Cisco CCNA 200-301 Certification Study Notes

Code: cisco-ccna

Network Fundamentals

OSI and TCP/IP Reference Models

Foundational models for network communication

OSI Model (7 Layers)

Layer	Name	PDU	Key Protocols
7	Application	Data	HTTP, HTTPS, DNS, DHCP, FTP, SMTP
6	Presentation	Data	TLS/SSL, JPEG, ASCII
5	Session	Data	NetBIOS, RPC
4	Transport	Segment	TCP, UDP
3	Network	Packet	IP, ICMP, OSPF
2	Data Link	Frame	Ethernet, 802.1Q, STP
1	Physical	Bits	Cables, Hubs, Signals

TCP vs UDP

TCP

Connection-oriented, reliable, ordered delivery. Uses 3-way handshake (SYN, SYN-ACK, ACK). Used by HTTP, FTP, SSH.

UDP

Connectionless, fast, no delivery guarantee. Used by DNS, DHCP, TFTP, VoIP, video streaming.

Common Port Numbers

- **20/21:** FTP | **22:** SSH/SFTP | **23:** Telnet | **25:** SMTP
- **53:** DNS | **67/68:** DHCP | **69:** TFTP | **80:** HTTP
- **110:** POP3 | **143:** IMAP | **161/162:** SNMP | **443:** HTTPS
- **514:** Syslog | **830:** NETCONF | **8080:** HTTP Alternate

Exam Focus Areas

- TCP uses 3-way handshake – SYN, SYN-ACK, ACK
- UDP is used where speed > reliability (DHCP, DNS, TFTP)
- Memorize well-known ports 0-1023
- IPv4 header = 20 bytes minimum; IPv6 header = 40 bytes fixed

IPv4 Addressing and Subnetting

IPv4 address classes, CIDR, and subnetting

IPv4 Address Classes

Class	Range	Default Mask	Use
A	1.0.0.0 – 126.255.255.255	/8 (255.0.0.0)	Large networks
B	128.0.0.0 – 191.255.255.255	/16 (255.255.0.0)	Medium networks

Class	Range	Default Mask	Use
C	192.0.0.0 – 223.255.255.255	/24 (255.255.255.0)	Small networks
D	224.0.0.0 – 239.255.255.255	N/A	Multicast
E	240.0.0.0 – 255.255.255.255	N/A	Reserved/Experimental

Private Address Ranges (RFC 1918)

- 10.0.0.0/8 — Class A private (10.0.0.0 – 10.255.255.255)
- 172.16.0.0/12 — Class B private (172.16.0.0 – 172.31.255.255)
- 192.168.0.0/16 — Class C private (192.168.0.0 – 192.168.255.255)

Subnetting Key Formula

- Subnets = 2^(borrowed bits) | Hosts = 2^(host bits) – 2
- /30 = 255.255.255.252 → 4 addresses, 2 usable (point-to-point links)
- /29 = 255.255.255.248 → 8 addresses, 6 usable
- Broadcast = last address in subnet; Network = first address

> ⚠ **Exam Focus Areas**
>
> - 127.0.0.0/8 is loopback (localhost = 127.0.0.1)
> - 169.254.x.x is APIPA (link-local) — no DHCP response
> - /32 = single host route; /31 = point-to-point (RFC 3021)
> - VLSM allows different subnet masks within same network

Network Access

VLANs and 802.1Q Trunking

Layer 2 segmentation with VLANs

VLAN Concepts

- VLAN = broadcast domain at Layer 2; isolates traffic without separate switches
- Access ports: carry traffic for a single VLAN (untagged)
- Trunk ports: carry multiple VLANs using 802.1Q tags
- Native VLAN: untagged traffic on a trunk (default VLAN 1 — change for security)
- VTP (VLAN Trunking Protocol): propagates VLAN DB — Server/Client/Transparent modes

Key Commands

Command	Purpose
switchport mode access	Configure port as access port
switchport access vlan 10	Assign port to VLAN 10
switchport mode trunk	Configure port as trunk
switchport trunk native vlan 99	Set native VLAN to 99
switchport nonegotiate	Disable DTP negotiation
show vlan brief	Display all VLANs
show interfaces trunk	Show trunk ports and allowed VLANs

Spanning Tree Protocol (STP)

- STP (802.1D): Prevents Layer 2 loops — elects Root Bridge (lowest BID)
- BID = Bridge Priority (default 32768) + VLAN ID + MAC address
- Port states: Blocking → Listening → Learning → Forwarding (Disabled)

- RSTP (802.1w): faster convergence — same concept but port roles: Root, Designated, Alternate, Backup
- PortFast: skip Listening/Learning for access ports; BPDU Guard: err-disable if BPDU received on PortFast port

⚠ Exam Focus Areas

- Trunk requires same native VLAN on both ends — mismatch = CDP warning
- 802.1Q native VLAN is sent untagged — VLAN hopping attack vector
- PVST+: Cisco's per-VLAN STP — one STP instance per VLAN
- EtherChannel bundles links — LACP (802.3ad) or PAgP (Cisco)

IP Connectivity

Routing Concepts and OSPF

Static routing, dynamic routing, and OSPF

Administrative Distance (AD)

Route Source	AD Value
Connected	0
Static	1
EIGRP Summary	5
eBGP	20
EIGRP Internal	90

Route Source	AD Value
OSPF	110
IS-IS	115
RIP	120
EIGRP External	170
iBGP	200

OSPF Key Concepts

- Link-state IGP; uses SPF (Dijkstra) algorithm; metric = cost (100/bandwidth Mbps)
- Hello/Dead timers: 10s/40s (broadcast) or 30s/120s (NBMA)
- Router ID: highest loopback IP > highest physical IP (or manual)
- DR/BDR elected on multi-access segments (highest priority, then RID)
- Area 0 = backbone; inter-area routes shown as 'O IA' in routing table
- Stub area: blocks Type 5 LSAs; Totally Stubby: blocks Type 3/4/5; NSSA: allows Type 7

Static Routes

- ip route [dest-network] [mask] [next-hop | exit-int]
- Floating static: higher AD than dynamic (e.g., AD 5 vs OSPF 110 — use AD 115)
- Default route: ip route 0.0.0.0 0.0.0.0 [next-hop]

⚠ Exam Focus Areas

- Longest prefix match wins regardless of AD
- OSPF cost = 100 Mbps reference / link bandwidth
- EIGRP composite metric uses bandwidth + delay by default

- BGP is used for Internet routing (eBGP between AS, iBGP within AS)

IP Services

DHCP, NAT, and NTP

Network services: address assignment, translation, and time

DHCP

- DORA process: Discover → Offer → Request → Acknowledge
- ip helper-address: forwards DHCP broadcasts to server on different subnet
- T1 timer = 50% of lease (unicast renewal to server)
- T2 timer = 87.5% of lease (broadcast rebind if T1 fails)
- DHCP snooping: filters untrusted DHCP messages; trust uplink ports

NAT Types

Static NAT

One-to-one mapping (internal server → public IP)

Dynamic NAT

Pool of public IPs; one-to-one but temporary

PAT/NAT Overload

Many-to-one; tracks port numbers. Most common.

NTP

- Stratum 0 = atomic clock (reference); Stratum 1 = directly connected to Stratum 0
- Each hop adds one stratum level; Stratum 15 = max usable; 16 = unsynchronized
- ntp server [IP] → configure device as NTP client
- ntp master [stratum] → configure as locally authoritative NTP source

Exam Focus Areas

- ip nat inside/outside commands on interfaces — direction matters
- ip nat inside source — translates inside local to inside global
- DHCP bindings: show ip dhcp binding
- SNMPv3 > SNMPv2c (authentication + encryption with authPriv)

Security Fundamentals

ACLs and Security

Standard/extended ACLs, port security, and AAA

ACL Types

Standard ACL (1-99, 1300-1999)

Matches source IP only. Apply close to destination.

Extended ACL (100-199, 2000-2699)

Matches source/dest IP, port, protocol. Apply close to source.

Named ACL

Descriptive names; easier to edit with sequence numbers.

Port Security

- Limits MAC addresses on access ports
- Violation modes: Protect (drop, no log), Restrict (drop + log), Shutdown (err-disable)
- switchport port-security maximum 2
- switchport port-security violation restrict

AAA Framework

- Authentication: Who are you? (username + password)
- Authorization: What can you do? (privilege level, commands)
- Accounting: What did you do? (logging, audit)
- RADIUS: UDP 1812/1813; encrypts only password; used for network access
- TACACS+: TCP 49; encrypts entire payload; used for device admin
- 802.1X: port-based NAC — Supplicant, Authenticator, Authentication Server

⚠ Exam Focus Areas

- ACLs have implicit deny all at the end
- Apply ACL to interface: ip access-group [ACL] in|out
- VTY ACL: ip access-class [ACL] in
- DHCP Snooping trusted: uplinks/DHCP server ports; untrusted: access ports

Automation

Network Automation and Programmability

SDN, APIs, automation tools, and data formats

SDN Architecture

- Control plane: routing decisions (moved to controller in SDN)
- Data plane: actual packet forwarding (hardware/ASIC)
- Management plane: device management (SSH, SNMP, NETCONF)
- Northbound API: SDN controller ↔ applications (REST, JSON/XML)
- Southbound API: SDN controller ↔ network devices (OpenFlow, NETCONF)

Configuration Management Tools

Puppet

Pull model; agent-based; DSL language; master/agent architecture

Chef

Pull model; agent-based; Ruby; recipes and cookbooks

Ansible

Push model; agentless; YAML playbooks; SSH-based; most common in CCNA

APIs and Data Formats

- REST API: uses HTTP verbs (GET/POST/PUT/DELETE); stateless; JSON/XML
- NETCONF: XML-based; uses SSH (TCP 830); RPCs for get-config, edit-config
- RESTCONF: REST over NETCONF; YANG models; HTTP(S); JSON or XML
- YANG: data modeling language used by NETCONF/RESTCONF; leaf/container/list

Data Formats

Format	Syntax	Use
JSON	Key-value, curly braces	REST APIs, most common
XML	Tags, hierarchical	NETCONF, SOAP
YAML	Indentation-based	Ansible playbooks, config files

⚠ Exam Focus Areas

- Ansible playbooks use YAML — ios_config module for Cisco IOS
- NETCONF uses port 830 (SSH); RESTCONF uses HTTPS (443/8443)
- Python: dictionaries accessed with ['key'] or .get('key')

- IaC (Infrastructure as Code): version control for network configs