โปรแกรมสำรวจเครือข่ายและระบบคอมพิวเตอร์

Network and Computer System Viewer

นายบพิธ ธนกิจไพบูลย์

นายมานะ หวังสัจจะโชค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

โปรแกรมสำรวจเครือข่ายและระบบคอมพิวเตอร์

Network and Computer System Viewer

โดย

นายบพิธ ธนกิจไพบูลย์

นายมานะ หวังสัจจะโชค

อาจารย์ที่ปรึกษา

อาจารย์ ธนา หงษ์สุวรรณ

อาจารย์ อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมสำรวจเครือข่ายและระบบคอมพิวเตอร์

Network and Computer System Viewer

ผู้จัดทำ

1. นายบพิธ ธนกิจไพบูลย์ รหัสประจำตัว 43010226

2.นายมานะ หวังสัจจะโชค รหัสประจำตัว 43010338

_____ อาจารย์ที่ปรึกษา

อาจารย์ ธนา หงษ์สุวรรณ

_____ อาจารย์ที่ปรึกษา

อาจารย์ อัครเดช วัชระภูพงษ์

โปรแกรมดูแลระบบคอมพิวเตอร์และเครือข่าย

นายบพิธ	ธนกิจไพบูลย์	43010226
นายมานะ	หวังสัจจะโชค	43010338
อาจารย์ ธนา	หงษ์สุวรรณ	อาจารย์ที่ปรึกษา
อาจารย์ อัครเดช	วัชระภูพงษ์	อาจารย์ที่ปรึกษา

บทคัดย่อ

ในปัจจุบันนี้มีการใช้งานเครือข่ายการสื่อสารข้อมูลภายในองค์กรต่าง ๆ อย่างแพร่หลายทั้งในด้านการสื่อสารข้อมูลภายในองค์กรและการสื่อสารข้อมูลระหว่างองค์กร การใช้เครือข่ายการสื่อสารข้อมูลจึงจำเป็นที่จะต้องมีประสิทธิภาพในการทำงานที่สูงเพื่อเป็นการสนับสนุนระบบการทำงานขององค์กร

โครงงานนี้เป็นการนำเสนอ โครงการพัฒนาโปรแกรมสำรวจเครือข่ายและระบบคอมพิวเตอร์ (Network and Computer System Viewer) ซึ่งเป็นโปรแกรมที่ใช้สำหรับผู้ดูแลระบบเครือข่ายเพื่ออำนวยความสะดวกในการแสดงให้เห็นสถานะปัจจุบันของระบบเครือข่ายและสภาพของระบบคอมพิวเตอร์ที่ใช้งานระบบเครือข่าย โดยนำเสนอเทคนิคต่าง ๆ ที่ใช้ในการรวบรวมข้อมูลของระบบคอมพิวเตอร์บนเครือข่าย, ส่วนที่ใช้ในการจัดเก็บข้อมูล และส่วนที่ใช้ในการจัดการเพื่อนำมาแสดงผลบนโปรแกรม

รวมทั้งมีรายละเอียดในการพัฒนาโปรแกรมเพื่อให้ผู้ที่สนใจสามารถนำไปเป็นแนวทางในการพัฒนาเครื่องมือที่ใช้อำนวยความสะดวกในการจัดการเครือข่ายอื่น ๆ เพื่อลดการนำเข้าโปรแกรมต่าง ๆ จากต่างประเทศ

Network and Computer System Viewer

Borpit Thanakijpaiboon	43010226
Mana Wongsatjachock	43010338
Thana Hongsuwan	Advisor
Akkradach Watcharapupong	Advisor

ABSTRACT

Presently, Communication Network is applied widely in many organizations include both internal communication and external communication. Therefore, Communication Network must have high performance to support the organization work.

This project describes the network and computer system viewer that is the administrator tool program for network monitoring . Including describes the use of enumeration technique for gathering network information, storing database and information processing to display. Addition developing details for developer.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่สำเร็จลุล่วงไปด้วยดี หากปราศจากคำแนะนำและการให้คำปรึกษาจาก อาจารย์ ธนา หงษ์สุวรรณ และ อาจารย์ อัครเดช วัชระภูพงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอขอบคุณเป็นอย่างสูง

ขอขอบคุณพี่ๆและเพื่อนๆ ห้องปฏิบัติการ ISAG ทุกคนที่ให้ความช่วยเหลือในเรื่องต่างๆ จนสำเร็จไปได้ด้วยดี

ขอขอบคุณตัวข้าพเจ้าทั้งสองที่ไม่ท้อแท้ไปเสียก่อนที่จะประสบความสำเร็จ ขอบคุณที่มีชีวิตมาถึงทุกวันนี้ และโอกาสดีๆ ที่ได้รับมา

และต้องขอขอบคุณบุคคลสำคัญที่สุดทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างดี และยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆ ด้านอันหาที่เปรียบมิได้ ขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

บพิธ ธนกิจไพบูลย์

มานะ หวังสัจจะโชค

สารบัญ

สารบัญตาราง

สารบัญรูปภาพ

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

ในปัจจุบันนี้ระบบเครือข่ายคอมพิวเตอร์กำลังเป็นที่นิยมอย่างสูง ทำให้จำนวนผู้ใช้ในระบบเพิ่มขึ้นอย่างรวดเร็วและมีผลให้ระบบเครือข่ายคอมพิวเตอร์มีขนาดใหญ่และมีความซับซ้อนมากขึ้น การดูแลระบบเครือข่ายก็ทำได้ยากขึ้นตามไปด้วย โดยการดูแลระบบเครือข่ายเบื้องต้นก็จำเป็นต้องทราบแผนภาพเครือข่ายว่าสถานะปัจจุบันของเครือข่ายนั้นเป็นเช่นไรเช่น ในองค์กรขนดใหญ่นั้นมีเครือข่ายย่อย (LAN)เชื่อมต่อกันอย่างไร มีการเชื่อมต่อจากไหนไปไหนบ้าง เพื่อไว้สำหรับศึกษาและแก้ไขระบบของเครือข่ายได้ ว่ามีปัญหาอย่างไรหรือว่าต้องมีการปรับปรุงอย่างไร

ในการที่จะสร้างรูปแผนภาพจำลองระบบเครือข่ายคอมพิวเตอร์ เราจำเป็นต้องทราบข้อมูลเบื้องต้นของระบบเครือข่ายที่ต้องการจะสร้างก่อน โดยเครื่องมือตัวนี้รวบรวมข้อมูลเพื่อการดูแลเครือข่ายคอมพิวเตอร์ได้โดยการ ส่ง Packet ข้อมูลไปและตรวจสอบ Packet ข้อมูลที่ได้รับกลับมา (Active SCAN) เพื่อเก็บข้อมูลเบื้องต้นต่าง ๆ เช่น มีอุปกรณ์เปิดอยู่ในเครือข่ายนี้จำนวนเท่าไร, อุปกรณ์ทำหน้าที่เป็นอะไร โดยใช้ เทคนิค การตรวจสอบแบบต่าง ๆ เช่น การ Ping Sweep , การอ่านค่าจาก SNMP Service และ อื่นๆ โดยการรวบรวมจะพยายามหาข้อมูลที่จำเป็นให้มากที่สุด เมื่อได้ข้อมูลต่างๆ มาแล้วก็จะนำข้อมูลเหล่านี้มาสร้างเป็นแผนภาพจำลองระบบเครือข่ายขึ้นโดยจะนำมาแสดงในรูปแบบของการเชื่อมต่อจริงของเครือข่ายนั้นๆ และมีการเปลี่ยนแปลงสถานะหากในระบบเครือข่ายเปลี่ยนแปลง เพื่อสะดวกในการดูแลระบบเครือข่ายแบบตลอดเวลา

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อพัฒนาต้นแบบของโปรแกรมที่ใช้ในการตรวจสอบระบบเครือข่ายคอมพิวเตอร์ให้สามารถอำนวยความสะดวกแก่ผู้ดูแลหรือผู้ที่ต้องการศึกษาระบบเครือข่าย

1.2.2 เพื่อพัฒนาโปรแกรมที่ใช้ในการอำนวยความสะดวกในการศึกษา ดูแล และตรวจสอบระบบเครือข่ายคอมพิวเตอร์

1.2.3 เพื่อนำไปใช้ทดแทนการนำเข้าโปรแกรมจากต่างประเทศ

1.3 ขอบเขตของโครงการ

โดยโปรแกรมจะมีคุณสมบัติพื้นฐานดังนี้

1.3.1 สามารถหาคอมพิวเตอร์ที่เปิดใช้งานเครือข่ายอยู่ในขณะนั้นได้และเก็บสถิติวัน เวลาการเปิด ปิด

1.3.2 สามารถหา OS ได้จำเพาะเจาะจงมากขึ้น

1.3.3 สามารถดู service หรือ port ที่เปิดของแต่ละเครื่องได้

1.3.4 สามารถตรวจสอบ พาสเวิร์ดโพลีซีได้

1.3.4 สามารถอ่านค่า SNMP Service ที่สนใจในการบอกลักษณะของคอมพิวเตอร์ในระบบเครือข่ายได้

1.3.5 สามารถที่จะตรวจสอบการเปิด service ที่ไม่เหมาะสมของเครื่องที่อยู่ในเครือข่ายได้

1.3.6 สามารถตรวจสอบการเปิด share directory ที่ไม่เหมาะสมของเครื่องในเครือข่ายได้

1.3.7 สามารถสั่ง run โปรแกรมภายนอกให้สะดวกแก่การใช้ เช่น เพิ่มเติมโปรแกรม Script สำหรับ Update

1.3.8 สามารถหา Mac Address และบอก Vendor ของผู้ผลิตอุปกรณ์นั้นได้

1.3.9 สามารถดู Traffic ของเครือข่ายแต่ละเครือข่ายได้

1.3.10 สามารถที่จะแสดงสถานะของเครือข่ายว่า สามารถที่จะเชื่อมต่อกับ internet ได้ตลอดเวลาหรือไม่ พร้อมกับเก็บข้อมูลเป็นสถิติ

1.3.11 สามารถนำข้อมูลที่รวบรวมมาได้แสดงผลในรูปของ กราฟิกและแผนภูมิ

1.3.12 สามารถนำผลลัพธ์แลดงผลออกเป็นรายงานและ system log ได้

1.3.13 สามารถที่จะตรวจสอบเครื่องที่มีการใช้งานที่ไม่เหมาะสมเช่น เครื่องที่มีการเล่นเกมได้

1.3.14 สามารถที่จะตรวจสอบเครื่องที่อาจจะเป็นอันตรายต่อระบบเครือข่าย เช่น เครื่องที่ติดโทรจัน หนอนอินเตอร์เน็ต ได้

1.4 แผนการดำเนินงาน

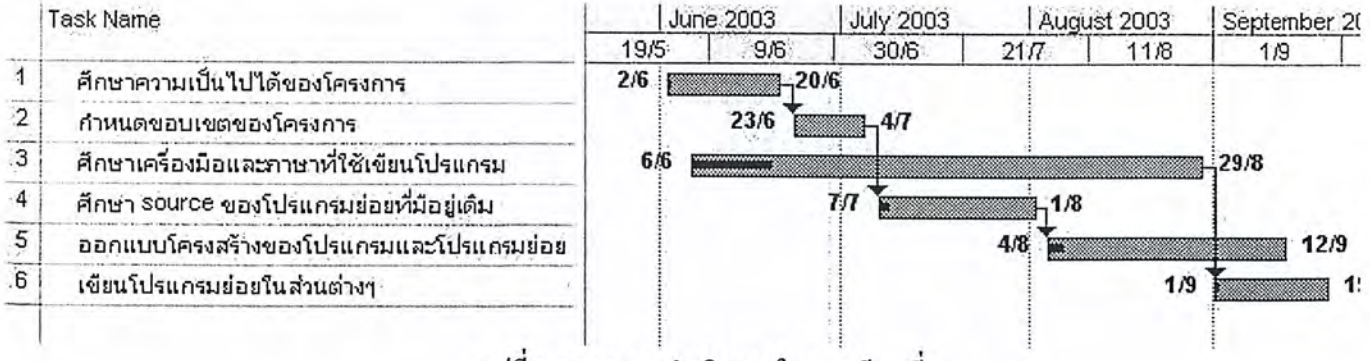
การดำเนินงานในภาคเรียนที่ 1 มีดังต่อไปนี้

ระยะเวลา : 80 วัน

วันเริ่มต้น : 2/6/2003

วันสิ้นสุด : 19/9/2003

% ความคืบหน้า : 45%



รูปที่ 1-1 แผนการดำเนินงานในภาคเรียนที่ 1

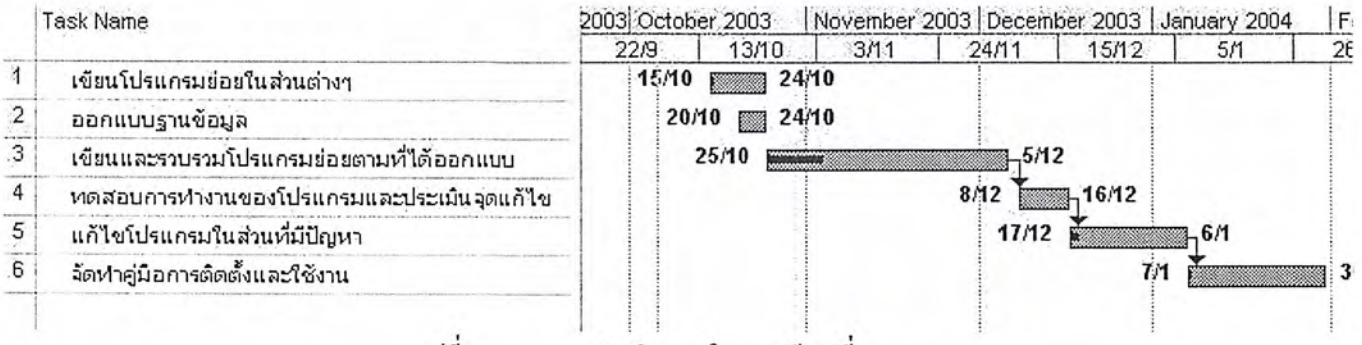
การดำเนินงานในภาคเรียนที่ 2 มีดังต่อไปนี้

ระยะเวลา : 78 วัน

วันเริ่มต้น : 15/10/2003

วันสิ้นสุด : 30/1/2004

% ความคืบหน้า : 55%



รูปที่ 1-2 แผนการดำเนินงานในภาคเรียนที่ 2

1.5 ประโยชน์ที่ได้รับ

1.5.1 โปรแกรมต้นแบบเพื่อใช้ในการจัดการและดูแลระบบเครือข่าย

1.5.2 ได้รับความรู้จากการศึกษา เกี่ยวกับเรื่องความปลอดภัย , การเดินทางของข้อมูลในระบบเครือข่ายและการเขียนโปรแกรมทางด้านเครือข่าย

1.5.3 ช่วยลดการนำเข้าโปรแกรมที่ใช้ในการดูแลระบบเครือข่ายจากต่างประเทศ

บทที่ 2
ทฤษฎีการบริหารเครือข่าย

ในปัจจุบันองค์กรต่างๆ มีการตื่นตัวอย่างมากในการที่จะปรับปรุงเทคโนโลยีของระบบการทำงานและการติดต่อสื่อสารภายในองค์กร เพื่อผลของความรวดเร็วและความคล่องตัวในการดำเนินงานต่างๆ สิ่งหนึ่งที่องค์กรเหล่านั้นต้องกระทำคือ การนำเครื่องคอมพิวเตอร์ภายในองค์กรมาเชื่อมต่อกัน สร้างเป็นระบบเครือข่ายคอมพิวเตอร์ ที่ทำงานภายใต้มาตรฐานใดมาตรฐานหนึ่งตามวัตถุประสงค์ขององค์กรนั้น และมาตรฐานหนึ่งซึ่งเป็นที่นิยมในปัจจุบันคือ มาตรฐานโพรโตคอล ทีซีพี/ไอพี (TCP/IP)

เหตุผลที่มาตรฐานนี้เป็นที่นิยมสืบเนื่องมาจากการที่ระบบอินเทอร์เน็ตมีการทำงานภายใต้มาตรฐานนี้และหากองค์กรสามารถเชื่อมต่อเข้ากับระบบอินเทอร์เน็ตได้ก็จะทำให้เข้าถึงแหล่งข้อมูลทั่วโลกรวมทั้งบริการต่างๆ ที่มีอยู่มากมาย ดังนั้นเพื่อความง่ายและความสมบูรณ์แบบที่จะเข้ากันได้กับระบบอินเทอร์เน็ต หลายองค์กรจึงเลือกใช้มาตรฐานโพรโตคอลทีซีพี/ไอพีหรือเรียกเป็นระบบอินทราเน็ต

2.1 บทบาทของผู้ดูแลระบบและความจำเป็นในการดูแลระบบเครือข่าย

เมื่อมีระบบเครือข่ายก็จำเป็นจะต้องมีผู้ดูแลระบบเครือข่าย (Administrator) หมายถึงผู้ที่ทำหน้าที่ตรวจสอบดูแลและแก้ไขปัญหาอันเกิดกับการทำงานและสถานะต่างๆ ของอุปกรณ์ในระบบเครือข่าย จากหน้าที่ที่กล่าวมาจะเห็นว่า บุคคลที่จะเป็นผู้ดูแลระบบได้นั้นต้องมีความรู้ ประสบการณ์และความอดทนสูง และยังต้องเป็นผู้มีคุณธรรมด้วย ซึ่งถ้าในระบบเครือข่ายใหญ่หรือมีความซับซ้อนมากๆ อุปกรณ์ต่างๆ ภายในระบบก็มีมากขึ้นจนเกินความสามารถของผู้ดูแลระบบที่จะรับภาระได้จึงได้มีกลุ่มบุคคลที่พยายามแก้ไขและพัฒนาเทคโนโลยีด้านนี้ โดยทำการศึกษาและวิจัยพฤติกรรมการทำงานของระบบ องค์ประกอบที่จำเป็นต่อการบริหารระบบ ปัจจัยที่มีผลกระทบต่อระบบปัญหาของความหลากหลายของผลิตภัณฑ์ในระบบ รูปแบบการจัดเก็บของข้อมูลจัดการระบบ และอื่นๆ อีกมากมาย จนกระทั่งก่อกำเนิดเป็นรูปแบบของการบริหารของการบริหารงานระบบเครือข่ายอย่างง่าย (Simple Network Management) ต่อมาได้ถูกกำหนดให้เป็นมาตรฐานหนึ่งของ ISO (The International Organization for Standardization) และ CCITT (The International Telegraph and Telephone Consultative Committee) และในส่วนหนึ่งของมาตรฐานนี้ก็ได้มีการกำหนดโพรโตคอลที่ทำหน้าที่รับส่งข้อมูลการจัดการ โพรโตคอลที่ทำหน้าที่รับส่งข้อมูลการจัดการ โพรโตคอลนี้คือ โพรโตคอล เอสเอ็นเอ็มพี

2.2 ความต้องการในการจัดการระบบเครือข่าย

2.2.1 ใช้ในการควบคุมระบบเครือข่าย และทรัพยากรต่างๆ ให้มีประสิทธิภาพ

2.2.2 เพื่อควบคุมความซับซ้อนอันเนื่องมาจากการที่มีจำนวนของอุปกรณ์ในระบบเครือข่าย, ผู้ใช้งาน, โพรโตคอลที่ถูกใช้งานอยู่บนระบบเครือข่าย เพิ่มจำนวนมากขึ้น

2.2.3 เป็นการเพิ่มการบริการ(Service) ในการใช้ข้อมูลและทรัพยากรต่างๆ

2.2.4 เพื่อเพิ่มดุลยภาพการใช้ข้อมูลของโปรแกรมประยุกต์ หมายถึงว่า การที่ภายในองค์กรมีการใช้ข้อมูลและทรัพยากรสำหรับโปรแกรมประยุกต์หลายๆตัว โดยโปรแกรมประยุกต์ดังกล่าวนั้นมีการใช้ข้อมูลและทรัพยากรที่แตกต่างกัน ดังนั้นก็ต้องมีการจัดลำดับความสำคัญของโปรแกรมโดยการระบุถึงระดับความปลอดภัย (Security) รวมถึงทรัพยากรที่มีจะใช้ในโปรแกรมประยุกต์ด้วย ผู้ควบคุมระบบเครือข่ายจะต้องกำหนดและควบคุมทรัพยากร ให้สมดุลจากการที่มีความต้องการหลากหลายเหล่านี้

2.2.5 ช่วยลดความซับซ้อน (Redundant) ในการออกแบบและบริหารระบบเครือข่าย

2.2.6 สามารถวิเคราะห์การใช้งานของระบบเครือข่ายได้ว่าช่วงเวลาใดที่มีการใช้งานระบบเครือข่ายมากหรือน้อย

2.3 จุดประสงค์ในการทำการบริหารระบบ

2.3.1 จัดการกับความผิดพลาดต่างๆที่เกิดขึ้นในระบบเครือข่าย (Fault Management)

เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นกับอุปกรณ์ต่างๆ ในระบบซึ่งเมื่อเกิดความผิดพลาดในการทำงาน ขึ้นก็จะมีการปฏิบัติดังต่อไปดังนี้

2.3.1.1 ทำการตรวจสอบจุดที่เกิดความผิดพลาดหรือทำงานล้มเหลว

2.3.1.2 ทำการแยกส่วนของระบบเครือข่ายที่ทำงานผิดพลาดแล้วแยกส่วนที่ทำงานผิดพลาดนั้นออกจากระบบ เพื่อให้ระบบเครือข่ายที่เหลืออยู่สามารถทำงานต่อไปได้

2.3.1.3 ทำการรีคอนฟิกระบบใหม่ให้สามารถทำงานได้ เพื่อทดแทนการทำงานในส่วนถูกแยกไปแล้ว

2.3.1.4 ทำการซ่อมแซมหรือเปลี่ยนแปลงส่วนที่เกิดความผิดพลาดขึ้น ให้กลับมาใช้งานได้ใหม่

2.3.2 จัดการเก็บข้อมูลต่างๆ ของระบบเครือข่าย (Accounting Management)

เพื่อให้ผู้ดูแลระบบสามารถที่จะติดตามและตรวจสอบดูเหตุการณ์(Event) รวมถึงจำนวนการใช้ทรัพยากรของระบบเครือข่าย ข้อมูลที่เราติดตามจะมีประโยชน์ดังนี้

2.3.2.1 สามารถตรวจสอบจำนวนผู้ใช้งานหรือกลุ่มของที่อาจมีการเรียกใช้ข้อมูลในส่วนที่ไม่ได้รับอนุญาต รวมถึงดูความแออัดของระบบเครือข่ายในช่วงเวลาใดเวลาหนึ่ง

2.3.2.2 ผู้ใช้งานอาจจะมีการใช้ระบบเครือข่ายได้ไม่เต็มประสิทธิภาพผู้ดูแลระบบสามารถวิเคราะห์ได้ว่าควรจะมีการเพิ่มทรัพยากรใดบ้าง เพื่อจะช่วยให้ผู้ใช้งานได้ใช้ระบบเครือข่ายได้อย่างเต็มประสิทธิภาพ

2.3.2.3 ผู้ดูแลระบบสามารถที่จะวางแผนการขยายระบบเครือข่าย ถ้าทราบรายละเอียดของกิจกรรมต่างๆ ของผู้ใช้งานเพียงพอ

2.3.3 จัดการเกี่ยวกับการคอนฟิกูเลชันระบบเครือข่าย (Configuration Management)

สามารถทำการคอนฟิกกูเลชั่นการทำงานของอุปกรณ์ต่างๆให้สามารถทำงานโดยอาศัยโปรแกรมประยุกต์ที่แตกต่างกันได้กับอุปกรณ์ชนิดเดียวกัน การที่เราจะเซตค่าต่างๆ หรือทำการคอนฟิกูเลชันเราจะทำจากสถานีจัดการ (Network Management Station:NMS)โดยการเซตจะทำการเซตค่าของแอตทริบิวต์และแวรรูของอุปกรณ์แต่ละตัว

2.3.4 จัดการบริหารประสิทธิภาพระบบเครือข่าย (Performance Management)

ในการติดต่อสื่อสารในระบบเครือข่ายคอมพิวเตอร์ จะประกอบด้วยส่วนประกอบต่างๆ มากมายที่ทำการเชื่อมต่ออยู่ในระบบเครือข่ายและมีการแบ่งปันทรัพยากรซึ่งกันและกัน การบริหารเรื่องประสิทธิภาพของระบบเครือข่ายจึงแบ่งได้ 2 ประเภทด้วยกันคือ

2.3.4.1 การตรวจสอบ (Network Monitoring)

2.3.4.2 การควบคุมระบบ (Network Controlling)

การตรวจสอบก็คือการที่ผู้ดูแลระบบ Administrator จะทำการตรวจสอบ Track ดูกิจกรรมต่างๆที่เกิดขึ้นบนระบบเครือข่าย ส่วนการควบคุมระบบก็คือการที่ผู้ดูแลระบบสามารถจะควบคุมหน้าที่การทำงานและบริหารอุปกรณ์ต่างๆ โดยการปรับหรือเซตค่าพารามิเตอร์ต่างๆ ภายในตัวอุปกรณ์ที่มีอยู่ในระบบเครือข่ายให้ทำงานได้ตามต้องการ

2.3.5 การบริหารงานด้านความปลอดภัย (Security Management)

การบริหารในด้านความปลอดภัยจะยึดหลักการกระจายข้อมูล (Data Distributing) และการเก็บรหัสลับรวมถึงการเข้ารหัสข้อมูลเพื่อให้เราสามารถเข้าถึงข้อมูลต่างๆได้อย่างปลอดภัย รวมถึงการใช้ในการตรวจสอบและควบคุมระบบเครือข่าย

2.4 หลักการบริหารระบบเครือข่ายโดยใช้โพรโตคอลเอสเอ็นเอ็มพี (Simple Network Management Protocol: SNMP)

ซิมเปิลเน็ทเวิร์กแมเนจเมนต์โพรโตคอล (Simple network management protocol: SNMP) ถูกพัฒนาโดยยึดหลักความง่ายและเป็นเครื่องมือที่ถูกสร้างมาในยุคต้นๆ โดยใช้ทำงานกับ โพรโตคอลทีซีพี/ไอพี (TCP/IP) ที่มีการใช้งานกันอย่างแพร่หลาย โดยเอสเอ็นเอ็มพี(SNMP) ได้ถูกกำหนดให้เป็นมาตรฐานหนึ่งของไอเอสโอ (The International Organization for Standardization: ISO) และซีซีไอทีที (The International Telegraph and Telephone Consultative Committee: CCITT) รายละเอียดของโพรโตคอลเอสเอ็นเอ็มพีจะได้กล่าวถึงอย่างละเอียดอีกครั้งหนึ่ง

โมเดลของการบริหารระบบเครือข่าย (Network Management) ที่ใช้สำหรับเครือข่ายทีซีพี/ไอพี (TCP/IP) ประกอบด้วยส่วนต่างๆดังต่อไปนี้

2.4.1 สถานีจัดการ (Management Station)

ตัวสถานีจัดการ (Management Station) เป็นเครื่องที่ใช้ในการบริหารระบบเครือข่าย หรืออาจจะกล่าวว่าเป็นเครื่องที่ใช้ติดต่อกับยูเซอร์ (User) นั้นเองซึ่งส่วนของสถานีจัดการ จะประกอบด้วยส่วนต่างๆดังต่อไปนี้คือ

2.4.1.1 โปรแกรมจัดการ (Management Application) สำหรับใช้วิเคราะห์ข้อมูลหรือใช้เฝ้าตรวจสอบระบบเครือข่าย

2.4.1.2 ส่วนของอินเตอร์เฟส (Interface) ทั้งที่เป็นส่วนที่ใช้ในการแสดงผล (Monitor) และส่วนที่ใช้ในการควบคุม (Control) ระบบเครือข่าย

2.4.1.3 ส่วนที่ใช้ในการดึงข้อมูลจากฐานข้อมูลการจัดการ (MIB) บนอุปกรณ์ต่างๆ ในระบบเครือข่ายมาใช้ หรือเปลี่ยนแปลงค่าได้

2.4.2 แมเนจเมนต์เอเจนต์ (Management Agent)

ส่วนอุปกรณ์อื่นๆที่มีซอฟต์แวร์เอเจนต์ฝังอยู่ เราจะเรียกว่าแมเนจเมนต์เอเจนต์ (Management Agent) เช่น โฮล (Host), บริจด์(Bridge), เราท์เตอร์(Router) และ ฮับ(Hub) เป็นต้น โดยอุปกรณ์ดังกล่าวสามารถถูกบริหารจากสถานีจัดการได้ ซึ่งอุปกรณ์แต่ละตัวจะมีโปรแกรมเอเจนต์ (Agent Software) ติดตั้งอยู่ และจะถูกมองเป็น ออปเจกต์ (Object) หนึ่งของระบบการจัดการ แต่ละอ๊อปเจกต์จะมีการนิยามฟังก์ชั่นการจัดการขึ้นกับอุปกรณ์นั้นๆ ในการตรวจสอบระบบเครือข่าย สถานีจัดการจะร้องขอข้อมูลจากเอเจนต์โดยระบุตำแหน่งข้อมูลที่ต้องการในฐานข้อมูลการจัดการ ส่วนการกำหนดค่า (Setting) การทำงานต่างๆ ก็สามารถทำในลักษณะเดียวกัน โดยการกำหนดค่าออปเจกต์ไอดีแล้วตามด้วยค่าที่ต้องการให้เปลี่ยนแปลงในฐานข้อมูลการจัดการของตัวเอเจนต์ที่ต้องการ

2.4.3 ฐานข้อมูลการจัดการ (Management Information Base: MIB)

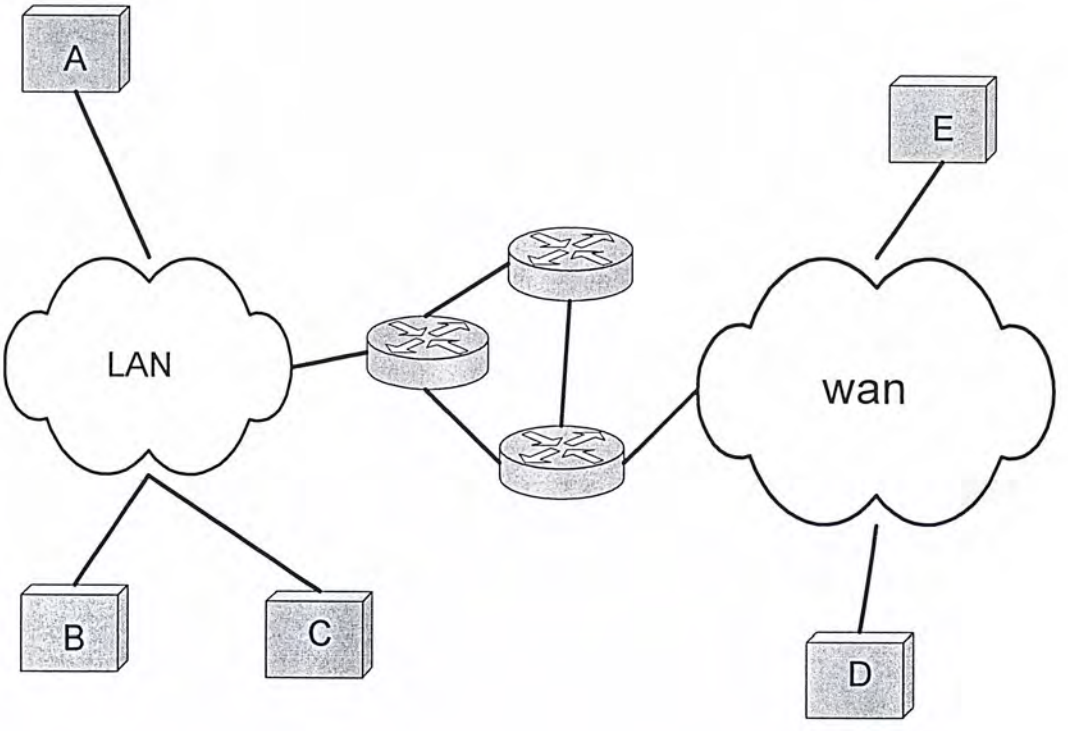
ฐานข้อมูลการจัดการคือส่วนของฐานข้อมูลที่ใช้สำหรับเก็บตัวเลขที่ใช้ระบุถึงโหนดของเอเจนต์ที่เราสนใจ การอ้างถึงโหนดภายในฐานข้อมูลการจัดการ สามารถใช้วิธีการอ้างอิงได้สองแบบ คือ การอ้างอิงแบบตัวเลข (Numerical) และการอ้างอิงแบบตัวอักษร (Symbolic)

2.5 เครือข่าย ทีซีพี/ไอพี(TCP/IP Network)

เครือข่ายทีซีพี/ไอพีหรือเครือข่ายอินเตอร์เน็ทเป็นเครือข่ายที่ใช้โพรโตคอลทีซีพี/ไอพีในการสื่อสารแลกเปลี่ยนข้อมูล อินเทอร์เน็ทโพรโตคอล (IP) เป็นขั้นตอนที่ทำให้ข้อมูลไปถึงจุดหมายปลายทางที่ต้องการ ส่วน Transmission Control Protocol (TCP) เป็นขั้นตอนที่จะรับประกันความถูกต้องของข้อมูลที่ส่งซึ่งถูกนำมาใช้งานร่วมกันในการติดต่อสื่อสารเรียกว่าทีซีพี/ไอพี

โพรโตคอลทีซีพี/ไอพีถูกออกแบบมาให้ทำหน้าที่เชื่อมโยงเครือข่ายกลุ่มย่อยๆเข้าด้วยกันจนกลายเป็นเครือข่ายที่มีขนาดใหญ่ ทำให้ผู้ใช้งานมองเป็นเครือข่ายเดียว ระบบปฏิบัติการยูนิกซ์(Unix) ได้นำเอาโพรโตคอลนี้รวมเข้าไว้ระบบปฏิบัติการเพื่อใช้ทำเป็น Network Operating System และมีผู้นิยมใช้เป็นจำนวนมาก การสื่อสารบนเครือข่ายทีซีพี/ไอพี ก็ถูกแบ่งการทำงานออกเป็นลำดับชั้น (Layer) 5

ชั้นประกอบด้วย Application Layer, transport Layer, Network Layer, Data Link Layer และ Physical Layer



รูปที่ 2-1 แสดงการเชื่อมต่อของเครือข่าย ทีซีพี/ไอพี

2.5.1 Application Layer

เป็นชั้นกล่าวถึงแอพพลิเคชั่นต่างๆที่ใช้งานหรือให้บริการบนเครือข่าย ทีซีพี/ไอพี เช่น E-mail, File transfer Protocol (FTP) และ Remote login

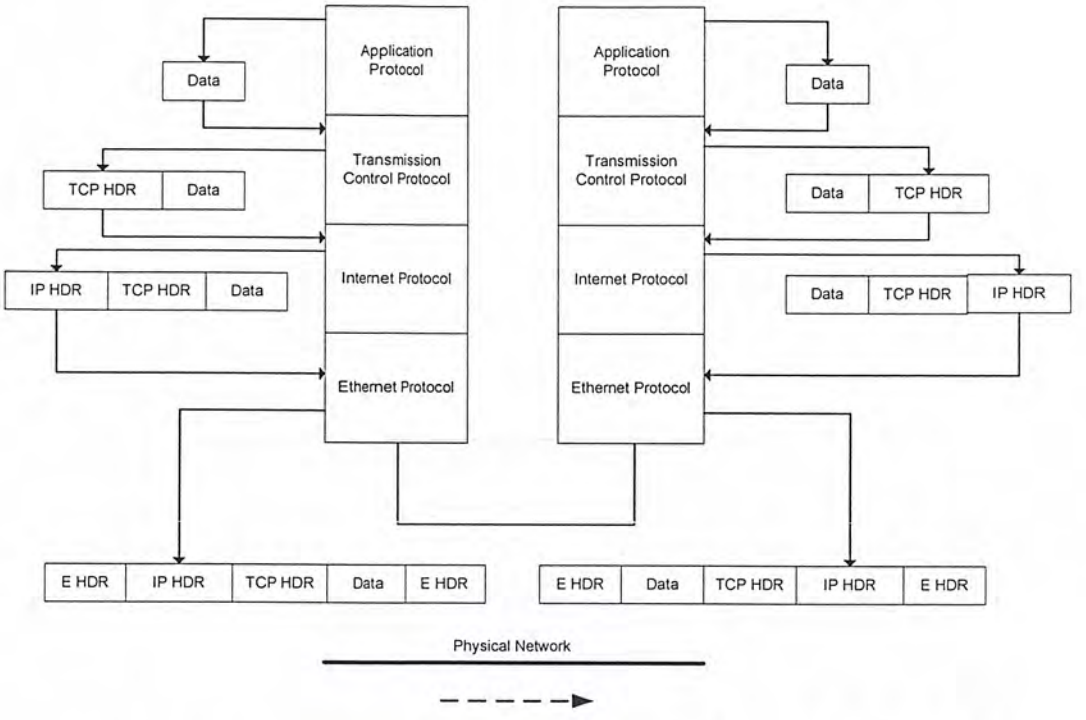
2.5.2 Transport Layer

ในชั้นนี้จะมี 2 โพรโตคอลที่ทำงานอยู่คือ Transport Control Protocol (TCP) และ User Datagram (UDP) โดยทีซีพีจะเป็นส่วนที่ทำงานในคอมพิวเตอร์มีหน้าที่ทำให้แน่ใจว่าไม่มีความผิดพลาดของการรับส่งข้อมูลเป็นลำดับครบถ้วน โดยแอพพลิเคชั่นที่ต้องการส่งข้อมูลจะผ่านการทำงานของทีซีพี

ซึ่งในทีซีพีจะทำการแบ่งข้อมูลที่รับเข้ามาออกเป็นส่วนย่อยๆเรียกว่าการทำเซกเมนต์ (Segment) เพื่อไม่ให้มีข้อมูลที่ยาวเกินไปช่วยลดความผิดพลาดของการส่งข้อมูลขนาดใหญ่ ในขณะเดียวกันทีซีพี (ฝั่งรับ) จะทำหน้าที่ส่งข้อมูลตอบรับ (Acknowledge) แจ้งให้ผู้ส่งข้อมูลรับทราบว่า

ได้รับข้อมูลเรียบร้อยแล้ว แต่ถ้าฝั่งส่งข้อมูลไม่ได้รับข้อมูลตอบรับภายในเวลาที่กำหนด ฝั่งส่งก็จะทำการส่งข้อมูลซ้ำไปอีกที เนื่องจากมีการแบ่งข้อมูลออกเป็นส่วนย่อยๆแล้วส่งออกไปซึ่งอาจไปทำให้ข้อมูลไปถึงปลายทางไม่เป็นลำดับดังนั้นหน้าที่อีกอย่างของทีซีพีคือการเรียงลำดับข้อมูลให้ถูกต้องแล้วประกอบกลับให้เป็นข้อมูลที่เหมือนกับฝั่งส่งข้อมูลทุกประการ

ส่วนยูดีพี (UDP) เป็นโพรโตคอลที่ไม่จัดการเกี่ยวกับการรับรองความถูกต้องของข้อมูลว่ามีข้อผิดพลาดหรือไม่ ไม่มีการจัดเรียงข้อมูลซึ่งยูดีพีเป็นโพรโตคอลที่มีกลไกในการทำไม่ซับซ้อนทำให้ง่ายต่อการใช้งาน



รูปที่ 2-2 แสดงการเอนแคปซูเลชั่นข้อมูล และการแลกเปลี่ยนข้อมูลบนเครือข่ายทีซีพีไอพี

2.5.3 Network Layer

ขั้นตอนขอไอพี (IP) เป็นขั้นตอนการส่งข้อมูลระหว่างเครื่อง ดังนั้นจึงต้องมีการระบุถึงหมายเลขประจำเครื่อง (ตำแหน่ง) ในระบบอินเทอร์เน็ตเป็นตัวระบุว่าข้อมูลจะส่งไปยังตำแหน่งใดในเครือข่าย ลักษณะของหมายเลขประจำเครื่องมีรูปแบบดังนี้ "161.246.4.3" ส่วนของหมายเลข เครือข่าย

คือ 161.246.0.0 จะถูกกำหนดโดยหน่วยงาน NIC ของกระทรวงกลาโหมสหรัฐเพื่อไม่ให้ซ้ำซ้อนกัน ส่วนตัวหลังเป็นเลขที่กำหนดโดยผู้บริหารเครือข่ายให้กับเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่าย

2.5.4 Data Link Layer

เป็นชั้นกล่าวถึง ขั้นตอน รูปแบบการส่งข้อมูลไปยังปลายทาง รวมถึงการตรวจสอบข้อมูลและการชนกันของข้อมูล (Error Detection/collection) โดยมีผู้กำหนดมาตรฐานขึ้นมาใช้การสื่อสารข้อมูล เช่น X.25, IEEE ฯลฯ IEEE ได้กำหนดมาตรฐาน IEEE 802 ที่กล่าวถึง

IEEE 802.3: เป็นเครือข่าย Ethernet ที่ใช้การตรวจสอบการชนกันแบบ Carrier sense multiple accesses with collision detection: CSMA/CD

IEEE 802.4: เป็นเครือข่าย Token Bus ที่ใช้ตัว Token กำหนดสิทธิในการส่งข้อมูลอุปกรณ์ใดที่ไม่มี Token จะไม่มีสิทธิในการส่งข้อมูล เป็นการป้องกันการชนกันของข้อมูลในเครือข่าย

IEEE 802.5: เป็นเครือข่าย Token Ring ที่ใช้ตัว Token กำหนดสิทธิในการส่งข้อมูล เช่นเดียวกับ Token Bus

2.5.5 Physical Layer

เป็นส่วนที่จัดการเกี่ยวกับการเชื่อมต่อทางกายภาพของเครือข่ายโดยจะกล่าวถึงตัวกลางในการสื่อสาร (Medium) รูปแบบของสัญญาณที่ใช้ในการส่งข้อมูลและรวมไปถึงกรรมวิธีการแปลงข้อมูลเป็นสัญญาณต่างๆ ที่สามารถส่งตัวกลางต่างๆเช่นได้แก่ สายเคเบิล, ใยแก้วนำแสง, คลื่นวิทยุ ฯลฯ

บทที่ 3
โพรโตคอลทีซีพี/ไอพี

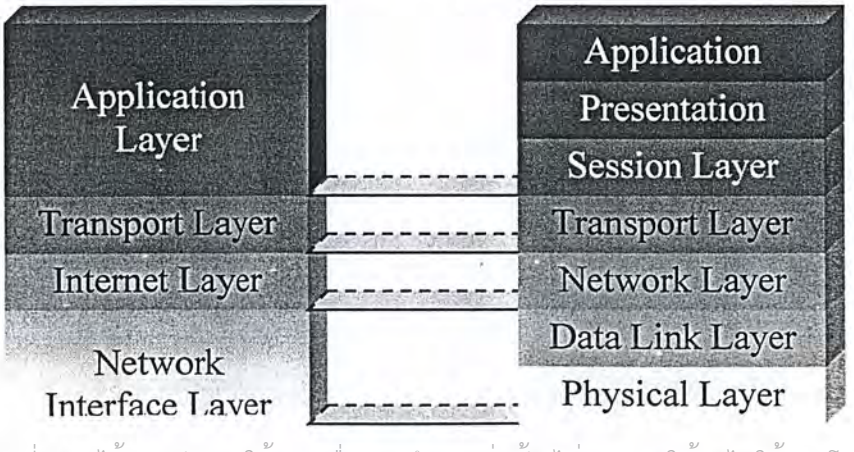
3.1 ความเป็นมาของโพรโตคอลทีซีพี/ไอพี

ทีซีพี/ไอพี เป็นโพรโตคอลมาตรฐานที่ใช้กันอยู่ในระบบปฏิบัติการแบบยูนิกซ์ เริ่มพัฒนาโดยกระทรวงกลาโหมของสหรัฐฯ ในปี ค.ศ. 1969 เพื่อเชื่อมโยงเครื่องคอมพิวเตอร์ทางทหารของแต่ละหน่วยที่อยู่ห่างไกลกัน โดยมีจุดประสงค์ คือสร้างระบบเครือข่ายให้เครื่องคอมพิวเตอร์สามารถรับส่งข้อมูลกันได้ แม้ว่าสายส่งข้อมูลบางส่วนจะถูกทำลายเสียหายไปก็ตาม เพื่อใช้งานในยามเกิดสงคราม โดยเครือข่ายที่จัดตั้งในระยะแรกชื่อว่า Advanced Research Projects Agency Network หรือ อาร์พาเน็ต (ARPANET)

ต่อมาได้พัฒนาเป็นเครือข่ายอินเทอร์เน็ต (INTERNET) โพรโตคอลนี้เหมาะสำหรับเชื่อมต่อคอมพิวเตอร์ทั้งใกล้ และไกลเข้าด้วยกัน และมีมาตรฐานรองรับทำให้ผู้ผลิตฮาร์ดแวร์ และซอฟต์แวร์สามารถสร้างอุปกรณ์ และโปรแกรมที่จะรองรับการทำงานของโพรโตคอลนี้ ทำให้เครื่องคอมพิวเตอร์สามารถรับส่งข้อมูลกันได้ไม่ว่าจะเป็นเครื่องขนาดเล็กหรือขนาดใหญ่ หรือใช้ระบบปฏิบัติการอะไรก็ตาม ทีซีพี/ไอพี (TCP/IP) เป็นชุดโพรโตคอลที่ประกอบด้วยโพรโตคอลต่างๆ หลายโพรโตคอล แต่ละโพรโตคอลมีคุณลักษณะ และมีความสามารถในการทำงานแตกต่างกัน โดยที่ในบทนี้ได้กล่าวถึงรายละเอียดและคุณสมบัติของโพรโตคอลที่สำคัญบางโพรโตคอล

3.2 การเชื่อมต่อของโพรโตคอลทีซีพี/ไอพี (TCP/IP Linking)

ทีซีพี/ไอพี (TCP/IP หรือ Transmission Control Protocol/Internet Protocol) เป็นโพรโตคอลในการสื่อสารในระบบอินเทอร์เน็ต และอินทราเน็ต มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ของฝ่ายรับ และฝ่ายส่งให้ได้รับข้อมูลที่ถูกต้องครบถ้วน หากข้อมูลที่ส่งมาเกิดการสูญหายระหว่างทางจะมีการแจ้งให้ต้นทางส่งข้อมูลมาใหม่ การทำงานของทีซีพี/ไอพีสามารถเปรียบเทียบกับโมเดลอ้างอิงโอเอสไอ (Open System Interconnection Reference Model: OSI) ตามมาตรฐานไอเอสโอ (International Organization for Standardization: ISO) ได้ดังรูปที่ 2-1

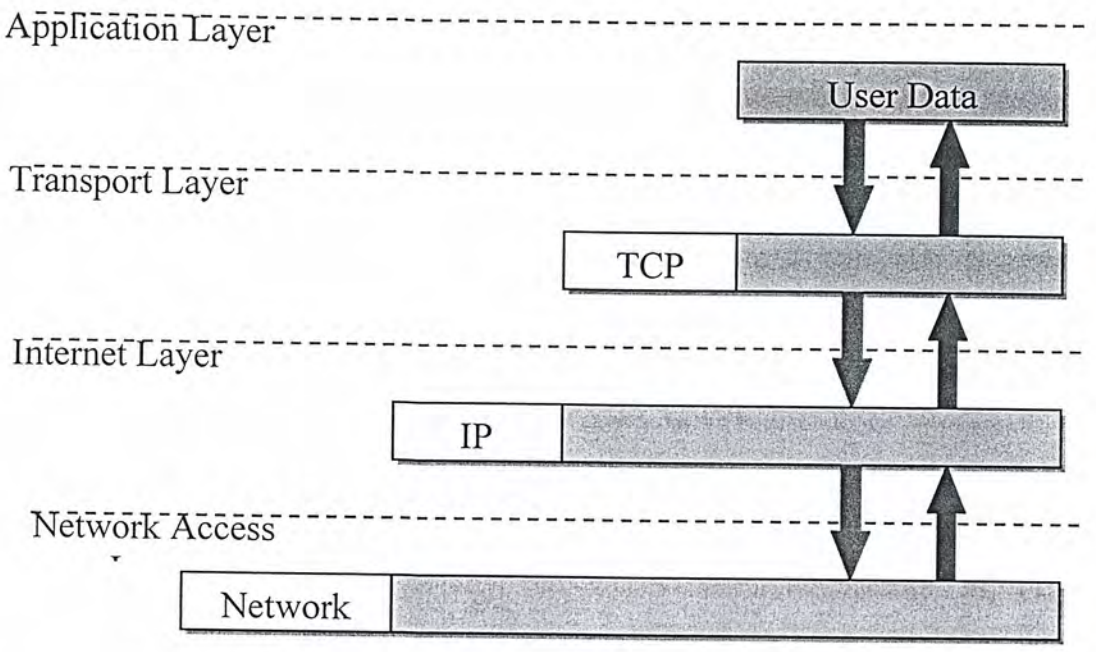


รูปที่ 3-1 แสดงการเปรียบเทียบเลเยอร์ของโอเอสไอกับเลเยอร์ของทีซีพี/ไอพี

ในแต่ละระดับชั้นของทีซีพี/ไอพีมีการทำงานที่แตกต่างกัน ตั้งแต่การติดต่อกับแอพพลิเคชันจนกระทั่งแปลงเป็นสัญญาณส่งไปตามสายสัญญาณ ซึ่งการทำงานในแต่ละระดับชั้นของทีซีพี/ไอพี มีดังตารางที่ 2-1

ชื่อระดับชั้น	หน้าที่
1. ชั้นแอพพลิเคชัน (Application Layer)	รองรับการทำงานของแอพพลิเคชันต่างๆ ที่ทำงานเป็นโพรเซสอยู่ในเครื่องต้นทางและปลายทาง โดยจัดการเชื่อมต่อระหว่างโพรเซส หรือแอพพลิเคชันที่อยู่ต่างเครื่องกัน โดยการทำงานของแอพพลิเคชันต่างๆมีการติดต่อกันตามแต่ละโพรโตคอลเฉพาะแล้วแต่แอพพลิเคชันที่ใช้งาน ซึ่งจะขอบริการจากชั้นทรานสปอร์ตอีกทีหนึ่ง
2. ชั้นทรานสปอร์ต (Transport Layer)	สร้างการเชื่อมต่อกันระหว่างแอพพลิเคชันแบบ end-to-end โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต (port) หรือซ็อกเก็ต (Socket) ในชั้นนี้มีบริการหลักอยู่ 2 แบบ คือ Connection Oriented โดยเรียกผ่านโพรโตคอลทีซีพี (TCP: Transmission Control Protocol) และ Connectionless ซึ่งเรียกผ่านโพรโตคอลยูดีพี (UDP: User Datagram Protocol) ซึ่งกล่าวถึงในหัวข้อถัดไป
3. ชั้นอินเทอร์เน็ต (Internet Layer)	ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโพรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆ ในอินเทอร์เน็ต คือ ไอพี (Internet Protocol: IP) ซึ่งกล่าวถึงในหัวข้อถัดไป นอกจากนี้ในชั้นนี้ยังมีโพรโตคอลทำงานอยู่ด้วยอีก 2 ชนิด คือ ไอซีเอ็มพี (Internet Control Message Protocol: ICMP) และเออาร์พี (Address Resolution Protocol: ARP)
4. ชั้นเน็ตเวิร์กอินเตอร์เฟส (Network Interface Layer)	แปลงข้อมูลให้อยู่ในรูปที่เหมาะสมกับเครือข่ายแต่ละแบบ ซึ่งแตกต่างกันออกไป และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่าย

ตารางที่ 3-1 การทำงานของแต่ละระดับชั้นของทีซีพี/ไอพี

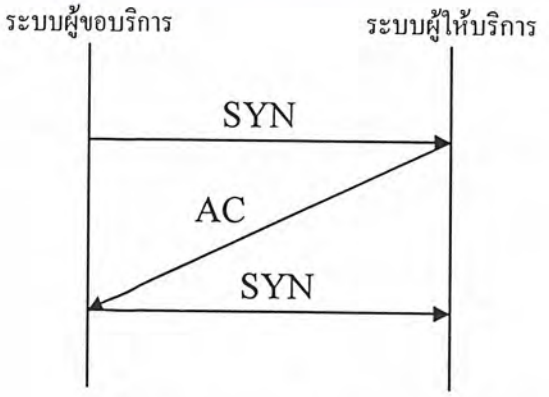


รูปที่ 3-2 แสดงการข้อมูลที่ส่งผ่านในโมเดลของทีซีพี/ไอพี

ในชุดโพรโตคอลทีซีพี/ไอพีนี้ มีโพรโตคอลหลัก ที่ขอกล่าวถึง 3 โพรโตคอล ได้แก่ โพรโตคอลทีซีพี โพรโตคอลยูดีพี ซึ่งทำงานในชั้นทรานสปอร์ต และโพรโตคอลไอพี ซึ่งทำงานในชั้นอินเทอร์เน็ต โดยมีรายละเอียดดังต่อไปนี้

3.3 โพรโตคอลทีซีพี (TCP: Transmission Control Protocol)

การทำงานที่สำคัญอย่างหนึ่งของโพรโตคอลทีซีพี คือ การทำ "3-Way Handshake" ซึ่งเป็นกระบวนการเริ่มต้นในการสร้างการเชื่อมต่อในชั้นทรานสปอร์ต กล่าวคือ ในการติดต่อกันระหว่างระบบในเครือข่ายต้องมีการสร้างการเชื่อมต่อไปยังระบบที่ให้บริการก่อน โดยผู้ขอบริการส่งสัญญาณ SYN เพื่อขอบริการ จากนั้นผู้ให้บริการจะส่งสัญญาณ ACK เพื่อตอบรับการเชื่อมต่อที่ร้องขอมา จึงสามารถรับส่งข้อมูลกันได้ ดังรูปที่ 2-3



รูปที่ 3-3 แสดงการทำ 3-Way Handshake

การเชื่อมต่อแบบ 3-Way Handshake นี้ เป็นการตรวจสอบความพร้อมของทั้งฝ่ายส่ง และฝ่ายรับ และการกำหนดค่าเริ่มต้นของพารามิเตอร์ต่างๆ ของทั้งสองฝ่ายให้ตรงกัน หลังจากกระบวนการทำ 3-Way Handshake สิ้นสุด ทั้งสองฝ่ายจึงสามารถรับ และส่งข้อมูลซึ่งกัน และกันได้

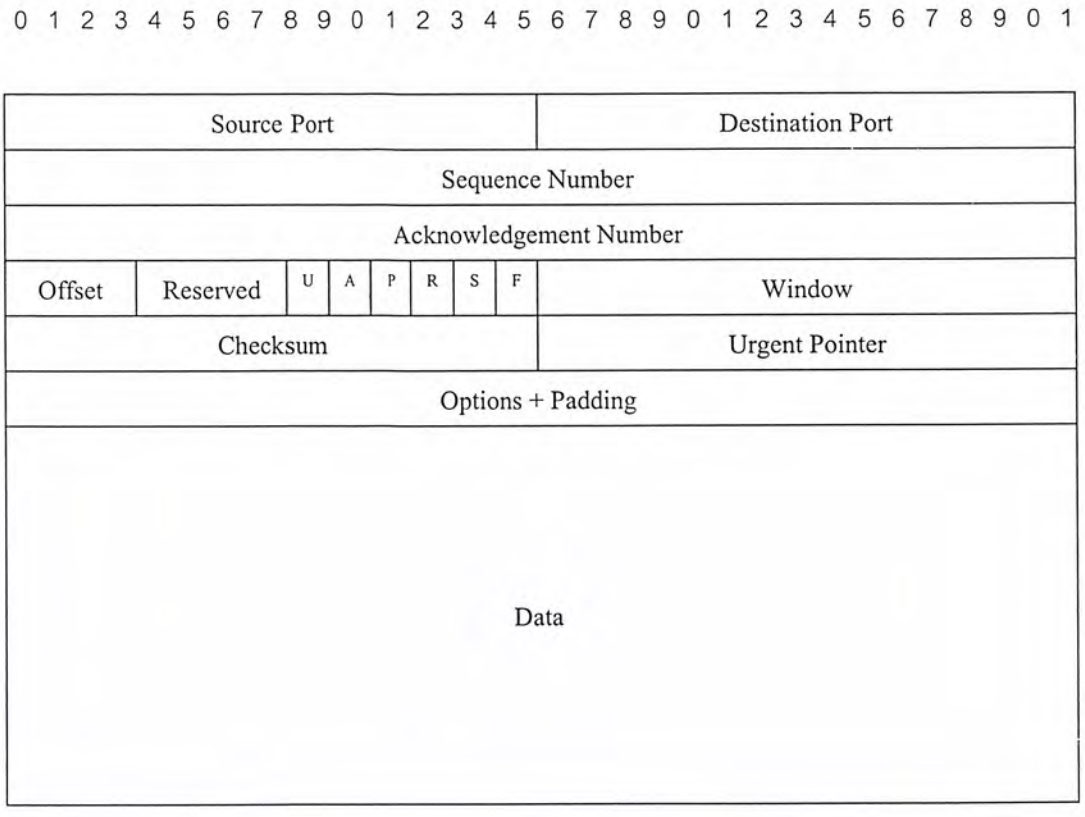
ดังนั้นโพรโตคอลทีซีพีจึงเป็นโพรโตคอลที่มีการรับส่งข้อมูลแบบ "Connection Oriented" ทำให้การทำงานของทีซีพีมีความน่าเชื่อถือมากขึ้น หน้าที่การทำงานของทีซีพีในการรับส่งข้อมูลมีหน้าที่หลัก 6 ข้อ คือ

1. ควบคุมการรับส่งข้อมูล (Basic Data Transfer)
2. ความน่าเชื่อถือในการรับส่งข้อมูล (Reliability)
3. ควบคุมการไหลของข้อมูล (Flow Control)
4. การทำมัลติเพล็ก (Multiplexing)
5. ควบคุมการเชื่อมต่อ (Connection)
6. ความปลอดภัยในการรับส่งข้อมูล (Security)

<u>ส่วนประกอบของทีซีพีเฮดเดอร์</u>

1. *Source Port* : เป็นหมายเลขพอร์ตของบริการที่เครื่องต้นทาง
2. *Destination Port* : เป็นหมายเลขพอร์ตของบริการเครื่องปลายทาง
3. *Sequence Number* : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลของเครื่องที่ต้องการขอส่งข้อมูล
4. *Acknowledgement Number* : เป็นหมายเลขที่บอกลำดับของการรับส่งข้อมูลที่ฝั่งรับข้อมูลปกติ ค่าของ Acknowledgement Number มีค่าเท่ากับ Sequence Number (ของอีกฝั่งหนึ่ง) + 1 เสมอ
5. *Data Offset* : เป็นตัวบอกค่าออฟเซตของข้อมูล เพราะทีซีพีนั้นไม่มีการกำหนดความยาวที่แน่นอนของข้อมูล จึงต้องมีออฟเซตเป็นตัวบอก
6. *Flag* : เป็นบิตที่บอกชนิดของข้อมูล ได้แก่
    - URG : Urgent Pointer Field Significant - แสดง Urgent Pointer
    - ACK : Acknowledgement Field Significant – แสดงการ Acknowledgement
    - PSH : Push Function
    - RST : Reset The Connection - แสดงเมื่อรีเซ็ทการเชื่อมต่อ
    - SYN : Synchronize Sequence Number - หมายเลขแพ็กเก็ตที่ส่งแบบซิงโครนัส
    - FIN : No more data from sender - แสดงว่าไม่มีข้อมูลที่ส่งจากผู้ส่งแล้ว
7. *Window* : เป็นเลขบอกจำนวนของอ็อกเต็ต (octet) ของข้อมูล จัดการในส่วนของ end-to-end flow control
8. *Checksum* : เป็นส่วนที่ตรวจสอบความถูกต้องของข้อมูล
9. *Urgent Pointer* : เป็นตัวชี้ตำแหน่งของ Urgent Data

10. *Option and Padding* : เป็นตัวบอกออปชันของโพรเซสที่ใช้ทีซีพี
11. *Data* : เนื้อข้อมูลที่ต้องการสื่อสาร มีขนาดได้ไม่ต่ำกว่า 5 32-บิตเวิร์ด (6 บิตแรกสงวนไว้และกำหนดให้เป็นศูนย์)



รูปที่ 3-4 แสดงแพ็กเก็ตทีซีพี

3.4 โพรโตคอลยูดีพี (UDP: User Datagram Protocol)

โพรโตคอลยูดีพีเป็นโพรโตคอลในการติดต่อสื่อสารในชั้นทรานสปอร์ต (Transport Layer) การทำงานคล้ายกับทีซีพีมาก คือจัดการเกี่ยวกับการสื่อสารระหว่างเครื่อง แต่เป็นแบบ Connectionless คือทั้งฝ่ายส่ง และฝ่ายรับไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน โดยไม่ต้องมีการแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโพรโตคอลทีซีพี และไม่มีการส่งสัญญาณตรวจสอบว่าข้อมูลถึงเครื่องปลายทางอย่างถูกต้องครบถ้วนในการส่งข้อมูลแต่ละครั้ง จึงไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล

<u>ส่วนประกอบของ UDP Frame</u>

1. *Source Port* : เป็นค่าตัวเลข 16 บิต บอกพอร์ตของบริการที่เครื่องต้นทาง
2. *Destination Port* : เป็นค่าตัวเลข 16 บิต บอกพอร์ตของบริการที่เครื่องปลายทาง
3. *Length* : เป็นค่าตัวเลข 16 บิต บอกความยาวของข้อมูล

4. *Checksum* : เป็นค่าตัวเลข 16 บิต ตรวจสอบความถูกต้องของข้อมูลที่ส่ง

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

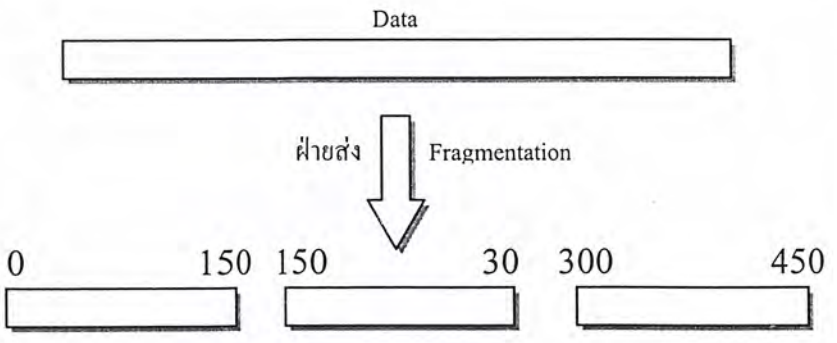
Source Port	Destination Port
Length	Checksum
Data	

รูปที่ 3-5 แสดงแพ็กเก็ตยูดีพี

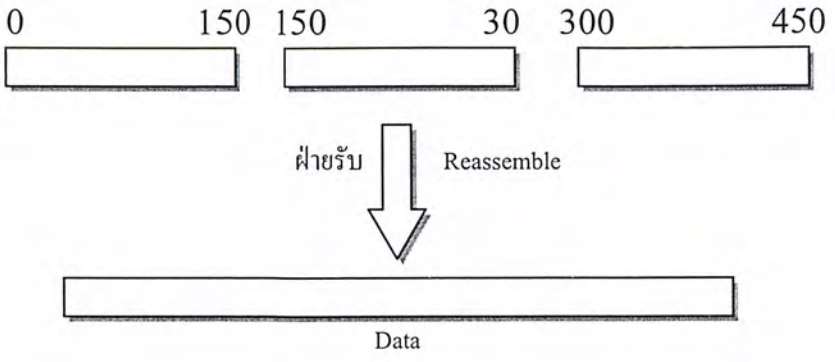
3.5 โพรโตคอลไอพี (IP: Internet Protocol)

โพรโตคอลไอพีเป็นโพรโตคอลที่จัดการเกี่ยวกับแอดเดรสของแต่ละแพ็กเก็ต เพื่อให้ส่งแพ็กเก็ตต่างๆ ไปยังเป้าหมายได้ถูกต้อง การทำงานของไอพีเป็นเพียงการส่งข้อมูลไปยังเครื่องเป้าหมายเท่านั้น ไม่มีการส่งสัญญาณขอบริการ หรือสัญญาณให้บริการระหว่างกันเหมือนทีซีพี เรียกว่าการเชื่อมต่อแบบ Connectionless ซึ่งระบบทั้งสองตั้งสมมติฐานว่าการเชื่อมต่อระหว่างกันไม่มีความผิดพลาดเกิดขึ้นแน่

เนื่องจากมาตรฐานในเครือข่ายมีหลากหลาย ขนาดของแพ็กเก็ตในแต่ละมาตรฐานจึงมีความแตกต่างกันออกไป ทำให้การส่งข้อมูลระหว่างอุปกรณ์ในเครือข่ายนั้นอาจมีการแบ่งข้อมูลออกเป็นแพ็กเก็ตย่อยๆ ในระหว่างการส่ง เรียกว่า การทำแฟร็กเมนเตชัน (Fragmentation) เช่น แพ็กเก็ตของ FDDI มีขนาด 4,500 ไบต์ หากเครื่องปลายอยู่ในเครือข่าย Ethernet ซึ่งมีขนาดของแพ็กเก็ตสูงสุดเพียง 1,500 ไบต์ ดังนั้นการส่งแพ็กเก็ตไปยังเครื่องปลายทางจึงต้องมีการแบ่งเป็นแพ็กเก็ตย่อย และเมื่อแพ็กเก็ตย่อยมาถึงเครื่องเป้าหมายก็จะมารวมกันเป็นแพ็กเก็ตเดิมที่มีขนาด 4,500 ไบต์อีกครั้ง เรียกการรวมกันนี้ว่า การรีแอสเซมเบิ้ล (Reassemble) ซึ่งทำให้ได้ข้อมูลเหมือนที่ส่งมาจากเครื่องต้นทาง



รูปที่ 3-6 แสดงการทำแฟร็กเมนเตชัน



รูปที่ 3-7 แสดงการรีแอสเซมเบิ้ล

ส่วนประกอบของแพ็กเก็ตไอพี

1. *version* : เป็นค่าตัวเลข 4 บิต บอกเวอร์ชันของมาตรฐานไอพีที่ใช้ โดยปกติมีค่าเป็น 4 ซึ่งหมายถึง IPv4
2. *Internet Header Length (IHL)* : เป็นตัวบอกความยาวเฮดเดอร์ของไอพี
3. *Type of Service* : เป็นส่วนที่บอกการทำงานของแพ็กเก็ตที่ส่งว่าทำหน้าที่อะไร มีทั้งหมด 8 บิต โดย

   Bit 0-2 : บอกรายละเอียดการทำงานของแพ็กเก็ตนั้นๆ

   111 - Network Control

   110 - Internetwork Control

   101 - CRITIC / ECP

   100 - Flash Override

   011 - Flash

   010 - Immediate

   001 - Priority

   000 - Routine

   Bit 3 : บอกถึงลักษณะของดีเลย์

   0 = Normal Delay - มีดีเลย์ปกติ

   1 = Low Delay - มีดีเลย์ต่ำ

   Bit 4 : บอกถึงประเภทของทรูพุต

   0 = Normal Throughput – มีทรูพุตปกติ

   1= High Throughput – มีทรูพุตสูง

   Bit 5 : บอกถึงประเภทของความน่าเชื่อถือ

0 = Normal Reliability - มีความน่าเชื่อถือพอประมาณ

1 = High Reliability - มีความน่าเชื่อถือสูง

Bit 6-7 : กันไว้ใช้ในอนาคต

4. *Total Length* : มีขนาด 16 บิต บอกถึงความยาวในดาต้าแกรมของไอพี
5. *Identification field* : เป็นตัวเลข 16 บิต เป็นค่าประจำตัวของไอพีนั้น โดยโฮสต์ที่ส่งเป็นผู้กำหนด และเพิ่มค่าขึ้นหนึ่งเมื่อมีการส่งดาต้าแกรมของไอพีใหม่ ซึ่งใช้ในการประกอบกลับ
6. *Flag* : เป็นตัวเลข 3 bit บอกลักษณะของแพ็กเก็ตว่ามีการแฟร็กเมนต์หรือไม่

   Bit 0 : สงวนไว้ ปกติเป็น 0

   Bit 1 : 0 = บอกว่าแพ็กเก็ตมีการแตกแพ็กเก็ตย่อย

   1 = บอกว่าแพ็กเก็ตไม่มีการแตกแพ็กเก็ตย่อย

   Bit 2 : 0 = บอกว่าแพ็กเก็ตนั้นเป็นแพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

   1 = บอกว่าแพ็กเก็ตนั้นยังไม่ใช่แพ็กเก็ตสุดท้ายที่ได้จากการแตกแพ็กเก็ตย่อย

7. *Fragment Offset* : เป็นค่าตัวเลข 13 บิต บอกออปเซตของแฟร็กเมนต์เมื่อเทียบในดาต้าแกรม
8. *Time To Live (TTL)* : เป็นตัวเลข 8 บิต บอกช่วงเวลาของแพ็กเก็ตที่ยังอยู่ในเครือข่ายได้ โดยกำหนดค่าเป็นจำนวนเราท์เตอร์สูงสุดที่ดาต้าแกรมผ่านได้ ซึ่งโดยทั่วไปทีค่าระหว่าง 32 ถึง 64 และลดค่าลงเรื่อยๆ เมื่อผ่านเราเตอร์ เพื่อเป็นการป้องกันแพ็กเก็ตล้นเครือข่าย
9. *Protocol* : เป็นตัวเลข 8 bit บอกถึงโพรโตคอลที่อยู่เหนือขึ้นไป ว่าเป็นโพรโตคอลระดับสูงกว่าประเภทใด
10. *Header Checksum* : เป็นค่าตัวเลข 32 บิต ใช้ตรวจสอบความถูกต้องของเฮดเดอร์
11. *Source Address*: เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องต้นทาง
12. *Destination Address* : เป็นค่าตัวเลข 32 บิต บอกถึงไอพีแอดเดรสของเครื่องปลายทาง

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

Ver	IHL	Type of Service	Total Length	
Identifier			Flags	Fragment
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options + Padding				
Data				

รูปที่ 3-8 แสดงแพ็กเก็ตไอพี

บทที่ 4

SNMP

SNMP(SIMPLE NETWORK MANAGEMENT PROTOCOL)

SNMP เป็นโปรโตคอลในระดับประยุกต์ (Application Layer) ที่กำหนดรูปแบบ และ กรรมวิธีการจัดการเครือข่าย โดยมีสถานีจัดการเครือข่ายส่วนกลางทำหน้าที่ดูแล ตรวจสอบ และควบคุมการทำงานของอุปกรณ์เครือข่าย

4.1 พื้นฐานการบริหารเครือข่าย

การบริหารเครือข่าย คือ การตรวจ ควบคุม และวางแผนการใช้ทรัพยากรระบบเพื่อให้เครือข่ายทำงานได้อย่างมีประสิทธิภาพ และ สามารถตรวจหาจุดบกพร่องที่เกิดขึ้นเพื่อแก้ไขปัญหาได้อย่างรวดเร็ว

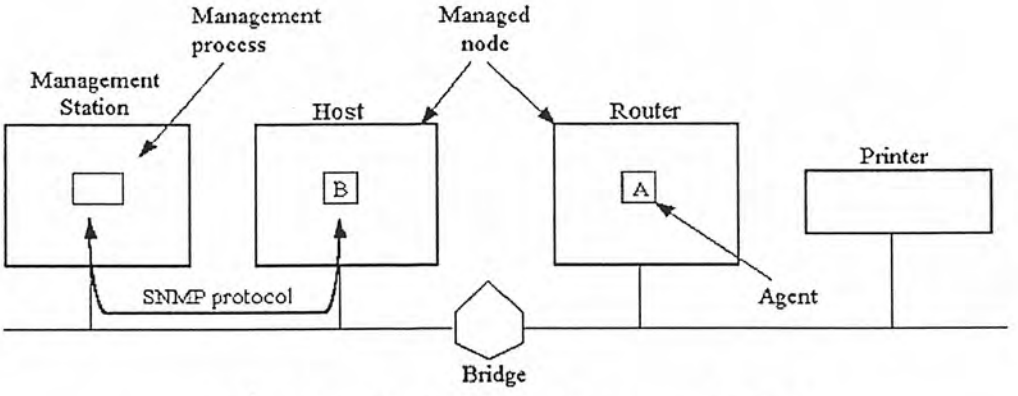
ในระบบเครือข่ายใดๆที่ เอสเอ็นเอ็มพี จัดการ จะต้องประกอบด้วย เอสเอ็นเอ็มพีโมเดล (SNMP Model) 4 ส่วนคือ

4.1.1. Managed nodes: อาจจะเป็น โฮสต์, เราท์เตอร์, ปรินท์เตอร์ หรืออุปกรณ์อื่นๆก็ได้ที่สามารถส่งข้อมูลสถานะของมันออกไปยังระบบได้ จะเป็นอุปกรณ์ ฮาร์ดแวร์ หรือ ซอฟท์แวร์ก็ได้ แต่ต้องมี เอสเอ็นเอ็มพีเอเจนต์ อยู่ในตัวด้วย

4.1.2. Management stations: คือ อุปกรณ์ใดๆที่มีฟังก์ชั่น ให้ตรวจสอบและปรับเปลี่ยนการทำงานได้ ซึ่งทำหน้าที่ ตรวจตรา และ ควบคุม Managed nodes

4.1.3. Management information.

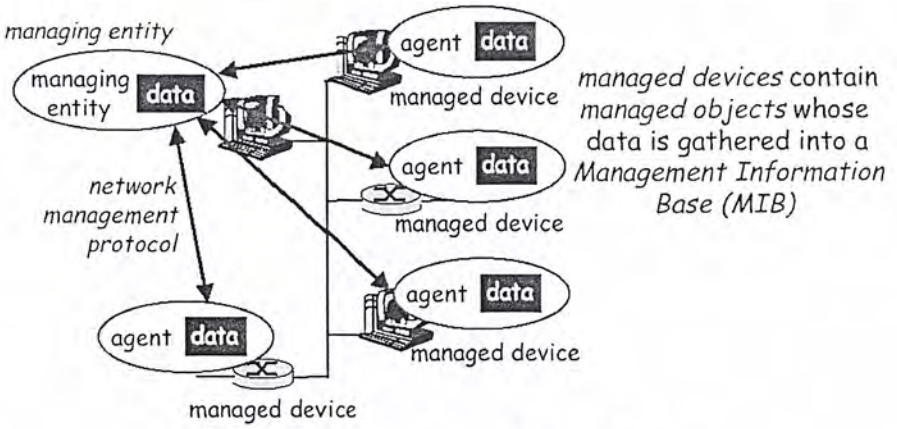
4.1.4. A management protocol.



รูปที่ 4-1 โมเดลส่วนประกอบการจัดการของเอสเอ็นเอ็มพี

4.2 เอสเอ็นเอ็มพีเอเจนต์

การจัดการเครือข่าย TCP/IP อาศัยรูปแบบการจัดการมาตรฐานตามข้อกำหนดของโปรโตคอล SNMP ซึ่งเป็นโปรโตคอลประยุกต์ที่กำหนดรูปแบบและกรรมวิธีการจัดการเครือข่าย โดยการทำงาน agent จะนำข้อมูลจากส่วน ซอฟท์แวร์ หรือ ฮาร์ดแวร์ เมื่อ Management stations ร้องขอข้อมูล และปรับเปลี่ยนการทำงานของ ซอฟท์แวร์ หรือ ฮาร์ดแวร์ เมื่อ Management stations สั่งงาน โดยมีการแจ้งยืนยันสิทธิในรูปในรหัสผ่านว่าManagement stations มีอำนาจหน้าที่ในการร้องขอและปรับค่า



รูปที่ 4-2 องค์ประกอบในระบบจัดการเครือข่าย

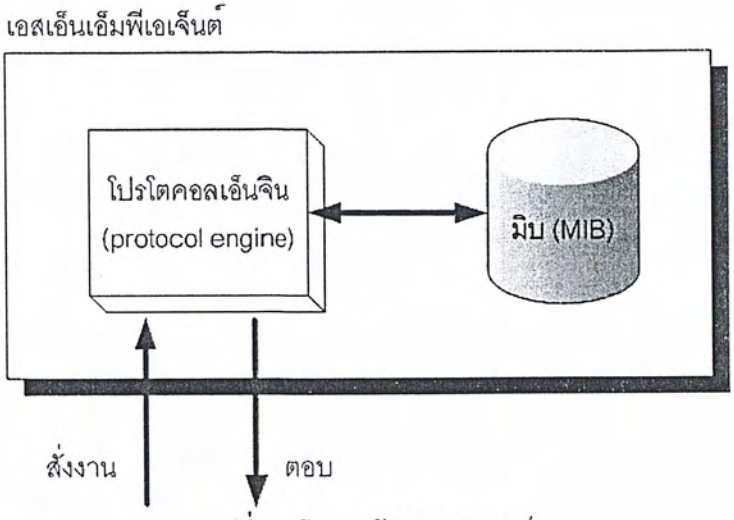
4.3 MIB (Management Information Base)

เอเจนต์จะประกอบด้วยส่วนสำคัญ 2 ส่วนคือ โปรโตคอลเอ็นจิน (Protocol engine) และฐานข้อมูลสารสนเทศการจัดการ (Management Information Base: MIB) โปรโตคอลเอ็นจินทำหน้าที่ประมวลคำสั่งที่มาจากNMS (Network Management Station) ซึ่งได้แก่ รับคำสั่ง ถอดรหัสคำสั่ง ทำงานตามคำสั่งและส่งผลตอบกลับ MIB เป็นส่วนที่เก็บตัวแปรและค่ากำหนดการทำงานประจำอุปกรณ์

ภายใน MIB ประกอบด้วยตัวแปรจำนวนมากที่เรียกโดยทั่วไปว่า อ๊อบเจ็กจัดการ (managed object) อ๊อบเจ็กในความหมายนี้เป็นชื่อที่ใช้เรียกตัวแปรและลักษณะเฉพาะของตัวแปรในMIBโดยไม่เกี่ยวข้องกับ เชิงวัตถุพิสัย (object oriented) แต่อย่างใด โดยจะพิจารณาอ๊อบเจ็กใน SNMP มีลักษณะเช่นเดียวกับเรคอร์ดในฐานข้อมูล

แต่ละอ๊อบเจ็ก จะมีชื่อเรียกเฉพาะเรียกว่า อ๊อบเจ็กไอเด็นติไฟเออร์ (Object Identifier) หรือเรียกโดยย่อว่า ไอเด็นติไฟเออร์ (Identifier) เพื่อใช้อ้างอิงถึงอ๊อบเจ็กนั้น

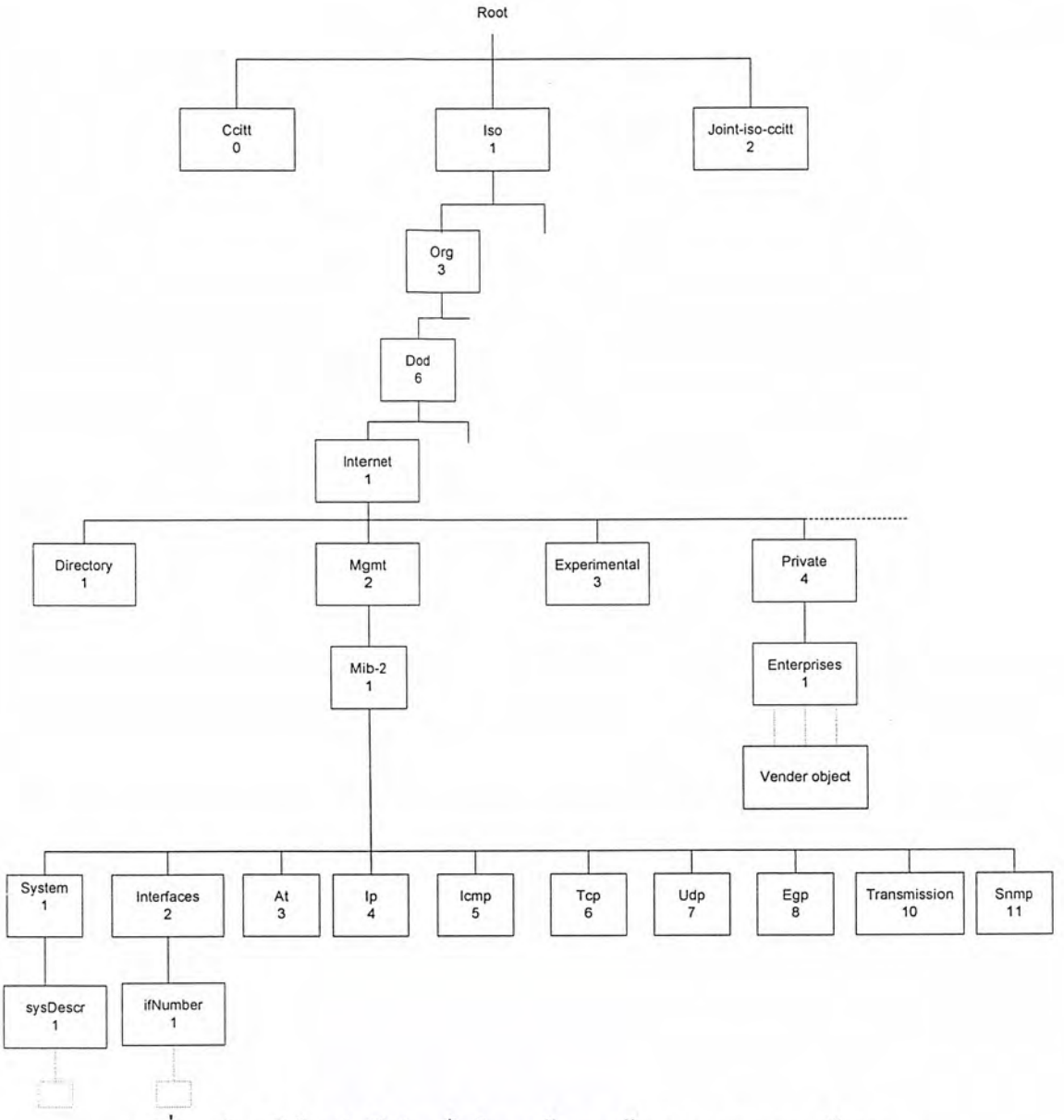
อ๊อบเจ็กทุกตัวมีนิยามที่กำหนด ชื่อ แบบข้อมูล สิทธิ์การเข้าถึง คำอธิบายลักษณะ และค่าข้อมูลการนิยามอ๊อบเจ็กมีกฎเกณฑ์ตามข้อกำหนด โครงสร้างฐานข้อมูลสารสนเทศ(Structure of Management Information: SMI)[RFC 1155]



รูปที่ 4-3 โครงสร้างของเอเจนต์

4.3.1 โครงสร้าง MIB

ข้อมูลประจำอุปกรณ์เครือข่ายชิ้นหนึ่งๆมีได้อย่างหลากหลาย อีกทั้งอุปกรณ์ต่างประเภทกันย่อมมีข้อมูลประจำอุปกรณ์แตกต่างกัน ดังนั้นการสอบถาม (อ่าน) หรือเปลี่ยนค่า (เขียน) ฐานข้อมูลจึงต้องมีรูปแบบมาตรฐานให้กับอุปกรณ์ทุกประเภท โครงสร้างต้นไม้แบบลำดับชั้นเป็นโครงสร้างที่เหมาะสมสำหรับใช้เป็นฐานข้อมูลเพื่อจัดเก็บตัวแปรเหล่านี้



รูปที่ 4-4 อ๊อบเจ็กไอเด็นติไฟเออร์ในโครงสร้างฐานข้อมูลสารสนเทศการจัดการ

ดังรูปที่ 4-4 แสดงข้อมูลหรืออ๊อบเจ็กของเอสเอ็นเอ็มพีในโครงสร้างต้นไม้ซึ่งนิยมเรียกว่า มิบทรี (MIB Tree) แต่ละโหนดซึ่งแทนอ๊อบเจ็กหนึ่งๆมีชื่อพร้อมทั้งตัวเลขฐานสิบกำกับประจำโหนดเพื่อใช้อ้างอิง ยกเว้นรากซึ่งไม่มีชื่อกำกับ

ลำดับชั้นแรกจะมีโหนดหลัก 3 โหนดซึ่งกำหนดกลุ่มองค์กร 3 กลุ่มคือ ITU-T(0),ISO(1) และ Joint-ISO-ITU-T(2) ภายใต้โหนด ISO มีโหนดลำดับที่ 3 คือ org(3) กำหนดองค์กรนานาชาติ และส่วนหนึ่งขององค์กรนี้คือ dod (6) หรือ Department of Defense และมี โหนด internet(1) เพื่อกำหนดกลุ่มการจัดการเครือข่ายใน internet

เมื่อต้องการอ้างอิงถึงโหนดใดในโครงสร้าง ให้เขียนหมายเลขจากรากไปตามเส้นทางถึงโหนดนั้นและคั่นด้วยจุด ลำดับตัวเลขนี้เรียกว่า อ๊อบเจ็กไอเด็นติไฟเออร์(Object Identifier) หรือ โอไอดี (OID)

ตัวอย่างเช่น 1.3.6.1.2.1.1 เป็นอ๊อบเจ็กไอเด็นติไฟเออร์โดยมีชื่อที่สมนัยกันคือ iso.org.dod.internet.mgmt.mib-2.system โหนดที่อยู่ภายใต้ 1.3.6.1.2.1 หรือในกลุ่ม mib-2 เป็นโหนดสำหรับใช้งานเอสเอ็นเอ็มพี แต่ละโหนจะมีโหนดย่อยเพื่ออ้างอิงถึงตัวแปรเช่น 1.3.6.1.2.1.1.1 คือตัวแปร sysDescr (System Description) ซึ่งเก็บคำอธิบายเกี่ยวกับอุปกรณ์นั้น

4.3.2 กลุ่มในมิบ

มิบภายใต้ internet มีกลุ่มย่อยทั้งหมด 6 กลุ่มคือ

- directory(1) สงวนไว้สำหรับใช้งานในอนาคต
- mgmt(2) กลุ่มมิบที่ใช้ในการจัดการภายใต้เอสเอ็นเอ็มพีรุ่น 1
- experimental(3) ใช้สำหรับการทดลอง
- private(4) สำหรับผู้ผลิตกำหนดตัวแปรเฉพาะอุปกรณ์
- security(5) ใช้ในระบบรักษาความปลอดภัย
- SNMPv2(6) ใช้ในเอสเอ็นเอ็มพีรุ่น2

ภายใต้กลุ่ม mib-2 บรรจุกลุ่มย่อยที่ใช้ในเอสเอ็มเอ็นพีซึ่งประกอบด้วย interface, at, ip และอื่นๆ ความหมายของแต่ละกลุ่มอธิบายไว้ในตารางที่ 4-1 แต่ละกลุ่มประกอบด้วยตัวแปรซึ่งมีแบบต่างๆกันไป

ลำดับ	ชื่อ	ความหมาย
1	system	ข้อมูลระบบ
2	interface	ข้อมูลอินเทอร์เฟสที่ใช้เชื่อมต่อ
3	at	ข้อมูลการแปลงแอดเดรส
4	ip	ข้อมูลไอพี
5	icmp	ข้อมูลไอซีเอ็มพี
6	tcp	ข้อมูลทีซีพี
7	udp	ข้อมูลยูดีพี
8	egp	ข้อมูลโปรโตคอลเกตเวย์ภายนอก
10	transmission	ข้อมูลสายสื่อสาร
11	SNMP	ข้อมูลเอสเอ็นเอ็มพี

ตารางที่ 4-1 กลุ่มย่อยภายใต้ mgnt

4.3.3 ชนิดของตัวแปรมิบ

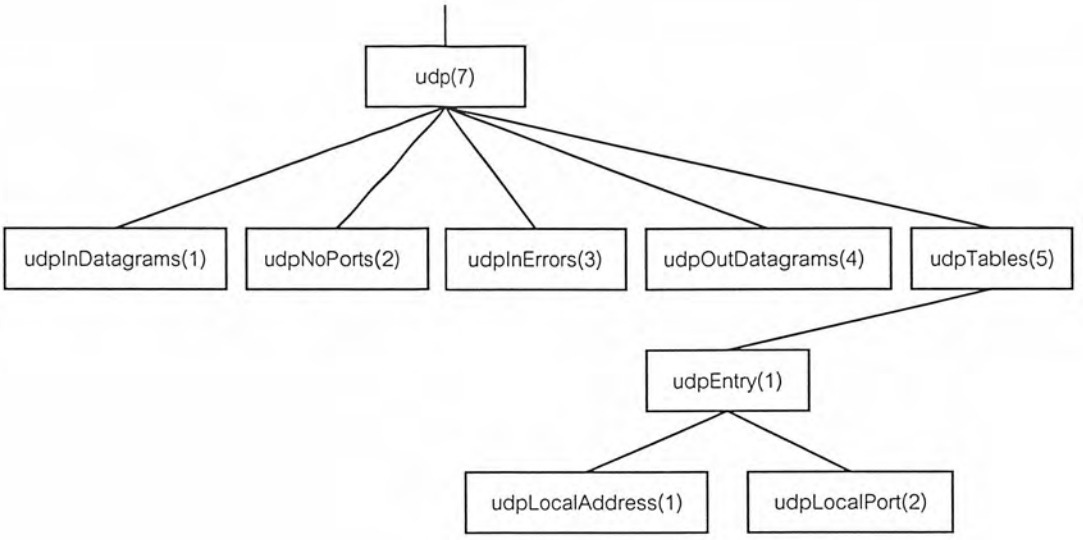
แต่ละตัวแปรในเอสเอ็นเอ็มพีมีข้อมูลประจำ แบบข้อมูลที่ใช้อยู่ในเอสเอ็นเอ็มพีมีดังนี้

- Integer: จำนวนเต็มเช่นหมายเลขพอร์ตของโปรโตคอลทีซีพีหรือยูดีพี มีค่าได้ตั้งแต่ 0 ถึง 65535
- OctedString: สายอักขระขนาดตั้งแต่ 0 อ๊อกเท็ต แต่ละอ๊อกเท็ตมีค่าตั้งแต่ 0 ถึง 255 ตัวอย่างแบบข้อมูลสายอักขระได้แก่ รหัสผ่าน
- DisplayString: สายอักขระตั้งแต่ 0 อ๊อกเท็ตแต่ละอ๊อกเท็ตต้องเป็นรหัสแอสกีเอ็นวีที ข้อมูลประเภทนี้มีความยาวตั้งแต่ 0 ถึง 255 ตัวอักษร
- Null: ใช้บอกว่าตัวแปรนั้นไม่มีค่าข้อมูลใดอยู่ เช่นเมื่อสอบถามข้อมูลด้วยคำสั่ง get หรือ get-next-request จะกำหนดแบบข้อมูลตัวแปรเท่ากับ null
- ObjectIdentifier: ชื่อตัวแปรในรูปแบบของการอ้างถึงแบบตัวเลขตามโครงสร้างมิบ
- IpAddress: สายอักขระ 4 อ๊อกเท็ต แต่ละอ๊อกเท็ตแทนไอพีแอดเดรสแต่ละตำแหน่ง
- PhysicalAddress: สายอักขระกำหนดฮาร์ดแวร์แอดเดรสเช่น อีเทอเน็ตแอดเดรสใช้สายอักขระ 6 อ๊อกเท็ต
- Counter: เลขจำนวนเต็มไม่คิดเครื่องหมาย มีค่าตั้งแต่ 0 ถึง 2 – 1 (4,294,967,295) การใช้ข้อมูล counter เป็นแบบเพิ่มค่าขึ้นอย่างเดียว เมื่อเพิ่มถึงค่ามากสุดจะกลับเป็น 0 ใหม่
- Gauge: เลขจำนวนเต็มไม่คิดเครื่องหมาย มีค่าตั้งแต่ 0 ถึง 2 -1 โดยสามารถเพิ่มหรือลดค่าได้ แต่เมื่อเพิ่มไปสูงสุดแล้วจะคงค่าไว้จนกว่าจะถูกปรับค่ากลับมาเป็น 0 อีกครั้ง ตัวอย่างตัวแปรที่ใช้ค่านี้เช่น จำนวนการเชื่อมโยงทีซีพีที่อนุญาตให้มีได้
- TimeTicks: เลขจำนวนเต็มใช้นับเวลาให้หน่วยเศษหนึ่งส่วนร้อยของวินาที เช่น เวลานับตั้งที่ระบบเริ่มทำงาน
- Sequence: โครงสร้างแบบเร็คคอร์ด หรือคล้ายกับแบบข้อมูล struct ในภาษาซี
- Sequence of: โครงสร้างแบบตารางหรือมองในรูปของอาร์เรย์ เช่นตารางเลือกเส้นทางของไอพี

4.3.4 ตัวอย่างแบบข้อมูลอาร์เรย์

แบบข้อมูล sequence of ใช้กำหนดข้อมูลแบบเวกเตอร์ซึ่งสมาชิกทั้งหมดภายในมีข้อมูลแบบเดียวกัน หากสมาชิกมีแบบข้อมูลเบื้องต้น เช่น แบบจำนวนเต็มก็จะได้เวกเตอร์แบบมิติเดียว หรือหากสมาชิกมีข้อมูลแบบ sequence ก็จะได้อาร์เรย์ 2 มิติ

ตัวอย่างของตัวแปรโครงสร้างอาร์เรย์ในมิบมีอยู่หลายตัวแปร แต่จะยกตัวอย่างเฉพาะตัวแปรที่มีขนาดเล็กเพื่อที่จะทำความเข้าใจได้ง่ายได้แก่ udpTable ซึ่งสังกัดอยู่ภายใต้กลุ่ม udp ตามโครงสร้างข้อมูลดังรูปที่ 4-5



รูปที่ 4-5 กลุ่ม udp

udpTable มีแบบข้อมูล sequence of และภายใต้ udpTable มี udpEntry ซึ่งมีแบบข้อมูล sequence โดยประกอบด้วย udpLocalAddress และ udpLocalPort

udpLocalAddress มีแบบข้อมูล IpAddress ใช้กำหนดไอพีแอดเดรสที่รอให้บริการส่วนของ udpLocalPort กำหนดหมายเลขพอร์ตดังนั้น udpTable จึงมีโครงสร้างเป็นอาร์เรย์ 2 มิติหรือเขียนด้วยตาราง
ดังรูปที่ 4-6

	udpLocalAddress	udpLocalPort
udpEntry	(IpAddress)	(Integer)
udpEntry	...	...
udpEntry	...	...
udpEntry	...	...
udpEntry	...	...

รูปที่ 4-6 udpTable ในรูปอาร์เรย์สองมิติ (ตาราง)

4.4 การแทนข้อมูลด้วย ASN.1

ไอเอสโอและซีซีไอทีทีกำหนดวิธีการนิยามชนิดของตัวแปรโดยใช้ไวยากรณ์ ASN.1 (Abstract Syntax Notation One) ซึ่งเป็นเป็นเสมือนภาษาอธิบายแบบข้อมูลที่ไม่ขึ้นกับฮาร์ดแวร์ต้นกำเนิดของ

ASN.1 นำมาใช้นิยามชุดโปรโตคอลของไอเอสโอ แต่สำหรับเอสเอ็นเอ็มพีจำใช้เพียงไวยากรณ์เพียงบางส่วนของ ASN.1 เท่านั้น

4.4.1 Module Definition

การใช้ ASN.1 ช่วยให้ผู้นำมาตรฐานไปใช้เข้าใจถึงสิ่งที่ผู้สร้างมาตรฐานกำหนดไว้โดยไม่เกิดความกำกวมในเรื่องของความหมายและการแทนข้อมูล เช่นการกำหนดตัวแปรชนิด integer จะต้องกำหนดให้แน่นอนว่ามีค่าอยู่ในช่วงใด เพราะว่าคอมพิวเตอร์ต่างชนิดกันอาจมีความแตกต่างกันในการแทนข้อมูลและช่วงของตัวแปรที่แตกต่างกัน มิบอ๊อบเจ็กในเอสเอ็นเอ็มพีมีรูปแบบที่กำหนดด้วย ASN.1 ตัวอย่างต่อไปนี้เป็นนิยามของอ๊อบเจ็ก sysContact

```
Syscontact      OBJECT-TYPE
                SYNAX   DisplayString (size (0..255))
                ACCESS  read-write
                STATUS  mandatory
                DESCRIPTION
                        "The textual identification of the contact person for this managed node,
                        together with information on how to contact this person."
                ::= (system 4)
```

นิยามข้างต้นมีความหมายดังนี้

- SYNTAX : กำหนดแบบข้อมูลของตัวแปรเช่นจำนวนเต็มหรืออักขระ ในที่นี้คือ DisplayString
- ACCESS : แบบการเข้าใช้ซึ่งอาจเป็น read-only(อ่านอย่างเดียว), read-write(อ่านเขียนได้),write-only(เขียนอย่างเดียว) หรือ not-accessible (ห้ามเข้าถึง) เป็นต้น
- STATUS : กำหนดสถานะของตัวแปรว่าจำเป็นต้องมีตัวแปรนี้หรือไม่ ค่าที่เป็นไปได้ เช่น mandatory(จำเป็น),optional(ออพชั่นซึ่งมีหรือไม่ก็ได้), depreiate (จำเป็นต้องมีแต่อาจยกเลิกในรุ่นถัดไป), obsoleted (ไม่จำเป็นเนื่องจากยกเลิกไม่ใช้แล้ว)
- DESCRIPTION : ข้อความอธิบายตัวแปร
- บรรทัดสุดท้ายของนิยามกำหนดว่าตัวแปร sysContact จะเชื่อมกับโครงสร้างต้นไม้ต่อจากโหนด system และมีค่าเท่ากับ 4 ซึ่งแสดงถึงไอเด็นติไฟเออร์ประจำ sysContact คือ iso.org.dod.internet.mgmt.mib-2.system.4 หรือ 1.3.6.1.2.1.1.4

ASN.1 คือภาษาที่สามารถใช้ในการกำหนดโครงสร้างข้อมูล โดยโครงสร้างที่ถูกกำหนดนี้จะอยู่ในรูปแบบที่ชื่อว่าโมดูล โดยชื่อของโมดูลสามารถที่จะถูกใช้อ้างอิงถึงโครงสร้างได้ ตัวอย่างเช่น ชื่อโมดูล

สามารถที่จะถูกใช้เสมือน ชื่อ abstract syntax ซึ่ง application สามารถที่จะส่งชื่อไปเพื่อกำหนด abstract syntax ของ APDUs ที่ application ต้องการจะเปลี่ยน

โดยจะมีรูปแบบพื้นฐานของโมดูลเป็นดังนี้

<modulereference> DEFINITIONS ::=

BEGIN

EXPORTS

IMPORTS

Assignmentlist

End

เมื่อ Modulereference :: ชื่อของโมดูลนั้นๆ

EXPORTS :: เป็นตัวชี้ว่าโมดูลนี้อาจจะถูกอิมพอร์ตจากโมดูลอื่น

IMPORTS :: เป็นตัวบอกว่ารูปแบบและค่าจากโมดูลอื่นได้ถูกอิมพอร์ตมายังโมดูลนี้

Assignment :: จะประกอบด้วย ชนิดของ assignment ค่าของ assignment และการกำหนดค่าของมาโคร

4.4.2 Macro Definitions

เป็นการอนญาติให้ผู้ใช้สามารถที่จำขยาย syntax ของ ASN.1 โดยการกำหนดรูปแบบใหม่และค่าของรูปแบบนั้นๆ โดยจำมีระดับของกำหนดคือ

Macro definition: เป็นการกำหนดกฎของ macro instance โดยจะกำหนด syntax ของเซตของรูปแบบที่เกี่ยวข้อง

Macro instance: ถูกสร้างจาก การกำหนด macro instance

Macro instance value: การกำหนดค่าของ macro instance

การกำหนดมาโครจะมีรูปแบบดังนี้

<macroname> MACRO ::=

BEGIN

TYPE NOTATION ::= <new-type-syntax>

VALUE NOTATION ::= <new-value-syntax>

<supporting-productions>

END

ตัวอย่างการกำหนดมาโคร

```
OBJECT-TYPE MACRO ::=
    BEGIN
        TYPE NOTATION ::=  "Syntax"      type (TYPE ObjectSyntax)
                           "ACCESS"      Access
                           "STATUS"      Status
        VALUE NOTATION ::= value (VALUE ObjectName)
        Access   ::= "read-only"|"read-write"|"write-only"|"not-accessible"
    Status   ::= "mandatory"|"optional"|"obsolete"
    END
```

4.4.3 Defining Tables

การกำหนดฐานข้อมูลของอ๊อบเจ็กนั้นจะมีการเกี่ยวพันกับการใช้ sequence และ sequence-of โดยทางที่ดีที่สุดที่จะอธิบายนั้นคงจะเป็นการยกตัวอย่างให้ดู โดยที่เราจะพิจารณาที่ tcpConnTable ที่มีอ๊อบเจ็กไอเด็นติไฟเออร์เป็น 1.3.6.1.2.1.6.13 ตาราง TCP connection จะประกอบด้วย ข้อมูลเกี่ยวกับรูปแบบการติดต่อของ TCP ไอพีแอดเดรสและพอร์ต ของเครื่องที่ติดต่อ จาก code ตัวอย่างด้านล่าง สามารถบอกเราได้ว่า มีการประกาศตารางซึ่งประกอบด้วย TcpConnEntry ซึ่ง TcpConnEntry จะมี tcpConnState(รูปแบบการติดต่อของ TCP), tcpConnLocalAddress (ไอพีแอดเดรสของเครื่องที่ติดต่อ), tcpConnLocalPort(พอร์ตของเครื่องที่ติดต่อ), tcpConnRemAddress(ไอพีแอดเดรสของเครื่องที่ติดต่ออีกเครื่องหนึ่ง), tcpConnRemPort(พอร์ตของเครื่องที่ติดต่ออีกเครื่องหนึ่ง)

```
tcpConnTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF TcpConnEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
          "A table containing TCP connection-specific
          information."
        ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
        SYNTAX  TcpConnEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
```

```
                    "Information about a particular current TCP
                     connection.  An object of this type is transient,
                     in that it ceases to exist when (or soon after)
                     the connection makes the transition to the CLOSED
                     state."
INDEX   { tcpConnLocalAddress,
          tcpConnLocalPort,
          tcpConnRemAddress,
          tcpConnRemPort }
::= { tcpConnTable 1 }
   TcpConnEntry ::=
        SEQUENCE {tcpConnState INTEGER,
          tcpConnLocalAddress IpAddress,
          tcpConnLocalPort INTEGER (0..65535),
                    tcpConnRemAddress IpAddress,
          tcpConnRemPort INTEGER (0..65535)}

  tcpConnState OBJECT-TYPE
        SYNTAX  INTEGER {closed(1),
                         listen(2),
                         synSent(3),
                         synReceived(4),
                    established(5),
                         finWait1(6),
                         finWait2(7),
                         closeWait(8),
                         lastAck(9),
                    closing(10),
                         timeWait(11),
                         deleteTCB(12)}
        ACCESS  read-write
      STATUS  mandatory
      DESCRIPTION
                    "The state of this TCP connection.
```

```
                The only value which may be set by a management
        station is deleteTCB(12).  Accordingly, it is
                appropriate for an agent to return a `badValue'
        response if a management station attempts to set
                this object to any other value.
                If a management station sets this object to the
        value deleteTCB(12), then this has the effect of
                deleting the TCB (as defined in RFC 793) of the
        corresponding connection on the managed node,
                resulting in immediate termination of the
        connection.

                As an implementation-specific option, a RST
                    segment may be sent from the managed node to the
        other TCP endpoint (note however that RST segments
        are not sent reliably)."
    ::= { tcpConnEntry 1 }

tcpConnLocalAddress OBJECT-TYPE
        SYNTAX  IpAddress
        ACCESS  read-only
        STATUS  mandatory
       DESCRIPTION
                "The local IP address for this TCP connection.  In
      the case of a connection in the listen state which
      is willing to accept connections for any IP
     interface associated with the node, the value
     0.0.0.0 is used."
    ::= { tcpConnEntry 2 }

tcpConnLocalPort OBJECT-TYPE
          SYNTAX  INTEGER (0..65535)
          ACCESS  read-only
          STATUS  mandatory
```

DESCRIPTION

"The local port number for this TCP connection."

::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE

SYNTAX IpAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The remote IP address for this TCP connection."

::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE

SYNTAX INTEGER (0..65535)

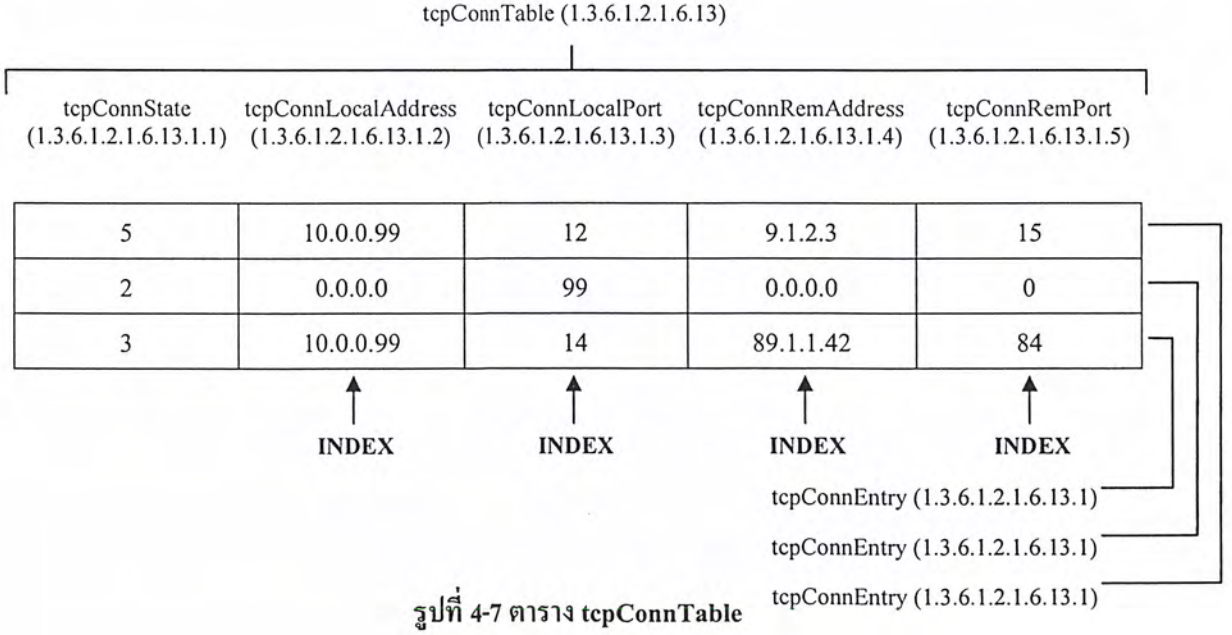
ACCESS read-only

STATUS mandatory

DESCRIPTION

"The remote port number for this TCP connection."

::= { tcpConnEntry 5 }



รูปที่ 4-7 ตาราง tcpConnTable

4.5 Communities and Community Names

การบริหารเครือข่ายจะถือได้ว่าเป็นการทำงานในลักษณะระบบกระจาย (Distributed application) รูปแบบหนึ่ง ซึ่งจะเห็นได้ว่าความสัมพันธ์ระหว่างสถานีจัดการเครือข่าย (Network Management Station : NMS) กับเอเจนต์ (Agent) จะเป็นในรูปแบบ many-to-many คือ สถานีจัดการเครือข่ายจะบริหารจัดเอเจนต์ได้หลายเครื่อง และในขณะเดียวกันเอเจนต์ก็สามารถถูกบริหารควบคุมจากสถานีจัดการเครือข่ายหลายเครื่องเช่นกัน

จากความสัมพันธ์ดังกล่าวจึงจำเป็นที่เอเจนต์แต่ละเครื่องจำเป็นที่จะต้องมีมาตรการควบคุมความปลอดภัยในการใช้งานฐานข้อมูลสารสนเทศการจัดการ (Management Information Base : MIB) ของตนเองโดยจะมีมุมมองทางด้านความปลอดภัย 3 ประการได้แก่

- การพิสูจน์ตัวตน (Authentication service) จะเป็นการจำกัดให้เฉพาะสถานีจัดการเครือข่ายที่จะเข้ามาบริการควบคุม
- นโยบายการเข้าถึง (Access policy) จะมีการกำหนดระดับการอนุญาตการเข้าถึงฐานข้อมูลสารสนเทศการจัดการให้แต่ละสถานีจัดการเครือข่ายไม่เท่ากันในแต่ละเครื่อง
- การให้บริการ Proxy (Proxy service) เอเจนต์อาจทำหน้าที่เป็น Proxy ให้กับเอเจนต์เครื่องอื่นซึ่งจะรวมถึงการพิสูจน์ตัวตนและนโยบายการเข้าถึงของเอเจนต์ตัวอื่นที่อยู่ในระบบ Proxy

เอสเอ็นเอ็มพี (SNMP) ได้มีการกำหนดการทำงานเพื่อสนับสนุนมุมมองทางด้านทางปลอดภัยดังกล่าวในรูปแบบของ SNMP Community โดยการทำงานคือ เอเจนต์แต่ละเครื่องจะมีการสร้าง Community name เพื่อกำหนดให้กับสถานีจัดการเครือข่าย โดนในหนึ่ง Community name จะสามารถมีสถานีจัดการเครือข่ายมากกว่าหนึ่งตัว

เนื่องจาก Community name จะถูกกำหนดในแต่ละเอเจนต์จึงอาจเป็นไปได้ว่ามีการตั้งชื่อ Community name ซ้ำกันในแต่ละเอเจนต์ แต่สถานีจัดการเครือข่ายจะสามารถแยกความแต่ต่างของ Community ที่มีชื่อซ้ำกันเหล่านี้เองได้ถ้าอยู่ในคนละเอเจนต์กัน ดังนั้นจึงจำเป็นที่ว่าสถานีจัดการเครือข่ายจะต้องเก็บข้อมูลของ Community name และข้อมูลที่เกี่ยวข้องของแต่ละเอเจนต์เพื่อใช้ในการบริหารควบคุม

4.5.1 การพิสูจน์ตัวตน (Authentication service)

การพิสูจน์ตัวตนมีไว้เพื่อให้แน่ใจการติดต่อนั้นเป็นของแท้ ในกรณีของเอสเอ็นเอ็มพีการพิสูจน์ตัวตนจะมีไว้เพื่อให้แน่ใจว่า message ที่ได้รับมานั้นเป็นข้อความที่แท้จริง โดยในทุกๆ message ของเอสเอ็นเอ็มพีจะมีการระบุ Community name ซึ่งจะมีหน้าที่เสมือนกับรหัสผ่าน (Password) ในการพิสูจน์ตัวตน นอกจากนั้นยังอาจจะมีการเข้ารหัส (Encryption) เพื่อเพิ่มความปลอดภัยในการพิสูจน์ตัวตนมากยิ่งขึ้น

4.5.2 นโยบายการเข้าถึง (Access policy)

การควบคุมการเข้าถึงในเอสเอ็นเอ็มพีจะประกอบด้วย 2 องค์ประกอบหลักที่เกี่ยวข้องคือ

- SNMP MIB view: คือกลุ่มของอ็อบเจ็กในฐานข้อมูลสารสนเทศการจัดการที่ตั้งขึ้นโดยในแต่ละกลุ่มอาจจะประกอบด้วยหลาย Sub tree ในฐานข้อมูลสารสนเทศการจัดการได้
- SNMP access mode: คือรูปแบบของการเข้าถึงได้แก่ READ-ONLY และ READ-WRITE

ในเอเจนต์จะมีการกำหนด Access mode ให้แต่ละ MIB view ซึ่ง Access mode จะมีผลกับทุกๆ Object ที่อยู่ในกลุ่มของ MIB view โดยทั้ง Access mode และ MIB view จะถูกรียกรวมกันว่า SNMP community profile ซึ่งจะถูกกำหนดในแต่ละ Community

ใน Access mode ของเอสเอ็นเอ็มพีจะมีการประนีประนอมต่อรองระดับการเข้าถึงกับระดับของการเข้าถึงของฐานข้อมูลสารสนเทศการจัดการ (MIB Access Category) ดังตารางที่ 2 (Access mode ของเอสเอ็นเอ็มพีกับระดับของการเข้าถึงของฐานข้อมูลสารสนเทศการจัดการเป็นคนละส่วนกัน) และจะเรียกรวม SNMP community และ SNMP community profile ว่า SNMP access policy

MIB Access Category	**SNMP Access Mode**	
	READ-ONLY	READ-WRITE
read-only	สามารถใช้คำสั่ง get และ trap ได้	
read-write	สามารถใช้คำสั่ง get และ trap ได้	สามารถใช้คำสั่ง get, set และ trap ได้
write-only	สามารถใช้คำสั่ง get และ trap แต่ต้องเป็นค่าที่เป็น implementation-specific	สามารถใช้คำสั่ง get, set และ trap ได้แต่ต้องเป็นค่าที่เป็น implementation-specific สำหรับคำสั่ง get และ trap
not accessible	ไม่สามารถเข้าถึงได้	

ตารางที่ 4-2 ความสัมพันธ์ระหว่าง MIB Access Category และ SNMP Access Mode

4.5.3 การให้บริการ Proxy (Proxy service)

แนวคิดของ Community จะสามารถสนับสนุนการทำ Proxy ได้ โดยเอเจนต์จะสามารถทำหน้าที่เป็นตัวแทนในการติดต่อกับสถานีจัดการเครือข่ายให้กับเอเจนต์หรืออุปกรณ์อื่นได้ ในกรณีที่อุปกรณ์เครื่องนั้นไม่สนับสนุน TCP/IP และเอสเอ็นเอ็มพีหรือในกรณีที่ต้องการลดปริมาณการติดต่อระหว่างอุปกรณ์นั้นกับสถานีจัดการเครือข่าย โดยเอเจนต์ที่เป็น Proxy จะสนับสนุน SNMP access policy ของเอเจนต์ที่อยู่ในระบบ Proxy ด้วย

4.6 การอ้างอิงถึงค่าในอ็อบเจ็ก (Instance Identification)

ในการอ้างอิงถึงค่าของอ็อบเจ็กในฐานข้อมูลสารสนเทศการจัดการของเอสเอ็นเอ็มพีนั้นจะอาศัยการอ้างอิงของอินสแตนท์ไอเด็นติไฟเออร์ (Instance Identifier)

สำหรับการอ้างถึงค่าของอ็อบเจ็กแบบปกติ (Simple Object Value) โดยทั่วไปจะใช้อินสแตนท์ไอเด็นติไฟเออร์ที่ประกอบไปด้วยค่าของ อ็อบเจ็กไอเด็นติไฟเออร์ (Object Identifier) แล้วปิดท้ายด้วย 0 คือจะอยู่ในรูปแบบของ

Instance Identifier = Object Identifier.0

เช่นการอ้างถึงค่าในอ็อบเจ็ก sysDescr ด้วยค่าอินสแตนท์ไอเด็นติไฟเออร์คือ 1.3.6.1.2.1.1.1.0 ซึ่งก็คือจะประกอบด้วยค่าอ็อบเจ็กไอเด็นติไฟเออร์ของอ็อบเจ็ก sysDescr คือ 1.3.6.1.2.1.1.1 แล้วต่อท้ายด้วย 0 นอกจากนั้นยังอาจเขียนในรูปของชื่อได้คือ iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 หรือเขียนสั้นๆเพียง sysDescr.0 ก็ได้

สำหรับการอ้างถึงค่าของอ็อบเจ็กในรูปแบบตาราง (Sequence of Object Value) นั้น เอสเอ็นเอ็มพีไม่อนุญาตให้อ้างอิงทั้งตารางได้ในคราวเดียว การอ้างอิงจะต้องทำที่อ็อบเจ็กที่เป็นโหนดปลายเท่านั้น (Leaf object) หรือต้องเจาะจงถึงค่าในตารางเลยนั้นเอง ดังนั้นจึงต้องอาศัยเทคนิคเฉพาะในการอ้างอิงดังกล่าวซึ่งจะมีอยู่ 2 เทคนิคได้แก่ Serial-access technique และ Random-access technique สำหรับเทคนิคของ Random-access technique จะอธิบายไว้ในหัวข้อนี้ส่วนเทคนิคของ Serial-access technique จะอธิบายไว้ในหัวข้อของคำสั่ง GetNextRequest ต่อไป

Random-access technique นั้นจะอาศัยแนวคิดที่ว่า สิ่งแยกความแตกต่างของแต่ละคอลัมน์ในตารางก็คืออ็อบเจ็กไอเด็นติไฟเออร์ และสิ่งแยกความแตกต่างของแต่ละแถวในตารางก็คือค่าในอินเดกซ์อ็อบเจ็ก (Value of INDEX Object)

จากตัวอย่างตาราง tcpConnTable ดังรูปที่ 6 ซึ่งมีแบบข้อมูล sequence of และภายใต้ตาราง tcpConnTable มี tcpConnEntry ซึ่งมีแบบข้อมูลเป็น sequence โดยประกอบด้วย tcpConnState, tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress และ tcpConnRemPort และจะมีอินเดกซ์อ็อบเจ็ก (INDEX Object) คือ tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress และ tcpConnRemPort

การอ้างอิงถึงค่าของอ็อบเจ็กในรูปแบบตารางตาม Random-access technique นี้ จะใช้อินสแตนท์ไอเด็นติไฟเออร์ที่ประกอบไปด้วยการรวมกันของ อ็อบเจ็กไอเด็นติไฟเออร์ (Object Identifier) และ ค่าในอินเดกซ์
อ็อบเจ็ก คือจะอยู่ในรูปแบบของ

Instance Identifier = Object Identifier.(ค่าในอินเดกซ์อ็อบเจ็ก1).(ค่าในอินเดกซ์อ็อบเจ็ก2). (ค่าในอินเดกซ์อ็อบเจ็ก3).....(ค่าในอินเดกซ์อ็อบเจ็ก N)

ดังนั้นจากตัวอย่างตาราง tcpConnTable จะใช้อินสแตนท์ไอเด็นติไฟเออร์ตามรูปแบบดังนี้คือ

x.i.(tcpConnLocalAddress).(tcpConnLocalPort).(tcpConnRemAddress).(tcpConnRemPort)

เมื่อ x = 1.3.6.1.2.1.6.13.1 ซึ่งคืออ็อบเจ็กไอเด็นติไฟเออร์ของ tcpConnEntry

i = เป็นหมายเลยที่ใช้ระบุเพื่อแยกความแตกต่างของแต่ละคอลัมน์ เช่น ถ้าต้องการ ระบุถึงคอลัมน์ tcpConnLocalAddress ก็ให้ค่าของอ็อกเท็ต (Octet) i เป็น 1 แต่ถ้าต้องการระบุถึงคอลัมน์ tcpConnLocalPort ก็ให้ค่าของอ็อกเท็ต (Octet) i เป็น 2 เป็นต้น ตามตัวเลขตัวสุดท้ายในอ็อบเจ็กไอเด็นติไฟเออร์ของอ็อบเจ็กแต่ละตัว

(name) = ค่าที่อยู่ในอินเดกซ์อ็อบเจ็ก เช่น (tcpConnLocalAddress) ก็จะหมายถึงค่าที่อยู่ในอ็อบเจ็ก tcpConnLocalAddress นั้นเอง

สำหรับอินสแตนท์ไอเด็นติไฟเออร์ของทั้งตาราง tcpConnTable ที่อาศัยค่าในอินเดกซ์อ็อบเจ็กจากตัวอย่างในรูป 7.1 จะแสดงได้ดังรูปที่ 4-8

tcpConnState (1.3.6.1.2.1.6.13.1.1)	tcpConnLocalAddress (1.3.6.1.2.1.6.13.1.2)	tcpConnLocalPort (1.3.6.1.2.1.6.13.1.3)	tcpConnRemAddress (1.3.6.1.2.1.6.13.1.4)	tcpConnRemPort (1.3.6.1.2.1.6.13.1.5)
x.1.10.0.0.99.12.9.1.2.3.15	x.2.10.0.0.99.12.9.1.2.3.15	x.3.10.0.0.99.12.9.1.2.3.15	x.4.10.0.0.99.12.9.1.2.3.15	x.5.10.0.0.99.12.9.1.2.3.15
x.1.0.0.0.0.99.0.0.0.0.0	x.2.0.0.0.0.99.0.0.0.0.0	x.3.0.0.0.0.99.0.0.0.0.0	x.4.0.0.0.0.99.0.0.0.0.0	x.5.0.0.0.0.99.0.0.0.0.0
x.1.10.0.0.99.14.89.1.1.24.84	x.2.10.0.0.99.14.89.1.1.42.84	x.3.10.0.0.99.14.89.1.1.42.84	x.4.10.0.0.99.14.89.1.1.42.84	x.5.10.0.0.99.14.89.1.1.42.84

รูปที่ 4-8 อินสแตนท์ไอเด็นติไฟเออร์ของทั้งตาราง tcpConnTable เมื่อ x = 1.3.6.1.2.1.6.13.1 ซึ่งคืออ็อบเจ็กไอเด็นติไฟเออร์ของ tcpConnEntry

เนื่องจากชนิดของ Object มีมากมายดังนั้นจึงมีการตั้งกฎเกณฑ์ในการนำเอาค่าในอินเดกซ์อ็อบเจ็กที่จะเอามาประกอบเป็นอินสแตนท์ไอเด็นติไฟเออร์ดังนี้

- *ค่าของ Integer (Integer-valued):* ให้นำค่าของ Integer ดังกล่าวมาประกอบรวมเป็น 1 อ็อกเท็ต (Octet) ของอินสแตนท์ไอเด็นติไฟเออร์ได้ทันที

- *ค่าของ String ที่มีความยาวคงที่ (String-valued, fixed-length):* ให้นำค่าของตัวอักษรในแต่ละอ็อกเท็ตมาแปลงเป็นตัวเลขในแต่ละ อ็อกเท็ตในอินสแตนท์ไอเด็นติไฟเออร์ เช่น ถ้า String นั้นมีความยาว N ตัวอักษรก็หมายความว่า ค่าในอินเดกซ์อ็อบเจ็กแปลงเป็นตัวเลขได้ N อ็อกเท็ต
- *ค่าของ String ที่มีความยาวไม่คงที่ (String-valued, variable-length):* ให้นำค่าของตัวอักษรในแต่ละอ็อกเท็ตมาแปลงเป็นตัวเลขในแต่ละอ็อกเท็ตในอินสแตนท์ไอเด็นติไฟเออร์และต่อท้ายด้วยค่าของจำนวนตัวอักษรทั้งหมดอีกหนึ่งอ็อกเท็ต เช่น ถ้า String นั้นมีความยาว N ตัวอักษรก็หมายความว่า ค่าในอินเดกซ์อ็อบเจ็กแปลงเป็นตัวเลขได้ N+1 อ็อกเท็ต โดยอ็อกเท็ตที่ N+1 จะคือค่าของจำนวนตัวอักษรทั้งหมด
- *ค่าของอ็อบเจ็กไอเด็นติไฟเออร์ (Object-identifier-valued):* ค่าในแต่ละอ็อกเท็ตของอ็อบเจ็กไอเด็นติไฟเออร์จะกลายเป็นค่าตัวเลขในแต่ละอ็อกเท็ตในอินสแตนท์ไอเด็นติไฟเออร์โดยอัตโนมัติและต่อท้ายด้วยค่าของจำนวนอ็อกเท็ตทั้งหมดอีกหนึ่งอ็อกเท็ต เช่น ถ้าค่าของอ็อบเจ็กไอเด็นติไฟเออร์มีความยาว N อ็อกเท็ตก็หมายความว่า ค่าในอินเดกซ์อ็อบเจ็กแปลงเป็นตัวเลขได้ N+1 อ็อกเท็ต โดยอ็อกเท็ตที่ N+1 จะคือค่าจำนวนอ็อกเท็ตของอ็อบเจ็กไอเด็นติไฟเออร์ทั้งหมดที่มี
- *ค่าของ IP address (IpAddress-valued):* ค่าของ IP Address จะแปลงเป็น 4 อ็อกเท็ตของอินสแตนท์ไอเด็นติไฟเออร์ได้ทันที

4.7 ลักษณะของโปรโตคอล (Protocol Specification)

การติดต่อระหว่างสถานีจัดการกับเอเจนต์มีรูปแบบในการติดต่อที่เรียกว่า protocol data units หรือ พีดียู (PDU) หลายรูปแบบด้วยกันตามวัตถุประสงค์ในการติดต่อ แบบของการติดต่อในเอสเอ็นเอ็มพี รุ่น 1 มี 5 แบบคือ

4.7.1 GetRequest PDU ใช้สอบถามข้อมูลจากตัวเอเจนต์ที่อยู่บนอุปกรณ์ที่ต้องการตรวจสอบในระบบเครือข่าย

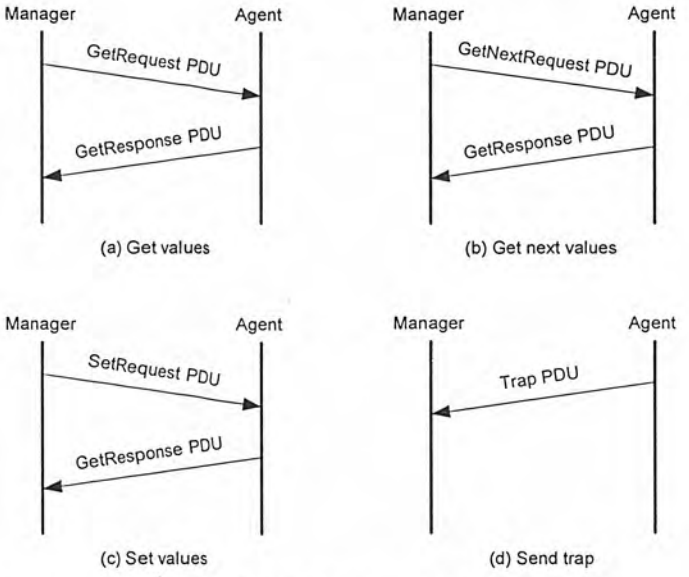
4.7.2 GetNextRequest PDU ใช้สอบถามข้อมูลที่เรียงเป็นลำดับ เช่นข้อมูลที่เก็บอยู่ในรูปตาราง หรือในกรณีที่ไม่ทราบชื่อตัวแปรที่แน่ชัด

4.7.3 GetResponse PDU เอเจนต์ส่งคำตอบกลับมายังผู้สอบถาม

4.7.4 SetRequest PDU ใช้เปลี่ยนแปลงค่าของอ็อบเจ็กที่เอเจนต์รับผิดชอบอยู่

4.7.5 Trap PDU ใช้แจ้งเหตุการณ์ที่เกิดขึ้นในระบบเครือข่าย เช่นการเริ่มต้นทำงานใหม่ของอุปกรณ์ หรือเส้นทางขัดข้อง

เอสเอ็นเอ็มพีอาศัยโพรโตคอลยูดีพี โดยเอเจนต์จะใช้พอร์ตหมายเลข 161 สำหรับการติดต่อในแบบที่ 1 ถึง 3 จากสถานีจัดการเครือข่ายและสถานีจัดการเครือข่ายจะใช้พอร์ตหมายเลข 162 สำหรับการติดต่อในแบบที่ 5 จากเอเจนต์ โดยจะมีลำดับการทำงานในการติดต่อทั้ง 5 รูปแบบดังรูปที่ 4-9

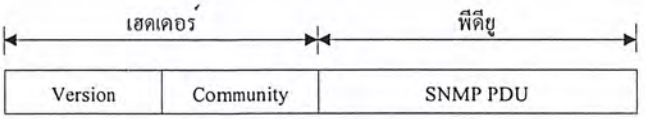


รูปที่ 4-9 ลำดับการทำงานของ SNMP PDU

4.7.1 การเอ็นแคปซูเลต (Encapsulation)

การเอ็นแคปซูเลตคำสั่งและข้อมูลในเอสเอ็นเอ็มพีมีวิธีตามรูปที่ 4-10 ฟอร์แมตของเอสเอ็นเอ็มพีประกอบด้วย 2 ส่วนคือ เฮดเดอร์และพีดียู เฮดเดอร์ประกอบด้วยฟิลด์ย่อยสองฟิลด์คือ

- Version รุ่นของโพรโตคอลที่ใช้ ถ้าเป็นโพรโตคอลรุ่น 1 จะมีค่า 0 หากเป็นรุ่น 2 จะมีค่า 1
- Community รหัสผ่านในรูปสายอักขระเพื่อให้เอเจนต์ใช้ในการพิสูจน์ตัวตนว่าข้อความที่ส่งมามีสิทธิ์ในการสอบถามหรือเปลี่ยนแปลงข้อมูลหรือไม่ ซึ่งรายละเอียดได้อธิบายไว้ในหัวข้อ Communities and Community Names ที่ผ่านมา



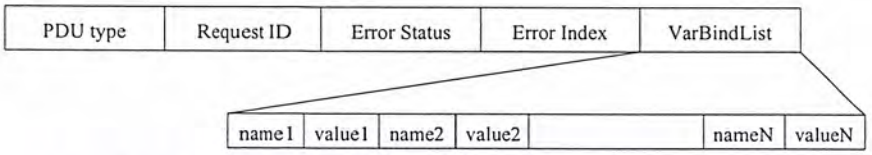
รูปที่ 4-10 การเอ็นแคปซูเลตเอสเอ็นเอ็มพี

ในส่วนของพีดียูประกอบด้วยฟิลด์ย่อยตามชนิดของข้อความ หากเป็นข้อความ GetRequest PDU, GetNextRequest PDU, GetResponse PDU และ SetRequest PDU จะมีโครงสร้างเดียวกัน รูปที่ 4-11 แสดงโครงสร้างของพีดียูโดยแต่ละฟิลด์มีความหมายดังนี้

- PDU type รูปแบบการติดต่อ (1 ถึง 5)
- Request ID กำหนดบอกหมายเลขข้อความเพื่อใช้จับคู่เมื่อรับคำตอบกลับมา
- Error Status สถานะความผิดพลาดที่เกิดขึ้นโดยรหัสความผิดพลาดและสถานะผิดพลาดที่ใช้ในเอสเอ็นเอ็มพีจะแสดงในตารางที่ 3 สำหรับข้อความ

GetRequest PDU, GetNextRequest PDU และ SetRequest PDU จะมีค่าในฟิลด์นี้เป็น 0 เสมอ

- Error Index ดรรชนีชี้ค่าผิดพลาดที่เกิดขึ้นเกิดจากตัวแปรตัวลำดับที่เท่าไหร่ของตัวแปรทั้งหมดที่สอบถามไปสำหรับข้อความ GetRequest PDU, GetNextRequest PDU และ SetRequest PDU จะมีค่าในฟิลด์นี้เป็น 0 เสมอ
- VarBindList ค่าผูกพันตัวแปร(variable binding) แสดงอยู่ในรูปของการอ้างอิงตัวแปรหรืออ็อบเจ็ก (name) และค่าของตัวแปร (value) ต่อเนื่องกันไปเป็นรายการสำหรับข้อความ GetRequest PDU, GetNextRequest PDU และ SetRequest PDU ค่าของตัวแปรจะมีค่าเป็น NULL เสมอ



รูปที่ 4-11 โครงสร้างพีดียูของ GetRequest PDU, GetNextRequest PDU, GetResponse PDU และ SetRequest PDU

รหัสผิดพลาด	ชื่อ	คำอธิบาย
0	noError	ไม่มีข้อผิดพลาด
1	tooBig	เอเจนต์ไม่สามารถส่งคำตอบได้ในเฟรมเดียว
2	noSuchName	ไม่มีตัวอ็อบเจ็กที่ต้องการสอบถามอยู่ในฐานข้อมูล
3	badValue	ค่าที่กำหนดให้อ็อบเจ็กไม่ถูกต้อง
4	readOnly	เปลี่ยนค่าอ็อบเจ็กไม่ได้เพราะอ่านค่าได้เพียงอย่างเดียว
20	genErr	มีข้อผิดพลาดอื่นๆเกิดขึ้น

ตารางที่ 4-3 รหัสและสถานะความผิดพลาดในเอสเอ็นเอ็มพี

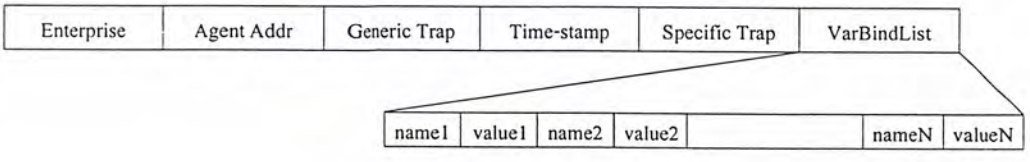
สำหรับข้อความ Trap PDU มีลักษณะแตกต่างกันออกไปดังรูปที่ 4-12 โดยแต่ละฟิลด์มีความหมายดังนี้

- Enterprise ชนิดของตัวแปรที่สร้าง Trap นี้ขึ้นมา โดยค่าในฟิลด์นี้จะอ้างอิงกับค่าในอ็อบเจ็ก sysObjectID
- Agent Addr ค่า IP Address ของอ็อบเจ็กที่สร้าง Trap นี้ขึ้นมา
- Generic Trap ชนิดของ Trap ที่เกิดขึ้นโดยชนิดของ Trap และรหัสของ Trap ที่ใช้ในเอสเอ็นเอ็มพีจะแสดงในตารางที่ 4-4

- Specific Trap ชนิดของเหตุการณ์ผิดปกติเกิดขึ้นกับ enterprise-specific
- Time-stamp เวลาที่ใช้ไปทั้งหมดตั้งแต่การเริ่มการติดต่อในเครือข่ายรวมถึงเวลาที่ใช้ในการสร้าง Trap โดยค่าดังกล่าวจะเก็บอยู่ในอ็อบเจ็ก sysUpTime

รหัส Trap	ชื่อ	คำอธิบาย
0	coldStart	มีการเริ่มต้นการทำงานใหม่ ด้วยสาเหตุของการเปลี่ยนแปลงค่าในเอเจนต์ หรือมีการเปลี่ยนแปลงโปรโตคอลเอ็นจิน เช่น การ restart อุปกรณ์หลังจากเกิดการล้มเหลวของระบบ (crash)
1	warmStart	มีการเริ่มต้นการทำงานใหม่ แต่ไม่ได้เกิดจากสาเหตุของการเปลี่ยนแปลงค่าในเอเจนต์ หรือโปรโตคอลเอ็นจิน เช่น การ restart routine
2	linkDown	การเชื่อมต่อของเอเจนต์มีปัญหา จะมีการค่าของอ็อบเจ็ก ifIndex เพื่อบอกจุดเชื่อมต่อ (interface) ที่มีปัญหา
3	linkUp	การเชื่อมต่อของเอเจนต์กลับมาใช้งานได้ จะมีการค่าของอ็อบเจ็ก ifIndex เพื่อบอกจุดเชื่อมต่อ (interface) ที่มีการกลับมาใช้งานได้
4	authenticationFailure	การพิสูจน์ตัวตนของ message ที่เข้ามาไม่ผ่านหรือมีการผิดพลาดเกิดขึ้น
5	egpNeighborLoss	โปรโตคอลเกตเวย์ภายนอก (External gateway protocol) มีปัญหา
6	enterproseSpecific	มีเหตุการณ์ผิดปกติเกิดขึ้นกับ enterprise-specific โดยจะมีการประกาศชนิดของเหตุการณ์ผิดปกติเกิดขึ้นในฟิลด์ specific-trap

ตารางที่ 4-4 รหัสและชนิดของ Trap ในเอสเอ็นเอ็มพี



รูปที่ 4-12 โครงสร้างพีดียูของคำสั่ง Trap PDU

โปรดสังเกตว่าในที่นี้ไม่ได้กล่าวขนาดความยาวของแต่ละฟิลด์ เพราะทุกฟิลด์ในเอสเอ็นเอ็มพีต้องเข้ารหัสและจะได้ขนาดของแต่ละฟิลด์ที่มีความยาวแตกต่างกันไปตามชนิดข้อมูล

4.7.2 กลไกในการส่ง SNMP Message (Transmission of an SNMP Message)

กลไกในการส่ง SNMP Message ของ PDU ทั้ง 5 แบบในส่วนของโปรโตคอลเอ็นจิน โดยการทำงานจะมีขั้นตอนดังต่อไปนี้

4.7.2.1 สร้าง PDU ตามโครงสร้างที่กำหนดไว้ใน ASN.1 (Abstract Syntex Notation One)

4.7.2.2 PDU ที่สร้างในขั้นตอนที่หนึ่งรวมค่า transport addresses ของต้นทางและปลายทาง และ Community name จะถูกส่งไปให้ส่วนของการบริการพิสูจน์ตัวตน (Authentication service) ซึ่งจะทำการปรับเปลี่ยนข้อมูลที่จำเป็นในการแลกเปลี่ยน เช่น การเข้ารหัสลับ (encryption) หรือการสร้างโค้ดที่ใช้ในการพิสูจน์ตัวตน หลังจากการปรับเปลี่ยนเสร็จก็จะคืนค่ากลับมายังโปรโตคอลเอ็นจิน

4.7.2.3 โปรโตคอลเอ็นจินจะสร้าง message ของเอสเอ็นเอ็มพีโดยการรวมฟิลด์รุ่นของโพรโตคอล (Version), Community name, และผลลัพธ์ที่ได้ในขั้นตอนที่ 2

4.7.2.4 ทำการเข้ารหัส message เอสเอ็นเอ็มพีที่ได้จากขั้นตอนที่ 3 โดยวิธีการ Basic Encoding Rule (BER) แล้วส่ง message ที่เข้ารหัสแล้วไปยังโปรโตคอลในชั้น Transport ต่อไป

4.7.3 กลไกในการรับ SNMP Message (Receipt of an SNMP Message)

กลไกในการรับ SNMP Message ของ PDU ในส่วนของโปรโตคอลเอ็นจิน โดยการทำงานจะมีขั้นตอนดังต่อไปนี้

4.7.3.1 ตรวจสอบความถูกต้องทางไวยากรณ์ขั้นพื้นฐานของ message โดยจะกำจัด message ทิ้งถ้าพบความผิดพลาด

4.7.3.2 รุ่นของโพรโตคอล (Version) ที่ใช้ โดยจะกำจัด message ทิ้งถ้าพบว่าเป็นคนละรุ่นกัน

4.7.3.3 โปรโตคอลเอ็นจินจะนำค่าของ use name, ส่วนของ PDU และ transport addresses ของต้นทางและปลายทาง ส่งไปให้ส่วนของการบริการพิสูจน์ตัวตน (Authentication service)

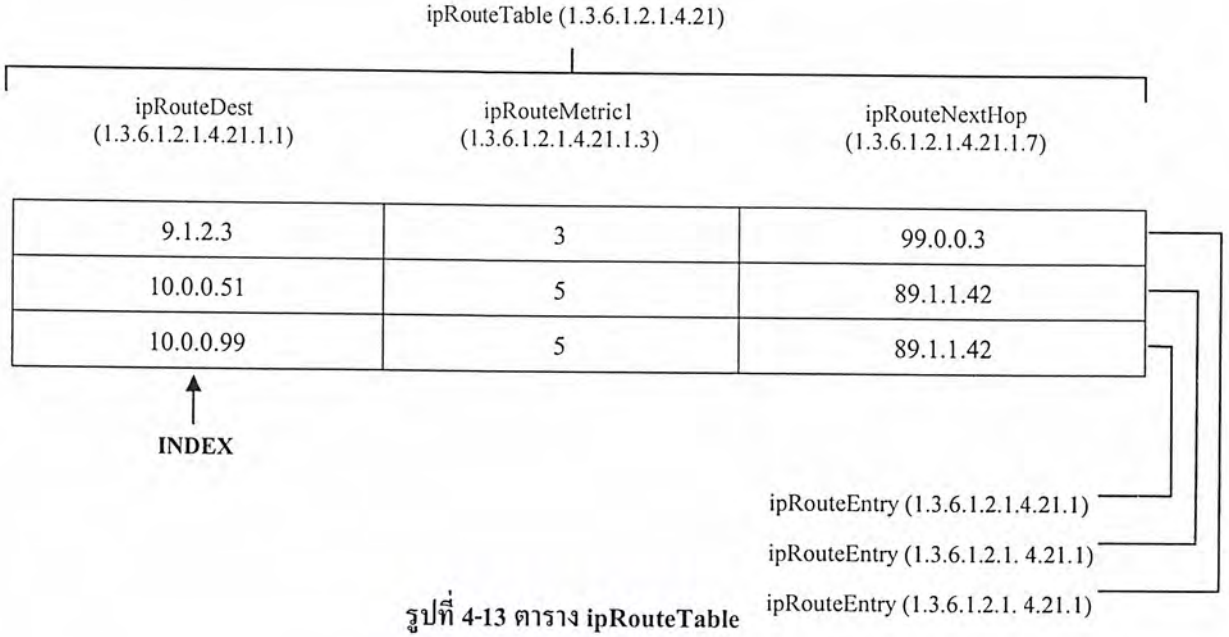
- ถ้าการพิสูจน์ตัวตนล้มเหลว ส่วนของการบริการพิสูจน์ตัวตนจะส่งสัญญาณบอกไปยังโปรโตคอลเอ็นจินเพื่อให้ทำการสร้าง Trap และจะกำจัด message นั้นทิ้ง
- ถ้าการพิสูจน์ตัวตนสำเร็จ ส่วนของการบริการพิสูจน์ตัวตนจะคืนค่า PDU ที่อยู่ในโครงสร้างของ ASN.1 มาให้กับโปรโตคอลเอ็นจิน

4.7.3.4 ตรวจสอบความถูกต้องทางไวยากรณ์ขั้นพื้นฐานของ PDU โดยจะกำจัด PDU ทิ้งถ้าพบความผิดพลาด หลังจากนั้นจะนำชื่อของ Community name เพื่อจัดสรรรูปแบบนโยบายการเข้าถึง (Access policy) ตาม Community ของ PDU นั้น และทำการประมวลผลตามคำสั่งใน PDU ต่อไป

4.7.4 GetRequest PDU

GetRequest PDU จะถูกส่งจากสถานีจัดการเครือข่ายเพื่อใช้สอบถามข้อมูลจากตัวเอเจนต์ที่อยู่บนอุปกรณ์ที่ต้องการตรวจสอบในระบบเครือข่าย โดยจะมีลำดับขั้นตอนการทำงานดังรูปที่ 8 (a)

เอสเอ็นเอ็มพีจะอนุญาตให้การอ้างอิงกระทำได้ที่โหนดปลายเท่านั้น และไม่อนุญาตอ้างอิงได้ทั้งโครงสร้างของตารางในคราวเดียวดังที่กล่าวในหัวข้อที่ผ่านมา อย่างไรก็ตามสถานีจัดการเครือข่ายสามารถที่จะอ้างอิงถึงค่าในทั้งแถวของตารางพร้อมกันได้ โดยอาศัยคุณสมบัติของฟิลด์ VarBindList ซึ่งสามารถผกพันค่าของตัวแปรเป็นรายการได้



รูปที่ 4-13 ตาราง ipRouteTable

ตัวอย่างของตาราง ipRouteTable ดังรูปที่ 12 สถานีจัดการเครือข่ายสามารถสอบถามค่าของอ็อบเจ็กได้ทั้งแถวของตารางโดยใช้รูปแบบของการติดต่อคือ

GetRequest (ipRouteDest.9.1.2.3, ipRouteMetricl.9.1.2.3, ipRouteNextHop.9.1.2.3)

สำหรับในบางกรณีที่ของ Object มีขนาดใหญ่มากและมีการสอบถามค่ามาที่ละหลายอ็อบเจ็กเกินกว่าที่ GetResponse PDU เพียง message เดียวจะตอบกลับมาได้ อาจจะเป็นผลให้ไม่มีการคืนค่าของข้อมูลกลับมาเลยโดยอาจมีการส่งค่าสถานะความผิดมาเป็น tooBig ได้

4.7.5 GetNextRequest PDU

GetNextRequest PDU จะถูกส่งจากสถานีจัดการเครือข่ายเพื่อใช้สอบถามข้อมูลที่เรียงเป็นลำดับเช่นข้อมูลที่เก็บอยู่ในรูปตาราง หรือในกรณีที่ไม่ทราบชื่อตัวแปรที่แน่ชัด โดยจะมีลำดับขั้นตอนการ

ทำงานดังรูปที่ 8 (b) ประโยชน์ของการใช้การติดต่อแบบ GetNextRequest PDU คือจะสามารถใช้ในการค้นหาโครงสร้างของฐานข้อมูลสารสนเทศการจัดการได้อย่างมีประสิทธิภาพ สำหรับรูปแบบของการสอบถามข้อมูลของ GetNextRequest PDU จะมีรูปแบบของการสอบถามข้อมูล 2 รูปแบบคือ

4.7.5.1 การสอบถามค่าของอ็อบเจ็กแบบปกติ (Simple Object Value)

สำหรับการสอบถามค่าของอ็อบเจ็กแบบปกติ จากตัวอย่างถ้าสถานีจัดการเครือข่ายต้องการสอบถามค่าของอ็อบเจ็กแบบที่ไม่ใช่ตารางทั้งหมดในกลุ่มของ udp สถานีจัดการเครือข่ายจะส่งการติดต่อแบบ GetRequest ซึ่งระบุค่าการอ้างอิงของอินสแตนท์ไอเด็นติไฟเออร์ คือ

GetRequest (udpInDatagrams.0, udpNoPorts.0, udpInErrors.0,
udpOutDatagrams.0)

ถ้าในกรณีที่ Community ที่อยู่ในเอเจนต์สนับสนุนอ็อบเจ็กเหล่านี้ทุกตัว เอเจนต์จะส่ง GetResponse PDU กลับมาในรูปแบบ

GetResponse ((udpInDatagrams.0 = 100), (udpNoPorts.0 = 1),
(udpInErrors.0 = 2), (udpOutDatagrams.0 = 200))

ค่าของ 100, 1, 2 และ 200 คือค่าของอ็อบเจ็กทั้ง 4 ที่สอบถามไป แต่อย่างไรก็ตามถ้าเกิดในกรณีที่อ็อบเจ็กตัวใดตัวหนึ่งหรือหลายตัวไม่มีตัวตน ทำให้ GetResponse PDU ที่ส่งกลับมาจะมีค่าสถานะความผิดพลาดเป็น noSuchName และไม่มีการคืนค่าที่สอบถามใดๆมา

แต่สำหรับกลไกการทำงานของการติดต่อแบบ GetNextRequest PDU คือเมื่อเอเจนต์ได้รับ GetNextRequest PDU ซึ่งระบุค่าของอ็อบเจ็กไอเด็นติไฟเออร์ไม่ใช่อินสแตนท์ไอเด็นติไฟเออร์ ในกรณีนี้ค่าที่ได้รับกลับมาจะคือค่าที่เก็บอยู่ในอ็อบเจ็กที่อ้างอิงตามอ็อบเจ็กไอเด็นติไฟเออร์ในกรณีถ้าอ็อบเจ็กนั้นมีตัวตน แต่ถ้าในกรณีที่อ็อบเจ็กที่สอบถามไปนั้นไม่มีตัวตนค่าที่ส่งคืนกลับมาแทนคือค่าที่เก็บอยู่ในอ็อบเจ็กที่อ้างอิงตาม
อ็อบเจ็กไอเด็นติไฟเออร์ตัวถัดไป ซึ่งกลไกการทำงานดังกล่าวจะมีประสิทธิภาพกว่าการติดต่อในแบบ GetRequrest

ถ้าตัวอย่างที่ผ่านมาถ้าสถานีจัดการเครือข่ายจะอาศัย GetNextRequest PDU โดยส่งการติดต่อในรูปแบบของ

GetNextRequest (udpInDatagrams, udpNoPorts, udpInErrors, udpOutDatagrams)

ในกรณีนี้จะได้รับ GetResponse PDU ตอบกลับมาคือ

GetResponse ((udpInDatagrams.0 = 100), (udpNoPorts.0 = 1),
(udpInErrors.0 = 2), (udpOutDatagrams.0 = 200))

และถ้าเกิดในกรณีที่อ็อบเจ็ก udpNoPorts ไม่มีตัวตนทำให้ค่าที่ได้รับคืนกลับมาแทนส่วนค่าของ
อ็อบเจ็ก udpNoPorts ก็คือค่าที่เก็บอยู่ในอ็อบเจ็กที่อ้างอิงตามอ็อบเจ็กไอเด็นติไฟเออร์ตัวถัดไปซึ่งก็คือค่าใน
อ็อบเจ็ก udpInErrors โดยจะได้รับ GetResponse PDU ตอบกลับมาคือ

GetResponse ((udpInDatagrams.0 = 100), (udpInErrors.0 = 2),
(udpInErrors.0 = 2), (udpOutDatagrams.0 = 200))

4.7.5.2 การสอบถามค่าของอ็อบเจ็กในรูปแบบตาราง (Sequence of Object Value)

สำหรับการสอบถามค่าของอ็อบเจ็กในรูปแบบตาราง โดยใช้การติดต่อแบบ GetNextRequest PDU นั้นจะอาศัยเทคนิคของ Serial-access technique เนื่องจากการติดต่อแบบ GetNextRequest จะสามารถนะค่าถัดไปมาได้โดยไม่จำเป็นต้องเจาะจงค่าถัดไปโดยตรงซึ่งจะสะดวกมาต่ออ็อบเจ็กที่เป็นชนิด sequence และ sequence of โดยลำดับการนำค่าในแต่ละอ็อบเจ็กของ GetNextRequest จะมีลำดับเริ่มจากคอลัมน์แรกไปจนสิ้นสุดทุกแถวในคอลัมน์นั้นก่อนที่จะขึ้นคอลัมน์ถัดไป

ตัวอย่างเช่นถ้าสถานีจัดการเครือข่ายต้องการสอบถามค่าของอ็อบเจ็กที่อยู่ในตาราง ipRouteTable ดังในรูปที่ 12 ที่ในหัวข้อผ่านมาโดยจะเริ่มที่ GetNextRequest PDU ซึ่งระบุค่าของอ็อบเจ็กไอเด็นติไฟเออร์ของอ็อบเจ็กในแต่ละคอลัมน์ดังนี้

GetNextRequest (ipRouteDest, ipRouteMetric1, ipRouteNextHop)

ซึ่งจะได้รับการตอบกลับจากเอเจนต์คือค่าของอ็อบเจ็กในตารางแถวแรกดังนี้

GetResponse ((ipRouteDest.9.1.2.3 = 9.1.2.3), (ipRouteMetric1.9.1.2.3 = 3),
(ipRouteNextHop.9.1.2.3 = 99.0.0.3))

สถานีจัดการเครือข่ายจะนำค่าที่ได้รับการตอบรับดังกล่าวมาใช้ในการอ้างอิงโดยใช้อินสแตนท์ไอเด็นติไฟเออร์เพื่อสอบถามข้อมูลในแถวที่สองของตารางดังนี้

GetNextRequest (ipRouteDest.9.1.2.3, ipRouteMetric1.9.1.2.3,
ipRouteNextHop.9.1.2.3)

ซึ่งจะได้รับการตอบกลับจากเอเจนต์คือค่าของอ็อบเจ็กในตารางแถวสองดังนี้

GetResponse ((ipRouteDest.10.0.0.51 = 10.0.0.51),
(ipRouteMetric1.10.0.0.51 = 5), (ipRouteNextHop.10.0.0.51 = 89.1.1.42))

และสถานีจัดการเครือข่ายจะนำค่าที่ได้รับการตอบรับดังกล่าวมาใช้ในการอ้างอิงโดยใช้อินสแตนท์ไอเด็นติไฟเออร์เพื่อสอบถามข้อมูลในแถวที่สามของตารางและได้รับการตอบกลับจากเอเจนต์ดังนี้

GetNextRequest (ipRouteDest.10.0.0.51, ipRouteMetric10.0.0.51,
ipRouteNextHop.10.0.0.51)

GetResponse ((ipRouteDest.10.0.0.99 = 10.0.0.99), (ipRouteMetric1.10.0.0.99 = 5),
(ipRouteNextHop.10.0.0.99 = 89.1.1.42))

เนื่องจากสถานีจัดการเครือข่ายไม่ทราบว่าข้อมูลในตารางหมดแล้ว จึงนำค่าที่ได้รับการตอบรับในแถวที่สามดังกล่าวมาใช้ในการอ้างอิงโดยใช้อินสแตนท์ไอเด็นติไฟเออร์เพื่อสอบถามข้อมูลในแถวที่สี่และได้รับการตอบกลับมาคือ

GetNextRequest (ipRouteDest.10.0.0.99, ipRouteMetric10.0.0.99,
ipRouteNextHop.10.0.0.99)

GetResponse ((ipRouteMetric1.9.1.2.3 = 3), (ipRouteNextHop.9.1.2.3 = 99.0.0.3),
(ipNetToMediaIfIndex.1.3 = 1))

จะเห็นได้ว่าค่าของอ็อบเจ็กที่ได้รับการตอบกลับจากเอเจนต์เป็นการขึ้นคอลัมน์ถัดไป ซึ่งก็จะหมายความว่าได้สิ้นสุดแถวของตารางแล้วนั้นเอง

4.7.6 SetRequest PDU

ใช้เปลี่ยนแปลงค่าของอ็อบเจ็กที่เอเจนต์รับผิดชอบอยู่ ซึ่งฟิลด์ของ VarBindList จะประกอบด้วยค่าอ้างอิงของตัวแปรหรืออ็อบเจ็ก และค่าของอ็อบเจ็กเหล่านั้นซึ่งค่าเหล่านี้จะเป็นค่าที่นำไปใช้เปลี่ยนแปลงค่าตัว

อ็อบเจ็กในฐานข้อมูลสารสนเทศการจัดการที่เอเจนต์รับผิดชอบอยู่ สำหรับลำดับขั้นตอนการทำงานจะแสดงดังรูปที่ 8 (c)

สำหรับการใช้งานของ SetRequest PDU กับค่าในอ็อบเจ็กที่อยู่ในรูปแบบตาราง นอกเหนือจากการเปลี่ยนแปลง (Update) ค่าในตาราง ยังจะสามารถที่จะเพิ่มแถว (Insert row) หรือลบ (Delete) ค่าในตารางโดยจะอาศัยตัวอย่างจากตาราง ipRouteTable ในรูปที่ 12 ซึ่งมีรายละเอียดดังนี้

4.7.6.1 การเปลี่ยนปลงค่าในตาราง (Update a Table) SetRequest PDU สามารถที่การเปลี่ยนแปลงค่าในตารางได้โดยการอ้างอิงถึงค่าของอ็อบเจ็กจะอาศัยการอ้างอิงของอินสแตนท์ไอเด็นติไฟเออร์เช่นเดียวกับ

อ็อบเจ็กปกติทั่วไปยกตัวอย่างเช่น ถ้าสถานีจัดการเครือข่ายส่งการติดต่อในรูปแบบ

SetRequest (ipRouteMetric1.9.1.2.3 = 9)

และเอเจนต์จะส่งการติดต่อกลับคืนมาในรูปแบบ

GetResponse (ipRouteMetric1.9.1.2.3 = 9)

การติดต่อดังกล่าวคือสถานีจัดการเครือข่ายได้ส่ง SetRequest PDU เพื่อทำการเปลี่ยนแปลงค่าของอ็อบเจ็ก ipRouteMetric1 ที่อยู่ในแถวแรกเนื่องจากในส่วนของการอ้างอิงได้ระบุค่าในอินเดกซ์อ็อบเจ็กของแถวแรกไว้คือ 9.1.2.3 และเมื่อเอเจนต์ได้รับก็จะทำการเปลี่ยนแปลงค่าในอ็อบเจ็ก ipRouteMetric1 จาก 3 เป็น 9 แล้วจะส่ง GetResponse กลับไปให้สถานีจัดการเครือข่ายเพื่อยืนยันการเปลี่ยนแปลง

4.7.6.2 การเพิ่มแถวในตาราง (Insert row to a Table) ถ้าสถานีจัดการเครือข่ายต้องการที่จะเพิ่มแถวในตาราง ipRouteTable โดยกำหนดค่าในแต่ละอ็อบเจ็ก ipRouteDest, ipRouteMetric1 และ ipRouteNextHop เป็น 11.3.3.12, 9 และ 91.0.0.5 ตามลำดับ สถานีจัดการเครือข่ายส่งการติดต่อในรูปแบบ

SetRequest ((ipRouteDest.11.3.3.12 = 11.3.3.12), (ipRouteMetric1.11.3.3.12 = 9),
(ipRouteNextHop.11.3.3.12 = 91.0.0.5))

ซึ่งการเปลี่ยนแปลงค่าดังกล่าวจะเห็นได้ว่าไม่มีค่าของอ็อบเจ็ก ipRouteDest ซึ่งเป็นอินเดกซ์อ็อบเจ็กในแถวใดเลยที่มีค่าเป็น 11.3.3.12 ใน RFC 1212 จะมี 3 ทางเลือกให้เอเจนต์จัดการกับเหตุการณ์นี้คือ

- เอเจนต์ทำการปฏิเสธการทำงานของ SetRequest PDU และทำการส่ง GetResponse PDU ที่มีสถานะความผิดพลาดเป็น noSuchName
- เอเจนต์จะทำการยอมรับการทำงานของ SetRequest PDU แต่ถ้าเกิดพบว่าค่าที่ส่งมาให้เปลี่ยนแปลงนั้นผิดหลักไวยากรณ์หรือมีชนิดของอ็อบเจ็กผิดไปจากที่ได้กำหนดไว้ ก็จะไม่มีการสร้างแถวใหม่ในตารางและทำการส่ง GetResponse PDU ที่มีสถานะความผิดพลาดเป็น badValue
- เอเจนต์จะทำการยอมรับการทำงานของ SetRequest PDU และเพิ่มแถวใหม่ลงในตาราง ถ้าสมมุติว่าเอเจนต์จัดการตามทางเลือกที่ 3 คือทำการสร้างแถวใหม่ให้ในตาราง และทำการส่ง GetResponse กลับคืนมาคือ

GetResponse ((ipRouteDest.11.3.3.12 = 11.3.3.12), (ipRouteMetric1.11.3.3.12 = 9),
(ipRouteNextHop.11.3.3.12 = 91.0.0.5))

แต่ถ้าในกรณีที่สถานีจัดการเครือข่ายต้องการที่จะเพิ่มแถวในตารางแต่ส่งค่ามากับ SetRequest PDU เพียงบางค่า คือ

SetRequest ((ipRouteDest.11.3.3.12 = 11.3.3.12))

จะมีสองทางเลือกที่เอเจนต์จะจัดการกับเหตุการณ์ดังกล่าวได้แก่

- เอเจนต์จะเพิ่มแถวใหม่ลงไปในตารางแต่ค่าในอ็อบเจ็กที่ SetRequest ไม่ได้ส่งมาจะถูกกำหนดเป็นค่าโดยปริยาย (Default values)
- เอเจนต์ทำการปฏิเสธการทำงานของ SetRequest PDU โดยในกรณีนี้เอเจนต์จะต้องการให้ SetRequest ส่งค่าของอ็อบเจ็กทุกตัวในแถวในกรณีต้องการเพิ่มแถวใหม่

4.7.6.3 การลบแถวในตาราง (Delete row in a Table) คำสั่ง set ยังสามารถใช้ลบแถวในตารางได้ เช่น สถานีจัดการเครือข่ายส่ง SetRequest PDU และได้รับการตอบกลับจากเอเจนต์เป็น

SetRequest (ipRouteNextHop.10.0.0.51 = invalid)
GetResponse (ipRouteDest.10.0.0.51 = invalid)

เอเจนต์เมื่อได้รับ SetRequest มาเพื่อเปลี่ยนแปลงค่าของ ipRouteNextHop ในแถวที่มีค่าของอินเดกซ์อ็อบเจ็กคือ ipRouteDest เป็น 10.0.0.51 ให้มีค่าเป็น invalid ซึ่งก็จะมีความหมายว่าให้ลบค่าของแถวที่มีค่าของอินเดกซ์อ็อบเจ็กคือ ipRouteDest เป็น 10.0.0.51 นั้นเอง และทำ

การส่ง GetResponse และที่ระบุค่าของอินเดกซ์อ็อบเจ็กในแถวที่ลบเป็น invalid เพื่อทำการยืนยัน

4.8 การเข้ารหัสโดยใช้ BER (Basic Encoding Rules)

การส่งข้อมูลในSNMP ไม่ได้ส่งในรูปแบบเป็น ออกเทตของของตัวเลขโดยตรง หากแต่ฝ่ายส่งต้องเข้ารหัสข้อมูลเพื่อนำส่งข้อมูลออก และ ถอดรหัสที่ฝ่ายรับ

4.8.1 โครงสร้างการเข้ารหัส

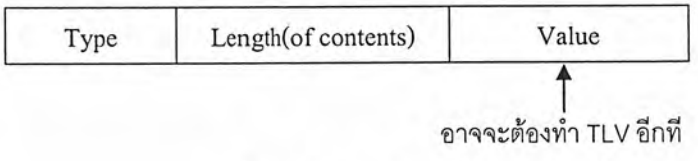
การเข้ารหัสตามแบบ BER จะมีข่าวสารกำกับอยู่ในตัวว่าเป็นข้อมูลชนิดใด และ มีความยาวเท่าใด ฟิลด์ที่ผ่านการเข้ารหัสแล้วประกอบด้วยค่า 3 ส่วน คือ

4.8.1.1 ชนิดของข้อมูล (type หรือ tag หรือ identifier)

4.8.1.2 ความยาวของข้อมูล (length)

4.8.1.3 ตัวข้อมูล (value หรือ contents)

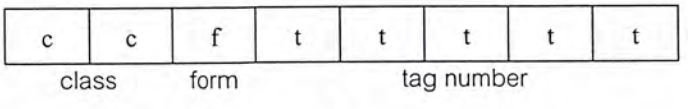
โครงสร้างรหัสเหล่านี้ เรียกว่า โครงสร้าง TLV (TLV : type-length-value structure) ค่าในส่วนของฟิลด์ value อาจจะต้องผ่านการเข้ารหัสค่าตามโครงสร้าง TLV ด้วยเช่นกัน



รูปที่ 4-14 โครงสร้างของ TLV

4.8.2 ฟิลด์กำหนดชนิดข้อมูล (Type)

การกำหนดค่าให้กับฟิลด์ Type ขนาด 8 bit มีการจัดวางตำแหน่งบิต เพื่อให้ได้ค่าตัวเลขที่ใช้แทนกลุ่มชนิดข้อมูล แบ่ง เป็น 3 ส่วนย่อย ดังนี้



รูปที่ 4-15 รูปแบบการเข้ารหัสประเภทข้อมูล (type)

4.8.2.1 ฟิลด์ class 2 bits แรก : กำหนดประเภทข้อมูล มี 4 แบบ คือ

- 00 = ประเภท Universal เช่น ตัวเลขทั่วไป
- 01 = ประเภท Application ข้อมูลสำหรับใช้กับ application program
- 10 = ประเภท Context Specific ข้อมูลเจาะจง เช่นคำสั่งใน SNMP
- 11 = ประเภท Private ข้อมูลสำหรับใช้กรณีเฉพาะ

4.8.2.2 Form 1 bit ถัดมา : กำหนดว่าแบบข้อมูลนั้นเป็นแบบ พื้นฐาน(Primitive) หรือ แบบโครงสร้าง(Constructed) เช่น ตัวอย่างแบบข้อมูลพื้นฐาน เช่น integer หรือ string ส่วนข้อมูลแบบโครงสร้าง ก็เช่น sequence

4.8.2.3 Tag number 5 bits สุดท้าย : เป็นส่วนกำหนดแบบข้อมูลซึ่งมีค่าได้ตั้งแต่ **0** ถึง **30**

	แบบข้อมูล	value (ฐาน 16)
Universal	Integer	02
	OctectString	04
	null	05
	objectIdentifier	06
	sequence	16
Application - wide	IPAddress	40
	Counter	41
	Guage	42
	TimeTicks	43
Context - specific	get-request	A0
	get-next-request	A1
	get-response	A2
	set-request	A3
	trap	A4

รูปที่ 4-16 ตัวอย่าง type value กำหนดรูปแบบข้อมูลที่ใช้ใน SNMP

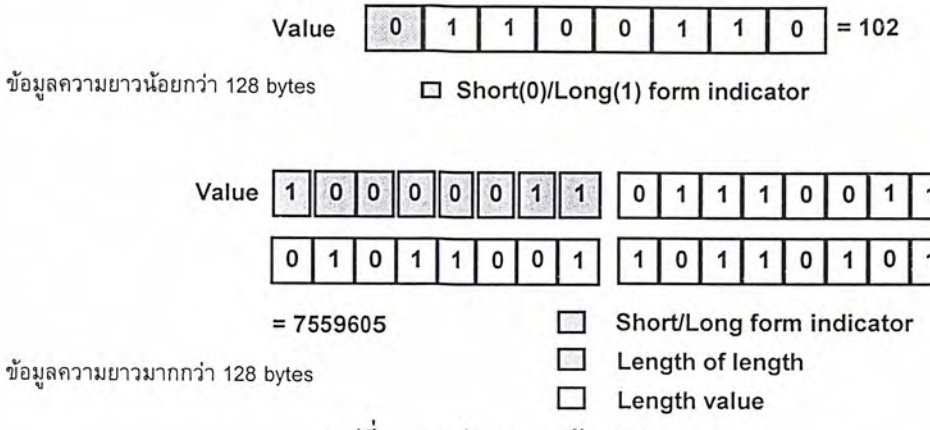
Example:

ข้อมูล Integer จะมี type เท่ากับ 02 หรือแยกออกมาเป็น bit ได้เป็น 0000 0010 ซึ่งมาจาก 00(Universal) , 0(Primative) และ 0 0010(Tag Number)

4.8.3 ฟิลด์กำหนดความยาว (Length)

หากข้อมูลความยาวน้อยกว่า 128 bytes ฟิลด์นี้จะกินเนื้อที่เพียง 1 byte โดยมี bit ซ้ายสุดเป็น "0" และ 7 bit ที่เหลือกำหนดความยาว เช่น ข้อมูลยาว 10 byte ค่าในฟิลด์จะเท่ากับ 0x0A

ข้อมูลมีความยาวตั้งแต่ 128 bytes ขึ้นไป ต้องใช้ฟิลด์ length หลายออกเทต โดยที่บิตซ้ายสุดจะมีค่าเป็น "1" และใช้ 7 bits ที่เหลือเป็นตัวนับจำนวนออกเทตที่กำหนดความยาว ถัดจากนั้นจึงเป็นไบท์กำหนดความยาว เช่น ข้อมูลยาว 1000 bytes (03 E8) ค่าในฟิลด์นี้จะเป็น 82 03 E8 โดยที่ 7 bits ขวาของ B2 คือ 000 0010 ซึ่งเท่ากับ 2 หมายถึงใช้ 2 bytes กำหนดความยาว และ 2 bytes นั้นคือ 03 E8



รูปที่ 4-17 รูปแบบการเข้ารหัสความยาว

4.8.4 ฟิลด์กำหนดข้อมูล (Value)

จะแตกต่างกันไปตามชนิดของข้อมูล

4.8.5 รูปแบบการเข้ารหัส TLV (เฉพาะบางประเภทข้อมูลที่ใช้ใน SNMP)

- ข้อมูลประเภทสายอักขระ ไม่ต้องเข้ารหัส ส่งไปตามลำดับของสายอักขระตามปกติ เช่น สายอักขระ "interfaces" จะอยู่ในรหัสTLV ดังนี้ 04 0A 'i' 'n' 't' 'e' 'r' 'f' 'a' 'c' 'e' 's' 04 คือ OctectString และ 0A คือความยาว
- ข้อมูลตัวเลข
  - ค่าไม่เกิน 127 จะใช้เพียง 1 byte เท่านั้น
  - ค่าเกิน 127 จะต้องผ่านการเข้ารหัสอีกที คือ set bit ซ้ายสุดให้เป็น 1 และใช้ 7 bit ที่เหลือกำหนดจำนวน octet ที่ต้องใช้ จากนั้นจึงค่อยตามด้วยค่า octet ถัดต่อๆไป เช่น ค่าตัวเลข 130 เมื่อเข้ารหัสแล้วจะได้ค่า 02 02 81 02 โดยที่ 02 คือ integer และ 81 02 แทนค่า 130
- Null ให้ส่งโดยเพียงแต่กำหนดค่าในฟิลด์ length เป็น 0 และไม่ต้องส่งค่าใดๆไปทั้งสิ้น
- ข้อมูลประเภท objectIdentifier ต้องเข้ารหัสด้วยวิธีพิเศษ

4.8.6 ตัวอย่างSNMP Frame และการเข้ารหัส

การส่งคำสั่ง request (GetRequest PDU)

ลำดับ Byte เมื่อใช้คำสั่ง GetRequest PDU สอบถามค่า 1.3.6.1.2.1.1.1.0 หรือ sysDescr.0 แต่ละฟิลด์ของSNMP จะผ่านการเข้ารหัสตามโครงสร้างTLV

UDP	30	27	02	01	00	04	06	73	23	6E
	6D	70	21	A0	1A	02	02	22	FB	02
	01	00	02	01	00	30	0E	30	0C	06
	08	2B	06	01	02	01	01	01	00	05
	00									

รูปที่ 4-18 SNMP Frame ของ GetRequest PDU

30	27								
02	01	00							
04	06	73	23	6E	6D	70	21		
A0	1A								
02	02	22	FB						
02	01	00							
02	01	00							
30	0E								
30	0C								
06	08	2B	06	01	02	01	01	01	00
05	00								

Type	Length	value	หมายเหตุ
sequence	39	-	มีข้อมูล 39 bytes ตามมา
integer	1	0	version = 1
string	6	#snmp!	community = #snmp!
context-spc.	26	-	get request 26 bytes
integer	2	22FB	id = 22FB
integer	1	0	error status = 0
integer	1	0	error status = 0
sequence	14	-	
sequence	12	-	
oid	8	sysDescr.0	1.3.6.1.2.1.1.1.0
null	0	0	รหัสปิดท้าย

รูปที่ 4-19 ความหมายของการเข้ารหัสข้อมูลของ GetRequest PDU

การตอบกลับคำสั่ง request (GetResponse PDU)

ลำดับ byte เมื่อ agent ใช้คำสั่ง GetResponse PDU ตอบค่า sysDescr.0 ส่งกลับไป แต่ละฟิลด์ที่ผ่านการเข้ารหัสตามโครงสร้าง TLV จะแสดงได้ดังนี้ (เนื่องจากสายอักขระที่ตอบกลับมามีความยาวมากกว่า 300 byte ดังนั้นจึงเลือกแสดงแค่บางส่วนเท่านั้น)

UDP	30	81	F9	02	01	00	04	06	73	23
	6E	6D	70	21	A2	81	EB	02	02	02
	FB	02	01	00	02	01	00	30	81	DE
	30	81	DB	06	08	2B	06	01	02	01
	01	01	04	81	CE	43	65	73	63	6F

รูปที่ 4-20 SNMP Frame ของ GetResponse PDU

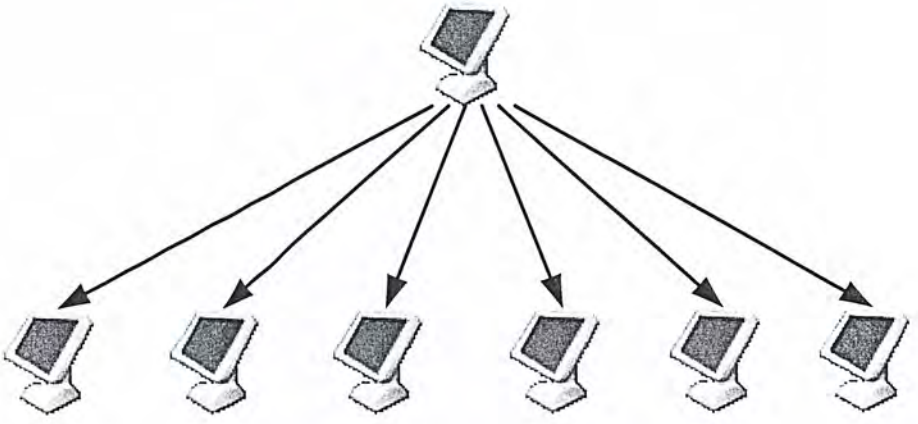
0	81	F9							
2	01	00							
4	06	73	23	6E	6D	70	21		
2	81	EB							
2	02	22	FB						
2	01	00							
2	01	00							
0	81	DE							
0	81	DB							
6	08	2B	06	01	02	01	01	01	00
4	81	CE							
3	65	73	63	6F					

Type	Length	value	หมายเหตุ
sequence	377	-	มีข้อมูล 377 bytes ตามมา
integer	1	0	version = 1
string	6	#snmp!	community = #snmp!
context-spc.	363	-	get response 363 bytes
integer	2	22FB	id = 22FB
integer	1	0	error status = 0
integer	1	0	error index = 0
sequence	14	-	
sequence	12	-	
oid	8	sysDescr.0	1.3.6.1.2.1.1.1.0
OctectString	334	-	
			Cisco.....

รูปที่ 4-21 ความหมายของการเข้ารหัสข้อมูลของ GetResponse PDU

บทที่ 5
การสแกนเพื่อตรวจสอบ

5.1 Network Ping Sweeps



รูปที่5-1 แสดงการทำงานของ ping sweep

หนึ่งในเทคนิคพื้นฐานที่นิยมกระทำกันก็คือ การ ping ไปยังเครื่องเป้าหมายจำนวนมากพร้อมๆ กัน ในลักษณะคล้ายกับการกวาดหรือกราดยิง ซึ่งเราเรียกว่าการทำ ping sweep เพื่อตรวจสอบว่าเครื่องปลายทางใดบ้างที่ยังเปิดทำงานอยู่ โดยปกติ ถ้าคุณใช้คำสั่ง ping ธรรมดา, ping จะมีการส่ง แพ็กเก็ต ICMP ECHO (Type 8) ออกไปยังเครื่องปลายทางและรอคอย ICMP ECHO_REPLY (Type 0) ที่ถูกส่งกลับมา ถึงแม้ ping จะมีประโยชน์สำหรับการทดสอบว่าเครื่องปลายทางเปิดอยู่หรือไม่ก็ตาม แต่มันจะเหมาะสำหรับเครื่องที่อยู่บนเน็ตเวิร์คขนาดเล็กถึงขนาดกลางเท่านั้น มันจะไม่มีประสิทธิภาพเพียงพอที่จะนำมาใช้ตรวจสอบเครื่องที่อยู่บนเน็ตเวิร์คขนาดใหญ่ได้ การตรวจสอบเครื่องที่อยู่ในเน็ตเวิร์คที่ใช้แอดเดรสในคลาส A อาจกินเวลานานหลายชั่วโมงกว่าจะทราบผล เทคนิคในการ ping sweep มีหลายเทคนิคแตกต่างกันไป เช่น ปกติแล้วการ ping จะรอคอยการตอบสนองจากเครื่องทีละเครื่อง ก่อนจะเปลี่ยนไปทดสอบเครื่องอื่นๆถัดไป มันจะใช้การส่ง แพ็กเก็ต ICMP ออกไปพร้อมๆกันแบบขนานไปยังเครื่องปลายทางหลายๆเครื่อง ในลักษณะคล้าย "Round Robin" (คือส่งแพ็กเก็ต ICMP ไปที่เครื่อง 1,2,3,.... ถึงเครื่องสุดท้าย แล้ววนกลับมาส่งแพ็กเก็ตไปที่เครื่อง 1,2,3 ใหม่ไปเรื่อยๆ แล้ววนกลับมาอีก โดยไม่จะเป็นต้องหยุดรอจากตอบสนองจากเครื่องแรก) ดังนั้น จะทำงานได้รวดเร็วกว่าคำสั่ง ping ธรรมดามาก แต่

อาจทำให้เกิดทราฟฟิกจำนวนมากที่เกิดขึ้นอาจเข้าไปรบกวนแบนด์วิธของ WAN Link ความเร็วอย่างต่ำ เช่น 128K ISDN หรือ เฟรมรีเลย์ (Frame relay)

แต่ในบางครั้งจะทำอย่างไรถ้าที่เน็ตเวิร์คเป้าหมายได้มีการบล็อกห้ามแพ็กเก็ต ICMP ไว้ไม่ให้เข้าถึงเน็ตเวิร์คภายในได้

ถ้าแพ็กเก็ต ICMP ถูกบล็อกเอาไว้ เรามีเทคนิคและเครื่องมืออื่นที่ใช้ตรวจสอบได้ว่าเครื่องปลายทางใดบ้างที่เข้าถึงได้และเปิดทำงานอยู่ แต่อย่างไรก็ตาม มันอาจไม่ถูกต้องและรวดเร็วเท่ากับตรวจสอบด้วย ping sweep

เทคนิคที่ว่านั้นก็คือ การสแกนพอร์ต นั่นเอง ซึ่งเป็นเทคนิคแรกที่ใช้ถ้าไซต์ปลายทางบล็อกแพ็กเก็ต ICMP ไว้ โดยการสแกนพอร์ตทั่วๆไปบนแต่ละเครื่องเราจะสามารถคาดการณ์ได้ว่าเครื่องไหนเปิดอยู่บ้าง เทคนิคนี้ค่อนข้างกินเวลาและอาจไม่สามารถสรุปแน่ชัดได้ TCP Ping Scan ด้วย TCP Ping scan ไปที่พอร์ตที่ต้องการส่วนมากจะเป็นพอร์ต 80 เนื่องจากเป็นพอร์ตมาตรฐานที่เราท์เตอร์หรือไฟร์วอลล์ของไซต์ส่วนใหญ่จะเปิดไว้ให้ผ่านเข้าไปได้ยังเว็บเซิร์ฟเวอร์ที่มักตั้งอยู่ในส่วนที่เรียกว่า Demilitarized zone (DMZ) หรือยิ่งไปกว่านั้น อาจทะลุผ่านเข้าไปในอินทราเน็ตภายในด้วย โดยจะทำงานดังนี้คือสร้างแพ็กเก็ตของโพรโตคอล TCP ซึ่งได้เซตแฟลก ACK (Acknowledge) ไว้เป็น 1 ด้วย (ต่อไปเราจะเรียกแพ็กเก็ตลักษณะนี้สั้นๆว่า แพ็กเก็ต TCP ACK) แล้วส่งแพ็กเก็ตเหล่านี้ไปยังเน็ตเวิร์คเป้าหมายปลายทางหรือไม่ ถ้ามีก็แสดงว่าเครื่องปลายทางสามารถติดต่อได้และเปิดทำงานอยู่ แพ็กเก็ต TCP ACK นี้มักวิ่งทะลุผ่านไฟร์วอลล์ที่ไม่ค่อยฉลาดนักหรือคอนฟิกไว้ไม่ดีพอให้เข้าไปได้

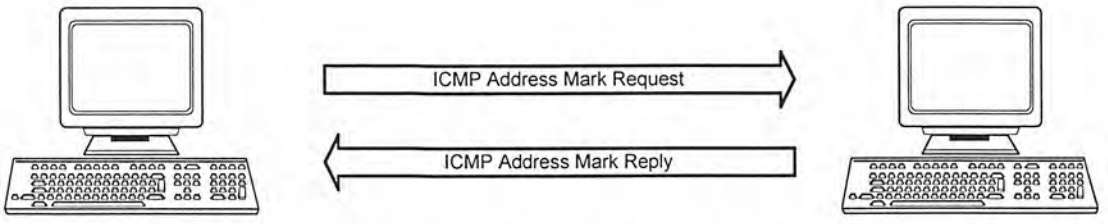
อย่างที่ได้เห็นไปแล้วว่าเทคนิคนี้ค่อนข้างใช้ได้ผลถึงแม้ไซต์ปลายทางจะบล็อกแพ็กเก็ต ICMP ไว้ ซึ่งถ้า พอร์ต 80 ใช้ไม่ได้อาจใช้เป็นพอร์ตมาตรฐานที่มักพบบ่อย อย่างเช่น พอร์ตของเซอร์วิส SMTP (หมายเลข 25), POP (110), AUTH (113), IMAP (143) หรือพอร์ตอื่นๆ ที่เป็นไปได้

ถ้าพอร์ตที่เครื่องปลายทางได้เปิดไว้แพ็กเก็ตของโพรโตคอล TCP ที่มีการเซตแฟลก SYN(S bit) ไว้และแพ็กเก็ต TCP ที่มีการเซตแฟลก ACK ไว้จะถูกส่งกลับมา

นอกจากนี้ยังมีเทคนิคอื่นในการที่จะตรวจสอบว่าเครื่องปลายทางยังเปิดทำงานอยู่หรือไม่ด้วย การส่งแพ็กเก็ต ICMP ECHO ออกไป พร้อมทั้งแพ็กเก็ตของโปรโคคอล ICMP ใน Type อื่นๆด้วย ได้แก่ ICMP Type TIME STAMP REQUEST และ ICMP INFO request เพื่อที่ว่า ถึงแม้แพ็กเก็ต ICMP ใน Type ECHO ถูกสกัดกั้นไว้ที่เราท์เตอร์ตัวที่เชื่อมต่อออกอินเทอร์เน็ตหรือที่ไฟร์วอลล์ก็ตาม แต่โอกาสที่แพ็กเก็ต ICMP ใน Type อื่นๆจะวิ่งทะลผ่านเข้าไปเพื่อตรวจสอบได้ก็ยังมีอยู่ หรืออาจจะส่งไปด้วยแพ็กเก็ตปลอม (spoofed packet) เพื่อป้องกันการตรวจสอบย้อนกลับไปว่าแพ็กเก็ตถูกส่งมาจากที่ไหน

กล่าวโดยสรุป ขั้นตอนนี้จะทำให้เราสามารถตรวจสอบได้อย่างแท้จริงว่าเครื่องปลายทางใดบ้างที่เราสามารถติดต่อได้โดยตรงผ่านอินเตอร์เน็ต ด้วยการส่งแพ็กเก็ต ICMP ใน TYPE ต่างๆ หรือใช้เทคนิคของการสแกนพอร์ตเข้าไปตรวจสอบ จากลิสต์รายการหมายเลข IP Address ที่อยู่ในคลาส C จำนวนกว่า 200 หมายเลข เราสามารถตรวจสอบได้ว่าแท้จริงแล้วมีเครื่องไหนบ้างที่อยู่ในขอบข่ายของการเป็นเครื่องเป้าหมายได้ เพื่อช่วยลดโฟกัสขอบเขตของการตรวจสอบให้แคบลงและประหยัดเวลามากขึ้น

5.2 ICMP Queries



รูปที่ 5-2 แสดงการทำงานของ ICMP QUERY

คือการส่งแพ็กเก็ตของโพรโตคอล ICMP ไปสอบถาม ตัวอย่างเช่น สอบถามเวลาของเครื่องปลายทาง (เพื่อดูว่าเครื่องๆนั้นอยู่ในโซนเวลา (time zone) แถบใด) ด้วยการส่งแพ็กเก็ต ICMP type TIMESTAMP, สอบถามค่าของ Subnet Mask ของเครื่องปลายทางด้วยการส่งแพ็กเก็ต ICMP type ADDRESS MASK REQUEST Subnet Mask เป็นค่าที่สำคัญพอสมควรเพราะมันจะทำให้คุณทราบหมายเลข Subnet Address ของเครื่องเน็ตเวิร์คเป้าหมาย การทราบค่าของ Subnet Address จะช่วยให้เราสามารถโฟกัสไปที่เน็ตเวิร์คที่ต้องการได้และหลีกเลี่ยงการส่งข้อมูลโดยระบุแอดเดรสปลายทางเป็น Broadcast Address

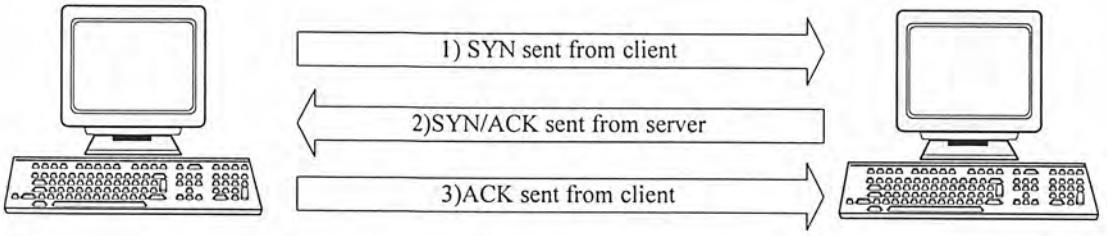
5.3 Port Scanning

การสแกนพอร์ตหรือ Port Scanning เป็นกระบวนการในการคอนเน็กเข้าไปที่ TCP Port หรือ UDP Port ของเครื่องปลายทางเพื่อค้นหาว่ามีเซอร์วิสอะไรบ้างที่ทำงานอยู่หรืออยู่ในสถานะ LISTENING

การค้นหาพอร์ตที่เปิดอยู่เป็นเรื่องสำคัญทีเดียวในการตรวจสอบประเภทของระบบปฏิบัติการและแอปพลิเคชันที่ใช้งานอยู่ในระบบ เพราะ ระบบปฏิบัติการหรือเซอร์วิสที่รันอยู่อาจมีข้อบกพร่องบางอย่างที่อนุญาตให้ผู้ใช้ที่ไม่ได้ผ่านการตรวจสอบเข้าไปในระบบได้ หรือมีข้อบกพร่องเกี่ยวกับระบบการรักษาความปลอดภัยที่เป็นที่รู้กันดี โดยเฉพาะเซอร์วิสบางเวอร์ชันที่ยังไม่สมบูรณ์ เครื่องมือและเทคนิคในการสแกนพอร์ตได้รับการพัฒนาอย่างต่อเนื่องมาหลายปี เทคนิคการสแกนพอร์ตที่เรากล่าวไปข้างต้นนั้น เป็นเพียงการสแกนพอร์ตเพื่อรู้ว่าเครื่องปลายทางเปิดอยู่และติดต่อผ่านทางอินเตอร์เน็ตหรือเครือข่ายได้หรือไม่ เท่านั้น แต่สำหรับเทคนิคการสแกนพอร์ตที่จะกล่าวถึงถัดนี้ไป เป็นการสแกนพอร์ตเพื่อค้นหาว่ามีพอร์ตหมายเลขใดเปิดอยู่ที่เครื่องปลายทางนั้นบ้าง

- TCP connect scan เป็นการคอนเน็กไปที่พอร์ตที่ต้องการบนเครื่องปลายทาง แล้วขอเปิดคอนเน็กชันของโพรโตคอล TCP ด้วยกลไกมาตรฐานที่เรียกว่า TCP three-way handshake

(SYN, SYN/ACK และ ACK) แต่วิธีนี้มักถูกตรวจจับได้โดยง่ายโดยไฟร์วอลล์ที่ไซต์ปลายทาง



รูปที่ 5-3 TCP 's 3-way handshake

รูปแสดงการเปิดคอนเน็กชันของโพรโตคอล TCP จะอาศัยกลไก three way handshake โดยขั้นแรกจะส่งแพ็กเก็ต TCP ที่เซตแฟลก SYN เป็น 1 ไว้ออกไปที่เครื่องปลายทาง ขั้นที่สอง เครื่องต้นทางจะได้รับ TCP SYN/ACK กลับมาและขั้นสุดท้าย จะส่งแพ็กเก็ต TCP ACK กลับไปอีกครั้งเพื่อยืนยันว่าได้รับแพ็กเก็ตในขั้นที่สองแล้ว

- TCP SYN scan เทคนิคนี้บางครั้งเรียกว่า half-open scanning สาเหตุเพราะว่า คอนเน็กชันที่สมบูรณ์ของโพรโตคอล TCP ยังไม่ได้ถูกเปิดขึ้น เพราะเมื่อแพ็กเก็ต TCP ที่เซตค่าแฟลก SYN และ ACK ไว้เป็น 1 (TCP SYN/ACK) นั่นก็เพียงพอแล้วที่จะสรุปว่า พอร์ตดังกล่าวอยู่ในสถานะ LISTENING แต่ถ้าพอร์ตดังกล่าวไม่ได้เปิดอยู่ แพ็กเก็ต TCP ที่เซตค่าแฟลก RST และ ACK ไว้เป็น 1 (TCP RST/ACK) จะถูกส่งกลับมาแทน เทคนิคค่อนข้างน่าใช้กว่าเทคนิคแรกเพราะส่วนใหญ่แล้วไฟร์วอลล์ที่ไซต์ปลายทางมักตรวจจับได้ค่อนข้างยาก
- TCP FIN scan เทคนิคนี้จะส่งแพ็กเก็ต TCP ที่เซตค่าแฟลก FIN เป็น 1 (TCP FIN) ไปยังพอร์ตที่ต้องการ ถ้าไดร์เวอร์ของโพรโตคอล TCP/IP ที่เครื่องปลายทางได้ถูกพัฒนาขึ้นมาโดยมีฟีเจอร์ตามในมาตรฐาน RFC หมายเลข 793 ครบถ้วน (http://www.ietf.org/rfc/rfc0793.txt) เครื่องปลายทางจะส่งแพ็กเก็ต TCP RST ของทุกๆพอร์ตที่ปิดอยู่กลับมาให้ (กล่าวง่ายๆก็คือ เราจะทราบหมายเลขพอร์ตที่ไม่ได้เปิดให้บริการ) โดยปกติแล้ว เทคนิคนี้มักใช้ได้กับเครื่องปลายทางที่รันยูนิกซ์
- TCP Xmas Tree scan เทคนิคนี้จะส่งแพ็กเก็ต TCP ที่เซตแฟลก FIN ,URG และ PUSH ไปยังพอร์ตเป้าหมายที่เครื่องปลายทาง และอาศัยมาตรฐาน RFC 793 อีกเช่นกัน เครื่องปลายทางจะส่งแพ็กเก็ต TCP RST ของทุกพอร์ตที่ปิดอยู่กลับมาให้

- TCP Null scan เทคนิคนี้จะส่งแพ็กเก็ต TCP ออกไปโดยเซตค่าของทุกๆแฟลกให้เป็น 0 หมด และ อาศัยมาตรฐาน RFC 793 อีกเช่นกัน เครื่องปลายทางจะส่งแพ็กเก็ต TCP RST ของทุกๆพอร์ตที่ปิดอยู่กลับมาให้
- TCP ACK scan เทคนิคนี้จะถูกใช้เพื่อค้นหา "rule" และ "policy" ต่างๆที่เซตไว้ที่ไฟร์วอลล์ เพื่อตรวจสอบดูว่าไฟร์วอลล์นั้นๆทำหน้าที่แค่เพียงกรองแพ็กเก็ตอย่างง่ายๆ หรือเป็นไฟร์วอลล์ที่มีความฉลาดพอสมควรและใช้เทคนิคการกรองแพ็กเก็ตขั้นสูง
- TCP Window scan เทคนิคนี้จะตรวจสอบพอร์ตที่เปิดอยู่ รวมทั้งตรวจดูว่า พอร์ตใดบ้างที่ถูกฟิลเตอร์เอาไว้ไม่ให้ผ่านเข้าไปถึง และพอร์ตหมายเลขใดได้รับการอนุญาตไว้บ้าง โดยอาศัยช่องโหว่จากความผิดปกติบางอย่างในการแจ้งค่าของ TCP Window Size ของโพรโตคอล TCP/IP
- TCP RPC scan เทคนิคนี้ใช้งานได้เฉพาะกับเครื่องปลายทางที่รันยูนิกซ์เท่านั้น มันถูกใช้เพื่อตรวจสอบดูว่ามีเซอร์วิสใดทำงานอยู่บนเซอร์วิส RPC บ้าง รวมทั้งตรวจดูเวอร์ชันของเซอร์วิสนั้นและโปรแกรมอื่นที่เกี่ยวข้อง
- UDP scan เทคนิคนี้จะส่งแพ็กเก็ตของโพรโตคอล UDP ไปยังพอร์ตเป้าหมาย ถ้าเครื่องปลายทางตอบกลับมาด้วยแพ็กเก็ต ICMP type PORT UNREACHABLE นั่นหมายความว่าพอร์ตนั้นปิดอยู่ในทางตรงกันข้าม ถ้าเราไม่ได้รับแพ็กเก็ต ICMP type ดังกล่าว เราสามารถสรุปได้ว่าพอร์ตนั้นเปิดอยู่ เนื่องจากโพรโตคอล UDP เป็นโพรโตคอลลักษณะ connectionless คือไม่รับรองว่าแพ็กเก็ตที่ส่งไปจะถึงเครื่องปลายทางครบถ้วนหรือไม่ ดังนั้นความถูกต้องของผลลัพธ์ที่ได้จากเทคนิคนี้ก็อาจขึ้นกับปัจจัยอื่นๆ ด้วยเช่น ปริมาณทราฟฟิกในเน็ตเวิร์คและทรัพยากรบนเครื่องปลายทาง นอกจากนั้นมันยังเป็นเทคนิคที่ค่อนข้างช้าอีกด้วยถ้าคุณกำลังสแกนเน็ตเวิร์กที่ใช้งานไฟร์วอลล์หรือเราท์เตอร์ที่มีการฟิลเตอร์กรองแพ็กเก็ต จึงขอให้เตรียมใจไว้ด้วยกับผลลัพธ์ที่ไม่คาดคิดของ UDP scan

ไดร์เวอร์ของโพรโตคอล TCP/IP ของระบบปฏิบัติการบางตัวอาจส่งแพ็กเก็ต TCP RST กลับไปยังเครื่องต้นทางตลอกไม่ว่าพอร์ตๆ นั้นจะเปิดหรือปิดอยู่ ดังนั้นจึงเป็นไปได้ว่า ผลลัพธ์ที่ได้อาจแตกต่างกันไปไม่แน่นอน แต่อย่างไรก็ดี การตรวจสอบด้วยแพ็กเก็ต TCP SYN ธรรมดานั้นจะทำงานได้ดีสำหรับทุกๆ โฮสต์

5.4 การตรวจหาประเภทของระบบปฏิบัติการ

จุดประสงค์ประการแรกของการสแกนคือ การค้นหา TCP port และ UDP port ที่เปิดอยู่บนเครื่องปลายทาง ส่วนจุดประสงค์ประการที่สองคือ การค้นหาประเภทของระบบปฏิบัติการที่รันอยู่บนเครื่องหมายปลายทาง ประเภทของระบบปฏิบัติการที่เราทราบจะถูกนำไปใช้ประโยชน์ในขั้นตอนการค้นหารายละเอียดต่างๆ (enumeration) โดยอาศัยข้อบกพร่องต่างๆ ที่แต่ละระบบปฏิบัติการนั้นมี ดังนั้นเราจึงต้องมั่นใจว่า ระบบปฏิบัติการที่เราทราบนั้นเป็นข้อมูลที่ถูกต้อง

ส่วนใหญ่เซิร์ฟเวอร์ที่ให้บริการมาตรฐานบนอินเตอร์เน็ตเช่น เว็บเซิร์ฟเวอร์, เมล์เซิร์ฟเวอร์, FTP Server นั้น แอพพลิเคชั่นที่ให้บริการพอจะสามารถบอกเราได้ในระดับหนึ่งว่าใช้ระบบปฏิบัติการอะไร เช่น IIS จะต้องทำงานอยู่บน Window NT หรือ Window2000 , Apache จะต้องรันอยู่บน Unix หรือ Linux ซึ่งหากแฮกเกอร์สามารถได้ข้อมูลเหล่านี้มาก็จะสามารถทราบได้เองว่าเป้าหมายใช้ระบบปฏิบัติการอะไรอยู่โดยไม่ต้องแสกนให้ยุ่งยาก แต่ก็มีหลายกรณีเช่นกันที่ไม่สามารถสืบค้นระบบปฏิบัติการของเป้าหมายได้ว่าเป็นระบบปฏิบัติการอะไร โดยอาศัยการตรวจสอบดูเฉพาะแอพพลิเคชั่นเพียงอย่างเดียว ดังตัวอย่างต่อไปนี้

5.4.1 มีแอพพลิเคชั่นมาตรฐานทำงานอยู่ แต่แอพพลิเคชั่นได้ถูกปรับแต่งมาอย่างดีไม่ให้บอกรายละเอียดของตนเองแก่ผู้อื่น คือมีหน้าที่ให้บริการอย่างเดียว โดยทั่วไปแล้วแอพพลิเคชั่นเซิร์ฟเวอร์ที่ทำงานอยู่ เมื่อมีไคลเอนต์ติดต่อขอรับบริการเข้ามา เซิร์ฟเวอร์ก็จะส่งข้อความให้ทราบว่าตนเองชื่ออะไร เวอร์ชั่นไหน ให้ไคลเอนต์ทราบเสียก่อน(จะตรวจสอบได้ง่ายด้วยการ Telnet ไปที่พอร์ตของแอพพลิเคชั่นดังกล่าว) ข้อมูลเหล่านั้นเรียกว่าแบนเนอร์(banner) ซึ่งเปรียบเสมือนป้ายประกาศที่จะประชาสัมพันธ์ให้ผู้ใช้ทราบทุกครั้งว่าขณะนี้กำลังใช้งานเซิร์ฟเวอร์ของผู้ผลิตรายใหนอยู่ เผื่อผู้ใช้ที่ฝั่งไคลเอนต์ประทับใจจะได้นำไปบอกต่อได้ และผู้ผลิตซอฟต์แวร์เหล่านี้ทุกรายก็จะมีแบนเนอร์ติดมากับซอฟต์แวร์เหล่านี้เสมอแต่สำหรับแอพพลิเคชั่นที่ถูกปรับแต่งมาอย่างดีนั้นจะไม่แสดงตัวและให้ข้อมูลเหล่านี้ออกมา เช่น มีเซิร์ฟเวอร์ที่ทำงานอยู่สามารถให้บริการแก่บราวเซอร์ได้ตามปรกติ แต่บราวเซอร์อาจไม่ได้รับข้อมูลใดๆจากเซิร์ฟเวอร์เลยว่าเป็นแอพพลิเคชั่นของใครยี่ห้ออะไร เวอร์ชั่นอะไร รู้เพียงแต่ว่าเป็นเซิร์ฟเวอร์ที่อาจเป็นได้ทั้ง Apache, IIS, Netscape หรือ เว็บเซิร์ฟเวอร์อื่นใดก็ได้ เมื่อไม่รู้ว่าแอพพลิเคชั่นมีรายละเอียดอย่างไรก็ย่อมทำให้ไม่สามารถรู้ถึงระบบปฏิบัติการด้วยเช่นกัน

5.4.2 ไม่มีแอพพลิเคชั่นมาตรฐานทั่วไปรันอยู่ หรือไม่มีการให้บริการใดๆ กรณีนี้เป้าหมายอาจเป็นเซิร์ฟเวอร์ธรรมดาที่ใช้งานเฉพาะอย่างและไม่ได้ให้บริการมาตรฐานทั่วๆไปบนอินเตอร์เน็ตเมื่อไม่มีการให้บริการก็ย่อมไม่มีช่องทางในการสอบถามข้อมูลจากการบริการเหล่านั้นได้เซิร์ฟเวอร์ดังกล่าวย่อมเป็นเหมือนกล่องดำบนเน็ตเวิร์กที่ไม่มีใครทราบว่าเป็นเซิร์ฟเวอร์อะไร, ระบบปฏิบัติการอะไร, ให้บริการอะไร เซิร์ฟเวอร์ประเภทนี้อาจเป็นที่สะดุดตาของแฮกเกอร์ทั้งหลายได้ เพราะดูเหมือนมีความลับอยู่ภายใต้การป้องกันที่แข็งแรง

เมื่อไม่สามารถหาข้อมูลของเปาหมายด้วยวิธีทั่วไปได้เนื่องจากติดขัดด้วยข้อจำกัดทั้ง 2 ประการข้างต้นนั้น จึงต้องใช้วิธีแสกนเพื่อให้ได้มาซึ่งข้อมูลเหล่านี้ การแสกนเซิร์ฟเวอร์เพื่อตรวจสอบดูว่าเป็นระบบปฏิบัติการอะไร เวอร์ชั่นไหนเป็นการใช้ข้อบกพร่องของโปรโตคอลที่มิได้มีการกำหนดชัดเจนครอบคลุมทุกเงื่อนไขของการสื่อสารข้อมูล ดังนั้นเมื่อเหตุการณ์ดังกล่าวเกิดขึ้น จึงไม่มีมาตรฐานที่ทุกคนต้องยึดร่วมกัน การตอบสนองของระบบปฏิบัติการแต่ละชนิดจึงแตกต่างกันออกไปขึ้นอยู่กับผู้ผลิตระบบปฏิบัติการนั้นจะอิมพลีเมนต์ TCP/IP อย่างไร และแน่นอนว่าระบบปฏิบัติการแต่ละรุ่นก็มีลักษณะ

การตอบสนองแตกต่างกันออกไป การแสกนด้วยเทคนิคนี้ส่วนใหญ่จะกระทำบนโปรโตคอล TCP เนื่องจากมีส่วนที่ไม่ได้กำหนดไว้ในโปรโตคอลอยู่ค่อนข้างมาก นั่นคือ TCP Flag เพราะ TCP Flag เป็นส่วนสำคัญที่ใช้ในการควบคุมการสื่อสารของ TCP แต่ Flag เหล่านั้นก็ไม่ได้ถูกนำไปใช้งานทุกๆเงื่อนไข จึงมีFlag บางเงื่อนไขซึ่งไม่มีโอกาสเกิดขึ้นได้จริงในการทำงานปรกติ และ Flag ที่ไม่ใช่ Flag ปรกติ และไม่ได้มีกำหนดไว้ในโปรโตคอลโดยเฉพาะโดยการสลับ Flag ของ TCP ให้เป็นค่าต่างๆ ที่ไม่มีการใช้งาน หลักการสำคัญก็คือระบบปฏิบัติการแต่ละชนิดแต่ละเวอร์ชั่นก็จะตอบสนองแพ็กเก็ตลักษณะนี้แตกต่างกันออกไป และเมื่อรวบรวมผลลัพธ์ที่ตอบมาจากระบบปฏิบัติการแต่ละชนิดแล้วจะพบว่าข้อมูลเหล่านี้สามารถบอกถึงระบบปฏิบัติการได้เป็นอย่างดี

Impossible TCP Flags กับ OS Fingerprint

ก่อนที่จะดูถึงรายละเอียดการแสกนจากโปรแกรมที่ใช้เทคนิคของ Flag นี้ขอให้ศึกษาถึงลักษณะของ Flag ปรกติ และ Flag ที่ไม่ปรกติของ TCP เสียก่อน การที่เราแบ่งแยก Flag ของ TCP ได้เป็น 2 ชนิด เนื่องจากในโปรโตคอลจะใช้ Flag เป็นสัญลักษณ์ที่บอกถึงลักษณะและจังหวะของข้อมูล TCP เซกเมนต์นั้นๆจะมี Flag ในลักษณะใดลักษณะหนึ่งตามที่กำหนดอยู่ในโปรโตคอลซึ่งทำให้ผู้ได้รับข้อมูลนั้นๆจะมี Flag ในลักษณะใดลักษณะหนึ่งตามที่กำหนดอยู่ในโปรโตคอลซึ่งทำให้ผู้ได้รับข้อมูลนั้นๆสามารถตอบสนองเป็นมาตรฐานเดียวกัน Flag ประเภทนี้จะเป็น Flag ปรกติแต่ Flag บางประเภทนั้นหากการสื่อสารยึดถือตามโปรโตคอลอย่างถูกต้องแล้ว จะไม่มีโอกาสเกิดขึ้นได้เลย เช่น SYN, Fin คือในแพ็กเก็ตเดียวกันมีทั้ง 2 อย่างซึ่งจะไม่มีโอกาสเกิดขึ้นได้อย่างเด็ดขาด ไม่ว่าในกรณีใดๆของ TCP แต่เนื่องจากแฮกเกอร์ได้ทำการส่งข้อมูลมายังเลเยอร์ล่างโดยตรงไม่ผ่านเลเยอร์ของ TCP ซึ่งด้วยวิธีการนี้ไม่ว่า Flag ใดๆ ก็ส่งออกมาได้เสมอ

Flags	Open port	Closed port
None	0	RA
F	0	RA
S	SA	RA
SF	SFA	RA
R	0	0
RF	0	0
SR	0	0
SRF	0	0
A	R	R
FA	R	R
SA	R	R
SFA	R	R

RA	0	0
RFA	0	0
SRA	0	0
SFRA	0	0

ตารางที่ 5-1 TCP Flags Combination

จากตาราง 5-1 แสดงทุกๆค่าที่เป็นไปได้ในการรวมกัน (Combination) ของ TCP Flag พร้อมด้วยการตอบรับต่อ Flag เหล่านั้นทั้งจากพอร์ตที่เปิดและพอร์ตที่ปิดซึ่งในการสื่อสารของ TCP จริงๆ นั้นจะไม่ได้ใช้งาน Flag ครบทุกกรณี ส่วนที่เหลือนั้นคือ Flag ที่ไม่มีโอกาสเกิดขึ้นในการสื่อสารจริง หรือเป็นไปไม่ได้ หารเป็นการสื่อสารผ่านโปรโตคอลแล้ว โปรโตคอลจะไม่อนุญาตให้ Flag เหล่านี้เกิดขึ้นได้โดยเด็ดขาด และแน่นอนว่าผู้ที่ทำการสื่อสารข้อมูลตามปรกติในฐานะผู้ใช้ซึ่งต้องสื่อสารข้อมูลผ่านเลเยอร์ต่างๆของโปรโตคอล ก็ไม่มีทางจะสร้างแพ็กเก็ตที่ออกนอกลู่นอกทางจากโปรโตคอลที่กำหนดได้ แต่นั่นมิได้หมายความว่าแฟลกเหล่านั้นไม่มีโอกาสเกิดขึ้นได้โดยสิ้นเชิง ในเมื่อผู้ใช้นั่นคือแฮกเกอร์ซึ่งเป็นผู้ควบคุมโฮสต์ของตัวเองโดยสมบูรณ์ก็ย่อมสามารถจะส่งข้อมูลไปยังโปรโตคอลในระดับดาต้าลิงค์ลเยอร์ คือโปรแกรมควบคุมฮาร์ดแวร์ระดับล่างซึ่งอยู่ต่ำกว่า TCP/IP ให้ส่งข้อมูลใดๆก็ได้โดยไม่ต้องติดขัดกับขีดจำกัด และเงื่อนไขต่างๆที่โปรโตคอลระดับบนกำหนดขึ้น เพียงเท่านี้โปรโตคอลระดับบนไม่ว่า IP, ICMP, UDP และ TCP จะกำหนดข้อห้ามไว้อย่างไรก็ไม่มีความหมาย

จากตารางข้างต้นจะเห็นได้ว่ามีการผสมกันของ Flag ที่เป็นไปไม่ได้ (Impossible Flags) อยู่หลายประเภท และแฟลกเหล่านี้เองที่โปรโตคอลไม่ได้บัญญัติไว้ว่าโฮสต์ที่ได้นับจะต้องตอบสนองอย่างไร (อาจจะคิดว่าอย่างไรเสียแฟลกเหล่านี้ก็ไม่มีโอกาสเกิดขึ้นได้อยู่แล้ว จึงไม่มีการกำหนดการตอบรับไว้ในโปรโตคอลอย่างชัดเจน) เมื่อมีการส่งแพ็กเก็ตเหล่านี้ไปยังโฮสต์ซึ่งมีระบบปฏิบัติการรับผิดชอบการสื่อสารข้อมูล แน่นอนว่าทุกๆระบบปฏิบัติการต้องมีการตอบสนองอย่างใดอย่างหนึ่ง เพียงแต่ไม่มีมาตรฐานให้ยึดถือว่าควรจะตอบอย่างไร (การไม่ส่งแพ็กเก็ตใดๆ ถือเป็นการตอบสนองอย่างนึงโดยการไม่ตอบ) และไม่ว่าจะตอบรับอย่างไรก็ล้วนแต่ไม่มีผลต่อการทำงานในสภาวะปกติทั้งสิ้น การตอบในส่วนนี้เป็นส่วนที่ทุกๆคนมองข้ามไปและนอกเหนือความคาดหมาย อาจมีระบบปฏิบัติการบางค่ายที่ให้ความสนใจเรื่องนี้ก็ให้ตอบด้วยแพ็กเก็ตต่างๆอย่างจงใจ แต่อาจมีบางระบบปฏิบัติการบางระบบที่การตอบสนองต่อแพ็กเก็ตเหล่านี้เป็นไปตามยถากรรม คือมิได้มีการกำหนดเงื่อนไขของแฟลกเหล่านี้ไว้ แต่การตอบสนองที่เกิดขึ้นเป็นเพราะแฟลกเหล่านั้นไม่เข้าเงื่อนไขใดๆเลย ผลที่ออกมาจากการถูกกระตุ้นโดยอิมพอสสิเบิลบางครั้งแม้แต่ผู้ผลิตระบบปฏิบัติการเองก็ยังไม่เคยทดสอบ และไม่ทราบด้วยซ้ำว่าระบบปฏิบัติการของตัวเองจะตอบอย่างไร

การตอบสนองที่เกิดขึ้นของอิมพอสสิเบิลแฟลกเหล่านี้ไม่ว่าผู้ผลิตจะตั้งใจให้เป็นเช่นนั้นหรือไม่ก็ตาม แต่การตอบสนองทุกๆครั้งก็จะเป็นเช่นนั้นอยู่เสมอ (เนื่องจากระบบปฏิบัติการก็เป็นโปรแกรมคอมพิวเตอร์ประเภทหนึ่งที่ทำงานตามเงื่อนไขของโปรแกรมที่ถูกกำหนดไว้ หากเงื่อนไขแบบหนึ่งโปรแกรมก็ตอบสนองแบบหนึ่ง และเงื่อนไขที่กำลังสนใจนี้คือแฟลก หากแฟลกเปลี่ยนไปการตอบสนอง

ก็จะเปลี่ยนไปอีกแบบหนึ่ง และเมื่อไรก็ตามที่เงื่อนไขเหมือนเดิม การตอบสนองของโปรแกรมก็จะยังคงเหมือนเดิม) คือไม่ว่าแฟลกที่ส่งเข้าไปจะแปลกประหลาดอย่างไร เมื่อไรการตอบสนองทุกๆครั้งก็จะออกมาเหมือนเดิม ยกตัวอย่างเช่นระบบปฏิบัติการ SunOS 4.13 เมื่อได้รับ TCP แพ็กเก็ตที่มีแฟลกเป็น SFRA มาที่พอร์ตที่ปิดอยู่ระบบปฏิบัติการก็จะไม่ตอบอะไรกลับไปเลย และหากได้รับมาที่พอร์ตที่เปิดอยู่ก็จะไม่ตอบสนองอะไรกลับไปเลยเช่นกัน ดังนั้นที่มีการส่งแพ็กเก็ตที่มีแฟลก SFRA ไปยัง SunOS 4.13 ก็จะได้รับการตอบสนองเช่นนี้อยู่เสมอ

สำหรับการตอบกลับมาของอิมพอสสิเบิลแฟลกเพียงแฟลกเดียวเป็นการยากที่จะชี้ชัดว่ามาจากระบบปฏิบัติการอะไร Linux Redhat 7.0 อาจจะตอบสนองต่อ SFRA แฟลกเช่นเดียวกับ SunOS 4.13 ก็เป็นได้ ดังนั้นเพื่อให้ชี้ชัดได้ถึงระบบปฏิบัติการได้อย่างแม่นยำที่สุดจึงต้องทดสอบโดยอิมพอสสิเบิลแฟลกหลายชนิด และดูผลการตอบรับทั้งหมด จึงมั่นใจได้ว่าไม่ว่าระบบปฏิบัติการชนิดใดเมื่อได้รับชุดของอิมพอสสิเบิลแฟลกเหล่านี้จะมีผลการตอบรับที่แตกต่างออกไปไม่ซ้ำกัน และหากนำผลลัพท์ที่ได้ไปเปรียบเทียบแล้วสามารถระบุได้ทันทีว่าเป็นการตอบสนองจากระบบปฏิบัติการใด

แพทเทิร์นการตอบกลับมาจากกการกระตุ้นหลายๆแบบของอิมพอสสิเบิลแฟลกที่เป็นเอกลักษณ์ของระบบปฏิบัติการแต่ละชนิดที่เป็นเสมือน ลายนิ้วมือของระบบปฏิบัติการ (OS Fingerrint) ไว้อ้างอิงว่าระบบปฏิบัติการอะไร เวอร์ชั่นใด ซับเวอร์ชั่นใด มีลายนิ้วมือเป็นอย่างไร

โปรแกรมที่ใช้สำหรับการทดสอบและแสกนหาระบบปฏิบัติการนี้ใช้หลักการที่เรียบง่ายโดยอาศัยฐานข้อมูลลายนิ้วมือที่เก็บจากระบบปฏิบัติการต่างๆที่มีอยู่บนโลกมาใช้เป็นฐานในการอ้างอิง หลังจากนั้นก็ทำการส่งชุดของแพ็เก็ตที่มีอิมพอสสิเบิลไปทดสอบยังโฮสต์เป้าหมายแล้วนำผลการตอบรับมาเปรียบเทียบกับลายนิ้วมือที่มีอยู่ในฐานข้อมูล หากตรงกับระบบปฏิบัติการอะไรก็ให้สันนิฐานได้ว่าระบบปฏิบัติการที่ทำงานที่โฮสต์เป้าหมายนั้นเป็นระบบปฏิบัติการตามที่ระบุในฐานข้อมูลนั่นเอง ความแม่นยำของการทดสอบหาระบบปฏิบัติการด้วยวิธีนี้จะขึ้นอยู่กับจำนวนของแพ็กเก็ตที่ใช้ทดสอบ ยิ่งมีการทดสอบหลายแพ็กเก็ตผลการตอบรับจะยิ่งชี้ไปยังระบบปฏิบัติการรุ่นใดรุ่นหนึ่งได้อย่างแม่นยำมากขึ้น ส่วนอีกปัจจัยหนึ่งคือความทันสมัยของการปรับปรุงฐานข้อมูลลายนิ้วมือให้ทันต่อระบบปฏิบัติการเวอร์ชั่น ใหม่ๆอยู่เสมอ

5.4.1 Active Stack Fingerprinting

Stack fingerprinting เป็นเทคโนโลยีที่ช่วยทำให้มั่นใจได้ว่าประเภทของระบบปฏิบัติการที่ค้นหาได้นั้นเป็นระบบปฏิบัติการที่ถูกต้อง มีเปอร์เซ็นต์ความน่าเชื่อถือสูง โดยอาศัยหลักการที่ว่า ผู้ผลิตระบบปฏิบัติการแต่ละรายมักพัฒนาไดรเวอร์ของโพรโตคอล TCP/IP และเซอร์วิสที่ทำงานบนโพรโตคอล TCP/IP ให้มีเอกลักษณ์เฉพาะตัวเป็นของตัวเอง ซึ่งมักแตกต่างกับของระบบปฏิบัติการอื่น

ดังนั้น โดยการตรวจวัดความแตกต่างเหล่านี้ เราจะสามารถเริ่มคาดเดาได้อย่างมีเหตุผล แต่เพื่อให้มีความน่าเชื่อถือสูงสุด Stack fingerprinting จำเป็นต้องอาศัยการคอนเน็กไปพอร์ตที่เปิดอยู่อย่างน้อยนึ่งพอร์ต แต่ การทำงานของโครงงานนี้ สามารถคาดเดาได้อย่างมีเหตุผล ถึงแม้ว่าจะไม่ได้คอนเน็กไปที่พอร์ตใดเลย แต่ผลที่ได้ก็อาจยังไม่น่าเชื่อถือนัก โดยให้สำรวจวิธีการตรวจวัดความแตกต่างกันทีละวิธี

- Fin probe ใช้การส่งแพ็กเก็ตของโพรโตคอล TCP โดยเซตแฟลก Fin ให้เป็น 1 ไว้ ตามมาตรฐาน RFC 793 ระบุไว้ว่า พฤติกรรมที่ถูกต้องจะต้องไม่มีการส่งแพ็กเก็ตอะไรตอบสนองกลับไป แต่ทว่า ไดรเวอร์ของโพรโตคอล TCP/IP ในระบบปฏิบัติการบางตัว อย่างเช่น วินโดว์เอ็นที จะตอบสนองกลับมาด้วยแพ็กเก็ตของโพรโตคอล TCP ที่เซตแฟลก Fin และ ACK ให้เป็น 1 ซึ่งเครื่องมือส่วนมากได้นำเทคนิคนี้ไปใช้กัน
- Bogus Flag probe ใช้การส่งแพ็กเก็ตของโพรโตคอล TCP โดยเซตแฟลก SYN ให้เป็น 1 พร้อมทั้งเซตบิตตำแหน่งที่ยังไม่ได้ใช้งานให้เป็น 1 ด้วย บางระบบปฏิบัติการเช่น ลีนุกซ์ จะตอบสนองกลับมาด้วยการเซตแฟลกบางตัวในแพ็กเก็ต ว่ามักเป็นค่าอะไร อย่างไรก็ตามบางระบบปฏิบัติการจะทำการ reset connection เมื่อได้รับแพ็กเก็ตนี้
- TCP Initial Sequence Number (ISN) sampling ใช้วิธีการตรวจหาแพทเทิร์นของ Sequence number ที่อยู่ในส่วนหัวของแพ็ตเก็ตที่ได้รับการตอบสนองต่อการร้องขอการเชื่อมต่อ ว่าเป็นค่าอะไร ซึ่งสามารถแบ่งได้หลายกลุ่มเช่น ถ้าเป็นแบบ traditional 64K จะพบใน UNIX รุ่นเก่า, แบบ Random increments จะพบใน Solaris, IRIX, FreeBSD, Digital UNIX, Cray ในเวอร์ชันใหม่, แบบ True "random" จะพบใน Linux 2.0.*, OpenVMS, AIX เวอร์ชันใหม่, ส่วน Windows จะเป็นแบบ "time dependent" model คือค่าของ ISN จะเพิ่มขึ้นจำนวนหนึ่งที่กำหนดไว้ในแต่ละช่วงเวลา, และในบางครั้งจะใช้ค่า ISN ค่าเดียวไม่เปลี่ยนเลย เช่นใน 3Com hubs จะใช้เป็น 0x803 ตลอดและใน Apple LaserWriter printers จะใช้เป็น 0xC7001
- "Don't fragment bit" monitoring บางระบบปฏิบัติการได้เซตบิต "Don't fragment bit" ไว้เพื่อเพิ่มความเร็วในการส่งข้อมูล ให้มอนิเตอร์บิตนี้เพื่อตรวจดูว่าระบบปฏิบัติการไหนเซตบิตนี้บ้าง
- TCP initial window size ใช้วิธีดูขนาดของ TCP window เพราะบางระบบปฏิบัติการจะมีการล็อคค่าไว้เลยว่าขนาดของ TCP window เป็นเท่าไหร่ (เช่น AIX เป็นเพียงระบบปฏิบัติการเดียวที่มีขนาดของ window เท่ากับ 0x3F25 ส่วน window NT มีขนาด 0x402E) ค่านี้มักเป็นค่าเฉพาะตัวของแต่ละระบบปฏิบัติการด้วย จึงยิ่งทำให้ผลที่ได้มีความถูกต้องมากขึ้น
- ACK value ระบบปฏิบัติการแต่ละระบบมักเซตค่าในฟิลด์ ACK ไม่เหมือนกัน บางระบบเซตค่าฟิลด์ ACK ให้สอดคล้องตามค่าของฟิลด์ SYN ที่เซตไว้ในฝั่งผู้ส่ง บางระบบจะเซตค่าฟิลด์ ACK ให้บวกจากค่าของฟิลด์ SYN ไปอีกหนึ่ง
- ICMP error message quenching บางระบบปฏิบัติการอาจปฏิบัติตามมาตรฐาน RFC 1812 และมีการจำกัดอัตราการส่งข้อความแจ้งข้อผิดพลาดดังนั้น (ใน Linux kernel จะจำกัดการส่งข้อความแจ้งข้อผิดพลาดไปยังปลายทางที่อัตรา 80 แพ็กเก็ตต่อ 4±0.25 วินาที) โดยการส่งแพ็กเก็ตของโพรโตคอล UDP ไปยังหมายเลขพอร์ตสูงๆ แล้วค่อยนับจำนวนของ ICMP type PORT UNREACHABLE Message ที่ตอบกลับมาภายในช่วงเวลาที่กำหนด
- ICMP message quoting เมื่อพบข้อผิดพลาดเกี่ยวกับโพรโตคอล TCP/IP ระบบปฏิบัติการแต่ละประเภทจะให้ข้อมูลและสาเหตุต่างๆมาในแพ็กเก็ตของ ICMP ไม่เท่ากัน บางระบบจะให้รายละเอียดมากบางระบบจะให้รายละเอียดน้อย (ใน Solaris จะส่งกลับมา a bit more และใน

Linux จะส่งกลับมา even more than that) ดังนั้นโดยการสำรวจข้อมูลที่ได้มานี้ เราพอจะใช้ในการคาดเดาประเภทของระบบปฏิบัติการได้

- ICMP error message-echoing integrity บางระบบปฏิบัติการอาจดัดแปลงส่วนหัวของแพ็กเก็ต IP เมื่อมีการส่ง ICMP error message ด้วยการตรวจสอบลักษณะของการดัดแปลงดังกล่าวนี้คุณสามารถนำมันมาใช้ในการคาดเดาได้
- Type of service (TOS) ให้คุณสำรวจฟิลด์ที่ชื่อ Type of service ที่อยู่ในแพ็กเก็ต ICMP ประเภท port unreachable ซึ่งโดยปกติหลายๆระบบปฏิบัติการจะใช้ค่าศูนย์ แต่บางระบบอาจใช้ค่าอื่น เช่น Linux ใช้ 0xC0
- Fragmentation handling แต่ละระบบปฏิบัติการจะจัดการกับแพ็กเก็ตที่ถูกแฟรกเมนต์หรือหั่นซอยไม่เหมือนกัน เมื่อมีการประกอบแพ็กเก็ตย่อยขึ้นมาเป็นแพ็กเก็ตที่สมบูรณ์ บางระบบจะเขียนทับ

  แพ็กเก็ตย่อยอันเก่าด้วยแพ็กเก็ตย่อยอันใหม่หรือบางระบบก็ทำในทางตรงกันข้าม โดยการสังเกตพฤติกรรมตรงนี้ เราพอจะใช้ในการคาดเดาประเภทของระบบปฏิบัติการได้เช่นเดียวกัน สามารถหาข้อมูลเพิ่มเติมได้ที่ www.secnet.com
- TCP options ได้ถูกกำหนดไว้ในมาตรฐาน RFC 793 และได้รับการปรับปรุงในมาตรฐาน RFC 1323 ในปัจจุบัน ผู้ผลิตหลายรายได้มีการอิมพลีเมนต์ออปชั่นพิเศษที่อยู่ใน RFC 1323 ไว้ในระบบปฏิบัติการของตน ดังนั้น ด้วยการส่งแพ็กเก็ตที่เซตออปชั่นหลายๆออปชั่นไปทดสอบอย่างเช่น no operation, maximum segment size, window scale factor และ timestampsมันเป็นไปได้ที่เราจะตั้งสมมติฐานเกี่ยวกับระบบปฏิบัติการที่รันที่เครื่องเป้าหมาย

  <u>ยกตัวอย่าง</u>

  Window Scale=10; NOP; Max Segment Size = 265; Timestamp; End of Ops;

  บางระบบปฏิบัติการ เช่น FreeBSD จะ support ทุกออปชั่นข้างต้นขณะที่ Linux 2.0.X จะ support บางออปชั่น Linux 2.1.x support ทุกออปชั่น

แม้ว่าหลายระบบปฏิบัติการจะ support ออปชั่นเดียวกันแต่บางทีเราก็สามารถแยกแยะได้โดยจากออปชั่น the_values_of เช่น ถ้าส่งค่า MSS เล็กๆไปที่ Linux โดยทั่วไปแล้วมันจะส่ง MSS echo กลับไป ส่วนระบบปฏิบัติการอื่นจะส่งค่าที่แตกต่างกันกลับไป แต่ถ้ายังได้ผลลัพธ์เหมือนกันอีกก็สังเกตลำดับของออปชั่นที่ได้รับมา เช่น Solaris จะเป็น 'NNTNWME' ซึ่งหมายความว่า <no op><no op><timestamp><no op><window scale><echoed MSS> ขณะที่ Linux 2.1.122 จะเป็น MENNTNW ซึ่งจะเห็นได้ว่าเป็นออปชั่นเดียวกัน ส่ง MSS echo เหมือนกัน แต่ลำดับที่ส่งมาไม่เหมือนกัน

- SYN Flood Resistance บางระบบปฏิบัติการจะไม่ยอมรับการ connection ครั้งใหม่ ถ้ามีการส่งแพ็กเก็ต SYN ไปมากเกิน ซึ่งหลายระบบปฏิบัติการนั้นสามารถที่จะจัดการกับแพ็กเก็ตที่ส่งเข้ามาได้ไม่เกินครั้งละ 8 แพ็กเก็ต แต่ใน kernelของ Linux รุ่นใหม่มีวิธีในการป้องกันปัญหานี้เช่น

การใช้ SYN cookies ดังนั้นเราก็สามารถรู้ได้ว่าเป็นระบบปฏิบัติการอะไรโดยการส่งแพ็กเก็ตไป 8 แพ็กเก็ตไปยังพอร์ตที่เปิดอยู่แล้วดูว่าสามารถ connect ไปยังพอร์ตนั้นได้หรือไม่

5.5 นัลเซสชั่น(Null session)

เนื่องจากระบบปฏิบัติการ วินโดวส์ 2000 และ เอ็นที มีจุดอ่อนอยู่ที่การพึ่งพาโปรโตคอล CIFS/SMB และโปรโตคอล NetBIOS ทั้งนี้เนื่องจาก CIFS/SMB และ NetBIOS นั้นเปิดโอกาสให้มีการสอบถามข้อมูลต่างๆของระบบผ่านทาง TCP port หมายเลข 139 ได้โดยที่ไม่จำเป็นต้องล็อกออนเข้าสู่ระบบเลย ขั้นตอนแรกในประโยชน์จากจุดอ่อนนี้ก็คือ ให้สร้างคอนเน็กชั่นไปยัง NT/2000 โดยไม่ต้องล็อกออนด้วยการเปิดคอนเน็คชั่นด้วยแบบที่เรียกว่า "null session" ดังนั้นสมมุติว่า เราทราบผลลัพท์ของการแสกนพอร์ตว่า TCP port 139 ได้เปิดอยู่ ให้ทำขั้นตอนต่อไปนี้

net use \\161.246.5.45\IPC$ ""/u:""

คำสั่งข้างต้นนี้เป็นการคอนเน็กไปที่แชร์พิเศษที่เปิดไว้สำหรับการสื่อสารระหว่างโปรเซส (Interprocess Communication : IPC) บนเครื่องที่มีหมายเลข แอดเดรส 161.246.5.45 โดยใช้แอคเคาท์ Anonymous User พร้อมด้วยรหัสผ่านเป็น null ("") ซึ่งถ้าสำเร็จ ก็เท่ากับว่าได้เปิดแชนแนลไว้เพื่อเตรียมดึงข้อมูลต่างๆ โดยข้อมูลที่เป็นไปได้ก็อย่างเช่น ข้อมูลเกี่ยวกับเน็ตเวิร์ก ชื่อแชร์โฟลเดอร์ ชื่อแอคเคาท์ของผู้ใช้และค่าต่างๆ ในรีจิสตรี้ของวินโดวส์ เป็นต้น

5.6 ตัวอย่างเครื่องมือที่ใช้ในการสำรวจระบบเครือข่าย

เราได้ทำการคัดเลือกโปรแกรมที่มีผู้ใช้ นิยมใช้ในปัจจุบันมาเปรียบเทียบกันในเรื่องฟังก์ชั่นการทำงาน ประสิทธิภาพการทำงานดังนี้

5.6.1. Tivoli NetView 6.0

แหล่งที่มา http://www-3.ibm.com/software/tivoli/products/netview/

ระบบปฏิบัติการที่สนับสนุน

AIX , Linux , Solaris , Windows NT , Windows 2000

5.6.2. 3Com(R) Network Supervisor 3.0

แหล่งที่มา http://www.3com.com/3ns/

ระบบปฏิบัติการที่สนับสนุน

Windows95 , Windows98 , WindowsNT4 , Windows2000 , WindowsXP

5.6.3. GFI LANguard Scanner 3.2

แหล่งที่มา http://www.gfi.com/lannetscan/

ระบบปฏิบัติการที่สนับสนุน

Windows2000/XP

5.6.4. Alchemy Eye 4.9.7

แหล่งที่มา http://www.alchemy-lab.com/products/eye/

ระบบปฏิบัติการที่สนับสนุน

Windows NT/2000/XP

5.6.5. WhatUp Gold 8.0

แหล่งที่มา http://www.ipswitch.com/Products/network-management.html

ระบบปฏิบัติการที่สนับสนุน

Windows 98/ME/NT/2000/XP/Server 2003

5.6.6. OptiView Console Viewer 6.0

แหล่งที่มา

http://www.flukenetworks.com/us/LAN/Monitoring+Analysis+Diagramming/OptiView+Console/Overview.htm

ระบบปฏิบัติการที่สนับสนุน Windows NT/2000/XP

บทที่ 6
การออกแบบและพัฒนาโปรแกรม

6.1 รายละเอียดของการพัฒนา

ในการจัดทำโปรแกรมสำรวจและสังเคราะห์เครือข่ายและระบบคอมพิวเตอร์จำเป็นต้องศึกษาทฤษฎีและข้อมูลที่ใช้ในการค้นหาและรวบรวมข้อมูลของคอมพิวเตอร์เครื่องอื่นที่อยู่ในเครือข่าย โดยในโปรแกรมของเราใช้เทคนิคหลายๆอย่าง ดังนี้

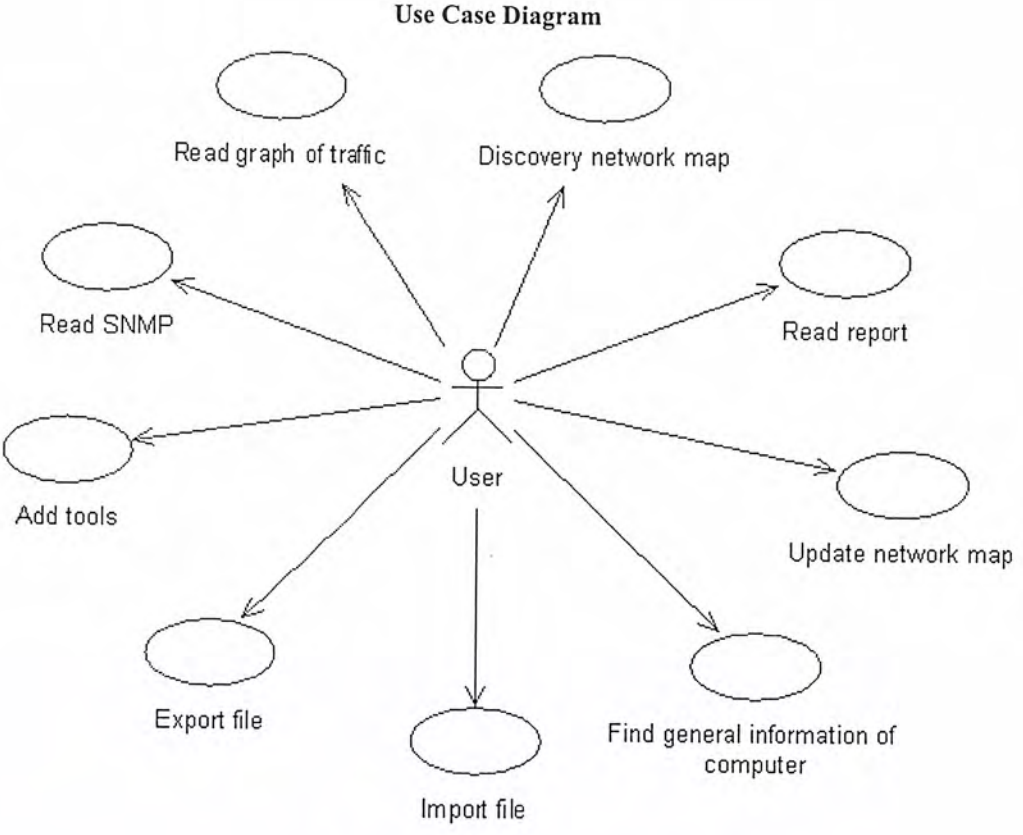
- ใช้เทคนิค Ping sweep เพื่อค้นหาว่ามีเครื่องไหนอยู่ในเครือข่ายบ้าง
- ใช้เทคนิคในการอ่าน Shared Folder
- ใช้เทคนิค Trace route เพื่อสำรวจ Router
- ใช้เทคนิค การสแกนพอร์ตหรือ Port scanning เพื่อค้นหาว่ามีเซอร์วิสอะไรบ้างที่ทำงานอยู่หรืออยู่ในสถานะ listening
- ใช้เทคนิค Active Stack fingerprinting เพื่อค้นหาประเภทของระบบปฏิบัติการ
- ใช้เทคนิคในการอ่านค่า MAC Address

6.1.1 ส่วนเครื่องมือต่าง ๆ ที่ใช้ในการพัฒนาได้แก่

- Microsoft Visual C++ 6.0 MSDN Library Visual Studio 6.0
- Winpcap 2.3
- Nmap 3.48
- Multi Route Traffic Grapher(MRTG)
- Snmp Utility
- Microsoft Windows 2000 หรือ XP

โดยรูปแบบของโปรแกรมจะมี Input เป็น IP address จะใส่แค่ค่าเดียวหรือ ใส่เป็นช่วงก็ได้ ผลลัพธ์ที่ได้ออกมาจะแสดงในรูปแบบ Graphic โดยจะวาดเป็นเครือข่ายแบบคร่าวๆ ไม่แสดงออกเป็น Topology แบบ Physical แต่จะวาดโดยแบ่งตามเครือข่ายที่เราทำการสำรวจโดยแสดงเครื่องคอมพิวเตอร์ทั้งหมดที่อยู่บนเครือข่ายที่ทำการสำรวจ และในแต่ละเครื่องก็จะแสดงรายละเอียดต่างๆ เช่น มี MAC Address อะไร, รายชื่อผู้ผลิตอุปกรณ์ Network ที่ใช้บนระบบเครือข่ายนั้น, มีเซอร์วิส Port อะไรบ้างที่ทำงานอยู่บ้าง, ประเภทของระบบปฏิบัติการ และหากมีการเปิด SNMP Service ก็จะอ่านรายละเอียดอื่น ๆ เพิ่มเติมได้เช่น CPU ที่ใช้เป็นรุ่นไหน มีจำนวน Interface ที่ติดต่ออยู่บนเครือข่ายเป็นเท่าไร เป็นต้น

6.2 การออกแบบโครงสร้างของโปรแกรม

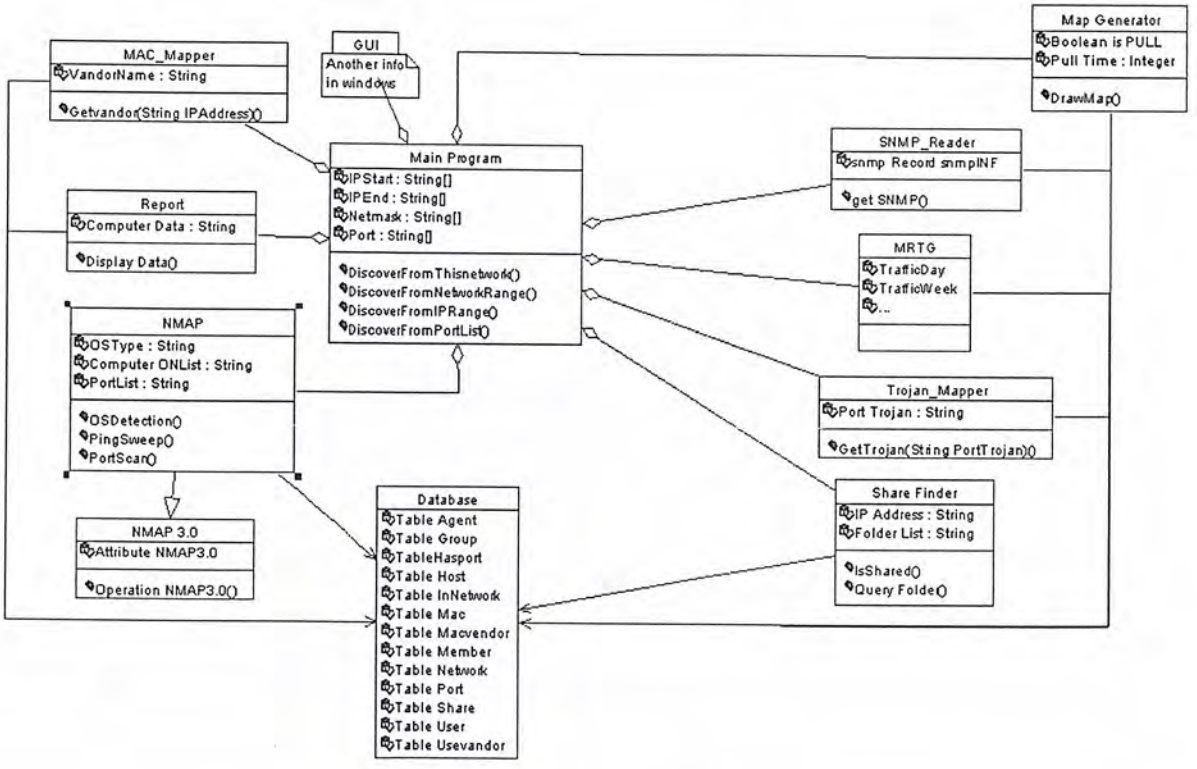


รูปที่ 6-1 แสดง Use Case ของโปรแกรม

คำศัพท์	คำอธิบาย
User	ผู้ที่ใช้โปรแกรมที่ต้องการทำการสำรวจระบบเครือข่าย เช่น ผู้ดูแลระบบ
Discover network map	ใช้โปรแกรมทำการสร้างแผนภาพของระบบเครือข่ายรวมโดยผู้ใช้ใส่ข้อมูลที่จำเป็นในการใช้สำรวจ
Update network map	สั่งให้โปรแกรมทำการสร้างแผนภาพของระบบเครือข่ายรวมอีกครั้ง โดยอาจจะใช้การกำหนดระยะเวลาในการอัพเดทเป็นระยะ ๆ
Find general information of computer	สั่งให้โปรแกรมทำการค้นหา ข้อมูลเบื้องต้นของคอมพิวเตอร์ เช่น ระบบปฏิบัติการ พอร์ทที่เปิด ชื่อเครื่อง Mac address Share directory User และ Group ที่อยู่ในเครื่อง เป็นต้น
Read SNMP data	สั่งให้โปรแกรมทำการอ่านข้อมูลจาก SNMP Service ของคอมพิวเตอร์ในระบบโดยเลือกคอมพิวเตอร์จากแผนภาพที่

	ได้จากการสำรวจ
Import file	สั่งให้โปรแกรมทำการเก็บข้อมูลของเครือข่ายลงไฟล์
Export file	สั่งให้โปรแกรมทำการดึงข้อมูลที่อยู่ในไฟล์เพื่อที่จะนำมาแสดงเป็นแบบกราฟิก
Add tools	ทำการเพิ่มโปรแกรมย่อยเข้าไปในโปรแกรมหลัก
Read graph of traffic	สั่งให้โปรแกรมทำการอ่านข้อมูลสภาพ ทราฟิกของเครื่องคอมพิวเตอร์แล้วนำมาแสดงผลให้อยู่ในรูปกราฟ

ตารางที่ 6-1 แสดงคำอธิบาย Use Case Diagram

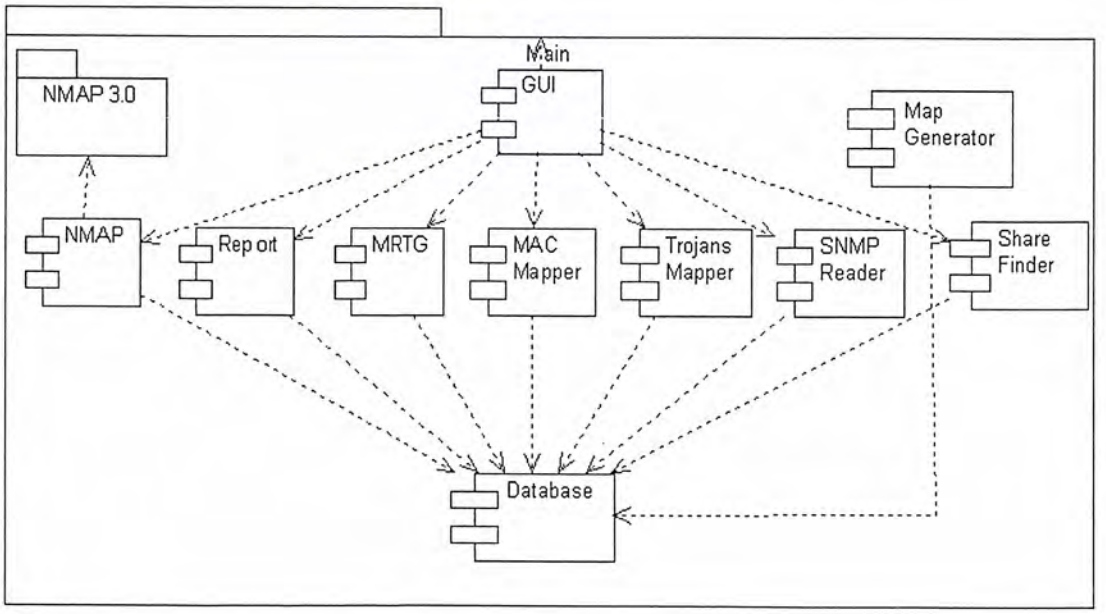


รูปที่ 6-2 แสดง Class Diagram

Class	คำอธิบาย
Main Program	Class หลักของโปรแกรมเพื่อทำหน้าที่ติดต่อกับส่วนต่าง ๆ ของโปรแกรม
NMAP 3.0	Class ของโปรแกรม NMAP v.3.0 ที่เรานำมาใช้
NMAP	Class ของโปรแกรม NMAP ที่เราสืบทอดมาเพื่อนำมาใช้เป็นส่วนที่เราใช้ในการ pingsweep เครือข่าย รวมทั้งการตรวจสอบ port และใช้ในการตรวจหา OS ของคอมพิวเตอร์ในระบบ

MRTG	Class ที่ทำหน้าที่ดึงภาพจากโปรแกรม MRTG มาแสดง
MAC_Mapper	Class ที่ทำหน้าที่เปรียบเทียบ MAC Address กับรายชื่อ ผู้ผลิตอุปกรณ์
Trojans Mapper	Class ที่ทำหน้าที่เปรียบเทียบพอร์ตกับ พอร์ตที่เป็นพอร์ตของโทรจัน
SNMP Reader	Class ที่ใช้ในการอ่านค่าจาก SNMP Service ของคอมพิวเตอร์ในระบบเครือข่าย
Share Finder	Class ที่ใช้ในการสำรวจว่าระบบมีการเปิด Shared Folder ไว้หรือไม่ และทำการอ่าน Folder List ที่ Share ไว้
Map Generator	Class ที่ใช้ในการสร้างแผนภาพของคอมพิวเตอร์ในระบบเครือข่ายจากข้อมูลใน Database
Report	Class ทำหน้าที่นำข้อมูลจากฐานข้อมูลมาออกเป็นรายงาน
Database	Class ที่ใช้ในการติดต่อกับ Database เพื่อทำการบันทึก / อ่านข้อมูล
GUI	Class ที่ทำหน้าที่ติดต่อกับผู้ใช้ในรูปแบบ Graphic

ตารางที่ 6-2 แสดงคำอธิบาย Class Diagram



รูปที่ 6-3 แสดง Component Diagram

จากรูป Component Diagram แสดงส่วนประกอบย่อย ๆ ของโปรแกรมโดยส่วนประกอบต่าง ๆ มีหน้าที่ดังนี้

- NMAP 3.0 เป็น Package ของโปรแกรม NMAP 3.0

- NMAP เป็นส่วนที่สืบทอดมาจากโปรแกรม NMAP 3.0 มีหน้าที่ในการทำ Ping sweep , Port Scan และ การทำ OS Detection ของคอมพิวเตอร์
- MRTG เป็นส่วนที่ทำหน้าที่แสดงรูป ทราฟิกของเครื่อข่ายนั้นๆ
- Share Finder เป็นส่วนประกอบที่ใช้ในการสำรวจว่าระบบมีการเปิด Shared Folder ไว้หรือไม่ และทำการอ่าน Folder List ที่ Share ไว้
- SNMP_Reader เป็นส่วนประกอบที่ใช้ในการติดต่อกับอุปกรณ์ในเครือข่ายด้วย SNMP Protocol เพื่ออ่านข้อมูลของอุปกรณ์นั้น
- MAC_Mapper เป็นส่วนประกอบที่ใช้ในการหา MAC Address ของอุปกรณ์ในระบบเครือข่ายและทำการค้นหา Vendor ของอุปกรณ์ในเครือข่าย
- Trojans Mapper เป็นส่วนที่ใช้ในการหาว่าพอร์ตที่เครื่องนั้นเปิดเป็นพอร์ตที่มีสิทธิ์จะเป็นโทรจันหรือไม่
- Report เป็นส่วนที่ทำหน้าที่ออกรายงานของเครื่องในเครือข่าย
- Database เป็นส่วนประกอบที่ใช้ในการติดต่อกับฐานข้อมูลเพื่อเก็บข้อมูลที่ได้มาจากส่วนประกอบย่อยอื่น ๆ และอ่านฐานข้อมูลเพื่อนำไปสร้างแผนภาพระบบเครือข่าย
- Map_Generator เป็นส่วนประกอบที่ใช้ในการวิเคราะห์ข้อมูลที่ได้จากโปรแกรมย่อยและนำไปสร้างเป็นแผนภาพระบบเครือข่าย
- GUI เป็นส่วนประกอบที่ติดต่อกับผู้ใช้โดยมีหน้าที่รับข้อมูลและแสดงผลในรูปแบบ Graphic เพื่ออำนวยความสะดวกแก่ผู้ใช้

6.3 โครงสร้างการทำงานของโปรแกรม

โครงสร้างการทำงานของโปรแกรมจะประกอบส่วนประกอบที่สำคัญได้แก่ ส่วนติดต่อกับผู้ใช้ โดยในขั้นตอนนี้จะได้รับ input จากผู้ใช้ถึงรายละเอียดหรือขอบเขตของเครือข่ายที่ผู้ใช้ต้องการให้โปรแกรมสำรวจโดยจะมีให้ผู้ใช้เลือก 3 รูปแบบ คือ 1. เครือข่ายของเครื่องที่ผู้ใช้ๆงานอยู่ 2. ช่วงของเครือข่ายที่ต้องการและสามารถที่จะกำหนดออกเป็นเครือข่ายย่อยๆได้ 3. ช่วงของ ไอพี แอดเดรสที่ต้องการ โดยเมื่อรับ input มาแล้วโปรแกรมจะมีการแยกส่วนการทำงานเป็น ส่วนสำรวจข้อมูลของเครือข่าย และส่วนสำรวจข้อมูลระบบคอมพิวเตอร์ โดยการทำงานของทั้งสองส่วนดังกล่าวจะนำผลลัพธ์ที่ได้เก็บในฐานข้อมูล และเมื่อได้ข้อมูลมาแล้วส่วนแสดงผลให้ผู้ใช้งานจะทำหน้าที่นำข้อมูลที่ได้มาจัดแสดงให้ผู้ใช้งานได้รับทราบ

รายละเอียดในการทำงานของส่วนต่างๆ มีดังนี้

1. ส่วนติดต่อกับผู้ใช้ เป็นส่วนที่แสดงฟังก์ชั่นการทำงาน และแสดงผลลัพธ์ของการทำงานให้แก่ผู้ใช้งาน
2. ส่วนสำรวจข้อมูลระบบคอมพิวเตอร์ เป็นส่วนที่ทำหน้าที่หาข้อมูลของเครื่องคอมพิวเตอร์ใดๆในเครือข่าย และเก็บข้อมูลที่ได้นั้นลงฐานข้อมูล ประกอบด้วย การทำงาน 3 ส่วนคือ

2.1 ส่วนสำรวจรายละเอียดของคอมพิวเตอร์ คือ ส่วนที่ทำหน้าที่หาข้อมูลพื้นฐานของแต่ละเครื่อง เช่น ระบบปฏิบัติการ พอร์ทที่เปิดให้บริการ เป็นต้น

2.2 ส่วนสำรวจความปลอดภัยของคอมพิวเตอร์ คือ ส่วนที่ทำหน้าที่หาข้อมูลของความไม่ปลอดภัยของเครื่องคอมพิวเตอร์ เช่น พอร์ทที่มีการเปิดอย่างไม่เหมาะสม รหัสผ่านของเครื่องที่มีการตั้งไม่เหมาะสม

2.3 ส่วนสำรวจข้อมูล SNMP ของคอมพิวเตอร์ คือ ส่วนที่ทำหน้าที่ ค้นหาข้อมูลจากตาราง mib ของโปรโตคอล SNMP ของเครื่องในเครือข่าย

3. ส่วนสำรวจข้อมูลเครือข่าย เป็นส่วนที่ทำหน้าที่หาข้อมูลจากเครือข่ายที่ผู้ดูแล ดูแลอยู่ไปเก็บไว้ในฐานข้อมูล ประกอบด้วยการทำงาน 2 ส่วน คือ

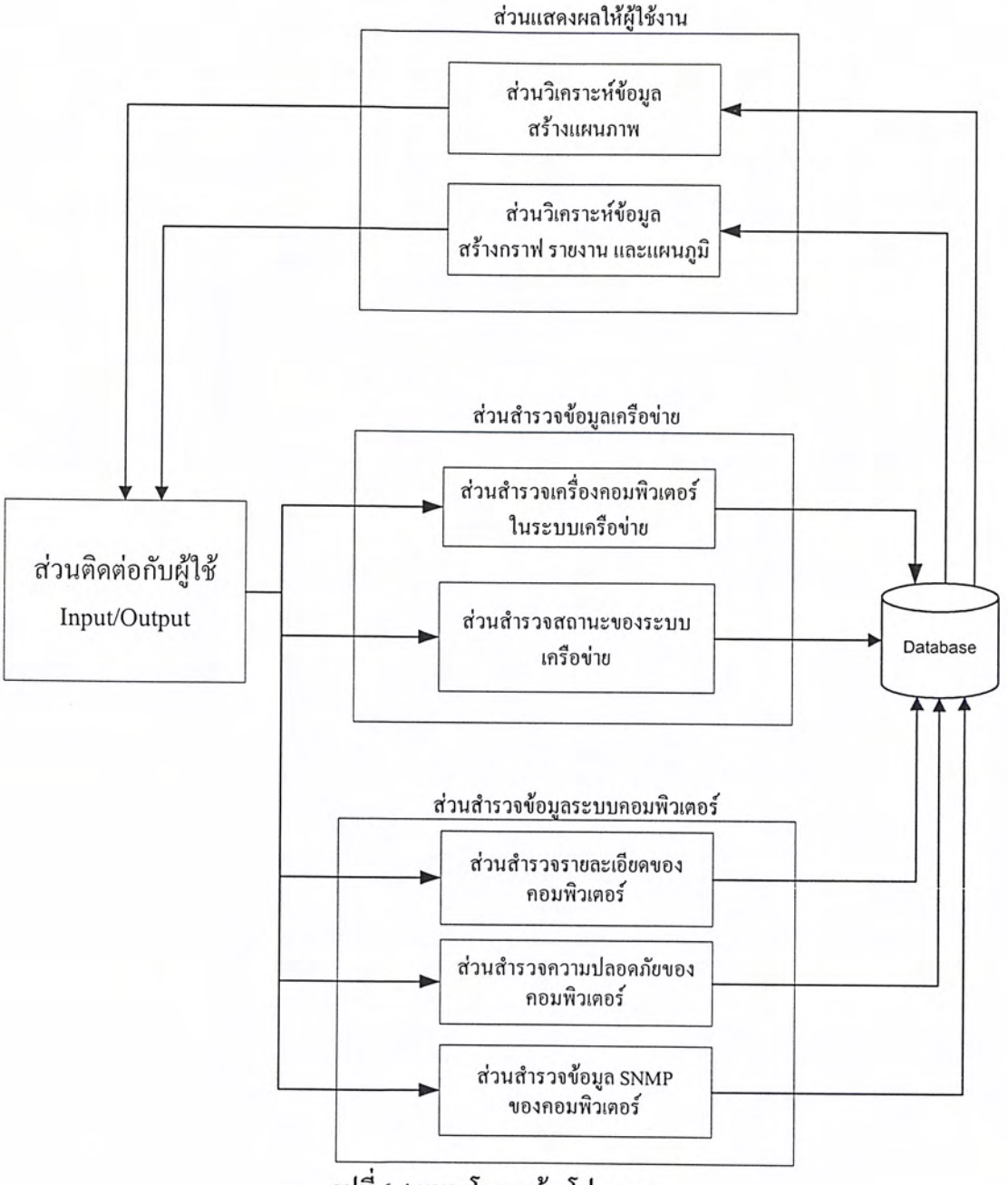
3.1 ส่วนสำรวจเครื่องคอมพิวเตอร์อุปกรณ์ในเครือข่าย คือ จะทำหน้าที่หาเครื่องคอมพิวเตอร์ และอุปกรณ์ในเครือข่ายนั้น ว่ามีอะไรบ้าง

3.2 ส่วนสำรวจสถานะของระบบเครือข่าย คือ จะทำหน้าที่เก็บข้อมูลสถานะของเครือข่ายนั้น เช่น ความสามารถในการติดต่อกับเครือข่ายภายนอก หรือ ปริมาณข้อมูลที่รับ-ส่ง ของเครือข่าย เป็นต้น

4. ส่วนแสดงผลให้ผู้ใช้งาน จะทำหน้าที่นำข้อมูลที่อยู่ในฐานข้อมูล มาจัดแสดงเป็น รูปภาพหรือรายงานแล้วส่งให้ส่วนติดต่อกับผู้ใช้ มี 2 ส่วนคือ

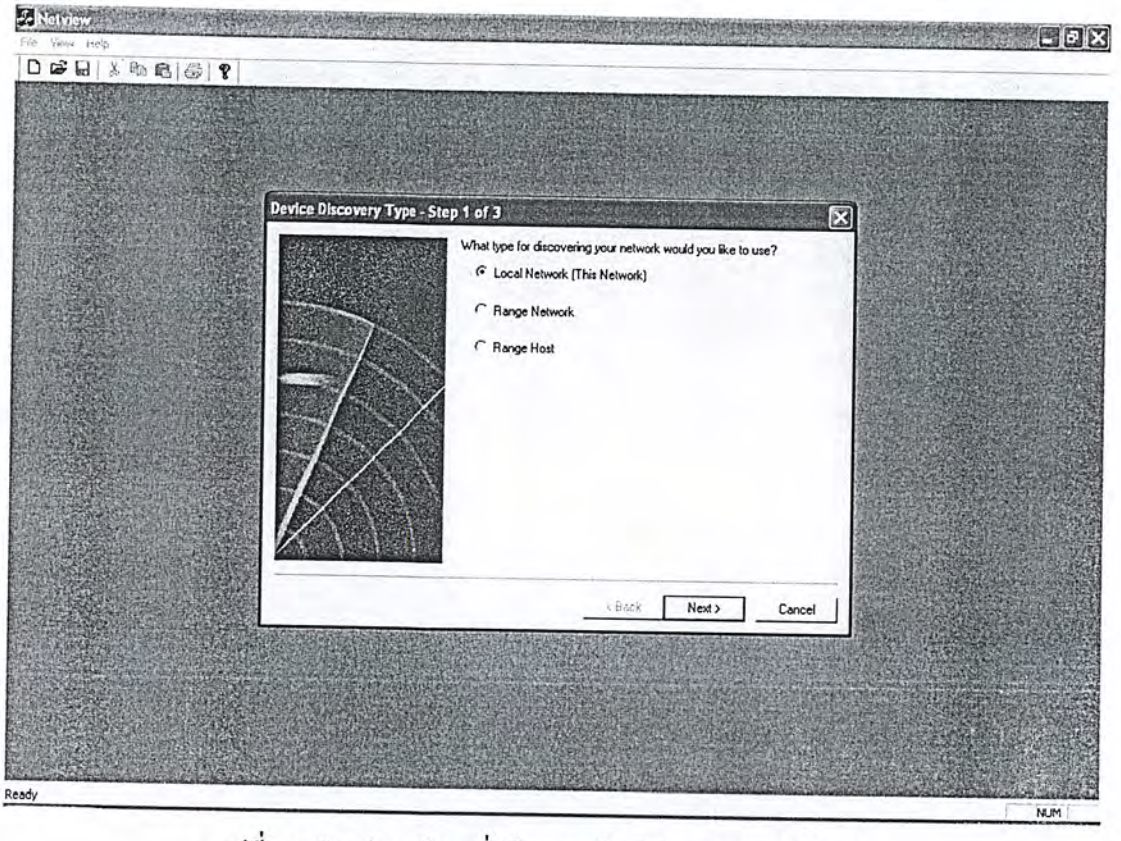
4.1 ส่วนวิเคราะห์ข้อมูลและสร้างแผนภาพ จะทำการนำข้อมูลที่ได้เช่น เครื่องคอมพิวเตอร์ที่อยู่ในเครือข่าย มาจัดทำเป็นรูปภาพ

4.2 ส่วนวิเคราะห์ข้อมูล สร้างกราฟ รายงาน และแผนภูมิ มีการนำข้อมูลบางส่วนมาจัดรูปแบบเป็นแผนภูมิ รายงาน เช่น ส่วนของ ปริมาณข้อมูลของเครือข่าย สถานะของเครื่องคอมพิวเตอร์ เป็นต้น



รูปที่ 6-4 แสดงโครงสร้างโปรแกรม

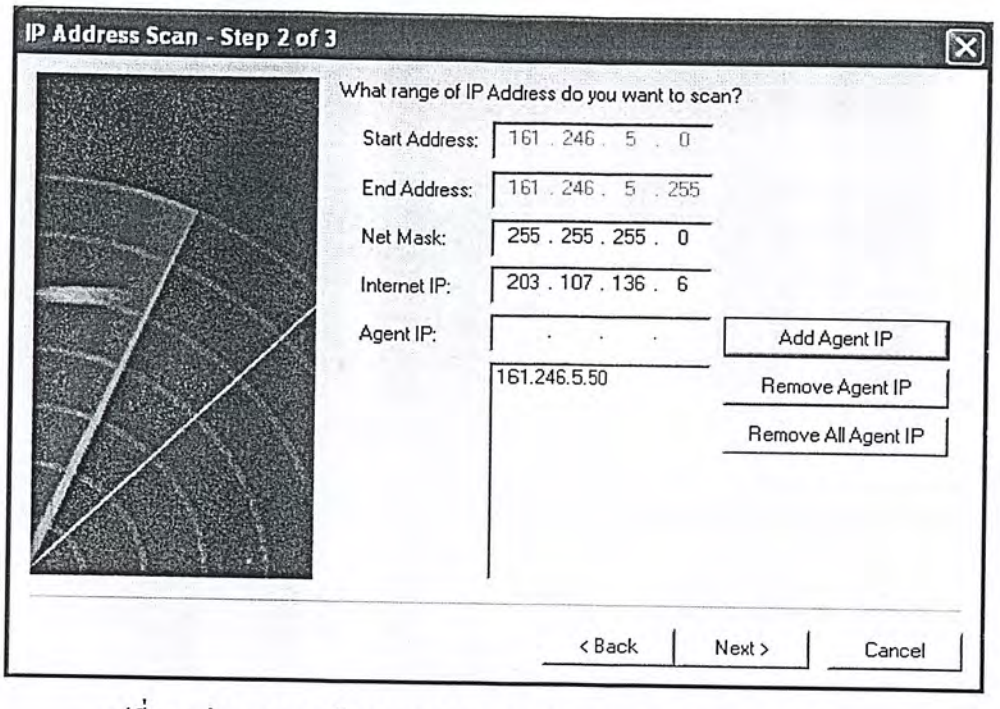
6.4 ตัวอย่างส่วนติดต่อกับผู้ใช้



รูปที่ 6-5 ตัวอย่างหน้าจอที่ใช้ในการเลือกใส่ Input ของโปรแกรม

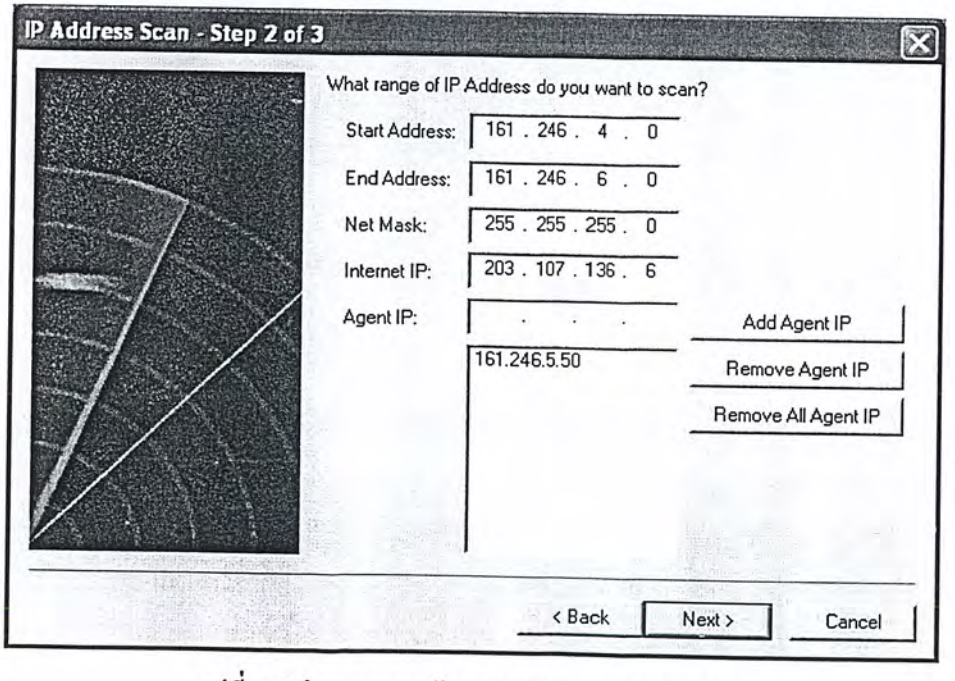
Input ที่ต้องการของโปรแกรม

1. กรณีของ Location Network (This Network) เป็นการค้นหาคอมพิวเตอร์ที่อยู่ในเครือข่ายของคอมพิวเตอร์เครื่องนั้น และสามารถที่จะทำการแบ่งเครือข่ายออกเป็นเครือข่ายย่อยๆ ได้ และยังมีส่วนของ Internet Address สำหรับกรอกไอพี ภายนอกเครือข่ายเพื่อที่จะดูความสามารถในการออกเน็ต และ ส่วนของ Agent IP ซึ่งเป็นไอพีสำหรับเป็นตัวแทนของเครือข่ายโดยสามารถที่จะดูทราฟิกของข้อมูลในเครือข่ายนั้นได้



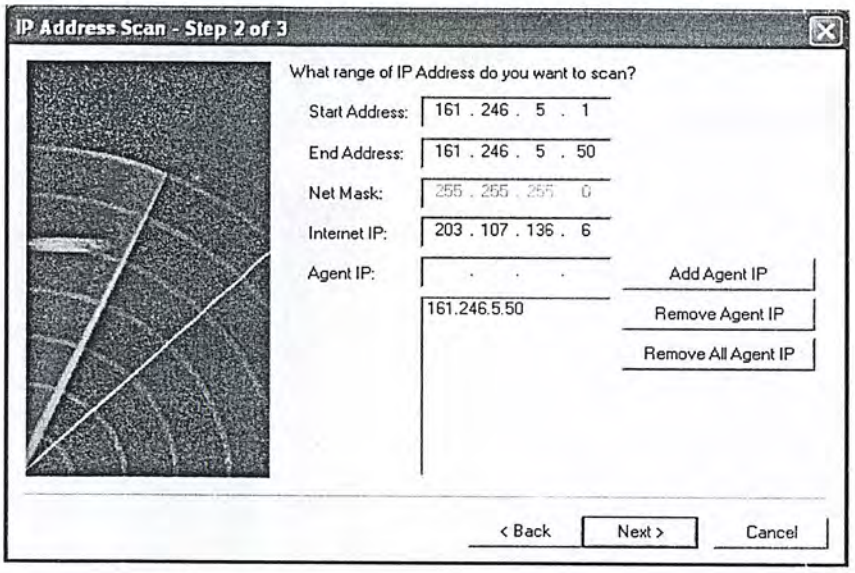
รูปที่ 6-6 ส่วนของการค้นหาแบบ Location Network (This Network)

2. กรณีของ Range Network เป็นการค้นหาช่วงของเครือข่ายที่ต้องการ และสามารถที่จะทำการแบ่งเครือข่ายออกเป็นเครือข่ายย่อยๆ ได้ และยังมีส่วนของ Internet Address สำหรับกรอกไอพี ภายนอกเครือข่ายเพื่อที่จะดูความสามารถในการออกเน็ต และส่วนของ Agent IP ซึ่งเป็นไอพีสำหรับเป็นตัวแทนของเครือข่ายโดยสามารถที่จะดูทราฟิกของข้อมูลในเครือข่ายนั้นได้



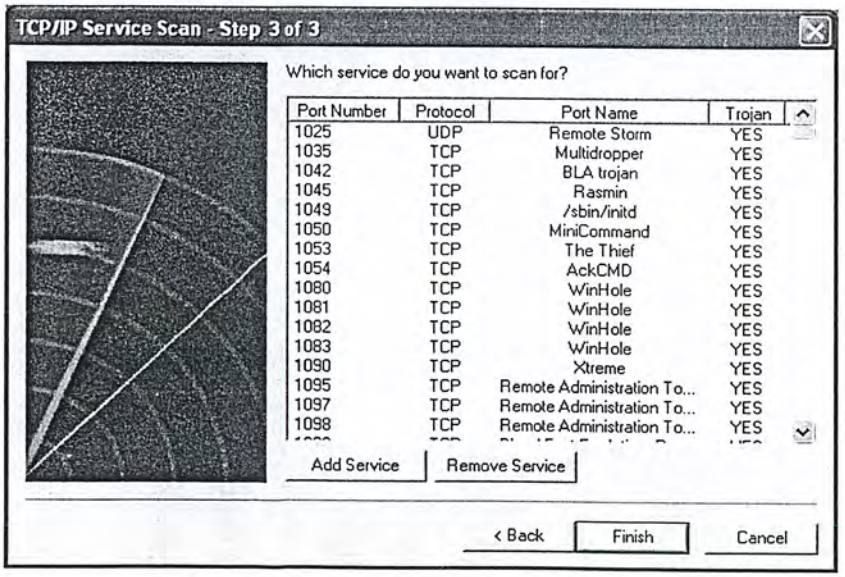
รูปที่ 6-7 ส่วนของการค้นหาแบบ Range Network

3. กรณีของ Range IP Address เป็นการค้นหาช่วงของไอพีที่ต้องการ แต่ไม่สามารถที่จะทำการแบ่งเครือข่ายออกเป็นเครือข่ายย่อยๆ ได้ และยังมีส่วนของ Internet Address สำหรับกรอกไอพี ภายนอกเครือข่ายเพื่อที่จะดูความสามารถในการออกเน็ต และส่วนของ Agent IP ซึ่งเป็นไอพีสำหรับเป็นตัวแทนของเครือข่ายโดยสามารถที่จะดูทราฟิกของข้อมูลในเครือข่ายนั้นได้



รูปที่ 6-8 ส่วนของการค้นหาแบบ Range IP Address

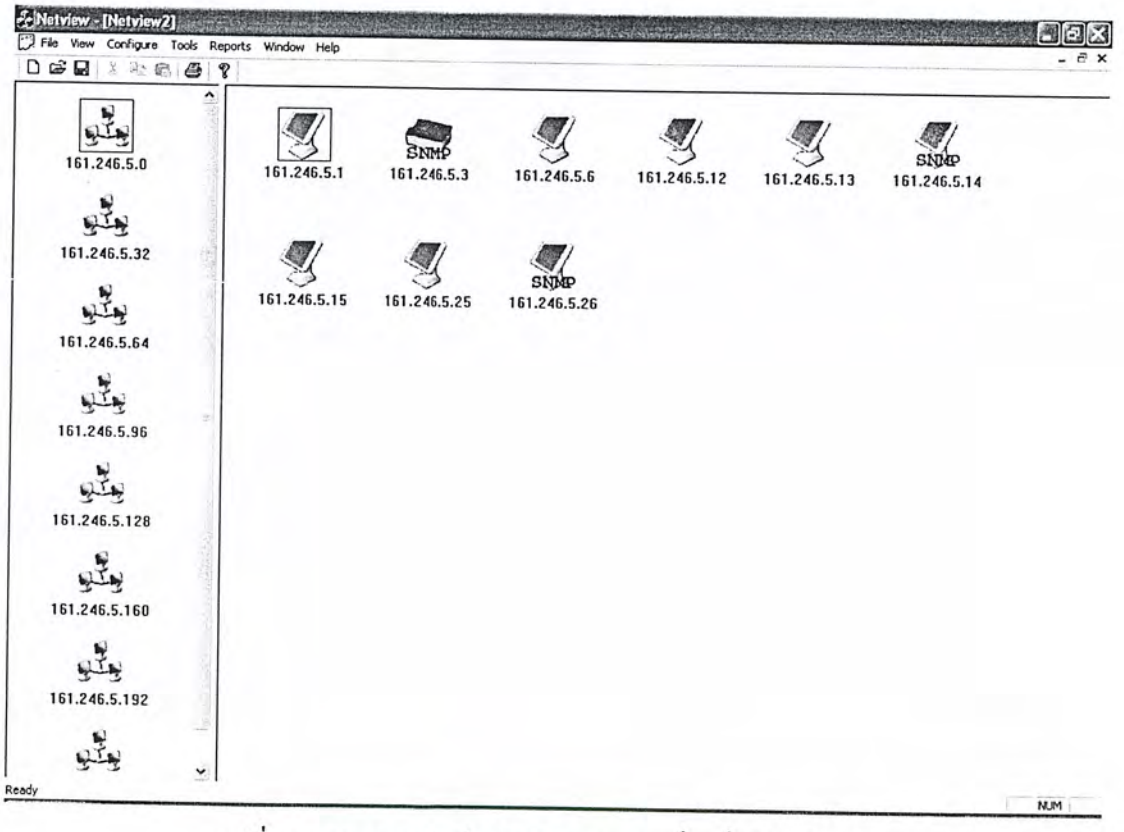
ส่วนถัดมาเป็นส่วนของการเลือกพอร์ตที่ต้องการตรวจสอบนอกเหนือจากพอร์ต 1024 แรกโดยพอร์ตที่มีให้เลือกนั้นจะเป็นรายชื่อพอร์ตของโทรจัน เพื่อสำหรับที่จะทำการตรวจสอบความไม่ปลอดภัยในระบบ



รูปที่ 6-9ส่วนของการเพิ่มพอร์ตที่ต้องการตรวจสอบ

6.5 Output ของโปรแกรมที่ได้จากการสำรวจ

- แผนภาพทาง logical ของระบบเครือข่ายที่สำรวจมาได้
- ข้อมูลที่ได้จากการอ่าน SNMP กรณีที่เครื่องที่ต้องการมีการเปิดบริการ SNMP Service
- บอก OS ของเครื่องที่ต้องการทราบ
- บอก Service Port ของเครื่องที่ต้องการทราบ
- บอก MAC Address และเปรียบเทียบเป็นชื่อ Vendor ของอุปกรณ์เครื่องที่ต้องการทราบ
- บอก Shared Folder ของเครื่องที่ต้องการทราบ
- บอก user และ group ของเครื่องที่ต้องการทราบ
- ออก report ของเครือข่ายและเครื่องที่อยู่ในเครือข่าย
- แสดงข้อมูล เข้า-ออก จากเครือข่าย หรือ จากเครื่องที่ต้องการทราบ



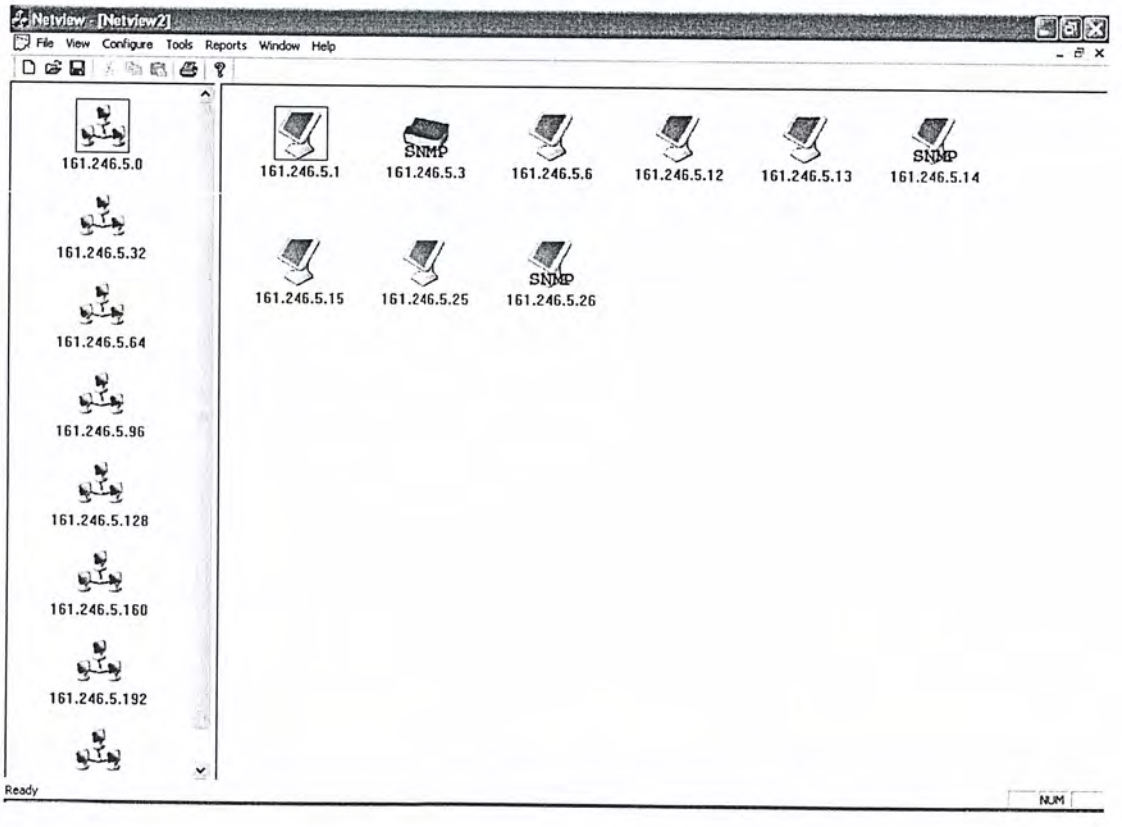
รูปที่ 6-10 แสดงผลลัพธ์หลังจากแสกนหาเครื่องที่อยู่ในเครือข่าย

บทที่ 7
การทำงานของโปรแกรมสำรวจเครือข่ายและระบบคอมพิวเตอร์

เป็นการแสดงการทำงานของโปรแกรม โดยแบ่งเป็น การสำรวจเครื่องคอมพิวเตอร์ที่มีการใช้งานเครือข่าย, การสำรวจพอร์ตที่เครื่องคอมพิวเตอร์เปิดให้บริการ, การสำรวจ Shared Folder ที่เปิดอยู่ในระบบคอมพิวเตอร์, การสำรวจหา user และ group ของเครื่องคอมพิวเตอร์ที่เปิดให้บริการ, การดูข้อมูลของเครื่องคอมพิวเตอร์ที่เปิดให้บริการเอสเอ็นเอ็มพี, การแสดงผลของข้อมูลโดยอยู่ในรูปของเอกสาร และการแสดงผลของข้อมูลที่เข้า-ออก ในระบบเครือข่ายหรือคอมพิวเตอร์ในเครือข่าย

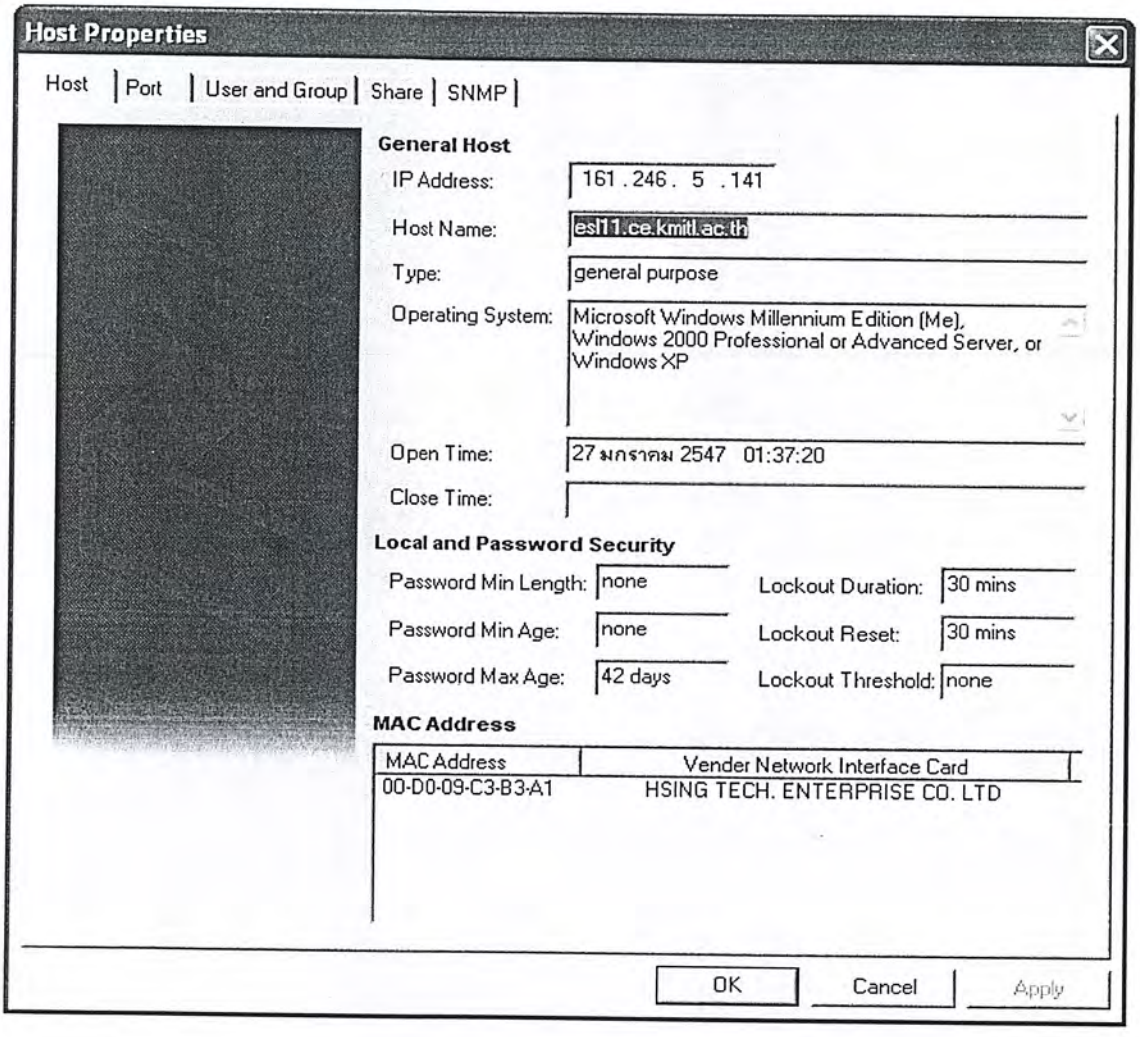
7.1 การสำรวจเครื่องคอมพิวเตอร์ที่มีการใช้งานเครือข่าย

หลังจากที่เราได้ทำการเลือกรูปแบบการค้นหาเครื่องคอมพิวเตอร์ในเครือข่ายและเลือกพอร์ตที่ต้องการให้มีการค้นหาเพิ่มเติมแล้ว โปรแกรมจะทำการแสกนโดยใช้เทคนิค ping sweep ,แสกนพอร์ต, arp โดยผลลัพธ์จะแสดงในรูป 7-1



รูปที่ 7-1 แสดงรูปเครือข่ายที่ได้จากการสำรวจ

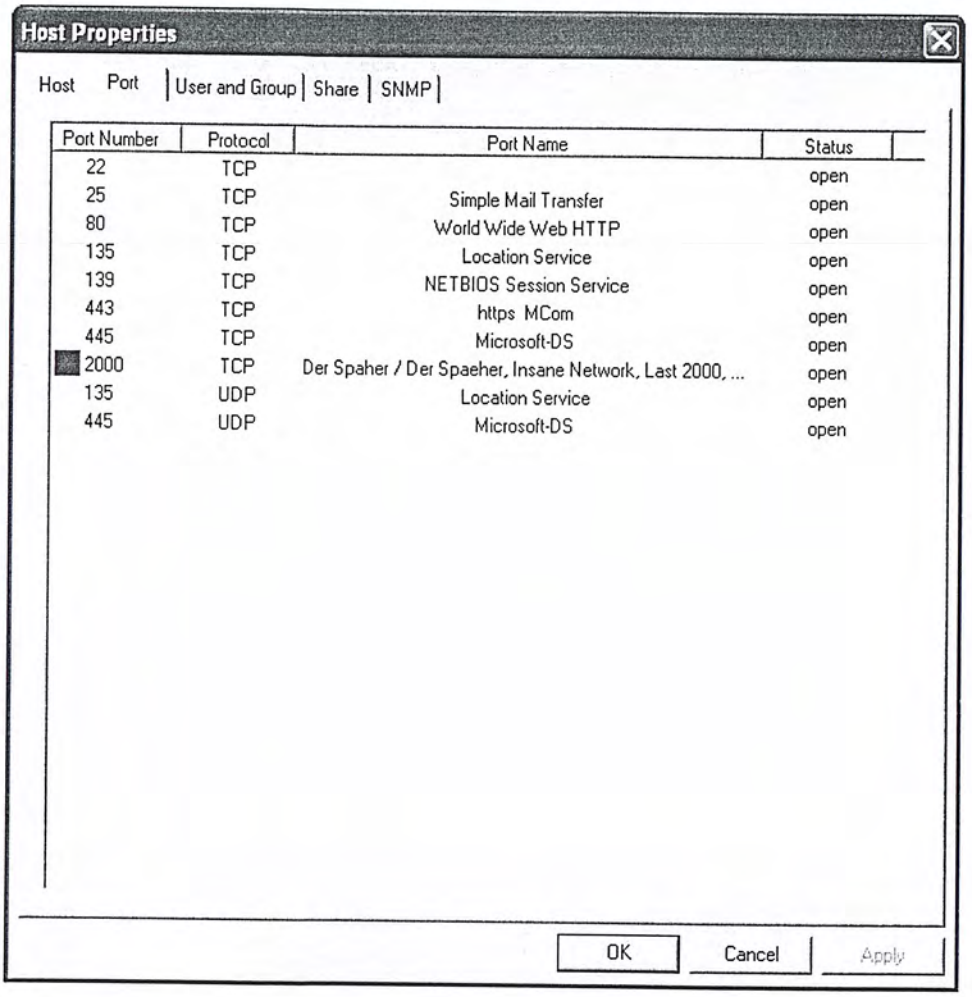
7.2 การสำรวจหาข้อมูลเบื้องต้นของเครื่องคอมพิวเตอร์



รูปที่ 7-2 แสดงรูปผลที่ได้จากการสำรวจข้อมูลของเครื่องคอมพิวเตอร์

โดยจากการรูป 7-2 นี้แสดงให้เห็นข้อมูลเบื้องต้นของเครื่องคอมพิวเตอร์ เช่นโฮสต์นี้ไอพี 161.246..5.141 มีชื่อเครื่องคือ esl.ce.kmitl.ac.th โดยมี type เป็น general purpose ซึ่งหมายถึงเครื่องคอมพิวเตอร์ทั่วๆไป คาดว่าน่าจะมีระบบปฏิบัติการเป็น Microsoft Windows Millennium Edition(ME) , Windows 2000 Professional or Advanced หรือ Windows XP เป็นต้น

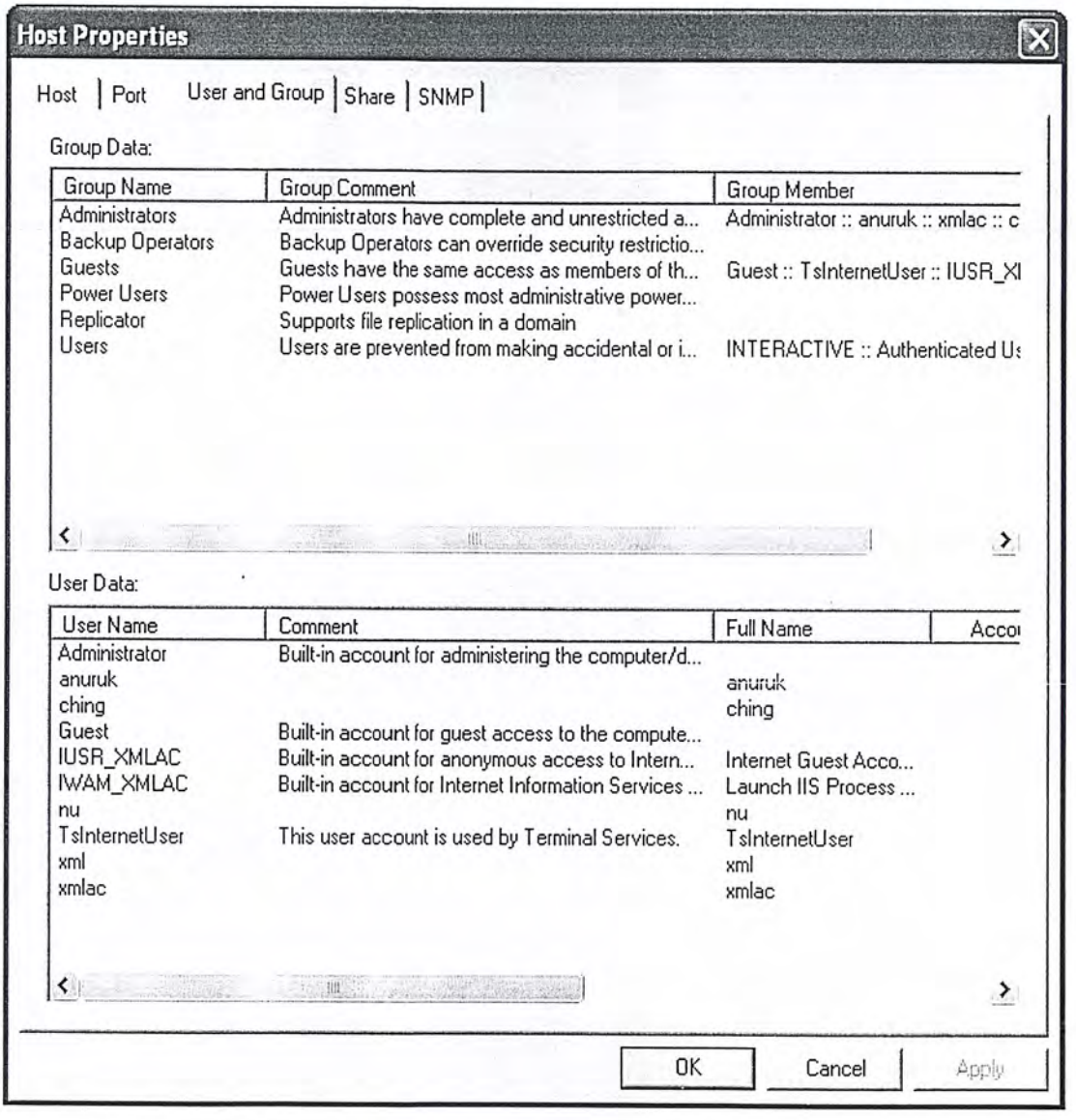
7.3 การสำรวจบริการที่เปิดของเครื่องคอมพิวเตอร์



รูปที่ 7-3 แสดงรูปผลที่ได้จากการสำรวจบริการที่เปิดของเครื่องคอมพิวเตอร์

จากผลลัพท์แสดงให้เราเห็นว่าเครื่องนี้มีการเปิดพอร์ต TCP คือ 22, 25, 80, 135, 139, 443, 445, 2000 และเปิดพอร์ต UDP คือ 135, 445 โดยที่พอร์ต 2000 อาจจะเป็นพอร์ตที่ไม่สมควรจะเปิด

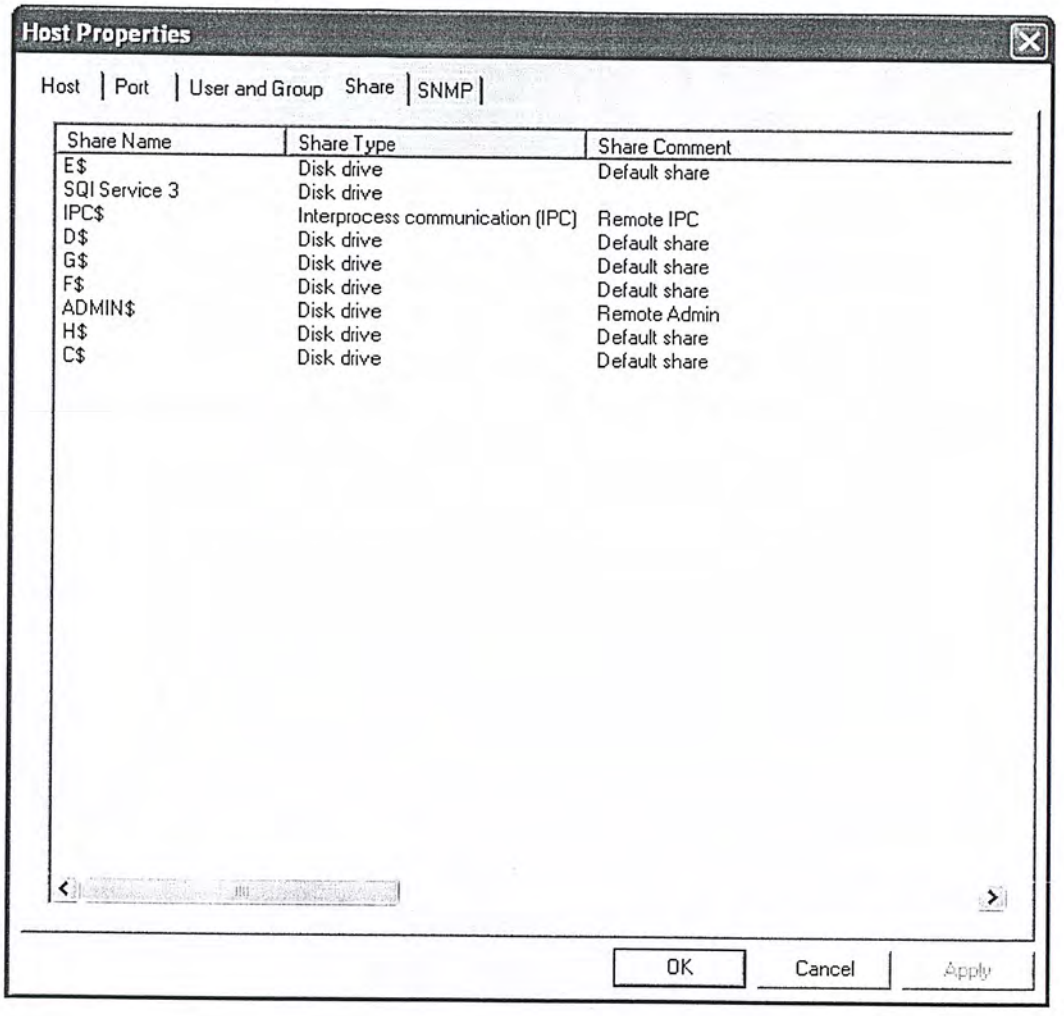
7.4 การสำรวจ user และ groupของเครื่องคอมพิวเตอร์



รูปที่ 7-4 แสดงรูปผลที่ได้จากการสำรวจ user และ group ของเครื่องคอมพิวเตอร์

รูปที่ 7-3 เป็นการแสดงข้อมูล user และ group ที่อยู่ในเครื่องว่ามี user อะไรและ group อะไรบ้าง โดยการค้นหาข้อมูลนั้นจะอาศัยเทคนิคของ null session ซึ่งจะทำให้ได้ข้อมูลของเฉพาะเครื่องที่เปิด null session เท่านั้น ซึ่งจากรูปทำให้เรารู้ว่าเครื่องนี้มี user ชื่อ Administrator, anuruk, ching, Guest เป็นต้น

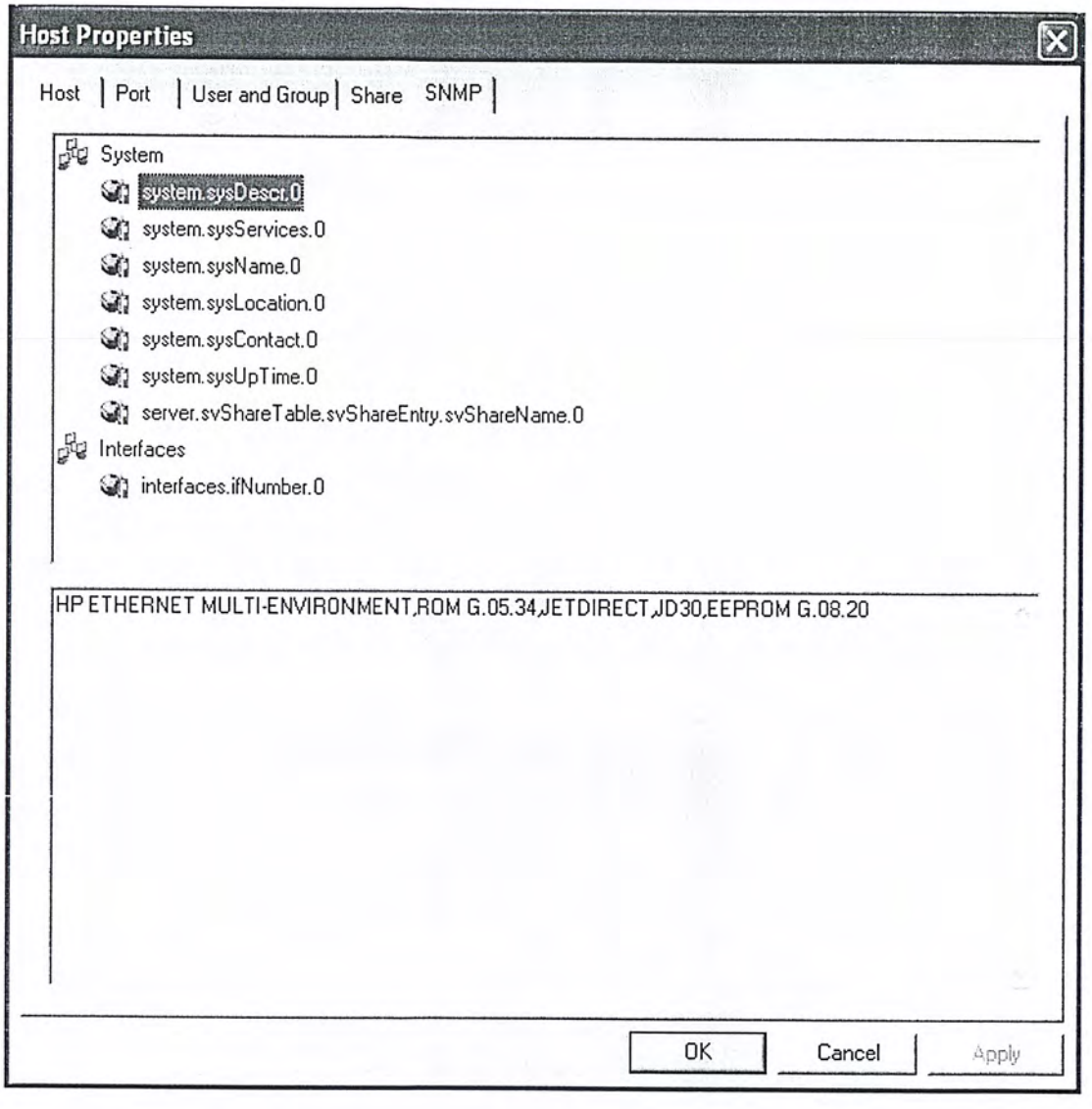
7.5 การสำรวจ การแชร์ไดร์ฟ และ การแชร์ไดเร็กทอรี่ ของเครื่องคอมพิวเตอร์



รูปที่ 7-5 แสดงรูปผลที่ได้จากการการแชร์ไดร์ฟ และ การแชร์ไดเร็กทอรี่ ของเครื่องคอมพิวเตอร์

จากรูปที่ 7-5 เป็นการค้นหาข้อมูลเกี่ยวกับการเปิดแชร์ไดร์ฟ และ แชร์ได้เร็กทอรี่ โดยจะมีส่วน Share Comment เป็นส่วนที่บอกว่าเป็นการแชร์ในลักษณะใด เช่น มีการแชร์ไดร์ฟ C, D, E, F, G, H เป็นแบบ Default Share และ IPC เป็นแบบ Remote IPC เป็นต้น

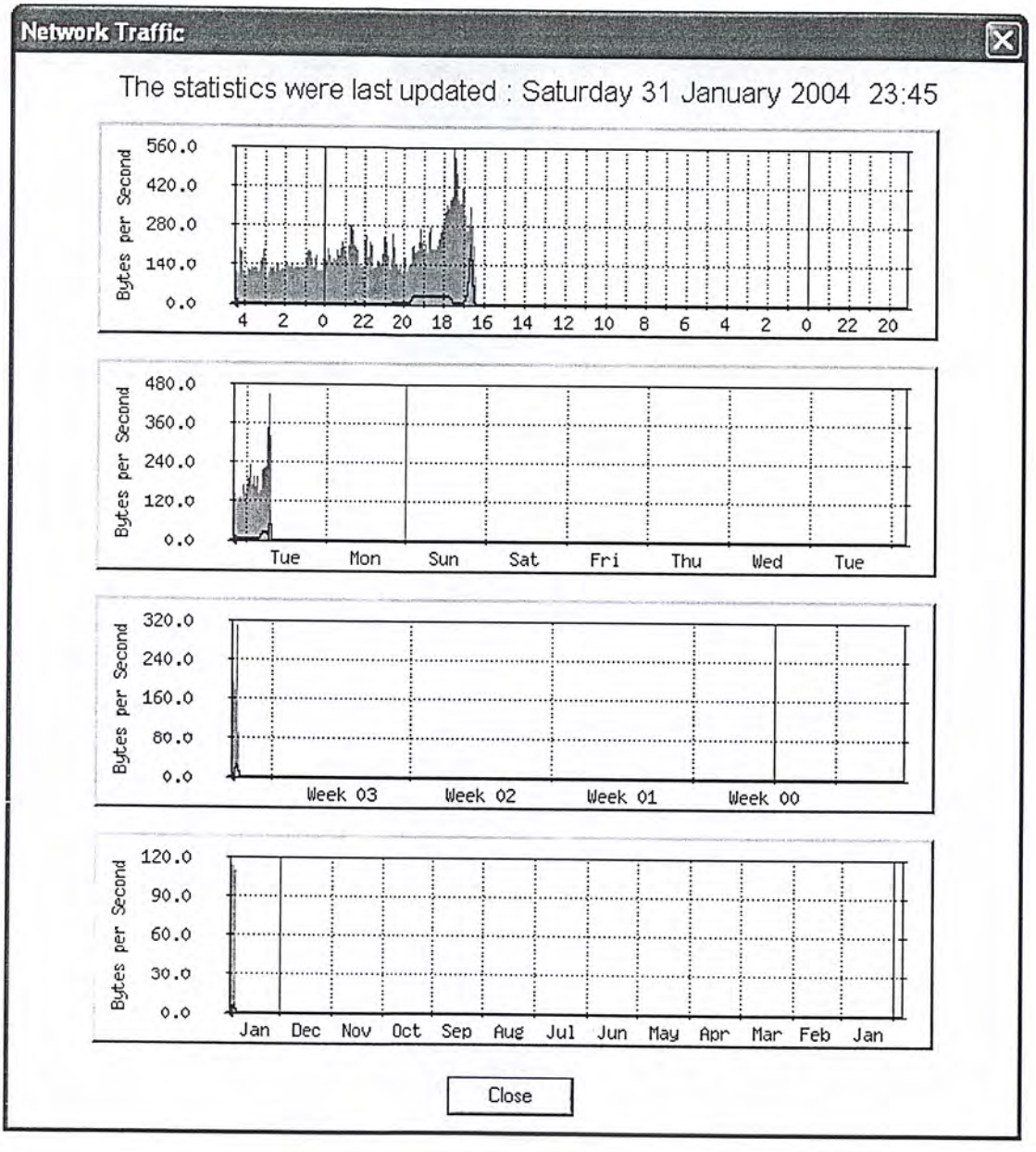
7.6 การสำรวจ ข้อมูลที่ให้บริการโดยโปรโตคอล เอสเอ็นเอ็มพีของเครื่องคอมพิวเตอร์



รูปที่ 7-6 แสดงรูปผลที่ได้จากการให้บริการโดยโปรโตคอล เอสเอ็นเอ็มพีของเครื่องคอมพิวเตอร์

เราสามารถที่จะดูข้อมูลของเครื่องคอมพิวเตอร์เครื่องนั้นได้ละเอียดมากขึ้นถ้าเกิดเครื่องนั้นเปิดให้บริการข้อมูลโดยโปรโตคอล เอสเอ็นเอ็มพี

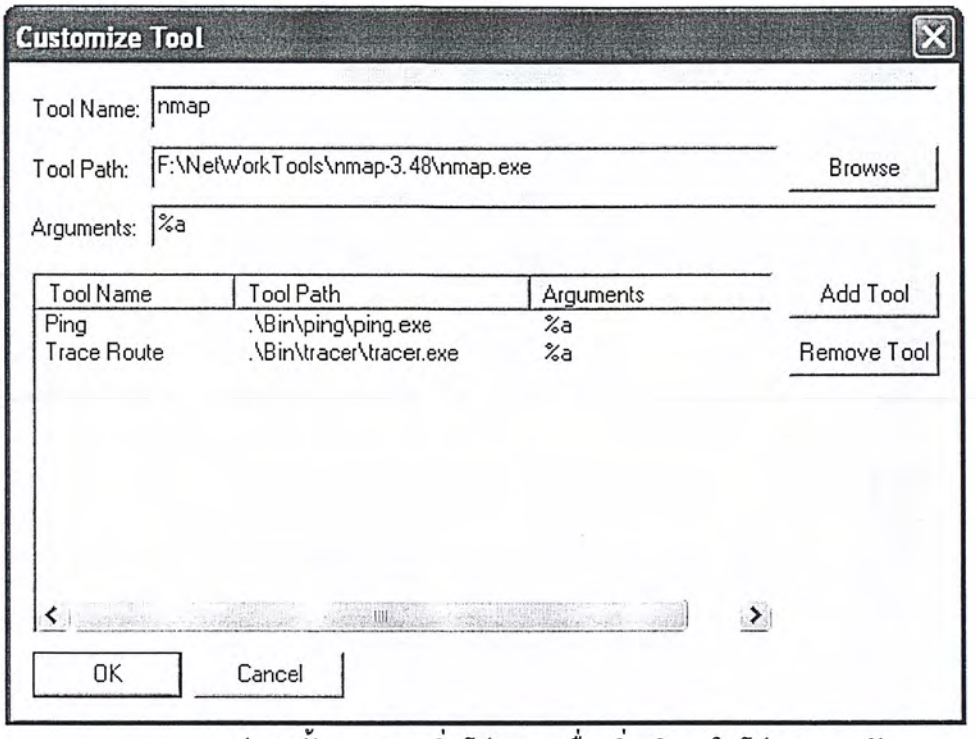
7.7 การดูสภาพเครือข่ายคอมพิวเตอร์



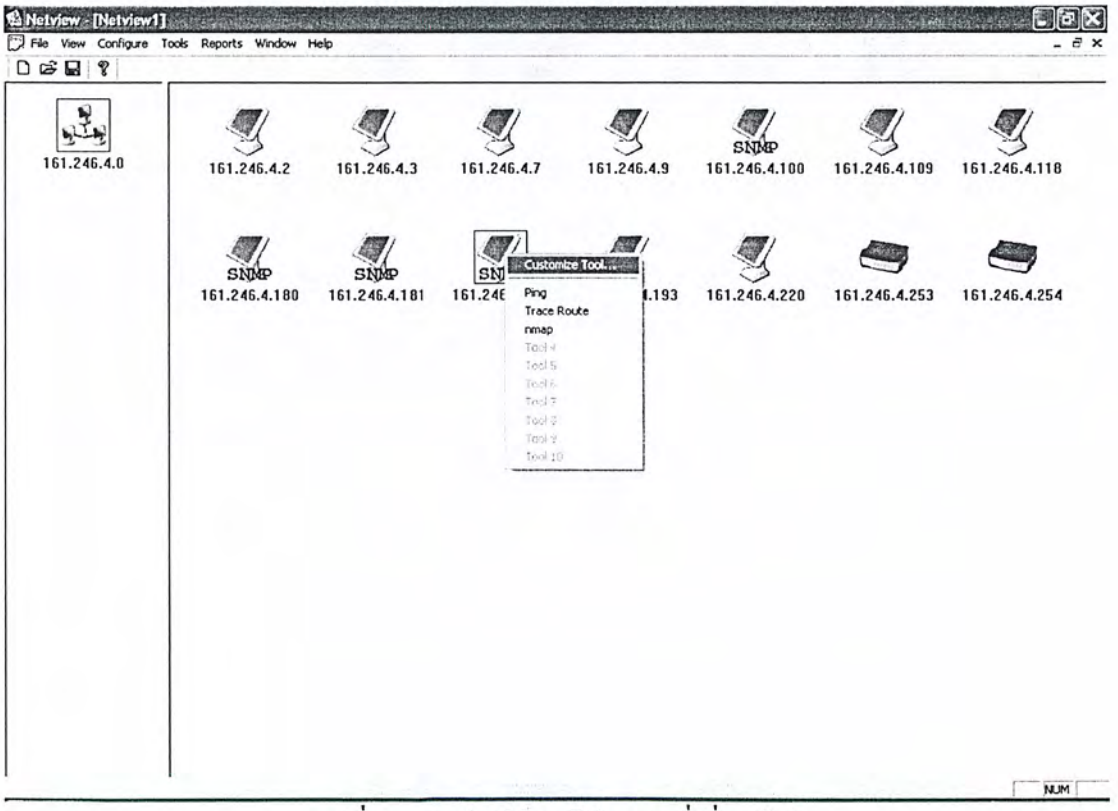
รูปที่ 7-7 แสดงผลของสภาพเครือข่ายคอมพิวเตอร์

จากรูปข้างบนเป็นรูปที่แสดงปริมาณการ รับ – ส่ง ข้อมูลของเครื่องในเครือข่ายคอมพิวเตอร์ในช่วงเวลา ใน 1 วัน 1 สัปดาห์ 1 เดือน และ 1 ปี โดยจะสามารถใช้ฟังก์ชั่นนี้ได้ถ้าเครื่องนั้นเปิดให้บริการเอสเอ็นเอ็มพี เท่านั้น

7.8 การเพิ่มโปรแกรมอื่นลงในโปรแกรมหลัก



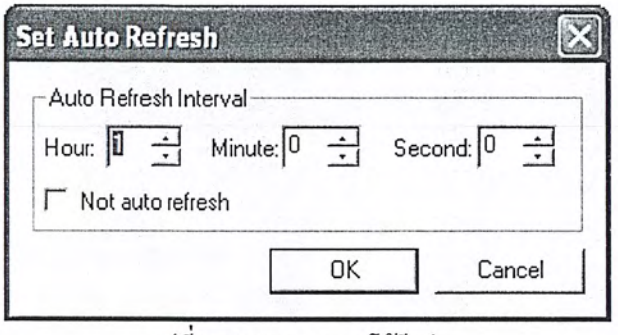
รูป 7-8 ขั้นตอนการเพิ่มโปรแกรมอื่นเพิ่มเติมลงในโปรแกรมหลัก



รูปที่ 7-9 การเรียกใช้งานโปรแกรมที่เพิ่มลงไป

จากรูปข้างต้นเป็นส่วนของการเพิ่มโปรแกรมย่อยเข้าไปในโปรแกรมหลัก โดยจะต้องมีการกำหนดชื่อโปรแกรม พาร์ทของโปรแกรม และอาร์กูเมนท์ ของโปรแกรมว่าจะให้เป็นอย่างไร ส่วนการเรียกใช้งานสามารถใช้ได้โดย คลิ้กขวาที่เครื่องที่ต้องการที่จะใช้โปรแกรม และเลือกโปรแกรมที่ต้องการใช้งานจากแถบเมนู

7.9 การเซตออโต้รีเฟรช



รูปที่ 7-10 การเซตออโต้รีเฟรช

จากรูป จะมีช่วงเวลาให้เลือกว่าเราจะกำหนดช่วงเวลาที่จะทำการแสกนได้ว่าจะต้องการที่จะแสกน ทุกๆเวลาเท่าไร

บทที่ 8
วิเคราะห์ผลการทดลองและสรุป

8.1 วิเคราะห์ผลการทดลอง

โปรแกรมสามารถทำงานได้ตามวัตถุประสงค์ดังนี้

- สามารถหาคอมพิวเตอร์ที่เปิดใช้งานเครือข่ายอยู่ในขณะนั้นได้และเก็บสถิติวัน เวลาการเปิด ปิด
- สามารถหา OS ได้จำเพาะเจาะจงมากขึ้น
- สามารถดู service หรือ port ที่เปิดของแต่ละเครื่องได้
- สามารถอ่านค่า SNMP Service ที่สนใจในการบอกลักษณะของคอมพิวเตอร์ในระบบเครือข่ายได้
- สามารถที่จะตรวจสอบการเปิด service ที่ไม่เหมาะสมของเครื่องที่อยู่ในเครือข่ายได้
- สามารถตรวจสอบการเปิด share directory ที่ไม่เหมาะสมของเครื่องในเครือข่ายได้
- สามารถสั่ง run โปรแกรมภายนอกให้สะดวกแก่การใช้ เช่น เพิ่มเติมโปรแกรม Script สำหรับ Update
- สามารถหา Mac Address และบอก Vendor ของผู้ผลิตอุปกรณ์นั้นได้
- สามารถดู Traffic ของเครือข่ายแต่ละเครือข่ายได้
- สามารถที่จะแสดงสถานะของเครือข่ายว่า สามารถที่จะเชื่อมต่อกับ internet ได้ตลอดเวลาหรือไม่ พร้อมกับเก็บข้อมูลเป็นสถิติ
- สามารถนำข้อมูลที่รวบรวมมาได้แสดงผลในรูปของ กราฟิกและแผนภูมิ
- สามารถนำผลลัพธ์แลดงผลออกเป็นรายงานและ system log ได้
- สามารถที่จะตรวจสอบเครื่องที่มีการใช้งานที่ไม่เหมาะสมเช่น เครื่องที่มีการเล่นเกมส์ได้
- สามารถที่จะตรวจสอบเครื่องที่อาจจะเป็นอันตรายต่อระบบเครือข่าย เช่น เครื่องที่ติดโทรจันหนอนอินเตอร์เน็ต ได้

	โปรแกรม						
คุณสมบัติ	Tivoli NetView 6.0	3Com(R) 3.0	GFI 3.2	Alchemy 4.9.7	WhatUp 8.0	OptiView 6.0	ISAG View
traceroute แสดงเส้นทางเครือข่าย	OK	OK	NOT OK	NOT OK	NOT OK	OK	OK
แสดงสถานะการเชื่อมต่อกับอินเตอร์เน็ต	OK	NOT OK	NOT OK	OK	OK	OK	OK
แสดงสถานะและเก็บสถิติการเปิดปิด	OK	NOT OK	NOT OK	OK	OK	NOT OK	OK
Scan Port	OK	NOT OK	OK	OK	OK	NOT OK	OK
แสดงการเปิดพอร์ตที่ไม่เหมาะสมของเครื่อง	NOT OK	NOT OK	OK	NOT OK	NOT OK	NOT OK	OK
แสดงระบบปฏิบัติการอย่างละเอียด	NOT OK	NOT OK	OK	NOT OK	NOT OK	NOT OK	OK
แสดง Share Folder	NOT OK	NOT OK	NOT OK	NOT OK	NOT OK	NOT OK	OK
แสดง Account Policy ของเครื่อง	NOT OK	NOT OK	NOT OK	NOT OK	NOT OK	NOT OK	OK
แสดง Group และ User ที่ใช้งานอยู่ในเครื่อง	NOT OK	NOT OK	NOT OK	NOT OK	NOT OK	NOT OK	OK
แสดง MAC Address ของเครื่อง	NOT OK	NOT OK	OK	NOT OK	NOT OK	OK	OK
แสดง SNMP	OK	NOT OK	OK	NOT OK	OK	OK	OK
แสดงข้อมูลสถิติรายงานในรูปแบบกราฟ	OK	NOT OK	NOT OK	NOT OK	OK	NOT OK	OK
เพิ่มเติมโปรแกรมภายนอก	NOT OK	NOT OK	OK	NOT OK	OK	NOT OK	OK
แสดง Traffic ของเครือข่าย	NOT OK	OK	NOT OK	NOT OK	NOT OK	NOT OK	OK
ออกรายงานของเครื่องคอมพิวเตอร์และเครือข่าย	OK	OK	OK	OK	OK	OK	OK

NOT OK	ไม่มีความสามารถ
OK	มีความสามารถ

Tivoli NetView 6.	Tivoli NetView 6.0
3Com(R) 3.0	3Com(R) Network Supervisor 3.0
GFI 3.2	GFI LANguard Scanner 3.2
Alchemy 4.9.7	Alchemy Eye 4.9.7
WhatUp 8.0	WhatUp Gold 8.0
OptiView 6.0	OptiView Console Viewer 6.0
ISAG View	ISAG View

รูปที่ 8-1 แสดงคุณสมบัติเปรียบเทียบของโปรแกรม

8.2 สรุปผล

การทำงานของโปรแกรมสามารถทำงานได้เป็นที่น่าพอใจ โดยสามารถทำให้ผู้ดูแลระบบได้รับความสะดวกในการทำงานมากขึ้น

- โปรแกรมสามารถทำงานได้ ตามวัตถุประสงค์ที่วางไว้
- สามารถนำผลลัพธ์ที่ได้จากการสำรวจไปใช้ประโยชน์ได้หลากหลาย

ข้อจำกัดของโปรแกรม

- ข้อมูลรายละเอียด SNMP ยังไม่สามารถบอกได้ทั่วทั้งหมด เนื่องจากมีเครื่องคอมพิวเตอร์ที่เปิดบริการ SNMP Service จำนวนไม่มาก
- สามารถตรวจสอบ OS จากชนิดที่มีอยู่ในรายชื่อข้อมูล OS ของ NMAP เท่านั้น เช่น Window , Linux , Unix และอุปกรณ์เช่น Router ของ CISCO, Printer ของ HP

8.3 แนวทางในการพัฒนาสำหรับผู้สนใจในอนาคต

เพื่อให้โปรแกรมมีความสามารถในการทำงานด้านการดูแลเครือข่ายและระบบคอมพิวเตอร์มากยิ่ง ๆ ขึ้น และเพื่อความสะดวกในการจัดการข้อมูลที่ได้

1. เพิ่มส่วนการแสดงโทโปโลยีของเครือข่ายว่ามีการเชื่อมต่ออยู่ในลักษณะใด
2. เพิ่มการตรวจสอบแบนเนอร์ของเซอร์วิสที่ให้บริการเพื่อความถูกต้องของการตรวจสอบ
3. เพิ่มการแสดงผลของทราฟิกของแต่ละพอร์ตของเครื่องในเครือข่าย

บรรณานุกรม

[1] ยุทธนา ลีลาศวัฒนกุล: "*คู่มือการเขียนโปรแกรม และใช้งาน Visual C++ 6.0 ฉบับโปรแกรมเมอร์*" อินโฟเพรส, 2001

[2] Gilbert Held: " *LAN Management with SNMP and RMON* ", Jonh Wiley, 1996

[3] โรเบิร์ต ลาฟอเร่,ราบินเดอร์ ศรีกิจจาภรณ์: "การเขียนโปรแกรมแบบโอโอพี ด้วยเทอร์โบและบอร์แลนด์ C++", ซีเอ็ด, 1994

[4] สุรศักดิ์ สงวนพงษ์:"*สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี*", ซีเอ็ด, 2002

[5] Joel Scambray, Stuart McClure, George Kurtz, *"Hacking Exposed Network Security & Solution"*, 2nd Edition, Mc Graw Hill

Larry J.Hughes, Jr., *"Internet Security Techniques"*, New Riders Publishing, 1961, pp. 130-144, 170-174, 192-207

[6] สุวัฒน์ ปุณณชัยยะ, ต้น ตันฑ์สุทธิวงศ์, สุพจน์ ปุณณชัยชนะ, *"เปิดโลกของ TCP/IP และโปรโตคอลของอินเตอร์เน็ต"*, โปรวิชั่น, 2543

[7] Ian Sommerville:"*Software Engineering* ", 6th Edition, Addison-Wesley, 2001

[8] Microsoft MSDN Library, Microsoft Corporation, August 1999

[9] Gary R. Wright, W. Richard Stevens, TCP-IP illustrated Volume 2, chapter 31. Addison-Wesley professional computing series.

[10] เรืองไกร รังสิพล : "เจาะ*ระบบ TCP/IP จุดอ่อนของโปรโตคอลและวิธีป้องกัน*" , โปรวิชั่น , 2001

[11] Brad Jones, Kelly Marshall, Matt Purcell, "*Sams Teach Yourself Visual C++ 6 in 21 Days*", Sams Publishing, 1998

[12] William Stallings, "*SNMP, SNMP v2, SNMP v3, RMON1 and 2 Third Edition*", Addison-Wesley, 1996

[13] J. Richard Burke, "*Network Management concepts and Practice a Hands-on Approach*", Pearson Education, Inc., 2004

เว็บไซต์อ้างอิง

[1] http://www.hack.co.za

[2] http://www.insecure.org

[3] http://winpcap.polito.it/

[4] http://www.rootshell.com

[5] http://www.codeguru.com

[6] http://www.codeproject.com

[7] http://astalavista.box.sk

[8] http://neworder.box.sk

[9] http://www.technotronic.com

[10] http://www.ussrback.com

[11] http://www.nmap.org/

[12] http://www.securitybugware.org/libnetnt/

[13] http://www.laurentconstantin.com/en/lcrzoex/

[14] http://www.ff.iij4u.or.jp/~ebata/soft/winpcaparp/