiber links
- STP disabled accidentally
- Broadcast storm occurred
- Network outage
- STP re-enabled with RSTP

🎯 FINAL EXAM KEYWORDS

- CAM table
- Broadcast storm
- Bridge ID
- Root bridge
- BPDU
- STP priority

🟫 SESSION 11 – LAYER 2 SWITCHING & SPANNING TREE PROTOCOL

PART A: LAYER 2 SWITCHING

---

1. INTRODUCTION TO LAYER 2 SWITCHING

1.1 Definition

Layer 2 Switching is the process of **forwarding Ethernet frames based on MAC addresses** using **Data Link Layer (OSI Layer 2)** logic.

1.2 Purpose of Layer 2 Switching

- Fast packet forwarding
- Efficient LAN communication
- Reduced collisions
- Support for VLANs & redundancy

1.3 Switching vs Routing (Quick Recall)

Feature	Switching	Routing
OSI Layer	Layer 2	Layer 3
Address Used	MAC	IP
Device	Switch	Router

Feature	Switching	Routing
Scope	LAN	Inter-network

---

2. SWITCH ARCHITECTURE & COMPONENTS

2.1 Switch Internal Components

```
+-------------------------+
| MAC Address Table (CAM) |
+-------------------------+
| Switching Fabric        |
+-------------------------+
| Interfaces / Ports      |
+-------------------------+
```

2.2 MAC Address Table (CAM Table)

Stores:

- Source MAC address
- Port number
- VLAN ID
- Timestamp

Example:

```
MAC Address       Port
AA:BB:CC:11:22  Fa0/1
```

---

3. SWITCHING OPERATIONS (CORE LOGIC)

3.1 Frame Forwarding Process

1. Frame arrives on port
2. Source MAC learned
3. Destination MAC checked
4. Frame forwarded or flooded

3.2 MAC Address Learning

- Switch learns MAC from **source address**
- Entries age out (default 300 seconds)

3.3 Frame Handling Scenarios

Scenario	Action
Known Unicast	Forward
Unknown Unicast	Flood
Broadcast	Flood
Multicast	Flood (unless controlled)

---

4. SWITCHING METHODS

4.1 Store-and-Forward

- Entire frame received
- CRC checked
- High reliability

4.2 Cut-Through

- Forwards after reading destination MAC
- Low latency
- No error checking

4.3 Fragment-Free

- Reads first 64 bytes
- Balance of speed & reliability

---

5. COLLISION DOMAINS & BROADCAST DOMAINS

Device	Collision Domain	Broadcast Domain
Hub	1	1
Switch	One per port	1
Router	One per interface	Multiple

---

6. LAYER 2 LOOP PROBLEM (WHY STP IS NEEDED)

6.1 Layer 2 Loop – Definition

A **Layer 2 loop** occurs when **redundant links cause frames to circulate endlessly**.

6.2 Problems Caused by Loops

- Broadcast storms
- MAC table instability
- Duplicate frames
- Network outage

```
Switch A ---- Switch B
   |             |
   +---- Switch C+
```

PART B: SPANNING TREE PROTOCOL (STP)

---

7. STP – DEFINITION (THEORY ASSIGNMENT)

7.1 What is STP?

Spanning Tree Protocol (STP) is a **Layer 2 loop prevention protocol** that **logically blocks redundant paths** while maintaining **physical redundancy**.

7.2 Standard

- **IEEE 802.1D**

---

8. WHY STP IS REQUIRED (EXAM CORE)

- Prevents loops
- Ensures single active path
- Provides redundancy backup
- Protects Layer 2 networks

---

9. STP WORKING PRINCIPLE (STEP-BY-STEP)

9.1 STP Uses BPDU

BPDU – Bridge Protocol Data Unit

- Exchanged between switches
- Used to elect topology

9.2 STP Election Process

1. Root Bridge Election
2. Root Port Selection
3. Designated Port Selection
4. Blocking redundant ports

---

10. ROOT BRIDGE ELECTION (VERY IMPORTANT)

10.1 Bridge ID (BID)

```
Bridge ID = Priority + MAC Address
```

10.2 STP Priority Values (ASSIGNMENT)

- Default priority: **32768**
- Range: **0 – 65535**
- Lower value wins

Example:

```
Switch A: Priority 32768
Switch B: Priority 24576 → ROOT
```

---

11. STP PORT ROLES

Role	Description
Root Port	Best path to root
Designated Port	Forwarding port per segment
Blocking Port	Prevents loop

---

12. STP PORT STATES

State	Function
Blocking	No traffic
Listening	BPDU exchange
Learning	MAC learning
Forwarding	Data transfer
Disabled	Admin down

---

13. STP TIMERS (EXAM FAVORITE)

Timer	Default
Hello	2 sec
Forward Delay	15 sec
Max Age	20 sec

---

14. TYPES OF STP (THEORY ASSIGNMENT)

14.1 STP (802.1D)

- Original
- Slow convergence

14.2 RSTP (802.1w)

- Rapid STP
- Faster convergence

14.3 MSTP (802.1s)

- Multiple STP
- VLAN grouping

14.4 PVST+ (Cisco)

- Per-VLAN STP

14.5 Rapid PVST+

- Cisco implementation of RSTP

---

15. STP COST CALCULATION

Speed	Cost
10 Mbps	100
100 Mbps	19
1 Gbps	4

Speed	Cost
10 Gbps	2

---

16. STP CONFIGURATION LOGIC (CONCEPTUAL)

```
spanning-tree vlan 1 priority 24576
```

---

17. DISABLING STP – RISKS (THEORY ASSIGNMENT)

17.1 Can STP be Disabled?

Yes, but **NOT recommended**.

17.2 Risks of Disabling STP 🚨

- Broadcast storms
- MAC table instability
- Network collapse
- CPU overload
- Packet duplication

17.3 When STP May Be Disabled (Controlled)

- Lab environments
- Single-switch networks
- No redundant links

---

18. STP FAILURE & TOPOLOGY CHANGE

- Link failure detected
- BPDU recalculation
- New root path selected
- Convergence occurs

---

19. REAL-WORLD ENTERPRISE CASE STUDY

Campus Network

- Multiple access switches
- Redundant uplinks
- RSTP enabled
- Core switch set as root using priority
- Zero broadcast storms

---

20. TROUBLESHOOTING STP ISSUES

Problem	Cause
Loop	STP disabled
Wrong root	Priority misconfig
Slow convergence	Using 802.1D
Port blocking	STP design

---

21. EXAM-ORIENTED QUICK REVISION

- STP prevents Layer 2 loops
- Root bridge = lowest BID
- Default priority = 32768
- RSTP faster than STP
- Disabling STP is dangerous

---

22. INTERVIEW Q&A

Q: Why is STP required in switches but not routers?
A: Routers operate at Layer 3 and do not forward broadcasts.

Q: What happens if STP is disabled?
A: Network loops cause broadcast storms.

Q: Which STP version converges fastest?
A: RSTP (802.1w).

---

23. MINI CASE STUDY

Data Center Switch Loop

- Redundant fiber links
- STP disabled accidentally
- Broadcast storm occurred
- Network outage
- STP re-enabled with RSTP

🎯 FINAL EXAM KEYWORDS

- CAM table
- Broadcast storm
- Bridge ID
- Root bridge
- BPDU
- STP priority

🟫 SESSION 14 – NAT, IPv6 & WAN TECHNOLOGIES

PART A: NETWORK ADDRESS TRANSLATION (NAT)

---

1. NETWORK ADDRESS TRANSLATION – INTRODUCTION

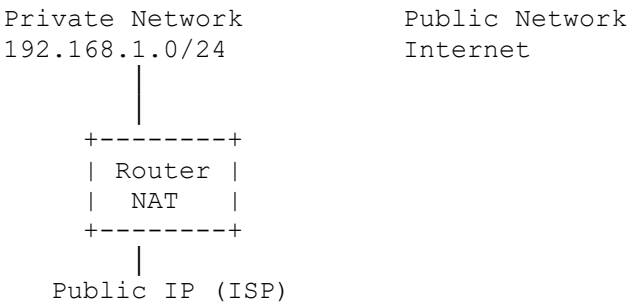
1.1 Definition

Network Address Translation (NAT) is a **network-layer (Layer 3) mechanism** that **modifies IP address information in packet headers** while traffic is in transit between networks.

1.2 Why NAT is Required

- IPv4 address exhaustion
- Allows use of **private IP addresses**
- Improves basic security
- Enables multiple devices to share a single public IP

1.3 NAT Placement (Architecture)



---

2. IP ADDRESS TYPES USED IN NAT

Type	Description
Inside Local	Private IP (LAN)
Inside Global	Public IP representing inside host
Outside Local	Internal view of external host
Outside Global	Actual public IP of external host

---

3. TYPES OF NAT (VERY IMPORTANT)

3.1 Static NAT

One-to-one mapping between private and public IP.

```
192.168.1.10 ↔ 203.0.113.10
```

Use case:

- Public servers (Web, Mail)

Pros:

- Predictable
  Cons:
- Wastes public IPs

3.2 Dynamic NAT

- Maps private IPs to a **pool of public IPs**
- Allocation is temporary

```
Pool: 203.0.113.10 – 203.0.113.20
```

3.3 PAT (Port Address Translation) / NAT Overload

- **Many-to-one**
- Uses **TCP/UDP port numbers**

```
192.168.1.10:1025 → 203.0.113.5:50001
```

✔ **Most commonly used NAT**

3.4 NAT Comparison Table

Feature	Static	Dynamic	PAT
Mapping	1:1	Many:Many	Many:1
IP Efficiency	Low	Medium	High
Usage	Servers	Enterprises	Home/ISP

---

4. NAT WORKING (PACKET FLOW)

```
PC (192.168.1.10)
   ↓
NAT Router
   ↓ (Translate IP & Port)
203.0.113.5:50001
   ↓
Internet Server
```

---

5. NAT CONFIGURATION LOGIC (CISCO – CONCEPTUAL)

```
ip nat inside
ip nat outside
ip nat inside source list 1 interface g0/0 overload
```

---

6. ADVANTAGES & DISADVANTAGES OF NAT

Advantages

- Conserves IPv4 addresses
- Hides internal network
- Easy to deploy

Disadvantages

- Breaks end-to-end connectivity
- Complicates VoIP & IPsec
- Adds processing overhead

---

7. REAL-WORLD NAT USE CASES

- Home broadband routers
- Enterprise edge routers
- ISP customer aggregation

---

8. EXAM & INTERVIEW POINTS (NAT)

- NAT works at **Layer 3**
- PAT = NAT Overload
- Static NAT used for servers

PART B: INTERNET PROTOCOL VERSION 6 (IPv6)

---

9. IPv6 – INTRODUCTION

9.1 Definition

IPv6 is the **next-generation Internet Protocol** designed to replace IPv4, providing **128-bit addressing**.

9.2 Why IPv6?

- IPv4 address exhaustion
- Larger address space
- Simplified header
- Built-in security
- Better mobility & scalability

---

10. IPv6 ADDRESS STRUCTURE

10.1 IPv6 Format

```
2001:0db8:85a3:0000:0000:8a2e:0370:7334
```

- 128 bits
- Hexadecimal
- 8 groups of 16 bits

10.2 IPv6 Address Shortening Rules

1. Remove leading zeros
2. Replace consecutive zeros with `::` (once only)

Example:

```
2001:db8::1
```

---

11. TYPES OF IPv6 ADDRESSES

11.1 Unicast

- One-to-one communication

11.2 Multicast

- One-to-many
- Replaces broadcast

11.3 Anycast

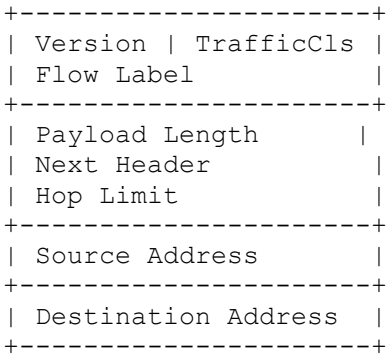
- One-to-nearest
- Used in CDN, DNS

11.4 IPv6 Address Scope Table

Type	Prefix
Global Unicast	2000::/3
Link-Local	FE80::/10
Multicast	FF00::/8
Loopback	::1

---

12. IPv6 HEADER STRUCTURE (SIMPLIFIED)



✔ No checksum
✔ No fragmentation by routers

---

13. IPv6 vs IPv4 (COMPARISON)

Feature	IPv4	IPv6
Address Size	32-bit	128-bit
Notation	Decimal	Hex
Broadcast	Yes	No
NAT	Required	Not required
Security	Optional	Built-in IPsec

---

14. IPv6 TRANSITION MECHANISMS

14.1 Dual Stack

- IPv4 + IPv6 simultaneously

14.2 Tunneling

- IPv6 over IPv4 (6to4)

14.3 Translation

- NAT64

---

15. REAL-WORLD IPv6 USE CASES

- Mobile networks (5G)
- Cloud providers
- IoT environments

---

16. EXAM & INTERVIEW POINTS (IPv6)

- IPv6 = 128 bits
- No broadcast
- Link-local always exists
- NAT not required

PART C: WAN TECHNOLOGIES

---

17. WAN – INTRODUCTION

17.1 Definition

A **Wide Area Network (WAN)** connects **geographically distant networks** using **service provider infrastructure**.

---

18. WAN CHARACTERISTICS

- Large geographic coverage
- Higher latency than LAN
- Lower bandwidth
- Carrier-managed links

---

19. WAN ARCHITECTURE (BASIC)

```
LAN → Router → WAN Cloud → Router → LAN
```

---

20. CLASSIC WAN TECHNOLOGIES

20.1 Leased Line

- Dedicated connection
- Constant bandwidth

Examples:

- T1 (1.544 Mbps)
- E1 (2.048 Mbps)

20.2 ISDN

- Digital circuit-switched
- BRI / PRI
- Largely obsolete

20.3 Frame Relay (Legacy)

- Packet-switched
- Virtual Circuits (PVC)
- DLCI identifiers

20.4 ATM

- Cell-based (53 bytes)
- QoS support
- Expensive

---

21. MODERN WAN TECHNOLOGIES

21.1 MPLS (Multi-Protocol Label Switching)

- Label-based forwarding
- High performance
- ISP backbone standard

21.2 Metro Ethernet

- Ethernet over WAN
- High bandwidth
- Cost effective

21.3 Broadband WAN

- DSL
- Cable
- Fiber

21.4 SD-WAN (Modern Enterprise)

- Software-defined
- Centralized control
- Policy-based routing

---

22. WAN ENCAPSULATION PROTOCOLS

Protocol	Use
HDLC	Cisco default
PPP	Authentication (PAP/CHAP)
Frame Relay	Virtual circuits
MPLS	Label switching

---

23. WAN DEVICES

- Router
- CSU/DSU
- Modem
- WAN switch (ISP side)

---

24. WAN SECURITY CONSIDERATIONS

- VPN (IPsec, SSL)
- Encryption
- Authentication
- Firewall integration

---

25. REAL-WORLD ENTERPRISE CASE STUDY

Enterprise WAN

- MPLS for branch connectivity
- NAT at edge
- IPv6 enabled core
- SD-WAN for redundancy

---

26. TROUBLESHOOTING WAN ISSUES

Issue	Cause
High latency	Long distance
Link down	Carrier issue
Packet loss	Congestion
NAT failure	Wrong config

---

27. EXAM-ORIENTED QUICK REVISION

- NAT conserves IPv4
- PAT = many-to-one
- IPv6 has no broadcast
- MPLS uses labels
- SD-WAN = software control

---

28. INTERVIEW Q&A

Q: Why NAT is not required in IPv6?
A: Because IPv6 provides a huge address space.

Q: Which NAT type is most used?
A: PAT (NAT Overload).

Q: Which WAN technology is ISP backbone standard?
A: MPLS.

---

29. MINI CASE STUDY

ISP Network

- IPv4 NAT for customers
- Dual-stack IPv6
- MPLS core
- SD-WAN for enterprises

🎯 FINAL EXAM KEYWORDS

- NAT Overload
- Inside/Outside IP
- IPv6 Unicast
- MPLS
- SD-WAN
- WAN encapsulation

🟫 SESSION 15 – ENTERPRISE WAN PROTOCOLS & BGP

PART A: POINT-TO-POINT PROTOCOL (PPP)

---

1. PPP – INTRODUCTION

1.1 Definition

Point-to-Point Protocol (PPP) is a **Layer 2 WAN encapsulation protocol** used to establish **direct links between two network nodes** over serial links.

1.2 Why PPP is Needed

- Replaces basic HDLC with **authentication**
- Supports **multiple Layer 3 protocols**
- Error detection & link quality monitoring
- Widely used in **ISP and enterprise WANs**

---

2. PPP ARCHITECTURE

```
Network Layer (IP, IPv6)
        ↑
   NCP (IPCP, IPv6CP)
        ↑
   LCP (Link Control Protocol)
        ↑
   PPP Encapsulation
        ↑
 Physical Layer
```

---

3. PPP COMPONENTS (VERY IMPORTANT)

3.1 LCP – Link Control Protocol

Responsible for:

- Link establishment
- Authentication negotiation
- Link maintenance
- Link termination

3.2 NCP – Network Control Protocol

- Configures Layer 3 protocols
- Examples:
    - IPCP (IPv4)
    - IPv6CP (IPv6)

---

4. PPP LINK ESTABLISHMENT PHASES

1. **Link Establishment (LCP)**
2. **Authentication (Optional)**
3. **Network Layer Configuration (NCP)**
4. **Data Transfer**
5. **Link Termination**

---

5. PPP FRAME FORMAT

```
| Flag | Address | Control | Protocol | Data | FCS | Flag |
```

- Flag = 0x7E
- Protocol field identifies IP, LCP, NCP

---

6. PPP AUTHENTICATION METHODS

6.1 PAP (Password Authentication Protocol)

- Two-way handshake
- Password sent in **plain text**
- Insecure

6.2 CHAP (Challenge Handshake Authentication Protocol)

- Three-way handshake
- Encrypted password
- Periodic re-authentication
- **More secure**

```
Server → Challenge
Client → Hash Response
Server → Success/Failure
```

---

7. PPP CONFIGURATION LOGIC (CISCO – CONCEPTUAL)

```
interface s0/0
encapsulation ppp
ppp authentication chap
```

---

8. PPP ADVANTAGES & LIMITATIONS

Advantages

- Secure authentication
- Multi-protocol support
- Error detection

Limitations

- Point-to-point only
- Legacy in modern WANs

---

9. REAL-WORLD PPP USE CASES

- ISP leased lines
- Legacy serial WANs
- Backup links

PART B: MULTILINK PPP (MLPPP)

---

10. MLPPP – INTRODUCTION

10.1 Definition

Multilink PPP (MLPPP) allows **multiple physical links to be bundled into one logical link**, increasing **bandwidth and redundancy**.

---

11. WHY MLPPP?

- Bandwidth aggregation
- Load balancing
- Link redundancy
- Cost-effective WAN scaling

---

12. MLPPP ARCHITECTURE

```
Logical Bundle (MLPPP)
   ├── Serial 0/0
   ├── Serial 0/1
   └── Serial 0/2
```

---

13. MLPPP WORKING

- Packets are **fragmented**
- Sent across multiple links
- Reassembled at destination

---

14. MLPPP CONFIGURATION LOGIC

```
interface multilink 1
ip address 10.0.0.1 255.255.255.252

interface s0/0
ppp multilink
ppp multilink group 1
```

---

15. MLPPP ADVANTAGES & DISADVANTAGES

Advantages

- Increased throughput
- Automatic failover

Disadvantages

- Complexity

- Latency variation issues

---

16. MLPPP ENTERPRISE USE CASE

- Branch offices using multiple low-cost leased lines
- Bandwidth upgrade without new circuit

PART C: PPPoE (PPP over Ethernet)

---

17. PPPoE – INTRODUCTION

17.1 Definition

PPPoE encapsulates **PPP frames inside Ethernet frames**, commonly used in **DSL broadband connections**.

---

18. WHY PPPoE?

- ISP authentication over Ethernet
- Session management
- Per-user billing
- Supports PAP/CHAP

---

19. PPPoE ARCHITECTURE

```
Customer Router → DSL Modem → ISP Access Concentrator
```

---

20. PPPoE PHASES

20.1 Discovery Phase

- PADI

- PADO
- PADR
- PADS

20.2 Session Phase

- PPP negotiation
- Authentication
- Data transfer

---

21. PPPoE FRAME FLOW

```
PPP → Ethernet → ISP
```

---

22. PPPoE CONFIGURATION LOGIC (CISCO – CONCEPTUAL)

```
interface dialer 1
encapsulation ppp
ppp chap hostname user
ppp chap password pass
```

---

23. PPPoE ADVANTAGES & LIMITATIONS

Advantages

- Scalable ISP access
- Secure authentication

Limitations

- MTU overhead
- Slight performance loss

---

24. REAL-WORLD PPPoE USE CASE

- Home broadband
- SME internet connections

PART D: GRE TUNNELING

---

25. GRE – INTRODUCTION

25.1 Definition

Generic Routing Encapsulation (GRE) is a **Layer 3 tunneling protocol** that **encapsulates packets inside IP packets**, enabling **virtual point-to-point links**.

---

26. WHY GRE?

- Connect remote networks
- Carry multicast & routing protocols
- Logical topology over Internet

---

27. GRE ARCHITECTURE

```
LAN → GRE Tunnel → Internet → GRE Tunnel → LAN
```

---

28. GRE PACKET STRUCTURE

```
| Outer IP | GRE Header | Inner IP | Payload |
```

---

29. GRE CHARACTERISTICS

Feature	Value
Security	None
Encryption	No
Multicast	Yes

Feature	Value
Protocol Support	Any

---

30. GRE CONFIGURATION LOGIC

```
interface tunnel 0
ip address 10.10.10.1 255.255.255.252
tunnel source g0/0
tunnel destination 203.0.113.2
```

---

31. GRE LIMITATIONS (SECURITY ALERT 🚨)

- No encryption
- Requires IPsec for security

---

32. GRE REAL-WORLD USE CASE

- Dynamic routing over Internet
- Site-to-site overlay networks

PART E: eBGP (SINGLE-HOMED BRANCH)

---

33. BGP – INTRODUCTION (RECAP)

Definition

Border Gateway Protocol (BGP) is a **path-vector routing protocol** used for **inter-AS routing**.

---

34. eBGP vs iBGP

Feature	eBGP	iBGP
AS	Different	Same

Feature	eBGP	iBGP
Distance	External	Internal
AD	20	200

---

35. SINGLE-HOMED BRANCH – DEFINITION

A **single-homed branch** connects to **only one ISP** using **one BGP session**.

```
Branch Router → ISP Router → Internet
```

---

36. WHY eBGP IN ENTERPRISE?

- ISP-controlled routing
- Public IP advertisement
- Policy control
- Scalability

---

37. eBGP WORKING LOGIC

1. TCP session (Port 179)
2. BGP OPEN message
3. KEEPALIVE exchange
4. UPDATE messages
5. Routing table population

---

38. BGP ATTRIBUTES (EXAM CORE)

Attribute	Purpose
AS_PATH	Loop prevention
NEXT_HOP	Reachability
LOCAL_PREF	Path preference
MED	ISP influence

---

39. eBGP CONFIGURATION LOGIC (CONCEPTUAL)

```
router bgp 65001
neighbor 203.0.113.1 remote-as 65000
network 198.51.100.0 mask 255.255.255.0
```

---

40. SINGLE-HOMED eBGP CHARACTERISTICS

- Simple configuration
- No redundancy
- Relies on ISP availability
- Often combined with default route

---

41. eBGP ADVANTAGES & DISADVANTAGES

Advantages

- Internet reachability
- Policy control

Disadvantages

- Complexity
- Slow convergence
- Requires public AS/IP

---

42. REAL-WORLD ENTERPRISE CASE STUDY

Branch Office WAN

- PPP link to ISP
- GRE tunnel to HQ
- eBGP for Internet routing
- IPsec added for security
- Reliable enterprise connectivity

PART F: COMPARISON TABLES (EXAM ESSENTIAL)

---

43. PPP vs MLPPP vs PPPoE

Feature	PPP	MLPPP	PPPoE
Link Type	Serial	Multiple Serial	Ethernet
Bandwidth	Single	Aggregated	Single
ISP Usage	Legacy	Rare	Common

---

44. GRE vs IPsec

Feature	GRE	IPsec
Encryption	No	Yes
Multicast	Yes	No
Routing Protocols	Yes	No

---

45. STATIC ROUTE vs eBGP

Feature	Static	eBGP
Scalability	Low	High
Automation	No	Yes
ISP Use	No	Yes

PART G: TROUBLESHOOTING & SECURITY

---

46. COMMON ISSUES

Problem	Cause
PPP auth failure	CHAP mismatch
MLPPP reorder	Latency mismatch
PPPoE MTU	Fragmentation
GRE down	Source/destination error
BGP idle	AS mismatch

---

47. SECURITY BEST PRACTICES

- Use CHAP over PAP
- GRE + IPsec
- BGP prefix filtering
- MD5 authentication for BGP

---

48. EXAM-ORIENTED QUICK REVISION

- PPP = L2 WAN protocol
- MLPPP = bandwidth aggregation
- PPPoE = broadband access
- GRE = tunneling, no security
- eBGP = inter-AS routing

---

49. INTERVIEW Q&A

Q: Why GRE is combined with IPsec?
A: GRE provides tunneling; IPsec provides security.

Q: Which PPP authentication is secure?
A: CHAP.

Q: Why BGP is used instead of OSPF with ISP?
A: OSPF is IGP; BGP is EGP.

🎯 FINAL EXAM KEYWORDS

- LCP / NCP
- CHAP
- Multilink
- PPPoE Discovery
- GRE Tunnel
- eBGP Single-homed

SESSION 16 – SOFTWARE DEFINED NETWORKING

---

1. INTRODUCTION TO SDN (Software Defined Networking)

1.1 Definition

Software Defined Networking (SDN) is a **network architecture paradigm** that **decouples the control plane from the data plane**, enabling **centralized, programmable, and automated network control**.

1.2 Traditional Networking vs SDN

Traditional Network

- Control plane & data plane tightly coupled
- Configuration device-by-device
- Vendor-specific hardware
- Limited automation

SDN Network

- Centralized control
- Programmable via software
- Hardware abstraction
- Automation & orchestration

1.3 Why SDN Was Introduced (Need)

- Complex network management
- Poor scalability in data centers
- Slow provisioning
- Vendor lock-in
- Need for automation & agility

1.4 Evolution of SDN (Brief History)

- Early ideas: Active Networks
- 2008: OpenFlow (Stanford)
- Growth of cloud & data centers
- Adoption by Google, Amazon, Microsoft
- Foundation: **ONF (Open Networking Foundation)**

1.5 Key Characteristics of SDN

- Separation of planes
- Centralized control
- Programmability
- Network abstraction
- Vendor neutrality (open standards)

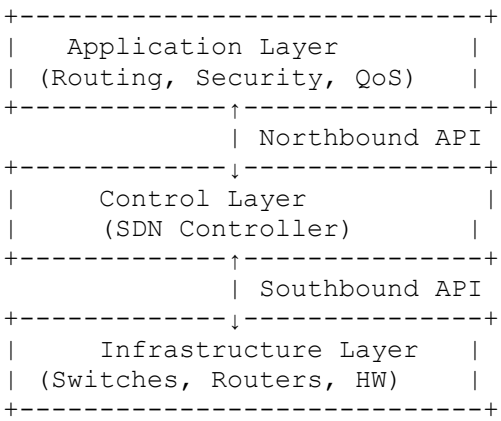
1.6 Real-World SDN Examples

- Google B4 WAN
- AWS VPC networking
- Azure Virtual Networks
- OpenDaylight, ONOS controllers

---

2. SDN ARCHITECTURE (CORE TOPIC)

2.1 SDN Architectural Overview



2.2 SDN Layers Explained

2.2.1 Application Layer

Role:

- Network applications
- Business logic
- Policy definition

Examples:

- Load balancing
- Firewall
- Traffic engineering
- Network monitoring

2.2.2 Control Layer (Controller)

Role:

- Brain of the network
- Maintains global network view
- Makes forwarding decisions

Popular Controllers:

- OpenDaylight
- ONOS
- Ryu
- Floodlight

2.2.3 Infrastructure Layer

Role:

- Forwards packets

- Executes controller instructions

Devices:

- OpenFlow switches
- White-box switches
- Virtual switches (Open vSwitch)

2.3 SDN Interfaces (APIs)

2.3.1 Southbound APIs

- Communication between controller & devices

Examples:

- OpenFlow
- NETCONF
- OVSDB
- P4 Runtime

2.3.2 Northbound APIs

- Communication between controller & applications

Examples:

- REST APIs
- gRPC

2.3.3 East-West APIs

- Communication between controllers
- Used in distributed SDN

2.4 SDN Control vs Data Plane (Exam Favorite)

Plane	Function
Control Plane	Decision making
Data Plane	Packet forwarding

---

3. SCALABILITY IN SDN

3.1 Definition

Scalability is the ability of SDN to **handle increasing network size, devices, and traffic without performance degradation**.

3.2 Scalability Challenges in Traditional Networks

- Distributed control
- Configuration complexity
- Hardware limitations

3.3 How SDN Achieves Scalability

3.3.1 Centralized Network View

- Global topology awareness
- Efficient resource utilization

3.3.2 Distributed Controllers

```
Controller Cluster
 ├── Controller 1
 ├── Controller 2
 └── Controller 3
```

- Load sharing
- Horizontal scaling

3.3.3 Flow Aggregation

- Reduce flow table entries
- Improve switch performance

3.3.4 Hierarchical SDN

- Local controllers
- Global orchestrator

3.4 Real-World Scalability Example

- Google SDN WAN handles **millions of flows**
- Data center fabrics scale dynamically

3.5 Exam Points (Scalability)

- SDN uses controller clustering
- Flow aggregation improves scalability "

---

4. RELIABILITY IN SDN

4.1 Definition

Reliability is the ability of SDN to **maintain continuous network operation despite failures**.

4.2 Reliability Issues in SDN

- Single controller failure
- Link or switch failure
- Controller-switch communication failure

4.3 Reliability Mechanisms in SDN

4.3.1 Controller Redundancy

- Active-Active
- Active-Standby

4.3.2 Fast Failover Groups (OpenFlow)

- Pre-configured backup paths
- Local switch decision

4.3.3 Path Diversity

- Multiple paths available
- Dynamic rerouting

4.3.4 Heartbeats & Monitoring

- Continuous health checks
- Failure detection

4.4 Reliability vs Availability

Aspect	Reliability	Availability
Focus	Failure handling	Uptime

Aspect	Reliability	Availability
SDN Support	High	High

4.5 Exam Focus (Reliability)

- Controller clustering improves reliability
- Failover groups reduce recovery time

---

5. CONSISTENCY IN SDN

5.1 Definition

Consistency ensures that **all network devices have a coherent and correct view of network policies and state**.

5.2 Why Consistency Matters

- Prevents configuration conflicts
- Avoids routing loops
- Ensures policy correctness

5.3 Consistency Models

5.3.1 Strong Consistency

- All devices updated simultaneously
- Slower but safer

5.3.2 Eventual Consistency

- Updates propagate over time
- Faster but temporary inconsistencies

5.4 Techniques to Achieve Consistency

5.4.1 Transaction-Based Updates

- All-or-nothing policy deployment

5.4.2 Versioning & Rollback

- Safe configuration changes

5.4.3 Distributed Datastores

- Raft / Paxos algorithms
- Used in controller clusters

5.5 CAP Theorem & SDN (Advanced)

Property	Meaning
Consistency	Same view everywhere
Availability	Always responds
Partition Tolerance	Survives failures

SDN often balances **Consistency vs Availability**

5.6 Exam Points (Consistency)

- Strong vs eventual consistency
- Distributed controllers need synchronization

6. OPPORTUNITIES OF SDN

6.1 Network Automation

- Zero-touch provisioning
- Script-based management

6.2 Cost Reduction

- Commodity hardware
- Reduced OPEX

6.3 Innovation & Programmability

- Rapid service deployment
- Custom network applications

6.4 Cloud & Data Center Enablement

- Virtual networking
- Multi-tenant isolation

6.5 Security Opportunities

- Centralized policy enforcement
- Dynamic threat mitigation

6.6 SDN Use Cases

- Data centers
- ISP networks
- Campus networks
- Network slicing (5G)

---

7. CHALLENGES OF SDN

7.1 Technical Challenges

- Controller scalability
- Latency between controller & switches
- Flow table limitations

7.2 Security Challenges

- Controller as attack target
- API security
- Insider threats

7.3 Operational Challenges

- Skill gap
- Migration from legacy networks
- Debugging complexity

7.4 Standardization Challenges

- Vendor implementations vary
- Interoperability issues

7.5 Performance Challenges

- Control plane bottlenecks
- Real-time responsiveness

7.6 Exam Alert 🚨

- SDN is not plug-and-play
- Requires careful design

---

8. SDN vs TRADITIONAL NETWORKING

Feature	Traditional	SDN
Control	Distributed	Centralized
Programmability	Low	High
Automation	Limited	Extensive
Scalability	Moderate	High
Cost	High	Lower

---

9. MINI CASE STUDY

Enterprise Data Center

- Legacy VLAN-based network
- Migrated to SDN fabric
- Central controller
- Automated provisioning
- Reduced downtime & cost

---

10. EXAM-ORIENTED QUICK REVISION

- SDN decouples control & data plane
- Controller = brain
- OpenFlow = southbound API
- Scalability via controller clustering
- Reliability via redundancy
- Consistency via distributed databases

---

11. INTERVIEW Q&A

Q: What is the main idea of SDN?
A: Centralized, programmable network control.

Q: What happens if SDN controller fails?
A: Redundant controllers take over.

Q: Why consistency is important in SDN?
A: To avoid incorrect forwarding and policy conflicts.

🎯 FINAL EXAM KEYWORDS

- Control plane
- Data plane
- SDN controller
- OpenFlow
- Scalability
- Reliability
- Consistency

🟫 SESSION 17 – VIRTUAL NETWORKING & MODERN ENTERPRISE USE CASES

---

1. VIRTUAL NETWORKING

1.1 Definition

Virtual Networking is the abstraction of physical network resources (switches, routers, links) into **software-defined logical networks**, enabling **flexible, scalable, and isolated communication** between virtual machines, containers, and services.

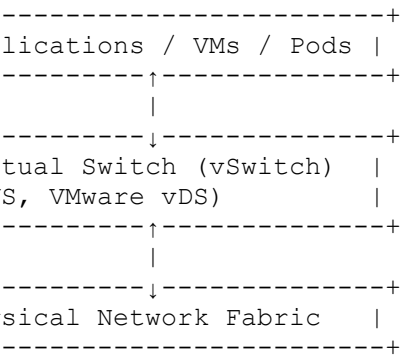
1.2 Why Virtual Networking is Required

- Cloud & virtualization adoption
- Multi-tenant environments
- Dynamic workload mobility
- Rapid provisioning
- Infrastructure abstraction

1.3 Evolution of Virtual Networking

- VLAN-based segmentation
- Virtual switches (vSwitch)
- Overlay networks (VXLAN, GRE)
- SDN-controlled fabrics
- Cloud-native networking

1.4 Virtual Networking Architecture



1.5 Key Components of Virtual Networking

Component	Role
Virtual Switch	L2 forwarding
Virtual Router	L3 routing
Overlay Tunnel	Logical isolation
Controller	Centralized control
Underlay Network	Physical transport

1.6 Virtual Switches (vSwitch)

Definition

A **virtual switch** is a **software-based Layer 2 switch** inside a hypervisor.

Examples

- Open vSwitch (OVS)
- VMware vSwitch / vDS
- Linux Bridge

Functions

- MAC learning
- VLAN tagging
- Port mirroring
- QoS enforcement

1.7 Overlay Networking (CORE TOPIC)

Definition

Overlay networking creates **logical networks over existing physical networks** using **tunneling protocols**.

```
VM → VXLAN Tunnel → Physical Network → VXLAN Tunnel → VM
```

Common Overlay Protocols

Protocol	Identifier
VXLAN	24-bit VNI
GRE	Tunnel endpoints
NVGRE	Tenant ID
Geneve	Flexible TLVs

1.8 VXLAN (Exam Important)

- 24-bit VNI → ~16 million networks

- Uses UDP (Port 4789)
- Solves VLAN scalability limits

1.9 Virtual Networking in Cloud

AWS

- VPC
- Subnets
- Route tables
- Internet/NAT Gateways

Azure

- Virtual Network (VNet)
- NSGs
- UDRs

GCP

- Global VPC
- Subnet-based routing

1.10 Advantages & Limitations

Advantages

- Scalability
- Isolation
- Automation
- Cloud readiness

Limitations

- Complexity
- Troubleshooting difficulty
- Overlay overhead

1.11 Exam Keywords

- Overlay
- Underlay
- vSwitch
- VXLAN
- Network abstraction

---

2. NETWORK ACCESS CONTROL (NAC)

2.1 Definition

Network Access Control (NAC) is a **security mechanism** that **controls device and user access to the network** based on **identity, posture, and policy**.

2.2 Objectives of NAC

- Prevent unauthorized access
- Enforce security policies
- Device compliance checks
- Network segmentation

2.3 NAC Architecture

```
Endpoint → Switch/AP → NAC Server → Policy Decision
```

2.4 NAC Components

Component	Function
Supplicant	Client software
Authenticator	Switch/AP
Authentication Server	RADIUS/NAC
Policy Engine	Access decision

2.5 NAC Authentication Methods

- 802.1X
- MAC Authentication Bypass (MAB)
- Web Authentication

2.6 802.1X Authentication Flow (EXAM CORE)

```
Client → EAPOL → Switch → RADIUS → NAC Server
```

Steps:

1. Identity request
2. Credential validation
3. Policy decision
4. Access granted/denied

2.7 Posture Assessment

- OS version
- Antivirus status
- Patch level
- Compliance check

2.8 NAC Deployment Models

Model	Description
Inline	Traffic passes through NAC
Out-of-band	NAC controls switches
Agent-based	Client software
Agentless	Web/MAC-based

2.9 NAC Actions

- Full access
- Limited (quarantine VLAN)
- Guest access
- Deny access

2.10 Enterprise NAC Solutions

- Cisco ISE
- Aruba ClearPass
- Microsoft NPS

2.11 NAC Use Cases

- BYOD environments
- Campus networks
- Zero Trust architecture

2.12 Exam & Interview Focus

- NAC uses **identity-based access**
- 802.1X is central protocol
- Quarantine VLAN is common enforcement

---

3. VIRTUAL CUSTOMER EDGE (vCE)

3.1 Definition

A **Virtual Customer Edge (vCE)** is a **software-based customer edge router/firewall** deployed as a **virtual machine**, replacing traditional hardware CE devices.

3.2 Why Virtual Customer Edge?

- Cloud connectivity
- Reduced hardware cost
- Rapid deployment
- SD-WAN integration

3.3 Traditional CE vs vCE

Feature	Traditional CE	vCE
Hardware	Physical	Virtual
Scalability	Limited	High
Deployment	Slow	Fast
Cost	High	Lower

3.4 vCE Architecture

```
Customer Network
      |
   vCE (VM)
      |
   Provider Network (MPLS/Internet)
```

3.5 vCE Functionalities

- Routing (BGP, OSPF)
- VPN termination
- Firewall
- NAT
- QoS

3.6 vCE in Cloud Environments

- AWS Transit Gateway + virtual router
- Azure vWAN
- Google Cloud Router

3.7 vCE with SD-WAN

- Centralized control
- Policy-based routing
- Dynamic path selection

3.8 Advantages & Challenges

Advantages

- Elastic scaling
- Automation
- Cloud-native

Challenges

- Performance tuning
- Licensing
- Security hardening

3.9 Exam Keywords

- CE vs vCE
- Virtual router
- Cloud edge
- SD-WAN

---

4. DATACENTER OPTIMIZATION

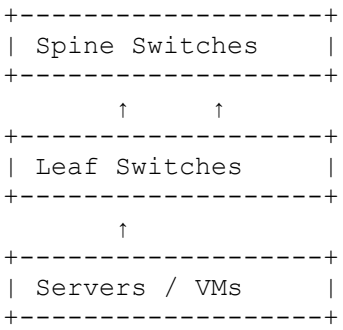
4.1 Definition

Datacenter Optimization involves **improving performance, scalability, reliability, and efficiency** of data center networking and infrastructure.

4.2 Traditional Datacenter Challenges

- East-West traffic congestion
- VLAN scaling limits
- Manual provisioning
- Poor visibility

4.3 Modern Datacenter Architecture



Spine-Leaf Architecture

4.4 Key Optimization Techniques

4.4.1 Spine-Leaf Fabric

- Predictable latency
- High bandwidth
- No STP dependency

4.4.2 East-West Traffic Optimization

- VM-to-VM communication
- Micro-segmentation
- Overlay networks

4.4.3 SDN & Automation

- Centralized control
- API-based provisioning
- Faster service rollout

4.4.4 Network Virtualization

- VXLAN overlays
- Logical isolation
- Multi-tenant support

4.4.5 Load Balancing

- L4/L7 load balancers
- Traffic distribution
- Application resilience

4.5 Datacenter Security Optimization

- Micro-segmentation
- Zero Trust networking
- Distributed firewalls

4.6 Performance Optimization Metrics

Metric	Purpose
Latency	Response time
Throughput	Bandwidth
Packet loss	Reliability
Utilization	Efficiency

4.7 Real-World Datacenter Optimization Example

- Cloud provider using:
  - Spine-leaf
  - VXLAN
  - SDN controller
  - Automated scaling

Result:

- Reduced latency
- Faster provisioning
- Improved resilience

4.8 Challenges in Datacenter Optimization

- Migration complexity
- Skill requirements
- Tool integration
- Cost management

4.9 Exam-Oriented Summary (Datacenter)

- Spine-leaf replaces core-distribution
- VXLAN solves VLAN scaling
- SDN improves automation

---

5. INTEGRATED ENTERPRISE CASE STUDY

Large Enterprise Cloud Network

- Virtual networking with VXLAN
- NAC for BYOD security
- vCE for cloud connectivity
- Optimized datacenter fabric
- Result: Secure, scalable, automated network

---

6. COMPARISON TABLES (EXAM ESSENTIAL)

6.1 VLAN vs VXLAN

Feature	VLAN	VXLAN
ID Space	4094	16M
Scope	L2	L2 over L3
Scalability	Limited	Very High

6.2 Traditional CE vs vCE

Aspect	CE	vCE
Deployment	Physical	Virtual
Cloud-ready	No	Yes
Automation	Low	High

---

7. TROUBLESHOOTING CONCEPTS

Issue	Cause
VM no connectivity	vSwitch config
NAC block	Policy mismatch
vCE down	VM resource issue
High latency	Fabric congestion

---

8. EXAM-ORIENTED QUICK REVISION

- Virtual networking uses overlays
- NAC controls access based on identity
- vCE replaces physical CE
- Datacenter optimization uses spine-leaf & SDN

---

9. INTERVIEW Q&A

Q: Why VXLAN is preferred in cloud?
A: It provides massive scalability and isolation.

Q: What is NAC's main goal?
A: Secure, policy-based network access.

Q: Why enterprises move to vCE?
A: Cost efficiency and cloud integration.

🎯 FINAL EXAM KEYWORDS

- Virtual networking
- Overlay/Underlay
- NAC
- 802.1X
- Virtual Customer Edge
- Spine-leaf
- Datacenter optimization

🟫 SESSION 18 – OPENFLOW & SDN CONTROLLERS

---

1. OPENFLOW

1.1 Definition of OpenFlow

OpenFlow is a **southbound communication protocol** used in **Software Defined Networking (SDN)** that enables an **SDN controller to directly program the forwarding behavior of network devices** (switches/routers).

➡️ It provides **standardized control over the data plane**.

1.2 Why OpenFlow is Important

- Enables **separation of control and data planes**
- Allows **centralized control**
- Eliminates vendor lock-in
- Enables **programmable networks**
- Foundation protocol for SDN

1.3 Role of OpenFlow in SDN

```
SDN Application
      ↑
Controller (OpenDaylight)
      ↑  (OpenFlow)
SDN Switch (Data Plane)
```

1.4 Key Characteristics of OpenFlow

- Controller-driven forwarding
- Flow-based packet handling
- Standardized interface
- Hardware and software switch support

1.5 OpenFlow Network Components

Component	Role
SDN Controller	Decision making
OpenFlow Switch	Packet forwarding
Secure Channel	Controller–switch communication
Flow Table	Packet handling rules

1.6 OpenFlow Switch Architecture

```
+----------------------+
| Flow Table 0         |
+----------------------+
| Flow Table 1         |
+----------------------+
| Flow Table N         |
+----------------------+
| Group Table          |
+----------------------+
| Meter Table          |
+----------------------+
```

1.7 OpenFlow Flow Table Structure (CORE EXAM TOPIC)

Each **flow entry** contains:

Field	Description
Match Fields	Packet header fields
Priority	Rule precedence
Counters	Statistics
Instructions	Actions to perform
Timeouts	Rule lifetime

1.8 Match Fields Examples

- Source IP
- Destination IP
- MAC address
- VLAN ID
- TCP/UDP ports
- Protocol type

1.9 Actions in OpenFlow

- Forward to port
- Drop packet
- Modify header
- Send to controller
- Flood packet

1.10 OpenFlow Packet Processing Workflow

```
Packet arrives at switch
        ↓
Flow table lookup
        ↓
Match found?
   ├── Yes → Apply action
   └── No  → Send packet to controller
```

1.11 OpenFlow Message Types

Message Type	Purpose
Hello	Version negotiation
Features	Capability exchange
Packet-In	Packet to controller
Packet-Out	Packet from controller
Flow-Mod	Add/modify/delete flow
Stats	Statistics

1.12 Advantages of OpenFlow

- Centralized traffic control
- Fine-grained policies
- Dynamic network behavior
- Research & innovation friendly

1.13 Limitations of OpenFlow

- Controller dependency
- Scalability concerns
- Flow table size limits
- Debugging complexity

1.14 Real-World OpenFlow Use Cases

- Traffic engineering
- DDoS mitigation
- Load balancing
- Research networks

---

2. HISTORY AND EVOLUTION OF OPENFLOW

2.1 Origin of OpenFlow

- Developed at **Stanford University (2008)**
- Goal: Enable network experimentation
- Introduced by **Nick McKeown**

2.2 Evolution Timeline

Year	Milestone
2008	OpenFlow v1.0
2011	ONF formed
2012	OpenFlow v1.3
2015+	Decline in favor of model-driven APIs

2.3 OpenFlow Versions Overview

Version	Key Features
1.0	Single flow table
1.1	Multiple tables
1.3	Meters, IPv6
1.5	Enhanced matching

2.4 Role of ONF (Open Networking Foundation)

- Standardization body
- Promotes open SDN
- Developed OpenFlow standards

2.5 Decline & Evolution Beyond OpenFlow

- Rise of **NETCONF/YANG**
- P4 programmable pipelines
- gRPC-based controllers
- Hybrid SDN models

⚠ **Still exam-critical as SDN foundation**

---

3. CONTROL PLANE vs DATA PLANE

3.1 Control Plane – Definition

The **control plane** is responsible for:

- Making forwarding decisions
- Routing calculations
- Policy enforcement

3.2 Data Plane – Definition

The **data plane** is responsible for:

- Actual packet forwarding
- Applying rules given by control plane
- High-speed packet processing

3.3 Traditional Networking (Coupled Planes)

```
Router/Switch
+------------------+
| Control Plane    |
| Routing Logic    |
+------------------+
| Data Plane       |
| Packet Forwarding|
+------------------+
```

3.4 SDN Networking (Decoupled Planes)

```
Controller (Control Plane)
          ↓ OpenFlow
Switch (Data Plane)
```

3.5 Control Plane vs Data Plane – Comparison

Aspect	Control Plane	Data Plane
Function	Decision making	Packet forwarding
Location	Centralized	Distributed
Speed	Slower	Very fast
Programmability	High	Limited

3.6 Benefits of Separation

- Centralized intelligence
- Simplified devices
- Network automation
- Faster innovation

3.7 Exam Focus Points

- OpenFlow moves control plane to controller
- Data plane becomes rule executor

---

4. OPENDAYLIGHT (ODL) ARCHITECTURE

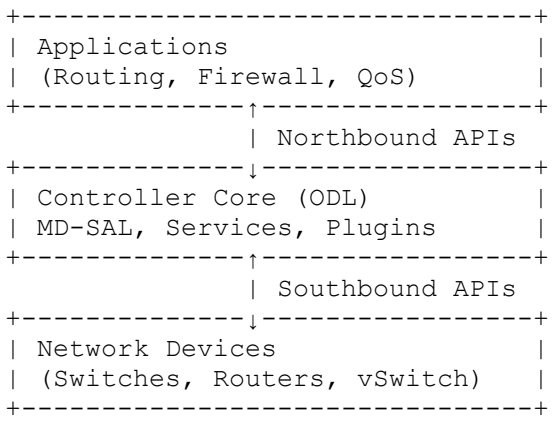
4.1 OpenDaylight – Definition

OpenDaylight (ODL) is an **open-source SDN controller platform** designed to **manage and program networks using open APIs and protocols**.

4.2 Purpose of OpenDaylight

- Act as SDN controller
- Provide abstraction layer
- Support multi-vendor networks
- Enable automation & orchestration

4.3 OpenDaylight Architecture Overview



4.4 Key Components of OpenDaylight

4.4.1 MD-SAL (Model Driven Service Abstraction Layer)

Heart of OpenDaylight

Functions:

- Data storage
- Service abstraction
- Model-driven architecture
- YANG-based data models

4.4.2 YANG Models

- Data modeling language
- Defines network configuration/state
- Used with NETCONF/RESTCONF

4.4.3 Southbound Plugins

Protocol	Purpose
OpenFlow	Flow programming

Protocol	Purpose
NETCONF	Device configuration
OVSDB	Virtual switch control
BGP-LS	Topology learning

4.4.4 Northbound APIs

- REST APIs
- Used by applications
- Abstract hardware complexity

4.5 OpenDaylight Controller Services

- Topology manager
- Flow manager
- Statistics manager
- Device inventory
- Authentication & authorization

4.6 OpenDaylight Workflow (STEP-BY-STEP)

```
Application sends intent
        ↓
Controller translates intent
        ↓
Flow rules generated
        ↓
Rules pushed via OpenFlow
        ↓
Switch forwards packets
```

4.7 OpenDaylight Deployment Models

- Standalone controller
- Clustered controller
- Integrated with NFV/Cloud

4.8 OpenDaylight Advantages

- Vendor-neutral
- Modular architecture
- Strong community support
- Scales via clustering

4.9 OpenDaylight Challenges

- Complexity
- Resource intensive
- Steep learning curve
- Operational maturity

4.10 Real-World Use Cases of OpenDaylight

- SDN labs & research
- ISP traffic engineering
- Data center automation
- NFV orchestration

---

5. OPENFLOW vs OPENDAYLIGHT

Aspect	OpenFlow	OpenDaylight
Type	Protocol	Controller
Role	Southbound API	Control plane
Function	Flow control	Network management
Dependency	Used by controller	Uses OpenFlow

---

6. SECURITY CONSIDERATIONS

6.1 OpenFlow Security Issues

- Controller compromise
- Man-in-the-middle attacks
- Unauthorized flow injection

6.2 Security Best Practices

- TLS between controller & switch
- Role-based access
- Controller redundancy
- Monitoring & logging

---

7. TROUBLESHOOTING CONCEPTS

Issue	Cause
No flows installed	Controller unreachable
Packet-In flood	Missing rules
High latency	Controller overload
Switch not detected	Protocol mismatch

---

8. EXAM-ORIENTED QUICK REVISION

- OpenFlow = SDN southbound protocol
- Flow tables define forwarding
- Control plane centralized
- Data plane simplified
- OpenDaylight = SDN controller
- MD-SAL is core of ODL

---

9. INTERVIEW Q&A

Q: What problem does OpenFlow solve?
A: It enables centralized, programmable control of network forwarding.

Q: Difference between OpenFlow and NETCONF?
A: OpenFlow programs forwarding; NETCONF configures devices.

Q: Why MD-SAL is important?
A: It abstracts services and enables model-driven control.

🎯 FINAL EXAM KEYWORDS

- OpenFlow
- Flow table
- Packet-In / Flow-Mod
- Control plane
- Data plane
- OpenDaylight
- MD-SAL
- YANG

📓 SESSION 19 – OPENDAYLIGHT PLATFORM & SDN LAB STACK

Topics Covered

1. OpenDaylight Basics
2. Clustering
3. MD-SAL
4. Internal Datastore
5. OpenFlow Plugin
6. Open vSwitch (OVS)
7. Mininet
8. L2Switch Application

---

1. OPENDAYLIGHT BASICS

1.1 What is OpenDaylight (ODL)?

OpenDaylight is an **open-source, modular Software Defined Networking (SDN) controller** that provides:

- Centralized network intelligence
- Abstraction of network devices

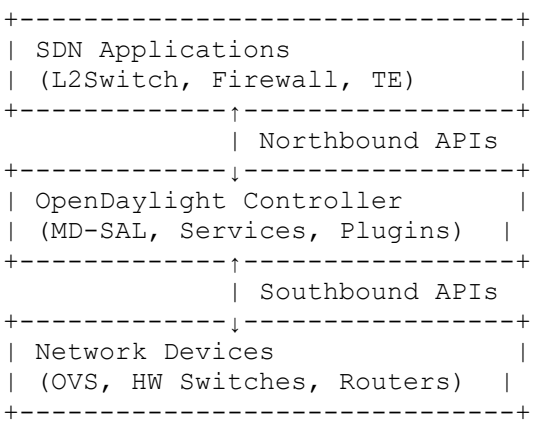
- Programmable control via APIs
- Multi-protocol southbound support

It acts as the **control plane** in SDN architectures.

1.2 Why OpenDaylight?

- Vendor-neutral SDN controller
- Supports **OpenFlow, NETCONF, OVSDB, BGP**
- Model-driven (YANG)
- Extensible via plugins and bundles
- Used in **enterprise, ISP, data center, and labs**

1.3 OpenDaylight Position in SDN Stack



1.4 OpenDaylight Features

- **Model-Driven Architecture**
- **REST/RESTCONF APIs**
- **Plugin-based**
- **Cluster-ready**
- **Multi-protocol southbound**

1.5 OpenDaylight Use Cases

- SDN labs & teaching
- Data center automation
- NFV orchestration
- Traffic engineering
- Network research

1.6 Exam Points

- OpenDaylight = SDN **controller**
- Control plane centralized
- Uses **MD-SAL** internally

---

2. OPENDAYLIGHT CLUSTERING

2.1 What is Clustering in OpenDaylight?

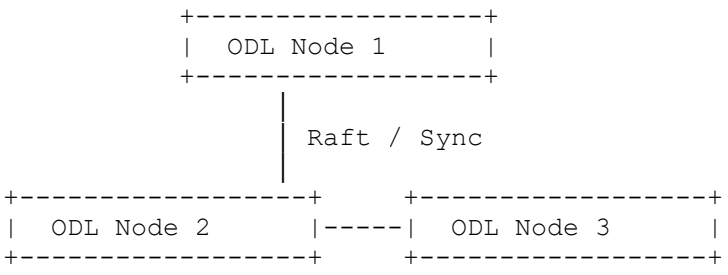
Clustering allows **multiple OpenDaylight controller instances** to work together as **one logical controller** to achieve:

- High availability
- Scalability
- Fault tolerance

2.2 Why Clustering is Required

- Avoid **single point of failure**
- Handle large-scale networks
- Distribute load
- Improve reliability

2.3 OpenDaylight Cluster Architecture



2.4 Key Concepts in Clustering

Concept	Description
Leader	Active controller
Follower	Standby / replica
Shard	Data partition
Consensus	Agreement on state

2.5 Consensus Algorithm Used

- **Raft** (most common in ODL)
- Ensures **data consistency**
- Handles leader election & failover

2.6 Clustered Datastores

- Config datastore → replicated
- Operational datastore → synchronized
- Data consistency maintained across nodes

2.7 Advantages of Clustering

- High availability
- Fast failover
- Horizontal scaling

2.8 Limitations

- Complex setup
- Increased resource usage
- Debugging complexity

2.9 Exam Focus

- Clustering improves **reliability & scalability**
- Uses **Raft-based datastore synchronization**

---

3. MD-SAL (MODEL-DRIVEN SERVICE ABSTRACTION LAYER)

3.1 What is MD-SAL?

MD-SAL is the **core framework of OpenDaylight** that provides:

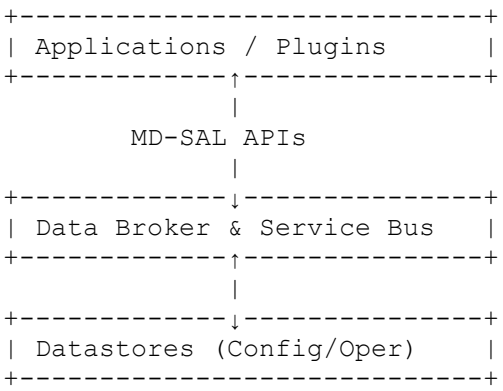
- Data abstraction
- Service registration
- Inter-module communication
- Model-driven data handling

➡ **Heart of OpenDaylight**

3.2 Why MD-SAL is Needed

- Decouple applications from protocols
- Enable YANG-based models
- Provide consistent data access
- Support multi-protocol control

3.3 MD-SAL Architecture



3.4 MD-SAL Components

Component	Function
Data Broker	Read/write data
RPC Broker	Remote procedure calls
Notification Service	Event handling
Binding Aware	Java/YANG binding
Binding Independent	Generic access

3.5 MD-SAL Data Models

- Defined using **YANG**
- Represent topology, flows, stats
- Shared across all modules

3.6 Advantages of MD-SAL

- Loose coupling
- Extensibility
- Consistency
- Vendor neutrality

3.7 Exam Points

- MD-SAL = core of ODL
- Uses **YANG models**
- Abstracts device complexity

---

4. INTERNAL DATASTORE (ODL)

4.1 What is Internal Datastore?

The **internal datastore** stores **network state, configuration, and operational data** used by OpenDaylight.

4.2 Types of Datastores

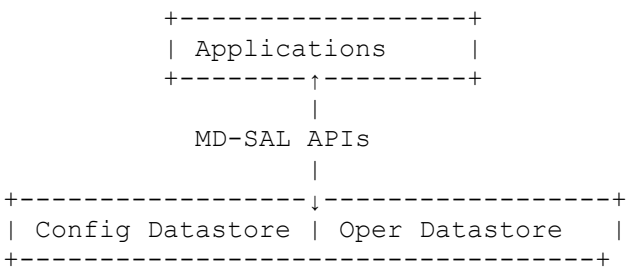
4.2.1 Configuration Datastore

- Intended (desired) configuration
- Set by applications/admins
- Persistent

4.2.2 Operational Datastore

- Real-time network state
- Device-reported data
- Dynamic

4.3 Datastore Architecture



4.4 Datastore Consistency

- Clustered via Raft
- Sharded for scalability
- Transaction-based updates

4.5 Datastore Operations

- Read
- Write
- Commit
- Rollback

4.6 Exam Focus

- Two datastores: **Config & Operational**
- Used by **MD-SAL**

---

5. OPENFLOW PLUGIN

5.1 What is OpenFlow Plugin?

The **OpenFlow Plugin** is a **southbound module** in OpenDaylight that enables:

- Communication with OpenFlow switches
- Installation of flow rules
- Collection of statistics

5.2 Role of OpenFlow Plugin

```
Application
    ↓
```

```
MD-SAL
    ↓
OpenFlow Plugin
    ↓
OpenFlow Switch
```

5.3 Functions of OpenFlow Plugin

- Device discovery
- Flow programming
- Packet-In/Out handling
- Statistics collection

5.4 OpenFlow Message Handling

Message	Direction
Packet-In	Switch → Controller
Flow-Mod	Controller → Switch
Stats	Both

5.5 Flow Installation Logic

1. Packet arrives at switch
2. No match → Packet-In
3. Controller computes action
4. Flow-Mod sent
5. Subsequent packets forwarded locally

5.6 Exam Focus

- OpenFlow Plugin = southbound interface
- Programs **data plane**

---

6. OPEN vSWITCH (OVS)

6.1 What is Open vSwitch?

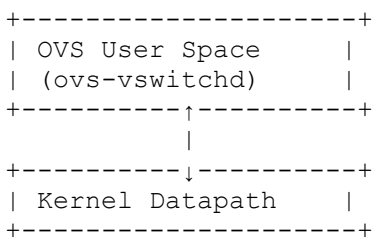
Open vSwitch (OVS) is a **software-based virtual switch** designed for:

- Virtualized environments
- SDN integration
- High-performance switching

6.2 Why OVS?

- Works with hypervisors
- Supports OpenFlow
- Supports OVSDB
- Production-grade

6.3 OVS Architecture



6.4 OVS Capabilities

- VLAN tagging
- VXLAN tunnels
- OpenFlow rules
- QoS & mirroring

6.5 OVS + OpenDaylight

```
OVS  ← OpenFlow / OVSDB →  OpenDaylight
```

6.6 Real-World Use

- Cloud platforms
- NFV
- SDN labs

6.7 Exam Points

- OVS = software switch
- Common with **Mininet**

---

7. MININET

7.1 What is Mininet?

Mininet is a **network emulation tool** that creates:

- Virtual hosts
- Virtual switches
- Virtual links
  on a **single Linux machine**

7.2 Purpose of Mininet

- SDN experimentation
- OpenFlow testing
- Controller validation
- Teaching & labs

7.3 Mininet Architecture

```
+---------------------------------+
| Linux Kernel                    |
|   ├─ Network Namespaces         |
```

```
|   ├─ Virtual Ethernet (veth)      |
|   ├─ Open vSwitch                 |
+-----------------------------------+
```

7.4 Mininet Components

Component	Role
Host	Linux namespace
Switch	OVS
Controller	ODL / Remote
Link	veth pair

7.5 Typical Mininet + ODL Setup

```
Mininet → OVS → OpenDaylight Controller
```

7.6 Advantages & Limitations

Advantages

- Lightweight
- Fast setup
- Real OpenFlow behavior

Limitations

- Not production
- Single machine scale

7.7 Exam Focus

- Mininet used for **SDN labs**
- Uses **OVS**

---

8. L2SWITCH APPLICATION

8.1 What is L2Switch Application?

The **L2Switch application** is a **basic SDN application** in OpenDaylight that:

- Emulates Layer-2 switching behavior
- Learns MAC addresses
- Installs flow rules dynamically

8.2 Purpose of L2Switch

- Demonstrate SDN concepts
- Replace traditional switch logic
- Learning example for SDN apps

8.3 L2Switch Working Principle

```
Packet-In →
Controller learns MAC →
Flow installed →
Future packets switched locally
```

8.4 MAC Learning Logic

1. Source MAC learned
2. Destination MAC lookup
3. Known → install flow
4. Unknown → flood

8.5 L2Switch vs Traditional Switch

Feature	Traditional	L2Switch App
Control	Local	Controller
Learning	Hardware	Software
Flexibility	Low	High

8.6 Limitations of L2Switch

- Controller dependency
- Not scalable
- Demo/learning focused

8.7 Exam Focus

- L2Switch = **reference SDN app**
- Demonstrates control/data separation

---

9. INTEGRATED LAB WORKFLOW (EXAM + LAB)

```
Mininet
  ↓
Open vSwitch
  ↓ (OpenFlow)
OpenDaylight
  ↓
L2Switch Application
```

---

10. REAL-WORLD CASE STUDY

SDN Lab Environment

- Mininet topology
- OVS switches
- OpenDaylight controller
- L2Switch application
- Demonstrated MAC learning & flow programming

---

11. COMPARISON TABLES (EXAM ESSENTIAL)

11.1 OVS vs Hardware Switch

Feature	OVS	Hardware
Type	Software	Physical
Flexibility	High	Medium
SDN Support	Native	Vendor-dependent

11.2 Mininet vs Real Network

Aspect	Mininet	Real
Cost	Free	High
Scale	Small	Large
Use	Lab	Production

---

12. TROUBLESHOOTING CONCEPTS

Issue	Cause
No flows	L2Switch not running
Switch not visible	OpenFlow plugin down
Mininet no connect	Controller IP mismatch
Cluster inconsistency	Datastore sync issue

---

13. EXAM-ORIENTED QUICK REVISION

- OpenDaylight = SDN controller
- MD-SAL = core abstraction layer
- Datastores = Config & Operational
- OpenFlow Plugin programs switches
- OVS = software switch
- Mininet = SDN emulator
- L2Switch = basic SDN app

---

14. INTERVIEW Q&A

Q: Why MD-SAL is important in ODL?
A: It abstracts data/services and enables model-driven control.

Q: Why Mininet is popular in SDN labs?
A: Lightweight, real OpenFlow behavior.

Q: What does L2Switch application demonstrate?
A: MAC learning and flow installation via SDN.

🎯 FINAL EXAM KEYWORDS

- OpenDaylight
- Clustering
- MD-SAL
- Config datastore
- OpenFlow plugin
- Open vSwitch
- Mininet
- L2Switch

🟫 SESSION 20 – ADVANCED SDN & NFV CONCEPTS

Topics Covered

1. OpenDaylight AAA
2. OVSDB Virtualization
3. Application Intents
4. Group-Based Policy (GBP)
5. Service Function Chaining (SFC)
6. LISP Flow Mapping
7. Virtual Tenant Networks (VTN)

---

1. OPENDAYLIGHT AAA (Authentication, Authorization, Accounting)

1.1 Definition

OpenDaylight AAA is a **security framework** that provides:

- **Authentication** – Who are you?

- **Authorization** – What can you do?
- **Accounting** – What did you do?

It secures **controller access, APIs, and applications**.

1.2 Why AAA is Required in SDN

- Centralized controller = high-value target
- Multi-user & multi-tenant environments
- Secure API access
- Compliance & auditing

1.3 OpenDaylight AAA Architecture

```
User / Application
        ↓
   Authentication Layer
        ↓
   Authorization Engine
        ↓
   Accounting / Logs
        ↓
   OpenDaylight Services
```

1.4 AAA Components in OpenDaylight

Component	Function
Authentication	Identity verification
Authorization	Role-based access
Accounting	Logging & auditing
Token Service	Session management

1.5 Authentication Methods

- Local user database
- LDAP integration
- REST API tokens

1.6 Authorization (RBAC)

- Role-Based Access Control
- Roles: Admin, Operator, User
- Fine-grained permissions on APIs

1.7 Accounting

- Tracks login/logout
- API usage
- Configuration changes

1.8 Exam Focus

- AAA secures **controller & APIs**
- Uses **RBAC**
- Accounting = auditing

---

2. OVSDB VIRTUALIZATION

2.1 What is OVSDB?

OVSDB (Open vSwitch Database) is a **management protocol and database schema** used to **configure and manage Open vSwitch instances**.

2.2 Why OVSDB is Needed

- Configure bridges & ports
- Manage tunnels (VXLAN, GRE)
- Control virtual switches centrally
- Complement OpenFlow

2.3 OVSDB vs OpenFlow

Aspect	OVSDB	OpenFlow
Purpose	Configuration	Forwarding
Scope	Management	Packet handling
Layer	Control	Data plane

2.4 OVSDB Virtualization Architecture

```
OpenDaylight Controller
   ├── OpenFlow Plugin
   └── OVSDB Plugin
          ↓
       Open vSwitch
```

2.5 Virtualization Using OVSDB

- Create virtual bridges
- Add/remove ports
- Configure VXLAN tunnels
- Manage QoS policies

2.6 Exam Focus

- OVSDB = **configuration**
- OpenFlow = **forwarding**

---

3. APPLICATION INTENTS

3.1 What are Application Intents?

Application Intents express **WHAT the application wants**, not **HOW the network should do it**.

➡ **Intent-Based Networking (IBN)** concept.

3.2 Why Intents?

- Simplifies network management
- Hides complexity
- Aligns network with business goals

3.3 Intent-Based Networking Workflow

```
Application Intent
        ↓
Intent Engine
        ↓
Policy Translation
        ↓
Flow Rules
```

3.4 Examples of Intents

- “Connect App A to DB B securely”
- “Limit bandwidth to 100 Mbps”
- “Isolate tenant X traffic”

3.5 Benefits of Application Intents

- Automation
- Reduced human error
- Faster provisioning

3.6 Exam Focus

- Intents = declarative model
- Focus on **outcomes**, not commands

---

4. GROUP-BASED POLICY (GBP)

4.1 Definition

Group-Based Policy (GBP) is a **policy-driven SDN model** where **security and connectivity rules are applied to groups of endpoints** instead of IP addresses.

4.2 Why GBP?

- Traditional ACLs are complex
- Dynamic environments (VMs, containers)
- Application-centric policies

4.3 GBP Core Components

Component	Description
Endpoint	VM, container, server
Endpoint Group (EPG)	Logical grouping
Contract	Policy between groups
Classifier	Traffic match
Action	Permit, deny, redirect

4.4 GBP Architecture

```
Endpoint → EPG → Contract → EPG → Endpoint
```

4.5 GBP Workflow

1. Endpoints assigned to groups
2. Policies defined between groups
3. Controller enforces rules
4. Flows programmed dynamically

4.6 GBP vs Traditional ACL

Feature	ACL	GBP
Basis	IP/Port	Group
Scalability	Low	High
Automation	Low	High

4.7 Exam Focus

- GBP is **policy-centric**
- Uses **EPGs and Contracts**

---

5. SERVICE FUNCTION CHAINING (SFC)

5.1 Definition

Service Function Chaining (SFC) is a technique to **steer traffic through an ordered sequence of network services** (firewall, IDS, load balancer).

5.2 Why SFC?

- Traditional service insertion is rigid
- NFV introduces virtual services
- Dynamic traffic steering needed

5.3 Service Functions Examples

- Firewall
- IDS/IPS
- DPI
- Load balancer
- WAN optimizer

5.4 SFC Architecture

```
Client → FW → IDS → LB → Server
```

5.5 SFC Components

Component	Role
Classifier	Identifies traffic
Service Function Forwarder (SFF)	Forwards traffic
Service Function (SF)	Actual service
Controller	Orchestrates chain

5.6 SFC Control Methods

- NSH (Network Service Header)
- MPLS labels
- OpenFlow rules

5.7 Exam Focus

- SFC = ordered service path
- Works with **NFV**

---

6. LISP FLOW MAPPING

6.1 What is LISP?

LISP (Locator/ID Separation Protocol) separates:

- **Endpoint Identifier (EID)** – Who
- **Routing Locator (RLOC)** – Where

6.2 Why LISP?

- Mobility
- Scalability
- Simplified routing

6.3 LISP Flow Mapping Concept

```
EID → Map-Request → Map-Server → RLOC
```

6.4 LISP Components

Component	Function
ETR	Encapsulates
ITR	Decapsulates
Map Server	Mapping database
Map Resolver	Lookup

6.5 Flow Mapping Use Cases

- SD-WAN
- Mobile networks
- Multi-site enterprises

6.6 Exam Focus

- LISP separates **identity & location**
- Used for scalable flow mapping

---

7. VIRTUAL TENANT NETWORKS (VTN)

7.1 Definition

Virtual Tenant Network (VTN) is a **logical, isolated network** created over a **shared physical infrastructure** for each tenant.

7.2 Why VTN?

- Multi-tenancy
- Cloud isolation
- Secure resource sharing

7.3 VTN Architecture

```
Tenant A → VTN A → Physical Network
Tenant B → VTN B → Physical Network
```

7.4 VTN Components

Component	Role
Tenant	User/customer
Virtual Network	Logical topology
Virtual Router	L3 routing
Policies	Security & QoS

7.5 VTN Implementation Technologies

- VXLAN
- GRE
- OpenFlow
- OpenDaylight VTN Manager

7.6 VTN Use Cases

- Cloud providers

- Enterprises
- NFV platforms

7.7 Exam Focus

- VTN = tenant isolation
- Uses overlays

---

8. INTEGRATED SDN/NFV CASE STUDY

Cloud Service Provider

- OpenDaylight controller
- AAA for secure access
- OVSDB for switch management
- Application intents for automation
- GBP for security
- SFC for service chaining
- VTN for tenant isolation

---

9. COMPARISON TABLES (EXAM ESSENTIAL)

9.1 OpenFlow vs OVSDB

Aspect	OpenFlow	OVSDB
Function	Forwarding	Configuration
Scope	Data plane	Control plane

9.2 ACL vs GBP

Feature	ACL	GBP
Policy Model	Address-based	Group-based
Automation	Low	High

---

10. TROUBLESHOOTING CONCEPTS

Issue	Cause
AAA login failure	Auth backend issue
No tunnel	OVSDB misconfig
Intent not applied	Translation failure
Service chain broken	SFF misconfig
Tenant leak	Policy error

---

11. EXAM-ORIENTED QUICK REVISION

- AAA secures controller
- OVSDB manages OVS
- Intents simplify networking
- GBP uses groups & contracts
- SFC chains services
- LISP separates ID & location
- VTN enables multi-tenancy

---

12. INTERVIEW Q&A

Q: Why GBP is better than ACLs in cloud?
A: It scales dynamically and is application-centric.

Q: Why SFC is important in NFV?
A: It enables dynamic service insertion.

Q: Why LISP is used in SD-WAN?
A: For scalable endpoint mobility and mapping.

🎯 FINAL EXAM KEYWORDS

- AAA
- OVSDB
- Intent-based networking
- Group-Based Policy

- Service Function Chaining
- LISP
- Virtual Tenant Network