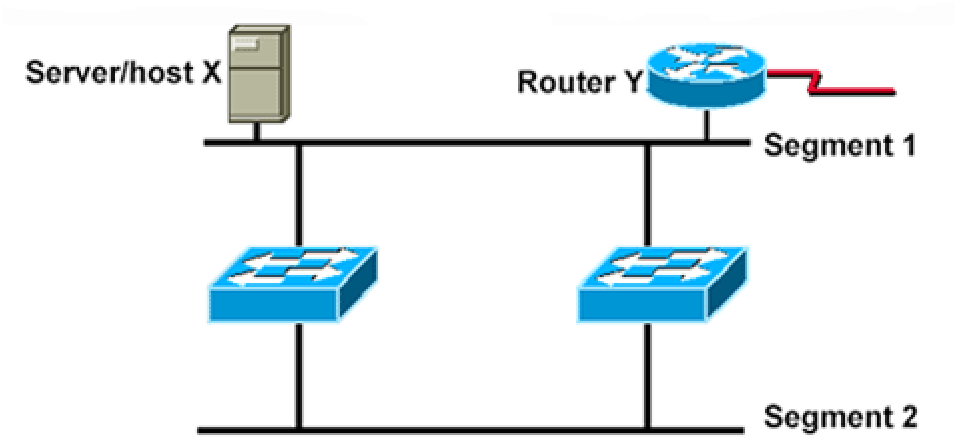
บทที่ 6
สแปนนิ่งทรีโพรโตคอล

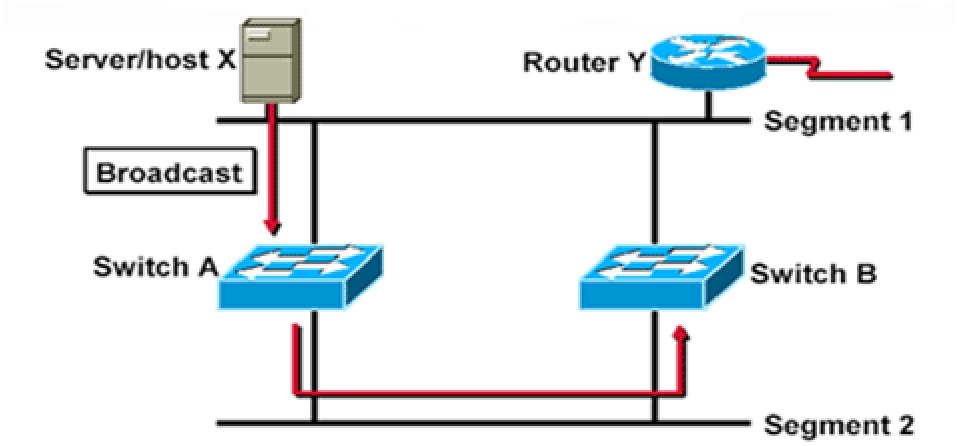
ในการสร้างระบบเครือข่ายที่ดีนั้น ควรต้องคำนึงถึงความน่าจะเป็นในทุกด้าน เมื่อเกิดกรณีฉุกเฉิน ระบบเครือข่ายจะทำการแก้ไขเช่นไร เช่น ถ้าในระบบ สวิตช์เกิดเสีย ใช้งานไม่ได้ จะทำเช่นไร ด้วยเหตุนี้ จึงทำให้เกิดเป็นแนวคิด Redundant Topology ขึ้น คือเป็นการสร้างระบบเครือข่ายแบบต่อให้มี สวิตช์ 2 ตัวในเส้นทาง (path) เดียวกัน เพื่อป้องกันในเหตุที่ถ้ามีสวิตช์ตัวใดตัวหนึ่งเสีย สวิตช์ตัวที่เหลือจะยังสามารถใช้งานต่อไปได้ ทำให้ระบบเครือข่ายไม่หยุดชะงัก



รูปที่ 6-1 แสดงตัวอย่างของเครือข่ายที่ใช้ Redundant Topology

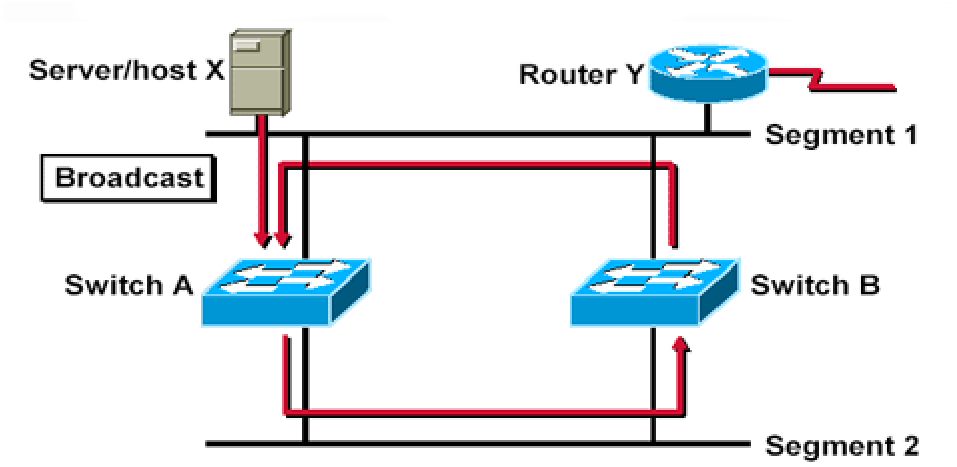
เมื่อเราสร้างระบบเครือข่ายโดยใช้ Redundant Topology แล้ว อาจทำให้เกิดลูป (loop) ขึ้นได้ ซึ่งเป็นสาเหตุทำให้เกิดบอร์ดคาสต์สตอร์มขึ้นในระบบเครือข่ายจนอาจเป็นสาเหตุทำให้เครือข่ายล่มได้

โดยขั้นตอนของการเกิดลูปจะเป็นดังนี้คือ



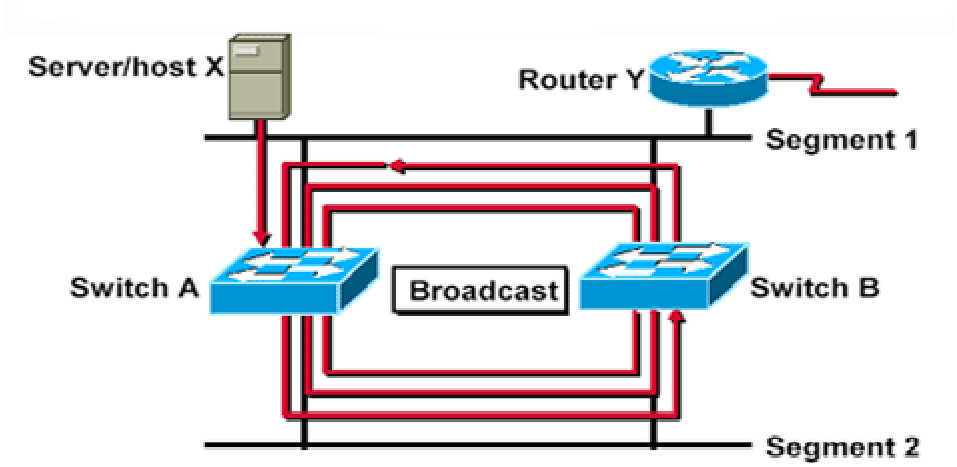
รูปที่ 6-2 แสดงการส่งข้อมูลจากโฮสต์ X ไปยังสวิตช์ A

เมื่อโฮสต์ X ส่งเฟรมข้อมูลออกไปให้กับสวิตช์ A จากนั้นสวิตช์ A จะทำการใส่แอดเดรสต้นทาง (Source Address) เข้าไปในตารางแมคแอดเดรสแล้วจึงทำการส่งออกไปยังทุกพอร์ตออกไป ทำให้สวิตช์ B ได้รับเฟรมข้อมูลนี้ด้วย



รูปที่ 6-3 แสดงการฟลัดเอาท์ (Flood out) ของสวิตช์ A

เมื่อสวิตช์ B ได้รับเฟรมข้อมูล ก็จะทำการใส่แอดเดรสต้นทางลงไปในตารางแมคแอดเดรสแล้วทำการส่งออกไปยังทุกพอร์ต (Flood Out) อีกครั้ง ทำให้สวิตช์ A ได้รับเฟรมข้อมูลนี้อีกครั้ง



รูปที่ 6-4 แสดงการเกิดบอร์ดคาสต์สตอร์ม (Broadcast Storm)

เมื่อสวิตช์ A ได้รับเฟรมข้อมูล ก็จะกระทำการเหมือนเดิม และต่อเนื่องไปถึงสวิตช์ B วนเป็นลูปไปเรื่อย ๆ ไม่รู้จบ และแม้ว่า เฟรมข้อมูลไปถึง ณ จุดหมายที่สเตชันที่ต้องการแล้ว ตัวสวิตช์ A และสวิตช์ B ก็จะยังส่งเฟรมข้อมูลกันต่อไปแบบไม่รู้จบ จนทำให้เกิดเป็นบอร์ดคาสต์สตอร์ม สร้างทราฟฟิกให้กับระบบเครือข่าย จนอาจทำให้เครือข่ายหยุดชะงักได้

6.1 สแปนนิ่งทรีโพรโตคอล (Spanning-Tree Protocol)

สแปนนิ่งโพรโตคอลถูกสร้างมาเพื่อใช้ในการกำจัดลูปที่เกิดขึ้นในระบบเครือข่าย โดยจะเสมือนจัดให้ระบบเครือข่ายมีลักษณะเหมือนกับต้นไม้ (Tree) คือจากจุดเริ่มต้นแล้วมีการแตกกิ่งก้านออกมาเรื่อย ๆ เป็นสายแห่งการติดต่อสื่อสาร จะทำให้เป็นเครือข่ายที่ไม่มีลูปเลย

หลักการทำงานของสแปนนิ่งโพรโตคอลคือ สวิตช์ทุกตัวจะทำการส่งเฟรมชื่อว่า Bridge Protocol Data Unit (BPDU) ออกไปยังทุกพอร์ต เพื่อแสดงถึงความมีตัวตนของสวิตช์ แล้วสวิตช์ทุกตัวจะได้รับ BPDU จากสวิตช์ตัวข้างเคียงเพื่อนำมาใช้ในการคำนวณโดยใช้สแปนนิ่งอัลกอริธึม (Spanning Tree Algorithm) ซึ่งมีหลักการคือ

- เลือกรูทบริดจ์ (Root Bridge)
- เลือกรูทพอร์ต (Root Port)
- เลือกดีไซเนเตทพอร์ต (Designated Port)

สวิตช์ทุกตัวจะทำการส่ง BPDU ออกไปยังทุกพอร์ต เป็นการแสดงถึงความมีตัวตนของสวิตช์ จากนั้นสวิตช์ทุกตัวจึงจะได้รับ BPDU มาเพื่อทำการคำนวณโดยใช้สแปนนิ่งอัลกอริธึมจะได้เป็นสแปนนิ่งทรีโพรเซส (Spanning-Tree Process) ซึ่งก็คือ

6.1.1 เลือกรูทบริดจ์

Root Bridge คือจุดที่เป็นจุดอ้างอิง (Reference) ของสแปนนิ่งทรี ซึ่งก็คือ จุดยอดของต้นไม้นั่นเอง โดยรูทบริดจ์จะเป็นสวิตช์ตัวที่มีบริดจ์ไอดี (Bridge ID) น้อยที่สุด

บริดจ์ไอดีประกอบด้วย

- Bridge Priority (2 bytes) ไพรออริตี้ (Priority) หรือ Weight ของสวิตช์สามารถมีแค่ได้ตั้งแต่ 0 – 65535 และมีค่าดีฟอลต์ (default) คือ 32768
- แมคแอดเดรส (8 bytes) เป็นสิ่งที่แสดงในเห็นถึงความเป็นเอกของแต่ละสวิตช์ เนื่องจากแมคแอดเดรสมีคุณสมบัติคือ มีเพียงแมคแอดเดรสเดียวในโลก ไม่เหมือนใคร และไม่สามารถเปลี่ยนแปลงได้

ในตอนเริ่มต้นนั้น สวิตช์จะตั้งค่ารูทบริดจ์เป็นบริดจ์ไอดีตัวเองก่อน จากนั้นจึงมีการรับ BPDU จากสวิตช์ข้างเคียง แล้วนำมาคำนวณหาตัวที่น้อยกว่า ทำเช่นนี้ไปเรื่อย ๆ จนกว่า สวิตช์ทุกตัวจะมี Root Bridge เป็นตัวเดียวกัน และหลังจากนั้นสวิตช์ก็ยังจะส่ง BPDU ทุก ๆ 2 วินาที (เป็นค่าดีฟอลต์)

6.1.2 เลือกรูทพอร์ต

สำหรับทุก ๆ นอนรูทบริดจ์ (Non-root Bridge) คือสวิตช์ที่ไม่ใช่รูทบริดจ์ต้องทำการเลือกรูทพอร์ต โดยเลือกพอร์ตที่ดีที่สุดในการสื่อสารไปยังรูทบริดจ์ โดยคำนวณจากพอร์ตคอสท์ (Port cost) หรือรูทพาร์ธคอสท์ (Root path cost – จำนวนฮอปทั้งหมดจากรูทบริดจ์จนถึงสวิตช์)

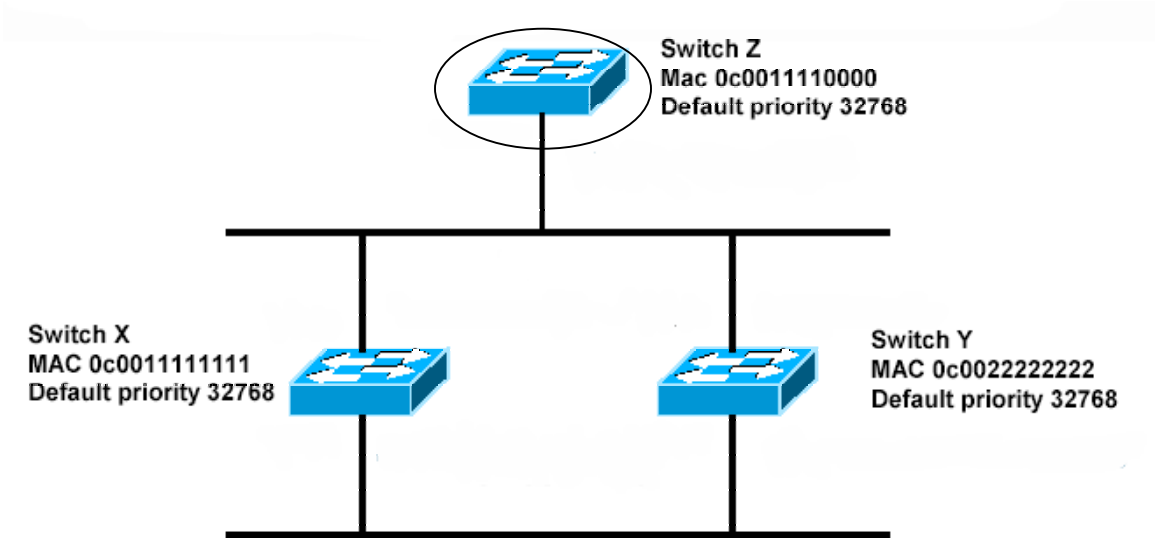
6.1.3 เลือกดีไซเนเตทพอร์ต

ดีไซเนเตทพอร์ตมีคอนเซ็ปต์ว่า ลิงค์เพียงหนึ่งเดียวในเซกเมนต์ที่ใช้ในการส่งและรับทราฟฟิก โดยสำหรับตัวรูทบริดจ์นั้นถือว่า ทุก ๆ พอร์ตของรูทบริดจ์จะเป็นดีไซเนเตทพอร์ตและสำหรับนอนรูทบริดจ์จะให้พอร์ตที่ต่อกับรูทพอร์ตของสวิตช์ตัวข้างเคียงเป็นดีไซเนเตทพอร์ต และสำหรับพอร์ตที่ไม่ใช่รูทพอร์ตและดีไซเนเตทพอร์ตจะถูกบล็อก (Block)

ด้วยหลักการทำงานนี้ จะทำให้แต่ละเครือข่ายแลนสามารถส่งเฟรมข้อมูลไปยังสวิตช์ได้เพียงเครื่องเดียว และเกิดสภาพของต้นไม้ (Tree) ขึ้น โดยพอร์ตที่ไม่ได้ใช้งานจะเป็นพอร์ตสำรอง และเมื่อดำเนินการทำจนเกิดเป็นสแปนนิ่งทรีแล้ว จะได้เครือข่ายที่มีลักษณะ

- **One Root Bridge per Network**

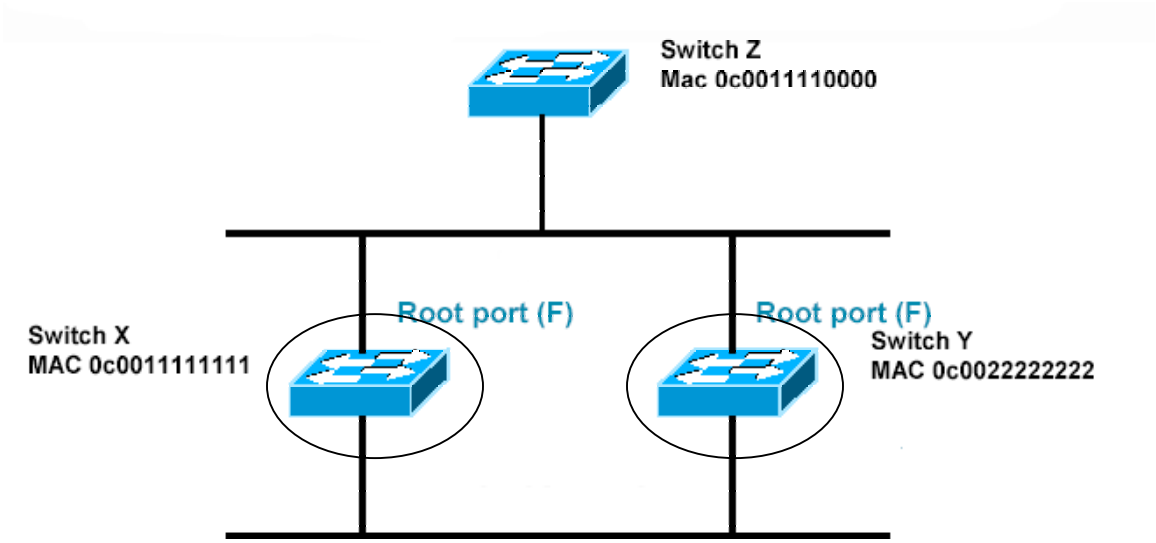
  1 Root Bridge ต่อ 1 ระบบเครือข่าย



รูปที่ 6-5 แสดงระบบเครือข่ายที่มีสวิตช์ Z เป็นรูทบริดจ์

- **One Root port per Non-Root Bridge**

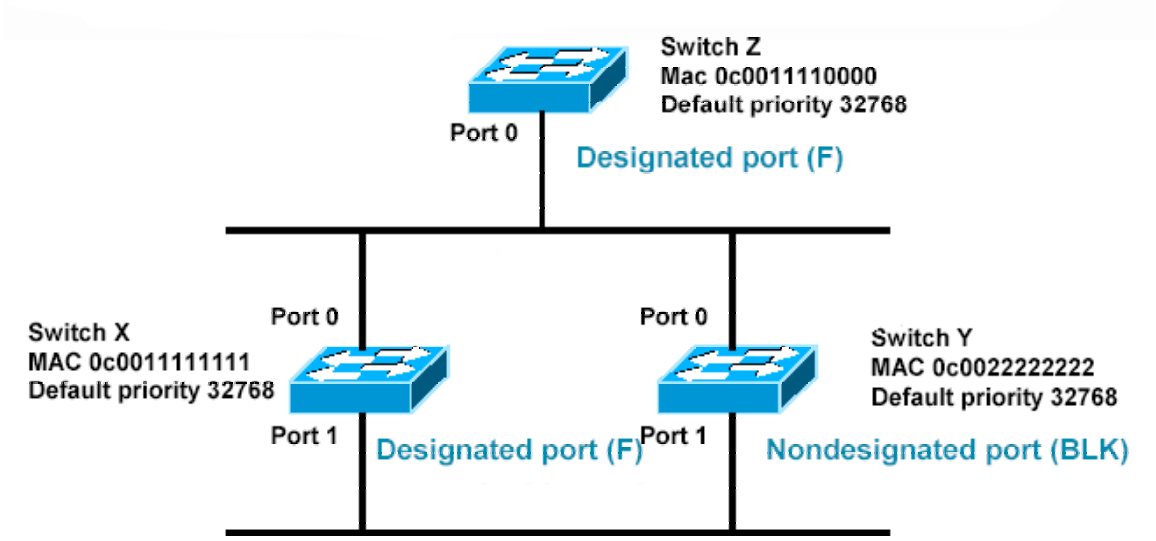
  1 Root Port ต่อ 1 Non-root Bridge



รูปที่ 6-6 แสดงระบบเครือข่ายที่มีสวิตช์ X และสวิตช์ Y เป็นนอนรูทบริดจ์

- **One Designated port per Segment**

  1 Designated port ต่อ 1 เซกเมนต์



รูปที่ 6-7 แสดงระบบเครือข่ายที่มี 1 ดีไซเนเตทพอร์ตต่อ 1 เซกเมนต์

6.2 สแปนนิ่งทรีสเตท (Spanning Tree Port States)

ในการใช้สแปนนิ่งโพรโตคอลทุก ๆ พอร์ตของสวิตช์ จะต้องทำงานเป็นขั้นตอนผ่านไปในแต่ละสเตทโดยจะเริ่มต้นที่ Disabled State และสิ้นสุดที่สเตทสุดท้ายซึ่งอนุญาตให้พอร์ตสามารถรับ-ส่งเฟรมข้อมูลได้ โดยสเตทมีดังต่อไปนี้

6.2.1 **Disabled** – เป็นสเตทของพอร์ตที่ชัทดาวน์ (Shut down) คือไม่สามารถทำอะไรได้เลย สเตทนี้ไม่ถือว่าเป็นส่วนหนึ่งของสแปนนิ่งทรีสเตท

6.2.2 **Blocking** – หลังจากการติดตั้งค่าให้กับพอร์ตแล้ว พอร์ตจะเข้าสู่ Blocking State เพื่อไม่ให้การส่งเฟรมข้อมูลแบบลูป โดยในสเตทนี้จะไม่สามารถรับ – ส่งข้อมูลได้ รวมถึงไม่มีการเรียนรู้แมคแอดเดรส คือ ไม่มีการเพิ่มแมคแอดเดรสลงตารางแมคแอดเดรสอีกด้วย แต่สามารถรับ BPDU จากสวิตช์ข้างเคียงได้ นอกจานี้พอร์ตที่อยู่ในโหมดสแตนด์บาย (Standby Mode) จะเข้าสู่ Blocking State เช่นเดียวกัน

6.2.3 **Listening** – พอร์ตจะเปลี่ยนสเตทจาก Blocking State ไปเป็น Listening State ก็ต่อเมื่อพอร์ตได้รับการเลือกให้เป็นรูทพอร์ตหรือดีไซเนเตทพอร์ต หรือในอีกนัยหนึ่งก็คือ พอร์ตกำลังจะสามารถฟอร์เวิร์ดทราฟฟิก (Forward Traffic) ได้นั่นเอง โดยในสเตทนี้พอร์ตจะสามารถรับ – ส่ง BPDU ได้แต่ถ้าพอร์ตสูญเสียสถานะการเป็นรูทพอร์ตหรือดีไซเนเตทพอร์ต เมื่อใด พอร์ตจะกลับไปสู่ Blocking State อีกครั้ง

6.2.4 **Learning** – เมื่อผ่านช่วงเวลาที่เรียกว่า Forward Delay ใน Listening State แล้วพอร์ตจะเปลี่ยนไปสู่ Learning State โดยที่พอร์ตจะสามารถรับ – ส่ง BPDU ได้และสวิตช์ยังสามารถเรียนรู้แมคแอดเดรสได้อีกด้วย

6.2.5 **Forwarding** – เมื่อผ่านช่วงเวลาที่เรียกว่า Forward Delay ใน Learning State แล้วพอร์ตจะเปลี่ยนไปสู่ Forwarding State โดยพอร์ตจะสามารถรับ-ส่งข้อมูล, เรียนรู้แมคแอดเดรสและรับ-ส่ง BPDU ได้โดยในสเตทนี้ถือว่าได้ทำสแปนนิ่งทรีเสร็จสิ้นแล้วนั่นเอง

6.3 ชนิดของสแปนนิ่งโพรโตคอล (Types of Spanning Tree Protocol)

ในตอนเริ่มต้นนั้นสแปนนิ่งโพรโตคอลได้รับการพัฒนามาเพื่อใช้กับเครือข่ายแลน (VLAN) เดียวเท่านั้น การสร้างสแปนนิ่งโพรโตคอลเพื่อให้สามารถใช้ได้กับ Multiple VLANs นั้นจำเป็นต้องประกอบไปด้วยหลาย ๆ สิ่ง ด้วยเหตุนี้ จึงได้มีการสร้างสแปนนิ่งโพรโตคอลที่แตกต่างกันออกมา ซึ่ง ณ ที่นี้เราจะขอหยิบขึ้นมา 3 ประเภทด้วยกันคือ

6.3.1 คอมมอนสแปนนิ่งทรี (Common Spanning Tree (CST))

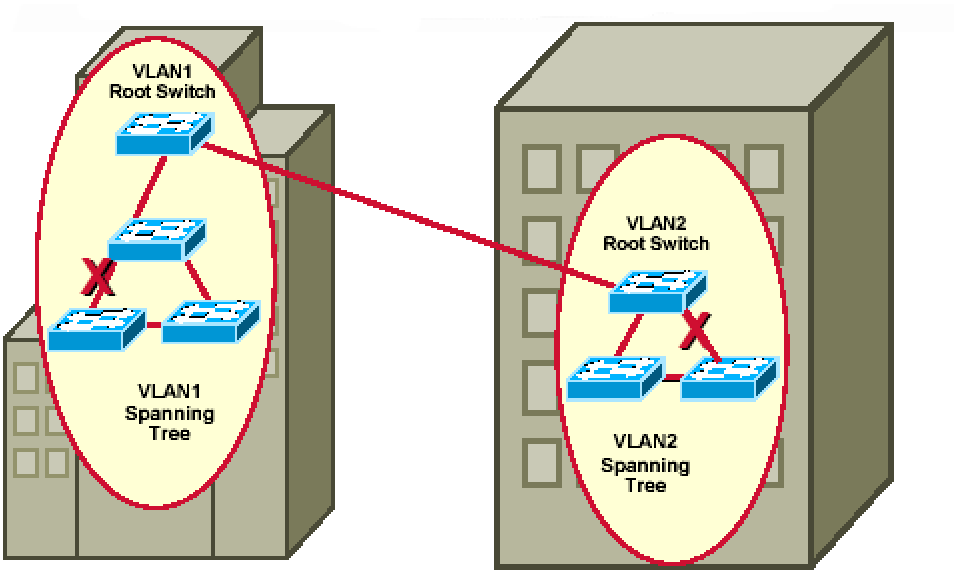
มาตรฐาน IEEE 802.1Q ได้กำหนดถึงการสร้างทรังค์ลิงค์ (Trunk Link) ที่ใช้ระหว่างสวิตช์และได้กำหนดให้ใช้เป็น 1 สแปนนิ่งทรีต่อหนึ่งเครือข่าย ก็คือทุกวีแลนมีชื่อเรียกว่าคอมมอนสแปนนิ่งทรี (Common Spanning Tree (CST)) หรือ โมโนสแปนนิ่งทรี (Mono Spanning Tree (MST)) ทุก ๆ BPDUs จะถูกส่งไปทั่วทั้งเครือข่าย

การใช้เพียง 1 สแปนนิ่งทรีต่อหลาย ๆ วีแลน มีข้อดีคือ สามารถใช้คำสั่งได้ง่าย ไม่ยุ่งยาก และยังลดการใช้ซีพียู (CPU) ของสวิตช์ในการคำนวณอีกด้วย อย่างไรก็ตาม การกระทำเช่นนี้ยังมีข้อเสียอยู่ก็คือ Redundant Link ที่ได้กำหนดขึ้นจะไม่ได้ใช้งาน ถูกบล็อกเพื่อกำจัดลูปและยังมีข้อจำกัดในเรื่องวีแลนคือพอร์ตที่ถูกบล็อค อาจจะเป็นพอร์ตที่ใช้ส่งเฟรมข้อมูลของวีแลนใด ๆ ก็ได้ จึงอาจจะเป็นเหตุให้ไม่สามารถส่งเฟรมข้อมูลได้

6.3.2 เปอร์วีแลนสแปนนิ่งทรี (Per-VLAN Spanning Tree (PVST))

เปอร์วีแลนสแปนนิ่งทรี (Per-VLAN Spanning Tree) เป็นสแปนนิ่งทรีที่มีความยืดหยุ่นมากกว่า CST เนื่องจากมีหลักการทำงานในการสร้างสแปนนิ่งทรีแยกเป็นของแต่ละวีแลนเลย เพื่อสามารถใช้คำสั่งได้อย่างเป็นอิสระ และยังมีประสิทธิภาพมากกว่า Multiple Spanning Tree สามารถสร้าง Load Balancing บน Redundant Links เมื่อลิงค์นั้นถูกกำหนดไว้ในคนละวีแลนกัน

เนื่องจากคุณสมบัติของ PVST จึงจำเป็นต้องใช้ Cisco Inter-Switch Link (ISL) ในการส่งข้อมูลผ่านทรังค์ลิงค์ระหว่างสวิตช์

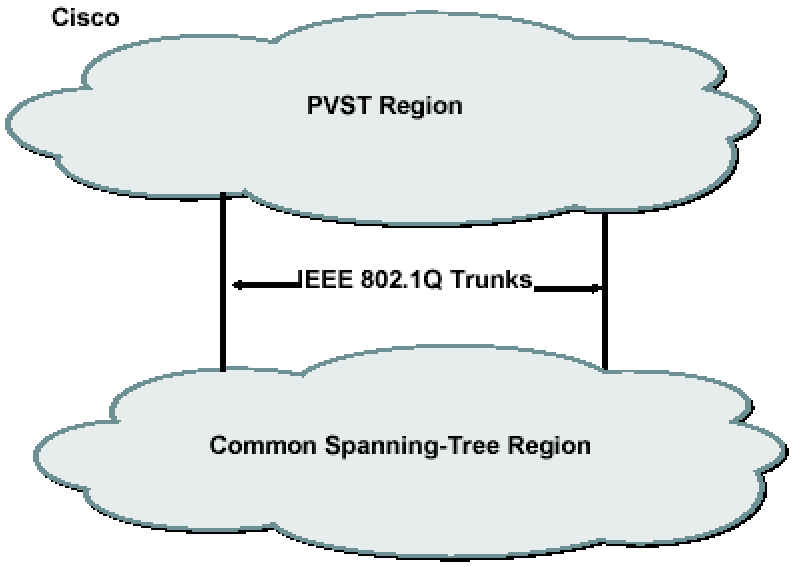


รูปที่ 6-8 แสดงตัวอย่างเครือข่ายที่ใช้เปอร์วีแลนสแปนนิ่งทรี

6.3.3 เปอร์วีแลนสแปนนิ่งทรีพลัส (Per-VLAN Spanning Tree Plus (PVST+))

เปอร์วีแลนสแปนนิ่งทรีพลัส (Per-VLAN Spanning Tree Plus) มีความสามารถในการติดต่อสื่อสารระหว่าง CST และ PVST PVST+ สามารถสนับสนุนการทำงานของสแปนนิ่งทรีทั้ง 3 กลุ่มคือ Catalyst ที่ใช้ PVST, Catalyst ที่ใช้ PVST+ และสวิตช์ที่ใช้งาน CST/MST บน 802.1Q

หลักในการทำงานคือ PVST+ จะเป็นเสมือนตัวที่ใช้ติดต่อสื่อสารระหว่างกลุ่มของ CST สวิตช์และกลุ่มของ PVST สวิตช์โดย PVST+ สามารถติดต่อกับ PVST ได้โดยตรงโดยผ่าน ISL Trunks และติดต่อกับ CST อย่างไรก็ตาม PVST+ จะแลกเปลี่ยน BPDUs กับ CST บนวีแลน1 สำหรับ BPDUs ที่มาจากสแปนนิ่งทรีของแต่ละวีแลนนั้นจะเดินทางไปยังส่วนของ CST ในระบบเครือข่ายด้วย Tunnel PVST+ จะส่ง BPDUs ที่กล่าวมานี้โดยการใช้ Multicast Address ดังนั้น CST สวิตช์จะสามารถ Forward BPDUs ไปยังสวิตช์ข้างเคียงได้ จนในที่สุด BPDUs จึงจะสามารถเดินทางไปถึง PVST+ สวิตช์ได้



รูปที่ 6-9 แสดงการทำงานของเปอร์วีแลนสแปนนิ่งทรีพลัส