

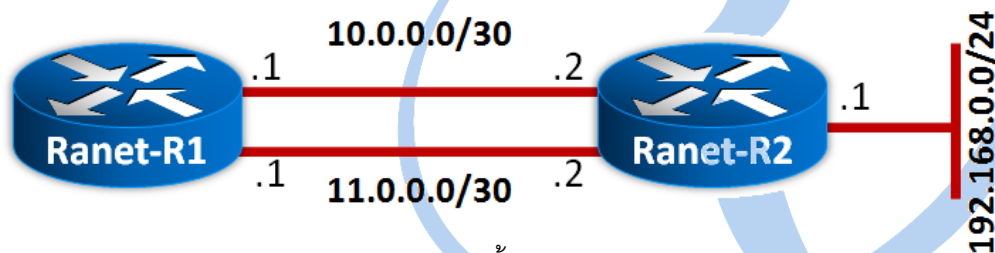
เน็ต 2 เส้น WAN 2 ลิงค์ อยากใช้ประโยชน์ให้ได้มากกว่า Redundancy

เกือบทุกครั้งที่เราไปเสนองานลูกค้าที่ต้องการเพิ่มเน็ตสองเส้น หรือซื้อลิงค์ WAN ไม่ว่าจะ เป็น Leased Line หรือ Frame Relay ที่มีลิงค์สำรอง (Backup Link) ล้วนแต่อยากได้ทั้ง Redundancy ให้เร็วที่สุด รวมถึงถ้าใช้ประโยชน์ได้ทั้งสองเส้น (เช่น การแบ่งโหลด) ก็ยิ่งดีใหญ่ ทำนองเสียตั้งค์ทั้งทีก็เอาให้คุ้ม

ก่อนอื่นคงต้องถามลูกค้าว่าเข้าใจ Load Balance หรือ Load Sharing ตรงกันหรือเปล่า เพราะหลายคนจะไปเข้าใจว่าเป็นงานของเราที่ติ่งโปรโตคอล ทั้งที่ของจริงเป็นกลไกพื้นฐานของเราเตอร์เองแท้ๆ

Load Sharing ระดับ L3: การแบ่งโหลดบนลิงค์ Layer 3 โดยใช้หน้าที่หลักของเราเตอร์ในการเลือกเส้นทาง (Path Selection) ให้แพ็กเก็ต กระทบที่มีเส้นทางที่ดีที่สุดสำหรับแพ็กเก็ตนั้นมากกว่า 1 เส้นทาง บน Routing Table

สรุปง่ายๆ ถ้าบน Routing Table มีเส้นทางสำหรับ Prefix เดียวกันตั้งแต่สองเส้นทางขึ้นไป เราเตอร์จะแบ่งโหลดบนเส้นทางเหล่านั้นให้ โดยไม่ขึ้นกับที่มาของเส้นทาง (ไม่ว่าจะเป็น Static หรือจากเราเตอร์ติ่งโปรโตคอลตัวใด) เช่น กรณีเขียน Static ไปยัง Prefix เดียวกันสองเส้นทางตามภาพ



เขียน Static Route ไปยัง 192.168.0.0/24 ทั้งสองเส้นทางบน Ranet-R1:

```
Ranet-R1(config)#ip route 192.168.0.0 255.255.255.0 10.0.0.2 ← ชี้ออกเส้นบน
Ranet-R1(config)#ip route 192.168.0.0 255.255.255.0 11.0.0.2 ← ชี้ออกเส้นล่าง
```

เนื่องจากทั้งสองเส้นทางเป็น Static เหมือนกัน (AD = 1 เท่ากัน) และระยะทางเท่ากัน จึงปรากฏเส้นทางที่ดีที่สุดไปยัง 192.168.0.0/24 ขึ้นมาทั้งสองเส้นทางดังกล่าวบน Routing Table:

S	192.168.0.0/24	[1/0] via 10.0.0.2
		[1/0] via 11.0.0.2

[ค่า AD / ค่า Metric]

หลักการคัดเส้นทางที่ได้อยู่บน Routing Table:

- เราท์ติ้งโปรโตคอล แต่ละตัว จะคัดเส้นทางที่ดีที่สุดในแต่ละ Prefix ส่งให้ Routing Table โดยพิจารณาจาก Metric (ระยะทาง ซึ่งโปรโตคอลแต่ละชนิดมีหลักการคิดไม่เหมือนกัน)
- Routing Table ถ้าได้รับเส้นทางใน Prefix เดียวกัน จากหลายที่มา (Static, RIP, OSPF, ฯลฯ) จะคัดเลือกเส้นทางที่ดีที่สุดโดยดูจากความน่าเชื่อถือของแหล่งที่มา (ค่า AD) เพื่อแสดงบน Table ให้เราท์เตอร์หยิบไปใช้
- นั่นคือ การจะมีหลายเส้นทางต่อ Prefix เดียวกันแสดงบน Routing Table ให้เราท์เตอร์แบ่งโหลดได้นั้น จะต้อง
 - มีระยะทาง Metric เท่ากัน
 - มาจาก “ที่มา” เดียวกัน

เพื่อดูรายละเอียดการแบ่งโหลดระหว่างเส้นทางเหล่านี้ ให้ใช้คำสั่ง “show ip route (prefix ที่ต้องการ)” ในโหมดอีนาเบิล:

```
Ranet-R#show ip route 192.168.0.0
Routing entry for 192.168.0.0/24
Known via "static", distance 1, metric 0
Routing Descriptor Blocks:
* 10.0.0.2
  Route metric is 0, traffic share count is 1
  11.0.0.2
    Route metric is 0, traffic share count is 1
```

- เครื่องหมายดอกจัน "*" ช้างหน้าเส้นทางใด แสดงว่าแพ็กเก็ตต่อไปจะออกเส้นทางนั้น
- "Traffic share count" แสดงอัตราส่วนในการแบ่งโหลด ซึ่งปกติจะแบ่ง 1-1 (ยกเว้นเราท์ติ้งโปรโตคอลที่มีภาษีดีกว่าคู่แข่งบนเราท์เตอร์ซิสโก้ อย่าง EIGRP ที่สามารถกำหนดสัดส่วนแบ่งโหลดบนแต่ละเส้นทางอย่างไม่เท่ากันตามระยะทาง (Unequal Cost Load Balance) ได้ ด้วยคำสั่ง "*variance (multiplier หรือตัวคูณกับ FD)*" ในโหมดคอนฟิกเราท์เตอร์)

วิธีพิสูจน์ว่าแบ่งโหลดได้จริง นิยมใช้ทูลที่ค้นหาค้นหาด้วยกันคืออย่าง Traceroute ซึ่งถ้าเราตั้งค่าให้แบ่งโหลดต่อแพ็กเก็ต (Per Packet) แล้ว จะเห็นจากผลลัพธ์ได้ว่าที่ Hop เดียวกัน ทั้ง 3 Repeat จะตอบกลับมาจากคนละ Next Hop:

```
Ranet-R#traceroute 192.168.0.1
Tracing the route to 192.168.0.1
 1  10.0.0.2  2 msec
   11.0.0.2  4 msec
   10.0.0.2  3 msec
```

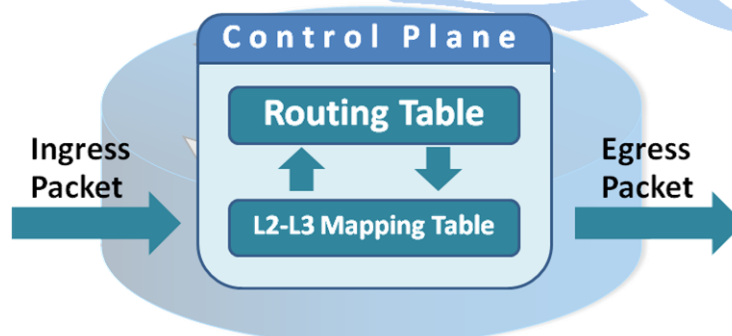
แต่ต้องตั้งข้อสังเกตไว้ก่อนว่า การแบ่งโหลดโดยดีฟอลต์ของเราท์เตอร์จะแบ่งตามไอพีปลายทาง (Per Destination) ซึ่งการตั้งค่าให้แบ่งโหลดแบบ Per Packet หรือ Per Destination จะขึ้นกับรูปแบบการฟอร์เวิร์ดและสวิตชิงแพ็กเก็ตของเราท์เตอร์ (Process/Fast/CEF) ดังนี้

หน้าที่ของเราเตอร์เมื่อ “มีแพ็กเก็ตเข้ามาที่อินเทอร์เฟซ” มี 2 ประการ:

- **ฟอร์เวิร์ดแพ็กเก็ต (Path Selection)** คือการเลือกเส้นทางที่ดีที่สุดสำหรับแพ็กเก็ตนี้จาก Routing Table (หรือดาต้าเบสที่คล้ายกัน*) เพื่อให้รู้ว่าจะฟอร์เวิร์ดแพ็กเก็ตออกอินเทอร์เฟซไหน
- **สวิตชิงเฟรม (Switching)** Switch = เปลี่ยน ในที่นี้คือการเปลี่ยนเทคโนโลยีระดับ Physical-DataLink ให้สอดคล้องกับอินเทอร์เฟซใหม่ที่จะฟอร์เวิร์ดออกไป เช่น เปลี่ยนจาก Ethernet เป็น Frame-Relay โดยใช้ข้อมูลจาก Layer 2 Table (เช่น ARP Table, Frame-Relay Map) (หรือดาต้าเบสที่คล้ายกัน*) ในการเขียน Layer 2 Header ใหม่
- คำว่า “ดาต้าเบสที่คล้ายกัน*” คือ ฐานข้อมูลอื่นนอกจาก Table ที่ประมวลผลหลักบนซีพียู-แรมอย่าง Routing Table, ARP Table, Frame-Relay Map แล้ว ก็จะมี:
 - **Route Cache** คือแคชข้อมูลการฟอร์เวิร์ด-สวิตชิงที่ได้สำรวจข้อมูลมาจากการฟอร์เวิร์ดเฟรมปลายทางนั้นๆ ของซีพียูเป็นครั้งแรก เพื่อความรวดเร็วในการฟอร์เวิร์ด-สวิตชิงข้อมูลไปยังปลายทางเดียวกัน ของการฟอร์เวิร์ดแพ็กเก็ตแบบ Fast Switching
 - **Forwarding Information Base (FIB) และ Adjacency Table** คือฐานข้อมูลอย่างง่ายที่สรุปมาจาก Table หลักในซีพียู-แรม เพื่อให้ฟอร์เวิร์ดแพ็กเก็ตได้อย่างรวดเร็ว ไม่เปลืองการประมวลผลของซีพียู ใช้ในการฟอร์เวิร์ดแพ็กเก็ตแบบ Cisco Express Forwarding (CEF) ซึ่งใช้เป็นโหมดดีฟอลต์บนเราเตอร์ซิสโก้

ขออธิบายพฤติกรรมของการฟอร์เวิร์ดแพ็กเก็ตของเราเตอร์แต่ละรูปแบบ (Process/Fast/CEF) โดยตัดขวางการทำงานของเราเตอร์ Ranet-R1 ให้ดูเพื่อความเข้าใจดังนี้:

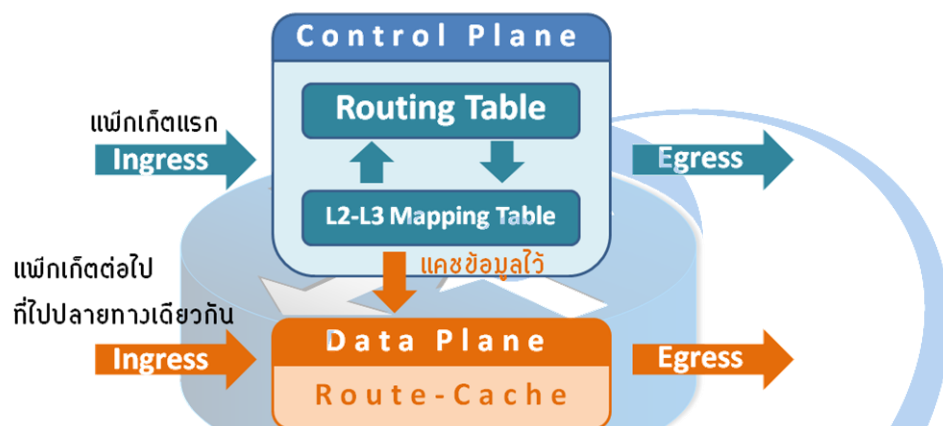
1. Process Switching: CPU (Control Plane) เป็นตัว Switching หลัก



การทำ Process Switching เป็นการฟอร์เวิร์ดแพ็กเก็ตแบบที่ระดับ CCNA เข้าใจ คือ พอข้อมูลเข้าอินเทอร์เฟซ ก็ต้องถอด Frame Header เพื่อส่งไปให้เราเตอร์เลือกเส้นทางจากใน Routing Table ก่อนส่งกลับมาเลือกข้อมูลใน L2-L3 (Address Resolution) Table เช่น ARP Table หรือ Frame Relay Map เพื่อเขียน Frame Header ให้เหมาะกับ Data-Link ของอินเทอร์เฟซขาออก (ล็อกให้ใช้ Switching แบบนี้ด้วยคำสั่ง “no ip route-cache” บนโหมดคอนฟิกอินเทอร์เฟซนั้นๆ)

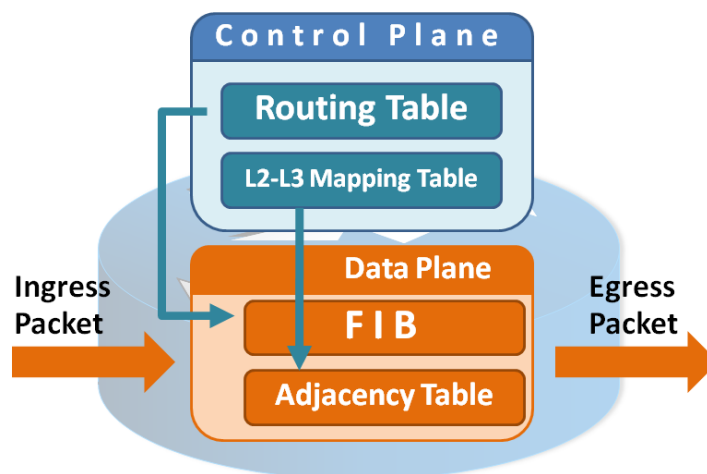
Process Switching จะถูกบังคับใช้โดยเฉพาะในกรณีที่ต้องประมวลผลแพ็กเก็ตขั้นสูง เช่น ผ่าน ACL, Route-Map หรือ NAT ซึ่งต้องใช้ Table อื่นที่ซับซ้อนประกอบการตัดสินใจ (ดังนั้น ตาม Cisco Hierarchy Model จึงบอกว่าไม่ควรใช้พีเจอร์ที่ไปขัดขวางการ Switching ให้ช้าลงกับ Core)

2. Fast Switching: ทำ Route-Cache เพื่อฟอร์เวิร์ดข้อมูลเดิมๆ เร็วขึ้น ไม่ต้องประมวลผลซ้ำ



เนื่องจากแพ็กเก็ตส่วนใหญ่วเวลาเข้ามาที่เรอเตอร์มักไม่ได้มีแพ็กเก็ตเดี่ยวต่อปลายทาง จึงพัฒนาระบบ Fast Switching ขึ้นโดยให้แพ็กเก็ตแรกประมวลผลบน Control Plane ตามปกติ แต่เก็บแคชผลลัพธ์ที่ประมวลผลสำหรับแพ็กเก็ตประเภทเดียวกันนี้ไว้เรียกว่า Route-Cache ใน Data Plane ทำให้เมื่อมีแพ็กเก็ตอื่นที่อยู่ในกลุ่มเดียวกัน (ออกปลายทางเดียวกัน ประมวลผลเหมือนกัน) เข้ามาอีก ก็สามารดึงข้อมูลจาก Route-Cache มาใช้ฟอร์เวิร์ดได้ทันทีโดยไม่ต้องวิ่งขึ้นไป Control Plane ซ้ำ จึงฟอร์เวิร์ดได้เร็วมากกว่าเดิม (Fast) (สั่งเปิดใช้ด้วยคำสั่ง “ip route-cache” บนโหมดคอนฟิกอินเทอร์เฟซนั้นๆ)

3. Cisco Express Forwarding (CEF): บริหารข้อมูลบน Data-Plane ให้พร้อมใช้อย่างรวดเร็วตลอดเวลาแยกต่างหากเลย



เราท์เตอร์ซิสโก้พัฒนาระบบ Switching ใหม่ที่ทำได้เร็วกว่า Fast Switching คือ CEF โดยสร้างฐานข้อมูลพร้อมใช้ในการฟอร์เวิร์ดอย่างรวดเร็วขึ้นต่างหากบนส่วน Data-Plane เลย (ไม่ต้องคอยแคชจากการฟอร์เวิร์ดแพ็กเก็ตแรกเหมือน Fast Switching) โดยฐานข้อมูลใหม่มีสองส่วน คือ Forwarding Information Base (FIB) ที่สรุปมาจาก Routing Table และ Adjacency Table ที่สรุปมาจาก L2-L3 Mapping Table (ถือเป็นโหมดดีฟอลต์บนเราท์เตอร์ ด้วยคำสั่ง "ip cef" ในโหมดโกลบอล)

ประเภทการ Switching ทั้งสามแบบ มีผลต่อการเลือกโหมด Load Balance ดังนี้:

โหมด Switching	ภาระ CPU	โหมดแบ่งโหลดดีฟอลต์	เปลี่ยนโหมดแบ่งโหลด
Process Switching	หนักมาก	Per-Packet	ไม่ได้
Fast Switching	เบา	Per-Destination	ไม่ได้
CEF	หนักปานกลาง	Per-Destination	ได้ ใช้คำสั่ง "ip load-sharing (per-destination/per-packet)" ในโหมดคอนฟิกอินเทอร์เฟซ

ที่นี้ ลองมาดูกลไกการส่งเส้นทางให้ Routing Table แบ่งโหลดของเราที่ตั้งโปรโตคอลแต่ละตัวกัน

- โดยดีฟอลต์ Routing Table รองรับเส้นทางแบ่งโหลดได้มากที่สุดต่อ Prefix (Maximum-Path) ได้ 4 เส้นทาง แต่สามารถตั้งค่าให้เราที่ตั้งโปรโตคอลส่งเพิ่มได้ถึง 16 เส้นทาง (ด้วยคำสั่ง **“maximum-path (จำนวนเส้นทาง)”** ในโหมดคอนฟิกเราท์เตอร์)
- OSPF มีลูกเล่นเพิ่ม คือสามารถตั้งค่า Cost ในแต่ละอินเทอร์เฟซที่เป็นทางออกของเส้นทางให้เท่ากันได้ (เหมือนโหนดเราท์เตอร์) เพื่อให้ได้หลายๆ เส้นทางส่งไปให้ Routing Table แบ่งโหลด (ตั้งค่าด้วยคำสั่ง **“ip ospf cost (ค่า cost)”** ในโหมดคอนฟิกอินเทอร์เฟซ) แต่แบ่งโหลดได้ด้วยสัดส่วนที่เท่ากัน (Traffic Share Count 1:1) เท่านั้น
- EIGRP มีลูกเล่นมากกว่า (ด้วยความที่เป็นซิสโก้เหมือนกัน) จึงสามารถแบ่งโหลดบนหลายเส้นทางที่ Cost ไม่เท่ากันได้ (ด้วยคำสั่ง **“variance (ตัวคูณ FD)”** ในโหมดคอนฟิกเราท์เตอร์) พร้อมทั้งได้สัดส่วนแบ่งโหลด (Traffic Share Count) แปรผันตามค่า Metric ของแต่ละเส้นทางได้ด้วย จึงสอดคล้องกับความเป็นจริงมากกว่า OSPF

เทคนิคแบ่งโหลดบน Data-Link (L2):

- นอกจากเทคนิคด้าน Routing แล้ว บางกรณี (เช่น มีจำกัดด้าน IP Addressing หรือ อุปกรณ์) ก็อาจจะหันมาเลือกเทคนิคบน Data-Link แทนก็ได้ ซึ่งมีหลายพีเจอร์ให้ใช้:
- EtherChannel เป็นพีเจอร์ที่รวมพอร์ตกายภาพหลายพอร์ตที่มี Characteristic เหมือนกัน ผูก (Bundle) เป็นพอร์ตระดับ L2 (หรือ L3) เหมือนกัน ซึ่งให้หลักการแบ่งโหลด (ที่หลายคนนึกว่าเป็นการเพิ่มแบนด์วิธตามสัดส่วนโดยตรง แต่จริงๆ ไม่ใช่) ได้มากที่สุดถึง 8 ลิงค์อีเทอร์เน็ต และเลือกแบ่งโหลดได้ตาม Src/Des MAC, Src/Des IP หรือ Src/Des Port No. ได้
- แบ่งโหลดด้วย Rapid Per-VLAN STP: เป็นดีฟอลต์บนสวิตช์ที่พอร์ต Trunk จะมี Spanning Tree ไม่เหมือนกันบนแต่ละ VLAN ซึ่งเพิ่มประสิทธิภาพได้ด้วยการเซตใหม่ RSTP และ Manual/VTP Pruning
- First Hop Redundancy Protocol: เช่น HSRP แยกเกตเวย์แต่ละ VLAN แบบ Manual หรือใช้ GLBP ที่จัดการให้โดยอัตโนมัติ สำหรับแบ่งโหลดภายในแลนผ่านลิงค์ Trunk ไปหาเกตเวย์