

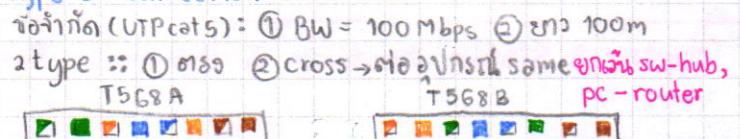
Chapter 1 NW Overview

- NW diagrams = โครงสร้าง กด กู้แล้ว ข้า. เข็มต่ออย่างไร?
- atype :: ① physical → port/interface เชื่อมต่ออย่างไร / อย่างไร (เน้นเทคโนโลยี) ② Logical → มีส่วนไหนบ้าง
- NW protocol → TCP/UDP, FTP, ARP, SMTP, POP3, IMAP, ICMP (Internet Ctrl Msg P.) → ใช้ ping (File Transfer P.) → รับส่ง file: client-server (Addr. Resolution P.) → map IP → MAC Addr. **NW Representation**
- Components of NW → HW → NW device ถ้า 3 type :: ① end devices = หลักจากที่เป็นคน
- ② intermediary devices = อุปกรณ์ที่ต้องผ่าน NW access device, Internetworking devices, Security devices
 - hub, repeater @ L1 → ส่งพร้อมกัน ไม่เกิด collision :: ใช้ CSMA/CD (Carrier Sense Multiple) ถ้าเกิด collision จะหยุด
 - switch, bridge @ L2 → Learning / Flooding / Filtering / Forwarding / Aging Routers @ L3 → Routing
- ③ network media - สื่อสากล บน fibre Optic, Wireless, copper → UTP, STP, coaxial
- SW → ① switch เลือกแบบใด ② router เลือกไปทางที่ต้องการ
- Types of NW → size :: ① small home nw → ใช้ช่องด้านนอก only ② small Office/Home Office → config จัดตั้งนอก → ใช้ไฟฟ้า
 - ③ Medium to Large nw → ติดต่อภายใน 100-1000 com ④ World Wide NW บน internet → infrastructure :: ① Local Area NW (LAN) → มี 1 กลุ่ม admin และ policy / NW ต้องต่ออยู่ที่ไฟฟ้า
 - ② Wide ~ (WAN) → มีหลายกลุ่ม admin
- บริเวณ :: Metropolitan Area NW (MAN), Wireless LAN (WLAN), Storage Area NW (SAN), Personal Area NW (PAN)
- Reliable NW ① fault Tolerance → ทนต่อ A. ผิดพลาด ② Scalability → สามารถเพิ่มขนาดได้ ใจกลางเมืองต่อ user มาก
- ③ Security → จำกัด or limit ก. เม้าส์ ④ Quality of Service (QoS) → ผู้ให้บริการมี quality ในการให้บริการ

Layer with TCP/IP & OSI Model

Application	Application	⑦ PDU
	Presentation	⑥ (P. data unit)
	Session	⑤ Data
Transport	Transport	④ Segment
Internet	NW	③ Packet
NW Access	Data Link	② Frame
	Physical	① Bits

Type of Connection in a LAN



▪ WAN connection → 2 ฝ่าย DCE (ตัว) → มีนาฬิก clock rate 56000 (เชื่อม router → router) DTE (ตัว) 25-45-to-DB-9 Adapter labeled DTE

Chapter 2 Basic Router Configuration NW addr. 3 type

- ① Port Addr. (Service Addr.) @ L4 : กำหนดโดย Internet Assigned Number Authority : IANA (Ex.
 - 0-1023 : requesting entities "well known ports" destination port
 - 1024-49,151 : registered port = publish ให้ทุกคนรู้
 - 49,152-65,535 : dynamic or private port "Randomly generate" source port
- 53 : DNS (domain name server) [TCP/UDP], 80 : www (world wide web) HTTP, 81 : HOSTS (Name Server), mail transfers

② Logical Addr. (IP Addr.: IPv4) @ L3 : class A-C แต่ละตัวที่จะ max workstation required

- 66 ตัว = nw/node or com → 2 logical name (domain name) / IP unique		Private addressing + ip can reuse
192.168.1 /24 → prefix range	class A: 0 Host 24 bit	RFC 1918 Internal Addr. Range 10.0.0.0-10.255.255.255 CIDR Prefix 10.0.0.0/8 → ใจกลาง internet
255.255.255.0 → subnet mask	B: 10 16 bit	128-191 172.16.0.0-172.16.255.255 192-223 192.168.0.0-192.168.255.255 224-239 multicast → ส่วนๆ ทุก com ใน 1 LAN 192.168.0.0/16
192.168.1.255 → broadcast ip addr.	C: 110 8 bit	192.168.0.0/16
255.255.255.255 → broadcast NW	D: 1110	240-255 experimental
	E: 111	

③ Physical Addr. (MAC Addr.)

- Ethernet : 48 bit ฐาน 2 = 12 ตัว ส่วน 7 บิต (0x 37E616)
- ก่อตัวโดย IEEE → กำหนด 3 byte (24 bit) code "Organizationally Unique Identifier (OUI)" → 2 กฎ
 - ① ทุก MAC ที่ก่อตัวให้ NIC or Ethernet device ต้อง → ใช้ OUI ที่ 3 byte แรก
 - ② ~ same OUI ต้องมี unique ที่ 3 byte หลัง

▪ Msg Delivery ผ่านไฟเบอร์ออฟฟิวชัน ② ~ same OUI ต้องมี unique ที่ 3 byte หลัง

▫ Unicast = ส่งต่อ dest. com โดยตรง Broadcast = ส่งต่อทุก com ที่ต้องการ DHCP, ARP

▫ Multicast = ส่งต่อ com ที่ต้องการ com ที่ต้องการ service → ต้องตั้งค่า : 01-00-5E-xx-xx-xx

• Cisco IOS (Internetwork Operating System)

- f# :: ① Addressing ② Interface ③ Routing ④ Managing Resource ⑤ Security ⑥ QoS
- Router & Switch Boot Sequence

ROM	① POST (Power On Self Test) → check hw ทำงานได้ดีไหม?	NVRAM
Bootstrap	② Run boot loader SW	TFTP Server
Flash/ TFTP server	③ Boot loader does low-level CPU initialization	Console
	④ ~ initializes the flash filesystem	
	⑤ ~ locates & load a default IOS ที่ run บน RAM	

locate and load config
file or enter setup mode



Accessing a Cisco IOS Device

- ① Console port ② Telnet ③ Secure Shell (SSH) ④ Aux Port
- ↳ Terminal Emulation Program :: PUTTY, Tera Term, SecureCRT, HyperTerminal, OS X Terminal

• Navigating the IOS → 2 mode :: ① User ">" ② privileged (enable) "#"

The Command Structure

- ① Context Sensitive Help: "?"
- ② Command Syntax Check = enter `show` แล้ว show อะไรก็ได้
- ③ Hot Keys and Shortcuts ④ IOS Examination Commands → `show`
- Getting Basic ① ตั้งชื่อ hostname ② จัดการเบ้าไฟฟ้า ③ กำหนด intf addressing ④ `check config + n. reboot`
- `R(config)# hostname name#`
- `R(config)# banner motd # text #`
- ⑤ save → `R# copy running-config startup-config`
- ⑥ Addressing device
1. เลือก interface to config
- Physical Interface / Loopback interface
`R(config)# interface type port`
~ type slot/port
~ type slot/subslot/port
- switch virtual interface (SVIs)
`Switch(config) # interface vlan number`

2) set ip addr.

`R(config-if)# ip address ip-addr. subnet-mask`
~ no shutdown

④ Verify Connectivity

- `R# show running-config` → กด config ? บันทึก
- ~ `startup-config` → ~ set ~
- ~ `ip route` → ~ route table
- ~ `interface` } show info ของ interface
- ~ `ip interface` } show config + status
- ~ `ip interface brief` → show interface ทุกๆสิ่ง, set? status?
- ~ `traceroute`
- ~ `ping`
- PC > `ping`
- ~ `tracerc`
- ~ `route print` ตรวจสอบ ที่ไว้ไฟ
- ~ `nslookup`
- ★ `section, include, exclude, begin`

Chapter 3 Static Routing & Dynamic Routing Protocol

of of Router → Characteristic :: ① Topology ② Speed ③ Cost ④ Security ⑤ Availability ⑥ Scalability ⑦ Reliability

▷ Packet Forwarding Methods: ① Process switching = นำ pack ผ่านทุกๆ router → process @CPU 7 บน interface ไหน?

② Fast switching = มีตัวช่วยให้ forward รวดเร็ว ③ Cisco Express Forwarding (CEF) = forward pack ตามลักษณะ

• Connect Devices (.)¹ (.)² ① subnet nw = nw ip & subnet mask

▷ Default gateway → ไป first or last usable host

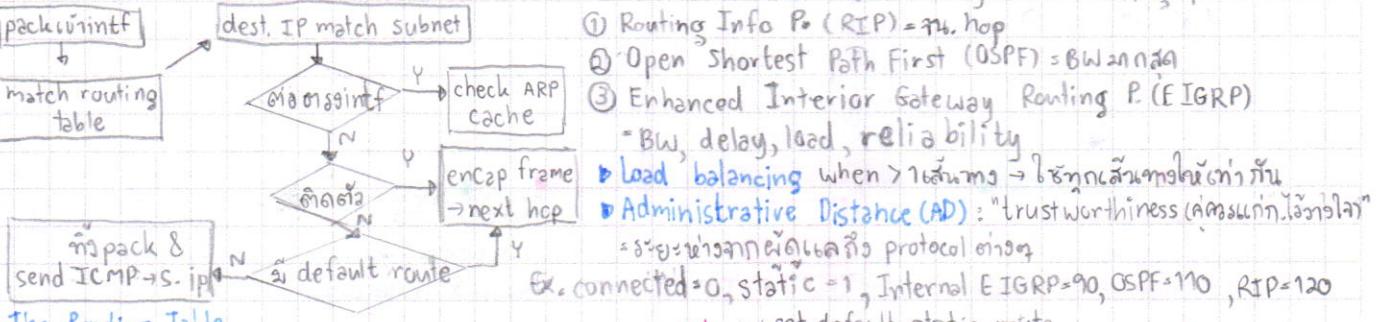
▷ Enable IP on a Host :: ① statically Assigned IP addr.

② Dynamically ~ → can find ip ให้ when nw ใช้ DHCP (Dynamic Host Config P.)

▷ Switching Packet between NWs: ปัจจัย :: นำ dest. ip (L#3) → map บน routing table → นำ mac ให้ dest. MAC (L#2)

Path Determination

▷ Best Path : lowest metric (cost) ⇒ Dynamic routing p. 78



The Routing Table

D dest. nw [AD/metric] via next-hop ip addr. 00:00:00 serial0/0/0 interface

↳ หมายความว่า L=Local, D=EIGRP

set default static route

`R(config)# ip route 0.0.0.0 0.0.0.0 ?ip-addr.interface`

② Dynamic Routing Protocol → auto

② EGP (Exterior Gateway Routing P.) :: BGP

② IGP (Interior) ~ :: RIP, OSPF, EIGRP, IS-IS (Intermediate System to Intermediate System)

• Classful Addressing → Update one class

• Classless Inter-Domain Routing (CIDR)

▷ summarization: ปัจจัย ① อยู่ใน同一个 subnet ② ไม่มีเส้นทางเดิน

▷ ตั้ง set ใหม่ต่อ ว่าจะ ① ระบุ ip → ราก

no ip ต่อ ก่อน

② Group bit ที่ same ทุกๆ subnet

• VLSM & Fixed Length Subnet Masking

ว่าจะ ① prefix ทุกๆ - prefix ที่ =

= 26 bit ที่ป้องกัน

② ฝ่ายที่ทุกๆ bit ที่ป้องกันแล้วค่าฐาน 10 ที่

+ ip start ที่ส่วน + 1 ค่าอนุที่ป้องกัน

Routing

① Static Routing → manual

จุดเด่น: security, ใช้ resource น้อยใน process, ไม่ routing entry

จุดเสียหาย: จำกัดใน scalability, can't ปรับเปลี่ยน

กรณี when: nw ขนาดเล็ก, 76, router น้อย, nw จำนวนมาก "Stub nw"

4 type : ① Standard ② Default = ให้ when dest. ip ไม่ match

③ Summary ④ Floating = ไม่ backup

`R(config)# ip route nw-addr. subnet-mask ?ip-addr. exit-interface`

Config : Next-hop Option

`R(config)# interface G0/0`

~ -if # ip address 172.168.1.0 255.255.255.0 } ไม่ต้อง router

~ no shutdown

`R(config)# ip route 172.168.1.0 255.255.255.0 172.168.1.2 1 2 2`

↳ S0/0/0 4 5 6

Chapter 4 Distance Vector Routing Protocol & RIPv1

☆ Dynamic Routing Protocol (DRP)

- **fn:** ① share info. ระหว่าง router ② auto update routing table when topology เปลี่ยน ③ หา best path
- **purpose:** ① รีโมท (ไกล) nw ② ปรับปรุง routing info. ③ หาน์ best path \rightarrow dest. gw ④ หาทางใหม่ when path เสื่อมคุณภาพ
- **component:** ① Algorithm: ใช้ routing table & best path ② Routing P. Msg.: แจ้ง neighbor & แลกเปลี่ยน routing info.

Dynamic Routing

VS Static Routing

A. ผู้ใช้งาน config	ผู้ใช้งาน gw (same device)	ผู้ใช้ gw (ผ่าน command ทุก router)
Required by admin	Advanced (better config than basic \rightarrow nw ใน gw + รับส่งข้อมูลที่ซับซ้อน)	No เพิ่มเติม (กำหนดโดยอัตโนมัติ)
Topology change	ปรับ auto	admin config ให้กับ all
Scaling	เน็ตเวิร์ก simple & complex (router ต่อตัวกันไม่ต่อ gw)	เรียบง่าย simple topology
Security	น้อยกว่า	มากกว่า
Resource usage	ใช้ CPU, mem (เก็บ routing info.), link bandwidth	No เพิ่มเติม
Predictability	Route & current topology	Route \rightarrow dest. คาดเดาได้ยาก

▷ Classifying (ผู้จัดจำแนก) Routing Protocol

DRP

(ภายใน AS) Interior Gateway Protocol (IGP) (ภายนอก AS) Exterior Gateway Protocol (EGP)

Distance Vector Protocol

- ประเมิน vector [distance, direction]
- incomplete view ของ nw topology
- periodic update (ระยะตามที่ตั้งค่า)

RIP (Routing Info. P.) IGRP

RIPv2

EIGRP (Enhanced Interior Gateway Routing P.)

Link-State Protocol

- complete nw topology
- update ไม่บ่อย periodic

OSPF IS-IS (Intermediate System to Intermediate System)

BGP (Border Gateway Protocol)

- Autonomous System (AS) ผูกพัน
- ต้องผ่าน router ก่อนถึง gw ได้คุณภาพ
- ห้าม single authority
- กำหนด policy or กฎต่างๆ

▷ Routing Protocol Metrics

- metric = ค่าที่ใช้ในการคิด/หา ไป gw dest. nw จะเลือก best path บนพื้นฐาน Hop count, BW, Cost, Delay, Load, Reliability
- Load balancing : if nw มี gw มากกว่า 1 ต้อง metric เท่ากัน \rightarrow แบ่งเท่าๆ กัน. ใช้ gw แต่ละ gw ลากเส้นทางต่างๆ กัน

▷ Administrative Distance of a Router (AD)

- ตั้งค่า gw ประจำตัว : เป็นตัวตั้งค่า gw ที่จะใช้ใน gw ประจำตัว particular (เฉพาะตัว) route , ชี้ว่า AD ของ gw นั้นต้องมากกว่า gw อื่นๆ
- เพิ่มเติม 瓜 G Page 1-2 : EIGRP summary route = 5, External BGP = 20, IGRP = 100, IS-IS = 115, External EIGRP = 170,

▷ Distance Vector

Routing Protocol Ex. RIP, IGRP, EIGRP

Internal BGP = 200

- Technology : ต้องรู้สึกว่าต้องคิด : ① vector or direction, traffic ควรถูกคุ้มครอง ② Distance (cost) to final dest.
- ลักษณะ : ① periodic (กำหนดเวลา) update ② neighbors (router ใกล้เคียง) ③ broadcast (ass. 255.255.255) update

④ 未必 routing table all ให้ routing update เสมอไป

- ~ ตรวจสอบ Routing Protocol = ตรวจสอบว่า check ต่อ DV ดี / ไม่ดี ?: @ Time to convergence \rightarrow ลากยาว \rightarrow steady-state ของ routing table ต้องติดต่อ gw และ \oplus Scalability \rightarrow ขนาดใหญ่เพิ่ม \oplus Resource usage \oplus Implementation & maintenance

▷ NW Discovery (ค้นพบ) (ทำ basic config ก่อน)

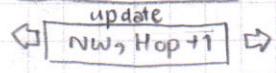
- 3 stage : ① cold Starts : Router Initial Start Up

② Initial Exchange of Routing Info \rightarrow ผู้จัดการและ exchange

③ Exchange of Routing Info. \rightarrow update (เพิ่ม hop count) routing info.

\rightarrow บน router ต้องตั้งค่าอะไรบ้าง?

NW	Intf.	Hop
update		



▷ Routing Table Maintenance

- periodic update : RIP update timer (default 30s), Invalid routing info. (เชิงลบ) timer (default 180s), Hold down (ถ้า down \rightarrow hold ไม่รีบตัว gw up) timer (เชิงลบ) timer (default 180s), Flush (ลบ) timer (default 240s)

• Bounded Update : EIGRP \rightarrow update เมื่อที่เปลี่ยน

• Triggered Update \rightarrow update เมื่อเวลาต่อเนื่อง periodic time

• Random Jitter \rightarrow ใช้ใน nw ที่มี gw มากกว่า multiple access router ขนาดตัวเพิ่มขึ้นต่อไป \rightarrow if ไม่ได้ update ก็ไม่ต้องรีบ : จึงใช้ random

▷ Avoid Routing Loops \rightarrow เกิดจาก ไม่ standard DV, ตัว intf ไม่ลง down \rightarrow ถูกตั้งลงใน routable \rightarrow กันตัวกัน gw \rightarrow

▪ แก้ปัญหา ① Set max hop = 15 \rightarrow if hop = 16 \rightarrow unreachable (จะ down ไม่เหลือ)

② holddown timer (ตัว intf. down \rightarrow hold)

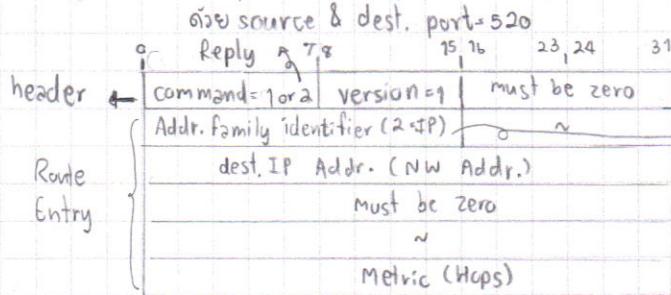
③ split Horizon Rule \rightarrow ไม่ส่งข้อมูลกลับไปทาง intf. ที่รับมา



- ④ Route Positioning → เก็บ down : set unreachable → สั่ง unreachable ที่ไป่nb position(hop=16) ถ้าเก็ง)
- ⑤ ③ with ④ → ตั้งเก็บ unreachable ที่ over rule horizon [อยู่ช่วง ip intf. ก็ down แล้ว intf. an] → set hop=16
- ⑥ IP & TTL (Time to Live) → เก็บ down แล้ว update (⇒) but, จะเขย่า when TTL=0

♥ RIP version 1 [AD = 120]

- คุณสมบัติ: ① classful, DV ② metric = hop count
- hop count > 15 เป็น unreachable
- update broadcast ทุกๆ 30s
- Msg Format encapsulation (css) ใน UDP segment



	RIPv1	RIPv2	IGRP	EIGRP
Speed of convergence	slow	slow	slow	fast
Scalability - size nw.	small	small	small	small
Use of VLSM	X	✓	X	✓
Resource usage	Low	Low	Low	Medium
Implement & maintain	Simple	Simple	Simple	Complex

→ 2 type ① Request [command = 1] →

→ แก้ routing table

→ แก้ config ไว้เชื่อมกับ update

② Response → แก้ info. ของ routing table

• Ip addr แบบ class A, B, C

■ Basic RIPv1 config ① An basic config

② แก้ router rip R1(config)# router rip แก้ตัวง่ายๆ
+ แก้ส่วน R1(config-router)# network nw-ip

■ Verification (ตรวจสอบ) & troubleshooting (ปัญหา): show running-config / ip route / ip protocols, debug ip rip

• passive intf. command คำสั่ง update intf. ต้องเป็น R(config-router)# passive-interface intf-type intf-number

■ Automatic Summarization: RIP Auto summarizes classful nw → ลดขนาด routing table (Fe/B/S) (0/0, 0/0/0)

→ กรณี: ① ขนาด routing update ② single router ไม่สามารถ handle multiple route ได้ใน routing table

→ จุดเด่น: รองรับ discontiguous nw (major nw ต้องติดกัน but ไม่ต้องติดกัน) → สามารถ load balancing ได้

• Boundary Routers: summarize RIP subnet from 1 major nw to another (จะมี nw ที่ไม่ major ให้ตัดกัน)

■ Processing RIP update

• กรณีที่ต้อง update ที่ intf. ว่าเป็น classful เดียวกันหรือ? → Y: Update subnet nw 182.172.16.1.0

↳ N: Update classful 182.172.16.0.0

■ default route & RIPv1 ต้องการ config บน routing table (อาจต้อง config protocol) → แก้ default route

R(config)# ip route 0.0.0.0 0.0.0.0 50/0/1 (intf-type intf-number)

• default info. originate command → คำสั่ง update ที่จะ rip ว่า 182.172.16.0.0 : static ↔ dynamic

Router ต้องยังคงเป็น 2 protocol & R(config-router) # default-information originate

Chapter 5 RIP version 2 & Access Control Lists variable Length Subnet Masking

RIP v 1

VS

RIP v 2

classful (ใช้ subnet mask, ไม่ support CIDR) not support discontiguous subnet	classless (update subnet mask, support VLSM, support route update next hop addr.)
▪ VLSM b/c ไม่ใช้ subnet mask routing update → broadcast (255.255.255.255)	▪ รองรับ authentication (รองรับ discontiguous ทาง物理) (Prefix Summarization) ▪ Routing update → multicast

▪ timer จำกัด routing loop

▪ split horizon or split horizon with poison reverse

▪ triggered update

max hop count = 15

▪ จุดเด่นของ RIPv1

- เป็น virtual interface { • loop back intf. → ping ไป ip virtual intf. → reply กลับ (timeout)
configurable in routing table } • Null intf. → ใช้กันคนเดียวใน channel ที่ไม่ต้องส่ง → วิ่งบน null intf. → packet discard ถ้า update ออกมาก
- Static route & Null intf. → null intf. จะรับน้ำหนึ่งที่ไม่ต้องอ่าน static route

R(config)# ip route summary-static-route subnet-mask Null0

(major-nw) → รองรับ super net route

▪ Route redistribution (ตัวตั้ง) → บันทึก rip ที่ static ลงใน db รองรับ static ไม่ลับ

R(config-router) # redistribute static

▪ Verify & Test Connectivity: show ip interface brief, ping (wa: 1-10, u: 1-10, . = timeout), traceroute

▪ RIPv1: ① classful ② ไม่ support subnet mask ③ summarize nw @ major nw boundaries ④ if nw ไม่ discontiguous & RIPv1 config convergence ต้องไม่ติดกัน

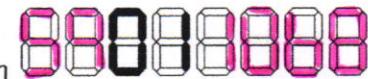
▪ ตรวจสอบ routing table: debug ip rip (content of routing update), ต่อไป RIPv1 จะใช้ subnet mask สำหรับ nw. addr. #

♥ RIP version 2

▪ show ip protocols

▪ config • Enabling & verify (ตรวจสอบ) RIP v2

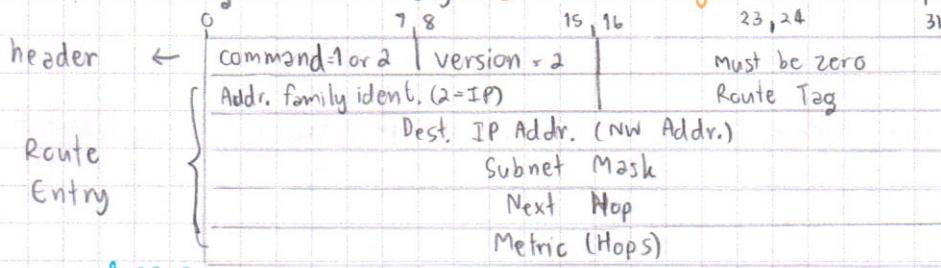
- config RIP → RIPv1 → ค้นพบได้ทั้ง v1 & v2 แต่ต้องเลือก v1
→ RIPv2 → ค้นพบ & จำกัดที่ v2



- Auto-Summary & RIPv2 → auto sum route @ major nw boundaries

→ sum route ต้อง subnet mask ที่ < classful subnet mask

- disabling Auto-Summary: no auto-summary b/c. when nw topology ต่อไป discontiguous



VLSM & CIDR

→ verify info. ที่ sent by RIPv2 debug ip rip

→ VLSM → เบื้องต้น nw addr. & subnet mask

→ CIDR → หู superhosting (= bunch ของ contiguous classful nw ที่ เป็น addr. same single nw)

→ verify show ip route, debug ip rip

D Access Control List = คุณลักษณะที่ต้องการ → check source → dest คืออะไร?

→ ท่าที่ conversation

→ ท่าที่อยู่ใน (เบื้องต้น) คืออะไร?

- Packet Filtering ① dest, source @ L2 ② protocol หรือ ③ ที่ nw ไหน, หรือ ที่อยู่ใน interface or block ที่?

- Operation → ฝ่าทาง ที่ sequence statement

→ last statement ที่ implicit deny → block = discard

E standard IPv4 ACLs

check source addr.

จะ permit or denies ที่ๆ ที่ protocol

access-list 10 permit 192.168.30.0 0.0.0.255

number ACL: 1-99 & 1300-1999

F Extended IPv4 ACLs

check source & dest addr

จะ permit or denies specific (เฉพาะ) protocol

access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80

number ACL: 100-199 & 2000-2699

G wild card

→ invert ของ subnet mask หรือ 255.255.255.255 - subnet mask

→ bit ที่ A1 0 = match / fix , 1 = ignore / ถ้าต้องการ

→ กรณีที่ set ip : ① หา A1 ที่ match ของ bit ที่ 0 ก็ได้ ให้ wild card mask 0110.0110 = 0

(match range) ② bit ที่ 1 หรือ ไม่ใช่ 1

→ if กรณีที่ ไม่ใช่ 0 ให้ ลากยาว Pattern or/and ร่วม กัน → สุดท้าย wild card จะ same กัน

→ keyword → 0.0.0.0 = match all ให้ host เข้า access-list 1 permit host 192.168.10.10

→ 255.255.255.255 = ignore all ให้ any หรือ ~ any

H Guideline for (3Ps)

→ One ACL / protocol = ctrl traffic flow ผ่าน intf., ACL ที่ define (def) protocol enable on intf.

ACL creation → One ACL/direction = ACL ctrl traffic in 1 direction at time on intf., ใช้ ACL ctrl in & out bound traffic

→ One ACL/interface = 1 ~ 8 สำหรับ intf. Ex G0/0

• where pass: ① Extend ACL: @ ไฟล์ source ② Standard ACL: @ ไฟล์ destination

I config ACLs

→ Standard

→ เลข number R(config)# access-list ACL-num deny/permit|remark source [s-wildcard] [log]

R(config)# access-list ACL-num permit any

ไฟล์ config 1 ไฟล์

ไฟล์ config 2 ไฟล์

but name can ไม่ได้

remove all : no access-list

name (permit, deny,

remark) หาย

① no access-list ACL-num → ลบไว้

② no access-list # → ลบไว้

comment → ให้ คำอธิบาย

comment → ให้ คำอธิบาย

→ เลข name R(config)# ip access-list [standard | extended] name

verify: show ip interface, show access-list

Securing VTY port (Telnet) → อย่างไรก็ตามเราต้อง permit ให้ทางที่เราต้องการ ให้เข้า

R(config-line)# access-class ACL-num {in|out}

→ Extended : filter = source /dest addr., protocol, port number

R(config)# access-list ACL-num {deny|permit|remark} protocol s. (s.-wildcard) [operator operand] [port p-num or name]

dest. (dest.-wildcard) [operator operand] [established]

→ คำสั่งที่ เหมือน standard แต่ 2 แบบ : number & name

- debug - output : debug ip packet ACL-num

Chapter 6 OSPF & DHCP

D Link-State Routing Protocol = บัญชี protocol ที่จัดการเรื่อง complete (all info. all) map ของ nw topology

ที่ 1 ก่อน → นำ shortest path first (SPF) → หา dijkstra



- **benefits**: ① large nw ② fast convergence ③ admin մէջք
- **processes update** ① learn info. առ link → ② say hello → ③ տես info. առ մէջք Link-State Packet (LSP) → ④ router flood LSP to all neighbors → կօբյեկտները db ⑤ router եռ all LSP եղինի db (լուսնային tree) + Adding OSPF

- խօսք: ① մը առ topology map կան առ shortest path ② fast convergence info մէջք. ③ LSP sent only when topology պայման (ուժ հայտ առ info մէջք → մը առ shortest path) ④ hierarchical design (NW հայտ մէջք) + alloc resource bcc. Առ info. մէջք area

- խօսք: ① կօբյեկտները link-state այդ ② ի՞չ CPU հեռ մէջք նա ③ ուստի LSP մէջք ի՞չ BW մէջք

♥ OSPF [AD=110]

↳ 3-table: ① Neighbor **show ip ospf neighbor** ② Topology (մէջք map) **show ip ospf database** ③ Routing (մէջք shortest path)

• message → Encapsulating Հ-MAC dest. = Multicast: 01-00-5E-00-00-05 or 01-00-5E-00-00-06 path) - protocol field = 89

→ type OSPF packet: 01 Hello → 7 տար 10s (default: multiaccess & point to point nw),
→ 7 տար 50s (default: non-broadcast multiaccess [NBMA] nw)
→ Cisco default 4 time → (40s)

02 Db Description (BBD) → synchronization db info.

03 Link-State Request (LSR) → request link-state

04 ~ Update (LSU) → send update link-state

05 ~ Acknowledgment (LS Ack) → ըստ առ լուսային

• Operation: (հայտ, մէջք) ① Down state (հայտ) → ② init state (հայտ մէջք hello) → ③ Two-way State (հայտ մէջք hello)

(Sync OSPF PB) → Gx Start State → Exchange State → Loading State → Full State (ստուգային router update info. մէջք մէջք)

↳ config Single-Area OSPFv2 R(config)# **router ospf** process-id → 7-65,535, և վեց locally significant (այդ մէջք մէջք)

• OSPF Router IDs R(config-router)# **router-id** 1.1.1.1 → ըստ լօպբաք, active intf. ip մէջք բայց մասնաւոր լայն թիմ R(config-router)# **network** nw-addr wildcard-mask **area** area-id

↳ OSPF cost → ի՞չ BW մէջք (default reference BW = 10⁸)

$$\rightarrow \text{cost} = \frac{108 \text{ bps}}{\text{intf. BW bps}}$$

→ 10 Gb Ethernet	=	100×10^8	→ cost = 1
→ Gb	~	10×10^8	→ cost = 1
→ Fast	~	10^8	→ cost = 1
→ Serial	~	1.544×10^6	→ cost = 64

• ուստի վայենակոստ → ① վայենակոստ ref bw: **auto-cost reference-bandwidth** [BW - mbps (10⁸)

- Fast Ethernet → ի՞չ ref BW = 100 mbps $\Rightarrow F/G = 10G = 1$

- Gigabit ~ ~ ~ = 1000 mbps $\Rightarrow F \cdot 10, G = 10G = 1$

- 10Gigabit ~ ~ ~ = 10000 mbps $\Rightarrow F = 100, G = 10, 10G = 1$

→ ② վայենակոստ bw: R(config-if)# **bandwidth** 64 (EIGRP & OSPF կանոնական)

→ ③ վայենակոստ : ~ ~ # **ip ospf cost** 15625

• verify OSPF: **show ip ospf neighbor**, **show ip protocols**, **show ip ospf interface brief**, **show ip ospf**

• more config - Redistributing an OSPF Default Route

R(config)# **ip route 0.0.0.0 0.0.0.0 loopback N**

~ # **router ospf** process-id

R(config-router)# **default-information originate** → ուստի վայենակոստ default route update մէջք ospf

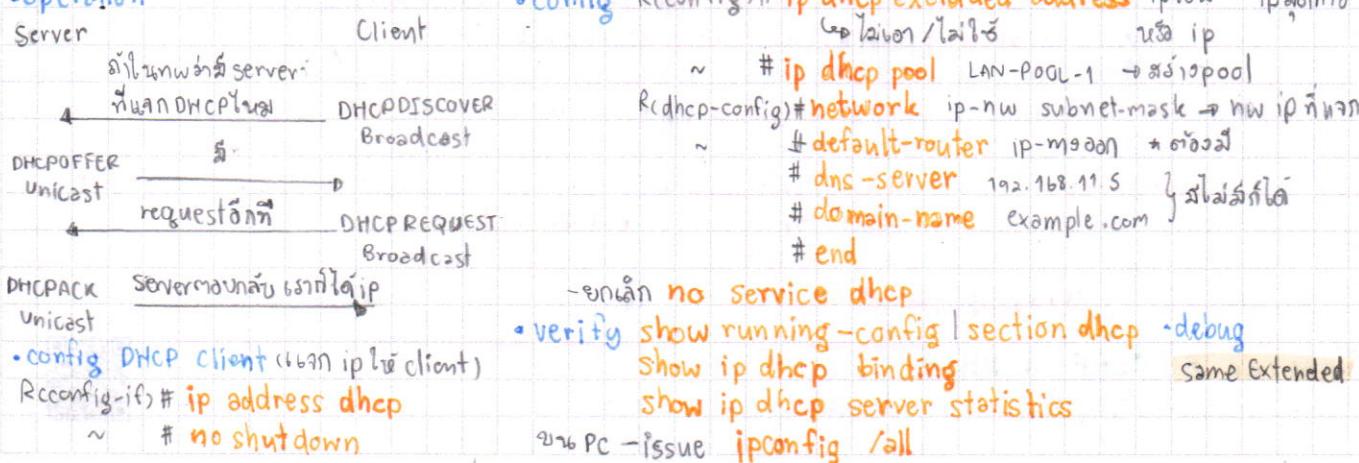
▷ DHCP(Dynamic Host Configuration P.) → ուստի config եղանակ host ի՞ւ auto (առ ip, subnet mask, default gateway, dns)

• method: ① Manual Allocation : admin assigning

② Automatic Allocation: DHCPv4 auto assign addr. յուն pool և յուն lease (սահմանափակ) time

③ Dynamic Allocation: յուն պահանջման յուն ip լուսային lease time → re ip լուսային lease time → re ip լուսային lease time

• operation



Chapter 7 Basic Switch Address Resolution Protocol

▷ LAN Design → Borderless sw nw design: แบ่งออกเป็น Hierarchy, -Modularity, -Resiliency, -flexibility

▷ ลักษณะ: ① 3-Tier LAN Design ขั้นตอน: 1) Core 2) Distribution 3) Access ② 2-Tier ~ ขั้นตอน: 1) Collapsed Core / Distribution 2) Access ทุก [1] Core → อุปกรณ์ที่รองรับ BW สูง → ทำให้มี speed ที่สูงกว่าเดิม [Layer 3 Support, Gigabit Ethernet]] Link aggregation Layer [2] Distribution → ใช้ส่วนของ 1 ชั้น [3] Security Policy/Access Ctrl [Redundant components] ตัวเลือกของ [Switches/BW device ที่ต้องการ] ผ่านไป [3] Access → ต่อไปยัง end device, Port Security, VLAN, Fa/Gig Ethernet, Power over Ethernet รองรับ or ไม่รองรับ Quality of Service (QoS) ระบบคานโยน LAN BW & จำกัด max of 8.1.1.1 Server @ Enterprise S (รวมทั้งจุดกลาง) → ต่อตัว @ MDF (Main Distribution Facility) (Core) → มีจุดย่อยไปยังอุปกรณ์ที่ต้องการ

③ Workshop S. (ออกแบบ) → ต่อตัว @ IDF (Intermediate Distribution) → เชื่อมต่อ cross-link access ที่ต้องการ

◦ Collision detection issue (ปัญหาต้องตรวจสอบ)

◦ Segmentation issue (ปัญหาต้องตรวจสอบ) ทำให้ต้องมีปัญหาในตอนนี้ ปัจจุบันที่ต้องใช้ VCC (vertical cross-connect): optical fibre 8 → MDF ↔ IDF

◦ Broadcast domain issue → ไม่สามารถส่ง broadcast ที่ต้องการได้ HCH (Horizontal) ~ UTP → Distribution → Access

◦ Segmentation ที่ process split single collision domain → Smaller collision domain ลดจำนวน collision บน LAN segment แต่ device ที่ต้อง bridge, SW

◦ Broadcast domain รับจากอุปกรณ์ port but router (L3) ผ่าน filter/segment broadcast ทำให้ไม่สามารถส่ง broadcast ที่ต้องการได้

▷ SW Environment

▷ SW Operation ① learning: รับ frame แล้วจะเรียก Source MAC Addr. ต่อไปที่ port ใหม่ + reset Aging

② Aging: 老化 MAC Addr. → if หมด → ลบ dest. ออกจาก table

③ Flooding: รับ frame แล้วนำ port ของ SW when frame ที่เป็น broadcast, 2) multicast, 3) unknown unicast

④ Forwarding: 送到 dest. (ตรวจสอบใน table)

⑤ Filtering: if ตรวจสอบ frame ที่ dest. ทาง port ที่ไม่ถูก aging ให้ dest. (source & dest. ไม่ same interface) → ไม่ filter ที่ต้องการ

▷ SW Methods: ① Store & Forward SW → check CRC ต่อ error หรือ? → yes: ทิ้ง → no: ผ่าน, auto buffer

② Cut-Through SW → check 6bytes ของ frame (dest, src บวกช่อง 12 byte (long) 10ms), No FCS & auto buffer

↳ 2 Mode: ① fast-forward ~ 12 byte ② Fragment-free ~ 64 byte: < 64 = 7ms → 64ms → ไวด์

▷ SW Domains: ① Collision Domains → domain ที่เกิด col. ทางเดียวที่จะส่งข้อมูลจากผู้รับ "ที่ SW ผ่านตัวเอง"

② Broadcast: ~ ～ ～ ～ broadcast → อยู่ domain ใดเดียวกันจะได้รับ "ที่ router ที่ต้องการ"

▷ Basic SW Concept & Configuration manage intf: S(config-if)# interface vlan num: default Gateway: ip address ip subnet S(config-if)# ip default-gateway ip

▷ Basic SW Config SW Boot Sequence = Same Router [S(config-if)# no shutdown]

[Verify Port Config] Preparing of Basic SW Management: SW ต้องเป็น SVI (SW Virtual Interface) → VLAN

Show int fo/0/ → Config SW port → Duplex Communication ① Full ② Half (SW ต้องเป็นตัวเดียวกันเดียวกัน)

startup-config/running-config/flashing/version, เผาไฟ intf → S(config-if)# duplex full → #speed 100 (ไม่ระบุ speed)

/history/ip fo/0/mac-address-table → Auto-MDIX: ป้องกันจะต้องต่อสาย cross-over บต. ต้องกับปุ่มสวิตช์ตรง

เผาไฟ intf → S(config-if)# duplex auto → #speed auto → #mdix auto

▷ SW Security: Security Remote Access → SSH (Secure Shell) TCP port 22, Telnet: TCP port 23

◦ Config: S_change_hostname(config)# ip domain-name 8.8.8.8 → crypto key generate rsa → #username 8.8.8.8 password pass →

#line vty 0 15 → line# transport input ssh → line# login local [Verify SSH: show ip ssh, show ssh]

▷ SW Port Security → กันคนอื่น policy ต่อ MAC Addr. ไม่รับเข้า/ออก ได้ or ไม่ได้

S(config-if)# switchport mode access → #switchport port-security → เลือกมา 1 ไป-มา

◦ Type of Secure MAC Addr.: ① static: S(config-if)# switchport port-security mac-address MAC-ADD

② dynamic: S(config-if)# switchport port-security mac-address sticky → learn frame ที่ต้องการ record ไว้

◦ กรณีไม่รับ MAC : # switchport port-security maximum MAX Forward Traffic Send Syslog Displays Error Msg Increase Violation Counter Shutdown Port

◦ Violation mode: ① protect: security violation protect mode

② restrict: security violation restrict mode → ไม่ต้องรับก็ต้องไม่หัก

③ shutdown: ~ shutdown ~ default

	Forward Traffic	Send Syslog msg	Displays Error Msg	Increase Violation Counter	Shutdown Port
No	No	No	No	No	No
Yes	No	No	No	Yes	No
No	No	No	No	Yes	Yes

[Verify: show port-security int fa0/0, show port-security address]

▷ Addr. Resolution Protocol (ARP): ARP Cache เก็บ MAC Addr. ที่อยู่ LAN ใด map นามสกุล dest. (if) ไปทางอุปกรณ์ MAC

▷ IPv4: Classless (เริ่มต้น P.1-2): Variable Length Subnet Masking (VLSM): แบ่งจาก gw ไปทาง gw

- Fixed ~: แบ่งจาก gw ไปทาง gw → เลือก gw: ชั้นในอยู่ gw

Chapter 8 LAN Redundancy & Spanning Tree Protocol (STP)

▷ Issue with Layer 1 Redundancy: ① MAC Addr. instability → MAC Addr. table หายเสียหาย bec. เปลี่ยน IP ประจำตัว

② Broadcast storm → จำนวนมาก ต่อไปเรื่อยๆ ③ Multiple frame transmission → หลาย unknown unicast → ไม่ถูก dest. ได้รับแต่ frame บต. ต่อ 1 frame

▷ STP → ต้องเลือก 1 บต. block port → block ต่อไป → ไม่ให้เกิด traffic ที่นี่ ไม่ต้องป้องกัน สำคัญมาก. ① BPDU flag (#priority ต่ำสุด)

② BID ต่ำสุด

③ Path Cost <

④ Sender's BID <

⑤ Sender's Port <



□ Config: (1) S1, S2 ปั๊บ R8)	Link Speed: 10 Gb/s = $\frac{cost}{2}$, 1Gb/s = 4, 100 Mb/s = 19, 10 Mb/s = 100				
วงจร I: S1(config)# spanning-tree VLAN 1 root primary	วงจร II: S3(config)# spanning-tree VLAN 1 priority 24576				
S1(config)# ~ Secondary [Verify: show spanning-tree]	(byte) (12 bit) (6 byte) (byte) (6 byte)				
D แนว Extended System ID: B. Priority \Rightarrow B. Priority (VLAN) + Extended Sys ID (VLAN)	MAC Addr.: BID				
▪ PVST+ (พื้นที่ IEEE 802.1D STP) \Rightarrow เผื่องการ load balancing [โดยเน้น root/vlan] [Verify: show spanning-tree active] = 8 byte					
▪ Rapid PVST+ \Rightarrow ณ Alternate Port (ไม่ใช่ block ไม่ลาก but. can รักษาอุบัติเหตุในส่วน)					
↳ config.set Edge Port @ port ของ host, router :: s1(config-if)# spanning-tree portfast port ที่ไม่ต้องกังวล					
↳ link type: port ที่เชื่อมต่ออย่าง point-to-point :: -if# spanning-tree bpdu guard enable \Rightarrow BPDU ไม่ถูก弃用					
↳ config: s1(config)# spanning-tree mode rapid-pvst \Rightarrow ผ่าน intf กันไป point-to-point# spanning-tree link-type clear spanning-tree detected protocol : ล้าง clear all \leftarrow point-to-point					
Protocol	STP	PVST+	RSTP	Rapid PVST+	MSTP
Standard	802.1D	Cisco	802.1W	Cisco	802.1s Cisco
Resources Needed	Low	High \rightarrow ค่าความต้องการของ VLAN	Medium	Very high	Medium or high
Convergence	Slow	Slow \rightarrow ค่าความต้องการของ VLAN	Fast	Fast	Fast
Tree Calculation	All VLANs	Per VLAN	All VLANs	Per VLAN	Per Instance

Chapter 9 VLANs & Inter VLAN (Virtual LAN)

- ▷ VLAN = ภูมิ partition (แบ่งช่องทาง: อุปกรณ์ที่อยู่ใน domain หรือ broadcast domain) L2 # ไฟ SW ไม่สนใจ ต้องต่อ interface ให้เป็น VLAN เดียวกัน
- ▷ ข้อดี: -security ตั้งค่า, -cost, -bus. cost, -broadcast domain เล็กๆ, -ประสิทธิภาพ IT, รักษาไฟล์ [Verify: show vlan brief]
- In a Multi-SW Environment
- VLAN Trunk: set intf เชื่อมต่อระหว่าง Port ที่มี VLAN \Rightarrow can carry มากกว่า 1 VLAN
 - ③ Loconfig: ผ่าน intf \rightarrow if# switchport mode trunk [Verify: show int folo switchport] Native (base) VLAN: ที่มี tag & Untag Tag ที่ไม่มีอยู่ใน VLAN when it's a tag
 - Tagging Ethernet Frames (IEEE 802.1Q): Ethernet frame \Rightarrow Dest MAC Src MAC Tag Type / Length Data FCS Trunk
 - Assignment: VLAN number \Rightarrow 1-1005 กรณี config@ vlan.dat (flash) ① สร้าง: S1(config)# vlan vlan.num \rightarrow S1(config)# name ชื่อ \Rightarrow 1006-9096 \rightarrow @running-config หรือ NVRAM ② S#vlan database \rightarrow (vlan)# vlan num name ชื่อ
 - ④ assign port ให้ VLAN: ทำ intf \rightarrow if# switchport mode access \Rightarrow switchport access vlan num (if ที่ต้องมี no shutdown) (if ที่ต้องไม่มี no VLAN num)

▷ Inter-VLAN Routing \Rightarrow router set ไฟ trunk โดยตั้งค่า "sub interface"

- config: ① set basic routing (set ip address, no shutdown) [Verify: show vlan, show ip route, show running-config]
 ② Rconfig# interface g0/0.10 \rightarrow VLAN + subif# encapsulation dot1q 10 \rightarrow #ip address ip subnet-mask

Chapter 10 VTP (VLAN Trunking Protocol) \rightarrow จัดการ VLAN & NAT (NW Addr. Translation)

D VTP [msg: ISL or IEEE 802.1Q] \rightarrow n.manage SW VTP: 1. ปั๊บ manage ให้ domain		*locally significant only			
Operation	Value	Src VTP Msg	Listen to VTP Msg	Router VLAN	Remember VLAN
① update VTP	จะอัปเดต revision number 32 bit (0-4294927295) 1% every 10 sec	Yes	Yes	Yes	Yes
② mode: ① Server \rightarrow can add, remove, rename VLAN ทุกตัวใน domain ของตัวเอง	Yes	Yes	No	No	No
② Client \rightarrow รับข้อมูล VTP แล้ว process, รับ VTP msg ของทุก trunk	No	No	Yes*	Yes*	Yes*
③ Transparent \rightarrow can add, remove, rename VLAN ของตัวเอง, ไม่รับคำสั่ง					

□ Config: 2 แบบ ฝั่งเดียว: 1) SW Cisco 2) SW Cisco ที่ต้องต่อ SW 3) ต่อ domain 4) ต่อ mode

- ① In global conf: S1(config)# vtp version 2 \Rightarrow #vtp domain ชื่อ \rightarrow #vtp password รหัส \Rightarrow #vtp mode server | client | transparent
 ② In VLAN conf: S1(vlan)# vtp v2-mode [Verify: show vtp status, counters] \Rightarrow #vtp server | client | transparent

▷ Pruning \rightarrow manage ให้ traffic ที่ไม่ต้องการ interface ต่อ config ให้ intf ของ SW ที่ต้อง remove ออก \Rightarrow ไม่ต้องซ้ำ S(vlan)# vtp pruning \rightarrow ผ่าน intf \rightarrow S(config-if)# switchport trunk pruning vlan remove vlan-num

▷ DNAT \Rightarrow ให้ ip private \leftrightarrow publish/real ip

- Terminology: 4 type: ① Inside local Addr. (private ip)
 ② Outside local Addr. (publish ip)
 ③ Inside global Addr.
 ④ Outside global Addr. \rightarrow same local
- | Private Internet Addr. [RFC 1918] | | | |
|-----------------------------------|-------------------------------|----------------|--|
| Class | RFC 1918 Internet Addr. range | CIDR Prefix | |
| A | 10.0.0.0 - 10.255.255.255 | 10.0.0.0/8 | |
| B | 172.16.0.0 - 172.31.255.255 | 172.16.0.0/12 | |
| C | 192.168.0.0 - 192.168.255.255 | 192.168.0.0/16 | |

▪ type: ① Static: กับ config [map: 1:1] \Rightarrow ① Rconfig# ip nat inside source static local-ip global-ip

② Dynamic: กรณี pool ของ Global / Real ip [map: ที่ต้อง many:1] ไม่ต้องก่อไปซ้ำๆ ② #ip nat pool ที่ start-ip end-ip

③ PAT (Port Address Translation) \rightarrow port ที่ต้องร่วมกับ nw addr. [map: many:1] netmask netmask | prefix-length prefix } หรือ II (single Addr.)

④ สร้าง ACL ⑤ ip nat inside source list ACL-num int folo overload

[Verify: show ip nat translations]

□ Config: 3 ส่วน ① NAT

- เพิ่มที่ intf { ② INSIDE: Rconfig-if# ip nat inside
 ③ OUTSIDE: Rconfig-if# ip nat outside

④ set ACL

⑤ ip nat inside source list ACL-num pool ที่ overload (กรณี Dynamic \rightarrow PAT)

Operation

Initial Route Discovery (เริ่มต้น): ① R1 say hello กับ neighbor router ② R2 ต้องส่ง hello or update info
 ③ R1 รับ ack & update info. ④ R2 DUAL คือ route best route and update routing table
 Metrics = BW [lowest], Delay [slow], Reliability [worst], Load [worst] = value : show int
 Default Composite: metric = $[K_1 \cdot bw + K_2 \cdot bw + K_3 \cdot delay] + 256$
 Formula: $= \left[\frac{(10,000,000)}{bw} + \frac{\text{sum of delay}}{10} \right] + 256$ complete: $= \frac{[K_1 \cdot bw + K_2 \cdot bw + K_3 \cdot delay] + 256}{(256 - load)} \cdot \frac{K_5}{\text{reliability} + K_4}$

- R(config-router) # metric weights to 5 k1 k2 k3 k4 k5 - set bw: เบอร์ intf. \Rightarrow if # bandwidth kbits/bn

DUAL and the Topology Table (BFSM Finite State Machine) ในการเลือกเส้นทาง

→ show ip eigrp topology [all-link], show ip route

✓ Successor (S) [router ที่ dest. ล้วนๆ] = neighbor router ที่จะส่งถูกต้องdest ที่ cost min

✓ Feasible Successor (FS) [จะได้มา Feasible condition (if RD < FD \Rightarrow FS)] = Backup path (เส้นทางสำรอง)

✓ Reported Distance (RD) [distance ที่ neighbor รายงาน, report distance ที่ใน表] = "advertised distance" ของอัปเดต

→ dest ที่ cost min ของ route hop

✓ Feasible Distance (FD) [distance ที่ใน S] > rd distance ที่คำนวณได้ใน dest. nw ที่ min lowest

▷ IPv6

▷ IPv4 Issue

△ Need for IPv6 → เน็ตเวิร์ก ip จำกัด (private ip, NAT), ใช้ใน IoT ได้

△ ไม่ใช้共存 (coexistence)

- Migration IPv4 → IPv6 Techniques: ① Dual Stack = run ทั้งคู่ if สองบ้านใช้ ไลน์ของ user

② Tunneling (อย่างที่ v6 but. core ไม่ support)

= แปลงหน้าที่ของ v4 ให้เป็น v6 ไม่ต้องต่อ

③ Translation (ถ้าใช้ NAT) = IPv6 → IPv4

▷ IPv6 Addressing: 128 bit แบ่ง 8 ส่วน [1 ส่วน ที่ 2 byte = 16 bit] \Rightarrow represent base 16 ทุกๆ 4 bit

○ ตัวแบบสำหรับการเขียนIPv6

Rule 1 - Omit Leading 0s \Rightarrow หัวตัวอักษร partition "0" ที่อยู่ต่อเนื่องกันติดกัน เช่น 0000x, 00xx, 0xxx

Rule 2 - Omit All 0 segment \Rightarrow หาก segment ที่มี "0" ติดกันช่วงๆ ตัวอย่าง ":" เป็นตัวที่ต้อง only

▷ Type of IPv6 Address

○ IPv6 Addr. Type ① unicast: ② link-local \Rightarrow ไม่ต้องต่อที่นี่ (@ interface เดียวเท่านั้น)

③ Global Unicast \Rightarrow 3 bit แรก = "001" or "2000::/3"

④ Multicast \Rightarrow static config: ipv6 address ipv6-addr/prefix-length \Rightarrow no shutdown

⑤ Anycast \Rightarrow สำหรับ device

△ IPv6 Prefix Length = 0-128, most LANs is /64 bcc. LAN เล็กๆ 64 bit

▷ IPv6 Routing

○ Config static route

next-hop-addr exit-interface

R(config) # ipv6 route ipv6-prefix/prefix-length { ipv6-addr | exit-intf } \Rightarrow ต้องมี

* ลักษณะ config ให้เรา routing ipv6 ipv6 unicast-routing \Rightarrow ต้องมีต่อไปนี้

△ verify: show ipv6 route static, show ip route ipv6, show running-config|section ipv6 route

△ Default Static IPv6 Route

R(config) # ipv6 route ::/0 { ipv6-addr | exit-intf }

- verify: show ipv6 route static

○ config EIGRP for IPv6

R(Config)# ipv6 unicast-routing

R(config)# ipv6 router eigrp AS-#

R(config-rtr)# eigrp router ip 2.0.0.0 \Rightarrow กำหนด IP

R(config-rtr)# no shutdown

△ network command ต่อ intf \Rightarrow ipv6 eigrp AS-#

but. passive-interface \Rightarrow ต้องต่อที่ไม่ใช่ global config same as

△ verify: show ipv6 eigrp neighbors, show ipv6 protocols, show ipv6 route

**Router****① basic configuration**เมื่อ intf (เช่น Giga, Fa) \Rightarrow Rconfig-if # ip address nw-ip subnet-mask \Rightarrow no shutdownเมื่อ intf (serial ต่อ DCE) \Rightarrow ~ \Rightarrow Rconfig-if # clock rate 56000 \Rightarrow ~verify: show running-config \rightarrow ดู config ที่ตั้งไว้show startup-config \rightarrow ~ set ~show ip route \rightarrow ดู routing table

show interface

show ip interface

} show info. ของ interface

show ip interface brief show intf. แบบง่าย ชื่อ ip ?, status ?

② protocol+ static routing ให้ intf \Rightarrow Rconfig-if # ip route nw-ip subnet-mask {ip addr. | exit-intf} next hop addr exit-interface

△ Default route ~

ip route 0.0.0.0 0.0.0.0 {ip addr. | exit-intf}

+ Dynamic routing

- Interior Gateway P.

x Distance Vector Routing P.

■ RIP: R(config)# router rip \Rightarrow R(config-router)# network nw-ip

verify: show running-config, show ip route, show ip protocols, debug ip rip

passive intf: R(config-router)# passive-interface intf-type intf-num

เมื่อ RIP \Rightarrow static: R(config)# router rip \Rightarrow R(config-router)# {redistribute static}

{เมื่อ router ต้องการ set default route @ intf ที่ต้อง protocol default-information originate}

■ RIPv2: R(config)# router rip \Rightarrow R(config-router)# version 2 \Rightarrow no auto-summary \Rightarrow network nw-ip

verify: same RIP แล้ว show ip interface brief

■ EIGRP: R(config)# router eigrp AS-# \Rightarrow -router# eigrp router-id \Rightarrow network nw-ip [wildcard]

passive intf: R(config-router)# passive-interface intf-type intf-num

verify: show ip protocols, show ip eigrp neighbors, show ip route, show ip eigrp topology [all link]

metrics: R(config-router)# metric weights tos k1 k2 k3 k4 k5

- set bw: ให้ intf \Rightarrow R(config-if)# bandwidth kbytes-bw-value

x Link State Routing P. 1-65,535

■ OSPF: R(config)# router ospf process-id \Rightarrow -router# router-id 1.1.1.1 \Rightarrow network nw-ip wildcard area numset bw: ให้ intf \Rightarrow R(config-if)# bandwidth b4

set cost: ~ # ip ospf cost 15625

passive intf: R(config-router)# passive-interface intf-type intf-num

verify: show ip protocols, show ip ospf neighbor, show ip ospf int brief, show ip ospf

au: clear ip ospf process

redistribute (OSPF \leftrightarrow default route): R(config)# ip route 0.0.0.0 0.0.0.0 loopback N

~ # router ospf process-id

 \Rightarrow R(config-router)# default-information originateredistribute (OSPF \leftrightarrow ตัวอย่าง): R(config)# router ospf process-id \Rightarrow R(config-router)# redistribute ?**③ ตั้งค่า**

- ACL: if ชื่อ Name : R(config)# ip access-list [standard | extended] name

set ACL: R(config)# access-list ACL-num {permit | deny | remark} source [source-wildcard] [log]

set @intf: ให้ intf \Rightarrow Rconfig-if# ip access-group {ACL-num | ACL-name} {in | out}

au: no access-list ACL-num

verify: show ip interface so/0/0, show access-lists class

- Securing VTP with standard IPv4 ACL: R(config-line)# access-lists ACL-num {in cvrf-aslo} | out }

- Extended IPv4 ACL:

R(config)# access-list ACL-num {deny | permit | remark} protocol source [source-wildcard] [operator operand] [port num or name]
destination [dest-wildcard] [operator operand] [port port-num or name] [established]

- DHCP: R(config)# ip dhcp excluded-address ip-addr-start ip-addr-end

~ # ip dhcp excluded-address ip-addr-gateway

~ # ip dhcp pool LAN-POOL-1

R(dhcp-config)# network nw-ip subnet-mask

~ # default-router ip-addr-gateway

verify: show running-config | section dhcp, show ip dhcp binding, show ip dhcp server statistics



Switch

① basic configuration

- management intf.: `S(config) # interface vlan num` \Rightarrow `-if# ip address ip-addr subnet-mask` \Rightarrow `no shutdown`
- default gateway: `S(config) # ip default-gateway ip`
- verify: `show running-config`, `show ip interface brief`

② configure switch port

- duplex communication: `(V)intf \Rightarrow S(config-if) # duplex full` \Rightarrow `speed 100`
- auto-MDIx: `(V)intf \Rightarrow S(config-if) # duplex auto` \Rightarrow `speed auto` \Rightarrow `mdix auto`
- verify: `show int [intf-id]`, `show startup-config`, `show running-config`, `show flash`, `show version`, `show history`, `show ip [intf-id]`, `show mac-address-table`
- security Remote Access
 - + SSH (TCP port 22): `S(config) # ip domain-name cisco.com` \Rightarrow `crypto key generate rsa`
 - \Rightarrow `username admin password cisco` \Rightarrow `(line vty 0 15 \Rightarrow -line) # transport input ssh` \Rightarrow `login local`
 - verify: `show ip ssh`, `show ssh`

- switch Port Security: `(V)intf \Rightarrow S(config-if) # switchport mode access` \Rightarrow `switch port-security`
- + static secure MAC Addr: \Rightarrow `switchport port-security mac-address MAC-ADD`
- + dynamic ~ \Rightarrow ~ maximum MAX
- + max MAC Addr, ~ \Rightarrow ~ violation {protect | restrict | shutdown} mode
- + violation mode \Rightarrow ~ sticky
- + verify: `show pppoe-security int fa0/3`, `show port-security address`

③ STP

1. `S(config) # spanning-tree VLAN 1 root {primary | secondary}`

2. `S(config) # spanning-tree VLAN 1 priority 24576` (기준 푸리티 설정)

+ verify: `show spanning-tree [active]`, `show running-config`

□ Rapid PVST+

- + Port Fast: `(V)intf \Rightarrow S(config-if) # spanning-tree portfast`
- + BPDU Guard: `(V)intf \Rightarrow S(config-if) # spanning-tree bpduguard mode`
- + config: `S(config) # spanning-tree mode rapid-pvst` \Rightarrow `(V)intf. # spanning-tree link-type point-to-point`
- + clear STP: `S# clear spanning-tree detected-protect`

④ VLAN

1) set VTP mode: `S(config) # vtp version 2` \Rightarrow `vtp mode {server | client | transparent}` \Rightarrow `vtp domain name`

\Rightarrow `vtp password pass`

2) set trunk `(V)intf. \Rightarrow S(config-if) # switchport mode trunk`

3) assign VLAN @ Server: `S(config) # vlan num` \Rightarrow `name name`

4) assign intf.: `(V)intf. \Rightarrow S(config-if) # switchport mode access` \Rightarrow `switchport access vlan num`

5) set inter-VLAN: `R(config) # int f0/0.10` \Rightarrow `description VLAN10` \Rightarrow `encapsulation dot1q 10`

\Rightarrow `ip address ip subnet`

+ verify: `Show vlan name name`, `show vlan summary`, `show int vlan num`, `show int f0/0`,
`show vlan [brief]`

⑤ NAT

- static: `R(config) # ip nat inside source static local-ip global-ip` \Rightarrow `(V)intf. \Rightarrow -if# ip nat {inside | outside}`

- dynamic: `R(config) # ip nat pool name start-ip end-ip {netmask netmask|prefix-length prefix}`

\Rightarrow `access-list ACL-num permit source [source-wildcard] \Rightarrow ip nat inside source list ACL-num`

`pool name overload \Rightarrow (V)intf. \Rightarrow S(config-if) # ip nat {inside | outside}`

□ PAT = same dynamic 맵핑과 "overload"