

for Staples

Chapter 1 Network Overview

- Network diagrams = แบบจำลอง NW ที่ใช้ในการplanning
- Network protocol → TCP/UDP, FTP, ARP, SMTP, POP3, IMAP, ICMP
→ (internet control message ping คำสั่ง ipconfig cmd)
- Network address → IP address (Logical Address) → MAC address (Physical Address)

NW ADDRS. ① IP addrs. (Logical Addrs.) ② Mac addrs. (physical Addrs.) ③ Port number (Service addrs.)

- Component of Network → NW Device of 3 type ① end devices = ผู้ใช้งานทั่วไป

② intermediary devices = อุปกรณ์ที่ต้องมี NW access devices, Internetworking devices, Security devices

hub switch router

2.1 hub, repeater ② จัดการชนกัน collision : จัด CSMA/CD จัดการชนกัน collision ของ LAN

2.2 switch, bridges ② Learning / Flooding / Filtering / Forwarding / Aging

2.3 Routers ② Routing

③ network media = วิธีการ เช่น copper, fibre optic, Wireless LAN — WAN

- Types of Network → SW ① ขนาดเล็ก ② ขนาดใหญ่

Reliable Network

① Fault Tolerance → ความต้านทานต่อข้อผิดพลาด

② Scalability → ความสามารถในการขยายตัวตามจำนวนผู้ใช้งาน

③ Security ความปลอดภัย

- Quality of Service คือ Service Quality

- Type of Connection in LAN ที่ต่อ LAN (VTP cat5) 1. star 100 Mbps 2. mesh 100m.

2 types ① star ② cross อยู่ในเครือข่ายเดียวกัน same switch SW -- hub, PC -- router

WAN Connection

→ auto DCE (female) and clock rate 56000

DTE (male) → RJ45 to DB9

→ serial console (Rollover Cable) → router — PC

→ manage config mode

- Logical Address : IP Address (IPv4) - 5 class A, B, C, D, E

reserved

Chapter 2 Basic Router Configuration

- Port number หมายเลข IANA 25 SMTP 23 WWW 80 HTTP 8080 name server

0-1023 = requesting entities "well known ports" ของ port

1024-49151 = registered port = published หมายเลข port

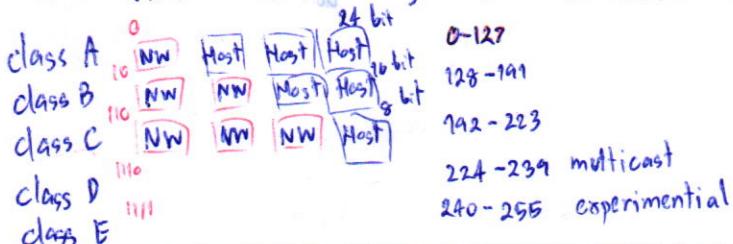
49152-65535 = dynamic/private port "Randomly generated"

IP address ของ host ที่ต้องมี max 255 workstation required

Ser. port

host ที่ต้องมี LAN interface

→ private addressing → ip reuse ได้



→ Physical Address = Mac Addr - Ethernet 48 bits 24 bit 12 bytes 16 bits 0x 00-00-00-00-00-00

Message Delivery

Unicast จัดการโดยผู้ใช้งาน NW ผู้ใช้งาน

Broadcast จัดการโดยผู้ให้บริการ DHCP, ARP ของ NW ผู้ให้บริการ

Multicast จัดการโดยผู้ให้บริการ ผู้ให้บริการ service

01-00-0C

Class A	Internal Addr Range	CIDR Prefix
10.0.0.0 - 10.255.255.255	10.0.0.0/8	
172.16.0.0 - 172.16.255.255	172.16.0.0/12	
192.168.0.0 - 192.168.255.255	192.168.0.0/16	

→ IEEE 3 byte code "Organizationally Unique Identifier" OUI

→ Mac Addr หมายเลข NIC ② หมายเลข MAC ที่ SAMe OUI

→ Ethernet device ที่มี OUI หมายเลข unique 3 byte ที่ต้องมี

→ 96 OUI 3 byte ที่ต้องมี

Cisco IOS (Internet network operation system)

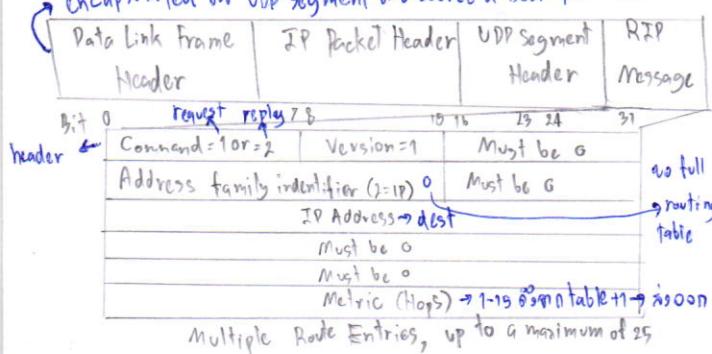
- Function ① Addressing ② Interface ③ Routing ④ Managing Resource ⑤ Security ⑥ QoS
- Router & SW Boot sequence ① POST ② Run boot loader ③ Boot loader does low-level CPU initialization ④ Boot loader initializes the flash file system ⑤ Boot loader locates & loads a default IOS to run in RAM

- Bounded (converge) Update : EIGRP update จำกัดพื้นที่
 - Triggered Update : update baseless periodic time
 - Random timer : กำหนดเวลาการรับข้อมูลจาก multiple access router ตามเวลาสุ่มของตัวเอง → ทำให้ update ไม่ต่อเนื่อง และ random
- ▷ ปัญหา Standard DV 1.) Routing Loops กรณีเกิดเหตุการณ์ bundown ของรุ่นต่อรองใน table → ผู้ที่ไม่ใช้ neighbor จะรับ update (new update → hop ↑ → infinity)
- | | RIP v1 | RIP v2 | IGRP | EIGRP |
|------------------------------|--------|--------|--------|---------|
| 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 |
- set max hop = 15 → ถ้า hop = 16 → unreachable (down 255.255.255.255)
 - hold down timer (fin intf down → hold)
 - split horizon rule → ไม่ลากซึ่งกัน update ระหว่าง intf ที่ต้อง update กัน
 - Route Positioning ① กรณี down set unreachable ② กรณี unreachable นั้น position ตัวเอง
 - with ④ กรณี unreachable นั้น over rule split horizon กรณี ip intf ไม่ down (hop = 16)
 - IP & TTL (Time to Live) เก็บ msmt update (เมื่อ TTL=0)

▷ RIP Version 1 AD=120

- กรณีที่เป็น classful, DV = metric = hop count > 15 unreachable = update broadcast ทุก 30s

encapsulated บน UDP segment ที่ source & Dest port = 520



- Automatic Summarization RIP Auto Summarizes classful nw → ทำให้ size routing table update, single router สามารถ sum multiple route ลงใน 1 routing table
- กรณีที่ support discontiguous nw (major nw ต้องติดต่อกัน) จึงต้อง load balancing ทั้ง boundary Routers : summarize RIP subnet from major nw to another และ update subnet nw ที่ ip 172.16.1.0
- Processing RIP update กรณีที่ต้อง update ทุก interface ที่เป็น classful แต่ต้องลงใน ip protocol, debug ip rip → passive intf command ทำให้ update intf ไม่ทำงาน
- default route & RIP V1 ไม่สามารถ mention ใน routing table (convergence protocol) → ต้อง mention default route

R(config-router) # ip route 0.0.0.0 0.0.0.0 50/0/1

default into . originate command ทำให้ update ทุก intf เป็น static → dynamic

router rip ระบุตัวเอง 2 protocol R(config-router) # default-information originate

Chapter 5 RIP version 2 & Access Control Lists

RIP V1

- classful (fix subnet mask, ไม่ support CIDR)
- not support discontiguous subnet
- not support VLSM
- routing update → broadcast

RIP V2

- classless (update subnet mask, support variable Length Subnet Masking (VLSM))
- support route summarization (prefix Aggregation)
- update next hop addr
- NE authentication routing (สามารถ discontiguous network)
- Routing update → multicast

ไม่ timer จึงไม่ routing loop

ไม่ split horizon or split horizon with poison reverse

ไม่ triggered update

max hop count = 15

การตั้งค่า RIP V1

- ใช้ virtual interface
- configuring routing ที่ update
- loopback

• loopback intf ping 255.255.255.255 → ip virtual intf → reply 255

• Null intf บีบีดูน้ำหนึ่งช่องทางที่ต้องการ → ไม่เก็บ null intf → packet discard หลัง timeout

• Static route & null intf null intf คือต้องไม่เก็บ static route

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

(major-nw) → ไม่ static supernet route

for Staples

- Route redistribution อย่างไรที่ rip จึง static หรือ redistribute ให้เป็น static ใช้ `R(config-router) # redistribute static`
- Verify & Test connectivity show ip interface brief, ping (wait 1 วินาที, timeout), traceroute
- RIPV1 classful ไม่ระบุ subnet mask, summarize network @ major network boundaries, if network บานปลาย discontiguous & RIPV1 config convergence
- more routing table debug ip rip (content of routing update) ตรวจสอบ RIPV1 ถูกต้อง subnet mask ของ各 network address.

▷ RIP v2 • Config

- Enabling & verify RIPV2
- Config RIP → RIPV1 → can route V1 & V2 ไม่สามารถ V1 → RIPV2 → can route & ไม่สามารถ V2
- Auto-Summary & RIPV2 → auto sum route @ major nw boundaries
 - sum route ของ subnet mask ไม่ต่อเนื่อง
 - classful subnet mask
- disabling Auto-summary : no auto summary (ไม่ต่อเนื่อง topology ไม่ต่อเนื่อง discontiguous)

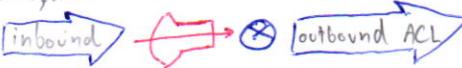
▷ VLSM & CIDR

- Verify info ที่ sent by RIPV2 debug ip rip - VLSM ตรวจสอบ nw addr. & subnet mask
- CIDR คือ supernetting (บันทุณย์ contiguous classful nw ให้เป็น addr ไม่ต่อเนื่อง single nw) - verify show ip route, debug ip rip

▷ Access Control List

- Packet filtering ① dest, source @ L2 ② protocol ③ nw layer, ports (การอนุญาตผ่าน block ทั้งหมด)

- Operation - รูปแบบ sequence statement

- last statement คือ implicit deny → block → discard 

• Standard IPv4 ACLS vs

- check source addr.
- who permits or denies specific protocol
- access-list 10 permit 192.168.30.0 0.0.0.255
- number ACL : 1-99 & 1300-2699

Extended IPv4 ACLS

- check source & destination addr
- who permits or denies specific protocol
- access-list 103 permit top 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 = 1 ทั้งหมด
- ให้เราสามารถ set ของ ip ① ให้เราสามารถ บิตที่ต้องการที่ wildcard mask ที่มี 0 = 0

(match range) ② bit ที่ต้องการที่ 1

ต้องมี ต้องมี 1 ที่ต้องการ pattern or/and ไม่ว่าจะเป็น wildcard อยู่ในหนึ่งเดียว

→ wildcard ของ subnet = 255.255.255.255. - subnet mask

→ keyword → 0.0.0.0 match all หรือ host

→ 255.255.255.255 ignore all หรือ any

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

`R1(config) # access-list 1 permit host 192.168.0.0`

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

`R1(config) # access-list 1 permit any`

• Guideline Por (3Ps) → one ACL/protocol = ctrl traffic flow on intf, ACL ต้อง define what protocol enable on intf

ACL creation - One ACL/direction = ctrl traffic in direction at time on an intf, ใช้ ACL ctrl in & out bound traffic

- One ACL/interface = ACL ctrl traffic for an intf, Ex Golo

where Extend ACL : ① close source → standard ACL : ② close destination

• Config ACLs

standard
number

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

ในการ remove all : `no access-list`

ถ้าไม่มี ① `no access-list num#` → ลบหาย

② `no vrrp#` → ลบหาย # หายไป

→ in intf Router (config) # ip access-group

(access-list-number | access-list-name)
 (in|out)

ในการ remove all `no ip access-group`

ถ้าไม่มี → ลบหาย # หายไป

↳ ลบหายก็ไม่ได้

- Verify show ip interface, show access-lists

- Securing VTY port ตรวจสอบ configuration permit ระหว่างที่ตั้งค่า password

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

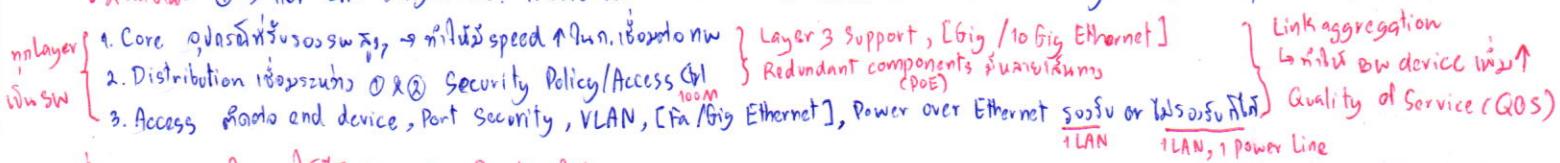
→ Extend : filter source /dest addr, protocol, port number



for Staples Chapter 7 Basic Switch Address Resolution Protocol

► LAN Design → Borderless sw new design: โครงสร้าง: Hierarchical, Modularity, Resiliency, Flexibility

• 2 ลักษณะ ① 3-Tier LAN Design ฯลฯ. 1. Core 2. Distribution 3. Access ② 2-Tier LAN Design ฯลฯ. 1. Collapsed Core / Distribution 2. Access



• คำแนะนำเพื่อเพิ่ม LAN BW & ลด Latency MAX

• LAN & WAN Server ① Enterprise S. (ขนาดใหญ่) ตั้งต่อ @ MDF (Core) ผ่านไฟเบอร์ออฟฟ์ที่ไม่ต้องผ่าน SW

② Workshop S. (ขนาดกลาง) ตั้งต่อ @ IDF (Distribution) ใช้สวิตช์ Cross หรือ access ที่ไม่ต้องผ่าน SW

• Collision detection issue (ปัญหานี้ตรวจสอบใน LAN) ทำให้เกิด collision ระหว่าง broadcast ของตัวเอง VCC Vertical cross-connect optical fibre MDF ↔ IDF

• Segmentation issue (ปัญหานี้ตรวจสอบใน LAN) ทำให้เกิด broadcast ของตัวเอง HCC Horizontal cross-connect UTP: Distribution ↔ Access

• Broadcast domain issue (ปัญหานี้ตรวจสอบใน LAN) broadcast ของตัวเอง ทำให้เกิด broadcast ของตัวเอง เนื่องจาก broadcast ไม่ได้ถูกตัดต่อ

• Segmentation ผ่าน process split single collision domain → smaller collision domain ลดลง collision บน LAN segment: LAN 2 device

• Broadcast domain ผ่านตัวเอง port บน router (LAN 3) ผ่านตัวเอง filter /segment broadcast ที่ต้องตรวจสอบต่อหากต้องผ่าน LAN bridges, SW

► SW Environment

• SW Operation 1. Learning รับ帧จาก SW คือ 1. รับ Source Mac Addr. บันทึก Port/Link + reset Aging

2. Aging ของ Mac Addr. บันทึก → ตัด

→ next dest block in table

3. Flooding นำ frame ของ port ของ SW ที่รับเมื่อมา 1.) broadcast 2.) multicast 3.) unknown unicast

กรณีที่ต้องการ { 4. Forwarding นำ dest (next block table)

5. Filtering นำ dest frame ที่ dest บน port ที่ต้องการ dest (source & dest ไม่ใช้同一 interface) เนื่อง filter ที่ต้องการ

• SW Methods ① Store & Forward SW → check CRC 如果有 error 丢弃 → ต้อง → ต้อง: ล่าช้า, auto buffer

② Cut-Through SW → check แต่เร็ว得多, No FCS & auto buffer

→ 2 mode : 1. fast-forward ~12 byte 2. Fragment-free ~64 byte < $14 = 84 - 70 = 14$ >

• SW Domains ① Collision Domains → domain ที่ไม่สามารถสื่อสารกันได้ เช่น 2 ตัวต่อ 2 ตัวต่อ " @ SW เป็นตัวต่อ "

② Broadcast Domains → domain ที่ broadcast ของ domain ต้องสื่อสารกันได้ " @ router เป็นตัวต่อ "

► Basic SW Concept & Configuration

• Basic SW Config . SW Boot Sequence

manage intf : s(config)# interface vlan num default gateway :

s(config-if)# ip address ip subnet s(config-if)# ip default-gateway ip

SW ที่ตั้ง Loopback ที่จะตั้งค่า Preparing of Basic SW Management router s(config-if)# no sh.

กรณี SVI → VLAN Contig SW port → Duplex communication ① Full ② Half (SW ที่ต้องการต่อไปยังตัวเรา)

ใน intf → s(config-if)# duplex full → s(config-if)# speed 100 (กำหนด speed)

→ Auto MDIX หมาย SW จะตั้งค่าตัวเองโดย cross-over เครื่องเราไปต่อเครื่องเรา กรณี intf → s(config-if)# duplex auto → s(config-if)# speed auto

• SW Security : Security Remote Access → SSH (Secure Shell) TCP port 22, telnet : TCP port 23 → s(config-if) mdix auto

config : s-c(config)# ip domain-name ido → # crypto key generate rsa → # username admin pass cisco → line vty 0 15

→ transport input ssh → login local [Verify SSH : show ip ssh, show ssh]

• SW Port Security ผ่าน policy ที่ตั้ง MAC Addr. 1. un-trusted 2. trusted

s(config-if) # switchport mode access → # switchport port security → เลือก mode ที่ต้องการ

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

กรณีที่ Max #switchport port-security maximum max

Violation mode : ① protect : security violation protect mode

② restrict: security violation restrict mode

③ shutdown: security violation shutdown mode

for Staples [verify : show port-security int fa 0, show port-security address]

• Addr. Resolution Protocol (ARP) ARP cache ที่ตั้ง Mac Addr. ที่ map ระหว่าง dest (ไปจัดการต่อ Mac gateway)

IPv4 : Classless [subnet mask] : Variable Length Subnet Masking (VLSM) แบ่งทูน้ำตามต้องการ ⇒ ได้ออกมาเป็นทูน้ำ - เส้น

Fixed Length Subnet Masking แบ่งทูน้ำเป็นเท่าๆ กัน



Chapter 8 LAN Redundancy & Spanning Tree Protocol (STP)

- Issue with Layer 1 Redundancy
 - ① Mac Addr. instability: Mac Addr table oscillates between two ports
 - ② Broadcast storms: redundant loops
 - ③ Multiple frame transmission: start unknown unicast in highest timestamp frame
- STP prevents unblock port → block port → traffic from bypassing
- Root 1) w/ Root Bridge = low priority min 2) w/ path cost all 3) w/ Root port → path cost min to destination port
- 3) w/ segment w/ path cost min → q. BID min w/ designated port or w/ block port
- Config S1(config) # spanning-tree VLAN 1 root primary
 - In S1, S2, S3 S2(config) # w/ secondary [Verify: show spanning-tree]
 - 2 byte 1 bit 12 bit bbyte
- 3. Path Cost <
- 4. Sender's BID <
- 5. Sender's Path <
- w/ Extended System ID
 - B. Priority → B. Priority per VLAN + Extended Sys ID (VLAN) + Mac Addr. ∴ BID = 8 byte
- PVST+ w/ Load balancing between root / VLAN
 - Verify: show spanning-tree active
 - > Rapid PVST+ w/ Alternate port
 - runs set edge port -> # spanning-tree port fast
 - link type port type sw-sw or point-to-point # w/ bpdu guard enable
- config S1(config) # spanning-tree mode rapid-pvst → # int w/ p-p → # spanning-tree link-type point-to-point
 - ↳ for clear all: clear spanning-tree link-type point-to-point

Chapter 9 VLANs & Inter VLAN

- VLAN w/ partition (multiple broadcast domain in LAN) Layer 2 is SW machine's function VLAN is a part
 - goal: security, cost, bandwidth, broadcast domain, Vlan ID, name VLAN. [Verify: show vlan brief]
- in a multi-SW Environment
 - > VLAN Trunk set in Intf w/ supporting switch VLAN → can carry multiple > 1 VLAN
 - ③ config. int. intf # switchport mode trunk [Verify: show int tolo switchport]
 - > Tagging Ethernet Frames Dest MAC | Src MAC | Tag | Type/Length | Data | FCS → Tag multiplex VLAN w/ native Trunk
 - Native (base) VLAN w/ no tag & untag → receive intf 10 raw (Cisco)
 - Assignment: VLAN number 1-1005 (S1 config) ② VLAN database (In flash)
 - ① show config # VLAN 1006-4096 (S1 config) @ running-config (In NVRAM) II S1# VLAN database → # VLAN name 1006
 - assign port w/ VLAN w/ intf -if # switchport mode access → # switch port access VLAN num (if: 10 no such VLAN num)
 - verify show vlan name 10, show vlan summary, show int vlan num
 - Inter-VLAN Routing router set intf trunk Tag w/ meta "sub interface"
 - [Verify: show vlan, show ip route, show running-config]
 - config ① set basic routing (set ip add, no sh) 2. R(config)# interface g0/1 ⑩ -subif # encapsulation dot1q 10 → # ip address ip subnet mask
 - VLAN

Chapter 10 VTP (VLAN Trunking Protocol) to manage VLAN & NAT (NW. Addr. Translation)

- VTP ms manage gw VTP v2: ms manage in domain
- Operation ms update VTP revision number 32bit (0-4294967295) (both sides)
 - 3 mode 1. Server can add, remove, rename VLAN within domain w/ config ③
 - 2. Client receives VTP in process, to VTP msg doesn't trunk
 - 3. Transparent can add, remove, rename VLAN, but doesn't config
- Config 2 w/ Cisco 1) SW cisco 2) w/ trunk (between SW) 3) w/ domain 4) 2.3 mode
 - 1) in global configuration vtp version 2 → # vtp domain 10 → # vtp password pass → # vtp mode server | mode client
 - 2) in VLAN configuration vtp v2-mode
 - [Verify: show vtp status/counters] ↗ # vtp server | client | transparent
 - Pruning manage traffic in-between interface (down ↓ to config in interface) to remove VLANs
 - sc(vlan) # vtp pruning → w/ interface → S(config-if) # switch port trunk pruning vlan remove VLAN-num
 - NAT was private ip ↔ publish/real ip
 - terminology 4 type 1) Inside local Addr. (private ip) 2) outside local Addr.
 - 3) Inside global Addr. 4) Outside global Addr.
 - Type 1. static route [map: 1:1] ① R(config) # ip nat inside source static local-ip global-ip
 - 2. Dynamic ② pool w/o Global/Real ip [map: 1:many ↔ 1] real IP to local IP
 - { ② # ip pool & start-ip end-ip { netmask subnetmask } prefix-length }
 - 3. PAT (Port Addr. Translation) port mapping to new addr. [map: many ↔ 1] ③ set ACL ④ ip nat inside source list ACL-num pool
 - to overload ↑
(PAT over dynamic)
 - config 3 ways ① NAT ② INSIDE : R(config-if) # ip nat inside
 - ③ OUTSIDE: R(config-if) # ip nat outside

Feature	Server	Client	Transparent
Source VTP msg	✓	✓	X
Listen to VTP msg	✓	✓	X
Create VLANs	✓	X	✓ *
Remember VLANs	✓	X	✓ *

Private Internet addresses are defined in RFC 1918:

Class	RFC 1918 Internal Address 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

for Staples Chapter 11 EIGRP

▶ EIGRP (Enhanced IGRP)

• Characteristics

◦ Basic features Cisco-proprietary منذ 1992 ◦ classless version of IGRP ◦ ไม่ต้องมี subnet mask ใน routing protocol

◦ DUAL (Diffusing Update Algorithm) กรณี loop-free & back-up path ของ routing domain \rightarrow in best path
▪ กรณี routing มาก very fast convergent (converge time < OSPF) กรณี backup path (กรณีต้องรอ)

◦ Establishing Neighbor = สองตัวร่วมกันใน directly 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

EIGRP can update

= RIP and neighbor adjacencies are used by DUAL (how to maintain)

update in routing table
when down link ↑

◦ Partial and Bounded = update แค่ส่วนที่เปลี่ยนแปลง ไม่ต้อง update ทั้งหมด สำหรับการติดต่อเพียงบางส่วน :: update < RIP

◦ Equal and Un-equal Cost = กำหนด行政距離ต่างๆ ของ route ที่ต้องการได้ เช่น

OSPF, RIPV2 Load Balancing \rightarrow if cost \neq but. ให้ load balance ทำ

▪ PDMS (Protocol-Dependent Modules) ที่ต้อง protocol ที่ต้องการ เช่น IPv4, IPv6, legacy protocol IPX และ AppleTalk
▪ topology \leftarrow OSPF \neq DUAL \rightarrow in shortest path in backup shortest path

▪ PDMS ข้อมูล

▪ maintain EIGRP neighbor and topology table (Neighbor Table \rightarrow ที่ Topology Table \rightarrow ที่ routing table ที่ routing)

▪ metric ของ DUAL ที่ DUAL ไม่ routing table

▪ implement filtering and access lists \rightarrow in redistribution with other routing protocol

▪ RIP is EIGRP Transport layer protocol สำหรับ delivery & reception ของ EIGRP packets

▪ msg ของ application layer ที่ maintain ต้อง msg ที่ต้องของ EIGRP

for Staples

▪ ต้องรู้ว่า RIP packet คืออะไร (msg \approx OSPF)

▪ Reliable packet require explicit ack ณ dest & Update, Query, Reply

▪ Unreliable packet do not require ack ณ dest & Hello, ACK

▪ ไม่มี authentication (no encrypt routing update) แต่ recommend (ในที่สุด) Cathen \approx RIPV2, OSPF
(protocol 88 แต่ transport layer) 0-00-8E-00-00-0A IGRP multicast 224.0.0.9
IPV4 : 224.0.0.10, IPV6 : FF02::A อย่างไรก็ตาม RIPV1 broadcast 255.255.255.255

◦ Packet Type routing update or queries EIGRP multicast IPV4 : 224.0.0.10, IPV6 : FF02::A อย่างไรก็ตาม RIPV1 broadcast 255.255.255.255

◦ Hello \rightarrow 通知 adjacency router 2 ตัวที่เป็น neighbor ที่มี response, ดังนั้น unreliable

◦ Update \rightarrow update info ณ dest, update info ณ routing ที่ต้อง neighbor router

◦ Acknowledgement \rightarrow จะต้อง update ก่อน ACK

◦ Query \rightarrow request info. routing ณ neighbor router } แจ้งต้อง info ณ routing ที่ต้อง query ที่ต้อง router ที่ต้อง \rightarrow ไม่ reply ณ neighbor

◦ Reply \rightarrow ตอบ query ที่ reply

▪ Implement EIGRP for IPV4

▪ Autonomous System (AS) is a collection of nw ภายใต้ control ของ single authority (อ้างอิง RFC 1930)

▪ AS number \rightarrow exchange routes between AS

\rightarrow managed by IANA & assigned by RIRs to ISPs, Internet Backbone providers, and institution ขนาดใหญ่

\rightarrow 16 bit : 0-65535 \Rightarrow since 2007, 32 bit : over 4 billion | A verify: show ip eigrp neighbors

show ip protocols

▪ configure: R(config)# router eigrp AS-# (\approx router-id @ OSPF)

show ip route

show ip protocols R(config-router)# eigrp router-id \rightarrow ต้องตั้งเป็น interface ที่ต้องเป็น loopback intf. \rightarrow IPV4 addr ที่ต้องเป็น Active

R(config-router)# network nw-number [wildcard-mask] \rightarrow ต้องตั้งเป็น interface ที่ต้องเป็น serial

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

(loopback)

for Staples



□ Operation

- Initial Route Discovery
 - R1 say hello to neighbor router
 - R2 answer hello or update info
 - R1 send ACK & update info.
 - ④ R2 DUAL finds best route and update routing table

Metrics BW [lowest], $Delay$ [longest], $Reliability$ [worst], $Load$ [worst] α / β value : show interface

Default Composite Formula:
$$\text{metric} = [k_1 * bw + k_2 * delay] * 256$$

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

Complete:
$$= \left[k_1 * bw + \frac{(k_2 * bw) + k_3 * delay}{(256 - load)} \right] * \frac{k_5}{\text{reliability} + k_4}$$

- $R(\text{config-router})$ # metric weights tos k_1 k_2 k_3 k_4 k_5 - set bw : $\text{intf} \rightarrow R(\text{config-if})$ # bandwidth k_i kilobits/bw-value

o DUAL and the Topology Table (FSM (Finite State Machine)) \rightarrow show ip eigrp topology [all-link], show ip route

+ Successor (S) [router \rightarrow dest] = neighbor router \rightarrow destination \rightarrow min \rightarrow cost

+ Feasible Successor (FS) [neighbor \rightarrow Feasible condition] = Backup path (neighbor \rightarrow dest)

+ Reported Distance (RD) [distance \rightarrow neighbor \rightarrow report distance \rightarrow dest] = "advertised distance" \rightarrow dest \rightarrow cost in this hop

+ Feasible Distance (FD) [distance \rightarrow [VMS]] = ph distance \rightarrow dest. now \rightarrow cost lowest \rightarrow dest.

□ IPv4 Issue

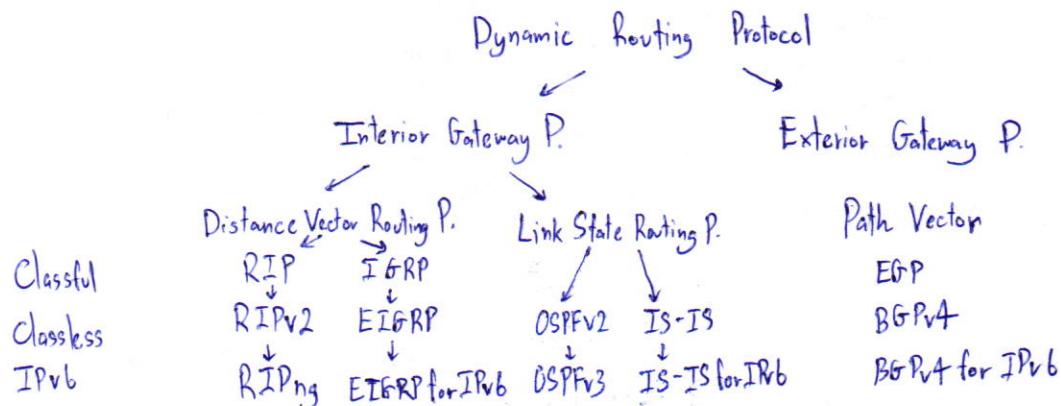
■ Need for IPv6 \rightarrow IPv4 ip จำกัด (private ip, NAT), IPv6 IoT จำกัด

■ coexistence (coexistence)

- Migration IPv4 \rightarrow IPv6 Techniques: ① Dual stack = running if want dual stack user

② Tunneling (convert v4 but. core to support) = วิธีการที่จะทำให้ IPv4 สามารถใช้ IPv6

③ Translation (NAT) = IPv6 \leftrightarrow IPv4



■ IPv6 Addressing: 128 bit uses 8 bytes [1 byte = 2 bytes = 16bit] \rightarrow represent base 16, 32 + bit

• 00000000000000000000000000000000 IPv6

Rule 1 - Omit Leading 0s \rightarrow partition "0" \rightarrow 00000000000000000000000000000000 \rightarrow 0-F

Rule 2 - Omit All 0 Segment \rightarrow "0" \rightarrow :: \rightarrow :: \rightarrow :: only

■ Type of IPv6 Address \rightarrow 3 bits use "001" or "2000::/3"

• IPv6 Addr. Type ① Unicast : ② Global Unicast ③ Link-local ④ Unique Local
 \rightarrow 3 bits use "001" or "2000::/3" \rightarrow 3 bits use "001" or "2000::/3" \rightarrow 3 bits use "001" or "2000::/3"
 static config \rightarrow intf. no global (④ Unique Local), 2 bits use FE80::/10 \rightarrow link-local
 \rightarrow ip6 address ip6-addr/prefix-length \rightarrow no shutdown

② Multicast

③ Anycast ร่วมกับ device

■ IPv6 Prefix Length = 0-128, most LANs is /64 LAN วงกว้าง 64 bit

Static routing

R(config-if)# ip route nw-ip subnet-mask { ip addr. /exit-intf }
ip route 0.0.0.0 0.0.0.0

Dynamic Distance Vector Routing P.

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

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

(\Rightarrow) RIR \leftrightarrow static: R(config) # router rip \Rightarrow R(config-router) # [redistribute static | default-information originate]
In router \Rightarrow set default route @ intf. \Rightarrow 2 protocols

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

-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

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

-set bw: in intf \Rightarrow R(config-if) # bandwidth kbits-bw-value

Link State Routing P.

1-65535

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

set bw: in intf \Rightarrow R(config-if) # bandwidth 64

ip ospf cost 15625

get cost:

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

av: clear ip ospf process

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

R(config) # router ospf process-id

\Rightarrow R(config-router) # default-information originate

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

-ACL

as1) R(config) # ip access-list [Standard | extended] name

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

set @ intf. in intf \Rightarrow R(config-if) ip access-group { ACL-num | ACL-name } in/out

av: no access-list ACL-num

-Securing VTP with standard IPv4 ACL: R(config-line) # access-class ACL-num h in [vrf - aslog] 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 nw-ip subnet-mask

R(dhcp-config) # default-router ip-address-gateway

1. basic configuration

S(config) # interface vlan N ⇒ S(config-if) # ip address ip-addr subnet-mask ⇒ no shutdown

default gateway: S(config) # ip default-gateway ip

2. Configure switch port

switch

duplex communication^ S(config-if) # duplex full ⇒ speed 100

auto-mdix : intf ⇒ S(config-if) # duplex auto ⇒ speed auto ⇒ mdix auto

- security Remote Access

+ SSH (TCP port 22) S(config) # ip domain-name cisco.com ⇒ crypto key generate rsa ⇒

username admin password ecna ⇒ line vty 0 15 ⇒ S(config-line) # transport input ssh ⇒ login local

+ Telnet (TCP port 23)

- switch Port Security : intf ⇒ S(config-if) # switchport mode access ⇒ switchport port-security

+ static secure Mac addr. ⇒ switchport port-security mac-address MAC-ADD

+ dynamic

⇒ 1 ————— 2 mac-address sticky

+ Max Mac Address

⇒ 1 ————— n maximum MAX

+ Violation Mode

⇒ 1 ————— n violation [protect] [restrict] [shutdown]

3. STP

nvu 1 S(config) # spanning-tree VLAN 1 root [primary] [secondary]

nvu 2 S(config) # spanning-tree VLAN 1 priority 2+3+7+6 < 63 priority of

II Rapid PUST

+ Port Fast intf. ⇒ S(config-if) spanning-tree port fast

+ BPDU Guard intf ⇒ S(config-if) Spanning-tree bpdu guard mode

+ config : S(config) # spanning-tree mode rapid-pvst ⇒ intf. ⇒ S(config-if) # spanning-tree link-type point-to-point

+ clear STP ⇒ S# clear spanning-tree detected-protocol

4. VLAN

① mode - S(config) # vtp version 2 ⇒ vtp mode [server] [client] [transparent] ⇒ vtp domain name ⇒ vtp password pass

② trunk - intf ⇒ S(config-if) # switchport mode trunk | ③ VLAN@server - S(config) # vlan num ⇒ name name

④ assign intf intf ⇒ switchport mode access ⇒ switch access vlan num

⑤ Set inter-VLAN R(config) # int folo.10 ⇒ description vlan 10 ⇒ encapsulation dot1q 10 ⇒ ip address ip subnet

5. NAT Static - R(config) # ip nat inside source static local-ip global-ip ⇒ intf # ip nat [inside|outside]

dynamic - R(config) # ip nat pool name start-ip end-ip netmask netmask prefix-length prefix

⇒ access-list Acl-num permit source [source-wildcard] ⇒ ip nat inside source list Acl-num pool name overload

⇒ intf ⇒ # ip nat [inside|outside]

* PAT = same dynamic intf in "overload"