

..... LAN ————— straight WAN
..... LAN ————— cross WAN

2. Network Diagrams แบบร่าง NW ที่ใช้กัน มีอะไรบ้าง?

2.1 Physical : บนตัว port/interface ที่เราสามารถมองเห็นได้ ตั้งแต่ชั้น Layer 1 (physical) ถึงชั้น detail (doc)

2.2 Logical : บน ip (File Transfer protocol) → ไปยัง file ของ client หรือ server

3. Network Protocol TCP/UDP, FTP, ARP, STMP, POP3, IMAP, ICMP -- (internet control message protocol) → ping (โปรโตคอล cmd)
(Addr. Resolution Protocol) → Map ระหว่าง IP addr. กับ MAC addr. โปรโตคอล

4. Types of Network LAN < campus NW < WAN

4.1 SW ① switch ผู้จัดการ ② Router เพื่อไปทางต่อไป

4.2 size ① small home nw → สำหรับบ้านหรือห้องเดียว

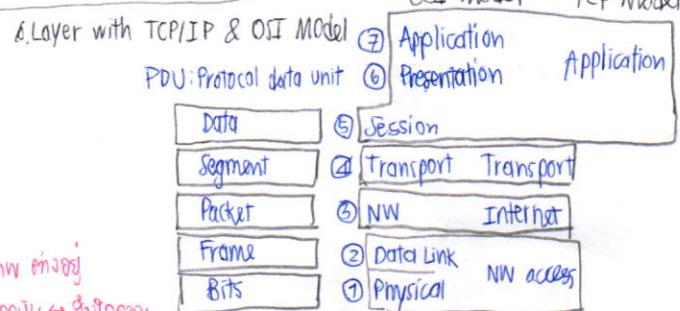
② small office/home office → config ทางต้นออกที่ต้องการ

③ Medium to Large → มีเครื่อง Mac 100-1000 (很多)

④ World wide NW คือ internet

4.3 infrastructure ① Local Area NW (LAN) → ผู้管 Admin กำหนด policy/nw ผ่านๆ ให้
② Wide Area NW (WAN) → ผู้管 Admin ผ่านๆ ผู้管 Admin

⇒ ต้องมีการเชื่อมต่อ彼此彼此



⇒ MAN : Metropolitan Area NW (MAN), Wireless LAN (WLAN), Storage Area NW (SAN), Personal Area NW (PAN)

5. Reliable Network

5.1 fault tolerance → ทนต่อการล้มเหลว

5.2 scalability → สามารถปรับเปลี่ยนขนาดได้ ไม่ว่าจะเพิ่มหรือลด User จำนวนมาก

5.3 Security → ต้องการเข้าถึง

5.4 quality of service (QoS) → ให้บริการด้วยคุณภาพที่ต้องการ

Chapter 2 Basic Router Configuration

1. Port Address @L4: Internet Assigned Number Authority : IANA

0-1023: requesting entities "well known ports" destination port

1024-49151: registered port = publish ภายนอกโลก

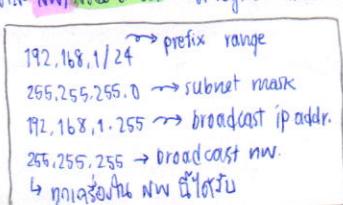
49152-65535: dynamic or private port "Randomly generate" source port

Ex. 20: FTP(data), 21 FTP(control), 25: SMTP (Simple Mail Transfer), 53: DNS (Domain Name Server) TCP/UDP, 80(www) HTTP, 81: HOSTS NAME Server

2. Logical Address : IP Address (IPv4) @ L3

- 5 class : A,B,C,D,E → Reserved (not) เนื่องจากต้องมี max 1024 workstation required
→ Multicast host → multicast Addr. → ต้องใช้เฉพาะใน LAN เท่านั้น

- 16 bit NW/Nodes or COM → logical name (domain names) & IP unique.



| | | | | | |
|----------------|----|------|------|------|----------------------|
| class A : 0 | NW | Host | Host | Host | 0-127 |
| class B : 10 | NW | Host | Host | Host | 128-191 |
| class C : 110 | NW | NW | Host | Host | 192-223 |
| class D : 1110 | | | | | 224-239 multicast |
| class E : 1111 | | | | | 240-255 experimental |

private addressing

→ IP can reuse อยู่ในตัว

→ อยู่ในตัวของตัวเองใน internet อยู่ในตัว

| | RFC 1918 Internal Addr Range | CIDR Prefix |
|---------|-------------------------------|-----------------|
| Class A | 100.0.0 - 10.255.255.255 | 10.0.0.0 /8 |
| Class B | 172.16.0.0 - 172.16.255.255 | 172.16.0.0 /12 |
| Class C | 192.168.0.0 - 192.168.255.255 | 192.168.0.0 /16 |

3. Physical Address : MAC Address

- Ethernet : 48 bit จํานวน 2 = 12 บิต จํานวน 16 จํานวน 0x จํานวน

- institution IEEE → มีตัว 3 byte (24 bit) code "Organizationally Unique Identifier (OUI)"
→ 2 บิต → ① ภูม MAC สำหรับ NIC ของ Ethernet device จํานวน 16 → OUI ที่ 3 bytes เท่านั้น
→ ② ภูม MAC ที่ same OUI จําเป็นต้อง unique ที่ 3 bytes จําเป็นต้อง

Message Delivery

◦ Unicast = ส่งมาให้เครื่องคอมพิวเตอร์ที่หนึ่งเดียว

◦ broadcast = ส่งไปทุกเครื่องที่อยู่ใน LAN หรือ DHCP, ARP ที่ hw ต้องการ

◦ multicast = ส่งไปทุกเครื่องที่อยู่ใน LAN และที่ได้รับ service ที่

→ 13 บิตต่อตัวเลข (01-00-5E) XX-XX-XX



◦ Cisco IOS (Internet Work Operating System)

- function 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.

② Run boot loader 5W minios

ROM ③ Boot loader does low-level CPU initialization

Flash ④ Boot loader initializes the flash file system

FTP server ⑤ Boot loader locates & load a default IOS in run via RAM

◦ 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) "# → ① Global Configuration Mode "(config) "#"

② Other Configuration Mode "(config-mode) "#"

◦ The Command Structure

① Context Sensitive Help: "?"

② Command Syntax Check = enter 110# show interface brief

③ Hot Keys and Shortcut

④ IOS Examination Commands → show ...

• Getting Basic ① Mōo hostname ② Router interface addressing ④ Mōo config + n.180.info ⑤ save config

↳ Router(config)#hostname name ↳ Banner Msg. Router(config)# banner motd# text #

↳ Securing Device Access: Enable password/secret, console pass, vty pass, Encrypting pass display

③ Addressing Devices

1) Assign interface to config

- physical interface/loopback interface

- Router(config)# interface type port

↳ type slot/port

- switch virtual interface (SVI) -> Switch(config)# interface vlan number

2) set ip addr.
router(config-if)# ip address ip address subnet mask
↳ no shutdown

④ Verify Connectivity → check connection section, include, exclude, begin
Router# show running-config → get config of all UUUV
show startup-config → get set configuration
show ip route → of routing table
show interface → show info via interface name
show ip interface → show into via interface name
show ip interface brief → show interface number, set?, status
traceroute → ping
ping → traceroute
PC → ping
traceroute → route ping
route ping → nslookup

Chapter 3 Static Routing & Dynamic Routing Protocol

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

▷ Packet Forwarding Methods ① Process Switching = ผ่าน packet ผ่าน router → process @ CPU & von interface 11uu

② Fast switching = ผ่าน packet forward 100% 11uu ③ Cisco Express Forwarding (CEF) = forward packet 100% 11uu

• Connect Devices

▷ Default gateway → 11uu ① first usable host (1) ② last usable host (.254)

→ mōo assign gw nw 11uu

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

② Dynamically Assigned IP Addr. → mōo assign ip of 11uu nw & DHCP (Dynamic Host Configuration Protocol)

11uu ip 11uu

gwip ③ subnet nw = nw ip & subnet mask

Ex 192.168.1.0 = 192.168.1.1 & 255.255.255.0

④ broadcast nw = 11uu 11uu 11uu host gw subnet nw 11uu 11uu

= 192.168.1.255

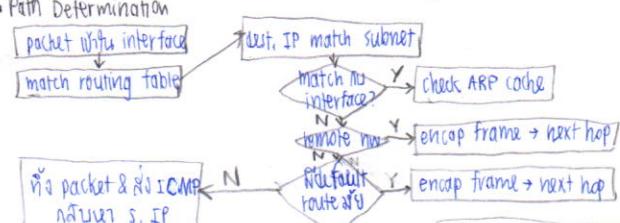
ip 11uu

subnet

• Switching Packet between NW

ผู้ต้องการ : 11uu dest. ip (L#2) → 11uu 11uu 11uu 11uu MAC Addr. → 11uu dest. MAC(L#2)

• Path Determination



The Routing Table

10.1.1.0/24 [192.168.1.12] via 209.165.200.226, 00:00:05, serial 0/0/0

Legend

- identifies how the nw was learned by the router. 11uu 11uu 11uu 11uu
- identifies the destination nw 11uu 11uu
- identifies the administrative distance (trustworthiness) of route source
- identifies the metric to reach the remote network (cost)
- identifies the next-hop IP address to reach the remote nw 11uu 11uu 11uu 11uu
- identifies the amount of elapsed time since the nw discovered.
- identifies the outgoing interface on the router to reach the destination nw 11uu 11uu 11uu 11uu

• Classful Addressing → update emul class

• Classless Inter-Domain Routing

▷ summarization : 11uu 11uu 11uu 11uu

▷ offset first 11uu no ip 11uu 11uu 11uu 11uu

② Group bit 11uu 11uu 11uu 11uu

11uu 11uu 11uu 11uu 11uu 11uu 11uu 11uu

Chapter 4 Distance Vector Routing Protocol RIP ver 1

▷ Dynamic Routing Protocol

- fn : share info ระหว่าง router → auto update routing table when topologyเปลี่ยน (เมื่อมีปัญหา) → un best path
- Purpose : บน remote nw (ห่วงโซ่ภายนอก) แลกเปลี่ยน routing info → เลือก best path ไป dest. nw - can use new best path ถ้า path หาย/ไม่ถูกต้อง
- component :
 - ① Algorithm : 99% ใช้ exchange routing info & best path
 - ② Routing protocol msg : ให้ข้อมูล neighbor & 110 ไปยัง routing info. (best path)

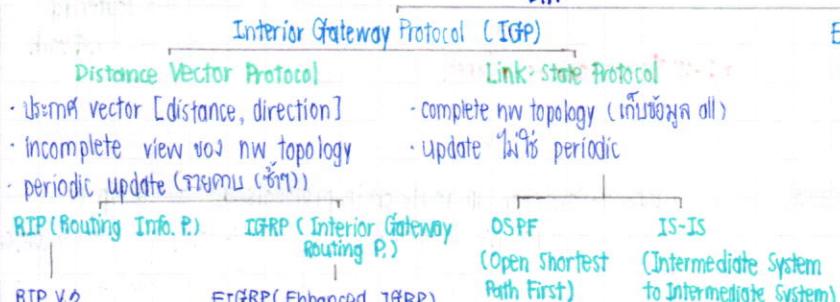
□

Dynamic Routing

- 1. คุณสมบัติ config
 - 2. Required A.Y Admin
 - 3. Topology change
 - 4. Scaling
 - 5. Security
 - 6. Resource Usage
 - 7. Predictability
- ข้อดี nw (same เท่าเดิม)
- Advanced (bec. config basic → nw 99%) + ซับซ้อนอาจมีปัญหา
- ข้อเสีย auto
- mean simple & complex (router ต้องต่อ direct ที่ต้องการ)
- ข้อเสีย CPU, mem (เก็บ routing info), link bandwidth
- ข้อเสีย Route & current topology

▷ Classifying Routing Protocols

DRP



▫ 2 type

- ① Classful Routing P. update ตาม class ให้ subnet mask ที่ routing update
- ② Classless Routing P. ที่ subnet mask ที่ routing update
- Convergence : ประมาณ 10 sec. when routing table ของ all router มีสходимกัน (ไม่เปลี่ยนแปลง)
- 2 type : 1) Slower : RIP & IGRP ... 2) Faster (เมื่อต้องการเปลี่ยนแปลงดี) : EIGRP & OSPF

▷ Routing Protocol Metrics

- Metric : ตัวที่กำหนดโดยตัว dest. NW จะเลือก best path ใหม่ เช่น Hop count, BW, Cost, Delay, Load, Reliability
- Load balancing : nw ที่ต้องการ > 1 ที่ metric เท่ากัน → เลือกที่ก. ที่งาน เท่ากัน มาก่อน

▷ Administrative Distance of a Router (AD) ⇒ 99% เรียก protocol ใหม่ routing

- ตัวอย่าง : ตัวที่เราต้องการจะไป particular (เฉพาะเจาะจง) route

| Route Score | Connected | Static | Internal | EIGRP | OSPF | RIP | EIGRP summary route | External BGP | IGRP | IS-IS | External EIGRP | Internal BGP |
|-------------|-----------|--------|----------|-------|------|-----|---------------------|--------------|------|-------|----------------|--------------|
| AD | 0 | 1 | 90 | 110 | 120 | 120 | 5 | 20 | 100 | 115 | 170 | 200 |

▷ Distance Vector Routing Protocol Ex. RIP, IGRP, EIGRP

- Distance Vector Technology ตัวที่ 2 ที่เราได้ ① vector or direction , ② ระยะทางครุภาระ ③ Distance to final dest. (cost)
- ลักษณะ : รีบ : periodic (定期的) update, neighbor (เพื่อนบ้าน), broadcast (255.255.255.255) update, ลง routing table all ไปอัปเดต
- ลักษณะที่สำคัญ : ต้อง check in DV ต่อไปดี? ① Time to convergence → เวลาที่เป็น steady state ของ routing table ที่ต้องการเปลี่ยนแปลง ② Scalability ขนาดมากถูกต้อง ③ Resource usage ④ Implementation & maintenance

▷ NW Discovery (พื้นที่)

- 3 stage ① Cold State : Router Initial start up
- ② Initial Exchange of Routing info. → ทำตารางค่าไปร์เซนต์
- ③ Exchange of Routing info → update (ที่ hop count) routing info. → บน router จึงต้องการที่ต้องการ

| Network | Interface | Hop |
|----------|-----------|-----|
| 10.2.0.0 | S0/0/0 | 0 |
| 10.3.0.0 | S0/0/0 | 0 |
| 10.1.0.0 | S0/0/0 | 1 |
| 10.4.0.0 | F0/0/0 | 1 |

update #3
R2 S0/0/1

▷ Routing Table Maintenance

- Periodic update : RIP update timer (default 30s), Invalid timer (info ถูก lost) (default 180), Hold down timer (ต้อง down → hold ไว้ 10 sec. up ใหม่ก่อน) (default 180), Flush (ลบ) timer (default 240)
- Bounded (มีขอบเขต) Update : EIGRP → update แต่ที่เปลี่ยน
- Triggered Update → update เมื่อผ่าน periodic time
- Random Jitter → ถ้ามี nw ที่เป็น multiple access router หมายความว่ามี共用ต่อ กัน → if เกิด update ซ้ำกันจะยังไงนี่? จึงต้อง random



▷ ปัญหา standard DV. Routing Loops กรณี intf ที่ต้อง down จะถูกอัปเดตลงใน table → กรณี neighbor จะกลับมา update ค่า update → hop เพิ่ม↑∞

| | RIP v.1 | RIP v.2 | IGRP | EIGRP | แก้ไข |
|------------------------------|---------|---------|--------|---------|-------|
| speed convergence | slow | slow | slow | fast | |
| scalability-size nw | small | small | small | large | |
| use of VLSM | x | ✓ | x | ✓ | |
| resource usage | low | low | low | medium | |
| implementation & maintenance | simple | simple | simple | complex | |

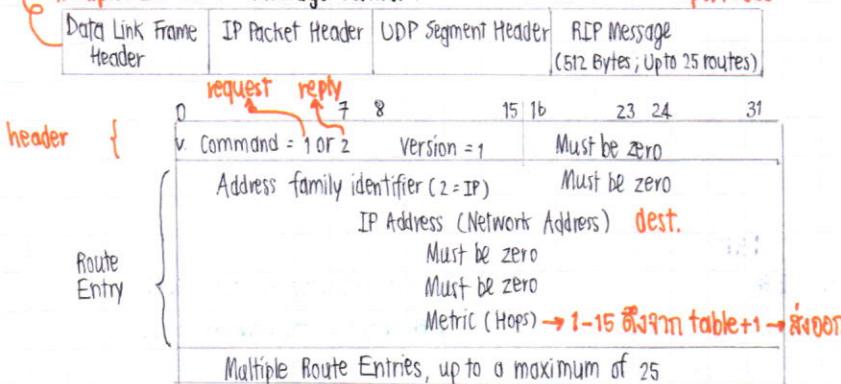
- ① set max hop = 15 → if hop = 16 → unreachable (_intf down ไม่รู้)
- ② holddown timer (เมื่อ intf down → hold)
- ③ split horizon rule → กรณีที่ update นั้นไป intf ที่ไม่รู้ update ไม่
- ④ Route Positioning → ① เมื่อ intf down set unreachable ② สำหรับ unreachable ที่มี positioning อยู่ปัจจุบัน inf นั้น hop = 16
- ⑤ ③ with ④ → กรณี unreachable จะ over rule split horizon กรณี ip intf ที่ down (hop=16)
- ⑥ IP & TTL (Time to Live) เกิดภัย update ไม่สามารถ TTL = 0

▷ RIP version 1 (AD = 120)

คุณสมบัติ : • classful , DV • metric = hop count • hop count > 15 unreachable • update broadcast ทุก 30 s

encapsulated (ใช้ส่วน UDP segment ของ source & dest.)

Transport L. RIPv1 Message Format



msg 有 2 types

① Request → รับ routing table
→ ให้ intf ที่ config ให้เพื่อ update

② Response → ยื่น info. ให้ routing table

• ip addr. universal class A, B, C

• Basic RIPv1 Config

① กำหนด basic config

② กำหนด router rip R1(config)# router rip

+ กำหนด nw R1(config-router)# network (nw ip ที่ต้องรู้) -----

▷ Verification & troubleshooting (เช็ค.รีบูต)

• passive interface command (ในการ update intf ที่ไม่ต้องการ) R(config-router)# passive-interface intf-type (Fa 1/0/1) intf-number (0/0/0/0/0)

▷ Automatic Summarization : RIP Route summarizes classful nw → เพิ่ม size routing table

→ ข้อดี : • เพิ่ม size routing update • single router ไม่ต้องดูทุก route ใน multiple route ที่มีใน routing table

→ ข้อเสีย : • ไม่ support discontiguous nw (major nw เดียว กับ minor nw ต่างๆ) → อาจจะเกิด local balancing มาก

• boundary Routers : summarize RIP subnet from 1 major nw to another

▷ Processing RIP Update

กรณีที่ update ที่ intf ทำให้ classful ใดๆ ก็ได้ ?

→ Y : update subnet nw ที่เป็น 172.16.1.0

→ N : update classful ที่เป็น 172.16.0.0

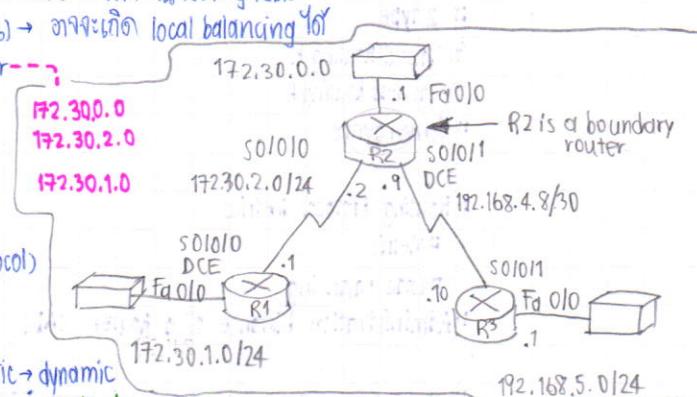
▷ Default route & RIP v1

กำหนด default route

R(config)# ip route 0.0.0.0 0.0.0.0 S0/0/1

default info. originate command → ในการ update ที่มี rip กำลังจะ : static → dynamic

Router จะจัดการ 2 protocol ← R(config-router)# default information originate



Chapter 5 RIP version 2 & Access Control Lists (ACL)

| RIP v1 | vs | RIP v2 |
|--|----|--|
| classful (มีต้อง subnet mask, ไม่ support CIDR) | | classless (update subnet mask, support Variable Length Subnet) |
| not support discontiguous subnet | | support route summarization (Prefix Aggregation) |
| not support VLSM b/c ไม่มี subnet mask (255,255,255,255) | | update next hop addr. |
| routing update → broadcast | | for authentication routing (จัดการ discontiguous ทาง物理) |
| 9s timer ป้องกัน routing loop | | Routing update → multicast |
| 9s split horizon or split horizon with poison reverse | | |
| 9s triggered update | | |
| max hop count = 15 | | |

▷ ข้อจำกัดของ RIPv1

ไม่มี virtual interface
can ไม่ต้อง routing ที่ต้องการ
โดยทั่วไป

- loopback intf → ping ที่ → ip virtual intf → reply ที่
- Null intf → ใช้กับที่ต้องการ channel ที่ต้องไม่ต้องการ → วิ่งบน null intf → packet discard แล้ว → timeout
- Static route & null intf → null intf กรณีที่ไม่สามารถ reach static route

R(config)# ip route summary-static-route subnet-mask Null 0
(major-nw) → รอง static supernet route

* ความต่าง Link-state

| OSPF | IS-IS |
|---------|---------|
| Fast | Fast |
| Large | Large |
| Yes | Yes |
| High | High |
| Complex | Complex |

for Staples

- Route redistribution (รีดิสทริบูชัน)** → วนลูป rip ยัง static ต้องเพิ่มให้ (Taru rip แล้ว static ไม่ได้) : Router(config-router)# redistribute static
- Verify & Test Connectivity** : show ip interface brief , ping (WA: 1=ถูก, U=ไม่ถูก, . = timeout), trace route
- RIPv1** : classful, ไม่สนใจ subnet mask, summarize nw @ major nw boundaries, if nw เป็น discontiguous & RIPv1 config convergence จะไม่ถูก
- ตรวจสอบ routing table debug ip rip** (content of routing update, ที่มีใน RIPv1 แต่ไม่ใช่ subnet mask จะมี nw addr.

▷ RIPv2

▫ config

- Enabling & verify configuration RIPv2
- Config RIP → RIP v1 → can แบ่งออกเป็น V1 & V2 หากแบ่งออกเป็น V1
 - RIP v2 → can แบ่งออกเป็น V1 & V2
- Auto summary & RIPv2 → auto sum route @ major nw boundaries
 - sum route ต้อง subnet mask ที่มีอยู่ใน classful subnet mask (Multiple Route Entries, upto a max of 25)
- disabling Auto-summary : no auto-summary bec when ที่ nw topology ต้องเป็น discontiguous

▫ VLSM & CIDR

- verify info. ที่ sent by RIPv2 debug ip rip
- VLSM → ตรวจสอบ nw addr. & subnet mask
- CIDR → ใช้ superneting (= bunch of contiguous classful nw ที่มีใน addr. ไม่ใช่ single nw.)
- verify show ip route , debug ip rip

▷ Access Control List = กฎของเครือข่าย → ตรวจสอบ → check → source → dest อะไรเป็น?

→ ถ้าเป็น conversation ต้องเป็น (เช่น FTP) อะไรเป็น?

▫ Packet Filtering ① dest, source @ L2 ② protocol ที่มา ③ ไป nw ไหน, ทำ甚子 → ทำ甚子นี้กับพื้นที่ Block ยัง

▫ Operation → ทำงานเป็น sequence statement

→ last statement เป็น implicit deny → block → discard

▫ Standard IPv4 ACLs

vs

▫ Extended IPv4 ACLs

- check source addr.
- ให้ permit or denies ทั้งหมด protocol

access-list 10 permit 192.168.30.0 0.0.0.255

- number ACL : 1-99 & 1300 - 1999

- check source & dest. addr.

- ให้ permit or denies specific protocol

access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80

- Number ACL 100-199 & 2000-2699

▫ wildcard → invert ของ subnet mask

→ 0 = match/fix , 1 = ignore/0:ถ้าเป็น 1

→ วิธีการ set ของ ip ① หา bit ที่ต้องการ ให้ wildcard mask ตามนั้น = 0
(match range) ② bit ที่เหลือ = 1

if กรณีต้องการ คุณภาพ Pattern or/and ทั้งคู่ ต้องห้าม wildcard จะเน้นอ่อนกัน

→ ของ wildcard ของ subnet = 255.255.255.255 - subnet mask

→ key word → 0.0.0.0 = match all 98 host

→ 255.255.255.255 = ignore all 98 any

R1(config)# access-list 1 permit 192.168.10.10 0.0.0.0

R1(config)# access-list 1 permit host 192.168.10.10

R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255

R1(config)# access-list 1 permit any

▫ Guideline for (3Ps) → ONE ACL / protocol = ctrl traffic flow บน intf, ACL ต้อง define ให้ Mac protocol enable on intf

ACL creation

→ One ACL/direction = ctrl traffic in 1 direction at time on an intf, แปลง ACL ctrl in & out bound traffic

→ One ACL/interface = ACL ctrl traffic for an intf, Ex. G 0/0

▪ where

→ Extend ACL : @ close source → standard ACL : @ close destination

▫ Config ACLs

▪ Standard number

เลือก 1 แบบ
ตัวส่วนใหญ่จะเลือกน้ำ
ที่จะ can ให้ต่อตัว
(permit, deny, remark)
ได้เลย

▪ in intf

Router(config)# access-list access-list-number num#
deny | permit | remark = comment
source [source-wildcard] [log]

access-list num# permit any,

Router(config)# ip access-group name
{access-list-number} {access-list-name}
{in | out}

▪ out

Router(config)# ip access-list [standard | extended] name

▫ verify : show ip interface , show access-list

▫ Securing VTY port → อย่างที่คนนอกเข้ามา permit ทางที่เราต้องการให้เข้า

Router(config-line)# access-class access-list-number {in [vrf-also] | out }

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

standard : 0-99,1300-1999 Extended : 100-199,2000-2699

access-list access-list-number { deny | permit | remark }

protocol source [source-wildcard] [operator operand]

[port port-number or name] destination [destination-wildcard]

[operator operand] [port port-number or name][established]

ใช้คลื่น same standard

เมื่อถึงได้ก็แบบ number & name

- debug - output : debug ip packet ACL-number

ip,tcp,udp

http://paperkit.net

ไม่เสียตัง



Chapter 6 OSPF & DHCP

▷ Link-state Routing Protocol = เป็น protocol ที่ต้องรู้ complete map ของ nw topology รันก์ตาม rule shortest path first (SPF)

ข้อดี : ① large nw ② fast convergence ③ admin ควบคุมได้

ข้อเสีย update ① learn info. ของ link ② say hello neighbor ③ 信息 info. ของ Link-state Packet (LSP)

④ router flood LSP to all neighbors + ต้องมี db ⑤ router ต้องรู้ db (minimum tree) + Adding OSPF to routing table

ข้อดี ① รู้ topology map ค้นใน shortest path, ② fast convergence รู้最快 route ③ LSP sent only when change topology (การเปลี่ยนแปลงที่สำคัญ) ที่เปลี่ยนแปลง → คำนวณ shortest path ④ hierarchical design (NN ตั้งแต่ที่) → ลด resource req. ไม่ต้องสืบสานทั่วทุก area

ข้อเสีย ① memory fun. ที่ต้อง all link-state route ② ต้อง CPU ใช้หนัก คำนวณ ③ คำนวณ LSP ต้อง 98% SW งาน

▷ OSPF AD < 110

↳ stable : ① Neighbor show ip ospf neighbor ② Topology (ที่ map) show ip ospf database ③ Routing (มี shortest path)

message → Encapsulating : MAC Dest. = Multicast : 01-00-SE-00-00-05 or 01-00-SE-00-00-06

Protocol Field = 89

- Type OSPF Packet : 01 Hello → ประมาณ 10 s (default: multi access & point to point nw), ประมาณ 30 s (default: non-broadcast multi access (NBMA) nw)
- : 02 DB Description (DBD) → synchronization db info
- : 03 Link-state Request (LSR) → request link-state
- : 04 Link-state Update (LSU) → send update link-state
- : 05 Link-state Acknowledgment (LSAck) → ยืนยันการรับ

operation : ผ่านๆ แล้ว ① Down state (เริ่มต้น) → ② Init state (เริ่มต้น hello) → ③ Two-Way State (มี hello) → Ex State state

→ Exchange state → Loading state → Full state (ทั้งหมด router update ข้อมูลเรียบร้อย)

config Single-Area OSPFv2 router ospf process-id → 1-65,535, เป็น locally significant

R(config-router)# router-id 1.1.1.1 → ต้อง set can 98 loopback, active interface ip ถูกต้อง but กรณี ① หาย ② ต้อง config

• router ospf process-id
network network-address wildcard-mask area-id

OSPF cost → 98 SW ที่ต้องคำนึง [default reference BW = 10^8] [ที่ต้องคำนึง physical]

$$\text{cost} = \frac{10^8 \text{ bps}}{\text{intf BW bps}} \begin{cases} \rightarrow 10 \text{ Gb Ethernet} = 100 \times 10^8 \rightarrow \text{cost} = 1 \\ \rightarrow 1 \text{ Gb Ethernet} = 10 \times 10^8 \rightarrow \text{cost} = 1 \\ \rightarrow \text{Fast Ethernet} = 10^8 \rightarrow \text{cost} = 1 \\ \rightarrow \text{Serial Ethernet} = 1.544 \times 10^6 \rightarrow \text{cost} = 64 \end{cases}$$

→ เมื่อ input BW = R(config-if)# bandwidth 64 (EIGRP & OSPF config ต้อง)

→ เมื่อ input cost R(config-if)# ip ospf cost 15625

→ ต้องปรับปรุง cost → ปรับปรุง ref BW

การคำนวณ cost ต้องคำนึง cost ต่อตัว ไม่ใช่ sum cost นั้น auto-cost reference-bandwidth bandwidth...mbps

- Fast Ethernet F=1, G=10F=1 auto-cost reference-bandwidth 100 (หน่วย 10^8)

- Gigabit Ethernet F=10, G=10F=1 auto-cost reference-bandwidth 1000 (หน่วย 10^8)

- 10 Gigabit Ethernet F=100, G=10, 10F=1 auto-cost reference-bandwidth 10000 (หน่วย 10^8)

verify ospf show ip ospf neighbor, show ip protocol, 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

R(config)# router ospf process-id

R(config-router)# default-information originate

↳ เรียกว่า redistribute default route update ที่ OSPF

▷ DHCP (Dynamic Host Configuration Protocol) → ช่วย config ให้ host 98 auto (ให้ ip, subnet mask, default gateway, DNS)

method ① Manual Allocation : admin assign IP

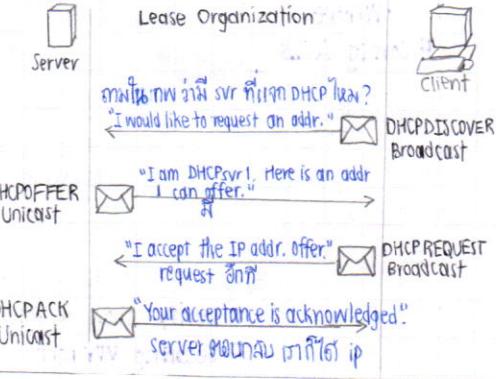
② Automatic Allocation : DHCPv4 auto assign addr. บน pool & ต้อง lease (เวลา) time

③ Dynamic Allocation : 98 ต้องรู้ ip ที่จะให้ ip ให้ lease time → กำหนด lease time ก็จะ re-ip ใหม่

config

R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9 ip 98
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1 98 pool
R1(dhcp-config)# network 192.168.10.2 255.255.255.0 nw ip 98
R1(dhcp-config)# default-router 192.168.10.1 * 98
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com } 98/98
R1(dhcp-config)# end

operation [DHCPv4]



To disable dhcp - no service dhcp

ใน ip ที่ต้อง → cmd → ip config /release → ip config /renew
→ รีเซ็ตตัวเอง (@ จะลบ set ip)

verify

show running config | section dhcp

show ip dhcp binding

show ip dhcp server statistics

On the PC issue the ipconfig /all command

debug same Extended

config DHCP client (กัน ip 98 client) : -if) # ip address dhcp

-if) # no shutdown

Chapter 7 Basic Switch Address Resolution Protocol

▷ LAN Design → Borderless sw nw design : มีลักษณะเป็น :: - Hierarchical, - Modularity, - Resiliency, - Flexibility

- 2 ลักษณะ : ① 3-Tier LAN Design :: 1) Core 2) Distribution 3) Access ② 2-Tier LAN Design :: 1) Collapsed core/Distribution 2) Access
- ① Core → จุดที่เชื่อมต่อ SW ทุกๆ จุด → ทำให้ speed ↑ ของ nw ที่ต้องการ { Layer 3 Support, [Gig/ 10 Gig Ethernet] }
- ② Distribution → เผื่องผ่าน { 1 & 2 }, Security Policy / Access Ctrl } Redundant Components → มีลักษณะเป็นป้อม { Link aggregation }
- ③ Access → ต่อไปยัง End Device, Port Security, VLAN [PoE], Power over Ethernet { Quality of Service (QoS) }

▫ เพิ่มค่าความจุของ LAN BW & ปรับปรุง max

- f & n ของ Server ① Enterprise S. (ตรวจสอบค่า) → ต่อตัว @ MDF (Main Distribution Facility : Core) → ไม่เพียงแค่หน้าบอร์ด ก็ต้องกับบันได
- ② Workshop S. (ตรวจสอบค่า) → ต่อตัว @ IDF (Intermediate D. F. : Distribution) → ต้องมีช่องทาง cross กัน access ก็ต้องกับบันได

- Collision detection issue (ดูจาก ตรวจสอบ, ตัวอย่าง)
- Segmentation issue (ดูจาก ตรวจสอบ) ⇒ ทำให้ max ของ nw ที่ expand ได้ยาก
- Broadcast domain issue → ทุก nw (ต้องตรวจสอบ broadcast MAC Addr.); Broadcast nw ของอื่นมาแล้ว

VCC (Vertical cross-connect) : optical fibre ⇒ MDF ↔ IDF

HCC (Horizontal cross-connect) : UTP ⇒ Distribution ↔ Access

- Segmentation เป็น process split single collision domain → smaller collision domain ลดลง collision บน LAN segment : L#2 device ทุกๆ 16 ports
- Broadcast domain เป็น port แต่ router (L#3) เป็นตัว filter/segment broadcast ที่ต้องตรวจสอบจากตัวเอง

▷ SW Environment

- SW Operation
 - ① Learning : รับ frame ที่ SW จะรู้สึก Source MAC Addr. ที่อยู่บน port ที่แล้ว + reset Aging
 - ② Aging : ตรวจสอบ MAC Addr. → if ถูกต้อง → หัก → ลบออก dest. ของ queue table
 - ③ Flooding : ถ้า frame อยู่บน port ใดๆ ของ SW ที่ไม่รับ frame (เป็น 1) broadcast, 2) multicast, 3) unknown unicast
 - ④ Forwarding : หัก dest. ของ queue table
 - ⑤ Filtering : if หัก frame ที่ dest. บน port ที่เป็นช่วง dest. (source & dest. ใช้同一 interface) แล้วทันที filter หัก

- SW Methods
 - ① Store & Forward Sw ⇒ check CRC กรณี error แล้ว → ลาก่อน → หักแล้ว, auto buffer
 - ② Cut-through Sw ⇒ check ขนาดของ frame (dest, source อยู่ในตัวเดียวกัน 12 byte first) [10 ms], no FCS & auto buffer
 - ③ Cut-through mode :: ① fast-forward ~12 byte ② Fragment-free ~64 byte :: < 64 = ทุกๆ 1 byte → ไม่ต้อง → ไม่ต้อง

- SW Domains :: ① Collision Domains ⇒ domain ที่ต้องไม่ซ้อนกัน “ถ้า SW ไม่รับเข้าไป”
- ② Broadcast ~ ⇒ domain ที่รับ broadcast → ทุก domain ต้องหักตัวเอง “ถ้า router ไม่รับเข้าไป”

▷ Basic SW Concept & Configuration

- Basic SW Config
 - ① SW Boot Sequence = same router same \$config-if# ip address ip subnet router | s(config-if)# no shutdown
 - ② Verify Port Config

show int fa0/0, /start-up-config
/running-config, /flash, /version
/history, / ip fa0/0, / mac-address-table

▪ Preparing of Basic SW Management : sw ต้องเป็น SVI (switch Virtual Interface)

▪ Config SW Port → Duplex Communication : ① Full ② Half (sw ต้องต่อสายต่อไปน้ำหนึ่งต่อหนึ่ง)

▪ ตั้งค่า intf → s(config-if)# duplex full → s(config-if)# speed 100 (เปลี่ยน speed)

▪ Auto-MDIX : กรณี sw จะต้องต่อสาย cross-over หรือ直通 ไม่รู้จะเป็นต่อแบบไหน

▪ ตั้งค่า intf → s(config-if)# duplex auto → s(config-if)# speed auto → s(config-if)# mdix auto

- SW Security : Security Remote Access → SSH (Secure Shell) TCP port 22, telnet : TCP port 23
- Config : S-CC(config)# ip domain-name ๑๐ → # crypto key generate rsa → # username admin pass ๑๒๓ → line vty 0 15 → -line# transport input ssh → -line# login local [Verify SSH: show ip ssh, show ssh]

- SW Port Security → กำหนด policy ของ MAC Addr. ให้เข้า/ออกได้ - ไม่ได้

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

secure MAC Addr. : → ① Static :: S(config-if)# switchport port-security mac-address MAC ADDR

 ② Dynamic :: S(config-if)# switchport port-security mac-address Sticky → learn frame ที่ร่วมกันแล้ว → record ไว้

กรณีใหม่ๆ MAC : # switchport port-security maximum MAX

Violation mode : ① protect : security violation protect mode

 ② restrict : security violation restrict mode ⇒ ถ้าไปเข้ามาแล้วจะเกิดขึ้นปั๊บ -----

 ③ shutdown : security violation shutdown mode ⇒ default

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

▫ Addr. Resolution Protocol (ARP)

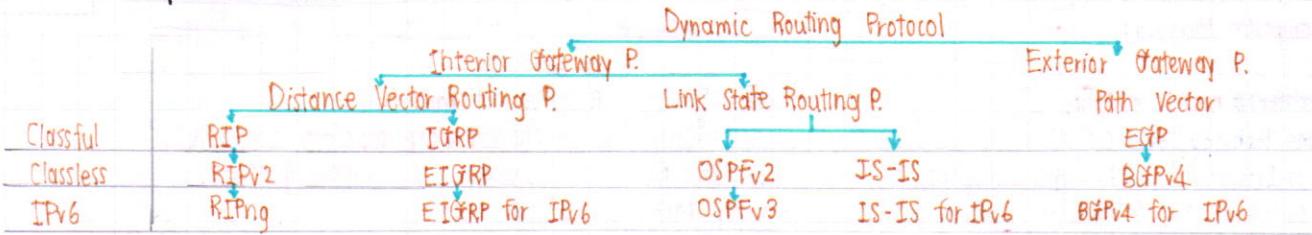
IPv4: classless [int P.1-2]: Variable Length Subnet Masking (vLSM) : ไม่จำกัด nw ที่ต้องการ ⇒ เสื้อผ้าของคนเดียว → เล็ก - Fixed ~ : ไม่จำกัด nw ที่ต้องการ

Security violation Modes

| Violation Mode | Forward Traffic | Send Syslog Message | Displays Error Msg | Increase Violation Counter | Shut Down Port |
|----------------|-----------------|---------------------|--------------------|----------------------------|----------------|
| Protect | No | No | No | No | No |
| Restrict | No | Yes | No | Yes | No |
| Shutdown | No | No | No | Yes | Yes. |



Chapter 11 : EIGRP



▷ EIGRP (Enhanced IGRP)

□ Characteristics (ลักษณะ-ประจักษ์)

▪ Basic Features △ Cisco-proprietary (พึงงานเฉพาะ) protocol von cisco ปี 1994

⊕ ร่าง classless version of IGRP ⊕ รองรับทั้ง IPv4 และ IPv6 protocol, รองรับทุกๆ ที่ส่งข้อมูล cisco router เป็นเพียง

▼ DUAL (Diffusing Update Algorithm) = ทำให้ loop-free & back up path ที่ดีของ routing domain → in best path

→ ทำให้ routing ทำงาน very fast convergent (convergent time < no OSPF) รวมถึง backup path
(n. ที่ต้อง ≈ OSPF บน router)

▼ Establishing Neighbor

= สองฝ่ายที่ติดต่อ direct connected EIGRP routers

Adjacencies

= Adjacencies are used to track the status of these neighbors

▼ Reliable Transport Protocol

= RIP provides delivery of EIGRP packets to neighbors.

▼ Partial and Bounded

= update เท่าเดิมที่ต้องการ เช่น ไม่ต้อง update ไปที่ต้องไม่ต้องไป

▼ Equal and Unequal Cost

= กรณี admin กำหนด cost ให้ต่างๆ กัน ระบบจะคำนึงถึง cost ที่ต่ำที่สุด

Load Balancing

△ All protocol-dependent modules (PDMs) หรือ router protocol ที่สามารถรองรับ IPv4, IPv6, legacy protocol IPX and AppleTalk
△ PDMs คืออะไร :

⊖ maintain EIGRP and topology table (Neighbor Table → รู้ว่า Topology Table → สร้าง routing table ใหม่ routing

⊖ คำนวณ metric ด้วย DUAL ⊖ คำนวณ DUAL สร้าง routing table

⊖ implement filtering and access lists ⊖ ในการ redistribution with other routing protocol

△ RTP is EIGRP Transport layer protocol ที่รับ Delivery & reception von EIGRP packets

= ใหม่ msg ที่อยู่ application layer ที่ maintain โหนด, msg ใหม่ ที่ส่ง von EIGRP

△ ใหม่เท่านั้น von RIP Packet แต่ต้องมีตัวต่อ (msg ≈ OSPF)

⊖ Reliable packet require explicit (ต้อง) ack ที่ต้อง dest. ⊖ Update, Query, Reply

⊖ Unreliable packet do not require ack ที่ต้อง dest ⊖ Hello, ACK (authen ≈ RIPv2, OSPF)

△ ต้อง authentication (no encrypt routing update) แทน : is recommend (ดูต่อไป)

buy IGRP multicast 224.0.0.9

RIPV1 broadcast 255.255.255.255

▫ Packet Type routing update or queries EIGRP multicast IPv4 : 224.0.0.10, IPv6 : FF02::A

① Hello → ติดต่อ adjacency ระหว่าง router 2 ตัวที่ติด neighbor ที่ต้อง response, แต่ unreliable

② Update → update info. ที่ต้อง dest. update info von routing ที่ต้อง neighbor router

③ Acknowledgement → ต้อง dest. update ที่ต้อง ACK

④ Query → request info. routing ที่ต้อง neighbor router } ต้อง dest. info. von routing ที่ต้อง query ที่ต้อง router ที่ต้อง

⑤ Reply → ต้อง dest. query ที่ต้อง reply } ต้อง dest. reply ต้อง dest. query ที่ต้อง reply

□ Implement EIGRP for IPv4

△ Autonomous System (AS) is a collection of network under von single Authority (ดูใน RFC 820)

↳ AS number → ใช้ exchange routes between AS

→ managed by IANA & Assigned by RIPE to ISPs, Internet Backbone providers, and institution ไม่ต้อง管

→ 16 bit : 0-65535 → since 2007, 32 bit : over 4 billion

△ Configure : R(config)# router eigrp AS-# (≈ router-id @ OSPF)

R(config-router)# eigrp router-id → ต้องตั้งต่อไปloop back intf. → IPv4 address ที่ต้อง active

R(config-router)# network nw number [wildcard-mask] ต้องตั้งต่อไป

R(config-router)# passive-interface type number [default] : ต้อง update ที่ต้อง interface ที่ต้อง (เช่น LAN, B, F) ↗ ต้องตั้งต่อไป serial

△ verify : show ip eigrp neighbor

show ip protocols

show ip route



□ Operation

- Initial Route Discovery (initial) ① R1 say hello to neighbor router ② R2 送出 hello or update 邻居表 ③ R1 now ack & update info.

④ If DUAL finds best route and update routing table

$$\text{Metrics} = \text{BW} [\text{lowest}], \text{Delay} [\text{best}], \text{Reliability} [\text{Worst}], \text{load} [\text{worst}] \text{ ค่า value: show interface}$$

Default Composite Formula:

$$\text{metric} = [k_1 \times \text{bw} + k_3 \times \text{delay}] \times 256 \\ = \left[\frac{10,000,000}{\text{bw}} + \frac{\text{sum of delay}}{10} \right] \times 256$$

$$\text{Complete} = \left[k_1 \times \text{bw} + (k_2 \times \text{bw}) + k_3 \times \text{delay} \right] \times \left[\frac{k_5}{(\text{cost} - \text{load})} \right]$$

- R(config-router)# metric weights tos k1 k2 k3 k4 k5 - set bw: N/A intf. → R(config-if)# bandwidth kilobits/bw-value

◦ DUAL and the Topology Table (FSM (Finite state Machine) ผู้ติดตาม) → show ip eigrp topology [all/link], show ip route

+ Successor (S) [router ที่สู่ dest. ดีสกูต] = neighbor router ที่สู่ dest. ดีสกูต min cost

+ Feasible Successor (FS) [ที่ดีกว่า & feasible condition] = Backup path (ดูมายด์)

+ Reported Distance (RD) [distance ที่ neighbor รายงาน report distance มาก] = "advertised distance" ทางเดินสู่ dest. ไม่ cost min

+ Feasible Distance (FD) [distance ที่ min] = min distance ที่เดินสู่ dest. now เป็น cost lowest → dest. NOVIA hop

▷ IPv6

▷ IPv4 Issue

- Need for IPv6 → ไม่มี range ip จำกัด (private ip, NAT), รองรับ IoT ได้
- ▷ ใช้共存 (coexistence)

- Migration IPv4 → IPv6 Techniques :: ① Dual stack = run ทั้ง 2 if ไม่-authenticated user

② Tunneling (IPv6 ไม่ core ไม่ support) = ปล่อยเฉพาะ route เผื่องผ่าน v4 ไม่ต้องต่อ

③ Translation (ผ่าน NAT) = IPv4 ↔ IPv4

▷ IPv6 Addressing : 128 bit แบ่ง 8 ช่วง [1 ช่วง 8 บิต = 2 byte = 16 bit] → represent base 16 ทุก 4 bit

▷ จุดเด่นของ IPv6 ที่สำคัญ

Rule 1 - Omit Leading 0s = หักมุม partition "0" ที่อยู่ต้นน้ำที่ต้องมี 000X , 00XX , 0XXX

Rule 2 - Omit All 0 Segment = กรณี segment ที่มี "0" ติดกันมากๆ ":" ตัวอักษรเดียว only

▷ Type of IPv6 Address

- IPv6 Addr. Type ① Unicast : ② Global Unicast ③ Link-local ④ Unique Local

static config

↳ ipv6 address ipv6-address/prefix-length → no shutdown

② Multicast

③ Anycast = สำหรับ device

△ IPv6 Prefix Length = 0-128, most LANs is /64 b/c. LAN เส้นสูง 64 bit

▷ IPv6 Routing

◦ Config Static Route

R(config)# ipv6 route ipv6-prefix/ prefix-length { ipv6-addr | exit-interface } → next hop addr. exit-interface

* ต้อง config ต่อไป routing ipv6 ipv6 unicast-routing

△ 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 → ฟูนัล IP v4

R(config-rtr)# no shutdown

△ network command: ไป intf → ipv6 eigrp AS-#

but. passive-interface ต้องมีทั้ง 2 global config same local

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

| IPv6 | vs | IPv4 |
|---------|----|---------|
| 128 bit | | 32 bit |
| base 16 | | base 10 |

Summary Config

► ROUTER

① Basic Configuration

เมื่อ intf (port Gig, Fa) \Rightarrow R(config-if)# ip address ip-addr subnet-mask \Rightarrow R(config-if)# no shutdown
 เมื่อ intf (serial หรือ DCE) \Rightarrow ~ \Rightarrow R(config-if)# clock rate 56000
 Verify : show running-config \rightarrow ดูว่า config อยู่ในปัจจุบัน
 show startup-config \rightarrow set
 show ip route \rightarrow ดู routing table
 show interface \rightarrow } show info ของ intf, เช่น
 show ip interface \rightarrow show info. ของ intf, เช่น
 show ip interface brief \rightarrow show intf. ที่มีการตั้งค่า ip?, status?

② Protocol

- static routing

เมื่อ intf \Rightarrow R(config-if)# ip route nw-ip subnet-mask { ip addr. | exit-intf }
 Default route \Rightarrow # ip route 0.0.0.0 0.0.0.0 { ip addr. | exit-intf }

next hop addr
↑
{ ip addr. | exit-intf }

- Dynamic routing

- Interior Gateway P.

- 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-number

from RIP \Rightarrow static: R(config)# router rip \Rightarrow R(config-router)# { redistribute static | default-information originate }

หมาย router ที่ไม่ต้อง set default route @ Intf. ที่ไม่ใช่ gw 2 protocol ด้วย!

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

verify : same RIP ที่ไม่ show ip interface brief

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

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

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

~ see bw: เมื่อ intf. \Rightarrow R(config-if)# bandwidth kbytes/bw-value

- Link state Routing P.

- OSPF

: R(config)# router ospf process-id \Rightarrow -router)# router-id 1.1.1.1 \Rightarrow network nw-ip wildcard-mask area area-id

set BW: เมื่อ intf \Rightarrow R(config-if)# bandwidth 64

set cost: ip ospf cost 15625

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

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

All: clear ip ospf process

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

R(config)# router ospf process-id

R(config-router)# default-information originate

redistribute (OSPF \Rightarrow bgp): R(config)# router ospf process-id \Rightarrow R(config-router)# redistribute ?

③ จัดการ

- ACL : If กรณี 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 : เมื่อ intf \Rightarrow R(config-if)# ip access-group { ACL-num | ACL-name } { in | out }

All: no access-list ACL num

verify : show ip interface S0/0/0, show access-lists

- securing VTP with standard IPv4 ACL : R(config-line)# access-class ACL-num { in [urf-also] | out }

- Extended IPv4 ACL:

R(config)# access-list ACL-num [deny|permit|remark] protocol source [source-wildcard][operator operand][port 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
 R(config)# ip dhcp excluded-address ip-addr
 R(config)# ip dhcp pool LAN-POOL-1
 R(dhcp-config)# network hw-ip subnet-mask
 R(dhcp-config)# default-router ip address-gateway
 verify : show running-config | section dhcp, show ip dhcp binding, show ip dhcp server statistics

► SWITCH

① Basic Configuration

- manage inf: s(config)# interface vlan N \Rightarrow s(config-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: int intf \Rightarrow s(config-if)# duplex full \Rightarrow speed 100
- auto-MDIx: int 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 s(config-line)# transport input ssh
 - Verify: show ip ssh, show ssh \Rightarrow login local
- + Telnet (TCP port 23)

- Switch Port Security: int intf \Rightarrow s(config-if)# switchport mode access \Rightarrow switchport port-security

+ static secure MAC Addr. \Rightarrow switchport port-security mac-address MAC-ADD

+ dynamic \Rightarrow switchport port-security mac-address sticky

+ max MAC Addr. \Rightarrow switchport port-security maximum MAX

+ violation Mode \Rightarrow switchport port-security violation { protect | restrict | shutdown } mode

+ Verify: show port-security int fa 0/3, show port-security address

③ STP: int 1 s(config)# spanning-tree VLAN 1 root { primary | secondary }

int 2 s(config)# spanning-tree VLAN 1 priority 24576 \Rightarrow < its priority >

verify show spanning-tree [active], show running-config

□ Rapid PVST+

+ Port Fast: int intf. \Rightarrow s(config-if)# spanning-tree port fast

+ BPDU Guard: int intf. \Rightarrow s(config-if)# spanning-tree bpduguard mode

+ config: s(config)# spanning-tree mode rapid-pvst \Rightarrow int intf. \Rightarrow s(config-if)# spanning-tree

+ clear STP \Rightarrow s# clear spanning-tree detected-protocol link-type point-to-point

④ VLAN verify: show vlan name, show vlan summary, show int vlan num, show int f0/0 switchport, show vlan [brief]

1) set VTP mode: s(config)# vtp version 2 \Rightarrow vtp mode { server | client | transport } \Rightarrow vtp domain name \Rightarrow vtp password pass

2) set trunk: int intf. \Rightarrow s(config-if)# switchport mode trunk 3) assign VLAN to server: s(config)# vlan num \Rightarrow name name

4) assign intf: int 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 vlan 10 \Rightarrow encapsulation dot1q 10

\Rightarrow ip address ip subnet

⑤ NAT: verify: show ip nat translations [verbose], show ip nat statistics

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

□ PAT = same dynamic mapping "overload" \Rightarrow ip nat