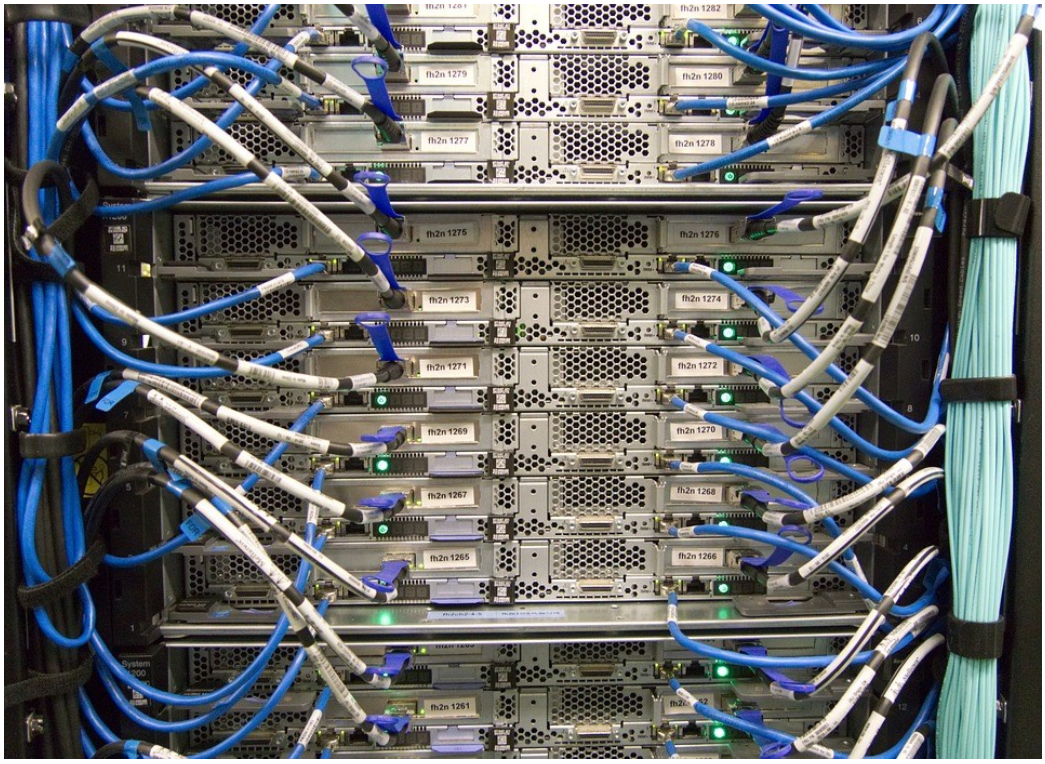
Computer Network Administration

การบริหารเครือข่ายคอมพิวเตอร์



เอกสารคำสอนวิชาการบริหารเครือข่ายคอมพิวเตอร์ (รหัสวิชา 1204320, 1204326)

ผศ.ดร.สุชาติ คุ้มมะณี

สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

มกราคม 2561

คำนำ

เอกสารคำสอนนี้ใช้สำหรับการเรียนการสอนในรายวิชา การบริหารเครือข่ายคอมพิวเตอร์ (Computer Network Administration รหัสวิชา 1204320, 1204326) เนื้อหาของรายวิชาเกี่ยวข้องกับระบบสื่อสารและเครือข่าย โดยเน้นวิธีการออกแบบระบบเครือข่ายตั้งแต่ระดับต้นไปจนถึงระดับสูง วิธีการวิเคราะห์และการแก้ไขปัญหาระบบเครือข่าย การบริหารจัดการและการบำรุงรักษาระบบเครือข่าย เป้าหมายของรายวิชานี้ คือ ผู้เรียนต้องมีความสามารถบริหารจัดการเครือข่ายได้จริง โดยเนื้อหาในเอกสารคำสอนนี้ครอบคลุมแผนการสอน (มคอ 3) ทั้งหมดในรายวิชา Computer Network Administration (รหัสวิชา 1204320, 1204326)

เนื่องจากวิชาดังกล่าวจำเป็นต้องมีการปฏิบัติควบคู่ไปกับทฤษฎีเพื่อเสริมทักษะให้นักศึกษาเกิดความเข้าใจ มีความชำนาญ และประยุกต์เอาความรู้ที่ได้จากภาคทฤษฎีมาบูรณาการร่วมกับภาคปฏิบัติ จะทำให้นักศึกษาเกิดความเข้าใจในกระบวนการของการสื่อสารมากยิ่งขึ้น เนื้อหาของรายวิชามีทั้งหมด 4 ภาค คือ

ภาคที่หนึ่ง อธิบายถึงทฤษฎีและการออกแบบเครือข่ายพื้นฐาน ซึ่งเป็นเนื้อหาเบื้องต้นเกี่ยวกับระบบเครือข่าย เช่น โครงสร้างพื้นฐานในการสื่อสาร และมาตรฐานการสื่อสารข้อมูล การออกแบบเครือข่ายเบื้องต้นทั้งในระดับ LAN และ WAN

ภาคที่สอง อธิบายถึงการออกแบบและการวิเคราะห์เครือข่ายขั้นสูง เนื้อหาประกอบไปด้วยการเชื่อมต่อเครือข่ายที่มีความซับซ้อนและประยุกต์ใช้งานกับแอพพลิเคชันที่จำเป็นต้องใช้บนเครือข่าย เช่น DNS, DCHP, NTP, FTP เป็นต้น รวมถึงการวิเคราะห์หาสาเหตุและตรวจสอบข้อมูลอย่างละเอียดที่เกิดขึ้นบนเครือข่าย เช่น ARP, Broadcast เป็นต้น

ภาคที่สาม อธิบายถึงโอเพนซอร์สเราเตอร์ (Open source router) เพื่อเป็นทางเลือกในการเลือกใช้ซอฟต์แวร์ที่ไม่มีค่าใช้จ่ายกับระบบงานด้านเครือข่าย

ภาคที่สี่ เป็นตัวอย่างการติดตั้งและปรับแต่งระบบเครือข่ายโดยผู้สอนทำการสาธิตผ่านวิดีโอ (Video training) นิสิตสามารถเรียนรู้ได้ด้วยตนเองเพิ่มเติม นอกเหนือจากการเรียนในห้องเรียน และไม่จำกัดเวลาเรียน

ผู้เขียนหวังเป็นอย่างยิ่งว่าเอกสารคำสอนนี้จะช่วยให้ผู้เรียนมีความรู้ความเข้าใจในเนื้อหารายวิชาระบบเครือข่ายได้อย่างละเอียด ถูกต้อง และที่สำคัญคือต้องประยุกต์ใช้กับงานระบบเครือข่ายจริงได้ ถ้ามีเนื้อหาในเอกสารส่วนใดส่วนหนึ่งผิดพลาด ผู้เขียนต้องขออภัยไว้ ณ ที่นี้ด้วย

ผศ.ดร.สุชาติ คุ้มมะณี
สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
email : suchart.k@msu.ac.th
ปรับปรุงล่าสุด มกราคม 2561

สารบัญ

สารบัญภาพ

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญตาราง

สารบัญตาราง (ต่อ)

การบริหารเครือข่ายคอมพิวเตอร์

การบริหารและจัดการระบบเครือข่าย (Computer network administrator) คือ การติดตั้ง การปรับแต่ง การดูแล การแก้ไขปัญหา และการบำรุงรักษาอุปกรณ์ต่างๆ ที่เชื่อมกันเข้าเป็นระบบเครือข่าย รวมไปถึงการเก็บรวบรวมข้อมูล วิเคราะห์ข้อมูล การควบคุมดูแลปริมาณของข้อมูลที่มีอยู่ในเครือข่าย และการจัดสรรทรัพยากรที่ติดตั้งอยู่บนเครือข่ายให้สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพสูงสุด การบริหารระบบเครือข่ายคอมพิวเตอร์นั้นสามารถแบ่งออกเป็นส่วนต่าง ๆ ได้เป็น 5 ส่วนหลักคือ

การปรับแต่งค่าเครือข่าย (Configuration management) คือ การปรับแต่งค่าต่างๆ ของอุปกรณ์ในเครือข่าย เช่น หมายเลขไอพี เกตเวย์ เวอร์ชันของซอฟต์แวร์ที่ใช้ในแต่ละเซิร์ฟเวอร์ ค่าคอนฟิกของเราเตอร์ สวิตช์ ผังการเชื่อมต่อของอุปกรณ์ต่าง ๆ เป็นต้น

การจัดการความผิดพลาดที่เกิดขึ้นบนเครือข่าย (Fault management) คือ การเฝ้าระวัง การเก็บล๊อคไฟล์ (Log file) การแจ้งเตือน การตรวจสอบและการแก้ไขข้อผิดพลาดต่างๆ ที่เกิดขึ้นในเครือข่าย การบริหารข้อผิดพลาดนั้นจะเน้นที่การแก้ปัญหา หรือข้อผิดพลาดของเครือข่ายให้ได้ทันเวลา เช่น สายสัญญาณขาด สวิตซ์หรือเราเตอร์ชำรุด เป็นต้น

การบริหารประสิทธิภาพ (Performance Management) คือ การบริหารให้อุปกรณ์สามารถทำงานได้อย่างเต็มประสิทธิภาพและมีแบนด์วิธเพียงพอต่อการใช้งาน การบริหารประสิทธิภาพเกี่ยวข้องกับการเฝ้าระวัง การประเมิน วิเคราะห์ปริมาณการใช้ และอัตราส่งผ่านข้อมูลของอุปกรณ์เครือข่ายต่าง ๆ เช่น สายนำสัญญาณ ฮับ สวิตซ์ เราเตอร์ โฮสและไฟร์วอลล์ เป็นต้น ไปจนถึงเส้นทางข้อมูลผ่านอุปกรณ์เครือข่ายต่างๆ เพื่อให้การใช้แบนด์วิธและทรัพยากรอื่น ๆ มีประสิทธิภาพการทำงานสูงสุด

การจัดการระบบบัญชีของเครือข่าย (Accounting management) คือ การควบคุมการใช้งานทรัพยากรบนเครือข่าย เช่น การเก็บค่าบริการ จัดการบัญชีผู้ใช้ การกำหนดสิทธิและการควบคุมการเข้าถึงทรัพยากรต่างๆ เป็นต้น

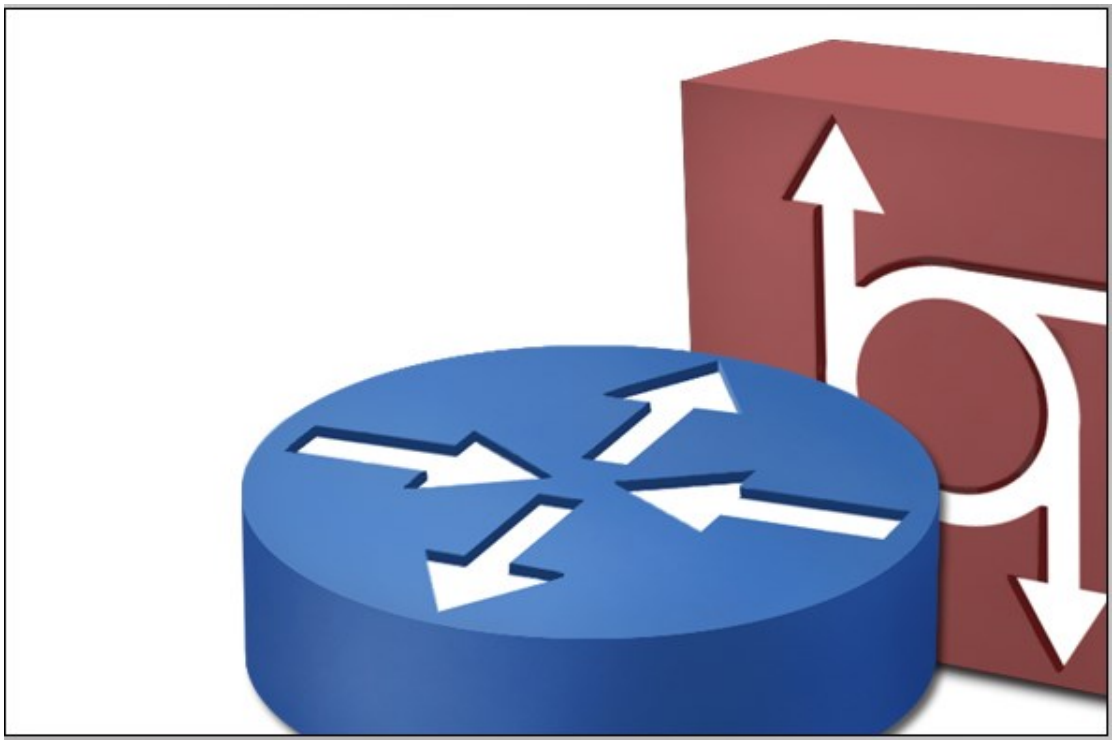
การจัดการระบบความปลอดภัยของเครือข่าย (Security management) คือ การควบคุมการเข้าใช้ทรัพยากรบนเครือข่ายให้เป็นไปตามนโยบายที่ได้กำหนดไว้

องค์ประกอบของการบริหารเครือข่ายทั้ง 5 ส่วนตามที่กล่าวมาแล้วข้างต้น จะแทรกอยู่ในเนื้อหาของบทต่าง ๆ ของเอกสารคำสอนนี้ ซึ่งเอกสารคำสอนเล่มนี้เขียนขึ้นตามพัฒนาการการเรียนรู้ของนิสิต โดยเริ่มตั้งแต่ความรู้พื้นฐานของระบบเครือข่าย การปรับแต่งเครือข่ายระดับต้น ระดับกลาง ไปจนถึงระดับสูงที่สามารถบริหารเครือข่ายได้จริง (เป็นวิธีการเขียนเอกสารคำสอนแบบบูรณาการ) ซึ่งคำอธิบายรายวิชาใน มคอ 3 จะแทรกตัวอยู่ตามบทต่างๆ โดยไม่ได้เรียงลำดับตามคำอธิบายรายวิชาที่ปรากฏใน มคอ 3

ภาคที่หนึ่ง

ทฤษฎีและการออกแบบเครือข่ายพื้นฐาน

บทที่ 1
มาตรฐานการเชื่อมต่อระบบเครือข่าย (Network Connectivity Standards)



- OSI model layers
- TCP/IP
- Device communications

แนวคิด

ปัจจุบันระบบสื่อสารและเครือข่ายเชื่อมโยงเข้ากันอย่างสลับซับซ้อนแบบใยแมงมุม ผลการเชื่อมโยงและเจริญก้าวหน้าของเทคโนโลยีทางด้านเครือข่าย ทำให้มีอิทธิพลต่อการดำรงชีวิตของเรา ในบทนี้จะกล่าวถึง โครงสร้างพื้นฐานในการสื่อสาร และมาตรฐานการสื่อสารข้อมูล ที่ถูกกำหนดโดยองค์กรมาตรฐานสากลหรือไอเอสโอ (ISO)

วัตถุประสงค์

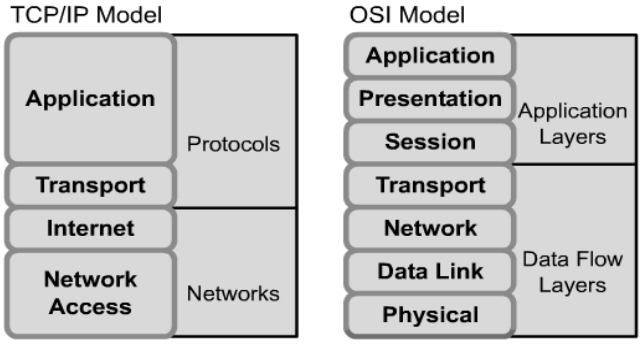
1. เพื่อให้เข้าใจความหมายและความรู้พื้นฐานที่เกี่ยวข้องกับระบบเครือข่าย
2. เพื่อให้เข้าใจลักษณะการสื่อสาร โครงสร้างพื้นฐานในการสื่อสาร และมาตรฐานการสื่อสารข้อมูล

1. มาตรฐานการเชื่อมต่อระบบเครือข่ายแบบโอเอสไอ (Open System Interconnection: OSI)

มาตรฐานการเชื่อมต่อแบบโอเอสไอได้แบ่งการสื่อสารออกเป็นชั้นย่อย ๆ จำนวน 7 ชั้น หรือ 7 เลเยอร์ [1, 19, 21] เหตุผลที่ทำให้ต้องมีการแบ่งออกเป็น 7 ชั้น เพื่อให้ง่ายต่อการกำหนดมาตรฐานการเชื่อมต่อและการอ้างอิงในแต่ละชั้น ซึ่งการกำหนดมาตรฐานดังกล่าวจะช่วยลดปัญหาในการสื่อสาร และการบริหารจัดการระบบเครือข่ายได้ง่ายขึ้น เช่น ผู้ผลิตอุปกรณ์เครือข่ายชนิดต่าง ๆ สามารถพัฒนาผลิตภัณฑ์ที่ตนเองมีความถนัดหรือเชี่ยวชาญได้อย่างเต็มที่ สำหรับการเชื่อมต่อกับอุปกรณ์อื่น ๆ ในลำดับชั้นเดียวกัน หรือชั้นอื่น ๆ ผู้ผลิตแต่ละรายจะต้องปฏิบัติตามข้อกำหนดการเชื่อมต่อระหว่างอุปกรณ์หรือระหว่างชั้นการสื่อสารโดยการอ้างอิงกับมาตรฐานโอเอสไออย่างเคร่งครัด นอกจากนี้มาตรฐานโอเอสไอยังช่วยให้ผู้พัฒนาซอฟต์แวร์ระบบสื่อสาร สามารถพัฒนาระบบโดยไม่จำเป็นต้องเริ่มต้นจากศูนย์เสมอไป หรือไม่มีความจำเป็นต้องพัฒนาซอฟต์แวร์สื่อสารให้ครบทุกองค์ประกอบตั้งแต่ลำดับชั้นที่ 1 ถึง 7 นั่นเอง

1.1 การเปรียบเทียบมาตรฐานการเชื่อมต่อโอเอสไอ (OSI Model) และทีซีพี/ไอพี (TCP/IP Model)

รูปแบบการเชื่อมต่อระบบเครือข่ายที่ได้รับความนิยมในปัจจุบันมี 2 แบบ คือ โอเอสไอ (OSI) และทีซีพี/ไอพี (TCP/IP) [1, 9, 23, 31] แสดงดังรูปที่ 1.1 สำหรับโอเอสไอนั้นจะแบ่งออกเป็น 7 ชั้น โดยรายละเอียดในแต่ละชั้นจะอธิบายในหัวข้อถัดไป และแบบทีซีพี/ไอพี จะมีจำนวนชั้นน้อยกว่าแบบโอเอสไอ คือ มีทั้งหมด 4 ชั้น โดยทั้งสองรูปแบบมีความแตกต่างกัน คือ แบบทีซีพี/ไอพีจะยุบรวมบางชั้นเข้าไว้ด้วยกันเพื่อความยืดหยุ่นในการใช้งาน (รูปที่ 1.2) รูปแบบการเชื่อมต่อแบบทีซีพี/ไอพี มีอิทธิพลต่อการเชื่อมต่อระบบเครือข่ายในปัจจุบันมากกว่าโอเอสไอ เนื่องทีซีพี/ไอพีถูกนำมาใช้งานจริงในปัจจุบัน

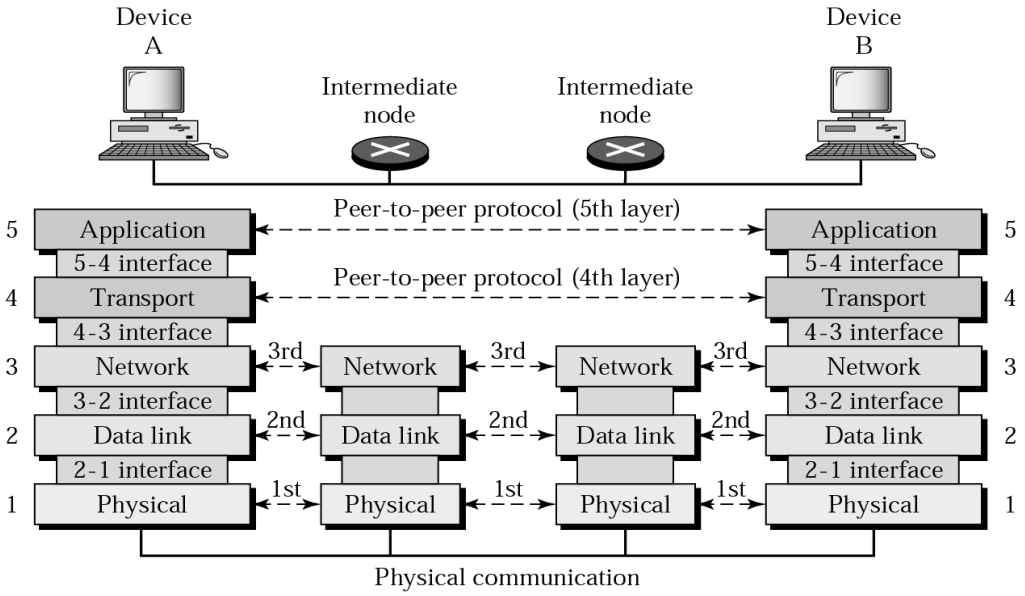


รูปที่ 1.1 โอเอสไอและทีซีพี/ไอพี

โดยหลักการแล้วแต่ละชั้นจะสื่อสารกับชั้นในระดับชั้นเดียวกันที่อยู่ต่างอุปกรณ์กัน (Peer-to-peer) ดังรูปที่ 1.3 เช่น การสื่อสารในระดับชั้นที่ 5 ของอุปกรณ์ A จะสามารถสื่อสารกับอุปกรณ์ B ในระดับชั้นที่ 5 เท่านั้น แบบจำลองโอเอสไอนั้น เป็นต้นแบบของสถาปัตยกรรมการสื่อสารที่เกิดขึ้นมาก่อน จากนั้นจึงเริ่มออกแบบและสร้างอุปกรณ์เครือข่ายในแต่ละชั้นตามมาในภายหลัง อุปกรณ์ที่ถูกสร้างขึ้นจะต้องสอดคล้องกับมาตรฐานชั้นใดชั้นหนึ่งของสถาปัตยกรรมนี้อย่างสมบูรณ์ อย่างไรก็ตาม ในโลกแห่งความเป็นจริงนั้นมีเทคโนโลยีจำนวนไม่น้อยที่ถูกพัฒนาขึ้นก่อนที่จะมีแบบจำลองนี้ และในบางกรณีเทคโนโลยีที่เกิดขึ้นภายหลังจากนี้ บางอย่างก็ไม่ได้ดำเนินตามแบบจำลองนี้อย่างสมบูรณ์เสมอไป แต่ผลที่ได้รับจากแบบจำลองโอเอสไอนี้ ก็ช่วยให้เกิดแรงผลักดันเพื่อนำไปสู่การพัฒนาระบบสื่อสารที่สามารถทำงานร่วมกันได้ของผู้ผลิตอุปกรณ์รายต่าง ๆ ได้เป็นอย่างดี จากรูปที่ 1.2 แสดงการเปรียบเทียบระหว่างมาตรฐานโอเอสไอและทีซีพี/ไอพี จากรูปแสดงให้เห็นว่าทีซีพี/ไอพี จะควบรวมชั้นการให้บริการระหว่างในชั้นกายภาพ (Physical) และดาตาลิงค์ (Data link) เข้าด้วยกัน และเรียกชั้นที่รวมเข้ากันใหม่นี้ว่าชั้นเน็ตเวิร์ค (Network) สืบเนื่องจากตัวแบบทีซีพี/ไอพี เล็งเห็นว่าการทำงานระหว่างชั้นกายภาพและชั้นดาตาลิงค์ มีความจำเป็นต้องทำงานร่วมกันอย่างใกล้ชิด ดังนั้นไม่ควรแยกชั้นทั้งสองออกจากกัน สำหรับชั้นเน็ตเวิร์ค (ชั้นที่ 3) ของทีซีพี/ไอพี จะถูกเปลี่ยนชื่อใหม่จากชั้นเน็ตเวิร์คเป็นชั้นอินเทอร์เน็ต (Internet) แทน ส่วนชั้นที่ 5, 6 และ 7 ของโอเอสไอ จะถูกมองว่าเป็นชั้นเดียวกันในตัวแบบของทีซีพี/ไอพี

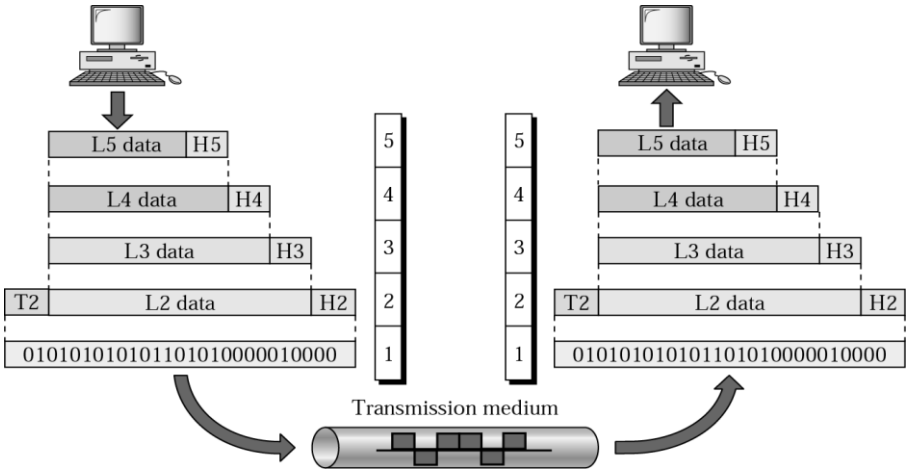
<table>
<tr><th>OSI</th><th>TCP/IP</th><th></th></tr>
<tr><td>Application</td><td rowspan="3">Application</td><td rowspan="3"></td></tr>
<tr><td>Presentation</td></tr>
<tr><td>Session</td></tr>
<tr><td>Transport</td><td>Transport</td><td>TCP, UDP</td></tr>
<tr><td>Network</td><td>Internet</td><td>IP</td></tr>
<tr><td>Data Link</td><td rowspan="2">Network Interface</td><td rowspan="2">Ethernet, Token Ring, Frame Relay, etc.</td></tr>
<tr><td>Physical</td></tr>
</table>

รูปที่ 1.2 การเปรียบเทียบการจัดลำดับชั้นของโอเอสไอและทีซีพี/ไอพี



รูปที่ 1.3 ตัวแบบโอเอสไอจะสื่อสารกันในชั้นที่ตรงกันเท่านั้น [1]

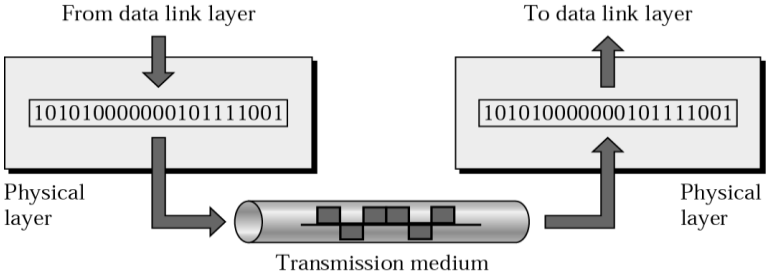
ข้อมูลในลำดับชั้นสื่อสารที่สูงกว่าจะถูกห่อหุ้ม (Data encapsulation) ด้วยข้อมูลของชั้นสื่อสารที่อยู่ในลำดับที่ต่ำกว่า จากตัวอย่างในรูปที่ 1.4 ข้อมูลของชั้นสื่อสารระดับที่ 5 (Application) ของฝั่งส่ง ซึ่งประกอบไปด้วยข้อมูลส่วนหัว (Header: H5) และเนื้อข้อมูล (Payload: L5 data) จะถูกห่อหุ้มด้วยข้อมูลในลำดับชั้นสื่อสารที่ 4 คือ H5 + L5 data = L4 data ซึ่งข้อมูล L4 data ก็จะเป็นเนื้อข้อมูล (Payload) ในลำดับชั้นการสื่อสารที่ 4 นั่นเอง วิธีการห่อหุ้มข้อมูลจะมีลักษณะเหมือนกันทุกชั้น ในทางตรงกันข้าม ฝั่งรับจะต้องทำการถอดข้อมูลที่ถูกห่อหุ้มในลักษะตรงกันข้ามกับทางฝั่งส่ง เช่น ข้อมูลสื่อสารในระดับชั้นที่ 1 ซึ่งเป็นสัญญาณดิจิทอลจะถูกแปลงข้อมูลให้อยู่ในรูปของเฟรมข้อมูลในระดับชั้นที่ 2 ก่อน จากนั้นฝั่งรับจะดำเนินการถอดข้อมูลส่วนหัวในชั้นที่ 2 ออก ก่อนส่งต่อไปเพื่อถอดข้อมูลในระดับชั้นที่ 3 ต่อไปเรื่อย ๆ จนกว่าจะเหลือเฉพาะข้อมูลที่ใช้สื่อสารจริง ๆ เท่านั้น



รูปที่ 1.4 ข้อมูลฝั่งผู้ส่งจะถูกห่อหุ้มตามลำดับชั้นส่วนผู้รับจะถอดออกตามลำดับชั้น [1]

ชั้นกายภาพ (Physical Layer)

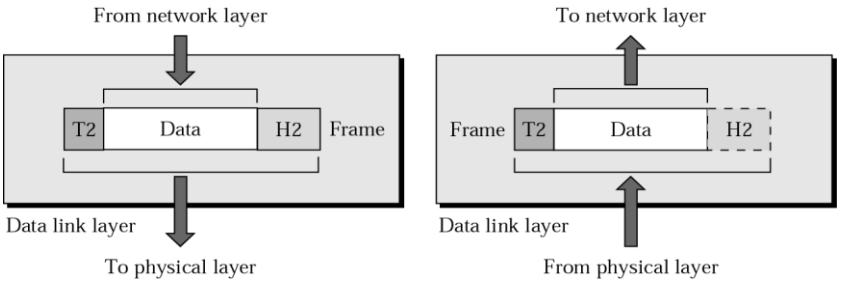
ชั้นกายภาพเป็นการสื่อสารระดับล่างสุดของตัวแบบโอเอสไอ [3, 15, 25] ซึ่งมีหน้าที่ให้บริการเกี่ยวกับการส่งข้อมูลในระดับบิต ระหว่างอุปกรณ์ 2 ชนิด เช่น อุปกรณ์ทางฝั่งของผู้ใช้งานกับอุปกรณ์ของฝั่งระบบเครือข่าย เป็นต้น ข้อมูลในระดับชั้นกายภาพจะถูกประกอบเข้าเป็นชุดข้อมูลเพื่อส่งต่อให้กับลำดับชั้นที่สูงกว่า (ชั้นดาตาลิงค์) เรียกว่าเฟรมข้อมูล นอกจากนั้นชั้นกายภาพยังมีหน้าที่รับผิดชอบดูแลในรายละเอียดการส่งข้อมูลกับฮาร์ดแวร์จริง เช่น การควบคุมการ์ดเน็ตเวิร์ค การส่งสัญญาณผ่านสายสัญญาณแบบต่าง ๆ เป็นต้น เมื่อกล่าวโดยสรุปแล้วชั้นกายภาพจะจัดการเกี่ยวกับสัญญาณทางไฟฟ้า สัญญาณเสียง หรือสัญญาณแสงที่จำเป็นต่อการสื่อสารโดยตรง ดังรูปที่ 1.5



รูปที่ 1.5 แสดงการทำงานของชั้นกายภาพ [1]

ชั้นดาต้าลิงค์ (Data link layer)

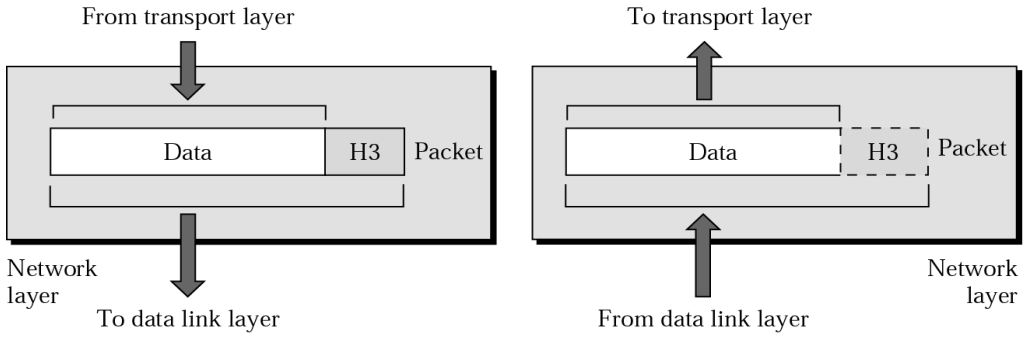
ดาตาลิงค์มีหน้าที่ดังนี้ คือ จัดเตรียมข้อมูลเกี่ยวกับการสื่อสารให้กับชั้นเน็ตเวิร์ค (รูปที่ 1.6) การควบคุมลำดับและอัตราการรับส่งข้อมูล (Flow Control) และบริหารจัดการชื่อที่อยู่ทางกายภาพของอุปกรณ์ในระบบเครือข่าย (Media Access Control Address: MAC) [15-17] เป็นต้น ข้อมูลที่ใช้รับส่งในระดับชั้นดาตาลิงค์เรียกว่า เฟรม (Frame) รูปแบบการรับส่งข้อมูลในชั้นนี้จะมีลักษณะเป็นแบบ Hop-to-Hop ดาตาลิงค์จะใช้วิธีการตรวจสอบความถูกต้องของข้อมูลด้วยวิธีการที่เรียกว่า Checksum คือ การคำนวณผลรวมของข้อมูลทั้งหมดและบันทึกลงในส่วนท้ายของเฟรมข้อมูล ปลายทางที่ได้รับเฟรมข้อมูลจะอาศัย Checksum เพื่อคำนวณว่าเฟรมข้อมูลที่ส่งมาดังกล่าวครบถ้วนหรือไม่ ถ้าไม่ครบก็จะแจ้งไปยังต้นทางว่าให้ส่งเฟรมข้อมูลมาใหม่อีกจนกว่าจะครบถ้วน สำหรับเทคโนโลยีที่ใช้ในชั้นดาตาลิงค์มีหลายแบบ เช่น Ethernet, Token Ring หรือ FDDI เป็นต้น



รูปที่ 1.6 แสดงการทำงานของชั้นดาตาลิงค์ [1]

ชั้นเน็ตเวิร์ค (Network layer)

ชั้นเน็ตเวิร์คมีหน้าที่ดังนี้ คือ ควบคุมการสื่อสารระหว่างจุดต่อจุดบนเครือข่าย คำนวณหาเส้นทางที่ดีที่สุดหรือสั้นที่สุด และการบริหารจัดการที่อยู่เสมือนของอุปกรณ์บนเครือข่าย (IP Address) [1, 32] เป็นต้น ข้อมูลที่ใช้สื่อสารในระดับชั้นเน็ตเวิร์คเรียกว่า แพ็กเก็ต (Packet) โดยแพ็กเก็ตนั้นถูกออกแบบมาเพื่อสนับสนุนการรับส่งข้อมูลข้ามเครือข่ายได้ง่ายขึ้น เนื่องจากในสถานการณ์จริงระบบเครือข่ายจะมีขนาดของแบนด์วิดท์ไม่เท่ากัน ดังนั้นแต่ละแพ็กเก็ตจะต้องมีขนาดน้อยกว่าหรือเท่ากับแบนด์วิดท์ที่เล็กที่สุดในระบบเครือข่ายที่สื่อสารกัน ชั้นเน็ตเวิร์คยังมีหน้าที่หลักสำคัญอีกประการ คือ การจัดเตรียมข้อมูลสำหรับชั้นสื่อสารทรานสปอร์ต (Transport Layer) และชั้นสื่อสารดาต้าลิงก์ด้วย โพรโทคอลที่นิยมใช้งานในชั้นเน็ตเวิร์ค เช่น IP และ IPX เป็นต้น การทำงานของชั้นเน็ตเวิร์คแสดงดังรูปที่ 1.7

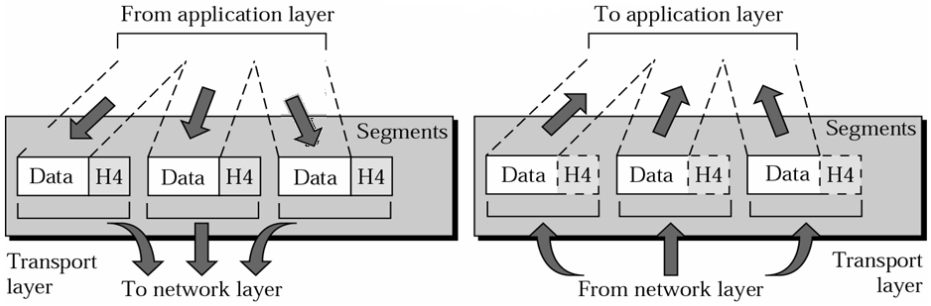


รูปที่ 1.7 แสดงการทำงานของชั้นสื่อสารเน็ตเวิร์ค [1]

ชั้นทรานสปอร์ต (Transport layer)

ทำหน้าที่ควบคุมการสื่อสารข้อมูลบนเครือข่ายให้เป็นไปด้วยความราบรื่นและข้อมูลครบถ้วน รวมไปถึงการกำหนดปริมาณการรับส่งข้อมูลให้เหมาะสมกับสภาพแวดล้อมการเชื่อมต่อเครือข่ายในขณะนั้น ๆ ด้วย ข้อมูลที่ใช้รับส่งในชั้นทรานสปอร์ตเรียกว่า เซกเมนต์ (Segment) [1, 19] การรับส่งข้อมูลเป็นแบบ Process-to-Process คือ การสื่อสารระหว่างโพรเซสในโปรแกรมประยุกต์ใด ๆ ที่ทำงานอยู่ต่างสถานที่กัน หรือกล่าวอีกนัยหนึ่งคือ อุปกรณ์บนเครือข่ายสามารถเชื่อมต่อได้พร้อม ๆ กันมากกว่า 1 ช่องทางนั้นเอง โดยอาศัยหมายเลขพอร์ต (Port number) ในการเชื่อมต่อ แสดงในรูปที่ 1.8 ชั้นสื่อสารทรานสปอร์ตมีความสามารถในการตรวจสอบความครบถ้วนของข้อมูล โดยอาศัยโพรโทคอลชื่อว่าทีซีพี (Transmission Control Protocol: TCP) และทำงานร่วมกับโพโทคอลไอพี

ในชั้นเน็ตเวิร์คอย่างใกล้ชิด ดังนั้นโพรโทคอลทั้งสองจะถูกจับคู่กันเสมอในตัวแบบทีซีพี/ไอพีนั่นเอง

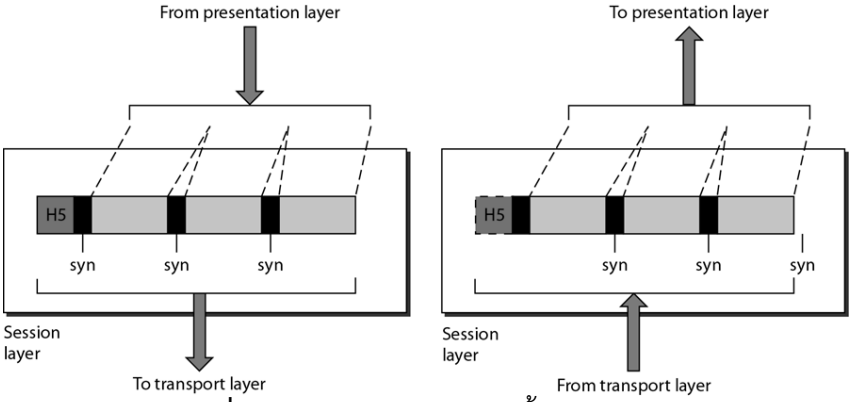


รูปที่ 1.8 แสดงการทำงานของชั้นทรานสปอร์ต [1]

ชั้นเซสชัน (Session layer)

เป็นชั้นสื่อสารที่จัดการเรื่องของการสร้าง "การเชื่อมต่อแต่ละครั้ง" ให้กับอุปกรณ์บนระบบเครือข่ายทั้งสองฝ่าย (ฝ่ายรับและฝ่ายส่ง) กล่าวคือ ชั้นเซสชันจะให้บริการแก่โปรแกรมประยุกต์ โดยทำหน้าที่ตั้งแต่เริ่มการเชื่อมต่อ ดูแลช่องสัญญาณการสื่อสารข้อมูลในแต่ละครั้ง ไปจนถึงยกเลิกการเชื่อมต่อเมื่อสิ้นสุดการสื่อสาร [1, 21] (ดังรูปที่ 1.9) ข้อมูลที่ใช้สื่อสารในระดับชั้นเซสชันเรียกว่าข้อความ (Message) นอกจากนั้นยังให้บริการด้านอื่น ๆ เช่น

- กำหนดเงื่อนไขการรับส่งข้อมูลชนิดผลัดกันรับส่ง (Haft-Duplex) และแบบทั้งสองทิศทางพร้อม ๆ กัน (Full-Duplex)
- กำหนดรูปแบบการสื่อสารระยะไกล เช่น กำหนดช่วงเวลาในการสื่อสารใหม่เมื่อเกิดการสื่อสารที่ผิดพลาดขึ้น
- รายงานข้อผิดพลาดเกี่ยวกับการสื่อสารให้กับชั้นโปรแกรมประยุกต์ให้ทราบ

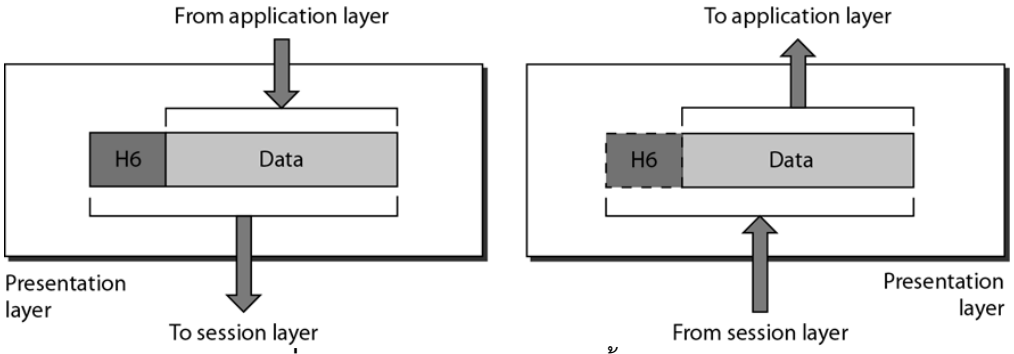


รูปที่ 1.9 แสดงการทำงานของชั้นเซสชัน [1]

ชั้นนำเสนอ (Presentation Layer)

หน้าที่หลัก คือ จัดรูปแบบและนำเสนอข้อมูลระหว่างการสื่อสาร ให้เป็นไปตามที่ต้องการ โดยมีการกำหนดรูปแบบการรับส่งข้อมูลสำหรับใช้ในการแลกเปลี่ยน [1, 33, 39] ทั้งนี้ยังรวมไปถึง

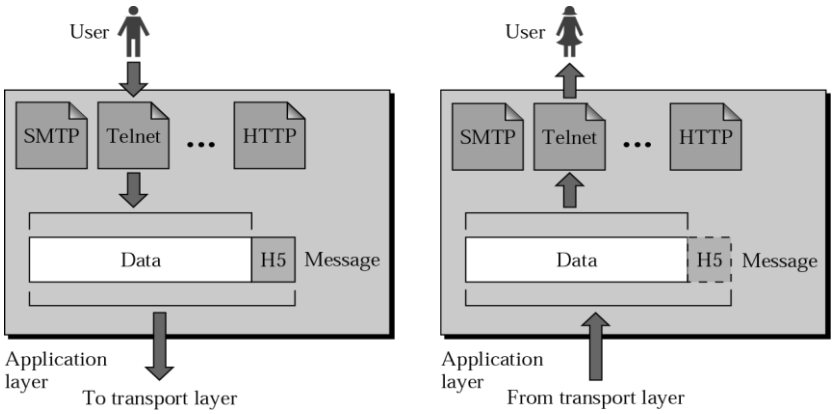
การแปลงข้อมูลให้อยู่ในรูปแบบรหัสมาตรฐาน เช่น ASCII หรือ EBCDIC ลดขนาดข้อมูล (Data compression) การเข้ารหัสหรือถอดรหัสข้อมูล (Data encryption/decryption) เพื่อความปลอดภัยในการสื่อสาร เป็นต้น ดังรูปที่ 1.10 ข้อมูลที่ใช้สื่อสารในระดับชั้นนำเสนอเรียกว่า ข้อความ (Message)



รูปที่ 1.10 แสดงการทำงานของชั้นนำเสนอ [1]

ชั้นแอพพลิเคชันหรือประยุกต์ (Application layer)

เป็นชั้นบนสุดของตัวแบบโอเอสไอ มีหน้าที่อำนวยความสะดวกในการติดต่อสื่อสารระหว่างโปรแกรมประยุกต์กับผู้ใช้งานให้เป็นไปตามที่ผู้ใช้ต้องการ [1, 33] ตัวอย่าง โปรแกรมประยุกต์ เช่น การรับส่งจดหมายอิเล็กทรอนิกส์ (E-mail) การโอนย้ายแฟ้มข้อมูลข้ามเครือข่าย (File transfer) การขอเข้าใช้ระบบคอมพิวเตอร์ในเครือข่าย (Host Terminal) การจัดแฟ้มข้อมูลในลักษณะต่าง ๆ เป็นต้น ดังรูปที่ 1.11 การใช้บริการในระดับโปรแกรมประยุกต์ โดยปกติจะใช้การพิมพ์คำสั่งต่าง ๆ ผ่านทางระบบปฏิบัติการ (Command line interface) หรือการใช้งานด้วยกราฟฟิก (Graphic user interface)



รูปที่ 1.11 แสดงการทำงานของชั้นประยุกต์

จากตารางที่ 1.1 แสดงหน้าที่ในแต่ละชั้นของตัวแบบโอเอสไอโดยสรุป ซึ่งตัวแบบทั้ง 2 ชนิด (โอเอสไอและทีซีพี/ไอพี) จะถูกใช้ในการอ้างอิงสำหรับการออกแบบโครงสร้างเครือข่ายในบทต่อ ๆ ไปเสมอ

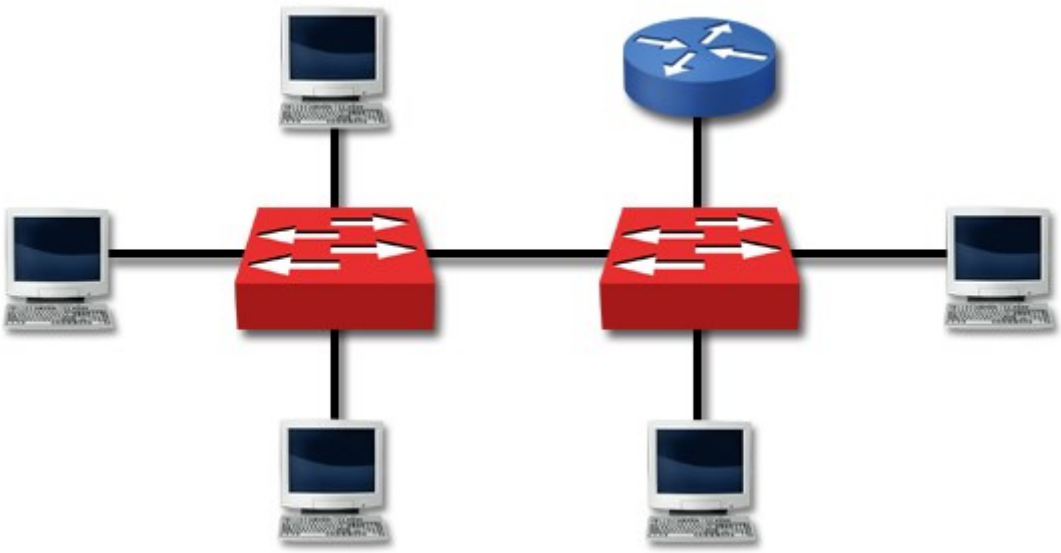
ตารางที่ 1.1 สรุปการทำงานของแต่ละเลเยอร์

OSI Layer	Functions
7 Application (Message)	ควบคุมการตอบสนองกับผู้ใช้งานและจัดเตรียมบริการต่าง ๆ ให้กับผู้ใช้งาน
6 Presentation (Message)	ควบคุมการแสดงผลข้อมูลและจัดเตรียมกลไกลเพื่อรองรับการเข้ารหัส การถอดรหัส การบีบอัดข้อมูล ข้อมูลจะถูกพิจารณาเป็นข้อความ
5 Session (Message)	ดูแลเกี่ยวกับจัดการแลกเปลี่ยนข้อมูล เช่น การสถาปนาการเชื่อมต่อและการยกเลิกการเชื่อมต่อ ข้อมูลจะถูกพิจารณาเป็นข้อความ
4 Transport (Segment)	การส่งข้อมูลเป็นแบบโปรเซสกับโปรเซส ควบคุมการทำงานให้โปรแกรมประยุกต์สามารถทำงานได้พร้อม ๆ กัน ข้อมูลจะถูกพิจารณาเป็นเซ็กเมนต์
3 Network (Packet)	เป็นการส่งข้อมูลแบบต้นทางไปยังปลายทาง กำหนดเส้นทางโดยใช้ที่อยู่เสมือนจริง (Addressing) ค้นหาเส้นทางที่ดีที่สุดสำหรับการสื่อสาร และพิจารณาข้อมูลในระดับแพ็กเก็ต
2 Data Link (Frame)	เป็นการเชื่อมต่อแบบจุดต่อจุด ควบคุมการไหลของข้อมูล ตรวจสอบข้อผิดพลาดของข้อมูล และพิจารณาข้อมูลในระดับเฟรม
1 Physical (Bit)	ไม่สนใจความถูกผิดของข้อมูล จะทำหน้าที่แปลงและรับส่งข้อมูลเป็น 0 และ 1 พิจารณาข้อมูลในระดับบิต

แบบฝึกหัดท้ายบท

1. โครงสร้างการทำงานของโมเดลแบบ OSI กับโมเดล TCP/IP มีลักษณะเหมือนหรือแตกต่างกันอย่างไร
2. การสื่อสารแบบ peer-to-peer มีลักษณะเป็นอย่างไร
3. ในโครงสร้างการทำงานของโมเดลแบบ OSI มีทั้งหมดกี่ชั้นและแต่ละชั้นทำหน้าที่อย่างไร
4. ข้อมูลที่ใช้สื่อสารในแต่ละชั้นของโมเดลแบบ OSI แตกต่างกัน มีชื่อเรียกว่าอะไรบ้างและแต่ละชนิดมีความแตกต่างกันอย่างไร

บทที่ 2
การออกแบบระบบเครือข่ายท้องถิ่น (Local Area Network Design)



- Ethernet
- Bridging and switching
- Routing
- LAN segmentation
- Using show commands

แนวคิด

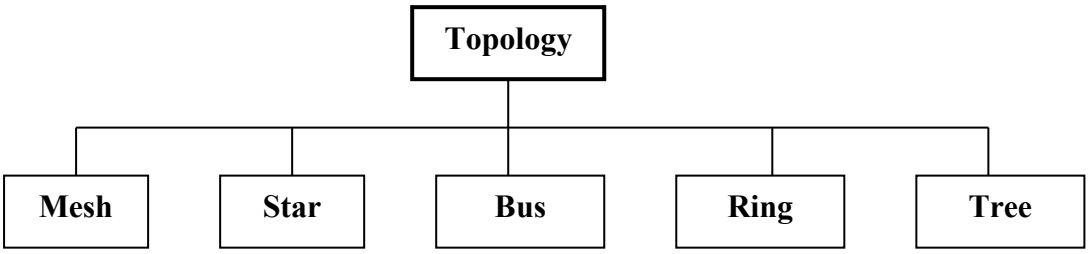
ก่อนการติดตั้งอุปกรณ์ระบบเครือข่ายจริงจำเป็นอย่างยิ่งที่จะต้องทำความเข้าใจเกี่ยวกับขั้นตอนการออกแบบ การวิเคราะห์โครงสร้างของระบบเครือข่ายรวมถึงเทคโนโลยีที่มีอยู่ในปัจจุบันได้อย่างเหมาะสม ซึ่งความรู้และความเข้าใจดังกล่าวจะนำไปสู่การสร้างระบบเครือข่ายที่เหมาะสมต่อองค์กรของตัวเองให้มากที่สุด

วัตถุประสงค์

1. เพื่อให้ทราบถึงขั้นตอนการวิเคราะห์และออกแบบระบบเครือข่ายในระดับท้องถิ่น
2. เพื่อให้ทราบถึงขั้นตอนการดำเนินงานสำหรับติดตั้งระบบเครือข่ายระดับท้องถิ่น

1. ประเภทของการเชื่อมต่อเครือข่าย (Categories of Topology)

ปัจจุบันการเชื่อมต่อที่นิยมและใช้งานมีอยู่ 5 ประเภท คือ Mesh, Start, Bus, Ring, Tree [1, 5, 18] ดังรูปที่ 2.1 ซึ่งการเชื่อมต่อแต่ละประเภทมีข้อดีข้อเสียต่างกันดังนี้



รูปที่ 2.1 ประเภทของการเชื่อมต่อเครือข่ายท้องถิ่น

1.1 การเชื่อมต่อแบบ Mash Topology

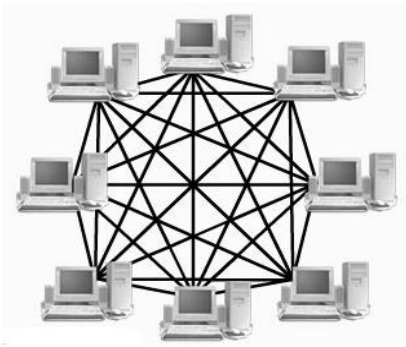
การเชื่อมต่อแบบนี้จะเชื่อมสายสัญญาณทุกเส้นถึงกันทั้งหมด ดังรูปที่ 2.2

ข้อดี

1. ถ้าเครื่องใดเครื่องหนึ่งไม่สามารถใช้งานได้จะไม่ส่งผลกระทบกับเครื่องอื่น ๆ
2. เมื่อต้องการส่งข้อมูลไม่จำเป็นต้องรอสามารถส่งข้อมูลได้ทันที
3. มีความน่าเชื่อถือสูง

ข้อเสีย

1. สิ้นเปลืองสายสัญญาณที่ใช้เชื่อมต่อเป็นอย่างมาก
2. ไม่สะดวกเมื่อต้องการย้ายสะถานที่ตั้งของเครื่องใหม่
3. สิ้นเปลืองพอร์ตสำหรับใช้เชื่อมต่อ เช่น ใช้เน็ตเวิร์คการ์ดมากกว่า 1 ใบ
4. ถ้าจำนวนสายสัญญาณมาก ๆ จะไม่สะดวกในการจัดให้เป็นระเบียบ



รูปที่ 2.2 การเชื่อมต่อแบบ Mesh topology

1.2 การเชื่อมต่อแบบ Star Topology

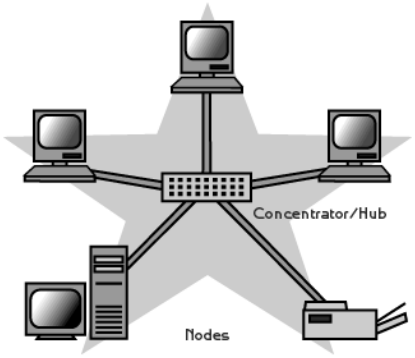
ลักษณะการเชื่อมต่อโครงสร้างแบบสตาร์ มีลักษณะเป็นแบบดาวกระจาย คือ จะมีอุปกรณ์ เช่น ฮับ หรือสวิตช์เป็นศูนย์กลาง ซึ่งการเชื่อมต่อลักษณะนี้มีประโยชน์คือ กรณีมีสายสัญญาณเส้นใดเส้นหนึ่งหลุดหรือเสียก็จะไม่มีผลกระทบต่อการทำงานของระบบ นอกจากนี้ถ้าหากมีการเพิ่มเครื่องคอมพิวเตอร์เข้าไปอีกในเครือข่ายก็สามารถทำงานได้ทันที การเชื่อมต่อแบบนี้เป็นที่นิยมมากในปัจจุบัน เนื่องจากอุปกรณ์ที่ใช้เป็นศูนย์กลาง เช่น ฮับหรือสวิตช์ มีราคาถูกลงอย่างมาก ในขณะที่ประสิทธิภาพการทำงานเพิ่มสูงขึ้นเรื่อย ๆ จนในปัจจุบันอุปกรณ์ดังกล่าวมีความเร็วในระดับกิกะบิตแล้ว ดังรูปที่ 2.3

ข้อดี

1. ถ้าเครื่องใดเครื่องหนึ่งไม่สามารถใช้งานได้จะไม่ส่งผลกระทบกับเครื่องอื่น ๆ
2. การเชื่อมต่อทำได้ง่ายและสะดวก
3. จำนวนเส้นของสัญญาณใช้เท่ากับจำนวนของเครื่องที่ทำการเชื่อมต่อ (น้อยกว่า Mesh)
4. ปรับปรุงได้ง่ายและอุปกรณ์มีราคาถูก

ข้อเสีย

1. ถ้าจุดที่รวมศูนย์เช่น ฮับ เสียหายจะส่งผลกระทบต่อทุก ๆ เครื่องที่เชื่อมต่อด้วย
2. การส่งข้อมูลต้องผลัดกันส่ง ถ้าสัญญาณไม่ว่างจะต้องเสียเวลาในการคอย
3. เมื่อปริมาณข้อมูลเพิ่มขึ้นถึงระดับหนึ่งจะทำให้เกิดคอขวด

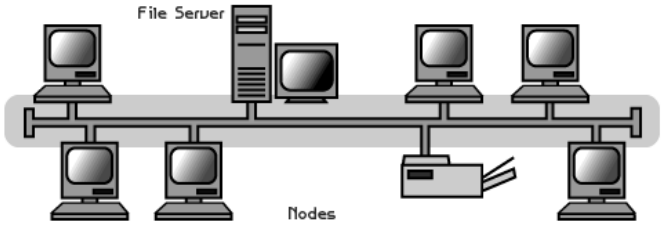


รูปที่ 2.3 การเชื่อมต่อแบบ Star topology

1.3 การเชื่อมต่อแบบ Bus Topology

ลักษณะการเชื่อมต่อแบบบัสจะมีลักษณะเป็นแบบอนุกรม โดยใช้สายเคเบิลหรือสายสัญญาณาณเพียงเส้นเดียว เชื่อมต่อกันไปตลอดเส้นทาง ทำให้โครงสร้างแบบนี้มีจุดอ่อนก็ คือ เมื่อคอมพิวเตอร์ตัวใดตัวหนึ่งเกิดปัญหา ก็จะทำให้คอมพิวเตอร์ทั้งระบบประสบปัญหาไปด้วย ข้อดีของโครงสร้างแบบนี้ก็คือ ไม่จำเป็นต้องมีอุปกรณ์ที่ทำหน้าที่เป็นศูนย์กลางควบคุมการทำงาน อย่างเช่น ฮับหรือสวิตช์ ใช้สายสัญญาณเพียงเส้นเดียวก็เพียงพอ โครงสร้างแบบนี้เหมาะกับเครือข่ายที่มีขนาดเล็กและมีจำนวนเครื่องคอมพิวเตอร์ในปริมาณไม่มาก ในปัจจุบันไม่นิยมใช้กันแล้ว เนื่องจาก

ไม่ได้มีการพัฒนาความสามารถเพิ่มเติม ส่วนความเร็วในการรับส่งข้อมูลได้ถึง 10 เมกกะบิตต่อวินาที (Mbps) ดังรูปที่ 2.4



รูปที่ 2.4 การเชื่อมต่อแบบ Bus topology

ข้อดี

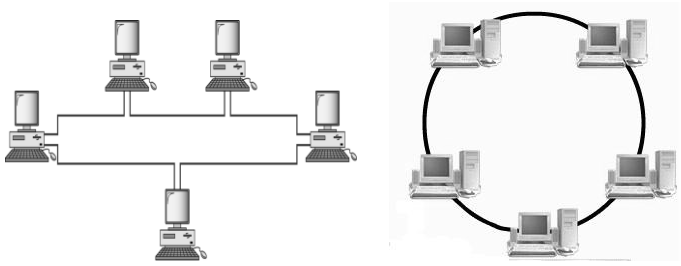
1. ปริมาณการส่งข้อมูลทำได้สูงเพราะสายสัญญาณหลักเป็นสายประเภทป้องกันสัญญาณรบกวน
2. ใช้สายนำสัญญาณไม่มาก

ข้อเสีย

1. ข้อมูลที่ส่งและรับจะผ่านไปยังทุก ๆ เครื่อง ซึ่งทำให้ประสิทธิภาพโดยรวมเสียไป
2. ต้องใช้อุปกรณ์เชื่อมต่อค่อนข้างมาก ทำให้โอกาสเกิดความผิดพลาดได้ง่ายกว่าแบบอื่น
3. เมื่อเครื่องใดเครื่องหนึ่งเสียหายหรือการเชื่อมต่อไม่สมบูรณ์จะส่งผลกระทบต่อเครื่องอื่น ๆ ทั้งหมด
4. อุปกรณ์ค่อนข้างมีราคาแพง เชื่อมต่อยาก และไม่เป็นที่นิยมในปัจจุบัน

1.4 การเชื่อมต่อแบบ Ring Topology

ลักษณะการเชื่อมต่อจะเป็นแบบวงแหวน ดังรูปที่ 2.5 การส่งข้อมูลจะเป็นแบบทิศทางเดียว ซึ่งถ้าข้อมูลที่ส่งออกไปแล้วไม่ตรงกับคอมพิวเตอร์เครื่องรับตามที่เครื่องต้นทางระบุมา ข้อมูลจะถูกส่งต่อไปยังเครื่องถัดไป จนกว่าจะถึงเครื่องปลายทางที่ระบุไว้ จุดอ่อนของการเชื่อมต่อมีลักษณะคล้าย ๆ แบบบัส เมื่อสายนำสัญญาณขาดจุดใดจุดหนึ่งระบบจะหยุดทำงานทันที ปัจจุบันการเชื่อมต่อแบบริงยังมีการใช้งานอยู่บ้าง โดยส่วนใหญ่ใช้เชื่อมต่อแบบวงแหวน 2 เส้น เพื่อใช้เป็นเส้นทางสำรองและนิยมเชื่อมต่อเพื่อทำหน้าที่เป็นเครือข่ายหลักของระบบ (Backbone) ด้วย



รูปที่ 2.5 การเชื่อมต่อแบบ Ring topology

ข้อดี

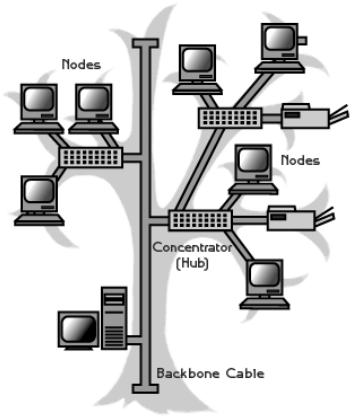
1. ใช้สายนำสัญญาณไม่มาก
2. การส่งข้อมูลจะไม่ชนกันเนื่องจากจะใช้ Token ควบคุมจังหวะของการส่งข้อมูลแบบเป็นลำดับ เครื่องที่ได้ Token เท่านั้นจึงจะส่งข้อมูลได้

ข้อเสีย

1. ถ้าเครื่องคอมพิวเตอร์เครื่องหรือสายนำสัญญาณในระบบเกิดปัญหาจะทำให้ระบบไม่สามารถทำงานต่อไปได้
2. ความรวดเร็วในการส่งข้อมูลไม่มีประสิทธิภาพเพราะต้องได้รับ Token ก่อนจึงสามารถส่งข้อมูลออกไปได้

1.5 การเชื่อมต่อแบบ Tree Topology

โครงสร้างการเชื่อมต่อแบบทรีเป็นแบบสุดท้ายของโครงสร้างเครือข่ายที่ได้รับความนิยม โดยลักษณะโครงข่ายแบบนี้ คือ การนำเครือข่ายย่อย ๆ ที่มีโครงข่ายตามแบบที่กล่าวไว้ข้างต้นทั้ง 4 แบบมารวมกันหรือเชื่อมต่อกันให้มีขนาดใหญ่ขึ้น เช่น เครือข่ายที่ผสมผสานจากการนำเอาเครือข่ายที่มีโครงสร้างแบบบัสและแบบสตาร์มาผสมกัน ดังรูปที่ 2.6



รูปที่ 2.6 การเชื่อมต่อแบบ Tree topology

ข้อดี

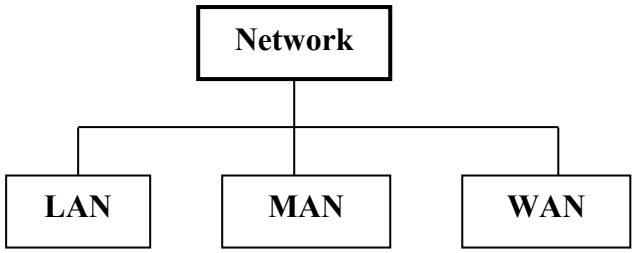
1. ผสมผสานการเชื่อมต่อเครือข่ายหลาย ๆ ประเภทเข้าด้วยกันเพื่อให้เหมาะสมกับสภาพแวดล้อมจริงที่ทำงานอยู่
2. ความเร็วในการส่งข้อมูลใกล้เคียงกับความเป็นจริง คือ โครงข่ายหลักจะมีขนาดของแบนวิดธ์มาก ส่วนเชื่อมต่อกับผู้ใช้ก็ขึ้นอยู่กับโทโพโลยีที่ใช้งาน
3. การปรับปรุงโครงข่ายสามารถทำได้ง่าย

ข้อเสีย

1. โครงข่ายหลักส่วนมากต้องรองรับความเร็วที่สูง ดังนั้นราคาการเชื่อมต่อโครงสร้างเครือข่ายจึงสูงและต้องใช้ความชำนาญในการติดตั้งด้วย
2. เมื่อปริมาณข้อมูลสูงถึงจุดที่โครงข่ายหลักไม่สามารถรองรับได้ เครือข่ายจะเกิดปัญหาคอขวดของเครือข่าย (Bottleneck network)
3. กรณีของโครงข่ายหลักเสียหายจะทำให้ระบบทั้งหมดหยุดทำงานทันที แต่สามารถแก้ไขด้วยการสร้างสายนำสัญญาณสำรองไว้อีกชุดหนึ่ง
4. เมื่อโทโพลียีที่ใช้เชื่อมต่อมีความหลากหลายมาก จะทำให้การบริหารจัดการเครือข่ายยุ่งยากเพิ่มขึ้น

2. ประเภทของระบบเครือข่าย (Categories of Network)

โดยหลัก ๆ แล้วแบ่งออกเป็น 3 ประเภท คือ Local Area Network (LAN), Metropolitan Area Network (MAN), Wide Area Network (WAN) [1, 17, 18] ดังรูปที่ 2.7 โดยขึ้นอยู่กับขนาดของเครือข่าย เช่น เครือข่ายที่มีความเร็วสูงและมีบริเวณไม่กว้างจะเป็นชนิด LAN ส่วนเครือข่ายที่ต้องสื่อสารกันระหว่างเมืองใหญ่ ๆ จะเป็นลักษณะของ MAN ส่วน WAN จะใช้เชื่อมต่อผ่าน ISP หรือระหว่างประเทศซึ่งจะมีความเร็วในการส่งข้อมูลที่ต่ำที่สุด

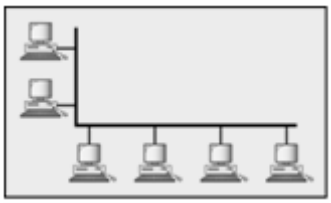


รูปที่ 2.7 ประเภทของระบบเครือข่าย

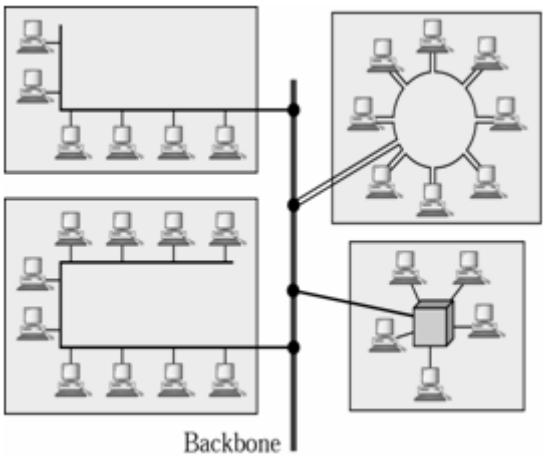
2.1 เครือข่ายท้องถิ่น (LAN)

เครือข่ายท้องถิ่นมีขอบเขตการเชื่อมต่อแคบ เช่น ภายในอาคาร ออฟฟิศ สำนักงาน หรืออาคารที่อยู่ติด ๆ กัน ระยะทางไม่ควรเกิน 2,000 ฟุต เครือข่ายท้องถิ่นได้รับความนิยมมากในการเชื่อมต่ออุปกรณ์สำนักงานเข้าด้วยกัน เช่น คอมพิวเตอร์ตั้งโต๊ะ คอมพิวเตอร์โน้ตบุ๊ค เครื่องพิมพ์งานและอุปกรณ์ในสำนักงาน เป็นต้น โดยอาศัยโทโพโลยีในการเชื่อมต่อ เช่น บัส ริง สตาร์ หรือหลาย ๆ แบบทำงานร่วมกัน ดังรูปที่ 2.8 และ 2.9 สำหรับความเร็วในการสื่อสารข้อมูลในเครือข่ายแบบท้องถิ่นสูงมาก ปัจจุบันข้อมูลสามารถสื่อสารด้วยความเร็วระดับ 40 กิกกะบิตต่อวินาที (Gbps) สำหรับเครือข่ายชนิดใช้สายนำสัญญาณ และความเร็ว 1 กิกกะบิตสำหรับเครือข่ายชนิดไร้สาย

(Wireless) เครือข่ายท้องถิ่นทำหน้าที่รวบรวมอุปกรณ์สื่อสารของผู้ใช้ทั้งหมดเข้าไว้ด้วยกันรวมถึงเครือข่ายย่อย ๆ ด้วย โดยมีโครงข่ายหลักที่ผสานเครือข่ายย่อย ๆ เข้าไว้ด้วยกันเรียกว่า แกนหลักเครือข่าย (Backbone network) การสื่อสารข้อมูลภายในเครือข่าย LAN สามารถสื่อสารได้ทันทีตราบใดที่สายนำสัญญาณว่าง แต่เมื่อมีอุปกรณ์ใดในเครือข่ายต้องการสื่อสารข้ามเครือข่าย ข้อมูลจำเป็นต้องถูกส่งออกไปตรงตำแหน่งทางเข้าออกของระบบเครือข่าย ตำแหน่งดังกล่าว เรียกว่าเกตเวย์ (Gateway) ของเครือข่าย



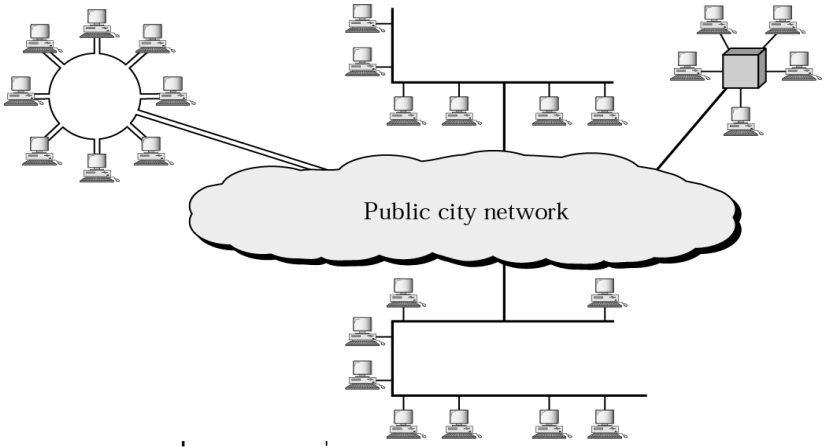
รูปที่ 2.8 การเชื่อมต่อ LAN เพียงโทโพโลยีเดียว



รูปที่ 2.9 การเชื่อมต่อ LAN หลาย ๆ topology เข้าด้วยกัน

2.2 เครือข่ายเมือง (MAN)

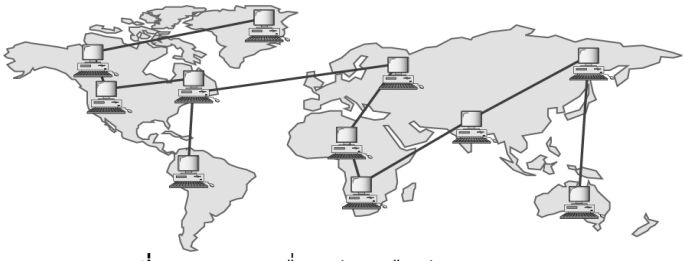
โดยพื้นฐานแล้วระบบเครือข่ายเมือง (MAN) มีลักษณะคล้ายกับระบบเครือข่ายท้องถิ่น แต่มีอาณาเขตที่กว้างไกลกว่า คือ มีขอบเขตตั้งแต่ภายในเมืองเดียวกันหรือหลาย ๆ เมืองที่อยู่ติดกันก็ได้ เช่น ระบบเครือข่ายที่เชื่อมต่อภายในจังหวัด เป็นต้น โดยปกติการเชื่อมต่อเครือข่ายดังกล่าวจะอาศัยระบบบริการเครือข่ายสาธารณะ เช่น โครงข่ายโทรศัพท์ เป็นต้น จึงเป็นเครือข่ายที่ใช้กับองค์กรที่มีสำนักงานห่างไกลและต้องการเชื่อมสำนักงานเหล่านั้นเข้าด้วยกัน เช่น ธนาคาร เครือข่ายสถาบันการศึกษา บริษัทเอกชน เป็นต้น เครือข่ายระดับเมืองจะเชื่อมโยงเครือข่ายในระยะทางที่ไกลมาก ดังนั้นความเร็วในการสื่อสารจึงไม่สูงเท่ากับเครือข่ายท้องถิ่น เนื่องจากมีสัญญาณรบกวนมาก เทคโนโลยีที่ใช้เชื่อมเครือข่ายเมืองมีความหลากหลาย เช่น ผ่านระบบสัญญาณดาวเทียม เส้นใยแก้วนำแสง คลื่นไมโครเวฟ คลื่นวิทยุ สายเคเบิล เป็นต้น ดังรูปที่ 2.10



รูปที่ 2.10 การเชื่อมต่อแบบเครือข่ายเมือง (MAN)

2.3 เครือข่ายบริเวณกว้าง (WAN)

เป็นระบบที่มีขอบเขตการใช้งานกว้างไกลกว่าระบบเมือง ซึ่งอาจกล่าวได้ว่าเป็นระบบที่ไร้ขอบเขต เช่น ระบบการสื่อสารข้อมูลผ่านดาวเทียมของสถานีโทรทัศน์ต่าง ๆ แต่การที่จะเชื่อมต่อเครือข่ายที่มีระยะทางห่างกันมาก ๆ ให้เป็นเครือข่ายเดียวกันทั้งหมดนั้น จำเป็นต้องอาศัยเครือข่ายสาธารณะ (Public Networks) ที่ให้บริการการสื่อสาร โดยเชื่อมต่อกับโมเด็มผ่านเครือข่ายโทรศัพท์สาธารณะ (Public Switching Telephone Network: PSTN) ซึ่งมีทั้งลักษณะที่ต้องมีการเชื่อมต่อก่อน (Dial-up) หรือเชื่อมต่อแบบตายตัว เช่น สายเช่า (Lease Line) ดังรูปที่ 2.11 ปริมาณข้อมูลที่สื่อสารบนระบบเครือข่ายบริเวณกว้างมีความเร็วต่ำกว่าระดับเมืองและมีต้นทุนในการเชื่อมต่อสูง



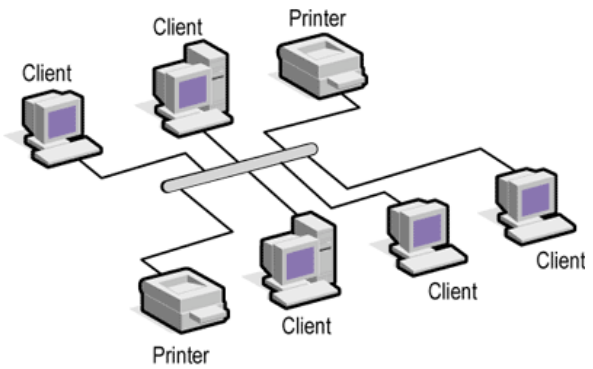
รูปที่ 2.11 การเชื่อมต่อเครือข่ายแบบ WAN

3. วิธีการเชื่อมเครือข่ายแบบท้องถิ่น

ในหัวข้อนี้จะกล่าวถึงวิธีการเชื่อมต่อระบบเครือข่ายท้องถิ่นในทางปฏิบัติ โดยปกตินิยมเรียกว่า "การเชื่อมต่อแลน" ซึ่งรูปแบบการเชื่อมต่อเครือข่ายในปัจจุบันมีอยู่ 3 ประเภท คือ การเชื่อมต่อแบบบัส วงแหวน และแบบสตาร์ ในทางปฏิบัติการเชื่อมต่อเครือข่ายทั้ง 3 วิธี จะใช้อุปกรณ์และวิธีการที่แตกต่างกัน โดยพิจารณาเลือกจากข้อดีข้อเสียหรือจากความเหมาะสมกับงานแต่ละประเภท ปัจจุบันพบว่าการเชื่อมต่อแบบสตาร์นิยมใช้งานมากที่สุด เพราะสามารถตรวจสอบหาข้อผิดพลาดได้ง่าย ในตัวอย่างเกือบทั้งหมดในเอกสารคำสอนนี้จะกล่าวถึงการเชื่อมต่อแบบสตาร์เป็นหลัก สำหรับการเชื่อมต่ออีก 2 แบบ จะมีการใช้งานอยู่บ้างแต่ไม่มากนัก โดยมีรายละเอียดดังต่อไปนี้

3.1 การติดตั้งเครือข่ายแบบบัส

วิธีการเชื่อมต่อแบบนี้มีลักษณะเหมือนกับการสร้างถนนสายหลักแล้วมีซอยแยกจากถนนหลักแตกเขนงไปเรื่อยตามจำนวนผู้ใช้งาน โดยเริ่มต้นจากการวางสายสัญญาณเป็นแกนหลักเรียกว่าบัสหลักหรือแบ็คโบน (Backbone) จากนั้นเชื่อมสายสัญญาณรองจากแกนหลักไปยังเครื่องคอมพิวเตอร์ตามจุดต่าง ๆ โดยที่ปลายของสายสัญญาณหลักทั้งสองข้างจะมีเทอร์มิเนเตอร์ (Terminator) ปิดอยู่ เพื่อให้สัญญาณไฟฟ้าครบวงจร [22, 23] ดังรูปที่ 2.12



รูปที่ 2.12 การเชื่อมต่อแบบบัส

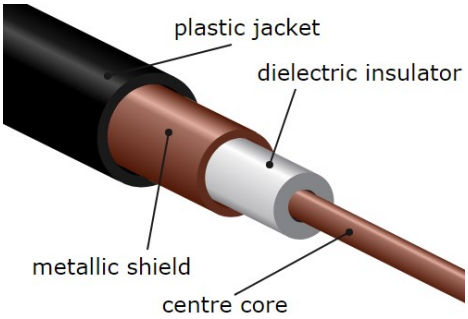
สายสัญญาณที่ใช้ในการเชื่อมต่อแบบบัสมีอยู่สองชนิด คือ

1) สายโคแอ็กซ์แบบบาง (Thin Coaxial Cable) เป็นสายที่มีขนาดเล็ก เส้นผ่านศูนย์กลางประมาณ 0.64 เซนติเมตร (รูปที่ 2.13) เนื่องจากสายประเภทนี้มีขนาดเล็กและมีความยืดหยุ่นสูง จึงสามารถใช้ได้กับการติดตั้งเครือข่ายเกือบทุกประเภท สายประเภทนี้สามารถนำสัญญาณไฟฟ้าได้ไกลถึง 185 เมตร สายโคแอ็กซ์แบบบางอยู่ในประเภท RG-58 ซึ่งจะมีความต้านทานในสาย (Impedance) ที่ 50 โอห์ม สายประเภทนี้จะมีแกนกลางอยู่ 2 ลักษณะคือ แบบที่เป็นสายทองแดงเส้นเดียวและแบบที่เป็นใยโลหะหลายเส้น

รูปที่ 2.13 สายโคแอ็กซ์แบบบาง

2) สายโคแอ็กซ์แบบหนา (Thick Coaxial Cable) เป็นสายที่ค่อนข้างแข็งและขนาดใหญ่กว่าสายโคแอ็กซ์แบบบาง โดยมีเส้นผ่านศูนย์กลางประมาณ 1.27 เซนติเมตร (รูปที่ 2.14) สายชนิดนี้เป็นสายนำสัญญาณประเภทแรกที่ใช้งานกับเครือข่ายแบบอีเธอร์เน็ต (Ethernet) ส่วน

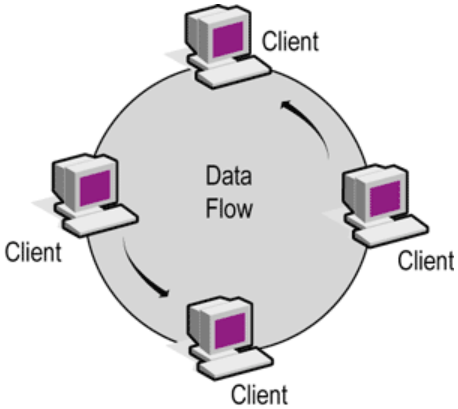
แกนกลางที่เป็นสายทองแดงจะมีขนาดใหญ่ ดังนั้นสายโคแอ็กซ์แบบหนานี้จึงสามารถนำสัญญาณได้ไกลกว่าแบบบาง โดยปกติประมาณ 500 เมตร ด้วยความสามารถนี้สายโคแอ็กซ์แบบหนาจึงนิยมใช้ ในการเชื่อมต่อเส้นทางหลักของข้อมูล หรือแบ็คโบนของเครือข่ายสมัยเริ่มแรก ๆ แต่ปัจจุบันได้ยกเลิกใช้สายโคแอ็กซ์แล้ว สายสัญญาณที่นิยมใช้ทำเป็นแบ็คโบน คือ สายใยแก้วนำแสง ซึ่งจะได้กล่าวในรายละเอียดในส่วนต่อไป



รูปที่ 2.14 สายโคแอ็กซ์แบบหนา

โคแอ็กซ์แบบบางนิยมติดตั้งภายในอาคาร ส่วนสายชนิดหนาจะใช้ติดตั้งระหว่างอาคารหรือเชื่อมระหว่างชั้นต่าง ๆ ข้อดีของการเดินสายแบบบัสนี้ คือ ติดตั้งง่าย อุปกรณ์ราคาถูก ไม่ต้องใช้อุปกรณ์รวมสัญญาณ (ฮับหรือสวิตช์) ระยะติดตั้งไกล และสามารถติดตั้งอุปกรณ์ทวนสัญญา (Repeater) เพื่อเพิ่มระยะการติดตั้งสายได้ แต่มีข้อเสียคือ ถ้าเกิดข้อผิดพลาดในสายสัญญาณ หรือเกิดการชำรุดที่จุดหนึ่งจุดใดบนบัส จะทำให้เครือข่ายทั้งระบบไม่สามารถใช้งานได้ และการตรวจสอบหาจุดเสียทำได้ยาก ภายหลังจึงไม่นิยมติดตั้งสายแบบบัสมากนัก

3.2 การติดตั้งเครือข่ายแบบวงแหวนหรือริง



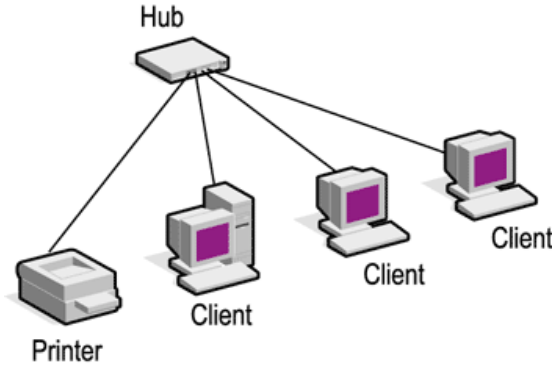
รูปที่ 2.15 การเชื่อมต่อแบบวงแหวน

การติดตั้งเครือข่ายแบบริงหรือวงแหวนนี้ มีลักษณะทางกายภาพเป็นวงกลม [22, 34] โดยเชื่อมจากเครื่องแรกไปยังเครื่องสุดท้ายและวนกลับมายังเครื่องแรกอีกครั้ง ดังรูปที่ 2.15 วิธีการเดินสายสัญญาณแบบวงแหวนนี้ ปรากฏในระบบเครือข่ายมานานแล้ว แต่ปัจจุบันไม่เป็นที่นิยมนัก เนื่องจากมีข้อด้อยหลายประการ (ตามที่กล่าวมาแล้วในหัวข้อประเภทของการเชื่อมต่อเครือข่าย) เช่น จุดใดจุดหนึ่งในวงแหวนเสียหายจะทำให้เครื่องอื่น ๆ ไม่สามารถส่งข้อมูลได้ และอุปกรณ์มีราคา

ค่อนข้างสูง แต่อย่างไรก็ตาม การเชื่อมต่อด้วยวิธีการนี้นิยมออกแบบให้เป็นเครือข่ายสำรอง โดยการออกแบบให้เป็นลักษณะวงแหวนซ้อนกันสองเส้นทาง

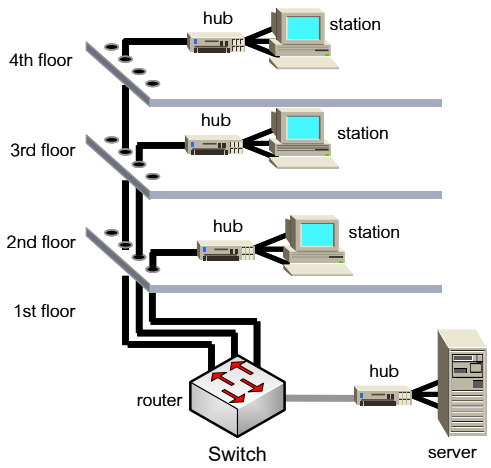
3.3 การติดตั้งเครือข่ายแบบสตาร์

การเชื่อมต่อเครือข่ายในลักษณะสตาร์นั้นสามารถพบเห็นได้โดยทั่วไป โดยรูปแบบการเชื่อมต่อนั้นจะมีลักษณะคล้ายรูปดาว ตัวอย่างเช่น เครื่องคอมพิวเตอร์ลูกข่ายแต่ละเครื่องจะใช้สายสัญญาณเชื่อมไปที่ฮับหรือสวิตช์ซึ่งเป็นศูนย์กลางในการเชื่อมต่อ [22, 34] ดังรูปที่ 2.15 โดยจำนวนของเครื่องลูกข่ายจะขึ้นอยู่กับจำนวนพอร์ตของฮับหรือสวิตช์ ซึ่งโดยปกติจะมีให้เลือกใช้งานตั้งแต่ 8, 16, 24, 32 และ 48 พอร์ต เป็นต้น



รูปที่ 2.15 การเชื่อมต่อเครือข่ายแบบสตาร์

ลักษณะการเชื่อมสายสัญญาณแต่ละเครื่องไปยังฮับหรือสวิตช์แบบสตาร์มีข้อดี คือ ถ้าสายสัญญาณเส้นหนึ่งเส้นใดขาด ก็จะไม่ส่งผลกระทบต่อเครื่องคอมพิวเตอร์อื่น ๆ ทำให้การบำรุงรักษาและแก้ไขปัญหาทำได้ง่าย ปัจจุบันการเชื่อมต่อในลักษณะนี้สามารถรับส่งข้อมูลด้วยความเร็วตั้งแต่ 10 เมกกะบิตต่อวินาที ไปจนถึง 40 กิกะบิตต่อวินาที ในขณะที่ราคาของอุปกรณ์ถูกลงเรื่อย ๆ ทำให้ได้รับความนิยมอย่างสูง การเชื่อมต่อแบบสตาร์นี้สามารถวางตำแหน่งของเครื่องคอมพิวเตอร์และเชื่อมสายสัญญาณอย่างไรก็ได้ โดยไม่จำเป็นต้องวางให้เรียงตามลำดับอย่างการเดินสายแบบบัสหรือแบบวงแหวน สายสัญญาณแบบสตาร์แต่ละเส้นมีความยาวได้ไม่เกิน 100 เมตร แต่ในทางปฏิบัติความยาวของสายสัญญาณไม่ควรเกิน 85 เมตร สำหรับตัวอย่างการเชื่อมต่อเครือข่ายแบบสตาร์ในสำนักงานมีลักษณะดังรูปที่ 2.16 ซึ่งผู้ออกแบบเครือข่ายนิยมวางฮับหรือสวิตช์ไว้ที่มุมใดมุมหนึ่งของห้องแล้วเชื่อมสายสัญญาณขึ้นมาจากพื้นอาคาร หรือใต้หลังคาเพื่อความเรียบร้อยและสวยงาม

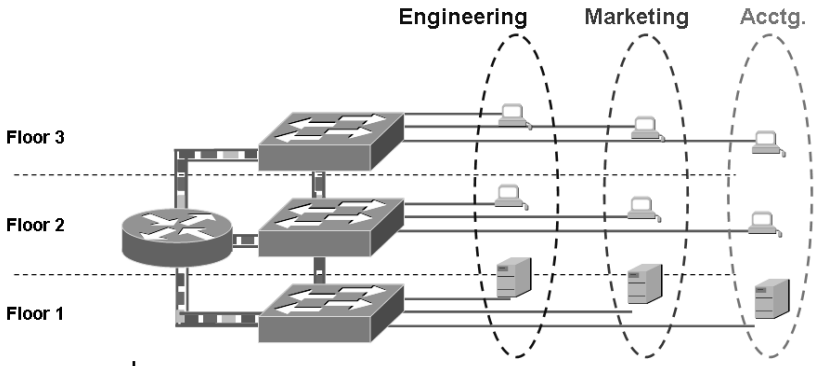


รูปที่ 2.16 แสดงตัวอย่างการเชื่อมโยงเครือข่ายแบบสตาร์ภายในอาคาร

ก่อนการเชื่อมต่อเครือข่ายจริงทุกครั้งจำเป็นต้องออกแบบเครือข่ายในแผนผังหรือในซอฟต์แวร์จำลองเครือข่ายเสียก่อน เพื่อเป็นการประหยัดเวลาการทำงาน ป้องกันการทำงานที่ซ้ำซ้อน มองภาพรวมเครือข่ายได้อย่างครบถ้วน และป้องกันความผิดพลาดที่คาดไม่ถึง เป็นต้น

4. เครือข่ายเสมือนจริง (Virtual Area Network: VLAN)

เครือข่ายเสมือนจริง คือ การแบ่งแยกระบบเครือข่ายคอมพิวเตอร์ออกเป็นส่วน ๆ หรือเป็นกลุ่มย่อย ๆ ด้วยซอฟต์แวร์ โดยไม่มีความจำเป็นต้องปรับแต่งหรือเพิ่มเติมอุปกรณ์ที่ใช้เชื่อมต่อทางกายภาพเลย คอมพิวเตอร์ที่อยู่ภายในเครือข่ายเสมือนจริงเดียวกันสามารถสื่อสารกันได้เลย [2, 23, 34] สำหรับคอมพิวเตอร์ที่อยู่ต่างเครือข่ายเสมือนจริง จะไม่สามารถสื่อสารกันได้ ถ้าต้องการสื่อสารไปยังเครือข่ายเสมือนจริงอื่น ๆ จะต้องสื่อสารผ่านเกตเวย์ (Network gateway) ระหว่างเครือข่ายเสมือนจริงเท่านั้น จุดประสงค์หลักของการสร้างเครือข่ายเสมือนจริง เพื่อจำกัดการเข้าถึงข้อมูลของเครื่องคอมพิวเตอร์ที่อยู่ต่างกลุ่ม หรือต่างเครือข่าย ทั้งนี้เพื่อความปลอดภัยของเครือข่าย รวมทั้งสามารถเพิ่มประสิทธิภาพการทำงานของเครือข่ายด้วย ในทางปฏิบัติเครือข่ายหนึ่ง ๆ อาจประกอบด้วยอุปกรณ์กระจายสัญญาณ (Switching) ได้หลาย ๆ ตัว และในอุปกรณ์กระจายสัญญาณแต่ละตัว อาจจะมีเครือข่ายเสมือนจริงปรากฏอยู่ได้มากกว่า 1 เครือข่าย โดยทั่วไปอุปกรณ์กระจายสัญญาณ 1 ตัว สามารถสร้างเครือข่ายเสมือนจริงได้มากถึง 1,024 เครือข่าย ดังรูปที่ 2.17



รูปที่ 2.17 แสดงลักษณะของการสร้างเครือข่ายเสมือนจริง

จากรูปที่ 2.17 เครือข่ายเสมือนจริง ชื่อว่า Engineering ปรากฏอยู่ในอุปกรณ์ขยายสัญญาณที่ติดตั้งไว้บนชั้นที่ 1 (Floor 1), 2 และ 3 ตามลำดับ เช่นเดียวกันกับเครือข่ายเสมือนจริง Marketing และ Accounting ซึ่งแสดงให้เห็นว่า วิศกรที่ทำงานอยู่ต่างสถานที่กันสามารถทำงานเสมือนอยู่ในสภาวะแวดล้อมเดียวกันได้ คุณสมบัติของเครือข่ายเสมือนจริงจะไม่อนุญาตให้ผู้ใช้งานของแต่ละเครือข่ายเสมือนจริงสื่อสารกันได้โดยตรง เช่น ฝ่ายวิศวกรไม่สามารถเข้าถึงข้อมูลของฝ่ายบัญชีและการตลาดได้ ถ้าต้องให้เครือข่ายเสมือนจริงแต่ละเครือข่ายสามารถสื่อสารระหว่างกันได้ จำเป็นต้องอาศัยตัวกลางที่ทำหน้าที่เชื่อมต่อเครือข่ายเสมือนจริงเข้าด้วยกัน เรียกว่า เกตเวย์ (Gateway) จากรูปที่ 2.17 อุปกรณ์เกตเวย์ คือ เราเตอร์ ที่ติดตั้งบนชั้นที่ 2)

ข้อดีของการใช้เครือข่ายเสมือนจริง

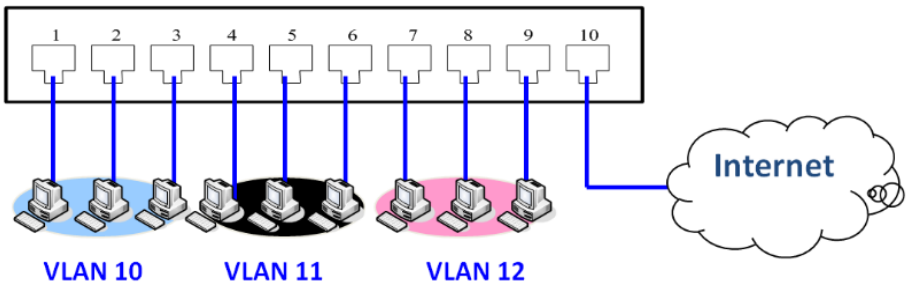
1. สามารถป้องกันปัญหาการบอร์ดคลาส (Broadcast) ข้อมูลมิให้กระจายไปทั่วทั้งเครือข่าย
2. สามารถจำกัดข้อมูลการจราจร (Traffic) ให้อยู่ในบริเวณที่สามารถควบคุมได้
3. การรักษาความปลอดภัย เนื่องจากการสร้างเครือข่ายเสมือนจริงทำให้อุปกรณ์ที่อยู่ต่างเครือข่ายไม่สามารถสื่อสารกันได้ (เมื่อสื่อสารกันไม่ได้ ก็ไม่สามารถโจมตีกันได้)
3. สามารถกำหนดขอบเขตการแพร่กระจายข้อมูลเฉพาะกลุ่มได้ (Multicast)
4. สามารถเพิ่มประสิทธิภาพการทำงานของเครือข่าย เนื่องจากเครือข่ายเสมือนจริงสามารถลดปัญหาของบอร์ดคลาส ส่งผลให้สื่อนำสัญญาณว่าง ดังนั้นข้อมูลจึงสามารถสื่อสารกันได้เพิ่มมากขึ้น

4.1 ชนิดของเครือข่ายเสมือนจริง

เครือข่ายเสมือนจริงแบ่งออกได้หลายประเภทขึ้นอยู่กับชนิดของอุปกรณ์กระจายสัญญาณ ลักษณะของงาน และการจัดโครงแบบของเครือข่าย (Configuration) เป็นต้น สำหรับรูปแบบทั่วไปที่นิยมใช้สำหรับจำแนกรูปแบบเครือข่ายเสมือนจริง ดังต่อไปนี้

1) เครือข่ายเสมือนจริงชนิด Port-Based

เป็นการจัดแบ่งเครือข่ายเสมือนจริง โดยอาศัยหมายเลขพอร์ตเป็นหลัก คือ กำหนดว่าในอุปกรณ์กระจายสัญญาณแต่ละตัวมีเครือข่ายเสมือนจริงจำนวนเท่าใด มีชื่ออะไรบ้าง และต้องการให้พอร์ตใดเป็นสมาชิกของเครือข่ายเสมือนจริงวงใด เป็นต้น ดังรูปที่ 2.18 [25, 34]



รูปที่ 2.18 แสดงเครือข่ายเสมือนจริงชนิด Port-Based

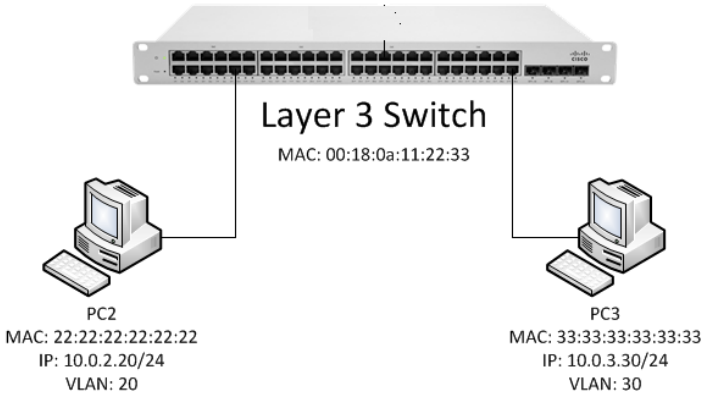
ขั้นตอนในการติดตั้งเครือข่ายเสมือนจริงชนิด Port-Based สามารถดำเนินการโดยมีขั้นตอนคร่าว ๆ ดังนี้

1. กำหนด VTP Domain เป็นอันดับแรก
2. กำหนดชื่อของเครือข่ายเสมือนจริง รวมทั้งเลขหมายของเครือข่าย
3. กำหนดหมายเลขพอร์ตให้กับเครือข่ายเสมือนจริงในแต่ละวงที่ถูกสร้างขึ้น

ข้อดีของเครือข่ายเสมือชนิด Port-Based คือ สามารถย้ายกลุ่มจากเครือข่ายเสมือนจริงหนึ่งไปอีกเครือข่ายเสมือนจริงหนึ่งได้โดยง่าย ตัวอย่างเช่น สมมุติว่า อุปกรณ์กระจายสัญญาณตัวหนึ่งซึ่งมีจำนวนพอร์ตเท่ากับ 10 พอร์ต และทำการสร้างเครือข่ายเสมือนจริงไว้ 3 วง คือ VLAN 10, VLAN 11 และ VLAN 12 โดย VLAN 10 มีหมายเลขพอร์ตที่เป็นสมาชิก คือ 1, 2 และ 3 สำหรับ VLAN 11 มีพอร์ตสมาชิก คือ 4, 5 และ 6 สำหรับ VLAN 12 มีพอร์ตสมาชิกหมายเลข 7, 8 และ 9 ตามลำดับ ถ้าต้องการให้อุปกรณ์สื่อสารซึ่งเป็นสมาชิกของ VLAN 10 ไปเป็นสมาชิกของ VLAN 11 สามารถทำได้โดยปรับแต่งคอนฟิกกูเรชันผ่านซอฟต์แวร์บริหารจัดการบนอุปกรณ์กระจายสัญญาณ โดยเปลี่ยนความเป็นสมาชิกจาก VLAN 10 เป็น VLAN 11 เท่านั้น หรือสามารถปรับแต่งทางกายภาพโดยการย้ายอุปกรณ์สื่อสารจากพอร์ตที่เป็นสมาชิกบน VLAN 10 ไปยังพอร์ตที่เป็นสมาชิกของ VLAN 11 (เปลี่ยนจากพอร์ตหมายเลขระหว่าง 1 ถึง 3 บน VLAN 10 ไปยังพอร์ตหมายเลข 4 ถึง 6 บน VLAN 11)

2) เครือข่ายเสมือนจริงชนิด MAC Address-Based

เครือข่ายเสมือนจริง MAC Address-Based [25, 34] คือ การสร้างเครือข่ายเสมือนจริงโดยอาศัยที่อยู่ทางกายภาพ (MAC Address) ของอุปกรณ์เครือข่ายเป็นหลัก ซึ่งที่อยู่ทางกายภาพนี้เป็นหมายเลขฐานสิบหกขนาด 48 บิต ที่ไม่ซ้ำกัน โดยถูกกำหนดไว้กับเน็ตเวิร์คการ์ดของอุปกรณ์เครือข่ายทุก ๆ ตัว การแบ่งเครือข่ายเสมือนจริงด้วยวิธีการทางกายภาพนี้ง่ายต่อการปรับแต่งคอนฟิกกูเรชัน (Configuration) มาก เนื่องจากไม่จำเป็นต้องกำหนดเลขหมายของพอร์ต และไม่ต้องสนใจว่าอุปกรณ์เครือข่ายจะติดตั้งอยู่บนพอร์ตหมายเลขใด และไม่ต้องกลัวว่าจะมีผู้ใดย้ายพอร์ตเพื่อเปลี่ยนวงของเครือข่ายเสมือ เนื่องจาก ไม่ว่าจะย้ายไปอยู่ที่ใด บนอุปกรณ์กระจายสัญญานตัวใด ตราบใดที่กำหนดที่อยู่ทางกายภาพให้กับเครือข่ายเสมือนจริงแล้ว จะเปลี่ยนแปลงสมาชิกของเครือข่ายเสมือ ได้ก็ต่อเมื่อ มีการเปลี่ยนเน็ตเวิร์การ์ดเท่านั้น ดังรูปที่ 2.19



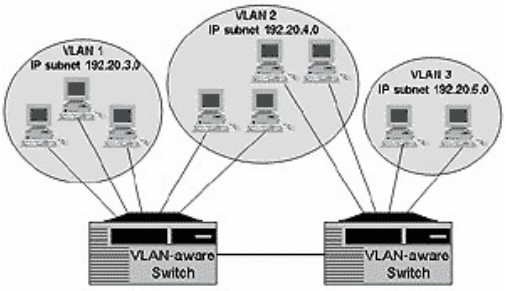
รูปที่ 2.19 ตัวอย่างเครือข่ายชนิด MAC Address-Based

ข้อจำกัดของ MAC Address-Based VLAN

พอร์ตที่ใช้งานร่วมกับ MAC-Based VLAN นั้นจะต้องไม่เป็น Static VLAN หมายความว่าจะต้องไม่มีการกำหนดหมายเลขพอร์ตที่ตายตัวให้กับ VLAN ใด ๆ MAC Based VLAN ถูกออกแบบมาให้สามารถรองรับจำนวนผู้ใช้งาน 1 คน (Client) ต่อหนึ่งพอร์ตเท่านั้น แต่ปัจจุบันมีสวิตช์บางรุ่นสามารถรองรับจำนวนผู้ใช้งานมากกว่า 1 คน ต่อ 1 พอร์ตได้

3) Subnet-Based VLAN

Subnet-Based VLAN [25, 34] บางครั้งถูกเรียกว่า Layer-3 Based VLAN เป็น VLAN ที่ถูกสร้างขึ้นโดยอาศัยข้อมูลข่าวสารในระดับชั้นเน็ตเวิร์ค (Network Layer: Layer 3) โดยอุปกรณ์สวิตช์จะตรวจสอบข้อมูลไอพีที่ส่วนหัว (Header) ของแพ็กเก็ต ปกติ Subnet-based VLAN จะถูกติดตั้งบนสวิตช์ที่ทำงานในระดับชั้นเน็ตเวิร์คเท่านั้น ขณะที่ VLAN ชนิด MAC Address-Based จะทำงานบนระดับชั้นดาตาลิงค์ (Data link layer: Layer 2)



รูปที่ 2.20 Subnet-Based VLAN [2]

จากรูป 2.20 แสดงการทำงานของอุปกรณ์สวิตช์ (Layer 3 Switching) เพื่อสร้าง VLAN จำนวน 2 กลุ่ม คือ VLAN 1 และ VLAN 2 จะสังเกตเห็นว่า VLAN ถูกแบ่งออกเป็นส่วน ๆ โดยใช้หมายเลขไอพีมาเป็นตัวกำหนด VLAN ที่ต่างกัน ข้อดีของการจัด VLAN ประเภทดังกล่าวนี้ ได้แก่ ความยืดหยุ่นของการประยุกต์ใช้งาน เนื่องจากสามารถปรับเปลี่ยน VLAN โดยการเปลี่ยนหมายเลขของไอพีเท่านั้น

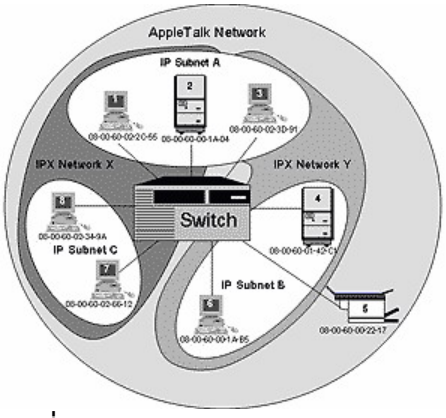
ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์สื่อสารบนพอร์ตหมายเลขใดก็ได้บนอุปกรณ์สวิตช์ โดยไม่มีความจำเป็นต้องแก้ไขคอนฟิกกูเรชันใหม่ วิธี Subnet-Based VLAN เหมาะสำหรับเครือข่ายที่ใช้โพรโทคอลทีซีพี/ไอพี (TCP/IP) เป็นหลัก ค่าใช้จ่ายในการดูแลรักษา VLAN ประเภทนี้ จะถูกกว่า MAC Address-Based มาก

ข้อเสียของ Subnet-Based VLAN

ข้อเสียของ Subnet-Based VLAN คือ การสร้างกลุ่มของไอพีหลาย ๆ ชุดบนอุปกรณ์สวิตช์ตัวเดียวกัน ทำให้เกิดความสับสนได้ง่ายกว่าการสร้าง VLAN ประเภทอื่น ๆ รวมทั้งปัญหาของสวิตช์บางรุ่นที่อาจสนับสนุนหลายไอพีแอดเดรสบนพอร์ตเดียวกัน

4) Protocol-Based VLAN

รูปแบบของ VLAN ประเภทนี้ช่วยให้การสร้าง VLAN สามารถดำเนินการได้ง่ายกว่าการสร้าง VLAN แบบอื่น ๆ [25, 34] เนื่องจากการสร้าง VLAN ดังกล่าวจะอาศัยโพรโทคอลซึ่งอยู่ในระดับเน็ตเวิร์คเป็นตัวจำแนกการทำงานของแต่ละ VLAN ซึ่งโพรโทคอลเหล่านี้ได้แก่ IP และ IPX เป็นต้น Protocol-Based VLAN ถูกนำมาใช้ในสถานการณ์ที่ระบบเครือข่ายมีอุปกรณ์ที่มีความหลากหลาย โดยเฉพาะอย่างยิ่งมีการใช้งานโพรโทคอลที่แตกต่างกันทำงานอยู่ด้วยกัน หรือในสถานการณ์ที่ระบบเครือข่ายถูกแบ่งออกเป็นหลาย ๆ เซ็กเมนต์ (Segment) เป็นต้น จากรูป 2.21 แสดงรูปแบบการสร้าง Protocol-Based VLAN โดย VLAN 1 จะประกอบไปด้วยพอร์ตที่เป็นสมาชิกหมายเลข 1 ถึง 4 สำหรับประยุกต์ใช้กับโพรโทคอล IP และ VLAN 2 รองรับการทำงานโพรโทคอล IPX โดยมีสมาชิกพอร์ตหมายเลข 5 ถึง 8



รูปที่ 2.21 Protocol-Based VLAN [2]

ข้อดีของการใช้ Protocol-Based VLAN

ข้อดีของ Protocol-Based VLAN ได้แก่ความยืดหยุ่นในการใช้งาน เนื่องจากอุปกรณ์สื่อสารต่าง ๆ ไม่จำเป็นต้องติดตั้งอยู่ในสถานที่เดียวกัน สารสามารถติดตั้งกระจายอยู่บนเครือข่าย ณ

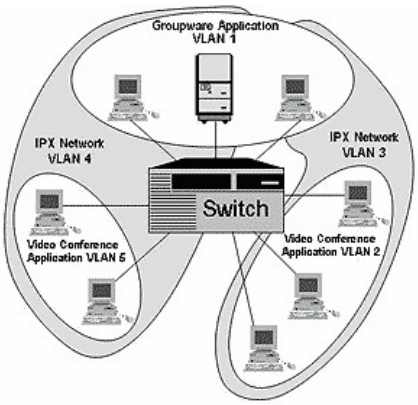
ตำแหน่งใด ๆ ก็ได้ แต่ละอุปกรณ์จะอยู่ภายใต้ VLAN เดียวกัน โดยอาศัยโพรโทคอลในการเชื่อมโยงอุปกรณ์เข้าด้วยกันแทน

ข้อเสียของการใช้ Protocol-Based VLAN

เครื่องคอมพิวเตอร์อาจติดตั้งและใช้งานโพรโทคอลหลายประเภทบนเครื่องเดียวกัน เช่น มีการใช้งาน IP กับ NetBIOS ร่วมกัน ซึ่งอาจจะส่งผลให้อุปกรณ์ที่ควบคุมดูแล VLAN ทำงานหนักมากกว่าปกติและอุปกรณ์ที่ควบคุม VLAN ต้องมีคุณภาพสูงและมีราคาแพงตามไปด้วย

5) Application-Based VLAN

Application-Based VLAN [25, 34] จะอาศัยประเภทของโปรแกรมประยุกต์ในการจัดกลุ่มความเป็นสมาชิกของแต่ละ VLAN ตัวอย่างเช่น โปรแกรมการประชุมทางไกล (Video conference) จะถูกจัดให้อยู่กลุ่มเดียวกัน (แสดงในรูปที่ 2.22) โดยอุปกรณ์คอมพิวเตอร์ที่สื่อสารกันจะอยู่บนเครือข่ายส่วนไหนก็ได้ ไม่จำเป็นต้องติดตั้งอยู่บนอุปกรณ์สวิตช์ตัวเดียวกัน จุดประสงค์ของการแยก VLAN โดยอาศัยโปรแกรมประยุกต์นี้ เพื่อเป็นการเอื้อประโยชน์ให้กับโปรแกรมประยุกต์แต่ละประเภท ให้สามารถใช้แบนด์วิชท์ได้อย่างเต็มประสิทธิภาพ อีกทั้งยังสามารถแยกประเภทของงานออกได้อย่างชัดเจน Application-Based VLAN จึงมีประโยชน์สำหรับหน่วยงานที่ต้องการใช้งานโปรแกรมประยุกต์ที่มีลักษณะจำเพาะเจาะจง ปัจจุบันอุปกรณ์ที่สนับสนุน VLAN ประเภท Application-Based VLAN ไม่เป็นที่นิยมใช้งาน เนื่องจากมีราคาแพง

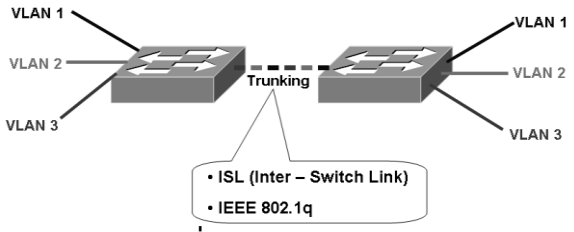


รูปที่ 2.22 Protocol-Based VLAN [2]

ข้อเสีย คือ ถ้าบางองค์กรใช้โปรแกรมประยุกต์ที่มีความหลากหลายและโปรแกรมประยุกต์เหล่านั้นมีปริมาณการใช้งานที่ต่างกันมาก จะส่งผลให้ขนาดของ VLAN มีขนาดใหญ่ไม่สมดุลกับ VLAN อื่น ๆ ทำให้เกิดปัญหาในเรื่องของบรอดคลาสได้

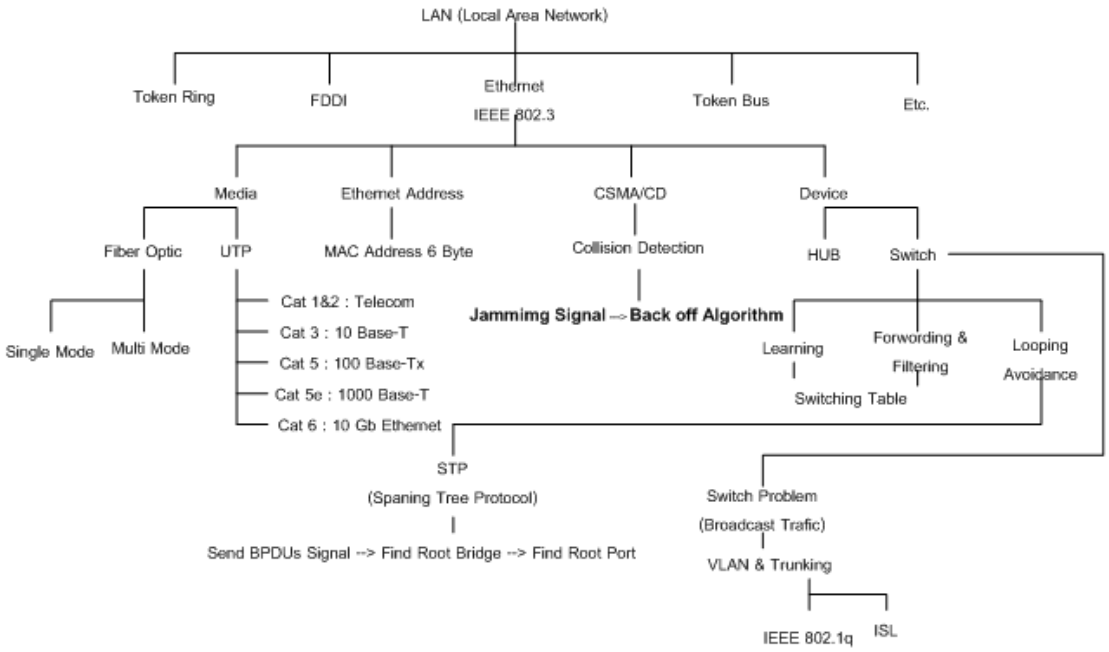
6) VLAN Trunk

VLAN Trunk มีหน้าที่เชื่อมต่อ Switching Hub ที่ติดตั้ง VLAN 2 Hub เข้าด้วยกัน และที่สำคัญได้แก่ การเชื่อมต่อ Switching Hub ที่ติดตั้ง VLAN กระจายไปตาม Hub ต่างๆ เหล่านี้ หลังจากที่เชื่อมต่อ Switches พร้อม VLAN เข้าด้วยกันแล้ว จะสามารถส่งผ่านสมาชิกของ VLAN หนึ่งจาก Switches Hub หนึ่ง ไปยัง Switches อีกหนึ่ง ภายใต้ VLAN ชุดเดียวกัน ด้วยสายสัญญาณที่เชื่อมต่อเป็น Trunk เพียงเส้นเดียว ไม่ว่า Switches Hub ตัวแรกจะมี VLAN กี่ชุดก็ตาม แต่เมื่อต้องการสื่อสารกับสมาชิก VLAN ชื่อเดียวกัน แต่อยู่ต่าง Hub กัน ก็สามารถทำได้อย่างง่ายดาย ด้วยสายสัญญาณเพียงเส้นเดียว และนี่คือประโยชน์ของการใช้ VLAN Trunk ดังรูปที่ 2.23



รูปที่ 2.23 VLAN Trunk

รูปที่ 2.24 แสดงพัฒนาการของเทคโนโลยีเครือข่าย LAN ตั้งแต่อดีตจนถึงปัจจุบัน

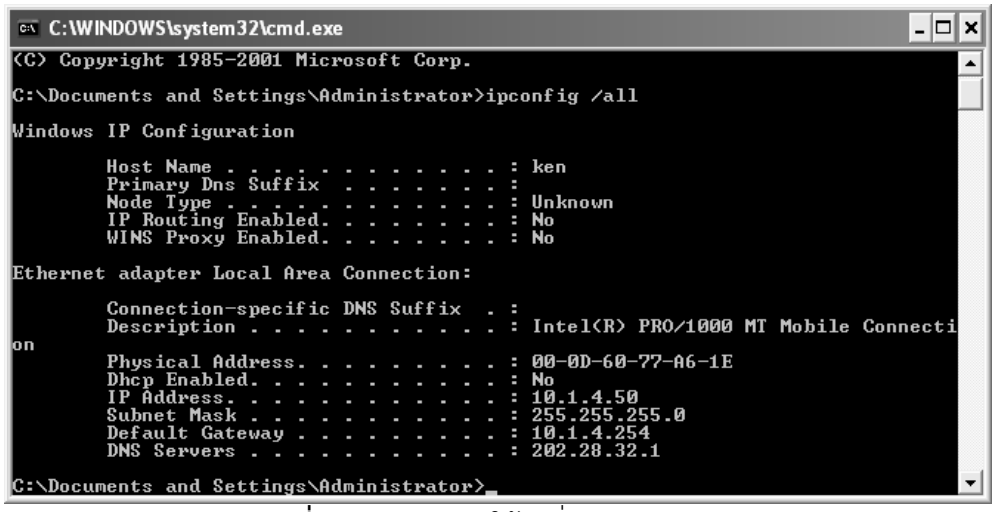


รูปที่ 2.24 แสดงพัฒนาการของเทคโนโลยีเครือข่าย LAN

อุปกรณ์กระจายสัญญาณระดับผู้ใช้งาน (Access Layer)

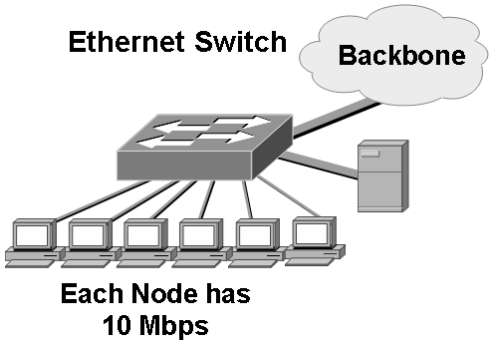
ในส่วนของเลเยอร์ผู้ใช้งานนั้นไม่มีความซับซ้อนมากนัก โดยส่วนประกอบหลัก ๆ ที่จะต้องมีเมื่อต้องการต่อเชื่อมกับระบบเครือข่ายคือ

1. เครื่องคอมพิวเตอร์ (PC) โดยต้องการมีการเซ็ตหมายเลขของไอพี ในกรณีที่ระบบไม่มีการใช้งาน DHCP คำสั่งเบื้องต้นที่ใช้สำหรับตรวจสอบหมายเลขไอพี คือคำสั่ง ipconfig ดังรูปที่ 2.25



รูปที่ 2.25 แสดงการใช้คำสั่ง ipconfig

2. เครื่องที่ต้องการเชื่อมต่อเข้าระบบเครือข่ายต้องมีกราดแลน (NIC Card)
3. ทำการต่อเชื่อมเข้ากับอุปกรณ์เครือข่ายประเภท ฮับ หรือสวิตช์ เพื่อเข้าสู่ระบบเครือข่ายหลัก ดังรูปที่ 2.26



รูปที่ 2.26 การเชื่อมต่อคอมพิวเตอร์เข้าสู่เครือข่าย

แบบฝึกหัดท้ายบท

1. ประเภทของการเชื่อมต่อเครือข่าย (categories of topology) มีกี่ประเภท อะไรบ้าง
2. ข้อดีและข้อเสียของการเชื่อมต่อเครือข่ายแบบ mesh คืออะไร
3. การเชื่อมต่อแบบสตาร์ มีลักษณะเป็นอย่างไร และมีข้อดีข้อเสียอย่างไร
4. การเชื่อมต่อแบบบัส ใช้ในงานในลักษณะใด และทำไมจึงต้องใช้ลักษณะดังกล่าว

5. ในการแก้ปัญหาสำหรับการเชื่อมต่อแบบวงแหวน ในกรณีที่สายนำสัญญาณขัดข้อง สามารถทำได้อย่างไร จงอธิบาย
6. ประเภทของระบบเครือข่ายแบ่งได้กี่ประเภท อะไรบ้าง
7. ประเภทการเชื่อมต่อแบบ LAN มีลักษณะเป็นอย่างไร และสนับสนุนการสื่อสารข้อมูลในลักษณะใด
8. ประเภทการเชื่อมต่อแบบ MAN มีลักษณะเป็นอย่างไร และสนับสนุนการสื่อสารข้อมูลในลักษณะใด
9. ประเภทการเชื่อมต่อแบบ WAN มีลักษณะเป็นอย่างไร และสนับสนุนการสื่อสารข้อมูลในลักษณะใด
10. การเข้าถึงระบบเครือข่ายมี 3 ระดับประกอบไปด้วยอะไรบ้าง และแต่ละประเภทแตกต่างกันอย่างไร

บทที่ 3
อุปกรณ์เครือข่ายและการคอนฟิกกูเรชัน
(Networking Devices and Configuration)

- Connecting Devices
- Command line interface
- System startup
- Managing system files
- Using show commands

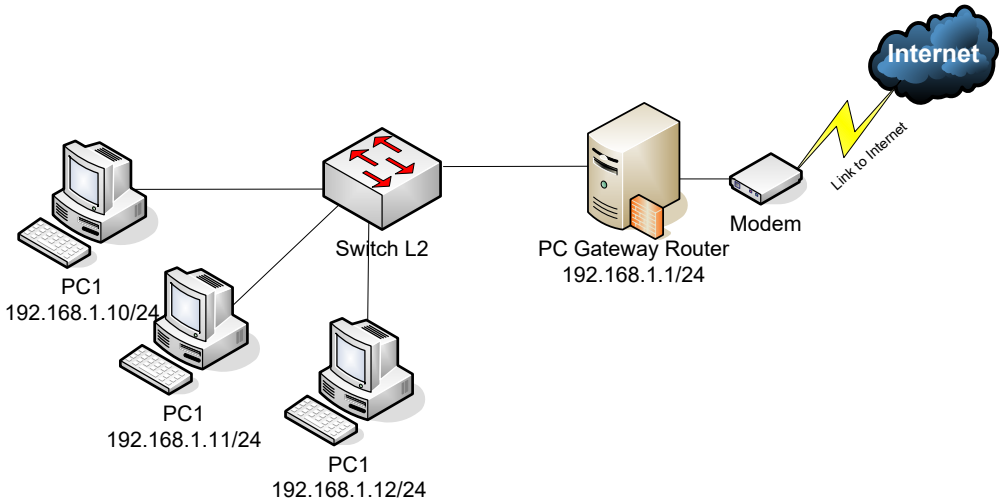
แนวคิด

การทำความเข้าใจถึงหน้าที่การทำงาน คุณลักษณะ ความสามารถต่างๆ ของอุปกรณ์เครือข่ายในแต่ละส่วนว่าเป็นอย่างไร (โดยเฉพาะเราเตอร์ และสวิชต์) จะทำให้ผู้ดูแลระบบสามารถกำหนดโครงสร้างและการทำงานของระบบเครือข่ายโดยรวมได้เป็นอย่างดี

วัตถุประสงค์

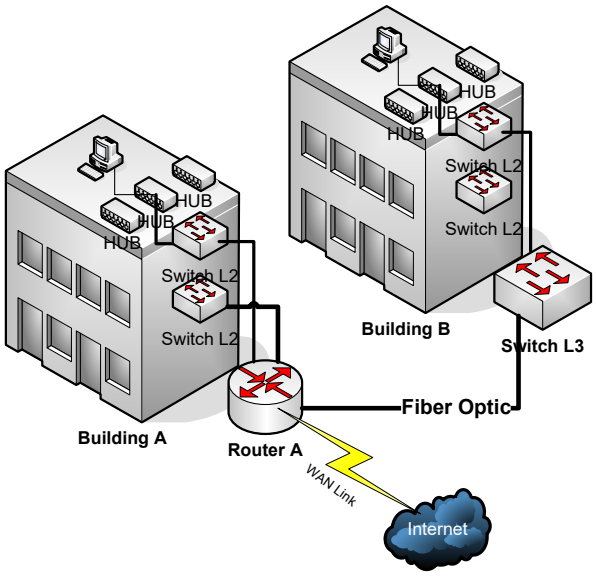
1. เพื่อให้ทราบถึงหน้าที่การทำงานของอุปกรณ์ที่ทำงานอยู่บนระบบเครือข่าย
2. เพื่อให้ทราบถึงประเภทของอุปกรณ์ ตำแหน่งที่ใช้สำหรับจัดวางอุปกรณ์บนระบบเครือข่าย ส่วนประกอบของอุปกรณ์แต่ละชนิด
3. เพื่อให้ทราบถึงโครงสร้างการทำงานของฮาร์ดแวร์ในอุปกรณ์ระบบเครือข่าย
4. เพื่อให้ทราบถึงวิธีการติดตั้งและคอนฟิกอุปกรณ์ที่ทำหน้าที่สำคัญๆ บนระบบเครือข่าย

การแบ่งประเภทของอุปกรณ์ที่ทำงานอยู่บนระบบเครือข่ายสามารถแบ่งตามความสามารถในการส่งถ่ายปริมาณของข้อมูล ได้แก่ ชนิดที่สามารถส่งข้อมูลได้ในปริมาณที่มาก ๆ ควรจะทำการติดตั้งไว้สำหรับเป็นแกนหลักของระบบ (Core Layer) อุปกรณ์ที่ทำหน้าที่ในส่วนนี้ มักจะเป็นเราเตอร์ (Router) หรืออาจจะเป็นสวิตช์ที่สามารถทำงานได้ในเลเยอร์ที่สามได้ (Switch L3) หรือถ้าหน่วยงานที่มีงบประมาณที่จำกัดอาจจะมีเพียงสวิตช์เลเยอร์สองต่อเข้ากับคอมพิวเตอร์ที่ติดตั้งซอฟท์แวร์เพื่อทำหน้าที่หาเส้นทางได้เช่น ใช้ซอฟท์แวร์เราเตอร์ รูปที่ 3.1 แสดงลักษณะการเชื่อมต่อประเภทนี้



รูปที่ 3.1 แสดงการเชื่อมต่อโดยใช้ PC เป็นอุปกรณ์เลเยอร์สาม

ส่วนในองค์กรที่มีขนาดปานกลางถึงใหญ่และมีงบประมาณในการจัดซื้ออุปกรณ์ค้นหาเส้นทางได้ ก็ต้องมีการวางแผนในเบื้องต้นก่อนว่าจะออกแบบแกนหลักของเครือข่ายอย่างไรเช่น ติดตั้งอุปกรณ์เหล่านี้ตามพื้นที่ของอาคาร หรือติดตั้งตามหน้าที่การทำงานของผู้ใช้ เช่น แผนกการเงินก็ควรจะอยู่ใน VLAN (การสร้าง LAN ขึ้นมาโดยไม่ขึ้นกับทางกายภาพ) เดียวกัน ขึ้นอยู่กับความเหมาะสมกับลักษณะของปัญหาในแต่ละที่แต่ละแห่ง จุดนี้ไม่สามารถที่จะกำหนดลงไปได้ว่าการวางอุปกรณ์แบบดะจะดีกว่ากัน แต่จากประสบการณ์ของผู้เขียน มีความเห็นส่วนตัวว่าการวางตามลักษณะทางกายภาพของอาคารหรือตึก ดังรูปที่ 3.2 จะทำได้สะดวกมีแบบแผนและดูแลรักษาได้ง่าย ไม่สับสน เนื่องจากว่าอาคารต่าง ๆ มีลักษณะที่ไม่มีการเปลี่ยนแปลงเลยตลอดอายุการใช้งานของมัน และตอนวางผังหรือเขียน Network Diagram ออกมาสามารถกำหนดได้อย่างชัดเจนว่า ตึกแต่ละตึกมีหมายเลขประจำตึกเป็นหมายเลขอะไร ชั้นที่เท่าไหร่ ซึ่งในตอนหลังจะสะดวกเป็นอย่างมากในการทำ VLAN



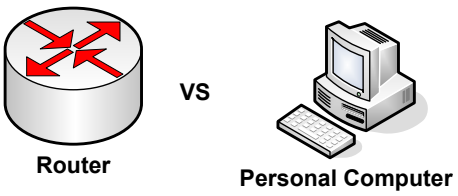
รูปที่ 3.2 การวางอุปกรณ์หลักตามลักษณะทางกายภาพ

อุปกรณ์กระจายสัญญาณหลัก (Core Layer)

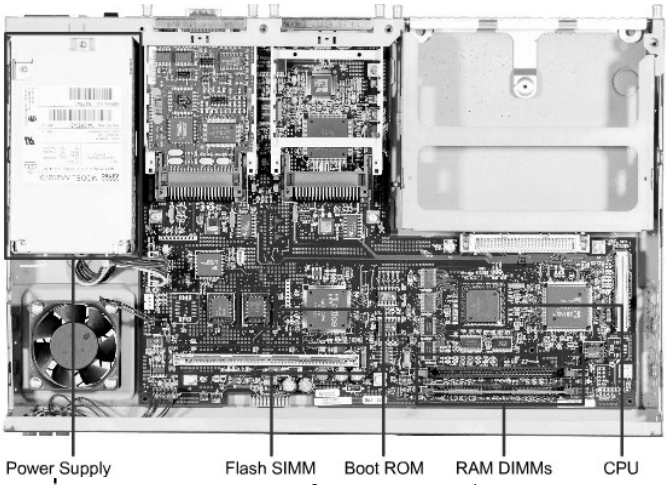
จากหลักการที่กล่าวมาแล้วว่าอุปกรณ์ที่ทำหน้าที่ใน Core Layer นั้นจะต้องโอนถ่ายข้อมูลได้ในปริมาณที่มากและรวดเร็ว พร้อมทั้งต้องฉลาดในการหาเส้นทางที่ดีที่สุดและสั้นที่สุดด้วย ซึ่งหนีไม่พ้นอุปกรณ์ที่เรียกว่า "เราเตอร์" ในที่นี้จะกล่าวถึงเราเตอร์ของบริษัทซิสโก้ (Cisco) เป็นหลักเนื่องจากเป็นอุปกรณ์ที่ค่อนข้างจะได้รับความนิยมเป็นอย่างกว้างขวางและมีประสิทธิภาพในการทำงานที่มีเสถียรภาพเป็นอย่างมาก ซึ่งเมื่อเข้าใจการทำงานในอุปกรณ์ของบริษัทนี้แล้วก็ไม่เป็นการยากที่จะเรียนรู้และทำความเข้าใจกับอุปกรณ์ของบริษัทอื่น ๆ และที่สำคัญคำสั่งที่ใช้งานบนอุปกรณ์เราเตอร์หรือสวิชต์ของบริษัทนี้นั้น เข้าใจได้ง่าย และมี Help ช่วยเหลือ ทำให้ผู้ดูแลระบบที่ไม่สามารถจดจำคำสั่งได้ทั้งหมด ก็ไม่จำเป็นต้องกังวลมากนัก ซึ่งการใช้งานคำสั่งก็จะกล่าวในลำดับต่อ ๆ ไป

เราเตอร์ (Router)

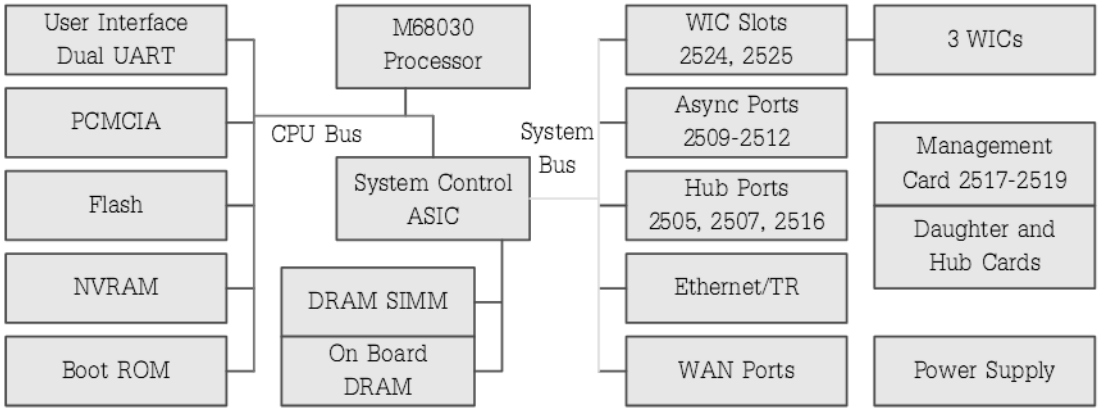
อุปกรณ์หาเส้นทางเช่น เราเตอร์จริง ๆ แล้วโครงสร้างของเครื่องก็คือ PC เหมือนที่ใช้งานกันอยู่ทั่วไปนั่นเอง ดังรูปที่ 3.3, 3.4 จะแตกต่างออกไปตรงที่เราเตอร์ถูกสร้างขึ้นมาทำงานเฉพาะทางเพื่อรองรับการโอนถ่ายข้อมูลให้เร็วที่สุดและค้นหาเส้นทางที่ดีที่สุดเป็นหลัก เราเตอร์ไม่สนใจเรื่องการคำนวณกราฟฟิก ไม่สนใจการประมวลผลเกมส์ต่าง ๆ ที่ PC ทำอยู่ในปัจจุบัน เราเตอร์สนใจแต่รับข้อมูลมาจากไหนและจะส่งข้อมูลไปทางไหนให้ดีที่สุดเท่านั้น



รูปที่ 3.3 โครงสร้างสถาปัตยกรรมของเราเตอร์คล้ายกับ PC



รูปที่ 3.4 ส่วนประกอบภายในของเราเตอร์ รุ่น Cisco 2600



รูปที่ 3.5 ผังโครงสร้างของเราเตอร์ Cisco

ส่วนประกอบหลัก ๆ ของเราเตอร์จะมีดังนี้ (ดังรูปที่ 3.5)

- **ROM** ทำหน้าที่เซ็ตค่าเริ่มต้นให้กับเราเตอร์ (POST) ดังรูปที่ 3.6 เหมือนกับการทำงานของ Bios ของ PC คือในรอมจะมีระบบปฏิบัติการขนาดเล็ก ๆ คอยจัดการเรื่องการตรวจสอบความพร้อมของฮาร์ดแวร์ เมื่อการทำงานของฮาร์ดแวร์ไม่มีข้อผิดพลาดแล้ว ก็จะส่งการทำงานต่อให้กับ IOS (เป็นระบบปฏิบัติการของบริษัท Cisco สร้างเพื่อทำงานกับเราเตอร์โดยเฉพาะ) ที่เก็บอยู่ใน Flash อีกทีหนึ่ง

รูปที่ 3.6 ROM

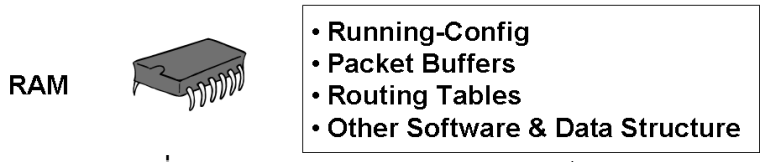
- **Flash Memory** เป็นหน่วยความจำแบบกึ่งถาวร ที่กล่าวเช่นนี้เพราะว่าข้อมูลที่เก็บไว้จะสูญหายไปได้ถ้ามีการปล่อยกระแสไฟฟ้าเข้าไปยังตัวมัน (EPROM Erasable Programmable Read-Only Memory) มันจะทำหน้าที่เก็บระบบปฏิบัติการสำหรับควบคุมการทำงานของเราเตอร์ ดังรูปที่ 3.7

รูปที่ 3.7 Flash

- **NVRAM** (ดังรูปที่ 3.8) ทำหน้าที่เก็บคอนฟิกกูเรชันไฟล์ เมื่อ IOS เข้าควบคุมการทำงานของระบบเรียบร้อยแล้วมันจะโหลดคอนฟิกไฟล์ที่เก็บอยู่ใน NVRAM ไปทำงาน เมื่อเกิดการเปลี่ยนแปลงคอนฟิกเมื่อได ๆ ในหน่วยความจำหลัก (Running Configuration) เราจะต้องใช้คำสั่ง Write Memory หรือ wr เพื่อบันทึกข้อมูลที่เปลี่ยนแปลงไปไว้ยัง NVRAM ด้วยมิเช่นนั้นถ้าเครื่องมีการเริ่มต้นทำงานใหม่ คอนฟิกที่เปลี่ยนแปลงจะหายไปด้วย

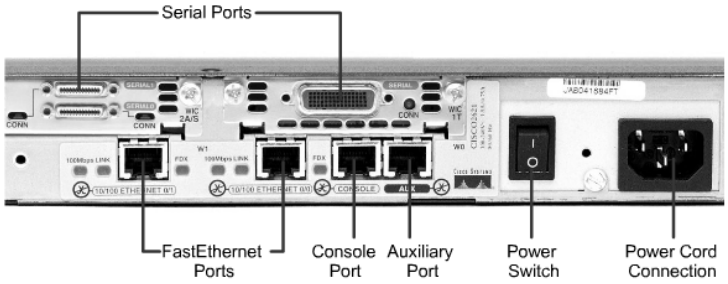
รูปที่ 3.8 NVRAM

- **RAM/DRAM** (ดังรูปที่ 3.9) ทำหน้าที่เป็นหน่วยความจำหลักของเราเตอร์ เก็บคำสั่งที่ทำงานในปัจจุบัน (Running Configuration) ซึ่งคำสั่งเหล่านี้ได้โหลดเอามาจาก NVRAM มาทำงาน ต่อมาเมื่อมีการเปลี่ยนแปลงคอนฟิกใหม่เพิ่มเติม มันจะเก็บข้อมูลที่เปลี่ยนแปลงไว้ในหน่วยความจำหลักนี้ ถ้าไม่มีการเขียนข้อมูลลง NVRAM ข้อมูลที่เปลี่ยนแปลงก็จะหายไปเมื่อมีการรีเซ็ตเราเตอร์

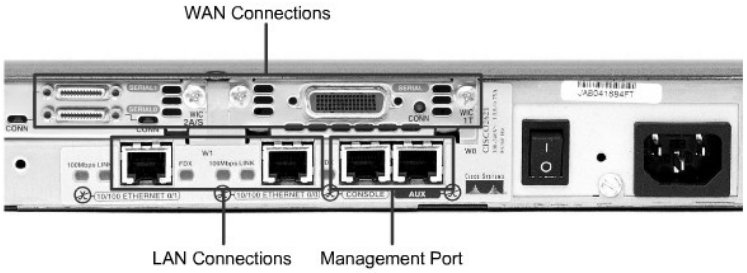


รูปที่ 3.9 หน่วยความจำหลักของเราเตอร์ RAM

- **Router Interface** คือจุดที่ใช้สำหรับเชื่อมต่ออุปกรณ์ภายนอกที่จะเข้ามาต่อกับตัวมัน ซึ่งอินเตอร์เฟสหลัก ๆ มี 2 ชนิดคือ อินเตอร์เฟสที่ใช้เชื่อมต่อกับ WAN Link และอินเตอร์เฟสที่ใช้เชื่อมต่อภายในเครือข่ายท้องถิ่น ดังตัวอย่างรูปที่ 3.10 และ 3.11 อินเตอร์เฟสเครือข่ายท้องถิ่นส่วนมากจะเป็นชนิด Fast Ethernet มีความเร็วที่ประมาณ 100 Mbps การอ้างถึงจะอ้างผ่าน slot แล้วตามด้วยพอร์ต เช่น Fast Ethernet 0/0 (slot 0/port 0) ส่วนขา WAN จะเรียกว่า WIC ซึ่งเราเตอร์รุ่น 2600 จะมี 2 แบบคือ Serial และ BRI การอ้างถึงก็จะทำคล้าย ๆ Fast Ethernet เช่น Serial 0/0 เป็นต้น



รูปที่ 3.10 โครงสร้างภายนอกของเราเตอร์ Cisco รุ่น 2600

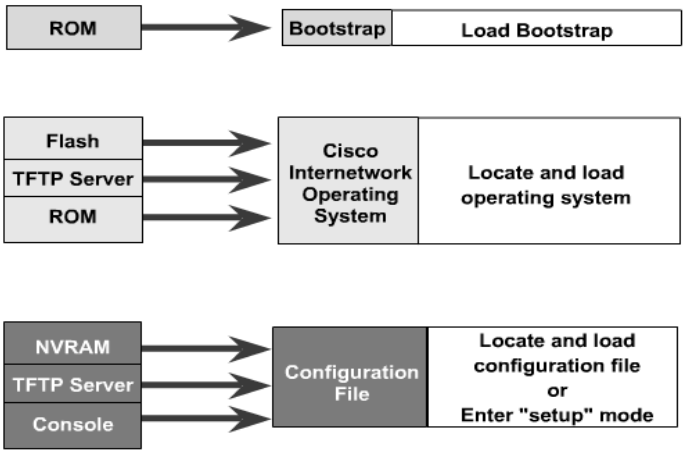


รูปที่ 3.11 ช่องทางสำหรับใช้เชื่อมต่อกับ WAN และ LAN

- **IOS (Internetworking Operating System)** เป็นระบบปฏิบัติการที่คอยควบคุมการทำงานของเราเตอร์ พร้อมทั้งสนับสนุนการทำงานของ โพลโตคอลการหาเส้นทางแบบต่าง ๆ ด้วย

ลำดับการทำงานของเราเตอร์

เริ่มต้นเมื่อทำการเปิด Power ของเราเตอร์ กระบวนการจะเริ่มทำงานดังรูปที่ 3.12



รูปที่ 3.12 ลำดับการทำงานของเราเตอร์

1. เราเตอร์จะเข้าไปอ่านโค๊ตเล็ก ๆ ที่เก็บไว้ใน ROM เรียกขั้นตอนนี้ว่า Bootstrap จากนั้นจะทำการตรวจสอบฮาร์ดแวร์ว่า ทำงานได้ถูกต้องหรือไม่ ถ้าไม่จะแสดงข้อผิดพลาดออกมา ก็จะทำข้อ 2 ต่อ
2. เมื่อทำ Bootstrap เสร็จแล้ว ลำดับต่อไปจะอ่านข้อมูลที่อยู่ใน Flash มาทำงานต่อ ซึ่งใน Flash นี้จะเก็บระบบปฏิบัติการตัวสมบูรณ์อยู่ ซึ่งเรียกว่า IOS ซึ่งชื่อของ IOS จะมีสัญลักษณ์บอกความหมายอยู่ เช่น XXXX-YYYY-WW Code XXXX จะบอกว่ามันเป็นรุ่นอะไร เช่น C2600 YYY จะบอกถึงคุณลักษณะของมันเช่น รองรับการทำงานแบบ IPX WW บอกถึงรูปแบบของไฟล์เช่น เป็นอิมเมจที่มีการบีบอัดข้อมูลไว้ (.zip) ดังรูปที่ 3.13

The name has three parts separated by dashes, xxx-yyy-ww:
- xxxx = Platform
- yyyy = Features
- ww = Format - where the image runs and whether it has been zipped or compressed

Name Codes	
Platform (Hardware) (Partial list)	
c1005	1005
c1600	1600
c1700	1700, 1720, 1750
c2500	25xx, 3xxx, 5100, AO (11.2 and later only)
c2600	2600
c2800	Catalyst 2800
c2900	2910, 2950
c3620	3620

Features (Partial list)	
b	Appletalk
boot	boot image
c	CommServer lite (CiscoPro)
drag	IOS based diagnostic images
g	ISDN subset (SNMP, IP, Bridging, ISDN, PPP, IPX, Atalk)
i	IP subset (SNMP, IP, Bridging, WAN, Remote Node, Terminal Services)
n	IPX
q	Async
t	Telco return (12.0)

Format (Where the image runs in the router)	
f	flash
m	RAM
r	ROM
l	image will be relocated at run time
Compression Types	
z	zip compressed (note lower case)
x	mzip compressed
w	"STAC" compressed

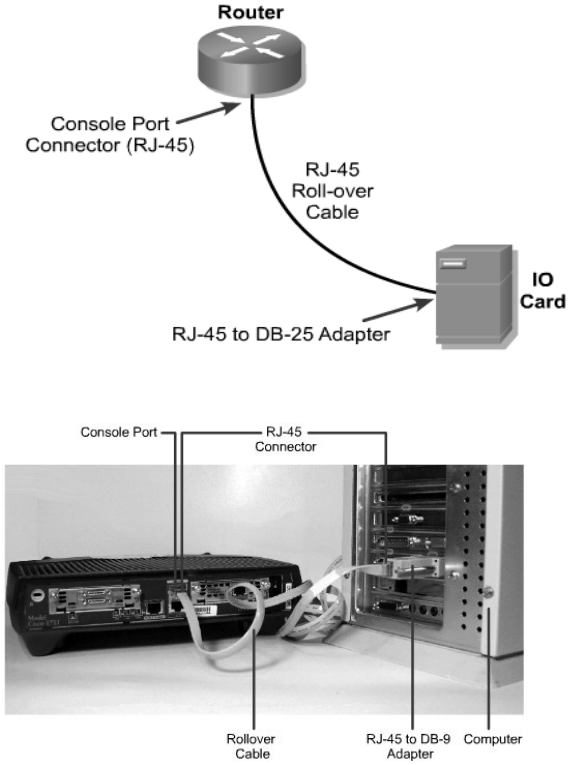
รูปที่ 3.13 แสดงความหมายรหัสของ IOS Image

ในขั้นตอนที่ 2 นี้ยังสามารถโหลด IOS ได้จากแหล่งที่เก็บอื่น ๆ ได้อีก เช่น โหลดจาก ROM ซึ่งจริง ๆ แล้วมันคือ IOS ตัวเล็ก ๆ ที่ไม่มีฟีเจอร์อะไรมากนัก หรือจะเลือกโหลด IOS ได้ผ่านทาง TFTP เซิร์ฟเวอร์ก็สามารถทำได้เช่นกัน

3. เมื่อ IOS เข้าควบคุมเราเตอร์โดยสมบูรณ์แล้ว มันจะโหลดคอนฟิกกูเรชันไฟล์จาก NVRAM เข้าไปทำงานในลำดับถัดไป เนื่องจากคอนฟิกกูเรชันไฟล์จะเก็บคำสั่งต่าง ๆ ที่สั่งให้เราเตอร์ทำอะไรบ้างเช่น ใช้โพลโตคอลหาเส้นทางแบบ RIP เป็นต้น ในส่วนนี้ก็สามารถทำการโหลดคอนฟิกไฟล์ได้จากแหล่งข้อมูลอื่น ๆ เช่นเดียวกัน เช่น โหลดมาจาก TFTP เซิร์ฟเวอร์ ในทางกลับกัน เราสามารถแบ็คอับคอนฟิกกูเรชันไฟล์ไปไว้ยังที่ปลอดภัยเพื่อไว้กรณี เผลอลบคอนฟิกไฟล์ทิ้ง ก็สามารถนำคอนฟิกที่แบ็คอับไว้มาทำงานต่อได้ทันที
4. เมื่อ IOS โหลดคอนฟิกกูเรชันไฟล์มาทำงานเสร็จแล้วก็จะขึ้นข้อความพร้อมที่จะรับคำสั่งให้ทำงานต่อไป ซึ่งก็เป็นการจบกระบวนการเริ่มต้นการทำงานของเราเตอร์

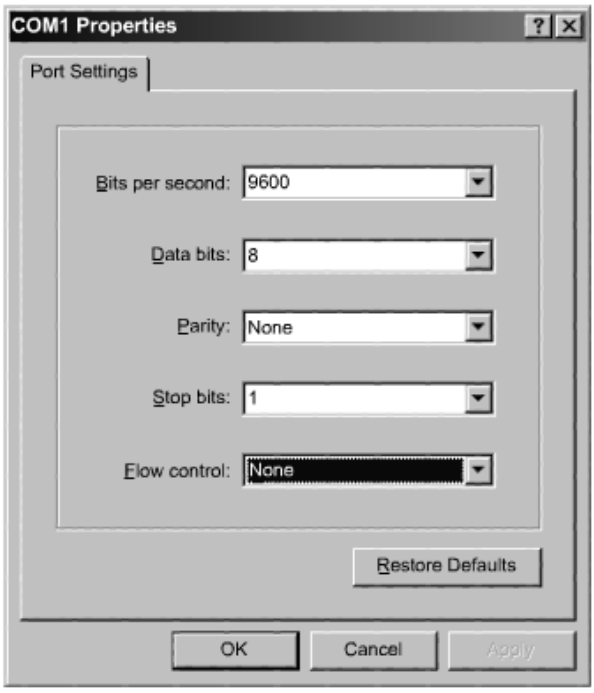
การเชื่อมต่อเพื่อคอนฟิกเราเตอร์ผ่านคอนโซล (Console Port)

จากรูปที่ 3.14 แสดงการเชื่อมต่อ PC เข้ากับเราเตอร์เพื่อคอนฟิก ปกติจะเชื่อต่อโดยผ่าน Serial Port ของ PC แต่ปัจจุบันพอร์ต serial หายากส่วนมากจะเป็น USB ก็จำเป็นจะต้องหาอุปกรณ์ที่แปลงจาก USB เป็น serial จากนั้นก็จะต่อผ่านหัวต่อ RJ-45 to DB9 ด้วยสายชนิด Rollover เข้ากับ Console port ของเราเตอร์ด้วยหัวแบบ RJ-45 ดังรูป 3.14



รูปที่ 3.14 แสดงการเชื่อมต่อ PC กับเราเตอร์เพื่อทำการคอนฟิกอุปกรณ์เราเตอร์

เมื่อต่อสายเรียบร้อยแล้ว ให้เปิดเครื่อง PC ส่วนการคอนฟิกจะต้องใช้โปรแกรมประเภท HyperTerminal หรือ SecureCRT แล้วเซ็ตค่าต่าง ๆ ดังรูปที่ 3.15, 3.16



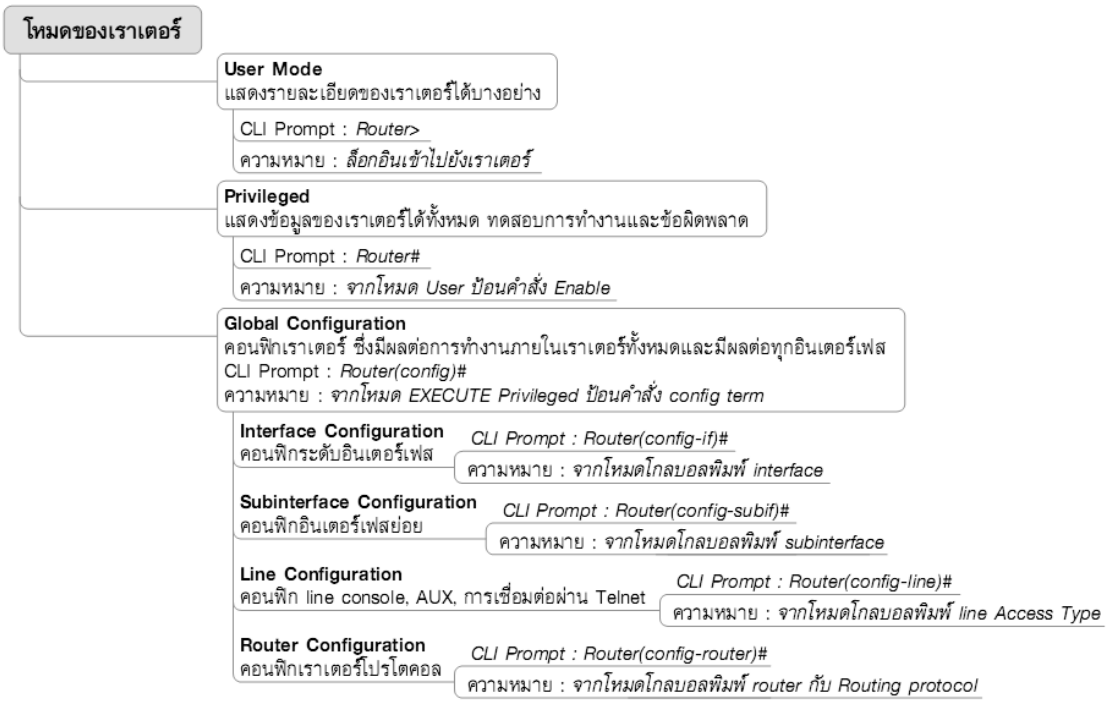
รูปที่ 3.15 การเซ็ตค่าต่าง ๆ เพื่อเชื่อมต่อกับเราเตอร์ผ่านทาง Console Port

```
Router con0 is now available.

Press RETURN to get started.

User Access Verification
Password:
Router>  <------------------ User-Mode Prompt
```

รูปที่ 3.16 เราเตอร์พร้อมรับคำสั่ง

โหมดการทำงานของเราเตอร์ (Cisco Router Modes)

เราเตอร์ของ Cisco จะมีโหมดการทำงานหลายโหมด ดังรูป 3.17 เหตุผลที่เป็นเช่นนี้เนื่องจากเรื่องของความปลอดภัยเป็นหลัก บางโหมดจะยอมให้ผู้ใช้งานสามารถคอนฟิกได้ทุก ๆ อย่าง บางโหมดก็ยอมให้ทำงานได้บางอย่างเท่านั้น ในหัวข้อนี้จะกล่าวถึงการทำงานของโหมดต่าง ๆ ที่มีอยู่ในเราเตอร์ เมื่อเราเตอร์พร้อมที่จะรับคำสั่งจะแสดง Prompt ให้ผู้ใช้ป้อนคำสั่ง ลักษณะคล้าย ๆ กับ Shell ของระบบปฏิบัติการยูนิกส์หรือลีนุกส์ แต่ Cisco เรียกว่า CLI (Command Line Interface) คือการประมวลผลคำสั่งจากผู้ใช้งานผ่านทาง "การคีย์ข้อมูลเป็นลักษณะบรรทัดต่อบรรทัด" คำสั่งที่ป้อนเข้าไปให้กับเราเตอร์จะมีผลทันที นั่นคือเราเตอร์จะประมวลผลคำสั่งทันทีเมื่อมีการกดคีย์ Enter



รูปที่ 3.17 โหมดการทำงานของเราเตอร์

User Mode

เป็นโหมดที่ระบบให้สิทธิ์บางอย่างเท่านั้น เช่น ดูข้อมูลของแต่ละอินเตอร์เฟส ปริมาณข้อมูลที่ส่งและรับ การใช้งานโหมดนี้จะทำงานได้จำกัดเท่าที่จำเป็นเท่านั้น เมื่อเข้าสู่การทำงานของโหมดนี้สังเกตุง่าย ๆ ได้จาก Prompt จะมีลักษณะเป็น ***Router>***

Privileged Mode

เป็นโหมดที่สามารถทำงานได้สูงมากขึ้นคือ สามารถแสดงข้อมูลของทุก ๆ อินเตอร์เฟส แสดงรันนิ่งคอนฟิก แสดงปริมาณการรับส่งข้อมูล ปริมาณการใช้งานของ CPU หน่วยความจำ โปรเซสที่ทำงานอยู่ และสามารถตรวจสอบข้อผิดพลาดของข้อมูลได้ว่าเกิดจากสาเหตุใด วิธีที่จะเข้ามาทำงานในโหมดนี้โดยการพิมพ์คำสั่ง enable จากโหมดของ User เมื่อเข้าสู่การทำงานของโหมดนี้สังเกตุง่าย ๆ ได้จาก Prompt จะมีลักษณะเป็น ***Router#***

ตัวอย่าง Router>enable ← Enter

Router#

Global Configuration Mode

ในโหมด Privileged นั้นไม่สามาทำการคอนฟิกให้เราเตอร์สามารถทำงานได้ทั้งหมดทุกอย่างจะทำได้เฉพาะบางอย่างเช่น เพิ่มดาต้าเบสของ VLAN เป็นต้น แต่สำหรับโหมดนี้จะสามารถคอนฟิกเราเตอร์ได้เพิ่มขึ้น เช่น การสร้าง user การเซ็ตรหัสผ่าน เป็นต้น การเข้าสู่การทำงานของโหมดนี้ทำ

ได้ด้วยการพิมพ์ configuration terminal ที่โหมด Privileged เมื่อเข้าสู่การทำงานของโหมดนี้สังเกตุง่าย ๆ ได้จาก Prompt จะมีลักษณะเป็น ***Router(config)#***

ตัวอย่าง Router#configuration terminal ← Enter

Router(config)#

Interface Configuration Mode

เป็นโหมดการทำงานเสริมเพิ่มขึ้นจากโหมดโกบอล สำหรับคอนฟิกเราเตอร์ในส่วนที่เกี่ยวข้องกับอินเตอร์เฟส เช่น การกำหนดไอพีแอดเดรสให้แต่ละอินเตอร์เฟส การกำหนดความเร็วในการรับส่งข้อมูล การกำหนด ACL (ทำคล้าย ๆ ไฟล์วอล์ของเราเตอร์) เป็นต้น การเข้าสู่โหมดนี้ทำได้ด้วยการพิมพ์ Interface TYPE Slot/Port เช่น interface serial 0/1 ที่โหมด Configuration เมื่อเข้าสู่การทำงานของโหมดนี้สังเกตุง่าย ๆ ได้จาก Prompt จะมีลักษณะเป็น ***Router(config-if)#***

ตัวอย่าง Router(config)#interface serial 0/0 ← Enter

Router(config-if)#

Subinterface Configuration Mode

ทำงานคล้ายกับโหมดของ Interface Configuration แต่ Cisco เราเตอร์จะยอมให้ในแต่ละอินเตอร์เฟสสามารถทำอินเตอร์เฟสซ้อนเข้าไปได้อีก เช่น

Interface serial 0/0 ← Interface

Interface serial 0/0.100 ← subinterface

การเข้าสู่โหมดนี้ทำได้ด้วยการพิมพ์ Interface TYPE Slot/Subinterfce จากโหมด Configuration เมื่อเข้าสู่การทำงานของโหมดนี้สังเกตุง่าย ๆ ได้จาก Prompt จะมีลักษณะเป็น ***Router(config-subif)#***

ตัวอย่าง Router(config)# interface serial 0/0.100 ← Enter

Router(config-subif)#

Line Configuration Mode

เป็นโหมดที่ทำงานเกี่ยวกับการสร้างช่องทางสำหรับเข้ามาคอนฟิกตัวอุปกรณ์เราเตอร์ เช่น การคอนฟิกคอนโซล การคอนฟิกเกี่ยวกับ Telnet เป็นต้น การเข้าสู่โหมดนี้ทำได้ด้วยการพิมพ์ line VTY 0 4 (กรณี Telnet) จากโหมด Configuration เมื่อเข้าสู่การทำงานของโหมดนี้สังเกตุง่าย ๆ ได้จาก Prompt จะมีลักษณะเป็น ***Router(config-line)#***

ตัวอย่าง Router(config)# line VTY 0 4 ← Enter

Router(config-line)#

Router Configuration Mode

ทำหน้าที่หลักคือการคอนฟิกโพรโทคอลสำหรับหาเส้นทาง (Routing Protocol) เช่น rip igrp eigrp ospf bgp เป็นต้น การเข้าสู่โหมดนี้ทำได้ด้วยการพิมพ์ Router PROTOCOL_TYPE จากโหมด Configuration เมื่อเข้าสู่การทำงานของโหมดนี้สังเกตุง่าย ๆ ได้จาก Prompt จะมีลักษณะเป็น ***Router(config-protocol)#***

ตัวอย่าง Router(config)# router ← Enter

Router(config-router)# rip ← Enter

TIPS : *เมื่อต้องการออกจาก Configuration โหมดใดโหมดหนึ่งเพียง 1 ชั้นให้ใช้คำสั่ง exit แต่ถ้าต้องการออกมาถึงโหมด Privileged ให้ใช้คำสั่ง end*

TIPS: *ถ้าจำคำสั่งของเราเตอร์ไม่ได้ให้ใช้คำสั่ง ? เช่น rou?*

TIPS: *คำสั่งของเราเตอร์สามารถเขียนแบบย่อ ๆ ก็ได้ ถ้าคำนั้นไม่ซ้ำกับคำอื่น ๆ เช่น config term (configuration terminal)*

นอกจากโหมดการทำงานที่กล่าวมาทั้งหมดแล้วยังมีโหมดที่ถูกซ้อนไว้อีก 3 โหมดเพื่อใช้สำหรับทำงานบางอย่างที่พิเศษนอกเหนือไปจากนี้ เช่น การกู้รหัสผ่าน การเลือก IOS ตัวใหม่เป็นต้น

Set up Mode

เป็นโหมดที่ใช้เมื่อเริ่มการทำงานของเราเตอร์ใหม่ครั้งแรกหรือพิมพ์คำสั่ง Setup ที่โหมด Configuration หรือเมื่อมีการเคลียร์ค่าของคอนฟิกบนเราเตอร์ทั้งหมดออก ด้วยคำสั่ง Write Erase แล้วเริ่มบู๊ตระบบใหม่ ดังรูป 3.18 เป็นต้น

```
#setup

--System Configuration Dialog--
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Continue with configuration dialog? [yes/no].

First, would you like to see the current interface summary?
[yes/no]

Interface    IP-Address   OK?   Method    Status   Protocol
TokenRing0   unassigned   NO    not set    down     down
Ethernet0    unassigned   NO    not set    down     down
Serial0      unassigned   NO    not set    down     down
Fddi0        unassigned   NO    not set    down     down
```

รูปที่ 3.18 Set up Mode

ROM Monitor

ใช้สำหรับแก้ไขค่าคอนฟิกของเราเตอร์ในกรณีที่เกิดการทำงานที่ไม่เป็นปกติ เช่น การเปลี่ยนแปลงการ Boot การเปลี่ยนรหัสผ่าน และการเซ็ตค่าของระบบต่าง ๆ วิธีการเข้าสู่โหมดนี้โดยขณะที่เราเตอร์เริ่มต้น Boot ผ่านไปประมาณ 60 วินาที ให้กดปุ่ม CTL + Break เมื่อเข้าสู่โหมดนี้ Prompt จะมีลักษณะเป็น ***rommon>*** หรือ ***>***

Subset IOS (ROM IOS หรือ RxBoot)

ปกติจะใช้ในกรณีที่ต้องการปรับปรุงระบบ Prompt จะเป็น ***Router(boot)>***

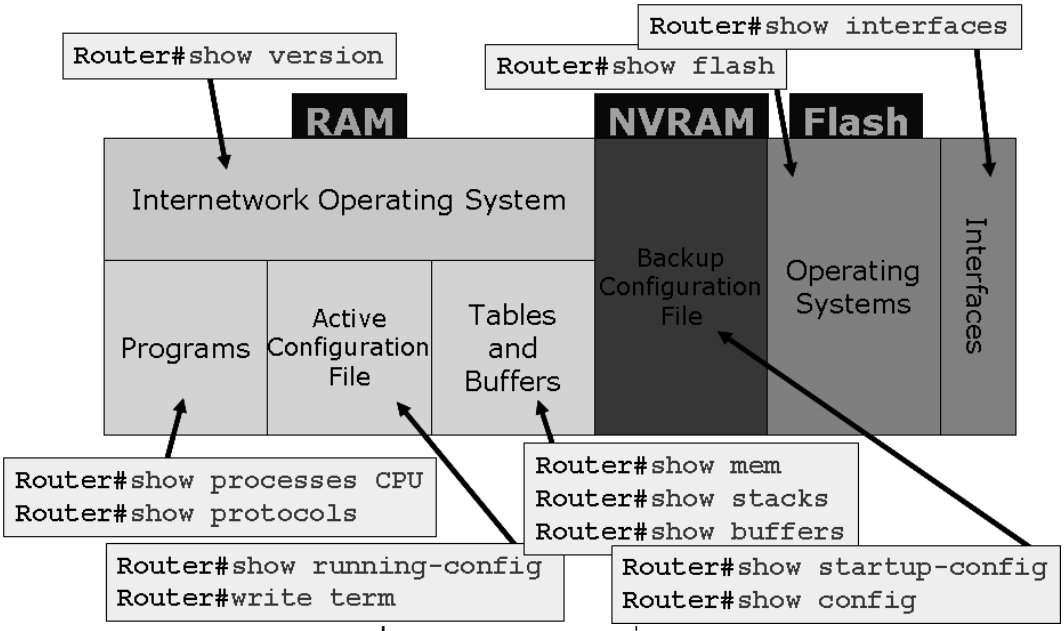
การแสดงข้อมูลต่าง ๆ ของเราเตอร์ด้วยคำสั่ง SHOW

เราเตอร์ของ Cisco จะใช้คำสั่ง show ดังตารางที่ 3.1 เมื่อต้องการแสดงค่าต่าง ๆ ของเราเตอร์ เช่น ถ้าต้องการแสดงเวอร์ชันของ IOS เราสามารถใช้คำสั่ง show version ที่โหมด Privileged จากรูปข้างล่างแสดงถึงคำสั่ง show ที่สัมพันธ์กับส่วนต่าง ๆ ของเราเตอร์ ตัวอย่างเช่นถ้าป้อนคำสั่ง show version เราเตอร์จะดึงข้อมูลมาจากในหน่วยความจำหลักที่เก็บ IOS อยู่มาแสดงเป็นต้น ดังรูปที่ 3.19

ตารางที่ 3.1 รูปแบบคำสั่ง show แบบต่าง ๆ

คำสั่ง	ส่วนที่ถูกแสดง	ความหมาย
Show version	ข้อมูลของ IOS ในหน่วยความจำหลัก	แสดงเวอร์ชันของ IOS, ปริมาณหน่วยความจำหลัก จำนวนของหน่วยความจำ flash เป็นต้น
Show interfaces	ข้อมูลของอินเตอร์เฟส	แสดงข้อมูลรายละเอียดต่าง ๆ ของอินเตอร์เฟสนั้น ๆ
Show flash	ข้อมูลของ flash	แสดงข้อมูลของ IOS ที่เก็บไว้ใน flash และประมาณหน่วยความจำ flash
Show process cpu	ข้อมูลเกี่ยวกับโปรแกรมที่กำลังทำงาน	แสดงเปอร์เซ็นการใช้งานของ CPU
Show mem	ข้อมูลเกี่ยวกับหน่วยความจำหลัก	แสดงเปอร์เซ็นการใช้งานของหน่วยความจำหลัก
Show protocols	ข้อมูลของ Routing โพรโทคอล	แสดงถึงโพรโทคอลที่ทำงานอยู่ปัจจุบันว่าเป็นแบบใดพร้อมทั้งรายละเอียดต่าง ๆ
Show running-config	ข้อมูลของคอนฟิกที่กำลังทำงานในปัจจุบัน	แสดงรายการทั้งหมดของคอนฟิกกูเรชันไฟล์ที่กำลังทำงานอยู่ในปัจจุบันในหน่วยความจำหลัก
Show stack	ข้อมูลของ stack	แสดงข้อมูลทั้งหมดของ stack ที่กำลังใช้งานอยู่ว่าใช้งานเป็นอย่างไร
Show buffers	ข้อมูลของบัฟเฟอร์	แสดงการใช้งานบัฟเฟอร์ของเราเตอร์

Show startup-config	ข้อมูลของไฟล์คอนฟิกที่เก็บอยู่ใน NVRAM	แสดงข้อมูลของไฟล์คอนฟิกที่ถูกเก็บในลักษณะกึ่งถาวรโดยต้องมีการบันทึกจาก running-config มาเก็บไว้



รูปที่ 3.19 แสดงการใช้คำสั่ง show

การ Boot เราเตอร์แบบต่าง ๆ ผ่านทางรีจีสเตอร์ (Register)

เราสามารถเปลี่ยนวิธีการ Boot ระบบของเราเตอร์ได้โดยการเซ็ตค่าผ่านรีจีสเตอร์ ซึ่งปกติแล้วจะไม่ได้ทำบ่อยนัก สถานการณ์ที่จะทำก็คือกรณีที่ IOS เกิดความเสียหาย หรือจำรหัสผ่านไม่ได้ เป็นต้น ชนิดของการ Boot สามารถดูได้จากตารางที่ 3.2

ตารางที่ 3.2 ชนิดของการ Boot เราเตอร์โดยเซ็ตค่าผ่านทาง Register

Boot Field Value	Meaning
0x**0**---	Bypass network booting (TFTP)
0x**2**---	Attempt to boot the IOS from a network TFTP server first.
0x2100 or 0x0100	Boot to ROM monitor mode
0x2101 or 0x0101	Boot to subset IOS
0x2102 – 0x210F Or 0x0102 – 0x010F	Boot according to boot commands in saved configuration (default boot)
0x2142 – 0x214F Or 0x0142 – 0x014F	Boot default ROM software if network boot fails & Ignore NVRAM contents.

ขั้นตอนของการ Boot ผ่านรีจีสเตอร์

วิธีที่ 1 เมื่อสามารถเข้าระบบได้

1. เข้าสู่ Configuration Mode → ***Router>enable*** → ***Router#configuration terminal***
2. เซ็ตค่าของ register เป็นค่าที่ต้องการให้ boot แบบใด เช่น ต้องการให้ boot ด้วย subset IOS → ***Router(config)#config-register 0x101***
3. ออกจากโหมดคอนฟิกด้วย ***exit***
4. ทดลอง show version ดูก่อนว่าเปลี่ยนแปลงหรือไม่ก่อนทำการ boot เราเตอร์ด้วยคำสั่ง ***reload***

วิธีที่ 2 เมื่อยังไม่สามารถเข้าระบบได้

1. ปิดเราเตอร์แล้วเปิดเครื่องใหม่
2. รอเวลาให้เครื่องทำงานก่อนประมาณ 60 วินาทีแล้วจึงกดปุ่ม CTL กับ Break พร้อม ๆ กัน
3. ระบบจะเข้าสู่ ROM โดยแสดง Prompt เป็น ***rommon>***
4. ใส่ค่าให้กับรีจีสเตอร์ตามที่ต้องการ สมมุติต้องการ boot เป็น 0x2142 → ***rommon>confreg 0x2142***
5. สั่งคำสั่งให้เริ่มการทำงานใหม่ → ***rommon>reset***

การ Backup และ Restore IOS และคอนฟิกกูเรชันไฟล์

IOS และ คอนฟิกกูเรชันไฟล์เป็นส่วนที่สำคัญเป็นอย่างยิ่งที่จะทำให้เราเตอร์สามารถทำงานได้อย่างราบลื่น แต่ก็มีบ่อยครั้งที่อุปกรณ์เกิดเสียขึ้นมาแบบไม่ทันตั้งตัว เช่น เกิดจากไฟฟ้ากระชาก ทำให้อุปกรณ์ต่าง ๆ ไม่สามารถทำงานต่อไปได้ ถ้าเกิดเหตุการณ์เช่นนี้ ก็จะต้องทำการเปลี่ยนอุปกรณ์กันใหม่ ปัญหาคือ "แล้วคอนฟิกกูเรชันไฟล์ล่ะ?" จะเอามาจากไหน ถ้าตอบว่าก็คีย์เข้าไปใหม่ซิ วิธีนี้จะเหมาะสำหรับระบบหรือองค์กรที่มีขนาดเล็ก ๆ แต่ถ้าองค์กรที่มีขนาดใหญ่ ๆ ล่ะ คำสั่งในคอนฟิกไฟล์อย่างต่ำ ๆ ก็มีเป็นร้อย แล้วจะจำได้หรือ ? วิธีแก้ไขที่ดีที่สุดคือ "แบ็คอัพ" ไว้ที่เครื่องใดเครื่องหนึ่งที่ปลอดภัย โดยการติดตั้งโปรแกรมประเภท TFTP หรือ FTP ไว้ เมื่อเกิดเหตุกราณ์ฉุกเฉินก็ copy ข้อมูลจากเครื่องที่ทำสำรองไว้เข้าไปในอุปกรณ์ตัวใหม่ได้ทันที ข้อมูลที่จะต้องเก็บมี 2 ส่วนคือ

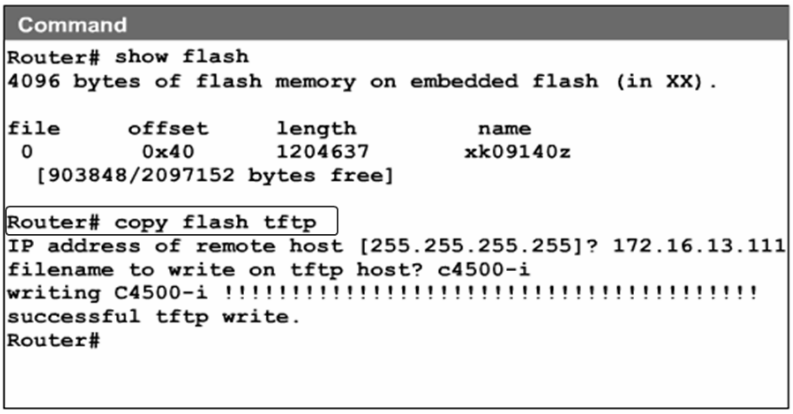
1. **Backup และ Restore IOS**

อย่างที่ทราบแล้วว่า IOS จะทำหน้าที่เป็น OS ของระบบ แต่ก็จะมีคำถามว่าเมื่ออุปกรณ์ตัวใหม่มามันจะมี OS มาให้อยู่แล้วทำไมจึงต้อง Backup เหตุผลเนื่องจาก IOS ที่เราใช้งานจะมีการปรับปรุงอยู่เรื่อย ๆ บางครั้งอาจจะต้องมีการ Patch (คือการแก้ไขจุดบกพร่องของซอฟท์แวร์เมื่อพบ Bug) ดังนั้น IOS ของเราอาจจะมีการอัพเดทอะไรบางอย่างเข้าไป ซึ่งถ้าเอา IOS ตัวใหม่มา

ทำงานอาจจะไม่สามารถทำงานได้เต็มประสิทธิภาพก็เป็นได้ และอีกกรณีหนึ่งคือ กรณีที่ไม่ได้นำอุปกรณ์ใหม่แกะกล่องมาใช้ แต่หาอุปกรณ์ตัวเก่าแต่มาใส่ IOS ใหม่ให้มันก็จะทำได้อย่างรวดเร็ว

วิธีการ Backup IOS

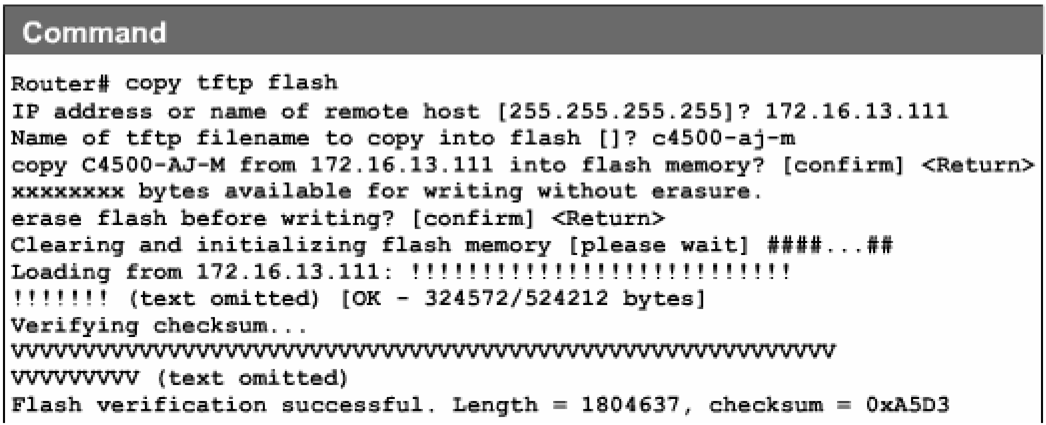
1. ทดลองแสดง IOS ว่าชื่ออะไร รุ่นอะไรดูก่อนด้วยคำสั่ง ***show flash*** ที่ enable mode (privileged) ดังรูปที่ 3.20
2. จากนั้นก็ใช้คำสั่ง ***copy flash tftp*** **←** ***enter***
3. ใส่หมายเลขไอพีแอดเดรสของ TFTP เซิร์ฟเวอร์ แล้วกด enter ถ้าไม่ผิดพลาดอะไร โปรแกรมจะเริ่มทำการ copy IOS ไปยัง TFTP เซิร์ฟเวอร์ สถานะการณ์ก๊อบปี้จะเป็นสัญลักษณ์ !!!!!!!


```
Command
Router# show flash
4096 bytes of flash memory on embedded flash (in XX).

file     offset     length         name
 0        0x40      1204637       xk09140z
  [903848/2097152 bytes free]

Router# copy flash tftp
IP address of remote host [255.255.255.255]? 172.16.13.111
filename to write on tftp host? c4500-i
writing C4500-i !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
successful tftp write.
Router#
```

รูปที่ 3.20 แสดงคำสั่งสำหรับ Backup IOS

วิธีการ Restore IOS

ทำในลักษณะเดียวกันแต่มีทิศทางตรงกันข้าม ดังรูปที่ 3.21


```
Command
Router# copy tftp flash
IP address or name of remote host [255.255.255.255]? 172.16.13.111
Name of tftp filename to copy into flash []? c4500-aj-m
copy C4500-AJ-M from 172.16.13.111 into flash memory? [confirm] <Return>
xxxxxxxx bytes available for writing without erasure.
erase flash before writing? [confirm] <Return>
Clearing and initializing flash memory [please wait] ####...##
Loading from 172.16.13.111: !!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!! (text omitted) [OK - 324572/524212 bytes]
Verifying checksum...
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVVVVV (text omitted)
Flash verification successful. Length = 1804637, checksum = 0xA5D3
```

รูปที่ 3.21 แสดงคำสั่งสำหรับ Restore IOS

2. Backup และ Restore คอนฟิกกูเรชันไฟล์

running-config เป็นไฟล์ที่ถูกโหลดมาจาก NVRAM เข้ามายังหน่วยความจำหลักเพื่อทำงาน เมื่อมีการรีเซ็ตเครื่องจะทำให้ข้อมูลที่อยู่ในหน่วยความจำหลักหายไป ส่งผลให้ต้องโหลดไฟล์คอน

ฟิกจาก NVRAM มาทำงานใหม่อีกครั้ง ขณะที่เราป้อนคำสั่งหนึ่ง ๆ ลงไปที่ CLI จะส่งผลทันทีต่อเราเตอร์ เช่น เราสั่ง shutdown อินเตอร์เฟส จะทำให้อินเตอร์เฟสดังกล่าวนั้นหยุดทำงานทันที แต่ถ้าเรายังไม่มีการเขียนข้อมูลทับลงไปที่ NVRAM เมื่อมีการรีเซ็ตเราเตอร์ใหม่อีกครั้งจะทำให้อินเตอร์เฟสที่ถูก shutdown นั้นจะกลับมาทำงานอีกครั้งหนึ่ง ดังรูปที่ 3.22, 3.23

Command

```
tokyo#copy running-config tftp
Remote host []? 131.108.2.155
Name of configuration file to write[tokyo-config]?tokyo.2
Write file tokyo.2 to 131.108.2.155? [confirm] y
Writing tokyo.2 !!!!!! [OK]
tokyo#
```

รูปที่ 3.22 การ Backup คอนฟิกกูเรชันไฟล์

Command

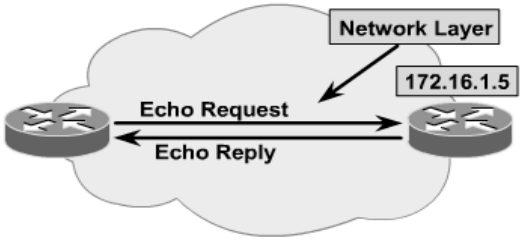
```
tokyo#copy tftp running-config
Host or network configuration file [host]?
IP address of remote host [255.255.255.255]? 131.108.2.155
Name of configuration file [Router-config]? tokyo.2
Configure using tokyo.2 from 131.108.2.155? [confirm] y
Booting tokyo.2 from 131.108.2.155:!! [OK-874/16000 bytes]
tokyo#
```

รูปที่ 3.23 การ Restore คอนฟิกกูเรชันไฟล์

การทดสอบการเชื่อมต่อระบบด้วยคำสั่ง ping

ในการตรวจสอบระบบเครือข่ายว่าสามารถทำงานได้จริงหรือยัง ส่วนมากจะใช้โปรแกรมตัวหนึ่งคือ ping สำหรับตรวจสอบ โดยหลักการคือมันจะส่งแพ็กเก็ตจำนวนหนึ่ง (Echo Request)ไปยังปลายทาง ส่วนปลายทางเมื่อได้รับแพ็กเก็ตแล้วจะส่งข้อมูลกลับมา (Echo Reply) ต่อจากนั้นเรา

เตอร์ต้นทางจะประเมินค่าถ้าข้อมูลที่กลับมามีค่าตามที่กำหนดไว้ (TTL) ก็เป็นอันว่าสามารถสื่อสารกันได้ รูปแบบคำสั่งคือ ***ping หมายเลขไอพีปลายทาง*** ดังรูปที่ 3.24

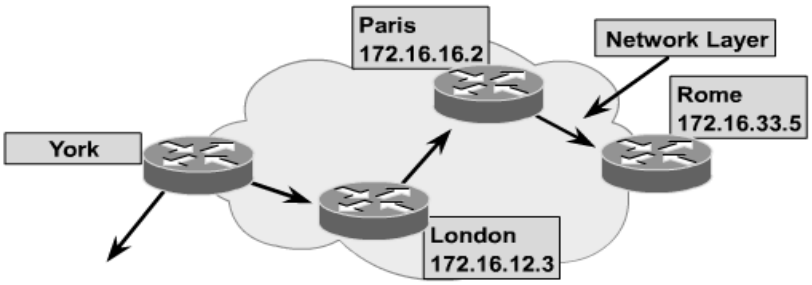



```
Router>ping 172.16.1.5
Type escape sequence to abort.
Sending 5, 100 byte ICMP Echos to 172.16.1.5,
timeout is 2 seconds:
!!!!!
Success rate is 100 percent,
round-trip min/avg/max = 1/3/4 ms
Router>
```

รูปที่ 3.24 แสดงการใช้คำสั่ง ping

การทดสอบการเชื่อมต่อระบบด้วยคำสั่ง trace

หลักการคล้ายกับ ping แต่ trace จะแสดงข้อมูลในรูปของเส้นทางจากต้นทางไปยังปลายทางว่าได้ผ่านเราเตอร์ตัวใดบ้างในเครือข่ายพร้อมกับระยะเวลาที่มันเดินทางไปถึงด้วย รูปแบบคำสั่งคือ ***trace หมายเลขไอพีหรือชื่อของเราเตอร์*** ดังรูปที่ 3.25




```
York#trace ROME
Type escape to abort.
Tracing the route to Rome (172.16.33.5)
  1 LONDON (172.16.12.3) 1000 msec 8 msec 4 msec
  2 PARIS (172.16.16.2) 8 msec 8msec 8msec
  3 ROME (172.16.35.5) 8msec 8msec 4msec

York#
```

รูปที่ 3.25 แสดงการใช้คำสั่ง trace

TIPS: *ผู้ดูแลระบบสามารถเข้าไปคอนฟิกเราเตอร์ได้ผ่าน Remote คือการใช้คำสั่ง* ***Telnet ชื่อโฮสต์***

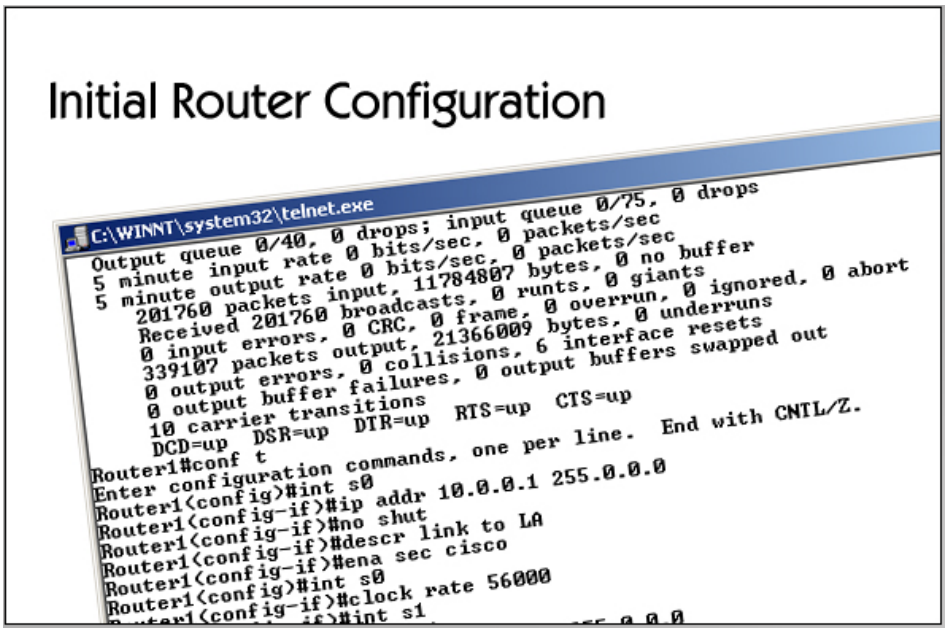
TIPS: *ผู้ใช้งานสามารถสลับ session ของ Telnet ที่เปิดไว้หลาย ๆ session ด้วยการกดคีย์* ***ctl + shift + 6 + x***

แบบฝึกหัดท้ายบท

1. การเข้าถึงระบบเครือข่ายมี 3 ระดับประกอบไปด้วยอะไรบ้าง และแต่ละประเภทแตกต่างกันอย่างไร
2. เราเตอร์มีหน้าที่อย่างไรบนระบบเครือข่าย พร้อมยกตัวอย่าง
3. จงอธิบายส่วนประกอบต่างๆ ของเราเตอร์ต่อไปนี้ว่าทำงานอย่างไร
    - ROM
    - Flash Memory
    - NVRAM
    - RAM/DRAM
    - Router Interface
    - IOS
4. จงอธิบายหลักการทำงานของเราเตอร์มาพอเข้าใจ
5. โหมดการทำงานบนเราเตอร์มีกี่โหมด อะไรบ้าง
6. จงอธิบายคำสั่งการทำงานของเราเตอร์ต่อไปนี้มาพอเข้าใจ
    - Show version
    - Show interface
    - Show flash
    - Show process
    - Show mem
    - Show protocol
    - Show running-config
    - Show startup-config
7. จงอธิบายกระบวนการ boot ผ่านรีจีสเตอร์ มาพอเข้าใจ
8. จงอธิบายวิธีการ backup และ restore ไฟล์ configuration ว่ามีขั้นตอนการทำงานอย่างไร

บทที่ 4

เริ่มต้นการคอนฟิกเราเตอร์ (Initial Router Configuration)



- Hostname and interface descriptions
- System passwords
- Banners
- Interfaces
- Back-to-back router
- Cisco discovery protocol (CDP)

แนวคิด

ในส่วนนี้จะใช้สำหรับฝึกหัดการคอนฟิกระบบเครือข่ายผ่านตัวจำลอง เนื้อหาในแต่ละแล็บนั้นจะมีลักษณะที่ต่อเนื่องและเรียงลำดับตามความสำคัญของเนื้อหาในการติดตั้งและดูแลระบบเครือข่าย เมื่อจบในบทนี้แล้วผู้เรียนจะสามารถเข้าใจการติดตั้งและปรับแต่งอุปกรณ์เครือข่ายเบื้องต้นเพื่อนำไปสู่การปรับแต่งกับอุปกรณ์จริง

วัตถุประสงค์

1. เพื่อให้สามารถปรับแต่งอุปกรณ์ที่ใช้งานบนระบบเครือข่ายได้อย่างเหมาะสม
2. เพื่อสร้างความชำนาญในการใช้งานอุปกรณ์ที่สำคัญๆ บนระบบเครือข่าย

การตั้งค่าและปรับแต่งอุปกรณ์เครือข่าย (เช่น เร้าเตอร์ สวิทช์ เป็นต้น)

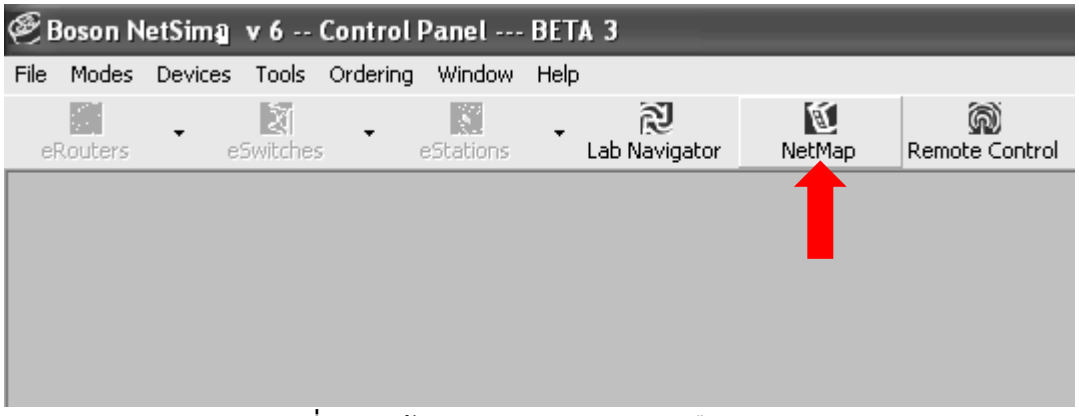
Connecting และ Logging ไปยัง Cisco Router

จุดมุ่งหมาย : เริ่มต้นเรียนรู้คอนฟิกของเราเตอร์

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัว

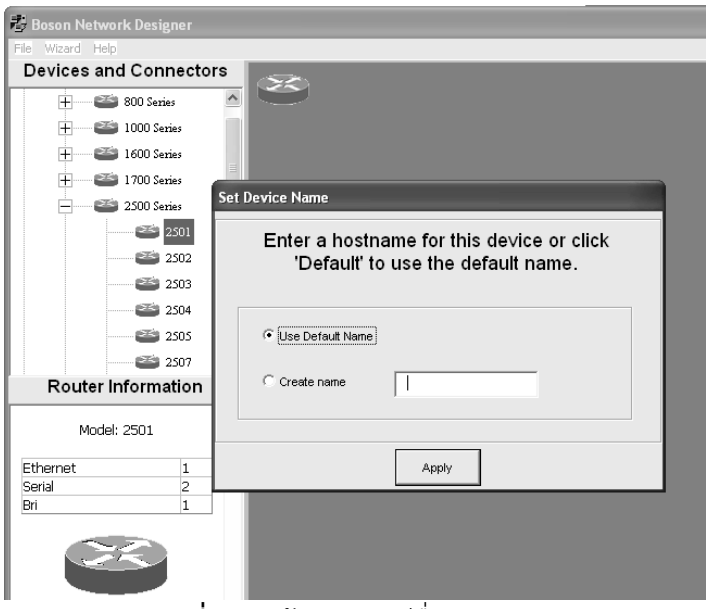
การสร้าง Network Map:

1. จากหน้าต่างหลัก (Simulator) คลิกเลือก NetMap เพื่อสร้างผังเน็ตเวิร์ค ดังรูปที่ 4.1



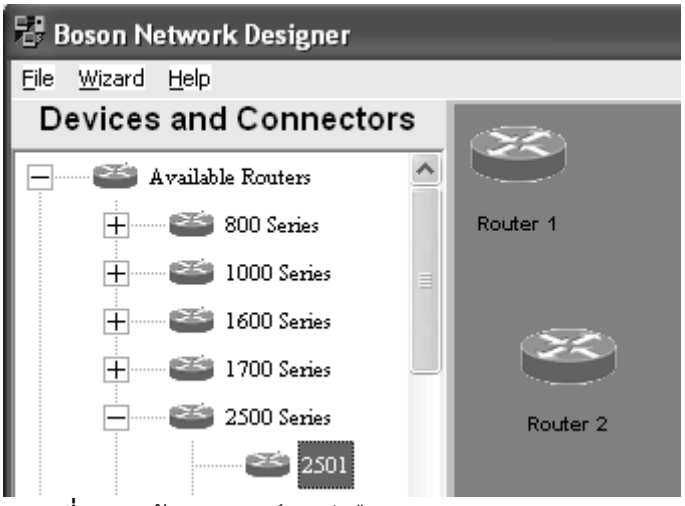
รูปที่ 4.1 หน้าต่างหลัก (Simulator) เลือก NetMap

2. ขั้นต่อไป จะปรากฏหน้าต่าง Boson Network Designer → คลิกเลือก Available Routers → คลิกเลือก 2500 series → เลือกรุ่น 2501 โดยการลากมาวางที่พื้นที่สีเทาทางด้านขวามือ → ให้เลือก Use Default Name (เราเตอร์จะชื่อ Router1) → คลิก Apply ดังรูปที่ 4.2



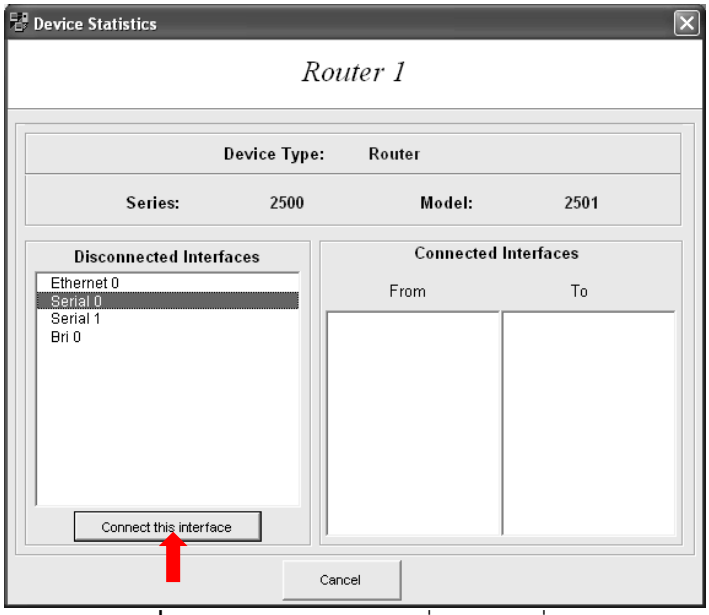
รูปที่ 4.2 สร้างเราเตอร์ชื่อ Router1

3. จะปรากฏอุปกรณ์เราเตอร์ (Router1) บนผังของเน็ตเวิร์ค จากนั้นให้สร้างเราเตอร์รุ่นเดิมอีก 1 ตัว (Router2) เนื่องจาก Boson NetSim จะยอมให้คอนฟิกต้องมีอุปกรณ์ 2 ตัวขึ้นไปเชื่อมต่อกัน ดังรูปที่ 4.3



รูปที่ 4.3 สร้างเราเตอร์ 2 ตัวคือ Router1 และ Router2

4. จากนั้นให้เชื่อมต่อเราเตอร์ 2 ตัวเข้าด้วยกันโดยการดับเบิลคลิกที่เราเตอร์ตัวใดตัวหนึ่งก็ได้ (หรือคลิกขวาที่ตัวเราเตอร์แล้วเลือก Add Connection to) ลำดับต่อไปให้เลือกอินเตอร์เฟสที่ต้องการใช้เชื่อมต่อ ให้ทำการเลือกอินเตอร์เฟสชื่อ serial 0 (สาย serial เป็นสายนำสัญญาณแบบจุดต่อจุด ส่วนใหญ่มีขนาดของแบนด์วิดธ์ไม่เกิน 2 เม็กกะบิตต่อวินาที) จากนั้นให้เลือก Connect this interface ดังรูปที่ 4.4

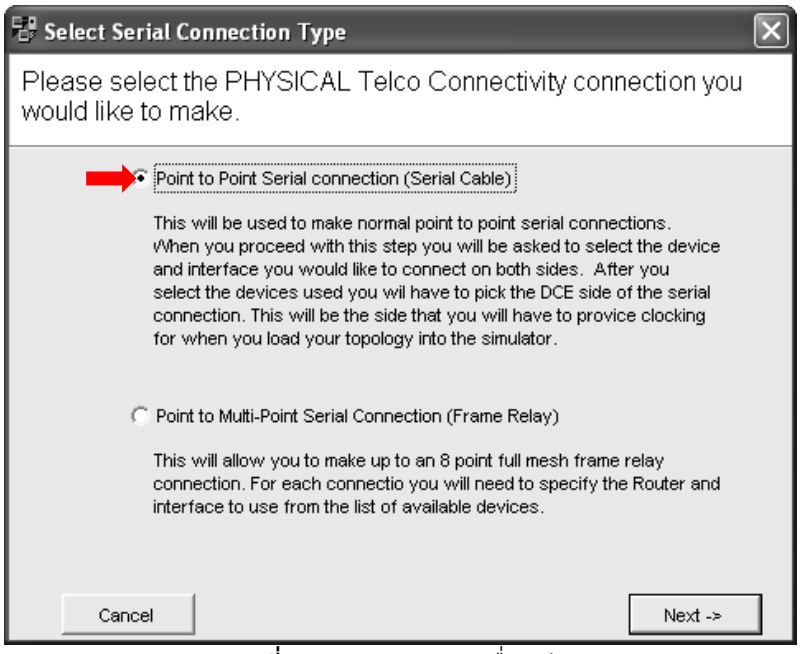


รูปที่ 4.4 เลือกอินเตอร์เฟสที่ต้องการเชื่อต่อ

5. ลำดับต่อไปให้เลือกชนิดของการเชื่อมต่อซึ่งมีอยู่ 2 แบบคือ

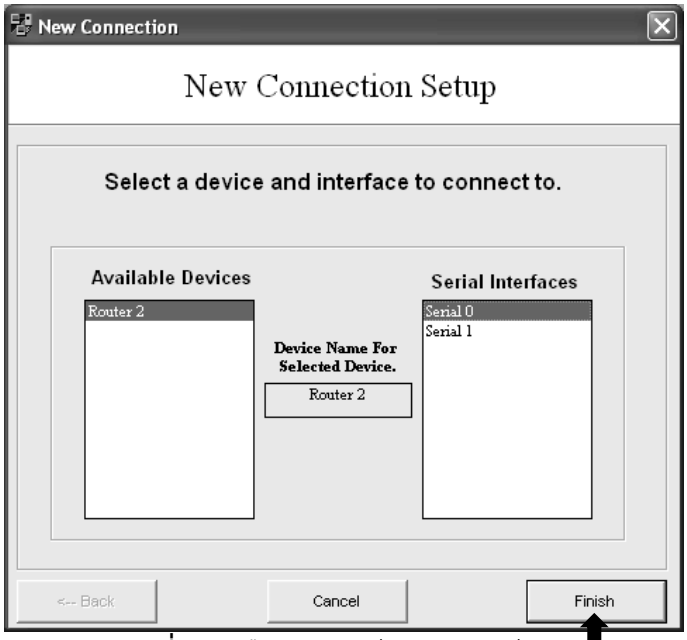
1. Point to Point Serial Connection (Serial Cable) คือสายที่มีการเชื่อมต่อกันแบบจุดต่อจุด หรือ 1 ต่อ 1

2. Point to Multi-Point Serial Connection (Frame Relay) เป็นการเชื่อมต่อแบบเฟรมรีเลย์ ซึ่งมีลักษณะ 1 ต่อ หลาย ๆ เครื่อง ให้เลือกการเชื่อมต่อเป็นแบบ Point to Point Connection แล้วคลิก Next-> ดังรูปที่ 4.5



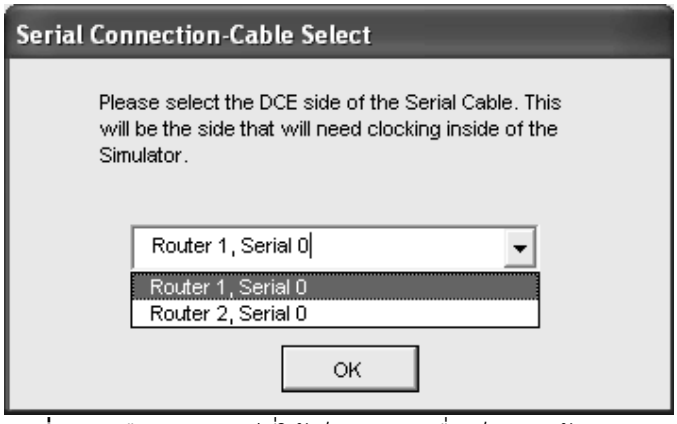
รูปที่ 4.5 ชนิดของการเชื่อมต่อ

6. จากนั้นจะปรากฏรายการของเราเตอร์ที่ต้องการจะเชื่อมต่อด้วยขึ้นมา ให้เลือกเชื่อมต่อกับอินเตอร์เฟสที่ต้องการในที่นี้ ในช่อง Available Devices เลือก Router 2 ในช่อง Serial Interfaces ให้เลือก Serial 0 เมื่อเลือกครบแล้วให้กดปุ่ม Finish ซึ่งเป็นการสิ้นสุดการเชื่อมต่ออินเตอร์เฟสของเราเตอร์ทั้งสองตัวเข้าด้วยกัน ดังรูปที่ 4.6



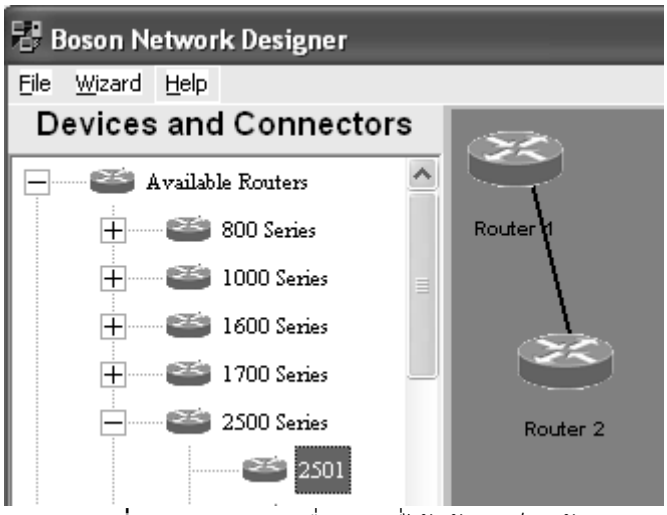
รูปที่ 4.6 เลือกเราเตอร์และอินเตอร์เฟส

7. ขั้นต่อไปจะปรากฏหน้าต่างเพื่อให้เลือกเราเตอร์ที่เป็น DCE (เราเตอร์จะใช้อินเตอร์เฟสที่เป็น DCE (Data Circuit-Terminating Equipment) ในการสร้าง Clock เพื่อให้เราเตอร์ทั้ง 2 ฝ่ายทำงานได้พร้อมเพียงกัน) ในที่นี้ให้เลือกที่ Router 1 เป็น DCE และเราเตอร์อีกฝั่งจะเรียกว่า DTE (Data Terminating Equipment) ดังรูปที่ 4.7



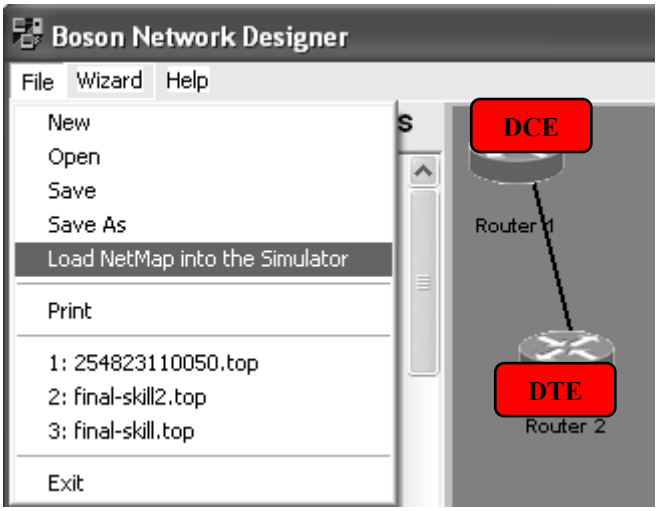
รูปที่ 4.7 เลือกเราเตอร์ที่ให้เป็น DCE เพื่อเป็นตัวสร้าง Clock

8. ขั้นตอนต่อไป ที่หน้าต่างของ NetMap ให้เลือก File → Load NetMap into Simulator เพื่อโหลดผังเน็ตเวิร์คให้ไปทำงานที่หน้าต่างหลัก (Simulator) ซึ่งจะมีเมนูแสดงข้อมูลว่าต้องการโหลดผังเน็ตเวิร์คเข้าไปยังหน้าต่างการทำงานหลักหรือไม่ (Simulator) ให้เลือก yes ดังรูปที่ 4.8



รูปที่ 4.8 แสดงการเชื่อมต่อที่ได้สร้างเสร็จแล้ว

9. เมื่อโหลดผังเน็ตเวิร์คไปยังหน้าต่างหลัก (Simulator) แล้ว เป็นอันว่าพร้อมที่จะเริ่มต้นคอนฟิกต่อไป ดังรูปที่ 4.9, 4.10



รูปที่ 4.9 แสดงวิธีการโหลดผังเน็ตเวิร์คเข้าไปยังหน้าต่างหลัก (Simulator)

รูปที่ 4.10 หน้าต่างหลัก (Simulator) พร้อมที่จะรับคำสั่งการทำงานต่อไป

หมายเหตุ : ควรจะมีการบันทึกผังเน็ตเวิร์คไว้ก่อนให้เลือก File → Save หรือ Save As (เมื่อไม่ต้องการชื่อที่โปรแกรมตั้งให้) ข้อมูลที่บันทึกจะมีนามสกุลจะเป็น .top

หมายเหตุ : ถ้าต้องการทราบว่าอินเตอร์เฟสของเราเตอร์ตัวไหนเป็น DCE ให้ใช้คำสั่ง Router#show controllers ในโหมดการทำงาน privilege ดังรูปที่ 4.11

```
Router1#sh controllers

HD unit 0, idb = 0x1AE828, driver structure at 0x1B4BA0
buffer size 1524  HD unit 0[redacted]
cpb = 0x7, eda = 0x58DC, cda = 0x58F0
RX ring with 16 entries at 0x4075800
00 bd_ptr=0x5800 pak=0x1B5E24 ds=0x4079108 status=80 pak_size=13
01 bd_ptr=0x5814 pak=0x1B85B8 ds=0x4080384 status=80 pak_size=13
02 bd_ptr=0x5828 pak=0x1B880C ds=0x4080A40 status=80 pak_size=69
```

รูปที่ 4.11 หน้าต่างหลักแสดงคำสั่ง show controllers

หน้าต่าง Simulator :

1. กดปุ่มคีย์บอร์ด *Enter* เพื่อเริ่มต้น(ถ้าต้องการดูแผนผังเครือข่ายให้เลือกที่เมนู NetMap)
2. **Router>** [กำลังอยู่ในโหมด User]
3. **Router>*enable*** [เข้าสู่โหมด privileged ด้วยคำสั่ง enable]

4. **Router#** [เข้าสู่โหมดของ privileged เรียบร้อยแล้ว]
5. **Router#*disable*** [กลับเข้าไปโหมด User]
6. **Router>*exit*** [ออกจากเราเตอร์ (ออกจาก console)]
7. **กดคีย์ enter** [เพื่อเข้าสู่โหมด User อีกครั้ง]

หมายเหตุ : ลองทดสอบคำสั่ง **Router>*ena?***

หมาเหตุ : การบันทึกข้อมูลจะมี 2 ส่วนคือที่หน้าต่าง Boson Network Designer (ใช้นามสกุล .top) และหน้าต่าง Boson Simulator (ใช้นามสกุล .nwc) ควรจะทำการตั้งชื่อให้ตรงกัน เช่น บันทึกผังเน็ตเวิร์คชื่อ lab1.top และควรบันทึกคอนฟิกไฟล์ใน simulator เป็น lab1.nwc ด้วยเช่นกัน เพราะจะเป็นการง่ายต่อการทำความเข้าใจและโปรแกรมจะทำงานได้อย่างถูกต้องด้วย

💻 เบื้องต้นเกี่ยวกับวิธีการสื่อสารของผู้ใช้งานกับเราเตอร์

จุดมุ่งหมาย : เริ่มต้นเรียนรู้คำสั่งพื้นฐานที่เกี่ยวกับโหมดการทำงานของเราเตอร์และคำสั่งที่ใช้สำหรับช่วยเหลือ

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัว

การสร้าง Network Map: ให้สร้างผังเน็ตเวิร์คเหมือนรูปที่ 4.8

หน้าต่าง Simulator:

1. **enter** [กดคีย์บอร์ด enter เพื่อเข้าสู่โหมด User]
2. **Router>*?*** [แสดงคำสั่งทั้งหมดที่ใช้งานในโหมด User] ดังรูปที่ 4.12

```
Router>?
show                    Show running system information
enable                  Turn on privileged commands
exit                    Exit from the EXEC
help                    Description of the interactive help system
disable                 Turn off privileged commands
disconnect              Disconnect an existing network connection
logout                  Exit from the EXEC
ping                    Send echo messages
terminal                Set terminal line parameters
traceroute              Trace route to destination
lock                    Lock the terminal
login                   Log in as a particular user
mrinfo                  Request neighbor and version information from a multicast router
mstat                   Show statistics after multiple multicast traceroutes
mtrace                  Trace reverse multicast path from destination to source
name-connection         Name and existing network connection
pad                     Open a X.29 PAD connection
ppp                     Start IETF Point-to-Point Protocol (PPP)
rlogin                  Open an rlogin connection
slip                    Start a Serial-line IP (SLIP)
systat                  Display information about terminal lines
tunnel                  Open a tunnel connection
udptn                   Open an updtn connection

--MORE--
```

รูปที่ 4.12 ผลลัพธ์ของคำสั่ง ?

3. **Router>*enable*** [เข้าสู่โหมด privilege]
4. **Router#*?*** [แสดงคำสั่งทั้งหมดที่ใช้งานในโหมด privilege]

5. Router#*show ?* [แสดงคำสั่งที่ใช้งานร่วมกับคำสั่ง show] ดังรูปที่ 4.13

```
Router#show ?
version                 System hardware and software stat
cdp                     CDP information
clock                   Display the system clock
flash                   display information about flash: 
history                 Display the session command histo
hosts                   IP domain-name, nameservers, and 
interfaces              Interface status and configuratio
protocols               Active network routing protocols
sessions                Information about Telnet connecti
terminal                Display terminal configuration pa
users                   Display information about termina
frame-relay             Frame-Relay information
isdn                    ISDN information
ntp                     Network time protocol
controllers             Interface controller status
running-config          Current operating configuration
startup-config          Contents of startup configuration
access-lists            List access lists
configuration           Contents of Non-Volatile memory
ip                      IP information
arp                     ARP table
isis                    IS-IS routing information
clns                    CLNS network information

--MORE--
```

รูปที่ 4.13 ผลลัพธ์ของคำสั่ง show ?

6. Router#*running-config* [แสดงคอนฟิกกูเรชันไฟล์ที่กำลังทำงานอยู่] ดังรูปที่ 4.14

```
Router#[redacted]
Building configuration...

!
Version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
!
!
ip subnet-zero
!
```

รูปที่ 4.14 ผลลัพธ์ของคำสั่ง show running-config

7. –MORE-- *<spacebar>* [more เป็นการแสดงหน้าถัดไป โดยกดคีย์ space bar]
8. Router#[*exit*|*disable*] [จากโหมด privilege ให้ออกจากโหมดนี้ด้วยคำสั่ง exit หรือ disable]

เรียนรู้การใช้คำสั่ง show

จุดมุ่งหมาย : เรียนรู้การใช้คำสั่ง show และคำสั่งที่ทำงานร่วมกับมัน

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัว

การสร้าง Network Map : ให้สร้างผังเน็ตเวิร์คเหมือน รูปที่ 4.8

หน้าต่าง Simulator :

1. **Router>** [เข้าสู่การทำงานของโหมด User]
2. **Router>*enable*** [เข้าสู่โหมด privilege]
3. **Router#*show running-config*** [แสดงคอนฟิกกูเรชันที่ทำงานอยู่ในหน่วยความจำหลัก คอนฟิกนี้จะหายไปหากมีการรีเซ็ตเราเตอร์]
4. **Router#*show flash*** [แสดงข้อมูลของระบบปฏิบัติการ (IOS) ที่เก็บอยู่บนหน่วยความจำที่เป็น flash หน่วยความจำแบบนี้ข้อมูลจะไม่สูญหายจากการรีเซ็ตระบบ] ดังรูปที่ 4.15

```
Router#[redacted]
System flash directory:
File  Length   Name/status
  1   5880916  [redacted]
[5880980 bytes used, 2507628 available, 8388608 total]
8192K bytes of processor board System flash (Read/Write)
```

รูปที่ 4.15 ผลลัพธ์ของคำสั่ง show flash

5. **Router#*show history*** [แสดงคำสั่งที่เคยใช้งานผ่านมาแล้ว CLI ของเราเตอร์จะเก็บ 10 คำสั่งที่เคยใช้งานแล้วไว้ในหน่วยความจำ]
6. **Router#*<ctl>P*** [แสดงคำสั่งที่เพิ่งใช้งานผ่านมา หรืออาจกดคีย์บอร์ดลูกศรขึ้น]
7. **Router#*<ctl>N*** [แสดงคำสั่งต่อไป กรณีที่กลับไปใช้คำสั่งย้อนหลังแล้วต้องการจะกลับไปคำสั่งล่าสุด หรือกดคีย์บอร์ดลูกศรลง]
8. **Router#*show protocols*** [แสดงโพรโทคอลค้นหาเส้นทางที่กำลังทำงานอยู่]
9. **Router#*show version*** แสดงข้อมูลหลัก ๆ ของเราเตอร์ เช่น เราเตอร์แพลทฟอร์ม, รุ่นของ IOS, เวลาที่บู๊ตระบบครั้งล่าสุด, ตำแหน่งที่อยู่ของไฟล์ ISO, หน่วยความจำ, จำนวนของอินเตอร์เฟสที่มีในระบบ, รีจีสเตอร์คอนฟิกกูเรชัน]
10. **Router#*show* clock** [แสดงสัญญาณนาฬิกาของเราเตอร์] ดังรูปที่ 4.16

```
Router#show clock
*00:38:35.755 UTC Mon Mar 1 1993
Router#
```

รูปที่ 4.16 ผลลัพธ์ของคำสั่ง show clock

11. **Router#*show hosts*** [แสดงรายชื่อของโฮสต์ และและแอดเดรสของมันทั้งหมด ในเบื้องต้นอาจจะไม่ปรากฏชื่อหรือข้อมูลใดๆ เนื่องจากยังไม่มีการคอนฟิก] ดังรูปที่ 4.17

```
Router#show hosts
Default domain is not set
Name/address lookup uses static mappings

Host Flags Age Type Address(es)
Router#
```

รูปที่ 4.17 ผลลัพธ์ของคำสั่ง show hosts

12. Router#*show users* [แสดงรายชื่อผู้ใช้งานทั้งหมดที่กำลังใช้งานอยู่ ในเบื้องต้นอาจจะไม่ปรากฏชื่อหรือข้อมูลใดๆ เนื่องจากยังไม่มีการคอนฟิก] ดังรูปที่ 4.18

```
Router#show users
Line User Host(s) Idle Location
* 0 con 0 idle 00:00:00
```

รูปที่ 4.18 ผลลัพธ์ของคำสั่ง show users

13. Router#*show interfaces* [แสดงรายละเอียดของอินเตอร์เฟสทั้งหมด]
14. Router#*show protocols* [แสดงชนิดของโพรโทคอลที่ใช้ทำงานและแสดงสถานะของอินเตอร์เฟสที่มีอยู่ในระบบทั้งหมด] ดังรูปที่ 4.19

```
Router#sh protocols
[redacted]s:
[redacted]rotocol routing is enabled
[redacted]
Ethernet0 is administratively down, line protocol is down
Ethernet1 is administratively down, line protocol is down
```

รูปที่ 4.19 ผลลัพธ์ของคำสั่ง show protocols

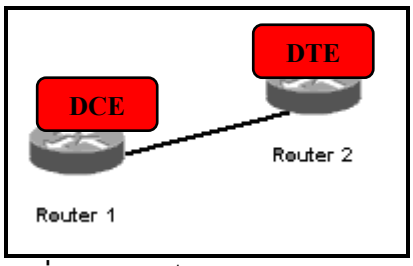
รู้จักกับ CDR (Cisco Discovery Protocol)

หน้าที่หลัก ๆ ของ CDP คือการส่งและรับข้อมูลของคอนฟิกกูเรชันพื้นฐานของเราเตอร์เพื่อนบ้าน เพื่อใช้ในการหาสาเหตุเมื่อเราเตอร์ทำงานผิดปกติ

จุดมุ่งหมาย : เข้าใจการทำงานของ CDR ว่ามีประโยชน์อย่างไร

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัวคือ R1 และ R2

การสร้าง Network Map : ให้สร้างผังเน็ตเวิร์คเหมือน ดังรูปที่ 4.20



รูปที่ 4.20 ผังเน็ตเวิร์คสำหรับ LAB 4

หน้าต่าง Simulator :

<u>บนเราเตอร์ 1</u>

1. Router>*enable*

Router#*conf t*

Router(config)# [เข้าสู่โหมดคอนฟิกกูเรชันของเราเตอร์ 1]

2. **Router(config)#*hostname R1***

R1(config)# [เปลี่ยนชื่อเราเตอร์จาก Router เป็น R1]

3. **R1(config)#*interface serial 0*** [เข้าสู่อินเตอร์เฟส serial 0 ของเราเตอร์ R1]

R1(config-if)#*no shutdown* [สั่งให้อินเตอร์เฟส serial 0 ของเราเตอร์ R1 ทำงาน]

R1(config-if)#*clock rate 5600* [เซ็ต Clock rate เพื่อให้ R1 สร้างสัญญาณนาฬิกาสำหรับใช้สื่อสารกันระหว่างแต่ละอินเตอร์ของเราเตอร์]

หมายเหตุ: ปกติทุก ๆ อินเตอรเฟสจะถูก Disabledไว้ (ไม่ทำงาน)

<u>บนเราเตอร์ 2</u>

1. **Router>*enable***

Router#*conf t*

Router(config)# [เข้าสู่โหมดคอนฟิกกูเรชันของเราเตอร์ 2]

Router(config)#*hostname R2*

R2(config)# [เปลี่ยนชื่อเราเตอร์จาก Router เป็น R2]

2. **R2(config)#*interface serial 0*** [เข้าสู่อินเตอร์เฟส serial 0 ของเราเตอร์ R2]

R2(config-if)#*no shutdown* [สั่งให้อินเตอร์เฟส serial 0 ของเราเตอร์ R2 ทำงาน]

<u>บนเราเตอร์ 1</u>

1. **R1(config)#*interface ethernet 0*** [เข้าสู่อินเตอร์เฟส ethernet 0 ของ R1]

R1(config-if)#*no shutdown* [สั่งให้อินเตอร์เฟส ethernet 0 ของเราเตอร์ R1 ทำงาน]

หมายเหตุ: CDR เป็นโพรโทคอลที่ใช้สำหรับแลกเปลี่ยนคอนฟิกกูเรชันทั่ว ๆ ระหว่างตัวอุปกรณ์ที่เชื่อมต่อกับมันโดยตรง โดยที่ตัว CDR ไม่ได้เกี่ยวข้องกับโพรโทคอลเราติ้ง และ CDR จะทำงานอยู่ที่เลเยอร์ที่ 2

2. **R1(config-if)#*exit***

R1(config)#

R1(config)#*show cdp interface* [แสดงข้อมูลของทุกอินเตอร์เฟสเมื่อสั่งให้ CDP ทำงาน] ดังรูปที่ 4.21

```
R1#sh cdp interface
Serial0 is up, line protocol is up
encapsulation HDLC
Sending CDP packets every 60 seconds
Holdtime is 180 seconds
```

รูปที่ 4.21 ผลลัพธ์ของคำสั่ง show cdp interface

3. R1#*show cdp neighbors* [แสดงรายชื่อของเพื่อนบ้านทั้งหมด ที่กำลังเชื่อมต่ออยู่] ดังรูปที่ 4.22

```
R1#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S -Switch, H - Host, i - IGMP, r - Repeater
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

รูปที่ 4.22 ผลลัพธ์ของคำสั่ง show cdp neighbors

หมายเหตุ: จากผลที่ได้จากคำสั่ง show cdp neighbors ที่เราเตอร์ R1 แสดงให้เห็นว่า R1 รับข้อมูลจาก R2 และส่งข้อมูลไปให้ R2 ด้วย โดยข้อมูลที่รับมีดังนี้

Device ID คือ ชื่อของเราเตอร์ที่เชื่อมต่ออยู่ ในที่นี้คือเราเตอร์ R2

Local Interface คืออินเตอร์เฟสที่เชื่อมต่อกัน ในที่นี้คือ Serial 0

Holdtme คือระยะเวลาที่เราเตอร์ตั้งไว้สำหรับแก้ปัญหาการแลกเปลี่ยนข้อมูลระหว่างกัน ถ้าไม่มีการแลกเปลี่ยนข้อมูลเกินค่า Hold down time R1 จะลบข้อมูลของเพื่อนบ้านออกจากตารางเร้าติ้ง จากรูปมีค่าเท่ากับ 176 วินาที

Capability แสดงว่ากำลังรับข้อมูลจาก R2

Platform แสดงรุ่นของเราเตอร์เพื่อนบ้าน ในที่นี้ R2 คือรุ่น 2501

Port ID เป็นพอร์ทที่เพื่อนบ้านใช้ในการเชื่อมต่อ ในที่นี้ R2 ใช้ serial 0 ติดต่อ

4. R1#*show cdp neighbors detail* [แสดงข้อมูลของเพื่อนบ้านโดยละเอียด] ดังรูปที่ 4.23

```
R1#sh cdp neighbors detail
-------------------------
Device ID: R2
Entry address(es):
Platform: Boson 2501 , Capabilities: Router
Interface: Ser0, Port ID (outgoing port): 1
Holdtime:  174 sec

Version :
Boson Operating System Software
Software, Version 12.1(16), RELEASE SOFTWARE (fc2)
Copyright (c) 1986-2001 by Systems, Inc.
Compiled Fri 02-Mar-01 17:34 by dchih
```

รูปที่ 4.23 ผลลัพธ์ของคำสั่ง show cdp neighbors detail

5. R1#*show cdp* [แสดงระยะเวลาสำหรับทำการอัพเดทข้อมูลระหว่างกันรวมถึงระยะเวลาของ Hold Down Timer ที่ตั้งไว้โดยดีฟอลท์] ดังรูปที่ 4.24

```
R1#sh cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is enabled
```

รูปที่ 4.24 ผลลัพธ์ของคำสั่ง show cdp

6. R1#*conf t*

R1(config)#*cdp timer 45* [เปลี่ยนค่าระยะเวลาของการอัพเดทข้อมูลของ CDP] ดังรูปที่ 4.25

```
R1#sh cdp
Global CDP information:
        [redacted]
          Sending a holdtime value of 180 seconds
          Sending CDPv2 advertisements is enabled
```

รูปที่ 4.25 ผลลัพธ์ของคำสั่ง cdp timer 45

7. **R1#*conf t***

   R1(config)#*cdp holdtime 60* [เปลี่ยนระยะเวลาของ Hold Down Timer] ดังรูปที่ 4.26

```
R1#sh cdp
Global CDP information:
         Sending CDP packets every 45 seconds
        [redacted]
```

รูปที่ 4.26 ผลลัพธ์ของคำสั่ง cdp holdtime 45

หมายเหตุ : เมื่อไม่มีอุปกรณ์อื่นๆ เชื่อมต่ออยู่เลย (มีอุปกรณ์เราเตอร์เพียงตัวเดียว) ไม่ควรทำการ Enable CDP ไว้ เนื่องจากไม่เกิดประโยชน์อะไรและลดภาระของอุปกรณ์ลงด้วย

8. **R1#*conf t***

   R1(config)#*no cdp run* [Disable CDP บนเราเตอร์ R1]

9. **R1#*conf t***

   R1(config)#*cdp run* [Enable CDP บนเราเตอร์ R1]

   หมายเหตุ : คุณสามารถ Disable CDP เฉพาะบางอินเตอร์เฟสก็ได้ ถ้าอินเตอร์เฟสนั้น ๆ มีช่องทางส่งสัญญาณแคบ

10. **R1(config)#*interface ethernet 0***

    R1(config-if)#*no cdp enable* [Disable CDP เฉพาะบางอินเตอร์เฟส ในที่นี้คืออินเตอร์เฟส Ethernet 0 ของเราเตอร์ R1]

    R1#*show cdp interface* [แสดงอินเตอร์เฟสที่ enable CDP ไว้ ส่วน ethernet 0 ไม่แสดงให้เห็นเพราะได้ Disable CDP ที่อินเตอร์เฟสนี้ไปแล้ว]

💻 เรียนรู้การการคอนฟิกเราเตอร์พื้นฐาน

จุดมุ่งหมาย : เรียนรู้การคอนฟิกเราเตอร์ขั้นพื้นฐานเพิ่มเติม

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัว

การสร้าง Network Map : ให้สร้างผังเน็ตเวิร์คเหมือน รูปที่ 4.20

หน้าต่าง Simulator :

1. Router>*enable*

   Router#*conf t*

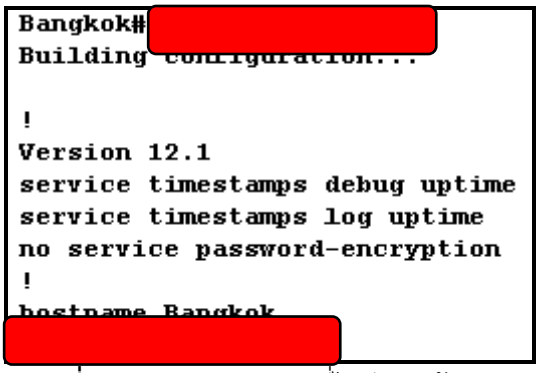
   Router(config)# [เข้าสู่โหมดคอนฟิกกูเรชันของเราเตอร์ 1]

2. Router(config)#*hostname Bangkok* [เปลี่ยนชื่อให้กับเราเตอร์ใหม่]

   Bangkok(config)#

3. Bangkok(config)#*enable password 123* [เซ็ตค่ารหัสผ่านที่จะเข้าสู่โหม privilege เป็น 123 เนื่องจากโหมดนี้มีสามารถเปลี่ยนแปลงค่าของคอนฟิกบางอย่างได้ จึงสมควรที่จะต้องมีการตรวจสอบสิทธิของผู้ที่จะเข้าใช้งาน]

หมายเหตุ : ให้ลอง Logout จากเราเตอร์และลอง Login ใหม่แล้วดูผลการทำงาน เนื่องจากการ enable รหัสผ่านแบบนี้จะไม่มีการเข้ารหัส จึงไม่ปลอดภัยและจำเป็นที่จะต้องใช้คำสั่งสำหรับเข้ารหัส เพื่อความปลอดภัยอีกครั้ง ดังรูปที่ 4.27



รูปที่ 4.27 แสดงรหัสผ่านที่ไม่มีการเข้ารหัส

4. Bangkok(config)#*enable secret 123* [เป็นการเข้ารหัสผ่านเพื่อเข้าโหมด privilege]

หมายเหตุ : ถ้ามีการเซ็ตค่าของรหัสผ่านที่เป็นทั้งแบบไม่เข้ารหัสและแบบที่เข้ารหัสไว้ทั้งคู่ เราเตอร์จะให้ความสำคัญ secret มากกว่า เมื่อ login เข้ามาจะต้องป้อนรหัสของ secret

5. ทดลอง logout และ Login ใหม่อีกครั้ง แล้วดูการเปลี่ยนแปลงที่เกิดขึ้น

การกำหนดแบนเนอร์ให้เราเตอร์ (MOTD Message of the Day)

MOTD เป็นคำสั่งที่ใช้สำหรับแสดงข้อความ เมื่อทำการ login เข้าสู่เราเตอร์ ซึ่งข้อความที่แนะนำให้เขียนควรจะเป็นข้อความที่เป็นคำเตือน เช่น ข้อความที่เกี่ยวกับการละเมิด การบุกรุก การโจมตี ยกตัวอย่าง "การบุกรุก ที่มีจุดประสงค์เพื่อสร้างความเสียหาย เป็นสิ่งที่ผิดกฎหมาย หากละเมิดจะถูกดำเนินการตามกฎหมาย" เป็นต้น อย่าแสดงข้อความที่แสดงถึง รุ่น ยี่ห้อ ของอุปกรณ์เพราะจะเป็นสิ่งที่ผู้ไม่หวังดี นำไปโจมตี หรือบุกรุกได้

จุดมุ่งหมาย : เรียนรู้การสร้างแบนเนอร์ให้เราเตอร์

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัว

การสร้าง Network Map : ให้สร้างผังเน็ตเวิร์คเหมือน รูปที่ 4.20

หน้าต่าง Simulator :

1. Router>

   Router>*enable*

   Router#*config t*

   Router(config)# [เข้าสู่โหมดคอนฟิกกูเรชันของเราเตอร์ 1]

2. Router(config)#*banner motd z* [เป็นการสร้างแบนเนอร์ให้กับเราเตอร์ โดยที่ตัวอักษร z เป็นสัญลักษณ์ที่ใช้สำหรับบอกให้ทราบว่าเป็นตัวปิดหัวและท้ายของข้อความ (delimiting) คือให้เราเตอร์ทราบว่าตัวอักษรที่ต่อจากคำสั่ง motd คือตัวที่แสดงจุดเริ่มต้นของข้อความและมันจะต้องเป็นตัวปิดท้ายของข้อความด้วย] ดังรูปที่ 4.28

หมายเหตุ : ตัวอักษรที่เป็นตัวปิดหัวและท้ายของข้อความจะเป็นตัวอะไรก็ได้ แต่ต้องเหมือนกันทั้ง 2 ตัว

```
Router(config)#[redacted]
Enter the text followed by the 'z' to finish
[redacted]

Router(config)#
```

รูปที่ 4.28 แสดงการสร้างแบนเนอร์ของเราเตอร์

3. *You do not have permission to be here. This router for eats hacker for lunch! z* [ใส่ข้อความเตือน แล้วจบด้วย z สำหรับใช้เป็นตัวปิด]

4. ทดลอง logout และ login ใหม่อีกครั้ง สังเกตข้อความที่แสดงบนเราเตอร์ ดังรูปที่ 4.29

```
You do not have permission to be here. this router for eats hacker lunch!
Router>
Router>
```

รูปที่ 4.29 เราเตอร์แสดงแบนเนอร์เมื่อทำการ login

เรียนรู้คำสั่ง Copy

คำสั่ง Copy มีไว้สำหรับใช้ประโยชน์ในการสำเนาข้อมูลของไฟล์ไปเก็บไว้อีกที่หนึ่ง เช่น การสำเนาข้อมูลจากรันนิ่งคอนฟิกกูเรชันที่อยู่ในหน่วยความจำหลักไปเก็บที่ NVRAM หรือสำเนาข้อมูลจาก NVRAM ไปยัง TFTP เซิร์ฟเวอร์ หรือสำเนาข้อมูลจาก TFTP เซิร์ฟเวอร์มาเก็บไว้ในรันนิ่งคอนฟิก ก็ได้

จุดมุ่งหมาย : เรียนรู้การใช้คำสั่ง Copy

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัว

การสร้าง Network Map : ให้สร้างผังเน็ตเวิร์คเหมือนรูปที่ 4.20

หน้าต่าง Simulator :

1. Router>

Router>*enable*

Router#

2. **Router#*show running-config*** [แสดงรันนิ่งคอนฟิกของเราเตอร์ที่กำลังทำงานอยู่ในหน่วยความจำหลัก รันนิ่งคอนฟิกจะไม่ได้ถูกบันทึกไว้ให้อัตโนมัติ ผู้ดูแลระบบจะต้องบันทึกเอง ถ้าเราเตอร์เริ่มทำงานใหม่รันนิ่งคอนฟิกนั้นจะสูญหายไป แต่มันจะโหลดคอนฟิกจาก NVRAM มาทำงาน เมื่อเราคอนฟิกที่ CLI ด้วยคำสั่งใด ๆ ก็ตามถ้ายังไม่มีการบันทึกลง NVRAM คำสั่งนั้น ๆ อาจจะสูญหายไปเนื่องจาก การรีเซ็ตระบบใหม่ ถ้าต้องการบันทึกคอนฟิกจะต้องใช้คำสั่ง copy]
3. **Router#*show startup-config*** [แสดงคอนฟิกไฟล์ที่อยู่ใน NVRAM]
4. **Router#*copy running-config startup-config*** [บันทึกค่าของรินนิ่งคอนฟิกที่กำลังทำงานอยู่ไปยัง NVRAM เพื่อป้องกันการเสียหายเนื่องจากการรีบู๊ตเราเตอร์]
5. **Router#*show startup-config*** [ทดลองแสดง startup-config ใน NVRAM]
6. **Router#*erase startup-config*** [ลบคอนฟิกกูเรชันไฟล์ของเราเตอร์ทั้งหมด ออกซึ่งจะใช้เมื่อต้องการเริ่มต้นการคอนฟิกเราเตอร์ใหม่ทั้งหมด]
7. **Router#*reload*** [เมื่อลบคอนฟิกกูเรชันไฟล์ทั้งหมดออกแล้วให้สั่ง reload เพื่อเริ่มต้นการทำงานของเราเตอร์อีกครั้ง]
8. **Router#*config terminal***

   Router(config)#*hostname Bangkok*

   Router(config)#*exit*

   Router#*reload* [เปลี่ยนชื่อของเราเตอร์เป็น Bangkok แล้วทดลอง reload เราเตอร์จะถามว่าให้บันทึกคอนฟิกกูเรชันไฟล์หรือไม่ให้ตอบว่า yes]

   Bangkok> [ชื่อของเราเตอร์จะไม่หายไปเนื่องจากมีการบันทึกไว้ที่ NVRAM เรียบร้อยแล้ว]

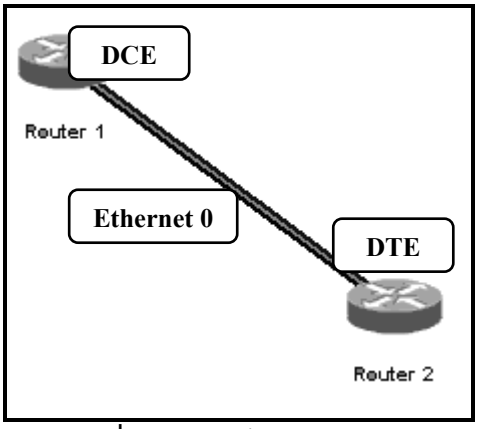
💻 การคอนฟิกอินเตอร์เฟส

เราเตอร์มีอินเตอร์เฟสที่เชื่อมต่อกับอุปกรณ์อื่น ๆ ได้หลายชนิด เช่น token ring, Ethernet, Serial, ISDN เป็นต้น บ่อยครั้งที่เราจำเป็นที่จะต้องตรวจสอบสถานะของแต่ละอินเตอร์เฟส ซึ่งจะมีอยู่หลายคำสั่งที่ใช้แสดงข้อมูลของอินเตอร์เฟส ในส่วนนี้เราจะมาเรียนรู้ว่าจะสั่งให้แต่อินเตอร์เฟสทำงานได้อย่างไร และจะแสดงรายละเอียดของแต่ละอินเตอร์เฟสได้อย่างไร

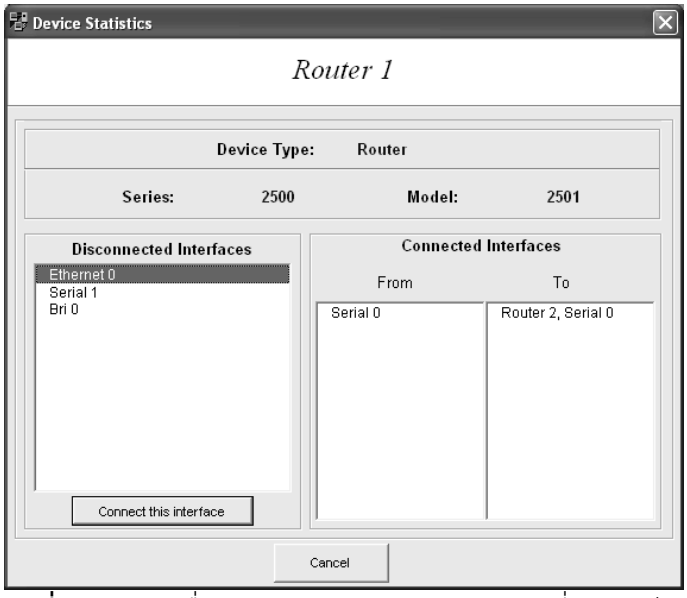
จุดมุ่งหมาย : เรียนรู้การใช้คำให้แต่อินเตอร์เฟสทำงานและหยุดทำงาน

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัวคือ Router1 และ Router2

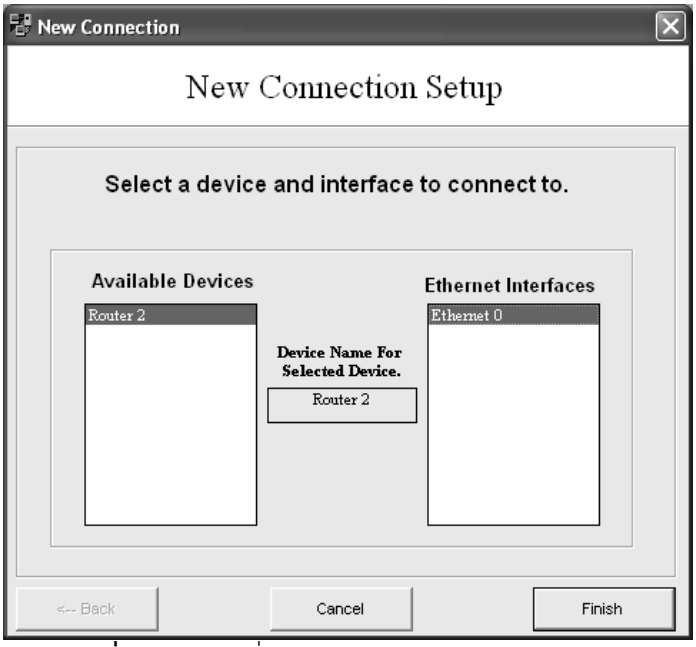
การสร้าง Network Map : ให้สร้างผังเน็ตเวิร์ค ดังรูปที่ 4.30



รูปที่ 4.30 ผังเน็ตเวิร์ค LAB 8



รูปที่ 4.31 การเชื่อมต่อ Ethernet to Ethernet ที่เราเตอร์ 1



รูปที่ 4.32 การเชื่อมต่อ Ethernet to Ethernet 2

หน้าต่าง Simulator :

1. Router>

   Router>*enable*

   Router#*config t*

   Router(config)#*hostname Router1* [เปลี่ยนชื่อเราเตอร์เป็น Router1]

2. Router1(config)#*interface ethernet 0* [เข้าสู่โหมดคอนฟิกอินเตอร์เฟส ethernet 0]

   Router1(config-if)#*?* [แสดงคำสั่งทั้งหมดในโหมดอินเตอร์เฟส] ดังรูปที่ 4.33

```
Router1(config-if)#
exit                    Exit from interface configuration mode
shutdown                Shutdown the selected interface
end                     Exit Configuration Mode
cdp                     CDP interface subcommands
ip                      Interface Internet Protocol config commands
description             Interface specific description
interface               Select an interface to configure
```

รูปที่ 4.33 แสดงคำสั่ง ? ในโหมดคอนฟิกอินเตอร์เฟส

3. Router1(config-if)#*no shutdown* [ซีสโก้เราเตอร์จะใช้คำสั่ง no เพื่อยกเลิกคำสั่งที่อยู่ในรันนิ่งคอนฟิกกูเรชัน ในกรณีนี้ ethernet 0 จะไม่ทำงานโดยดีฟอลท์แต่ถ้า ใช้คำสั่งข้างต้นอินเตอร์เฟสนี้จะทำงานทันที] ดังรูปที่ 4.34

```
Router1(config-if)#no shutdown

Router1(config-if)#
```

รูปที่ 4.34 แสดงสถานะของอีเทอร์เน็ต 0 หลังจากใช้คำสั่ง no shutdown

4. Router1(config-if)#*description Ethernet Interface on Router 1* [ใส่คำอธิบายไปยังอินเตอร์เฟสอีเทอร์เน็ต 0 เพื่อใช้อธิบายว่าอินเตอร์เฟสขานี้ทำหน้าที่อะไร]

5. Router1(config-if)#*end*

   Router1#*show interface* [แสดงรายละเอียดของแต่ละอินเตอร์เฟส ให้สังเกตที่อินเตอร์เฟสอีเทอร์เน็ต 0 ว่ามีการเปลี่ยนแปลงอย่างไรบ้าง]

6. ให้เชื่อมต่อเข้าไปยังเราเตอร์ตัวที่สอง

   Router>

   Router>*enable*

   Router#*config term*

   Router(config)#*hostname Router2* [กำหนดชื่อเราเตอร์เป็น Router2]

   Router2(config)#*interface ethernet 0* [เข้าสู่โหมดคอนฟิกอินเตอร์เฟสอีเทอร์เน็ต 0]

   Router2(config-if)#

7. **Router2(config-if)#*no shutdown*** [สั่งให้อินเตอร์เฟสอีเทอร์เน็ต 0 ทำงาน]
8. **Router2(config-if)#*end***

   Router2#*show cdp neighbors* [แสดงรายละเอียดเพื่อนบ้านที่เชื่อมต่ออยู่ด้วย]

ดังรูปที่ 4.35

```
Router2#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S -Switch, H - Host, i - IGMP, r - Repeater
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

รูปที่ 4.35 แสดงรายละเอียดของเพื่อนบ้านโดยใช้ sh cdp neighbors

ตารางที่ 4.1 สถานะของอินเตอร์เฟส

Interface eth 0 is	Line protocol is	**ความหมาย**
Administratively Down	Down	สั่งด้วยคำสั่ง shutdown โดยผู้ดูแลระบบ
Up	Down	สายสัญญาณเชื่อมต่ออยู่แต่ ข้อมูลที่เรียกว่า keep alive ไม่มี
Down	Down	สายสัญญาณมีปัญหาหรือ DCE ไม่ได้จ่ายสัญญาณนาฬิกา หรือเราเตอร์ตัวอื่น ๆ ไม่ทำงาน
Up	Up	ทำงานได้เป็นปกติ (สิ่งที่เราต้องการ)

แบบฝึกหัดท้ายบท

1. เมื่อเข้าสู่การทำงานของโหมด user ถ้าต้องการจะแสดงคำสั่งทั้งหมดที่ใช้งานในโหมดนี้ คุณจะใช้คำสั่งอะไร?_____
2. เมื่อคุณต้องการเข้าสู่โหมด privilege จะต้องใช้คำสั่งอะไร?_____
3. เมื่อเข้าสู่การทำงานของโหมด privilege ถ้าต้องการจะแสดงคำสั่งทั้งหมดที่ใช้งานในโหมดนี้ คุณจะใช้คำสั่งอะไร?_____
4. ทำอย่างไรถ้าคุณต้องการแสดงคำสั่งที่ใช้งานร่วมกับคำสั่ง show (ใช้คำสั่งอะไร)?_____
5. ต้องการแสดงไฟล์คอนฟิกกูเรชันที่ทำงานในปัจจุบันจะต้องใช้คำสั่งอะไร?_____
6. เมื่อต้องการออกไปสู่โหมด User ควรจะใช้คำสั่งอะไร?_____
1. เมื่อคุณเข้าสู่เราเตอร์แล้วจะใช้คำสั่งอะไรเพื่อจะเข้าสู่ prompt ที่มีรูปแบบเป็น Router# ?__
2. ต้องการแสดงรันนิ่งคอนฟิกจะใช้คำสั่งอะไร?_____
3. ต้องการแสดงข้อมูลของหน่วยความจำแบบแฟรสจะใช้คำสั่งอะไร?_____
   - จงแสดงชื่อของ IOS?_____

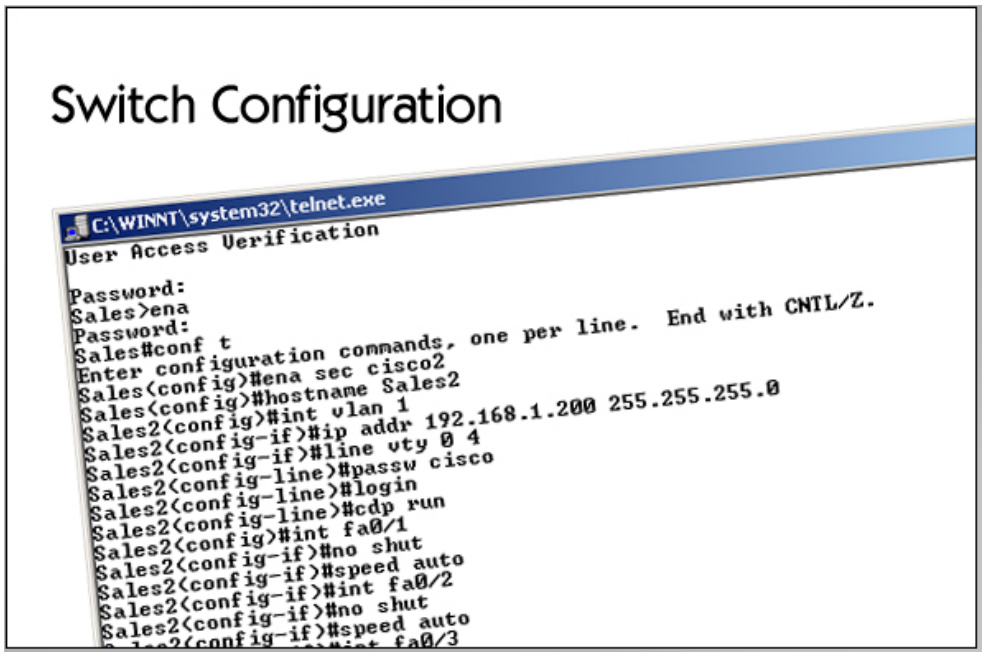
- จงแสดงขนาดของ IOS?_____
- จงแสดงปริมาณหน่วยความจำแฟรสที่ยังไม่ได้ใช้งาน?_____

4. ต้องการแสดงโพรโทคอลค้นหาเส้นทางที่กำลังทำงานอยู่ใช้คำสั่งอะไร?_____
    - โพรโทคอลอะไรที่กำลังทำงาน?_____
    - มีจำนวนอินเตอร์เฟสเท่าไหร่ที่กำลังทำงาน (up) และมีกี่อินเตอร์เฟสที่ผู้ดูแลระบบไม่สั่งให้ทำงาน (Administratively down)?_____
5. ใช้คำสั่งอะไรที่จะแสดงคำสั่งที่ได้ใช้งานผ่านมาแล้ว?_____
6. กดปุ่มคีย์บอร์ดอะไรที่แสดงคำสั่งก่อนหน้า?_____และ_____
7. คำสั่งอะไรที่แสดง เราเตอร์แพลทฟอร์ม, รุ่นของ IOS, เวลาที่บู๊ตระบบครั้งล่าสุด, ตำแหน่งที่อยู่ของไฟล์ ISO, หน่วยควมาจำ, จำนวนของอินเตอร์เฟสที่มี, รีจีสเตอร์คอนฟิกกูเรชัน?_____
    - IOS เก็บอยู่ที่ไหน ?_____
    - แพลทฟอร์มของเราเตอร์คืออะไร?_____
    - จำนวนความจุของ NVRAM?_____
    - ค่าของรีจีสเตอร์คอนฟิกกูเรชัน?_____
    - จำนวนของอินเตอร์เฟสแบบอีเทอร์เน็ตและแบบซีเรียลมีจำนวนเท่าไหร่?_____
8. คำสั่งอะไรที่แสดงวันและเวลาของเราเตอร์?_____
9. เราเตอร์ใช้เวลาไปทำอะไร?_____
10. คำสั่งอะไรที่ใช้แสดงโฮสต์ทั้งหมดบนเราเตอร์?_____
11. คำสั่งอะไรใช้แสดงผู้ใช้ทั้งหมดที่กำลังทำงานบนเราเตอร์?_____
12. คำสั่งอะไรแสดงค่าของโกลบอลและสถานะของอินเตอร์เฟส?_____
13. เมื่อคุณทำงานอยู่บนเราเตอร์แล้วต้องการแสดงคำสั่งทั้งหมดของเราเตอร์ที่มีอยู่ต้องใช้คำสั่งอะไร?_____
14. ถ้าต้องการเข้าสู่การทำงานในโหมด privilege จะต้องใช้คำสั่งอะไร?_____
15. เมื่อเข้าสู่โหมด privilege แล้วต้องการดูคำสั่งทั้งหมดที่ใช้งานได้ จะต้องใช้คำสั่งอะไร?_____
16. จะคำสั่งอะไร เมื่อต้องการเข้าสู่โหมดคอนฟิก?_____
17. ถ้าต้องการเปลี่ยนชื่อของเราเตอร์เป็น Thailand จะต้องใช้คำสั่งอะไร?_____
18. ถ้าต้องการกำหนดรหัสผ่านเป็น abc ในโหมด privilege จะใช้คำสั่งอะไร?_____
19. เมื่อเปลี่ยนรหัสผ่านแล้ว ให้ทดสอบ logout และ login ใหม่อีกครั้ง
20. ถ้าต้องการกำหนดรหัสผ่านแบบเข้ารหัสเป็น 123 ในโหมด privilege จะใช้คำสั่งอะไร?_____

21. เมื่อเปลี่ยนรหัสผ่านแล้ว ให้ทดสอบ logout และ login ใหม่อีกครั้ง ให้ตรวจสอบว่าเราเตอร์ใช้รหัสผ่านตัวไหนเข้าสู่การทำงานของโหมด privilege
22. ให้ล็อกอินเข้าไปยังเราเตอร์และเข้าสู่โหมด privilege.
    1. จงแสดงรันนิ่งคอนฟิก?_____
    2. จงแสดงคอนฟิกกูเรชันไฟล์ที่อยู่ใน NVRAM?_____
    3. ให้ก็อบปี้ running-config ไปยัง NVRAM?_____
    4. จงแสดงคอนฟิกกูเรชันไฟล์ที่อยู่ใน NVRAM อีกครั้ง?_____
    5. ลบคอนฟิกกูเรชันไฟล์ที่อยู่ใน NVRAM?_____
    6. reload เราเตอร์และไม่ต้องบันทึกข้อมูล?_____
    7. แสดงคอนฟิกกูเรชันไฟล์ที่อยู่ใน NVRAM อีกครั้ง?_____
    8. เปลี่ยนชื่อของเราเตอร์เป็น bangkok?_____
23. reload เราเตอร์และให้บันทึกผลการทดลอง?_____
24. จงแสดงคำสั่งที่เมื่อต้องการเข้าคอนฟิกอินเตอร์เฟสอีเทอร์เน็ต 0 ?_____
    1. ปกติทุก ๆ อินเตอร์เฟสจะมีสถานการณ์ทำงานเป็นอย่างไร?_____
    2. คำสั่งอะไรที่ใช้สั่งให้อินเตอร์เฟสทำงาน?_____
    3. คำสั่งอะไรที่สั่งให้อินเตอร์เฟสหยุดทำงาน?_____
    4. แสดงคำสั่งที่แสดงถึงเพื่อนบ้านที่เชื่อมต่ออยู่?_____

บทที่ 5

การคอนฟิกอุปกรณ์สวิชต์ (Switch Configuration)



- Switch basics
- Virtual LANs (VLANs)
- Trunking
- Spanning tree
- Advanced switching

แนวคิด

ในส่วนนี้จะใช้สำหรับฝึกหัดการคอนฟิกอุปกรณ์สวิชต์บนระบบเครือข่ายผ่านตัวจำลอง เนื้อหาในแต่ละแล็บนั้นจะมีลักษณะที่ต่อเนื่องและเรียงลำดับตามความสำคัญของเนื้อหาในการติดตั้งและดูแลระบบเครือข่าย

วัตถุประสงค์

1. เพื่อให้สามารถปรับแต่งอุปกรณ์สวิชต์ที่ใช้งานบนระบบเครือข่ายได้อย่างเหมาะสม เพื่อสร้างความชำนาญในการใช้งานอุปกรณ์ที่สำคัญๆ บนระบบเครือข่าย

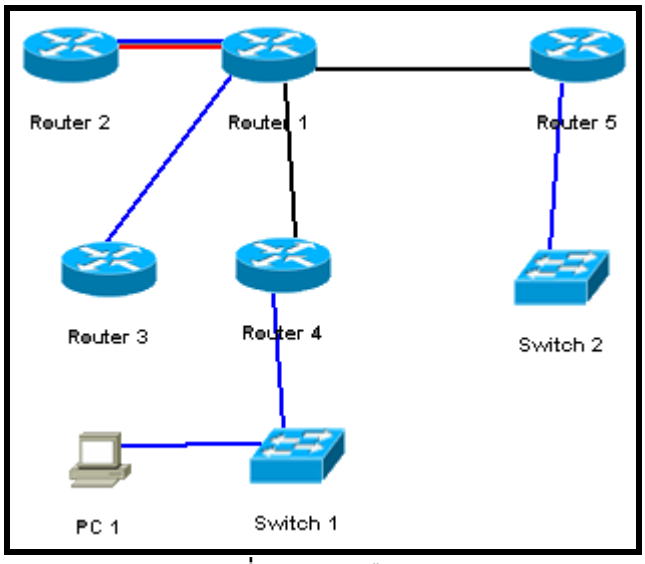
ความรู้เบื้องต้นเกี่ยวกับอุปกรณ์สวิตช์

สวิตช์ทำงานที่เลเยอร์ 2 ของ OSI โมเดล (Data Link Layer) โดยหน้าที่หลักคือการรวมเครื่องผู้ใช้งานเข้ามาในเน็ตเวิร์ค และเชื่อมต่ออุปกรณ์อื่นเข้ามาสู่ระบบ เช่น เซิร์ฟเวอร์ ฮับ และสวิตช์ ส่วนประกอบของสวิตช์ก็คล้าย ๆ กับ PC มันก็จะประกอบไปด้วย CPU, RAM และระบบปฏิบัติการ (IOS) การดูแลและจัดการกับสวิตช์ก็ทำได้เหมือนกับเราเตอร์คือ คอนฟิกผ่านพอร์ทคอนโซล ผ่านการเทลเน็ต และสามารถปรับเปลี่ยน IOS ได้เช่นกัน สวิตช์จะใช้คำสั่งคล้าย ๆ กับเราเตอร์ เช่น ต้องการแสดงรายละเอียดของอินเตอร์เฟสจะใช้คำสั่ง show interface ต้องการหาข้อมูลเกี่ยวกับ IOS ให้ใช้คำสั่ง show version หรือเมื่อต้องการแสดงคอนฟิกกูเรชันที่กำลังทำงานอยู่จะใช้คำสั่ง show running-config และมีบางคำสั่งที่ใช้ไม่เหมือนกันเช่น ต้องการทราบข้อมูลของ MAC Address จะใช้คำสั่ง show mac-address-table เป็นต้น

จุดมุ่งหมาย : เรียนรู้วิธีการคอนฟิกสวิตช์เบื้องต้น

เครื่องมือที่ใช้ทดลอง : ใช้สวิตช์ 1 ตัวคือ switch 1

การสร้าง Network Map : ใช้ผังเน็ตเวิร์คตามรูปที่ 5.1



รูปที่ 5.1 ผังเครือข่าย

หน้าต่าง Simulator :

1. บน Switch 1 เมื่อ Enter จะปรากฏ > แสดงว่าพร้อมรับคำสั่ง

   >

2. แสดงเวอร์ชันของ IOS บนสวิตช์ ดังรูปที่ 5.2

   >show version

```
>sh version
Boson Operating Switch Simulator (BOSS) 1900/2820 Enterprise Edition
Version V5.0
Copyright (c) Boson Software, Inc.  1998-2003
 uptime is 0 days, 0 hours, 55 minutes
Switch 1912 (BOSS) processor with 2048K/1024K bytes of simulated memory
Emulator revision is 5
Upgrade Status: No upgrade currently in progress.
Config File Status: No configuration upload/download is in progress
14 Fixed Ethernet/IEEE 802.3 interface(s)
Base Ethernet Address: 00-0C-69-01-96-31
```

รูปที่ 5.2 show version

- IOS เวอร์ชันอะไร?
- หมายเลข Model ของสวิตช์คือ ?
- หมายเลข Base Internet Address คือ ?

3. แสดงอินเตอร์เฟสของสวิตช์

   >show interface

   - มีจำนวนของอินเตอร์เฟสที่มีความเร็ว 10 Mbps ?
   - มีจำนวนของอินเตอร์เฟสที่มีความเร็ว 100 Mbps ?

4. แสดงข้อมูลของตาราง MAC Address ดังรูปที่ 5.3

   >show mac-address-table

```
>sh mac-address-table
          Mac Address Table
-------------------------------------------
Vlan    Mac Address       Type        Ports
----    -----------       ----        -----
   1    000C.4425.5252    DYNAMIC     Ethernet0/2

Total Mac Addresses for this criterion: 1
```

รูปที่ 5.3 show mac-address-table

- มีจำนวนของ MAC Address ในตาราง ?

5. แสดงคอนฟิกกูเรชันที่กำลังทำงานอยู่

   >show running-config

คำสั่งเบื้องต้นบนอุปกรณ์สวิตช์

จุดมุ่งหมาย : เรียนรู้วิธีการคอนฟิกสวิตช์เบื้องต้นกับสวิตช์รุ่น 1912

เครื่องมือที่ใช้ทดลอง : ใช้สวิตช์ 1 ตัวคือ switch 1

การสร้าง Network Map : ใช้ผังเน็ตเวิร์คตามรูปที่ 5.1

หน้าต่าง Simulator :

1. บน Switch 1 ให้กด Enter เพื่อเข้าสู่โหมด User

   >

2. แสดงคำสั่งทั้งหมดที่สามารถใช้ได้ในโหมดผู้ใช้ (user) โดยใช้คำสั่ง ?

>?

3. โหมด privilege จะมีความสามารถควบคุมการทำงานของสวิตช์ได้ทั้งหมด

>enable [เข้าสู่โหมด privilege]

#

4. แสดงคำสั่งที่สามารถใช้งานได้ทั้งหมดในโหมด privilege

#?

5. เมื่อต้องการคอนฟิกสวิตช์จะใช้คำสั่ง config terminal เหมือนกับเราเตอร์

#config term

(config)#

6. กำหนดชื่อของ สวิตช์ ให้เป็น Boson

(config)#*hostname Boson*

Boson(config)#

7. ในสวิตช์ตระกูล 1900 สามารถกำหนดรหัสผ่านที่เข้าสู่โหมด privilege ได้หลายระดับ

Boson(config)#*enable password level 15 boson* [กำหนดรหัสผ่านเป็น boson ระดับที่ 15]

8. เมื่อกำหนดรหัสผ่านแล้วต้องการทดสอบ ให้ออกไปสู่โหมด User แล้วล็อกอินเข้ามาใหม่อีกครั้ง

Boson(config)#*exit*

Boson#*exit*

Boson>

Boson>*enable*

password:

Boson#

9. รหัสผ่านที่กำหนดในขั้นตอนที่ 8 จะเป็นแบบ Plain text ซึ่งจำไม่มีการเข้ารหัสทำให้ไม่ปลอดภัย ตัวสวิตช์จึงเตรียมคำสั่งเพื่อให้ผู้ใช้งานสามารถเข้ารหัสรหัสผ่านได้

Boson#*config terminal*

Boson(config)#*enable secret level 15 cisco* [กำหนดรหัสผ่านเป็น cisco และเข้ารหัสข้อมูลด้วย ถ้ามีทั้งรหัสผ่านแบบ plain text และ Secret สวิตช์จะมองว่า Secret สำคัญมากกว่า] ดังรูปที่ 5.4

```
!
enable secret 5 89E$$3634D$sdf0923SD4837
enable password level 15 "boson"
!
```

รูปที่ 5.4 แสดงรหัสผ่านที่เป็นแบบ plain text และแบบ secret

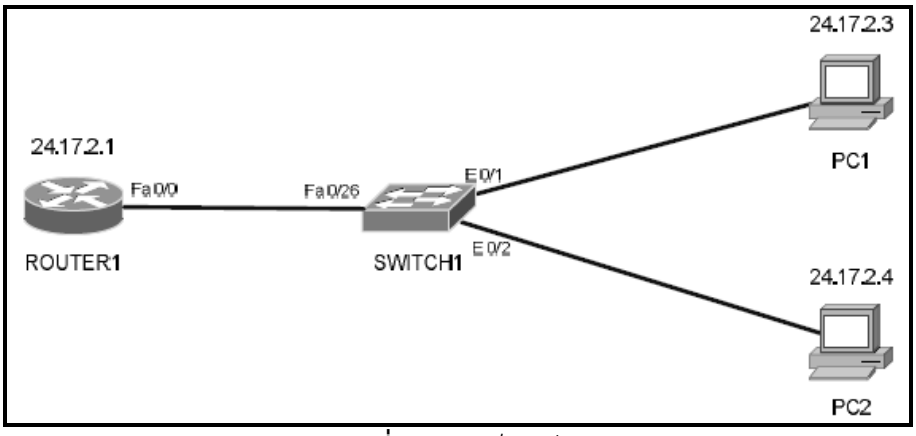
ออกไปสู่ User โหมดแล้วทดสอบใส่รหัสผ่านที่ตั้งไว้ สังเกตว่ารหัสผ่านชนิดไหนที่สามารถเข้าสู่โหมด privilege ได้

💻 เบื้องต้นกับ VLAN

จุดมุ่งหมาย : เรียนรู้การคอนฟิก **VLAN**

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 1 ตัว (Router1 2516) สวิตช์ 1 ตัว (1900 series) และ PC 2 เครื่อง

การสร้าง Network Map : ดังรูปที่ 5.5



รูปที่ 5.5 ผังเน็ตเวิร์ค

Simulator :

1. บนเราเตอร์ 1 ให้เปลี่ยนชื่อเป็น Router1 และกำหนดไอพีแอดเดรสเป็น 24.17.2.1 255.255.255.0 บนอินเตอร์เฟสอีเทอร์เน็ต 0

   Router>*enable*

   Router#*config terminal*

   Router(config)#*hostname Router1*

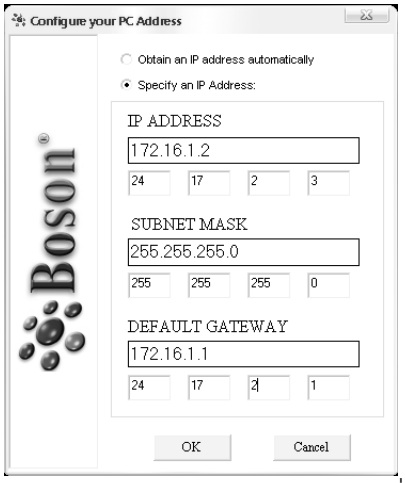
   Router1(config)#

   Router1(config)#*interface ethernet 0*

   Router1(config-if)#*ip address 24.17.2.1 255.255.255.0*

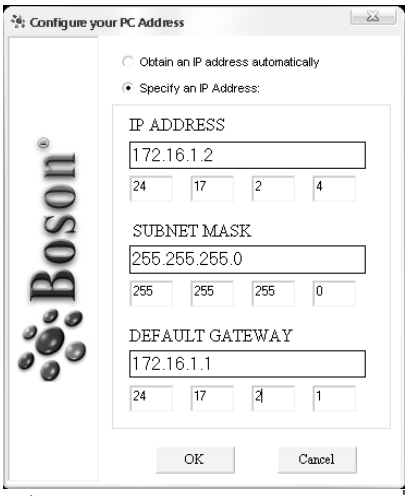
   Router1(config-if)#*no shutdown*

2. บน PC1 ให้กำหนดไอพีแอดเดรสเป็น 24.17.2.3 255.255.255.0 gateway เป็น 24.17.2.1 โดยใช้คำสั่ง winipcfg ดังรูปที่ 5.6



รูปที่ 5.6 ระบุไอพีแอดเดรสให้กับ PC1 ด้วยคำสั่ง winipcfg

3. บน PC2 ให้กำหนดไอพีแอดเดรสเป็น 24.17.2.4 255.255.255.0 gateway เป็น 24.17.2.1 ดังรูปที่ 5.7



รูปที่ 5.7 ระบุไอพีแอดเดรสให้กับ PC2 ด้วยคำสั่ง winipcfg

4. ถึงขั้นตอนนี้คุณควรจะสามารถ ping จาก PC2 ไปยัง Router1 และ PC1 ได้แล้ว ดังรูปที่ 5.8, 5.9

```
C:>ping 24.17.2.1
Pinging 24.17.2.1 with 32 bytes of data:

Reply from 24.17.2.1: bytes=32 time=60ms TTL=241
Reply from 24.17.2.1: bytes=32 time=60ms TTL=241
Reply from 24.17.2.1: bytes=32 time=60ms TTL=241
Reply from 24.17.2.1: bytes=32 time=60ms TTL=241
Reply from 24.17.2.1: bytes=32 time=60ms TTL=241
```

รูปที่ 5.8 ping จาก PC2 ไปยัง Router1

```
C:>ping 24.17.2.3
Pinging 24.17.2.3 with 32 bytes of data:

Reply from 24.17.2.3: bytes=32 time=60ms TTL=241
Reply from 24.17.2.3: bytes=32 time=60ms TTL=241
Reply from 24.17.2.3: bytes=32 time=60ms TTL=241
Reply from 24.17.2.3: bytes=32 time=60ms TTL=241
Reply from 24.17.2.3: bytes=32 time=60ms TTL=241
```

รูปที่ 5.9 ping จาก PC2 ไปยัง PC1

5. ขั้นตอนนี้จะเป็นการคอนฟิก VLAN ปกติทุก ๆ อินเตอร์เฟสจะอยู่ที่ VLAN 1 อัตโนมัติ เราจะเริ่มแยก VLAN ออกมาจาก VLAN 1 ซึ่งจะเป็น VLAN ที่เราสร้างขึ้นมาเอง
   บนสวิตช์ 1 ให้สร้าง VLAN หมายเลข 22 มีชื่อเป็น pcs
   >en
   #config terminal
   (config)*#vlan 22 name pcs*
6. เมื่อสร้าง vlan แล้ว ต่อไปจะต้องกำหนดพอร์ตที่จะใช้งานกับ vlan นี้ ในที่นี้กำหนดให้พอร์ต e0/1 เป็นสมาชิกของ vlan นี้
   (config)*#interface e0/1*
   (config-if)*#vlan-membership static 22* [ระบุให้พอร์ตหรืออินเตอร์เฟส e0/1 เป็นสมาชิกของ vlan 22 แบบถาวร]
7. ถึงจุดนี้เราได้กำหนด vlan ให้กับ PC1 เป็นสมาชิกของ vlan 22 แล้วให้ทดสอบ ping จาก PC2 ไปยัง Router1 และ PC1 เหมือนในขั้นตอนที่ 4 อีกครั้ง
   บน PC2
   C:>*ping 24.17.2.1*
   C:>*ping 24.17.2.3*

ผลที่ได้เราไม่สามารถจะ ping PC1 ได้เนื่องจาก เราได้สร้าง vlan 22 และกำหนดให้ PC1 เป็นสมาชิกของ vlan นี้แล้วนั่นหมายความว่า PC1 ไม่ได้อยู่ที่ vlan 1 ซึ่งเป็นดีฟอลท์ vlan อีกแล้ว แต่เมื่อเรา ping ไปที่ Router1 จะสามารถ ping ได้เนื่องจากทั้ง PC2 และพอร์ตที่เชื่อมต่อไปยัง Router1 ยังคงเป็นสมาชิกของ vlan 1 เหมือนกัน

8. กลับไปที่สวิตช์แล้วทำการคอนฟิกให้พอร์ต e0/2 (ซึ่ง PC2 เชื่อมต่ออยู่) ให้เป็นสมาชิกของ vlan 22
   (config-if)*#exit*
   (config)*#interface e0/2*
   (config-if)*#vlan-membership static 22*
9. กลับไปที่ PC2 อีกครั้งแล้วลองทดสอบ ping เหมือนเดิมอีกครั้ง
   บน PC2
   C:>*ping 24.17.2.1*
   C:>*ping 24.17.2.3*

ผลปรากฏว่าคราวนี้ไม่สามารถ ping Router1 ได้แล้วเพราะอยู่คนละ vlan แต่สามารถ ping PC1 ได้เนื่องจากเป็นสมาชิกของ vlan หมายเลข 22 เหมือนกัน

10. กลับไปที่สวิตช์อีกครั้ง ลองแสดง vlan ที่ได้คอนฟิกไปแล้วด้วนคำสั่ง show vlan และ show vlan-membership

(config-if)#*end*

#show vlan

#show vlan-membership ดังรูปที่ 5.10

```
#show vlan-membership
   Port  VLAN    Membership Type      Port  VLAN    Membership Type
   ------------------------------     ------------------------------
   [redacted]
   3        1        Static
   4        1        Static
   5        1        Static
   6        1        Static
```

รูปที่ 5.10 แสดงคำสั่ง show vlan-membership

11. ขั้นตอนสุดท้ายให้ทำการคอนฟิกให้พอร์ต FA0/26 ซึ่งเป็นพอร์ตที่เชื่อมต่อไปยัง Router1 เป็นสมาชิกของ vlan 22

    บนสวิตช์

    #config t

    (config)*#interface FastEthernet 0/26*

    (config-if)*#vlan-membership static 22*

12. ทดสอบ ping ไปยัง PC1, 2 ให้ครบ

Virtual Truking Protocol (VTP)

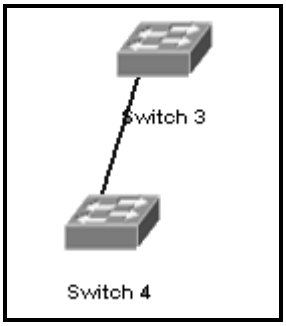
การสร้าง VTP เปรียบเสมือนการสร้างท่อขนส่งข้อมูลแบบจำลอง ๆ ขึ้นมา ซึ่งท่อแต่ละท่อจะมีข้อมูลที่สอดคล้องหรือเป็นกลุ่มเดียวกันหรือมาจากแหล่งที่ใกล้เคียงกัน ถ้าจะยกตัวอย่างง่าย ๆ ของ VTP ก็คือ เราเห็นสายโทรศัพท์ที่เดินมาตามเสาไฟฟ้า (เปรียบเสมือน vlan) แล้วเราก็ทำการมัดรวมเอาสายโทรศัพท์เหล่านั้นเข้าไว้ด้วยกันเป็นมัดใหญ่ ๆ มัดหนึ่ง (VTP) ซึ่งก็เทียบเคียงได้เหมือนกับ Virtual Trunking เหมือนกัน

จุดมุ่งหมาย : เรียนรู้การคอนฟิก **VTP**

เครื่องมือที่ใช้ทดลอง : ใช้สวิตช์ 2 ตัว (Catalyst 2950)

- คอนฟิก vlan บน Cisco Catalyst 2950
- ระบุพอร์ตให้กับ vlan
- คอนฟิก VTP ในโหมดของ server และ client
- คอนฟิก VTP ระหว่างสวิตช์
- ทดสอบการคอนฟิก VTP

การสร้าง Network Map : การเชื่อมต่อกันระหว่าง Switch3 และ Switch4 ผ่านพอร์ต fastEthernet 0/12 (Trank Port) ดังรูปที่ 5.11 และตารางที่ 5.1



รูปที่ 5.11 ผังเน็ตเวิร์คสำหรับ LAB VTP

ตารางที่ 5.1 รายละเอียดของอุปกรณ์ switch

อุปกรณ์	Switch 3	Switch 4
ชื่ออุปกรณ์	Switch3	Switch4
ไอพีแอดเดรส (VLAN 1)	10.1.1.1	10.1.1.2
Subnet mask	255.255.255.0	255.255.255.0

Simulator :

1. เริ่มต้นด้วยการกำหนดหมายเลขไอพีแอดเดรสให้กับสวิตช์ เปลี่ยนชื่อและสั่งให้ทำงานบนสวิตช์หมายเลข 3

   Switch>enable

   Switch#config t

   Switch(config)#*hostname Switch3*

   Switch3(config)#*interface vlan 1*

   Switch3(config-if)#*ip address 10.1.1.1 255.255.255.0*

   Switch3(config-if)#*no shutdown*

   Switch3(config-if)#*end*

   Switch3#

2. ทดสอบการเชื่อมต่อของสวิตช์โดยทดสอบ ping จาก สวิตช์ 4 มายัง สวิตช์ 3 (ไม่สามารถ ping ได้)

3. ขั้นต่อไป ให้สร้าง vlan 8 และ vlan 14 โดยกำหนดให้พอร์ต 0/2 ถึง 0/5 เป็นสมาชิกของ vlan 8 และ พอร์ต 0/6 ถึง 0/10 เป็นสมาชิกของ vlan 14

   Switch3#*vlan database* [ฐานข้อมูลสำหรับเก็บข้อมูลของ vlan]

   Switch3(vlan)#*vlan 8* [สร้าง vlan หมายเลข 8]

   Switch3(vlan)#*vlan 14*

   Switch3(vlan)#*exit*

   Switch3#*config t*

Switch3(config)#*interface range fast0/2 – 5* [กำหนดช่วงของพอร์ตที่ต้องการ]

Switch3(config-if)#*switchport access vlan 8* [กำหนดช่วงของพอร์ตที่ต้องการเข้า เป็นสมาชิกของ vlan]

Switch3(config)#*exit*

Switch3(config)#*interface range fast0/6 – 10*

Switch3(config-if)#*switchport access vlan 14*

Switch3(config-if)#*exit*

Switch3(config)#

4. ใช้คำสั่ง show vlan เพื่อตรวจสอบความถูกต้องของคอนฟิกกูเรชัน

   Switch3(config)#*exit*

   Switch3#*show vlan* ดังรูปที่ 5.12

```
Switch3#sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/11, Fa0/12
```

รูปที่ 5.12 แสดงคำสั่ง show vlan

5. โดยดีฟอลท์แล้ว Catalyst สวิตช์จะถูกตั้งเป็นโหมด Server เราจะคอนฟิกให้สวิตช์ 3 เป็น server และ สวิตช์ 4 เป็นโหมด client เปลี่ยน VTP Domain เป็น Boson และกำหนดรหัสผ่านเป็น rules

   Switch3#*vlan database*

   Switch3(vlan)#*vtp server* [กำหนดโหมดการทำงานของสวิตช์เป็น server]

   Switch3(vlan)#*vtp domain Boson* [กำหนดชื่อของโดเมน]

   Switch3(vlan)#*vtp password rules* [กำหนดรหัสผ่าน]

6. เข้าไปยัง switch 4 คอนฟิก VTP

   Switch4#*vlan database*

   Switch4(vlan)#*vtp client* [กำหนดโหมดการทำงานของสวิตช์เป็น client]

   Switch4(vlan)#*vtp domain Boson* [กำหนดชื่อของโดเมน]

   Switch4(vlan)#*vtp password rules* [กำหนดรหัสผ่าน]

7. โหมดการทำงานของสวิตช์นั้นจะมีทั้งหมด 3 โหมดคือ server จะทำหน้าที่หลัก ๆ คือจะส่งข้อมูลของ vlan ไปให้ยังสมาชิกที่เชื่อมต่อกับมัน (client) และเป็นการสร้าง vlan จากจุด

ศูนย์กลางเพียงที่เดียวเท่านั้น ส่วน client จะไม่มีความสามารถพิเศษอะไรเพียงแต่รับข้อมูลที่ serve ส่งมาให้แล้วเก็บคอนฟิกเหล่านั้นไว้แล้วปรับปรุงข้อมูล vlan ของตัวเอง ส่วน transparent จะไม่สนใจข้อมูลที่ server ส่งให้มันจะสร้าง vlan ขึ้นมาเป็นของตัวมันเอง ขั้นตอนนี้เราจะทำการสร้างท่อเพื่อเชื่อมเอา switch3 และ switch4 เข้าหากันและส่งข้อมูลของ vlan แต่ละฝ่ายให้สามารถคุยกันได้ โดยต้องเรียกวิธีการ encapsulation ในที่นี้ให้เลือกเป็นมาตรฐานของ 802.1Q ใช้พอร์ต fast0/12 เป็น Trank

Switch3(config)#*interface fast0/12*

Switch3(config-if)#*switchport mode trank*

Switch3(config-if)#*end*

Switch4(config)#*interface fast0/12*

Switch4(config-if)#*switchport mode trank*

Switch4(config-if)#*end*

8. เมื่อถึงขั้นตอนนี้แล้วการเชื่อมต่อทั้งหมดจะสามารถทำงานได้แล้วให้ทดสอบการเชื่อมต่อด้วยการใช้คำสั่ง

   show vlan

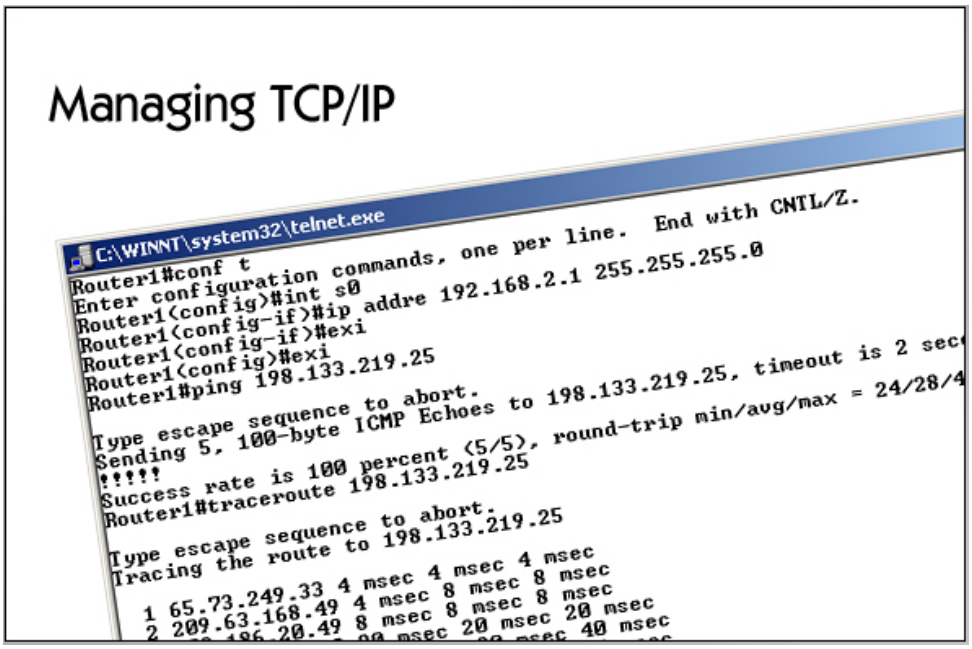
   show vtp status

แบบฝึกหัดท้ายบท

1. เมื่อต้องการดูคำสั่งที่ใช้งานในโหมดของ User จะต้องใช้คำสั่งอะไร?_____
2. ถ้าต้องการเข้าสู่โหมด privilege จำต้องใช้คำสั่งอะไร?_____
3. เมื่อต้องการดูคำสั่งที่ใช้งานในโหมดของ privilege จะต้องใช้คำสั่งอะไร?_____
4. ถ้าต้องการเข้าสู่โหมดคอนฟิกจะต้องใช้คำสั่งอะไร?_____
5. ต้องการเซ็ตชื่อของสวิตช์เป็นBoson จะต้องใช้คำสั่งอะไร?_____
6. ต้องการกำหนดรหัสผ่านแบบ plain text จะต้องใช้คำสั่งอะไร?_____
7. ต้องการทดสอบเมื่อเซ็ตรหัสผ่านแล้วจะต้องอย่างไร?_____
8. ต้องการกำหนดรหัสผ่านแบบ secret จะต้องใช้คำสั่งอะไร?_____
9. ให้ logout ออกจากสวิตช์แล้ว Login เข้าใหม่แล้วเข้าสู่โหมด privilege ต้องทำอย่างไร?__

บทที่ 6

การบริหารจัดการ ไอพีแอดเดรส (Managing TCP/IP)



- Subnetting
- Configuring IP
- Address resolution
- Troubleshooting IP

แนวคิด

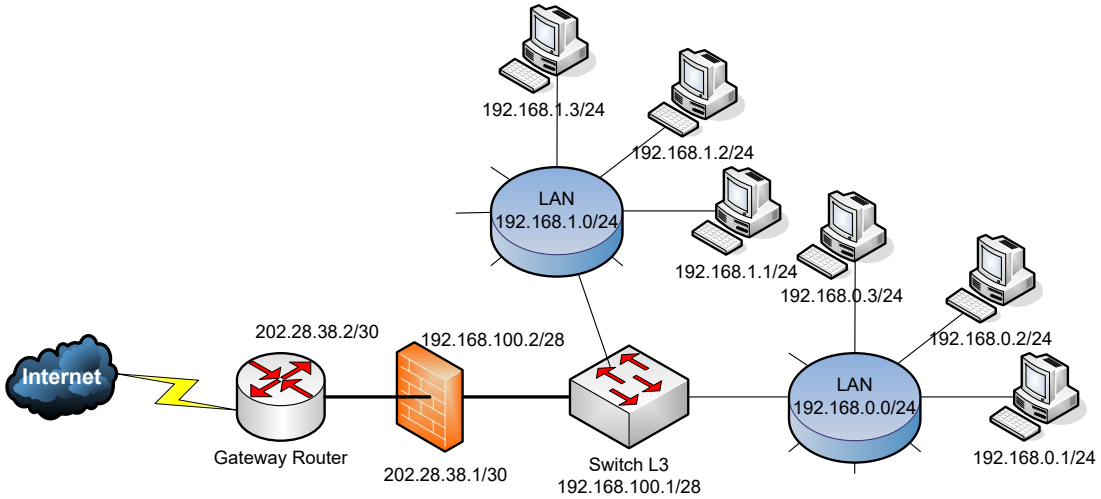
การจัดสรรหมายเลขที่อยู่บนระบบเครือข่ายมีความสำคัญเป็นอย่างมาก อาจกล่าวได้ว่าเป็นหัวใจของการบริหารจัดการระบบเครือข่ายเลยดีเทียว ความเหมาะสมของการจัดสรรไอพี ขึ้นอยู่กับความต้องการใช้งานภายในองค์กร ในส่วนนี้จะอธิบายวิธีการแบ่งหมายเลขที่อยู่ให้เกิดประสิทธิภาพสูงที่สุด

วัตถุประสงค์

1. เพื่อให้สามารถจัดสรรหมายเลขที่อยู่ได้อย่างถูกต้องและมีประสิทธิภาพ
2. เพื่อให้ทราบถึงวิธีการจัดสรรหมายเลขที่อยู่แบบต่างๆ
3. เพื่อให้สามารถเอาวิธีการต่างๆ เหล่านี้ไปประยุกต์ใช้ในงานจริงได้

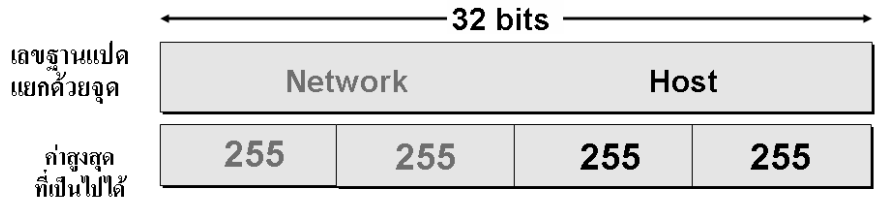
ไอพีเอดเดรสและการออกแบบ

หมายเลขไอพีแอดเดรสเป็นที่อยู่ที่ใช้ในการระบุแยกแยะความแตกต่างของเครื่องคอมพิวเตอร์ และอุปกรณ์เครือข่ายต่าง ๆ ที่มีการเชื่อมต่อเข้าด้วยกันเป็นเครือข่ายคอมพิวเตอร์ออกจากกัน โดยคอมพิวเตอร์แต่ละเครื่องภายในเครือข่าย จะได้รับการจัดสรรหมายเลขไอพีแอดเดรสที่ไม่ซ้ำกัน เพราะจะทำให้เกิดความสับสนในการติดต่อสื่อสารภายในเครือข่าย หากการกำหนดจัดสรรไอพีแอดเดรสเป็นไปตามกติกาดังกล่าวแล้ว ก็จะทำให้คอมพิวเตอร์แต่ละเครื่องสามารถติดต่อสื่อสารถึงกัน เพื่อแลกเปลี่ยนข้อมูล คอมพิวเตอร์ภายในเครือข่ายจะทำการติดต่อสื่อสารกันโดยการรับส่งข้อมูลในรูปแบบของแพ็กเกต (Packet) ผ่านเครือข่ายคอมพิวเตอร์ที่มีการเชื่อมต่อโดยอาศัยอุปกรณ์เครือข่ายชนิดต่าง ๆ เช่น บริดจ์ ฮับ สวิตช์ หรือเราเตอร์ และประเด็นที่สำคัญคือจะต้องมีการเชื่อมต่อเข้าสู่เครือข่ายอินเตอร์เน็ต ซึ่งมีจำนวนเครื่องคอมพิวเตอร์เชื่อมต่ออยู่อย่างมหาศาล การกำหนดเลขหมายไอพีแอดเดรสให้กับเครือข่ายกลายเป็นสิ่งที่ควบคุมได้ยากลำบากมาก หากไม่มีการกำหนดกฎเกณฑ์และกติกาในการจัดสรรหมายเลขไอพีแอดเดรสเพื่อให้แต่ละองค์กรยึดถือร่วมกัน เพื่อเป็นการวางมาตรฐานสำหรับปฏิบัติร่วมกัน หน่วยงาน InterNIC (Internet Network Information Center) ซึ่งเป็นหน่วยงานที่ได้รับการจัดตั้งขึ้นโดยรัฐบาลสหรัฐอเมริกา จึงได้ทำหน้าที่เป็นผู้ออกกฎกติกา สำหรับการจัดสรรหมายเลขไอพีแอดเดรส ให้กับเครื่องคอมพิวเตอร์ทั่วโลกที่จะต้องมีการเชื่อมต่อเข้ากับเครือข่ายอินเตอร์เน็ต โดยมีจุดประสงค์เพื่อป้องกันปัญหาการกำหนดไอพีแอดเดรสซ้ำซ้อนกันขึ้น อย่างไรก็ตามในกรณีของการวางเครือข่ายอินทราเน็ตสำหรับภายในองค์กรโดยไม่คิดว่าจะมีการเชื่อมต่อเข้ากับเครือข่ายอินเตอร์เน็ต ผู้ดูแลระบบสามารถเลือกกำหนดไอพีแอดเดรสให้กับคอมพิวเตอร์ภายในเครือข่ายของท่านอย่างไรก็ได้โดยยึดถือเพียงหลักในการจัดสรรไอพีแอดเดรสมาตรฐาน ดังรูปที่ 6.1 เป็นการเปรียบเทียบการจัดสรรหมายเลขไอพีแอดเดรสที่มีการเชื่อมต่อกับอินเตอร์เน็ต โดยอินทราเน็ตจะใช้หมายเลขไอพีแอดเดรสแบบ private คือ 192.168.0.0/24, 192.168.1.0/24, 192.168.100.0/24 เป็นต้น ส่วนอินเตอร์เน็ตจะใช้ไอพีแอดเดรสที่เป็น public คือ 202.28.32.1/30, 202.28.32.2/30 เป็นต้นดังรูป



รูปที่ 6.1 การจัดสรรไอพีแอดเดรสภายในองค์กร

หมายเลขไอพีแอดเดรสที่มีการใช้งานในเครือข่ายคอมพิวเตอร์ในปัจจุบัน มีชื่อเรียกว่า “**IPv4**” อ้างอิงโดยใช้เป็นตัวเลขฐานสองความยาว 32 บิต ออกเป็น 4 กลุ่ม ๆ ละ 8 บิตเรียงตามลำดับโดยมีจุดขั้นไว้ แต่การอ่านค่าของหมายเลขไอพีแอดเดรสโดยทั่วไป มักอยู่ในรูปแบบของเลขฐานสิบเพื่อความสะดวกเพราะมนุษย์ค้นเคยกับเลขฐานสิบมากกว่าฐานอื่น ๆ ตัวอย่าง เช่น 202.28.33.50 เมื่อแปลงเป็นเลขฐานสองจะได้ 11001010.00011100.00100001.00110010 สำหรับการแปลงค่าตัวเลขฐานสองไปเป็นฐานสิบนั้นสามารถทำได้ง่าย ๆ โดยการเทียบกับรูปที่ 6.2 ตัวอย่างเช่น ต้องการหาค่า 00100011 ว่าเป็นค่าเท่าไหร่ในเลขฐานสิบ ขั้นตอนแรกให้นำเลข



รูปที่ 6.2 รูปแบบแอดเดรสของ IPV4

ฐานสองเขียนลงในตารางให้ตรงกัน เช่น 1 ในตำแหน่งสุดท้ายก็ใส่ให้ตรงกับ 2 ยกกำลัง 0 เมื่อใส่ครบทั้ง 8 ตัวแล้วก็พิจารณาว่าค่าใดที่มีค่าเป็น 1 ให้นำค่าประจำตำแหน่งมาคำนวณ เช่น 1 ในช่องของ 2 ยกกำลัง 5 จะมีค่าเท่ากับ 32 เสร็จแล้วก็นำค่าฐานสิบที่ตรงกับเลข 1 ในเลขฐานมารวมกันทุกค่าก็จะได้ค่าเป็นเลขฐานสิบดังรูปที่ 6.3

ตัวอย่าง : 35 = 32 + 2 + 1 = 00100011

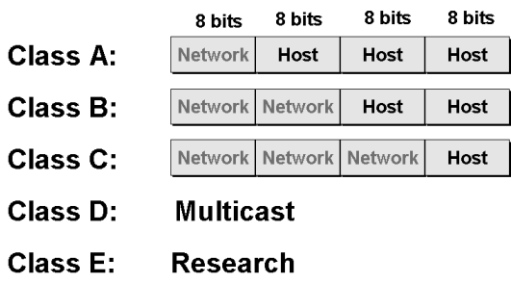
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	0	1	0	0	0	1	1

รูปที่ 6.3 การแปลงเลขฐานสองเป็นเลขฐานสิบ

ผู้ดูแลระบบจะต้องทราบว่า InterNIC มีการกำหนดแบ่งกลุ่มของไอพีแอดเดรสที่มีการใช้งานทั่วโลกออกเป็น 5 กลุ่ม หรือ 5 คลาส (Class) ซึ่งในการกำหนดใช้งานทางปฏิบัติจะมีอยู่เพียง 3 คลาสเท่านั้น คือ คลาส A คลาส B และ คลาส C ดังรูปที่ 6.4 ข้อกำหนดในการแบ่งคลาสของไอพีแอดเดรสนั้นระบุว่า Class A กำหนดให้ 7 บิตแรกทางซ้ายมือเป็นหมายเลขของเน็ตเวิร์คซึ่งจะขึ้นต้นด้วยเลข 0 ส่วนบิตที่เหลืออีก 24 บิตเป็นหมายเลขของโฮสต์ที่สามารถใช้งานได้ ซึ่งก็มีค่าที่เป็นไปได้ดังนี้

เน็ตเวิร์คมีจำนวน 7 บิต ค่าที่เป็นไปได้สูงสุดคือ 2 ยกกำลัง 7 = 128 - 2 = 126 เน็ตเวิร์ค (ลบออก 2 เนื่องจากเป็นหมายเลขเน็ตเวิร์คและบรอดคลาสแอดเดรส)

โฮสต์มีจำนวน 24 บิต ค่าที่เป็นไปได้สูงสุดคือ 2 ยกกำลัง 24 = 16,777,216 ล้านเครื่อง



รูปที่ 6.4 IP Address Classes

แนวคิดในการแบ่งกลุ่มไอพีแอดเดรสออกเป็นคลาส เนื่องมาจากที่มองว่าเครือข่ายคอมพิวเตอร์ย่อย ๆ ที่มีการเชื่อมต่อเข้ากับเครือข่ายอินเตอร์เน็ตนั้น มีขนาดเล็กใหญ่ไม่เท่ากัน เครือข่ายของบางองค์กรอาจมีขนาดใหญ่และต้องการหมายเลขไอพีแอดเดรสเป็นจำนวนมาก ในขณะที่เครือข่ายขององค์กรบางกลุ่มกลับมีเครื่องคอมพิวเตอร์อยู่เพียงไม่กี่เครื่อง จำนวนเครือข่ายคอมพิวเตอร์ที่แต่ละคลาสสามารถรองรับได้แสดงไว้ในรูปที่ 6.4 การคำนวณจำนวนเครือข่ายและจำนวนโฮสที่รองรับในแต่ละคลาสสามารถทำได้ดังนี้

คลาส A

ในแต่ละกลุ่มของ 8 บิตจะเรียกชื่อว่า Octet ซึ่งจะประกอบด้วย 4 Octet ค่าบิตแรกของ Octet ที่ 1 (8 บิตแรก) ในไอพีแอดเดรสมีค่าเป็น 0 ในคลาสนี้มีการแบ่งกลุ่มเครือข่ายออกได้เป็น 126 เครือข่าย(ใช้ 7 บิตลบด้วย 2) โดยที่แต่ละเครือข่ายสามารถมีเครื่องคอมพิวเตอร์ต่อเชื่อมได้ทั้งสิ้นภายในเครือข่ายละ 16,777,214 เครื่อง ปัจจุบันไอพีแอดเดรสที่จัดว่าอยู่ในคลาส A แทบจะไม่ได้มีการจัดสรรให้กับหน่วยงานใดมากนัก แอดเดรสในคลาส A นี้ได้รับการกำหนดขึ้นเพื่อใช้จัดสรรให้กับองค์กรขนาดใหญ่มากที่มีเครื่องคอมพิวเตอร์เชื่อมต่อภายในองค์กรเป็นจำนวนมาก รูปแบบของการจัดวางไอพีแอดเดรสสำหรับคลาส A นี้มีการกำหนดให้ Octet แรกใช้แทนแอดเดรสของเครือข่ายและอีก 3 Octet ที่เหลือใช้แทนแอดเดรสเครื่องคอมพิวเตอร์ภายในเครือข่ายนั้น ๆ ดังรูปที่ 6.5, 6.6

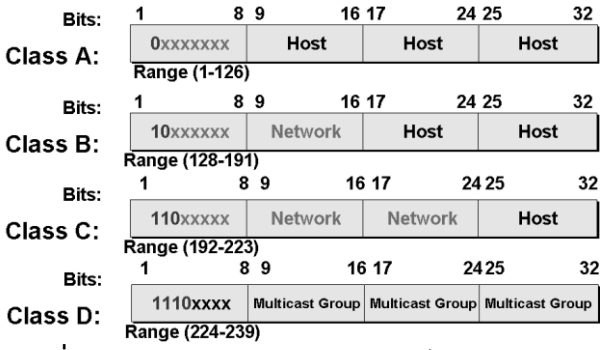
คลาส B

เหมาะสำหรับการใช้งานกับองค์กรขนาดกลาง โดยกำหนดให้ 2 บิตแรกของOctet แรกมีค่าเป็น 10 ดังนั้นแอดเดรสของเครือข่ายจะมี 14 บิตและอีก 2 Octet ที่เหลือใช้แทนแอดเดรสของเครื่องคอมพิวเตอร์ ในคลาสนี้สามารถจัดแบ่งจำนวนเครือข่ายออกได้ทั้งสิ้น 16,384 เครือข่าย (จริง ๆ สามารถได้เพียง 16,384-2) โดยที่แต่ละเครือข่ายรองรับจำนวนคอมพิวเตอร์ได้ 65,534 เครื่อง

คลาส C

เป็นคลาสที่มีการกำหนดใช้งานกับเครื่องคอมพิวเตอร์ส่วนใหญ่ภายในเครือข่ายอินเตอร์เน็ต โดยกำหนดว่าข้อมูลสามบิตแรกของ Octet แรกมีค่าเป็น 110 ดังนั้นไอพีแอดเดรสที่ใช้แทนแอดเดรสของเครือข่ายทีทั้งหมด 21 บิต ส่วน Octet สุดท้าย (8 บิตสุดท้าย) ใช้แทนแอดเดรสของเครื่องคอมพิวเตอร์ ซึ่งหมายความว่าคลาส C นี้สามารถแบ่งเครือข่ายออกได้เป็นทั้งสิ้น 2,097,152 เครือข่าย โดยแต่ละเครือข่ายมีจำนวนเครื่องคอมพิวเตอร์ได้ทั้งสิ้น 254 เครื่อง การคำนวณจำนวน

แอดเดรสของเครือข่าย และแอดเดรสของเครื่องคอมพิวเตอร์ สามารถทำได้โดยการคำนวณเลขฐานสอง โดยต้องมีการหักแอดเดรสที่ไม่อนุญาตให้ใช้ได้ 2 ชุด คือ แอดเดรสแรกของเน็ตเวิร์คและค่าสุดท้ายคือค่าที่เป็น 0 และ 255



รูปที่ 6.5 การแบ่งคลาสของหมายเลขไอพีแอดเดรส

คลาส D

ถ้าไอพี 3 บิตแรกเป็น 111 จะถือว่าไอพีนั้นเป็นไอพีพิเศษที่ไม่อนุญาตให้ใช้งานได้ ซึ่งปัจจุบันจะใช้งานประเภทการส่งข้อมูลเป็น Multicast

Class A ค่าของอพีแอดเดรสมีค่าต่ำกว่า 128

Class B ค่าของอพีแอดเดรสมีค่าตั้งแต่ 128 ถึง 191

Class C ค่าของอพีแอดเดรสมีค่าต่ำกว่า 192 ถึง 233

Class D ค่าของอพีแอดเดรสมีตั้งแต่ 224 ขึ้นไป

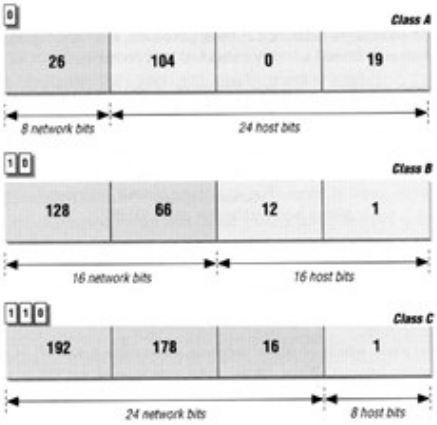
หมายเลขไอพีที่ไม่ได้ใช้นอกเหนือจาก 224 แล้วยังมีอีกหลายตัวดังนี้

0 ถูกนำไปใช้แทน default route

127 ถูกนำไปใช้แทน loopback Address สำหรับใช้จำลองการส่งข้อมูลภายในเครื่อง เพื่อให้สามารถทำงานเหมือนกับการส่งผ่านไปบนเครือข่ายจริง

หมายเลข 0 ของแต่ละเน็ตเวิร์ค จะถูกใช้ในการอ้างถึงตัวเน็ตเวิร์คเช่น 203.28.55.0

หมายเลข 255 ของแต่ละเน็ตเวิร์ค จะถูกใช้ในการอ้างถึงการส่งบอดคลาสเช่น 203.28.55.255



รูปที่ 6.6 แสดงโครงสร้างของไอพีแอดเดรส

การจัดสรรไอพีแอดเดรส (Subnet Addressing)

จากหลักเกณฑ์ในการกำหนดหมายเลขไอพีแอดเดรสที่กล่าวในหัวข้อก่อนหน้านั้นจะก่อให้เกิดความสิ้นเปลือง โดยเฉพาะอย่างยิ่งในกรณีของการจัดสรรหมายเลขไอพีแอดเดรสคลาส C ซึ่งมีการใช้งานกันทั่วไป ตัวอย่างเช่น เครือข่ายคอมพิวเตอร์แห่งหนึ่ง ต้องการสร้างระบบเครือข่ายอินทราเน็ตและเชื่อมต่อเข้ากับอินเตอร์เน็ต จำนวนเครื่องคอมพิวเตอร์ภายในเครือข่ายมีอยู่ทั้งสิ้น 30 เครื่อง สมมุติว่าได้รับการจัดสรรไอพีจาก InterNIC ให้ใช้แอดเดรสเป็นหมายเลข 202.28.35.10 ซึ่งเป็นไอพีแอดเดรสในคลาส C ดังนั้นไอพีแอดเดรสมีขนาด 254 ไอพี (256-2-30 = 224) แอดเดรสที่เหลือจากการใช้งานอีก 224 อาจจะไม่ได้ถูกใช้งานเลยจึงทำให้เสียแอดเดรสไปแบบสูญเปล่า อีกตัวอย่างหนึ่ง ซึ่งเกี่ยวข้องกับการจัดเครือข่ายคอมพิวเตอร์ภายในองค์กรออกเป็นเซ็กเมนต์ โดยใช้อุปกรณ์บริดจ์ สวิตช์ หรือเราเตอร์แยกเครือข่ายออกจากกัน เนื่องจากเหตุผลทางด้านการลดการรบกวนกันของสัญญาณหรืออาจจะเป็นการจัดวางตามสภาพของตึกสำนักงานออกจากกันหรือแยกตามภาระหน้าที่ สมมุติว่าเครื่องคอมพิวเตอร์ทั้งหมดในองค์กรอยู่ทั้งสิ้น 150 เครื่อง เชื่อมต่อกันเป็นเครือข่ายอินทราเน็ต มีการแบ่งกลุ่มการเชื่อมต่อเครื่องคอมพิวเตอร์ออกเป็นเซ็กเมนต์ทั้งสิ้น 3 เซ็กเมนต์ ๆ ละ 50 เครื่อง ถ้าไม่มีการทำ Subnet จะต้องมีการขอเลขหมายไอพีแอดเดรสในคลาส C จำนวน 3 คลาส เพื่อจัดสรรแอดเดรสให้กับเครื่องคอมพิวเตอร์ในแต่ละเซ็กเมนต์ การทำเช่นนั้นจะก่อให้เกิดความสิ้นเปลืองไอพีแอดเดรสเป็นจำนวนมาก หากพิจารณาเพียงหนึ่งเซ็กเมนต์จะเห็นว่าต้องสูญเสียหมายเลขไอพีแอดเดรสไปถึงเซ็กเมนต์ละ 256-2-50 = 224 แอดเดรส ผู้อ่านพึงควรตระหนักว่าหมายเลขไอพีแอดเดรสแม้จะมีอยู่เป็นจำนวนมหาศาล ณ ปัจจุบันมีแนวโน้มว่าจะไม่เพียงพอต่อความต้องการใช้งาน เนื่องจากจำนวนเครื่องคอมพิวเตอร์ที่มีการเชื่อมต่อกับเครือข่ายอินเตอร์เน็ตทวีจำนวนมากขึ้นเป็นทวีคูณตลอดเวลา แนวทางในการบริหารเลขหมายไอพีแอดเดรสในทางปฏิบัติจำเป็นต้องทำความเข้าใจกับเรื่อง Subnet (ซับเน็ท) ซึ่งมาคู่กับเลขหมายไอพีแอดเดรสตลอดเวลา การกำหนดค่าไอพีแอดเดรสควบคู่กับ Subnet จะทำให้ผู้ดูแลเครือข่ายสามารถจัดสรรจำนวนไอพีแอดเดรสที่เหมาะสมให้กับเครื่องคอมพิวเตอร์ในแต่ละเซ็กเมนต์ เพื่อเป็นการสร้างความเข้าใจของการจัดการไอพีแอดเดรสร่วมกับ Subnet ผู้เขียนจะขอให้ลองพิจารณาถึงการจัดสรรไอพีแอดเดรสในกลุ่มคลาส C ซึ่งผู้อ่านทราบดีแล้วว่า ข้อมูลภายใน 3 Octet แรก ใช้แทนแอดเดรสของเครือข่าย และ Octet สุดท้ายใช้แทนแอดเดรสของเครื่องคอมพิวเตอร์ที่เชื่อมต่อภายในเครือข่ายนั้น ในการแทนค่าข้อมูลในแต่ละ Octet เราจะแทนค่าด้วยตัวเลขฐาน 2 โดยกำหนดให้ข้อมูลใน 3 Octet แรกมีค่าเป็น "1" ทั้งหมดเพื่อใช้แทนหมายเลขเครือข่าย และกำหนดค่า "0" ให้กับทุกบิตใน Octet สุดท้ายเพื่อแทนเครื่องคอมพิวเตอร์หรือโฮสภายในเครือข่ายซึ่งเรียกว่า Subnet Mask

11111111.1111111.11111111.00000000 ฐาน 2
255.255.255.0 ฐานสิบ

รูปที่ 6.7 แสดง Subnet Mask ของคลาส C

จากรูปที่ 6.7 เป็นค่า Subnet Mask ของกลุ่มไอพีแอดเดรสในคลาส C ซึ่งแทนค่าเป็นเลขฐานสิบได้ 255.255.255.0 สำหรับค่า Subnet ในคลาสอื่น ๆ เช่นคลาส A จะมีค่าเป็น 255.0.0.0 และคลาส B เป็น 255.255.0.0 ตามลำดับ จากข้อกำหนดของไอพีแอดเดรสคลาส C จะรองรับจำนวนเครื่องคอมพิวเตอร์จำนวน 254 (256-2) เครื่อง เทคนิคที่จะใช้ในการแบ่งย่อยจำนวนเครือข่ายจากกลุ่มไอพีแอดเดรสคลาส C สามารถทำได้โดยการกำหนดค่าของเครื่องที่ใช้งานให้พอดีกับจำนวนบิตของไอพีแอดเดรส โดยการใช้บิตของกลุ่มของหมายเลขเครื่องใน Octet ที่ 4 คำนวณค่าให้พอดีกับเครื่องที่ต้องการใช้ ลำดับแรกทดลองปรับค่าของบิตของ Subnet Mask จากตัวอย่าง ทดลองเปลี่ยนค่า Subnet จากค่าปกติคือ 255.255.255.0 (ในตัวอย่างนี้ใช้คลาส C)

จาก 11111111.11111111.11111111.00000000 หรือ 255.255.255.0
เพิ่มบิตใน Octet ที่ 4 เพิ่มอีก 4 บิต (คือการเติม 1 เพิ่ม 4 ตัว)
11111111.11111111.11111111.11110000 หรือ 255.255.255.240

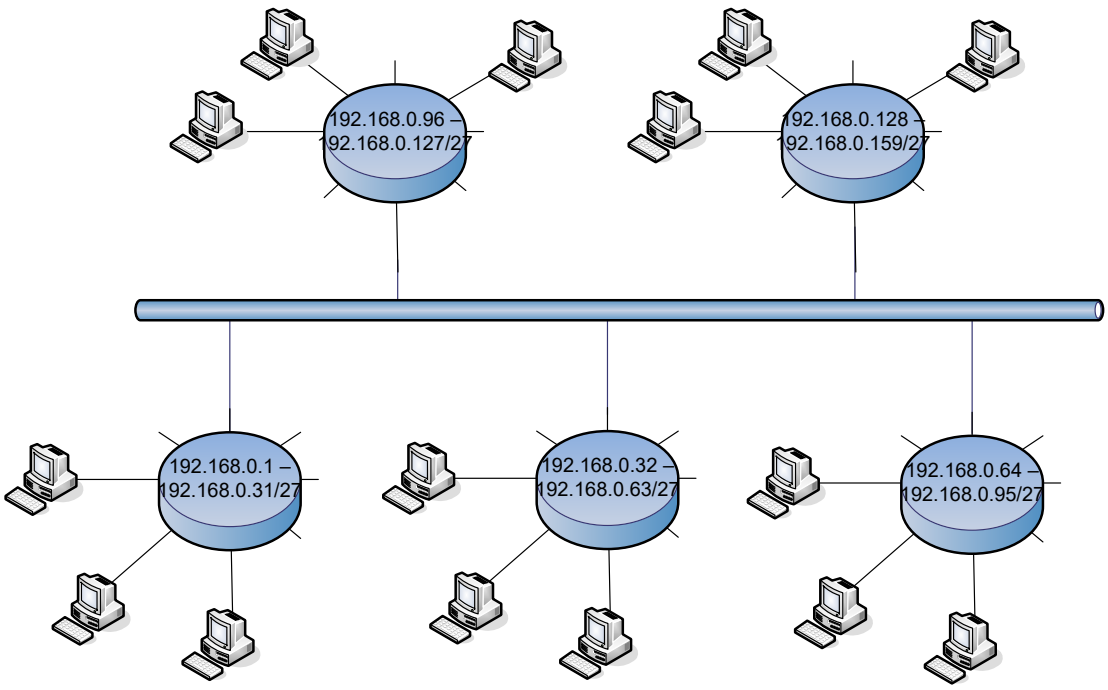
ซึ่งเมื่อคำนวณจำนวน Subnet หรือเครือข่ายที่ย่อยที่เกิดขึ้นในคลาส C แล้วสามารถมีจะมีค่าได้เท่ากับ 256-240 = 16 แล้วลบออก 2 จะได้จำนวนเครื่องที่ใช้งานได้ในแต่ละเน็ตเวิร์คคือ 14 เครื่อง ส่วนจำนวนของเน็ตเวิร์คจะคำนวณได้จาก บิตที่เป็น 1 เพิ่มขึ้น 4 บิตคือ 16-2 =14 เครือข่ายย่อย

ถ้าจะสรุปเพื่ออ่านเกิดความเข้าใจที่ชัดเจนขึ้นคือ หมายเลขแอดเดรสที่ท่านต้องเกี่ยวข้องกับการจัดสรรให้กับเครือข่ายคอมพิวเตอร์นั้น ประกอบไปด้วยไอพีแอดเดรสซึ่งได้รับการกำหนดจาก InterNIC โดยทั่วไปมักเป็นแอดเดรสคลาส C ที่มีอยู่ทั้งสิ้น 2,097,152 ชุด แต่ละชุดถูกกำหนดให้แต่ละองค์กรหรือบริษัท โดยในแอดเดรสคลาส C แต่ละชุดสามารถนำไปใช้กำหนดให้กับเครื่องคอมพิวเตอร์ภายในองค์กรได้ 254 เครื่อง สำหรับข้อมูล Subnet Mask ซึ่งมีรูปแบบเป็นข้อมูล 32 บิตเช่นเดียวกับไอพีแอดเดรสจะเป็นข้อมูลอีกชนิดหนึ่งที่ท่านต้องใช้กำกับให้กับเครื่องคอมพิวเตอร์ภายในเครือข่ายขององค์กรด้วย ข้อมูลภายใน Subnet Mask ในกรณีของการใช้ไอพีแอดเดรสคลาส C นั้นจะอยู่ในรูปแบบเลขฐานสองเป็น 11111111.11111111.11111111.XXXXXXXX โดยผู้ดูแลระบบมีสิทธิในการแบ่งกลุ่มเครือข่ายย่อย ๆ ออกได้ตามสภาพการใช้งานภายในองค์กร การหนดค่า "1" ให้กับบิต X ใด ๆ จะหมายความว่าให้บิตดังกล่าวเป็นส่วนขยายของแอดเดรสเครือข่าย ในทำนองกลับกันการกำหนดค่า "0" ให้กับบิตใด ๆ ก็จะเป็นการแจ้งให้เครื่องคอมพิวเตอร์และเครือข่ายภายในองค์กรทราบว่าบิตนั้น ๆ ถูกใช้ในการกำหนดค่าแอดเดรสให้กับเครื่องคอมพิวเตอร์แต่ละเครื่องภายในเครือข่ายย่อยนั้น ๆ

ตารางที่ 6.1 แสดงการแบ่ง Subnet ของคลาส C

จำนวนบิตที่ใช้ทำ Subnet	Subnet Mask ฐานสิบ	Subnet Mask ฐานสอง(Octet 4)	จำนวนเครือข่ายที่ได้จากการทำ Subnet	จำนวนโฮสต์ที่ได้ในแต่ละเครือข่ายหลัง Subnet
2	255.255.255.192	X.X.X.11000000	2 บิต 2^2 = 4 -2 =2	2^6 = 64 -2 = 62
3	255.255.255.224	X.X.X.11100000	2^3 = 8 -2 = 6	32 - 2 = 30
4	255.255.255.240	X.X.X.11110000	16 – 2 = 14	16 -2 = 14
5	255.255.255.248	X.X.X.11111000	32 – 2 = 30	8 – 2 = 6
6	255.255.255.252	X.X.X.11111100	64 – 2 = 60	4 – 2 = 2
7	255.255.255.254	X.X.X.11111110	128 – 2 = 126	2 – 2 = 0 ไม่ใช้งาน
8	255.255.255.255	X.X.X.11111111	256 – 2 = 254	ไม่ใช้งาน

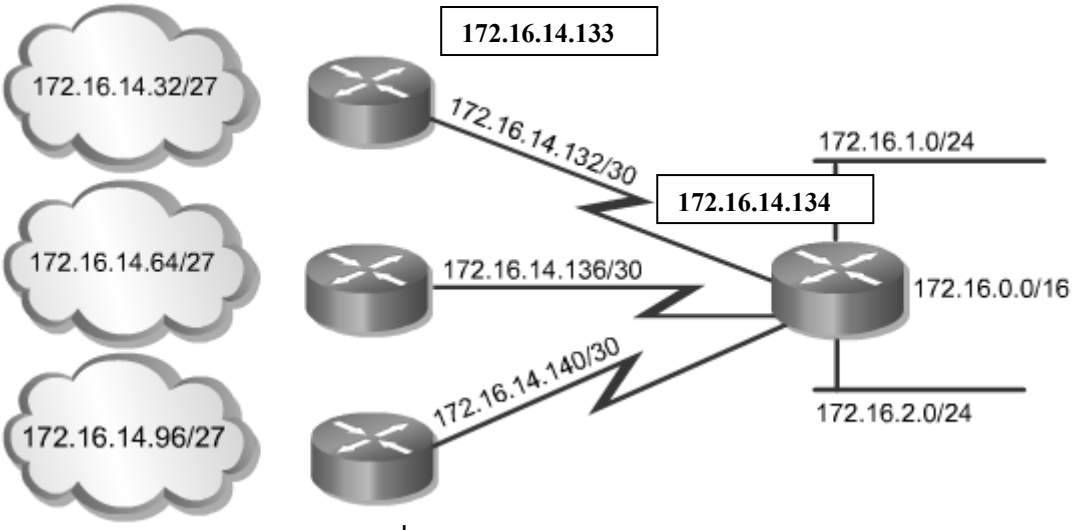
ในการกำหนดค่า Subnet Mask สำหรับการใช้งานร่วมกับเครือข่ายคลาส C สามารถดูได้จากตารางที่ 6.1 โดยผู้ดูแลระบบจะต้องพิจารณาให้ดีเสียก่อนว่าจะมีการแบ่งเซ็กเมนต์กลุ่มของเครื่องคอมพิวเตอร์ภายในเครือข่ายขององค์กรออกเป็นกี่กลุ่ม และแต่ละกลุ่มจะมีจำนวนเครื่องคอมพิวเตอร์เชื่อมต่ออยู่มากที่สุดนั้นมีอยู่กี่เครื่อง ตารางที่ 6.1 เป็นการสรุปแนวทางในการกำหนดค่า Subnet Mask ให้กับเครือข่ายคลาส C ตัวอย่างจากการหาค่าจากตาราง ถ้าต้องการให้เครื่องภายในเครือข่ายหรือเซ็กเมนต์หนึ่ง ๆ มีจำนวนเครื่องทั้งหมดไม่เกิน 30 เครื่อง ถ้าดูค่าจากตารางเราควรใช้ Subnet เท่ากับ 255.255.255.224 ซึ่งจะได้จำนวนของเน็ตเวิร์คเท่ากับ 6 เน็ตเวิร์ค เป็นต้น เมื่อนำมาวางระบบเครือข่ายจะได้ดังรูปที่ 6.8



รูปที่ 6.8 ตัวอย่างการใช้งาน 6 Subnet, Subnet ละ 30 เครื่อง

การทำ Subnet ซ้อน Subnet (Variable-Length Subnet Mask VLSM)

VLSM คือการทำ Subnet ภายใน Subnet อีกที ดังรูปที่ 6.9 ที่ต้องทำอย่างนี้เนื่องจากมีเหตุผลอยู่หลายอย่าง ที่สำคัญที่สุดคือ กรณีที่ต้องการให้เครือข่ายใดเครือข่ายหนึ่งมีขนาดพอดีกับการใช้งาน เช่นขา WAN Link ขององค์กรซึ่งต้องการไอพีเอดเดรสเพียงแค่ 4 ไอพี ต่อการเชื่อมต่อ 1 Link 2 ไอพีสำหรับเป็นเน็ตเวิร์คและบรอสคลาส ที่เหลืออีก 2 ไอพีเพื่อใช้กำหนดให้ขาอินเตอร์เฟสของเราเตอร์ที่เชื่อมต่อกัน ดังรูปที่ 6.9 จากรูป WAN Link ที่เชื่อมต่อไปที่เน็ตเวิร์ค 172.16.14.32/27 จะใช้เน็ตเวิร์คหมายเลข 172.16.14.32 ส่วนบรอดคลาสเป็นหมายเลข 172.16.14.35 ส่วนไอพี 172.16.14.33 และ 172.16.14.34 จะใช้เป็นหมายเลขประจำขาของอินเตอร์เฟสของเราเตอร์ จะเห็นได้ว่าไอพีที่ต้องการในกรณีนี้ใช้เพียง 4 ไอพีถ้าไม่มีการทำ VLSM จะทำให้ต้องเสียไอพีไปเป็นจำนวนมากและไม่คุ้มค่ากับการใช้งาน แต่การทำ VLSM มากไปก็จะทำให้เสียไอพีไปในเน็ตเวิร์คและบรอสคลาสมากเช่นเดียวกัน ดังนั้นก่อนจะใช้ควรพิจารณาถึงจุดนี้ด้วย



รูปที่ 6.9 VLSM ของ WAN Link

ตัวอย่างการทำ VLSM

สมมุติว่ามีไอพีแอดเดรสที่ผ่านการทำ Subnet มาแล้วคือ

172.16.32.0/20

10101100.00010000.**0010**0000.00000000 ส่วนที่ขีดเส้นไว้คือบิตที่ทำ Subnet /20 ต้องการทำ VLSM โดยให้มีจำนวนเครื่องที่ต้องการใช้งานใน 1 เน็ตเวิร์ค ไม่เกิน 60 เครื่อง บิตที่ต้องใช้คือ 6 บิต = 64 – 2 ซึ่งเพียงพอ ดังนั้นการคำนวณจะได้ดังนี้

10101100.00010000.**0010**0000.00000000 บิตที่เอียงคือบิตที่ใช้คำนวณจำนวนเครื่อง

10101100.00010000.**00100000.00**000000 บิตที่เป็นตัวหนาจะใช้สำหรับทำ VLSM ซึ่งจะได้ Subnet ของ VLSM เป็น

จำนวนบิตของ Subnet เดิม + จำนวนบิตที่ทำ VLSM

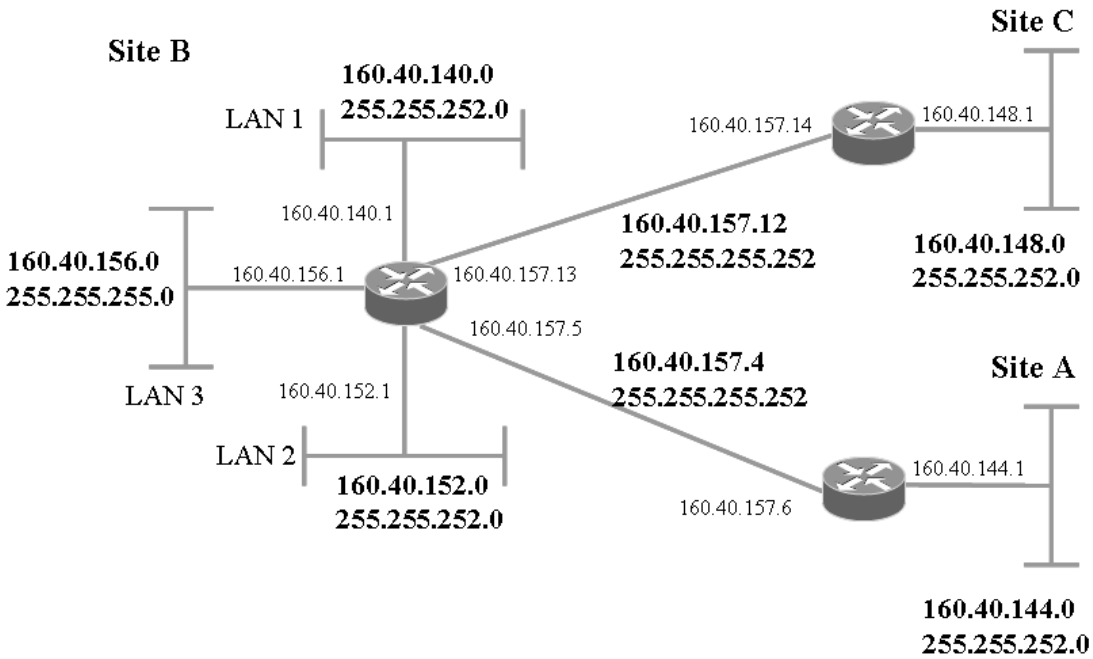
20 + 6 = 26

เพราะฉะนั้นหมายเลข Subnet Mask ใหม่จะมีค่าเท่ากับ 172.16.32.0/26 ดังรูปที่ 6.10

1st subnet:	172	•	16	.0010	0000.00	000000 = 172.16.32.0/26
2nd subnet:	172	•	16	.0010	0000.01	000000 = 172.16.32.64/26
3rd subnet:	172	•	16	.0010	0000.10	000000 = 172.16.32.128/26
4th subnet:	172	•	16	.0010	0000.11	000000 = 172.16.32.192/26
5th subnet:	172	•	16	.0010	0001.00	000000 = 172.16.33.0/26
	Network			Subnet	VLSM Subnet	Host

รูปที่ 6.10 แสดงการหา Subnet Mask ของ VLSM

ตัวอย่างการทำ VLSM ในงานจริง (ดังรูปที่ 6.11)



รูปที่ 6.11 VLSM ในงานจริง

💻 เรียนรู้เบื้องต้นกับ IP (Internet Protocol)

หมายเลขไอพีเป็นตัวบ่งบอกถึงตำแหน่งที่อยู่บนเครือข่ายอินเตอร์เน็ตเพื่อใช้สำหรับติดต่อและสื่อสารกัน บนเราเตอร์สามารถกำหนดไอพีได้ไม่ยาก แต่การคำนวณไอพีแอดเดรส ซับเน็ต และโฮสต์ ค่อนข้างยุ่งยาก ตารางที่ 6.2 แสดงหมายเลขของ subnet mask ทั้งหมด

ตารางที่ 6.2 การหาค่า subnet mask ของคลาส A, B, C

Slash	Dotted Decimal	Slash	Dotted Decimal	Slash	Dotted Decimal
/8	255.0.0.0	/16	255.255.0.0	/24	255.255.255.0

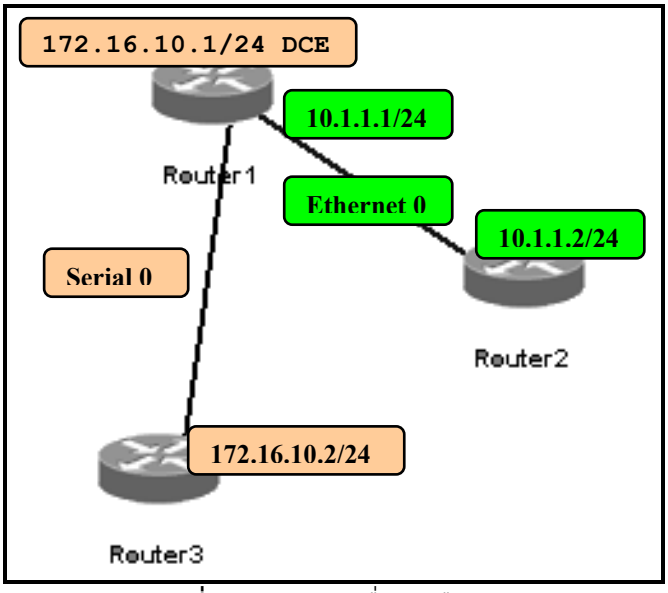
/9	255.128.0.0		/17	255.255.128.0		/25	255.255.255.128
/10	255.192.0.0		/18	255.255.192.0		/26	255.255.255.192
/11	255.224.0.0		/19	255.255.224.0		/27	255.255.255.224
/12	255.240.0.0		/20	255.255.240.0		/28	255.255.255.240
/13	255.248.0.0		/21	255.255.248.0		/29	255.255.255.248
/14	255.252.0.0		/22	255.255.252.0		/30	255.255.255.252
/15	255.254.0.0		/23	255.255.254.0		/31	255.255.255.254

จุดมุ่งหมาย : เรียนรู้การใช้คำให้แต่อินเตอร์เฟสทำงานและหยุดทำงาน

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 3 ตัวคือ Router1, Router2, Router3

การสร้าง Network Map : ดังรูปที่ 6.12

1. ที่หน้าต่างหนัก (Simulator) เลือก File → new NetMap → จะปรากฏหน้าต่าง Boson Network Design
2. ให้คลิกเลือก Available Routers → 2500 series → ดับเบิลคลิก 2516 → ตั้งชื่อป็น Router1
3. เลือก Available Routers → 2500 series → ดับเบิลคลิก 2501 → ตั้งชื่อป็น Router2
4. เลือก Available Routers → 2500 series → ดับเบิลคลิก 2501 → ตั้งชื่อป็น Router3
5. คลิกขวาที่ Router1 → Add Connection to ให้เลือก Ethernet 0 → ในช่อง Available Devices ให้เลือก Router 2 → ในช่อง Ethernet Interfaces ให้เลือก Ethernet 0 →Finish
6. คลิกขวาที่ Router1 → Add Connection to ให้เลือก Serial 0 → เลือก Point-to-Point Serial Connection → Next → ในช่อง Available Devices ให้เลือก Router 3 → ในช่อง Ethernet Interfaces ให้เลือก Serial 0 →Finish → เลือก DEC เป็น Router1, Serial 0 → แล้วกดปุ่ม OK จากนั้นให้บันทึกผังเน็ตเวิร์คโดยมีนามสกุลเป็น .top แล้วให้โหลดไปยังหน้าต่างหลัก (Simulator) โดยการเลือกที่ File → Load NetMap into the Simulator



รูปที่ 6.12 ผังการเชื่อมเครือข่าย

หน้าต่าง Simulator :

1. Router> [ให้เข้าไปที่ Router1]

   Router>*enable*

   Router#*config t*

   Router(config)#*hostname Router1* [ตั้งชื่อเป็น Router1]

   Router1(config)#

2. Router1(config)#*interface ethernet 0* [เข้าสู่โหมดคอนฟิกอินเตอร์เฟสอีเทอร์เน็ตหมายเลข 0]

3. Router1(config-if)#*ip address 10.1.1.1 255.255.255.0* [กำหนดไอพีแอดเดรสให้กับอินเตอร์เฟสอีเทอร์เน็ตหมายเลข 0 เป็น 10.1.1.1 ซับเน็ต 255.255.255.0 หรือ /24]

4. Router1(config-if)#*no shutdown* [สั่งให้อินเตอร์เฟสอีเทอร์เน็ต 0 ทำงาน] ดังรูปที่ 6.13

```
Router1(config)#interface ethernet 0
Router1(config-if)#ip address 10.1.1.1 255.255.255.0
Router1(config-if[redacted]
%LINK-3-UPDOWN: Interface Ethernet0, [redacted]
Router1(config-if)#
```

รูปที่ 6.13 สั่งให้อินเตอร์เฟสอีเทอร์เน็ตทำงาน

5. Router1(config-if)#*exit* [ออกไปสู่โหมดคอนฟิก]

   Router1(config)#*interface serial 0* [เข้าสู่โหมดคอนฟิกอินเตอร์เฟสซีเรียล 0]

Router1(config-if)#*ip address 172.16.10.1 255.255.255.0* [กำหนดไอพีแอดเดรสให้กับอินเตอร์เฟสซีเรียลหมายเลข 0 เป็น 172.16.10.1 ซับเน็ต 255.255.255.0 หรือ /24]

Router1(config-if)#*clock rate 5600* [กำหนดสัญญาณนาฬิกาให้กับอินเตอร์เฟสซีเรียล 0 เป็นจำนวน 5600 เพราะว่าซีเรียล 0 เป็น DEC]

Router1(config-if)#*no shutdown* [สั่งให้อินเตอร์เฟสนี้ทำงาน]

6. **Router>** [เข้าไปที่ Router2]

   Router>*enable*

   Router#*config t*

   Router(config)#*hostname Router2* [ตั้งชื่อเป็น Router2]

   Router2(config)#

7. **Router2(config)#*interface ethernet 0*** [เข้าสู่โหมดคอนฟิกอินเตอร์เฟสอีเทอร์เน็ตหมายเลข 0]
8. **Router2(config-if)#*ip address 10.1.1.2 255.255.255.0*** [กำหนดไอพีแอดเดรสให้กับอินเตอร์เฟสอีเทอร์เน็ตหมายเลข 0 เป็น 10.1.1.2 ซับเน็ต 255.255.255.0 หรือ /24]
9. **Router2(config-if)#*no shutdown*** [สั่งให้อินเตอร์เฟสอีเทอร์เน็ต 0 ทำงาน]
10. **Router>** [เข้าไปที่ Router3]

    Router>*enable*

    Router#*config t*

    Router(config)#*hostname Router3* [ตั้งชื่อเป็น Router3]

    Router3(config)#*interface serial 0* [เข้าสู่โหมดคอนฟิกอินเตอร์เฟสซีเรียลหมายเลข 0]

    Router3(config-if)#*ip address 172.16.10.2 255.255.255.0* [กำหนดไอพีแอดเดรสให้กับอินเตอร์เฟสซีเรียลหมายเลข 0 เป็น 172.16.10.2 ซับเน็ต 255.255.255.0 หรือ /24]

    Router3(config-if)#*no shutdown* [สั่งให้อินเตอร์เฟสซีเรียล 0 ทำงาน]

11. เข้าไปที่ Router1 อีกครั้ง

    Router1#*ping 10.1.1.2* [ping เพื่อตรวจสอบว่าการเชื่อมต่อสมบูรณ์ไปยังอินเตอร์เฟสอีเทอร์เน็ตของเราเตอร์ 2 ถ้าปรากฏเครื่องหมาย !!!!! แสดงว่าการเชื่อมต่อสมบูรณ์] ดังรูปที่ 6.14

```
Router1#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
```

รูปที่ 6.14 ทดสอบการเชื่อมต่อไปยัง Router2 ด้วยคำสั่ง ping

12. Router1*#ping 172.16.10.2* [ping เพื่อตรวจสอบว่าการเชื่อมต่อสมบูรณ์ไปยังอินเตอร์เฟสซีเรียล 0 ของเราเตอร์ 3] ดังรูปที่ 6.15

```
Router1#ping 172.16.10.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
!!!!!
```

รูปที่ 6.15 ทดสอบการเชื่อมต่อไปยัง Router3 ด้วยคำสั่ง ping

13. Router1*#show ip interface brief* [แสดงสถานะของอินเตอร์เฟสและโพรโทคอล ว่าทำงานอยู่หรือไม่] ดังรูปที่ 6.16

```
Router1#sh ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
Serial0                172.16.10.1     YES unset  up                    up
Serial1                unassigned      YES unset  administratively down down
Ethernet0              10.1.1.1        YES unset  up                    up
Bri0                   unassigned      YES unset  administratively down down
Bri0:1                 unassigned      YES unset  administratively down down
Bri0:2                 unassigned      YES unset  administratively down down
```

รูปที่ 6.16 แสดงข้อมูลของ Interface line status และ Protocol status

14. Router1*#show running-config* [ตรวจสอบไอพีแอดเดรสที่ได้กำหนดไว้ โดยดูที่รันนิ่งคอนฟิก ให้สังเกตที่อินเตอร์เฟสอีเทอร์เน็ตหมายเลข 0 และอินเตอร์เฟสซีเรียลหมายเลข 0 ว่าตรงกับที่กำหนดหรือไม่] ดังรูปที่ 6.17

```
interface Serial0
 no ip directed-broadcast
 clock rate 56000
 bandwidth 1544
!
interface Serial1
 no ip address
 no ip directed-broadcast
 bandwidth 1544
 shutdown
!
interface Ethernet0
 no ip directed-broadcast
 bandwidth 10000
```

รูปที่ 6.17 แสดง running-config บน Router1

15. **Router1#*sh ip interface*** [แสดงรายละเอียดของแต่ละอินเตอร์เฟสบน Router1]
หมายเหตุ: เมื่อเจอข้อความแสดงบนเราเตอร์เป็น - -MORE – ให้กดปุ่ม Space bar เพื่อทำงานต่อไป โดยจะแสดงทีละหน้าจอภาพ แต่ถ้ากดปุ่ม Enter จะแสดงต่อไปทีละบรรทัด
หมายเหตุ: ถ้าไม่ต้องการดูรายการต่อให้กดปุ่ม <ctl>+c เพื่อยกเลิกคำสั่งใด ๆ ในขณะนั้น

ARP

ARP เป็นวิธีในการค้นหาตำแหน่งที่อยู่ของอุปกรณ์ โดยกระทำผ่าน MAC Address ที่มีอยู่ประจำแต่ละตัวอุปกรณ์ที่มีค่าไม่ซ้ำกัน เราเตอร์จะมีตารางสำหรับจัดเก็บ MAC Address เรียกว่า ARP Table ในตาราง ARP นี้จะมีข้อมูลของ MAC Address, หมายเลขไอพีแอดเดรสของแต่ละอินเตอร์เฟส เราเตอร์จะต้องใช้ตารางนี้ในการหาที่อยู่ของอุปกรณ์แต่ละตัวที่เชื่อมต่อกับมัน โดยการดูจากหมายเลข MAC ดังนั้นตารางนี้มีความจำเป็นอย่างยิ่ง ถ้าข้อมูลในตาราง ARP ผิดพลาด จะทำให้การรับส่งข้อมูลผิดพลาดด้วย ผู้ดูแลระบบอาจจำเป็นที่จะต้องเข้ามาจัดการลบและสร้างตารางของ ARP ให้ถูกต้อง เราสามารถดูข้อมูลจากตาราง ARP ด้วยคำสั่ง show arp

จุดมุ่งหมาย : เรียนรู้การทำงานของ ARP

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัวคือ Router1, Router2

การสร้าง Network Map : สร้างเราเตอร์ 2 ตัวชื่อ Router1, Router2

1. หน้าต่างหลัก → เลือก File → New NetMap → OK
2. หน้าต่าง Boson Network Design → คลิกที่ Available Routers → เลือก 2500 series → คลิกเลือกเราเตอร์รุ่น 2505 แล้วลากไปวางยังผังเน็ตเวิร์ค (มีสีเทา) → ให้คลิก apply โดยไม่ต้องตั้งชื่อ(Boson จะตั้งชื่อว่า Router1 ให้อัตโนมัติ)
3. หน้าต่าง Boson Network Design → คลิกที่ Available Routers → เลือก 2500 series → คลิกเลือกเราเตอร์รุ่น 2505 แล้วลากไปวางยังผังเน็ตเวิร์ค (มีสีเทา) → ให้คลิก apply โดยไม่ต้องตั้งชื่อ (Boson จะตั้งชื่อว่า Router2 ให้อัตโนมัติ)
4. คลิกขวาที่ Router1 → เลือก add Connection to ให้ใช้ ethernet 0 → ในช่อง Available Devices ให้เลือก Router2 ในช่อง ethernet interfaces ให้เลือก ethernet 0 → กด Finish
5. หน้าต่าง Boson Network Design → ให้เลือก File → Save as → ใส่ชื่อเป็น Lab10.top แล้วกด OK
6. หน้าต่าง Boson Network Design → Load NetMap into the Simulator → กดปุ่ม OK

หน้าต่าง Simulator :

1. **Router>** [เราเตอร์ 1]

   Router>*enable*

   Router# [เข้าสู่โหมดของ privilege]

2. **Router#*show arp*** [แสดงตารางของ ARP ในตอนแรกจะยังไม่มีข้อมูลในตาราง ARP ต้องมีการกำหนดไอพีแอดเดรสและสั่งให้อินเตอร์เฟสทำงานก่อน] ดังรูปที่ 6.18

```
Router#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface

Router#
```

รูปที่ 6.18 ไม่มีข้อมูลในตาราง ARP

3. ให้กำหนดไอพีแอดเดรสของอินเตอร์เฟสอีเทอร์เน็ต 0 เป็น 10.1.1.1 ซับเน็ต 255.255.255.0 พร้อมกับสั่งให้ทำงาน

   Router#*conf t* [เข้าสู่โหมดคอนฟิก]

   Router(config)#*interface ethernet 0* [เข้าสู่โหมดอินเตอร์เฟส]

   Router(config-if)#*ip address 10.1.1.1 255.255.255.0* [กำหนดไอพีให้อินเตอร์เฟส]

   Router(config-if)#*no shutdown* [สั่งให้อินเตอร์เฟสอีเทอร์เน็ต 0 ทำงาน]

   Router(config-if)#*exit* [ออกไปสู่โหมดคอนฟิก]

4. **Router(config)#*exit***

   Router#*show arp* [แสดงข้อมูลในตาราง ARP] ดังรูปที่ 6.19

```
Router#sh arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
```

รูปที่ 6.19 ข้อมูลในตาราง ARP ของ Router1

5. เข้าไปที่ Router2 แล้วกำหนดหมายเลขไอพีแอดเดรสของอีเทอร์เน็ต 0 เป็น 10.1.1.2/24 และสั่งให้ทำงาน

6. **Router>** [เราเตอร์ 2]

   Router>*enable*

   Router#

   Router#*conf t* [เข้าสู่โหมดคอนฟิก]

   Router(config)#*interface ethernet 0* [เข้าสู่โหมดอินเตอร์เฟส]

   Router(config-if)#*ip address 10.1.1.2 255.255.255.0* [กำหนดไอพีให้อินเตอร์เฟส]

   Router(config-if)#*no shutdown* [สั่งให้อินเตอร์เฟสอีเทอร์เน็ต 0 ทำงาน]

   Router(config-if)#*exit* [ออกไปสู่โหมดคอนฟิก] ดังรูปที่ 6.120

```
Router#sh arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.1.1.2                -   000C.4809.8723  ARPA   Ethernet0
```

รูปที่ 6.20 ข้อมูลในตาราง ARP ของ Router2

7. ทดสอบการเชื่อมต่อจองเราเตอร์ทั้ง 2 โดยการใช้คำสั่ง ping จาก Router2 ไปยัง Router1

   Router#*ping 10.1.1.1* ดังรูปที่ 6.21

```
Router#ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 6.21 แสดงสถานการณ์ ping สมบูรณ์

8. ทดสอบลบตารางของ ARP โดยใช้คำสั่ง clear arp ที่ Router2

   Router#*clear arp* [สั่งลบข้อมูลในตาราง ARP เมื่อลบแล้วเราเตอร์จะสร้างข้อมูลใหม่ให้อัตโนมัติ]

9. ทดสอบแสดงข้อมูลในตาราง ARP อีกครั้งด้วยคำสั่ง show arp

สร้างตารางความสัมพันธ์ระหว่างไอพีแอดเดรสกับชื่อ

เราสามารถสร้างชื่อขึ้นมาแทนตัวเราเตอร์ได้ เพื่อความสะดวกในการอ้างถึงตัวเราเตอร์ ตัวอย่างเช่น ถ้าต้องการทำการทดสอบด้วยคำสั่ง ping ไปยังเราเตอร์ตัวใดตัวหนึ่ง ไม่จำเป็นต้องอ้างไอพีแอดเดรส สามารถอ้างชื่อแทนตัวเราเตอร์นั้น ๆ ได้เลย

จุดมุ่งหมาย : สร้างชื่อ (Host name) เพื่อใช้เรียกแทนเราเตอร์

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัวคือ Router1, Router2

การสร้าง Network Map : เหมือน LAB ARP

หน้าต่าง Simulator :

1. ที่ Router1

   Router>

   Router>*enable* [เข้าสู่โหมดของ privilege]

   Router#*config t* [เข้าสู่โหมดของคอนฟิก]

   Router(config)#*hostname Carifornia* [กำหนดชื่อเราเตอร์เป็น Carifornia]

   Carifornia(config)#

2. ให้กำหนดไอพีแอดเดรสที่อินเตอร์เฟสอีเทอร์เน็ต 0 เป็น 195.42.36.10 255.255.255.240 และสั่งให้ทำงาน

   Carifornia(config)#*interface ethernet 0*

   Carifornia(config-if)#*ip address 195.42.36.10 255.255.255.240*

   Carifornia(config-if)#*no shutdown*

3. ที่ Router2 กำหนดชื่อของเราเตอร์เป็น Tampa และหมายเลขไอพีแอดเดรสอีเทอร์เน็ต 0 เป็น 195.42.36.12 255.255.255.0 พร้อมสั่งให้ทำงาน

Router>

Router>*enable* [เข้าสู่โหมดของ privilege]

Router#*config t* [เข้าสู่โหมดของคอนฟิก]

Router(config)#*hostname Tampa* [กำหนดชื่อเราเตอร์เป็น Tampa]

Tampa(config)#

Tampa(config)#*interface ethernet 0*

Tampa(config-if)#*ip address 195.42.36.12 255.255.255.240*

Tampa(config-if)#*no shutdown*

4. ที่เราเตอร์ Tampa กำหนดให้ชื่อสำหรับเรียกแทนหมายเลขไอพีที่อินเตอร์เฟสอีเทอร์เน็ต 0 (เรียกชื่อ Carifornia แทนไอพี 195.42.36.10)

Tampa(config-if)#*exit*

Tampa(config)#*ip host Carifornia 195.42.36.10* [กำหนดชื่อเรียกแทนไอพี]

5. ทดสอบการทำงานโดยการ ping ชื่อแทนไอพี จาก Tampa

Tampa#*ping Carifornia* [ทดสอบด้วยการ ping] ดังรูปที่ 6.22

Tampa#*show host* [แสดงรายชื่อของโฮสต์] ดังรูปที่ 6.23

```
Tampa#ping Carifornia

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 195.42.36.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 6.22 ทดสอบการ ping Carifornia

```
Tampa#sh hosts

Carifornia                (perm, OK)  0   IP    195.42.36.10
```

รูปที่ 6.23 ผลลัพธ์จากคำสั่ง show hosts ที่เราเตอร์ Tampa

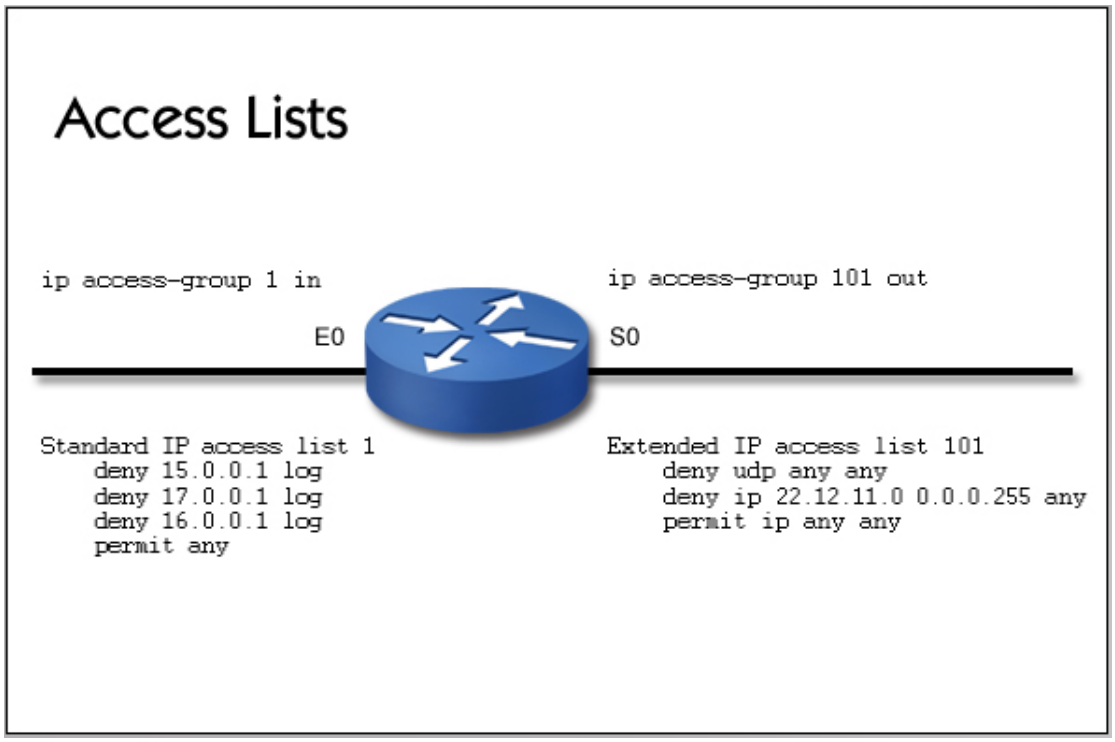
แบบฝึกหัดท้ายบท

1. เข้าไปที่ Router1 แล้วต้องการดูตาราง ARP จะใช้คำสั่งอะไร?_____
2. ให้กำหนดไอพีแอดเดรสที่อินเตอร์เฟสอีเทอร์เน็ต 0 เป็น 10.1.1.1/24 ?_____
3. ให้ตรวจสอบว่าในตาราง ARP มีข้อมูลอะไรบ้าง?_____
4. เข้าไปที่ Reouter2 และกำหนดไอพีแอดเดรสของอินเตอร์เฟสอีเทอร์เน็ต 0 เป็น 10.1.1.2/24?_____
5. ให้ทดสอบการเชื่อมต่อระหว่าง Router1 และ Router2 จะต้องทดสอบด้วยคำสั่งอะไร?_____

6. ถ้าต้องการสร้างตาราง ARP ใหม่ จะต้องใช้คำสั่งอะไรในการลบข้อมูลออกจากตาราง ARP?_____
7. ให้ตรวจสอบข้อมูลในตาราง ARP หลังจากทำข้อที่ 6 แล้ว ว่าเป็นอย่างไร?_____
8. ให้นิสิตคำนวณหา Network ID, Subnet และ broadcast
    a) 55.110.67.205/16
    b) 88.248.235.250 255.255.255.248
    c) 56.58.128.76 255.128.0.0
    d) 198.13.70.25 255.255.255.192
    e) 150.75.222.94/18
    f) 198.134.190.70/28
    g) 47.165.237.34/21
9. Is Ip address 79.246.255.1 with a subnet mask 255.224.0.0 a valid host ip address?
10. What is the first valid host on the subnetwork that the node 172.20.182.215 255.255.255.128 belongs to?
11. Which subnet does host 172.21.112.211/23 belong to?
12. What is the first valid host on the subnetwork that the node 10.245.110.177/20 belongs to?
13. What is the broadcast address of the network 172.16.144.0 255.255.248.0?
14. Which subnet does host 192.168.210.125/30 belong to?
15. Which subnet does host 172.27.239.230 255.255.240.0 belong to?
16. Which subnet does host 172.27.161.115 255.255.255.240 belong to?
17. What valid host range is the IP address 172.25.155.112 255.255.255.0 a part of?
18. Which subnet does host 172.23.210.182/21 belong to?
19. What is the last valid host on the subnetwork 10.5.208.0/20?
20. What is the first valid host on the subnetwork that the node 192.168.253.130/28 belongs to?
21. You are designing a subnet mask for the 172.24.0.0 network. You want 70 subnets with up to 300 hosts on each subnet. What subnet mask should you use?
22. What is the first valid host on the subnetwork that the node 172.25.118.106 255.255.255.0 belongs to?

23. Which subnet does host 172.26.32.39/25 belong to?
24. What is the broadcast address of the network 192.168.202.112 255.255.255.248?
25. What is the broadcast address of the network 172.25.100.0 255.255.255.0?
26. What is the last valid host on the subnetwork 172.19.222.112 255.255.255.240?
27. How many subnets and hosts per subnet can you get from the network 172.26.0.0 255.255.255.128?
28. What is the first valid host on the subnetwork that the node 192.168.51.193/26 belongs to?
29. What is the first valid host on the subnetwork that the node 172.30.181.215/23 belongs to?

บทที่ 7
การสร้างรายการควบคุมการเข้าถึง (Access Control Lists)



- Access list concepts
- Configuring access lists
- Working with wildcard masks
- Designing and monitoring access lists

แนวคิด

Access Lists คือลำดับของคำสั่งที่เรียงต่อกัน เพื่อใช้ควบคุมทิศทางการไหลของข้อมูลภายในเน็ตเวิร์คให้เป็นไปตามที่เราต้องการ เหตุผลของการใช้งาน Access Lists มีอยู่หลายประการ เช่น ควบคุมทิศทางของข้อมูล, การจำกัดปริมาณของข้อมูลในเครือข่าย, และเรื่องของความปลอดภัย ในส่วนนี้จะเน้นในเรื่องของความปลอดภัยเป็นหลัก

วัตถุประสงค์

1. เพื่อให้ทราบถึงขั้นตอนการวิเคราะห์และการติดตั้ง access control list
2. เพื่อให้ทราบถึงขั้นตอนการดำเนินการของ access control list
3. เพื่อใช้ควบคุมปริมาณและทิศทางการไหลของข้อมูลภายในเน็ตเวิร์ค

💻 Access Lists

Access Lists คือลำดับของคำสั่งที่เรียงต่อกัน เพื่อใช้ควบคุมทิศทางการไหลของข้อมูลภายในเน็ตเวิร์คให้เป็นไปตามที่เราต้องการ ซึ่งประเภทของคำสั่งที่ใช้งานมี 2 ประเภทคือ "ยอมให้ผ่าน" (permit) ตามด้วยเงื่อนไขที่ต้องการให้ผ่าน และอีกประเภทคือ "ไม่ยอมให้ผ่าน" (deny) และก็ตามด้วยเงื่อนไขที่ไม่ยอมให้ข้อมูลผ่าน เหตุผลของการใช้งาน Access Lists มีอยู่หลายประกาศ เช่น ควบคุมทิศทางของข้อมูล, การจำกัดปริมาณของข้อมูลในเครือข่าย, และเรื่องของความปลอดภัย ในส่วนนี้จะเน้นในเรื่องของความปลอดภัยเป็นหลัก การสร้าง Access Lists ในส่วนของความปลอดภัยก็สามารถแยกได้หลายประการเช่น การป้องกันการโจมตีจากภายนอก, การจำกัดขอบเขตุข้อมูลบางชนิดให้อยู่เฉพาะภายในหน่วยงานของตน, ไวรัสบนเน็ตเวิร์ค, และแอพพลิเคชันบางอย่างที่ส่งแพ็กเก็ตออมายังระบบเครือข่ายมากเกินไป เป็นต้น หรือสามารถใช้ Access Lists ทำหน้าที่เป็นไฟล์วอลล์ก็สามารถทำได้โดยการกำหนด policy ไว้กับเราเตอร์ที่ทำหน้าที่เชื่อมเน็ตเวิร์คภายในกับอินเตอร์เน็ต

รูปแบบทั่วไปของ Access Lists

รูปแบบมาตรฐานทั่วไปของ Access Lists มีรูบแบบดังนี้

access-list [#] [permit|deny] [source-address] [keywork any] [source mask]

[] หมายถึงคำสั่งที่อยู่ในเครื่องหมายนี้จะมีหรือไม่มีก็ได้

Access Lists จะมีกี่บรรทัดก็ได้ การทำงานจะเริ่มตรวจสอบไปตั้งแต่บรรทัดที่ 1 เป็นลำดับไปเรื่อย ๆ (จากบนลงล่าง) การสร้าง access lists สามารถสร้างเป็นกลุ่มหรือแยกออกเป็นประเภทได้โดยการระบุด้วยหมายเลขของ access lists รูปแบบของ access lists แบ่งออกเป็น 2 ประเภทคือ access lists ที่ใช้งานทั่ว ๆ ไป (standard access lists) ซึ่งมีหมายเลขตั้งแต่ 1-99 และ access lists ที่สามารถสร้างเงื่อนไขได้ซับซ้อนมากขึ้นเรียกว่า (extended access lists) ซึ่งมีหมายเลขตั้งแต่ 100-199 การกำหนดค่าเน็ตเวิร์คและ subnet mask ให้กับ access lists จะต้องใช้วิธีการที่เรียกว่า "ไวด์คลาส (Wildcard Mask)" คือจะพิจารณาในบิตที่เป็นค่า 0 เท่านั้นค่าที่ปรากฏเป็น 1 จะไม่สนใจ ซึ่งแตกต่างกับ subnet mask แบบทั่ว ๆ ไปที่จะพิจารณาบิตที่เป็น 1 เท่านั้น ตัวอย่างเช่น กำหนดให้ไอพีคลาส C เป็น 192.168.1.1 subnet mask เป็น 255.255.255.0 แต่ถ้าใช้ไวด์คลาสจะต้องกำหนดเป็น 192.168.1.1 0.0.0.255 แทน

ทดสอบสร้าง access lists แบบง่าย ๆ

จากรูปแบบของ access lists

access-list [#] [permit|deny] [source-address] [keywork any] [source mask]

การสร้าง access lists ต้องทำในโกบอลคอนฟิกโหมด ในที่นี้จะใช้ access lists ที่อยู่ใน startard คือมีหมายเลข 1-99 ให้ทดลองสร้าง access lists ที่มีหมายเลข 1 และอนุญาตให้ไอพีแอดเดรส 1.1.1.1 ผ่านไปได้ส่วนไอพีอื่น ๆ นอกจากนี้จะไม่ยอมให้ผ่านไป ซึ่งจะมีรูปแบบดังนี้

Router(config)#access-list 1 permit 1.1.1.1

ถ้าไม่มีการระบุหมายเลข subnet mask เราเตอร์จะเข้าใจว่าเป็นการใช้ subnet mask เป็น 0.0.0.0 โดยดีฟอลท์ ไม่มีความจำเป็นที่จะต้องระบุให้ทุก ๆ ไอพีนอกจาก 1.1.1.1 เป็น deny ที่บรรทัดสุดท้านเพราะเราเตอร์จะรู้ทันทีว่านอกจากไอพี 1.1.1.1 แล้วจะไม่ยอมให้ทั้งหมด

การกำหนด ACL ไปยังอินเตอร์เฟส

เมื่อต้องการให้ access lists ทำงานจะต้องทำการกำหนดเข้าไปยังอินเตอร์เฟสก่อนจึงจะสามารถทำงานได้ ซึ่งมีรูปแบบดังนี้

ip access-group [access-list-number] [in|out]

in คือ เราเตอร์จะรับ packet เข้ามาและทำการตรวจสอบตามกฎที่ตั้งไว้

out คือ เราเตอร์จะส่ง packet ออกไปโดยทำการตรวจสอบกับกฎที่ได้ตั้งไว้

ตัวอย่างเช่นต้องการกำหนดให้อินเตอร์เฟสซีเรียล 0 ใช้งาน access list 1 สามารถทำได้ดังนี้

Router(config)#interface serial 0

Router(config-if)#ip access-group 1 in

Router(config-if)#ip access-group 1 out

การตรวจสอบ Access lists

ในบางครั้งการกำหนด access lists บนเราเตอร์อาจคาบเกี่ยวหรือ overlap กันได้ ดังนั้นเราเตอร์จะต้องมีวิธีการตรวจสอบว่าลำดับความสำคัญว่า access lists อันไหนที่สำคัญที่สุดซึ่งมีกฎดังนี้

1. ลำดับแรกจะพิจารณาในระดับของโฮสต์ก่อน เช่น

   access-list 101 permit 192.168.1.1 0.0.0.0 จะอนุญาตให้เฉพาะไอพี 192.168.1.1 ผ่านเท่านั้น

2. ลำดับต่อไปจะพิจารณาในระดับของ subnet เช่น

   access-list 101 permit 192.168.1.1 0.0.0.255 จะอนุญาตให้ทุก ๆ เครื่องในเน็ตเวิร์ค 192.168.1.0 ผ่านได้ทั้งหมด ส่วนเน็ตเวิร์คอื่นๆ ไม่สามารถผ่านไปได้

3. ลำดับต่อไปจะพิจารณาในระดับช่วงของไอพี เช่น เมื่อต้องการให้ช่วงของไอพีตั้งแต่ 10.3.16.0 ถึง 10.3.31.255 ผ่านได้ต้องทำดังนี้

   นำหมายเลขของไอพีต้นลบไอพีสุดท้ายของช่วง

   10.3.31.255 –

   <u>10.3.16.0</u>

   <u>0.0.15.255</u>

   access-list 101 permit 10.3.16.0 0.0.25.255

4. ลำดับสุดท้ายจะพิจารณาในส่วนที่เป็น any เช่น

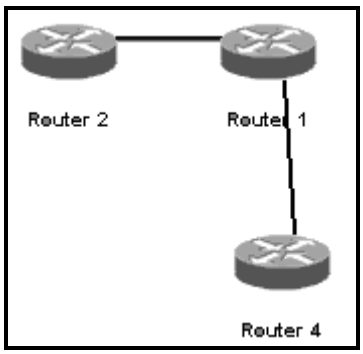
   access-list 101 permit any

   access-list 101 ip permit any any

จุดมุ่งหมาย : เข้าใจรูปแบบการคอนฟิก access lists เบื้องต้น

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 3 ตัวคือ เราเตอร์ 1, 2, 4 ดังรูปที่ 7.1 และตารางที่ 7.1

การสร้าง Network Map :



รูปที่ 7.1 ผังเน็ตเวิร์คสำหรับคอนฟิก Access Lists

ตารางที่ 7.1 รายละเอียดของแต่ละอินเตอร์เฟสสำหรับทดลอง Access Lists

	Router1	Router2	Router4
Interface Ethernet 0	24.17.2.1 255.255.255.240	24.17.2.2 255.255.255.240	
Interface Serial 0	24.17.2.17 255.255.255.240		24.17.2.18 255.255.255.240
โพลโทคอล RIP	ให้ทำงานบน interface serial 0 และ ethernet 0	ให้ทำงานที่ ethernet 0	ให้ทำงานที่ serial 0

Simulator :

1. บนเราเตอร์ 1 ให้เปลี่ยนชื่อเป็น Router1 และกำหนดให้อินเตอร์เฟสอีเทอร์เน็ต 0 มีหมายเลขไอพีแอดเดรสเป็น 24.17.2.1 netmask เป็น 255.255.255.240 (ใช้งานได้ทั้งหมด 16-2 = 14 เครื่อง) และกำหนดให้อินเตอร์เฟสซีเรียล 0 มีหมาลเลขไอพีเป็น 24.17.2.17 submask เป็น 255.255.255.240 พร้อมกับสั่งให้ทำงาน

   Router>*enable*

   Router(config)#*hostname Router1*

   Router1(config)#

   Router1(config)#*interface ethernet 0*

   Router1(config-if)#*ip address 24.17.2.1 255.255.255.240*

   Router1(config-if)#*no shutdown*

   Router1(config-if)#*exit*

   Router1(config)#*interface serial 0*

   Router1(config-if)#*ip address 24.17.2.17 255.255.255.240*

   Router1(config-if)#*no shutdown*

2. ขั้นต่อไปให้เข้าไปที่เราเตอร์ 2 กำหนดชื่อเป็น Router2 ที่อินเตอร์เฟสอีเทอร์เน็ต 0 กำหนดไอพีเป็น 24.17.2.2 255.255.255.240 และกำหนดให้ทำงาน

   Router>*enable*

   Router(config)#*hostname Router2*

   Router2(config)#

   Router2(config)#*interface ethernet 0*

   Router2(config-if)#*ip address 24.17.2.2 255.255.255.240*

   Router2(config-if)#*no shutdown*

   Router2(config-if)#*end*

   Router2#*ping 24.17.2.1* [ทดสอบ ping ไปยัง Router1 อินเตอร์เฟสอีเทอร์เน็ต 0]

3. ขั้นต่อไปให้กเข้าไปคอนฟิกที่เราเตอร์ 4 ให้เปลียนชื่อเป็น Router4 กำหนดให้อินเตอร์เฟสซีเรียล 0 มีไอพีแอดเดรสเป็น 24.17.2.18 255.255.255.240 และสั่งให้ทำงาน พร้อมกับทดสอบ ping ไปยัง Router1 ว่าการเชื่อมต่อสมบูรณ์หรือไม่

   Router>*enable*

   Router(config)#*hostname Router4*

   Router4(config)#

   Router4(config)#*interface serial 0*

   Router4(config-if)#*ip address 24.17.2.18 255.255.255.240*

   Router4(config-if)#*no shutdown*

   Router4(config-if)#*end*

   Router4#*ping 24.17.2.17* [ทดสอบ ping ไปยัง Router1 อินเตอร์เฟสซีเรียล 0]

4. ถึงขั้นตอนนี้อินเตอร์เฟสที่เชื่อมต่อกันทั้งหมดต้องสามารถเชื่อมต่อกันได้แล้ว ขั้นต่อไปให้อีนาเบิลเร้าติ้งโพรโทคอล RIP เพื่อทำให้เน็ตเวิร์คระหว่างเราเตอร์ 2 และเราเตอร์ 4 สามารถส่งข้อมูลถึงกันได้

   บน Router1

   Router1(config-if)#*exit*

   Router1(config)#*router rip*

   Router1(config-router)#*network 24.0.0.0*

   Router1(config-router)#*end*

5. บน Router2 ให้อีนาเบิล RIP และประกาศเน็ตเวิร์คที่เชื่อมต่อกับอินเตอร์เฟส ethernet 0

   Router2#*config term*

   Router2(config)#*router rip*

   Router2(config-router)#*network 24.0.0.0*

Router2(config-router)#*end*

6. บน Router4 ให้อีนาเบิล RIP และประกาศเน็ตเวิร์คที่เชื่อมต่อกับอินเตอร์เฟส serial 0

   Router4#*config term*

   Router4(config)#*router rip*

   Router4(config-router)#*network 24.0.0.0*

   Router4(config-router)#*end*

7. ถึงขั้นตอนนี้ เราได้อีนาเบิลเร้าติ้งโพรโทคอลเรียบร้อยแล้ว เราควรจะสามารถสร้างการเชื่อมต่อระหว่างเน็ตเวิร์คของ Router2 และ Router4 ได้ ด้วยการทดสอบ ping จาก Router4 ไปยังอินเตอร์เฟสของ Router1 ดังนี้

   Router4#*ping 24.17.2.2* ดังรูปที่ 7.2

```
Router4#ping 24.17.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 24.17.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 7.2 ทดสอบการเชื่อมต่อของเราเตอร์ 2 กับเราเตอร์ 4

8. ทดลองสร้าง access lists โดยไม่อนุญาตให้ Router4 ทำการ ping หรือไม่ให้มีการเชื่อมต่อมายัง Router2

   บน Router2

   Router2#*config term*

   Router2(config)#

9. ให้สร้าง access lists แบบ standard หมายเลข 1 โดยไม่อนุญาตให้ Router4 ไอพีแอดเดรส 24.17.2.18 เข้ามาเพียงไอพีเดียวส่วนไอพีอื่น ๆ สามารถเข้ามาได้

   Router2(config)#*access-list 1 deny 24.17.2.18 0.0.0.0* **หรือ**

   Router2(config)#*access-list 1 deny host 24.17.2.18* **หรือ**

   Router2(config)#*access-list 1 deny 24.17.2.18* [ไม่อนุญาตให้ไอพี 24.17.2.18 ผ่าน]

   Router2(config)#*access-list permit any* [อนุญาตให้ทุกไอพีผ่านไปได้]

การทำงานของเราเตอร์จะเริ่มประมวลผลคำสั่งที่บรรทัดแรกก่อนคือไม่ยอมให้ไอพีหมายเลข 24.17.2.18 ผ่าน จากนั้นบรรทัดที่ 2 จะยอมให้ทุกไอพีผ่านได้ สรุปจากทั้ง 2 บรรทัดคือยอมให้ทุกไอพีผ่านได้ยกเว้นไอพี 24.17.2.18

10. หลักจากกำหนด access lists แล้วให้กำหนดไปยังอินเตอร์เฟสที่ต้องการใช้งาน ในที่นี้เราใช้อีเทอร์เน็ต 0 บนเราเตอร์ 2

    Router2(config)#*interface ethernet 0*

Router2(config-if)#*ip access-group 1 in* [ให้อินเตอร์เฟสนี้ใช้ policy หมายเลข 1]

Router2(config-if)#*exit*

in หมายถึง packet ที่มาจากภายนอกเครือข่ายและเข้าสู่เราเตอร์ปัจจุบัน

out หมายถึง packet ที่มาจากเราเตอร์ปัจจุบันเพื่อออกไปสู่เครือข่ายภายนอก

ตรวจสอบการทำงานของ Access Lists

จุดมุ่งหมาย : ตรวจสอบการทำงานของ access lists ว่าทำงานถูกต้องหรือไม่

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์และคอนฟิกกูเรชันที่ได้จาก รูปที่ 7.1

การสร้าง Network Map : ใช้ผังเน็ตเวิร์คจาก รูปที่ 7.1

Simulator :

1. เข้าไปที่ Router4 จาก LAB ที่ผ่านมาเมื่อยังไม่มีการคอนฟิก access lists เราจะสามารถ ping จาก Router4 ไปยัง Router2 ได้ แต่ตอนนี้เราได้กำหนด access lists ให้กับอินเตอร์เฟสอีเทอร์เน็ต 0 ของ Router2 ซึ่งไม่ยอมให้ packet ที่มาจาก Router4 ผ่านได้ ทดลอง ping จาก Router4 ไปยัง Router2 อีกครั้ง

   Router4#*ping 24.17.2.2* ดังรูปที่ 7.3

```
Router4#ping 24.17.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 24.17.2.2, timeout is 2 seconds:
[redacted]
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 7.3 ทดสอบการ ping หลังจากมีการกำหนด access lists

จากรูปจะถ้าเห็นสัญลักษณ์ U แสดงว่าการ ping ไม่สำเร็จเนื่องจากเรากำหนด access lists ไว้

2. บน Router2 ให้ทดลองแสดงคอนฟิกกูเรชันที่ทำงาน ด้วยคำสั่ง show running-config ดังรูปที่ 7.4

```
interface Ethernet0
 ip address 24.17.2.2 255.255.255.240
 no ip directed-broadcast
 bandwidth 10000
 [redacted]
!
router rip
 network 24.0.0.0
!
ip classless
no ip http server
[redacted]
```

รูปที่ 7.4 แสดง access lists ของเราเตอร์ 2

3. เราสามารถแสดง access lists ที่กำหนดให้แต่ละอินเตอร์เฟสด้วยคำสั่ง show ip interface เป็นคำสั่งที่ใช้แสดงข้อมูลเฉพาะอินเตอร์เฟสที่มีการระบุหมายเลขของไอพีแอดเดรสไว้เรียบร้อยแล้ว ดังรูปที่ 7.5

```
Router2#sh ip interface
Ethernet0 is up, line protocol is up
  Internet address is 24.17.2.2/28
  Broadcast address is 255.255.255.240
  MTU 1500 bytes,
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  [redacted]
  Proxy ARP Is Enabled
```

รูปที่ 7.5 แสดงคำสั่ง show ip interface

4. คำสั่ง show access-lists จะช่วยให้เราสามารถดูได้ว่าตอนนี้เราเตอร์ของเราสร้าง access lists อะไรไว้บ้าง รวมถึงบรรทัดไหนถูกใช้งานบ้าง และมีเปอร์เซ็นการใช้งานเป็นอย่างไรบ้าง ดังรูปที่ 7.6

```
Router2#sh access-lists
Standard IP access list 1
    1 deny   host 24.17.2.18 (5 matches)
    1 permit any (465 matches)
```

รูปที่ 7.6 แสดงการใช้คำสั่ง show access-lists

หมายเหตุ : เมื่อต้องการยกเลิก acess lists ให้เข้าไปยังโกลบอลคอนฟิกแล้วใช้คำสั่ง ***no access-list 1*** (ในกรณีที่ต้องการ access lists ทั้งหมด) แต่ถ้าต้องการยกเลิกบางอินเตอร์เฟส ให้เข้าไปที่อินเตอร์เฟสที่ต้องการยกเลิกแล้วใช้คำสั่ง ***no ip access-group 1***

Extended Access Lists

จุดมุ่งหมาย : เรียนรู้การใช้งาน Extended Access Lists

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์และคอนฟิกกูเรชันที่ได้จาก LAB ที่ผ่านมา

การสร้าง Network Map : ใช้ผังเน็ตเวิร์คจาก LAB ที่ผ่านมา กำหนดค่าดังตารางที่ 7.2

ตารางที่ 7.2 รายละเอียดของแต่ละอินเตอร์เฟสสำหรับทดลอง Extended Access Lists

	Router1	Router2	Router4
Interface Ethernet 0	24.17.2.1 255.255.255.240	24.17.2.2 255.255.255.240	
Interface Serial 0	24.17.2.17 255.255.255.240		24.17.2.18 255.255.255.240
โพลโทคอล RIP	ให้ทำงานบน interface serial 0 และ ethernet 0	ให้ทำงานที่ ethernet 0	ให้ทำงานที่ serial 0

Simulator :

1. จากคอนฟิกกูเรชันใน LAB ที่ 28 ให้ใช้คำสั่ง no ip access-group 1 ที่อินเตอร์เฟสอิเทอร์เน็ตบนเราเตอร์ 2 (ถ้ายังไม่มีการคอนฟิกให้กลับไปทำตาม LAB ที่ 28 ก่อน)

   Router2#*config terminal*

   Router2(config)#*interface ethernet 0*

   Router2(config-if)#*no ip access-group 1*

2. บน Router1 จะใช้ Extended Access Lists สำหรับกำหนดให้เน็ตเวิร์ควง 24.17.2.16 สามารถใช้คำสั่ง Telnet เข้ามาใช้งาน Router1 ได้เท่านั้น ในที่นี้เน็ตเวิร์คที่เชื่อมต่อกับ Router4 จะสามารถ Telnet เข้ามาใช้งานที่ Router1 ได้เพียงเน็ตเวิร์คเดียวเท่านั้น บน Router1 ให้สร้าง Extended Access Lists หมายเลข 101

   Router1#*config terminal*

   Router1(config)#*access-list 101 permit tcp 24.17.2.16 0.0.0.15 any eq telnet log*[อนุญาตให้เน็ตเวิร์ควง 24.17.216 สามารถเข้ามายัง Router1 ผ่านโปรแกรม Telnet ได้]

3. ขั้นตอนต่อไปเราจะสร้าง Access Lists ที่อนุญาตให้เน็ตเวิร์ควง 24.17.2.0 เข้ามาใช้งาน Router1 ได้ทุก ๆ อย่าง โดยใช้ ACL หมายเลข 102

   Router1(config)#*access-list1 102 permit ip 24.17.2.0 0.0.0.15 any log*

คำสั่ง log จะเป็นคำสั่งที่ใช้บันทึกการใช้งานเมื่อตรงกับกฎข้อนี้

4. ให้กำหนด Extended ACL ที่สร้างไว้ไปยังอินเตอร์เฟสที่ต้องการใช้งาน หมายเลข 101 สำหรับอินเตอร์เฟสซีเรียล 0 และหมายเลข 102 สำหรับอินเตอร์เฟสอีเทอร์เน็ต 0 มีชนิดเป็น in

   Router1(config)#*interface serial 0*

   Router1(config-if)#*ip access-group 101 in*

   Router1(config-if)#*exit*

   Router1(config)#*interface ethernet 0*

   Router1(config-if)#*ip access-group 102 in*

ตรวจสอบการทำงานของ Extended Access Lists

จุดมุ่งหมาย : ตรวจสอบการทำงานของ Extended Access Lists

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์และคอนฟิกกูเรชันที่ได้จาก LAB ที่ผ่านมา

การสร้าง Network Map : ใช้ผังเน็ตเวิร์คจาก LAB ที่ผ่านมา

Simulator :

1. เราจะใช้คอนฟิกใน LAB ที่ผ่านมาสำหรับใช้งาน โดยทดลองตรวจสอบว่า Extended ACL ที่เรากำหนดไปนั้นสามารถทำงานได้อย่างถูกต้องหรือยัง โดยเข้าไปที่ Router4 แล้วทดสอบ ping ไปยังเน็ตเวิร์ค 24.17.2.16 ที่ขาอินเตอร์เฟสซีเรียล 0 ของ Router1

   Router4#*ping 24.17.2.17* ดังรูปที่ 7.7

```
Router4#ping 24.17.2.17

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 24.17.2.17, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 7.7 ทดสอบ EACL ด้วยการใช้คำสั่ง ping

2. ต่อไปเราจะทำการทดสอบการ Telnet ไปที่ Router1 แต่ก่อนอื่นจำเป็นจะต้องเซ็ตค่าเพื่อให้ใช้งาน Telnet ที่ Router1 ได้ก่อน

   Router1#*config terminal*

   Router1(config)#*line vty 0 4* [กำหนดการใช้งาน Telnet โดยสามารถใช้งานได้มากที่สุด 4 session พร้อม ๆ กัน]

   Router1(config-line)#*login* [ระบุว่าต้องมีการ login ก่อน]

   Router1(config-line)#*password boson* [กำหนดรหัสผ่านเป็น boson]

   Router1(config-line)#*exit*

3. ขั้นต่อไปให้ลอง Telnet จาก Router4 ไปยัง Router1

   Router4#*telnet 24.17.2.17*

จะปรากฏหน้าต่างให้ใส่รหัสผ่าน ให้ทดลองใส่รหัสผ่านเป็น boson ถ้าต้องการเปลี่ยนกลับมาใช้งาน Router4 อีกครั้งขณะใช้งาน Telnet อยู่ให้กดปุ่ม Ctrl + Shift + 6 + x พร้อมกัน ก็จะสามารถเปลี่ยนกลับไปกลับมาระหว่างที่ทำการ Telnet อยู่ หรือใช้ exit เพื่อยกเลิกการติดต่อ

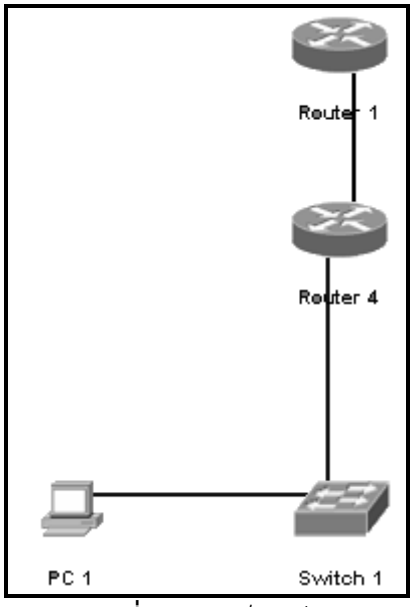
4. ทดลองใช้คำสั่ง show running-config, show ip interface, show access-lists ว่าคอนฟิกเป็นอย่างไรบ้าง

Name Access Control Lists

จุดมุ่งหมาย : เรียนรู้การสร้าง ACL แบบใช้ชื่อแทนการใช้หมายเลข

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัว (Router1,4) สวิตช์ 1 ตัว และ PC 1 เครื่อง

การสร้าง Network Map : ดังรูปที่ 7.8, ตารางที่ 7.3



รูปที่ 7.8 ผังเน็ตเวิร์ค

ตารางที่ 7.3 ตารางรายละเอียดสำหรับคอนฟิก

อุปกรณ์	Router 1	Router 2	PC1
ชื่อของอุปกรณ์	Router1	Router2	C:>
Ethernet 0/0		192.168.1.17/28	192.168.1.18/28
Serial 0	192.168.1.1/28	192.168.1.2/28	
Default Gateway			192.168.1.17/28

Simulator :

1. คอนฟิกรายละเอียดต่าง ๆ ตามตารางพร้อมกับอีนาเบิลโพรโทคอล RIP

บนเราเตอร์ 1 ให้เปลี่ยนชื่อเป็น Router1 กำหนดไอพีเป็น 192.168.1.1 255.255.255.240

ที่ อินเตอร์เฟสซีเรียล 0 และไอพี 192.168.1.17 255.255.255.240 สำหรับอีเทอร์เน็ต 0

Router>*enable*

Router#*config terminal*

Router(config)#*hostname Router1*

Router1(config)#*interface serial 0*

Router1(config-if)#*ip address 192.168.1.1 255.255.255.240*

Router1(config-if)#*no shutdown*

Router1(config-if)#*exit*

Router1(config)#*interface ethernet 0*

Router1(config-if)#*ip address 192.168.1.17 255.255.255.240*

Router1(config-if)#*no shutdown*

2. บนเราเตอร์ 4 ให้เปลี่ยนชื่อเป็น Router4 กำหนดไอพีเป็น 192.168.1.2 255.255.255.240 ที่อินเตอร์เฟสซีเรียล 0

   Router>*enable*

   Router#*config terminal*

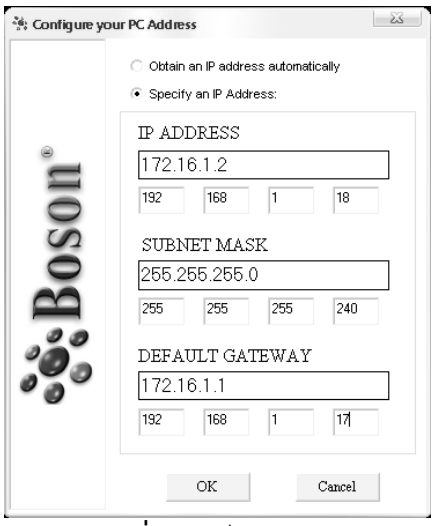
   Router(config)#*hostname Router4*

   Router4(config)#*interface serial 0*

   Router4(config-if)#*ip address 192.168.1.2 255.255.255.240*

   Router4(config-if)#*no shutdown*

3. ขั้นต่อไปให้ทำการกำหนดหมายเลขไอพีให้กับเครื่อง PC1 เป็น 192.168.1.18 255.255.255.240 เกตเวย์คือ 192.168.1.17 โดยการใช้คำสั่ง winipcfg ดังรูปที่ 7.9



รูปที่ 7.9 เซ็ตค่า PC1

4. ขั้นต่อไปให้อีนาเบิลโพรโทคอล RIP ให้เราเตอร์ทุกตัว

   Router1(config-if)#*exit*

   Router1(config)#*router rip*

   Router1(config-router)#*network 192.168.1.0*

   Router4(config-if)#*exit*

   Router4(config)#*router rip*

   Router1(config-router)#*network 192.168.1.0*

ทดสอบ ping จาก PC1 ไปยัง Router1 ดังรูปที่ 7.10

```
C:>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=60ms TTL=241
Reply from 192.168.1.1: bytes=32 time=60ms TTL=241
Reply from 192.168.1.1: bytes=32 time=60ms TTL=241
Reply from 192.168.1.1: bytes=32 time=60ms TTL=241
Reply from 192.168.1.1: bytes=32 time=60ms TTL=241
```

รูปที่ 7.10 ทดสอบการ ping จาก PC1 ไปยัง Router1

5. ทดลองสร้าง ACL ที่เป็นชื่อแทนหมายเลขโดยไม่อนุญาตให้ PC1 ping ไปยัง Router1 ได้ บน Router1 ให้สร้าง ACL ชื่อ deny_ping

   Router1(config-router)#*exit*

   Router1(config)#*ip access-list extended deny_ping*

   Router1(config-ext-acl)#*deny icmp host 192.168.1.18 192.168.1.1 0.0.0.0 log*

   deny = ไม่อนุญาต

   icmp = โพรโทคอล icmp ถูกใช้โดยคำสั่ง ping

   host 192.168.1.18 = โฮสต์ที่ไม่ต้องการให้ผ่านเข้ามาใช้งาน

   192.168.1.1 = กำหนดหมายเลขไอพีที่ packet จะเข้ามา

   0.0.0.0 = หมายเลข wildcard ของ 192.168.1.1 ในที่นี้จะใช้เพียงไอพีเดียว

   Router1(config-ext-acl)#*permit ip any any log* [กำหนดให้ไอพีอื่นๆ ผ่านไปได้]

6. ขั้นต่อไปให้ใช้ ACL ที่ได้ในข้อที่ 5 กำหนดไปยังซีเรียล 0 บน Router1 ใน packet ขาเข้า

   Router1(config-ext-acl)#*exit*

   Router1(config)#*interface serial 0*

   Router1(config)#*ip access-group deny_ping in*

7. ขั้นตอนนี้ให้ลองทดสอบ ping จาก PC1 ไปยัง Router1 ไอพี 192.168.1.1 ว่าได้ผลเป็นอย่างไร ต่อจากนั้นให้เข้าไปที่ Router4 แล้วลองทดสอบ ping ไปยัง Router1 อีกครั้ง แล้วสังเกตว่าผลเป็นอย่างไร

   ผลลัพธ์ที่ถูกต้องคือ จะไม่สามารถ ping จาก PC1 ไปยัง Router1 ได้ แต่สามารถ ping จาก Router4 ไปยัง Router1 ได้

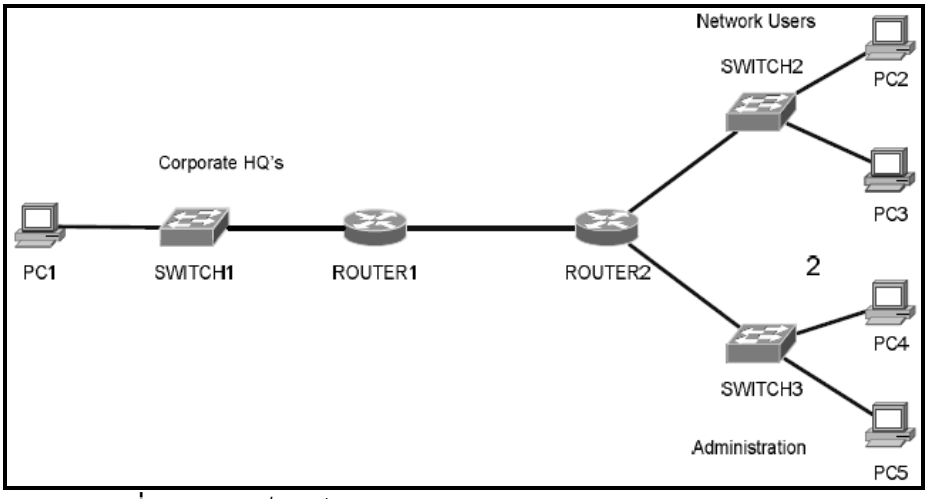
8. เมื่อกลับไปดูที่ Router1 ควรจะปรากฏข้อความที่เกิดจาก log เป็น 2 ประเภทคือ ข้อความที่ปฏิเสธ packet ที่มาจาก PC1 และข้อความที่ยอมรับทุก ๆ ไอพีที่ไม่ใช่ 192.168.1.18

Advance Extended Access Lists

จุดมุ่งหมาย : เรียนรู้การคอนฟิก ACL ที่จัดการเกี่ยวกับความคับคั่งของข้อมูลจากเน็ตเวิร์คหนึ่งไปยังอีกเน็ตเวิร์คหนึ่ง, จากโฮสต์ไปสู่โฮสต์, และจากเน็ตเวิร์คไปยังโฮสต์

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัว (Router1,4) สวิตช์ 3 ตัว และ PC 5 เครื่อง

การสร้าง Network Map : ดังรูปที่ 7.11, ตารางที่ 7.4, ตารางที่ 7.5



รูปที่ 7.11 ผังเน็ตเวิร์คสำหรับ Advance Extended Access Lists

ตารางที่ 7.4 รายละเอียดการเชื่อมต่อของเราเตอร์

อุปกรณ์	Router 1	Router 2
ชื่ออุปกรณ์	Router1	Router2
อินเตอร์เฟส FA0/0	192.168.3.1/24	192.168.1.129/25
อินเตอร์เฟส FA0/1		192.168.1.1/25
อินเตอร์เฟส serial 0	192.168.2.1/24	192.168.2.2/24

ตารางที่ 7.5 รายละเอียดการเชื่อมต่อของ PC

โฮสต์	**ไอพีแอดเดรส**	**Subnet mask**	**Default gateway**
PC1	192.168.3.2	255.255.255.0	192.168.3.1
PC2	192.168.1.130	255.255.255.128	192.168.1.129
PC3	192.168.1.131	255.255.255.128	192.168.1.129
PC4	192.168.1.2	255.255.255.128	192.168.1.1
PC5	192.168.1.3	255.255.255.128	192.168.1.1

Simulator :

1. ให้ทำการสร้างผังเน็ตเวิร์คและคอนฟิกค่าของไอพีแอดเดรสตามตารางที่กำหนดให้
2. อีนาเบิลโพรโทคอล RIP บนเราเตอร์ทุก ๆ ตัวให้ครบแล้วทดสอบการเชื่อมต่อด้วยคำสั่ง ping

การสร้าง ACL เพื่อจัดการ Traffic ที่เกิดจาก Network to Network

3. เมื่อพิจารณาผังเน็ตเวิร์คใน LAB นี้ สิ่งแรกที่ต้องทำคือการอนุญาตให้ traffic ที่เกิดขึ้นของแต่ละเน็ตเวิร์คสามารถสื่อสารกันได้ เมื่อพิจารณาที่เราเตอร์ 1 ที่อินเตอร์เฟสซีเรียล 0 จะมีเน็ตเวิร์คที่จะเชื่อมต่อด้วย 2 เน็ตเวิร์คคือ เน็ตเวิร์ค 192.168.1.0, 192.168.2.0 การสร้าง ACL สามารถทำได้ดังนี้

access-list 100 permit ip 192.168.1.0 0.0.0.127 192.168.3.0 0.0.0.255 log

อนุญาตให้เน็ตเวิร์ค 192.168.1.0 (128 ไอพี) ผ่านไปยังเน็ตเวิร์ค 192.168.3.0 ได้เท่านั้นไม่สามารถผ่านเข้าไปที่เน็ตเวิร์คอื่น ๆ ของเราเตอร์ 1 ได้

access-list 100 permit ip 192.168.2.0 0.0.0.0 any

อนุญาตให้เน็ตเวิร์ค 192.168.2.0 ผ่านไปยังทุก ๆ เน็ตเวิร์คที่เชื่อมกับ Router1 อยู่ได้ ส่วนไอพีอื่น ๆ จะถูก drop หมดเพราะเราเตอร์จะทำ Implicit deny โดยดีฟอลท์ที่บรรทัดสุดท้ายโดยอัตโนมัติ

4. กำหนด access lists หมายเลข 100 ให้กับอินเตอร์เฟสซีเรียล 0 ของ Router1

Router1(config)#*interface serial 0*

Router1(config-if)#*ip access-group 100 in*

5. ทดลอง ping ทดสอบจาก PC2, 3, 4, 5 ไปยัง PC1 ซึ่งผลที่ได้ PC2, 3 ไม่ควรที่จะ ping ได้ แต่ PC4, 5 ต้องสามารถ ping ได้

การสร้าง ACL เพื่อจัดการ Traffic ที่เกิดจาก Host to Host

6. สมมุติว่าเราไม่ต้องการให้ PC2 (ไอพี 192.168.1.130) เข้าไปใช้งานเครื่อง PC5 (ไอพี 192.168.1.3) ได้ เราจะต้องสร้าง Access lists ที่ Router2 แล้วกำหนดลงไปยังอินเตอร์เฟส FA0/0 ที่เชื่อมต่ออยู่กับ PC2 เราสามารถเขียน access lists ได้ดังนี้

access-list 101 deny ip host 192.168.1.130 192.168.1.3 0.0.0.0 log

ไม่อนุญาตให้โฮสต์ 192.168.1.130 ผ่านไปยังโฮสต์ 192.168.1.3 ได้

access-list 101 permit ip any any [นอกนั้นผ่านได้หมด]

7. กำหนด access list 101 ไปยังอินเตอร์เฟส FA0/0 ของเราเตอร์ 2

Router2(config)#*interface FastEthernet 0/0*

Router2(config-if)#*ip access-group 101 in*

ทดลอง ping จาก PC2 ไปยัง PC5 และจาก PC3 ไปยัง PC5 ผลที่ได้ PC2 จะไม่สามารถ ping PC5 ได้ แต่ PC3 ต้องสามารถ ping ได้

การสร้าง ACL เพื่อจัดการ Traffic ที่เกิดจาก Network to Host

8. ก่อนการคอนฟิกจำเป็นที่จะต้องนำเอา access lists หมายเลข 100 และ 101 ออกจากอินเตอร์เฟสของแต่ละเราเตอร์เสียก่อน โดยใช้คำสั่ง no

Router1(config)#*interface serial 0*

Router1(config-if)#*no ip access-group 100 in*

Router2(config)#*interface FastEthernet 0/0*

Router2(config-if)#*no ip access-group 101 in*

9. เราจะทำการ drop ทุก packet ที่ต้องการเข้ามายัง PC1 จาก Network Users ซึ่งมีสมาชิก คือ PC2 และ PC3 สามารถเขียน access lists ได้ดังนี้

   access-list 102 deny ip 192.168.1.128 0.0.0.127 host 192.168.3.2 log

   access-list 102 permit ip any any

10. กำหนด access lists หมายเลข 102 เข้าไปยังอินเตอร์เฟสซีเรียล 0 ของ Router2 โดยใช้ out

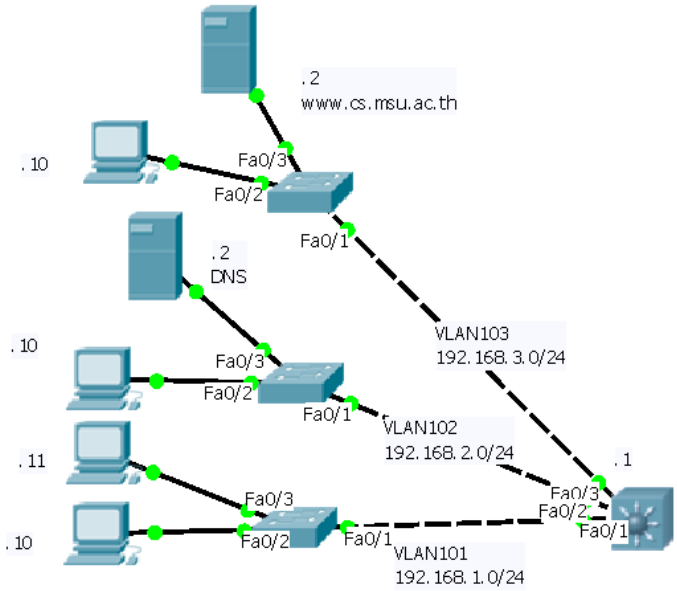
    Router2(config)#*interface serial 0*

    Router1(config-if)# *ip access-group 102 out*

ทดสอบโดยการ ping จาก PC2, 3 ไปยัง PC1 ซึ่งที่ถูกต้องแล้วไม่สมควรจะ ping ได้ สุดท้ายให้สังเกต log ที่เกิดขึ้นที่ Router2

แบบฝึกหัดท้ายบท

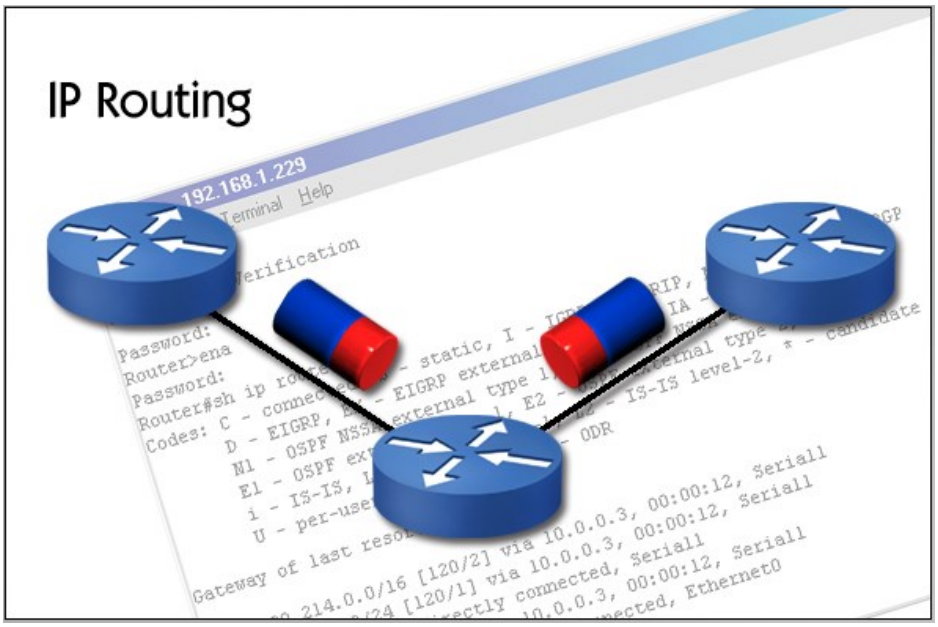
1. จาก network diagram ดังรูปที่ 1 ข้างล่าง ให้นิสิตทำ ACL ดังต่อไปนี้ พร้อมทดสอบการทำงาน



รูปที่ 1

- Block IP address 192.168.1.10 ไม่ให้ใช้งาน Telnet
- Block IP address 192.168.2.10 ไม่ให้ใช้งาน WWW
- Block IP address 192.168.3.10 ไม่ให้ทำการ ping Network 192.168.2.0/24
- Block IP address 192.168.1.11 ไม่ให้ทำการใช้งาน DNS ได้

บทที่ 8
การกำหนดเส้นทางด้วยไอพี (IP Routing)



- Routing concepts
- Static routes
- Configuring RIP
- Troubleshooting RIP
- Configuring IGRP
- Troubleshooting IGRP
- OSPF
- Routing protocol comparison

แนวคิด

ในบทนี้จะเน้นลงไปที่การคอนฟิกเราเตอร์เพื่อทำหน้าที่ในการค้นหาเส้นทาง และเลือกโพรโทคอลที่ใช้ในการค้นหาเส้นทางได้อย่างเหมาะสม

วัตถุประสงค์

1. เพื่อให้ทราบถึงโปรโทคอลที่ทำหน้าที่ค้นหาเส้นทางบนระบบเครือข่าย
2. เพื่อให้ทราบถึงโพรโทคอลที่เป็นพื้นฐานสำคัญๆบนระบบเครือข่าย

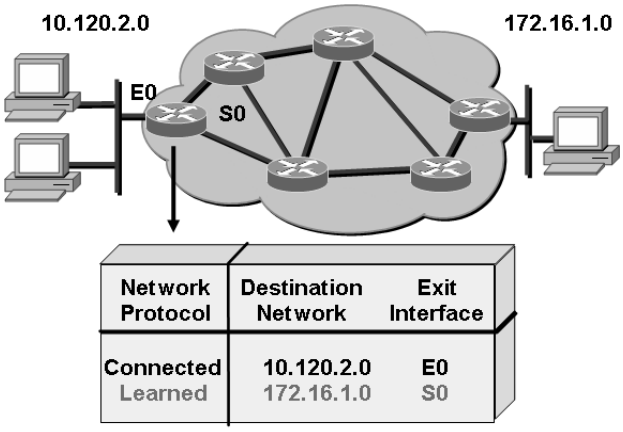
โพรโทคอลค้นหาเส้นทาง

ส่วนนี้จะกล่าวถึงโพลโตคอลที่ใช้สำหรับค้นหาเส้นทางของอุปกรณ์ในเลเยอร์ที่สามของ OSI Model ซึ่งอุปกรณ์ในเลเยอร์นี้มีความจำเป็นอย่างยิ่งที่จะต้องเลือกใช้โพรโทคอลตัวใดตัวหนึ่งในการหาเส้นทางเพื่อให้ข้อมูลถึงที่หมายปลายทางอย่างถูกต้อง

เนื้อหาที่จะกล่าวถึงนี้จะกล่าวโดยย่อ ๆ และเอาแต่ใจความสำคัญของแต่ละโพรโทคอลมาแสดงเท่านั้น เนื่องจากเนื้อหาของโพรโทคอลค้นหาเส้นทางนั้นมีมากและค่อนข้างที่จะทำความเข้าใจได้ยาก แต่ก็จะกล่าวไว้โดยย่อ ๆ เพื่อให้พอเข้าใจถึงแนวคิดและวิธีออกแบบเท่านั้น ซึ่งถ้าจะไม่กล่าวถึงเรื่องนี้เลยเนื้อหาก็จะดูไม่สมบูรณ์ และเหมือนกับขาดจุดสำคัญไป ที่สำคัญคือจะต้องนำความรู้เรื่องของการค้นหาเส้นไปประยุกต์ใช้กับการทำ Lab ด้วย

การค้นหาเส้นทางคืออะไร

การค้นหาเส้นทางคือกระบวนการหรือวิธีการที่ใช้เลือกเส้นทางสำหรับรับและส่งข้อมูลจากต้นทางไปยังไปทาง โดยเส้นทางที่ได้ไม่จำเป็นต้องมีเส้นทางเดียว



รูปที่ 8.1 แสดงเน็ตเวิร์คที่มีเส้นทางหลายทิศทาง

จากรูปที่ 8.1 เราเตอร์ที่ต่ออยู่กับเน็ตเวิร์ค 10.120.2.0 จะบันทึกข้อมูลในตารางเราติ้งของตนเองว่าเน็ตเวิร์ค 10.120.2.0 ต่ออยู่โดยตรงกับตัวมันที่อินเตอร์เฟส E0 ส่วนบรรทัดที่ 2 มันจะไม่ทราบว่าเน็ตเวิร์ควง 172.16.1.0 อยู่ไกลหรือซับซ้อนเพียงใด มันจะมีข้อมูลเพียงแค่ถ้าต้องการจะไปที่เน็ตเวิร์ค 172.16.1.0 มันจะส่งข้อมูลออกไปทางอินเตอร์เฟส S0 เท่านั้น ส่วนข้อมูลจะเดินทางไปอย่างไรนั้นเป็นเรื่องของกระบวนการหาเส้นทางอีกที

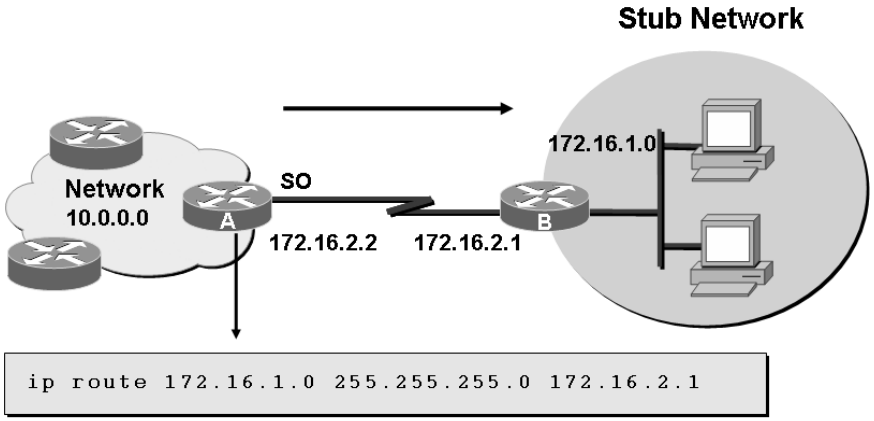
องค์ประกอบสำหรับค้นหาส้นทาง

- ที่อยู่ปลายทางที่จะรับและส่งข้อมูล
- แหล่งของข้อมูลที่มันสามารถเรียนรู้เส้นทางได้
- เส้นทางที่เป็นไปได้ทั้งหมด

- เส้นทางที่ดีที่สุด
- จัดการปรับปรุงและแก้ไขตาราเส้นทางของตัวมันเองให้ทันสมัยอยู่เสมอ

ประเภทของการค้นหาเส้นทางแบ่งออกเป็น 2 ประเภทคือ

- Static Route เป็นการระบุเส้นทางโดยผู้ดูแลระบบจะเป็นผู้กำหนดเองว่าจะให้ข้อมูลเดินทางไปเส้นทางไหน อย่างไร



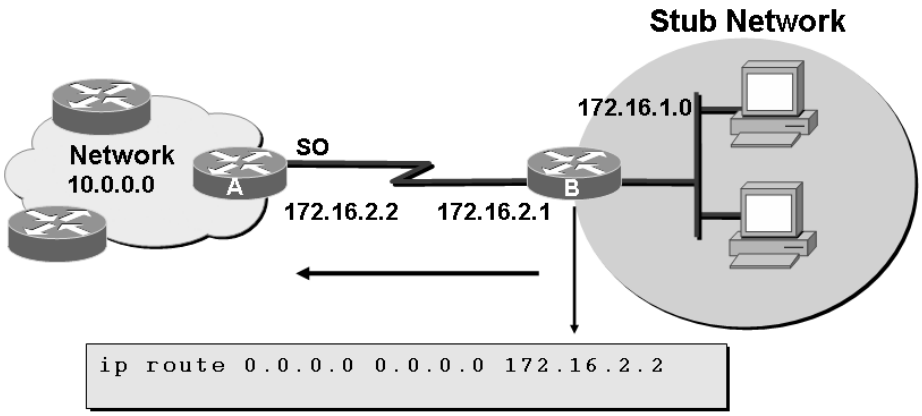
รูปที่ 8.2 ตัวอย่างการคอนฟิก Static Route

จากรูปที่ 8.2 ทำการคอนฟิกแบบ static route โดยกำหนดเส้นทางให้ข้อมูลเดินทางไปในเส้นทางเดียวเท่านั้น ที่เราเตอร์ A ต้องการส่งข้อมูลไปที่เราเตอร์ B จะให้ส่งไปที่เกตเวย์ที่มีหมายเลขไอพีแอดเดรสเบอร์ 172.16.2.1

- Dynamic Route ผู้ดูแลระบบจะเลือกโพลโตคอลใดโพรโทคอลหนึ่งขึ้นมาทำงาน โดยที่โพรโทคอลเหล่านั้นจะเป็นผู้ที่ดูแลเรื่องของกระบวนการหาเส้นทาง สถานะการเชื่อมต่อ การแลกเปลี่ยนข้อมูลกันระหว่างเราเตอร์ และการปรับปรุงตารางเราติ้งของตัวเอง ผู้ดูแลระบบไม่จำเป็นต้องเข้าไปจัดการเหมือนแบบ Static Route

Default Route

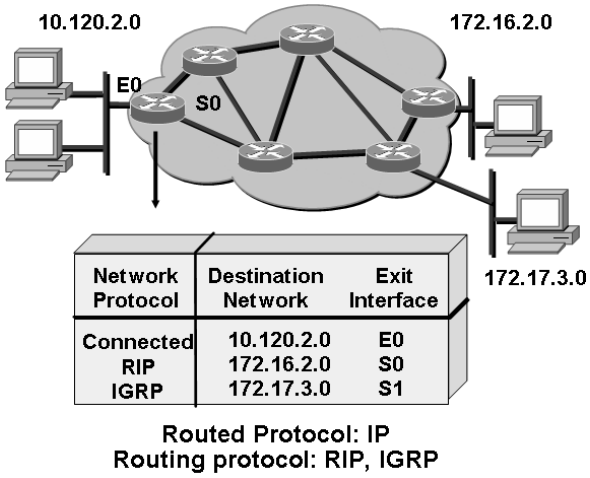
คือการอนุญาตให้ทุก ๆ เครื่องในเน็ตเวิร์คภายใน ออกไปสู่เน็ตเวิร์คภายนอกได้จากเส้นทางนี้ ซึ่งจำเป็นจะต้องมีการเซ็ตไว้เสมอ จากตัวอย่างรูปที่ 8.3 คำสั่ง ip route 0.0.0.0 0.0.0.0 172.16.2.2 จะทำให้ทุก ๆ เน็ตเวิร์คที่เชื่อมต่อหลังเราเตอร์ B จะส่งข้อมูลไปทางอินเตอร์เฟส S0 หมายเลขไอพีแอดเดรส 172.16.2.2



รูปที่ 8.3 แสดงการใช้งาน default route

โพรโทคอล**ค้นหาเส้นทางคืออะไร ?**

โพรโทคอลค้นหาเส้นทาง (Routing Protocol) คือขบวนการค้นหาเส้นทางที่ดีทีสุด และจะต้องทำงานกับข้อมูลที่จะถูก Route เสมอ เช่น โพรโทคอล IP (Routed) มันจะถูกใช้เพื่อกำหนดเส้นทางระหว่างเราเตอร์และคอยดูแลจัดการเรื่องตารางเราติ้งของตัวเอง



Network Protocol	Destination Network	Exit Interface
Connected	10.120.2.0	E0
RIP	172.16.2.0	S0
IGRP	172.17.3.0	S1

รูปที่ 8.4 แสดงตัวโพลโตคอลค้นหาเส้นทาง

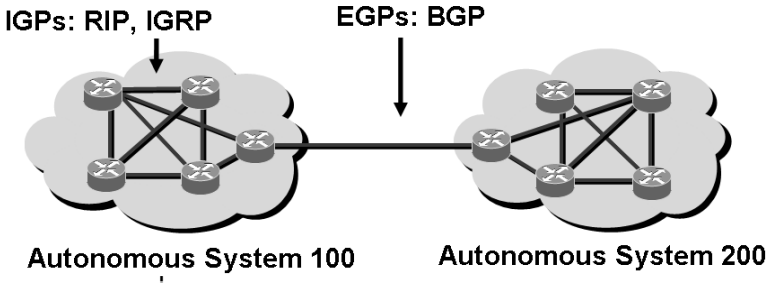
จากรูปที่ 8.4 เน็ตเวิร์ค 10.120.2.0 ที่ต่อกับ E0 ต้องการติดต่อไปยังเน็ตเวิร์ค 172.16.2.0 มันจะหาเส้นทางจากตารางเราติ้งซึ่งสามารถผ่านไปยังเน็ตเวิร์ค 172.16.2.0 ได้ 2 ทางคือทาง S0 ผ่านโพรโทคอลค้นหาเส้นทางแบบ RIP และอีกทางโดยผ่านทาง S1 ด้วยโพรโทคอล IGRP เป็นต้น

Autonomous System (AS)

Autonomous System คือการประกาศชื่อกลุ่มของเน็ตเวิร์คที่รวมตัวกันเข้าเป็นกลุ่มขนาดใหญ่ ประโยชน์ที่ต้องทำ AS คือการค้นหาเส้นทางที่ดีที่สุดและลดปริมาณของเราติ้งเทเบิลลง ตัวอย่างเช่น กลุ่มของสถานศึกษามีจำนวนสมาชิก 10 แห่ง กำหนดให้ AS เป็น 100 เมื่อมีการส่งข้อมูลมายังสถานศึกษาใด ๆ ในกลุ่ม ข้อมูลในตารางเราติ้งจะเก็บข้อมูลไว้เพียงบรรทัดเดียวคือ

สถานศึกษาหมายเลข 1-10 ส่งไปที่ AS 100 เป็นต้น AS จะมี 2 แบบคือ Interior (IGPs) และ Exterior (EGPs) ดังรูปที่ 8.5

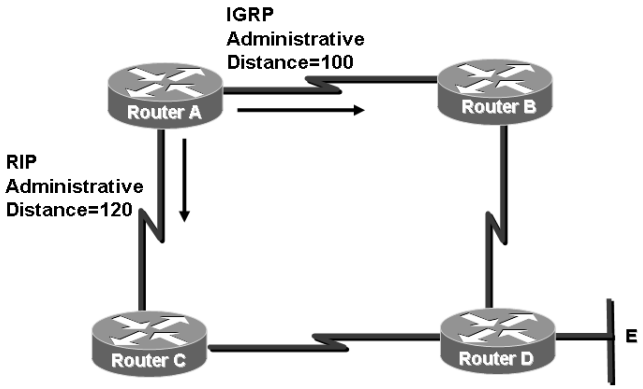
- IGPs จะเป็น AS ที่ใช้สื่อสารภายใน AS เดียวกัน คือจะใช้สื่อสารกันระหว่างสมาชิกภายในกลุ่มของตนเองจะใช้กับ โพรโทคอลประเภท RIP, IGRP
- EGPs จะเป็น AS ที่ใช้สื่อสารข้าม AS คือระหว่าง AS กับ AS ใช้กับโพลโตคอลประเภท BGP



รูปที่ 8.5 การทำงานของ Autonomous System

Administrative Distance (AD)

AD คือค่าของ Cost ที่กำหนดขึ้นมาเพื่อกำหนดน้ำหนักให้แต่ละโพลโตคอล ค่า AD ที่ได้คิดจากความน่าเชื่อถือของโพรโทคอลนั้น ๆ ว่าน่าเชื่อถือเพียงใด เช่นวิธีการค้นหาเส้นทางใช้ปัจจัยอะไรบ้างเข้ามาวิเคราะห์ จากตัวอย่างรูปที่ 8.6 จะเห็นว่าค่า AD ของ IGRP = 100, RIP = 120 ซึ่งเลขที่มีจำนวนน้อย ๆ จะดีกว่า ในที่นี้ IGRP จะมีความน่าเชื่อถือมากกว่า RIP ดังนั้นถ้ามีการส่งข้อมูลจากเราเตอร์ A ไป D เราเตอร์จะเลือกส่งข้อมูลไปทางเราเตอร์ B



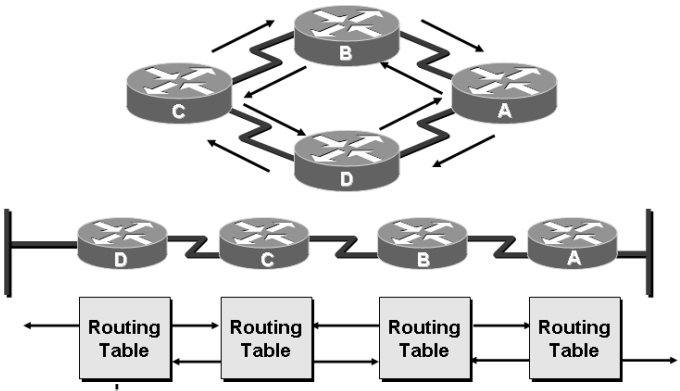
รูปที่ 8.6 ตัวอย่าง Administrative Distance (AD)

ประเภทของโพลโตคอลค้นหาเส้นทาง

สามารถแบ่งออกได้เป็น 3 กลุ่มคือ Distance Vector, Link State, Hybrid Routing แต่ละประเภททำงานดังนี้

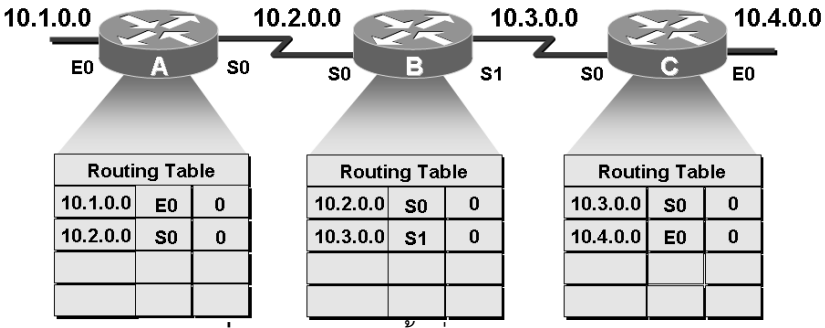
Distance Vector เป็นโพรโทคอลที่พิจารณาแต่ระยะทางที่เชื่อมต่อกันเท่านั้น โดยการนับจำนวนเพื่อนบ้านที่ต่อด้วยเป็น 1 ฮอบ ถ้ามีเพื่อนบ้านต่อไปอีกก็นับเพิ่มอีก 1 ฮอบไปเรื่อย ๆ ดัง

รูปที่ 8.7 ข้อมูลของตารางเราติ้งจะถูกส่งไปให้เพื่อบ้านที่ต่ออยู่เป็นระยะเวลาที่แน่นอน โดยจะส่งข้อมูลออกไปทั้งตาราง เมื่อเราเตอร์ได้รับข้อมูลแล้วก็จะทำการคำนวณค่าของฮอบว่าเป็นเท่าไร



รูปที่ 8.7 แสดงการทำงานของ Distance Vector

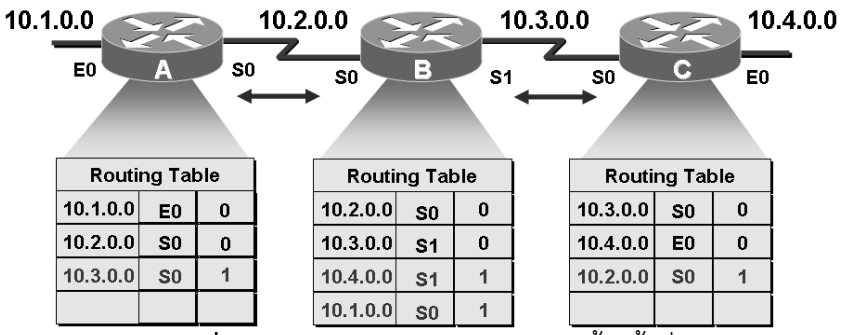
ตัวอย่างการอัพเดทตารางเราติ้งของ Distance Vector



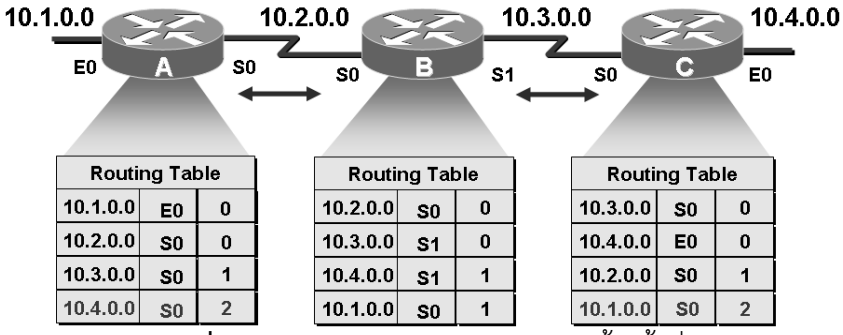
รูปที่ 8.8 ตารางเราติ้งเริ่มต้นก่อนการสื่อสาร

จากรูปที่ 8.8 แสดงเราเตอร์ 3 ตัวที่เชื่อมต่อกันและทำการอีนาเบิลโพรโทคอล RIP ไว้ เริ่มแรกตาราเราติ้งของเราเตอร์แต่ละตัวจะเก็บข้อมูลของเพื่อนบ้านไว้คือ หมายเลขของเน็ตเวิร์ค, หมายเลขของอินเตอร์เฟส, จำนวนฮอบ จากรูป เราเตอร์ B เก็บเน็ตเวิร์ค 10.2.0.0 อินเตอร์เฟส S0 จำนวนฮอบเท่ากับ 0 ไว้บรรทัดแรกไว้ในตารางเราติ้ง บรรทัดที่สองเก็บข้อมูลของเน็ตเวิร์ค 10.3.0.0 อินเตอร์เฟส S0 ฮอบเท่ากับ 0

เมื่อถึงระยะเวลาที่ต้องมีการปรับปรุงตารางเราติ้ง เราเตอร์ทุกตัวจะเริ่มส่งข้อมูลทั้งตารางไปให้เพื่อบ้านเพื่อทำการปรับปรุงข้อมูลของตัวเอง รูปที่ 8.9 แสดงการปรับปรุงข้อมูลของเราเตอร์แต่ละตัว จากรูป เราเตอร์ B จะได้รับข้อมูลจากเราเตอร์ C ว่ามีเน็ตเวิร์ค 10.4.0.0 เชื่อมต่อด้วย เราเตอร์ A จึงทำการใส่ค่าข้อมูลลงไปในตารางเราติ้งเป็น 10.4.0.0 อินเตอร์เฟส S1 แล้วบวกค่าของ ฮอบที่ได้รับมาอีก 1 เราเตอร์ C ก็รับเน็ตเวิร์ค 10.2.0.0 มาเหมือนกันมันจึงทำการเพิ่มฮอบเข้าไปอีก 1 จากนั้นเมื่อเวลาปรับปรุงตารางเราติ้งมาถึงอีกครั้ง ดังรูปที่ 8.10 เราเตอร์ทุกตัวก็จะเริ่มปรับปรุงข้อมูลอีกครั้ง จำนวนของฮอบก็จะถูกเพิ่มขึ้นไปเรื่อย ๆ ตามจำนวนที่เราเตอร์เชื่อมต่ออยู่



รูปที่ 8.9 แสดงการปรับปรุงตารางเราติ้งครั้งที่ 1



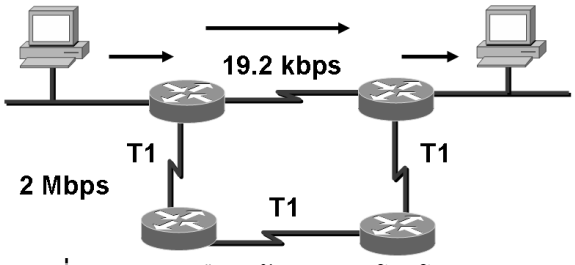
รูปที่ 8.10 แสดงการปรับปรุงตารางเราติ้งครั้งที่ 1

แต่การใช้โพลโตคอลแบบ Distance Vector จะมีปัญหาตามมาในเรื่องของการปรับปรุงตารางเราติ้ง แล้วเกิดลูปคือข้อมูลมีการปรับปรุงซ้ำไปซ้ำมา ทำให้ข้อมูลไม่ตรงกับความเป็นจริง แต่ก็มีวิธีแก้ไข เช่น split horizon, Route Poisoning, Poisoning Reverse, Hold-Down Timers, Triggered Updates ซึ่งไม่ขอกล่าวไว้ในที่นี้เพราะจะทำให้หนังสือมีขนาดใหญ่เกินไป และไม่ใช่จุดประสงค์ของหนังสือเล่มนี้ ถ้าต้องการรายละเอียดเพิ่มเติมให้หาอ่านได้จากหนังสือเน็ตเวิร์คทั่ว ๆ ไป

โพลโตคอล RIP (Distance Vector)

โพรโทคอล RIP เป็นโพรโทคอลชนิด Distance Vector การตัดสินใจของมันจะพิจารณาจำนวนฮอบเท่านั้น ไม่สนใจว่าขนาดของแบนด์วิด คุณสมบัติของ RIP จะประกอบด้วย

1. จำนวนฮอบที่มีได้มากสุดเท่ากับ 6 ฮอบ
2. จะใช้ฮอบเป็นข้อมูลในการตัดสินใจเลือกเส้นทาง
3. เราเตอร์จะปรับปรุงตารางเร้าติ้งทุก ๆ 30 วินาที



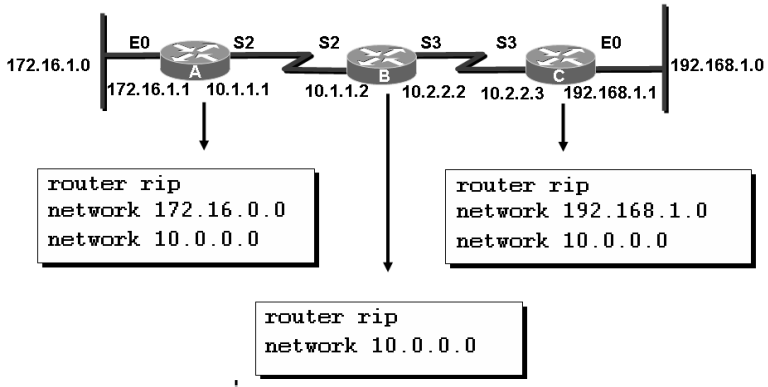
รูปที่ 8.11 การเลือกเส้นทางของโพรโทคอล RIP

จากรูปที่ 8.11 เส้นทางที่ RIP เลือกใช้ในการสื่อสารมีแบนด์วิดเท่ากับ 19.2 Kbps ซึ่งน้อยกว่าเส้นทางที่เป็น T1 ซึ่งมีแบนด์วิดสูงกว่ามาก ทั้งนี้เนื่องจาก RIP จะคิดเฉพาะฮอบเท่านั้น

ตัวอย่างการคอนฟิก RIP

Router(config)#router rip เริ่มการทำงานของโพรเซส RIP ดังรูปที่ 8.12

Router(config-router)#network network-number ใส่เน็ตเวิร์คที่ต้องการและต้องเป็นเน็ตเวิร์คที่เป็น Classful ด้วย (Classful หมายถึงหมายเลขของ subnet-mask เป็น 255.255.0.0 สำหรับคลาส B, 255.255.255.0 สำหรับคลาส C เป็นต้น) รายละเอียดสำหรับการคอนฟิก RIP จะกล่าวในส่วนของแล็ปต่อไป



รูปที่ 8.12 ตัวอย่างการคอนฟิก RIP

โพรโทคอล **IGRP**

IGRP เป็นโพรโทคอลชนิด Distance Vector เช่นเดียวกับ RIP แต่ IGRP จะถูกปรับปรุงให้มีประสิทธิภาพที่ดีขึ้นแก้ข้อเสียของ RIP คุณลักษณะที่เด่น ๆ ของมันคือ

1. รองรับขนาดที่ใหญ่ขึ้นมากกว่า RIP
2. ตอบสนองได้ดีกับเน็ตเวิร์คที่มีขนาดใหญ่ ๆ
3. มีการคิดค่าของ Cost ที่รัดกุมขึ้นหรือที่เรียกว่า metric
4. รองรับการทำงานได้หลายเส้นทาง

การคำนวณหาค่าของ Metric จะประกอบไปด้วย

1. ขนาดของแบนด์วิดในปัจจุบัน (Bandwidth)
2. อัตราการหน่วงของข้อมูล (Delay Time)
3. ความน่าเชื่อถือ (Reliability)
4. ปริมาณของโหลดที่ใช้ (Loading)
5. MTU

การคำนวณหาค่าของ Metric สามารถหาได้จากสูตร

metric = [k1 * bandwidth + (k2 * bandwidth) / (256 - load) + k3 * delay]

เมื่อ k5 ไม่เท่ากับ 0 จะมีสัมมการดังนี้คือ

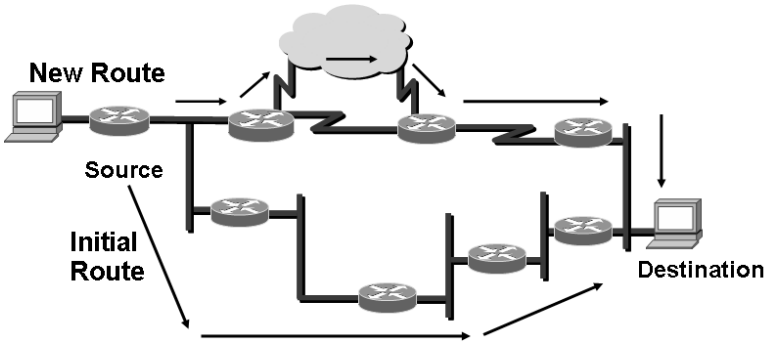
metric = metric * (k5/(reliability + k4))

โดยปกติค่าดีฟอลท์ของ k1=k3=1 และ k2=k4=k5=0

เมื่อค่าของของ k1 ถึง k5 เป็น 0 จะทำให้เขียนเป็นสมการได้ดังนี้

metric = bandwidth + delay

เส้นทางที่มากที่สุดที่ IGRP รองรับได้คือ 6 เส้นทาง (ดีฟอลท์คือ 4 เส้นทาง) ค่าของ Metric ภายในสามารถเปลี่ยนแปลงได้ตามโครงสร้างจริงของเครือข่าย ดังรูปที่ 8.13



รูปที่ 8.13 การเลือกเส้นทางของ IGRP

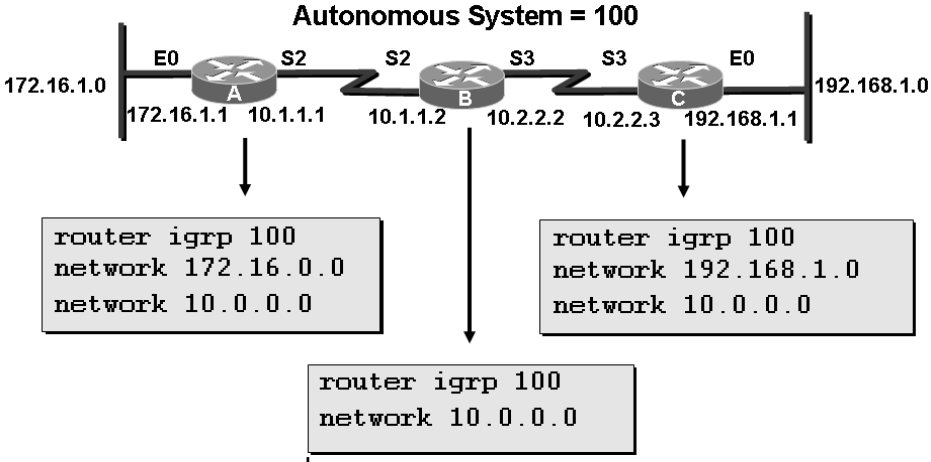
ตัวอย่างการคอนฟิก IGRP

Router(config)#router igrp autonomous-system กำหนดเราติ้งแบบ IGRP และหมายเลขของ AS ดังรูปที่ 8.14

Router(config-router)#network network-number กำหนดหมายเลขเน็ตเวิร์ค

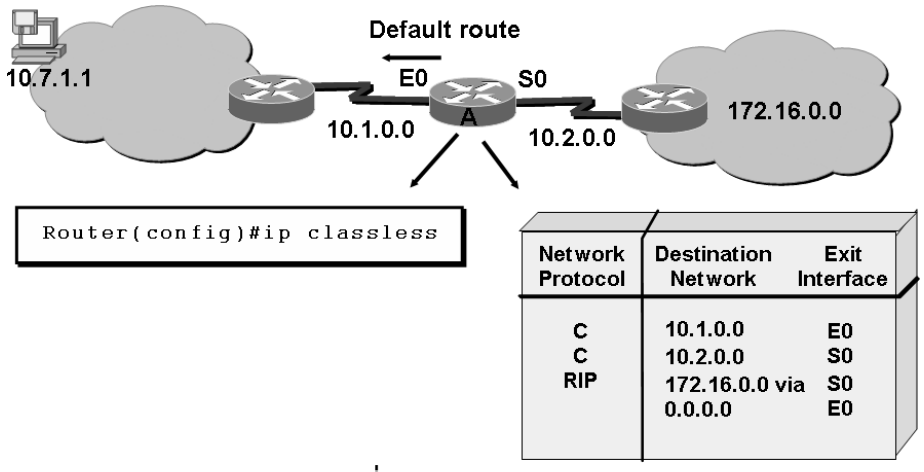
Router(config-router)#networkr variance multiplier ควบคุมการทำงานของ Load Balance

Router(config-router)#network traffic-share{blance/min} ควบคุม Load Balance ว่าจะให้กระจายทราฟฟิกอย่างไร



รูปที่ 8.14 ตัวอย่างการคอนฟิก IGRP

IP Classless คำสั่งนี้ไม่ได้เกี่ยวข้องกับเรื่องของคลาสแต่อย่างใด มันใช้สำหรับระบุว่า ถ้าข้อมูลในตารางเร้าติ้งไม่มีแล้วมันจะตัดสินใจทำอย่างไรต่อไป



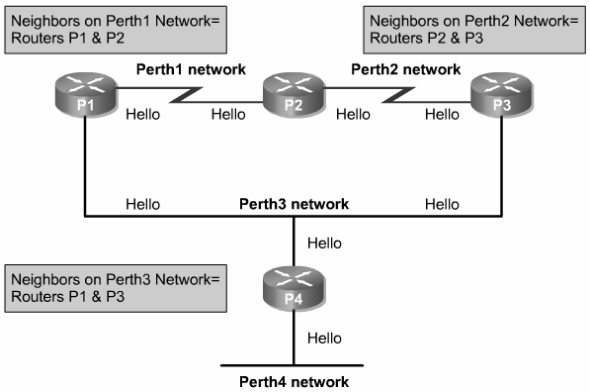
รูปที่ 8.15 ip classless

จากตัวอย่างรูปที่ 8.15 เราเตอร์ A จะมีเส้นทางของเน็ตเวิร์คที่มันเชื่อมต่ออยู่ เก็บไว้ในตารางเร้าติ้ง สมมุติว่าต้องการส่งข้อมูลไปยัง ip หมายเลข 10.7.1.1 ถ้าเราเตอร์ A ไม่ได้ทำ ip calssless ไว้จะทำให้ข้อมูลที่ส่งไปถูก Drop ทันที เพราะว่าในโพรโทคอลแบบ RIP จะไม่ยอมให้ทำการประการเน็ตเวิร์คเป็นคลาสเลส ปกติบนเราเตอร์จะอีนาเบิลไอพีคลาสเลสไว้ให้โดยดีฟอลท์อยู่แล้ว

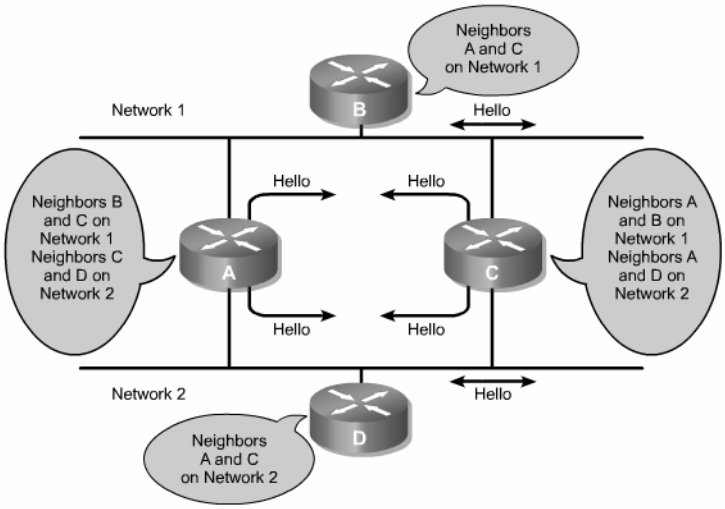
Link State

เป็นโพรโทคอลที่พิจารณาขนาดของแบนด์วิดเป็นหลัก คุณสมบัติของเราติ้งแบบ Link State นี้จะคุณสมบัติดังนี้

1. เมื่อเน็ตเวิร์คมีการเปลียนแปลงจะเกิดการตอบสนองได้อย่างรวดเร็ว
2. ส่งข้อมูลเฉพาะที่เปลี่ยนแปลงออกไปเท่านั้น ไม่ได้ส่งข้อมูลทั้งหมด
3. ส่งข้อมูลเพื่ออัพเดทซึ่งกันและกันตามเวลาที่กำหนด ซึ่เรียกว่า Link-state refreshes
4. ใช้วิธีการที่เรียกว่า Hello เพ็กเก็ตเพื่อตรวจสอบเพื่อนบ้านว่ายังอยู่หรือไม่

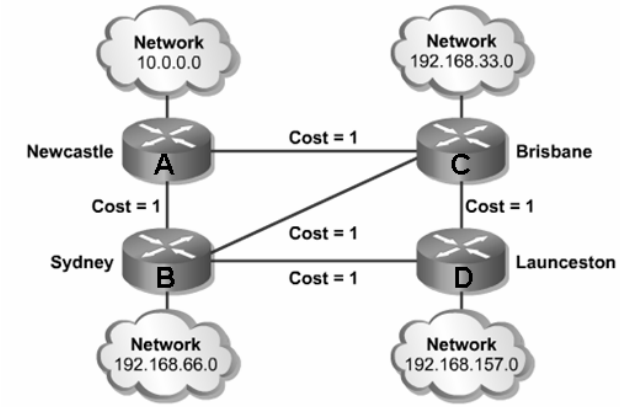


รูปที่ 8.16 ส่ง Hello Packet เพื่อตรวจสอบสถานะของเพื่อนบ้าน



รูปที่ 8.17 วิธีการที่ใช้ในการแลกเปลี่ยนข่าวสาร

จากรูปที่ 8.16, 8.17 เป็นวิธีการที่ใช้สำหรับแลกเปลี่ยนข่าวสารของเราเตอร์แบบ link-state จากรูปเน็ตเวิร์คจะถูกแบ่งออกเป็น 2 เน็ตเวิร์ค คือ เน็ตเวิร์คที่ 1 ประกอบด้วย อินเตอร์เฟสขาหนึ่งของ เราเตอร์ A, เราเตอร์ B, ขาอินเตอร์เฟสข้างหนึ่งของ C ส่วนเน็ตเวิร์คที่ 2 ประกอบด้วย อินเตอร์เฟสข้างหนึ่งของเราเตอร์ A และ C และเราเตอร์ D สมมุติว่าเราเตอร์ A ต้องการส่ง hello แพ็กเก็ตออกไป มันจะส่งไปให้กับเราเตอร์ B และ C ที่เน็ตเวิร์คที่ 1 และส่งไปให้ D และ C ที่เน็ตเวิร์คที่ 2 ทำเช่นนี้ไปเรื่อย ๆ จนครบทุก ๆ เราเตอร์



Router	Destination	Next Hop	Cost
A	192.168.66.0	B	1
A	192.168.33.0	C	1
A	192.168.157.0	B	2
A	192.168.157.0	C	2
B	10.0.0.0	A	1
B	192.168.33.0	C	1
B	192.168.157.0	D	1
C	10.0.0.0	A	1
C	185.134.0.0	B	1
C	192.168.157.0	D	1
D	10.0.0.0	B	2
D	10.0.0.0	C	2
D	192.168.66.0	B	1
D	192.168.33.0	C	1

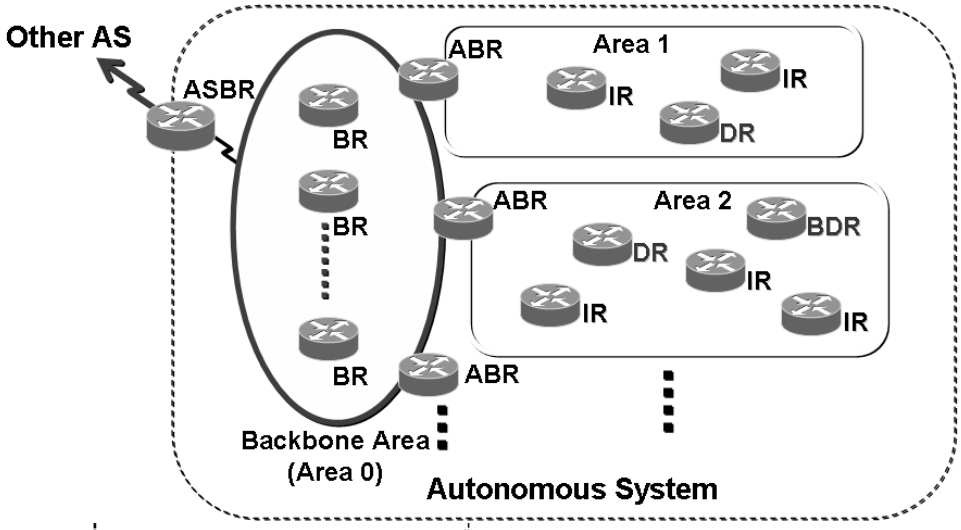
รูปที่ 8.18 การค้นหาเส้นทางด้วยวิธีการแบบ Link-state

จากรูป 8.18 เมื่อเราเตอร์ A ต้องการติดต่อไปยังเน็ตเวิร์ค 192.168.66.0 ซึ่งต่ออยู่กับเราเตอร์ B ในตารางเราติ้งจะบันทึกข้อมูลไว้คือ จาก A ไปยังเน็ตเวิร์ค 192.168.66.0 Next Hop คือ B และมีค่า Cost เป็น 1 แต่ถ้าต้องการเดินทางไปยังเน็ตเวิร์ค 192.168.157.0 ต้องผ่านไปยังเราเตอร์ B ก่อนจากนั้นจึงไปยัง D ซึ่งมีค่า Cost เป็น 2

โพรโทคอล OSPF

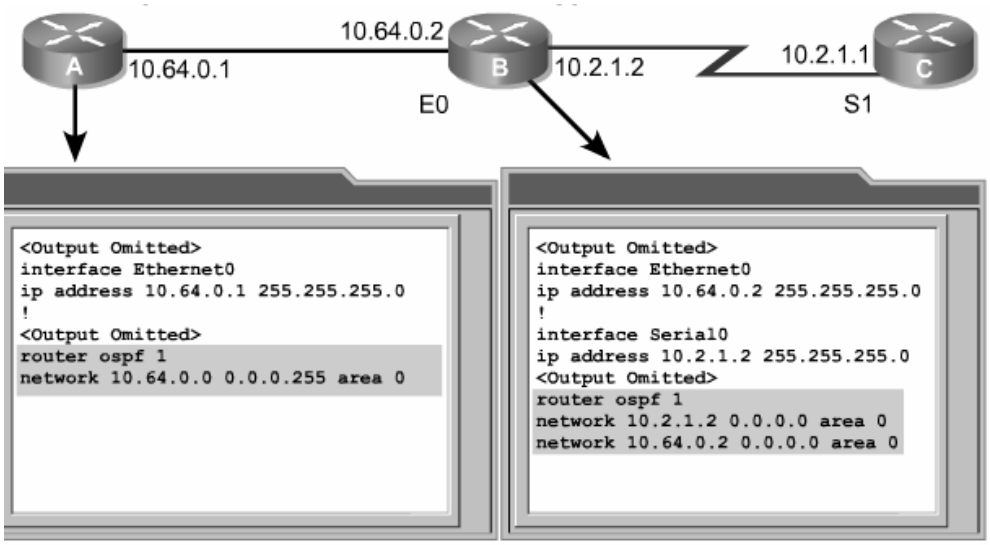
โพรโทคอล OSPF มีการทำงานแบบ Link-state คือการพิจารณาค่า cost ที่เกิดจากการคำนวณขนาดของแบนด์วิดเป็นหลัก ส่วนประกอบหลัก ๆ ของ OSPF มีดังนี้

1. Routing Table เป็นตารางสำหรับเก็บเส้นทาง จะถูกสร้างขึ้นเมื่อมีการรันโพรโทคอลเราติ้งขึ้นมา โดยตารางเราติ้งนี้จะมีเพียง 1 ตารางเท่านั้นและจะไม่มีการซ้ำกัน
2. Cost ค่าที่ถูกกำหนดให้แต่ละอินเตอร์เฟสซึ่งการคำนวณ Cost มีพื้นฐานอยู่กับขนาดของแบนด์วิด
3. Adjacencies database เป็นฐานข้อมูลที่เก็บข้อมูลของเพื่อนบ้านไว้
4. Designated router (DR) เป็นเราเตอร์ตัวหนึ่งที่ถูกโปรโหมดขึ้นมาเพื่อทำหน้าที่เป็นตัวแทนของ Area หนึ่ง ๆ
5. backup designated router (BDR) เป็นเราเตอร์ที่ถูกเลือกให้ทำหน้าที่เป็นตัวแทนเช่นเดียวกันแต่ไว้สำหรับ DR ไม่สามารถทำงานได้ BDR จะทำหน้าที่แทน
6. Area คือกลุ่มของเราเตอร์จำนวนหนึ่งที่รวมตัวกันเข้าเพื่อแบ่งเราเตอร์ออกเป็นกลุ่มย่อย ๆ ประโยชน์เพื่อต้องการลดจำนวนของข้อมูลในตารางเร้าติ้ง ข้อมูลจะถูกแลกเปลี่ยนกันในกลุ่มของตนเอง แต่ถ้าต้องการส่งข้อมูลไปยังกลุ่มอื่น ๆ จะต้องให้ DR เป็นผู้ส่งให้ซึ่งเปรียบเสมือนเป็นตัวแทนของกลุ่ม ดังรูปที่ 8.19



รูปที่ 8.19 การแบ่งกลุ่มของ OSPF เพื่อลดจำนวนของทราฟฟิกและจำนวนของตารางเร้าติ้ง

7. IR คือสมาชิกที่อยู่ภายในกลุ่มหรือ Area ของตัวเอง
8. ABR เป็นเราเตอร์ตัวหนึ่งในกลุ่มที่มีอินเตอร์เฟสขาหนึ่งต่อไปยัง Backbone Area
9. BR อินเตอร์เฟสที่ต่ออยู่กับ Backbone Area 0
10. ASBR แลกเปลี่ยนข้อมูลที่อยู่ต่าง AS กัน



รูปที่ 8.20 ตัวอย่างการคอนฟิก OSPF

เนื้อหาที่กล่าวมาจะเป็นเนื้อหาที่สรุป ๆ เกี่ยวกับเราติ้งโพลโตคอลที่ใช้ทำงานอยู่ในปัจจุบัน ซึ่งยังมีอีกหลายโพรโทคอลที่ยังไม่ได้กล่าวถึงซึ่งกล่าวโดยสรุปเป็นดังนี้คือ

การคำนวณหา cost แบ่งได้ 2 วิธีคือ

1. คำนวณจากระยะทาง (ฮอบ)

   โพลโตคอลที่ใช้วิธีนี้คือ RIP v1,2, IGRP

2. คำนวณจากแบนด์วิด

   โพลโตคอลที่ใช้วิธีนี้คือ OSPF

3. ผสมระหว่าง 2 วิธี (Hybrid)

   โพลโตคอลที่ใช้วิธีนี้คือ EIGRP

หมายเหตุ: ถ้าผู้อ่านต้องการทราบเนื้อหาของโพรโทคอลค้นหาเส้นทางโดยละเอียด สามารถหาอ่านได้จากรายการหนังสือที่ผู้เขียนได้อ้างถึงไว้ในส่วนของเอกสารอ้างอิง

การคอนฟิกเส้นทางแบบ static

เส้นทางชนิด static ผู้ดูแลระบบจะเป็นคนกำหนดเส้นทางในการรับส่งข้อมูลเอง ตามความเหมาะสมที่ผู้ดูแลระบบเห็นว่าถูกต้องที่สุด

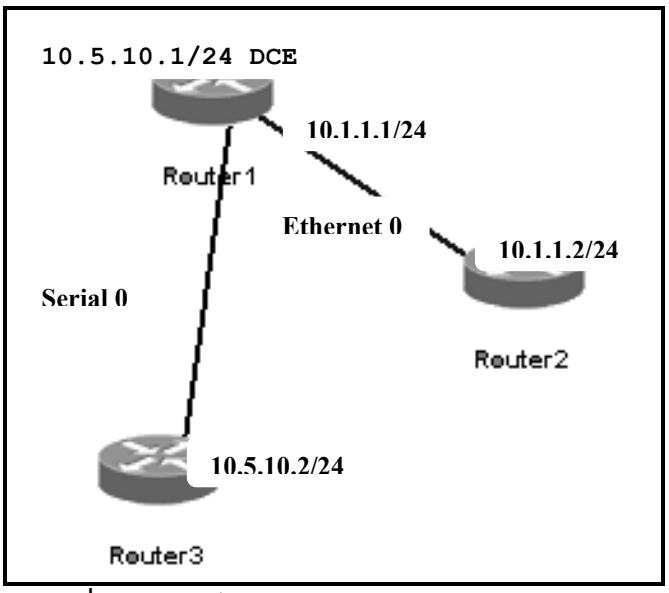
จุดมุ่งหมาย : เรียนรู้การคอนฟิกเส้นทางแบบ Static

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 3 ตัวคือ Router1, Router2, Router3

การสร้าง Network Map : เหมือน ดังตารางที่ 8.1 และดังรูปที่ 8.21

ตารางที่ 8.1 หมายเลขไอพีของเราเตอร์

	Router1	Router2	Router3
Interface Ethernet 0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Interface Serial 0	12.5.10.1 255.255.255.0		12.5.10.2 255.255.255.0



รูปที่ 8.21 ผังเน็ตเวิร์คสำหรับทดสอบ Static Route

หน้าต่าง Simulator :

1. ที่ Router1

   Router>*enable*

   Router#*config t*

   Router(config)#*hostname Router1*

   Router1(config)#*interface ethernet 0* [คอนฟิกอินเตอร์เฟสอีเทอร์เน็ต 0]

   Router1(config-if)#*ip address 10.1.1.1 255.255.255.0*

   Router1(config-if)#*no shutdown*

   Router1(config-if)#*exit*

   Router1(config)#*interface serial 0* [คอนฟิกอินเตอร์เฟสซีเรียล 0]

   Router1(config-if)#*ip address 12.5.10.1 255.255.255.0*

   Router1(config-if)#*clock rate 5600*

   Router1(config-if)#*no shutdown*

2. ที่ Router2

Router>*enable*

Router#*config t*

Router(config)#*hostname Router2*

Router2(config)#*interface ethernet 0*

Router2(config-if)#*ip address 10.1.1.2 255.255.255.0*

Router2(config-if)#*no shutdown*

Router2(config-if)#*end*

Router2#*ping 10.1.1.1* [ทดสอบการเชื่อมต่อกับ Router1]

3. ที่ Router3

Router>*enable*

Router#*config t*

Router(config)#*hostname Router3*

Router3(config)#*interface serial 0*

Router3(config-if)#*ip address 12.5.10.2 255.255.255.0*

Router3(config-if)#*no shutdown*

Router3(config-if)#*end*

Router3#*ping 12.5.10.1* [ทดสอบการเชื่อมต่อกับ Router1]

4. จากรูปเราเตอร์ทั้งหมดได้เชื่อมต่อกันเรียบร้อยแล้ว แต่ถ้าเน็ตเวิร์ควง 10.5.10.0 ต้องการสื่อสารกับเน็เวิร์ควง 10.1.1.0 จะไม่สามารถทำได้เนื่องจากเส้นทางไม่ได้ถูกประกาศไว้ ดังนั้นข้อมูลที่ส่งไปจะถูกดรอปทิ้งทั้งหมด ทดลองโดยการ ping จาก Router3 ไปยัง Router2 ด้วยคำสั่ง ping 10.1.1.2 จะได้ผลลัพธ์คือ (....) ซึ่งหมายความว่าไม่สำเร็จ แล้วคำถามที่ตามมาคือเราจะทำอย่างไรให้เน็ตเวิร์คทั้ง 2 วงสามารถคุยกันได้? คำตอบก็คือเราจะต้องกำหนดเส้นทางให้เราเตอร์รู้ว่าจะส่งข้อมูลผ่านไปยังเน็ตเวิร์คหนึ่งไปอีกเน็ตเวิร์คอื่นๆ นั้นมีเส้นทางเป็นอย่างไร ดังนั้นใน Lab นี้จึงต้องมีการประกาศเส้นทางแบบ static บนอุปกรณ์เราเตอร์

Router3#*config t* [เราเตอร์ 3]

Router3(config)#*ip route 10.1.1.0 255.255.255.0 12.5.10.1* [กำหนดเส้นทางแบบ Static]

5. การกำหนด ip route จากข้อที่ 4 แล้วควรจะสามารถ ping ไอพี 12.5.10.1, 10.1.1.1, 10.1.1.2 สำเร็จ

บน Router3

Router3#*ping 12.5.10.1* [ping อินเตอร์เฟสซีเรียล 0 ของ Router1]

Router3#*ping 10.1.1.1* [ping อินเตอร์เฟสอีเทอร์เน็ต 0 ของ Router1]

Router3#*ping 10.1.1.2* [ping อินเตอร์เฟสอีเทอร์เน็ต 0 ของ Router2]

เมื่อ Packet ที่ส่งจาก Router3 ไปยังเน็ตเวิร์ค 10.1.1.0 มันจะต้องส่งผ่านไปทาง Router1 ก่อน (เพราะการกำหนด Static Route เป็น 12.5.10.1) เมื่อ Router1 ได้รับ Packet มันจะเปิดตารางเร้าติ้งว่ามีเน็ตเวิร์คที่ต้องการส่งต่ออยู่หรือไม่ ถ้าไม่มีมันจะดร็อบทิ้ง ถ้ามีมันจะส่งไปยังอินเตอร์เฟสที่เป็นของเน็ตเวิร์คนั้น

6. **Router3#*show ip route*** [เป็นคำสั่งที่ใช้ดูตารางเร้าติ้ง]

C 192.168.1.0 is directly connected, Ethernet 0 [เน็ตเวิร์ค 192.168.1.0 เชื่อต่อโดยตรงกับอินเตอร์เฟสอีเทอร์เน็ต 0 ของ Router3]

C 12.5.10.0 is directly connected, Serial 0 [เน็ตเวิร์ค 12.5.10.0 เชื่อต่อโดยตรงกับอินเตอร์เฟสซีเรียล 0 ของ Router3]

S 10.1.1.0 [1/0] via 12.5.10.1 [เส้นทางที่จะไปเน็ตเวิร์ค 10.1.1.0 จะต้องผ่านไอพีแอดเดรส 12.5.10.1 ของ Router1 ค่า [1/0] 1 หมายถึง Administrative distance โดยดีฟอลท์มีค่าเป็น 1, 0 หมายถึงค่าของ Matric จำนวนของ hop ในที่นี้มีค่าเป็น 0 ดังรูปที่ 8.22

```
Router3#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is not set

     192.168.1.0/24 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Ethernet0
     12.0.0.0/24 is subnetted, 1 subnets
C       12.5.10.0 is directly connected, Serial0
     10.0.0.0/24 is subnetted, 1 subnets
S       10.1.1.0 [1/0] via 12.5.10.1
```

รูปที่ 8.22 แสดงตารางเร้าติ้งด้วยคำสั่ง show ip route

7. เมื่อต้องการให้ Router2 สามาถติดต่อกับ Router3 ได้ จำเป็นต้องกำหนดเส้นทางให้กับ Router2 เช่นเดียวกับที่กำหนดให้ Router3

Router2(config)#*ip route 12.5.10.0 255.255.255.0 10.1.1.1* [static route]

Router2#(config)#*end*

Router2#*show ip route* [แสดงตารางเร้าติ้ง]

Router2#*ping 10.1.1.1* [ping อินเตอร์เฟสอีเทอร์เน็ต 0 ของ Router1]

Router2#*ping 12.5.10.1* [ping อินเตอร์เฟสซีเรียล 0 ของ Router1]

Router2#*ping 12.5.10.2* [ping อินเตอร์เฟสซีเรียล 0 ของ Router3]

การคอนฟิกเร้าติ้งโพรโทคอลแบบ RIP

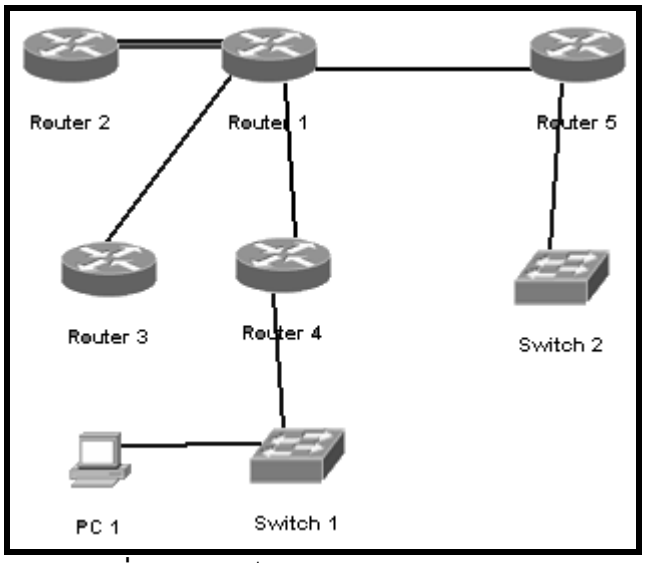
LAB นี้เราจะมาเรียนรู้เรื่องของโพรโทคอลเร้าติ้งแบบ RIP จากบทก่อน เราทราบแล้วว่า RIP เป็นโพรโทคอลชนิด Distance Vector คือจะสนใจเฉพาะจำนวน hop (hop คือจำนวนของเราเตอร์ที่เชื่อมต่อกันและสามารถมี hop ได้สูงสุดไม่เกิน 15 hop เท่านั้น) ของเราเตอร์ที่เชื่อมต่อกันเท่านั้น จะไม่สนใจเรื่องของแบนด์วิด RIP จะมี 2 เวอร์ชันคือ v1, 2 rip จะส่งข้อมูลตารางเร้าติ้งของตนเองไปให้เพื่อนบ้านทุก ๆ 30 วินาที และเมื่อได้รับข้อมูลแล้วมันจะทำการปรับปรุงตารางเร้าติ้งของตนเอง

จุดมุ่งหมาย : เรียนรู้การคอนฟิกโพรโทคอลแบบ RIP

1. กำหนดชื่อเราเตอร์และสั่งให้อินเตอร์เฟสทำงาน
2. คอนฟิก RIP เร้าติ้งโพรโทคอล
3. ตรวจสอบตารางเร้าติ้ง
4. สำรวจข้อมูลของ RIP ที่ใช้สื่อสารกัน

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 5 ตัวคือ Router1, Router2, Router3, Router4, Router5 สวิตช์ 2 ตัวและ PC 1 ตัวดังรูปข้างล่าง

การสร้าง Network Map : ดังรูปที่ 8.23 และ ตารางที่ 8.2



รูปที่ 8.23 ผังเน็ตเวิร์คสำหรับใช้ทดลอง RIP

ตารางที่ 8.2 ข้อมูลการเชื่อมต่อ

อุปกรณ์	การเชื่อมต่อ
Router 1 (2516)	Router 1(Ethernet 0) → Router 2(Ethernet 0) Router 1(Ethernet 1) → Router 3(Ethernet 0) Router 1(Serial 0) → Router 4(Serial 0) DCE Router 1(Serial 1) → Router 5(Serial 0) DCE Router 1(Bri 0) → Router 2(Bri 0)
Router 2 (2501)	Router 2(Ethernet 0) → Router 1(Ethernet 0) Router 2(Bri 0) → Router 2(Bri 0)

Router 3 (2501)	Router 3(Ethernet 0) → Router 1(Ethernet 1)
Router 4 (2501)	Router 4(Ethernet 0) → Switch 1(Ethernet 0/1) Router 4(Serial 0) → Router 1(Serial 0) DTE
Router 5 (2501)	Router 5(Ethernet 0) → Switch 2(FastEthernet 0/1) Router 5(Serial 0) → Router 1(Serial 1) DTE
Switch 1 (1912)	Switch 1(Ethernet 0/1) → Router 4(Ethernet 0) Switch 1(Ethernet 0/2) → PC 1(Ethernet 0)
Switch 2 (2905)	Switch 2(FastEthernet 0/1) → Router 5(Ethernet 0)
PC 1	PC 1(Ethernet 0) → Switch 1(Ethernet 0/2)

สีดำ หมายถึง สายชนิดซีเรียล (Serial)

สีน้ำเงิน หมายถึง สายชนิดอีเทอร์เน็ต (Ethernet)

สีแดง หมายถึง สายชนิด Bri (สำหรับใช้คอนฟิกกับ ISDN)

ขั้นตอนการสร้างผังเน็ตเวิร์ค

1. **จากหน้าต่างหลักเลือก New NetMap → Ok → ในช่อง Avialable Routers เลือก 2500 Series → ดับเบิลคลิกที่**เราเตอร์ **2516 แล้วเลือก Apply**
2. **ในช่อง Avialable Routers เลือก 2500 Series → ดับเบิลคลิกที่**เราเตอร์ **2501 แล้วเลือก Apply → ให้ทำซ้ำข้อ 2 ทั้งหมด 4 ตัว คือ Router2,3,4,5**
3. **ในช่อง Avialable Switchs เลือก 1900 Series → ดับเบิลคลิกที่สวิตซ์ 1912 แล้วเลือก Apply**
4. **ในช่อง Avialable Switchs เลือก 2900 Series → ดับเบิลคลิกที่สวิตซ์ 2950 แล้วเลือก Apply**
5. **ในช่อง Other Devices → ดับเบิลคลิกที่ PC เลือก Apply**
6. **คลิกขวาที่ Router1 → Add Connection to → Serial 0 → Point-to-Point Connection → Next → ในช่อง Available Devices ให้เลือก Router4 และช่อง Serial Interfaces ให้เลือก Serial 0 → Finish → Router1 Serial 0 เป็น DCE → OK**
7. **คลิกขวาที่ Router1 → Add Connection to → Serial 1 → Point-to-Point Connection → Next → ในช่อง Available Devices ให้เลือก Router5 และช่อง Serial Interfaces ให้เลือก Serial 1 → Finish → Router1 Serial 1 เป็น DCE → OK**

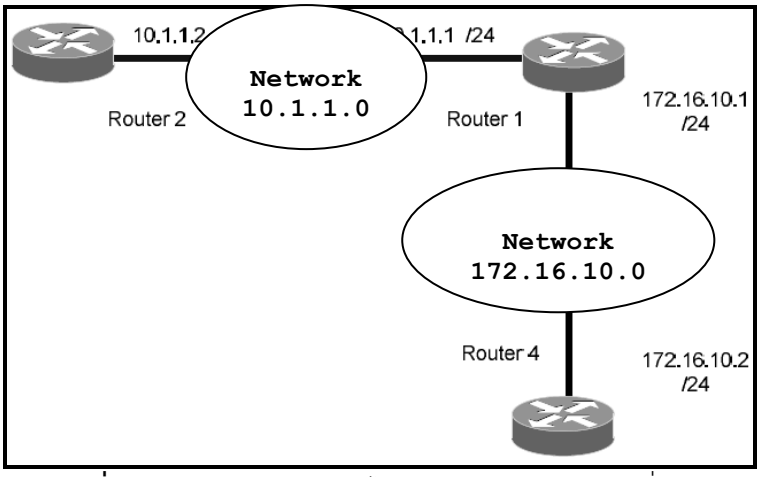
8. **คลิกขวาที่ Router1 → Add Connection to → Ethernet 0 → ในช่อง Available Devices ให้เลือก Router3 และช่อง Serial Interfaces ให้เลือก Ethernet 0 → Finish**
9. **คลิกขวาที่ Router1 → Add Connection to → Ethernet 1 → ในช่อง Available Devices ให้เลือก Router2 และช่อง Serial Interfaces ให้เลือก Ethernet 0 → Finish**
10. **คลิกขวาที่ Router1 → Add Connection to → Bri 0 → ในช่อง Available Devices ให้เลือก Router2 และช่อง Serial Interfaces ให้เลือก Bri 0 → Finish**
11. **คลิกขวาที่ Router4 → Add Connection to → Ethernet 0 → ในช่อง Available Devices ให้เลือก Switch1 และช่อง Serial Interfaces ให้เลือก Ethernet 0/1 → Finish**
12. **คลิกขวาที่ Router5 → Add Connection to → Ethernet 0 → ในช่อง Available Devices ให้เลือก Switch2 และช่อง Serial Interfaces ให้เลือก FastEthernet 0/1 → Finish**
13. **คลิกขวาที่ Switch1 → Add Connection to → Ethernet 0/2 → ในช่อง Available Devices ให้เลือก PC และช่อง Serial Interfaces ให้เลือก Ethernet 0 → Finish**

สำหรับ LAB นี้จะใช้เราเตอร์เพียง 3 ตัวเท่านั้น

ตารางที่ 8.3 หมายเลขไอพีแอดเดรสของเราเตอร์

	Router1	Router2	Router4
Interface Ethernet 0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Interface Serial 0	172.16.10.1 255.255.255.0		172.16.10.2 255.255.255.0

ให้กำหนดหมายเลขไอพีในตารางที่ 8.3 ให้กับ Router1, 2, 4 ดังรูป



รูปที่ 8.24 กำหนดหมายเลขไอพีแอดเดรสจากตารางที่ 8.3

หน้าต่าง Simulator :

1. บน Router1

Router>*enable*

Router#*config t*

Router(config)#*hostname Router1*

Router1(config)#*interface ethernet 0*

Router1(config-if)#*ip address 10.1.1.1 255.255.255.0*

Router1(config-if)#*no shutdown* [เชื่อมต่อกับ Router2 ผ่านทางอินเตอร์เฟสอีเทอร์เน็ต 0]

Router1(config-if)#*exit*

Router1(config)#*interface serial 0*

Router1(config-if)#*ip address 172.16.10.1 255.255.255.0*

Router1(config-if)#*no shutdown*

Router1(config-if)#*clock rate 5600*

2. บน Router2

Router>*enable*

Router#*conf t*

Router(config)#*hostname Router2*

Router2(config)#*interface ethernet 0*

Router2(config-if)#*ip address 10.1.1.2 255.255.255.0*

Router2(config-if)#*no shutdown*

Router2(config-if)#*end*

Router2#*show cdp neighbors*

Router2#*ping 10.1.1.1* [ping ทดสอบว่าการเชื่อมต่อสำเร็จ]

3. บน Router4

Router>*enable*

Router#*conf t*

Router(config)#*hostname Router4*

Router4(config)#*interface serial 0*

Router4(config-if)#*ip address 172.16.10.2 255.255.255.0*

Router4(config-if)#*no shutdown*

Router4(config-if)#*exit*

Router4#*show cdp neighbors*

Router4#*ping 172.16.10.1* [ping ทดสอบการเชื่อมต่อระหว่าง Router1 กับ Router4 ผ่านซีเรียล 0]

4. การคอนฟิกโพรโทคอล RIP จะต้องเข้าไปในโหมดของคอนฟิก (config term) บน Router1

Router1(config-if)#*exit*

Router1(config)#

5. ป้อนคำสั่งเพื่ออีนาเบิลโพรโทคอล RIP ด้วยคำสั่ง router <protocol>

Router1(config)#*router rip* [เข้าสู่โหมดเราเตอร์]

Router1(config-router)#

6. ใส่เน็ตเวิร์คที่ต้องการประกาศ ด้วยคำสั่ง network <network id>

Router1(config-router)#*network 10.1.1.0* [เน็ตเวิร์คระหว่าง Router1 และ Router2 RIP 1,2 จะต้องทำการประกาศเน็ตเวิร์คเป็นแบบ Classful เท่านั้น]

Router1(config-router)#*network 172.16.10.0* [เน็ตเวิร์คระหว่าง Router1 และ Router4]

7. บน Router2 ให้อีนาเบิล RIP และประทำการประกาศเน็ตเวิร์ค

Router2#conf t

Router2(config)#*router rip*

Router2(config-router)#*network 10.1.1.0* [ประกาศเน็ตเวิร์คเพียง 1 เน็ตเวิร์คเท่านั้นเนื่องจาก Router2 มีเพียงอินเตอร์เฟสเดียวที่เชื่อมต่อกับเพื่อนบ้าน]

8. บน Router4 ให้อีนาเบิล RIP และประทำการประกาศเน็ตเวิร์ค

Router4#*conf t*

Router4(config)#*router rip*

Router4(config-router)#*network 172.16.10.0* [ประกาศเน็ตเวิร์คเพียง 1 เน็ตเวิร์คเท่านั้นเนื่องจาก Router4 มีเพียงอินเตอร์เฟสเดียวที่เชื่อมต่อกับเพื่อนบ้าน]

9. ถ้าไม่มีการผิดพลาด ถึงตรงนี้เราเตอร์ทุกตัวจะต้องสามารถติดต่อกันได้ เช่น เราสามารถ ping จาก Router2 ไปยัง Router4 ได้ โดยก่อนอีนาเบิล RIP จะไม่สามารถ ping ได้

บน Router2

Router2(config-router)#*end*

Router2#*ping 172.16.10.2* [ping ไปยังอินเตอร์เฟสซีเรียลของ Router4] ดังรูปที่ 8.25

```
Router2#ping 172.16.10.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 8.25 ping จาก Router2 ไปยัง Router4 สำเร็จ

10. บน Router4

Router4(config-router)#*end*

Router4#*ping 10.1.1.2* [ping ไปยังอินเตอร์เฟสอีเทอร์เน็ตของ Router4]

Router4#*ping 10.1.1.1* [ping ไปยังอินเตอร์เฟสอีเทอร์เน็ตของ Router1] ดังรูปที่ 8.26

```
Router4#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 8.26 ping จาก Router4 ไปยัง Router2 สำเร็จ

11. บน Router1 แสดงตารางเร้าติ้ง

จากรูปที่ 8.24 จะเห็นว่าเน็ตเวิร์ควง 10.1.1.0 จะเชื่อมต่อกับอินเตอร์เฟสอีเทอร์เน็ต 0 และเน็ตเวิร์ค 172.16.10.0 จะเชื่อมต่อกับอินเตอร์เฟสซีเรียล 0 เน็ตเวิร์คทั้ง 2 วงจะมีซับเน็ตเป็น 255.255.255.0 ทั้งคู่ ดังรูปที่ 8.27

```
Router1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA
       E1 - OSPF external type 1, E2 - OSPF externa
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS le
       U - per-user static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial0
```

รูปที่ 8.27 ตารางเร้าติ้งบน Router1

12. เมื่อต้องการดูข้อมูลของโพรโทคอล RIP ที่ส่งระหว่างตัวเราเตอร์ให้ใช้คำสั่ง show ip protocols

Router1#*show ip protocols* ดังรูปที่ 8.28

```
Router1#sh ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 29 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing:   rip
  Default version control: send version 1, receive any version
    Interface         Send  Recv   Key-chain
    Serial0             1    1 2
    Ethernet0           1    1 2
  Routing for Networks:
    10.0.0.0
    172.16.0.0
  Routing Information Sources:
  Distance: (default is 120)
```

รูปที่ 8.28 แสดงคำสั่ง show ip ptotocols ที่ Router1

สังเกตว่า RIP จะส่งข้อมูลข่าวสารไปให้เพื่อนบ้านทุก ๆ 30 วินาที ถ้าไม่มีการส่งข้อมูลมาในระยะเวลาเกิน 180 นาที (hold down) เราเตอร์จะลบตารางเร้าติ้งของเน็ตเวิร์คนั้นทิ้ง ค่าของ administrative distance มีค่าเท่ากับ 120

การวิเคราะห์ข้อมูลของโพรโทคอลแบบ RIP

จาก LAB ที่ผ่านมาเป็นการคอนฟิกโพรโทคอล RIP ใน LAB นี้จะทำการตรวจสอบการส่งข้อมูลที่เกิดขึ้นโดยโพรโทคอล RIP ว่าข้อมูลที่ส่งนั้นประกอบไปด้วยอะไรบ้าง เพื่อเป็นประโยชน์ในกรณีที่จะต้องหาจุดบรกพร่องที่เกิดขึ้นบนเน็ตเวิร์คว่าเกิดขึ้นจากสาเหตุใด โดยการใช้คำสั่ง debug

จุดมุ่งหมาย : เรียนรู้การใช้คำสั่ง debug บนโพรโทคอลแบบ RIP

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 3 ตัวคือ Router1, Router2, Router4 ใช้รูปที่ 8.23

การสร้าง Network Map : ค่าของไอพีแอดเดรสของเราเตอร์ดังตารางที่ 8.4

ตารางที่ 8.4 หมายเลขไอพีแอดเดรสของเราเตอร์

	Router1	Router2	Router4
Interface Ethernet 0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Interface Serial 0	172.16.10.1 255.255.255.0		172.16.10.2 255.255.255.0

หน้าต่าง Simulator :

1. การกำหนดหมายเลขไอพีแอดเดรสแต่ละตัวให้ดูได้จากรูปที่ 8.23

   ที่ Router 1 ให้ใช้คำสั่ง debug ip rip ในโหมด privilege

   Router1>*enable*

Router1#*debug ip rip*

```
Router1#debug ip rip
RIP protocol debugging is on
RIP: sending update to 255.255.255.255 via Serial0 (172.16.10.1)
 subnet 10.1.1.0, metric 1

RIP: sending update to 255.255.255.255 via Ethernet0 (10.1.1.1)
 subnet 172.16.10.0, metric 1

RIP: received update from 10.1.1.2 on Ethernet0

RIP: received update from 172.16.10.2 on Serial0
```

รูปที่ 8.29 แสดงคำสั่ง debug ip rip ที่ Router1

จากรูปข้อมูลที่ส่งกันระหว่างเราเตอร์ที่ทำงานกับโพรโทคอล RIP จะประมาณ 60 วินาที

2. ที่ Router1 ต้องการปิดการใช้งานโพรโทคอล RIP โดยใช้คำสั่ง no debug ip rip หรือ undebug all

Router1#*no debug ip rip* [หยุดการทำงานโพรโทคอล RIP]

Router1#*undebug all* [เป็นคำสั่งอีกคำสั่งหนึ่งที่หยุดการดีบักทั้งหมดบนเราเตอร์]

💻 การคอนฟิกโพรโทคอลแบบ IGRP

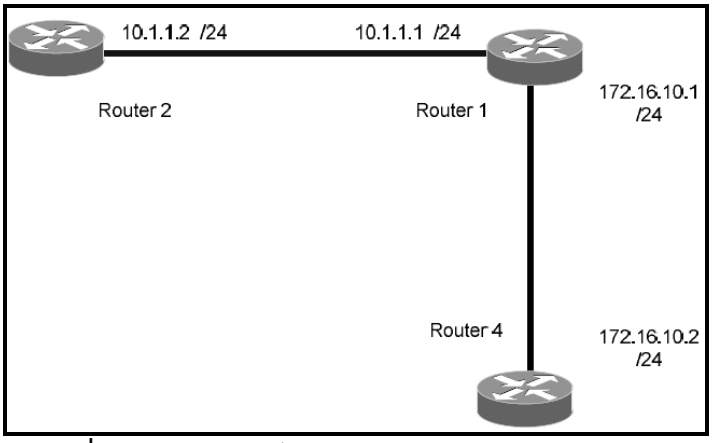
Interior Gate Way Routing Portocols (IGRP) เป็นโพรโทคอลที่มีพื้นฐานมาจาก Distance Vector คือ การใช้ค่าของ hop คล้ายกับ RIP แต่ IGRP มีความแตกต่างออกไป โดยมันจะนำเอาค่าของ Metric คือค่าของ แบนด์วิดและค่าของความล่าช้าของการส่งข้อมูล (Delay) ที่เกิดขึ้นระหว่างการเชื่อมต่อมาคิดด้วย ค่าของ Metric ยังรวมไปถึงค่าของอัตราการส่งข้อมูลได้สูงสุดในสาย (MTU), ความน่าเชื่อถือของสาย (Reliability) และโหลดในสายสัญญาณ (load) บนเน็ตเวิร์คที่ทำงานด้วยโพรโทคอล IGRP จะส่งข้อมูลของตารางเร้าติ้งมีระยะเวลาประมาณ 90 วินาที เมื่อเราเตอร์ได้รับข้อมูลที่ส่งมาให้ มันจะทำการปรับปรุงข้อมูลในตารางเร้าติ้งของตนเองและส่งข้อมูลที่ปรับปรุงแล้วออกไปให้พื่อนบ้านอื่น ๆ ต่อไป กระบวนการปรับปรุงตารางเร้าติ้งตั้งแต่เริ่มต้นถึงสิ้นสุดทุก ๆ ตัวในเน็ตเวิร์คจะเรียกว่า Convergence การคอนฟิก IGRP จะต้องใช้ AS (Autonomous System) เข้ามาเกี่ยวข้องด้วย ซึ่งสามารถอ่านรายละเอียดได้จากบทก่อน ๆ

จุดมุ่งหมาย : เรียนรู้การคอนฟิกโพรโทคอลแบบ IGRP

1. สามารถกำหนดชื่อของเราเตอร์และทำให้อินเตอร์เฟสสามารถทำงานได้
2. สามารถคอนฟิกโพรโทคอล IGRP
3. สามารถเชื่อมต่อเราเตอร์ทั้งหมดเข้าด้วยกันได้
4. สามารถตรวจสอบตารางเร้าติ้งได้
5. สามารถตรวจสอบข้อมูลของโพรโทคอล IGRP ได้

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 3 ตัวคือ Router1, Router2, Router4 ดังรูปที่ 8.30

การสร้าง Network Map :



รูปที่ 8.30 แสดงผังเน็ตเวิร์คสำหรับทดลองคอนฟิก IGRP

ตารางที่ 8.5 หมายเลขไอพีแอดเดรสของเราเตอร์ที่ใช้กับ LAB 15

	Router1	Router2	Router4
Interface Ethernet 0	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0	
Interface Serial 0	172.16.10.1 255.255.255.0		172.16.10.2 255.255.255.0

ให้สร้างผังเน็ตเวิร์คตามรูปที่ 8.30 และกำหนดไอพีแอดเดรสเหมือนตารางที่ 8.5 เมื่อคอนฟิกเสร็จแล้วคุณควรจะ ping จาก Router1 ไปยัง Router2 ซึ่งเชื่อมต่อกันด้วยอินเตอร์เฟสอีเทอร์เน็ต 0 และควรจะ ping จาก Router1 ไปยัง Router4 ได้เช่นเดียวกัน

หน้าต่าง Simulator :

บน Router1

1. **Router1>*enable***

   Router1#*config terminal*

   Router1(config)#

2. คอนฟิก Router1 ให้ทำงานด้วยโพรโทคอล IGRP โดยมีหมายเลข AS เท่ากับ 100

   Router1(config)#*router igrp 100*

   Router1(config-router)#

3. กำหนดเน็ตเวิร์คที่เชื่อมต่อโดยตรงกับ Router1 ให้ครบทุกเน็ตเวิร์ค ในที่นี้มี 2 เน็ตเวิร์คคือ 10.1.1.0 และ 172.16.10.0

   Router1(config-router)#*network 10.1.1.0*

   Router1(config-router)#*network 172.16.10.0*

4. บน Router2 ให้ทำการคอนฟิก IGRP ซึ่งมีเน็ตเวิร์คที่เชื่อมต่อกับ Router2 เพียงเน็ตเวิร์คเดียวคือ 10.1.1.0 และมีหมายเลข AS เหมือนกับ Router1 คือ 100

   Router2>*enable*

   Router2#*config terminal*

   Router2(config)#

   Router2(config)#*router igrp 100*

   Router2(config-router)#

   Router2(config-router)#*network 10.1.1.0*

5. บน Router4 ให้ทำการคอนฟิก IGRP ซึ่งมีเน็ตเวิร์คที่เชื่อมต่อกับ Router4 เพียงเน็ตเวิร์คเดียวคือ 172.16.10.0 และมีหมายเลข AS เหมือนกับ Router1 คือ 100

   Router4>*enable*

   Router4#*config terminal*

   Router4(config)#

   Router4(config)#*router igrp 100*

   Router4(config-router)#

   Router4(config-router)#*network 172.16.10.0*

6. ถึงขั้นตอนนี้เราเตอร์ทุกตัวจะต้องสามารถเชื่อมต่อเข้าหากันได้ทั้งหมดแล้ว ทดลองโดยใช้คำสั่ง ping จาก Router2 ไปยัง Router4 ที่ขาอินเตอร์เฟสซีเรียล 0 ไอพีแอดเดรส 172.16.10.2

   Router2#*ping 172.16.10.2* ดังรูปที่ 8.31

```
Router2#ping 172.16.10.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 8.31 ทดสอบ IGRP ด้วยการ ping

7. ทดลองโดยใช้คำสั่ง ping จาก Router4 ไปยัง Router2 ที่ขาอินเตอร์เฟสอีเทอร์เน็ต 0 ไอพีแอดเดรส 10.1.1.2

   Router4#*ping 10.1.1.2* ดังรูปที่ 8.32

8. เมื่อทดสอบ ping แล้วไม่สำเร็จให้กลับไปคอนฟิกในขั้นตอนต้น ๆ ให้ครบ จากนั้นที่ Router4 ให้ทดสอบการตารางเร้าติ้งของโพรโทคอล IGRP ด้วยคำสั่ง show ip route

```
Router4#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobi
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF in
       E1 - OSPF external type 1, E2 - OSPF external type 2,
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * -
       U - per-user static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial0
     10.0.0.0/24 is subnetted, 1 subnets
I       10.1.1.0 [100/651] via 172.16.10.1, 00:05:44, Serial0
```

รูปที่ 8.32 แสดงตารางเร้าติ้งของโพรโทคอล IGRP

จากรูปที่ 8.32 บน Router4 จะมี 2 เน็ตเวิร์ค โดยเน็ตเวิร์ค 172.16.10.0 จะต่อโดยตรงกับตัวมันเองและเน็ตเวิร์ค 10.1.1.0 จะเชื่อมต่อกับอินเตอร์เฟสซีเรียล 0 ที่ Router1 ค่า AD/Cost คือ [100/651] ส่วนฟิลด์ 00:05:44 บอกถึงข้อมูลของเน็ตเวิร์คนี้ได้รับมาแล้วเป็นเวลา 5 นาที 44 วินาที

9. เมื่อต้องการว่าเราเตอร์ ทำงานด้วยเร้าติ้งโพรโทคอลชนิดใดให้ใช้คำสั่ง show ip protocols Router4#*show ip protocols* ดังรูปที่ 8.33

```
Router4#sh ip protocols
[redacted]
  Sending updates every 90 seconds, next due in 51 seconds
  Invalid after 270 seconds, hold down 280, flushed after 630
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  IGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  IGRP maximum hopcount 100
  IGRP maximum metric variance 1
  Redistributing: igrp 100
  Routing for Networks:
    172.16.0.0
  Routing Information Sources:
    172.16.10.1           100      00:00:06
  Distance: (default is 100)
```

รูปที่ 8.33 แสดงคำสั่ง show ip protocols

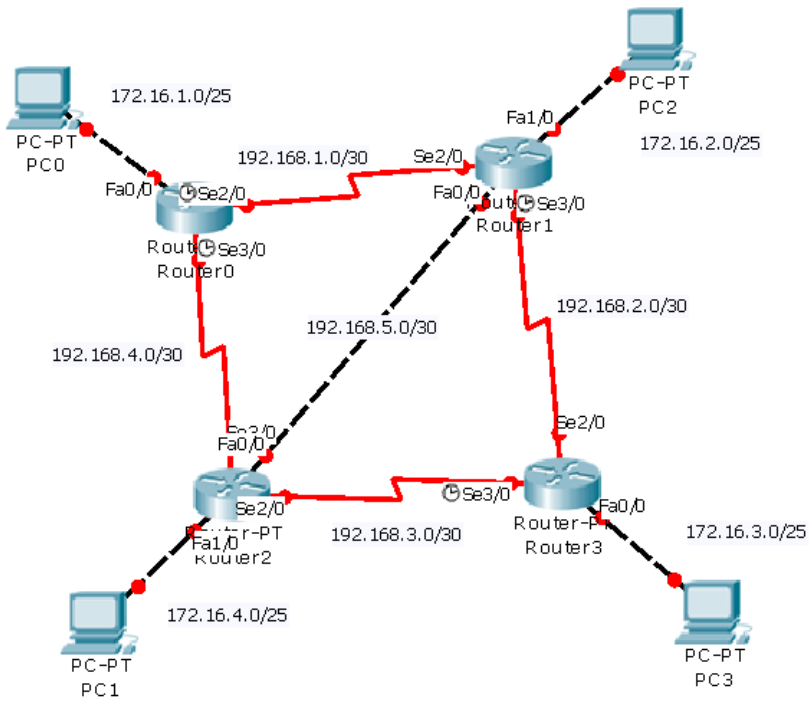
แบบฝึกหัดท้ายบท

ตอนที่ 1

1. การค้นหาเส้นทางคืออะไร
2. องค์ประกอบการค้นหาเส้นทางมีอะไรบ้าง
3. default route คืออะไร
4. Autonomous system (AS) คืออะไรและมีความสำคัญอย่างไร
5. ประเภทของโพรโทคอลค้นหาเส้นทางมีอะไรบ้าง
6. จงอธิบายหลักการทำงานของโพรโทคอล RIP มาพอเข้าใจ
7. จงอธิบายหลักการทำงานของโพรโทคอล IGRP มาพอเข้าใจ

8. จงอธิบายหลักการทำงานของโพรโทคอล OSPF มาพอเข้าใจ
9. ให้คอนฟิกเราเตอร์ตามรูปที่ 8.30โดยใช้โพรโทคอล IGRP และไอพีแอดเดรสเหมือนใน ตารางที่ 8.7 ของ LAB IGMP
10. หลังจากคอนฟิกแล้วให้ทำการ ping ทดสอบการเชื่อมต่อกันระหว่างเราเตอร์ว่าเชื่อมต่อกันได้หรือไม่?_____
11. เมื่อเชื่อมต่อเรียบร้อยแล้ว ให้ทำการคอนฟิกเราเตอร์ทุก ๆ ตัวทำงานด้วยโพรโทคอล IGRP โดยกำหนดให้มีหมายเลข AS เท่ากับ 100?_____
12. จงแสดงคำสั่งที่ใช้สำหรับแสดงตารางเร้าติ้งของเราเตอร์แต่ละตัว?_____
13. จงแสดงคำสั่งที่ใช้สำหรับแสดงชนิดของเร้าติ้งโพรโทคอลที่เราเตอร์แต่ละตัวทำงาน?_____

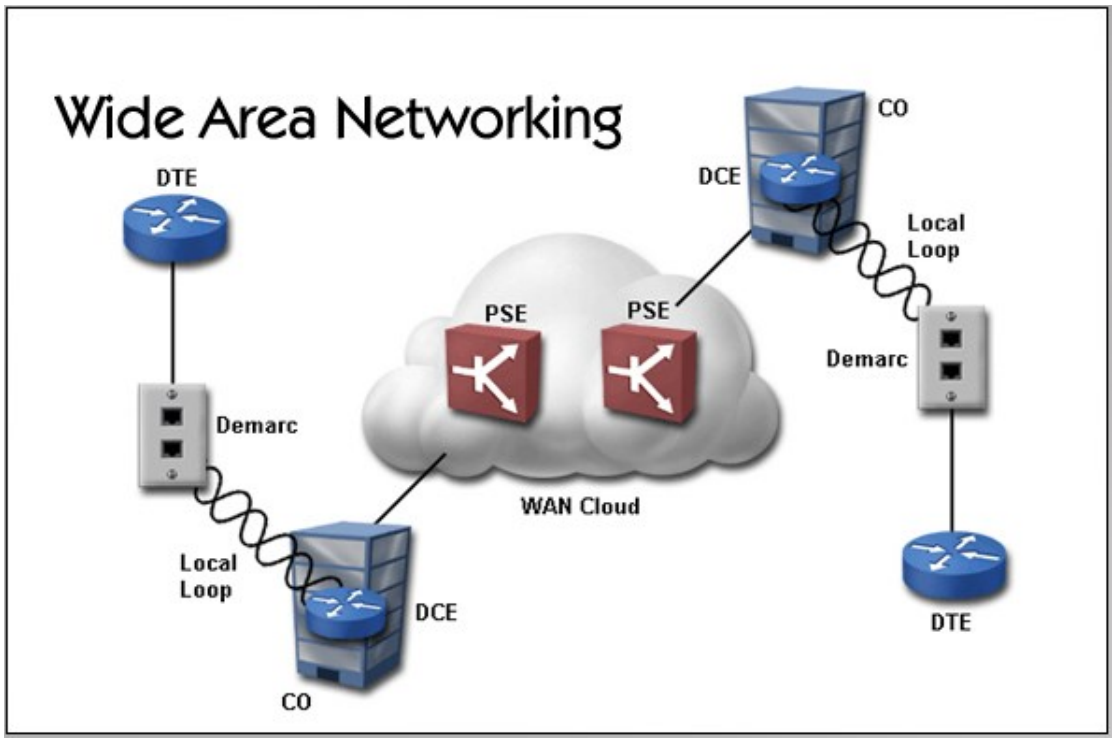
ตอนที่ 2



รูปที่ 1

1. จากรูปที่ 1 ให้นิสิตทำการออกแบบเครือข่าย โดยการคำนวณ subnet, config IP address, IP Route, Default Route ให้สามารถทำการ ping ได้ทุกๆ จุดบนเครือข่าย โดยใช้การ routing แบบ static route เท่านั้น
2. รูปที่ 1 ให้นิสิตทำการออกแบบเครือข่าย โดยใช้การ routing แบบ dynamic route (RIP) เท่านั้น

บทที่ 9
ระบบเครือข่ายบริเวณกว้าง (Wide Area Networking)



- WAN Concepts

แนวคิด

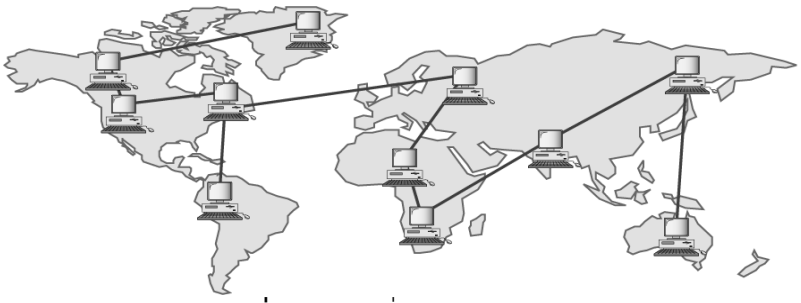
ในบทนี้จะมาเรียนรู้วิธีการเชื่อมต่อระบบเครือข่ายภายใน (Intranet) เข้ากับระบบเครือข่ายภายนอก (Internet) ด้วยวิธีการเชื่อต่อแบบต่างๆ

วัตถุประสงค์

1. เพื่อให้ทราบถึงขั้นตอนการเชื่อมต่อระบบเครือข่ายภายในกับภายนอกเข้าด้วยกัน
2. เพื่อให้ทราบถึงโพรโทรคอลที่นิยมใช้สำหรับการเชื่อมต่อระบบเครือข่ายระดับ WAN

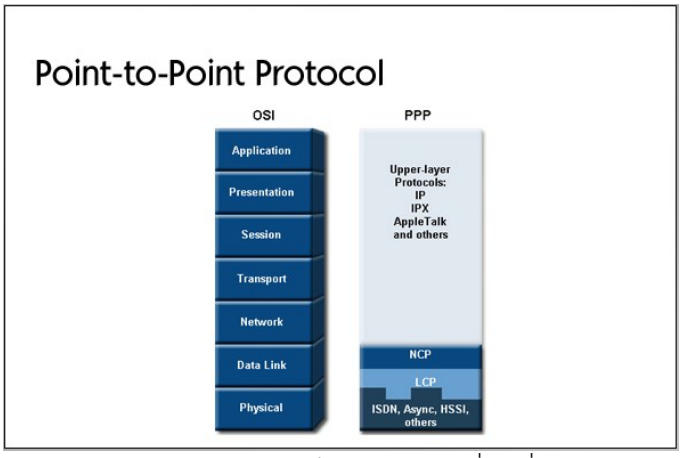
WAN (Wide Area Network)

เป็นระบบที่มีขอบเขตการใช้งานกว้างไกลกว่าระบบแลน ซึ่งอาจกล่าวได้ว่าเป็นระบบที่ไร้ขอบเขตแล้ว เช่นระบบการสื่อสารข้อมูลผ่านดาวเทียมของสถานีโทรทัศน์ต่างๆ แต่การที่จะเชื่อมต่อเครือข่ายที่มีระยะห่างกันมาก ๆ ให้เป็นเครือข่ายเดียวกันทั้งหมดนั้น จำเป็นต้องอาศัยเครือข่ายสาธารณะ (Public Networks) ที่ให้บริการการสื่อสาร โดยเชื่อมต่อกับโมเด็มผ่านเครือข่ายโทรศัพท์สาธารณะ (Public Switching Telephone Network PSTN) ซึ่งมีทั้งลักษณะที่ต้องมีการเชื่อมต่อก่อน (Dial-up) หรือเชื่อมต่อแบบตายตัว เช่น สายเช่า (Lease Line) ดังรูปที่ 9.1



รูปที่ 9.1 การเชื่อมต่อแบบ WAN

การคอนฟิกโพรโทคอลแบบ PPP และ CHAP



PPP (Point-to-Point Protocol) เป็นโพรโทคอลที่ใช้เชื่อมต่อคอมพิวเตอร์ 2 เครื่องเข้าด้วยกันผ่านทางอินเตอร์เฟสชนิดซีเรียล รูปแบบการใช้งานที่นิยมคือ การเชื่อมต่อเครื่องคอมพิวเตอร์พีซีเชื่อมต่อผ่านสายโทรศัพท์เข้าไปยังเซิร์ฟเวอร์ที่ให้บริการหรือผู้ให้บริการอินเตอร์เน็ต PPP ถูกออกแบบให้สนับสนุนการทำงานกับโพรโทคอลได้หลาย ๆ อย่าง เช่น TCP/IP โดยการขี่ไปบน PPP ตามมาตรฐานของ OSI โมเดล PPP จะอยู่ที่เลเยอร์ที่ 2 (Data Link) การส่งข้อมูลเป็นแบบ full-duplex สามารถใช้งานได้กับสายนำสัญญาณได้หลายชนิดตัวอย่างเช่น สายคู่ตีเกลียว สายใยแก้วนำแสง และ สัญญาณผ่านดาวเทียมเป็นต้น

CHAP (Challenge-Handshake Authentication Protocol) เป็นกลไกในการพิสูจน์ตัวตนที่มีความปลอดภัย วิธีการที่ CHAP ใช้ในการพิสูจน์ตัวตนมีขั้นตอนดังนี้

1. เมื่อผ่านการเชื่อมต่อของ Link แล้ว ผู้ให้บริการจะสร้างข้อมูลอย่างหนึ่งที่เรียกว่า challenge message ไปยังเครื่องของผู้ต้องการใช้บริการ
2. เมื่อผู้ใช้บริการรับ challenge message ไปแล้วมันจะคำนวณข้อมูลที่ได้โดยใช้อัลกอริทึมที่เรียกว่า one-way hash แล้วส่งกลับไปให้ผู้ให้บริการ
3. ผู้ให้บริการก็จะคำนวณโดยใช้ one-way hash เหมือนกันแล้วเปรียบเทียบค่าที่ได้รับมาจากผู้ใช้บริการว่าตรงกันหรือไม่ ถ้าตรงกันก็จะยอมรับ ถ้าไม่ใช่ก็จะหยุดการเชื่อมต่อ

การพิสูจน์ตัวตนชนิด CHAP จะให้ความปลอดภัยมากกว่า PAP

จุดมุ่งหมาย : เรียนรู้และเข้าใจการทำงานของการเชื่อมต่อแบบ PPP และการพิสูจน์ตัวตนโดยใช้โพรโทคอลแบบ CHAP

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัวคือ Router1, Router4 ตามผังเน็ตเวิร์ครูปที่ 8.23

การสร้าง Network Map : สามารถใช้ผังเน็ตเวิร์คของรูปที่ 8.23 ได้โดยคอนฟิกเฉพาะ Router1 และ Router4 เท่านั้น

หน้าต่าง Simulator :

บน Router1

1. **Router>*enable***
   Router#*config terminal*
   Router(config)#*hostname Router1*
   Router1(config)#
2. อีนาเบิล secret password เพื่อใช้ในการส่งรหัสผ่านไปให้เราเตอร์ตัวที่เชื่อมต่อด้วยซึ่งจะใช้กับ CHAP รหัสผ่านนี้จำเป็นต้องเหมือนกันทั้ง 2 ฝ่าย สมมุติว่ารหัสผ่านกำหนดเป็น someone
   Router1(config)#*enable secret someone*
3. บน Router1 ให้สร้างรายชื่อผู้ใช้งานของเราเตอร์ที่เชื่อมต่อด้วย ในที่นี้คือ Router4 รหัสผ่านคือ myboson
   Router1(config)#*username Router4 password myboson*
4. บน Router1 ให้กำหนดหมายเลขไอพีแอดเดรสเป็น 10.1.1.1 255.255.255.0 ที่อินเตอร์เฟสซีเรียล 0
   Router1(config)#*interface serial 0*
   Router1(config-if)#*ip address 10.1.1.1 255.255.255.0*
5. บน Router1 ให้ทำการอีนาเบิล PPP บนอินเตอร์เฟสซีเรียล 0
   Router1(config-if)#*encapsulation ppp* [อีนาเบิลโพรโทคอล PPP]
6. บน Router1 ให้ทำการอีนาเบิล CHAP บนอินเตอร์เฟสซีเรียล 0
   Router1(config-if)#*ppp authentication chap* [อีนาเบิลโพรโทคอล CHAP]

7. บน Router1 ให้ทำการอีนาเบิลอินเตอร์เฟสซีเรียล 0 ให้ทำงาน

   Router1(config-if)#*no shutdown* [อีนาเบิลอินเตอร์เฟสให้ทำงาน]

   Router1(config-if)#*exit*

   Router1(config)#

8. บน Router4 ให้คอนฟิกคล้ายกับ Router1 แต่แตกต่างกันที่ Username และรหัสผ่านที่ต้องสลับกันระหว่างเราเตอร์ และหมายเลขไอพีแอดเดรส

ตารางที่ 9.1 ข้อมูลการคอนฟิก PPP และ CHAP

	Router1	Router4
Username	Router4	Router1
Password	Myboson	Someone
Secret Password	Someone	Myboson
IP Address	10.1.1.1 255.255.255.0	10.1.1.2 255.255.255.0

   Router>*enable*

   Router#*config terminal*

   Router(config)#*hostname Router4*

   Router4(config)#

9. กำหนดรหัส secret ให้ตรงกับรหัสผ่านของ Router1

   Router4(config)#*enable secret myboson*

10. กำหนด username เป็นรายชื่อของผู้ใช้บน Router1 และรหัสผ่านให้ตรงกับรหัสผ่าน secret ของ Router1

    Router4(config)#*username Router1 password someone*

11. บน Router4 ให้กำหนดหมายเลขไอพีแอดเดรสเป็น 10.1.1.2 255.255.255.0 ที่อินเตอร์เฟสซีเรียล 0

    Router4(config)#*interface serial 0*

    Router4(config-if)#*ip address 10.1.1.2 255.255.255.0*

12. บน Router4 ให้ทำการอีนาเบิล PPP บนอินเตอร์เฟสซีเรียล 0

    Router4(config-if)#*encapsulation ppp* [อีนาเบิลโพรโทคอล PPP]

13. บน Router4 ให้ทำการอีนาเบิล CHAP บนอินเตอร์เฟสซีเรียล 0

    Router4(config-if)#*ppp authentication chap* [อีนาเบิลโพรโทคอล CHAP]

14. บน Router4 ให้ทำการอีนาเบิลอินเตอร์เฟสซีเรียล 0 ให้ทำงาน

    Router4(config-if)#*no shutdown* [อีนาเบิลอินเตอร์เฟสให้ทำงาน]

Router4(config-if)#*exit*

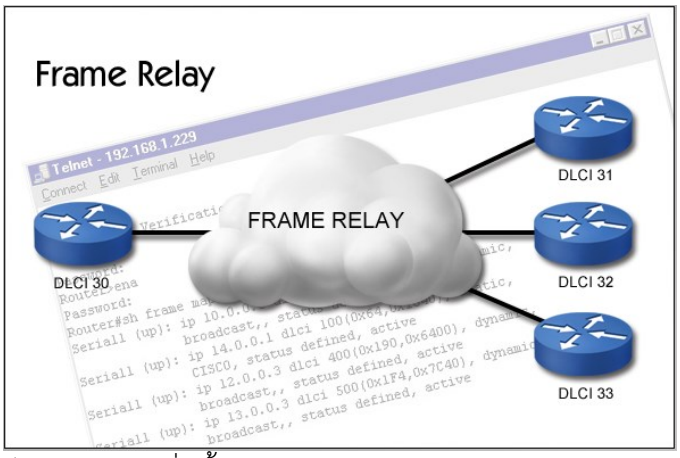
Router4(config)#*exit*

Router4#*ping 10.1.1.1* [ping ทดสอบว่าอินเตอร์เฟสของเราเตอร์ตรงข้ามใช้งานได้หรือไม่] ดังรูปที่ 9.2

```
Router4#ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 9.2 แสดงการ ping เพื่อทดสอบเมื่อคอนฟิก PPP เรียบร้อยแล้ว

การคอนฟิกเฟรมรีเลย์



เฟรมรีเลย์เป็นโพรโทคอลที่มีพื้นฐานคล้าย ๆ กับ X.25 ความแตกต่างหลัก ๆ ของทั้ง 2 คือ การตรวจสอบข้อผิดพลาดและการแก้ไขข้อมูล X.25 จะทำการตรวจสอบและแก้ไขข้อมูลในระดับเน็ตเวิร์คเลเยอร์ มันต้องทำงานทั้ง 2 อย่างจึงทำให้การสื่อสารข้อมูลเกิดการรีเลย์มาก ส่วนเฟรมรีเลย์จะทำการตรวจสอบข้อมูลเท่านั้นไม่ได้แก้ไขข้อมูลด้วย และเฟรมรีเลย์ใช้ใช้ฮาร์ดแวร์สำหรับตรวจสอบข้อมูลจึงทำให้ประสิทธิภาพของเฟรมรีเลย์ดีกว่า X.25 มาก

คำสั่งที่เกี่ยวข้องสำหรับใช้คอนฟิกกับเฟรมรีเลย์

Virtual Circuit เฟรมรีเลย์จะส่งเฟรมข้อมูลระหว่างต้นทางและปลายทางด้วย Virtual Circuit วงจรแบบพิเศษนี้จะมีได้ 2 แบบคือ Permanent Virtual Circuit (PVC) หรือ Switched Virtual Circuit (SVC) สำหรับ Lab ที่ทำการทดลองจะเป็นแบบ PVC เป็นส่วนมาก วงจรชนิดนี้จะสร้างเส้นทางขึ้นมาอย่างถาวร โดยผู้ให้บริการเครือข่ายแบบเฟรมรีเลย์จะเป็นผู้กำหนดให้ โดยแต่ละด้านของ PVC จะต้องมีการระบุค่าตัวเลขค่าหนึ่งขึ้นมาเรียกว่า DLCI (Data Link Connection Identifier) เป็นแอดเดรสที่มีขนาด 10 บิต โดยมันจะถูกแมปเข้ากับแอดเดรสปลายทางในเลเยอร์ 3 LMI เป็นมาตรฐานที่ใช้สำหรับการส่งสัญญาณควบคุม ให้การสื่อสารกันระหว่างเราเตอร์กับเฟรมรีเลย์

สวิตซ์ที่อยู่ใกล้มันมากที่สุด ข้อมูลที่ได้รับจาก LMI ทำให้เราเตอร์ทราบสถานะและข้อมูลเกี่ยวกับเฟรมรีเลย์เน็ตเวิร์คได้

encapsulation frame-relay [cisco|ietf] กำหนดรูปแบบของการห่อหุ้มข้อมูล

frame-relay interface dlci [broadcast] เป็นคำสั่งที่ใช้ระบุหมายเลขของ dlci เพื่อใช้สำหรับโต้ตอบกันของซับอินเตอร์เฟส หมายเลขของ dlci จะถูกกำหนดให้กับทุก ๆ PVC (คือการสร้างวงจรเสมือน) พูดง่าย ๆ คือ dlci จะเป็นหมายเลขใดหมายเลขหนึ่งที่สร้างขึ้นมาสำหรับกำหนดให้กับ PVC สำหรับใช้อ้างถึงกันว่าเป็นวงจรเสมือนอันไหน broadcast ใช้สำหรับเมื่อต้องการส่งข้อมูลแบบกระจายไปทุก ๆ เน็ตเวิร์ค

frame-relay lmi-type เป็นโพรโทคอลที่ใช้สำหรับควบคุมการทำงานของเฟรมรีเลย์

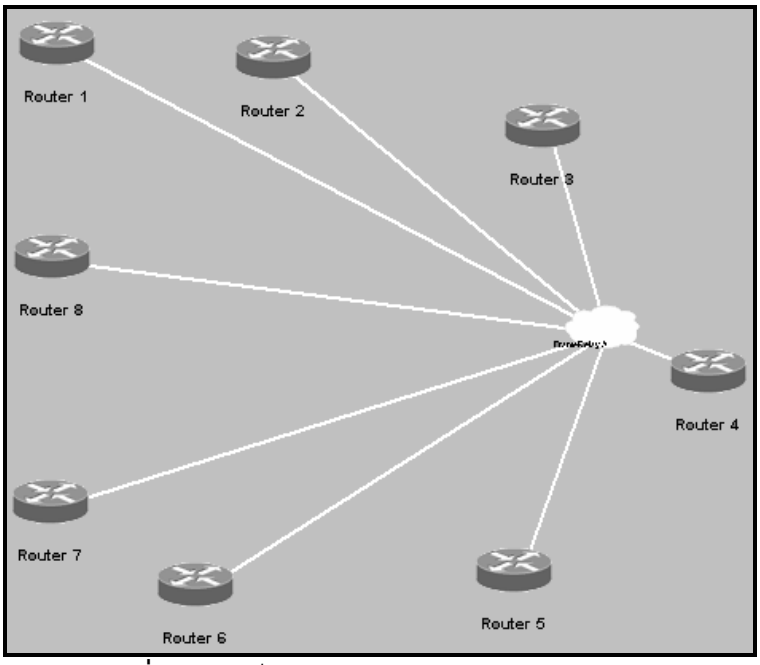
frame-relay map-ip ip-address dlci [broadcast] เป็นคำสั่งที่ใช้สำหรับเชื่อมต่อเฟรมรีเลย์ที่มีลักษณะเป็น Multipoint

interface serial0.subinterface [point-to-point] เป็นคำสั่งที่ต้องการสร้าง logical อินเตอร์เฟสหลาย ๆ อินเตอร์เฟสไว้ใน physical อินเตอร์เฟสเดียว

จุดมุ่งหมาย : เข้าใจการทำงานว่าเฟรมรีเลย์สร้างการเชื่อมต่อกันอย่างไร

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัวคือ Router1 และ Router 2 ดังรูปที่ 9.2 และตารางที่ 9.2

การสร้าง Network Map :



รูปที่ 9.2 ผังเน็ตเวิร์คสำหรับการคอนฟิกเฟรมรีเลย์

ตารางที่ 9.2 ผังเน็ตเวิร์คเฟรมรีเลย์

อุปกรณ์	การเชื่อมต่อ
Router 1 (1005)	Router 1(Serial 0) → Router 2(Serial 0),Router 3(Serial 0)

IP = 10.1.1.1 255.255.255.0	Router 4(Serial 0),Router 5(Serial 0) Router 6(Serial 0),Router 7(Serial 0) Router 8(Serial 0)
Router 2 (1005) IP = 10.1.1.2 255.255.255.0	Router 2(Serial 0) → Router 1(Serial 0),Router 3(Serial 0) Router 4(Serial 0),Router 5(Serial 0) Router 6(Serial 0),Router 7(Serial 0) Router 8(Serial 0)
Router 3 (1005)	Router 3(Serial 0) → Router 1(Serial 0),Router 2(Serial 0) Router 4(Serial 0),Router 5(Serial 0) Router 6(Serial 0),Router 7(Serial 0) Router 8(Serial 0)
Router 4 (1005)	Router 4(Serial 0) → Router 1(Serial 0),Router 2(Serial 0) Router 3(Serial 0),Router 5(Serial 0) Router 6(Serial 0),Router 7(Serial 0) Router 8(Serial 0)
Router 5 (1005)	Router 5(Serial 0) → Router 1(Serial 0),Router 2(Serial 0) Router 3(Serial 0),Router 4(Serial 0) Router 6(Serial 0),Router 7(Serial 0) Router 8(Serial 0)
Router 6 (1005)	Router 6(Serial 0) → Router 1(Serial 0),Router 2(Serial 0) Router 3(Serial 0),Router 4(Serial 0) Router 5(Serial 0),Router 7(Serial 0) Router 8(Serial 0)
Router 7 (1005)	Router 7(Serial 0) → Router 1(Serial 0),Router 2(Serial 0) Router 3(Serial 0),Router 4(Serial 0) Router 5(Serial 0),Router 6(Serial 0) Router 8(Serial 0)
Router 8 (1005)	Router 8(Serial 0) → Router 1(Serial 0),Router 2(Serial 0) Router 3(Serial 0),Router 4(Serial 0) Router 5(Serial 0),Router 6(Serial 0) Router 7(Serial 0)

หน้าต่าง Simulator :

1. บนเราเตอร์ 1 เข้าสู่โหมดการคอนฟิก

   Router>*enable*

   Router#*config terminal*

   Router(config)#

2. ให้เปลี่ยนชื่อของเราเตอร์เป็น R1

   Router(config)#*hostname R1*

3. กำหนดไอพีแอดเดรสให้กับอินเตอร์เฟสซีเรียล 0 ของ R1 เป็น 10.1.1.1 255.255.255.0 และสั่งให้อินเตอร์เฟสนี้ทำงาน

   R1(config)#*interface serial 0*

   R1(config-if)#*ip address 10.1.1.1 255.255.255.0*

   R1(config-if)#*no shutdown*

4. บนเราเตอร์ 2 ให้เปลี่ยนชื่อเป็น R2 กำหนดไอพีแอดเดรสเป็น 10.1.1.2 255.255.255.0 ให้กับอินเตอร์เฟสซีเรียล 0 และสั่งให้อินเตอร์เฟสทำงาน

   Router>*enable*

   Router#*config terminal*

   Router(config)#

   Router(config)#*hostname R2*

   R2(config)#*interface serial 0*

   R2(config-if)#*ip address 10.1.1.2 255.255.255.0*

   R2(config-if)#*no shutdown*

5. เราเตอร์ทั้งสองจะยังไม่สามารถเชื่อมต่อกันได้ (มีสถานะ down อยู่) เราจะต้องมีการกำหนดวิธีการสื่อสารให้ทั้งสองเราเตอร์เข้าใจเสียก่อน เริ่มต้นด้วยการกำหนดรูปแบบของ encapsulation ให้กับเราเตอร์ 1

   R1(config-if)#*encapsulation frame-relay*

6. ขั้นต่อไปให้ทำการเซ็ตค่าของเฟรมรีเลย์อินเตอร์เฟสที่ใช้สำหรับเชื่อมต่อระหว่างเราเตอร์ 1 และเราเตอร์ 2 โดยใช้หมายเลขดีฟอลท์ของ DLCI เป็น 102

   R1(config-if)#*frame-relay interface-dlci 102*

7. บนเราเตอร์ 2 ก็เช่นเดียวกันให้ทำการกำหนดค่า encapsulation, dlci = 201

   R2(config-if)#*encapsulation frame-relay*

   R2(config-if)#*frame-relay interface-dlci 201*

8. ถึงขั้นตอนนี้ที่ขาอินเตอร์เฟสของทั้งสองเราเตอร์คือ R1 และ R2 จะมีสถานะเป็น Active โดยผ่านวงจรของเฟรมรีเลย์ ให้ทดสอบด้วยการ ping จากเราเตอร์ R1 ไปยัง R2 ด้วย ping 10.1.1.2

R1#*ping 10.1.1.2* ดังรูปที่ 9.3

```
R1#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

รูปที่ 9.3 ทดสอบการทำงานของเฟรมรีเลย์ด้วยการ ping

9. นอกจากการทดสอบด้วยคำสั่ง ping แล้วยังมีคำสั่งที่แสดงสถานการณ์ทำงานของเฟรมรีเลย์อีกหลายคำสั่ง คำสั่งแรกที่เราจะทดสอบคือคำสั่ง show frame-relay lmi คำสั่งนี้จำใช้แสดงการแลกเปลี่ยนข้อมูลระหว่างเราเตอร์กับวงจรรีเลย์สวิตช์

   R1#*show frame-relay lmi* ดังรูปที่ 9.4

```
R1#show frame-relay lmi
show frame-relay lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = CISCO
Invalid Unnumbered info 0              Invalid Prot Disc 0
Invalid dummy Call Ref 0               Invalid Msg Type 0
Invalid Status Message 0               Invalid Lock Shift 0
Invalid Information ID 0               Invalid Report IE Len 0
Invalid Report Request 0               Invalid Keep IE Len 0
Num Status Enq. Sent 160               Num Status msgs Rcvd 137
Num Update Status Rcvd 0               Num Status Timeouts 13
```

รูปที่ 9.4 แสดงคำสั่ง show frame-relay lmi

10. คำสั่งต่อไปจะเป็นการแสดงปริมาณการแลกเปลี่ยนข้อมูลระหว่างเราเตอร์กับวงจรของเฟรมรีเลย์ โดยใช้คำสั่ง show frame-relay traffic

    R1#*show frame-relay traffic* ดังรูปที่ 9.5

```
R1#show frame-relay traffic
show frame-relay traffic
Frame Relay statistics:
        ARP requests sent 0, ARP replies sent 0
        ARP request recvd 0, ARP replies recvd 0
```

รูปที่ 9.5 แสดงการใช้คำสั่ง show frame-relay traffic

11. คำสั่งต่อไปจะเป็นการแสดงการแมปของเลเยอร์ 2 (DLCI) ไปเป็นเลเยอร์ 3 (ซึ่งเป็นไอพีแอดเดรส) ของเราเตอร์ ซึ่งขั้นตอนต่อ ๆ ไปเราจะใช้คำสั่งนี้ในการสร้างแมปเองได้

    R1#*show frame-relay map* ดังรูปที่ 9.6

```
R1#sh frame-relay map
Serial0 (up): ip 10.1.1.2 dlci 102(0x66,0x1860), dynamic,
              broadcast,CISCO, status defined, active
```

รูปที่ 9.6 แสดงการใช้คำสั่ง show frame-relay map

จากรูปจะเห็นว่าจะเชื่อมต่อจากเราเตอร์ 1 ไปยังเราเตอร์ 2 ผ่านเฟรมรีเลย์ที่มีหมายเลข dlci เป็น 102

12. คำสั่งต่อไปจะเป็นการแสดงข้อมูลของเราเตอร์กับวงจรของเฟรมรีเลย์สวิตช์ PVC (Virtual Circuit) ซึ่งเป็นแบบโลคอล คือการมองการเชื่อมต่อแค่เราเตอร์กับเฟรมรีเลย์สวิตช์เท่านั้น

    R1#*show frame-relay pvc* ดังรูปที่ 9.7

```
R1#sh frame-relay pvc
PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL,                    INTERFACE = Serial0
```

รูปที่ 9.7 PVC ที่ถูกใช้งานโดย DLCI จะมีสถานะ Active

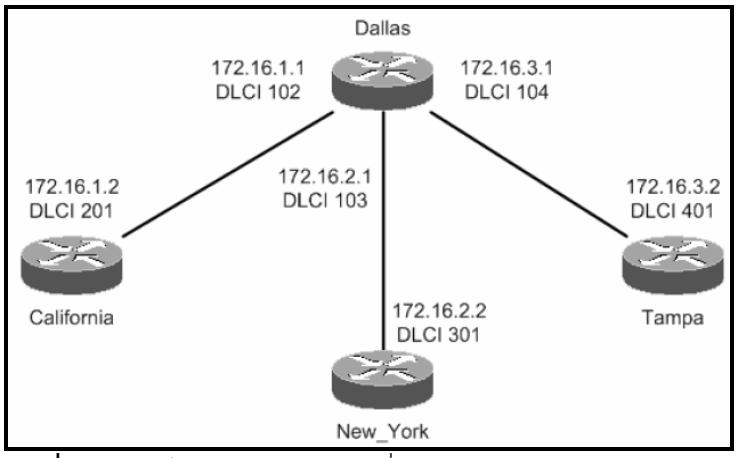
การคอนฟิกเฟรมรีเลย์ชนิด HUB และ SPOKE

วิธีการทั้ง 2 แบบให้นึกภาพง่าย ๆ คือเป็นการเชื่อมต่อในลักษณะที่มีศูนย์กลางอยู่ที่จุดใดจุดหนึ่ง แล้วทำการเชื่อมต่อไปยังสาขาที่อยู่กันต่างเมืองหรือคนละพื้นที่ การเชื่อมต่อแบบนี้จะต้องสร้างอินเตอร์เฟสย่อยขึ้นมาในส่วนของ Office ที่เป็นศูนย์กลางเพื่อรองรับการเชื่อมต่อที่เข้ามาจากสาขาย่อย ๆ

จุดมุ่งหมาย : เข้าใจรูปแบบการเชื่อมต่อของเฟรมรีเลย์แบบ HUB หรือ SPOKE

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 4 ตัวคือ Dallas, Carlifornia, New_York, Tampa ดังรูปที่ 9.8

การสร้าง Network Map :



รูปที่ 9.8 ผังเน็ตเวิร์คสำหรับการเชื่อมต่อแบบ HUB หรือ SPOKE

หน้าต่าง Simulator :

1. บนเราเตอร์ 1 ให้เข้าสู่โหมดคอนฟิกและเปลี่ยนชื่อเป็น Dallas

   Router>*enable*

   Router#*config terminal*

   Router(config)#*hostname Dallas*

   Dallas(config)#

2. บนเราเตอร์ Dallas ให้ทำการคอนฟิกที่อินเตอร์เฟสซีเรียล 0 เป็น encapsulation แบบเฟรมรีเลย์ และสั่งให้ทำงาน

   Dallas(config)#*interface serial 0*

   Dallas(config-if)#*encapsulation frame-relay*

   Dallas(config-if)#*no shutdown*

3. ขั้นตอนนี้จะต้องทำการคอนฟิกอินเตอร์เฟสย่อยของเราเตอร์ที่ทำหน้าที่เป็นหน่วยงานหลักสำหรับรองรับการเชื่อมต่อจากสาขาย่อย (Main Office) ให้เป็นชนิด point-to-point

Dallas(config-if)#*exit*

Dallas(config)#

Dallas(config)#*interface serial 0.100 point-to-point* [สร้างอินเตอร์เฟสย่อยหมายเลข 100 (subinterface) เพื่อรองรับการเชื่อมต่อจากสาขาย่อย]

Dallas(config-subif)#

4. ขั้นต่อไปให้ทำการกำหนดหมายเลข DLCI ให้กับอินเตอร์เฟสย่อยสำหรับใช้เชื่อมต่อไปยังเราเตอร์ California มีค่าไอพีแอดเดรสเป็น 172.16.1.1 255.255.255.0 และ DLCI เท่ากับ 102

Dallas(config-subif)#*ip address 172.16.1.1 255.255.255.0*

Dallas(config-subif)#*frame-relay interface-dlci 102*

5. ขั้นต่อไปให้ทำการกำหนดหมายเลข DLCI ให้กับอินเตอร์เฟสย่อยสำหรับใช้เชื่อมต่อไปยังเราเตอร์ New_York มีค่าไอพีแอดเดรสเป็น 172.16.2.1 255.255.255.0 และ DLCI เท่ากับ 103 แบบ point-to-point

Dallas(config-subif)#*exit*

Dallas(config)#

Dallas(config)#*interface serial 0.200 point-to-point* [สร้างอินเตอร์เฟสย่อยหมายเลข 200 (subinterface) เพื่อรองรับการเชื่อมต่อจากสาขาย่อย]

Dallas(config-subif)#*ip address 172.16.2.1 255.255.255.0*

Dallas(config-subif)#*frame-relay interface-dlci 103*

6. ขั้นต่อไปให้ทำการกำหนดหมายเลข DLCI ให้กับอินเตอร์เฟสย่อยสำหรับใช้เชื่อมต่อไปยังเราเตอร์ Tampa มีค่าไอพีแอดเดรสเป็น 172.16.3.1 255.255.255.0 และ DLCI เท่ากับ 104 แบบ point-to-point

Dallas(config-subif)#*exit*

Dallas(config)#

Dallas(config)#*interface serial 0.300 point-to-point* [สร้างอินเตอร์เฟสย่อยหมายเลข 300 (subinterface) เพื่อรองรับการเชื่อมต่อจากสาขาย่อย]

Dallas(config-subif)#*ip address 172.16.3.1 255.255.255.0*

Dallas(config-subif)#*frame-relay interface-dlci 104*

7. บนเราเตอร์ 2 ให้เปลี่ยนชื่อเป็น California

Router>*enable*

Router#*config terminal*

Router(config)#*hostname California*

California(config)#

8. คอนฟิกอินเตอร์เฟสซีเรียล 0 ให้มี encapsulation เป็นเฟรมรีเลย์และสั่งให้ทำงาน

   California(config)#*interface serial 0*

   California(config-if)#*encapsulation frame-relay*

   California(config-if)#*no shutdown*

9. จากเราเตอร์ของ California ไม่จำเป็นต้องสร้างอินเตอร์เฟสย่อย (subinterface) เพราะว่ามีการเชื่อมต่อเพียงเส้นทางเดียวเท่านั้น ให้ทำการกำหนด DLCI เป็น 201 และไอพีแอดเดรสเป็น 172.16.1.2 255.255.255.0

   California(config-if)#*ip address 172.16.1.2 255.255.255.0*

   California(config-if)#*frame-relay interface-dlci 201*

10. บนเราเตอร์ 3 ให้เปลี่ยนชื่อเป็น New_York

    Router>*enable*

    Router#*config terminal*

    Router(config)#*hostname New_York*

    New_York(config)#

11. คอนฟิกอินเตอร์เฟสซีเรียล 0 ให้มี encapsulation เป็นเฟรมรีเลย์และสั่งให้ทำงาน

    New_York(config)#*interface serial 0*

    New_York(config-if)#*encapsulation frame-relay*

    New_York(config-if)#*no shutdown*

12. จากเราเตอร์ของ New_York ไม่จำเป็นต้องสร้างอินเตอร์เฟสย่อย (subinterface) เพราะว่ามีการเชื่อมต่อเพียงเส้นทางเดียวเท่านั้น ให้ทำการกำหนด DLCI เป็น 301 และไอพีแอดเดรสเป็น 172.16.2.2 255.255.255.0

    New_York(config-if)#*ip address 172.16.2.2 255.255.255.0*

    New_York(config-if)#*frame-relay interface-dlci 301*

13. บนเราเตอร์ 4 ให้เปลี่ยนชื่อเป็น Tampa

    Router>*enable*

    Router#*config terminal*

    Router(config)#*hostname Tampa*

    Tampa(config)#

14. คอนฟิกอินเตอร์เฟสซีเรียล 0 ให้มี encapsulation เป็นเฟรมรีเลย์และสั่งให้ทำงาน

    Tampa(config)#*interface serial 0*

    Tampa(config-if)#*encapsulation frame-relay*

Tampa(config-if)#*no shutdown*

15. จากเราเตอร์ของ Tampa ไม่จำเป็นต้องสร้างอินเตอร์เฟสย่อย (subinterface) เพราะว่ามีการเชื่อมต่อเพียงเส้นทางเดียวเท่านั้น ให้ทำการกำหนด DLCI เป็น 401 และไอพีแอดเดรสเป็น 172.16.3.2 255.255.255.0

    Tampa(config-if)#*ip address 172.16.3.2 255.255.255.0*

    Tampa(config-if)#*frame-relay interface-dlci 401*

16. ถึงขั้นตอนนี้ทุกอินเตอร์เฟสที่เชื่อมต่อกันต้องสามารถติดต่อกันได้แล้ว ให้ทำการทดสอบโดยการใช้คำสั่ง ping จากเราเตอร์ Dallas

    Dallas#*ping 172.16.2.1* [ping ทดสอบไปยังเราเตอร์ California]

    Dallas#*ping 172.16.3.1* [ping ทดสอบไปยังเราเตอร์ New_York]

    Dallas#*ping 172.16.4.1* [ping ทดสอบไปยังเราเตอร์ Tampa]

การเชื่อมต่อทั้งหมดสามารถใช้งานได้แล้ว แต่ยังไม่สามารถส่งข้อมูลได้ ต้องมีการคอนฟิกเร้าติ้งโพรโทคอลที่ใช้ค้นหาเส้นทางให้กับเฟรมรีเลย์อีกครั้งจึงจะทำงานได้อย่างสมบูรณ์

ให้ทดลองใช้คำสั่งต่าง ๆ เหล่านี้แล้วสังเกตค่าที่ได้

show frame-relay map

show frame-relay pvc

show frame-relay lmi

show frame-relay route

show frame-relay traffic

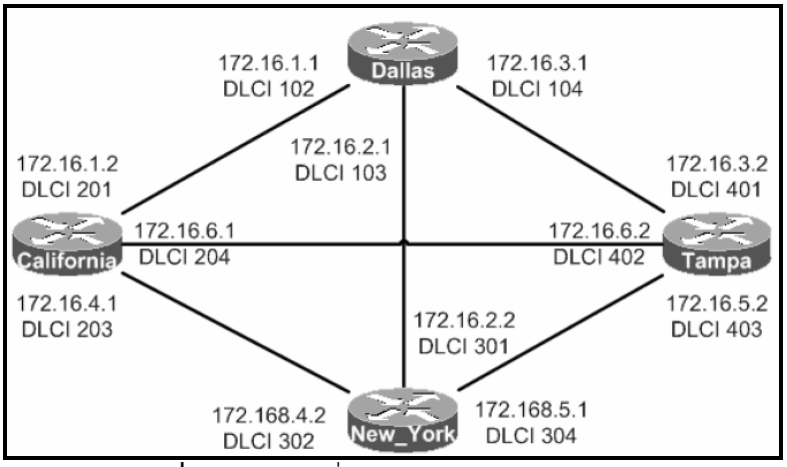
การคอนฟิกเฟรมรีเลย์ชนิด Mesh

ข้อแตกต่างหลัก ๆ ของการเชื่อมต่อแบบ HUB และ SPOKE กับ Mesh คือทุก ๆ หน่วยงานจะมีการเชื่อมต่อถึงกันหมด ซึ่งจะส่งผลให้เกิดความเสถียรภาพมากในกรณีที่มีหน่วยงานหรือสาขาใดสาขาหนึ่งดาวน์ลงไปจะไม่ส่งผลกระทบต่อสาขาอื่น ๆ เพราะสามารถเลือกใช้เส้นทางอื่น ๆ ในการสื่อสารกันได้

จุดมุ่งหมาย : เข้าใจรูปแบบการเชื่อมต่อของเฟรมรีเลย์แบบ Mesh

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 4 ตัวคือ Dallas, Carlifornia, New_York, Tampa ดังรูปที่ 9.9

การสร้าง Network Map :



รูปที่ 9.9 ผังการเชื่อมต่อเฟรมรีเลย์แบบ Mesh

Simulator :

1. ที่เราเตอร์ 1 (Dallas) ที่จะเชื่อมต่อไปยังเราเตอร์ 2 (California) ให้ทำขั้นตอนต่าง ๆ ดังนี้
   - เปลี่ยนชื่อเราเตอร์เป็น Dallas
   - กำหนดให้อินเตอร์เฟสซีเรียล 0 มี encapsulation แบบ frame-relay
   - สั่งให้อินเตอร์เฟสซีเรียล 0 ทำงาน
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.100 ชนิด point-to-point
   - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 100 เป็น 172.16.1.1 255.255.255.0
   - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 100 เป็น 102

   Router>*enable*

   Router#*config terminal*

   Router(config)#*hostname Dallas*

   Dallas(config)#*interface serial 0*

   Dallas(config-if)#

   Dallas(config-if)#*encapsulation frame-relay*

   Dallas(config-if)#*no shutdown*

   Dallas(config-if)#*exit*

   Dallas(config)#*interface serial 0.100 point-to-point*

   Dallas(config-subif)#*ip address 172.1.1.1 255.255.255.0*

   Dallas(config-subif)#*frame-relay interface-dlci 102*

2. ที่เราเตอร์ 1 (Dallas) ที่จะเชื่อมต่อไปยังเราเตอร์ 3 (New_York) ให้ทำขั้นตอนต่าง ๆ ดังนี้
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.200 ชนิด point-to-point

- กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 200 เป็น 172.16.2.1 255.255.255.0
- กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 200 เป็น 103

Dallas(config-subif)#*exit*

Dallas(config)#*interface serial 0.200 point-to-point*

Dallas(config-subif)#*ip address 172.1.2.1 255.255.255.0*

Dallas(config-subif)#*frame-relay interface-dlci 103*

3. ที่เราเตอร์ 1 (Dallas) ที่จะเชื่อมต่อไปยังเราเตอร์ 4 (Tampa) ให้ทำขั้นตอนต่าง ๆ ดังนี้
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.300 ชนิด point-to-point
   - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 300 เป็น 172.16.3.1 255.255.255.0
   - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 300 เป็น 104

   Dallas(config-subif)#*exit*

   Dallas(config)#*interface serial 0.300 point-to-point*

   Dallas(config-subif)#*ip address 172.1.3.1 255.255.255.0*

   Dallas(config-subif)#*frame-relay interface-dlci 104*

4. ขั้นตอนต่อไปให้เข้าไปยังเราเตอร์ 2 ให้ทำการคอนฟิกตามรายละเอียดดังนี้
   - เปลี่ยนชื่อเราเตอร์เป็น California
   - กำหนดให้อินเตอร์เฟสซีเรียล 0 มี encapsulation แบบ frame-relay
   - สั่งให้อินเตอร์เฟสซีเรียล 0 ทำงาน
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.100 ชนิด point-to-point
   - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 100 เป็น 172.16.1.2 255.255.255.0
   - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 100 เป็น 201

   Router>*enable*

   Router#*config terminal*

   Router(config)#*hostname California*

   California(config)#*interface serial 0*

   California(config-if)#

   California(config-if)#*encapsulation frame-relay*

   California(config-if)#*no shutdown*

   California(config-if)#*exit*

   สร้างการเชื่อมต่อจากเราเตอร์ California ไปยังเราเตอร์ Dallas

California(config)#*interface serial 0.100 point-to-point*

California(config-subif)#*ip address 172.1.1.2 255.255.255.0*

California(config-subif)#*frame-relay interface-dlci 201*

5. ที่เราเตอร์ 2 (California) ที่จะเชื่อมต่อไปยังเราเตอร์ 3 (New_York) ให้ทำขั้นตอนต่าง ๆ ดังนี้
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.200 ชนิด point-to-point
   - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 200 เป็น 172.16.4.1 255.255.255.0
   - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 200 เป็น 203

   California(config-subif)#*exit*

   California(config)#*interface serial 0.200 point-to-point*

   California(config-subif)#*ip address 172.1.4.1 255.255.255.0*

   California(config-subif)#*frame-relay interface-dlci 203*

6. ที่เราเตอร์ 2 (California) ที่จะเชื่อมต่อไปยังเราเตอร์ 4 (Tampa) ให้ทำขั้นตอนต่าง ๆ ดังนี้
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.300 ชนิด point-to-point
   - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 300 เป็น 172.16.6.1 255.255.255.0
   - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 300 เป็น 204

   California(config-subif)#*exit*

   California(config)#*interface serial 0.300 point-to-point*

   California(config-subif)#*ip address 172.1.3.1 255.255.255.0*

   California(config-subif)#*frame-relay interface-dlci 204*

7. ขั้นตอนต่อไปให้เข้าไปยังเราเตอร์ 3 ให้ทำการคอนฟิกตามรายละเอียดดังนี้
   - เปลี่ยนชื่อเราเตอร์เป็น New_York
   - กำหนดให้อินเตอร์เฟสซีเรียล 0 มี encapsulation แบบ frame-relay
   - สั่งให้อินเตอร์เฟสซีเรียล 0 ทำงาน
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.100 ชนิด point-to-point
   - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 100 เป็น 172.16.2.2 255.255.255.0
   - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 100 เป็น 301

   Router>*enable*

   Router#*config terminal*

Router(config)#*hostname New_Work*

New_York(config)#*interface serial 0*

New_York(config-if)#

New_York(config-if)#*encapsulation frame-relay*

New_York(config-if)#*no shutdown*

New_York(config-if)#*exit*

สร้างการเชื่อมต่อจากเราเตอร์ New_York ไปยังเราเตอร์ Dallas

New_York(config)#*interface serial 0.100 point-to-point*

New_York(config-subif)#*ip address 172.1.2.2 255.255.255.0*

New_York(config-subif)#*frame-relay interface-dlci 301*

8. ที่เราเตอร์ 3 (New_York) ที่จะเชื่อมต่อไปยังเราเตอร์ 2 (California) ให้ทำขั้นตอนต่าง ๆ ดังนี้
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.200 ชนิด point-to-point
   - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 200 เป็น 172.16.4.2 255.255.255.0
   - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 200 เป็น 302

   New_York(config-subif)#*exit*

   New_York(config)#*interface serial 0.200 point-to-point*

   New_York(config-subif)#*ip address 172.1.4.2 255.255.255.0*

   New_York(config-subif)#*frame-relay interface-dlci 302*
9. ที่เราเตอร์ 3 (New_York) ที่จะเชื่อมต่อไปยังเราเตอร์ 4 (Tampa) ให้ทำขั้นตอนต่าง ๆ ดังนี้
   - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.300 ชนิด point-to-point
   - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 300 เป็น 172.16.5.1 255.255.255.0
   - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 300 เป็น 304

   New_York(config-subif)#*exit*

   New_York(config)#*interface serial 0.300 point-to-point*

   New_York(config-subif)#*ip address 172.1.5.1 255.255.255.0*

   New_York(config-subif)#*frame-relay interface-dlci 304*
10. ขั้นตอนต่อไปให้เข้าไปยังเราเตอร์ 4 ให้ทำการคอนฟิกตามรายละเอียดดังนี้
    - เปลี่ยนชื่อเราเตอร์เป็น Tampa
    - กำหนดให้อินเตอร์เฟสซีเรียล 0 มี encapsulation แบบ frame-relay

- สั่งให้อินเตอร์เฟสซีเรียล 0 ทำงาน
- สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.100 ชนิด point-to-point
- กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 100 เป็น 172.16.3.2 255.255.255.0
- กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 100 เป็น 401

Router>*enable*

Router#*config terminal*

Router(config)#*hostname Tampa*

Tampa(config)#*interface serial 0*

Tampa(config-if)#

Tampa(config-if)#*encapsulation frame-relay*

Tampa(config-if)#*no shutdown*

Tampa(config-if)#*exit*

สร้างการเชื่อมต่อจากเราเตอร์ Tampa ไปยังเราเตอร์ Dallas

Tampa(config)#*interface serial 0.100 point-to-point*

Tampa(config-subif)#*ip address 172.1.3.2 255.255.255.0*

Tampa(config-subif)#*frame-relay interface-dlci 401*

11. ที่เราเตอร์ 4 (Tampa) ที่จะเชื่อมต่อไปยังเราเตอร์ 2 (California) ให้ทำขั้นตอนต่าง ๆ ดังนี้
    - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.200 ชนิด point-to-point
    - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 200 เป็น 172.16.6.2 255.255.255.0
    - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 200 เป็น 402

Tampa(config-subif)#*exit*

Tampa(config)#*interface serial 0.200 point-to-point*

Tampa(config-subif)#*ip address 172.1.6.2 255.255.255.0*

Tampa(config-subif)#*frame-relay interface-dlci 402*

12. ที่เราเตอร์ 4 (Tampa) ที่จะเชื่อมต่อไปยังเราเตอร์ 3 (New_York) ให้ทำขั้นตอนต่าง ๆ ดังนี้
    - สร้างอินเตอร์เฟสย่อยที่ซีเรียล 0 (subinterface) เป็น 0.300 ชนิด point-to-point
    - กำหนดหมายเลขไอพีกับอินเตอร์เฟสย่อย 300 เป็น 172.16.5.2 255.255.255.0
    - กำหนดหมายเลข DLCI ของอินเตอร์เฟสย่อย 300 เป็น 403

Tampa(config-subif)#*exit*

Tampa(config)#*interface serial 0.300 point-to-point*

Tampa(config-subif)#*ip address 172.1.5.2 255.255.255.0*

Tampa(config-subif)#*frame-relay interface-dlci 403*

13. ถึงขั้นตอนนี้ทุกอินเตอร์เฟสที่เชื่อมต่อกันจะต้องสามารถทำงานได้ ทดสอบโดยการใช้คำสั่ง ping จากเราเตอร์ 1 (Dallas) ไปยังเราเตอร์ทุก ๆ ตัวที่เชื่อมต่ออยู่ด้วย

Dallas(config-subif)#*end*

Dallas#*ping 172.16.1.2* [ping ไปยังเราเตอร์ California]

Dallas#*ping 172.16.2.2* [ping ไปยังเราเตอร์ New_York]

Dallas#*ping 172.16.3.2* [ping ไปยังเราเตอร์ Tampa]

14. ทดสอบ ping จากเราเตอร์ 2 (California) ไปยังทุก ๆ ตัว

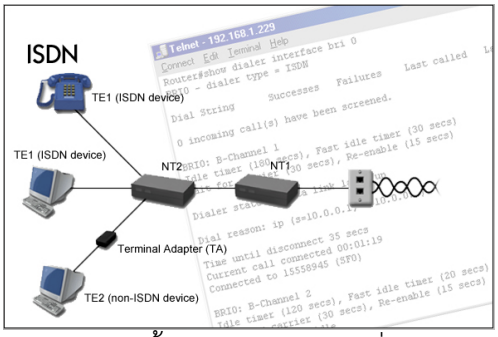
California(config-subif)#*end*

California#*ping 172.16.1.1* [ping ไปยังเราเตอร์ Dallas]

California#*ping 172.16.4.2* [ping ไปยังเราเตอร์ New_York]

California#*ping 172.16.6.2* [ping ไปยังเราเตอร์ Tampa]

การคอนฟิก ISDN (Integrated Services Digital Network)

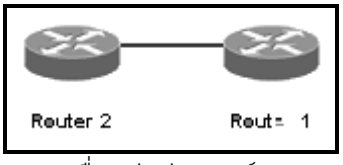


ISDN เป็นบริการที่ทำงานอยู่บนพื้นฐานของโทรศัพท์ที่ใช้งานอยู่แล้ว โดยเป็นการเพิ่มประสิทธิภาพหรือมูลค่าของเครือข่ายสัญญาณให้สูงขึ้น ซึ่งแต่ก่อนโทรศัพท์จะส่งข้อมูลที่เป็นสัญญาณเสียง แต่ ISDN จะเพิ่มข้อมูลภาพ, วิดีโอ, ดิจิตอล เข้าไปด้วยจึงทำให้มีบริการหลากหลายมากขึ้นบนโครงข่ายเดิม (รายละเอียดของ ISDN สามารถหาอ่านได้จากหนังสือเน็ตเวิร์คทั่วไป)

จุดมุ่งหมาย : เรียนรู้วิธีการคอนฟิก ISDN

เครื่องมือที่ใช้ทดลอง : ใช้เราเตอร์ 2 ตัวคือ Router 1 และ Router 2 จากรูปที่ 9.10

การสร้าง Network Map : เราเตอร์ 2 ตัวคือ Router1, Router2



รูปที่ 9.10 การเชื่อมต่อผ่านพอร์ท BRI ของ ISDN

หน้าต่าง Simulator :

1. เชื่อมต่อ Router 1 และ Router 2 ผ่านพอร์ท Bri

   บน Router 1 ให้กำหนดชื่อเป็น Router1 ไอพีเป็น 42.34.10.1 255.255.255.0

   Router>*enable*

   Router#*config treminal*

   Router(config)#*hostname Router1*

   Router1(config)#*interface bri 0* [เข้าสู่โหมดอินเตอร์เฟสของ Bri]

   Router1(config-if)#*ip address 42.34.10.1 255.255.255.0*

   Router1(config-if)#*no shutdown*

   Router1(config-if)#*end*

2. บน Router 2 ให้กำหนดชื่อเป็น Router2 ไอพีแอดเดรสเป็น 42.34.10.121 255.255.255.0

   Router>*enable*

   Router#*config treminal*

   Router(config)#*hostname Router2*

   Router2(config)#*interface bri 0* [เข้าสู่โหมดอินเตอร์เฟสของ Bri]

   Router2(config-if)#*ip address 42.34.10.121 255.255.255.0*

   Router2(config-if)#*no shutdown*

   Router2(config-if)#*exit*

3. กลับไปที่ Router1 เพื่อเริ่มการเซ็ตอัพ ISDN สิ่งแรกที่จะต้องรู้ก่อนคือ จะเลือก switch-type (ชนิดของอินเตอร์เฟสของ ISDN ซึ่งมีอยู่หลายชนิด การใช้งานขึ้นอยู่ว่าจะอยู่โซนไหน เช่น basic-5ess จะใช้แถบประเทศอเมริกา ส่วน ntt จะใช้แถบยุโรป เป็นต้น ส่วนประเทศไทยส่วนมากจะใช้ร่วมกับยุโรปคือ basic-net3 ในที่นี้จะใช้แบบมาตรฐานที่ใช้ได้ทั่วโลกคือ basic-ni) แบบไหน Boson จะมีดีฟอลท์ switch-type เป็น basic-ni

   Router1(config)#*isdn switch-type basic-ni*

4. ขั้นตอนต่อไปเราจะต้องทราบถึงหมายเลข ISDN SPID ซึ่งเป็นหมายเลขสำหรับใช้สร้างการเชื่อมต่อเข้ากับ ISP ซึ่งข้อมูลนี้ทาง ISP จะเป็นผู้กำหนดมาให้ แต่เมื่อต้องการทดสอบการเชื่อมต่อแบบ ISDN จะมีค่าดีฟอลท์ให้ใช้งานได้ซึ่งหมายเลขของ SPID คือ 32177820010100

   Router1(config)#*interface bri 0*

   Router1(config-if)#*isdn spid1 32177820010100* [หมายเลข SPID สำหรับคอนฟิก ISDN]

5. เมื่อถึงขั้นตอนนี้แล้วการเชื่อมต่อในเลเยอร์ที่ 1 ควรจะสมบูรณ์แล้ว โดยสามารถดูได้จากคำสั่ง

   show isdn status

   Router1(config-if)#*end*

   Router1#*show isdn status*

   ถ้าการทำงานถูกต้องในฟิลด์ state จะเท่ากับ MULTIPLE_FRAME_ESTABLISHED ซึ่ง จะเป็นสถานการทำงานที่เลเยอร์ 2

6. ขั้นต่อไปจะต้องมีการเซ็ตเลขหมายที่จะใช้ dial สำหรับสร้างการเชื่อมต่อระดับเลเยอร์ 3 ซึ่งถูกเรียกว่า "dialer string" ให้กำหนดค่าของ dialer string ที่อินเตอร์เฟส Bri 0 ค่าที่ใช้โดยดีฟอลท์คือ 7782001

   Router1#*config terminal*

   Router1(config)#*interface bri 0*

   Router1(config-if)#*dialer string 7782001*

7. เมื่อมีการเชื่อมต่อในระบบ ISDN จะต้องเสียค่าใช้จ่ายทันที ดังนั้นควรเชื่อมต่อเมื่อต้องการใช้งานเท่านั้น สิ่งที่สมควรจะต้องทำคือต้องมีการวางกฎการใช้งานให้เหมาะสมกับงาน เราจะใช้วิธีการที่เรียกว่า Dialer groups และ Dialer lists ซึ่งกล่าวง่าย ๆ ก็คือ แบ่งกลุ่มผู้ใช้งานตามความเหมาะสม (Dialer groups) เป็นกลุ่ม ๆ และแต่ละกลุ่มก็ควรจะมีกฎ (policy) ของแต่ละกลุ่ม (Dialer lists) ใน Dialer lists นั้นจะมีข้อกำหนดว่าจะให้โพรโทคอลใดบ้าง ที่สามารถจะผ่านไปได้ (permit) หรือไม่เช่นนั้นก็ไม่ให้ผ่านเลย (deny)

   Router1(config-if)#*exit*

   Router1(config)#*dialer- list 1 protocol ip permit* [อนุญาตให้โพรโทคอล IP ผ่านได้]

8. จากข้อที่ 7 จะสร้าง dialer-list 1 ไว้ที่โกบอลคอนฟิก แต่ยังไม่มีผลต่ออินเตอร์เฟสของ Bri ที่เชื่อมต่อกับระบบ ISDN ดังนั้นจึงจำเป็นต้องคอนฟิกที่อินเตอร์เฟสที่ Bri อีกครั้ง

   Router1(config)#*interface bri 0*

   Router1(config-if)#*dialer- group 1* [กำหนดให้อินเตอร์เฟส bri 0 เป็นกลุ่มที่ 1 และจะใช้ policy ที่กำหนดไว้ที่ dialer-list 1 ให้มาควบคุมการทำงานของกลุ่มนี้]

   สรุปขั้นตอนการคอนฟิก ISDN อีกครั้ง

   - กำหนดชนิดของ ISDN Switch ที่จะใช้เชื่อมต่อ
   - กำหนดหมายเลข SPID ที่ใช้สื่อสารกันระหว่าง ISDN Switch
   - กำหนดหมายเลขสำหรับใช้เชื่อมต่อกันระหว่างเราเตอร์
   - สร้างกฎการใช้งานที่เรียกว่า Dialer-list โดยอนุญาตให้ใช้ IP ได้

- กำหนด Dialer-list ให้กับอินเตอร์เฟสที่เกี่ยวข้องกับกลุ่มผู้ใช้งาน (Dialer-group)

9. บน Router2 ให้คอนฟิกตามขั้นตอนเหมือนกับ Router1 แต่จะมีส่วนที่แตกต่างกันบ้างเล็กน้อย

   Router2(config)#*isdn switch-type basic-ni*

10. กำหนดหมายเลขของ SPID เป็น 32177820020100 ที่อินเตอร์เฟส Bri 0

    Router2(config)#*interface bri 0*

    Router2(config-if)#*isdn spid1 32177820020100* [หมายเลข SPID สำหรับคอนฟิก ISDN]

11. แสดงสถานะของ ISDN

    Router2(config-if)#*end*

    Router2#*show isdn status*

    ถ้าการทำงานถูกต้องในฟิลด์ state จะเท่ากับ MULTIPLE_FRAME_ESTABLISHED ซึ่ง จะเป็นสถานะการทำงานที่เลเยอร์ 2

12. เซ็ตเลขหมาย dialer string ที่อินเตอร์เฟส Bri 0 ค่าที่ใช้โดยดีฟอลท์คือ 7782002

    Router2#*config terminal*

    Router2(config)#*interface bri 0*

    Router2(config-if)#*dialer string 7782002*

    Router2(config-if)#*exit*

13. **Router2(config)#*dialer- list 1 protocol ip permit*** [อนุญาตให้โพรโทคอล IP ผ่านได้]

14. **Router2(config)#*interface bri 0***

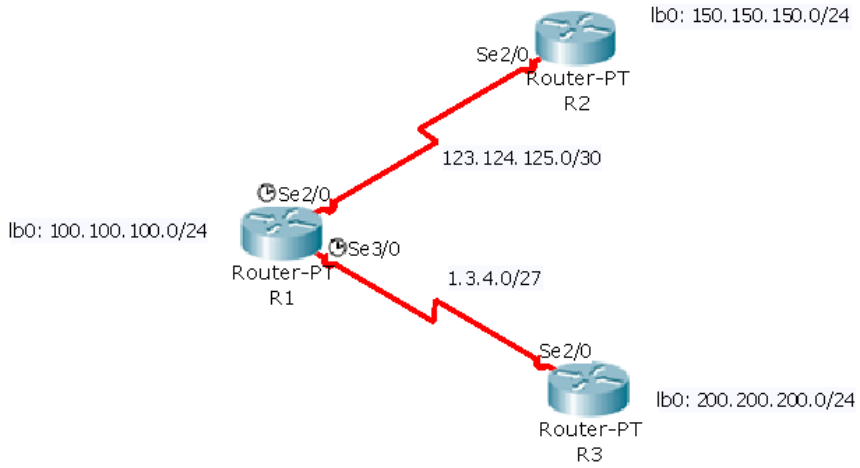
    Router2(config-if)#*dialer- group 1*

15. เมื่อคอนฟิกเราเตอร์ครบทั้ง 2 ตัวแล้ว ให้ทดสอบการทำงานว่าใช้งานได้หรือไม่ โดยใช้คำสั่ง ping ไปยังอินเตอร์เฟสตรงของอีกฝ่าย เช่น สมมุติว่าตอนนี้เราอยู่บนเราเตอร์ 1 มีหมายเลขไอพีแอดเดรสเป็น 24.34.10.1 และเราเตอร์ 2 มีหมายเลขไอพีแอดเดรสคือ 24.34.10.121

    Router1#*ping 42.34.10.121*

แบบฝึกหัดท้ายบท

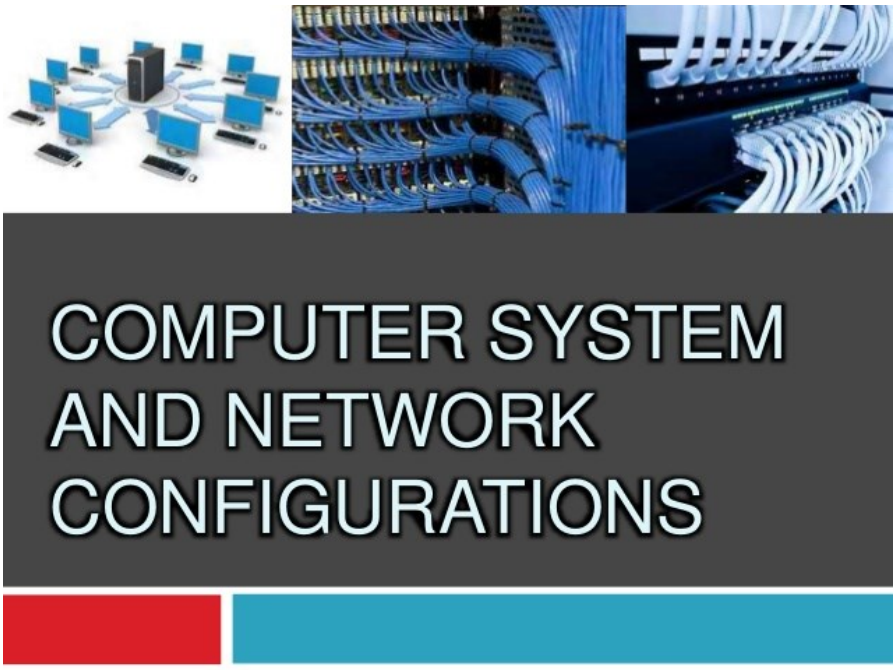
1. จากรูปที่ 1 จงออกแบบและคอนฟิกระบบเครือข่าย WAN แบบ PPP ที่มีการ authentication แบบ CHAP ตาม รูปด้านล่าง ให้สามารถทำงานได้



รูปที่ 1

2. จงออกแบบและคอนฟิก Frame Relay แบบ Multipoint (เราเตอร์อย่างน้อย 3 ตัว)

ภาคที่สอง

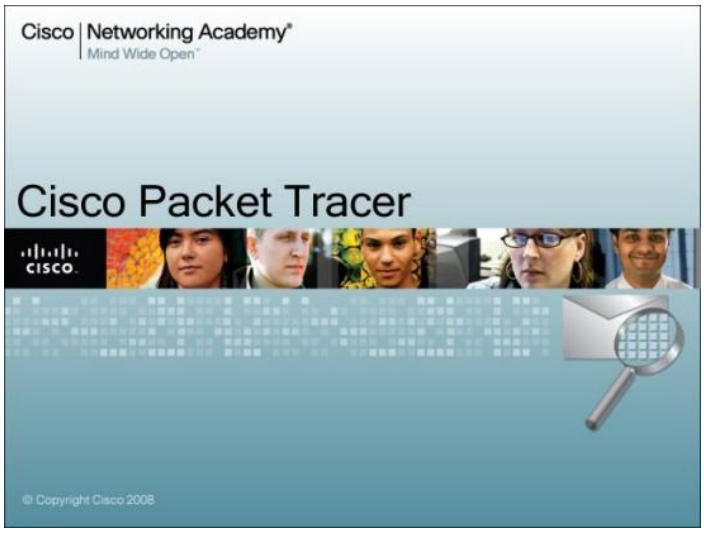


การออกแบบ ติดตั้ง และการวิเคราะห์เครือข่ายด้วย

โปรแกรมจำลองเครือข่าย

บทที่ 10

โปรแกรมจำลองเครือข่าย (Network Simulation)



แนวคิด

ในบทนี้จะมาเรียนรู้คุณสมบัติ และความสามารถ ของโปรแกรมจำลองเครือข่ายชื่อว่า Packet Tracer ของบริษัท Cisco เพื่อใช้ในการออกแบบและวิเคราะห์ระบบเครือข่ายได้อย่างมีประสิทธิภาพ

วัตถุประสงค์

1. เพื่อให้ทราบถึงความสามารถ และการใช้งานของโปรแกรมจำลองเครือข่าย
2. เพื่อใช้สำหรับช่วยวิเคราะห์การทำงานของระบบเครือข่าย

ปฏิบัติการดูแลเครือข่ายโดยใช้ซอฟต์แวร์ อุปกรณ์ทางพาณิชย์

การจำลองการทำงานของระบบเครือข่าย (Network Simulation) และโปรแกรมจำลองเครือข่าย (Network Simulator) ทำหน้าที่จำลองการทำงานของอุปกรณ์เครือข่าย (Physical Device) เช่น เครื่องคอมพิวเตอร์ เครื่องเซิร์ฟเวอร์ เราเตอร์ สวิตช์ สายนำสัญญาณ เป็นต้น และทำหน้าที่จำลองการทำงานของโพรโทคอลที่ใช้สื่อสารบนระบบเครือข่าย (Protocol) เช่น TCP/IP, UDP, RIP, OSPF, BGP, DHCP, DNS, HTTP เป็นต้น เป็นเทคโนโลยีใหม่ที่ทำงานในรูปแบบเสมือนจริง หรือ Virtual Packet Technology เพื่อช่วยในการออกแบบ วิเคราะห์ ติดตั้งระบบเครือข่ายเหมือนสถานการณ์จริง ศึกษาพฤติกรรมการทำงานของระบบเครือข่าย ศึกษาการทำงานของโพรโทคอล วางแผนระบบเครือข่าย ปรับปรุงระบบเครือข่ายที่มีอยู่แล้วในองค์กร ลดระยะเวลาการเรียนรู้สร้างผู้ดูแลระบบเครือข่ายให้เกิดเชี่ยวชาญได้อย่างรวดเร็ว ลดต้นทุน ประหยัดเวลา ลดความเสี่ยงทดสอบการทำงานก่อนติดตั้งอุปกรณ์จริง ค้นหาข้อผิดพลาดที่เกิดขึ้นในระบบเครือข่าย ช่วยในการวางแผนจัดซื้อ ประเมินราคาเบื้องต้น วางแผนในการเปลี่ยนแปลงเทคโนโลยี และเพื่อศึกษาทำการวิจัยในระดับสูง เป็นต้น ถึงแม้ว่าโปรแกรมจำลองเครือข่ายจะมีคุณสมบัติที่เด่นมากหลายประการ แต่ก็ยังมีข้อเสีย อยู่หลายประการเช่นกัน คือ การจำลองไม่สามารถทดแทนการทำงานของอุปกรณ์จริงได้ 100 เปอร์เซ็น ความสามารถของโปรแกรมจำลองขึ้นอยู่กับเจ้าของซอฟต์แวร์ว่าต้องการใส่คุณสมบัติอะไรเข้าไปให้ผู้ใช้ใช้งานได้บ้าง หรือคำสั่งในการทำงานไม่ครบ ดังนั้นโปรแกรมจำลองส่วนใหญ่จะมีประสิทธิภาพและคุณสมบัติน้อยกว่าอุปกรณ์จริงเสมอ เว้นแต่ มีโปรแกรมจำลองบางประเภทที่ใช้ในทางวิจัย เช่น NS-2 ที่เน้นให้ผู้วิจัยสามารถสร้างโพรโทคอลขึ้นมาใหม่ได้ ข้อด้อยอีกประการหนึ่งคือโปรแกรมจำลองส่วนใหญ่ทำงานอยู่ภายในระบบแบบปิด (Closed System) คือไม่สามารถทำการส่งข้อมูลไปยังโปรแกรมจำลองตัวอื่นๆ ที่อยู่ต่างเครื่องกันได้ แต่ในปัจจุบันมีโปรแกรมจำลองหลายตัวได้พัฒนาให้มีความสามารถดังกล่าวแล้ว เช่น Packet Tracer เวอร์ชัน 5 ขึ้นไป

โปรแกรมจำลองเครือข่าย ที่นิยมใช้งานในปัจจุบันมีอยู่หลายยี่ห้อ โดยแต่ละยี่ห้อก็มีข้อเด่นข้อด้อยที่แตกต่างกันไป สำหรับในภาคที่ 2 นี้เลือกใช้ Packet Tracer เวอร์ชัน 5.3 ซึ่งทางบริษัทซิสโก้ (Cisco) เป็นผู้สร้างขึ้น เนื่องจากเหตุผลหลายประการในการเลือกใช้ เช่น โปรแกรม Packet Tracer มีอินเทอร์เฟสที่ง่ายต่อการใช้งาน โดยมีลักษณะการใช้งานแบบกราฟฟิก โปรแกรมมีอุปกรณ์ให้เลือกค่อนข้างมาก โดยมีตั้งแต่ การ์ดอินเตอร์เฟส เครื่องคอมพิวเตอร์ สายนำสัญญาณ สวิตช์ เราเตอร์ DSL อุปกรณ์ประมวลผลแบบกลุ่มเมฆ ไวเลส อื่นๆ อีกมาก โปรแกรมสามารถแสดงข้อมูลที่วิ่งบนเครือข่ายได้อย่างละเอียด ทำให้ผู้ใช้งานเห็นภาพการทำงานของเครือข่ายได้เป็นอย่างดี โปรแกรมถูกพัฒนาอย่างต่อเนื่องและในอนาคตอาจจะครอบคลุมเนื้อหาเกี่ยวกับระบบเครือข่ายได้ทั้งหมด ปัจจุบัน สามารถครอบคลุมเนื้อหาของ CCNA เกือบทั้งหมด และ CCNP บางส่วน โปรแกรมสามารถติดตั้งและใช้งานได้ทั้งระบบปฏิบัติการลินุกซ์และวินโดวส์ มีผู้ใช้งานเป็นจำนวนมาก ซึ่งในมหาวิทยาลัยส่วนใหญ่ๆ ทั่วโลกจะใช้สำหรับสอนเสริมในวิชาคอมพิวเตอร์เครือข่าย เครือข่ายชั้นสูง เป็นต้น ซึ่งจะกล่าวอย่างละเอียดสำหรับคุณสมบัติของ Packet Tracer ในหัวข้อถัดไป

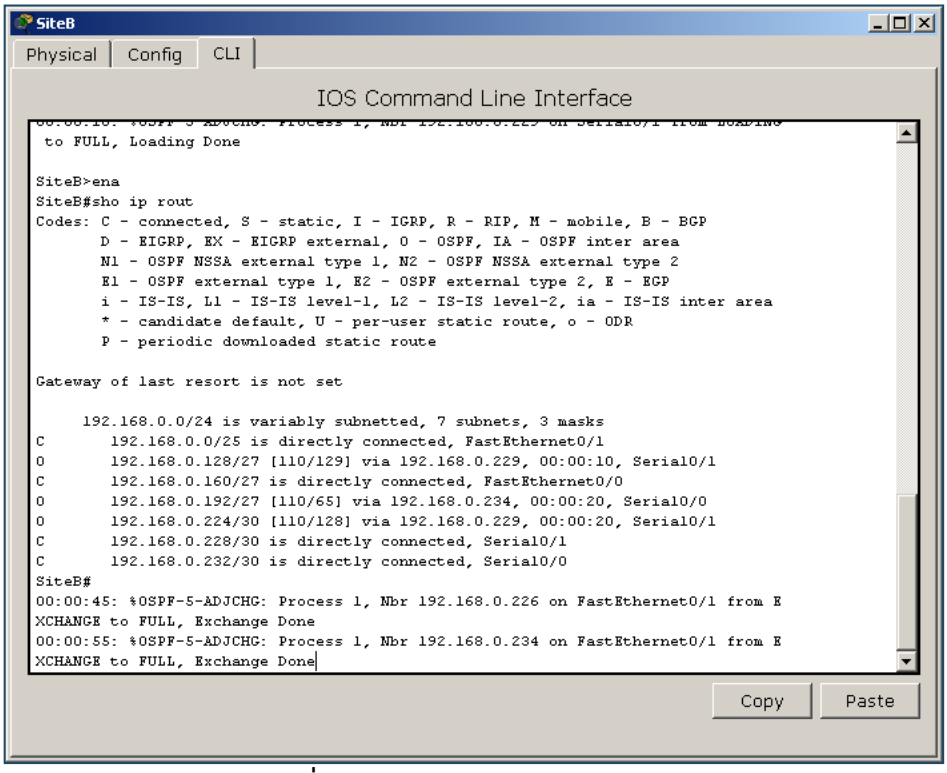
Simulation-Based Learning

โปรแกรม Packet tracer เป็นโปรแกรมประเภท Simulation-Based Learning คือ เป็นโปรแกรมที่ทำการสร้างสถานการณ์จำลอง เพื่อทำให้ผู้เรียนเห็นภาพได้ชัดเจนขึ้น ทำให้ผู้เรียนและผู้สอนสามารถเรียนรู้ในการกระบวนการทำงานของเครือข่ายได้เป็นอย่างดี โปรแกรมดังกล่าวสนับสนุนให้ผู้เรียนผู้สอนสามารถทำงานร่วมกันเป็นทีม สร้างองค์ความรู้ใหม่ แก้ปัญหาที่มีความซับซ้อน ออกแบบระบบเครือข่าย กระตุ้นการเรียนรู้ เน้นให้สามารถทำงานได้จริง

คุณสมบัติและความสามารถของ Packet Tracer (version 5.3 ขึ้นไป)

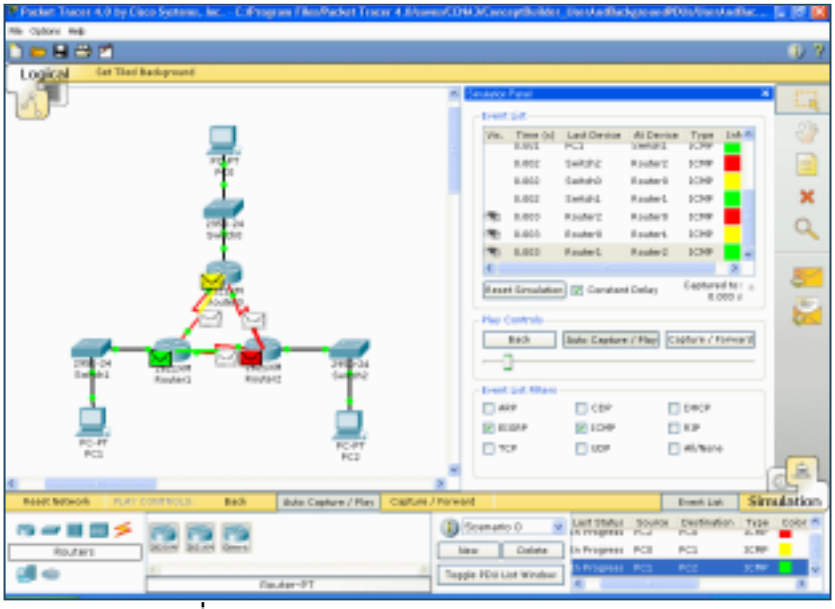
Packet tracer มีความสามารถดังต่อไปนี้คือ

1. Simulation โปรแกรมมีความสามารถจำลองการทำงานของระบบปฏิบัติการ (IOS) และคำสั่งต่างๆ ที่ทำงานอยู่บนอุปกรณ์ของซิสโก้ได้เกือบสมบูรณ์แบบ สามารถจำลองการทำงานของโพรโทคอลที่ทำหน้าที่เราท์โพรโทคอล (Routing Protocol) อื่นๆ ให้ทำงานได้ เช่น RIP, OSPF, BGP เป็นต้น รวมถึงโพรโทคอลที่ถูกเราท์ด้วย เช่น FTP, HTTP, DNS, SMTP เป็นต้น ดังรูปที่ 10.1



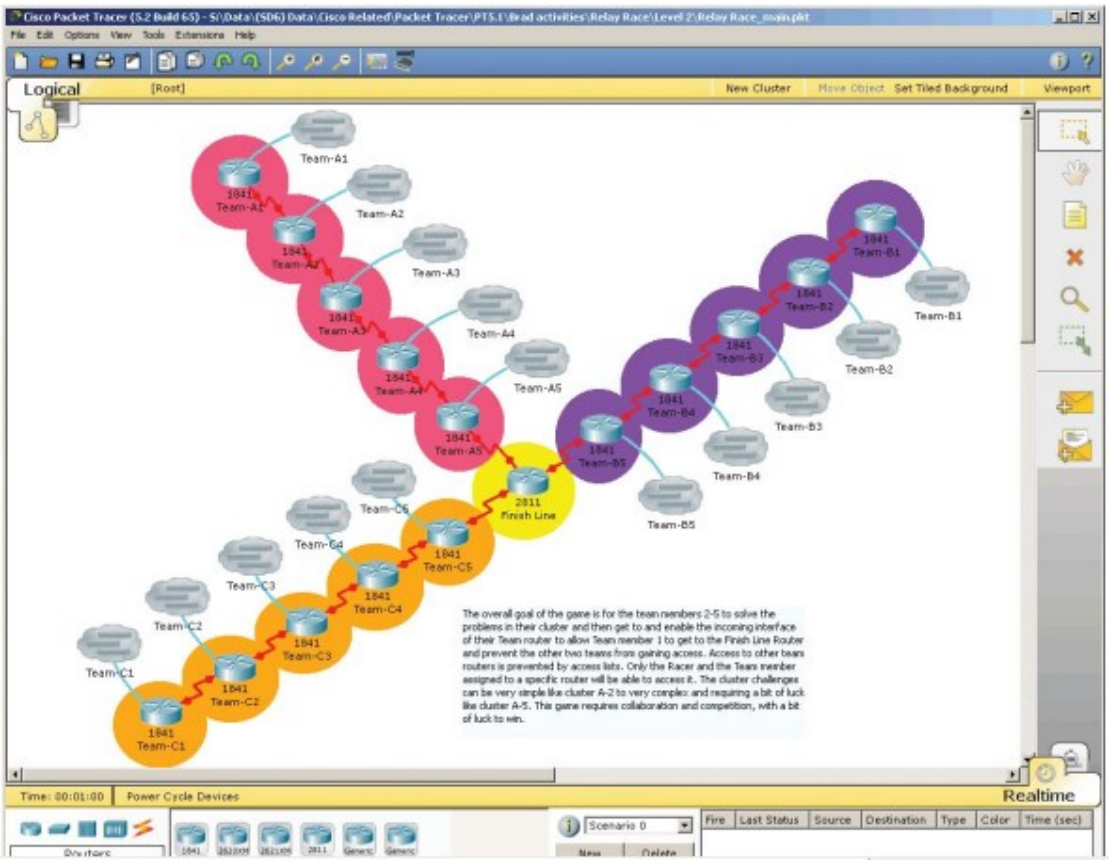
รูปที่ 10.1 จำลองการทำงานของ IOS

2. Visualization โปรแกรมมีความสามารถแสดงกระบวนการทำงานของเครือข่าย ในรูปแบบที่ง่ายต่อการทำความเข้าใจ เช่น แสดงเป็นรูปภาพ สี เสียง และมัลติมีเดีย ดังรูปที่ 10.2



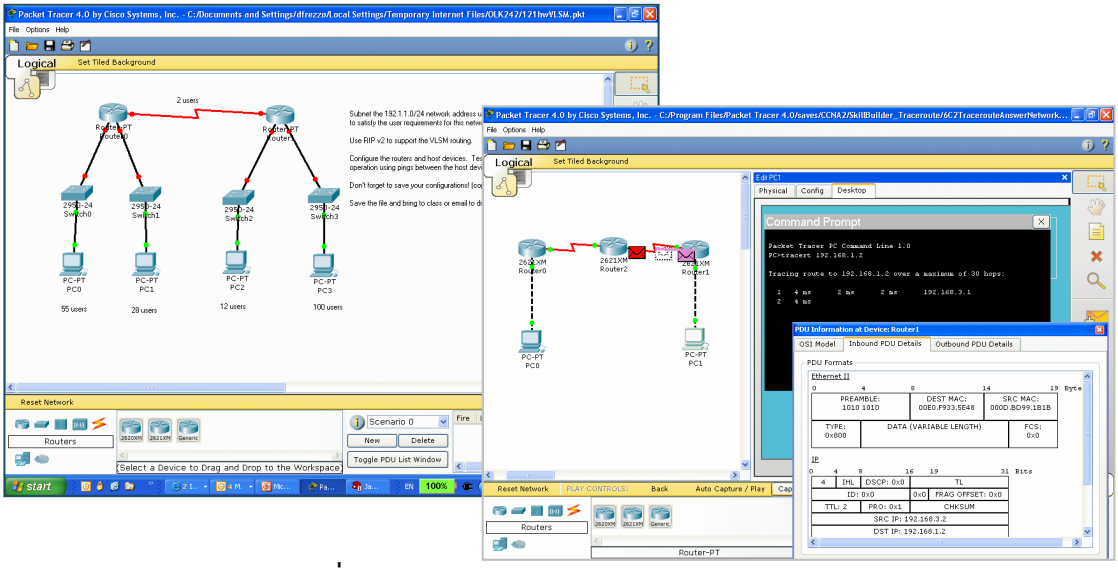
รูปที่ 10.2 แสดงการทำงานแบบ visualization

3. Collaboration on Multiuser Activities โปรแกรมมีความสามารถเชื่อมโยงเครือข่ายที่อยู่ต่างสถานที่กันให้สามารถเชื่อมต่อกันได้ ดังรูปที่ 10.3



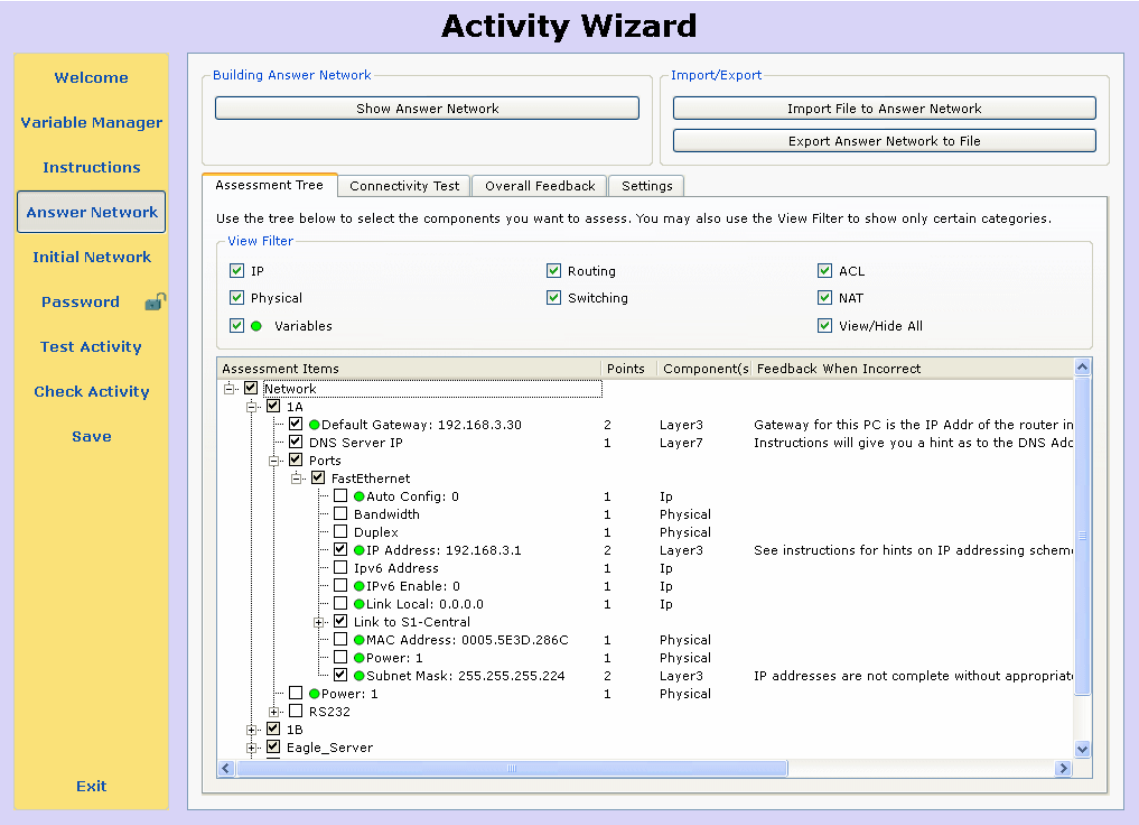
รูปที่ 10.3 แสดงการเชื่อมโยงเครือข่ายที่อยู่ต่างไซต์ (Collaboration)

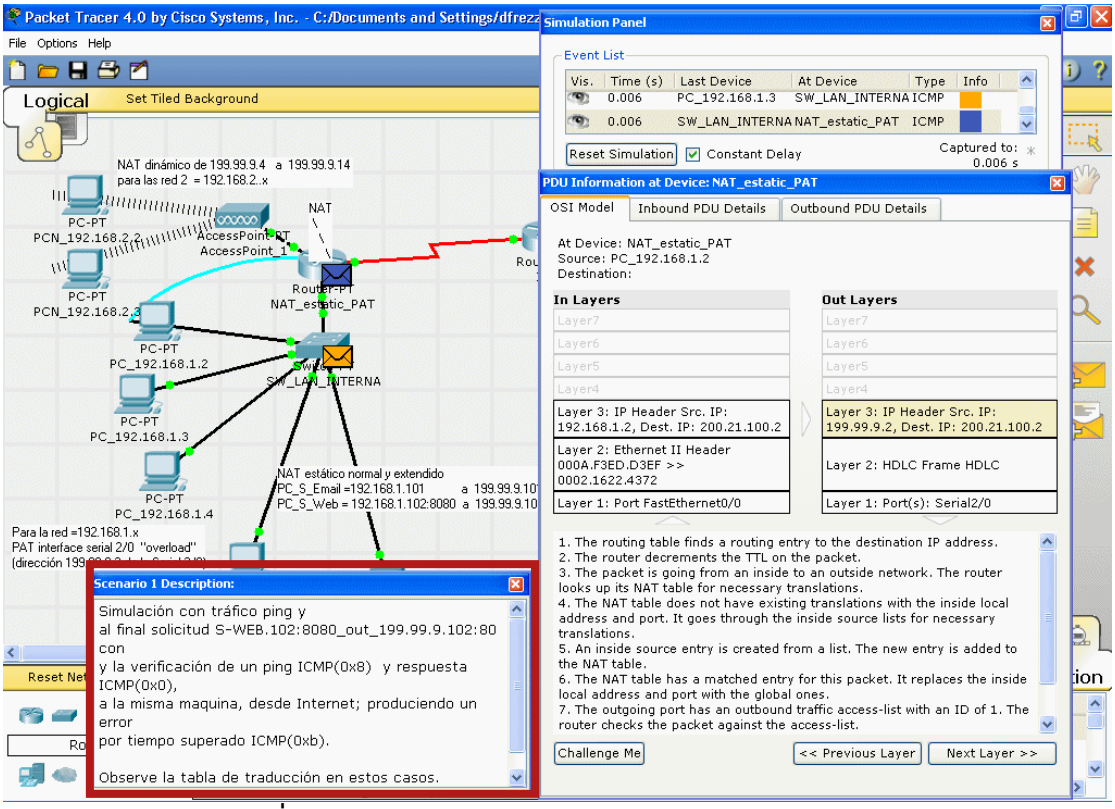
4. Homework and Pre-Lab รองรับและสนับสนุนให้ผู้เรียนสามารถทดสอบทำ LAB เกี่ยวกับเครือข่ายได้โดยสมบูรณ์ ดังรูปที่ 10.4



รูปที่ 10.4 แสดงการทดสอบ LAB เครือข่าย

5. Activity Wizard คือ ผู้สอนหรือผู้ที่มีหน้าที่ในการสอนเกี่ยวกับระบบเครือข่ายสามารถสร้างสถานการณ์ในลักษณะเป็นขั้นๆ โดยผู้เรียนต้องทดสอบและแก้ปัญหาไปทีละขั้นๆ จนกว่าจะแก้ปัญหาได้ทั้งหมด ซึ่งโปรแกรมสามารถแสดงคะแนนที่ผู้เรียนแก้ปัญหาในแต่ละขั้นตอนหรือทั้งหมดได้ โดยผู้สอนจะกำหนดคำตอบไว้ก่อนร่วงหน้า ดังรูปที่ 10.5



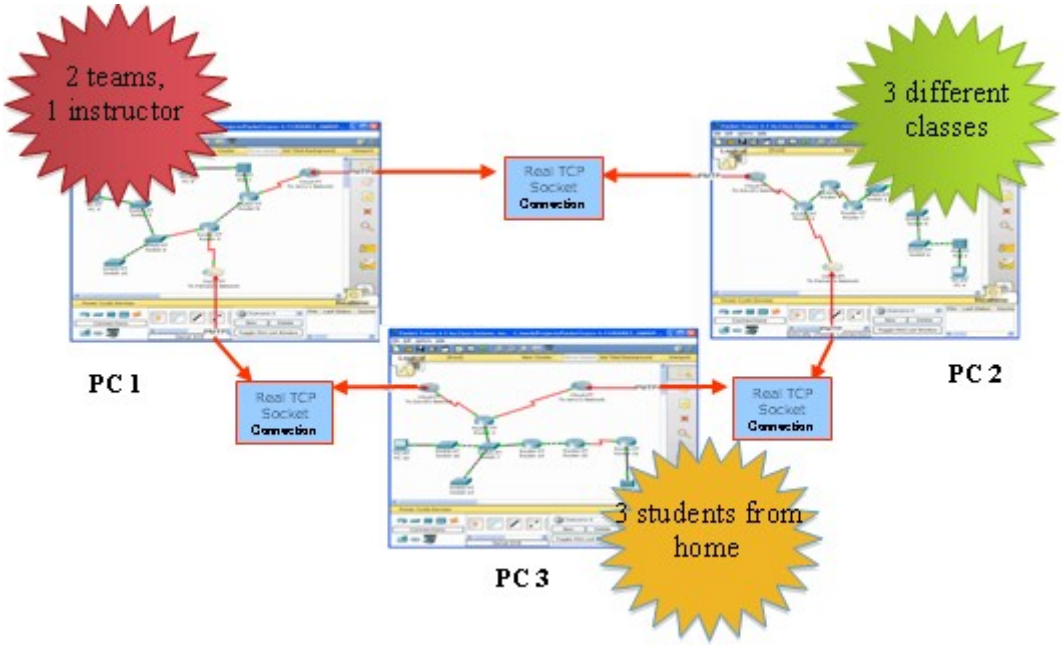


รูปที่ 10.5 แสดงการสร้าง LAB ด้วย Activity Wizard

6. Multiuser Functionality โปรแกรมมีความสามารถในการเชื่อมต่อกับผู้เรียนรายอื่นๆ ซึ่งอยู่ที่ใดๆ ก็ได้ โดยผ่านโพรโทคอล TCP/IP ผ่านเครือข่ายอินเทอร์เน็ตได้เป็นอย่างดี ทำให้ผู้เรียนสามารถสร้างกลุ่มเครือข่าย ทดสอบเครือข่ายขนาดใหญ่ ทดสอบการเชื่อมต่อที่ซับซ้อน สามารถแข่งขันกันระหว่างภายในกลุ่มเครือข่ายได้ ดังรูปที่ 10.6

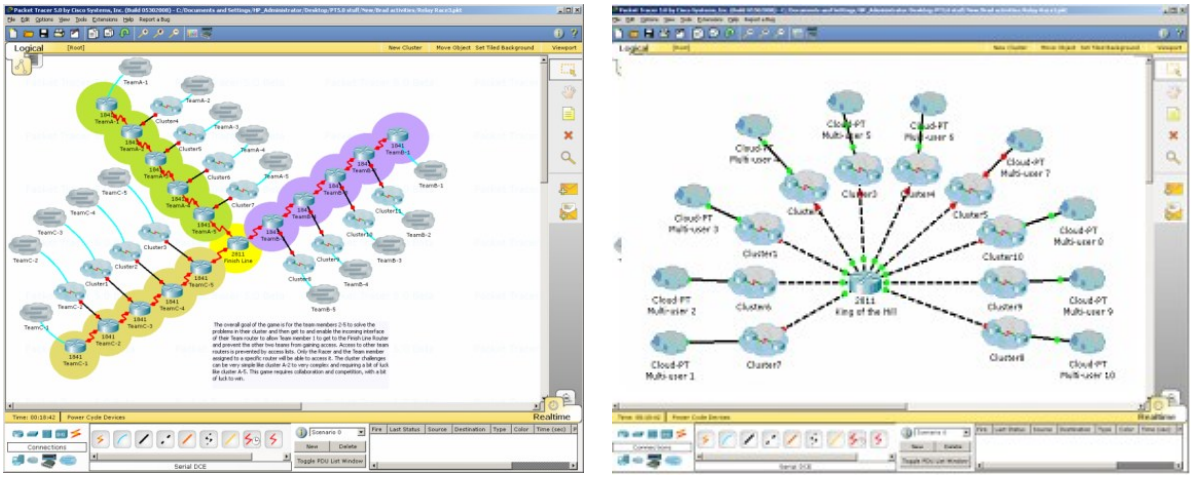
รูปที่ 10.6 แสดงการเชื่อมโยงเครือข่ายด้วย packet tracer Multiuser Functionality

จากตัวอย่างรูปที่ 10.7 สมมุติว่า มีการแข่งขันการออกแบบและเชื่อมต่อเครือข่ายระหว่าง 3 ทีม โดยแต่ละทีมอยู่ต่างสถานที่กัน เช่น ทีมที่หนึ่ง อยู่ที่มหาวิทยาลัย ทีมที่สองอยู่ที่ทำงาน และทีมที่สามอาจจะอยู่ที่บ้าน ผ่านโพรโทคอล Packet Tracer Messaging Protocol (PTMP) ใน packet tracer



รูปที่ 10.7 แสดงการเชื่อมโยงเครือข่ายระหว่าง 3 ทีมด้วยโพรโทคอล PTMP

7. Multiuser Games for Social Learning โปรแกรมมีคุณสมบัติในการเชื่อมโยงผู้ใช้ต่างๆ เข้าด้วยกัน จึงส่งผลให้ผู้เรียนและผู้สอนสามารถสร้างสรรค์เกมส์ ที่เกี่ยวข้องกับระบบเครือข่ายได้ ทำให้ผู้เรียนผ่อนคลายได้ ดังรูปที่ 10.8

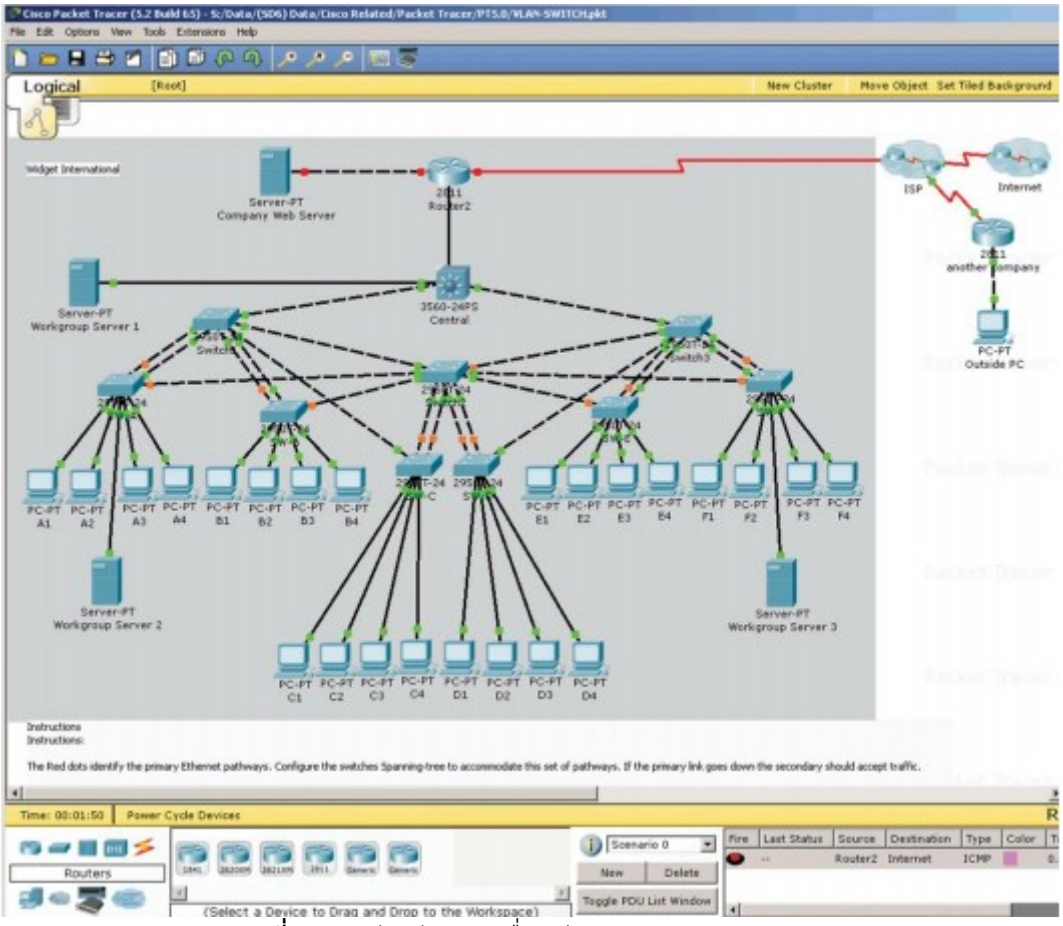


เกมส์ Relay Race เกมส์ King of the Hill

รูปที่ 10.8 ตัวอย่างการสร้างสรรค์เกมส์ด้วย Packet Tracer

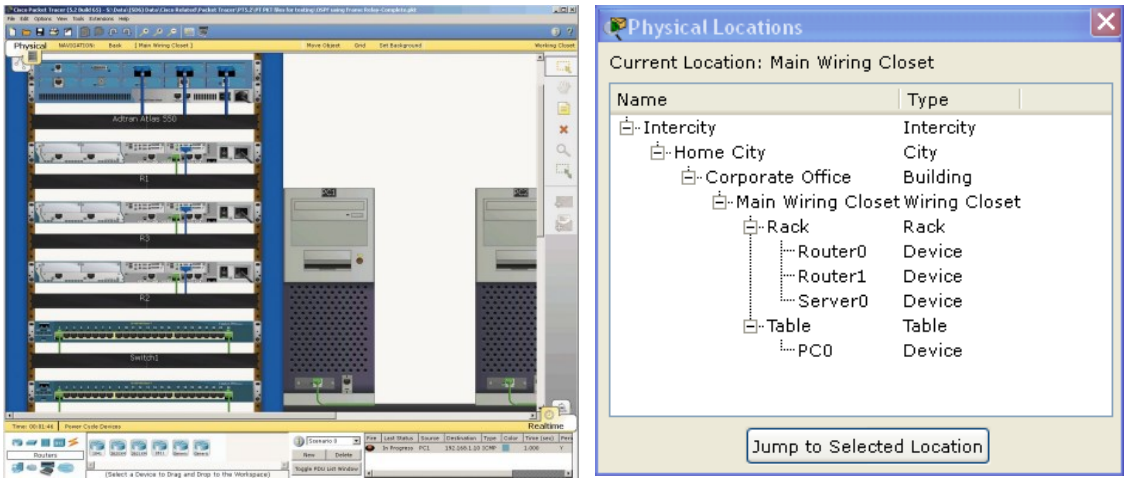
8. Logical and Physical Workspaces โปรแกรมออกแบบให้ผู้ใช้สามารถทำงานได้ 2 แบบ คือ

- Logical Workspaces แสดงรูปการเชื่อมต่อเครือข่ายทางลอจิคอล แบบนี้ผู้ใช้สามารถสร้างรูปแบบการเชื่อมต่ออุปกรณ์ต่างๆ เช่น เครื่องคอมพิวเตอร์ เซิร์ฟเวอร์ สวิตช์ เราเตอร์ สายนำสัญญาณ และอื่นๆ ในลักษณะที่เป็นรูปภาพสัญลักษณ์แทนการเชื่อมต่อจริง ดังรูปที่ 10.9



รูปที่ 10.9 ตัวอย่างการเชื่อมต่อแบบ logical

- Physical Workspaces แสดงรูปแบบการเชื่อมต่อทางกายภาพ โดยอ้างอิงกับตำแหน่งที่ตั้งของสถานที่ติดตั้งระบบเครือข่ายจริง เช่น อำเภอที่ติดตั้ง ตำบล อาคาร ห้อง และ ตู้ Rack เป็นต้น ดังรูปที่ ดังรูปที่ 10.10



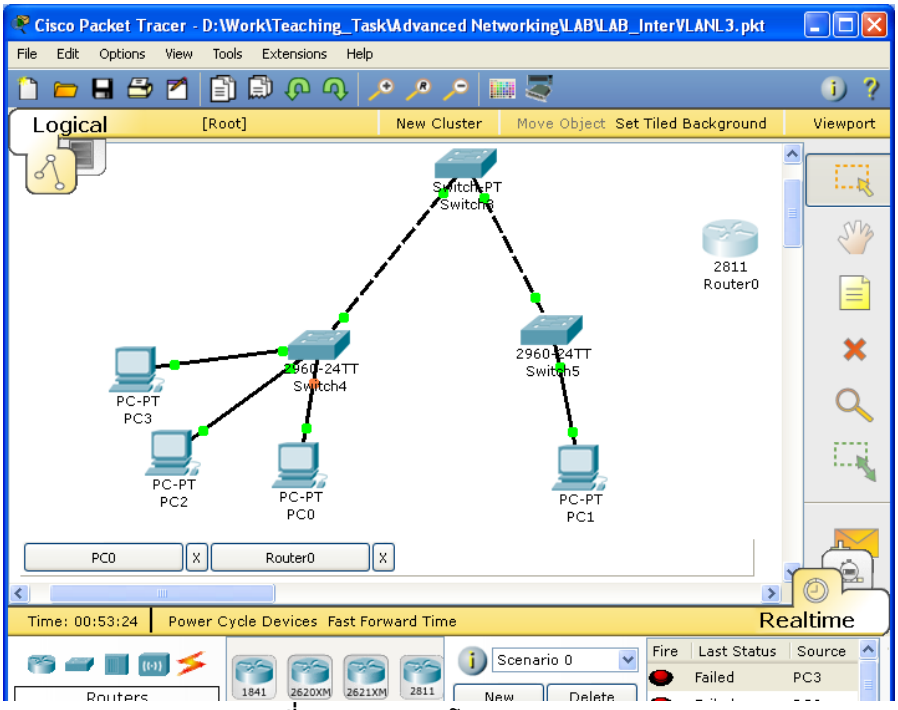
ก. แสดงการเชื่อมต่ออุปกรณ์ภายในตู้ Rack ข. แสดงการลำดับชั้นการเชื่อมต่อ

รูปที่ 10.10 แสดงการเชื่อมต่อแบบ physical

จากรูปที่ 10.10 ก แสดงให้เห็นถึงรูปแบบการจัดวางอุปกรณ์เครือข่ายต่างๆ เช่น เซิร์ฟเวอร์ เราเตอร์ บน Rack cabinet และรูปที่ 10.10 ข แสดงลำดับที่ตั้งของอุปกรณ์ โดยเริ่มตั้งแต่เมือง หรือจังหวัด ต่อจากนั้นก็ค่อยๆ ขยับพื้นที่ให้แคบลงมาเรื่อยๆ เป็น อาคาร, ห้องเก็บอุปกรณ์, Rack, โต้, เราเตอร์ และอุปกรณ์เครือข่ายอื่นๆ ตามลำดับ

9. Real-Time and Simulation Modes โปรแกรมมีความสามารถแสดงผลการทำงานใน 2 รูปแบบคือ

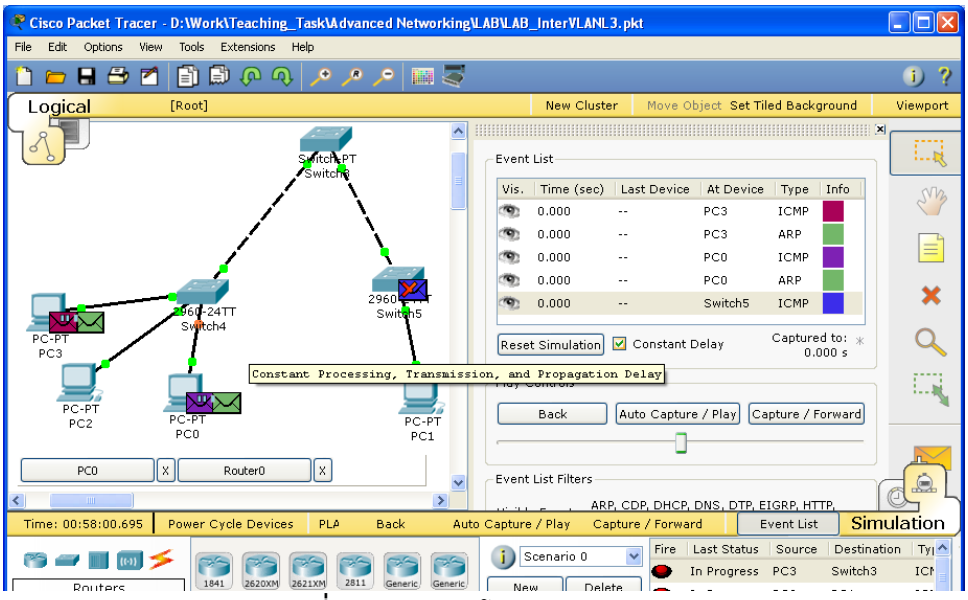
- โหมด Real-Time ในโหมดนี้ผู้ใช้สามารถเลือกอุปกรณ์ต่างๆ ที่ต้องการ มาเชื่อมต่อกันในลักษณะรูปสัญลักษณ์ เหมือนการวาดแผนผังเครือข่ายบนกระดาษ หรือ การใช้โปรแกรมประเภทเขียนผังเครือข่าย เช่น Visio, Smart draw เป็นต้น ซึ่งผู้ใช้จะต้องเข้าใจเรื่องของสัญลักษณ์ของอุปกรณ์ต่างๆ ที่นำมาเชื่อมต่อกัน เช่น เราเตอร์จะใช้สัญลักษณ์ทรงกลมแบนโดยมีลูกศรหัวเข้า 2 ทิศทาง และหัวออก 2 ทิศทาง เป็นต้น ระหว่างการเชื่อมต่ออุปกรณ์ต่างๆ ลงบนโหมดนี้ อุปกรณ์จะแสดงสถานการทำงานให้ผู้ใช้เห็นด้วย เช่น ไฟกระพริบสีเขียวแสดงการเชื่อมต่อสมบูรณ์ สีส้มแสดงกำลังเริ่มกระบวนการเชื่อมต่อ โปรแกรมจะแสดงชื่ออุปกรณ์ พอร์ตที่ทำการเชื่อมต่อ รวมถึงเสียงที่เกิดจากการเชื่อมต่อด้วย โหมด Real-Time นี้จะเป็นโหมดที่ผู้ใช้จะต้องเริ่มต้นวางผังเครือข่ายก่อนเสมอ และใช้งานบ่อยที่สุดด้วย ดังรูปที่ 10.11



รูปที่ 10.11 แสดงโหมด Real-Time

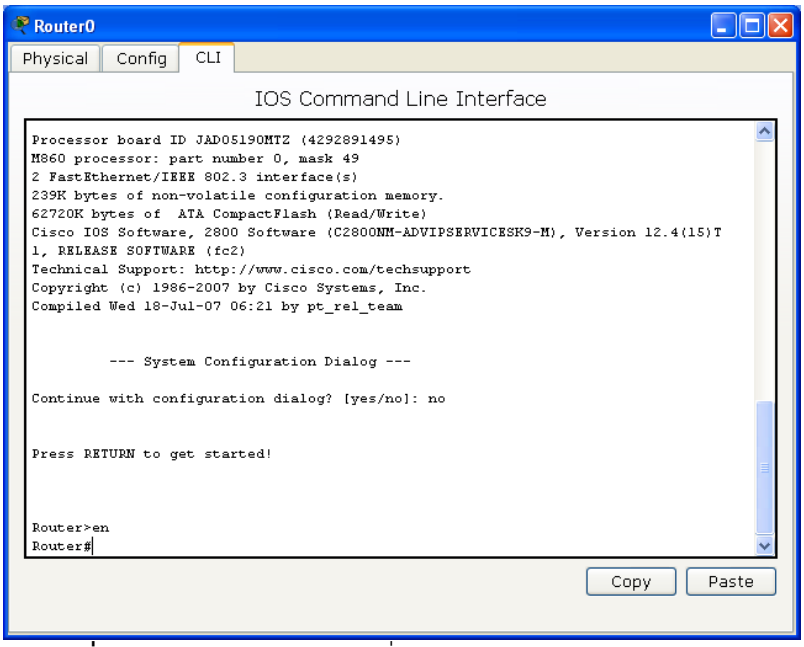
- โหมด Simulation ในโหมดนี้ผู้ใช้สามารถสร้างข้อมูล (packet) เข้าไปยังระบบเครือข่ายที่สร้างขึ้นแล้ว เพื่อเฝ้าดูและสังเกตพฤติกรรมการทำงานของเครือข่ายที่ได้

ออกแบบไว้ ซึ่งโปรแกรมจะแสดงผลการทำงานเป็นแบบอนิเมชัน สี เสียง ทำให้ผู้ใช้รู้สึกตื่นตัว ส่งผลให้เห็นภาพทิศทางการไหลของข้อมูลทั้งระบบ ซึ่งช่วยให้ผู้ออกแบบสามารถวิเคราะห์และแก้ปัญหาที่เกิดขึ้นได้อย่างรวดเร็ว ดังรูปที่ 10.12



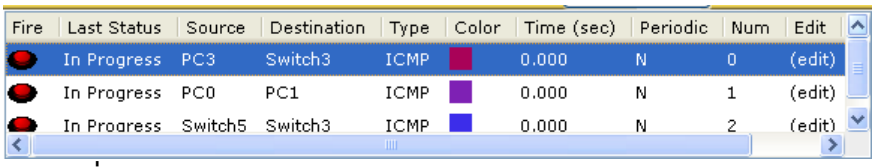
รูปที่ 10.12 แสดงโหมด Simulation

10. User friendly Command Line Interface (CLI) โปรแกรมจัดเตรียมส่วนติดต่อกับผู้ใช้งาน ผ่านทาง command line (คือการคีย์คำสั่งทีละคำสั่งผ่านทาง console ในรูปแบบ text) สำหรับผู้เรียนที่ต้องการคีย์คำสั่งควบคุมเราเตอร์เหมือนกับเราเตอร์ของจริง และสำหรับผู้ใช้ที่ไม่คล่องในการคีย์คำสั่งแบบใช้ command line ก็สามารถคอนฟิกอุปกรณ์ต่างๆ ได้เหมือนกันโดยผ่านทางกราฟฟิกแทน แต่ก็จะมีข้อจำกัดและไม่คล่องตัวเหมือนการใช้ CLI ดังรูปที่ 10.13



รูปที่ 10.13 แสดงรูปแบบการสั่งงานด้วย command line (CLI)

11. Global event list (packet sniffer) โปรแกรมสามารถรายงาน สถานการเชื่อมต่อ ทิศทางการไหลของข้อมูล ชนิด เวลา จำนวน ของแพ็คเก็ตได้อย่างละเอียดผ่านทาง event list ทำให้ผู้เรียนเข้าใจการพฤติกรรมการทำงานของแพ็คเก็ตได้เป็นอย่างดี ดังรูปที่ 10.14

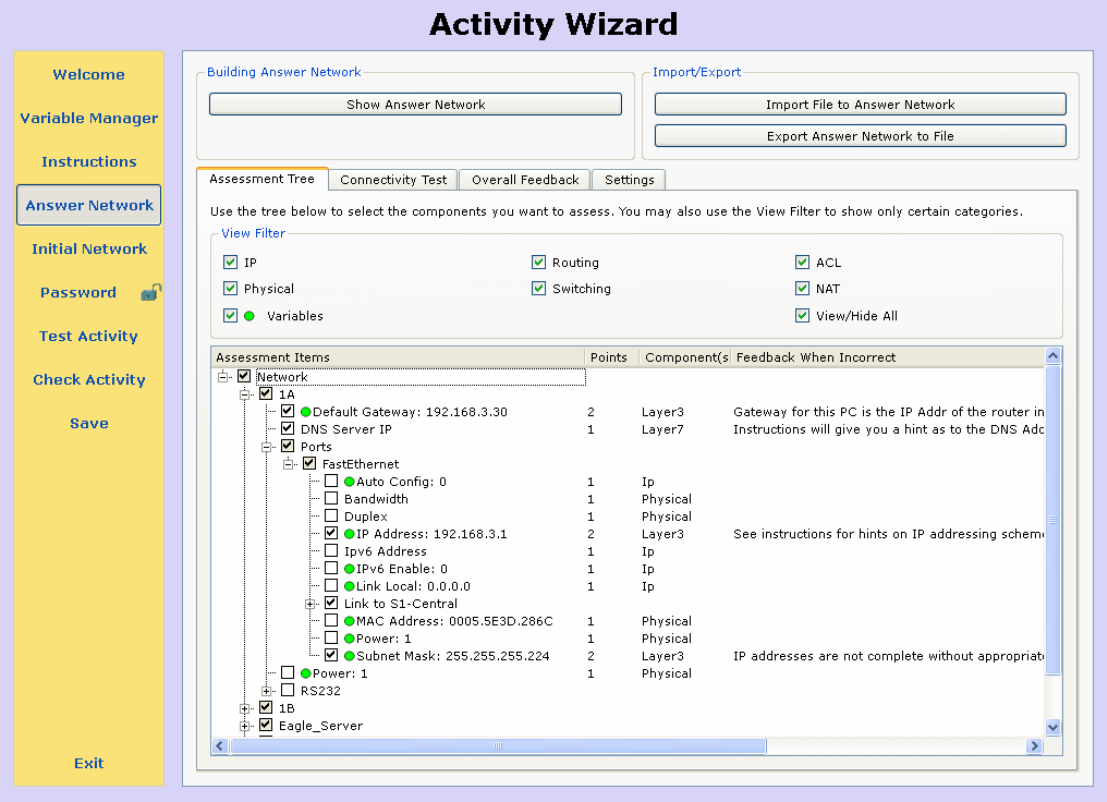


Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit
	In Progress	PC3	Switch3	ICMP		0.000	N	0	(edit)
	In Progress	PC0	PC1	ICMP		0.000	N	1	(edit)
	In Progress	Switch5	Switch3	ICMP		0.000	N	2	(edit)

รูปที่ 10.14 แสดงการทำงานของฟังก์ชัน event list (packet sniffer)

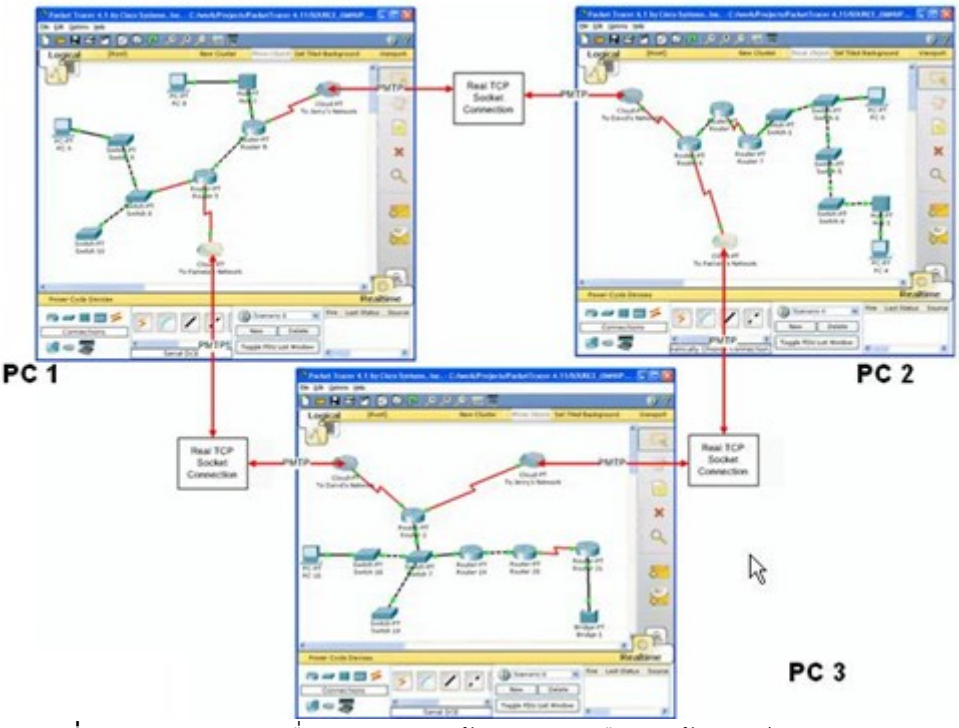
12. LAN, switching, TCP/IP, routing and WAN protocols โปรแกรมรองรับการทำงานแบบเครือข่าย LAN, switching, โพรโทคอลทีซีพี-ไอพี, โพรโทคอลที่ทำหน้าที่ผลักดันให้โพรโทคอลอื่นๆ เดินทางไปบนเครือข่าย (routing protocol), และรองรับโพรโทคอลบางส่วนของการเชื่อมต่อระดับ WAN ด้วย ซึ่งรายละเอียดจะแสดงในหัวข้อ Packet Tracer ทำอะไรได้บ้าง

13. Activity Wizard, Lab grading โปรแกรมถูกออกแบบขึ้นมาเพื่อใช้สำหรับการเรียนการสอนด้านระบบเครือข่ายโดยเฉพาะ ดังนั้นจึงมีคุณสมบัติให้ผู้สอนสามารถสร้าง LAB ขึ้นมาในลักษณะ activity wizard คือ ผู้สอนจะเตรียม LAB ที่ถูกออกแบบไว้อย่างมีขั้นตอน (โดยการกำหนดคำตอบที่ถูกต้องไว้ล่างหน้า) ผู้เรียนจะต้องปฏิบัติหรือแก้ปัญหาโจทย์ไปทีละขั้นๆ โดยไม่สามารถข้ามขั้นตอนได้ ทำให้ผู้สอนสามารถควบคุมแผนการสอน หรือสร้างบทเรียนที่สอดคล้องให้กับผู้เรียนได้เป็นอย่างดี ดังรูปที่ 10.15



รูปที่ 10.15 แสดงการสร้าง LAB ด้วย activity wizard

14. Multiuser functionality โปรแกรมจำลองเครือข่ายในสมัยก่อนมีข้อจำกัดประการที่สำคัญ คือ จะจำลองเครือข่ายแบบ stand-alone ทำให้ผู้เรียนเห็นภาพการเชื่อมต่อเฉพาะที่ โดยมีโปรแกรมจำลองเครือข่ายเป็นตัวสร้างให้เท่านั้น ไม่สามารถทดสอบได้ว่า เครือข่ายที่ได้สร้างขึ้นจะสามารถทำงาน ในสถานการณ์จริงได้หรือไม่ แต่สำหรับ packet tracer ตั้งแต่เวอร์ชันที่ 5 เป็นต้นไป มีคุณสมบัติรองรับการเชื่อมต่อกันระหว่างผู้เรียนแต่ละรายที่อยู่ต่างสถานที่กันได้ โดยผ่านโพรโทคอล PTMP ซึ่งส่งผลให้ผู้เรียนแต่ละรายสามารถสร้างเครือข่ายเสมือนจริงซ้อนบนเครือข่ายอินเทอร์เน็ตอีกชั้นหนึ่ง ผู้เรียนแต่ละรายสามารถสร้างเครือข่ายภายในหรือ local area network ของตนเองที่แตกต่างกันได้ และสามารถทดสอบเชื่อมต่อเครือข่ายของตนเองที่สร้างขึ้นผ่านทางเครือข่าย WAN เข้าด้วยกัน ดังรูปที่ 10.16



รูปที่ 10.16 แสดงการเชื่อมต่อระหว่างผู้ออกแบบเครือข่ายด้วยฟังก์ชัน multiuser

15. Multiple platform support โปรแกรม packet tracer มีความสามารถทำงานได้หลายระบบปฏิบัติการ ปัจจุบันสามารถทำงานได้ทั้งวินโดวส์(2000, XP, Vista) และลินุกซ์ (Ubuntu, Fedora, Centos)
16. Multiple language support รองรับได้หลายภาษา
17. Integrated Help and Tutorials โปรแกรม packet tracer เวอร์ชัน 5.3.1 (ล่าสุด) ตัวเต็มมีขนาดประมาณ 73.4 Mb จะรวมเอาโปรแกรมช่วยสอนที่อยู่ในรูปแบบอะนิเมชันและคู่มือการใช้งานไว้ด้วยในตัว ทำให้ผู้เรียนสามารถศึกษาได้ด้วยตนเอง
18. Supports Networking Academy Curricula โปรแกรม packet tracer นั้นถูกออกแบบมาเพื่อใช้สำหรับโครงการ Cisco Networking Academy ซึ่งทางบริษัทซิสโก้มีนโยบายส่งเสริมให้สถาบันการศึกษาในประเทศต่างๆ ที่มีการเรียนการสอนด้านระบบเครือข่ายได้ใช้

งาน และใช้สำหรับผู้ที่ต้องการสอบ certificate CCNA (หลักสูตร CCNA Discovery, CCNA Exploration, and CCNA Security) หรือ CCNP บางส่วน

Packet Tracer ทำอะไรได้บ้าง ? (เวอร์ชัน 5.0 ขึ้นไป)

1. อุปกรณ์เราเตอร์ (Routers) รองรับอุปกรณ์เราเตอร์ตั้งแต่รุ่น 1841-2811 พร้อมกับ generic router ซึ่งผู้ใช้สามารถเพิ่มลดอุปกรณ์ได้ตามความต้องการ เช่น การ์ดเน็ตเวิร์คแบบต่าง (fiber, fast-ethernet, serial, Ethernet เป็นต้น)

2. อุปกรณ์สวิตช์ (Switches) รองรับอุปกรณ์สวิตช์ในระดับเลเยอร์ 2(2950-2960, generic switch สามารถเพิ่มลดอินเทอร์เฟสในการเชื่อมต่อได้เอง), เลเยอร์ 3 เตรียมไว้ให้ 1 ตัวคือ สวิตช์ 3560 ซึ่งมีคุณสมบัติในการ VLAN

3. ฮับ (Hubs) โปรแกรมสนับสนุนอุปกรณ์ชนิดฮับ รีพีทเตอร์ และสปริตเตอร์

4. ไวเลสแลน (wireless devices) สนับสนุนอุปกรณ์ไวเลสแลน access point ทั้งแบบมีเสาและไม่มีเสา

5. คอนเน็คชัน (connections) สนับสนุนสายนำสัญญาณที่ใช้สำหรับเชื่อมต่ออุปกรณ์เครือข่ายหลากหลายรูปแบบ เช่น สายแบบอัตโนมัติ (กรณีที่ผู้ใช้ตัดสินใจไม่ได้ว่าจะใช้สายชนิดใดในการเชื่อมต่อ), สาย (console), สายตรง (copper straight-through) เป็นต้น

6. อุปกรณ์เชื่อมต่อปลายทาง (end devices) สนันสนุนอุปกรณ์เชื่อมต่อปลายทางหลายประเภท เช่น เครื่องคอมพิวเตอร์พีซี โน๊ตบุค เซิร์ฟเวอร์ ปริ้นเตอร์ VoIP โทรศัพท์ ทีวี ไวเลส เป็นต้น

7. โครงข่ายการเชื่อมต่อ WAN (WAN emulation) สนันสนุนโครงข่ายการเชื่อมต่อจำลองในระดับ WAN เช่น โครงข่ายแบบกลุ่มเมฆ (cloud) และ DSL เป็นต้น

8. ประกอบอุปกรณ์ใช้งานเอง (custom made devices) โปรแกรมสนับสนุนให้ผู้ใช้สามารถเลือกและประกอบอุปกรณ์ได้ด้วยตนเองตามความต้องการ เช่น ต้องการเพิ่มการ์ดใหม่ให้กับเราเตอร์ เพิ่มอุปกรณ์ไวเลสแลนให้เครื่องคอมพิวเตอร์พีซีเป็นต้น

9. เชื่อมโยงเครือข่ายระหว่างผู้ใช้เข้าด้วยกัน (multiuser connection) โปรแกรมมีความสามารถในการเชื่อมต่อเครือข่ายระหว่างผู้ใช้ ผ่าน multiuser connection

10. สนับสนุนโพรโทคอล ดังตารางที่ 10.1

ตารางที่ 10.1 โพรโทคอลที่ Packet Tracer สนับสนุน

Layer	Layer Cisco Packet Tracer Supported Protocols
Application	• FTP, SMTP, POP3, HTTP, TFTP, Telnet, SSH, DNS, DHCP, NTP, SNMP, AAA, ISR VOIP, SCCP config and calls ISR command support, Call Manager Expres
Transport	• TCP and UDP, TCP Nagle Algorithm & IP Fragmentation, RTP
Network	• BGP, IPv4, ICMP, ARP, IPv6, ICMPv6, IPSec, RIPv1/v2/ng, Multi-Area OSPF, EIGRP, Static Routing, Route Redistribution, Multilayer Switching, L3 QoS, NAT, CBAL, Zone-based policy firewall and Intrusion Protection System on the ISR, GRE VPN, IPSec VPN
Network Access/Interface	• Ethernet(802.3), 802.11, HDLC, Frame Relay, PPP, PPPoE, STP, RSTP, VTP, DTP, CDP, 802.1q, PAgP, L2 QoS, SLARP, Simple WEP, WPA, EAP

11. คุณสมบัติเพิ่มขึ้นใหม่ในเวอร์ชันที่ 5.3 ดังต่อไปนี้

- รองรับหลักสูตร CCNA Discovery ซึ่งเป็นหลักสูตรที่เน้นการเชื่อมต่อเครือข่ายแบบ home networking และบริษัทขนาดเล็ก ซึ่งการเชื่อมต่อเครือข่ายไม่ซับซ้อน โดยเน้นรูปแบบการเชื่อมต่อดังต่อไปนี้
    - บริหารจัดการ อุปกรณ์ไวเลสแลน การจัดการไอพี และอุปกรณ์ปลายทาง
    - บริหารจัดการ ดีเอ็นเอส(DNS) ดีเอชซีพี(DHCP) ความปลอดภัยเครือข่ายไวเลสแลน เอพทีพี(FTP) เอสเอ็มทีพี(SMTP) และป๊อบทรี(POP3)
    - บริหารจัดการโพรโทคอล OSPF แบบ multi-area, EIGRP และ BGP

- บริหารจัดการ ISR VoIP, Call Manager Express

- รองรับหลักสูตร CCNA Exploration ซึ่งเป็นหลักสูตรที่เน้นการเชื่อมต่อเครือข่ายแบบเจาะลึกมากกว่าแบบ Discovery โดยเน้นหัวข้อดังต่อไปนี้
  - HTTP, DNS, DHCP, new FTP, SMTP, POP3
  - multiarea OSPF, EIGRP, new BGP
  - Linksys models, wireless security, 802.11
  - New PPPoE, enhanced IPSec, Cable and DSL enhancements
- รองรับหลักสูตร CCNP
  - multiarea OSPF, EIGRP, new BGP

12. คำถามที่น่าสนใจ

a. Packet tracer คืออะไร? **ตอบ** packet tracer คือโปรแกรมที่ใช้สำหรับช่วยเหลือให้ผู้เรียนด้านระบบเครือข่าย สามารถมองเห็นภาพการทำงานของระบบเครือข่ายในรูปแบบ simulation และ visualization รวมถึงมีความสามารถในการสร้างบนเรียน การเชื่อมโยงเครือข่ายระหว่างไซต์ การแก้ปัญหาเป็นทีม เป็นต้น

b. ใครใช้ Packet tracer ได้บ้าง? **ตอบ** Packet tracer สามารถใช้งานได้ฟรี สำหรับอาจารย์และนิสิตที่เรียนในหลักสูตร Cisco Networking Academy ซึ่งจะได้รับ account ให้สามารถดาวน์โหลดไปใช้งานได้ resource ทั้งหมดที่ cisco จัดเตรียมไว้ให้สำหรับหลักสูตรดังกล่าว

c. เมื่อติดตั้ง packet tracer เวอร์ชันเดิมอยู่แล้วจำเป็นต้องเปลี่ยนหรือ upgrade เป็นเวอร์ชันใหม่หรือไม่? **ตอบ** แนะนำว่าควรใช้เวอร์ชันที่ใหม่กว่า

d. สามารถใช้ activity ที่สร้างจากเวอร์ชันเก่า มาใช้กับเวอร์ชันใหม่ได้หรือไม่?
ได้ เวอร์ชันใหม่สามารถรองรับ activity ที่สร้างจากเวอร์ชันเดิมได้ทั้งหมด

e. สามารถใช้ activity ที่สร้างจากเวอร์ชันใหม่ ไปทำงานกับเวอร์ชันที่ต่ำกว่าได้หรือไม่? **ตอบ** ไม่ได้

f. ผู้สอนหรือผู้เรียนนำเอา packet tracer ไปติดตั้งใช้งานที่เครื่องส่วนตัวได้ไหม? **ตอบ** ได้ เนื่องจากหลักสูตรดังกล่าวต้องการให้ผู้เรียนผู้สอนมีความเป็นอิสระในการใช้งาน ดังนั้นจึงอนุญาตให้นำไปติดตั้งในเครื่องของตนเองเพื่อทำ LAB หรือปฏิบัติได้

g. โปรแกรม packet tracer ติดตั้งบนระบบปฏิบัติการอะไรบ้าง? **ตอบ** ติดตั้งได้ทั้งวินโดวส์และลินุกซ์ เช่น Windows (Windows XP, Windows 2000, Vista Home Basic, and Vista Home Premium) and Linux (Ubuntu 7.10 and Fedora 7)

h. เครื่องที่ต้องการติดตั้ง packet tracer จะต้องมีคุณสมบัติอย่างไร? **ตอบ** ในการติดตั้ง packet tracer เวอร์ชัน 5 ขึ้นไป เครื่องคอมพิวเตอร์ควรจะมีคุณสมบัติดังนี้คือ

- CPU: Intel Pentium 300 MHz or equivalent
- OS: Microsoft Windows 2000, Windows XP, Vista Home Basic, Vista Home Premium, Fedora 7, or Ubuntu 7.10
- RAM: 96 MB
- Storage: 250 MB of free disk space
- Screen resolution: 800 x 600 or higher
- Macromedia Flash Player 6.0 or higher
- Language fonts supporting Unicode encoding (if viewing in languages other than English)
- Latest video card drivers and operating system updates

i. Packet Tracer รองรับโพรโทคอล IGRP หรือไม่? **ตอบ** ซิสโก้ใช้โพรโทคอล EIGRP แทน IGRP ดังนั้นใน packet tracer จึงไม่มี

j. ผู้ใช้ที่ไม่ได้เรียนหลักสูตร Cisco Networking Academy จะใช้โปรแกรมหรือดาวน์โหลดโปรแกรม packet tracer ได้หรือไม่? **ตอบ** เมื่อพูดตามความหมายในลิขสิทธิ์แล้วตอบว่าไม่ได้ ต้องใช้เฉพาะผู้ที่เรียนในหลักสูตร หรือ alumni เท่านั้น แต่ถ้าผู้ที่สนใจและเห็นว่าโปรแกรมดังกล่าวมีประโยชน์ เบื้องต้นอาจดาวน์โหลดได้จากเว็บทั่วไป (google) เมื่อเห็นว่าโปรแกรมดังกล่าวมีความเหมาะสมที่จะใช้ในสถานศึกษาของตนก็ควรจะสมัครเป็นสมาชิกกับ Cisco Networking Academy เพื่อรับสิทธิประโยชน์อย่างอื่นมากมาย (สำหรับประเทศไทยสามารถอ่านข้อมูลเพิ่มเติมได้จาก http://www.cisco.com/web/TH/index.html)

แบบฝึกหัดท้ายบท

1. ซอฟต์แวร์ระบบเครือข่ายทำหน้าที่อะไร
2. อะไรคือ Simulation-Based Learning
3. ซอฟต์แวร์ Packet Tracer มีคุณสมบัติที่แตกต่างจากซอฟต์แวร์จำลองเครือข่ายของบริษัทอื่นอย่างไร
4. ทำไมเราจึงจำเป็นต้องใช้ซอฟต์แวร์จำลองเครือข่าย
5. ซอฟต์แวร์จำลองเครือข่ายทำงานแตกต่างจากการเรียนรู้จากอุปกรณ์จริงอย่างไร

บทที่ 11
การใช้งานซอฟต์แวร์ Packet Tracer

แนวคิด

ในบทนี้จะมาเรียนรู้วิธีการใช้งานของโปรแกรมจำลองเครือข่ายชื่อว่า Packet Tracer ของบริษัท Cisco เพื่อใช้ในการออกแบบและวิเคราะห์ระบบเครือข่ายได้อย่างมีประสิทธิภาพ

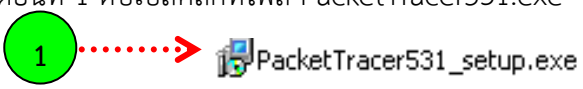
วัตถุประสงค์

1. สามารถใช้งานซอฟต์แวร์ Packet Tracer ได้อย่างมีประสิทธิภาพ
2. สามารถจัดวางอุปกรณ์บนผังเครือข่าย (workspace) ได้
3. สามารถสร้างระบบเครือข่ายได้
4. สามารถวิเคราะห์ข้อมูลเครือข่ายได้

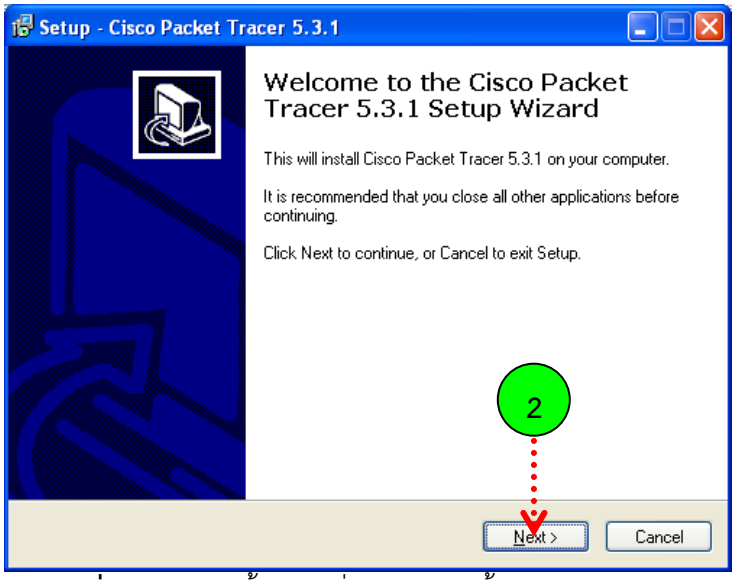
ในบทนี้จะอธิบายถึงการใช้งานโปรแกรม Packet Tracer เวอร์ชัน 5.3.1

1. ติดตั้งโปรแกรม Packet Tracer 5.3.1

- ขั้นตอนที่ 1 ดับเบิ้ลคลิกที่ไฟล์ PacketTracer531.exe

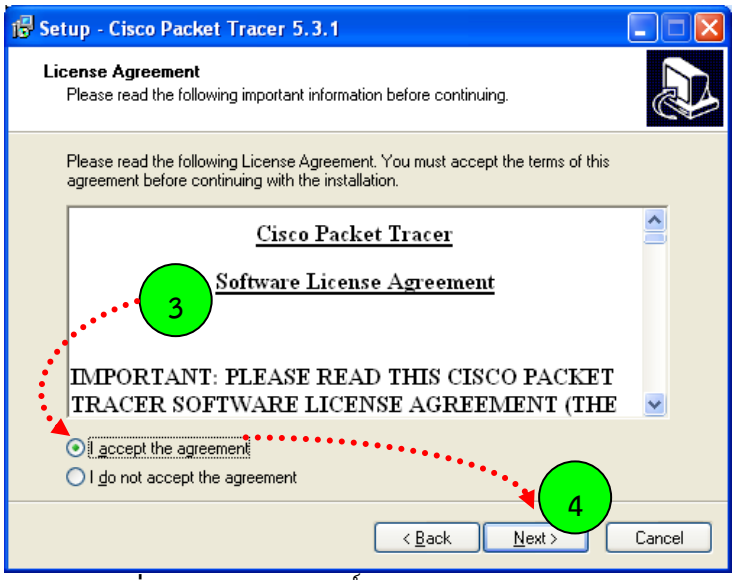


- ขั้นตอนที่ 2 โปรแกรมจะแสดงหน้าต่าง Welcome to the Cisco Packet Tracer 5.3.1 ให้คลิก Next> ดังรูป 11.1



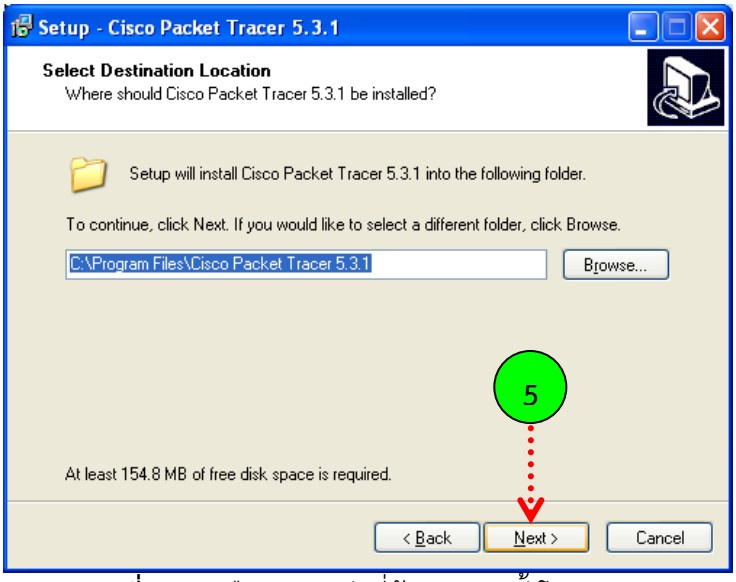
รูปที่ 11.1 แสดงขั้นตอนเริ่มต้นการติดตั้ง packet tracer

- ขั้นตอนที่ 3, 4 โปรแกรมแสดง license agreement ให้เลือก I accept a agreement ให้คลิก Next> ดังรูป 11.2



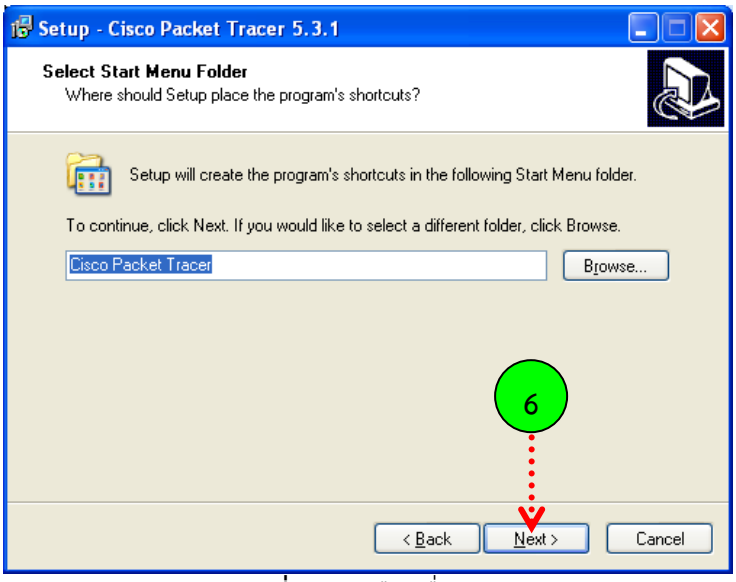
รูปที่ 11.2 แสดงลิขสิทธิ์โปรแกรม packet tracer

- ขั้นตอนที่ 5 โปรแกรมให้เลือกตำแหน่งที่ต้องการติดตั้งบนเครื่องคอมพิวเตอร์ ค่า default จะติดตั้งไว้ใน C:\Program Files ให้เลือก Next> ดังรูป 11.3



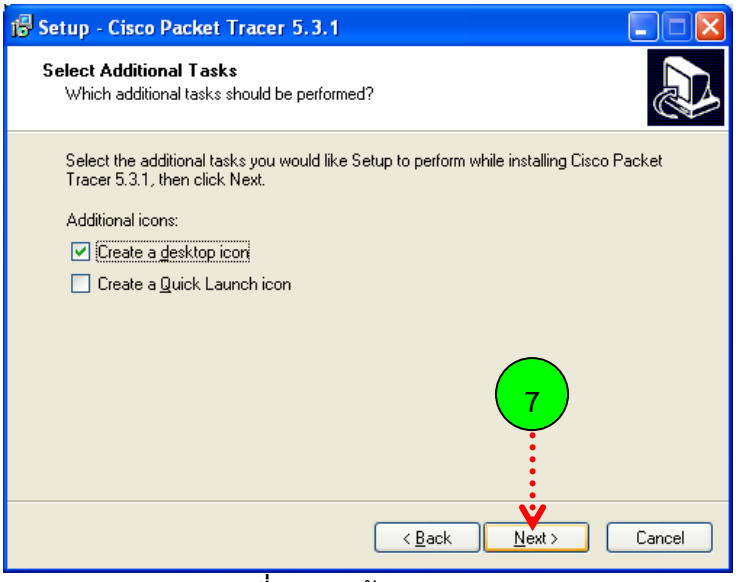
รูปที่ 11.3 เลือกตำแหน่งที่ต้องการติดตั้งโปรแกรม

- ขั้นตอนที่ 6 โปรแกรมถามว่าต้องการใช้ชื่อเมนูชื่อ Cisco Packet Tracer หรือไม่ ให้เลือก Next> ดังรูป 11.4



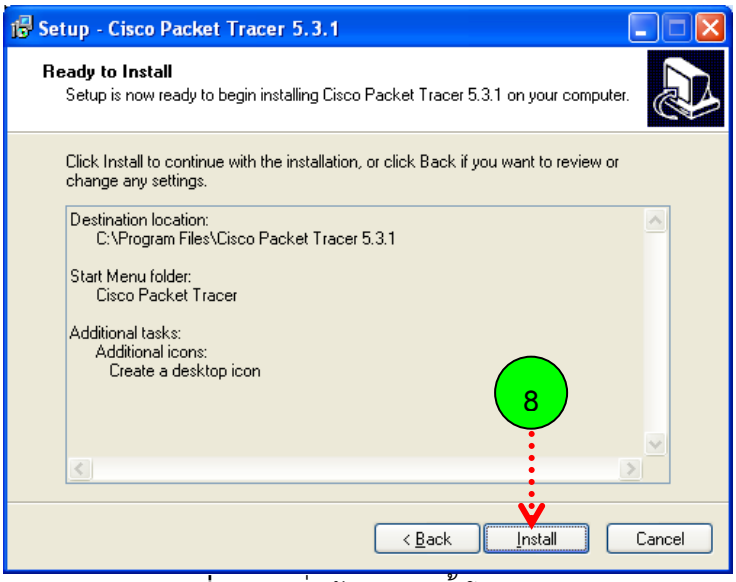
รูปที่ 11.4 เลือกชื่อเมนู

- ขั้นตอนที่ 7 select additional tasks เลือก Next> ดังรูป 11.5



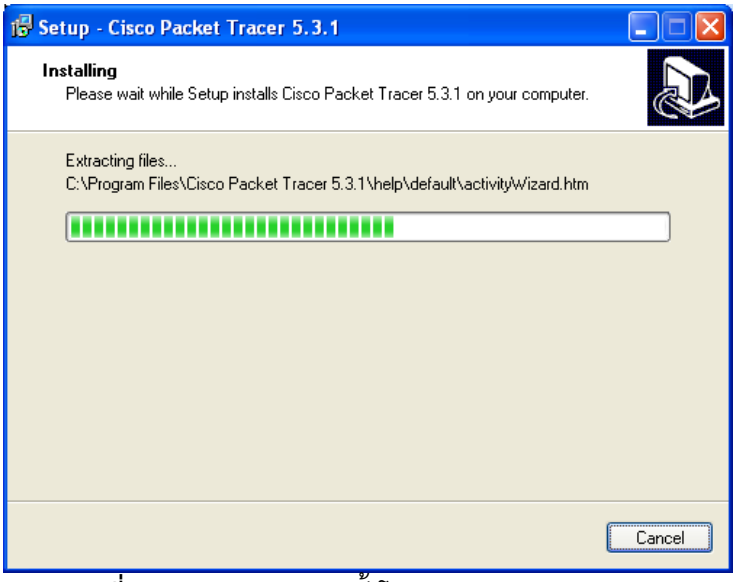
รูปที่ 11.5 สร้าง shortcut

- ขั้นตอนที่ 8 Ready to install เลือก Next> ดังรูป 11.6

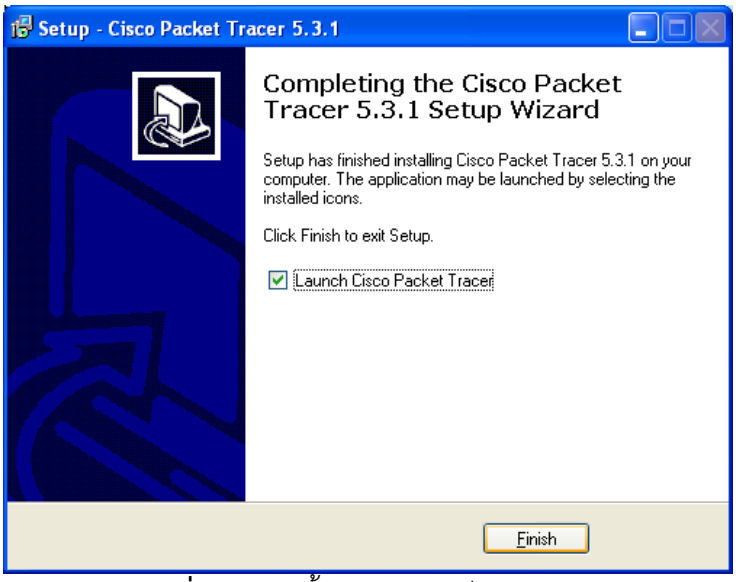


รูปที่ 11.6 เริ่มต้นการติดตั้งโปรแกรม

- ขั้นตอนที่ 9 โปรแกรมเริ่มทำการติดตั้ง โปรดรอสักครู่ เมื่อติดตั้งเสร็จ ให้เลือก Finish ดังรูป 11.7, 11.8



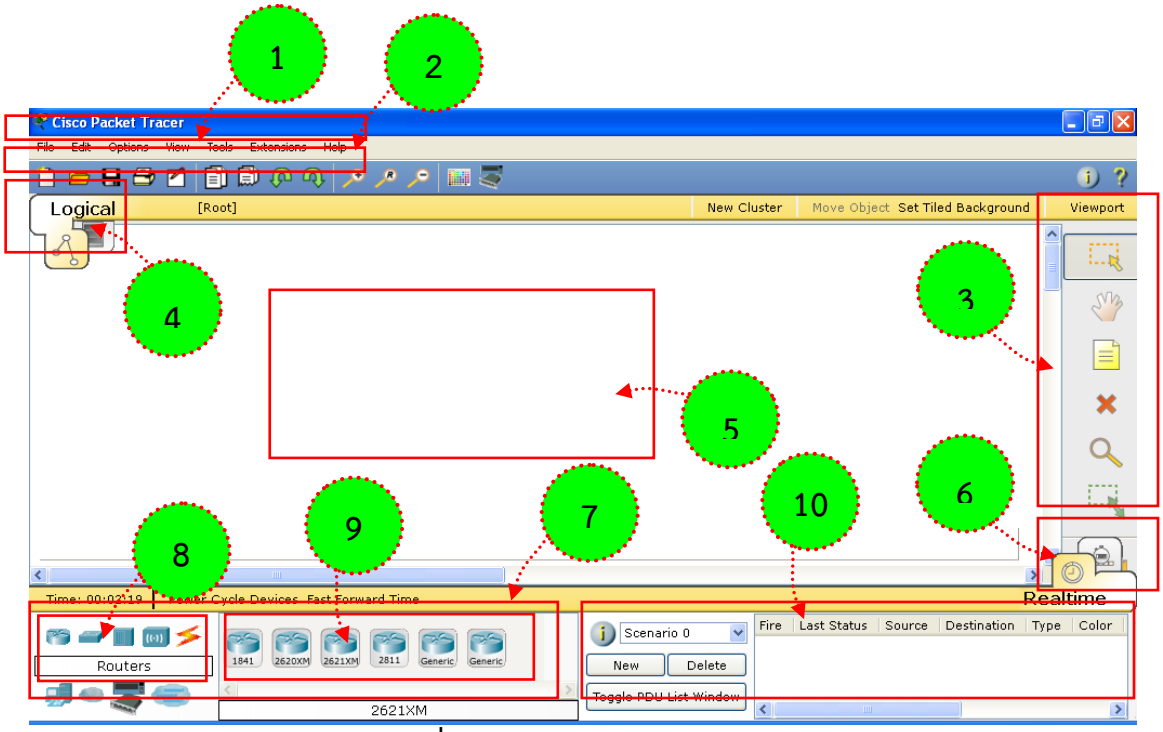
รูปที่ 11.7 ดำเนินการติดตั้งโปรแกรม packet tracer



รูปที่ 11.8 ติดตั้งโปรแกรมเสร็จเรียบร้อย

2. **เริ่มต้นการใช้งาน Packet Tracer**

เมื่อผู้ใช้ทำการติดตั้งโปรแกรม packet tracer แล้ว ต้องการเรียกใช้งานโปรแกรม ให้เลือก Start ⇨ Programs ⇨ Cisco Packet Tracer ⇨ Cisco Packet Tracer ดังรูป 11.9



รูปที่ 11.9 โปรแกรม packet tracer 5.3.1

ส่วนประกอบหลักของโปรแกรม packet tracer 5.3.1 มี 10 ส่วนดังนี้

1. Menu Bar
2. Main Tool Bar
3. Common Tools Bar
4. Logical/Physical Workspace and Navigation Bar
5. Workspace
6. Realtime/Simulation Bar
7. Network Component Box
8. Device-Type Selection Box
9. Device-Specific Selection Box
10. User Created Packet Window

- Menu Bar เป็นเมนูหลักของโปรแกรมประกอบไปด้วยเมนูย่อยๆ ดังนี้

1. เมนู File ใช้สำหรับสร้างผังเครือข่าย (logical network) ใหม่, แก้ไขผังเครือข่ายที่เคยสร้างมาแล้ว, เปิดตัวอย่างผังเครือข่ายที่มากับโปรแกรมแกรม, บันทึกผังเครือข่าย (นามสกุลเป็น .pkt), พิมพ์ผังเครือข่าย เป็นต้น

2. เมนู Edit ใช้สำหรับ แก้ไขผังเครือข่าย เช่น ก็อบปี้, วางอุปกรณ์, ยกเลิกคำสั่งเดิม เป็นต้น
3. เมนู Options ใช้สำหรับกำหนดคุณสมบัติของโปรแกรม ประกอบไปด้วย
    - Preferences ใช้สำหรับกำหนดการแสดงผลของโปรแกรม ประกอบไปด้วย แท็บ Interface, Administrator, hide และ font ตามลำดับ

ตารางที่ 11.1 แสดงคุณสมบัติในแท็บ Interface

คุณสมบัติในแท็บ Interface	
Show animation	แสดงภาพเคลื่อนไหวขณะทำงาน
Play sound	มีเสียงขณะมีการคลิกเลือกอุปกรณ์ต่างๆ
Show link lights	แสดงไฟกระพริบขณะที่อุปกรณ์กำลังเชื่อมต่อ
Show device labels	แสดงชื่อของอุปกรณ์ในผัง logical
Show port labels when mouse over	แสดงชื่อพอร์ตเมื่อเคลื่อนเมาส์ไปใกล้
Show QoS stamps on packets	แสดงข้อมูลที่ถูก stamps เมื่อข้อมูลนั้นถูกทำ QoS
Enable cable length effects	เปิดใช้งาน cable length effects
Enable auto cable	เปิดใช้งาน auto cable
Show device dialog taskbar	แสดงชื่ออุปกรณ์ใน taskbar

ตารางที่ 11.2 แสดงคุณสมบัติในแท็บ Administrator

คุณสมบัติในแท็บ Administrator	
Choose password	ตั้งรหัสผ่านค่าคอนฟิกที่ทำการปรับแต่งไว้ (ไม่ให้บุคคลอื่นมาแก้ไขค่าคอนฟิกในโปรแกรมในภายหลังได้)
Interface locking	เปิด/ปิดการใช้งานเมนูต่างๆ เช่น เมื่อคลิกเลือกที่ multiuser menu เมนูดังกล่าวจะไม่สามารถใช้งานได้
Write option to PT	อนุญาตให้ผู้ใช้งานสามารถเขียนลงใน folder ที่ติดตั้งโปรแกรม packet tracer ได้

ตารางที่ 11.3 แสดงคุณสมบัติในแท็บ Hide

คุณสมบัติในแท็บ Hide	
Customize user experience	ซ่อนเมนู เมื่อผู้สอนไม่ต้องการให้ผู้เรียนใช้งาน เช่น ซ่อนแท็บ physical, CLI, Desktop เป็นต้น

ตารางที่ 11.4 แสดงคุณสมบัติในแท็บ Font

คุณสมบัติในแท็บ Font	
Dialogs	ปรับแต่งตัวอักษร ขนาด การแสดงผลของฟอนท์ใน CLI และ headers
Workspace/activity wizard	ปรับแต่งตัวอักษร ขนาด การแสดงผลของฟอนท์ใน Workspace/activity wizard
General interface	ปรับแต่งตัวอักษร ขนาด การแสดงผลของฟอนท์ใน เมนูต่างๆ เช่น File, tooltips เป็นต้น
Colors	ปรับแต่งสีการแสดงผลในการคอนฟิกอุปกรณ์ เช่น สีของตัวอักษรขณะคอนฟิก สีพื้นหลัง เป็นต้น

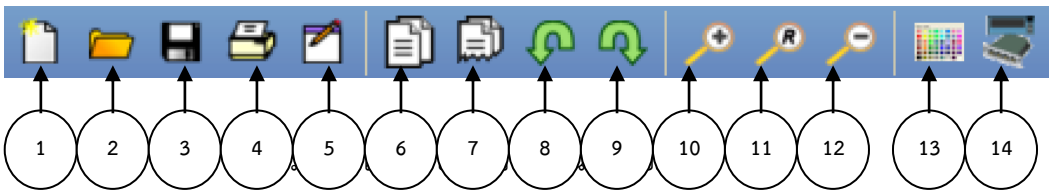
- Uses profile เป็นการกำหนดข้อมูลผู้ออกแบบและคอนฟิกเครือข่ายว่าคือใคร
- Algorithm settings เป็นการกำหนดคุณสมบัติในการเชื่อมต่อระหว่างผู้ใช้งานต่างไซต์ เช่น กำหนดจำนวน connections, sessions เป็นต้น

4. เมนู view กำหนดการซูมภาพ ให้มีขนาด เล็ก หรือ ใหญ่ตามที่ผู้เรียนต้องการ และกำหนดการแสดงผลของแท็บ toolbar
5. เมนู tools ใช้สำหรับปรับแต่งสี ของปุ่ม หรือสีเส้น ที่เชื่อมต่อ
6. เมนู extensions เป็นเมนูที่ใช้กำหนดคุณลักษณะพิเศษของโปรแกรมดังต่อไปนี้
    - Activity wizard ใช้สำหรับสร้างบทเรียนแบบ step by step (จะกล่าวอย่างละเอียดในหัวข้อ activity wizard)
    - Multiuser กำหนดคุณลักษณะการเชื่อมต่อระหว่างผู้ใช้งานหลายคน (อ่านเพิ่มในหัวข้อ multiuser)
    - IPC ใช้สำหรับอำนวยความสะดวกให้ผู้ใช้งานที่ต้องการสร้างโปรแกรมขึ้นเอง โดยทำงานร่วมกับ packet tracer

ผ่านทางโพรโทคอล IPC(Inter-Process Communication)

7. เมนู help สำหรับช่วยเหลือผู้ใช้งาน ในการใช้งานโปรแกรม packet tracer ซึ่งมีทั้งแบบมัลติมีเดีย และแบบ text

- Main Tool Bar จัดเตรียมเครื่องมือที่ใช้งานบ่อยๆ ไว้ให้กับผู้ใช้งาน ประกอบไปด้วย



ตารางที่ 11.5 แสดง Main Tool Bar

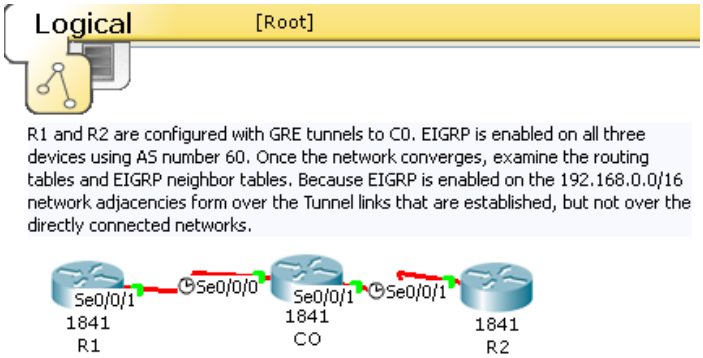
Main Tool Bar	
1. New	สร้างผังเครือข่ายใหม่
2. Open	เลือกผังเครือข่ายเดิมที่สร้างไว้ขึ้นมาทำงานต่อ
3. Save	บันทึกผังเครือข่ายที่สร้างขึ้น
4. Print	สั่งพิมพ์ผังเครือข่าย
5. Activity Wizard	สร้างบทเรียนแบบ step by step
6. Copy	คัดลอกอุปกรณ์
7. Paste	วางอุปกรณ์ที่คัดลอกลงบนผังเครือข่าย
8. Undo	ยกเลิกคำสั่งที่ทำงานหลังสุด
9. Redo	ทำคำสั่งหลังสุดอีกครั้ง
10. Zoom In	ขยายผังเครือข่าย
11. Zoom Reset	คืนสภาพของผังเครือข่ายให้มีขนาดเป็นค่า default
12. Zoom Out	ย่อผังเครือข่าย
13. Drawing Palette	ปรับแต่งสี ของปุ่ม หรือสีเส้น ที่เชื่อมต่อ
14. Custom Devices Dialog	เพิ่มลด template ใหม่

- Common Tools Bar จัดเตรียมเครื่องมือที่ใช้งานบ่อยๆ กับพื้นที่ตรงส่วน workspace ซึ่งเป็นพื้นที่ที่ใช้เขียนผัง

ตารางที่ 11.6 แสดง Common Tools Bar

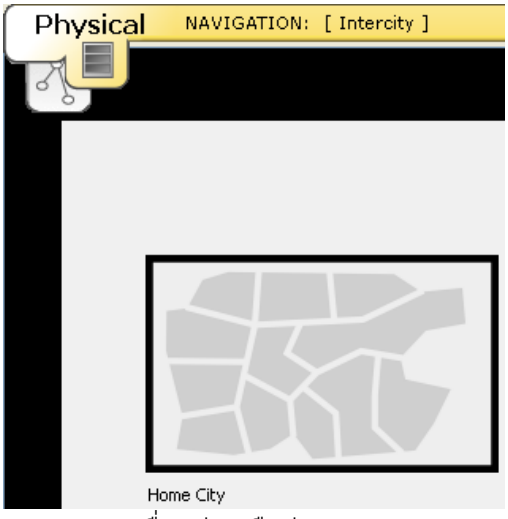
Common Tools Bar	
	เลือกอุปกรณ์
	เคลื่อนย้ายผังเครือข่าย
	บันทึกย่อ หรือ note บนผังเครือข่าย
	ลบอุปกรณ์ที่เลือก
	ตรวจสอบคุณสมบัติของอุปกรณ์ เช่น MAC table, ARP table เป็นต้น
	ลดและเพิ่มขนาดของอุปกรณ์
	ทดสอบการทำงานของเครือข่ายโดยเพิ่ม packet เข้าไปทดสอบ เช่น ping
	เพิ่ม packet เข้าไปทดสอบที่ซับซ้อนขึ้น

- Logical/Physical Workspace and Navigation Bar จัดเตรียมพื้นที่ใช้งานสำหรับสร้างผังเครือข่าย 2 แบบคือ logical(แสดงการเชื่อมต่อโดยใช้สัญรูปสัญลักษณ์) และ physical(แสดงการเชื่อมต่อทางกายภาพ เช่น สถานที่ติดตั้ง, Rack Cabinet เป็นต้น) ดังรูปที่ 11.10



รูปที่ 11.10 แสดงการเชื่อมต่อเครือข่ายบน logical workspace

ในพื้นที่ logical workspace ผู้ใช้สามารถจัดกลุ่มอุปกรณ์ (cluster), กำหนดภาพพื้นหลัง, แสดงเครือข่ายแบบ view point ได้ ดังรูปที่ 11.11



รูปที่ 11.11 แสดงการเชื่อมต่อเครือข่ายบน physical workspace

physical workspace สามารถแสดงแผนที่ตั้งในลักษณะแบบ navigation คือ เริ่มจากพื้นที่ขนาดใหญ่เช่น เมือง ⇨ อำเภอ ⇨ ตำบล ⇨ ตึก ⇨ ห้องเครือข่าย ⇨ ตู้ Rack หรือ ย้อนกลับจาก ตู้ Rack ไปถึง เมืองก็ได้เช่นกัน

- Workspace เป็นพื้นที่ตรงส่วนกลางโปรแกรมที่ใช้สำหรับสร้างผังเครือข่าย
- Realtime/Simulation Bar ผู้ใช้สามารถเลือกโหมดในการทำงานได้ 2 แบบคือ โหมด Realtime แสดงการเชื่อมต่อแบบปกติ อุปกรณ์ทุกตัวจะแสดงสถานการทำงานโดยมีลักษณะกระพริบ โหมด Simulation แสดงทิศทางการไหลของข้อมูลชนิดของแพ็คเก็ต ในรูปแบบอนิเมชัน ดังรูปที่ 11.12

โหมด Realtime โหมด Simulation

รูปที่ 11.12 แสดงการเลือกโหมด Realtime/Simulation

- Network Component Box จัดเตรียมอุปกรณ์ทั้งหมด รวมไว้ให้ผู้ใช้นำไปเชื่อมต่อเป็นโครงสร้างเครือข่าย เช่น เราเตอร์ สวิตช์ สายนำสัญญาณ เป็นต้น
- Device-Type Selection Box แสดงกลุ่มของอุปกรณ์ เช่น กลุ่มอุปกรณ์เราเตอร์ สวิตช์ สายนำสัญญาณ ฮับ ไวเลสแลน เป็นต้น
- Device-Specific Selection Box แสดงรายการของอุปกรณ์ในแต่ละกลุ่ม ซึ่งสอดคล้องกับ Device-Type เช่น เมื่อผู้ใช้เลือก Device-Type เป็น เราเตอร์ ใน Device-Specific จะปรากฏรายการรุ่นต่างๆ ของเราเตอร์ ขึ้นมาให้ผู้ใช้เลือกใช้งาน
- User Created Packet Window เมื่อผู้ใช้ต้องการทดสอบระบบเครือข่าย โปรแกรม packet tracer เตรียมเครื่องมือในการอำนวยความสะดวกให้คือ add

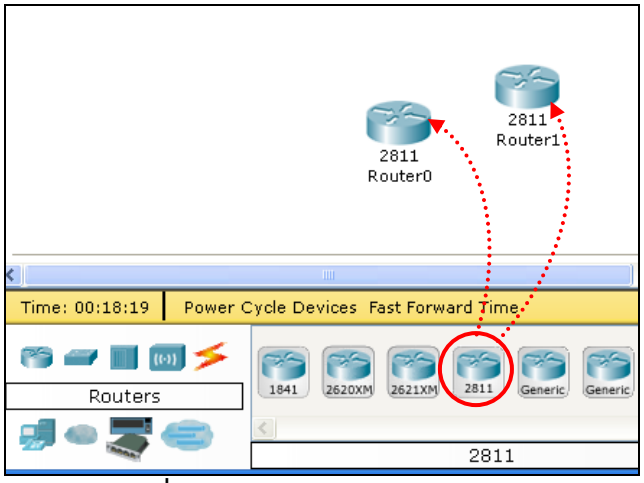
PDU (สร้าง packet โดยผู้ใช้เอง) เข้าไปในระบบเครือข่ายที่สร้างไว้ ผู้ใช้สามารถดูข้อมูลที่ประมวลผลได้จากพื้นที่ตรงส่วน User Created Packet Window

3. การจัดวางอุปกรณ์บนผังเครือข่าย (workspace)

ในการจัดวางอุปกรณ์ต่างๆ บนผังเครือข่าย ผู้ใช้งานสามารถทำได้ง่ายๆ โดยการเลือกอุปกรณ์ที่ต้องการใช้งานวางลงบนพื้นที่ workspace ในโหมด Realtime

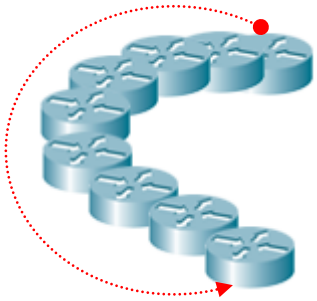
- การวางอุปกรณ์เราเตอร์

  เลือก ไอคอน Routers ตรงส่วน Device-Type (หรือกด CTRL+ALT+R พร้อมกัน) จะปรากฏไอคอนรายการเราเตอร์รุ่นต่างๆ ในส่วน Device-Specific ให้ผู้ใช้เลือกรุ่นของเราเตอร์ที่ต้องการแล้วลากนำไปวางบน workspace ได้ทันที ดังรูปที่ 11.13



รูปที่ 11.13 การวางอุปกรณ์เราเตอร์

เมื่อผู้ใช้ต้องการจัดวางตำแหน่งของอุปกรณ์บน workspace ให้เลือก select (หรือกดปุ่ม esc) ในส่วน Common Tools Bar ด้านขวามือ แล้วคลิกลากอุปกรณ์ไปยังตำแหน่งที่ต้องการได้ทันที ดังรูปที่ 11.14



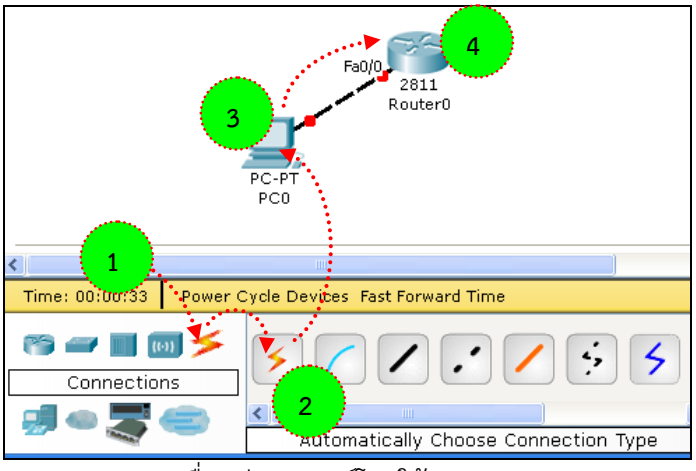
รูปที่ 11.14 การจัดวางอุปกรณ์บน workspace

เมื่อต้องการลบอุปกรณ์ออกจาก workspace ให้ผู้ใช้เลือก Delete (หรือกดปุ่ม del) ในส่วน Common Tools Bar ด้านขวามือ แล้วนำมาวางทับอุปกรณ์ อุปกรณ์ตัวดังกล่าวก็จะถูกลบทันที, สำหรับการเคลื่อนย้ายผังของเครือข่ายทั้งหมด ให้ผู้ใช้

เลือก select แล้วลากเส้นประล้อมรอบผังเครือข่ายทั้งหมดก่อน จากนั้นเลือก แล้วลากไปยังตำแหน่งที่ต้องการ, ถ้าต้องการเขียน comment หรือคำอธิบายสั้นๆ ให้เลือก แล้วไปวางยังจุดที่ต้องการอธิบายพร้อมกับเขียนคำอธิบายลงไป, สำหรับการย่อผังเครือข่ายให้เล็กหรือใหญ่ ผู้ใช้สามารถทำได้โดยการเลือกผังเครือข่ายทั้งหมดก่อนด้วย select แล้วเลือก แล้วย่อขยายผังตามความต้องการ

- การเชื่อมต่ออุปกรณ์

  สำหรับการเชื่อมต่ออุปกรณ์ผู้ใช้จำเป็นต้องเข้าใจถึงสายนำสัญญาณแต่ละประเภทว่าใช้งานอย่างไร เช่น สาย straight through ใช้สำหรับเชื่อมระหว่างอุปกรณ์ปลายทาง เช่น คอมพิวเตอร์ โน๊ตบุค ปริ้นเตอร์ แฟรกซ์ เป็นต้น เข้ากับสวิตซ์ ซึ่งบางครั้งผู้ใช้ใหม่อาจจะไม่คุ้นเคย หรือไม่ทราบ ดังนั้น packet tracer จึงเตรียมเครื่องมือเอาไว้ช่วยเหลือคือ (Automatically choose connection type) เครื่องมือดังกล่าวจะช่วยเลือกชนิดของสายที่ใช้ในการเชื่อมต่อที่เหมาะสมสำหรับอุปกรณ์แต่ละประเภทให้อัตโนมัติ



รูปที่ 11.15 แสดงการเชื่อมต่ออุปกรณ์โดยใช้ automatically connection

จากรูปที่ 11.15 แสดงการเชื่อมต่ออุปกรณ์ปลายทางคือ PC เข้ากับเราเตอร์ผ่านพอร์ต FastEthernet 0/0 ซึ่งต้องใช้สายประเภท cross over (สายชนิดไขว้) ขั้นที่ 1 ให้เลือก ในส่วน Device-Type ขั้นที่ 2 ให้เลือก automatically choose connection type ในส่วน Device-Specific ขั้นที่ 3 คลิกเลือกที่ตัวอุปกรณ์ (PC) แล้วลากไปยังเราเตอร์ โปรแกรมจะเลือกสายสัญญาณให้เองโดยอัตโนมัติ

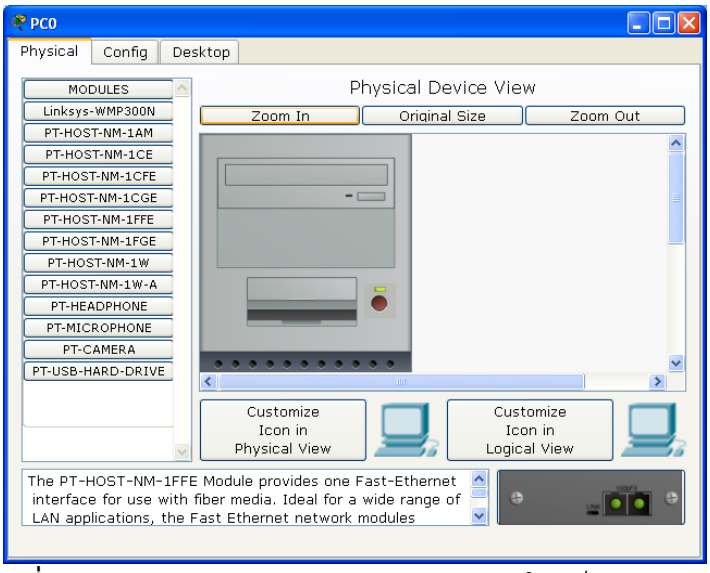
- การคอนฟิกหรือปรับแต่งอุปกรณ์

  สำหรับการคอนฟิกคุณสมบัติเพิ่มเติมของอุปกรณ์ สามารถทำได้โดยการดับเบิ้ลคลิกที่ตัวอุปกรณ์ได้โดยตรง ซึ่งจะปรากฏแท็บคล้ายๆ กัน 3 แท็บคือ แท็บ Physical,

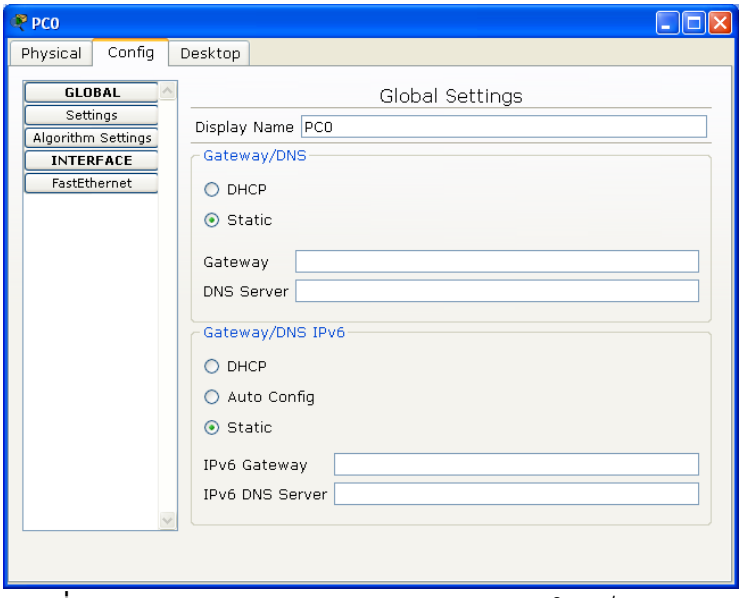
Config, Desktop (เมื่อเป็นอุปกรณ์ปลายทาง เช่น PC, Server) หรือ CLI (กรณีที่เป็นอุปกรณ์เครือข่าย เช่น สวิตช์ เราเตอร์)

การคอนฟิก PC และ Server

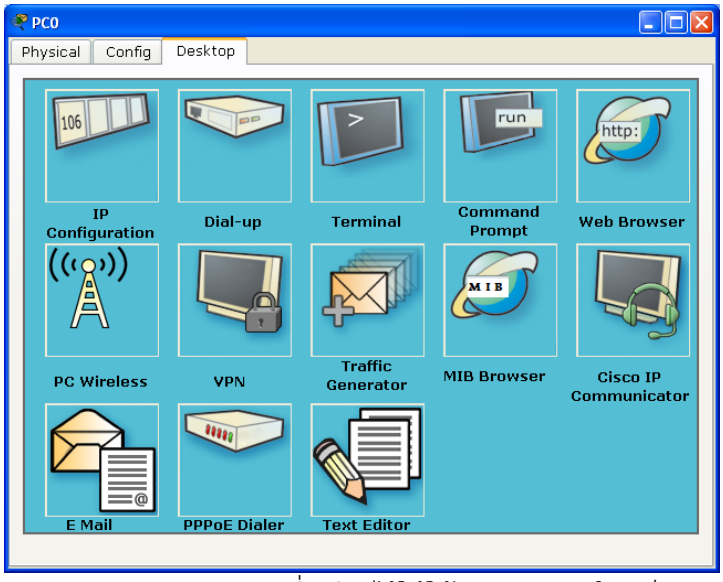
การคอนฟิก PC และ Server สามารถทำได้โดยดับเบิ้ลคลิกที่ตัวอุปกรณ์ ในแท็บ Physical จะกำหนดคุณสมบัติทางด้านกายภาพ เช่น ชนิดของการ์ดเน็ตเวิร์ค แบบต่างๆ เช่น การ์ดโทรศัพท์ FastEthernet, Gigabit เป็นต้น ในแท็บ Config จะกำหนดคุณสมบัติในการเชื่อมต่อ เช่น หมายเลขไอพี subnet, gateway เป็นต้น แท็บ Desktop จัดเตรียมเครื่องมือต่างๆ (ในระดับแอพพลิเคชัน) ที่จำเป็นสำหรับทดสอบระบบเครือข่าย เช่น terminal, command prompt, web browser, wireless, e-mail เป็นต้น ดังรูปที่ 11.16, 11.17, 11.18 ตามลำดับ



รูปที่ 11.16 แสดงการกำหนดคุณสมบัติของ PC ในแท็บ Physical



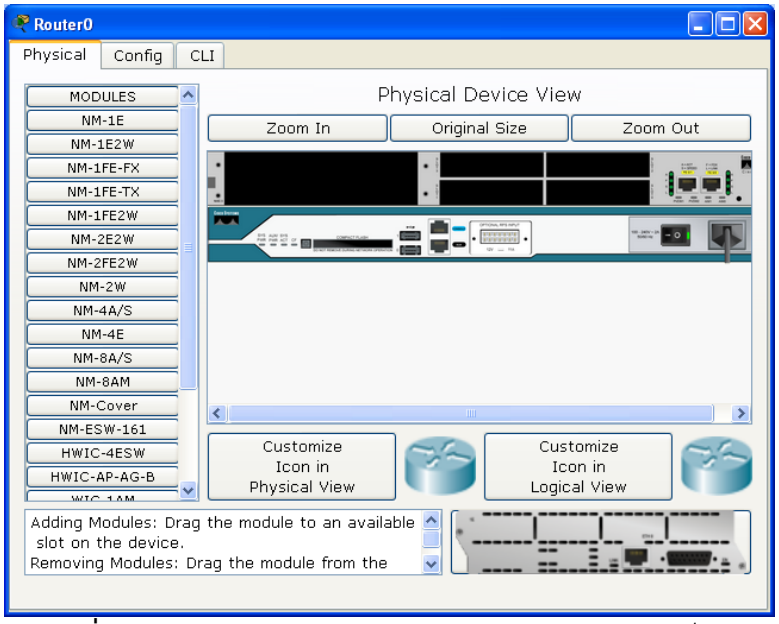
รูปที่ 11.17 แสดงการกำหนดคุณสมบัติของ PC ในแท็บ Config



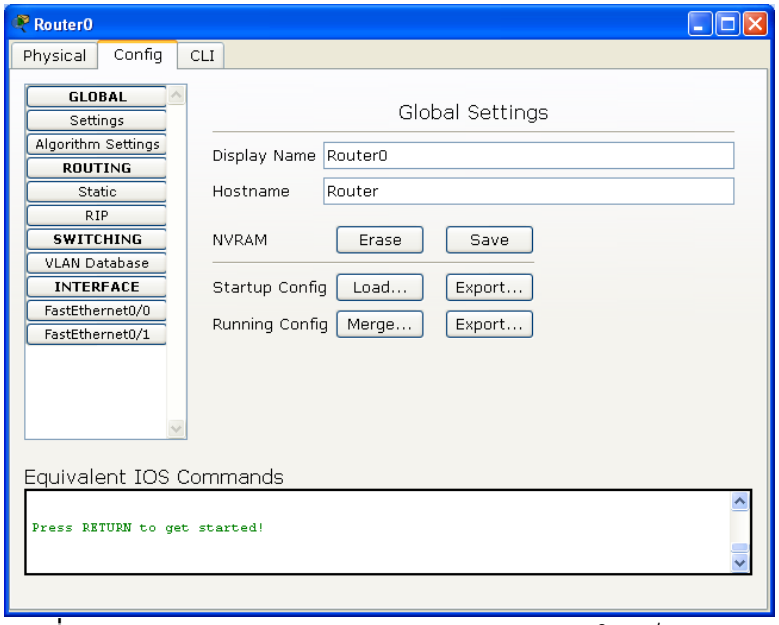
รูปที่ 11.18 แสดงแอพพลิเคชันที่เตรียมไว้ให้ใช้งานของ PC ในแท็บ Desktop

การคอนสวิตช์หรือฟิกเราเตอร์

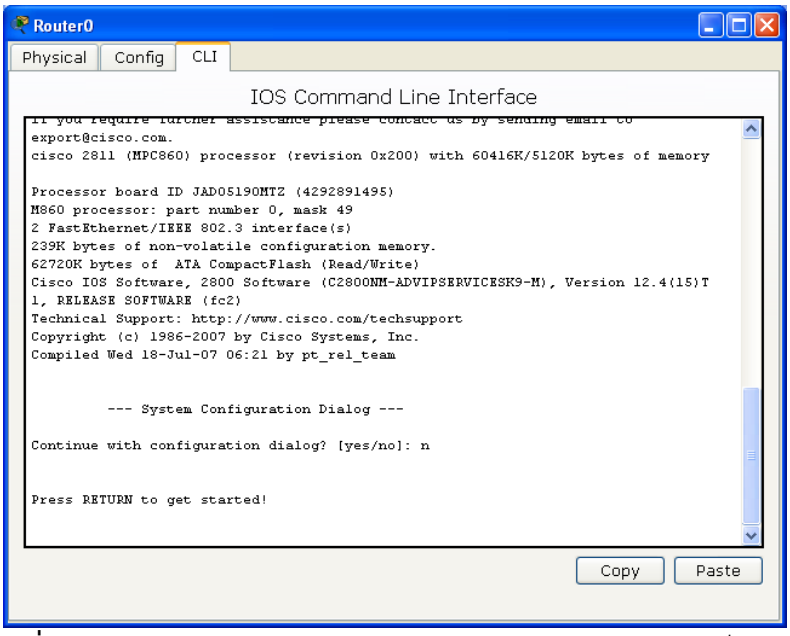
การคอนฟิกสวิตช์หรือเราเตอร์จะคล้ายกัน คือทำได้โดยดับเบิ้ลคลิกที่ตัวอุปกรณ์ ในแท็บ Physical จะกำหนดคุณสมบัติทางด้านกายภาพ เช่น ชนิดของการ์ดเน็ตเวิร์ค แบบต่างๆ การ์ดโทรศัพท์ FastEthernet, Gigabit เป็นต้น ในแท็บ Config จะกำหนดคุณสมบัติในการเชื่อมต่อ เช่น หมายเลขไอพีของแต่ละอินเทอร์เฟส, โพรโทคอลเราติ้ง, VLAN เป็นต้น แท็บ CLI เป็นการควบคุมและปรับแต่งสวิตช์หรือเราเตอร์โดยใช้การป้อนคำสั่งครั้งละ 1 บรรทัด (command line) ดังรูปที่ 11.19, 11.20, 11.21 ตามลำดับ



รูปที่ 11.19 แสดงการกำหนดคุณสมบัติของ Router ในแท็บ Physical



รูปที่ 11.20 แสดงการกำหนดคุณสมบัติของ Router ในแท็บ Config



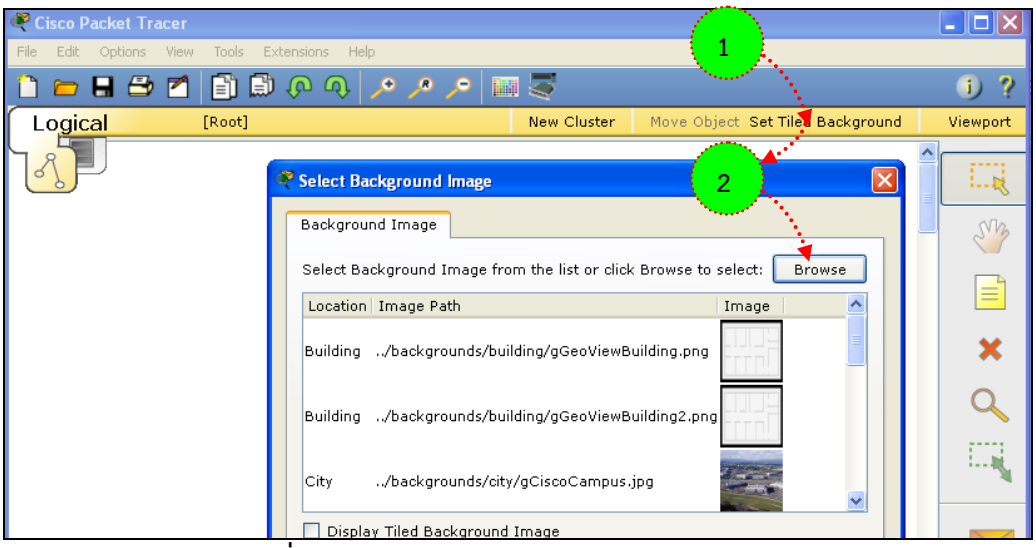
รูปที่ 11.21 แสดงการคอนฟิกเราเตอร์ด้วย command line ในแท็บ CLI

- การกำหนดพื้นหลังให้ผังเครือข่าย

  ในโหมด Realtime แท็บ Set Tield Background ผู้ใช้สามารถกำหนดภาพของ background ได้เอง ซึ่งจะช่วยให้ผู้ออกแบบสามารถนำผังเครือข่ายในทาง logical วางทับซ้อนเข้ากับแผนที่เมืองหรือสำนักงานจะทำให้เห็นภาพระบบเครือข่ายได้ชัดเจนขึ้น สำหรับขั้นตอนการกำหนด background มีดังนี้

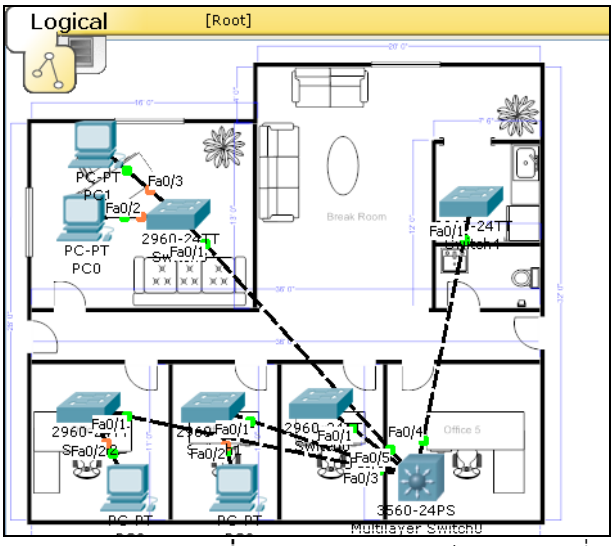
  1. เลือกแผนที่ที่จะใช้วางระบบเครือข่ายจริง เช่น แผนที่เมือง ตึกหรืออาคาร ผังห้องภายในอาคารเป็นต้น (ควรเป็นนามสกุล .jpg หรือ .png)

2. ในแท็บ Logical เลือก Set Tiled Background ⇨ Browser ⇨ เลือกภาพแผนที่เตรียมไว้ ดังรูปที่ 11.22



รูปที่ 11.22 กำหนด background ในผังเครือข่าย

3. วางอุปกรณ์ลงในแผนที่ตามความต้องการ ดังรูปที่ 11.23



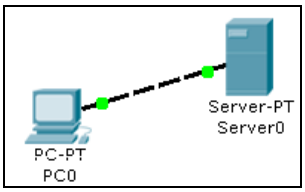
รูปที่ 11.23 วางอุปกรณ์ลงบนแผนที่

4. Creating a First Network

การออกแบบระบบเครือข่ายแต่เดิมนั้นผู้ออกแบบจะต้องเขียนผังในกระดาษ หรือเขียนด้วยโปรแกรมที่ใช้สำหรับออกแบบเครือข่ายโดยเฉพาะ เช่น Smart Draw, Visio, EDraw เป็นต้น จากนั้นก็ทดสอบติดตั้งจริง ผลของการติดตั้งจริงอาจจะไม่ตรงกับที่ออกแบบไว้ อาจเนื่องมาจากหลายสาเหตุ เช่น ผู้ใช้ปรับเปลี่ยนความต้องการ สถานที่ติดตั้งไม่อำนวย ปัญหากับโครงสร้างของอาคาร หรือไม่สามารถเข้ากันได้กับเครือข่ายเดิมที่ติดตั้งไว้แล้ว เป็นต้น ทำให้ผู้ออกแบบจะต้องมาปรับแก้ผังเดิมที่ได้ออกแบบไว้ให้เข้ากับงานจริงที่ได้ติดตั้ง ซึ่งเมื่อเทียบกับการออกแบบเครือข่ายในปัจจุบัน จะประหยัดเวลาได้มากเนื่องจาก การ

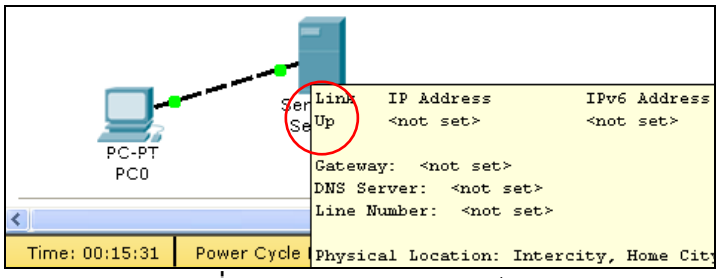
ออกแบบและการทดลองเชื่อมต่อปัจจุบัน เบื้องต้นนั้นจะทำผ่านโปรแกรมจำลองเครือข่ายได้เลย ทำให้ลดขั้นตอนและเวลาได้มาก และประโยชน์ที่เห็นได้ชัดอีกประการหนึ่งคือ ผู้ออกแบบสามารถสร้างเครือข่ายจำลอง (Logical Network Prototype) ให้ลูกค้าเห็นร่วงหน้าก่อนได้ ว่าจะออกมาเป็นลักษณะอย่างไร (คล้ายกับสร้างบ้านตัวอย่างให้ผู้ซื้อได้เห็นก่อนนั่นเอง) เรามาเริ่มสร้างเครือข่ายแรกกันดีกว่าครับ ดังรูปที่ 11.24

- เลือกอุปกรณ์เชื่อมต่อปลายทางในส่วน Device-Type เป็น End Devices ⇨ Device-Specific เลือก Generic PC และ Generic Server ⇨ ลากไปวางไว้ในส่วน workspace
- เลือก Connections ⇨ Copper Straight Through (เส้นสีดำไม่มีเส้นประ) ⇨ คลิกขวาที PC เลือก FastEthernet ⇨ ลากไปที่ Server คลิกขวา แล้วเลือก FastEthernet



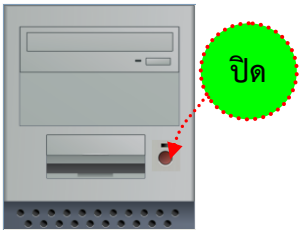
รูปที่ 11.24 เชื่อมต่อ PC กับ Server ด้วยสายแบบ Cross Over

- จากรูป 11.24 เมื่อใช้สายแบบ Copy Straight Through ลิงค์ที่เชื่อมต่อจะแสดงสถานะเป็นสีแดงแสดงว่า การเชื่อมต่อยังไม่สำเร็จ (เลือกสายเชื่อมต่อผิด) ให้เปลี่ยนเป็นสายแบบ Copper Cross Over แทน สถานะของลิงค์จะเป็นสีเขียว แสดงว่าใช้งานได้แล้ว เมื่อเลื่อนเมาส์เข้าไปใกล้เครือ PC หรือ Server จะมี message ว่าลิงค์ up ดังรูปที่ 11.25

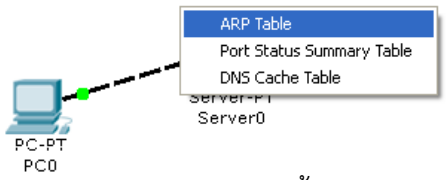


รูปที่ 11.25 แสดงสถานะลิงค์ up

- ทดสอบปิดเครื่องโดยการดับเบิ้ลคลิกที่เครื่อง PC หรือ Server ในแท็บ Physical กดปุ่ม power switch ลิงค์จะ down



- ทดสอบโดยการเปิด power switch อีกครั้ง ลิงค์จะกลับมาเป็น up อีกครั้ง จากนั้นทดลองตรวจสอบค่าของเครื่อง PC และ Server โดยใช้ Inspect คลิกที่เครื่องทั้งสอง ให้ทำการตรวจสอบ ARP Table, Port Status Summary Table และ DNS Cache Table



จากการทดสอบ ตาราง ARP, Port, DNS ในเบื้องต้นจะไม่มีค่าใดๆ เนื่องจากยังไม่มีการคอนฟิกเครื่อง PC และ Server

- ดับเบิ้ลคลิกเครื่อง PC ในแท็บ Config ให้ทดลองเปลี่ยนค่า Display name เป็น Pacman และ DNS Server เป็น 192.168.0.105, ที่ Interface เลือกอินเทอร์เฟสชนิด FastEthernet ให้กำหนด IP Address เป็น 192.168.0.110 แล้วทดสอบโดยใช้ Inspect (ยกเลิก Inspect ให้กดปุ่ม esc) อีกครั้ง ที่ Port Status จะปรากฏหมายเลข MAC Address และ IP Address เมื่อต้องการกำหนดค่า Bandwidth, Duplex, DHCP, IPv6 ก็สามารถกำหนดได้ในเมนูนี้เช่นเดียวกัน การกำหนดค่า IP Address, Subnet Mask, Default Gateway, DNS Server สามารถกำหนดได้ ในแท็บ Desktop ⇨ IP Configuration ได้เช่นเดียวกัน
- ขั้นตอนต่อไป ให้แก้ไขคอพฟิกเครื่อง Server โดยการดับเบิ้ลคลิกที่ Server ⇨ Config ⇨ เปลี่ยนชื่อในช่อง Display Name เป็น Web Server ⇨ FastEthernet กำหนด IP Address เป็น 192.168.0.105 ⇨ ตรวจสอบ Port Status เป็น On ⇨ เลือก DNS แท็บ กำหนดในช่อง Name เป็น www.firstlab.com type เป็น A Record และ ช่อง Address เป็น 192.168.0.105 ⇨ คลิก Add สุดท้ายอย่าให้ตรวจสอบว่า DNS Service มีสถานะเป็น On หรือไม่
- ตรวจสอบเครื่อง Server อีกครั้ง โดยใช้ Inspect
- ขั้นตอนสุดท้ายให้ทำการบันทึกผังเครือข่ายโดยเลือก เมนู File ⇨ Save As.. ⇨ เลือกตำแหน่งที่ต้องการบันทึก ตั้งชื่อเป็น Lab11-1.pkt แสดงว่าเครือข่ายแรกเสร็จเรียบร้อยแล้วครับ

ทดลองสร้างข้อมูล (Add Simple PDU) เพื่อทดสอบการเชื่อมต่อในโหมด Realtime

- คลิกเลือก Add Simple PDU ทางด้านขวามือ ซึ่งหมายถึงโพรโทคอลชนิดหนึ่งที่ใช้สำหรับทดสอบเครื่องปลายทางว่าทำงานอยู่หรือไม่ (เรียกว่า ping message

หรือเรียกว่า echo request) คลิกที่เครื่อง PC 1 ครั้ง และคลิกที่เครื่อง Server อีก 1 ครั้ง เมื่อ ping สำเร็จจะมี message ที่เรียกว่า echo reply ตอบกลับมาจาก Server ให้ดูผลลัพธ์การทำงานได้ในส่วน User Created Packet Window อยู่ด้านขวาล่าง

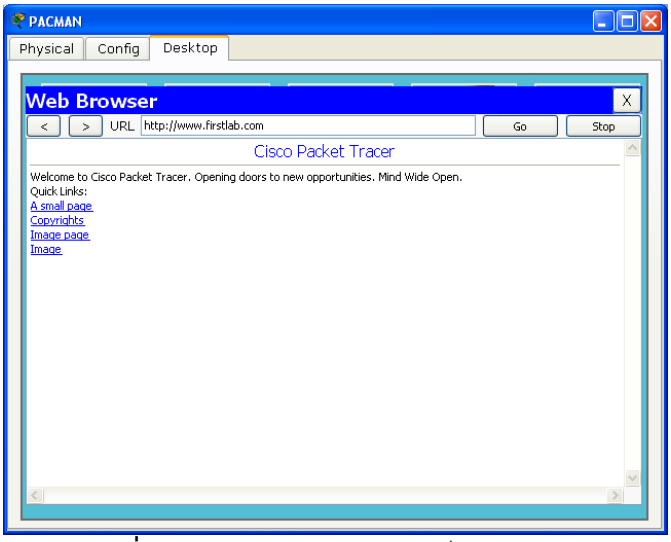
Fire	Last Status	Source	Destination	Type	Color
	Successful	PACMAN	Web Server	ICMP	
	Successful	PACMAN	Web Server	ICMP	

โปรแกรมจะปราแสดงผลลัพธ์การ ping ใน Scenario 0 เสมอ ถ้าต้องการทดสอบผลลัพธ์แต่ละครั้งแยกกันให้สร้าง Scenario ใหม่ เช่น Scenario 1, 2, 3 (เพิ่ม scenario ใหม่โดยกดปุ่ม New และกดปุ่ม Delete เพื่อลบ) ตามลำดับ สำหรับข้อมูลที่แสดงใน Scenario จะแสดง Last Status ว่า Successful แปลว่าการทำงานสำเร็จ (จากตัวอย่าง Source=Pacman, Destination=Web Server, Type=ICMP เป็นต้น) ลองทดสอบ ping กลับทิศทางอีกครั้งว่าเป็นอย่างไร

- สิ้นสุดการทดสอบด้วยการใช้ Simple PDU (ping request/reply)

ทดสอบการทำงานของเว็บเซิร์ฟเวอร์ โดยใช้โปรแกรม Browser ที่ฝั่งไคล์แอนท์

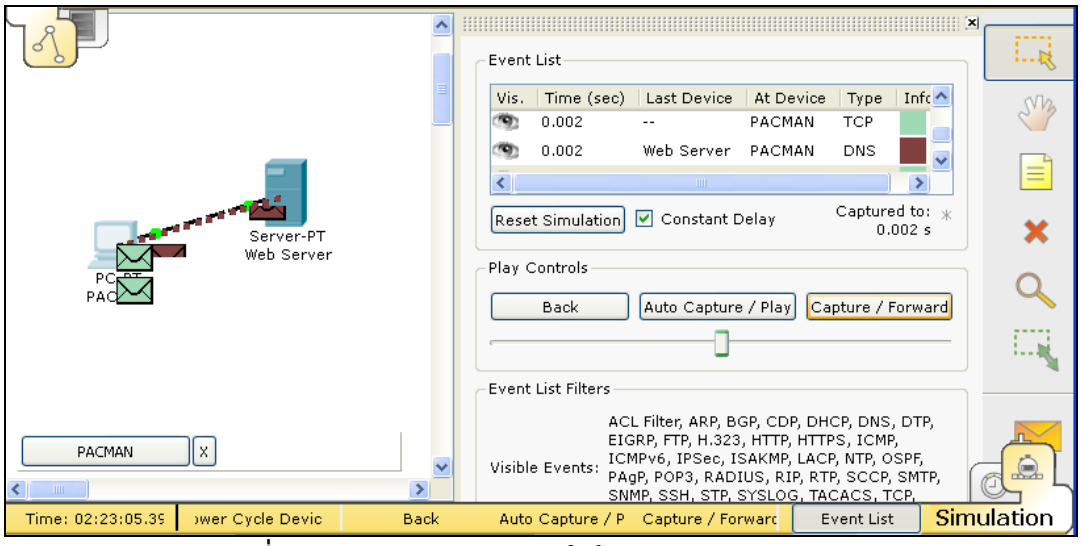
- คลิกที่เครื่อง PC เลือกแท็บ Desktop ⇨ เลือก Web Browser ⇨ ในช่อง URL ใส่ข้อมูลเป็น www.firstlab.com เสร็จแล้วกดปุ่ม go โปรแกรมจะแสดงข้อมูลใน Web Browser ดังรูปที่ 11.26



รูปที่ 11.26 ทดสอบการใช้งานเว็บเซิร์ฟเวอร์

- ทดสอบอีกครั้งโดยการป้อนข้อมูลใน URL www.abc.com แล้วกดปุ่ม go โปรแกรม Browser จะแสดง Host Name Unresolved แสดงว่าไม่สามารถค้นหาเว็บเซิร์ฟเวอร์ดังกล่าวได้

- ทดลองอีกครั้งโดยการใส่ข้อมูลในช่อง URL เป็นหมายเลข IP Address ของเครื่องเว็บเซิร์ฟเวอร์ คือ 192.168.0.105 แล้วกดปุ่ม go ผลปรากกฏว่าสามารถแสดงผลได้ถูกต้อง
- ทดลองเปลี่ยนโหมดการทำงานเป็น Simulation ซึ่งในโหมดนี้ผู้ใช้สามารถควบคุมเวลาการทำงานได้ ส่งผลให้เวลาในการทำงานของโปรแกรมจะช้ากว่าปกติ และผู้ใช้ก็สามารถสังเกตพฤติกรรมของข้อมูลได้ชัดเจน เลือกที่เครื่อง PC ⇨ เลือกแท็บ Desktop ⇨ Web Browser ⇨ ใส่ในช่อง URL เป็น www.firstlab.com กดปุ่ม go ⇨ กลับไปยัง workspace ⇨ สังเกตที่ Even List จะปรากฏ โพรโทคอล DNS อยู่ และมีซองจดหมายปรากฏบนเครื่อง PC ด้วย ⇨ เลือก Auto Capture/Play เมื่อต้องการเฝ้าดูแพ็คเก็ตแบบต่อเนื่อง หรือเมื่อต้องการเฝ้ามองทีละ step ให้เลือก Capture/Forward ดังรูปที่ 11.27



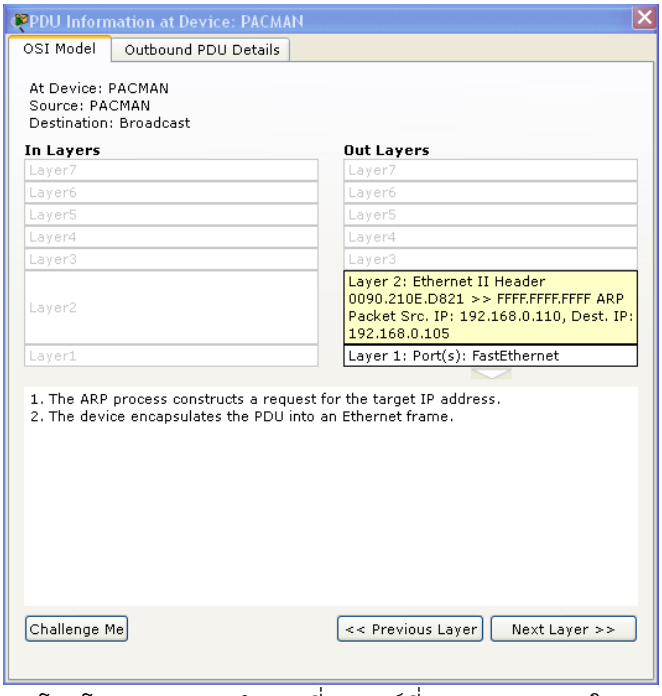
รูปที่ 11.27 ทดสอบการทำงานในโหมด simulation

สังเกตว่า ในการทดสอบครั้งนี้จะมีโพรโทคอลปรากฏใน Even List 2 ชนิดคือ DNS และ TCP (Web Server) เนื่องจากเครื่อง PC จะต้องสอบถามชื่อผ่าน Domain Name Server ก่อนเสมอ เพื่อแปลง URL (www.firstlab.com) เป็นหมายเลข IP Address จากนั้น PC จึงใช้หมายเลขไอพีดังกล่าวเข้าใช้บริการเว็บเซิร์ฟเวอร์ต่อไป

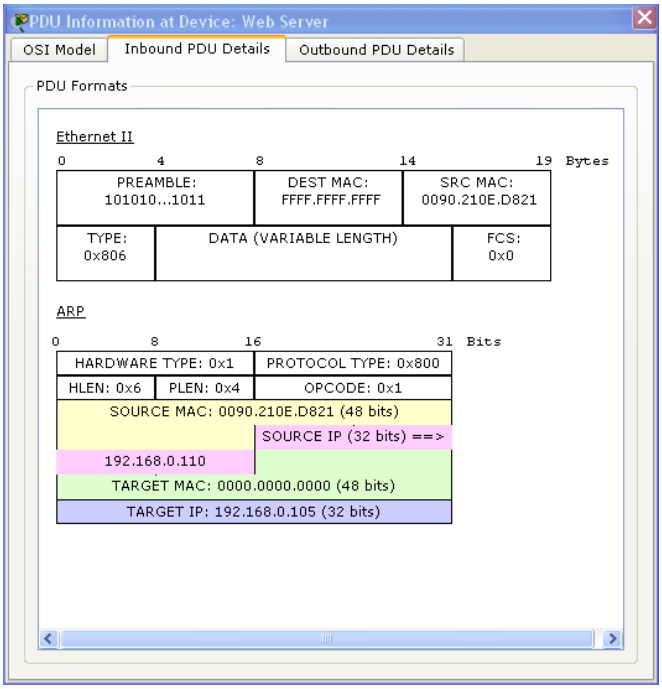
สำรวจเนื้อในของแพ็คเก็ต

ในหัวข้อนี้จะแสดงข้อมูลภายในแพ็คเก็ตว่ามีหน้าตาเป็นอย่างไร เริ่มต้นโดยคลิกเลือกที่ โหมด Simulation (อย่าลืมกดปุ่ม Reset Simulation เพื่อเคลียร์ค่าข้อมูลเดิมก่อน) ทดสอบ Add Simple PDU (ping) จากเครื่อง PC ไปยัง Server อีกครั้ง เมื่อโปรแกรมแสดงการส่งแพ็คเก็ตเป็นลักษณะซองจดหมาย  ให้คลิกที่ซองดังกล่าว สีของซองจดหมายในแต่ละซองแสดงถึงโพรโทคอลในแต่ละเลเยอร์ที่ทำงานอยู่ภายใต้ OSI Model ตัวอย่าง เช่น สีม่วงคือ โพรโทคอล ICMP ที่

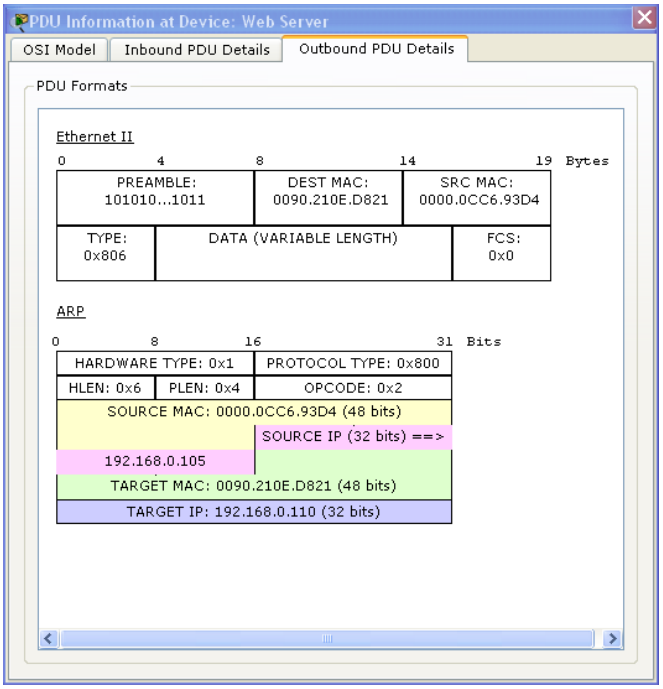
กำลังทำงานอยู่ในชั้นของ Network Layer หรือ สีเขียวหมายถึงโพรโทคอล ARP ที่ทำงานอยู่ในระดับเลเยอร์ data link (เลเยอร์ 2 ใน OSI Model แทบสีเหลืองแสดงถึงกำลังทำงานที่ชั้นดังกล่าว) เมื่อผู้ใช้ต้องการดูข้อมูลอย่างละเอียดให้คลิกเลือกที่แท็บ Inbound/Outbound PDU Details ดังรูปที่ 11.28, 11.29, 11.30



รูปที่ 11.28 โพรโทคอล ARP ทำงานที่เลเยอร์ที่ 2 (data link) ใน OSI Model

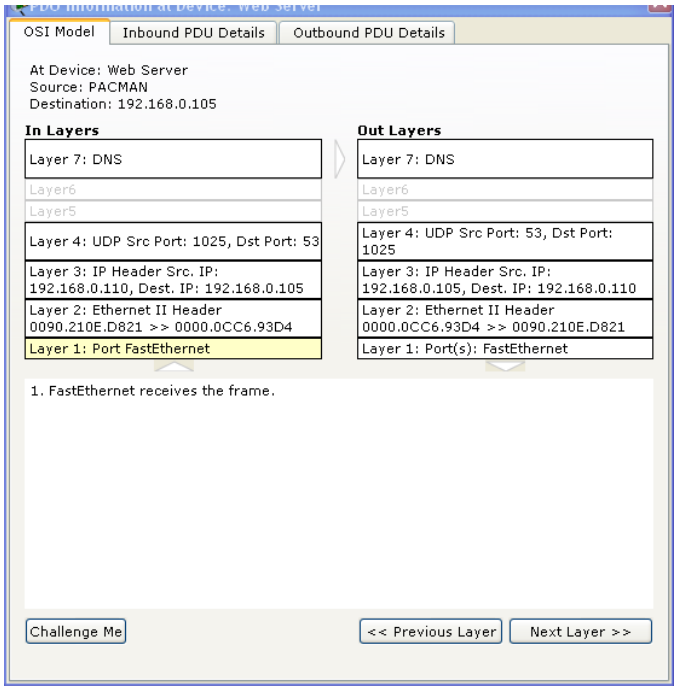


รูปที่ 11.29 แสดงข้อมูลอย่างละเอียดภายในโพรโทคอล ARP และ โพรโทคอล Ethernet ในทิศทางข้อมูลเข้า Inbound PDU Details



รูปที่ 11.30 แสดงข้อมูลอย่างละเอียดภายในโพรโทคอล ARP และ โพรโทคอล Ethernet ในทิศทางข้อมูลออก Outbound PDU Details

ทดลองอีกครั้งโดยการเลือกแท็บ Desktop ในเครื่อง PC ⇨ Web Browser ใส่ข้อมูลในช่อง URL เป็น www.firstlab.com แล้วคลิก Go จากนั้นให้กลับไปที่ workspace อีกครั้ง แพ็คเก็ตจะหยุดอยู่ให้ผู้ใช้เลือก Capture/Forward เพื่อผลักดันให้แพ็คเก็ตเคลื่อนที่ออกไปจากเครื่อง PC ให้สังเกตว่ามีซองสีน้ำตาลซึ่งเป็นแพ็คเก็ตของ DNS เกิดขึ้นมาก่อน อาศัยโพรโทคอล UDP (ชั้นที่ 4) พอร์ต 53 แพ็คเก็ต DNS นั้นทำงานถึงระดับที่ 7 ของ OSI Model ดังรูปที่ 11.31



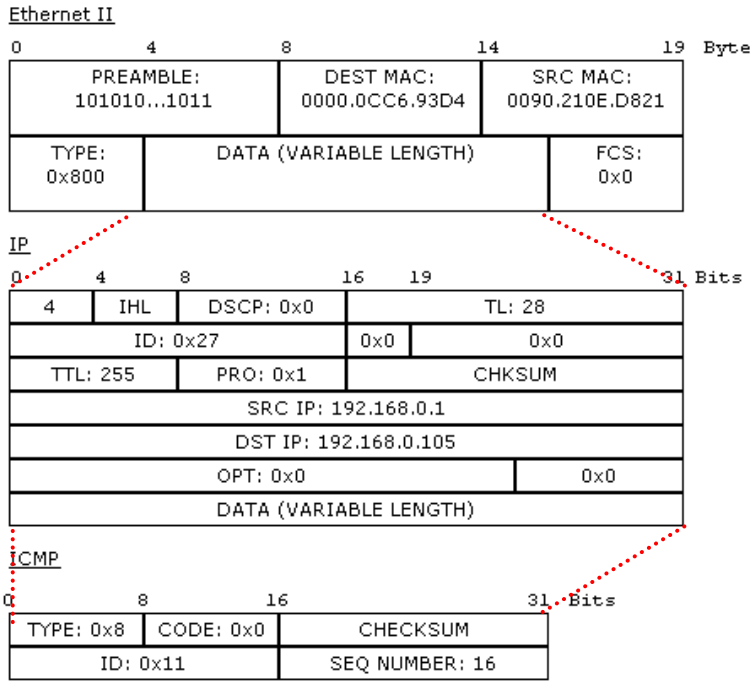
รูปที่ 11.31 แสดงข้อมูลโพรโทคอล DNS อาศัยโพรโทคอล UDP/IP ในการส่งข้อมูล

ในทำนองเดียวกันเมื่อกดปุ่ม Capture/Forward ไปเรื่อยๆ พร้อมๆ กับสังเกตุแพ็คเก็ตที่วิ่งไปวิ่งมาทุกๆ แพ็คเก็ตจะพบว่า แต่ละโพรโทคอลทำงานในชั้นของ OSI Model ที่ต่างกัน ดังนี้

ตารางที่ 11.7 ตัวอย่างโพรโทคอลที่ทำงานแตกต่างกันในแต่ละชั้นบน OSI Model

Protocol	Layer1	Layer2	Layer3	Layer4	Layer5	Layer6	Layer7
ICMP(ping)	Port	MAC	IP+ICMP	-	-	-	-
DNS	Port	MAC	IP	UDP(53)	-	-	DNS
HTTP	Port	MAC	IP	TCP(80)	-	-	HTTP
HTTPS	Port	MAC	IP	TCP(443)	-	-	HTTPS
DHCP	Port	MAC	IP	UDP(67,68)	-	-	DHCP

จากตารางที่ 11.7 สังเกตุว่าโพรโทคอล ICMP (ping) จะทำงานเพียงแค่ชั้นที่ 3 ใน OSI Model เท่านั้น โดยเริ่มต้นที่ เครื่อง PC ทำการสร้างแพ็คเก็ต ICMP จากนั้นส่งออกไปยัง port (ในที่นี้คือ FastEthernet ทำงานในเลเยอร์ที่ 1) ในระดับเลเยอร์ที่ 2 จะอาศัยโพรโทคอล Ethernet ในการผลักดันให้แพ็คเก็ตเคลื่อน ไปยัง Next Hop โดยอาศัย MAC Address ในการค้นหาที่อยู่ของโหนดหรือ Hop เพื่อนบ้าน โดยมีข้อมูลในเลเยอร์ที่ 3 เป็นผู้กำหนดเส้นทางที่จะไป (กำหนดโดยใช้หมายเลข IP Address) ในเลยอร์นี้จะมีโพรโทคอล IP เป็นผู้ผลักดันแพ็คเก็ตในการหาเส้นทาง และภายในแพ็คเก็ตของ IP ก็จะมี โพรโทคอล ICMP ซ้อนอยู่ข้างในอีกที เพื่อทำหน้าที่ตรวจสอบเครื่องปลายทางว่าอยู่หรือไม่ สังเกตเห็นว่าในแต่ละแพ็คเก็ตจะมีโพรโทคอลทำงานอยู่หลายตัว แต่ละตัวทำงานคนละหน้าที่กัน ไม่ก้าวก่ายกันเลย ดังรูปที่ 11.32

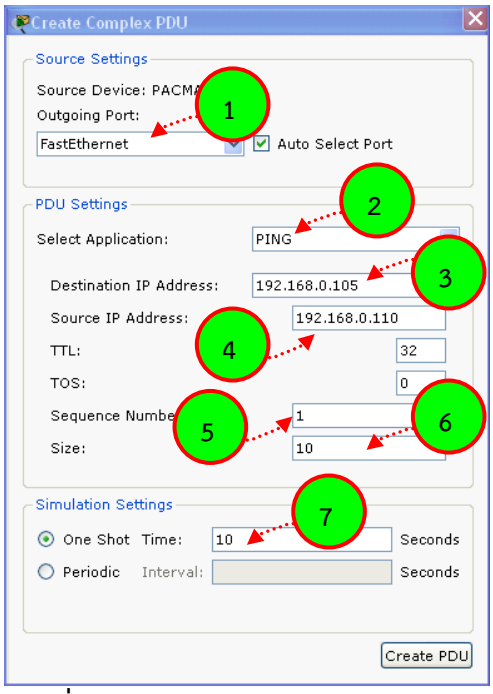


รูปที่ 11.32 แสดงความสัมพันธ์ของโพรโทคอลภายในแต่ละแพ็คเก็ตของ ICMP

การใช้งาน Add Complex PDU

ในหัวข้อก่อนๆ ได้ทำการทดสอบเครือข่ายโดยใช้ Add Simple PDU ซึ่งเป็นการทดสอบโดยใช้โพรโทคอล ICMP (ping) เท่านั้น เมื่อผู้ใช้ต้องการทดสอบเครือข่ายในขั้นที่สูงขึ้น ให้ใช้เครื่องมือที่ชื่อว่า Add Complex PDU ซึ่งเตรียมโพรโทรคอลให้ผู้ใช้ทดสอบเครือข่ายได้หลายแบบ เช่น ทดสอบด้วย DNS, Finger, FTP, HTTP, HTTPS, IMAP, POP, NETBIOS, SFTP, SMTP เป็นต้น ซึ่งมีขั้นตอนดังนี้

- เลือกโหมดได้ทั้งแบบ Realtime และ Simulation ⇨ เลือก Add Complex PDU ⇨ คลิกลงไปที่ตัวอุปกรณ์ที่ต้องการให้สร้างแพ็คเก็ตในการทดสอบ ⇨ ปรากฏ Dialog Create Complex PDU ⇨ ในช่อง Outgoing Port ให้เลือกอินเทอร์เฟสการ์ดที่ต้องการส่งข้อมูลออกไปยังเครือข่าย (เช่น เลือก FastEthernet) ⇨ ในแท็บ PDU Setting เป็นการกำหนดคุณสมบัติในการสร้างแพ็คเก็ต ในช่อง Selection Application ให้เลือกโพรโทคอลที่ต้องการสร้าง เช่น ICMP ⇨ คุณสมบัติของโพรโทคอลแต่ละตัวจะกำหนดไม่เหมือนกัน ในที่นี้ขอยกตัวอย่างเฉพาะ ICMP เท่านั้น ดังรูปที่ 11.33



รูปที่ 11.33 แสดงการกำหนดค่าใน PDU

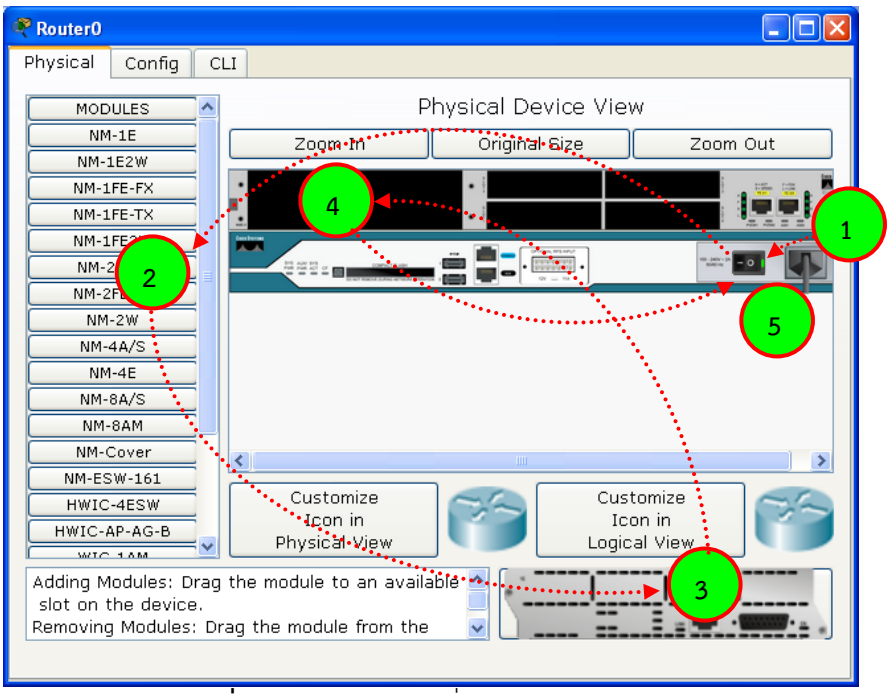
1. เลือกอินเทอร์เฟสที่ต้องการส่งแพ็คเก็ตออกสู่เครือข่าย (เลือกเป็น FastEthernet)
2. เลือก Application ที่ต้องการใช้ทดสอบ (PING)
3. กำหนดหมายเลข IP ปลายทางที่ต้องการทดสอบ (เช่น 192.168.0.105)

4. กำหนดหมายเลข IP ต้นทางที่ทดสอบ (เช่น 192.168.0.110)
5. กำหนดลำดับของแพ็คเก็ตที่ใช้ทดสอบ (เป็นจำนวนเต็ม)
6. กำหนดขนาดของแพ็คเก็ต (เป็นจำนวนเต็ม ค่า default=0)
7. กำหนดเวลาในการแสดงผลแต่ละครั้งเมื่อทำงานในโหมด Simulation

เมื่อกำหนดคุณสมบัติครบถ้วนแล้ว (ถ้ากำหนดไม่ครบจะมี Dialog เตือนว่ายังกำหนดไม่ครบ) ให้เลือก Create PDU แล้วคลิกที่อุปกรณ์ที่ใช้ทดสอบ เช่น PC จะปรากฏซองจดหมาย ต่อไปให้เลือก Capture/Forward เมื่อต้องการสังเกตุพฤติกรรมของแพ็คเก็ตทีละ step แต่ถ้าต้องการทดสอบอย่างต่อเนื่องให้ใช้ Auto Capture/Play แล้วทำการทดสอบเหมือนที่อธิบายไว้แล้ว

5. Devices and Modules

โดยปกติอุปกรณ์เครือข่ายทุกๆ ตัวสามารถเพิ่ม/ลด อุปกรณ์ใหม่ได้ เช่น การ์ดเน็ตเวิร์ดชนิดต่างๆ จอภาพ ไวเลสแลน เป็นต้น สำหรับใน packet tracer ก็เช่นเดียวกัน ผู้ใช้สามารถเพิ่ม/ลด อุปกรณ์ต่างๆ ได้เช่นเดียวกัน มีขั้นตอนดังนี้ เลือกแท็บ Physical ⇨ ปิดสวิตช์เครื่อง ⇨ คลิกแล้วลากอุปกรณ์เดิมออกจากเครื่อง (ในกรณีที่ไม่มีพอร์ต หรือ slot ว่างเหลืออยู่) หรือ เลือก Module ทางด้านขวามือที่เหมาะสมมาวางไว้ในพอร์ตหรือ slot ที่ว่าง (ในกรณีที่มีพอร์ต หรือ slot ว่างเหลืออยู่) ⇨ เปิดเครื่อง ดังรูปที่ 11.34



รูปที่ 11.34 แสดงการเพิ่ม/ลด Module

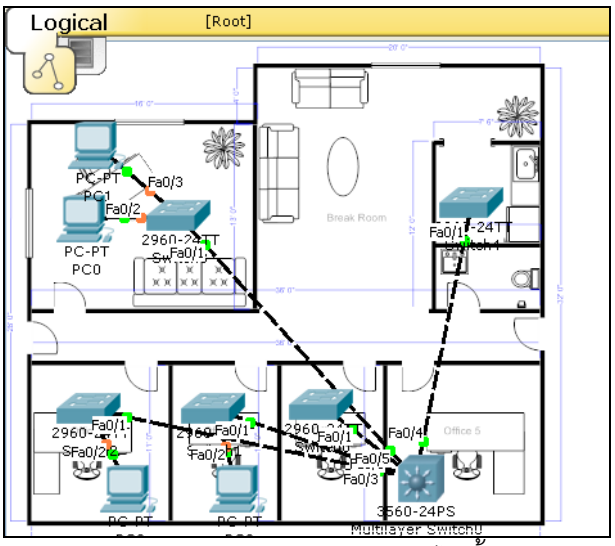
1. เลือกแท็บ Physical แล้วปิดเครื่อง
2. เลือก Module ที่ต้องการในแท็บ Modules

3. ลากอุปกรณ์ที่ปรากฏทางด้านล่างขวาไปใส่ใน slot ที่ว่าง (กรณีเอา Module ออกให้เลือก Module ที่ต้องการเอาออกลากมาวางที่ด้านล่างขวา)
4. วางอุปกรณ์ลงยัง slot ที่ว่างและเข้ากันได้พอดี
5. เปิดเครื่องอีกครั้ง

Feature ของ packet tracer ยังมีอีกมากมาย ซึ่งสามารถอ่านเพิ่มเติมได้จาก Contents หรือ Tutorials ในบทนี้แนะนำเฉพาะคุณสมบัติบางอย่างที่ผู้เขียนเห็นว่าควรจะรู้ในเบื้องต้นก่อน สำหรับคุณสมบัติอื่นๆ จะค่อยๆ เพิ่มเติมทีละเล็กละน้อยใน บทที่ 3 How to Network Connectivity ต่อไป

แบบฝึกหัดท้ายบท

1. ให้นิสิตทำการเชื่อมต่อเครือข่ายด้วยซอฟต์แวร์ Packet Tracer ดัง diagram ต่อไปนี้



2. ให้นิสิตทำการตรวจสอบข้อมูลภายในของโพรโทคอลดังต่อไปนี้
   a. ICMP(ping)
   b. DNS
   c. HTTP
   d. HTTPS
   e. DHCP

บทที่ 12
เทคนิคการเชื่อมต่อเครือข่าย (Networking Connectivity Techniques)

แนวคิด

ในบทนี้จะมาเรียนรู้วิธีการเชื่อมต่อระบบเครือข่ายแบบต่างๆ ผ่านโปรแกรมจำลองเครือข่าย สำหรับรูปแบบการเชื่อมต่อจะกำหนดตามสถานการณ์ โดยเริ่มจากการเชื่อมต่อที่ง่ายๆ ไปจนถึงเครือข่ายที่ซับซ้อน และพร้อมกับทำการทดสอบและวิเคราะห์ระบบไปพร้อมๆ กัน

วัตถุประสงค์

1. สามารถออกแบบและสร้างระบบเครือข่ายตามที่กำหนดได้
2. สามารถปรับแต่งและแก้ไขระบบเครือข่ายให้สามารถทำงานได้อย่างถูกต้องได้
3. วิเคราะห์ปัญหาและหาแนวทางในการแก้ไขเมื่อระบบเครือข่ายเกิดปัญหาได้อย่างถูกต้อง และเหมาะสม
4. สามารถบูรณาการความรู้เกี่ยวกับระบบเครือข่ายในงานจริงได้

ในบทนี้จะอธิบายถึงวิธีการการออกแบบและติดตั้งเครือข่ายโดยใช้โปรแกรม Packet Tracer เวอร์ชัน 5.3.1 ในลักษณะแบบ step by step โดยแบ่งออกเป็น Scenario ย่อย ๆ เพื่อเป็นพื้นฐานในการสร้างระบบเครือข่ายขนาดใหญ่ๆ ในบทต่อไป ผู้อ่านสามารถเลือก Scenario ที่สนใจได้โดยไม่จำเป็นต้องเริ่มตั้งแต่ Scenario 1 (ในกรณีที่มีพื้นฐานด้านเครือข่ายอยู่แล้ว) แต่ถ้าเป็นผู้ที่เริ่มต้นเรียนเกี่ยวกับคอมพิวเตอร์เครือข่าย ผู้เขียนแนะนำว่าควรเริ่มอ่านและลงมือปฏิบัติด้วยตนเองโดยเริ่มตั้งแต่ Scenario ที่ 1 ไปเรื่อยๆ ตามลำดับ

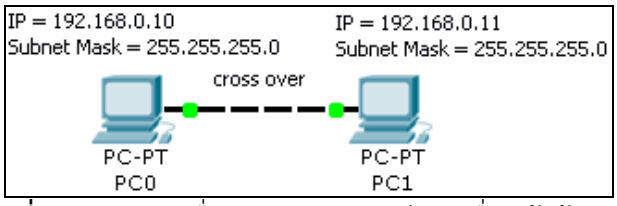
ปฏิบัติการด้านการบริหารจัดการเครือข่ายที่หลากหลาย (เช่น การติดตั้งเครื่องแม่ข่ายเว็บ ระบบพร็อกซี่ ดีเอชซีพี เครื่องแม่ข่ายบริการไฟล์และเครื่องแม่ข่ายบริการดีเอ็นเอส เป็นต้น)

Scenario 1: เชื่อมต่อคอมพิวเตอร์ PC กับ PC

คำอธิบาย :

สำหรับการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง เช่น PC กับ PC, PC กับ Notebook หรือ PC กับ Server ไม่จำเป็นต้องใช้อุปกรณ์เครือข่ายเพิ่มเติมเช่น สวิตช์ หรือ ฮับ อุปกรณ์ที่จำเป็นต้องใช้คือ การ์ดเน็ตเวิร์คและสายไขว้ (Cross Over) ก็เพียงพอต่อการเชื่อมต่อ ดังรูปที่ 12.1

แผนผังการเชื่อมต่อ :



รูปที่ 12.1 ผังการเชื่อมต่อคอมพิวเตอร์ 2 เครื่องเข้าด้วยกัน

รายการอุปกรณ์ :

1. เครื่อง PC 2 เครื่อง พร้อมการ์ดเน็ตเวิร์คชนิด FastEthernet
2. สายนำสัญญาณชนิดไขว้ (Cross Over)

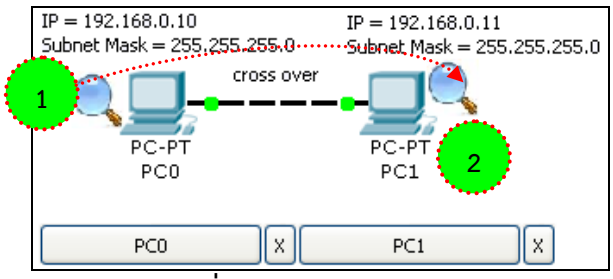
ขั้นตอนการเชื่อมต่อ :

1. เลือก End Devices (ในส่วน Device-Type)
2. เลือก Generic (ในส่วน Device-Specific) แล้วลากมาวางใน workspace ให้ครบ 2 เครื่อง (โดยปกติจะมีชื่อเป็น PC0, PC1, PCn ตามลำดับ)
3. เลือก Connections (ในส่วน Device-Type)
4. เลือก Automatically Choose Connection Type (ในส่วน Device-Specific) แล้วคลิกที่เครื่อง PC0 แล้วลากไปคลิกที่ PC1 โปรแกรมจะเลือกสายชนิด Cross Over เชื่อมต่อให้อัตโนมัติ ที่เครื่องคอมพิวเตอร์จะปรากฏไฟสีเขียว แสดงว่าเชื่อมต่อสำเร็จ

5. ทดสอบการเชื่อมต่อโดยการเลือก Inspect (ในส่วน Common Tools Bar ด้านขวามือ) คลิกเลือกที่ PC0 แล้วเลือก Port Status Summary Table ให้สังเกตุที่ Link จะมีสถานะเป็น up และ MAC Address จะปรากฏหมายเลขเป็นฐาน 16 เช่น 0060.5C25.1EC1 (มีขนาด 48 บิต โดยตัวอักษร 1 ตัวแทนข้อมูล 4 บิต)
6. คลิกที่ PC0 เลือกแท็บ Desktop ⇨ IP Configuration ⇨ Static ⇨ กำหนดค่าในช่อง IP Address เป็น 192.168.0.10 และ Subnet Mask เป็น 255.255.255.0
7. คลิกที่ PC1 เลือกแท็บ Desktop ⇨ IP Configuration ⇨ Static ⇨ กำหนดค่าในช่อง IP Address เป็น 192.168.0.11 และ Subnet Mask เป็น 255.255.255.0
8. เสร็จสิ้นการเชื่อมต่อ

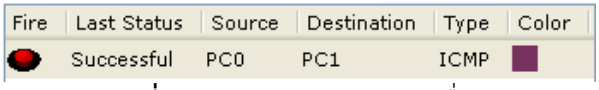
การทดสอบ :

1. เลือก Add Simple PDU ในส่วน Common Tools Bar ด้านขวามือ
2. คลิกบนเครื่อง PC0 1 ครั้ง และคลิกที่ PC1 อีก 1 ครั้ง ดังรูปที่ 12.2



รูปที่ 12.2 การใช้งาน PDU

3. สังเกต ในส่วน User Created Packet Window (ด้านล่างขวามือของโปรแกรม) ถ้า Last Status เป็น Successful แสดงว่าเชื่อมต่อสมบูรณ์แล้ว ดังรูปที่ 12.3



Fire	Last Status	Source	Destination	Type	Color
	Successful	PC0	PC1	ICMP	

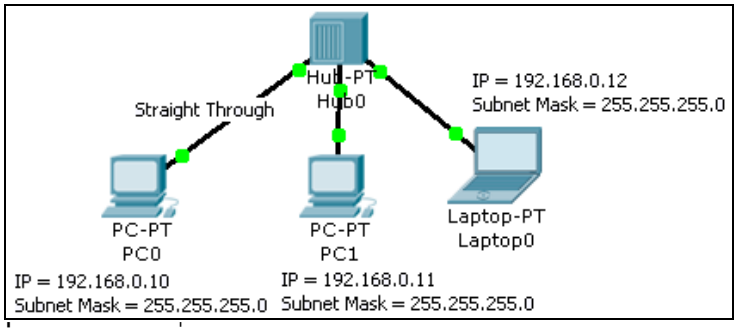
รูปที่ 12.3 การตรวจสอบการเชื่อม

Scenario 2: **เชื่อต่อคอมพิวเตอร์ PC0, PC1 และ Labtop0 กับ HUB**

คำอธิบาย :

การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์มากกว่า 2 เครื่องขึ้นไป จำเป็นต้องอาศัยอุปกรณ์เครือข่ายเพิ่มเติมเช่น สวิตช์ หรือ ฮับ เพื่อรวมอุปกรณ์เข้าด้วยกันเป็นเครือข่าย และใช้สายนำสัญญาณประเภทสายตรง (Straight Through) สำหรับเชื่อมอุปกรณ์เข้าด้วยกัน ดังรูปที่ 12.4

แผนผังการเชื่อมต่อ :



รูปที่ 12.4 ผังการเชื่อมต่อคอมพิวเตอร์ PC0, PC1 และ Labtop0 กับ HUB

รายการอุปกรณ์ :

1. เครื่อง PC 2 เครื่อง พร้อมการ์ดเน็ตเวิร์คชนิด FastEthernet
2. เครื่อง Laptop 1 เครื่อง พร้อมการ์ดเน็ตเวิร์คชนิด FastEthernet
3. สายนำสัญญาณชนิดตรง (Straight Through)

ขั้นตอนการเชื่อมต่อ :

1. เลือก End Devices (ในส่วน Device-Type)
2. เลือก Generic (ในส่วน Device-Specific) แล้วลากมาวางใน workspace ให้ครบ 2 เครื่อง (โดยปกติจะมีชื่อเป็น PC0, PC1 ตามลำดับ)
3. เลือก Labtop-PT (ในส่วน Device-Specific) แล้วลากมาวางใน workspace
4. เลือก Connections (ในส่วน Device-Type)
5. เลือก Copper Straight-Through (ในส่วน Device-Specific) แล้วคลิกที่เครื่อง PC0 ไปยัง HUB และ PC1, Labtop0 ไปยัง HUB ตามลำดับ ที่เครื่องคอมพิวเตอร์จะปรากฏไฟสีเขียว แสดงว่าเชื่อมต่อสำเร็จ
6. คลิกที่ PC0, PC1 และ Laptop0 ทีละเครื่อง ในแต่ละเครื่องเลือกแท็บ Desktop ⇨ IP Configuration ⇨ Static ⇨ กำหนดค่าหมายเลข IP Address และ Subnet Mask ดังตารางด้านล่าง

   ตารางที่ 12.1 ไอพีแอดเดรส และ subnet mask

เครื่อง	IP Address	Subnet Mask
PC0	192.168.0.10	255.255.255.0
PC1	192.168.0.11	255.255.255.0
Labtop0	192.168.0.12	255.255.255.0

7. เสร็จสิ้นการเชื่อมต่อ

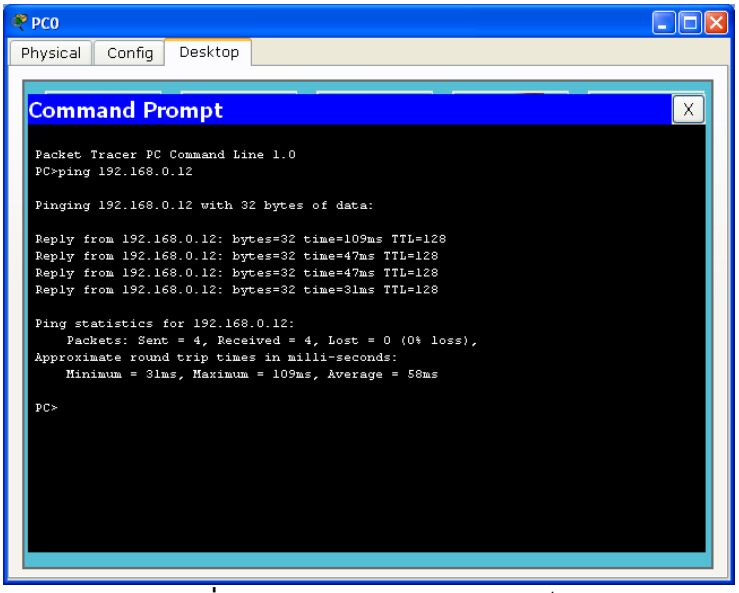
การทดสอบ :

1. ที่เครื่อง PC0 เลือก Desktop ⇨ Command Prompt

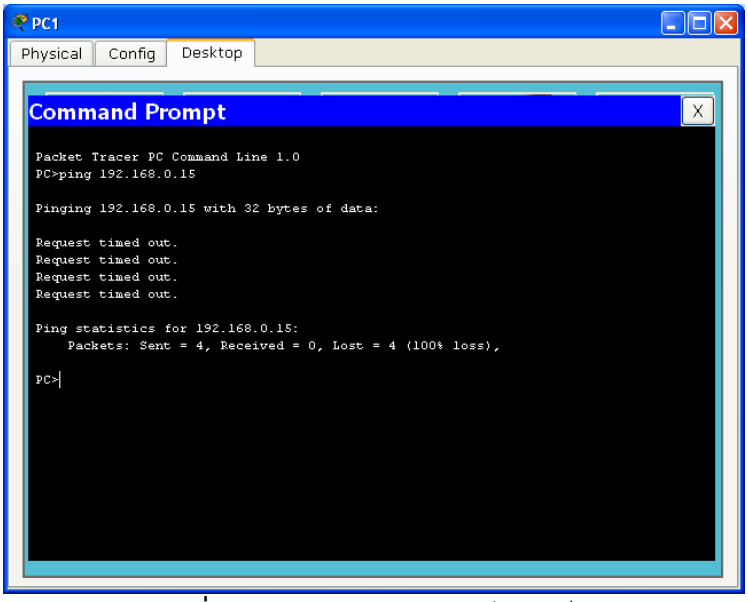
2. เมื่อปรากฏหน้าต่าง Command Prompt (หน้าต่างเป็นสีดำ) ให้ผู้ใช้ออกคำสั่งทดสอบ คือ ping ตามด้วยหมายเลข IP Address ที่ต้องการทดสอบ ในที่นี้ให้ใช้คำสั่งคือ

PC>ping 192.168.0.10	//ทดสอบเครื่องตัวเอง (PC0)
PC>ping 192.168.0.11	//ทดสอบเครื่อง PC1
PC>ping 192.168.0.12	//ทดสอบเครื่อง Labtop0

3. จากตัวอย่างการทดสอบข้างบน ถ้าเครื่องที่ถูก ping มีสถานะเป็นปกติ คือทำงานอยู่จะตอบกลับด้วย ICMP Reply (เช่น Reply from 192.168.0.12: bytes=32 time=109ms TTL=128) แต่ถ้า ping แล้วไม่มีเครื่องปลายทางที่ทำงานอยู่จริงจะแสดง Message คือ Request timed out ดังรูปที่ 12.5, 12.6

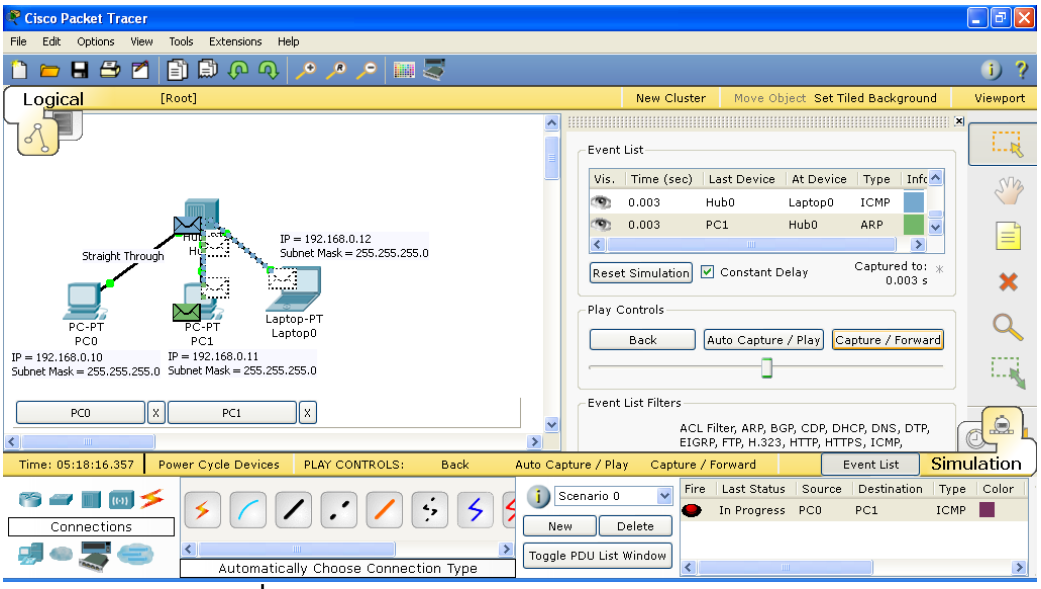


รูปที่ 12.5 กรณีทดสอบ ping สำเร็จ



รูปที่ 12.6 กรณีทดสอบ ping ไม่สำเร็จ

4. ลองทดสอบด้วย ping อีกครั้ง ใน โหมด Simulation สังเกตพฤติกรรมการทำงานใน Even List ดังรูปที่ 12.7



รูปที่ 12.7 การทดสอบด้วย ping ในโหมด Simulation

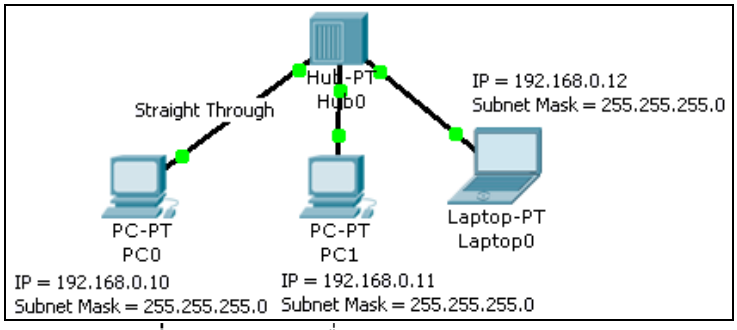
หมายเหตุ : MAC Address คือหมายเลขของการ์ดเน็ตเวิร์คที่ไม่ซ้ำกัน มีขนาด 48 บิต ใช้สำหรับติดต่อกันระหว่างอุปกรณ์ในระดับเลเยอร์ที่ 2 ของ OSI Model ซึ่งคุณสมบัติของอุปกรณ์ที่ทำงานในเลเยอร์ 2 จะติดต่อกับแบบ Hop ต่อ Hop เท่านั้น แต่เมื่อต้องการส่งข้อมูลให้ไกลออกไปจะต้องอาศัยคุณสมบัติของเลเยอร์ 3 คือ IP Address แทน

การแก้ไขปัญหาในเครือข่าย

Scenario 3: การวิเคราะห์แพ็คเก็ตอย่างละเอียด ตอนที่ 1

คำอธิบาย :

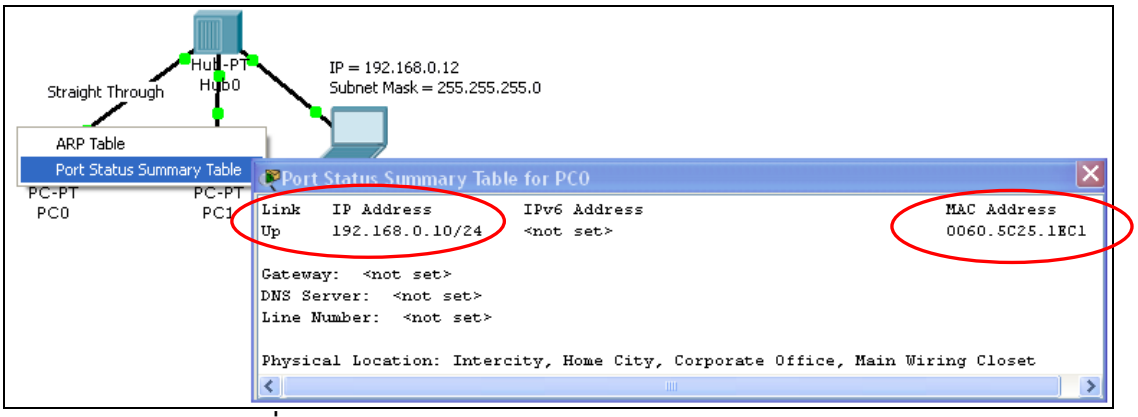
การวิเคราะห์แพ็คเก็ตอย่างละเอียดจะทำให้เราสามารถเข้าใจและค้นหาปัญหาที่เกิดขึ้นบนระบบเครือข่ายได้อย่างมีประสิทธิภาพ ดังนั้นในหัวข้อนี้ผู้เขียนจะแนะนำ เทคนิคการเฝ้ามองแพ็คเก็ตอย่างเป็นระบบ เพื่อให้ผู้อ่านเข้าใจกระบวนการทำงานของเครือข่ายอย่างเป็นรูปธรรม จากตัวอย่างเครือข่ายใน Scenario 2 ให้ปฏิบัติตามขั้นตอนดังนี้ แผนผังการเชื่อมต่อ : ดังรูปที่ 12.8



รูปที่ 12.8 ผังการเชื่อมต่อสำหรับ scenario 2

ขั้นตอนการวิเคราะห์ :

1. ในเบื้องต้นให้ทำการตรวจสอบ Port Status ก่อนว่ามีสถานะเป็น up หรือไม่ กับทุกๆ เครื่องที่เชื่อมอยู่บนเครือข่ายโดยใช้ Inspect ดังรูปที่ 12.9 เมื่อแน่ใจว่าทุกๆ Port มีสถานะเป็น up แล้ว ให้ทำขั้นตอนต่อไป



รูปที่ 12.9 การตรวจสอบสถานการทำงานของ Port Status

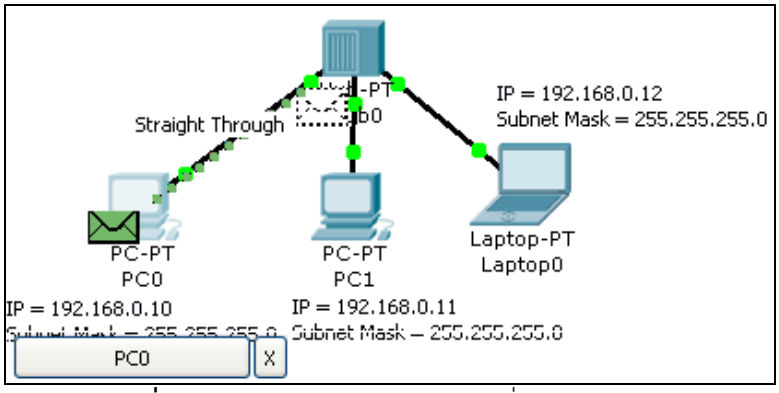
2. ตรวจสอบตาราง ARP ในเครื่องคอมพิวเตอร์ทุกๆ เครื่อง (ARP เป็นโพรโทคอลที่ทำหน้าที่สอบถามว่า หมายเลข IP Address ที่ต้องการติดต่อมีหมายเลข MAC ใด และ Reverse ARP เป็นการถามกลับว่าหมายเลข MAC ที่ต้องการติดต่อตรงกับหมายเลข IP Address ใด) ซึ่งในเบื้องต้นเมื่อยังไม่มีการแลกเปลี่ยนข้อมูลกันระหว่างเครื่องผู้ใช้งานจะยังไม่มีข้อมูลใดๆ ในตาราง ARP
3. ให้เลือกทำงานในโหมด Simulation เพื่อสังเกตุพฤติกรรมการทำงานของแพ็คเก็ต
4. ที่เครื่อง PC0 คลิกเลือก Desktop ⇨ Command Prompt
5. ใช้คำสั่ง ping ไปยัง IP Address ที่ต้องการทดสอบ ในที่นี้ให้ ping 192.168.0.12 (เครื่อง Labtop0)

```
PC>ping 192.168.0.12
```

6. ในโหมด Simulation แพ็คเก็ตจะหยุดรอให้ผู้ใช้สั่งงานเป็น step ได้ โดยการเลือก Capture/Forward
7. คลิก Capture/Forward 1 ครั้ง แพ็คเก็ต ICMP จะวิ่งไปยัง HUB (สังเกตใน Even List จะเห็นว่ามีแพ็คเก็ต ICMP วิ่งจากเครื่อง PC0 ที่เวลา 0.000 วินาที) รูปที่ 12.10, 12.11

Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	

รูปที่ 12.10 ข้อมูลที่เริ่มต้นส่งจาก PC0 คือ ICMP แพ็คเก็ต



รูปที่ 12.11 ทดสอบโดยการ ping เริ่มต้นจาก PC0

8. เนื่องจากคำสั่ง ping นั้นใช้หมายเลข IP Address ในการทดสอบ ณ สถานการณ์ปัจจุบันเครื่อง PC0 ไม่ทราบว่าเครื่องเป้าหมาย (192.168.0.12) คือใคร เพราะในตาราง ARP Table ยังไม่มีข้อมูลใดๆ เลย ดังนั้นเครื่อง PC0 จึงส่งแพ็คเก็ต ARP กระจายออกไปยังทุกๆ พอร์ตยกเว้นตัวมันเอง (พอร์ต PC0) รูปที่ 12.12

Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	
	0.000	--	PC0	ARP	

รูปที่ 12.12 เวลาที่ 0.000 PC0 ส่งแพ็คเก็ต ARP เพื่อถามว่าใครคือ IP เป้าหมาย (192.168.0.12)

9. แพ็คเก็ต ARP จะส่งต่อไปยัง HUB (เวลา 0.001 ใน Even List)

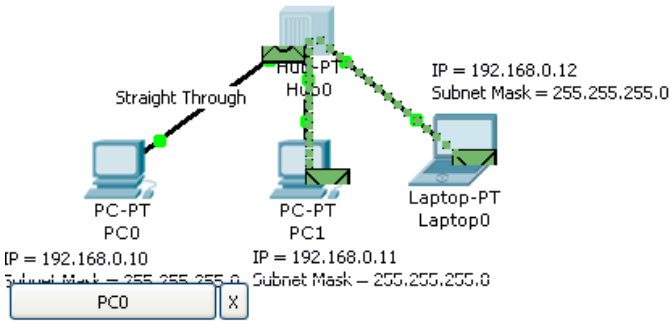
Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ARP	
	0.001	PC0	Hub0	ARP	

รูปที่ 12.13 แพ็คเก็ต ARP

10. HUB จะส่งแพ็คเก็ต ARP ต่อไปยัง PC1 และ Labtop0 พร้อมกัน (เวลา 0.002 ใน Even List) ดังรูปที่ 12.24

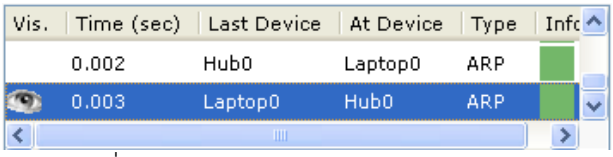
Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.002	Hub0	PC1	ARP	
	0.002	Hub0	Laptop0	ARP	

รูปที่ 12.14 เวลา 0.002 HUB ส่ง ARP ไปยัง PC1 และ Labtop0 พร้อมกัน



รูปที่ 12.15 การเดินทางของแพ็คเก็ต ARP

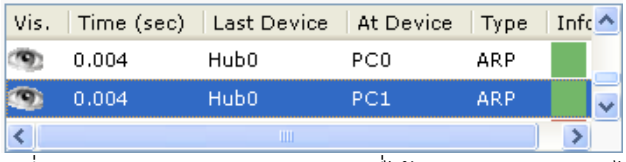
11. Labtop0 จะตอบกลับ ARP reply กลับมา เนื่องจากเป็นเครื่องที่มี IP Address เท่ากับ 192.168.0.12 แต่ เครื่อง PC1 จะไม่ตอบกลับเพราะไม่ใช่ IP ของตนเอง รูปที่ 12.16



Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.002	Hub0	Laptop0	ARP	
	0.003	Laptop0	Hub0	ARP	

รูปที่ 12.16 เวลาที่ 0.003 Labtop0 ส่ง ARP Reply กลับไปให้กับ HUB

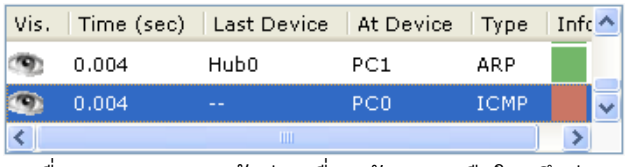
12. HUB จะกระจายแพ็คเก็ตที่ส่งมาจาก Laptop0 ไปยังทุกๆ เครื่องเนื่องจากคุณสมบัติของ HUB จะกระจายข้อมูลไปยังทุกๆ พอร์ตเสมอ ดังรูปที่ 12.17



Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.004	Hub0	PC0	ARP	
	0.004	Hub0	PC1	ARP	

รูปที่ 12.17 เวลาที่ 0.004 HUB ส่ง ARP Reply ที่ได้รับจาก Labtop0 ไปยังทุกๆ พอร์ต

13. ในเวลาที่ 0.004 เครื่อง PC0 ก็จะทราบแล้วว่า IP 192.168.0.12 คือใครจึงทำการส่ง ICMP ออกไปยังเครื่องเป้าหมายทันที ดังรูปที่ 12.18



Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.004	Hub0	PC1	ARP	
	0.004	--	PC0	ICMP	

รูปที่ 12.18 เครื่อง PC0 ทราบแล้วว่าเครื่องเป้าหมายคือใครจึงส่ง ICMP ออกไป

เมื่อถึงขั้นตอนนี้ ARP Table ของเครื่อง PC0 และ Labtop0 ก็จะถูก Update ดังนี้

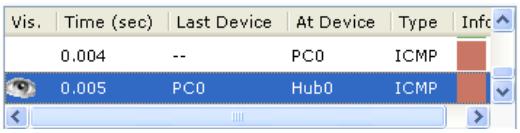
IP Address	Hardware Address (MAC)	Interface
192.168.0.10	0060.5C25.1EC1	FastEthernet

ตาราง ARP Table ของ PC1

IP Address	Hardware Address (MAC)	Interface
192.168.0.10	0060.5C25.1EC1	FastEthernet

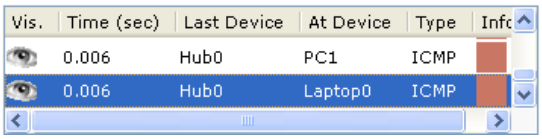
ตาราง ARP Table ของ Labtop0

14. เวลาที่ 0.005 เครื่อง PC0 ส่ง ICMP อีกครั้งไปยัง HUB



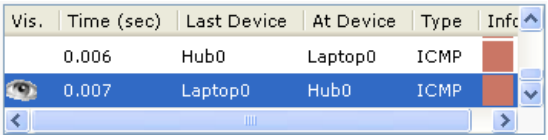
Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.004	--	PC0	ICMP	
	0.005	PC0	Hub0	ICMP	

15. เวลาที่ 0.006 HUB จะกระจายแพ็คเก็ต ICMP ไปยังทุกๆ พอร์ต (คุณสมบัติของ HUB)



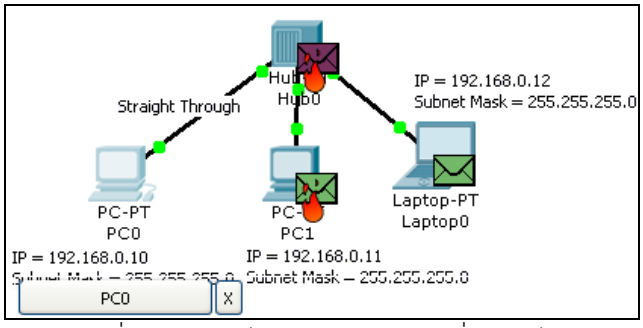
Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.006	Hub0	PC1	ICMP	
	0.006	Hub0	Laptop0	ICMP	

16. เวลาที่ 0.007 Labtop0 ส่งแพ็คเก็ต ICMP reply กลับไปยัง HUB



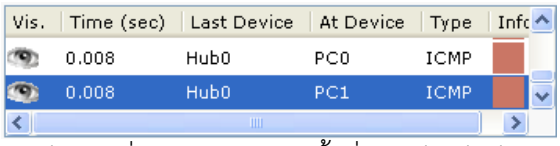
Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.006	Hub0	Laptop0	ICMP	
	0.007	Laptop0	Hub0	ICMP	

Laptop0 Reply ข้อมูลกลับ



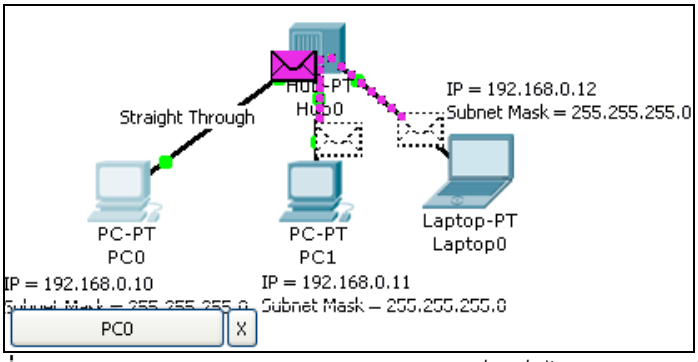
รูปที่ 12.19 เครื่อง PC1 จะไม่ส่ง ICMP กลับเนื่องจากไม่ใช่เป้าหมาย

17. เวลาที่ 0.008 HUB จะกระจายแพ็คเก็ต ICMP ไปยังทุกๆ พอร์ต (คุณสมบัติของ HUB)



Vis.	Time (sec)	Last Device	At Device	Type	Info
	0.008	Hub0	PC0	ICMP	
	0.008	Hub0	PC1	ICMP	

18. การเชื่อมต่อก็จะสำเร็จลง เมื่อทำการ ping ครั้งที่ 2 แพ็คเก็ตก็ยังคงเดินทางไปหาทุกเครื่องเหมือนเดิม (ตอบกลับเฉพาะเครื่องที่เป็นเป้าหมายเท่านั้น) เพราะคุณสมบัติของ HUB นั้นจะส่งไปยังทุกๆ พอร์ต แต่จะไม่เกิดกระบวนการ ARP ครั้งที่ 2 จนกว่า จะถึงเวลาที่ ARP Cache expire ซึ่งจะใช้เวลาประมาณ 10 นาที (600 วินาที)



รูปที่ 12.20 คุณสมบัติของ HUB จะกระจายแพ็คเก็ตไปยังทุกๆ เครื่อง

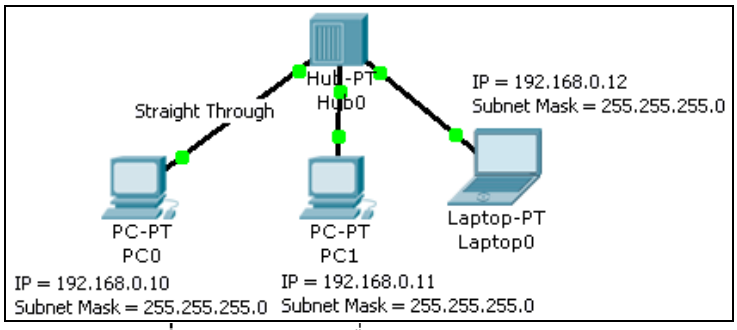
หมายเหตุ : ให้ทดลอง ping จากเครื่อง PC0 ไปยัง PC1 อีกครั้งแล้วสังเกตุพฤติกรรมการเปลี่ยนแปลง

Scenario 4: การวิเคราะห์แพ็คเก็ตอย่างละเอียด ตอนที่ 2

คำอธิบาย :

การวิเคราะห์แพ็คเก็ตใน Scenario 3 ได้ทำการวิเคราะห์ทิศทางการส่งข้อมูล สำหรับใน Scenario 4 นี้ จะพิจารณาแพ็คเก็ตในระดับที่ลึกซึ้งลงไปถึงระดับบิตข้อมูล จากตัวอย่างเครือข่ายใน Scenario 2 ให้ปฏิบัติตามขั้นตอนดังนี้

แผนผังการเชื่อมต่อ :



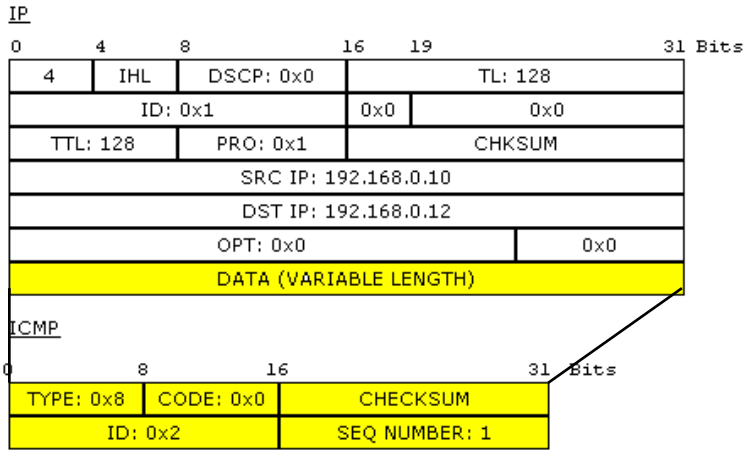
รูปที่ 12.21 ผังการเชื่อมต่อของ scenario 4

ขั้นตอนการวิเคราะห์ :

1. เริ่มต้นที่เครื่อง PC0 โดยการใช้คำสั่ง ping จาก Command Prompt ไปยังเครื่อง Labtop0 อีกครั้ง

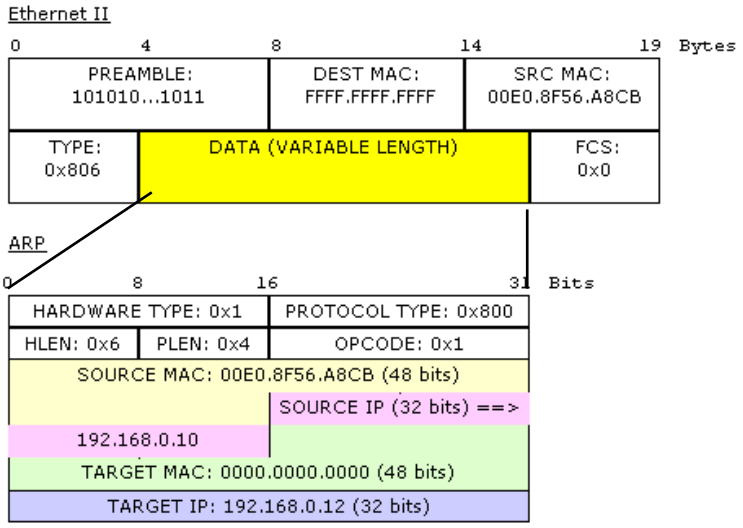
```
PC>ping 192.168.0.12
```

2. ในโหมด Simulation เมื่อออกคำสั่ง ping แล้ว ให้ใช้ Inspect ตรวจสอบแพ็คเก็ต ที่มีรูปเป็นซองจดหมาย (สีเขียวคือ ARP ■ และสีเทาคือ ICMP ■) โดยการคลิกที่ซองจดหมายที่ต้องการ ดังรูปที่ 12.32



รูปที่ 12.22 ตัวอย่างแพ็คเก็ตของ ICMP ที่ซ่อนอยู่ใน IP

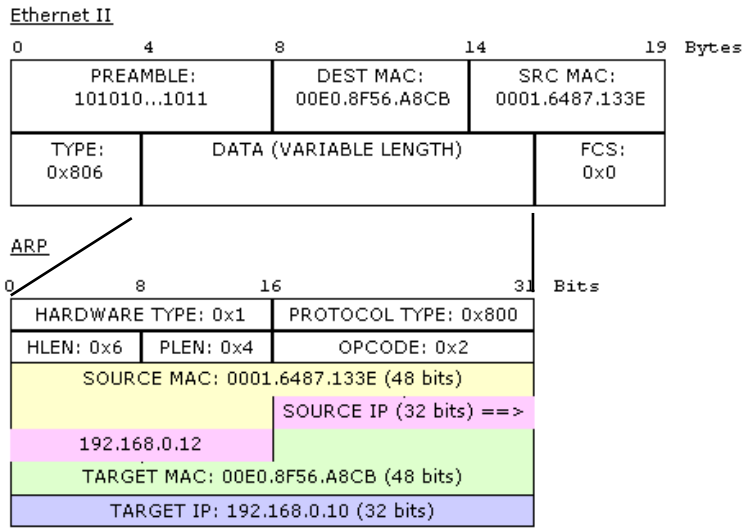
จากรูปข้างบน การสื่อสารข้อมูลในเน็ตเวิร์คจะแบ่งออกเป็นชั้นๆ ตามหลักการของ OSI Model (สามารถอ่านเพิ่มเติมได้ใน เรียนรู้เครือข่ายและอุปกรณ์ Cisco ด้วยโปรแกรม Simulation: ผู้เขียน สุชาติ คุ้มมะณี) จากในตัวอย่าง แพ็คเก็ตของ ICMP นั้นจะถูกซ่อนอยู่ใน IP (encapsulation) เพื่อให้แพ็คเก็ต IP นั้นเป็นผู้ส่งแพ็คเก็ต ICMP ไปให้ถึงปลายทาง ข้อมูลของ ICMP จะเป็นข้อมูลในส่วนของ DATA ใน IP แพ็คเก็ต โดย ICMP แพ็คเก็ตมีขนาดเท่ากับ 32 บิต x 2 คือ 64 บิต ประกอบไปด้วย Type มีขนาด 8 บิต เอาไว้บอกว่าเป็นโพรโทคอล ICMP ชนิด Echo Request, CODE มีค่าเท่ากับ 0, CHECKSUM เป็นค่าที่ใช้สำหรับตรวจสอบความผิดพลาดของข้อมูล, ID มีค่าเป็น 2, SEQ NUMBER คือลำดับของแพ็คเก็ต ซึ่งจะเปลี่ยนไปเรื่อยๆ ในที่นี้คือ 1 ดังรูปที่ 12.33



รูปที่ 12.23 ตัวอย่างแพ็คเก็ต ARP ที่ซ่อนอยู่ในแพ็คเก็ต Ethernet II

จากรูปข้างบน แสดงข้อมูลของแพ็คเก็ต ARP ที่อาศัยโพรโทคอล Ethernet (ทำงานในเลเยอร์ที่ 2) ส่งไปยังปลายทาง ข้อมูลที่อยู่ใน DATA ของ Ethernet frame จะเป็นแพ็คเก็ตของ ARP มีข้อมูลคือ HARDWARE TYPE=1, PROTOCOL TYPE=0x800, HLEN= ความยาวของ Header, PLEN=ความยาวของเนื้อข้อมูล, OPCODE=0x1, SOURCE MAX=48 บิต, SOURCE IP=32 บิต (192.168.0.10), TARGET MAC=48 บิต (เริ่มต้นจะต้องทำการกระจายข้อมูลไปทุกๆ เครื่อง โดยใช้ MAC=000.000.000), TARGET IP=32 บิต (192.168.0.12) สังเกตุว่าใน Ethernet frame จะ DEST MAC= FFF.FFF.FFF แสดงว่าเป็นการ broadcast ข้อมูลไปทุกๆ เครื่อง

3. เมื่อเครื่องปลายทางได้รับแพ็คเก็ตแล้วจะส่ง ARP Reply กลับไปยังเครื่องที่ส่งข้อมูลมา โดยการ update ค่า SOURCE MAC, SOURCE IP, TARGET MAC, TARGET IP ใน ARP frame คือ



รูปที่ 12.24 ARP Reply

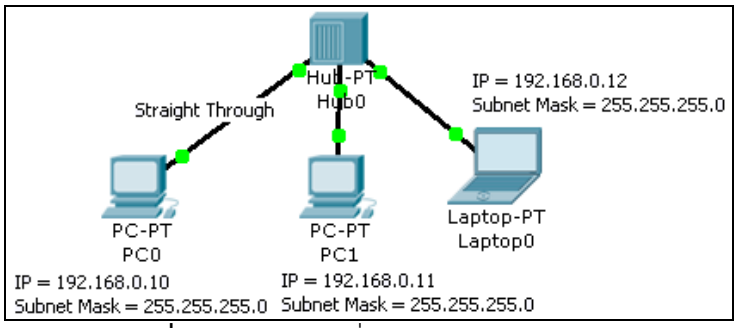
ค่าของ SOURCE IP เป็น 192.168.0.10 และ TARGET IP เป็น 192.168.0.12 เมื่อตอบกลับ จะสลับค่าเป็น SOURCE IP เป็น 192.168.0.12 และ TARGET IP เป็น 192.168.0.10 เช่นเดียวกัน ค่าของ MAC ก็จะสลับตามหมายเลข IP สำหรับแพ็คเก็ตอื่นๆ ก็จะสามารถสังเกตุได้ด้วยวิธีการเดียวกัน

Scenario 5: หลักการทำงานของ ARP โพรโทคอล

คำอธิบาย :

ARP เป็นโพรโทคอลที่มีความสำคัญมากในการสื่อสารข้อมูล และเป็นจุดอ่อนที่ Hacker นิยมใช้การดักจับข้อมูลด้วย ดังนั้นใน Scenario นี้จะมาทดลองวิเคราะห์แพ็คเก็ตของ ARP กันว่ามีหลักการทำงานเป็นอย่างไร

แผนผังการเชื่อมต่อ :



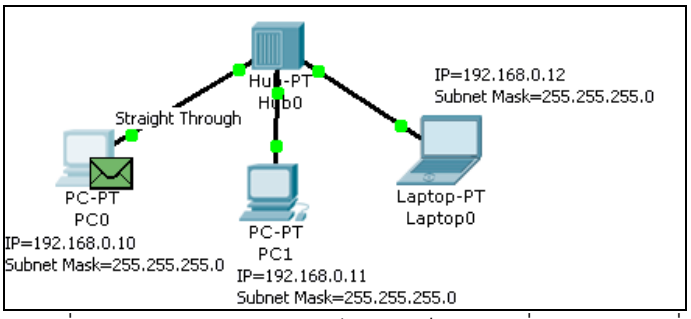
รูปที่ 12.25 ผังการเชื่อมต่อของ scenario 5

ขั้นตอนการวิเคราะห์ :

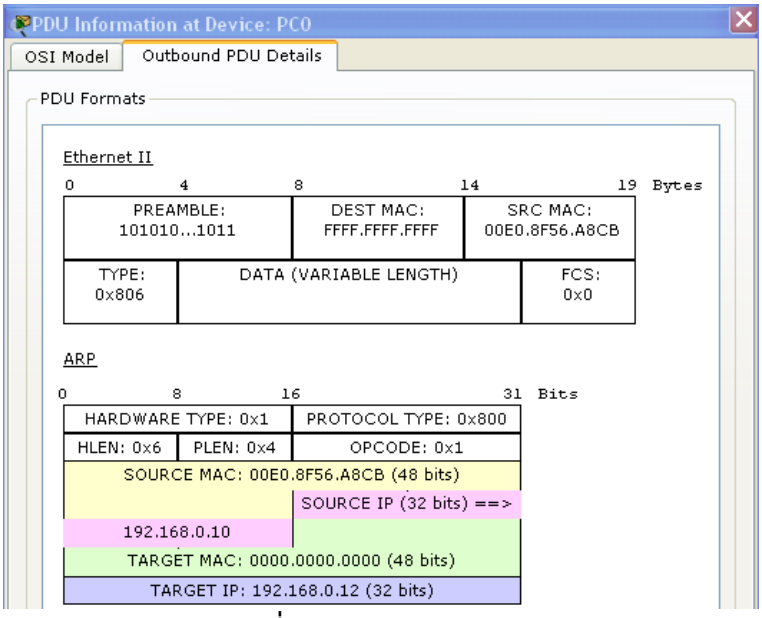
1. เริ่มต้นให้เลือกที่โหมด Simulation ก่อน เครื่อง PC0 ให้ใช้คำสั่ง ping จาก Command Prompt ไปยังเครื่อง Labtop0 (จาก IP 192.168.0.10 ไปยัง 192.168.0.12)

```
PC>ping 192.168.0.12
```

2. ในส่วน Even List Filters ให้เลือก Edit Filters ⇨ คลิกบล็อกซ์ Show All/None ออก ⇨ คลิกบล็อกซ์ ARP เพียงโพรโทคอลเดียวเท่านั้น เมื่อออกคำสั่ง ping แล้ว ให้ใช้ Inspect ตรวจสอบแพ็คเก็ต ที่มีรูปเป็นซอง สีเขียวคือ ARP ■ โดยการคลิกที่ซอง จดหมายที่ต้องการ ดังรูปที่ 12.26



รูปที่ 12.26 จะเริ่มส่ง ARP request ออกไปบนเน็ตเวิร์คเพื่อค้นหาว่าเครื่องใด คือ IP 192.168.0.12

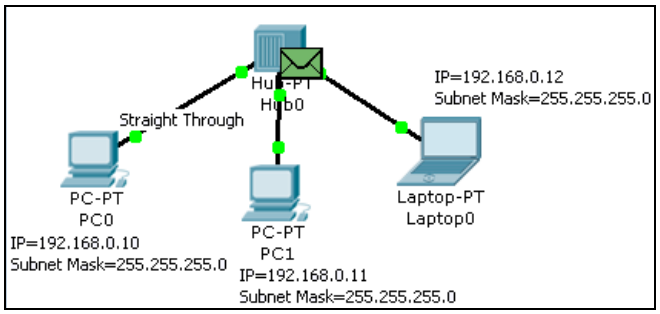


รูปที่ 12.27 ARP request

ข้อมูลของแพ็คเก็ตในเลเยอร์ที่ 2 ขาออก (Outbound PDU ของ การ์ดเน็ตเวิร์ค PC0) ประกอบไปด้วย Ethernet frame และ ARP frame ที่ซ่อนอยู่ใน DATA ของ Ethernet frame ข้อมูลใน Ethernet frame ที่สำคัญคือ DEST MAC มีค่าเป็น FFF.FFF.FFF คือการถามไปยังทุกๆ เครื่องในเครือข่าย (broadcast frame) ว่าใครที่มีหมายเลข IP เป็น 192.168.0.12 (อยู่ใน TARGET IP ของ ARP frame) และ SRC MAC แสดงถึง MAC Address ของผู้ส่งข้อมูล (ในที่นี้คือ PC0=00E0.8F56.A8CB) สำหรับใน ARP frame เบื้องต้นจะรู้เฉพาะว่า SOURCE MAC=00E0.8F56.A8CB, SOURCE IP=192.168.0.10 ซึ่งเป็นเครื่อง PC0 ที่ส่งข้อมูลออกมา และรู้ว่าจะต้องส่งไปที่ IP 192.168.0.12 (TARGET IP)

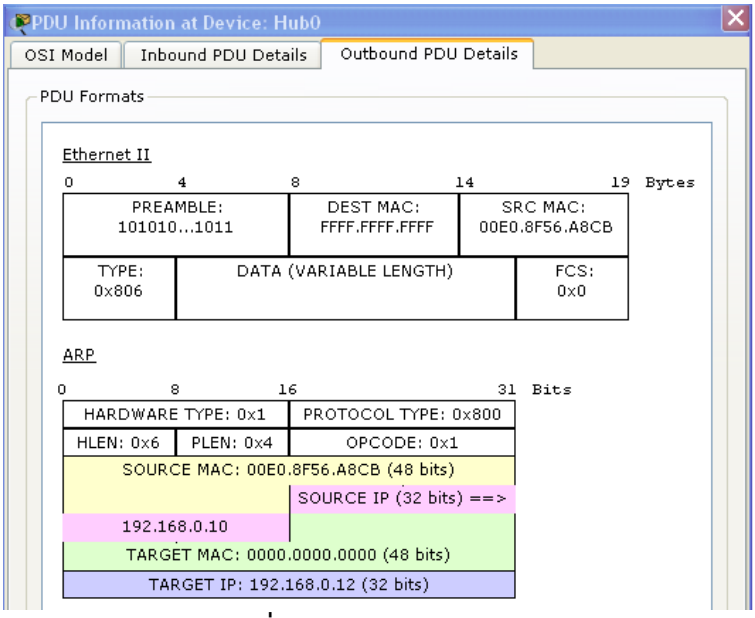
แต่ไม่รู้ว่าเป้าหมายเป็น MAC Address หมายเลขใด (TARGET MAC=000.000.000) จึงต้องอาศัยให้ Ethernet frame ส่งข้อมูลกระจายไปทั้งเครือข่าย เพื่อให้คนที่มีหมายเลข IP 192.168.0.12 ตอบกลับมา

3. แพ็คเก็ตจะถูกส่งจาก PC0 ไปยัง HUB ดังรูปที่ 12.28



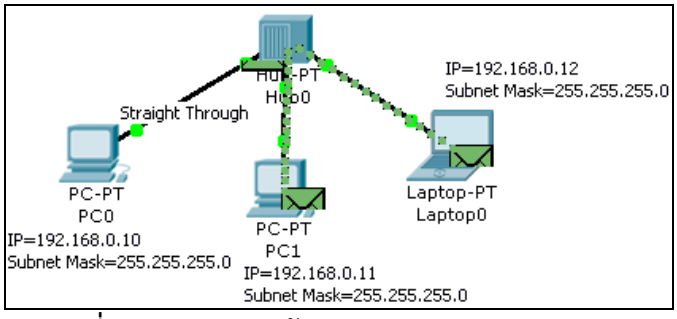
รูปที่ 12.28 แพ็คเก็ต Ethernet และ ARP จะถูกส่งจาก PC0 ไปยัง HUB

ข้อมูลที่ HUB รับเข้ามาเรียกว่า Inbound PDU และ ส่งออกจาก HUB เรียกว่า Outbound PDU ซึ่งในสถานะตอนนี้จะมีค่าเหมือนกันทุกประการ ดังรูปที่ 12.29



รูปที่ 12.29 Inbound PDU

4. แพ็คเก็ตจะถูกส่งออกจาก HUB ไปยังทุกๆ พอร์ตพร้อมกัน ยกเว้นพอร์ตที่รับแพ็คเก็ตเข้ามา เพื่อค้นหาว่า MAC Address ของ IP 192.168.0.12 คือเครื่องใด



รูปที่ 12.30 ส่งเฟรมข้อมูลกระจายแบบ Broadcast

เครื่อง Labtop0 ซึ่งเป็นเจ้าของ IP ดังกล่าวตอบกลับด้วย ARP Reply พร้อมกับ update ข้อมูลใน frame Ethernet และ ARP ดังนี้

เครื่อง Labtop0 แพ็คเก็ตขาเข้า

Inbound PDU

Ethernet frame

SRC MAC=00E0.8F56.A8CB (เครื่อง PC0)

DEST MAC=FFFF.FFFF.FFFF (ทุกๆ เครื่อง)

ARP frame

SOURCE MAC: 00E0.8F56.A8CB (เครื่อง PC0)

TARGET MAC: 0000.0000.0000 (ยังไม่ทราบว่าเครื่องใด)

OPCODE: 0x1 (ARP Request)

SOURCE IP=192.168.0.10 (IP เครื่อง PC0)

TARGET IP: 192.168.0.12 (IP เครื่อง Labtop0)

เครื่อง Labtop0 แพ็คเก็ตขาออก

Outbound PDU

Ethernet frame

SRC MAC= 0001.6487.133E (เครื่อง Labtop0)

DEST MAC=00E0.8F56.A8CB (เครื่อง PC0)

ARP frame

SOURCE MAC: 0001.6487.133E (เครื่อง Labtop0)

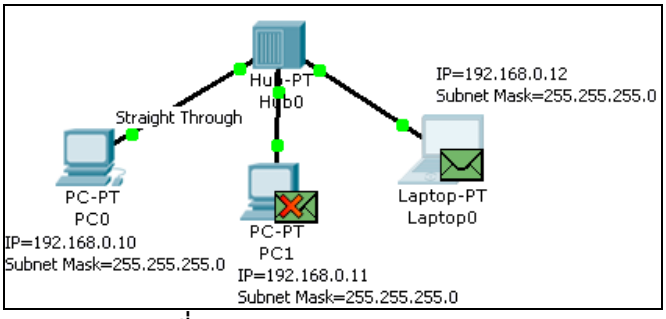
TARGET MAC: 00E0.8F56.A8CB (เครื่อง PC0)

OPCODE: 0x2 (ARP Reply)

SOURCE IP=192.168.0.10 (IP เครื่อง PC0)

TARGET IP: 192.168.0.12 (IP เครื่อง Labtop0)

5. เมื่อแพ็คเก็ตเดินทางจาก Labtop0 กลับไปยัง HUB เพื่อกลับไปให้เครื่องที่เรียกมา คือ PC1, HUB จะทำการกระจายข้อมูลที่รับมาจาก Labtop0 ไปยังทุกๆ พอร์ตตามหน้าที่ของมัน แต่เครื่อง PC1 จะไม่รับแพ็คเก็ตดังกล่าว เนื่องจากไม่ใช่ IP ที่แพ็คเก็ตต้องการส่งให้ แต่สำหรับเครื่อง PC1 จะรับแพ็คเก็ตดังกล่าวไว้ เป็นอันจบกระบวนการของ ARP ดังรูปที่ 12.31



รูปที่ 12.31 กระบวนการของ ARP

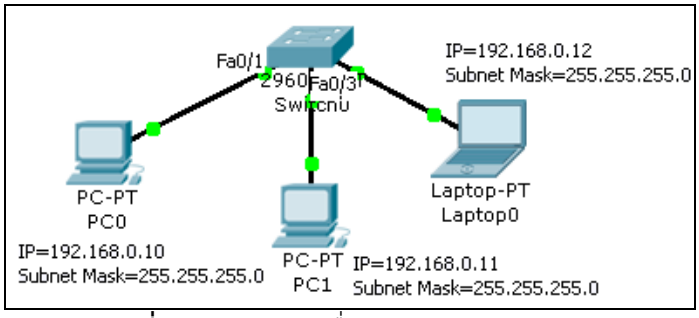
การบริหารเครื่องแม่ข่ายและเครื่องลูกข่าย การแก้ไข เพิ่มเติม อุปกรณ์และผู้ใช้ใหม่ในระบบ

Scenario 6: เชื่อมต่อคอมพิวเตอร์ PC, Labtop กับ Switch L2 (เลเยอร์ 2)

คำอธิบาย :

การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์มากกว่า 2 เครื่องขึ้นไป โดยใช้อุปกรณ์ HUB นั้นไม่มีความปลอดภัย เนื่องจาก HUB จะกระจายข้อมูลที่รับเข้ามาออกไปยังทุกๆ พอร์ต ดังนั้นจึงเป็นเครื่องมือที่ Hacker นิยมใช้ในการดักจับข้อมูล ซึ่งแตกต่างจากอุปกรณ์ประเภทสวิตช์ที่ไม่มีการกระจายข้อมูลไปยังเครื่องลูกข่าย ซึ่งจะช่วยลดปัญหาการดักจับข้อมูลได้ระดับหนึ่ง

แผนผังการเชื่อมต่อ :



รูปที่ 12.32 ผังการเชื่อมต่อของ scenario 6

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	Interface Type	Cable Type
PC0	192.168.0.10	255.255.255.0	FastEthernet	Straight-Through
PC1	192.168.0.11	255.255.255.0	FastEthernet	Straight-Through
Labtop0	192.168.0.12	255.255.255.0	FastEthernet	Straight-Through
Switch0	-	-	FastEthernet0/1 to PC0 FastEthernet0/2 to PC1 FastEthernet0/3 to Labtop0	Straight-Through

ขั้นตอนการเชื่อมต่อ :

1. เลือก End Devices  (ในส่วน Device-Type)
2. เลือก Generic  (ในส่วน Device-Specific) แล้วลากมาวางใน workspace ให้ครบ 2 เครื่อง (โดยปกติจะมีชื่อเป็น PC0, PC1 ตามลำดับ)
3. เลือก Labtop-PT  (ในส่วน Device-Specific) แล้วลากมาวางใน workspace
4. เลือก Connections  (ในส่วน Device-Type)
5. เลือก Copper Straight-Through  (ในส่วน Device-Specific) แล้วคลิกที่เครื่อง PC0 ไปยัง HUB และ PC1, Labtop0 ไปยัง Switch0 ตามลำดับ ที่เครื่องคอมพิวเตอร์จะปรากฏไฟสีเขียว (ถ้าไฟเป็นสีส้มแสดงว่ายังอยู่ในช่วงการเชื่อมต่ออยู่ รอประมาณ 30 วินาที) แสดงว่าเชื่อมต่อสำเร็จ
6. คลิกที่ PC0, PC1 และ Laptop0 ทีละเครื่อง ในแต่ละเครื่องเลือกแท็บ Desktop ⇨ IP Configuration ⇨ Static ⇨ กำหนดค่าหมายเลข IP Address และ Subnet Mask ดังต่อไปนี้

เครื่อง	IP Address	Subnet Mask
PC0	192.168.0.10	255.255.255.0
PC1	192.168.0.11	255.255.255.0
Labtop0	192.168.0.12	255.255.255.0

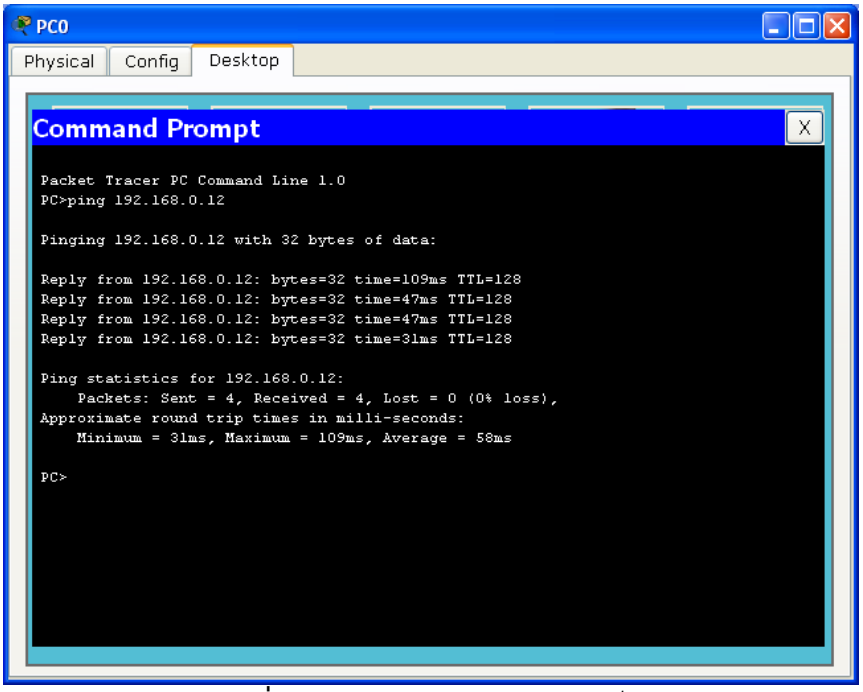
7. เสร็จสิ้นการเชื่อมต่อ

การทดสอบ :

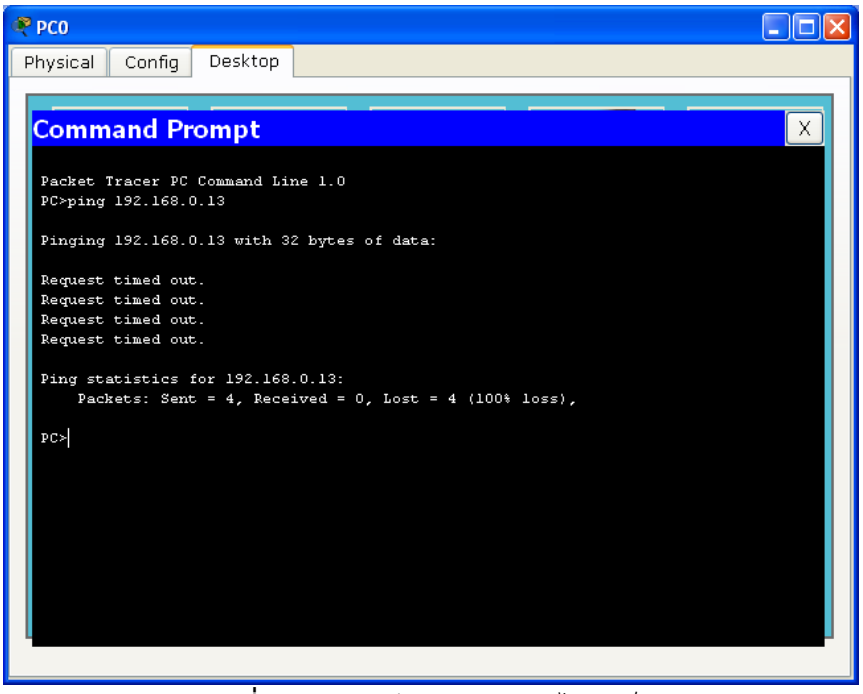
1. ที่เครื่อง PC0 เลือก Desktop ⇨ Command Prompt
2. เมื่อปรากฏหน้าต่าง Command Prompt ให้ผู้ใช้ออกคำสั่งทดสอบคือ ping หมายเลข IP Address 192.168.0.12

```
PC>ping 192.168.0.12
```

3. เครื่องที่ถูก ping จะตอบกลับด้วย ICMP Reply เมื่อเครื่องปลายทางทำงานปกติ แต่ถ้า ping แล้ว ผลที่ได้รับคือ Request timed out แสดงว่าเครื่องปลายทางไม่ได้ทำงาน ดังรูปที่ 12.33, 12.34

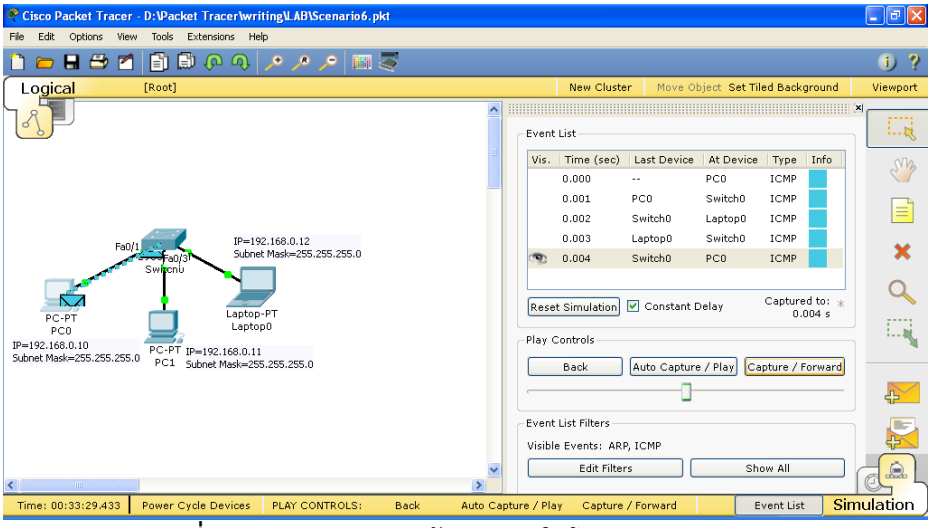


รูปที่ 12.33 กรณีทดสอบ ping สำเร็จ



รูปที่ 12.34 กรณีทดสอบ ping ไม่สำเร็จ

4. ลองทดสอบด้วย ping อีกครั้ง ในโหมด Simulation สังเกตุพฤติกรรมการทำงานใน Even List ปรากฏว่าแพ็คเก็ตจะไม่กระจายไปยังทุกๆ พอร์ตเหมือนกรณีของ HUB ทำให้ข้อมูลที่ส่งและรับมีความปลอดภัยเพิ่มขึ้น ดังรูปที่ 12.35

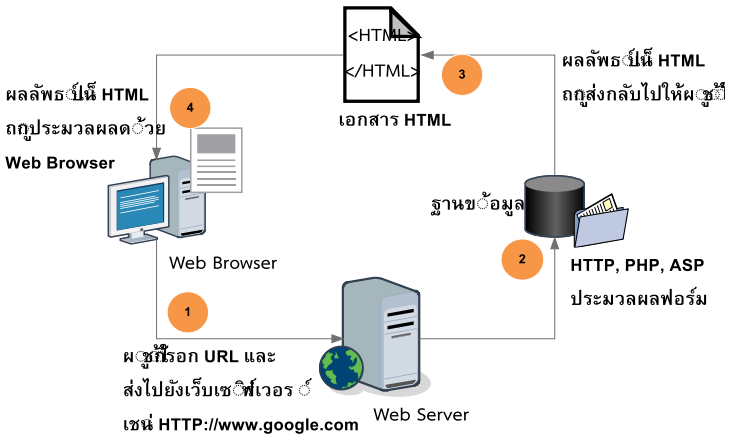


รูปที่ 12.35 การทดสอบด้วย ping ในโหมด Simulation

Scenario 7: การติดตั้งเว็บเซิร์ฟเวอร์ (Web Server : HTTP)

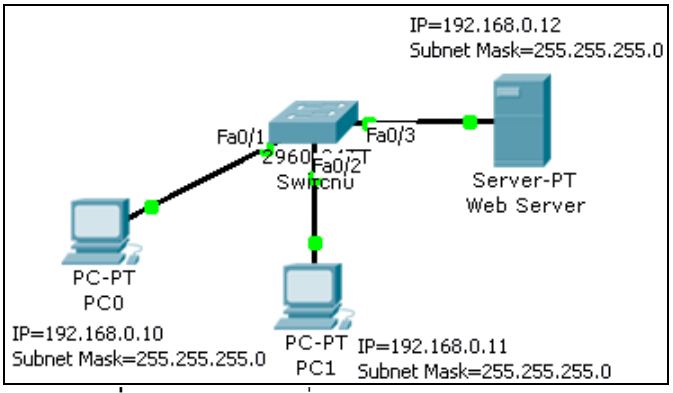
คำอธิบาย :

เว็บเซิร์ฟเวอร์ (Web Server) คือ เครื่องคอมพิวเตอร์ที่ทำหน้าที่ให้บริการเว็บเพจให้แก่ผู้ร้องขอ ด้วยโปรแกรมประเภทเว็บบราวเซอร์ (Web Browser เช่น Internet Explorer, FireFox เป็นต้น) โดยร้องขอข้อมูลผ่านโปรโตคอลเฮชทีทีพี (HTTP = Hyper Text Transfer Protocol) เครื่องผู้ให้บริการจะส่งข้อมูลให้กับผู้ร้องขอในรูปของข้อความ ภาพ เสียง หรือสื่อผสม (Multimedia) เครื่องให้บริการเว็บจะเปิดพอร์ท 80 (เป็นพอร์ตที่นิยม แต่ผู้ให้บริการก็สามารถเปลี่ยนเป็นพอร์ตอื่นๆ ก็ได้ เช่น 8080 เป็นต้น) เครื่องผู้ใช้เริ่มการเชื่อมต่อโดยการระบุที่อยู่เว็บเพจที่ร้องขอ (Web Address หรือ URL = Uniform Resource Locator) เช่น http://www.google.co.th เป็นต้น สำหรับโปรแกรมที่นิยมใช้เป็นเครื่องให้บริการเว็บ เช่น Apache Web Server, IIS (Internet Information Server) เป็นต้น หลักการทำงานดังแสดงในรูปที่ 12.36



รูปที่ 12.36 ขั้นตอนการทำงานของเว็บเซิร์ฟเวอร์

แผนผังการเชื่อมต่อ :



รูปที่ 12.37 ผังการเชื่อมต่อสำหรับ scenario 7

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	Interface Type	Cable Type
PC0	192.168.0.10	255.255.255.0	FastEthernet	Straight-Through
PC1	192.168.0.11	255.255.255.0	FastEthernet	Straight-Through
Server-PT	192.168.0.12	255.255.255.0	FastEthernet	Straight-Through
Switch0	-	-	FastEthernet0/1 to PC0 FastEthernet0/2 to PC1 FastEthernet0/3 to Server-PT	Straight-Through

ขั้นตอนการเชื่อมต่อ :

1. เลือก End Devices (ในส่วน Device-Type)
2. เลือก Generic (ในส่วน Device-Specific) แล้วลากมาวางใน workspace ให้ครบ 2 เครื่อง (โดยปกติจะมีชื่อเป็น PC0, PC1 ตามลำดับ)
3. เลือก Server-PT (ในส่วน Device-Specific) แล้วลากมาวางใน workspace
4. เลือก Connections (ในส่วน Device-Type)
5. เลือก Copper Straight-Through (ในส่วน Device-Specific) แล้วคลิกที่เครื่อง PC0 ไปยัง HUB และ PC1, Server-PT ไปยัง Switch0 ตามลำดับ ที่เครื่องคอมพิวเตอร์จะปรากฏไฟสีเขียว (ถ้าไฟเป็นสีส้มแสดงว่ายังอยู่ในช่วงการเชื่อมต่ออยู่ รอประมาณ 30 วินาที) แสดงว่าเชื่อมต่อสำเร็จ
6. คลิกที่ PC0, PC1 ทีละเครื่อง ในแต่ละเครื่องเลือกแท็บ Desktop ⇨ IP Configuration ⇨ Static ⇨ กำหนดค่าหมายเลข IP Address และ Subnet Mask ดังตารางด้านล่าง

เครื่อง	IP Address	Subnet Mask

PC0	192.168.0.10	255.255.255.0
PC1	192.168.0.11	255.255.255.0

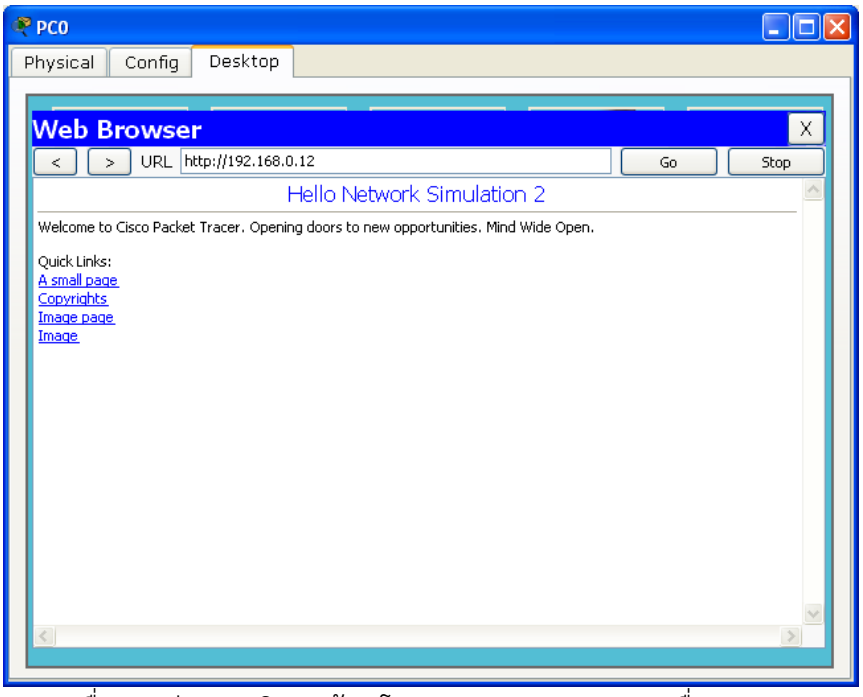
7. ทำการ Enable Web Server โดยเลือกที่ Desktop ⇨ Config ⇨ เลือกแท็บ HTTP ⇨ ตรวจสอบ HTTP และ HTTPS อยู่ในสถานะ On หรือยัง ถ้ายังให้เลือกเป็น On
8. เครื่อง HTTP จะมีไฟล์ 3 ไฟล์ให้ผู้ใช้สามารถปรับแต่งได้ ซึ่งเขียนภายใต้ภาษา HTML คือ ไฟล์ index.html, helloworld.html, image.html ผู้ใช้สามารถแก้ไขได้ เช่น ในไฟล์ index.html จาก HTML **Cisco Packet Tracer** ทดสอบแก้ไขเป็น **Hello Network Simulation 2** เป็นต้น
9. ในแท็บ Desktop เลือก IP Configuration กำหนดค่าดังต่อไปนี้

   IP Address = 192.168.0.12

   Subnet Mask = 255.255.255.0
10. เสร็จสิ้นการเชื่อมต่อ

การทดสอบ :

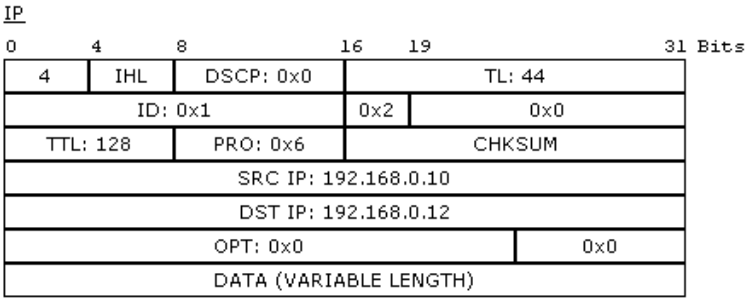
1. ให้ทำการทดสอบเว็บเซิร์ฟเวอร์ โดยการคลิกที่ PC0 ⇨ Desktop ⇨ Web Browser ⇨ ให้กรอกในช่อง URL เป็น http://192.168.0.12 (ยังไม่สามารถเรียกแบบชื่อได้ เช่น www.google.co.th เนื่องจากยังไม่มีการติดตั้ง DNS) แล้วกดปุ่ม Go ดังรูปที่ 12.48



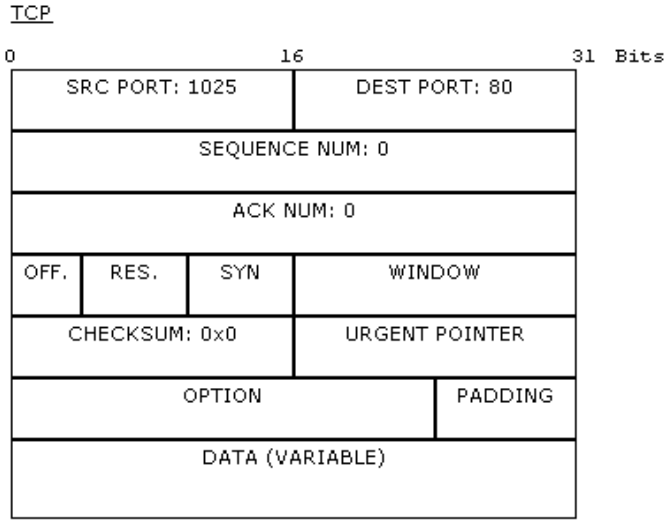
รูปที่ 12.38 เมื่อทุกอย่างคอนฟิกถูกต้อง โปรแกรม Browser ของเครื่อง PC0 จะแสดงผลที่ส่งมาจาก Web Server ได้ถูกต้อง

การวิเคราะห์แพ็คเก็ต HTTP:

1. เลือกการทำงานเป็นโหมด Simulation
2. สำรวจตาราง ARP Table ของ PC0, PC1, Server-PT และ Switch0 โดยใช้ Inspect (ในเบื้องต้นจะต้องไม่มีการเก็บข้อมูลใดๆ ในตารางดังกล่าว) สำหรับ Port Status ต้องมีสถานะเป็น up
3. คลิกเลือก PC0 ⇨ Desktop ⇨ Web Browser ⇨ ป้อน URL เป็น http://192.168.0.12 แล้วกดปุ่ม Go
4. แพ็คเก็ตจะหยุดรอให้ผู้ใช้ควบคุมการทำงานของแพ็คเก็ต โดยผู้ใช้เลือกเป็น Capture/Forward เป็นการเลือกการทำงานแบบ Step by Step
5. ข้อมูลใน Even List จะเห็นว่าแพ็คเก็ตจะถูกส่งออกมาพร้อมกัน 2 ประเภทคือ TCP/IP (แพ็คเก็ตของ HTTP ซึ่งใช้ port 80) และแพ็คเก็ตที่ 2 คือ ARP โปรโทรคอล



แพ็คเก็ต IP ทำหน้าที่ส่งข้อมูลไปในเส้นทางที่ดีที่สุดบนเครือข่าย ฟิลด์ที่สำคัญประกอบไปด้วย SRC IP คือ IP ต้นทางที่ต้องการร้องขอจากเว็บเซิร์ฟเวอร์ (192.168.0.10), DST IP คือ IP ของเครื่องเว็บเซิร์ฟเวอร์ (192.168.0.12) ข้อมูลที่อยู่ใน DATA คือ แพ็คเก็ตของ TCP

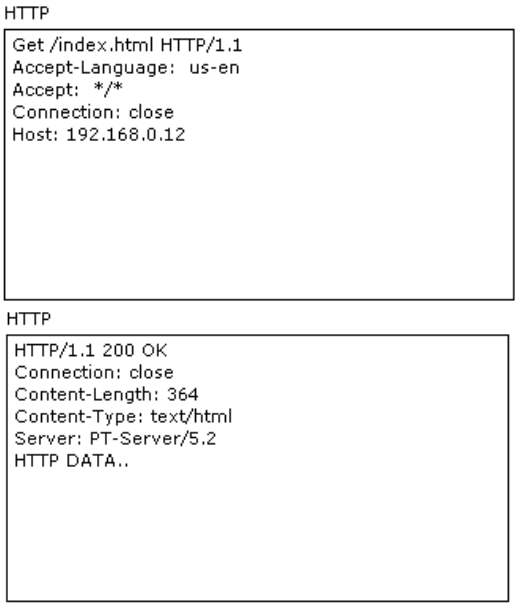


สำหรับแพ็คเก็ต TCP ทำหน้าที่ส่งข้อมูลให้ครบและถูกต้อง ฟิลด์ที่สำคัญได้แก่ SRC PORT เป็นหมายเลขพอร์ตต้นทางของเครื่องผู้ใช้ (PC0), DEST PORT เป็นหมายเลขพอร์ตที่ต้องการเชื่อมต่อบนเครื่องเว็บเซิร์ฟเวอร์ (พอร์ต 80), SEQUENCE NUM คือลำดับการส่งข้อมูลของแพ็คเก็ต, ACK NUM หมายถึง สถานะการทำงาน, WINDOW คือ จำนวนขนาดของหน้าต่างที่ใช้รับส่งข้อมูล, DATA เป็นส่วนที่ใช้เก็บข้อมูลของแพ็คเก็ตของโพรโทคอล HTTP (เว็บ)

6. สำหรับใน Switch0 นั้นเริ่มต้นค่าในตารางต่างๆ เช่น ARP, MAC, QoS Queues จะว่างเปล่า แต่ Port Status จะมีสถานะเป็น up ทั้งหมด 3 ports คือ FastEthernet 0/1 (ต่อกับ PC0), FastEthernet 0/2 (PC1) และ FastEthernet 0/3 (Web Server)
7. กดปุ่ม Capture/Forward อีกครั้ง พร้อมสังเกตุแพ็คเก็ตที่ปรากฏในแท็บ Event List

ข้อสังเกตุ : แพ็คเก็ตของ HTTP จะยังไม่สามารถทำงานได้ในทันที จำเป็นต้องให้กระบวนการของ ARP เสร็จสิ้นก่อน (อ่านเพิ่มเติมในหัวข้อ หลักการทำงานของ ARP)

8. สำหรับข้อมูลในฟิลด์ของ DATA ในแพ็คเก็ต TCP นั้นคือ ข้อมูลของแพ็คเก็ต HTTP ซึ่งประกอบไปด้วย วิธีการเชื่อมต่อ เช่น Get หรือ Post, ชื่อไฟล์ที่ต้องการแสดงผล (index.html), เวอร์ชันของ HTTP (1.1), ภาษาที่ใช้สื่อสาร (Accept-Language: us-en), สถานะการเชื่อมต่อ (Connection: close), หมายเลข IP ที่ต้องการเชื่อมต่อ (Host: 192.168.0.12) เป็นต้น ดังรูปที่ 12.39



รูปที่ 12.39 ตัวอย่างข้อมูลในแพ็คเก็ต HTTP

9. เมื่อการร้องขอบริการเว็บทำงานสำเร็จลง ข้อมูลต่างๆ ในอุปกรณ์จะมีสถานะดังนี้

PC0 (Web Browser)

ARP Table			
IP Address	**MAC Address**	**Interface**	**Comment**
192.168.0.12	0030.A398.4866	FastEthernet	เครื่อง Web Server

Server-PT (Web Server)

ARP Table			
IP Address	**MAC Address**	**Interface**	**Comment**
192.168.0.10	000A.4193.A5E1	FastEthernet	เครื่อง PC0

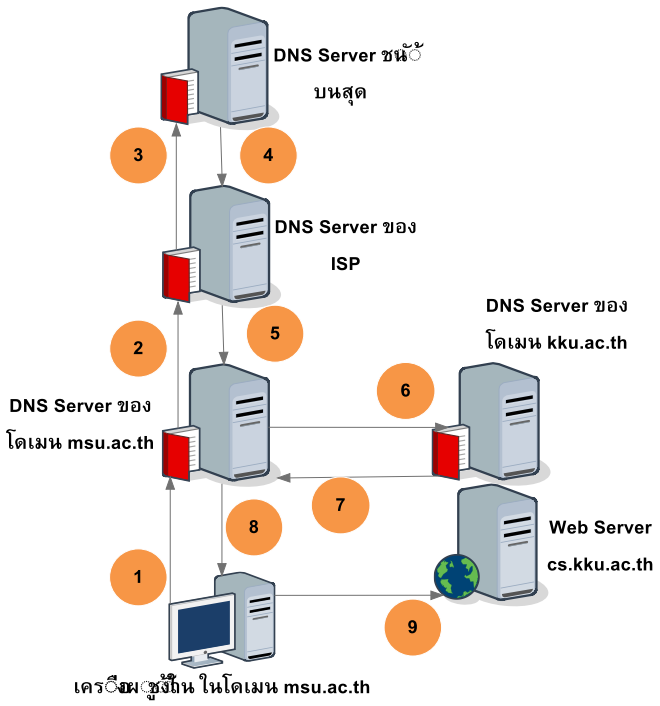
Switch0

ARP Table			
VLAN	MAC Address	Port	Comment
1	000A.4193.A5E1	FastEthernet0/1	เครื่อง PC0
1	0030.A398.4866	FastEthernet0/3	เครื่อง Web Server

Scenario 8: การติดตั้งโดเมนเนมเซิร์ฟเวอร์ (DNS)

คำอธิบาย :

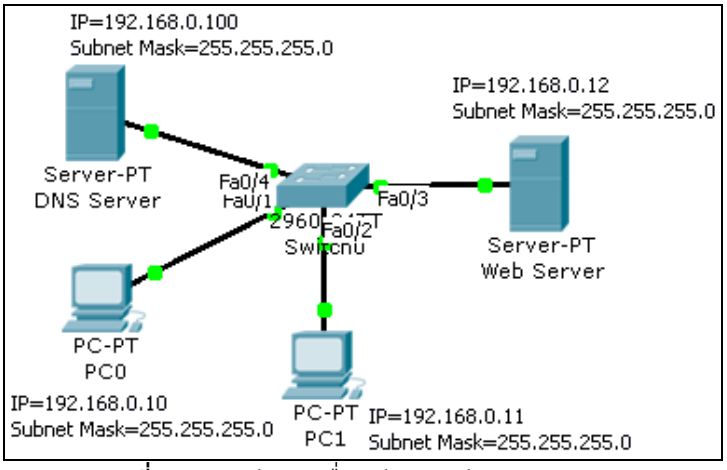
DNS ย่อมาจาก Domain Name System หมายถึง ระบบใช้สำหรับอ้างถึงหมายเลขเครื่อง หรือหมายเลข IP Address เข้ากับชื่อของเว็บไซต์ เพื่อให้ง่ายต่อการจดจำ DNS จะทำหน้าที่คล้ายกับสมุดโทรศัพท์ คือเมื่อมีผู้ใช้ต้องการจะโทรศัพท์หาบุคคลใด บุคคลหนึ่ง จะต้องเปิดสมุดโทรศัพท์เพื่อค้นหาเบอร์โทรศัพท์ของบุคคลที่ต้องการจะติดต่อด้วย คอมพิวเตอร์ก็เช่นกัน เมื่อต้องการจะสื่อสารกับคอมพิวเตอร์เครื่องอื่น เครื่องนั้นก็จะทำการสอบถามหมายเลข IP ของเครื่องที่ต้องการจะสื่อสาร กับ DNS server ซึ่งจะทำการค้นหาหมายเลขดังกล่าว การเชื่อมต่อสื่อสารระหว่างคอมพิวเตอร์ในระบบ internet นั้นใช้มาตรฐาน TCP/ IP ที่เครื่องคอมพิวเตอร์นั้นต้องมีหมายเลข IP Address ไม่ซ้ำกัน โดยปกติเครื่อง Web Server จำเป็นต้องมี IP Address เสมอ จึงเกิดปัญหาในการจำ เพราะว่า IP Address มีตัวเลขถึง 12 ตัว จากจุดนี้จึงได้มีการคิดที่จะแปลง IP Address ให้เป็นชื่อที่จำได้ง่าย เช่น IP Address " 64.233.181.106" เรียกเป็น HTTP://www.google.co.th" (กูเกิ้ล) เป็นต้น สำหรับลำดับการทำงานของ DNS มีขั้นตอนดังรูป 12.40



รูปที่ 12.40 ขั้นตอนการทำงานของ DNS เซิร์ฟเวอร์

1. เครื่องผู้ใช้หรือ client อยู่ในโดเมน msu.ac.th ร้องของบริการเว็บชื่อว่า cs.kku.ac.th ซึ่งจะค้นหาในเครื่องของตนเองก่อน (local DNS ในวินโดวส์จะเก็บใน \system32\drivers\etc\hosts, linux อยู่ใน /etc/resolv.conf) เมื่อค้นหาไม่เจอจะสอบถามไปที่ DNS ของหน่วยงานของตนเองก่อน
2. เมื่อค้นหาใน DNS Server ของ msu.ac.th เมื่อไม่เจอจะค้นหาต่อไปยัง ISP ที่เชื่อมต่ออยู่ในระดับที่สูงขึ้น
3. สมมุติว่าค้นหาที่ ISP ก็ไม่เจอ จะส่งการร้องขอไปยัง Root DNS ซึ่งให้บริการอยู่ทั่วโลก โดยปกติจะต้องเจอ ถ้าไม่เจอแสดงว่าชื่อที่ค้นหาไม่มีในโลกนี้
4. Root DNS จะส่งที่อยู่ของ DNS ที่ ISP บริหารโดเมน kku.ac.th อยู่กลับมา
5. DNS ของ msu.ac.th ทราบว่า kku.ac.th อยู่ที่ไหน
6. DNS ของ msu.ac.th ส่งคำร้องขอไปยัง kku.ac.th
7. kku.ac.th จะตอบกลับเป็นหมายเลข IP ของ cs.kku.ac.th กลับมาให้
8. ได้รับหมายเลข IP ของ cs.kku.ac.th
9. เชื่อมต่อไปยัง Web Server ดังกล่าวด้วยหมายเลข IP ที่ได้รับมา

แผนผังการเชื่อมต่อ :



รูปที่ 12.41 ผังการเชื่อมต่อสำหรับ scenario 8

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	Interface Type
PC0	192.168.0.10	255.255.255.0	FastEthernet
PC1	192.168.0.11	255.255.255.0	FastEthernet
Web Server	192.168.0.12	255.255.255.0	FastEthernet
DNS Server	192.168.0.100	255.255.255.0	FastEthernet
Switch0	-	-	FastEthernet0/1 to PC0 FastEthernet0/2 to PC1 FastEthernet0/3 to Web Server FastEthernet0/4 to DNS Server

ขั้นตอนการเชื่อมต่อ :

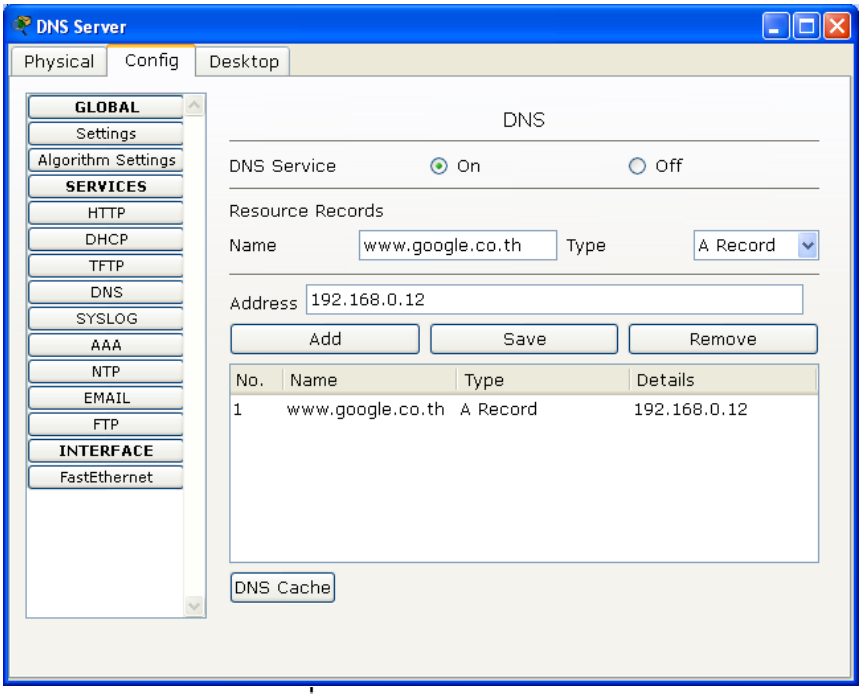
1. ให้ทำการเชื่อมต่อ PC0, PC1, Web Server เหมือนกับ Scenario 7
2. สำหรับเครื่อง DNS Server ให้ทำการคอนฟิก IP Address โดยเลือกที่ Desktop ⇨ IP Configuration ⇨กำหนด IP ดังนี้

   IP Address = 192.168.0.100

   Subnet Mask = 255.255.255.0
3. ทำการ Enable DNS Server โดยเลือกที่ Desktop ⇨ Config ⇨ เลือกแท็บ DNS ⇨ ตรวจสอบ DNS Service อยู่ในสถานะ On หรือยัง ถ้ายัง ให้เลือกเป็น On
4. ในส่วน Resource Records ฟิลด์ Name ให้ใส่ชื่อของเว็บเซิร์ฟเวอร์ ในที่นี้ทดลองใช้เป็น www.google.co.th, ฟิลด์ Type ให้เลือกเป็น A Record (A Record=มีความสำคัญมากที่สุด ซึ่งเป็นที่อยู่ของ IP (ขนาด 32 บิต) ของโฮสตัวหนึ่ง ซึ่งต้องมีอย่างน้อย 1 หมายเลข, CNAME=ชื่อที่ใช้เรียกแทน คล้ายชื่อเล่น, SOA=นำเสนอแหล่งที่มา

หลักข้อมูลเกี่ยวกับชื่อของ zone server, admin's email, flags และ Timeout, NS Record=ทำการระบุ Name Server)

5. ในฟิลด์ Address ให้ใส่หมายเลข IP Address ที่ต้องการใช้งาน ในที่นี้ใช้ IP 192.168.0.12 แปลงเป็นชื่อ www.google.co.th แล้วคลิกปุ่ม Add ดังรูป 12.42

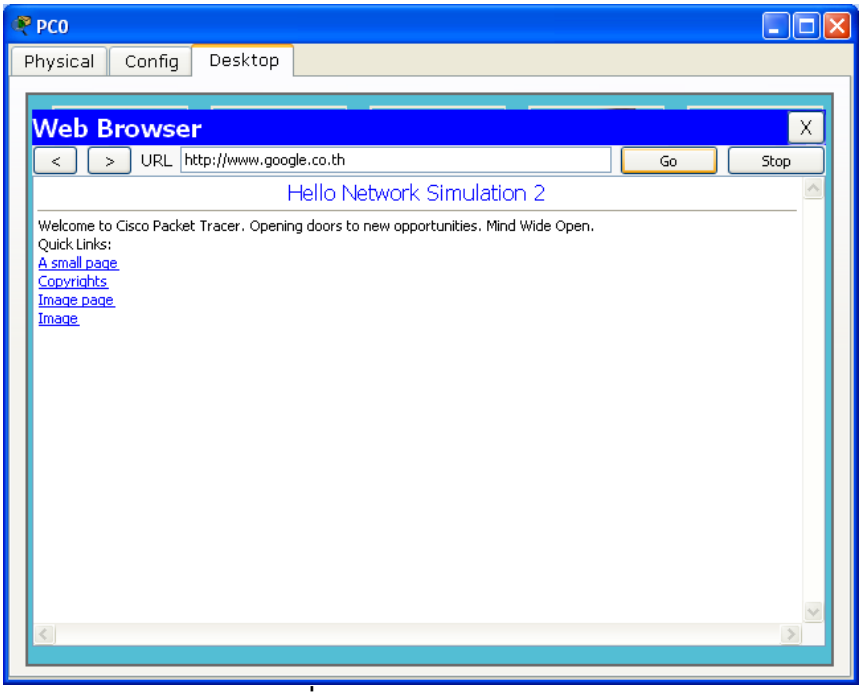


รูปที่ 12.42 การคอนฟิก DNS

6. ทำการกำหนดค่า DNS Server ในเครื่อง PC0 เพื่อทดสอบ DNS Sever ว่าใช้งานได้หรือไม่ โดยเลือกที่ Desktop ⇨ IP Configuration ⇨ ฟิลด์ DNS Server ให้ใส่ IP Address ของ DNS Server ในที่นี้คือ 192.168.0.100
7. ให้ทดสอบเว็บเซิร์ฟเวอร์อีกครั้งว่าทำงานอยู่หรือไม่ (IP 192.168.0.12) ทดสอบโดยการ ping หรือใช้ http://192.168.0.12
8. เสร็จกระบวนการเชื่อมต่อและคอนฟิกเครือข่าย

การทดสอบ :

1. ให้ทำการทดสอบ DNS และเว็บเซิร์ฟเวอร์ โดยการคลิกที่ PC0 ⇨ Desktop ⇨ Web Browser ⇨ ให้กรอกในช่อง URL เป็น http://www.google.co.th แล้วกดปุ่ม Go ดังรูปที่ 12.43

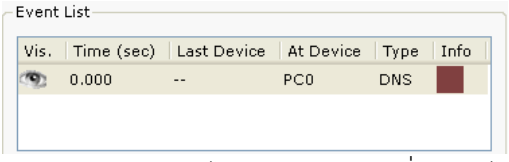


รูปที่ 12.43 การทดสอบ DNS

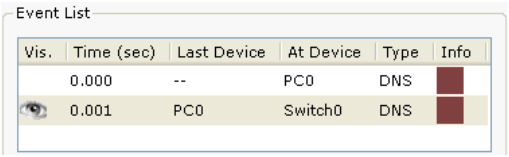
เมื่อทุกอย่างคอนฟิกถูกต้อง โปรแกรม Browser ของเครื่อง PC0 จะแสดงผลที่ส่งมาจาก Web Server ได้ถูกต้องคือ โดเมนเนมชื่อ www.google.co.th จะมี IP คือ 192.168.0.12

การวิเคราะห์แพ็คเก็ต HTTP:

1. เลือกการทำงานเป็นโหมด Simulation
2. ในส่วนนี้จะข้ามขั้นตอนการทำงานของ ARP (สามารถอ่านได้ใน Scenario 5) เพื่อให้เนื้อหากระชับขึ้น โดยเริ่มจาก PC0 ร้องขอผ่านทาง URL คือ www.google.co.th
3. PC0 จะร้องขอไปยัง DNS Server (IP 192.168.0.100) ที่เวลา 0.000 ใน Event List

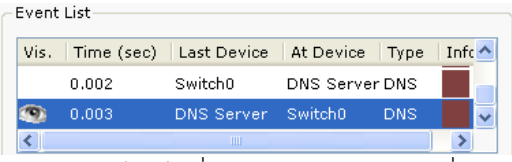


4. แพ็คเก็ตของ DNS จะถูกส่งจาก PC0 ไปยัง Switch0 เพื่อส่งต่อไปยังเครื่อง DNS Server ในเวลาที่ 0.001

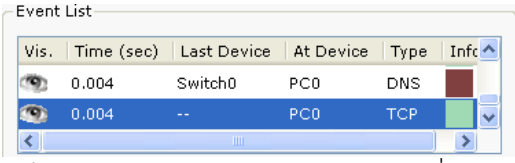


5. ในเวลาที่ 0.002 Switch0 ทำการกระส่งแพ็คเก็ตต่อไปยังเครื่อง DNS Sever เนื่องจาก Switch0 เรียนรู้แล้วว่า DNS Server อยู่ที่ใด (รู้ด้วยการโพรโทคอล ARP)

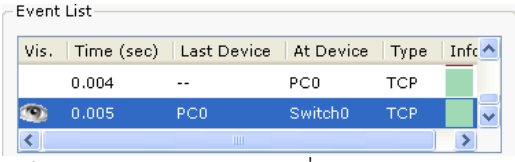
6. เวลาที่ 0.003 DNS Server ตอบกลับว่าจาก www.google.co.th เป็น 192.168.0.12 ส่งกลับไปให้ Switch0



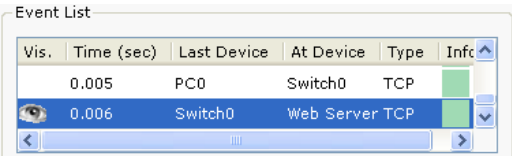
7. เวลาที่ 0.004 Switch0 ส่งแพ็คเก็ตที่ DNS ส่งให้ถึง PC0 เมื่อ PC0 ได้รับก็จะนำหมายเลข IP ดังกล่าวที่ได้รับ ร้องขอไปยังเว็บเซิร์ฟเวอร์ อีกครั้ง



8. เวลาที่ 0.005 แพ็คเก็ตจาก PC0 ไปถึง Switch0 (TCP) เพื่อไปยังเว็บเซิร์ฟเวอร์เพื่อขอเปิดพอร์ต 80 (พอร์ตที่ให้บริการเว็บ)

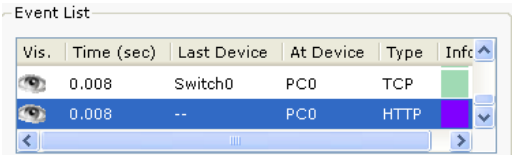


9. เวลาที่ 0.006 แพ็คเก็ตจาก Switch0 ไปถึงเครื่อง Web Server

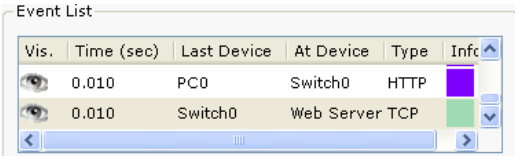


10. เวลาที่ 0.007 Web Server ตอบกลับ โดยยอมให้ทำการเปิดพอร์ต 80 ตามที่ร้องขอมา โดยส่งกับไปยัง Switch0

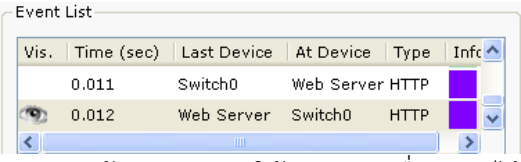
11. เวลาที่ 0.008 PC0 ได้รับตอบรับว่าเปิดพอร์ตแล้ว จึงร้องขอไปอีกครั้ง ด้วยโพรโทคอล HTTP



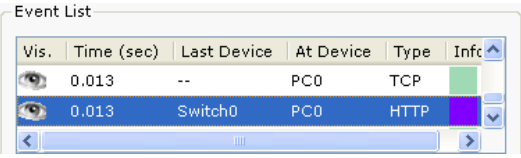
12. เวลาที่ 0.010 Switch0 ส่งแพ็คเก็ต HTTP ต่อไปยัง Web Server

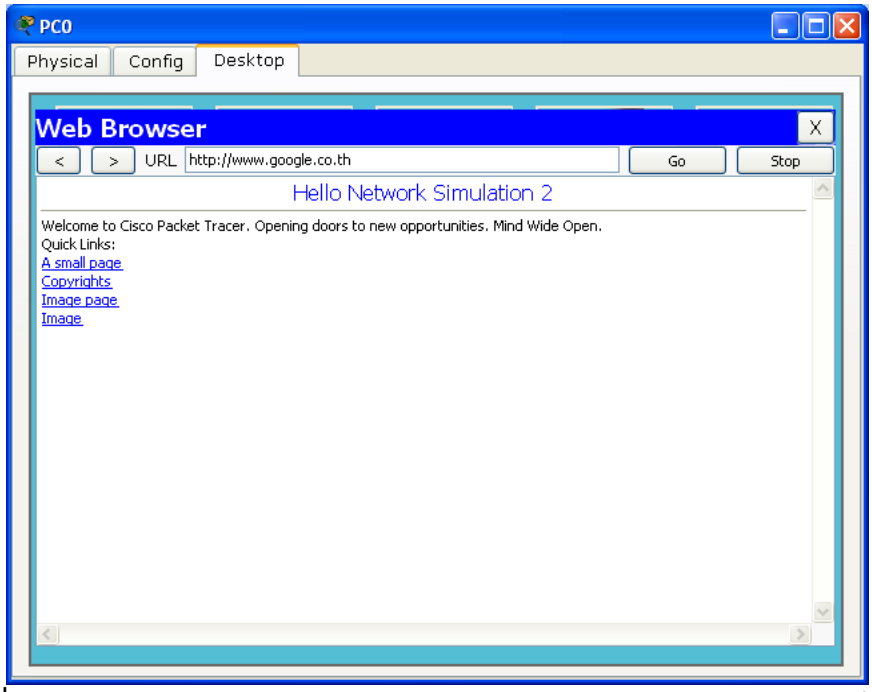


13. เวลาที่ 0.012 Web Server นำข้อมูล HTML ส่งกลับไปให้ Switch0



14. เวลาที่ 0.013 Switch0 ส่งข้อมูล HTML ให้กับ PC0 เมื่อ PC0 ได้รับก็จะทำการแสดงผลด้วย Web Browser ดังรูป 12.44





รูปที่ 12.44 กระบวนการทำงานของ DNS, Web Server และ Web Browser เสร็จสิ้น

Scenario 9: การติดตั้งอีเอชซีพีเซิร์ฟเวอร์ (DHCP)

คำอธิบาย :

DHCP หรือ Dynamic Host Configuration Protocol คือ โพรโทคอลที่ใช้ในการกำหนดเลขหมาย IP Address อัตโนมัติแก่เครื่องลูกข่ายบนระบบเครือข่ายที่ติดตั้งโพรโทคอล TCP/IP, สำหรับ DHCP server มีหน้าที่แจก IP ในเครือข่ายโดยไม่ซ้ำกัน เนื่องจากองค์กรหรือหน่วยงานที่มีเครื่องลูกข่ายมากๆ จะประสบปัญหาในการจัดสรรหมายเลข IP แบบกำหนดตายตัว (fix IP) โดยการทำงานนั้นจะเริ่มต้นเมื่อเครื่องลูกข่ายเริ่มเปิดเครื่องก็จะขอ IP address, Subnet mark, หมายเลข DNS และ Default gateway จาก DHCP Server อัตโนมัติ

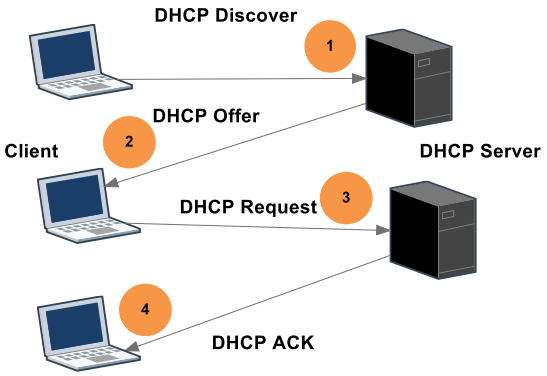
ขั้นตอนการเชื่อมต่อของเครื่องลูกข่ายกับ DHCP server มีดังนี้

1. เครื่องลูกข่ายค้นหาเครื่อง DHCP server ในเครือข่าย โดยส่ง DHCP discover เพื่อร้องขอ IP address

2. DHCP server จะค้นหา IP ที่ว่างอยู่ในฐานข้อมูล แล้วส่ง DHCP offer กลับไปให้เครื่องลูกข่าย

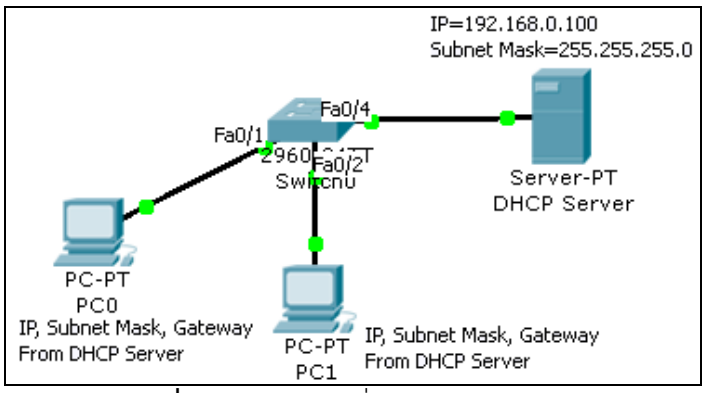
3. เมื่อเครื่องลูกข่ายได้รับ IP ก็จะส่งสัญญาณตอบกลับคือ DHCP Request ให้เครื่องแม่ทราบ

4. DHCP server ส่งสัญญาณ DHCP ACK กลับไปให้เครื่องลูกข่าย เพื่อแจ้งว่าเริ่มใช้งานได้ ดังรูปที่ 12.45



รูปที่ 12.45 ขั้นตอนการทำงานของ DHCP เซิร์ฟเวอร์

แผนผังการเชื่อมต่อ :



รูปที่ 12.46 ผังการเชื่อมต่อ scenario 9

รายการอุปกรณ์ :

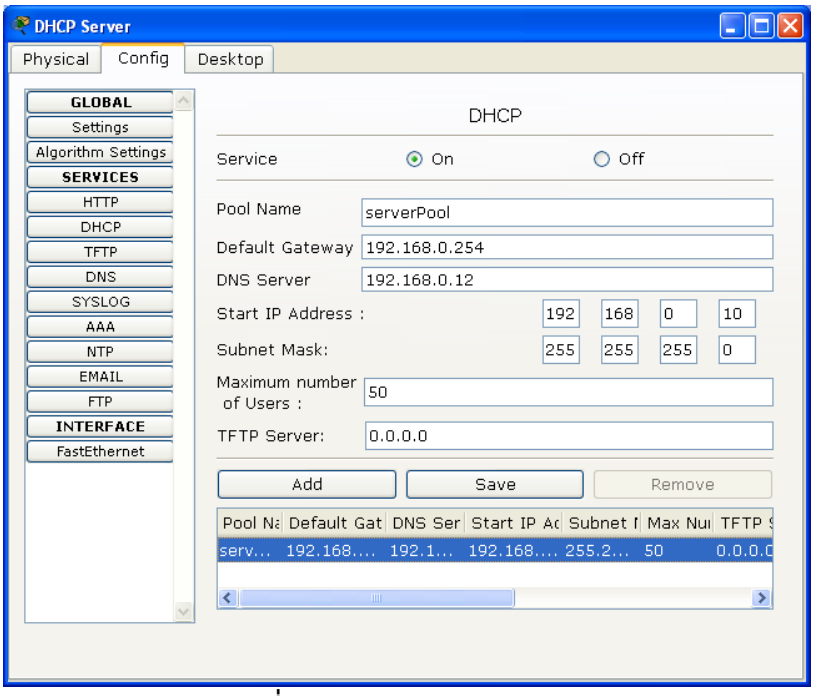
อุปกรณ์	IP Address	Subnet Mask	Interface Type
PC0	จาก DHCP	จาก DHCP	FastEthernet
PC1	จาก DHCP	จาก DHCP	FastEthernet
DNS Server	192.168.0.100	255.255.255.0	FastEthernet
Switch0	-	-	FastEthernet0/1 to PC0 FastEthernet0/2 to PC1 FastEthernet0/4 to DNS Server

ขั้นตอนการเชื่อมต่อ :

1. ห้ทำการเชื่อมต่อ PC0, PC1 เหมือนกับ Scenario 7 โดยไม่ต้องกำหนดหมายเลข IP
2. สำหรับเครื่อง DHCP Server ให้ทำการคอนฟิก IP Address โดยเลือกที่ Desktop ⇨ IP Configuration ⇨กำหนด IP ดังนี้

   IP Address = 192.168.0.100

   Subnet Mask = 255.255.255.0
3. ทำการ Enable DHCP Server โดยเลือกที่ Desktop ⇨ Config ⇨ เลือกแท็บ DHCP ⇨ ตรวจสอบ DHCP Service อยู่ในสถานะ On หรือยัง ถ้ายัง ให้เลือกเป็น On
4. ฟิลด์ Pool Name ให้ใส่ชื่อที่ต้องการ (ชื่อที่ตั้งควรสอดคล้องกับสถานที่ที่แจก IP แต่สำหรับ Packet Tracer ควรใช้ชื่อเดิมคือ ServerPool) เช่น Building A Floor Lab1, ฟิลด์ Default Gateway ใส่หมายเลข IP เกตเวย์ของเน็ตเวิร์คที่ต้องการแจกให้เครื่องลูกข่าย เช่น 192.168.0.254, DNS Server ใส่หมายเลข IP ของ DNS Server, Start IP Address ใส่ IP Address เริ่มต้นที่ต้องการแจกให้เครื่องลูกข่าย (ยกเว้น Network IP, Gateway IP, Broadcast IP) เช่น 192.168.0.10, Subnet Mask กำหนดหมายเลข Subnet Mask เช่น 255.255.255.0, Maximum Number of Users กำหนดจำนวน IP ที่ต้องการแจกให้กับเครื่องลูกข่าย เช่น 50, TFTP Server กำหนด IP ของ TFTP Server ต่อจากนั้นให้กดปุ่ม Add หรือ Save (ควรแก้ไขจาก ServerPool แล้วเลือก Save)

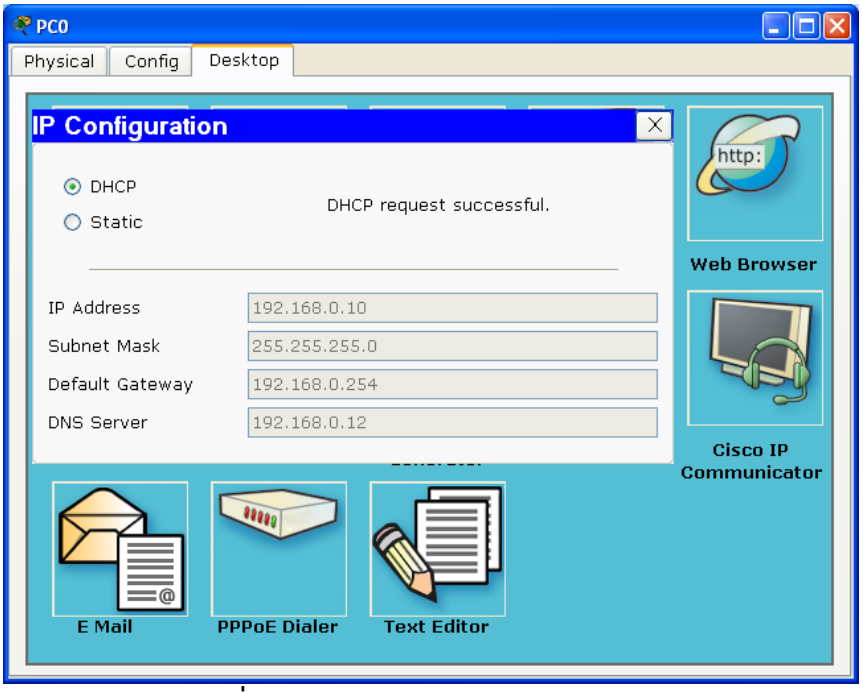


รูปที่ 12.47 การคอนฟิก DHCP

5. เสร็จขั้นตอนการเชื่อมต่อ

การทดสอบ :

1. ให้ทำการทดสอบ DHCP เซิร์ฟเวอร์ โดยการคอนฟิกให้เครื่องลูกข่ายรับหมายเลข IP Address จากเครื่อง DCHP Server โดยการคลิกที่ PC0 ⇨ Desktop ⇨ IP Configuration ⇨ เลือก DHCP แล้วสังเกตการเปลี่ยนแปลงในช่อง IP Address, Subnet Mask, Default Gateway เป็นต้น ดังรูปที่ 12.48 (ถ้ายังไม่มีการเปลี่ยนแปลง ให้กดสลับกันระหว่าง DHCP กับ Static)



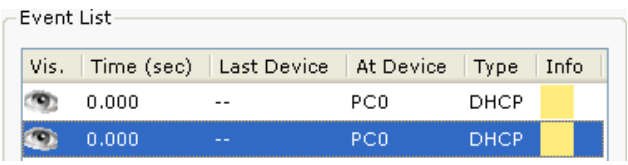
รูปที่ 12.48 การทดสอบ DHCP เซิร์ฟเวอร์

เมื่อทุกอย่างคอนฟิกถูกต้อง เครื่อง PC0 จะแสดงค่า IP Address, Subnet Mask, Default Gateway, DNS Server อย่างถูกต้อง

การวิเคราะห์แพ็คเก็ต DHCP:

1. เลือกการทำงานเป็นโหมด Simulation
2. ในแท็บ Event List Filters ให้เลือกปุ่ม Edit Filters ⇨ Show All/None แล้วเลือกแสดงผลเฉพาะ DHCP เท่านั้น
3. คลิกเลือก PC0 ⇨ Desktop ⇨ IP Configuration ⇨ เลือก DHCP แล้วกลับไปพิจารณาในแท็บ Event List อีกครั้ง
4. เริ่มต้น PC0 ทำการร้องขอไปยัง DHCP Server (IP 192.168.0.100)

DHCP Discovery



ข้อมูลใน Ethernet เฟรม

DEST MAC=FFF.FFF.FFF.FFF (กระจายไปทุกๆ เครื่อง)

SRC MAC=000A.4193.A5E1 (เครื่อง PC0)

Ethernet II

0	4	8	14	19 Byte
PREAMBLE: 101010...1011		DEST MAC: FFFF.FFFF.FFFF	SRC MAC: 000A.4193.A5E1	
TYPE: 0x800	DATA (VARIABLE LENGTH)			FCS: 0x0

ข้อมูลใน IP แพ็คเก็ต

SRC IP: 0.0.0.0

DST IP: 255.255.255.255

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 62		
ID: 0x18			0x0	0x0	
TTL: 128		PRO: 0x11	CHKSUM		
SRC IP: 192.168.0.12					
DST IP: 255.255.255.255					
OPT: 0x0					0x0
DATA (VARIABLE LENGTH)					

ข้อมูลใน UDP แพ็คเก็ต

SRC PORT: 68

DEST PORT: 67

UDP

0	16	31 Bits
SRC PORT: 68	DEST PORT: 67	
LENGTH: 0x2a	CHECKSUM: 0x0	
DATA (VARIABLE)		

ข้อมูลใน DCHP แพ็คเก็ต

OP: 0x1

"YOUR" CLIENT ADDRESS: 0.0.0.0

SERVER ADDRESS: 0.0.0.0

CLIENT HARDWARE ADDRESS: 000A.4193.A5E1

DHCP

0	8	16	31 Bits
OP: 0x7	HW TYPE	HW LEN	HOPS
TRANSACTION ID (4 BYTES)			
SECS		FLAGS	
CLIENT ADDRESS: 0.0.0.0			
"YOUR" CLIENT ADDRESS: 0.0.0.0			
SERVER ADDRESS: 0.0.0.0			
RELAY AGENT ADDRESS			
CLIENT HARDWARE ADDRESS: 000A.4193.A5E1			
SERVER HOSTNAME (64 BYTES)			
FILE (128 BYTES)			
OPTIONS (312 BYTES)			

ในขั้นตอนของ DHCP Discovery โพรโทคอล DHCP จะใช้ UDP ในการส่งข้อมูล ใช้หมายเลขพอร์ตต้นทางคือ 68, พอร์ตปลายทางคือ 67, IP ต้นทาง (SRC=0.0.0.0) เพราะยังไม่ได้รับจัดสรร IP มาให้ เริ่มต้นจึงมีค่าเป็น 0.0.0.0 ก่อนเสมอ, IP ปลายทาง คือเครื่อง DHCP Server จะเป็น 255.255.255.255 ซึ่งเป็นการ Broadcast ซึ่งกระจายไปทั่วเครือข่าย และ Option จะถูกกำหนดเป็น 0x1

5. ขั้นตอนของ DHCP Offer แพ็คเก็ตของ DHCP จะถูกส่งจาก DHCP Server กลับไปยังเครื่อง PC0 ผ่าน Switch0

   DHCP Offer

   ข้อมูลใน Ethernet เฟรม

   DEST MAC=FFF.FFF.FFF.FFF (กระจายไปทุกๆ เครื่อง)

   SRC MAC= 00E0.F978.6CE3 (เครื่อง DHCP Server)

   ข้อมูลใน IP แพ็คเก็ต

   SRC IP: 192.168.0.100

   DST IP: 255.255.255.255

   ข้อมูลใน UDP แพ็คเก็ต

   SRC PORT: 67

   DEST PORT: 68

   ข้อมูลใน DCHP แพ็คเก็ต

   OP: 0x2 (DHCP Offer)

   "YOUR" CLIENT ADDRESS: 192.168.0.12 (DHCP แจกให้ client)

   SERVER ADDRESS: 192.168.0.100 (เครื่อง DHCP Server)

   CLIENT HARDWARE ADDRESS: 000A.4193.A5E1

   ในขั้นตอนของ DHCP Offer โพรโทคอล DHCP จะใช้หมายเลขพอร์ตต้นทางคือ 67, พอร์ตปลายทางคือ 68, IP ต้นทาง SRC=192.168.0.100, IP ปลายทางคือ 255.255.255.255 และ Option จะถูกกำหนดเป็น 0x2

6. ขั้นตอนของ DHCP Request แพ็คเก็ตของ DHCP จะถูกส่งจาก PC0 กลับไปยังเครื่อง DHCP Server อีกครั้ง ผ่าน Switch0

   DHCP Request

   ข้อมูลใน Ethernet เฟรม

   DEST MAC=FFF.FFF.FFF.FFF

   SRC MAC= 00A.4193.A5E1

   ข้อมูลใน IP แพ็คเก็ต

   SRC IP: 0.0.0.0

DST IP: 255.255.255.255

ข้อมูลใน UDP แพ็คเก็ต

SRC PORT: 68

DEST PORT: 67

ข้อมูลใน DCHP แพ็คเก็ต

OP: 0x3

"YOUR" CLIENT ADDRESS: 192.168.0.12

SERVER ADDRESS: 192.168.0.100

CLIENT HARDWARE ADDRESS: 000A.4193.A5E1

7. ขั้นตอนสุดท้ายคือ DHCP ACK แพ็คเก็ตของ DHCP จะถูกส่งจาก DHCP Server กลับไปยังเครื่อง PC0 อีกครั้ง ผ่าน Switch0

DHCP ACK

ข้อมูลใน Ethernet เฟรม

DEST MAC=FFF.FFF.FFF.FFF

SRC MAC= 00E0.F978.6CE3

ข้อมูลใน IP แพ็คเก็ต

SRC IP: 192.168.0.100

DST IP: 255.255.255.255

ข้อมูลใน UDP แพ็คเก็ต

SRC PORT: 67

DEST PORT: 68

ข้อมูลใน DCHP แพ็คเก็ต

OP: 0x5

"YOUR" CLIENT ADDRESS: 192.168.0.12

SERVER ADDRESS: 192.168.0.100

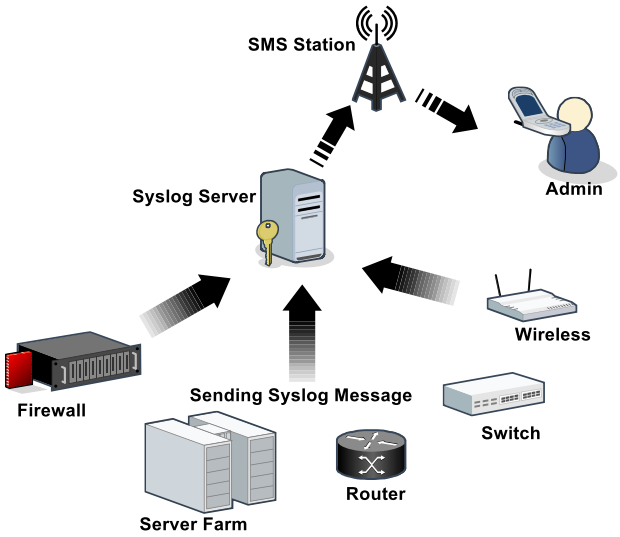
CLIENT HARDWARE ADDRESS: 000A.4193.A5E1

Scenario 10: การติดตั้ง SYSLOG เซิร์ฟเวอร์

คำอธิบาย :

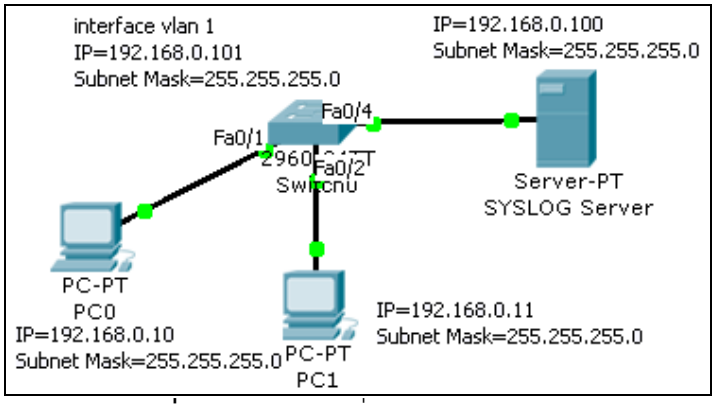
SYSLOG หรืออาจจะเรียกเป็น Syslog Daemon เป็นซอฟต์แวร์ที่ทำหน้าที่ รับ, บันทึกไฟล์ logs, แสดงผลไฟล์ logs และส่งข้อมูลการทำงาน เรียกว่า Syslog message จากเครื่องให้บริการ

เช่น เราเตอร์, สวิตช์, เซิร์ฟเวอร์, โฮสต์ และอุปกรณ์อื่นๆ ที่มีการ enable โปรแกรม Syslog ไว้ Syslog ยังมีความสามารถอื่น ๆ อีก เช่น ส่งเสียงเตือน, ส่ง email message เมื่อมีเหตุการณ์ผิดปกติเกิดขึ้น เป็นต้น ดังรูปที่ 12.49



รูปที่ 12.49 Syslog server

แผนผังการเชื่อมต่อ :



รูปที่ 12.50 ผังการเชื่อมต่อ scenario 10

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	Interface Type
PC0	192.168.0.10	255.255.255.0	FastEthernet
PC1	192.168.0.11	255.255.255.0	FastEthernet
SYSLOG Server	192.168.0.100	255.255.255.0	FastEthernet
Switch0	192.168.0.101	255.255.255.0	FastEthernet0/1 to PC0 FastEthernet0/2 to PC1 FastEthernet0/4 to SYSLOG Server

ขั้นตอนการเชื่อมต่อ :

1. ให้ทำการเชื่อมต่อ PC0, PC1 ตามรูป พร้อมกำหนด IP ตามตารางด้านบน
2. สำหรับเครื่อง SYSLOG Server ให้ทำการคอนฟิก IP Address โดยเลือกที่ Desktop ⇨ IP Configuration ⇨กำหนด IP ดังนี้

   IP Address = 192.168.0.100

   Subnet Mask = 255.255.255.0
3. ทำการ Enable SYSLOG Server โดยเลือกที่ Desktop ⇨ Config ⇨ เลือกแท็บ SYSLOG ⇨ ตรวจสอบ SYSLOG Service อยู่ในสถานะ On หรือยัง ถ้ายัง ให้เลือกเป็น On
4. กำหนด IP Address ให้กับ Switch0 เนื่องจากการใช้ SYSLOG จำเป็นต้องระบุ IP ในการเชื่อมต่อ สามารถทำได้ดังนี้ คลิกที่ Switch0 ⇨ CLI ⇨ ให้กด Enter 1-2 ครั้ง เพื่อให้เข้าสู่โหมดผู้ใช้ทั่วไป โดยแสดงเป็น prompt คือ Switch> ⇨ ให้คีย์คำสั่ง enable แล้วกดปุ่ม Enter

```
Switch>
Switch>enable <ENTER>
Switch#configuration terminal <ENTER>
Switch(config)#interface vlan 1 <ENTER>
Switch(config-if)#ip address 192.168.0.101 255.255.255.0 <ENTER>
Switch(config-if)#^Z   (กดปุ่ม CTRL พร้อมกับอักษร z)
Switch#
```

5. เสร็จขั้นตอนการเชื่อมต่อ

การทดสอบ :

1. ให้ทำการทดสอบ SYSLOG เซิร์ฟเวอร์ โดยการสั่งบันทึก log จาก Switch0 ไปเก็บไว้ใน SYSLOG Server โดยใช้คำสั่ง ดังนี้

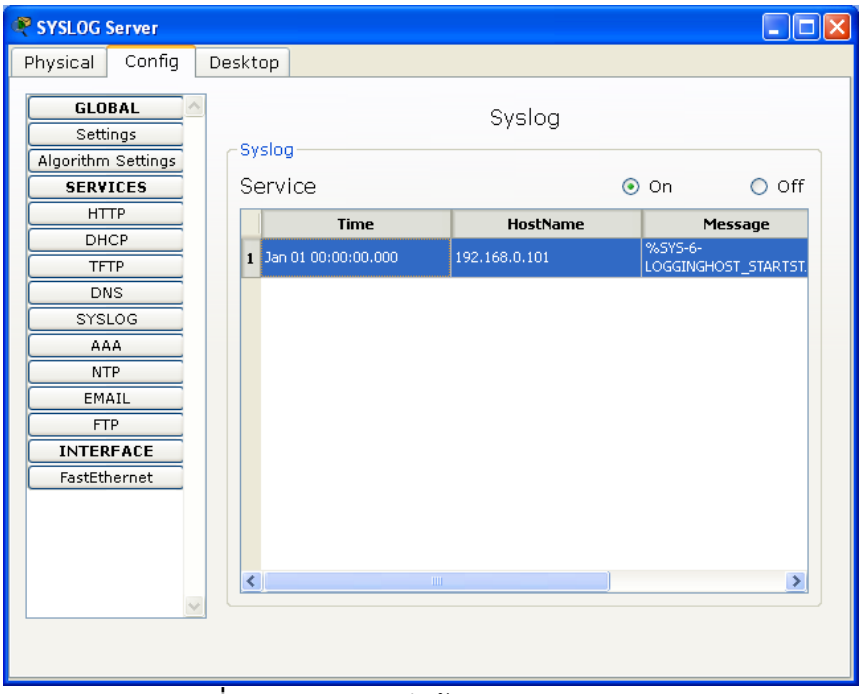
```
Switch>
Switch>enable <ENTER> ❶
Switch#configure terminal <ENTER> ❷
Switch(config)#logging 192.168.0.100 <ENTER> ❸
Switch(config)#%SYS-6-LOGGINGHOST_STARTSTOP: Logging to host
192.168.0.100 port 514 started - CLI initiated
```

ขั้นตอนการงานของ SYSLOG ส่งข้อมูล log จาก Switch0 ไปเก็บยัง SYSLOG Server

❶ เข้าสู่โหมดผู้ดูแลระบบ

❷ เข้าสู่โหมดการคอนฟิกอุปกรณ์

❸ สั่งให้บันทึกข้อมูล log ไฟล์การทำงานของ Switch0 ไปเก็บยังเครื่อง SYSLOG Server (SYSLOG คือ IP 192.168.0.100) ดังรูปที่ 12.51



รูปที่ 12.51 การบันทึกข้อมูล log ของ syslog

บนเครื่อง SYSLOG Server จะเริ่มทำการบันทึก log การทำงานของ Switch0

Scenario 11: การติดตั้ง AAA/TACACS เซิร์ฟเวอร์

คำอธิบาย :

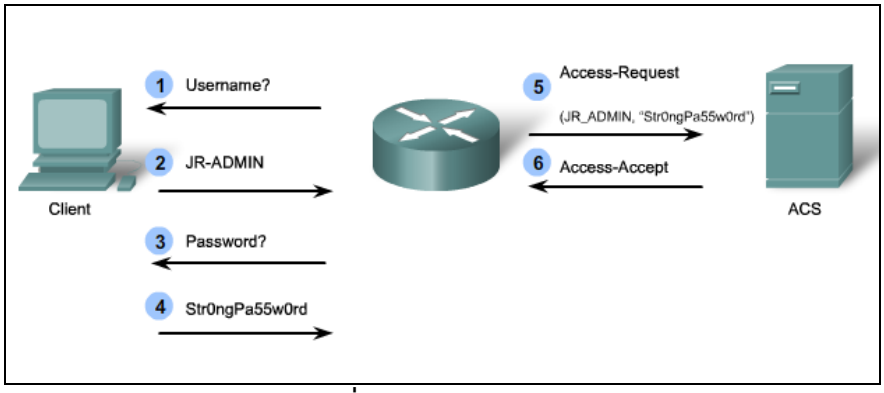
AAA เป็นมาตรฐานของการรักษาความปลอดภัยของข้อมูล (IEEE 802.1X) เช่น การจัดการ Account, การตรวจสอบสิทธิ เป็นต้น

AAA Server ย่อมาจาก 3 คำ คือ Authenticate, Authorization และ Accounting server เป็นการเพิ่มความปลอดภัยในการใช้งานแบบ Remote-Access ซึ่งเมื่อมีการเชื่อมต่อเข้ามาจะต้องถูกตรวจสอบด้วย AAA Server ก่อน ซึ่งจะตรวจสอบข้อมูลดังนี้ คือ

1. คุณเป็นใคร Who you are (authentication การยืนยัน, ระบุตัวตน - เป็นใคร?)
2. คุณได้รับอนุญาตให้ทำอะไรบ้าง What you are allowed to do (authorization การมอบสิทธิใช้งาน - คนๆนี้ มีสิทธิแค่ไหน อ่าน/เขียน/ประมวลผล)
3. คุณทำอะไรไปบ้าง What you actually do (accounting การทำบัญชีผู้ใช้ - คนๆนี้เข้ามาทำอะไรบ้าง)

เทคนิคที่นิยมใช้งานมี 2 แบบคือ RADIUS และ TACACS+ (Terminal Access Controller Access Control System) ซึ่งแต่ละวิธีมีขั้นตอนการทำงานที่แตกต่างกันดังรูปที่ 12.52

ขั้นตอนการทำงานของ RADIUS



รูปที่ 12.52 การทำงานของ RADIUS

❶ เมื่อผู้ใช้เชื่อมต่อผ่าน Remote-Access จะถูกสอบถาม User Name

❷ ผู้ใช้กรอก User Name

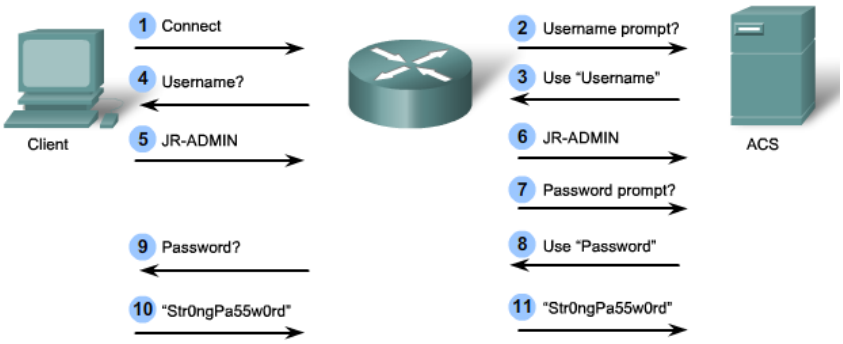
❸ ถามรหัสผ่าน

❹ ใส่รหัสผ่านของผู้ใช้

❺ อุปกรณ์ทำการส่งข้อมูล User Name และรหัสผ่านไปสอบถามยัง Access Control System (ACS) เพื่อตรวจสอบว่าข้อมูลถูกต้องหรือไม่

❻ อนุญาต เมื่อ User Name และรหัสผ่านถูกต้อง

ขั้นตอนการทำงานของ TACACS+



รูปที่ 12.53 การทำงานของ TACACS+

❶ ผู้ใช้ร้องขอการเชื่อมต่อ

❷ ตรวจสอบกับ ACS ว่ามีการเปิดใช้ User Name ?

❸ เมื่อระบบเปิดใช้งาน จะส่งกลับว่าเปิดใช้งาน User Name

❹ ส่งข้อมูลกลับให้ผู้ใช้งานว่าใช้งานแบบ User Name

❺ ส่ง User Name

❻ User Name ส่งให้ ACS ตรวจสอบความถูกต้อง

❼ เมื่อ User Name ถูกต้อง ตรวจสอบว่าระบบเปิดใช้ Password ?

❽ เมื่อระบบเปิดการใช้งาน Password ส่งข้อมูลกลับไปให้ผู้ใช้ว่าต่อไปให้ป้อนรหัสผ่าน

❾ ส่งข้อมูลกลับไปให้ผู้ใช้ว่าต่อไปให้ป้อนรหัสผ่าน

❿ ใส่รหัสผ่าน

❶❶ ตรวจสอบรหัสผ่าน เมื่อถูกต้องจะอนุญาตให้ใช้งาน

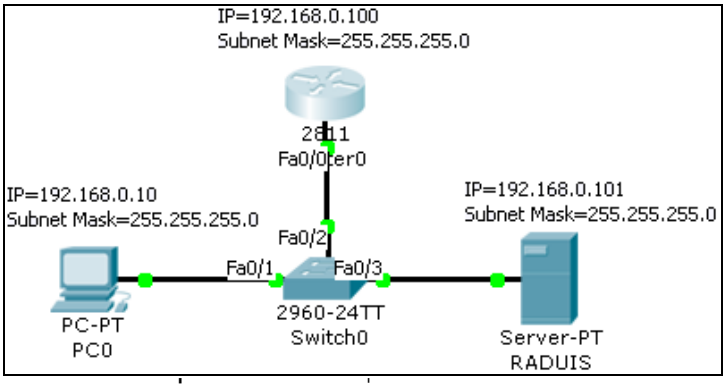
จากสองรูปด้านบน จะเห็นได้ว่า RADIUS มีการทำงานที่ซับซ้อนน้อยกว่า แต่ TACACS+ ยืดหยุ่นกว่าเพราะ TACACS+ แยกแต่ละขั้นตอนออกจากกัน ทำให้สามารถนำไปใช้กับการ Authentication ชนิดอื่นได้สะดวกกว่า แต่อย่างไรก็ตามการใช้งาน ขึ้นอยู่กับความเหมาะสม หรือสิ่งที่จะประยุกต์ใช้

สำหรับการใช้งาน AAA บนอุปกรณ์ของ Cisco แบ่งออกได้เป็นสองรูปแบบใหญ่ๆ คือ

1. Local AAA
2. Server-Based AAA

ในหัวข้อนี้จะทดสอบเฉพาะ Server-Based AAA (RADIUS) ซึ่งเป็นการนำ Server มาช่วยในเรื่องของการควบคุมและจัดการรหัสผ่านต่างๆ ของผู้ใช้ในเครือข่าย เครื่องลูกข่ายจะทำการ Authentication กับ Server โดยอาจจะมีเราเตอร์, สวิชต์ หรือ ไฟล์วอลล์ เป็นอุปกรณ์ที่ต้องการเข้าถึง

แผนผังการเชื่อมต่อ :



รูปที่ 12.54 ผังการเชื่อมต่อ scenario 11

รายการอุปกรณ์ :

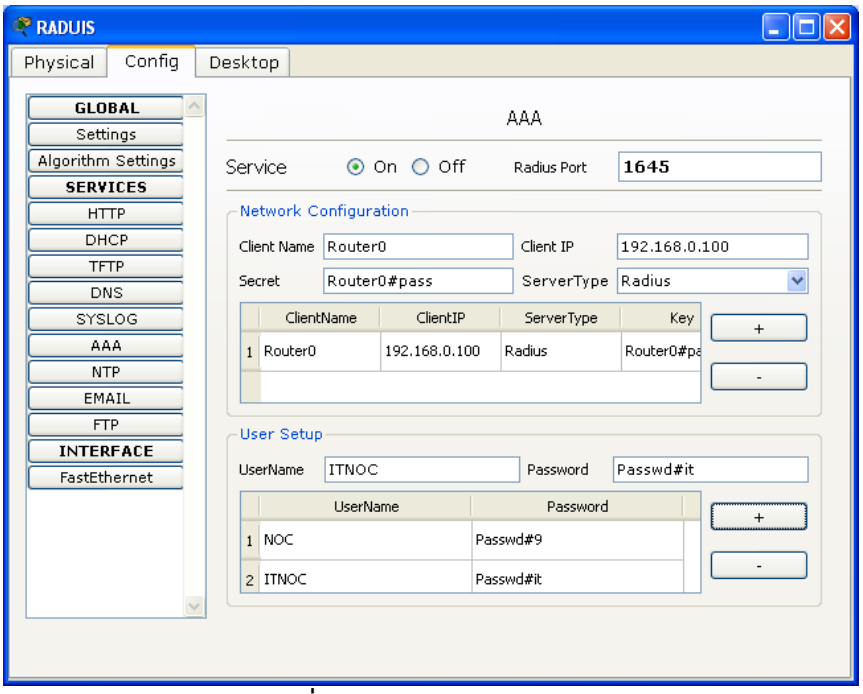
อุปกรณ์	IP Address	Subnet Mask	Interface Type
PC0	192.168.0.10	255.255.255.0	FastEthernet
Router0	192.168.0.100	255.255.255.0	FastEthernet 0/0
RADIUS Server	192.168.0.101	255.255.255.0	FastEthernet
Switch0	-	-	FastEthernet0/1 to PC0 FastEthernet0/2 to Router0 FastEthernet0/3 to RADIUS Server

ขั้นตอนการเชื่อมต่อ :

1. ให้ทำการเชื่อมต่อ PC0 ตามรูป พร้อมกำหนด IP ตามตารางด้านบน
2. สำหรับเครื่อง RADIUS Server ให้ทำการคอนฟิก IP Address โดยเลือกที่ Desktop ⇨ IP Configuration ⇨กำหนด IP ดังนี้

   IP Address = 192.168.0.101

   Subnet Mask = 255.255.255.0
3. ทำการ Enable RADIUS Server โดยเลือกที่ Desktop ⇨ Config ⇨ เลือกแท็บ RADIUS ⇨ ตรวจสอบ RADIUS Service อยู่ในสถานะ On หรือยัง ถ้ายัง ให้เลือกเป็น On
4. ในส่วน Network Configuration ฟิลด์ Client Name ให้ใส่ชื่ออุปกรณ์ที่ต้องการ Authentication เช่น Router0, Client IP ให้ใส่ IP อุปกรณ์ที่ต้องการ Authen ให้กำหนดเป็น 192.168.0.100, Secret ให้ใส่รหัสผ่าน กำหนดเป็น Router0#pass, ServerType เลือกเป็น RADIUS แล้วคลิกเลือก + เพื่อเพิ่มข้อมูลที่ป้อนเข้าไปเก็บลงฐานข้อมูลของ RADIUS สำหรับข้อมูลในส่วน Network Configuration เป็นเหมือน Key ที่ใช้สำหรับผูกความสัมพันธ์ระหว่างชื่อผู้ใช้ เข้ากับอุปกรณ์เท่านั้น
5. ในส่วน User Setup ให้ทำการกำหนด User Name และรหัสผ่านที่ใช้ในการ Login จริงๆ ในฟิลด์ UserName ใส่ชื่อที่ต้องการ login ในที่นี้ใส่ NOC, Password ใส่ Passwd#9 แล้วคลิก + เพื่อเพิ่มรายชื่อ ในส่วนนี้สามารถเพิ่มรายชื่อผู้ใช้กี่คนก็ได้ ดังรูปที่ 12.55



รูปที่ 12.55 การคอนฟิก AAA

6. บนเครื่อง Router0 ให้ดับเบิ้ลคลิกที่ตัว Router0 ⇨ แท็บ CLI ⇨ ถ้าขึ้นข้อความ --- System Configuration Dialog --- ให้เลือก No แล้วกด Enter ⇨ จะเข้าสู่โหมดผู้ใช้ทั่วไป จากนั้นให้ทำคำสั่งต่อไปนี้ เพื่อคอนฟิกอินเตอร์เฟส และ Enable AAA บน Router0

การคอนฟิก AAA และ Telnet

```
Router>
Router>enable <ENTER> ❶
Router#configure terminal <ENTER> ❷
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#aaa new-model <ENTER> ❸
Router(config)#radius-server host 192.168.0.101 key Router0#pass
<ENTER> ❹
Router(config)#aaa authentication login default group radius local
<ENTER>❺
Router(config)#line vty 0 4 <ENTER> ❻
Router(config-line)#login authentication default <ENTER> ❼
Router(config-line)#exit <ENTER> ❽
Router(config)#exit <ENTER>
Router#write memory <ENTER> ❾
```

❶ เข้าสู่โหมดผู้ดูแลระบบ

❷ เข้าสู่โหมดการคอนฟิก

❸ enable AAA authentication

❹ กำหนด IP ของเครื่อง Radius และ Key (shared secret) ที่ต้องการใช้จับคู่กับ User Name

❺ เลือกวิธีการ authentication ในที่นี้เลือก Radius โดย login โดยใช้ค่า default

❻ enable การ login ให้สามารถใช้ telnet ได้

❼ เลือกวิธี login ด้วย Radius

❽ ออกจากโหมดคอนฟิก

❾ บันทึกคอนฟิกกูเรชันลงบนเครื่องเราเตอร์

การคอนฟิก IP Address ที่อินเทอร์เฟส FastEthernet 0/0

```
Router>
Router>enable <ENTER> ❶
Router#configure terminal <ENTER> ❷
Router(config)#interface fastEthernet 0/0 <ENTER> ❸
Router(config-if)#ip address 192.168.0.100 255.255.255.0 <ENTER> ❹
Router(config-if)#no shutdown <ENTER> ❺
Router#write memory <ENTER> ❻
```

❶ เข้าสู่โหมดผู้ดูแลระบบ

❷ เข้าสู่โหมดการคอนฟิก

❸ เข้าโหมดการคอนฟิกอินเทอร์เฟส FastEthernet 0/0

❹ กำหนด IP Address ของอินเทอร์เฟส FastEthernet 0/0

❺ เปิดการใช้งานอินเทอร์เฟส FastEthernet 0/0

7.

การทดสอบ :

1. ให้ทำการทดสอบ RADUIS เซิร์ฟเวอร์ โดยมีขั้นตอนดังนี้ เลือก PC0 ⇨ Desktop ⇨ Command Prompt จากนั้นคีย์คำสั่งดังต่อไปนี้

```
PC>telnet 192.168.0.100 <ENTER> ❶
Trying 192.168.0.100 ...Open

User Access Verification
Username: NOC <ENTER> ❷
Password:#### <ENTER> ❸
Router> ❹
```

ขั้นตอนการทดสอบ RADIUS

❶ ใช้คำสั่ง telnet ทำการ remote login ไปยังเราเตอร์ (IP 192.168.0.100)

❷ เมื่อการเชื่อมต่อสำเร็จ ระบบจะให้ผู้ใช้ใส่ User Name

❸ ใส่รหัสผ่าน

❹ เมื่อ login สำเร็จ จะสามารถเข้าสู่โหมดผู้ใช้ทั่วไปได้

Scenario 12: การติดตั้ง NTP เซิร์ฟเวอร์

คำอธิบาย :

Network Time Protocol (NTP) เป็นโพรโทคอลในระดับ Application Layer ทำหน้าที่ในการเทียบเวลาระหว่างอุปกรณ์คอมพิวเตอร์ การทำงานของโพรโทคอล NTP จะต้องอาศัยเครื่องให้บริการ (NTP Server) ที่เปิดบริการพอร์ตหมายเลข 123 ชนิด UDP สำหรับการร้องขอการเทียบเวลาจากเครื่องลูกข่าย จะอยู่ในรูปแบบลำดับชั้น ที่เรียกว่า “Clock Strata” โดยแบ่งลำดับชั้นของการเทียบเวลาดังนี้

Stratum 0 เป็นอุปกรณ์ของแหล่งกำเนิดเวลา เช่น Atomic clocks, GPS เป็นต้น

Stratum 1 เป็นเครื่องคอมพิวเตอร์แม่ข่ายที่เชื่อมต่อกับ Stratum 0 ได้รับค่าเวลามาจาก Stratum 0 โดยตรงผ่านการเชื่อมต่อในระบบคอมพิวเตอร์ เช่น RS-232 เป็นต้น

Stratum 2 เป็นเครื่องคอมพิวเตอร์ที่ร้องขอการเทียบเวลาจากเครื่องคอมพิวเตอร์แม่ข่าย Stratum 1 ผ่านระบบเครือข่าย TCP/IP ด้วยการใช้งาน NTP เครื่องคอมพิวเตอร์ในระดับนี้อาจจะร้องขอการเทียบเวลาจาก Stratum 1 ได้มากกว่า 1 แหล่งเพื่อรองรับการทำงานแบบทดแทนกันเมื่อไม่สามารถเข้าถึง Stratum 1 ตัวใดตัวหนึ่งก็จะสามารถร้องขอการเทียบเวลาจาก Stratum 1 ตัวอื่นได้ต่อไป

Stratum 3 เป็นเครื่องคอมพิวเตอร์ที่ร้องขอการเทียบเวลาจากเครื่องคอมพิวเตอร์แม่ข่าย Stratum 2 ผ่านระบบเครือข่าย TCP/IP ด้วยการใช้งาน NTP เครื่องคอมพิวเตอร์ในระดับนี้จะสามารถอ้างอิง Stratum 2 ได้มากกว่า 1 แหล่ง NTP นั้นสามารถรองรับระดับของการเทียบเวลาได้ถึง 16 ระดับ

หากอุปกรณ์คอมพิวเตอร์ หรืออุปกรณ์เครือข่ายในระบบสารสนเทศมีค่าเวลาที่แตกต่างกันแล้วนั้นจะส่งผลให้เกิดปัญหากับผู้ใช้งาน รวมทั้งผู้ดูแลระบบในการปฏิบัติงานต่างๆ เช่น

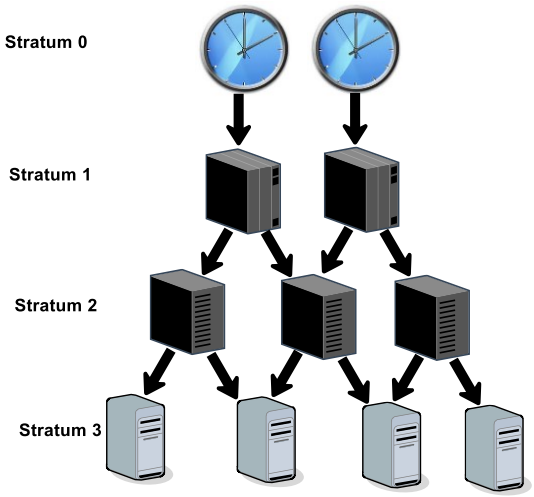
1. ความคาดเคลื่อนของเวลาในการการแจ้งปัญหาของระบบสารสนเทศ ระหว่างผู้ใช้งานและผู้ดูแลระบบ

2. ความสับสนในการตรวจสอบ และวิเคราะห์เหตุการณ์ต่างๆ เช่น เหตุการณ์การบุกรุก เหตุการณ์ของปัญหาด้านเครือข่าย หรือระบบคอมพิวเตอร์

3. ผู้พัฒนามีความสับสนในเวอร์ชันของโค้ดระหว่างการพัฒนา

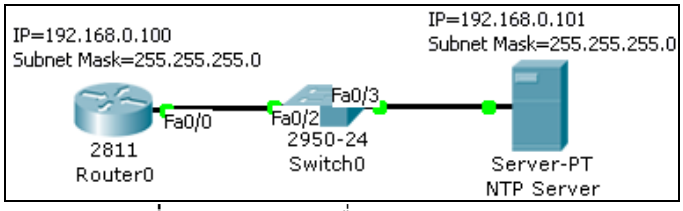
4. มีการใช้งานไฟล์ข้อมูล หรือฐานข้อมูล ที่ซ้อนทับกัน

หมายเหตุ: ข้อมูล NTP อ้างอิงจาก http://netco.ku.ac.th/law/ntp.htm



รูปที่ 12.56 ลำดับชั้นของการเทียบเวลาใน NTP

แผนผังการเชื่อมต่อ :



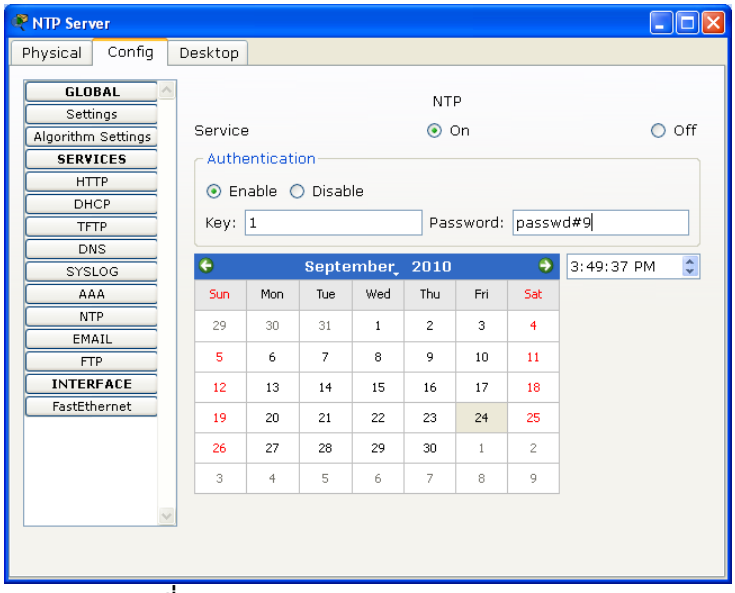
รูปที่ 12.57 ผังการเชื่อมต่อ scenario 12

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	Interface Type
Router0	192.168.0.100	255.255.255.0	FastEthernet 0/0
NTP Server	192.168.0.101	255.255.255.0	FastEthernet
Switch0	-	-	FastEthernet0/2 to Router0 FastEthernet0/3 to NTP Server

ขั้นตอนการเชื่อมต่อ :

1. ให้ทำการเชื่อมต่อและกำหนด IP Address ของ Router0, NTP Server, Switch0 ตามรูป พร้อมกำหนด IP ตามตารางด้านบน
2. ทำการ Enable NTP Server โดยเลือกที่ Desktop ⇨ Config ⇨ เลือกแท็บ NTP ⇨ ตรวจสอบ NTP Service อยู่ในสถานะ On หรือยัง ถ้ายัง ให้เลือกเป็น On
3. ในส่วน Authentication ให้เลือกเป็น Enable ในฟิลด์ Key ให้ใส่ Key ที่ต้องการ Authentication, ในฟิลด์ Password ให้ใส่รหัสผ่านที่ต้องการ Authen, จากนั้นให้เลือกวันเวลาที่ต้องการให้อุปกรณ์เข้ามาเทียบเวลา ซึ่งใน Packet Tracer จะดึงเวลาจากเครื่องที่ทำงานอยู่โดยอัตโนมัติ แต่ในการทำงานจริง NTP Server จะต้องเชื่อมต่อกับเวลาของ Stratum Server อีกที ดังรูปที่ รูปที่ 12.58



รูปที่ 12.58 แสดงการคอนฟิก NTP Server

4. ขั้นตอนต่อไปจะเป็นการคอนฟิกให้เราเตอร์หรือเชื่อมต่อกับ NTP Server เพื่อเทียบเวลา ในเบื้องต้นให้ตรวจสอบ Interface FastEthernet 0/0 ว่ามีหมายเลข IP Address และสามารถ ping เครื่อง NTP Server ได้หรือไม่ ถ้ายังให้กลับไปทำขั้นตอนการเชื่อมต่อก่อน (โดยดูตัวอย่างจาก Scenario 12) ขั้นต่อไปให้ทำการ Enable NTP Server โดยใช้คำสั่งต่อไปนี้บน Router0

```
Router>
Router>enable <ENTER> ❶
Router#show clock <ENTER> ❷
*3:59:22.756 UTC Mon Mar 1 1993
Router#configure terminal <ENTER> ❸
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#ntp server 192.168.0.101 key 1 <ENTER> ❹
Router(config)# ^Z <ENTER> ❺
Router#sh clock <ENTER>
*16:3:23.469 UTC Fri Sep 24 2010
```

❶ เข้าสู่โหมดผู้ดูแลระบบ

❷ แสดงเวลาของ Router0 เดิม ในที่นี้คือ 3:59:22.756 UTC Mon Mar 1 1993

❸ เข้าโหมดโกบอลคอนฟิก

❹ เปิดการทำงานของ NTP บนเราเตอร์ ให้ทำการระบุเครื่อง NTP Server และ Key

❺ กดปุ่ม CTRL พร้อม z เพื่อออกไปสู่โหมดผู้ดูแลระบบ

❻ ทดสอบเวลาของ Router0 อีกครั้ง ปรากฏว่าเวลาจะเปลี่ยนตาม NTP Server คือ 16:3:23.469 UTC Fri Sep 24 2010

การทดสอบ :

1. ให้ทำการทดสอบ NTP เซิร์ฟเวอร์ โดยใช้คำสั่ง show clock บนเราเตอร์ดังต่อไปนี้

```
Router#sh clock <ENTER>
*16:3:23.469 UTC Fri Sep 24 2010
```

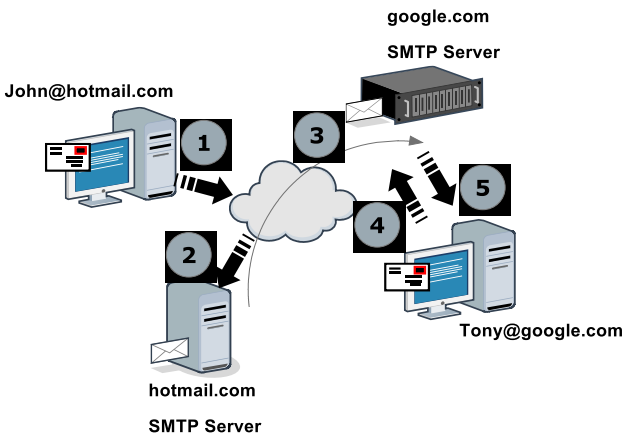
Scenario 13: การติดตั้ง EMAIL เซิร์ฟเวอร์ (SMTP/POP3)

คำอธิบาย :

E-mail คือ จดหมายอิเล็กทรอนิกส์ หรือ ไปรษณีย์อิเล็กทรอนิกส์ (electronic mail, ย่อ e-mail หรือ email) เป็นการส่งข้อความจากบุคคลหนึ่งไปยังอีกบุคคลหนึ่ง ที่ใช้รับส่งกันโดยผ่านเครือข่ายคอมพิวเตอร์ E-mail จำเป็นต้องมีระบบการกำหนดชื่อที่อยู่ (e-mail address เช่น suchart.k@msu.ac.th ซึ่งประกอบด้วยชื่อบัญชี (suchart.k) ตามด้วยเครื่องหมาย @ และปิดท้ายด้วยชื่อโฮสต์ ชื่อองค์กร หรือ domain name ที่ลงทะเบียนไว้ การส่งจดหมายอิเล็กทรอนิกส์เป็นวิธีการส่งเหมือนจดหมายจริง โดยจะไปเก็บไว้ในเมล์บ็อกซ์ของผู้รับปลายทาง รอจนกว่าผู้รับปลายทางจะมาเปิดเมล์บ็อกซ์นำจดหมายไป e-mail ประกอบไปด้วย 2 ส่วนคือ E-mail Server คือ

1. คอมพิวเตอร์ที่ทำหน้าที่ให้บริการด้านจดหมายอิเล็กทรอนิคส์ หรือ SMTP Server (Mail Server) และโพรโทคอลสำหรับการเข้าถึงจดหมายที่อยู่ใน SMTP Server เช่น POP และ IMAP

2. โปรแกรมที่ใช้สำหรับอ่านจดหมาย เขียนจดหมาย ส่งจดหมาย และรับจดหมาย (Mail Client) มีอยู่หลายตัวด้วยกันยกตัวอย่าง เช่น Pine, Netscape, Outlook, Webmail เป็นต้น ดังรูปที่ 12.59



รูปที่ 12.59 หลักการทำงานของ E-Mail Server

❶ ผู้ใช้ชื่อ John (ใช้ email ของ hotmail) ต้องการส่ง email ไปหา Tony (ใช้ email ของ google) โดย John จะเปิดโปรแกรม email client เช่น outlook หรือ web mail (ใช้งานอีเมล์ผ่านเว็บ) เช่น hotmail, yahoo เป็นต้น

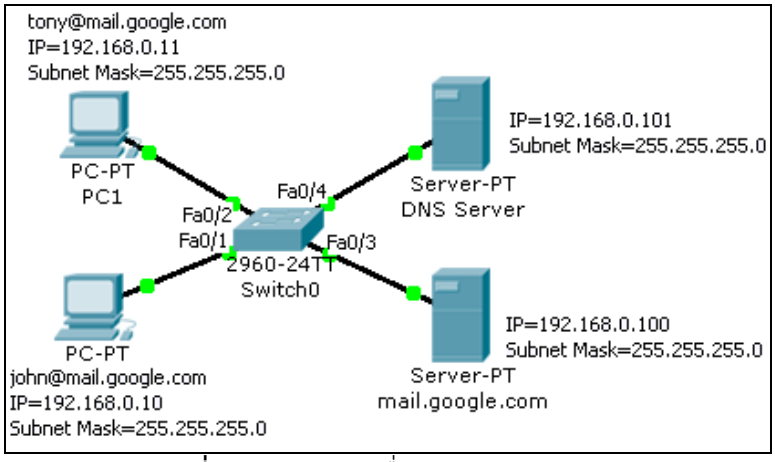
❷ เมื่อผู้ใช้เขียนเมล์เสร็จ จะทำการส่งไปยัง Mail Server ของ hotmail เพื่อให้ทำการส่งเมล์ดังกล่าวไปยัง email ปลายทางซึ่งอยู่ที่ google.com

❸ Server hotmail ส่งจดหมายไปเก็บไว้ยัง Server google อยู่ในกล่องจดหมายของ Tony

❹ Tony เข้ามาเปิดอ่านจดหมายในกล่องจดหมายของตนเอง

❺ Tony อ่านจดหมาย ในทางกลับกันถ้า Tony ต้องการส่งจดหมายบ้างก็จะมีหลักการทำงานที่เหมือนกัน

แผนผังการเชื่อมต่อ : ดังรูปที่ 12.60



รูปที่ 12.60 ผังการเชื่อมต่อ scenario 13

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	Function
PC0 (john)	192.168.0.10	255.255.255.0	john@mail.google.com
PC1 (tony)	192.168.0.11	255.255.255.0	tony@mail.google.com
MAIL Server	192.168.0.100	255.255.255.0	mail.google.com
DNS Server	192.168.0.101	255.255.255.0	Domain Name Server
Switch0	-	-	-

ขั้นตอนการเชื่อมต่อ :

1. ให้ทำการเชื่อมต่ออุปกรณ์ต่างๆ ดังผังเครือข่ายด้านบน และคอนฟิกค่าต่างๆ ตามตารางด้านบน ดังนี้

   บนเครื่อง PC0 (john)

เลือก Desktop ⇨ IP Configuration

IP Address = 192.168.0.10

Subnet Mask = 255.255.255.0

DNS Server = 192.168.0.101

เลือก Desktop ⇨ E Mail ⇨ Configure Mail

Your Name = john

Email Address = john@mail.google.com

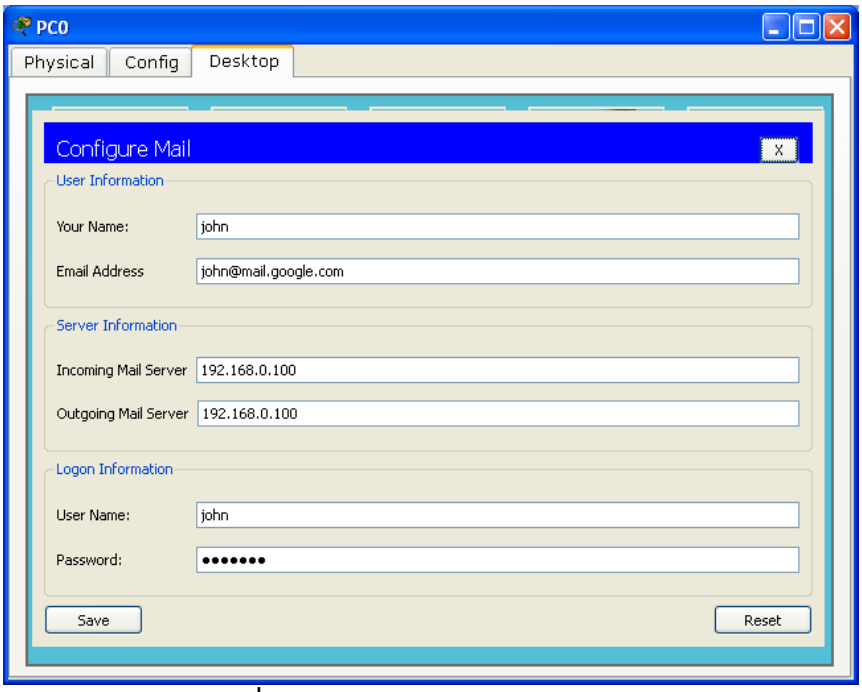
Incoming Mail Server = 192.168.0.100

Outgoing Mail Server = 192.168.0.200

User Name = john (ชื่อผู้ใช้ที่ใช้ติดต่อกับ Mail Server)

Password = john123 (รหัสผ่านผู้ใช้ที่ใช้ติดต่อกับ Mail Server)

เมื่อกรอกข้อมูลครบแล้วเลือก Save ดังรูปที่ 12.71



รูปที่ 12.61 แสดงการ Configure Mail

บนเครื่อง PC1 (tony)

เลือก Desktop ⇨ IP Configuration

IP Address = 192.168.0.11

Subnet Mask = 255.255.255.0

DNS Server = 192.168.0.101

เลือก Desktop ⇨ E Mail ⇨ Configure Mail

Your Name = tony

Email Address = tony@mail.google.com

Incoming Mail Server = 192.168.0.100

Outgoing Mail Server = 192.168.0.200

User Name = tony

Password = tony123

เมื่อกรอกข้อมูลครบแล้วเลือก Save

บนเครื่อง Mail Server

เลือก Desktop ⇨ IP Configuration

IP Address = 192.168.0.100

Subnet Mask = 255.255.255.0

เลือก Config ⇨ MAIL ⇨ เพิ่มรายชื่อผู้ใช้ดังต่อไปนี้

Domain Name = mail.google.com กรอกเสร็จแล้วเลือก Set

เพิ่มผู้ใช้ชื่อ john

User = john

Password = john123

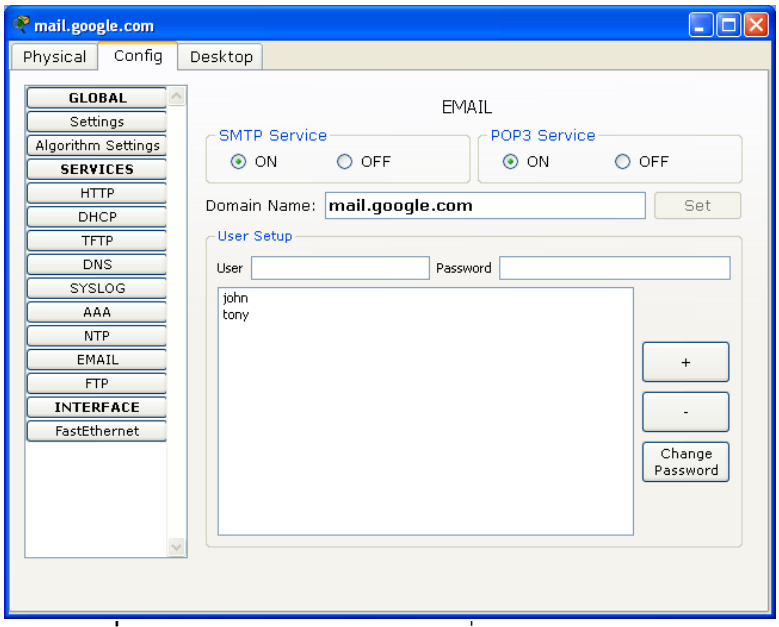
เมื่อกรอกข้อมูลครบแล้วเลือก + เพื่อเพิ่มรายชื่อ (เป็นการลงทะเบียนขอใช้ email บน Mail Server ของ google.com)

เพิ่มผู้ใช้ tony

User = tony

Password = tony123

เมื่อกรอกข้อมูลครบแล้วเลือก + เพื่อเพิ่มรายชื่อ ดังรูปที่ 12.62



รูปที่ 12.62 คอนฟิก Mail Server ชื่อ mail.google.com

บนเครื่อง DNS Server

เลือก Desktop ⇨ IP Configuration

IP Address = 192.168.0.101

Subnet Mask = 255.255.255.0

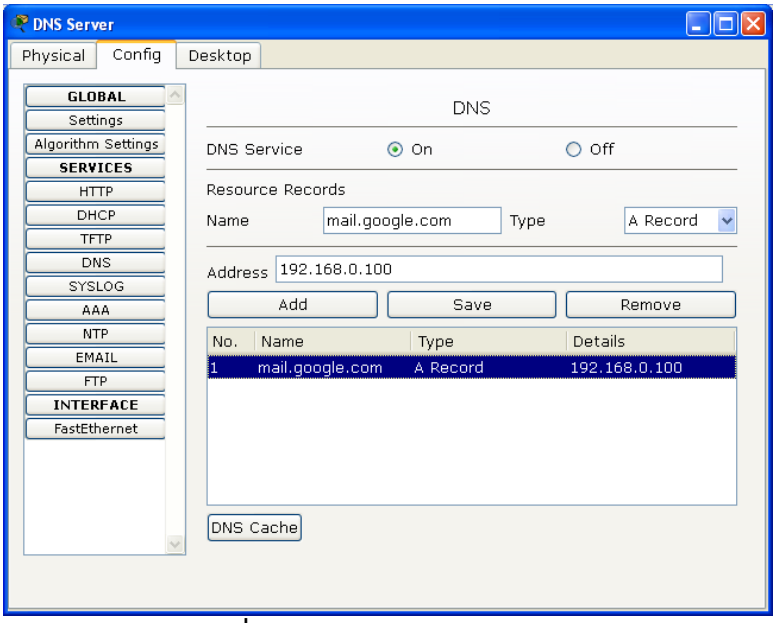
เลือก Config ⇨ DNS ⇨ เพิ่มโดเมนเนมดังต่อไปนี้

Name = mail.google.com

Type = A Record

Address = 192.168.0.100

เมื่อกรอกข้อมูลครบแล้วเลือก Save ดังรูปที่ 12.63



รูปที่ 12.63 คอนฟิก DNS Server

การทดสอบ :

1. ให้ทำการทดสอบ MAIL เซิร์ฟเวอร์ โดยการส่ง email จากผู้ใช้ชื่อ john ไปหา tony โดยมีขั้นตอนดังนี้

   เลือก PC0 ⇨ Desktop ⇨ E Mail ทำการเขียนจดหมาย เพื่อส่งไปยัง tony ⇨ เลือก Compose Mail กรอกข้อมูลดังต่อไปนี้

   To: = tony@mail.google.com

   Subject: = Hi tony.

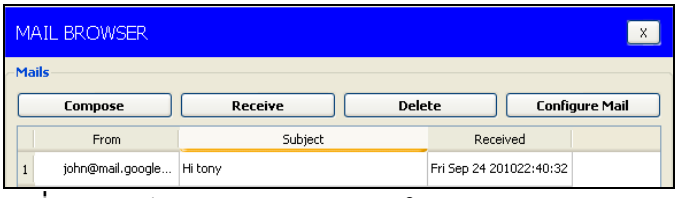
   ในช่องเขียนจดหมายให้ผู้ใช้เขียนจดหมาย เช่น

Dear Mr. Tony We wish to thank you once again for inviting us to your anniversary party, where good time was had by all. It was a successful event, and we really enjoyed ourselves.

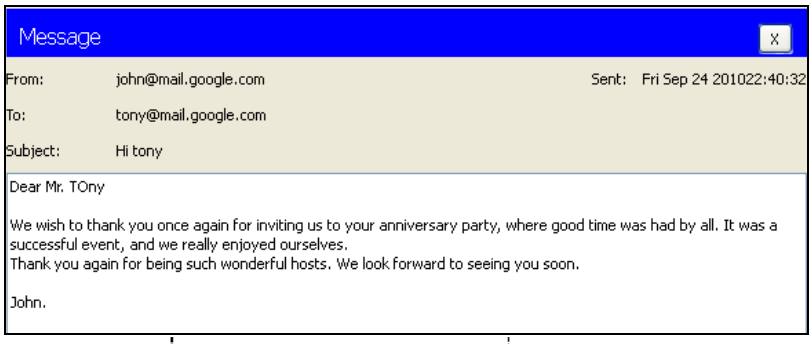
Thank you again for being such wonderful hosts. We look forward to seeing you soon. John.

เมื่อเขียนจดหมายเสร็จแล้วให้กดปุ่ม Send จดหมายจะส่งไปยัง Mail Box ของ tony

2. ที่เครื่องของ tony ให้เลือก ⇨ Desktop ⇨ E Mail เพื่อทำการอ่านเมล์ที่ส่งมาจาก john ⇨ ที่ Mail Browser ให้เลือก Receive จะปรากฏจดหมายของ john อยู่ใน Mail Box ให้ดับเบิ้ลคลิกเมล์ดังกล่าวเพื่ออ่านเมล์ ดังรูปที่ 12.64, 12.65



รูปที่ 12.64 มีจดหมายจาก john อยู่ใน Mail Box ของ Tony



รูปที่ 12.65 Tony เปิดจากหมายที่ส่งมาจาก John

3. ให้ทดลองส่งเมล์จาก tony ไปยัง john เพื่อทดสอบการทำงานของ Mail Server ว่าทำงานถูกต้อง

การสำรองข้อมูลและการคืนข้อมูล

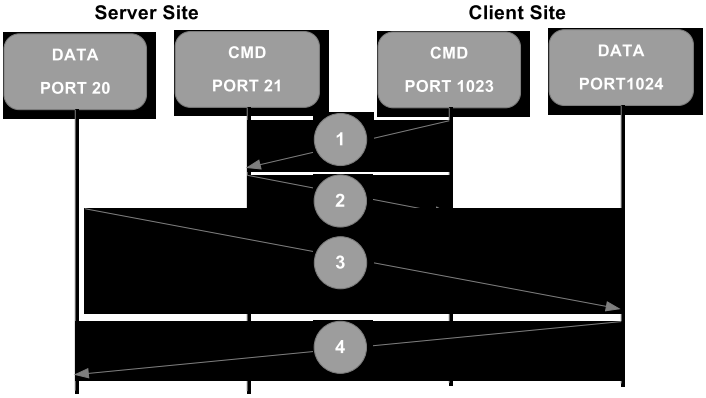
Scenario 14: การติดตั้งเอฟทีพีเซิร์ฟเวอร์ (FTP)

คำอธิบาย :

เอฟทีพี (FTP = File Transfer Protocol) คือ โปรแกรมที่ใช้สำหรับส่งแฟ้ม (Send) หรือรับแฟ้ม (Receive) ระหว่างเครื่องคอมพิวเตอร์ของผู้ใช้ (Client Computer) กับเครื่องให้บริการ (FTP Server) ผู้ให้บริการจะสร้างรหัสผู้ใช้(User Name) และรหัสผ่าน(Password) ให้ผู้ใช้แต่ละคนได้เป็นเจ้าของพื้นที่แต่ละห้อง (User Folder), FTP มีการทำงานโดยใช้พอร์ตสองพอร์ตคือ data port และ command port (หรือ control port) ซึ่งโดยทั่วไปจะใช้พอร์ตที่ 21 เป็น command port และใช้พอร์ต 20 สำหรับ data port แต่จะมีความสับสนเกิดขึ้นเมื่อ data port ไม่เป็นพอร์ต 20 ซึ่งจะขึ้นอยู่กับโหมดการทำงาน แบ่งได้เป็น 2 โหมดคือ Active FTP, Passive FTP

โหมด Active FTP

โหมดการทำงานที่เป็น Active FTP เครื่อง client จะเชื่อมต่อจากพอร์ตที่ไม่มีสิทธิพิเศษ (unprivileged port) แบบสุ่มที่มีค่าพอร์ตมากกว่า 1024 (N > 1024) ไป ยัง command port (21) ของ FTP Server จากนั้นเครื่อง client ก็จะเริ่มคอยฟัง (listening) พอร์ต N+1 และส่ง FTP command port N+1 ไปยัง FTP Server แล้วเครื่อง FTP Server ก็จะเชื่อมต่อกลับมายังเครื่อง client ตาม data port ที่ได้กำหนดไว้ โดยที่ FTP Server จะเป็น Data Port 21 ซึ่งสามารถแสดงรูปการเชื่อมต่อของ Active FTP ดังรูปที่ 12.66



รูปที่ 12.66 การทำงานของ FTP โหมด active

❶ Command port ของเครื่อง client ติดต่อ command port ของเครื่อง Server และส่ง command PORT 1023

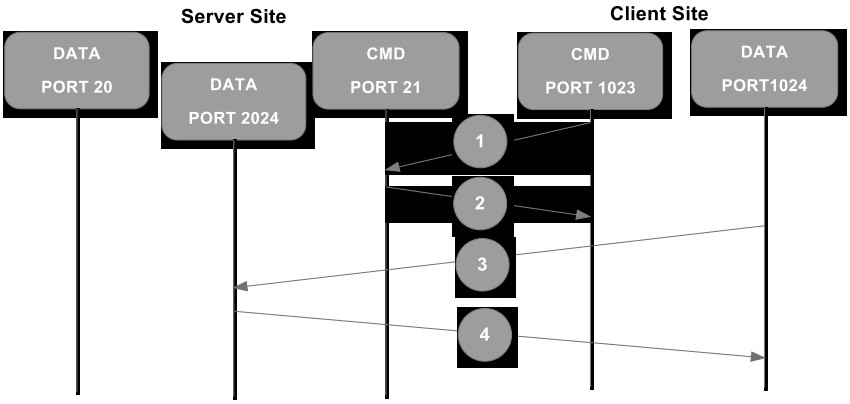
❷ Server ส่ง ACK กลับไปยัง command port ของเครื่อง client

❸ Server เริ่มต้นการเชื่อมต่อ (initiate) โดยใช้ data port ของตัวเองคือ 20 ไปยัง data port ของเครื่อง client ที่ถูกกำหนดไว้

❹ เครื่อง client ส่ง ACK กลับไปยัง Server

โหมด Passive FTP

รูปการเชื่อมต่อของ Passive FTP ดังรูปที่ 12.67



รูปที่ 12.67 การทำงานของ FTP โหมด passive

❶ client ติดต่อ Server บน command port และส่ง PASV command

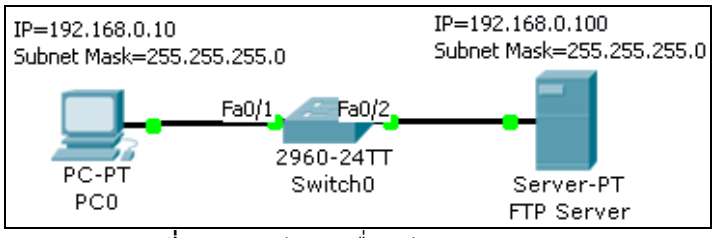
❷ Server ตอบกลับด้วยพอร์ต 2024 เพื่อบอก client ว่าพอร์ตไหนที่ Server กำลัง listening เพื่อการเชื่อมต่อ data

❸ client เริ่มต้นการเชื่อมต่อ data จาก data port ของตัวเองไปยัง data port ของ server ที่ถูกระบุ

❹ Server ส่ง ACK กลับไปยัง data port ของ client

หมายเหตุ: ข้อมูล FTP อ้างอิงจาก http://www.tkc.ac.th/osunun/

แผนผังการเชื่อมต่อ :



รูปที่ 12.68 ผังการเชื่อมต่อ scenario 15

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask
PC0	192.168.0.10	255.255.255.0
FTP Server	192.168.0.100	255.255.255.0
Switch0	-	-

ขั้นตอนการเชื่อมต่อ :

1. ให้ทำการเชื่อมต่ออุปกรณ์ต่างๆ ดังผังเครือข่ายด้านบน และคอนฟิกค่าต่างๆ ตามตารางด้านบน ดังนี้

บนเครื่อง PC0

เลือก Desktop ⇨ IP Configuration

IP Address = 192.168.0.10

Subnet Mask = 255.255.255.0

บนเครื่อง FTP Server

เลือก Desktop ⇨ IP Configuration

IP Address = 192.168.0.100

Subnet Mask = 255.255.255.0

เลือก Config ⇨ FTP ⇨ ตรวจสอบข้อมูลดังต่อไปนี้

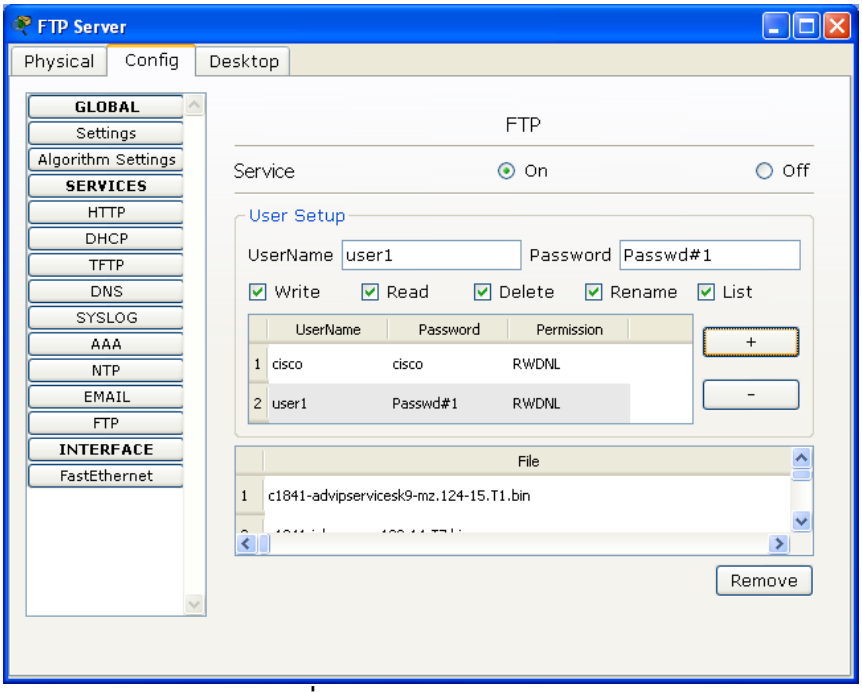
Service = on

User Name = user1 (กำหนดรายชื่อผู้ใช้งานบน FTP Server)

Password = Passwd#1 (กำหนดรหัสผ่านผู้ใช้งานสำหรับ user1)

คลิกเลือกสิทธิการใช้งานบน FTP Server โดยมีคุณสมบัติให้เลือกดังนี้

Write(เขียนได้), Read(อ่านได้), Delete(ลบได้), Rename(เปลี่ยนชื่อได้), List(แสดงรายชื่อในไดเรคทรอรีได้) เมื่อเลือกคุณสมบัติเสร็จแล้ว ให้เลือก + ดังรูปที่ 12.69



รูปที่ 12.69 กำหนดคุณสมบัติของ FTP Server

การทดสอบ :

1. ให้ทำการทดสอบ FTP เซิร์ฟเวอร์ โดยการลองโอนย้ายไฟล์จากเครื่องผู้ใช้ไปยัง FTP Server โดยมีขั้นตอนดังนี้

   เลือก PC0 ⇨ Desktop ⇨ Command Prompt ใช้คำสั่งดังต่อไปนี้

```
PC>
PC>? <ENTER> (แสดงคำสั่งทั้งหมดที่ Packet Tracer สนับสนุน)
Available Commands:
  ?        Display the list of available commands
  arp      Display the arp table
  delete   Deletes the specified file from C: directory.
  dir      Displays the list of files  in C: directory.
  ftp      Transfers files to and from a computer running an FTP server.
  help     Display the list of available commands
  ipconfig Display network configuration for each network adapter
```

```
  ipv6config   Display network configuration for each network adapter
  netstat      Displays protocol statistics and current TCP/IP network
               connections
  nslookup     DNS Lookup
  ping         Send echo messages
  snmpget      SNMP GET
  snmpgetbulk  SNMP GET BULK
  snmpset      SNMP SET
  ssh          ssh client
  telnet       Telnet client
  tracert      Trace route to destination
```

PC>**dir** <ENTER> (แสดงข้อมูลในไดเรคทรอรีปัจจุบัน สำหรับใน Packet Tracer ได้เตรียมไฟล์ไว้ให้ 1 ไฟล์คือ sampleFile.txt)

```
 Volume in drive C has no label.
 Volume Serial Number is 5E12-4AF3
 Directory of C:\
  2/7/2106    13:28 PM      26                sampleFile.txt
                            26 bytes            1 File(s)
```

PC>**ftp 192.168.0.100** <ENTER> (เชื่อมต่อไปยัง FTP Server)

Trying to connect...192.168.0.100

Connected to 192.168.0.100

220- Welcome to PT Ftp server

Username:**user1** <ENTER> (ป้อนรายชื่อผู้ใช้ที่ลงทะเบียนไว้บน FTP Server)

331- Username ok, need password

Password:**Passwd#1** <ENTER> (ป้อนรหัสผ่านที่ลงทะเบียนไว้บน FTP Server)

230- Logged in

(passive mode On) (การ login สำเร็จ และเป็นโหมด Passive)

ftp>**put sampleFile.txt** <ENTER> (โอนย้ายไฟล์จากเครื่องผู้ใช้ชื่อ sampleFile.txt ไปเก็บไว้บนเครื่อง FTP Server)

Writing file sampleFile.txt from 192.168.0.100:

File transfer in progress...

[Transfer complete - 26 bytes]

26 bytes copied in 0.141 secs (184 bytes/sec) (การโอนย้ายสำเร็จ)

ftp>**dir** <ENTER> (ตรวจสอบรายการไฟล์ข้อมูลที่อยู่บน FTP Server)

Listing /ftp directory from 192.168.0.100:

0 : c1841-advipservicesk9-mz.124-15.T1.bin 33591768

1 : c1841-ipbase-mz.123-14.T7.bin 13832032

2 : c1841-ipbasek9-mz.124-12.bin 16599160

3 : c2600-advipservicesk9-mz.124-15.T1.bin 33591768

4 : c2600-i-mz.122-28.bin 5571584

5 : c2600-ipbasek9-mz.124-8.bin 13169700

6 : c2800nm-advipservicesk9-mz.124-15.T1.bin 50938004

7 : c2800nm-ipbase-mz.123-14.T7.bin 5571584

8 : c2800nm-ipbasek9-mz.124-8.bin 15522644

9 : c2950-i6q4l2-mz.121-22.EA4.bin 3058048

10 : c2950-i6q4l2-mz.121-22.EA8.bin 3117390

11 : c2960-lanbase-mz.122-25.FX.bin 4414921

12 : c2960-lanbase-mz.122-25.SEE1.bin 4670455

13 : c3560-advipservicesk9-mz.122-37.SE1.bin 8662192

14 : pt1000-i-mz.122-28.bin 5571584

15 : pt3000-i6q4l2-mz.121-22.EA4.bin 3117390

16 : sampleFile.txt 26

ftp>**get sampleFile.txt** <ENTER> (ทดสอบการโอนย้ายข้อมูลกลับจาก FTP Server มายังเครื่องผู้ใช้)

Reading file sampleFile.txt from 192.168.0.100:

File transfer in progress...

[Transfer complete - 26 bytes]

26 bytes copied in 0.109 secs (238 bytes/sec) (การโอนย้ายสำเร็จ)

Scenario 15: การติดตั้งทีเอฟทีพีเซิร์ฟเวอร์ (TFTP)

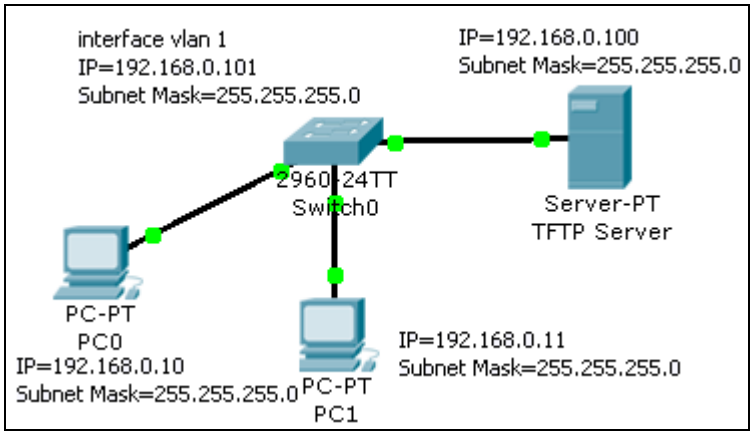
คำอธิบาย :

TFTP เป็นกระบวนการรับส่งไฟล์ที่เรียบง่ายกว่า FTP โดยใช้กลไกการสื่อสารแบบ UDP (User Datagram Protocol) ซึ่งเป็นโปรโตคอลที่ทำงานแบบ Connectionless ผู้ใช้ไม่จำเป็นต้องใส่รหัสผ่าน (Password) แต่จะต้องจัดเตรียมข้อมูลที่จะโอนย้ายไว้ก่อนเสมอ TFTP จะมีคุณสมบัติเพิ่มเติมอื่นๆ ให้ผู้ใช้ปรับแต่งได้เล็กน้อย เช่น การแสดงรายชื่อไฟล์, การเปลี่ยนไดเร็คทอรี เป็นต้น

กลไกการทำงานของ TFTP จะกำหนดขนาดของบล็อคข้อมูลที่โอนย้ายไว้ 512 ไบต์คงที่ และมีขนาดของการรับส่งข้อมูลที่โต้ตอบเป็น 512 ไบต์เช่นกัน การรับส่งข้อมูลในแต่ละบล็อค ผู้ส่งจะต้องรอให้ผู้รับยืนยันความถูกต้องของข้อมูลบล็อคที่ได้รับก่อน จึงจะสามารถส่งข้อมูลบล็อคต่อไปได้ กรณีที่ไม่มีกายืนยันความถูกต้องของข้อมูลในเวลาที่กำหนด (timeout) จะถือว่าไม่มีผู้รับข้อมูลดังกล่าว และจะต้องส่งข้อมูลหรือยืนยันความถูกต้องใหม่อีกครั้งหนึ่ง (retransmit) แต่ถ้าหากเกิดปัญหาขึ้นในระหว่างการรับส่งข้อมูลการทำงานจะถูกยกเลิกและ TFTP ก็ไม่สามารถจะรับส่งข้อมูลต่อจากส่วนเดิมได้ ใน TFTP ได้รับการพัฒนาประสิทธิภาพต่อมา ให้ผู้รับและผู้ส่งสามารถกำหนดขนาดของบล็อกได้ตั้งแต่ 8 – 64 ไบต์ กำหนดระยะเวลา Timeout ได้ตั้งแต่ 1 ถึง 255 วินาที รวมทั้งกำหนดขนาดของไฟล์ที่จะรับส่งกัน การทำงานของ TFTP จะไม่ซับซ้อน ดังนั้นโปรแกรมที่ใช้งานจะมีขนาดเล็ก ใช้เนื้อที่ในหน่วยความจำน้อย สามารถบรรจุโปรแกรมลงในชิพประเภทที่เป็น Programmable Read-Only Memory (PROM) เพื่อนำไปใช้งานในเครื่องที่ใช้พกพาหรือเครื่องขนาดเล็กได้ง่ายกว่าการใช้โปรแกรมประเภท FTP ใน Scenario นี้จะใช้ TFTP เพื่อสำรองไฟล์คอนฟิกและระบบปฏิบัติการของอุปกรณ์เครือข่าย เช่น สวิชต์, เราเตอร์ เป็นต้น

หมายเหตุ: ข้อมูล TFTP อ้างอิงจาก http://wich246.tripod.com/tftp.htm

แผนผังการเชื่อมต่อ : รูปที่ 12.70



รูปที่ 12.70 ผังการเชื่อมต่อ scenario 15

ขั้นตอนการเชื่อมต่อ :

1. ให้ทำการเชื่อมต่ออุปกรณ์ต่างๆ ดังผังเครือข่ายด้านบน และคอนฟิกค่าต่างๆ ดังนี้

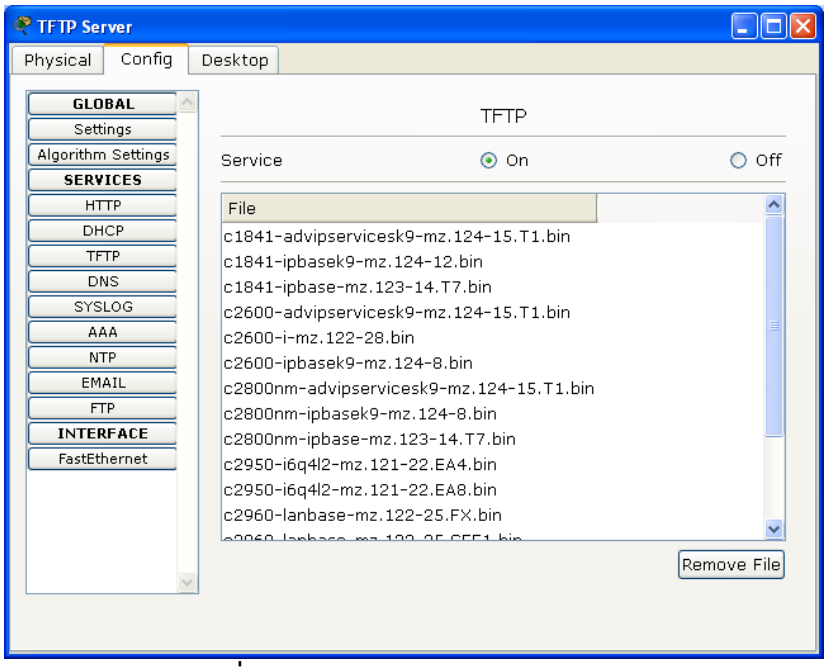
   บนเครื่อง PC0, PC1

   PC0: IP Address = 192.168.0.10, Subnet Mask = 255.255.255.0

   PC2: IP Address = 192.168.0.11, Subnet Mask = 255.255.255.0

   บนเครื่อง TFTP Server (แท็บ Config)

   IP Address = 192.168.0.100, Subnet Mask = 255.255.255.0 และตรวจสอบว่า TFTP อยู่ในสถานะ on หรือไม่ ถ้าไม่ให้เลือกเป็น on โดยปกติบนเครื่อง TFTP จะมีรายชื่อของไฟล์ที่เคยสำรองไว้แล้ว ดังรูปที่ 12.71



รูปที่ 12.71 กำหนดคุณสมบัติของ TFTP Server

2. บนอุปกรณ์สวิชต์ L2 (Switch 0) ให้ทำการกำหนดหมายเลขไอพีให้กับ vlan 1 เนื่องจาก ในทางทฤษฎีอุปกรณ์ที่ทำงานอยู่ในระดับเลเยอร์ 2 จะไม่จำเป็นต้องใช้หมายเลขไอพี แต่ปัจจุบันอุปกรณ์ L2 บางตัวจะต้องถูก monitoring เพื่อหาจุดบกพร่องหรือประเมินประสิทธิภาพ จึงอนุญาตอุปกรณ์ดังกล่าวสามารถมีไอพีเพื่อ monitor ได้ โดยทั่วไปจะกำหนดไว้บน vlan 1 เสมอ

   บน Switch 0: โหมด Configure

```
Switch(config)# interface vlan 1
Switch(config-if)#ip address 192.168.0.101 255.255.255.0 <ENTER>
(กำหนดให้ vlan 1 บนสวิชต์ L2 มีหมายเลขไอพี)
```

Switch(config-if)#**end** <ENTER> ออกจากโหมดคอนฟิก ไปยังโหมด admin
Switch#**ping 192.168.0.100** <ENTER> ทดสอบ ping เครื่อง TFTP server
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.0.100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/28/32 ms
จากผลลัพธ์ข้างต้นแสดงว่า ping สำเร็จ

การทดสอบ :

1. ให้ทำการทดสอบ TFTP เซิร์ฟเวอร์ โดยการสำรองไฟล์คอนฟิกและ IOS (Backup) จากสวิชต์ไปยัง TFTP Server โดยมีขั้นตอนดังนี้

Switch 0: โหมด admin

Switch#**copy running-config tftp:** <ENTER> สำรองไฟล์คอนฟิก (running-config) จากเครื่องสวิชต์ไปยัง tftp server
Address or name of remote host []? **192.168.0.100** <ENTER> กำหนดไอพีของ tftp server ที่ต้องการส่งไฟล์ไปเก็บไว้
Destination filename [Switch-confg]? **Room2Floor2-281110** <ENTER> กำหนดชื่อของไฟล์ที่ต้องการสำรอง ควรจะตั้งชื่อให้สื่อกับสถานที่ที่อุปกรณ์ดังกล่าวติดตั้งอยู่

Writing running-config...!!
[OK - 1024 bytes]

1024 bytes copied in 0.078 secs (13000 bytes/sec)
การโอนย้ายสำเร็จ จากนั้นให้เปิดดูที่ TFTP server จะมีไฟล์ดังกล่าวปรากฏอยู่

2. ทำสอบการโอนย้ายไฟล์คอนฟิกกลับคืนมา (Restore)

Switch#**copy tftp: running-config** <ENTER> สำรองคืนไฟล์คอนฟิกจากเครื่อง tftp กลับมายัง switch 0
Address or name of remote host []? **192.168.0.100** <ENTER> กำหนดไอพีของ tftp server ที่ต้องการโอนไฟล์กลับมา
Source filename []? **Room2Floor2-281110** <ENTER> กำหนดชื่อของไฟล์ที่ต้องการโอนย้ายกลับ
Destination filename [running-config]? <ENTER> กำหนดชื่อไฟล์ที่ต้องการเขียนทับ

ในที่นี้ไม่ควรเปลี่ยนชื่อไฟล์เพราะว่า ชื่อไฟล์คอนฟิกใน cisco เป็นค่า default คือ running-config สำหรับเก็บข้อมูลการคอนฟิกที่อยู่ในหน่วยความจำหลัก (สูญหายเมื่อปิดเครื่อง) และ startup-config เก็บคอนฟิกอยู่ในหน่วยความจำแบบไม่สูญหาย

```
Accessing tftp://192.168.0.100/Room2Floor2-281110...
Loading Room2Floor2-281110 from 192.168.0.100: !
[OK - 1024 bytes]


1024 bytes copied in 0.031 secs (33032 bytes/sec)
```

การ backup คอนฟิกสำเร็จ

3. ทำการสำรองและกู้คืนไฟล์ระบบปฏิบัติการหรือ IOS

Switch#**sh flash:** <ENTER> เพื่อตรวจสอบชื่อของ IOS ที่ต้องการสำรอง (สำหรับ Switch ตัวดังกล่าวจะมีชื่อว่า c2960-lanbase-mz.122-25.FX.bin สำหรับสวิชต์หรือเราเตอร์ตัวอื่นๆ จะมีชื่อที่แตกต่ากันไปตามรุ่น แต่ให้สังเกตุว่าไฟล์ IOS จะมีส่วนขยายเป็น .bin เสมอ)

Switch#**copy flash: tftp:** <ENTER> สั่งให้ทำการสำรอง IOS ไปยัง TFTP

Source filename []? **c2960-lanbase-mz.122-25.FX.bin** <ENTER> กำหนดชื่อไฟล์ IOS ที่ต้องการสำรอง

Address or name of remote host []? **192.168.0.100** <ENTER> ระบุไอพีของเครื่อง TFTP Server

Destination filename [c2960-lanbase-mz.122-25.FX.bin]? <ENTER> ตั้งชื่อ IOS (ควรใช้ค่า default)

```
Writing c2960-lanbase-mz.122-
25.FX.bin...!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 4414921 bytes]


4414921 bytes copied in 2.594 secs (1701000 bytes/sec)
```

โอนย้าย IOS สำเร็จ

ขั้นตอนการสำรอง IOS กลับจาก TFTP

Switch#**copy tftp: flash:** <ENTER> Backup IOS กลับคืนสู่สวิชต์

```
Address or name of remote host []? 192.168.0.100 <ENTER>
Source filename []? c2960-lanbase-mz.122-25.FX.bin <ENTER>
Destination filename [c2960-lanbase-mz.122-25.FX.bin]? <ENTER>
%Warning:There is a file already existing with this name
Do you want to over write? [confirm] <ENTER>
Erase flash: before copying? [confirm] <ENTER>
Erasing the flash filesystem will remove all files! Continue? [confirm]
<ENTER>
Erasing device...
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Erase of flash: complete
Accessing tftp://192.168.0.100/c2960-lanbase-mz.122-25.FX.bin...
Loading c2960-lanbase-mz.122-25.FX.bin from 192.168.0.100:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 4414921 bytes]

4414921 bytes copied in 2.594 secs (46242 bytes/sec)
```

Scenario 16: การติดตั้ง Wireless Access Point

คำอธิบาย :

ระบบเครือข่ายไร้สาย (Wireless LAN: WLAN) หมายถึง เทคโนโลยีที่ช่วยให้การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง หรือกลุ่มของเครื่องคอมพิวเตอร์สามารถสื่อสารกันได้ ร่วมถึงการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์เครือข่ายคอมพิวเตอร์ด้วยเช่นกัน โดยปราศจากการใช้สายสัญญาณในการเชื่อมต่อ แต่จะใช้คลื่นวิทยุเป็นช่องทางการสื่อสารแทน การรับส่งข้อมูลระหว่างกันจะผ่านอากาศ ทำให้ไม่ต้องเดินสายสัญญาณ และติดตั้งใช้งานได้สะดวกขึ้น

ระบบเครือข่ายไร้สายใช้แม่เหล็กไฟฟ้าผ่านอากาศ เพื่อรับส่งข้อมูลข่าวสารระหว่างเครื่องคอมพิวเตอร์ และระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์เครือข่าย โดยคลื่นแม่เหล็กไฟฟ้านี้อาจเป็นคลื่นวิทย (Radio) หรืออินฟาเรด (Infrared) ก็ได้ การสื่อสารผ่านเครือข่ายไร้สายมีมาตราฐาน IEEE802.11 เป็นมาตราฐานกำหนดรูปแบบการสื่อสาร ซึ่งมาตราฐานแต่ละตัวจะบอกถึงความเร็วและคลื่นความถี่สัญญาณที่แตกต่างกันใน การสื่อสารข้อมูล เช่น 802.11b และ 802.11g ที่ความเร็ว 11 Mbps และ 54 Mbps ตามลำดับ ขอบเขตของสัญญาณคลอบคุลพื้นที่ประมาณ 100 เมตร ในพื้นที่โปรง และประมาณ 30 เมตร ในอาคาร ซึ่งระยะทางของสัญญาณมีผลกระทบจากสิ่งรอบข้างหลายๆ อย่าง เช่น โทรศัพท์มือถือ ความหนาของกำแพง เครื่องใช้ไฟฟ้า อุปกรณ์อิเล็กทรอนิคส์ต่างๆ รวมถึงร่างกายมนุษย์ด้วยเช่นกัน สิ่งเหล่านี้มีผลกระทบต่อการใช้งานเครือข่ายไร้สายทั้งสิ้น

การเชื่อมต่อเครือข่ายไร้สายมี 2 รูปแบบ คือแบบ Ad-Hoc และ Infrastructure ทั้งสองรูปแบบมีการทำงานดังต่อไปนี้

1. การเชื่อมต่อแบบกลุ่มส่วนตัว(Ad-Hoc)

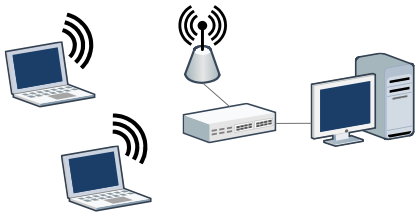
การเชื่อมต่อแบบ Ad-Hoc เป็นการเชื่อมต่อที่ประกอบด้วยเครื่องคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไปที่ติดตั้งการ์ดแลนไร้ ทำการเชื่อมต่อสื่อสารกันโดยตรงไม่ต้องผ่านอุปกรณ์กระจายสัญญาณ (Access Point) โดยเครื่องคอมพิวเตอร์ที่เชื่อมต่อแบบนี้สามารถสื่อสารแลกเปลี่ยนข้อมูลได้เช่น แชร์ไฟล์ เครื่องพิมพ์หรืออุปกรณ์ต่างๆ ดังรูปที่ 12.72

รูปที่ 12.72 การเชื่อมต่อแบบกลุ่มส่วนตัว (Ad-Hoc)

2. การเชื่อต่อแบบกลุ่มโครงสร้าง (Infrastructure)

การเชื่อมต่อแบบ Infrastructure เป็นการเชื่อมต่อที่มีอุปกรณ์กระจายสัญญาณ (Access Point) เป็นตัวกลาง ทำหน้าที่รับส่งสัญญาณและข้อมูลจากเครื่องคอมพิวเตอร์ไร้สายของเครือข่ายไร้สายไปสู่เครือข่ายมีสาย หากสังเกตจะพบว่า Access Point มีการทำงานเหมือนอุปกรณ์ฮับ (HUB)

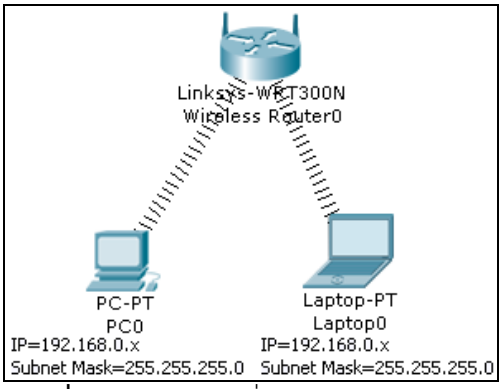
ในเครือข่ายคอมพิวเตอร์แบบมีสาย และที่สำคัญหากมีการเข้าใช้งานเครือข่ายไร้สายของเครื่องลูกข่ายในจำนวนมาก ต่อหนึ่ง Access Point จะมีผลทำให้ความเร็วของการสื่อสารเครือข่ายไร้สายช้าลงด้วยเช่นกัน ดังรูปที่ 12.83



รูปที่ 12.73 การเชื่อต่อแบบกลุ่มโครงสร้าง (Infrastructure)

หมายเหตุ: WLAN อ้างอิงจาก http://wise.swu.ac.th/

แผนผังการเชื่อมต่อ : ดังรูปที่ 12.74



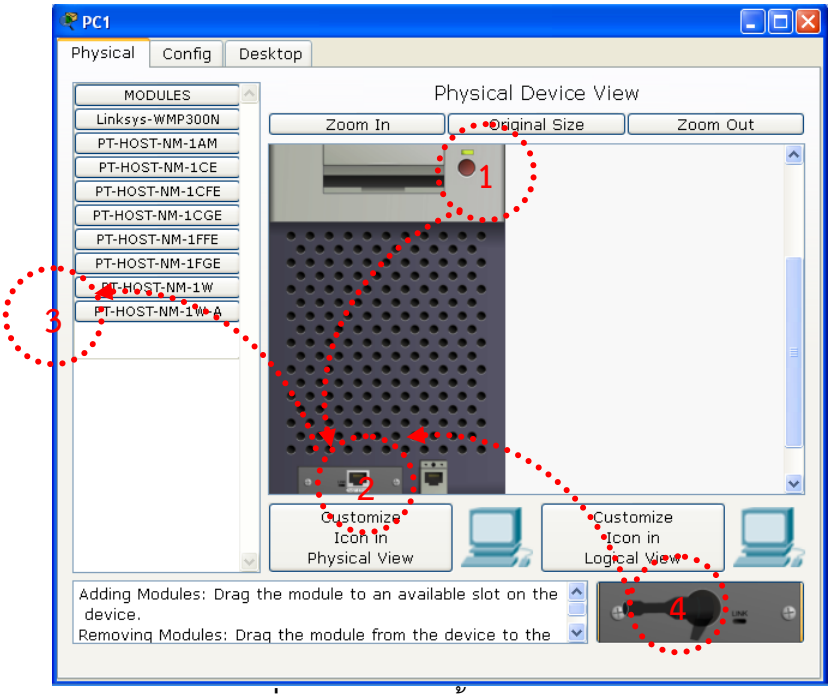
รูปที่ 12.74 ผังการเชื่อมต่อ scenario 16

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask
PC0	192.168.0.X	255.255.255.0
Laptop0	192.168.0.X	255.255.255.0
Wireless Router0	-	-

ขั้นตอนการเชื่อมต่อ :

1. เลือก Wireless Devices ⇨ Linksys-WRT300N วางใน workspace
2. เลือก End Devices ⇨ เลือก PC-PT และ Labtop-PT มาลงบน workspace
3. คลิกเลือก PC0 ⇨ แท็บ Physical ⇨ ปิดปุ่ม switch power ของเครื่อง PC0 แล้วคลิกลากเน็ตเวิร์คการ์ดออกจากเครื่อง PC0 แล้วเลือกการ์ด Wireless lan มาใส่แทน ดังรูปที่ 12.75



รูปที่ 12.75 การติดตั้ง wireless card

❶ ปิดปุ่ม power

❷ คลิกที่การ์ดเน็ตเวิร์ค

❸ ลาก FastEthernet ไปทิ้งใน MODULES (จะมีช่องว่างปรากฏ)

❹ คลิกลากเอาการ์ดไวเลสแลนด์ มาวางในช่องว่างแทน FastEthernet

4. คลิกเลือก PC0 ⇨ แท็บ Desktop ⇨ IP Configuration ⇨ เลือก DHCP จะได้รับ IP Address เป็นหมายเลข 192.168.0.x (x หมายถึง IP ที่ได้รับแจกมาจาก Wireless Access Point ซึ่งทำหน้าที่เป็น DHCP ในตัว)
5. คลิกเลือก Labtop0 ให้ทำตามขั้นตอนเหมือน PC0 คือเปลี่ยนจาก FastEthernet ไปเป็นการ์ด Wireless แทน

การทดสอบ :

1. ให้ทำการทดสอบการเชื่อมต่อโดย คลิกที่ PC0 ⇨ แท็บ Desktop ⇨ Command Prompt ⇨ แล้วทดสอบ ping ระหว่างเครื่อง PC0 กับ เครื่อง Labtop0 ว่ามีการตอบสนองหรือไม่

   บนเครื่อง PC0 (IP 192.168.0.100)

```
PC>
PC>ping 192.168.0.101 <ENTER> (ทดสอบโดยการ ping เครื่อง Labtop0)
Pinging 192.168.0.101 with 32 bytes of data:
```

```
Reply from 192.168.0.101: bytes=32 time=126ms TTL=128
Reply from 192.168.0.101: bytes=32 time=125ms TTL=128
Reply from 192.168.0.101: bytes=32 time=125ms TTL=128
Reply from 192.168.0.101: bytes=32 time=125ms TTL=128

Ping statistics for 192.168.0.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 125ms, Maximum = 126ms, Average = 125ms
```

Scenario 17: การติดตั้ง Wireless Access Point (WEP Authentication)

คำอธิบาย :

เนื่องจากระบบ Wireless LAN ใช้คลื่นวิทยุในการส่ง ดังนั้นการที่ผู้ไม่ประสงค์ดีสามารถดักจับสัญญาณ และ Hack เอาข้อมูลที่มีการรับ-ส่งระหว่างเครื่องลูกข่าย (Client) กับ Access Point ในการเชื่อมต่อแบบ Infrastructure หรือระหว่าง Client กับ Client ดังนั้นการใช้งาน Wireless LAN นั้น ควรที่จะต้องมีการกำหนดคุณสมบัติในด้านความปลอดภัยให้กับระบบด้วย ถ้าพูดถึงระบบ Security ของ Wireless LAN นั้น มีอยู่มากมายหลายวิธี โดยขึ้นอยู่กับอุปกรณ์แต่ละประเภท ในที่นี้จะเป็นการกำหนด Security ที่ผู้ใช้จำเป็นต้องทำ 5 ข้อคือ

1. เปลี่ยน SSID คือชื่อของ Network ที่เราตั้งขึ้นมาเอง โดยที่ทุกๆ เครื่องในระบบที่เชื่อมต่อด้วยต้องตั้งค่า SSID เป็นค่าเดียวกัน เมื่อซื้อ Wireless Access Point มาใหม่ๆ จะมีการตั้งค่า SSID ไว้แล้วเป็นค่า default แต่เราควรที่จะเปลี่ยนชื่อ SSID ในทันทีที่ติดตั้ง การตั้งชื่อ SSID นั้นต้องไม่เกิน 32 ตัวอักษร เช่น ITNetwork เป็นต้น

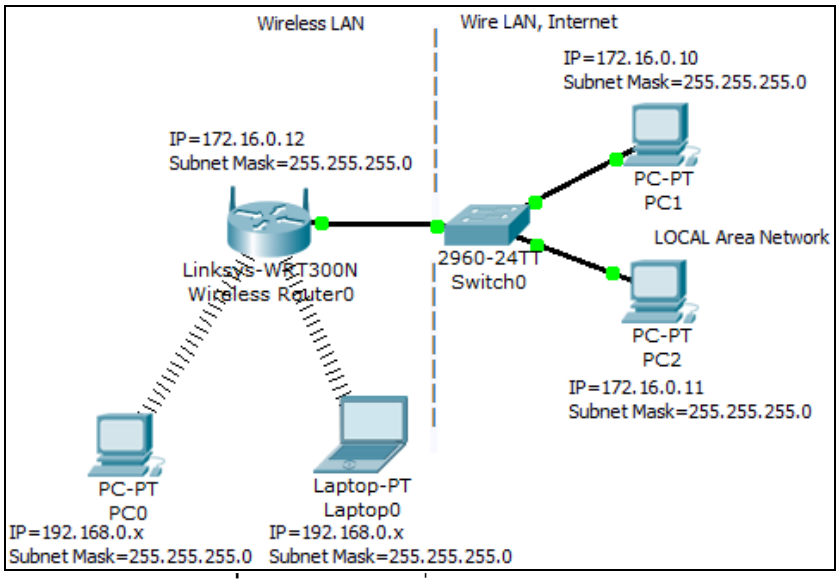
2. เปลี่ยน Password สำหรับ Admin ซึ่งโดยปกติค่าเริ่มต้นจะเป็นค่าที่ง่ายๆ เช่น admin, password เป็นต้น เพราะฉะนั้นคุณจำเป็นต้องเปลี่ยน Default Password ของ Wireless Access Point ของคุณทันทีที่เริ่มติดตั้งระบบครับ

3. กำหนดค่า SSID Broadcast = Disabled, SSID Broadcast คือการยอมให้เผยแพร่ SSID ให้ทุกๆ เครื่องที่อยู่ในระยะส่งของ Wireless Access Point (AP) สามารถที่จะเห็น AP ของเราได้ ซึ่งเป็นสิ่งที่ดีในขณะที่เราทำการติดตั้งระบบในครั้งแรก เพราะจะทำให้ง่ายในการทดสอบระบบและเซตเครื่องลูกข่าย แต่หลังจากที่เราติดตั้งระบบ Wireless Network เรียบร้อยแล้ว เราควรที่จะยกเลิก SSID Broadcast ในทันที เพราะการที่เราเปิดเผย SSID ของเรานั้น อาจทำให้ผู้ไม่ประสงค์ดี สามารถที่จะแอบเข้ามาในระบบ Network ของเราได้ง่ายขึ้น

4. กำหนด WEP Encryption = Enabled, WEP ย่อมาจาก Wired Equivalent Privacy เป็นรูปแบบการเข้ารหัสของอุปกรณ์ Wireless LAN ที่แพร่หลายที่สุด ไม่ว่าจะเป็น Wireless Adapter รุ่นใดๆ ก็ใช้ WEP ได้, ในการเข้ารหัสแบบ WEP นั้น สามารถที่จะเลือกระดับของการเข้ารหัสได้ว่า จะใช้ 64-bit, 128-bit หรือ 256-bit โดยการใช้จำนวน Bit ที่มากขึ้นนั้น ทำให้ความเร็วในการเชื่อมต่อลดลง แต่ว่าจำนวน Bit ยิ่งมาก ก็ยิ่งทำให้มีความปลอดภัยมากขึ้น แต่ในปัจจุบัน WEP ถูก Hack ได้ง่ายแล้ว ส่วนการเข้ารหัสแบบใหม่ ซึ่งออกมาแทน WEP นั้นมีชื่อว่า WPA ย่อมาจาก Wi-Fi Protected Access ซึ่งมีความปลอดภัยสูงมาก ปัจจุบันถึง WPA2 แล้ว

5. MAC Address Filtering, MAC Address ทำหน้าที่เสมือนเลขประจำตัวของอุปกรณ์ Network ต่างๆ ซึ่งอุปกรณ์ Network ทุกชิ้นในโลกนี้ จะไม่มี MAC Address ที่ซ้ำกันเลยครับ ดังนั้น การที่เราสามารถที่จะกำหนดให้แค่เครื่องคอมพิวเตอร์ของเราเท่านั้น ที่สามารถเข้าสู่ Wireless Network ของเราได้ ก็ย่อมที่จะเสริมความปลอดภัยให้กับระบบ Wireless LAN ของเราให้ดีขึ้นไปอีกขั้นหนึ่ง

แผนผังการเชื่อมต่อ :



รูปที่ 12.76 ผังการเชื่อมต่อ scenario 17

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask
PC0	192.168.0.X	255.255.255.0
Laptop0	192.168.0.X	255.255.255.0
PC1	172.16.0.10	255.255.255.0
PC2	172.16.0.11	255.255.255.0
Wireless Router0	172.16.0.12	255.255.255.0

ขั้นตอนการเชื่อมต่อ :

1. เลือก Wireless Devices ⇨ Linksys-WRT300N วางใน workspace
2. เลือก End Devices ⇨ เลือก PC-PT และ Labtop-PT มาลงบน workspace
3. เครื่อง PC0 และ Labtop0 ให้เปลี่ยนการ์ดเน็ตเวิร์คจาก FastEthernet เป็น Wireless มาใส่แทน
4. เครื่อง PC1 และ PC2 ให้คอนฟิก IP Address และ Subnet Mask ตามตารางด้านบน
5. เครื่อง Wireless Router0 ให้เลือก Config ⇨ เมนูด้านซ้ายเลือก Internet ⇨ Internet Settings (เป็น IP ที่ใช้เชื่อมต่อเพื่อออกสู่อินเทอร์เน็ต) ⇨ ในส่วน Connection Type ให้เลือก Static และกำหนด หมายเลข IP เป็น 172.16.0.12, Subnet Mask เป็น 255.255.255.0
6. เครื่อง Wireless Router0 ให้เลือก Config ⇨ เมนูด้านซ้ายเลือก LAN ⇨ LAN Settings (เป็น IP ที่ใช้เพื่อเป็นทางออกให้กับเครื่องลูกข่ายใน Wireless LAN) ⇨ กำหนด หมายเลข IP เป็น 192.168.0.1, Subnet Mask เป็น 255.255.255.0
7. เครื่อง Wireless Router0 ให้เลือก Config ⇨ เมนูด้านซ้ายเลือก Wireless ⇨ Wireless Settings (กำหนดค่าการรักษาความปลอดภัยแบบ WEP) ⇨ ให้กำหนดค่าพารามิเตอร์ดังนี้

พารามิเตอร์	**ค่าที่กำหนด**	**คำอธิบาย**
SSID	IT01	กำหนดค่า SSID
Channel	6	กำหนดช่องสัญญาณในการสื่อสาร
Authentication	WEP	เลือกการเข้ารหัสการสื่อสารชนิด WEP
Key	1a2b3c4d5e	กำหนดคีย์ในการเข้ารหัส ต้องไม่น้อยกว่า 10 ตัวอักษร และเป็นเลขฐาน 16
Encryption Type	40/64 bits	เลือกขนาดของบิตในการเข้ารหัส ยิ่งมากยิ่งปลอดภัย

8. คลิกเลือก PC0 ⇨ แท็บ Desktop ⇨ PC Wireless ⇨ เลือกแท็บ Connect ต่อจากนั้นให้กดปุ่ม Refresh 1-2 ครั้ง Wireless Access Name จะขึ้นชื่อเป็น IT01 ให้เลือก Connect ⇨ ในช่อง Security เลือก WEP, ในช่อง WEP เลือก 64 bit และช่อง Key 1 ให้ใส่คีย์มีค่าเป็น 1a2b3c4d5e เข้าไป ⇨ กดปุ่ม Connect
9. คลิกเลือก PC0 ⇨ แท็บ Desktop ⇨ IP Configuration ⇨ เลือก DHCP จะปรากฏ IP หมายเลข 192.168.0.x (x หมายถึง IP ที่ Wireless Router0 แจกให้อัตโนมัติ) จากนั้นทดสอบโดยการ ping ไปยัง IP Gateway ของตนเองคือ 192.168.0.1 และ ping ไปยังเครื่อง PC1 (IP 172.16.0.10), และ PC2 (IP 172.16.0.10) ตาลำดับ ต้องมี

การตอบสนองกลับมาจากเครื่องทั้ง 3 จากนั้นให้คอนฟิกค่าของ Labtop0 เหมือนขั้นตอนที่กระทำกับเครื่อง PC0 ทุกประการ (เหมือนกับขั้นตอนที่ 8)

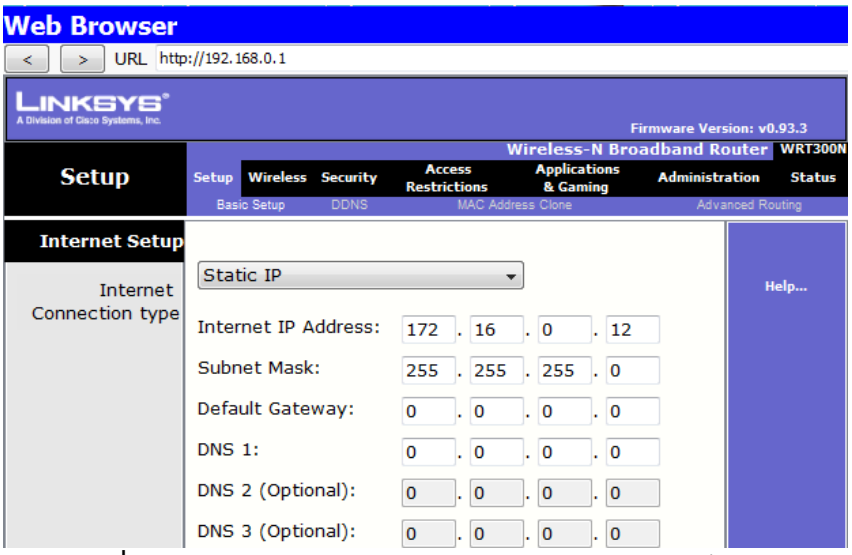
การทดสอบ :

1. ให้ทำการทดสอบการเชื่อมต่อโดย คลิกที่ PC0 ⇨ แท็บ Desktop ⇨ Command Prompt ⇨ แล้วทดสอบ ping ระหว่างเครื่อง PC0 กับ เครื่อง Labtop0 ว่ามีการตอบสนองหรือไม่

บนเครื่อง PC0 (IP 192.168.0.100)

PC>
PC>**ping 192.168.0.1** <ENTER> (ทดสอบโดยการ ping gateway)
PC>**ping 192.168.0.101** <ENTER> (ทดสอบโดยการ ping เครื่อง Labtop0)
PC>**ping 172.16.0.10** <ENTER> (ทดสอบโดยการ ping เครื่อง PC1)
PC>**ping 172.16.0.11** <ENTER> (ทดสอบโดยการ ping เครื่อง PC2)

2. ทดสอบแบบที่ 2 โดยการเข้าไปจัดการคอนฟิก Wireless Router0 โดยผ่านหน้าเว็บเพจ (ต้องคอนฟิกให้ Wireless Router0 เปิดการใช้งาน Web Management ก่อน โดยเข้าไปที่ Wireless Router0 ⇨ GUI ⇨ Administration ⇨ กำหนดรหัสผ่าน (กำหนดเป็น 123) ⇨ เลือกลงมาด้านล้างคลิก Save Settings) ไปที่ PC0 เลือก Desktop ⇨ Web Browser ⇨ ช่อง URL ให้ใส่หมายเลข IP Gateway ของ Wireless ในที่นี้คือ 192.168.0.1 แล้วกด Go ⇨ จะปรากฏเมนูให้ใส่ Username และ Password ให้ใส่ Username = admin, Password=123 แล้วกด Ok เครื่อง PC0 ก็จะสามารถคอนฟิก Wireless Router0 ผ่านเว็บเพจได้ ดังรูปที่ 12.77



รูปที่ 12.77 แสดงการคอนฟิก Wireless Router0 ผ่านเว็บเพจ

สำหรับการเลือกวิธีเข้ารหัสแบบอื่นๆ จะมีหลักการคล้ายกับ WEP ซึ่งจะมีแทรกอยู่ใน Scenario อื่นๆ ต่อไป

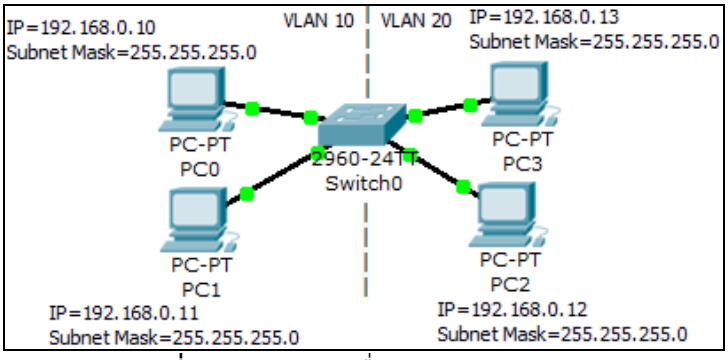
Scenario 18: การคอนฟิก VLAN (บน switch 2900 series)

คำอธิบาย :

VLAN ย่อมาจาก Virtual LAN เป็นเทคโนโลยีที่ใช้ในการจำลองสร้างเครือข่าย LAN แต่ไม่ขึ้นอยู่กับการต่อทางกายภาพ เช่น สวิตช์หนึ่งตัวสามารถใช้จำลองเครือข่าย LAN ได้มากกว่า 1 เครือข่าย หรือสามารถใช้สวิตช์หลายตัวจำลองเครือข่าย LAN เพียงหนึ่งเครือข่าย เป็นต้น

ในการสร้าง VLAN โดยใช้อุปกรณ์เครือข่ายหลายตัว จะมีพอร์ตที่ทำหน้าที่เชื่อมต่อระหว่างอุปกรณ์เครือข่ายแต่ละตัว เรียก Trunk port ซึ่งเสมือนมีท่อเชื่อม เนื่องจาก VLAN เป็น LAN แบบจำลอง ถึงแม้ว่าจะต่อทางกายภาพอยู่บนอุปกรณ์เครือข่ายตัวเดียวกัน แต่การติดต่อกันนั้นจำเป็นต้องใช้อุปกรณ์ที่มีความสามารถในการค้นหาเส้นทาง เช่น เราเตอร์ หรือสวิตช์เลเยอร์ 3 ในการเชื่อมต่อ VLAN ให้สามารถสื่อสารกันได้

แผนผังการเชื่อมต่อ :



รูปที่ 12.78 ผังการเชื่อมต่อ scenario 18

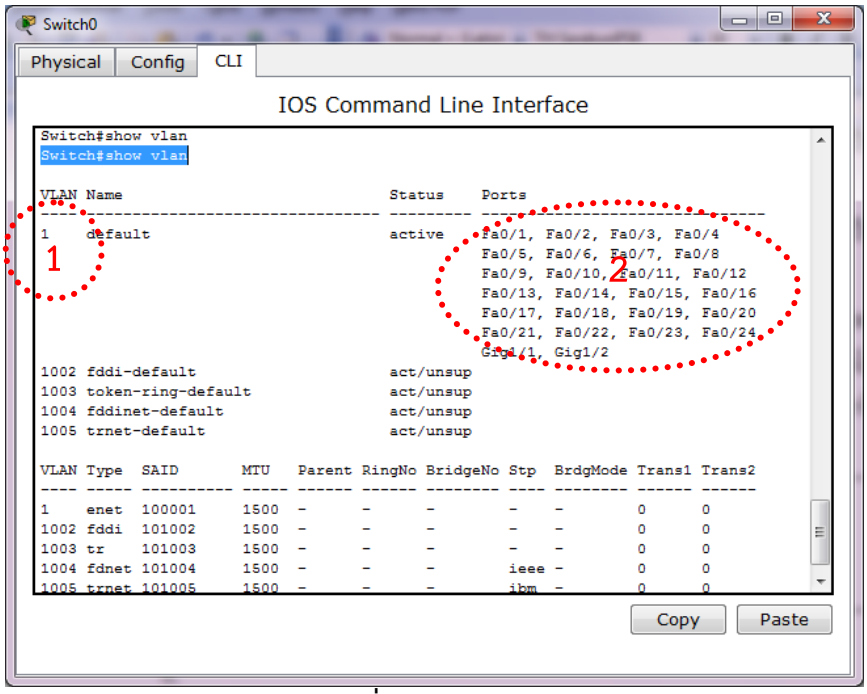
รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	VLAN
PC0	192.168.0.10	255.255.255.0	VLAN 10
PC1	192.168.0.11	255.255.255.0	VLAN 10
PC2	192.168.0.12	255.255.255.0	VLAN 20
PC3	192.168.0.13	255.255.255.0	VLAN 20
Switch0	-	-	-

ขั้นตอนการเชื่อมต่อ :

1. เลือก PC0 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.0.10, Subnet Mask=255.255.255.0
2. เลือก PC1 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.0.11, Subnet Mask=255.255.255.0
3. เลือก PC2 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.0.12, Subnet Mask=255.255.255.0
4. เลือก PC3 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.0.13, Subnet Mask=255.255.255.0
5. ทดสอบ ping จาก PC0 ไปยัง PC1, PC2, PC3 ซึ่งผลของการ ping จะมีการตอบสนองจากเครื่องปลายทางทุกๆ เครื่อง เนื่องจาก switch0 จะมี VLAN สร้างขึ้นอัตโนมัติคือ VLAN1 เสมอ ทำให้ทุกๆ เครื่องสามารถสื่อสารกันได้ทันทีเมื่อกำหนด IP Address เสร็จ การแสดงข้อมูล VLAN จะใช้คำสั่ง show vlan บนสวิตช์ โดยคลิกที่ switch0 ⇨ CLI ⇨ <ENTER>

Switch>
Switch>**enable** <ENTER> เข้าสู่โหมดผู้ดูแลระบบ
Switch#**show vlan** <ENTER> แสดงรายการ VLAN ดังรูปที่ 12.79



รูปที่ 12.79 แสดงรายการ VLAN

❶ แสดง default VLAN คือ VLAN 1 จะสร้างมาพร้อมกับสวิตช์เสมอ

❷ แสดงพอร์ตที่เป็นสมาชิกของ VLAN 1 ในเบื้องต้นทุกๆ พอร์ตจะเป็นสมาชิกของ VLAN 1 ทั้งหมด

6. เพื่อเป็นการทดสอบคุณสมบัติของ VLAN จะทดลองสร้าง VLAN 10 และ VLAN 20 จากนั้นกำหนดให้ เครื่อง PC0, PC1 เป็นสมาชิกของ VLAN 10 และ PC2, PC3 เป็นสมาชิกของ VLAN 20 โดยมีขั้นตอนดังนี้ คลิกที่ Switch0 ⇨ CLI ⇨ <ENTER>

Switch>

Switch>**enable** <ENTER> เข้าสู่โหมดผู้ดูแลระบบ

Switch#**configure terminal** <ENTER> เข้าสู่โหมดการคอนฟิก

Enter configuration commands, one per line. End with CNTL/Z.

Switch(config)#**interface fastEthernet 0/1** <ENTER> เข้าไปยังอินเทอร์เฟส Fa0/1

Switch(config-if)#**switchport access vlan 10** <ENTER> เปลี่ยนสมาชิกจาก VLAN 1 เป็น VLAN 10 ของเครื่อง PC0

% Access VLAN does not exist. Creating vlan 10 กรณีไม่มี VLAN สวิตช์จะสร้างให้อัตโนมัติ

Switch(config-if)#**exit** <ENTER> ออกจากการคอนฟิกอินเทอร์เฟส Fa0/1

Switch(config)#**interface fastEthernet 0/2** <ENTER> เข้าไปยังอินเทอร์เฟส Fa0/2

Switch(config-if)#**switchport access vlan 10** <ENTER> เปลี่ยนสมาชิกจาก VLAN 1 เป็น VLAN 10 ของเครื่อง PC1

Switch(config-if)#**exit** <ENTER> ออกจากการคอนฟิกอินเทอร์เฟส Fa0/2

Switch(config)#**interface fastEthernet 0/3** <ENTER> เข้าไปยังอินเทอร์เฟส Fa0/3

Switch(config-if)#**switchport access vlan 20** <ENTER> > เปลี่ยนสมาชิกจาก VLAN 1 เป็น VLAN 20 ของเครื่อง PC2

% Access VLAN does not exist. Creating vlan 20

Switch(config-if)#**exit** <ENTER> ออกจากการคอนฟิกอินเทอร์เฟส Fa0/3

Switch(config)#**interface fastEthernet 0/4** <ENTER> เข้าไปยังอินเทอร์เฟส Fa0/4

Switch(config-if)#**switchport access vlan 20** <ENTER> เปลี่ยนสมาชิกจาก VLAN 1 เป็น VLAN 20 ของเครื่อง PC3

7. แสดงการคอนฟิก VLAN อีกครั้ง โดยใช้คำสั่ง show vlan ผลที่ถูกต้องคือ จะมี vlan ใหม่เกิดขึ้น 2 VLAN คือ VLAN 10 และ 20 ซึ่งใน VLAN 10 จะมีพอร์ตที่เป็นสมาชิกคือ Fa0/1 (PC0), Fa0/2 (PC1) และ VLAN 20 มีพอร์ตที่เป็นสมาชิกคือ Fa0/3 (PC2), Fa0/4 (PC3)

Switch#**show vlan** <ENTER>

VLAN Name **Status** **Ports**

10	VLAN0010	active	Fa0/1, Fa0/2
20	VLAN0020	active	Fa0/3, Fa0/4

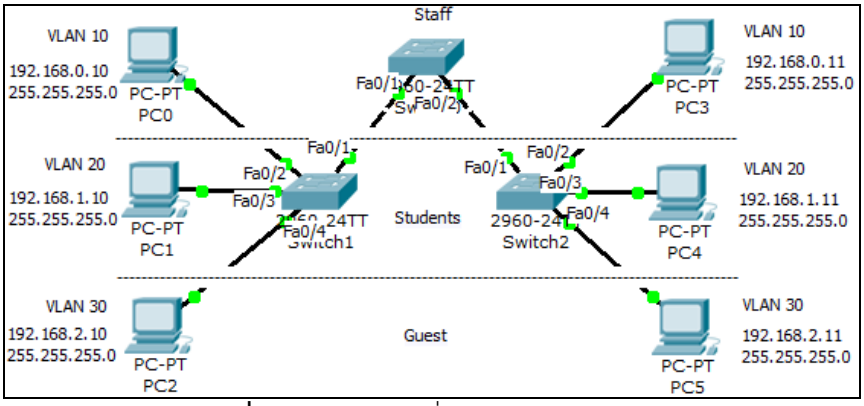
8. ทดสอบ ping เหมือนข้อที่ 5 อีกครั้ง ผลที่ถูกต้องคือ เครื่อง PC0 และ PC1 จะสามารถ ping กันได้ (อยู่ใน VLAN เดียวกัน), PC2 และ PC3 สามารถ ping กันได้ แต่เครื่องที่อยู่ต่าง VLAN กันจะไม่สามารถ ping กันได้ ซึ่งเป็นไปตามคุณสมบัติของ VLAN

Scenario 19: การคอนฟิก VLANs และ Trunks (บน switch 2900 series)

คำอธิบาย :

ในการสร้าง VLAN โดยใช้อุปกรณ์เครือข่ายหลายตัว จะมีพอร์ตที่ทำหน้าที่เชื่อมต่อระหว่างอุปกรณ์เครือข่ายแต่ละตัว เรียก Trunk port ซึ่งเสมือนมีท่อเชื่อม หรือ Trunk เป็นตัวเชื่อม VLAN ของแต่ละสวิตช์เข้าด้วยกัน ซึ่งเป็นการเชื่อมต่อของ vlan ในแต่ละเครื่องที่เป็น vlan ชื่อเดียวกันเข้าด้วยกัน แต่ไม่สามารถเชื่อม vlan ที่มีชื่อต่างกันเข้ากันได้ ถ้าต้องการให้ vlan ที่แตกต่างกันสามารถสื่อสารกันได้ต้องอาศัยอุปกรณ์ที่ทำงานในเลเยอร์ที่ 3 เช่น สวิตช์เลเยอร์ 3 หรือเราเตอร์ เป็นต้น

แผนผังการเชื่อมต่อ :



รูปที่ 12.80 ผังการเชื่อมต่อ scenario 19

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	VLAN
PC0	192.168.0.10	255.255.255.0	VLAN 10
PC3	192.168.0.11	255.255.255.0	VLAN 10
PC1	192.168.1.10	255.255.255.0	VLAN 20
PC4	192.168.1.11	255.255.255.0	VLAN 20
PC2	192.168.2.10	255.255.255.0	VLAN 30
PC5	192.168.2.11	255.255.255.0	VLAN 30

Switch0	-	-	Trunk Fa0/1, Fa0/2
Switch1	-	-	Trunk Fa0/1
Switch2	-	-	Trunk Fa0/1

ขั้นตอนการเชื่อมต่อ :

1. เลือก PC0 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.0.10, Subnet Mask=255.255.255.0
2. เลือก PC3 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.0.11, Subnet Mask=255.255.255.0
3. เลือก PC1 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.1.10, Subnet Mask=255.255.255.0
4. เลือก PC4 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.1.11, Subnet Mask=255.255.255.0
5. เลือก PC2 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.2.10 Subnet Mask=255.255.255.0
6. เลือก PC5 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.2.11, Subnet Mask=255.255.255.0
7. เลือก Switch0 ⇨ CLI ⇨ <ENTER>

บนเครื่อง Switch0

Switch>
Switch0>**enable** <ENTER> เข้าสู่โหมดผู้ดูแลระบบ
Switch0#**configure terminal** <ENTER> เข้าสู่โหมดการคอนฟิก
Switch0(config)#**vlan 10** <ENTER> สร้าง vlan 10 บนเครื่อง Switch0
Switch0(config-vlan)#**name Staff** <ENTER> ตั้งชื่อให้ vlan 10 เป็นของ Staff
Switch0(config)#**vlan 20** <ENTER> > สร้าง vlan 20 บนเครื่อง Switch0
Switch0(config-vlan)#**name Students** <ENTER> ตั้งชื่อให้ vlan 20 เป็นของ Students
Switch0(config)#**vlan 30** <ENTER> สร้าง vlan 30 บนเครื่อง Switch0
Switch0(config-vlan)#**name Guest** <ENTER> ตั้งชื่อให้ vlan 30 เป็นของ Guest
Switch0(config)#**vlan 99** <ENTER> สร้าง vlan 99 บนเครื่อง Switch0 เพื่อเป็นพอร์ต manage
Switch0(config-vlan)#**name Management** <ENTER> ตั้งชื่อให้ vlan 99 เป็นของ Management

บนเครื่อง Switch0 สามารถแสดง vlan ด้วยคำสั่งดังต่อไปนี้

Switch0#**show vlan** <ENTER> แสดงรายการ vlan ทั้งหมดบน Switch0

VLAN	Name	Status	Ports
10	Staff	active	
20	Students	active	
30	Guest	active	
99	Management	active	

8. บน Switch1, Switch2 ให้ทำการสร้าง vlan เหมือนกับ Switch0 ทุกประการ

9. ทำการเปลี่ยนสมาชิกของพอร์ตเข้าสู่ vlan ที่กำหนด บน Switch1 และ Switch2 บนเครื่อง Switch1 ให้ทำการปลี่ยนพอร์ต Fa0/2 เข้า vlan 10, Fa0/3 เข้า vlan 20, Fa0/4 เข้า vlan 30

Switch1(config-if)#**interface fastEthernet 0/2** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/2
Switch1(config-if)#**switchport mode access** <ENTER>
Switch1(config-if)#**switchport access vlan 10** <ENTER> ย้ายจาก vlan 1 เข้าสู่ vlan 10
Switch1(config-if)#**interface fastEthernet 0/3** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/3
Switch1(config-if)#**switchport mode access** <ENTER>
Switch1(config-if)#**switchport access vlan 20** <ENTER> ย้ายจาก vlan 1 เข้าสู่ vlan 20
Switch1(config-if)#**interface fastEthernet 0/4** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/4
Switch1(config-if)#**switchport mode access** <ENTER>
Switch1(config-if)#**switchport access vlan 30** <ENTER> ย้ายจาก vlan 1 เข้าสู่ vlan 30

10. บน Switch2 ให้ทำการย้ายพอร์ตเข้าสู่ vlan เหมือนขั้นตอนที่ 9 ทุกประการ

11. ทำการสร้าง Trunking ระหว่าง Switch0 กับ Switch1 และ Switch0 กับ Switch2 บน Switch0

Switch0(config)#**interface fastEthernet 0/1** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/1
Switch0(config-if)#**switchport mode trunk** <ENTER> เปลี่ยนเป็นโหมด Trunk
Switch0(config-if)#**switchport trunk native vlan 99** <ENTER> กำหนด Fa0/1 ให้เข้าสู่ vlan 99 ซึ่งเป็น Trunk port
Switch(config)#**interface fastEthernet 0/2** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/2
Switch(config-if)#**switchport mode trunk** <ENTER> เปลี่ยนเป็นโหมด Trunk
Switch(config-if)#**switchport trunk native vlan 99** <ENTER> กำหนด Fa0/2 ให้เข้าสู่ vlan 99 ซึ่งเป็น Trunk port

บน Switch1 ทำการสร้าง vlan trunk ดังนี้

Switch1(config)#**interface fastEthernet 0/1** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/1
Switch1(config-if)#**switchport mode trunk** <ENTER> เปลี่ยนเป็นโหมด trunk

Switch1(config-if)#**switchport trunk native vlan 99** <ENTER> ย้ายจาก vlan 1 เข้าสู่ vlan 99 ซึ่งเป็นโหมด trunk

บน Switch2 ทำการสร้าง vlan trunk ดังนี้

Switch2(config)#**interface fastEthernet 0/1** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/1 Switch2(config-if)# **switchport mode trunk** <ENTER> เปลี่ยนเป็นโหมด trunk Switch2(config-if)# **switchport trunk native vlan 99** <ENTER> ย้ายจาก vlan 1 เข้าสู่ vlan 99 ซึ่งเป็นโหมด trunk

การทดสอบ :

1. ให้ทำการทดสอบการเชื่อมต่อโดย การ ping ภายใน vlan เดียวกัน คือ PC0 และ PC3, PC2 และ PC4, PC3 และ PC5 ต้องสามารถเชื่อมต่อกันได้ แต่ถ้า ping ข้าม vlan จะไม่สามารถ ping กันได้

Scenario 20: การคอนฟิก VTP (บน switch 2900 series)

คำอธิบาย :

VTP = Virtual Trunking Protocol เป็นโปรโตคอลที่ช่วยให้ง่ายต่อการสร้าง VLAN ในสวิตซ์บนเครือข่าย ถ้าเป็น cisco จะนิยามว่าเป็นโปรโตคอลพิเศษที่ใช้เพื่อช่วยให้ง่ายต่อการสร้าง ลบ และเปลี่ยนชื่อของ VLAN ในเน็ตเวิร์ค ผ่านพอร์ต Trunk สิ่งที่ต้องศึกษาและต้องรู้ในเรื่อง VTP ก็คือ

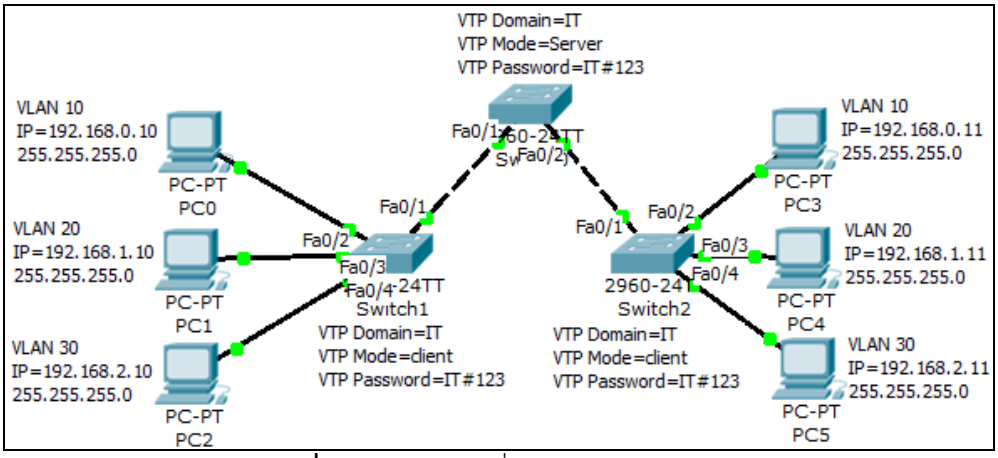
VTP Domains กำหนดชื่อของ VTP Domain

VTP Modes กำหนด Mode การทำงานของ Domain

VTP Password กำหนด Password เพื่อป้องกัน Domain

VTP เอาไว้ใช้เวลาที่เราต้องการเซ็ต VLAN บนสวิตซ์หลายๆตัวครับ โดยเราจะทำการสร้าง VLAN DATABASE ขึ้นมาที่สวิตซ์ตัวหนึ่ง แล้วตั้งให้เป็น VTP Server จากนั้นที่สวิตซ์ตัวอื่นๆ ก็ตั้งให้เป็น VTP Client โดยทั้ง Server และ Client ต้องอยู่ใน VTP Domains เดียวกัน หลังจากนั้นก็ enable โพรโทคอลให้สวิตซ์สื่อสารถึงกัน ตัว Client ก็จะรับ VLAN Database มาจาก Server โดยอัตโนมัติ โดยที่เราไม่ต้องเข้าไปสร้างเองในทุกๆสวิตซ์ VTP จะมีประโยชน์มากถ้าเราต้องสร้าง VLAN โดยใช้สวิตซ์หลายๆ ตัว

แผนผังการเชื่อมต่อ :



รูปที่ 12.81 ผังการเชื่อมต่อ scenario 20

รายการอุปกรณ์ :

อุปกรณ์	IP Address	Subnet Mask	VLAN
PC0	192.168.0.10	255.255.255.0	VLAN 10
PC3	192.168.0.11	255.255.255.0	VLAN 10
PC1	192.168.1.10	255.255.255.0	VLAN 20
PC4	192.168.1.11	255.255.255.0	VLAN 20
PC2	192.168.2.10	255.255.255.0	VLAN 30
PC5	192.168.2.11	255.255.255.0	VLAN 30
Switch0	-	-	Trunk Fa0/1, Fa0/2

ขั้นตอนการเชื่อมต่อ :

1. เลือก PC0 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.0.10, Subnet Mask=255.255.255.0
2. เลือก PC3 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.0.11, Subnet Mask=255.255.255.0
3. เลือก PC1 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.1.10, Subnet Mask=255.255.255.0
4. เลือก PC4 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.1.11, Subnet Mask=255.255.255.0
5. เลือก PC2 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.2.10 Subnet Mask=255.255.255.0
6. เลือก PC5 ⇨ Desktop ⇨ IP Configuration ⇨ กำหนดหมายเลข IP เป็น 192.168.2.11, Subnet Mask=255.255.255.0

7. เลือก Switch0 ⇨ CLI ⇨ <ENTER>

บนเครื่อง Switch0

Switch0#**configure terminal** <ENTER> เข้าสู่โหมดการคอนฟิก
Switch0(config)#**vtp domain IT** <ENTER> กำหนดโดเมนชื่อ IT
Switch0(config)#**vtp mode server** <ENTER> กำหนดโหมดการทำงานเป็น server
Switch0(config)#**vtp password IT#123** <ENTER> กำหนดรหัสผ่านของโดเมน IT
Switch0(config)#**vlan 10** <ENTER> สร้าง vlan 10 บนเครื่อง Switch0
Switch0(config-vlan)#**name Staff** <ENTER> ตั้งชื่อให้ vlan 10 เป็นของ Staff
Switch0(config-vlan)#**exit** <ENTER>
Switch0(config)#**vlan 20** <ENTER> > สร้าง vlan 20 บนเครื่อง Switch0
Switch0(config-vlan)#**name Students** <ENTER> ตั้งชื่อให้ vlan 20 เป็นของ Students
Switch0(config-vlan)#**exit** <ENTER>
Switch0(config)#**vlan 30** <ENTER> สร้าง vlan 30 บนเครื่อง Switch0
Switch0(config-vlan)#**name Guest** <ENTER> ตั้งชื่อให้ vlan 30 เป็นของ Guest
Switch0(config-vlan)#**exit** <ENTER>
Switch0(config)#**vlan 99** <ENTER> สร้าง vlan 99 บนเครื่อง Switch0 เพื่อเป็นพอร์ต manage
Switch0(config-vlan)#**name Management** <ENTER> ตั้งชื่อให้ vlan 99 เป็นของ Management
Switch0(config-vlan)#**exit** <ENTER>
Switch0(config)#**interface fastEthernet 0/1** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/1
Switch0(config-if)#**switchport mode trunk** <ENTER> เปลี่ยนเป็นโหมด Trunk
Switch0(config-if)#**switchport trunk native vlan 99** <ENTER> กำหนด Fa0/1 ให้เข้าสู่ vlan 99 ซึ่งเป็น Trunk port
Switch0(config-vlan)#**exit** <ENTER>
Switch(config)#**interface fastEthernet 0/2** <ENTER> เข้าสู่อินเทอร์เฟส Fa0/2
Switch(config-if)#**switchport mode trunk** <ENTER> เปลี่ยนเป็นโหมด Trunk
Switch(config-if)#**switchport trunk native vlan 99** <ENTER> กำหนด Fa0/2 ให้เข้าสู่ vlan 99 ซึ่งเป็น Trunk port
Switch0(config-vlan)#**exit** <ENTER>

8. บน Switch1, Switch2 ให้ทำการคอนฟิกเฉพาะ vtp domain, mode และรหัสผ่านเท่านั้น ข้อมูล vlan ทั้งหมดจะถูกสำเนามาจาก Switch0 ทันที

บนเครื่อง Switch1, Switch2 ให้คอนฟิกเหมือนกันดังนี้

Switch1#**configure terminal** <ENTER> เข้าสู่โหมดการคอนฟิก

Switch1(config)#**vtp domain IT** <ENTER> กำหนดโดเมนชื่อ IT
Switch1(config)#**vtp mode client** <ENTER> กำหนดโหมดการทำงานเป็น client
Switch1(config)#**vtp password IT#123** <ENTER> กำหนดรหัสผ่านของโดเมน IT

การทดสอบ :

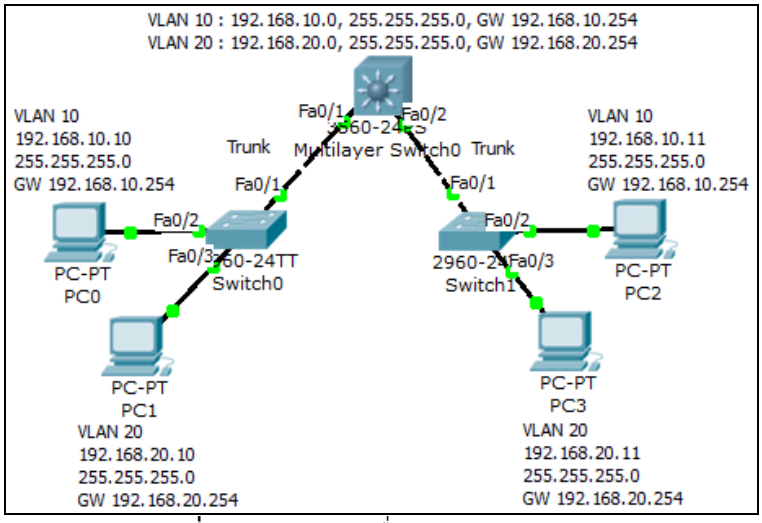
1. ทดสอบว่าเครื่อง Switch1, Switch2 ได้รับข้อมูลของ vlan มาเรียบร้อยแล้ว โดยใช้คำสั่ง show vlan
2. ให้ทำการทดสอบการเชื่อมต่อโดย การ ping ภายใน vlan เดียวกัน คือ PC0 และ PC3, PC2 และ PC4, PC3 และ PC5 ต้องสามารถเชื่อมต่อกันได้ แต่ถ้า ping ข้าม vlan จะไม่สามารถ ping กันได้
3. ทดสอบโดยการทำงานของ VTP โดยใช้คำสั่ง show vtp status

Scenario 21: การคอนฟิก Switch L3 to L2 InterVLANs

คำอธิบาย :

จากที่กล่าวมาแล้วใน Scenario ที่ 20 ว่าเมื่อต้องการเชื่อม vlan ที่สร้างใน switch L2 หลายๆ vlan ให้สามารถเชื่อมต่อกันได้ต้องอาศัย อุปกรณ์ที่ทำงานในระดับเลเยอร์ที่ 3 ใน Scenario นี้จึงขอแนะนำการเชื่อม vlan ใน switch L2 ให้สามารถคุยกันข้าม vlan ได้

แผนผังการเชื่อมต่อ :



รูปที่ 12.82 ผังการเชื่อมต่อ scenario 21

จากรูป switch L2 จะสร้าง vlan ไว้ 2 vlan คือ vlan 10 และ 20 โดยพอร์ต Fa0/1 จะทำหน้าที่เป็น Trunk เพื่อเชื่อมไปยัง switch L3 ใน switch L3 จะสร้าง vlan 10 และ 20 เช่นเดียวกับ switch L2 เช่น เดียวกัน แต่มีการกำหนดหมายเลข IP ให้กับ vlan 10 คือ 192.168.10.254, Subnet Mask คือ 255.255.255.0 และ vlan 20 กำหนดเป็น 192.168.20.254, 255.255.255.0

เพื่อรองรับการเชื่อมต่อที่มาจาก switch L2 (ถ้า switch L2 มีจำนวนมากๆ อาจจะประยุกต์เอาวิธีการแบบ vtp มาใช้แทนการสร้าง vlan แบบ manual ก็ได้) สำหรับเครื่อง PC ให้กำหนดหมายเลข IP, Subnet Mask และ Gateway (ใช้อักษรย่อคือ GW) ตามรูป

ขั้นตอนการคอนฟิก :

1. บนเครื่อง Switch0 และ Switch1 (Switch L2) ให้คอนฟิกเหมือนกันทุกประการ โดยสร้าง vlan 10, 20 บน Switch database

```
! สร้าง vlan 10 และ 20
Layer2-Switch#configure terminal <ENTER>
Layer2-Switch(config)#vlan 10 <ENTER>
Layer2-Switch(config-vlan)#end <ENTER>
Layer2-Switch(config)#vlan 20 <ENTER>
Layer2-Switch(config-vlan)#end <ENTER>

!  กำหนดให้พอร์ต Fa0/2 เป็นสมาชิก vlan 10
Layer2-Switch(config)#interface fastethernet0/2 <ENTER>
Layer2-Switch(config-if)#switchport mode access <ENTER>
Layer2-Switch(config-if)#switchport access vlan 10 <ENTER>
Layer2-Switch(config-if)#end <ENTER>

!  กำหนดให้พอร์ต Fa0/3 เป็นสมาชิก vlan 20
Layer2-Switch(config)#interface fastethernet0/3 <ENTER>
Layer2-Switch(config-if)#switchport mode access <ENTER>
Layer2-Switch(config-if)#switchport access vlan 20 <ENTER>
Layer2-Switch(config-if)#end <ENTER>

!  สร้าง Trunk บน Fa0/1
Layer2-Switch(config)#interface fastethernet0/1 <ENTER>
Layer2-Switch(config-if)#switchport mode trunk <ENTER>
Layer2-Switch(config-if)#switchport trunk encapsulation dot1q <ENTER>
Layer2-Switch(config-if)#end <ENTER>
```

2. บนเครื่อง Multilayer Switch (Switch L3) สร้าง vlan 10, 20 และ Trunk

```
!  สร้าง vlan database คือ vlan 10, 20
```

```
Layer3-Switch#configure terminal <ENTER>
Layer3-Switch(config)#vlan 10 <ENTER>
Layer3-Switch(config-vlan)#end <ENTER>
Layer3-Switch(config)#vlan 20 <ENTER>
Layer3-Switch(config-vlan)#end <ENTER>

!  สร้าง Trunk พอร์ต บน Fa0/1 และ Fa0/2
Layer3-Switch(config)#interface fastethernet0/1 <ENTER>
Layer3-Switch(config-if)#switchport mode trunk <ENTER>
Layer3-Switch(config-if)#switchport trunk encapsulation dot1q <ENTER>
Layer3-Switch(config-if)#end <ENTER>
Layer3-Switch(config)#interface fastethernet0/2 <ENTER>
Layer3-Switch(config-if)#switchport mode trunk <ENTER>
Layer3-Switch(config-if)#switchport trunk encapsulation dot1q <ENTER>
Layer3-Switch(config-if)#end <ENTER>

!  คอนฟิก IP, Subnet Mask และ Gateway ให้ vlan 10, 20
Layer3-Switch(config)#interface vlan10 <ENTER>
Layer3-Switch(config-if)#ip address 192.168.10.254 255.255.255.0 <ENTER>
Layer3-Switch(config-if)#no shut <ENTER>
Layer3-Switch(config)#interface vlan20 <ENTER>
Layer3-Switch(config-if)#ip address 192168.20.254 255.255.255.0 <ENTER>
Layer3-Switch(config-if)#no shut <ENTER>
```

3. ที่เครื่อง PC0, PC1, PC2, PC3 ให้กำหนด IP, Subnet Mask และ Gateway ดังต่อไปนี้

```
PC0 : IP=192.168.10.10, Subnet Mask=255.255.255.0, GW=192.168.10.254
PC1 : IP=192.168.20.10, Subnet Mask=255.255.255.0, GW=192.168.20.254
PC2 : IP=192.168.10.11, Subnet Mask=255.255.255.0, GW=192.168.10.254
PC3 : IP=192.168.20.11, Subnet Mask=255.255.255.0, GW=192.168.20.254
```

การทดสอบ :

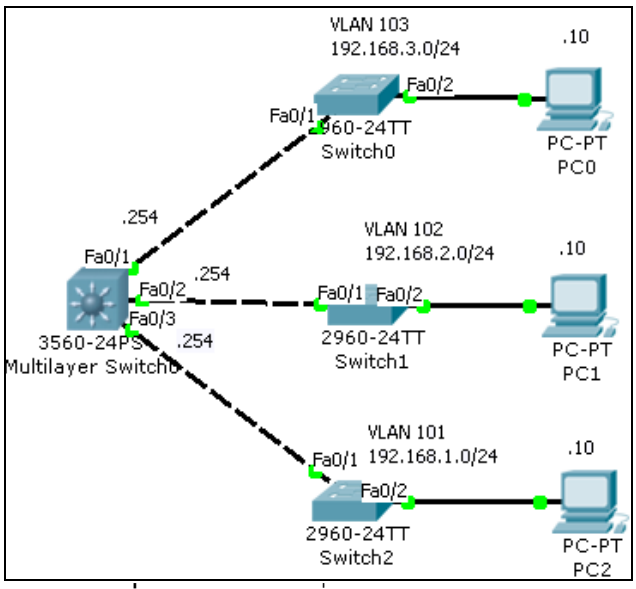
1. ทดสอบโดยการ ping จาก PC0 ไปยังเครื่อง PC1, PC2 และ PC3 ทุกๆ เครื่องต้องสามารถสื่อสารกันได้ทั้งหมด

Scenario 22: การคอนฟิก Switch L3 InterVLANs (Route VLAN)

คำอธิบาย :

ใน Scenario นี้จะกล่าวถึงการสร้าง vlan บนอุปกรณ์เลเยอร์ 3 ซึ่งอุปกรณ์ดังกล่าวมีคุณสมบัติการเราต์ระหว่าง vlan อยู่แล้ว โดยไม่จำเป็นต้องใช้ Trunk แต่อุปกรณ์เลเยอร์ที่ 2 ที่นำมาเชื่อมต่อกับอุปกรณ์เลเยอร์ที่ 3 ควรจะทำหน้าที่เป็นแค่ access เท่านั้น ไม่ควรทำ vlan ที่สวิชต์ L2 เพิ่มอีก (ใช้เฉพาะ vlan 1 ที่เป็นค่า default เท่านั้น) สรุปคืออุปกรณ์ในเลเยอร์ 2 จะทำหน้าที่เป็นแค่อุปกรณ์ที่เชื่อมอุปกรณ์ปลายทาง เช่น PC, printer เป็นต้น เข้าสู่เครือข่ายเท่านั้น

แผนผังการเชื่อมต่อ :



รูปที่ 12.83 ผังการเชื่อมต่อ scenario 22

จากรูป switch L3 (3560) จะสร้าง vlan ไว้ 3 vlan คือ vlan 101 (192.168.1.0/24 และ gateway คือ .254), vlan 102 (192.168.2.0/24), vlan 103 (192.168.3.0/24) และให้ port Fa0/1 เป็นสมาชิก vlan 101, Fa0/2 เป็นสมาชิก vlan 102 และ Fa0/3 เป็นสมาชิก vlan 103 ตามลำดับ โดยมีสวิชต์ L2 ทำหน้าที่เชื่อมต่ออุปกรณ์ปลายทางเข้าสู่เครือข่ายผ่าน vlan 1 เครื่องลูกข่ายจะกำหนดให้เป็นสมาชิกของแต่ละ vlan โดยใช้หมายเลขไอพีคือ .10 (สำหรับเครื่องหมายเลขไอพี .10 ที่อยู่บน vlan 101 จะกำหนดค่าดังนี้ IP=192.168.1.10, subnet=255.255.255.0, gateway=192.168.1.254) ตามรูป

ขั้นตอนการคอนฟิก :

1. บนเครื่องสวิชต์ L3 (3560) สร้าง vlan 101, 102, 103 และกำหนดไอพีให้กับแต่ละ vlan ดังนี้

! สร้าง vlan 101, 102, 103

Switch(config)#**interface vlan 101** <ENTER> โหมด config ให้สร้าง vlan 101

Switch(config-if)#**ip address 192.168.1.254 255.255.255.0** <ENTER> กำหนดไอพีสำหรับ vlan 101 เพื่อเป็น gateway ให้กับเครื่องลูกข่าย

Switch(config-if)#**inte vlan 102** <ENTER>

Switch(config-if)#**ip address 192.168.2.254 255.255.255.0** <ENTER>

Switch(config-if)#**inte vlan 103** <ENTER>

Switch(config-if)#**ip address 192.168.3.254 255.255.255.0** <ENTER>

สร้าง vlan 102, 103 พร้อมกับกำหนดไอพีให้แต่ละ vlan ตามลำดับ

! กำหนด port ให้เป็นสมาชิกของแต่ละ vlan

Switch(config)#**interface fastEthernet 0/1** <ENTER> โหมด config เข้าสู่ port fa0/1

Switch(config-if)#**switchport access vlan 103** <ENTER> กำหนดให้ fa0/1 เป็นสมาชิก vlan 103

Switch(config-if)#**exit** <ENTER> ออกไปยังโหมด config

Switch(config)#**interface fastEthernet 0/2** <ENTER> เข้าสู่ fa0/2

Switch(config-if)#**switchport access vlan 102** <ENTER> กำหนดเป็นสมาชิก vlan 102

Switch(config-if)#**exit** <ENTER> ออกไปยังโหมด config

Switch(config)#**interface fastEthernet 0/3** <ENTER> เข้าสู่ fa0/3

Switch(config-if)#**switchport access vlan 103** <ENTER> กำหนดเป็นสมาชิก vlan 103

เสร็จการคอนฟิกที่อุปกรณ์สวิชต์ L3

1. สำหรับบนสวิชต์ L2 ไม่ต้องปรับแต่งคอนฟิกใดๆ ทั้งสิ้น ต่อไปให้กำหนดไอพีให้กับเครื่อง PC0, PC1 และ PC3 ตามลำดับดังนี้

   PC0 (vlan 103):

   IP=192.168.3.10, subnet=255.255.255.0, gateway=192.168.3.254

   PC1 (vlan 102):

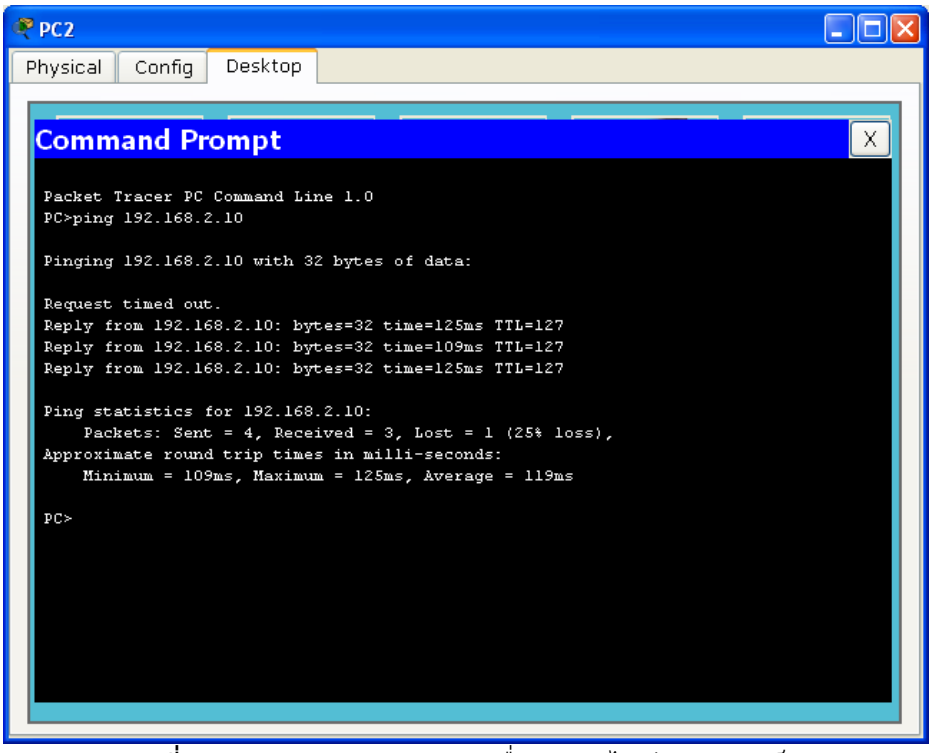
   IP=192.168.2.10, subnet=255.255.255.0, gateway=192.168.2.254

   PC2 (vlan 101):

   IP=192.168.1.10, subnet=255.255.255.0, gateway=192.168.1.254

การทดสอบ :

ทดสอบโดยการ ping จาก PC0 ไปยังเครื่อง PC1, PC2 และ ping กลับทิศทางกันคือ PC2 ไปยัง PC3, PC1 เป็นต้น จะต้อง ping ได้ทุกๆ จุดจึงจะถือได้ว่ากระบวนการคอนฟิกทั้งหมดสำเร็จ ดังรูปที่ 12.84



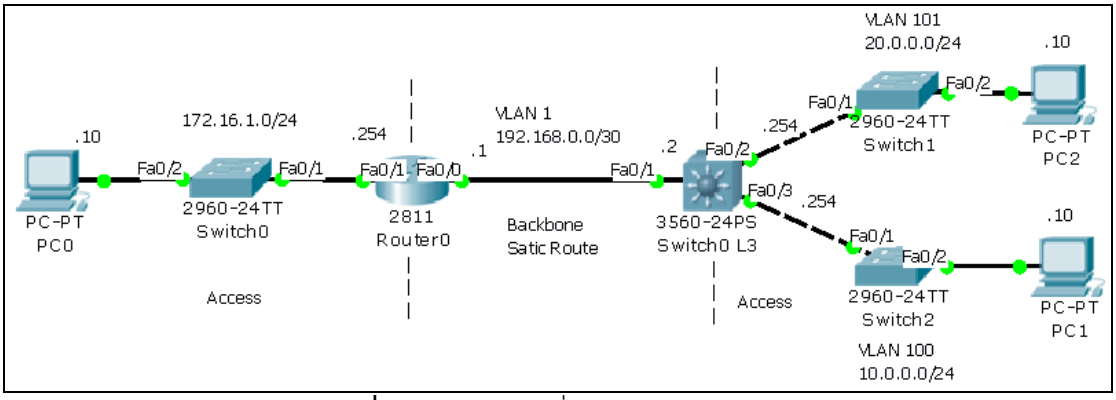
รูปที่ 12.84 ทดสอบ ping จากเครื่อง PC2 ไปยัง PC1 สำเร็จ

Scenario 23: การคอนฟิก Switch L3 กับ Router และ Static Routing

คำอธิบาย :

ความสามารถของเราเตอร์และสวิชต์ในปัจจุบันใกล้เคียงกันมาก แทบจะแยกไม่ออกว่าต่างกันอย่างไร มีอยู่จุดหนึ่งที่เห็นได้ชัดคือ เราเตอร์นั้นจะเชื่อมเครือข่ายที่แตกต่างกันให้สามารถสื่อสารกันได้ เช่น Ethernet topology กับ ATM หรือ Frame-relay เป็นต้น แต่สวิชต์มีสามารถเชื่อมต่อเครือข่ายที่มีโทโพโลยีเดียวกันเข้าด้วยกัน เช่น Ethernet กับ Ethernet เป็นต้น (แต่ปัจจุบันสวิชต์เริ่มจะมีความสามารถเชื่อมต่อเครือข่ายต่างๆ เข้ากันได้เหมือนเราเตอร์แล้ว ซึ่งเป็นสวิตช์รุ่นใหญ่ๆ เช่น 7000 series เป็นต้น)

แผนผังการเชื่อมต่อ :



รูปที่ 12.85 ผังการเชื่อมต่อ scenario 23

ขั้นตอนการคอนฟิก :

1. บนเครื่องเราเตอร์ (2811) กำหนดไอพีสำหรับใช้เป็น gateway ให้ fa0/0 สำหรับเชื่อมกับสวิชต์ 3560 และ fa0/1 สำหรับเชื่อมต่อกับเน็ตเวิร์ค 172.16.1.0 ตามไดอะแกรมด้านบน ดังนี้

บนเราเตอร์ 2811

Router(config)#**interface fastEthernet 0/0** <ENTER> เข้าสู่ fa0/0 ในโหมด config

Router(config-if)#**ip address 192.168.0.1 255.255.255.252** <ENTER> กำหนดไอพีให้ fa0/0 เป็น .1 และทำ subnet .252 ซึ่งจะใช้ไอพีได้เพียง 2 ไอพีเท่านั้น (ไม่รวม Network ID และ Broadcast IP)

Router(config-if)#**no shutdown** <ENTER> สั่งให้ fa0/0 ทำงาน เพราะอินเทอร์เฟสของเราเตอร์โดยปกติจะปิดการทำงานไว้เสมอ แต่สวิชต์จะเปิดไว้เสมอ

Router(config-if)#**exit** <ENTER> ออกไปสู่โหมด config

Router(config)#**inte fa0/1** <ENTER> เข้าสู่อินเทอร์เฟส fa0/1 เพื่อกำหนด gateway ให้เครื่องลูกข่ายของเราเตอร์

Router(config-if)#ip **address 172.16.1.254 255.255.255.0** <ENTER> กำหนด gateway เป็น .254 และ subnet /24 หรือ 255.255.255.0 แม้ว่าไอพีเป็น class B ก็ตาม แต่เมื่อทำ subnet ดังกล่าวแล้ว สามารถใช้เครื่องลูกข่ายได้เพียง 256 ไอพีเท่านั้น

Router(config-if)#**no shutdown** <ENTER>

Router(config)#**end** <ENTER> ออกสู่โหมด admin

Router#**wr** <ENTER> บันทึกข้อมูลการคอนฟิกบนเราเตอร์

2. บนสวิชต์ L3 (3560) ต้องสร้าง vlan 100 และ vlan 101 พร้อมกับกำหนดไอพีให้กับ vlan และไอพีที่ใช้เชื่อมต่อกับเราเตอร์ ตามลำดับดังนี้

บนสวิชต์ L3 (3560)

Switch(config)#**interface vlan 1** <ENTER> เข้าสู่ vlan 1 ในโหมด config

Switch(config-if)#**ip address 192.168.0.2 255.255.255.252** <ENTER> กำหนด gateway ที่

เชื่อมต่อกับเราเตอร์เป็น .2 subnet คือ /30 หรือ 255.255.255.252

Switch(config-if)#**no shutdown** <ENTER> สั่งให้อินเทอร์เฟสทำงาน (เพื่อความมั่นใจว่าอินเทอร์เฟสดังกล่าวทำงานแล้ว)

! ทดสอบโดยการ ping ไปยัง gateway ของเราเตอร์

Switch#**ping 192.168.0.1** <ENTER>

! สร้าง vlan 100 และ 101 เพื่อใช้เป็น gateway ให้เน็ตเวิร์ค 10.0.0.0 และ 20.0.0.0

Switch(config)#**interface vlan 100** <ENTER>

witch(config-if)#**ip address 10.0.0.254 255.255.255.0** <ENTER>

Switch(config-if)#**inte vlan 101** <ENTER>

Switch(config-if)#**ip address 20.0.0.254 255.255.255.0** <ENTER>

Switch(config-if)#**exit** <ENTER>

Switch(config)#**interface fa0/2** <ENTER> กำหนดให้ fa0/2 เป็นสมาชิก vlan 101

Switch(config-if)#**switchport access vlan 101** <ENTER>

Switch(config-if)#**inte fa0/3 <ENTER> กำหนดให้ fa0/3** เป็นสมาชิก vlan 100

Switch(config-if)#**switchport access vlan 100** <ENTER>

3. กำหนดหมายเลขไอพีให้กับเครื่อง PC ทั้งหมดตาม diagram ด้านบน ดังนี้

PC0 (ต่อหลังเราเตอร์):

IP=172.16.1.10, subnet=255.255.255.0, gateway=172.16.1.254

PC1 (ต่อหลังสวิชต์ L3 vlan 100):

IP=10.0.0.10, subnet=255.255.255.0, gateway=10.0.0.254

PC2 (ต่อหลังสวิชต์ L3 vlan 101):

IP=20.0.0.10, subnet=255.255.255.0, gateway=20.0.0.254

การทดสอบ :

1. ทดสอบโดยการ ping gateway ของเน็ตเวิร์คที่เครื่อง PC ดังกล่าวต่อเชื่อมอยู่ เช่น PC0 ทดลอง ping ไปยังไอพี 172.16.1.254 และ 192.168.0.1 ซึ่งต้องมีการตอบสนองกลับมาจึงถือว่าถูกต้อง แต่ในขั้นตอนนี้จะยังไม่สามารถ ping ไปยังไอพีอื่นๆ เช่น 192.168.1.2 หรือ 10.0.0.X และ 20.0.0.X เนื่องจากยังไม่มีการทำ routing ให้กับเครือข่ายดังกล่าว
2. การกำหนด routing เป็นการบอกให้อุปกรณ์บนเครือข่ายทราบว่าข้อมูลหรือแพ็คเก็ตที่ต้องการส่งไปยังเครือข่ายอื่นๆ ควรจะไปทางไหน อย่างไร ซึ่งมี 2 แบบ คือ static route และ dynamic route เมื่อกล่าวโดยย่อ static route ผู้ดูแล

ระบบจะเป็นผู้กำหนดเส้นทางเองทั้งหมด ส่วน dynamic route นั้นโปรแกรม routing ที่ฝังมากับเราเตอร์จะเป็นผู้หาเส้นทางเอง (แต่เบื้องต้นผู้ดูแลระบบจะต้องมีการคอนฟิกวิธีการ routing ให้แกรมรู้ก่อนเสมอ ซึ่งจะกล่าวในscenario ต่อๆ ไป)

การกำหนด Static route บนเราเตอร์

เมื่อสังเกตุจาก Diagram ข้างบน เมื่อต้องการให้แพ็คเก็ตที่อยู่หลังเราเตอร์ส่งไปยังเน็ตเวิร์ค 10.0.0.0 หรือ 20.0.0.0 เราจะต้องบอกให้เราเตอร์ทราบว่าจะโยนข้อมูลไปอย่างไร ในที่นี้จะต้องกำหนดให้โยนไปที่ไอพีของสวิชต์ L3 ซึ่งอยู่ตรงกันข้ามกับเราเตอร์ ดังนี้

Router(config)#**ip route 10.0.0.0 255.255.255.0 192.168.0.2** <ENTER> ในโหมด config กำหนดให้เราเตอร์โยนข้อมูลไปยังไอพี 192.168.0.2 ของสวิชต์ L3 เมื่อมีแพ็คเก็ตใดๆ ต้องการส่งข้อมูลไปยังเน็ตเวิร์ค 10.0.0.0 หรือ 20.0.0.0

Router(config)#**ip route 20.0.0.0 255.255.255.0 192.168.0.2** <ENTER>

การกำหนด Static route บนสวิชต์ L3

ในทางกลับกันถ้าต้องการส่งข้อมูลจาก 10.0.0.0 หรือ 20.0.0.0 ไปยังเน็ตเวิร์ค 172.16.1.0 จะต้องโยนข้อมูลไปยังไอพีของเราเตอร์ฝั่งตรงกันข้าม และอย่างลืมว่าการคอนฟิกด้วยวิธีการ Static จะต้องทำ route ทั้ง 2 ข้างให้ครบก่อนข้อมูลจึงจะเดินทางได้อย่างสมบูรณ์

Router(config)#**ip route 172.16.1.0 255.255.255.0 192.168.0.1** <ENTER> ในโหมด config กำหนดให้เราเตอร์โยนข้อมูลไปยังไอพี 192.168.0.1 ของเราเตอร์ เมื่อมีแพ็คเก็ตใดๆ ต้องการส่งข้อมูลไปยังเน็ตเวิร์ค 172.16.1.0

3. ทดสอบโดยการ ping จาก PC0 ไปยัง PC1 และ PC2 ต้องมีการตอบสนอง ถ้าไม่มีให้กลับไปตรวจสอบกับ diagram อีกครั้งและ ตรวจสอบคอนฟิกบนเราเตอร์ สวิชต์ว่าถูกต้องหรือยัง (ใช้คำสั่ง show running-config)

หมายเหตุ 1: ถ้าพิมพ์คำสั่งใดๆ ในเราเตอร์หรือสวิชต์ผิด และต้องการแก้ไขหรือลบออกให้ใช้คำว่า no และตามด้วยคำสั่งดังกล่าว ในโหมดเดิม เช่น

Router(config)#**ip route 172.16.1.0 255.255.255.0 192.168.0.1 คำสั่งเก่าที่ผิด**

Router(config)#**no ip route 172.16.1.0 255.255.255.0 192.168.0.1 ลบคำสั่งเก่าที่ผิดออก**

หมายเหตุ 2: ในกรณีที่เริ่มการทำงานของเราเตอร์ใหม่และยังไม่มีการคอนฟิกใดๆ เราเตอร์จะแสดงเมนูดังนี้คือ

--- System Configuration Dialog ---

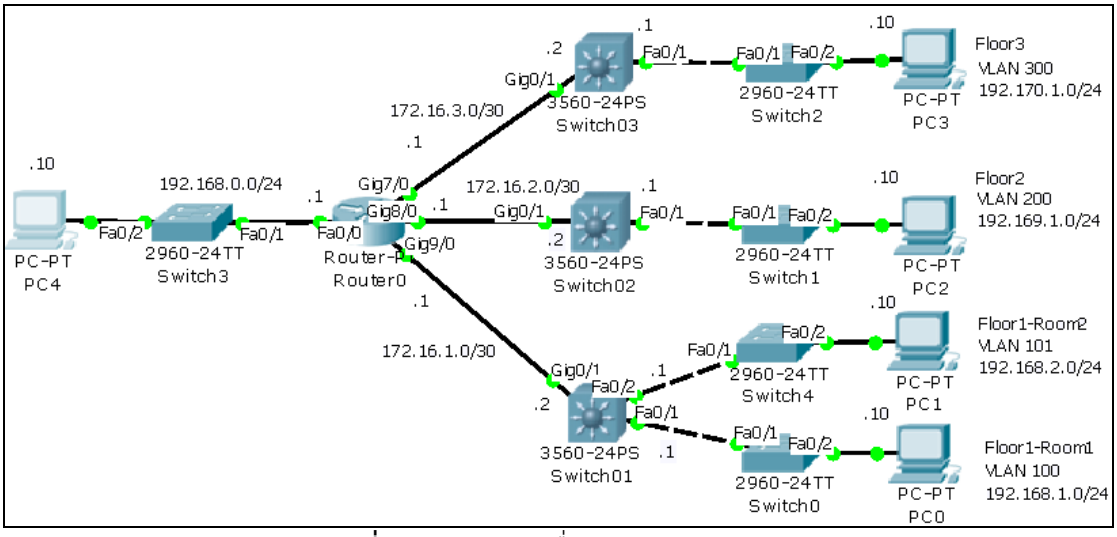
Continue with configuration dialog? [yes/no]: no ← ให้เลือก no <ENTER>

Scenario 24: การคอนฟิกให้ Router ควบคุมสวิชต์ L3 หลายๆ ตัว

คำอธิบาย :

ใน Scenario นี้จะแสดงการใช้เราเตอร์ควบคุมการทำงานของสวิชต์ L3 หลายๆ ตัวเข้าไว้ด้วยกัน คล้ายกับมีสำนักงานอยู่ 1 ตึกและประกอบไปด้วย 3 ชั้น แต่ละชั้นมีแต่ละแผนกอยู่ ดังนั้นตามหลักการคือ ควรจะให้ L3 ควบคุม vlan ของแต่ละห้องในแต่ละชั้น จากนั้นให้เราเตอร์หรือสวิชต์ L3 คุณภาพสูงควบคุม L2 ในแต่ละชั้นอีกทีหนึ่ง ดัง Diagram

แผนผังการเชื่อมต่อ :



รูปที่ 12.86 ผังการเชื่อมต่อ scenario 24

ขั้นตอนการเชื่อมต่อ :

ใน Diagram ข้างต้นการเชื่อมต่อระหว่างเราเตอร์ (ควรใช้ Router-PT สำหรับทดสอบใน scenario นี้) และสวิชต์ L3 ในแต่ละชั้นจะเชื่อมด้วย Gigabit Ethernet (Gig) ซึ่งผู้ใช้งานจำเป็นต้องติดตั้ง Module เพิ่มที่เราเตอร์คือ The single-port Cisco Gigabit Ethernet Network Module (part number PT-ROUTER-NM-1CGE) จำนวน 3 พอร์ตบนเราเตอร์ (สำหรับขั้นตอนการติดตั้งให้กลับไปดูในบทแรกๆ เรื่องของเพิ่มถอนอินเทอร์เฟส)

ขั้นตอนการคอนฟิก :

1. บนเครื่องเราเตอร์ (Router-PT) จะต้องคอนฟิก ซึ่งประกอบไปด้วย

Interface	IP/Subnet
Fa0/0	192.168.0.1/255.255.255.0 หรือ /24
Gig9/0	172.16.1.1/255.255.255.252 หรือ /30
Gig8/0	172.16.2.1/255.255.255.252 หรือ /30
Gig7/0	172.16.3.1/255.255.255.252 หรือ /30

ด้วยคำสั่งดังนี้

```
! อินเทอร์เฟส Fa0/0
Router(config)#interface fastEthernet 0/0 <ENTER>
Router(config-if)#ip address 192.168.0.1 255.255.255.0 <ENTER>
Router(config-if)#no shutdown <ENTER>

! บนอินเทอร์เฟส Gig9/0
Router(config)#interface gigabitEthernet 9/0 <ENTER>
Router(config-if)#ip address 172.16.1.1 255.255.255.252 <ENTER> subnet = /30
Router(config-if)#no shutdown <ENTER>

! บนอินเทอร์เฟส Gig8/0
Router(config)#interface gigabitEthernet 8/0 <ENTER>
Router(config-if)#ip address 172.16.2.1 255.255.255.252 <ENTER> subnet = /30
Router(config-if)#no shutdown <ENTER>

! บนอินเทอร์เฟส Gig7/0
Router(config)#interface gigabitEthernet 7/0 <ENTER>
Router(config-if)#ip address 172.16.3.1 255.255.255.252 <ENTER> subnet = /30
Router(config-if)#no shutdown <ENTER>
```

2. บนสวิชต์ L3 (Switch01) ชั้นที่ 1 ต้องสร้าง vlan อย่างน้อย 2 vlan คือ vlan 100 และ 101 และกำหนดไอพีให้แต่ละ vlan ดังนี้

Interface	Vlan name	IP/Subnet
Gig0/1	Vlan 1	172.16.1.2/255.255.255.252 หรือ /30
Fa0/1	Vlan 100	192.168.1.1/255.255.255.0 หรือ /24
Fa0/2	Vlan 101	192.168.2.1/255.255.255.0 หรือ /24

ด้วยคำสั่งดังนี้ (สำหรับผู้เริ่มต้นสังเกตให้ดีว่าการคอนฟิกจะเปลี่ยนโหมดไปตาม หน้าที่เช่น โหมด admin จะเป็นสัญลักษณ์ #, โหมด config เป็น (config)#, โหมดอินเทอร์เฟส เป็น (config-if)# เป็นต้นของให้ระวังด้วย ถ้ายังไม่เข้าใจให้อ่านเพิ่มเติมได้จาก หนังสือ network simulation เล่ม 1 ของผู้เขียนควบคู่ไปด้วย หรือดูจากหนังสือท้ายเล่มประกอบ)

```
! อินเทอร์เฟส vlan 1
```

Switch01(config)#**interface vlan 1** <ENTER>

Switch01(config-if)#**ip address 172.16.1.2 255.255.255.252** <ENTER> กำหนดไอพีให้ vlan 1 เพื่อใช้เชื่อมต่อกับเราเตอร์ (ถามว่าทำไมต้องใช้ vlan 1 ในการเชื่อมต่อ ไม่เห็นเราเตอร์ต้องทำงานกับ vlan เลย ขอตอบว่า คุณสมบัติของสวิชต์ L3 คือ vlan ดังนั้นการเชื่อมต่อทุกๆ อย่างสวิชต์ L3 จะต้องมี vlan เข้ามายุ่งเกี่ยวเสมอ ส่วนทำไมต้องใช้ vlan 1 ตอบว่า ไม่จำเป็นก็ได้ จะสร้าง vlan ใหม่ขึ้นมาทดแทน vlan 1 ก็ได้ แต่ในเบื้องต้นเป็นที่ทราบกันดีว่าทุกๆ พอร์ตจะอยู่ที่ vlan 1 เสมอ จึงสามารถใช้งานพอร์ตต่างๆ ได้ทันที โดยไม่จำเป็นต้องย้ายพอร์ตไป vlan ใหม่ที่สร้างขึ้น)

Switch01(config-if)#**no shutdown** <ENTER>

ไม่ต้องย้ายพอร์ต Gig0/1 มาที่ vlan 1 เพราะอยู่ที่ vlan 1 โดย default อยู่แล้ว

! สร้าง vlan 100 และ 101

Switch01(config)#**interface vlan 100** <ENTER>

Switch01(config-if)#**ip address 192.168.1.1 255.255.255.0** <ENTER>

Switch01(config-if)#**inte vlan 101** <ENTER>

Switch01(config-if)#**ip address 192.168.2.1 255.255.255.0** <ENTER>

! ย้ายพอร์ต Fa0/1 มาเป็นสมาชิกของ vlan 100 และ Fa0/2 มาเป็นสมาชิกของ vlan 101

Switch01(config)#**interface fastEthernet 0/1** <ENTER>

Switch01(config-if)#**switchport access vlan 100** <ENTER>

Switch01(config-if)#**inte fa0/2** <ENTER>

Switch01(config-if)#**switchport access vlan 101** <ENTER>

3. บนสวิชต์ L3 (Switch02) ชั้นที่ 2 ต้องสร้าง vlan 200 และกำหนดค่าดังนี้

Interface	Vlan name	IP/Subnet
Gig0/1	Vlan 1	172.16.2.2/255.255.255.252 หรือ /30
Fa0/1	Vlan 200	192.169.1.1/255.255.255.0 หรือ /24

! อินเทอร์เฟส vlan 1

Switch02(config)#**interface vlan 1** <ENTER>

Switch02(config-if)#**ip address 172.16.2.2 255.255.255.252** <ENTER>

Switch02(config-if)#**no shutdown** <ENTER> !!! อย่าลืมต้องสั่งให้ vlan 1 ทำงานด้วย

! สร้าง vlan 200

```
Switch02(config)#interface vlan 100 <ENTER>
Switch02(config-if)#ip address 192.169.1.1 255.255.255.0 <ENTER>

! ย้ายพอร์ต Fa0/1 มาเป็นสมาชิกของ vlan 200
Switch02(config)#interface fastEthernet 0/1 <ENTER>
Switch02(config-if)#switchport access vlan 200 <ENTER>
```

4. บนสวิชต์ L3 (Switch03) ชั้นที่ 3 ต้องสร้าง vlan 300 และกำหนดค่าดังนี้

Interface	Vlan name	IP/Subnet
Gig0/1	Vlan 1	172.16.3.2/255.255.255.252 หรือ /30
Fa0/1	Vlan 300	192.170.1.1/255.255.255.0 หรือ /24

```
! อินเทอร์เฟส vlan 1
Switch03(config)#interface vlan 1 <ENTER>
Switch03(config-if)#ip address 172.16.3.2 255.255.255.252 <ENTER>
Switch03(config-if)#no shutdown <ENTER> !!! อย่าลืมต้องสั่งให้ vlan 1 ทำงานด้วย

! สร้าง vlan 300
Switch03(config)#interface vlan 300 <ENTER>
Switch03(config-if)#ip address 192.170.1.1 255.255.255.0 <ENTER>

! ย้ายพอร์ต Fa0/1 มาเป็นสมาชิกของ vlan 300
Switch03(config)#interface fastEthernet 0/1 <ENTER>
Switch03(config-if)#switchport access vlan 300 <ENTER>
```

5. กำหนดไอพีของเครื่อง PC ตาม Diagram ดังนี้

เครื่อง PC	Vlan name	IP/Subnet
PC0	Vlan 100	192.168.1.10/255.255.255.0
PC1	Vlan 101	192.168.2.10/255.255.255.0
PC2	Vlan 200	192.169.1.10/255.255.255.0
PC3	Vlan 300	192.170.0.10/255.255.255.0
PC4	No vlan	192.168.0.10/255.255.255.0

6. ทำการกำหนด Static Route บนเราเตอร์ดังต่อไปนี้

Router(config)#**ip route 192.168.1.0 255.255.255.0 172.16.1.2** <ENTER> สำหรับเราต์ไปยังเน็ตเวิร์ค 192.168.1.0

Router(config)#**ip route 192.168.2.0 255.255.255.0 172.16.1.2** <ENTER> เราต์ไปยังเน็ตเวิร์ค 192.168.2.0

Router(config)#**ip route 192.169.1.0 255.255.255.0 172.16.2.2** <ENTER> เราต์ไปยังเน็ตเวิร์ค 192.169.1.0

Router(config)#**ip route 192.170.1.0 255.255.255.0 172.16.3.2** <ENTER> เราต์ไปยังเน็ตเวิร์ค 192.170.1.0

7. ทำการกำหนด Static Route บนสวิชต์ Switch01 ดังต่อไปนี้

Switch01(config)#**ip route 192.168.0.0 255.255.255.0 172.16.1.1** <ENTER>

Switch01(config)#**ip route 192.169.1.0 255.255.255.0 172.16.1.1** <ENTER>

Switch01(config)#**ip route 192.170.1.0 255.255.255.0 172.16.1.1** <ENTER>

8. ทำการกำหนด Static Route บนสวิชต์ Switch02 ดังต่อไปนี้

Switch02(config)#**ip route 192.168.0.0 255.255.255.0 172.16.2.1** <ENTER>

Switch02(config)#**ip route 192.168.1.0 255.255.255.0 172.16.2.1** <ENTER>

Switch02(config)#**ip route 192.168.2.0 255.255.255.0 172.16.2.1** <ENTER>

Switch02(config)#**ip route 192.170.1.0 255.255.255.0 172.16.2.1** <ENTER>

9. ทำการกำหนด Static Route บนสวิชต์ Switch03 ดังต่อไปนี้

Switch03(config)#**ip route 192.168.0.0 255.255.255.0 172.16.3.1** <ENTER>

Switch03(config)#**ip route 192.168.1.0 255.255.255.0 172.16.3.1** <ENTER>

Switch03(config)#**ip route 192.168.2.0 255.255.255.0 172.16.3.1** <ENTER>

Switch03(config)#**ip route 192.169.1.0 255.255.255.0 172.16.3.1** <ENTER>

การทดสอบ :

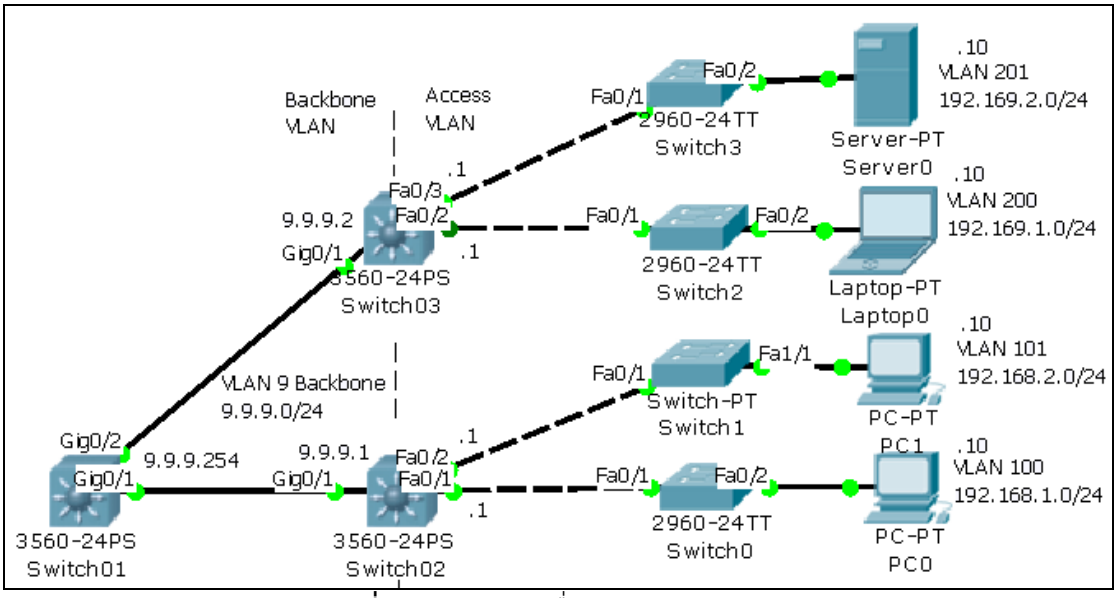
เมื่อการคอนฟิกเสร็จสมบูรณ์เครื่อง PC ทุกๆ เครื่องจะต้องสามารถ ping กันได้ทั้งหมด

Scenario 25: การคอนฟิกให้สวิชต์ L3 ควบคุมสวิชต์ L3 หลายๆ ตัว

คำอธิบาย :

ใน Scenario นี้จะแสดงการใช้สวิชต์ L3 ควบคุมการทำงานของสวิชต์ L3 หลายๆ ตัวเข้าไว้ด้วยกัน หลักการคล้าย Scenario ที่ 24 แต่ในสถานการณ์นี้ สมมุติว่าไม่มีเราเตอร์ โดยใช้สวิชต์ควบคุมการทำงานแทน

แผนผังการเชื่อมต่อ :



รูปที่ 12.87 ผังการเชื่อมต่อ scenario 25

ขั้นตอนการเชื่อมต่อ :

ใน Diagram ข้างต้น สวิชต์ L3 แต่ละตัวเชื่อมต่อโดยใช้ Vlan Backbone เดียวกัน ซึ่งมีความพิเศษกว่าเครือข่ายอื่นๆ เล็กน้อยคือ อินเทอร์เฟส Gig0/1 และ Gig0/2 จะอยู่ใน Vlan เดียวกัน ซึ่งการเชื่อมต่อที่ผ่านมาจะใช้ Vlan ในส่วนของ Backbone แยกกัน

ขั้นตอนการคอนฟิก :

1. บนสวิชต์ (Switch01) ประกอบไปด้วย

Interface	IP/Subnet	Vlan
Gig0/1	9.9.9.1/255.255.255.0	Vlan 9
Gig0/2	9.9.9.2/255.255.255.0	Vlan 9
Vlan 9	9.9.9.254/255.255.255.0	Vlan 9

ขั้นตอนการคอนฟิกด้วยคำสั่งดังต่อนี้

```
! สร้าง vlan 9
Switch01(config)#interface vlan 9 <ENTER>
```

Switch01(config-if)#**ip address 9.9.9.254 255.255.255.0** <ENTER> กำหนดไอพีสำหรับ vlan 9
Switch01(config-if)#**exit** <ENTER>
Switch01(config)#**interface gigabitEthernet 0/1** <ENTER> ย้าย Gig0/1 จาก vlan 1 ไปยัง vlan 9 ที่สร้างขึ้นใหม่
Switch01(config-if)#**switchport access vlan 9** <ENTER>
Switch01(config-if)#**inte gig0/2** <ENTER> ย้าย Gig0/2 จาก vlan 1 ไปยัง vlan 9
Switch01(config-if)#**switchport access vlan 9** <ENTER>

2. บนสวิชต์ L3 (Switch02) ชั้นที่ 1 ต้องสร้าง vlan 2 vlan คือ vlan 100 และ 101 พร้อมกำหนดไอพีให้แต่ละ vlan ดังนี้

Interface	Vlan name	IP/Subnet
Gig0/1	Vlan 9	9.9.9.1/255.255.255.0
Fa0/1	Vlan 100	192.168.1.1/255.255.255.0
Fa0/2	Vlan 101	192.168.2.1/255.255.255.0

คำสั่งที่คอนฟิกบน Switch02

! สร้างอินเทอร์เฟส vlan 9
Switch02(config)#**interface vlan 9** <ENTER> สร้าง vlan 9 เพื่อใช้เชื่อมเป็น backbone
Switch02(config-if)#**ip address 9.9.9.1 255.255.255.0** <ENTER> กำหนดไอพีของ backbone
Switch02(config-if)#**inte vlan 100** <ENTER> สร้าง vlan 100 เพื่อควบคุมชั้นที่ 1 ห้อง 1
Switch02(config-if)#**ip address 192.168.1.1 255.255.255.0** <ENTER>
Switch02(config-if)#**inte vlan 101** <ENTER> สร้าง vlan 101 เพื่อควบคุมชั้นที่ 1 ห้อง 2
Switch02(config-if)#**ip address 192.168.2.1 255.255.255.0** <ENTER>
Switch02(config-if)#**exit** <ENTER>
Switch02(config)#**interface gigabitEthernet 0/1** <ENTER>
Switch02(config-if)#**switchport access vlan 9** <ENTER> ย้าย gig0/1 เข้า vlan 9
Switch02(config-if)#**inte fa0/1** <ENTER>
Switch02(config-if)#**switchport access vlan 100** <ENTER> ย้าย Fa0/1 เข้า vlan 100
Switch02(config-if)#**inte fa0/2** <ENTER>
Switch02(config-if)#**switchport access vlan 101** <ENTER> ย้าย Fa0/2 เข้า vlan 101

3. บนสวิชต์ L3 (Switch03) ชั้นที่ 2 ต้องสร้าง vlan 2 vlan คือ vlan 200 และ 201 พร้อมกำหนดไอพีให้แต่ละ vlan ดังนี้

Interface	Vlan name	IP/Subnet
Gig0/1	Vlan 9	9.9.9.2/255.255.255.0
Fa0/2	Vlan 200	192.169.1.1/255.255.255.0
Fa0/3	Vlan 201	192.169.2.1/255.255.255.0

คำสั่งที่คอนฟิกบน Switch03

! สร้างอินเทอร์เฟส vlan 9

Switch03(config)#**interface vlan 9** <ENTER> สร้าง vlan 9 เพื่อใช้เชื่อมเป็น backbone

Switch03(config-if)#**ip address 9.9.9.2 255.255.255.0** <ENTER> กำหนดไอพีของ backbone

Switch03(config-if)#**inte vlan 200** <ENTER> สร้าง vlan 200 เพื่อควบคุมชั้นที่ 2 ห้อง 1

Switch03(config-if)#**ip address 192.169.1.1 255.255.255.0** <ENTER>

Switch03(config-if)#**inte vlan 201** <ENTER> สร้าง vlan 201 เพื่อควบคุมชั้นที่ 2 ห้อง 2

Switch03(config-if)#**ip address 192.169.2.1 255.255.255.0** <ENTER>

Switch03(config-if)#**exit** <ENTER>

Switch03(config)#**interface gigabitEthernet 0/1** <ENTER>

Switch03(config-if)#**switchport access vlan 9** <ENTER> ย้าย gig0/1 เข้า vlan 9

Switch03(config-if)#**inte fa0/2** <ENTER>

Switch03(config-if)#**switchport access vlan 200** <ENTER> ย้าย Fa0/2 เข้า vlan 200

Switch(config-if)#**inte fa0/3** <ENTER>

Switch03(config-if)#**switchport access vlan 201** <ENTER> ย้าย Fa0/3 เข้า vlan 201

4. กำหนดไอพีของเครื่อง PC ตาม Diagram ดังนี้

เครื่อง PC	Vlan name	IP/Subnet
PC0	Vlan 100	192.168.1.10/255.255.255.0
PC1	Vlan 101	192.168.2.10/255.255.255.0
Labtop0	Vlan 200	192.169.1.10/255.255.255.0
Server0	Vlan 201	192.169.2.10/255.255.255.0

5. ทำการกำหนด Static Route บนสวิชต์ Switch01 ดังต่อไปนี้

```
Switch01(config)#ip route 192.168.1.0 255.255.255.0 9.9.9.1 <ENTER>
Switch01(config)#ip route 192.168.2.0 255.255.255.0 9.9.9.1 <ENTER>
Switch01(config)#ip route 192.169.1.0 255.255.255.0 9.9.9.2 <ENTER>
Switch01(config)#ip route 192.169.2.0 255.255.255.0 9.9.9.2 <ENTER>
```

7. ทำการกำหนด Static Route บนสวิชต์ Switch02 ดังต่อไปนี้

```
Switch02(config)#ip route 192.169.1.0 255.255.255.0 9.9.9.254 <ENTER>
Switch02(config)#ip route 192.169.2.0 255.255.255.0 9.9.9.254 <ENTER>
```

8. ทำการกำหนด Static Route บนสวิชต์ Switch03 ดังต่อไปนี้

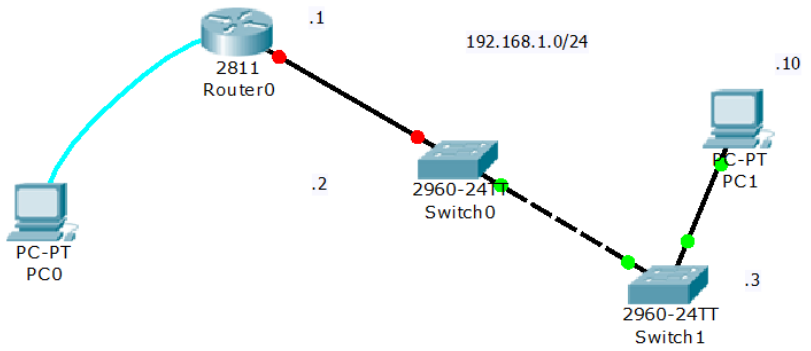
```
Switch03(config)#ip route 192.168.1.0 255.255.255.0 9.9.9.254 <ENTER>
Switch03(config)#ip route 192.168.2.0 255.255.255.0 9.9.9.254 <ENTER>
```

การทดสอบ :

เมื่อการคอนฟิกเสร็จสมบูรณ์เครื่อง PC ทุกๆ เครื่องจะต้องสามารถ ping กันได้ทั้งหมด

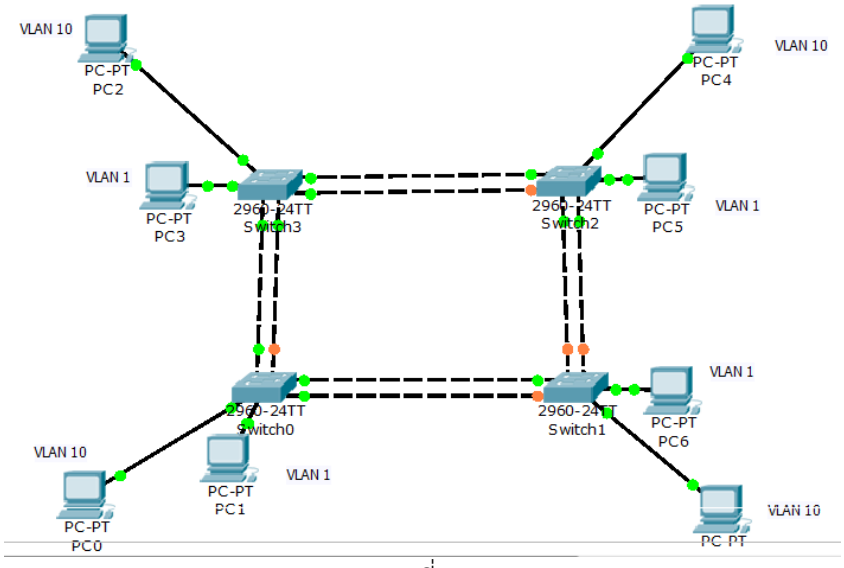
แบบฝึกหัดท้ายบท

1. ให้นิสิตเปิดการใช้งาน console, telnet, tftp, และกำหนดค่า IP address ตามรูป เครื่อง client สามารถ backup & restore config, IOS ได้ และสามารถ ping switch & router ได้ทุกจุด ดังรูปที่ 1



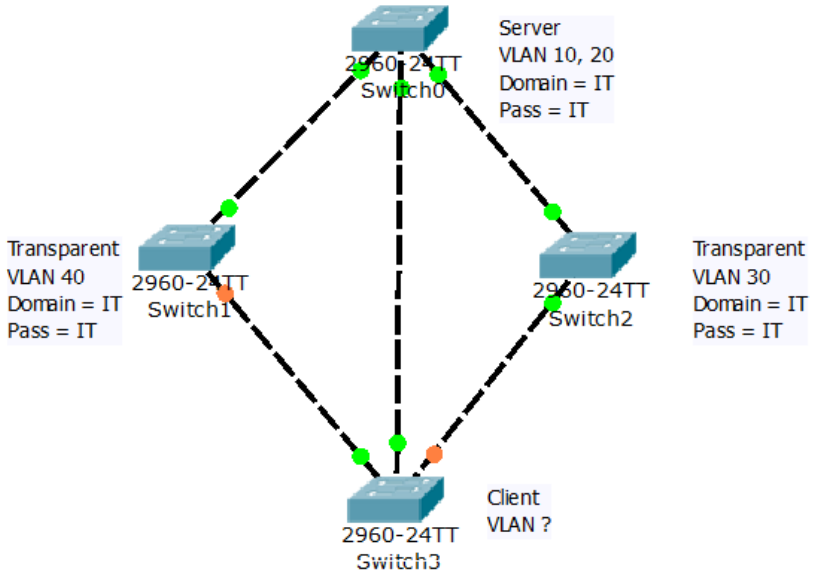
รูปที่ 1

2. ให้นิสิต config เครือข่ายโดยใช้ trunk ตามรูป โดยวงในสุดให้เป็น vlan 1 (192.168.1.0/24) และ วงนอกเป็น vlan 10 (192.168.2.0/24) โดยให้เครื่องภายในวงแลนเดียวกันสามารถ ping กันได้ ดังรูปที่ 2



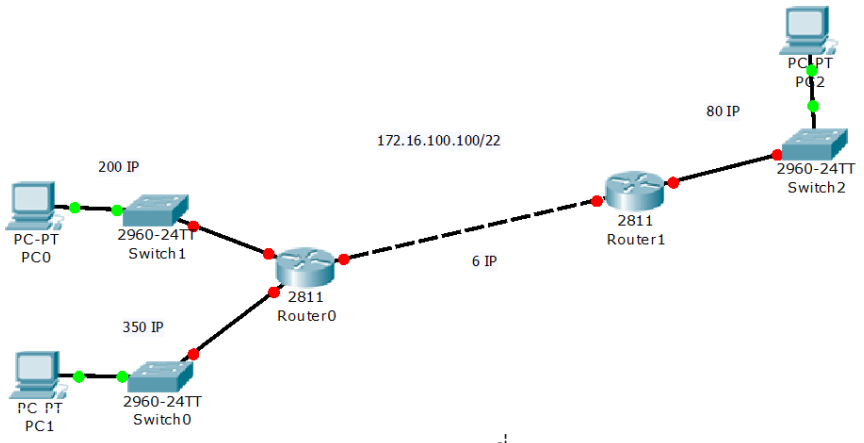
รูปที่ 2

3. ให้นิสิต config VTP ตามรูปแล้วสังเกตว่าเกิดอะไรขึ้นกับตัว client ดังรูปที่ 3



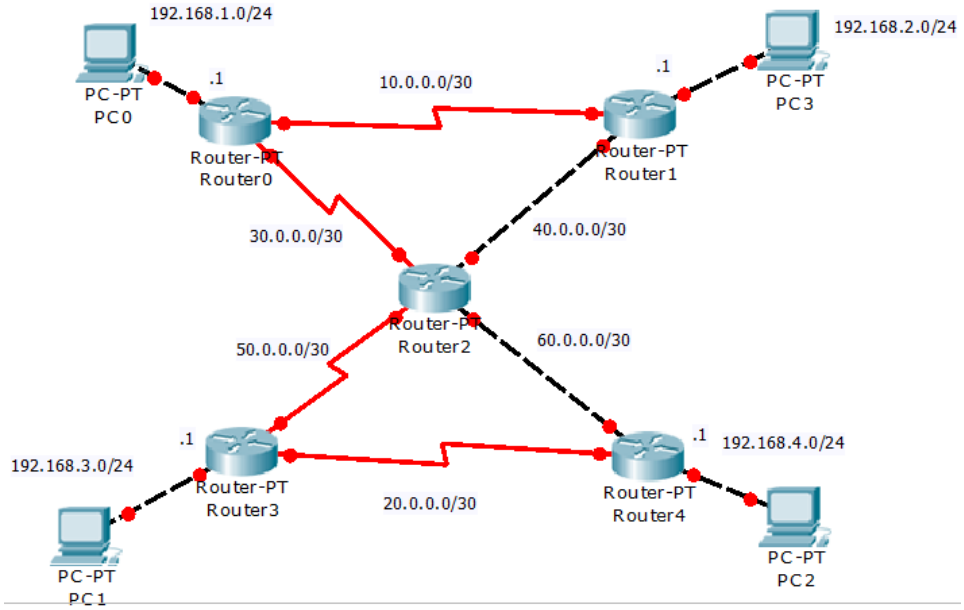
รูปที่ 3

4. จากรูปจงคำนวณ VLSM พร้อมกับ config เครือข่ายให้สามารถทำงานได้ โดยใช้ static route พร้อมสามารถ ping ได้ทุกๆ จุด ดังรูปที่ 4



รูปที่ 4

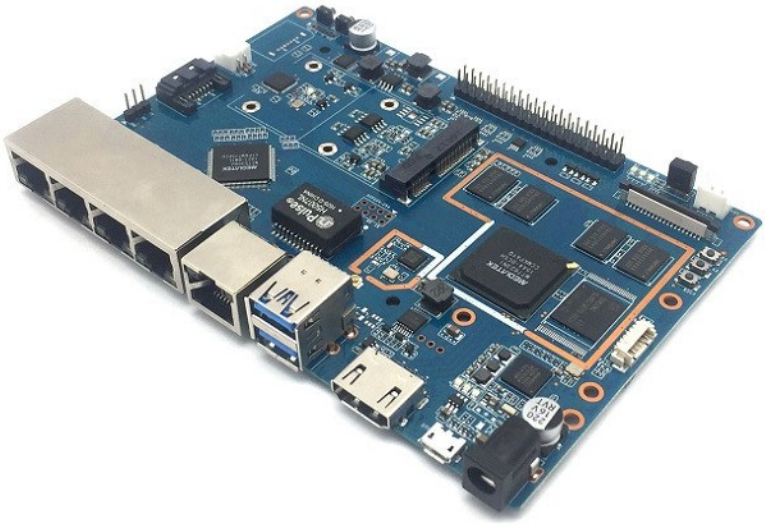
5. จากรูปที่ 5 จง config เครือข่ายให้สามารถทำงานได้ โดยใช้ static route พร้อมสามารถ ping ได้ทุกๆ จุด



รูปที่ 5

6. จากรูปที่ 5 ด้านบนจง config เครือข่ายให้สามารถทำงานได้ โดยใช้ dynamic route (RIP) พร้อมสามารถ ping ได้ทุกๆ จุด
7. จากรูปที่ 5 จง config เครือข่ายให้สามารถทำงานได้ โดยใช้ OSPF area 0 พร้อมสามารถ ping ได้ทุกๆ จุด
8. จากรูปที่ 5 ด้านบนจง config เครือข่ายให้สามารถทำงานได้ โดยใช้ EIGRP ที่ router 0, 1, 2 และใช้ IGRP ที่ router 2, 3, 4 พร้อมสามารถ ping ได้ทุกๆ จุด

ภาคที่สาม



โอเพนซอร์สเราเตอร์

บทที่ 13
โอเพนซอร์สเราเตอร์ (Open Source Router)

แนวคิด

ในบทนี้จะมาเรียนรู้ความหมายของคำว่าโอเพนซอร์ส ซอฟต์แวร์เราเตอร์ฟรีจากหลายๆ ค่าย คุณสมบัติ และความสามารถที่แตกต่างกัน

วัตถุประสงค์

1. เข้าใจความหมาย ข้อกำหนด และขอบเขตการใช้งานของซอฟต์แวร์โอเพนซอร์ส
2. เข้าใจคุณสมบัติ และความสามารถของซอฟต์แวร์โอเพนซอร์สเราเตอร์
3. สามารถเลือกซอฟต์แวร์โอเพนซอร์สฟรีที่เหมาะสมกับงาน

การสร้างสคริปโพรไฟล์และปฏิบัติการดูแลเครือข่ายโดยใช้ซอฟต์แวร์และฮาร์ดแวร์แบบระบบเปิดเผยโค้ดต้นแบบ

13.1 โอเพนซอร์ส & โอเพนซอร์สเราเตอร์

เป็นที่ทราบกันดีว่าอุปกรณ์หลักในระบบเครือข่ายคือ อุปกรณ์จัดเส้นทาง [14] เป็นอุปกรณ์คอมพิวเตอร์ที่ใช้ในระบบเครือข่ายคอมพิวเตอร์ ทำหน้าที่ในการจัดหาเส้นทางเพื่อส่งแพกเกตข้อมูลไปยังเครือข่ายปลายทางที่ต้องการ เราเตอร์ทำงานบนเลเยอร์ที่ 3 ตามมาตรฐานของ OSI Model เราเตอร์มีโครงสร้างทางด้านกายภาพคล้ายเครื่องคอมพิวเตอร์ตระกูล x86 ที่เราใช้งานกันอยู่ทั่วๆ ไป จะแตกต่างกันตรงที่เราเตอร์ถูกสร้างขึ้นมาให้ทำหน้าที่เฉพาะทาง คือ การจัดหาเส้นทาง ทำให้ในระบบเครือข่าย เปรียบเสมือนเจ้าหน้าที่ที่ควบคุมเส้นทางการจราจรทั้งหมด ถ้าเจ้าหน้าที่มีความฉลาด รอบรู้ก็จะสามารถเลือกเส้นทางหรือบริหารเส้นทางจราจรได้เป็นอย่างดี ส่งผลให้ระบบเครือข่ายทั้งระบบมีความเร็วตามไปด้วย จะเห็นได้ว่าเราเตอร์มีบทบาทสำคัญบนระบบเครือข่ายเป็นลำดับต้นๆ เพราะถ้าไม่มีเครือข่ายแล้วก็จะไม่มีข้อมูลที่สื่อสารกันได้ ในบทนี้จะอธิบายถึงส่วนประกอบหลักๆ ของเราเตอร์และโอเพนซอร์สเราเตอร์ เพื่อใช้เป็นพื้นฐานในการพัฒนาเราเตอร์ขึ้นมาใช้งานเองได้ต่อไปในอนาคต

13.1.1 โอเพนซอร์สซอฟต์แวร์ [15]

ซอฟต์แวร์โอเพนซอร์ส (open source software - OSS) คือ ซอฟต์แวร์ที่เปิดเผยหลักการหรือแหล่งที่มาของเทคโนโลยี่ของซอฟต์แวร์นั้น ให้บุคคลภายนอกได้ใช้ ภายใต้เงื่อนไขบางประการที่เปิดโอกาสให้ผู้ใช้ทำการแก้ไข ดัดแปลงและเผยแพร่โปรแกรมต้นฉบับ (source code) ได้ภายใต้เงื่อนไขทางข้อตกลงทางกฎหมาย เช่น GPL License หรือ BSDLicense จะแตกต่างกับฟรีแวร์ (freeware) ซึ่งหมายถึง ซอฟต์แวร์ที่สร้างขึ้นและสามารถนำไปใช้ได้ในทุกจุดประสงค์โดยไม่ต้องเสียค่าใช้จ่าย (เช่นราคาขายหรือค่าลิขสิทธิ์) ฟรีแวร์เป็นลักษณะก้ำกึ่งระหว่างซอฟต์แวร์พาณิชย์และซอฟต์แวร์โอเพนซอร์ส คืออนุญาตให้กลุ่มผู้พัฒนามีส่วนร่วมในการสร้างซอฟต์แวร์ แต่ก็ไม่เผยแพร่รหัสต้นฉบับสู่สาธารณชนเพื่อรักษาความลับทางการค้า

ส่วนแชร์แวร์ (Shareware) คือ เป็นซอฟแวร์ที่สามารถใช้งานได้ฟรี ให้มีการทดลองใช้ แต่มีข้อจำกัดในเรื่องอายุ การใช้งาน Commercial Software (ซอฟต์แวร์เชิงพาณิชย์) เป็นซอฟต์แวร์ที่ผลิตขึ้นมาเพื่อจำหน่ายเป็นโปรแกรมที่ถูกจดลิขสิทธิ์ถูกต้องตามกฎหมาย หากผู้ใดต้องการใช้ต้องจ่ายเงินเพื่อเป็นค่าบริการ ตัวอย่างโปรแกรมประเภท OSS และ Freeware ได้แก่

- Joomla, Mambo
- FileZilla
- FreeMind
- LearnSquare, Moodle
- GreenStone, Dreamweaver

- WinFTP
- MindManager
- e-Learning Solution
- Digital Library

โอเพนซอร์ส (open source) คือ การพัฒนาระบบใดระบบหนึ่งทางด้านคอมพิวเตอร์ด้วยเงื่อนไขที่ผู้สร้างสรรค์ หรือผู้คิดค้นไม่ถือเอาสิทธิแต่เพียงผู้เดียวในการพัฒนาระบบนั้น ๆ พร้อมทั้งเปิดเผยแหล่งต้นกำเนิดของระบบนั้นเช่น ซอร์สโค้ด หรือความเป็นมาทางด้านเทคนิคของการพัฒนาระบบดังกล่าว เพื่อเปิดโอกาสให้บุคคลอื่นนำเอาระบบนั้น ๆ ไปพัฒนาได้ต่อไป โดยมีเงื่อนไขทางกฎหมายบางประเภท เช่น สัญญาอนุญาตสาธารณะทั่วไปของกนู (จีพีแอล) หรือ สัญญาอนุญาตแจกจ่ายซอฟต์แวร์ของเบิร์กลีย์ (บีเอสดี) เป็นต้น

ซอฟต์แวร์โอเพนซอร์ส (open source software - OSS) คือ ซอฟต์แวร์ที่เปิดแผยหลักการหรือแหล่งที่มาของเทคโนโลยีของซอฟต์แวร์นั้นให้ บุคคลภายนอกได้ใช้ ภายใต้เงื่อนไขบางประการที่เปิดโอกาสให้ผู้ใช้ทำการแก้ไข ดัดแปลงและ เผยแพร่โปรแกรมต้นฉบับ ได้ภายใต้เงื่อนไขทางข้อตกลงทางกฎหมาย เช่น จีพีแอล หรือ บีเอสดี ซึ่งปัจจุบันมีการกำหนดโดยกลุ่มผู้กำหนดโอเพนซอร์สที่วางข้อกำหนดคำนิยาม 10 ประการในการกำหนดว่าเงื่อนไขที่เกี่ยวกับโอเพนซอร์ส คือ

1. เงื่อนไข จะต้องไม่จำกัดผู้หนึ่งผู้ใดในการจำหน่ายหรือการจ่ายแจกซอฟต์แวร์ให้เป็น ส่วนใดส่วนหนึ่งของซอฟต์แวร์แบบแยกส่วนที่ประกอบด้วยซอฟต์แวร์จากหลายแหล่ง และจะต้องไม่มีข้อกำหนดใด ๆ ที่เกี่ยวกันกับค่าใช้สิทธิหรือค่าสิทธิใด ๆ ในการจำหน่ายซอฟต์แวร์นั้น กล่าวคือให้มีการจ่ายแจกได้อย่างไม่มีการคิดค่าตอบแทน

2. โปรแกรม นั้นจะต้องเผยแพร่โปรแกรมต้นฉบับ และจำต้องยินยอมให้มีการแจกจ่ายโปรแกรมต้นฉบับได้เช่นเดียวกันกับโปรแกรมที่อยู่ในรูปของการแปลงเป็นโปรแกรมที่ใช้งานได้แล้ว โดยหากแม้ไม่สามารถนำสินค้านั้นแจกจ่ายได้พร้อมโปรแกรมต้นฉบับ ก็จำเป็นต้องมีแหล่งแห่งที่อันเป็นสาธารณะที่สามารถเข้าถึงโปรแกรมต้นฉบับ ซอร์สโค้ดได้โดยปราศจากค่าใช้จ่ายหรือต้นทุนอื่นใด ทั้งนี้โปรแกรมต้นฉบับนั้นจะต้องอยู่ในรูปแบบที่นักโปรแกรมสามารถที่จะแก้ไข ได้ โดยต้องปราศจากซึ่งการเขียนโปรแกรมต้นฉบับในลักษณะที่เป็นการสับสนโดย เจตนา รวมทั้งต้องไม่มีลักษณะของโครงสร้างการทำงานของโปรแกรมต้นฉบับที่จำต้องมี ตัวแปลภาษาเฉพาะ (translator) หรือมีส่วนที่ต้องนำเข้าสู่โปรแกรมในรูปแบบของโปรแกรมที่แปลงสภาพแล้ว (preprocessor)

3. เงื่อนไขจะต้องยินยอมให้สามารถทำการพัฒนาต่อยอดได้ ภายใต้เงื่อนไขการจ่ายแจกเช่นเดียวกันกับเงื่อนไขของโปรแกรมฉบับเริ่มต้น

4. เงื่อนไข อาจจะวางข้อกำหนดในการจำกัดเผยแพร่โปรแกรมต้นฉบับ ฉบับที่แก้ไขแล้วได้ต่อเมื่อเงื่อนไขนั้นได้ยินยอมให้มีการแจกจ่ายแพตซ์ไฟล์ (patch file) พร้อม โปรแกรมต้นฉบับเพื่อประโยชน์ในการแก้ไขโปรแกรมนั้นในเวลาทำการสร้างโปรแกรม ทั้งเงื่อนไขจำต้องยินยอมให้มีการ

แจกจ่ายโปรแกรมนั้นที่ได้รับการแก้ไข โปรแกรมต้นฉบับได้ แต่เงื่อนไขนั้นอาจจะกำหนดให้โปรแกรมฉบับต่อยอดใช้ชื่อที่แตกต่างหรือใช้ รุ่นที่แตกต่างจากโปรแกรมฉบับเริ่มต้นก็ได้

5. เงื่อนไขจะต้องไม่จำกัดเฉพาะบุคคลหรือกลุ่มบุคคลใด ๆ

6. เงื่อนไขต้องไม่จำกัดการใช้งานของโปรแกรมในรูปแบบใดรูปแบบหนึ่งอันเป็นการเฉพาะ

7. เงื่อนไขที่กำหนดจะต้องใช้กับทุกคนที่เกี่ยวข้องกับโปรแกรมนั้น

8. สิทธิใด ๆ ของโปรแกรมนั้นจะต้องไม่มีเงื่อนไขที่เฉพาะเจาะจงกับสินค้าหนึ่งสินค้าใด

9. เงื่อนไข ต้องไม่กำหนดอันเกี่ยวกับข้อจำกัดในการใช้ร่วมกันกับโปรแกรมอื่น เช่น กำหนดให้ต้องใช้โปรแกรมดังกล่าวกับโปรแกรมแบบโอเพนซอร์ซเท่านั้น

10. ต้องไม่มีข้อกำหนดใด ๆ ในเงื่อนไขที่กำหนดให้ใช้เทคโนโลยีของใครหรือเทคโนโลยีแบบใดเป็นการเฉพาะต่างกัน

รูปที่ 13.1 ริชาร์ด สตอลแมน ผู้ที่มีบทบาทสำคัญกับโอเพนซอร์ส

โอเพนซอร์ส OSS (Open Source Software) คืออะไร?

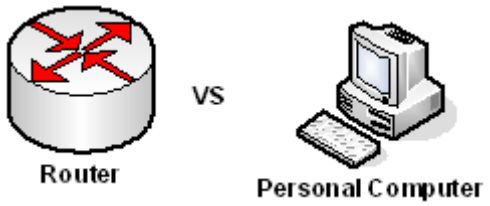
ซอฟต์แวร์ โอเพนซอร์สคือ ซอฟต์แวร์ที่ผู้ใช้มีอิสระอย่างเต็มที่ในการใช้งานโดยไม่ต้องกังวลเรื่อค่า ลิขสิทธิ์ สมามารถก็อปปี้เพื่อแจกจ่ายต่อไปได้นอกจากนี้ ผู้ใช้ยังสามารถปรับปรุง ซอฟต์แวร์เพื่อให้เมหาะสมแก่การใช้งานของตนเอง

ข้อดีของซอฟต์แวร์ โอเพนซอร์ส OSS (Open Source Software)

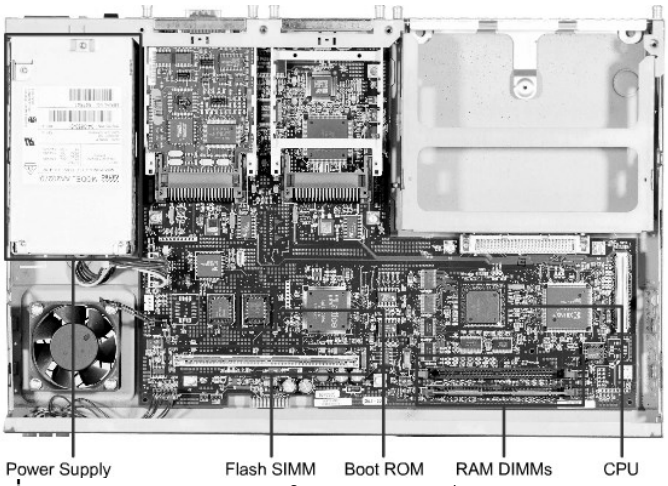
1. ลดค่าใช้จ่ายจากค่าลิขสิทธิ์ซอฟแวร์
2. ปรับเปลี่ยนได้ตามความต้องการของผู้ใช้
3. เพิ่มทางเลือกให้แก่ผู้ใช้
4. เปิดโอกาสในการพัฒนาทักษะของโปรแกรมเมอร์ไทย
5. ลด ความเสี่ยงที่จะใช้ ซอฟต์แวร์มีลิขสิทธิ์ รวมถึงประหยัดไม่ให้เงินตรารั่วไหลออกต่างประเทศ

13.1.2 โครงสร้างของเราเตอร์ [16]

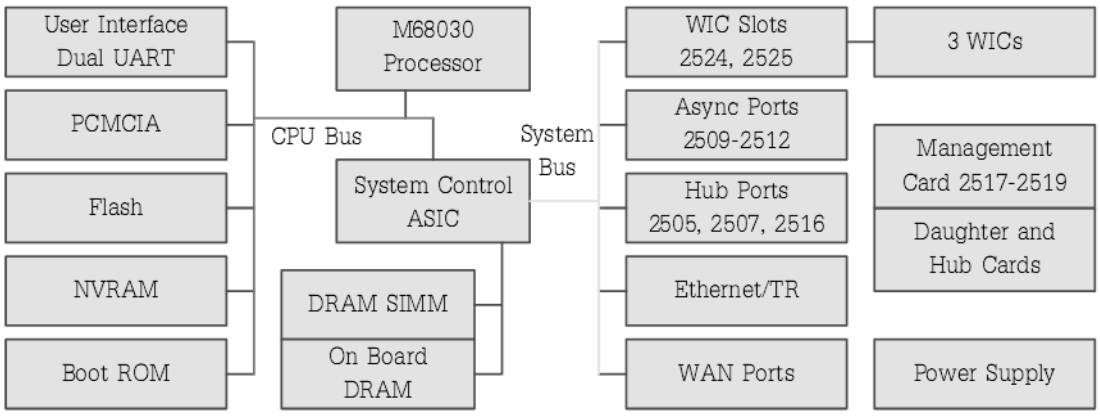
เราเตอร์มีสถาปัตยกรรมคล้ายคอมพิวเตอร์ส่วนบุคคลที่ใช้งานกันอยู่ในปัจจุบัน ซึ่งมีองค์ประกอบดังต่อไปนี้



รูปที่ 13.2 เราเตอร์ vs คอมพิวเตอร์ส่วนบุคคล



รูปที่ 13.3 ส่วนประกอบภายในของเราเตอร์ รุ่น Cisco 2600 [17]



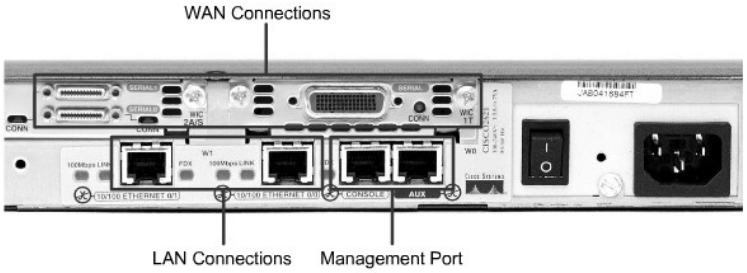
รูปที่ 13.4 ผังโครงสร้างของเราเตอร์

ส่วนประกอบหลัก ๆ ของเราเตอร์จะมีดังนี้

- **ROM** ทำหน้าที่เซ็ตค่าเริ่มต้นให้กับเราเตอร์ (POST) เหมือนกับการทำงานของ Bios ของเครื่องคอมพิวเตอร์ส่วนบุคคล รอมซอฟต์แวร์ขนาดเล็ก ๆ จัดการตรวจสอบความพร้อมของฮาร์ดแวร์ เมื่อฮาร์ดแวร์มีข้อผิดจะแจ้งเตือนเป็นลักษณะของเสียงว่าความผิดพลาดเกิดขึ้นจากอุปกรณ์ตัวใด
- **Flash Memory** เป็นหน่วยความจำแบบกึ่งถาวร ข้อมูลที่เก็บไว้จะสูญหาย แต่ถ้าต้องการลบข้อมูลจะต้องใช้กระแสไฟฟ้าช่วยในการลบข้อมูล (EPROM Erasable Programmable

Read-Only Memory) flash หน้าที่เก็บระบบปฏิบัติการสำหรับควบคุมการทำงานของเราเตอร์ถ้าในเครื่องคอมพิวเตอร์ทั่วไป ส่วนใหญ่จะเก็บอยู่ในฮาร์ดดิสก์

- **NVRAM** เก็บคอนฟิกกูเรชันไฟล์ เมื่อระบบปฏิบัติการเข้าควบคุมการทำงานของเราเตอร์เรียบร้อยแล้ว ขั้นตอนต่อไป เราเตอร์จะโหลดคอนฟิกไฟล์ที่เก็บอยู่ใน NVRAM ไปทำงาน เมื่อเกิดการเปลี่ยนแปลงค่าของคอนฟิกเกิดขึ้นในเราเตอร์ เราจะต้องบันทึกข้อมูลการเปลี่ยนแปลงทั้งหมดไปเก็บไว้ใน NVRAM ด้วย มิเช่นนั้นถ้าเครื่องมีการเริ่มต้นทำงานใหม่ คอนฟิกเดิมมีอยู่จะหายไปด้วย
- **RAM/DRAM** ทำหน้าที่เป็นหน่วยความจำหลักของเราเตอร์ เก็บคำสั่งที่ทำงานในปัจจุบัน (Running Configuration) เก็บข้อมูลตารางเส้นทาง ใช้ประมวลผลข้อมูล และซอฟต์แวร์อื่นๆ ที่ทำหน้าที่อยู่บนเราเตอร์
- **Router Interface** ใช้สำหรับเชื่อมต่อกับอุปกรณ์เครือข่ายอื่นๆ ภายนอก ชนิดอินเตอร์เฟสหลัก ๆ มี 2 ชนิดคือ อินเตอร์เฟสที่ใช้เชื่อมต่อกับ WAN Link และอินเตอร์เฟสที่ใช้เชื่อมต่อภายในเครือข่ายท้องถิ่น LAN Link ดังตัวอย่างรูปที่ **13.**4 อินเตอร์เฟสเครือข่ายท้องถิ่นในปัจจุบันนิยมใช้เทคโนโลยีแบบอีเทอร์เน็ต ดังนั้นจึงเป็นชนิด Ethernet/FastEthernet/Gigabit เป็นต้ร มีความเร็วที่ประมาณ 10/100/1,000 บิตต่อวินาทีตามลำดับ ส่วนอินเทอร์เฟสที่เชื่อมต่อโครงข่าย WAN มันนิยมใช้กิกกะบิตอีเทอร์เน็ต, ATM, Frame Relay เป็นต้น



รูปที่ 13.5 อินเทอร์เฟสชนิด WAN และ LAN

13.1.3 Open Source Router [18, 19, 20, 21]

โอเพนซอร์สเราเตอร์ คือซอร์ฟเวอร์ที่ทำหน้าที่จัดเส้นทางจราจร ค้นหาเส้นทาง ให้กับระบบเครือข่าย สามารถทำงานร่วมกับเครื่องคอมพิวเตอร์ที่นิยมใช้งานในปัจจุบัน (สถาปัตยกรรมแบบ x86) ได้เป็นอย่างดี การสร้างเราเตอร์บนสถาปัตยกรรม x86 มีข้อดีหลายประการคือ มีราคาถูก สามารถหาซื้อได้ง่าย ปรับแต่งคุณสมบัติของเครื่องได้เอง มีอุปกรณ์สำรองเพียงพอ และที่สำคัญคืออยู่ใกล้ตัวเราอย่างมาก (ใช้งานอยู่เกือบทุกวัน) จึงเป็นการสมเหตุสมผล ถ้าเราสามารถนำเครื่องคอมพิวเตอร์ที่ใช้งานอยู่เป็นประจำ มาทำหน้าที่เป็นอุปกรณ์เครือข่าย คือเราเตอร์ ซึ่งบางคนที่เรียนวิชาด้านคอมพิวเตอร์เน็ตเวิร์คอาจจะไม่เคยได้สัมผัสเราเตอร์จริงเลย ถ้าสามารถสร้างเราเตอร์บนเครื่องพีซีได้

อย่างสมบูรณ์แบบ คาดว่าน่าจะเป็นการปฏิวัติการศึกษาเรื่องการเรียนรู้วิชาคอมพิวเตอร์เน็ตเน็ตเวิร์คในอนาคตอย่างแน่นอน เพราะอุปกรณ์ที่สำคัญ และมีราคาแพงอย่างยิ่งบนเครือข่าย จะสามารถหามาใช้งานได้โดยง่าย และราคาไม่แพงอีกต่อไป สำหรับประเทศที่กำลังพัฒนาอย่างเรา การลงทุนด้านเทคโนโลยีด้านสารสนเทศก็ดูจะเป็นปัญหาหนักอยู่เหมือนกัน รัฐบาลผลักดันให้ทุกๆ โรงเรียนมีคอมพิวเตอร์และระบบเครือข่ายใช้แต่ก็ยังไม่สามารถทำได้เต็มที่นัก เพราะยังติดในเรื่องของงบประมาณนั่นเอง ในหนังสือเล่มนี้จะพยายามทำลายสิ่งกีดขวางในด้านเทคโนโลยีของเครือข่ายให้ได้มากที่สุด เพราะประโยชน์ที่จะได้รับในภายภาคหน้าจะมีอย่างมากมายคือ ผู้ที่เรียนด้านคอมพิวเตอร์เน็ตเวิร์คจะสามารถเข้าถึงอุปกรณ์เครือข่ายที่สำคัญอย่างเราเตอร์ได้จริง

คุณสมบัติของโอเพนซอร์สเราเตอร์ที่น่าสนใจมีหลายประการคือ

- สามารถหามาใช้งานได้ฟรี โดยไม่เสียค่าใช้จ่ายแต่อย่างใด
- สามารถปรับแต่งแก้ไขความสามารถของเราเตอร์ได้เอง
- Source code เปิดเผยทำให้สามารถสร้างองค์ความรู้ หรือวิจัยต่อยอดได้
- มีทีมพัฒนาช่วยกันทำงานอยู่ทั่วโลก
- ลดการนำเข้าซอฟต์แวร์และฮาร์ดแวร์ที่มีราคาแพงได้ (ช่วยชาติ)
- สามารถนำไปสร้าง Product ของตนเองได้

ผู้พัฒนาโอเพนซอร์สเราเตอร์

ปัจจุบันมีผู้พัฒนาโอเพนซอร์สที่เกี่ยวข้องกับ เราเตอร์หลายกลุ่ม ซึ่งแต่ละกลุ่มก็มีความโดดเด่นแตกต่างกันไป ในส่วนนี้จะแนะนำกลุ่มที่พัฒนาโอเพนซอร์สเราเตอร์ที่ได้รับความนิยม ดังต่อไปนี้

Zebra Project [22]

Zebra เป็นกลุ่มผู้พัฒนาซอฟต์แวร์ประเภทจัดเส้นทาง (Routing) ที่ค่อนข้างจะเก่าแก่ที่สุด เริ่มตั้งแต่ปี 1996 ผู้ริเริ่มก่อตั้งคือ Kunihiro Ishiguro ตอนที่เขาทำงานอยู่ที่ NIS ซึ่งเป็นงานด้านระบบเครือข่าย ในเวลานั้นซอฟต์แวร์ด้านจัดเส้นทางยังไม่ค่อยมีประสิทธิภาพมากนัก เขามีความคิดที่จะพัฒนา Routing Protocol ใหม่ด้วย ซึ่งพอดีได้เจอกับ Yoshinari Yoshikawa ซึ่งมีความคิดที่ตรงกัน Zebra Project จึงถือกำเนิดขึ้นในเวลานั้นเอง และอยู่ภายใต้ลิขสิทธ์ GNU General Public License เมื่อตั้งกลุ่มขึ้นมาแล้วก็มีผู้เชี่ยวชาญหลายสาขาเข้ามาร่วมกันพัฒนา ทำให้โครงการดังกล่าวเป็นที่ยอมรับและรู้จักกันเป็นอย่างดี

คุณลักษณะเด่นของ Zebra คือ

- Modularity ซอฟต์แวร์ที่พัฒนาขึ้นเป็นลักษณะแยกส่วน จึงทำให้ลดภาระการปรับปรุงแก้ไข และการดูแลรักษา

- Speed ทางทีมผู้พัฒนายืนยันว่าซอฟต์แวร์ Routing ของพวกเขามีความเร็วในการสื่อสารข้อมูลได้มากกว่าซอฟต์แวร์ Routing ที่ใช้งานอยู่
- Reliability เมื่อซอฟต์แวร์เกิดข้อผิดพลาดขึ้นไม่จำเป็นต้องหยุดระบบทั้งหมด เนื่องจากพัฒนาแบบแยกส่วน ขณะเกิดข้อผิดพลาดก็จะสามารถแก้ไขเฉพาะส่วนที่ทำงานผิดพลาดเท่านั้น จะไม่ส่งผลกระทบต่อโมดูลอื่นๆ ที่ทำงานอยู่

ความสามารถของ Zebra

- Zebra สามารถทำงานได้กับระบบปฏิบัติการ GNU/Linux 2.2.X and 2.4.X, FreeBSD 4.X, FreeBSD 5.X, NetBSD **13.1.**6.X, OpenBSD 3.X, Solaris
- สนับสนุนการทำงานกับ IPv6 แต่ต้องติดตั้งบนระบบปฏิบัติการ FreeBSD, NetBSD, OpenBSD, NU/Linux เท่านั้น
- สนับสนุนการทำงานของ Routing Protocol ดังต่อไปนี้

ตารางที่ 13.1 Routing Protocol ที่ Zebra สนับสนุน

Process Name	Protocol
bgpd	Manages BGP-4 and BGP-4+ protocol
ripd	Manages RIPv1, v2 protocol
ripngd	Manages RIPng protocol
ospfd	Manages OSPFv2 protocol
ospf6d	Manages OSPFv3 protocol
zebra	for Kernel routing table update and routing information redistribution between above protocols

ข้อมูลเพิ่มเติม : http://www.zebra.org/

Quagga Routing Software Suite [19]

Quagga เป็นชุดของซอฟต์แวร์ Routing เช่นเดียวกับ Zebra ซึ่งพัฒนาต่อยอดมาจาก Zebra อีกทีหนึ่ง โดยใช้ Core Deamon เดิมจาก Zebra และพัฒนาในส่วนของ library เพิ่มเติมเพื่อให้ง่ายต่อการพัฒนาและปรับแต่งระบบ

ข้อมูลเพิ่มเติม : http://www.quagga.net/about.php

Quagga Routing Suite

XORP (eXtensible Open Router Platform) [18]

XORP

XORP สนับสนุนการทำงาน Routing Protocol ทั้ง IPv4 และ IPv6 เต็มรูปแบบ และเพิ่ม Routing แบบ Multicast เข้าไปด้วย รูปแบบการคอนฟิกก็สามารถทำความเข้าใจได้ง่าย XORP ถูกออกแบบในลักษณะแยกส่วน ทำให้สามารถพัฒนาต่อยอด และขยายเพิ่มเติมในอนาคตได้โดยง่าย ลักษณะเด่นของ XORP มีดังนี้

- ออกแบบ CLI ให้สามารถใช้งานได้ง่ายๆ
- รองรับการทำงานของแอพพลิเคชั่นอื่นๆ ด้วย เช่น QoS, NAT, DHCP, Firewall
- อนุญาตให้ผู้ใช้งานสามารถเปิดใช้งาน Routing Protocol แต่ละตัวสามารถทำงานในสภาพแวดล้อมที่แตกต่างกันได้
- รองรับโพรโทคอลครบถ้วนสำหรับเราเตอร์คือ BGP4+, OSPFv2, OSPFv3, RIP and RIPng for unicast routing, and PIM-SM and IGMP/MLD for multicast
- XORP สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลายได้แก่ Linux, BSD, Mac OS X, Microsoft Windows
- ทำงานได้บนฮาร์ดแวร์ที่หลากหลาย โดยเฉพาะบน x86 เมื่อต้องการประสิทธิภาพการส่งข้อมูลที่สูงก็สามารถใช้งานร่วมกับ softwate stack เพื่อควบคุมอัตราการ forward ข้อมูล
- XORP เปิดเผย source code ทำให้สามารถพัฒนาโพรโทคอล routing ใหม่เข้าไปได้
- สามารถใช้งานร่วมกับเครื่องจักรเสมือน เพื่อทดสอบเราเตอร์ได้หลายตัว
- สามารถเขียนภาษา script เพื่อตอบโต้กับ XORP ได้
- สามารถตรวจสอบติดตามการทำงานของเราเตอร์ได้ เช่น ปริมาณข้อมูลที่วิ่งผ่าน routing message เป็นต้น
- มีเอกสารคู่มือการพัฒนา รวมถึง API ให้โดยละเอียด
- เขียนด้วยภาษา C, C++ จึงสามารถพัฒนาต่อยอดได้สะดวก
- มีคู่มือการใช้งานและคอนฟิกอย่างละเอียด
- ที่สำคัญคือ สามารถใส่ความคิดหรือนวตกรรมใหม่บน XORP ได้อย่างเป็นอิสระ

Vyatta [21]

Vyatta เป็นผู้พัฒนากลุ่มแรกที่มีแนวคิดที่จะสร้างเราเตอร์ราคาถูกที่ทำงานภายใต้สถาปัตยกรรม x86 เพื่อลดต้นทุนสำหรับองค์กรที่ต้องการเชื่อมต่อเครือข่ายโดยใช้ต้นทุนต่ำ

Vyatta พยายามเปรียบเทียบระหว่างอุปกรณ์เครือข่ายของบริษัทกับโอเพ่นซอร์สเราเตอร์ ในเรื่องประสิทธิภาพการทำงาน ราคา การขยายตัวของเครือข่าย การจัดหาอุปกรณ์สำรอง และความยืดหยุ่นในการใช้งาน Vyatta ยังสร้างตัวเราเตอร์ Product ออกมา 2-3 รุ่น หวังจะตีตลาดอุปกรณ์เครือข่ายอย่างเราเตอร์แบบต้นทุนต่ำ ซึ่งก็ดูเหมือนว่าแนวโน้มน่าจะไปได้ดี ทางทีมผู้พัฒนาได้ทดสอบการประสิทธิภาพการเชื่อมต่อ ฮาร์ดแวร์ที่สนับสนุน และความต้องการพื้นฐานที่จำเป็นสำหรับการติดตั้งและใช้งานเราเตอร์ โดยรายได้ส่วนหนึ่งมาจากผลิตภัณฑ์ที่สร้างขึ้น และอีกส่วนหนึ่งมาจากการให้คำแนะนำติดตั้ง และดูแลรักษา

Vyatta ได้พัฒนาซอฟต์แวร์หลายตัวให้สามารถทำงานร่วมกับเราเตอร์ได้ เพื่อให้ครอบคลุมด้านเครือข่ายทั้งหมด เช่น Firewall, VPN และ Antivirus เป็นต้น สำหรับอินเทอร์เฟสที่ใช้เชื่อมต่อก็ได้ทำการทดสอบกับเทคโนโลยีหลายประเภท เช่น Ethernet, T1/E1, T2/E3, Gigabit เป็นต้น และอีกหลายเรื่องที่มีความน่าสนใจมาก เช่น Data Center, Colocation, Security, Virtualization เป็นต้น (สามารถอ่านได้จากบทที่ 2) หรือสามารถค้นหาข้อมูลเพิ่มเติมได้ที่ http://www.vyatta.com/index.php Vyatta สนับสนุนแอพพลิเคชันและโพรโทคอลต่างๆ ดังตารางที่ 13.2

ตารางที่ 13.2 Vyatta software support

Software	
IP and Routing Protocols	» IPv4 » RIPv2 » OSPFv2 » Static routes » BGPv4 » IPv6*
IP Address Management	» Static » DHCP Relay » DHCP Server » Dynamic DNS » DHCP Client » DNS Forwarding
Encapsulations	» Ethernet » Frame Relay » 802.1Q VLANs » MLPPP » PPP » HDLC

	» PPPoE » GRE » IP in IP
Performance Optimization	» WAN Link Load Balancing » Priority Queuing (QoS) » Ethernet Link Bonding » Classful Queuing (QoS) » MLPPP » Bandwidth Management » ECMP » Web Caching
Logging and Monitoring	» Syslog » SNMPv2c
Security	» Stateful Inspection Firewall » Network Address Translation » SSL-based OpenVPN » Site to Site VPN (IPSec) » Remote VPN (PPTP, L2TP, IPSec) » DES, 3DES, AES Encryption » MD5 and SHA-1 Authentication » Intrusion Prevention » URL Filtering
High Availability	» VRRP » IPSec VPN Clustering » Protocol Fault Isolation
Administration	» Integrated CLI » Web GUI » Single Configuration File » Telnet » SSHv2
Diagnostics & Packet Sniffing	» tcpdump » Wireshark Packet Capture » BGP MD5 support

	» Serial Loopback Commands
Virtualization Readiness	» Integrated open-vm-tools » Xen Para-virtualization

13.2 คุณสมบัติของฮาร์ดแวร์สำหรับออกแบบพัฒนาโอเพนซอร์สเราเตอร์

ในบทนี้จะกล่าวถึงคุณสมบัติของฮาร์ดแวร์ที่สนับสนุนการเชื่อมต่อระบบเครือข่ายด้วย Open-Source การเชื่อมต่อระบบเครือข่ายโดยใช้เทคโนโลยีโอเพนซอร์สสามารถใช้อุปกรณ์เครือข่ายทั่วๆ ไปได้อยู่แล้ว แต่มีความพิเศษคือ อุปกรณ์เครือข่ายประเภทเราเตอร์สามารถใช้คอมพิวเตอร์ส่วนบุคคล หรือคอมพิวเตอร์เวิร์ดสเตชัน แทนอุปกรณ์เราเตอร์จริงๆ ได้ อีกทั้งมีความได้เปรียบทางด้านความรวดเร็ว หน่วยความจำ และอื่นๆ ที่คอมพิวเตอร์ส่วนบุคคลมีความรุดหน้าไปไกลมาก แล้ว ต่างจากเราเตอร์ที่สร้างขึ้นมาทำหน้าที่โดยเฉพาะ เนื่องจากคุณสมบัติของฮาร์ดแวร์ที่มีอยู่อย่างจำกัด เช่น ความเร็วในการประมวลผล หน่วยความจำ และหน่วยเก็บข้อมูลเป็นต้น และถ้าจำเป็นต้องมีการขยายเครือข่ายจะต้องใช้งบในการลงทุนที่สูงมาก ซึ่งต่างจากการสร้างเครือข่ายด้วยโอเพนซอร์สมากเพราะอุปกรณ์สามารถจัดหาได้อย่างง่ายดาย สามารถปรับปรุงระบบได้ทันที และราคาถูก (ดูเพิ่มเติมเรื่องการเปรียบเทียบระหว่าง open-source และ Propietary) สำหรับข้อจำกัดของฮาร์ดแวร์ที่ใช้กับโอเพนซอร์สจะมีอยู่เพียงอย่างเดียวคือ การ์ดอินเทอร์เฟส ที่ยังไม่ค่อยมีการสนับสนุนหรือผลิตออกมาจำหน่วยตามท้องตลาดมากนัก ดังนั้นในบทนี้จึงต้องมาทำความเข้าใจก่อนว่าอุปกรณ์ใดบ้างที่สามารถรองรับการทำงาน เมื่อต้องการสร้างเครือข่ายด้วยเทคโนโลยีโอเพนซอร์ส

13.2.1 อุปกรณ์ฮาร์ดแวร์ที่สนับสนุนการสร้างเครือข่ายด้วยโอเพนซอร์สเทคโนโลยี

โอเพนซอร์สส่วนใหญ่สามารถทำงานได้บนสถาปัตยกรรม x86 (เครื่องคอมพิวเตอร์ทั่วๆ ไปในปัจจุบัน [23]) สามารถประมวลผลกับหน่วยประมวลผลกลางแบบตัวเดียว (Single Processor) หรือหลายๆ ตัว (Multi-Processor) ได้ ในปัจจุบันราคาของคอมพิวเตอร์ที่มีหน่วยประมวลผลหลายตัวก็มีราคาถูกลงมาก ทำให้การพัฒนาเราเตอร์ด้วยโอเพนซอร์สมีประสิทธิภาพสูงขึ้นอย่างมาก สามารถออกแบบระบบเครือข่ายได้ตั้งแต่ขนาดเล็ก (ใช้ฮาร์ดแวร์ที่มีหน่วยประมวลผลแบบตัวเดียว ขนาดของคอมพิวเตอร์จะมีขนาดเล็ก เช่น พีซีทั่วไป) ขนาดกลาง (ใช้ฮาร์ดแวร์ที่มีหน่วยประมวลผลกลางแบบตัวเดียวหรือหลายตัว คอมพิวเตอร์จะมีความสามารถเพิ่มขึ้น เสถียรกว่าแบบแรก เรียกว่าเวิร์ดสเตชัน) ขนาดใหญ่มาก จนถึงระดับศูนย์กลางให้บริการข้อมูล (Data Center) ก็สามารถทำได้ (ใช้ฮาร์ดแวร์ที่มีหน่วยประมวลผลกลางหลายตัว คอมพิวเตอร์จะมีประสิทธิภาพในการทำงานสูงมากๆ ประมวลผลได้เร็ว ขนาดหน่วยความจำสูง เสถียร สามารถทำงานได้ตลอดเวลา มีระบบสำรองในกรณีที่เกิดเหตุชุกเฉินสามารถทำงานต่อได้) เราเตอร์โอเพนซอร์สสามรถใช้เครื่องที่ประกอบขึ้นเอง (3rd-party) ก็สามารถทำงานได้เช่นกัน ดังรูปที่ 13.6

(a) คอมพิวเตอร์ส่วนบุคคล

(b) คอมพิวเตอร์เวิร์คสเตชัน

(c) คอมพิวเตอร์คุณภาพสูง High-End

รูปที่ 13.6 แสดงเครื่องคอมพิวเตอร์ตะกูล x86

หน่วยประมวลผลกลาง (Processor)

โอเพนซอร์สเราเตอร์สนับสนุนการทำงานแบบ muti-processing เมื่อนำมาทำงานบนเครื่องที่มีสถาปัฎตยกรรมที่มีหน่วยประมวลผลแบบหลายตัว (mulit-processor) จะทำให้ประสิทธิภาพการทำงานโดยรวมสูงมาก โดยแต่ละอินเทอร์เฟสที่มีการสื่อสารข้อมูลจะทำงานแยกเป็นอิสระออกจากกันโดยควบคุมจากหน่วยประมวลผลที่แยกกันทำงานนั่นเอง สำหรับในการพิจารณาว่าจะใช้หน่วยประมวลผลชนิดไหน หรือมีขนาดของหน่วยประมวลผลเท่าไหร่ จึงจะเหมาะกับงานนั้นให้พิจารณาคุณสมบัติดังต่อไปนี้

- จำนวนช่องทางที่ใช้สื่อสาร และปริมาณการสื่อสาร (ในทางปฏิบัติจะพิจารณาว่าใช้ปริมาณการ์ดอินเทอร์เฟสเท่าใด)
- จำนวนของซอฟต์แวร์ที่ทำหน้าที่เลือกเส้นทาง (Routing Protocol) ในเราเตอร์ เช่น RIP, OSPF, BGP เป็นต้น ถ้ามีโพรโทคอลทำงานบนเราเตอร์มากกว่า 1 โพรโทคอล จำเป็นจะต้องเพิ่มจำนวนของหน่วยประมวลผล หน่วยความจำ และหน่วยเก็บข้อมูลให้สูงขึ้นด้วย
- จำนวนของซอฟต์แวร์ที่จำเป็นต้องทำงานอยู่ตลอดเวลา (Active Program) เช่น กำหนดให้เราเตอร์ทำหน้าที่เป็น VPN, NAT, Firewall หรือ SNMP เป็นต้น ซึ่งซอฟต์แวร์เหล่านี้จำเป็นต้องทำงานอยู่ตลอดเวลาจนกว่าจะปิดเครื่อง ซึ่งธรรมชาติของซอฟต์แวร์ดังกล่าวจะต้องคอยรองรับการร้องขอจากผู้ใช้บริการตลอดเวลา หรือซอฟต์แวร์ที่ซอฟต์แวร์ที่จำเป็นพื้นฐาน เช่น โปรแกรมป้องกันไวรัส โปรแกรมตรวจจับผู้บุกรุก เป็นต้น
- ซอฟต์แวร์อื่นๆ ที่อาจจะจำเป็นต้องมีการติดตั้งเพิ่มเติมในภายหลัง

ประสิทธิภาพในการประมวลผลของเราเตอร์จะเพิ่มสูงขึ้นได้อีก โดยการพิจารณาคุณสมบัติพิเศษบางประการของคอมพิวเตอร์คือ ความเร็วของสัญญาณนาฬิกา (Clock Speed), ความเร็วของ font side bus, ขนาดของแคช เป็นต้น ดังรูปที่ 13.7

รูปที่ 13.7 หน่วยประมวลผลกลางตระกูล x86 [24, 25, 26]

หน่วยความจำหลัก (Main Memory)

หน่วยความจำหลัก มีหน้าที่ ในการเก็บ ระบบปฏิบัติการ ข้อมูล และโปรแกรม ที่จะให้ ซีพียู เรียกไปใช้งานได้อย่างรวดเร็ว สำหรับในเราเตอร์หน่วยความจำหลักมีความจำเป็นและมีความสำคัญมาก ข้อมูลที่เก็บลงในหน่วยความจำมีดังนี้คือ

- เส้นทางการเดินทางของข้อมูลทั้งหมด (Routes)
- ซอฟต์แวร์ค้นหาเส้นทางทั้งหมดที่กำลังทำงาน (Routing Protocol)
- ข้อมูลของอุปกรณ์เราเตอร์อื่นๆ ที่กำลังเชื่อมต่อกันอยู่ (peers)
- จำนวนช่องทางสื่อสารแบบท่อ (Tunnel) เมื่อมีการเชื่อมต่อเครือข่ายแบบ VPN เกิดขึ้น
- ข้อมูลของการท่องเว็บ (Web Caching)
- เก็บข้อมูลอื่นๆ ที่เกิดจากการติดตั้งหรือใช้งานเพิ่มเติมในภายหลัง

ขนาดของหน่วยความจำจะมีผลอย่างมากต่อความเร็วในการประมวลผลของเราเตอร์ เนื่องจากการมีหน่วยความจำในการใช้งานมากย่อมสามารถเก็บข้อมูลต่างๆ ที่จำเป็นต้องใช้งานได้มาก เพราะความเร็วในการเข้าถึงหน่วยความจำหลักจะเร็วกว่าการเข้าถึงข้อมูลในฮาร์ดดิสก์หลายร้อยเท่าทีเดียว

รูปที่ 13.8 แสดงหน่วยความจำหลัก

หน่วยเก็บข้อมูลสำรอง (Storage)

Storage คือสื่อบันทึกหรือจัดเก็บข้อมูล หรือที่เรียกกันอีกชื่อหนึ่งว่า หน่วยความจำสำรอง(secondary storage) เช่น แผ่นดิสก์, เทป, ฮาร์ดดิสก์, แผ่นซีดี, แผ่นดีวีดี, SD-card, CF-card, handy drive เป็นต้น ความเร็วในการเข้าถึงจะต่ำกว่าหน่วยความจำหลักเป็นร้อยเท่า ข้อมูลที่

อยู่ในหน่วยความจำประเภทนี้จะไม่สามารถประมวลผลข้อมูลได้โดยตรง เมื่อต้องการประมวลผลจะถูกโหลดไว้ในหน่วยความจำหลักก่อนจึงสามารถประมวลผลข้อมูลได้ ในเราเตอร์จะใช้หน่วยความจำประเภทนี้ในการจัดเก็บประวัติกิจกรรมที่ทำงาน (Loging Activity), ข้อมูลการใช้งานเว็บไซต์ (Web Caching) และซอฟต์แวร์อื่นๆ ที่ติดตั้งและใช้งานเพิ่มเติมในภายหลัง รวมถึงการอัพเดทซอฟต์แวร์ก็จำเป็นต้องใช้หน่วยความจำนี้ทำงานด้วย การใช้ซอฟต์แวร์เราเตอร์ให้ทำงานบนเครื่องคอมพิวเตอร์นั้นจำเป็นต้องใช้เนื้อที่ของหน่วยความจำสำรองอย่างน้อยประมาณ 2 GB ขนาดของหน่วยความจำสำรองอาจจะจำเป็นต้องเพิ่มขึ้นได้ในอนาคต สิ่งที่ควรพิจารณาว่าจะใช้หน่วยความจำที่มีขนาดเท่าใดให้พิจารณาปัจจัยดังต่อไปนี้

- มีการจัดเก็บประวัติกิจกรรมการทำงานที่ละเอียดมากๆ จำเป็นต้องใช้เพิ่มที่เพิ่มขึ้นตามไปด้วย
- มีการเปิดใช้งานหรือติดตั้งซอฟต์แวร์ที่ทำหน้าที่เป็นแคชบนเราเตอร์ (เมื่อต้องการความเร็วที่สูงขึ้นควรเก็บข้อมูลของแคชไว้ในหน่วยความจำประเภท flash แทน
- เมื่อเปิดใช้งานซอฟต์แวร์ประเภทป้องกันการโจมตีจากแฮกเกอร์บนเราเตอร์ (Intrusion Prevention) จำเป็นต้องใช้หน่วยความจำดังกล่าวเก็บกฎและข้อบังคับเพิ่ม
- เก็บข้อมูลที่เกิดจากการทำงานของซอฟต์แวร์อื่นๆ ที่ติดตั้งเพิ่มเติมภายหลัง

รูปที่ 13.9 แสดงหน่วยความจำสำรอง (Storage)

จากที่กล่าวมาแล้วข้างต้น สรุปว่าโอเพนซอร์สเราเตอร์จะมีประสิทธิภาพที่ดีต้องขึ้นอยู่กับปัจจัยหลายๆ อย่างทำงานร่วมกัน ตารางที่ 13.3 แสดงข้อแนะนำพื้นฐานที่ควรนำมาพิจารณาสำหรับการออกแบบและสร้างเราเตอร์ด้วยเทคโนโลยีของโอเพนซอร์ส

ตารางที่ 13.3 ปัจจัยพื้นฐานที่ควรนำมาพิจารณาสำหรับการออกแบบและสร้างเราเตอร์

คุณสมบัติของเราเตอร์ที่ต้องการออกแบบ	**ความต้องการของระบบที่จำเป็นต้องใช้**		
	หน่วยความจำ	**หน่วยความจำสำรอง (HD)**	**หน่วยประมวลผล**
• การ์ดอินเทอร์เฟสแบบ Ethernet จำนวน 1-4 ใบ ความเร็ว 100 Mbps			

• การ์ดอินเทอร์เฟสแบบ T1/E1 จำนวน 1-2 ใบ • เปิดใช้งานโพรโทคอลค้นหาเส้นทาง 1 โพรโทคอล • สามารถรองรับจำนวนเส้นทางได้ถึง 100 เส้นทาง	512 MB	2 GB	1 Core
• การ์ดอินเทอร์เฟสแบบ Ethernet ความเร็ว 1 GB, T1, E1, T3, E3 จำนวนตั้งแต่ **13.2-**6 ใบ • เปิดใช้งานโพรโทคอลค้นหาเส้นทาง 2 โพรโทคอล • สามารถรองรับจำนวนเส้นทางได้ถึง 10,000 เส้นทาง • รองรับการทำไฟล์วอลล์และ NAT	1 GB	2 GB ขึ้นไป	1-2 Core
• การ์ดอินเทอร์เฟสแบบ Ethernet ความเร็ว 1 GB, T1, E1, T3, E3 จำนวนตั้งแต่ 6-10 ใบ • เปิดใช้งานโพรโทคอลค้นหาเส้นทางเกิน 2 โพรโทคอล • ไม่จำกัดจำนวนเส้นทาง • รองรับการทำไฟล์วอลล์, NAT, VPN	2 GB	2 GB ขึ้นไป	**13.2-**4 Core
• การ์ดอินเทอร์เฟสแบบ Ethernet ความเร็ว 1 GB, T1, E1, T3, E3 เกิน 10 ใบขึ้นไป • เปิดใช้งานโพรโทคอลค้นหาเส้นทางเกิน 2 โพรโทคอล • ไม่จำกัดจำนวนเส้นทาง และสามารถรองรับ peer ของโพรโทคอล BGP ได้ด้วย • รองรับการทำไฟล์วอลล์ที่ซับซ้อน, NAT, VPN, Intrusion Prevention, Antivirus, web caching	4 GB	4 GB ขึ้นไป	4-8 Core
• การ์ดอินเทอร์เฟสแบบ Ethernet ความเร็ว 1 GB, T1, E1, T3, E3 ไม่จำกัดเต็มจำนวน slot ที่สามารถใส่ได้ • เปิดใช้งานโพรโทคอลค้นหาเส้นทางไม่จำกัด	8 GB ขึ้นไป	20 GB ขึ้นไป และขึ้นอยู่กับความถี่ในการบันทึก log,	**13.2-**4 CPU แต่

• ไม่จำกัดจำนวนเส้นทาง และสามารถรองรับ peer ของโพรโทคอล BGP ได้ด้วย • รองรับการทำไฟล์วอลล์ที่ซับซ้อน, NAT, VPN, Intrusion Prevention, Antivirus, web caching, SNMP, Spam filtering และซอฟต์แวร์อื่นๆ ที่จำเป็น		เก็บข้อมูลเว็บ, กฎ และข้อมูลอื่นๆ	ละ CPU ควรมีอย่างน้อยตัวละ 4 Core

จากข้อมูลในตารางที่ 13.3 เป็นข้อมูลที่จำเป็นอย่างยิ่งสำหรับใช้พิจารณาในการออกแบบและสร้างเราเตอร์เท่านั้น ไม่ได้ครอบคลุมข้อมูลอื่นๆ ที่มีผลกระทบต่อการออกแบบและสร้างเราเตอร์เหมือนกัน แต่ไม่สำคัญมากนัก เช่น ความเร็วของ font side bus, ขนาดของแคช, ชนิดของอุปกรณ์จัดเก็บข้อมูลสำรอง เช่น IDE, SATA, SCSII, Solid State เป็นต้น ข้อมูลที่แสดงดังกล่าวนี้เป็นข้อมูลที่ผ่านการทดสอบแล้วจาก vyatta [27] และแนะนำให้ลูกค้าใช้หรือผู้ที่สนใจจะสร้างเราเตอร์ขึ้นมาใช้งานเอง

การวัดประสิทธิภาพของฮาร์ดแวร์

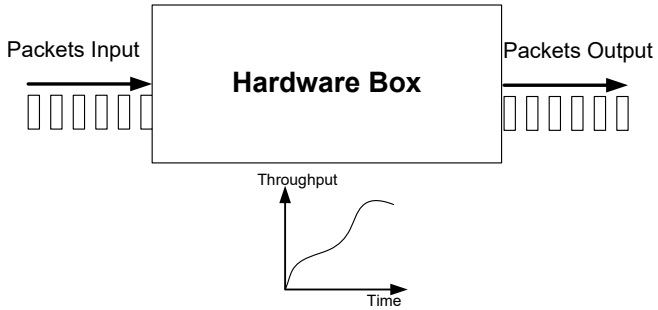
เพื่อให้เกิดความมั่นใจในการออกแบบ สร้าง และใช้งานเราเตอร์จากคอมพิวเตอร์ที่มีโครงสร้างสถาปัตยกรรมแบบ x86 จะต้องมีกระบวนการวัดประสิทธิภาพของฮาร์ดแวร์ ซึ่งในการทดสอบวัดประสิทธิภาพดังกล่าวนี้จะทดสอบกับคอมพิวเตอร์ที่ผู้ใช้สามารถประกอบเองได้หรือสามารถหาซื้อได้โดยง่ายที่มีขายโดยทั่วไปตามท้องตลาดหรือที่เรียกว่า Third Party (3rd Party) สาเหตุหลักที่เลือกทดสอบกับฮาร์ดแวร์ดังกล่าวเนื่องจากเราเตอร์ที่สร้างขึ้นจะมีความสามารถทำงานได้บนเครื่องที่หาได้ง่าย เพื่อลดความยุ่งยากและสิ้นเปลืองงบประมาณในการจัดซื้อฮาร์ดแวร์ที่มีคุณภาพสูง แต่ก็ไม่ได้หมายความว่าฮาร์ดแวร์ที่มีประสิทธิภาพสูงอย่าง เช่น HP, Compaq, IBM, Digital จะไม่สามารถทำงานได้ ในความเป็นจริงถ้าสามารถหาฮาร์ดแวร์ที่มีคุณภาพดีและได้มาตรฐานก็ยิ่งทำให้เราเตอร์ที่สร้างขึ้นมีประสิทธิภาพสูงขึ้นด้วย ดังนั้นการสร้างเราเตอร์จึงควรพิจารณาด้วยว่าจะใช้งานด้านไหน เช่น ถ้าต้องการสร้างเราเตอร์เพื่อใช้ทดสอบเครือข่ายหรือใช้ทดสอบการเรียนการสอนเรื่องเราเตอร์ก็ไม่มีความจำเป็นต้องใช้ฮาร์ดแวร์ที่มีราคาแพงและมีประสิทธิภาพสูง แต่ถ้าต้องการใช้เราเตอร์ในการสร้างเครือข่ายให้สามารถใช้งานได้จริงๆ ก็ควรจะหาฮาร์ดแวร์ที่มีประสิทธิภาพที่ดีมาใช้งานจึงเหมาะสมกว่า

เป้าหมายในการทดสอบ

มาตรฐานตัวชี้วัดสำหรับการทดสอบประสิทธิภาพของฮาร์ดแวร์สามารถดูข้อมูลเพิ่มเติมได้จาก RFC 2544 [28] ในส่วนนี้จะแสดงผลจากการทดสอบหาประสิทธิภาพฮาร์ดแวร์ที่ส่งผลกระทบ

หลักๆ กับการทำงานของเราเตอร์เท่านั้น เช่น Throughput, ขนาดของข้อมูลแพ็กเก็ต (Packet) ที่รับส่งมีขนาดที่แตกต่างกัน, ขนาดของบัส และเทคโนโลยีที่ใช้เป็นต้น

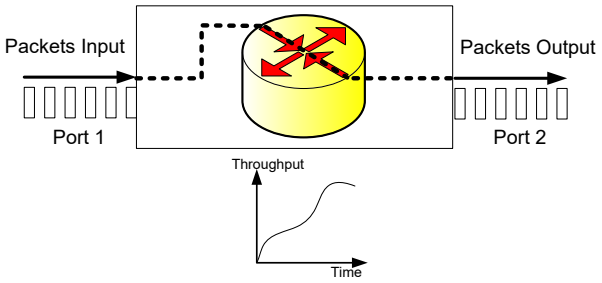
- Throughput คือปริมาณของข้อมูลที่สามารถส่งได้ต่อหนึ่งหน่วยเวลา เช่น สามารถส่งข้อมูลได้ที่ความเร็ว 100 Mbps (เม็กกะบิตต่อวินาที) ในการวัดประสิทธิภาพของ Throughput สามารถทำได้ง่ายๆ โดยการสร้างข้อมูลจำนวนมากๆ แล้วส่งเข้าไปยังอุปกรณ์ที่ต้องการวัดประสิทธิภาพ และวัดปริมาณข้อมูลที่ไหลออกทาง output ในช่วงเวลาที่กำหนดเช่น อาจจะใช้หน่วยวัดเป็นวินาที นาที หรือชั่วโมงก็ได้



รูปที่ 13.10 แสดงการวัดประสิทธิภาพฮาร์ดแวร์ Throughput

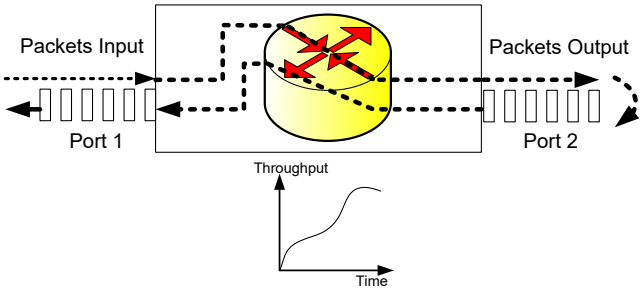
การวัดประสิทธิภาพด้วยวิธีดังกล่าวนี้จะไม่มีการเชื่อมต่ออุปกรณ์เข้ากับสภาพแวดล้อมที่ใช้งานจริงๆ พูดง่ายๆ คือนำอุปกรณ์มาทดสอบเพียงตัวเดียวเท่านั้นแล้ววัดประสิทธิภาพ เรียกวิธีการแบบนี้ว่า zero-loss throughput

- Undirectional คือการวัดปริมาณข้อมูลที่ไหลผ่านอุปกรณ์ในทิศทางเดียว เทคนิคนี้จะทดสอบโดยการป้อนข้อมูลอย่างต่อเนื่องเข้าพอร์ตหนึ่ง และให้อุปกรณ์ที่ทดสอบทำการส่งข้อมูลต่อ (Route) ไปยังอีกพอร์ตหนึ่งที่ไม่ใช่พอร์ตเดียวกัน แล้ววัดปริมาณข้อมูลที่แตกต่างกันขณะป้อนเข้าไปในระบบกับข้อมูลที่ไหลออกมาจากระบบในทิศทางเดียวเท่านั้น



รูปที่ 13.11 แสดงการวัดประสิทธิภาพฮาร์ดแวร์ Unidirection

- Bidirectional คือการการวัดปริมาณข้อมูลที่ไหลผ่านอุปกรณ์ทั้ง 2 ทิศทาง เทคนิคนี้จะทดสอบโดยการป้อนข้อมูลอย่างต่อเนื่องเข้าพอร์ตหนึ่ง (สมมุติว่า port 1) และให้อุปกรณ์ที่ทดสอบทำการส่งข้อมูลต่อ (Route) ไปยังอีกพอร์ตหนึ่งที่ไม่ใช่พอร์ตเดียวกัน (สมมุติว่า port 2) จากนั้นให้ทำการส่งข้อมูลเข้ากลับทิศทางกันคือ ข้อมูลป้อนเข้าที่พอร์ตที่ 2 และข้อมูลไหลออกที่พอร์ตที่ 1 จากนั้นทำการวัดปริมาณข้อมูลเมื่อข้อมูลเดินทางครบทั้ง 2 ทิศทาง ตามหน่วยเวลาที่กำหนด เช่น ใน 1 วินาทีเป็นต้น



รูปที่ 13.12 แสดงการวัดประสิทธิภาพฮาร์ดแวร์ Bidirection

ผลการทดสอบประสิทธิภาพอินเทอร์เฟสชนิด Fast Ethernet

สำหรับการทดสอบในอันดับแรก จะทำการทดสอบการสื่อสารข้อมูลด้วยเทคโนโลยีแบบอีเทอร์เน็ตที่ความเร็ว 100 Mbps ในตารางที่ 13.4 แสดงผลการทดสอบประสิทธิภาพเราเตอร์แบบ A (ใช้หน่วยประมวลผลยี่ห้อ VIA C7 ที่ความถี่สัญญาณนาฬิกาเท่ากับ 1.5 GHz) และทดสอบแบบทิศทางเดียว (Bidirectional)

ตารางที่ 13.4 ผลการทดสอบประสิทธิภาพฟาสต์อีเทอร์เน็ต 100 Mbps แบบ Unidirection [29]

ขนาดของเฟรมข้อมูล (ไบต์)	ความเร็วที่ทดสอบ 100 Mbps (ทิศทางเดียว : pps)	เราเตอร์ A ใช้ชิบแบบ VIA ความถี่ 1.5 GHz (pps)	ปริมาณข้อมูลที่วัดได้
64	148,809	66,848	45%
128	84,459	59,055	70%
256	45,289	45,289	100%
512	23,496	23,496	100%
768	15,862	15,862	100%
1,024	11,973	11,973	100%
1,280	9,615	9,615	100%
1,518	8,127	8,127	100%

จากตารางที่ 13.4 คอลัมน์ที่ 1 เป็นอินพุตที่ใช้ทดสอบ ขนาดของเฟรมข้อมูลมีหน่วยเป็นไบต์ การทดสอบจะเปลี่ยนขนาดของเฟรมข้อมูลมูลไปเรื่อยๆ เริ่มตั้งแต่ 64 ไบต์ (ขนาดเล็กที่สุดตามมาตรฐานของอีเทอร์เน็ต) ไปจนถึง 1,518 ไบต์ (ขนาดใหญ่ที่สุดตามมาตรฐานของอีเทอร์เน็ต) คอลัมน์ที่ 2 คือค่าของเอาต์พุตที่วัดได้ ที่ความเร็ว 100 Mbps อัตราความสามารถในการส่งข้อมูลมีหน่วยเป็น (pps : packets per second) จากข้อมูลดังกล่าวจะเห็นว่า แม้แพ็กเก็ตส่งไปได้มากๆ ก็จริงแต่ขนาดของข้อมูลต่อเฟรมน้อยจะทำให้อัตราการสื่อสารข้อมูลลดลงไปด้วย เช่น ในแถวที่ 1 จำนวนข้อมูลต่อ 1 เฟรมมีขนาดเท่ากับ 64 ไบต์ สามารถส่งข้อมูลได้ 148,809 แพ็กเก็ตต่อวินาที แต่ข้อมูลที่ส่งได้มี

ปริมาณแค่ 45% เท่านั้น ต่างจากแถวสุดท้าย คือขนาดของเฟรมข้อมูล มีขนาดใหญ่ที่สุดเท่ากับ 1,518 ไบต์ต่อเฟรม สามารถส่งข้อมูลได้ที่ความเร็ว 8,128 แพ็กเก็ตต่อวินาที ส่งผลให้ปริมาณข้อมูลที่วัดได้สูงถึง 100% คอลัมน์ที่ 3 แสดงปริมาณข้อมูลที่ไหลผ่านอินเทอร์เฟสที่ใช้ชิบ VIA ที่ความถี่ 1.5 GHz และคอลัมน์สุดท้ายเป็นปริมาณข้อมูลที่สามารถวัดได้มีหน่วยเป็นเปอร์เซ็นต์ จากตารางที่ 13.4 สรุปได้ว่าอัตราการส่งข้อมูลได้ 100% เมื่อเฟรมข้อมูลมีขนาดตั้งแต่ 256 ไบต์เป็นต้นไป

ผลการทดสอบประสิทธิภาพอินเทอร์เฟสชนิด Gigabit Ethernet

ตารางที่ 13.5 และ 13.6 แสดงผลการทดสอบความเร็วอินเทอร์เฟสชนิดกิกกะบิตอีเทอร์เน็ต โดยเลือกใช้เราเตอร์ที่มีหน่วยประมวลผลที่แตกต่างกัน และทดสอบแบบสองทิศทาง (Bidirectional) ผลจากการทดสอบเมื่อได้ 100% จะมีปริมาณของข้อมูลเป็น 2 เท่าของข้อมูลที่ใช้ทดสอบ (เท่ากับ 2 Gbps)

ตารางที่ 13.5 ผลการทดสอบประสิทธิภาพกิกกะบิตอีเทอร์เน็ต แบบ Bidirection (pps)

ขนาดของเฟรมข้อมูล (ไบต์)	ความเร็ว 1000 Mbps (สองทิศทาง : pps)	เราเตอร์ B ชิบแบบ VIA ความถี่ 1.5 GHz (pps)	เราเตอร์ C ชิบแบบ Intel celeron ความถี่ 2.8 GHz (pps)	เราเตอร์ D ชิบแบบ Intel celeron ความถี่ 2 GHz (pps)	เราเตอร์ E ชิบแบบ Intel celeron ความถี่ 2.53 GHz (pps)	เราเตอร์ F ชิบแบบ Intel Xeon ความถี่ 2.8 GHz (pps)
64	2,976,190	174,386	393,082	383,649	569,661	604,539
128	1,689,188	98,976	255,102	349,715	514,674	587,257
256	905,796	81,380	153,562	297,214	452,898	622,735
512	469,924	51,398	91,240	187,235	400,170	469,924
768	317,258	39,657	Untested	133,843	306,104	317,258
24	239,462	31,804	46,434	103,829	239,462	239,462
280	192,306	26,292	38,486	85,636	192,306	192,306
518	162,548	23,493	31,494	73,020	162,548	162,548

ตารางที่ 13.6 เปอร์เซ็นต์การทดสอบประสิทธิภาพกิกกะบิตอีเทอร์เน็ต ที่แบนด์วิดท์ 2 Gbps

ขนาดของเฟรมข้อมูล (ไบต์)	เราเตอร์ B ชิบแบบ VIA ความถี่ 1.5	เราเตอร์ C ชิบแบบ Intel celeron	เราเตอร์ D ชิบแบบ Intel celeron	เราเตอร์ E ชิบแบบ Intel celeron	เราเตอร์ F ชิบแบบ Intel Xeon

	GHz (pps)	ความถี่ 2.8 GHz (pps)	ความถี่ 2 GHz (pps)	ความถี่ 2.53 GHz (pps)	ความถี่ 2.8 GHz (pps)
64	6%	13%	13%	19%	20%
128	6%	15%	21%	30%	35%
256	9%	17%	33%	50%	69%
512	11%	19%	40%	85%	100%
768	13%	untested	42%	96%	100%
1024	13%	19%	43%	100%	100%
1280	14%	20%	45%	100%	100%
1518	14%	19%	45%	100%	100%

ตารางที่ 13.5 แสดงการทดสอบเราเตอร์แบบ A-F โดยใช้ยี่ห้อ ความเร็ว ของหน่วยประมวลผลที่แตกต่างกัน ทดสอบกับอินเทอร์เฟสที่ความเร็วในระดับกิกกะบิต และตารางที่ 13.6 เป็นผลลัพธ์จากการทดสอบ ผลที่ได้จะสังเกตุเห็นว่า ความเร็วของหน่วยประมวลผลที่มีความเร็วสูงๆ อาจจะไม่สามารถส่งข้อมูลได้สูงตามไปด้วย เช่น เราเตอร์แบบ C และ D ถึงแม้ว่า C จะมีความเร็วในการประมวลผลสูงกว่าแต่ผลลัพธ์ที่ได้แสดงให้เห็นว่า ส่งข้อมูลได้น้อยกว่าเราเตอร์แบบ D ซึ่งมีความเร็วในการประมวลผลน้อยกว่ามาก ทั้งนี้เนื่องจากมีปัจจัยอื่นๆ ที่ทำให้ความเร็วเพิ่มขึ้นได้อีก เช่น ไอโอบัส (IO Bus) เป็นต้น

สรุปผลการทดสอบ

หน่วยประมวลผลกลาง มีผลอย่างมากต่อประสิทธิภาพการ forward ข้อมูล ข้อมูลของแพ็กเก็ตที่มีขนาดเล็กจะทำให้การส่งข้อมูลลดลงไปด้วย เนื่องจากขณะที่เราเตอร์กำลังทำงาน เราเตอร์จำเป็นต้องถอดข้อมูลที่อยู่ปลายทางในทุกๆ แพ็กเก็ตที่ผ่านเข้ามา เมื่อแพ็กเก็ตมีขนาดเล็กเราเตอร์จำเป็นต้องถอดรหัสบ่อยครั้งมากกว่าแพ็กเก็ตที่มีขนาดใหญ่กว่า ตัวอย่างเช่น ถ้าข้อมูลมีขนาด 1,024 ไบต์ ในกรณีแรกทำการแบ่งข้อมูลเป็นแพ็กเก็ตย่อยๆ เพื่อส่งผ่านระบบเครือข่าย ในที่นี้สมมุติว่าใช้จำนวนเท่ากับ 256 ไบต์ ดังนั้นจะได้เท่ากับ 1,024/256 = 4 แพ็กเก็ต ดังนั้นในกรณีนี้เราเตอร์ต้องถอดข้อมูลทั้งหมด 4 ครั้ง ถ้าสมมุติว่าเวลาในการถอดครั้งละ 1 วินาที จะใช้เวลาทั้งหมด 4 วินาที ในกรณีที่สอง ถ้าแบ่งแพ็กเก็ตมีขนาด 1,024 ไบต์ต่อ 1 แพ็กเก็ตจะใช้เวลาเพียง 1 วินาทีเท่านั้น แต่ได้ปริมาณข้อมูลที่ส่งเท่ากัน แสดงให้เห็นว่าขนาดของแพ็กเก็ตมีผลต่อปริมาณการส่งข้อมูลด้วย จากข้อมูลในตารางเราเตอร์ C, D, E ใช้หน่วยประมวลผลรหัสเดียวกันคือ Intel Celeron แต่แตกต่างกันที่ความเร็วในการประมวลผล ผลที่ได้เราเตอร์แบบ E ให้ประสิทธ์ภาพดีที่สุด สำหรับเราเตอร์แบบ B ใช้ความเร็วหน่วยประมวลผลเท่ากับ 1.5 GHz ผลที่ได้คือมีความสามารถในการส่งข้อมูลได้น้อยที่สุด

ส่วนเราเตอร์ F ใช้รหัสหน่วยประมวลผลคือ Intel Xeon ผลที่ได้คือสามารถส่งข้อมูลได้สูงที่สุด และทำงานได้ดีแม้ว่าขนาดของแพ็กเก็ตข้อมูลจะมีขนาดเล็กก็ตาม ยังมีปัจจัยอีกบางประการที่ทำให้อัตราการส่งข้อมูลสูงต่ำต่างกันคือ ขนาดช่องสัญญาณของ IO bus, ขนาดของหน่วยความจำ, คุณภาพของอินเทอร์เฟสการ์ด และจำนวนของหน่วยประมวลผลกลาง (single-core, muti-core) ที่เลือกใช้งานด้วย

I/O Bus [30]

ปัจจัยสำคัญที่ส่งผลให้ความเร็วในการส่งข้อมูลเพิ่มขึ้นคือ ระบบโครงสร้างของไอโอบัส ในสถาปัตยกรรมแบบ x86 จะมีบัสที่ใช้งานในปัจจุบันหลายประเภท เช่น ISA, PCI และ AGP สำหรับมาตรฐานที่ได้รับความนิยมมากคือ PCI ซึ่งก็มีหลายชนิดคือ PCI, PCI-X และ PCI-Express [31] และทั้ง 3 ชนิดดังกล่าวก็มีอัตราการส่งข้อมูลที่แตกต่างกัน จากตารางที่ 13.6 แสดงให้เห็นถึงความเร็วในการสื่อสารข้อมูลของ PCI ที่มีความเร็วต่างกัน PCI และ PCI-X เป็นบัสตระกูลเดียวกัน โดย PCI-X ได้เพิ่มความเร็วในการสื่อสารข้อมูลได้มากกว่า แต่สามารถใช้งานร่วมกับบัส PCI ธรรมดาได้ PCI Express (3GIO หรือ 3rd Generation I/O โดยมี ISA เป็น 1st Generation และ PCI เป็น 2nd Generation) ซึ่งเป็น Technology PCI แบบใหม่ที่ยอมให้อุปกรณ์ภายในคอมพิวเตอร์ เช่น CPU ติดต่อโดยตรงกับอุปกรณ์ที่ต้องการติดต่ออยู่ได้ทั้งหมด โดยแยกจากกันเป็นอิสระ และสามารถเชื่อมต่อกันได้เต็มแบนด์วิดท์ (bandwidth) ซึ่งเป็นแบบ Point to Point หรือติดต่อกันโดยไม่ต้องแบ่งแบนด์วิดท์กับอุปกรณ์ต่อพ่วงอื่นๆเลย สถาปัตยกรรมนี้ก็ยังคำนึงถึงผู้บริโภคเป็นหลัก กล่าวคือในระบบใหม่นี้สามารถทดแทน PCI แบบเดิมอยู่ ถึงแม้จะมีรูปแบบของ PCI-Express แล้วก็ตาม เพื่อลดปัญหาการสับสนระหว่าง PCI กับ PCI-Express ลง โดยทำให้ PCI Card แบบเก่านั้นไม่สามารถใส่ในช่อง PCI-Express ได้ และ PCI-Express Card ก็ไม่สามารถใส่ในช่อง PCI แบบเก่าได้เช่นกัน อีกอย่างหนึ่งคือ PCI-Express นี้ สามารถรองรับ Gigabit Ethernet, 10 Gigabit Ethernet, 1394b หรือ อุปกรณ์ที่มีความเร็วสูงได้เป็นอย่างดี จากรูปที่ 13.7 แสดงการเปรียบเทียบมาตรฐาน PCI

ตารางที่ 13.7 เปรียบเทียบมาตรฐานของ PCI

	มาตรฐาน PCI	**มาตรฐาน PCI-X**	**มาตรฐาน PCI-Express**
เทคโนโลยี	ใช้บัสร่วมกัน	ใช้บัสร่วมกัน	ส่งข้อมูล 2 ทิศทาง, Point-to-Point, Swithed
ความกว้างของบัส	32 bits	64 bits	1 บิตต่อ lane (serial), สื่อสาร 2 ทิศทาง
ความถี่ของบัส	33 MHz	66, 100, or 133 MHz	2.5 GHz ต่อ lane
ความเร็วสูงสุดต่อวินาที (ทางทฤษฎี)	1.056 Gbps	8.512 Gbps @ 133 MHz	2 Gbps ต่อ 1 ทิศทาง ต่อ lane

Routed สูงสุดต่อวินาที (ทางทฤษฎี)	528 Mbps	4.256 Gbps@133 MHz	2 Gbps ต่อ lane
ความเข้ากันได้ของระบบ (compatable)	PCI การ์ด	PCI และ PCI-X การ์ด	ใช้กับ PCI- Express อย่างเดียวเท่านั้น

เมื่อกลับไปพิจารณาข้อมูลในตารางที่ 13.5 และ 13.6 เปรียบเทียบกับตารางที่ 13.7 แล้ว เราเตอร์ C มี Troughput ที่ 400 Mbps (20% ของ 2 Gbps) ที่ความเร็วหน่วยประมวลผล 2.8 Gbps แสดงว่าจะใช้บัสแบบ PCI ธรรมดาที่มีการแชร์ข้อมูลบนบัส ณ จุดพิจารณาเดียวกันเราเตอร์ D สามารถสื่อสารข้อมูลได้มากว่าคือ 900 Mbps (45% ของ 2 Gbps) แสดงว่าเราเตอร์ D ใช้บัสที่ไม่แชร์บัสร่วมกันทำให้อัตราการส่งข้อมูลเพิ่มขึ้นมากว่า 2 เท่า สำหรับเราเตอร์ E และ F สามารถส่งข้อมูลได้เต็มความเร็วคือ 2 Gbps แสดงว่าเราเตอร์ทั้งสองใช้บัสแบบ PCI-Express

จากผลการทดสอบในตารางสามารถสรุปข้อมูลเกี่ยวกับ I/O บัสไว้ให้พิจารณาดังต่อไปนี้

- ถ้าต้องการความเร็วในการสื่อสารระดับ Fast Ethernet (100 Mbps) สามารถใช้บัสชนิด PCI ธรรมดาได้ และที่ความเร็ว 1 Gbps ก็สามารถใช้ได้ แต่ไม่แนะนำให้ใช้งาน
- บัสแบบ PCI-X สามารถรองรับความเร็วที่ระดับ 1 Gbps ได้ แต่ไม่สนับสนุนความเร็ว 10 Gbps
- บัสแบบ PCI-Express ใช้งานแทน PCI, PCI-X ได้เป็นอย่างดี แต่ก็ยังไม่สามารถทำความเร็วระดับ 10 Gbps ได้เช่นเดียวกัน แต่ถ้าต้องการใช้งานพอร์ต 1 Gbps หลายๆ พอร์ต ให้ทำการเลือกการ์ดที่มีจำนวนพอร์ตต่อการ์ดสูงตามจำนวนที่ต้องการ โดยมากจะเป็น 2, 4, 6, 8 พอร์ตต่อการ์ด แต่มีข้อเสียคือจะเกิดคอขวดบนการ์ดแทน เพราะว่าถ้าใช้ 4 พอร์ต กิกกะบิต (4 x 1 Gbps) แต่ Troughput ของ PCI-Express รองรับสูงสุดได้เพียง 2 Gbps เท่านั้น

(a) PCI-Express (2 ports) (b) PCI-Express (4 ports) (c) PCI-Express (6 ports)

รูปที่ 13.13 แสดงการ PCI-Express

หน่วยความจำหลัก

ข้อมูลในระบบคอมพิวเตอร์ที่ต้องการประมวลผลจำเป็นต้องนำมาเก็บไว้บนหน่วยความจำหลักเสียก่อน เช่นเดียวกับข้อมูลที่สื่อสารกันภายในเครือข่าย ข้อมูลจะต้องถูกถอดรหัสบ้าง เข้ารหัสบ้าง ประมวลผลบ้าง ดังนั้นหน่วยความจำเป็นจุดที่ต้องพิจารณาในการสร้างเราเตอร์เป็นอย่างมาก แม้ว่าปัจจุบันหน่วยความจำหลักจะสามารถส่งถ่ายข้อมูลที่ความเร็วได้ตั้งแต่ 80 – 160 Gbps แต่ว่าหน่วยความจำหลักในคอมพิวเตอร์จะต้องมีข้อมูลอื่นๆ เก็บไว้ประมวลผลในหน่วยความจำด้วย เช่น ระบบปฏิบัติการ ซอฟต์แวร์ระบบ และอุปกรณ์ต่อพ่วงอื่น ๆ สำหรับการทดสอบหน่วยความจำยังไม่ได้มีการทดสอบอย่างจริงจังมากนัก แต่ถ้าวิเคราะห์ข้อมูลจากตารางที่ 13.1 หน่วยความจำที่มีขนาดตั้งแต่ 2 Gbps จะทำให้เราเตอร์ทำงานได้อย่างไม่มีปัญหา (การทดสอบหน่วยความจำกับโอเพนซอร์สเราเตอร์ยังไม่มีผลการวิจัยออกมาเป็นรูปธรรม) เมื่อต้องการสร้างเราเตอร์ควรพิจารณาให้มีหน่วยความจำมากที่สุดเท่าที่จะทำได้ เพราะว่าในอนาคตเราเตอร์อาจจะมีความจำเป็นต้องเปิดใช้โพรโทคอลที่ต้องเก็บข้อมูลตารางเส้นทางเพิ่มขึ้น หรือจำเป็นต้องเปิดใช้งานหลายๆ โพรโทคอลในเราเตอร์เพียงเครื่องเดียวก็เป็นไปได้ (หน่วยความจำที่ให้กับเราเตอร์ไม่ควรรวมกับหน่วยความจำที่ใช้งานกับระบบปฏิบัติการ ซอฟต์แวร์ระบบ และซอฟต์แวร์ประยุกต์อื่นๆ)

เน็ตเวิร์คอินเทอร์เฟส [32]

ประสิทธิภาพการทำงานของการ์ดเน็ตเวิร์คมีผลอย่างมากต่อการทำงานของเราเตอร์ เน็ตเวิร์คการ์ดยุกเดิมใช้วิธีการเขียนโปรแกรมคอยตรวจสอบข้อมูลที่ส่งออกและรับเข้ามาจากการ์ดเน็ตเวิร์คเอง ทำให้ต้องเสียเวลาหน่วยประมวลผลในการตรวจสอบการเข้ามาถึงของข้อมูล และเมื่อมาถึงซีพียูจะต้องทำหน้าที่ส่งถ่ายข้อมูลจากต้นทางไปยังปลายทางให้ถูกต้องและครบถ้วนด้วย ทำให้ความเร็วในการสื่อสารข้อมูลสมัยเก่าช้ามาก แต่ในการ์ดเน็ตเวิร์คยุคปัจจุบันมีเทคโนโลยีที่เรียกว่า DMA [33] สามารถเพิ่มความเร็วในการสื่อสารข้อมูลได้ตั้งแต่ 10 Mbps, 100 Mbps, 1 Gbps คุณสมบัติพิเศษของ DMA คือสามารถส่งข้อมูลจากการ์ดเน็ตเวิร์คไปยังหน่วยความจำหลักได้ทันทีโดยไม่ต้องรบกวนการทำงานของซีพียูเลย ไม่เหมือนระบบเดิมที่ซีพียูต้องทำการตรวจสอบเอง

(a) Ethernet Card (b) T1/E1 Card

รูปที่ 13.14 แสดงการ์ดเน็ตเวิร์คชนิดต่างๆ

ข้อมูลภาพอ้างอิงจาก: http://www.shopricom.com/

http://people.msoe.edu/~westabyd/XBC%20wireless%20guide/ethernet%20card.jpg

เมื่อโอนถ่ายข้อมูลเสร็จแล้วจึงค่อยส่งสัญญาณไปบอกซีพียูว่าเสร็จแล้ว ซีพียูจึงค่อยนำเอาข้อมูลเหล่านั้นไปทำงานต่อไป การขัดจังหวะซีพียู (Interupt) เป็นปัจจัยสำคัญที่ส่งผลให้การประมวลผลต่ำลง สมมุติว่าถ้าข้อมูลแต่ละแพ็กเก็ตที่มาถึงจะต้องร้องขอให้ซีพียูตรวจสอบ ถ้าจำนวนแพ็กเก็ตมีขนาดเล็กๆ จะต้องตรวจสอบหลายครั้งมากกว่าแพ็กเก็ตที่มีขนาดใหญ่ๆ ในการแก้ปัญหานี้การ์ดเน็ตเวิร์คในปัจจุบันได้ใช้เทคนิคที่เรียกว่า polled interrupt คือปล่อยให้ข้อมูลที่เข้ามายังระบบสะสมจำนวนไปสักพักหนึ่งจนถึงค่าค่าหนึ่งที่ตั้งไว้ คือ threshold เมื่อถึงจุดที่ตั้งไว้แล้วจะทำการเซ็ตบิตข้อมูลของรีจีสเตอร์ตรวจสอบสถานะ (status register) เป็นการบอกให้ซีพียูทราบว่าข้อมูลเสร็จเรียบร้อยแล้ว ซีพียูจึงนำข้อมูลนั้นๆ ซึ่งมีขนาดใหญ่ไปประมวลเสร็จภายในครั้งเดียวเท่านั้น เทคนิคแบบนี้ทำให้ซีพียูลดจำนวนการขัดจังหวะของข้อมูลที่ต้องการประมวลผลได้ ส่งผลให้ประสิทธิภาพโดยรวมของการสื่อสารข้อมูลสูงขึ้น สำหรับบนลีนุกซ์ซึ่งเป็นระบบปฏิบัติการที่ใช้สร้างเราเตอร์นั้นเรียกวิธีนี้ว่า NAPI

หน่วยประมวลผลแบบหลายคอและแบบหลายตัว (Multi-core & Multiprocessor)

สำหรับการทำสอบโอเพนซอร์สเราเตอร์กับหน่วยประมวลผลแบบหลายคอและแบบหลายตัวยังไม่มีการยืนยันประสิทธิภาพอย่างชัดเจน ซึ่งจำเป็นต้องวิจัยและทดสอบกันต่อไป ซึ่งอาจจะต้องมีการปรับแต่งเคอร์เนลของลินุกซ์เพื่อให้สนับสนุนหน่วยประมวลผลดังกล่าวก่อน แต่ปัจจุบันมีบริษัทยักใหญ่หลายบริษัทที่สนับสนุนลินุกซ์เต็มที่ ล่าสุดลินุกซ์เคอร์เนล 2.6 [34] สนับสนุนการทำงานแบบ multiprocessor แล้ว

สรุปการเลือกฮาร์ดแวร์ในการสร้างโอเพนซอร์สเราเตอร์

- เราเตอร์จำเป็นต้องมีหลายๆ อินเทอร์เฟส เทคโนโลยีที่นิยมในปัจจุบันคืออีเทอร์เน็ต ที่ระดับความเร็ว 1 Gbps ไม่ควรใช้การ์ด PCI เพียงการ์ดเดียวแต่มีจำนวนหลายๆ พอร์ต เพราะจะทำให้เกิดคอขวดในระบบและในบางครั้งระบบจะล้มเหลว (Hang) ได้
- เมื่อจำเป็นต้องสื่อสารข้อมูลที่ความเร็วสูงกว่า 400 Mbps ควรใช้การ์ดแบบ PCI-Express ดีกว่า แต่ถ้าไม่สามารถจัดหาได้ควรใช้ PCI-X แทน PCI ธรรมดา
- ชนิดของหน่วยประมวลผลและความเร็วมีผลต่อขนาดของแพ็กเก็ตที่เล็กๆ แต่ถ้าหลีกเลี่ยงเครือข่ายที่จำเป็นต้องใช้ขนาดของแพ็กเก็ตที่เล็กๆ แนะนำให้เลือกหน่วยประมวลผลที่มีความเร็วสูงๆ ไว้ก่อน แต่ถ้าสามารถจัดการหรือควบคุมขนาดของแพ็กเก็ตบนเครือข่ายได้ก็สามารถใช้หน่วยประมวลผลที่มีความเร็วต่ำๆ ได้ ทำให้สามารถประหวัดงบประมาณไปได้บางส่วน
- ชนิดของหน่วยประมวลผลและความเร็วมีผลต่อซอฟต์แวร์ที่ทำหน้าที่พิเศษบางอย่าง เช่น IDS, ไฟล์วอลล์ เนื่องจากจำเป็นต้องใช้ซีพียูในการเข้ารหัสข้อมูล ตรวจสอบความ

ผิดปกติของแพ็กเก็ต จำเป็นจะต้องใช้หน่วยประมวลผลที่มีความเร็วสูงๆ จึงจะให้ประสิทธิภาพที่ดี

- ขนาดของหน่วยประมวผลมีผลต่อความเร็วของเราเตอร์พอสมควร เมื่อต้องการสร้างเราเตอร์ควรเลือกหน่วยความจำอย่างน้อยไม่ควรต่ำกว่า 2 GB
- เน็ตเวิร์คอินเทอร์เฟสมีการขัดจังหวะการทำงานของซีพียู แต่ปัจจุบันสามารถแก้ไขให้ดีขึ้นได้แล้ว (บนลินุกซ์สามารถใช้งานคุณสมบัติข้อนี้ได้อย่างเต็มที่) ในหัวข้อนี้จึงไม่จำเป็นต้องกังวล
- หน่วยประมวลผลแบบหลายคอและแบบหลายตัว มีผลต่อการสื่อสารข้อมูล สำหรับในหนังสือเล่มนี้จะยังไม่เขียนไว้ เนื่องจากยังไม่มีการทดสอบออกมาอย่างเป็นรูปธรรม

13.2.2 รายการฮาร์ดแวร์ที่ผ่านการทดสอบและสนับสนุนโอเพนซอร์สเราเตอร์

ในส่วนนี้จะแสดงรายการของเครื่องเซิร์ฟเวอร์ที่ได้รับการทดสอบแล้วว่าสามารถใช้สร้างโอเพนซอร์สเราเตอร์ได้ [35] สำหรับฮาร์ดแวร์ที่ไม่ได้อยู่ในรายการที่แสดงไว้นี้ ถ้าเป็นสถาปัตยกรรมแบบ x86 สามารถทำงานได้เกือบทั้งหมด

รายการเซิร์ฟเวอร์ฮาร์ดแวร์

ผู้ผลิต (Vendor)	รุ่น (Model)	ขนาด (Form Factor)	ซีพียู (Processor)
Via (ประกอบเอง)	epia 5000	mini-itx, less than 10 watts power usage	via C3, 128 mb ram, 1GB compact flash IDE "hard drive"
Supermicro	SYS-5015B-MRB	1U rack moun 200Watt	Intel Dual Core Xeon 3.0GHz
Sample	Superbad 2000	2RU Server	Intel Core(TM)2 Duo CPU E4600 @ 2.40GHz
PC Engines	ALIX 2D3	SBC	AMD Geode (500Mhz)
oracle	N/A	8.0	N/A
IBM	x336	1U	Xeon
IBM	x Series 305	1U	2.0GHz Pentium 4

IBM	x Series 325	1U	Dual AMD Opteron
HP/Compaq	Proliant DL360G2	1U	Dusl 1.4GHz Pentium III
HP	DL 380	Server 2U	2 x (Xeon 3,2 Ghz + HT)
HP	T5720 Thin Client	SFF Desktop	AMD Geode NX 1Ghz
DMP Electronics Inc.	eBox-2300	Mini ITX Box 115 x 115 x 35 mm	Vortex86 SoC-200MHz (Fanless) 128MB SDRAM onboard
Dell™	OptiPlex™ GX110	Desktop (Low-Profile Chassis)	Intel Pentium III (Coppermine)
Dell	Poweredge 1950	1U rack mount	2 Dual-Core Xeon X526 @ 3.33 GHz
DELL	PowerEdge 2950	Rack Server	Quad-Core Intel
DELL	T100	Tower	Intel Xeon Quad Core
Compaq	Deskpro EN	Desktop (SFF)	Pentium III
ASUS	RS100-X5-PI2/2.0G /1G/250G	1U Rack	Intel Core Duo 2
ASUS	asus rs100-e4	Dual Core Xeon	intel dual core xeon
AGNET	ECS	Desktop Small Factor	AMD
Dell	PowerEdge	1U Server	N/A
IBM	BladeCenter	Server Blade	Dual Core Opteron, Dual Core Intel Xeon,
IBM	System X	1U server	N/A
Source Code	Thinkmate	1U server	Dual Core

Sun	Sun Fire	2U server	Dual Core AMD Opteron

รายการอินเทอร์เฟสฮาร์ดแวร์

ผู้ผลิต (Vendor)	รุ่น (Model)	WAN/LAN Connection	ชนิดของบัส (bus type)
VMWARE	Vmware Fusion 2	LAN	PCI
Sample	A123	T3, 1 Port	PCI
Routerboard	R44	10/100	PCI
Linksys	WMP300N	WLAN	PCI
Intel®	Intel® PRO/1000 PF	GigE Fiber Optic	PCI Express
Intel®	Intel PRO VT PCIe-4 Quad	GigE	PCI Express
Intel	Intel(R) PRO/1000 Network	LAN 1 GigE	PCI-X/PCI
HP	NC7170	Dual GigE	PCI-X/PCI
HP	T5720 Thin Client	10/100 Onboard, SIS chipset, but covered by the VT6102 [Rhine-II] Driver	PCI
D-Link System Inc	DGE-530T Gigabit Ethernet	GigE	PCI
3Com Corporation	3c905C-TX/TX-M	100BaseTX	PCI
3Com Corporation	3c905B	100BaseTX	PCI
Sangoma	S518	ADSL	PCI
Sangoma	A101	T1/E1, 1 Port	PCI
Sangoma	A102	T1/E1, 2 Port	PCI
Sangoma	A104	T1/E1, 4 Port	PCI
Sangoma	A301	T3, 1 Port	PCI
Sangoma	A142	Dual Serial Card	PCI

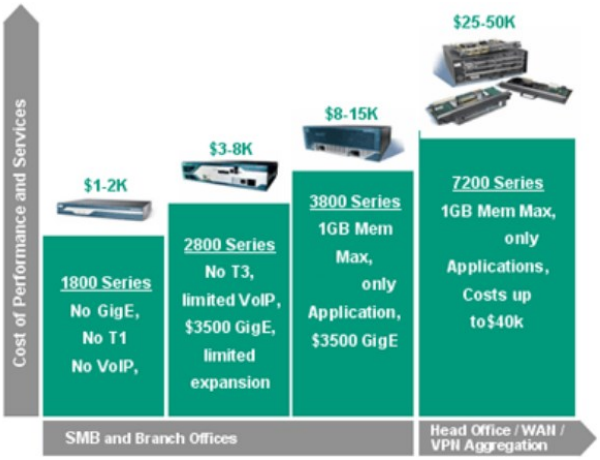
Sangoma	A144	Quad Serial Card	PCI
Neterion	Xframe E	10GbE	PCI Express
Intel	EXPI9404PT	Quad port 10/100/1000	PCI Express
Intel	EXPI9402PT	Dual port 10/100/1000	PCI Express
Intel	EXPI9400PT	10/100/1000	PCI Express
Intel	PWLA8494GT	Quad Port 10/100/1000	PCI-X / PCI
Intel	PWLA8494MT	Quad Port 10/100/1000	PCI-X / PCI
Intel	PWLA8494MT	Dual Port 10/100/1000	PCI-X / PCI
Intel	PWLA8490MT	10/100/1000	PCI-X / PCI
Intel	PWLA8490MT	10/100/1000	PCI

13.2.3 เปรียบเทียบประสิทธิภาพระหว่าง Specialized hardware กับสถาปัตยกรรมแบบ x86

ในส่วนนี้จะทำการเปรียบเทียบคุณสมบัติของฮาร์ดแวร์ที่ถูกสร้างมาสำหรับใช้งานเป็นเราเตอร์โดยเฉพาะกับฮาร์ดแวร์ตระกูล x86 สำหรับสร้างโอเพนซอร์สเราเตอร์ ปัจจุบันมีหลายบริษัทที่เป็นผู้นำด้านเทคโนโลยีของเครือข่าย เช่น บริษัท Cisco [36], Juniper [37], Linksys [38], D-Link [39] เป็นต้น หนึ่งในจำนวนนั้นบริษัท Cisco ถือว่าเป็นผู้นำด้านอุปกรณ์กรระบบเครือข่ายในปัจจุบัน ดังนั้นในหัวข้อนี้จะการนำเสนอข้อมูลการเปรียบเทียบคุณสมบัติหลายๆ ด้านระหว่างอุปกรณ์ของบริษัทผลิตอุปกรณ์เครือข่ายของบริษัทหนึ่ง (สมมุติว่าเป็นบริษัท A) กับสถาปัตยกรรมแบบ x86 โดย Vyatta [35] เป็นผู้เปรียบเทียบไว้ ดังนี้

PROPRIETARY HARDWARE – PURCHASE, AUCTION, REPLACE

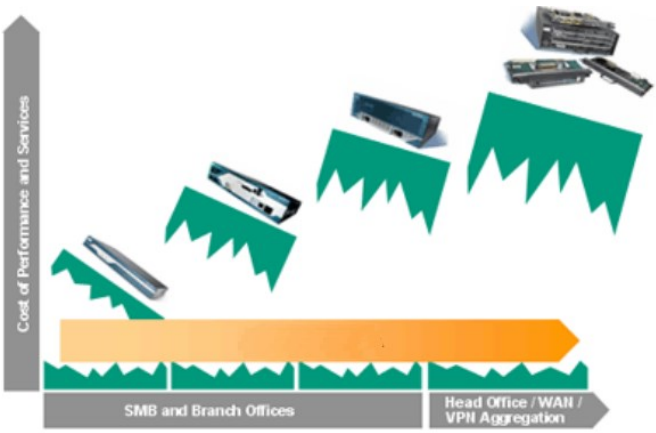
Vyatta ได้เสนอในรายงานว่า อุปกรณ์ด้านเครือข่ายของบริษัท A ค่อนข้างจำกัดคือความเร็วของซีพียูต่ำ หน่วยความจำก็มีอยู่อย่างจำกัด ในบางรุ่นไม่สามารถเพิ่มเติมไปได้อีก ทำให้เกิดข้อจะกัดในการใช้งานเป็นอย่างมาก และเมื่อต้องการเพิ่มคุณสมบัติใหม่เข้าไปในอุปกรณ์ บางรุ่นไม่สามารถทำได้ ถึงแม้ทำได้ก็จะมีราคาที่แพงมาก ตัวอย่างเช่น ไม่มีอินเทอร์เฟสสนับสนุนอย่างเพียงพอ มีหน่วยความจำที่จำกัดและไม่สามารถเพิ่มเติมได้อีก เป็นต้น



รูปที่ 13.15 แสดงการเปรียบเทียบข้อจำกัดของ Special Hardware

X86 HARDWARE - PRICE, AVAILABILITY, SCALABILITY

สำหรับโอเพนซอร์สเราเตอร์จะทำงานอยู่บนสถาปัตยกรรม x86 ทำให้ราคาต้นทุนต่อหน่วยต่ำลงอย่างมาก สามารถปรับแต่งอุปกรณ์ได้เอง เพิ่มเติมอุปกรณ์ใหม่ๆ เข้าไปได้โดยสะดวก ทำให้ในอนาคตสามารถขยายขนาดของระบบเครือข่ายได้อย่างไม่จำกัด ไม่มีข้อจำกัดทางด้านความเร็วของซีพียู หน่วยความจำ ระบบบัสและอินเทอร์เฟส เนื่องจากสามารถหาซื้อได้เองตามท้องตลอดได้ ประกอบกับราคาของเครื่องคอมพิวเตอร์ตระกูล x86 ถูกลงอย่างมากด้วยทำให้โอเพนซอร์สเราเตอร์เป็นที่น่าจับตามองเป็นอย่างมากในอนาคต



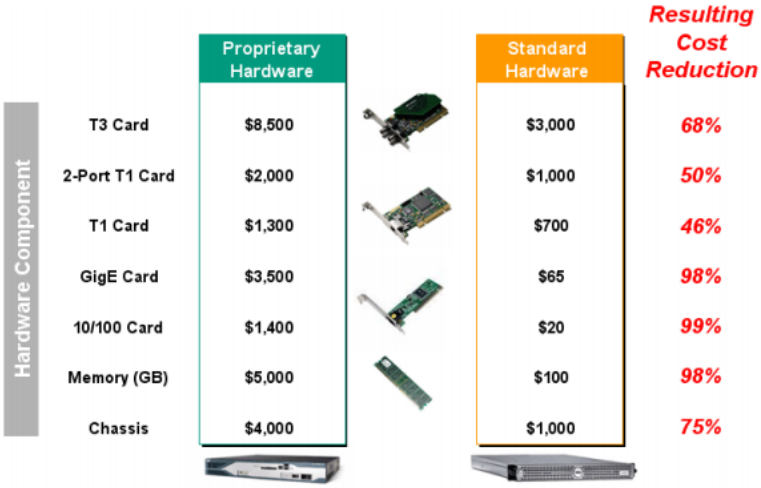
รูปที่ 13.16 แสดงการความได้เปรียบของโอเพนซอร์สเราเตอร์ด้านราคาและการขยายเครือข่าย

X86 PERFORMANCE - TWICE THE THROUGHPUT, HALF THE PRICE

เปรียบเทียบด้านราคา (PRICE COMPARISON)

เมื่อต้องการปรับปรุงระบบเครือข่ายให้มีความเร็วและประสิทธิภาพที่สูงขึ้น จำเป็นต้องเพิ่มงบประมาณในการจัดซื้อจัดจ้างหรืออัพเกรดอุปกรณ์เดิมให้มีสมรรถนะที่สูงขึ้น โดยปกติอุปกรณ์เครือข่ายที่ใช้ฮาร์ดแวร์จำเพาะจะคิดราคาอุปกรณ์ที่ต้องซื้อใหม่หรืออัพเกรดเพิ่มเติม ต่อหน่วยโดยประมาณคือ 100 เท่าของราคาอุปกรณ์ที่มีขายตามท้องตลาดทั่วไป สำหรับตัวอย่างเช่น ราคาของ

การ์ดอินเทอร์เฟสชนิดอีเทอร์เน็ตที่ใช้กับเครื่องเซิร์ฟเวอร์ ราคาต่อหน่วยอยู่ที่ประมาณ 700 บาท (20$ x 35) แต่เมื่อซื้ออุปกรณ์กับบริษัทที่ขายอุปกรณ์เครือข่ายราคาต่อหน่วยจะตกอยู่ที่ประมาณ 49,000 บาท (1,400$ x 35) ตัวอย่างที่สองคือราคาของหน่วยความจำขนาด 1GB ที่ใช้กับเซิร์ฟเวอร์ทั่วไปจะอยู่ที่ราคาต่อประมาณ 3,500 บาท (100$ x 35) แต่บริษัทจะขายในราคาต่อหน่วยประมาณ 175,000 บาท (5000$ x 35) ซึ่งแสดงไว้ในรูปที่ **13.17**



Hardware Component	Proprietary Hardware	Standard Hardware	Resulting Cost Reduction
T3 Card	$8,500	$3,000	68%
2-Port T1 Card	$2,000	$1,000	50%
T1 Card	$1,300	$700	46%
GigE Card	$3,500	$65	98%
10/100 Card	$1,400	$20	99%
Memory (GB)	$5,000	$100	98%
Chassis	$4,000	$1,000	75%

รูปที่ 13.17 เปรียบเทียบราคาอุปกรณ์ระหว่างโอเพนซอร์สเราเตอร์กับบริษัทขายอุปกรณ์เครือข่าย

การปรับเปลี่ยนหรือขยายโครงเครือข่าย

เป็นที่ทราบกันดีว่าเมื่อระบบเครือข่ายทำงานไปสักระยะหนึ่งแล้ว มีความจำเป็นอย่างยิ่งที่จะต้องปรับเปลี่ยนหรือทดแทน เพื่อให้ทันการเปลี่ยนแปลงของเทคโนโลยี ถ้าอุปกรณ์ที่ใช้ในเครือข่ายเป็นอุปกรณ์ที่เป็นฮาร์ดแวร์พิเศษเฉพาะจะปรับเปลี่ยนได้ยากมาก ถึงแม้ว่าปรับเปลี่ยนได้ก็ต้องใช้งบประมาณในการลงทุนสูง และเมื่อปรับปรุงไปแล้วก็มักจะปรับปรุงได้ไม่มากเนื่องจากข้อจำกัดของฮาร์ดแวร์เองเช่น เมนบอร์ดไม่สามารถเพิ่มอุปกรณ์ใหม่เข้าไปได้ หน่วยความจำเต็มไม่สามารถใส่เพิ่มได้ เมื่อสถานะการณ์เช่นนี้เกิดขึ้นจะทำให้ผู้ใช้อยู่ในสถานะจำยอมคือ อุปกรณ์เก่าก็ทิ้งไม่ได้เนื่องจากราคาแพงยังใช้ไม่คุ้มกับเงินที่ต้องลงทุนไป อุปกรณ์ใหม่หรือปรับปรุงเพิ่มก็ไม่ได้เพราะราคาสูงปรับปรุงไปก็ไม่คุ้มอีก ในที่สุดจึงต้องจำใจใช้งานต่อไปจนกว่าอุปกรณ์จะหมดอายุหรือถ้ามีงบประมาณเยอะก็อาจจะสามารถถอดทิ้ง หรือนำอุปกรณ์ดังกล่าวไปใช้ในสถานที่ที่ไม่ใช่จุดสำคัญของเครือข่าย แต่สำหรับโอเพนซอร์สเราเตอร์จะไม่คอยมีปัญหาตามที่ได้กล่าวมาเนื่องจาก เราเตอร์สามารถใช้คอมพิวเตอร์ตระกูล x86 ทั่วๆ ไปทำงานได้ มีผู้ผลิตมากและได้มาตรฐาน เช่น IBM, Dell, Sun, SuperMicro เป็นต้น หรือถ้าต้องการให้เราเตอร์มีประสิทธิภาพดีขึ้นอาจจะใช้เครื่องคอมพิวเตอร์รุ่นที่ใหญ่ขึ้น เช่น Server blades เป็นต้น ความโดดเด่นอีกประการหนึ่งคือโอเพนซอร์สเราเตอร์สามารถทำงานได้บนเครื่องจักรเสมือน (Virtual Machine เช่น Xen, Vmware, Virtual BOX) ซึ่งจะช่วยลดการจัดหาอุปกรณ์จำนวนมากได้

STANDARD HARDWARE PLATFORMS

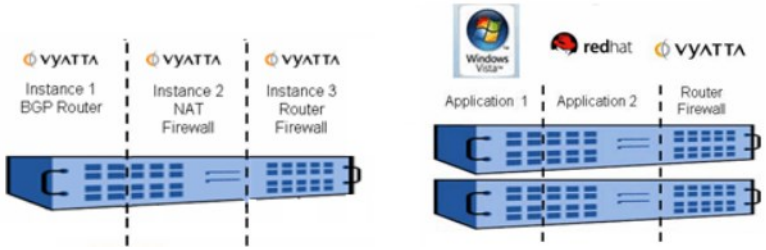
ผู้ใช้งานจะใช้ฮาร์ดแวร์ที่มีมาตรฐานเดียวกันดังนั้น ทุกๆ องค์กรที่ใช้งานสามารถแลกเปลี่ยนกันได้ทั้งทางด้านฮาร์ดแวร์ ซอฟต์แวร์ และพีเพิลแวร์ ในอนาคตได้

SERVER BLADES

เมื่อผู้ใช้ต้องการเราเตอร์ที่มีประสิทธิภาพในการประมวลผลที่สูงขึ้น ก็สามารถเลือกใช้เซิร์ฟเวอร์ที่มีประสิทธิภาพสูงขึ้น เช่น Blades ได้

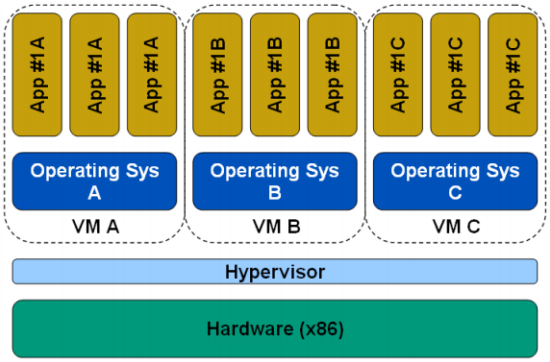
VIRTUALIZATION

ในปัจจุบันได้มีเทคโนโลยีที่เรียกว่า เครื่องจักรเสมือน (Virtual Machine) เป็นซอฟต์แวร์ที่ทำหน้าที่จำลองการทำงานของฮาร์ดแวร์จริงๆ เป็นฮาร์ดแวร์เสมือนได้หลายๆ เครื่อง ทำให้ผู้ใช้งานสามารถติดตั้งระบบปฏิบัติการที่แตกต่างกัน และสามารถทำงานได้พร้อมๆ กันบนฮาร์ดแวร์เครื่องเดียวกันได้ ทำให้สามารถประหยัดงบประมาณในการลงทุนซื้อฮาร์ดแวร์จำนวนมาก ข้อดีดังกล่าวนี้ได้ถูกนำเอาใช้ในระบบเครือข่ายด้วยคือ เมื่อผู้ดูแลระบบต้องการทดสอบความสามารถใหม่ๆ ของเราเตอร์ หรือต้องการทดสอบติดตั้งเราเตอร์ใหม่ ก็สามารถจำลองการทำงานบนเครื่องจักรเสมือนเสียก่อน ทำให้ไม่รบกวนการทำงานของเราเตอร์บนระบบเครือข่ายเดิมที่ทำงานอยู่แล้วได้ ดังรูปที่ 13.18, 13.19



(a) Horizontal Virualization (b) Vertical Virualization

รูปที่ 13.18 แสดงการใช้เครื่องจักรเสมือนกับแอพพลิเคชันและโอเพนซอร์สเราเตอร์



รูปที่ 13.19 ระบบปฏิบัติการหลายตัวบนเครื่องจักรเสมือนเครื่องเดียวกัน

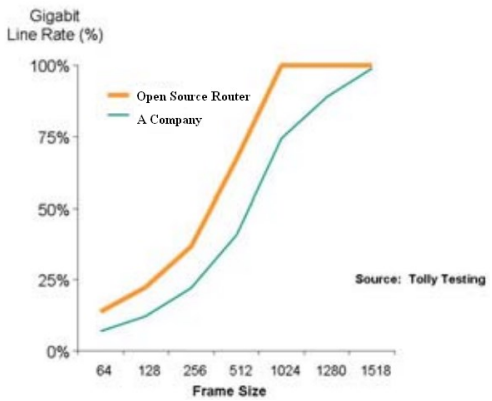
OPEN SOURCE – ADD & GO

โอเพนซอร์สเราเตอร์ถูกสร้างบนพื้นฐานของซอฟต์แวร์เสรี (Open Source) โดยพัฒนาอยู่บนระบบปฏิบัติการลินุกซ์ซึ่งในปัจจุบันกำลังได้รับความนิยามเป็นอย่างมาก เนื่องจากมีความเสถียร พัฒนาการต่อเนื่อง ผู้พัฒนาช่วยกันพัฒนาตัวอยู่ทั่วโลก มีลินุกซ์หลายสายพันธุ์ที่ยืนยันว่าสามารถ

ทำงานร่วมกันได้แบบไม่มีปัญหา (Debian, Quagga, IPtables, OpenSwan, OpenVPN, Snort) รวมทั้งซอฟต์แวร์ใหม่ๆ สามารถเพิ่มเติมในภายหลังได้เอง ซึ่งทำให้ในอนาคตโอเพนซอร์สเราเตอร์จะไม่ได้ทำหน้าที่เป็นเราเตอร์อย่างเดียวเท่านั้นแต่จะทำหน้าที่อื่นๆ ไปพร้อมๆ กันด้วย เช่น ไฟล์วอลล์, Intrusion Detection, Antivirus, Snort เป็นต้น

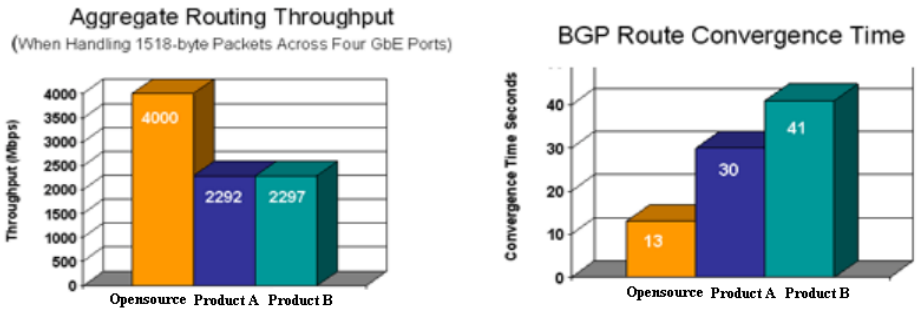
เปรียบเทียบความสามารถในการค้นหาเส้นทางและปริมาณการส่งข้อมูล

ในส่วนนี้จะแสดงการเปรียบเทียบประสิทธิการค้นหาเส้นทางและปริมาณข้อมูลที่สามารถส่งได้ ระหว่างฮาร์ดแวร์เราเตอร์ด้านเครือข่ายจำเพาะกับโอเพนซอร์สเราเตอร์ [40] จากรูปที่ 13.20 แสดงการเปรียบเทียบประสิทธิภาพการส่งข้อมูลระดับกิกกะบิตของบริษัท A (นามสมมุติ) กับโอเพนซอร์สเราเตอร์ จากรูปเห็นได้ว่าอัตราการส่งข้อมูลของโอเพนซอร์สเราเตอร์สูงกว่ามาก ในราคาที่ต่างกันด้วย Open source ใช้งบประมาณ 62,895 บาท แต่ Proprietary ใช้งบประมาณ 116,725 บาท



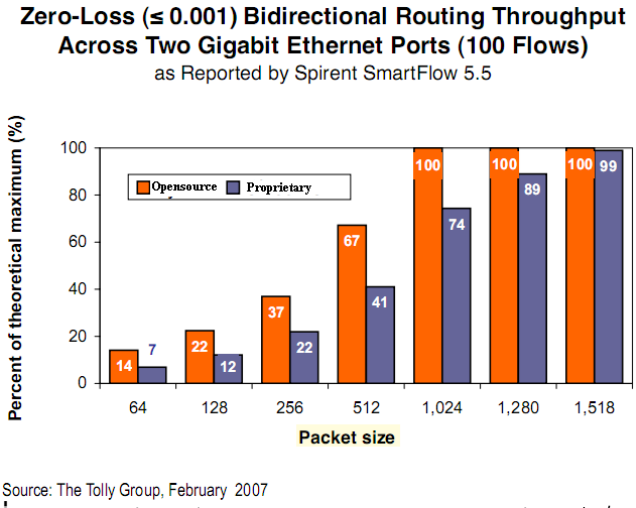
รูปที่ 13.20 เปรียบเทียบการส่งข้อมูลความเร็วระดับกิกกะบิต

รูปที่ 13.21 เปรียบเทียบประสิทธิภาพการค้นหาเส้นทาง (a) และ BGP Route Convergence Time (b) ผลจากการทดสอบเห็นได้ชัดว่าโอเพนซอร์สเราเตอร์สามารถทำงานได้ดีกว่า และใช้งบประมาณที่ต่างกันมากด้วยคือ Open source ใช้งบประมาณ 280,000 บาท ส่วน Proprietary ประมาณ 1,225,000



(a) Aggregate Routing Throughput (b) BGP Convergence Time

รูปที่ 13.21 เปรียบเทียบ Aggregate Routing Throughput & BGP Convergence Time [41]



รูปที่ 13.22 เปรียบเทียบ Sero-Less บนกิกกะบิตอีเทอร์เน็ต [42]

การเปรียบเทียบคุณสมบัติด้านอื่นๆ

สำหรับการเปรียบเทียบคุณสมบัติอื่นๆ เช่น Structural Economic Shift, STANDARD HARDWARE REDUCES PRICING POWER, Networking and the Myth of Special Hardware, OUTSPENT, OUTPERFORMED, VENDOR-DRIVEN SEGMENTATION & THE “BOX REPLACEMENT” MODEL, Standard Hardware Comes To Networking, Structurally Better Economics สามารถดูข้อมูลได้จาก http://www.vyatta.com; 2007 [43]

ศูนย์ข้อมูลแบบเสมือน การประมวลผลแบบแบ่งปันทรัพยากรผ่านเครือข่าย

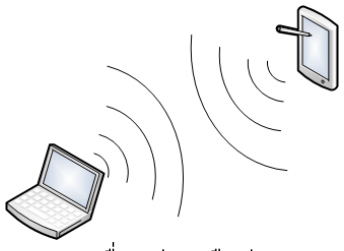
13.3 โมเดลการเชื่อมต่อเครือข่าย

เมื่อเราต้องการซื้อบ้านหรือที่อยู่อาศัย สิ่งแรกที่ควรจะทำต่อจากเรื่องงบประมาณคือ ดูแบบบ้านหรือโมเดลของบ้านว่าตรงใจผู้อยู่อาศัยหรือเปล่า ในกรณีแรกนี้สำหรับผู้ที่ไม่มีความรู้หรือพอจะมีความรู้เรื่องแบบบ้านมาบ้าง ก็ต้องอาศัยแบบหรือโมเดลที่เขาสร้างเอาไว้ให้แล้ว เป็นตัวอย่างนำทางไปก่อนแล้วค่อยต่อเติมทีหลังอีกที สำหรับกรณีที่สองผู้ซื้อมีความรู้เรื่องโครงสร้างเกี่ยวกับบ้านเป็นอย่างดีก็สามารถเขียนแบบได้เอง โดยไม่ต้องอาศัยวิศวกรเขียนแบบให้ หรือไม่ต้องอาศัยโมเดลที่เขาสร้างไว้เช่นเดียวกันกับระบบเครือข่าย ผู้ที่ไม่เคยได้ออกแบบระบบเครือข่ายหรือเพิ่งเรียนรู้เครือข่ายก็คงไม่สามารถจินตนาการโครงร่างของเครือข่ายได้ ต้องอาศัยโครงข่ายที่เขาสร้างไว้หรือทำเป็นแบบไว้ให้ เมื่อศึกษาเข้าใจแล้วก็สามารถปรับแต่ง เพิ่มถอน ให้เหมาะสมกับหน่วยงานหรือองค์กรของตนเองได้ต่อไป สำหรับผู้ที่มีความรู้เรื่องระบบเครือข่ายและเคยออกแบบเครือข่ายมาแล้วจนชำนาญแล้ว อาจจะไม่มีความจำเป็นต้องอ่านบทนี้ก็ได้ สามารถข้ามไปอ่านบทต่อไปได้เลย

จากทฤษฎีของระบบเครือข่ายแบ่งชนิดของเครือข่ายออกเป็น 4 ประเภทคือ

Personal Area Networks (PAN) [44] หรือเครือข่ายส่วนบุคคลเป็นเครือข่ายที่มีขนาดเล็กที่สุด โดยปกติจะเชื่อมต่อระหว่างคอมพิวเตอร์ส่วนบุคคลเข้ากับอุปกรณ์ต่อพ่วง เช่น เครื่องพิมพ์

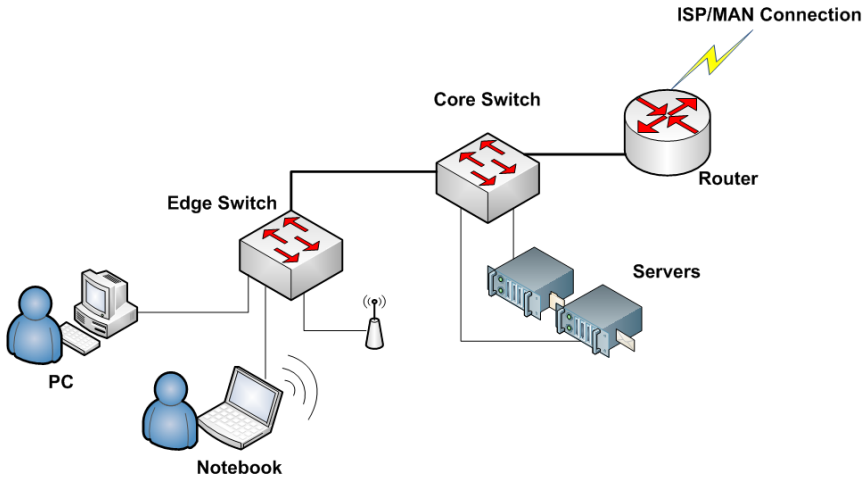
เครื่องเสียง หรือเชื่อมต่อเครื่องคอมพิวเตอร์ด้วยกันเอง เทคโนโลยีที่ใช้เช่น บลูทูธ (bluetooth), wireless ระยะการเชื่อมต่อประมาณ 1 เมตร ความเร็วประมาณ 10 เม็กกะบิตต่อวินาที (Mbps) ดังรูปที่ 13.23



รูปที่ 13.23 การเชื่อมต่อเครือข่ายประเภท PAN

Local Area Networks (LAN)

LAN หรือเครือข่ายท้องถิ่นเชื่อมโยงระบบคอมพิวเตอร์เข้าด้วยกันในระยะทางใกล้ๆ ระยะทางไม่เกิน 10 กิโลเมตร เช่นภายในตึก องค์กร หรือสำนักงาน ห้องสมุด หรือภายในมหาวิทยาลัย เพื่อแชร์ข้อมูลประเภทฐานข้อมูล เครื่องพิมพ์ หรือเครือข่ายอินเทอร์เน็ต เทคโนโลยีที่ใช้ปัจจุบันนิยมใช้อีเทอร์เน็ต (Ethernet) ความเร็วสูงถึง 10 กิกกะบิตต่อวินาที (Gbps) โครงสร้างหลักของเครือข่าย LAN จะประกอบด้วยอุปกรณ์สวิตช์ซิ่งที่มีความเร็วในการส่งถ่ายข้อมูลได้สูง เครื่องเซิร์ฟเวอร์ให้บริการจะเชื่อมต่อไปยังสวิตช์หลัก (switch layer 3) ที่สื่อสารได้รวดเร็ว และนิยมใช้เราเตอร์เชื่อมต่อเครือข่ายไปยังอินเทอร์เน็ต สื่อนำสัญญาณที่ใช้ เช่น Coaxial, UTP, STP, Fiber Optical, Wireless เป็นต้น ดังรูปที่ 13.24

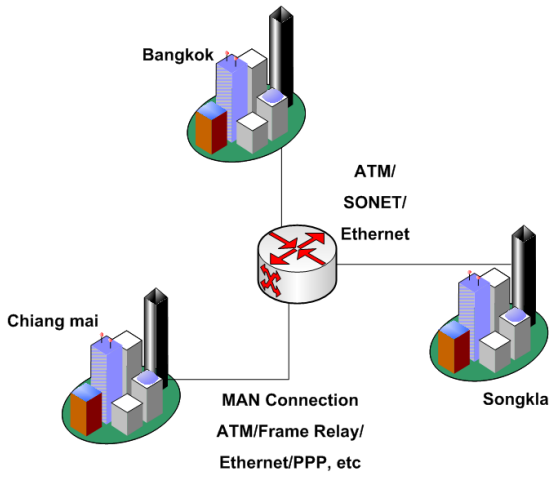


รูปที่ 13.24 การเชื่อมต่อเครือข่ายประเภท LAN

Metropolitan Area Networks (MAN)

MAN หรือเครือข่ายระดับเมือง เชื่อมโยงเครือข่ายที่อยู่ห่างไกลระดับจังหวัดหรือภูมิภาค ระยะทางประมาณ 100 กิโลเมตร สายนำสัญญาณที่ใช้เช่น Fiber Optical, Microware, Satellite เทคโนโลยีที่ใช้เช่น ATM, Frame Relay, xDSL, cable Modem หรือ Ethernet เป็นต้น MAN จะใช้เชื่อมโยงเครือข่ายแบบ LAN เข้าด้วยกัน ตัวอย่างเช่น สาขาใหญ่อยู่กรุงเทพฯ และต้องการเชื่อมไปยังสาขาย่อยๆ ต่างจังหวัด เช่น เชียงใหม่ สงขลา ซึ่งสาขาย่อยก็จะมีเครือข่ายที่ใช้งานของตนเองอยู่

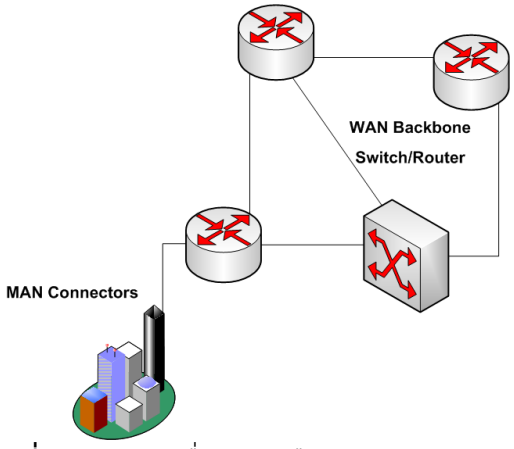
แล้ว ความเร็วในการเชื่อมต่อขึ้นอยู่กับงบประมาณ แต่ MAN สามารถสื่อสารข้อมูลได้สูงระดับ 10 Gbps ดังรูปที่ 13.25



รูปที่ 13.25 การเชื่อมต่อเครือข่ายประเภท MAN

Wide Area Networks (WAN)

WAN หรือแวนเป็นเครือข่ายที่ครอบคลุมพื้นที่กว้างไกลมาก ครอบคลุมไปทั่วโลก จะถูกใช้สำหรับเชื่อมโยงเครือข่ายระดับ MAN เข้าด้วยกัน และเชื่อมโยงเครือข่ายที่มี Topology ที่แตกต่างกันให้สามารสื่อสารกันได้ ภาพที่ชัดเจนที่สุดของ WAN ปัจจุบันคือเครือข่ายอินเทอร์เน็ต อุปกรณ์ที่ใช้เชื่อมต่อส่วนมากจะใช้เราเตอร์ หรือสวิตช์ที่สามารถหาเส้นทางได้ (switch layer 3) ความเร็วในการสื่อสารปัจจุบันสูงสุดอยู่ในระดับ 10 Gbps ดังรูปที่ 13.26



รูปที่ 13.26 การเชื่อมต่อเครือข่ายประเภท WAN

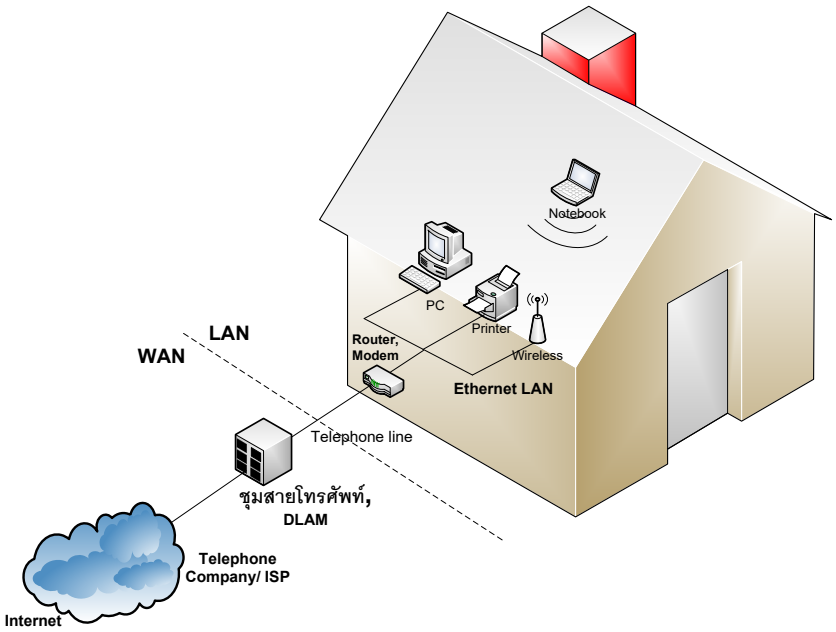
ถ้าพิจารณาชนิดของเครือข่ายทั้ง 4 ชนิดแล้ว PAN ไม่มีความจำเป็นต้องใช้เราเตอร์ ชนิดที่ 2 คือ LAN ถ้าขนาดของเครือข่ายใหญ่มากๆ ก็ควรจะใช้เราเตอร์ในการกำจัดแพ็กเก็ตที่ไม่พึงประสงค์ (Collision, Broadcast) ชนิดที่ 3 และ 4 จำเป็นต้องใช้เป็นอย่างมาก ฉนั้นในการสร้างโมเดลเครือข่ายในบทนี้จะออกแบบให้ครอบคลุมเครือข่ายทั้งระดับ LAN, MAN และ WAN แต่จะเน้นหนักไปที่เทคโนโลยีอีเทอร์เน็ต (Ethernet) เป็นหลักเพราะได้รับความนิยมสูง สำหรับเทคโนโลยีอื่นๆ เช่น T1/E1, T3/T3 เป็นต้นนั้น จะแตกต่างกันที่เลเยอร์ที่ 1 และ 2 เท่านั้น สำหรับเลเยอร์ที่ 3 และ 4 ส่วนใหญ่จะใช้โพรโทคอล TCP/IP

โมเดล A: เครือข่ายในที่พักอาศัย (Home Network)

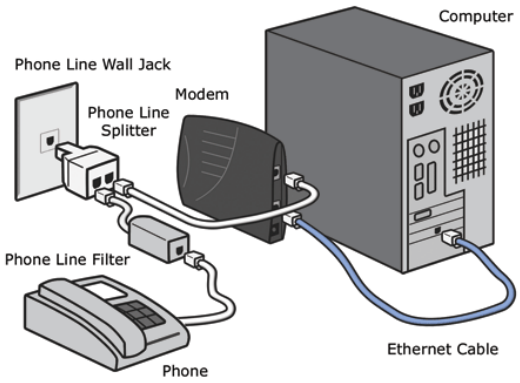
โมเดลการเชื่อมต่อเครือข่ายชนิด A เป็นโครงข่ายที่ใช้กันตามที่พักอาศัยส่วนใหญ่

โครงสร้างพื้นฐานด้านฮาร์ดแวร์เครือข่าย

- Router ADSL หรือ Modem ทำหน้าที่เชื่อมต่อไปยังชุมสายโทรศัพท์หรือ ISP ที่ให้บริการทางด้านอินเทอร์เน็ต ในยุคเดิมจะใช้อุปกรณ์โมเด็มทำหน้าที่แปลงสัญญาณดิจิทอลจากเครื่องคอมพิวเตอร์เป็นสัญญาณอนาล็อกส่งไปตามสายโทรศัพท์ ความเร็วสูงสุดอยู่ที่ 56 kbps (กิโลบิตต่อนาที) และใช้เทคโนโลยีการเชื่อมต่อระดับเลเยอร์ 2 แบบ PPP [45], PPPoE [46] หรือ SLIP สำหรับในยุคปัจจุบันนิยมใช้เทคโนโลยี ADSL [47] แทน เนื่องจากให้ความเร็วที่สูงกว่ามาก ปัจจุบันมีความเร็วได้ถึง 8 Mbps อุปกรณ์ที่ใช้ก็คือ โมเด็ม ADSL หรือ Router ADSL ดังรูปที่ 13.27, 13.28

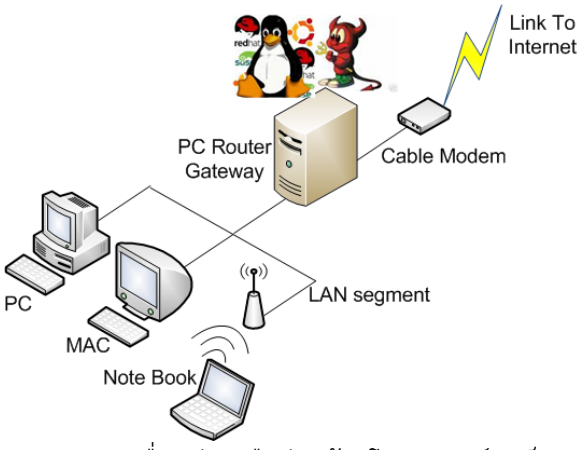


รูปที่ 13.27 โมเดลการเชื่อมต่อแบบ A เครือข่ายในที่พักอาศัย



รูปที่ 13.28 การเชื่อมต่อเครือข่ายด้วยโมเด็ม [48]

รูปที่ 13.29 การเชื่อมต่อเครือข่ายด้วย ADSL Modem หรือ Router [48]



รูปที่ 13.30 การเชื่อมต่อเครือข่ายด้วยโอเพนซอร์ส เป็นเกตเวย์

- สายนำสัญญาณโทรศัพท์ เชื่อมโยงมาจากตู้ชุมสายโทรศัพท์
- อุปกรณ์ ฮับหรือ สวิตช์ เมื่อผู้ใช้ต้องการขยายเครือข่ายภายในที่พักอาศัย หรือ อจจะใช้ wireless ก็จะสะดวกในการเคลื่อนย้ายหรือปรับเปลี่ยนเครื่องลูกข่ายภายในบ้าน
- สายนำสัญญาณชนิด CAT5, CAT5E ใช้สำหรับเชื่อมต่อเครือข่าย LAN จำนวน 2-5 เส้น พร้อมเข้าหัวสายแบบ RJ-45 [50]

ข้อดี

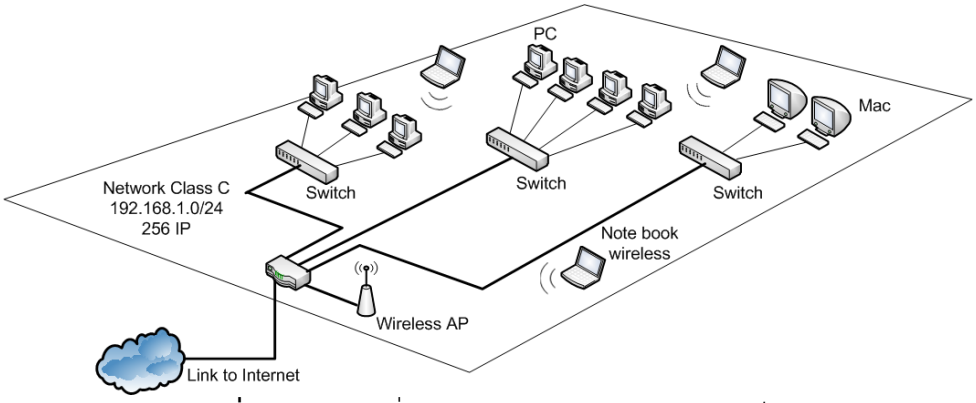
- ราคาของอุปกรณ์ถูก และหาได้ง่าย
- การเชื่อมต่อไม่ซับซ้อน
- ความเร็วขึ้นอยู่กับราคาค่าเช่ารายเดือน หรือ Package
- สะดวกในการเพิ่มลดหรือปรับเปลี่ยนอุปกรณ์ใหม่
- เหมาะสำหรับผู้ที่ใช้งานทั่วๆ ไป ใช้ความรู้ด้านเครือข่ายเล็กน้อยก็สามารถเชื่อต่อได้

ข้อเสีย

- ไม่สะดวกเมื่อจำเป็นต้องดาวน์โหลดข้อมูลที่มีขนาดใหญ่มากๆ

- เมื่อเป็น Package แบบเหมาจ่ายรายเดือน บางครั้งไม่คุ้มกับราคาที่เช่า เนื่องจากผู้ใช้ไม่ได้เปิดใช้งานอยู่ตลอดเวลา
- ความเร็วเมื่อถึงระดับหนึ่งจะไม่สามารถขยายต่อไปได้เนื่องจากเป็นข้อจำกัดของสายนำสัญญาณโทรศัพท์ (ADSL สามารถดาวน์โหลดข้อมูลสูงสุดได้ถึง 8 Mbps)
- ความสามารถในการรับส่งข้อมูลไม่เท่ากัน คือสามารถดาวน์โหลดข้อมูล (Downstream) มากกว่าอัพโหลดข้อมูล (Upstream)

โมเดล B: เครือข่ายสำนักงานขนาดเล็ก (Small Office Network)



รูปที่ 13.31 การเชื่อมต่อเครือข่ายสำนักงานขนาดเล็ก

สำนักงานขนาดเล็กจะประกอบไปด้วยเครื่องคอมพิวเตอร์ลูกข่ายจำนวนตั้งแต่ 5 – 30 เครื่องโดยประมาณ ภายในสำนักงานอาจจะแบ่งเครื่องคอมพิวเตอร์ออกเป็นกลุ่มๆ เช่นกลุ่มละ 1-24 เครื่อง เนื่องจากข้อจำกัดทางด้านอุปกรณ์ของสวิตช์ขนาดเล็กจะมีจำนวนพอร์ตจำกัด เช่น 2, 4, 8, 24 เป็นต้น เครื่องคอมพิวเตอร์ลูกข่ายทั้งหมดมักจะรวมกันอยู่ในห้องเดียวกัน

โครงสร้างพื้นฐานด้านฮาร์ดแวร์เครือข่าย

- เคเบิลโมเด็ม หรือ ADSL เราเตอร์ อย่างน้อย 1 ตัว และมีพอร์ตสำหรับเชื่อมต่อ LAN ไม่ควรต่ำกว่า 2 พอร์ต ถ้ามีเพียงพอร์ตเดียวให้ใช้สวิตช์ต่อพ่วงขยายเพิ่มได้
- สวิตช์เลเยอร์ 2 ที่มีจำนวนพอร์ตตั้งแต่ 4, 8, 16 พอร์ต จำนวน 1 – 4 เครื่อง
- Link เชื่อมต่อไปยังเครือข่ายอินเทอร์เน็ต สำหรับแบนด์วิดท์ขึ้นอยู่กับปริมาณการใช้งาน
- อุปกรณ์เชื่อมต่อแบบไร้สาย (Wireless LAN) เมื่อต้องการความสะดวกในการเคลื่อนย้าย

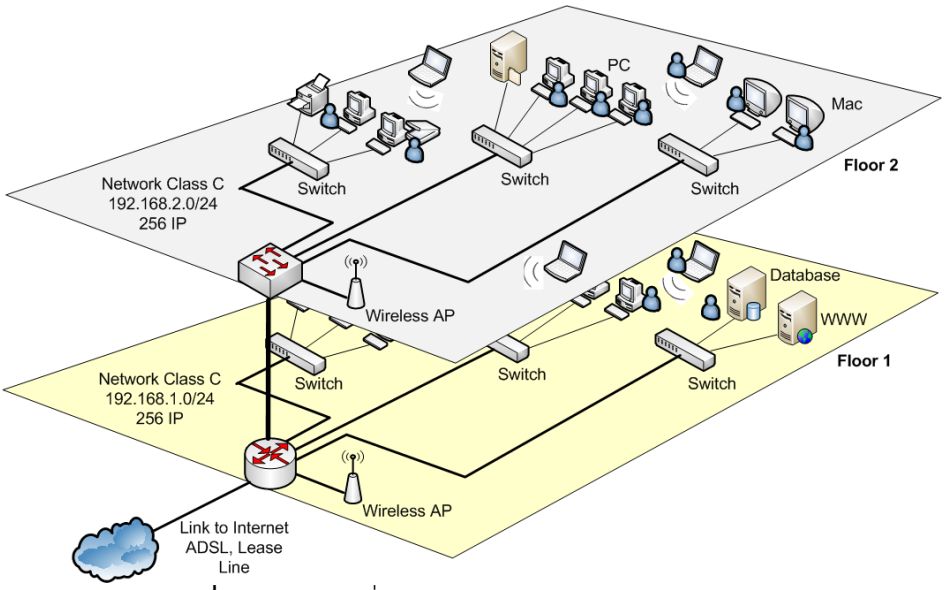
ข้อดี

- ข้อดีหลักๆ คล้ายกับเครือข่ายแบบ Home Network
- เครือข่ายไม่ซับซ้อนทำให้ออกแบบได้ง่าย ปรับปรุงและเปลี่ยนอุปกรณ์สามารถทำได้ทันที
- ค่าใช้จ่ายไม่สูง

ข้อเสีย

- ความเร็วในการใช้งานขึ้นอยู่กับจำนวนเครื่องลูกข่ายที่เชื่อมต่อ
- ข้อมูล broadcast เริ่มมีจำนวนมากขึ้นตามจำนวนเครื่องลูกข่าย
- กรณีเครื่องลูกข่ายติดไวรัส (Virus) หรือหนอนอินเทอร์เน็ต (Worm) ส่งผลให้เครือข่ายทำงานช้าลง
- เราเตอร์ ADSL หรือโมเด็มทำงานหนัก

โมเดล C: เครือข่ายสำนักงานขนาดกลาง (Medium Office Network)



รูปที่ 13.32 การเชื่อมต่อเครือข่ายสำนักงานขนาดกลาง

สำนักงานขนาดกลางอาจจะมีพื้นที่สำนักงานได้หลายชั้น มีจำนวนเครื่องลูกข่ายโดยประมาณ 30-100 เครื่อง ภายในสำนักงานอาจจะแบ่งเครือข่ายออกเป็นส่วนๆ ตามภาระหน้าที่ แต่เนื่องจากพื้นที่ใช้งานยังไม่มากจึงทำให้หน่วยงานต่างๆ ยังคงอยู่ในพื้นที่เดียวกันอยู่ เช่น แผนกบัญชี กับการเงินอาจจะใช้พื้นที่ร่วมกัน เป็นต้น

โครงสร้างพื้นฐานด้านฮาร์ดแวร์เครือข่าย

- เราเตอร์หรือสวิตช์ที่มีความสามารถทำงานในระดับเลเยอร์ที่ 3 (เรียกย่อว่า สวิตช์ L3) ของ OSI โมเดล และต้องมีคุณสมบัติของการทำแลนเสมือนได้ (Virtual Local Area Network : VLAN [49)
- สวิตช์ที่ทำงานอยู่ในระดับเลเยอร์ที่ 2 ของ OSI โมเดล (เรียกย่อว่า สวิตช์ L2) จำนวน 16, 24 พอร์ต ชั้นละประมาณ 1-3 เครื่อง
- สายนำสัญญาณชนิด CAT5, CAT5E, CAT6 พร้อมเข้าหัวสายด้วย RJ-45 ใช้สำหรับเชื่อมต่อเครือข่าย LAN จำนวนเท่ากับเครื่องลูกข่ายในแต่ละชั้นและควรมีสำรองไว้อย่างน้อย 1-2 เส้น

- สายนำสัญญาณชนิด CAT5E หรือ CAT6 พร้อมเข้าหัวสาย ใช้สำหรับเชื่อมต่อเครือข่ายระหว่างชั้น และควรมีสายสัญญาณสำรองไว้อย่างน้อย 1 เส้น
- อุปกรณ์ไร้สาย (เมื่อภายในสำนักงานมีความต้องการใช้งาน)
- Link สำหรับเชื่อมต่อเครือข่ายอินเทอร์เน็ต แบนด์วิดท์ขึ้นอยู่กับความต้องการของผู้ใช้ ถ้าใช้เครือข่ายแบบ ADSL อาจจะใช้สายนำสัญญาณเกิน 1 เส้นก็ได้ หรืออาจเลือกใช้การเชื่อมต่อแบบจุดต่อจุดแบบ lease line [51] เพราะให้ความเร็วที่ดีกว่า ADSL มาก
- เครื่องคอมพิวเตอร์เซิร์ฟเวอร์สำหรับให้บริการเช่น ฐานข้อมูล ไฟล์เซิร์ฟเวอร์ หรือเว็บเซิร์ฟเวอร์เป็นต้น
- ไฟล์วอลล์ (option) เพื่อใช้รักษาความปลอดภัยของระบบเครือข่าย ถ้าไม่มีงบประมาณในการจัดหามาใช้งาน ก็สามารถใช้เราเตอร์หรือสวิตช์ L3 ทำ Access Control List: ACL [52] แทนก่อนก็ได้

ออกแบบและคำนวณ IP Address ตามปริมาณเครื่องลูกข่ายในแต่ละชั้น ในเบื้องต้นอาจจะกำหนดจำนวน IP Address [53] ในแต่ละชั้นประมาณ 256 เครื่อง และให้พิจารณาการทำแลนเสมือนด้วย เพราะจะช่วยเพิ่มประสิทธิภาพการทำงานของเครือข่ายได้มาก

ข้อดี

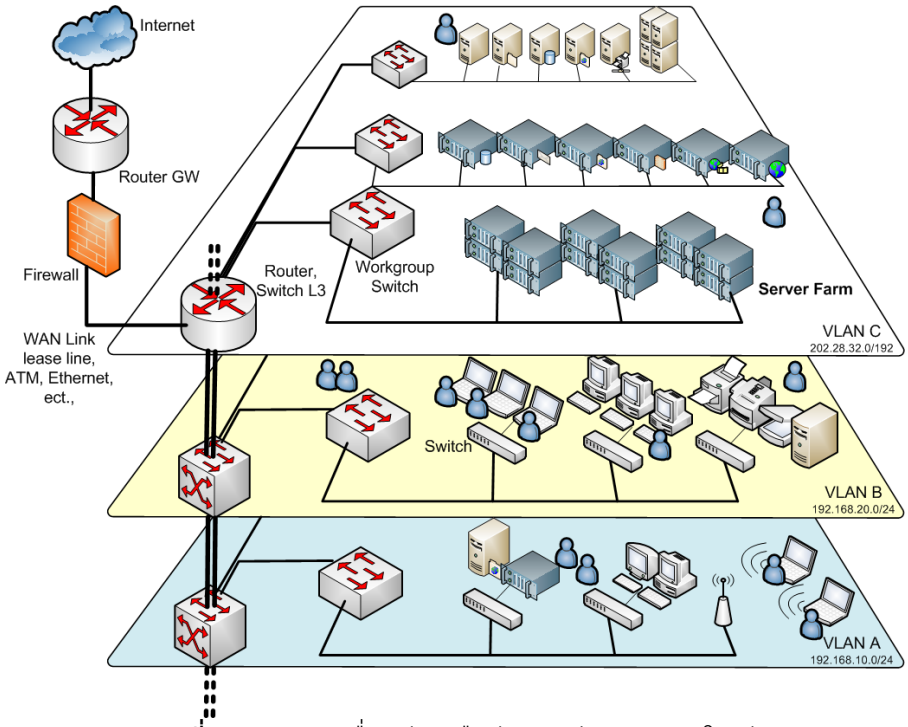
- เครือข่ายเริ่มแบ่งออกเป็นส่วนๆ ตามหน้าที่การทำงาน ทำให้การบริหารจัดการระบบทำได้ง่าย
- กิจกรรมต่างๆ ในสำนักงานสามารถทำได้สะดวกและรวดเร็ว เนื่องจากเครือข่ายไปถึงผู้ใช้งานทุกจุด
- เพิ่มความสามารถในการสื่อสาร แต่เดิมการติดต่อระหว่างแผนกที่อยู่ต่างชั้นจะต้องเดินทางไปติดต่อเอง แต่เมื่อมีระบบเครือข่ายทำให้สามารถติดต่อผ่านโปรแกรมประยุกต์แทน
- สามารถแชร์ข้อมูลร่วมกันได้ เช่น ฐานข้อมูล ไฟล์ข้อมูล เป็นต้น
- ใช้ทรัพยากรร่วมกันทำให้ประหยัดค่าใช้จ่าย เช่น เครื่องพิมพ์ เครื่องสแกนเนอร์ เป็นต้น
- ช่วยเพิ่มสักยภาพในการแข่งขันเนื่องจากมีเทคโนโลยีเครือข่ายรองรับ
- ลดปริมาณการใช้ทรัพยากรในหน่วยงาน เช่น กระดาษ เครื่องเขียน อุปกรณ์สำนักงาน
- ความเร็วในการใช้งานทั้งภายในองค์กรและอินเทอร์เน็ตค่อนข้างสูง

ข้อเสีย

- สิ้นเปลืองพลังงาน เช่นพลังงานไฟฟ้า เครื่องปรับอากาศ เป็นต้น
- อุปกรณ์มีอายุการใช้งาน จึงต้องวางแผนในเรื่องการดูรักษาหรือซ่อมแซม
- อุปกรณ์บางตัวเช่น เราเตอร์และสวิตช์ L3 มีราคาค่อข้างแพง

- เทคโนโลยีด้านคอมพิวเตอร์เปลี่ยนแปลงเร็ว จึงต้องวางแผนการใช้เทคโนโลยีสารสนเทศให้ลัดกุม
- เสียงบประมาณในการเช่าสายสัญญาณอินเทอร์เน็ต
- มีความเสี่ยงในเรื่องความปลอดภัยของข้อมูล
- ต้องจัดหา Staff ดูแลระบบเครือข่าย

โมเดล D: เครือข่ายสำหรับองค์กรขนาดใหญ่ (Enterprise Network)



รูปที่ 13.33 การเชื่อมต่อเครือข่ายสำนักงานขนาดใหญ่

สำหรับการเชื่อมต่อในองค์ขนาดใหญ่จะมีจะนวนเครื่องลูกข่าย ตั้งแต่ 100 เครื่องขึ้นไป บางครั้งอาจจะใช้พื้นที่ทั้งตึก หรือบางส่วนของตึก โดยส่วนมากจะแบ่งเป็นแผนกอย่างชัดเจน เช่น แผนกบัญชี การเงิน IT เป็นต้น ภายในองค์กรจะมีกิจกรรมมาก ต้องมีการติดต่อสื่อสารกันเกือบตลอดเวลา อุปกรณ์มีความหลากหลาย พนักงานทำงานอยู่เป็นจำนวนมาก มีระบบเทคโนโลยีสารสนเทศครบถ้วน

โครงสร้างพื้นฐานด้านฮาร์ดแวร์เครือข่าย

- เราเตอร์ทำหน้าที่เชื่อมโครงข่ายในองค์กรกับเครือข่ายอินเทอร์เน็ต (Router Gateway)
- ไฟล์วอลล์สำหรับจัดการความปลอดภัยของข้อมูลที่ผ่านเข้าออกภายในองค์กร และทำ NAT (Network Adress Translation [52] ได้
- เราเตอร์หรือสวิตช์ L3 (Core Layer [54]) เป็นโครงข่ายหลัก (Backbone) ทำหน้าที่ควบคุมการทำงานของเครือข่ายรอง (Distribute Layer) ลงไปอีกชั้นหนึ่ง ที่ทำงานอยู่ภายในแต่ละ

ชั้น ต้องมีคุณสมบัติของการทำแลนเสมือนได้ อุปกรณ์ที่เป็นแบ็คโบนต้องมีความเร็วในการส่งข้อมูลสูง คงทน สามารถให้บริการได้ตลอดเวลา สายนำสัญญาณที่เชื่อมไปยังแต่ละชั้นควรจะมีคุณภาพดีและมีแบนด์วิดท์สูง เช่น สายใยแก้นำแสง หรือสาย CAT6 เป็นต้น และควรจะมีสายนำสัญญาณสำรองในกรณีที่สายเส้นแรกไม่ทำงาน

- สวิตช์ L3 (Distribute Layer) เป็นโครงข่ายรองที่ทำหน้าที่ควบคุมการทำงานในแต่ละแผนกหรือแต่ละชั้น ต้องมีคุณสมบัติแลนเสมือน, ACL และ QoS เป็นต้น สามารถส่งข้อมูลได้รวดเร็ว จำนวนชั้นละ 1 เครื่อง
- สวิตช์ L2 (Access Layer) จำนวน 16, 24 หรือ 48 พอร์ต ปริมาณเท่ากับหรือมากกว่าจำนวนเครื่องลูกข่ายในแต่ละแผนกหรือแต่ละชั้น
- ห้องปฏิบัติการเครือข่ายหรือ Server Room หรือ Server Farm ประกอบไปด้วยเครื่องคอมพิวเตอร์แม่ข่ายที่ทำหน้าที่ให้บริการ เช่น เว็บเซิร์ฟเวอร์ อีเมลล์ พร็อกซี่ เป็นต้น ภายในห้องต้องมีอุณหภูมิที่เหมาะสำหรับคอมพิวเตอร์ มีเครื่องจ่ายกระแสไฟฟ้าสำรอง มีทีม staff คอยดูแล และที่สำคัญต้องมีระบบรักษาความปลอดภัยทั้งทางด้านกายภาพ (มีการกำหนดสิทธิ์การเข้าออก) และซอฟต์แวร์ (มีไฟล์วอลล์หรือ ACL เพื่อป้องกันการบุกรุกเข้ามายัง Server Farm)
- สายนำสัญญาณชนิด CAT5E, CAT6 ใช้สำหรับเชื่อมต่อเครือข่าย LAN
- สายนำสัญญาณชนิด CAT6 หรือ ใยแก้วนำแสง ใช้สำหรับเชื่อมต่อระหว่างชั้นภายในอาคาร และควรมี Backup ลิงค์ไว้ด้วย
- อุปกรณ์ไร้สาย จำนวนจุดขึ้นอยู่กับรัศมีทำการของ wireless
- Link สำหรับเชื่อมต่อเครือข่ายอินเทอร์เน็ต lease line ความเร็วขึ้นอยู่กับความต้องการและงบประมาณ
- เครื่องคอมพิวเตอร์เซิร์ฟเวอร์สำหรับให้บริการเช่น ฐานข้อมูล ไฟล์เซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ โดเมนเนมเซิร์ฟเวอร์ เมล์เซิร์ฟเวอร์ พร็อกซี่ ระบบยืนยันตัวตน เป็นต้น

การออกแบบจำเป็นต้องมีการแบ่ง VLAN ตามแผนกหรือตามชั้น เพื่อให้ง่ายต่อการจัดการและควบคุม มีระบบการแจกไอพีแบบอัตโนมัติ (DHCP) การเข้าถึงข้อมูลสามารถทำได้ตลอดเวลา สามารถเชื่อมต่อจากที่บ้านเข้ามาใช้ทรัพยากรเครือข่ายในองค์กรได้ (ใช้เทคโนโลยี VPN)

ข้อดี

- ข้อดีหลักๆ เหมือนกับโครงข่ายแบบ C แต่มีคุณสมบัติในเรื่องความมีเสถียรภาพมากขึ้น มีการทำ Backup และ Restore ระบบ
- มีระบบสำรองไฟฟ้าในกรณีฉุกเฉิน

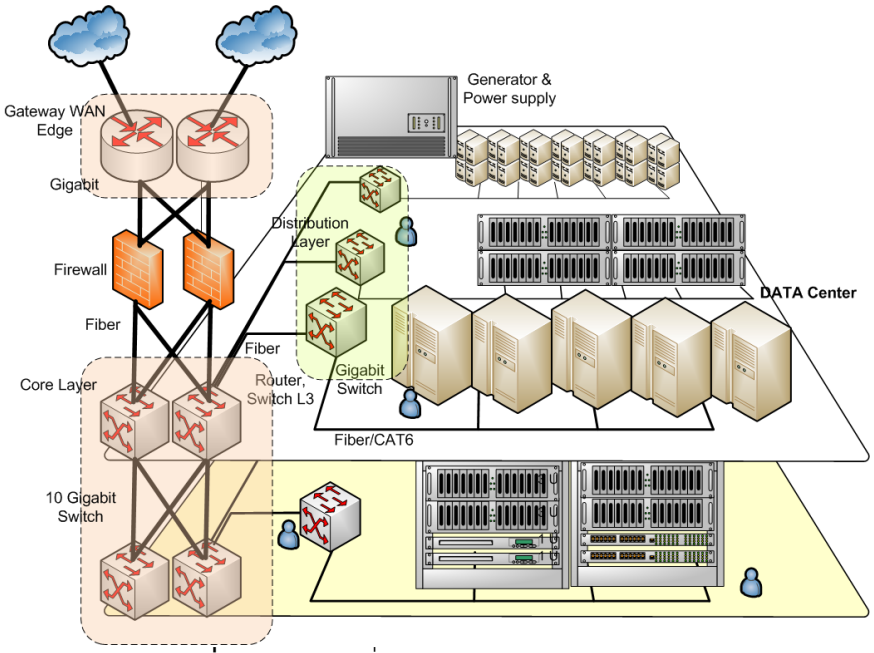
- สามารถให้บริการได้ตลอด 24 ชั่วโมง เป็นลักษณะสำนักงานเปิด คือไม่จำเป็นต้องทำงานอยู่เฉพาะภายในสำนักงานก็ได้
- มี Staff คอยดูและระบบเครือข่าย 24 ชั่วโมง
- โครงข่ายมีการสำรองเส้นทางไว้ในกรณีเส้นทางหลักไม่สามารถทำงานได้

ข้อเสีย

- ข้อเสียหลักๆ คล้ายโมเดล C
- ต้องจ้าง Staff เพิ่มเติมในการดูแลระบบ
- ต้องมีระบบรักษาความปลอดภัยที่เข้มแข็ง เช่น มีไฟล์วอลล์ Intrusion Detection System (IDS [55]) หรือ Intrustion Dectection and Prevention (IDP [56]) เข้ามาช่วยสนับสนุนการทำงาน
- เสียงบประมาณในการ Maintanance สูง เนื่องจากอุปกรณ์ราคาสูง
- ต้องมีการอบรมการใช้เทคโนโลยีสารสนเทศ และต้องมีกฎในการบริหารจัดการไอที
- ไม่ยืดหยุ่นในการปรับแต่งระบบเครือข่าย

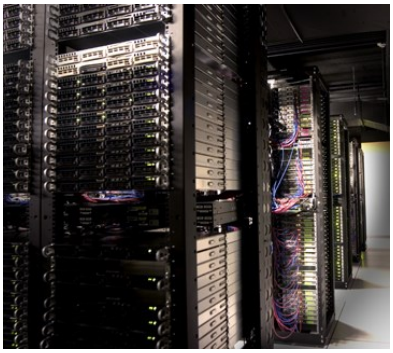
จะเห็นได้ว่าองค์กรขนาดใหญ่ที่มีไอทีใช้งานอยู่จำเป็นต้องคำนึงถึงจุดคุ้มทุนในการใช้ระบบไอทีด้วย จากคำเสนอแนะข้างต้นอาจจะไม่ครอบคลุมทั้งหมด เนื่องจากยังมีรายละเอียดปลีกย่อยมาก ถ้าส่วนใดยังขาดตกปรกพร่องผู้เขียนต้องขออภัยไว้ด้วย

โมเดล E: เครือข่ายสำหรับศูนย์บริการข้อมูล (Data Center Network)



รูปที่ 13.34 การเชื่อมต่อเครือข่าย Data Center

(a)



(b)

รูปที่ 13.35 ห้องเครือข่ายและ Storage ในศูนย์ Data Center
อ้างอิงจาก: ttp://scienceblogs.com/goodmath/upload/2009/05/cloud_computing/data-center-t01.jpg, http://farm1.static.flickr.com/61/395678424_66404fde9d.jpg

Data Center คือ ศูนย์ข้อมูลที่ประกอบด้วยระบบคอมพิวเตอร์และอุปกรณ์ที่เกี่ยวข้อง โดยทั่วไปศูนย์ข้อมูลจะควบคุมสภาพแวดล้อมภายในเป็นอย่างดี เช่น การไหลเวียนอากาศ การเกิดอัคคีภัย การสำรองไฟฟ้า การเชื่อมต่ออินเตอร์เน็ต และการรักษาความปลอดภัย วัตถุประสงค์หรือหน้าที่หลักของ Data Center คือ จัดการ รักษา ประมวลผล และให้บริการข้อมูลที่สำคัญต่อการดำเนินการ ตัวอย่างเช่น ธนาคารจะมี Data Center เพื่อข้อมูลลูกค้าและรายการธุรกรรม (Transaction) ให้บริการรับฝาก Server (Co-location services) เป็นต้น เนื่องด้วยข้อมูลใน Data Center มีความสำคัญต่อการดำเนินการขององค์กร ดังนั้นในการออกแบบและสร้างจะต้องมีมาตราฐาน ศูนย์ ข้อมูล อาจจะมีขนาด 1 ห้อง, 1 ชั้น หรือทั้งอาคาร อุปกรณ์ภายในจะบรรจุอยู่ในตู้ Rack ขนาดมาตรฐาน ที่สะดวกต่อผู้ดูแลระบบโดยสามารถเข้าถึงอุปกรณ์ที่ติดตั้งในตู้ดังกล่าวได้ จากทั้งข้างหน้าและข้างหลัง ระบบต่างๆ ในศูนย์ข้อมูลมีดังต่อไปนี้

- มีระบบควบคุมอากาศ (Air Conditioning) เพื่อให้อุณหภูมิเย็นเหมาะแก่การทำงานของอุปกรณ์คอมพิวเตอร์ โดยมักปรับอุณหภูมิไว้ที่ 20 – 22 องศาเซลเซียส
- ระบบสำรองไฟฟ้า ซึ่งอาจเป็นเครื่องสำรองไฟ (UPS: Uninterrupted power supply) หรือเครื่องกำเนิดไฟฟ้า (Power Generator)
- Redundancy เพื่อป้องกันการผิดพลาดของอุปกรณ์ อุปกรณ์ทุกตัวจะมีอุปกรณ์สำรองที่ทำงานไปพร้อมๆ กัน หรือพร้อมทำงานทดแทนทันที ที่อุปกรณ์หลักหยุดทำงาน
- พื้น ยก (Raised Floor) เพื่อหมุนเวียนอากาศและวางสายไฟ ส่วนสายสัญญาณบางครั้งจะเดินตามรางนำสาย (Wire Way) ซึ่งพื้นยกสามารถเปิดแผ่นพื้นได้เพื่อสะดวกต่อการบำรุงรักษา (Maintenance)
- ระบบรักษาความปลอดภัยทางกายภาพ เช่น Access control (สแกนนิ้ว หรือ Access Card) กล้องวงจรปิด เป็นต้น

- มีระบบความปลอดภัยของเครือข่าย เช่น Fire walls, VPN, gateways, Intrusion detection systems
- ระบบ Application เพื่อดำเนินการ จัดการข้อมูลในศูนย์ข้อมูล เช่น ERM, CRM, Data warehouse, security เป็นต้น

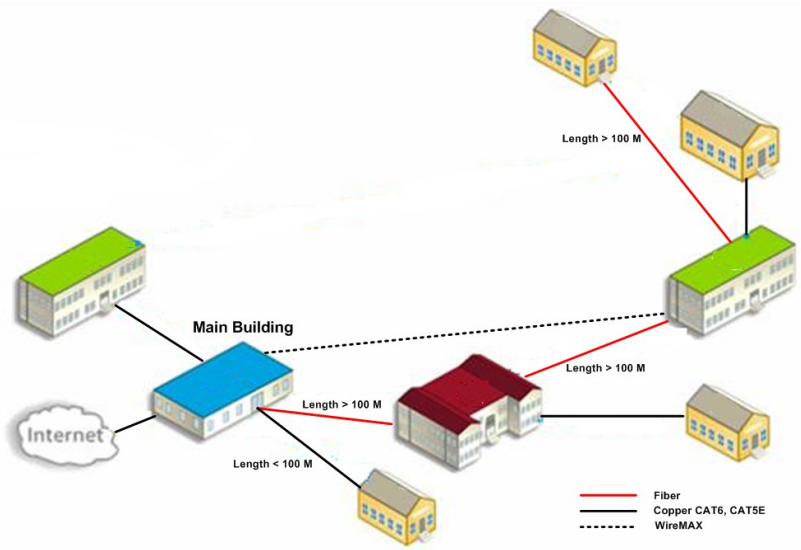
ข้อดี

- ความน่าเชื่อถือของระบบ
- การเชื่อมต่อที่มีประสิทธิภาพสูง เพราะหากมีการวางระบบ Data Center เอาไว้ในต่างประเทศ การขยายโครงข่ายและการแลกเปลี่ยนข้อมูลกับเครือข่ายต่างประเทศจะทำได้ง่าย และสะดวก
- มีการวางระบบ Internet Data Center เอาไว้หลายที่ด้วยกัน เพื่อให้บริการแก่ลูกค้าที่ต้องการเช่าพื้นที่วางโฮสต์หรือเซิร์ฟเวอร์ของตนเอง รวมทั้งยังรองรับลูกค้าที่ไม่ต้องการมีเซิร์ฟเวอร์เป็นของตนเอง แต่ต้องการใช้ทรัพยากรจาก Data Center ในการจัดเก็บข้อมูล ไม่ว่าจะเป็นการฝากเว็บไซต์ การทำอีคอมเมิร์ซต่างๆ
- ระบบความปลอดภัยที่ผู้ใช้มั่นใจได้ ไม่วาจะเป็นไฟร์วอลล์ และการดูแลระบบด้วยเทคโนโลยีที่ทันสมัย

ข้อเสีย

- ลงทุนสูงมาก
- Staff ต้องดูแลเน็ตเวิร์คลอดเวลา และจำนวนต้องเพียงพอ
- การ Maintenance ต้องใช้งบประมาณสูง

โมเดล F: เครือข่ายโรงเรียนและมหาวิทยาลัย (School & University Network)



รูปที่ 13.36 ห้องเครือข่ายโรงเรียน

การเชื่อมต่อประเภทสุดท้ายคือการเชื่อต่อเครือข่ายแบบมีหลายจุดหรือมี campus สำหรับฮาร์ดแวร์ในแต่ละจุดหรือแต่ละตึกจะเหมือนกับเครือข่ายที่กล่าวมาแล้ว โดยดูจากจำนวนของเครื่องลูกข่ายว่าสามารถเข้าได้กับโมเดลใด แต่ที่สำคัญที่สุดคืองบประมาณในการวางเครือข่าย เช่น เครื่องลูกข่ายมีจำนวนมากเกิน 100 เครื่อง แต่ไม่มีงบประมาณในการเชื่อมต่อ ดังนั้นโครงสร้างภายตึกในอาจจะไม่ใช่แบบ Enterprise ตามที่ได้กล่าวมาแล้ว อาจจะมีอุปกรณ์สวิตช์ L2 แบบ 48 พอร์ต จำนวน 1-2 ตัวและเครื่องที่เหลือก็ไม่ได้มีการเชื่อมต่อเข้าสู่ระบบเครือข่าย เป็นต้น ในมุมมองของการเชื่อมต่อที่มีหลายจุดหรือหลายตึกจะพิจารณาเส้นทางของสายนำสัญญาณว่าเป็นชนิดไหน และจะวางสายเคเบิลอย่างไร จึงจะครอบคลุม ได้ความเร็วสูง และประหยัดงบประมาณ การเดินสายนำสัญญาณภายในและนอกอาคาร ในทางปฏิบัติมักจะไม่ได้เดินบ่อย อาจจะเดินเพียงครั้งเดียวแล้วใช้ตลอดอายุการใช้งาน หรือเปลี่ยนเทคโนโลยีใหม่จึงจะมีการรื้อและเดินสายใหม่ หรือสัญญาณนำสัญญาณขาดเนื่องจากสาเหตุทางธรรมชาติบ้าง มนุษย์ทำบ้าง เป็นต้น จากรูปที่ 13.36 แสดงให้เห็นว่าเส้นทางหลักหรือเส้นทางที่มีระยะทางเกิน 100 เมตรควรพิจารณาใช้เป็นสายใยแก้นำแสงแทน หรือ CAT6 ก็จะพอใช้งานได้ แต่ระยะของสายไม่ควรเกิน 120 เมตร ส่วนระยะทางต่ำกว่า 100 เมตร นั้นควรจะใช้ CAT6 หรือ CAT5E ก็ได้ สำหรับเส้นทางไหนที่ไม่สามารถเชื่อมเครือข่ายด้วยสายได้ เช่นมีแม่น้ำขวาง ชุมชนหนาแน่น หรือสถานที่ที่เสี่ยงต่อการเสียหายควรหลีกเลี่ยงโดยการใช้เทคโนโลยีแบบไร้สายแทน เช่น Wire MAX หรือดาวเทียมเป็นต้น

13.4 การติดตั้งโอเพนซอร์สเราท์เตอร์

บทนี้จะกล่าวถึงขั้นตอนการติดตั้งโอเพนซอร์สเราท์เตอร์บนสถาปัตยกรรม x86 (เครื่องคอมพวเตอร์ที่ใช้งานอยู่ทั่วๆ ไป หรือเครื่องคอมพิวเตอร์ที่ใช้ซีพียูจากค่าย Intel, AMD, Cyrix หรือ Via เป็นต้น) สำหรับโอเพนซอร์สเราท์เตอร์ที่จะใช้ติดตั้งและคอนฟิกตลอดทั้งเล่มจะเลือกใช้จากค่าย XORP [18] และ Vyatta [21] เพราะมีโพรโทคอลให้เลือกใช้งานได้หลากหลายและสามารถนำไปพัฒนาต่อยอดเป็นเราท์เตอร์สำหรับใช้ในหน่วยงานของตน หรือในสถานศึกษาที่ต้องการลดต้นทุนการซื้ออุปกรณ์เครือข่ายในราคาที่แพง หรือใช้สำหรับประกอบการทำแลบในวิชาที่เกี่ยวข้องกับคอมพิวเตอร์เน็ตเวิร์ค หรือจะใช้ในการทำวิจัยเรื่องของเราท์เตอร์และโพรโทคอลที่ใช้ทำ Routing หรือต้องการสร้างเครือข่ายเพื่อเป็นกรณีศึกษาเปรียบเทียบการทำงานระหว่างเราท์เตอร์ที่ออกแบบมาโดยเฉพาะกับเราท์เตอร์ที่เป็นโอเพนซอร์ส หรือถ้ามีศักยภาพเพียงพอก็อาจจะตั้งบริษัทในการพัฒนาเราท์เตอร์ให้สามารถใช้งานได้อย่างเต็มรูปแบบอย่างจริงๆ จังๆ ก็เป็นแนวทางที่น่าสนใจอย่างยิ่ง

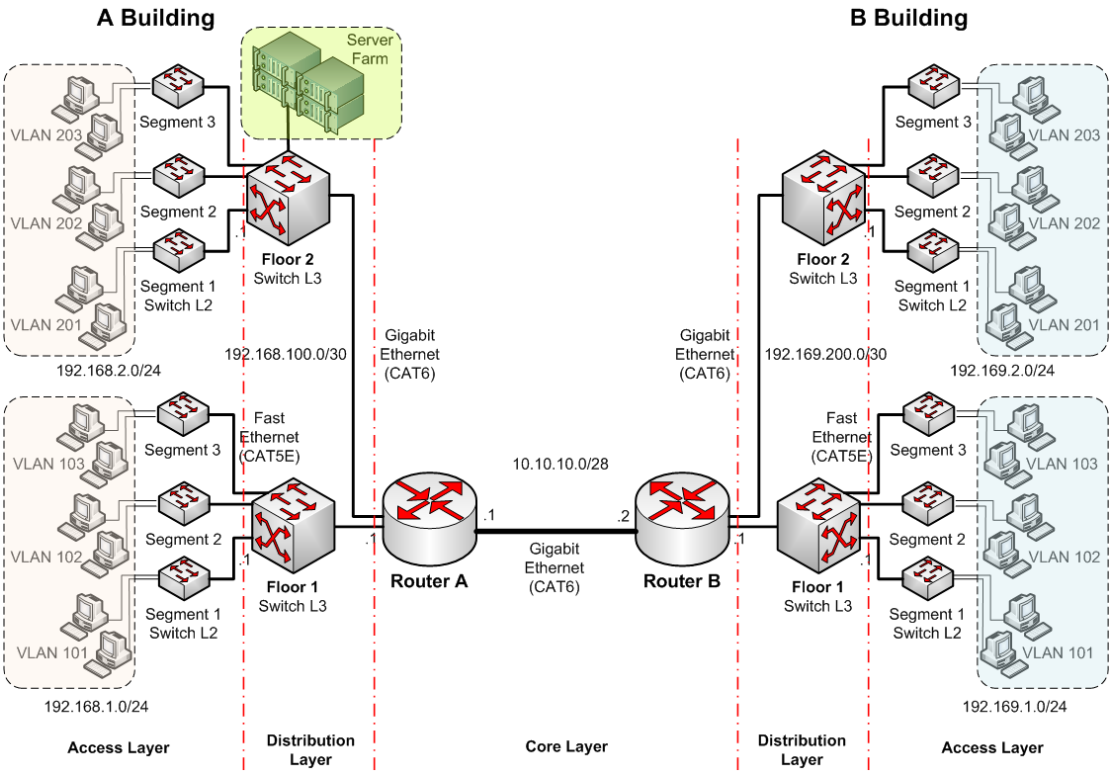
คำแนะนำก่อนการติดตั้งเราท์เตอร์

สิ่งแรกก่อนที่จะติดตั้งเราท์เตอร์คือต้องมีการวางแผนหรือวางโครงข่ายของเน็ตเวิร์คไว้ก่อนว่าเครือข่ายที่จะสร้างขึ้นมีหน้าตาเป็นอย่างไร เพราะเป็นงานที่สอดคล้องกับการจัดเตรียมอุปกรณ์ที่

ต้องใช้ เช่น ต้องใช้ความเร็วของซีพียูเท่าใดจึงจะเหมาะสม ใช้หน่วยความจำขนาดเท่าใดจึงจะรองรับ Routing ที่จะเกิดขึ้น หรือจะเลือกการ์ดอินเทอร์เฟสรุ่นไหน ยี่ห้อใด เป็นต้น ดังนั้นในหัวข้อนี้จะขอแนะนำการเตรียมตัวก่อนที่จะติดตั้งเราท์เตอร์ก่อน

ขั้นตอนที่ 1 ออกแบบผังเครือข่าย

ในขั้นตอนนี้ให้ผู้ออกแบบใช้ซอฟต์แวร์ออกแบบเครือข่าย เช่น Visio, SmartDraw, Concept Draw, Dig หรือกระดาษวาดภาพของเครือข่ายตามความต้องการให้เหมาะสมกับหน่วยงานของตนเองก่อน ผังเครือข่ายมีความสำคัญอย่างมากเปรียบเสมือนแผนที่ในการเดินทางเลยทีเดียว ถ้าเขียนผังผิด จะทำให้เครือข่ายที่สร้างขึ้นมีโอกาสสูงมากที่จะไม่สามารถทำงานได้ ประโยชน์ของผังเครือข่ายที่สำคัญอีกประการหนึ่งคือ ใช้สำหรับวางแผนและดูแลระบบเครือข่ายในอนาคต เช่น ถ้าต้องการเพิ่มอุปกรณ์เข้าไปในเครือข่ายควรจะเพิ่มตรงไหน เมื่อดูผังเครือข่ายก็จะรู้ได้ทันทีไม่ต้องเสียเวลาไปสำรวจดูเครือข่ายจริง และยังช่วยให้หน่วยงานที่มีผู้ดูแลระบบหลายๆ คนเข้าใจระบบเครือข่ายได้ตรงกัน เวลาเกิดปัญหาจะได้แก้ปัญหาได้ตรงจุดและทันเวลา ตัวอย่างการเขียนผังเครือข่ายดังรูปที่ 13.37



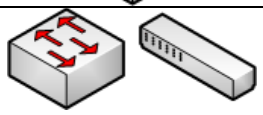
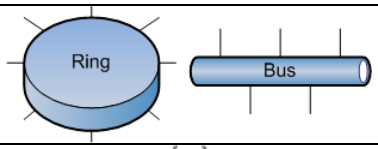
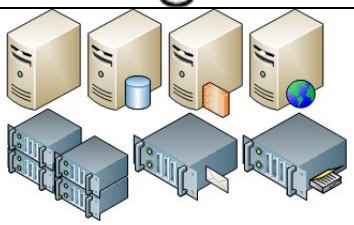
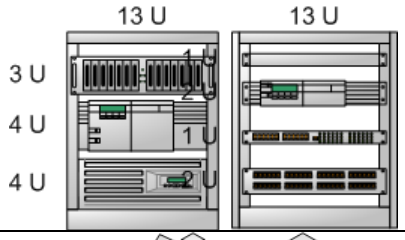
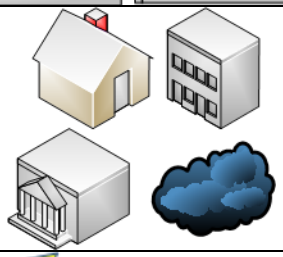
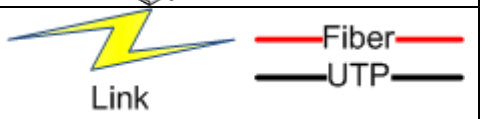
รูปที่ 13.37 การเขียนผังเครือข่ายด้วยโปรแกรม Visio ของ Microsoft

สำหรับหลักการเขียนแบบผังระบบเครือข่ายให้พิจารณารายละเอียดดังต่อไปนี้

- ภาพรวมของผังเครือข่ายต้องสามารถแสดงถึงรูปแบบเครือข่ายได้อย่างชัดเจนและตรงตามความเป็นจริง
- สามารถสื่อสารให้ผู้อ่านสามารถเข้าใจได้โดยง่าย

- เลือกใช้สัญลักษณ์ให้ตรงกับอุปกรณ์เครือข่ายแต่ละชนิดอย่างเหมาะสม ควรปรับขนาดของสัญลักษณ์ให้เหมาะสม

ตารางที่ 13.8 สัญลักษณ์ของอุปกรณ์เครือข่าย

ชื่อ	หน้าที่	สัญลักษณ์
เราท์เตอร์	จัดเส้นทางและเชื่อมโยง Topology ที่ต่างกัน	
สวิตช์เลเยอร์ 3 (L3)	จัดเส้นทางและทำ VLAN	
สวิตช์เลเยอร์ 2 (L2)	รับส่งข้อมูลจากเครื่องลูกข่ายมายัง Backbone	
สวิตช์และฮับ	รับส่งข้อมูลจากเครื่องลูกข่ายปลายทาง	
Topology	เน็ตเวิร์คโทโพโลยี	
เครือข่ายไร้สาย	เครือข่ายไร้สาย	
เซอร์ฟเวอร์	ให้บริการต่างๆ เช่น เมล์เซิร์ฟเวอร์ DNS FTP Proxy DHCP Database เป็นต้น	
Rack, Cabinet	จัดเก็บอุปกรณ์ให้เป็นหมวดหมู่ และเป็นระเบียบ	
Locations	ตำแหน่งที่ตั้งของเครือข่าย	
Cable	สายนำสัญญาณ	

NOTE: สัญลักษณ์ของอุปกรณ์เครือข่ายที่แสดงในตารางที่ 13.8 ยังมีอีกมากมายในที่นี้ขอยกตัวอย่างสัญลักษณ์ที่ใช้บ่อยๆ เท่านั้น

- จุดใดในเครือข่ายที่สำคัญควรจะมีการเน้นด้วยเส้นประหรือใช้สีประกอบ
- การวางผังเครือข่ายควรเริ่มจากส่วนหลักของเครือข่าย (Core Layer) แล้วค่อยๆ ขยายออกยังส่วนปลายคือ Distribution Layer และส่วนสุดท้ายเป็น Access Layer
- จัดกลุ่มของอุปกรณ์ตามลักษณะทางกายภาพและสอดคล้องกับเครือข่ายจริง

สรุปคือให้เขียนผังเครือข่ายให้สามารถทำความเข้าใจได้ง่าย ผังเครือข่ายที่ดี เมื่อผู้อ่านผังเครือข่ายในครั้งแรกจะต้องทำความเข้าใจระบบเครือข่ายได้ทั้งหมด การออกแบบผังเครือข่ายมีรายละเอียดที่จะต้องพิจารณาเยอะพอสมควร จึงไม่สามารถอธิบายได้อย่างละเอียดในที่นี้ได้

ขั้นตอนที่ 2 พิจารณาและประเมินองค์ประกอบของเครือข่าย

เมื่อออกแบบผังเครือข่ายเสร็จเรียบร้อยแล้วขั้นตอนต่อไปคือการพิจารณ์และประเมินเครือข่ายที่เขียนขึ้นว่ามีความเป็นไปได้หรือไม่โดยเทียบเคียงกับเทคโนโลยีที่มีอยู่ในปัจจุบัน (สามารถอ่านเพิ่มเติมได้เรียนรู้เครือข่ายและอุปกรณ์ Cisco ด้วยโปรแกรม Simulation เรื่องการออกแบบระบบเครือข่าย) เช่นเมื่อต้องการเพิ่มขนาดแบนด์วิดท์ในเครือข่ายจากเดิม 100 Mbps เป็น 1 Gbps จะสามารถทำได้หรือไม่ ด้วยโครงข่ายที่มีอยู่เดิม หรือจำเป็นต้องมีการวางโครงข่ายใหม่เป็นต้น ในหนังสือเล่มนี้จะมุ่งประเด็นไปที่การออกแบบเราท์เตอร์ให้สามารถรองรับเครือข่ายได้ถึงระดับกิกกะบิต ดังนั้นในหัวข้อนี้ผู้เขียนจะกล่าวถึงตัวเราท์เตอร์เป็นหลัก จากเครือข่ายดังรูปที่ 13.17 จะเห็นว่าเราท์เตอร์ A จะประกอบไปด้วยอย่างน้อย 3 อินเทอร์เฟสคือ

1. กิกกะบิตอินเทอร์เฟสระหว่างเราท์เตอร์ที่เชื่อมระหว่างตึก A และ B
2. กิกกะบิตอินเทอร์เฟสที่เชื่อมไปยังชั้นที่ 1 ของตึก A
3. กิกกะบิตอินเทอร์เฟสที่เชื่อมไปยังชั้นที่ 2 ของตึก A
4. กิกกะบิตอินเทอร์เฟสที่เชื่อมไปยังชั้นอื่นๆ ของตึก A (ถ้าตึกมีมากกว่า 2 ชั้น)

เมื่อพิจารณาความต้องการพื้นฐานที่เราท์เตอร์ A แล้วจะประกอบไปด้วยรายละเอียดดังต่อไปนี้ดังตารางที่ 13.9

ตารางที่ 1

เราท์เตอร์ A ตึก A	
ความต้องการของระบบ	**เทคโนโลยีที่สนับสนุน**
โพรโทคอลที่ Routing ทำงาน	OSPF โพรโทคอล
จำนวน Routing ที่สามารถรองรับได้	< 50 เส้นทาง (คำนวณง่ายๆ โดยดูจากจำนวนเครือข่ายย่อยๆ ที่เกิดขึ้นทั้งหมด)
ขนาดของแบนด์วิดท์รวม	ไม่เกิน 1 Gbps
จำนวนอินเทอร์เฟสที่ใช้	อย่างน้อย 3 อินเทอร์เฟส ความเร็ว 1 Gbps

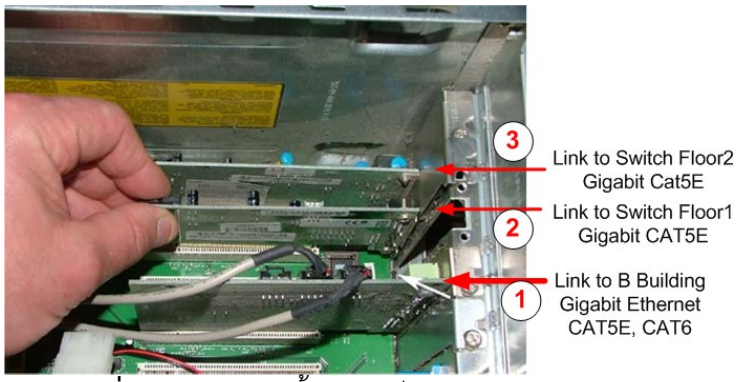
เทคโนโลยีที่ใช้เชื่อมต่อ	อีเทอร์เน็ต
ชนิดของสายนำสัญญาณ	CAT5E, CAT6 (มาตรฐาน ANSI/TIA/EIA-568-B.1)
คอนเน็คเตอร์ที่ใช้เชื่อมต่ออินเทอร์เฟส	RJ45
สรุป	เราท์เตอร์ A จะใช้เครื่องคอมพิวเตอร์ 1 Core ความเร็วของซีพียู 1 GB ขึ้นไป หน่วยความจำอย่างน้อย 1 GB และใช้ฮาร์ดดิสก์อย่างน้อย 2 GB (อ่านเพิ่มเติมได้ในบาที่ 2) เน็ตเวิร์คการ์ดอย่างน้อยเป็น PCI-X หรือถ้าให้ได้ประสิทธิภาพสูงขึ้นให้ใช้ PCI-Express

จากตารางที่ 13.9 เป็นการประเมินประสิทธิภาพของเราท์เตอร์ A ที่ทำหน้าที่เป็นผู้จัดการเส้นทางหลัก (Backbone) เป็นแกนกลางในการสื่อสารข้อมูลระหว่างตึก (Core Layer) เราท์เตอร์เครื่องดังกล่าวจะต้องมีประสิทธิภาพที่สูงพอสมควร เพราะถ้าเครื่องไม่มีประสิทธิภาพแล้วจะส่งผลต่อการสื่อสารทั้งหมดบนเครือข่าย ผลจากการประเมินจะเห็นว่าเราท์เตอร์ตัวดังกล่าวจะมีความต้องการทางด้านฮาร์ดแวร์และลอจิคอล ดังนี้ ทางด้านกายภาพ เราท์เตอร์ A จะต้องเป็นซีพียูที่มีความเร็วอย่างน้อย 1 GB (ถ้าเป็นซีพียู 2 core จะให้ประสิทธิภาพที่ดีกว่า) หน่วยความจำหลักอย่างน้อย 1 GB พื้นที่ใช้จัดเก็บหรือฮาร์ดดิสก์อย่างน้อย 2 GB และการ์ดเครือข่ายอย่างน้อย 3 ใบ ชนิด PCI-X (ความเร็วสูงสุดที่ 1 Gbps) ใช้เทคโนโลยีการเชื่อมต่อแบบอีเทอร์เน็ต ชนิดผสมระหว่าง บัส (ระหว่างตึก A และ B) กับดาว (ภายในแต่ละ segment) สายนำสัญญาณเป็นทองแดงชนิด CAT5E ก็เพียงพอ แต่ถ้าต้องการความเร็วที่เพิ่มขึ้นเป็น 2 เท่าให้ใช้ CAT6 แทน (สัญญาณความถี่ของ CAT5E ประมาณ 100 MHz ส่วน CAT6 เพิ่มไปเป็น 200 MHz ทำให้ส่งข้อมูลเพิ่มได้เป็น 2 เท่าโดยประมาณ) สำหรับความต้องการทางด้านลอจิคอลคือ เราท์เตอร์ A ต้องค้นหาเส้นทางด้วยโพรโทคอล Routing ชนิด OSPF และต้องสามารถรองรับเส้นทางได้ถึง 50 เส้นทาง ส่วนอุปกรณ์ตัวอื่นๆ ในเครือข่ายก็ใช้หลักการพิจารณาเหมือนกัน แต่แตกต่างกันตรงที่ขนาดของแบนด์วิดท์และหน้าที่การทำงานในแต่ละส่วน เช่น ระดับ Access Layer ไม่ต้องคำนึงถึงเรื่องการกำจัด collision, broadcast เป็นต้น มีหน้าที่หลักคือส่งและรับข้อมูลจากเครื่องลูกข่ายเข้ามายังเครือข่ายหลักให้ได้ก็เพียงพอ แต่ในส่วนของอุปกรณ์ที่ทำหน้าที่อยู่ใน Distrition Layer ควรจะต้องกัดการเกี่ยวกับการทำ VLAN การรักษาความปลอดภัย การกำจัดแพ็คเก็ตที่ไม่พึงประสงค์ การจัดเส้นทาง ส่วน Core Layer ทำหน้าที่จัดเส้นทางเชื่อมต่อ WAN และรับส่งข้อมูลให้เร็วที่สุดเท่าที่จะทำได้

ขั้นตอนที่ 3 จัดหาฮาร์ดแวร์ที่สามารถรองรับเครือข่ายที่ออกแบบ

ในขั้นตอนนี้จะต้องจัดเตรียมอุปกรณ์เพื่อรองรับกับเครือข่ายที่ต้องการสร้างขึ้น จากขั้นตอนที่ 2 เมื่อพิจารณาที่เราท์เตอร์ A เราต้องจัดเตรียมอุปกรณ์ในการสร้างเราท์เตอร์คือใช้เครื่อง PC หรือถ้าต้องการให้เราท์เตอร์มีประสิทธิภาพการทำงานที่ดีขึ้น อาจเลือกใช้เครื่องเซิร์ฟเวอร์หรือเครื่องที่มีคุณสมบัติที่ดีกว่ามาใช้แทน ความเร็วของซีพีอยู่ไม่ควรต่ำกว่า 1 GB หน่วยความจำไม่น้อยกว่า 1 GB

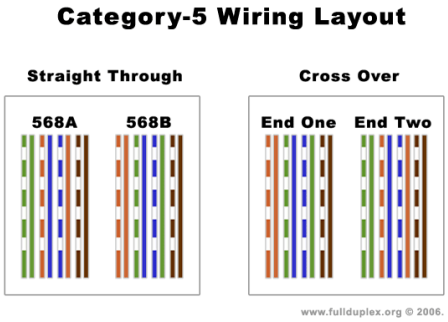
และการ์ดเน็ตเวิร์คสำหรับใช้เชื่อมต่อเครือข่าย อย่างน้อย 3 การ์ด (ความต้องการด้านฮาร์ดแวร์ตามที่กล่าวมานี้ใช้กับเราท์เตอร์ A สำหรับในตัวอย่างนี้เท่านั้น) ดังรูปที่ 13.38 แสดงการติดตั้งการ์ดเน็ตเวิร์คบนเครื่องคอมพิวเตอร์เพื่อรองรับการเชื่อมต่อตามที่ได้ออกแบบไว้



รูปที่ 13.38 การติดตั้งการ์ดเน็ตเวิร์คสำหรับเราท์เตอร์

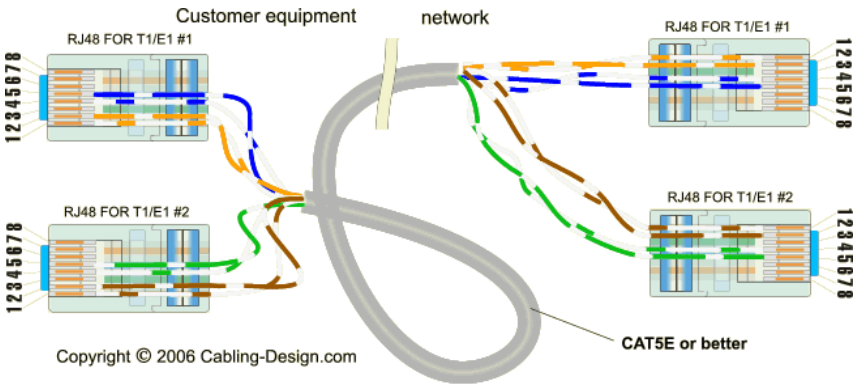
จากรูปจะเห็นได้ว่า การ์ดใบที่ 1 ใช้สำหรับเชื่อมเราท์เตอร์ระหว่างตึก A และ B เข้าด้วยกัน การ์ดใบที่ 2 ใช้เชื่อมต่อเข้ากับสวิตช์ภายในตึก ชั้นที่ 1 และการ์ดใบสุดท้ายใช้สำหรับเชื่อมต่อเข้ากับสวิตช์ภายในตึกชั้นที่ 2

สายนำสัญญาณที่ใช้จะต้องสอดคล้องกับความเร็วที่ได้ออกแบบไว้ ในตัวอย่างนี้เราจะใช้สายสัญญาณประเภท CAT5, CAT5E และ CAT6 ทั้ง 2 แบบคือสายตรง (Straight through) และสายไขว้ (Cross Over) ซึ่งแสดงไว้ในรูปที่ 13.39

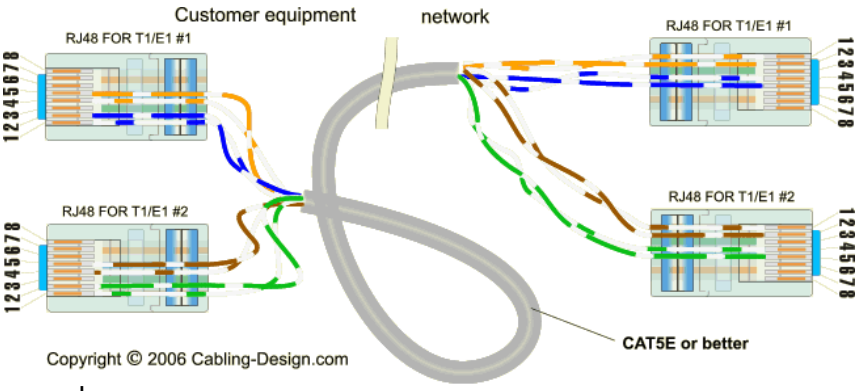


รูปที่ 13.39 การจัดเรียงสาย CAT5, CAT5E, CAT6 ชนิดสายตรงและไขว้

รูปที่ 13.40 คีมเข้าหัวสาย CAT5, CAT5E, CAT6



รูปที่ 13.41 การจัดเรียงสาย T1/E1 ชนิดสายตรง (Straight-through)



รูปที่ 13.42 การจัดเรียงสาย T1/E1 ชนิดสายไขว้ (Cross Over Cable)

สำหรับรายละเอียดการเข้าหัวสายสามารถค้นคว้าเพิ่มเติมได้จาก [63], [64]

ขั้นตอนที่ 4 ติดตั้งเราท์เตอร์

เมื่อฮาร์ดแวร์ทั้งหมดพร้อมเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการติดตั้งเราท์เตอร์ ในหนังสือนี้จะทำการติดตั้งโอเพนซอร์สเราท์เตอร์ 2 ค่าย คือ XORP และ Vyatta ซึ่งมีขั้นตอนการติดตั้งดังต่อไปนี้

การติดตั้งเราท์เตอร์ค่าย XORP

XORP เป็นเราท์เตอร์ที่มี Feature มากมายให้ใช้งาน ดังนั้นผู้เขียนจะใช้เป็นตัวหลักในการสร้างเราท์เตอร์ ซึ่งวิธีการติดตั้งมีขั้นตอนดังนี้

XORP สามารถทำการติดตั้งได้ 3 แบบคือ

1. ติดตั้งบนระบบปฏิบัติการลินุกซ์หรือ BSD ด้วยวิธีการคอมไพล์
2. ติดตั้งบนระบบปฏิบัติการวินโดวส์
3. LiveCD (สามารถใช้งานได้โดยไม่มีการติดตั้ง)

1. ติดตั้งบนระบบปฏิบัติการลินุกซ์หรือ BSD ด้วยวิธีการคอมไพล์

XORP พัฒนาด้วยภาษา C++ สามารถทำงานได้บนระบบปฏิบัติการ BSD ทุกเวอร์ชันและสามารถทำงานได้บนลินุกซ์เคอร์เนล 2.4.x, 2.6.x หรือสูงกว่า XORP สามารถทำงานได้บนระบบปฏิบัติการต่อไปนี้ได้อีกคือ DragonFlyBSD, NetBSD, OpenBSD, MacOS X (10.2 หรือดีกว่า) และวินโดวส์ 2003 ด้วย ถ้าต้องการใช้งานโพรโทคอลมัลติคาสท์ (Multicast) ระบบปฏิบัติการ MacOS และวินโดวส์ ยังไม่สนับสนุน

วิธีการติดตั้ง

1. ดาวน์โหลไฟล์ source code จาก http://www.xorp.org/downloads.html ไฟล์ชื่อ xorp-x.y.tar.gz (x.y คือเวอร์ชัน ปัจจุบันเวอร์ชัน 1.6) เก็บไว้ในไดเร็คทรอรี่ที่ต้องการ เช่น /root/install
2. คลายไฟล์ที่ดาวน์โหลดมาด้วยคำสั่ง tar –zxvf ชื่อไฟล์

   Router1# tar –zxvf xorp-1.6.tar.gz
3. เมื่อคลายไฟล์แล้วจะ**ปรากฏ**ไดเร็คทรอรี่ชื่อว่า xorp-1.6 จากนั้นให้ย้ายเข้าไปในไดเร็คทรอรี่ดังกล่าว ด้วยคำสั่ง cd

   Router1# cd xorp-1.6
4. ความต้องการพื้นฐานก่อนการติดตั้ง XORP มีดังต่อไปนี้
   - gcc คอมไพล์เลอร์ (เวอร์ชันของ gcc ที่สนับสนุนคือ gcc 2.95.x, 2.96, 3.2.x, 3.3.x, 3.4.x, 4.0.x, 4.1.x, 4.2.0, 4.3.0 หรือสูงกว่า) ทดสอบโดยใช้คำสั่ง

     Router1# gcc version หรือ

     Router1# gcc -v
   - gmake (บนลินุกซ์จะติดตั้งมาอยู่แล้วแต่บน BSD อาจจะต้องติดตั้งเพิ่มเติม ในบางเวอร์ชัน สามารถอ่านข้อมูลเพิ่มเติมได้จาก การติดตั้งระบบปฏิบัติการ FreeBSD ภาคผนวก)

     Router1# make

     Router1# gmake //หรือ /usr/local/bin/gmake
   - แสดงเคอร์นอลปัจจุบันที่ทำงานอยู่ว่าเป็นเวอร์ชั่นที่สนับสนุนหรือไม่ ใช้คำสั่ง

     Router1# uname -a

5. ใช้คำสั่ง configure เพื่อบอกให้ระบบปฏิบัติการทำการตรวจสอบที่อยู่ของไฟล์ต่างๆ ที่จำเป็นสำหรับคอมไพล์โปรแกรม ผลลัพธ์จากคำสั่งนี้คือ จะได้ไฟล์ใหม่ชื่อว่า Makefile

   Router1# ./configure --with-snmp //ใช้งานร่วมกับ snmp

   Router1# ./configure //เมื่อไม่ต้องการใช้ snmp

ถ้าต้องการให้เราท์เตอร์สามารถ monitor ได้ (SNMP) ต้องติดตั้ง net-snmp เพิ่มเติมและควรใช้เวอร์ชันที่สูงกว่า 5.0.6 อาจจะมีข้อผิดพลาดในบางเวอร์ชัน ต้องมีการแก้ไขด้วย patch ข้อมูลจาก http://www.xorp.org/snmp.html เมื่อต้องการใช้งาน SNMP ให้ใช้ร่วมกับ configure (ต้องติดตั้ง net-snmp ก่อน) ในบางครั้งอาจจะเกิด error ในลักษณะดังต่อไปนี้ แสดงว่าไม่มี gmake จะต้องติดตั้ง gmake เสียก่อน (สามารถดูได้จากภาคผนวก)

GNU make was not found during configure process and is essential for building

XORP. If you believe this is an error, please let feedback@xorp.org know.

6. ถ้าไม่เกิดข้อผิดพลาดอะไรขึ้นมาให้ทำการ make ต่อไป (กรณีเกิดข้อผิดพลาดต้องตรวจสอบว่าเกิดขึ้นจากที่ไหน เช่น ไม่มี gcc เป็นต้น)

   Router1# gmake //หรือ

   Router1# /usr/local/bin/gmake //หรือ

   ขั้นตอนนี้อาจจะใช้เวลานานหลายนาทีขึ้นอยู่กับสมรรถนะของคอมพิวเตอร์ที่ใช้งาน
7. เมื่อไม่มีข้อผิดพลาดขั้นตอนต่อไปให้ทำการ validate module และต้องใช้โปรแกรมอีก 2 ตัวคือ bash และ python เมื่อติดตั้งโปรแกรมทั้ง 2 แล้วให้ใช้คำสั่ง gmake check

   Router1#gmake check //หรือ

   Router1#/usr/local/bin/gmake check
8. ในการ validate นี้ใช้เวลานานมาก (อาจเป็นชั่วโมง) ในขั้นตอนนี้อาจจะข้ามไปได้ แต่เพื่อควรมั่นใจควรจะทำการ validate จะดีกว่า เมื่อเสร็จแล้วจะมีข้อความแสดงว่าเสร็จแล้ว

   Router1# PASS: test_trie

   แต่ถ้าแสดงข้อความว่า FAIL: test_peering1.sh แสดงว่าการ make ล้มเหลว แต่ถ้ามีข้อความที่มีค่าว่า WARNING เช่น

   [2004/04/22 00:48:57 WARNING coord BGP] TCP connection from test_peer: peer1 to localhost closed

   แสดงว่าสามารถทำงานได้ ไม่รุนแรง แต่ในอนาคตบางโมดูลอาจจะเกิดปัญหาได้
9. ขั้นตอนสุดท้ายคือการติดตั้ง execute table ไฟล์ ที่ผ่านกระบวนการ make มาเรียบร้อยแล้วลงบนเครื่อง โปรแกรม xorp ที่ติดตั้งบน FreeBSD นั้น จะแยกไปวางยังส่วนต่างๆ บนโครงสร้างไฟล์ของ FreeBSD ดังนี้

คอนฟิกกูเรชันไฟล์จะอยู่ที่	/usr/local/etc/ หรือ /usr/local/xorp/etc
โปรแกรมที่ execute	/usr/local/bin หรือ /usr/local/xorp/bin
ไฟล์บันทึกการทำงาน	/var/log

   คำสั่งติดตั้งโปรแกรมคือ make install หรือ gmake install

   Router1# gmake install หรือ

Router1# /usr/local/bin/gmake install

> **NOTE:** การสั่ง make, make install, make check จะต้องอยู่ในไดร์เรอทรอรี่ของโปรแกรมที่ต้องการติดตั้ง หรือให้พูดง่ายๆ คือ อยู่ในโฟร์เดอร์เดียวกับ Makefile นั่นเอง

10. สั่งรันโปรแกรม

XORP มีโปรแกรมสำหรับควบคุมการทำงานของเราท์เตอร์ ชื่อว่า xorp_rtrmgr (short for XORP Router Manager) โดยปกติจะติดตั้งอยู่ในโฟร์เดอร์ชื่อว่า rtrmgr เมื่อ xorp_rtrmgr เริ่มทำงานมันจะค้นหาไฟล์คอนฟิกที่มีชื่อว่า config.boot หรือ xorp.conf (ปกติไฟล์ดังกล่าวจะไม่มีขณะติดตั้ง จำเป็นต้องสร้างขึ้นมาเอง) มาทำงานในอันดับแรก ในการสั่งให้ xorp_rtrmgr ทำงานจะต้องใช้สิทธิ์ของ root เนื่องจากจำเป็นต้องกำหนดเส้นทางให้กับเคอร์เนลทำงานด้วย เมื่อรัน XORP จาก Live CD โปรแกรมทุกอย่างจะทำงานจากแผ่นโดยไม่จำเป็นต้องมีการติดตั้ง สำหรับไฟล์คอนฟิกจำเป็นต้องอ่านจาก floppy drive แทน ในการสั่งงานหรือตอบโต้กับเราท์เตอร์จะกระทำผ่านเชลล์ชื่อว่า xorpsh (xorp_rtrmgr ต้องทำงานก่อน xorpsh จึงจะทำงานได้)

11. การสั่งงานเราท์เตอร์

การสั่งงานหรือคอนฟิกเราท์เตอร์ต้องทำ 2 ขั้นตอนตามลำดับ คือ

1. สั่งให้ตัวควบคุมเราท์เตอร์ทำงานก่อน(xorp_rtrmgr) โดยการใช้คำสั่ง xorp_rtrmgr –b “ชื่อไฟล์คอนฟิก” (โดยปกติจะมีไฟล์คอนฟิกแบบ default มาให้ อยู่ที่ /usr/local/etc/xorp.conf.example แต่ไม่ควรนำไปใช้งานทันทีเพราะจะทำให้เราท์เตอร์ไม่สามารถทำงานได้ ต้องปรับแต่งก่อน ควรใช้ไฟล์คอนฟิกชื่อว่า xorp.conf)

/usr/local/xorp/bin/xorp_rtrmgr –b /usr/local/xorp/etc/xorp.conf

```
[ 2009/09/08 00:09:02  ERROR xorp_rtrmgr:1451 RTRMGR +142 userdb.cc add_user ] G
roup "xorp" does not exist on this system.
[ 2009/09/08 00:09:02  ERROR xorp_rtrmgr:1451 RTRMGR +142 userdb.cc add_user ] G
roup "xorp" does not exist on this system.
[ 2009/09/08 00:09:02  INFO xorp_rtrmgr:1451 RTRMGR +249 master_conf_tree.cc exe
cute ] Changed modules: interfaces, firewall, fea, rib, fib2mrib
[ 2009/09/08 00:09:02  INFO xorp_rtrmgr:1451 RTRMGR +101 module_manager.cc execu
te ] Executing module: interfaces (fea/xorp_fea)
[ 2009/09/08 00:09:04 INFO xorp_fea MFEA ] MFEA enabled
[ 2009/09/08 00:09:04 INFO xorp_fea MFEA ] CLI enabled
[ 2009/09/08 00:09:04 INFO xorp_fea MFEA ] CLI started
[ 2009/09/08 00:09:04 INFO xorp_fea MFEA ] MFEA enabled
[ 2009/09/08 00:09:04 INFO xorp_fea MFEA ] CLI enabled
[ 2009/09/08 00:09:04 INFO xorp_fea MFEA ] CLI started
[ 2009/09/08 00:09:04  INFO xorp_rtrmgr:1451 RTRMGR +101 module_manager.cc execu
te ] Executing module: firewall (fea/xorp_fea)
[ 2009/09/08 00:09:08  INFO xorp_rtrmgr:1451 RTRMGR +101 module_manager.cc execu
te ] Executing module: fea (fea/xorp_fea)
[ 2009/09/08 00:09:12  INFO xorp_rtrmgr:1451 RTRMGR +101 module_manager.cc execu
te ] Executing module: rib (rib/xorp_rib)
[ 2009/09/08 00:09:14  INFO xorp_rtrmgr:1451 RTRMGR +101 module_manager.cc execu
te ] Executing module: fib2mrib (fib2mrib/xorp_fib2mrib)
[ 2009/09/08 00:09:16  INFO xorp_rtrmgr:1451 RTRMGR +2233 task.cc run_task ] No
more tasks to run
```

รูปที่ 13.43 แสดงการทำงานของ xorp_rtrmgr

2. คอนฟิกเราท์เตอร์ผ่าน command line ด้วยเชลล์ xorpsh ซึ่งเมื่อเราท์เตอร์ทำงานแล้ว ให้ใช้คำสั่ง xorpsh เพื่อบอกให้เชลล์ทำงาน ต่อจากนั้นจึงสามารถใช้คำสั่งต่างๆ ของเราท์เตอร์ได้

Router1#/usr/local/bin/xorpsh

```
Router1# /usr/local/xorp/bin/xorpsh
Welcome to XORP on Router1.msu.ac.th
root@Router1.msu.ac.th>
```

ผลลัพธ์จากการเรียก xorpsh เชลล์

ทำการคอนฟิกเราท์เตอร์โดยใช้คำสั่ง configure การเข้าไปปรับแต่งในโหมดนี้สำหรับ Unix จะต้องเป็นสมาชิกในกลุ่มชื่อว่า xorp เท่านั้นจึงจะสามารถทำงานได้ ดังนั้นต้องทำการเพิ่มผู้ใช้งานในระบบให้อยู่ในกลุ่ม xorp ก่อน มีขั้นตอนดังนี้

/usr/sbin/sysinstall → configure → User Management → Group→ Group name: xorp →User→ Login ID: xorp → Group: xorp→ OK → Exit

จากนั้นทำการ log in ใหม่ด้วย user xorp หรือใช้คำสั่ง su xorp เพื่อเปลี่ยนผู้ใช้งานจาก root เป็น xorp

Router1# **su xorp**

$ **/usr/local/xorp/bin/xorpsh**

Welcome to XORP on Router1.msu.ac.th

xorp@Router1.msu.ac.th> **configure**

Entering configuration mode.

There are no other users in configuration mode.

[edit]

xorp@Router1.msu.ac.th# //เราท์เตอร์พร้อมใช้งาน

NOTE: ถ้าต้องการให้ทำงานทุกครั้งเมื่อมีการเปิดเครื่อง อ่านข้อมูลเพิ่มเติมในภาคผนวก

12. เน็ตเวิร์คอินเทอร์เฟสและการคอนฟิก

Xorp จะใช้คำว่า “interface” แทนเน็ตเวิร์คอินเทอร์เฟส และตามด้วยชื่อของอินเทอร์เฟสที่**ปรากฏ**บนระบบปฏิบัติการ เช่น dc0, dc1,le0,..., dcn เป็นต้น เช่น

interface dc0 {}

เครื่องหมาย {} ท้ายอินเทอร์เฟส จะเป็นการบอกรายละเอียดต่างๆ ของอินเทอร์เฟส ในแต่ละอินเทอร์เฟสสามารถสร้างอินเทอร์เฟสเสมือนได้ (virtual interface) ซึ่งจะใช้ชื่อว่า vif ความหมายของอินเทอร์เฟสเสมือนคือ การสร้างอินเทอร์เฟสในทางลอจิคอลหลายๆ ใบ (ด้วยซอฟต์แวร์) บนการ์ดเน็ตเวิร์คจริงที่มีเพียงใบเดียวได้ คอนฟิกของอินเทอร์เฟสเสมือนจะอยู่ภายใต้ interface ดังนี้

```
interfaces {
        restore-original-config-on-shutdown: false
        interface dc0 {
                description: "data interface" //คำอธิบายเพิ่มเติม
                disable: false      //สั่งให้การ์ดทำงานหรือไม่ทำงาน
                /* default-system-config */
                vif dc0 {                       //อินเทอร์เฟสเสมือน
                disable: false //สั่งให้อินเทอร์เฟสเสมือนทำงานหรือไม่ทำงาน
                address 10.10.10.10 { //กำหนด IPv4 ให้กับอินเทอร์เฟส
เสมือน
                        prefix-length: 24
                        broadcast: 10.10.10.255
                        disable: false
                 }
/*
                address 2001:DB8:10:10:10:10:10:10 {
                        //กำหนด IPv6 ให้กับอินเทอร์เฟสเสมือน
                        prefix-length: 64
                        disable: false
                }
*/

            }
        }
    }
```

2. ติดตั้งบนระบบปฏิบัติการวินโดวส์

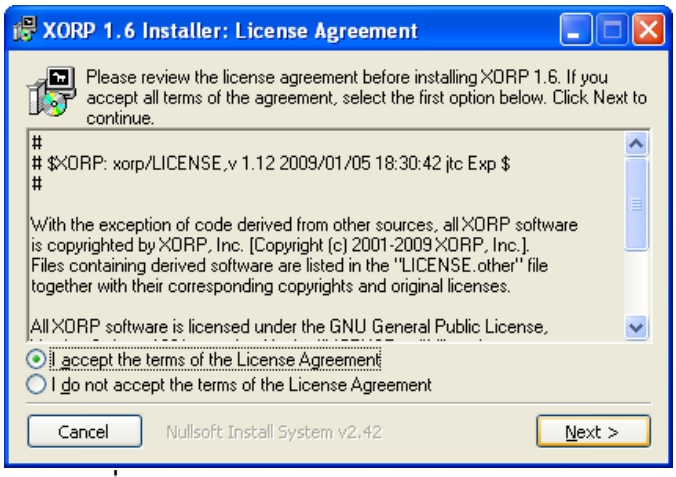
การติดตั้ง XORP บนวินโดวส์(ควรใช้วินโดวส์เซิร์ฟเวอร์ เช่น Windows 2003) สามารถทำได้ 2 วิธีคือ

- ติดตั้งด้วยการคอมไพล์จากซอร์สโค้ด (รหัสต้นฉบับ) จำเป็นต้องมีการติดตั้งซอฟต์แวร์เพิ่มเติมคือ Minimalist GNU for Windows (MinGW and MSYS) ซึ่งทำหน้าที่สร้างสภาพแวดล้อมการคอมไพล์ซอร์สโค้ด ประกอบไปด้วย 2 ส่วนคือ

MinGW เป็นคอมไพล์เลอร์ (GCC สำหรับวินโดวส์) และ MSYS ทำหน้าที่เป็นเชลล์สำหรับใช้งานบนวินโดวส์ (ผู้เขียนแนะนำว่ายังไม่ควรใช้งาน)

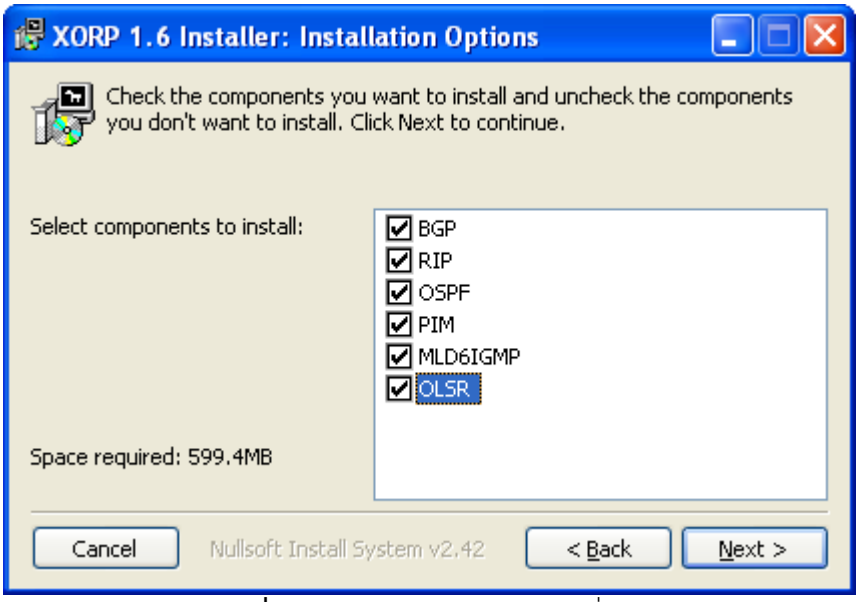
- ติดตั้งด้วยการ setup จากไฟล์ execute table มีชื่อว่า xorp-1.6-setup.exe (ดาวน์โหลดจาก http://www.xorp.org/downloads.html) ซึ่งมีขั้นตอนดังนี้

1. ดับเบิลคลิกที่ไฟล์ xorp-1.6-setup.exe จะปรากฏ license agreement



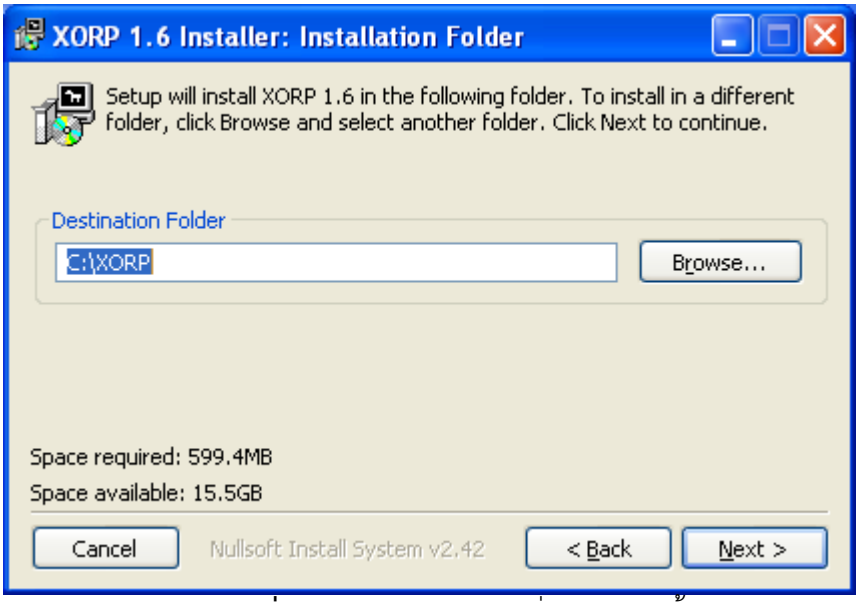
รูปที่ 13.44 xorp license agreement

2. เลือก I accept the ... คลิก Next >



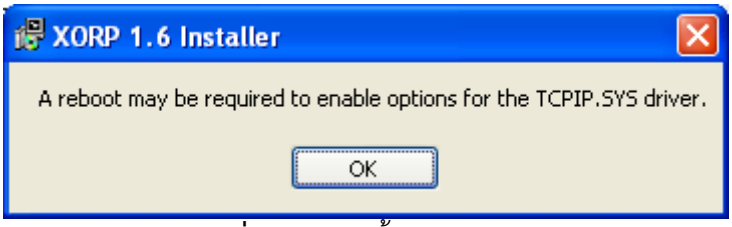
รูปที่ 13.45 Routing Protocol ที่สนับสนุน

3. **ปรากฏ**เมนูให้เลือกโพรโทคอล routing ที่ต้องการ ถ้าต้องการทดสอบการเราท์ติ้งทั้งหมดให้เลือกทุกตัว จากนั้นคลิก Next > และทำการเลือกไดเรคทรอรี่ที่ต้องการติดตั้ง default จะอยู่ที่ C:\XORP จากนั้นคลิก Next >



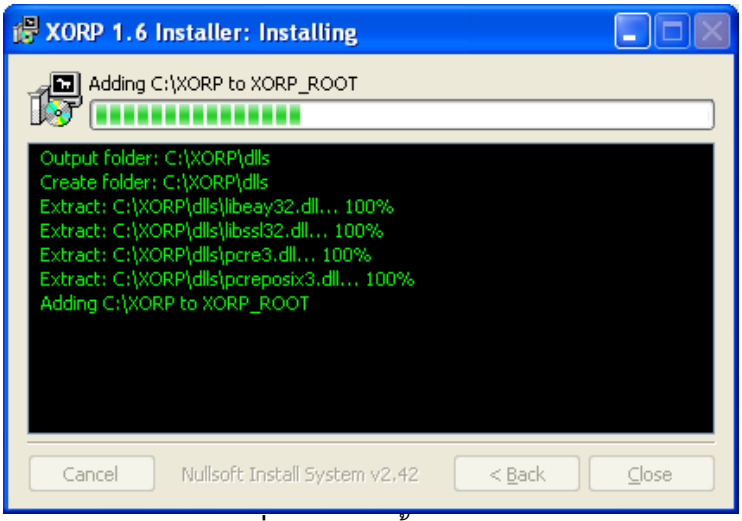
รูปที่ 13.46 เลือกโฟร์เดอร์ที่ต้องการติดตั้ง

4. จะมีเมนูแจ้งเตือนว่าอาจจะจำเป็นต้อง restart เครื่องใหม่หลังจากการติดตั้ง เนื่องจากจำเป็นต้องมีการลงไดร์เวอร์ ชื่อว่า TCPIP.SYS ด้วย ต่อจากนั้นเลือก OK



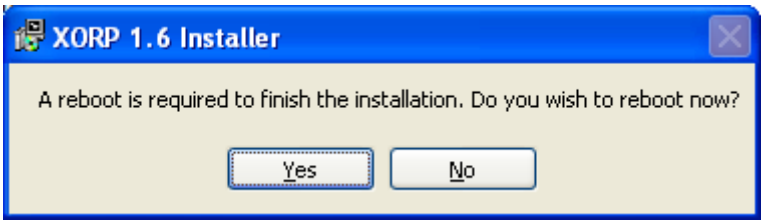
รูปที่ 13.47 ติดตั้ง TCPIP.SYS

5. จะ**ปรากฏ**เมนูให้ทำการเลือกว่าจะสั่งให้โปรแกรมทำงานใน feature ใดบ้างใน เบื้องต้นให้เลือกเฉพาะ IP Forward ทำงานเท่านั้นสำหรับคุณสมบัติอื่นๆ ยังไม่จำเป็นต้องเปิดใช้งาน จากนั้นเลือก Install



รูปที่ 13.48 ติดตั้งโปรแกรม

6. จะต้องมีการ restart ในขั้นตอนสุดท้าย



3. LiveCD (สามารถใช้งานได้โดยไม่มีการติดตั้ง)

XORP Live CD สามารถใช้งานได้ทันทีโดยที่ไม่จำเป็นต้องมีการติดตั้ง (สำหรับสถาปัตยกรรมแบบ x86 เท่านั้น) ข้อดีในการใช้ Live CD คือ

- ใช้งานได้ทันทีโดยไม่ต้องติดตั้งหรือคอมไพล์ซอร์สโค้ด และไม่จำเป็นต้องจัดเตรียมฮาร์ดดิสก์ เช่น format เพื่อจัดเก็บข้อมูล
- สามารถนำไปทำงานที่อื่นได้ทันที (Portable)
- สามารถใช้ในด้านการเรียนการสอน เช่น LAB ที่เกี่ยวข้องกับเน็ตเวิร์คได้ โดยไม่ต้องติดตั้ง

ข้อมูลไฟล์คอนฟิกกูเรชันจะถูกเก็บลงใน USB (ตั้งแต่เวอร์ชัน 1.5 เป็นต้นมา) Live CD จะใช้ระบบปฏิบัติการ FreeBSD 7.0 เป็นพื้นฐานในการพัฒนา

ดาวน์โหลด Live CD

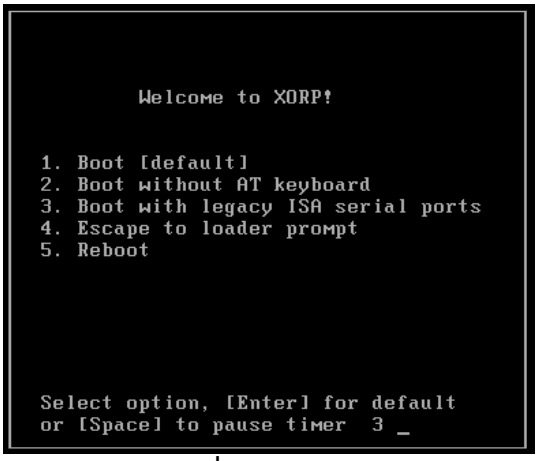
สามารถดาวน์โหลดผ่าน HTTP ได้จาก http://www.xorp.org/releases/1.6/XORP-1.6-LiveCD.iso.gz หรือดาวน์โหลดด้วยบิตทอเรนท์จาก http://releases.xorp.org.s3.amazonaws.com/1.6/XORP-1.6-LiveCD.iso.gz มีขนาดประมาณ 44 MB เมื่อดาวน์โหลดแล้วจำเป็นต้องเขียนลงแผ่น CD ก่อน (ข้อมูลการเขียนแผ่น CD สามารถอ่านข้อมูลเพิ่มเติมได้จาก Mac OSX[57], Windows[58], Linux[59], FreeBSD[60])

ก่อนการใช้งาน Live CD

เมื่อเขียน XORP ลงแผ่น CD เรียบร้อยแล้ว ขั้นตอนต่อไปให้ทำการเซ็ตไบออสให้ทำการอ่านข้อมูลจาก CD-ROM เป็นลำดับที่ 1 จากนั้นให้ทำการรีเซ็ตเครื่องคอมพิวเตอร์ เครื่องคอมพิวเตอร์จะเริ่มทำการโหลด XORP จากแผ่นมาทำงาน (เมื่อจำเป็นต้องบันทึกค่าคอนฟิกต่างๆ ขณะที่ XORP ทำงานจำเป็นต้องมี USB ติดตั้งอยู่ด้วย)

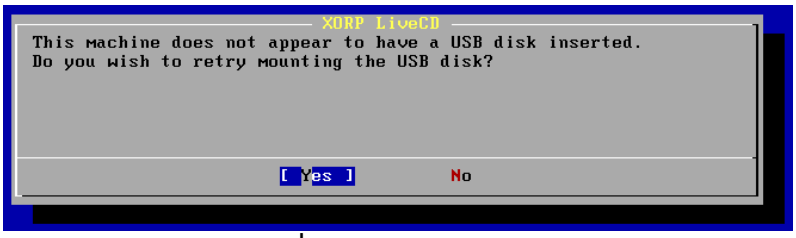
เริ่มต้นทำงาน Live CD

1. เลือกเมนูสำหรับเริ่มต้นระบบ ให้เลือก 1 หรือกดปุ่ม Enter ดังรูปที่ 13.49



รูปที่ 13.49 xorp

2. เมื่อเครื่องไม่มีการติดตั้งหรือ format USB ไว้ xorp จะเตือนว่าจะเพิ่ม USB เข้ามาในระบบก่อนหรือไม่ (เลือก Yes ต้องใส่ USB ไปยังคอมพิวเตอร์, เลือก No คือไม่ต้องการติดตั้ง USB) ดังรูปที่ 13.50

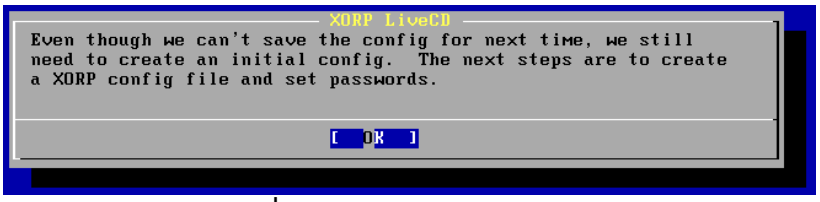


รูปที่ 13.50 format USB

3. ในกรณีที่ไม่ใช้ USB xorp จะเตือนว่าจะไม่สามารถบันทึกข้อมูลใดๆ เก็บไว้ได้เลย ในกรณีที่มีการแก้ไขเปลี่ยนแปลง แต่เราท์เตอร์ยังสามารถทำงานได้ (กรณีที่ไม่มี USB เลือก No) ดังรูปที่ 13.51

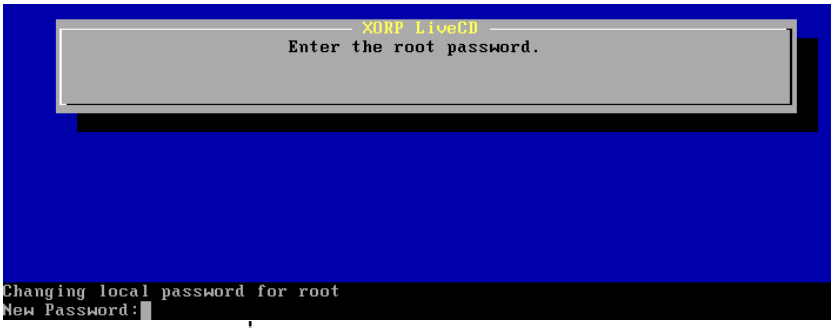
รูปที่ 13.51 XORP LiveCD

4. เมื่อ xorp ไม่สามารถบันทึกข้อมูลได้ ดังนั้นทุกครั้งที่มีการใช้งาน xorp จะต้องมีการคอนฟิกข้อมูลต่างๆ ในตอนเริ่มต้นทุกครั้ง เช่น รหัสผ่าน การ์ดเน็ตเวิร์ค เป็นต้น ดังรูปที่ 13.52



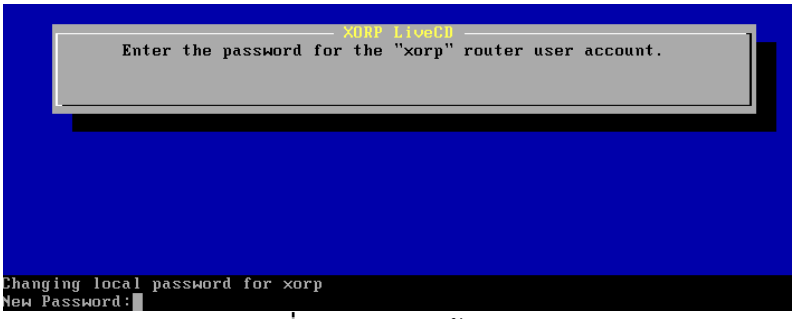
รูปที่ 13.52 การกำหนดรหัสผ่าน

5. กำหนดรหัสผ่านให้กับ root ดังรูปที่ 13.53



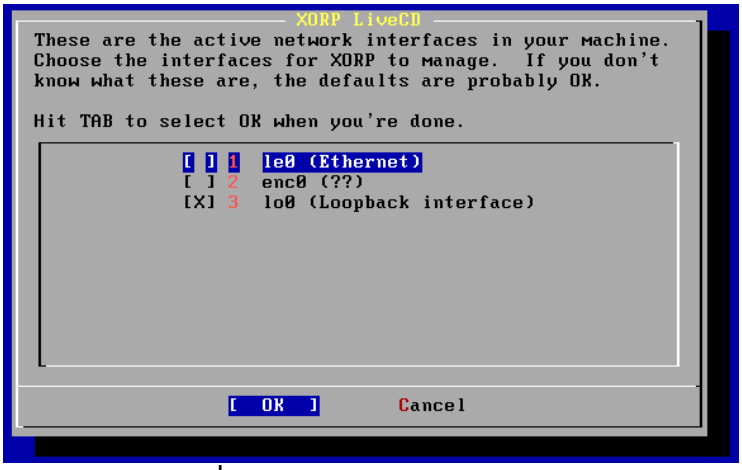
รูปที่ 13.53 กำหนดรหัสผ่านของ root

6. xorp จะสร้างชื่อผู้ใช้ขึ้นมาอัตโนมัติคือ user xorp ดังนั้นจำเป็นต้องกำหนดรหัสผ่านให้ xorp ด้วย ดังรูปที่ 13.54



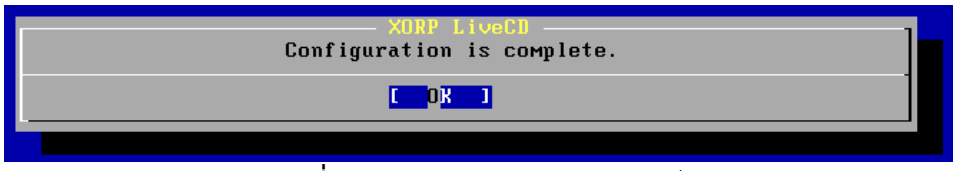
รูปที่ 13.54 การสร้าง user

7. เลือกการ์ดเน็ตเวิร์คที่ต้องการใช้งาน โดย default xorp จะเลือก Loopback interface ไว้แล้ว ให้ทำการเลือกการ์ดที่ต้องการใช้งาน (ในที่นี่เลือก le0 ethernet) ดังรูปที่ 13.55



รูปที่ 13.55 การกำหนด loopback

8. การปรับแต่งเราท์เตอร์เรียบร้อยแล้ว ให้เลือก OK จากนั้นเราท์เตอร์จะเริ่มทำงาน ให้รอสักครู่ ดังรูปที่ 13.56



รูปที่ 13.56 configuration สมบูรณ์

9. xorp เราท์เตอร์พร้อมใช้งาน โดยต้อง login เป็นผู้ใช้งานคนใดคนหนึ่งก่อน ระหว่าง root กับ xorp (ถ้าต้องการปรับแต่งที่เกี่ยวข้องกับระบบ เช่น การปรับแต่ง forwarding ให้ login ด้วย root แต่ถ้าต้องการทำงานอื่นๆ ที่ไม่เกี่ยวกับ system ให้ใช้ user xorp แทน) ดังรูปที่ 13.57

```
Fri Sep  4 22:36:20 UTC 2009

FreeBSD/i386 (xorpcd.local) (ttyv0)

login:
```

รูปที่ 13.57 XORP login

10. เมื่อต้องการสั่งงานเราท์เตอร์ให้ใช้คำสั่ง xorpsh เพื่อเข้าสู่เชลล์ จากนั้นทดลองใช้คำสั่ง ? ดังรูปที่ 13.58

```
xorpcd# xorpsh
Welcome to XORP on xorpcd.local
root@xorpcd.local> ?
Possible completions:
  configure            Switch to configuration mode
  exit                 Exit this command session
  help                 Provide help with commands
  ping                 Ping a hostname or IP address
  ping6                Ping an IPv6 hostname or IPv6 address
  quit                 Quit this command session
  show                 Display information about the system
  traceroute           Trace the IP route to a hostname or IP address
  traceroute6          Trace the IPv6 route to a hostname or IPv6 address
  usb                  -- No help available --
root@xorpcd.local>
```

รูปที่ 13.58 XORP เชลล์

การบันทึกข้อมูล

ไฟล์ควบคุมการทำงานของ xorp (xorp.conf) โดยปกติจะเก็บอยู่ที่ /etc/local/xorp.conf เมื่อมีการปรับแต่งเราท์เตอร์ ควรจะบันทึกข้อมูลที่ปรับแต่ง เก็บไว้ด้วย ซึ่งจะใช้คำสั่ง save ตามด้วยที่อยู่ที่ต้องการจัดเก็บ เช่น

xorp@LiveCD# save /etc/local/xorp.conf

ถ้าต้องการบันทึกข้อมูลลงบน USB ให้ใช้คำสั่งดังนี้

xorp@LiveCD> usb save

อินเทอร์เฟส

ชื่อของอินเทอร์เฟสที่ใช้งานอาจจะมีได้หลายชื่อ เช่น fxp0, fxp1, dc0, xl3, eth0, eth1 เป็นต้น ทั้งนี้ขึ้นอยู่กับชนิดของการ์ดที่นำมาติดตั้งและไดร์เวอร์

การติดตั้งเราท์เตอร์ค่าย Vyatta

Vyatta สามารถเลือกใช้งานได้หลายรูปแบบ ขึ้นอยู่กับความต้องการของผู้ใช้งานว่าต้องการใช้งานในลักษณะใด ตารางที่ **13.10** แสดงการเลือกใช้งานซอฟต์แวร์เราท์เตอร์ Vyatta

ตารางที่ 13.10 แสดงการเลือกใช้งานซอฟต์แวร์เราท์เตอร์ Vyatta

รูปแบบ	ลักษณะการทำงาน	เหมาะสำหรับ
Live CD	ไม่จำเป็นต้องมีการติดตั้ง แต่จำเป็นต้องมี floppy disk เอาไว้บันทึกค่าที่คอนฟิกกูเรชันที่เปลี่ยนแปลง	ทดสอบการใช้งานเราท์เตอร์, ทำ LAB เน็ตเวิร์ค
ติดตั้งลงบนเครื่องคอมพิวเตอร์	ทำการติดตั้งซอฟต์แวร์ทั้งหมดลงบนเครื่องจาก Live CD ซึ่งจำเป็นต้องมีฮาร์ดดิสก์, compact flash, USB ค่าคอนฟิกจะบันทึกลงบนคอมพิวเตอร์	ติดตั้งใช้งานจริง, ทดสอบประสิทธิภาพของฮาร์ดแวร์ที่ใช้
ติดตั้งบนเครื่องจักรเสมือน	ทำการติดตั้งซอฟต์แวร์ทั้งหมดลงบนเครื่องจักรเสมือนจาก Live CD วิธีการนี้ทำให้สามารถติดตั้งเราท์เตอร์ได้มากกว่า 1 ตัวบนฮาร์ดแวร์เครื่องเดียว	ติดตั้งใช้งานจริง, ทดสอบประสิทธิภาพของฮาร์ดแวร์ที่ใช้, ประหยัดการจะซื้อฮาร์ดแวร์จำนวนมาก

การรัน Vyatta จาก Live CD

- อันดับแรกให้ทำการดาวน์โหลด ISO ไฟล์ จากเว็บไซต์ http://www.vyatta.com จากนั้นทำการเขียนไฟล์ลงบนแผ่น CD (เป็นการเขียน Image ลง CD ไม่ใช่การ Copy ลงแผ่น CD เพราะกรณีหลังจะทำให้แผ่นไม่สามารถ boot ได้)
- เซ็ตให้ไบออสสามารถอ่านข้อมูลจาก CD-ROM เป็นอันดับแรก
- ใส่แผ่น CD ที่เขียนจาก Image เข้าไปใน CD-ROM จากนั้นทำการ restart เครื่องใหม่
- กระบวนการเริ่มต้นของระบบจะใช้เวลาสักพักหนึ่ง เมื่อเสร็จแล้วจะมีข้อความแสดงความพร้อมสำหรับการทำงาน

  vyatta login:
- เมื่อถึงขั้นตอนดังกล่าวแสดงว่าพร้อมใช้งานแล้วให้ข้ามไปหัวข้อ “ทดสอบการติดตั้ง”

การติดตั้งลงบนฮาร์ดดิสก์

- เขียนแผ่น CD ด้วยไฟล์ ISO จากนั้น boot ด้วย CD-ROM

- ขนาดพื้นที่ที่ใช้ติดตั้งซอฟต์แวร์จะใช้ประมาณ 450 MB สำหรับพาทิชันของ root (ถ้าต้องการติดตั้งและใช้งานจริงๆ ควรมีพื้นที่อย่างน้อย 2 GB และชื่อของฮาร์ดดิกส์โดยปกติจะมีชื่อเป็น /dev/sda, /dev/hda)
- ใส่แผ่น CD ที่เขียนจาก Image เข้าไปใน CD-ROM จากนั้นทำการ restart เครื่องใหม่
- กระบวนการเริ่มต้นของระบบจะใช้เวลาสักพักหนึ่ง เมื่อเสร็จแล้วจะมีข้อความแสดงความพร้อมสำหรับการทำงาน

  vyatta login:
- ทำการ Login ด้วยชื่อผู้ใช้ชื่อ root รหัสผ่านเบื้องต้นคือ vyatta
- ให้พิมพ์คำสั่ง install-system

  vyatta:~# install-system
- โปรแกรม Installer จะเริ่มทำงาน จะปรากฏข้อความแนะนำให้ทำการคอนฟิกระบบไปทีละขั้นตอน ดังนี้

  Would you like to continue? (Yes/No) [Yes]: **<Enter>**

  ต้องการติดตั้งเราท์เตอร์หรือไม่ ให้กดปุ่ม Enter
- ขั้นตอนต่อไปจะเป็นการจัดการ partition ว่าต้องการใช้วิธีใด กดปุ่ม Enter ทำการสร้าง partition แบบอัตโนมัติ

  Partition (Auto/Union/Parted/Skip) [Auto]: **<Enter>**
- ต้องการติดตั้งซอฟต์แวร์เราท์เตอร์ลงบน ฮาร์ดดิสก์แบบใด (โดย default จะเป็น sda) กดปุ่ม Enter

  Install the image on? [sda] **<Enter>**
- ปรากฏข้อความเตือนว่าข้อมูลที่มีอยู่บนฮาร์ดดิสก์ที่ต้องการติดตั้ง อาจจะเสียหายได้ เลือก Yes

  This will destroy all data on /dev/sda.

  Continue? (Yes/No) [No]: **Yes**
- ปรากฏข้อความว่า ไดร์เรคทรอรี่หลักของระบบคือ root จำเป็นต้องใช้พื้นที่ประมาณ 1000 – 1074 MB ซึ่งผู้ใช้งานสามารถเลือกขนาดพื้นที่ได้ตามความต้องการ ในกรณีนี้แนะนำให้ใช้พื้นที่ไม่ควรต่ำกว่า 1000 MB (ถ้าจำเป็นต้องมีการเก็บ Log file ต่างๆ เพิ่ม ควรใช้พื้นที่มากขึ้น)

  How big of a root partition should I create? (1000MB - 1074MB) [1074]MB: **1000**

- ระบบจะทำการสร้างโครงสร้างของไฟล์ตามขนาดที่กำหนด และคัดลอกซอฟต์แวร์ระบบลงบนฮาร์ดดิสก์ ให้รอสักครู่

  Creating filesystem on /dev/sda1: OK

  Mounting /dev/sda1

  Copying system image files to /dev/sda1:OK

  I found the following configuration files

- ขั้นตอนต่อไป ระบบจะแสดงผลการติดตั้งว่า ไฟล์ที่ควบคุมระบบ (configuration file) จะอยู่ที่ /opt/vyatta/etc/config/config.boot จะให้บันทึกไฟล์ลงฮาร์ดดิสก์หรือไม่ (ควรคัดลอกลงไปเนื่องจากเป็นไฟล์ที่ใช้เป็นตัวอย่างในการปรับแต่งระบบในอนาคตได้) ให้กดปุ่ม Enter

  /opt/vyatta/etc/config/config.boot

  Which one should I copy to sda?

  [/opt/vyatta/etc/config/config.boot]: **<Enter>**

- ขั้นตอนต่อไป ระบบจะแจ้งให้ทราบว่าต้องการกำหนดรหัสผ่านให้กับผู้ดูแลระบบหรือไม่ (ถ้าต้องการกำหนดรหัสผ่านใหม่ให้เลือก Yes ถ้าไม่ต้องการให้กด NO) ในที่นี้ให้พิมพ์ No

  Would you like to set passwords for system users (Yes/No) [Yes]: No

- ขั้นตอนต่อไป ระบบจะถามว่าต้องการติดตั้ง GRUB boot loader บนฮาร์ดดิสก์หรือไม่ ให้กดปุ่ม Enter

  Which drive should GRUB modify the boot partition on? [sda]:

  <Enter>

- เมื่อไม่มีข้อความผิดพลาดเกิดขึ้น แสดงว่าการติดตั้งสมบูรณ์แล้ว จะ**ปรากฏ**ข้อความดังต่อไปนี้

  Setting up grub: OK

  Done!

  vyatta:~#

- Restart เครื่องคอมพิวเตอร์อีกครั้ง จากนั้นรอสักครู่ จะปรากฏหน้าจอพร้อมใช้งาน ทดสอบโดยการ login ด้วย root รหัสผ่าน default คือ vyatta (หรือตามที่กำหนดไว้ข้างต้น) ดังรูปที่ 13.59

```
Welcome to Vyatta - vyatta tty1

vyatta login: root
Password:
Linux vyatta 2.6.26-1-486-vyatta #1 SMP Fri Feb 27 01:04:20 GMT 2009 i686
Welcome to Vyatta.
This system is open-source software. The exact distribution terms for
each module comprising the full system are described in the individual
files in /usr/share/doc/*/copyright.
vyatta:~#
```

รูปที่ 13.59 Vyatta พร้อมใช้งาน

การติดตั้งลงบน USB Memory Strick

ในการติดตั้งลงบน USB จะมีขั้นตอนคล้ายกับการติดตั้งลงบนฮาร์ดดิสก์ แตกต่างกันโดยสื่อที่ใช้จัดเก็บเท่านั้น (เครื่องคอมพิวเตอร์บางเครื่องอาจจะไม่สนับสนุน USB ต้องทำการตรวจสอบใน BIOS เสียก่อนว่ารองรับ USB หรือไม่) ซึ่งมีขั้นตอนดังต่อไปนี้

- ตรวจสอบ BIOS ว่าสามารถสั่งให้ทำงานจาก USB ได้
- เขียนแผ่น Image ลงบน CD
- กำหนด BIOS ให้เริ่มระบบจาก CD-ROM
- การติดตั้งซอฟต์แวร์ vyatta ลงบน USB ควรมีพื้นที่อย่างน้อย 512 MB สำหรับโครงสร้างไฟล์หลักคือ root (ควรมีขนาดอย่างน้อย 2 GB สำหรับการใช้งานจริง)
- ใส่แผ่นติดตั้งใน CD-ROM จากนั้นให้ restart เครื่องคอมพิวเตอร์
- ขั้นตอนถัดไปจะเหมือนกับการติดตั้งจากฮาร์ดดิสก์ จะแตกต่างกันตรงที่เมื่อระบบให้เลือกสื่อที่จะติดตั้งให้เลือกเป็น USB แทนฮาร์ดดิสก์ (โดยปกติฮาร์ดดิสก์จะมีโครงสร้างเป็น /dev/sda สำหรับ USB จะเป็น /dev/sdb)
- เมื่อทำการติดตั้งเสร็จสิ้น ให้ทำการ restart เครื่องอีกครั้ง ในระหว่างเริ่มต้นระบบให้เข้าไปเซ็ต BIOS ให้เลือก Boot จาก USB แทนฮาร์ดดิสก์หรือจากแผ่น CD-ROM (โดยทั่วไปการเข้าไปกำหนดค่า BIOS จะกดปุ่ม F2 ระหว่างระบบทำงาน จากนั้นเข้าไปเมนูเปลี่ยนลำดับการ boot ให้เลือกเป็น USB HDD เลือก save แล้ว restart เครื่องใหม่อีกครั้ง) ดังรูปที่ 13.60

รูปที่ 13.60 USB Memory Strick

การติดตั้งลงบน Compact Flash

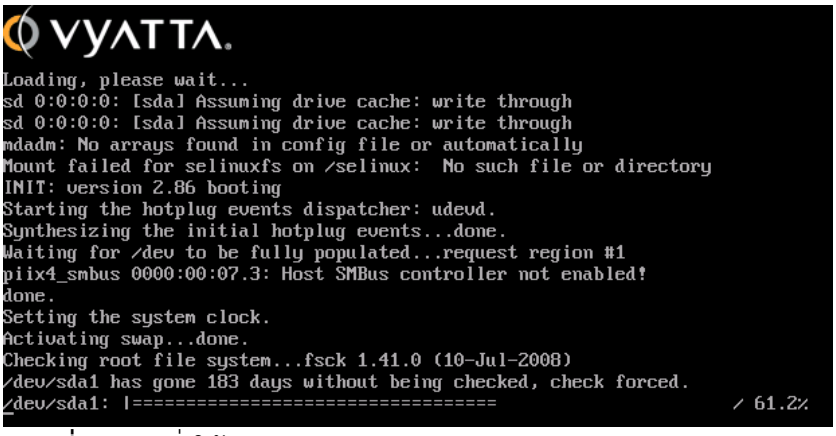
ในการติดตั้งลงบน Compact Flash จะมีขั้นตอนการติดตั้งเช่นเดียวกับการลงบนฮาร์ดดิสก์หรือ USB แตกต่างกันโดยสื่อที่ใช้จัดเก็บเป็น Compact Flash แทน ดังรูปที่ 13.61

รูปที่ 13.61 Compact Flash

การติดตั้งลงบนเครื่องจักรเสมือน (Virtual Machine)

Vyatta สามารถทำงานบนเครื่องจักรเสมือนได้ การใช้เทคโนโลยีดังกล่าวนี้จะช่วยให้ผู้ใช้งานสามารถลดต้นทุนในการจัดซื้อฮาร์ดแวร์ได้มาก รวมถึงต้องการทดสอบการทำงานของระบบบางอย่างก่อนจะใช้งานจริง และที่สำคัญคือสามารถติดตั้งเราท์เตอร์ได้หลายๆ ตัวบนเครื่องฮาร์ดแวร์เพียงตัวเดียว สำหรับ virtual machine ที่สนับสนุนการทำงานมีให้เลือกหลายค่าย เช่น VMware[60], XEN[61], VirtualBox[62], Hyper-V [61] เป็นต้น เพื่อความสะดวกในการใช้งาน Vyatta ได้ทำการสร้าง virtual machine บน VMware และ Xen ไว้ให้เรียบร้อยแล้ว สามารถสั่งให้ทำงานได้ทันทีโดยไม่จำเป็นต้องมีการติดตั้งใดๆ เพิ่มเติม ซึ่งมีขั้นตอนการใช้งานดังนี้

- ดาวน์โหลดไฟล์ Wmware Virtual Machine หรือ Xen Virtual Machine ได้จาก http://www.vyatta.com/downloads/swdl.php
- ทำการแตกไฟล์ที่บีบอัดไว้ ด้วยโปรแกรม winzip หรือ winrar
- เปิดโปรแกรม VMware Server Console (ต้องทำการติดตั้งโปรแกรม VMware หรือ Xen ก่อน) เมนู file → Open ให้เลือกไฟล์ในโฟร์เดอร์ที่แตกไฟล์บีบอัดไว้ มีนามสกุล .vmx ดังรูปที่ 13.62



รูปที่ 13.62 สั่งให้ Vyatta ทำงานจาก VMware Virtual Machine

- เมื่อกระบวนการ boot เสร็จ ให้ทำสอบด้วยการ login ด้วย user root รหัสผ่านคือ vyatta

NOTE: เมื่อต้องการสลับทำงานระหว่างระบบปฏิบัติการที่ใช้งานอยู่กับ VMware ให้กดปุ่ม Ctrl พร้อมกับ alt พร้อมๆ กัน

ทดสอบการติดตั้ง

สำหรับการทดสอบสามารถทำได้ โดยการคอนฟิกอินเทอร์เฟสใดอินเทอร์เฟสหนึ่งในเราท์เตอร์ และทำการทดสอบโดยการ ping (โพรโทคอล ICMP ใช้ทดสอบการทำงานของอุปกรณ์บนเครือข่ายว่าทำงานอยู่หรือไม่) ซึ่งมีขั้นตอนการทดสอบดังต่อไปนี้

- Login ด้วยชื่อผู้ใช้คือ root รหัสผ่าน vyatta (default) หรือรหัสผ่านที่ได้กำหนดไว้
- ที่ command prompt ทำการกำหนดไอพีแอดเดรสและซับเน็ตเวิร์ค ให้กับอินเทอร์เฟส eth0 (ethernet interface) มีข้อมูลดังนี้

  IP Address = 192.168.1.1

  Subnet = 255.255.255.0 หรือ /24
- เริ่มต้นคอนฟิก โดยใช้คำสั่ง configure แล้วกดปุ่ม Enter

  vyatta@vyatta:~$ **configure**

  [edit]
- ออกคำสั่ง set เพื่อกำหนดข้อมูลต่างๆ ให้กับอินเทอร์เฟส eth0

  root@vyatta# **set interfaces ethernet eth0 address 192.168.1.1/24**

  [edit]
- สั่งให้เราท์เตอร์ประมวลผลคำสั่งด้วยคำสั่ง commit และออกจากการคอนฟิกเราท์เตอร์ด้วยคำสั่ง exit

  root@vyatta# **commit**

  [edit]

  root@vyatta# **exit**

  exit

  vyatta@vyatta:~$
- ให้ทดสอบด้วยการ ping จากเครื่องที่เชื่อมต่อใน subnet เดียวกัน (อาจจะเชื่อมต่อในสวิชต์ตัวเดียวกันโดยมีไอพีอยู่ในช่วง 192.168.1.2-192.168.1.254)

  ping 192.168.1.1

ผลลัพธ์เมื่อเราท์เตอร์มีการตอบสนองที่ถูกต้อง

ping 192.168.1.1

PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.

64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=1 ttl=52 time=87.7 ms

64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=2 ttl=52 time=95.6 ms

64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=3 ttl=52 time=85.4 ms

64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=4 ttl=52 time=95.8 ms

64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=5 ttl=52 time=87.0 ms
64 bytes from 192.168.1.1 (192.168.1.1): icmp_seq=6 ttl=52 time=97.6 ms

--- 192.168.1.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 8998ms
rtt min/avg/max/mdev = 78.162/89.213/97.695/6.836 ms

13.5 การคอนฟิกเราท์เตอร์ขั้นพื้นฐาน (Basic Router Configurations)

ในบทนี้จะกล่าวถึงโครงสร้างของคำสั่ง คำสั่งพื้นฐานเบื้องต้น และวิธีการคอนฟิกเราท์เตอร์ในเบื้องต้น เพื่อใช้เป็นพื้นฐานในการคอนฟิกเราท์เตอร์ในขั้นลึกซึ้งต่อไป

13.5.1 เกริ่นนำ

ในการควบคุมหรือคอนฟิกเราท์เตอร์ค่าย XORP สามารถทำได้โดยใช้ Command line interface (CLI) ผ่านโปรแกรมเชลล์ชื่อว่า xorpsh รูปแบบของคำสั่งจะมีความคล้ายคลึงกับ Juniper Router สำหรับเราท์เตอร์จากค่าย Vyatta จะสามารถคอนฟิกเราท์เตอร์ได้ 2 แบบคือ ผ่าน CLI และสามารถคอนฟิกผ่านเว็บอินเทอร์เฟส (GUI) ได้อีกทางหนึ่งด้วย ทั้ง 2 ค่ายจะมีรูปแบบคำสั่งที่ใกล้เคียงกัน อาจจะมีรายละเอียดบางอย่างที่ไม่เหมือนกัน เมื่อเข้าใจคำสั่งของเราท์เตอร์ค่ายใดค่ายหนึ่งก็จะทำให้สามารถเข้าใจคำสั่งของอีกค่ายได้อย่างไม่ยากนัก

13.5.2 โครงสร้างคำสั่งของเราท์เตอร์ XORP และการสร้างสคริป

Xorp จะควบคุมการเราท์เตอร์ด้วยโปรแกรม xorp_rtrmgr เมื่อต้องการสั่งงานเราท์เตอร์สามารถสั่งงานผ่านเชลล์ xorpsh โดยมีเงื่อนไงว่าผู้ใช้งานที่ต้องการคอนฟิกเราท์เตอร์ต้องเป็นสมาชิกของกลุ่ม xorp เสียก่อน

- คำสั่ง ?

  เมื่อสั่งเชลล์ xorpsh ทำงานจะ**ปรากฏ** prompt ดังนี้

```
user@hostname>
```

เมื่อต้องการออกจาก xorpsh ให้ใช้คำสั่ง exit หรือ ctrl – d (กดปุ่ม ctrl พร้อมกับปุ่ม d)

ต้องการแสดงรายการคำสั่งอื่นๆ ที่เราท์เตอร์มีให้ใช้งาน กดปุ่ม “?”

```
user@hostname> ?
Possible completions:
        configure       Switch to configuration mode
        exit            Exit this command session
        help            Provide help with commands
        quit            Quit this command session
        show            Display information about the system
```

การป้อนคำสั่งสามารถป้อนอักษรเพียง 1-2 ตัวแรก จากนั้นกดปุ่ม TAB คำสั่งที่ตรงกันก็จะปรากฏขึ้น หรือใช้สัญลักษณ์ ? ร่วมกับตัวอักษร จะส่งผลให้คำสั่งที่ใกล้เคียงกันปรากฏขึ้น

```
user@hostname> config?
Possible completions:
        configure        Switch to configuration mode
user@hostname> conf
```

เมื่อใช้สัญลักษณ์หลังคำสั่ง จะทำให้เชลล์แสดงคำสั่งที่ต่อเนื่องกันมาแสดงผล

```
user@hostname> configure ?
Possible completions:
        <[Enter]>        Execute this command
        exclusive        Switch to configuration mode, locking out other users
user@hostname> configure
```

- คีย์คำสั่งต่างๆ

คีย์คำสั่ง	ความหมาย
Up-arrow or control-p	เรียกคำสั่งเก่าที่ใช้งานผ่านมา
down-arrow or control-n	เรียกคำสั่งต่อไปที่ใช้งานผ่านมา
Left-arrow or control-b	เลื่อนที่เคอร์เซอร์ไปยังส่วนหัวของคำสั่งครั้งละ 1 ตัวอักษร
right-arrow or control-f	เลื่อนที่เคอร์เซอร์ไปยังส่วนท้ายของคำสั่งครั้งละ 1 ตัวอักษร
control-a	เลื่อนที่เคอร์เซอร์ไปส่วนหัวของคำสั่ง
control-e	เลื่อนที่เคอร์เซอร์ไปส่วนท้ายของคำสั่ง
control-d	ลบคำสั่งที่เคอร์เซอร์อยู่
control-t	สลับอักษรที่ตำแหน่งเคอร์เซอร์อยู่กับตัวอักษรก่อนหน้า
control-space	ระบุตำแหน่งที่เคอร์เซอร์อยู่
control-w	ลบข้อความตั้งแต่ตำแหน่งที่ระบุถึงตำแหน่งที่เคอร์เซอร์อยู่ และเก็บข้อมูลที่ลบไว้ในบัฟเฟอร์ด้วย
control-k	ลบข้อความตั้งแต่ตำแหน่งที่เคอร์เซอร์อยู่ถึงตำแหน่งสุดท้ายของคำสั่ง และเก็บข้อมูลที่ลบไว้ในบัฟเฟอร์ด้วย
control-y	คัดลอกข้อมูลจากบัฟเฟอร์วางลงในตำแหน่งที่เคอร์เซอร์อยู่

- การแสดงผลการทำงาน

xorpsh จะแสดงผลลัพธ์ของการทำงานภายในขอบเขตที่จำกัด ดังนั้นถ้ามีการแสดงผลคำสั่งที่มากว่าที่กำหนดไว้ จะแสดงผลเหมือนกับ unix คือมีคำว่า –More- ท้ายข้อความ เมื่อต้องการความช่วยเหลือในการใช้คำสั่ง more ให้พิมพ์อักษร h เชลล์ก็จะแสดงข้อมูลของการใช้คำสั่ง more ให้ทราบ เช่น

SUMMARY OF MORE COMMANDS

-- Get Help --

h	* Display this help.
-- Scroll Down --	
Enter Return j	* Scroll down one line.
ˆM ˆN DownArrow	
Tab d ˆD ˆX	* Scroll down one-half screen.
Space ˆF	* Scroll down one whole screen.
ˆE G	* Scroll down to the bottom of the output.
N	* Display the output all at once instead of one screen at a time. (Same as specifying the | no-more command.)
-- Scroll Up --	
k ˆH ˆP	* Display the previous line of output.
UpArrow	
u ˆU	* Scroll up one-half screen.
b ˆB	* Scroll up one whole screen.
ˆA g	* Scroll up to the top of the output.
-- Misc Commands --	
ˆL	* Redraw the output on the screen.
q Q ˆC ˆK	* Interrupt the display of output.
--More-- (END)	

- การคัดกรองคำสั่ง (Filtering)

Xorp มีคำสั่งที่ช่วยให้สามารถคัดกรองคำสั่งได้ โดยการใช้เครื่องหมาย | ต่อท้ายคำสั่งที่ต้องการคัดกรอง ผลที่ได้จะคล้าย pipe ของคำสั่ง UNIX

```
user@hostname> show host date | ?
Possible completions:
        count           Count occurrences
        except          Show only text that does not match a pattern
        find            Search for the first occurrence of a pattern
        hold            Hold text without exiting the --More-- prompt
        match           Show only text that matches a pattern
        no-more         Don't paginate output
        resolve         Resolve IP addresses (NOT IMPLEMENTED YET)
        save            Save output text to a file (NOT IMPLEMENTED YET)
        trim            Trip specified number of columns from the start line
        (NOT IMPLEMENTED YET)
```

คำสั่งข้างต้นคือการแสดงผลข้อมูลของวัน และเวลาของระบบ เมื่อนำมารวมกับ | และ ? แสดงถึง ให้คัดกรองคำสั่งที่ได้จาก show host date อีกครั้ง ถ้าเราใส่ count เพิ่มเข้าไปผลลัพธ์ที่ได้จะเป็นการนับวันและเวลาเป็นจำนวนบรรทัด

```
user@hostname> show host date | count
Count : 1 lines
user@hostname>
```

- โหมดของคำสั่ง

  Xorp แบ่งออกเป็น 2 โหมดคือ

Operational mode ใช้สำหรับงานทั่วๆ ไปได้ เช่น แสดงสถานะ ดูข้อมูลต่างๆ บางส่วน Configuration mode ใช้กำหนด เปลี่ยนแปลง แก้ไขค่าคอนฟิกกูเรชัน โหลดและบันทึกข้อมูล เมื่อจะกล่าวให้ง่ายๆ คือ Operation mode เป็นโหมดของผู้ใช้ธรรมดา (User Mode) ที่ไม่ควรมีสิทธิ์ในการจัดการเราท์เตอร์ แต่ถ้าเป็น Configuraion mode จะเป็นผู้ใช้งานประเภทผู้ดูแลระบบ(Administrator Mode) ที่มีสิทธิ์ทุกอย่างในการบริหารจัดการเราท์เตอร์ได้

- โหมด Operation

  สัญลักษณ์ที่แสดงถึง Operation Mode คือ >

```
user@hostname> ?
Possible completions:
        configure       Switch to configuration mode
        exit            Exit this command session
        help            Provide help with commands
        ping            Ping a hostname or IP address
        ping6           Ping an IPv6 hostname or IPv6 IP address
        quit            Quit this command session
        show            Display information about the system
        traceroute      Trace the IP route to a hostname or IP address
        traceroute6     Trace the IPv6 route to a hostname or IPv6 address
```

สำหรับคำสั่งที่สามารถใช้ได้ในโหมดนี้มีดังนี้คือ

configure: คือคำสั่งที่ใช้เปลี่ยนโหมดระหว่าง Operation mode กับ Configuration mode

exit: ออกจาก xorp เชลล์

help: คำสั่งที่ช่วยเหลือ

ping: คำสั่ง ping เครื่องปลายทาง

ping6: คำสั่ง ping เครื่องปลายทาง สำหรับ IP เวอร์ชัน 6

quit: คำสั่งออกจาก xorp เชลล์เหมือนกับ exit

show: แสดงข้อมูลของระบบของเราท์เตอร์

traceroute: แสดงเส้นทางไปยังโฮสต์ปลายทาง

traceroute6: แสดงเส้นทางไปยังโฮสต์ปลายทาง สำหรับ IP เวอร์ชัน 6

- คำสั่ง show

เป็นคำสั่งที่ใช้สำหรับแสดงรายละเอียดของคำสั่งที่มีให้ใช้งาน และข้อมูลการทำงานของเราเตอร์ ตัวอย่างเช่น เมื่อต้องการแสดงข้อมูลของโพรโทคอล BGP จะใช้คำสั่ง show bgp

```
user@hostname> show ?
Possible completions:
        bgp             Display information about BGP
        host            Display information about the host
        igmp            Display information about IGMP
        interfaces      Show network interface information
        mfea            Display information about IPv4 MFEA
        mfea6           Display information about IPv6 MFEA
        mld             Display information about MLD
        pim             Display information about IPv4 PIM
        pim6            Display information about IPv6 PIM
        rip             Display information about RIP
        route           Show route table
        version         Display system version
user@hostname> show
```

```
user@hostname> show bgp peers detail
OK
        Peer 1: local 192.150.187.108/179 remote 192.150.187.109/179
        Peer ID: 192.150.187.109
        Peer State: ESTABLISHED
        Admin State: START
        Negotiated BGP Version: 4
        Peer AS Number: 65000
        Updates Received: 5157, Updates Sent: 0
        Messages Received: 5159, Messages Sent: 1
        Time since last received update: 4 seconds
        Number of transitions to ESTABLISHED: 1
        Time since last entering ESTABLISHED state: 47 seconds
        Retry Interval: 120 seconds
        Hold Time: 90 seconds, Keep Alive Time: 30 seconds
        Configured Hold Time: 90 seconds, Configured Keep Alive Time: 30 seconds
        Minimum AS Origination Interval: 0 seconds
        Minimum Route Advertisement Interval: 0 seconds
```

- โหมด Configuraion

```
user@hostname> configure
Entering configuration mode.
There are no other users in configuration mode.
[edit]
user@hostname#
```

การเข้าสู่โหมด Configuration ด้วยคำสั่ง configure สัญลักษณ์ของ prompt จะเปลี่ยนจาก > เป็น #

```
[edit]
user@hostname# ?
Possible completions:
        commit          Commit the current set of changes
        create          Alias for the "set" command (obsoleted)
        delete          Delete a configuration element
        edit            Edit a sub-element
        exit            Exit from this configuration level
        help            Provide help with commands
        load            Load configuration from a file
        quit            Quit from this level
        run             Run an operational-mode command
```

```
        save            Save configuration to a file
        set             Set the value of a parameter or create a new element
        show            Show the configuration (default values may be suppressed)
        top             Exit to top level of configuration
        up              Exit one level of configuration
user@hostname#
```

โครงสร้างของไฟล์คอนฟิกกูเรชันของเราท์เตอร์จะมีโครงสร้างเหมือนกับโครงสร้างของระบบไฟล์ใน UNIX เมื่อต้องการแสดงผลคอนฟิกกูเรชันปัจจุบันหรือส่วนใดส่วนหนึ่งของคอนฟิกให้ใช้คำสั่ง show แล้วตามด้วยข้อมูลที่ต้องการ เช่น เมื่อต้องการแสดงผลคอนฟิกกูเรชันของอินเทอร์เฟส ให้ใช้คำสั่ง show interfaces

```
[edit]
user@hostname# show interfaces
        interface rl0 {
                description: "control interface"
                vif rl0 {
                        address 192.168.1.1 {
                        prefix-length: 24
                        broadcast: 192.168.1.255
                        }
                }
        }
```

คำสั่ง show จะไม่แสดงข้อมูลที่เป็น default ให้เห็นแต่ถ้าต้องการแสดงข้อมูลในส่วนของ default ให้ใช้คำสั่ง show –all แล้วตามด้วยข้อมูลที่ต้องการทราบ เช่น show –all insterfaces

```
[edit]
user@hostname# show -all interfaces
        interface rl0 {
                description: "control interface"
                vif rl0 {
                        address 192.150.187.108 {
                        prefix-length: 25
                        broadcast: 192.150.187.255
                        disable: false
                        }
                disable: false
                }
        disable: false
        discard: false
        unreachable: false
        management: false
}
targetname: "fea"
```

- โครงสร้างคำสั่ง

โครงสร้างคำสั่งของ xorp มีรูปแบบคล้ายทรี ในขณะที่คอนฟิกจำเป็นต้องเคลื่อนที่ไปยังส่วนต่างๆ ของโครงสร้างทรีเสมอๆ ดังนั้นจำเป็นต้องทราบถึงคำสั่งที่จำเป็นต้องใช้ในการท่องไปยังโครงของคำสั่งดังนี้

- edit <element name>: แก้ไขคอนฟิก
- exit: ออกจากคำสั่งในแต่ละ level เมื่ออยู่ที่ level สูงที่สุดแล้วจะออกจากโหมด config
- quit: : ออกจากคำสั่งใน level ปัจจุบัน
- top: ออกจาก level ใดๆ ไปยัง level สูงสุด
- up: ออกไป 1 level

```
[edit]
user@hostname# edit interfaces interface rl0 vif rl0
[edit interfaces interface rl0 vif rl0]
user@hostname# show
        address 192.150.187.108 {
                prefix-length: 25
                broadcast: 192.150.187.255
        }

[edit interfaces interface rl0 vif rl0]
user@hostname# up
[edit interfaces interface rl0]
user@hostname# top
[edit]
user@hostname#
```

- Load และ save คอนฟิกกูเรชัน

คอนฟิกกูเรชันไฟล์เป็นไฟล์สำคัญที่เก็บข้อมูลเพื่อบอกว่าเราท์เตอร์ต้องทำอะไรบ้าง ในการทำงานของเราท์เตอร์ xorp_rtrmgr จะทำการโหลดคอนฟิกกูเรชันไฟล์มาทำงานในตอนเริ่มเปิดเครื่อง ไฟล์คอนฟิกกูเรชันจะมีชื่อว่า xorp.conf ปกติจะเก็บไว้ใน /usr/local/xorp/etc/ หรือสามารถอเก็บไว้ในตำแหน่งใดๆ บนเครื่องก็ได้ ไฟล์ดังกล่าวสามารถสร้างขึ้นจาก text editor ตัวใดก็ได้ เช่น notepad และไม่จำเป็นต้องสร้างบนเครื่องเราท์เตอร์ก็ได้ สำหรับคำสั่งที่ใช้โหลดและบันทึกไฟล์คอนฟิกกูเรชันคือ load และ save

- **save** ***<filename>*** บันทึกคอนฟิกกูเรชันปัจจุบันไปเก็บยังไฟล์ที่กำหนด
- **load** ***<filename>*** โหลดไฟล์คอนฟิกกูเรชัน ทำงานแทนคอนฟิกกูเรชันเดิม

```
user@hostname> configure
Entering configuration mode.
There are no other users in configuration mode.
[edit]
user@hostname# save /usr/home/xorp/router1.conf
Save done.
user@hostname# load /usr/home/xorp/router1.conf
[edit]
Load dene.
[edit]
```

- คำสั่ง set

คำสั่ง set เป็นคำสั่งที่ใช้ เพิ่ม เปลี่ยนแปลงหรือแก้ไขค่าคอนฟิก คำสั่งดังกล่าวจะยังไม่มีผลต่อเราท์เตอร์ในทันที จะมีผลก็ต่อเมื่อใช้คำสั่ง commit รูปแบบของคำสั่งคือ

- **set** ***<path to config> <value>***

จากตัวอย่าง คอนฟิกกูเรชันเดิมมีค่า subnet mask เป็น /25 เมื่อแก้ไขคอนฟิกค่า subnet mask บนอินเทอร์เฟสเสมือน vif rl0 เป็น /24 ให้กับไอพี 192.151.188.109 ผล**ปรากฏ**ว่ามีการแก้ไขค่าในคอนฟิกกูเรชันจริง แต่จะยังไม่ทำงานทันที สังเกตุได้จากสัญลักษณ์ “>” แสดงว่ามีการแก้ไขหรือเพิ่มเติมคำสั่งเข้าไป แต่ยังไม่มีการ commit

```
user@hostname# edit interfaces interface rl0
[edit interfaces interface rl0]
user@hostname# show                //ก่อนการแก้ไข config
        description: "control interface"
        vif rl0 {
                address 192.151.188.109  {
                prefix-length: 25
                broadcast: 192.151.188.255
              }
            }

[edit interfaces interface rl0]
user@hostname# set vif rl0 address 192.151.188.109 prefix-length 24
OK

[edit interfaces interface rl0]
user@hostname# show
        description: "control interface"
        vif rl0 {
                address 192.151.188.109 {
>               prefix-length: 24          //แก้ไขแต่ยังไม่มีการ commit
                broadcast: 192.151.188.255
              }
          }
```

เมื่อใช้คำสั่ง set มีรูปแบบการใช้งานได้ 2 ประเภทคือ

set *<path to new config node>* : สร้างคอนฟิกกูเรชันใหม่และเสร็จในทันที

set *<path to new config node> {* : สร้างคอนฟิกกูเรชันใหม่ และสามารถปรับแต่งเพิ่มได้ ตัวอย่างเช่น เมื่อต้องการกำหนด IP address ให้กับอินเทอร์เฟส rl0 เป็น 2 ไอพีในอินเทอร์เฟสเดียวกัน สามารถอธิบายการทำงานได้ดังนี้

(1) ขณะอยู่ในอินเทอร์เฟสเสมือน vif le0 ซึ่งเข้ามายัง level นี้ด้วยคำสั่ง **edit interfaces interface rl0 vif rl0**

(2) แสดงคอนฟิกกูเรชันใน level ปัจจุบัน vif rl0 มีหมายเลขไอพีคือ 192.150.187.108/24

(3) กำหนดไอพีแอดเดรสใหม่ให้กับอินเตอร์เฟส vif rl0 ชุดที่ 2 เป็น 10.0.0.1/16 ต้องใส่เครื่องหมาย { ก่อนการกำหนดไอพี

(4) ปิดท้ายด้วยเครื่องหมาย } เมื่อหมดคำสั่งแล้ว

(5) แสดงคำสั่งที่คอนฟิกไปแล้ว

(6) ปรากฏคอนฟิกใหม่ และมีสัญลักษณ์ “>” เพื่อแสดงว่าคำสั่งดังกล่าวยังไม่มีผลเพราะยังไม่มีการ commit (ยืนยันให้เราท์เตอร์ปฏิบัติตามคำสั่ง)

```
[edit interfaces interface rl0 vif rl0]                    (1)
user@hostname# show                                        (2)
        address 192.150.187.108 {
                prefix-length: 24
                broadcast: 192.150.187.255
            }

[edit interfaces interface rl0 vif rl0]
user@hostname# set address 10.0.0.1 {                      (3)
>               prefix-length 16
>               broadcast 10.0.255.255
>           }                                                   (4)
[edit interfaces interface rl0 vif rl0]
user@hostname# show                                        (5)
        address 192.150.187.108 {
                prefix-length: 24
                broadcast: 192.150.187.255
            }
> address 10.0.0.1 {                                            (6)
>       prefix-length: 16
>       broadcast: 10.0.255.255
> }
```

- คำสั่ง delete

เป็นคำสั่งที่ใช้ลบคอพฟิกกูเรชันของเราท์เตอร์ เมื่อใช้คำสั่งลบแล้ว จะยังไม่มีผล(ใช้สัญลักษณ์ “-”) ต่อเราท์เตอร์จนกว่าจะสั่ง commit(สัญลักษณ์ “-” จะหายไป) ดังตัวอย่าง

```
user@hostname# show interfaces interface rl0 vif rl0
        address 192.150.187.108 {
                prefix-length: 24
                broadcast: 192.150.187.255
            }
        address 10.0.0.1 {
                prefix-length: 16
                broadcast: 10.0.255.255
            }
[edit]
user@hostname# delete interfaces interface rl0 vif rl0 address 10.0.0.1
Deleting:
        address 10.0.0.1 {
                prefix-length: 16
                broadcast: 10.0.255.255
            }
OK
[edit]
user@hostname# show interfaces interface rl0 vif rl0
        address 192.150.187.108 {
                prefix-length: 24
                broadcast: 192.150.187.255
            }
-       address 10.0.0.1 {
-               prefix-length: 16
-               broadcast: 10.0.255.255
-           }
```

- คำสั่ง commit

เป็นคำสั่งที่ใช้ยืนยันให้เราท์เตอร์ทำคำสั่งที่ได้สั่งงานไปก่อนหน้านั้น คำสั่งที่สั่งงานไปจะยังไม่ทำงานในทันที โดยมีสัญลักษณ์ “-” อยู่ข้างหน้า เมื่อสั่งคำสั่ง “commit” สัญลักษณ์ “-” จะหายไปและคำสั่งดังกล่าวจะเริ่มทำงานทันที

```
[edit interfaces interface rl0]
user@hostname# commit
OK
```

ขณะที่เราท์เตอร์มีผู้ใช้งานอยู่มากกว่า 1 คน เมื่อคนใดคนหนึ่งใช้คำสั่ง commit จะส่งผลให้เราท์เตอร์แสดงข้อมูล การเปลี่ยนแปลงแก้ไขคอนฟิกกูเรชันให้คนอื่นๆ ในระบบได้ทราบด้วย

```
[edit]
user@hostname#
The configuration had been changed by user suchart
user@hostname#
```

- ไม่ต้องการบันทึกการเปลี่ยนแปลง (exit discard)

เมื่อผู้ใช้งานทำการเพิ่มเติมหรือแก้ไขคอนฟิกกูเรชันแล้ว แต่ภายหลังไม่ต้องการบันทึกหรือ commit คอนฟิกดังกล่าวให้ทำงาน สามารถทำได้โดยใช้คำสั่ง exit

```
[edit]
user@hostname# exit
ERROR: There are uncommitted changes
Use "commit" to commit the changes, or "exit discard" to discard them
user@hostname# exit discard                    //ไม่บันทึกค่าคอนฟิกกูเรชัน
user@hostname>
```

- การปรับแต่งเชลล์ xorpsh

รูปแบบของเชลล์ xorpsh สามารถปรับแต่งได้ เหมือนกับเชลล์ของ UNIX หรือ Linix ในหัวข้อนี้จะอธิบายถึงวิธีการปรับแต่งเชลล์ให้เหมาะสมกับตามความต้องการของผู้ใช้

  - ปรับแต่ง prompt ของของเชลล์ xorpsh

โดยค่า default ของเชลล์ xorpsh รูปแบบ prompt ของโหมด Operation คือ user@hostname> และรูปแบบ prompt ของโหมด Configuration คือ user@hostname# ผู้ใช้งานสามารถแก้ไขรูปแบบของ prompt ได้โดยการเซ็ตตัวแปรของระบบชื่อว่า XORP_PROMPT_PERATIONAL และ XORP_ROMPT_ONFIGURATION หลังคำสั่ง env ดังนี้

ณ ตำแหน่ง Unix prompt

$ **env XORP_PROMPT_OPERATIONAL=”MSU-Operation> ”**

XORP_PROMPT_CONFIGURATION=”MSU-Config# ”

/usr/local/xorp/bin/xorpsh

Welcome to XORP on hostname

MSU-Operation> **configure**

Entering configuration mode.

There are no other users in configuration mode.

[edit]

MSU-Config#

- การทำงานแบบ non-interactive mode

โดยปกติแล้วการสั่งงานเราท์เตอร์ควรจะสั่งให้ทำงานใน active mode คือทำงานผ่านเชลล์ อย่างไรก็ตาม ก็เป็นไปได้ที่บางครั้งมีความจำเป็นต้องสั่งงานเราท์เตอร์ในโหมดของ non-interactive mode (ตัวอย่างเช่นการใช้งานผ่านเชลล์สคริปของ UNIX) สำหรับงานที่ต้องใช้ non-interactive mode นั้นคืองานที่ต้องการให้ทำงานแบบอัตโนมัติเมื่อเราท์เตอร์ทำงาน หรือทำหน้าที่เฉพาะอย่างในเวลาที่ไม่แน่นอน ตัวอย่างของการใช้ non-interactive mode ดังต่อไปนี้

- การสั่งให้ xorpsh ทำงาน โดยผ่านทาง UNIX เชลล์ command ร่วมกับไปป์ เป็นคำสั่งที่แสดงชื่อของระบบปฏิบัติการที่เราท์เตอร์ทำงานอยู่

  $ **echo “show host os” | /usr/local/xorp/bin/xorpsh**

  คำสั่งจะอยู่ในไฟล์ชื่อว่า filename คำสั่ง cat จะอ่านข้อมูลในไฟล์แล้วส่งต่อให้ xorp ทำงานต่อ

  $ **cat filename | /usr/local/xorp/bin/xorpsh**

  คำสั่งจะอยู่ในรูปของไฟล์ เพื่อป้อนให้กับเชลล์ xorpsh ทำงานตรงๆ โดยผ่านทาง command ของ UNIX คือ < (reditection)

  $ **/usr/local/xorp/bin/xorpsh < filename**

- การสั่งให้ xorpsh ทำงาน โดยผ่านทาง Shell Script

  $**!/bin/sh**

  xorpsh <<!

  show host os

  !

- หรือใช้ option “–c”

  $ **/usr/local/xorp/bin/xorpsh –c “show host os”**

- หรือสามารถเขียนโปรแกรมด้วยภาษาที่ต้องการได้

  ภาษา python (ใช้คำสั่ง expect)

```
#!/usr/bin/env python
import time
import sys
import pexpect

child=pexpect.spawn ('/usr/local/xorp/bin/xorpsh')
```

```
child.expect('user@hostname> ')
child.sendline('show host os | no-more')
child.sendeof()
while 1:
        line = child.readline()
        if not line:
                break
        print line,
child.close()
```

13.5.3 Network Interfaces Technology and Concepts

หน้าที่หลักของเราท์เตอร์คือการค้นหาเส้นทางและการส่งข้อมูลต่อ(forward) ไปยังเป้าหมายปลายทางให้สำเร็จ อุปกรณ์หลักที่สำคัญในการส่งและรับข้อมูลคือเน็ตเวิร์คอินเทอร์เฟส (Network Card) ซึ่งปัจจุบันมีหลายชนิดเช่น Ethernet, ATM, DS3 และ ISDN เป็นต้น บนเราท์เตอร์ XORP จะเรียกอินเทอร์เฟสที่เชื่อมต่อว่า interfaces เช่น lr0, fxp0, eth0 และเรียกอินเทอร์เฟสเสมือนว่า vifs (vifs เป็นการจำลองอินเทอร์เฟสที่มีอยู่จริงให้มีได้หลายๆ อินเทอร์เฟส โดยใช้ซอฟต์แวร์ในการจำลอง) ซึ่งอินเทอร์เฟสเสมือนจะใช้ในกรณีที่ต้องการทำ VLAN ในแนวความคิดการออกแบบของ XORP ทุกๆ อินเทอร์เฟสควรจะต้องมีอินเทอร์เฟสเสมือนเสมอและควรจะต้องมีหมายเลขไอพีประจำแต่ละ vif ด้วย จะมีบางอินเทอร์เฟสที่จะต้องมี vif เพียงอันเดียวซึ่งอินเทอร์เฟสดังกล่าวจะใช้เป็น default vif เพื่อทำการส่งต่อข้อมูลไปยังอินเทอร์เฟสอื่นๆ

- การคอนฟิกอินเทอร์เฟส

ชื่อของอินเทอร์เฟสบนเราท์เตอร์จะขึ้นอยู่กับระบบปฏิบัติการที่ติดตั้ง เมื่อใช้ FreeBSD มักจะใช้ชื่อเป็น fxp หรือ lro แต่ช้าติดตั้งบนลินุกซ์จะใช้ชื่อเป็น eth เป็นต้น โพรโทคอล routing บางตัวก็จำเป็นต้องอ้างถึงอินเทอร์เฟสโดยตรง เช่น RIP แต่บางโพรโทคอลก็ไม่จำเป็นเช่น BGP

1. Configuration Syntax

รูปแบบของ syntax ในการคอนฟิกอินเทอร์เฟสมีดังนี้

```
interfaces {
        restore-original-config-on-shutdown: bool
        interface text {
                description: text
        mac: macaddr
        mtu: uint
        default-system-config
        disable: bool
        discard: bool
        unreachable: bool
        management: bool
        vif text {
                disable: bool
```

```
                vlan {
                        vlan-id: int(0..4095)
                }
                address IPv4-addr {
                        prefix-length: int(1..32)
                        broadcast: IPv4-addr
                        destination: IPv4-addr
                        disable: bool
                }
                address IPv6-addr {
                        prefix-length: int(1..128)
                        destination: IPv6-addr
                        disable: bool
                }
        }
    }
}
```

interfaces : เป็นส่วนที่แสดงขอบเขตคอนฟิกกูเรชันของอินเทอร์เฟสทั้งหมดบนเราท์เตอร์ หรือพูดง่ายๆ คือ เป็นส่วนที่ใช้กำหนดคุณสมบัติของอินเทอร์เฟสที่ต้องการใช้งานทั้งหมด

restore-original-config-on-shutdown: เป็น flag ที่อนุญาตให้ทำการ restoring คอนฟิกกูเรชันเดิมของอินเทอร์เฟสเมื่อ FEA(อ่านข้อมูลเกี่ยวกับ FEA ได้ในหัวข้อถัดไป) ไม่ทำงาน ค่าเริ่มต้นจะเป็น false คือไม่สั่งให้ทำงาน

interface: เป็นส่วนที่ใช้กำหนดคอนฟิกกูเรชันของแต่ละอินเทอร์เฟสที่มีการติดตั้งจริงบนเราท์เตอร์ เช่น fxp, rl0 เป็นต้น

description: เป็นส่วนที่ใช้อธิบายความหมายของแต่ละอินเทอร์เฟส เพื่อช่วยให้ผู้ดูแลระบบสามารถทำความเข้าใจได้ง่ายขึ้น

mac: เป็นส่วนที่ใช้สำหรับกำหนดหมายเลขของ MAC โดยปกติ MAC จะถูกกำหนดมาแบบตายตัวจากโรงงานอยู่แล้ว แต่ในบางกรณีก็มีความจำเป็นที่จะต้องมีการปรับเปลี่ยน MAC ในบางครั้ง MAC จะมีขนาด 48 ไบต์ แบ่งออกเป็น 6 ชุด เช่น 00:0a:59:9a:f2:ba

mtu: เป็นส่วนที่ให้ผู้ใช้งานสามารถกำหนดขนาดของแพ็กเก็ตที่ต้องการโอนย้ายได้ ค่าที่ใช้กำหนดจะเป็นเลขจำนวนเต็ม ขนาดที่สามารถกำหนดได้ไม่ควรมากว่าความสามารถของเน็ตเวิร์คการ์ดที่สามารถส่งข้อมูลได้ มิฉะนั้นข้อมูลที่ส่งจะเสียหาย แต่ถ้าต้องการส่งแพ็กเก็ตในปริมาณที่เกินความสามารถของฮาร์ดแวร์ก็ทำได้โดยการกำหนดบิต DF

default-system-config: ปกติทุกๆ อินเทอร์เฟส, vifs และไอพีแอดเดรสจะถูกกำหนดโดยผ่านทาง CLI แต่ในบางกรณีเราท์เตอร์มีความจำเป็นต้องอ่านคอนฟิกกูเรชันใดคอนฟิกกูเรชันหนึ่งขึ้นมาทำงานในเบื้องต้นก่อน จากนั้นเมื่อเราท์เตอร์สามารถทำงานได้แล้ว จึงจะทำการปรับแต่งคอนฟิกในแต่ละส่วนต่อไป ค่าในส่วนของ default-system-config ก็จะทำหน้าที่ตามที่ได้กล่าวมาแล้วข้างต้น

disable: เป็นส่วนที่ใช้กำหนดว่าต้องการให้อินเทอร์เฟสทำงานหรือไม่ โดยค่าเริ่มต้นจะเป็น true (หมายถึงปิดการใช้งาน)

discard: เป็นส่วนที่ให้ผู้ใช้สามารถกำหนดได้ว่าอินเทอร์ใดที่ไม่สนใจ แม้ว่าอินเทอร์เฟสดังกล่าวจะมีการติดตั้งหรือใช้งานจริงอยู่ ค่าเริ่มต้นเป็น false

unreachable: เป็นส่วนที่ผู้ใช้สามารถกำหนดได้ว่าให้อินเทอร์เฟสดังกล่าวมีการตอบรับแบบ unreachable เช่นเมื่อทำการ ping โดยใช้โพรโทคอล ICMP เราท์เตอร์จะตอบกลับไปยังต้นทางเป็น destination unreachable ค่าเริ่มต้นกำหนดเป็น false (คือตอบรับแบบปกติ โดยไม่ส่ง destination unreachable ไปให้)

management: เป็นส่วนที่กำหนดให้อินเทอร์เฟสดังกล่าวสามารถตรวจสอบการทำงานได้ ค่าเริ่มต้นคือ false (ไม่อนุญาตให้ทำการตรวจสอบ)

vif: เป็นส่วนที่ใช้สำหรับสร้างอินเทอร์เฟสเสมือน เมื่อต้องการใช้อินเทอร์เฟสเสมือนทำงานหรือมีการ forward ข้อมูลให้กำหนด flag disable เป็น false

vlan: เป็นส่วนย่อยที่อยู่ภายใต้อินเทอร์เฟสเสมือนใช้สำหรับสร้าง VLAN โดย VLAN แต่ละ VLAN จำมีชื่อที่ไม่ซ้ำกัน ปัจจุบันจะสนับสนุนเฉพาะ 802.1Q

address: เป็นส่วนที่ใช้กำหนดหมายเลขไอพีในอินเทอร์เฟสเสมือน ในแต่ละ vif สามารถมีได้มากกว่า 1 ไอพีได้ และจะประกอบไปด้วยคอนฟิกกูเรชันของ IPV4 และ IPV6 ด้วย

prefix-length: เป็นส่วนย่อยที่อยู่ภายใต้ address เพื่อใช้กำหนดการทำ subnet mask ให้กับเครือข่าย ใน IPV4 มีค่าตั้งแต่ 4-32 และ IPV6 มีค่าตั้งแต่ 8-128 บิต

broadcast: ใช้สำหรับกำหนดหมายเลขไอพี broadcast

destination: ใช้กำหนดไอพีปลายทาง สำหรับใช้ในกรณีเชื่อมต่อแบบ point-to-point

disable: เป็นส่วนย่อยที่อยู่ภายใต้ vif ใช้สำหรับระบุว่า ให้ vif ดังกล่าวสามารถทำงานได้หรือไม่ (false = vif ทำงาน, true= vif ไม่ทำงาน)

- ตัวอย่างการคอนฟิกอินเทอร์เฟส

Cofiguring Interface Addresses ก่อนการคอนฟิกให้ตรวจสอบก่อนว่าเราท์เตอร์ที่เราทำการติดตั้งการ์ดเน็ตเวิร์คกี่ใบ ชื่ออะไร และมีจำนวนเท่าไดก่อน โดยเข้าไปที่ UNIX เชลล์ แล้วใช้คำสั่ง ifcofig -a

```
$ ifconfig -a
rl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:06:4f:07:42:60
        inet 10.114.251.90 netmask 0xffff0000 broadcast 10.114.255.255
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
nfe0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:13:8f:9f:5c:fc
        media: Ethernet autoselect (none)
        status: no carrier
plip0: flags=108810<POINTOPOINT,SIMPLEX,MULTICAST,NEEDSGIANT> metric 0 mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
```

```
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
```

ผลจากการใช้คำสั่ง ifconfig ด้านบนแสดงว่ามีอินเฟส 3 ใบคือ rl0, nfe0 และ lo0 (loopback interface) เลือกอินเทอร์เฟสที่ต้องการคอนฟิก(อย่าลืมต่อสายนำสัญญาณเข้ากับอินเทอร์เฟสที่ต้องการคอนฟิกด้วย) และเข้าไปคอนฟิกที่เราท์เตอร์ ดังนี้

```
Welcome to XORP on Rx1.msu.ac.th
xorp@Rx1.msu.ac.th>                          //โหมด Operation
xorp@Rx1.msu.ac.th> configure                //เข้าสู่โหมด configuration
Entering configuration mode.
There are no other users in configuration mode.
[edit]
xorp@Rx1.msu.ac.th#                          //โหมด Configuration
xorp@Rx1.msu.ac.th# set interfaces interface rl0 description Ethernet_interface
                    //กำหนดคำอธิบายให้อินเทอร์เฟส
[edit]
xorp@Rx1.msu.ac.th# set interfaces interface rl0 disable false          //สั่งให้ทำงาน
[edit]
xorp@Rx1.msu.ac.th# set interfaces interface rl0 vif rl0 //สร้าง vif บนอินเทอร์เฟส rl0
[edit]
xorp@Rx1.msu.ac.th# set interfaces interface rl0 vif rl0 disable false //สั่งให้ vif ทำงาน
[edit]
xorp@Rx1.msu.ac.th# set interfaces interface rl0 vif rl0 address 10.114.251.99 prefix-length
16  //กำหนดหมายเลขไอพีและซับเน็ต
[edit]
xorp@Rx1.msu.ac.th# set interfaces interface rl0 vif rl0 address 10.114.251.99 broadcast
10.114.255.255  //กำหนดไอพี broadcast
[edit]
xorp@Rx1.msu.ac.th# set interfaces interface rl0 vif rl0 address 10.114.251.99 disable false
          //สั่งให้ไอพีแอดเดรสดังกล่าวทำงาน
[edit]
xorp@Rx1.msu.ac.th# set interfaces interface rl0 vif rl0 address
2001:DB8:10:10:10:10:10:10 prefix-length 64  //กำหนดไอพีแอดเดรสของ IPv6
[edit]
xorp@Rx1.msu.ac.th# set interfaces interface rl0 vif rl0 address 2001:DB8:10:10:10:10:10:10
disable false     //สั่งให้ไอพี V6 ทำงาน
[edit]
xorp@Rx1.msu.ac.th# commit                   //ยืนยันการคอนฟิกที่ผ่านมาแล้วว่าถูกต้อง
OK
[edit]
xorp@Rx1.msu.ac.th#
```

แสดงคอนฟิกที่กำหนดไปแล้วโดยใช้คำสั่ง show interfaces interface rl0

```
xorp@Rx1.msu.ac.th# show interfaces interface rl0
   description: "Ethernet_interface"
   vif rl0 {
      address 10.114.251.99 {
         prefix-length: 16
         broadcast: 10.114.255.255
      }
      address 2001:DB8:10:10:10:10:10:10 {
         prefix-length: 64
      }
   }

[edit]
xorp@Rx1.msu.ac.th# save /home/xorp/xorp.conf          //บันทึกข้อมูล
Save done.
```

การบันทึกข้อมูลจำเป็นต้องให้สิทธิ์ในการเขียนไฟล์กับ user ที่สั่งรัน xorpsh เชลล์ทำงานด้วย (ในตัวอย่างนี้ได้สร้าง user=xorp, group=xorp) ไม่เช่นนั้นจะไม่สามารถเขียนข้อมูลลงดิกส์ได้ วิธีที่ง่ายที่สุดคือเขียนในตำแหน่ง Home directory ของผู้ที่รัน xorpsh เชลล์

Using Pre-Configured Interface Addresses เป็นการคอนฟิก default-system-config สำหรับใช้ในกรณีที่ต้องการเริ่มต้นการทำงานบางอย่างที่จะต้องใช้คอนฟิกร่วมกันกับอินเทอร์เฟสทั้งหมด เมื่อคอนฟิกที่ default-system-config จะต้องไม่มีการสร้าง vif และกำหนดไอพี

```
interfaces {
        interface rl0 {
                description: "data interface"
                disable: false
                default-system-config
                }
}
```

Configuring VLANs เราสามารถทำการสร้าง VLAN ภายใต้ vif ได้ แต่อินเทอร์เฟสดังกล่าวต้องสนับสนุนการทำ VLAN ด้วย

```
interfaces {
        interface nfe0 {
                description: "Ethernet interface with a VLAN"
                vif nfe0 {
                        address 10.10.10.10 {
                        prefix-length: 24
                        }
                }
        vif vlan1 {
                vlan {
                        vlan-id: 1
                    }
                address 10.10.20.20 {
                        prefix-length: 24
                }
            }
    }
}
```

จากตัวอย่างข้างบน ทำการสร้าง VLAN ภายใต้อินเทอร์เฟส nfe0 ชื่อว่า vlan1 และมีหมายเลข id = 1 มีหมายเลขไอพีของ vlan1 คือ 10.10.20.20/24

2. การตรวจสอบหรือแสดงผลการทำงานของอินเทอร์เฟส

การแสดงผลการทำงานของอินเทอร์เฟสจะใช้คำสั่ง show interfaces จะแสดงรายละเอียดของทุกๆ อินเทอร์เฟสบนเราท์เตอร์ แต่ถ้าต้องการแสดงผลในบางอินเทอร์เฟสจะใช้คำสั่ง show interfaces interface <อินเทอร์เฟสที่ต้องการแสดง> เช่น

```
user@hostname> show interfaces
dc0/dc0: Flags:<ENABLED,BROADCAST,MULTICAST> mtu 1500
```

```
        inet 172.13.0.1 subnet 172.13.0.0/24 broadcast 172.13.0.255 physical index 1
        ether 00:80:c8:b9:61:09
dc1/dc1: Flags:<ENABLED,BROADCAST,MULTICAST> mtu 1500
        inet 172.13.1.1 subnet 172.13.1.0/24 broadcast 172.13.0.255 physical index 2
        ether 00:80:c8:b9:61:0a
dc2/dc2: Flags:<ENABLED,BROADCAST,MULTICAST> mtu 1500
        inet 172.13.2.1 subnet 172.13.2.0/24 broadcast 172.13.0.255 physical index 3
        ether 00:80:c8:b9:61:0b
dc3/dc3: Flags:<ENABLED,BROADCAST,MULTICAST> mtu 1500
        inet 172.13.3.1 subnet 172.13.3.0/24 broadcast 172.13.0.255 physical index 4
        ether 00:80:c8:b9:61:0c
fxp0/fxp0: Flags:<ENABLED,BROADCAST,MULTICAST> mtu 1500
        inet 192.150.187.112 subnet 192.150.187.0/25 broadcast 192.150.187.255
        physical index 5
        ether 00:02:b3:10:b4:6c
user@hostname> show interfaces dc1
        dc1/dc1: Flags:<ENABLED,BROADCAST,MULTICAST> mtu 1500
        inet 172.13.1.1 subnet 172.13.1.0/24 broadcast 172.13.0.255 physical index 2
        ether 00:80:c8:b9:61:0a
```

- **Firewall**

โดยปกติเราท์เตอร์มีความสามารถในการคัดกรองหรือจำกัดชนิดของแพ็กเก็ตบนเครือข่ายได้อยู่แล้ว โดยการ ตั้งกฎขึ้นมาเรียกว่า ไฟล์วอลล์รูล (firewall rule) ในกฎจะบอกให้เราท์เตอร์กระทำต่อแพ็กเก็ตอย่างไร โดยมีรูปแบบดังนี้

Rule ID	Source IP	Destination IP	Protocol	action
1	10.10.10.0/24	20.20.0.0/16	TCP	Pass
2	10.10.10.0/24	30.30.30./24	UDP	reject
…	…	…	…	…
n	Any	Any	Any	Drop

กฎส่วนมากจะประกอบไปด้วย 5 ส่วนคือ Rule ID คือหมายเลขของกฎที่ตั้งขึ้น ต้องไม่ซ้ำกัน นิยมใช้เป็นตัวเลขจำนวนเต็ม ไฟล์วอลล์จะทำงานจากกฎที่ 1 เสมอ ถ้าแพ็กเก็ตที่เข้ามาไม่ตรงกับกฎที่กำหนดไว้ก็จะเลื่อนไปเรื่อยๆ ตามลำดับ จนเมื่อถึงกฎสุดท้ายก็จะถูกโยนทิ้งไป (drop) โดยอัตโนมัติ source ip คือหมายเลขไอพีต้นทางที่ต้องการส่งข้อมูลไปยังหมายเลขไอพีปลายทาง destination ip หรืออาจจะเป็นกลุ่มของไอพีก็ได้ protocol หมายถึงชนิดของโพรโทคอลที่ใช้ในการติดต่อสื่อสาร ปัจจุบันเครือข่าย TCP/IP ได้รับความนิยมสูง จึงนิยมใช้โพรโทคอลประเภท TCP, UDP และส่วนสุดท้ายคือ action ทำหน้าที่ตัดสินว่าแพ็กเก็ตที่เข้ามายังกฎดังกล่าวจะให้ทำอย่างไร โดยแบ่งออกเป็นประเภทหลักๆ คือ pass= ยอมให้ผ่านไปได้ reject=โยนแพ็กเก็ตทิ้งแต่ส่งข้อความไปบอกด้วยว่าได้ทำการโยนทิ้ง drop=โยนแพ็กเก็ตทิ้งและไม่มีการแจ้งเตือน XORP สนับสนุนการทำไฟล์วอลล์บนเราท์เตอร์ (cisco เรียกว่า ACL) แต่การทำไฟล์วอลล์ไม่ควรสร้างกฎเยอะเกินไป จะทำให้เราท์เตอร์ด้อยประสิทธิภาพลง

1. Firewall Configuration Syntax ตัวอย่างด้านล่างเป็น syntax ของไฟล์วอลล์

```
firewall {
```

```
        rule4 int(1..65534) {
                action: text
                protocol: int(0..255)
                source {
                        interface: text
                        vif: text
                        network: IPv4/int(0..32)
                        port-begin: int(0..65535)
                        port-end: int(0..65535)
                }
                destination {
                        network: IPv4/int(0..32)
                        port-begin: int(0..65535)
                        port-end: int(0..65535)
                }
        }
        rule6 int(1..65534) {
                action: text
                protocol: int(0..255)
                source {
                        interface: text
                        vif: text
                        network: IPv6/int(0..128)
                        port-begin: int(0..65535)
                        port-end: int(0..65535)
                }
                destination {
                        network: IPv6/int(0..128)
                        port-begin: int(0..65535)
                        port-end: int(0..65535)
                }
        }
}
```

firewall: เป็นส่วนที่ใช้กำหนดขอบเขตของไฟล์วอลล์คอนฟิกกูเรชันบนเราท์เตอร์

rule4 int: เป็นส่วนที่ใช้กำหนดกฎของไฟล์วอลล์ด้วย IPv4 สำหรับ key word int เป็นหมายเลขของกฎที่ต้องไม่ซ้ำกัน สามารถใช้ได้ตั้งแต่ 1-65534

action: ทำหน้าที่ตัดสินว่าแพ็กเก็ตที่เข้ามาจะให้ทำอย่างไร โดยแบ่งออกเป็น 4 ประเภทดังนี้คือ

- ***none:*** ไม่ทำอะไรทั้งสิ้น การตรวจสอบจะเลื่อนไปยังกฎต่อไป
- ***pass:*** ผ่านได้
- ***drop:*** ไม่ยอมให้ผ่าน
- ***reject:*** ไม่ยอมให้ผ่านแต่มี message แจ้งเตือนกับไปยังต้นทางที่ส่งข้อมูลมา

protocol: เป็นส่วนที่ใช้ระบุถึงโพรโทคอลที่ต้องการตรวจสอบ เป็นตัวเลขจำนวนเต็ม [65] ตั้งแต่ 0-255 ตัวอย่างเช่น TCP มีหมายเลขเป็น 6 และ UDP มีหมายเลขเป็น 17 เป็นต้น เมื่อค่าที่กำหนดเป็น 0 จะหมายถึงใช้งานได้ทุกๆ โพรโทคอล

source: เป็นส่วนที่ใช้ระบุขอบเขตของหมายเลขไอพีต้นทาง ประกอบด้วยส่วนย่อยๆ ดังนี้

- ***interface:*** เป็นชื่อของอินเทอร์เฟสที่รับแพ็กเก็ตจากต้นทางเข้าสู่เราท์เตอร์
- ***vif:*** เป็นชื่อของอินเทอร์เฟสเสมือนที่รับแพ็กเก็ตจากต้นทางเข้าสู่เราท์เตอร์
- ***network:*** เป็นส่วนที่ใช้ระบุขอบเขตของไอพีต้นทาง มีรูปแบบคือ IP Address/Prefix-length เช่น 10.10.10.0/24 ค่าเริ่มต้นคือ 0.0.0.0/0 หมายถึงทุกๆ ไอพีแอดเดรส
- ***port-begin:*** ระบุจุดเริ่มต้นของหมายเลขพอร์ต มีค่าระหว่าง 0-65535 ค่าเริ่มต้นคือ 0
- ***port-end:*** ระบุจุดสิ้นสุดของหมายเลขพอร์ต มีค่าระหว่าง 0-65535 ค่าเริ่มต้นคือ 0

destination: เป็นส่วนที่ใช้ระบุขอบเขตของหมายเลขไอพีปลายทาง ประกอบด้วยส่วนย่อยๆ ดังนี้

- ***network, port-begin, port-end:*** เหมือนกันกับ source

rule6: เป็นส่วนที่ใช้กำหนดกฎของไฟล์วอลล์ด้วย IPv6 สำหรับ key word int เป็นหมายเลขของกฎที่ต้องไม่ซ้ำกัน สามารถใช้ได้ตั้งแต่ 1-65534

สำหรับความหมายอื่นๆ เช่น protocol, source, vif จะเหมือนกับ IPv4

แสดงตัวอย่างของคอนฟิกกูเรชันของไฟล์วอลล์

```
firewall {
        rule4 100 {
                action: "pass"
                protocol: 6 /* TCP */
                destination {
                        network: 10.10.10.10/32
                        port-begin: 80
                        port-end: 80
                        }
                }
        rule4 200 {
                action: "drop"
                protocol: 6 /* TCP */
                source {
                        interface: "fxp0"
```

```
                        vif: "fxp0"
                        network: 0.0.0.0/0
                        port-begin: 0
                        port-end: 65535
                        }
                destination {
                        network: 10.10.0.0/24
                        port-begin: 0
                        port-end: 1024
                        }
                }
        rule4 65000 {
                action: "pass"
                protocol: 6 /* TCP */
                }
        }
```

จากตัวอย่างข้างบนเราท์เตอร์สร้างไฟล์วอลล์ไว้ทั้งหมด 3 กฎ กฎแรกหมายเลข 100(IPv4) จะอนุญาตให้เครื่องต้นทางเป็นไอพีอะไรก็ได้ ที่ใช้ไพรโทคอล TCP ผ่านไปยังเครื่องไอพีปลายทางเพียงหมายเลขเดียวเท่านั้นคือไอพี 10.10.10.10 โดยใช้บริการเว็บเซิร์ฟเวอร์เท่านั้น กฎที่ 2(200) จะทำการปิดกัน TCP ทุกๆ แพ็กเก็ตที่เข้ามายังอินเทอร์เฟส fxp0 ไปยังเน็ตเวิร์คปลายทางคือ 10.10.0.0/24 ชนิด TCP โดยมีช่วงของพอร์ตอยู่ระหว่าง 0-1024 กฎสุดท้าย(65000) จะอนุญาตให้แพ็กแก็ตทั้งหมดผ่านได้

สำหรับการตรวจสอบการทำงานของไฟล์วอลล์ xorp ยังไม่สนับสนุน แต่สามารถตรวจสอบการทำงานได้โดยใช้คำสั่งในเชลล์ของ UNIX สั่งงานผ่าน ipfw(สำหรับ FreeBSD) และ iptables(สำหรับ Linux)

- **Forwarding Engine**

Forward engine เป็นส่วนประกอบที่สำคัญในเราท์เตอร์เนื่องจากมันทำหน้าที่ส่งข้อมูลและรับข้อมูลจากอินเทอร์เฟสหนึ่งไปยังอีกอินเทอร์เฟสหนึ่งบนเราท์เตอร์ XORP จะใช้คำว่า fea เพื่อแทน Forward Engine Abstrection และใช้ mfea แทน Multicast Forwarding Engine Abstraction และคำว่า abstrection เพื่อบอกว่าเป็นการคอนฟิกเราท์เตอร์ในระดับ hi-level ซึ่งคอนฟิกระดับนี้จะสั่งให้มีการ forward ข้อมูลในระดับล่าง คือระดับ kernel ของระบบปฏิบัติการต่อไป

1. Configguration of the Forwarding Engine

โดยปกติ XORP เราท์เตอร์จะต้องคอนฟิกให้ fea ทำงานโดย default เนื่องจากเราท์เตอร์ต้องทำหน้าที่ forward ข้อมูลเป็นปกติอยู่แล้ว การคอนฟิก fea จะมีการแยกกันระหว่าง unicast และ multicast แยกระหว่าง IPv4 และ IPv6 ด้วย ดังตัวอย่างข้างล่าง

Configuration Syntax

```
fea {
        targetname: txt
        unicast-forwarding4 {
                disable: bool
                table-id: u32
```

```
                forwarding-entries {
                        retain-on-startup: bool
                        retain-on-shutdown: bool
                        }
                }
        unicast-forwarding6 {
                disable: bool
                table-id: u32
                forwarding-entries {
                        retain-on-startup: bool
                        retain-on-shutdown: bool
}
}
        click {
                disable: bool
                duplicate-routes-to-kernel: bool

                kernel-click {
                        disable: bool
                        install-on-startup: bool
                        kernel-click-modules: text
                        mount-directory: text
                        kernel-click-config-generator-file: text
                }

                user-click {
                        disable: bool
                        command-file: text
                        command-extra-arguments: text
                        command-execute-on-startup: bool
                        control-address: IPv4-addr
                        control-socket-port: uint(1..65535)
                        startup-config-file: text
                        user-click-config-generator-file: text
                        }
                  }
        }
```

continued overleaf....

```
plumbing {
        mfea4 {
                disable: bool
                interface text {
                        vif text {
                                disable: bool
                        }
                }
                interface register_vif {
                        vif register_vif {
                                disable: bool
                        }
                }
                traceoptions {
                        flag all {
                                disable: bool
                        }
                }
        }
        mfea6 {
                disable: bool
                interface text {
                        vif text {
                                disable: bool
```

```
                        }
                }
        interface register_vif {
                vif register_vif {
                        disable: bool
                        }
                }
        traceoptions {
                flag {
                        all {
                                disable: bool
                        }
                    }
            }
        }
}
```

fea: เป็นส่วนที่ใช้กำหนดให้เราท์เตอร์สามารถ forward ข้อมูล สำหรับโพโทคอลมัลติคาสท์

targetname: เป็นส่วนที่ใช้กำหนดชื่อของ fea ชื่อ default คือ fea

unicast-forwarding4: เป็นส่วนที่กำหนดว่าเป็นการ forward แบบ unicast ของ IPv4

disable: เป็นค่าที่ใช้กำหนดให้ fea ทำงานหรือไม่ ค่า default คือ false (fea ทำงาน)

table-id: ใช้ระบุถึง Table ID ของunicast forwarding เมื่อไม่กำหนด จะใช้ Table ID ของระบบแทน

forwarding-entries: กำหนดคุณสมบัติของ IPv4 forwarding entries

retain-on-startup: ควบคุม unicast forwarding entries ขณะเริ่มต้นทำงาน ค่า default คือ false

retain-on-shutdown: ควบคุม unicast forwarding entries ขณะสิ้นสุดการทำงาน ค่า default คือ false

unicast-forwarding6: ใช้สำหรับคอนฟิก IPv6 forward ค่าอื่นๆ จะเหมือนกับ IPv4

click: คำสั่งที่ใช้สำหรับสั่งให้ Click ทำการ forward ข้อมูล

disable: สั่งให้ Click ทำงานหรือหยุดทำงาน ค่า default คือ false

duplicate-routes-to-kernel: ควบคุมให้ click ทำการเพิ่มเส้นทางไปยัง kernel ด้วย ค่าเริ่มต้นคือ false

kernel-click: เป็นคำสั่งที่ใช้ควบคุม click ให้สามารถทำงานร่วมกับ kenel ของระบบฏิบัติการ

install-on-startup: เป็นคำสั่งที่ใช้กำหนดว่าควรจะทำการติดตั้ง kernel-click ขณะเริ่มต้นทำงานหรือไม่ ค่าเริ่มต้นคือ false

kernel-click-modules: เป็นส่วนที่ใช้ระบุว่าจะให้โหลดโมดูลของ click ตัวใดมาใช้งานบ้าง เมื่อใช้ xorp กับลินุกซ์จำเป็นต้องโหลดโมดูล “/us-r/local/click/linuxmodule/proclikefs.o:/usr/local/click/

linuxmodule/click.o” แต่ละโมดูลแยกกันโดยใช้สัญลักษณ์ “:” สำหรับ FreeBSD จะใช้โมดูล “/path/to/click.ko”

mount-directory: เป็นไดเรคทรอรี่ที่ click ติดตั้งอยู่

kernel-click-config-generator-file: เป็นส่วนที่ใช้ระบุถึงตัวโปรแกรมที่สำหรับใช้สร้าง click คอนฟิกกูเรชันไฟล์ โดยปกติจะอยู่ที่ “/usr/local/xorp/fea/xorp fea click config generator”

user-click: ใช้สำหรับกำหนด user-level click

command-file: ใช้กำหนดตำแหน่งที่อยู่ของโปรแกรมที่จะประมวลผล user-level click ปกติจะเก็บอยู่ที่ “/usr/local/bin/click”

command-extra-arguments: เป็นส่วนที่ใช้กำหนดคุณสมบัติเพิ่มเติมใน user-level click

command-execute-on-startup: เป็นส่วนที่กำหนดว่าจะใช้โปรแกรมใดที่จะให้ user-level click ทำงานในตอนเริ่มต้นระบบ

control-address: เป็นส่วนที่ใช้กำหนด address สำหรับเฝ้ารอการเชื่อมต่อจากคอนเนคชันภายนอก ค่าเริ่มต้นจะมีตำแหน่ง address เป็น 127.0.0.1

control-socket-port: กำหนดหมายเลขพอร์ตของ TCP เพื่อใช้สำหรับเฝ้ารอการเชื่อมต่อ พอร์ตที่ใช้จะเริ่มตั้งแต่ 1-65535 ค่าเริ่มต้นคือ พอร์ต 13000

startup-config-file: เป็นชื่อของคอนฟิกกูเรชันไฟล์ของ click ใช้ในตอนเริ่มต้นการทำงาน เริ่มต้นจะชี้ไปที่ “/dev/null”

plumbing: เป็นส่วนที่ใช้กำหนดคุณสมบัติของ plumbing ในมัลติคาสท์แบบ IPv4

disable: กำหนดให้ forward ข้อมูลมัลติคาสท์หรือไม่ ค่าเริ่มต้นคือ false

interface: ระบุอินเทอร์เฟสที่จะใช้สำหรับ forward ข้อมูลของมัลติคาสท์

vif: ใช้สำหรับกำหนดมัลติคาสท์ forward บนอินเทอร์เฟสเสมือน

traceoptions: ใช้สำหรับกำหนดให้มีการ debug และ trace สำหรับการสื่อสารแบบมัลติคาสท์

flag: กำหนดให้ trace ทำงาน

mfea6: เป็นส่วนที่ใช้กำหนดการ forward ข้อมูลมัลติคาสท์ใน IPv6 ค่าอื่นๆ จะเหมือนกับ IPv4

ตัวอย่างไฟล์คอนฟิกของ Forwarding Engine

```
fea {
        unicast-forwarding4 {
                disable: false
        }
        unicast-forwarding6 {
                disable: true
        }
}
plumbing {
        mfea4 {
                disable: false
                interface dc0 {
                        vif dc0 {
```

```
                                        disable: false
}
}
                    interface register vif {
                              vif register vif {
                              /* Note: this vif should be always enabled */
                              disable: false
                              }
                    }
                    traceoptions {
                              flag all {
                                        disable: false
                                        }
                              }
          }
mfea6 {
          disable: false
          interface dc0 {
                    vif dc0 {
                              disable: false
                              }
                    }
          interface register vif {
                    vif register vif {
                    /* Note: this vif should be always enabled */
                    disable: false
                    }
               }
       }
}
```

ตัวอย่างคอนฟิกด้านบน อนุญาตให้ IPv4 สามารถ forward ข้อมูลได้ แต่ไม่อนุญาตให้ IPv6 forward ข้อมูล ในส่วนของ plumbing คืออนุญาตให้ทำการ forward ข้อมูลของมัลติคาสท์โพรโทคอล IPv4 และ IPv6 ได้ บนอินเทอร์เฟส interface/vif = dc0/dc0 และเปิดให้ vif สามารถใช้โพรโทคอล routing ของมัลติคาสท์คือ PIM-SM ทำงาน

```
interfaces {
          interface eth0 {
                    description: "control interface"
                    vif eth0 {
                              address 10.10.10.10 {
                              prefix-length: 24
                              broadcast: 10.10.10.255
                              }
                    }
                    mac: aa:bb:cc:dd:ee:ff
                    mtu: 1500
                    }
          }
fea {
          unicast-forwarding4 {
                    disable: false
                    }
click {
          disable: false
          duplicate-routes-to-kernel: false

          kernel-click {
                    disable: true
                    install-on-startup: true
```

```
                    kernel-click-modules: "/path/to/proclikefs.o:/path/to/click.o";
                    mount-directory: "/click"
                    kernel-click-config-generator-file: "/path/to/kernel click config generator"
}

          user-click {
                    disable: false
                    command-file: "/path/to/click"
                    command-extra-arguments: "-R"
                    command-execute-on-startup: true
                    control-address: 127.0.0.1
                    control-socket-port: 13000
                    startup-config-file: "/dev/null"
                    user-click-config-generator-file: "/path/to user click config generator"
                    }
          }
}
```

สำหรับคอนฟิกกูเรชันด้านบนจะอนุญาตให้ kernel-level และ user-level click ทำงาน

2. การตรวจสอบการทำงานของ Forwarding Engine

การตรวจสอบการทำงานของ fea สามารถทำได้โดยใช้คำสั่ง show mfea dataflow, show mfea interface, show mfea address สำหรับ IPv4 และ show mfea6 dataflow, show mfea6 interface, show mfea6 address สำหรับ IPv6 ตัวอย่างเช่น

```
user@hostname> show mfea dataflow
Group                                   Source
224.0.1.20 10.2.0.1
          Measured(Start|Packets|Bytes)   Type   Thresh(Interval|Packets|Bytes)   Remain
          1091667269.982158|0|?           <=     210.0|0|?                        202.434319
          1091667269.984406|?|0           >=     100.0|?|102400                   92.436567
```

```
user@hostname> show mfea interface
Interface       State    Vif/PifIndex    Addr        Flags
dc0             UP        0/6            10.4.0.1    MULTICAST BROADCAST KERN UP
dc2             UP       1/8             10.3.0.2    MULTICAST BROADCAST KERN UP
register_vif    UP       2/6             10.4.0.1    PIM REGISTER KERN UP
```

```
user@hostname> show mfea interface address
Interface       Addr        Subnet        Broadcast     P2Paddr
dc0             10.4.0.1    10.4.0.0/24   10.4.0.255    0.0.0.0
dc2             10.3.0.2    10.3.0.0/24   10.3.0.255    0.0.0.0
register_vif    10.4.0.1    10.4.0.1/32   10.4.0.1      0.0.0.0
```

โครงสร้างคำสั่งของเราท์เตอร์ Vyatta

สำหรับคำสั่ง CLI บนเราท์เตอร์ Vyatta จะคล้ายกับ XORP ดังนั้นในส่วนนี้จำกล่าวถึงเฉพาะการใช้ GUI โดยปกติเราเตอร์ Vyatta จะปิดการทำงานของ GUI โดย default เนื่องจากปัญหาเรื่องความปลอดภัยในการถูกเจาะระบบผ่านทางเว็บ แต่เมื่อผู้ใช้มีความจำเป็นต้องใช้งาน GUI ก็สามารถทำได้ ซึ่งมีขั้นตอนดังต่อไปนี้

1. เปิดบริการ Web GUI

Login เข้าไปที่เราท์เตอร์(default เป็น user=vyatta, password=vyatta) จากนั้นใช้คำสั่งเปิดบริการเว็บเซิร์ฟเวอร์ ด้วยคำสั่ง set service https แล้วตามด้วย commit ดังตัวอย่าง

vyatta@vyatta:$ **configure**

vyatta@vyatta# **set service https**

[edit]

vyatta@vyatta# **commit**

[edit]

vyatta@vyatta#

```
vyatta@vyatta# commit
Generating a 1024 bit RSA private key
.............................................................++++++
...++++++
writing new private key to '/etc/lighttpd/server.pem'
-----
Stopping web server: lighttpd.
Starting web server: lighttpd.
Stopping PAGER server
Starting PAGER server
[edit]
vyatta@vyatta#
```

รูปที่ 13.63 แสดงการเปิด service เว็บเซิร์ฟเวอร์

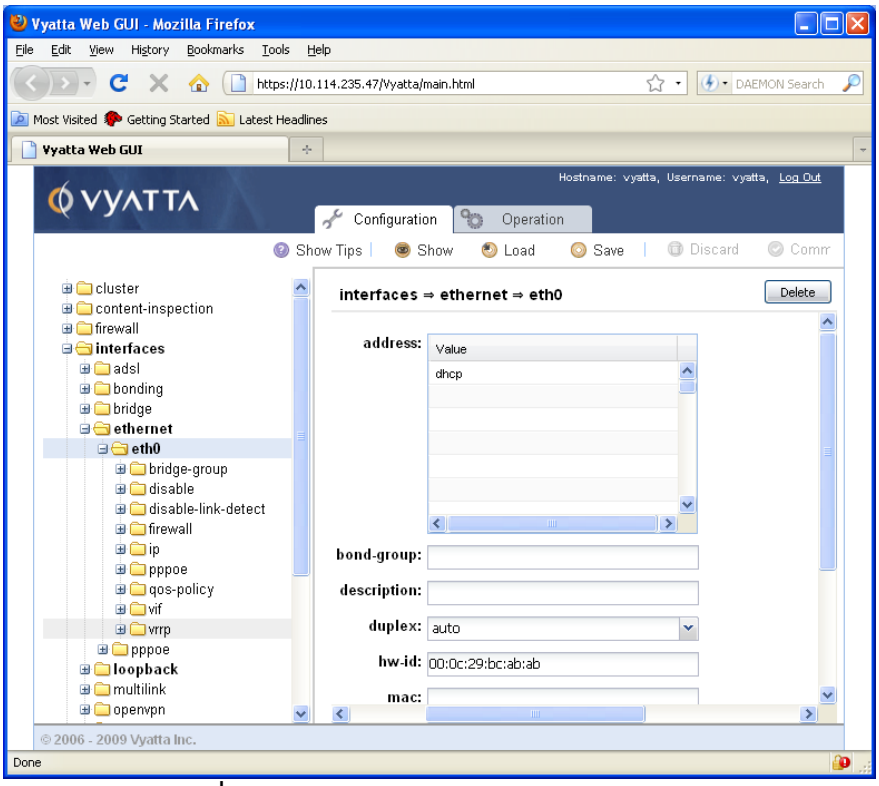
NOTE: คำสั่งที่ยังไม่ได้ commit ของ vyatta จะใช้สัญลักษณ์ “+” แทน “-”

2. ทดสอบการทำงานของ GUI โดยผ่านทางเว็บบราว์เซอร์ ซึ่งมีรูปแบบคือ https://IP Address เช่น https://192.168.1.10 user ที่สำหรับใช้ login จะไม่อนุญาตให้ใช้ root เนื่องการรักษาความปลอดภัย ดังนั้นให้ทำการล็อกอินด้วย user ชื่อว่า vyatta และรหัสผ่านเป็น vyatta

รูปที่ 13.64 ล็อกอินเราท์เตอร์ vyatta

NOTE: เว็บบราว์เซอร์สนับสนุน firefox 3 ขึ้นไปหรือ Internet Exploere 7 ขึ้นไป ในการใช้งาน HTTPS อาจจะมีข้อความเตือนเรื่อง Certificate ในเบื้องต้นให้ยอมรับ certificate ของ vyatta ก่อน

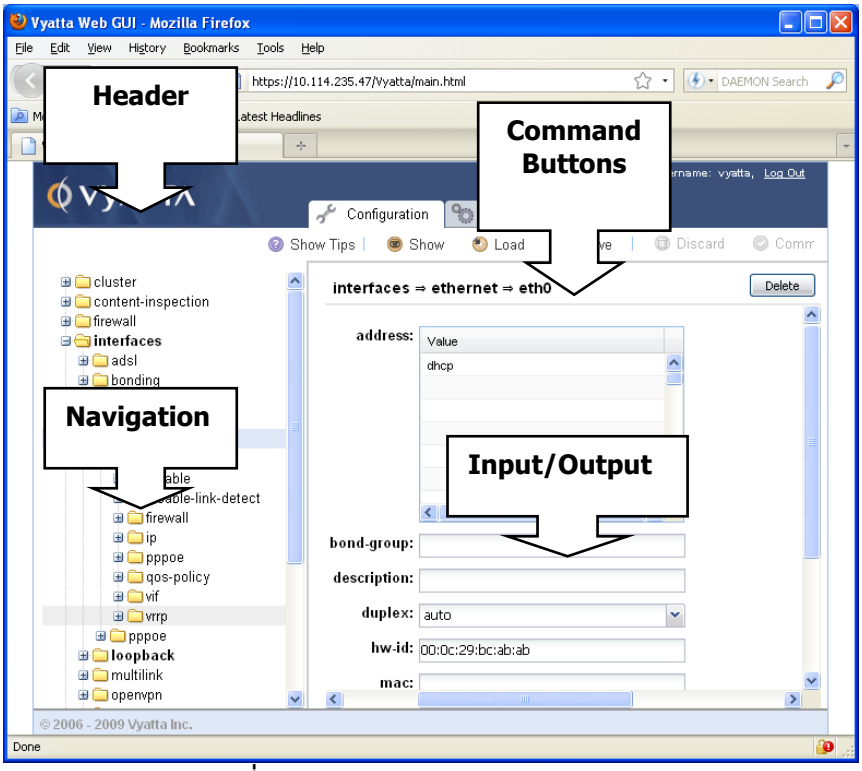
เมื่อทำการ login เรียบร้อยแล้ว จะ**ปรากฏ**เมนูให้ทำการปรับแต่งเราท์เตอร์ผ่านทางเว็บบราว์เซอร์ดังรูปที่ 13.64 สำหรับประโยชน์ของการใช้ GUI คือ ง่ายต่อการใช้งานโดยไม่จำเป็นต้องจดจำคำสั่งเหมือนใช้ CLI (ผู้เขียนแนะนำว่าควรจะใช้ CLI จะปลอดภัยกว่า)



รูปที่ 13.65 แสดงการคอนฟิก vyatta ผ่าน GUI

3. ส่วนประกอบของ GUI

Vyatta GUI แบ่งส่วนประกอบออกเป็น 5 ส่วนคือ Header, Navigation, Command Buttons, Input และ Output ดังรูปที่ 13.65 Header ทำหน้าที่แสดง logo, Host name, ชื่อผู้ใช้ที่ login, logout และแท็บของโหมดการคอนฟิกเราท์เตอร์คือ แท็บ Configuration และ Operation ส่วนของ Navigation แสดงรายละเอียดของเมนูเป็นลักษณะโครงสร้างแบบทรีเหมือนกับที่แสดงใน CLI (ยกเว้นคุณสมบัติบางอย่างที่ไม่ครบเหมือน CLI เช่น telnet, terminal, update proxy เป็นต้น) ส่วนของ Command Buttons แสดงรายละเอียดของปุ่มที่ใช้งานบ่อยๆ เช่น Show, Load, Save, Discard, Commit เป็นต้น ส่วนสุดท้ายคือ Input/Output แสดงรายละเอียดข้อมูลที่ต้องการสั่งงานเราท์เตอร์หรือรับผลเมื่อสั่งงานเรียบร้อยแล้ว กลับมาให้ผู้ใช้งานได้ทราบ



รูปที่ 13.66 แสดงส่วนประกอบของ GUI

ในการใช้งานผู้ใช้สามารถเลือกคลิกที่โหนดใดๆ ที่ Navigation GUI ก็จะแสดงรายละเอียดในส่วนนั้นๆ ให้ผู้ใช้งานได้ทราบ เมื่อกำหนดค่าคอนฟิกให้กับเราท์เตอร์ เบื้องต้นเราท์เตอร์จะแสดงปุ่มสีเหลืองหมายถึงยังไม่มีการ commit ถ้าแสดงปุ่มสีแดงแสดงว่าข้อมูลที่ใส่ไม่ถูกต้องต้องมีการแก้ไขให้ถูกต้องก่อน

4. การสร้างโหมดด้วย GUI ทำตามขั้นตอนดังต่อไปนี้
    - ไปยังส่วนของ Navigation คลิกเลือกโหนดที่ต้องการสร้างโหนดย่อย
    - ในส่วนของ Input/Output ให้ป้อนโหนดที่ต้องการสร้างใหม่ ต่อจากนั้นเลือก set โหนดที่สร้างใหม่จะแสดงตัวอักษรเป็นสีดำหนา เนื่องจากยังไม่มีการ commit เมื่อมั่นใจแล้วว่าโหนดคอนฟิกที่สร้างใหม่ถูกต้องให้กดปุ่ม commit ตัวอักษรก็จะกลายเป็นสีเหลือง
5. การลบโหนด ก็เป็นลักษณะเดียวกันคือ ไปยัง Navigation แตกโครงสร้างของทรีออก แล้วหาโหนดที่ต้องการลบ เมื่อเจอโหนดคอนฟิกที่ต้องการลบแล้วให้กดปุ่ม delete ถ้าเผลอไปลบโหนดผิดให้กดปุ่ม discard แต่ถ้าโหนดที่ต้องการลบถูกต้องแล้วให้กดปุ่ม commit ขบวนการลบโหนดคอนฟิกก็จะทำงานทันที

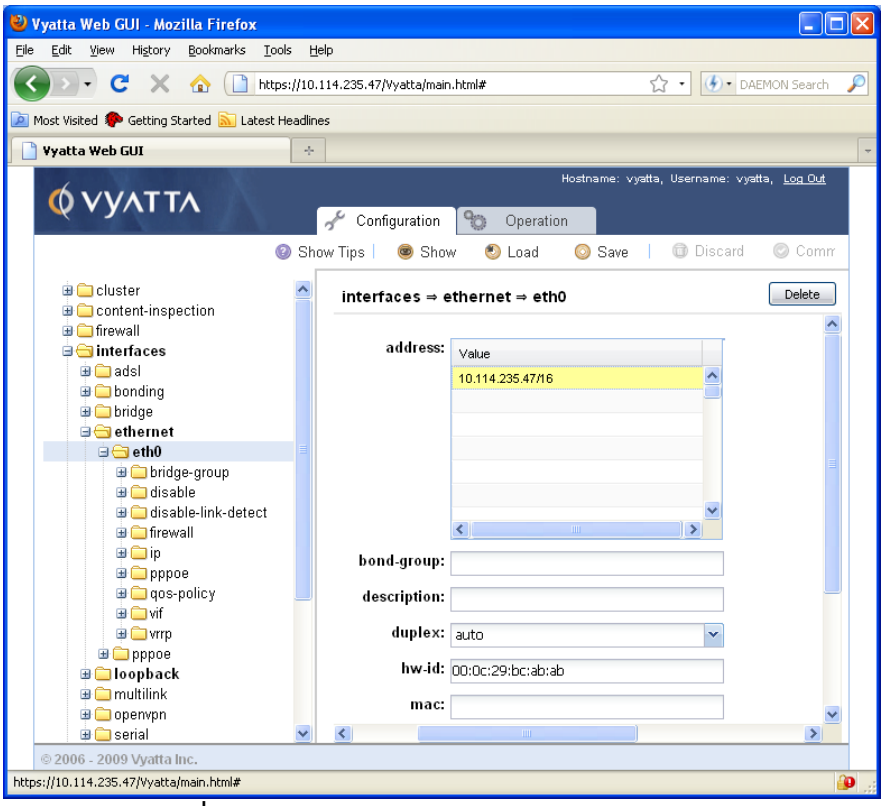
ตัวอย่างการคอนฟิกระหว่าง CLI กับ GUI

เมื่อทำการคอนฟิกเราท์เตอร์ผ่าน CLI ให้อินเทอร์เฟส eth1 มีหมายเลขไอพีคือ 192.168.1.61/24 สามารถกำหนดได้ดังนี้

```
vyatta@R1# set interfaces ethernet eth1 address 10.14.235.47/16
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

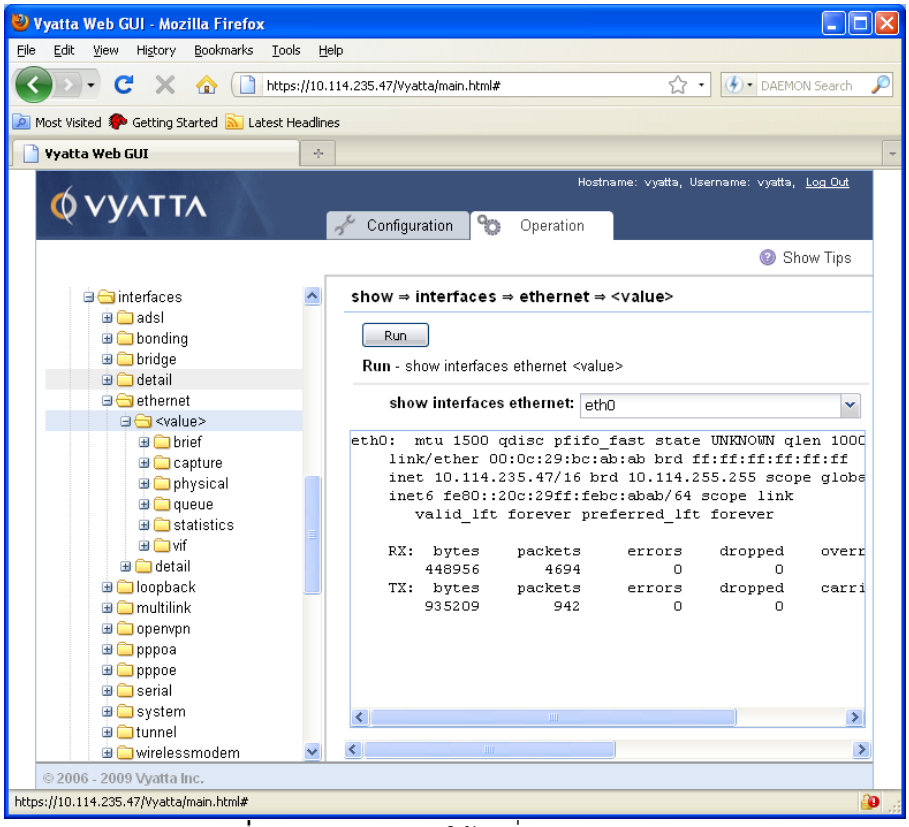
สำหรับการคอนฟิกในสถานกราณ์เดียวกันนี้ผ่าน GUI สามารถทำได้ดังนี้คือ

- Login เข้าสู่ vyatta GUI
- เลือก Tab Configuration
- ในส่วนของ Navigation เลือก > interfaces > ethernet > eth1
- ป้อนหมายหมายไอพีแอดเดรสในช่อง address ในส่วนของ Input/Optput



รูปที่ 13.67 แสดงการคอนฟิกอินเทอร์เฟสผ่าน GUI

- คลิกเลือกที่ปุ่ม set จะ**ปรากฏ**วงกลมสีเหลืองเล็กๆ ตรงส่วนข้อมูลที่ป้อนเข้าไปใหม่
- คลิกปุ่ม commit เพื่อยืนยันการเปลี่ยนแปลง วงกลมสีเหลืองที่ปรากฏจะหายไป
- ทดสอบใช้คำสั่ง show เพื่อตรวจสอบค่าคอนฟิกที่ปรับปรุง โดนเลือกที่ Navigation > show > interfaces > ethernet <value>
- เลือก eth1 จาก show interfaces ethernet
- คลิกปุ่ม Run ผลลัพธ์จะแสดงในส่วนของ Input/Output ดังรูปที่ 13.68



รูปที่ 13.68 แสดงการใช้คำสั่ง show บน GUI

- **ตัวอย่างการสร้างระบบเครือข่าย**

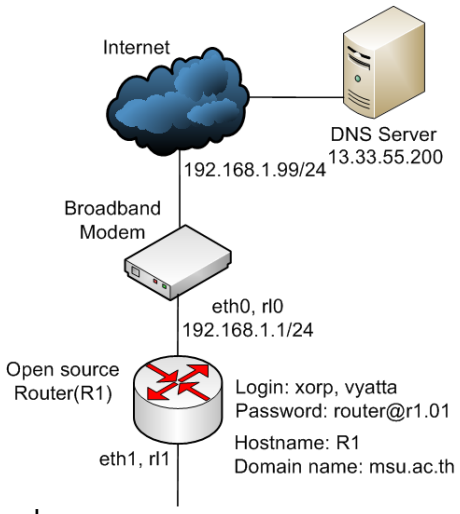
ในหัวข้อนี้จะทดสอบการสร้างเครือข่ายอย่างง่ายสัก 2 ตัวอย่าง เพื่อให้เห็นภาพการใช้โอเพนซอร์สเราท์เตอร์ สร้างระบบเครือข่ายจริง หลังจากที่เรียนรู้การใช้คำสั่งพื้นฐานในหัวข้อก่อนหน้ามาเรียบร้อยแล้ว

Scenario 1: Basic System Configuration

ใน scenario ที่ 1 จะทำการสร้างเครือข่ายแบบง่ายๆ เพื่อสร้างความเข้าใจเบื้องต้นสำหรับการคอนฟิกเราท์เตอร์เสียก่อน และใน scenario ต่อๆ ไป เครือข่ายจะมีความซับซ้อนเพิ่มขึ้นเรื่อยๆ ใน scenario นี้จะต้องสามารถเรียนรู้การคอนฟิกเราท์เตอร์ดังนี้

- การ login
- การเข้าสู่ configuration mode
- การกำหนด host name
- การกำหนด domain name
- การเปลี่ยนรหัสผ่าน
- การคอนฟิกอินเทอร์เฟส
- การกำหนดการเข้าถึง DNS server

- การกำหนด default gateway



รูปที่ 13.69 แสดงผังเครือข่าย Scenario 1

ขั้นตอนการคอนฟิก

1. login เข้าไปยังเราท์เตอร์ (user ต้องไม่ใช้ root สำหรับ default ของ Vyatta คือ user=vyatta, password=vyatta)

```
Welcome to Vyatta - vyatta tty1
vyatta login: vyatta
Password:
Linux vyatta 2.6.20 #1 SMP Fri Sep 21 02:22:08 PDT 2007 i686
Welcome to Vyatta.
This system is open-source software. The exact distribution
terms for each module comprising the full system are described
in the individual files in /usr/share/doc/*/copyright.
Last login: Sat Nov 10 16:48:48 2007 on tty1
vyatta@vyatta:~$
```

2. เข้าสู่โหมด Operation > Configuration

```
vyatta@vyatta:~$ configure
[edit]
vyatta@vyatta#
```

3. กำหนดชื่อ hostname เป็น R1 ด้วยคำสั่ง set system hostname

```
vyatta@vyatta# set system host-name R1
[edit]
vyatta@vyatta# commit
[edit]
vyatta@vyatta#
```

4. กำหนดชื่อของโดเมนเนม ด้วยคำสั่ง set system domain-name

```
vyatta@R1# set system domain-name msu.ac.th
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

5. เปลี่ยนรหัสผ่าน vyatta จะมีผู้ใช้ 2 ประเภทคือ root รหัสผ่าน default คือ vyatta และผู้ใช้ชื่อ vyatta รหัสผ่าน vyatta เปลี่ยนรหัสผ่านเป็น router@r1.01 ด้วยคำสั่ง set system login user

```
vyatta@R1# set system login user vyatta authentication
plaintext-password router@r1.01
```

```
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

6. โดยปกติ vyatta จะค้นหาอินเทอร์เฟสเมื่อเริ่มต้นทำงานอัตโนมัติและใช้ชื่ออินเทอร์เฟสเป็น ethx (x คือ หมายเลขลำดับของการ์ดอินเทอร์เฟส เช่น eth0 หมายถึงการ์ดใบที่ 1) การกำหนดหมายเลขไอพีให้กับอินเทอร์เฟสจะใช้คำสั่ง set interfaces ในผังไดอะแกรมจะกำหนดให้ eth0 ที่เชื่อมต่อไปยังอินเทอร์เน็ตมีหมายเลขไอพีคือ 192.168.1.1/24

```
vyatta@R1# set interfaces ethernet eth0 address 192.168.1.1/24
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

ถ้ามีการรับหมายเลขไอพีอัตโนมัติ(DHCP) จาก ISP ให้เปลี่ยนเป็นคำสั่ง set interfaces ethernet eth0 address dhcp แทน และทำการตรวจสอบการคอนฟิกด้วยคำสั่ง show interfaces

```
vyatta@R1# show interfaces
ethernet eth0 {
        address 192.168.1.1/24
        hw-id 00:40:63:e2:e4:00
}
loopback lo {
}
[edit]
vyatta@R1#
```

7. กำหนด DNS Server ให้กับเราท์เตอร์ DNS ทำหน้าที่แปลงโดเมนเป็นหมานเลขไอพีและแปลงหมายเลขไอพีเป็นชื่อโดเมน การกำหนด DNS จะใช้คำสั่ง set system name-server

```
vyatta@R1# set system name-server 13.33.55.200
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

8. กำหนดค่า default gateway ให้เราท์เตอร์ ด้วยคำสั่ง set system gateway-address

```
vyatta@R1# set system gateway-address 192.168.1.99
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

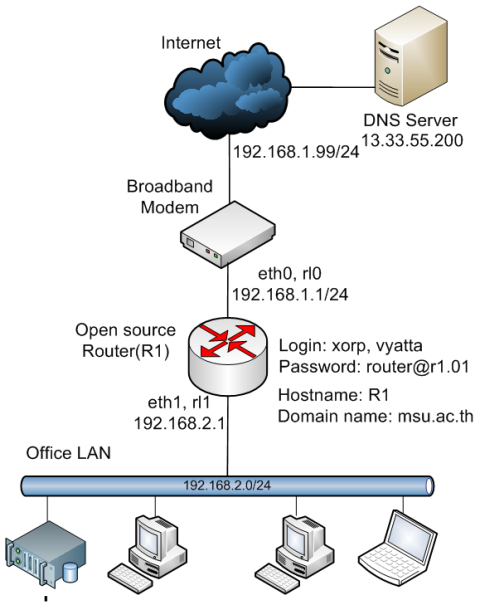
Scenario 2: Internet Gateway

ในตัวอย่างนี้จะสร้างเครือข่ายให้สามารถเชื่อมต่ออินเทอร์เน็ตได้ โดยการใช้ NAT (Network Address Translation) และเปิดให้บริการภายในเครือข่ายหลายรูปแบบ เช่น DHCP, SSH, Firewall เป็นต้น ซึ่งมีรายละเอียดดังนี้

- เครือข่ายสามารถสื่อสารได้ทั้งภายในเครือข่าย LAN และ Internet
- เปิดให้ผู้ใช้งานสามารถใช้ SSH เพื่อเข้าถึงข้อมูลอย่างปลอดภัยได้
- เปิดให้บริการ DHCP ซึ่งจะแจกจ่ายไอพีกับเครื่องลูกข่ายภายใน LAN ได้

- เปิดบริการ NAT ซึ่งจะช่วยให้เครื่องลูกข่ายสามารถเชื่อมต่ออินเทอร์เน็ตได้มากขึ้น

ความต้องการพื้นฐานสำหรับเครื่องเราท์เตอร์คือ จำเป็นต้องมีการ์ดเครือข่ายอย่างน้อย 2 ใบ ใบแรกใช้สำหรับเชื่อมต่ออินเทอร์เน็ตและใบที่สองสำหรับเชื่อมต่อภายในเครือข่าย LAN ดังรูป



รูปที่ 13.70 แสดงผังเครือข่าย Scenario 2

ลำดับขั้นตอนการคอนฟิกเราท์เตอร์จะเชื่อมโยงกับ Scenario ที่ 1 สิ่งที่ต้องคอนฟิกเพิ่มเติมมีดังต่อไปนี้

- คอนฟิกอินเทอร์เฟสภายในเครือข่าย LAN
- เปิดการใช้งานโพรโทคอล SSH
- เปิดให้บริการ DHCP server
- เปิดให้บริการ NAT
- เปิดบริการไฟล์วอลล์

1. การคอนฟิกอินเทอร์เฟสภายในเครือข่าย LAN

จากรูปที่ 13.70 การ์ดอินเทอร์เฟสที่เชื่อมต่อเครือข่าย LAN คือ eth1 มีหมายเลขไอพีคือ 192.168.2.1 subnet mask /24 สามารถกำหนดบนเราท์เตอร์ดังนี้

```
vyatta@R1# set interfaces ethernet eth1 address 192.168.2.1/24
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

ทดสอบแสดงผลการคอนฟิก

```
vyatta@R1# show interfaces
ethernet eth0 {
address 192.168.1.1/24
hw-id 00:40:63:e2:e4:00
}
ethernet eth1 {
address 192.168.2.1/24
hw-id 00:13:46:e6:f6:87
```

```
}
loopback lo {
}
[edit]
vyatta@R1#
```

2. เปิดการใช้งานโพรโทคอล SSH

SSH เป็นโพรโทคอลที่อนุญาตให้ผู้ใช้งานสามารถเข้าถึงข้อมูลได้อย่างปลอดภัย โดยการใช้เทคนิคการเข้ารหัสข้อมูล ซึ่งสามารถเปิดบริการได้ดังนี้

```
vyatta@R1# set service ssh
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

3. เปิดให้บริการ DHCP server

```
vyatta@R1# set service dhcp-server shared-network-name
ETH1_POOL subnet 192.168.2.0/24 start 192.168.2.100 stop
192.168.2.199                    //กำหนดให้แจกไอพีเริ่มตั้งแต่หมายเลข 100 - 199
[edit]
vyatta@R1# set service dhcp-server shared-network-name
ETH1_POOL subnet 192.168.2.0/24 default-router 192.168.2.1
        //กำหนดทางออกคือ default gateway ให้กับเน็ตเวิร์คที่แจก DHCP
[edit]
vyatta@R1# set service dhcp-server shared-network-name
ETH1_POOL subnet 192.168.2.0/24 dns-server 13.33.55.200
        //กำหนดไอพีของ DNS เพื่อให้ลูกข่ายสามารถเรียกใช้งานโดเมนเนมบนอินเทอร์เน็ตได้
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

DHCP เป็นโพรโทคอลที่ทำหน้าที่แจกไอพีแอดเดรสให้เครื่องลูกข่ายแบบอัตโนมัติ ซึ่งจะช่วยให้ง่านต่อการบริหารจัดการหมายเลขไอพีในองค์กรที่มีขนาดค่อนข้างใหญ่ สำหรับใน Scenario นี้จำทำการแจกหมายเลขไอพีเริ่มตั้งแต่ 192.168.2.100 – 192.168.2.199 จำนวนทั้งสิ้น 100 ไอพี สามารถคอนฟิกบนเราท์เตอร์ได้ดังนี้

ตรวจสอบการสั่งงานด้วยคำสั่ง show service dhcp-server

```
vyatta@R1# show service dhcp-server
shared-network-name ETH1_POOL {
        subnet 192.168.2.0/24 {
                start 192.168.2.100 {
                        stop 192.168.2.199
                }
                dns-server 13.33.55.200
                default-router 192.168.2.1
            }
}
[edit]
vyatta@R1#
```

4. เปิดให้บริการ NAT

NAT เป็นบริการที่สามารถแปลงไอพีเครือข่ายภายในซึ่งเป็นไอพีปลอม (IP Private) ให้สามารถใช้งานเครือข่ายอินเทอร์เน็ตได้ (การใช้งานอินเทอร์เน็ตต้องเป็นไอพีจริงเท่านั้น) และช่วย

ในเรื่องของไอพีจริงที่ไม่อยู่ไม่เพียงพอต่อการใช้งาน และช่วยในเรื่องของความปลอดภัยในการซ่อนเครือข่ายจากการโจมตีจากอินเทอร์เน็ตด้วย การทำ NAT อันดับแรกต้องสร้างกลุ่มของลูกข่ายที่ต้องการใช้งานอินเทอร์เน็ตก่อน อันดับที่ 2 คือการผูกกลุ่มที่ตั้งไว้เข้ากับกระบวนการแปลงไอพีให้สามารถเชื่อมต่ออินเทอร์เน็ตได้ ซึ่งมีขั้นตอนดังนี้

```
vyatta@R1# set service nat rule 1 source address 192.168.2.0/24 //สร้างลูกข่ายกลุ่มที่ 1
[edit]
vyatta@R1# set service nat rule 1 outbound-interface eth0
                    //ผูกกลุ่มที่ 1 เข้ากับ NAT ที่อินเทอร์เฟส eth0
[edit]
vyatta@R1# set service nat rule 1 type masquerade        //แปลงเครื่องลูกข่ายเป็น IP จริง
[edit]
vyatta@R1# commit
[edit]
vyatta@R1#
```

ตรวจสอบการสั่งงานด้วยคำสั่ง show service nat

```
vyatta@R1# show service nat
          rule 1 {
                    type masquerade
                    outbound-interface eth0
                    source {
                              address 192.168.1.0/24
                            }
                    }
[edit]
vyatta@R1#
```

13.6 Network Routing Protocols

13.6.1 Unicast Routing [66, 67]

เราท์เตอร์ทำหน้าที่ในการ forward ข้อมูลไปยังเป้าหมายปลายทาง ดังนั้นการดูแลและจดจำเส้นทางจึงเป็นสิ่งที่เราท์เตอร์ต้องให้ความสำคัญ เส้นทางที่เราท์เตอร์ต้องดูแลนั้นถ้าจะกล่าวย่อๆ คือที่อยู่ปลายทางที่เราท์เตอร์ต้องส่งข้อมูลไปให้ถึง ซึ่งประกอบไปด้วย subnet และ next hop สำหรับ subnet จะประกอบไปด้วยหมายเลข IP address และ prefix-length ตัวอย่างเช่น subnet 128.13.64.9/24 มีความยาวของ prefix-length จำนวน 24 บิต เมื่อคำนวณจำนวนเครือข่ายทั้งหมดได้เท่ากับ 2^{24} = 1,677,7216 เครือข่ายและมีจำนวนโฮสต์ในแต่ละเครือข่ายเท่ากับ 2^{8} = 256(32 – 24 = 8) เครื่อง เมื่อเราท์เตอร์ต้องการจะส่งข้อมูลไปยังเครื่อง 128.13.64.9 จะต้องทำการ forward แพ็กเก็ตไปยังเครือข่าย 128.**13.**40.0(วิธีการคำนวณให้นำเอา IP address มา AND กับ prefix-length คือ 128.13.64.9 AND 255.255.255.0 = 128.13.64.0) ในเน็ตเวิร์คนี้จะมีขอบเขตของช่วงไอพีคือเริ่มตั้งแต่ 128.13.64.0 – 128.13.64.255 (โดยปกติไอพีแรกคือ network id = 128.13.64.0 และไอพีหมายเลขสุดท้ายคือ broadcast = 128.13.64.255) สำหรับ next hop คือหมายเลขไอพีของเราท์เตอร์ตัวถัดไปที่ต้องการ forward ข้อมูลไปยังปลายทาง เราท์เตอร์ตัวดังกล่าวจะเชื่อมต่อกันโดยตรง (neighbor) เราท์เตอร์จะเก็บข้อมูลของเส้นทางไว้ในตาราง routing เมื่อรับแพ็กเก็ตเข้ามาเราท์เตอร์จะทำการตรวจสอบเส้นทางที่จะส่งข้อมูล เมื่อตรวจสอบว่าเป็นเน็ตเวิร์คที่ตนเองดูแลอยู่ก็

จะส่งข้อมูลต่อไปยังอินเทอร์เฟสที่เหมาะสม แต่ถ้าข้อมูลนั้นไม่ใช่เน็ตเวิร์คที่ตัวเองดูแลอยู่ก็จะส่งต่อไปยัง next hop ที่เหมาะสมต่อไป การค้นหาหรือตรวจสอบเส้นทาง เราท์เตอร์จะใช้วิธีการที่เรียกว่า longest prefix match วิธีการนี้จะให้ความสำคัญกับ prefix-length ที่มีความยาวมากที่สุดก่อน ตัวอย่างเช่น สมมุติว่าเราท์เตอร์มีเส้นทางในตาราง routing 2 เส้นทางคือ

Subnet: 128.13.0.0/16, nexthop: 10.0.0.1 เส้นทางที่ 1

Subnet: 128.13.64.0/24, nexthop: 10.0.0.2 เส้นทางที่ 2

เมื่อแพ็กเก็ตที่รับเข้ามาเป็น 128.13.0.1 เส้นทางที่ 1 เท่านั้นที่จะต้องทำงาน และข้อมูลจะถูกส่งต่อไปยัง next hop หมายเลข 10.0.0.1 แต่ถ้าปรากฏว่าแพ็กเก็ตที่รับเข้ามาเป็น 128.13.64.1 จะเกิดการ match เข้ากับเส้นทางทั้งสองเส้นทาง คำถามที่ตามมาคือจะเลือกเส้นทางไหนล่ะ? คำตอบคือจะเลือกเส้นทางที่มี prefix-length ยาวกว่าทำงาน (24 บิต > 16 บิต) สำหรับในที่นี้จะเลือกเส้นทางที่ 2 คือ 128.13.64.0/24 ต่อจากนั้นข้อมูลจะถูกส่งต่อไปยัง next hop คือ 10.0.0.2

เราท์เตอร์จะสร้างเส้นทางในตาราง routing ได้ 3 รูปแบบคือ

- เส้นทางที่ได้จากเน็ตเวิร์คที่เชื่อมต่อกับเราท์เตอร์โดยตรงจะถูกสร้างเส้นทางโดยอัตโนมัติ เพราะว่าเราท์เตอร์จะรู้จักเป็นอย่างดี
- เส้นทางที่เกิดจากการคอนฟิกของผู้ดูแลระบบ เช่น static routing
- เส้นทางที่เราท์เตอร์เรียนรู้ได้เราท์เตอร์เพื่อนบ้าน โดยอาศัยโพรโทคอลค้นหาเส้นทางแบบ dynamic routing

13.6.1.1 Dynamic Routing [C603]

Dynamic routing คือโพรโทคอลที่มีความสามารถในการค้นเส้นทางได้ด้วยตนเองแบบอัตโนมัติโดยอาศัยการแลกเปลี่ยนข้อมูลกันระหว่างเราท์เตอร์เพื่อนบ้าน สำหรับโพรโทคอลประเภท dynamic มีหลายตัวดังนี้

- Routing Information Protocol (RIP) เป็นโพรโทคอลที่ใช้งานกับเครือข่ายที่มีขนาดเล็กๆ หรือใช้ภายในองค์กร(intra-domain) เช่นในเครือข่าย LAN, intranet เป็นต้น
- Open Shortest Path First (OSPF) เป็นโพรโทคอลที่ใช้งานกับเครือข่ายที่มีขนาดกลางถึงขนาดใหญ่ แต่ยังมีขอบเขตการใช้งานอยู่ภายในองค์กรเช่นเดียวกันกับ rip
- Integrated IS-IS ใช้งานเหมือนกับ OSPF
- IGRP เป็นโพรโทคอลที่ใช้งานกับเครือข่ายที่มีขนาดเล็กถึงขนาดกลาง และยังมีขอบเขตการใช้งานอยู่ภายในองค์กร ผู้พัฒนาคือบริษัท cisco
- Border Gateway Protocol (BGP) เป็นโพรโทคอลที่ใช้งานกับเครือข่ายที่มีขนาดใหญ่และขทำงานกับเครือข่ายอินเทอร์เน็ต

ปัจุบัน XORP ให้การสนับสนุนโพรโทคอล dynamic routing คือ RIP, OSPF, BGP รวมถึง multicast ด้วย แต่ยังไม่สนับสนุน IS-IS

13.6.1.2 Administrative Distance [68]

เราท์เตอร์มีความสามารถในการสั่งให้โพรโทคอล routing ทำงานได้พร้อมๆ กันหลายๆ โพรโทคอล ตัวอย่างเช่น อาจจะใช้โพรโทคอล RIP ทำงานใน Intranet และ BGP ทำงานกับเครือข่าย Internet จากในตัวอย่างนี้เราท์เตอร์จะต้องสร้างตาราง routing 2 ประชนิดคือ ตาราง rotuing สำหรับ RIP และอีกตารางสำหรับ BGP เช่น

Subnet: 128.13.64.0/24, nexthop: 192.150.187.1 สำหรับ BGP โดยเรียนรู้จาก external peer ที่มี AS 123, 567, 987 และ

Subnet: 128.13.64.0/24, nexthop: 10.0.0.2 สำหรับ RIP มี metric เท่ากับ 13 เป็นต้น จากในตัวอย่าง เส้นทางทั้งสองเหมือนกัน แล้วเราท์เตอร์จะตัดสินใจอย่างไร? จะใช้ prefix-length ก็ไม่สามารถทำได้เนื่องจาก prefix-length เท่ากัน หรือจะเปรียบเทียบด้วย metric ก็ไม่ได้เนื่องจาก metric ของ BGP และ RIP ไม่สามารถเปรียบเทียบกันได้ ในกรณีนี้ XORP เราท์เตอร์จะตัดสินด้วยวิธีการที่เรียกว่า administrative distance คือการกำหนดเส้นทางใดเส้นทางหนึ่งเป็นเส้นทางหลักด้วยตัวของผู้ดูแลเครือข่ายเอง โดยในแต่ละโพรโทคอล routing จะถูกกำหนดค่าตัวเลขจำนวนเต็มค่าหนึ่ง เพื่อเป็นการบ่งบอกว่าเราท์เตอร์ควรจะเลือกเส้นทางไหนเมื่อเส้นทางทั้ง 2 เหมือนกันดังกรณีที่ยกตัวอย่างข้างต้น ซึ่งโดยปกติจะให้ค่าจำนวนเต็มที่เล็กกว่ามีนัยสำคัญสูงกว่า บนเราท์เตอร์ XORP ได้กำหนดตัวเลขจำนวนเต็มดังกล่าวไว้แล้วดังนี้

ตารางที่ 13.11 แสดงค่า administrative distance ของแต่ละโพรโทคอล

ประเภทของโพรโทคอล routing	**ค่า administrative distance**
Directly connected subnets	0
Static routes	1
BGP, heard from external peer	20
OSPF	110
IS-IS	115
RIP	120
BGP, heard from internal peer	200
FIB2MRIB routes (XORP-specific, in MRIB only)	254

จากตารางที่ 13.11 แสดงว่าเราท์เตอร์ที่เกิดปัญหาข้างต้นจะเลือกเส้นทางที่โพรโทคอล BGP ได้สร้างไว้ก่อน เนื่องจาก administrative distance สูงกว่า (20 มีนัยสำคัญสูงกว่า 120)

สำหรับการตรวจสอบหรือดูข้อมูลของ administrative distance นั้นสามารถทำได้โดยใช้คำสั่ง show route admin distance ipv4 unicast

```
user@hostname> show route admin distance ipv4
unicast
Protocol            Administrative distance
connected            0
static               1
eigrp-summary        5
ebgp                 20
eigrp-internal       90
igrp                 100
ospf                 110
is-is                115
rip                  120
eigrp-external       170
ibgp                 200
fib2mrib             254
unknown              255
```

สำหรับ IP เวอร์ชัน 6 จะใช้คำสั่ง show route admin distance ipv6 unicast และมัลติคาสท์ใช้คำสั่ง show route admin distance ipv4 multicast สำหรับ IPv4 และ show route admin distance ipv6 multicast สำหรับ IPv6

13.6.1.3 Route Redistribution(การจัดเส้นทางใหม่) [69]

สำหรับเหตุผลที่จำเป็นต้องมีการจัดเส้นทางใหม่ระหว่างโพรโทคอล routing คือ

- มีบางเส้นทางที่เกิดขึ้นจาก static route และบางเส้นทางเกิดจากโพรโทคอล dynamic route บนเราท์เตอร์ ในกรณีนี้อาจจะเกิดขึ้นจากในการคอนฟิกเราท์เตอร์เริ่มต้นอาจจะใช้ static route ก่อน เมื่อเวลาผ่านไปจึงใช้ Dynamic route เช่น RIP เป็นต้น วิธีนี้สามารถแก้ไขได้ง่ายๆ โดยการยกเลิกเส้นทางแบบ static ออก แล้วนำเส้นทางที่ยกเลิกนั้นไปประกาศใน dynamic route แทน
- อีกกรณีหนึ่งคือ เมื่อเน็ตเวิร์คภายในใช้โพรโทคอล RIP (LAN) และเราท์เตอร์ตัวเดียวกันก็ทำการรันโพรโทคอล BGP ที่ค้นหาเส้นทางบนอินเทอร์เน็ตพร้อมกันด้วย ซึ่งโดยปกติมักนิยมคอนฟิกให้โพรโทคอล BGP บนเราท์เตอร์ที่ทำหน้าที่เป็น Border Router กระจายข้อมูลเกี่ยวกับเส้นทางบนเครือข่ายให้กับเครือข่ายภายใน แต่เมื่อเครือข่ายภายในเพิ่มขึ้นใหม่(RIP จะทราบแต่ BGP ไม่ทราบข้อมูลนี้) Border Router จำเป็นต้องทราบการเปลี่ยนแปลงนี้

ด้วย วิธีแก้ปัญหาคือเราต้องคอนฟิกให้เราท์เตอร์ที่ทำหน้าที่เป็น Border Router ทำการ Redistribution ข้อมูลเส้นทางของ RIP ไปให้กับ BGP ด้วย

สำหรับ XORP เราท์เตอร์มีความสามารถในการทำ Route Redistribution โดยใช้คำสั่ง import และ export ซึ่งจำกล่าวต่อไปในส่วนของ routing policy framework

13.6.2 Static Route [70]

Static route เป็นการกำหนดเส้นทางให้เราท์เตอร์จากผู้ดูแลระบบเอง เป็นแบบ manual ฉะนั้นผู้ดูแลระบบจะต้องทราบสถานะกราณ์ระบบเครือข่ายเป็นอย่างดี เมื่อลิงค์ใดลิงค์หนึ่งดาวน์ลงไป เราท์เตอร์ที่คอนฟิกไว้แบบ static จะไม่มีการเปลี่ยนเส้นทางให้ ข้อมูลจะถูกส่งไปยังลิงค์ที่ดาวน์นั้นอย่างต่อเนื่องจนกว่าผู้ดูแลระบบจะทราบ ซึ่งโดยปกติ static route จะใช้กับเครือข่ายที่ผู้ดูแลระบบสามารถควบคุมได้ หรือเป็นเครือข่ายที่ไม่ซับซ้อนและเชื่อมต่อแบบง่ายๆ หรือทดแทนโพรโทคอล dynamic routing ที่ทำงานบนเครือข่ายที่มีเส้นทางไม่มากและขนาดเล็ก static route สามารถทำงานได้ทั้ง IPv4 และ IPv6 บนโพรโทคอลยูนิคาสท์ หรือใช้กับมัลติคาสท์ก็ได้

RIP (Routing Information Base) [71] คือฐานข้อมูลที่รวบรวมหรือจัดเก็บข้อมูลเส้นทางทั้งหมดของเราท์เตอร์ที่ได้จากการการคอนฟิกโดยผู้ดูแลระบบหรือจากการเรียนรู้ของเราท์เตอร์เองโดยผ่านโพรโทคอล dynamic routing RIP จะแยกเก็บเส้นทางระหว่าง IPv4 และ IPv6 ออกจากกันและภายในแต่ละส่วนก็ทำการจัดเก็บข้อมูลของยูนิคาสท์กับมัลติคาสท์แยกกันอีก โดยเส้นทางของยูนิคาสท์จะถูกใช้ forward แพ็กเก็ตโดยโพรโทคอลที่ทำงานแบบยูนิคาสท์ สำหรับมัลติคาสท์จะไม่ทำการ forward แพ็กเก็ตออกไปตรงๆ แบบยูนิคาสท์ แต่จะอาศัยโพรโทคอลมัลติคาสท์ routing คือ PIM ให้ทำการ forward ข้อมูลแทน เนื่องจากมัลติคาสท์จำเป็นต้องทำการบวนการหนึ่งที่เรียกว่า RPF (Reverse-Path Forwarding) [72] สำหรับ RIB ของมัลติคาสท์จะเรียกชื่อว่า Multicast RIB หรือ MRIB

13.6.2.1 การคอนฟิก static routes

ในการคอนฟิก static route จำเป็นต้องทราบข้อมูล 2 อย่างคือ

- Subnet ปลายทาง และ next-hop
- ฐานข้อมูลที่จะต้องใส่เส้นทางลงไป ระหว่าง RIB หรือ MRIB หรือต้องใส่ทั้งคู่

Configuration Syntax

```
protocols {
  static {
    targetname: text
    disable: bool
    route IPv4-addr/int(0..32) {
      next-hop: IPv4-addr
```

```
    metric: uint(1..65535)
    qualified-next-hop IPv4-addr {
        metric: uint(1..65535)
    }
}
route IPv6-addr/int(0..128) {
    next-hop: IPv6-addr
    metric: uint(1..65535)
    qualified-next-hop IPv6-addr {
        metric: uint(1..65535)
    }
}
mrib-route IPv4-addr/int(0..32) {
    next-hop: IPv4-addr
    metric: uint(1..65535)
    qualified-next-hop IPv4-addr {
        metric: uint(1..65535)
    }
}
mrib-route IPv6-addr/int(0..128) {
    next-hop: IPv6-addr
    metric: uint(1..65535)
    qualified-next-hop IPv6-addr {
        metric: uint(1..65535)
    }
}
interface-route IPv4-addr/int(0..32) {
    next-hop-interface: text
    next-hop-vif: text
    next-hop-router: IPv4-addr
    metric: uint(1..65535)
    qualified-next-interface text {
        qualified-next-vif text {
```

```
                next-hop-router: IPv4-addr
                metric: uint(1..65535)
            }
        }
    }
interface-route IPv6-addr/int(0..128) {
    next-hop-interface: text
    next-hop-vif: text
    next-hop-router: IPv6-addr
    metric: uint(1..65535)
    qualified-next-interface text {
        qualified-next-vif text {
            next-hop-router: IPv6-addr
            metric: uint(1..65535)
        }
    }
}
mrib-interface-route IPv4-addr/int(0..32) {
    next-hop-interface: text
    next-hop-vif: text
    next-hop-router: IPv4-addr
    metric: uint(1..65535)
    qualified-next-interface text {
        qualified-next-vif text {
            next-hop-router: IPv4-addr
            metric: uint(1..65535)
        }
    }
}
mrib-interface-route IPv6-addr/int(0..128) {
    next-hop-interface: text
    next-hop-vif: text
    next-hop-router: IPv6-addr
```

```
                metric: uint(1..65535)
                qualified-next-interface text {
                        qualified-next-vif text {
                                next-hop-router: IPv6-addr
                                metric: uint(1..65535)
                        }
                }
            }
        }
}
```

ความหมายของ parameters ดังต่อไปนี้

protocols: เป็นส่วนที่ใช้สำหรับระบุขอบเขตของของโพรโทคอล routing ทั้งหมดที่ต้องการใช้งานบนเราท์เตอร์

static: เป็นคอนฟิกกูเรชันของ static route

targetname: เป็นการกำหนดชื่อให้กับ static route โดยค่าเริ่มต้นจะมีชื่อเป็น static routes

disable: สั่งให้ static route ทำงาน ถ้ากำหนดให้เป็น True เราท์เตอร์จะมองว่าไม่ได้คอนฟิก static route เอาไว้ แต่ไฟล์คอนฟิกกูเรชันยังคงมีข้อมูลอยู่(เป็นการบอกเราท์เตอร์ว่าไม่ต้องสนใจคอนฟิกดังกล่าวเมื่อมีสถานะเป็น true)

route: เป็นส่วนที่บอกเส้นทางการและจะถูกจัดเก็บไว้ใน RIB รูปแบบในการกำหนดข้อมูลคือ address/prefix-length

next-hop: ระบุถึงหมายเลขไอพีของเราท์เตอร์ตัวถัดไปที่จะส่งข้อมูลต่อไปยังปลายทาง

metric: ระบุถึงค่าใช้จ่ายในการเดินทางของแต่ละเส้นทาง(cost) เมื่อค่าดังกล่าวเป็นเลขจำนวนเต็มลบ จะไม่ใช้ใน static route แต่อาจจะใช้กับโพรโทคอลแบบ dynamic เช่น BGP หรือ PIM-SM ตัวอย่างเช่น BGP จะใช้ IGP metric ในการตัดสินใจเลือก next-hop ที่จะส่งข้อมูลให้ถึงปลายทาง ค่า cost ที่น้อยจะถูกเลือกให้มีลำดับความสำคัญสูงกว่า

qualified-next-hop: ใช้สำหรับระบุเส้นทางอื่นๆ ไว้ในกรณีที่เส้นทางหลักไม่สามารถใช้งานได้ ซึ่งค่าของ cost จะมีความสำคัญต่ำกว่าเส้นทางหลัก ค่าเริ่มต้นคือ 10

mrib-route: ระบุเส้นทางของมัลติคาสท์ซึ่งจะเก็บใน MRIB ค่าที่ใช้กำหนดคือ address/prefix-length เส้นทางที่อยู่ใน MRIB นี้จะไม่ใช้เส้นทางที่เราท์เตอร์จะทำการ forward ข้อมูลไปทันที มันจะถูกใช้โดยโพรโทคอล PIM-SM อีกทีหนึ่ง

interface-route: ระบุเส้นทางการ forward ข้อมูลในยูนิคาสท์ สำหรับตัวแปรนี้นิยมใช้กับเครือข่ายแบบ wireless

next-hop-interface: ระบุถึง next-hop ของอินเทอร์เฟสที่จะส่งข้อมูลต่อไปยังปลายทาง

next-hop-vif: ระบุถึง next-hop ของ vif อินเทอร์เฟสที่จะส่งข้อมูลต่อไปยังปลายทาง

next-hop-router: ระบุถึง next-hop ของเราท์เตอร์ที่จะส่งข้อมูลต่อไปยังปลายทาง ค่าเริ่มต้นคือ 0.0.0.0

metric: ค่า cost ในการเดินทางของข้อมูล

qualified-next-hop-interface: ใช้สำหรับระบุอินเทอร์เฟสอื่นๆ ไว้ในกรณีที่อินเทอร์เฟสหลักไม่สามารถใช้งานได้

qualified-next-hop-vif: บุถึง next-hop สำรองของ vif อินเทอร์เฟสที่จะส่งข้อมูลต่อไปยังปลายทาง

next-hop-router: ระบุถึง next-hop ของเราท์เตอร์สำรองที่จะส่งข้อมูลต่อไปยังปลายทาง

mrib-interface-route เป็นการระบุเส้นทางของโพรโทคอลมัลติคาสท์และมี parameter เหมือนกับยูนิคาสท์แต่แตกต่างกันที่เก็บลงบนฐานข้อมูล MRIB แทน RIB

Example Static Configurations

```
protocols {
  static {
    route 10.20.0.0/16 {
      next-hop: 10.10.10.20
      metric: 1
      qualified-next-hop 172.17.0.2 {
        metric: 10
      }
    }
    route 2001:DB8:AAAA:20::/64 {
      next-hop: 2001:DB8:10:10:10:10:10:20
      metric: 1
    }
    mrib-route 10.20.0.0/16 {
      next-hop: 10.10.10.30
      metric: 1
    }
    mrib-route 2001:DB8:AAAA:20::/64 {
      next-hop: 2001:DB8:10:10:10:10:10:30
      metric: 1
    }
```

```
    interface-route 10.30.0.0/16 {
      next-hop-interface: rl0
      next-hop-vif: rl0
      metric: 1
      qualified-next-hop-interface rl1 {
        qualified-next-hop-vif rl1 {
          metric: 10
        }
      }
    }
    interface-route 2001:DB8:AAAA:30::/64 {
      next-hop-interface: rl0
      next-hop-vif: rl0
      metric: 1
    }
    mrib-interface-route 10.30.0.0/16 {
      next-hop-interface: rl1
      next-hop-vif: rl1
      metric: 1
    }
    mrib-interface-route 2001:DB8:AAAA:30::/64 {
      next-hop-interface: rl1
      next-hop-vif: rl1
      metric: 1
    }
  }
}
```

13.6.2.2 การตรวจสอบ static route

การแสดงผลการทำงานของ static route แบบยูนิคาสท์บน IPv4 ใช้คำสั่ง show route table ipv4 unicast static

```
user@hostname> show route table ipv4 unicast static
192.168.0.0/24          [static(1)/1]
                         > to 192.150.187.1 via fxp0/fxp0
```

```
192.168.1.0/24          [static(1)/1]
                         > to 192.150.187.2 via fxp0/fxp0
```

ในการแสดงผลด้วยคำสั่ง show ของ static route ข้างต้น แสดงรายละเอียดของ network (192.168.0.0/24), nexthop (192.150.187.1), matric (/1) และอินเทอร์เฟสที่จะส่งข้อมูลออกไปด้วย (via fxp0/fxp0) ถ้าสมมุติว่าไม่ได้ทำการกำหนด next-hop ก็จะไม่แสดงอินเทอร์เฟสที่จะส่งข้อมูลตามไปด้วยเช่นกัน สำหรับ IPv6 ยูนิคาสท์ใช้คำสั่ง show route table ipv6 unicast static และมัลติคาสท์ใช้คำสั่ง show route table ipv4 multicast static สำหรับ IPv4 และ show route table ipv6 multicast static สำหรับ IPv6

13.6.3 RIP and RIPng [72, 73]

Routing Information Protocol (RIP) เป็นโพรโทคอลค้นหาเส้นทางแบบ dynamic ชนิดยูนิคาสท์ โพรโทคอลถูกออกแบบและคอนฟิกให้ทำงานแบบง่ายๆ ใช้กับเครือข่ายขนาดเล็กและไม่มีความซับซ้อน ปัจจุบัน RIP มี 2 เวอร์ชันคือ RIPv1 และ RIPv2 ในเวอร์ชันที่ 1 ไม่รองรับการทำ classless addressing บนเราท์เตอร์ XORP จึงไม่บรรจุ RIPv1 ไว้เนื่องจากไม่เหมาะสมที่จะใช้งานในปัจจุบันแล้ว สำหรับ RIPv2 อนุญาตให้สามารถทำ classless addressing ได้ XORP รองรับการทำงานของ RIPv2 ได้อย่างสมบูรณ์ และ RIPng เป็น RIPv2 ซึ่งจะใช้กับ IPv6 เท่านั้น

คุณลักษณะการทำงานของ RIP มีดังต่อไปนี้

- RIP อาศัย ค่าของจำนวน Hop เป็นหลัก เพื่อการเลือกเส้นทาง โดยจำกัดที่ไม่เกิน 15 Hop
- RIP จะส่งข่าวสารเกี่ยวกับการปรับปรุงเส้นทางออกไปทุก 30 วินาที
- การส่งข้อมูลเกี่ยวกับการปรับปรุงตารางเส้นทาง เป็นการส่งออกไปทั้งหมดของตารางทั้งตารางเก่าและตารางใหม่
- การส่งข่าวสารเกี่ยวกับการปรับปรุงเส้นทาง จะเกิดขึ้นกับเราท์เตอร์ที่เชื่อมต่อกันโดยตรงเท่านั้น

RIP เป็นโพรโทคอลที่ไม่สามารถมองภาพรวมทั้งหมดของระบบเครือข่ายได้ การทำงานจะอาศัยข้อมูลข่าวสารที่ได้ รับจากเราท์เตอร์เพื่อนบ้าน ว่ามีเราท์เตอร์ที่เชื่อมต่อกันอยู่กี่ตัว และเชื่อมต่อที่ใดบ้าง ด้วยเหตุนี้ เมื่อใดที่เกิดการเปลี่ยนแปลงของเครือข่ายเกิดขึ้น เราท์เตอร์จะไม่ได้รับการปรับปรุงข่าวสารนี้โดยทันทีพร้อมกันทุกตัว ทำให้ได้รับข้อมูลข่าวสารล่าช้า ดังนั้นเราท์เตอร์ส่วนใหญ่ยังเข้าใจว่า การเปลี่ยนแปลงบนเครือข่ายยังไม่เกิดขึ้น แต่เราท์เตอร์เหล่านี้ ยังใช้ข้อมูลเดิมแลกเปลี่ยนเส้นทางเก่าให้แก่กัน ทำให้เกิดปัญหาที่เรียกว่า Routing Loop โดย Routing Loop เป็นเรื่องของ Packet ที่วิ่งกลับไปกลับมาระหว่าง Router 2 ตัวหรือมากกว่า ทำให้ไม่สามารถสื่อสารข้อมูลกันได้ในที่สุด เพื่อหลีกเลี่ยงปัญหานี้ RIP จะใช้กลไกการทำงานหลายประการ ดังนี้

- Count To Infinity RIP จะยอมให้มีจำนวนของ Hop ทั้งหมดบนเส้นทางไม่เกิน 15 Hop เครือข่ายใดที่อยู่บนเส้นทางที่มีจำนวน Hop มากกว่า 15 Hop จะไม่สามารถไปถึงได้
- Split Horizon เราท์เตอร์จะต้องไม่ประกาศเส้นทาง กลับไปยังเราท์เตอร์ที่ส่งตารางปรับปรุงมาให้ หมายความว่าเราท์เตอร์จะต้องไม่เอาข้อมูลที่ได้รับจากเราท์เตอร์ต้นทาง ส่งกลับไปให้เราท์เตอร์ ต้นทาง
- Poison Reverse Update มีการทำงานที่คล้ายคลึงกับ Split Horizon โดยการตั้งค่า Count จาก 16 เป็น Infinity และประกาศค่าของเส้นทางปรับปรุงแบบย้อนกลับอย่างรวดเร็ว อาจทำให้กระบวนการของ Loop ถูกตัดขาด
- Hold Down Counter มีตัวตั้งเวลาอยู่ตัวหนึ่ง เรียกว่า Hold Down Timer เป็นตัวตั้งเวลาที่ มีไว้เพื่อป้องกันมิให้เราท์เตอร์ทำการรับข้อมูลข่าวสารเกี่ยวกับเส้นทางในช่วงเวลาสั้นๆ หลังจากที่เพิ่งได้เคลียร์เส้นทางที่เปลี่ยนแปลงแล้วออกจากตาราง แนวคิดการใช้ Hold Down คือการทำให้แน่ใจว่าเราท์เตอร์ทุกตัวได้รับข่าวสารเกี่ยวกับเส้นทาง ให้เรียบร้อยเสียก่อน อย่าเพิ่งทำการส่งข่าวสารเกี่ยวกับเส้นทางที่ไม่ถูกต้องออกมา
- Triggered Updates ส่งข้อมูลการข่าวสารเมื่อเน็ตเวิร์คมีการเปลี่ยนแปลงเท่านั้น

13.6.3.1 การคอนฟิก RIP

การสั่งให้ RIP ทำงานจำเป็นต้องมีอินเทอร์เฟสรองรับเสมอ โดยแต่ละอินเทอร์เฟสจะต้องมีการกำหนดหมายเลขไอพีที่ไม่ซ้ำกัน ในการใช้งาน RIP จำเป็นต้องมีการประกาศเส้นทางด้วยคำสั่ง export

RIP Configuration Syntax

```
protocols {
  rip {
    targetname: text
    export: text
    interface text {
      vif text {
        address IPv4 {
          metric: uint
          horizon: text
          disable: bool
          passive: bool
          accept-non-rip-requests: bool
```

```
            accept-default-route: bool
            route-timeout: uint
            deletion-delay: uint
            triggered-delay: uint
            triggered-jitter: uint(0..100)
            update-interval: uint
            update-jitter: uint(0..100)
            request-interval: uint
            interpacket-delay: uint
            authentication {
              simple-password: text
              md5 uint(0..255) {
                password: text
                start-time: text("YYYY-MM-DD.HH:MM")
                end-time: text("YYYY-MM-DD.HH:MM")
              }
            }
          }
        }
      }
    traceoptions {
      flag {
        all {
          disable: bool
          }
        }
      }
    }
  }
```

protocols: เป็นขอบเขตสำหรับระบุโพรโทคอล routing

rip: เป็นส่วนใช้สำหรับกำหนดโพรโทคอลแบบ RIP

targetname: เป็นส่วนที่ใช้กำหนดชื่อให้กับโพรโทคอล RIP ค่าเริ่มต้นจะใช้ชื่อว่า “rip”

export: เป็นคำสั่งที่บอกให้เราท์เตอร์ประกาศข้อมูลให้สามารถใช้งานร่วมกัน คล้ายการประกาศตัวแปรแบบ global(รายละเอียดอ่านเพิ่มเติมในส่วนของการสร้าง policy)
interface: ระบุอินเทอร์เฟสที่จะใช้สำหรับรันโพรโทคอล RIP
vif: อินเทอร์เฟสเสมือน ในอินเทอร์เฟสหนึ่งๆ สามารถสร้างอินเทอร์เฟสเสมือนได้หลายๆ ตัว
address: เป็นส่วนที่ใช้ประกาศหมายเลขไอพีแบบ v4 ให้กับแต่ละอินเทอร์เฟสที่ต้องการรัน RIP
metric: เป็นตัวเลขที่ใช้คำนวณ cost เพื่อใช้ในการเลือกเส้นทางที่ดีที่สุด ตัวเลขดังกล่าวจะมีค่าระหว่าง 1-15 ผลรวมของค่า cost ไม่ควรเกิน 15 เนื่องจากเป็นข้อกำหนดของ RIP
horizon: เป็นส่วนที่ใช้กำหนดวิธีการจัดการกับลิงค์ที่ fail ลงไปให้เร็วที่สุด โดยมี option ให้เลือก 3 แบบคือ split-horizon-poison-reverse, split-horizon, none ค่าเริ่มต้นคือ split-horizon-poison-reverse
disable: กำหนดให้ RIP ของแต่ละอินเทอร์เฟสทำงาน
passive: เป็นส่วนที่บอกให้ RIP ทำงานภายใต้โหมด passive การทำงานของโหมด passive คือเราท์เตอร์จะรับเส้นทางเข้ามาทางอินเทอร์เฟสนี้ แต่จะไม่ประกาศเส้นทางใดๆ ออกไปทางอินเทอร์เฟสเดียวกันกับที่รับเข้ามา ค่าเริ่มต้นเป็น false หมายถึงรับเส้นทางและประกาศที่อินเทอร์เฟสเดียวกัน
accept-non-rip-requests: โดยปกติ RIPv2 จะทำการร้องขอการ update เส้นทาง แบบมัลติคาสท์ไปยังทุกๆ พอร์ตที่รัน RIP อยู่ แต่เมื่อต้องการให้มีการ update เส้นทาง แบบยูนิคาสท์ ชนิด UDP และสามารถเลือกให้ update เส้นทางกับพอร์ตที่ไม่ได้รัน RIP ได้ด้วย โดยการกำหนด accept-non-rip-requests ให้เป็น true
accept-default-route: อนุญาตให้ RIP รับค่า default route จากเราท์เตอร์เพื่อบ้านได้ ค่าเริ่มต้นคือ false
route-timeout: RIP กำหนดอายุของการคงอยู่ใน Table ว่าเส้นทางใดก็ตาม ถ้าไม่มีการรับ Message เพื่อมา Update ใน Table เป็นเวลา 180 วินาที จะถือว่าเส้นทางนั้นใช้ไม่ได้ แล้วจะปรับค่า Metric ของเส้นทางนั้นเป็น 16 และจะคงค่า Metric นี้ไว้อีก 120 วินาที ถ้าภายในเวลา 120 วินาทีนี้ยังไม่มี Message มา Update จะทำการลบเส้นทางนั้นออกจาก Routing Table ของตัวเอง) ค่าเวลา default คือ 180 วินาที
deletion-delay: เมื่อที่เส้นทางไม่มีการ update นานเกินกว่าเวลาที่กำหนดไว้ จะเรียกว่า route expire เราท์เตอร์จะต้องทำการลบตารางเส้นทางทิ้ง แต่ก่อนที่จะลบจะต้องรออยู่ช่วงหนึ่งก่อนว่าจะไม่มีเราท์เตอร์เพื่อนบ้านมา update ข้อมูลใหม่เข้ามา ระยะเวลา deletion-delay จะต้องไว้ที่ 120 วินาที
triggered-delay: เมื่อเราท์เตอร์ได้รับตารางเส้นทางใหม่มาจากเพื่อนบ้านแล้ว มันจะไม่รอให้ถึงเวลาที่ตั้งไว้(ในที่นี้คือ 30 วินาที) แต่มันจะส่งสัญญาณที่เรียกว่า triggered update(ประมาณ 3 วินาที) ออกไปให้เราท์เตอร์เพื่อนบ้านทราบทันทีว่าเส้นทางมีการเปลี่ยนแปลง หลังจากที่สัญญาณนี้ส่งออกไปเราท์เตอร์จะทำการตั้งเวลาซึ่งเกิดจาการสุ่มไว้ช่วงระยะเวลาหนึ่ง เพื่อเฝ้าดูต่อไปว่าช่วงเวลา

ดังกล่าวเส้นทางจะมีการเปลี่ยนแปลงอีกหรือไม่ ถ้ามีการเปลี่ยน สัญญาณ triggered update ก็ถือว่าเป็นโมฆะทันที สำหรับสูตรในการหาค่าการสุ่มเวลาหาได้จาก (triggered-delay - triggered-delay * triggered-jitter / 100) และ (triggered-delay+ triggered-delay * triggered-jitter / 100)

triggered-jitter: คล้ายกับ triggered-delay ค่าเริ่มต้นควรอยู่ในช่วง 1-5 วินาที

update-interval: RIP จะส่งตารางเส้นทางไปให้กับเราท์เพื่อนบ้านทุก ๆ 30 วินาที

update-jitter:

request-interval: ถ้าอินเทอร์เฟสใดๆ ของ RIP ไม่มีข้อมูลจากเพื่อนบ้าน เราท์เตอร์อาจจะจำเป็นต้องส่งสัญญาณไปถามว่ายังมีตัวตนอยู่หรือไม่ ค่าเริ่มต้นคือ 30 วินาที เมื่อกำหนดเป็น 0 จะไม่มีการส่งสัญญาณดังกล่าวออกไป

interpacket-delay: ใช้กำหนดค่า delay สำหรับการเชื่อมต่อกันแบบ back-to-back ค่าเริ่มต้นคือ 50 milliseconds

authentication: กำหนดวิธีการยืนยันตัวตนของเราท์เตอร์ก่อนมีการ update ข้อมูลกัน

simple-password: การหนดยืนยันตัวตนโดยใช้รหัสผ่านชนิด plaintext

md5: การหนดยืนยันตัวตนโดยใช้ MD5

start-time: กำหนดเวลาเริ่มต้นเมื่อ Key ของ MD5 เริ่มทำงาน มีรูปแบบคือ “YYYY-MM-DD.HH:MM”

end-time: กำหนดเวลาสิ้นสุดเมื่อ Key ของ MD5 หยุดทำงาน มีรูปแบบคือ “YYYY-MM-DD.HH:MM”

traceoptions: ใช้กำหนดเงื่อนไขในการตรวจสอบโพรโทคอล RIP

flag: เป็นเงื่อนไขการตรวจสอบโดยใช้ flag

all: ตรวจสอบค่าที่เป็นไปได้ทั้งหมดของ RIP

disable: เปิดหรือเปิดการตรวจสอบ ค่าเริ่มต้นเป็น false

13.6.3.2 การคอนฟิก RIPng

การคอนฟิก RIPng จะเหมือนกับ RIP ธรรมดา แต่มีข้อแตกต่าง 2 จุดคือ

- ขนาดความยาวของไอพีแอดเดรส IPv4 และ IPv6 ต่างกัน
- IPv6 ไม่มีระบบการยืนยันตัวตน เพราะ RFC 2081 ไม่ได้ระบุเอาไว้

Example RIP Configurations

```
policy {
  policy-statement connected-to-rip {
    term export {
      from {
        protocol: "connected"
```

```
        }
        then {
            metric: 0
        }
    }
  }
}
policy {
  policy-statement static-to-rip {
    term export {
        from {
            protocol: "static"
        }
        then {
             metric: 1
        }
    }
  }
}
protocols {
  rip {
    /* Redistribute connected and static routes */
    export: "connected-to-rip,static-to-rip"
    /* Run on specified network interface addresses */
        interface fxp0 {
            vif fxp0 {
                address 69.110.224.158 {
            }
        }
    }
  }
}
```

จากตัวอย่างการคอนฟิกข้างบน RIP ทำการ export เส้นทางไปให้กับเน็ตเวิร์คที่เชื่อมกับตัวมันเองโดยตรงและ ที่ใช้งานผ่าน static route ด้วย ค่าของ metric ถูกกำหนดให้มีค่าเป็น 0 เมื่อเน็ตเวิร์คเชื่อมต่อกับตัวเราท์เตอร์โดยตรง และเป็น 1 เมื่อใช้การเราท์แบบ static ประกาศให้อินเทอร์เฟส fxp0 มีหมายเลขไอพีเป็น 69.110.224.158 ใช้สำหรับส่งและรับการ update กันระหว่างเราท์เตอร์เพื่อนบ้าน

Monitoring RIP

สำหรับคำสั่งที่ใช้ตรวจสอบการทำงานของ RIP แบบยูนิคาสท์ ใน IPv4 จะใช้คำสั่ง show route table ipv4 unicast rip IPv6 ใช้คำสั่ง show route table ipv6 unicast rip ส่วนคำสั่งที่ใช้ตรวจสอบการทำงานของ RIP แบบมัลติคาสท์ ใน IPv4 จะใช้คำสั่ง show route table ipv4 multicast rip และ IPv6 ใช้คำสั่ง show route table ipv6 multicast rip

```
user@hostname> show route table ipv4 unicast rip
172.13.0.0/24          [rip(120)/1]
                          > to 172.13.0.1 via dc0/dc0
172.13.1.0/24          [rip(120)/1]
                          > to 172.13.1.1 via dc1/dc1
172.13.2.0/24          [rip(120)/1]
                          > to 172.13.2.1 via dc2/dc2
172.13.3.0/24          [rip(120)/1]
                          > to 172.13.3.1 via dc3/dc3
192.150.187.0/25    [rip(120)/1]
                          > to 192.150.187.112 via fxp0/fxp0
```

ภาคที่สี่

การติดตั้ง ปรับแต่ง และวิเคราะห์ระบบเครือข่าย

(สื่อมัลติมีเดีย)

บทที่ 14

Workshop on Packet Tracer Simulation (Video Training)

แนวคิด

ในบทนี้จะมาเรียนรู้วิธีการเชื่อมต่อระบบเครือข่ายแบบต่างๆ ผ่านโปรแกรมจำลองเครือข่าย โดยผู้เขียนได้สาธิตและบันทึกอยู่ในรูปแบบของไฟล์วิดีโอ เพื่อให้ผู้เรียนสามารถเรียนรู้ได้สะดวก สามารถทบทวนเนื้อหาได้ทุกเวลาที่ต้องการ

วัตถุประสงค์

1. นิสิตสามารถทบทวนเนื้อหาได้ตลอดเวลา
2. นิสิตสามารถเรียนรู้ได้ตามที่ตัวเองต้องการ

ในบทนี้จะอธิบายถึงวิธีการออกแบบและติดตั้งเครือข่ายโดยใช้โปรแกรม Packet ในรูปแบบของวีดิโอ ซึ่งผู้อ่านสามารถเลือก Workshop ที่ต้องการโดยการอ่านคำอธิบายสั้นๆ ได้จากบทนี้ก่อน จากนั้นจึงทำการเปิด Video ในแผ่น DVD ที่แนบมากับหนังสือเล่มนี้ โดยผู้ใช้จำเป็นต้องใช้เครื่องมือเหล่านี้คือ

1. โปรแกรม Packet Tracer ตั้งแต่เวอร์ชัน 5.3 ขึ้นไป
2. โปรแกรมที่สามารถเล่นไฟล์ประเภทมัลติมิเดียเหล่านี้ได้ (.avi, .flv)
3. ลำโพง

หมายเหตุ ใน Workshop ตั้งแต่ 1-36 ยังไม่ครอบคลุมคุณสมบัติทั้งหมดของโปรแกรม Packet Tracer ที่มีให้ เนื่องจากข้อมูลที่บันทึกอยู่ในรูปของวีดิโอนั้นค่อนข้างใหญ่ ซึ่งถ้าอธิบายให้ครบทุกเนื้อหา จะต้องใช้ DVD หลายแผ่น ผู้เขียนหวังเป็นอย่างยิ่งว่าอาจจะมีโอกาสบันทึกเนื้อหาที่ยังขาดอยู่ในหนังสือเล่มถัดไป

Workshop 0: เบื้องต้นก่อนคอนฟิก

คำกล่าวนำเบื้องต้นเกี่ยวกับหนังสือของผู้เขียน

Workshop 1: Introduction to Packet Tracer 5.3

เป็นการแนะนำ Packet Tracer เบื้องต้น เพิ่มเติมจากที่อธิบายแล้วในหนังสือ

Workshop 2: How to Packet Tracer 5.3

เริ่มต้นการใช้งานใช้งานโปรแกรม Packet Tracer

Workshop 3: การเพิ่ม/ลด อุปกรณ์

แสดงตัวอย่างการเพิ่มลดอุปกรณ์ เช่น เพิ่มอุปกรณ์ การ์ดอินเทอร์เฟสให้เราเตอร์ การเชื่อมสายสัญญาณ เป็นต้น

Workshop 4: การใช้คำสัง IOS เบื้องต้น

อธิบายการใช้คำสั่งระบบปฏิบัติการของ Cisco คือ IOS ว่ามีหลักการทำงานอย่างไร มีหมวดอะไรบ้าง แต่ละหมวดทำงานอย่างไร เป็นต้น

Workshop 5: การเริ่มต้นคอนฟิกอุปกรณ์โดยผ่าน console

สาธิตการปรับแต่งค่า Console เพื่อให้ผู้ใช้งานเบื้องต้นสามารถเข้าไป config อุปกรณ์ได้โดยผ่านทาง console (ซึ่งมีความจำเป็นมาก)

Workshop 6: การคอนฟิกอุปกรณ์โดยผ่านโพรโทคอล Telnet

สาธิตการปรับแต่งเมื่อผู้ใช้ต้องการ Remote เข้าไป config อุปกรณ์เราเตอร์ หรือสวิชต์ผ่านเครือข่ายเน็ตเวิร์ค เพื่อช่วยประหยัดเวลาในการทำงาน และควบคุมเครือข่ายจากจุดเดียว

Workshop 7: การคอนฟิกอุปกรณ์โดยผ่านโพรโทคอล Secure Shell

สาธิตการปรับแต่งเมื่อผู้ใช้ต้องการ Remote แบบปลอดภัยเข้าไป config อุปกรณ์เครือข่าย

Workshop 8: การคอนฟิก loopback interface

สาธิตการสร้างเครือข่ายที่ใช้สำหรับทดสอบ โดยไม่จำเป็นต้องใช้อุปกรณ์ end device เพิ่มเติม ช่วยลดงบประมาณ ประหยัดเวลา และสะดวก

Workshop 9: การคอนฟิก vlan บน switch L2

สาธิตการสร้างเครือข่ายเสมือนบนอุปกรณ์สวิชต์ระดับเลเยอร์ที่ 2

Workshop 10: การคอนฟิก IP & Backup & Restore Config/IOS guration file บน Switch L2

สาธิตการสำรองข้อมูล configuration file, IOS operating system เพื่อใช้ในกรณีฉุกเฉิน

Workshop 11: การคอนฟิก vlan บน swith L3

สาธิตการสร้าง VLAN บนอุปกรณ์สวิชต์ L3 (3560) ว่ามีขั้นตอนอย่างไร

Workshop 12: การคอนฟิก static route บนเราเตอร์

สาธิตการคอนฟิก static route บนเราเตอร์ว่ามีขั้นตอนอย่างไร ซึ่งจะใช้เป็นพื้นฐานในการสร้างเครือข่ายที่มีความซับซ้อนต่อไป

Workshop 13: การคอนฟิก static route บน Switch L3

สาธิตการคอนฟิก static route บนสวิชต์ว่ามีขั้นตอนอย่างไร

Workshop 14: การคอนฟิก static route ระหว่าง Router และ Switch L3

สาธิตการคอนฟิก static route ระหว่าง Router และ Switch L3 ให้สามารถส่งข้อมูลกันได้ เนื่องจากบางครั้งหน่วยงานหรือองค์ อาจจะใช้ Router ทำหน้าที่เป็น gateway เชื่อมต่ออินเทอร์เน็ต และใช้ Switch L3 ควบคุมเครือข่ายภายใน จึงจำเป็นต้อง เข้าใจการเชื่อมต่อระหว่างอุปกรณ์ทั้ง 2 ตัวเข้าด้วยกัน

Workshop 15: การเชื่อมต่อ Router ด้วยสาย Serial Interface

สาธิตการเชื่อมต่อและคอนฟิกเราเตอร์ด้วยสายชนิด Serial เนื่องจากเป็นสายที่พิเศษกว่าแบบอื่นๆ คือ สามารถกำหนดแบนด์วิดท์ ได้เอง โดยการกำหนด Clock Rate

Workshop 16: การเชื่อมต่อ Network บนโปรแกรม Packet Tracer เข้าด้วยกันโดยผ่าน Cloud (Multiuser)

สาธิตการเชื่อมต่อเครือข่ายที่ทำงานอยู่บนโปรแกรม Packet Tracer ที่อยู่ต่างที่กันให้สามารถเชื่อมต่อเป็นเครือข่ายที่มีขนาดใหญ่ได้ (เสมือนเป็นเครือข่ายอินเทอร์เน็ตซ้อนอยู่บนเครือข่ายอินเทอร์เน็ตจริงอีกชั้นหนึ่ง)

Workshop 17: การคอนฟิก Dynamic Routing (RIPv2)

สาธิตการเชื่อมต่อเครือข่ายด้วยโพรโทคอล RIP ซึ่งเป็นโพรโทคอลพื้นฐานที่สำคัญที่ใช้กับเครือข่ายภายใน หรือ Intranet

Workshop 18: การคอนฟิก Dynamic Routing OSPF(Single Area)

สาธิตการเชื่อมต่อเครือข่ายด้วยโพรโทคอล OSPF ซึ่งเป็นโพรโทคอลพื้นฐานที่สำคัญมากที่ใช้กับเครือข่ายภายใน หรือ Intranet โดยใช้ Area เดียว

Workshop 19: การคอนฟิก Dynamic Routing OSPF(Multiple Area)

สาธิตการเชื่อมต่อเครือข่ายด้วยโพรโทคอล OSPF โดยใช้หลาย Area

Workshop 20: การคอนฟิก OSPF Authentication

สาธิตการเชื่อมต่อเครือข่ายด้วยโพรโทคอล OSPF โดยมีการยืนยันตัวตนของอุปกรณ์เพื่อป้องการถูก Hack

Workshop 21: การคอนฟิก Dynamic Routing EIGRP

สาธิตการเชื่อมต่อเครือข่ายด้วยโพรโทคอล EIGRP ซึ่งเป็นโพรโทคอลแบบ Dynamic ของ Cisco

Workshop 22: การคอนฟิก DHCP ข้ามเครือข่ายด้วย IP Helper

สาธิตการสร้าง DHCP Server และให้สามารถ forward ข้อมูลของโพรโทคอล DHCP ข้ามเครือข่ายได้ เพราะในสถานะการณ์จริงองค์กรต่างๆ นิยมติดตั้ง DHCP Server บน เน็ตเวิร์คใดเน็ตเวิร์คหนึ่ง ส่งผลให้ไม่สามารถแจกไอพีข้าม VLAN ได้

Workshop 23: การคอนฟิกให้เราเตอร์ทำหน้าที่เป็น DHCP Server

สาธิตการสร้าง DHCP Server บนอุปกรณ์ Router เอง เนื่องจากได้เปรียบทางด้านความเร็วและการกระจายความผิดพลาดของ DHCP Server

Workshop 24: การคอนฟิก OSPF กับ EIGRP โดยใช้ Redistribution

สาธิตการสร้างเครือข่ายที่มีโพรโทคอลตั้งแต่ 2 ชนิดขึ้นไปทำงานอยู่ด้วยกัน ใน workshop นี้จะสาธิตการทำงานของ OSPF ให้ทำงานร่วมกับ EIGRP

Workshop 25: การคอนฟิก Standard ACL (1)

อธิบายหลักการทำงาน สาธิตการสร้างไฟร์วอลล์บนเราเตอร์ เรียกว่า ACL โดยเป็น ACL ชนิดทั่วๆ ไป (Standard) สามารถตรวจสอบการทำงานได้เฉพาะเลเยอร์ที่ 3 ของ OSI Model

Workshop 26: การคอนฟิก Standard ACL (2)

สาธิตการสร้าง ACL แบบ Standard และมีการอธิบายรายละเอียดการใช้งานที่เพิ่มขึ้น

Workshop 27: การคอนฟิก Standard ACL (3)

สาธิตการสร้าง ACL แบบ Standard เพิ่มขึ้น

Workshop 28: การคอนฟิก Extended ACL (1)

อธิบายหลักการทำงาน สาธิตการ ACL โดยเป็น ACL ชนิดพิเศษ (Extended) สามารถตรวจสอบการทำงานได้เฉพาะเลเยอร์ที่ 3, 4 ของ OSI Model

Workshop 29: การคอนฟิก Extended ACL (2)

สาธิตการสร้าง ACL แบบ Extened เพิ่มขึ้น

Workshop 30: การคอนฟิก Extended ACL (3)

สาธิตการสร้าง ACL แบบ Extened อย่างละเอียด และประยุกต์ใช้งานจริง

Workshop 31: การคอนฟิก Link สำรอง โดยใช้ Floating Static Route

สาธิตการสร้าง link สำหรับ เพื่อใช้สำหรับกรณีที่ เครือข่ายหลักไม่สามารถใช้งานได้

Workshop 32: การคอนฟิก Static NAT

สาธิตการคอนฟิก NAT แบบ Static ซึ่งเป็นที่นิยมใช้งานในปัจจุบัน เพื่อให้สามารถเชื่อมต่อเครือข่ายอินเทอร์เน็ตได้

Workshop 33: การคอนฟิก Static/Dynamic NAT ร่วมกับ ACL

สาธิตการประยุกต์ใช้งานในเครือข่ายที่ทำงานจริงในปัจจุบัน คือ การผสมผสานระหว่าง Dynamic NAT + Static และ ACL เข้าด้วยกัน

Workshop 34: การคอนฟิก Switch L3 ร่วมกับ Router

สาธิตการประยุกต์ใช้งานอุปกรณ์สวิชต์ L3 ทำหน้าที่เป็น Backbone network ในองกรณ์ และใช้ Router ทำหน้าที่เชื่อมต่อเป็น gateway กับ ISP

Workshop 35: การคอนฟิก BGP เบื้องต้น

สาธิตการคอนฟิกโพรโทคอล BGP ซึ่งเป็นโพรโทคอลที่ใช้สำหรับเชื่อมต่อเครือข่าย WAN ในปัจจุบัน

Workshop 36: การใช้งาน Activity Wizard

สาธิตการใช้งาน Activity Wizard ซึ่งมีประโยชน์อย่างมากสำหรับอาจารย์หรือคุณครูที่สอนด้านระบบเครือข่าย คุณสมบัตินี้จะช่วยให้สร้างบนเรียนหรือแบบทดสอบการสร้างเครือข่ายแบบ step-by-step ทำให้ผู้เรียนเข้าใจเครือข่ายได้อย่างรวดเร็ว และอาจารย์ก็สามารถสร้างข้อสอบได้อย่างมีประสิทธิภาพ

หมายเหตุ: สุดท้ายนี้ผู้เขียนหวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้ น่าจะมีประโยชน์ไม่มากก็น้อยสำหรับ ผู้ที่ชื่นชอบระบบเครือข่าย สำหรับเนื้อหาทั้งหมดยังไม่ครอบคลุมระบบเครือข่ายทั้งหมด ซึ่งหวังอีกครั้งว่าอาจจะมีเล่มที่ 3 ออกมาให้ผู้สนใจทุกท่านได้อ่านกันอีก (ถ้ามีโอกาส) สวัสดีครับ

สุชาติ คุ้มมะณี

suchart.k@msu.ac.th

เอกสารอ้างอิง

1. Cisco Networking Academy Program (CCAI). Cisco Networking Academy Program: Cisco Press; April 7, 2005.
2. Inc. Cisco Systems. **Internetworking Technology Handbook** In; 1992-2006
4. Inc. Cisco Systems. [cited]; Available from: wwww.cisco.com
5. Inc. Cisco Systems. **Cisco Networking Academy Program (CCAI)**. [cited; Available from: http://www.cisco.com/en/US/learning/netacad/academy/index.html
6. Behrouz A. Forouzan. **Data Communications and Networking** 3ed: McGraw-Hill; 2003.
7. William A. Shay. **Understanding Data Communications and Networks**. 3 ed: Course Technology; 2003.
9. ดร. วิรินทร์ เมฆประดิษฐ์สิน. คัมภีร์ระบบเครือข่าย ซีเอ็ดยูเคชั่น จำกัด (มหาชน); 2547.
10. เรื่องไกร รังสิพล. เจาะระบบ **TCP/IP** โปรวิชั่น; 2537.
11. เอกสิทธิ์ วิริยจารี. เรียนรู้ระบบเน็ตเวิร์คจากอุปกรณ์ของ **Cisco** ซีเอ็ดยูเคชั่น จำกัด (มหาชน); 2548.
12. ลัญฉการ วุฒิสิทธิกุลกิจ. โครงข่ายอินเทอร์เน็ตและโพรโทคอลทีซีพี/ไอพี สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย; 2545
13. สุชาติ คุ้มมะณี, ธวัชชัยชมศิริ. เรียนรู้ระบบเครือข่ายและอุปกรณ์ cisco ด้วยโปรแกรมจำลองเครือข่าย สำนักพิมพ์โปรวิชัน; กรุงเทพฯ;2551
14. วิกิพีเดีย สารานุกรมเสรี; **เราเตอร์**; http://th.wikipedia.org/wiki/เราเตอร์; กรกฎาคม 2552
15. **Open source software**; http://stkswiki.pbworks.com/open-source-software
16. สุชาติ คุ้มมะณี; **เรียนรู้เครือข่ายและอุปกรณ์ Cisco ด้วยโปรแกรม Simulation**; กรุงเทพฯ : ซีเอ็ดยูเคชั่น , 2550
17. Cisco Systems, Inc.; **Cisco 2600 Series Router Architecture** http://www.cisco.com/en/US/products/hw/routers/ps259/products_tech_note 09186a0080 094e92.shtml; 2009
18. XORP; **eXtensible Open Router Platform**; http://www.xorp.org/; January 2009
19. Quagga; **Quagga Routing Suite**; http://www.quagga.net/about.php; Jul 2006

20. GNU Zebra; **Free routing software**; http://www.zebra.org/what.html; 2003 IP Infusion Inc.
21. Vyatta Inc.; **Open Networking**; http://www.vyatta.com/; 2008
22. GNU Zebra; **Free routing software distributed**; http://www.zebra.org/spec.html; 2003
23. **วิกิพีเดีย สารานุกรมเสรี;** สถาปัตยกรรมแบบ x86**; http://th.wikipedia.org/wiki/X86; 2009**
24. Intel; **Intel® Core™2 Processor with vPro™ Technology**; http://www.intel.com/products/vpro/index.htm?iid=prod+prod_core2vpro; 2007
25. **Advanced Micro Devices, Inc.**; http://products.amd.com/en-us/DesktopCPUResult.aspx; 2009
26. **Cyrix Corporation**; http://en.wikipedia.org/wiki/Cyrix; August 2009
27. **Vyatta Inc.**; http://www.vyatta.com/downloads/whitepapers.php
28. S. Bradner, J. McQuaid; **Benchmarking Methodology for Network Interconnect Devices**; RFC 2544; 1999; http://tools.ietf.org/html/rfc2544
29. Vyatta; **Hardware Recommendation Guidelines**; http://www.vyatta.com/downloads/whitepapers.php; 2008
30. Wikipedia; **Bus (computing)**; http://en.wikipedia.org/wiki/Bus_(computing); July 2009
31. PCI-SIG; **PCI-X 2.0 Overview**; http://www.pcisig.com/specifications/pcix_20/; 2009
32. Wikipedia; **network interface controller (NIC)**; http://en.wikipedia.org/wiki/Network_card; June 2009
33. Wikipedia; **Direct memory access (DMA)**; http://en.wikipedia.org/wiki/Direct_memory_access; July 2009
34. M. Tim Jones, Consultant Engineer, Emulex; **Linux and symmetric multiprocessing**; http://www.ibm.com/developerworks/library/l-linux-smp/; Mar 2007
35. Vyatta Inc.; **DOCUMENTATION**; http://www.vyatta.com/; 2008
36. **Cisco Systems, Inc.**; http://www.cisco.com/; 2009
37. **Juniper Networks, Inc.**; http://www.juniper.net/us/en/; 2009
38. **Linksysbycisco**; http://www.linksysbycisco.com/APAC/en/home; 2009

39. **D-Link Corporation**; http://www.d-link.com/corporate/worldwideoffices/?redirect=%2fdefault.aspx; 2009

40. Tolly Group; **The Tolly Report**; http://promo.attachmate.com/cs-tolly/index.cfm?gclid=CNCi2dySupwCFcEtpAodZ1Qnug; 2009

41. Vyatta, Inc.; **BGP Performance Evaluation versus Cisco 7204VXR Router**; No.208289; March 2008

42. Vyatta, Inc.; **Competitive Gigabit Ethernet LAN Routing Throughput Evaluation versus Cisco 2821 Integrated Services Router**; No.207190; March 2007

43. VYATTA, INC.; The Structural Cost Efficiencies of Open Source Networking; White Paper; http://www.vyatta.com; 2007

44. Yezid Donoso, **Network Design for IP Convergence**, CRC Press, 2009

45. วิกิพีเดีย สารานุกรมเสรี; **Point-to-Point Protocol (PPP)**; http://th.wikipedia.org/wiki/Point-to-Point_Protocol

46. Wikipedia, the free encyclopedia; **Point-to-Point Protocol over Ethernet (PPOE)**; http://en.wikipedia.org/wiki/Point-to-Point_Protocol_over_Ethernet

47. Wikipedia, the free encyclopedia; **Asymmetric digital subscriber line (ADSL)**; http://en.wikipedia.org/wiki/Asymmetric_digital_subscriber_line

48. http://africanstarline.net/images/dsl_modem_setup_html_m43c3651e.png

49. วิกิพีเดีย สารานุกรมเสรี; **Virtual Local Area Network: VLAN**; http://th.wikipedia.org/wiki/Virtual_LAN; สิงหาคม 2552

50. Wikipedia, the free encyclopedia; **TIA/EIA-568-B**; http://en.wikipedia.org/wiki/T568B#T568A_and_T568B_termination; May 2009

51. Wikipedia, the free encyclopedia; **Leased line**; http://en.wikipedia.org/wiki/Leased_line; August 2009

52. วิกิพีเดีย สารานุกรมเสรี; **Network address translation: NAT;** http://th.wikipedia.org/wiki/Network_address_translation; 2009

53. Wikipedia, the free encyclopedia; **IP address**; http://en.wikipedia.org/wiki/IP_address; August 2009

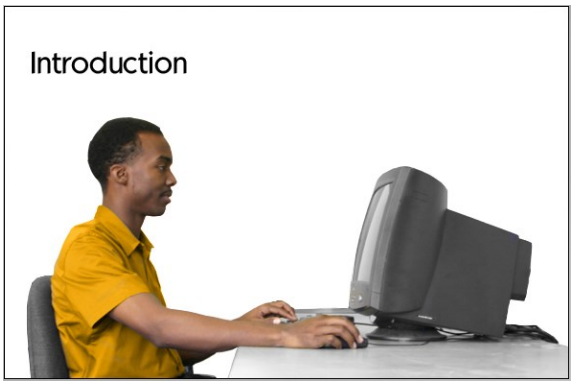
54. Todd Lammle; **Cisco Certified Network Associate Study Guide: Exam 640-802**; John Wiley and Sons, 2007

55. Wikipedia, the free encyclopedia; **Intrusion detection system: IDS**; http://en.wikipedia.org/wiki/Intrusion_detection_system

56. Wikipedia, the free encyclopedia; **Intrusion prevention system : IDP**; http://en.wikipedia.org/wiki/Intrusion_prevention_system
57. Instructions for Windows.; **Burning Iso-Images for FreeBSD 4.x**; http://users.owt.com/kstewart/freebsd/iso-images.html
58. Instructions for Linux; **Burning CDs on Linux**; http://www.ibm.com/developerworks/linux/library/l-cdburn.html
59. Instructions for FreeBSD; **FreeBSD 'burncd' command**; http://www.redantigua.com/burncd.html
60. VMware Inc.; **Vmware Cloud Computing**; http://www.vmware.com/
61. Citrix Systems, Inc.; **The Xen® hypervisor**; http://www.xen.org/
62. Sun Microsystems, Inc.; **Welcome to VirtualBox.org**; http://www.virtualbox.org/
63. Wikipedia, the free encyclopedia; **Category 5 cable**; http://en.wikipedia.org/wiki/Category_5_cable; August 2009
64. Division of Atcom inc.,; **How to Make a Category 6 Patch Cable**; http://www.lanshack.com/make_cat_6_cable.aspx;
65. **Internet Assigned Numbers Authority:IANA**.; http://www.iana.org/
66. Behrouz A. Forouzan; **Data Communications and Networking third edition**
67. **william A. Shay**; Understanding Data Communications and Networks
68. Cisco Networking Academy Program Secound Edition (CCAI)
69. William R. Parkhurst; **Cisco OSPF Route Redistribution**; Cisco Press
70. **Internetworking Technology Handbook**; http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/index.htm
71. Cisco Inc,; http://www.cisco.com
72. IBM Inc.; **OMPROUTED RIP and RIPng concepts**; http://publib.boulder.ibm.com/infocenter/iseries/v6r1m0/index.jsp?topic=/rzal6/rzal6ripngconcepts.htm
73. Internet Engineering Task Force; RIP Version 2; http://www.ietf.org/rfc/rfc2453.txt

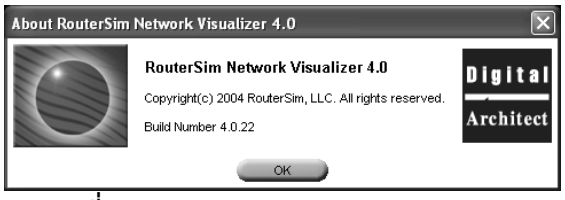
ภาคผนวก

ซอฟต์แวร์จำลองเครือข่าย (Network Simulation)



ซอฟท์แวร์จะลองระบบเครือข่ายที่เราจะทดลองใช้งานนี้มีอยู่หลายค่ายแต่ผู้เขียนได้เลือกมาทดลองในหนังสือนี้ 3 ค่าย ซึ่งเป็นที่นิยมและรู้จักกันในหมู่ของผู้ดูแลระบบว่าเป็นซอฟท์แวร์ที่ใช้งานง่าย มีเครื่องมือให้ใช้งานมากมาย มีการอัพเดทอยู่เสมอ และมีฟังก์ชันการทำงานครบครัน ค่ายแรกเป็นของบริษัท Boson Software ชื่อ Boson Netsim ค่ายที่ 2 ชื่อ RouterSim และค่ายที่ 3 คือ Packet Tracer

รูปที่ 1 Boson Netsim



รูปที่ 2 RouterSim Network Visualizer

ซอฟท์แวร์จำลองเครือข่ายเหล่านี้ จะมีอุปกรณ์ที่ให้เราใช้สำหรับสร้างระบบเครือข่ายจริง ๆ เราจะได้ทดลองและออกแบบระบบเครือข่ายได้แบบง่ายดาย ซึ่งก็จะเป็นการปฏิวัติการเรียนรู้เลยทีเดียว เนื่องจากสมัยก่อนการที่จะได้จับอุปกรณ์จริง ๆ นั้นเป็นไปได้ยากมาก (ราคาแพงมาก) ถึงแม้จะได้จับแล้วก็ใช่ว่าจะคอนฟิกอุปกรณ์เหล่านั้นได้ทันที และแม้แต่จะได้คอนฟิกแล้วก็ไม่แน่เสมอไปว่าจะทำได้

หรือเข้าใจการทำงานของมันอย่างถ่องแท้ ดังนั้น ณ ***จุดนี้จึงเป็นการดีที่เราจะได้ออกแบบและคอนฟิกระบบเครือข่ายโดยไม่จำเป็นต้องมีอุปกรณ์เลยแม้แต่สักตัวเดียว*** และเมื่อถึงเวลาที่เราจะต้องจับอุปกรณ์ขึ้นมาจริง ๆ แล้วล่ะก็ จะไม่เสียเวลาในการเรียนรู้เลย และถึงแม้ว่าในอนาคตจะไม่มีโอกาสได้จับอุปกรณ์จริง ๆ (อาจจะเป็นผู้บริหาร) ก็จะได้ทราบถึง Specification ของตัวอุปกรณ์พร้อมทั้งความสามารถของมัน เพื่อใช้ประกอบในการตัดสินใจสำหรับการออกแบบและวางระบบเครือข่ายต่อไป

การติดตั้งซอฟท์แวร์จำลองเครือข่าย

ซอฟท์แวร์ที่จะใช้ทดลองและติดตั้งสามารถหาได้จากการเข้าไปดาวน์โหลดในเว็บไซต์ของบริษัทผู้พัฒนาซอฟท์แวร์ ซึ่งเป็นตัว Demo คือเป็นซอฟท์แวร์ที่ทางผู้พัฒนาอนุญาตให้ผู้สนใจนำไปติดตั้งและใช้งานได้ โดยกำหนดอายุการใช้งานของซอฟท์แวร์ไว้ เช่น ใช้งานได้ 7 วันบ้าง 15 วันบ้าง หรืออาจจะไม่มีอายุการใช้แต่ได้ตัดคุณสมบัติของซอฟท์แวร์บางอย่างออกไป ซึ่งถ้าใช้งานแล้วเกิดความพอใจก็ทำการสั่งซื้อต่อไปในภายหลัง

1. เว็บไซต์ของ Boson Software www.boson.com
2. เว็บไซต์ของ NetSim http://www.routersim.com,
   http://www.soft32.com/download_75090.html
3. เว็บไซต์ที่น่าสนใจคือ Packet Tracer http://www.cisco.com

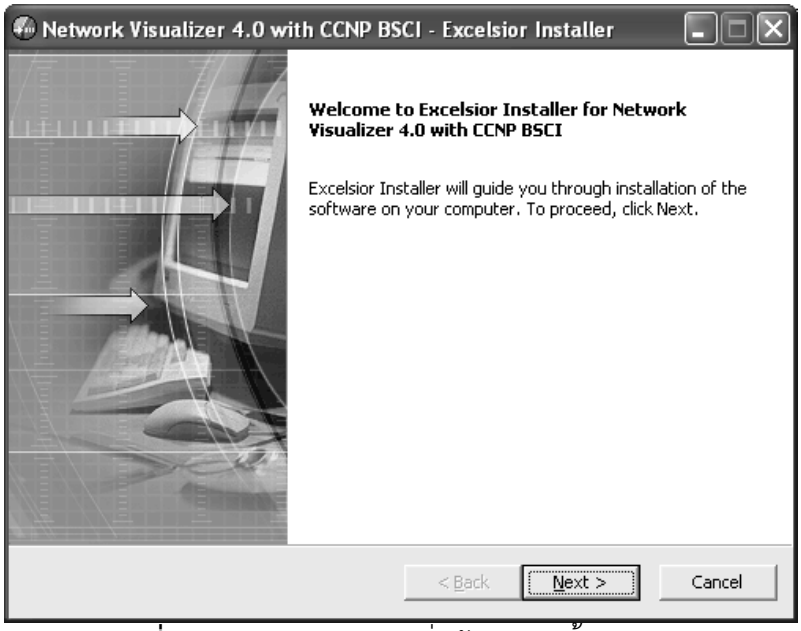
การติดตั้ง RouterSim Network Visualizer 4.0

จุดเด่นของซอฟท์แวร์ค่ายนี้คือ ความง่าย ต่อไปผู้เขียนจะขอเรียกว่า NetSim ซอฟท์แวร์ตัวนี้จะมีอุปกรณ์ไม่มากนักคือ มีเร้าเตอร์ Cisco รุ่น 2600 สวิตช์เลเยอร์ 2 รุ่น Catalyst 1900, 2950 และ

เครื่อง PC ที่นำเอามาอธิบายก่อนเนื่องจาก ผู้เขียนเห็นว่าเหมาะสำหรับผู้ที่จะเริ่มต้นคอนฟิกใหม่ ๆ ในตอนแรก ๆ ซึ่งควรจะใช้ซอฟท์แวร์ที่ใช้งานง่าย ๆ ไม่ยุ่งยาก อุปกรณ์น้อย ๆ

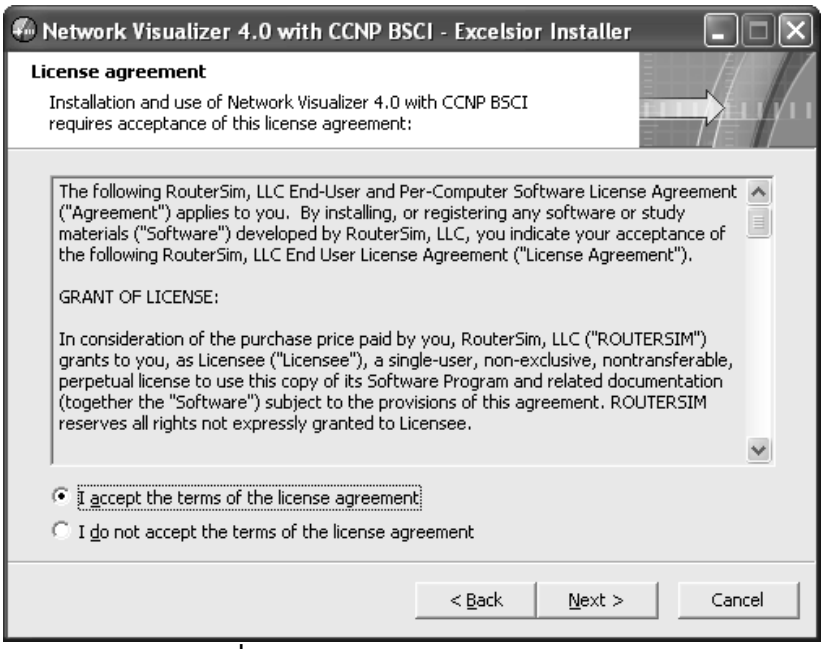
ขั้นตอนการติดตั้ง NetSim

1. ดับเบิลคลิ๊กไฟล์ชื่อ Netvis4.exe
2. จะปรากฏ Dialog Box ขึ้นมาให้คลิ๊กเลือก Next >



รูปที่ 3 แสดง Dialog Box เริ่มต้นการติดตั้ง

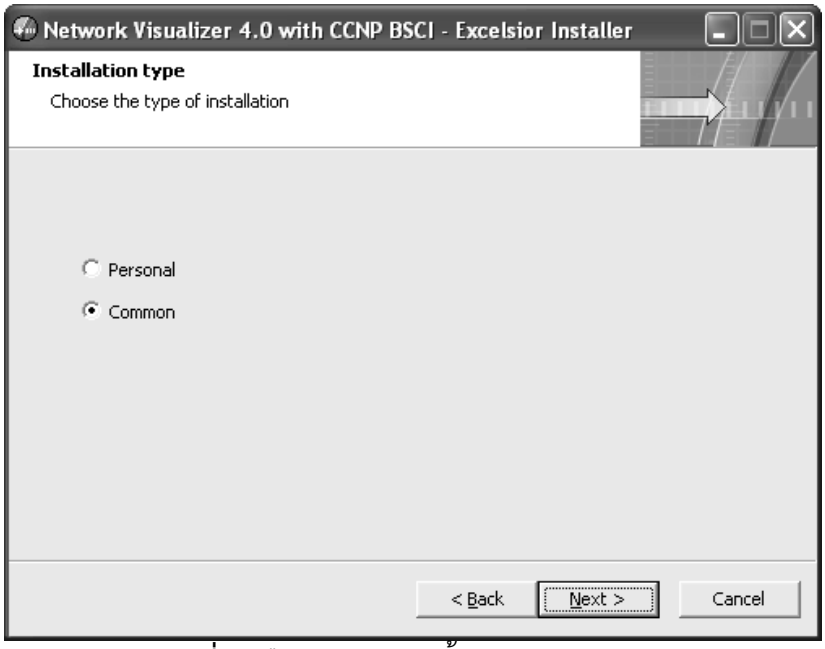
3. เมื่อคลิ๊ก Next แล้วจะแสดง Dialog Box แสดงถึง lisence ของซอฟท์แวร์ ให้เลือก I Accept the term of the license agreement แล้วเลือก Next >



รูปที่ 4 แสดงข้อความ license

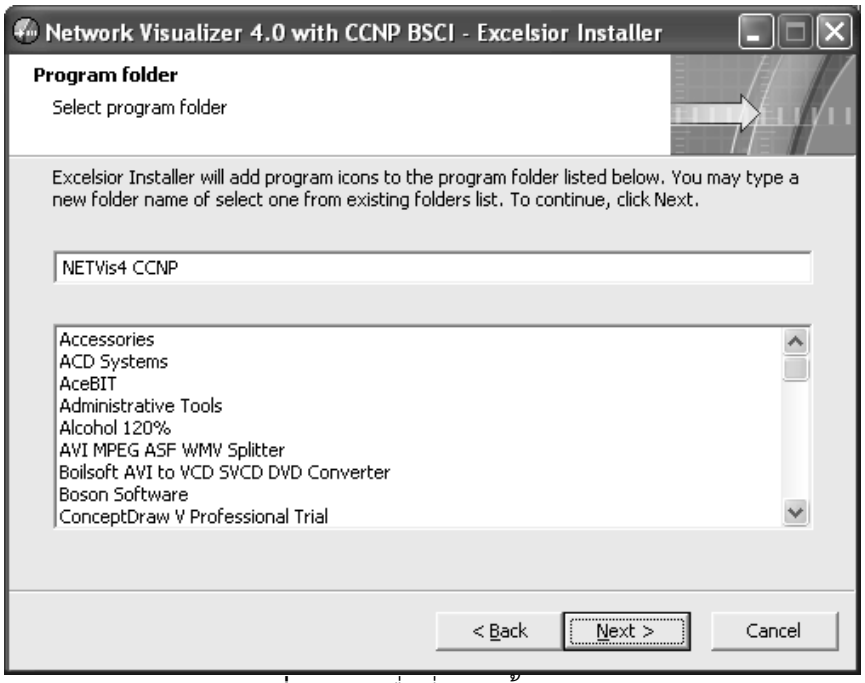
4. ขั้นตอนต่อไปจะมี Dialog ขึ้นมาบอกว่าจะเลือกติดตั้งแบบไหนซึ่งมี 2 รูปแบบคือ
   - personal เลือก Option นี้จะทำการติดตั้งให้ใช้งานได้เฉพาะผู้ที่ทำการติดตั้งเท่านั้น
   - common เลือก Option นี้จะทำให้ผู้ใช้คนอื่น ๆ สามารถทำงานได้

ให้เลือก option comon แล้วกดปุ่ม Next >



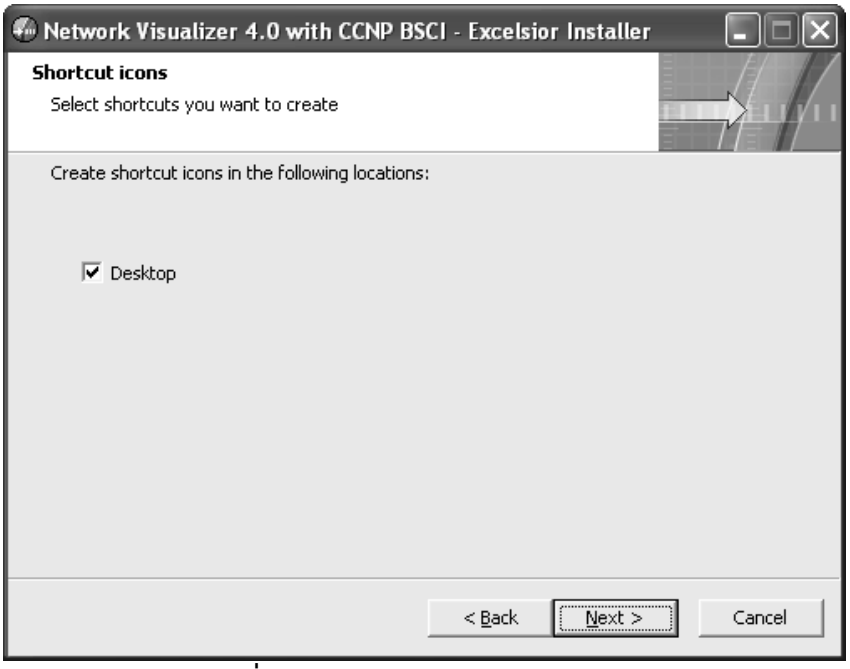
รูปที่ 5 เลือกชนิดการติดตั้ง

5. ขั้นต่อไปจะปรากฏ Dialog แสดงตำแหน่งที่ต้องการติดตั้งบนเครื่องพร้อมกับพื้นที่ที่ต้องการจะใช้ติดตั้ง ให้เลือก Next >



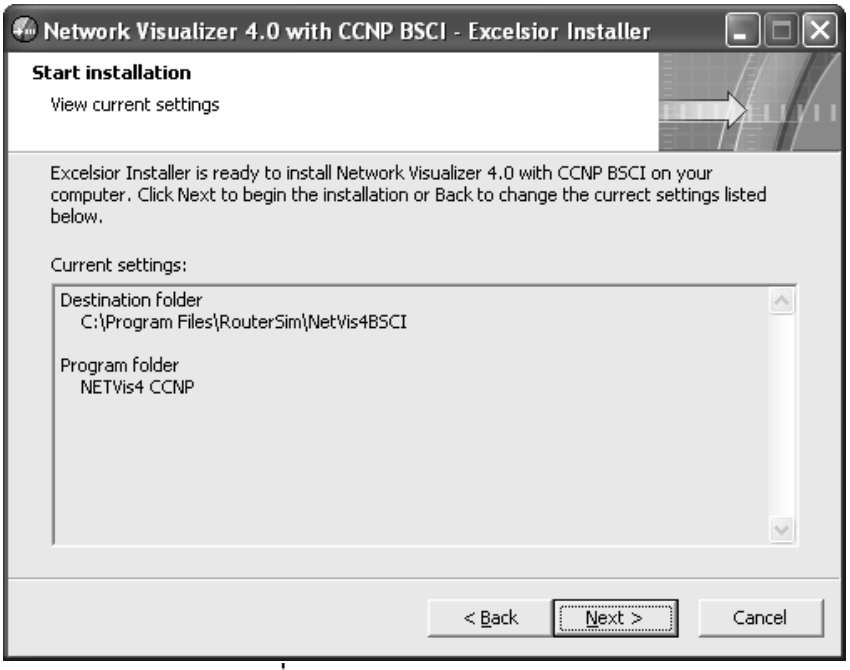
รูปที่ 6 แสดงชื่อที่จะติดตั้ง

6. ขั้นตอนต่อไปให้เลือก Next > ต่อไป



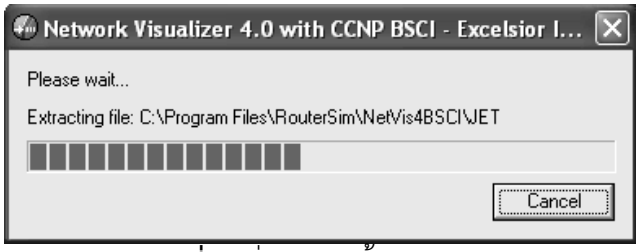
รูปที่ 7 สร้าง Shortcut icons

7. ขั้นต่อไปจะขึ้น Dialog รายการสรุปที่ได้เลือกไว้ก่อนหน้าให้เลือก Next >



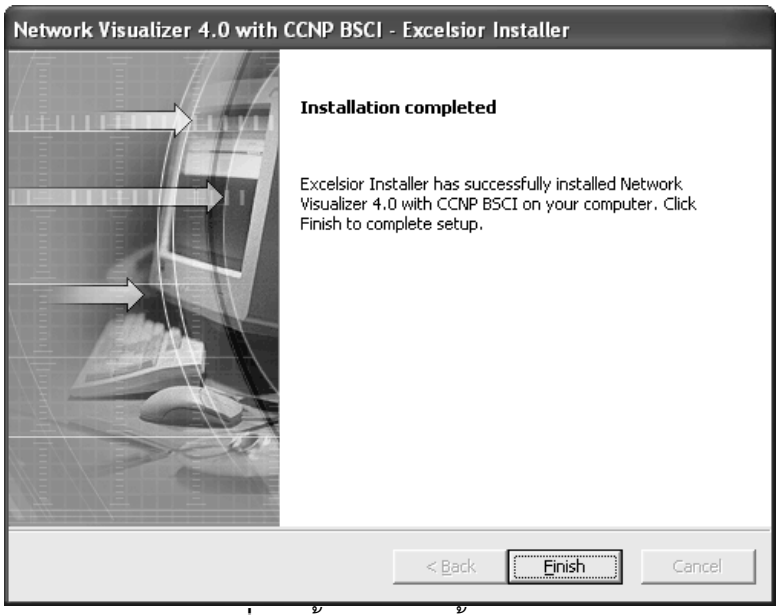
รูปที่ 8 แสดงรายการสรุป

8. โปรแกรมจะเริ่มทำการติดตั้ง



รูปที่ 9 เริ่มการติดตั้ง NetSim

9. ขั้นตอนสุดท้ายให้กดปุ่ม finish ซึ่งแสดงว่าสิ้นสุดการติดตั้ง



รูปที่ 10 สิ้นสุดการติดตั้ง NetSim

เริ่มต้นการใช้งาน NetSim

เปิดโปรแกรม NetSim โดยเข้าไปที่ Start → Programs → Netvis4 รอสักครู่ เมื่อโปรแกรมทำงานจะปรากฏวินโดส์ขึ้นมา 2 วินโดส์ คือ วินโดส์ RouterSim Network Visualizer 4.0 และอีกวินโดส์คือHow do

วินโดส์ RouterSim Network Visualizer จะเป็นตัวหลักในการทำงาน ในตอนแรกจะไม่สามารถใช้งานได้เนื่องจากต้องมีการติดตั้ง license ให้กับโปรแกรมก่อน

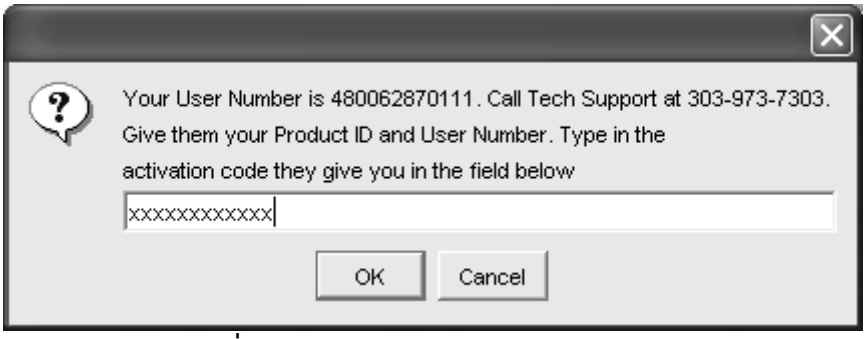
รูปที่ 11 วินโดส์หลักของ NetSim

การติดตั้ง license

1. ให้คลิกที่ปุ่ม Install License จะปรากฏ Dialog box ว่าต้องการลงทะเบียนแบบใด เช่น Web Install, Manual ให้เลือกเป็นแบบ Manual install

รูปที่ 12 เลือก Install License แบบ Manual

2. จะปรากฏ Dialog box เพื่อให้เราใส่ Activation Code ให้ทำการใส่ลงไปแล้วกดปุ่ม Ok



รูปที่ 13 การทำ Activation Code

3. ถ้า Code ที่ใส่ถูกต้องโปรแกรมจะแสดงข้อความว่าสำเร็จแล้ว ให้กดปุ่ม Ok

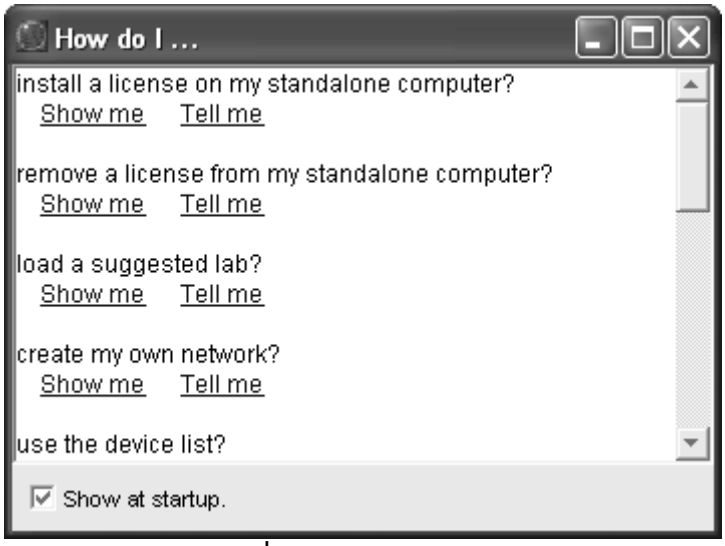
รูปที่ 14 การ activation สำเร็จ

4. เมื่อ Install License สมบูรณ์โปรแกรมก็แสดงเมนูให้สามารถใช้งานได้แล้ว

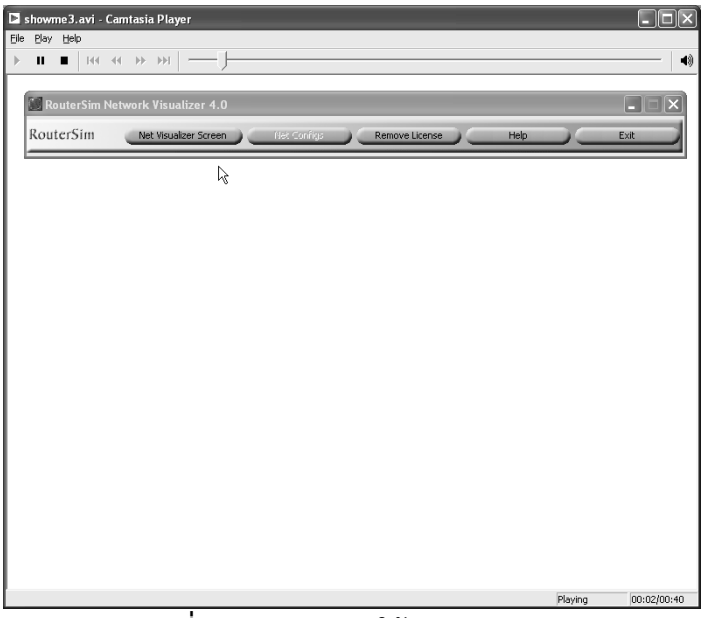
รูปที่ 15 โปรแกรมพร้อมที่จะทำงาน

การใช้งาน How do..

วินโดส์ How do ใช้สำหรับสอนการใช้งานโปรแกรม มีลักษณะเป็นไฟล์มัลติมีเดีย คือมีเสียงอธิบายการทำงานพร้อมกับเมาส์ที่เลื่อนไปยังตำแหน่งที่เสียงนั้น ๆ กล่าวถึง การใช้งานก็เพียงแค่คลิกเลือกหัวข้อที่เราสนใจ โปรกรมก็จะทำงานให้อัตโนมัต



รูปที่ 16 วินโดส์ How do



รูปที่ 17 ตัวอย่างการใช้งาน How do

เมนูต่าง ๆ ของ RouterSim

1. เมนู Net Visualizer Screen เป็นเมนูที่ใช้การออกแบบและคอนฟิกอุปกรณ์ เมื่อทำการคลิกเลือกเมนูนี้จะปรากฏหน้าต่างสำหรับเริ่มต้นออกแบบเครือข่าย

รูปที่ 18 แสดงเมนูหลักของ RouterSim

อธิบายการทำงานของเมนูต่าง ๆ

เมนู File

เป็นเมนูที่เกี่ยวข้องกับการทำงานของไฟล์ เช่น การโหลดโปรแกรมใหม่ การบันทึกโปรแกรม การพิมพ์ ข้อมูลที่ทำการบันทึก จะมีนามสกุลเป็น .rsm

- **Net Visualizer Screen** ทำหน้าที่สร้างเมนูสำหรับใช้ในการออกแบบเน็ตเวิร์ค
- **Load Network** เรียกโปรแกรมที่บันทึกไว้มาทำงาน
- **Save Network** บันทึกโปรแกรมที่ทำงานโดยใช้นามสกุล .rsm
- **Download from Server** สามารถดาวน์โหลดไฟล์การคอนฟิกเพิ่มเติมจากเซิร์ฟเวอร์ที่เราสร้างไว้เพื่อเก็บไฟล์คอนฟิกต่าง ๆ ไว้ ออบชันนี้มีใช้ตั้งแต่ 4.1 ขึ้นไป
- **Publish from Server** ส่งไฟล์ไปเก็บไว้ที่เซิร์ฟเวอร์ที่ใช้สำหรับเก็บข้อมูล
- **Option** สำหรับเซ็ตค่าพื้อหลังของเมนู
- **Print** พิมพ์โครงสร้างของเครือข่ายที่ได้ออกแบบไว้
- **Quit** ออกจากโปรแกรม

เมนู Edit ใช้สำหรับแก้ไขโครงสร้างเครือข่ายที่ได้ออกแบบไว้

เมนู Add ใช้สำหรับต้องการเพิ่มอุปกรณ์เข้าไปในโครง

เมนู View

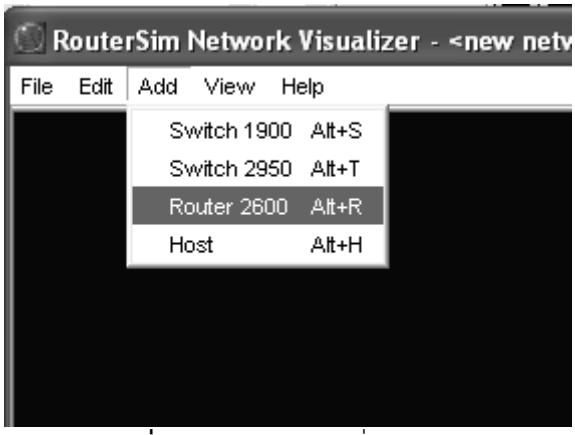
- **Device List** เรียกดูอุปกรณ์ที่มีทั้งหมด
- **Clear Visualizer** ลบอุปกรณ์ต่าง ๆ ทั้งหมดออกจากหน้าต่างทำงาน
- **Net Config** เรียกหน้าต่างการทำงานที่แสดงออกมาเป็นรูปของทรี
- **Lab** เลือกชนิดของ Lab ที่ต้องการทดลอง
    1. Extend เป็น Lab ที่มีความซับซ้อนมากและยากขึ้นมากกว่า standard
    2. Standard เป็น Lab ที่มีความยากพอประมาณ ขนาดของเน็ตเวิร์คจะมีอุปกรณ์ที่ใช้เชื่อมต่อปานกลาง
    3. Two Router เป็นการเชื่อมต่อเร้าเตอร์ 2 ตัวเท่านั้น เหมาะสำหรับผู้ที่หัดคอนฟิกใหม่ ๆ
    4. VLANs เป็น Lab เกี่ยวกับการสร้าง VLAN
    5. VSLM เป็น Lab ที่เกี่ยวกับการทำ Subnet ซ้อน Subnet

- **IP Addresses, Port Numbers, Hostnames** ให้แสดงข้อมูลที่ได้เลือกไว้ว่าต้องการแสดง อะไรบ้าง

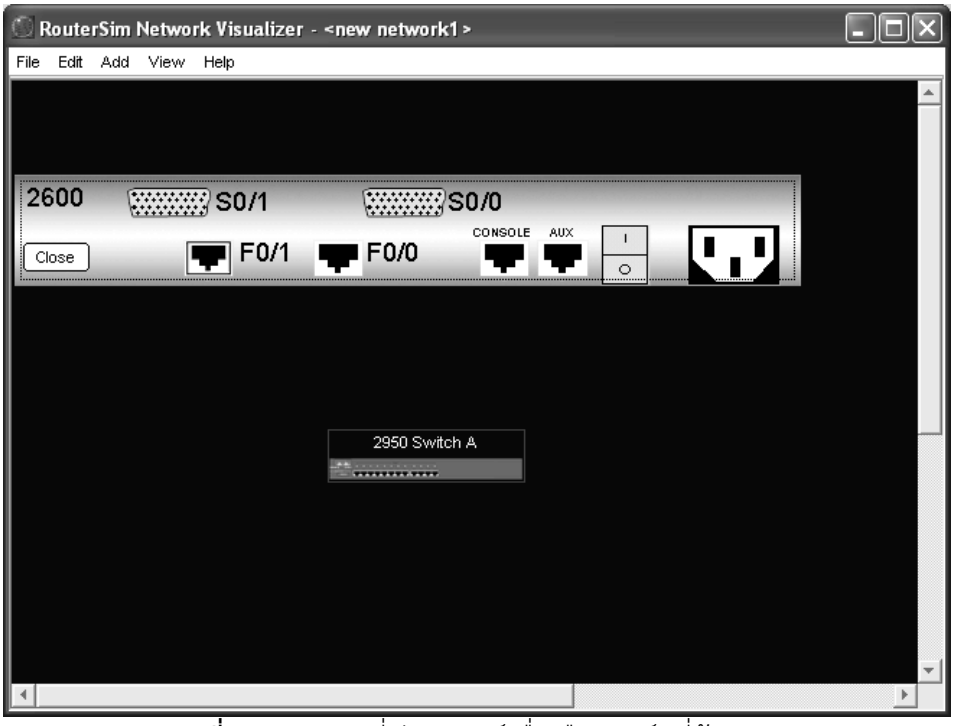
เมนู Help เป็นเมนูที่ใช้สำหรับต้องการความช่วยเหลือ เช่น กด F1 หรือเรียกดูมัลติมีเดียเกี่ยวกับการคอนฟิกอุปกรณ์

การออกแบบเน็ตเวิร์คและการคอนฟิกด้วย RouterSim

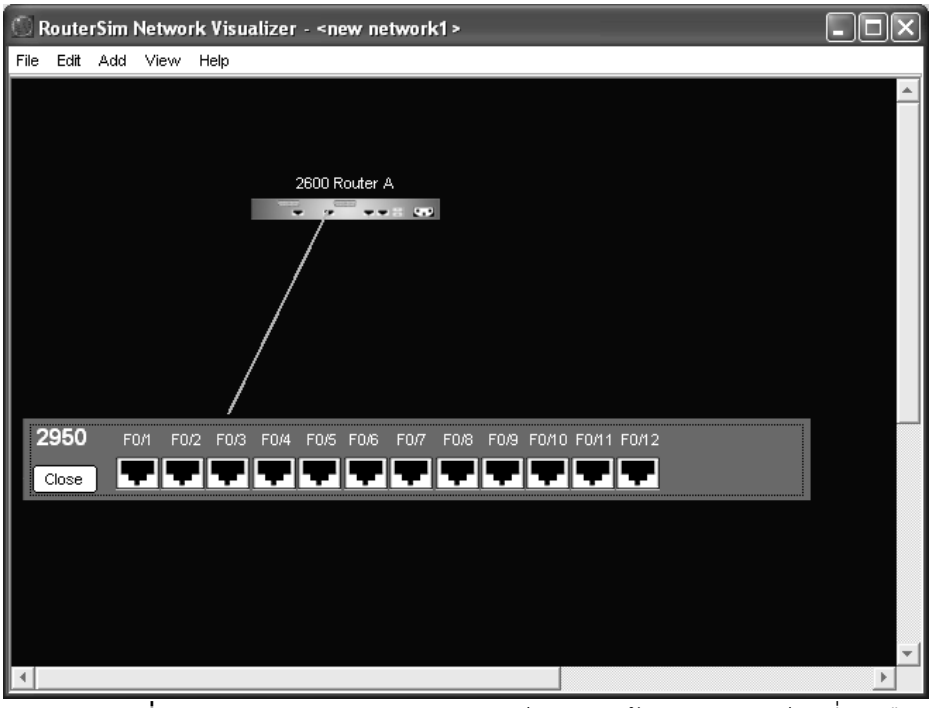
RouterSim เป็นซอฟท์แวร์จำลองเครือข่ายที่ใช้งานได้ง่ายมาก โดยแค่เลือกอุปกรณ์ที่ต้องการแล้วลากมาวางไว้ที่หน้าต่างการทำงาน ส่วนการคอนฟิกก็ทำได้โดยการดับเบิลคลิกที่ตัวอุปกรณ์เท่านั้น ถ้าต้องการเชื่อมโยงระหว่างอุปกรณ์ 2 ตัวก็ทำได้ง่าย ๆ โดยการคลิกขวาที่ตัวอุปกรณ์แล้วเลือกพอร์ตที่ต้องการ จากนั้นให้ลากมาที่อุปกรณ์ที่ต้องการเชื่อมต่อด้วยแล้วคลิกขวา จากนั้นเลือกพอร์ตแล้วคลิกที่พอร์ต



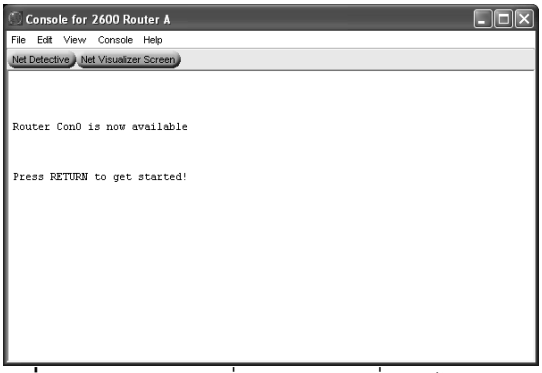
รูปที่ 19 แสดงวิธีการเพิ่มอุปกรณ์



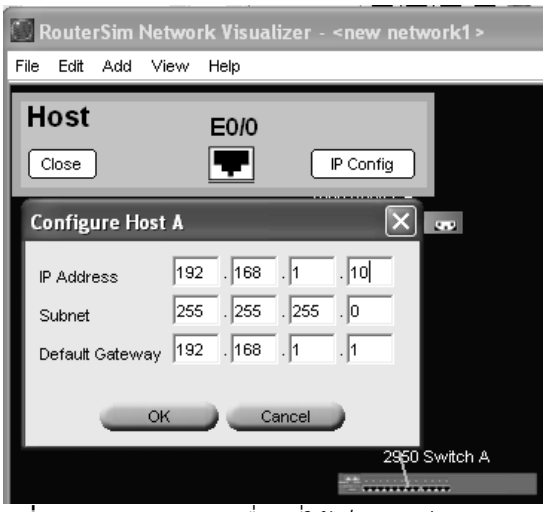
รูปที่ 20 คลิกขวาที่ตัวอุปกรณ์เพื่อเลือกพอร์ตที่ต้องการ



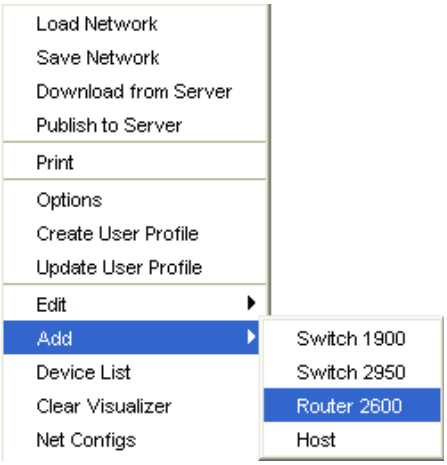
รูปที่ 21 ลากสายสัญญาณจากอุปกรณ์ตัวแรกแล้วคลิกอุปกรณ์ตัวที่ 2 เลือกพอร์ตที่จะเชื่อมต่อกัน



รูปที่ 22 ดับเบิลคลิกที่ตัวอุปกรณ์เพื่อเข้าไปคอนฟิก



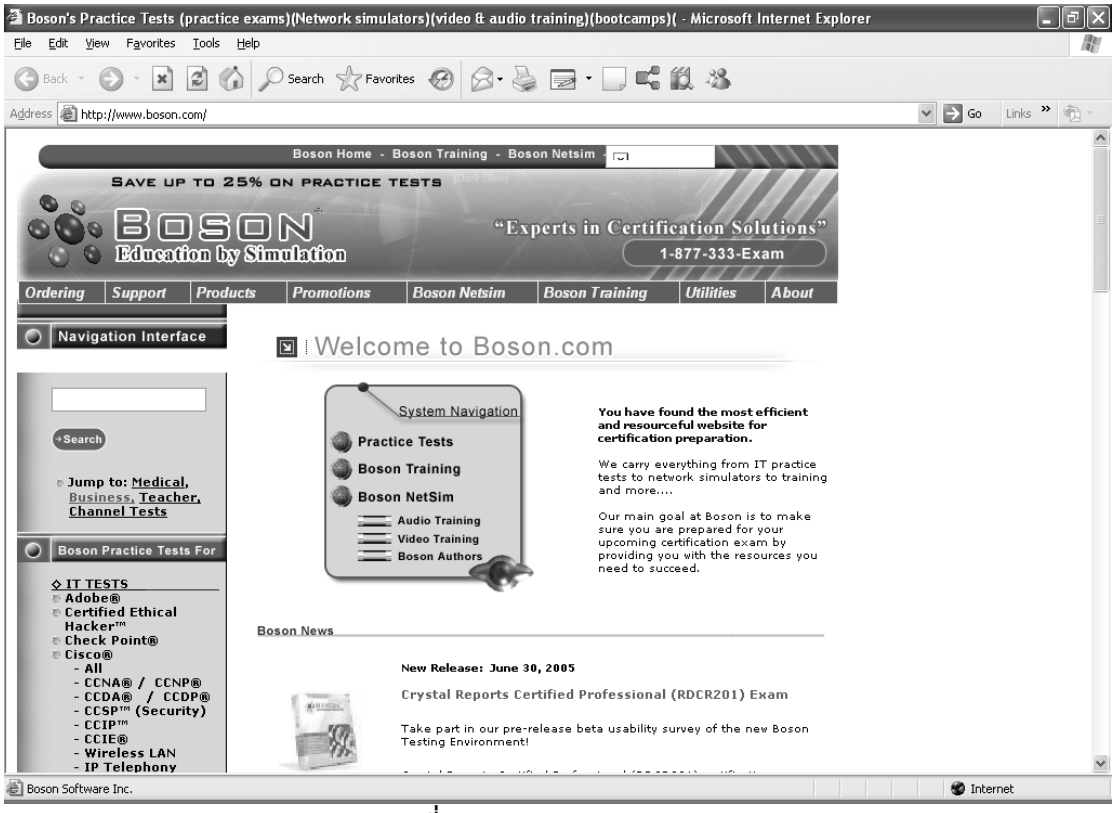
รูปที่ 23 การคอนฟิกเครื่องที่ใช้เป็นเทอร์มินอล (PC)



รูปที่ 24 สามารถคลิกขวาที่หน้าต่างทำงานจะปรากฏ pop up เมนูให้เลือกใช้งานได้

การติดตั้ง Boson NetSim

จุดเด่นของค่าย Boson คือมีอุปกรณ์ให้เลือกใช้งานได้มากมายและมีหลายรุ่น ซึ่งประกอบไปด้วย เร้าเตอร์ สวิตซ์ และอุปกรณ์ที่จำเป็นในการออกแบบเครือข่ายอื่นๆ จุดเด่นอีกอย่างที่สำคัญอีกประการหนึ่งคือ ซอฟท์แวร์จากค่ายนี้จะมีการอัพเดทและมีเวอร์ชันใหม่ ๆ ออกมาอย่างต่อเนื่อง ซึ่งสามารถค้นหาข้อมูลเพิ่มเติมเกี่ยวกับการอัพเดท หรือ ซอฟท์แวร์ใหม่ ๆ ได้จากเว็บไซต์ www.boson.com



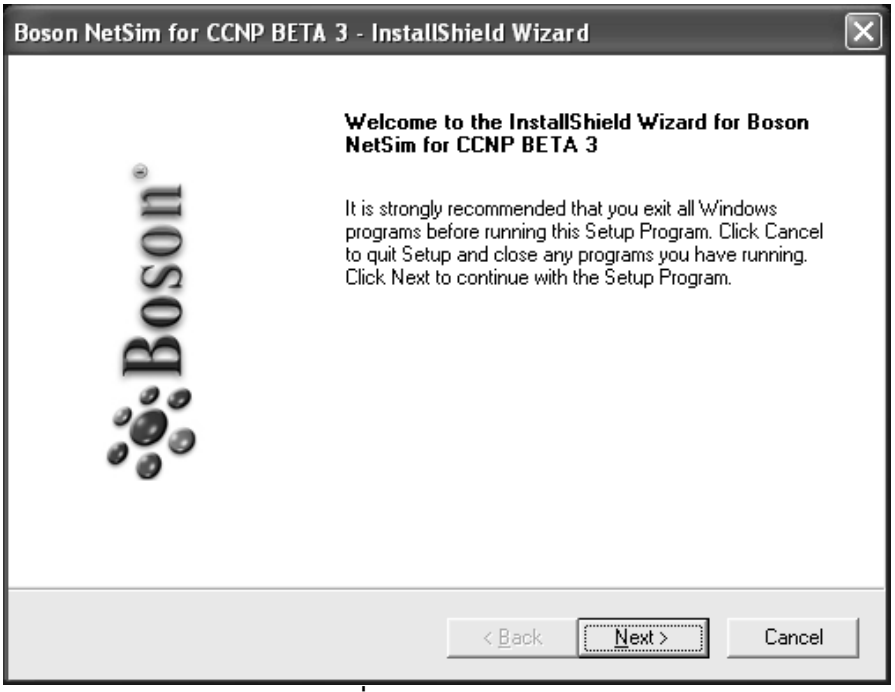
รูปที่ 25 Boson.com

ขั้นตอนการติดตั้ง Boson NetSim

1. ดับเบิลคลิกไฟล์สำหรับติดตั้งชื่อ NetSim.exe

รูปที่ 26 เริ่มต้นติดตั้ง Boson NetSim

2. ขั้นต่อไปจะปรากฏ Dialog box "welcom to NetSim" ให้เลือก Next >

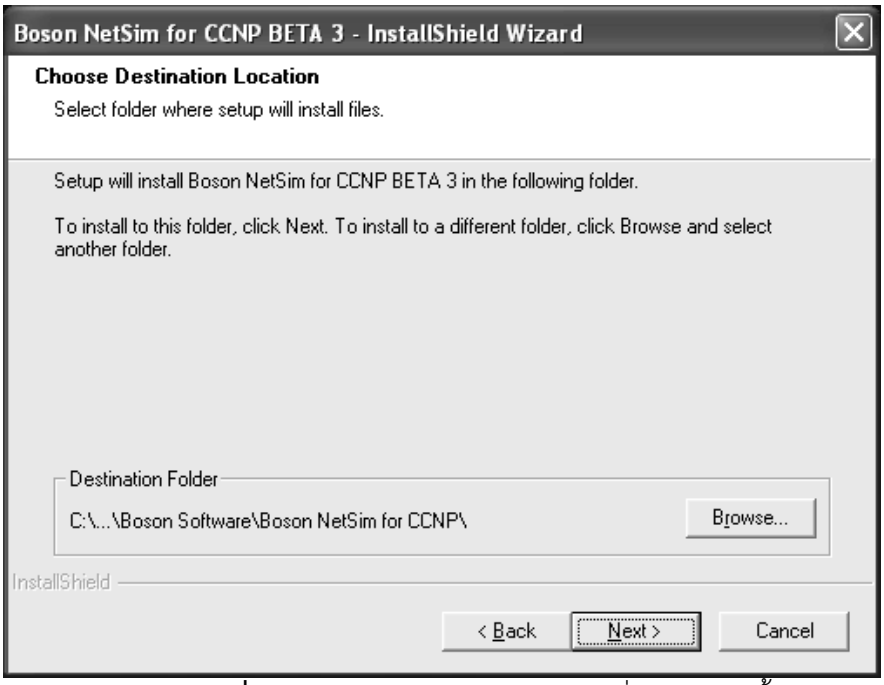


รูปที่ 27 welcome to NetSim

3. ขั้นต่อไป จะปรากฏ License Agreement ให้เลือก yes

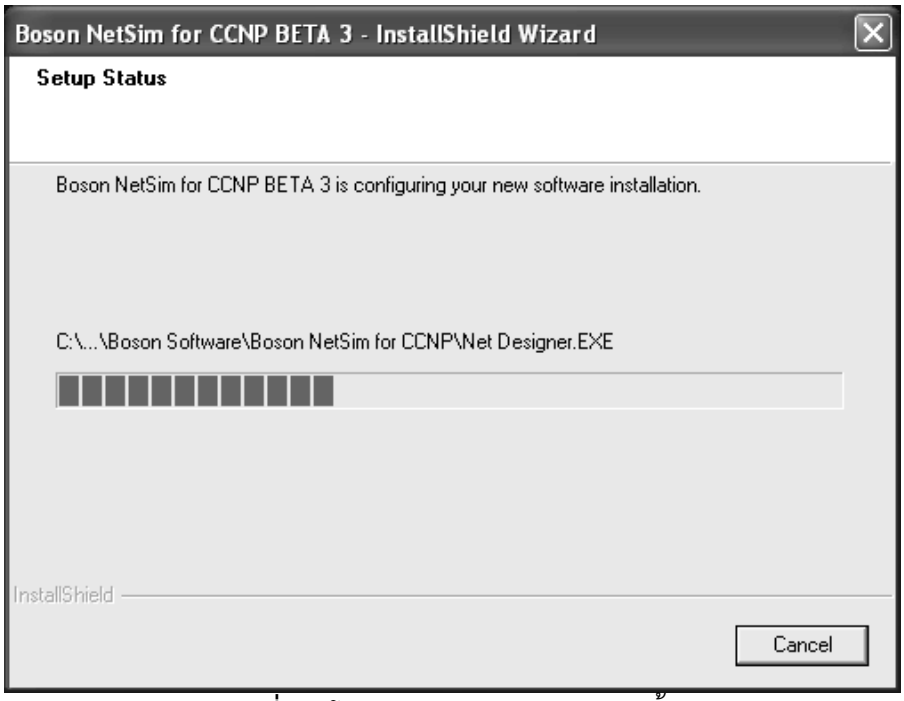
รูปที่ 28 License Agreement

4. ขั้นต่อไปจะปรากฏเลือกตำแหน่งที่ต้องการติดตั้งซอฟท์แวร์ ให้เลือก Next >



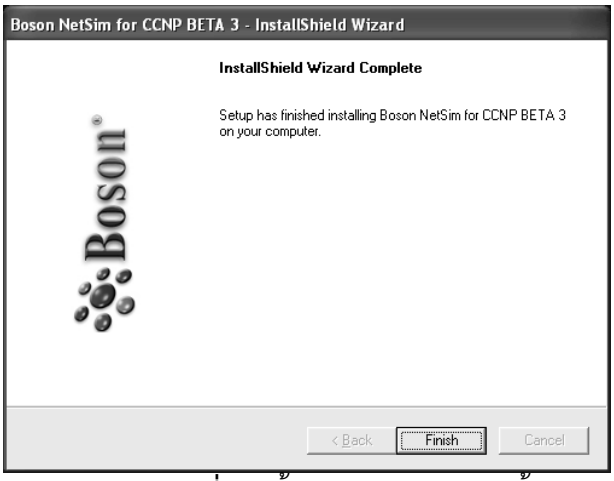
รูปที่ 29 เลือกตำแหน่งของซอฟท์แวร์ที่ต้องการติดตั้ง

5. ขั้นตอนต่อไปจะปรากฏ Start Copying ให้เลือก Next > จากนั้นโปรแกรมจะเริ่มทำการติดตั้งลงบนเครื่อง



รูปที่ 30 โปรแกรมกำลังดำเนินการติดตั้ง

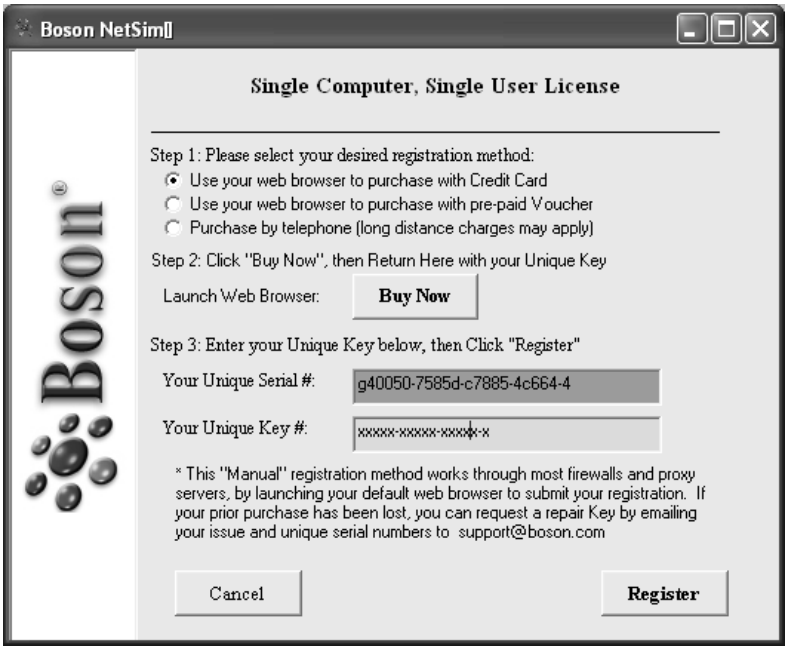
6. ขั้นตอนสุดท้ายโปรแกรมจะแสดงข้อความว่าได้ทำการติดตั้งเรียบร้อยแล้วให้เราเลือก Finish



รูปที่ 31 สิ้นสุดกระบวนการติดตั้ง

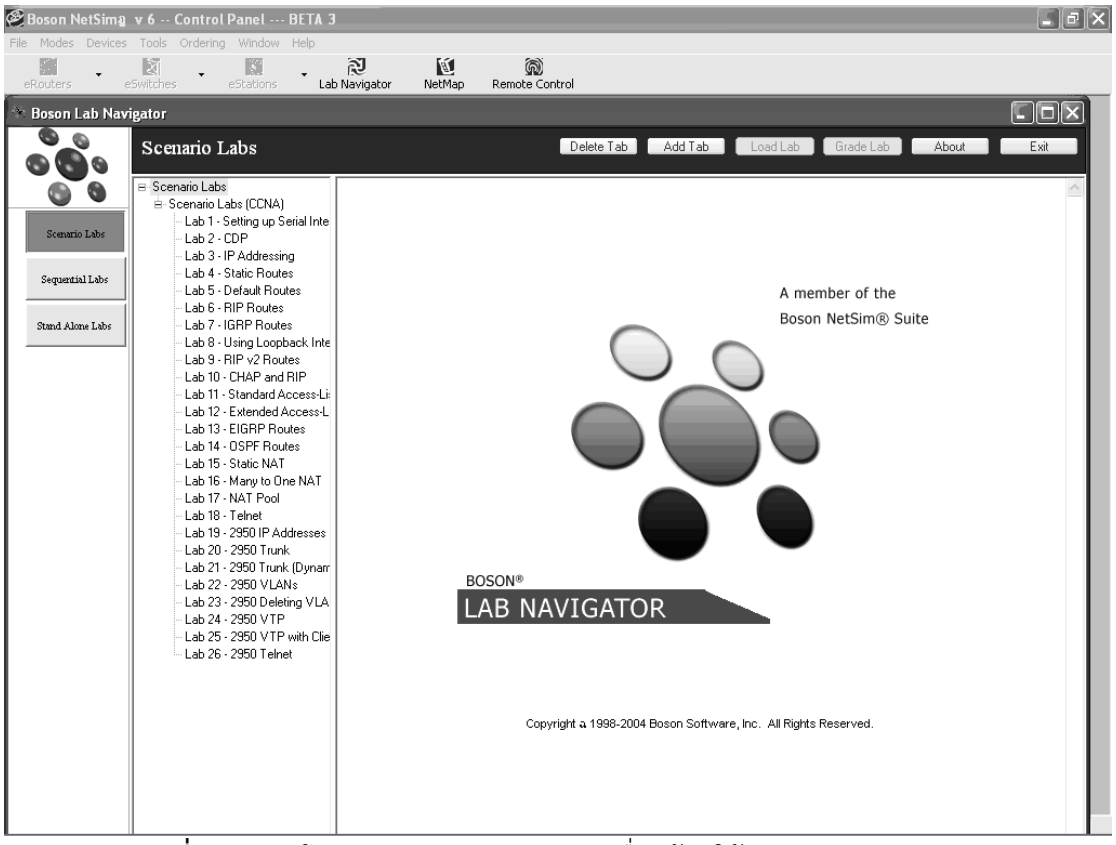
เริ่มต้นใช้งาน Boson NetSim

1. เปิดโปรแกรม Boson NetSim โดยคลิกเลือกที่ Start → Programs → Boson Software → Boson NetSim for CCNx → Boson NetSim
2. ผู้ใช้งานจะต้องมี Unique Key ถ้าไม่มี จะสามารถใช้งานโปรแกรมได้อยู่เหมือนกัน แต่จะมีอายุการใช้งานที่จำกัด หรือ อาจจะปิดการทำงานฟังชันบางอย่างไป ถ้ามี Unique Key แล้วให้คลิกที่เมนู Ordering → Enter Repire Key ซึ่งจะปรากฏไดอะล็อกบล็อกให้ผู้ใช้งานใส่ Unique Key เข้าไปแล้วคลิก Register



รูปที่ 32 แสดงการลงทะเบียนใช้งานซอฟท์แวร์ Boson NetSim

3. เมื่อลงทะเบียนเสร็จฟังชันการทำงานทุกอย่างก็จะสามารถใช้งานได้ทั้งหมด



รูปที่ 33 แสดงโปรแกรม Boson NetSim เมื่อพร้อมใช้งาน

เมนูต่าง ๆ Boson NetSim

หัวข้อนี้จะกล่าวถึงวิธีการใช้งานเมนูต่าง ๆ ของ Boson NetSim ส่วนของการคอนฟิกและทดลองออกแบบระบบจริง ๆ จะกล่าวถึงอย่างละเอียดในส่วนที่ 5 ต่อไป

เมนู File หน้าที่หลักคือจัดการกับไฟล์คอนฟิกของ Boson NetSim

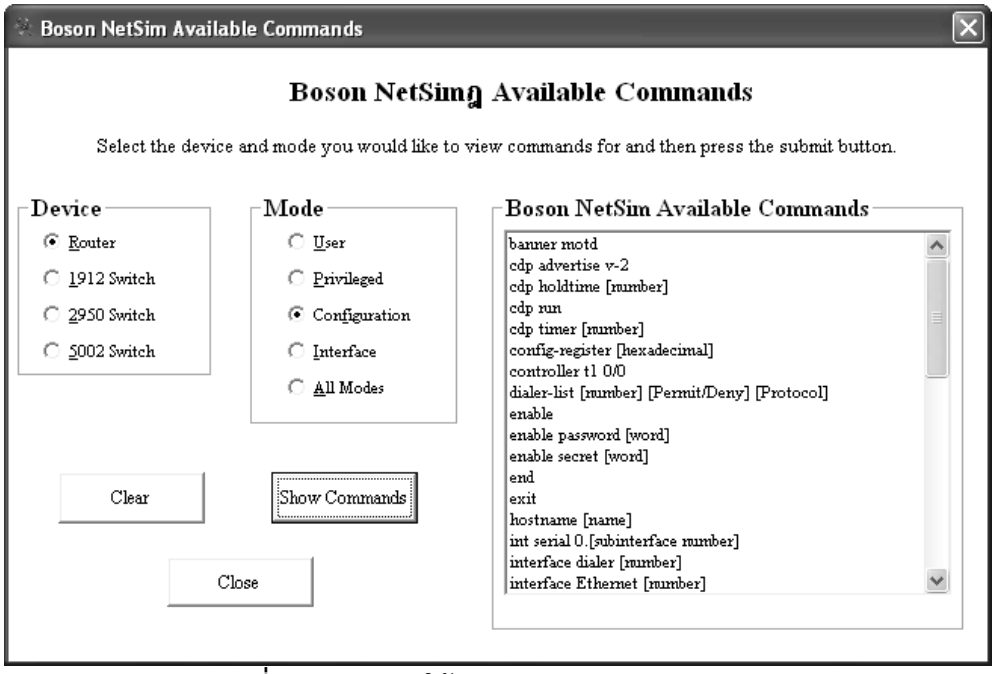
- **New NetMap** ใช้สำหรับสร้างแผนที่เน็ตเวิร์คใหม่
- **Load NetMap** ใช้สำหรับโหลดแผนที่เครือข่ายซึ่งมีนามสกุล .top
- **Paste Real Router Configs** ใช้สำหรับป้อนคำสั่งให้กับเร้าเตอร์ตัวปัจจุบันทำงานได้โดยตรง
- **Load Single Device Config** ใช้สำหรับกรณีที่ต้องการเลือกโหลดคอนฟิกเฉพาะอุปกรณ์เพียงบางตัวขึ้นมาทำงาน ซึ่งขึ้นอยู่กับตอนบันทึกว่า ได้บันทึกไว้ในลักษณะใด
- **Load Multi Device Config** ใช้สำหรับกรณีที่ต้องการเลือกโหลดคอนฟิกไฟล์ที่เก็บข้อมูลของหลาย ๆ อุปกรณ์ไว้ในไฟล์เดียวกัน

- **Save Single Device Config** ใช้บันทึกข้อมูลของคอนฟิกกูเรชันไฟล์สำหรับอุปกรณ์เพียงตัวเดียว โดยใช้นามสกุล .rtr
- **Save Multi Device Config** ใช้บันทึกข้อมูลของคอนฟิกกูเรชันไฟล์สำหรับอุปกรณ์หลาย ๆ ตัวไว้ในไฟล์เดียวกัน โดยใช้นามสกุล .nwc
- **Exit** ออกจากโปรแกรม

เมนู Modes ทำหน้าที่เปลี่ยนโหมดการทำงาน ซึ่งมีให้ใช้ 2 โหมดคือ Beginner Mode และ Advance Mode

เมนู Device แสดงรายการของอุปกรณ์ที่กำลังทำงานอยู่ที่ simulator ปัจจุบัน

เมนู Tools ทำหน้าที่ตรวจสอบซอฟท์แวร์ที่มีการอัพเดทจาก Boson เมนูที่สำคัญคือ Available Commands ใช้สำหรับตรวจสอบว่าการทำงานของอุปกรณ์ในแต่ละโหมดสามารถใช้คำสั่งอะไรได้บ้าง เช่น ถ้าต้องการทราบว่าเร้าเตอร์ที่อยู่ในโหมดของ Configuration สามารถใช้คำสั่งอะไรได้บ้าง ให้เลือกดังรูป

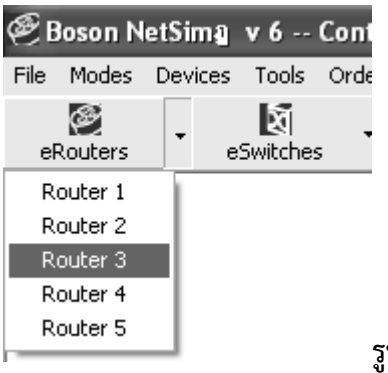


รูปที่ 34 แสดงการใช้งาน Available Commands

เมนู window ทำหน้าที่แสดงให้การแสดงผลในรูปแบบต่าง ๆ เช่น Cascade เป็นต้น

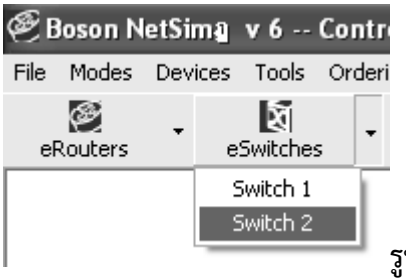
เมนู Help ทำหน้าที่ให้คำแนะนำ และให้ความช่วยเหลือการใช้งานซอฟท์แวร์

eRouters เป็น Tool ที่แสดงเร้าเตอร์ทั้งหมดที่กำลังทำงานอยู่ใน Simulator



รูปที่ 35 eRouters

eSwitches เป็น Tool ที่แสดงสวิตช์ทั้งหมดที่กำลังทำงานอยู่ใน Simulator

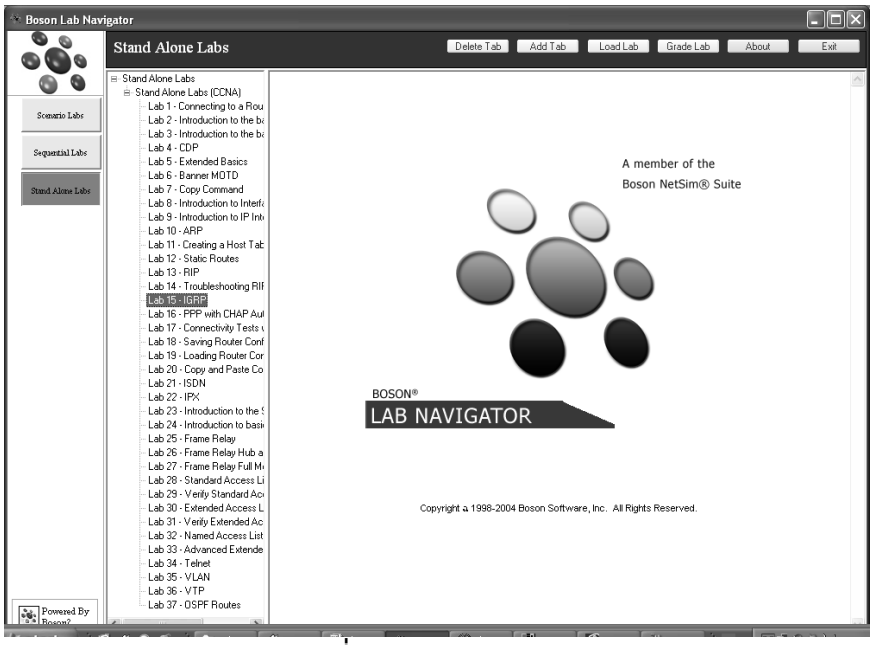


รูปที่ 36 eSwitches

eStations เป็น Tool ที่แสดงอุปกรณ์ที่ใช้เป็นเทอร์มินัล ในที่นี้ใช้ PC

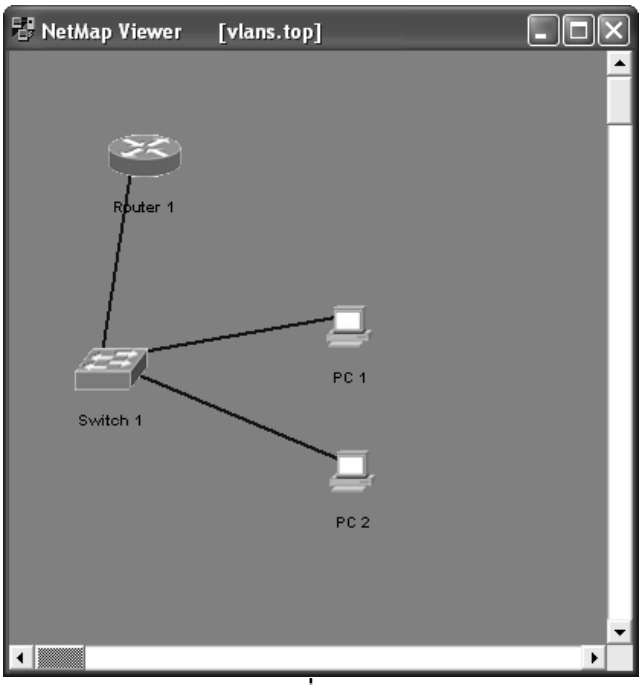
รูปที่ 37 eStations

Lab Navigator เป็น Tool ที่ใช้สำหรับโหลดแล็บทั้งหมดของ NetSim ขึ้นมาเพื่อให้ผู้ใช้งานสามารถเลือกทดลองแล็บใดก็ได้



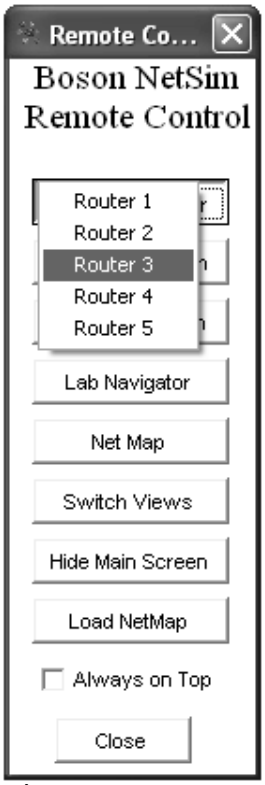
รูปที่ 38 Lab Navigator

Net Map เป็น Tool ที่ใช้สำหรับเรียกดูแผนที่ของเน็ตเวิร์คว่าเชื่อมต่อกันอย่างไรบ้าง



รูปที่ 39 Net Map

Remote Control เป็น Tool ที่รวบรวมเอาคำสั่งที่จำเป็นและใช้งานบ่อย ๆ ไว้ด้วยกัน เพื่อให้สะดวกขึ้นเมื่อต้องการเรียกใช้งานคำสั่งต่าง ๆ เช่น ต้องการ Telnet ไปยัง Router ก็คลิกที่ปุ่ม Telnet to eRouter แล้วเลือกเร้าเตอร์ที่ต้องการ ก็จะทำงานทันที

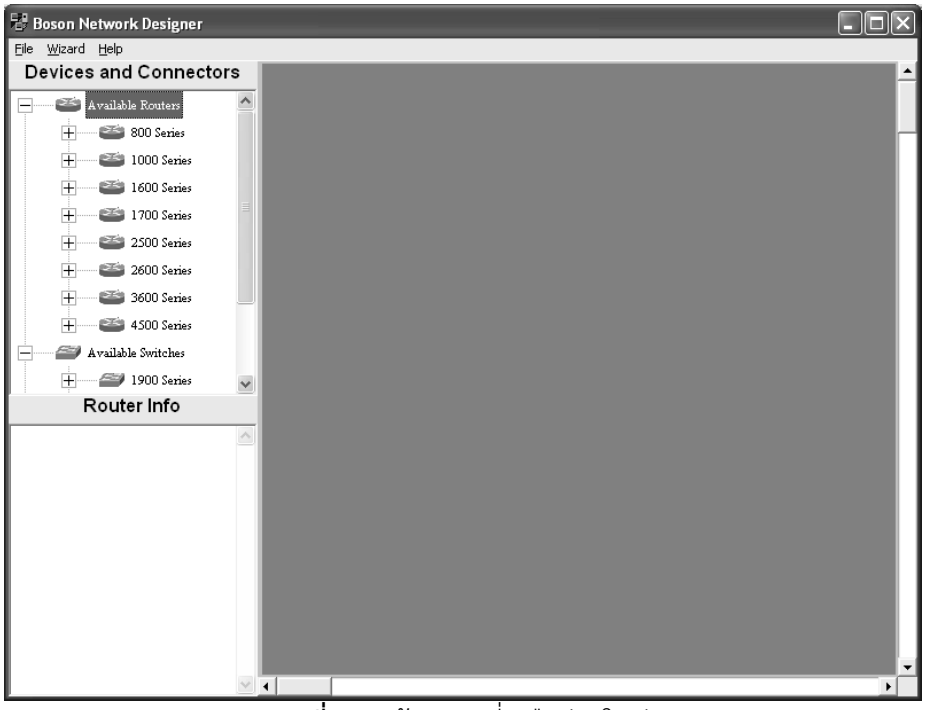


รูปที่ 40 Remote Control

การออกแบบเน็ตเวิร์คและการคอนฟิกด้วยเบื้องต้นด้วย Boson NetSim

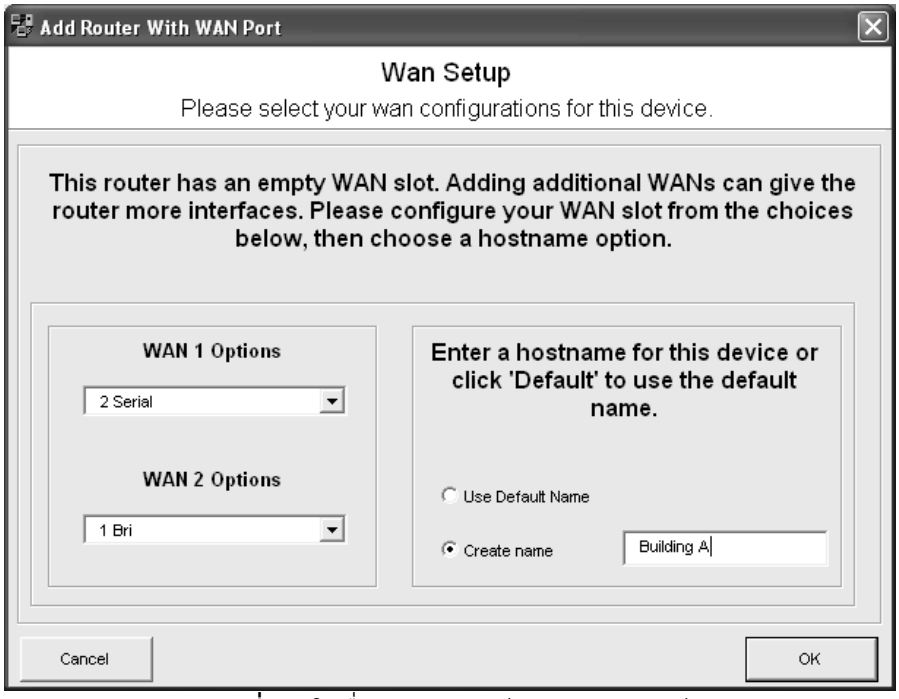
ขั้นตอนการออกแบบเน็ตเวิร์ตมีขั้นตอนหลัก ๆ ดังนี้

1. วางแผนและออกแบบเครือข่ายอย่างคร่าว ๆ ในกระดาษ
2. สร้างแผนที่เน็ตเวิร์คที่เราต้องการทดลอง โดยการเลือกเมนู File → new NetMap

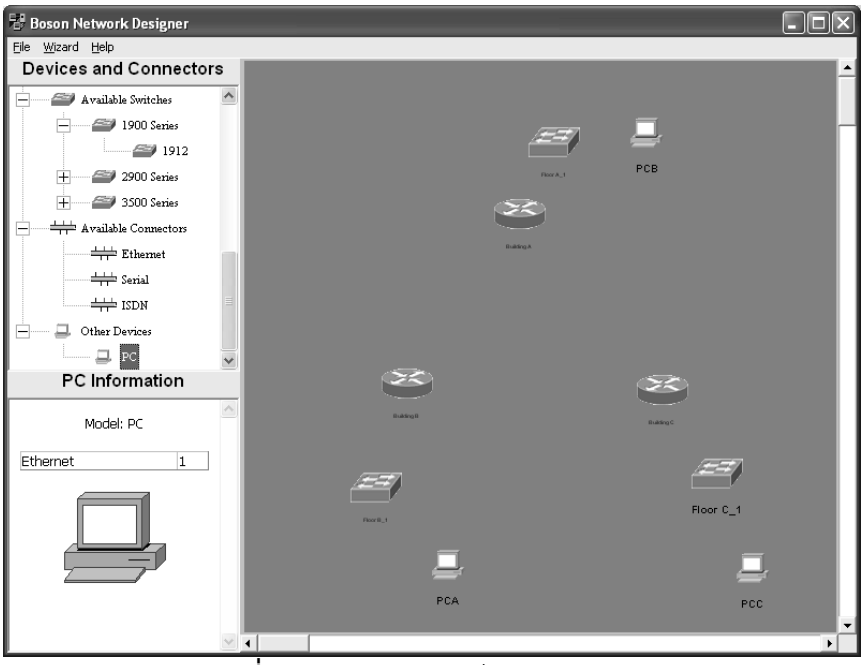


รูปที่ 41 สร้างแผนที่เครือข่ายใหม่

3. เลือกอุปกรณ์ที่เหมาะสมแล้ววางลงบนแผนที่เน็ตเวิร์ค NetMap

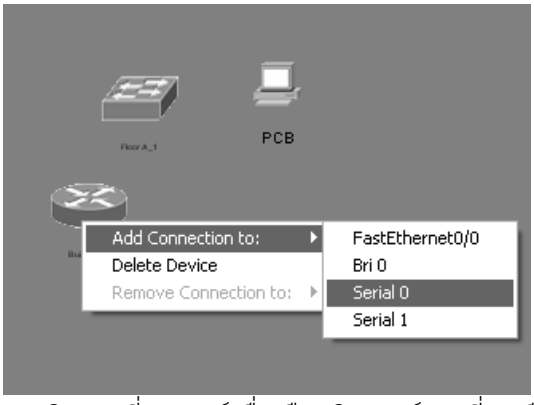


รูปที่ 42 ใส่ชื่อและอินเตอร์เฟสของอุปกรณ์

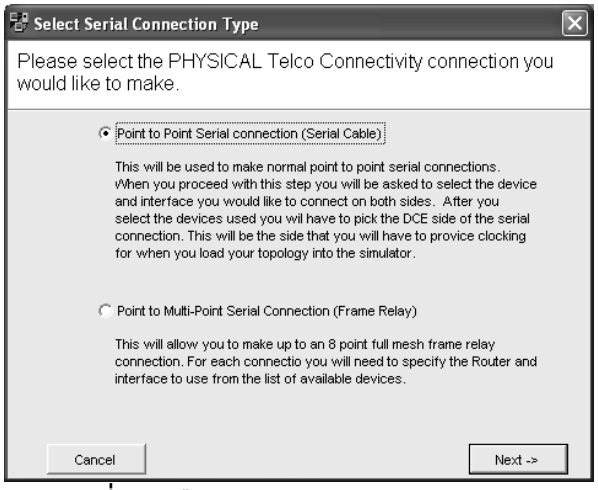


รูปที่ 43 การวางอุปกรณ์บน Simulator

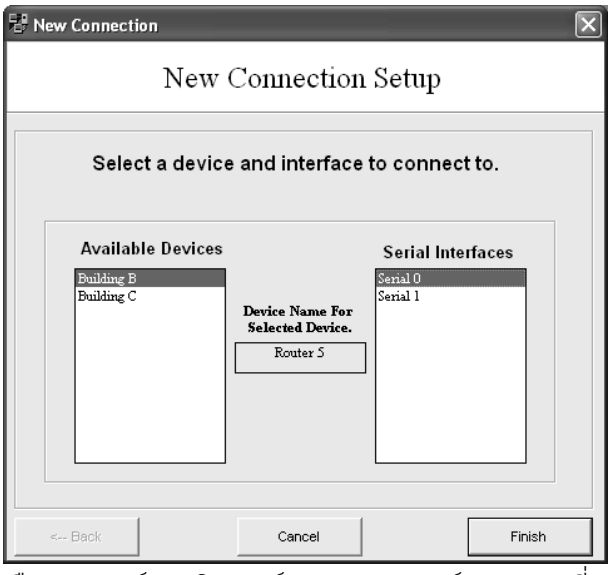
4. เชื่อมโยงเน็ตเวิร์คเข้าด้วยกันตามที่ได้ออกแบบไว้



รูปที่ 44 คลิกขวาที่อุปกรณ์เพื่อเลือกอินเตอร์เฟสที่จะเชื่อมต่อ

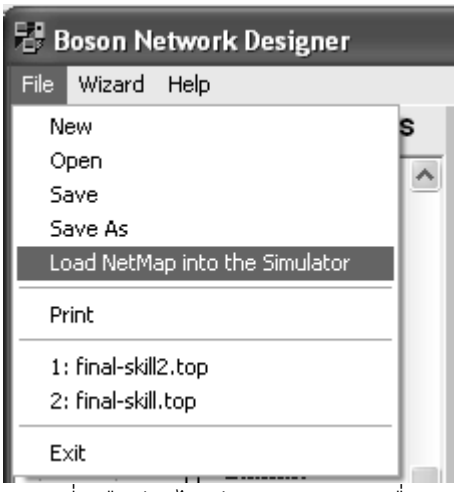


รูปที่ 45 เลือกประเภทของสายนำสัญญาณ

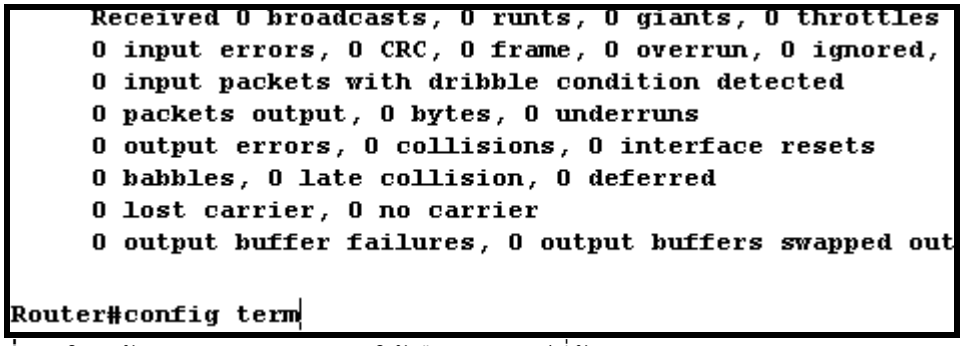


รูปที่ 46 เลือกอุปกรณ์และอินเตอร์เฟสของอุปกรณ์ปลายทางที่จะเชื่อมต่อ

5. เข้าไปคอนฟิกอุปกรณ์แต่ละตัวให้ครบ โดยการโหลดแผนที่เน็ตเวิร์คที่สร้างเข้าไปยัง Simulator ในหน้าต่างหลักเพื่อทำงาน โดยการเลือกเมนู File → Load NetMap to Simulator

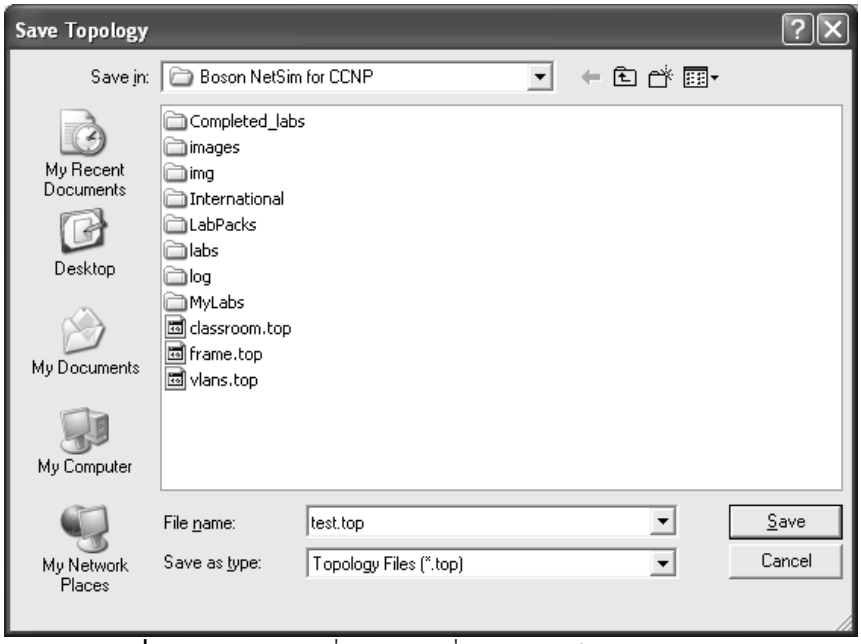


รูปที่ 47 โหลดแผนที่เครือข่ายไปยัง Simulator เพื่อทำการรันทดสอบ



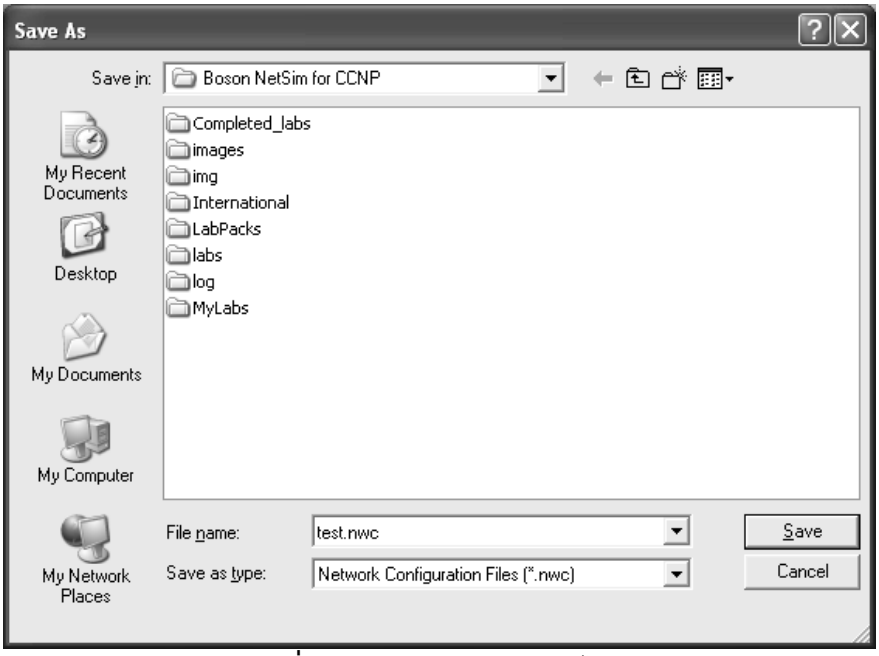
รูปที่ 48 ในหน้าต่างของ Simulator ให้เลือกอุปกรณ์ที่ต้องการคอนฟิก เช่น eRouters -> R1

6. บันทึกแผนที่ของเน็ตเวิร์ค (ซึ่งเป็นไฟล์ .top)



รูปที่ 49 บันทึกแผนที่เครือข่ายที่ออกแบบเป็นนามสกุล .top

7. บันทึกคอนฟิกของอุปกรณ์ทั้งหมด ซึ่งเป็นไฟล์ .rtr หรือ .nwc



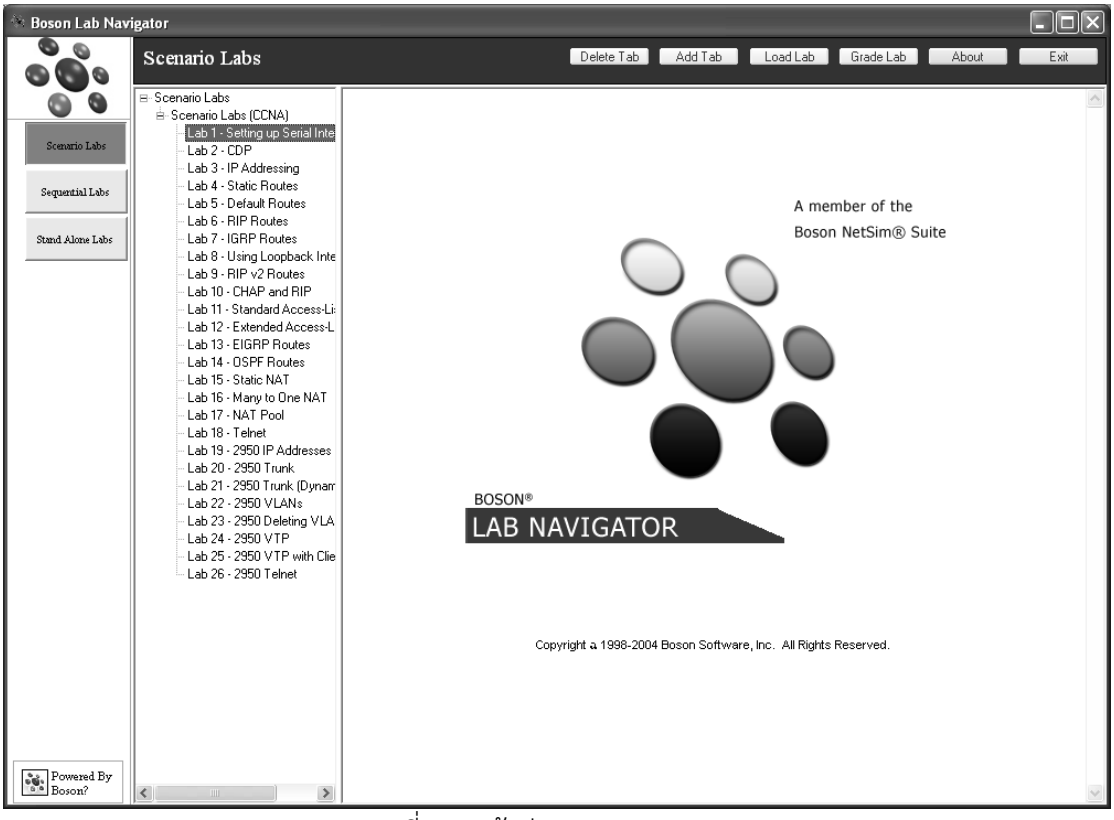
รูปที่ 50 บันทึกคอนฟิกกูเรชันไฟล์

8. ทดสอบการทำงานของเน็ตเวิร์คที่คอนฟิกไว้ว่าถูกต้องหรือไม่
9. บันทึกข้อมูลทั้งหมด

การใช้งาน Lab Navigator

เป็นหน้าต่างที่ใช้สำหรับทดลองการทำแล็ป ซึ่งทาง Boson NetSim ได้บรรจุหลักสูตรและเนื้อหาที่เหมาะสมไว้สำหรับแต่ละระดับของผู้ใช้งาน เช่น NetSim บางรุ่นจะเหมาะสำหรับหลักสูตร CCNA บางรุ่นก็เหมาะสำหรับหลักสูตร CCNP (CCN_ เป็น Certificate ของบริษัท Cisco ที่สอบผ่านแต่ละหลักสูตร เรียงลำดับกันไป ตามความยากง่ายซึ่งมีอยู่หลายหลักสูตร เช่น CCNA, CCNP, CCIE เป็นต้น สนใจอ่านข้อมูลเพิ่มเติมได้จากที่ www.cisco.com) การทดลองทำแล็ปจะมีขั้นตอนดังนี้

1. คลิกเลือกที่ทูล Lab Navigator ในหน้าต่างของ Simulator



รูปที่ 51 หน้าต่าง Lab Navigator

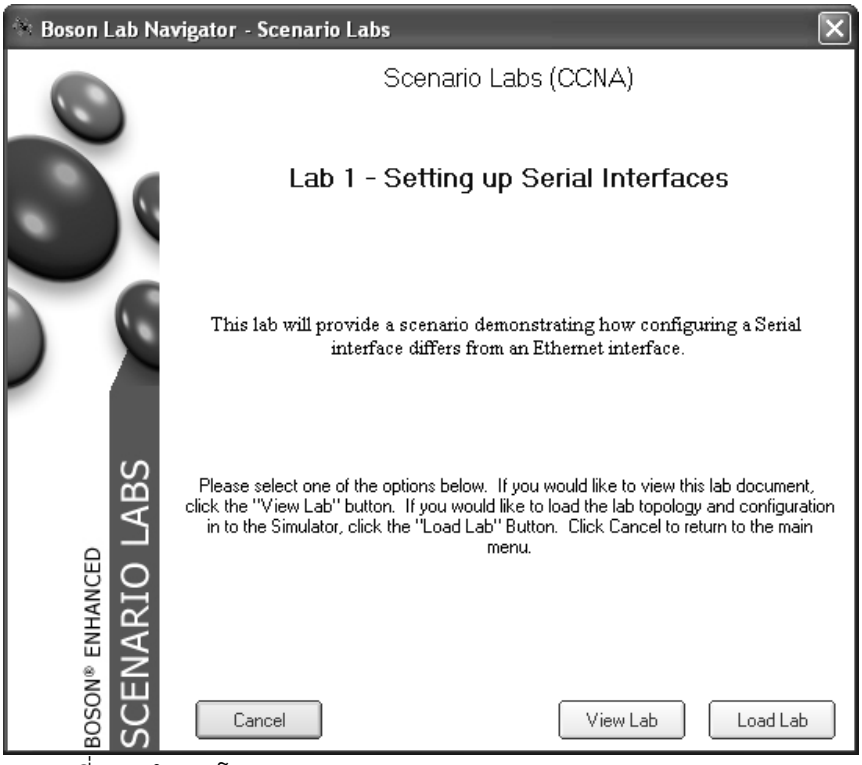
จะมีเมนูให้เลือก 3 เมนูคือ

Scenario Labs เป็นแล็ปที่ใช้สำหรับทดลองการคอนฟิก ซึ่งมีลักษณะเป็นหัวข้อหลัก ๆ หรือเรื่องหลัก ๆ ที่สมควรจะรู้ในเนื้อหาแต่ละหลักสูร เช่น CCNA ก็จะมีเนื้อหาน้อยกว่า CCNP

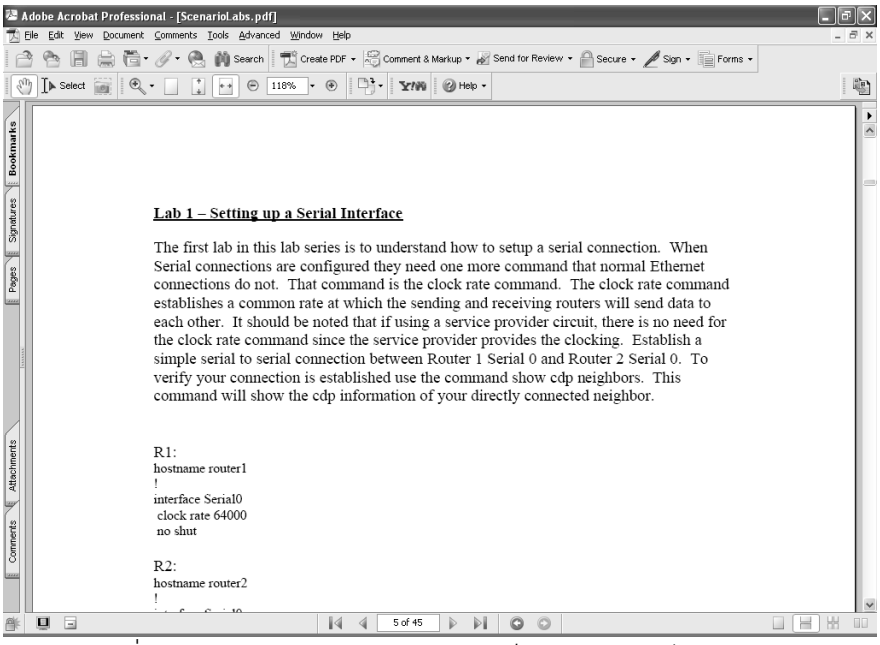
Sequential Labs เป็นแล็ปที่ใช้สำหรับทดลองการคอนฟิก คล้ายกับ Senario Labs แต่แตกต่างตรงที่ Sequential Labs จะเรียงลำดับจากเรื่องที่ง่าย ๆ ก่อนแล้วจึงไปยังเรื่องที่ยาก ๆ ตามลำดับ

Stand Alone Labs เป็นแล็ปที่ใช้ทดลองการคอนฟิก ในลักษณะแยกเป็นข้อ ๆ ซึ่งหัวข้อในแล็ปนี้อาจจะไปปรากฏเป็นส่วนใดส่วนหนึ่งของ Scenatio Labs หรือ Sequential Labs ก็ได้

2. ดับเบิลคลิกแล็ปที่ต้องการทดลอง ซึ่งจะปรากฏเมนูให้เลือกว่าต้องการดูเอกสารที่บอกถึงจุดมุ่งหมายที่ต้องการให้ทดลองในแต่ละแล็ปว่าต้องการอย่างไรบ้าง (View Lab) หรือจะโหลดแผนที่ของเน็ตเวิร์คที่ต้องการทดลองที่ตรงกับแล็ปนั้น ๆ มาทำงาน (Load Lab)

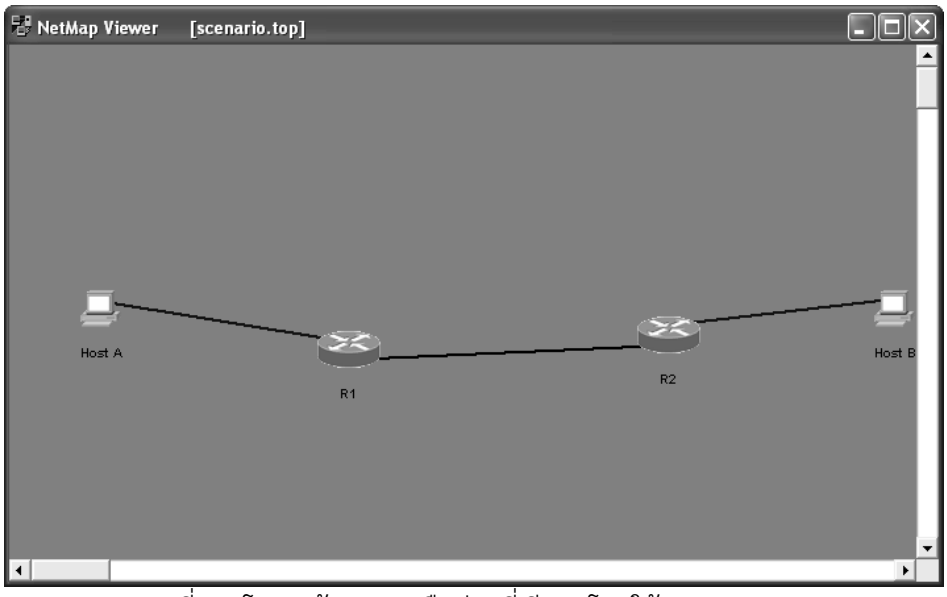


รูปที่ 52 ทำลองโหลด Scenario Labs (Lab 1 Setting up Serial Interfaces)



รูปที่ 53 แสดงจุดมุ่งหมายของแต่ละแล็ปว่าต้องทำอะไรบ้าง (View Lab)

การโหลดข้อมูลของแล็ปนั้น (Load Lab) โปรแกรมจะไปทำงานที่เมนูหลักหรือ Simulator ทันที ขณะที่ผู้ใช้งานอยู่ที่เมนูหลัก (Simulator) และต้องการจะดูโครงสร้างของเครือข่ายให้คลิกเลือกทูล NetMap



รูปที่ 54 โครงสร้างของเครือข่ายที่เรียกดูโดยใช้ทูล NetMap

Profile

Suchart Khummanee

Lecture at Faculty of Informatics, Mahasarakham University,

Kantarawichai, Mahasarakham, 44150 Thailand.

E-mail: suchart.k@msu.ac.th

URL http://www.cs.it.msu.ac.th/

Telephone: +66 (0) 43754333-4 ext. 2500 Fax: +66 (0) 43754359

ACADEMIC QUALIFICATIONS:

- PhD in Computer Engineering, Faculty of Engineering, Khonkaen University Thailand
- MSc in Computer Science, Faculty of Science, Khonkaen University Thailand
- BSc in Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand, 2000

OFFICIAL WEBSITE:

http://cs.it.msu.ac.th/people

CV:

https://drive.google.com/drive/folders/1CeiQMU5rJF98htBNCiQawikO-TVNUyld?usp=sharing