ชื่อ-สกุล ณัฐรกิตติ์ นวังเจริญ กระดาษแผ่นที่ 1 รหัสนักศึกษา

for Staples

(1)

- NW diagrams - โครงสร้าง nw การเชื่อมต่อ
  - Physical = บอก port / interface การเชื่อมต่อต่างๆ
  - Logical = บอกพวก ip
- components of NW
  - HW = 1. end devices = ขยังจากมันเป็นคน
    2. intermediary devices อยู่ระหว่างอุปกรณ์ switch เลือกทางออก, router ทางที่ดีสุด
    3. nw media = สื่อกลาง
- Types of nw
  - size → small home = ใช้งานด้านนอก
    small office = config ด้านนอกมาใช้ด้านใน
    medium to large = มีการเชื่อมต่อภายใน 100-1000 เครื่อง
    world wide = internet
- infrastructur
  - LAN มี 1 กลุ่ม Admin ดูแลอยู่
    WAN มี หลายกลุ่ม Admin
- Reliable NW
  - fault Tolerance = ทนต่อความผิดพลาด
    Scalability - ปรับขนาดได้โดยไม่ส่งผล
    security - ตำกัด การเข้าถึง
    Quality of Service = แต่ละ service ใช้ Quality ไม่เท่ากัน
- Type of connection in LAN
  ข้อจำกัด (UTP cat 5) - BW 100 Mbps : ยาว 100 m มีตรงและ cross ต่ออุปกรณ์เหมือนกัน ยกเว้น sw-hub, Pc-router
  WAN connection = เชื่อมระหว่าง router → DCE (female) ใส่ clock 56000 และ DTE
  console = router - Pc

for Staples

(2)

Port : 0-1023 (well Known port) destination port Ex. 20 FTP(data), 21 FTP(control), 25 SMTP
1024-49,151 : registered port = publish ให้ทุกคนใช้
49,152-65535 : dynamic " random port " source port

IPv4 : class A 0-127
B 128-191
C 192-223
D 224-239 multicast
E 240-255 experimental

Private
A 10.0 - 10.255 | 10.0.0.0/8
B 172.16.0 - 172.16.255 | 172.16.0.0/12
C 192.168.0 - 192.168.255 | 192.168.0.0/16

Mac - Ethernet = 48 bit ฐาน2 = 12 ตัวฐาน 16
- IEEE = กำหนด 3 byte (24 bit) code " Organizationally Unique Identifier (OUI)
  - ทุก MAC ที่กำหนดให้ NIC หรือ Ethernet device อื่นๆ → ใช้ OUI ที่ 3 byte แรก
  - ทุก MAC ที่มี OUI เหมือนกันต้องกำหนด unique ที่ 3 byte สุดท้าย

Message Delivery - unicast = มาเครื่องปลายทางใน nw เดียวกัน
- Broadcast = ส่งทุกเครื่อง
- multicast = ส่งหลายเครื่อง รับเฉพาะเครื่องที่เปิด service ไว้

cisco IOS - function 1. Addressing, 2. interface 3 routing 4 Managing Resource 5 Security 6. Qos

Router & SW Boot Sequence

ROM 1. POST (Power on Self test) → check hw ว่าพร้อมไหม
2. Run boot loader sw
ROM 3. Boot loader dose low-level cpu intialization (mini IOS)
Flash 4. " intializes the flash filesystem

for Staples

TFTP 5. " locates & load a default IOS มา run บน RAM

③ Packet Forwarding Methods

- Process switching = ทุก packet ที่เข้า router ↑ process @ cpu ↓ ออก interface ใหม่
- Fast switching = สร้างข้อมูลที่ใช้ forward ไว้ล่วงหน้า
- cisco Express Forwarding (CEF) = forward ได้เร็วสุด

connect Devices

- Default geteway → ใช้ .1 หรือ 254
- Enable IP = statically, Dynamic ก็ได้ DHCP

Switching packet ใน NW

หา dest.ip (L3) → เทียบใน routing table → หาทางออกไปเทียบ mac → ใส่ des.MAC (L2)

Path Determination

Packet เข้าใน interface → match couting table → dest.IP match subnet

↓

match กับ interface → Y → check ARP cache

↓ N

remote nw → Y → encap frame → next hop

↓ N

มี default route ? → Y → encap frame → next hop

→ N → ทิ้ง packet ส่ง ICMP กลับ ขา S.IP

Best Path ; lowest metric => Dynamic routing protocol

1. Routing Information Protocol (RIP) = จำนวน hop
2. Open Shortest Path Frist (OSPF) = BW มากสุด
3. Enchanced Interior Gate way Routingm Protocol (EIGRP) = BW, delay, load, reliability

load balancing = ใช้ทุกเส้นทางน้ำหนักเท่ากัน

Administrative (AD) = ยิ่งน้อยยิ่งเชื่อถือ protocol ต่างๆ → connect = 0, static = 1, Internal EIGRP = 90, OSPF = 110, RIP = 120

D 10.1.1.0/24 [90/2170112] via 209.165.200.226, 00:00:05, serial 0/0/0

1 2 3 4 5 6 7

1. ได้มาจาก C = directly connect ; B = BGP ; D = EIGRP ; S = static
2. destination ที่รู้จัก
3. AD
4. cost
5. next hop IP
6. ระยะที่ไร
7. ทางออก

Routing

① static

ข้อดี ; security, ใช้ resource น้อย

ข้อเสีย ขึ้นกับการ scalibility

4 type 1. standrad
2. Default → ใช้เมื่อ dest.ip ไม่ match
3. summary
4. Floating = backup

② Dynamic → auto

2.1 EGPC Exterior Gateway Routing Protocol): BGP

2.2 IGPC Interior ——→ ;; RIP, OSPF, EIGRP, IS-IS

classful Address → update macclass

classless Inter-Domain Routing

- summaarization: เพื่อนบ้าน 1. อยู่ติดกัน 2. ไปทางเดียวกัน

ถ้า set เพื่อนต้อง no ip เก่าก่อน ⇒ แปลง ip → ฐาน 2 → group bit ที่เหมือนกันสุด

VLSM

- fixed Length Subnet Masking ;; Prefix ใหม่ − Prefix เก่า = จำนวน bit ที่เปลี่ยน

เปลี่ยนทุก bit เป็น 1 แล้วนับ ตำแหน่ง 10 ไป + ip เรียงสลับ กบ + 1 ที่ใช้เปลี่ยน

ชื่อ-สกุล ณัฐกิตติ์ หลังเจริญ กระดาษแผ่นที่ 2 รหัสนักศึกษา

for Staples

4. Distance Vector Routing Protocol RIP v.1

• Dynamic Routing Protocol
- f^n: share info ระหว่าง router, auto update routing table when topology เปลี่ยน (มีปัญหา), หา best path
- purpose: หา remote nw (nw ที่อยู่ห่างไกล), ปรับปรุง routing info, เลือก best path ไป dest. nw, can หา new best path ถ้า path ปัจจุบันไม่ว่าง
- component: 1. Algorithm: ใช้เพื่อส่งออก routing info & best path
  2. Routing protocol msg: ใช้หา neighbor & แลกเปลี่ยน routing info. (best path)

	Dynamic routing	Static routing
- ความยุ่งยากในการ config	- ∝ ขนาด nw (same เดิม)	- ∝ ขนาด nw (เพิ่ม command ทุก router)
- Required ค.รู้ Admin	- Advanced (bec. config basic → nw ใหญ่ + ซับซ้อน ตรวจปัญหา)	- No เพิ่มเติม (กำหนดเส้นทางเองโดย command)
- Topology change	- ปรับ auto	- admin config ใหม่ all
- Scaling	- เหมาะ simple & complex (router ที่ไม่ได้ directly ไปก็ได้)	- เหมาะ simple topologies
- Security	- น้อยกว่า	- มากกว่า
- Resource usage	- ใช้ CPU, mem (เก็บ routing info), link bandwith	- No เพิ่ม
- Predictability	- route ∝ current topology	- route → dest. ทางเดิมตลอด

• Classifying Routing Protocol

DRP
- Interior Gateway Protocol (IGP) (ใน AS)
  - Distance Vector P.
    - ประกาศ vector [distance, direction]
    - incomplete view ของ nw topology
    - periodic update (รายงานเป็นรอบ)
    - RIP (Routing Info P.) → RIP
    - IGRP (Interior Gateway Routing P.) → EIGRP (Enhanced Interior Gateway Routing P.)
  - Link-state P.
    - complete nw topology (เก็บข้อมูล all)
    - update ไม่ใช่ periodic
    - OSPF (open shortest Path First)
    - IS-IS (Intermediate system to Intermediate System)
- Exterior Gateway Protocol (EGP) (ภายนอก AS)
  - BGP (Border Gateway P.)

· Autonomous system (AS) เป็นกลุ่มของ router ภายใต้ ก. ควบคุมของ single authority (1 กลุ่ม policy or ผู้ดูแล)

for Staples

• 2 type
1. classful routing p. → update ตาม class ไม่ส่ง subnet mask ใน routing update
2. classless " → ส่ง subnet mask ใน routing update

• convergence: จุดกำเนิด when routing table ของ all router มีสถานะคงที่ (ไม่เปลี่ยนแปลง)
  ↳ 2 type: Slower: RIP & IGRP, Faster (ส่ง ก. update when เกิด ก. เปลี่ยนแปลง, ดี >): EIGRP & OSPF

• Routing Protocol Metrics
- Metric: ค่าที่ใช้ในการวัด/นำ ในการไป dest. nw จะเลือก best path ไหน เช่น hop count, Bw, cost, Delay, load, Reliability
- load balancing: nw มีเส้นทาง > 1 ที่ metrics เท่ากัน → สลับในการทำงาน (แต่ละเส้นทางเท่ากัน)

Administrative Distance of a Router (AD) → ใช้เลือก protocol ใน routing

connected	static	Internal EIGRP	OSPF	RIP	EIGRP summary	E.BGP	IGRP	IS-IS	Ex.EIGRP	Int.BGP
0	1	90	110	120	5	20	100	115	170	200

Distance Vector Routing Protocol (RIP, IGRP, EIGRP)
- Distance Vector Technology ต้องรู้ 2 สิ่งคือ 1. vector or direction ก. แลกเปลี่ยนควรทุกคนทุก ② Distance to final dest (cost)
- ลักษณะที่สำคัญ: peridic (ตามเวลาที่ผ่านมา) update, neighbor (① ใกล้เคียง), broad cast (255.255.255.255) update, เอา routing table all ไป update
- " ของ Routing Protocol (เกณฑ์ที่ใช้ check ว่าดีหรือไม่?) 1. Time to convergence ⇒ ตอนที่เป็น steady state ของ routing table ที่เกิด ก. ปป. แล้ว ② scalability ขยายมากสุดเท่าไร ③ resource usage ④ Implementation ⑤ maintanance

• Nw Discovery (ทำ basic configrou)
  มี 3 - 1. cold state: Router Initial start up
  2. Initial Exchange of Routing info. → ทำตารางแลกเปลี่ยน
  3. Exchange of routing info → update (เพิ่ม hop count) routing info., ของ router อีกตัวว่าห่างเท่าไร

Routing Table Maintanance
• Periodic update: RIP update timer (default 30s), Invalid timer (info เริ่ม lost) (default 180), Holddown timer (ถ้า down → hold ไว้ว่าไม่ up ตอนไหน) (default 180), Flush (ทิ้ง), default (240)
• Bounded (ขอบเขต) Update: EIGRP → update แค่ที่เปลี่ยน
• Triggered Update → update เมื่อไม่รอ periodic time
• Random Jitter → ใช้ใน nw ที่เป็น multiple access router หลายตัว เชื่อมต่อกัน → ไม่เกิด update พร้อมกัน ∴ ใช้ Random

for Staples

ปัญหา standard DV.

1. routing loops เกิด when intf ไหน down จะถูกลบออกจาก table → ค่าเก่าใน neighbor จะกลับมา update (วน update → hop เพิ่ม → ∞)

↓ แก้ปัญหา: 1. set max hop = 15 → if hop = 16 → unreachable (เส้น down ทางไปแล้ว)
2. holddown timer (ถ้า intf down → hold)
3. Split Horizon Rule → ไม่ส่งข้อมูล update กลับไปทาง intf ที่ได้รับ update มา
4. Route Poisoning → ①. เกิด down set unreachable ②. ส่ง unreachable ซึ่งเป็น poision ออกไปใน intf นั้น hop = 16
5. ③ with ④. → ถ้าเกิด unreachable จะ over rule split horizon โดยส่ง ip intf ที่ down (hop = 16)
6. IP & TTL (Time to Live) เกิด ก.วน update but. จะหยุด when TTL = 0

	RIP_{v1}	RIP_{v2}	IGRP	EIGRP
speed convergance	slow	slow	slow	fast
scalability - size nw	small	small	small	Large
use of VLSM	×	✓	×	✓
Resource usage	Low	Low	Low	Medium
implementation & maintain	simple	simple	simple	simple

- RIP v.1 (AD = 120)
  - คุณสมบัติ: classful, DV = metric = hop count = hop count > 15 unreachable • update broadcast ทุก 30 s

msg มี 2 type
1. request → ส่ง routing table → ไป intf ที่ config ไว้เพื่อรอรับ ก. update
2. Responce → ส่ง info. ของ routing table

ip addr. แบ่งตาม class A, B, C

- Basic RIP_{v1} config ①. ทำ basic config ②. สั่ง router rip + กำหนด nw R1(config)# router rip
  R1(config-router)# network nw ip ที่ต้องการรู้จัก
- Verification (ตรวจสอบ) & troubleshooting (แก้ ก. ปัญหา): show running-config or ip route or ip protocol, debug ip rip
  - passive intf command (ไม่ update intf ที่ไม่ต้องการ) R(config-router)# passive-interface intf-type (Fa/E/S) intf-number (0/0, 0/0/0)
- Automatic Summarization: RIP auto Summarizes classful nw → เพื่อลด size routing table
  ข้อดี: ลด size routing update, single router (แสดงผลใน ก.หา multiple route เร็วมากใน routing table
  ข้อเสีย: ไม่ support discontiguous nw (major nw เดียวกัน but ไม่อยู่ติดกัน) → อาจจะเกิด load balancing ได้
  - boundary Routers: summarize RIP subnet from 1 major nw to anther
  - Processing RIP update → ดูตัวที่ส่ง & ที่ update ไป (intf) ว่า classful เดียวกันไหม? → Y: update subnet nw เช่น 172.16.1.0 → N: update classful เช่น 172.16.0.0
  - default route & RIP_{v1} ถ้าไม่เจอทางออกใน routing table (bec. ไม่มี protocol) → จะออก default route
    R(config)# ip route 0.0.0.0 0.0.0.0 S0/0/1
    default info originate command → เพื่อ update ไปใน rip ว่าเชื่อม: static ↔ dynamic

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

⑤. RIP_{v2} & Access control lists

V_1	V_2
classful (ไม่ส่ง subnet mask, ไม่ support CIDR)	classless (update subnet mask, support Variable length Subnet Masking (VLSM), support Route summarization (Prefix Aggregation))
not support discontiguous subnet	update next hop address
not support VLSM bec. ไม่ส่ง subnet mask	ใช้ authentication routing (รองรับ discontiguous มากกว่า)
routing update → broadcast	Routing update → multicast

* ใช้ Timer ป้องกัน routing loop
  ใช้ split horizon or split horizon with posion reverse
  ใช้ triggered update ; max hop count = 15

. ข้อดีกว่าของ RIP_{v1}

เป็น virtual interface can เพิ่มใน routing ให้ update ออกไปได้
- loopback intf → ping ไป ip virtual intf → reply ได้
- Null intf → ใช้ดักกันทำหน้าที่ channel ที่เราไม่ต้องการ. → วิ่งออก null intf → packet discard เอง → time out
- static route & null intf → null intf จะรับหน้าที่เป็นทางออกของ static route
  R(config)# ip route summary-route subnet-mask Null0 (major-nw) → ของ static superet ronte

. Route redistribution (จัดสรร) → อยากให้ rip รู้จัก static ตัวที่เพิ่มใหม่ (โดย rip ส่ง static ไปด้วย): R(config-router)# redistribute static

. Verify & Test connectivity: show ip interface brief, ping (ผล ! = ได้, U = ไม่ได้, . = timeout), traceroute

. RIP_{v1}: classful, ไม่ส่ง subnet mask, summarize nw ที่ major nw boundaries, if nw เป็น discontiguous & RIP_{v1} config convergence จะไม่เกิด

. ตรวจ routing table debug ip rip (content of routing update), ถ้าเป็น RIP_{v1} จะไม่ได้ subnet mask รวมไป nw addr.

ชื่อ-สกุล ศุภฤกษ์ หวังเจริญ กระดาษแผ่นที่ 3 รหัสนักศึกษา

for Staples

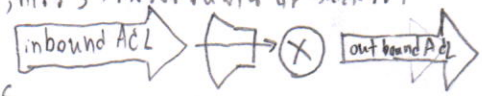
RIPv2

- config
  - Enabling & verify (ตรวจสอบ) RIP_{v2} → show ip protocols
  - config RIP → RIP_{v1} → รับได้ทั้ง v_1 & v_2 but ส่งได้แค่ v_1
    - → RIP_{v2} → รับ & ส่งได้แค่ v.2
  - Auto - Summary & RIP_{v2} → auto sum route @ major nw boundries
    - → sum route ด้วย subnet mask ที่น้อยกว่า classful subnet mask
  - disabling Auto-Summary: no auto-summary bec. when ถ้า nw topology ต้องเป็น discontiguous

VLSM & CIDR → verify info ที่ sent by RIP_{v2} debug ip rip
- → VLSM → เผยแพร่ nw addr. & subnet mask
- → CIDR → ใช้ supernetting (= bunch ของ contiguous classful nw ที่เป็น addr. เหมือน single nw)
  - → Verify show ip route, debug ip rip

Access control list = ควบคุม ก. เข้าออก → ตรวจสอบ → check → source → dest ผ่านไหม?
↳ ทำอย่างไร (เช่น FTP) ผ่านไหม?
=> ดูจาก conversation

- packet filtering ดู ①. dest, source @ L2 ② protocol ที่ใช้ ③ ไป nw ไหน, ที่ใช้ 3 → ทำการอนุญาตให้ผ่าน or block ไว้?
- Operation → ทำงานเป็น sequence statement
  - → last statement เป็น implicit deny → block → discard



- Standard IPv4 ACLS
  - check sourse addr.
  - ปกติ permits or denies ทั้งชุด protocol
    access-list 10 permit 192.168.30.0 0.0.0.255
  - number ACL : 1-99 & 1300-1999

Extendend IPv4 ACLS
- check sourse & destination addr.
- ปกติ permits 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

- wild card → invert ของ subnet mask
  - → 0 = match/fix , 1 = ignore/อะไรก็ได้
  - → วิธีหาจาก set ของ ip (match range) ①. หาค. สัมพันธ์ของ bit ที่ซ้ำกันใส่ wild card mask ตน.นั้น=0 ② bit ที่เหลือใส่ 1
  - if กรณีคิดไม่ได้รองแยก Pattern or/and ต่อน ตอนสุดท้าย wild card จะ same กัน
  - → หา wild card ของ subnet = 255.255.255.255 − subnet mask
  - → key word → 0.0.0.0 = match all ใช้ host
    - R1(config)# access-list 1 permit 192.168.10.10 0.0.0.0
    - R1(config)# access-list 1 permit 192.168.10.10
  - → 255.255.255.255 = ignore all ใช้ any
    - R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255
    - R1(config)# access-list 1 permit any

for Staples

- Guideline for (3P) ACL creation
  - → one ACL/protocol = ctrl traffic flow บน intf, ACL ต้อง define แต่ละ protocol enable on intf
  - → one ACL/direction = ctrl traffic in 1 direction at time on an intf, แยก ACL ctrl in & out bound traffic
  - → one ACL/interface = ACL ctrl traffic for an intf, Ex. G0/0
- where → Extend ACL : @ close source → Standard ACL ; @ close destination

config ACLs (เลือกมา 1 แบบ คำสั่งในแบบเดียวกัน but ชื่อ can อยู่หน้าคำสั่ง (permit, deny, remark) ได้เลย)
- → standard → number ; Router(config)# access-list access-list-number deny | Permit | Remark source [source-wildcard] [log]
- → in intf ; Router(config-if)# ip access-group {access-list-number | access-list-name} {in | out}
- → ชื่อ Router(config)# ip access-list [standard | extended] name

ถ้า remove all : no access-list, ถ้าแก้ I no access list num# → สั่งใหม่ II no บรรทัด# → บรรทัด# คำสั่ง

ถ้า remove all: no ip access-group ถ้าแก้ → บรรทัด# คำสั่งใหม่ ↳ บรรทัดแทรก

- verify : show ip interface, show access-lists
- Securing VTY port → อยากให้คนนอกเข้ามา permit เฉพาะที่เราต้องการให้เข้า R(config-line)# access-class access-list-number {in [vrf-also] | out}

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

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]

for Staples

ที่เหลือ same standard & ใช้ทั้งแบบ number & name
-debug-output: debug ip packet ACL-number

⑥. OSPF & DHCP

- link-state Routing Protocol = เป็น protocol ที่ต้องสร้าง complete map ของ nw topology ขึ้นก่อน (มี info. all) → นำ shortest path first (SPF) (dijkstra)
  - เหมาะกับ: ① large nw, ② fast convergence ③ admin ควรมีค.รู้ดี
  - ขั้นตอน update ①. learn info ของ link ②. say hello neighbor ③. เอา info มาสร้าง link-state Packet (LSP)
  ④. router flood LSP to all neighbors → ได้รับแล้วเก็บลง db ⑤ router เอา all LSP เก็บใน db (รูปแบบ tree)
  + Adding OSPF → routing table
  - ข้อดี: ① สร้าง topology map ตามนำ shortest part, ② fast convergence ข้อมูลล่าสุด ③ LSP sent only when change topology (ให้เฉพาะข้อมูลที่ ปป. → คำนวณ shortest path) ④ hierarchical design (ทพ ในผู้จะทำ) → ลด resource bec. แยกข้อมูลให้ไม่area
  - ข้อเสีย: ① ใช้ mem ใน ก.เก็บ all link-state เยอะ ②. ใช้ CPU ในการคำนวณ ③ ตอนส่ง LSP ครั้งแรก ใช้ BW มาก

- OSPF (AD = 110)
  - ↳ stable: ① Neighbor show ip ospf neighbor ②. Topology (ทั้ง map) show ip ospf database ③ Routing (cal shortest path)
  - message → Encapsulating: MAC Dest. = multicast: 01-00-5E-00-00-05 or 01-00-5E-00-00-06
    Protocol field = 89
    → type OSPF Packet:
    - 01 Hello → ทุก 10s (default: multiaccess & point to point nw), ทุก 30s (default: nonbroadcast multiaccess [NBMA] nw) , disco default 4 times 40s
    - 02 Db Description (DBD) → synchronization db info
    - 03 Link-state Request (LSR) → request link-state
    - 04 " update (LSU) → send update link-state
    - 05 " Acknowledgment (LS Ack) → ตอบกลับไปเลย
  - operation: สร้าง ก.เชื่อมต่อ ①. Down state (เริ่มต้น) → ②. Init state (เริ่มส่ง hello) → ③ Two-way state (ตอบกลับ hello) → Exstate State → Exchange static → loading state → Full state (ระหว่าง router update ข้อมูลเรียบร้อย)
  - config Single-Area $OSPF_{v2}$ router ospf process-id → 1-65,535, เป็น locally significant
    R(config-router)# router-id 1.1.1.1 → ① ถ้าไม่ set ② ดู loop back, ③ active interface ip สูงสุด but. กรณี ไม่ทำ ④
    router ospf process-id
    network network-address wildcard-mask area area-id
  - OSPF cost → ใช้ BW คำนวณ (default reference BW = 10^8)

$$cost = \frac{10^8 \text{ bps}}{\text{intf BW bps}}$$

→ 10 GP = 100×10^8 → cost = 1
→ Gb = 10×10^8 → " = 1
→ fast = 10^8 → " = 1
→ serial = 1.544×10^6 → " = 64

→ ต้องการเปลี่ยนค่า cost
→ เปลี่ยนค่าที่ BW
- fast = 100
- Gb = 1000
- 10Gb = 10000

} จะค่อย 10^8

→ เปลี่ยนค่า BW; R(config-if)# bandwidth 64 (EIGRP & OSPF ดูลไปได้)

→ เปลี่ยนค่า cost: " ip ospf cost 1562 S

verify OSPF show ip ospf neighbor, show ip protocol, show ip ospf interface brief, show ip ospf

more config R(config)# ip route 0.0.0.0 0.0.0.0 loopback N
R " router ospf process-id
R(config-router)# default-information originate

- DHCP
  1. manual Allocation: admin assign IP
  2. Automatic Allocation: DHCPv4 auto assign addr จาก pool & ไม่มี lease (เช่า) time
  3. Dynamic Allocation: ไคร่เปิดขึ้นมาก่อนจะได้ ip ไป มี lease time → ถ้าหมด re ip ใหม่

ชื่อ-สกุล ณัฐภัทร์ หมั่นเจริญ กระดาษแผ่นที่ 5 รหัสนักศึกษา 59030908

or Staples

Chapter 11 EIGRP IPv6 & Routing

Dynamic Routing Protocol

Interior Gateway P. — Exterior Gateway P.

	Distance Vector Routing P		Linkstate Routing P		Path Vector
Classful	RIP	IGRP			EGP
Classless	RIPv2	EIGRP	OSPFv2	IS-IS	BGPv4
IPv6	RIPng	EIGRP for IPv6	OSPFv3	IS-IS for IPv6	BGPv4 for IPv6

EIGRP (Enhanced IGRP)

- Basic features △ cisco-proprietary (ใช้เฉพาะ protocolของcisco) ใช้ในปี 1992
ครับ classless version of IGRP ⊝ ทางเลือกที่เหมาะสมสำหรับ nw หลายprotocol, หากใบนี้ที่สร้างขึ้นบน cisco router เป็นหลัก
v DUAL (Diffusing Update Algorithm) = การันตี loop-free & back up path ทั่วทั้งหมดของ routing domain → หา best path
→ทำให้routingทำงานเร็ว รวมทั้งมี backup path เร็วขึ้นถ้า link down

v Establishing Neighbor = เชื่อมค.สัมพันธ์กับ directly connected EIGRP routers.

v Reliable Transport P. = RTP provides delivery of EIGRP packets to neighbors.
v Adjacencies = Adjacencies are used to track the status of these neighbors
= RTP and neighbor adjacencies are used by DUAL (ในก. maintain)
v Partial and Bounded = update เฉพาะส่วนที่มี ก.ปป. รวมถึง update ไปให้เฉพาะส่วนที่มี ก.ปป. ∴ update < RIP
v Equal and Unequal cost = ข้อมูลให้ Admin ดูแลระบบ เพื่อกระจาย ก.รับข้อมูลในเครือข่ายได้ดีขึ้น
Load Balancing ↳ค่า cost ≠ but ทำเป็น load balance ได้
△ ใช้ protocol-dependent modules (PDMs) เพื่อรองรับ protocol ที่แตกต่างกัน เช่น IPv4, IPv6, legacy P, IPX and Apple talk
△ PDMs รับผิดชอบ:
⊝ maintain EIGRP neighbor and topology table (Neighbor table →DUAL สร้าง Topology → ใส่ routing table เป็น routing / EIGRP เลือก successor
↳ เก็บ SP กับ backup SP
เก็บทั้ง topology ≠ OSPF ≠
⊝ คำนวณ metric ที่ใช้ DUAL ⊝ เชื่อมต่อ DUAL และ routing table
⊝ implement filtering and access lists ⊝ ทำ redistribution with other routing protocol
△ RTP is EIGRP Transport layer protocol สำหรับ delivery & reception ของ EIGRP packets
△ ไม่ใช่ทั้งหมดของ RTP packet จะส่งได้น่าเชื่อถือ (msg ≈ OSPF) = เป็น msg ที่อยู่ใน app layer, ใช้ maintain ข้อมูล, msg ต่างๆ ที่ส่งของ EIGRP
⊝ Reliable packets require explicit (ชัดเจน) ack จาก dest ⊝ update, Query, Reply
⊝ unreliable packets do not require ack จาก dest ⊝ hello, ack

△ รองรับ authentication (on encrypt routing update) และ is recommend (authen ≈ RIPv2, OSPF)
o packet Type = routing update or queries EIGRP multicast IPv4: 224.0.0.10 IPv6: FF02::A IGRP 224.0.0.9 RIPv1 broadcast 255.255.255.255
① hello → ใช้ adjacencies ระหว่าง router 2 ตัวที่เป็น neighbor กัน, ไม่ต้องรอ response, ส่ง unreliably
② Update → update info ของ dest, update info ของ routing ไปให้ neighbor router
③ Ack → พอได้รับ ก.update ก็ตอบ Ack
④ Query → request info routing จาก neighbor router } เวลาต้องการ ก. info ของ routing/อยากให้ Query ไปกับ router ที่อยู่ข้างๆแล้ว
⑤ Reply → หลังจากได้รับ query ก็ reply } reply ตอบกลับว่ามี หรือไม่มี

□ Implement EIGRP for IPv4
△ Autonomous System (AS) is a collection of nw ภายใต้ ก.ดูแลของ single authority (อ้างอิง RFC 1930)
↳ AS number → ใช้ exchange routes between AS
→ managed by IANA & Assigned by RIPs to ISPs, Internet Backbone providers, and institution ที่ต่อ...
→ 16 bit: 0-65535 → since 2007, 32 bit: over 4 billion △ verify: sh ip eigrp neighbors
sh ip protocols
sh ip route
△ configure: R(config)# router eigrp AS-#
" eigrp router-id → ถ้าไม่กำหนดจะใช้ loopback intf. → IPv4 Addr ที่สูงสุด active
" network nw-number [wildcard mask] or ไม่ใส่ก็ได้
ไม่ใส่ก็ได้ → " passive-interface type number [default]: ไม่ให้ update ไปที่ int นั้น

or Staples

or Staples

□ Operation

• Initial Route Discovery ตัวอย่าง ①. R1 say hello กับ neighbor router ②R2 อาจส่ง hello or update กลับมา ③. R1 ตอบ ack & update info ④ใช้ DUAL คำนวณ best route and update routing table

• Metrics: BW [lowest] (K1=1), Delay [สะสม] (K3=1), Reliability [worst (K4=0, K5=0)], Load [worst] (K2=0) ดูค่าใช้ sh int

Default composite formula: metric = [K1*bw + K3*delay] * 256 $= \left(\left(\frac{10,000,000}{bw}\right) + f\frac{\text{sum of delay}}{10}\right) * 256$

complete: $= \left[K1*bw + \frac{(K2*bw)}{(256-load)} + K3*delay\right] * \left[\frac{K5}{reliability + K4}\right]$

-R(config-router)# Metric weights tos K1 ... K5 - set bw เข้า int → bandwidth Kilobits -bw-value

• DUAL and the Topology Table ใช้ FSM (Finite state machine) ในการเลือกเส้นทาง ⇒ sh ip eigrp topology (all-link)

+ Successor(s) [router ที่ไป dest ดีสุด] = neighbor router ที่จะเลือกจะไป dest ที่มี min นี้สุด

+ Feasible Successor (FS) [มีได้ไม่ ≤ Feasible condition] = Backup path (เส้นทางสำรอง) → จาก neighbor

+ Reported Distance (RD) [distance ที่ neighbor ส่งข้อมูล report distance มาให้เรา] = AD จากตัวมัน → dest ที่ cost เท่าไร" ของอีก hop

+ Feasible Distance (FD) [distance ที่ไปทาง 5] = ค่า distance ที่คำนวณไปยัง dest nw เป็นค่า cost lowest → dest

D. IPV6

IPV6	vs	IPVv4
128 bit		32 bit
base 16		base 10

□ IPV4 Issue

Δ need for IPv6 → แก้ปัญหา ip ที่จำนวนจำกัด (private ip, NAT) ให้ใช้ IOT ได้

Δ ก. ใช้งานร่วมกัน (coexistence)

- Migration IPv4 → IPv6 Tech ① Dual stack = run ทั้งคู่ if พังอันใดไม่กระทบ user
② Tunneling (อันนี้ใช้ v6 but. เขาไม่ supports) = เปลี่ยนเฉพาะเรา แล้วส่งผ่าน v4 same เดิม
③. Translation (คล้าย NAT) = $IP_{v6} \Leftrightarrow IPV4$

IPV6 Addr: 128 bit แบ่ง 8 ส่วน [1 ส่วนมี 2byte = 16 bit] → represent base 16 ทุกๆ 4 bit

ออกแบบวิธีก.ทำนองนี้ IPV6

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

Rule 2 - Omit All 0 segment = แทน segment ที่มี 0 ติดกันยาวๆ ด้วย "::" ได้ครั้งเดียว

□ Type of IPv6 Addr

• IPv6 Addr. Type

①. unicast: ⓐ Global unicast ⓑ Link-local ⓒ Unique Local
→ 3bit แรก "001" or 2000::/3
ⓑ ใช้ภายใน intf. ที่อยู่ติดกัน (อยู่ใน local เดียวกัน)
↳ ในช่วง FE80::/10 = ใช้แบบทำให้รู้ว่าคือ link-local ไม่ได้ใช้นอก local

static config
↳ ipv6 address ipv6-addr/prefix-length → no sh

②. Multicast

③. Anycast = ส่งไปหลาย device

Δ IPv6 Prefix Length = 0-128, most LAN is /64 be. LAN เล็กสุด 64 bit

□ IPv6 Routing

• config static route

ipv6 route ipv6-prefix /prefix-length next hop exit

Δ ได้ก่อน config ให้ run routing ipv6 ipv6 unicast-routing เปิดการใช้

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

Δ Default static IPv6 Route

ipv6 route ::/0

Δ verify: show ipv6 route static

• config EIGRP for IPv6

ipv6 unicast-routing

ipv6 router eigrp AS-#

eigrp router ip 2.0.0.0 → รูปแบบ IPv4

no sh

Δ nw command: เข้า int f → ipv6 eigrp AS-#

but passive-int ยังคงใช้ใน global config same เดิม

Δ verify: show ipv6 Eigrp neighbors, sh ipv6 protocols, show ipv6 route