Day 1 Network

computer network is digital telecommunicating network allow nodes to share resources

nodes: router
switch
firewall
server
client

client
is a device that access a service made available by server
server
is a device that provide function or services for client

a device can be a client in some situation and be a server vice versa

LAN (Local Area Network)
usually connected through a switch
Switch
have many network interface/port for end host to connect to
act like a road between village in a small area
eg: help connect pc in a floor of the office building of the company
provide connectivity for hosts in the same LAN
cant connect between other LANs or the internet
=> like a small village, connect each house to a culture house (nhà văn hoá) in Vietnam

Router
have less network interface/ port than switch
provide conectivity between LANs, through the internet
act like a bridge that connect roads between 2 small area

Network Firewall
specialty network security hardware device (not the firewall programm like window defender)
control network traffic entering and exiting the network
can be placed inside or outside the LAN
config with rule to determine which should be allowed and denied / to monitor and control
Next-generation firewall: combine traditional firewall feature with more advanced filtering functionalities
Host-based firewall
software application that filter traffic entering and exiting a host machine like a PC
eg: window defender

Labs = Hand on practice with technology you studying
in CCNA practicing configuring cisco switch and router

Cisco Package Tracer
simulator for building network without hardware

CLI (Command Line Interface)
is a command line for configure router and switch

Day 2 Interfaces(port) and cables

RJ-45 (Registered Jack)
is the connector used on the end of a copper Ethernet cable

Ethernet
is a collection of network protocals/standards
Defined in IEEE (Institute of Electrica and Eletctronics Engineers) 802.3 standard in 1983
Ethernet standards (copper wire)

Speed	Common Name	IEEE Standard	Informal Name	Maximum Length
10 Mbps	Ethernet	802.3i	10BASE-T	100 m
100 Mbps	Fast Ethernet	802.3u	100BASE-T	100 m
1 Gbps	Gigabit Ethernet	802.3ab	1000BASE-T	100 m
10 Gbps	10 Gig Ethernet	802.3an	10GBASE-T	100 m

BASE = baseband signaling
T = twisted pair cable

Network Protocols
establish a common language between devices to them for communicating

Bits value represent by 0 or 1 to operate computer/devices... => speed (internet speed, download speed, install speed) measured in bits per secs (Kbps, Mbps, Gbps)
byte series of 8 bits and can convert to data => size measured in bytes (MB,GB) => 8 time larger than bit
=>Mbps slower 8 times to MB/s

1 kilobit (Kb) = 1000 bits
1 megabit (Mb) = 1000 Kb

UTP Cable Unshielded twisted pair
Have 4 pair of twisted string, total 8 string (pin)
network interface card on Client/Server/Router/Firewall network interface on Switch

	Transmit	1	-------	1	Receive
	Tx	2	-------	2	Rx
	Receive	3	-------	3	Transmit
		4		4	
		5		5	
	Rx	6	-------	6	Tx
		7		7	
		8		8	

10Base-T and 100Base-T only use wire 1, 2, 3, 6

Device	Transmit(Tx)	Receive(Rx)
PC	Pin 1 and 2	Pin 3 and 6
Server	Pin 1 and 2	Pin 3 and 6
Firewall	Pin 1 and 2	Pin 3 and 6
Router	Pin 1 and 2	Pin 3 and 6
Switch	Pin 3 and 6	Pin 1 and 2

Full-duplex transmission is both device can sent/receive data at the same time

Straight-through cable

a cable that both RJ-45 end connect pin 1 vs pin 1, pin 2 vs pin 2... on each end

Crossover cable

a cable that a pin from one end connect to other index pin from other end
Using for connecting Pc/server/router/firewall with each other without switch
Pin 1 to 3
Pin 2 to 6
Pin 3 to 1
Pin 6 to 2

Auto MDI-X Auto Medium Dependent Interface Crossover

a **feature** on device **port** allow device to automatically change it Receive (Rx) and Transmit (Tx) pin base on it neighbor transmitting pin

10Base-T and 100Base-T use all 8 wire 4 pair (1,2) (3,6) (4,5) (7,8)

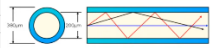
all pair is bidirectional (can be both Rx and Tx)

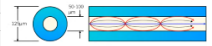
Fiber-Optic Cable

sent light as signal over cable, using SFP port instead RJ45 port
usually for connecting router
have 2 separate wire/connector to transmit and receive data (each Rx and Tx)
2 type

single mode fiber
- have smaller core diameter
- light enter at single angle
- allow longer cable (more than 500m)
- expensive

multimode fiber
- wider core diameter than single mode fiber
- allow multiple angles of light wave to enter the fiberglass core than single mode fiber
- allow longer cable than UTP but shorter cable than single mode fiber (max 550m)
- cheaper than single-mode fiber





Fiber-Optic cable standards (light cable)

Informal Name	IEEE Standard	Speed	Cable Type	Maximum Length
1000BASE-LX	802.3z	1 Gbps	Multimode or Single-Mode	550 m (MM) 5 km (SM)
10GBASE-SR	802.3ae	10 Gbps	Multimode	400 m
10GBASE-LR	802.3ae	10 Gbps	Single-Mode	10 km
10GBASE-ER	802.3ae	10 Gbps	Single-Mode	30 km

UTP vs Fiber-Optic

UTP	Fiber-Optic
Lower cost	Expensive
Shorter in maximum lenght	Longer
Vulnerable to EMI	No vulnerable
RJ45 port cheaper	SFP port more expensive
Emit faint signal => securitiy risk while	Fiber-Optic none

Day 3

OSI Model & TCP/IP suite

net working model categorize and provide structure for networking protocol

protocol is a set of rule how network device and software should communicate/work

OSI model Open systems Interconnection

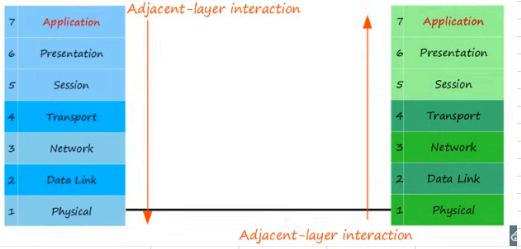
no longer in used but have big impact on referring

created by ISO (International Standardize Organization)

have 7 layer, data will start add information from each layer top to bottom layer and transfer to the receiver and strip of layer from bottom to the top

encapsulation decapsulation

OSI Model – Application Layer

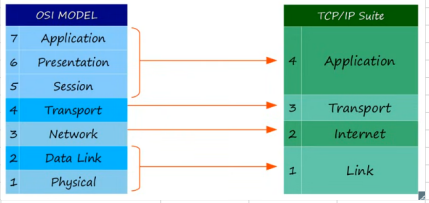


		Application	closet to end user	
			the protocol that interact with software application like web browser	
			eg: http and https	
			function:	identifying communication partner
				synchronizing communication
	Data			
		Presentation	data in the application layer is in application format => need to translate (eg: encrypt) to sent over network	
			function:	translate between application and network format
			eg: encrypt and decrypt	
		Session	control dialogue (sessions) between communicating host	
			function:	establish, manage terminate connection between local application
			eg: connection between web browser and youtube	
	add Layer 4 header	Transport	function:	segment and reassemble data for communication between end host
	=> Data + L4 header = "segment"			eg: segment a video into many small frame and reassemble vice versa
				provide host-to-host communication
	add layer 3 header	Network	function	provide connectivity between end host on different network (eg: outside the LAN)
	=> "segment" + layer 3 header = packet			provide logical addressing (IP addresses)
				provide path selection between source and destination
			eg: router	
		Data link	function	provide node to node connectivity and data transfer
	add layer 2 header and trailer			eg: PC to switch, switch to router,...
	=> layer 2 trailer + packet + layer 2 header = frame			define how data is formatted for transmission over a physical medium (eg: copper UTP cable)
				detect and correct physical layer error
				use layer 2 addressing separate from layer 3
			eg: switch	
		Physical		define physical characteristic of the medium use to transfer data
				eg: voltage, distance, connector,...
				digital bit are convert to electrical or radio
			eg: cable	

data, segment, packet, frame is a protocal data unit (PDUs)

TCP/IP Suite a conceptual model and set of communication protocol used in the internet and other network
similar structure to OSI model but less layer
actually using in modern network

have 4 layer



Day 4 Cisco IOS CLI is the operating system used on Cisco device (like window, macOS, Linux,...)

CLI (command-line interface)
GUI (graphical user interface)

console port => port to connect a cisco device to a monitor device (eg: pc laptop)
can connect with RJ45-DB9 cable **(rollover cable)**
rollover cable: pin 1 connect with pin 8, 2-7, 3-6,...

Terminal Emulator (PuTTy)

	Cisco default:	9600 bit per sec
		8 data bits

1 stop bit
no parity
no flow control

EXEC mode — is when CLI shown hostname + >
eg: Router>

can see some information
can't config or make any change

privileged EXEC mode
prompt enable to access
provide complete access to view the device configuration, restart device,...

is when CLI shown hostname + #
eg: Router#

global configuration mode
after in privileged EXEC mode prompt "configure terminal" or "conf t" to access

enable password
can set password with "enable password " + password

show running-config / startup-config
to display the file on CLI

saving the configuration
write
write memory
copy running-config startup-config — will copy every config in running file to startup file (at first cisco device will not have startup file yet)

service password-encryption
encrypt the password (if not user can see through show config file)
this like a switch for current and future password (if disable the current pass will not be decrypt)
enable secret
more security encrypt password (automaticaly encrypt)
cant use both secret and password => both enable => secret will have precedented

run +command
execute a privilege EXEC level command from global

no + command
remove the command already prompt before

config file — 2 separate config file
running-config = the current active configuration, enter command in CLI will edit this file
startup-config = the config file will be loaded upon restart of the device

Day 5 — Ethernet LAN Switching

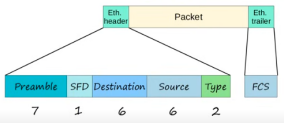
LAN
combine by end host and switchs connect to each other
data sent between LAN is frame (Layer 2)

Ethernet Frame
Layer 2 Header

preamble and SFD (start frame delimiber)
for synchronization and allow the receiving device to be prepare
preamble — lenght: 7 byte (56 bits)
alternating 1 and 0 (binary)
eg: 10101010 * 7
allow device to synchronize their receiver clock
SFD — length: 1 byte (8bit)
10101011
mark the end of the preamble and beginning of the rest

destination and source
indicate the device sending and receiving
consist of MAC address
MAC = Media Access Control
length 6 byte (48 bit) address of the physical device

type (or length)
2 byte (16 bit)
value > 1536 indicate type of the encapsulate packet (usually IPv4 or IPv6), and the length determined via other method
value <= 1500 indicate the length of the encapsulate packet in byte



Packet

Layer 2 trailer
frame check sequence

4 byte (32 bits)
to detect corrupted data by running a CRC algorithm over received data
CRC = Cyclic redundandcy check

MAC Address
6 byte (48 bits) physical address assigned to the device when it is made
aka burned in address
globally unique
first 3 byte are OUI (Organizationally unique identifier) => assigned to company making device like cisco
last 3 byte are unique to device itself
written in hexadecimal

Hexadecimal 16 digit
0-9 A-F

unicast frame: frame destined for single target

switch using MAC Address table that have MAC and Interface column to match and remember
first switch receive data through sender => it add sender MAC address to learn/remember it interface
then switch send data to receiver
unknown unicast frame (but if the receiver is unknown) => switch flood (forward the frame) out of all other interface

if receiver is not the one have MAC address => it will drop the packet

known unicast frame (frame that switch know the receiver through MAC address table)
Dynamic MAC address are removed from table every 5 mins of inactivity

Day 6 Ethernet Frame
Payload (packet) minimum size is 46 bytes
if a packet smaller than 46 bytes is sent => a padding byte are added and these byte are all 0
when sent a packet, IP source and destination will be known, only destination MAC is unknown
eg: 192.168.1.1 sent to 192.168.1.2 => last part is address on the LAN for each end host

ARP
Address Resolution Protocol => to find destination MAC address from sender
discover layer 2 address (MAC address) of a known layer 3 address (IP andress)
consist 1 message
ARP request — sent by sender to receiver to request MAC address
boardcast = flood = sent to all host on the network

ARP reply — sent by receiver to reply MAC address
unicast

Ping
a network utility used to test reachability
measure round-trip time (go to and go back)
use two message
ICMP Echo request — ICMP = Internet Control Message Protocol
ICMP Echo reply
similar to ARP but both unicast (known address)
command: ping + IP address

Day 7 + 8 IPv4 addressing
(Internet Protocol version 4)
IP meaning
eg: 192.168.1.0/24
"/24" mean first 3 part (192.168.1) represent the net work and the last part represent the end host (0)
because IP address is 32 bits length => 24/32 = 3/4 => first 3 is network
Boardcast only limit on LAN => if it meet a router (to reach for the internet) => it will stop
because ARP request only sent for IP on LAN (router will detect it and stop sending out)

IPv4 Address
length: 4 bytes 32 bits
written in decimal base 10 (actually binary but convert to easy read)

octet = 8 bits group (binary with 8 figure) eg: 11001001

5 classes

Class	First octet	First octet numeric range
A	0xxxxxxx	0-127
B	10xxxxxx	128-191
C	110xxxxx	192-223
D	1110xxxx	224-239
E	1111xxxx	240-255

numeric range base on convert octet to decimal
eg: 01111111 => 127

Class A
first octet normaly from 0-126
because 127 is reserved for loopback addresses
have more possible end host than B C but fewer possible network
usable range is from 1-126
maximum usable = 2^24 - 2 (network address and boardcast network) = 16777214

Loopback addresses

used to test network stack
eg: if try to ping to 127.() address => the computer will try to ping itself

Class B
maximum usable = 2^16 - 2 (network address and boardcast network) = 65534

Class C
maximum usable = 2^8 - 2 (network address and boardcast network) = 254

Class D
Multicast addresses
Class E
ussualy reserved for experiment

Class	First octet	First octet numeric range	Prefix Length
A	0xxxxxxx	0-127	/8
B	10xxxxxx	128-191	/16
C	110xxxxx	192-223	/24

Prefix length
is the /24, /16, /8 after the IP address to separate network and end host part
some device use netmasks to replace
eg netmasks: 255.0.0.0 = /8
255.255.255.0 = /24

Network address
Host portion of the address is all 0 (eg: 192.168.1.0/24) => is the network address itself
network address can't assign to host

Boardcast Address
host portion of the address is all 1 (eg: 192.168.1.255) => sent to this address will be sent to all
cant assign to host too
MAC address = FFFF.FFFF.FFFF.FFFF

Router
the router connect with LAN usually have the last usable IP
eg: in a 192.168.1.0/24 network => IP of router is 192.168.1.254
in 10.0.0.0/8 => IP router: 10.255.255.254

Configuration
show interface list
on the CLI command => enable => show ip interface brief
Status column is layer 1 status
protocol column is layer 2 status
enter configuration for specific interface
entering interface configuration mode => " interface" + interface_name
eg: "interface gigabitethernet 0/0"
set the ip for the choosed interface
set ip "ip address"+ address + "subnet mask"
eg: ip address 10.255.255.254 255.0.0.0

show specific interface
"show interfaces" + interface

show interface description
"show interfaces description"
set interface description
enter configuration mode for that specific interface
"description" + description

set the interface status up
enter interface config mode => no shutdown

Day 9 Switch interfaces
CLI show ip interface brief
usually switch will not have shutdown status on the interfaces when booting up by default
only router does

Router interfaces have the `shutdown` command applied by default
=will be in the `administratively down/down` state by default

Switch interfaces do NOT have the '`shutdown`' command applied by default
=will be in the `up/up` state if connected to another device
OR
in the `down/down` state if not connected to another device

"show interfaces status" to list down and check all interface status

multilayer switching => need to assign IP address to a switch
normally dont needed

configuring interface speed and duplex (full mean data can be both sent and receive coincide and half mean only 1 at a time)
enter the configurate mode for that specific interface

config speed "speed" + speed

config duplex "duplex" + auto/full/half

configurate multiple/range interface
"interface range" + range (eg: f0/5 - 12)

CSMA/CD for using Hub (precedent of Switch) => not using now cuz low tech and can only apply half duplex

Speed/duplex autonegotiation
interface that can run at different speed (eg: 10/100) will have default setting of speed auto and duplex auto

autonegotiation between 2 interface and match with the highest speed/duplex

if autonegotiation is disable in the device connect to the switch
=> speed: switch will try to sense the speed that other device is operating at
=> if fail => use the slowest supported speed

=> duplex: if speed is 10 or 100 Mbps => use half duplex and if 1 Gbps => full duplex

Interface Error
"show interfaces" + interface

Runt: frame are smaller than minimum frame size (64 bytes)
Giants: frame are bigger than maximum frame size (1518 byte)
CRC: frame that fail CRC check (layer 2 trailer)
frame: frames that have incorrect format
input error: total of 4 above
output error: frame switch try to sent but fail due to error

Day 10 IPv4 header (Internet protocol version 4)

Routing Help sent data between device on separate network

Layer 3 header
Version Length: 4 bits
for identify the version of IP is used
eg: IPv4 = 4 (0100)
IPv6 = 6 (0110)

IHL Length: 4 bits
indicate the total length of the header due to option field is variable in length
indentify the length of the header in 4 byte increment
eg: value = 5 => length of the header = 4 * 5 = 20 bytes
minimum value = 5 (length of IP packet without any IP option)
maximum value is 15 (60 bytes)

DSCP Length: 6 bits
"Differentiated services codepoint"
use for QoS (Quality of service)
use to prioritize delay-sensitive data (streaming voice, video,...)
if not use = 0

ECN Explicit congestion notification
Length: 2 bits
provide end-to-end notification of network congestion without dropping packet
optional feature because need require both endpoint and the network infrastructure to support it
if not use = 0

total length Length: 16 bits
indicate total length of the packet (L3 header + segment)
measure in bytes
min value = 20 (IPv4 header with no encapsulate data)
max = 65535 (maximum 16 bit value)

identification field
Length: 16 bits
used to identify which packet the fragment belong to if it being fragmented because being too large
all fragment of the same packet will have their own IPv4 header with the same value in this field

fragmented when packet larger than MTU
MTU = maximum transmission unit ~ 1500 bytes

fragment is assemble by receiving host

flag field
- Length: 3 bits
- used to control/identify fragment
- bit 0: reserved, alway set to 0
- bit 1: dont fragment (DF) to indicate packet shouldnt be fragmented => if 0 can be both fragmented or not and have to check bit 2
- bit 2: more fragment (MF) set to 1 if there are more fragments in the packet, to 0 for the last fragment

fragment offset field
- Length: 13 bits
- used to indicate the position of the fragment within the original (index), unfragmented IP packet
- allow fragment to reassemble in order if it receive not in order

time to live field
- Length: 8 bits
- a router will drop a packet with a TTL of 0
- use to prevent infinity loop
- originally design to indicate packet maximum lifetime in seconds

- in practice indicate a hoop count
- hoop count = each time the packet arrive at a router the router decrease the TTL by 1
- recomment default is 64 (00100000)

protocol field
- Length: 8 bits
- indicate the protocol of the encapsulate L4 PDU
- value of 6 : TCP
- value of 17: UDP
- value of 1: ICMP
- **value of 89: OSPF (open shortest path first) (help router to learn routes to destination from their neighbor without manually config the route)**

header checksum field
- Length: 8 bits
- used to check for error in the **IPv4 header only**
- when a router receive packet => compare this with the calculate the checksum of the header => if diff is error and router drop the packet

Source/destination IP Address field (day 8)
- length: 32 bits (each)

options fields
- length: 0 - 320 bits
- rarely used

- if IHL is greater than 5, it mean there are options

Wireshark

application for examine network traffic

Day 11

Routing Fundamental

routing is the process that router use to determine the path that IP packet should take over a network to reach their destination

routing table

where router store routes to all of their known destination

when router receive packet => look in routing table to find the best route to foward the packet

a route matches a packet destination if the packet destination IP address is part of the network specified in the route

- if there are more matched route => it will choose the most specific matching route (longest prefix length)
  - eg: match 2 route: 192.168.1.0/24 and 192.168.1.1/32
    - => the first include 256 different IP addressess and the second only 1 => router will choose 2

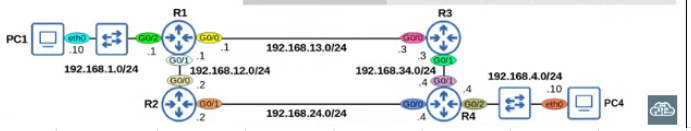
two main routing method

Dynamic routing
- router use dynamic routing protocol to share routing information with each other
- automatically and build their routing table
- eg: OSPF

Static routing
- manually configure routes by a sys admin / network engineer
- if there are more than 1 routes for matching route
  - admin can configure the router to
    - load balance (split and use both route)
    - or can use 1 as a backup and 1 as main path
- each router in the path will need two route: one to the receiver and one to the sender to ensure two way reachability

- To allow PC1 and PC4 to communicate with each other over the network, let's configure these **Static routes** on R1, R3, and R4.

Router	Destination	Next-Hop
R1	192.168.1.0/24	Connected
	192.168.4.0/24	192.168.13.3
R3	192.168.1.0/24	192.168.13.1
	192.168.4.0/24	192.168.34.4
R4	192.168.1.0/24	192.168.34.3
	192.168.4.0/24	Connected



CLI command

enter the configure terminal mode

3 way to configurate the route

1. "ip route" + ip address (destination network) + netmask + next-hop
   eg: ip route + 192.168.1.0 255.255.255.0 192.168.13.1 for R3 to R1

2. "ip route" + ip address (destination network) + netmask + exit-interface
   eg: ip route + 192.168.1.0 255.255.255.0 gig0/0
   this way rely on feature called proxy ARP to function properly

3. "ip route" + ip address (destination network) + netmask + exit-interface + next-hop
   eg: ip route + 192.168.1.0 255.255.255.0 gig0/0 192.168.13.1

Default Gateway

is mean default route (usually the last router that connect to the network/internet)

eg: a company with 3 router connect with each other in a internal network and only 1 reach out to the internet => it should be the default router

in default is the route to 0.0.0.0/0 => all variable => all the IP addresses

is the least specific route (shortest prefix length)

usually this will be set to the router IP address as default

CLI command setting up default gateway in the router (the interface that connect to the internet)

enter the config for specific interface

"ip route" + 0.0.0.0 0.0.0.0 next-hop (ip address of the internet)

configure CLI show ip route

list all the protocol which router can use to learn routes

L - Local

a route to the actual IP address configure on the interface (with /32 netmask => all octet is fixed => only 1 destination)

auto created when config the IP address on an interface and enable it with no shutdown

=> route to itself (is the IP address of the router) and keep the packet

C - connected

a route to the network the interface is connected to (with the actual netmask configure on the interface eg: /24 => the last octet is variable from 1 to 254)

auto created when config the IP address on an interface and enable it with no shutdown

=> route to all the host in the network

S - static

[x/y] is the administrative distance/metric

route

a route tell a router to send a packet to destination X, you should sent to next-hop Y

Next-hop

the next router in the path to the destination

or if the destination is directly connect to router => sent it directly to the destination

or if the destination is the router own IP address => receive the packet for yourself and not foward it

WAN wide area network

large geographical area

Day 12 Life of a packet

is the entire process of sending a packet to a remote destination

including ARP, encapsulation, de-capsulation

if a destination IP is not in the network IP => end host will sent it to the default gateway

before packet be sending to a router => it need to know it MAC address => ARP request sent out and received the ARP reply => sent packet to that router => sent to next router will need the MAC address of that router => router 1 sent ARP request to router 2 and receive ARP reply => router 1 sent packet to router 2 =>.. keep continue until r

after reach the destination => on the way back of the ping will no need ARP because MAC is known for each device

IP address destination/source will never change on the movement of the packet, only MAC address destination/source will be change

the MAC address destination/source will not be the switch MAC Address (only for router and end host) => act like a middle-man => not involve in the frame (layer 2 header)

CLI command for view MAC Address

on end-host (PC)

"ipconfig /all"

check the physical address

on router
enable => "ip interfaces" + interface
check the address is...
if there are (BIA / Burned in address) => that mean the router MAC address have been modify/assign new MAC address

Day 13/14/15 **Subnetting** **Divide network at Layer 3 level**

IPv4 Address classes

old fashion way

IANA (Internet assigned number authority) will assign IPv4 addresses/network to companies base on their size
big company => class A or B
small company => class C
however this lead to waste IP address
eg: point to point network (network only contain 2 router connect to each other)

Classless Inter-Domain Routing (CIDR)

the requirement of class A, B, C were removed
=> allow larger network to be split into smaller network, allowing greater efficiency

=> these small network called Subnet/Subnetwork

Fixed-Length subnet masks (FLSM)

CIDR allow to assign different prefix length
All of the subnet in the network use the same prefix length => equal size of subnet

Subneting class C network

eg: can be /25, /26, /27 instead of /24
note can't be /32 because the usable addresses will = 0 => only work on specify a identical IP address

eg: /27 will have the subnet mask as 255.255.255.224
224 = 11100000 = 2^7 + 2^6+ 2^5
and will have 2^(32-27)-2 = 30 usable addresses

/30 prefix length will have subnet mask as 255.255.255.252
252 = 11111100 = 2^7+...+2^2
will have 2^(32-30)-2 = 2 usable addresses
=> suit to apply for point to point network

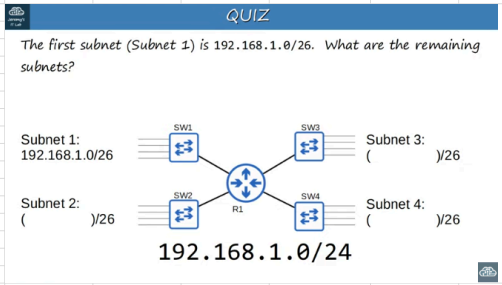
/25, /26,.../30 is the subnet of that larger class C network /24

CIDR (/31)

prefix length /31 can only apply for point to point network (router to router)
and have the subnet mask 255.255.255.254
=> only 2 ip address to be assigned (00000000 and 00000001 last octets)
although should reverse 2 address for network address and boardcast => but in point to point network is no needed
=> use ultimize only 2 address => save more than /30
=> the other remaining address in the /24 gonna be available to be used in other network

Dotted Decimal	CIDR Notation
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30
255.255.255.254	/31
255.255.255.255	/32

eg:



subnet 1: from 192.168.1.0 to 192.168.1.63 => 64 used address (include 1 network address and 1 boardcast)
subnet 2: 192.168.1.64/26 (from 192.168.1.64 to 192.168.1.127)
subnet 3: 192.168.1.128/26 (from 192.168.1.128 to 192.168.1.191)
subnet 4: 192.168.1.192/26 (from 192.168.1.192 to 192.168.1.255)
=> /24 can split to 4 /26 because is = 2^2 = 4 portion

Trick: number of subnet can create = 2^ number of borrowed bit
Trick: find boardcast address for a known IP address => write the octet in binary, find the borrowing bit and fixed it, change all the host bit into 1 => convert back to decimal (or can just find the next subnet and minus 1)
Trick: identify the subnet which IP address belong for => write the last octet to binary => replace all host bit to 0, borrowed bit stay the same => change that binary to decimal => that the subnet it belong to
eg: 192.168.29.219/29
=> 219 = 11011011 , /29 => 5 borrowed bit
=> 11011/000 = 216
=> ip address belong to 192.168.29.216/29 subnet

Prefix Length	Number of Subnets	Number of Hosts
/25	2	126
/26	4	62
/27	8	30
/28	16	14
/29	32	6

/30	64	2
/31	128	0 (2)
/32	256	0 (1)

Subneting class B network

same with class C but bigger number
eg: create 80 subnet for 172.16.0.0/16 network
/16 network => class B
number of borrowed bit need = 2^X > 80 => X = 7
=> borrowed 7 bit from 3rd octets
=> prefix is 172.16.0.0/23 (16+7)

example of those subnet
172.16.0.0/23
172.16.2.0/23
172.16.4.0/23
172.16.6.0/23

eg: what subnet host 172.25.217.192/21 belong to
/21 => class B and borrowed 5 bits
=> 217 = 11011001
=> remain 5 bit , replace 3 last bit with 0 = 11011000 = 216
=> belong to 172.25.216.0/21

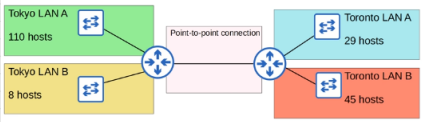
Subneting class A network

PC1 has an IP address of **10.217.182.223/11.**
Identify the following for PC1's subnet:

1) Network address:	10.192.0.0
2) Broadcast address:	10.223.255.255
3) First usable address:	10.192.0.1
4) Last usable address:	10.223.255.254
5) Number of host (usable) addresses:	2,097,150

Variable-Length subnet masks (VLSM)

creating subnet of different size, to make use of network addresses more efficient



192.168.1.0/24

step 1: assign the largest subnet at the start of the address space
step 2: assign the second-largest subnet after it
step 3: repeat the process until done

eg:

Tokyo LAN A	Network address: 192.168.1.0/25	110 host +2 => 2^7 = 128 => can borrow 8-7 = 1 bit (/25 prefix length)
	Boardcast address: 192.168.1.127/25	=> "0.1111111" = 127
	First usable address: 192.168.1.1/25	
	Last usable address: 192.168.1.126/25	
	Total number of usable host addresses	usable host = 2^7 -2 = 126 host
Toronto LAN B	Network address: 192.168.1.128/26	45 host +2 => 2^6 = 64 => can borrow 8-6 = 2 bit (/26 prefix length)
	Boardcast address: 192.168.1.191/26	128 = "10000000" '=> "10.111111" = 191
	First usable address: 192.168.1.129/26	
	Last usable address: 192.168.1.190/26	
	Total number of usable host addresses	usable host = 2^6-2 = 62 host
Toronto LAN A	Network address: 192.168.1.192/27	29 host + 2 => 2^5 = 32 => can borrow 8-5 = 3 bit (/27 prefix length)
	Boardcast address: 192.168.1.223/27	192 = "11000000" '=> "110.11111" = 223
	First usable address: 192.168.1.193/27	
	Last usable address: 192.168.1.222/27	
	Total number of usable host addresses	usable host = 2^5-2 = 30 host
Tokyo LAN B	Network address: 192.168.1.224/28	8 host + 2 (network and boardcast) => 2^4 = 16 => can borrow 8-4 = 4 bit (/28 prefix length)
	Boardcast address: 192.168.1.239/28	224 = "11100000" '=> "1110.1111" = 239
	First usable address: 192.168.1.225/28	
	Last usable address: 192.168.1.238/28	
	Total number of usable host addresses	usable host = 2^4-2 = 14 host

point to point connection (between router)
can use /31 prefix length
=> address: 192.168.1.240 and .241
=> but not recommend using in CCNA test

instead can use /30 prefix length

Network address: 192.168.1.240/30 — 2 host + 2 (network and boardcast) => 2^2 = 4 => can borrow 8-2 = 6 bit (/30 prefix length)
Boardcast address: 192.168.1.243/30 — 240 = "11110000" '=> "111100.11" = 243
First usable address: 192.168.1.241/30
Last usable address: 192.168.1.242/30
Total number of usable host addresses — usable host = 2^2-2 = 2 host

Convert prefix length to subnet mask

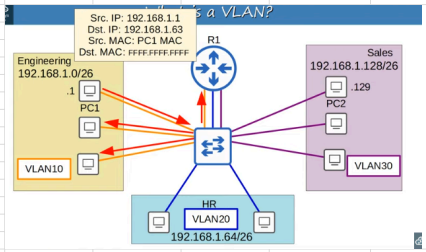
if is it a subnet => find the borrowed bit and convert to 1, convert all host bit to 0
eg: /27 => borrow 3 bit and host 5 bit left => "11100000" = 226 => subnet mask = 255.255.255.226
eg: /28 => borrow 4 bit and host 4 bit left => "11110000" = 240 => subnet mask = 255.255.255.240
eg: /18 => borrow 2 bit and host 6 bit left in class B => "11000000.00000000" = 192.0 => subnet mask = 255.255.192.0

Day 16/17/18 — VLANs (Virtual Local Area Networks)

LAN — is a single broadcast domain, including all device in that broadcast domain

Broadcast domain — the group of device which will receive a broadcast frame (destination MAC address of FFFF.FFFF.FFFF) sent by any one of the members

VLANs — a Virtual LAN can be set up on the switch (assign interfaces in to smaller LAN) **(divide the network at Layer 2)**
so if a broadcast/unknown unicast traffic is sent from a specific VLAN to switch => it only transfer to the interface that have the same VLANs
configured on a **per-interface** basis (separate VLAN with interface / logically separate end host at Layer 2)



the switch not perform **inter-VLAN routing** once VLANs is set, router will do it so all the connection between VLAN have to go through router
but when a host from a VLAN ping to host from another VLAN => have to go to the router and then go back to switch before going to other VLAN

CLI — VLAN configuration on switch
switch will auto have 5 default VLAN, VLAN 1 will be the default and have all the interface inside it, other 4 was for another purpose
VLAN name will be set as 10, 20, 30,...

enable
show vlan brief
to check current assigned VLANs

choose interface range to set to a specific VLAN
"switchport mode access" — to set interface as an access port

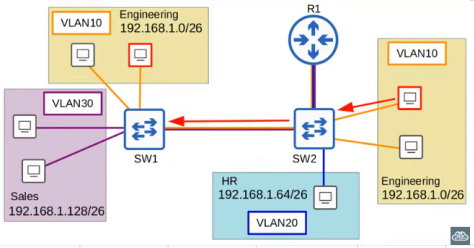
access port (untagged port) is a switchport belong to a single VLAN and usually connect to end host like PC

"switchport access vlan" + VLAN number
eg: switchport access vlan 10

enter config mode for specific VLAN or create new one
"vlan" + VLAN number

set name:
"name" + name/description

trunk port (tagged port) is a switchport belong to multiple VLAN on single interface



in this picture, instead of going through router to be inter-LAN routing, the switch 2 read the frame layer 2 header to know it from VLAN 10 => can sent directly to switch 1

because router only use 1 interface for trunk port -> must use sub-interfaces
VLAN tagging: switch will tag all frame that send over a trunk link (allow receiving switch to know which VLAN the frame belong to to forward it)
there are two main trunking protocols:
ISL (Inter-switch link)
old Cisco proprietary protocol create before

IEEE 802.1Q (dot1q)

industry standard protocol created by IEEE

it will insert the 802.1Q tag into the Layer 2 header (Ethernet frame) between source and type/length

length: 4 bytes (32 bits)

consist of two main field

Tag protocol identifier (TPID)

length: 2 bytes (16 bits)

value = 0x8100 (0x mean hexacedimal)

to indicate the frame is 802.1Q tagged

Tag control information (TCI)

consist of 3 sub-fields

802.1Q tag format

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

PCP (priority code point)

length: 3 bits

use for Class of Services (CoS) which prioritize important traffic in congested network

DEI (drop eligible indicatior)

length: 1 bits

use to indicate frame that can be drop if network is congested

VID (VLAN ID)

12 bit in length

idenfity the VLAN the frame belong to

12 bits = 4096 total VLAN range from 0 - 4095

but like IP address VLAN 0 and 4095 is for reserved and cant be used

=> actual VLAN usable range 1 - 4094

VLAN range

divided into 2 sections

normal VLANs: 1 - 1005

Extended VLANs: 1006 - 4094

older device cant use extended range

Native VLAN

a feature of 802.1Q (like the default gate way => can set to specific VLAN and if the packet sent on trunk port not have 802.1Q tag => sent to all host in that VLAN

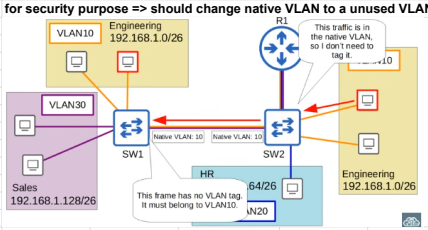
switch will not add an 802.1Q tag to frame in the native VLAN

default value = 1 => VLAN 1 is the default

when a switch receive untagged frame on a trunk port it assume the frame belong to the native VLAN

Native VLAN need to be matched between switch (should be consolidate in all switch in LAN)

for security purpose => should change native VLAN to a unused VLAN for performance and security (only allowed connection from specific VLAN can going through)



CLI

trunk port configuration on switch (apply truck on switch to switch only, do not config as switch on host connection)

enable

choose the interface that will be the trunk port (connect to other switch)

set the trunking protocol to 802.1Q / dont need if dont have

"switchport trunk encapsulation" + "dot1q"

set trunk port

"switchport mode trunk"

show trunk port list

"show interfaces trunk"

allow specific VLAN on a trunk port (write new list of allow)

"switchport trunk allowed vlan" + VLAN name

add specific VLAN on a trunk port (add to the list)

"switchport trunk allowed vlan add" + VLAN name

remove specific VLAN on a trunk port (remove from the list)

"switchport trunk allowed vlan remove" + VLAN name

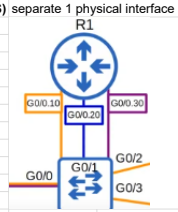
set native VLAN

"switchport trunk native vlan" + VLAN name (should be the one unused)

show trunk port list

"show interfaces trunk"

Router on a Stick (ROAS) separate 1 physical interface into multiple separate sub-interface



is used to route between multiple VLANs using a single interface on the router and switch
the switch interface is configured as a regular trunk
the router interface is configured using sub-interfaces, need configure the VLAN tag and IP address on each sub-interface

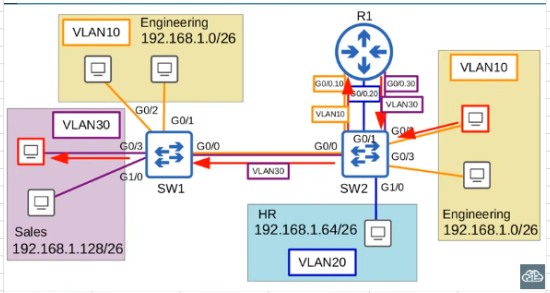
the router will behave as if frame arriving with a VLAN tag on the sub-interface configured with that VLAN tag
and will tag frames sent out of each sub-interface with the VLAN tag configured on the sub-interface

CLI **ROAS configuration on router**
enable + choose the specific interface
"no shutdown"

enter sub-interface mode
"interface" + interface name +"." + VLAN name
eg: "interface gig0/0.10" for VLAN 10

"encapsulation dot1q" + VLAN name
tell the router if the received frame tagged with the specific VLAN name as they arrived on this sub-interface
and it tagged any frame leaving this sub-interface with VLAN name tagged using dot1q

"ip address" + IP address + subnet mask
assign the IP address to the sub-interface



Native VLAN on a router
CLI 2 way of configure native VLAN on a router
assigned it to a sub-interface
"encapsulation dot1q" + VLAN id + "native" after enter conf t mode for that sub-interface
"ip address" + IP address + subnet mask
assign IP to sub-interface
configure IP address for native VLAN on router physical interface
"ip address" + IP address + subnet mask
assign directly to the interface => no tagging and untagging in the router

Step to configure VLAN network
1. Configure the switch interfaces connected to PCs/host as access ports in the correct VLAN.
   CLI
2. Configure the connection between SW1 and SW2 as a trunk, allowing only the necessary VLANs.
   Configure an unused VLAN as the native VLAN.
   Make sure all necessary VLANs exist on each switch
   CLI
3. Configure the connection between SW2 and R1 using 'router on a stick'.
   Assign the last usable address of each subnet to R1's subinterfaces.
   CLI and remember to configure on both SW and R
4. Test connectivity by pinging between PCs. All PCs should be able to reach each other.

Layer 3 (Multilayer) Switches
capable of both switching and routing
it is "layer 3 aware"
can assign IP address to it interface like router
can configure "routed ports" which function like an interface on router

can create virtual interface for each VLAN and assign IP address to those interface
can configure routes on it like router
can be used for inter-VLAN routing

suit for large network using inter-VLAN routing

Layer 3 Switches will act like a router and will stream/tagged the connection with dot1q
=> router dont have to do it, the flow will no need to go to the router to be tagged VLAN
Switch Virtual Interface (SVIs) are virtual interfaces and can assign IP addresses to in a multilayer switch

need to configure each PC to use the SVI as their gateway address instead of router

CLI enable

"default interface" + interface set the interface configure back to default

"ip routing" enable layer 3 routing on the switch, let it build it own routing table

get in the specific interface configuration (the one connect to router because now this switch act like a router so the connection to other router will be normal)
"no switchport"
change from layer 2 switchport to layer 3 routed port for a specific interface

"ip address" + ip + subnet mask
to set IP for that specific routed port

"ip route 0.0.0.0 0.0.0.0" + next hop (other router IP address)
to assign default route/gateway to the router when host want to connect to the outside internet

configure the SVIs IP address for each VLAN
requirement VLAN must exist on the switch
if not create it first
"vlan" + vlan name
requirement switch must have at least one access port in the VLAN in up/up state or 1 trunk port that allow the VLAN that is in up/up state

"interface vlan" + vlan id
eg: "interface vlan 10"

assign IP address
"ip address" + ip + subnet mask
to set IP for specific VLAN

"no shutdown"
SVIs are shutdown by default (like router interface)

Day 19 DTP / VTP

Cisco proprietary (dev by Cisco and only used on Cisco device)

DTP (Dynamic Trunking Protocol)
allow switch to dynamically determine their interface status (access or trunk) without manual configuration
DTP is enable by default on all Cisco switch interface
DTP will not form a trunk with a router, PC,... and the switchport will be in access mode (=> only apply for switch vs switch)
but for security purposes, manual configuration is recommended => DTP should be disable on all switchport

CLI switchport mode dynamic (beside learned access and trunk)

switchport mode dynamic + "auto" => switch not actively intend to form a trunk => it only follow the other switches mode
form a trunk if other switch have switchport mode trunk/dynamic desirable

switchport mode dynamic + "desirable" => switch intend to form a trunk if it can
actively try to form a trunk with other Cisco switches if the other switch have switchport mode trunk/dynamic desirable / dynamic auto
if the other switch on access mode => switch will not form a trunk, just only use access mode

"show interfaces" + interface + "switchport"
show switchport mode

"switchport nonegotiate"
to disable DTP negotiation
or can use "switchport mode access" instead

Administrative Mode	Trunk	Dynamic Desirable	Access	Dynamic Auto
Trunk	Trunk	Trunk	X	Trunk
Dynamic Desirable	Trunk	Trunk	Access	Trunk
Access	X	Access	Access	Access
Dynamic Auto	Trunk	Trunk	Access	Access

VTP (VLAN trunking protocol)
allow to configure VLAN on a central VTP server switch, and other switches (as VTP clients) will synchronize their VLAN database to the server

designed for large network with many VLANs, so dont have to configure each VLAN on every switch

rarely used and not recommended using it because incase add a switch that have largest revision number => replace all database with that configure of the old switch

three VTP version 1,2,3

three VTP mode: server, client, transparent, switch operate in VTP server mode in default
VTP domain include
server
can modify VLAN
store VLAN database in non-volatile ram (NVRAM)

increase the revision number every time a VLAN is modify
and advertise the latest version (by revision number) of the VLAN on trunk interface and synchronize their VLAN database to it

also function as client
=> highest revision number will be synchronize to all VTP

client
cant modify VLAN

VTP update the database when any switch got modify

transparent mode
not participate in VTP domain => not sync it VLAN database
maintain it own VLAN database in NVRAM
only foward it advertise to any switch in the same mode

Day 20 Spanning Tree Protocol (STP)

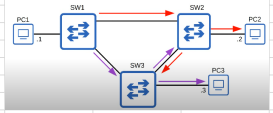
Network redundancy

essential part of network design
modern network expected to run 24/7/365 => if one network component fail => need backup for smooth operation
as much as possible => must implement redundancy at every possible point in the network

STP is layer 2 protocol, enable redundant layer 2 network
provide alternative path if one connection fail

Boardcast storms

is when switch flood out arp request on boardcast or unknown unicast frame => loop forever between switch as below



TTL (time to live) only have on layer 3 => not have in layer 2 so the switch will loop it arp indefinitely. => congested traffic => called boardcast storms
=> STP prevent this loop

Classic Spanning Tree Protocol

is IEEE 802.1D

all switches from all vendors run STP by default

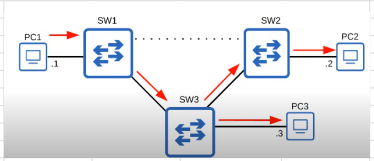
prevent layer 2 loop by placing redundant port in a blocking state, essentially disabling the interface

these interface will act as backup that can enter a forwarding state if an active (currently forwarding) interface fail

interface in forwarding state behave normally. send and receive all normal traffic

interface in blocking state only send or receive STP message (called BPDUs - bridge(switch) protocol data units) and other specific traffic

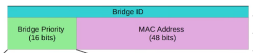
eg: dot line is 2 interface in blocking state => backup



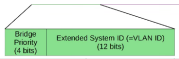
STP-enable switch send/receive Hello BPDU out of all interface (once every 2 seconds)
if switch receive a hello BPDU on an interface -> it know the interface connect to another switch

Switch use one field in the STP BPDU (the Bridge ID field) to elect a root bridge for the network (main path between switch)
switch with the lowest bridge ID become the root bridge

all port in root bridge will in forwarding state and other switch must have path to reach the root bridge



default bridge priority is 32768 = > by default smaller MAC address is used to elect root bridge



for update bridge ID
default bridge priority is 32769 = > bridge priority is 1000 and by default VLAN ID = 1

PVST (Per VLAN Spanning Tree) => use update bridge ID to run separate STP in each VLAN so in each VLAN different interface can be forwarding/blocking

STP bridge priority can only change in unit of 4096

Bridge Priority				Extended System ID (VLAN ID)											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

STP step:

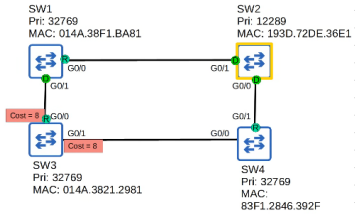
When a switch powered on => it assume it is te root bridge
give up the root bridge position once receive a BPDU with lower bridge ID
once the topology converge and all switch agree on the root bridge => only root bridge sent BPDUs
other switch will foward these BPDUs but not generate their own
all switch interface/port became designated port (forwarding state).

Each remaining switch will select one of it interface to be it root port. the interface with the lowest root cost will be the root port, also in a forwarding state

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

root port is the interface of switch which path to the root bridge

root port selection :
lowest root cost
lowest neignbor bridge ID
lowest neignbor port ID (default on the switch interface)

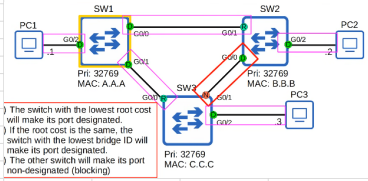


SW2 is the root bridge
all interface of SW2 is designed port
=> G0/0 Sw1 and G0/1 Sw4 will be the root port

Sw3 not connect directly to Sw2 => need to decide designed port by the lowest neignbor bridge ID (have same cost) is Sw1 (lower than Sw4)
=> G0/0 Sw3 became the root port and G0/1 Sw 1 is the designed port

each remaining collision domain will select ONE interface to be a designated port (forwarding state) and the other will be non-designated (blocking)

Designated port selection
interface on switch with lowest root cost
interface on switch with lowest bridge ID



SW2 G0/0 to SW3 G0/1 will be the collision domain
both have same root cost but G0/0 SW2 have lower bridge ID (MAC address) => is designated port
SW3 G0/1 will be non-designated port

CLI

"show spanning-tree"
to show the designated port, root port, non-designated port on the switch

"show spanning-tree vlan" + VLAN ID
to show if set up each VLAN separately

"show spanning-tree detail"
with more detail and total roots cost calc

"show spanning-tree summary"
list each VLAN and show how many interface are in each STP state

Day 21

Port State

STP Port State	Stable/Transitional
Blocking	Stable
Listening	Transitional
Learning	Transitional
Forwarding	Stable

Non-designated port are in blocking state (only receive STP BPDU, not learn MAC address)
effectively disable to prevent loop (not send/ receive regular network traffic => drop it)

After the blocking state, interface with designated or root role enter listening state in 15 sec by default
only forward/receive STP BPDU, not learn MAC address
After listening state, designated or root port will enter learning state, 15 secs long by default
only sent/receive STP BPDU, learn MAC address
After that become stable and in forwarding state
operate as normal
send/receive both BPDU and normal traffic and learn MAC address

STP Timers

STP Timer	Purpose	Duration
Hello	How often the root bridge sends hello BPDUs	2sec
Forward delay	How long the switch will stay in the Listening and Learning states (each state is 15 seconds = total 30 seconds)	15sec
Max Age	How long an interface will wait after ceasing to receive Hello BDPUs to change the STP topology.	20sec (10* hello)

if BPDU is not received by switch in 20 sec (when receive the countdown will restart) => the switch will re-evaluate it STP choice => including root bridge, local root, designated and non-designated port
if non-designated port is selected to become a designated or root port
=> transition from the blocking state to the listening (15 sec) to learning (15 sec) => can take total 20s + 15s + 15s = 50s for blocking interface to transition to fowarding

PVST+ destination MAC address of: (01:00:0c:cc:cc:cd)
regular STP destination MAC address of: 0180.c200.0000

STP Toolkit

Portfast

solve: help connect to end host right away instead of waiting 30 second to go through listening and learning state
bypass listening and learning state
should only be enable on interface which are connected to end host
not enable on interface connect to other switch => layer 2 loop occur

risk: unplug with end host and connect with another/old switch => risk of layer 2 loop can occur

CLI
2 way
1 configure for individual interface
enter interface config mode
"spanning-tree portfast"
enable only for the interface configured
2 configure for all access port
enter config mode by conf t
"spanning-tree portfast default"

BPDU Guard

function: if interface with a BPDU enable when it's received BPDU from another switch it will shut down the interface to prevent Layer 2 loop
usually use with portfast

The interface will got ErrDisable if receive BPDU

CLI
2 way
per port
enter interface config mode
"spanning-tree bpduguard enable"
global (all portfast enable port)
enter config mode by conf t
"spanning-tree portfast bpduguard default"

To re-enable the errDisable port
Manual — use shutdown and no shutdown to reset the disable port
Auto with ErrDisable Recovery — recover after a certain period of time
"errdisable recovery cause" + bpduguard to enable

BPDU Filter

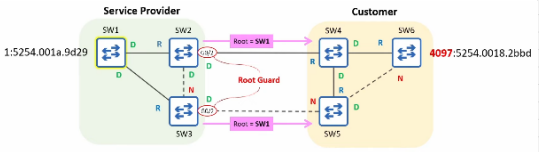
function: stop a port from sending BPDU, use when need maximum security (not sending a STP topology to end host user) and reduce bandwidth just a little

CLI
per port
enter interface config mode
"spanning-tree bpdufilter enable"
the interface will not send BPDU and ignore any BPDU it receive
use with caution because can lead to boardcast storm permanent if wrong port

global (enable on portfast enable port)
"spanning-tree portfast bpdufilter default"

root guard

if enable even it receive the superior BPDU (lower Bridge ID) on that interface the switch will not recognize/accept the new switch as the root bridge, interface will be disable
eg: when connect LAN to other LAN but not want to remain the root bridge
even when set the priority to 0, if connect to the LAN have other root bridge have the same priority => the lower MAC Address will be elect as root bridge => root guard prevent this
protect STP topology by preventing switch from accepting superior
should use on designated port to prevent it become root port
to fix the issue => 1 root bridge have to increase the priority value of their switch => and then the main root bridge can be active again after 20 sec (BPDU max age)



CLI per port

enter interface config mode

"spanning-tree guard root"

Loop guard

if enable even if the interface stop receiving BPDU it will not forwarding and the interfere will be disable

eg: unidirectional link (broken fiber optic cable => can only sent or receive, other direction is broken)

when loop guard-enable port max age timer count down to 0 => it will not become a designated port , instead it enter broken (loop inconsistent) state

should be enable on root and non-designated port to prevent it become designated port

CLI

configure primary root bridge and second root bridge (for back up incase the root bridge fail)

enter conf t mode

"spanning-tree vlan " + vlan ID + "root primary"

"spanning-tree vlan " + vlan ID + "root secondary"

STP Load balancing

if multiple VLAN, shouldnt set the same root bridge or path.

should set diff root bridge instead, for avoid wasted bandwidth

Day 22

Rapid Spanning Tree protocol

default on most Cisco device

Spanning tree version comparable

IEEE	**Cisco version**
Spanning tree version	**Per VLAN spanning tree plus (PVST+)**
the original STP (802.1D)	updated version from 802.1D
all VLAN share 1 STP	each VLAN have separate STP
=> can not load balance	=> can load balance
Rapid STP (802.1w)	**Rapid Per VLAN spanning tree plus (PVST+)**
much faster at converging /ada	updated version from 802.1w
all VLAN share 1 STP	each VLAN have separate STP
=> can not load balance	=> can load balance

Multiple STP (802.1s)

use modified RSTP mechanic

can group multiple VLAN into different instance to perform load balancing

eg: VLANS 1-5 to STP 1 and VLAN 6-10 to STP 2

Rapid Spanning Tree Protocol (RSTP)

a evolution of classic STP

Different:

port cost

Speed	STP Cost	RSTP Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2000
100 Gbps	X	200
1 Tbps	X	20

port state

combine Blocking and Disable state into Discarding state

remove listening state

STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Discarding	NO/YES	NO	NO	Stable
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable

root port

unchange

root bridge is the only switch doesnt have a root port

designated port

unchange

non-designated port

split into 2 separate role

alternate port role

is a discarding port that receive a superior BPDU from another **switch**

function as backup to the root port

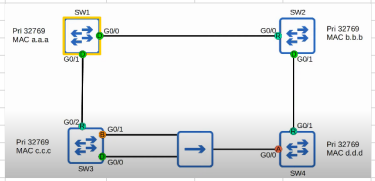
if root port fail => switch can immediately move it best alternate port to forwarding

backup port role

is a discarding port that receive a superior BPDU from another **interface on the same switch**

only happen when 2 interface connected to the same collision domain (via a hub)

function as backup to the designated port



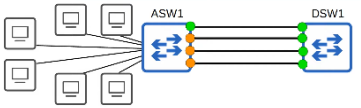
BPDU Type: 0x00: classic STP
0x02: rapid STP
flag classic STP use 2 bit
repid STP use all 8 bit
source: classic STP only root bridge originated BPDU, other just forward it
rapid STP all switches originated and sent their own BPDU from their designated port
age classic STP switch wait 10 hello interval (20 sec)
rapid STP switch wait only 3 hello interval (6 second) it will then flush all MAC addresses learned on that interface

Link type edge: port that connected to an endhost => move directly to forwarding without negotiation => it is the PortFast on classic STP which built in to rapid STP (no more optional feature)
point to point : direct connection between 2 switches => function in full duplex (dont need to configure because auto detected)
shared: connect to a hub => half duplex (auto detected by switches)

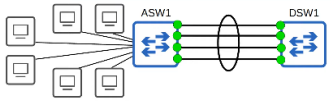
CLI "spanning-tree mode rapid-pvst"

Day 23 EtherChannel aka Port Channel / LAG (Link Aggregation Group)

Problem if a access switch with loads of end host connect to a distribution switch => congestion may happen and add more link/cable will not solve due to STP => 1 interface up only



Solve: EtherChannel act like a portal to group multiple interface to act like a single interface and STP will treat the group as single interface



now all link behave like a single interface
EtherChannel loads balances based on flows (communication between two nodes in the network)

EtherChannel have a algorithm to determine which data flow go with which physical interface

the interface selection calc can be change by using source/destination MAC and source/destination IP to be determine

CLI "show etherchannel load-balance"
to see which being use to calculate the interface selection for load-balance

"port-channel load-balance" + method
eg: "port-channel load-balance src-dst-mac"
manual configure method to calc selection

"show etherchannel summary"
show summary of the etherchannel

EtherChannel configuration: 3 way

PAgP (Port Aggregation Protocol)
for Cisco device
dynamically negotiates the creation/maintenance of the EtherChannel (like DTP for trunks)

LACP (Link Aggregation Control Protocol)
industry standard IEEE 802.3ad
same like PAgP

Static EtherChannel
not using any protocol, interface are statically configured to form an EtherChannel

up to 8 interface can be group in EtherChannel

CLI enable configuration for specific/ range interface
"channel-group 1 mode" + way

Member/interfaces in EtherChannel must have matching configuration
same duplex
same speed
same switchport mode (access/trunk)

same allowed VLAN/native VLAN if is trunk interface

Layer 3 EtherChannel
apply for multiple layer switches
using IP address

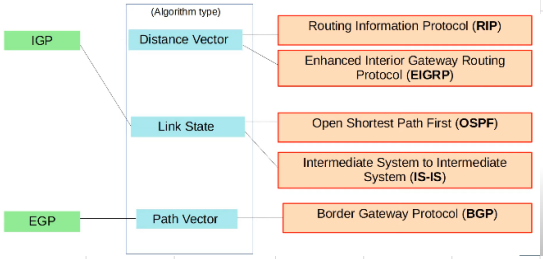
Day 24 Dynamic Routing

Network route
a route to network/subnet (mask length < /32)

Host route
a route to specific address/node (mask length = 32)

Dynamic routing
instead manually configure route like static route => in Dynamic routing, router will advertise to other route that you can reach this subnet/network through me...
if the route have some error => it will dynamic find the next best route
Router will form adjacencies / neighbor relationship with adjacent router to excange this information
if multiple route to a destination are learned, router determine which is superior and add it to the routing table
superior route is the route that have lowest metric (like root cost)

two main categories (type of dynamic routing protocol)
- IGP (Interior Gateway Protocol)
  - use to share route within a single autonomous system (AS) which is a single organization
  - having 2 algorithm (the processes that protocol use to share route information and choose the best route to each destination)
  - Algorithm
    - Distance Vector
      - consist 2 protocol RIP and EIGRP
      - operate by sending the following to their directly connected neignbor:
        - their known destination network
        - their metric to reach them destination network
      - called routing by rumor (router doesnt know about the network beyond it neignbor, only know information through neighbor)
      - only learn the distance (metric) and vector (direction, next-hop router) of each route
    - Link State
      - every router together create a connectivity map of the network
      - each router advertise information about it interface to it neignbor, these advertisement passed along to other route => until all router in the network dev the same map of the network
      - each router independently use this map to calc best route to each destination
      - using more resources (CPU) on the router because more information is shared
      - faster reacting to change than distance vector
- EGP (Exterior Gateway Protocol)
  - use to share route between different autonomous system
  - Algorithm: Path Vector



IGP	Metric	Explanation
RIP	Hop count	Each router in the path counts as one 'hop'. The total metric is the total number of hops to the destination. **Links of all speeds are equal.**
EIGRP	Metric based on bandwidth & delay (by default)	Complex formula that can take into account many values. By default, the bandwidth of the **slowest link in the route** and the total delay of all links in the route are used.
OSPF	Cost	The cost of each link is calculated based on bandwidth. The total metric is the total cost of each link in the route.
IS-IS	Cost	The total metric is the total cost of each link in the route. The cost of each link is **not** automatically calculated by default. All links have a cost of 10 by default.

Metric
- lower metric = superior
- different calc between protocol
- if router learn more than 1 route via the same routing protocol (RIP,EIGRP,OSPF,...) to the same destination with same metric
  - both will be added to the routing table, traffic will be load-balance over both route
  - called ECMP (equal cost multi-path)

Administrative distance (AD)
- in most cases a company will only use a single IGP - usually OSPF or EIGRP
- in same rare case will using 2 (eg: connect network to share information between 2 company)
  - => need something to compare which preferred protocol to be choose between 2 protocol
- lower AD = choosed

Route protocol/type	AD
Directly connected	0
Static	1
External BGP (eBGP)	20
EIGRP	90
IGRP (old ver of EIGRP)	100
OSPF	110
IS-IS	115
RIP	120
EIGRP (external)	170
Internal BGP (iBGP)	200
unusable route	255

eg:

```
10.0.24.0/30 [110/2] via 10.0.12.2, 00:00:09, GigabitEthernet0/0
10.0.34.0/30 [110/2] via 10.0.13.2, 00:00:09, GigabitEthernet1/0
```

[110/2] mean [AD/metric] => AD will be consider first to choose route via lowest (best) protocol and then if there are multiple path learn through that protocol => compare lower metric

Floating static route

although AD of static is 1 which is most superior
=> can manually changing AD of a static route => called floating static route
to make it less preferred than route learned by a dynamic routing protocol to the same destination (make sure AD of static route higher than protocol AD)

Day 25 | RIP & EIGRP

EIGRP
Cisco priorpiary, support unequal-cost load balancing (determine propotion by bandwidth
CLI "router eigrp" + AS (autonomous system)
AS must me match between routers to be form an adjacency
"no auto-summary"
to change prefix length back to class (A,B,C)
"passive-interface" + interface

"network" + network IP address + wildcard mask
to active EIGRP to sending out advertise about route to that network

Wildcard mask
opposite to subnet mask
convert 1 to 0 and 0 to 1 vs subnet mask

Day 26/27/28 | OSPF | Open shortest path first
using link state routing protocol

use the shortest path first algorithm (dijkstra algorithm)

OSPFv2: used for IPv4
OSPFv3: used for IPv6

router store the information about the network in LSAs (Link State advertisement) which are organized in a structure called the LSDB (Link state data base)
router will flood LSAs until all routers in the OSPF area develop the same map of the network (LSDB)
LSAs have aging timer 30 mins -> after 30mins LSAs will be flood again

3 step
1. become neighbor with other router connected to the same segment
2. exchange LSAs with neighbor router
3. calc the best route to each destination and insert them to the routing table

OSPF areas
OSPF use area to divide up the network
small network can be single-area without any negative effect on performance
in larger network => single - area can have negative:
the SPF algorithm take more time to calc (which exponentially)
take more memory on the router
any small change cause every router flood LSAs and run SPF algorithm again

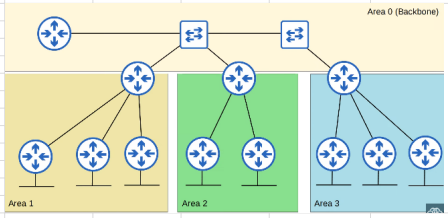
Area is a set of router and link that share the same LSDB

Backbone area (area 0) special area that all other area connected to

router with all interface in the same area called internal router
router with interfaces in multiple area called area border router (ABRs)
ABRs maintain separate LSDB for each area (recommend maximum of 2 area)
router connected to or inside the backbone area (area 0) called backbone routers

interarea route is a route to a destination in a different OSPF area
autonomous system boundary router (ASBR) is an OSPF router that connect the OSPF network to an external network (maybe not using OSPF)

OSPF areas should be contiguous (each member in individual area should be connected and not split eg: Area 3 become Area 1 => error occur)



OSPF interfaces in the same subnet must be in the same area

CLI
"router ospf" + ID
enable ospf protocol, ID is locally significant

"network" + IP address + wildcard mask + "area" + area ID
this command tell OSPF to look for any interface with an IP address contained in the range specified
activate OSPF on the interface in the specified area
the router will then try to become OSPF neighbor with other OSPF activated neighbor route
=> this command only tell router which interface will turn on OSPF, not advertise any

"passive-interface" + interface
command to tell router to stop sending OSPF "hello" message out of the interface
however still continue to send LSAs through other interface informing it neignbor about the subnet configured on the interface

should use this command on interface which dont have any OSPF neighbor

"default-information originate"
creating new LSAs and sending/ flood it

"show ip protocols"
show the routing protocol

router ID order of priority:
manual configuration
highest IP address on a loopback interface
highest IP address on a physical interface

OSPF cost (metric)
calc base on the bandwidth (speed) of the interface
cost = reference bandwidth / interface bandwidth (if <1 => convert to 1)
default reference bandwidth = 100 mbps
eg: interface 10 mbps => cost = 10
interface 100 mbps => cost = 1
interface 1 gbps = 1000 mbps => cost = 1

CLI change reference bandwidth
"auto-cost reference-bandwidth" + mbps

manual config/set cost for a interface
enter that interface config mode
"ip ospf cost" + cost
this will overwrite the cost calc by auto
OSPF cost to a destination = tocal cost of the outgoing/exit interface

OSPF neighbors
when OSPF activate on an interface => sending OSPF hello message out of the interface at regular interval (hello timer = 10s) to introduce the router to potential OSPF neighbor
hello message are multicast to 224.0.0.5 (multicast address for all OSPF router)
the message encapsulate in an IP header, with value of 89 in the protocol field

stage down state
both interface dont know each other is OSPF interface
router will sent out hello package out of the interface
init state
the other router received the hello package
inside have the router ID of the sending router and neighbor router ID (which dont have yet, default 0.0.0.0)

2-way state
other router response with hello package that have both router ID of two
DR/BDR election (designated router / backup designated router)

exstart state
both exchange DBD packet to determine master/slave
master is the one have higher RID (router ID)
the master will start the exchange state

exchange state
sending DBD packet, describe content of LSDB (LSAs)
telling each other what LSAs each one have

Loading state
sending to other router the LSAs that it dont have yet => both replenish to each other the missing other needed
sending LSR (Link state request to request the missing LSAs from neighbor) LSU (link state update to response the LSR to neighbor) LSAck (Link state acknowledgement used to acknowledge that receive a message)

=> both will have same LSDB after this

full state
both have fully OSPF adjaction

dead timer default 40 sec => after receiving hello message if 40 sec going out without receive another hello message => the neighbor be remove and write off from LSDB

CLI
"show ip ospf neighbor"
to show summary of the neighbor

"show ip ospf interface" + interface
to show detail adjacent neighbor in that interface

"show ip ospf database"
to show LSDB

Requirements
Area ID number must match
Interface must be in same subnet
OSPF must not be shutdown
Router ID must be unique
Hello and dead timer must match

authentication seting must be match
OSPF can set up password (for security purpose, same router with OSPF password can be add in the LSDB)
IP MTU setting must match
OSPF network type must match

Loopback interface
is a virtual interface (that can config IP address)
eg: can be use in the case if the connection interface down => other router need IP address to reach the router can use IP address of loopback interface instead to reach out to it through another connection

OSPF network type
is the type of connection between OSPF neighbor (Ethernet,...)
3 main network type
broadcast
enable by default on Ethernet and FDDI (Fiber distributed data interface) interface
point to point
enable by default on PPP (point to point protocol) and HDLC (high-level data linkc control) interface

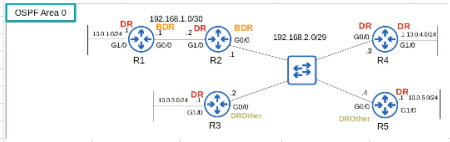
non-boardcast
enable by default on frame relay and X.25 interface

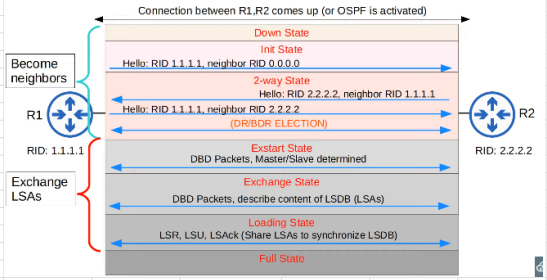
DR/BDR
election order
highest OSPF interface priority (default = 1 for all router interface)
highest OSPF router ID

BDR will become the DR if the current DR is removed

function:
in broadcast network type, router will only form a full OSPF adjacency with the DR and BDR of the segment
therefore router only exchange LSAs with the DR and BDR. DROthers will not exchange LSAs with each other

all router will have the same LSDB but LSAs flooding the network will be lesser => optimize network traffic
broadcast IP address for DR/BDR is 224.0.0.6





CLI
"show ip ospf interface brief"
show brief summary OSPF information
Nbrs F/C
F is number of full adjacency and C is total count of neighbor

Day 29
First hop redundancy protocols (FHRPs)
FHRPs
create redundant/backup default gateway for subnet
protocols use to protect the default gateway used on a subnetwork by allowing two or more router to provide backup for that address
eg: if the default gateway failure/broken => will be another router as backup become the new default gateway (take over the IP address)
by using the same virtual interface IP address and MAC address for routers and set the default gateway as that IP address
routers in FHRPs will elected active and standby one
routers in FHRPs send hello message to eachother to update the status (up/down)
=> if active router broken/failure down => the standby router will take over the virtual address and become active router
standby router will send gratuitous ARP to router to make the switches update their MAC address table => the virtual MAC address will assign to the standby router
gratuitous ARP: ARP reply sent without being requested => to switch update the virtual MAC address to new active router (standby one)

FHRPs are default non-preemptive => if the active one failure and restore the connection => it will not take the role active again, only once the new active router broken it will take the role

include 3 protocol
HSRP (Hot Standby Router Protocol)
Cisco proprietary
active and standby router are elected
two version 1/2
version 2 add IPv6 support and increase the number of group that can be configure

multicast IPv4 address:
v1 = 224.0.0.2
v2 = 224.0.0.102

virtual MAC address:
v1 = 0000.0c07.acXX (XX= HSRP group number)
v2 = 0000.0c9f.fXXX (XXX= HSRP group number)
in situation with multiple subnet/VLAN => can configure different active router in each subnet/VLAN to load balance

VRRP (Virtual router redundancy protocol)
open standard
master and backup router are elected

multicast IPv4 address = 224.0.0.2
Virtual MAC address = 0000.5e00.01XX (XX= VRRP group number)
in situation with multiple subnet/VLAN => can configure different master router in each subnet/VLAN to load balance

GLBP (Gateway load balancing protocol)
Cisco proprietary
load balance among multiple router within a single subnet
AVG (active virtual gateway) is elected and up to four AVFs (Active Virtual Forwarders) are assigned by the AVG
each AVGF act as the default gateway for a portion of the host in the subnet

multicast IPv4 address = 224.0.0.102
Virtual MAC address = 0000.b400.XXYY (XX= GLBP group number, YY = AVF number)

FHRP	Terminology	Multicast IP	Virtual MAC	Cisco proprietary?
HSRP	Active/Standby	v1: 224.0.0.2 v2: 224.0.0.102	v1: 0000.0c07.acXX v2: 0000.0c9f.fXXX	Yes
VRRP	Master/Backup	224.0.0.18	0000.5e00.01XX	No
GLBP	AVG / AVF	224.0.0.102	0007.b400.XXYY	Yes

CLI
configure HSRP
enter interface config mode
"standby"

Day 30 TCP vs UDP

Basic of Layer 4
provide transparent transfer of data between end host
provide various services to applications (apply for TCP)
reliable data transfer
error recovery
data sequencing
flow control ensure source host doesnt sent traffic faster than endhost can handle

provide layer 4 addressing (port numbers which is not reference to the physical interface)
identify the application layer protocol
eg: TCP port 80 = HTTP => mean host want to connect to a website
TCP port 21 = FTP (file transfer protocol)
destination port address will tell about the data transfer protocol and sources port address is create random to distinguish with other connection (eg multi browser tab
destination port range designated by IANA (Internet Assigned Numbers Authority)
well-known port number: 0-1023
registered port number: 1024-49151
ephemeral/private/dynamic port number: 49152-65535 => use for source port address that endhost random generate
provide sessions multiplexing
eg: access multiple tab on browser

TCP (Transmission control protocol)
TCP is connection-oriented
before exchange data, two host communicate to establish a connection. once connection is established => data exchange begin

provide reliable communication
destination host must acknowledge that it received each TCP segment
if a segment isnt acknowledge => it is sent again

provide sequencing
allow destination host to put segment in the correct order

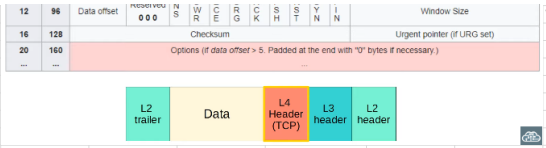
provide flow control
destination host can tell source host to increase/decrease the rate/speed that data is sent

TCP Header

TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
		Reserved			N	C	E	U	A	P	R	S	F																					

Source port and destination port indicate layer 4 port address
Sequence number and Acknowledgement number field provide sequencing and realiable communication
ACK,SYN,FIN flags used to establish and terminate connection



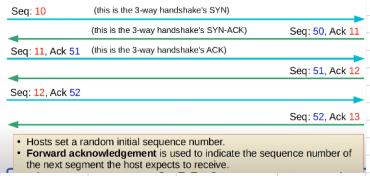
Window size used for flow control

Establishing connection: three-way handshake
- 1st: source host send TCP segment to destination host with SYN flag set = 1
- 2nd: destination reply with SYN and ACK flags set to 1
- 3rd: source host send TCP segment with ACK flag to finish establishing connection

Terminating connection: four-way handshake
- 1st: source send TCP segment with FIN flag
- 2nd: destination reply with ACK flags
- 3rd: destination reply with FIN flags
- 4th: source reply with ACK flags

Sequencing / acknowledgement

- when source start to send TCP segment, the sequence number initial random
- destination reply => with the sequence number initial random too and the ack of source sequence number + increase => meaning the ack is the expected sequence number to receive from source
- and then source sent with sequence number = previous + increase = ack of the destination reply and with the ack = sequence number of destination + increase
- continue the loop



flow control: window size
- allow more data to be sent before and acknowledgement is require
- eg: can accept size 3 => can receive up to 3 seg from source and then start to reply them with the ack
  - eg: receive sequence 10, 11, 12 => reply with ack 13

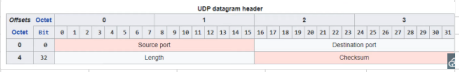
UDP (User datagram protocol)

not connection-oriented
- not establish connection before exchange data => data simply sent right away

not provide reliable communication
- acknowledgement not sent for received segment => if segment is loss/drop, no mechanism to re-transmit it. segment are sent on best effort

not provide sequencing
- no sequence number field in the UDP header. if receive out of order => can not put back in right order

not provide flow control
- has no window size to control flow of data



different
- TCP provide more feature, but the cost of additional overhead => can cause decrease speed
- TCP is preferred for application require reliable communication (eg: downloading file)
- UDP is preferred for application require real-time voice/video

common port number

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

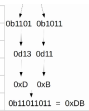
UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

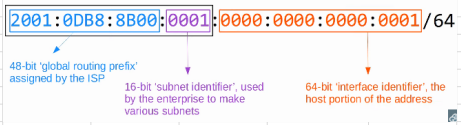
Day 31/32/33 IPv6

Hexadecimal
- 0x prefix to indicate that number is in hexadecimal
- 0b prefix to indicate that number is in binary
- 0d prefix to indicate that number is in decimal
- include [0-9] and [A-F] => 0 to 15
- **each hexadecimal digit contain 4 bit of information**
- convert from binary to hexadecimal

0b11011011 = 0x??

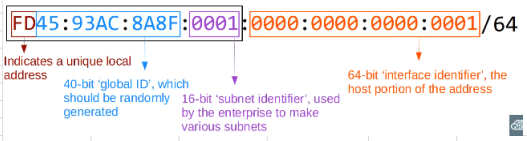


Why IPv6	because IPv4 only have 2^32 address available	
	=> transition to IPv6 for more address available	
IPv6	128 bits = 4 times vs IPv4 => 2^128 address available	
	written in 32 hexadecimal, divided into 8 group of 4 colon	
	using slash to indicate prefix length	
		eg: 2001:0DB4:1241:0014:0BC2:151A:40BD:1412 /64
	shortening IPv6 addresses by remove leading 0	
		eg: 2001:DB4:1241:14:BC2:151A:40BD:1412 /64
	consecutive quartet of all 0 can replace with double colon (::) (1 time only)	



CLI	"ipv6 unicast-routing"	
		to enable IPv6 on the router and allow router to perform IPv6 routing
	enter interface config mode	
		"ipv6 address" + IP address + /prefix lengh
		eg: ipv6 address 2001:db8:0:0::1/64
		to set IPv6 for the interface
		"no shutdown"
		to turn on the interface
	"show ipv6 interface brief"	
		to show brief about IP address in interfaces list
		each interface will have 2 IPv6 address, one is your configure address and one is Link-Local addresses
	"traceroute" + IPv6 address	
		to show the path to the destination go through what router
EUI-64 (Extended unique identifier)	another way to configure IPv6 addresses by using MAC address	
	converting MAC address (48bits) into a 64-bit interface identifier	
	this interface identifier can then become the host portion of a /64 IPv6 address	
	step	divide the MAC address in half
		because MAC address is 48 bit => 2 half of 24 bit
		insert FFFE in the middle
		invert the 7th bit
		if the 7th bit is 0 => convert to 1 and vice versa (1 convert to 0)
		eg: 1234567890AB => 1234 56FF FE78 90AB => 1034 56FF FE78 90AB (this 64 bit hexadecimal will be the host portion/right half of the IP address)
	CLI	enter interface config mode
		"ipv6 address" + IP address prefix + eui-64
		eg: "ipv6 address 2001:db8:0:1::/64 eui-64"
		=> router auto generate the IP address with host portion base on the MAC address
Global unicast addresses	are public addresses which can be used over the internet	
	must register to use them (expected to be unique)	
	define as from 2000:: to 3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF	
Unique local addresses	are private addresses which cannot be used over the internet	
	dont need register, can use freely within the internal network and dont need to be globally unique	
	cant be routed over the internet (ISP will drop any packet go to private/local address) but still can be routed in internal network	
	use addres from FC00:: to FDFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF	
	new update => 8th bit need to set to 1 => first two digit must be FD	

- *The global ID should be unique so that addresses don't overlap when companies merge.



Link local addresses	are IPv6 address automatically generated on IPv6 enable interface (once enable)	
	use CLI "ipv6 enable"	
	use address from (FE80:: to FEBF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)	

standard state that 54 bit after FE80/10 should be all 0 => only start with FE8

generated using EUI-64 rule

these address only use for communication within single link (subnet)
=> router will not route packet with a link-local destination IPv6 address

Multicast address

addresses one to many
1 source to multiple destination (that have join the specific multicast group)
IPv6 not using Broadcast

Purpose	IPv6 Address	IPv4 Address
All nodes/hosts (functions like broadcast)	FF02::1	224.0.0.1
All routers	FF02::2	224.0.0.2
All OSPF routers	FF02::5	224.0.0.5
All OSPF DRs/BDRs	FF02::6	224.0.0.6
All RIP routers	FF02::9	224.0.0.9
All EIGRP routers	FF02::A	224.0.0.10

Scopes — indicate how far the packet should be forwarded

Interface local (FF01): packet not leave the local device, only move between interface to interface in same device

link local (FF02): packet remain in the local subnet, router will not route the packet between subnet

site local (FF05): packet can be forwarded by routers, should be limit to a single physical location (not forwarded over a WAN)

organization-local (FF08): wider in scope than site local (an entire company/organization)

global (FF0E): no boundaries, can be routed over the internet

Solicited - node multicast address — calculate from a unicast address. for NDP purpose

the address begin with a fixed prefix (FF02:0000:0000:0000:0000:0001:FF) + last 6 hex digit of unicast address
eg: 2001:0db8:0000:0001:0489:4eda:073a:12b8
=> FF02::1:FF3A:12B8

Anycast addresses — addresses from one to one of many

IPv6 Header

Fixed header format

Offsets	Octet	0								1								2								3			
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class								Flow Label																	
4	32	Payload Length																Next Header								Hop Limit					
8	64	Source Address																													
12	96																														
16	128																														
20	160																														
24	192	Destination Address																													
28	224																														
32	256																														
36	288																														

version field
length 4 bits
indicate the version of IP that used (v4 or v6)
fixed value of 6 (0b0110 = "0110") to indicate IPv6

traffic class field
length 8 bit
used for QoS (quality of service) to indicate high-priority traffic

flow label field
length 20 bit
used to identify specific traffic flow (communication between a specific source and destination)

payload length field
length 16 bit
indicate the length of the payload (the encapsulated layer 4 segment) in bytes
the length of IPv6 header isnt included because alway 40 bytes

next header field
length 8 bits
indicate the type of the next header (header of the encapsulated segment)
eg: TCP or UDP
same function as IPv4 header protocol field

hop limit field
length 8 bits
value in this field is decremented by 1 by each router that forward it => reach 0 and the packet is discarded

same function as TTL in IPv4

source/destination field — length 128 bits each
contain the IPv6 address of the packet source and destination

NDP — Neighbor Discovery Protocol
used with IPv6 to replace ARP in IPv4

using ICMPv6 and solicited-node multicast addresses to learn the MAC address of other host (ARP in IPv4 use broadcast message)

using two message

Neighbor solicitation (NS) = ARP request = ICMPv6 type 135
This request will have source IP: is the IP of the sender and destination IP: is the solicited-node multicast addess of the receiver
it know the solicited-node multicast address through the IP address destination => convert to solicited-node mul address

- **Source IP**: R1 G0/0 IP
- **Destination IP**: R2 solicited-node multicast address
- **Source MAC**: R1 G0/0 MAC
- **Destination MAC**: Multicast MAC based on R2's solicited-node address

```
> Frame 6: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:01:09:6d:00:08 (ca:01:09:6d:00:08), Dst: IPv6mcast_ff:78:9a:bc (33:33:ff:78:9a:bc)
> Internet Protocol Version 6, Src: 2001:db8::12:3456, Dst: ff02::1:ff78:9abc
> Internet Control Message Protocol v6
```

Neighbor advertisement (NA) = ARP reply = ICMPv6 type 136
reply with the sender IP in source IP and MAC address

- **Source IP**: R2 G0/0 IP
- **Destination IP**: R1 G0/0 IP
- **Source MAC**: R2 G0/0 MAC
- **Destination MAC**: R1 G0/0 MAC

```
> Frame 7: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:02:09:7c:00:08 (ca:02:09:7c:00:08), Dst: ca:01:09:6d:00:08 (ca:01:09:6d:00:08)
> Internet Protocol Version 6, Src: 2001:db8::78:9abc, Dst: 2001:db8::12:3456
> Internet Control Message Protocol v6
```

IPv6 neighbor table — like ARP table

CLI — "show ipv6 neighbor"

another function of NDP allow host to automatically discover router on the local network
using two message for this process
Router solicitation (RS) = ICMPv6 type 133
sent to multicast addess FF02::2 (all router)
ask all router on the local link to identify themselves
sent when an interface is enable/host is connected to the network

Router advertisement (RA) = ICMPv6 type 134
sent ti multicast address FF02::1 (all nodes)
annouce it presence, as well as other information about the linl
these message sent in response RS
also sent periodically, even if the router hasnt received an RS

SLAAC — Stateless address auto-configuration
host will use RS/RA message to learn the IPv6 prefix of the local link (eg: 2001:db8::/64) and then automatically generate an IPv6 address

just like using "ipv6 address" + prefix/prefix-length eui-64"
but just need "ipv6 address autoconfig"
=> NDP using to auto discover prefix length (subnet mask)

DAD — Duplicate address detection
check if other device on the local link are using the same IPv6 address
once IPv6 enable on interface => it perform DAD by using NS and NA to check
it will send NS to it own IPv6 address => if no reply => it the unique, and vice versa (got reply => already exist that IP

IPv6 Static routing
work like IPv4 static routing

IPv6 routing is disable by default => must be enable with "ipv6 unicast-routing"

```
ipv6 route destination/prefix-length {next-hop | exit-interface [next-hop]} [ad]
```

directly attached static route: only exit interface is specified

recursive static route: only next hop is specified

fully specified static route: both specified

Network route:

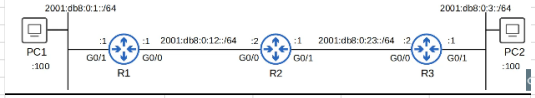
```
R1(config)# ipv6 route 2001:db8:0:3::/64 2001:db8:0:12::2
```

Host route:

```
R2(config)# ipv6 route 2001:db8:0:1::100/128 2001:db8:0:12::1
R2(config)# ipv6 route 2001:db8:0:3::100/128 2001:db8:0:23::2
```

Default route:

```
R3(config)# ipv6 route ::/0 2001:db8:0:23::1
```



Day 34/35 Standard Access Control Lists

ACLs

function as a packet filter, instructing the router to permit or discard specific traffic
can filter traffic based on source/destination IP addresses, source/destination Layer 4 port,...

ACLs are configured globally on the router (global config mode)
and after configured must be applied to an interface (either inbound (enter) or outbound (exit) the interface)
they are an ordered sequence of ACEs

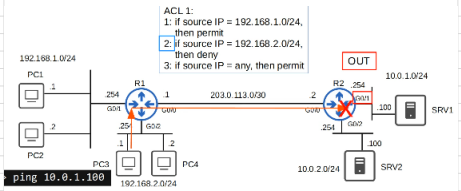
Access control Entries (ACEs)

eg: 1.if source IP = ..., then permit
2. if source IP = ..., then deny

=> order sequence is important
if the packet matches one of the ACEs top to bottom => router take action and stop processing the ACL => all below ACEs will be ignore

a maximum of one ACL can be applied to a single interface per direction
inbound: maximum 1 ACL
outbound: maximum 1 ACL
=> if applied another ACL with the same direction to that interface => will replace the old one

eg: requirement: 192.168.1.0/24 can access to 10.0.1.0/24
192.168.2.0/24 can't access to 10.0.1.0/24
=> best option to apply ACL to the outbound G0/1 of R2



Implicit deny

if a packet doesnt match any of the entries in an ACL
=> router will deny the packet

is like add the last ACEs: if source IP = any, then deny

ACL types

standard IP ACLs: match based on **source IP** address only
should be applied as close to the destination as possible to limit the constant

standard numbered ACLs
are identified with a number
different type have different range of number can be used
standard ACLs can use 1-99 and 1300-1999

standard named ACLs
are identified with a name (instead of group ACLs number)
only config once entering "standard named ACL config mode" to config

extended ACLs: match based on source/destination IP, source/destination port, layer 4 protocol...
extended numbered ACLs
extended ACLs can use 100-199 and 2000-2699

extended name

CLI

```
R1(config)# access-list number {deny | permit} ip wildcard-mask
```

the number is the group ACLs number, the sequence of ACEs will be determine by the command enter order
can use any to replace ip and wildcard-mask because it will = 0.0.0.0 255.255.255.255 (broadcast address)

"access-list" + number +"remark" + remark (## comment here ##)
adding description to the ACEs

"show access-lists"
to show all kind of ACLs
"show ip access-lists"
to show all IP ACLs
"show running-config | include access-lists"
to show all command that set ACLs in the router

enter interface config mode

"ip access-group" + number + "{in | out}"

to apply the ACLs

"ip access-list standard" + ACL name

to enter ACLs named mode (more preferable way than ACLs number because it can edit individual ACEs)

```
R1(config)# ip access-list standard acl-name
R1(config-std-nacl)# [entry-number] {deny | permit} ip wildcard-mask
```

to set up ACLs by name

"no" + sequence number

to remove that ACEs from ACLs

usually 10, 20, 30,...

sequence number + "deny | permit" + IP + wildcard-mask

to insert new entrie (ACEs) in between other entry by specifying the sequence number

"ip access-list resequence" + ACL-ID + starting seg-num + increment

to resequence ACL (increase the ID so can insert ACEs in between)

CLI for Extended ACLs

```
R1(config)# access-list number [permit | deny] protocol src-ip dest-ip
```

config by numbered

```
R1(config)# ip access-list extended {name | number}
R1(config-ext-nacl)# [seq-num] [permit | deny] protocol src-ip dest-ip
```

config by named

common protocol

1: ICMP (Layer 3 by ping command)

6:TCP (Layer 4)

17: UDP (Layer 4)

88: EIGRP (Layer 3)

89: OSPF (Layer 3)

- When matching TCP/UDP, you can optionally specify the source and/or destination port numbers to match.

```
R1(config-ext-nacl)#deny tcp src-ip  eq    src-port-num    dest-ip  eq    dst-port-num
                                     gt                            gt
                                     lt                            lt
                                     neq                           neq
                                     range                         range
```

- **eq 80** = equal to port 80
- **gt 80** = greater than 80 (81 and greater)
- **lt 80** = less than 80 (79 and less)
- **neq 80** = NOT 80
- **range 80 100** = from port 80 to port 100

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

Day 36 | CDP & LLDP | Layer 2 discovery protocol

share information with and discover information about neighboring (connected) device

because they share information about the device in network => security risk so often not used

the information can include IP (although this protocol dont need IP to sent, instead it using MAC address)

Cisco Discovery Protocol (CDP)

enable on Cisco device by default

CDP message are periodically sent to multicast MAC address 0100.0CCC.CCCC

when a dvice receive a CDP message, it processes and discard the message it does not foward it to other device

CDP message sent once every 60 sec

by default CDP holdtime is 180 sec => if a message isnt received from a neignbor for 180 sec, the neighbor is remove from the CDP neignbor table

CDPv2 message sent by default

CLI

"show cdp"

to show the summary of the CDP enable on device

"show cdp traffic"

to show how many CDP packet/advertisement have been sent/received

"show cdp interface"

to show detail for each interface

"show cdp neighbors"

to show CDP neighbor table

"show cdp neignbors detail"

list each CDP neignbor with more detailed information

"show cdp entry" + device name

display detail for specified neignbor only

"cdp timer" + second

to config timer (sent every x secs)

"cdp holdtime" + second
to config hold time (delete if not receive after x secs)

Link layer discovery protocol (LLDP)
industry standard protocol
usually disable on Cisco device by default
LLDP message are periodically sent to multicast MAC address 0180.C200.000E

device can run CDP and LLDP same time
when receive, is discard the message, not forward it
timer = 30 secs (send every 30 secs)
holdtime = 120 secs

CLI "lldp run"
enable LLDP globally

"lldp transmit"
enable sending out on specific interface
have to config Tx and Rx separately
"lldp receive"
enable receiving in on specific interface
have to config Tx and Rx separately
"lldp timer" + second
to config timer (sent every x secs)

"lldp holdtime" + second
to config hold time (delete if not receive after x secs)

"show lldp"
show summary

"show lldp traffic"
show detail

"show lldp interface"
show status of each interface

"show lldp neighbors"
show neighbor table for LLDP

Day 37 NTP (Network time Protocol)
all device have internal clock (router, switch, end host,...)
manually configured clock will drift => resulting inaccurate time
all device need accurate time to have accurate logs for troubleshooting
NTP clients will request the time from NTP server (eg: window, google,..)
a device can be an NTP server and an NTP client at the same time
allow accuracy of time within 1 milisecond if the NTP server is in the same LAN or within 50 milisecond if connecting to the NTP server over a WAN/internet

the distance of an NTP server from the original reference clock called **stratum**

reference clock are stratum 0 within the NTP hierarchy
is a very accurate time device like an atomic clock or GPS clock
NTP server directly connect to referece clock are stratum 1

NTP use UDP port 123 to communicate

software clock = clock
hardware clock = calendar (run with bios battery)
CLI "show clock"
to show time
"show clock detail"
to show time in detail, sources

"show logging"
to show log

"clock set" + time/date
to manually config the time on the device (software clock)

"calendar set" + time/date
to manually config the hardware clock (calendar)

"clock update-calendar" to sync the calendar to the clock time

"clock read-calendar" to sync the clock to the calendar time

"clock summer-time"
to config/adjust the summer-time

"ntp server" + IP address of NTP server
can find NTP server through "nslookup" + NTP server domain (eg: time.google.com) on CLI
to configure connect to NTP server to gather time by NTP clients
"show ntp associations"

to show all NTP server this device connected to and their status

"show ntp status"

to show NTP status

"ntp master" + stratum

to make that device become a NTP server to other device conected to it can sync the time

default stratum is 8

"ntp peer" + IP address of NTP peer

to make them sync to eachother if the NTP server is down/unconnected

Day 38 Domain name system (DNS)

work like a domain converter to IP address

which DNS server will do the convert => endhost will reach to the DNS server to find the IP address of that domain

standard DNS queries/responses will use UDP

TCP is used for DNS message greater than 512 bytes

DNS servers using port 53 and end host (DNS clients) use ephemeral port

device will save the DNS server response to a local DNS cache (dont need to query every time)

a Cisco router can be config as a DNS server (if end host already config the DNS server as on the internet => no need to config DNS server/client on router)

CLI

on end host (computer)

"ipconfig /all"

to display various IP info of the end host

eg: IP, MAC address

"nslookup" + domain

to display IP address for that domain

"ipconfig /displaydns"

to display local DNS cache

"ipconfig /flushdns"

to detele the local DNS cache

on Cisco router

"ip dns server"

to configure router as DNS server

"ip host" + name/domain + IP address

to config list of domain/IP address mapping

"ip name-server" + IP address

config a DNS server that router will query if the requested record isnt in it host table

"ip domain lookup"

to enable DNS queries on router

"show hosts"

to show host/domain mapping table on the router

Day 39 Dynamic Host Configuration Protocol (DHCP)

allow end device (DHCP clients) to auto/dynamic learn various aspects of their network configuration like IP address, subnetmask, default gateway, DNS server,... without manual/static configuration

typically used for client devices (eg: PC, phone,...)

device such as routers, server,... are usually static/manually configured

in small network, router typically act as the DHCP server for host in the LAN

in large network, DHCP server usually a Window/Linux server

DHCP server lease IP address to clients, these lease is temporary => client must give up the address at the end of the lease

to preserve the available IP address (public wifi have loads of unique end host)

eg: find on end host with ipconfig /all

DHCP servers use UDP port 67 and DHCP clients use UDP port 68

DHCP release message

from end-host to DHCP server

to tell server it no longer using/communicating with the DHCP server => remove IP being assigned, other information like subnetmask, default gateway,...

DHCP renew (4 message)

DHCP discover message (from endhost to DHCP server)

ask (broadcast) to find any DHCP server in the network, and ask for end-host IP address (assign by DHCP server/router)

DHCP offer message (from DHCP server to endhost)

reply with a IP address for client to use

include other information like default gateway, DNS server

it also unicast at Layer 3 because already have endhost -IP address (in the message)

DHCP request message (from endhost to DHCP server)

to indicate that end-host want to use that IP address given/offered

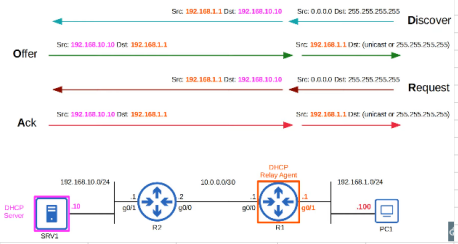
because maybe multiple DHCP server on the network => multiple offer message=> it only choose 1 DHCP to gain the IP address

this message will sent in broadcast (to inform other DHCP server too) that which DHCP server being choose

Discover	Client → Server	Broadcast
Offer	Server →Client	Broadcast or Unicast
Request	Client → Server	Broadcast
Ack	Server →Client	Broadcast or Unicast

DHCP ack (acknowledgement) (from DHCP server to endhost)
to confirm that the IP will be use by end-host and add it to the route table

DHCP relay
big organization often choose using centralized DHCP server (only 1)
so if there are subnet=> it won't receive the DHCP client broadcast (because broadcast not leave the local subnet)
=> can config a router to act as a DHCP relay agent
router will forward the clients broadcast DHCP message to the remote DHCP server as unicast message



CLI
on end host
"ipconfig /release"
to flush/remove the IP address being assign by router/wifi
simply tell router that im done communication
"ipconfig /renew"
to contact the router/DHCP server and gain new IP address
tell router to we need renew information

on router (DHCP server)
"ip dhcp excluded-address" + IP1 - IP2
to specified the IP address range that not be given to DHCP clients
usually for testing + static configurate later
"ip dhcp pool" + pool name
create DHCP pool = a subnet of address that can be assigned to DHCP client, as well as other information like default gateway

"network" + network address + network mask/prefix length
to specified the subnet of addresses to be assigned to clients (except the excluded addresses)

"dns-server" + DNS server IP address
specified the DNS server that DHCP client should use

"domain-name" + domain name
to specified the domain name of the network

"default-router" + IP address
specified the default gateway (normally is the router IP address)

"lease" + days + hours + minutes
to config the lease timelapse

"show ip dhcp binding"
to show binding table (list of IP address assigned end-host/DHCP clients)

on DHCP relay agent router
enter config mode of the interface connected to the subnet of the client device
"ip helper-address" + DHCP server IP address
to assign it as the DHCP relay agent for that subnet

"ip address dhcp"
to config the router as DHCP clients (not recommended, just use static config)
the router interface now will ask IP address from DHCP server

Day 40
Simple Network Management Protocol (SNMP)
is industry-standard framework and protocol which include multiple protocol
RFC 1065 — RFC = Request of comment (a publication in a series from the principal technical development and standards-setting bodies for the Internet, most prominently the Internet Engineering Task Force)
RFC 1066
SNMPv1 — RFC 1067
SNMPv2c — is SNMPv1 and allow the NMS to retrieve large amount of information in a single request => more efficient (GetBulk)
SNMPv3 — **support encryption and authentication**
=> in SNMP v2 below => other can capture SNMP packet and see the password (community string) as plain text

functions: used to monitor the status of devices, make configuration changes,...

two main type of devices in SNMP:
managed devices
devices being managed using SNMP
eg: network device like router/switch, firewall,...

network management station (NMS)

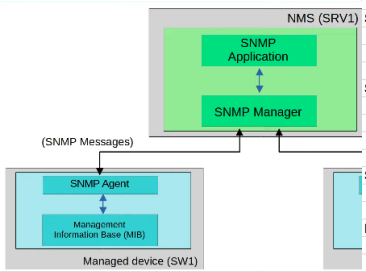
device managing the managed device
this is the SNMP server

3 main operation used in SNMP
managed device can notify the NMS of events
eg: if a managed router interface is down => it can sent notify to NMS

NMS can ask the managed device for their current status
eg: query for usage, flow,..

NMS can tell the managed device to change aspect of their configuration
eg: change IP address of a managed router,...

SNMP component



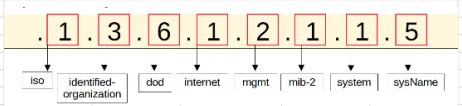
SNMP application provide interface for the network admin to interact with
display alert, statistic, chart,..
eg: SolarWinds

SNMP manager is the software on NMS that interact with the managed device
received notification, send request for information, config change,...
UDP port 162

SNMP agent is the SNMP software running on the managed device that interact with the SNMP manager
sent notification to / receive message from NMS
UDP port 161

Management information base (MIB) is the structure the contain the variables that are managed by SNMP
each variable is identified with an Object ID (OID)
eg: CPU usage, interface status, temperature variable,traffic loads ...

SNMP OIDs
organized in a hierarchical structure



SNMP messages

Message Class	Description	Messages
Read	Messages sent by the **NMS** to read information from the **managed devices**. (ie. What's your current CPU usage %?)	*Get* *GetNext* *GetBulk*
Write	Messages sent by the **NMS** to change information on the **managed devices.** (ie. change an IP address)	*Set*
Notification	Messages sent by the **managed devices** to alert the **NMS** of a particular event. (ie. interface going down)	*Trap* *Inform*
Response	Messages sent in response to a previous message/request.	*Response*

Get: to retrieve the value of 1 or multiple OID, agent will sent response message for each OID
getnext: to discover the available in the MIB
getbulk: more efficient getnext

set: request to change value of 1 or more variables, agent will sent response message with the new value

trap: from agent to manager, manager does not response to acknowledge => unreliable
inform: from agent to manager, and manager will acknowledge through response message => more reliable

response: message sent in response to other message

CLI
on managed device (router, switch)
"snmp-server community" + password + ro/rw
to configure the SNMP password (community string)
ro = read only => can't use Set message
rw = read and write => can use Set message
if enter the ro password => can only read. must enter the rw password for write

"snmp-server host" + NMS IP address + version + password
to config the NMS IP address and version

"snmp-server enable traps snmp linkdown linkup"
"snmp-server enable traps config"
to config the trap type to send to the NMS

Day 41 Syslog

Industry standard protocol for message logging
on network device, Syslog can be used to log event such as change in interface status, change in OSPF neighbor status, system restarts,...

essential when troubleshooting issue, examining the cause of incident

both Syslog and SNMP used for monitoring and troubleshooting of device, they are complementary but different functionalities
Syslog is used for message logging
event that occur within the system are categorized based on facility/severity and logged

message are sent from the device to the server (server cant active pull information from the device (like SNMP get) or modify variable (SNMP Set))
used for system management, analysis and trouble shooting

SNMP used for retrieve and organize information about the SNMP managed device information

Syslog message format

seq:time stamp: %facility-severity-MNEMONIC:description

seq: sequence number indicating the order/sequence of the message
time stamp: indicating time the message was generated
facility: value indicate which process on the device generate the message (eg: OSPF)
severity: indicate the severity (8 level)

Level	Keyword	Description
0	emergency	system is unusable
1	alert	action must taken immediately
2	critical	critical condition
3	error	error condition
4	warning	warning condition
5	notice	normal but significant condition (notification)
6	informational	informational message
7	debugging	debug level

MNEMONIC: short code for the message indicate what happen
description: detail what happen

Syslog logging location

Console line: syslog message will be displayed in the CLI when connected to the device via the console port
by default all message are displayed

VTY line: syslog message will be displayed in the CLI when connected to the device via Telnet/SSH.
disable by default => need manual configuration to operate

Buffer: syslog message will be saved to RAM
by default all message are displayed
"show logging" to show the buffer

external server
device can be configured to send syslog message to an external server
syslog server will listen for message on UDP port 514

CLI

"logging console" + level
logging to the console line (default all already configured) to display the level smaller and equal that input level (higher severity)
eg: logging console 6
to restrict/remove level 7 message

"logging monitor" + level
logging to the VTY line
need to use "terminal minitor" command every time connect to the device via Telnet or SSH

"logging buffered" + buffer size + level
to enable logging saved to RAM (total size in bytes)

"logging" + server IP
to logging to an external server

"logging trap" + level to config the displayed level for external server

"service timestamps log datetime"
to enable timestamps on message

"service sequence-numbers"
to enable seq on message

Day 42 — Secure Shell (SSH)

Console port security

by default no password is needed to access the CLI if connect through console port

can configure a password on the console line
so user have to enter a password to access the CLI via the console port each time connecting

CLI: "line console 0"
enter console line config on the device (router, firewal,...)
there only a single console line (1 connect at a time) => number always 0

"password" + password
configure the console line password

"login"
tell device to require user to enter password when login

"username" + username + "secret" + password
to configure the local username and password on the device
if want to login the device with both username and password => more security

"login local"
tell device to require the username and password local to login

"exec-timeout" + minutes + seconds
to auto exit if inactive in minutes:seconds

Layer 2 Switch management IP
switch dont have IP address but can assign IP address to a Switch virtual interface to allow remote connection to the CLI of the switch (using Telnet or SSH)

CLI on switch configure the IP address on the SVI in the same way as on a multilayer switch

"interface" + "vlan1" (virtual interface name)
to enter the SVI config mode

"ip address" + ip address + subnet mask
assign the IP address
"no shutdown"
enable the virtual interface

"ip default-gateway" + IP address
to configure the switch default gateway to communicate to the router that go to outside the LAN
just like static route to a specific IP address outside the network but just need to indicate default gateway
should be a IP address of the router

Telnet (Teletype Network) is a protocol used to remotely access the CLI of a remote host
being replaced by SSH due to security
if captured => can read the data in plain text (no encryption)

the telnet server (device being connected to like router/switch) listen for Telnet traffic on TCP port 23
telnet client (the device want connecting like end host)

CLI on the telnet server
"enable secret" + password
must set password in order to access privileged exec mode when connecting via Telnet

"username" + username + "secret" + password
to configure the local username and password on the device

can con figure an ACL to limit which device can connect to the VTY line
and apply it to the VTY line to limit the IP that can be use Telnet/SSH
"line vty 0 15"
to enter config mode for all 16 line connect
Telnet/SSH is configured on the VTY line, there are 16 available one => up to 16 user can be connected at once
VTY = Virtual Teletype

"login local"
"exec-timeout" + time

"transport input" + telnet/SSH
to allow telnet/SSH connection (can be both, or none)

"access-class 1 in"

on the telnet client
"telnet" + telnet server IP address + port
to remotely access the CLI with Telnet

Secure Shell (SSH) provide security feature as data encryption and authentication
=> man at the middle cant read the packet
TCP port 22
CLI on SSH server
"show ip ssh"
to check the version, brief information

"ip domain name" + FQDN
FQDN = Fully qualified domain name (host name + domain name)
the FQDN of the device is used to name the RSA keys
RSA = encrypt algorithm with a keys to decrypt
have to configure the hostname and domain name first
to set the RSA keys name (just name not the keys)

"crypto key generate rsa" + size
to generate the key
input the size of the keys in bits

and then can configure like on Telnet command

"enable secret" + password
"username" + username + "secret" + password
can con figure an ACL to limit which device can connect to the VTY line

"ip ssh version 2"
to restrict SSH to version 2 only

"line vty 0 15"
to enter config mode for all 16 line connect

"login local"

must use login local in order to use SSH

"exec-timeout" + time

"transport input" + SSH

to limit only SSH => disable Telnet connection

"ssh -1" + username + SSH server IP address

"ssh" + username@ip-address

use one of those command to connect to the SSH

step to configure SSH

1. configure host name
2. configure DNS domain name
3. generate RSA key pair
4. configure enable PW, username/PW
5. enable SSHv2 (only)
6. configure VTY lines

Day 43 FTP / TFTP

both are industry standard protocol used to transfer file over a network

both use a client-server model

client can use FTP/TFTP to copy file from and to a server

most common use in network engineer is in the process of upgrading the OS of a network device

use to download newer version of IOS from a server and reboot the device with the new IOS image

File transfer protocol (FTP)

listen on TCP port 20 and 21

have authentication (username/password) => server will response to authentication FTP request

no encryption

for greater security FTPS (FTP over SSL/TLS) or SFTP (SSH file transfer protocol) can be used

more complex than TFTP => client can also navigate the file directories and add/remove directories, list file,...

FTP control connection (2 type)

TCP port 21: FTP control connection

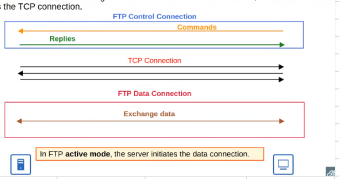
to send FTP command and replies

TCP port 20: FTP data

to send file/data using this port

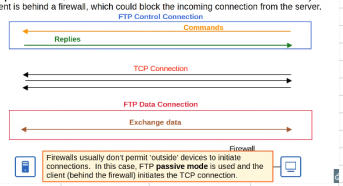
The default method of establishing FTP data connections is **active mode**, in which the server initiates the TCP connection.

Active mode



passive mode

In FTP **passive mode**, the client initiates the data connection. This is often necessary when the client is behind a firewall, which could block the incoming connection from the server.



Trivial file transfer protocol (TFTP)

simple and have basic feature compared to FTP

only allow client to copy a file to or from a server

no authentication (username/password) => server will response to all TFTP request

no encryption so all data is sent in plain text

best used in a controlled environment to transfer small file quicky

listen on UDP port 69

TFTP reliability

although UDP not retransmissions but TFTP have build-in function work like that

every TFTP data message is acknowledged

if client transfering a file to server => server will send Ack message and vice versa

timer are used => if isnt receive ack in time => device will sent the previous message

TFTP connection (3 phase)

1. Connection

TFTP client send request to the server and the server responds back, initializing the connection

2. data transfer

client and server exchange TFTP message, 1 send data and the other send acknowledgement

3. connection termination

after last data being sent, final ack is sent to terminate the connection

TFTP TID (transfer identifier)

to replace the destination port from the 2nd transfer onward

to indentifer the file

IOS file system — is a way of controlling how data is stored and retrieved

CLI
- "show file systems" — view the file system on device
- "show version" — show OS version of the device
- "copy" + source + destination — to start copying file from other host/device/server; eg: copy tftp: flash:
- "show flash" — show file in the device
- "boot system" + filepath — to reboot system with that file
- "write memory" — to write on the running config file
- "reload" — to restart the device and using that file as IOS
- "delete" + filepath — to delete file
- "ip ftp username" + username
- "ip ftp password" + password — configure username/password for FTP device (both client and server)

Day 44/45 — Network address translation (NAT)

Private IPv4 addrsses (RFC 1918)

because IPv4 doesnt provide enough address for all device that need IP address in the modern world
- long term solution : IPv6
- short term solution:
  - CIDR (FLSM, VLSM)
  - Private IPv4 addresses
  - NAT

RFC 1918 specified IPv4 address range as private => can be duplicate between other LAN
- 10.0.0.0/8
- 172.16.0.0./12
- 192.168.0.0/16

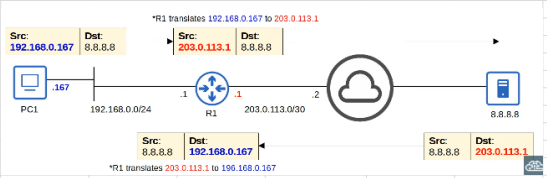
private IP address cannot be used over the internet
- => ISP will drop any traffic to or from private IP address

NAT — used to modify the source/destination IP address of packet (maybe borrow the public IP of the router and used it or config a public IP address to used)

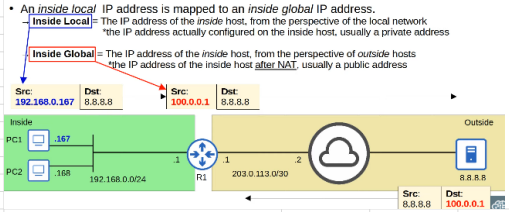
router will translate private IP address to public IP address

allo host with private IP address to communicate with other host over the internet

Source NAT — translate the source NAT from private IP of the end host to the public IP of the router (the one interface connect to the internet) and vice versa



Static NAT — involve statically config one-to-one mapping of private IP addresses to public IP addresses
- an inside local IP address is mapped to an inside global IP address
- outside local and outside global is the same in source nat (only different in destination nat)



each host have mapped to 1 public IP address

because is the same as using public IP right at the beginning for end host => should not used because cant preserve IP address

CLI
- "ip nat inside" — **inside local** — define the inside interface connection to the private network on router
- "ip nat outside" — **inside global**

define the outside interface connection to the internet on router

"ip nat inside source static" + inside private IP + inside global IP
configure/ mapping 1 to 1 private IP to public IP
if the inside global IP already mapped to other inside private IP => the command not overwrite and just do nothing
"clear ip nat translation"
remove all the dynamic translation (not static)
"show ip nat translations"
show static nat table

"show ip nat statistics"
show statistic for nat

Dynamic NAT
router dynamically map inside local address to inside global address as needed
an ACL is used to identify which traffic should be translated
if the souce IP is permitted by hte ACL => source will be translated
if the source IP is denied by the ACL => source will not translated but the traffic will not be dropped

a NAT pool is used to define the avaiable inside global addresses

On R1:
ACL 1: permit **192.168.0.0/24**
deny any
POOL1: **100.0.0.1** to **100.0.0.10**
If a packet with a source IP permitted by **ACL 1** arrives, translate the source IP to an address from **POOL1**.

mapping are still one to one (one inside local IP address per inside global IP addres)
NAT pool enhaustion occur when more inside local IP permitted than inside global IP in pool
if packet from another inside host arrive and need NAT but no available => packet will be drop

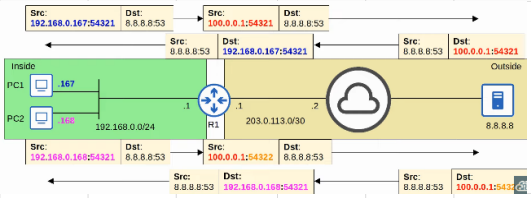
Dynamic NAT entries will time out auto if not used or can clear manually

CLI "ip nat pool" + pool name + inside global IP range + netmask
to config the pool IP range

need to create create ACL to permit/deny IP to reach internet

"ip nat inside source list" + ACL name + "pool" + pool name
config dynamic NAT by mapping ACL to the pool

PAT (NAT Overload) (Port & network address translation)
translate both the IP address and the port number (if necessary)
by using unique port number for each communication flow => single public IP address can be used by multiple internal private IP
the router will keep track of which inside local address is using which inside global address and port



useful for preserving public IP address and being used in network all over the world

CLI config like dynamic NAT, different in config command

"ip nat inside source list" + ACL name + "pool" + pool name + "overload"
config the PAT with manual set public IP address (in pool)

"ip nat inside source list" + ACL name + "interface" + interface + "overload"
config PAT with the public IP address as the outside interface IP (the interface that connect to internet)

Day 46/47 QoS (Quality of Service)
IP Phones
traditional phone operate over the public switched telephone network (PSTN) or Plain old telephone service (POTS)

IP phone use VoIP (Voice over IP) technology to enable phone call over an IP network suck as internet
IP phone connected to a switch like other end host
IP phone have internal 3-port switch
1 port is the uplink to the external switch
1 port is the downlink to the PC
1 port conect internally to the phone itself (inside the phone)

to allow PC and IP phone to share a single switch port, traffic from PC passes through IP phone and then the switch

CLI "Switchport voice vlan" + VLAN ID
config the IP phone and assign it to a VLAN

Power over Ethernet (PoE)

allow power sourcing equipment (PSE) to provide power to the powered device over an Ethernet cable => dont need another electric cable

PSE: switch

PDs: IP Phone, IP camera, wireless access point

normal RJ45 have 8 wire (4 for data transmission) => other 4 (4 5 7 8) for power transmission

Quality of Service (QoS)

voice traffic use PSTN

data traffic use IP network

but in modern network typically converged network in which IP phone, video traffic, regular data... all share the same IP network

same cable => compete for bandwidth

so QoS is set of tool used by network device to apply different treatment/priority to different packet

QoS used to manage the following characteristic of network traffic

Bandwidth

eg: 20% voice traffic, 30% specific kind of data traffic, and 50% for all other traffic

Delay

one-way delay = the amount of time it take traffic to go from source to destination

two-way delay = the amt of time from source to destination and return

Jitter

the variation in one-way delay between packet sent by the same application

"Jitter buffer" used to fixed delay to audio packet

Loss

% packet that not reach their destination

Standard acceptable interactive audio

one-way delay: 150ms or less

Jitter: 30ms or less

Loss: 1% or less

Queuing — by default message will be forwarded in a FIFO order

if network device receive message faster than it can forward them out => pending message place in a queue

tail drop = if the queue is full => new packet will be drop

tail drop can lead to TCP global synchronization (wave of unbalance traffic loads)

TCP sliding window: host using TCP will increase/decrease the rate at which they send traffic as needed

when packet is drop => will re-transmitted and reduce the rate it sent traffic and then increase gradually

tail drop can lead to multiple TCP sliding window => create wave that send loads of traffic => congestion => send little => send loads of traffic => congestion (loop of wave)

Network congestion => tail drop => global TCP window size decrease => network underutilized => global TCP window size increase => network congestion

solution to prevent is Random Early Detection (RED)

when amt of traffic in queue reach a certain threshold => device start randomly dropping packet from select TCP flows

=> avoid the loop of congestion wave

in standard RED => all kind of traffic are treated the same

an improved version Weighted RED (WRED) => allow to control which packet are dropped depending on the traffic class

Classification — to organized network traffic into traffic classes to priority of QoS

many method of classifying traffic

eg: ACL, NBAR (Network based application recognition), PCP (Priority code point) field in dot1q, DSCP (differentiated service code point)

PCP/CoS — Priority code point / Class of Service

3 bit field in the layer 2 header (dot1q) = 8 possible value

PCP value	Traffic types
0	Best effort (default)
1	Background
2	Excellent effort
3	Critical applications
4	Video
5	Voice
6	Internetwork control
7	Network control

the higher value => the higher priority

IP ToS byte (Type of service) — using layer 3 header to priority

IPP (IP precedence) similar to PCP (old one, replaced by DSCP)

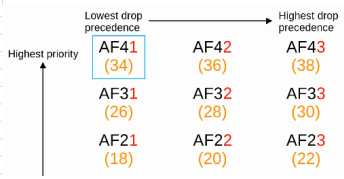
DSCP — RFC 2474

standard marking (type/priority):

default forwarding (DF) - best effort

expedited forwarding (EF) - low loss/latency/jitter traffic (usually voice) DSCP 46

assured forwarding (AF) - set of 12 standard value



Lowest priority | AF11 (10) | AF12 (12) | AF13 (14)

Class selector (CS) - set of 8 standard value, provide backward compatibility with IPP

RFC 4954 offer specfic recommendation for each type of traffic

voice traffic: EF
interactive video: AF4x
streaming video: AF3x
high priority data: AF2x
best effort: DF

Queuing management

common scheduling method is weighted round-robin
round-robin = packet are taken from each queue in order, cyclically
CBWFQ (Class based weighted fair queuing) using weighted round-robin while guaranteeing each queue a certain % of interface bandwidth during congestion

LLQ (Low latency queuing)
designate 1 or more queue as strict priority queue
if there traffic in the queue, the scheduler will alway take the next packet from that queue until it is empty
effective for reducing the delay and jitter of voice/video traffic
downside: starving other queue if there alway trafic in the designated strict priority queue

Shaping / policing
using to control the rate of traffic
shaping buffer traffic in a queue if the traffic rate go over the configured rate
policing drop traffic if the traffic rate go over the configured rate
burst traffic over the configured rate is allow for a short period of time

Day 48 Security Fundamental

CIA triad form the foundation of security
Confidentiality only authorized user should be able to access data
Intergrity data shoulnt be tampered with by unauthorized user
Availability network/system should be operational and accessible to authorized user

Common attack
DoS (denial of service) attack
threaten the availability of a system
a common DoS is TCP SYN flood
TCP three-way handshake: SYN / SYN-ACK/ ACK
attacker send countless TCP SYN message to the target
target reply with SYN/ACK to each SYN it receive
attacker never reply with ACK
=> fill up the target TCP connection table as incomplete
target no longer able to make legitimate TCP connection

DDos (distributed DoS)
the attacker infect many target computer with malware and use them all to initiate a DoS
the group of infected computer called botnet

Spoofing attack
to spoof an address is to use a fake source address (IP or MAC address)
numerous attack involve spoofing
eg: DHCP exhaustion attack
attacker use spoofed MAC addresses to flood DHCP descover message
the target server DHCP pool become full => result in a DoS

Reflection/Amplification attack
attacker send traffic to a reflector and spoofs the source address using the target IP address => reflector (eg DNS server) send the reply to the target IP address
can be use to hide the identical of attacker
amplification attack is when traffic send by attacker is small but trigger a large amt of traffic to be sent from the reflector to the target
eg: DNS amplification ddos and NTP amplification ddos

Man in the middle attack
attacker place himself between the source and destination to eavesdrop on communication or modify traffic before it reach the destination
eg: ARP spoofing/poisoning
a host send ARP request asking for the MAC address of another device
attacker will reply after the legitimate replier with it MAC address
because it arrive later => overwrite the legitimate ARP reply of the legitimate
so attacker can stood between the connection
eg: DHCP poisoning
spurious DHCP server reply to client DHCP discover message and assign them IP address

but make the client used the spurious server IP as the default gateway
this cause the client to sent traffic to the attacker instead of legitimate default gateway
attacker can examine/modify the traffic before forwarding it to the legitimate default gateway
common attack is to combine DHCP starvation with DHCP poisoning.
First, the attacker will send spoofed DHCP discover messages until the pool is completely drained. They will then pose as a DHCP server. This way they can guarantee that the clients will receive addressing from them.

this compromise the confidentiality and integrity of communication between PC1 and SRV1

Reconnaissance attack
used to gather information about target which can be used for later attack

Malware — variety of harmful program that can infect a computer
- Virus — infect other software and spread as the software is shared by user
  - they corrupt or modify file on the target computer
- Worm — not require a host program
  - able to spread on their own without user interaction
  - can congest the network but the payload of a worm can cause additional harm to target device
- Trojan horse — harmful software that is disguised as legitimate software, they spread through user interaction such as opening email attachment or downloading a file from the internet

Social engineering attack
manipulate people
eg: phishing

Password attack
guessing
dictionary attack: brute force with a dictionary or list of common words/password to find

AAA triad
- Authentication — verifying user identity
- Authorization — granting user appropriate access and permission
- Accounting — process of recording the user activity on the system

a framework for controlling and monitor user of a computer system

Day 49 Port Security

is a security featureof Cisco switches
allow to control which source MAC address are allowed to enter the swithport
if unauthorized source MAC address enter the port => action will be taken
eg: place interface in an "err-disabled" state
port security allow network admin to control which device are allowed to access the network
however MAC address spoofing is easy => rather than manually specifying the allowed MAC address on each port => port security ability to limit the number of MAC addresses allowed on an interface is more useful

when enable port security => default setting will be 1 MAC address allow on 1 that interface
can config that MAC address manually or default will be the 1 st source MAC address that enter/plug into the port

can change the number of MAC address allowed

CLI
port security can only enable on access port or trunk port spesificially (dynamic mode cant enable port security)

"switchport mode" +"access"/"trunk"
to enable access or trunk mode

"switchport port-security"
to enable port security in default setting

"switchport port-security" + MAC address
to enable port security and authorized that MAC address

"switchport port-security mac-address sticky"
to enable sticky mac-address (which is not affect by the timer/aging time)

"switchport port-security aging time" + minutes
to config the aging time for that authorized MAC address
- 2 aging type — absolute
  - inactivity
- config with "switchport port-security aging type" + type

"show port-security interface" + interface
show port security information of an interface

"show port-security"
to show brief about all port-security on all interface to get overview

re-enabling an interface manually
enter interface config mode
"shutdown"=> "no shutdown"

re-enabling an interface (ErrDisable recovery) => auto recover after a timer
"show errdisable recovery"
to show status of errdisable recovery reason

psecure-violation = port security

"errdisable recovery cause" + reason

"errdisable recovery interval" + seconds

ErrDisable recovery is useless if not unplug the unauthorized device

Violation modes
3 mode
Shutdown

place it an err-disable state
generate a syslog and/or SNMP message when the interface is disable
violation counter is set to 1
Restrict
switch discard traffic from unauthorized MAC address
interface not disable
generate a syslog / SNMP message each time detect an unauthorized MAC
violation counter is set to 1
Protect
switch discard traffic from unauthorized MAC address
interface not disable
not generate syslog/SNMP and violation counter not set

Day 50 DHCP Snooping

a secure feature of switches that used to filter DHCP message received on untrusted port
only filter DHCP message
all port are untrusted by default

uplink port = port that toward the router on switch
downlink port = port that toward the end host on switch

usually uplink port are configured as trusted port and downlink is untrusted

used to protect from DHCP starvation and DHCP poison attack

Operation: if DHCP message is received on untrusted port => inspect it and act as follow
- if it a DHCP server message => discard it (because DHCP server should be sent from trusted port)
- if it a DHCP client message perform the following
  - if Discover/Request message => check if the frame source MAC address and the DHCP message CHADDR field match
    - if match => forward
    - mismatch => discard
      - but attack still can spoof the CHADDR to match the MAC address
  - if Release/Decline message => check if the packet source IP address and the receiving interface match the entry in the DHCP snooping binding table
    - if match => forward
    - mismatch => discard

Rate-Limiting: use to limit the rate at which DHCP message are allowed to enter an interface
- in crosses the threshold => interface is err-disable
- useful to protect against DHCP exhaustion attack

DHCP option 82 (Information option)

is the DHCP relay agent information option
provide additional information about which DHCP relay agent receive the client message on which interface, VLAN,...
DHCP relay agent can add option 82 to message they forward to the remote DHCP server
with DHCP snnoping enable => by default will add option 82 to all DHCP message they receive from client => switch can wrongly discard message with that extra option 82 => should be disable

CLI:
- "ip dhcp snooping"
  - enable globally dhcp snooping (mandatory to config each VLAN)
- "ip dhcp snooping vlan" + VLAN ID
  - to enable dhcp snooping for specific VLAN
- "no ip dhcp snooping information option"
  - disable DHCP option 82
- enter interface config mode
- "ip dhcp snooping trust"
  - to assign the interface as trusted port
- "ip dhcp snooping limit rate" + second
  - to enable rate-limiting DHCP message
- "show ip dhcp snooping binding"
  - to show the binding table
- "errdisable recovery cause dhcp-rate-limit"
  - to enable the auto recovery err-disable on rate-limit reach

Day 51 Dynamic ARP inspection

a security feature of switches that used to filter ARP message received on untrusted port
only filter ARP message
all port are untrusted by default

usually all port connected to other network device (switch, routers) should be config as trusted and connect to end host remain untrusted

to protect/prevent ARP poisoning

Operation: it will inspect the sender MAC and IP field of ARP message received on untrusted port and check that is there a matching entry in the DHCP snooping binding table
- if match => forward
- mismatch => discard

ARP ACLs can be manually configured to map IP address/MAC address for Dynamic ARP inspection to check
- useful for host that dont use DHCP

support rate-limit to prevent overwhelming switch CPUs with ARP message
- allow burst interval variable => allow to configure rate limit as x (rate) packet per y (burst interval) seconds
- if ARP message received faster than the limit rate => interface will be err-disabled

CLI:
- "ip arp inspection vlan" + VLAN ID

to enable dynamic arp inspection on that VLAN

enter config mode for specific interface
"ip arp inspection trust"
to set that port to be trusted => not get inspect/check when data receive from that port

"ip arp inspection limit rate" + x packet + "burst interval" + y second
config the rate limit as x packet in y second

"show ip arp inspection interfaces"
to show dynamic arp inspection status table with all interface listed

"ip arp inspection validate" + validate types to add additional check (can add multiple)
dst-mac enable validation of the destination MAC address (layer 2 header) vs the target MAC address in the ARP body for ARP responses
device classified packet with different MAC address as invalid and drop them

ip enable validation of the ARP body for invalid and unexpected UP address (eg: 0.0.0.0, 255.255.255.255, multicast address)
check the sender IP address in all ARP request and response and check the target IP address only in ARP response

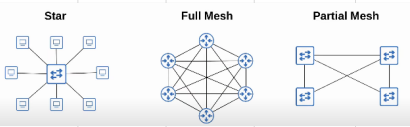
src-mac enable validation of the source MAC address (layer 2 header) vs the target MAC address in the ARP body for ARP request/responses

"do show running-config | include validate"
to display config for optional/additional validate

"show ip arp inspection"
to show the summary of the dynamic arp inspection configuration

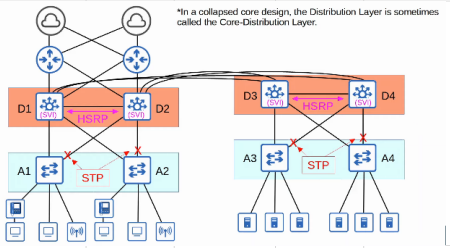
Day 52 LAN Architectures

common terminology
star topology: when several device all connect to one central device
full mesh: when each device is connected to each other device
partial mesh: when some device are connected to each other but not all



Common LAN design

Two-Tier campus LAN design
aka collapsed core design (Because it omit a layer that found in the 3-tier design: core layer)
consist of two hierarchical layer
access layer
layer that end host connected to (pc, printer, camera)
QoS marking, port security, DAI,... typically done here
switchport might be PoE-enable for wireless AP, IP phone,...
eg: access layer switches
distribution layer (aka aggregation layer)
aggregate connection from the access layer switches
typically is the border between layer 2 vs layer 3
connect to service like internet, WAN

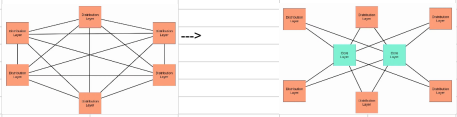


Full mesh: connect between distribution layer

Partial mesh: connection between access layer vs distribution layer

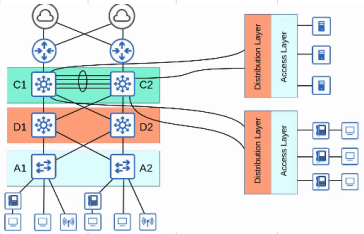
star: connection from end host to access layer

Three-tier campus LAN design
in large LAN network with many distribution layer switches => if full mesh apply will need a loads of connection require
=> recommend adding a core layer if there more than 3 distribution layer in a single location



consist of 3 layer:
access layer
distribution layer
core layer connect distribution layer together in large LAN network
the focus is speed (fast transport)
CPU-intensive operation such as security, QoS marking,... should be avoid at this layer
connection are all layer 3 (no spanning tree)

should maintain connectivity throughout the LAN even if device fail



Spine-leaf architecture — aka Clos architecture (Cisco ACI architecture)

use for Data center (dedicate space/building used to store computer system such as server and network device)

traditional data center design used a three-tier architecture (access-distribution-core)

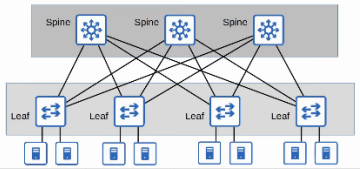
work well when most traffic is North-South (mean from end host to server and vice versa / not go horizontal like from end host to end host)

rule

every leaf switch is connected to every spine switch and vice versa

leaf switches not connect to each other and spine switchs not connect to spine switch

end host only connect to leaf switch



the path taken by traffic is randomly chose to balance the traffic load among the spine switches

each server is separate by the same number of hops => providing consisten latency for East-West traffic

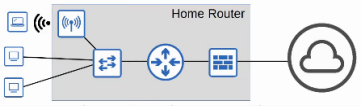
avoid bottleneck in bandwidth and reduce latency of traffic vs 3 tier layer

SOHO (Small office / Home office) network

few device (eg: really small company, house)

all network function are typically provided by a single device, often called a home router or wireless router

that device can serve as a router/switch/firewall/wireless access port/ modem



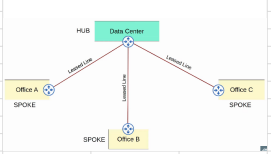
Day 53 — WAN Architecture

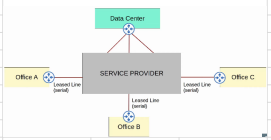
WAN (Wide area network) — a network that extend over a large geographic area

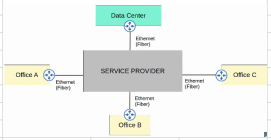
WAN over dedicated connection (Leased Line)

Hub: the central site (usually Data center)

Spoke: the LAN connect to Hub







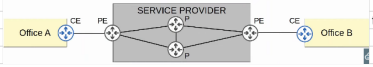
Leased Line — dedicated physical link, typically connecting 2 site

use serial connection (PPP or HDLC encapsulation)

various standard that provide different speed and available in different country
due to higher cost, installation lead time and slower speed of leased line => Ethernet WAN technology is more popular

Multi Protocol Label Switching (MPLS)
similar to the internet, service providers MPLS network are shared infrastructure because many customer enterprise connect to and share the same infrastructure to make WAN connection
allow VPNs to be created over the MPLS infrastructure through the use of labels (instead of IP address)

CE router = Customer Edge router
PE router = Provider edge router
P router = provider core router

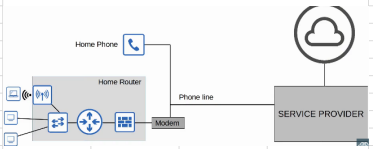


the service provider act like a huge switch for WAN connection of each customer

when the PE router receive frame from CE router, they add label to the frame => to create private flow for each customer (label work like IP address but only apply in the MPLS
only PE/P router using MPLS

many different technology can be used to connect to a service provider MPLS network for WAN service

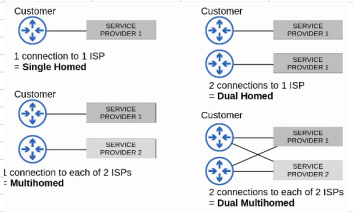
Digital Subscriber Line (DSL) type of connection
provide internet connectivity to customer over phone line, can share the same phone line with that already install in most home



Modem (Modulator-Demodulator) is device to convert data into a format suitable to be sent over the phone line

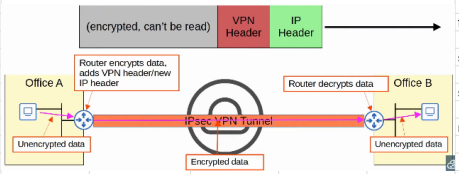
Cable internet type of connection
provide internet access via the same CATV (Cable television) line used for TV
a modem being use too to convert data into format that suit the CATV cable/line

Redundant Internet Connection
Single Homed = 1 connection to 1 ISP
Dual Homed = 2 connection to 1 ISP
Multihomed = 1 connection to 2 ISPs
Dual multihomed = 2 connection to each of 2 ISPs



Virtual Private Network (VPN)
when using internet as a WAN to connect site together => no built-in security by default
to provide secure communication over the internet, VPN is used
2 type of internet VPNs (maybe more)

Site-to-Site VPNs (IPsec)
is a VPN between two device and is used to connect two site together over the internet
a VPN tunnel is created between two device by encapsulating the original IP packet with a VPN header and a new IP header
using IPsec for encrypted the original packet before being encapsulate with the new header



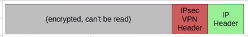
first sending device combine the original packet and session key (encryption key) and run them through encryption formula

sending device encapsulate the encrypted packet with a VPN header and new IP header

sending device send the new packet to the device on the other side of the tunnel

receving device decrypt the packet and forward it to destination

VPN tunnel only formed between 2 tunnel endpoint (eg: router)
all other device dont need to create a VPN for themselves
use to permanently connect two site over the internet
Limitation: not support broadcast and multicast traffic, only unicastm that mean routing protocol such as OSPF cant be used over the tunnel
but can be solved with GRE over IPsec
GRE (Generic Routing Encapsulation) create tunnel like IPsec but not encrypt the original packet but can encapsulate broadcast and multicast message
combine GRE and IPsec = GRE over IPsec
original packet will be encapsulate by GRE header and new IP header and then be encrypt and encapsulate with IPsec VPN header and new IP header

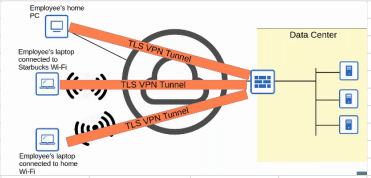


take time to config full mesh of tunnel between many site
but can be solved with Cisco DMVPN
DMVPN (Dymanic multipoint VPN) allow router to dynamically create a full mesh of IPsec tunnel without having to manually configure every single tunnel
just need to configure IPsec tunnel to a hub site
and then the hub router will automatically give each router information about how to form an IPsec tunnel with the other router

Remote-Access VPNs
use to allow end device (PC, phone) to access the company internal resource securely over the internet

typically use TLS (transport layer security)
TLS provide security for HTTPS too
TLS formerly known as SSL (Secure socket layer) and developed by netscape and globalized by IETF with TLS name

VPN client software (eg: Cisco AnyConnect, Sophos Connect) is installed on end device (eg: wfh laptop)
these end device then form secure tunnel to one of the company router/firewall acting as a TLS server
allow end user to securely access resources on the company internal network without being directly connected to the company network

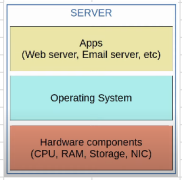


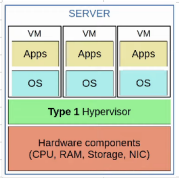
use to provide on-demand access for end device that want to securely access company resource while connected to a network which is not secure

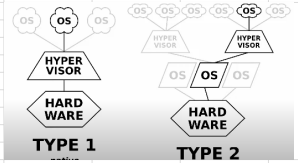
Day 54 Virtualization & Cloud

Virtualization

before virtualization, there only 1 to 1 relationship between physical server and OS
only 1 app providing services such as web server, email server,... can run
virtualization allow to break it and allow multiple OS to run on a single physical server
pros
- Partitioning: divide system resource between VM and run multiple OS
- Isolation: provide fault and security isolation at hardware level, preserve performance with advanced resource control
- Encapsulation: save entire state of a VM to file, move and copy VM easily
- Hardware Independence: provision or mitigate any VM to any physical server



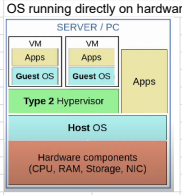




each instance called VM (Virtual Machine)
a hypervisor (Virtual Machine Monitor / VMM) is used to manage and allocate the hardware resource (CPU, RAM,...) to each VM
the hypervisor run directly on top of the hardware called Type 1 hypervisor / bare-metal hypervisor / native hypervisor
- use in data center
- eg: Microsoft Hyper-V, VMware ESXi

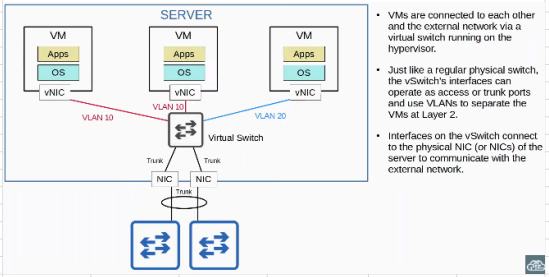
type 2 (hosted) hypervisor run as a program on an OS like a regular computer program
- use for personal-use device
- eg: Oracle VirtualBox, VMware workstation

OS running directly on hardware called host OS and OS running on VM called Guest OS



Connect VMs to the network

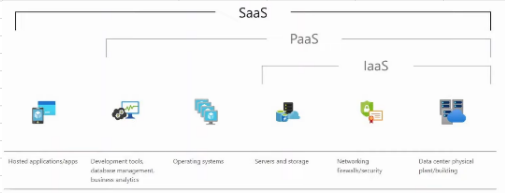
VMs are connected to each other and external network via a virtual switch running on the hypervisor



Cloud services

Traditional IT infrastructure deployment combine:
- On-premises: all infrastructure located on company property
- Colocation: data center that rent out space for customer to put their infrastructure

Cloud computing
model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (eg: network, server, stoage, application and services)
that can be rapidly provisioned and released with minimal management effort or service provider interaction
compose of
5 essential characteristics
on-demand self-service
consumer can unilaterally provision computing capability as needed automatically without requiring human interaction with each service provider
able to use the service freely without direct communicate to the service provider
eg: set up server on AWS => no need to human interaction, just do it by your own
broad network access
capability are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms
services available through standard network connection and be accessed through many kind of device
eg: access AWS with multiple device
resource pooling
pool of resource provided by the service provider, when customer request a service => the resources to fulfill that request are allocated from shared pool
eg: storage drive, server,.. in data center pool of a provider like AWS
rapid eslasticity
elastically provisioned and released. the capability available can be scale rapidly and appropriate in any quantity at any time
customer can quickly expand the service they use in the cloud from a pool of resource that appear to be infinite and can reduce service when not needed
eg: add new VMs, expand storage,...
measured service
cloud system automatically control and optimize resource use by leveraging a metering capability that appropriate to the type of service. and that resouce can be monitored, controlled and reported
cloud service provider measure the customer usage of cloud resource and customer can measure their own use, customer charged based on usage
eg: AWS charging ... dollars per GB of storage per day
3 service model
in cloud computing everything provided on a service model
Software as a Service (SaaS)
the capability provided to the consumer is to use the provider application running on a cloud infrastructure
the app are accessible from various client, device through thin client interface (web browser,..) or a program interface the consumer does not manage or control the underlying cloud infrastructure including network, server, OS, storage...
eg: microsoft office 365



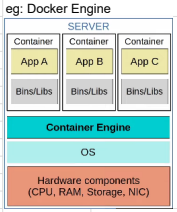
the service provider in control of everything as above image
end user simply use the app
Platform as a service (PaaS)
capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired application created using programming languages, library, service and tool support by the provider
consumer not manage or control the underlying cloud infrastructure but can control over the deployed application and possibly configuration setting for the application hosting enviroment
the service provider offer a platform for dev to use to make application
eg: AWS lambda, Google app engine
Infrastructure as a service (IaaS)
capability provided to the consumer storage, processing, storage, network, and other fundamental computing resource where the consumer is able to deploy and run arbitrary software like OS and app
consumer not manage or control underlying cloud infrastructure but can control over OS, storage, deployed application and possibly limited control of select networking component like firewall
eg: VMs on AWS (Amazon EC2), Google compute engine
4 deployment model
Private cloud
the cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumer (eg: business unit).
it maybe owned, managed and operated by the oranization, or third party, or combination of them
maybe on or off premises
eg: AWS provide private cloud service for American DoD
Community cloud
provisioned for excludesive use by specific communit of consumer that have shared concern (eg: mission, policy, compliance consideration,..)
it maybe owned, managed and operated by the oranizations in the community, or third party, or combination of them
maybe on or off premises
Public cloud
provisioned for open use by general public
maybe owned, managed and operated by a business, academic, government organization or combination
only exist off premises (located on premises of the cloud provider only)
eg: AWS, Azure, GCP, OCI, IBM cloud, Alibaba cloud
Hybrid cloud
composition of two or more distinct cloud infrastructure that remain unique entity but bound together by standardized or proprietary technology that enable data and pplication portability
any combination of the other 3 model
eg: a private cloud which can offload to a public cloud when necessary

Benefit
reduce cost of hardware and software,..
global scale
speed/agility
provided on demand
productivity
remove time consuming task in physical data center
reliability
backup easy, can be mirrored at multiple site in different geographic location to support disaster recovery

Connecting to cloud resouces (public cloud)
through private WAN service provider
through internet (less secure)
through IPsec VPN tunnel

Containers
are software package that contain an App and all dependencies (Binaries/Libraries) for the contained app to run
multiple app can be run in a single container but usually 1 app for each container

run on a container engine (which run on a host OS usually Linux)
container are lightweight and include only what dependencies required to run specific app
a container orchestrator is a software platform for auto the deployment, management, scaling,... of container
eg: Docker Engine



different between VMs
container run directly on top a shared OS while each VMs run with it own OS
container can boot up in miliseconds while VMs take minutes
container take very few space (size in Mb) while VMs take large (Gb)
container use fewer CPU/RAM than VMs
container more portable and can be run on any container service while VMs can only move between physical system running the same hypervisor
VMs are more isolate because each VMs run each OS => if 1 OS crash other still operate

Virtual Routing & Forwarding (VRF)
work like VLAN but for router
used to divide single router into multiple virtual router
allow a router to build multiple separate routing table
interface (layer 3 only) & route are configured to be in a specific VRF
router interface, SVIs and routed port on multilayer switches can be configured in a VRF
traffic in one VRF cannot be forwarded out of an interface in another VRF

VRF commonly used by service provider to allow one device to carry traffic from multiple customer
each customer trafffic is isolated from the other
customer IP address can overlap without issue

CLI
"ip vrf" + VRF name
to create a VRF
"do show ip vrf"
show VRF table

enter interface config mode
"ip vrf forwarding" + VRF name
to assign that interface to a VRF

"show ip route vrf" + VRF name
to show routing table of a VRF

Day 55 Wireless Fundamental
IEEE 802.11 wireless LANs
Wi-Fi alliance test and certifiesequipment for 802.11 standard compliance interoperability with other device

Issues
all device within range receive all frame (like connected to an Ethernet hub which flood out the frames)
CSMA/CA (carrier sence multiple access with collision avoidance) is used to facilitate half-duplex communication
when using CSMA/CA a device will wait for other device to stop transmitting before it transmits data itself
radio frequency is regulated by various international and national body to allow transmit data
signal coverage area must be considered
signal range
signal absorption: happen when wireless signal passes through material and converted into heat, weaken the original signal
signal reflection: when a signal bouces off a material (eg: metal) and reduce the signal received
signal refraction: when wave is bent when entering a substance where it travel at different speed (eg: water)
signal diffraction : when wave encounter an obstacle and travel around it and result blind spot behind the obstacle
signal scattering: when material cause a signal to scatter in all direction (eg sand, dust,..)

Radio frequency
to send wireless signal, sender apply an alternating current to an antenna => create electromagnetic field which propagate out as wave
electromagnetic wave can measure in multiple way like amplitude and frequency
amplitude is maximum strength of the electric and magnetic field
frequency measure the number of up/down cycle per a given time (eg: Hz/Hertz)
Hz is number of cycle in a second
visible frequency range is from 400 THz to 790 THz
radio frequency range is from 30 Hz to 300 GHz
Wireless LAN use ultra high frequency and super high frequency range with 2 main bands
2.4 GHz band: range from 2.4 GHz to 2.4835 GHz
provide further reach in open space and better penetration of obstacle
5 GHz band: range from 5.150 GHz to 5.825 GHz
i

Channels
each band is divided up into mutiple channels => device are configured to transmit and receive traffic on 1 or more channel
in large WLAN with mutiple access point => important that adjacent APs not using overlapping channel to avoid interference
2.4 Ghz band => using channel 1, 6 , 11
5 GHz band => consist of non-overlapping channel

Standard	Frequencies	Max Data Rate (theoretical)	Alternate Name
802.11	2.4 GHz	2 Mbps	
802.11b	2.4 GHz	11 Mbps	

802.11a	5 GHz	54 Mbps	
802.11g	2.4 GHz	54 Mbps	
802.11n	2.4 / 5 GHz	600 Mbps	'Wi-Fi 4'
802.11ac	5 GHz	6.93 Gbps	'Wi-Fi 5'
802.11ax	2.4 / 5 / 6 GHz	4*802.11ac	Wi-Fi 6'

Service set

are group of wireless network device

3 main type

- Independent
- Infrastructure
- Mesh

all device in a service set share the same SSID (service set identifier)

- SSID is human-readable name which identifies the service set
  - is the wifi name

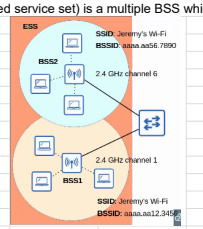
IBSS (idependent basic service set) is wireless network which two or more wireless device connect directly without using access point

- called ad hoc network
  - eg: Airdrop

BSS (basic service set) kind of infrastructure service set which client connect to each other via an access point but not directly to each other

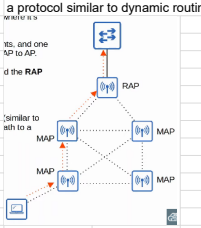
- BSSID is used to uniquely indentify the AP
- other APs can use the same SSID but not the same BSSID
  - is the MAC address of AP radio
- area around an AP where is signal is usable called BSA (basic service area)

ESS (extended service set) is a multiple BSS which AP connected through Ethernet (router/switch) (infrastructure type)



MBSS (Mesh basic service set) use 2 radio 1 to provide a BSS to wireless client and 1 to form a backhaul network which used to bridge traffic from AP to AP

- used in situation where is difficult to run an Ethernet connection to every AP
- at least 1 AP connected to the wired network called RAP (root access point)
- other called MAP (mesh access point)
- a protocol similar to dynamic routing to determine the best path



Distribution system

most wireless network are not standalone network => there is a way for wireless client to connect to the wired network infrastructure

the upstream wired network is called DS (distribution system)

each wireless BSS or ESS is mapped to a VLAN in the wired network

each WLAN is mapped to separate VLAN and connected to the wired network via a trunk

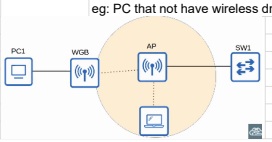
Additional AP operational mode

AP can use another AP to be in repeater mode => to extend the range od a BSS

the repeater AP will operate like a bridge (only retransmit data)

workgroup bridge (WGB) operate as a wireless client of another AP => can be use to connect wired devce to the wireless network

- eg: PC that not have wireless driver card have to use this

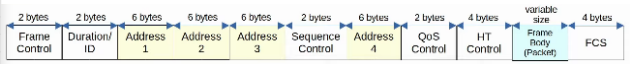


outdoor bridge to be use antenna that focus most of the signal power in one direction which allow wireless connection tobe made over longer distance than normally possible

Day 56 Wireless architecture

802.11 message/frame format

different than Ethernet frame



frame control: provide information about message type and subtype

duration/ID indicate TTL and ID

addresses: up to 4 address can be present

- destination address: final recipient of the frame
- source address: original sender of the frame
- receiver address: immediate receiver of the frame
- transmitter address immediate sender of the frame

sequence control: use to reassemble fragment and eliminate duplicate frame

QoS control: prioritize certain traffic

High Throughput control to enable high throughput operation

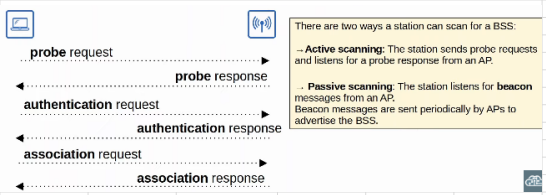
FCS (frame check sequence) check for error in frame

802.11 association process

3 connections state

- not authenticated, not associated
- authenticated, not associated
- authenticated and associated

the station must be in the last state with AP to be send traffic through it



Message type

3 type

Management: to manage the BSS

- beacon
- probe request/response
- authentication reqeust/response
- association request/response

Control: to control access to the medium (radio frequency) assist with delivery of management and data frame

- request to send (RTS)
- clear to send (CTS)
- ACK

Data: contain actual data packet

AP deployment method

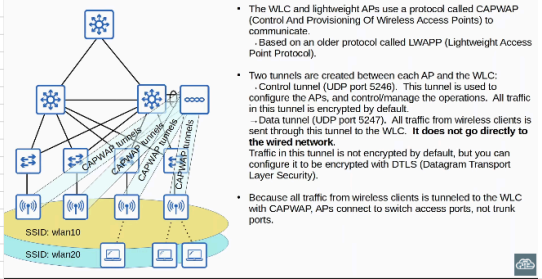
3 main method

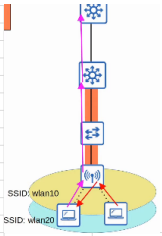
Autonomous APs

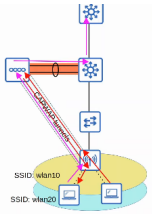
- autonomous AP are self contained system that dont rely on a WLC (wireless LAN controller)
  - configured individually by console cable (CLI), telnet/SSH or HTTP/HTTPS web connection)
- aotunomous AP connect to the wired network with a trunk link
- data traffic from wireless client has a very direct path to the wired network or to other wireless client associated with the same AP

Lightweight APs

- using AP and WLC to split function
  - AP will handle real time operation like transmitting, encryption/decryption and sending out beacon,probe,...
  - WLC will manage RF, security, QoS, client authentication, client association/roamming management, configure the lightweight AP
- called split-MAC architecture
- WLC can be located in same or different subnet/VLAN vs lightweight AP
- each authenticate each other using digital certificated installed on each device





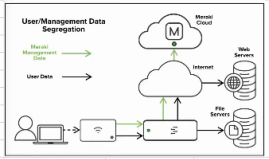


Autonomous AP | Lightweight APs traffic

lightweight APs can be configured to operate in various mode
- local: the default
- flexconnect: like the default but if the connection between WLC vs switch is down => operate like autonomous
- sniffer: not offer BSS for client, just capturing 802.11 frame and sending them to device running sniffer software like wireshark
- monitor: the AP not offer BSS for client, just receiving 802.11 to detect rogue device ->send de-authentication message to disassociate with that rouge device
- rogue detector
- SE-connect (spectrum expert connect)
- bridge/mesh
- flex plus bridge

WLC deployment
- 4 main model
  - unified: WLC is a hardware appliance in a central location of the network
  - cloud-based: WLC is a VM running on a service in a private cloud in a data center
  - embedded: WLC is intergrate within a switch
  - mobility express: WLC intergrated within an AP

Cloud-based AP
- in between autonomous AP and lightweight AP
- is the autonomous AP that centrally manage in the cloud (by a cloud server like Cisco Meraki)

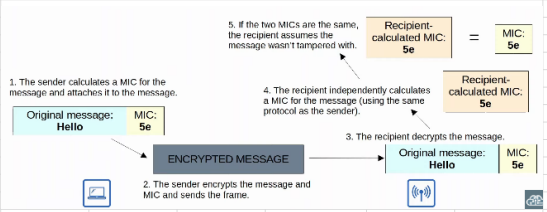


Day 57 | Wireless security

Authentication
- all client must be authenticated before can associate with an AP
- in corporate setting a guest users can use separated SSID which not have access to the corporate network (only the internet)

Encryption
- traffic sent between client and AP must be encrypted because any device can received the frame not like wired network
- all device on WLAN will use the same protocol, however each client will use a unique encryption/decryption key so other device cant read it traffic
- a group key can be use to send to all of it client

Integrity
- ensure the message not be modified (man in the middle attack)
- a MIC (message integrity check) is added to help protect integrity



- if detect packet being modified => discard it

Authentication method

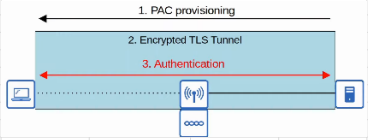
Open authentication
- client send and authentication request => AP accept it (no questions asked)
- not a secure authentication method
- usually combine with other method like have to go to the browser to verify...

WEP (wired equivalent privacy)
- used to provide both authentication and encryption
- for encryption is use RC4 algorithm
- is a shared key protocol => both client and AP must have the same key
  - key can be from 40 bit or 104 in bit length (combined with 24 bit Initialization Vector to modify to 64 bits or 128 bit
- is not secure and easily be cracked

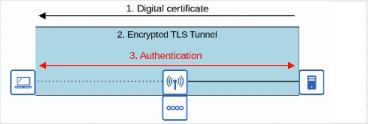
EAP (extensible authencation protocol) is a authentication framework
- it define a standarf set of authentication function that used by various EAP method (LEAP, EAP-FAST, PEAP, EAP-TLS)
- integrate with 802.1X to limit network access for client connected to LAN or WLAN until they authenticate

LEAP (Lightweight EAP)
- client must provide username and password to authenticate
- in addition mutual authentication is provide by both client and server sending a challenge phrase to each other
- dynamic WEP is used => WEP key changed frequently
- is vulnerable and should not be used

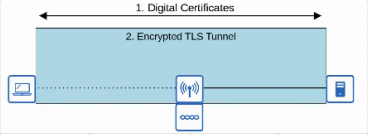
EAP-FAST (EAP flexible authentication via secure tunneling)
- 3 phase
  - a PAC (protected access credential) is generated and passes from the server to the client
  - a secure TLS tunnel is establish between client and authentication server
  - inside of the secure (encrypted) TLS tunnel, client and server communicate further to authenticate/authorize the client



PEAP (Protected EAP)
- like EAP-FAST but instead of PAC, the server has a digital certificate which client will use to authenticate with the server
- client must still be authenticated within the secure tunnel (eg: MS-CHAP)



EAP-TLS (EAP transport layer security)
- like EAP-FAST but require a certificate on the server and every single client
- the most secure wireless authentication method but more difficult to implement (because all client need a certificate)
- because client and server authenticate each other with digital cert => no need to authenticate the client within the TLS tunnel



Encryption and Integrity method

TKIP (temporal key integrity protocol)
- use MIC to protect the integrity of message
- a key mixing algorithm use to create a unique WEP key for every frame
- initialization vector is double in length from 24 bit to 48 bit => make brute force longer
- the MIC include the sender MAC address to identify the frame sender
- a timestamp is added to MIC to prevent replay attack (re-sending a frame that already been transmitted)
- use in WPA1

CCMP (counter/CBC-MAC protocol)
- use in WPA2
- consist 2 algorithm to provide encryption and MIC
  - AES (advance encryption standard) counter mode encryption
  - CBC-MAC (cipher block chaining message authentication code) used as a MIC to ensure the integrity of message

GCMP (galois/counter mode protocol)
- use in WPA3 (Wifi protected access 3)
- consist 2 algorithm to provide encryption and MIC
  - AES (advance encryption standard) counter mode encryption
  - GMAC (galois message authentication code) is used as a MIC

WPA (wifi protected access)

a certfication for wireless device (being tested in labs) for security capable

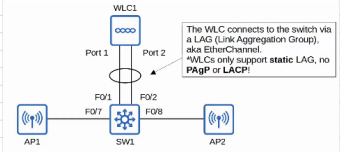
WPA, WPA2,WPA3

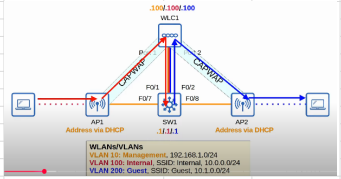
2 authentication mode
- personal mode:
  - a pre-shared key (PSK) used for authentication
  - eg: house wifi (just need password)
  - a four-way handshake is used for authentication and PSK used to generate encryption key
- enterprise mode (802.1X)
  - used with an authentication server (AS) (RADIUS server)

protocol
- WPA
  - encryption/MIC : TKIP
  - authentication: 802.1X for enterprise and PSK for personal
- WPA2
  - encryption/MIC : CCMP
  - authentication: 802.1X for enterprise and PSK for personal
- WPA3
  - encryption/MIC : GCMP
  - authentication: 802.1X for enterprise and PSK for personal

PMF (protected management frames): protect frame from eavesdropping/forging
SAE (simultaneous authentication of equal) protect the four-way handshake when using personal mode authentication
forward secrecy: prevent data from being decrypted after it has been transmitted over the air, so attacker cant capture and decrypt them

Day 58 Wireless configuration





Network topology when using WLC to control
traffic will have to go through CAPWAP (Control and Provisioning of Wireless Access Points)tunnel

CLI — configure on switch
- config vlan
- config interface (connect to AP) to access mode and asign to vlan
- config interface (connect to WLC) to Etherchannel (LAG) and config as trunk mode and assign all the vlan
- config SVI for each VLAN (default gateway for each subnet)
- config DHCP pool for each VLAN
  - for VLAN management
    - "option 43 ip" + WLC ip address
      - to tell other AP the WLC address when is being assigned IP
- "ntp master" — to make the switch to be NTP server

configure on WLC
- WLC will have initial setup with guidance question

GUI
- WLC can be configure with GUI (by connecting to console port of the switch or the WLC)
- enter browser => enter url as the WLC IP
- https://www.youtube.com/watch?v=r9o6GFI87go&list=PLxbwE86jKRgMpuZuLBivzlM8s2Dk5lXBQ&index=116

WLC port are the physical port that cable connect to
- service port: management port, used for out-of-band management, must connect to a switch access port because it only support 1 VLAN
- distribution system port: standard network port connect to the distribution system (wired network) and for data traffic
- console port: a standard console port, either RJ45 or USB
- redundancy port: use to connect to another WLC to form a high availability (HA) pair

WLC interface are logical interface within the WLC (eg: SVIs on a switch)
- management interface: use for management traffic such as Telnet, SSH, HTTP, HTTPS, RADIUS authentication, NTP, syslog,... CAPWAP tunnel also form to/from the WLC management interface
- redundancy management interface:when 2 WLC are connected by their redundancy port, 1 will be active and 1 will be standby
- virtual interface: use when communicating with wireless client to relay DHCP request, perform client web authentication
- service port interface: if service port is use => bound to it and used for out-of-band management
- dynamic interface: use to map WLAN to a VLAN, for example, traffic from the internal WLAN will be sent to the wired network from WLC internal dynamic interface

Day 59 Network automation

in traditional model => egineers manage device one at a time by connecting to their CLI via SSH
- typo and small mistake are common
- time consuming
- difficult to ensure all device adhere to the organization standard configuration
- data plane and control plane are both distributed, each device has it own data plane and control plane

in modern model => appy automation
- human error is reduce
- network become more scaleable, new deployment, changes and troubleshooting can be implement in fraction of the time
- network wide policy compliance can be assured
- reduce operating expense

Logical "planes" — is categorized various functions of network device into smaller part (planes)

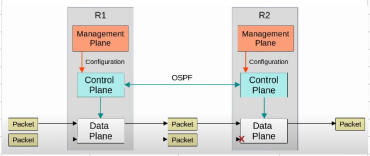
Router function:
- forward message between network by examining information in the layer 3 header
- use protocol like OSPF to share route information and build a routing table
- use ARP to build ARP table, mapping IP address to MAC address
- use Syslog to keep log of event
- allow user connect via SSH and managed
- etc

switch function
- forward message within a LAN by examining information in the layer 2 header
- use STP to ensure no layer 2 loop in the network
- build MAC address table by examining source MAC address of frame
- use syslog
- allow user connect via SSH and managed
- etc

Data plane — aka forwarding plane
- all task involve in forwarding user data/traffic from interface to another
- NAT (network address translation)
- deciding to forward or discard message due to ACLs, port security,...

Control plane
- function that build table like routing table, MAC address, ARP, STP,...
- the control plane controls what the data plane does (eg: building the router routing table)
- perform overhead work
  - eg: OSPF itself not forward user data packet, but it inform the data plane about how packet should be forwarded
  - STP itself not directly involved in the process of forwarding frame but it inform the data plane about which interface should and shouldnt be use to forward frame

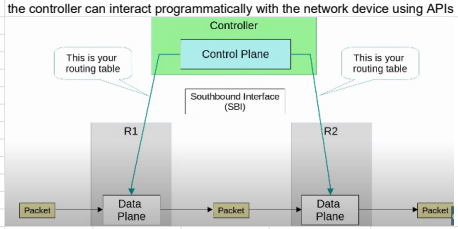
Management plane
- perform overhead work
- consist protocol that use to manage device
  - eg: SSH/telnet, syslog, SNMP, NTP,...



the operation of the management plane and control plane usually manage by the CPU (but this way is slow)
- instead a specialized hardware ASIC (Application - specific intergrated circuit) is used which is a chips built for specific purpose
- so when device receive control/manangement traffic (design for itself) it will be processing in CPU
- when device receive data traffic (going through the device) it will be processing by ASIC for maximum speed
- eg: switch
  - when a frame is received, ASIC will responsible for switching logic
  - MAC address table is stored in a memory called TCAM (ternary content-addressable memory)
  - ASIC feed the destination MAC address of the frame into the TCAM which return the matching MAC address table entry
  - the frame then forwarded out of the appropriate interface

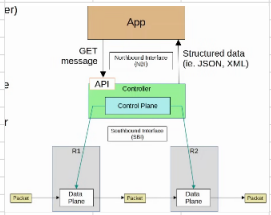
Software-Defined networking (SDN) — aka Software-defined architecture (SDA) or controller-based networking
- is an approach to networking that centralize the control plane into an application called a controller (just like WLC)
- traditional control plane use a distributed architecture
  - eg: each router run OSPF the router share routing information and then calc thei preferred route by themself
  - an SDN controller will centralized control plane function like calculating route instead

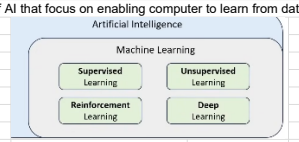
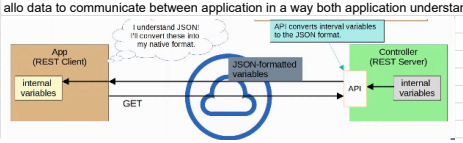
the controller can interact programmatically with the network device using APIs



SBI (southbound interface) — not a physical interface, just a software
- use for communication between the controller and the network device it control
- typically consist of a communication protocol and API
- APIs facilitate data exchange between program
  - data is exchanged between controller and network device by API on the network device that allow them
- eg: OpenFlow, Cisco Opflex, Cisco onePL, NETCONF

NBI (northbound interface)
- use for interact with the controller, access it data, program it and make change in the network via the SBI
- a REST API is used on the controller as an interface for app to interact with it
  - REST = representational state transfer
- data is sent in a structure (serialized) format such as JSON or XML
  - make easier for program to use the data



			Pros	controller collect information about all device in the network => have a wider overview		
				northbound API allow app to access information in a format that is easy for program to understand (eg: JSON, XML)		
				the centrailized data facilitate network-wide analytic		
				SDN tool can provide the benefit of automation withou the requirement of third party script and app		
				APIs allow third-party application to interact with the controller		
	AI & Machine Learning					
		AI (Artificial intelligence)	use computer to simulate intelligence, allowing them to exhibit behaviors typically associated with human like recognize pattern, learning, making decision and solving problems			
			eg: Virtual assistance, recommendation system, self-driving car and robotic, chatbots			
		ML (Machine learning)	is a subset of AI that focus on enabling computer to learn from data and improve without the need for explicit programming (hard-coded instruction)			
			4 type			
				supervised learning		
					advantage	highly accurate when labeled data is available
						straigthforward
					disadvantage	require large labeled dataset
						output is limited to the label in the trainning data
				unsupervised learning		
					advantage	no need for labeled data
						reveal hidden pattern
					disadvantage	less accurate
						interpretation and labeling of the result is require
				reinforcement learning		train model by rewarding or penalizing it action in given environment to maximize it performance overtime
						the model learn to take action that achieve the highest reward or best outcome
					advantage	capabla of learning complex behaviors
						adapt to dynamic environment
					disadvantage	resource intensive
						risk of suboptimal learning of the reward system isnt properly designed
				deep learning		use artificial neural network to process and learn from large and complex dataset
						neutral network is a computational model inspired by how biological neural network like human brain
						data is passes through multiple layer of nodes with each layer extracting increasingly abstract feature
						the neural network can be trained using supervised, unsupervised and or reinforcement method
					advantage	excel at handling large, unstructure data set
						archive state-of-the-art performance in task like image recognition, natural language procesing (NLP) and autonomous driving
					disadvantage	resource intensive
						model can be a black blox making it difficult to interpret how it arrive at it decision
		Predictive AI	use ML to analyze historical data and predict future outcome or trend			
				security annomaly detection, eather forecasting		
				eg: in networks		
					traffic forecasting	
					security threat detection	
					predictive maintenance	
			advantage	improve decision making by providing actionable insight		
				detect potential problem before they occur		
			disadvantage	require high quality relevant historical data		
				accuracy depend on how well the pattern in past data generalize to new scenario		
		Generative AI	use ML to learn pattern from existing data and create new content like text, image, audio,...			
				chatGPT, midjourney, gemini,...		
				eg: in networks		
					network documentation	
					configuration generation	
					network design	
					troubleshooting	
					script generation	
			advantage	great for creative task where human input is limited		
				enable automation of content creation across various field		
			disadvantage	risk of misuse		
				generate content only good was the quality of the training material		
				hallucination		
		AI in Cisco catalyst center	feature a variety of AI-enable feature to identify issue before they impact user, reduce time required to resolve issue and increase the performance and security of the network			
				AI network analytic		
				Machine reasoning engine		
				AI endpoint analytic		
				AI-enhance radio resource management		
Day 60	JSON, XML, YAML		data serialization languages/format			
		Data serialization	is the process of converting data into standardized format/structure that can be stored in a file or transmitted over a network and reconstructed later (by a different application)			
			allo data to communicate between application in a way both application understand			
						

JSON (javascript object notation)
is an open standard file format and data interchange format that use human-readable text to store and transmit data objects
REST APIs often use JSON
whitespace is insignificant
represent four primitive data type: string, number, boolean, null
string surround by " "
number not surround by quotes
boolean is true, false
null is null
have two structured data type: object, array
object is an unordered list of key-value pairs (variables) (just like dictionary in python)
object surrounded by curly brackets {}
key is a string
value is any JSON data type including another object (nested objects)
key and value are separated by a colon :
each key-value pair is separate by a comma
eg: {"name": "kien", "smart":true}

array is a series of values separated by commas
the value dont have to be in the same data type
array surrounded by square brackets []

XML (Extensible markup language) was dev as a markup language, but now used as a general data serialization language
markup language are used to format text (font, size, color, heading,...)
XML is generally less human-readable than JSON
whitespace is insignificant
REST APIs often use XML too

syntax: <key>value</key>

```
R1#show ip interface brief | format
<?xml version="1.0" encoding="UTF-8"?>
  <ShowIpInterfaceBrief xmlns="ODM://built-in//show_ip_interface_brief">
    <SpecVersion>built-in</SpecVersion>
    <IPInterfaces>
      <entry>
        <Interface>GigabitEthernet0/0</Interface>
        <IP-Address>192.168.1.1</IP-Address>
        <OK>YES</OK>
        <Method>manual</Method>
        <Status>up</Status>
        <Protocol>up</Protocol>
      </entry>
      <entry>
        <Interface>GigabitEthernet0/1</Interface>
        <OK>YES</OK>
        <Method>unset</Method>
        <Status>administratively down</Status>
        <Protocol>down</Protocol>
      </entry>
    </IPInterfaces>
  </ShowIpInterfaceBrief>
```

YAML (Yet another markup language)
used by the network automation tool Ansible
very human-readable
whitespace is significant
YAML file start with ---
"-" is used to indicate a list
key and value are represented as key:value

```
---
ip_interfaces:
- Interface: GigabitEthernet0/0
  IP-Address: 192.168.1.1
  OK?: 'YES'
  Method: manual
  Status: up
  Protocol: up
- Interface: GigabitEthernet0/1
  IP-Address: unassigned
  OK?: 'YES'
  Method: unset
  Status: administratively down
  Protocol: down
```

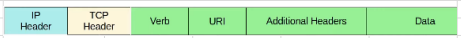
Day 61 REST APIs

APIs (Application programming interface)
is a software interface that allow two application to communicate with each other
in SDN architecture, APIs are used to communicate between app and the SDN controller (via NBI) and between the SDN controller and the network devices (via the SBI)
NBI typically use REST APIs

CRUD (Create, Read, Update, Delete) refer to operation performing using REST APIs
Create operation are used to create new variable and set their initial value
Read operation used to retrieve value of a variable
Update to change value of a variable
Delete used to delete variable
eg: HTTP used verbs (methods) that map to these CRUD operation

Purpose	CRUD Operation	HTTP Verb
Create new variable	**Create**	**POST**
Retrieve value of variable	**Read**	**GET**
Change the value of variable	**Update**	**PUT, PATCH**
Delete variable	**Delete**	**DELETE**

HTTP request when HTTP client send request to an HTTP server, HTTP header include information like HTTP verb, URI (Uniform resource identifier) indicating the resource it trying to access
HTTP request can include additional header which pass additional information to the server



HTTP response
the HTTP server will response which include a status code indicating if the request succeeded or failed as well as other detail
the first digit indicate the class of the response
detail

URI (Uniform Resource Identifier)

https://sandboxdnac.cisco.com/dna/intent/api/v1/network-device

scheme authority path

URL (uniform resource locator) is a type of URI

REST APIs

Representational State Transfer

is a framework for building APIs / describe a set of rule about how API should work

for application to communicate over a network, networking protocol must be used to facilitate those communication (usually use HTTP(S))

have 6 constraints

- Uniform interface
- client-server
  - mean they can both change and evolve independently of each other
  - when client application change or server application change, interface between them must not break
- stateless
  - mean that each API exchange is a separate event, independent of all past exchange between the client and server
    server will not store information about previous request from the client to determine how it should response to new request
  - if authentication requred => each request must have authenticate
- cacheable or non-cacheable
  - REST API must support catching of data
  - is storing data for future use
  - not all resource have to cacheable but cacheable resource must be declared as cacheable
- layered system
- code-on-demand (optional)

Cisco DevNet

is a Cisco dev program to help automation/write application with Cisco product, platform and APIs

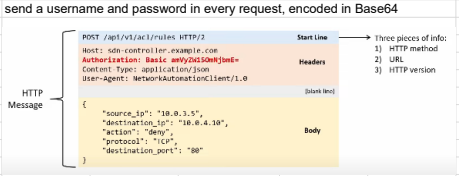
REST APIs Authentication

use to prevent malicious accessing sensitive data or modifying the application

use various type of authentication to verify client identity and secure access to resource (also called methods/schemes)

Basic authentication:

send a username and password in every request, encoded in Base64



these credential are encoded in base64 format but not encrypted

but Base64 can easily decoded => man in the middle can read and stole account:password

=> must alway use HTTPS to apply encryption when using basic authentication

advantage: simple and easy to implement

disadvantage: since credential are sent in every request => attacker could steal if not using HTTPS for encryption

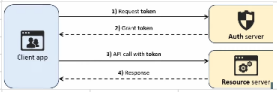
Bearer authentication

using a token instead of a username/password

the client first obtain a token by authenticating with authorization server (can be done using basic authentication or another method

this step separate vs the API call

and then for each API call, the client include the token in the HTTP authorization header



token usually expire after a set period of time

advantage: more secure than basic authentication

token expire, so stolen token will only temporary valid

disadvantage: if token stolen => attacker can access API until it expires

token need to be refreshed periodically => extra complexity to implement

should only use with HTTPS

API key authentication

use static key issued by the API provider

just like bearer tokens but API key will not expire until revoked

eg: ChatGPT APIs, Googlemap API

API key can be sent in:

- the HTTP authorization header (recommend)
- the URL by adding parameter to the end (not recommend because URL are logged by web server, proxies, browser,... => attacker can read easily)
- a cookie (sometime used for browser-based APIs)

advantage: easier to implement

good for tracking API usage (often use by cloud service and third-party APIs)

disadvantage: if stolen => attacker grant full access until revoke

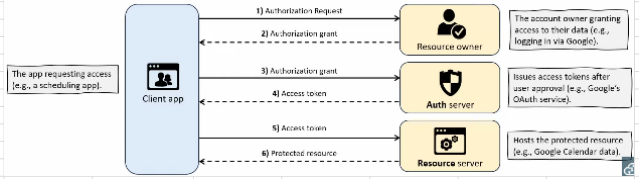
API keys must be rotated manually to maintain security

OAuth 2.0

is a secure authentication framework widely use in modern web application

it provide access delegation, granting third-party application limited access to resources on behalf of the resource owner
no need to share the resource owner credential with the third party
eg: logging in with Google, connecting app to social media account, calendar integration

6 step client app request authorization from the resource owner to access the resource
resource owner grant authorization by logging into their account and give permission
client app exchange the authorization grant for access token from the auth server
auth server provide access token to client app this access token like token in bearer authentication but OAuth 2.0 use refresh token to obtain new one without requiring user to log in every time
client app sent the access token to resource server
resource server sent the requested resource if the access token valid



Day 62 Software-defined networking (SDN)

the SDN controller can interact programmatically with the network device using APIs
SBI used for communication between controller vs network device
NBI allow to interact with the controller with application and script from another software

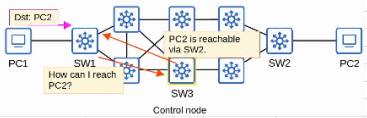
Cisco Software-defined access (SD-Access) is Cisco SDN solution for automating campus LANs
ACI (application centric infrastructure) is their SDN solution for automating data center network
SD-WAN is their SDN solution for automating WANs

Cisco DNA (Digital network architecture) center is the controller at the center of SD-Access

the underlay is the underlying physical network of device and connection (including wired and wireless) which provide IP connectivity (eg: using IS-IS)
eg: multilayer switches and their connection
purpose is to support the VXLAN tunnel of the overlay
3 different role for switches in SD-acces
edge node: connect to end host
border node: connect to device outsite of the SD-access domain (eg: WAN router)
control node: use LISP (Locator ID separation protocol) to perform various control plane function

if add SD-access on top of existing network (brownfield deployment) => DNA center wont configure the underlay
if add SD-access on new deployment (greenfield deployment) => DNA center will use to configure with the optimal SD-access underlay
all switches are layer 3 and use IS-IS as their routing protocol
all link between switch are routed port (STP is not needed)
edge node act as the default gateway of end host
the overlay is the virtual network built on top of the physical underlay network
eg: SD-Access use VXLAN (Virtual Extensible LAN) to build virtual tunnels
LISP provide control plane of SD-access
a list of mapping EIDs (endpoint identifier) to RLOCs (routing locators) is kept
EIDs identify end host connected to edge switches and RLOC identify the edge switch which can be used to reach the end host
Cisco TrustSec (CTS) provide the policy control (QoS, security policy,...)
VXLAN provide the data plane of SD-access

the fabric is the combination of the overlay and underlay (physical and virtual network as a whole)



Cisco DNA Center 2 main role be a SDN controller in SD-access
network manager in traditional network (non SD-access)
is an application installed on Cisco UCS server hardware
has REST API which can be used to interact with it
have SBI support protocol like NETCONF, RESTCONF, Telnet, SSH. SNMP

DNA center enable Intent-based networking (IBN)
which allow egineer to communicate their intent for network behavior to DNA center and then DNA center will take care of the detail (configuration and policy) on device

Compare vs traditional network management
traditional device are manually configure one by one via SSH or console connection
configuration and policy are managed per device (distributed)
new network deployment can take a long time due to the manual labor required
error and failure are more likely due to increase manual effort

DNA center-based network management
device are centrally managed and monitored from the DNA center GUI or other application using it REST API
the administrator communicate their intended network behavior to DNA center, which change those intention into configuation on the managed network device
configuration and policy are centrally managed
software version are centrally managed and can monitor cloud server for new version and update
net network deployment faster and reduce human error

Day 63 — Ansible, Puppet, Chef

configuration management tool that facilitate the centralized control of large number of network device

Configuration drift
is when individual change made over time cause a device configuration to deviate from the standard configuration as defined by the company
include configuring new device too

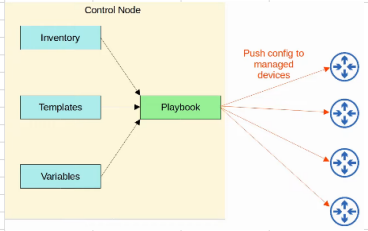
configuration provisioning
refer to how configuration change are applied to device
traditional way: configurate device one by one via SSH

modern way: use configuration management tool to make change to device on a mass scale with a fraction of the time/effort
template and variables is used in those tool to generic a config to sent to a device

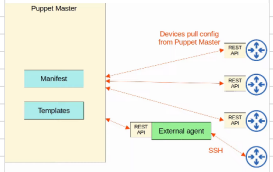
configuration management tool
can perform task like
- generate configuration for new device on a large scale
- perform configuration change on device (all device in the network or a certain subset of device)
- check device configuration for compliance with defined standard
- compare configuration between device and between different version of configuration on the same device

Ansible
owned by Red Hat
written in Python
agentless => not require any special software to run on the managed device
use SSH to connect to device, make configuration change, extract information,...
use a push model which asible server (control node) use SSH to connect to managed device and push configuration change to them

after install need to create some text file
- playbook: are blueprint of automation task, outline the logic and action of the task, written in YAML
- inventory: file list the device that will be managed by ansible, as well as characteristic of each device like role,... written in INI, YAML ,...
- template: represent a device configuration file, but specific value for variable are not provided, written in Jinja2
- variable: file list variable and their value which will substitute into the template to create complete configuration file, written in YAML



Puppet (not in CCNA anym
written in Ruby
agent-based => specific software must be installed on the managed device
use a pull model (client pull configuration from the puppet master/server)
client use TCP 8140 to comunicate with the puppet master
instead of YAML, it use proprietary language for file
file — manifest: this file define the desired configuration state of a network device
template: use to generate manifest



Chef (not in CCNA anymore
written in Ruby
agent-based => specific software must be installed on the managed device
use a pull model
server use TCP 10002 to send configuration to client
file — resource: the ingredient. configuration object managed by chef
recipes: outline the logic and action of the task performed on the resource
cookbook: set of related recipe groupped together
run-list: order list of recipe that are run to bring device to desired configuration state

	Ansible	Puppet	Chef
Key Files defining actions	Playbook	Manifest	Recipe, Run-list
Communication Protocol	SSH	HTTPS (via REST API)	HTTPS (via REST API)
Key Port	22 (SSH port)	8140	10002
Agent-based/ Agentless	Agentless	Agent-based (or Agentless)	Agent-based
Push/Pull	Push	Pull	Pull

Day 63 — Terraform

Infrastructure as Code (IaC)
the practice of provisioning and managing infrastructure (server, network, cloud resource) using machine-readable configuration file (code) instead of manual configuration (CLI/GUI)

automate infrastructure deployment and management, ensuring consistency scalability

configuration management (Ansible, puppet, chef)
manage existing infrastructure by installing software, configuring setting and maintaining system state
ensure consistency by applying and enforcing configuration across multiple device
use mutable infrastructure approach => modify/update after deployment, change made in place

infrastructure provisioning (Terraform)
create, modifies and delete infrastructure resources such as server and network infrastructure
focus on initial setup rather than ongoing configuration management
use immutable infrastructure approach => replacing the previous resource with a new one => no configuration drift

they can use together (Terraform provision infrastructure like VMs, network, storage,... and Ansible provide ongoing configuration and management

Procedural vs declarative approach
procedural (imperative) eg: Ansible, Chef
follow explicit step in a specific order to achieve the desired outcome
user must define each action to configure the infrastructure
provide greater control than declarative
declarative eg: Puppet, Terraform
define the desired end state
the tool will figure out the step need to be done to achieve it
easier to maintain and ensure consistency across deployment

Terraform
open-source IaC by HashiCorp
a provisioning tool focus on deploying infrastructure resource on various cloud & on-prem platform (AWS, Azure, ...)
use push model and agentless

3 main step
write: define the desired state of your infrastructure resource in configuration file
plan: verify the chang that will be execute before applying them
apply: execute the plan to provision and manage the infrastructure resource

terraform core written in Go, configuration file are written in HCL (domain-specific language)