ชื่อ-สกุล นางสาว ลัวพร วดีรัตน์ กระดาษแผ่นที่ 1 รหัสนักศึกษา 68088958

for Staples

Component of network

- end of user
- hub
- switch
- router
- LAN —— straight
- ------ cross → อุปกรณ์เดียวกัน อีกฝั่ง sw, hub, router, PC
- WAN ~
- Wireless))))
- Tropology Diagram
  - Physical การต่อ
  - Logical IP

Type of NW

LAN 1 LAN : 1 admin
WAN : 1+ admin

Reliable NW

- Fault Tolerance ทนความผิดพลาด
- Scalability ปรับเปลี่ยนขนาดได้ ไม่กระทบส่วนอื่น
- Security ข้อกำหนดการเข้าถึงภายใน ป้องกัน access ภายนอก
- Quality of service แต่ละ SV คุณภาพต่างกัน

OSI	TCP/IP	
7. Application		Physical: Timing and sync bit
6. Presentation	Application	(L2) Data: Dest, src Physical address
5. Session		(L3) NW: Dest, src logical NW address
4. Transport	Transport	(L4) Trans: Dest, src Process Num. (Port)
3. Network	Internet	Upper: Encoded App Data
2. Data link	Network Access	L2: MAC Addr.
1. Physical		if NW เดียวกัน ใส่ MAC เครื่อง else ใส่ default gateway

*NW เดียวกัน : NW Addr. & broadcast เดียวกัน

Messege Delivery

- Unicast => ส่งหาเครื่องปลายทางโดยตรง ๆ NW เดียวกัน
- Broadcast => ส่งหาทุกเครื่องใน NW เดียวกัน broadcast ip/NW = 255.255.255.255
- Multicast => ส่งหลายเครื่อง รับเฉพาะที่เปิด service ไว้ ขึ้นต้นด้วย 01-00-5E-xx-xx-xx

IPV4

192.168.1/24 ← Prefix range
255.255.0 → subnet mask
192.168.1.255 → broadcast ip addr.
255.255.255.255 → broadcast NW
↳ ทุกเครื่องใน NW นี้ได้รับ

for Staples

(0, 2^7) 1-127 — 8 bit — Host สามารถมีได้ 2^{24} → useable $2^{24}-2$
Class A: NW.Host.Host.Host subnet = 255.0.0.0
($10, 2^8$) 128-191
Class B: NW.NW.Host.Host " = 255.255.0.0
($11, 2^8$) 192-223 (Host 2^{16})
Class C: NW.NW.NW.Host " = 255.255.255.0 (Host 2^8)

CIDR Prefix

IP = Bit หน้า /x Bit หน้าเปลี่ยน
Broadcast = Bit หน้า /x Bit ไม่เปลี่ยน
Subnet = /x bit หน้า = 1 ที่เหลือ = 0

Cisco Ios (Internetwork Operating System)

- function ① Addressing ② Interface ③ Routing ④ Managing Resource ⑤ Security ⑥ QoS
- Router & Switch Boot Sequence (เปิด Router)

ROM 1. POST (เช็คว่า HW พร้อมทำงาน)
2. Run boot loader software
ROM 3 Boot loader does low-level CPU initialization
flash/ 4. Boot loader initializes the flash filesystem
TFTP Server 5. Boot loader locates & load a default IOS มา Run ใน RAM

④ Router# show running-config ; config
#startup-config ; memory ไว้ใช้บ่อย
#ip route ; route table ใน router
#show interface } show information
#show ip interface } interface ต่างๆ
#show ip interface brief ; show interface แบบสรุป ที่ set อยู่ใช้บ่อย
#ping ; response
↳ reply
↳ unreachable
↳ timeout

Accessing a Cisco IOS Device.

① Console port ② telnet ⑤ Secure Shell (SSH) ④ Aux Port
↳ Terminal Emulation Program :: PuTTY, Tera Term, SecureCRT, HyperTerminal, OS x Terminal

Navigating the IOS → 2 mode :: ① user ">" (guess) ② privileged (enable) (admin) "#"

↳ (2.1) Global Configuration Mode "(config)#"
↳ (2.2) Other " —— → "(config-mode)#"

⑤ #Copy running-config startup-config (ย่อว่า cp)

Command structure

switch> ping 192.168.0.12
prompt command space keyword or Argument

Getting Basic

① (config)#hostname name
② 2.1 Banner (config)# banner motd # #
2.2 PW (config)# enable password
2.3 Secret (config)# enable secret
⑤ (config)# interface
type port
type slot/port
type slot/sublot/port
vlan number → สั่งงาน switch
(config-if)# ip address ip_address subnet_mask ← set IP

① • Host names : ตั้งชื่ออุปกรณ์
② • limiting Access to Device config จำกัดการเข้าถึง user, pass
③ • Addressing Devices : กำหนด interface addressing
④ • Verifying Connectivity : ตรวจสอบการ config / connect
⑤ • Saving Config : มีใหม่ควร Save

fast internet, Gigs internet, serial internet } interface config
ต่อพอร์ต # no shut down ↳ เปิด port ขึ้นใช้ได้

Hot Keys

Tab : Complete Command
Ctrl+R : Redisplays line
Ctrl+A : Move Cursor to Begining line
Ctrl+Z : Exit config mode and return to user
↓ ↑ : command history
Ctrl+Shift+6 : Allow user interupt IOS process
Ctrl+C : ออก process ที่ทำอยู่

for Staples

B2H1 E2

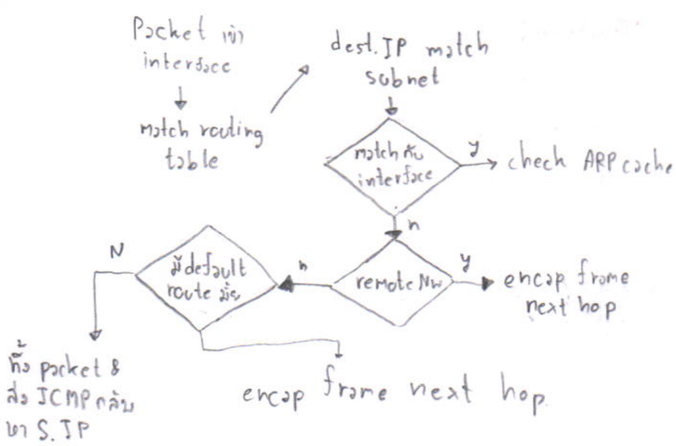
function of router ① Topology ② speed ③ cost ④ Security ⑤ Availability ⑥ Scalability ⑦ Reliability
↳ การกำหนด + ค้นหาเส้นทาง
↳ ป้ายบอกทางใน router แต่ละตัว , อาศัย routing table

Packets Forwarding Methods
① **Process switching** = ทุก packet ที่วิ่งเข้า router ต้องไป process ที่ cpu เพื่อเลือก interface ออก
② **Fast switching** - มีตัวช่วย ทำให้ forward ได้เร็วขึ้น
③ **Cisco Express Forwarding (CEF)** = forward packet ได้เร็วสุด

Connect Devices
- Default gateway
  - ↗ ทางออกเชื่อมไป ทพ อื่น
  - → เป็นได้ ① first usable host (.1) ② last usable host (.254)
- Enable IP on a Host
  - → ① statically Assigned IP addr.
  - → ② Dynamically ⟶ => กำหนด IP ให้ เมื่อ NW มี DHCP (Dynamic host config protocol)

Switching Packet Between NW
ขั้นตอน: นำ dest.(L3) ip เทียบใน routing table → นำทางออกไป เทียบหา MAC address → ใส่ dest MAC (L2)

Path Determination



- **Best Path** : lowest Metric (cost)
  => Dynamic routing protocol ได้
  ① routing Information Protocol (RIP) = จน. hop
  ② Open Shortest Path First (OSPF) = BW มากสุด
  ③ Enhanced Interior Gateway Routing Protocol (EIGRP)
  = BW, delay, load, reliability
- **Load balancing** = กรณีมี > 1 เส้นทาง
  ↳ ใช้ทุกเส้นทางพอๆกัน
- **Administrative Distance (AD)** : trustworthiness (คู่ควรแก่การไว้วางใจ)
  ↳ ระยะห่างจากผู้ดูแล ถึง Protocols ต่างๆ
  ↳ ช่วยตัดสินใจว่าเลือกตัวไหน ทางไหน
  ↳ ยิ่งใกล้ยิ่งน่าเชื่อถือ

connected = 0
static = 1
Internal EIGRP = 90
OSPF = 110
RIP = 120

Routing Table

D 10.1.1.0/24 [90/2170112] via 209.165.200.226 00:00:05 Serial0/0/0

- D ↳ ที่มา (มาจากไหน) C : directly connect, B = BGP, D = EIGRP, S = static
- 10.1.1.0/24 ↳ dest. NW ที่รู้จัก
- [90/2170112] ↳ Metric to reach the remote NW ; ↳ AD of route source
- via 209.165.200.226 ↳ บอกว่าจะไป ทพ นี้ต้อง next hop ไหน
- 00:00:05 ↳ ระยะเวลาที่ nw discovered
- Serial0/0/0 ↳ ถ้าจะไป ทพ นี้ต้องออกทางนี้ บอกทางออก interface

Routing

① **static Routing** => Manual
ข้อดี : security , resource น้อยใน process
ลด routing entry
ข้อเสีย : ลำบากการ scalibility , สามารถเปลี่ยนเส้นทาง
เหมาะกับ : ทพ ขนาดเล็ก , router น้อย , ทพ มีทางออกทางเดียว (stub NW)
4 type : ① standard ② Default = ใช้เมื่อ dest. ip ไม่ match
③ summary ④ Floating = backup

*Router(config)# ip route NW-addr subnet-mask {ip-addr | exit-intf}

Config : Next-hop option
R(config)# interface G0/0
R(config-if)# ip address nw-addr subnet } basic set ทำทุก itf
R(config-if)# no shutdown
R1(config)# ip route nw-addr subnet ⟶ ที่ router S0/0/0 / 172.16.0.0

Set default static route *

② **Dynamic Routing Protocol** → auto
2.1 EGP (Exterior Gateway Routing Protocol) : BGP
2.2 IGP (Interior Gateway Routing Protocol) : RIP, OSPF, EIGRP, IS-IS

Classful Addressing → update ตาม class
Classless Inter-Domain Routing
↳ **summarization** เงื่อนไข : ① อยู่ติดกัน ② ไปทางเดียวกัน
ถ้า set ใหม่ ต้อง no ip เก่าก่อน วิธีทำ ① แปลง ip → ฐาน 2 ② Group คล้ายกัน

VLSM → **Fixed Length Subnet Masking** ① Prefix ใหม่ - Prefix เก่า = จน. bit ที่เลื่อน
② เปลี่ยนทุก bit เป็น 1 แล้วนำค่าฐาน 10 ไป + ip เริ่มลบด้วย 1 ก่อนไปเรื่อยๆ

ชื่อ-สกุล น.ส. สุวมร วลัรัตน์ กระดาษแผ่นที่ 1 รหัสนักศึกษา 88888888

for Staples

Dynamic Routing Protocol

- share info ระหว่าง router, auto update when topology เปลี่ยน (หา best path)
- หา remote nw, ปรับปรุง router info, หา best path
- component ① Algorithm
  ② Routing protocol msg : หา neighbor & แลกเปลี่ยน routing info (best path)

	Dynamic	Static
ความยากใน ก. config	• independent nw size	• ∝ ขนาด nw
Required knowledge	• Advanced (config ทุกในจะซับซ้อน)	• None (route manual)
Topology change	• ปรับ Auto	• admin config All
Scaling	• เหมาะ simple & complex	• เหมาะ simple topologies
Security	• น้อยกว่า	• มากกว่า
Resource usage	• ใช้ cpu, mem(เก็บ routing info), link BW	• None
Predictability	• Route ∝ current topology	• Route → dest. ทางเดิม

Classifying Routing Protocols

- classful → update ตาม class ไม่ส่ง subnet mask ใน routing update
- classless → ส่ง subnet mask ใน routing update

DRP
- Exterior Gateway Protocol (EGP)
  - BGP (Border Gateway Protocol) => Autonomous System (AS) เป็นกลุ่มของ Router ภายใต้ ก.ควบคุมของ single authority (1 กลุ่ม policy or ผู้ดูแล)
- Interior Gateway Protocol (IGP)
  - Link-state P. => complete nw topology (เก็บข้อมูลทั้งหมด) => update ไม่ใช่ periodic — OSPF (Open Shortest Path First), IS-IS; (IS = intermediate System)
  - Distance Vector P. => ประกาศ Vector [distance, direction] => incomplete view of nw topology => periodic update (รายคาบ) (ซ้ำๆ)
    - RIP (Routing Info P.) — RIP v.2
    - IGRP (Interior Gateway Routing P.) → EIGRP (Enchanced Interior Gateway Routing P.)

• **Convergence** : ถูกทำเมื่อ routing table ของ all router มีสถานะคงที่
  - 2 type — slower : RIP & IGRP
  - faster (มัก ก. update เมื่อเกิดการเปลี่ยนแปลงด้วย) : EIGP & OSPF

Routing Protocol Metrics

- **Metric** : ค่าที่ใช้ในการคิด/หาทางในการไป dest. NW จะเลือก best path ไหน เช่น hop count, BW, Cost, Delay, Load, Reliability
- **Load balancing** : nw มี > 1 เส้นทาง ที่มี metric เท่ากัน → เฉลี่ยในการใช้งานแต่ละเส้นทางเท่าๆ กัน

Administrative Distance of a Router (AD) => ใช้เลือก protocol ในก.routing (เจาะจงส่งข้อมูลเฉพาะส่วน)

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

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

- Distance Vector Technology ต้องรู้ 2 ส่วน ① vector or direction, ก.แลกเปลี่ยนความถูกควบคุม ② Distance to final dest. (cost)
- ช่วงเวลาในการ update ตามคาบเวลาแน่นอน, neighbor (ข้างๆ), broadcast (255.255.255.255) update, เอา all routing table ไป update
- เกณฑ์ที่ใช้ check ว่า DV ดีหรือไม่ ① Time to convergence => เวลาที่เป็น steady state ของ routing table ที่เปลี่ยนแปลงแล้ว ② Scalability ขยายมากสุด ③ Resource usage ④ Implementation & maintenance

NW Discovery (ทำ basic config ก่อน)

① cold State : Router Initial Start up
② Initial Exchange of Routing Info. → ทำการแลกเปลี่ยน
③ Exchange of Routing info. → update (เพิ่ม hop count) routing info → บอก router อื่นต่อไปเรื่อยๆ

for Staples

Routing Table Maintenance

- Periodic update : RIP update timer (default 30s), Invalid timer (Info เริ่ม lost) (default 180), Hold down timer (ถ้า down → hold ไว้ ไม่ว่า up หรือไม่) (default 180), flush (ทิ้ง) timer (default 240)
- Bounded (ขอบเขต) Update : EIGRP → update แค่ที่เปลี่ยน
- Triggered Update → update เลยไม่รอ periodic time
- Random Jitter → ใช้ใน nw ที่เป็น multiple access router หลายตัวเชื่อมต่อกัน → ถ้า update มาชนกันตรงนี้ ใช้ random

ปัญหาที่เกิดจาก DV

	RIP v.1	RIP v.2	IGRP	EIGRP
• speed convergence	slow	slow	slow	fast
• Scalability size nw	small	small	small	Large
• Use of VLSM	×	✓	×	✓
• Resource usage	Low	Low	Low	Medium
• Implementation & maintenance	Simple	Simple	Simple	Complex

→ ① **Routing Loops** เกิดเมื่อ interface ในนั้น down จะถูกตัดออกจาก table → ค่าเก่าใน neighbor จะกลับมา update (วน update → hop เพิ่มขึ้นเรื่อยๆ)

solve
① **set max hop** ถ้า hop มากกว่า max => down ไปแล้ว
② **holddown timer** ถ้า interface down → hold
③ **split Horizon Rule** → ไม่ส่งข้อมูล update กลับไปทาง interface ที่ได้รับ update มา
④ **Route Poisioning** → ① เกิด down set unreach ② ส่ง unreachable ที่เป็น poison ออกไป
⑤ ③ + ④ → ถ้าเกิด unreachable จะ Over rule split horizon โดยส่ง ip intf ที่ down
⑥ **IP & TTL** (time to live) เกิดการวน update but จะหยุดเมื่อ TTL = 0

RIP v.1 AD = 120

↳ เป็น classful, DV = metric = hop count = hop count > 15 unreachable = update broadcast ทุก 30 s
encapsulated (บรรจุ) ใน UDP segment ด้วย source and dest.
Transport Layer — Port = 520

Data link frame Header	IP Packet Header	UDP Segment Header	RIP Message (512 Bytes; Up to 25 routes)

header ←
1 = Request, 2 = Reply

0–7	8–15	16–23	24–31
Command = 1 or 2	Version = 1	Must be Zero	
Address family ID (2 = IP)		Must be Zero	
IP Address (NW Address) → dest.			
Must be Zero			
Must be Zero			
Metric (Hops) → 1-15 ตัวจาก table + 1 ส่งออก			
Multiple Route Entries, up to max of 25			

→ full routing table

• **msg มี 2 type**
① Request → ส่ง routing table
→ ไป intf ที่ config ไว้เพื่อรอรับการ update
② Response → ส่ง info ของ routing table

• **IP addr.** แบ่งตาม class A, B, C

• **Basic RIP v.1 Config**
① ทำ basic config
② สั่ง router rip + คำสั่ง nw
R1(config)# router rip
R1(config-router)# network nw ip ที่ต้องรู้จัก

• **verification & troubleshooting** : show running-config | ip route | ip protocols | debug ip rip
↳ passive intf command (ไม่ update intf ที่ไม่ต้องการ) R(config-router)# passive-interface intf-type (Fa0/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 แสดงผลในการแทน multiple route เยอะมากใน 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 another

172.30.0.0
172.30.2.0 ← boundary
172.30.1.0

• **Processing RIP update** ดูตัวที่ส่ง & ตัวที่ update ไป (intf) ว่ามี classful เดียวกันไหม → Y : update subnet nw เช่น 172.16.1.0
↳ N : update classful เช่น 172.16.0.0

• **default route & RIP v.1** ถ้าไม่เจอทางออกใน routing table (อาจจะ bcc. คนละ protocol) → จะออกมา 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
R(config-router)# default-information originate ← router ที่อยู่ระหว่าง 2 protocol

RIP v.1	RIP v.2
• classful (ไม่ส่ง subnet mask, ไม่ support CIDR)	• classless (update subnet mask, support Variable Length Subnet masking, support Route Sum (Prefix Aggregation))
• not support discontigous subnet	• update next hop addr
• not support VLSM bcc ไม่ส่ง subnet mask	• ใช้ authentication routing รองรับ discontiguous บางกรณี
• 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

• **ข้อจำกัด**

เป็น virtual interface สามารถเพิ่มใน routing ไม่ update ออกไปได้ {
• loopback intf → ping ไป ip virtual intf → reply ไม่
• Null intf → ใช้ตัดคนที่จะเข้ามา channel ที่เราไม่ต้องการ → ส่งออก null intf → packet discard เอง → time out
• Static route & null intf → null intf จะรับหน้าที่เป็นทางออกของ static route
R(config)# ip route summary-static-route supnet-mask Null0
(major-nw) → ของ static supernet route
• Route redistribution => อยากให้ rip รู้จัก static ต้องทำเพิ่มไปยัง rip ส่ง static ไปด้วย R(config-router)# redistribute static
• Verify & Test Connectivity : show ip interface brief, ping, trace route
• RIP v.1 : classful, ไม่ส่ง subnet mask, Summarize nw ที่ major nw boundaries, if nw เป็น discontiguous & RIP v.1 config convergence จะมาไม่ถึง
• ทดสอบ routing table : debug ip rip (content of routing update), ถ้าเป็น RIP v.1 จะไม่มี subnet mask รวมใน nw addr

ชื่อ-สกุล น.ส. สุวมร วลีรัตน์ กระดาษแผ่นที่ 3 รหัสนักศึกษา

for Staples

0	7/8	15/16	23/24 ~ 31
Command = 1 or 2	Version = 2	Must be zero	
Address Family id (2 = IP)		Route Tag	
IP Address (NW Addr.)			
SUBNET MASK			
NEXT HOP			
Metric (Hops)			
Multiple Route Entries, up to a max of 25			

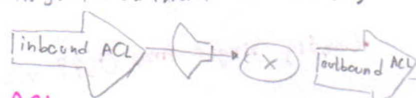
RIP v.2

- **Config**
  - Enabling & verify (ตรวจสอบ) RIP v.2 → show ip protocols
  - Config RIP → RIP v.1 → สามารถรับได้ทั้ง v_1 & v.2 but ส่งได้แค่ 1
    - → RIP v.2 → สามารถรับ & ส่งได้แค่ v.2
  - Auto-Summary & RIP v.2 → auto sum route ที่ major nw boundary.
    - → sum route ด้วย subnet mask ที่น้อยกว่า classful subnet mask
  - disabling Auto-summary : no auto-summary bec. when อัน nw topology ต้องเป็น discontiguous
- **VLSM & CIDR** → verify info ที่ sent by RIP v.2 debug ip rip
  - → VLSM → เผยแพร่ nw addr & Subnet mask
  - → CIDR → ใช้ supernetting (= bunch ของ contiguous classful nw ที่เป็น addr เพียง single nw)
    - → verify show ip route, debug ip rip

Access Control List

= คอยควบคุมการเข้าออก → ตรวจสอบ → check → Source → dest ผ่านไหม? / → พอร์ต (เช่น FTP) ผ่านไหม?
= ดูจาก Conversation

- **Packet Filtering** ดู ① dest, source ที่ L2 ② protocol อะไร ③ ไปทางไหน, อะไร ใช้ตรวจว่าให้ผ่าน or block ไว้
- **Operation** → ทำงานเป็น sequence statement
  - → last statement เป็น implicit deny → block → discard



- **Standard IPv4 ACLs** vs **Extendend IPv4 ACLs**

Standard IPv4 ACLs	Extendend IPv4 ACLs
- check source addr,	- check source & dest. addr.
- อนุญาต permits or deny ทั้งหมด protocols	- อนุญาต permits or denies specific protocol
access-list 10 permit 192.168.30.0 0.0.0.255	access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
- number ACL : 1-99 & 1300-1999	- Number ACL 100-199 & 2000-2699

- **Wild card** → invert ของ subnet mask
  - → 0 = match/fix, 1 = ignore / อะไรก็ได้
  - → ใช้บอก set ของ ip (match range) ① นำความเหมือนของ bit ที่ซ้ำกัน ใส่ wild card mask ตำแหน่งนั้น = 0
    - ② bit ที่เหลือใส่ 1
    - if กรณีค่อนไปได้ออกนอก pattern or/and ล้นต่อ สุดท้าย wild card จะ same กัน.
  - → หา wild card ของ subnet = 255.255.255.255 − subnet mask
  - → keyword → 0.0.0.0 = match all ใช้ host
    - → 255.255.255.255 = ignore all ใช้ any
- **Guideline for (3Ps)** → One ACL / protocol = ctrl traffic flow บน intf, ACL ต้อง define แต่ละ protocol enable on intf
- **ACL creation** → 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 : at close source → standard ACL : at close destination
- **Config ACLs** → 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

  เลือกมา 1 แบบ ตำแหน่งในพรอมต์ แต่ต้องสามารถลบได้ ด้วย (permit, deny, remark) ได้

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

debug ip packet ACL-num

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

access-list access-list-number {deny | permit | remark} protocol source [wild] [operation operand] [port port-number or name] [dest] [dest.wild] [operator operand] [port port-number or name] [Established]

for Staples

for Staples

รู้ info all

Link-State Routing Protocol

: เป็น Protocol ที่ต้องสร้าง complete map ของ topology ขึ้นก่อน → เอา shortest path first (SPF) → ดูว่า

- ↳ เหมาะกับ ① Large NW ② Fast Convergence ③ admin ควรมีความรู้ดี
- ↳ ขั้นตอนการทำงาน ① learn info ของ link ② Say hello neighbor ③ เอา info มาสร้าง Link-state Packet (LSP) ④ router flood LSP to all neighbor → ได้เก็บลงฐาน db ⑤ router เอา all LSP เก็บใน db (tree) + adding OSPF ลง route table
- ↳ ข้อดี ① สร้าง topology map ซึ่งเอา shortest path ② fast convergence ข้อมูลที่ change ③ LSP send only when change topology คำนวณ shortest. ④ hierarchical design (แบ่ง NW เป็นส่วนๆ) → ลด resource bec. แลกเปลี่ยนข้อมูล แค่ในarea
- ↳ ข้อเสีย : ① ใช้ mem ในการเก็บ all link-state เยอะ ② ใช้ CPU ในการคำนวณ ③ ตอนส่ง LSP ตอนๆ ใช้ BW มาก

• OSPF AD < 110

↳ 3 table : ① Neighbor show ip ospf neighbor ② topology (ทำ map) show ip ospf database ③ Routing (เก็บ shortest path)

• msg → 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 print nw), ทุก 30s (default: non-broadcast multiaccess [NBMA] NW), Cisco default 4 times (40s)
- : 02 Db Description (DBD) → Synchronize ข้อมูล db info.
- : 03 Link-state Request (LSR) → request link-state
- : 04 Link-state Update (LSU) → Send update link-state
- : 05 Link-state Acknowledgment (LS Ack) → ตอบกลับว่าได้แล้ว

• Operation สร้างความสัมพันธ์ ① Down state (เริ่มต้น) → ② init state (เริ่มส่ง hello) → ③ Two-way state (ตอบรับ hello) → ExStart State → Exchange state → Loading state → Full state (ระหว่าง router update ข้อมูลครบ)

• Config Single-Area OSPF v.2 router ospf process-id → 1-65,535, เป็น locally significant

R(config-router)# router-id 1.1.1.1 → ถ้าไม่ set จะใช้ loopback, active interface ip สูงสุด but ควรมี ① ไม่งั้น ②

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 Gb Ethernet = 100×10^8 → Cost = 1
- → Gb " = 10×10^8 → " = 1
- → Fast " = 10^8 → " = 1
- → Serial = $1{,}544 \times 10^6$ → " = 64

ต้องการเปลี่ยนค่า cost
- ↳ เปลี่ยนค่า ref BW
- ↳ เปลี่ยนค่า BW
  R(config-if)# bandwidth 64
- ↳ เปลี่ยนค่า cost.
  R(config-if)# ip ospf cost 15625

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

more config
```
#ip route 0.0.0.0 0.0.0.0 loopback N
#router ospf process-id
#default-information orginate
```

DHCP (Dynamic Host Configuration Protocol) → ช่วย config ให้ host ให้ auto

method
- ① Manual Allocation : admin assign เอง
- ② Automatic Allocation : DHCPv4 auto assign addr. จาก pool & ไม่มี lease(เช่า) time
- ③ Dynamic Allocation : คล้ายกับ ตัวแรก ต่อจะได้ ip ไปตาม lease time → ถ้าหมด lease time ต้อง re ip ใหม่
  [illegible]

Config
```
R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9      ← 1-9 จะไม่ถูกแจก
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1      ← สร้าง pool
R1(dhcp-config)# network 192.168.10.0 255.255.255.0      ← nw ip ที่เราจะแจก
R1(dhcp-config)# default-router 192.168.10.1      * ต้องรู้
R1(dhcp-config)# dns-server 192.168.11.5      } ไม่รู้ก็ได้
R1(dhcp-config)# domain-name example.com      }
R1(dhcp-config)# end
```

• To disable dhcp - no service dhcp ยกเลิก
  ถ้า ip ไม่ขึ้น → cmd → ip-config/release → ip-config/renew
  → คลิกสลับกัน (@ หน้า set ip)

Verify
- show running-config | section dhcp
- show ip dhcp binding
- show ip dhcp server statistics

debug
same Extended

On the PC - issue the ipconfig/all command

config DHCP client (แจก ip ให้ client) : -if)# ip address dhcp
-if) no shutdown

ชื่อ-สกุล น.ส. สุวมร วดีรัตน์ กระดาษแผ่นที่ 4 รหัสนักศึกษา 88888888

for Staples

Chap 7 Basic Switch Address Resolution Protocol

LAN Design → Borderless SW NW design มักทั่วไป : ① Hierarchical, ② Modularity, ③ Resiliency, ④ Flexibility

- **2 ลักษณะ** : ① 3-Tier : 1) Core, 2) Distribution, 3) Access ② 2-Tier 1) Collapsed Core/Distribution 2) Access

switch {
- **Core** => อุปกรณ์ที่รองรับ SW สูง → ทำให้ speed NW สูง
- **Distribution** => เชื่อมระหว่าง Core & Acc, Security Policy / Access Ctrl
- **Access** Access => ต่อกับ end device, Port Security, VLAN, (Fa/Gig Eth), Power over Ethernet => รองรับ 1 LAN ไม่ต้องเดิน 1 LAN 1 Power Line

} layer 3 Support, [Gig/10 Gig Eth], Redundant components => มีแหล่งจ่าย

} Link aggregation → ทำให้ BW device ↑, Quality of Service (QoS)

- **เพิ่มความสามารถในการใช้ LAN BW และประสิทธิภาพให้ Max**
  - f- [redacted] จัดหา Server — Enterprise S. (รวมทั่วองค์กร) => ติดตั้ง at MDF (Main Distribution Facility - Core) => ทางผ่านไปข้างนอก ที่ใกล้ / Workshop S (เฉพาะกลุ่ม) => IDF (Intermediate D. F. : Distribution)
  - **Collision detection issue**
  - **Segmentation issue** => ทำให้การใช้งาน NW มีประสิทธิภาพดีขึ้น
  - **Broadcast domain issue** => ใน NW เกิดตลอด เช่น broadcast หา MAC Addr. ∴ broadcast NW ควรมีขนาดเล็ก

VCC (vertical cross-connect) : optical fibre => MDF ↔ IDF
MCC (Horizontal ~) : UTP => Distribution ↔ Access

- **Segmentation** เป็น process split single collision domain → smaller collision ลดจำนวน collision บน LAN segment :: L#2 device ใช้ เช่น bridges, SW
- **Broadcast domain** รับหมดทุก port but. router (L#3) เป็นตัว filter/segment broadcast ไม่ให้กระจายออกจากตัวมัน

SW Environment

- **SW Operation** : ① learning : รับ frame เข้า SW จะเรียน Source MAC Addr. ว่าต่อกับ port ไหน → reset Aging
  ② Aging : อายุของ MAC Addr. → if หมด → ทิ้ง

  การส่ง {
  ③ Flooding : ส่ง frame ออกทุก port ของ SW when frame เป็น 1) broadcast, 2) multicast, 3) unknown unicast → กรณี dest. ไม่อยู่ใน table
  ④ Forwarding : ส่งไป dest. (กรณีมีใน table)
  ⑤ Filtering :: if ได้รับ frame ไป dest. จาก port ที่เป็นทางไป dest. (source & dest บน same interface) จะทำการ filter ทิ้ง
  }
- **SW Methods** : ① Store & Forward SW => check CRC ว่า error ไหม → มี : ทิ้ง → ไม่มี : ส่ง, auto buffer
  ② Cut-Through SW => check เฉพาะบางส่วน (dest., source อย่างน้อย 12 byte แรก) [10ns], No FCS & auto buffer
  มี 2 mode :: [1] fast-forward ~ 12 byte [2] Fragment-free ~ 64 byte แรก :: (64 = ขนาด → หลุด → ไม่ส่ง)

for Staples

- **SW Domains** : ① Collision Domains => domain ที่เกิดการชน ตอนที่ส่งข้อมูลมา พร้อมกัน "@@ SW เป็นตัวแบ่ง"
  ② Broadcast ~ => domain ที่ส่ง broadcast → อยู่ domain เดียวกันจะได้รับ "@@ router เป็นตัวแบ่ง"

Basic SW Concept & Configuration

- **Basic SW Config**
  - **SW Boot Sequence** = Same Router
  - **Preparing of Basic SW Management** : SW ไม่มี loopback ∴ กำหนดเป็น SVI (SW Virtual interface) → VLAN
    manage intf : S(config)# interface vlan num default Gateway
    same router { S(config-if)# ip address ip subnet S(config)# ip default-gateway ip ; S(config-if)# no shutdown }
  - **Config SW Port → Duplex Communication** : ① Full ② Half (SW ที่ต่อกันต้องเลือก same กัน)
    เข้าไป intf → S(config-if)# duplex full → S(config-if)# speed 100 (ปรับจำนวน speed)
    → **Auto-MDIX** : ถ้าต่อ SW จะเชื่อมกันด้วย cross-over but. ต่อตรง ปกติจะเป็นสายตรง
    เข้า Intf → # duplex auto → # speed auto → mdix auto

- **Verify Port Config**
  show int fa0/0 / flash / version / history / ip fa0/0 / mac-address-table / startup-config / running-config

- **SW Security** : Security Remote Access → SSH (Secure Shell) TCP port 22, Telnet : TCP port 23
  Config : (config)# ip domain-name ชื่อ → # crypto key generate rsa → # username admin pass (ค่า) → line vty 0 15 → # → # login local [Verify SSH : show ip ssh, show ssh]

- **SW Port Security** => กำหนด policy ว่าให้ MAC Addr. ไหนเข้า/ออกได้-ไม่ได้
  (config-if) # switchport mode access → # switchport port-security → เลือกมา 1 ประเภท
  Secure MAC Addr. → ① Static :: S(config-if)# switchport port-security mac-address MAC-ADD
  ② dynamic :: (config-if)# switchport port-security mac-address sticky → learn frame ที่วิ่งมาก่อน → record ไว้
  กรณีเพิ่ม จน. MAC : # switchport port-security maximum MAX
  violation mode : ① protect : security violation protect mode => ส่งไม่ได้ แต่ว่ามีอะไรเข้ามาบ้าง
  ② restrict : security violation restrict mode
  ③ shutdown : Security violation shutdown mode => default
  (Verify : show port-security int fa0/0, show port-security address)

- **Addr. Resolution Protocol (ARP)** : ARP Cache เก็บ MAC Addr. ไว้ map แทนไป dest. (if ไปข้างนอกต้องใช้ MAC gateway)

IPv4 : Classless [ต่อ P.1-2] : - Variable Length Subnet Masking (VLSM) : แบ่งทุก NW ไม่เท่ากัน => เลือกของใหญ่ → เล็ก
- Fixed ~ : แบ่งทุก NW เท่ากัน

for Staples

Chap 8 LAN Redundancy & Spanning Tree Protocol (STP)

Issue with Layer 2 Redundancy : ① **MAC Addr. instability** => MAC Addr. table จะไม่เสถียร bec เปลี่ยนแปลงบ่อย ② **Broadcast storms** แล้วไปทั่วไม่หยุด
③ **Multiple frame transmission** ↳ start : unknown unicast → ทำให้ dest ได้รับหลาย frame but source ส่ง 1 frame

STP => จุดเฉพาะ : หา block port → block วน → ไม่ให้เกิด traffic วน เพื่อแก้ปัญหา

> ส่วนที่ใช้เทียบ ① BPDU flag #Priority น้อยสุด
> ② BID น้อยสุด
> ③ Path Cost น้อยสุด
> ④ Sender's BID น้อยสุด
> ⑤ Sender's Port น้อยสุด

- **ขั้นตอน** ① หา Root Bridge = เลข Priority min **Rule : [1] 1 RB/1 NW [2] 1 RP/1 RB [3] 1 DP/segment**
  ดู BPDU (Bridge Protocol data unit) ↳ ได้ RB & Source [802.1D]
  ② หา path cost ทั้งหมด ③ หา Root Port → path cost min → อีกฝั่ง ของ เป็น Destinated port
  ④ ส่วน segment ที่ path cost เท่ากัน → ดู BID min เป็น destinated port → อีกฝั่งเป็น block port
- **Config** : วิธีที่ 1 : S1(config)#spanning-tree VLAN 1 root primary วิธีที่ 2 : S3(config)#spanning-tree VLAN 1 priority 24576 ← มากๆ
  (if S1 & S2 เป็น RB) S2(config)#spanning-tree VLAN 1 root secondary (verify : show spanning-tree)
- **แบบ Extended System ID** : B Priority (2 byte) => B Priority (4 bit) (per VLAN + Extended Sys ID (VLAN) (12 bit) + MAC Addr. (6 bit) ∴ BID = 8 bit
  - PVST+ (พัฒนามาจาก IEEE 802.1D STP) => เพื่อทำ Load balancing โดยแยก root / VLAN (Verify : show spanning-tree active)
  - Rapid PVST+ → หา Alternate port (ที่ไม่ใช่ block ไปเลย สามารถรับส่งข้อมูลได้บางส่วน)
    ↳ ทำการ set Edge Port @ port host, router : -if)#spanning-tree port fast
    ↳ link type : port ที่เชื่อมระหว่าง sw ต้องเป็น point to point : -if)#spanning-tree bpduguard enable → port ปิดก่อนถ้าได้ BPDU ไปต่อ
    ↳ config : (config)#spanning-tree mode rapid-pvst → #int ที่ p-to-p → #spanning-tree link-type point-to-point
    ถ้าจะ clear all : clear spanning-tree detected-protocol

Chap 9 VLANs & Inter VLAN

VLAN => ทำ partition (แบ่งแยก NW / Broadcast Domain) Layer 2 เช่น SW และ host ที่ต่ออยู่ได้แค่ใน VLAN เดียวกัน

- **ข้อดี** · security ↑ · cost ↓ · ประสิทธิภาพ ↑ · broadcast domain ↓ **Verify :** show vlan brief
- **In a Multi-SW Environment**
  - **VLAN Trunk** : set ที่ intf ที่เชื่อมระหว่าง SW ที่ VLAN → เก็บข้อมูลได้มากกว่า 1 VLAN
    ↳ Config : เข้า intf → if)#switchport mode trunk [Verify : show int f0/0 switchport]
  - **Tagging Ethernet Frames** (IEEE 802.1Q) : Ethernet frame → Dest MAC | Source MAC | Tag | Type/length | Data | FCS

  > Tag ทำให้ส่งออก VLAN ถ้าส่งผ่าน Trunk
  > ↳ Native VLAN ไม่ต้องทำ tag / เอา tag เปิดก่อน tag

- **Assignment : VLAN number** => 1-1005 เก็บ config at vlan.dat (ใน flash) => 1006-4096 เก็บ Config at running-config (ใน NVRAM)

  > สร้าง (config)# vlan vlan_num → # name ชื่อ
  > or # vlan database → # vlan num name

  - assign port ให้ vlan : เข้า intf → #switchport mode access → # switchport access vlan vlan_num

  > Verify
  > - show vlan name ชื่อ
  > - show vlan summary
  > - show int vlan num

Inter-VLAN Routing => router set เป็น trunk ให้ใช้ sub intf **verify :** show vlan, show ip route, show run

- **config** : ① set basic routing (set ip address, no shutdown)
  ② R(config)#int g0/0.10 → -subif)# encapsulation dot1q

Chap 10 VTP (VLAN Trunking Protocol) → จัดการ VLAN & NAT (NW Addr. Translation)

VTP [msg : ISL or IEEE 802.1Q] → การจัดการ SW VTP เป็นแบบ จัดกลุ่ม domain

- **operation :** การ update VTP จะใช้เลข revision number 32 bit (0-4294967295) เลือกค่าสูง
  การ สร้าง, remove, rename VLAN ทุกตัวใน domain ยกเว้น ③
  ↳ 3 mode : ① **Server** → สามารถ สร้าง, remove, rename VLAN, process, ส่ง VTP msg ออกทุก trunk
  ② **Client** → รับข้อมูล VTP มา process, ส่งต่อผ่าน
  ③ **Transparent** → สามารถสร้าง, remove, rename ในตัวเอง, ส่งต่อผ่าน
- **Config 2 แบบ** สิ่งสำคัญ 1) SW cisco 2) มี trunk เชื่อมระหว่าง SW 3) ต้อง domain 4) มี 3 mode
  ① In global conf : (config)# vtp version 2 => # vtp domain ชื่อ → # vtp password pw → # vtp mode server/clt → # vtp server/client/tran
  ② In VLAN conf : (vlan)# vtp v2-mode **verify** : show vtp status/counters
- **Pruning** → manage ให้ traffic ทำงานในintf น้อย โดย config ที่ intf ต่างๆ เพื่อ remove ข้อมูลที่ไม่ใช้
  S(vlan)#vtp pruning → เข้า intf → -if)#switchport trunk pruning vlan remove number-of-vlan

NAT = แปลง private ip ↔ public/real ip **verify :** show ip nat translation

- **terminology** : 4 type ① Inside local Addr. (private ip) ② Outside local Addr.
  ③ Inside global addr. ④ Outside global Addr. ↳ same เดิม
- **type** : ① **Static** : กำหนดค่าเอง [map : 1 → 1] → (config)# ip nat inside source static local-ip global-ip
  ② **Dynamic** : มี pool ของ Global/Real ip (map : many ↔ 1) & real ไม่พอใช้ร่วมกัน
  ③ **PAT** (Port Addr. Translation) → port ที่อยู่ร่วมกัน NW Addr (map : many ↔ 1)
- **Config.** ① NAT (1.1) (config)# ip nat pool ชื่อ start-ip end-ip {netmask netmask | prefix-length}
  (1.2) set ACL
  (1.3) ip nat inside source list ACL-num pool ชื่อ overload → กรณีเป็น PAT
  ② Inside : -if)# ip nat inside
  ③ outside -if)# ip nat outside
  or PAT config (1.1) สร้าง ACL
  (1.2) ip nat inside source list ACL-num interface f0/0 overload.

ชื่อ-สกุล น.ส. สุพร วดีรัตน์ กระดาษแผ่นที่ 5 รหัสนักศึกษา 59011458

for Staples

Chapter 11 EIGRP IPv6 & Routing.

Dynamic Routing Protocol
- Interior Gateway P.
  - Distance Vector Routing P.
  - Link State Routing P.
- Exterior Gateway P.
  - Path Vector

	Distance Vector Routing P.		Link State Routing P.		Path Vector
Classful	RIP	IGRP			EGP
Classless	RIP v.2	EIGRP	OSPF v2	IS-IS	BGP v.4
IPv6	RIP ng	EIGRP for IPv6	OSPF v3	IS-IS for IPv.6	BGP v.4 for IPv.6

▷EIGRP (Enchanced IGRP)

□Characteristics (ลักษณะประจำตัว)

- Basic features - Cisco-proprietary (ใช้งานเฉพาะ protocol ของ cisco) เปิดให้ใช้ ปี 1992
  - # สร้าง classless version of IGRP # ทางเลือกที่เหมาะสม สำหรับ nw หลาย Protocol, ขนาดใหญ่ ที่สร้างขึ้นบน cisco router เป็นหลัก
- * DUAL (Diffusing Update Algorithm) = การันตี loop-free & back up path ทั้งหมดของ routing domain → หา best path → ทำให้ routing ทำงาน very fast convergent (น้อยกว่า OSPF) รวมทั้งมี back up path (ทำงานเร็วขึ้น)
- * Establishing Neighbor = เชื่อมความสัมพันธ์กับ directly connected EIGRP routers
  Adjacencies · Adjacencies are used to track the status of these neighbors
- * Reliable Transport Protocol · RTP provides delivery of EIGRP packets to neighbors
  · RTP and neighbor adjacencies are used by DUAL (ใน ก. maintain)
- * Partiable Transport Protocol · update เฉพาะส่วนที่มีการเปลี่ยนแปลง รวมถึง update ไปให้เฉพาะส่วนที่มีการเปลี่ยนแปลง ∴ update < RIP
- * Equal and Unequal Cost = ช่วยให้ admin ดูแลระบบ หลีกเลี่ยงการรับข้อมูลในเครือข่ายได้ดีขึ้น
  Load Balancing ↳ค่า cost ไม่เท่ากัน แต่ทำเป็น load balance ได้
  - ใช้ protocol-dependent modules (PDMs) เพื่อรองรับ protocol ที่แตกต่างกัน เช่น IPv.4, IPv6, Legacy protocol IPX and AppleTalk
  - PDMs รับผิดชอบ:
    - # maintain EIGRP neighbor and topology table (Neighbor Table → สร้าง Topology Table → ใส่ routing table เป็น routing) (เก็บค่า tropology ← OSPF *; ผ่าน DUAL; เก็บ shortest path กับ backup shortest path; EIGRP เลือก successor.)
    - # 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
    ∘ เป็น msg ที่อยู่ใน application layer ใช้ maintain ข้อมูล, msg ต่าง ๆ ทั้งของ EIGRP
  - ไม่ใช่ทั้งหมดของ RTP packet จะส่งได้น่าเชื่อถือ (msg ≈ OSPF)
    - # Reliable packet require explicite (ชัดเจน) ack จาก dest. # Update, Query, Reply.
    - # Unreliable packet do not require ack จาก dest. # Hello, Ack
  - รองรับ authentication (no encrypt routing update) และ is recommend (เป็นที่ยอมรับ) (authen ≈ RIPv2 & OSPF) (protocol อยู่ใน transport layer)
- Packet Type routing update or queries EIGRP multicast IPv.4 : 224.0.0.10, IPv6 : FF02::A อันนี้ IGRP multicast 224.0.0.9 RIPv1 broadcast 255.255.255.255
  ① Hello → ทำให้ Adjacenies ระหว่าง router 2 ตัวที่เป็น neighbor กัน, ไม่ต้องตอบ resp., ส่ง unreliably
  ② Update → update info ของ dest, update info ของ routing ไปให้ neighbor router.
  ③ Acknowledgement → พอได้รับการ update ก็ตอบ Ack.
  ④ Query → request info routing จาก neighbor router } เวลาต้องการ. info ของ routing เพิ่มก็ query ไปกับ router ที่อยู่ข้าง แล้ว reply ตอบกลับว่ามี หรือ ไม่มี
  ⑤ Reply → หลังจากได้รับ query มี reply

□ Inplement EIGRP for IPv4

- Actonomous System (AS) is a collection of nw ภายใต้การดูแลของ single authority (อ้างอิง RFC 1930)
  ↳ AS number → ใช้ exchange routes between AS
  → managed by IANA & assigned by RIRs to ISPs, Internet backbone providers, and institution to other institution
  → 16 bit : 0-65535 ⇒ Since 2007, 32 bit : over 4 billion
- Configure : R(config)# router eigrp AS-#
  R(config-router)# eigrp router-id → ถ้าไม่กำหนด จะใช้ loopback intf
  R(config-router)# network nw-number [wildcard-mask] → มี/ไม่มีก็ได้ ↳ IPv.4 addr. ที่อยู่ภายใต้ active
  (มี/ไม่มีก็ได้) R(config-router)# passive-interface type number (defaults: ไม่ให้ update ไปที่ intf. ไหน (เช่น LAN, G, F) → not serial)
- verify — show ip eigrp neighbors, show ip protocols, show ip route
- ip protocol

□ Operation.

∘ Initial Route Discover (ค้นพบ) ① R1 ส่ง hello กับ neighbor router ② R2 อาจจะส่ง hello or update กลับมา
③ R1 ตอบ ack & update info. ④ ใช้ DUAL คำนวณ best route and update routing table

□ Metric : BW [Lowest] (k1=1) Delay [สะสม] (k3=1), Reliability [worst] (k4=0, k5=0), Load [worst] (k2=0) ดูค่า value = show interface

Default Composite Formula: metric = [k1 * bw + k3 * delay] * 256 complete: = $\left[k1 * bw + \frac{(k2 * bw)}{(256 - load)} + k3 * delay\right] * \left[\frac{k5}{reliability + k4}\right]$

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

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

□ DUAL and the Topology Table ใช้ FSM (Finite state machine) ในการเลือกเส้นทาง) → show ip eigrp topology [all-links], show ip route

+ Successor (S) [router ที่ไป dest. ดีที่สุด] = neighbor router ที่จะเลือกทางไป dest ที่มีค่า min ที่สุด.

+ Feasible Successor (FS) [ที่ได้ผ่าน Feasible condition] = Backup path (เส้นทางสำรอง)

+ Reported Distance (RD) [distance ที่ neighbor ก่อนถึง report distance มาให้เรา] = advertised distance = จาก neighbor → dest มี cost เท่าไร

+ Feasible Distance (FD) [distance ที่ไปถึง S] = ค่า distance ทั้งหมดที่ไป dest. ทั้งหมด เป็นค่า cost lowest dest | ของจาก hop

ชื่อ-สกุล น.ส. สุมง วดีรัตน์ กระดาษแผ่นที่ 6 รหัสนักศึกษา 59011458

for Staples

Summary Config

Router

① basic config

เข้า intf => R(config-if)# ip address (ip-address) 192.168.1.1 (subnet mask) 255.255.255.0 ⇨, # no shutdown

เข้า intf (serial ฝั่ง DCE) => R(config-if)# clock rate 56000

verify: show running-config → ดู config
show startup-config → ดู setting
show ip route. } → ดู routing table
show ip route.
show interface. } interface
show ip interface
show ip int brief show intf แบบย่อ

② protocol

- static routing

เข้า intf => R(config-if)# ip route nw-ip 192.168.1.1 subnet mask 255.255.255.0 {ip addr. | exit-intf} (next hop, exit-intf)

- default route # ip route 0.0.0.0 0.0.0.0 { ——— }

- Dynamic routing
  - interior Gateway P.
    - Distance Vector Routing P.
      - RIP : R(config)# router rip => R(config-router)# network nw-ip
        verify: show ip route, show ip protocols, debug. ip rip
        passive intf: R(config-router)# passive-interface intf-type intf-number
        เพิ่ม RIP ↔ static: R(config)# router rip => # {redistribute static | default-information originate}
        ถ้าจะ router ตัวอื่น set default route @ intf ที่ต่อกับ protocol อื่น
      - RIP v.2: R(config)# router-rip => # version 2 => # no auto-summary => network nw-ip
        verify: RIP, show ip int brief
      - EIGRP: R(config)# router eigrp AS-# => # eigrp router-id => network nw-ip [wildcard-mask]
        passive intf: R(config-router)# passive-intface intf-type intf-number
        verify: sh ip protocols, show ip eigrp neighbors, show ip route, show ip eigrp topology (all link)
        metrics: R(config-router)# metric weights tos k1 k2 k3 k4 k5
        - set bw: เข้า intf => R(config-if)# bandwidth Kbits-bw-value
    - Link State Routing P.
      - OSPF: R(config)# router ospf (1-65,535) process-id ⇨ -router)# router-id 1.1.1.1 => network nw-ip wildcard-mask area area-id
        set bw: intf ⇨ 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
        ลบ: clear ip ospf process
        redistribute (OSPF ↔ default route): R(config) # ip route 0.0.0.0 0.0.0.0 loopback N
        # router ospf process-id
        => R(config-router) # default-information originate.
      redistribute (OSPF ⇔ อื่นๆ): R(config)# router ospf process-id ⇨ R(config-router)# redistribute!

③ อื่นๆ
- ACL: if ระบุ Name: R(config)# ip access-list [standard | extended] name
- set ACL: R(config) # access-list Acc-num {permit | deny | remark} source [source-wildcard] [log]
- set @ intf: ไป intf => R(config-if)# ip access-group {ACL-num | ACL-name} {in | out}
ลบ: no access-list ACL-num
verify: show ip interface s0/0/0, show access-lists

for Staples

for Staples

-Securing VTP with standard IPv4 ACL : R(config-line) # **access-class** ACL-num {**in**[vrf-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
ip dhcp excluded-address ip addr
ip dhcp pool LAN-POOL-1
R(dhcp-config) # **network** nw-ip subnet-mask
default router ip-address-gateway

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

SWITCH

① basic config
-management intf : s(config)# **interface vlan** N ⇒ s(config-if) # **ip address** ip-addr subnet-mask ⇒ **no sh**
-default gateway : s(config)# **ip default-gateway** ip
-verify : **show running-config** **show ip interface bries**

verify : **show int** [intf-id], **show startup-config**, **sh running-config**. **sh flash**, **show ip** [intf-id], **sh version**, **sh history**, **sh mac-address-table**.

② configure switch port
-duplex communication : s(config-if)# **duplex full** ⇒ **speed** 100
-auto-MDIX : 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** (ch)
⇒ **line vty 0 15** ⇒ s(config-line)# **transport input ssh** ⇒ **login local**
-verify **show ip ssh**, **show ssh**
+ telnet (TCP port 23)

-switch Port Security : s(config-if)# **switch port mode access** ⇒ **switchport port-security**
+static secure MAC Addr. : **switch port port-security mac-address** MAC-ADD
+dynamic ⇒ **switchport port-security mac-address sticky**
+MAX MAC Address : ⇒ **switchport port-security maximum** MAX
+Violation mode : ⇒ **switchport port-security violation** {**protect** | **restrict** | **shutdown**} **mode**
+Verify : **show port security intf** fa0/5, **show port-security address**

③ STP : แบบที่ 1 s(config)# **spanning-tree VLAN** 1 **root** {**primary** | **secondary**}
แบบที่ 2 s(config)# **spanning-tree VLAN** 1 **priority** 24576 (↑ Priority ต่ำ)

verify : **show spanning-trees** [active] PSPT, **show running-config**

-Rapid PVST
+Port Fast : s(config-if)# **spanning-tree port fast**
+BPDU Guard : s(config-if)# **spanning-tree bpduguard mode**
+Config : s(config)# **spanning-tree mode rapid-port** ⇒ เข้า intf ⇒ **spanning-tree link-type point-to-point**
+clear STP ⇒ s# **clear spanning-tree detected-protocol**

④ VLAN : verify : **show vlan name** sc, **show vlan summary** **show ints vlan** num, **show int** fa0/0 **switch port**, **show vlan** [brief]
1) set VTP mode : s(config)# **vtp version 2** ⇨ **vtp mode** {**server** | **client** | **transparent**} ⇨ **vtp domain** name ⇒ **vtp password** pass
2) set trunk : s(config-if)# **switchport mode trunk** 3). สร้าง VLAN@server : s(config)# **vlan** name ⇒ **name** name
4). assign Intf : s(config-if)# **switchport mode access** ⇒ **switchport access vlan** num
5) set inter-VLAN : R(config)# **int** fa0/0.10 ⇒ **encapsulation dot1q** 10 ⇨ **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 ⇒ เข้า 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** secure [source-wildcard] ⇨ **ip nat inside source list** ACL-num **pool** name (**overload**) ← pat dynamic
⇨ เข้า intf. ⇒ s(config-if)# **ip nat** {**Inside** | **outside**}