ระบบตรวจสอบและป้องกันการรับส่งข้อมูลคอมพิวเตอร์ผ่านระบบเครือข่าย

HOST-BASED MONITORING AND PROTECTING DATA TRANSFER

วรินทร์ เหล่าเจริญ
ศิวัช อิทธิไกวัล
สุรจินต์ ศิริโสภาพงษ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

HOST-BASED MONITORING AND PROTECTING DATA TRANSFER

WARIN LAOCHAROEN

SIWAT ITTIKAIWAN

SURAJINT SIRISOPARPONG

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LARDKRABANG

ACADEMIC YEAR 2004

หัวข้อโครงงานพิเศษ	ระบบตรวจสอบและป้องกันการรับส่งข้อมูลคอมพิวเตอร์ผ่านระบบเครือข่าย HOST-BASED MONITORING AND PROTECTING DATA TRANSFER
ปริญญา	วิทยาศาสตรบัณฑิต
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2547
อาจารย์ที่ปรึกษา	อ.ศังกรศรัณย์ ล่องชูผล

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2547

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	รศ.ดร.วีระ บุญจริง	
กรรมการ	ผศ.ดร.จีรพร ศรีสวัสดิ์	
กรรมการและอาจารย์ที่ปรึกษา	อ.ศังกรศรัณย์ ล่องชูผล	

(รองศาสตราจารย์ ดร.วีระ บุญจริง)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

หัวข้อโครงงานพิเศษ	ระบบตรวจสอบและป้องกันการรับส่งข้อมูลคอมพิวเตอร์ผ่านระบบเครือข่าย		
ชื่อนักศึกษา	นายวรินทร์	เหล่าเจริญ	44050456
	นายศิวัช	อิทธิไกวัล	44050467
	นายสุรจินต์	ศิริโสภาพงษ์	44050473
ปริญญา	วิทยาศาสตรบัณฑิต		
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์		
สาขา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2547		
อาจารย์ที่ปรึกษา	อ.ศังกรศรัณย์ ล่องชูผล		

บทคัดย่อ

ปัญหาพิเศษนี้เป็นการพัฒนาระบบตรวจสอบและป้องกันการรับส่งข้อมูลคอมพิวเตอร์ผ่านระบบเครือข่าย

ระบบนี้เป็นเครื่องมือที่สามารถตรวจสอบและป้องกันการสื่อสารผ่านพอร์ตที่อาจจะมีช่องโหว่ของระบบรักษาความปลอดภัยได้

ระบบตรวจสอบและป้องกันการรับส่งข้อมูลคอมพิวเตอร์ผ่านระบบเครือข่ายนี้ พัฒนาขึ้นมาโดยใช้ Microsoft Visual C++ .NET ซึ่งสามารถทำงานเชื่อมต่อกับเครือข่ายได้อย่างมีประสิทธิภาพอีกทั้งโปรแกรมที่ได้ออกมายังมีขนาดเล็กและใช้ทรัพยากรระบบน้อย ทำให้โปรแกรมสามารถทำงานได้อย่างมีประสิทธิภาพและรวดเร็ว

Special Project Title	HOST-BASED MONITORING AND PROTECTING DATA TRANSFER		
Student	Mr. Warin	Laocharoen	44050456
	Mr. Siwat	Ittikaiwan	44050467
	Mr. Surajint	Sirisoparpong	44050473
Degree	Bachelor of Science		
Department	Mathematics and Computer Science, Faculty of Science		
Programme	Computer Science		
Academic Year	2004		
Special Project Adviser	Sungkornsarun Longchupole		

ABSTRACT

This special problem is a development of the data transfer monitoring and blocking system for communication port of computers in network.

The system offers utility to monitor and block communication port which is an only security hole.

This system is developed under Microsoft Visual C++ .NET which offers efficient network programming and generates small and resource-friendly executables, thus make the program work fast and efficient.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องระบบตรวจสอบและป้องกันการรับส่งข้อมูลคอมพิวเตอร์ผ่านระบบเครือข่าย สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณอาจารย์ ศังกรศรัณย์ ล่องชูผล อาจารย์ผู้รับผิดชอบปัญหาพิเศษนี้ ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆรวมทั้งเป็นผู้ตรวจสอบความถูกต้องของปัญหาพิเศษนี้

นอกจากนี้คณะผู้จัดทำยังขอขอบพระคุณอาจารย์ทุกท่านที่ได้อบรมสั่งสอนให้ความรู้ทั้งทางทฤษฎีและปฏิบัติแก่คณะผู้จัดทำ ขอขอบพระคุณบิดา มารดาสำหรับทุกสิ่งทุกอย่าง และขอขอบคุณ พี่ๆ และเพื่อนๆ ทุกๆคนที่มีส่วนช่วยเหลือ, ให้คำแนะนำ, คำปรึกษาในด้านต่างๆ เกี่ยวกับปัญหาพิเศษมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

มีนาคม 2548

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันนี้ระบบคอมพิวเตอร์ต่างๆที่เชื่อมต่อกับระบบเครือข่าย ได้ประสบปัญหาเกี่ยวกับระบบการรักษาความปลอดภัยทางด้านข้อมูล ที่เกิดขึ้นจากการรับส่งข้อมูลผ่านทางระบบเน็ตเวิร์ค การรับส่งจดหมายอิเล็กทรอนิกส์ และการค้นหาข้อมูลผ่านทางระบบอินเตอร์เน็ต ซึ่งข้อมูลเหล่านี้สามารถถูกทำให้เกิดความเสียหาย หรือ ถูกเผยแพร่ออกไปยังภายนอกโดยไม่ได้รับอนุญาตได้ ซึ่งเกิดจาก การมีโปรแกรมสอดแนมที่ลักลอบเข้ามาในระบบคอมพิวเตอร์ขณะที่ทำการรับส่งข้อมูล หรือ การที่บุคลากรรู้ไม่เท่าทันโปรแกรมที่ลักลอบส่งข้อมูลโดยไม่ได้รับอนุญาต หรือการถูกโจมตีจากบุคคลภายนอก ทำให้ระบบคอมพิวเตอร์ไม่มีความปลอดภัยในข้อมูลเพียงพอ ทั้งนี้การป้องกันข้อมูลของระบบคอมพิวเตอร์จึงจำเป็นที่จะต้องมีระบบรักษาความปลอดภัยที่มีประสิทธิภาพเพียงพอในการติดตามการรับส่งของข้อมูลที่เกิดขึ้นจากโปรแกรมต่างๆที่ บุคลากรได้ทำการใช้งาน

โครงงานนี้จึงได้จัดทำการพัฒนาระบบรักษาความปลอดภัยในการรับส่งข้อมูล ซึ่งสามารถเพิ่มความปลอดภัยในการรับส่งข้อมูลได้ดีขึ้น และ สะดวกในการใช้งานสำหรับบุคลากรที่ไม่ได้มีความชำนาญในเรื่องดังกล่าวเพียงพอ

1.2 วัตถุประสงค์ของปัญหาพิเศษ

1) เพื่อศึกษาและพัฒนาระบบรักษาความปลอดภัยในการรับส่งข้อมูลผ่านทางระบบเครือข่าย
2) เพื่อให้ผู้ใช้สามารถป้องกันข้อมูลในระบบคอมพิวเตอร์ของตนเอง ได้เองโดยไม่จำเป็นที่จะต้องมีความเชี่ยวชาญในเรื่องดังกล่าว
3) เพื่อศึกษาและพัฒนาระบบที่สะดวกต่อผู้ใช้งานในการรักษาความปลอดภัยในการรับส่งข้อมูล
4) เพื่อศึกษาวิธีการทำงานของระบบตรวจจับผู้บุกรุกและรักษาความปลอดภัยระบบเครือข่าย

1.3 ขอบเขตของปัญหาพิเศษ

1) จัดทำระบบป้องกันการเข้าถึงข้อมูลของระบบคอมพิวเตอร์ที่มีอยู่ในระบบเครือข่าย โดยไม่ได้รับอนุญาตจากระบบคอมพิวเตอร์ที่ติดตั้งโปรแกรม
2) จัดทำระบบป้องกันการส่งข้อมูลออกจากระบบคอมพิวเตอร์ที่ติดตั้งโปรแกรมไปยังระบบเครือข่ายโดยไม่ได้รับอนุญาต
3) จัดทำระบบแจ้งเตือนเมื่อมีโปรแกรมในระบบคอมพิวเตอร์ร้องขอการใช้งานเพื่อเชื่อมต่อเครือข่าย
4) จัดทำระบบบันทึกการให้อนุญาตโปรแกรมในการเข้าใช้งานระบบเครือข่าย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ศึกษาการทำระบบรักษาความปลอดภัยของข้อมูลในระบบคอมพิวเตอร์
2) ได้ศึกษาการรับส่งข้อมูล ของระบบคอมพิวเตอร์ กับ ระบบเครือข่าย
3) ได้ศึกษาแนวทางการป้องกันและการเพิ่มความปลอดภัยให้กับระบบคอมพิวเตอร์
4) ผู้ใช้สามารถนำโครงงานนี้ไปใช้ในการเพิ่มความปลอดภัยของข้อมูลให้กับระบบคอมพิวเตอร์ของตัวเองได้

1.5 ขั้นตอนในการดำเนินการ

1) ศึกษาและทำความเข้าใจในการตรวจสอบการรับส่งข้อมูลระหว่างระบบคอมพิวเตอร์และระบบเครือข่าย
2) เก็บรวบรวมข้อมูลที่เกี่ยวข้องในการตรวจสอบการรับส่งข้อมูล และ การใช้ Win32 API ที่เกี่ยวข้องในการทำโปรแกรม
3) วิเคราะห์และออกแบบโครงสร้างของโปรแกรม
4) ออกแบบอินเตอร์เฟส (interface) ในส่วนติดต่อกับผู้ใช้
5) เขียนโค้ดโปรแกรม (Code Program) ในส่วนต่างๆที่ออกแบบไว้
6) ทดสอบการทำงานของโปรแกรมเพื่อดูว่าโปรแกรมเป็นไปตามที่ตั้งจุดประสงค์ไว้หรือไม่
7) แก้ไขข้อบกพร่องที่เกิดขึ้น และปรับปรุงโปรแกรมให้มีความสมบูรณ์

บทที่ 2
ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

2.1 โปรโตคอล TCP/IP

ปัจจุบันโปรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นโปรโตคอลที่นิยมใช้ในเครือข่ายมากที่สุด เหตุผลหนึ่งที่โปรโตคอลชุดนี้เป็นที่นิยมมาก เนื่องจากหลายบริษัทที่ผลิตอุปกรณ์หรือซอฟต์แวร์ของเครือข่ายนำมาใช้เป็นมาตรฐาน และอีกอย่าง TCP/IP เป็นโปรโตคอลพื้นฐานของเครือข่ายอินเตอร์เน็ต (Internet) ซึ่งเป็นเครือข่ายที่ใหญ่ที่สุดในโลก และเป็นเครือข่ายที่ทำให้คอมพิวเตอร์กลายเป็นส่วนที่สำคัญในชีวิตประจำวันของเราในปัจจุบัน ดังนั้น TCP/IP จึงได้กลายเป็นโปรโตคอลมาตรฐานที่ใช้ในองค์กรธุรกิจและรัฐบาล ในบทนี้เราจะมาศึกษาหลักการทำงานของโปรโตคอลชุดนี้

ชุดโปรโตคอล TCP/IP ได้ถูกพัฒนามาแล้วกว่า 30 ปี ซึ่งเริ่มจากการวิจัยที่สนับสนุนโดยกระทรวงกลาโหมสหรัฐฯ จุดประสงค์ของการวิจัยนี้เพื่อใช้ในการเชื่อมต่อคอมพิวเตอร์ที่ต่างระบบกัน ให้สามารถสื่อสารกันผ่านเครือข่ายได้ ซึ่งสมัยนั้นคอมพิวเตอร์ที่เป็นแพลตฟอร์มเดียวกันเท่านั้นจึงจะสามารถสื่อสารกันผ่านเครือข่ายได้ ดังนั้นจึงได้พัฒนาโปรโตคอลชุดนี้ขึ้น เนื่องจากขั้นตอนการสื่อสารระหว่างคอมพิวเตอร์เป็นสิ่งที่ค่อนข้างซับซ้อน ดังนั้นโปรโตคอลจึงแบ่งเป็นชั้นย่อยหรือเลเยอร์ (Layer) เพื่อเป็นการแยกการทำงานของโปรแกรมประยุกต์ของผู้ใช้ออกจากฮาร์ดแวร์ที่ใช้รับส่งข้อมูลผ่านเครือข่าย โปรโตคอลชุดนี้จะมีการจัดรูปแบบที่แตกต่างจากแบบอ้างอิง OSI เล็กน้อย

เครือข่ายคอมพิวเตอร์ในปัจจุบันประกอบด้วยหลากหลายอุปกรณ์ และฮาร์ดแวร์ที่ผลิตโดยบริษัทต่างๆ แต่อุปกรณ์เครือข่ายเหล่านี้มีระบบการสื่อสารข้อมูลที่เหมือนกัน เครือข่ายที่ใช้อุปกรณ์จากหลายบริษัทนี้สามารถทำงานร่วมกันได้ เนื่องจากอุปกรณ์แต่ละชิ้นผลิตตามมาตรฐานที่กำหนดโดยองค์กรกลาง โปรโตคอล TCP/IP เป็นมาตรฐานที่ได้รับความนิยมมากที่สุดสำหรับเครือข่ายในปัจจุบันเนื่องจากหลายเหตุผลดังนี้

1) เป็นโปรโตคอลระบบเปิด (Open System) ที่ไม่มีบริษัทใดบริษัทหนึ่งเป็นเจ้าของลิขสิทธิ์ และข้อกำหนดของทุกโปรโตคอลจะถูกพัฒนาโดยองค์กรสาธารณะและมีการตีพิมพ์ให้ทราบ
2) TCP/IP ถูกออกแบบมาเพื่อให้แพลตฟอร์มต่างกันสามารถติดต่อสื่อสารกันได้ โปรแกรมบริการต่างๆ เช่น FTP (File Transfer Protocol) และ Telnet เป็นโปรแกรมที่ไม่ขึ้นต่อระบบ เพียงแต่บริษัทนั้นๆพัฒนาระบบของตัวเองให้สามารถรองรับ TCP/IP ได้ก็สามารถสื่อสารกับระบบอื่นได้

3) โปรโตคอล TCP/IP ได้ถูกพิสูจน์แล้วว่าเป็นโปรโตคอลที่แข็งแกร่ง มีประสิทธิภาพสูง และมีความสามารถในการขยายตัวสูง ด้วยการใช้งานในเครือข่ายอินเตอร์เนต ซึ่งเป็นเครือข่ายที่ใหญ่ที่สุดในโลก
4) โปรโตคอล TCP/IP ได้กลายเป็นโปรโตคอลมาตรฐานกลางในการสื่อสารข้อมูลของคอมพิวเตอร์ เนื่องจากเป็นภาษาที่นิยมใช้ในระบบอินเตอร์เนต

ไม่ว่าจะด้วยเหตุผลอะไรก็แล้วแต่ที่ทำให้โปรโตคอล TCP/IP เป็นโปรโตคอลสุดยอดนิยมในปัจจุบัน แต่สิ่งที่สำคัญคือ โปรโตคอลนี้ได้กลายเป็นมาตรฐานเครือข่ายไปโดยปริยาย ดังนั้นเครือข่ายสมัยใหม่จึงจำเป็นที่จะต้องสร้างให้สามารถรองรับโปรโตคอลนี้

2.1.1 TCP/IP และแบบอ้างอิง OSI

โครงสร้างของเครือข่ายเป็นสิ่งที่ซับซ้อนมาก และยากต่อการออกแบบและพัฒนาทั้งระบบโดยคนกลุ่มใดกลุ่มหนึ่ง ดังนั้นจึงมีการแบ่งโครงสร้างออกเป็นชั้นหรือเลเยอร์ (Layer) ซึ่งจะช่วยให้ทั้งผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์พัฒนาผลิตภัณฑ์ของตัวเองได้ โดยไม่ต้องกังวลกับส่วนอื่นๆ แต่ยังสามารถทำงานร่วมกันได้ แบบอ้างอิง OSI (Open System Interconnect) ได้อธิบายสถาปัตยกรรมเครือข่ายโดยแบ่งฟังก์ชันการเคลื่อน ย้ายข้อมูลจากคอมพิวเตอร์เครื่องหนึ่งไปยังคอมพิวเตอร์อีกเครื่องหนึ่งออกเป็น 7 เลเยอร์

การออกแบบชุดโปรโตคอล TCP/IP ก็จะคล้ายๆกับแบบอ้างอิง OSI คือจะแบ่งออกเป็นเลเยอร์เช่นกัน แต่การออกแบบจะมุ่งเน้นไปที่การเชื่อมต่อระหว่างระบบที่ต่างกันมากกว่า ในขณะที่แบบอ้างอิง OSI จะเน้นไปที่การแบ่งการทำงานของโปรโตคอลออกเป็นเลเยอร์ จึงทำ OSI มีโครงสร้างที่ดีกว่า ดังนั้นส่วนใหญ่จะนิยมใช้โปรโตคอล OSI เป็นแบบอ้างอิงในการอธิบายสื่อสารระหว่างคอมพิวเตอร์ในเครือข่าย ในขณะที่ชุดโปรโตคอล TCP/IP เป็นที่นิยมมากกว่าในการนำไปใช้จริง

<table>
<tr><td colspan="2">OSI Reference Model</td><td colspan="3">TCP/IP</td></tr>
<tr><td>7</td><td>Application</td><td rowspan="2">Application</td><td colspan="2" rowspan="2">FTP, Telnet, HTTP, SMPT,
SNMP, DNS, etc.</td></tr>
<tr><td>6</td><td>Presentation</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Session</td><td rowspan="2">Host-to-Host</td><td rowspan="2">TCP</td><td rowspan="2">UDP</td></tr>
<tr><td>4</td><td>Transport</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Network</td><td rowspan="2">Internet</td><td>ICMP</td><td rowspan="2">ARP,
RARP</td></tr>
<tr><td>IP</td></tr>
<tr><td>2</td><td>Data Link</td><td rowspan="2">Network Access</td><td colspan="2" rowspan="2">Not Specified</td></tr>
<tr><td>1</td><td>Physical</td></tr>
</table>

รูปที่ 2.1 เปรียบเทียบแบบอ้างอิง OSI และ TCP/IP

รูปที่ 2.1 แสดงการเปรียบเทียบชั้นของโปรโตคอลระหว่างแบบอ้างอิง OSI และ TCP/IP ซึ่งแบ่งโปรโตคอลออกเป็น 4 เลเยอร์คือ ชั้นประยุกต์ใช้งาน (Application Layer) ชั้นเชื่อมต่อระหว่างโฮสต์ (Host-to-Host Layer) ชั้นอินเตอร์เน็ต (Internet Layer) และชั้นเข้าใช้เครือข่าย (Network Access Layer) การเปรียบเทียบการทำงานของโปรโตคอล TCP/IP กับแบบอ้างอิง OSI นั้นอาจไม่ตรงมากนัก เพราะมีบางโปรโตคอลของ TCP/IP ที่ทำงานมากกว่าชั้นหนึ่ง แต่รูปดังกล่าวก็เปรียบเทียบให้เห็นภาพพอคร่าวๆ

หลักการทำงานของโปรโตคอล TCP/IP สรุปได้คร่าวๆดังนี้คือ การสื่อสารจะเริ่มจากโปรแกรมประยุกต์ของผู้ใช้ส่งข้อมูลให้กับโปรโตคอลในชั้นโปรแกรมประยุกต์ หลังจากนั้นชั้นโปรแกรมประยุกต์จะเพิ่มข้อมูลส่วนหัวซึ่งจะประกอบด้วยชื่อของคอมพิวเตอร์ที่ต้องการสื่อสารด้วยและหมายเลขพอร์ตของเครื่องนั้น ข้อมูลก็จะถูกส่งต่อไปยังชั้นเชื่อมต่อโฮสต์ ซึ่งอาจจะใช้โปรโตคอล TCP หรือ UDP ขึ้นอยู่กับโปรแกรมประยุกต์ที่ใช้ เมื่อชั้นนี้ได้รับข้อมูลก็จะแบ่งข้อมูลออกเป็นส่วนย่อยๆ ซึ่งแต่ละส่วนจะถูกเพิ่มข้อมูลส่วนหัวเข้าไป ข้อมูลส่วนย่อยๆ นี้จะเรียกว่า “เซ็กเมนต์ (Segment)”

ข้อมูลส่วนหัวของแต่ละเซ็กเมนต์จะถูกเพิ่มเข้าไปอย่างเหมาะสม หลังจากนั้นแต่ละเซ็กเมนต์ก็จะถูกส่งต่อไปให้ชั้นอินเตอร์เน็ต เมื่อข้อมูลมาถึงชั้นนี้ก็จะถูกเพิ่มข้อมูลส่วนหัวให้แต่ละเซ็กเมนต์เช่นกัน ข้อมูลที่เพิ่มเข้าไป เช่น หมายเลข IP ประเภทของโปรโตคอลที่ใช้ (TCP หรือ UDP)

และ Checksum เป็นต้น ถ้าข้อมูลที่ส่งมามีการแบ่งย่อยอีกก็จะมีการเพิ่มข้อมูลที่เกี่ยวกับการแบ่งย่อยนี้เพิ่มเข้าไปด้วย ชุดข้อมูลที่อยู่ในชั้นนี้จะเรียกว่า "แพ็กเก็ต" หลังจากนั้นแต่ละแพ็กเก็ตข้อมูลก็ส่งต่อไปให้ชั้นเข้าใช้เครือข่าย เพื่อทำการส่งข้อมูลไปตามช่องสื่อสารต่อไป เมื่อแพ็กเก็ตเดินทางไปถึงที่หมาย เครื่องปลายทางก็จะทำตามขั้นตอนที่ตรงกันข้ามกับเครื่องส่ง และข้อมูลก็จะถูกส่งผ่านต่อไปให้โปรแกรมประยุกต์เพื่อนำข้อมูลไปโพรเซสต่อไป

ในชุดโปรโตคอล TCP/IP นั้นประกอบด้วยโปรโตคอลย่อยที่ทำงานในเลเยอร์ต่างๆ หลายโปรโตคอล ดังแสดงในรูปที่ 1 ซึ่งรายละเอียดเกี่ยวกับการทำงานของแต่ละโปรโตคอลจะได้กล่าวในหัวข้อต่อไป

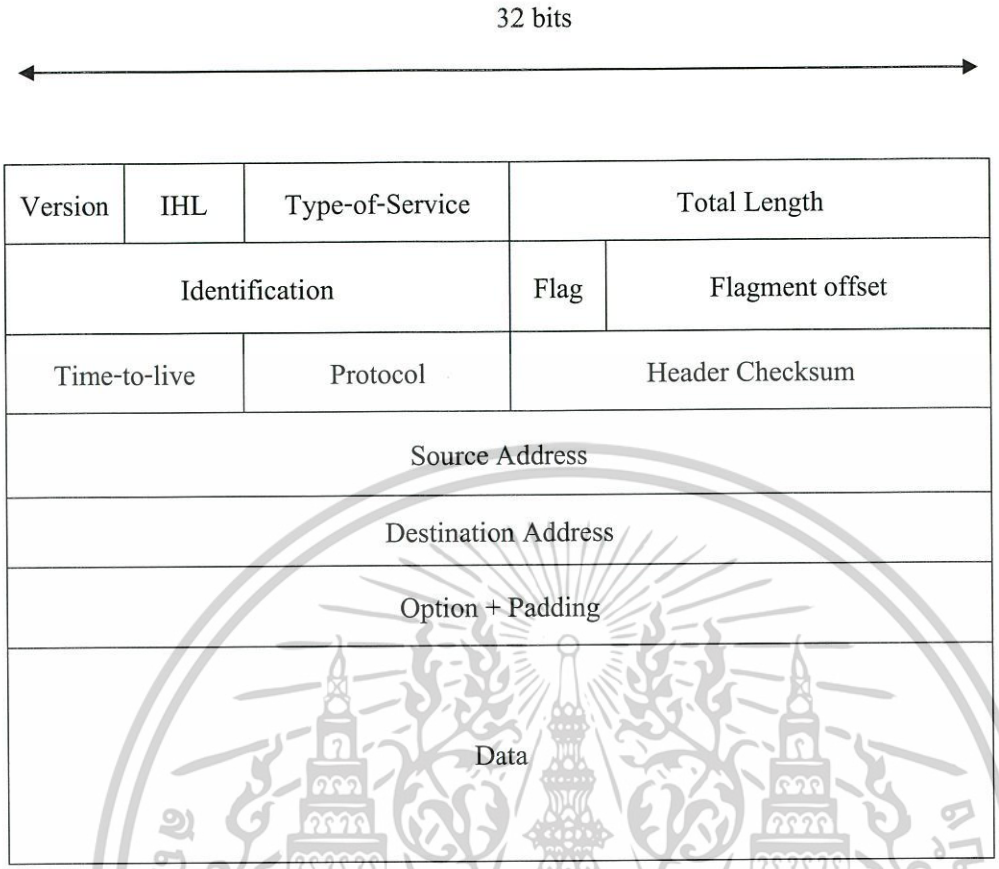
2.1.2 อินเตอร์เน็ตเลเยอร์ (Internet Layer)

โปรโตคอลที่สำคัญที่ทำงานในเลเยอร์อินเตอร์เน็ตคือ IP,ARP,ICMP และ IGMP เราควรที่จะทำความเข้าใจหลักการทำงานของโปรโตคอลนี้

2.1.2.1 Internet Protocol (IP)

โปรโตคอล IP ทำหน้าที่เสมือนกับที่ทำการไปรษณีย์ กล่าวคือ โปรโตคอล IP จะทำหน้าที่จัดการเกี่ยวกับการรับส่งแพ็กเก็ต หรือบางทีก็เรียกว่า "ดาต้าแกรม (Datagram)" คือหน่วยของข้อมูลที่ได้รับมาจากโปรโตคอลที่อยู่เลเยอร์สูงกว่า เช่น TCP และ UDP ถ้าโฮสต์ปลายทางอยู่คนละเครือข่ายกับโฮสต์ที่ส่งข้อมูล IP จะรับผิดชอบในการจัดเส้นทาง (Routing) ให้แพ็กเก็ตส่งไปยังเครือข่ายเหล่านั้นโดยทั่วไปแล้วอุปกรณ์ที่ทำหน้าที่รับส่งข้อมูลระหว่างเครือข่ายจะเรียกว่าเร้าท์เตอร์ แต่บางทีอุปกรณ์ตัวนี้ก็จะเรียกว่า "เกตเวย์ (Gateway)" ซึ่งทำหน้าที่เป็นเสมือนประตูไปยังเครือข่ายอื่นๆ อย่างไรก็ตามทั้งเราท์เตอและเกตเวย์เป็นอุปกรณ์ที่ทำหน้าที่ในเลเยอร์ที่ 3 เหมือนกัน

โปรโตคอล IP เป็นโปรโตคอลที่ให้บริการแบบคอนเน็กชันเลสส์ (Connectionless) ซึ่งทำให้มีความเชื่อถือน้อย เนื่องจากไม่มีการสร้างการเชื่อมต่อก่อนที่จะทำการรับส่งข้อมูล กล่าวคือ ในการส่งข้อมูลแต่ละครั้ง โฮสต์ส่งจะไม่ทำการติดต่อโฮสต์ปลายทางเพื่อตกลงเกี่ยวกับการรับส่งข้อมูลก่อน แต่โฮสต์ที่ต้องการส่งข้อมูลจะทำการส่งแพ็กเก็ตออกไปทันที โดยที่คาดหวังว่าโฮสต์ปลายทางจะได้รับแพ็กเก็ตนั้นในที่สุด ดังนั้นความเชื่อถือในการส่งข้อมูลจึงมีน้อย เพราะแพ็กเก็ตอาจสูญหายระหว่างทาง หรือถ้าข้อมูลประกอบด้วยหลายแพ็กเก็ต แต่ละแพ็กเก็ตอาจเดินทางมาถึงปลายทางไม่เป็นลำดับได้ หรือมีการส่งแพ็กเก็ตซ้ำกันหรือแพ็กเก็ตส่งถึงล่าช้า การแก้ปัญหานี้จะปล่อยให้เป็นหน้าที่ของโปรโตคอลที่อยู่ในเลเยอร์ที่สูงกว่ารับผิดชอบ



รูปที่ 2.2 ฟอร์แมตของแพ็กเก็ต IP

ฟอร์แมตของแพ็กเก็ต IP ประกอบด้วยหลายฟิลด์ดังแสดงใน รูปที่ 2.2 ข้อมูลในส่วนหัวของแพ็กเก็ต IP มีดังนี้

1) Version (4บิต) : ข้อมูล 4 บิตแรกจะเป็นข้อมูลที่บอกถึงเวอร์ชันของโปรโตคอล IP ที่ใช้ ซึ่งในปัจจุบันจะใช้เวอร์ชัน 4 หรือเรียกสั้นๆ ว่า IPv4 ในอนาคตอันใกล้อาจจะมีการเปลี่ยนไปใช้เวอร์ชันใหม่คือ เวอร์ชัน 6 หรือ IPv6 เนื่องจากเวอร์ชัน 4 กำลังมีปัญหาเกี่ยวกับหมายเลข IP ไม่เพียงพอต่อการใช้งาน
2) Internet Header Length หรือ IHL (4บิต) : เป็นตัวเลขที่บอกความยาวของข้อมูลในส่วนหัว (Header)
3) Type of Service (8บิต) : ในแต่ละบิตของส่วนข้อมูลนี้จะเป็นธงหรือแฟลก (Flag) ที่แสดงถึงลำดับความสำคัญ (precedence), ความล่าช้า (Delay), อัตราส่งผ่าน (Throughput) และค่ากำหนดความเชื่อถือได้ของแพ็กเก็ตข้อมูลนี้
4) Total Length (16บิต) : ข้อมูลส่วนนี้จะบอกถึงความยาวแพ็กเก็ตทั้งหมดซึ่งมีหน่วยเป็นไบต์ ซึ่งความยาวของแพ็กเก็ตนี้เป็นไปได้ตั้งแต่ 576 – 65,536 ไบต์

5) Identifier (16บิต) : ถ้าดาต้าแกรมประกอบด้วยหลายแพ็กเก็ต หมายเลขนี้จะถูกกำหนดให้กับแต่ละแพ็กเก็ตย่อย ซึ่งแพ็กเก็ตแต่ละแพ็กเก็ตจะมีหมายเลขนี้ที่ไม่ซ้ำกับหมายเลขแพ็กเก็ต อื่นในช่วงเวลานั้นๆ
6) Flag (3บิต) : เป็นฟิลด์ที่ใช้ในการจัดการเกี่ยวกับการแบ่งข้อมูลเป็นแพ็กเก็ตย่อย
7) Flagment Offset (13บิต) : เป็นค่าที่บอกจุดเริ่มต้นในส่วนของข้อมูลย่อย (Fragmented Content) ซึ่งเป็นตัวเลขที่บอกว่าแพ็กเก็ตย่อยนี้อยู่ห่างจากจุดเริ่มต้นของดาต้าแกรมทั้งหมดเท่าใด โดยจำนวนนี้มีหน่วยวัดเป็น 64 บิต
8) Time to Live หรือ TTL (8บิต) : แพ็กเก็ตจะไหลเวียนอยู่ในเครือข่ายได้ในช่วงเวลาหนึ่งเท่านั้น การกำหนดว่าแพ็กเก็ตแต่ละแพ็กเก็ตจะอยู่ได้ในเครือข่ายนานเท่าใดนั้น จะบอกเป็นจำนวนของ Hop หรือจำนวนครั้งที่ผ่านเราท์เตอร์ ทุกครั้งที่ผ่านเราท์เตอร์ค่า TTL จะลดลงทีละหนึ่ง เมื่อค่านี้เป็นศูนย์แพ็กเก็ตนี้ก็จะถูกละทิ้งไป
9) Protocol (8บิต) : เป็นข้อมูลที่บอกโปรโตคอลของชั้นที่เหนือกว่า เช่น TCP, UDP, VINES เป็นต้น
10) Header Checksum (16บิต) : จะเป็นข้อมูลส่วนที่ใช้ในการตรวจสอบข้อผิดพลาดในส่วนเฮดเดอร์ของแพ็กเก็ตซึ่งเมื่อผ่านอุปกรณ์เครือข่ายแต่ละครั้งก็จะทำการเช็คข้อผิดพลาดทุกครั้งไป
11) Source IP Address (32บิต) : หมายเลข IP ของเครื่องที่ส่งข้อมูล
12) Destination IP Address (32บิต) : หมายเลข IP ของเครื่องปลายทาง
13) Padding : เป็นเลข 0 ที่เพิ่มให้กับส่วนหัวของแพ็กเก็ตเพื่อให้ส่วนหัวมีความยาวที่หารด้วย 32 บิตลงตัว หรือเป็นข้อมูลเกี่ยวกับฟีเจอร์อื่นๆ เช่น การรักษาความปลอดภัย
14) Data : ข้อมูลของโปรโตคอลที่อยู่สูงกว่าซึ่งความยาวจะไม่คงที่

2.1.2.2 Address Resolution Protocol (ARP)

การที่คอมพิวเตอร์ที่อยู่ในเครือข่ายเดียวกันต้องการที่จะสื่อสารกันจำเป็นที่ต้องทราบหมายเลขเน็ตเวิร์คการ์ด หรือแม็กแอดเดรส (MAC Address) ของกันและกัน แพ็กเก็ตไอพีจะถูกห่อหุ้มด้วยเฟรมในระดับดาต้าลิงค์ ซึ่งแม็กแอดเดรสของเครื่องส่งและเครื่องรับจะต้องถูกใส่ไปด้วย ปัญหาก็คือเครื่องส่งอาจไม่ทราบหมายเลขแม็กแอดเดรสของเครื่องรับ

โปรโตคอล ARP (Adress Resolution Protocol) จะทำหน้าที่ค้นหาหมายเลขแม็กแอดเดรสของเครื่องที่มีหมายเลขไอพีที่ต้องการ หลักการทำงานของ ARP คือ โฮสต์ที่ต้องการทราบหมายเลขแม็กแอดเดรสของเครื่องที่มีหมายเลขไอพีนั้น จะทำการบรอดคาสต์แพ็กเก็ตไปยังคอมพิวเตอร์ทุกเครื่องที่อยู่ในเครือข่ายเดียวกัน ถ้ามีเครื่องที่มีหมายเลขไอพีดังกล่าว เครื่องนั้นก็จะตอบกลับพร้อม

หมายเลขแม็กแอดเดรสของเครื่องนั้น หลังจากนั้นเครื่องส่งก็สามารถสื่อสารกับเครื่องปลายทางได้โดยตรง โดยใช้แม็กแอดเดรสที่ส่งมาพร้อมกับแพ็กเก็ตตอบกลับ

ส่วนโปรโตคอลที่ทำหน้าที่ตรงกันข้ามกับโปรโตคอล ARP คือโปรโตคอล RARP (Reverse Address Resolution Protocol) ซึ่งโปรโตคอลนี้จะช่วยให้โฮสต์ที่รู้หมายเลขแม็กแอดเดรสแต่ไม่รู้หมายเลขไอพี

2.1.2.3 Internet Control Message Protocol (ICMP)

โปรโตคอล ICMP (Internet Control Message Protocol) ทำหน้าที่รายงานข้อผิดพลาดต่างๆ ที่เกิดขึ้นในระหว่างที่มีการส่งแพ็กเก็ตในเครือข่าย ICMP ใช้ในการส่งแบบคอนเน็กชันเลสต์ (Connectionless) ซึ่งหมายถึงการรับส่งข้อมูลที่ฝ่ายรับและฝ่ายส่งไม่ได้ประสานกันก่อน กล่าวคือ ฝ่ายรับจะไม่ทราบว่ามีแพ็กเก็ตส่งมาหาตัวเอง ดังนั้นโอกาสที่แพ็กเก็ตจะส่งไม่ถึงปลายทางจึงเป็นไปได้สูง

โปรโตคอล ICMP จะทำหน้าที่รายงานข้อผิดพลาดบางอย่างที่เกิดขึ้นในระหว่างการส่งข้อมูล ฟังก์ชันที่สำคัญของโปรโตคอลนี้ เช่น

1) ประกาศข้อผิดพลาดของเครือข่าย เช่น โฮสต์ หรือบางส่วนของเครือข่ายไม่สามารถติดต่อได้เนื่องจากสาเหตุบางอย่าง โปรโตคอล TCP หรือ UDP ที่ส่งไปยังโฮสต์ที่ไม่มีโพรเซสรอรับข้อมูลที่พอร์ตนั้นก็จะถูกรายงานโดย ICM เช่นกัน
2) ประกาศความคับคั่งของเครือข่าย เช่น ในกรณีที่เราท์เตอร์ได้รับแพ็กเก็ตมากเกินกว่าที่จะรองรับได้ เราท์เตอร์ก็จะส่งข้อความ ICMP Source Quench เพื่อแจ้งให้สถานีส่งทราบเกี่ยวกับสถานะ
3) ช่วยในการค้นหาข้อผิดพลาด โดย ICMP จะรองรับฟังก์ชัน Echo ซึ่งเป็นการส่งแพ็กเก็ตไปกลับระหว่างสองโฮสต์ใดๆ Ping เป็นเครื่องมือที่ใช้ฟังก์ชันนี้ โดย Ping จะส่งแพ็กเก็ตหลายๆ แพ็กเก็ตไปยังโฮสต์ปลายทางและวันเวลาที่แพ็กเก็ตเดินทางไปกลับแล้วหาค่าเฉลี่ยและอัตราสูญหายของแพ็กเก็ต
4) ประกาศการหมดเวลา กล่าวคือ เมื่อแพ็กเก็ต IP ไหลเวียนอยู่ในเครือข่ายนานเกินไป ค่า TTL ก็จะกลายเป็นศูนย์ในที่สุด เราท์เตอร์ก็จะละทิ้งแพ็กเก็ตนี้และรายงานข้อผิดพลาดนี้ผ่าน ICMP เครื่องมือที่ใช้ฟังก์ชันนี้คือ Tranceroute โดยจะส่งแพ็กเก็ตที่มีค่า TTL ต่ำแล้วก็ข้อความ ICMP ที่ประกาศเกี่ยวกับการหมดเวลาของ TTL ของแพ็กเก็ตที่ส่งออกไป

ฟังก์ชันต่างๆ ของ ICMP จะมีประโยชน์มากสำหรับการวิเคราะห์และค้นหาจุดเสียของระบบ

2.1.2.4 Internet Group Management Protocol (IGMP)

โปรโตคอล IGMP (Internet Group Management Protocol) ทำหน้าที่แจ้งให้เราท์เตอร์ทราบเกี่ยวกับกลุ่มของเลขหมายไอพีที่เป็นมัลติคาสต์ (Multicast) ซึ่งข้อมูลนี้จะถูกส่งต่อๆ กันไปยังเราท์เตอร์ต่างๆ ที่อยู่ในเครือข่ายเพื่อให้เครือข่ายสามารถรองรับข้อมูลแบบมัลติคาสต์ได้ การส่งแพ็กเก็ตของ ICMP จะส่งเป็นไอพีดาต้าแกรมซึ่งเป็นการส่งแบบคอนเนคชันเลสส์

2.1.3 โฮสต์ทูโฮสต์เลเยอร์ (Host to Host Layer)

อย่างที่ได้กล่าวมาแล้วข้างต้น โปรโตคอลโฮสต์ทูโฮสต์เลเยอร์ (Host-to-Host Layer) นี้จะประกอบด้วย 2 โปรโตคอลคือ TCP (Transmission Control Protocol) และ UDP (User Datagram Protocol) ซึ่งโปรโตคอลแต่ละตัวจะให้บริการที่ต่างกัน และมีข้อดีข้อเสียต่างกัน โปรโตคอลทั้งสองตัวมีรายละเอียดดังนี้

2.1.3.1 Transmission Control Protocol (TCP)

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่ให้บริการแบบคอนเน็กชันโอเรียนเต็ด (Connection-Oriented) ซึ่งเป็นการส่งข้อมูลที่เชื่อถือได้ TCP จะส่งข้อมูลมั้งหมดจนสำเร็จ ซึ่งถ้าข้อมูลมีขนาดใหญ่ก็จะถูกแบ่งย่อยเป็นหลายแพ็กเก็ต โปรโตคอล TCP จะทำหน้าที่ควบคุมการรับส่งแพ็กเก็ตย่อยๆ เหล่านี้ สำหรับกลไกในการควบคุมการไหลของข้อมูลมีรายละเอียดดังนี้

1) การจัดการเกี่ยวกับเซสชั่น

เนื่องจาก TCP เป็นโปรโตคอลที่ให้บริการแบบคอนเน็กชันโอเรียนเต็ด ดังนั้นก่อนที่จะมีการส่งข้อมูลจำเป็นที่จะต้องสร้างเซสชั่นเพื่อเชื่อมต่อกับโฮสต์ปลายทางก่อน เซสชั่นเป็นการสร้างการสนทนาอย่างเป็นรูปแบบระหว่างทั้งสองโฮสต์เพื่อใช้สำหรับกู้คืนข้อมูลเมื่อเกิดข้อผิดพลาดระหว่างการรับส่งข้อมูล ขั้นตอนในการสร้างเซสชั่นนี้จะมีอยู่ 3 ขั้นตอนซึ่งบางทีก็เรียกว่า “ทรีเวย์แฮนเช็ค (Three-Way Handshake)”

1. โฮสต์ที่ต้องการส่งข้อมูลจะส่งแพ็กเก็ตไปยังโฮสต์ปลายทางเพื่อแจ้งให้ทราบว่าต้องการส่งข้อมูล
2. โฮสต์ปลายทางก็จะตอบตกลงกลับมาพร้อมทั้งรหัสที่จะใช้ในการรับส่งข้อมูล
3. โฮสต์ต้นทางก็จะส่งแพ็กเก็ตพร้อมรหัสที่ได้รับ เพื่อเป็นการยืนยันการเชื่อมต่อ

หลังจากที่ได้มีการสร้างเซสชั่นสำเร็จแล้วถึงจะเริ่มขบวนการรับส่งข้อมูลจริงๆ ซึ่งการรับส่งข้อมูลแต่ละครั้งก็จะมีการยืนยันรับข้อมูลจากโฮสต์ปลายทางทุกครั้ง เมื่อรับส่งข้อมูลเสร็จก็เป็นขั้นตอนการยกเลิกเซสชั่น ซึ่งจะคล้ายๆ กับการสร้างเซสชั่น

2) การควบคุมการไหลและการกู้คืนข้อมูล

ในแต่ละเซสชั่น โฮสต์ฝ่ายรับต้องตอบกลับทุกๆ แพ็กเก็ตที่ได้รับภายในเวลาที่กำหนด เพื่อเป็นการยืนยันการรับข้อมูลทุกๆ แพ็กเก็ตที่ส่ง ฝ่ายรับจะทำการเช็คความถูกต้องของแพ็กเก็ตข้อมูลทุกครั้ง และแจ้งให้ทราบถึงผลการตรวจสอบนั้น ถ้าฝ่ายส่งไม่ได้รับการตอบรับจากฝ่ายรับภายในเวลาที่กำหนด ฝ่ายรับก็จะคาดเอาว่าแพ็กเก็ตสูญหายระหว่างทาง ฝ่ายรับก็จะทำการส่งแพ็กเก็ตนั้นใหม่อีกครั้ง เพื่อทำให้มั่นใจได้ว่าข้อมูลทุกๆ แพ็กเก็ตส่งถึงปลายทางอย่างสมบูรณ์ นอกจากนี้การแบ่งข้อมูลขนาดใหญ่ออกเป็นแพ็กเก็ตย่อยๆ TCP ก็จะกำหนดหมายเลขลำดับ (Sequence Number) ให้แต่ละแพ็กเก็ต เพื่อใช้สำหรับการจัดรวมแพ็กเก็ตย่อยๆ เหล่านั้นให้เป็นข้อมูลเหมือนเดิม นอกจากนี้หมายเลขลำดับยังใช้สำหรับการตรวจสอบว่าข้อมูลส่งถึงปลายทางครบทุกแพ็กเก็ตหรือไม่

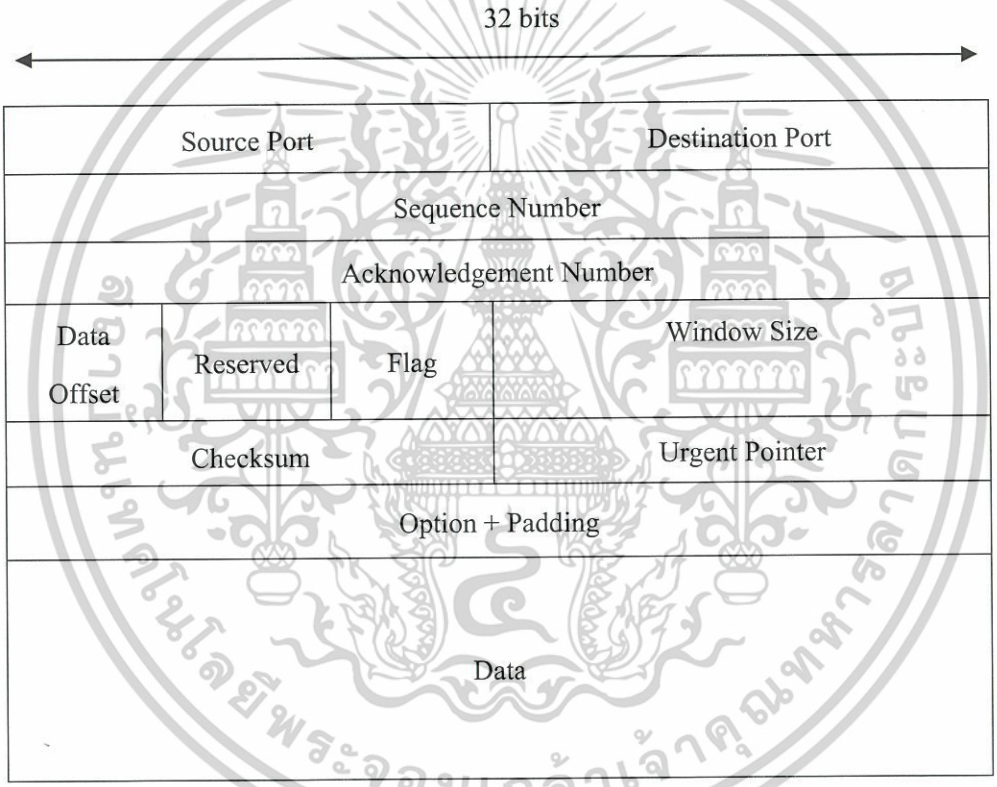
กลไกการตอบกลับแพ็กเก็ตนั้นมีอยู่ 2 ประเภท ประเภทแรกคือ PAR (Positive Acknowledgment and Retransmission) กลไกการทำงานคือ เมื่อฝ่ายส่งทำการส่งแพ็กเก็ตหนึ่ง ก็จะรอการตอบกลับจากฝ่ายรับ แล้วค่อยส่งแพ็กเก็ตต่อไป ถ้าไม่ได้รับการตอบกลับภายในเวลาที่กำหนดก็จะส่งแพ็กเก็ตนั้นอีกครั้ง ปัญหาของกลไกนี้ก็คือ ถ้าข้อมูลประกอบด้วยหลายๆ แพ็กเก็ต และการที่ฝ่ายรับต้องส่งแพ็กเก็ตตอบรับต่อทุกๆ แพ็กเก็ตที่ได้รับนั้นอาจเป็นการสิ้นเปลืองแบนด์วิธ และเป็นขบวนการที่ไร้ประสิทธิภาพเนื่องจากฝ่ายส่งจะใช้เวลาในการรอมากกว่าการส่งข้อมูล กลไกที่สองที่จะแก้ปัญหานี้ซึ่งกลไกนี้จะเรียกว่า “สไลดิ้งวินโดว์ (Sliding Window)” กลไกการทำงานคือ ฝ่ายรับสามารถยืนยันการรับแพ็กเก็ตโดยส่งแพ็กเก็ตเดียวสำหรับการยืนยันการรับหลายแพ็กเก็ต วิธีนี้จะช่วยลดจำนวนแพ็กเก็ตที่ต้องไหลเวียนในเครือข่าย และฝ่ายส่งสามารถส่งทีละหลายๆ แพ็กเก็ตก่อนที่จะรอการตอบรับ

เมื่อสร้างเซสชั่นสำเร็จ ขั้นตอนต่อไปคือการต่อรองเกี่ยวกับขนาดของวินโดว์ (Window Size) ขนาดของวินโดว์คือ จำนวนไบต์ที่ฝ่ายรับได้รับก่อนที่จะทำการตอบกลับ หรือจำนวนไบต์ที่ฝ่ายส่งสามารถส่งได้ก่อนที่จะรอการตอบกลับ การทำงานของสไลดิ้งวินโดว์มีขั้นตอนดังนี้

1. เมื่อโฮสต์ต้องการที่จะส่งข้อมูล TCP จะย้ายข้อมูลไปไว้ที่บัฟเฟอร์ที่จะใช้ส่งข้อมูล ซึ่งข้อมูลส่วนนี้จะเรียกว่า “เซ็กเมนต์ (Segment)” ซึ่งแต่ละเซ็กเมนต์อาจจะถูกแบ่งย่อยเป็นหลายแพ็กเก็ตซึ่งแต่ละแพ็กเก็ตก็จะถูกกำหนดหมายเลขลำดับ
2. ทุกๆ แพ็กเก็ตในเซ็กเมนต์จะถูกส่งต่อไปให้โปรโตคอล IP เพื่อทำการส่งไปยังโฮสต์ปลายทาง
3. เซ็กเมนต์ข้อมูลจะยังคงถูกเก็บไว้ที่บัฟเฟอร์จนกว่าจะได้รับการตอบรับจากโฮสต์ฝ่ายรับก่อนและโฮสต์ฝ่ายส่งจะตั้งเวลาเพื่อรอการตอบกลับ ถ้าโฮสต์ฝ่ายรับไม่ตอบกลับภายในเวลาที่กำหนด ข้อมูลที่อยู่ในบัฟเฟอร์ก็จะถูกส่งใหม่อีกครั้ง

4. เมื่อแพ็กเก็ตเดินทางมาถึงฝ่ายรับ โฮสต์ฝ่ายรับก็จะใช้หมายเลขลำดับในการเรียบเรียงแพ็กเก็ตให้ได้เป็นเซ็กเมนต์เหมือนเดิม
5. เมื่อโฮสต์ฝ่ายรับได้รับแพ็กเก็ตครบและตรวจสอบแล้วไม่มีข้อผิดพลาดใดๆ ก็จะส่งแพ็กเก็ตตอบกลับไปยังโฮต์ฝ่ายรับว่าได้รับข้อมูลครบหมดแล้ว
6. เมื่อโฮสต์ฝ่ายรับได้รับการตอบกลับ เซ็กเมนต์ในบัฟเฟอร์ก็จะถูกลบทิ้งไปแล้วทำการส่งเซ็กเมนต์ถัดไปถ้ามี จนกว่าข้อมูลจะถูกส่งทั้งหมด

ขบวนการส่งข้อมูลแบบนี้จะทำให้มั่นใจได้ว่าข้อมูลจะส่งถึงปลายทางอย่างแน่นอนและถูกต้อง ซึ่งการให้บริการแบบนี้จะเรียกว่า “คอนเน็กชันโอเรียนเต็ด (Connection-Oriented) นั่นเอง



รูปที่ 2.3 ฟอร์แมตข้อมูลของแพ็กเก็ต TCP

ข้อมูลในส่วนหัวของโปรโตคอล TCP จะประกอบด้วยข้อมูลมากที่สุด 20 ไบต์ และประกอบด้วยส่วนต่างๆ ดังแสดงในรูปที่ 2.3 ซึ่งแต่ละฟิลด์มีความหมายดังนี้

1) TCP Source Port (16บิต) : ส่วนนี้จะเป็นหมายเลขพอร์ตที่เป็นจุดเริ่มต้นการสื่อสาร หมายเลขพอร์ตเมื่อรวมกับหมายเลข IP จะเป็นที่อยู่ของการส่งข้อมูลกลับ

2) TCP Destination Port (16บิต) : เป็นหมายเลขพอร์ตของเครื่องรับ ซึ่งพอร์ตนี้จะเป็นพอร์ตที่ใช้เชื่อมต่อกับโปรแกรมประยุกต์ที่เราจะนำข้อมูลที่ส่งไปให้นี้ไปโพรเซสต่อไป

3) TCP Sequence Number (32บิต) : เป็นหมายเลขที่บอกลำดับแพ็กเก็ตที่จะใช้ โดยฝั่งเครื่องรับในการเรียงข้อมูลให้อยู่ในรูปแบบเดิม ในการส่งข้อมูลผ่านเครือข่ายที่สลับซับซ้อนนั้นแพ็กเก็ตแต่ละชุดอาจจะถูกส่งไปบนเส้นทางที่ต่างกัน ดังนั้นจึงเป็นไปได้ที่แพ็กเก็ตจะเดินทางมาถึงปลายทางไม่เป็นไปตามลำดับที่ส่ง หมายเลขนี้จะใช้ในการจัดเรียงแพ็กเก็ตเหล่านี้อยู่ในลำดับเดิม

4) TCP Acknowledgement Number (32บิต) : เป็นหมายเลขลำดับแพ็กเก็ตถัดไปที่ทางฝั่งรับคาดหวังซึ่งเป็นการบอกเป็นนัยว่าแพ็กเก็ตที่มีหมายเลขลำดับก่อนหน้านี้ได้รับหมดแล้วนั่นเอง

5) Data Offset (4บิต) : เป็นตัวเลขที่บอกขนาดของข้อมูลส่วนหัว (TCP Header) ซึ่งมีหน่วยเป็น 32 บิต หรือ word

6) Reserved (6บิต) : ส่วนนี้จะถูกกำหนดให้เป็นศูนย์ตลอด ซึ่งข้อมูลในส่วนนี้ไม่มีความหมายอะไรเพียงแต่เป็นการสงวนไว้ใช้ในอนาคตเมื่อมีการปรับปรุงโปรโตคอล

7) Flags (6บิต) : เป็นข้อมูลที่ใช้สำหรับควบคุมการรับส่งแพ็กเก็ต เช่น บิต SYN และ ACK ใช้สำหรับการสร้างการเชื่อมต่อ ส่วนบิต FIN เป็นการยกเลิกการเชื่อมต่อ เป็นต้น

8) Window Size (16บิต) : เป็นตัวเลขที่เครื่องปลายทางบอกให้เครื่องต้นทางทราบขนาดวินโดว์ที่เครื่องปลายทางสามารถรับข้อมูลได้

9) Checksum (16บิต) : เป็นข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของข้อมูลในส่วนหัว โดยเครื่องส่งจะทำการคำนวณค่า เช็คซัม (Checksum) ของข้อมูลส่วนหัว เมื่อเครื่องปลายทางได้รับข้อมูลก็จะทำการคำนวณเช็คซัมด้วยวิธีเดียวกัน แล้วทำการเปรียบเทียบข้อมูลค่าที่คำนวณได้กับค่าที่อยู่ในฟิลด์นี้ ถ้าเหมือนกันแสดงว่าไม่มีข้อผิดพลาดในข้อมูลที่ได้รับ

10) Padding : เป็นข้อมูลที่เพิ่มเพื่อให้ข้อมูลส่วนหัวมีจำนวนบิตที่หารด้วย 32 ลงตัว

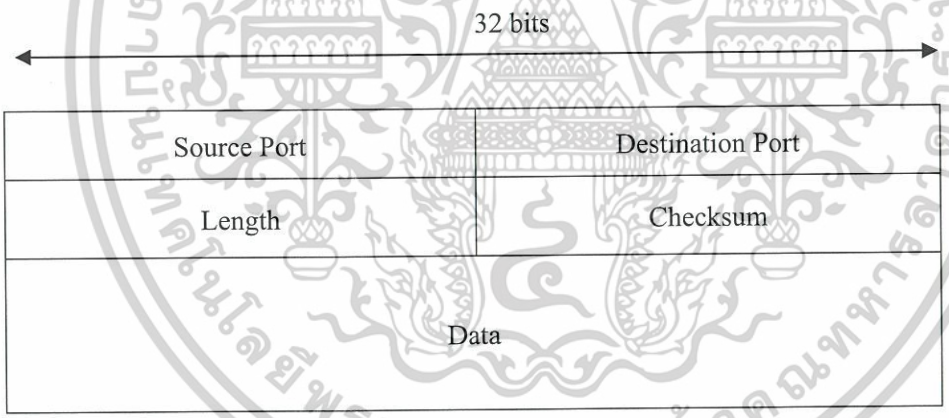
2.1.3.2 User Datagram Protocol (UDP)

โปรโตคอล UDP (User Datagram Protocol) จะให้บริการส่งข้อมูลแบบคอนเน็กชันเลส หรือ บางทีก็เรียกว่า “ดาต้าแกรม (Datagram)” ซึ่งเป็นการให้บริการแบบตรงกันข้ามกับคอนเน็กชันโอเรียนเต็ด ของโปรโตคอล TCP การส่งข้อมูลแบบนี้จะเป็นแบบที่เชื่อถือไม่ได้ โดยพยายามส่งข้อมูลให้ดีที่สุด ในการรับส่งข้อมูลแต่ละครั้งนั้น จะไม่มีการสร้างเซสชั่นก่อน และไม่มีกลไกการตอบกลับแพ็กเก็ตเหมือนโปรโตคอล TCP เหตุที่ตัดกลไกนี้ออกเพื่อเพิ่มประสิทธิภาพในการส่ง

ข้อมูลนั่นเอง แต่ข้อเสียก็คือ การรับส่งเชื่อถือไม่ได้ เพราะแพ็กเก็ตอาจสูญหายระหว่างทางซึ่งทางฝ่ายส่งจะไม่ทราบเลย ดังนั้นโปรโตคอลที่อยู่เหนือกว่าต้องรับผิดชอบเกี่ยวกับการตรวจสอบข้อผิดพลาดของการรับส่งข้อมูลเอง

ถึงแม้ว่าโปรโตคอล UDP จะมีความเชื่อถือน้อย แต่มันก็มีหลายอย่าง เช่น ถ้าข้อมูลที่เราต้องการส่งมีขนาดเล็กมากก็จะเป็นการเสียเวลา ถ้าต้องสร้างเซสชั่นการเชื่อมต่อระหว่าง 2 โฮสต์นั้นก่อนส่ง และอีกกรณีหนึ่งคือ การส่งข้อมูลแบบแพร่กระจาย หรือบรอดคาสต์ (Broadcast) และมัลติคาสต์ (Multicast) การสร้างเซสชั่นจะเป็นส่งที่เป็นไปไม่ได้ เนื่องจากเซสชั่นเป็นการเชื่อมต่อระหว่าง 2 โฮสต์เท่านั้น ดังนั้นการบรอดคาสต์และมัลติคาสต์จะใช้โปรโตคอล UDP เท่านั้น

เนื่องจากโปรโตคอล UDP ไม่รับรองว่าข้อมูลจะส่งถึงปลายทาง ดังนั้นถ้าโปรแกรมประยุกต์ที่ใช้โปรโตคอลนี้ต้องการความเชื่อถือได้ของการส่งข้อมูล โปรแกรมประยุกต์นั้นจะต้องควบคุมและตรวจสอบข้อผิดพลาดเอง ส่วนใหญ่โปรแกรมประยุกต์ที่ใช้โปโตคอลนี้จะไม่ต้องการการยืนยันการตอบรับ เช่น โปรโตคอลของระบบเครือข่ายของไมโครซอฟต์ เช่น การล็อกออน (Log on) การบราวซิ่ง (Browsing) และเนมรีโซลูชั่น (Name Resolution) เป็นต้น



รูปที่ 2.4 ฟอร์แมตข้อมูลของแพ็กเก็ต UDP

ข้อมูลในส่วนหัวของโปรโตคอล UDP นั้นแสดงในรูปที่ 2.4 ซึ่งแต่ละฟิลด์มีความหมายดังนี้

1) UDP Source Port Number (16บิต) : เป็นหมายเลขพอร์ตของเครื่องส่ง เมื่อรวมหมายเลขพอร์ตนี้กับหมายเลข IP ก็จะเป็นที่อยู่สำหรับเครื่องรับในการตอบกลับข้อความ
2) UDP Destination Port Number (16บิต) : เป็นหมายเลขพอร์ตของทางฝั่งเครื่องรับที่ใช้ในการส่งผ่านข้อมูลไปยังโปรแกรมประยุกต์ที่ต้องการติดต่อด้วย

3) UDP Checksum (16บิต) : เป็นข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของข้อมูล เครื่องทางฝั่งรับจะทำการคำนวณหมายเลขนี้ด้วยวิธีเดียวกัน แล้วเปรียบเทียบกับค่าที่ส่งมา ถ้าหมายเลขเท่ากันแสดงว่าไม่มีข้อผิดพลาดในข้อมูลส่วนหัว
4) UDP Message Length (16บิต) : เป็นข้อมูลที่บอกความยาวของข้อมูลทั้งหมด ซึ่งจะเป็นข้อมูลที่ช่วยให้ทางฝ่ายรับทราบว่าข้อมูลควรมีขนาดเท่าใด

2.1.3.3 หมายเลขพอร์ต

คอมพิวเตอร์ที่ใช้โปรโตคอล TCP/IP ส่วนใหญ่จะมีโปรแกรมประยุกต์หลายตัวที่ใช้โปรโตคอล TCP/IP ในการสื่อสารกับเครื่องอื่น ซึ่งโปรโตคอล TCP/IP จะจัดการส่งข้อมูลไปยังโปรแกรมประยุกต์ที่เหมาะสม เพื่อให้ TCP/IP สามารถรองรับโปรแกรมประยุกต์หลายโปรแกรมประยุกต์ในเครื่องเดียว จึงมีการใช้พอร์ตและซ็อกเก็ต (Port and Socket) เพื่อช่วยในการแยกแยะโปรแกรมประยุกต์ต่างๆ ในรูปที่ 2.5 แสดงการใช้พอร์ตและซ็อกเก็ตในการรับส่งข้อมูลของโปรแกรมประยุกต์ประเภทต่างๆ

รูปที่ 2.5 พอร์ตและซ็อกเก็ต

โปรแกรมประยุกต์แต่ละตัวที่จะรับส่งข้อมูลผ่านเครือข่ายจะใช้หมายเลขพอร์ตตั้งแต่ 0 ถึง 65,536 ดังนั้นเพื่อให้การรับส่งข้อมูลถูกต้อง โปรแกรมประยุกต์ที่รันในเครื่องเดียวกันจะต้องใช้หมายเลขพอร์ตต่างกันเพื่อช่วยลดความสับสน โปรแกรมประยุกต์ที่นิยมใช้กันทั่วไปส่วนใหญ่จะถูกกำหนดให้ใช้หมายเลขพอร์ตใดพอร์ตหนึ่ง ซึ่งองค์กรที่ทำหน้าที่กำหนดหมายเลขนี้คือ IANA

(Internet Assigned Numbers Authority) หมายเลขพอร์ตเหล่านี้จะถูกตีพิมพ์ใน RFC 1700 ซึ่งพอร์ตสำหรับโปรแกรมประยุกต์ที่นิยมใช้ทั่วไปได้แสดงในตารางข้างล่าง

ตาราง 2.1 หมายเลขพอร์ตของบางโปรแกรมประยุกต์

พอร์ต	โปรโตคอล	โปรแกรมประยุกต์
20	TCP	FTP (Data)
21	TCP	FTP (Control)
23	TCP	Telnet
25	TCP	SMTP (อีเมล)
53	TCP/UDP	DNS (Domain Name System)
80	TCP	HTTP (เว็บเซิร์ฟเวอร์)
110	TCP	POP3 (อีเมล)
161	UDP	SNMP (Simple Network Management Protocol)

โปรโตคอล TCP/IP จะแยกแยะโปรแกรมประยุกต์ที่รันในแต่ละโฮสต์ โดยใช้ข้อมูล 3 ส่วนต่อไปนี้

1. หมายเลขไอพีของโฮสต์นั้น
2. ประเภทของโปรโตคอลชั้นทรานสปอร์ต : TCP หรือ UDP
3. หมายเลขพอร์ตที่โปรแกรมประยุกต์นั้นใช้

ตาราง 2.1 แสดงหมายเลขพอร์ตที่ใช้โดยโปรแกรมประยุกต์ที่เป็นที่รู้จักโดยทั่วไป ส่วนในเครื่องไคลเอนท์ที่เชื่อมต่อกับเซิร์ฟเวอร์จะใช้หมายเลขพอร์ตที่ต่างจากเซิร์ฟเวอร์ ซึ่งหมายเลขพอร์ตที่ใช้บนเครื่องไคลเอนท์จะถูกจัดการโดยระบบปฏิบัติการของเครื่องนั้น กล่าวคือ ถ้าไคลเอนท์จะเชื่อมต่อกับเว็บเซิร์ฟเวอร์ หมายเลขพอร์ตของเซิร์ฟเวอร์ที่ใช้คือ พอร์ต 80 แต่เครื่องไคลเอนท์จะใช้หมายเลขพอร์ตอื่นที่ว่าง

2.1.4 แอพพลิเคชันเลเยอร์

การทำงานของโปรโตคอลในชั้นนี้จะเป็นการเข้าใช้ทรัพยากรระยะไกล (Remote Access) และการแชร์การใช้ทรัพยากร (Resource Sharing) โปรโตคอลโปรแกรมประยุกต์ที่จัดอยู่ในชั้นนี้ได้แก่

1) HTTP (Hyper Text Transfer Protocol) : ใช้สำหรับการรับส่งไฟล์เว็บเพจระหว่างเว็บบราวเซอร์และเว็บเซิร์ฟเวอร์

2) SMTP (Simple Mail Transfer Protocol) : ใช้สำหรับการรับส่งอีเมลระหว่างเมลเซิร์ฟเวอร์

3) POP (Post Office Protocol) : ใช้สำหรับการดาวน์โหลดอีเมลจากเมลเซิร์ฟเวอร์

4) IMAP (Internet Message Access Protocol) : ใช้สำหรับการดาวโหลดอีเมลจากเมลเซิร์ฟเวอร์

5) FTP (File Transfer Protocol) : ใช้สำหรับถ่ายโอนไฟล์ระหว่างโฮสต์

6) Telnet : ใช้ในการล็อกอินเข้าใช้โฮสต์ระยะไกล

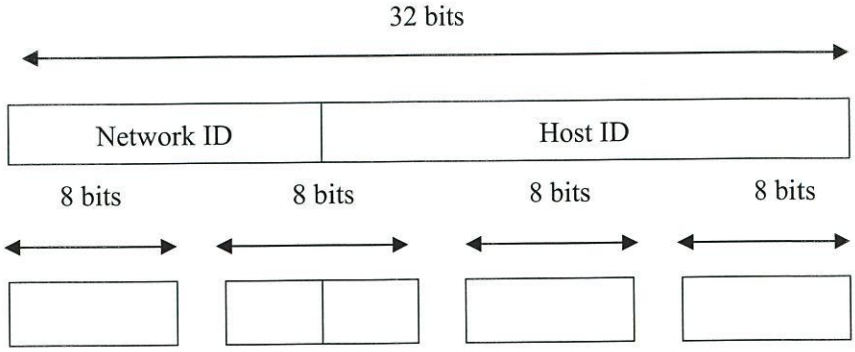
2.2 IP Addressing

หมายเลขไอพี (IP Address) คือเลขที่บอกที่อยู่เฉพาะของโหนดหรือโฮสต์ที่อยู่ในเครือข่าย รวมถึงคอมพิวเตอร์และเราท์เตอร์ (Router) ที่อยู่บนระบบเครือข่าย หมายเลขนี้จะเป็นที่อยู่ในเลเยอร์ที่ 3 หรือชั้นเครือข่ายหมายเลขไอพีของแต่ละเครื่องที่อยู่ในเครือข่ายเดียวกันจะต้องไม่ซ้ำกัน อย่างไรก็ตามโฮสต์หนึ่งอาจจะมีหมายเลขไอพีได้มากกว่าหนึ่งเลขหมายก็ได้ ซึ่งอาจจะมีประโยชน์ในการจัดการวงข่าย เช่น เราท์เตอร์ หรือ เกตเวย์ เป็นต้น

ปัจจุบันโปรโตคอล IP ที่ใช้งานอยู่ในเครือข่ายอินเตอร์เน็ตจะเป็นเวอร์ชัน 4 หรือเรียกสั้นๆ ว่า "IPv4" ซึ่งในเวอร์ชันนี้หมายเลขไอพีจะขนาด 32 บิต เนื่องจากเลขฐานสอง 32 บิตเป็นตัวเลขที่ยาวและยากต่อการจดจำ ดังนั้นเพื่อเป็นการง่ายหมายเลขไอพีจึงนิยมเขียนให้อยู่ในรูปแบบด็อตเดซิมอล (Dotted Decimal Notation) การเขียนให้อยู่ในรูปแบบนี้จะทำได้โดยการจัดกลุ่มเลขฐานสองออกเป็น 4 กลุ่มๆ ละ 8 บิต หลังจากนั้นให้แปลงเลขฐานสองของแต่ละกลุ่มให้เป็นเลขฐานสิบ เมื่อแปลงเสร็จแล้วให้เอาเลขทั้งสี่ตัวมารวมกันโดนใช้จุดเป็นตัวเชื่อม ตัวอย่างเช่น เลขไอพี 10101100 00010100 00000001 00011000 เขียนให้อยู่ในรูปแบบด็อตเดซิมอลได้เป็น 172.20.1.24 เป็นต้น ดังตัวอย่างนี้จะเห็นได้ว่าหมายเลขไอพีที่อยู่ในรูปด็อตเดซิมอลจะง่ายต่อการจำมากกว่าหมายเลขไอพีที่อยู่ในเลขฐานสอง เนื่องจากหมายเลขไอพีที่เป็นเลขฐานสิบนี้เป็นการแปลงมาจากเลขฐานสอง 8 บิต ดังนั้นเลขฐานสิบแต่ละตัวจะอยู่ระหว่าง 0 ถึง 255 เพราะฉะนั้นหมายเลขไอพีที่ถูกต้องจะอยู่ระหว่าง 0.0.0.0 ถึง 255.255.255.255

2.2.1 ประเภทของหมายเลขไอพี

โปรโตคอลไอพีเวอร์ชัน 4 (IPv4) ที่ใช้อยู่ในปัจจุบันจะแบ่งหมายเลขไอพีออกเป็น 5 ประเภท (Class) คือ A,B,C,D และ E โดยหมายเลขไอพีทั้ง 32 บิต จะถูกจัดให้เป็น 2 กลุ่มดังนี้คือ กลุ่มแรกจะเป็นตัวเลขที่ใช้บอกหมายเลขเครือข่าย (Network ID) และกลุ่มที่สองจะเป็นตัวเลขที่ใช้บอกหมายเลขโฮสต์ที่อยู่ในเครือข่ายนั้น ดังแสดงใน รูปที่ 2.6 ข้างล่าง ส่วนรูปที่ 2.7 แสดงการจัดประเภทของหมายเลขไอพี



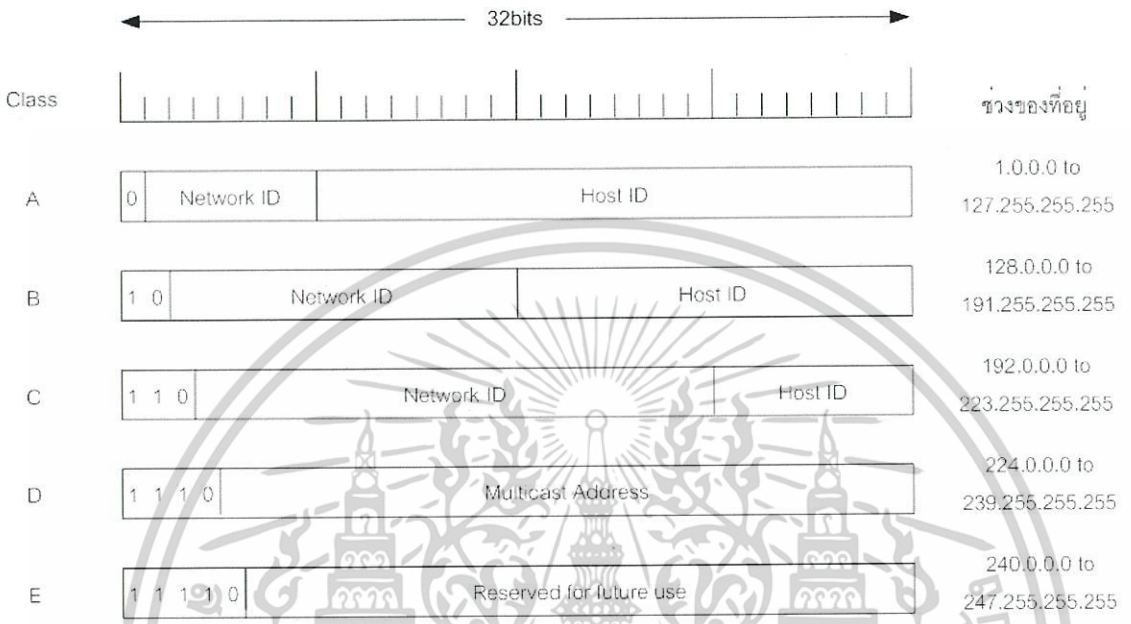
Dotted Decimal Notation 172 • 20 • 122 • 254

รูปที่ 2.6 การแบ่งส่วนของหมายเลขไอพีและด็อตเดซิมอลโนเตชัน

ข้อกำหนดที่ใช้ในการแบ่งประเภทของหมายเลขไอพีมีดังนี้

1) Class A : สำหรับหมายเลขไอพีประเภท A หรือคลาสเอนั้น บิตแรกจะเป็นเลข 0 เท่านั้น และส่วนที่บอกหมายเลขเครือข่าย (Network ID) คือ 8 บิตแรก ดังนั้นจะมีได้ทั้งหมด 126 เครือข่าย (หมายเลขเครือข่าย 0 จะไม่ใช้) ส่วนอีก 24 บิตที่เหลือจะเป็นเลขที่ใช้บอกหมายเลขโฮสต์ (Host ID) ดังนั้นในแต่ละเครือข่ายจะมีโฮสต์ทั้งหมด 16,777,224 เครื่อง (หมายเลข 0.0.0 และ 255.255.255 จะไม่ใช้) เนื่องจากเครือข่ายมีจำนวนน้อยมากเมื่อเทียบกับจำนวนโฮสต์ ฉะนั้นหมายเลขไอพีประเภทนี้จึงไม่เหมาะสำหรับเครือข่ายขนาดใหญ่ ซึ่งประกอบด้วยหลายเครือข่ายเชื่อมต่อกัน เช่น ระบบอินเตอร์เน็ต เพราะในการส่งข้อมูลระหว่างเครือข่ายนั้นเราท์เตอร์จะใช้เฉพาะหมายเลขเครือข่ายเท่านั้น ดังนั้นเครือข่ายประเภทนี้จึงเหมาะสำหรับเครือข่ายส่วนบุคคลมากกว่า
2) Class B : สำหรับหมายเลขไอพีประเภท B นั้นสองบิตแรกจะเป็น 10 เท่านั้น ส่วนหมายเลขเครือข่ายจะใช้ 16 บิตแรก ดังนั้นจะมีจำนวนเครือข่ายได้ทั้งหมด 16,382 เครือข่าย ส่วนอีก 16 บิตที่เหลือเป็นหมายเลขโฮสต์ ซึ่งจะทำให้ในแต่ละเครือข่ายมีโฮสต์ได้ทั้งหมด 65,534 เครื่อง
3) Class C :สำหรับประเภท C นั้นจะมีบิตเริ่มต้นเป็น 110 และเมื่อรวมกับอีก 21 บิตต่อมาก็จะเป็นหมายเลขเครือข่าย ซึ่งจะได้ทั้งหมด 2,097,145 เครือข่าย ส่วน 8 บิตสุดท้ายเป็นหมายเลขโฮสต์ ซึ่งมีทั้งหมด 254 เครื่อง
4) Class D : ส่วนประเภทที่สี่คือ เลขไอพีที่เริ่มต้นด้วย 1110 ซึ่งจะเป็นเลขไอพีที่ใช้สำหรับการมัลติคาสต์ (Multicasting) หรือสำหรับการส่งข้อมูลแบบมีโฮสต์ปลายทางหลายเครื่องแต่อาจจะอยู่คนละเครือข่ายกัน

5) ประเภทสุดท้าย คือ เลขไอพีที่เริ่มต้นด้วย 11110 เป็นหมายเลขที่สงวนไว้ใช้ในอนาคต หมายเลขเหล่านี้จะถูกกำหนดให้โดยศูนย์ข้อมูลเครือข่าย หรือ InterNIC (Internet Network Information Center)



รูปที่ 2.7 การแบ่งประเภทของ IP Address

จะเห็นได้ว่าในหมายเลขไอพีแต่ละประเภทจะมีหลายหมายเลขที่สงวนไว้ใช้สำหรับกรณีพิเศษ เช่น หมายเลข 0.0.0.0 จะใช้โดยโฮสต์ในขณะที่เริ่มเปิดเครื่อง (Boot up) เลขไอพีที่มีหมายเลขเครือข่ายเป็น 0 ทั้งหมดจะใช้อ้างถึงเครือข่ายที่โฮสต์นั้นอยู่ เช่น ถ้าโฮสต์หนึ่งมีเลขไอพีเป็น 172.20.1.24 ซึ่งจะจัดอยู่ในประเภท B โดยมีหมายเลขเครือข่ายเป็น 172.20 เมื่อโฮสต์นี้อ้างถึงหมายเลขไอพี 0.0.1.32 จะมีความหมายเช่นเดียวกับหมายเลข 172.20.1.32 เป็นต้น ส่วนหมายเลขไอพีที่ประกอบด้วยเลข 1 ทั้ง 32 บิตนั้น (255.255.255.255) จะถูกใช้สำหรับการส่งข้อมูลแบบแพร่กระจาย (Broadcast) ในเครือข่ายนั้นๆ ส่วนเลขไอพี 172.20.255.255 เป็นเลขที่ใช้สำหรับการส่งข้อมูลแบบแพร่กระจายไปยังโฮสต์ทุกเครื่องที่อยู่ในเครือข่าย 172.20 เป็นต้น ส่วนเลขหมาย 127.xx.yy.zz โดยเลข xx, yy และ zz จะเป็นเลขอะไรก็ได้ (แต่โดยทั่วไปจะนิยมใช้เป็นเลข 127.0.0.1) นั้น จะใช้สำหรับการส่งข้อมูลไปยังตัวเอง (Loopback) ซึ่งข้อมูลจะไม่ถูกส่งออกไปในเครือข่ายแต่จะถูกโพรเซสภายในเครื่องเท่านั้น แต่โฮสต์นั้นจะถือเสมือนว่าข้อมูลนั้นได้รับผ่านเครือข่าย ประโยชน์ของการทำเช่นนี้ เช่น การทดสอบซอฟต์แวร์ที่ต้องส่งข้อมูลผ่านเครือข่าย แต่ไม่ต้องการส่งข้อมูลออกไปรบกวนระบบเครือข่ายจริง ทำให้ลดความคับคั่งในเครือข่ายลงได้

2.2.2 Private/Public Internet

การที่จะเชื่อมต่อคอมพิวเตอร์เข้ากับอินเตอร์เน็ตนั้นจำเป็นที่ต้องร้องขอหมายเลขไอพีจากอินเตอร์นิก (InterNIC) เหราะอินเตอร์เน็ตจะเป็นองค์กรที่รับผิดชอบเกี่ยวกับการแจกจ่ายหมายเลขไอพี และเป็นที่ทราบกันดีแล้วว่าหมายเลขไอพีของแต่ละเครื่องที่อยู่ในเครือข่ายเดียวกันจะต้องไม่ซ้ำกัน ดังนั้นถ้าเป็นเครือข่ายส่วนบุคคลที่ไม่มีการเชื่อมต่อเข้าอินเตอร์เน็ต ก็สามารถใช้หมายเลขไอพีอะไรก็ได้ และไม่ต้องร้องขอหมายเลขไอพีจากอินเตอร์นิก เพียงแค่กำหนดให้หมายเลขไอพีของโฮสต์ในเครือข่ายไม่ซ้ำกันก็พอ

อย่างไรก็ตาม เมื่อมีการเชื่อมต่อเครือข่ายส่วนบุคคลเข้ากับอินเตอร์เน็ต อาจจะทำให้หมายเลขไอพีที่ใช้ในเครือข่ายส่วนบุคคลไปซ้ำกับโฮสต์ที่อยู่ในอินเตอร์เน็ตแล้วก็ได้ ดังนั้นเพื่อป้องกันปัญหาดังกล่าว องค์กร IETF (Internet Engineering Task Force) ได้กำหนดหมายเลขไอพีบางกลุ่มให้เป็นหมายเลขไอพีส่วนบุคคล ดังแสดงในตาราง 2.2

ตาราง 2.2 หมายเลขไอพีเครือข่ายส่วนบุคคล

ประเภท	ไอพีต่ำสุด	ไอพีสูงสุด
A	10.0.0.0	10.255.255.255
B	172.16.0.0	173.31.255.255
C	192.168.0.0	192.168.255.255

หมายเลขที่แสดงในตาราง 2.2 จะเป็นหมายเลขที่ไม่ใช้ในเครือข่ายอินเตอร์เน็ต กล่าวคือแพ็กเก็ตที่มีหมายเลขไอพีนี้จะไม่ถูกส่งต่อโดยเราท์เตอร์ของระบบอินเตอร์เน็ต ส่วนหมายเลขไอพีที่แจกจ่ายไปแล้วสามารถดูรายละเอียดได้ที่ http://www.iana.org/assignments/ipv4-address-space

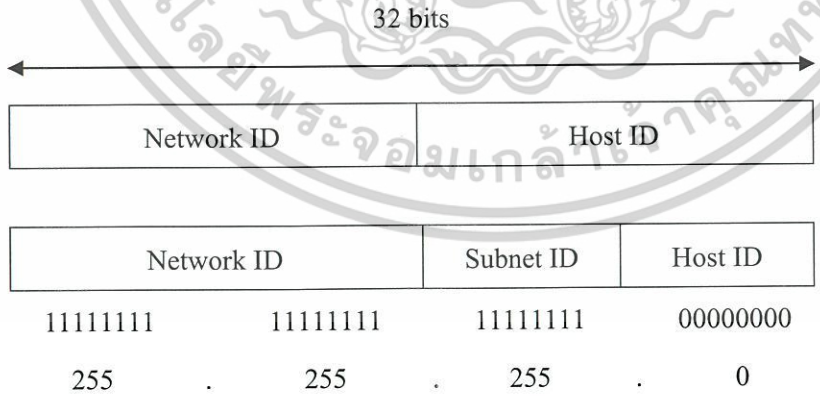
2.2.3 ซับเน็ตและซับเน็ตมาส์ค (Subnet and Subnet Mask)

ในระบบอินเตอร์เน็ตนั้นศูนย์ข้อมูลเครือข่าย หรือ InterNIC (Internet Network Information Center) จะเป็นองค์กรที่รับผิดชอบเกี่ยวกับการแจกจ่ายหมายเลขไอพี เนื่องจากอินเตอร์เน็ตเติบโตเร็วมาก ทำให้มีปัญหาเกี่ยวกับการใช้หมายเลขไอพี เพราะในไม่ช้าหมายเลขไอพีที่มีทั้งหมด 32 บิต อาจถูกใช้หมดได้นั่นคือ ต้องมีวิธีการใหม่ที่จะกำหนดหมายเลขไอพีให้กับโฮสต์ วิธีหนึ่งที่ใช้แก้ปัญหาคือ เพิ่มจำนวนบิตให้มีมากกว่า 32 บิต ในขณะนี้กำลังมีการพัฒนาโปรโตคอลไอพีเวอร์ชันใหม่ (IPv6) ซึ่งในเวอร์ชันนี้จะกำหนดให้หมายเลขไอพีมีขนาด 128 บิต และเมื่อมีการประกาศใช้แล้วจำนวนหมายเลขไอพีจะมีเพียงพออย่างแน่นอน แต่การที่จะอัพเกรดอินเตอร์เน็ตให้

ใช้โปรโตคอลเวอร์ชันนี้ได้นั้นคงต้องใช้เวลาอีกนานพอสมควร เพราะเป็นการเปลี่ยนแปลงที่ใหญ่มาก

การแบ่งซับเน็ต (Subneting) จะเป็นอีกวิธีหนึ่งที่ช่วยให้การใช้หมายเลขไอพีให้คุ้มค่ามากยิ่งขึ้น เนื่องจากหมายเลขไอพีบางประเภท เช่น คลาส B จะมีโฮสต์ได้ทั้งหมด 65,534 เครื่อง ซึ่งโดยส่วนใหญ่แล้วจะเป็นเครือข่ายที่ใหญ่เกินไปสำหรับองค์กรทั่วไป ดังนั้นจึงทำให้หมายเลขไอพีบางส่วนไม่ถูกใช้งาน การทำซับเน็ตเป็นการแก้ปัญหานี้ได้ โดนการแบ่งเป็นเครือข่ายย่อยตามที่ได้กล่าวมาแล้ว เราท์เตอร์จะใช้เฉพาะส่วนที่เป็นหมายเลขเครือข่ายเท่านั้น สำหรับการจัดการเส้นทางหรือเราท์ติ้ง (Routing) สำหรับบริษัทใหญ่ๆ ที่ต้องการมีเครือข่าย ย่อยๆ หลายเครือข่ายจำเป็นที่จะต้องใช้เลขไอพีประเภท C ถ้าบริษัทไม่มีวิธีที่จะจัดการสำหรับการแบ่งเครือข่ายย่อย แต่มีวิธีใช้แบ่งเครือข่ายใหญ่ให้เป็นเครือข่ายย่อยโดยวิธีที่เรียกว่า “ซับเน็ต (Subnet)” การแบ่งเป็นเครือข่ายย่อยๆ หรือซับเน็ตนั้นจะถูกจัดการโดยเราท์เตอร์หรือเกตเวย์ของเครือข่ายนั้นๆ เท่านั้น โดยที่โฮสต์อื่นๆ ที่อยู่นอกเครือข่ายนี้จะมองเห็นเครือข่ายนี้เป็นเครือข่ายเดียวกัน

การแบ่งเครือข่ายที่ใหญ่ให้เป็นหลายเครือข่ายย่อย นั้นทำได้โดยการแบ่งเลขไอพีส่วนที่เป็นหมายเลขของโฮสต์มาเป็นหมายเลขของเครือข่าย ดังแสดงในรูปที่ 8 ส่วนซับเนตมาส์ค (Subnet Mask) คือตัวเลขที่บ่งชี้ว่าส่วนไหนของเลขไอพีเป็นหมายเลขเครือข่าย และส่วนไหนเป็นหมายเลขโฮสต์ ซับเน็ตมาส์คจะมีความยาวเท่ากับหมายเลขไอพีคือ 32 บิต ซึ่งเริ่มต้นด้วยแถวของเลข 1 เรียงกันและตามด้วยแถวของเลข 0 การคำนวณหมายเลขเครือข่ายย่อยจะทำได้โดยการแอนด์ (AND) ระหว่างเลขซับเน็ตมาส์คและหมายเลขไอพี โดยดีฟอลต์แล้วเน็ตมาส์คของแต่หมายเลขไอพีแต่ละคลาสแสดงในตาราง 2.3



รูปที่ 2.8 การทำซับเน็ต

ตาราง 2.3 ดีฟอลต์เน็ตมาส์คของไอพีแต่ละประเภท

คลาส	เน็ตมาส์ค	ด็อตเดซิมอล
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

ส่วนของหมายเลขโฮสต์ที่นำมาทำเป็นหมายเลขเครือข่ายนั้น จะเริ่มจากเลขส่วนที่มีความสำคัญมากก่อน ตัวอย่างเช่น เมื่อนำหมายเลขคลาส C มาทำซับเน็ตก็จะได้ดังแสดงในตาราง 2.4

ตาราง 2.4 ซับเน็ตมาส์คในรูปเลขฐานสองและฐานสิบของเลข 8 บิต

จำนวนบิต	ซับเน็ตมาส์ค	เลขฐานสิบ	จำนวนซับเน็ต	จำนวนโฮสต์
1	11111111 11111111 11111111 10000000	255.255.255.128	2	126
2	11111111 11111111 11111111 11000000	255.255.255.192	4	62
3	11111111 11111111 11111111 11100000	255.255.255.224	8	30
4	11111111 11111111 11111111 11110000	255.255.255.240	16	14
5	11111111 11111111 11111111 11111000	255.255.255.248	32	6
6	11111111 11111111 11111111 11111100	255.255.255.252	64	2
7	11111111 11111111 11111111 11111110	255.255.255.254	128	0

8	11111111 11111111 11111111 11111111	255.255.255.255	254	1

จำนวนโฮสต์ในซับเน็ตนั้นสามารถคำนวณได้โดยใช้สูตร $2^n - 2$ โดย n คือ จำนวนบิตของหมายเลขโฮสต์ ตัวอย่างเช่น ถ้าเน็ตมาส์คของคลาส C เป็น 255.255.255.128 จำนวนบิตของหมายเลขโฮสต์คือ 7 ดังนั้นจำนวนโฮสต์ทั้งหมดในแต่ละเครือข่ายคือ $2^7 - 2 = 126$ โฮสต์ ส่วนจำนวนซับเน็ตสามารถคำนวณได้โดย 2n โดย n คือ จำนวนบิตของซับเน็ต ถ้าใช้สูตรการคำนวณจำนวนโฮสต์ข้างต้น จะมีกรณียกเว้นอยู่ 2 กรณี คือ กรณีแรกถ้าจำนวนบิตของโฮสต์เป็น 0 บิต กรณีนี้จะถือว่าหมายเลขเครือข่ายกับหมายเลขโฮสต์จะใช้หมายเลขเดียวกัน ส่วนกรณีที่จำนวนบิตของโฮสต์เป็น 1 จะไม่นิยมใช้เนื่องจากจะไปซ้ำซ้อนกับบรอดคาสต์แอดเดรสของเครือข่าย

การทำซับเน็ตไอพีนั้นเป็นสิ่งที่ยากมากที่สุดของการทำความเข้าใจโปรโตคอล TCP/IP ต่อไปนี้จะเป็นการทำความเข้าใจเกี่ยวกับการแบ่งเครือข่ายขนาดใหญ่ให้เป็นหลายเครือข่ายย่อย ซึ่งจะเรียกว่า การทำซับเน็ต จุดประสงค์หลักของการแบ่งเครือข่ายใหญ่ให้เป็นหลายเครือข่ายย่อยนั้นก็ด้วยเหตุผลหลายอย่าง เช่น ประสิทธิภาพและการรักษาความปลอดภัย การทำซับเน็ตนั้นจะทำให้สามารถแบ่งเครือข่ายหนึ่งให้เป็นหลายฟิสิคอลเน็ตเวิร์คได้ คำว่าฟิสิคอลเน็ตเวิร์คในที่นี้จะหมายถึงเครือข่ายที่ไม่มีเราท์เตอร์ เราท์เตอร์นั้นจะใช้สำหรับเชื่อมฟิสิคอลเน็ตเวิร์คเหล่านี้เข้าด้วยกัน

มีเหตุผลหลายอย่างที่ทำอย่างที่ทำให้การทำซับเน็ตเป็นเรื่องจำเป็น เช่น หมายเลขไอพีคลาส A และคลาส B จะมีจำนวนโฮสต์มากเกินกว่าที่จะทำให้เป็นเครือข่ายเดียวได้ เนื่องจากปริมาณข้อความบรอดคาสต์มากเกินไป เพื่อให้การใช้หมายเลขไอพีอย่างคุ้มค่า จึงจำเป็นที่ต้องแบ่งหมายเลขไอพีเหล่านั้นให้เป็นเครือข่ายย่อยแล้วใช้เราท์เตอร์เชื่อมเครือข่ายย่อยๆ เหล่านั้นทำให้จำกัดบรอดคาสต์แพ็กเก็ตให้อยู่ในเฉพาะเครือข่ายย่อยเท่านั้น ตัวอย่างเช่น องค์กรหนึ่งอาจมีสำนักงานอยู่หลายที่ แต่ไม่สามารถที่จะขอให้อินเตอร์นิก (InterNIC) แจกหมายเลขไอพีทีละหลายๆ เครือข่ายได้ ดังนั้นองค์กรอาจต้องทำซับเน็ตกับหมายเลขไอพี

นอกจากที่กล่าวมาแล้ว เครือข่ายอาจจะถูกแบ่งให้เป็นเครือข่ายย่อยเพื่อให้ง่ายต่อการจัดการ เนื่องจากเครือข่ายขนาดใหญ่อาจจะมีข้อมูลไหลเวียนในเครือข่ายมาก ซึ่งถ้ามากจนเกินไปก็จะทำให้เกิดความคับคั่ง หรือข้อมูลติดซึ่งเหมือนกับรถติดนั่นเอง การใช้เราท์เตอร์จะช่วยลดปัญหาดังกล่าวนี้ได้

เพื่อให้เห็นภาพชัดเจนมากยิ่งขึ้นมาดูตัวอย่างการทำซับเน็ตดังนี้ สมมติว่าโฮสต์หนึ่งมีหมายเลขไอพีเป็น 172.20.33.24 ซึ่งเป็นหมายเลขในคลาส B และถ้ากำหนดซับเน็ตมาส์คเป็น 255.255.224.0 เมื่อมีการส่งข้อมูลเราท์เตอร์จะทำการแอนด์ (AND) ระหว่างเลขไอพีและเลขซับเนตมาส์คเพื่อคำนวณหาหมายเลขเครือข่าย ผลที่ได้คือ หมายเลขเครือข่ายจะได้เป็น 172.20.32.0 (หมายเหตุ เลขศูนย์ต่อท้ายหมายเลขเครือข่ายจะเขียนหรือไม่ก็ได้ แต่ส่วนใหญ่จะนิยมเขียนเพื่อให้ได้

หมายเลขทั้งสี่ตัว) ส่วนหมายเลขโฮสต์จะได้เป็น 0.0.1.24 ตามข้อกำหนดแล้วหมายเลขเครือข่ายที่เป็นเลขศูนย์หมดจะหมายถึง เครือข่ายที่โฮสต์นั้นอยู่ ส่วนหมายเลขเครือข่ายที่เป็นเลขหนึ่งหมดจะเป็นหมายเลขสำหรับการส่งข้อมูลแบบแพร่กระจายหรือบรอดคาสต์ (Broadcast) ในเครือข่ายนั้นๆ

หมายเลขไอพี 172.20.33.24 10101100 00010100 00100001 00011000

ซับเน็ตมาส์ค 255.255.224.0 11111111 11111111 11100000 00000000

หมายเลขเครือข่าย 172.20.32.0 10101100 00010100 00100000 00000000

หมายเลขโฮสต์ 0.0.1.24 00000000 00000000 00000001 00011000

จากตัวอย่างข้างบนหมายเลขโฮสต์ 3 บิตได้ถูกนำมาทำเป็นหมายเลขซับเน็ต ซึ่งจะทำให้แบ่งหมายเลขไอพีในคลาสนี้ออกได้เป็น 8 เครือข่ายย่อยหรือซับเน็ตดังนี้

172.20.0.0 10101100 00010100 00000000 00000000

172.20.32.0 10101100 00010100 00100000 00000000

172.20.64.0 10101100 00010100 01000000 00000000

172.20.96.0 10101100 00010100 01100000 00000000

172.20.128.0 10101100 00010100 10000000 00000000

172.20.160.0 10101100 00010100 10100000 00000000

172.20.192.0 10101100 00010100 11000000 00000000

172.20.224.0 10101100 00010100 11100000 00000000

ตามมาตรฐานแล้วในการทำซับเน็ตนั้นจะมีกฎอยู่ 2 ข้อคือ

1) เนื่องจากหมายเลขเครือขายที่เป็นเลขศูนย์ทั้งหมดจะอ้างถึง “เครือข่ายนี้” ดังนั้นหมายเลขส่วนที่เป็นซับเน็ตนั้นไม่สามารถเป็นเลขศูนย์หมด เนื่องจากปลายทางจะอ้างถึงเครือข่ายนี้
2) เนื่องจากหมายเลขเครือข่ายที่เป็นเลขหนึ่งทั้งหมด จะเป็นหมายเลขที่ใช้สำหรับการบรอดคาสต์ ดังนั้นหมายเลขส่วนที่เป็นซับเนตไม่สามารถเป็นเลขหนึ่งหมดได้

กฎนี้จะจำกัดการแบ่งซับเน็ต จากตัวอย่างข้างบนตามกฎแล้วหมายเลขซับเน็ตต่อไปนี้จะใช้ไม่ได้ เนื่องจากหมายเลขซับเน็ตเป็นเลขศูนย์หรือเลขหนึ่งทั้งหมด

172.20.0.0 10101100 00010100 00000000 00000000

172.20.224.0 10101100 00010100 11100000 00000000

อย่างไรก็ตาม ยังไม่มีเหตุผลที่ดีพอว่าทำไมถึงไม่สามารถกำหนดให้หมายเลขซับเน็ตเป็นศูนย์หรือหนึ่งทั้งหมดได้ แต่ส่วนใหญ่บริษัทที่ใช้โปรโตคอล TCP/IP เช่น บริษัทไมโครซอฟท์ จะอนุญาตให้ใช้หมายเลขซับเน็ตที่เป็นศูนย์และหนึ่งได้ อย่างไรก็ตามควรสอบถามบริษัทนั้นก่อนถ้าจะให้ซับเน็ตที่มีเลขเครือข่ายเป็นศูนย์หมด

ตาราง 2.5 และตาราง 2.6 แสดงหมายเลขเน็ตมาส์คที่ใช้ได้ จำนวนซับเน็ต และจำนวนโฮสต์ของหมายเลขไอพีคลาส B และคลาส C ตามลำดับ ส่วนการแบ่งซับเน็ตคลาส A ไม่ได้อธิบายในที่นี้ แต่การคำนวณก็คล้ายกันกับคลาส B และ C

ตาราง 2.5 ซับเน็ตของเครือข่ายคลาส B

จำนวนบิต	ซับเน็ตมาส์ค	จำนวนซับเน็ต	จำนวนโฮสต์
2	255.255.192.0	2	16,382
3	255.255.224.0	6	8,190
4	255.255.240.0	14	4,094
5	255.255.248.0	30	2,046
6	255.255.252.0	62	1,022
7	255.255.254.0	126	510
8	255.255.255.0	254	254
9	255.255.255.128	510	126
10	255.255.255.192	1,022	62
11	255.255.255.224	2,046	30
12	255.255.255.240	4,094	14
13	255.255.255.248	8,190	6
14	255.255.255.252	16,382	2

ตาราง 2.6 ซับเน็ตของคลาส C

จำนวนบิต	ซับเน็ตมาส์ค	จำนวนซับเน็ต	จำนวนโฮสต์
2	255.255.255.192	2	62
3	255.255.255.224	6	30
4	255.255.255.240	14	14
5	255.255.255.248	30	6
6	255.255.255.252	62	2

จากตารางข้างบน จำนวนซับเน็ตสามารถคำนวณโดยสมการ $2^x - 2$ โดย x คือจำนวนบิตที่เป็นซับเน็ตมาส์คที่นำมาจากส่วนที่หมายเลขโฮสต์ ตัวอย่างเช่น ถ้านำ 3 บิตมาทำเป็นซับเน็ต จำนวนซับเน็ตที่ได้คือ $2^3 - 2 = 8 - 2 = 6$ เป็นต้น ส่วนจำนวนโฮสต์ในแต่ละเครือข่ายก็คำนวณได้โดยใช้สูตรเดียวกัน โดย x ในที่นี้จะหมายถึงจำนวนบิตที่เป็นหมายเลขโฮสต์ สำหรับหมายเลขไอพีที่อยู่ในแต่ละเครือข่ายนั้น จะมีหมายเลขสองเลขหมายที่ไม่สามารถกำหนดให้กับโฮสต์ได้คือ หมายเลขที่ต่ำที่สุดและหมายเลขสูงสุด เพราะหมายเลขที่ต่ำสุดจะเป็นหมายเลขที่ใช้อ้างถึงเครือข่ายนั้น ส่วนหมายเลขเครือข่ายที่สูงที่สุดจะใช้เป็นหมายเลขสำหรับการส่งข้อมูล บรอดคาสต์ในเครือข่ายนั้น ตัวอย่างเช่น ในเครือข่าย 172.20.32.0 ที่มีซับเน็ตมาส์คเป็น 255.255.244.0 หมายเลขคือ 172.20.32.0 จะเป็นหมายเลขที่อ้างถึงเครือข่ายนี้ ส่วนหมายเลข 172.20.63.255 จะเป็นหมายเลขสำหรับบรอดคาสต์ในเครือข่ายนี้

2.3 IP Routing

ในตอนที่แล้วเราได้กล่าวถึงการจัดคอมพิวเตอร์ให้เป็นกลุ่ม หรือการแบ่งเครือข่ายให้เป็นเครือข่ายย่อย มันคงจะไม่เกิดประโยชน์อะไรถ้าคอมพิวเตอร์ที่อยู่ต่างเครือข่ายกันไม่สามารถสื่อสารกันได้ ในตอนนี้จะกล่าวถึงการส่งข้อมูลข้ามเครือข่าย หรือเรียกว่า "การเราท์ (Routing)" หนึ่งในจุดประสงค์หลักของโปรโตคอล TCP/IP คือการที่คอมพิวเตอร์ที่อยู่คนละเครือข่ายกันสามารถส่งแพ็กเก็ตถึงกันได้ ในตอนนี้จะกล่าวถึงกลไกของโปรโตคอล IP เกี่ยวกับการค้นหาเส้นทางที่จะส่งแพ็กเก็ตไปยังเครือข่ายอื่น โปรโตคอล IP จะจัดการเกี่ยวกับเส้นทางของแพ็กเก็ตโดยใช้ข้อมูลที่อยู่ในตารางเส้นทาง หรือเราท์ติ้งเทเบิ้ล (Routing Table) ซึ่งเราท์เตอร์แต่ละเครื่องจะต้องมี

2.3.1 หลักการทำงานของเราท์เตอร์

โฮสต์โดยทั่วไปแล้วจะไม่สามารถจัดการเรื่องเส้นทางได้ (Routing) เนื่องจากต้องใช้ทรัพยากรของเครื่องมาก แต่โฮสต์จะมีข้อมูลเพียงพอที่จะส่งแพ็กเก็ตข้อมูลไปยังเราท์เตอร์ ซึ่งแต่ละเครือข่ายจะมีเราท์เตอร์ หลักที่เรียกว่า "ดีฟอลต์เราท์เตอร์ (Default Routing)" หรือ "ดีฟอลต์เกตเวย์ (Default Gateway)" เมื่อโฮสต์ต้องการที่จะส่งแพ็กเก็ตไปยังโฮสต์ที่อยู่คนละเครือข่ายกัน (โฮสต์จะทราบได้จากหมายเลขเครือข่ายซึ่งจะคำนวณจากหมายเลขไอพีและซับเน็ตมาส์ค) โฮสต์นั้นก็จะส่งแพ็กเก็ตไปยังเราท์เตอร์แทน หลังจากนั้นก็จะเป็นหน้าที่ของเราท์เตอร์นั้นจะรับผิดชอบในการส่งต่อแพ็กเก็ตดังกล่าวไปยังโฮสต์ปลายทาง ดังนั้นการรับส่งแพ็กเก็ต ระหว่างเครือข่ายจึงเป็นหน้าที่ของเราท์เตอร์

การส่งแพ็กเก็ตระหว่างโฮสต์นั้นจะมีอยู่สองกรณีคือ กรณีที่โฮสต์ปลายทางอยู่ในเครือข่ายเดียวกันและ กรณีที่โฮสต์ปลายทางอยู่คนละเครือข่ายกัน การที่โฮสต์จะรู้ได้ว่าโฮสต์ปลายทางอยู่ในเครือข่ายเดียวกันหรือไม่นั้น โฮสต์จะตรวจสอบหมายเลขเครือข่ายของโฮสต์ปลายทาง ซึ่งจะ

คำนวณได้จากหมายเลขไอพีและซับเน็ตมาส์ค ถ้าโฮสต์ปลายทางอยู่ในเครือข่ายเดียวกัน แพ็กเก็ตก็จะถูกส่งตรงไปยังโฮสต์ปลายทางโดยที่ไม่ต้องผ่านเราท์เตอร์ ซึ่งโฮสต์จะใช้โปรโตคอล ARP (Address Resolution Protocol) ในการค้นหาแม็กแอดเดรส (MAC Address) ของโฮสต์ปลายทาง และส่งตรงไปยังโฮสต์นั้นทันที

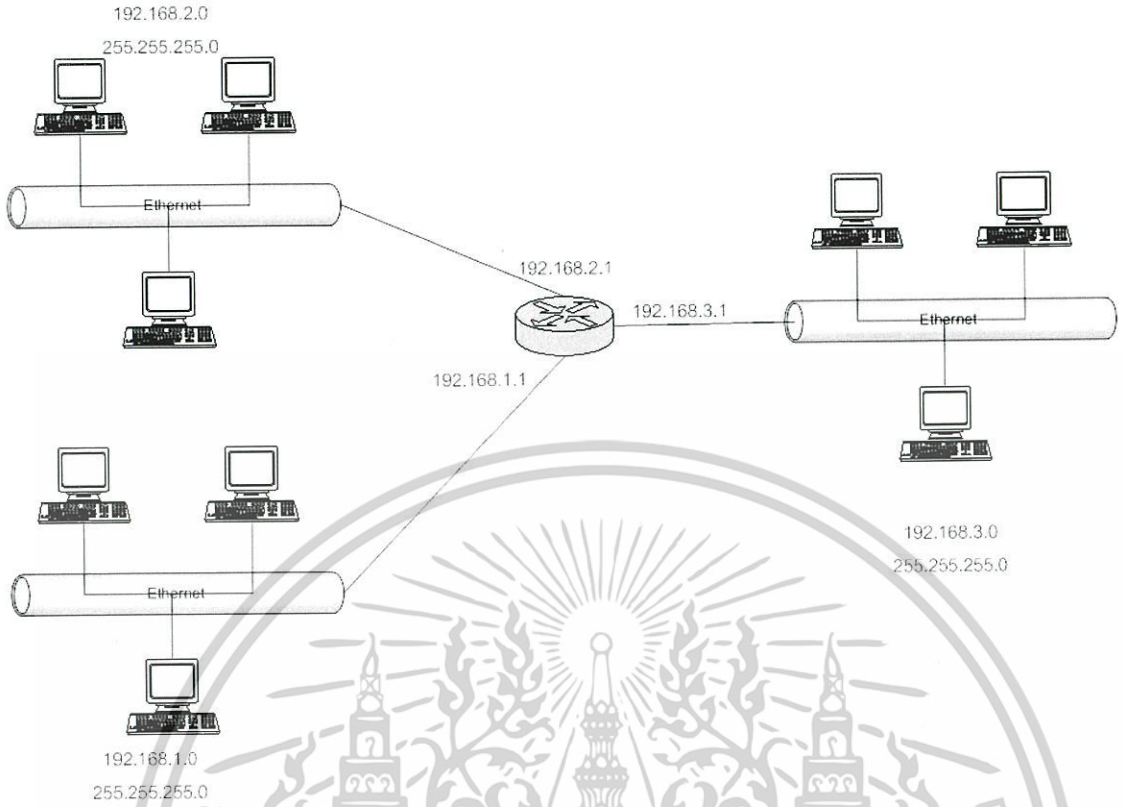
ตัวอย่างเช่น โฮสต์หนึ่งมีหมายเลขไอพีเป็น 172.20.1.22 และซับเนตมาส์คเป็น 255.255.0.0 ต้องการที่จะส่งแพ็กเก็ตไปยังโฮสต์ 172.20.16.32 โฮสต์ที่จะส่งแพ็กเก็ตดีจะทำการแอนด์ (AND) หมายเลขไอพีของตัวเองกับซับเน็ตมาส์ค ซึ่งจะได้หมายเลขเครือข่ายเป็น 172.20.0.0 (172.20.1.22 AND 255.255.0.0 = 172.20.0.0) และจะทำการแอนด์หมายเลขไอพีของโฮสต์ปลายทางกับซับเน็ตมาส์ค ซึ่งจะได้หมายเลขเครือข่ายของโฮสต์ปลายทางเป็น 172.20.0.0 (172.20.26.32 AND 255.255.0.0 = 172.20.0.0) เมื่อเปรียบเทียบหมายเลขเครือข่ายของตัวเองกับหมายเลขเครือข่ายของโฮสต์ปลายทางก็รู้ได้ว่าโฮสต์ปลายทางนั้นอยู่ในเครือข่ายเดียวกัน ดังนั้นโฮสต์ที่จะส่งก็จะใช้โปรโตคอล ARP เพื่อค้นหาหมายเลขแม็กแอดเดรสของโฮสต์ปลายทาง แล้วทำการส่งแพ็กเก็ตโดยตรงไปยังเครื่องปลายทาง โดยที่ไม่ต้องยุ่งเกี่ยวกับเราท์เตอร์เลย

ในกรณีที่สอง เมื่อโฮสต์ปลายทางอยู่คนละเครือข่ายกัน (ขั้นตอนการทดสอบก็เหมือนอย่างที่กล่าวข้างต้น) โฮสต์ก็จะใช้โปรโตคอล ARP ในการค้นหาหมายเลขแม็กแอดเดรสของดีฟอลต์เกตเวย์หรือเราท์เตอร์แล้วส่งแพ็กเก็ตไปยังเราท์เตอร์นั้น เราท์เตอร์จะทำหน้าที่ส่งต่อแพ็กเก็ตนั้นให้กับโฮสต์ จะเห็นได้ว่าการค้นหาเส้นทางของโฮสต์นั้นง่าย เนื่องจากโฮสต์ต้องการทราบว่าโฮสต์ปลายทางอยู่ในเครือข่ายเดียวกันหรือไม่เท่านั้น ถ้าไม่ใช่ก็ส่งแพ็กเก็ตต่อให้เราท์เตอร์จัดการต่อไป

2.3.2 Routing Table

หน้าที่ของเราท์เตอร์นั้นคือ การส่งผ่านแพ็กเก็ตระหว่างเครือข่าย ถ้าเปรียบเทียบระบบเครือข่ายกับระบบไปรษณีย์ เราท์เตอร์ก็เปรียบเสมือนที่ทำการไปรษณีย์นั่นเอง ดังนั้นเราท์เตอร์ต้องทราบข้อมูลเกี่ยวกับเครือข่ายต่างๆ เช่น เครือข่ายดังกล่าวสามารถส่งแพ็กเก็ตไปได้หรือไม่ ถ้าได้จะส่งไปทางใดได้บ้าง เป็นต้น ข้อมูลเกี่ยวกับเส้นทางนี้จะถูกเก็บไว้ในตารางที่เรียกว่า “เราท์ติ้งเทเบิ้ล (Routing Table)” ซึ่งตารางนี้จะมีรายการของหมายเลขไอพีของเราท์เตอร์และหมายเลขเครือข่ายที่เราท์เตอร์สามารถสื่อสารได้ เมื่อไรก็ตามที่เราท์เตอร์ต้องการส่งต่อแพ็กเก็ตมันจะใช้ข้อมูลในตารางนี้ในการตัดสินใจเลือกเส้นทาง

โดยทั่วไปแล้วในตารางเราท์ติ้งเทเบิ้ลจะประกอบด้วยหมายเลขเครือข่าย (Network ID) ซับเน็ตมาส์ค (Subnet Mask) หมายเลขไอพีของเกตเวย์ (Gateway Address) เน็ตเวิร์คการ์ด (Network Interface) และ เมตริก (Metric)



รูปที่ 2.9 การเชื่อมต่อเครือข่ายด้วยเราท์เตอร์

ตาราง 2.7 ตารางเราท์ติ้งเทเบิ้ล เป็นตารางของเราท์เตอร์ใน รูปที่ 2.9 ในตารางจะประกอบด้วยรายการซึ่งจะเป็นข้อมูลที่จะใช้สำหรับการส่งแพ็กเก็ตไปยังเครือข่ายปลายทาง เช่น เมื่อเราท์เตอร์ได้รับแพ็กเก็ตที่โฮสต์ปลายทางอยู่ในเครือข่าย 192.168.1.0 เราท์เตอร์ก็จะทำการส่งต่อแพ็กเก็ตไปยังเครือข่ายปลายทางโดยผ่านเน็ตเวิร์คการ์ด Eth01 ที่มีหมายเลขไอพีเป็น 192.168.1.1

ตาราง 2.7 ตารางเราท์ติ้งเทเบิ้ล

Network ID	Subnet Mask	Gateway	Interface	Metric
192.168.1.0	255.255.255.0	192.168.1.1	Eth01	1
192.168.2.0	255.255.255.0	192.168.2.1	Eth02	1
192.168.3.0	255.255.255.0	192.168.3.0	Eth03	1

ความหมายของแต่ละคอลัมน์มีดังนี้

1) หมายเลขเครือข่าย (Network ID) ในคอลัมน์นี้จะเป็นหมายเลขไอพีหรือหมายเลขเครือข่ายของโฮสต์ปลายทาง
2) ซับเน็ตมาส์ค (Subnet Mask) คือ หมายเลขที่เราท์เตอร์จะใช้แอนด์ (AND) กับหมายเลขไอพีของโฮสต์ปลายทางเพื่อคำนวณหาหมายเลขเครือข่าย
3) เกตเวย์ (Gateway) คือหมายเลขไอพีของเราท์เตอร์หรือเกตเวย์ที่สามารถส่งแพ็กเก็ตข้อมูลถึงเครือข่ายปลายทางได้
4) อินเตอร์เฟส (Interface) คือเน็ตเวิร์คอินเตอร์เฟสหรือเน็ตเวิร์คการ์ดของเราท์เตอร์ที่สามารถส่งข้อมูลถึงเกตเวย์ดังกล่าวได้
5) เมทริกซ์ (Metric) เป็นตัวเลขที่เป็นหน่วยวัดเกี่ยวกับความยากง่ายในการส่งแพ็กเก็ตไปยังเครือข่ายนั้น ส่วนใหญ่จะหมายถึงจำนวนฮอบ (Hop) หรือเราท์เตอร์ที่ต้องส่งแพ็กเก็ตผ่าน ก่อนที่จะถึงเครือข่ายปลายทาง

2.3.3 Routing Protocol

เราท์เตอร์จะใช้ข้อมูลที่อยู่ในตารางเราท์ติ้งเทเบิ้ลสำหรับการส่งแพ็กเก็ตระหว่างเครือข่าย ส่วนเส้นทางที่จะถูกเลือกนั้นจะขึ้นอยู่กับอัลกอริธึม (Algorithm) ที่ใช้หรือเรียกว่า เราท์ติ้งโปรโตคอล “(Routing Protocol)” ซึ่งจะแบ่งออกเป็น 2 ประเภทคือ

1) Static IP Routing
2) Dynamic IP Routing

2.3.3.1 Static IP Routing

สำหรับการจัดเส้นทางแบบนี้รายการในตารางเราท์ติ้งเทเบิ้ลจะถูกป้อนโดยผู้ดูแลระบบ ซึ่งข้อมูลในรายการนี้จะไม่มีการเปลี่ยนแปลงหลังจากนั้น การคอนฟิกตารางนั้นค่อนข้างจะง่าย แต่ผู้ติดตั้งระบบนั้นจะต้องป้อนข้อมูลทุกๆ ฟิลด์ในตารางเอง ซึ่งเป็นเหมือนกับการบอกเราท์เตอร์ให้ทราบว่าเครือข่ายไหนที่สามารถติดต่อได้ ความถูกต้องของข้อมูลในตารางเราท์ติ้งเทเบิ้ลจะขึ้นอยู่กับความรับผิดชอบของผู้ดูแลระบบนั้นๆ โดยทั่วไปแล้ว สำหรับเครือข่ายเล็กๆ จะใช้การเราท์แบบสแตติกนี้ แต่เมื่อเครือข่ายใหญ่ขึ้นก็จะใช้โปรโตคอลแบบไดนามิกซึ่งง่ายต่อการจัดการมากกว่า

2.3.3.2 Dynamic IP Routing

การจัดเส้นทางแบบไดนามิกนี้จะใช้โปรโตคอลในการสร้างตารางเราท์ติ้งเทเบิ้ลแทนการป้อนข้อมูลเองโดยคน ซึ่งวิธีการสร้างนั้นจะขึ้นอยู่กับโปรโตคอล เช่น โหลดของช่องสัญญาณแบนด์วิธของลิงค์ เป็นต้น ข้อได้เปรียบของการใช้โปรโตคอลประเภทนี้คือ รายการในตารางจะถูกอัพเดตโดยอัตโนมัติ ทำให้ผู้ดูแลระบบไม่ต้องกังวลว่ารายการในตารางจะผิดพลาด ส่วนข้อเสียคือ

ปริมาณการไหลเวียนของแพ็กเก็ตในเครือข่ายจะเพิ่มขึ้น โปรโตคอลแบบไดนามิกนี้ยังแบ่งออกเป็น 2 ประเภทคือ

1) Distance-Vector Routing Protocol

โปรโตคอลแบบดิสแทนส์เวคเตอร์จะเลือกเส้นทางที่ดีที่สุดโดยใช้เมตริก (Metric) เป็นเกณฑ์ โดยเมตริกนี้จะเป็นหน่วยที่วัดประสิทธิภาพของลิงค์ไปยังเครือข่ายนั้น และจะขึ้นอยู่กับโปรโตคอลที่ใช้ ซึ่งโดยส่วนใหญ่จะใช้จำนวนฮอบ (Hop) เป็นหลัก เราท์เตอร์ที่ใช้โปรโตคอลนี้จะรักษาตารางเราท์ติ้งเทเบิ้ล โดยรายการในตารางจะขึ้นอยู่กับสถานะของเครือข่ายนั้น ข้อเสียของโปรโตคอลนี้คือ เราท์เตอร์จะต้องทำการแลกเปลี่ยนข้อมูลซึ่งกันและกันเพื่ออัพเดตรายการในตาราง หรือเพื่อให้ตารางเราท์ติ้งเทเบิ้ลของแต่ละเราท์เตอร์ทันสมัยอยู่ตลอดเวลา เราท์เตอร์แต่ละตัวต้องบรอดคาสต์ในช่วงเวลาที่กำหนดเป็นประจำ ดังนั้นจึงทำให้จำนวนแพ็กเก็ตที่ไหลเวียนในเครือข่ายเพิ่มขึ้น โปรโตคอลที่จัดอยู่ในประเภทนี้ เช่น RIP (Routing Information Protocol), IGRP (Interior Gateway Routing Protocol) เป็นต้น

2) Link-State Routing Protocol

โปรโตคอลแบบลิงค์สเตท (Link State Routing Protocol) จะสร้างเส้นทางข้อมูลเหมือนกับต้นไม้ (Tree) โดยรากของต้นไม้นี้คือ เราท์เตอร์ตัวมันเอง โดยเราท์เตอร์แต่ละตัวจะบรอดคาสต์ข้อมูลที่เกี่ยวกับเครือข่ายที่เชื่อมต่อตรงกับเราท์เตอร์นั้นและเมตริก เราท์เตอร์จะทำการบรอดคาสต์เฉพาะตอนที่มีการเปลี่ยนแปลงเท่านั้น ทำให้ลดจำนวนแพ็กเก็ตในเครือข่ายลงได้

เมื่อเราท์เตอร์ค้นพบว่ามีการเปลี่ยนแปลงเกี่ยวกับเครือข่ายที่เชื่อมต่อตรงกับเราท์เตอร์นั้น มันก็จะทำการบรอดคาสต์ข้อมูลดังกล่าวไปยังทุกๆ เราท์เตอร์ ซึ่งขบวนการนี้จะเรียกว่า “การฟลัด (Flooding)” การฟลัดนี้จะอัพเดตฐานข้อมูลทุกๆ เราท์เตอร์เฉพาะรายการที่มีการเปลี่ยนแปลงเท่านั้น โดยทั่วไปแล้วแพ็กเก็ตเหล่านี้จะมีขนาดเล็กและมีการส่งไม่บ่อยมากนัก โปรโตคอลที่จัดอยู่ในประเภทนี้ เช่น OSPF (Open Shortest Path First)

2.3.3.3 Routing Information Protocol (RIP)

โปรโตคอล RIP (Routing Information Protocol) เป็นโปรโตคอลที่เราท์เตอร์ใช้สำหรับการแลกเปลี่ยนข้อมูลเกี่ยวกับเครือข่าย เช่น หมายเลขเครือข่าย ข้อมูลเกี่ยวกับว่ามีเครือข่ายใดที่เชื่อมต่อกับเราท์เตอร์นั้นโดยตรงเป็นต้น RIP เป็นโปรโตคอลที่นิยมมากเนื่องจากเป็นโปรโตคอลที่ใช้งานง่าย และเป็นโปรโตคอลแบบดิสแตนส์เวคเตอร์ ซึ่งเราท์เตอร์แต่ละเครื่องจะใช้วิธีส่งข้อมูลเหล่านี้แบบบรอดคาสต์ทุกๆ 30 วินาที เพื่ออัพเดตตารางเราท์ติ้งเทเบิ้ลกับเราท์เตอร์ที่อยู่ติดกัน การบรอดคาสต์นี้จะทำให้ตารางเราท์ติ้งเทเบิ้ลของแต่ละเราท์เตอร์ทันสมัยตลอดเวลา แต่ทำให้จำนวนแพ็กเก็ตที่ไหลเวียนในเครือข่ายเพิ่มมากขึ้น

การใช้โปรโตคอล RIP ซึ่งเป็นโปรโตคอลแบบดิสแตนส์เวคเตอร์มีข้อจำกัดอยู่หลายอย่าง เช่น การอัพเดตฐานข้อมูลของเราท์เตอร์ในเครือข่ายอาจใช้เวลานานพอสมควรสำหรับเครือข่าย

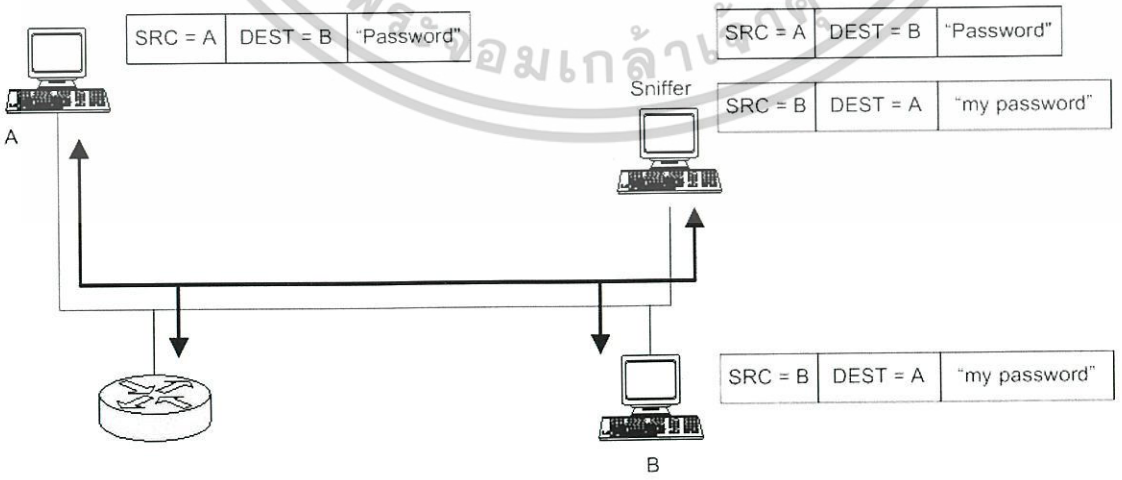
ขนาดใหญ่ และปัญหาอีกอย่างคือ การส่งแพ็กเก็ตเพื่ออัพเดตตารางเราท์ติ้งเทเบิ้ลนั้น อาจมีบางแพ็กเก็ตที่ไหลเวียนอยู่ในเครือข่ายโดยไม่มีที่สิ้นสุด เพื่อป้องกันปัญหานี้เราท์เตอร์แต่ละเครื่องที่ได้รับแพ็กเก็ตนี้จะทำการลด TTL (time-to-live) ของแพ็กเก็ต ซึ่งเมื่อค่าของ TTL เป็นศูนย์ แพ็กเก็ตนั้นก็จะถูกละทิ้ง ดังนั้น RIP จะกำหนดขนาดของเครือข่ายโดยแพ็กเก็ตของ RIP จะอนุญาตให้ส่งผ่านเราท์เตอร์ได้ไม่เกิน 15 เครื่อง ซึ่งถ้าแพ็กเก็ตส่งผ่านเราท์เตอร์เกิน 15 เครื่องก็จะถือว่าแพ็กเก็ตนี้ถูกส่งวนครบแล้วและจะถูกกำจัดออกจากเครือข่าย

2.4 การโจมตีเครือข่าย

เครือข่ายเป็นเทคโนโลยีที่น่าอัศจรรย์ แต่ยังคงมีความเสี่ยงอยู่มากถ้าไม่มีการควบคุมหรือป้องกันที่ดี การโจมตีหรือการบุกรุกเครือข่ายหมายถึงความพยายามที่จะเข้าใช้ระบบ (Access Attack) การแก้ไขข้อมูลหรือระบบ (Modification Attack) การทำให้ระบบไม่สามารถใช้งานได้ (Deny of Service Attack) และการทำให้ข้อมูลเป็นเท็จ (Repudiation Attack) ซึ่งจะทำโดยผู้ประสงค์ร้าย ผู้ที่ไม่มีสิทธิ์ หรืออาจเกิดจากความไม่ตั้งใจของผู้ใช้เอง ต่อไปนี้เป็นรูปแบบต่างๆ ที่ผู้ไม่ประสงค์ดีพยายามที่จะบุกรุกเครือข่ายเพื่อลักลอบข้อมูลที่สำคัญหรือเข้าใช้ระบบโดยไม่ได้รับอนุญาต

2.4.1 แพ็กเก็ตสนิฟเฟอร์

ข้อมูลที่คอมพิวเตอร์ส่งผ่านเครือข่ายนั้นจะถูกแบ่งย่อยเป็นก้อนเล็กๆ ซึ่งจะเรียกว่า "แพ็กเก็ต (Packet)" โปรแกรมประยุกต์หลายชนิดจะส่งข้อมูลโดยที่ไม่ได้เข้ารหัส (Encryption) หรือในรูปแบบเคลียร์เท็กซ์ (Clear Text) ดังนั้นข้อมูลอาจถูกคัดลอกและโพรเซสโดยโปรแกรมประยุกต์อื่นก็ได้



รูป 2.10 Packet Sniffing

เน็ตเวิร์คโปรโตคอลเป็นตัวกำหนดหมายเลขของแต่ละแพ็กเก็ต ซึ่งเป็นสิ่งที่คอมพิวเตอร์ใช้สำหรับบ่งบอกว่าแพ็กเก็ตนั้นส่งไปไหนหรือมาจากไหนเนื่องจากโปรโตคอลส่วนใหญ่ เช่น TCP/IP เป็นโปรโตคอลมาตรฐานและเป็นที่รู้จักกันโดยทั่วไป ทำให้บางกลุ่มพัฒนาโปรแกรมประยุกต์ที่สามารถตรวจจับแพ็กเก็ตที่วิ่งบนเครือข่ายได้ ซึ่งเทคนิคนี้เรียกว่า "แพ็กเก็ตสนิฟเฟอร์ (Packet Sniffer)" สิ่งที่น่ากลัวจริงๆ ในปัจจุบันนี้คือ โปรแกรมแพ็กเก็ตสนิฟเฟอร์มีให้ดาวน์โหลดบนอินเตอร์เน็ตมากมาย และผู้ที่ใช้งานไม่จำเป็นต้องมีความรู้เกี่ยวกับคอมพิวเตอร์มากก็สามารถใช้ซอฟต์แวร์เหล่านี้ได้ แพ็กเก็ตสนิฟเฟอร์เป็นโปรแกรมใช้เน็ตเวิร์คการ์ดในโหมดโพรมิสเชียส์ (Promiscuous mode) ซึ่งในโหมดนี้เน็ตเวิร์คการ์ดจะรับทุกๆ แพ็กเก็ตที่วิ่งบนสายสัญญาณและส่งต่อไปให้ยังโปรแกรมประยุกต์เพื่อโพรเซสต่อไป

เนื่องจากโปรแกรมประยุกต์ส่วนใหญ่จะส่งแพ็กเก็ตแบบเคลียเท็กซ์ แพ็กเก็ตสนิฟเฟอร์สามารถตรวจจับข้อมูลที่อาจเป็นประโยชน์ได้ เช่น ชื่อผู้ใช้และรหัสผ่าน เป็นต้น ถ้าหากมีการใช้ฐานข้อมูลผ่านเครือข่ายแพ็กเก็ต สนิฟเฟอร์อาจโจมตีโดยใช้ชื่อผู้ใช้และรหัสผ่านที่ตรวจจับได้ก็ได้ สิ่งที่น่ากลัวมากกว่าคือ ผู้ใช้มักจะใช้ชื่อผู้ใช้ และรหัสผ่านเดิมกับทุกๆ โปรแกรมประยุกต์ ทำให้ผู้บุกรุกสามารถโจมตีโปรแกรมประยุกต์ต่างๆ ได้อย่างง่ายดายและอาจทำให้เครือข่ายเกิดความเสียหายมากกว่าที่คิด

2.4.2 ไอพีสปูฟิง

ไอพีสปูฟิง (IP Spoofing) หมายถึง การที่ผู้บุกรุกอยู่นอกเครือข่ายแล้วแกล้งทำเป็นว่าเป็นว่าเป็นคอมพิวเตอร์ที่เชื่อถือได้ (Trusted) โดยอาจจะใช้ไอพีแอดเดรสเหมือนกับที่ใช้ในเครือข่าย หรืออาจจะใช้ไอพีแอดเดรสข้างนอกที่เครือข่ายเชื่อว่าเป็นคอมพิวเตอร์ที่เชื่อถือได้หรืออนุญาตให้เข้าใช้ทรัพยากรในเครือข่ายได้ โดยปกติแล้วการโจมตีแบบไอพีสปูฟิงเป็นการเปลี่ยนแปลง หรือเพิ่มข้อมูลเข้าไปในแพ็กเก็ตที่รับส่งระหว่างไคลเอนท์และเซิร์ฟเวอร์ หรือคอมพิวเตอร์สื่อสารกันในเครือข่าย การที่จะทำอย่างนี้ได้ผู้บุกรุกจะต้องปรับเราท์ติ้งเทเบิ้ลของเราท์เตอร์เพื่อให้ส่งต่อแพ็กเก็ตไปที่เครื่องของผู้บุกรุก หรืออีกวิธีหนึ่งคือการที่ผู้บุกรุกสามารถแก้ไขให้แอพลิเคชันส่งข้อมูลที่เป็นประโยชน์ต่อการเข้าถึงโปรแกรมประยุกต์นั้นผ่านทางอีเมล หลังจากนั้นผู้บุกรุกก็สามารถเข้าใช้โปรแกรมประยุกต์ได้โดยใช้ข้อมูลดังกล่าว

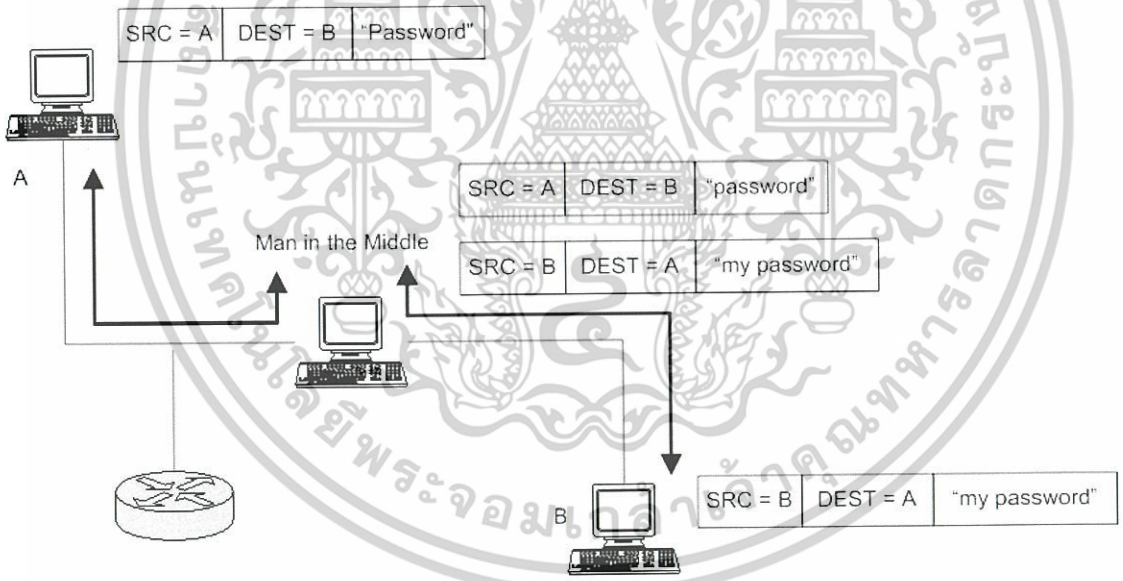
อย่างไรก็ตาม ถ้าผู้บุกรุกสามารถปรับเปลี่ยนเราท์ติ้งเทเบิ้ลเพื่อให้ส่งข้อมูลไปยังเครื่องปลอมได้ผู้บุกรุกสามารถรับส่งข้อมูลกับโปรแกรมประยุกต์นั้นเสมือนเป็นหนึ่งผู้ใช้ทั่วๆ ไปได้ ไอพีสปูฟิงไม่จำเป็นจะต้องเป็นคอมพิวเตอร์ที่อยู่นอกเครือข่ายเท่านั้น แต่อาจจะเป็นผู้ใช้ที่อยู่ข้างในที่ไม่มีสิทธิ์ก็ได้ ซึ่งอย่างที่ทราบกันดีว่า การโจมตีเครือข่ายนั้น 90% จะเป็นการโจมตีที่เกิดจากภายในเครือข่ายเอง

2.4.3 การโจมตีรหัสผ่าน

การโจมตีรหัสผ่าน (Password Attack) หมายถึง การโจมตีที่ผู้บุกรุกพยายามเดารหัสผ่านของผู้ใช้คนใดคนหนึ่ง ซึ่งวิธีการเดานั้นก็มีหลายวิธี เช่น บรู๊ทฟอร์ซ (Brute-Force), โทรจันฮอร์ส (Trojan Horse), ไอพีสปูฟิง, แพ็กเก็ตสนิฟเฟอร์ เป็นต้น การเดาแบบบรู๊ทฟอร์ซ หมายถึง การลองผิดลองถูกรหัสผ่านเรื่อยๆ จนกว่าจะถูก บ่อยครั้งที่การโจมตีแบบบรู๊ทฟอร์ซใช้การพยายามล็อกอินเข้าใช้รีซอร์สของเครือข่าย โดยถ้าทำสำเร็จผู้บุกรุกก็จะมีสิทธิ์เหมือนกับเจ้าของแอ็คเคาท์นั้นๆ ถ้าหากแอ็คเคาท์นี้มีสิทธิ์เพียงพอผู้บุกรุกอาจสร้างแอ็คเคาท์ใหม่เพื่อเป็นประตูหลัง (Back Door) และใช้สำหรับการเข้าระบบในอนาคตได้

2.4.4 การโจมตีแบบ Man-in-the-Middle

การโจมตีแบบ Man-in-the-Middle นั้นผู้โจมตีต้องสามารถเข้าถึงแพ็กเก็ตที่ส่งระหว่างเครือข่ายได้ เช่น ผู้โจมตีอาจอยู่ที่ ISP ซึ่งสามารถตรวจจับแพ็กเก็ตที่สามารถรับส่งระหว่างเครือข่ายภายในและเครือข่ายอื่นๆ โดยผ่าน ISP การโจมตีนี้จะใช้แพ็กเก็ตสนิฟเฟอร์เป็นเครื่องมือเพื่อขโมยข้อมูล หรือใช้เซสชั่นเพื่อแอ็กเซสเครือข่ายภายใน หรือวิเคราะห์การจราจรของเครือข่ายหรือผู้ใช้



รูปที่ 2.11 Man-in-the-Middle

2.4.5 การโจมตีแบบ DOS

การโจมตีแบบดีไนล์ออฟเซอร์วิส หรือ DOS (Denial-of-Service) หมายถึง การโจมตีเซิร์ฟเวอร์โดยการทำให้เซิร์ฟเวอร์นั้นไม่สามารถให้บริการได้ ซึ่งโดยปกติจะทำโดยการใช้รีซอร์สของเซิร์ฟเวอร์จนหมด หรือถึงขีดจำกัดของเซิร์ฟเวอร์ ตัวอย่างเช่น เว็บเซิร์ฟเวอร์ และเอฟทีพี

เซิร์ฟเวอร์ การโจมตีจะทำได้โดยการเปิดการเชื่อมต่อ (Connection) กับเซิร์ฟเวอร์จนถึงขีดจำกัดของเซิร์ฟเวอร์ ทำให้ผู้ใช้คนอื่นๆ ไม่สามารถเข้ามาใช้บริการได้ การโจมตีแบบนี้อาจใช้โปรโตคอลที่ใช้บนอินเตอร์เน็ตทั่วๆ ไป เช่น TCP (Transmission Control Protocol) หรือ ICMP (Internet Control Message Protocol) การโจมตีแบบดีไนล์ออฟเซอร์วิส เป็นการโจมตีจุดอ่อนของระบบหรือเซิร์ฟเวอร์มากกว่าการโจมตีจุดบกพร่อง (Bug) หรือช่องโหว่ของระบบการรักษาความปลอดภัย อย่างไรก็ตาม การโจมตีอาจทำให้ประสิทธิภาพของเครือข่ายลดลงโดยการส่งแพ็กเก็ตจำนวนมากเข้าไปในเครือข่าย ซึ่งแพ็กเก็ตอาจเป็นข้อมูลที่เป็นขยะ

2.4.6 โทรจันฮอร์ส เวิร์ม และไวรัส

คำว่า "โทรจันฮอร์ส (Trojan Horse)" นี้เป็นคำที่มาจากสงครามโทรจัน ระหว่างทรอย (Troy) และกรีก (Greek) ซึ่งเปรียบถึงม้าโครงไม้ที่ชาวกรีกสร้างทิ้งไว้แล้วซ่อนทหารไว้ข้างในแล้วถอนทัพกลับ พอชาวโทรจันออกมาดูเห็นโครงม้าไม้ทิ้งไว้ และคิดว่าเป็นของขวัญที่กรีซทิ้งไว้ให้จึงนำกลับเข้าเมืองไปด้วย พอตกดึกทหารกรีกที่ซ่อนอยู่ในม้าโครงไม้นี้ก็ออกมาและเปิดประตูให้กับทหารกรีกเข้าไปทำลายเมืองทรอย สำหรับในความหมายคอมพิวเตอร์แล้ว โทรจันฮอร์สหมายถึงโปรแกรมที่ทำลายระบบคอมพิวเตอร์โดยแฝงมากับโปรแกรมอื่นๆ เช่น เกม สกรีนเซฟเวอร์ เป็นต้น ซึ่งผู้ใช้อาจจะดาวน์โหลดโปรแกรมต่างๆ เหล่านี้มา แต่เมื่อติดตั้งแล้วรันโปรแกรมโทรจันฮอร์สที่แฝงมาด้วยก็จะทำลายระบบคอมพิวเตอร์ เช่น ลบไฟล์ต่างๆ หรืออาจสร้างแบ็คดอร์ให้กับโปรแกรมอื่นเข้ามาทำลายระบบก็ได้

เวิร์ม (Worm) หมายถึง โปรแกรมที่เป็นอันตรายต่อระบบคอมพิวเตอร์ โดยจะแพร่กระจายตัวเองไปยังคอมพิวเตอร์เครื่องอื่นๆ ที่อยู่ในเครือข่าย เวิร์มจะใช้ประโยชน์จากโปรแกรมประยุกต์ที่รับส่งไฟล์โดยอัตโนมัติ ส่วนไวรัส (Virus) หมายถึงโปรแกรมที่ทำลายระบบคอมพิวเตอร์ โดยจะแพร่กระจายไปยังโปรแกรมอื่นๆ ที่อยู่ในเครื่องเดียวกัน ไวรัสสามารถทำลายเครื่องได้ตั้งแต่ลบไฟล์ทั้งหมดที่อยู่ในฮาร์ดดิสก์ไปจนถึงแค่เป็นโปรแกรมที่สร้างความรำคาญให้กับผู้ใช้ในเครือข่าย เช่น แค่เปิดวินโดวส์แล้วแสดงข้อความบางอย่างไวรัสจริงๆ ไม่สามารถที่จะแพร่กระจายไปยังเครื่องอื่นๆ ด้วยตัวเองได้ แต่การแพร่กระจายไปยังเครื่องอื่นต้องอาศัยโปรแกรมอื่นหรือมนุษย์ เช่น การแชร์ไฟล์โดยใช้แผ่นดิสก์ เป็นต้น

จากคำจำกัดความข้างต้น โทรจันฮอร์ส เวิร์ม และไวรัส จะมีความหมายคล้ายๆ กัน ซึ่งบางคนอาจใช้คำทั้งสามนี้แทนกันก็ได้ แต่จริงๆ แล้วทั้งสามคำมีความหมายต่างกัน อย่างไรก็ตาม โปรแกรมทำลายคอมพิวเตอร์ในปัจจุบันอาจเป็นได้ทั้งสามชนิดก็ได้

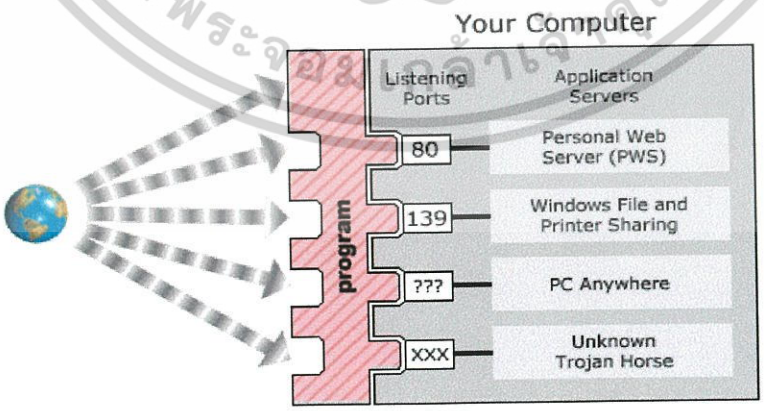
บทที่ 3
การออกแบบและพัฒนาโปรแกรม

3.1 ขอบเขตการทำงาน

การทำงานของโปรแกรมที่จัดทำ จะทำงานอยู่ในระบบคอมพิวเตอร์ที่มีระบบปฏิบัติการเป็นไมโครซอฟท์ วินโดวส์ ตั้งแต่รุ่น 98 ขึ้นไป โดยโปรแกรมจะทำงานเสมือนส่วนหนึ่งของระบบปฏิบัติการ (Background Process) ซึ่งจะคอยตรวจสอบการเรียกขอใช้ การรับส่งข้อมูลผ่านระบบเครือข่ายของโปรแกรมประยุกต์ต่างๆ ที่เริ่มต้นการทำงาน หรือ ทำงานอยู่บนระบบคอมพิวเตอร์ อยู่ก่อนแล้ว ซึ่งโปรแกรมจะทำการตัดสินใจในการเปิดให้โปรแกรมประยุกต์นั้นๆ สามารถใช้การรับส่งข้อมูลกับระบบเครือข่ายได้ หรือไม่ โดยดูจากการอนุญาตจากผู้ใช้ในขณะนั้น หรือ ตามที่ผู้ใช้ตั้งค่าไว้ในโปรแกรม ซึ่งถ้าผู้ใช้ไม่อนุญาตให้โปรแกรมดังกล่าวทำการรับส่งข้อมูลไปยังระบบเครือข่ายได้ โปรแกรมก็จะทำการ ปิดช่องทางในการรับส่งข้อมูลของโปรแกรมประยุกต์ นั้นๆ ไม่ให้สามารถทำการรับส่งข้อมูลกับระบบเครือข่ายได้ ทั้งนี้ หากระบบคอมพิวเตอร์ในเครือข่ายทำการเรียกขอข้อมูลจากระบบคอมพิวเตอร์ที่ทำการติดตั้งโปรแกรม โปรแกรมจะทำการปฏิเสธการร้องขอข้อมูลนั้นๆ โดยอัตโนมัติหรือ เปิดให้สามารถเรียกขอข้อมูลได้เมื่อผู้ใช้ได้ทำการตั้งค่าการอนุญาตรับส่งข้อมูลให้กับระบบคอมพิวเตอร์ที่อยู่ในเครือข่ายนั้นได้ ซึ่งในการตั้งค่าต่างๆ ของโปรแกรมผู้ใช้สามารถตั้งค่าได้เองตามรูปแบบการใช้งานของผู้ใช้ หรือเรียกใช้ค่ามาตรฐานที่ผู้พัฒนาได้ทำการตั้งค่าไว้ให้ ได้

3.2 การออกแบบสถาปัตยกรรมที่ใช้ในโปรแกรม

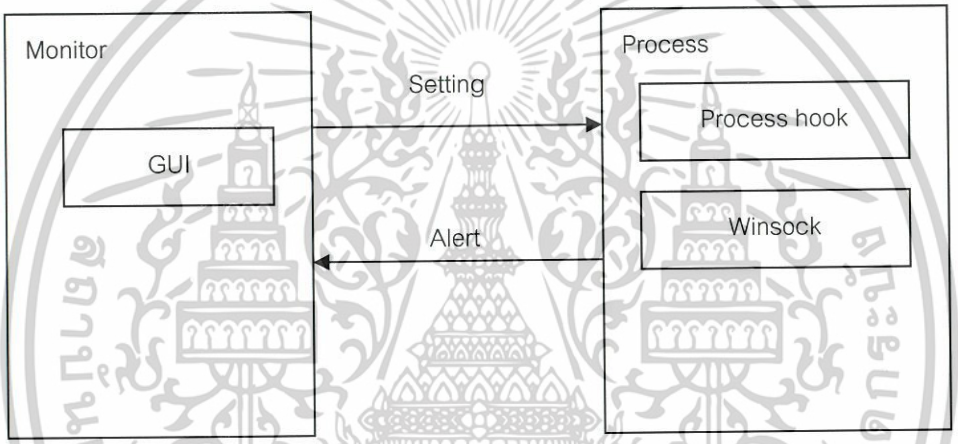
3.2.1 โครงสร้างสถาปัตยกรรมของโปรแกรม



จากรูปจะเป็นการแสดงให้เห็นถึงการทำงานของระบบโดยรวม ซึ่ง ระบบจะตรวจสอบการรับส่งข้อมูลผ่านทาง การเชื่อมต่อ (Port) ต่างๆ โดยจะเริ่มจากการร้องขอใช้ ไลบรารี (Library) ที่เป็น Win32 API (Windows Application Program Interface)คือ Winsock Control แล้วนำชื่อโปรแกรมประยุกต์ที่ทำการร้องขอนั้นไปตรวจสอบกับ ฐานข้อมูลโปรแกรมใน ระบบ เพื่อดูว่า ผู้ใช้อนุญาตให้โปรแกรมประยุกต์นั้นสามารถทำการเชื่อมต่อกับระบบเครือข่ายได้หรือไม่ ถ้าผู้ใช้ตั้งค่าไว้ไม่ให้ทำการเชื่อมต่อระบบจะทำการปิดกั้นการเชื่อมต่อดังกล่าว โดยอัตโนมัติ

3.2.2 โครงสร้างสถาปัตยกรรมภายในโปรแกรม

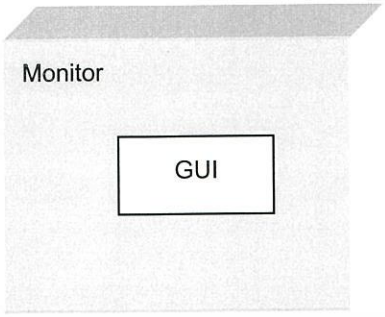
ภายในโปรแกรม จะแบ่งออกเป็นองค์ประกอบ 2 หลักคือ การแสดงผล และ การตรวจสอบการรับส่งข้อมูลผ่านระบบเครือข่าย



รูปที่ 3.2 แสดงการทำงานภายในระบบ

3.2.2.1 องค์ประกอบโครงสร้างสถาปัตยกรรมการแสดงผล

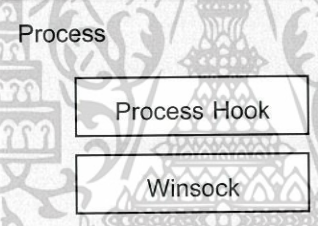
ในการของแสดงผลของโปรแกรมจะเป็นส่วนที่บอกรายละเอียดการทำงานของระบบให้กับผู้ใช้ ซึ่งจะมีลักษณะเป็น GUI (Graphic User Interface) โดยจะติดต่อกับส่วนตรวจสอบข้อมูลด้วยวิธี message passing ระหว่าง ระบบทั้ง 2



รูปที่ 3.3 แสดงองค์ประกอบโครงสร้างสถาปัตยกรรมส่วนแสดงผล

3.2.2.2 องค์ประกอบโครงสร้างสถาปัตยกรรมการตรวจสอบข้อมูล

ในส่วนของระบบตรวจสอบข้อมูล จะประกอบไปด้วย Win32 API (Window 32 Application Program Interface) ที่เกี่ยวข้องกับการรับส่งข้อมูลไปยังระบบเครือข่าย

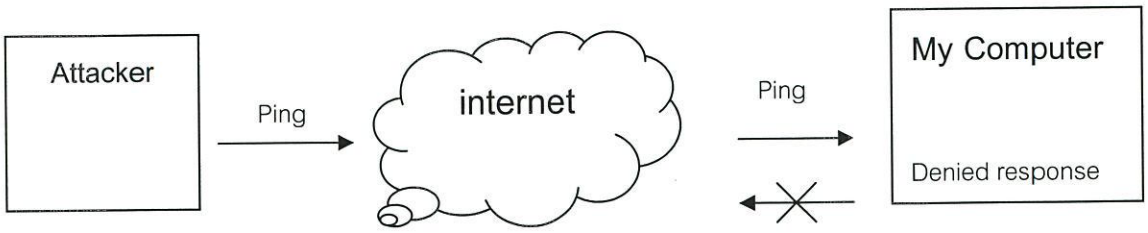


รูปที่ 3.4 แสดงองค์ประกอบโครงสร้างสถาปัตยกรรมการตรวจสอบข้อมูล

3.3 การออกแบบระบบการตรวจสอบการรับส่งข้อมูลผ่านระบบเครือข่าย

3.3.1 รูปแบบการตรวจสอบการรับส่งข้อมูลของโปรแกรม

รูปแบบการตรวจสอบข้อมูลของโปรแกรม จะกระทำการติดต่อ / ปิดกั้นการเชื่อมต่อก่อนที่โปรแกรมประยุกต์ในระบบจะทำการส่งหรือเปิดรับข้อมูล กรณีที่ถูกโจมตีจากบุคคลภายนอกจะป้องกันไม่ให้มีการตอบรับสัญญาณดังกล่าวเช่น การ ใช้โปรแกรม Ping (Packet Internet Groper) จากระบบคอมพิวเตอร์ที่อยู่ในเครือข่าย มายังระบบคอมพิวเตอร์ที่ได้ติดตั้งโปรแกรมของโครงงาน โปรแกรมจะปิดกั้นการตอบรับของสัญญาณดังกล่าวไม่ให้สามารถทราบสถานะของระบบคอมพิวเตอร์ได้ (Timeout)



รูปที่ 3.5 แสดงรูปแบบการป้องกันการรับส่งข้อมูลโดยใช้โปรแกรม Ping

3.3.2 การเชื่อมต่อกับระบบเครือข่ายที่เกี่ยวข้อง

การเชื่อมต่อกับระบบเครือข่ายในระบบปฏิบัติการวินโดวส์ Windows จะมีหมายเลข ตั้งแต่ 0-65535 ซึ่งหมายเลขการเชื่อมต่อที่เป็นมาตรฐาน จะอยู่ในช่วง 1-1024 นอกจากนั้นจะอนุญาตให้โปรแกรมต่างๆ สามารถใช้งานได้อย่างอิสระ

ในการพัฒนาโปรแกรม จำเป็นจะต้องรู้ความหมายของการเชื่อมต่อต่างๆ ซึ่งโดยปกติ จะประกอบไปด้วย การเชื่อมต่อ (Port) ดังต่อไปนี้

Service	Port
Web server	80/tcp
SSL (Secure Sockets Layer) Web server	443/tcp
FTP	21/tcp
POP3	110/tcp
SMTP	25/tcp
Remote Desktop (Terminal Services)	3389/tcp
IMAP3	220/tcp
IMAP4	143/tcp
Telnet	23/tcp
SQL Server	1433/tcp
LDAP	389/tcp
MSN Messenger	1863/tcp
Yahoo! Messenger	5050/tcp
AOL Instant Messenger and ICQ	5190/tcp
IRC (Internet Relay Chat)	6665-6669/tcp
DNS	53/udp

รูปที่ 3.6 รูปแบบการเชื่อมต่อที่ใช้กันโดยปกติ

3.3.3 นโยบายการรักษาความปลอดภัย

เนื่องจากระบบรักษาความปลอดภัยของข้อมูลจำเป็นที่จะต้องมีข้อบัญญัติในการรักษาความปลอดภัย ซึ่งในส่วนนี้ ผู้ใช้ สามารถทำการแก้ไข หรือปรับค่าได้ตามลักษณะการทำงานของผู้ใช้ได้ แต่ผู้ใช้ไม่ได้มีความชำนาญในเรื่องดังกล่าวได้เหมือนกัน ทางโครงงานจึงกำหนดนโยบายการรักษาความปลอดภัยของข้อมูลเบื้องต้นได้ดังต่อไปนี้

ตารางที่ 3.1 ตารางการควบคุมการเชื่อมต่อเบื้องต้นของโปรแกรม

Rule Number	Source IP	Destination IP	Service (port)	Access
1	Any	Web Server	HTTP(80) HTTPS(443)	Accept
2	Any	Mail Server	SMTP(25) POP3(110)	Accept
3	Any	DNS Server	DNS(53)	Accept
4	Mail Server	Any	SMTP(25)	Accept
5	DNS Server	Any	DNS(53)	Accept
6	Internat network	Any	HTTP(80) HTTPS(443) FTP(20-21) TELNET(23) SSH(22) SMTP(25) POP3(110) IMAP(143)	Accept
7	Any	Any	Any	Deny

3.3.4 Win32 API (Window 32 Application Interface) ที่เกี่ยวข้อง

3.3.4.1 WSOCK32

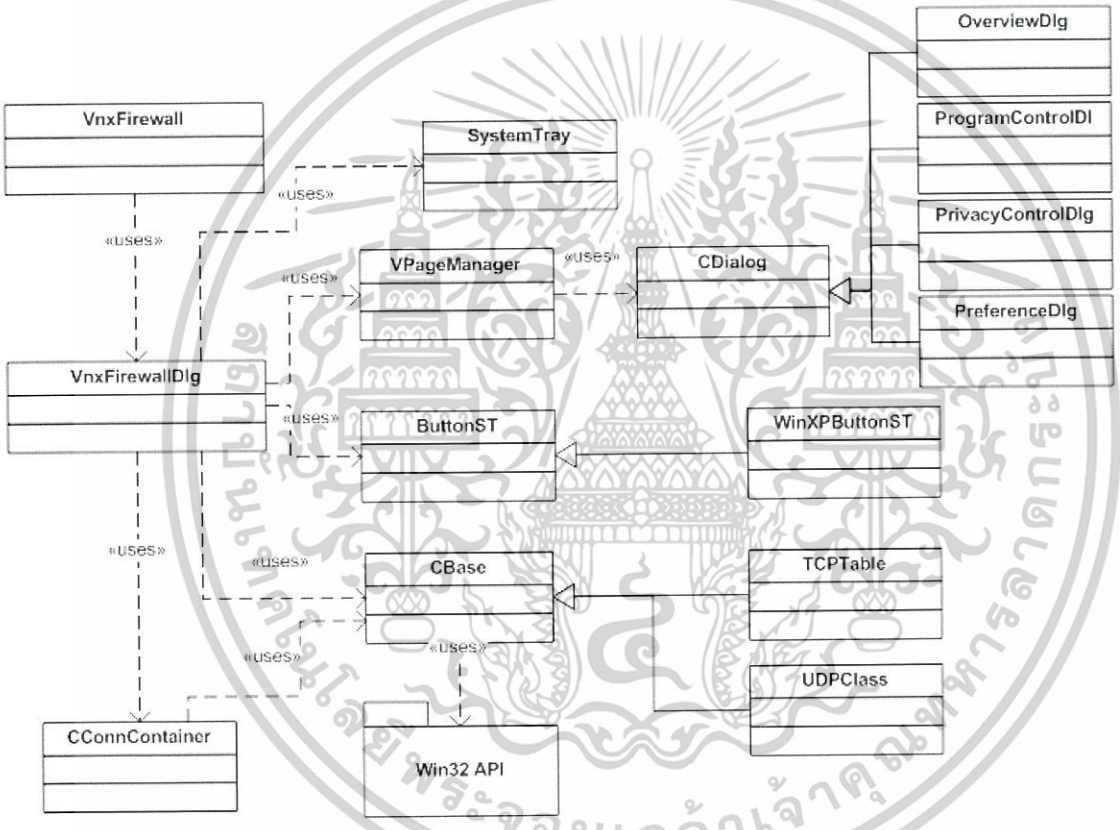
เป็น แอปพลิเคชั่นโปรแกรมอินเตอร์เฟส ที่ใช้ในการติดต่อกับระบบเครือข่ายโดย API ดังกล่าวจะใช้ในการตรวจสอบการร้องขอการเชื่อมต่อของโปรแกรมประยุกต์ต่างๆ รวมถึงการอ้างอิงต้นทางและปลายทางของข้อมูลที่จะทำการรับส่งข้อมูล โดยมีฟังก์ชันการทำงานที่จำเป็นได้แก่

1) closesocket – ทำการปิดการทำงานของ socket ที่ถูกสร้างเมื่อทำการร้องขอ
2) gethostbyname – ทำการขอข้อมูลเกี่ยวกับระบบคอมพิวเตอร์ที่ทำการส่งข้อมูลโดยดูจาก โดเมนเนม
3) WSAStartup – เป็นฟังก์ชันเริ่มต้นที่แต่ละโปรแกรมจะทำการใช้งาน socket ซึ่งจะช่วยในการตรวจสอบหากมีโปรแกรมประยุกต์ใดที่ทำการร้องขอการเชื่อมต่อกับระบบเครือข่าย
4) ntohl – เป็นฟังก์ชันการแปลง 32-บิต เนตเวิร์กแอดเดรส จากลำดับเนตเวิร์กไบต์ เป็นตามลำดับ โฮสต์ไบต์
5) htons – เป็นฟังก์ชันการแปลงค่า16-บิตจากโฮสต์ไบต์ เป็น เนตเวิร์กไบต์ htons จะเป็นการยืนยันว่า ลำดับไบต์ดังกล่าว สามารถถูกใช้ได้โดย winsock
6) ioctlsocket – จะเป็นฟังก์ชันในการควบคุมการรับส่งของข้อมูล ซึ่งสามารถตั้งค่าหรือแสดงค่าที่ควบคุมการรับส่งข้อมูลได้

บทที่ 4
ผลการศึกษาและการดำเนินงาน

4.1 โครงสร้างของโปรแกรม

ในส่วนของโปรแกรม ได้พัฒนาบนภาษา C++ ซึ่งส่วนติดต่อกับผู้ใช้ได้ใช้ชุดไลบารีของไมโครซอฟ ในการสร้าง ซึ่งคือ MFC (Microsoft Foundation Class) รูปแบบการทำงานจึงแบ่งออกเป็น คลาสต่างๆ ได้ดังต่อไปนี้



รูปที่ 4.1 โครงสร้างของโปรแกรม

4.1.1 คลาส VnxFireWall

เป็นคลาสโปรแกรมหลักที่ใช้ในการกำหนดค่าเริ่มต้นต่างๆ

4.1.2 คลาส VnxFireWallDlg

เป็นคลาสไดอะลอกหลักของโปรแกรม ที่เก็บการทำงานทั้งหมดของโปรแกรมไว้

4.1.3 คลาส CConnContainer

เป็นคลาสควบคุมการตรวจสอบการเชื่อมต่อของระบบปฏิบัติการว่า มีโปรแกรมใด ทำการเปิดพอร์ตไปยังระบบเครือข่ายโดยมีความถี่ในการตรวจสอบ คงที่

4.1.4 คลาส CBase

เป็นคลาสต้นแบบที่เก็บชุดคำสั่งในการเข้าถึงข้อมูลในระบบปฏิบัติการ โดยผ่าน Win32 application programming interface (API)

4.1.5 คลาส TCPTable และ UDPClass

เป็นคลาสที่เก็บข้อมูลของโปรโตคอลที่ได้จากการเรียกขอจากระบบปฏิบัติการ โดยใช้ชุดคำสั่งจากคลาส CBase ในการค้นหาข้อมูล

4.1.6 คลาส SystemTray

เป็นคลาสยูทิลิตี้ ที่ช่วยในการซ่อนโปรแกรมให้อยู่ในลักษณะของไอคอน โดยแสดงอยู่ ในมุมล่างขวาของระบบปฏิบัติการวินโดว์

4.1.7 คลาส ButtonST และ WinXPButtonST

เป็นคลาสยูทิลิตี้ ที่ช่วยปรับเปลี่ยนรูปแบบการแสดงผลของปุ่มให้ผู้ใช้สังเกตุได้ง่ายขึ้น

4.1.8 คลาส VPageManager

เป็นคลาสยูทิลิตี้ ที่ช่วยในการจัดการการเปลี่ยนแปลงหน้าไดอะลอค เมื่อมีการกดคำสั่งไปยังหน้าเพื่อแสดงข้อมูลตามลักษณะของแต่ละหน้า

4.1.9 คลาส OverviewDlg

เป็นคลาสหน้าจอที่เก็บการแสดงผลการทำงานโดยรวมของระบบ

4.1.10 คลาส PrivacyControlDlg

เป็นคลาสหน้าจอที่เก็บข้อมูลการเปิดพอร์ต ต่างๆ รวมไปถึงการกำหนดสิทธิ์ ในการเปิดพอร์ต เพื่อเชื่อมต่อไปยังระบบเครือข่าย

4.1.11 คลาส ProgramControlDlg

เป็นคลาสหน้าจอที่เก็บข้อมูลการเชื่อมต่อของโปรแกรมต่างๆ ไปยังระบบเครือข่าย

เป็นคลาสหน้าจอที่เก็บข้อมูลการเชื่อมต่อของโปรแกรมต่างๆ ไปยังระบบเครือข่าย

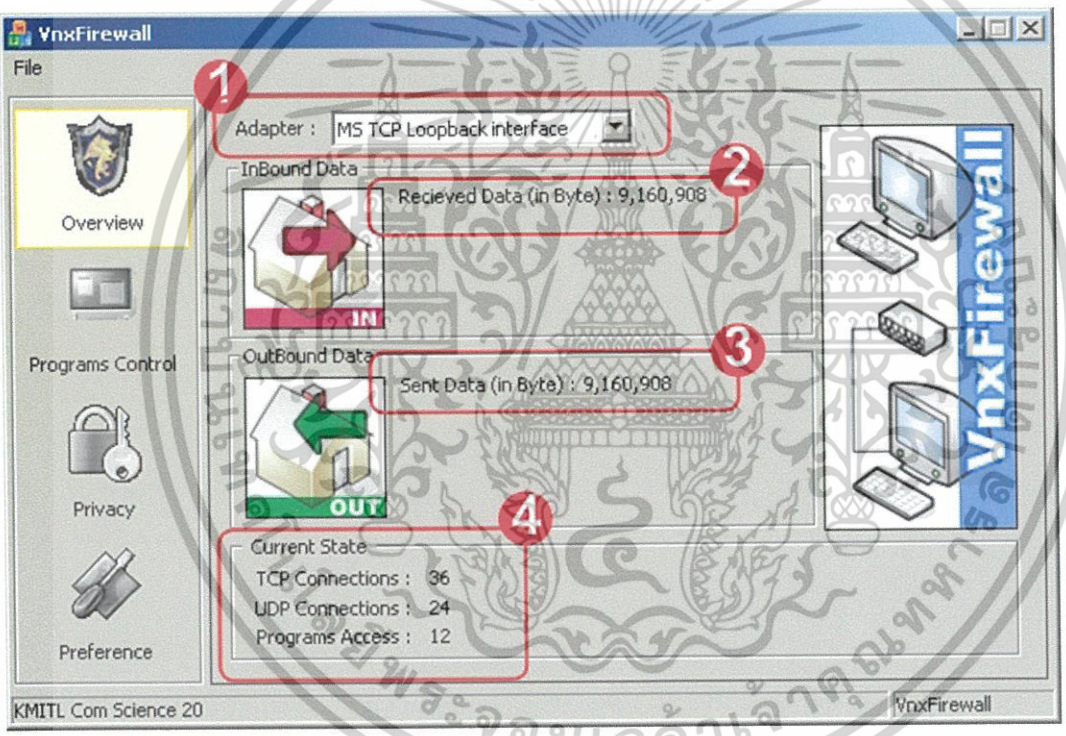
4.1.12 คลาส PreferenceDlg

เป็นคลาสหน้าจอที่เก็บข้อมูลการตั้งค่าต่างๆของโปรแกรม

4.2 หน้าจอของระบบ

โปรแกรมนี้ได้แบ่งหน้าจอต่างๆออกเป็น 4 ส่วน ได้แก่ หน้าจอภาพรวมของระบบ, หน้าจอส่วนควบคุมโปรแกรม, หน้าจอการตั้งกฎ และ หน้าจอการปรับแต่ง โดยแต่ละส่วนจะมีหน้าที่ต่างกันไปดังนี้

4.2.1 หน้าจอภาพรวมของระบบ



รูปที่ 4.2 หน้าต่างภาพรวมของระบบ

หน้าจอภาพรวมของระบบเป็นส่วนที่ใช้ตรวจดูเครือข่ายโดยรวมอย่างคร่าวๆ แบ่งออกเป็น 4 ส่วนด้วยกันคือ

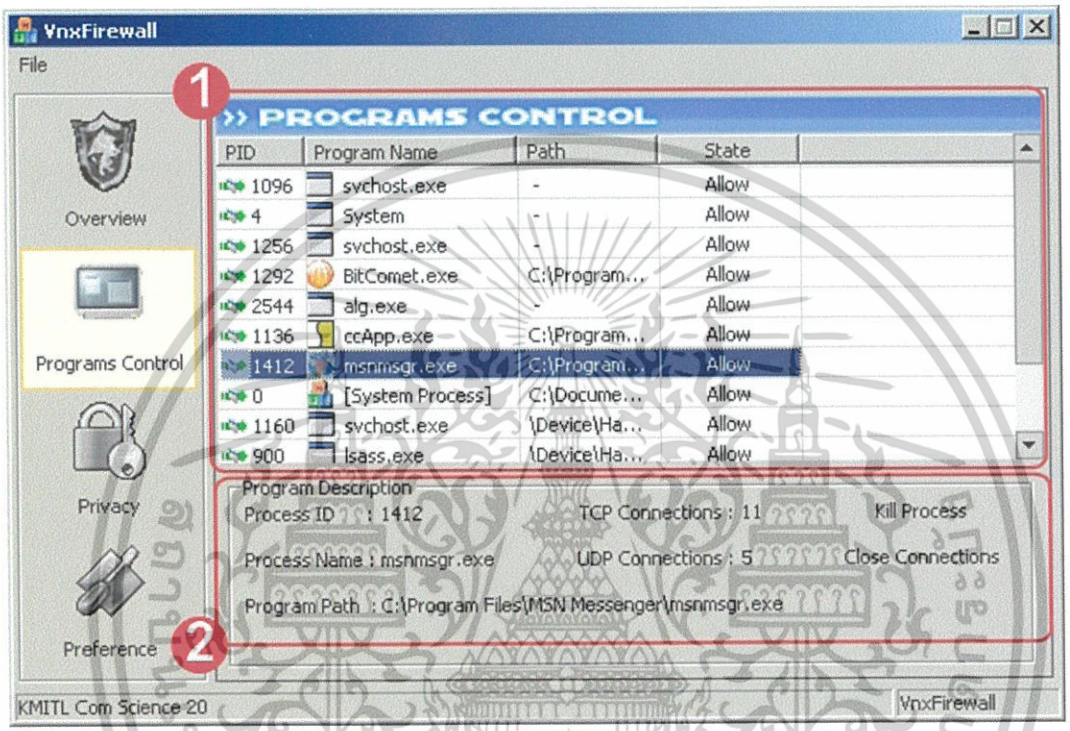
4.2.1.1 อแดพเตอร์ คือ ส่วนที่ให้ผู้ใช้เลือกตัวอแดพเตอร์ที่ต่อกับเน็ตเวิร์ค ที่จะให้แสดงผล

4.2.1.2 ข้อมูลเข้า คือ ส่วนแสดงถึงจำนวนข้อมูลแพ็กเก็ต ที่เข้ามาที่เครื่อง

4.2.1.3 ข้อมูลออก คือ ส่วนแสดงถึงจำนวนข้อมูลแพ็กเก็ต ที่ออกไปจากเครื่อ

4.2.1.4 สถานะปัจจุบัน คือ ส่วนที่แสดง ภาพรวมทั้งหมด ของแพ็กเก็ต และสถานะของเครื่องขณะนั้น

4.2.2 หน้าจอส่วนควบคุมโปรแกรม



รูปที่ 4.3 หน้าจอส่วนควบคุมโปรแกรม

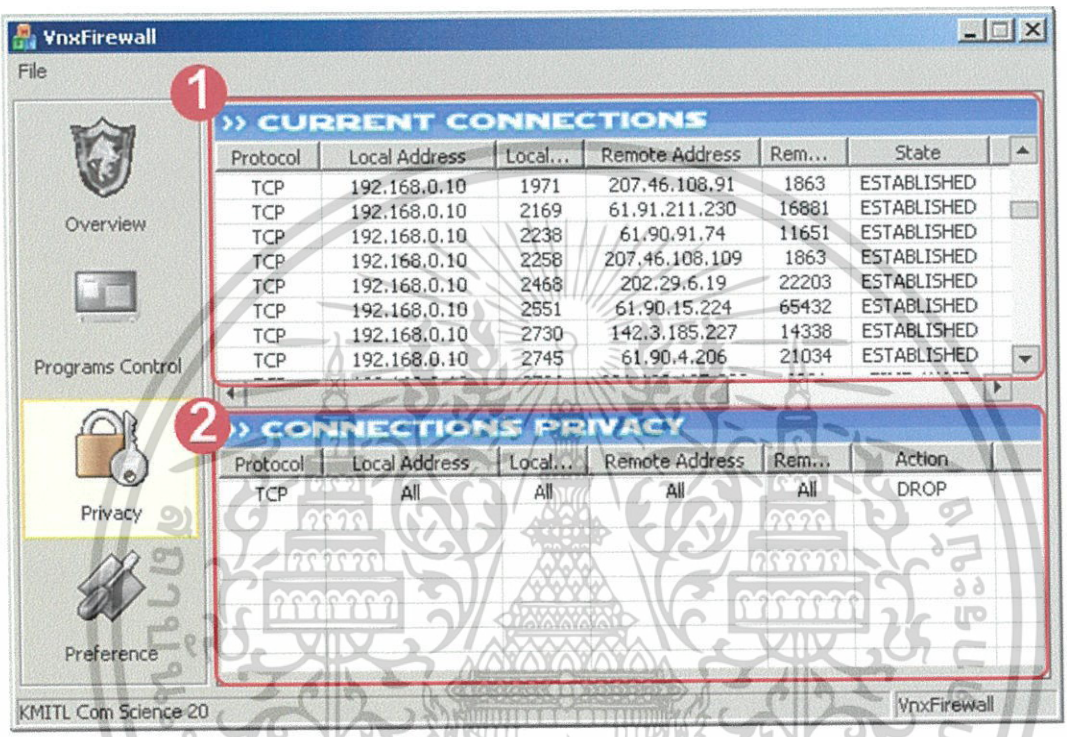
หน้าจอส่วนควบคุมโปรแกรม ใช้ในการตรวจสอบและควบคุมโปรแกรมทุกตัวที่ทำงานอยู่บนคอมพิวเตอร์โดยสามารถควบคุมการเชื่อมต่อกับเครือข่ายกับโปรแกรมเหล่านั้นได้ ในส่วนควบคุมโปรแกรม นี้ได้ถูกแบ่งออกเป็น 2 ส่วนได้แก่ ส่วนควบคุมโปรแกรม และ รายละเอียดของโปรแกรม

4.2.2.1 ส่วนควบคุมโปรแกรม คือ ส่วนที่แสดงว่าขณะนี้คอมพิวเตอร์กำลังเรียกโปรแกรมใดขึ้นมาทำงานบ้าง โดยจะแสดงถึง รหัสประจำตัว, ชื่อโปรแกรม, ที่อยู่ของโปรแกรม และสถานะว่าโปรแกรมตัวนั้นอนุญาตให้เชื่อมต่อกับเครือข่ายได้หรือไม่

4.2.2.2 ส่วนรายละเอียดของโปรแกรม คือ ส่วนที่จะแสดงถึง รายละเอียดของโปรแกรม ที่เราเลือก ซึ่งจะแสดงถึง รหัสประจำตัว ชื่อโปรแกรม จำนวนการเชื่อมต่อที่เป็น TCP

และ UDP ที่อยู่ของโปรแกรม และสามารถที่จะปิดโปรแกรมหรือปิดการเชื่อมต่อกับระบบเครือข่ายโดยทันทีได้

4.2.3 หน้าจอการตั้งกฎ



รูปที่ 4.4 หน้าจอการตั้งกฎ

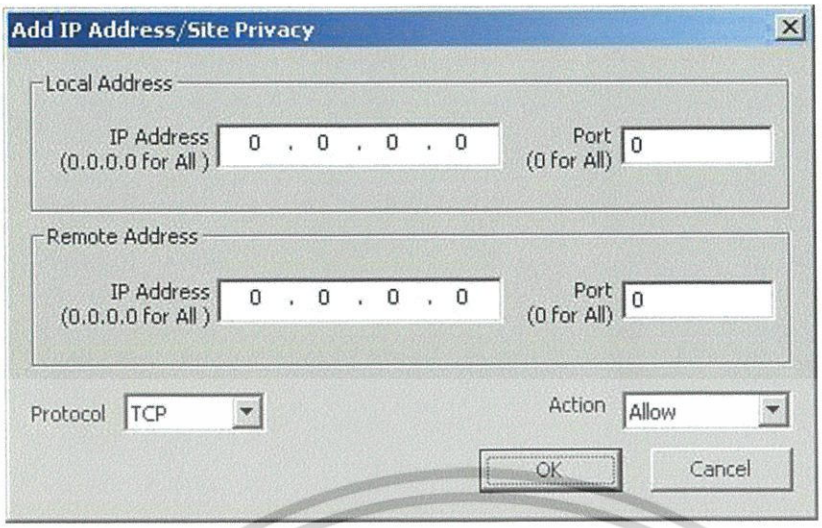
ในส่วนนี้เปรียบเสมือนส่วนตั้งกฎที่ทำหน้าที่ควบคุมพฤติกรรมของเครือข่ายให้อยู่ในกฎกติกาที่เราตั้งไว้โดยประกอบด้วย 2 ส่วนได้แก่ การเชื่อมต่อในปัจจุบัน และ กฎในการเชื่อมต่อ

4.2.3.1 การเชื่อมต่อในปัจจุบัน

แสดงถึงการเชื่อมต่อปัจจุบันระหว่างเครื่องเครื่องคอมพิวเตอร์ โดยแสดงถึงโปรโตคอลที่ใช้ ip address กับพอร์ต ของเครื่องคอมพิวเตอร์ที่ใช้ระบบนี้, ip address กับพอร์ตของคอมพิวเตอร์ที่เชื่อมต่อไป และสถานะการเชื่อมต่อโดยจะมีการเปลี่ยนค่าไปทุกระยะ

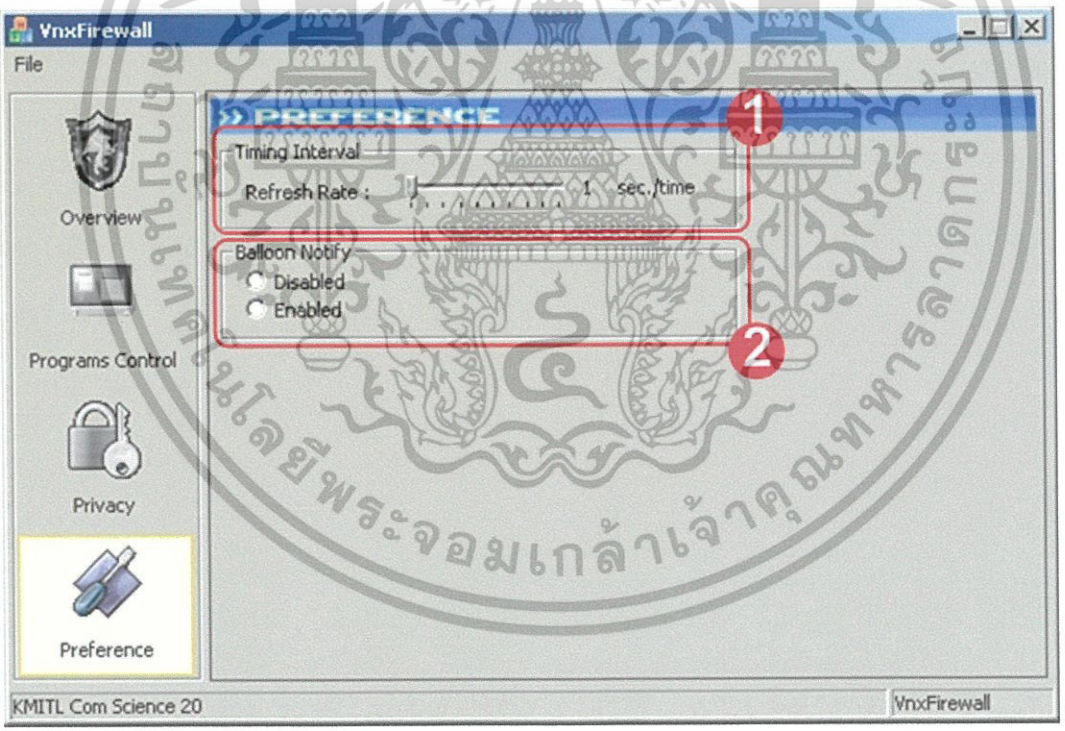
4.2.3.2 กฎในการเชื่อมต่อ

ในส่วนนี้ใช้ในการกำหนดค่ากฎกติกาในการเชื่อมต่อกับเครือข่ายทั้งหมดโดยสามารถตั้งค่าเพื่อใช้กับเครือข่ายทั้งหมดเพื่อป้องกันหรืออนุญาตการเชื่อมต่อกับเครือข่ายคอมพิวเตอร์ที่เรากำหนดได้ รูปที่ 4.5 แสดงการตั้งค่าการติดต่อจากเครือข่ายภายนอก



รูปที่ 4.5 การตั้งค่าการเชื่อมต่อกับเครือข่ายภายนอก

4.2.4 หน้าจอการปรับแต่ง



รูปที่ 4.6 หน้าจอการปรับแต่ง

จะเป็นส่วนที่ไว้สำหรับ ปรับแต่งค่าต่างๆ ของโปรแกรมโดยจะมี 2 ส่วนคือ

4.2.4.1 ระยะเวลาที่ใช้ในการปรับปรุง เป็นส่วนที่สามารถปรับค่าความเร็วในการที่จะปรับปรุง การเชื่อมต่อในหน้าจอส่วนควบคุมโปรแกรม และ หน้าจอการตั้งกฎ

4.2.4.2 การแสดงบอลลูน เป็นส่วนที่จะให้เลือกว่าให้โปรแกรมแสดงว่าเมื่อมีการเชื่อมต่อใหม่โปรแกรมก็จะแจ้งเตือนโดยใช้บอลลูนหรือไม่

บทที่ 5
สรุปผลปัญหาและข้อเสนอแนะ

5.1 สรุปผลปัญหาพิเศษ

จากการศึกษาวิจัยและพัฒนาระบบตรวจสอบและป้องกันการรับส่งข้อมูลคอมพิวเตอร์ผ่านระบบเครือข่าย สามารถที่จะสรุปความสามารถของโปรแกรมได้เป็นข้อๆ ดังนี้

1. สามารถตรวจสอบโปรแกรมที่ทำการเชื่อมต่อทั้งหมดของเครื่องที่ติดต่อไปยังเครือข่ายต่างๆได้
2. สามารถป้องกันโปรแกรมที่ไม่ได้รับอนุญาตเชื่อมต่อกับเครือข่ายภายนอกได้
3. สามารถป้องกันการเชื่อมต่อเข้ามาจากเครือข่ายภายนอกที่ไม่ได้รับอนุญาตได้
4. สามารถป้องกันการเชื่อมต่อออกจากเครื่องไปยังเครือข่ายภายนอกที่ไม่ได้รับอนุญาตได้
5. มีการใช้งานที่ง่ายและสะดวกต่อผู้ใช้งานที่ไม่มีความรู้ด้านเครือข่ายคอมพิวเตอร์

5.2 ข้อจำกัดของปัญหาพิเศษ

ในการทำปัญหาพิเศษนี้ เกิดข้อจำกัด ทำให้ไม่สามารถพัฒนาโปรแกรมได้เป็นไปอย่างดี ซึ่งข้อจำกัดที่เกิดขึ้น มีดังนี้

1. การศึกษาเกี่ยวกับการเขียนโปรแกรมเพื่อติดต่อกับระบบปฏิบัติการวินโดวส์เป็นไปได้ยากเนื่องจากระบบปฏิบัติการวินโดวส์ไม่โอเพนซอสและไม่มีหลักสูตรที่สอนเขียนโปรแกรมบนระบบปฏิบัติการวินโดวส์โดยตรง ทำให้การศึกษาเองเป็นไปได้ยาก
2. โปรแกรมนี้ยังมีข้อจำกัดในการตรวจจับคือในบางครั้งอาจจะตรวจจับไม่ทันในบางกรณีที่เกิดความเปลี่ยนแปลงขึ้นรวดเร็วมาก
3. โปรแกรมนี้ไม่สามารถใช้งานได้บนระบบปฏิบัติการวินโดวส์ที่ต่ำกว่า เอ็กซ์พี

5.3 ข้อเสนอแนะและแนวทางการศึกษาต่อ

1. โปรแกรมนี้สามารถนำเอาไปพัฒนาต่อให้เป็นโปรแกรมป้องกันโทรจันโดยทำให้โปรแกรมสามารถวิเคราะห์ได้ว่าโปรแกรมตัวใดมีพฤติกรรมที่อาจจะก่อให้เกิดความเสี่ยงต่อข้อมูลในเครื่องคอมพิวเตอร์ได้
2. ผู้พัฒนาควรจะมีความรู้ทางด้านการเขียนโปรแกรมเกี่ยวกับระบบปฏิบัติการวินโดวส์
3. ควรเขียนโปรแกรมให้อยู่ในรูปแบบวินโดวส์เซอวิส ไม่ใช่ในรูปซิงเกิลโปรเซส

บรรณานุกรม

จตุชัย แพงจันทร์. 2546. **เจาะระบบ Network ฉบับสมบูรณ์.** นนทบุรี : ไอดีซี

นิรุธ อำนวยศิลป์. 2544. **Visual C++ Version 6.0 ฉบับเพื่อใช้งานจริง.** พิมพ์ครั้งที่ 4. กรุงเทพฯ : ซัคเซส มีเดีย จำกัด

ยุทธนา ลีลาศวัฒนกุล. 2546. **คู่มือการเขียนโปรแกรมและใช้งาน Visual C++ .NET ฉบับสมบูรณ์.** นนทบุรี : อินโฟเพรส.

Abraham Silberschatz et. al. 2003. **OPERATING SYSTEM CONCEPTS**. 6th ed. NEW YORK : JOHN WILEY & SONS INC.

Chris Maunder. 2005. **The Code Project.** [online]. Available : http://www.codeproject.com.

Microsoft. 2005. **MSDN Library.** [online]. Available : http://msdn.microsoft.com/library/default.asp

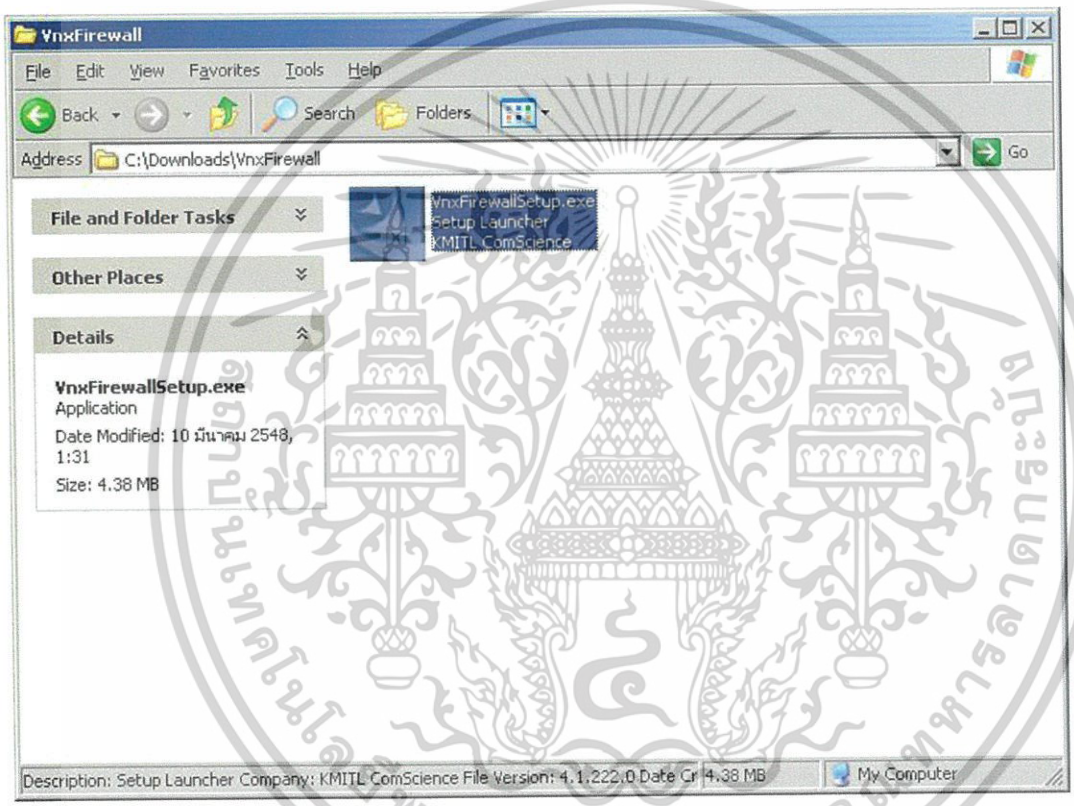
http:// rootkit.host.sk

http://www.wilson.demon.co.uk

ภาคผนวก ก
คู่มือการติดตั้งโปรแกรม

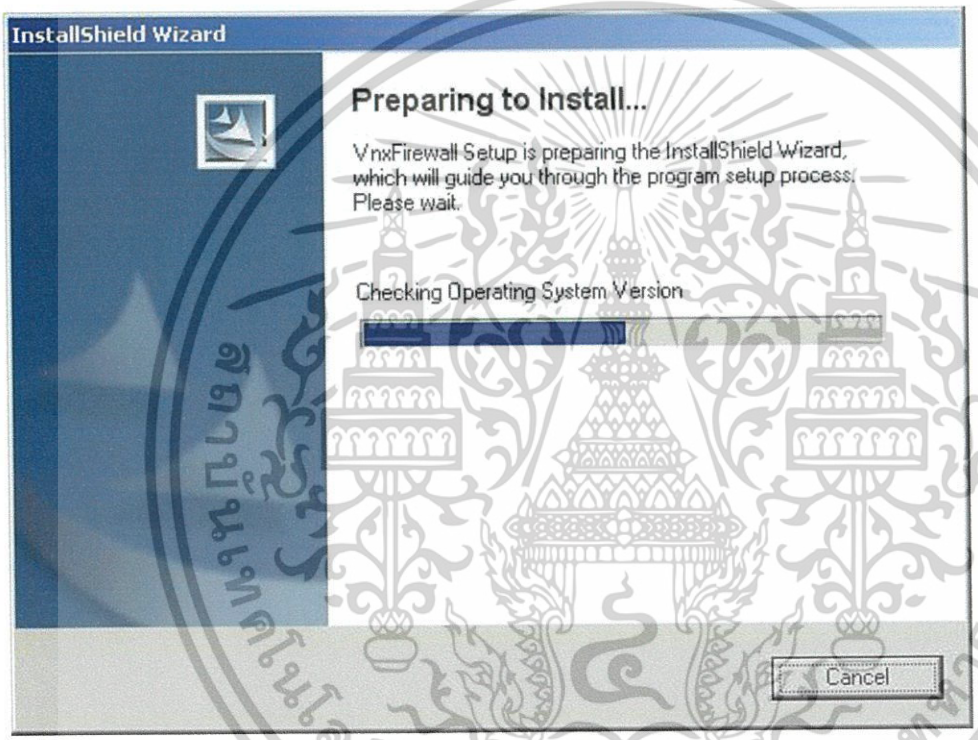
ขั้นตอนการติดตั้งโปรแกรม

1. เปิดไฟล์ VnxFirewallSetup.exe เพื่อเริ่มทำการติดตั้งโปรแกรม VnxFirewall



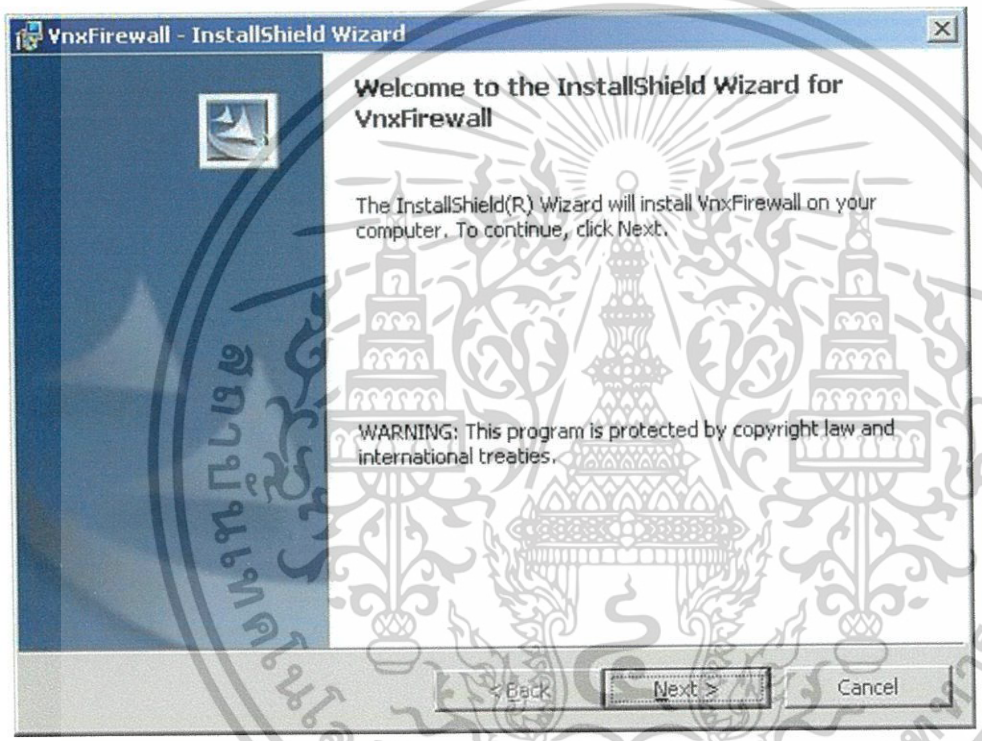
รูปที่ ก.1 : ไฟล์ที่ใช้ในการติดตั้งโปรแกรม

2. จะปรากฏหน้าต่างเตรียมพร้อมเพื่อการติดตั้ง



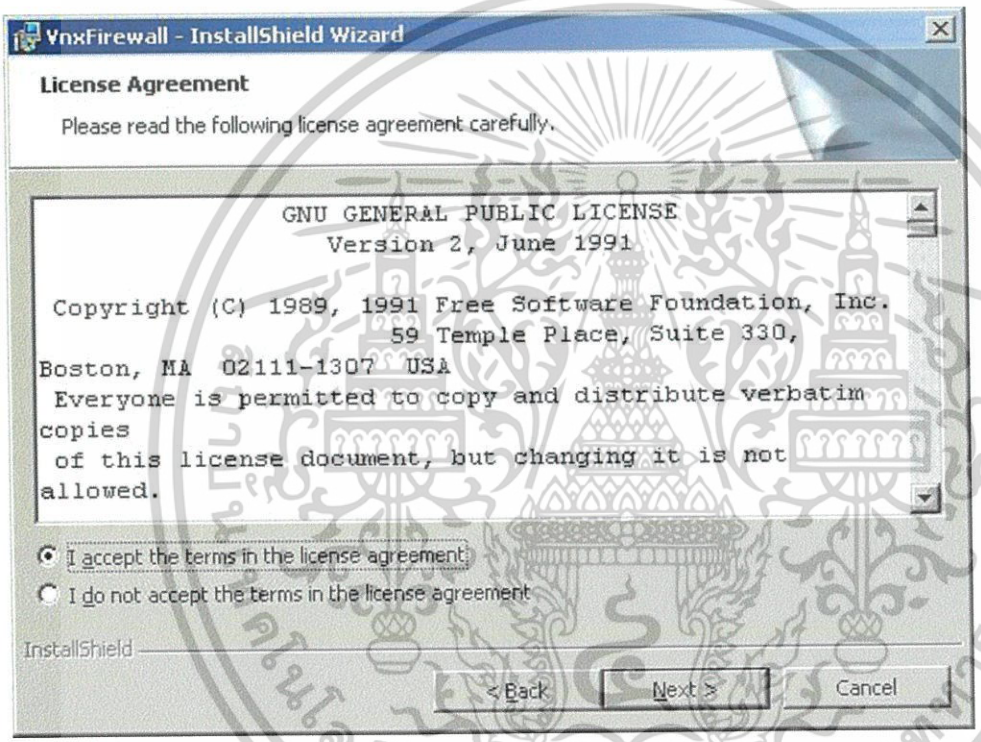
รูปที่ ก.2 : หน้าต่างเตรียมการติดตั้ง

3. เมื่อพบกับหน้าต้อนรับให้กด Next เพื่อดำเนินการต่อ



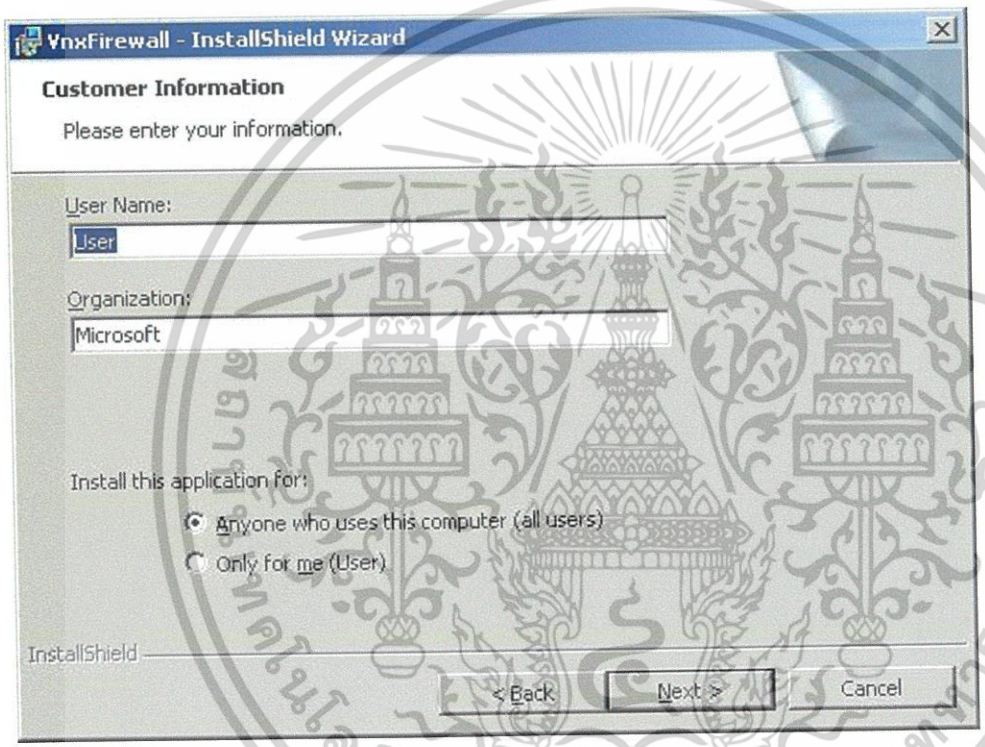
รูปที่ ก.3 : หน้าต่างต้อนรับ

4. หน้าต่างลิขสิทธิ์ให้เลือกที่ Accept the terms in the license agreement แล้วกด Next เพื่อดำเนินการต่อไป



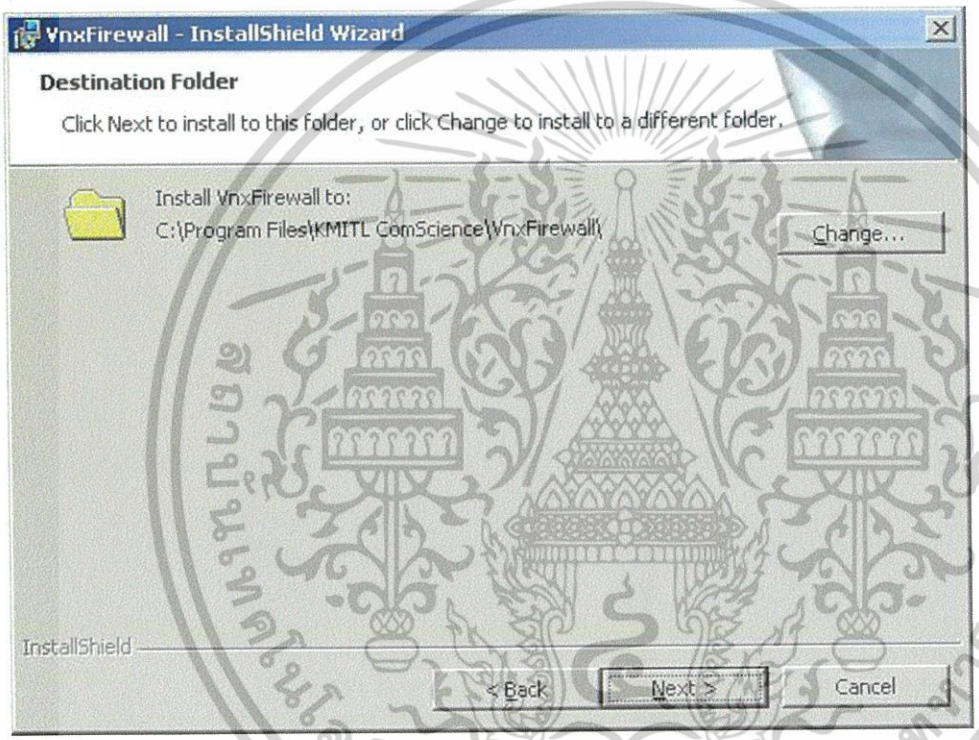
รูปที่ ก.4 : หน้าต่างลิขสิทธิ์

5. หน้าต่างผู้ใช้ใส่ชื่อผู้ใช้ หรือ บริษัท แล้วเลือกว่าจะให้โปรแกรมนี้ใช้ได้ทุกคนหรือว่าใช้ได้เพียงแค่ผู้ใช้ที่ใช้อยู่ตอนนี้คนเดียวเท่านั้น แล้วกด Next เพื่อดำเนินการต่อไป



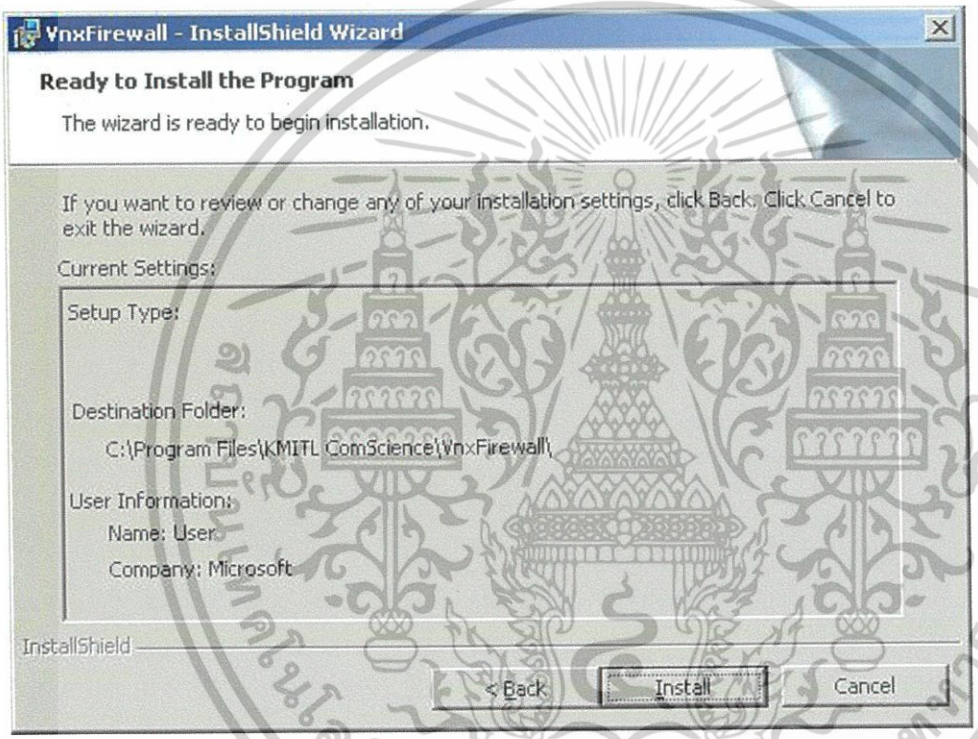
รูปที่ ก.5 : หน้าต่างระบุผู้ใช้งาน

6. เลือก Folder ที่จะลงโปรแกรม ซึ่งตามมาตรฐานแล้วจะลงที่ Folder Program Files แต่สามารถเปลี่ยนได้โดยกดเลือกที่ Change เมื่อเสร็จสิ้นแล้วกด Next เพื่อดำเนินการต่อไป



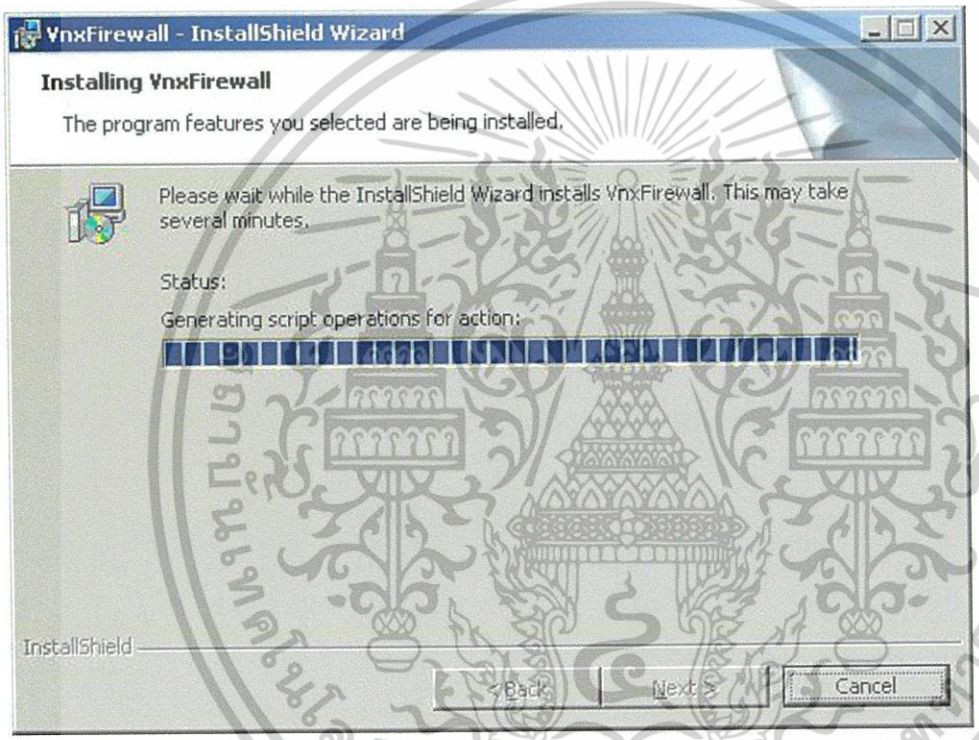
รูปที่ ก.6 : หน้าต่างเลือกตำแหน่งลงโปรแกรม

7. หน้าต่างบอกให้ทราบถึงข้อมูลของ Folder ที่จะลง และ ผู้ใช้ที่สามารถใช้งานได้ กด Install เพื่อทำการลงโปรแกรม



รูปที่ ก.7 : หน้าต่างแสดงรายละเอียดการลงโปรแกรม

8. กำลังทำการติดตั้งโปรแกรม VnxFirewall ไม่ควรเปิดโปรแกรมใดๆ ขณะทำการติดตั้ง แล้วรอสักครู่



รูปที่ ก.8 : หน้าต่างขณะติดตั้งโปรแกรม

9. การติดตั้งเสร็จเรียบร้อย กด Finish เพื่อเสร็จสิ้นการลงโปรแกรม

รูปที่ ก.9 : หน้าต่างเสร็จสิ้นการลงโปรแกรม