Jeremy's IT Lab CCNA 200-301: Study Notes

Free study notes for the Jeremy's IT Lab free CCNA 200-301 course, one section per day. Written by HoverNotes in our own words as a companion to the videos: watch the lesson, do the lab, then use these to review. Not affiliated with Jeremy's IT Lab or Cisco.

Course guide, study plan and reviews: https://hovernotes.io/courses/jeremys-it-lab-ccna

Made with HoverNotes, the notes panel that sits beside any video. Version 2026-11-02.

Day 1: Network Devices

Exam domain: 1. Network Fundamentals · **Video:** Network Devices (30 min) · Lab: Packet Tracer Introduction

Key ideas

- A **network** is two or more nodes connected so they can share resources. Two PCs and one cable is already a network.
- **Client vs server is a role, not a kind of machine.** A client uses a service; a server provides one. Your phone is a client while it streams a video and a server when it AirDrops a file to a friend.
- **Switch:** many ports (usually 24 or more), connects end hosts **within one LAN**. It doesn't connect LANs to each other or to the Internet. Cisco's enterprise switches are the **Catalyst** line.
- **Router:** few ports, connects **different networks** (LAN to LAN, or to the Internet). Traffic between two branch offices goes PC → switch → router → Internet → router → switch → server, and back the same way.
- **Firewall:** filters traffic entering and leaving a network using rules you configure (allow or deny). It can sit outside the router, inside it, or both. A **next-generation firewall (NGFW)** adds advanced filtering such as IPS. Cisco examples: ASA, Firepower.
- **Network firewall vs host-based firewall:** a hardware appliance that filters between networks, versus software on one PC. Use both: the host firewall is an extra layer.
- In diagrams, a **cloud** stands for the Internet, or any part of the network whose details don't matter.

Remember for the exam

- 30 PCs in one department need a **switch**, not a router: routers don't have that many ports and don't connect hosts in one LAN.
- Connecting **separate networks** together is the **router's** job. A firewall can connect networks too, but its purpose is to filter.
- Clients and servers are both **end hosts** (endpoints). "End host" describes the device, not its role.
- "Next-level" and "top-layer" firewalls don't exist; **next-generation** does.

Check yourself

1. Your laptop downloads a file from a coworker's shared folder. Which one is the server? *(The coworker's PC: it provides the service.)*
2. Why can't two branch LANs talk to each other through their switches alone? *(Switches only forward within a LAN; you need routers between networks.)*
3. Where can a network firewall be placed relative to the router? *(Outside it, inside it, or both.)*

Do next

Install Cisco Packet Tracer (free with a Cisco account) and follow the Day 1 lab. Download Jeremy's Anki deck from the video description and keep one CCNA deck for the whole course.

Day 2: Interfaces and Cables

Exam domain: 1. Network Fundamentals · **Video:** Interfaces and Cables (36 min) · Lab: Connecting Devices

Key ideas

- **Interface = port.** The wired port on most switches, routers and PCs is an **RJ-45** port, which takes the RJ-45 connector at the end of a copper Ethernet cable.
- **Ethernet is a family of standards**, defined by the IEEE under **802.3**, not one protocol. Standards exist so that gear from different vendors fits and talks together.
- **Speed is measured in bits per second**, not bytes. 1 byte = 8 bits, so a gigabyte is 8 times a gigabit. Units step up by 1,000: kilobit, megabit, gigabit, terabit.
- **UTP (unshielded twisted pair)** has 4 pairs (8 wires). The twisting reduces electromagnetic interference (EMI). Maximum length for twisted-pair Ethernet is **100 m**.

Common name	Informal name	IEEE	Speed	Pairs used
Ethernet	10BASE-T	802.3i	10 Mbps	2
Fast Ethernet	100BASE-T	802.3u	100 Mbps	2
Gigabit Ethernet	1000BASE-T	802.3ab	1 Gbps	4
10 Gigabit Ethernet	10GBASE-T	802.3an	10 Gbps	4

- **Pins for 10/100 Mbps:** two pairs, pins **1-2** and **3-6**. A PC or server NIC and a router (or firewall) **transmit on 1 and 2, receive on 3 and 6**. A **switch is the opposite**: it receives on 1 and 2 and transmits on 3 and 6. Separate transmit and receive wires allow **full duplex** (both sides send at once, no collisions).
- **1000BASE-T and 10GBASE-T** use all four pairs (adding pins **4-5** and **7-8**), and every pair is **bidirectional**.
- **Straight-through cable:** pin 1 to pin 1, 2 to 2, and so on. Use it between **unlike** devices (PC to switch, router to switch). **Crossover cable:** pairs swapped (1 to 3, 2 to 6), used between **like** devices (switch to switch, router to router, PC to router, PC to PC).
- **Auto MDI-X** lets a port detect which pins its neighbor transmits on and adjust, so on modern gear either cable type works.
- **Fiber** sends light through a glass core, with separate strands to transmit and receive (one connector each way). It plugs into an **SFP** transceiver in the device. Layers of the cable, centre outward: core, reflective cladding, buffer, outer jacket.

	Multimode (MMF)	Single-mode (SMF)
Core	Wider, several light angles (modes)	Narrower, one mode
Light source	Cheaper, often LED	Laser
Cost	Lower	Higher
Distance	Longer than UTP, shorter than SMF	Longest

Standard	IEEE	Speed	Cable	Max length
1000BASE-LX	802.3z	1 Gbps	MMF or SMF	550 m (MMF), 5 km (SMF)
10GBASE-SR	802.3ae	10 Gbps	MMF	400 m
10GBASE-LR	802.3ae	10 Gbps	SMF	10 km
10GBASE-ER	802.3ae	10 Gbps	SMF	30 km

- **UTP vs fiber:** UTP is cheaper, with cheaper ports, but is limited to 100 m, is exposed to EMI, and leaks a faint signal that could be tapped. Fiber costs more and needs pricier SFP ports, but goes much farther, ignores EMI and does not leak signal.

Remember for the exam

- NICs, routers and firewalls transmit on **1, 2**; switches transmit on **3, 6**.
- **Like devices need crossover** (without Auto MDI-X); unlike devices use straight-through. Two old routers on a straight-through cable will not communicate.
- Over 100 m but within a few hundred metres, cheaply: **multimode fiber**. Kilometres: **single-mode fiber**. Hosts to a nearby switch: **UTP**.
- Speed is in **bits**, and the 1000BASE-T / 10GBASE-T pairs are bidirectional.

Check yourself

1. A PC is cabled straight to a router's Ethernet port with no switch in between. Which cable fits, and why? *(Crossover. Both transmit on pins 1 and 2, so the pairs must be swapped, unless Auto MDI-X handles it.)*
2. Two buildings are 300 m apart and the budget is tight. What cable? *(Multimode fiber: UTP stops at 100 m, and single-mode costs more than needed.)*
3. Which pairs does 100BASE-T use, and how many does 1000BASE-T use? *(Pins 1-2 and 3-6, two pairs; 1000BASE-T uses all four pairs.)*

Do next

Do the Day 2 Packet Tracer lab (choosing the right cable between devices). Add the Day 2 Anki cards: the copper and fiber standard tables and the transmit pin pairs.

Day 3: How the TCP/IP Model Actually Works

Exam domain: 1. Network Fundamentals · **Video:** How the TCP/IP Model Actually Works (43 min) · Lab: OSI Model

Key ideas

- **Protocols** are rules for how devices exchange data; **standards** are vendor-neutral specifications of them. Open standards are why a Mac, a Linux server and an Android phone can talk. TCP/IP won because it was an open standard that could run over many network types. ARPANET switched to it on 1 January 1983.
- **Who writes standards:** the **IEEE** (Ethernet 802.3, Wi-Fi 802.11, mostly physical and local-network rules) and the **IETF** (TCP, IP, UDP, HTTP, DNS; published as **RFCs**, free online).
- **A layered model** groups protocols by job. Each layer uses the layer below and serves the layer above, and one layer can change without changing the others (swap Ethernet for Wi-Fi and the upper layers don't care).
- The lesson uses a **five-layer TCP/IP model**:

Layer	Name here	Job	Identifier	Examples
5 (OSI 7)	Application	Format, send and interpret data for an application	n/a	HTTP/HTTPS, FTP, TFTP
4	Transport	Process-to-process delivery between hosts	**Port numbers** (web 80, file server 21)	TCP, UDP
3	Internet	End-to-end delivery between hosts across networks	**IP addresses**	IPv4, IPv6, ICMP
2	Local network	Hop-to-hop delivery inside one LAN	**MAC addresses**	Ethernet, Wi-Fi
1	Physical	Send bits as electrical, optical or radio signals	n/a	UTP, fiber, Wi-Fi radio, NICs

- A **hop** is one step from a router or host to the next. Switches only extend the LAN and don't count as hops. PC to R1 to R2 to server is three hops.
- **Routers work mainly at layer 3.** Layer 4 is mostly a conversation between the two end hosts, and application data is interpreted only by them.
- **Encapsulation** (sending host, going down): the application data gets a layer 4 header, then a layer 3 header, then a layer 2 **header and trailer** (the trailer lets the receiver check for errors). **Decapsulation** is the reverse on the receiver. Bits go on the wire with the layer 2 header first and trailer last.
- **PDU names:**

Layer	PDU
4	**Segment** (TCP) or **datagram** (UDP)
3	**Packet**

Layer	PDU
2	**Frame**

Only frames travel on the wire. The **payload** of a PDU is everything inside it, excluding that layer's own header and trailer.

- **Adjacent-layer interaction:** each layer serves the layer above and relies on the layer below. **Same-layer interaction:** a layer talks to its counterpart on the other device (ports with ports, IP addresses with IP addresses, MAC with MAC).
- **OSI model:** seven layers (application, presentation, session, transport, network, data link, physical), designed by ISO. It lost to TCP/IP as a protocol suite but survives as a reference model. Most sources use five layers with OSI names: layer 3 is the **network** layer, layer 2 the **data link** layer, and the application layer is called **layer 7**.

Remember for the exam

- Know the layer jobs and numbers: L1 physical, L2 MAC and frames, L3 IP and packets, L4 ports and segments/datagrams, L7 application.
- **TCP makes segments, UDP makes datagrams.**
- Layer names vary between books; people say "layer 2" and "layer 3" in practice.
- A **port number** identifies a process on a host, not a physical port.

Check yourself

1. A frame is received. In what order are headers examined and removed? *(Layer 2 header and trailer first, then layer 3, then layer 4, then data goes to the application.)*
2. Why do routers and switches not need to understand HTTP? *(They forward using layer 3 and layer 2 information; only the end hosts interpret application data.)*
3. A server runs a web service and a file service on one IP address. What lets the PC reach the right one? *(The destination port number at layer 4.)*

Do next

Do the Day 3 OSI model lab, then add the Day 3 Anki cards (layers, PDU names, encapsulation order).

Day 4: Intro to the CLI

Exam domain: 1. Network Fundamentals · **Video:** Intro to the CLI (31 min) · Lab: Basic Device Security

Key ideas

- **Cisco IOS** is the operating system on Cisco routers and switches (unrelated to Apple's iOS). You configure it through a **CLI** (command-line interface); engineers generally prefer it to a GUI such as ASDM for firewalls.
- **First configuration is always through the console port.** The cable for the RJ-45 console port is a **rollover cable**: RJ-45 on one end, usually DB-9 on the other (so a USB-to-serial adapter on modern laptops). Pins run reversed: 1 to 8, 2 to 7, 3 to 6, 4 to 5. Many switches also have a USB mini-B console port.
- You connect with a **terminal emulator** such as PuTTY (choose Serial). Cisco's default console settings:

Setting	Default
Speed (baud)	9600 bps
Data bits	8
Stop bits	1
Parity	None
Flow control	None

- **CLI modes:**

Mode	Prompt	Notes
User EXEC	`Router>`	Limited, mostly look-only. Default host name on a router is `Router`.
Privileged EXEC	`Router#`	Full view of the config, reload, save the config. Enter with `enable`.
Global configuration	`Router(config)#`	Where you change the configuration. Enter with `configure terminal`.

- **Help and shortcuts:** `?` lists commands. `en?` (no space) lists words that complete what you typed; `enable ?` (with a space) lists what can come next, and `<cr>` means the command is complete. **Tab** completes a word. Any unambiguous abbreviation works (`en`, `conf t`); `e` alone is ambiguous in user mode (enable, exit).
- **Two config files:** `running-config` is the active one (in RAM; every command edits it). `startup-config` is loaded at boot. Until you save, startup-config doesn't exist and a restart loses your changes.
- **Passwords are case-sensitive** and are not echoed as you type. `enable password` stores in **plain text**. `service password-encryption` scrambles every password in the config with Cisco's weak **type 7** scheme (easily cracked online). `enable secret` is hashed (type **5**, MD5) and is stronger.
- **If both are set,** `enable secret` **wins** and the enable password is ignored. `service password-encryption` has no effect on `enable secret`, which is always hashed.

- **Turning** `service password-encryption` **off** doesn't decrypt passwords already encrypted, but new passwords are then stored in clear text.

Commands

```
enable                                  ! user EXEC -> privileged EXEC
configure terminal                      ! privileged EXEC -> global config (short:
conf t)
enable password CCNA                    ! plain-text password for privileged EXEC
enable secret Cisco                     ! hashed password; preferred, takes precedence
service password-encryption             ! type 7-encrypt all current and future
passwords
no service password-encryption          ! the "no" form cancels a command
do show running-config                  ! run a privileged EXEC command from config
mode
show running-config                     ! the active configuration (short: sh run)
show startup-config                     ! the saved configuration
write                                   ! save running-config to startup-config
write memory                            ! same as write
copy running-config startup-config     ! same again, longer form
exit                                    ! go back one mode
```

Remember for the exam

- RJ-45 console = **rollover** cable. It is neither straight-through nor crossover.
- Save with **write**, **write memory** or **copy running-config startup-config**, from privileged EXEC.
- Strongest option: `enable secret`. `service password-encryption` is weak and doesn't change what the password is, only how it's displayed.
- Know the full commands, not just the shortcuts.

Check yourself

1. You see `Switch(config)#`. Which mode is that, and which command got you there from `Switch#`? *(Global configuration;* `configure terminal`*.)*
2. Both `enable password` and `enable secret` are set. Which does the device ask for? *(Only the enable secret.)*
3. You reload a router after changing its config but never saved it. What config does it boot with? *(Whatever is in startup-config, or the default if none exists; the running-config changes are lost.)*

Do next

Do the Day 4 lab (basic device security in Packet Tracer): modes, `enable secret`, `service password-encryption`, saving the config. Add the Day 4 Anki cards.

Day 5: Ethernet LAN Switching (Part 1)

Exam domain: 2. Network Access · **Video:** Ethernet LAN Switching (Part 1) (38 min)

Key ideas

- **Ethernet spans layers 1 and 2.** Layer 1 is the cabling from Day 2; this lesson is layer 2. **Switches work at layer 2.**
- **A LAN is one local network.** Adding switches extends a LAN; it doesn't split it. Two switches linked together are one LAN. Two switches each connected to a different router interface are two LANs. Routers are what separate LANs.
- **Encapsulation recap:** data, plus a layer 4 header = segment, plus a layer 3 header = packet, plus a layer 2 header and trailer = **frame**.
- **Ethernet frame fields** (in transmission order):

Field	Size	Purpose
Preamble	7 bytes (`10101010` x 7)	Lets the receiver synchronize its clock
SFD (start frame delimiter)	1 byte (`10101011`)	Marks the end of the preamble and the start of the frame
Destination MAC	6 bytes	Who the frame is for
Source MAC	6 bytes	Who sent it
Type / length	2 bytes	What is inside (see below)
Data / packet		The encapsulated packet
FCS (trailer)	4 bytes	Error detection

The header plus trailer add up to **26 bytes**.

- **Type/length field:** a value of **1500 or less** is a length in bytes. A value of **1536 or more** is a type: `0x0800` = IPv4, `0x86DD` = IPv6. Almost always it is a type.
- **FCS** holds a **CRC** (cyclic redundancy check) result. The receiver recomputes it to detect corrupted frames.
- **MAC address:** 48 bits (6 bytes), written as **12 hexadecimal digits**, for example `aaaa.aaaa.0001` (Cisco style, a dot after every four digits). It is burned into the interface at manufacture (hence "burned-in address", BIA), and is globally unique in almost all cases.
- **OUI:** the first 3 bytes (24 bits), assigned to the manufacturer. The last 3 bytes identify the individual device.
- **Hexadecimal:** 16 digits, 0 to 9 then A to F (A = 10 ... F = 15). Hex `10` is decimal 16.
- **How a switch builds its MAC address table:**
  1. A frame arrives. The switch reads the **source MAC** and records it against the **interface it arrived on**. This is a **dynamic** MAC address.
  2. It looks up the **destination MAC**. If found, it is a **known unicast** frame, sent out only that interface.
  3. If not found, it is an **unknown unicast** frame, and the switch **floods** it out every interface **except** the one it arrived on.

4. Hosts that aren't the target discard the frame. When the target replies, the switch learns that host from the reply's source MAC.

- With several switches, a table entry on a switch can point to an interface that leads *toward* a host, not directly to it.
- **Aging:** on Cisco switches a dynamic entry is removed after **5 minutes** with no traffic from that address.

Remember for the exam

- Switches learn from the **source** MAC address and forward by the **destination** MAC address.
- Unknown unicast is **flooded**; known unicast is forwarded out one port.
- Know the sizes: preamble 7, SFD 1, addresses 6 and 6, type/length 2, FCS 4 bytes. MAC = 48 bits, IPv4 = 32 bits.
- The preamble does the clock sync; the SFD only marks where it ends.
- Switches don't divide a LAN; routers do.

Check yourself

1. How many bits is a MAC address, and which part is the OUI? *(48 bits; the first 24 bits, the first six hex digits.)*
2. A switch with an empty MAC table receives a frame from host A to host B. What does it learn, and what does it do with the frame? *(It learns A's MAC on the arrival port, and floods the frame out all other ports because B is unknown.)*
3. A frame's type/length field holds `0x0800`. What does that tell you? *(The field is a type, since it is above 1536, and the payload is an IPv4 packet.)*

Do next

No lab for this lesson; the Day 6 lab covers both parts. Add the Day 5 Anki cards (frame field sizes, OUI, MAC learning steps).

Day 6: Ethernet LAN Switching (Part 2)

Exam domain: 2. Network Access · **Video:** Ethernet LAN Switching (Part 2) (34 min) · Lab: Analyzing Ethernet Switching

Key ideas

- **Frame size details.** The preamble and SFD are sent with every frame but usually aren't counted as part of the Ethernet header. Counted that way, the header is 14 bytes (destination 6, source 6, type 2) and header plus FCS trailer is **18 bytes**. The **minimum frame size is 64 bytes** (excluding preamble and SFD), so the **minimum payload is 46 bytes**. A smaller packet is **padded** with zero bytes (a 34-byte packet gets 12 bytes of padding; a 36-byte ping gets 10).
- **Why ARP is needed.** A user types an IP address, but the frame needs a destination MAC address, and switches only understand MACs. **ARP (Address Resolution Protocol)** finds the **MAC address (layer 2) for a known IP address (layer 3)** on the same network.
- **ARP request:** sent as a **broadcast** frame to destination MAC `FFFF.FFFF.FFFF`. The switch floods it out every port except the one it arrived on. Every host reads it; only the host whose IP matches answers. The request's source MAC lets the target reply without broadcasting.
- **ARP reply:** a **unicast** frame back to the requester. The requester stores the result in its **ARP table** (IP address to MAC address).
- Switches learn each sender's MAC from the source address as the ARP frames pass through, so the later traffic is known unicast.
- **What a switch floods: broadcast** frames and **unknown unicast** frames, out all ports except the incoming one. **Known unicast** is forwarded out a single port.
- **Ping** tests reachability and measures round-trip time using **ICMP echo request** and **ICMP echo reply**, both unicast. Because ARP must run first, the first ping on Cisco IOS often fails. IOS sends **5 pings of 100 bytes** by default; `!` is a success and `.` a failure.
- **Ethernet type values:** IPv4 `0x0800`, IPv6 `0x86DD`, **ARP** `0x0806`.
- **MAC table columns on a Cisco switch:** VLAN (default 1), MAC address, type (dynamic or static), ports. Dynamic entries age out after **5 minutes** without traffic.
- Wireshark captures show the actual frames (broadcast ARP request, unicast ARP reply, then ICMP). Packet Tracer simulates devices; GNS3 runs real IOS images, which you must supply yourself.

Commands

```
arp -a                                  ! view the ARP table on Windows, macOS
or Linux
ping 192.168.1.3                        ! IOS or PC: 5 ICMP echo requests (100
bytes each)
ping 192.168.1.3 size 36                ! IOS: choose the packet size
show arp                                ! IOS: ARP table (privileged EXEC)
show mac address-table                  ! switch: MAC table (older IOS: show
mac-address-table)
clear mac address-table dynamic         ! remove all dynamic entries
```

```
clear mac address-table dynamic address 0c2f.b011.9d00   ! remove one MAC address
clear mac address-table dynamic interface g0/0  ! remove dynamic entries learned on one
interface
```

Remember for the exam

- **ARP request = broadcast; ARP reply = unicast.** ICMP echo request and reply are both unicast.
- A switch floods **broadcast and unknown unicast**, never known unicast.
- Minimum payload **46 bytes**, minimum frame **64 bytes**, padding made of zeros.
- The `show mac address-table` output has four fields: **VLAN, MAC address, type, ports**. The ARP table is a different thing, with IP address, MAC address and type.
- The clear command ends in `dynamic interface <id>` for a whole interface.

Check yourself

1. PC1 knows PC3's IP address but not its MAC. Which message does it send, and to which destination MAC? *(An ARP request, to the broadcast MAC `FFFF.FFFF.FFFF`.)*
2. A 20-byte packet is sent in an Ethernet frame. What happens to reach the minimum size? *(26 bytes of zero padding are added so the payload reaches 46 bytes.)*
3. You want to remove only the entries learned on `g0/1`. Which command? *(`clear mac address-table dynamic interface g0/1`.)*

Do next

Do the Day 6 lab (analyzing Ethernet switching in Packet Tracer): ping, then check `show mac address-table` and `show arp`. Add the Day 6 Anki cards.

Day 7: IPv4 Addressing (Part 1)

Exam domain: 1. Network Fundamentals · **Video:** IPv4 Addressing (Part 1) (40 min)

Key ideas

- **Layer 3** connects hosts on different networks. It provides **logical addressing** (IP addresses you assign, unlike MAC addresses burned in at manufacture) and **path selection**. **Routers** operate at layer 3.
- **A router splits LANs.** Switches only extend one LAN, so a broadcast reaches every host behind them. A router does not forward the broadcast, so each router interface sits in a separate network and needs its **own IP address** in that network.
- **IPv4 address = 32 bits**, in four 8-bit **octets**, written in **dotted decimal** (for example `192.168.1.254`). The source and destination address fields in the IPv4 header are each 32 bits.
- **Binary place values in an octet:** 128, 64, 32, 16, 8, 4, 2, 1. One octet ranges from 0 (`00000000`) to 255 (`11111111`).
  - Binary to decimal: add the values where there is a 1. `10001111` = 128 + 8 + 4 + 2 + 1 = 143.
  - Decimal to binary: go left to right, and put a 1 wherever the value fits, subtracting it. 221 - 128 = 93, - 64 = 29, skip 32, - 16 = 13, - 8 = 5, - 4 = 1, skip 2, - 1 = 0, giving `11011101`.
- **Network and host portions:** the **prefix length** (`/24`) says how many leading bits are the network part; the rest are the host part. Hosts on the same network share the network portion and differ in the host portion.
- **Netmask:** the same information as a prefix length, written in dotted decimal, with 1s for the network bits and 0s for the host bits. Cisco devices use netmasks (`/24` = `255.255.255.0`).
- **Classful addressing**, decided by the first octet:

Class	First octet	Leading bits	Default prefix / mask	Networks	Hosts per network
A	0-127 (usable 1-126)	`0`	/8, 255.0.0.0	128	about 16.7 million
B	128-191	`10`	/16, 255.255.0.0	about 16,000	about 65,000
C	192-223	`110`	/24, 255.255.255.0	about 2 million	254
D	224-239	`1110`	multicast		
E	240-255	`1111`	experimental		

- **Loopback:** the whole `127.0.0.0` to `127.255.255.255` range (that is why class A is counted as 1-126). Traffic to it never leaves the device; it tests the local network stack (`ping 127.0.0.1`).
- **Network address:** host bits all **0**. It identifies the network and can't be assigned to a host (`192.168.1.0/24`).
- **Broadcast address:** host bits all **1**. It reaches every host on the network and can't be assigned (`192.168.1.255`). Packets sent to it travel in frames addressed to `FFFF.FFFF.FFFF`.
- **Usable hosts = 2^(host bits) - 2.** In a /24 that is 254, from `.1` to `.254`.

Remember for the exam

- Class ranges by first octet: **A 0-127, B 128-191, C 192-223, D 224-239, E 240-255**. A, B and C are the ones used for unicast hosts.
- /8 = 255.0.0.0, /16 = 255.255.0.0, /24 = 255.255.255.0.
- You must convert quickly between dotted decimal and binary in both directions.
- The network and broadcast addresses are not usable as host addresses.

Check yourself

1. What is `11000000.10101000.00001010.01100100` in dotted decimal? *(192.168.10.100)*
2. A host has the address `172.20.5.9/16`. Which class, and which part is the network portion? *(Class B, since the first octet is 128-191; the first two octets, 172.20.)*
3. In `10.1.1.0/24`, what are the broadcast address and the last usable host address? *(10.1.1.255 and 10.1.1.254.)*

Do next

No lab for this lesson; the Day 8 lab covers addressing. Practise binary conversion until it is quick, and add the Day 7 Anki cards.

Day 8: IPv4 Addressing (Part 2)

Exam domain: 1. Network Fundamentals · **Video:** IPv4 Addressing (Part 2) (31 min) · Lab: Configuring IP Addresses

Key ideas

- **Class A range:** remember it as 0-127, but 0 and 127 are reserved, so the usable range is **1-126**.
- **Counting hosts.** With *N* host bits, a network has 2^N addresses, and **2^N - 2 can be assigned** (the all-zeros network address and the all-ones broadcast address are excluded).

Class	Example network	Host bits	Addresses	Max hosts
A	10.0.0.0/8	24	16,777,216	16,777,214
B	172.16.0.0/16	16	65,536	65,534
C	192.168.1.0/24	8	256	254

- **Finding the five values for a host address** (given the class prefix): zero the host bits for the **network address**; set them to all 1s for the **broadcast address**; **first usable = network + 1**; **last usable = broadcast - 1**. Worked example, `129.221.23.13/16`:
  - network `129.221.0.0`, broadcast `129.221.255.255`
  - first usable `129.221.0.1`, last usable `129.221.255.254`
  - max hosts 2^16 - 2 = 65,534
- **Configuring an IP on a Cisco interface** needs the **subnet mask** in dotted decimal (another name for netmask): /8 is 255.0.0.0, /16 is 255.255.0.0, /24 is 255.255.255.0.
- **Router interfaces are shut down by default** ("administratively down"), so you must enable them with `no shutdown`. **Switch interfaces are not shut down by default**: they are up when connected and down when not.
- `show ip interface brief` **columns:**

Column	Meaning
Interface	The interface name
IP-Address	The assigned address, or `unassigned`
OK?	Legacy; not useful on modern devices
Method	How the address was set: `manual` once you configure it, `unset` before
Status	**Layer 1** state (administratively down = `shutdown` applied; down = no cable or no device at the far end; up)
Protocol	**Layer 2** state

Status `down` with protocol `up` never happens; status up with protocol down can.

- After `no shutdown`, IOS prints two messages: the interface changed state to up (layer 1), and the line protocol changed state to up (layer 2).

- `show interfaces <id>` shows layer 1 and 2 detail (line status, hardware type, the MAC address, and `BIA` for the burned-in address) plus the IP address and prefix length. The MAC address can be changed, which is why the BIA is listed separately.
- **Interface descriptions** are optional free text that helps identify what a port connects to; `show interfaces description` lists them next to status and protocol.
- You can move from one interface's config mode straight to another without `exit`. Interface names accept short forms such as `g0/0`.

Commands

```
show ip interface brief             ! one-line summary of every interface, IP, status,
protocol
configure terminal
interface gigabitethernet0/0        ! enter interface config mode (short: int g0/0)
 ip address 10.255.255.254 255.0.0.0 ! address and subnet mask
 description ## to SW1 ##           ! optional label
 no shutdown                        ! enable the interface (short: no shut)
do show ip interface brief          ! "do" runs a privileged EXEC command from config
mode
show interfaces g0/0                ! detailed layer 1 and 2 information
show interfaces description         ! status, protocol and description per interface
```

Remember for the exam

- Usable hosts = 2^N - 2. Know the answer for a /8, /16 and /24: 16,777,214, 65,534 and 254.
- Router ports start administratively down; use `no shutdown`.
- **Status = layer 1, protocol = layer 2.** `up/up` is the healthy state.
- The `ip address` command takes a **mask**, not a prefix length.

Check yourself

1. `192.168.50.77/24`: what are the network address, broadcast address and last usable address? *(192.168.50.0, 192.168.50.255, 192.168.50.254.)*
2. An interface shows `administratively down` / `down`. What was done to it, and how do you fix it? *(`shutdown` is applied, which is the default on router interfaces; enter `no shutdown` in interface config mode.)*
3. How many hosts fit in a class B network without subnetting? *(2^16 - 2 = 65,534.)*

Do next

Do the Day 8 lab (configuring IP addresses on a router in Packet Tracer) and check the result with `show ip interface brief`. Add the Day 8 Anki cards.

Day 9: Switch Interfaces

Exam domain: 1. Network Fundamentals · **Video:** Switch Interfaces (32 min) · Lab: Configuring Interfaces

Key ideas

- **Switches have many ports** (a Catalyst 9200 can have 48 RJ-45 ports plus 4 SFP) because they connect end hosts; a router has few.
- **Default state differs from routers.** Router interfaces start `shutdown`. Switch interfaces do **not**: connected ports are **up/up** with no configuration, and unconnected ports are **down/down**. `down/down` (not connected) is not the same as `administratively down` (you shut it down).
- Switchports are layer 2, so they have no IP address (`unassigned` in `show ip interface brief`).
- `show interfaces status` columns: Port, Name (the interface description), Status, VLAN (default 1; `trunk` for inter-switch links, covered later), Duplex, Speed, Type.
  - Status values: `connected`, `notconnect`, `disabled` (shut down).
  - `a-full` / `a-100` mean duplex and speed were **auto-negotiated**; `full` / `100` without the "a" mean they were set by hand.
  - Defaults are speed auto and duplex auto.
- **Duplex:** *half* = a device can't send and receive at once; *full* = it can. Full duplex is the norm on switches.
- **Hubs** are layer 1 repeaters that flood every frame out all other ports. Hosts on a hub are one **collision domain**: simultaneous sends collide and must be repeated, so they run half duplex. A switch gives each port its own collision domain, so hosts on a switch can use full duplex and collisions should not occur.
- **CSMA/CD** (carrier sense multiple access with collision detection) is how half-duplex devices cope: listen first and send only when the medium is quiet; if a collision still happens, send a jam signal; every sender waits a random time, then tries again. (CSMA/CA is the wireless counterpart, covered later.)
- **Autonegotiation.** Interfaces that support several speeds (10/100 or 10/100/1000) advertise their capabilities and agree on the fastest speed and best duplex they share. If the neighbor does **not** negotiate:
  - speed: the port tries to **sense** it; if that fails it falls back to the slowest supported speed;
  - duplex: **half** if the resulting speed is 10 or 100 Mbps, **full** if 1000 Mbps or more.
- A **duplex mismatch** (one side full, the other half) causes collisions and poor performance. Best practice is autonegotiation on both sides.
- **Interface errors** appear in `show interfaces <id>` (on routers too):

Counter	Meaning
Runts	Frames smaller than the 64-byte minimum
Giants	Frames larger than the 1518-byte maximum
CRC	Frames that failed the FCS check
Frame	Frames with an illegal format

Counter	Meaning
Input errors	Total of the input counters above and others
Output errors	Frames the device failed to send

Broadcasts and total packet or byte counts are normal, not errors.

- **Best practice:** shut down unused switch ports, since they would otherwise accept any device plugged in.

Commands

```
show ip interface brief                  ! status (layer 1) and protocol (layer 2) per
interface
show interfaces status                   ! switch only: connected / notconnect /
disabled, speed, duplex
show interfaces f0/1                     ! counters and errors (run it for one
interface)
interface f0/1                           ! enter interface config mode
 speed 100                               ! 10, 100 or auto (default)
 duplex full                             ! auto (default), full or half
 description ## to R1 ##                 ! shows in the Name column
interface range f0/5 - 12                ! configure several interfaces at once
 shutdown                                ! disable them
interface range f0/5 - 6 , f0/9 - 12     ! ranges need not be consecutive
 no shutdown                             ! re-enable
```

Remember for the exam

- A fresh switch port is up/up when connected; a fresh router port is administratively down.
- Without autonegotiation on the far side: 10 or 100 Mbps gives half duplex, 1000 Mbps or more gives full.
- Hubs are one collision domain; each switch port is its own.
- Runts under 64 bytes, giants over 1518 bytes, CRC = failed FCS.
- The command for errors is `show interfaces`; `show interfaces status` and `show ip interface brief` have no error counters.

Check yourself

1. A switch port auto-detects that a neighbor runs 100 Mbps, but the neighbor does not negotiate. What duplex will the switch pick, and is that a problem if the neighbor is set to full? *(Half; yes, a duplex mismatch causing collisions.)*
2. What does `a-full` under Duplex tell you? *(Full duplex was negotiated automatically, not configured.)*
3. A counter is rising for frames that failed the FCS check. Which one is it? *(CRC.)*

Do next

Do the Day 9 lab (configuring switch interfaces in Packet Tracer): speed, duplex, descriptions, `interface range`. Add the Day 9 Anki cards.

Day 10: IPv4 Header

Exam domain: 1. Network Fundamentals · **Video:** IPv4 Header (30 min)

Key ideas

- The **IPv4 header** is the layer 3 header that wraps a TCP or UDP segment to form a **packet**. It holds what routers need to send the packet toward its destination. The lesson stresses knowing each field's purpose; exact sizes are less likely to be tested.
- **Header fields**, in order:

Field	Size	Purpose
Version	4 bits	IP version: `0100` (4) for IPv4, `0110` (6) for IPv6, which has a different header. There is no IPv5 in use.
IHL (Internet Header Length)	4 bits	Header length in **4-byte units**. Minimum 5 (= 20 bytes), maximum 15 (= 60 bytes).
DSCP	6 bits	Marks traffic for **QoS**, so delay-sensitive traffic such as voice and video can be prioritized.
ECN	2 bits	Signals congestion without dropping packets. Optional; needs support at both ends and in the network.
Total length	16 bits	Length of the whole packet (header plus segment) in **bytes**. Maximum 65,535.
Identification	16 bits	Same value on all fragments of one packet, so the receiver can reassemble them.
Flags	3 bits	Bit 0 reserved (always 0). Bit 1 **DF** (don't fragment). Bit 2 **MF** (more fragments).
Fragment offset	13 bits	Where this fragment fits in the original packet, so out-of-order fragments can be put back.
TTL (time to live)	8 bits	Decremented by 1 at each router; a packet reaching 0 is **dropped**. Stops packets circling forever in a routing loop. Recommended default is 64.
Protocol	8 bits	What the payload is (see below).
Header checksum	16 bits	Error check on the **header only**. A router that finds a mismatch drops the packet.
Source address	32 bits	Sender's IPv4 address.
Destination address	32 bits	Receiver's IPv4 address.
Options	0 to 320 bits (up to 40 bytes)	Rarely used. If IHL is greater than 5, options are present.

- **The 20-byte minimum** header is the fixed fields above without options. **Options** is the only variable-length field.

- **Protocol numbers to remember: 1** ICMP (ping), **6** TCP, **17** UDP, **89** OSPF (a dynamic routing protocol covered later).
- **Fragmentation:** a packet bigger than the link's **MTU** (usually **1500 bytes**, matching the Ethernet maximum payload) is split into fragments, each with its own IP header. Fragments share an identification value, have MF set on all but the last, and differ in fragment offset (first = 0). The **receiving host** reassembles them. If DF is set and the packet is too big, it can't be sent and is dropped.
- **Errors in the data** aren't caught by IP's checksum. TCP and UDP carry their own checksums for that.
- **Seen in a Wireshark capture of a ping:** version 4, IHL 5, total length 100 (Cisco's default ping size), protocol 1, TTL 255, no flags set. A `ping ... size 10000` was split into 1500-byte fragments, all marked "more fragments" except the last.

Commands

```
ping 192.168.1.2 size 10000      ! IOS: oversized ping, forces fragmentation (MTU 1500)
ping 192.168.1.2 df-bit          ! IOS: set the don't-fragment bit; oversized pings
then fail
```

Remember for the exam

- Version is always `0100` in an IPv4 header.
- **TTL reaching 0 = packet dropped.** The header checksum covers only the header.
- IHL 5 = 20 bytes; the IHL value is multiplied by 4. Total length is already in bytes.
- Protocol numbers: ICMP 1, TCP 6, UDP 17, OSPF 89.
- The **MF** bit is set on every fragment except the last; **DF** forbids fragmenting.

Check yourself

1. An IPv4 header has an IHL value of 8. How long is the header, and are options present? *(8 x 4 = 32 bytes; yes, 12 bytes of options.)*
2. Why does a packet with a routing loop eventually disappear? *(Each router decrements the TTL; at 0 the packet is dropped.)*
3. A router receives a packet with protocol 17. What is inside? *(A UDP datagram.)*

Do next

No lab for this lesson; the next routing lab comes after Day 11. Add the Day 10 Anki cards, which matter most here (field purposes and protocol numbers).

Day 11: Routing fundamentals and static routing

Exam domain: 3. IP Connectivity · **Video:** Routing Fundamentals (31 min) · Static Routing (38 min) · Lab: Configuring Static Routes · Lab: Troubleshooting Static Routes

Key ideas

- **Routing** is how a router decides where an IP packet goes next. The router keeps a **routing table** (the router's counterpart of a switch's MAC address table). Each route is an instruction: to reach network X, send to next hop Y, send out a directly connected interface, or keep the packet because it's for me.
- Routes get into the table in three ways: learned by a **dynamic routing protocol** (OSPF, later in the course), **configured by hand** (static routes), or **added automatically** when you configure an IP address on an interface and enable it.
- The automatic pair, per interface: a **connected route** (code `C`) to the network the interface sits on, with the configured mask, and a **local route** (code `L`) to the interface's exact address with a **/32** mask. Shut the interface down and both disappear.
- A route **matches** a packet when the destination IP falls inside the route's network. If several routes match, the router uses the **longest prefix length** (the most specific). That's why the /32 local route beats the /24 connected route for the router's own address, and the router keeps the packet instead of forwarding it.
- **No matching route means the packet is dropped.** Routers never flood; switches flood unknown unicast.
- The lines such as "is variably subnetted, 2 subnets, 2 masks" in `show ip route` are headings, not routes.
- Connected and local routes only cover directly attached networks. To reach a **remote** network you need a static route (or a dynamic protocol).
- **Two-way reachability:** every router on the path needs a route to the destination network and a route back to the source network. A ping only works if replies can return. A router doesn't need routes to the networks in between, only to the ends.
- **Default gateway:** a host sends traffic for other networks to its gateway (its router). That is a default route, `0.0.0.0/0`, the least specific route possible because it matches every address.
- As a packet crosses routers, the source and destination **IP addresses stay the same**, but the **MAC addresses are rewritten at each hop** (new frame each time). Only on the last hop do the destination IP and MAC belong to the same device.
- Static route targets: next-hop IP only, exit interface only, or both. With exit interface only, the table shows the network as "directly connected" and relies on proxy ARP (beyond the CCNA). Next hop only, or both, is the usual choice.
- A **default route** is a static route to `0.0.0.0 0.0.0.0`, typically pointing toward the Internet. It shows with `S*`, and `Gateway of last resort` appears at the top of the table. "Not set" means no default route.

Commands

```
show ip route                                    ! Display the routing table
ip route 192.168.4.0 255.255.255.0 192.168.13.3   ! Static route: network, mask, next-
hop IP
ip route 192.168.1.0 255.255.255.0 g0/0        ! Static route, exit interface only
ip route 192.168.4.0 255.255.255.0 g0/1 192.168.24.4   ! Static route, exit interface
and next hop
ip route 0.0.0.0 0.0.0.0 203.0.113.2           ! Default route
```

Remember for the exam

- Static route syntax takes a **dotted netmask**, not a slash prefix: `ip route 10.0.0.0/8 ...` is invalid.
- Configuring an IP address and enabling the interface adds **two** routes: `C` and `L`.
- A default route uses all-zero network and mask. Mask `255.255.255.255` would be a host route instead.
- In `[1/0]` the first number is the administrative distance and the second the metric (both covered later).
- To find the exit interface for a route with only a next hop, look up the next-hop address in the table: it matches a connected route.
- Count the routes: a router needs a static route for every remote network it has to reach.

Check yourself

1. A router interface is 10.5.5.1/24 and is up. Which two routes appear, and what are their destinations? *(`C` for 10.5.5.0/24 and `L` for 10.5.5.1/32.)*
2. A packet to 10.5.5.1 matches both of those routes. Which one wins, and what happens to the packet? *(The /32 local route, as it's longer; the router receives the packet itself.)*
3. PC A can send to PC B, but the ping fails. What is the likely cause? *(A router on the return path has no route back to A's network.)*

Do next

Do the two static route labs: configure the four-router topology, then fix the broken routes. Add the Day 11 Anki cards.

Day 12: The life of a packet

Exam domain: 1. Network Fundamentals · **Video:** The Life of a Packet (20 min) · Lab: Life of a Packet

Key ideas

This day adds no new theory. It ties together ARP, encapsulation, routing and switching by following one packet from PC1 (192.168.1.0/24) to PC4 (192.168.4.0/24) through R1, R2 and R4, with static routes already in place.

1. **PC1 decides the destination is remote.** The destination IP is not in its own subnet, so the frame must go to the **default gateway**, R1.
2. **PC1 needs the gateway's MAC address.** It sends an **ARP request**: a broadcast frame (destination MAC `FFFF.FFFF.FFFF`) asking who owns the gateway's IP. The switch floods it, and the switch learns PC1's source MAC.
3. **R1 answers with a unicast ARP reply** (it already learned PC1's addresses from the request). The switch learns R1's MAC from the reply.
4. **PC1 builds the packet and frame.** The IP header carries source PC1 and destination PC4. The Ethernet header carries source PC1's MAC and destination **R1's MAC**.
5. **R1 de-encapsulates** (strips the Ethernet header), looks up the destination in its routing table, and takes the most specific match: a route to 192.168.4.0/24 with next hop R2.
6. **R1 needs R2's MAC address**, so it ARPs for the next hop's IP. Then it re-encapsulates in a new frame: source = R1's outgoing interface MAC, destination = R2's incoming interface MAC.
7. **The same repeats at R2** (next hop R4, ARP, new frame) and at **R4**, whose route is a connected route to PC4's network, so it ARPs for **PC4's own IP** and sends the final frame with PC4's MAC as destination.
8. **Replies are the same journey in reverse**, except the ARP step is skipped because the devices already hold the MAC mappings in their ARP caches.

What changes and what doesn't:

Field	Changes per hop?
Source and destination **IP**	No, end to end, unchanged by routers
Source and destination **MAC**	Yes, rewritten by every router; each frame only covers one link
Frames passing through a **switch**	Not modified; the switch only learns MACs and forwards or floods

- The next-hop router's IP is used for the routing decision, but the **next hop's MAC** is what goes in the frame.
- Only on the final hop do the destination IP and destination MAC belong to the same device.
- Every router interface has its own MAC address.
- Header order differs between layers: in IPv4 the source address comes first, in Ethernet the destination MAC comes first.

Remember for the exam

- A host sending to a remote network puts the **default gateway's MAC** as the destination MAC, but the **remote host's IP** as the destination IP.
- ARP requests are broadcast; ARP replies are unicast.
- Routers de-encapsulate and re-encapsulate at every hop, so the Layer 2 source and destination change each time. The Layer 3 addresses do not (no NAT in this scenario).
- Switches never rewrite MAC addresses.
- Questions about "source MAC when received on interface X" want the MAC of the **previous router's outgoing interface**, not of the original sender.

Check yourself

1. PC4 sends to PC1. What is the destination IP address in the packet as it leaves R2 toward R1? *(PC1's IP address, unchanged.)*
2. Which MAC address does a router put in the destination field when forwarding to a next hop? *(The MAC of the next-hop router's receiving interface, found via ARP.)*
3. Why did the reply from PC4 need no ARP requests? *(Each device had already cached the neighbor's MAC address from the first exchange.)*

Do next

Do the Packet Tracer simulation-mode lab and inspect each PDU's MAC and IP fields at every hop. This day has no Anki deck; make your own cards for the table above if useful.

Day 13: Subnetting, part 1

Exam domain: 1. Network Fundamentals · **Video:** Subnetting (Part 1) (29 min)

Key ideas

- **Classful addressing** fixed the network size by the first octet:

Class	First octet	Default prefix	Addresses per network
A	0 to 127	/8	16,777,216
B	128 to 191	/16	65,536
C	192 to 223	/24	256
D (multicast)	224 to 239	-	-
E (reserved)	240 to 255	-	-

Only A, B and C are assigned to hosts. IANA handed out whole classful blocks by organization size.

- **The problem is waste.** A two-router link given a whole class C wastes 252 addresses; a company needing 5,000 hosts needs a class B and wastes about 60,000. Combined with the fast growth of the Internet, this led to IPv4 address exhaustion.
- **CIDR (classless inter-domain routing, 1993)** drops the rule that A must be /8, B /16 and C /24. Any prefix length can be used, so a large block can be cut into smaller **subnets**. Slash notation (`/26`) is called CIDR notation.
- **Host bits and usable addresses:** with *h* host bits there are 2^h addresses and **2^h - 2 usable** (one is the network address, all-zeros host part; one is the broadcast address, all-ones host part).

Prefix	Mask	Host bits	Usable hosts
/24	255.255.255.0	8	254
/25	255.255.255.128	7	126
/26	255.255.255.192	6	62
/27	255.255.255.224	5	30
/28	255.255.255.240	4	14
/29	255.255.255.248	3	6
/30	255.255.255.252	2	2
/31	255.255.255.254	1	0 by the formula, but **2 on a point-to-point link**
/32	255.255.255.255	0	none; names a single host

- **/30** is the classic point-to-point subnet (network, two hosts, broadcast). **/31** is valid on point-to-point links because no network or broadcast address is needed there; a Cisco router warns you to make sure it's a point-to-point link.
- **/32** isn't put on a normal interface. It's used to point a route at one exact host (the local route in the routing table is a /32).

- After carving out a small subnet, the rest of the original block stays free for other subnets. That is the point of subnetting.

Choosing a prefix for a host count

1. Add 2 to the hosts needed (network and broadcast).
2. Find the smallest power of two that covers it: that is the number of host bits.
3. Prefix length = 32 minus host bits.
4. Check the block you were given has room for all the subnets.

Worked example. You have 10.8.0.0/24 and need equal subnets for 28 hosts each (router included).

- 28 + 2 = 30 addresses needed. 2^4 = 16 is too small, 2^5 = 32 fits, so **5 host bits**.
- Prefix = 32 - 5 = **/27**, mask 255.255.255.224, 30 usable.
- Subnets are 32 addresses apart: 10.8.0.0, 10.8.0.32, 10.8.0.64, 10.8.0.96, and so on up to 10.8.0.224, eight in all.
- Finding the next subnet: the broadcast address of 10.8.0.0/27 is 10.8.0.31, so the next network address is 10.8.0.32.

Remember for the exam

- Usable hosts = 2^h - 2 (except the /31 point-to-point and /32 special cases).
- Class ranges by first octet: A 0-127, B 128-191, C 192-223, D 224-239, E 240-255.
- Network address: host bits all 0. Broadcast address: host bits all 1.
- The next subnet's network address is the previous subnet's broadcast address plus one.
- Convert between binary, dotted decimal and prefix length quickly; the next two days depend on it.

Check yourself

1. How many usable host addresses does a /28 give? *(2^4 - 2 = 14.)*
2. Which prefix is the smallest that fits 100 hosts per subnet? *(100 + 2 = 102; 2^7 = 128 fits, so 7 host bits and /25.)*
3. Why can a /31 be used between two routers but not for a LAN of hosts? *(Only a point-to-point link can do without network and broadcast addresses.)*

Do next

No lab for this day. Download the Day 13 Anki cards, and memorize the mask and usable-host table above before starting Day 14.

Day 14: Subnetting, part 2

Exam domain: 1. Network Fundamentals · **Video:** Subnetting (Part 2) (25 min)

Key ideas

- **Borrowed bits.** Subnetting takes bits from the host portion and adds them to the network portion. With *x* borrowed bits you get **2^x subnets** (no "minus 2" for subnets). With *h* host bits left, each subnet has **2^h - 2 usable hosts**.
- **Subnets for a requirement.** Borrow the fewest bits that give enough subnets, then check the hosts still fit. The count of subnets can overshoot the need; that's fine.
- **Finding every subnet.** Write the subnet's network address with host bits all 0. Its broadcast address has host bits all 1. The next subnet's network address is that broadcast plus 1.
- **The increment shortcut.** The value of the **last network bit** is the step between subnets. For /26 the last network bit in the fourth octet is worth 64, so subnets are at .0, .64, .128, .192. For /27 the step is 32, for /28 it's 16, and so on. This works the same in any octet: /20 on a class B network steps by 16 in the third octet, /23 steps by 2.
- **Which subnet does a host belong to?** Write the address in binary, set every host bit to 0, convert back. That is the network address. Set the host bits to 1 instead and you have the broadcast address.
- **Class doesn't change the method.** Subnetting a class B network (a /16 block) is the same process with more host bits to borrow from. You can borrow into the last octet too, so a /16 can be split into /25s.
- Patterns to know rather than memorize: each borrowed bit **doubles** the subnets; each host bit **doubles** the addresses per subnet (before subtracting 2).
- On a Cisco router you enter the **dotted decimal mask** (255.255.128.0), not the slash.

Class C table (fourth octet)

Prefix	Subnets	Usable hosts per subnet	Step
/25	2	126	128
/26	4	62	64
/27	8	30	32
/28	16	14	16
/29	32	6	8
/30	64	2	4

Worked examples

Splitting a class C block. Cut 192.168.50.0/24 into 6 equal subnets. 2^2 = 4 is too few; 2^3 = 8 works, so borrow **3 bits**, giving /27 (step 32, 30 hosts each). Subnets: 192.168.50.0, .32, .64, .96, .128, .192 (and two spare, .160 and .224).

Which subnet is 172.20.133.9/19 in?

- /19 means 16 bits plus 3 borrowed bits: the third octet is the one that matters.

- Third octet 133 = 1000 0101. The first 3 bits are network (100), the other 5 are host bits. Set the host bits to 0: 1000 0000 = 128.
- Network address: **172.20.128.0/19**. The step is 32, so the next subnet starts at 172.20.160.0.
- Broadcast: host bits all 1, which is 1001 1111 = 159 in the third octet, so **172.20.159.255**.

Both requirements. From a /16, you need 250 subnets and 250 hosts each: 8 borrowed bits (2^8 = 256 subnets) leaves 8 host bits (254 hosts), so /24 satisfies both.

Remember for the exam

- Number of subnets = 2^(borrowed bits); hosts = 2^(host bits) - 2.
- Round up to the next whole borrowed bit when counting subnets.
- Network address = host bits all 0; broadcast = host bits all 1; first usable = network + 1; last usable = broadcast - 1.
- Identify the "interesting octet" (the one containing the prefix boundary) and use its step.
- Questions can ask for the broadcast of the subnet a host sits in, or the Nth subnet of a split. Work in the interesting octet.

Check yourself

1. You split 10.1.0.0/16 into 100 subnets. What prefix length? *(2^7 = 128 is the smallest enough, so 7 borrowed bits: /23.)*
2. What network and broadcast address does 192.168.9.200/28 belong to? *(Step 16: 200 is in the block starting at 192, so network 192.168.9.192 and broadcast 192.168.9.207.)*
3. A subnet needs 20 hosts. Which prefix is the smallest that fits on a class C? *(22 addresses needed; 5 host bits give 30 usable, so /27.)*

Do next

No lab for this day; the subnetting lab comes after Day 15. Practice the "which subnet" and "split into N subnets" problems daily until they are quick.

Day 15: Subnetting, part 3 (class A and VLSM)

Exam domain: 1. Network Fundamentals · **Video:** Subnetting (Part 3 - VLSM) (24 min) · Lab: Subnetting (VLSM)

Key ideas

- **Class A is the same method with more bits.** A /8 block has 24 host bits to borrow from. The only new skill is working in whichever octet holds the prefix boundary (second octet for /9 to /16, third for /17 to /24).
- **Five answers for any subnet**, all from the host bits:

Answer	How
Network address	Host bits all 0
Broadcast address	Host bits all 1
First usable	Network + 1
Last usable	Broadcast - 1
Usable hosts	2^(host bits) - 2

- **FLSM vs VLSM.** *Fixed-length* subnet masks give every subnet the same prefix length, which wastes space when LANs differ in size. **Variable-length subnet masks** give each subnet the prefix that fits its host count.
- **VLSM steps:**
  1. List the subnets and the hosts each needs.
  2. Sort largest to smallest.
  3. Give the largest the first block at the start of the address space; pick the smallest prefix that fits (hosts + 2 addresses).
  4. The next subnet starts right after the previous broadcast address (broadcast + 1).
  5. Repeat down the list, then see what is left over.
- Allocating largest first keeps each subnet on a clean boundary and leaves the leftover space as one block at the end.
- **Point-to-point links:** two hosts need a /30 (4 addresses). A /31 also works on a point-to-point link, but on the exam use /30 for "two hosts" unless the question clearly points to /31.
- Watch for the off-by-two trap: 8 hosts do **not** fit in a /29 (6 usable); use /28.

Worked examples

Class A. Which subnet is 10.150.33.7/10 in?

- /10 borrows 2 bits in the second octet. 150 = 1001 0110; keep the first 2 bits (10), zero the other 6: 1000 0000 = 128.
- Network **10.128.0.0/10**, step 64 in the second octet.
- Broadcast: host bits all 1 gives second octet 1011 1111 = 191, so **10.191.255.255**.

- Range of usable addresses: 10.128.0.1 to 10.191.255.254. Host bits = 22, so 2^22 - 2 = **4,194,302** usable.

VLSM. Block 172.16.10.0/24; LANs of 60, 25 and 12 hosts, plus one router link.

Need	Prefix (usable)	Network	Broadcast	Usable range
60 hosts	/26 (62)	172.16.10.0	.63	.1 to .62
25 hosts	/27 (30)	172.16.10.64	.95	.65 to .94
12 hosts	/28 (14)	172.16.10.96	.111	.97 to .110
Router link	/30 (2)	172.16.10.112	.115	.113 to .114

Each network address is the previous broadcast + 1. Space from 172.16.10.116 onward is still free. With one fixed mask, a /26 would have been needed for all four and the block would be full after four subnets.

Remember for the exam

- Always size for hosts + 2 (network and broadcast), then pick the next power of two.
- VLSM order: largest subnet first, then next largest, and so on.
- Subnets must not overlap; each begins at the previous broadcast + 1 and on a multiple of its own block size.
- Practice every day for a week. Jeremy points to three free practice sites for this; do a few problems from each.
- Dotted decimal for the common prefixes should be automatic, for example /19 = 255.255.224.0 and /23 = 255.255.254.0.

Check yourself

1. How many usable hosts are in a /20 on a class B network? *(12 host bits: 2^12 - 2 = 4,094.)*
2. In VLSM, why allocate the largest subnet first? *(It keeps every subnet aligned on its boundary and leaves the free space as one block at the end.)*
3. You need 5 hosts on one LAN. Which prefix, and how many usable addresses? *(5 + 2 = 7 needs 3 host bits: /29 with 6 usable.)*

Do next

Do the Packet Tracer VLSM lab: subnet a block, then assign addresses to the devices. Subnetting has no Anki deck on this day; keep drilling on the practice sites.

Day 16: VLANs, part 1

Exam domain: 2. Network Access · **Video:** VLANs (Part 1) (24 min) · Lab: VLANs (Part 1)

Key ideas

- **A LAN is a single broadcast domain**: the set of devices that receive a broadcast frame (destination MAC `FFFF.FFFF.FFFF`) sent by any member. Switches flood broadcasts out every port except the one it arrived on. Routers do not forward broadcasts, so **each router interface starts a new broadcast domain**. A link between two routers is a broadcast domain too.
- **The problem.** One flat switched network with several departments floods every broadcast and unknown unicast to everyone. That hurts **performance** (needless traffic) and **security** (hosts reach each other directly, so router policies don't apply).
- **Splitting by subnet is not enough.** A switch only looks at Layer 2. Three subnets plugged into one switch are still one broadcast domain. Buying a switch per department works but is expensive and inflexible.
- **A VLAN (virtual LAN)** splits one physical switch into several separate Layer 2 broadcast domains. You assign VLANs **per switch interface**; the host plugged into that port belongs to that VLAN.
- A switch never forwards frames between VLANs, not even when the hosts share a subnet. Traffic from VLAN 10 is only forwarded out VLAN 10 ports. Getting from one VLAN to another needs a **router** (inter-VLAN routing; other methods come later). Typically each VLAN is its own subnet, with the router as the default gateway.
- Purposes: reduce unnecessary broadcast traffic (performance) and limit who receives broadcast and unknown-unicast frames (security).
- **Access port:** belongs to a single VLAN and usually connects to an end host. (Trunk ports carry several VLANs; that is Day 17.)
- **Default VLANs.** Every interface starts in **VLAN 1** (name `default`). VLANs **1 and 1002 to 1005** (FDDI and Token Ring) exist by default and **cannot be deleted**. They all appear in `show vlan brief`, so after you create VLANs 10, 20 and 30 there are 8 listed.
- Assigning an interface to a VLAN that doesn't exist **creates the VLAN automatically** (with a default name such as `VLAN0010`). You can also create and name it yourself.
- A broadcast such as `ping 255.255.255.255` from a host reaches only the other hosts in its own VLAN.

Commands

```
show vlan brief                 ! List VLANs and the access ports in each
interface range g1/0 - 3        ! Select several interfaces at once
 switchport mode access         ! Make the port an access port (set it explicitly)
 switchport access vlan 10      ! Put the port in VLAN 10 (creates the VLAN if
missing)
vlan 10                         ! Create VLAN 10 / enter its config mode
 name ENGINEERING               ! Give the VLAN a name
```

Remember for the exam

- Count broadcast domains: every VLAN on a switch is one; every router interface and every router-to-router link is one more (a router interface in VLAN 10 and another in VLAN 30 don't merge them).
- `show vlan brief` shows access ports only; trunk ports don't appear in a VLAN's port list.
- A command to assign an access port is `switchport access vlan <id>`. Creating a VLAN that doesn't exist is automatic.
- VLANs 1002 to 1005 are for old FDDI/Token Ring and are there but unusable for normal work.
- Switches don't route between VLANs; a router (or later a multilayer switch) does.
- A switch port set explicitly to access mode doesn't depend on auto-negotiation.

Check yourself

1. A broadcast from a host in VLAN 20 arrives at the switch. Where is it flooded? *(Out all other ports in VLAN 20 only.)*
2. You create VLANs 10, 20 and 30 on a new switch. How many VLANs does `show vlan brief` list? *(Eight: the three plus the default VLAN 1 and VLANs 1002 to 1005.)*
3. Two PCs share a switch, in different VLANs but the same subnet. Can they talk directly? *(No. The switch won't forward between VLANs; you need a router.)*

Do next

Do the Packet Tracer lab: make three VLANs, assign ports, name them, and test with a broadcast ping. Add the Day 16 Anki cards.

Day 17: VLANs, part 2 (trunks, 802.1Q, router on a stick)

Exam domain: 2. Network Access · **Video:** VLANs (Part 2) (40 min) · Lab: VLANs (Part 2)

Key ideas

- **Why trunks.** A VLAN can span several switches. Giving each VLAN its own switch-to-switch and switch-to-router cable wastes ports and runs out quickly. A **trunk port** carries traffic of **multiple VLANs over one link**. An access port carries one.
- **Tagging.** On a trunk, the switch tags each frame so the receiver knows its VLAN. Trunk = *tagged* port; access = *untagged* port.
- **Trunking protocols: IEEE 802.1Q** (the standard, "dot1q") and **ISL** (old Cisco-proprietary, effectively gone). Know what ISL is; study 802.1Q.
- **The 802.1Q tag** is **4 bytes (32 bits)**, inserted between the **source MAC** and **type/length** fields of the Ethernet header:

Field	Size	Purpose
TPID (tag protocol identifier)	16 bits	Always `0x8100`; marks the frame as 802.1Q tagged
PCP (priority code point)	3 bits	Class of service (priority)
DEI (drop eligible indicator)	1 bit	Frame may be dropped under congestion
VID (VLAN ID)	12 bits	The VLAN number

PCP, DEI and VID together form the TCI (tag control information).

- **VLAN ranges.** 12 bits gives 4096 values, but 0 and 4095 are reserved, so usable VLAN IDs are **1 to 4094**. **Normal range: 1 to 1005. Extended range: 1006 to 4094.** Most modern switches support both.
- **Native VLAN** (an 802.1Q feature, not in ISL): frames in the native VLAN are sent **untagged** on the trunk. A switch that receives an untagged frame on a trunk puts it in the native VLAN. The default is **VLAN 1**, set **per trunk port** (not globally). The two ends must match; with a mismatch traffic can end up in the wrong VLAN (for example VLAN 10 traffic arrives and is treated as VLAN 30) and be dropped. Best practice is to set the native VLAN to an **unused** VLAN.
- **Allowed VLANs.** By default **all** VLANs (1 to 4094) are allowed on a trunk. Restricting to the VLANs actually needed helps security and avoids pointless traffic.
- **Router on a stick (ROAS).** One physical router link to the switch, split into **subinterfaces**, one per VLAN, to do inter-VLAN routing. The switch port is a plain trunk. The router subinterface decides the VLAN by tag: frames arriving tagged for VLAN 20 are treated as arriving on the VLAN 20 subinterface, and frames leaving it are tagged with VLAN 20. The physical interface needs `no shutdown` and no IP address; each subinterface gets the VLAN's gateway address. Use the VLAN number as the subinterface number so it's obvious.
- Inter-VLAN traffic with ROAS: it goes up the trunk to the router and comes back down the same link, tagged for the destination VLAN.

Commands

```
interface g0/0
 switchport trunk encapsulation dot1q     ! Only needed on switches that support both
ISL and 802.1Q
 switchport mode trunk                    ! Make the port a trunk
 switchport trunk allowed vlan 10,30      ! Replace the allowed list
 switchport trunk allowed vlan add 20     ! Add to the list
 switchport trunk allowed vlan remove 20  ! Remove from the list
 switchport trunk allowed vlan all        ! Back to default (all VLANs)
 switchport trunk allowed vlan except 1-5,10  ! All VLANs except these
 switchport trunk allowed vlan none       ! No VLANs
 switchport trunk native vlan 1001        ! Native VLAN (match on both ends)
show interfaces trunk                     ! Mode, encapsulation, native VLAN, allowed
VLANs
!
interface g0/0                            ! Router: physical port
 no shutdown                              ! Routers' ports are down by default
interface g0/0.10                         ! Router: subinterface for VLAN 10
 encapsulation dot1q 10                   ! Tag/untag with VLAN 10
 ip address 192.168.1.62 255.255.255.192  ! Gateway address of that VLAN
```

Remember for the exam

- If `switchport mode trunk` is rejected, set `switchport trunk encapsulation dot1q` first (switches that support ISL default to auto).
- `show vlan brief` lists **access** ports only; use `show interfaces trunk` to check trunks.
- A VLAN that is allowed on a trunk but doesn't exist on the switch won't show under "VLANs allowed and active in management domain".
- To send a VLAN untagged across a trunk, make it the native VLAN. To undo allowed-list changes, use `allowed vlan all`.
- The VID field is 12 bits; TPID is 0x8100; the tag is 4 bytes.
- On the router, `encapsulation dot1q <vlan>` goes inside the subinterface.

Check yourself

1. How many bytes does an 802.1Q tag add, and where does it sit? *(4 bytes, between the source MAC and the type/length field.)*
2. Switch A's trunk has native VLAN 10 and switch B's has native VLAN 30. What goes wrong? *(Untagged frames are placed in a different VLAN at each end, so traffic leaks into or is lost in the wrong VLAN.)*
3. Why does a router-on-a-stick physical interface have no IP address? *(The subinterfaces carry the addresses, one per VLAN/subnet.)*

Do next

Do the Packet Tracer lab for trunks and router on a stick. Add the Day 17 Anki cards.

Day 18: VLANs, part 3 (native VLAN on a router, Layer 3 switching)

Exam domain: 2. Network Access · **Video:** VLANs (Part 3) (33 min) · Lab: VLANs (Part 3)

Key ideas

Native VLAN on a router (router on a stick). Untagged frames are smaller, so using the native VLAN saves a little overhead, but best practice is still an unused native VLAN. If you do use one, the router has two ways to do it:

1. Add `native` to the subinterface's encapsulation command: `encapsulation dot1q 10 native`.
2. Skip the subinterface and put the native VLAN's IP address on the **physical** interface. No `encapsulation` command is needed.

Both ends of the trunk must agree on the native VLAN.

What a capture shows. On a trunk, a frame in a non-native VLAN has an 802.1Q tag: type `0x8100` (the TPID), then PCP, DEI, VLAN ID, then the normal type field (for example IPv4). A frame in the native VLAN has no tag, and looks like a normal Ethernet frame.

Inter-VLAN routing: the three methods.

Method	How	Trade-off
Router with one interface per VLAN	Each VLAN has its own switch link and router port	Runs out of ports
Router on a stick	One trunk, router subinterfaces	One link, but all routed traffic goes to the router and back, which can congest
Layer 3 (multilayer) switch	The switch routes between its own VLANs with SVIs	Preferred in larger networks; traffic doesn't leave the switch

Layer 3 / multilayer switch. A switch that can switch **and** route. It can have routed ports, SVIs, a routing table and static routes. A Layer 2 switch is not IP-aware.

SVI (switch virtual interface). A virtual interface for a VLAN, configured as `interface vlan <id>`, with an IP address. It becomes the **default gateway of the hosts in that VLAN** (replacing the router subinterface). With SVIs, a frame between two VLANs is routed inside the switch.

Routed port. A physical switch port converted into a router-style interface with `no switchport`, so it can take an IP address. In `show interfaces status` its VLAN column reads `routed`.

Typical design. The multilayer switch routes between VLANs; a point-to-point routed link (for example a /30) goes up to the router; the switch has a default route pointing to that router for everything outside the LAN.

An SVI is up/up only if all of these are true:

- The VLAN exists on the switch. Creating an SVI does **not** create the VLAN (unlike assigning an access port to a missing VLAN).
- The switch has at least one up/up access port in that VLAN, or an up/up trunk that allows it.
- The VLAN is not administratively shut down (`shutdown` in VLAN config mode, on a real switch).
- The SVI itself is enabled. SVIs are shut down by default, so use `no shutdown`.

Commands

```
interface g0/0.10
 encapsulation dot1q 10 native          ! Router: this subinterface carries the native
VLAN
!
ip routing                              ! Switch: turn on Layer 3 routing (required)
interface g0/1
 no switchport                          ! Make the port a routed port
 ip address 192.168.1.193 255.255.255.252
ip route 0.0.0.0 0.0.0.0 192.168.1.194  ! Default route toward the router
interface vlan 10                       ! Create the SVI for VLAN 10
 ip address 192.168.1.62 255.255.255.192
 no shutdown                            ! SVIs start shut down
default interface g0/1                  ! Reset an interface to defaults
no interface g0/0.10                    ! Delete a subinterface
show ip route                           ! Routing table on the switch
show interfaces status                  ! Shows 'routed' for routed ports
show ip interface brief                 ! SVI and routed-port addresses and status
```

Remember for the exam

- Forget `ip routing` and inter-VLAN routing on the switch won't work.
- Native VLAN on a router: either `encapsulation dot1q <id> native` on the subinterface, or the address on the physical interface.
- `no switchport` makes a routed port; `interface vlan <id>` makes an SVI. They're not the same thing.
- A "down/down" SVI usually means the VLAN doesn't exist or has no active port in it.
- Hosts point their gateway at the SVI when a multilayer switch does the routing.
- Layer 2 switches get no IP routing; only a Layer 3 switch has a routing table.

Check yourself

1. You create `interface vlan 50` on a multilayer switch, add an address and `no shutdown`, but it stays down/down. Name two likely causes. *(VLAN 50 doesn't exist on the switch, or no up port carries VLAN 50.)*
2. How do you turn a switch port into a port with an IP address? *(`no switchport`, then `ip address`, with `ip routing` enabled globally.)*
3. Why is a multilayer switch better than router on a stick for a busy network? *(Inter-VLAN traffic is routed in the switch instead of crossing a single trunk to the router and back.)*

Do next

Do the Layer 3 switch lab: SVIs, a routed uplink and a default route. Add the Day 18 Anki cards. Day 19 covers DTP and VTP.

Day 19: DTP and VTP

Exam domain: 2. Network Access · **Video:** DTP/VTP (38 min) · Lab: DTP/VTP

Key ideas

Both protocols are **Cisco-proprietary**. They are no longer in the CCNA 200-301 topic list, but the lesson says questions can still appear, so know the basics.

DTP (Dynamic Trunking Protocol)

- Lets two Cisco switches negotiate whether a link becomes a **trunk or an access port** without manual configuration. It is on by default on switch ports.
- For security, **configure ports manually and turn DTP off**; DTP can be abused by an attacker.
- Modes set with `switchport mode`:
  - **access**: always access, DTP off.
  - **trunk**: always trunk (still sends DTP frames unless told not to).
  - **dynamic desirable**: actively tries to form a trunk.
  - **dynamic auto**: passive; becomes a trunk only if the other side asks.
- Result for each pair of modes (operational mode):

	access	**dynamic auto**	**dynamic desirable**	**trunk**
access	access	access	access	misconfiguration
dynamic auto	access	access	**trunk**	**trunk**
dynamic desirable	access	**trunk**	**trunk**	**trunk**
trunk	misconfiguration	**trunk**	**trunk**	**trunk**

- An access-to-trunk mismatch breaks traffic; avoid it.
- **Defaults:** older switches default to dynamic desirable, newer ones to **dynamic auto**. So two new switches default to access; swapping in an old switch can create a trunk unexpectedly.
- DTP never forms a trunk to a **router or PC**. A router-on-a-stick link must be set to `switchport mode trunk` by hand.
- `switchport nonegotiate` stops DTP frames on that port. `switchport mode access` also turns DTP off. A manually set trunk still sends DTP frames unless you add `nonegotiate`.
- On switches that support both ISL and 802.1Q, the default is `switchport trunk encapsulation negotiate`; ISL is preferred if both ends support it, and you must set the encapsulation manually before `switchport mode trunk`. DTP frames go in the native VLAN (VLAN 1 by default) for dot1q.
- In `show interfaces <int> switchport`, **administrative mode** is what you configured and **operational mode** is what the port actually is (trunk, or static access). "Negotiation of Trunking: On" means DTP is active.

VTP (VLAN Trunking Protocol)

- Lets you create VLANs on one **VTP server** switch and have other switches in the same **VTP domain** copy the VLAN database. It only syncs the **VLAN database**; you still assign ports to VLANs on each switch. It is rarely used today, and avoiding it is generally recommended.
- Advertisements go over **trunk ports** only.
- There are three versions (1, 2, 3). Version 1 is the default. Versions 1 and 2 support only VLANs up to 1005; **extended VLANs (1006 to 4094) need version 3**. Version 2 over version 1 only adds Token Ring VLAN support.
- Domain name is empty (null) by default. A switch with no domain **joins the domain of the first advertisement it receives**. Domain names must match for sync.

Mode	Add/edit/delete VLANs	Syncs to others	Advertises and forwards
Server (default)	Yes	Yes (acts as a client too)	Yes
Client	No (command rejected)	Yes	Yes (forwards)
Transparent	Yes, locally only	No	Forwards advertisements of its own domain; doesn't advertise its own VLANs

- **Revision number:** a server's number increases with every VLAN add, change or delete. Any switch in the domain adopts the VLAN database with the **highest revision number**, whoever sends it.
- **The danger:** a switch from another network with a higher revision number and the same domain name can overwrite every switch's VLANs and cut off hosts. Reset the number to 0 before connecting such a switch by **changing the domain to an unused name** or **changing the mode to transparent**.

Commands

```
interface g0/0
 switchport mode dynamic desirable   ! Actively try to trunk
 switchport mode dynamic auto        ! Trunk only if the other side asks
 switchport nonegotiate              ! Stop sending DTP frames
show interfaces g0/0 switchport      ! Administrative and operational mode, negotiation
status
!
vtp domain CISCO                     ! Set the VTP domain name
vtp mode client                      ! Also: server, transparent
vtp version 2                        ! Change VTP version (raises the revision number)
show vtp status                      ! Version, domain, mode, revision number, VLAN
count
```

Remember for the exam

- Two dynamic auto ports form an access link; auto with desirable forms a trunk.
- Server is the default VTP mode; servers also behave as clients. Clients can't change VLANs.

- Highest revision number wins. Changing the domain name or switching to transparent resets it to 0.
- To forward VTP updates without syncing, use `vtp mode transparent`.
- VTP does not put ports in VLANs; `switchport access vlan` is still needed per switch.

Check yourself

1. One port is dynamic auto and the other is dynamic desirable. What results? *(A trunk.)*
2. You want a switch to relay VTP updates but keep its own independent VLANs. Which mode? *(Transparent.)*
3. How do you stop a port from sending DTP frames? *(`switchport nonegotiate`, or set it to `switchport mode access`.)*

Do next

Do the DTP/VTP lab, trying each mode pair and reading `show interfaces switchport`. Add the Day 19 Anki cards.

Day 20: Spanning Tree Protocol, part 1

Exam domain: 2. Network Access · **Video:** Spanning Tree Protocol (Part 1) (39 min) · Lab: Analyzing STP

Key ideas

- **Redundancy** (extra links and devices) keeps a network up when something fails. In a switched LAN it creates loops, and a Layer 2 loop is dangerous.
- **Broadcast storm.** A switch floods broadcasts and unknown unicasts out every port except the one they arrived on. With a loop, the copies circle forever, because an Ethernet header has no TTL like the IP header does. They pile up until real traffic cannot get through.
- **MAC address flapping.** The same source MAC keeps arriving on different ports, so the switch keeps rewriting its MAC address table.
- **STP (IEEE 802.1D)** prevents loops by putting redundant ports in a **blocking** state. A blocking port still sends and receives STP messages (BPDUs) but no normal traffic. If an active link fails, a blocked port can start forwarding. Switches from every vendor run STP by default.
- STP says "bridge" where it means "switch".
- **BPDUs.** Switches send Hello BPDUs out every port every 2 seconds. A port that receives one knows a switch is on the other end; PCs and routers do not send them. A port with no BPDUs coming in cannot form a loop, so it forwards.
- Cisco switches run **PVST** (per-VLAN spanning tree): a separate STP instance in each VLAN, so a port can forward in one VLAN and block in another.

Bridge ID and the root bridge

The bridge ID is **bridge priority (16 bits) + MAC address (48 bits)**. The lowest bridge ID wins. The default priority is 32768, so by default the lowest MAC address wins.

The 16-bit priority field is split into a 4-bit priority and a 12-bit **extended system ID**, which holds the VLAN number. So the real value is the configured priority plus the VLAN ID: VLAN 1 defaults to 32769, VLAN 2 to 32770. You can only change the priority in steps of **4096** (0, 4096, 8192, and so on).

A switch assumes it is the root when it boots and gives up only when it hears a superior BPDU (lower bridge ID). Once the network has settled, only the root originates BPDUs.

How STP picks roles

1. **Root bridge:** the lowest bridge ID. Every port on it is a **designated** port (forwarding).
2. **Root port**, one on every non-root switch (forwarding). Tie-breakers in order:
    1. Lowest root cost.
    2. Lowest neighbor bridge ID.
    3. Lowest neighbor port ID (the neighbor's, not your own).
3. **Designated port**, one per link (collision domain), forwarding. Tie-breakers: lowest root cost for the switch, then lowest bridge ID. A port facing another switch's root port is always designated.
4. Every other port is **non-designated** and blocks.

Root cost is the sum of the costs of the outgoing interfaces along the path to the root. The root advertises 0, and each switch adds the cost of the port it sends out of.

Speed	STP cost
10 Mbps	100
100 Mbps (FastEthernet)	19
1 Gbps	4
10 Gbps	2

Port ID = port priority (default 128) + port number, so lower-numbered ports win ties.

Commands

```
SW1# show spanning-tree          ! root ID, local bridge ID, then each port's role,
state, cost and Prio.Nbr
```

Remember for the exam

- Lowest bridge ID is root; lowest root cost picks the root port.
- Costs: 100, 19, 4, 2 for 10M, 100M, 1G, 10G.
- Priority moves in steps of 4096 and the VLAN ID is added to it.
- Root bridge ports are all designated. Each link has exactly one designated port.
- Hello is 2 seconds.

Check yourself

1. SW1 has priority 32769 and SW2 has 28673 in VLAN 1. Which is root? *(SW2: the lower value wins whatever the MAC addresses are.)*
2. A switch reaches the root over two GigabitEthernet links in series. What is its root cost? *(4 + 4 = 8.)*
3. Why does a looping broadcast never die out on Ethernet? *(There is no TTL in the Ethernet header, so only STP blocking a port ends the loop.)*

Do next

Do the lab, then download the Day 20 Anki cards. Learn the cost table and the root port tie-breakers before Day 21.

Day 21: Spanning Tree Protocol, part 2 and the STP toolkit

Exam domain: 2. Network Access · **Video:** Spanning Tree Protocol (Part 2) (42 min) · PortFast (18 min) · BPDU Guard and BPDU Filter (24 min) · Root Guard (20 min) · Loop Guard (19 min) · Lab: Configuring STP (PVST+)

Key ideas

Port states and timers

State	Sends/receives BPDUs	Normal traffic	Learns MACs	Length
Blocking	receives only	no	no	until topology changes
Listening	yes	no	no	15 s (forward delay)
Learning	yes	no	yes	15 s (forward delay)
Forwarding	yes	yes	yes	stable

- Blocking and forwarding are the stable states. Root and designated ports settle in forwarding, non-designated ports in blocking. Listening and learning are transitional, and only root or designated ports pass through them. A shutdown port is disabled.
- **Hello** 2 s: how often the root sends BPDUs. Other switches relay them out their designated ports only.
- **Forward delay** 15 s: the length of each of listening and learning, so 30 s in total.
- **Max age** 20 s: how long a port waits without hearing a BPDU before the switch recalculates. A blocking port that must start forwarding therefore needs up to 20 + 15 + 15 = **50 s**. A forwarding port can go straight to blocking.
- The root bridge's timers are the ones every switch uses.
- BPDU destination MAC: PVST+ uses **0100.0ccc.cccd**; standard 802.1D uses **0180.c200.0000**. PVST+ supports dot1q trunks (original PVST was ISL only). The port ID in a BPDU shows priority and number, for example 0x8002 means priority 0x80 = 128, port 2.

Priorities and load balancing. Priorities are per VLAN, so make a different switch root for different VLANs and each VLAN blocks a different link. That spreads traffic across links. `root primary` sets the priority to 24576, or 4096 below the current lowest if that is lower. `root secondary` sets 28672.

The toolkit

- **PortFast:** the port skips listening and learning and forwards immediately. Use it only on ports to end hosts (or a router). On a switch link it can cause a temporary loop. The default command affects access ports only. `spanning-tree portfast trunk` allows it on a trunk, for a router-on-a-stick or a virtualization server. Newer IOS adds the keyword `edge` automatically.
- A PortFast port still sends BPDUs, and if it receives one it behaves as a normal STP port.
- **BPDU Guard:** a port that receives any BPDU goes to **err-disabled**. The default command affects PortFast-enabled ports, not all access ports. Re-enable with `shutdown` then `no shutdown` after fixing the cause, or let err-disable recovery do it (off by default, 300 s interval).
- **BPDU Filter:** stops a port sending BPDUs. Enabled on one interface, the port also ignores incoming BPDUs, which effectively turns STP off there and can cause permanent loops. Enabled globally, it

applies to PortFast ports and, if a BPDU arrives, PortFast and the filter switch off and the port acts normally. Prefer the global form. With both guard and filter on a port, a per-interface filter ignores the BPDU so the guard never fires; a global filter lets the guard fire.

- **Root Guard:** a port that receives a superior BPDU goes to **broken (root inconsistent)** and blocks. It recovers on its own once the superior BPDUs stop (after max age). Interface only, no global command. Used toward networks you do not control, such as a customer.
- **Loop Guard:** a root or non-designated port that stops receiving BPDUs (for example a unidirectional fiber link) goes to **broken (loop inconsistent)** instead of becoming designated. It recovers automatically when BPDUs return. Root Guard and Loop Guard cannot both be active on a port; the more recent or specific setting wins.

Commands

```
SW1(config)# spanning-tree mode pvst                        ! classic per-VLAN STP
(modern switches default to rapid-pvst)
SW1(config)# spanning-tree vlan 10 root primary             ! become root for VLAN 10
SW1(config)# spanning-tree vlan 20 root secondary           ! backup root for VLAN 20
SW1(config)# spanning-tree vlan 10 priority 4096            ! set the priority directly
(steps of 4096)
SW1(config-if)# spanning-tree vlan 10 cost 10               ! change the port cost
SW1(config-if)# spanning-tree vlan 10 port-priority 64      ! change the port priority
(default 128)
SW1(config-if)# spanning-tree portfast                      ! PortFast on this interface
SW1(config)# spanning-tree portfast default                 ! PortFast on all access
ports
SW1(config-if)# spanning-tree portfast disable              ! turn it off on one port
SW1(config-if)# spanning-tree bpduguard enable              ! BPDU Guard on this
interface
SW1(config)# spanning-tree portfast bpduguard default       ! BPDU Guard on all PortFast
ports
SW1(config-if)# spanning-tree bpdufilter enable             ! BPDU Filter on this
interface (risky)
SW1(config)# spanning-tree portfast bpdufilter default      ! BPDU Filter on all
PortFast ports
SW1(config-if)# spanning-tree guard root                    ! Root Guard
SW1(config-if)# spanning-tree guard loop                    ! Loop Guard on this
interface
SW1(config)# spanning-tree loopguard default                ! Loop Guard on all ports
SW1(config-if)# spanning-tree guard none                    ! turn off root/loop guard
here
SW1(config)# errdisable recovery cause bpduguard            ! auto re-enable BPDU-
guarded ports
SW1# show spanning-tree interface g0/1 detail               ! PortFast, guard and filter
```

```
status
SW1# show errdisable recovery                                  ! recovery causes and timers
```

Remember for the exam

- Timers: hello 2, forward delay 15, max age 20. Blocking to forwarding takes 30 s via listening and learning, 50 s after a max-age wait.
- Priorities are in steps of 4096; `root primary` is 24576, `root secondary` is 28672.
- PortFast is for hosts, BPDU Guard shuts the port on a BPDU, Root Guard blocks on a superior BPDU, Loop Guard blocks when BPDUs stop.
- `spanning-tree portfast default` affects access ports only.

Check yourself

1. Which two timers set the 30 seconds a new host port waits before forwarding? *(Two stages, listening and learning, of 15 s forward delay each.)*
2. A user plugs a small switch into a PortFast port with BPDU Guard on. What happens? *(The port goes err-disabled.)*
3. Which feature blocks a root or non-designated port that suddenly stops hearing BPDUs? *(Loop Guard.)*

Do next

Do the lab, practice root primary and secondary on two VLANs, and download the Day 21 Anki cards.

Day 22: Rapid Spanning Tree Protocol

Exam domain: 2. Network Access · **Video:** Rapid Spanning Tree Protocol (43 min) · Lab: Rapid STP

Key ideas

STP versions

Version	Standard	Instances	Notes
STP	802.1D	one for all VLANs	slow (up to 50 s), no load balancing
PVST+	Cisco	one per VLAN	Cisco's upgrade of 802.1D; load balances
RSTP	802.1w	one for all VLANs	fast, no load balancing
Rapid PVST+	Cisco	one per VLAN	Cisco's upgrade of 802.1w; the default on modern Cisco switches
MSTP	802.1s	VLANs grouped into instances	standard; Cisco runs the standard version

MSTP lets you map, say, VLANs 1 to 100 to one instance and 101 to 200 to another, so you only configure roots per instance. It is not tested in depth; Rapid PVST+ is the version in the exam topics.

What RSTP keeps and what it changes

- Same as STP: lowest bridge ID is root, and the same rules choose root ports and designated ports. It is an evolution of STP, not a new design.
- RSTP does not rely on long timers. Neighboring switches negotiate with a handshake so a port can go straight to forwarding when that is safe.
- **Port costs** were extended for faster links. The 802.1w values are 2,000,000 (10 Mbps), 200,000 (100 Mbps), 20,000 (1 Gbps), 2,000 (10 Gbps), 200 (100 Gbps) and 20 (1 Tbps). Cisco switches use the short values from Day 20 (100, 19, 4, 2) by default unless you configure the long path-cost method.
- **Three port states**, not five. Discarding replaces blocking and disabled, listening is dropped, and learning and forwarding stay. A shut down port is discarding.
- **Four port roles.** Root and designated are unchanged. The non-designated role is split:
  - **Alternate:** a discarding port that receives a superior BPDU from another switch. It is a ready-made backup for the root port and takes over immediately if the root port fails.
  - **Backup:** a discarding port that receives a superior BPDU from another port on the same switch. This only happens through a hub, so you will rarely see it. It backs up a designated port. The lowest port ID becomes the designated port.
- **Built-in features.** UplinkFast, BackboneFast and PortFast are part of RSTP and need no configuration. The first two speed up failover for blocked ports and are not tested in depth (know their names and purpose).
- **BPDUs.** The RSTP BPDU has protocol version **2** (classic is 0) and type 2, and uses all eight flag bits for the negotiation. Every switch originates BPDUs from its designated ports, not only the root.
- **Failure detection.** A neighbor is declared lost after **3 missed hellos (6 seconds)**, not 10 hellos (20 seconds). The switch then flushes the MAC addresses learned on that port.

- **Compatibility.** RSTP works with classic STP. The ports facing a classic switch fall back to classic behavior and timers.

Link types

Type	Connects to	Duplex	Notes
Edge	an end host	any	the same as PortFast; goes straight to forwarding
Point-to-point	another switch	full	detected automatically
Shared	a hub	half	detected automatically; essentially historical

Commands

```
SW1(config)# spanning-tree mode rapid-pvst                    ! Rapid PVST+ (the
default on modern switches)
SW1(config-if)# spanning-tree portfast                        ! makes the port an edge
port
SW1(config-if)# spanning-tree link-type point-to-point        ! force the link type if
needed
SW1(config-if)# spanning-tree link-type shared                ! link to a hub
SW1# show spanning-tree                                       ! mode shows "rstp";
roles and status per port
```

In `show spanning-tree` the status column may still say BLK for a discarding port.

Remember for the exam

- 802.1D = STP, 802.1w = RSTP, 802.1s = MSTP. The Cisco versions are PVST+ and Rapid PVST+.
- Port states: discarding, learning, forwarding. Port roles: root, designated, alternate, backup.
- Protocol version in the BPDU: 0 for STP, 2 for RSTP.
- An RSTP edge port is configured with `spanning-tree portfast`; there is no "edge" link-type command.
- In RSTP every switch sends BPDUs, and three missed BPDUs mean the neighbor is lost.

Check yourself

1. Which classic STP optional features are built into RSTP? *(PortFast, UplinkFast and BackboneFast.)*
2. Which two RSTP roles replace the non-designated role? *(Alternate and backup.)*
3. How long does RSTP wait before treating a neighbor as lost, with default hellos? *(Three missed BPDUs, 6 seconds.)*

Do next

Do the lab, then download the Day 22 Anki cards. Memorize the STP/RSTP comparison table.

Day 23: EtherChannel

Exam domain: 2. Network Access · **Video:** EtherChannel (42 min) · Lab: Configuring EtherChannel

Key ideas

- **The problem.** Add more links between two switches and STP blocks all but one, so you gain redundancy but no bandwidth.
- **EtherChannel** bundles several physical interfaces into one logical interface. STP treats the bundle as a single port, so every member forwards. Other names: **port channel** and **LAG** (link aggregation group). In diagrams it is a circle around the links.
- **No loop.** A broadcast flooded into the bundle arrives once, and the far switch will not flood it back out the same logical interface.
- **Load balancing is per flow.** Frames in one conversation always use the same member link, which avoids out-of-order delivery. The link is picked by a hash of source and/or destination MAC, IP or (on some models) Layer 4 ports. What is supported depends on the model. The default on the lecture's switch was source and destination IP, with non-IP traffic using MACs.
- **Up to 8** active member interfaces. LACP can have 16 with 8 active and 8 standby.

Ways to form one

Method	Standard	Modes
PAgP	Cisco proprietary	desirable, auto
LACP	IEEE 802.3ad, works with other vendors	active, passive
Static	none	on

Which combinations work

Side A	Side B	Result
desirable	desirable or auto	forms (PAgP)
auto	auto	does not form
active	active or passive	forms (LACP)
passive	passive	does not form
on	on	forms (static)
on	desirable, auto, active or passive	does not form
PAgP mode	LACP mode	does not form

Desirable and active start negotiation; auto and passive only reply. Negotiating protocols are preferred over static `on`, because they remove a failing link from the bundle. LACP is the one named in the exam topics, but know them all.

Rules for members. The configuration on every member must match: speed, duplex, switchport mode (access or trunk), and allowed and native VLANs on trunks. A mismatched interface is left out (it shows as suspended). Configure the whole range at once.

- The **channel-group number** is local to the switch. It only has to match between members on one switch, not with the neighbor. It creates the **Port-channel** interface, and you configure trunking, VLANs or an IP address on that interface; the members inherit it.
- `channel-protocol` is rarely needed. Once you set it, a conflicting mode is rejected (for example `mode on` after `channel-protocol lacp`).
- **Layer 3 EtherChannel.** Make the members routed ports first with `no switchport`, then add them to the group and put the IP address on the Port-channel interface. Routed links need no STP and flag as `R` instead of `S`.

Commands

```
SW1(config)# port-channel load-balance src-dst-mac     ! choose the hash inputs
SW1# show etherchannel load-balance                    ! current method
SW1(config)# interface range g0/0 - 3                  ! configure all members together
SW1(config-if-range)# channel-group 1 mode active      ! LACP; use passive, desirable,
auto or on
SW1(config-if-range)# channel-protocol lacp            ! optional, forces the protocol
SW1(config)# interface port-channel 1
SW1(config-if)# switchport mode trunk                  ! settings go on the logical
interface
SW1(config-if-range)# no switchport                    ! Layer 3 version: before
channel-group
SW1(config-if)# ip address 10.0.0.1 255.255.255.252    ! Layer 3: on the Port-channel
SW1# show etherchannel summary                         ! the main check
SW1# show etherchannel port-channel                    ! detail, including the group
mode
```

In `show etherchannel summary`, the port channel normally shows **SU** (Layer 2, in use) or **RU** (Layer 3, in use), and healthy members show **P** (bundled). **D** is down and lowercase **s** is suspended.

Remember for the exam

- Choose the right combos: on/on, desirable/desirable, desirable/auto, active/active, active/passive.
- PAgP is Cisco only; LACP is 802.3ad.
- Member settings must match. IP addresses go on the Port-channel, not the members.
- The group number is local. `port-channel load-balance` configures, `show etherchannel` verifies.
- Flags: S Layer 2, R Layer 3, U in use, P bundled.

Check yourself

1. Which pair forms an EtherChannel: passive/passive or active/passive? *(Active/passive; two passive sides wait for each other.)*
2. Why do three of four parallel links look orange without EtherChannel? *(STP blocks them to prevent loops.)*

3. A member shows lowercase s. What does it mean? *(It is suspended, usually because its settings do not match the others.)*

Do next

Do the lab, then download the Day 23 Anki cards. Memorize the mode combinations before Day 24.

Day 24: Dynamic routing

Exam domain: 3. IP Connectivity · **Video:** Dynamic Routing (45 min) · Lab: Floating Static Routes

Key ideas

- **Network route vs host route.** A network route points at a subnet (mask shorter than /32). A host route points at one address (/32); the router adds these itself as the "local" routes for its own interface addresses.
- **Dynamic routing** means routers run a protocol, form **adjacencies** (neighborships) with directly connected routers, and share what they know: their connected networks and routes learned from others. New LANs are advertised automatically, a dead route is withdrawn, and a backup route takes its place. A static route keeps pointing at a dead end because nothing tells the router the destination is gone. Static routes do not scale to large networks.

Categories of protocol

Category	Purpose	Algorithm	Protocols
IGP	routes within one autonomous system (one organization)	distance vector	RIP, EIGRP
IGP	same	link state	OSPF, IS-IS
EGP	routes between autonomous systems	path vector	BGP (the only EGP in use)

- **Distance vector** ("routing by rumor"): a router tells its neighbors the networks it knows and its metric to each. It knows only what neighbors say: the distance (metric) and the vector (next hop). Older design (early 1980s).
- **Link state:** each router advertises its links and connected networks until every router has the same map, then calculates best paths on its own. It needs more CPU and memory but reacts faster to changes.

Metric. The metric ranks routes learned by the **same** protocol; lower is better.

Protocol	Metric
RIP	hop count (all link speeds count the same)
EIGRP	bandwidth and delay by default (slowest bandwidth on the path, total delay)
OSPF	cost, based on interface bandwidth
IS-IS	cost, 10 per link by default, so effectively hop count

Routes to the same destination (same prefix and length) with an equal metric from the same protocol are both installed, and traffic is shared. This is **ECMP** (equal-cost multipath). Static routes can do it too, and they show metric 0.

Example: with RIP, a path through a FastEthernet link and a path through GigabitEthernet links both count as two hops and are used together. OSPF sees the slower link as costlier and uses only the faster path.

Administrative distance (AD). Different protocols' metrics cannot be compared, so AD ranks the protocol's trustworthiness first. Lower wins.

Route source	AD
Connected	0
Static	1
EBGP	20
EIGRP (internal)	90
IGRP (obsolete)	100
OSPF	110
IS-IS	115
RIP	120
EIGRP external	170
IBGP	200
Unusable	255

In `show ip route`, each entry shows `[AD/metric]`, for example `[110/3]`.

Order of route selection: the longest matching prefix wins first (a /28 beats a /24), then the lowest AD between protocols for the same prefix, then the lowest metric.

Floating static route. A static route given a higher AD than the dynamic protocol's. It stays out of the routing table until the dynamic route disappears, then takes over as a backup. The AD must be higher than the protocol's AD (higher than 110 for OSPF, for example), or the static route stays preferred.

Commands

```
R1(config)# ip route 192.168.4.0 255.255.255.0 10.0.12.2       ! normal static route, AD 1
R1(config)# ip route 192.168.4.0 255.255.255.0 10.0.13.2 200   ! floating static: AD 200 (the last number)
R1(config)# ip route 10.0.12.5 255.255.255.255 10.0.12.2       ! host route to one address
R1# show ip route                                              ! codes, [AD/metric], next hops
```

Remember for the exam

- IGP = RIP, EIGRP, OSPF, IS-IS. EGP = BGP. RIP and EIGRP are distance vector, OSPF and IS-IS link state, BGP path vector.
- AD order: connected 0, static 1, EBGP 20, EIGRP 90, OSPF 110, IS-IS 115, RIP 120.
- Metric compares routes within a protocol; AD compares protocols.
- Longest prefix match comes before AD and metric.

- The AD on a static route is the last argument of the command.

Check yourself

1. A router has an OSPF route and a RIP route to 10.1.1.0/24. Which is installed? *(OSPF, because AD 110 beats 120.)*
2. Which route is used for a packet to 10.20.0.14 when the table has a static /22 and an OSPF /28 for the same block? *(The /28, because the longest prefix match is used first.)*
3. What AD would make a static route a backup to EIGRP? *(Anything above 90, such as 100 or 200.)*

Do next

Do the lab, then download the Day 24 Anki cards. Memorize the AD table.

Day 25: RIP and EIGRP

Exam domain: 3. IP Connectivity · **Video:** RIP & EIGRP (44 min) · Lab: Configuring EIGRP

Key ideas

Neither protocol is named in the exam topics, but related questions can appear, and the configuration ideas carry over to OSPF.

RIP

- Industry standard, distance vector IGP. Metric is **hop count**, link speed is ignored. The maximum is **15 hops**; 16 means unreachable. AD **120**.
- Shares its whole routing table with neighbors every **30 seconds**. It uses request and response messages.
- **RIPv1:** classful only (no VLSM or CIDR), no mask in updates, broadcasts to 255.255.255.255. For example, 10.1.1.0/24 is advertised as 10.0.0.0, and 172.16.192.0/18 as 172.16.0.0.
- **RIPv2:** carries the mask (supports VLSM and CIDR), multicasts to **224.0.0.9**. RIPng is for IPv6.
- Used in small networks and labs only.

EIGRP

- Called an advanced (or hybrid) distance vector protocol. It was Cisco proprietary; Cisco has since published it, but in practice it is a Cisco-only protocol.
- Much faster to react than RIP and has no 15-hop limit. Multicasts to **224.0.0.10**.
- **Metric:** bandwidth and delay by default (K1 and K3 are 1; K2, K4 and K5 are 0). It uses the slowest bandwidth on the path plus the total delay. Values are large, such as 3072 or 28416.
- AD **90** internal, **170** external. Routes show as **D** in the table.
- The only IGP that can do **unequal-cost** load balancing, in proportion to bandwidth. It does ECMP over 4 paths by default, like RIP.
- The **AS number** must match between neighbors or they will not form an adjacency.

The `network` command (RIP, EIGRP and, later, OSPF). It does not say what to advertise. It says which interfaces to run the protocol on: any interface whose address falls inside the range. The router then advertises that interface's own prefix, such as 172.16.1.0/28, not the range you typed.

- RIP's `network` is classful: `network 10.0.0.0` matches every interface starting with 10.
- EIGRP accepts a **wildcard mask**: a 0 bit must match, a 1 bit may differ. It is the inverse of the subnet mask (255 minus each octet): /24 is 0.0.0.255, /28 is 0.0.0.15, /25 is 0.0.0.127, /19 is 0.0.31.255, /32 is 0.0.0.0. Without a mask it assumes classful.
- Example: interface 172.16.1.14 matches `network 172.16.1.0 0.0.0.15` and `network 172.16.1.8 0.0.0.7`, but not `network 172.16.1.0 0.0.0.7`.

Router ID (EIGRP, OSPF). A 32-bit number written like an IP address. Priority: manual setting, then the highest IP on a loopback interface, then the highest IP on any other interface.

Passive interface. Stops the router sending protocol messages out an interface that has no neighbors, while the interface's network is still advertised.

Commands

```
R1(config)# router rip
R1(config-router)# version 2                              ! always use v2
R1(config-router)# no auto-summary                        ! stop classful summarizing
R1(config-router)# network 10.0.0.0                       ! run RIP on interfaces in
10.0.0.0/8
R1(config-router)# passive-interface g2/0                 ! no updates out this port
R1(config-router)# default-information originate          ! advertise R1's default route
to neighbors
R1(config-router)# maximum-paths 8                        ! ECMP paths (default 4)
R1(config-router)# distance 85                            ! change the protocol's AD

R1(config)# router eigrp 1                                ! 1 is the AS number
R1(config-router)# no auto-summary
R1(config-router)# network 172.16.1.0 0.0.0.15            ! wildcard mask
R1(config-router)# eigrp router-id 1.1.1.1                ! set the router ID by hand
R1# show ip protocols                                     ! version, networks, passive
ports, neighbors, AD
```

`default-information originate`, `passive-interface`, `maximum-paths` and `distance` work the same way in OSPF.

Remember for the exam

- RIP: hop count, max 15, AD 120, updates every 30 s, v2 multicasts to 224.0.0.9.
- EIGRP: AD 90 (170 external), multicast 224.0.0.10, bandwidth and delay metric, unequal-cost load balancing.
- `network` chooses interfaces, not advertised prefixes. Wildcard mask = inverted subnet mask.
- Router ID: manual, then loopback, then highest physical IP.
- `default-information originate` is run on the router that has the default route.

Check yourself

1. What is the wildcard mask for a /27? *(0.0.0.31.)*
2. Why does RIP treat a 10 Mbps path and a 10 Gbps path as equal? *(Its metric is hop count only.)*
3. Router has no loopback; its interfaces are 10.1.1.1 and 192.168.5.1. What is its router ID? *(192.168.5.1, the highest address.)*

Do next

Do the lab, then download the Day 25 Anki cards. Practice converting masks to wildcard masks; OSPF uses them next.

Day 26: OSPF, part 1

Exam domain: 3. IP Connectivity · **Video:** OSPF Part 1 (40 min) · Lab: Configuring OSPF (1)

Key ideas

The exam topic is "configure and verify single-area OSPFv2", covering neighbor adjacencies, point-to-point, broadcast and router ID.

- **OSPF (Open Shortest Path First)** is a link-state IGP. It runs Dijkstra's shortest path first (SPF) algorithm. OSPFv1 (1989) is obsolete, **OSPFv2** (1998) is used for IPv4 and is the exam version, and OSPFv3 was built for IPv6 (it can also carry IPv4).
- **LSA and LSDB.** Routers describe their links in **LSAs** (link-state advertisements), which are stored in the **link-state database (LSDB)**. LSAs are **flooded** to neighbors until every router in the area has an identical LSDB. An LSA carries the originating router ID, the network and the cost, among other things. Each LSA is re-flooded every 30 minutes by default.
- **Three steps:** become neighbors with routers on the same segment, exchange LSAs, then each router runs SPF on its own copy of the LSDB and installs its best routes.
- **Areas.** Small networks work fine as a single area. In a big network one area means a large LSDB, slower SPF and heavy CPU and memory use, and every change floods and triggers a recalculation everywhere. Splitting into areas limits that. The exam covers only single-area configuration, but know the terms.

Term	Meaning
Area	routers and links sharing the same LSDB (each area has its own)
Backbone area	area 0; every other area must connect to it
Internal router	all interfaces in the same area
ABR (area border router)	interfaces in more than one area; keeps a separate LSDB per area; best connected to at most 2 areas
Backbone router	has an interface in area 0 (includes ABRs)
ASBR (autonomous system boundary router)	connects the OSPF domain to an outside network, for example the Internet
Intra-area route	destination in the router's own area
Interarea route	destination in a different area

Area rules

1. Areas must be contiguous (not split into separate pieces).
2. Every area needs an ABR connected to area 0.
3. Interfaces on the same subnet must be in the same area to become neighbors.

- **The `network` command** works as in EIGRP, with a wildcard mask, but it **requires the area**. It tells the router which interfaces to activate OSPF on in that area, and it does not say which networks to advertise. For single-area OSPF any area number works, but area 0 is the convention.

- **Process ID** is local to the router. Routers with different process IDs still become neighbors, and it has no relation to the area number. (Compare EIGRP, where the AS number must match.)
- **Passive interface** stops hellos out of an interface with no neighbors, while the interface's subnet is still advertised.
- `default-information originate` floods the router's default route, and that makes it an ASBR.
- **Router ID** order: manual `router-id`, then highest loopback IP, then highest IP on a physical interface. A new manual ID takes effect only after `clear ip ospf process` or a reload. Clearing the process drops OSPF routes briefly, so avoid it on a live network.
- **Load balancing and AD.** OSPF does ECMP on up to **4** paths by default (`maximum-paths`), but not unequal-cost balancing. AD is **110** (`distance`).

Commands

```
R1(config)# router ospf 1                                    ! 1 is the local process
ID
R1(config-router)# network 10.0.12.0 0.0.0.3 area 0          ! activate OSPF on
matching interfaces, in area 0
R1(config-router)# network 10.0.12.0 0.0.1.255 area 0        ! one command can cover
10.0.12.x and 10.0.13.x
R1(config-router)# passive-interface g2/0                    ! no hellos out this
port
R1(config-router)# default-information originate             ! advertise a default
route
R1(config-router)# router-id 1.1.1.1                         ! set the router ID
R1# clear ip ospf process                                    ! apply the new router
ID (asks to confirm)
R1(config-router)# maximum-paths 8                           ! ECMP paths (default 4)
R1(config-router)# distance 85                               ! change OSPF's AD
R1# show ip protocols                                        ! process ID, router ID,
networks, neighbors, AD
```

`show ip protocols` also says "autonomous system boundary router" on a router that originates a default route, and lists how many areas it is in.

Remember for the exam

- OSPF is link state, uses Dijkstra's SPF, AD 110, and builds the same LSDB in every router in an area.
- LSA = the advertisement; LSDB = the database.
- Know ABR, ASBR, backbone, internal, intra-area and interarea.
- Same-subnet interfaces must share an area. All areas connect to area 0.
- The `network` command needs a wildcard mask and an area.
- Process ID need not match between routers; the router ID command is `router-id` (not `ospf router-id`).

Check yourself

1. A router has interfaces in area 0 and area 2. What is it? *(An ABR, and also a backbone router.)*
2. Does OSPF process ID 1 on R1 prevent adjacency with process ID 5 on R2? *(No, the process ID is only locally significant.)*
3. Which command lets R1 give its default route to OSPF neighbors, and what does R1 become? *(`default-information originate`; an ASBR.)*

Do next

Do the lab, then download the Day 26 Anki cards. Learn the OSPF terms in the table.

Day 27: OSPF, part 2

Exam domain: 3. IP Connectivity · **Video:** OSPF Part 2 (37 min) · Lab: Configuring OSPF (2)

Key ideas

OSPF cost

- Cost = **reference bandwidth / interface bandwidth**, and anything below 1 is rounded up to 1. The default reference bandwidth is 100 Mbps.

Interface	Default cost
Ethernet (10 Mbps)	10
FastEthernet (100 Mbps)	1
GigabitEthernet	1
10 Gigabit	1
Loopback	1

- So FastEthernet and faster links all tie by default, which is a poor result. Raise the reference bandwidth above your fastest link and set it **the same on every router**. For example, 100,000 Mbps gives FastEthernet 1000 and Gigabit 100.
- A route's metric is the sum of the costs of the **outgoing** interfaces along the path (like STP root cost). Three gigabit hops at 100 each cost 300. A loopback destination adds 1 for the loopback itself.
- Three ways to change a cost: change the reference bandwidth, set `ip ospf cost` on the interface (this overrides the calculation), or change the interface `bandwidth` in kilobits per second (not recommended, because other features also use it). The `bandwidth` command does not change the real speed; `speed` does.
- Check the units with `?`. Reference bandwidth is in megabits per second; the `bandwidth` command is in kilobits per second.

Becoming neighbors

Neighbors are the key to OSPF: if they will not form, nothing works. When OSPF starts on an interface, the router sends **Hello** messages every **10 seconds** on Ethernet to multicast **224.0.0.5** (all OSPF routers). OSPF is carried directly in IP with **protocol number 89**. For comparison, RIPv2 uses 224.0.0.9 and EIGRP 224.0.0.10.

State	What happens
Down	no Hello received yet
Init	a Hello was received, but it does not list your router ID yet
2-way	Hellos now list each other's router ID, so the conditions to be neighbors are met. In some network types a DR and BDR are elected here
Exstart	master and slave are chosen by DBD packets. The higher router ID is master and starts the exchange

State	What happens
Exchange	routers swap DBDs, which list the LSAs each one holds (not the LSAs themselves)
Loading	missing LSAs are requested with LSRs, sent in LSUs, and acknowledged with LSAcks
Full	identical LSDBs; a full adjacency

- In Full, Hellos continue every 10 seconds. The **dead timer** (40 seconds) resets with each Hello. If it reaches 0, the neighbor is removed. On some other link types the defaults are 30 and 120.
- If routers do not reach 2-way, troubleshoot there. Down to 2-way is "becoming neighbors", Exstart to Loading is exchanging LSAs, and then SPF calculates the routes.

Type	Message
1	Hello
2	DBD (database description)
3	LSR (link-state request)
4	LSU (link-state update, carries the LSAs)
5	LSAck

More configuration

- OSPF can be enabled directly on an interface with `ip ospf <process> area <area>`, no `network` command needed. `show ip protocols` then lists them under "routing on interfaces configured explicitly".
- `passive-interface default` makes every interface passive; `no passive-interface <int>` re-enables the ones that need neighbors.

Commands

```
R1(config-router)# auto-cost reference-bandwidth 100000   ! in Mbps; match on all
routers
R1(config-if)# ip ospf cost 10000                         ! fix a cost on one interface
R1(config-if)# bandwidth 100000                           ! interface bandwidth in kbps
(not recommended)
R1(config-if)# ip ospf 1 area 0                           ! activate OSPF on this
interface directly
R1(config-router)# passive-interface default              ! all interfaces passive
R1(config-router)# no passive-interface g0/0              ! except this one
R1# show ip ospf interface g0/0                           ! cost, timers, neighbor
count, DR/BDR
R1# show ip ospf interface brief                          ! cost of each OSPF interface
at a glance
R1# show ip ospf neighbor                                 ! neighbor state, dead time,
address, interface
```

Remember for the exam

- Cost = reference bandwidth divided by bandwidth, minimum 1. Default reference is 100 Mbps. Total cost sums the outgoing interfaces.
- Hello 10, dead 40 on Ethernet. Multicast 224.0.0.5. IP protocol 89.
- State order: Down, Init, 2-way, Exstart, Exchange, Loading, Full.
- Master and slave are decided in Exstart (higher router ID is master).
- Message types 1 to 5: Hello, DBD, LSR, LSU, LSAck.

Check yourself

1. What `auto-cost reference-bandwidth` value gives a FastEthernet interface a cost of 100? *(10,000: 10,000 / 100 = 100.)*
2. In which state are the master and slave chosen? *(Exstart.)*
3. With the default settings, how long before a silent neighbor is dropped? *(40 seconds, the dead timer, with Hellos normally every 10 seconds.)*

Do next

Do the lab, then download the Day 27 Anki cards. Practice writing out the seven neighbor states in order from memory.

Day 28: OSPF, part 3

Exam domain: 3. IP Connectivity · **Video:** OSPF Part 3 (48 min) · Lab: Configuring OSPF (3)

Key ideas

The exam topic covers OSPFv2 neighbor adjacencies and the point-to-point and broadcast network types, plus the DR/BDR election.

- **Loopback interface.** A virtual interface that stays up/up unless you shut it down, so it gives a router an address that survives a failed physical link. That makes it a good router ID source.
- **Network types** describe the kind of link and change OSPF's behavior:

Type	Default on	Neighbor discovery	DR/BDR	Hello / dead
Broadcast	Ethernet, FDDI	Hellos to 224.0.0.5	Yes	10 s / 40 s
Point-to-point	PPP, HDLC serial	Hellos to 224.0.0.5	No	10 s / 40 s
Non-broadcast	Frame Relay, X.25	Neighbors typed in by hand	Yes	30 s / 120 s

- **DR and BDR (broadcast type).** Every subnet elects a designated router (DR) and a backup (BDR). Everyone else is a **DROther**. A lone router on a subnet is the DR with no BDR.
- **Election order:** highest OSPF interface priority wins (default **1**, range 0 to 255, **0 means never DR/BDR**), then highest router ID. First place is DR, second is BDR.
- **Non-preemptive.** A new router with a higher priority does not take over. Roles change only when OSPF resets, the interface goes down, or the DR fails. If the DR fails, the BDR becomes DR (even with lower priority) and a new BDR is elected.
- **Why a DR exists.** Routers form a FULL adjacency and swap LSAs only with the DR and BDR. Two DROthers stay in **2-way** with each other. This cuts LSA flooding on a busy segment. Routers send updates to the DR/BDR at **224.0.0.6**.
- **Serial basics.** The DCE end supplies the speed with `clock rate`. The default encapsulation on Cisco serial is HDLC, and PPP is an option; both ends must match. `show controllers` tells you which end is DCE. A serial link cannot use the broadcast type.

Neighbor requirements

Two routers will not become neighbors unless these match:

1. Area number
2. Subnet (same subnet on the connecting interfaces)
3. The OSPF process is not shut down
4. Router IDs are unique (a duplicate is reported and the neighbor stays down)
5. Hello and dead timers
6. Authentication settings

Two more problems are sneakier: the neighbors still come up, but OSPF misbehaves.

7. **IP MTU mismatch:** the neighbor sticks in **EXSTART** and never reaches FULL.

8. **Network type mismatch:** the neighbor shows FULL, yet routes from the other side are missing from the routing table.

LSA types to know

Type	Name	Created by	Describes
1	Router LSA	Every OSPF router	Its router ID and attached OSPF networks
2	Network LSA	The DR of a multi-access network	Routers on that segment
5	AS-external LSA	An ASBR	Routes outside the OSPF domain

Commands

```
R1(config-if)# ip ospf priority 255              ! 0 to 255; 0 = cannot be DR/BDR
R1(config-if)# ip ospf network point-to-point    ! change the network type
R1(config-if)# ip ospf hello-interval 5          ! timers must match the neighbor
R1(config-if)# ip ospf dead-interval 20
R1(config-if)# ip ospf authentication-key MyPass ! set the password
R1(config-if)# ip ospf authentication            ! turn authentication on
R1(config-if)# ip mtu 1400                       ! MTU must match the neighbor
R1(config-router)# shutdown                      ! disable the OSPF process
R1(config-router)# router-id 1.1.1.1             ! needs clear ip ospf process to apply
R1(config-if)# encapsulation ppp                 ! serial only; default is HDLC
R1(config-if)# clock rate 64000                  ! DCE end of a serial link only
R1# show ip ospf neighbor                        ! state and DR/BDR/DROTHER role ("-"
on point-to-point)
R1# show ip ospf interface brief                 ! Nbrs F/C = full adjacencies / total
neighbors
R1# show ip ospf interface g0/0                  ! priority, DR and BDR
R1# show ip ospf database                        ! the LSDB, listed by LSA type
```

Remember for the exam

- Broadcast type: DR/BDR elected, 224.0.0.5 and 224.0.0.6. Point-to-point type: no DR/BDR.
- Priority first, router ID second, and the election does not preempt.
- DROthers reach FULL only with the DR and BDR.
- Work through the eight neighbor requirements when an adjacency fails.
- LSA types 1, 2 and 5.

Check yourself

1. Two routers on an Ethernet segment both have priority 1. R1 has router ID 2.2.2.2 and R2 has 9.9.9.9. Who is DR? *(R2, the higher router ID.)*
2. A neighbor is stuck in EXSTART. What do you check first? *(IP MTU on both interfaces.)*
3. Which LSA type does the DR originate? *(Type 2, the network LSA.)*

Do next

Do the lab, then download the Day 28 Anki cards. Practice listing the neighbor requirements from memory.

Day 29: First hop redundancy protocols

Exam domain: 3. IP Connectivity · **Video:** First Hop Redundancy Protocols (40 min) · Lab: Configuring HSRP

Key ideas

The exam asks you to describe the purpose of FHRPs, not to configure them.

- **The problem.** Hosts have one default gateway address. If that router dies, a working backup router does not help, because the hosts keep sending to the dead address.
- **The fix.** Two or more routers share a **virtual IP (VIP)** and a **virtual MAC**. Hosts use the VIP as their default gateway. The routers trade multicast hellos and elect one **active** router (the default gateway) and one **standby**.
- **How it plays out.**
  1. A host ARPs for the VIP, and the active router answers with the virtual MAC.
  2. Frames to the gateway go to that virtual MAC.
  3. If the active router fails, the standby takes over and sends **gratuitous ARP** (a broadcast ARP reply nobody asked for) so the switches relearn which port reaches the virtual MAC. Hosts change nothing.
- **Preemption.** By default the old active router does not take its job back when it returns, much like the OSPF DR. Preemption can be turned on.
- **Load balancing.** HSRP and VRRP cannot split hosts of one subnet across routers. With one group per VLAN or subnet you can make a different router active in each, which spreads the load. GLBP balances within a single subnet.

The three protocols

	HSRP	VRRP	GLBP
Standard	Cisco proprietary	Open standard	Cisco proprietary
Roles	Active / standby	Master / backup	AVG (active virtual gateway) and up to 4 AVFs (active virtual forwarders)
Multicast (IPv4)	v1: 224.0.0.2, v2: 224.0.0.102	224.0.0.18	224.0.0.102
Virtual MAC	v1: 0000.0c07.acXX; v2: 0000.0c9f.fXXX	0000.5e00.01XX	0007.b400.XXYY
Balances within one subnet	No	No	Yes

- In every MAC, the end is the **group number in hexadecimal**. HSRP v1 has 2 hex digits (groups 0 to 255), v2 has 3 (groups 0 to 4095). GLBP uses XX for the group and YY for the AVF number.
- **HSRP v2** adds IPv6 support and many more groups. v1 and v2 do not interoperate, so both routers must use the same one. The default is v1.
- Group numbers must match between the routers.

- The AVG hands out virtual MACs, one per AVF, so different hosts get different gateway MACs.

Worked examples: HSRP v2 group 10 is 0000.0c9f.f00a. VRRP group 17 is 0000.5e00.0111. GLBP group 1, AVF 2 is 0007.b400.0102.

HSRP election

Highest priority wins (default **100**, range 0 to 255). On a tie, the highest IP address on the interface wins. Preemption is needed only on the router you want to retake the active role.

Commands

Not tested, but useful for the lab. HSRP is set per interface.

```
R1(config-if)# standby version 2               ! default is version 1
R1(config-if)# standby 1 ip 172.16.0.254       ! group 1 and the virtual IP
R1(config-if)# standby 1 priority 200          ! higher than the default 100
R1(config-if)# standby 1 preempt               ! retake the active role after
recovering
R1# show standby                               ! state, virtual IP and MAC, priority,
active and standby routers
```

Remember for the exam

- Purpose: a redundant default gateway for a subnet, using a virtual IP and MAC.
- HSRP is active/standby, VRRP is master/backup, GLBP has an AVG and AVFs.
- Only GLBP load balances within one subnet.
- Memorize the multicast addresses and virtual MAC formats in the table.
- The new active router sends gratuitous ARP so switches update their MAC tables.

Check yourself

1. Which two FHRPs are Cisco proprietary? *(HSRP and GLBP.)*
2. Which protocol uses the multicast address 224.0.0.18? *(VRRP.)*
3. After a failed active router recovers, why does it come back as standby? *(Preemption is off by default, so the current active router keeps the role.)*

Do next

Do the HSRP lab, then download the Day 29 Anki cards. Spend most of your time on the multicast and virtual MAC table.

Day 30: TCP and UDP

Exam domain: 1. Network Fundamentals · **Video:** TCP & UDP (34 min) · Lab: Wireshark Demo (TCP/UDP)

Key ideas

The exam asks you to compare TCP and UDP.

- **Layer 4 jobs.** The transport layer moves data between hosts without the hosts caring how the network works. It also provides **port numbers**, which identify the application protocol and let a host run many sessions at once (session multiplexing). The Layer 4 PDU is the **segment**.
- **Ports in a session.** The client picks a random **source port** and sends to a fixed **destination port** for the service. The reply swaps them, which is how the client matches the reply to the right session.

Range	Ports	Use
Well-known	0 to 1023	Major protocols
Registered	1024 to 49151	Registered applications
Ephemeral	49152 to 65535	Random client source ports

- Port fields are 16 bits, so there are 65,536 ports.

TCP

TCP is **connection-oriented** and gives reliable delivery, sequencing, error recovery and flow control.

- **Three-way handshake:** client sends **SYN**, server replies **SYN-ACK**, client sends **ACK**.
- **Closing:** FIN, ACK, then FIN, ACK (sometimes called the four-way handshake).
- **Sequence and acknowledgment numbers.** Each side starts at a random initial sequence number. The ACK number is the sequence number of the **next** segment expected (forward acknowledgment). A segment that is not acknowledged is sent again (retransmission).
- **Windowing.** The window size field lets a sender transmit several segments before waiting for one ACK. A sliding window grows until a segment is lost, then backs off, then grows again. This is flow control.
- Header fields to know: source port, destination port, sequence number, acknowledgment number, flags (SYN, ACK, FIN), window size.

UDP

UDP is **connectionless**, best effort and has no acknowledgment, sequencing, retransmission or flow control. Its header has just four fields: source port, destination port, length and checksum. The small header and lack of waiting make it suit delay-sensitive traffic such as voice and video calls.

	TCP	UDP
Connection	Established first	None
Reliability and recovery	Yes	No

	TCP	UDP
Sequencing	Yes	No
Flow control	Yes	No
Overhead	Higher	Lower
Port numbers	Yes	Yes
Typical use	File transfer, web, email	Voice, video, short lookups

Some applications use UDP and add their own reliability, for example TFTP. DNS uses UDP normally and TCP in some cases.

Ports to memorize

Protocol	Transport	Port
FTP	TCP	20, 21
SSH	TCP	22
Telnet	TCP	23
SMTP	TCP	25
DNS	UDP and TCP	53
DHCP	UDP	67, 68
TFTP	UDP	69
HTTP	TCP	80
POP3	TCP	110
SNMP	UDP	161, 162
HTTPS	TCP	443
Syslog	UDP	514

Remember for the exam

- Both TCP and UDP provide ports and session multiplexing. Only TCP provides reliability, sequencing, error recovery and flow control.
- SYN, SYN-ACK, ACK.
- ACK number = next expected sequence number.
- Client source ports come from the ephemeral range.
- The table of port numbers.

Check yourself

1. A server receives a segment with sequence number 40 and a window of one segment. What ACK number does it send? *(41, the next segment it expects.)*

2. Which transport protocol would you pick for a live video call, and why? *(UDP; no ACK or retransmission delays.)*
3. Is port 1500 well-known, registered or ephemeral? *(Registered, since it is in 1024 to 49151.)*

Do next

Watch the Wireshark demo, then download the Day 30 Anki cards. Review the port tags until you get all of them right.

Day 31: IPv6, part 1

Exam domain: 1. Network Fundamentals · **Video:** IPv6 Part 1 (39 min) · Lab: Configuring IPv6 (Part 1)

Key ideas

The exam topics are configuring and verifying IPv6 addressing and prefixes.

- **Hexadecimal refresher.** Base 16, digits 0 to 9 then A to F (A=10, B=11, C=12, D=13, E=14, F=15). One hex digit is exactly **4 bits**, so converting to binary is a digit-by-digit job. Example: hex A7 is 1010 0111.
- **Why IPv6.** IPv4 has about 4.3 billion addresses (32 bits) and the regional registries have run out. VLSM, private addresses and NAT only stretched the supply. IPv6 is the long-term fix.
- **Size.** An IPv6 address is **128 bits**, written as **32 hex digits in 8 groups of 4** (called quartets) separated by colons. Each extra bit doubles the count, so 128 bits is vastly more than four times 32 bits.
- **Prefix length** uses slash notation only, never a dotted mask.

Abbreviation rules

1. **Drop leading zeros** in each quartet (not trailing zeros).
2. Replace **one run** of consecutive all-zero quartets with `::`. You may use `::` only once per address, otherwise nobody could tell how many quartets each one stands for.

Example: `2001:0db8:0000:0000:0000:00a0:0000:0001` becomes `2001:db8::a0:0:1`.

To **expand** an address: put leading zeros back so each quartet has 4 digits, then replace `::` with enough all-zero quartets to reach 8.

Prefix and subnet structure

- An ISP usually gives an enterprise a **/48**. Subnets are normally **/64**, leaving 16 bits to number subnets and 64 bits for hosts.
- To find the prefix (network address) of a host address, keep the network bits and set the host bits to 0.
  - **Prefix is a multiple of 4:** the boundary falls between hex digits. For `2001:db8:1:2::5/60`, keep 15 digits and zero the rest: `2001:db8:1::/60`.
  - **Not a multiple of 4:** convert the digit that straddles the boundary to binary. For `2001:db8:abcd:ffff::5/50`, the first 48 bits are three quartets. The next 2 bits of `ffff` stay (11), the rest become 0, so 1100 0000 0000 0000 = `c000`. Prefix: `2001:db8:abcd:c000::/50`.
- Only leading zeros can be dropped, so the zeros in the host part do not vanish unless `::` covers them.
- `2001:db8::/32` is reserved for documentation and examples.

Commands

```
R1(config)# ipv6 unicast-routing                ! lets the router forward IPv6 (off
by default)
```

```
R1(config)# interface g0/0
R1(config-if)# ipv6 address 2001:db8:0:1::1/64    ! address and prefix length
R1(config-if)# no shutdown                        ! enable the interface
R1# show ipv6 interface brief                     ! addresses and status of all
interfaces
```

- Most IPv6 commands mirror IPv4 ones with `ip` replaced by `ipv6`.
- After you configure a global address, each interface also shows a second address, a **link-local** address, created automatically. These come next in Day 32.
- The router accepts the full, shortened or partly shortened form, and shows the shortened form back.

Remember for the exam

- 128 bits, 8 quartets, 4 bits per hex digit.
- Remove leading zeros; use `::` once, for one run of zero quartets.
- Typical split: /48 from the ISP, 16 bits of subnet ID, /64 subnets.
- `ipv6 unicast-routing` is needed for routing, and forgetting it is a common cause of "everything looks right but nothing works".
- Binary/hex/decimal conversions, 0 to 15.

Check yourself

1. How many bits does one hex digit represent? *(4.)*
2. Why is `2001::5::1` invalid? *(`::` appears twice.)*
3. What is the network prefix of `2001:db8:0:9:3c1f::1/64`? *(`2001:db8:0:9::/64`.)*

Do next

Do the lab, then download the Day 31 Anki cards. Practice shortening, expanding and finding prefixes on random addresses.

Day 32: IPv6, part 2

Exam domain: 1. Network Fundamentals · **Video:** IPv6 Part 2 (40 min) · Lab: Configuring IPv6 (Part 2)

Key ideas

The exam topics are modified EUI-64 and the IPv6 address types: global unicast, unique local, link-local, multicast, anycast.

Modified EUI-64

EUI-64 turns a 48-bit MAC address into a 64-bit interface ID, which becomes the host half of a /64 address.

1. Split the MAC in half.
2. Insert `FFFE` in the middle.
3. Invert the **7th bit** (the second-lowest bit of the first byte). In practice, first byte `00` becomes `02`, `52` becomes `50`.

Example: MAC `00:1B:44:11:3A:B7` becomes `021B:44FF:FE11:3AB7` (the first byte `00` turns into `02`). With prefix `2001:db8:0:1::/64` the full address is `2001:db8:0:1:21b:44ff:fe11:3ab7`.

Address types

Type	Range	Notes
Global unicast	Originally `2000::/3`; now anything not reserved	Public, internet-routable. Needs to be globally unique
Unique local	`FC00::/7`, in practice `FD..`	Private, like RFC 1918, not routed on the internet
Link-local	`FE80::/10`	Auto-generated, never routed past the link
Multicast	`FF00::/8`	One to many. There is **no broadcast** in IPv6
Anycast	A unicast address used on several routers	Delivered to the nearest one
Unspecified	`::`	"No address yet"; `::/0` is the default route
Loopback	`::1`	One address (IPv4 uses a whole 127.0.0.0/8)

- **Global unicast** structure: global routing prefix (48 bits, from the ISP), subnet ID (16 bits), interface ID (64 bits).
- **Unique local** structure: `FD`, a 40-bit global ID, 16-bit subnet ID, 64-bit interface ID. Make the global ID random so two merged companies are unlikely to clash.
- **Link-local** addresses appear on any IPv6-enabled interface. They are used for OSPFv3 neighbors, static route next hops and NDP, and a router will not forward a packet addressed to one. You can create one without any other address using `ipv6 enable`. In practice they start `fe80:0:0:0`.
- **Multicast addresses to know:**

Address	Meaning
`ff02::1`	All nodes (acts like broadcast)
`ff02::2`	All routers
`ff02::5`	All OSPF routers
`ff02::6`	OSPF DR and BDR
`ff02::9`	RIP routers
`ff02::a`	EIGRP routers

The last digits match their IPv4 equivalents (224.0.0.5 and `ff02::5`, 224.0.0.10 and `ff02::a`).

- **Multicast scope** is the 4th hex digit: `ff01` interface-local, `ff02` link-local, `ff05` site-local, `ff08` organization-local, `ff0e` global. Don't confuse a link-local multicast (`ff02`) with a link-local unicast (`fe80`).
- **Anycast:** several routers share one address and advertise it; senders reach whichever is nearest by routing metric.

Commands

```
R1(config-if)# ipv6 address 2001:db8:0:1::/64 eui-64   ! router builds the interface ID
from the MAC
R1(config-if)# ipv6 enable                               ! link-local address only
R1(config-if)# ipv6 address 2001:db8:0:1::99/128 anycast ! mark an address as anycast
R1# show ipv6 interface g0/0                             ! addresses (EUI, ANY flags)
and joined multicast groups
R1# show ipv6 interface brief
```

Remember for the exam

- EUI-64: split, insert FFFE, flip the 7th bit.
- Know the prefixes: 2000::/3, FC00::/7 (FD), FE80::/10, FF00::/8.
- Unique local is private; global unicast is public.
- No broadcast in IPv6; `ff02::1` reaches all nodes on the link.
- Interfaces join `ff02::1` and, on routers, `ff02::2` automatically.

Check yourself

1. MAC `5254.00AB.CD01`: what is the EUI-64 interface ID? *(`5054:00FF:FEAB:CD01`.)*
2. Which address does a router listen to for "all routers on this link"? *(`ff02::2`.)*
3. A packet's destination is `fe80::1`. Will a router forward it to another subnet? *(No, link-local is never routed.)*

Do next

Do the lab, then download the Day 32 Anki cards. Learn the address-type table and the multicast list.

Day 33: IPv6, part 3

Exam domain: 1. Network Fundamentals · **Video:** IPv6 Part 3 (44 min) · Lab: Configuring IPv6 (Part 3)

Key ideas

The exam focus is IPv6 static routing, with NDP and SLAAC as supporting knowledge.

- **Standard writing style (RFC 5952).** Remove leading zeros. Use `::` for the **longest** run of zero quartets, and for the leftmost one if tied. Don't use `::` for a single zero quartet. Write hex letters in lowercase. Cisco's own output is not always strict, so uppercase is not a big deal on the exam.
- **IPv6 header.** A fixed **40 bytes** (IPv4 varies from 20 to 60), so there is no header length field.

Field	Bits	Notes
Version	4	Always 6
Traffic class	8	QoS marking
Flow label	20	Identifies a flow
Payload length	16	Bytes after the header
Next header	8	Like IPv4 protocol (TCP, UDP)
Hop limit	8	Like IPv4 TTL, packet dropped at 0
Source and destination	128 each	

Neighbor discovery protocol (NDP)

NDP replaces ARP and runs on ICMPv6.

Message	ICMPv6 type	Sent to	Purpose
Neighbor solicitation (NS)	135	Target's solicited-node multicast	"What is your MAC?" (ARP request)
Neighbor advertisement (NA)	136	The asker	Answer with a MAC (ARP reply)
Router solicitation (RS)	133	`ff02::2`	"Any routers here?"
Router advertisement (RA)	134	`ff02::1`	Router announces itself and the link prefix

- **Solicited-node multicast** = `ff02::1:ff` + the **last 6 hex digits** of the unicast address. Example: `2001:db8::78:9abc` gives `ff02::1:ff78:9abc`. It is multicast, so only the intended host cares, unlike an ARP broadcast.
- The result is kept in the **IPv6 neighbor table** (the ARP table equivalent).
- **SLAAC** (stateless address autoconfiguration): a device learns the prefix from an RA, then builds its own interface ID (EUI-64 or random). On a Cisco router you use `ipv6 address autoconfig`.
- **DAD (duplicate address detection):** before using an address, a device sends an NS for its own address. A reply means someone else already has it.

IPv6 routing and static routes

- IPv6 and IPv4 routing tables are separate. IPv6 routing is off until `ipv6 unicast-routing`.
- Each configured address adds a connected `/64` route (C) and a local `/128` host route (L). Link-local addresses don't appear. There is also `ff00::/8` pointing to Null0.
- Three ways to write a static route:

Type	You give	Note
Directly attached	Exit interface only	Does **not** work on Ethernet interfaces in IPv6
Recursive	Next hop only	Router looks up the next hop in a second pass
Fully specified	Exit interface and next hop	Required if the next hop is a link-local address

- Route purposes: **network** route (usually /64), **host** route (/128), **default** route (`::/0`), and **floating static** (an AD higher than the main route, for example above 110 for OSPF or 90 for EIGRP).

Commands

```
R1(config)# ipv6 route 2001:db8:0:3::/64 2001:db8:0:12::2          ! network route,
recursive
R1(config)# ipv6 route 2001:db8:0:3::10/128 2001:db8:0:12::2       ! host route
R1(config)# ipv6 route ::/0 2001:db8:0:12::2                       ! default route
R1(config)# ipv6 route 2001:db8:0:3::/64 g0/0 fe80::2              ! fully specified,
link-local next hop
R1(config)# ipv6 route 2001:db8:0:3::/64 2001:db8:0:14::4 130      ! floating static,
AD 130
R1(config-if)# ipv6 address autoconfig                             ! SLAAC
R1# show ipv6 route                                                ! the IPv6 routing
table
R1# show ipv6 neighbor                                             ! NDP neighbor
table
```

Remember for the exam

- NDP: NS 135, NA 136, RS 133, RA 134. Solicited-node = `ff02::1:ff` plus last 24 bits.
- Static routes follow the same ideas as IPv4, with `ipv6 route` and `::/0` for the default.
- A link-local next hop needs the exit interface too.
- Floating static AD must exceed the main route's AD.
- A mysterious IPv6 failure often means a missing `ipv6 unicast-routing`.

Check yourself

1. Which NDP message replaces an ARP request? *(Neighbor solicitation, ICMPv6 type 135.)*
2. What is the solicited-node multicast address for `2001:db8::5a:7c0f`? *(`ff02::1:ff5a:7c0f`.)*

3. You want a backup route used only if OSPF's route disappears. What AD would you set? *(Anything above 110, for example 130.)*

Do next

Do the lab, then download the Day 33 Anki cards. Write the four NDP message types and numbers from memory.

Day 34: Standard ACLs

Exam domain: 5. Security Fundamentals · **Video:** Standard ACLs (47 min) · Lab: Standard ACLs

Key ideas

The exam topic is configuring and verifying ACLs (IPv4 is what you need). ACLs have several uses; this lesson treats them as **packet filters** that decide which traffic the router permits or drops, even when it has a route.

- An ACL is an ordered list of entries (**ACEs**) built in global config mode. It does nothing until you **apply it to an interface**, **inbound** or **outbound**.
- **Top-down, first match wins.** The router stops at the first matching entry and ignores the rest, so order matters. A broad `permit` above a narrow `deny` makes the `deny` useless.
- **Implicit deny.** Every ACL ends with an invisible "deny all". A packet that matches nothing is dropped, so a filter that only denies one host usually needs `permit any` at the end.
- **One ACL per interface per direction.** Applying a second one in the same direction replaces the first.
- **Direction is from the router's point of view.** Inbound is checked as the packet enters, outbound just before it leaves. An ACL applied outbound will not touch traffic arriving on that interface.

ACL types

	Matches on	Numbered ranges
Standard	Source IP only	1 to 99 and 1300 to 1999
Extended	Source and destination IP, protocol, ports (Day 35)	100 to 199 and 2000 to 2699

Both can be numbered or named.

Writing entries

- Match with an IP address and a **wildcard mask** (not a subnet mask). `192.168.1.0 0.0.0.255` matches the /24.
- Three equal ways to match one host: `192.168.1.1 0.0.0.0`, `192.168.1.1`, or `host 192.168.1.1`. A bare address means /32 and works only in standard ACLs.
- `any` equals `0.0.0.0 255.255.255.255`.
- Add a `remark` to describe the ACL. It shows in the running config but not in `show access-lists`.
- The router stores a /32 mask as `host` and shows `permit any` for the all-addresses form.
- **Sequence numbers.** Entries get 10, 20, 30 and so on. In named ACL mode you can choose the number yourself, for example 5 to put an entry first.

Placement

Standard ACLs only see the source address, so put them **as close to the destination as possible**. Placing one near the source blocks that source from everything beyond it, not only from the one destination you meant.

Commands

```
R1(config)# access-list 1 permit host 192.168.1.1        ! numbered entry, allow one PC
R1(config)# access-list 1 deny 192.168.1.0 0.0.0.255     ! then block the rest of the
/24
R1(config)# access-list 1 permit any                     ! allow everything else
R1(config)# access-list 1 remark Only PC1 reaches servers ! description
R1(config)# interface g0/2
R1(config-if)# ip access-group 1 out                     ! apply to the interface (in
or out)
R1(config)# ip access-list standard BLOCK_BOB            ! named ACL: enter its mode
R1(config-std-nacl)# 5 deny host 1.1.1.1                 ! optional sequence number
first
R1(config-std-nacl)# 10 permit any
R1# show access-lists                                    ! all ACLs with sequence
numbers
R1# show ip access-lists                                 ! IP ACLs only
R1# show running-config | section access-list            ! the ACL as configured
```

The command to apply is `ip access-group`, not `access-list`.

Remember for the exam

- Standard: source IP only, ranges 1 to 99 and 1300 to 1999.
- Top-down, first match, implicit deny.
- Wildcard masks, not subnet masks.
- One ACL per interface per direction.
- Place standard ACLs close to the destination.

Check yourself

1. An ACL has `deny 10.1.0.0 0.0.255.255` followed by `permit 10.1.1.1`. Can 10.1.1.1 get through? *(No. The deny matches first.)*
2. What number would you use for a standard numbered ACL: 99, 100, 150 or 1400? *(99 and 1400. 100 and 150 are extended.)*
3. What happens to a packet that matches no entry? *(It is dropped by the implicit deny.)*

Do next

Do the lab, then download the Day 34 Anki cards. Practice writing wildcard masks for /24, /16 and /30 networks.

Day 35: Extended ACLs

Exam domain: 5. Security Fundamentals · **Video:** Extended ACLs (41 min) · Lab: Extended ACLs

Key ideas

Everything from Day 34 still applies: top-down processing, first match, implicit deny, and one ACL per direction per interface. Extended ACLs can match much more.

- **Numbered ranges:** extended **100 to 199 and 2000 to 2699**; standard 1 to 99 and 1300 to 1999. Know both.
- **Placement is the opposite of standard ACLs.** Extended ACLs are specific, so place them **as close to the source as possible**, and unwanted packets are dropped before they use any network.

Two ways to configure a numbered ACL

1. In global config mode with `access-list <number> ...`. You cannot choose sequence numbers, and each new entry goes at the end. You also cannot remove a single entry: `no access-list 1 deny ...` deletes the **whole** ACL.
2. In named ACL mode, even for a number: `ip access-list standard 1` or `ip access-list extended 100`. The config still looks like the old style, but you can delete one entry with `no <sequence>` and insert an entry by giving a sequence number.

If entries are numbered too tightly (1, 2, 3...), there is no room to insert. Use **resequencing**:

`ip access-list resequence <name or number> <start> <increment>`, in global config mode. `ip access-list resequence 1 10 10` renumbers entries 10, 20, 30 and keeps their order. It works for numbered, named, standard and extended ACLs.

Writing extended entries

Syntax: `permit|deny <protocol> <source> [source port] <destination> [destination port]`

- **Protocol:** a name or its IP protocol number. Remember **1 ICMP, 6 TCP, 17 UDP, 88 EIGRP, 89 OSPF**. `ip` matches every IP packet (used for `permit ip any any`).
- **Addresses:** wildcard mask, `host`, or `any`. Unlike standard ACLs, a bare address is not accepted for a /32 here.
- **Port operators** (TCP or UDP only), placed after the address they apply to: `eq`, `gt`, `lt`, `neq`, `range`. `eq 80` can be written `eq www`, but many ports have no keyword, so know the numbers from Day 30.
- A packet must match **every** field in an entry for the entry to apply.
- Other optional matches exist (TCP flags, TTL, DSCP) but are not needed here.

Commands

```
R1(config)# ip access-list extended BLOCK_HTTPS           ! named extended ACL
R1(config-ext-nacl)# deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443  ! no HTTPS
to the server
R1(config-ext-nacl)# deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255     ! no pings
```

```
to that subnet
R1(config-ext-nacl)# permit ip any any                    ! allow everything else
R1(config-ext-nacl)# 15 deny udp any range 20000 30000 host 3.3.3.3  ! sequence number
inserts it
R1(config-ext-nacl)# no 15                                ! delete one entry
R1(config)# access-list 101 permit tcp 10.0.0.0 0.0.255.255 host 2.2.2.2 eq 443  !
numbered form
R1(config)# ip access-list resequence BLOCK_HTTPS 10 10   ! renumber the entries
R1(config)# interface g0/1
R1(config-if)# ip access-group BLOCK_HTTPS in             ! apply inbound, near the
source
R1# show access-lists                                     ! ACLs and entries
R1# show ip interface g0/1                                ! shows the inbound and
outbound ACL on the interface
```

`show ip interface` (without `brief`) prints "Inbound access list" and "Outbound access list" lines, showing "not set" if none.

Remember for the exam

- Extended: protocol, source, destination, ports. Ranges 100 to 199 and 2000 to 2699.
- Extended close to the source; standard close to the destination.
- Protocol numbers 1, 6, 17, 88, 89.
- `eq`, `gt`, `lt`, `neq`, `range`.
- Only named ACL mode can delete or insert single entries.
- HTTP and HTTPS run over TCP, so an entry that names UDP never matches them.

Check yourself

1. Write an entry that blocks Telnet from 172.16.5.0/24 to 10.0.0.1. *(`deny tcp 172.16.5.0 0.0.0.255 host 10.0.0.1 eq 23`.)*
2. Which command lets you keep ACL 120's order but give the entries room to insert new ones? *(`ip access-list resequence 120 10 10`.)*
3. Where do you apply an extended ACL, and why? *(Close to the source, so denied traffic is dropped early.)*

Do next

Do the lab, then download the Day 35 Anki cards. Write ten extended entries from plain-English requirements until it feels automatic.

Day 36: CDP and LLDP

Exam domain: 2. Network Access · **Video:** CDP & LLDP (39 min) · Lab: CDP & LLDP

Key ideas

The exam topic is configuring and verifying Layer 2 discovery protocols, CDP and LLDP.

- **What they do.** Directly connected devices share their host name, device type, interface, and sometimes IP address and software version. They run at Layer 2 with no IP packets, yet can carry Layer 3 details. A switch with no IP on an interface sends none.
- **Neighbors only.** A device processes a received message and discards it, so only directly connected devices become neighbors.
- **Security.** The shared information is useful to an attacker, so many networks disable these protocols.

	CDP	LLDP
Origin	Cisco proprietary	Industry standard, IEEE 802.1AB
Default	On, globally and per interface	Off, globally and per interface
Destination MAC	`0100.0ccc.cccc`	`0180.c200.000e`
Message timer	60 s	30 s
Hold time	180 s	120 s
Extra timer	None	Reinitialization delay, 2 s

- **Mixed vendors** need LLDP. A device can run CDP and LLDP at the same time.
- CDP sends version 2 messages by default.
- CDP's multicast MAC is shared with other Cisco protocols such as VTP and DTP.
- CDP v2 can flag native VLAN and duplex mismatches. CDP's detailed output shows VTP information; LLDP does not.
- If no message arrives within the hold time, the neighbor entry is removed.
- LLDP is turned on globally and also per interface, and transmit and receive are separate. CDP uses a single `cdp enable`.

Reading the neighbor table

`show cdp neighbors` columns: **Device ID** (the neighbor's host name), **Local Interface** (your port), **Holdtime** (counts down from 180 and resets on each message), **Capability**, **Platform** (model) and **Port ID** (the neighbor's port). Local interface and Port ID are easy to mix up on the exam.

Capability codes: **R** router, **S** switch (CDP). LLDP uses **B** for bridge instead of S. A multilayer switch can show both router and switch codes. LLDP's neighbor output shows the fixed configured hold time, and you need the `detail` form to see the time remaining. LLDP detail also lists "system capabilities" (what the device can do) and "enabled capabilities" (what it is doing).

`show cdp neighbors detail` and `show cdp entry <name>` add the neighbor's IP address, IOS version, native VLAN and duplex.

Commands

```
R1(config)# cdp run                          ! CDP globally on (default); no cdp run turns
it off
R1(config-if)# cdp enable                    ! per interface (default); no cdp enable
turns it off
R1(config)# cdp timer 30                     ! send interval in seconds
R1(config)# cdp holdtime 120                 ! hold time in seconds
R1(config)# cdp advertise-v2                 ! version 2 (default); no form uses v1
R1# show cdp                                 ! timer, hold time and version
R1# show cdp traffic                         ! messages sent and received
R1# show cdp interface                       ! per-interface settings
R1# show cdp neighbors                       ! neighbor table
R1# show cdp neighbors detail                ! plus IP address, IOS, VTP, native VLAN,
duplex
R1# show cdp entry SW1                       ! detail for one neighbor

R1(config)# lldp run                         ! LLDP globally on
R1(config-if)# lldp transmit                 ! send LLDP on this interface
R1(config-if)# lldp receive                  ! receive LLDP on this interface
R1(config)# lldp timer 15                    ! send interval
R1(config)# lldp holdtime 90                 ! hold time
R1(config)# lldp reinit 3                    ! reinitialization delay
R1# show lldp                                ! timers
R1# show lldp traffic
R1# show lldp interface                      ! TX and RX state per interface
R1# show lldp neighbors
R1# show lldp neighbors detail
R1# show lldp entry SW1
```

The `show` commands match: replace `cdp` with `lldp`.

Remember for the exam

- CDP: Cisco only, on by default, 60 s timer, 180 s hold, v2, multicast MAC `0100.0ccc.cccc`.
- LLDP: standard, off by default, 30 s timer, 120 s hold, 2 s reinit, multicast MAC `0180.c200.000e`.
- LLDP needs `lldp run`, then `lldp transmit` and `lldp receive` on interfaces.
- To see a neighbor's IP address, use the detail output.
- Local interface is yours; Port ID is the neighbor's.

Check yourself

1. Which protocol do you need in a network of Cisco and Juniper switches? *(LLDP.)*
2. Which two commands show the configured CDP timer and hold time? *(`show cdp` and `show cdp interface`.)*

3. A port should receive LLDP but not send it. What do you configure? *(*`lldp receive` *on that interface, without* `lldp transmit`*.)*

Do next

Do the lab, then download the Day 36 Anki cards. Memorize the timers and the two multicast MAC addresses.

Day 37: NTP

Exam domain: 4. IP Services · **Video:** NTP (43 min) · Lab: NTP

Key ideas

The exam topic is "configure and verify NTP operating in client and server mode" (4.2).

- **Why time matters.** Logs are only useful if timestamps agree across devices. If R2 thinks it is 1:17 AM and R3 thinks it is a day in 2008, you cannot line up their events when troubleshooting.
- **Two clocks.** The **software clock** (`show clock`) runs while the device is up; the **hardware clock**, the **calendar** (`show calendar`), survives reboots and seeds the software clock at boot. The default time zone is **UTC**. In `show clock detail`, a leading `*` means the time is not authoritative.
- **Manual setup does not scale** and the clocks drift, which is why NTP exists. NTP syncs a device to a server over the network. It reaches roughly 1 ms accuracy on a LAN and about 50 ms over a WAN or the Internet. It uses **UDP port 123** and only ever carries UTC, so each device needs its own time zone configured.
- **Stratum** is the distance from the original time source. Lower is better, and a router prefers the server with the lower stratum.

Stratum	What it is
0	Reference clock (atomic or GPS), not on the network
1	Primary server, attached directly to a reference clock
2 and up	Secondary servers; each syncs to the level above and is a client and a server at once
15	Highest usable value; 16 means unsynchronized and is not trusted

- **Three modes on Cisco devices**, usable at the same time: **server**, **client**, and **symmetric active** (peers of equal stratum that back each other up).
- A device that syncs to a server automatically serves time to others, with stratum one higher than its source.
- `ntp master` makes a device a server even with no upstream source. It uses itself as the reference (127.127.1.1, stratum 7), so the device is **stratum 8** by default.
- **Several `ntp server` lines** are fine: the router picks the best one and can switch. `prefer` favors one.
- **Source interface.** Pointing `ntp source` at a loopback keeps the server address reachable as long as any path to the router is up.
- **Authentication is optional.** Clients only sync to a server whose key matches.

Commands

```
R1# clock set 14:38:00 27 December 2020             ! set the software clock (exec
mode)
R1# clock update-calendar                           ! copy clock time into the
calendar
R1# clock read-calendar                             ! copy calendar time into the
```

```
clock
R1(config)# clock timezone JST 9                      ! name and hours offset from UTC
R1(config)# clock summer-time EDT recurring 2 Sunday March 2:00 1 Sunday November 2:00
! start, then end
R1(config)# ntp server 216.239.35.0                   ! sync to this server
R1(config)# ntp server 216.239.35.4 prefer            ! prefer this one
R1(config)# ntp update-calendar                       ! let NTP update the hardware
calendar
R1(config)# ntp source loopback0                      ! source NTP packets from the
loopback
R1(config)# ntp master                                ! be a server without upstream
(stratum 8)
R2(config)# ntp peer 10.0.23.3                        ! symmetric active mode
R1(config)# ntp authenticate                          ! turn on authentication
R1(config)# ntp authentication-key 1 md5 mysecret     ! define key 1
R1(config)# ntp trusted-key 1                         ! mark key 1 as trusted
R2(config)# ntp server 10.1.1.1 key 1                 ! clients only: use key 1 for this
server
R1# show ntp associations                             ! * synced, + candidate, ~
configured
R1# show ntp status                                   ! synchronized? own stratum,
reference
R1# show clock detail                                 ! time and its source
```

Remember for the exam

- NTP uses UDP 123 and UTC only; set the time zone separately.
- Stratum 1 is attached to the reference clock; each hop adds 1; lower wins; 15 is the maximum usable.
- `ntp master` gives stratum 8 by default.
- `show ntp associations` marks the chosen server with `*`; `show ntp status` shows your own stratum.
- Authentication needs the key, the trusted-key and, on clients, `key` on the server line.

Check yourself

1. A router syncs to a stratum 2 server. What stratum does it have? *(3.)*
2. Which command makes a router serve time with no upstream source, and what stratum does it get? *(`ntp master`; stratum 8.)*
3. A router synced via NTP but shows the wrong local time. What was missed? *(The time zone: NTP delivers UTC, so `clock timezone` must be set.)*

Do next

Do the lab, then download the Day 37 Anki cards. Learn the stratum table and the three modes.

Day 38: DNS

Exam domain: 4. IP Services · **Video:** DNS (30 min) · Lab: DNS

Key ideas

The exam topic is "explain the role of DHCP and DNS within the network" (4.3). Know the purpose and the basics, not the internals of DNS.

- **DNS resolves names to IP addresses.** People remember `youtube.com`; devices need an address. A host asks its DNS server, gets an answer, and then connects. Hosts learn the server by hand or through DHCP.
- **A router in the path needs no DNS configuration.** It just forwards the queries and replies like any other packets.
- **Record types in this lesson**

Record	Maps
A	name to an IPv4 address
AAAA ("quad A")	name to an IPv6 address
CNAME	name to another name (an alias)

- **Transport.** Port **53**. Ordinary queries and replies use **UDP**; messages larger than **512 bytes** use **TCP**. So a plain name lookup does not open a TCP connection, while the HTTP request that follows does (TCP 80).
- **DNS cache.** A host stores answers so it need not ask again. **Hosts file:** a local list of name and address pairs that works without DNS. It is an old, small-network alternative.
- **A Cisco router can be a DNS server, a DNS client, or both.** Being a server on a router is rare; real networks use a Windows or Linux server.
- **As a server**, it answers from its own host table. If it has no entry, it acts as a client and asks the server set with `ip name-server`, then passes the answer on.
- **As a client**, the router can use names in `ping` and similar commands. `ip domain lookup` is on by default, so you normally only need `ip name-server`.
- `ip domain name` sets a default domain appended to bare hostnames (`ping PC1` becomes `ping PC1.example.com`). SSH needs it later.
- `show hosts` lists both configured entries (flag `perm`) and learned ones (flag `temp`).

Commands

Windows prompt:

```
C:\> ipconfig /all            ! shows the configured DNS server
C:\> nslookup youtube.com     ! ask the DNS server for a name
C:\> ipconfig /displaydns     ! show the DNS cache
C:\> ipconfig /flushdns       ! clear the DNS cache
```

Cisco IOS:

```
R1(config)# ip dns server                    ! act as a DNS server
R1(config)# ip host PC1 192.168.0.101        ! add a static host entry
R1(config)# ip name-server 8.8.8.8           ! DNS server this router asks
R1(config)# ip domain lookup                 ! allow DNS queries (default; older form:
ip domain-lookup)
R1(config)# ip domain name example.com       ! default domain name (older form: ip
domain-name)
R1# show hosts                               ! static entries and cached DNS answers
```

Remember for the exam

- A = IPv4, AAAA = IPv6, CNAME = alias. Port 53, UDP normally, TCP above 512 bytes.
- Windows commands: `ipconfig /all` and `nslookup` show the DNS server; `/displaydns` shows the cache; `/flushdns` clears it.
- Routers forwarding DNS traffic need no DNS commands.
- `show hosts` is the IOS command for cached and static names.
- DHCP is how hosts normally learn their DNS server.

Check yourself

1. Which record type returns an IPv6 address for a name? *(AAAA.)*
2. PC1 uses 8.8.8.8 as its DNS server. What DNS configuration does the router between them need? *(None, it just forwards the packets.)*
3. A host's first visit to a website: does it open a TCP connection to the DNS server? *(No. The query goes over UDP 53; only the web server gets a TCP connection.)*

Do next

Do the lab, then download the Day 38 Anki cards. Practice the four Windows commands.

Day 39: DHCP

Exam domain: 4. IP Services · **Video:** DHCP (37 min) · Lab: DHCP

Key ideas

The exam topics are the role of DHCP (4.3) and "configure and verify DHCP client and relay" (4.6).

- **DHCP hands out settings automatically:** IP address, mask, default gateway, DNS server and more. It is used for client devices (PCs, phones). Routers and servers usually keep static addresses. Small networks use the home router as the server; large ones use a Windows or Linux server.
- **Leases.** Addresses are leased for a time and can be released early. Short leases return addresses of devices that left (a cafe, an airport) to the pool.
- **Ports.** Servers use **UDP 67**, clients use **UDP 68**. A client message is 68 to 67; a server message is 67 to 68.

The four messages (DORA)

Message	From	Sent as
Discover	client	broadcast (source 0.0.0.0, destination 255.255.255.255, destination MAC all Fs)
Offer	server	unicast or broadcast, as the client asked
Request	client	broadcast; names the chosen server, because several may have offered
Ack	server	unicast or broadcast, as the client asked

- A client that releases its address sends a **Release**, unicast to the server.
- Offer and Ack follow the client's choice through a flag in the message. Some clients cannot receive unicast before they have an address, so they ask for broadcast.
- **Relay.** Broadcasts stop at the router, so a central server in another subnet never hears Discover. A **relay agent** on the client-facing interface converts the broadcast into unicast to the server (source becomes the router's interface address) and passes replies back to the client. It needs a route to the server.
- **Server on a router.** A pool defines the network, gateway, DNS server and lease. Excluded addresses are never handed out. Make one pool per subnet served. The first free address after the excluded range goes to the first client.
- **Client on a router.** Rare, but `ip address dhcp` on an interface makes it ask.

Commands

```
R1(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.10   ! never lease these
R1(config)# ip dhcp pool LAN1                                    ! create a pool
R1(dhcp-config)# network 192.168.1.0 /24                         ! addresses to lease
(255.255.255.0 also works)
R1(dhcp-config)# dns-server 8.8.8.8                              ! DNS server given to
clients
R1(dhcp-config)# domain-name example.com                         ! domain name given to
```

```
clients
R1(dhcp-config)# default-router 192.168.1.1                    ! default gateway
R1(dhcp-config)# lease 0 5 30                                  ! days hours minutes
(lease infinite = never expires)
R1# show ip dhcp binding                                       ! clients with address,
MAC, expiry
R1(config)# interface g0/1
R1(config-if)# ip helper-address 192.168.10.10                 ! relay: DHCP server
address
R2(config-if)# ip address dhcp                                 ! make this interface a
DHCP client
R1# show ip interface g0/1                                     ! helper address, or
"address determined by DHCP"
```

Windows: `ipconfig /release` sends a Release; `ipconfig /renew` starts the exchange again; `ipconfig /all` shows lease times and the server.

Remember for the exam

- DORA order: Discover, Offer, Request, Ack. UDP 67 server, 68 client.
- Discover and Request are always broadcast. Offer, Ack and Release can be unicast.
- Relay agent = `ip helper-address <server>` on the interface facing the clients.
- Use a relay only when the router is not the server and the clients' LAN has no server.
- `show ip dhcp binding` lists current leases on a router server.

Check yourself

1. Which two DORA messages are always broadcast? *(Discover and Request.)*
2. A Cisco router has clients on g0/1 and the DHCP server is on another network. Which command goes on g0/1? *(`ip helper-address` with the server's address.)*
3. What source IP does a Discover carry, and why? *(0.0.0.0, because the client has no address yet.)*

Do next

Do the lab, then download the Day 39 Anki cards. Write out the three router roles from memory.

Day 40: SNMP

Exam domain: 4. IP Services · **Video:** SNMP (29 min) · Lab: SNMP

Key ideas

The exam topic is "explain the function of SNMP in network operations" (4.4). Configuration is not required, so treat the commands below as lab practice.

- **SNMP (Simple Network Management Protocol)** monitors and manages devices over the network. It dates from 1988.
- **Two roles. Managed devices** (routers, switches, firewalls) are the things being watched. The **NMS** (network management station or system) is the SNMP "server" that watches them.
- **Three jobs:** a managed device can notify the NMS of an event, such as an interface going down; the NMS can ask a device for its status, such as CPU load; and the NMS can change a device's settings, such as an IP address.
- **Software parts.** On the NMS, the **SNMP manager** talks to devices, and an SNMP application shows the admin alerts and charts. On each device, the **SNMP agent** talks to the manager and keeps the **MIB** (management information base). The MIB holds the variables, such as interface status, throughput, CPU use and temperature. Each variable has an **OID** (object ID) in a tree, for example the one for the system name (`sysName`).

Versions

Version	Notes
SNMPv1	original; community strings
SNMPv2c	adds GetBulk for efficient mass retrieval; community strings (the "c")
SNMPv3	adds authentication and encryption; the preferred, secure choice

- **Community strings** are plain passwords. In v1 and v2c the community and messages cross the network unencrypted, so a captured packet shows them.

Messages

Class	Messages	Direction
Read	Get, GetNext, GetBulk	NMS to agent
Write	Set	NMS to agent
Notification	Trap, Inform	agent to NMS
Response	Response	the reply to a request

- **Get** reads a variable, **GetNext** walks to the next OID, **GetBulk** (v2) is the efficient version. **Set** changes a variable.
- **Trap** is unacknowledged and so unreliable. **Inform** is acknowledged with a Response.
- **Ports (UDP):** agents listen on **161**; the manager listens on **162**. Gets and Sets go to 161; Traps and Informs go to 162.

Commands

Not tested, but useful for the lab (SNMPv2c):

```
R1(config)# snmp-server contact admin@example.com                ! optional contact
info
R1(config)# snmp-server location Seoul-rack1                     ! optional location
R1(config)# snmp-server community Read1 ro                       ! read-only
community string
R1(config)# snmp-server community Write1 rw                      ! read/write
community string
R1(config)# snmp-server host 192.168.1.1 version 2c Read1        ! send notifications
to the NMS
R1(config)# snmp-server enable traps snmp linkdown linkup        ! traps on interface
up and down
R1(config)# snmp-server enable traps config                      ! traps on
configuration changes
```

Remember for the exam

- NMS asks and manages; the agent answers and reports.
- MIB = variables; OID = the name of one variable.
- Trap is unreliable, Inform is acknowledged, GetBulk arrived with v2.
- UDP 161 agent, UDP 162 manager.
- Only v3 encrypts. `ro` can read; `rw` can read and change.

Check yourself

1. Which message does an agent send when an interface fails, and does it get an answer? *(A Trap; no, it is unacknowledged. An Inform would be answered.)*
2. Which UDP port does an NMS listen on for notifications? *(162.)*
3. Which SNMP version first offered encryption and authentication of messages? *(SNMPv3.)*

Do next

Do the lab, then download the Day 40 Anki cards. Drill the message table and both port numbers.

Day 41: Syslog

Exam domain: 4. IP Services · **Video:** Syslog (28 min) · Lab: Syslog

Key ideas

The exam topic is "describe the use of syslog features including facilities and levels" (4.5). Configuration is not listed, but the commands are good lab practice.

- **Syslog** is the standard for logging events: interfaces up or down, routing neighbors changing, restarts, configuration changes. Messages can appear in the CLI, be kept in RAM, or go to an external server. Timestamps are only trustworthy with NTP in place.
- **Message format:** `seq: timestamp: %FACILITY-SEVERITY-MNEMONIC: description`. The sequence number and timestamp may be absent, depending on configuration.

Example: `%LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up`. The facility is `LINK` (which process produced it), severity is `3` (error), the mnemonic is `UPDOWN`, and the rest is the description.

Severity levels (0 is the most severe)

Level	Keyword
0	Emergency
1	Alert
2	Critical
3	Error
4	Warning
5	Notice (called Notification in IOS)
6	Informational
7	Debugging

Mnemonic: **E**very **A**wful **C**at **E**ats **W**arm **N**oodles **I**n **D**ishes.

- **Setting a level includes everything more severe.** `logging buffered 6` keeps levels 0 through 6.
- **Vendors differ** in how they assign levels to events.

Where messages go

Destination	Default	Command
Console	on, levels 0 to 7	`logging console`
VTY lines (Telnet or SSH)	off	`logging monitor`
Buffer (RAM)	on, levels 0 to 7	`logging buffered`
Syslog server	not configured	`logging host`, level set by `logging trap`

- Syslog servers listen on **UDP 514**.

- A VTY session also needs `terminal monitor`, entered again in every new session.
- **Syslog versus SNMP:** syslog only pushes event messages from device to server. SNMP can also poll devices (Get) and change them (Set). They complement each other.

Commands

```
R1(config)# logging console 6                      ! console: levels 0-6
R1(config)# logging monitor informational          ! VTY sessions (name or number)
R1(config)# logging buffered 8192 6                ! RAM buffer: size in bytes
(optional), level
R1(config)# logging 192.168.1.100                  ! send to a server (logging host
works too)
R1(config)# logging trap debugging                 ! levels sent to the server
R1# terminal monitor                               ! show logs in this Telnet or SSH
session
R1(config)# line console 0
R1(config-line)# logging synchronous               ! reprint your typing after a log
message
R1(config)# service timestamps log datetime        ! timestamp with date and time (or
uptime)
R1(config)# service sequence-numbers               ! add sequence numbers
R1# show logging                                   ! settings and buffer contents
```

Remember for the exam

- Learn all eight numbers and keywords, plus Notice versus Notification for 5.
- Be able to pick facility, severity and mnemonic out of a message.
- Defaults: console and buffer yes (all levels); VTY no; server no.
- Syslog server port: UDP 514.
- Sequence number and timestamp are optional fields.

Check yourself

1. A message reads `%OSPF-5-ADJCHG`. What are the facility and severity? *(OSPF, and level 5, Notification.)*
2. After `logging buffered 4`, which levels are stored? *(0, 1, 2, 3 and 4.)*
3. Logs do not appear over SSH even with `logging monitor` set. What is missing? *(`terminal monitor` in that session.)*

Do next

Do the lab, then download the Day 41 Anki cards. Write the eight levels from memory twice.

Day 42: SSH

Exam domain: 4. IP Services · **Video:** SSH (31 min) · Lab: SSH

Key ideas

The exam topic is "configure network devices for remote access using SSH" (4.8). Unlike SNMP and syslog, the configuration is tested.

- **Console security.** By default the console needs no password. `line console 0` (only one console line, so one connection at a time) with `password` and `login` requires one. `login local` requires a username and password from the local database instead, and the line password stops being used. `exec-timeout` logs out idle sessions.
- **Management IP on a Layer 2 switch.** A switch does not route, but an SVI with an IP address lets you reach it. Because it has no routing table, give it a gateway with `ip default-gateway`.
- **Telnet** (TCP **23**, from 1969) gives remote CLI access but sends everything, including passwords, in plain text. The device you connect to is the server, the one connecting is the client.
- **SSH** (TCP **22**, 1995) encrypts and authenticates. SSHv2 (2006) is more secure than v1. A device that offers both reports version **1.99**, which is not a real version.
- **Requirements for SSH:**
  - The IOS image must support crypto; **K9** in the image name does, **NP** (no payload encryption) images do not. `show ip ssh` shows whether SSH is supported and enabled.
  - A non-default hostname, a domain name, and RSA keys. The keys are named from the hostname plus domain name (FQDN). Without a hostname or domain the key command refuses.
  - For SSHv2 the key modulus must be **768 bits or more** (2048 is typical).
- **VTY lines** carry Telnet and SSH sessions. There are 16 (0 to 15) and you configure them all together. For SSH, `login` alone does not work; use `login local`. `transport input` chooses allowed protocols: `telnet`, `ssh`, `telnet ssh`, `all` or `none`. Best practice is `ssh` only. Defaults vary by device.
- **Limiting who connects.** Apply a standard ACL to the VTY lines with `access-class`. Do not mix up `access-class` (VTY lines), `ip access-group` (interfaces) and `access-list` (creating the ACL). Other traffic, such as ping, is not affected.
- An enable secret is needed too, or remote users cannot enter privileged mode.

Commands

```
SW1(config)# line console 0
SW1(config-line)# password cisco123               ! console password
SW1(config-line)# login                           ! require that password
SW1(config-line)# login local                     ! or: require a local username
instead
SW1(config-line)# exec-timeout 3 30               ! log out after 3 min 30 s idle
SW1(config)# interface vlan 1
SW1(config-if)# ip address 192.168.1.11 255.255.255.0   ! management address
```

```
SW1(config-if)# no shutdown                          ! SVI may be down by default
SW1(config)# ip default-gateway 192.168.1.1          ! gateway for a Layer 2 switch
SW1# show ip ssh                                     ! supported? enabled? version?
SW1(config)# hostname SW1                            ! must not be the default name
SW1(config)# ip domain name example.com              ! needed for the key name
SW1(config)# crypto key generate rsa modulus 2048 ! create RSA keys
SW1(config)# enable secret mysecret                  ! privileged mode password
SW1(config)# username admin secret mypass            ! local account
SW1(config)# ip ssh version 2                        ! SSHv2 only (recommended)
SW1(config)# access-list 1 permit host 192.168.1.50   ! allowed admin PC
SW1(config)# line vty 0 15                           ! all 16 lines
SW1(config-line)# login local                        ! use the local username
SW1(config-line)# transport input ssh                ! SSH only
SW1(config-line)# access-class 1 in                  ! apply the ACL to VTY
SW1(config-line)# exec-timeout 5                     ! optional idle timeout
```

Connect from a client with `ssh -l admin 192.168.1.11`.

Remember for the exam

- Telnet 23, plain text. SSH 22, encrypted. Prefer SSHv2.
- SSH needs: crypto image (K9), hostname, domain name, RSA keys (768+ for v2), a local user, `login local`, `transport input ssh`.
- Layer 2 switch: SVI address plus `ip default-gateway`.
- The ACL goes on the VTY lines with `access-class`.

Check yourself

1. `crypto key generate rsa` is rejected on a router. Name two likely causes. *(Hostname still default, or no domain name configured.)*
2. Which command allows Telnet and SSH on the VTY lines, and which allows only SSH? *(`transport input telnet ssh`; `transport input ssh`.)*
3. Can `login` (no `local`) be used for SSH logins? *(No. SSH needs a username, so `login local`, or an authentication server.)*

Do next

Do the lab, then download the Day 42 Anki cards. Build the SSH setup from a blank switch without notes.

Day 43: FTP and TFTP

Exam domain: 4. IP Services · **Video:** FTP & TFTP (31 min) · Lab: FTP & TFTP

Key ideas

The exam topic is to describe the capabilities and function of TFTP and FTP in the network. Both move files between a client and a server. The most common job for a network engineer is upgrading a device's IOS: download the image from the vendor, place it on a server the device can reach, copy it into the device's flash, then reboot into it.

TFTP (Trivial File Transfer Protocol)

- Very basic: a client can only copy a file to or from the server. It cannot list files, make or delete directories, or ask what files exist, so you must know the file name in advance.
- **No authentication and no encryption.** Anyone who can reach the server can request files, and everything is plain text. Use it for small files in a controlled network.
- Uses **UDP port 69**. UDP gives no reliability, so TFTP adds its own: every data message is acknowledged, and a timer triggers a resend if the reply never arrives. The two sides alternate in lock step (one data message, one ack), which is simpler but slower than TCP's sliding window.

FTP (File Transfer Protocol)

- Older and richer: clients send commands to browse and create directories, list files, delete and so on, as well as transfer files.
- **Username and password authentication**, but still no encryption (the password crosses the network in plain text). Secure options: **FTPS** (FTP over SSL/TLS, an upgrade of FTP) and **SFTP** (SSH File Transfer Protocol, a different protocol with a similar name).
- Uses **TCP** and two kinds of connection:
  - **Control connection, TCP port 21.** The client opens it and sends commands; the server replies. It stays up for the whole session.
  - **Data connection, TCP port 20.** Opened and closed as needed to carry the actual data.

	FTP	TFTP
Transport	TCP, ports 21 (control) and 20 (data)	UDP, port 69
Authentication	Username and password	None
Encryption	None (use FTPS or SFTP)	None
Functions	Transfer, list, create and delete directories	Transfer only
Complexity	Higher	Minimal

Active vs passive FTP. These modes apply only to the data connection; the client always opens the control connection.

- **Active mode** (the default): the **server** initiates the data connection to the client.
- **Passive mode:** the **client** initiates the data connection. Use it when the client is behind a firewall, because the firewall normally blocks connections started from outside but allows replies to

connections started from inside.

IOS file systems. `show file systems` lists them. A *disk* type such as flash holds the IOS image (copied into RAM at boot). *NVRAM* keeps its contents without power and holds the startup-config. The *network* type represents remote servers such as FTP and TFTP. Jeremy notes this topic was dropped from the current exam blueprint, so treat it as background.

IOS upgrade steps

1. Check what you run now (`show version`) and what is in flash (`show flash`).
2. Copy the new image from the server into flash.
3. Tell the router to boot it with `boot system`; without this it boots the first IOS file it finds in flash.
4. Save the configuration, or the `boot system` line is lost at reload.
5. Reload, then confirm with `show version`.
6. Optionally delete the old image to free space.

The `copy` command is always source then destination. The same command saves your config (`copy running-config startup-config`).

Commands

```
R1# show version                                    ! running IOS image name and
version
R1# show flash:                                     ! files stored in flash
R1# show file systems                               ! list IOS file systems
R1# copy tftp: flash:                               ! prompts for server IP, source
file name, destination name
R1(config)# ip ftp username cisco                   ! FTP login the router presents to
the server
R1(config)# ip ftp password cisco                   ! matching password (must also
exist on the server)
R1# copy ftp: flash:                                ! same prompts as TFTP
R1(config)# boot system flash:c2900-universalk9-mz.SPA.155-3.M4a.bin   ! boot this
image (use your real file name)
R1# copy running-config startup-config             ! save so the boot system line
survives
R1# reload                                          ! restart into the new image
R1# delete flash:old-image.bin                      ! remove the old image (asks to
confirm)
```

Remember for the exam

- FTP: TCP, control on **21**, data on **20**, with username and password. TFTP: UDP **69**, no authentication.
- Neither FTP nor TFTP encrypts; FTPS and SFTP are the secure options.
- Active mode: server opens the data connection. Passive mode: client opens it, which suits a client behind a firewall.

- TFTP cannot list files or manage directories; it only copies a file to or from a server.
- NVRAM holds the startup-config; flash holds the IOS image.
- The `K9` in an image name means it supports cryptographic features such as SSH.

Check yourself

1. Which two ports does FTP use, and what does each carry? *(TCP 21 carries the control connection with commands; TCP 20 carries the data connections.)*
2. A client behind a firewall must reach an FTP server. Which mode should it use and why? *(Passive; the client starts the data connection, which the firewall allows, instead of the server trying to connect inward.)*
3. After copying a new IOS image into flash, what must you do before `reload` so the router actually boots it? *(Enter* `boot system flash:<file>` *and save the configuration.)*

Do next

Do the Day 43 lab, copying a file from a TFTP server into flash. Download the Day 43 Anki cards, then start Day 44 on NAT.

Day 44: NAT, part 1

Exam domain: 4. IP Services · **Video:** NAT (Part 1) (32 min) · Lab: Static NAT

Key ideas

The exam topic is to configure and verify inside source NAT using static and pools, and PAT. Today covers the background, the vocabulary and static NAT.

Why NAT exists. IPv4 does not have enough addresses. IPv6 is the long-term fix, but three short-term measures stretched IPv4's life: CIDR, private addresses and NAT.

Private IPv4 ranges (RFC 1918)

Range	Block	Span
10.0.0.0/8	one class A block	10.0.0.0 to 10.255.255.255
172.16.0.0/12	16 class B-sized blocks	172.16.0.0 to 172.31.255.255
192.168.0.0/16	256 class C-sized blocks	192.168.0.0 to 192.168.255.255

- The prefix lengths are just the size of the range; you can subnet inside them however you like.
- Private addresses need not be globally unique, so many networks reuse them. They **cannot be routed over the Internet**, because ISPs drop them. Public addresses must be unique.
- NAT solves both problems: it swaps a private address for a public one at the edge, so hosts can reach the Internet and duplicate private addresses do not collide.

What NAT does. It rewrites source and/or destination addresses in packets. The CCNA needs only **source NAT**, where the router rewrites the source address on the way out and reverses it on the reply (that reversal on the way back is not "destination NAT").

Static NAT is a fixed **one-to-one** mapping from one address to another, usually private to public. Each inside host needs its own public address, so it does not save addresses. It is used when a host must always appear as the same address (for example a server reachable from outside). The router will refuse a second mapping to a public address that is already used.

The four address terms. "Inside" or "outside" is where the host is. "Local" or "global" is whose point of view you take.

Term	Meaning	Typical example
Inside local	inside host's address as seen from the inside; what is configured on the host	192.168.0.167
Inside global	inside host's address as seen from the outside, after NAT	100.0.0.1
Outside local	outside host's address as seen from the inside	8.8.8.8
Outside global	outside host's address as seen from the outside	8.8.8.8

Outside local and outside global are identical unless destination NAT is used, which is beyond the CCNA. So for the exam, expect them to match.

Configuration in three parts: mark the inside interface, mark the outside interface, then define the mappings. NAT applies to traffic crossing from an inside interface to an outside interface. You must own any public addresses you map to.

Reading `show ip nat translations`**.** Columns are protocol, inside global, inside local, outside local, outside global (global comes before local, which is easy to get backwards). Static entries stay in the table permanently. Each time traffic uses a static mapping, extra dynamic entries (with port numbers) are added, and those time out. With static NAT the ports are not changed.

Commands

```
R1(config)# interface g0/1
R1(config-if)# ip nat inside                                        ! inside interface
(faces the LAN)
R1(config-if)# interface g0/0
R1(config-if)# ip nat outside                                       ! outside interface
(faces the Internet)
R1(config-if)# exit
R1(config)# ip nat inside source static 192.168.0.167 100.0.0.1    ! inside local, then
inside global
R1(config)# ip nat inside source static 192.168.0.168 100.0.0.2    ! second host needs
its own global address
R1# show ip nat translations                                        ! the translation
table
R1# show ip nat statistics                                          ! active and peak
translations, inside/outside interfaces
R1# clear ip nat translation *                                      ! clears dynamic
entries only; static entries stay
```

Remember for the exam

- Private ranges: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 (RFC 1918).
- Static NAT command order: `ip nat inside source static <inside local> <inside global>`.
- The table lists inside global first, then inside local.
- `clear ip nat translation *` does not remove static mappings; only removing the config line does.
- One public address cannot be statically mapped to two inside hosts.
- Without `ip nat inside` and `ip nat outside` on the interfaces, nothing is translated.

Check yourself

1. Is 172.32.5.9 a private address? *(No. The 172.16.0.0/12 range ends at 172.31.255.255.)*
2. PC1 (192.168.0.167) reaches 8.8.8.8 and is translated to 100.0.0.1. Which is the inside local and which the inside global address? *(Inside local is 192.168.0.167; inside global is 100.0.0.1.)*
3. A router has three static mappings plus several dynamic entries. How many translations remain after `clear ip nat translation *`? *(Three, the static ones.)*

Do next

Do the static NAT lab. Download the Day 44 Anki cards and be sure you can explain the four address terms from memory before Day 45.

Day 45: NAT, part 2

Exam domain: 4. IP Services · **Video:** NAT (Part 2) (30 min) · Lab: Dynamic NAT

Key ideas

Day 44 covered static NAT. Today adds the two forms you will use most: dynamic NAT and PAT. First, one point left over from static NAT: the one-to-one mapping works in both directions. An outside host can send traffic to the inside global address and the router translates it to the inside local address, even if the inside host never started a conversation.

Dynamic NAT

- The router creates one-to-one mappings automatically from a pool of inside global addresses, as hosts need them, and removes them later.
- Two ingredients:
  - An **ACL** that *identifies* which source addresses get translated. It is not applied to an interface, so it never blocks anything. A packet the ACL permits is translated; a packet it denies is simply **not translated** (still forwarded, just with its private address).
  - A **NAT pool** that defines the range of inside global addresses.
- The ACL uses a **wildcard mask**, for example `0.0.0.255` for a /24. A subnet mask there is a common mistake.
- Pool rules: you give the first address, last address and the prefix length (or netmask). IOS rejects the command if the two addresses are not in the same subnet for that prefix length.
- **Pool exhaustion:** the mappings are still one-to-one, so a pool of 10 addresses serves at most 10 inside hosts at once. If an 11th host needs NAT, the router **drops its packet** until an address frees up.
- Entries free up on a timeout (the mapping itself defaults to 24 hours, and the timer resets each time it is used) or when you clear them manually. The short-lived per-protocol entries (UDP, ICMP) disappear after about a minute. Timers are not an exam detail.

PAT (Port Address Translation), also called NAT overload

- Translates the IP address **and** the Layer 4 port when needed, so many inside hosts share **one** public address at the same time. TCP and UDP ports are 16 bits, so there are over 65,000 to hand out.
- The router tracks each flow by its port. If two hosts happen to choose the same source port, the router changes one of them, so the replies can be sorted back to the right host. If the ports are already different, it leaves them alone.
- In the table PAT shows many inside local addresses sharing one inside global address (many-to-one), not separate one-to-one lines.
- PAT is by far the most common NAT type; a home router uses it.

Two ways to configure PAT

Method	Translates to	Keyword
Pool	addresses in a pool (a small pool is enough)	`pool <name> overload`
Interface	the router's own outside interface address	`interface <name> overload`

The interface method is probably more common. When you translate to the router's own address, it should be the outside interface (the public one), not the inside one.

Summary of the three types

Type	Mapping	Saves public addresses?
Static	fixed one-to-one	No
Dynamic	one-to-one, assigned from a pool as needed	No (limited by pool size)
PAT	many-to-one using ports	Yes

Commands

```
R1(config)# access-list 1 permit 192.168.0.0 0.0.0.255                          ! ACL
picks the traffic to translate (wildcard mask)
R1(config)# ip nat pool POOL1 100.0.0.1 100.0.0.10 prefix-length 24             ! pool of
inside global addresses
R1(config)# ip nat inside source list 1 pool POOL1                              ! dynamic
NAT: ACL 1 maps to POOL1
R1(config)# ip nat inside source list 1 pool POOL1 overload                     ! PAT
using the pool
R1(config)# ip nat inside source list 1 interface g0/0 overload                 ! PAT
using the outside interface address
R1# show ip nat translations                                                    ! table
of translations
R1# show ip nat statistics                                                      ! counts,
plus which ACL maps to which pool
R1# clear ip nat translation *                                                  ! clear
the dynamic entries
```

Remember that the `ip nat inside` and `ip nat outside` interface commands from Day 44 are still required.

Remember for the exam

- The ACL in NAT selects traffic to translate; it does not drop anything. Denied traffic goes out untranslated.
- Dynamic NAT command: `ip nat inside source list <acl> pool <name>`. Add `overload` for PAT.
- `ip nat pool <name> <first> <last> {prefix-length <n> | netmask <mask>}`: the range must sit inside one subnet.
- Pool exhausted means the packet is dropped; use PAT to avoid it.
- ACL uses a wildcard mask; the NAT pool uses a normal mask or prefix length.
- Dynamic NAT entries are cleared by `clear ip nat translation *`; static entries are not.

Check yourself

1. Which NAT type lets hundreds of private hosts share one public address, and what makes it possible? *(PAT/overload, which tracks each flow by a unique Layer 4 port number.)*
2. A host's address is denied by the ACL used for NAT. What happens to its packets? *(They are not translated, but they are still routed normally.)*
3. A pool holds 10 addresses and dynamic NAT (no overload) is in use. What happens when an 11th host sends traffic while all 10 are in use? *(The router discards the packet until a translation expires or is cleared.)*

Do next

Do the dynamic NAT lab, then try changing it to PAT by adding `overload`. Download the Day 45 Anki cards and move on to QoS on Day 46.

Day 46: QoS, part 1

Exam domain: 4. IP Services · **Video:** QoS (Part 1) (33 min) · Lab: Voice VLANs

Key ideas

Today builds up to QoS through IP phones, voice VLANs and Power over Ethernet, because voice traffic is the usual reason to apply QoS. The exam topic is to describe the effect of QoS (classification, marking, queuing, congestion, policing, shaping) on a network; part 1 covers the foundations and part 2 the mechanisms.

IP phones and voice VLANs

- An IP phone sends voice as IP packets (VoIP), unlike the traditional telephone network (PSTN/POTS).
- A phone contains a small **three-port switch**: one port to the wall switch, one to a PC, one internal to the phone. The PC plugs into the phone, so a desk needs only **one switch port** instead of two.
- Best practice is to place voice and data in **separate VLANs**, so QoS can easily pick out the voice traffic and prioritize it.
- On the switch port: the PC sends **untagged** frames in the access (data) VLAN, and the switch uses **CDP** to tell the phone to tag its frames with the **voice VLAN** ID. (With LLDP-MED, the phone can learn it in a similar way.)
- Despite carrying two VLANs, the port is still an **access port**. `show interfaces trunk` does not list it, and `show interfaces g0/0 switchport` shows operational mode "static access" along with both the access VLAN and the voice VLAN. If no access VLAN is set, data is in VLAN 1.

Power over Ethernet (PoE)

- A **PSE** (power sourcing equipment, usually a switch) supplies power over the same Ethernet cable that carries data to a **PD** (powered device: IP phone, camera, access point). The switch converts AC from the outlet to DC.
- Before powering, the switch sends weak signals and checks the response, to learn whether the device needs power and how much, so it does not damage a device that does not.
- **Power policing** stops a PD drawing more than it was allotted. It is not a configuration topic on the exam, but know the purpose.

Standard	Name	Max power per port
Cisco ILP	Cisco Inline Power (pre-standard)	about 7 W
802.3af	PoE, Type 1	15.4 W
802.3at	PoE+, Type 2	30 W
802.3bt	Type 3 / Type 4	60 W / 100 W

Jeremy says memorizing these is unlikely to be needed; the concept matters more.

Why QoS exists. Voice and data once had separate networks. In a **converged network** they share one IP network and compete for bandwidth. When the network is busy, delay-sensitive traffic suffers. QoS is a set of tools that give different traffic different treatment.

What QoS manages

- **Bandwidth:** link capacity in bits per second; QoS can reserve a share for certain traffic.
- **Delay:** one-way delay (source to destination) or two-way (there and back).
- **Jitter:** variation in one-way delay between packets of the same application. Phones have a jitter buffer; too much jitter overruns it.
- **Loss:** percentage of packets that never arrive (bad cables, or queues so full that packets are dropped).

Targets for interactive audio (a call): one-way delay **150 ms or less**, jitter **30 ms or less**, loss **1% or less**.

Queuing and congestion

- When a device receives traffic faster than it can send, packets wait in a queue per interface. The default is **FIFO** (first in, first out), with no priority for any traffic.
- A full queue causes **tail drop**: new arrivals are discarded.
- Tail drop causes **TCP global synchronization**: every TCP sender slows down at once, the link goes underused, then they all speed up together and congest it again, in waves.
- **RED (random early detection)** starts dropping random packets from some TCP flows once the queue passes a threshold, so only some senders slow down. **WRED** (weighted RED) uses different thresholds per traffic class, so lower-priority traffic is dropped sooner.

Commands

```
SW1(config)# interface g0/0
SW1(config-if)# switchport mode access                  ! normal access port
SW1(config-if)# switchport access vlan 10               ! data VLAN (PC, untagged)
SW1(config-if)# switchport voice vlan 11                ! voice VLAN (phone, tagged;
learned via CDP)
SW1# show interfaces g0/0 switchport                    ! shows access VLAN and voice
VLAN
SW1(config-if)# power inline police                     ! power policing, default
action is err-disable
SW1(config-if)# power inline police action log          ! restart the port and log
instead of err-disabling
SW1# show power inline police g0/0                      ! policing status and action
```

An err-disabled PoE port is recovered with `shutdown` then `no shutdown`.

Remember for the exam

- `switchport voice vlan` adds a second VLAN to an access port; the port is not a trunk.
- Phone traffic is tagged; PC traffic through the phone is untagged.
- Audio targets: 150 ms delay, 30 ms jitter, 1% loss.
- FIFO is the default queue; tail drop leads to TCP global synchronization; RED and WRED prevent it by dropping early.

- RED and WRED decide what to drop; they are not forwarding methods.

Check yourself

1. A port has `switchport access vlan 10` and `switchport voice vlan 11`. Which traffic is tagged? *(The phone's voice traffic, in VLAN 11; the PC's data is untagged.)*
2. What problem does RED solve compared with tail drop? *(Global synchronization: dropping from only some flows avoids all TCP senders backing off and ramping up together.)*
3. Name the three recommended limits for a voice call. *(One-way delay of 150 ms or less, jitter of 30 ms or less, loss of 1% or less.)*

Do next

Do the voice VLAN lab. Download the Day 46 Anki cards, then watch Day 47 for how QoS classifies, marks and queues traffic.

Day 47: QoS, part 2

Exam domain: 4. IP Services · **Video:** QoS (Part 2) (42 min) · Lab: QoS

Key ideas

QoS configuration is not tested, but you must describe what each mechanism does. The flow on a device is: classify, mark, queue, schedule, transmit.

Classification sorts packets into traffic classes so they can be treated differently. Methods include an ACL (used only to identify traffic, as with NAT), **NBAR** (looks up to Layer 7 with deep packet inspection), and the QoS fields in the headers. **Marking** means setting those header fields so other devices can classify the packet easily.

Layer 2 marking: PCP / CoS

- A 3-bit field in the 802.1Q tag (defined by 802.1p), so values 0 to 7. Also called CoS (class of service).
- It exists only when there is a dot1q tag: trunks, and access ports with a voice VLAN. It cannot be used on untagged links such as router-to-router, which is its main limitation.
- Standard use: **0** best effort (the default), **3** critical applications (IP phones mark call signaling this way), **4** video, **5** voice. 6 and 7 are for network control traffic.

Layer 3 marking: the ToS byte in the IPv4 header (the IPv6 equivalent is the traffic class byte)

- Old use: 3 bits of **IP Precedence (IPP)**. Same idea as PCP: 5 voice, 4 video, 3 voice signaling, 0 best effort, 6 and 7 network control.
- Current use: **6 bits DSCP** (64 values) plus 2 bits ECN. DSCP was defined in 1998 (RFC 2474).

Standard DSCP markings

Name	DSCP	Purpose
DF (default forwarding)	0	best effort
EF (expedited forwarding)	46 (binary 101110)	low loss, delay and jitter; voice
AF (assured forwarding)	12 values, see below	four classes with three drop levels
CS (class selector)	8 times the number	backward compatible with IPP

- **AF** is written AF*xy*: *x* is the class (1 to 4; higher is better), *y* is the drop precedence (1 to 3; higher is dropped first by WRED). DSCP value = **8x + 2y**.

	Drop 1	Drop 2	Drop 3
Class 1	AF11 = 10	AF12 = 12	AF13 = 14
Class 2	AF21 = 18	AF22 = 20	AF23 = 22
Class 3	AF31 = 26	AF32 = 28	AF33 = 30
Class 4	AF41 = 34	AF42 = 36	AF43 = 38

AF41 is the best service of the set. There is no AF5x or AF6x.

- **CS0 to CS7** are DSCP 0, 8, 16, 24, 32, 40, 48, 56. CS0 equals DF.
- Common recommendations: voice EF, interactive video AF4x, streaming video AF3x, high-priority data AF2x, best effort DF. The network designer still chooses.

Trust boundary. The point where a device stops accepting the markings it receives and rewrites them by policy. If the boundary sits at the switch, even an IP phone's EF/CoS 5 gets reset. Best practice is to move the boundary to the **IP phone**: trust the phone's markings (set on the switch port), but do not trust the PC behind it, otherwise a user could mark their own traffic as high priority.

Queuing and scheduling

- Real QoS uses **multiple queues** per interface. Packets are classified (for example by DSCP), placed in a queue, and a **scheduler** picks which queue sends next. Only one frame leaves at a time.
- **Weighted round robin** takes packets from each queue in turn, taking more from higher-priority queues.
- **CBWFQ** (class-based weighted fair queuing): weighted round robin plus a guaranteed share of bandwidth per class during congestion.
- **LLQ** (low latency queuing): one or more queues are **strict priority**, always served first until empty. This keeps voice and video delay and jitter low. Risk: it can starve the other queues, so the priority queue is limited with policing.
- RED/WRED can run inside any queue to avoid tail drop.
- Marking alone does nothing. A packet marked EF is only prioritized if the device is configured with something like CBWFQ or LLQ.

Shaping vs policing: both limit the traffic rate.

	Shaping	**Policing**
Traffic above the rate	buffered in a queue and sent later	dropped
Short bursts above the rate	smoothed out by the buffer	allowed up to a configurable burst size

A common pairing: a customer pays for 300 Mbps on a gigabit port, so the ISP **polices** inbound traffic to 300 Mbps and the customer **shapes** outbound traffic to 300 Mbps to avoid drops.

Remember for the exam

- PCP/CoS: 3 bits in the dot1q tag, needs a tag. DSCP: 6 bits in the IP header, works end to end.
- EF = 46, DF = 0, AF = 8x + 2y, CS = 8 times n.
- AF41 beats AF43; both beat AF3x.
- Trust IP phone markings, not PC markings.
- LLQ = strict priority queue; CBWFQ = guaranteed bandwidth per class.
- Shaping buffers, policing drops.

Check yourself

1. What DSCP value is AF32, and what does the 2 mean? *(8x3 + 2x2 = 28; 2 is the middle drop precedence within class 3.)*

2. Why can't PCP classify traffic between two routers? *(PCP lives in the 802.1Q tag, and a routed link between routers has no VLAN tag.)*
3. The ISP polices to 300 Mbps while a customer sends faster. What happens to the excess traffic, and how could the customer avoid it? *(The excess is dropped after any allowed burst; the customer can shape outbound traffic to 300 Mbps so it is buffered instead.)*

Do next

Do the QoS lab. Download the Day 47 Anki cards and practice converting AF values to DSCP numbers before Day 48 on security.

Day 48: Security fundamentals

Exam domain: 5. Security Fundamentals · **Video:** Security Fundamentals (39 min) · Lab: Kali Linux Demo

Key ideas

This is a terminology-heavy day with no commands. Several of the terms are named directly in the exam topics, so learn the definitions precisely.

The CIA triad: the goals of security

Letter	Goal	Meaning
C	Confidentiality	only authorized users can read the data
I	Integrity	data is not changed by unauthorized users; it stays correct and authentic
A	Availability	systems and services are up and reachable by authorized users

Core terms

- **Vulnerability:** a potential weakness that could compromise confidentiality, integrity or availability.
- **Exploit:** something that can be used to take advantage of a vulnerability.
- **Threat:** the real possibility that a vulnerability gets exploited (for example an attacker who wants to do so).
- **Mitigation technique:** something that protects against a threat, applied wherever a vulnerability could be exploited: hosts, servers, switches, routers, firewalls, and physical measures such as locked racks. No system is ever perfectly secure.

Common attacks

Attack	What it is	CIA goal hit
DoS (denial of service)	overwhelm a target so it cannot serve legitimate users. Example: TCP SYN flood, where the attacker sends masses of SYNs, the target answers each with a SYN-ACK, and the final ACK never comes, filling the connection table	Availability
DDoS	the same, launched from many infected machines (a botnet)	Availability
Spoofing	using a fake source address (IP or MAC). A technique, not one attack. Example: DHCP exhaustion, where spoofed MAC addresses drain the DHCP pool	Often availability; ARP spoofing hits C and I
Reflection	attacker spoofs the target's IP as the source of requests to a reflector (for example a DNS server), so the replies flood the target	Availability
Amplification	a reflection where a small request triggers a much larger reply (DNS and NTP weaknesses are examples)	Availability
Man in the middle	attacker sits between two parties to eavesdrop or change traffic. Example: ARP spoofing/poisoning, where a forged ARP reply arriving last overwrites the victim's ARP entry with the attacker's MAC	Confidentiality and integrity

Attack	What it is	CIA goal hit
Reconnaissance	gathering information for later attacks, from public sources: `nslookup` for IP addresses, WHOIS for contact details, probing for open ports	Not an attack by itself

Malware (defined by how it spreads)

- **Virus:** attaches to a host program and spreads as that software is shared.
- **Worm:** standalone and spreads by itself without user action, which can congest a network; it may also carry a harmful payload.
- **Trojan horse:** harmful software disguised as something legitimate; spreads when users open or download it.

Social engineering targets people instead of systems, using psychological manipulation.

- **Phishing:** fraudulent emails that look like a trusted company and lead to a fake login page. **Spear phishing** is aimed at specific people or a company; **whaling** at high-profile targets such as executives; **vishing** is by phone; **smishing** is by text message.
- **Watering hole:** compromise a website the victim already trusts and visits.
- **Tailgating:** following an authorized person through a secured door.

Password attacks. Guessing; a **dictionary attack** (trying a list of common words and passwords); **brute force** (trying every combination). A strong password is long (more than 8 characters is better), mixes upper and lower case, numbers and special characters, and is changed regularly.

Multi-factor authentication (MFA) needs factors from at least two *different* categories: something you **know** (password, PIN), something you **have** (phone app, badge), something you **are** (fingerprint, face). Two biometrics, such as a fingerprint plus a retina scan, are one category, so that is not MFA.

Digital certificates prove the identity of the holder, mainly for websites. An organization sends a **CSR** (certificate signing request) to a **CA** (certificate authority), which creates and signs the certificate. A browser's padlock indicates a valid certificate.

AAA

Part	Question it answers	Example
Authentication	who are you?	logging in, ideally with MFA
Authorization	what may you do?	access to some files and services but not others
Accounting	what did you do?	logging logins, logouts and changes

A AAA server (Cisco's is ISE) uses **RADIUS** (open standard, UDP 1812 and 1813) or **TACACS+** (Cisco proprietary, TCP 49).

Security program elements

- **User awareness:** informing staff of threats, for example sending harmless fake phishing emails.
- **User training:** formal sessions on policy and safe behavior, at hiring and regularly after.
- **Physical access control:** only authorized people reach network closets and data centers, for example a door that needs a badge (have) plus a fingerprint (are). Badge permissions can be changed

centrally.

Remember for the exam

- CIA = confidentiality, integrity, availability. DoS and DDoS hit availability; man in the middle hits confidentiality and integrity.
- Vulnerability (weakness), exploit (tool), threat (the chance of exploitation), mitigation (protection).
- MFA needs different factor categories, not two of the same.
- Authentication = identity, authorization = permissions, accounting = records.
- RADIUS: UDP 1812/1813. TACACS+: TCP 49.
- Phishing variants: spear (targeted), whaling (executives), vishing (voice), smishing (SMS).

Check yourself

1. What makes an amplification attack different from a plain reflection attack? *(A small request causes the reflector to send a much larger response to the spoofed target.)*
2. A door needs a badge and a PIN. Which authentication factors are used? *(Something you have and something you know; this is MFA.)*
3. Which AAA function records when a user logged in? *(Accounting.)*

Do next

Do the Kali Linux demo if you want to see attacks in practice. Download the Day 48 Anki cards, then continue to the next security topics on Day 49.

Day 49: Port security

Exam domain: 5. Security Fundamentals · **Video:** Port Security (34 min) · Lab: Port Security

Key ideas

The exam topic is configuring Layer 2 security features, and port security is one of them.

- **Port security** limits which source MAC addresses, and how many, may send frames into a switch port. It is configured per interface. Addresses learned or configured this way are called **secure MAC addresses**.
- **It helps in two ways.** It keeps unknown devices off the network, although MAC spoofing means it is not a perfect lock. It also caps the number of MAC addresses on a port, which blunts attacks that use thousands of fake MACs, such as DHCP starvation or filling the switch's MAC table (after which the switch floods everything).
- **The port must be static.** The default mode is dynamic auto, and `switchport port-security` is rejected on a dynamic port. Set `switchport mode access` (or trunk) first.
- **Defaults once enabled:** maximum 1 secure address, violation mode shutdown, aging time 0 (never ages out), no sticky learning.
- **Allowed addresses** can be configured by hand, learned dynamically (the first frames that arrive fill the allowance), or a mix. A phone with a PC behind it needs a maximum of 2.
- **Aging.** The default type is **absolute**: the timer starts when the address is learned and removes it even if traffic continues. **Inactivity** restarts the timer on every frame from that address. By default only dynamically learned addresses age; `aging static` extends this to manually configured ones.
- **Sticky** learning saves dynamically learned addresses into the running config as `mac-address sticky` lines. They never age out, but are lost on reload unless you save the config. Enabling sticky converts addresses already learned; disabling it turns them back into ordinary dynamic ones. In the MAC address table, sticky and static secure addresses show type static, even though sticky ones were learned.

Violation mode	Drops unauthorized frames	Interface err-disabled	Syslog/SNMP	Violation counter
shutdown (default)	yes	yes	one message when it shuts down	set to 1; resets when the port comes back up
restrict	yes	no	on every violation	+1 per frame
protect	yes	no	none	not incremented

- **Recovering an err-disabled port.** First unplug the offending device. Then either `shutdown` / `no shutdown` on the interface, or enable **err-disable recovery** for the cause. Recovery is off for every cause by default, and the interval is 300 seconds. If you leave the intruder connected, a dynamically learned secure MAC is cleared on shutdown, so the intruder could become the new secure address.
- In `show port-security interface`, a shut-down port shows "Secure-shutdown"; `show interfaces status` shows it as err-disabled.

Commands

```
SW1(config-if)# switchport mode access                          ! port must be static
access or trunk
SW1(config-if)# switchport port-security                        ! enable port security
(defaults apply)
SW1(config-if)# switchport port-security maximum 2              ! allow 2 secure MAC
addresses
SW1(config-if)# switchport port-security mac-address aaaa.aaaa.aaaa  ! manually allow a
MAC
SW1(config-if)# switchport port-security mac-address sticky     ! learn dynamically and
save to running config
SW1(config-if)# switchport port-security violation restrict     ! protect | restrict |
shutdown
SW1(config-if)# switchport port-security aging time 30          ! minutes (default 0 =
never)
SW1(config-if)# switchport port-security aging type inactivity  ! absolute (default) |
inactivity
SW1(config-if)# switchport port-security aging static           ! also age manually
configured MACs
SW1(config)# errdisable recovery cause psecure-violation        ! auto re-enable after
a violation
SW1(config)# errdisable recovery interval 180                   ! seconds (default 300)
SW1# show port-security                                         ! summary of all
secured ports
SW1# show port-security interface g0/1                          ! mode, counts,
violations, aging
SW1# show mac address-table secure                              ! secure MAC addresses
SW1# show errdisable recovery                                   ! recovery causes and
timer
```

Remember for the exam

- Default violation mode is shutdown; default maximum is 1.
- Restrict and protect both drop traffic and keep the port up. Only restrict logs and counts.
- Port security needs a static access or trunk port.
- Sticky addresses sit in the running config; run `write memory` to keep them.
- Err-disable recovery cause for port security is `psecure-violation`.

Check yourself

1. A port is set to protect mode and an unknown laptop sends traffic. What happens? *(Its frames are silently dropped; the port stays up and nothing is logged or counted.)*

2. Why can't you enable port security on a port left at the default switchport mode? *(The default is dynamic auto; the port must be a static access or trunk port.)*
3. What is the difference between absolute and inactivity aging? *(Absolute removes the address when the timer ends regardless of traffic; inactivity restarts the timer on each frame from it.)*

Do next

Do the Port Security lab, then try all three violation modes and compare `show port-security interface` after each. Day 50 builds on this with DHCP snooping.

Day 50: DHCP snooping

Exam domain: 5. Security Fundamentals · **Video:** DHCP Snooping (28 min) · Lab: DHCP Snooping

Key ideas

- **DHCP snooping** is a switch feature that filters DHCP messages arriving on **untrusted** ports. Non-DHCP traffic is not affected.
- **All ports are untrusted by default.** Trust the ports that lead toward the legitimate DHCP server or the rest of your infrastructure (uplinks, links to the server or relay). Leave ports facing end hosts untrusted. Messages on trusted ports are forwarded without inspection.
- **Attacks it stops**
  - **DHCP starvation (exhaustion):** flooding Discover messages from spoofed MAC addresses to drain the server's pool, denying service to real clients.
  - **DHCP poisoning (rogue server):** an illegitimate server answers clients first, handing out itself as default gateway to sit in the middle of their traffic. It is most likely to win when the real server is remote behind a relay.
- **Who sends what.** Servers send OFFER, ACK and NAK. Clients send DISCOVER, REQUEST, RELEASE and DECLINE.

Message arrives on an untrusted port	Action
Server message (OFFER, ACK, NAK)	discarded
DISCOVER or REQUEST	forwarded only if the frame's source MAC matches the client hardware address (chaddr) field inside the DHCP message
RELEASE or DECLINE	forwarded only if the packet's source IP and the receiving interface match the binding table entry

- The MAC check can be defeated if the attacker spoofs both the frame and the chaddr, which is why rate limiting matters.
- **Binding table.** When a client leases an address successfully, the switch records its MAC, IP, lease time, VLAN and interface. It does not store the default gateway. Dynamic ARP inspection uses this table on Day 51.
- **Rate limiting.** You can cap DHCP messages per second per interface. Exceed the limit and the interface is err-disabled. It is off by default on every port. Recover with `shutdown` / `no shutdown` or err-disable recovery. This is the defence against Discover floods.
- **Option 82** (relay agent information) records which relay agent and interface a request came through. With snooping on, a Cisco switch adds Option 82 to client messages even when it is not a relay. A switch drops messages carrying Option 82 that arrive on an untrusted port, and a DHCP server may reject them because they did not come from a relay. In a plain Layer 2 design, turn the insertion off with `no ip dhcp snooping information option` on the switches.

Commands

```
SW2(config)# ip dhcp snooping                                    ! enable globally
SW2(config)# ip dhcp snooping vlan 1                             ! enable per VLAN (both
are needed)
SW2(config)# no ip dhcp snooping information option              ! stop adding Option 82
SW2(config)# interface g0/0
SW2(config-if)# ip dhcp snooping trust                           ! this port faces the
DHCP server
SW2(config-if)# ip dhcp snooping limit rate 1                    ! messages per second (1
is lab-only; too low for real use)
SW2(config)# errdisable recovery cause dhcp-rate-limit           ! auto re-enable after a
rate violation
SW2# show ip dhcp snooping binding                               ! the binding table
SW2# show errdisable recovery                                    ! recovery causes and
timers
```

Remember for the exam

- Two steps to enable: global `ip dhcp snooping`, then `ip dhcp snooping vlan`.
- Untrusted by default; trust ports toward the server and other switches.
- Server messages on an untrusted port are always dropped.
- Discover/Request are checked by source MAC against chaddr; Release/Decline against the binding table.
- Rate limit violation err-disables the port; the recovery cause is `dhcp-rate-limit`.

Check yourself

1. Which three DHCP messages are always dropped on an untrusted port? *(OFFER, ACK and NAK, because they come from servers.)*
2. A port faces a PC and another faces the DHCP server. How should each be set? *(PC port stays untrusted; the server-facing port gets* `ip dhcp snooping trust`*.)*
3. Why might clients get no address after you enable snooping on a chain of Layer 2 switches? *(Switches add Option 82 and the next switch or the server drops it; use* `no ip dhcp snooping information option`*.)*

Do next

Do the DHCP Snooping lab and read the binding table after each PC gets an address. You need that table for Day 51.

Day 51: Dynamic ARP inspection

Exam domain: 5. Security Fundamentals · **Video:** Dynamic ARP Inspection (33 min) · Lab: Dynamic ARP Inspection

Key ideas

- **ARP recap.** A host broadcasts an ARP request to learn the MAC of a known IP, and the owner sends a unicast reply. An ARP message sits straight in an Ethernet frame with no IP header, so its addresses are the sender MAC/IP and target MAC/IP fields inside the message. A **gratuitous ARP** is a reply sent unasked, broadcast to everyone, so others update their tables.
- **ARP poisoning.** An attacker sends gratuitous ARP (or forged replies) claiming the default gateway's IP at the attacker's MAC. Other hosts update their ARP tables and send outside traffic to the attacker, who can read or alter it and forward it on: a man-in-the-middle attack.
- **Dynamic ARP inspection (DAI)** filters ARP messages on **untrusted** ports. Like DHCP snooping, all ports are untrusted by default, and trusted ports are not inspected. Trust ports to switches and routers, keep host ports untrusted. Cisco recommends trusting every inter-switch link.
- **The core check.** DAI compares the ARP message's sender MAC and sender IP with the **DHCP snooping binding table**. A match is forwarded; no match is dropped. So DAI normally depends on DHCP snooping.
- **Hosts with static IPs** have no binding entry, so their ARP messages would be dropped. Fix this with an **ARP ACL** that maps IP to MAC, applied to the VLAN. ARP ACL detail goes beyond the CCNA.
- **Enabling is one step.** DHCP snooping needs a global command plus a per-VLAN one; DAI only needs the per-VLAN command. Every VLAN you want protected must be listed.
- **Rate limiting.** Unlike DHCP snooping, DAI limits are on by default for untrusted ports at **15 packets per second**, and off on trusted ports. Exceeding the rate err-disables the interface. It limits ARP messages received, not sent. DAI also takes an optional **burst interval**, so a limit can be "X packets per Y seconds"; the interval defaults to 1 second. Recovery cause: `arp-inspection`.
- **Extra validation checks** are optional and run in addition to the binding check; a message must pass all of them.

Option	What it checks
`src-mac`	Ethernet source MAC equals the sender MAC in the ARP message
`dst-mac`	Ethernet destination MAC equals the target MAC in the message (replies)
`ip`	Rejects invalid addresses such as 0.0.0.0, 255.255.255.255 and multicast; sender IP checked in requests and replies, target IP only in replies

- **Gotcha:** each `validate` command overwrites the previous one. List every check you want in a single command, in any order.

Commands

```
SW1(config)# ip arp inspection vlan 1                          ! enable DAI on VLAN 1
(all ports untrusted)
SW1(config-if)# ip arp inspection trust                        ! trust this port
SW1(config-if)# ip arp inspection limit rate 25 burst interval 2  ! 25 packets per 2
seconds (default 15 per 1)
SW1(config)# errdisable recovery cause arp-inspection          ! auto re-enable after
a rate violation
SW1(config)# ip arp inspection validate ip src-mac dst-mac     ! all three checks in
one command
SW1(config)# arp access-list ARP-ACL-1                         ! ARP ACL for hosts
without DHCP
SW1(config-arp-nacl)# permit ip host 192.168.1.100 mac host 0000.1111.2222  ! map one
static host
SW1(config)# ip arp inspection filter ARP-ACL-1 vlan 1         ! apply the ACL to the
VLAN
SW1# show ip arp inspection interfaces                         ! trust state and rate
limits
SW1# show ip arp inspection                                    ! config summary and
drop counters
```

Remember for the exam

- DAI uses the DHCP snooping binding table (or ARP ACLs) to judge sender MAC and IP.
- Untrusted by default; trust links to other network devices.
- Default rate limit is 15 pps on untrusted ports; the burst interval is DAI-only.
- Multiple `validate` options must go in one command.
- DAI stops ARP poisoning, a man-in-the-middle attack.

Check yourself

1. You type `ip arp inspection vlan 1` and change nothing else. What is the trust state of the ports? *(All untrusted.)*
2. A server with a static IP cannot communicate after DAI is enabled. Why, and what fixes it? *(It has no binding table entry; permit it with an ARP ACL applied to the VLAN.)*
3. You enter `validate dst-mac`, then `validate ip`. Which checks are active? *(Only `ip`; the second command replaced the first.)*

Do next

Do the DAI lab: break a static host, then fix it with an ARP ACL. Day 52 returns to network design.

Day 52: LAN architectures

Exam domain: 1. Network Fundamentals · **Video:** LAN Architectures (28 min) · Lab: STP & FHRP Synchronization

Key ideas

The exam topics are two-tier, three-tier, spine-leaf and SOHO designs. There is no configuration today. Design answers are often "it depends", so learn the standard patterns and the roles of each layer.

Topology terms

Term	Meaning
Star	many devices all connect to one central device
Full mesh	every device connects directly to every other device
Partial mesh	some devices connect to each other, but not all
Hybrid	a combination of these in one network

Campus LAN layers

Layer	Role
Access	where end hosts connect (PCs, printers, cameras, APs, phones). Many ports, often PoE. QoS marking happens here as early as possible, along with security features such as port security, DHCP snooping and DAI.
Distribution (also called aggregation)	aggregates the access switches, normally in a redundant pair. Usually the boundary between Layer 2 and Layer 3: access links are Layer 2 (with STP), hosts use SVIs here as default gateway (with an FHRP such as HSRP or VRRP), and routing such as OSPF runs here. Connects to services like the Internet and WAN.
Core	connects many distribution layers in a large LAN. Focus is speed ("fast transport"), so avoid CPU-heavy work such as security or QoS marking and classification. All links are Layer 3, with no STP. Redundancy matters because it is the backbone.

- **Two-tier (collapsed core).** Only access and distribution. The distribution layer takes on the core's job too, so it is sometimes called core-distribution. Distribution switches link to each other with Layer 3 links. Suits small to medium LANs.
- **Three-tier.** Adds the core layer. Without it, the number of links between distribution switches grows fast as you add buildings and the design is hard to scale. Cisco suggests adding a core when a location has more than three distribution layers. Distribution switches connect to the core instead of each other, and the core connects to the Internet routers.
- **Spine-leaf** (also called Clos) is a data center design. Traditional three-tier suits mostly **north-south** traffic (to and from outside the data center). Virtualized, distributed applications create heavy **east-west** traffic between servers, which caused bandwidth bottlenecks and uneven latency. Rules:
  - every leaf connects to every spine
  - leaf switches never connect to other leaves, and spines never connect to other spines
  - servers and other endpoints connect only to leaves

    - Traffic is spread across spines, and any two servers on different leaves are the same number of hops apart (leaf, spine, leaf), so latency is predictable. To grow, add a leaf and connect it to every spine.
- **SOHO** (small office/home office). One device, often called a home or wireless router, acts as router, switch, simple firewall and wireless access point, and sometimes as the modem too. Typically rented from the ISP. A home network with Internet access counts as SOHO.

Remember for the exam

- Access = hosts, QoS marking, port security/DAI/DHCP snooping, PoE.
- Distribution = aggregation and the Layer 2/Layer 3 boundary.
- Core = fast, all Layer 3, no STP, no QoS marking.
- Two-tier is called collapsed core.
- Spine-leaf: full leaf-to-spine connections, nothing leaf-to-leaf or spine-to-spine, hosts on leaves only.
- Spine-leaf addresses east-west traffic; three-tier is built for north-south.

Check yourself

1. In which layer would you expect STP and the default gateway SVIs, and why? *(Distribution, because it is the border between Layer 2 access links and Layer 3.)*
2. Can a leaf switch connect to another leaf switch? *(No. Each leaf connects only to spines, and servers connect only to leaves.)*
3. Why add a core layer to a growing campus? *(To avoid connecting every distribution switch to every other one.)*

Do next

Do the STP & FHRP Synchronization lab, which shows why the STP root and the active gateway should sit on the same distribution switch. Then start Day 53 on WANs.

Day 53: WAN architectures

Exam domain: 1. Network Fundamentals · **Video:** WAN Architectures (38 min) · Lab: GRE Tunnels

Key ideas

The lecture is concept-only, from the customer's point of view. Know the names and purposes.

- **WAN** (wide area network) joins geographically separate LANs, such as offices in different cities. The word usually means a company's private connections, not the Internet itself.
- **Hub and spoke** is the WAN word for a star: one central site (hub) and the others (spokes). Central control is easy, because inter-office traffic can pass a firewall at the hub.
- **Leased line.** A dedicated physical link between two sites, not shared and not the Internet. Traditionally serial with HDLC or PPP, not Ethernet. North American standards are T1, T2, T3 (T1 is 1.544 Mbps); elsewhere E1, E2, E3. They are losing ground to Ethernet WAN links (often fiber) because leased lines cost more, take longer to install and are slower.
- **MPLS** (multiprotocol label switching) is provider infrastructure shared by many customers. Labels keep each customer's traffic separate, creating VPNs. A label sits between the Layer 2 and Layer 3 headers (so "Layer 2.5"), and provider routers forward by label, not destination IP.

Router	Meaning	Runs MPLS?
CE	customer edge, the customer's router	no (not required)
PE	provider edge, connects to CE routers, adds labels	yes
P	provider core router, no connection to customers	yes

- **Layer 3 MPLS VPN:** CE and PE routers peer (for example OSPF), or the customer uses static routes pointing at the PE.
- **Layer 2 MPLS VPN:** the provider network is transparent, like one big switch. CE routers share a subnet and can peer directly with each other.
- Sites can reach the provider by any access type: fiber Ethernet, 4G/5G, cable or a leased line.
- **Internet access options:** DSL (over phone lines, needs a modem), cable (over the cable TV network, needs a cable modem), fiber Ethernet, wireless 4G/5G, or a leased line or MPLS link to the provider. A modem may be a separate box or built into the router.
- **Redundant Internet terms**

Term	Connections
Single homed	1 link to 1 ISP
Dual homed	2 links to the same ISP
Multihomed	1 link to each of 2 ISPs
Dual multihomed	2 links to each of 2 ISPs (most redundancy)

- **Internet VPNs.** The Internet has no built-in security, so traffic is protected with VPNs.

- **Site-to-site VPN (IPsec).** A tunnel between two devices, usually routers or firewalls. The original packet is encrypted, then wrapped in an IPsec header and a new IP header. Hosts at each site send plain traffic to their router. It permanently connects whole sites.
- **Limits of IPsec:** only unicast, so routing protocols using multicast (OSPF) cannot run over it, and a full mesh of tunnels is a lot of configuration.
- **GRE over IPsec.** GRE (generic routing encapsulation) can carry many protocols plus broadcast and multicast, but does not encrypt. Wrapping GRE inside IPsec gives both.
- **DMVPN** (Cisco). Configure tunnels from each spoke to the hub only, and the routers build direct spoke-to-spoke tunnels dynamically. It gives hub-and-spoke simplicity with full-mesh efficiency.
- **Remote-access VPN (TLS).** Client software such as Cisco AnyConnect on a laptop or phone builds a tunnel to a firewall or router. It serves one device, on demand, for users on untrusted networks. TLS is the security behind HTTPS.

Commands

The captions cover only the lecture; this GRE basic is the usual IOS form for the lab. Check it against the lab video.

```
R1(config)# interface tunnel 0                                  ! create the tunnel
interface
R1(config-if)# ip address 10.0.0.1 255.255.255.252             ! tunnel's own address
R1(config-if)# tunnel source g0/0                               ! local physical
interface
R1(config-if)# tunnel destination 203.0.113.2                   ! peer's reachable
address
R1# show interfaces tunnel 0                                    ! tunnel state and
endpoints
```

Remember for the exam

- CE does not run MPLS; PE and P do. MPLS forwards on labels.
- Layer 2 MPLS VPN: CE routers peer with each other. Layer 3: CE peers with PE.
- Single homed, dual homed, multihomed, dual multihomed.
- Site-to-site VPN = IPsec between sites; remote-access VPN = TLS from a device.
- IPsec alone cannot carry multicast; GRE over IPsec solves that. DMVPN solves tunnel-mesh configuration.

Check yourself

1. Which MPLS router type does not need to understand MPLS? *(The CE router.)*
2. A company has one link to ISP A and one to ISP B. What is this called? *(Multihomed.)*
3. Why can't OSPF run across a plain IPsec tunnel, and what is the fix? *(IPsec supports only unicast; use GRE over IPsec.)*

Do next

Do the GRE Tunnels lab.

Day 54: Virtualization, cloud, containers and VRF

Exam domain: 1. Network Fundamentals · **Video:** Virtualization & Cloud (39 min) · Containers (14 min) · VRF (18 min) · Lab: Oracle VirtualBox

Key ideas

Virtualization

- Without virtualization one physical server runs one operating system, so each app tends to get its own box and most of its CPU, RAM and storage sits idle. A **hypervisor** (VMM) splits one server into several **virtual machines (VMs)**, each with its own OS, and hands out the hardware resources.
- **Type 1** hypervisors run directly on the hardware (bare-metal or native), for example VMware ESXi and Microsoft Hyper-V. They are what data centers use. **Type 2** hypervisors run as a program on a host OS (hosted), for example VMware Workstation and Oracle VirtualBox, and suit a laptop or lab. The OS inside a VM is the *guest*; the OS under a Type 2 hypervisor is the *host*.
- Benefits: partitioning, isolation (a fault in one VM stays there), encapsulation (a VM can be saved, copied and moved like a file), hardware independence, and lower capital and operating costs.
- VMs reach each other and the outside through a **virtual switch** on the hypervisor. It has access and trunk ports and uses VLANs like a physical switch, and it connects to the server's physical NICs.

Cloud (NIST SP 800-145)

Five essential characteristics	In short
On-demand self-service	The customer provisions resources alone, through a portal
Broad network access	Reachable over the network from many device types
Resource pooling	One shared pool serves many tenants
Rapid elasticity	Scale out and in quickly; capacity looks unlimited
Measured service	Usage is metered, so you pay for what you use

Service model	The provider runs	The customer controls	Example
SaaS	Everything, up to the app	Only light app settings	Microsoft 365
PaaS	Infrastructure, OS and platform tools	The apps they deploy	AWS Lambda
IaaS	Data center, network, servers, storage	OS, storage and apps	Amazon EC2

- **Deployment models:** private (one organization; can be run by a third party and can be on or off premises), community (a group of organizations with shared concerns; the least common), public (open to everyone; AWS, Azure, GCP), and hybrid (two or more of the others joined together).
- Cloud shifts spending from CapEx to OpEx. Reach a public cloud over the Internet (cheap, add a VPN for security) or a private WAN such as MPLS VPN; use redundant links.

Containers

- A container packages one app with its binaries and libraries. Containers run on a **container engine** (Docker is the best known) on a host OS, on the hardware. There is no OS inside each container, and

that is the key difference from a VM.

- Containers start in milliseconds, take tens of MB instead of tens of GB, and use less CPU and RAM. VMs are better isolated, because a host OS crash takes down every container on it.
- An **orchestrator** automates deploying, managing and scaling many containers, which microservice systems need. Kubernetes is the most popular; Docker Swarm is another.

VRF

- Virtual routing and forwarding splits one router into several virtual routers, each with its own routing table. It is VLANs for routers. Only Layer 3 interfaces (router ports, SVIs, routed ports) can join a VRF.
- Traffic in one VRF is not forwarded out of an interface in another. Customers can reuse the same subnets, which a single routing table would not allow. Service providers use it to carry many customers on one router. The lecture covers VRF-lite (no MPLS). Configuration is not an exam topic.

Commands

```
R1(config)# ip vrf customer1                        ! create a VRF
R1(config-if)# ip vrf forwarding customer1          ! put the interface in the VRF (its
IP address is removed)
R1(config-if)# ip address 192.168.1.1 255.255.255.252  ! set the address again, after
the VRF command
R1# show ip vrf                                     ! list VRFs and their interfaces
R1# show ip route vrf customer1                     ! that VRF's routing table
R1# ping vrf customer1 192.168.1.2                  ! ping from inside the VRF
```

`show ip route` alone shows only the global table, which is empty if every interface is in a VRF.

Remember for the exam

- Type 1 = bare-metal/native, data center. Type 2 = hosted, runs on a host OS.
- Five characteristics, three service models, four deployment models. "Infinite resources" is not a characteristic.
- Private cloud can be on premises; cloud does not always mean off premises.
- VMs carry an OS each; containers share the host OS.
- VRFs divide routers into routing tables; VLANs divide switches into broadcast domains.

Check yourself

1. Which hypervisor type runs directly on hardware? *(Type 1.)*
2. A company rents virtual servers and installs its own OS and apps on them. Which service model is that? *(IaaS.)*
3. Why can two interfaces on one router use 192.168.1.0/30 when they are in different VRFs? *(Each VRF has its own routing table.)*

Do next

Do the VirtualBox lab, then download the Day 54 Anki cards. Learn the cloud tables above.

Day 55: Wireless fundamentals

Exam domain: 1. Network Fundamentals · **Video:** Wireless Fundamentals (36 min)

Key ideas

- Wireless LANs follow **IEEE 802.11**, as wired Ethernet follows 802.3. "Wi-Fi" is the Wi-Fi Alliance's trademark; it tests and certifies equipment for compliance and interoperability.
- Wireless behaves like a hub: every device in range hears every frame. So wireless traffic must be **encrypted even inside the LAN**, and devices share a half-duplex medium. **CSMA/CA** (collision avoidance) makes a device listen first and wait a random time if the channel is busy. Wired Ethernet's CSMA/CD detects collisions after they happen.
- **Signal problems:** absorption (a wall turns signal into heat), reflection (metal bounces it), refraction (bending in glass or water), diffraction (waves go around an obstacle and leave a blind spot), and scattering (dust or rough surfaces). Interference from neighbors on the same channel also hurts.
- **Frequency** is cycles per second (hertz); **amplitude** is the strength of the wave; the period is the time of one cycle (4 Hz means 0.25 s).

Bands and channels

- Two main bands: **2.4 GHz** (2.4 to 2.4835 GHz) and **5 GHz**. 2.4 GHz reaches farther and passes through walls better but is more crowded. Wi-Fi 6 (802.11ax) adds a **6 GHz** band.
- Each band is split into channels. 2.4 GHz channels are 22 MHz wide and overlap. Neighboring APs should use **1, 6 and 11** (in North America), which do not overlap. 5 GHz channels do not overlap, so planning is easier.
- Place APs in a honeycomb so coverage overlaps but channels do not.

Standard	Wi-Fi name	Year	Band	Max theoretical rate
802.11	-	1997	2.4 GHz	2 Mbps
802.11b	-	1999	2.4 GHz	11 Mbps
802.11a	-	1999	5 GHz	54 Mbps
802.11g	-	2003	2.4 GHz	54 Mbps
802.11n	Wi-Fi 4	2009	2.4 and 5 GHz	600 Mbps
802.11ac	Wi-Fi 5	2013	5 GHz	about 6.9 Gbps
802.11ax	Wi-Fi 6	2019	2.4 and 5 GHz (6 GHz with Wi-Fi 6E)	about 9.6 Gbps

There is no official Wi-Fi 1, 2 or 3. Real speeds are well below these maximums.

Service sets

An **SSID** is the human-readable network name. It need not be unique, but it should be.

Term	Meaning
IBSS	Devices connect directly with no AP (ad hoc, such as AirDrop)

Term	Meaning
BSS	Clients associate with one AP and talk through it, never directly
BSSID	The MAC address of the AP's radio; unique per BSS
BSA	The physical area where the AP's signal is usable
ESS	Several BSSs joined by a wired network; same SSID, different BSSIDs and channels
MBSS	Mesh: APs with two radios, one for clients and one as wireless backhaul

- In an ESS, clients **roam** between APs without reconnecting. Overlap coverage by about 10 to 15 percent.
- In a mesh, the AP wired to the network is the **RAP** (root access point); the others are **MAPs**.
- The wired network behind the APs is the **distribution system**. Each SSID maps to a VLAN. An AP with several SSIDs uses a trunk, and each SSID gets its own BSSID (usually the last digit incremented).

Other AP modes

- **Repeater:** retransmits an AP's signal to extend range. With one radio it must stay on the same channel and halves throughput; two radios avoid that.
- **Workgroup bridge (WGB):** acts as a wireless client of an AP so wired devices can join. Cisco's WGB bridges several wired clients; the 802.11 universal WGB (uWGB) bridges one.
- **Outdoor bridge:** joins sites over long distances with directional antennas, point-to-point or point-to-multipoint.

Remember for the exam

- 2.4 GHz non-overlapping channels: 1, 6, 11.
- BSS = one AP; ESS = several BSSs with one SSID; IBSS = no AP.
- BSSID is a MAC address; SSID is a name.
- CSMA/CA avoids collisions; it is used on wireless.
- Know the standards table: name, band, speed.

Check yourself

1. Which standard is Wi-Fi 5, and which band does it use? *(802.11ac, 5 GHz.)*
2. Two APs in an ESS share which value, and differ in which? *(Same SSID; different BSSIDs.)*
3. Why must wireless traffic be encrypted inside the LAN? *(Anyone in range can receive the frames.)*

Do next

No lab for this day. Download the Day 55 Anki cards and memorize the standards table before Day 56.

Day 56: Wireless architectures

Exam domain: 2. Network Access · **Video:** Wireless Architectures (38 min)

Key ideas

802.11 messages

- A station must be **authenticated and associated** before it can send traffic through an AP. It finds the AP by **active scanning** (probe request, probe response) or **passive scanning** (listening for the AP's periodic beacons). Then come authentication, then an association request and response.
- Three frame types: **management** (beacon, probe, authentication, association), **control** (RTS, CTS, ACK; they manage access to the medium) and **data**.
- The 802.11 header can carry up to four addresses (destination, source, receiver, transmitter), plus frame control, duration/ID, sequence control and QoS fields. Not every frame uses all of them. The frame ends with an FCS.

Three AP architectures

	Autonomous	Lightweight	Cloud-based
Managed by	Each AP on its own	A WLC	A cloud dashboard (Cisco Meraki)
Client data path	Straight into the wired network	Tunneled to the WLC first	Straight into the wired network
Switch port	Trunk	Access	Trunk
Scales to large networks	No	Yes	Yes

- **Autonomous APs** are configured one at a time (console, SSH/Telnet or web GUI), with RF settings, security and QoS set per AP, and no central view. Each needs a management IP, usually in a separate management VLAN, which is why the port is a trunk. VLANs end up stretched across the network, which means big broadcast domains, links blocked by spanning tree, and tedious VLAN changes.
- **Lightweight APs** use **split-MAC**: the AP handles real-time work (sending and receiving RF, encryption, beacons, probes), while the WLC handles RF management, client authentication, association, roaming, and security and QoS policy. The two authenticate each other with X.509 certificates.
- **CAPWAP** carries traffic between AP and WLC in two tunnels:

Tunnel	UDP port	Carries	Encrypted
Control	5246	Configuration and management	Yes, by default
Data	5247	All client traffic	No by default; DTLS can be enabled

- Because every client frame goes to the WLC, even traffic between two clients on the same AP, the AP only needs an access port, and the WLC connects with a trunk. The WLC becomes the border between wired and wireless.

- **Split-MAC benefits:** scalability, dynamic channel assignment, automatic transmit power, self-healing coverage (neighbors raise power if an AP fails), seamless roaming, client load balancing, and consistent security and QoS.
- **Cloud-based** APs are autonomous APs managed centrally from the cloud. Only management and control information goes to the cloud; user data does not.

Lightweight AP modes

Mode	Purpose
Local	Default; serves clients with one or more BSSs
FlexConnect	Serves clients, and switches traffic locally if the WLC tunnels go down
Sniffer	Captures 802.11 frames and sends them to a tool such as Wireshark
Monitor	Scans for rogue devices; no client service
Rogue detector	Listens on the wired side only, using the WLC's list of suspected rogue MAC addresses
SE-Connect	Spectrum analysis of all channels to find interference
Bridge/mesh	Links sites, even over long distances
Flex+bridge	Bridge/mesh with FlexConnect

WLC deployment models (for lightweight APs only)

Model	Where the WLC is	About how many APs
Unified	Dedicated hardware appliance	6,000
Cloud-based	VM on a server, often a private cloud	3,000
Embedded	Inside a switch	200
Mobility Express	Inside an AP	100

"Cloud-based WLC" is not the same as cloud-based APs: it only says where the controller runs.

Remember for the exam

- Probe, beacon, authentication and association are management frames; RTS, CTS and ACK are control frames.
- CAPWAP: control UDP 5246, data UDP 5247. Split-MAC means AP and WLC share the work.
- Autonomous and cloud APs use trunks; lightweight APs use access ports.
- Local is the default mode; FlexConnect survives a lost WLC.

Check yourself

1. Which frame type is a beacon? *(Management.)*
2. Two clients on the same lightweight AP talk to each other. Where does the traffic go? *(Through the CAPWAP data tunnel to the WLC, then back.)*
3. Which WLC deployment suits a small branch with no extra hardware? *(Mobility Express, with the WLC inside an AP.)*

Do next

No lab for this day. Download the Day 56 Anki cards and learn the port numbers and deployment table.

Day 57: Wireless security

Exam domain: 5. Security Fundamentals · **Video:** Wireless Security (34 min)

Key ideas

Three jobs: **authentication** (who is connecting), **encryption** (only AP and client can read the traffic) and **integrity** (frames were not changed in transit). Wireless needs all three because anyone in range can receive the frames.

- Clients should also verify the AP, to avoid a malicious AP that mounts man-in-the-middle attacks. A guest SSID can use lighter authentication and reach only the Internet.
- Each client has its own unicast encryption key. The AP also keeps a **group key** for traffic sent to all clients.
- A **MIC (message integrity check)** is computed by the sender, sent with the message, and recomputed by the receiver. If the two differ, the frame is dropped.

Authentication methods

Method	Notes
Open	The AP accepts anyone. Not secure alone, but used with web login pages (public Wi-Fi)
WEP	RC4, shared key of 40 or 104 bits plus a 24-bit IV (64 or 128 bits total). The AP sends a challenge and checks that the client encrypts it with the same key. Easily cracked
LEAP	Cisco; username and password, with both sides sending challenges and dynamic WEP keys. Vulnerable, not for use
EAP-FAST	Cisco; a PAC (shared credential) builds a TLS tunnel, then the client is authenticated inside it
PEAP	The server presents a certificate to build a TLS tunnel; the client authenticates inside it, for example with MS-CHAP
EAP-TLS	Certificates on the server and on every client. Most secure, hardest to deploy

- **EAP** is a framework; LEAP, EAP-FAST, PEAP and EAP-TLS are EAP methods. It works with **802.1X**, port-based access control with three roles: **supplicant** (the client), **authenticator** (the AP, or the WLC in split-MAC) and **authentication server** (usually RADIUS).
- With 802.1X the client first associates with open authentication, but may send only EAP traffic until the server permits access.

Encryption and integrity

Protocol	Based on	Integrity	Used in
WEP	RC4	none worth using	original 802.11
TKIP	WEP, made stronger	MIC with sender MAC, timestamp and sequence number; per-frame key; 48-bit IV	WPA

Protocol	Based on	Integrity	Used in
CCMP	AES counter mode	CBC-MAC	WPA2
GCMP	AES counter mode	GMAC	WPA3

- TKIP was a stopgap that ran on WEP-era hardware. CCMP needs hardware support. GCMP is the most secure and efficient. Order of development: WEP, TKIP, CCMP, GCMP.

WPA certifications (Wi-Fi Alliance)

- Each of WPA, WPA2 and WPA3 supports two modes: **Personal** (a pre-shared key, PSK, for homes and small offices) and **Enterprise** (802.1X with an authentication server and any EAP method).
- In Personal mode the PSK is not sent over the air. A four-way handshake uses it to generate encryption keys.

Certification	Encryption and integrity
WPA	TKIP
WPA2	CCMP
WPA3 (2018)	GCMP

WPA3 also adds **PMF** (protected management frames, optional in WPA2 but mandatory in WPA3), **SAE** (simultaneous authentication of equals, which protects the Personal-mode handshake) and forward secrecy (captured traffic cannot be decrypted later).

Remember for the exam

- Pairings: WPA-TKIP, WPA2-CCMP, WPA3-GCMP.
- Authentication proves identity, encryption hides data, a MIC checks integrity.
- Supplicant, authenticator, authentication server.
- PEAP needs a certificate on the server only; EAP-TLS needs one on both sides.
- Personal means PSK; Enterprise means 802.1X.

Check yourself

1. Which protocol uses AES counter mode with CBC-MAC? *(CCMP, used in WPA2.)*
2. In 802.1X on a wireless LAN, what role does the laptop play? *(Supplicant.)*
3. Which WPA3 feature protects the handshake in Personal mode? *(SAE.)*

Do next

No lab for this day. Download the Day 57 Anki cards, and learn the pairing tables before Day 58.

Day 58: Wireless configuration

Exam domain: 2. Network Access · **Video:** Wireless Configuration (47 min) · Lab: Wireless LANs

Key ideas

The exam expects you to configure a WLAN on a WLC through the **GUI**, using **WPA2 with PSK**. The GUI is reached over HTTP or HTTPS, not the console, so the WLC needs a management IP first.

The lecture topology: one switch, one WLC, two lightweight APs. Management is VLAN 10 (192.168.1.0/24), and two WLANs map to VLAN 100 `internal` (10.0.0.0/24) and VLAN 200 `guest` (10.1.0.0/24).

- APs plug into **access ports** in the management VLAN (they get power over PoE). Only the WLC link is a **trunk**.
- The WLC connects to the switch with a **LAG**. WLCs support only a **static** LAG, so the switch uses `channel-group 1 mode on`, never LACP or PAgP.
- APs get an address from DHCP. **DHCP option 43** tells them the WLC's IP address. It matters when the WLC is somewhere the APs' CAPWAP discovery broadcast cannot reach.
- First boot shows a CLI setup wizard on the console: system name, admin login, LAG, management interface (IP, VLAN ID, gateway, DHCP server), the first WLAN, country code, auto RF and NTP. The AP's regulatory domain (a letter in its model name) must match the country code or it cannot join.

Commands

On the switch in the lecture:

```
SW1(config)# interface range f0/1 - 2
SW1(config-if-range)# channel-group 1 mode on     ! static EtherChannel to the WLC (a
LAG)
SW1(config)# interface port-channel 1
SW1(config-if)# switchport mode trunk             ! trunk toward the WLC
SW1(config-if)# switchport trunk allowed vlan 10,100,200  ! management and WLAN VLANs
SW1(config)# interface range f0/7 - 8
SW1(config-if-range)# switchport mode access      ! APs use access ports
SW1(config-if-range)# switchport access vlan 10   ! management VLAN
SW1(config)# ip dhcp pool vlan10
SW1(dhcp-config)# network 192.168.1.0 255.255.255.0
SW1(dhcp-config)# default-router 192.168.1.1
SW1(dhcp-config)# option 43 ip 192.168.1.100      ! tells APs where the WLC is
SW1(config)# ntp master                           ! the switch serves time to the WLC
```

WLC ports versus interfaces

On a WLC "port" means a physical port and "interface" means a logical one.

Port	Use
Service port	Out-of-band management; connects to an access port (one VLAN)

Port	Use
Distribution system ports	Normal data to the wired network; usually trunks and can form a LAG
Console	RJ45 or USB
Redundancy port	Links two WLCs as a high-availability (HA) pair

Interface	Use
Management	Telnet/SSH, HTTP/HTTPS, RADIUS, syslog; CAPWAP tunnels end here
Redundancy management	Reaches the standby WLC
Virtual	Talks to clients for DHCP relay and web authentication
Service port interface	Bound to the service port
Dynamic	Maps a WLAN to a VLAN (like an SVI); one per WLAN VLAN

Creating a WLAN in the GUI

1. Controller > Interfaces > New: create a dynamic interface, name it, give it the VLAN ID, then enter its IP, mask, gateway and DHCP server.
2. WLANs > Create New: choose the type (regular WLAN), a **profile name**, an **SSID** and a **WLAN ID**. Profile name and SSID can differ.
3. General tab: set status to enabled and pick the dynamic interface. Leaving it on the management interface is a mistake.
4. Security tab, Layer 2: choose **WPA+WPA2**, untick 802.1X, tick **PSK**, choose ASCII or hex, and enter a key of at least 8 characters. Layer 3 offers web policies (web authentication needs a login; web pass-through needs only agreement). No RADIUS server is needed for PSK.
5. QoS tab: **Platinum** (voice), **Gold** (video), **Silver** (best effort, the default), **Bronze** (background).

Other tabs: the dashboard lists APs and clients, Wireless > AP > mode sets local, FlexConnect and the rest, and Management shows which protocols are on (Telnet off, management over wireless off by default). A **CPU ACL** restricts traffic sent to the WLC itself, not traffic passing through.

Remember for the exam

- The GUI is reached by HTTP/HTTPS, not the console.
- WLC LAG is static: `mode on`.
- Dynamic interface = WLAN to VLAN. Backup (redundancy) port = HA pair.
- WPA2-PSK in the GUI: Security tab, Layer 2, PSK.
- QoS tiers: Platinum voice, Gold video, Silver best effort, Bronze background.

Check yourself

1. Which WLC interface type ties a WLAN to a VLAN? *(Dynamic interface.)*
2. Why does the switch use `channel-group 1 mode on` toward the WLC? *(WLCs support only static LAG, not LACP or PAgP.)*

3. Which QoS profile suits video? *(Gold.)*

Do next

Do the Packet Tracer lab, then download the Day 58 Anki cards. Practise building a WLAN with PSK until it is automatic.

Day 59: Network automation, SDN and AI

Exam domain: 6. Automation and Programmability · **Video:** Intro to Network Automation (33 min) · AI & Machine Learning (42 min)

Key ideas

Why automate

- The traditional model is one device at a time: SSH (or Telnet, console, a GUI) to each router or switch and type the change. Across hundreds of devices that invites typos, takes a long time, and lets configurations drift away from the company standard.
- Automation reduces human error, scales better, enforces policy compliance (standard configs, correct software versions) and cuts OpEx because each task takes fewer engineer-hours. Tools include SDN controllers, Ansible, Puppet and Python scripts.
- The exam asks you to explain and describe these topics. You do not have to write automation yourself.

The three logical planes

Plane	What it does	Examples
Data (forwarding)	Moves user traffic through the device	Routing and MAC table lookups, re-encapsulation, VLAN tagging, NAT, ACL permit or drop, port security
Control	Builds the tables and decisions the data plane uses; overhead work	OSPF, ARP, STP, building the routing, ARP and MAC tables
Management	Lets people manage and monitor the device; does not affect forwarding	SSH, Telnet, Syslog, SNMP, NTP

- The data plane runs on a specialized chip (an **ASIC**), and the tables it reads sit in fast **TCAM** memory. Control and management traffic addressed to the device is handled by the CPU.
- In a traditional network each device has its own control plane, so control is **distributed**.

SDN

- SDN centralizes the control plane (fully or partly, depending on the product) in a **controller**, an application on a server. The devices keep forwarding. It is also called controller-based networking, and you have already met the idea in wireless LAN controllers.
- The **southbound interface (SBI)** links the controller to the network devices. It is a software interface (a protocol plus an API), not a physical port. Examples: OpenFlow, Cisco OpFlex, Cisco onePK, NETCONF.
- The **northbound interface (NBI)** links apps to the controller. It is normally a **REST API**. An app sends a request and gets structured data (JSON or XML) back.
- Because the controller holds a network-wide view in a program-friendly format, you avoid screen-scraping `show` output with scripts. Traditional devices can also be automated, for example with Python over SSH.

AI and machine learning (v1.1 addition)

- **AI** is computers simulating intelligence. **Machine learning (ML)** is the part of AI where a model learns patterns from data instead of following hard-coded rules. Not all AI uses ML.

ML type	Training	Notes
Supervised	Labeled data	Accurate, but labels are costly and it only outputs known labels
Unsupervised	Unlabeled data; finds clusters	No labeling cost, but a human interprets the groups and accuracy can be lower
Reinforcement	An agent acts in an environment and gets rewards or penalties	Games, robots, self-driving; heavy on compute, reward design matters
Deep learning	Multi-layer artificial neural networks	Strong on images, audio and text; resource hungry and hard to interpret

- Deep learning networks can be trained with supervised, unsupervised or reinforcement methods. Semi-supervised learning mixes labeled and unlabeled data.
- **Predictive AI** uses historical data to forecast outcomes. **Generative AI** learns patterns and creates new content (text, images, audio, video); large language models are an example. Both are strongest with deep learning, but deep learning is not required.
- Predictive uses in networking: traffic forecasting, security anomaly detection, predictive maintenance. Generative uses: documentation, configuration drafts, design suggestions, troubleshooting help, script generation. Generative output can be wrong (hallucination), so verify it.
- Catalyst Center AI features (background, not an exam requirement): AI Network Analytics (learns a baseline of normal), Machine Reasoning Engine (root-cause analysis and fixes), AI Endpoint Analytics (classifies devices), AI-enhanced RRM (tunes Wi-Fi radios).

Remember for the exam

- Data plane forwards, control plane builds tables, management plane manages the device. NTP and SNMP are management; OSPF and STP are control.
- SDN centralizes the control plane. Calculating routes moves to the controller; forwarding stays on the devices.
- SBI = controller to devices. NBI = apps to controller, usually REST.
- Labeled data = supervised. Unlabeled = unsupervised. Reward and penalty = reinforcement. Neural network layers = deep learning.
- Predictive forecasts from past data; generative creates new content.

Check yourself

1. Which plane does a device's SSH server belong to? *(Management.)*
2. An app asks the SDN controller for a list of devices. Which interface does it use? *(The northbound interface, typically a REST API.)*
3. A model groups unlabeled photos into clusters on its own. Which ML type is this? *(Unsupervised learning.)*

Do next

No lab for this day. Download the Day 59 Anki cards and learn the planes table and the four ML types.

Day 60: JSON, XML and YAML

Exam domain: 6. Automation and Programmability · **Video:** JSON, XML, & YAML (29 min)

Key ideas

- **Data serialization** turns data into a standard text format that can be stored in a file or sent over a network, then rebuilt by another program. Two programs written in different languages (say an app and an SDN controller) store variables differently, so they swap data in a shared format. The exam topic names JSON specifically: you must be able to **interpret JSON-encoded data**.
- A **variable** is a named container holding a value, such as an interface name or a status. In these formats, data is written as key-value pairs.

JSON (JavaScript Object Notation)

- Defined in RFC 8259, language-independent, human-readable, and widely used by REST APIs. **Whitespace is insignificant**, so indentation and line breaks are only for human eyes.
- Four primitive types and two structured types:

Type	How it looks	Notes
String	`"GigabitEthernet1/1"`	Always in double quotes; `"5"` and `"true"` are strings
Number	`1000`	No quotes
Boolean	`true` or `false`	Lower case, no quotes
Null	`null`	No value; lower case, no quotes
Object	`{ ... }`	Unordered key-value pairs (also called a dictionary)
Array	`[ ... ]`	Ordered values separated by commas; types can be mixed

- **Object rules:** curly brackets; each key must be a string in double quotes; a colon separates key from value; a comma separates pairs; **no comma after the last pair**. A value can be any type, including another object (a nested object) or an array.
- **Array rules:** square brackets; values separated by commas, no key names.

```
{
  "device": {
    "name": "R1",
    "vendor": "Cisco",
    "is_router": true,
    "uplink_speed": 1000,
    "backup": null
  },
  "interfaces": ["Gi0/0", "Gi0/1"]
}
```

Here `device` holds an object, `uplink_speed` is a number, `is_router` is a boolean, `backup` is null, and `interfaces` is an array of two strings.

Spotting invalid JSON

- A comma after the last pair or value.
- A comma where a colon belongs between key and value, or a colon between array values.
- A missing comma between pairs or array values.
- An opening `{` or `[` with no matching closer.
- Keys or text values without double quotes.

XML

- Began as a markup language and is now used for data serialization too. Whitespace is insignificant. Each value sits between an opening and closing tag, like HTML, for example `<interface>GigabitEthernet0/0</interface>`. Also used by REST APIs. On IOS, adding `| format` to some show commands displays the output as XML.

YAML

- Stands for "YAML Ain't Markup Language". Very human-readable and used by Ansible. **Whitespace (indentation) is significant.** A file starts with three hyphens, a single hyphen marks a list item, and pairs are written `key: value`.

	JSON	XML	YAML
Key-value form	`"key": value`	`<key>value</key>`	`key: value`
Whitespace matters	No	No	Yes
Brackets	`{}` and `[]`	Tags	Indentation and hyphens
Typical use	REST APIs	REST APIs	Ansible

Remember for the exam

- Only JSON has to be read in detail; for XML and YAML know the characteristics above.
- Keys are always strings. Values may be any of the six types.
- Quotes decide the type: `5` is a number, `"5"` is a string; `true` is a boolean, `"true"` is a string.
- `{}` means object, `[]` means array. An array can contain objects.
- Whitespace is significant only in YAML.

Check yourself

1. Which of the three formats treats indentation as meaningful? *(YAML.)*
2. What data type is the value in `"is_up": true`? *(A boolean, because it has no quotes.)*
3. Why is `{"a": 1, "b": 2,}` invalid JSON? *(A comma follows the last key-value pair.)*

Do next

No lab for this day. Read as many JSON samples as you can and check brackets, commas and quotes. Download the Day 60 Anki cards.

Day 61: REST APIs and API authentication

Exam domain: 6. Automation and Programmability · **Video:** REST APIs (32 min) · REST API Authentication (29 min)

Key ideas

REST APIs

- An **API** lets two programs exchange data. In SDN, REST APIs are the usual **northbound interface** (app to controller). NETCONF and RESTCONF are examples used toward devices; they are beyond this lesson.
- **REST** (representational state transfer) is not a specific API. It is a set of rules for building one, and REST APIs are also called RESTful. They normally run over HTTP, or HTTPS, which is HTTP with TLS encryption. REST itself does not require HTTP, but assume it for the exam.
- **CRUD** maps to HTTP verbs:

CRUD action	HTTP verb
Create	POST
Read	GET
Update	PUT or PATCH
Delete	DELETE

- An HTTP request carries a verb, a **URI** naming the resource, optional headers (for example `Accept`, which says which data formats the client can take) and sometimes a body. A URI has a **scheme** (the protocol, such as `https`), an **authority** (the host to contact) and a **path** (the resource).
- **Status code classes** (first digit):

Class	Meaning	Examples
1xx	Informational	102 Processing
2xx	Successful	200 OK, 201 Created
3xx	Redirection	301 Moved Permanently
4xx	Client error	401 Unauthorized (client must authenticate), 404 Not Found
5xx	Server error	500 Internal Server Error

- **REST constraints:** uniform interface, client-server, stateless, cacheable (the server marks which responses may be cached), layered system, and optional code on demand. Three to understand well:
  - **Client-server:** client and server evolve independently as long as the interface keeps working.
  - **Stateless:** every request stands alone and the server keeps no memory of earlier ones, so credentials go with each request. HTTP and REST are stateless even though they ride on stateful TCP.
  - **Cacheable:** caching must be supported, but not every resource has to be cacheable.

Cisco DNA Center (now Catalyst Center) example

1. A `POST` with basic authentication to the controller's token URL returns a `200 OK` and a token in the JSON body.
2. A `GET` to the device-inventory URL, with the token in an `X-Auth-Token` header, returns `200 OK` and a JSON list of the devices the controller manages (family, role, hostname, platform and so on).

REST API authentication

Authentication proves who is calling. Some APIs are open, but most are not. It also lets providers track usage for analytics and billing.

Method	How it works	Strengths and weaknesses
Basic	`username:password`, Base64-encoded, in the `Authorization` header of every request	Simple. Base64 is encoding, not encryption, so it is trivial to decode. Credentials travel on every call
Bearer (token)	Client first gets a token from an authorization server, then sends `Authorization: Bearer <token>`	No repeated password. Tokens expire. Whoever holds the token ("bearer") can use it until it expires
API key	A static key from the provider, sent in the `Authorization` header (a URL parameter or cookie is possible but not recommended; URLs get logged)	Easy, good for tracking each customer. Never expires, so a stolen key works until manually revoked
OAuth 2.0	Delegated access: a third-party app gets a limited, expiring access token without ever seeing the user's password	"Log in with Google" is the example. Defined in RFC 6749

- Every method should run over **HTTPS**.
- **OAuth 2.0 parties:** resource owner (the account holder), client app (the third party), authorization server (issues tokens after the owner approves), resource server (hosts the data). Flow: the app asks the owner for authorization, the owner grants it, the app trades the grant for an access token, then presents the token to the resource server, which validates it and returns the data. A **refresh token** lets the app get a new access token without making the user log in again.

Remember for the exam

- POST create, GET read, PUT/PATCH update, DELETE delete.
- 2xx success, 4xx client's fault, 5xx server's fault. 404 means the resource was not found.
- REST is stateless. "Stateful" is never a REST constraint.
- Base64 is not encryption. API keys do not expire on their own; bearer and OAuth tokens do.
- Bearer and OAuth 2.0 involve an authorization server that issues tokens; basic and API key do not.

Check yourself

1. Which HTTP verb matches the CRUD "create" action? *(POST.)*
2. A client receives a response code starting with 5. Whose problem is it? *(The server's; 5xx is a server error.)*

3. Which authentication method lets a scheduling app read your calendar without ever receiving your password? *(OAuth 2.0.)*

Do next

No lab on the course page. The DevNet Catalyst Center sandbox with Postman is optional hands-on practice. Download the Day 61 Anki cards and memorize the verb table, status classes and four authentication methods.

Day 62: Software-defined networking and Cisco SD-Access

Exam domain: 6. Automation and Programmability · **Video:** Software-Defined Networking (28 min)

Key ideas

SDN architecture, in three layers

Layer	Contains
Application	Scripts and apps that tell the controller what behavior you want (they use the northbound interface)
Control	The SDN controller, which holds the centralized control plane
Infrastructure	The switches and routers that forward traffic (reached through the southbound interface)

- These are SDN layers, not OSI layers. A traditional network has a distributed control plane, with every device running its own protocols such as OSPF. With SDN the devices report to the controller, which calculates routes for the whole network. How much is centralized depends on the product.

Cisco's SDN products

- **SD-Access** automates campus LANs (wired and wireless). Its controller is **Cisco DNA Center**, now named **Catalyst Center**; the lecture uses the older name.
- **ACI** automates data center networks (it uses spine-leaf). **SD-WAN** automates WANs.

Underlay, overlay and fabric

Term	Meaning in SD-Access
Underlay	The physical network of devices and links that provides IP connectivity
Overlay	The virtual network of tunnels built on top of the underlay
Fabric	The underlay and overlay together

- **Switch roles:** edge node (connects end hosts, like an access switch), border node (connects to outside the SD-Access domain, such as a WAN router), control node (runs the control plane). There is no "management node" role.
- **Overlay protocols: LISP** is the control plane. It keeps a DNS-like mapping of **EIDs** (endpoint identifiers, the hosts) to **RLOCs** (routing locators, the edge switch that reaches each host) instead of relying on a normal routing table. **VXLAN** is the data plane: it builds the tunnels that carry traffic between edge switches. **Cisco TrustSec** provides policy control such as QoS and security policy; only the name is needed.
- Flow in short: an edge switch registers its hosts with the control node. When another edge switch needs a destination, it asks the control node, then sends the traffic through a VXLAN tunnel to the switch the control node named.

Underlay design when DNA Center builds it (greenfield)

- A **greenfield** deployment is a brand-new network built for SD-Access. A **brownfield** deployment adds SD-Access to an existing network, and DNA Center does not reconfigure that underlay, because it could disrupt a working production network.
- In a greenfield underlay all switches are Layer 3 and run IS-IS, and every link between switches is a routed port. So there are no Layer 2 loops, spanning tree is not needed, and neither is an FHRP such as HSRP. Edge switches are the hosts' default gateways, which is called a **routed access layer**.

DNA Center / Catalyst Center roles

- It is the SD-Access controller. It can also be a management platform for a traditional (non-SD-Access) network, with central monitoring, analysis and configuration. It is software installed on Cisco UCS server hardware.
- Northbound it offers a **REST API** (and a GUI). Southbound it supports NETCONF and RESTCONF, plus traditional Telnet, SSH and SNMP.
- It enables **intent-based networking (IBN)**: the engineer states the desired behavior, and the controller turns it into device configuration. Example: in a group-based policy grid you mark which source group may reach which destination group, instead of writing and maintaining long ACLs on each device.
- Areas in the GUI: design (network hierarchy, IP address pools, DHCP and DNS settings), policy, provision (device inventory) and assurance (health monitoring). Inventory can flag a device as non-compliant, for example for an outdated software image.

Traditional management compared with controller-based

	Traditional	**DNA Center**
Configuring	Device by device over SSH or console	Centrally, from the GUI or REST API
New devices	Configured by hand on the console before deployment	Can receive their configuration automatically
Policy and config	Per device, distributed	Stored and managed centrally
Software versions	Updated device by device	Managed centrally
Deployment	Slow, more error-prone	Faster, less human error

Remember for the exam

- Application layer = apps and scripts, control layer = controller, infrastructure layer = devices.
- Underlay = physical, overlay = virtual tunnels, fabric = both.
- Edge, border and control nodes. LISP = control plane, VXLAN = data plane, TrustSec = policy.
- SD-Access underlay: all Layer 3 links, IS-IS, no STP or FHRP.
- DNA Center is both an SD-Access controller and a general network manager. Its goal is intent-based networking.

Check yourself

1. Which SD-Access protocol builds the data plane tunnels? *(VXLAN.)*
2. A switch connects laptops and phones to the fabric. Which node role is it? *(Edge node.)*
3. Why is spanning tree unnecessary in a DNA Center-built SD-Access underlay? *(Every switch-to-switch link is a routed Layer 3 link, so there are no Layer 2 loops.)*

Do next

No lab for this day. The free DevNet DNA Center sandbox is optional practice. Download the Day 62 Anki cards and learn the underlay, overlay and fabric terms.

Day 63: Ansible, Puppet, Chef and Terraform

Exam domain: 6. Automation and Programmability · **Video:** Ansible, Puppet, & Chef (22 min) · Terraform (22 min)

Key ideas

- **Exam topic 6.6:** recognize what configuration management tools can do. You don't have to use them. The v1.1 blueprint names Ansible and Terraform; the course keeps Puppet and Chef because they show the agent-based model by contrast.
- **Why configuration management:** in networks of hundreds or thousands of devices, manual CLI changes cause **configuration drift** (devices slowly stop matching the standard). These tools push one standard everywhere, check compliance, and make changes repeatable and auditable.
- **Infrastructure as code (IaC):** infrastructure is defined and managed through machine-readable files instead of the CLI or a GUI. All four tools are IaC.

	Ansible	**Puppet**	**Chef**
Written in	Python	Ruby	Ruby
Agents	**Agentless**	Agent-based (or a proxy agent)	Agent-based
Model	**Push**	Pull	Pull
Transport / port	SSH (TCP 22)	Agents reach the Puppet master on **TCP 8140**	Server talks to clients on **TCP 10002**
Files	YAML: playbooks, inventory, templates (Jinja2), variables	Proprietary language: manifests, templates	Ruby-based DSL: resources, recipes, cookbooks, run-lists
Server name	Control node	Puppet master	Chef server

- **Ansible files:** a **playbook** lists the tasks (the "what to do"), the **inventory** lists the devices and groups, **templates** hold configs with placeholders, and **variable** files fill them in. The control node connects over SSH; nothing is installed on the devices.
- **Chef's kitchen terms:** resources are the ingredients (configuration objects), recipes describe what to do with them, cookbooks group related recipes, and a run-list orders the recipes for a node. Few Cisco devices support a Chef agent, so it's the least used of the three for networking.

Terraform (added in v1.1)

- **Provisioning vs configuration management.** Provisioning tools (Terraform) create, change and delete infrastructure, such as VMs and virtual routers on a cloud platform. Configuration management tools (Ansible, Puppet, Chef) maintain systems that already exist. They're often paired: Terraform builds, Ansible configures.
- **Mutable vs immutable.** Configuration management usually changes things in place (mutable). Terraform typically replaces a resource with a new one built from the definition (immutable), which removes configuration drift.

- **Procedural vs declarative.** Procedural (imperative): you write the steps and their order (`hostname`, `ip address`, `no shutdown` …). Declarative: you describe the end state ("R1 with 192.168.1.1/24 on G0/1, interface up") and the tool works out the steps. Terraform and Puppet are declarative; Ansible and Chef are treated as procedural for the CCNA, though Ansible has declarative elements.
- **Terraform basics:** agentless; configuration files in HCL (HashiCorp Configuration Language); **providers** are plugins that talk to each platform's API (AWS, Azure, Cisco); a **state** file records what Terraform built. Workflow: write the config, `terraform plan` to preview, `terraform apply` to make it so, `terraform destroy` to remove it.

Remember for the exam

- Ansible: agentless, push, SSH, YAML playbooks, Python.
- Puppet: agent-based, pull, TCP 8140, manifests, Ruby.
- Chef: agent-based, pull, TCP 10002, recipes and cookbooks, Ruby.
- Terraform: provisioning, declarative, immutable, agentless, HCL, providers, plan then apply.

Check yourself

1. A team wants to manage switches without installing anything on them. Which tool fits? *(Ansible: agentless, over SSH.)*
2. Which tool's agents pull from a master on TCP 8140? *(Puppet.)*
3. Why does an immutable approach eliminate configuration drift? *(Changes replace the resource with a fresh one built from the definition, so no one-off edits pile up.)*

Do next

This is the last day of lectures. Review your weakest domain, do the Mega Lab, then start practice exams.

Made with HoverNotes · https://hovernotes.io/courses/jeremys-it-lab-ccna