EIGRP กันลูปได้อย่างไร

เป็น Distance Vector เหมือน RIP ทำไมถึงกล้าประกันว่า Loop-Free ไปพร้อมกับ Best Performance

EIGRP เราท์ติ้งโปรโตคอลที่เป็น Cisco Proprietary (ใช้ได้เฉพาะกับเราท์เตอร์ซิสโก้ด้วยกันเท่านั้น) และเราจะบอกว่า EIGRP อยู่ในกลุ่ม Distance Vector เช่นเดียวกับ RIP (เพราะระยะทางที่ส่งเส้นทางต่อๆ กันมา (Distance) และทิศทางหรืออินเทอร์เฟซที่รับอัพเดต (Vector) มีผลต่อข้อมูลเส้นทางบนเราท์เตอร์เช่นเดียวกับ RIP ถึงใครจะบอกว่าเป็น Hybrid เป็นพันธุ์ผสม หรือเป็น Advance Distance Vector ก็ตาม)

ความรู้เพิ่มเติม: เราท์ติ้งโปรโตคอลแบบยูนิคาสต์ (Unicast Routing Protocol) บนโลกนี้มีอยู่ 3 ประเภทเท่านั้น แบ่งตามกลไกอัพเดตและบริหารจัดการเส้นทาง ดังนี้:

- Distance Vector: ระยะทาง และทิศทางที่รับอัพเดต มีผลต่อข้อมูลเส้นทาง
  - **ได้แก่**: RIP, IGRP และ EIGRP
  - สังเกตจาก อินเทอร์เฟซที่รับอัพเดต จะกลายเป็น “ทางออก” ไปยังเครือข่ายปลายทางนั้นๆ
  - จึงเสี่ยงต่อการเกิดเราท์ติ้งลูป (อัพเดตวนกลับมาหาเราท์เตอร์ตัวเดิม)
  - มีพฤติกรรม Auto-Summarization โดยดีฟอลต์ เมื่ออัพเดตข้าม Major Network Boundary
- Link State: ทุกเราท์เตอร์ในพื้นที่ (Area) เดียวกัน จะรู้จักข้อมูลลิงค์ (Link State) ทุกลิงค์
  - **ได้แก่**: OSPF และ IS-IS
  - สังเกตจาก อินเทอร์เฟซที่รับอัพเดต ไม่มีผลต่อ “ทางออก” จะรับเข้าทางอินเทอร์เฟซไหน ก็ต้องมาประมวลข้อมูลใหม่ทั้งพื้นที่อยู่ดี
  - จึงไม่มีวันวนลูป เพราะทุกเราท์เตอร์มี “แผนที่” ในพื้นที่เดียวกัน เป็นฉบับเดียวกัน
  - และไม่มีการทำ Summarization ภายใน Area เดียวกัน เนื่องจากไม่ได้อัพเดตแบบมีทิศทาง
- Path Vector: เหมือน Distance Vector แต่ไม่ได้คำนวณระยะทาง ใช้พารามิเตอร์ของเส้นทาง (Path Attribute) มาใช้เลือกเส้นทางที่ดีที่สุดแทน (ไม่ได้แปรผันตามแบนด์วิธหรืออะไรอื่นเหมือนพวก IGP)
  - **ได้แก่**: BGP

สรุป ถ้านับเฉพาะเราท์ติ้งโปรโตคอลที่ทำงานภายใน AS (IGP) ก็เหลือแค่ 2 ประเภทเท่านั้น คือ Distance Vector และ Link State

ขึ้นชื่อกันอยู่แล้วว่ากลไกแบบ Distance Vector เอื้อต่อการเกิด Loop โดยกมลสันดานอยู่แล้ว ไม่งั้น RIP คงไม่ต้องพัฒนากลไกต่างๆ เพิ่มเติมมาป้องกันลูปมากมายเหมือนวัวหายแล้วล้อมคอก เจอช่องโหว่ตรงไหนก็อุดตรงนั้น จนทำให้ประสิทธิภาพในการปรับตัวเข้ากับการเปลี่ยนแปลงบนโดเมน (Convergence) ต่ำลงมากเกินกว่าความต้องการขององค์กรส่วนใหญ่ในปัจจุบันจะยอมรับได้

ความรู้เพิ่มเติม: กลไกป้องกันการอัพเดตเส้นทางวนลูปของเราท์ติ้งโปรโตคอลจำพวก Distance Vector ทุกตัวที่มีเหมือนกันคือ:

- Split Horizon "ห้ามอัพเดตเส้นทางออกมาทางอินเทอร์เฟซเดิมที่เคยได้รับเส้นทางเดิมนั้น" หรือสรุปคือ "ห้ามอัพเดตเส้นทางออกอินเทอร์เฟซที่เป็น "ทางออก" ของเส้นทางนั้น" เป็นกฎพื้นฐานของเราท์ติ้งโปรโตคอล Distance Vector ทุกตัว (แต่ EIGRP สั่งปิดกลไกนี้บนอินเทอร์เฟซที่ต้องการได้)
- Poison Reverse เป็นธรรมชาติอีกข้อหนึ่งของ Distance Vector ที่จะอัพเดตเส้นทางที่ "ตายแล้ว" ด้วยค่า Metric ที่มากที่สุดที่เป็นไปได้ เช่น RIP ใช้ Hop Count = 16 ส่วน EIGRP ใช้ Max Delay
- Trigger Update ส่งอัพเดตทันทีที่พบการเปลี่ยนแปลง (เครือข่ายอัพ/ดาวน์) ข้อนี้เป็นสิ่งที่ต้องทำบนทุกเราท์ติ้งโปรโตคอลในปัจจุบัน (เพียงแต่สิ่งที่สร้างความแตกต่างคือ RIP (ซึ่งจัดเป็น Low-Level) จะควบคู่ไปกับการอัพเดตเส้นทางตามคาบเวลา (ทุกๆ ช่วงเวลา - Periodic) แม้ไม่พบการเปลี่ยนแปลงด้วย ทำให้เปลืองแบนด์วิธ และประสิทธิภาพโดยรวมต่ำกว่าตัวอื่น)

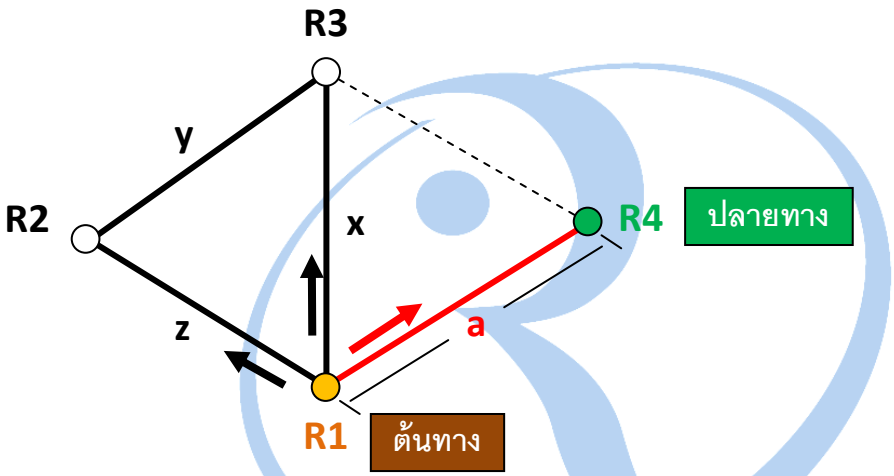
แต่ RIP เพิ่มกลไกเข้ามาอีกสองตัว ที่กระทบต่อประสิทธิภาพอย่างรุนแรง ได้แก่

- Maximum Hop Count ในเมื่อถ้า Loop แล้ว Hop Count จะถูกนับบวกไปเรื่อยๆ ไม่มีที่สิ้นสุด (Count-to-Infinity) ดังนั้น ก็จำกัดจำนวน Hop Count เสียเลย (15 คือมากสุด ถ้า 16 คือเน็ตเวิร์กตายแล้ว) แต่นั่นเท่ากับจำกัดขนาดโดเมนตัวเองไปด้วย ทำให้เสียความ Scalable ไป
- Holddown Timer ถ้าเกิดอัพเดตว่าเน็ตเวิร์กตายแล้วด้วย Poison Reverse แล้วลบออกจากฐานข้อมูลทันที เกิดมีแพ็กเก็ตอัพเดตเส้นทางที่ยัง "เป็น" อยู่ หลงเลือในโดเมนด้วยสาเหตุบางอย่าง เช่น ลิงค์นั้นแบนด์วิธช้ามาก พอมาถึงเราท์เตอร์ที่เคลียร์ดาต้าเบสแล้วก็จะนึกว่าเป็นเส้นทางใหม่ เส้นทางที่สมควรตายแล้วกลับมีชีวิตขึ้นมาใหม่บนโดเมนได้ ก็เลยตั้งตัวจับเวลาไว้ เมื่อใดที่รับ Poison Reverse *(มีต่อ)*

ความรู้เพิ่มเติม: (ต่อ) มา แทนที่จะลบเส้นทางนั้นออกจากฐานข้อมูลตัวเอง ก็จะค้างไว้อย่างนั้นก่อนพร้อมตีตราว่าเป็นเส้นทาง “ตายแล้ว” ต่ออีก **180 วินาที** ก่อนจะโดน Flush ออกจริงๆ ทำให้ถ้าเกิดกรณีข้อมูลเส้นทางที่ “เป็น” หลงเหลืออยู่ ก็ไม่สามารถเข้ามาอัพเดตได้ เพราะถูกตีตราว่าควรจะ “ตาย” อยู่ แต่ด้วยเหตุนี้จึงทำให้ ถ้าเกิดเน็ตเวิร์กดังกล่าวเกิด “อัพ” ใหม่ภายในช่วงเวลานั้น จะไม่สามารถอัพเดตได้ ต้องรอดีเลย์ถึง 180 วิ เหตุนี้เอง RIP จึงได้ชื่อว่าเป็นเราท์ติ้งโปรโตคอลที่ปรับตัวให้เข้ากับความเปลี่ยนแปลงบนโดเมน (Convergence) ได้น้อยที่สุดนั่นเอง

พอเป็น EIGRP ที่เป็น Distance Vector เหมือนกัน แต่ซิสโก้จะต้องเข็นขึ้นมาให้ดูดีกว่า OSPF (จะได้ใช้เป็นจุดขายของแบรนด์ตัวเอง) ก็คงใช้วิธีบ้านๆ อย่าง Maximum Hop Count และ Holddown Timer ไม่ได้ และควรใช้กลไกที่เป็น “เนื้อแท้” ของธรรมชาติการอัพเดตจริงๆ ที่ไม่กระทบต่อประสิทธิภาพ

จึงมีการวิจัยกันจริงๆ (เขียนเป็นเปเปอร์ระดับชาติกันเลย) ถึงเอกลักษณ์ของเส้นทางที่เกิดซึ่งมีคุณสมบัติที่ตรงกันคือ “**เส้นทางที่นับตั้งแต่จุดเริ่มต้นไปยังจุดปลายทางที่ไม่ย้อนกลับมาทับจุดเริ่มต้นเดิมอีกที จะต้องมีระยะทางน้อยกว่าระยะทางที่จุดถัดไป** (Neighbor) **ไปถึงจุดปลายทางที่สั้นที่สุด**” ถ้างงก็ให้นึกถึงกราฟถ่วงน้ำหนัก ดังตัวอย่างนี้



จุดถัดไป	เส้นทาง	ระยะไปยังจุดถัดไป	ระยะจากจุดถัดไปถึงปลายทาง	ระยะทางรวม	วนลูป?
ดีที่สุด R4	R1 → R4	a	0	a	ไม่
R3	R1 → R3 → R2 → R1 → R4	x	y+z+a	x+y+z+a	ลูป
R2	R1 → R2 → R3 → R1 → R4	Z	y+x+a	x+y+z+a	ลูป

ดังนั้น EIGRP จึงให้นิยามต่างๆ ที่โยงกับข้อพิสูจน์ข้างต้นดังต่อไปนี้:

- **Successor** เป็นเราท์เตอร์เพื่อนบ้านที่เป็น Next Hop ของเส้นทางที่ดีที่สุด
- **Reported Distance (RD)** (เดิมชื่อ Advertised Distance แต่ซิสโก้กลัวเอาไปจำสับกับ AD ที่ย่อมาจาก Administrative Distance เลยเปลี่ยนชื่อใหม่) เป็นระยะทางที่เพื่อนบ้านที่เป็น Next Hop แต่ละตัวรายงาน (Report) มาให้ ว่าตัวเองห่างจากเราท์เตอร์ปลายทางเท่าไร
- **Local Distance (LD)** เป็นระยะทางจากเราท์เตอร์ตัวเองไปยังเราท์เตอร์เพื่อนบ้านนั้นๆ
- **Feasible Distance (FD)** เป็นระยะจากตัวเองไปยังเราท์เตอร์ปลายทาง ของเส้นทางที่ **"สั้นที่สุด"**

ซึ่งพออ้างอิงจากข้อพิสูจน์ทางคณิตศาสตร์ข้างต้น ก็จะได้สูตรในการตรวจสอบเส้นทางลูปว่า:

ถ้าเราท์เตอร์เพื่อนบ้านใด บอก RD มากกว่าหรือเท่ากับ FD ของปลายทางนั้น แสดงว่า ลูป

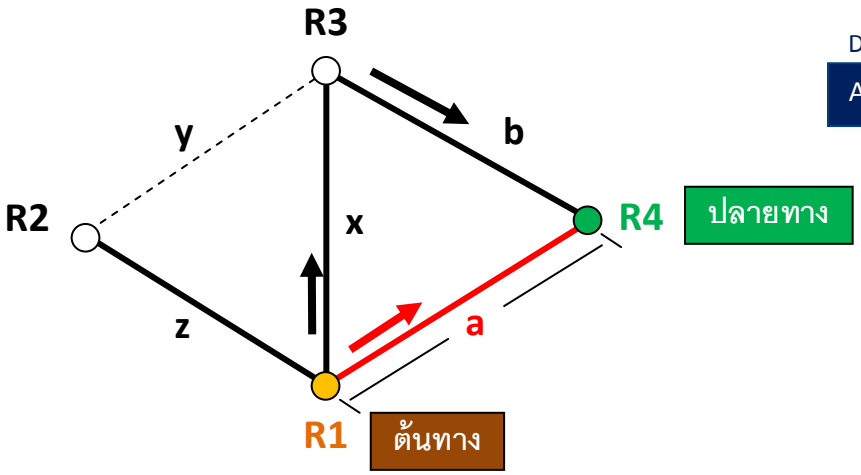
หรือ เส้นทางที่ไม่เกิดลูปจะต้อง RD < FD เสมอ

เพื่อให้กลไกป้องกันลูปนี้ไม่หน่วงประสิทธิภาพ จึงให้ข้อพิสูจน์นี้เป็นส่วนหนึ่งของอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดที่เรียกว่า Diffusing Update Algorithm (DUAL) หรือการอัพเดตแบบเบื้องหลัง

อย่างไรน่ะหรือ ก่อนอื่นต้องทำความเข้าใจสถานะของเส้นทางสองแบบก่อนคือ

- **Passive Route** คือเส้นทางที่เสถียรแล้ว คำนวณผลรวม RD+LD ไว้แล้ว และจะไม่เปลี่ยนแปลงค่าเมตริกซ์ดังกล่าวตราบเท่าที่ RD ที่เพื่อนบ้านตัวนั้นบอกมาแล้วยัง < FD
- **Active Route** คือเส้นทางที่ไม่เสถียร เช่น เกิดจากเพื่อนบ้านตัวนั้นขาดการติดต่อเกินเวลาที่กำหนด หรืออัพเดตค่า RD ที่มากกว่าหรือเท่ากับ FD (สัญญาณว่ากำลังลูป)

ลองดูตามแผนภาพต่อไปนี้



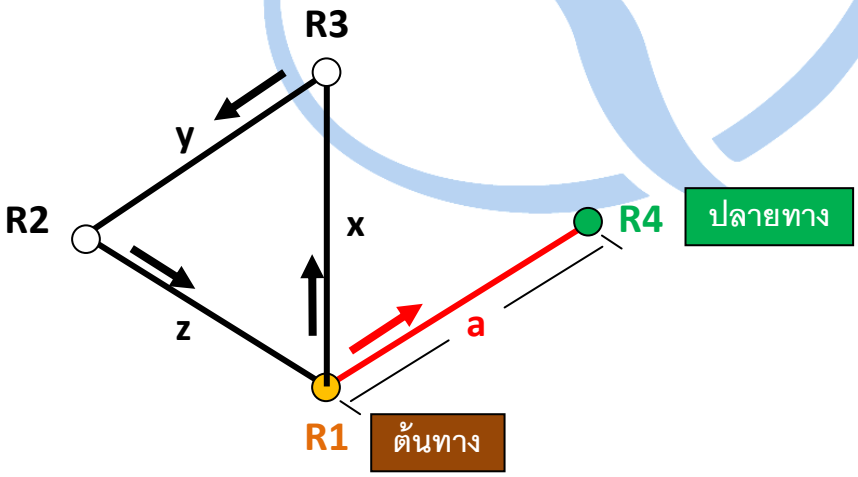
บนฐานข้อมูลเส้นทาง EIGRP (หรือเรียก Topology Table) จะพบเส้นทางไปยัง R4 สองเส้นทาง:

	Neighbor	Route	LD	RD	LD+RD	วนลูป?
Successor	R4	R1 → R4	a	0	a	ไม่
	R3	R1 → R3 → R4	x	b	x+b	ไม่

จะเห็นว่า R4 เป็น Successor ระยะ a จึงเป็น FD ของปลายทาง R4 ขณะที่ R3 เป็นเส้นทางที่ดีรองลงมา (เป็นเส้นทางสำรอง เรียก Feasible Successor) เพราะ LD+RD ดีรองมาเป็นอันดับสอง แต่ไม่ลูป เพราะไม่ได้มากกว่าหรือเท่ากับ FD

ทั้งสองเส้นทางถือว่าอยู่ใน Passive State ค่า LD+RD จะไม่เปลี่ยนแปลง แต่ค่า RD ตัวเพื่อนบ้านอาจอัพเดตมาใหม่ได้ถ้าพบการเปลี่ยนแปลง

แต่ถ้าในกรณีลิงค์ระหว่าง R3 กับ R4 Down ทำให้ R3 ต้องมาใช้เส้นทางไปทาง R2 แทนเพื่อไปหา R4 ค่า RD ใหม่ที่อัพเดตมาให้ R1 จะมากกว่า มากกว่าทั้ง LD+RD เดิมของตัวเอง และมากกว่า FD ด้วย เกิดลูป



	Neighbor	Route	LD	RD	LD+RD	วนลูป?
Successor	R4	R1 → R4	a	0	a	ไม่
	R3	R1 → R3 → R2 → R1 → R4	x	y+z+a	x+b	ลูป

ทำให้เส้นทาง R3 ไม่ใช่เส้นทางที่ “ควรใช้” อีก DUAL ก็จะปรับสถานะเป็น “active” สำหรับ Neighbor R3 แทน (ทำให้ถูกถอดออกจากตำแหน่ง Feasible Successor ด้วย หรือแม้กระทั่งเส้นทางที่มีสิทธิ์ร่วมวงเวลาตั้งค่า Load Balance แบบ Unequal Cost)

คำว่า Diffusing Update จึงสื่อถึง การอัพเดต RD มาเรื่อยๆ จาก Neighbor เมื่อตัว Neighbor พบการเปลี่ยนแปลง แต่จะยังไม่กระทบต่อข้อมูลเส้นทางหลักจนกว่าจะเข้าเงื่อนไขที่ทำให้เป็นเส้นทางไม่มีคุณภาพ

ด้วยเหตุนี้ EIGRP จึงกล้าโฆษณาตัวเองว่า Loop Free พร้อมๆ กับ Best Performance เพราะไม่ต้องพึ่งพากลไกแบบล้อมคอกอย่าง RIP นั่นเอง