โปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เว็บ

WEB-BASED SNMP MANAGER

ธีรกฤษณ์ จันทเบญจม์ภัทร

TEERAKRIT JUNTABENJAPAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-648-225-2

โปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เวบ

WEB-BASED SNMP MANAGER

ธีรกฤษณ์ จันทเบญจม์ภัทร

TEERAKRIT JUNTABENJAPAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-648-226-2

WEB-BASED SNMP MANAGER

TEERAKRIT JUNTABENJAPAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN

COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

ISBN 974-648-226-2

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ โปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เวบ WEB-BASED SNMP MANAGER

ชื่อนักศึกษา นายธีรกฤษณ์ จันทเบญจม์ภัทร

รหัสประจำตัว 37064417

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ อาจารย์อัครินทร์ คุณกิตติ

คณะกรรมการสอบวิทยานิพนธ์	ลายมือชื่อ
อาจารย์อัครินทร์ คุณกิตติ	
รศ.ดร.วิเชียร เปรมชัยสวัสดิ์	
ดร.ประจวบ วานิชชัชวาล	
ดร.โชติพัชร์ ภรณวลัย	
ดร.จันทร์บูรณ์ สถิตวิริยวงศ์	

วัน/เดือน/ปี 25 เมษายน 2544 **เวลา** 9.00 น. เป็นต้นไป

สถานที่สอบ ณ ห้อง LAB 316 ชั้น 3 คณะเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่ 31 **เดือน** พฤษภาคม **พ.ศ.** 2544

หัวข้อวิทยานิพนธ์	โปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เวบ
นักศึกษา	นาย ธีรกฤษณ์ จันทเบญจม์ภัทร
รหัสประจำตัว	37064417
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
พ.ศ.	2544
อาจารย์ผู้ควบคุมวิทยานิพนธ์	อาจารย์ อัครินทร์ คุณกิตติ

บทคัดย่อ

ปัจจุบันการจัดการเครือข่ายผ่านระบบเวบทำให้ผู้ใช้สามารถติดต่อกับระบบเครือข่ายได้จากทุกที่ในโลกด้วยระบบอินเตอร์เน็ต และมีข้อดีคือมีการแสดงผลแบบกราฟฟิกโดยไม่ทำให้เกิดปริมาณทราฟฟิกบนระบบเครือข่ายมากมายเหมือนกับระบบอื่น เช่นระบบเอ็กซ์วินโดวส์ โดยส่วนใหญ่แล้วการจัดการเครือข่ายผ่านระบบเวบจะมีลักษณะการทำงานแบบกระจายคือผู้ใช้สามารถติดต่อกับอุปกรณ์เครือข่ายโดยตรงด้วยการใช้เวบเบราเซอร์ ซึ่งทำให้ไม่สะดวกในการควบคุมการเข้าถึงอุปกรณ์เครือข่ายของผู้ใช้ ดังนั้นงานวิจัยนี้จึงได้นำเสนอระบบการจัดการเครือข่ายที่มีรูปแบบการทำงานแบบรวมศูนย์ เพื่อควบคุมการใช้เวบเบราเซอร์ในการเข้าถึงอุปกรณ์เครือข่ายโดยตรง โดยมีต้นแบบของโปรแกรมจัดการเครือข่ายที่ทำหน้าที่เป็นตัวกลางระหว่างเวบเบราเซอร์และอุปกรณ์เครือข่าย โปรแกรมจะติดต่อกับอุปกรณ์เครือข่ายด้วยโปรโตคอล SNMP โปรแกรมมีโครงสร้างเป็นแบบเลเยอร์ประกอบด้วยแอพพลิเคชั่น เซอร์วิสและไลบรารี่เลเยอร์ ซึ่งแต่ละเลเยอร์จะประกอบด้วยโมดูลย่อยที่มีหน้าที่ต่างกันแต่ทำงานร่วมกัน โมดูลเหล่านี้ถูกออกแบบและสร้างแบบโมดูลาร์ ซึ่งทำให้โปรแกรมสนับสนุนการเพิ่มเติมโมดูลใหม่โดยไม่กระทบต่อโมดูลที่ไม่เกี่ยวข้อง นอกจากนี้ระบบมีการรักษาความปลอดภัยเพื่อเสถียรภาพในการทำงานในระดับผู้ใช้ด้วยการใช้บัญชีรายชื่อผู้ใช้และเซสชั่นคีย์ในการตรวจสอบสิทธิของผู้ใช้ รวมถึงการรักษาความปลอดภัยในระดับโมดูลซึ่งยังไม่เคยปรากฏในโปรแกรมจัดการเครือข่ายอื่นๆ ด้วยการใช้ทิกเก็ต ไฟล์ซิกเนเจอร์ และเซอร์วิสคีย์สำหรับควบคุมการทำงานและตรวจสอบโมดูล การรักษาความปลอดภัยจะอยู่ภายใต้การดูแลของโมดูลหลักตัวหนึ่งในเซอร์วิสเลเยอร์ที่มีชื่อว่าคอนเน็กเตอร์ นอกจากการรักษาความปลอดภัยแล้วในงานวิจัยครั้งนี้ยังได้นำเสนอวิธีการสืบค้นเครือข่ายเพื่อใช้ในการสร้างแผนที่เครือข่าย ซึ่งมีประโยชน์อย่างมากในการทำความเข้าใจภาพรวมของความสัมพันธ์ระหว่างเราท์เตอร์และเครือข่ายย่อยต่างๆ ของระบบเครือข่าย โดยการสืบค้นเครือข่ายจัดเป็นฟังก์ชันหนึ่งของโมดูลในเซอร์วิสเลเยอร์ของโปรแกรมจัดการเครือข่าย ส่วนท้ายงานวิจัยนี้จะนำเสนอการทดลองเพื่อทดสอบคุณสมบัติต่างๆของโปรแกรม วิเคราะห์และสรุปผลการทดลองเพื่อแสดงให้เห็นว่างานวิจัยได้สร้างรูปแบบการจัดการเครือข่ายซึ่งมีโปรแกรมต้นแบบที่มีการรักษาความปลอดภัยถึงระดับโมดูล และมีความยืดหยุ่นในการเพิ่มเติมฟังก์ชันใหม่ๆเข้าสู่ระบบได้โดยไม่ต้องแก้ไขส่วนที่ไม่เกี่ยวข้อง

Thesis Title	Web-based SNMP Manager
Student	Mr. Teerakrit Juntabenjapat
Student ID.	37064417
Degree	Master of Science
Programme	Computer Science and Information Technology
Year	2001
Thesis Advisor	Mr. Akharin Khunkitti

ABSTRACT

According to the current web-based network management applications on Internet, users can connect to the network from anywhere of the world. Unlike other graphical systems, such as X-Windows, this kind of application uses graphical user interface what consumes much less traffic. Most of web-based network management applications are based on distributed model, users can directly connect to network devices via web browsers, the disadvantage of this model is that it is rather inconvenient to control their connectings. This study is going to present a different better method, Web-based SNMP Manager, a prototype of manager program based on centralized model, this program acts as an interface between web browser and network devices, and connect to the network devices using SNMP protocol. The structure of program is composed of 3 layers, Application, Service and Library. Each layer includes modular designed modules, having their own different tasks but work cooperately. With this design, the program can support new additional modules without disturbing onto the other unrelated modules. Moreover, this system can supply not only security for system stability in user level, by applying user account list and session keys to the system to verify the user authorization, but also the security of module levels, by using tickets, file signatures and service keys to verify and control modules's operation. This level of security is never used in other network management programs yet. Those secured functions are supervised by a module in Service layer, named Connector. In addition with security of system, this study also presents a network discovery algorithm as a function of service to construct image of network map that is very useful for users to easily understand the overall relationship of routers and sub-networks. The experiment is to test the characteristics of the program, analysis and conclusion to point that Web-based SNMP manager is a network management platform, having security of users and modules level, having flexible additional modules without disturbing other unrelated modules.

กิตติกรรมประกาศ

การทำวิทยานิพนธ์ฉบับนี้จะสำเร็จลุล่วงไม่ได้ถ้าปราศจากการสนับสนุนจากบุคคลเหล่านี้ ข้าพเจ้าจึงใคร่ขอกล่าวคำแสดงความขอบพระคุณมา ณ โอกาสนี้

ขอขอบพระคุณครอบครัวของข้าพเจ้า บุพการีผู้ให้สติปัญญา ความคิดอ่าน และคอยส่งเสริมทั้งทางด้านทุนทรัพย์และกำลังใจในการศึกษา น้องชายและน้องสาวผู้ซึ่งคอยยืนเคียงข้างข้าพเจ้าเสมอมาในยามที่มีปัญหา

ขอขอบพระคุณท่านอาจารย์อัครินทร์ คุณกิตติ อาจารย์ที่ปรึกษาซึ่งคอยให้คำชี้แนะรวมทั้งช่วยแก้ไขปัญหาทำให้การทำวิทยานิพนธ์ในครั้งนี้ผ่านพ้นไปได้ด้วยดี

ขอขอบพระคุณท่านอาจารย์ สุรสิทธิ์ วรรณไกรโรจน์ สำหรับคำชี้แนะและแนวทางในการทำวิทยานิพนธ์รวมทั้งเป็นอาจารย์ที่ปรึกษาในช่วงแรกของการทำวิทยานิพนธ์นี้

ขอขอบพระคุณพี่ๆและน้องๆ ฝ่ายเครือข่ายคอมพิวเตอร์ สำนักวิจัยและบริการคอมพิวเตอร์ ซึ่งคอยดูแลเอาใจใส่ สนับสนุนอุปกรณ์ฮาร์ดแวร์ซอฟต์แวร์ และสถานที่สำหรับการทำงานวิจัยในครั้งนี้

ท้ายที่สุดขอขอบคุณเพื่อนๆ และน้องๆ ทุกคนที่คอยให้คำปรึกษา คอยเป็นกำลังใจอยู่เสมอ คุณค่าใดๆ ที่จะเกิดขึ้นจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ธีรกฤษณ์ จันทเบญจม์ภัทร

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญตาราง (ต่อ)

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การใช้โปรแกรมจัดการเครือข่ายถือเป็นการดูแลและจัดการทรัพยากรบนระบบเครือข่ายวิธีหนึ่ง โดยโปรแกรมจัดการเครือข่ายส่วนใหญ่จะมีการทำงานแบบกราฟฟิก ซึ่งจะทำให้ผู้ใช้สามารถเข้าใจเนื้อหาของงานได้สะดวกรวดเร็วขึ้น ตัวอย่างเช่น แผนที่เครือข่ายของโปรแกรมจัดการเครือข่ายซึ่งจะทำให้ผู้ใช้สามารถเข้าใจภาพรวมของระบบเครือข่ายและการเชื่อมต่อของอุปกรณ์ต่างๆบนระบบเครือข่ายได้ง่ายดาย เป็นต้น แต่ในบางครั้งที่ผู้ดูแลระบบไม่สามารถทำงานที่เครื่องที่ถูกติดตั้งโปรแกรมจัดการเครือข่ายหรือเมเนเจอร์ (Manager) ได้ จึงได้มีการเทคโนโลยีต่างเพื่อมาสนับสนุนให้ผู้ดูแลระบบสามารถทำงานกับเมเนเจอร์ได้และการทำงานก็ยังคงเป็นการทำงานแบบกราฟฟิกด้วย โดยในช่วงก่อนที่ระบบอินเตอร์เน็ตจะได้รับความนิยมนั้น เทคโนโลยีที่ถูกนำมาใช้คือ X-Windows ซึ่งเป็นระบบที่สนับสนุนการทำงานแบบกราฟฟิก ผู้ดูแลระบบจะสามารถทำงานกับเมเนเจอร์จากเครื่องอื่นๆผ่านทางระบบเครือข่ายได้ แต่เทคโนโลยีนี้จะทำให้เกิดภาระทราฟฟิกบนระบบเครือข่ายจำนวนมาก จึงไม่เหมาะกับการทำงานที่ต้องการติดต่อกับผ่านทางเครือข่ายที่มีแบนวิดต์ต่ำเช่น โมเด็ม หรือ ระบบแวน เป็นต้น ต่อมาเมื่อเข้าสู่ยุคของระบบเว็บ โปรแกรมจัดการเครือข่ายก็ถูกสร้างให้สนับสนุนการทำงานแบบเว็บ โดยการทำงานจะมีทั้งแบบแบบกระจายกับแบบรวมศูนย์ โดยแบบกระจายผู้ใช้ทั่วไปสามารถใช้เว็บเบราเซอร์หรือปลั๊กอินบนเบราเซอร์ติดต่อกับอุปกรณ์เครือข่ายได้โดยตรงด้วยโปรโตคอล HTTP หรือ SNMP การทำงานในลักษณะนี้ผู้ใช้ทั่วไปจะได้รับความสะดวก แต่สำหรับผู้ดูแลระบบแล้ว จะทำให้เกิดความไม่สะดวกในการควบคุมการเข้าถึงอุปกรณ์เครือข่ายทั้งจากโปรโตคอลจัดการเครือข่ายและการคอนฟิกไฟร์วอลล์ในกรณีที่อุปกรณ์เครือข่ายอยู่ภายใต้ไฟร์วอลล์ด้วย สำหรับแบบรวมศูนย์ จะใช้เมเนเจอร์เป็นตัวกลางในการติดต่อระหว่างผู้ใช้หรือเว็บเบราเซอร์กับอุปกรณ์เครือข่าย ลักษณะการทำงานแบบนี้จะทำให้เกิดความสะดวกในการควบคุมการเข้าถึงอุปกรณ์เครือข่ายของผู้ใช้และการจัดการไฟร์วอลล์ รูปแบบต่างๆที่มาสนับสนุนการทำงานของโปรแกรมจัดการเครือข่าย ต่างๆ ก็มีจุดด้อยและจุดแข็งผสมกัน เช่น แบบ X-Window สามารถควบคุมผู้ใช้ได้โดยผ่านระบบปฏิบัติการเช่น ระบบยูนิกซ์ เป็นต้น แต่ทำให้เกิดภาระทราฟฟิกจำนวนมาก โปรแกรมที่สนับสนุนระบบเว็บแบบกระจายก็ทำให้ผู้ดูแลไม่สะดวกในการควบคุมผู้ใช้ในการติดต่อกับอุปกรณ์เครือข่าย และแบบรวมศูนย์ก็ไม่มีการรักษาความปลอดภัยในแบบที่ X-Windows มี ดังนั้นในงานวิจัยนี้จะทำการนำเสนอรูปแบบของโปรแกรมจัดการเครือข่ายที่รวบรวมเอาคุณสมบัติเด่นต่างๆ ของแต่ละรูปแบบที่ได้กล่าวไปแล้วนั้นมาอยู่ในรูปแบบเดียว

1.2 จุดมุ่งหมายและวัตถุประสงค์ของการศึกษา

งานวิจัยในครั้งนี้มีจุดมุ่งหมายคือต้องการนำเสนอต้นแบบของโปรแกรมจัดการเครือข่ายที่สามารถควบคุมการติดต่อกับอุปกรณ์เครือข่ายของผู้ใช้ โครงสร้างของโปรแกรมสนับสนุนการเปลี่ยนแปลงและปรับปรุงโมดูลของโปรแกรมได้โดยไม่กระทบต่อโมดูลหรือส่วนอื่นๆที่ไม่เกี่ยวข้อง มีการรักษาความปลอดภัยในระดับผู้ใช้เพื่อใช้ควบคุมการทำงานของผู้ใช้ และเพิ่มการรักษาความปลอดภัยในระดับโมดูลซึ่งเป็นคุณสมบัติที่ไม่ปรากฏในโปรแกรมรูปแบบอื่นๆ ดังที่ได้กล่าวมาในหัวข้อที่ผ่านมา เพื่อใช้ควบคุมการทำงานของโมดูลและทำให้โปรแกรมสามารถทำงานได้อย่างมีเสถียรภาพมากขึ้น

1.3 แนวคิดและทฤษฎีที่ใช้ในการศึกษา

แนวคิดคือการสร้างโปรแกรมจัดการเครือข่ายแบบรวมศูนย์ เพื่อใช้เป็นตัวกลางในการติดต่อระหว่างเว็บเบราเซอร์กับอุปกรณ์เครือข่าย โดยเมเนเจอร์จะติดต่อกับเว็บเบราเซอร์ด้วยโปรโตคอล HTTP และติดต่อกับอุปกรณ์เครือข่ายด้วยโปรโตคอล SNMP นอกจากนี้เมเนเจอร์จะมีโครงสร้างแบบเลเยอร์เพื่อสนับสนุนการปรับเปลี่ยนโมดูลโดยไม่กระทบต่อส่วนอื่นๆของโปรแกรมที่ไม่เกี่ยวข้อง สำหรับการรักษาความปลอดภัยจะใช้การตรวจสอบสิทธิของผู้ใช้และโมดูลต่างๆที่ทำงานอยู่บนระบบ โดยสร้างโมดูลที่ทำหน้าที่รับผิดชอบในการดูแลและควบคุมการทำงานของผู้ใช้และโมดูล และมีการตรวจสอบสิทธิในการติดต่อสื่อสารระหว่างโมดูลด้วยกันเอง นอกจากนี้ในการจัดการเครือข่ายเมเนเจอร์จะให้โปรโตคอล SNMP สำหรับติดต่อกับอุปกรณ์ต่างๆที่อยู่บนเครือข่าย

1.4 ขอบเขตการศึกษา

งานวิจัยในครั้งนี้ ผู้ทำวิจัยต้องการนำเสนอต้นแบบแพล็ตฟอร์มของโปรแกรมจัดการเครือข่ายที่สนับสนุนการทำงานผ่านระบบเว็บที่มีการรักษาความปลอดภัยทั้งในระดับผู้ใช้และระดับโมดูล ดังนั้นเพื่อทดสอบแนวคิดของการรักษาความปลอดภัย จึงกำหนดให้ระบบที่ทำการศึกษาเป็นระบบที่อุปกรณ์เครือข่ายต่างๆ อนุญาตให้ผู้ใช้สามารถเข้าถึงได้ด้วยโปรโตคอล SNMP เท่านั้น และผู้ดูแลระบบเป็นผู้เดียวที่ทราบคอมมิวนิตี้เนม (Community Name) [12] ซึ่งเป็นรหัสผ่านของโปรโตคอล SNMP ดังนั้นหากผู้ใช้ต้องการจะติดต่อกับอุปกรณ์เครือข่าย ผู้ใช้จำเป็นที่จะต้องติดต่อผ่านเมเนเจอร์ด้วยบัญชีผู้ใช้ (User Account) ที่อยู่ภายใต้การจัดการของเมเนเจอร์ และในส่วนของตัวโปรแกรมจะประกอบด้วยหน้าที่ในการทำงานด้านการจัดการเครือข่ายเบื้องต้นเท่านั้น และสามารถขยายหน้าที่ต่างๆเพิ่มเติมได้ในงานวิจัยต่อๆไปด้วยโครงสร้างแบบเลเยอร์ที่ได้ออกแบบไว้ของเมเนเจอร์

1.5 ขั้นตอนการดำเนินงาน

เริ่มจากการศึกษาทฤษฎีและหลักการทั่วไปที่เป็นพื้นฐานในงานวิจัย โดยทฤษฎีและหลักการทั่วไปที่เป็นพื้นฐานในงานวิจัย โดยทฤษฎีและหลักการพวกนี้ได้แก่ การทำงานพื้นฐานของโปรโตคอล TCP/IP โปรโตคอล SNMP เป็นต้น และเมื่อเข้าใจหลักการต่างๆ แล้วก็จะกำหนดจุดประสงค์และขอบเขตตลอดจนเลือกเครื่องมือและอุปกรณ์ที่จะต้องนำมาใช้ในงานวิจัย เครื่องมือเหล่านี้ได้แก่คอมไพเลอร์ที่จะต้องนำมาใช้ในงานเขียนโปรแกรม เครื่องคอมพิวเตอร์และระบบปฏิบัติการหลักที่ต้องใช้ในการพัฒนาโปรแกรม หลังจากได้อุปกรณ์ครบแล้วก็จะเป็นช่วงของการพัฒนาโปรแกรม ซึ่งประกอบด้วยการออกแบบโครงสร้างหลักและย่อยของโปรแกรม การเขียนโปรแกรม การตรวจสอบและแก้ไขข้อบกพร่องต่างๆ

1.6 รายละเอียดของแต่ละบท

ในส่วนของเนื้อหาของวิทยานิพนธ์ฉบับนี้นอกจากบทแรกยังประกอบด้วยบทอื่นๆดังต่อไปนี้

บทที่ 2 ระบบเว็บและโปรโตคอล SNMP จะกล่าวถึงหลักการและทฤษฎีที่ถูกนำมาใช้ในงานวิจัยนี้ได้แก่ การทำงานของโปรโตคอล TCP/IP SNMP HTTP เว็บแอพพลิเคชั่น เทคโนโลยี ISAPI

บทที่ 3 หลักการทำงานของโปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เว็บ ในบทนี้จะกล่าวถึงแนวคิดที่ใช้ในการออกแบบโปรแกรม ตลอดจนอธิบายหน้าที่และการทำงานของส่วนประกอบต่างๆของโปรแกรม

บทที่ 4 การทดลองและผลการทดลอง เป็นการทดสอบการทำงานของโปรแกรมเพื่อแสดงให้เห็นคุณสมบัติต่างๆ ของโปรแกรมเช่น การใช้งานผ่านระบบเว็บ การรักษาความปลอดภัยทั้งในระดับผู้ใช้และระดับโมดูล การเปลี่ยนแปลงเพิ่มเติมโมดูลโดยไม่กระทบต่อส่วนอื่นๆของโปรแกรมที่ไม่เกี่ยวข้อง และการวิเคราะห์ผลลัพธ์ที่ได้จากการทดลอง เพื่อค้นหาพารามิเตอร์ที่มีผลต่อการทำงานในส่วนต่างๆ ของโปรแกรม

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ เป็นการสรุปคุณสมบัติต่างๆ ของโปรแกรมเพื่อแสดงให้เห็นว่าโปรแกรมสามารถทำงานได้ตามแนวคิดที่วางไว้ และแสดงการสรุปเปรียบเทียบคุณสมบัติต่างๆ ของโปรแกรมกับโปรแกรมจัดการเครือข่ายรูปแบบอื่นๆ เช่น โปรแกรมจัดการเครือข่ายที่ทำงานผ่านระบบ X-Windows โปรแกรมจัดการเครือข่ายที่สนับสนุนการทำงานผ่านระบบเว็บที่ผู้ใช้สามารถใช้เว็บเบราเซอร์ติดต่อกับอุปกรณ์เครือข่ายได้โดยตรงหรือที่เรียกว่าแบบกระจาย กับ แบบที่ผู้ใช้ต้องติดต่อผ่านทางเมเนเจอร์หรือแบบรวมศูนย์ เพื่อแสดงให้เห็นข้อดีและข้อเสียของโปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เว็บที่ถูกพัฒนาขึ้นในงานวิจัยครั้งนี้

บทที่ 2
ระบบเวบและโปรโตคอล SNMP

การทำงานกับคอมพิวเตอร์ในปัจจุบันได้เข้าสู่ยุคของการนิยมใช้คอมพิวเตอร์ให้ทำงานร่วมกันเป็นกลุ่มโดยเชื่อมต่อกันเป็นระบบเครือข่าย เทคโนโลยีและมาตรฐานหลากหลายต่างถูกคิดค้นขึ้นเพื่อให้ระบบเครือข่ายคอมพิวเตอร์สามารถทำงานได้อย่างมีประสิทธิภาพและรองรับความต้องการที่มีรูปแบบที่ซับซ้อนเพื่อประโยชน์ทางการศึกษา ทางการค้า การพัฒนาองค์กรและประเทศชาติ เทคโนโลยีที่ได้รับความนิยมมากที่สุดในขณะนี้คือระบบเวิลด์ไวด์เวบ ทั้งนี้เนื่องจากระบบนี้มีจุดเด่นที่สำคัญคือ มีการแสดงผลการทำงานแบบกราฟฟิกและไม่ยึดติดอยู่กับแพล็ตฟอร์มใดๆ ทำให้สะดวกต่อการนำไปประยุกต์ใช้งานในด้านต่างๆ จึงได้มีแนวคิดที่จะนำระบบเวิลด์ไวด์เวบมาใช้กับการจัดการเครือข่ายซึ่งเป็นที่มาของการทำวิจัยในครั้งนี้ ทั้งระบบเวิลด์ไวด์เวบและการจัดการเครือข่ายต่างมีทฤษฎีที่เกี่ยวข้องมากมาย เพื่อให้เกิดความเข้าใจในการติดต่อเนื้อหาของงานวิจัยในบทถัดไป ในบทนี้จะนำเสนอทฤษฎีที่เกี่ยวข้องต่างๆ เช่น ระบบเวิลด์ไวด์เวบ โปรโตคอล TCP/IP SNMP HTTP และเทคโนโลยีของคุกกี้ เวบเซิร์ฟเวอร์แอพพลิเคชั่น เป็นต้น

2.1 เครือข่ายอินเตอร์เนตและโปรโตคอล TCP/IP

เครือข่ายอินเตอร์เนตถือเป็นเครือข่ายสาธารณะที่มีขนาดใหญ่ที่สุดในโลกเชื่อมต่อเครือข่ายย่อยจากองค์กรและหน่วยงานจากประเทศต่างๆทั่วโลกเข้าด้วยกัน ระบบเครือข่ายที่มีแพล็ตฟอร์มที่แตกต่างกันบนเครือข่ายอินเตอร์เนตสามารถสื่อสารกันได้ด้วยโปรโตคอลหลัก TCP/IP [2] ซึ่งโครงสร้างประกอบด้วยเลเยอร์ 4 เลเยอร์ คือ

1. Physical Layer เป็นเลเยอร์ที่อยู่ล่างสุด เกี่ยวข้องกับอุปกรณ์การสื่อสารและควบคุมวิธีการรับส่งข้อมูลที่อยู่ในรูปแบบบิต อุปกรณ์ที่ทำงานในชั้นนี้ได้แก่สายเคเบิลต่างชนิดต่างๆ ฮับ ทรานซีฟเวอร์ เป็นต้น

2. Data Link Layer ทำหน้าที่ตรวจสอบและรวบรวมข้อมูลจากชั้นฟิสิคอลและส่งข้อมูลที่ถูกต้องไปยังชั้นอินเตอร์เนต อุปกรณ์ที่อยู่ในชั้นนี้ได้แก่ บริดจ์ สวิทชิ่งเลเยอร์สอง เป็นต้น

3. Internet Layer (IP Layer) ดูแลเกี่ยวกับการค้นหาเส้นทางการเดินทางของข้อมูลบนเครือข่ายอินเตอร์เนต เราท์เตอร์คืออุปกรณ์ที่ทำงานอยู่ในชั้นนี้ แพ็กเก็ตของข้อมูลจะสามารถถูกส่งจากเราท์เตอร์หนึ่งไปยังเราท์เตอร์ตัวอีกตัวหนึ่งก่อนจะถึงโหนดปลายทางด้วยเส้นทางที่แตกต่างกันและไม่จำเป็นที่แพ็กเก็ตจะไปถึงอย่างเป็นลำดับ โดยหน้าที่ในการรวบรวมแพ็กเก็ตข้อมูลเป็นหน้าที่ของชั้นที่อยู่ถัดไป

4. Transport Layer (TCP Layer) จัดการตรวจสอบความถูกต้องของข้อมูลและรวบรวมโดยการจัดเรียงแพ็กเก็ตของข้อมูลให้อยู่ในลำดับที่ถูกต้อง

5. Application Layer เป็นแอพพลิเคชั่นที่ถูกสร้างขึ้นมาเพื่อใช้ประโยชน์จากโปรโตคอล TCP/IP แอพพลิเคชั่นที่ในเลเยอร์นี้ได้แก่ Telnet, FTP, HTTP, SNMP, SMTP และ DNS เป็นต้น

2.2 การจัดการเครือข่ายคอมพิวเตอร์

การเชื่อมต่อคอมพิวเตอร์เข้าเป็นระบบเครือข่ายในปัจจุบันมีความสำคัญและได้รับความนิยมมากขึ้น ประโยชน์ที่ได้จากระบบเครือข่ายคือการติดต่อสื่อสารและการแบ่งปันการใช้ทรัพยากรบนระบบเครือข่าย ยิ่งระบบมีทรัพยากรมากมายเพียงใด มูลค่าของระบบก็จะยิ่งเพิ่มมากขึ้นเท่านั้น ทรัพยากรที่กล่าวถึงอาจแบ่งออกเป็นสองส่วนคือทรัพยากรส่วนที่อยู่ในเครื่องคอมพิวเตอร์ที่ถูกนำมาเชื่อมต่อเข้ากับระบบเครือข่าย ทรัพยากรเหล่านี้ได้แก่ เวลาในการประมวลผลของหน่วยประมวลผลกลาง หน่วยความจำและข้อมูลทางสารสนเทศ เป็นต้น อีกส่วนคือทรัพยากรที่อยู่บนเครือข่ายได้แก่ สวิทชิ่ง เราท์เตอร์ บริดจ์ ฮับ แพทช์พาเนล ตลอดจนสายสัญญาณประเภทต่างไม่ว่าจะเป็นสายใยแก้วนำแสง สายยูทีพี และสายโคแอ็กเชียล เป็นต้น ดังนั้นจึงมีความจำเป็นที่จะต้องมีการจัดการเครือข่ายเข้ามาดูแลรักษาและพัฒนาระบบเครือข่ายให้สามารถทำงานได้อย่างมีประสิทธิภาพ

การจัดการระบบเครือข่ายคอมพิวเตอร์คือความสามารถในการดูแล ควบคุมและวางแผนจัดการทรัพยากรที่มีอยู่บนระบบเครือข่ายเพื่อประโยชน์สูงสุดต่อการทำงาน วัตถุประสงค์ของการจัดการเครือข่ายมีดังนี้

1. เพื่อดูแลให้ระบบอยู่ในสภาพที่อย่างน้อยที่สุดสามารถให้บริการได้ โดยไม่คำนึงถึงคุณภาพและความยากง่ายในการขอใช้บริการของผู้ใช้

2. เพื่อดูแลให้ระบบเครือข่ายมีประสิทธิภาพในการให้บริการอยู่ในระดับที่ผู้ใช้บริการสามารถยอมรับได้

3. เพื่อลดค่าใช้จ่ายต่างๆที่จะเกิดขึ้นกับระบบทั้งในปัจจุบันและอนาคต

2.3 โปรโตคอล SNMP

โปรโตคอลสำหรับจัดการเครือข่ายเป็นข้อกำหนดเพื่อใช้ในการสื่อสารระหว่างสถานีจัดการเครือข่ายกับตัวแทนจัดการ โปรโตคอลที่ได้รับความนิยมมากที่สุดในปัจจุบันคือ โปรโตคอล SNMP ซึ่งถูกออกแบบมาให้เป็นโปรโตคอลในชั้นโปรแกรมประยุกต์ (Application-level protocol) ของชุดโปรโตคอล TCP/IP (TCP/IP Suit) โดยทำงานอยู่บนโปรโตคอล UDP (User Datagram Protocol) ในปัจจุบันโปรโตคอล SNMP [12] ได้รับการพัฒนาและปรับปรุงจนถึงเวอร์ชัน 4

(SNMPv4) แต่คงมีเพียงเวอร์ชันแรกคือ โปรโตคอล SNMP เวอร์ชัน 1 (SNMPv1) เท่านั้นที่มีลักษณะง่ายต่อการใช้งานตามแนวคิดเดิม ในบทความนี้จะขอกล่าวถึงเฉพาะเวอร์ชัน 1 เท่านั้น ทั้งนี้เนื่องจากโปรโตคอล SNMP ที่ใช้ในงานวิจัยเป็นเวอร์ชัน 1 นั่นเอง องค์ประกอบของโปรโตคอล SNMP

1. สถานีจัดการเครือข่าย (Management Station) เป็นเครื่องคอมพิวเตอร์ที่มีโปรแกรมจัดการเครือข่ายทำงานอยู่และสามารถปฏิบัติการโดยใช้โปรโตคอลสำหรับจัดการเครือข่าย หน้าที่ของสถานีจัดการเครือข่ายคือดูแลและควบคุมหน่วยหรืออุปกรณ์ที่อยู่บนระบบเครือข่ายโดยผ่านตัวแทนการจัดการ

2. โหนดที่ถูกจัดการ (Managed Node) เป็นอุปกรณ์ต่างๆที่อยู่บนระบบเครือข่าย อุปกรณ์เหล่านี้ ได้แก่ เครื่องคอมพิวเตอร์ สวิทชิ่ง เราท์เตอร์ บริดจ์ พริ๊นเตอร์ เป็นต้น

3. ตัวแทนการจัดการหรือเอเจ็นต์ (Management Agent) เป็นโปรแกรมที่ทำหน้าที่ติดต่อสื่อสารเพื่อแลกเปลี่ยนข้อมูลข่าวสารกับสถานีจัดการเครือข่าย โดยใช้โปรโตคอลสำหรับจัดการเครือข่าย โหนดทุกโหนดที่สามารถถูกจัดการได้จะต้องมีตัวแทนการจัดการทำงานอยู่เพื่อคอยทำหน้าที่ส่งข้อมูลที่มีลักษณะเป็นตัวแปรให้กับสถานีจัดการตามที่ถูกสถานีจัดการร้องขอมา และยังทำหน้าที่ส่งข่าวสารเป็นสัญญาณแทรบ สถานีจัดการเมื่อเกิดเหตุการณ์ฉุกเฉินกับโหนด

4. ฐานข้อมูลของตัวแทนจัดการ หรือ MIB (Management Information Base) เป็นชุดของตัวแปรที่เก็บข้อมูลของโหนดที่ถูกจัดการ แต่ละโหนดจะมีฐานข้อมูลของมันเอง โครงสร้างของฐานข้อมูลนี้จะมีลักษณะเป็นรากไม้ (Hierarchy) ของความสัมพันธ์ระหว่างกลุ่ม โครงสร้างจะเริ่มจากส่วนบนสุดที่เป็นกลุ่มที่ใหญ่หลายๆกลุ่ม แล้วแตกออกเป็นกลุ่มย่อยๆตามมาตรฐาน หน่วยงานหรือองค์กร ลักษณะงานของข้อมูล โดยชั้นล่างสุดของโครงสร้างจะเป็นข้อมูลจริง (Leave Node) แต่ละกลุ่มจะมีเลขหมายอ๊อปเจ็กท์ (Object Identifier) และชื่ออ๊อปเจ็กท์ (Object Name) ประจำกลุ่มเพื่อใช้อ้างอิงในการเข้าถึงข้อมูล ตัวอย่างในการเข้าถึงข้อมูลทั้งแบบที่ใช้หมายเลขและใช้ชื่อในการเข้าถึงมีดังนี้

.1.3.6.1.2.1.1.1 เป็นการใช้หมายเลขในการเข้าถึงข้อมูล

.iso.org.dod.internet.mgmt.mib-2.system.sysDescr เป็นการใช้ชื่อเพื่อเข้าถึงข้อมูล

5. SMI (Structure of Management Information) เป็นมาตรฐานที่ถูกกำหนดออกมาเพื่อใช้ในการจัดการโครงสร้างของ MIB ข้อกำหนดเหล่านี้ได้แก่ ชนิดของข้อมูล โครงสร้างในแต่ละส่วนของ MIB ข้อกำหนดในการเข้าถึงข้อมูลแต่ละตัว สถานะภาพของข้อมูล เป็นต้น การกำหนดโครงสร้างของข้อมูลในลักษณะนี้นอกจากจะทำให้เกิดมาตรฐานและความเข้าใจที่ตรงกันสำหรับการร้องขอและเข้าถึงข้อมูลแล้ว ยังสนับสนุนการเพิ่มชนิดของข้อมูลใหม่ๆได้ในอนาคต

6. Management Protocol ของระบบอินเตอร์เน็ตใช้ SNMP Message ในการสื่อสารระหว่างผู้จัดการเครือข่ายกับเอเจ็นต์โดยไม่ขึ้นกับระบบหรือแพล็ตฟอร์มใดๆ ทำให้ระบบที่มีสถาปัตยกรรมที่แตกต่างกันสามารถสื่อสารกันได้

เวอร์ชัน	คอมมิวนิตี้เนม	ข้อความ

รูปที่ 2.1 รูปแบบข้อความของ SNMP

จากรูป 2.1 เวอร์ชันใช้ระบุเวอร์ชันของข้อความ SNMP ถัดมาคือคอมมิวนิตี้เนม (Community Name) หรือรหัสผ่านจะถูกตรวจสอบจากเอเจ็นต์ ค่าโดยปริยายของรหัสผ่านนี้คือ Public และสุดท้ายคือข้อความ (Protocol Data Unit) ใน SNMP เวอร์ชัน 1 มีข้อความอยู่ 5 ชนิด ได้แก่ GetRequest, GetNextRequest, SetRequest, GetResponse และ Trap

GetRequest และ GetNextRequest ต่างก็เป็นข้อความที่ผู้จัดการเครือข่ายใช้ในการร้องขอข้อมูลของอ๊อบเจ็กท์ของอุปกรณ์เป้าหมายจากเอเจ็นต์ เพียงแต่ GetRequest จะระบุหมายเลขอ๊อบเจ็กท์เป้าหมายโดยตรง ในขณะที่ GetNextRequest จะต้องระบุเป็นหมายเลขของอ๊อบเจ็กท์ที่อยู่ก่อนหน้าอ๊อบเจ็กท์เป้าหมาย ซึ่งมีประโยชน์ในการขอข้อมูลอ๊อบเจ็กท์ที่อยู่ในลักษณะของตารางและไม่ทราบค่าของดรรชนีที่ใช้เข้าถึงอ๊อบเจ็กท์ตัวนั้น

SetRequest ใช้ในการร้องขอเอเจ็นต์เพื่อทำการแก้ไขค่าของอ๊อบเจ็กท์เป้าหมาย การใช้ SetRequest ต้องมีการระบุรหัสผ่านที่ถูกต้องด้วย

GetResponse เป็นข้อความที่เอเจ็นต์ตอบกลับไปยังผู้จัดการเครือข่าย โดยในข้อความนี้จะบรรจุผลลัพธ์ที่ได้รับการร้องขอจาก GetRequest GetNextRequest และ SetRequest

Trap ใช้ในการส่งสัญญาณเตือนจากเอเจ็นต์ไปยังผู้จัดการเครือข่ายเมื่อเกิดเหตุการณ์ฉุกเฉินที่มีผลต่อการทำงานของอุปกรณ์ขึ้นเช่น หน่วยความจำในการประมวลผลไม่เพียงพอ อินเตอร์เฟสของอุปกรณ์ขัดข้อง เป็นต้น

2.4 ระบบเวิลด์ไวด์เวบ

ระบบเวิลด์ไวด์เวบ [7] หรือระบบเวบเป็นระบบหนึ่งที่ทำงานอยู่บนระบบอินเตอร์เนต มีโครงสร้างทางสถาปัตยกรรมแบบไคลเอ็นต์/เซิร์ฟเวอร์ โดยมีเวบเบราเซอร์ทำหน้าที่เป็นไคลเอ็นต์และเวบเซิร์ฟเวอร์เป็นเซิร์ฟเวอร์ ทั้งสองส่วนติดต่อสื่อสารกันด้วยโปรโตคอล HTTP เวบเบราเซอร์ที่ได้รับความนิยมในปัจจุบันได้แก่ Internet Explorer และ Netscape เป็นต้น เวบเบราเซอร์มีหน้าที่เชื่อมโยงระหว่างผู้ใช้กับทรัพยากรที่อยู่บนเวบเซิร์ฟเวอร์โดยเป็นภาคแสดงผลลัพธ์ที่ถูกส่งมาจากเวบเซิร์ฟเวอร์ในรูปของเอกสารที่ใช้ไวยากรณ์แบบ HTML ในขณะที่เวบเซิร์ฟเวอร์จะเป็นภาค

ประมวลผลคำร้องของที่ได้รับจากเวบเบราเซอร์โดยเวบเซิร์ฟเวอร์สามารถติดต่อกับเซิร์ฟเวอร์ซึ่งดูแลทรัพยากรที่ไม่อยู่ในเวบเซิร์ฟเวอร์โดยใช้เทคโนโลยีเวบเซิร์ฟเวอร์แอพพลิเคชั่น

เนื่องจากระบบเวบถูกออกแบบให้มีการใช้มาตรฐาน HTML [11] สำหรับการแสดงผลบนเวบเบราเซอร์และมีมาตรฐานในการสื่อสารระหว่างเวบเซิร์ฟเวอร์และเวบเบราเซอร์ที่ชัดเจนด้วยโปรโตคอล HTTP จึงทำให้เกิดข้อดีคือ ระบบการติดต่อสื่อสารเพื่อขอใช้บริการและการแสดงผลเป็นอิสระต่อแพล็ตฟอร์มทุกประเภท ทำให้การพัฒนางานใดๆสามารถทำได้โดยใช้มาตรฐานดังกล่าวเป็นหลักเกณฑ์และง่ายต่อการขยายงานในโอกาสต่อไป สำหรับการแสดงผลที่เกิดขึ้นบนเวบเบราเซอร์ยังมีความพิเศษกว่าโปรโตคอลอื่นๆของระบบอินเตอร์เนตคือ สามารถแสดงผลลัพธ์แบบกราฟฟิกทำให้ประสิทธิภาพการทำงานของผู้ใช้เพิ่มขึ้น นอกจากนี้มาตรฐานของโปรแกรมประเภทเวบเซิร์ฟเวอร์แอพพลิเคชั่นยังขยายขีดความสามารถทำให้ระบบเวบสามารถเชื่อมต่อกับระบบอื่นๆเพื่อการทำงานที่ซับซ้อนขึ้นและสนับสนุนความต้องการในลักษณะต่างๆที่มีเงื่อนไขมากขึ้น

2.5 โปรโตคอล HTTP (HyperText Transfer Protocol)

โปรโตคอล HTTP [8] เป็นโปรโตคอลที่ทำงานโดยใช้บริการจากโปรโตคอล TCP/IP และให้บริการกับแอพพลิเคชั่นของระบบเวบ โปรโตคอล HTTP เป็นการทำงานแบบรีเควสท์/เรสพอนส์ (Request/Response) การใช้โปรโตคอลเริ่มจากเวบเบราเซอร์จะเป็นฝ่ายส่งคำร้องขอในรูปของโปรโตคอล HTTP ไปยังเวบเซิร์ฟเวอร์และเมื่อเวบเซิร์ฟเวอร์ประมวลผลคำร้องขอที่ได้รับเสร็จสิ้นก็จะส่งคำตอบกลับไปในรูปของโปรโตคอล HTTP เช่นกัน เวบเซิร์ฟเวอร์มีพอร์ทมาตรฐานบนโปรโตคอล TCP ที่พอร์ทเลขหมาย 80 คำสั่งของโปรโตคอล HTTP ได้แก่

GET เป็นคำสั่งที่ใช้อ่านข้อมูลจากเวบเซิร์ฟเวอร์ ลักษณะพิเศษของคำสั่งนี้คือสามารถอ่านข้อมูลของเวบเซิร์ฟเวอร์เฉพาะส่วนที่มีการเปลี่ยนแปลงได้

HEAD คล้ายกับคำสั่ง GET แต่จะเป็นการขอให้เวบเซิร์ฟเวอร์ส่งข้อมูลกลับมายังเวบเบราเซอร์เฉพาะส่วนที่เป็นข้อมูลในเฮดเดอร์เท่านั้นไม่รวมส่วนที่เป็น HTML คำสั่งนี้ใช้ในการทดสอบว่าเอกสาร HTML ที่ระบุใน URL มีการเปลี่ยนแปลงเกิดขึ้นหรือไม่

POST เป็นคำสั่งที่ใช้ส่งพารามิเตอร์จากเวบเบราเซอร์ไปยังเวบเซิร์ฟเวอร์ในกรณีที่เอกสาร HTML ที่ระบุใน URL ต้องการพารามิเตอร์ในการทำงาน การส่งพารามิเตอร์ของ POST จะแตกต่างจากคำสั่ง GET โดยการส่งของ GET จะใส่พารามิเตอร์ลงใน URL แล้วใช้อักขรหมายคั่นเพื่อบอกให้เวบเซิร์ฟเวอร์รู้ว่าเป็นพารามิเตอร์ทำให้ผู้ใช้สามารถมองเห็นพารามิเตอร์เหล่านั้นได้ แต่ของ POST จะถูกใส่เข้าไปในโปรโตคอล HTTP ซึ่งทำให้ผู้ใช้ไม่สามารถมองเห็นพารามิเตอร์

2.6 คุกกี้ (Cookies)

ถึงแม้ว่าโปรโตคอล HTTP จะเป็นโปรโตคอลที่มีจุดเด่นมากมายก็ตามแต่จุดด้อยของ HTTP ก็คือการเป็นโปรโตคอลแบบสเตทเลส (Stateless) โดยหลังจากภาระหน้าที่ที่เว็บเซิร์ฟเวอร์ได้รับจากเว็บเบราเซอร์เสร็จสิ้นแล้ว การเชื่อมต่อระหว่างทั้งสองฝั่งจะถูกยุติลงโดยไม่มีฝ่ายใดฝ่ายหนึ่งทำการบันทึกข้อมูลที่เกี่ยวกับการเชื่อมต่อเอาไว้ ทำให้เกิดปัญหาสำหรับระบบที่จะต้องมีการตรวจสอบสิทธิ์ของผู้ใช้ด้วยรหัสผ่าน แอพพลิเคชั่นประเภทฐานข้อมูลบนระบบซึ่งจะมีการค้นหาตำแหน่งของระเบียนข้อมูลบนฐานข้อมูลบนเว็บเซิร์ฟเวอร์ ดังนั้นจึงมีการแก้ไขปัญหาด้วยการอนุญาตให้เว็บเซิร์ฟเวอร์สามารถขอให้เว็บเบราเซอร์ทำการบันทึกข้อมูลที่เป็นประโยชน์ต่อการสื่อสารลงในฐานข้อมูลของเว็บเบราเซอร์ วิธีนี้เรียกว่าคุ๊กกี้

คุ๊กกี้มีลักษณะเป็นข้อความซึ่งอยู่ในรูปแบบ “CookieName=<Cookie value>” [13] ผู้ใช้สามารถกำหนดให้เว็บเบราเซอร์อนุญาตหรือปฏิเสธที่จะรับ/ไม่รับคุ๊กกี้จากเว็บเซิร์ฟเวอร์ได้ ทั้งนี้เนื่องจากวิธีการนี้อาจเป็นช่องทางให้พวกแฮกเกอร์สามารถเล็ดลอดเข้าสู่ระบบที่ผู้ใช้ทำงานอยู่ก็ได้ ใน Netscape คุ๊กกี้จะถูกเก็บไว้ในไฟล์ชื่อ Cookies.txt ส่วน Internet Explorer จะถูกเก็บไว้ในโฟลเดอร์ \Windows\Cookies

2.7 เว็บเซิร์ฟเวอร์แอพพลิเคชั่น

เว็บเซิร์ฟเวอร์แอพพลิเคชั่น [6] หรือเว็บแอพพลิเคชั่นเป็นแอพพลิเคชั่นประเภทหนึ่งที่ถูกนำมาใช้เพื่อขยายขีดความสามารถในการทำงานให้กับเว็บเซิร์ฟเวอร์และจัดเป็นทรัพยากรประเภทหนึ่งของเว็บเซิร์ฟเวอร์นอกเหนือจากแฟ้มข้อมูลแบบ HTML เมื่อมีการร้องขอที่จะใช้บริการเว็บแอพพลิเคชั่นจากเว็บเบราเซอร์มายังเว็บเซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ก็จะทำการขอให้ระบบปฏิบัติการโหลดโปรเซสของเว็บแอพพลิเคชั่นให้เริ่มทำงานพร้อมด้วยคำร้องขอที่เว็บเซิร์ฟเวอร์ได้รับจากเว็บเบราเซอร์ให้กับเว็บแอพพลิเคชั่น เมื่อเว็บแอพพลิเคชั่นทำงานเสร็จสิ้น ผลลัพธ์ก็จะถูกส่งกลับไปยังเว็บเบราเซอร์โดยผ่านทางเว็บเซิร์ฟเวอร์อีกครั้ง ขั้นตอนการทำงานนี้ถือเป็นการเชื่อมต่อระหว่างเว็บเบราเซอร์และเว็บเซิร์ฟเวอร์ที่เกิดขึ้นในครั้งเดียวกันและการเชื่อมต่อนี้จะสิ้นสุดลงเมื่อเว็บเซิร์ฟเวอร์ทำการส่งผลลัพธ์ทั้งหมดไปยังเว็บเบราเซอร์ เทคโนโลยีที่ใช้ในการสร้างเว็บแอพพลิเคชั่นมีอยู่หลายชนิดได้แก่ เทคโนโลยีแบบ CGI (Common Gateway Interface) [8] เป็นเทคนิคแรกที่ใช้สำหรับการสร้างเว็บแอพพลิเคชั่น โปรเซส 1 โปรเซสของเว็บแอพพลิเคชั่นจะทำงานให้กับคำร้องขอจากเว็บเบราเซอร์ 1 รีเควทส์ และ CGI ยังมีการเก็บผลลัพธ์ที่จะต้องส่งกลับไปยังเว็บเบราเซอร์ไว้ในรูปของไฟล์ทำให้การใช้งานฮาร์ดดิสในการทำงานของ CGI จะแปรผันโดยตรงกับจำนวนการร้องขอที่ได้มีเข้าสู่เว็บเซิร์ฟเวอร์ เทคโนโลยีแบบ ISAPI และ NSAPI ถูกสร้างขึ้นเพื่อลดข้อด้อยของ CGI ทำงานบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์และโปรแกรมเว็บเซิร์ฟเวอร์ที่

สนับสนุนการทำงานแบบ ISAPI มีลักษณะเป็น DLL (Dynamic Link Library) ซึ่งจะถูกเวบเซิร์ฟเวอร์โหลดให้ทำงานโดยเมื่อมีคำร้องขอจากเวบเบราเซอร์ ISAPI และ NSAPI มีลักษณะการทำงานเป็นแบบเธรด (Thread) ซึ่งจะมีขนาดเล็กกว่าโปรเซสทำให้ใช้ทรัพยากรของระบบน้อยกว่า CGI และการส่งผลลัพธ์ที่ได้จากการทำงานจะถูกส่งให้เวบเซิร์ฟเวอร์โดยไม่ต้องสร้างเป็นไฟล์บนฮาร์ดดิส จากลักษณะดังกล่าวจึงทำให้ในปัจจุบันได้มีการใช้เทคโนโลยีแบบ ISAPI และ NSAPI กันอย่างกว้างขวาง แต่เทคโนโลยีนี้ก็ยังมีข้อจำกัดคือเวบเซิร์ฟเวอร์ต้องทำงานอยู่บนระบบปฏิบัติการไมโครซอฟท์วินโดวส์เท่านั้น

2.8 สรุป

ระบบเวบทำให้การใช้ประโยชน์เครือข่ายอินเตอร์เนตมีมากขึ้นด้วยคุณสมบัติของระบบเวบที่สามารถแสดงผลที่เป็นกราฟฟิก การไม่ยึดติดกับแพล็ตฟอร์มใดๆ และถึงแม้จะมีจุดด้อยในบางจุดเช่น การทำงานที่เป็นแบบสเตทเลส แต่ก็ได้นำวิธีการที่เรียกว่าคุ๊กกี้มาใช้ทำให้ระบบเวบสามารถทำงานที่ต้องมีการจดจำสถานภาพการเชื่อมต่อของเวบเบราเซอร์กับเวบเซิร์ฟเวอร์ได้ นอกจากนี้เวบเซิร์ฟเวอร์แอพพลิเคชั่นได้ทำให้ขีดความสามาถในการทำงานร่วมกับระบบอื่นๆเพิ่มมากขึ้น เทคโนโลยี ISAPI และ NSAPI ได้ทำให้การสิ้นเปลืองทรัพยากรบนเครื่องเวบเซิร์ฟเวอร์ลดน้อยลงกว่าเทคโนโลยีแบบ CGI ด้วยคุณสมบัติดังกล่าวทำให้แอพพลิเคชั่นหลายประเภทที่ถูกสร้างขึ้นสนับสนุนการทำงานด้วยระบบเวบ รวมทั้งโปรแกรมจัดการเครือข่ายด้วยเวบ

บทที่ 3
หลักการทำงานของโปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เวบ

โปรแกรมจัดการเครือข่ายด้วยโปรโตคอล SNMP ที่มีการแสดงผลแบบกราฟฟิกในช่วงก่อนที่ระบบเวบจะได้รับความนิยมมักจะถูกสร้างขึ้นให้ทำงานบนระบบ X-Windows [2] เช่น SunNet Manager [5], [10] ของ Sun Microsystems ทำงานบนระบบปฏิบัติการ Solaris และ HP OpenView [7] ของ Hewlett-Packard ทำงานบนระบบปฏิบัติการ HP-UX เป็นต้น ข้อดีของระบบ X-Windows คือผู้ใช้สามารถใช้ X-Emulator ในการติดต่อกับโปรแกรมจัดการเครือข่ายบนเซิร์ฟเวอร์จากทุกที่ที่มีการเชื่อมต่อเครือข่ายอินเตอร์เนต แต่ข้อด้อยของระบบนี้ก็คือการสื่อสารระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์บนระบบนี้ทำให้เกิดปริมาณข้อมูลบนระบบเครือข่ายจำนวนมาก จึงไม่เหมาะที่จะใช้งานผ่านระบบเครือข่ายที่การเชื่อมต่อระหว่างเครือข่ายมีขนาดของแบนด์วิธต่ำ เช่น การเชื่อมต่อผ่านโมเด็ม เป็นต้น ในเวลาต่อมาเมื่อความนิยมในระบบเวบได้เพิ่มมากขึ้นโดยโปรแกรมหลายชนิดถูกสร้างให้สนับสนุนการทำงานบนระบบเวบเช่นเดียวกับโปรแกรมจัดการเครือข่ายซึ่งถูกสร้างขึ้นให้สามารถทำงานโดยผ่านระบบเวบด้วยเช่นกัน ได้แก่ CiscoView [3] ของ Cisco ซึ่งอนุญาตให้ผู้ใช้สามารถใช้เวบเบราเซอร์ติดต่อกับอุปกรณ์เครือข่ายโดยตรง หรือโปรแกรมที่ถูกสร้างด้วยเทคโนโลยีจาวาแอ็พเพล็ตซึ่งจะให้แอ็พเพล็ตที่อยู่บนเวบเบราเซอร์ใช้โปรโตคอล SNMP ติดต่อกับอุปกรณ์เครือข่ายโดยตรงเช่นกัน เป็นต้น การทำงานในลักษณะทำให้ผู้ใช้เกิดความสะดวกในการจัดการอุปกรณ์ที่อยู่บนเครือข่าย แต่ไม่สะดวกในการควบคุมการเข้าถึงอุปกรณ์ของผู้ใช้ วิธีหนึ่งที่สามารถทำให้เกิดความสะดวกในการควบคุมการเข้าถึงอุปกรณ์เครือข่ายโดยตรงได้คือการสร้างตัวกลางสำหรับติดต่อสื่อสารระหว่างผู้ใช้และอุปกรณ์เครือข่าย ดังเช่นงานวิจัยชิ้นนี้ และนอกจากการควบคุมการทำงานของผู้ใช้ซึ่งถือว่าเป็นการรักษาความปลอดภัยในระดับผู้ใช้แล้วในงานวิจัยนี้จะสร้างโปรแกรมจัดการเครือข่ายที่มีการรักษาความปลอดภัยในระดับโมดูล เพื่อป้องกันโมดูลแปลกปลอมที่ต้องการเข้ามาทำลายหรือลักลอบขโมยข้อมูลจากระบบ เพื่อสเถียรภาพในการทำงานที่ดีขึ้นของระบบ

ในบทนี้จะกล่าวถึงหลักการที่นำมาใช้ในการสร้างโปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เวบ โดยจะเริ่มที่แนวคิดซึ่งแสดงถึงจุดมุ่งหมายและขอบเขตของงานวิจัยในครั้งนี้ โครงสร้างหลักการด้านความปลอดภัย และส่วนประกอบต่างๆของระบบซึ่งแสดงถึงสถาปัตยกรรมที่ได้พัฒนาขึ้นโดยคำนึงถึงแนวคิด ตามลำดับดังนี้

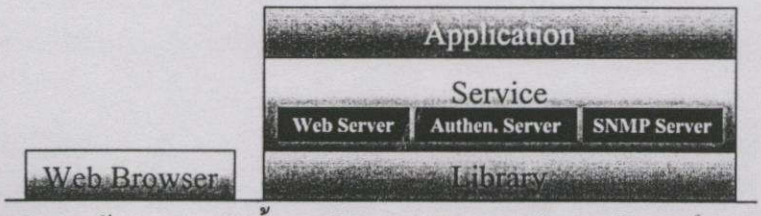
3.1 แนวคิดของระบบ

แนวคิดคือการนำเสนอแพล็ตฟอร์มของโปรแกรมจัดการเครือข่ายที่มีโครงสร้างสนับสนุนให้ระบบมีความสามารถในการปรับเปลี่ยนโมดูลเพื่อการทำงานที่ดีขึ้น สนับสนุนการติดต่อใช้งานของผู้ใช้ผ่านทางระบบเวบ และการรักษาความปลอดภัยให้กับระบบ การทำงานของโปรแกรมจะเป็นลักษณะของการทำงานร่วมกันของกลุ่มโมดูล หน้าที่การทำงานของแต่ละโมดูลจะถูกจัดแบ่งโดยอาศัยแนวคิดของการทำงานแบบเลเยอร์ กลุ่มของโมดูลที่อยู่ในเลเยอร์ล่างจะมีหน้าที่ให้บริการโมดูลที่อยู่ในเลเยอร์บนที่อยู่ติดกัน ด้วยลักษณะนี้เองที่ทำให้โปรแกรมมีแพล็ตฟอร์มที่สามารถแก้ไขเปลี่ยนแปลงเฉพาะส่วนหรือเฉพาะโมดูลได้โดยไม่กระทบต่อส่วนอื่นที่ไม่เกี่ยวข้อง ในด้านของการรักษาความปลอดภัยได้มีการตรวจสอบสิทธิเพื่อใช้ในการควบคุมการทำงานของผู้ใช้และโมดูลบนระบบ ผู้ใช้จะต้องแสดงตนทุกครั้งที่ต้องการเข้าใช้บริการจากระบบโดยใช้ชื่อและรหัสผ่านในบัญชีผู้ใช้เพื่อตรวจสอบว่าผู้ใช้เป็นใคร ได้ทำการลงทะเบียนกับระบบไว้หรือไม่ มีสิทธิในการขอใช้บริการจากระบบในระดับใด นอกจากตรวจสอบผู้ใช้แล้วก็ยังมีการตรวจสอบโมดูลขณะเริ่มทำงานด้วยว่าได้รับอนุญาตให้ทำงานหรือไม่และต้องการใช้บริการใดจากระบบบ้าง ระบบที่ถูกพัฒนาขึ้นนี้จะสนับสนุนให้ผู้ใช้สามารถติดต่อเพื่อเรียกใช้บริการจากระบบได้โดยผ่านทางระบบเวบทำให้เกิดความสะดวกในการใช้งานมากขึ้นเนื่องจากผู้ใช้สามารถติดต่อกับระบบได้จากที่ไหนก็ได้ที่มีการเชื่อมต่อเครือข่ายกับระบบอินเตอร์เนต นอกจากนี้ในส่วนของการใช้โปรโตคอลของการจัดการเครือข่าย ซึ่งในงานวิจัยชิ้นนี้ได้ใช้โปรโตคอล SNMP นั้น ทุกครั้งที่ผู้ใช้ร้องขอบริการที่ต้องมีการใช้โปรโตคอล SNMP ระบบจะทำหน้าที่ในการเรียกใช้โปรโตคอล SNMP แทน ดังนั้นการใช้โปรโตคอล SNMP จึงเกิดขึ้นที่ระบบเพียงแห่งเดียว ทำให้สามารถควบคุมการใช้งานโปรโตคอล SNMP ของผู้ใช้ได้ รายละเอียดของแนวคิดที่กล่าวไปแล้วนี้จะถูกอธิบายให้ชัดเจนยิ่งขึ้นในหัวข้อถัดไป

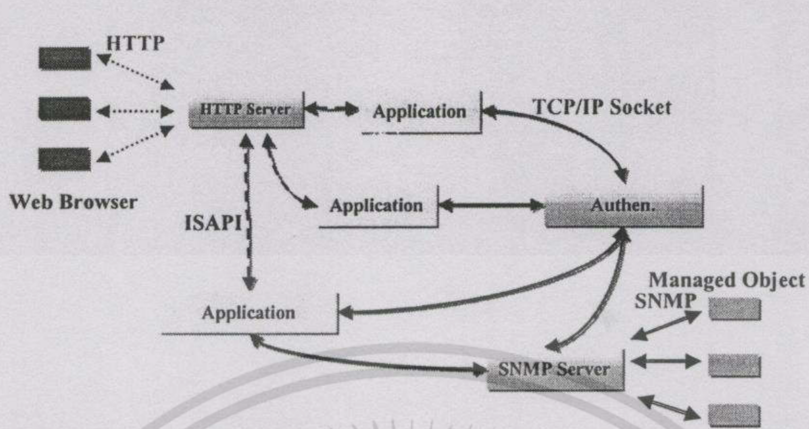
3.2 โครงสร้างของระบบ

ระบบจัดการเครือข่ายในงานวิจัยนี้ประกอบด้วยส่วนไคลเอ็นต์ทำหน้าที่เป็นยูสเซอร์อินเตอร์เฟสกับส่วนเซิร์ฟเวอร์หรือผู้จัดการเครือข่ายซึ่งทำหน้าที่ให้บริการแก่ผู้ใช้ โดยที่ไคลเอ็นต์สามารถติดต่อเพื่อขอใช้บริการจากผู้จัดการเครือข่ายด้วยเวบเบราเซอร์ซึ่งจะใช้โปรโตคอล HTTP ในการสื่อสารกับ HTTP เซิร์ฟเวอร์ของผู้จัดการเครือข่าย ด้วยหลักการดังกล่าวทำให้ผู้ใช้สามารถติดต่อกับระบบผ่านทางระบบอินเตอร์เนต ในส่วนของผู้จัดการเครือข่ายมีโครงสร้างการทำงานเป็นแบบเลเยอร์ แต่ละเลเยอร์จะประกอบด้วยกลุ่มของโมดูลซึ่งมีการแบ่งหน้าที่ออกเป็นสัดส่วน ข้อดีของโครงสร้างลักษณะนี้คือ เมื่อมีการเปลี่ยนแปลงในระดับชั้นใดๆ จะไม่มีผลกระทบต่อระดับชั้นอื่น ตราบใดที่จุดเชื่อมต่อระหว่างระดับชั้นไม่ถูกเปลี่ยนแปลง ทำให้สามารถลดเวลาในการปรับปรุง

ระบบลงและมีอายุการใช้งานยาวนานขึ้น แต่ข้อเสียคือในระยะแรกเริ่มต้องใช้เวลาในการพัฒนามากกว่าโปรแกรมปกติเพราะต้องมีขั้นตอนในการกำหนดวิธีการและรูปแบบของการสื่อสารระหว่างระดับชั้น และการออกแบบที่สนับสนุนการปรับเปลี่ยนโมดูลโดยไม่กระทบต่อโมดูลในส่วนอื่นๆ เป็นต้น นอกจากนี้หากโครงสร้างถูกออกแบบมาไม่เหมาะสมเช่นมีระดับชั้นในการทำงานมากเกินไปจะส่งผลกระทบต่อประสิทธิภาพการทำงานโดยรวม คืออาจทำให้ระบบทำงานช้าลงเนื่องจากขั้นตอนการทำงานที่มากเกินไปนั่นเอง นอกจากนี้ในการรักษาความปลอดภัย ระบบยังมีการตรวจสอบสิทธิทั้งในส่วนของผู้ใช้และการทำงานภายในของผู้จัดการเครือข่ายเพื่อให้เกิดเสถียรภาพแก่ระบบ จากลักษณะดังกล่าวนี้จะเป็นปัจจัยที่ใช้ในการออกแบบโครงสร้างของระบบ ซึ่งมีโครงสร้างดังรูปที่ 3.1 จากรูปจะเห็นว่าผู้จัดการเครือข่ายจะประกอบด้วย 3 เลเยอร์ ได้แก่ ชั้นแอพพลิเคชั่นเป็นชั้นของโมดูลที่จะถูกเรียกระบุโดยผู้ใช้เมื่อต้องการขอใช้บริการจากระบบ แต่ละโมดูลทำงานโดยการเรียกใช้บริการพื้นฐานของระบบและจัดการเกี่ยวกับการแสดงผลบนเวบเบราเซอร์ของผู้ใช้ ชั้นเซอร์วิสจะเป็นส่วนที่ให้บริการพื้นฐานแก่ชั้นแอพพลิเคชั่น โดยบริการเหล่านี้ได้แก่ การจัดการเกี่ยวกับผู้ใช้และโมดูลต่างๆบนระบบ การปรับเปลี่ยนโมดูลบนระบบ การตรวจสอบผู้ใช้เมื่อผู้ใช้ต้องการร้องขอใช้บริการโมดูลที่อยู่ในชั้นแอพพลิเคชั่น การตรวจสอบโมดูลของแอพพลิเคชั่นเมื่อมีการขอใช้บริการจากเซอร์วิส หน้าที่ตรวจสอบทั้งสองอย่างนี้จะกระทำโดยออเทนทิเคชั่นเซิร์ฟเวอร์ซึ่งอยู่ในชั้นเซอร์วิส นอกจากนี้ในชั้นเซอร์วิสยังประกอบด้วย HTTP เซิร์ฟเวอร์ซึ่งใช้รับรีเควสท์จาก HTTP ไคลเอ็นต์ และส่งรีเควสท์ต่อไปยังแอพพลิเคชั่นและรอจนกระทั่งได้รับผลลัพธ์จากแอพพลิเคชั่นจึงใช้โปรโตคอล HTTP ส่งผลลัพธ์นั้นกลับไปยัง HTTP ไคลเอ็นต์ ทำให้เซสชั่นในการร้องขอของ HTTPไคลเอ็นต์คงอยู่ตลอดจนการทำงานในชั้นแอพพลิเคชั่นเสร็จสิ้น ด้วยคุณสมบัตินี้จะสามารถหลีกเลี่ยงปัญหาอันเนื่องมาจากการกรองเลขหมายไอพีของไฟร์วอลล์ [4] ในกรณีที่เวบเบราเซอร์ของผู้ใช้และระบบอยู่ต่างเครือข่ายกัน นอกจากนี้ในส่วนของการใช้โปรโตคอลจัดการเครือข่ายคือโปรโตคอล SNMP นั้น ระบบจะมีเซอร์วิสที่คอยให้บริการเกี่ยวกับโปรโตคอล SNMP โดยเซอร์วิสดังกล่าวจะทำหน้าที่ให้บริการแก่ผู้ใช้โดยการใช้คำสั่งของโปรโตคอล SNMP ในการติดต่อกับอุปกรณ์เครือข่าย ผู้ใช้บนระบบจะไม่สามารถติดต่อกับอุปกรณ์เครือข่ายได้โดยตรง ทำให้สามารถควบคุมการใช้งานโปรโตคอล SNMP ในการเข้าถึงอุปกรณ์เครือข่ายของผู้ใช้ได้ ชั้นสุดท้ายคือชั้นไลบรารี่เป็นชั้นของโมดูลที่ให้บริการแก่เซอร์วิส โมดูลเหล่านี้จะทำหน้าที่ติดต่อกับระบบปฏิบัติการแทนเซอร์วิส



รูปที่ 3.1 โครงสร้างแบบลำดับชั้นของโปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เวบ



รูปที่ 3.2 การสื่อสารระหว่างโมดูลด้วยโปรโตคอลต่างๆ

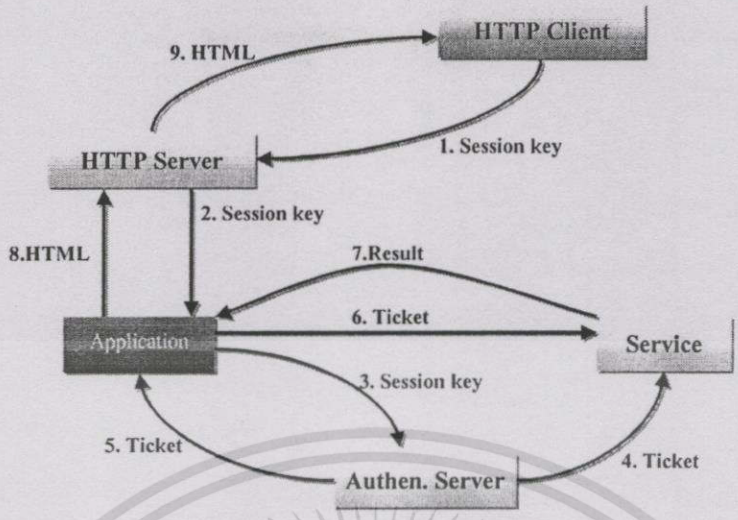
เนื่องจากระบบมีโครงสร้างเป็นเลเยอร์ และมีการทำงานร่วมกันของโมดูลในแอพพลิเคชั่นและเซอร์วิสเลเยอร์ จึงต้องมีวิธีการสื่อสารเพื่อใช้ในการแลกเปลี่ยนข้อมูล ซึ่งในงานวิจัยครั้งนี้ได้เลือกใช้เทคโนโลยี ISAPI และ TCP/IP Socket มาใช้ในการสื่อสาร ดังรูปที่ 3.2 จะเป็นรูปการติดต่อสื่อสารของโมดูลบนระบบด้วยโปรโตคอลต่างๆ โดย HTTP Client จะติดต่อกับ HTTP Server ด้วยโปรโตคอล HTTP ในขณะที่ HTTP Server จะติดต่อกับแอพพลิเคชั่นด้วย ISAPI ซึ่งเป็นลักษณะการทำงานแบบเวบเซิร์ฟเวอร์แอพพลิเคชั่น (รายละเอียดเพิ่มเติมอยู่ในบทที่ 2 หัวข้อเวบเซิร์ฟเวอร์แอพพลิเคชั่น) โดยความสัมพันธ์ระหว่าง HTTP Server และแอพพลิเคชั่นจะถูกกล่าวถึงอีกครั้งในแอพพลิเคชั่นของหัวข้อที่ 3.4 ส่วนประกอบของระบบ สำหรับการสื่อสารระหว่างแอพพลิเคชั่นกับเซอร์วิสหรือเซอร์วิสกับเซอร์วิสซึ่งเป็นการสื่อสารระหว่างโมดูลที่อยู่ในเลเยอร์เดียวกันจะใช้ Socket ของโปรโตคอล TCP/IP และในส่วนของ SNMP Server ก็จะติดต่อกับอุปกรณ์เครือข่ายด้วยโปรโตคอล SNMP

3.3 การรักษาความปลอดภัย

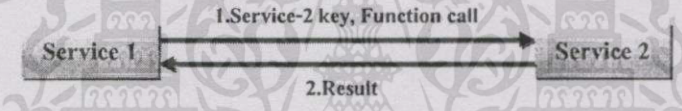
เนื่องจากระบบถูกออกแบบมาเพื่อสนับสนุนการทำงานแบบไทม์แชร์ลิ่งและมีการทำงานร่วมกันของโมดูลต่างๆที่อยู่ในแต่ละเลเยอร์ ดังนั้นเพื่อให้เกิดเสถียรภาพในการทำงานของระบบจึงมีการตรวจสอบและควบคุมการใช้บริการของผู้ใช้และการทำงานของโมดูลบนระบบ สำหรับการตรวจสอบได้แก่ การตรวจสอบตัวตน เลขหมายไอพีของเครื่องคอมพิวเตอร์ของผู้ใช้เมื่อแรกเข้าใช้งานระบบ และระดับสิทธิของผู้ใช้เพื่อตรวจสอบระดับของบริการที่ผู้ใช้ควรจะได้รับ การตรวจสอบโมดูลแบ่งออกเป็น 2 แบบตามชั้นที่โมดูลประจำอยู่ คือเมื่อระบบเริ่มทำการสตาร์ทอัพ ระบบจะทำการตรวจสอบโมดูลในชั้นเซอร์วิสด้วยเทคนิคซิกเนเจอร์ของโมดูลหรือเรียกว่าไฟล์ซิกเนเจอร์ (File Signature) เพื่อพิสูจน์ว่าโมดูลเป็นโมดูลเดียวกับที่ได้ถูกติดตั้งบนระบบ โดยค่าไฟล์ซิกเนเจอร์

ของโมดูลได้มาจากการคำนวณหาค่าเช็คซัม (Check Sum) ของโมดูล ไฟล์ซิกเนเจอร์จะเปลี่ยนไปหากมีการเปลี่ยนแปลงโค๊ดในโมดูล ขณะที่การตรวจสอบโมดูลของชั้นแอพพลิเคชั่นจะเป็นการตรวจสอบความสอดคล้องของการใช้ฟังก์ชันหรือบริการจากเซอร์วิส ความสอดคล้องดังกล่าวได้แก่ สถานะ เวอร์ชัน และฟังก์ชันของเซอร์วิส เป็นต้น ซึ่งความสอดคล้องนี่เองที่จะเป็นตัวกำหนดสถานะภาพโมดูลของชั้นแอพพลิเคชั่น นอกจากนี้เมื่อโมดูลของแอพพลิเคชั่นทำงาน ระบบจะสามารถควบคุมการใช้บริการจากเซอร์วิสด้วยการตรวจสอบฟังก์ชันที่โมดูลทำการเรียกใช้ว่าถูกต้องตรงกับที่ระบบได้ทำการบันทึกไว้หรือไม่ ดังนั้นเมื่อแรกเริ่มที่ได้ทำการติดตั้งโมดูลของแอพพลิเคชั่นเข้าสู่ระบบ รายชื่อฟังก์ชันหรือบริการที่โมดูลต้องการใช้จำเป็นที่จะต้องถูกต้องตรงกันเสมอ มิเช่นนั้นโมดูลจะไม่สามารถทำงานได้จนเสร็จสิ้นการทำงาน นอกจากการตรวจสอบดังกล่าวแล้ว ระบบยังมีการควบคุมการทำงานของผู้ใช้และโมดูลบนระบบ โดยมีพารามิเตอร์สำคัญที่ใช้สำหรับการควบคุม 2 ตัวคือ เซสชั่นคีย์ซึ่ง HTTP ไคลเอ็นต์จะได้รับจากการล็อกอินเข้าใช้ระบบของผู้ใช้ กับทิกเก็ตคีย์ของแอพพลิเคชั่นซึ่งแอพพลิเคชั่นจะได้รับหลังจากการตรวจสอบของระบบเพื่อนำไปใช้อ้างอิงในการขอใช้บริการจากเซอร์วิสต่างๆของระบบ จากรูปที่ 3.3 จะเป็นการแสดงขั้นตอนการทำงานของระบบที่เกี่ยวข้องกับกุญแจทั้งสองแบบ ทุกครั้งที่ HTTP ไคลเอ็นต์ต้องการใช้บริการจากแอพพลิเคชั่นบนระบบ HTTP ไคลเอ็นต์จะส่งรีเควสท์ด้วยโปรโตคอล HTTP ซึ่งประกอบด้วยเซสชั่นคีย์และชื่อของแอพพลิเคชั่นโมดูลไปยังแอพพลิเคชั่นโดยผ่าน HTTP เซิร์ฟเวอร์ เมื่อแอพพลิเคชั่นเริ่มทำงานและต้องการใช้บริการจากเซอร์วิสใดๆบนระบบ แอพพลิเคชั่นจำเป็นที่จะต้องมีทิกเก็ตคีย์เพื่อใช้ในการติดต่อกับเซอร์วิสเสียก่อน โดยแอพพลิเคชั่นจะต้องนำเซสชั่นคีย์ของผู้ใช้ส่งให้กับออเทนทิเคชั่นเซิร์ฟเวอร์ตรวจสอบความถูกต้องเพื่อรับทิกเก็ตคีย์ที่จะถูกสร้างโดยออเทนทิเคชั่นเซิร์ฟเวอร์เพื่อนำไปใช้อ้างอิงในการติดต่อและขอใช้บริการจากเซอร์วิสของระบบโดยตรง ออร์เทนทิเคชั่นเซิร์ฟเวอร์จะส่งทิกเก็ตคีย์ให้กับแอพพลิเคชั่นและเซอร์วิสที่แอพพลิเคชั่นต้องการขอใช้บริการ แอพพลิเคชั่นสามารถใช้ทิกเก็ตคีย์ติดต่อกับเซอร์วิสโดยตรงจนกระทั่งงานของแอพพลิเคชั่นเสร็จสิ้นและส่งผลลัพธ์กลับไปยัง HTTP ไคลเอ็นต์ผ่านทาง HTTP เซิร์ฟเวอร์

สำหรับการติดต่อสื่อสารที่เกิดในชั้นเดียวกันนั้นจะพบเพียงในชั้นเซอร์วิส โดยเป็นการติดต่อเพื่อขอใช้บริการระหว่างเซอร์วิสโปรเซสด้วยกันเอง โดยกำหนดให้แต่ละเซอร์วิสมีเซอร์วิสคีย์ประจำเซอร์วิสและทุกเซอร์วิสบนระบบจะรู้จักเซอร์วิสคีย์ของเซอร์วิสตัวอื่นๆที่มีการติดต่อด้วยเสมอ เซอร์วิสที่เป็นไคลเอ็นต์จะต้องใช้เซอร์วิสคีย์ของเซอร์วิสที่เป็นเซิร์ฟเวอร์อ้างอิงในการขอใช้บริการเสมอ เซอร์วิสคีย์ของเซอร์วิสจะถูกสร้างขึ้นโดยระบบตั้งแต่มีการสั่งให้ระบบเริ่มทำงาน พิจารณาจากรูป 3.4 ซึ่งแสดงการติดต่อสื่อสารระหว่างเซอร์วิสโปรเซสด้วยกัน



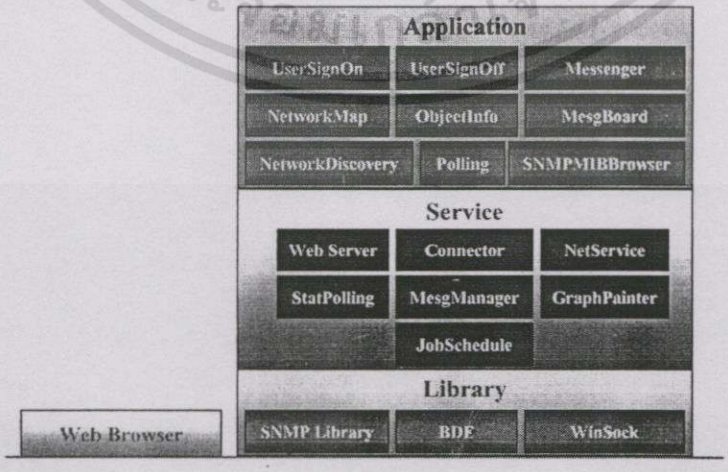
รูปที่ 3.3 ขั้นตอนการทำงานของระบบที่เกี่ยวข้องกับเซสชั่นคีย์และทิกเก็ต



รูปที่ 3.4 ขั้นตอนการสื่อสารระหว่างเซอร์วิสโปรเซส

3.4 ส่วนประกอบของระบบ

ในหัวข้อที่ 3.2 ได้กล่าวถึงโครงสร้างของระบบซึ่งประกอบด้วยเลเยอร์ต่างๆรวมถึงหน้าที่ของแต่ละเลเยอร์ไปบ้างแล้ว ในหัวข้อต่อไปนี้จะกล่าวถึงรายละเอียดที่มากขึ้นของแอพพลิเคชั่นและฟังก์ชันของเซอร์วิสต่างๆที่มีบนระบบ ดังรูปที่ 3.5



รูปที่ 3.5 เซอร์วิสและแอพพลิเคชั่นของโปรแกรมจัดการเครือข่ายด้วย SNMP โดยใช้เวบ

ผู้จัดการเครือข่ายมีหน้าที่ประมวลผลรีเควสท์ที่ได้รับมาจากเวบเบราเซอร์แล้วส่งผลลัพธ์กลับไปยังเวบเบราเซอร์ในรูปของเอกสาร HTML ผู้จัดการเครือข่ายจะประกอบด้วยกลุ่มของโมดูลทำงานร่วมกัน โดยสามารถแบ่งกลุ่มของโมดูลเหล่านี้ออกเป็น 3 เลเยอร์ คือเลเยอร์แอพพลิเคชั่นซึ่งเป็นชั้นบนสุด เลเยอร์เซอร์วิสเป็นชั้นที่ถัดลงมา และเลเยอร์ไลบารี่รี แต่ละเลเยอร์จะถูกกล่าวถึงต่อไปนี้ตามลำดับ

1. เลเยอร์แอพพลิเคชั่น เป็นเลเยอร์ที่อยู่บนสุดของแพล็ตฟอร์มมีหน้าที่ให้บริการแก่ผู้ใช้โดยผู้ใช้จะเรียกใช้บริการจากแอพพลิเคชั่นด้วยเวบเบราเซอร์ โมดูลที่อยู่ในชั้นแอพพลิเคชั่นจะเป็นโมดูลประเภทเวบแอพพลิเคชั่นซึ่งจะเริ่มทำงานโดยการเรียกของเวบเซิร์ฟเวอร์ ขณะที่แอพพลิเคชั่นทำงานสามารถเรียกใช้บริการที่ต้องการจากเซอร์วิสเพื่อนำผลลัพธ์ที่ได้มาใช้ในการทำงานของตัวเองและรับผิดชอบการแสดงผลบนเวบเบราเซอร์ของผู้ใช้ด้วยเอกสาร HTML ที่แอพพลิเคชั่นสร้างขึ้นขณะที่กำลังทำงาน เมื่อแอพพลิเคชั่นทำงานเสร็จสิ้นโปรเซสของแอพพลิเคชั่นก็จะสลายตัวไป ขั้นตอนการขอใช้บริการจากเซอร์วิสของแอพพลิเคชั่นจะถูกกล่าวถึงในหัวข้อการทำงานของระบบซึ่งจะอยู่ถัดไป โมดูลที่อยู่ในชั้นแอพพลิเคชั่นได้แก่

1.1 UserSignOn ให้บริการแก่ผู้ใช้เมื่อผู้ใช้ต้องการล็อกอินเข้าสู่ระบบ โดยจะรับข้อมูลของผู้ใช้ผ่านทาง Identify.html และทำการตรวจสอบสิทธิของผู้ใช้จากชื่อกับรหัสผ่านที่ได้รับจากผู้ใช้และตรวจสอบเลขหมายไอพีของเครื่องคอมพิวเตอร์ที่ผู้ใช้ใช้ติดต่อกับระบบ ทั้งนี้เพื่อป้องกันเลขหมายไอพีที่ไม่พึงประสงค์เข้ามารบกวนระบบ ข้อมูลที่ผู้ใช้ต้องส่งให้กับแอพพลิเคชั่นคือชื่อและรหัสผ่านของผู้ใช้ ในขณะที่เลขหมายไอพีจะแฝงไปกับแพ็กเก็ตข้อมูลของชื่อและรหัสผ่าน ถ้าข้อมูลทั้งหมดผ่านการตรวจสอบ UserSignOn จะให้ผลลัพธ์เป็นเมนูของผู้ใช้และเซสชั่นคีย์ของผู้ใช้เพื่อใช้อ้างอิงแทนชื่อผู้ใช้เมื่อผู้ใช้ติดต่อกับระบบเพื่อขอใช้บริการอื่นๆจากระบบ เซสชั่นคีย์ดังกล่าวนี้จะถูกส่งมาเก็บไว้ที่เวบเบราเซอร์โดยใช้เทคนิคของคุ๊กกี้ ทำให้เวบเบราเซอร์สามารถนำเซสชั่นคีย์มาใช้ได้เสมอเมื่อมีการติดต่อกับระบบ สำหรับเซอร์วิสที่แอพพลิเคชั่นทำการติดต่อคือคอนเนคเตอร์ โดยเรียกใช้บริการฟังก์ชัน SignOn ซึ่งมีหน้าที่เกี่ยวกับการตรวจสอบข้อมูลการแสดงตนของผู้ใช้จากฐานข้อมูลของระบบโดยตรง และฟังก์ชัน MenuList เพื่อขอข้อมูลของรายการนำมาสร้างเป็นเมนูของผู้ใช้

1.2 UserSignOff ทำหน้าที่แจ้งให้ระบบทราบถึงการขอเลิกใช้งานระบบของผู้ใช้ โดยแอพพลิเคชั่นจะขอให้คอนเนคเตอร์เพื่อเรียกใช้ฟังก์ชัน SignOff เพื่อจัดการกับข้อมูลของผู้ใช้บนฐานข้อมูลของระบบโดยตรง ในเวบเบราเซอร์ที่ผู้ใช้มีการเรียกใช้แอพพลิเคชั่นนี้เมื่อต้องการหยุดใช้งานกับระบบ เซสชั่นคีย์ซึ่งถูกส่งไปเก็บไว้ที่เวบเบราเซอร์ในรูปของคุ๊กกี้จะถูกลบข้อมูลออกไป ทำให้เวบเบราเซอร์ไม่สามารถเซสชั่นคีย์ที่มีอยู่มาใช้ได้อีก แต่ในกรณีที่ผู้ใช้ไม่ได้ทำการยุติการใช้งานระบบอย่างถูกต้องค่าของกุญแจรหัสก็จะยังอยู่ในคุกกี้ ทำให้ผู้ใช้ที่ไม่มีการร้องขอเข้าระบบ

อย่างถูกต้องสามารถเข้าใช้บริการของระบบได้ ดังนั้นระบบจะมีการตรวจสอบไทม์เอาท์เพื่อใช้คัดผู้ใช้ที่ขาดการติดต่อกับระบบนานเกินเวลาที่กำหนดไว้ออกจากระบบ ซึ่งจะเป็นหน้าที่ของเซอร์วิสในระบบ การเรียกใช้แอพพลิเคชั่นทำโดยเลือกรายการ Sign Off ในเมนู

1.3 SNMPMIBBrowser ให้บริการและแก้ไขข้อมูลที่อยู่บน MIB ของอุปกรณ์บนระบบเครือข่ายแก่ผู้ใช้ แอพพลิเคชั่นจะติดต่อกับเน็ตเซอร์วิส (NetService) เพื่อเรียกใช้ฟังก์ชันเก็ทรีเควสท์(GetRequest) และเก็ทเน็กรีเควทส์(GetNextRequest)ในการสืบค้นข้อมูลแบบคำตอบเดียวกับข้อมูลที่เป็นชุด และฟังก์ชันเซ็ทรีเควสท์(SetRequest)สำหรับแก้ไขข้อมูลที่อยู่บน MIB การเรียกใช้บริการจากแอพพลิเคชั่นนี้สามารถเรียกได้จากรายการชื่อ MIB Browser ในเมนู ข้อมูลที่ผู้ใช้ต้องป้อนให้กับแอพพลิเคชั่นเป็นอินพุตได้แก่ อ๊อปเจ็กไอเด็นติฟายเออร์ (Object Identifier) คอมมิวนิตี้เนม(Community Name) เอสเอ็นเอ็มพีพอร์ท(SNMP Port) และเลขหมายไอพี และต้องมีค่าของอ๊อปเจ็กต์ (Object Value) ในกรณีที่ต้องการแก้ไขข้อมูลที่อยู่บน MIB

1.4 NetworkDiscovery ทำหน้าที่ติดต่อกับเน็ทเซอร์วิสเพื่อขอใช้บริการฟังก์ชัน SNMPDiscovery โดยข้อมูลที่ต้องใช้เป็นพารามิเตอร์ได้แก่ เลขหมายไอพี คอมมิวนิตี้เนม และพอร์ท SNMP เอเจ็นท์ของสีดโหนด แอพพลิเคชั่นจะสิ้นสุดการทำงานและรายงานให้ผู้ใช้ทราบถึงการตอบรับหรือปฏิเสธการให้บริการของเซอร์วิสโดยไม่รอให้เซอร์วิสทำการสืบค้นเครือข่ายจนเสร็จสิ้น ทั้งนี้เนื่องจากการสืบค้นต้องใช้เวลาระยะหนึ่งซึ่งขึ้นอยู่กับขนาดของเครือข่ายที่ทำการสืบค้น ดังนั้นจึงไม่เหมาะสมนักที่จะให้ผู้ใช้รอจนกระทั่งการสืบค้นเสร็จสิ้น

1.5 NetworkMap และ ObjectInfo มีหน้าที่แสดงแผนที่เครือข่ายบนเวบเบราเซอร์และข้อมูลของอ๊อปเจ็กต์ที่อยู่ในแผนที่เครือข่าย ผู้ใช้สามารถเรียกใช้แอพพลิเคชั่นได้จากการเลือกรายการ Network Map ในเมนู NetworkMap มีหน้าที่นำแผนที่ของระบบเครือข่ายซึ่งได้มากจากการทำเน็ทเวิร์คดิสคอฟเวอรี่มาแสดงบนเวบเบราเซอร์ในรูปของเอกสาร HTML โดยภายในเอกสารจะเก็บข้อมูลของแต่ละอ๊อปเจ็กท์บนแผนที่เพื่อใช้เป็นอินพุตของ ObjectInfo ข้อมูลดังกล่าวได้แก่ ชื่อและชนิดของอ๊อปเจ็กท์ โดยชนิดของอ๊อปเจ็กท์จะมี 2 ชนิดคือโหนดที่เป็นเราท์เตอร์และโหนดที่เป็นเครือข่ายย่อย เมื่อใดก็ตามที่ผู้ใช้ทำการเลือกที่จะดูรายละเอียดของอ๊อปเจ็กท์บนแผนที่โดยการใช้เม้าท์คลิ๊กลงบนตำแหน่งของอ๊อปเจ็กท์บนแผนที่ เท่ากับว่าผู้ใช้ได้เรียกใช้บริการของ ObjectInfo โดยชื่อและชนิดของอ๊อปเจ็กท์ที่ถูกเก็บไว้ในเอกสาร HTML จะถูกใช้เป็นอินพุตของ ObjectInfo นำไปใช้ในการสืบค้นข้อมูลจากฐานข้อมูลและนำมาแสดงต่อผู้ใช้ในรูปของเอกสาร HTML ในกรณีของโหนดที่เป็นเราท์เตอร์ ข้อมูลที่จะถูกนำมาแสดงต่อผู้ใช้ได้แก่ ลักษณะทั่วๆไปของเราท์เตอร์ เช่น ชื่อบริษัทที่ผลิต เวลาทั้งหมดตั้งแต่เราท์เตอร์เริ่มทำงาน ชื่อผู้ดูแลรับผิดชอบ จำนวนอินเตอร์เฟส และรายละเอียดปลีกย่อยของแต่ละอินเตอร์เฟส เพื่อที่จะให้ผู้ใช้สามารถเข้าใจลักษณะของเราท์เตอร์ได้อย่างคล่าวๆ ส่วนกรณีอ๊อปเจ็กท์ที่เลือกเป็นโหนดของเครือข่ายย่อย ข้อมูลที่จะนำมาแสดงจะเป็นจำนวนเราท์เตอร์ที่เชื่อมต่ออยู่กับเครือข่ายย่อยนี้ ในกรณีที่ผู้ใช้ต้องการข้อมูลที่ละเอียดมากยิ่งขึ้น

ผู้ใช้สามารถใช้บริการของระบบจากรายการ MIB Browser เพื่อหาข้อมูลเพิ่มเติมได้ในภายหลัง แอพพลิเคชั่นทั้งสองนี้จะเรียกใช้บริการจากเซอร์วิสของระบบชื่อ เน็ทเวิร์คเซอร์วิส โดย NetworkMap จะร้องขอข้อมูลที่ใช้สำหรับประกอบแผนที่เครือข่าย ส่วน ObjectInfo ก็จะใช้บริการการสืบค้นข้อมูลของอ๊อปเจ็กท์บนแผนที่เพื่อนำไปแสดงต่อผู้ใช้

1.6 Polling ให้บริการเกี่ยวกับงานด้านการโพลลิ่งทางสถิติ หน้าที่ของแอพพลิเคชั่น ได้แก่ การรับรายการที่ต้องการโพลลิ่งมาให้ระบบทำการตั้งเวลาและทำการโพลลิ่งตามเวลาที่ตั้งไว้ การแสดงผลของรายการที่มีการโพลลิ่งต่อผู้ใช้ แอพพลิเคชั่นจะใช้บริการจากเซอร์วิสของระบบชื่อ JobSchedule ซึ่งจะทำหน้าที่เก็บรายการและผลลัพธ์ของการโพลลิ่ง ในส่วนของการรับรายการนั้น ผู้ใช้สามารถใช้งานได้จากการเลือกรายการ Poll Request ในเมนูของผู้ใช้ และป้อนข้อมูลของรายการที่ต้องการโพลลิ่งแก่แอพพลิเคชั่นผ่านทาง PollReqForm.html ข้อมูลที่ต้องป้อนประกอบด้วยข้อมูลของอุปกรณ์ที่ต้องการโพลลิ่งได้แก่หมายเลขของอ๊อปเจ็กท์ไอเดนติฟายเออร์ พอร์ทของ SNMP เอเจ็นต์และเลขหมายไอพี และข้อมูลที่เป็นช่วงของระยะเวลาที่ใช้สำหรับการโพลลิ่งได้แก่ วันเวลาเริ่มและสิ้นสุดการโพลลิ่ง และช่วงของเวลาในการโพลลิ่งแต่ละครั้ง ส่วนการแสดงผลลัพธ์ของการโพลลิ่งจะปรากฏรายการของผู้ใช้เองและรายการที่เป็นสาธารณะที่ได้รับความจำนงจากผู้ใช้ที่เป็นเจ้าของรายการให้อนุญาตให้ผู้ใช้คนอื่นๆสามารถเรียกผลลัพธ์จากการโพลลิ่งขึ้นมาแสดงได้ รายการทั้งหมดที่ถูกเลือกจะถูกแสดงผลเป็นรูปของกราฟทางสถิติด้วยเอกสาร HTML ที่แอพพลิเคชั่นได้สร้างขึ้นและจะมีการอัพเดทข้อมูลอยู่ตลอดเวลา โดยในการอัพเดทขั้นตอนจะเหมือนการแสดงผลรายการที่ถูกเลือกในครั้งแรกต่างกันเพียงรูปกราฟของรายการจะถูกอัพเดทให้ใหม่ขึ้น

1.7 Messenger และ MesgBoard เป็นแอพพลิเคชั่นที่ทำหน้าที่ในการรับและแสดงข้อความที่ผู้บริหารระบบและผู้ใช้ใช้สำหรับติดต่อสื่อสารซึ่งกันและกัน Messenger มีหน้าที่ในการรับฝากข้อความเพื่อนำไปส่งให้กับเมสเสจเมเนเจอร์ซึ่งเป็นเซอร์วิสที่จัดการเกี่ยวกับการจัดเก็บข้อความ ผู้ใช้สามารถเรียกใช้ Messenger จากรายการ New Message ในเมนูของผู้ใช้ ในขณะที่ MesgBoard จะเป็นตัวคอยรับข้อความจากเมสเสจเมเนเจอร์มาแสดงผลบนบอร์ดข้อความซึ่งเป็นส่วนหนึ่งของหน้าจอของผู้ใช้โดยจะปรากฏอยู่ส่วนล่างของจอ ทุกครั้งที่ผู้ใช้แสดงตนและเข้าสู่ระบบได้อย่างถูกต้อง MesgBoard ก็จะได้รับกุญแจรหัสของผู้ใช้เพื่อนำไปใช้ขอบริการฟังก์ชัน MesgGetReq จากเมสเสจเมเนเจอร์เพื่อขอข้อความนำมาแสดงบนบอร์ด การทำงานของ MesgBoard จะคล้ายกับส่วนที่ทำการแสดงผลลัพธ์ของการโพลลิ่งของ Polling คือเอกสาร HTML ที่ประกอบด้วยข้อความที่ได้รับจากเมสเสจเมเนเจอร์จะมีการอัพเดทตัวเองอยู่ตลอดเวลา

2. เลเยอร์เซอร์วิส เป็นชั้นที่อยู่ถัดลงมาจากชั้นแอพพลิเคชั่น ทำงานในลักษณะของเซิร์ฟเวอร์โปรเซสให้บริการแก่โมดูลทั้งที่อยู่ในเลเยอร์แอพพลิเคชั่นและเลเยอร์เซอร์วิสด้วยกันเอง

โดยปกติเซอร์วิสโปรเซสเริ่มทำงานพร้อมกับระบบและจะสลายตัวไปเมื่อระบบยุติการทำงาน ซึ่งแตกต่างจากโมดูลในชั้นแอพพลิเคชั่นที่โปรเซสจะสลายตัวไปเมื่อการทำงานที่ได้รับมอบหมายเสร็จสิ้น เซอร์วิสทุกเซอร์วิสบนระบบยกเว้นคอนเน็กเตอร์จะมีฟังก์ชันมาตรฐานที่มีหน้าที่คอยรับและบันทึกทิกเก็ตคีย์และเซอร์วิสคีย์คือฟังก์ชัน GetServiceKey ซึ่งมีหน้าที่รับและบันทึกเซอร์วิสคีย์ของเซอร์วิสทั้งหมดบนระบบและฟังก์ชัน InsertTicket มีหน้าที่รับและบันทึกทิกเก็ตของแอพพลิเคชั่นที่ต้องการขอใช้บริการจากเซอร์วิส ข้อมูลของทั้ง 2 ฟังก์ชันจะถูกบันทึกลงตารางทิกเก็ตประจำเซอร์วิสรวมทั้งตาราง SystemTicket ซึ่งเป็นตารางที่คอนเน็กเตอร์ใช้เก็บทิกเก็ตทั้งหมดของระบบ สาเหตุที่คอนเน็กเตอร์ไม่มีฟังก์ชันในการจัดเก็บเซอร์วิสคีย์หรือทิกเก็ตคีย์เพราะการตรวจสอบสิทธิของผู้ใช้เกิดขึ้นในคอนเน็กเตอร์ ดังนั้นเมื่อมีรีเควสท์ที่ต้องการใช้บริการจากคอนเน็กเตอร์ คอนเน็กเตอร์จะสามารถบันทึกทิกเก็ตลงในตาราง ConnectorTicket ซึ่งเป็นตารางเก็บทิกเก็ตของคอนเน็กเตอร์ได้ทันที พิจารณาจากรูปตารางที่ 3.9 สำหรับเซอร์วิสคีย์ของเซอร์วิสบนระบบจะถูกคอนเน็กเตอร์เก็บไว้ในตาราง ServiceTable พิจารณาจากรูปตารางที่ 3.5 สำหรับตารางทิกเก็ตของเซอร์วิสอื่นๆได้แก่ ตาราง NetService, JobScTicket, GraphicTicket และ MesgManService พิจารณาได้จากรูปตารางที่ 3.12, 3.18, 3.21 และ 3.11 ตามลำดับ นอกจากนี้ยังมีฟังก์ชัน Dead เป็นคำสั่งที่สั่งให้เซอร์วิสยุติการทำงานและสลายตัวไปจากระบบ ฟังก์ชันนี้จะถูกคอนเน็กเตอร์เรียกใช้เมื่อมีการชัทดาวน์ระบบ เซอร์วิสของระบบมีดังนี้

2.1 คอนเน็กเตอร์ (Connector) เป็นตัวเชื่อมโยงแอพพลิเคชั่นและเซอร์วิสให้ทำงานร่วมกัน หน้าที่ของคอนเน็กเตอร์ได้แก่ การตรวจสอบสิทธิ (Authentication) ของผู้ใช้และแอพพลิเคชั่น การตรวจสอบตัวตนที่แท้จริงของเซอร์วิส การสตาร์ทอัพและชัทดาวน์เซอร์วิสของระบบ ตารางที่อยู่ภายใต้การดูแลของคอนเน็กเตอร์คือ ตาราง Users, UserOnSystem Filter, AppTable, ServiceTable, FunctionTable, MenuList, SystemTicket และ ConnectorTicket ดังตารางที่ 3.1 – 3.9 ต่อไปจะกล่าวถึงรายละเอียดของฟังก์ชันต่างๆของคอนเนคเตอร์ ดังนี้

2.1.1 SignOn เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบสิทธิของผู้ใช้เมื่อผู้ใช้ล็อกอินเข้าสู่ระบบ แอพพลิเคชั่นที่เรียกใช้ฟังก์ชัน SignOn คือ UserSignOn ซึ่งจะส่งชื่อ รหัสผ่าน และเลขหมายไอพีของผู้ใช้ มาให้ SignOn ทำการตรวจสอบโดยอาศัยข้อมูลจากตาราง Users ซึ่งเก็บข้อมูลเกี่ยวกับผู้ใช้ที่ลงทะเบียนกับระบบ พิจารณาจากรูปตารางที่ 3.1 และทำการตรวจสอบเลขหมายไอพีของผู้ใช้กับเลขหมายไอพีต้องห้ามในตาราง FilterIP ซึ่งเป็นตารางที่เก็บเลขหมายไอพีที่ไม่ต้องการให้ผู้ใช้ใช้ติดต่อกับระบบ พิจารณาจากรูปตารางที่ 3.3 หลังจากทำการตรวจสอบเสร็จสิ้น คอนเน็กเตอร์จะสร้างเซสชั่นคีย์ขึ้นใช้เป็นตัวแทนของการล็อกอินเข้าสู่ระบบของผู้ใช้ ข้อมูลที่การล็อกอินเข้าสู่ระบบเช่น เซสชั่นคีย์ วันเวลาที่ผู้ใช้มีปฏิสัมพันธ์กับระบบ ชื่อและเลขหมายไอพีของผู้ใช้จะถูกบันทึกลงในตาราง UserOnSystem ซึ่งเป็นตารางที่เก็บข้อมูลของผู้ใช้ที่ล็อกอินอยู่บนระบบ

พิจารณาจากตาราง 3.2 เซสชั่นคีย์ที่ถูกสร้างขึ้นจะถูกส่งกลับไปให้เวบเบราเซอร์โดยผ่านทางแอพพลิเคชั่น UserSignOn

2.1.2 SignOff เป็นฟังก์ชันที่ใช้เมื่อผู้ใช้ต้องการออกจากระบบ ฟังก์ชัน SignOff จะทำการลบข้อมูลของผู้ใช้ที่ทำการล็อกเอ๊าท์ในตาราง UserOnSystem ออก

ตารางที่ 3.1 ตารางข้อมูลของผู้ใช้บนระบบ: Users.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	Uname	ชื่อผู้ใช้	Char(11)	P.K.
2	Upassword	รหัสผ่าน	Char(11)	
3	Ulevel	ระดับสิทธิ	Integer	
4	Ustatus	สถานะภาพ	Integer	
5	Udetail	ข้อมูลทั่วไป	Char(30)	
6	UlastSignOnDate	วันที่ล่าสุดที่ผู้ใช้เข้าสู่ระบบ	Date	
7	UlastSignOnTime	เวลาล่าสุดที่ผู้ใช้เข้าสู่ระบบ	Time	

ตารางที่ 3.2 ตารางข้อมูลของผู้ใช้ล็อกอินอยู่บนระบบ: UserOnSystem.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	RandomNum	เซสชั่นคีย์	Char(40)	P.K.
2	Uname	ชื่อผู้ใช้	Char(11)	
3	LastActiveDate	วันที่ล่าสุดที่ผู้ใช้ปฏิสัมพันธ์กับระบบ	Date	
4	LastActiveTime	เวลาล่าสุดที่ผู้ใช้ปฏิสัมพันธ์กับระบบ	Time	
5	RemoteIP	เลขหมายไอพี	Char(15)	

ตารางที่ 3.3 ตารางเลขหมายไอพีที่ห้ามเข้าสู่ระบบ: FilterIP.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	IPAddr	เลขหมายไอพี	Char(15)	P.K.
2	Reason	เหตุผลของการห้าม	Char(255)	

2.1.3 Auth เป็นฟังก์ชันที่มีหน้าที่ตรวจสอบสิทธิของผู้ใช้จากเซสชั่นคีย์ที่ได้รับจากแอพพลิเคชั่น แอพพลิเคชั่นทุกแอพพลิเคชั่นที่ต้องการใช้บริการจากระบบจำเป็นที่จะต้องมีการตรวจสอบเซสชั่นคีย์เพื่อรับทิกเก็ตคีย์นำไปใช้ในการขอบริการจากเซอร์วิสอื่นๆของระบบ สิ่งที่

ฟังก์ชันนี้ทำการตรวจสอบคือความมีอยู่จริงของเซสชั่นคีย์ สิทธิของผู้ใช้ในการขอใช้บริการแอพพลิเคชั่น เลขหมายไอพีของผู้ใช้ และวันเวลาที่ผู้ใช้มีปฏิสัมพันธ์กับระบบเพื่อตรวจสอบไทม์เอาท์ ทุกครั้งที่การเรียกใช้ฟังก์ชัน Auth คอนเน็กเตอร์จะทำการอัพเดทวันเวลาที่ผู้ใช้มีปฏิสัมพันธ์กับระบบด้วย ยกเว้น MesgBoard ซึ่งเป็นแอพพลิเคชั่นที่นำข้อความจากเมสเสจแมเนเจอร์ไปแสดงบนบอร์ดข่าวสาร รายละเอียดของ MesgBoard ได้ถูกกล่าวถึงไปแล้วในส่วนของเลเยอร์แอพพลิเคชั่น

2.1.4 Service/application start up เป็นฟังก์ชันที่ไม่ได้ให้บริการต่อแอพพลิเคชั่นหรือเซอร์วิสใดๆ ต้องดำเนินการที่คอนโซลของคอนเน็กเตอร์ ฟังก์ชันนี้มีหน้าที่สตาร์ทอัพเซอร์วิสและแอพพลิเคชั่นเป็นขั้นตอนหนึ่งในการสตาร์ทอัพระบบ เพื่อตรวจสอบตัวตนของเซอร์วิส ความถูกต้องและสอดคล้องในการใช้บริการเซอร์วิสของแอพพลิเคชั่น ขั้นตอนการทำงานมีดังนี้

1. คอนเน็กเตอร์สร้างเซอร์วิสคีย์ประจำคอนเน็กเตอร์

2. คอนเน็กเตอร์ทำการตรวจสอบความเป็นตัวตนของเซอร์วิส โดยการคำนวณค่าไฟล์ซิกเนเจอร์จากโมดูลของเซอร์วิสที่อยู่บนระบบในปัจจุบันและเปรียบเทียบกับค่าไฟล์ซิกเนเจอร์จากตาราง ServiceTable (ในรูปตารางที่ 3.5) ที่ถูกคำนวณไว้เมื่อติดตั้งเซอร์วิสเข้าสู่ระบบ ต่อจากนั้นคอนเน็กเตอร์จะสร้างเซอร์วิสคีย์ให้กับเซอร์วิสและสั่งให้เซอร์วิสเริ่มทำงานพร้อมกับมอบเซอร์วิสคีย์ของคอนเน็กเตอร์และเซอร์วิสคีย์ของตัวเซอร์วิสเองที่เพิ่งถูกคอนเนคเตอร์สร้างขึ้น รวมทั้งเซิร์ฟเวอร์พอร์ทของเซอร์วิสเพื่อรับรีเควสท์จากโมดูลอื่นๆบนระบบ การสตาร์ทอัพเซอร์วิสจะเสร็จสิ้นเมื่อคอนเน็กเตอร์จะทำการตรวจสอบเซอร์วิสทั้งหมดบนระบบ

3. หลังจากการสตาร์ทอัพเซอร์วิสเสร็จสิ้นแล้ว ข้อมูลที่คอนเนคเตอร์มีคือเซอร์วิสคีย์ของเซอร์วิสทั้งหมดที่ผ่านการตรวจสอบและทำการส่งเซอร์วิสคีย์ทั้งหมดให้กับเซอร์วิสทุกเซอร์วิสที่ถูกสตาร์ทอัพด้วยการเรียกใช้ฟังก์ชัน GetServiceKey ของเซอร์วิส (รายละเอียดได้ถูกกล่าวถึงไปแล้วในตอนต้นของเลเยอร์เซอร์วิส) โดยอาศัยข้อมูลของเซอร์วิสจากตาราง ServiceTable ขั้นตอนนี้จะทำให้เซอร์วิสที่ถูกสตาร์ทอัพทราบเซอร์วิสคีย์ของเซอร์วิสอื่นๆและสามารถใช้เซอร์วิสคีย์ในการขอใช้บริการจากเซอร์วิสที่ต้องการได้ในภายหลัง

4. คอนเน็กเตอร์ทำการตรวจสอบความสอดคล้องในการขอใช้บริการจากเซอร์วิสของแอพพลิเคชั่นโดยใช้ข้อมูลของแอพพลิเคชั่นบนระบบจากตาราง AppTable (รูปตารางที่ 3.4) พิจารณาสถานะภาพของฟังก์ชัน เวอร์ชันและสถานะภาพของเซอร์วิสที่แอพพลิเคชั่นต้องการขอใช้บริการ โดยใช้ข้อมูลจากตาราง ServiceTable และ FunctionTable (รูปตารางที่ 3.5 และ 3.6) การตรวจสอบความสอดคล้องสามารถลดขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการใช้บริการจากแอพพลิเคชั่นที่มีการเรียกใช้บริการฟังก์ชันจากเซอร์วิสที่ไม่ได้ทำงานหรือไม่มีอยู่บนระบบในขณะนั้น โดยระบบจะทำการตรวจสอบได้ที่สถานะภาพของแอพพลิเคชั่นทันที

5. หลังจากการตรวจสอบความสอดคล้องในการขอใช้บริการจากเซอร์วิสของแอพพลิเคชั่นเสร็จสิ้น ระบบจะอยู่ในสถานะพร้อมให้บริการแก่ผู้ใช้

ตารางที่ 3.4 ตารางข้อมูลแอพพลิเคชั่น: AppTable.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	AppName	ชื่อแอพพลิเคชั่น	Char(30)	P.K.
2	Level	ระดับสิทธิของแอพพลิเคชั่น	Integer	
3	Status	สถานะภาพของแอพพลิเคชั่น	Char(5)	
4	ServiceInfo	ชื่อฟังก์ชันและเซอร์วิสที่ขอใช้บริการ	Char(255)	
5	Version	เวอร์ชันของแอพพลิเคชั่น	Char(10)	

ตารางที่ 3.5 ตารางข้อมูลเซอร์วิส: ServiceTable.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	SerName	ชื่อเซอร์วิส	Char(30)	P.K.
2	Version	เวอร์ชัน	Char(10)	
3	FileSignature	ค่าซิกเนเจอร์ของโมดูลของเซอร์วิส	Char(20)	
4	Port	เซิร์ฟเวอร์พอร์ทของเซอร์วิส	Integer	
5	KeyCode	คีย์โค๊ดประจำเซอร์วิส	Char(30)	
6	Lock	สวิทช์สำหรับล็อกเซอร์วิส	Boolean	
7	Status	สถานะภาพของเซอร์วิส	Char(5)	
6	Detail	ข้อมูลอื่นๆ	Char(30)	

ตารางที่ 3.6 ตารางข้อมูลฟังก์ชันของเซอร์วิส: FunctionTable.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	SerName	ชื่อเซอร์วิส	Char(30)	P.K.
2	FunctionName	ชื่อฟังก์ชัน	Char(30)	P.K.
3	Level	ระดับสิทธิของผู้ใช้	Integer	
4	Status	สถานะภาพของฟังก์ชัน	Char(5)	

ตารางที่ 3.7 ตารางข้อมูลของเมนูผู้ใช้: MenuList.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	ServiceName	ชื่อรายการบริการของแอพพลิเคชั่น	Char(30)	P.K.
2	FileName	ชื่อแอพพลิเคชั่นหรือไฟล์ HTML	Char(30)	
3	Status	สถานะภาพของรายการ	Char(5)	
4	Group	ชื่อกลุ่มของรายการ	Char(20)	

ตารางที่ 3.8 ตารางข้อมูลทิกเก็ตของระบบ: SystemTicket.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	SessionKey	เซสชั่นคีย์	Char(40)	P.K.
2	AppName	ชื่อแอพพลิเคชั่น	Char(30)	P.K.
3	Ticket	ทิกเก็ตของเซสชั่นคีย์	Char(30)	
4	Date	วันที่สร้างทิกเก็ตขึ้น	Date	
5	Time	เวลาที่สร้างทิกเก็ตขึ้น	Time	

ตารางที่ 3.9 ตารางข้อมูลทิกเก็ตของคอนเน็กเตอร์: ConnectorTicket.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	TicketCode	ทิกเก็ตของแอพพลิเคชั่น	Char(30)	P.K.
2	AppName	ชื่อแอพพลิเคชั่น	Char(30)	
3	Ticket	ทิกเก็ตของเซสชั่นคีย์	Char(30)	
4	Date	วันที่สร้างทิกเก็ตขึ้น	Date	
5	Time	เวลาล่าสุดที่เซสชันคีย์	Time	
6	CalledFunction	ฟังก์ชันของคอนเน็กเตอร์ที่ถูกเรียกใช้	Char(120)	

2.1.5 Application / Service Shutdown ฟังก์ชันเป็นแบบเดียวกับการสตาร์ทอัพเซอร์วิส และแอพพลิเคชั่นคือไม่เป็นฟังก์ชันที่ให้บริการแก่แอพพลิเคชั่นหรือเซอร์วิสใดๆ ผู้ดูแลระบบต้องดำเนินการที่คอนโซลของคอนเน็กเตอร์ การชัทดาวน์เซอร์วิสและแอพพลิเคชั่น เป็นขั้นตอนหนึ่งในกระบวนการชัทดาวน์ระบบ โดยคอนเน็กเตอร์มีหน้าที่ในการสั่งให้แต่ละเซอร์วิสที่กำลังทำงานบนระบบหยุดทำงานและสลายตัวไปด้วยการเรียกใช้ฟังก์ชัน Dead ที่มีอยู่ในเซอร์วิสทุกเซอร์วิสของระบบ (รายละเอียดได้กล่าวถึงไปแล้วในตอนต้นของเลเยอร์เซอร์วิส) แต่ก่อนที่จะทำการชัทดาวน์เซอร์วิส ผู้บริหารระบบต้องตรวจสอบให้แน่ใจเสียก่อนว่าไม่มีเซอร์วิสใดกำลังให้บริการค้างอยู่

2.1.6 Active/InActive เป็นฟังก์ชันแบบเดียวกับสตาร์อัพ ชัทดาวน์แอพพลิเคชั่นและเซอร์วิส ซึ่งมีหน้าที่กำหนดสถานะภาพของการให้บริการของเซอร์วิสบนระบบ สถานะ Active จะทำให้คอนเน็กเตอร์อนุญาตให้แอพพลิเคชั่นสามารถใช้บริการจากเซอร์วิสด้วยการสร้างทิกเก็ตให้กับแอพพลิเคชั่นที่ร้องขอสิทธิในการใช้บริการจากเซอร์วิสของระบบมายังคอนเน็กเตอร์ ในขณะที่สถานะ InActive จะเป็นลักษณะที่ตรงข้ามกับ Active โดยคอนเน็กเตอร์จะไม่อนุญาตให้แอพพลิเคชั่นใช้บริการจากเซอร์วิส โดยทั่วไปแล้วการสั่งให้คอนเน็กเตอร์เข้าสู่สถานะ InActive จะเกิดขึ้นเมื่อผู้ดูแลระบบต้องการชัทดาวน์ระบบ

2.1.7 MainMenu เป็นฟังก์ชันที่ให้บริการข้อมูลของรายการสำหรับเมนูของผู้ใช้บนเวบเบราเซอร์ ฟังก์ชันนำข้อมูลจากตาราง MenuList (รูปในตารางที่ 3.7)

2.2 เมสเสจเมเนเจอร์ (Message Manager) เป็นเซอร์วิสที่ทำหน้าที่เกี่ยวกับการรับส่งข้อมูลข่าวสารระหว่างผู้บริหารระบบกับผู้ใช้บนระบบ บริการที่สำคัญของเมสเสจเมเนเจอร์เซอร์วิส คือ บริการรับฝากข้อความ และบริการเรียกข้อความ เซอร์วิสจะทำงานเป็นตัวกลางในการรับฝากข้อความซึ่งถูกส่งมาจากแอพพลิเคชั่น Messenger โดยใช้บริการรับฝากข้อความของเซอร์วิส และแอพพลิเคชั่น MesgBoard จะเป็นตัวเรียกข้อความที่ถูกรับฝากอยู่บนเซอร์วิสเพื่อไปแสดงบนกระดานข่าวสารบนเวบเบราเซอร์ ข้อความที่เซอร์วิสรับฝากประกอบด้วยพารามิเตอร์เช่น วันเวลาที่รับฝาก ผู้ส่ง ผู้รับ และใจความ ตารางที่อยู่ภายใต้การดูแลของเมสเสจเมเนเจอร์คือตาราง Messages และ MesgManService ดังรูปตารางที่ 3.10 และ 3.11

ตารางที่ 3.10 ตารางข้อความ: Messages.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	Date	วันที่รับข้อความ	Date	P.K.
2	Time	เวลาที่รับข้อความ	Time	P.K.
3	Sender	ชื่อผู้ส่งข้อความ	Char(20)	
4	Recipiant	ชื่อผู้รับข้อความ	Char(20)	
5	Content	ข้อความ	Char(50)	

ตารางที่ 3.11 ตารางข้อมูลทิกเก็ตของ Message Manager: MesgManService.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	TicketCode	ทิกเก็ต	Char(30)	P.K.
2	OwnerName	ชื่อแอพพลิเคชั่นหรือเซอร์วิส	Char(30)	
3	Port	พอร์ทของเซอร์วิส	Integer	

ตารางที่ 3.11 (ต่อ)

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
4	Date	วันที่สร้างทิกเก็ตขึ้น	Date	
5	Time	เวลาที่สร้างทิกเก็ตขึ้น	Time	
6	Status	สถานะภาพของเซอร์วิส	Char(10)	
7	CalledFunction	ฟังก์ชันของเซอร์วิสที่ถูกเรียกใช้	Char(120)	
8	Type	ประเภทของทิกเก็ต	Char(5)	

2.2.1 MesgPostReq เป็นฟังก์ชันที่ให้บริการในการรับฝากข้อความ ฟังก์ชันจะทำงานโดยนำข้อความไปเก็บไว้ในตาราง Messages ข้อความจะถูกจัดเรียงตามวันเวลาที่ฟังก์ชันถูกเรียกใช้บริการ

2.2.2 MesgGetReq เป็นฟังกชันที่ให้บริการข้อความที่ถูกฝากถึงผู้ใช้ แอพพลิเคชั่นจะได้รับข้อความตามชื่อผู้ใช้ที่ระบุมากับรีเควสท์

2.3 เน็ทเวิร์คเซอร์วิส (Network Service) บริการของเน็ทเวิร์คเซอร์วิสส่วนใหญ่จะเกี่ยวข้องกับงานด้านเครือข่ายและข้อมูลที่ได้มาจากการใช้โปรโตคอล SNMP ในการปฏิบัติงาน บริการดังกล่าวได้แก่ การสืบค้นเครือข่าย การเรียกค้น การแก้ไขข้อมูลโดยใช้ เก็ทรีเควทส์ เก็ทเน็กซ์รีเควสท์ เซ็ทรีเควสท์ ของโปรโตคอล SNMP

2.3.1 SNMPDiscovery เป็นฟังก์ชันหนึ่งของเน็ตเวิร์กเซอร์วิส ซึ่งให้บริการการสืบค้นเครือข่ายแก่แอพพลิเคชั่น NetworkDiscovery และข้อมูลที่ได้จากการสืบค้นจะถูกนำไปสร้างเป็นแผนที่เครือข่ายซึ่งจะถูกใช้ในการทำงานของแอพพลิเคชั่น NetworkMap และ ObjectInfo โดยวิธีการสืบค้นเครือข่ายที่ถูกใช้ในฟังก์ชันนี้จะเป็นการสืบค้นเครือข่ายของบทความที่อยู่ในภาคผนวก ข. เรื่อง การศึกษาองค์ประกอบของเครือข่ายที่มีผลต่อการสืบค้นเครือข่าย (A Study of Influential Factors to A Network Discovery) ซึ่งจะกล่าวถึงรายละเอียดต่างๆ ตั้งแต่วิธีการทำงาน การวิเคราะห์ การทดลอง และสรุปผลเกี่ยวกับวิธีการสืบค้นดังกล่าวนี้ สำหรับในหัวข้อนี้จะขอกล่าวถึงวิธีการสืบค้นเครือข่ายพอสังเขปดังนี้

การสืบค้นเครือข่ายเป็นการค้นหาและรวบรวมข้อมูลของอุปกรณ์ต่างๆที่เชื่อมต่ออยู่บนระบบเครือข่าย วิธีข้อมูลส่วนใหญ่จะถูกนำไปใช้ในการสร้างแผนที่เครือข่าย (Network map) ซึ่งมีประโยชน์มากสำหรับงานจัดการเครือข่าย เพราะทำให้ผู้บริหารระบบเครือข่ายสามารถมองเห็นและเข้าใจความสัมพันธ์ระหว่างองค์ประกอบต่างๆ บนเครือข่ายได้ดียิ่งขึ้น การสืบค้นที่ใช้ในงานวิจัยนี้จะทำงานโดยการค้นหาและรวบรวมข้อมูลของเราท์เตอร์และเครือข่ายย่อยทั้งหมดที่มีอยู่ในระบบเพื่อนำมาใช้ในการสร้างแผนที่เครือข่ายซึ่งจะอยู่ในรูปของการเชื่อมต่อระหว่างเครือข่ายย่อย

กับเราท์เตอร์ ซึ่งโปรโตคอลที่ใช้ในการทำงานคือ SNMPv1 (Simple Network Management Protocol version 1) เพื่อเข้าถึงข้อมูลที่อยู่ใน MIB-II บนเราท์เตอร์ซึ่งกระจายอยู่บนระบบเครือข่าย และเนื่องจากแผนที่เครือข่ายที่ถูกสร้างจะอยู่ในรูปของการเชื่อมต่อระหว่างเครือข่ายย่อยกับเราท์เตอร์ ทำให้ข้อมูลของเราท์เตอร์ที่นำมาใช้จะต้องสามารถบอกให้ทราบได้ว่าเราท์เตอร์ได้เชื่อมต่ออยู่กับเครือข่ายย่อยใดบ้างและต้องทำให้การสืบค้นข้อมูลดังกล่าวสามารถขยายออกไปได้ครอบคลุมทุกเครือข่ายย่อยในระบบ พารามิเตอร์บน MIB-II ที่ถูกเลือกมาใช้ในการสืบค้นได้แก่ IP-interface เป็นข้อมูลที่เป็นเลขหมายไอพีของอินเตอร์เฟส และ net mask โดยเลขหมายไอพีของอินเตอร์เฟส จะถูกเก็บอยู่ในตัวแปร ipAdEntAddr [1] และ netmask จะถูกเก็บอยู่ในตัวแปร ipAdEntNetMask [1] ตัวแปรทั้ง 2 จะอยู่ภายใต้ตาราง ipAddrTable [1] ซึ่งเป็นตาราง IP อินเตอร์เฟสของเราท์เตอร์ เมื่อนำเลขหมายไอพีและ net mask มาคำนวณจะทำให้ได้เลขหมายไอพีของเครือข่ายย่อยที่เราท์เตอร์ได้เชื่อมต่ออยู่ นอกจากนี้ยังมี Next-hop ซึ่งเป็นข้อมูลของเลขหมายเลขไอพีที่เป็น next hop และถูกเก็บอยู่ในตัวแปร ipRouteNextHop [1] ภายใต้ตาราง ipRouteTable [1] ซึ่งเป็นตารางเราท์ติ้งของเราท์เตอร์ เลขหมายไอพีเหล่านี้ก็คืออินเตอร์เฟสของเราท์เตอร์ตัวอื่นๆที่อยู่ในระบบเครือข่ายนั่นเอง ทำให้สามารถใช้เลขหมายไอพีเหล่านี้ขยายการสืบค้นไปทั่วทั้งระบบเครือข่ายได้

การทำงานจะเริ่มจากสีดเราท์เตอร์ (Seed router) คือเราท์เตอร์ที่ถูกกำหนดมาให้เป็นจุดเริ่มต้นของการทำงาน ขั้นตอนการทำงานมีดังต่อไปนี้

1. กำหนดสีดเราท์เตอร์ให้กับเมเนเจอร์เพื่อเป็นจุดเริ่มต้นของการสืบค้น

2. จากนั้นเมเนเจอร์จะทำการร้องขอข้อมูลของ IP-interface จากเราท์เตอร์ เพื่อนำข้อมูลเหล่านี้มาคำนวณหาเครือข่ายย่อยที่กำลังเชื่อมต่ออยู่กับเราท์เตอร์จากที่ได้กล่าวมาแล้วในเบื้องต้น ดังนั้นเมื่อเสร็จสิ้นกระบวนการในขั้นตอนนี้เมเนเจอร์จะทราบว่าเราท์เตอร์ประกอบไปด้วยอินเตอร์เฟสที่มีเลขหมายไอพี อะไรบ้างและกำลังเชื่อมต่ออยู่เครือข่ายย่อยใด

3. ถัดมาเมเนเจอร์จะทำการร้องขอข้อมูลจากเราท์เตอร์ตัวเดิม โดยในคราวนี้ข้อมูลที่ต้องการคือ Next-hop ซึ่งจะทำให้เมเนเจอร์ทราบเพิ่มเติมว่ามีเราท์เตอร์ใดอยู่ในระบบอีกบ้าง จากนั้นจะทำการเก็บเลขหมายไอพีของเราท์เตอร์เหล่านี้ไว้สำหรับร้องขอข้อมูลในส่วนของ IP-interface ในโอกาสต่อไป เมื่อมาถึงจุดนี้เท่ากับว่าเมเนเจอร์ได้รับข้อมูลทั้งหมดที่ต้องการจากเราท์เตอร์ที่ได้รับมาในเบื้องต้นครบแล้ว

4. เมื่อพิจารณาให้ดีจะพบว่ากระบวนการในข้อ 3 คือส่วนสำคัญที่ทำให้เกิดการสืบค้นข้อมูลที่ขยายออกไปเป็นทอดๆ จากเราท์เตอร์หนึ่งไปยังเราท์เตอร์อีกตัวหนึ่ง จึงจำเป็นอย่างยิ่งที่ต้องมีขั้นตอนการตรวจสอบความซ้ำซ้อน โดยเราท์เตอร์ที่เคยถูกเมเนเจอร์ร้องขอข้อมูลไปแล้ว (สมมุติว่าการร้องขอข้อมูลในแต่ละครั้งประสบความสำเร็จคือไม่มีความผิดพลาดที่ก่อให้เกิดเหตุการณ์ที่เมเนเจอร์ต้องเก็บเราท์เตอร์ตัวนี้ไว้เพื่อร้องขอข้อมูลในโอกาสถัดไป) จะต้องไม่ถูกนำกลับมาใช้ในการร้องขอข้อมูลชนิดนี้อีก ทั้งนี้เพื่อป้องกันมิให้การวนซ้ำที่ไม่มีจุดสิ้นสุดเกิดขึ้น ดังนั้นในขั้นตอนนี้

จึงต้องนำเลขหมายไอพีของ Next-hop ที่ได้มาตรวจสอบความซ้ำซ้อนกับเลขหมายไอพีทั้งหมดของ IP-interface ที่เมเนเจอร์มีอยู่ แล้วคัดเอาเฉพาะข้อมูลของเลขหมายไอพีของ Next-hop ที่ไม่ซ้ำซ้อน เก็บไว้ อย่างไรก็ตามขั้นตอนการตรวจสอบก็ยังไม่สิ้นสุดที่จุดนี้ ยังต้องมีการตรวจสอบความซ้ำซ้อนภายในระหว่างเลขหมายไอพีของ Next-hop ด้วยกันอีกครั้งหนึ่ง สาเหตุที่ต้องทำเช่นนี้เนื่องจากโดยปกติแล้วเราท์เตอร์จะใช้เลขหมายไอพีหนึ่งๆใน Next-hop เป็นช่องทางออกไปยังเครือข่ายปลายทาง (Destination network) ได้มากกว่า 1 เครือข่าย ทำให้ข้อมูลของ Next-hop มักจะมีหมายเลขซ้ำกันเกิดขึ้น ดังนั้นจึงต้องมีการตัดส่วนของเลขหมายไอพีที่ซ้ำออกไป

5. เลือกเราท์เตอร์จากเลขหมายไอพีของ Next-hop ที่เมเนเจอร์มีอยู่ มาใช้ในการทำงานต่อ โดยทำซ้ำในข้อ 2.-4. จนกระทั่งเราท์เตอร์ที่มีอยู่ถูกนำมาใช้ในการทำงานจนหมดจึงจบการทำงาน

หลังจากเสร็จสิ้นกระบวนการสืบค้นเครือข่ายข้อมูลที่ได้จะอยู่ในรูปของความสัมพันธ์ระหว่างเราท์เตอร์กับเลขหมายไอพีของเครือข่ายย่อย ซึ่งสามารถนำมาสร้างเป็นแผนที่เครือข่ายที่แสดงความสัมพันธ์ระหว่างเราท์เตอร์กับเครือข่ายย่อยได้ โดยแผนที่เครือข่ายที่ถูกสร้างขึ้นจะเป็นภาพบิทแมพที่มีการเก็บข้อมูลของตำแหน่งพิกัดบนภาพของเราท์เตอร์และเครือข่ายย่อย แผนที่เครือข่ายและข้อมูลดังกล่าวนี้จะถูกแอพพลิเคชั่นบางแอพพลิเคชั่นนำไปประมวลผลเพื่อให้บริการแก่ผู้ใช้ต่อไป รายละเอียดของแอพพลิเคชั่นที่เกี่ยวข้องได้กล่าวถึงไปแล้วในหัวข้อชั้นแอพพลิเคชั่น

ตารางที่ 3.12 ตารางข้อมูลทิกเก็ตของเน็ทเซอร์วิส: NetService.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	TicketCode	ทิกเก็ต	Char(30)	P.K.
2	OwnerName	ชื่อแอพพลิเคชั่นหรือเซอร์วิส	Char(30)	
3	Port	พอร์ทของเซอร์วิส	Integer	
4	Date	วันที่สร้างทิกเก็ตขึ้น	Date	
5	Time	เวลาที่สร้างทิกเก็ตขึ้น	Time	
6	Status	สถานะภาพของเซอร์วิส	Char(10)	
7	CalledFunction	ฟังก์ชันของเซอร์วิสที่ถูกเรียกใช้	Char(120)	
8	Type	ประเภทของทิกเก็ต	Char(5)	

ตารางที่ 3.13 ตารางข้อมูลของเราท์เตอร์: NodeTab.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	NodeName	ชื่อเราท์เตอร์	Char(30)	P.K.
2	SysDesc	System Description	Char(255)	

ตารางที่ 3.13 (ต่อ)

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
3	SysObjID	System Object ID.	Char(255)	
4	SysContact	System Contact	Char(255)	
5	SysLocation	System Location	Char(255)	
6	SysUpTime	System Uptime	Real	
7	SysService	System Service	Integer	
8	IfNumber	Interface number	Integer	
9	Checked	Flag	Integer	

ตารางที่ 3.14 ตารางข้อมูลอินเตอร์เฟสของเราท์เตอร์: IfTableTab.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	NodeName	ชื่อเราท์เตอร์	Char(30)	P.K.
2	IfIndex	Interface Index	Integer	P.K.
3	IfDesc	Interface Description	Char(255)	
4	IfType	Interface Type	Integer	
5	IfMTU	Interface Maximum Transfer Unit	Integer	
6	IfPhyAddress	Interface Physical Address	Char(17)	
7	IfOperStatus	Interface Operation Status	Integer	
8	IfAdminStatus	Interface Admin Status	Integer	
9	IfSpeed	Interface Speed	Real	
10	IfLastChange	Interface Last Change	Real	
11	IfSpecific	Interface Specific	Char(255)	

ตารางที่ 3.15 ตารางความสัมพันธ์ของเราท์เตอร์และเครือข่ายย่อย: RouterSubnetTab.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	IPAddress	IP Address	Char(15)	P.K.
2	NodeName	ชื่อเราท์เตอร์	Char(30)	
3	IfIndex	Interface Index	Integer	
4	NetMask	SubNetMask	Char(15)	
5	Subnet	Sub Network IP Address	Char(15)	

ตารางที่ 3.16 ตารางข้อมูลของแผนที่เครือข่าย: MapGeo.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	ObjectName	ชื่ออ๊อปเจ็กต์	Char(30)	P.K.
2	ObjectLabel	ป้ายชื่อของอ๊อปเจ็กต์	Char(30)	
3	ObjectType	ชนิดของอ๊อปเจ็กต์	Integer	
4	Status	สถานะภาพของอ๊อปเจ็กต์	Integer	
5	Flag	Flag	Integer	
6	Left	พิกัดของอ๊อปเจ็กต์	Integer	
7	Top	พิกัดของอ๊อปเจ็กต์	Integer	
8	Right	พิกัดของอ๊อปเจ็กต์	Integer	
9	Bottom	พิกัดของอ๊อปเจ็กต์	Integer	

2.3.2 GetRequset GetNextRequest และ SetRequest การเรียกค้นและแก้ไขข้อมูลด้วยโปรโตคอล SNMP เป็นบริการสำหรับแอพพลิเคชั่นที่ต้องการเรียกค้นข้อมูลทั้งที่เป็นแบบข้อมูลเดี่ยวหรือข้อมูลแบบเป็นชุดหรือต้องการแก้ไขข้อมูลที่อยู่บน MIB ของอุปกรณ์บนระบบเครือข่าย อุปกรณ์ที่สามารถถูกเรียกค้นหรือแก้ไขข้อมูลได้นั้นต้องมีเอเจนท์ของโปรโตคอล SNMP ทำงานอยู่เพื่อเป็นตัวกลางระหว่างเน็ทเวิร์กเซอร์วิสกับ MIB ของอุปกรณ์ การเรียกค้นข้อมูลที่เป็นข้อมูลเดี่ยวเน็ทเวิร์กเซอร์วิสจะใช้คำสั่งเก็ทรีเควทส์ ข้อมูลแบบเป็นชุดใช้เก็ทเน็กซ์รีเควทส์ และเมื่อต้องการแก้ไขข้อมูลใช้เซ็ทรีเควทส์ของโปรโตคอล SNMP พารามิเตอร์ที่ต้องใช้ในการทำงานคือหมายเลขอ๊อบเจ็กท์ ชื่อของคอมมิวนิตี้ เลขหมายไอพีและพอร์ทของเอเจนท์ของ SNMP ของอุปกรณ์ที่ต้องการปฏิบัติการ

2.3.3 MapObject บริการข้อมูลพิกัดของแผนที่เครือข่าย มีหน้าที่ให้บริการข้อมูลพิกัดของเราท์เตอร์และเครือข่ายย่อยบนภาพแผนที่เครือข่ายแก่แอพพลิเคชั่นที่ต้องการนำข้อมูลไปสร้างเป็นอินเตอร์เฟสให้กับผู้ใช้ได้ใช้งานบนเวบเบราเซอร์ โดยอาศัยข้อมูลจากตาราง MapGeo (ในรูปตารางที่ 3.16)

2.3.4 NodeInfo และ SubnetInfo บริการข้อมูลของอ๊อปเจ็กท์ที่บนแผนที่เครือข่าย โดยอาศัยข้อมูลจากตาราง NodeTab, IfTableTab และ RouterSubnetTab (ในตารางที่ 3.13 3.14 และ 3.15) ให้บริการข้อมูลของเราท์เตอร์และเครือข่ายย่อยที่ปรากฏบนภาพแผนที่เครือข่าย ข้อมูลจะแบ่งเป็น2 แบบคือ ข้อมูลของเราท์เตอร์ และข้อมูลของเครือข่ายย่อย เซอร์วิสจะให้ข้อมูลเกี่ยวกับอินเตอร์เฟสของเราท์เตอร์ และข้อมูลของเครือข่ายย่อย ซึ่งเซอร์วิสจะให้ข้อมูลเป็นเลขหมายไอพีที่อยู่ภายในเครือข่ายย่อยนั้น

2.4 Job Schedule การทำงานแบบตารางเวลางานเป็นระบบการทำงานตามระยะเวลาที่ถูกกำหนดไว้ล่วงหน้า สำหรับระบบในงานวิจัยนี้จะประกอบด้วย 2 ส่วนคือส่วนจัดการตารางเวลาเป็นส่วนที่เก็บรายการที่ต้องการให้มีการทำตารางเวลางานโดยจะเก็บไว้ในตาราง JobSc (ดังตารางที่ 3.17) และส่วนปฏิบัติการเป็นส่วนที่มีหน้าที่พิจารณากำหนดเวลาและทำงานตามรายการที่กำหนดไว้ โดยส่วนปฏิบัติการจะเป็นส่วนที่คอยร้องขอข้อมูลจากส่วนที่เก็บรายการเพื่อนำไปพิจารณาว่าถึงเวลาที่จะทำงานตามรายการใดบ้าง ดังนั้นส่วนปฏิบัติการจะต้องมีความสามารถที่จะทำงานหรือรู้ว่าสามารถเรียกใช้บริการของเซอร์วิสใดเพื่อที่จะทำงานตามรายการที่กำหนดไว้ได้ จากหน้าที่ดังกล่าวตารางเวลางานระบบหนึ่งจะสามารถมีส่วนที่เป็นส่วนปฏิบัติการได้หลายส่วนตามหน้าที่ดังเช่นในงานวิจัยนี้มีเซอร์วิสที่มีหน้าที่เป็นตารางเวลาคือ JobSchedule และมีส่วนปฏิบัติการคือ StatPolling ซึ่งจะใช้โปรโตคอล SNMP ในการเรียกค้นข้อมูลจากอุปกรณ์เครือข่ายเพื่อนำมาเก็บเป็นข้อมูลทางด้านสถิติ รายละเอียดของ StatPolling จะขอกล่าวอีกครั้งในหัวข้อถัดไป ต่อไปจะกล่าวถึงบริการของ JobSchedule ซึ่งเป็นบริการเกี่ยวกับข้อมูลของรายการซึ่งมีการดำเนินการแบบตารางเวลางาน ดังต่อไปนี้

ตารางที่ 3.17 ตารางข้อมูลรายการตารางเวลา: JobSc.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	LinkTable	ชื่อตารางที่เก็บผลลัพธ์ของรายการ	Char(30)	P.K.
2	Input	อินพุตของรายการ	Char(255)	
3	UserName	ชื่อเจ้าของรายการ	Char(11)	
4	IntTime	Interval Time	Integer	
5	StartDate	วันที่เริ่มดำเนินการ	Date	
6	StartTime	เวลาที่เริ่มดำเนินการ	Time	
7	StopDate	วันที่หยุดดำเนินการ	Date	
8	StopTime	เวลาที่หยุดดำเนินการ	Time	
9	LastDate	วันที่ล่าสุดที่ดำเนินการ	Date	
10	LastTime	เวลาล่าสุดที่ดำเนินการ	Time	
11	PublicFlag	บอกความต้องการเปิดเผยข้อมูล	Char(3)	
12	Status	สถานะภาพของรายการ	Char(3)	
13	Type	ประเภทของรายการ	Char(10)	

ตารางที่ 3.18 ตารางข้อมูลทิกเก็ตของ Job Schedule: JobScTicket.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	TicketCode	ทิกเก็ต	Char(30)	P.K.
2	OwnerName	ชื่อแอพพลิเคชั่นหรือเซอร์วิส	Char(30)	
3	Port	พอร์ทของเซอร์วิส	Integer	
4	Date	วันที่สร้างทิกเก็ตขึ้น	Date	
5	Time	เวลาที่สร้างทิกเก็ตขึ้น	Time	
6	Status	สถานะภาพของเซอร์วิส	Char(10)	
7	CalledFunction	ฟังก์ชันของเซอร์วิสที่ถูกเรียกใช้	Char(120)	
8	Type	ประเภทของทิกเก็ต	Char(5)	

ตารางที่ 3.19 ตารางต้นแบบของตารางผลลัพธ์ของรายการ: LinkTable.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	Date	วันที่	Date	P.K.
2	Time	เวลา	Time	P.K.
3	Output	ผลลัพธ์	Char(255)	

2.4.1 StoreJob บริการเก็บบันทึกรายการที่ต้องการให้มีการดำเนินการแบบตารางเวลางาน รายการที่ถูกเก็บบันทึกจะประกอบด้วยพารามิเตอร์ดังนี้ วันเวลาที่เริ่ม หยุดและเวลาล่าสุดที่ทำงาน ระยะห่างของเวลาในการทำงานแต่ละครั้ง อินพุตของงาน ชื่อผู้ใช้ และประเภทของงาน โดยอินพุตของงานจะขึ้นอยู่กับประเภทของงาน เช่น ประเภทของงานเป็นการเก็บข้อมูลทางสถิติโดยการใช้โปรโตคอล SNMP อินพุตของงานจะต้องประกอบด้วยพารามิเตอร์ที่ใช้ในโปรโตคอล SNMP เป็นต้น รายการที่ฟังก์ชัน StoreJob ทำการบันทึกจะถูกเก็บลงในตาราง JobSc และในการบันทึกแต่ละครั้งเซอร์วิส JobSchedule จะสร้างตารางที่เก็บเอ๊าพุตของรายการโดยการสำเนาจากตาราง LinkTable (ดังรูปตารางที่ 3.19) ซึ่งเป็นตารางต้นแบบ ตารางเอ๊าพุตที่ได้จากการสำเนานี้จะถูกตั้งชื่อและใช้เป็นชื่อฟิลด์ LinkTable ซึ่งเป็นชื่อรายการตารางเวลาในตาราง JobSc ด้วย

2.4.2 JobForPoll บริการข้อมูลของรายการในตารางเวลาที่ยังไม่สิ้นสุดการดำเนินงาน ข้อมูลที่ให้บริการสามารถแบ่งออกโดยตามประเภทของงาน โดยสามารถเรียกค้นรายการเฉพาะประเภทได้โดยอาศัยข้อมูลของรายการจากตาราง JobSc

2.4.3 AllJobData บริการข้อมูลของรายการที่มีอยู่ในตารางเวลาทั้งหมดทั้งที่สิ้นสุดและยังไม่สิ้นสุดการดำเนินงานโดยอาศัยข้อมูลจากตาราง JobSc

2.4.4 UpdateJobStatus เป็นบริการอัพเดทข้อมูลที่เป็นพารามิเตอร์ของรายการหลังจากที่รายการถูกนำดำเนินการเสร็จสิ้น พารามิเตอร์ที่มีการอัพเดทคือวันเวลาที่ดำเนินการล่าสุด ซึ่งจะต้องนำไปใช้ในการคำนวณระยะเวลาที่จะต้องมีการดำเนินการใหม่อีกครั้งในตาราง JobSc

2.4.5 UpdateLinkTable นอกจาก JobSchedule จะดูแลข้อมูลของรายการแล้ว ยังดูแลผลลัพธ์ที่ได้จากการทำงานด้วย UpdateLinkTable จะรับข้อมูลจากตัวปฏิบัติการมาเก็บบันทึกไว้ในตารางเอ๊าพุตของเซอร์วิส JobSchedule

2.4.6 LinkTableData บริการข้อมูลที่ได้จากตารางเอ๊าท์พุตซึ่งได้จากการดำเนินการตามตารางเวลา ข้อมูลเหล่านี้จะถูกนำไปแสดงแก่ผู้ใช้ตามลักษณะของข้อมูล เช่น ข้อมูลทางด้านสถิติ ก็จะถูกนำไปสร้างเป็นแผนภูมิทางสถิติ เพื่อนำเสนอต่อผู้ใช้ในโอกาสถัดไป เป็นต้น

2.5 StatPolling มีหน้าที่โพลข้อมูลทางสถิติของอุปกรณ์เครือข่าย เช่นปริมาณข้อมูลขาเข้า ขาออก ของอินเตอร์เฟสหนึ่งๆในเราท์เตอร์ StatPolling ทำงานร่วมกับ GraphPainter ซึ่งมีหน้าที่สร้างแผนภูมิทางสถิติโดยใช้ข้อมูลที่ได้จากการโพล StatPolling จะทำงานโดยใช้บริการของ JobSchedule และ NetService ซึ่งมีการเรียกใช้บริการตามลำดับขั้นตอนของการทำงานดังนี้ คือขั้นแรกจะเรียกใช้ JobForPoll เพื่อขอข้อมูลรายการที่ต้องการนำมาโพล โดยจะทำการคำนวณหาเวลาของการโพลด้วยค่าระยะห่างของเวลาในการทำงานแต่ละครั้ง กับวันเวลาที่มีการโพลครั้งล่าสุดถัดมาจะนำรายการที่ถึงกำหนดเวลามาทำการโพลโดยเรียกใช้บริการ GetRequest ของ NetService จากนั้นจึงทำการอัพเดตวันเวลาที่มีการโพลล่าสุด และบันทึกข้อมูลที่ได้จากการโพล โดยเรียกใช้บริการ UpdateJobStatus และ UpdateLinkTable ของ JobSchedule ตามลำดับ

ตารางที่ 3.20 ตารางข้อมูลทิกเก็ตของ Stat Polling: StatPollService.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	TicketCode	ทิกเก็ต	Char(30)	P.K.
2	OwnerName	ชื่อแอพพลิเคชั่นหรือเซอร์วิส	Char(30)	
3	Port	พอร์ทของเซอร์วิส	Integer	
4	Date	วันที่สร้างทิกเก็ตขึ้น	Date	
5	Time	เวลาที่สร้างทิกเก็ตขึ้น	Time	
6	Status	สถานะภาพของเซอร์วิส	Char(10)	
7	CalledFunction	ฟังก์ชันของเซอร์วิสที่ถูกเรียกใช้	Char(255)	
8	Type	ประเภทของทิกเก็ต	Char(5)	

2.6 GraphPainter มีหน้าที่นำข้อมูลที่ได้จากการโพลมาสร้างเป็นแผนภูมิทางสถิติ โดย GraphPainter ในงานวิจัยนี้จะสนับสนุนการสร้างเป็นรูปของกราฟเส้น ข้อมูลที่นำมาสร้างเป็นกราฟจะได้มาจากการขอใช้บริการ AllJobData และ LinkTableData ของ JobSchedule

ตารางที่ 3.21 ตารางข้อมูลทิกเก็ตของ Graph Painter: GraphicTicket.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	TicketCode	ทิกเก็ต	Char(30)	P.K.
2	OwnerName	ชื่อแอพพลิเคชั่นหรือเซอร์วิส	Char(255)	
3	Port	พอร์ทของเซอร์วิส	Integer	
4	Date	วันที่สร้างทิกเก็ตขึ้น	Date	
5	Time	เวลาที่สร้างทิกเก็ตขึ้น	Time	
6	Status	สถานะภาพของเซอร์วิส	Char(10)	
7	CalledFunction	ฟังก์ชันของเซอร์วิสที่ถูกเรียกใช้	Char(120)	
8	Type	ประเภทของทิกเก็ต	Char(5)	

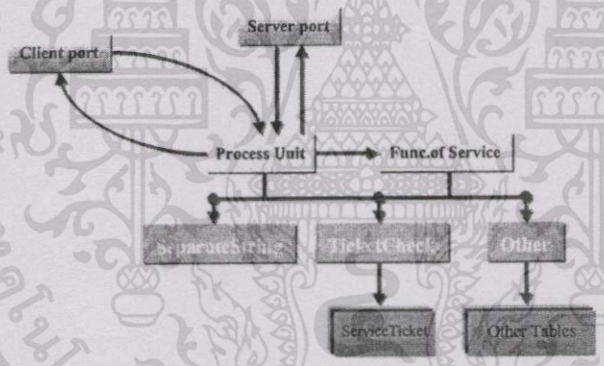
ตารางที่ 3.22 ตารางรายการที่ถูกดำเนินการไปแล้วของ Graph Painter: GraphicOutput.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	LinkTable	ชื่อตารางผลลัพธ์ของรายการ	Char(30)	P.K.
2	LastDate	วันที่ล่าสุดที่ดำเนินการ	Date	
3	LastTime	เวลาล่าสุดที่ดำเนินการ	Time	

3. เลเยอร์ไลบรารี่ เป็นเลเยอร์ที่อยู่ล่างสุดของระบบ บทบาทของโมดูลในเลเยอร์นี้จะเป็นตัวกลางในการติดต่อระหว่างเซอร์วิสกับระบบปฏิบัติการ โมดูลจะมีลักษณะเป็นชุดคำสั่งซึ่งอาจจะเป็นโมดูลที่ได้จากบริษัทต่างๆหรือมีมากับระบบปฏิบัติการอยู่ก่อนแล้ว สำหรับโมดูลของเลเยอร์ไลบารี่ในงานวิจัยชิ้นนี้จะเป็นโมดูลที่จัดการโปรโตคอล SNMP ของบริษัทดาทท์จำกัด BDE (Borland Database Engine) ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างเซอร์วิสโมดูลกับตารางข้อมูลแบบพาราด็อกซ์ และ WinSock เป็นชุดคำสั่งซ็อกเก็ตของโปรโตคอล TCP/IP ที่ถูกใช้ในการติดต่อสื่อสารระหว่างแอพพลิเคชั่นและเซอร์วิส

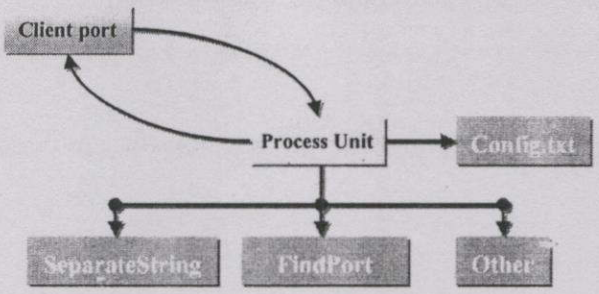
3.5 โครงสร้างของเซอร์วิสและแอพพลิเคชั่น

โครงสร้างของเซอร์วิสดังรูปที่ 3.6 ประกอบด้วยส่วนประกอบ 3 ส่วนคือ ส่วนที่ใช้ติดต่อกับโลกภายนอกได้แก่ ไคลเอ็นต์พอร์ท และเซิร์ฟเวอร์พอร์ท โดยไคลเอ็นต์พอร์ทจะใช้เป็นส่วนที่ให้เซอร์วิสใช้ติดต่อเพื่อขอใช้บริการเซอร์วิสด้วยกันเอง ในขณะที่เซิร์ฟเวอร์พอร์ทใช้สำหรับรอรับข้อความจากแอพพลิเคชั่นหรือเซอร์วิสอื่นๆที่ต้องการใช้บริการจากเซอร์วิส ส่วนถัดมาคือส่วนของฟังก์ชันและตารางฐานข้อมูล ซึ่งถูกฟังก์ชันใช้ในเก็บข้อมูลเพื่อใช้ในการทำงานต่างของฟังก์ชัน ฟังก์ชันจะแบ่งเป็น 2 แบบคือฟังก์ชันที่มีไว้ให้บริการแก่เซอร์วิสหรือแอพพลิเคชั่น และฟังก์ชันที่ใช้สำหรับกิจกรรมภายในของเซอร์วิส โดยฟังก์ชันหลักได้แก่ SeparateString และ TicketCheck ซึ่งทำงานกับข้อมูลที่เซอร์วิสได้รับมาจากทั้งไคลเอ็นต์พอร์ทและเซิร์ฟเวอร์พอร์ท โดย SeparateString ใช้สำหรับแยกแยะข้อมูลต่างๆที่อยู่ภายในข้อความ TicketCheck นำทิกเก็ตที่ได้จาก SeparateString มาตรวจสอบความถูกต้องกับทิกเก็ตที่มีอยู่ตารางทิกเก็ตของเซอร์วิส ในส่วนของรูปแบบของข้อความจะกล่าวถึงอีกครั้งในหัวข้อถัดไป



รูปที่ 3.6 โครงสร้างภายในของเซอร์วิส

สำหรับโครงสร้างของแอพพลิเคชั่นดังรูปที่ 3.7 จะมีลักษณะที่คล้ายกับเซอร์วิสแต่แอพพลิเคชั่นจะไม่มีเซิร์ฟเวอร์พอร์ท ทั้งนี้เนื่องจากแอพพลิเคชั่นไม่ได้ประพฤติตัวเป็นเซิร์ฟเวอร์ดังนั้นจึงไม่จำเป็นที่จะต้องมีเซิร์ฟเวอร์พอร์ท และสำหรับฟังก์ชันหลัก แอพพลิเคชั่นจะมีฟังก์ชัน FindPort ในการแยกแยะหมายเลขพอร์ทต่างที่ได้รับมาจากคอนเน็กเตอร์ซึ่งจะมาพร้อมกับทิกเก็ตเมื่อแอพพลิเคชั่นได้รับอนุญาตสำหรับใช้บริการจากเซอร์วิสของระบบ ในส่วนของ Config.txt ซึ่งจะเก็บเลขหมายไอพีของระบบเพื่อให้แอพพลิเคชั่นใช้ในการสร้างไดนามิก HTML ซึ่งเป็นหน้าที่หนึ่งของแอพพลิเคชั่น และเก็บหมายเลขพอร์ทของคอนเน็กเตอร์เพื่อให้แอพพลิเคชั่นใช้ในการติดต่อกับคอนเน็กเตอร์เมื่อแอพพลิเคชั่นเริ่มทำงาน



รูปที่ 3.7 โครงสร้างภายในของแอพพลิเคชั่น

3.6 รูปแบบของข้อความในระบบ

ข้อความที่ถูกรับและส่งระหว่างโมดูลต่างๆของระบบตั้งแต่ข้อความที่เริ่มจาก เวบเบราเซอร์ จนถึงอุปกรณ์เครือข่ายจะมีรูปแบบที่แตกต่างกันตามโปรโตคอลที่ใช้ โดยในส่วนของการสื่อสารด้วยโปรโตคอล HTTP ระหว่างเวบเบราเซอร์และเวบเซิร์ฟเวอร์จะเป็นไปตามข้อกำหนดของโปรโตคอล HTTP เช่นเดียวกับการทำงานของเวบเซิร์ฟเวอร์กับแอพพลิเคชั่นที่ใช้เทคโนโลยี ISAPI และการติดต่อกับอุปกรณ์เครือข่ายด้วยโปรโตคอล SNMP ซึ่งรูปแบบของข้อมูลก็จะเป็นไปตามข้อกำหนดของโปรโตคอล แต่สำหรับการติดต่อสื่อสารที่เกิดขึ้นระหว่างแอพพลิเคชั่นกับเซอร์วิส หรือเซอร์วิสกับเซอร์วิสซึ่งใช้ซ็อกเก็ตของโปรโตคอล TCP/IP จะแตกต่างออกไป เนื่องจากระบบจะใช้ส่วนเปย์โหลด (Pay Load) ของ TCP/IP เก็บพารามิเตอร์ต่างๆ ที่จะต้องใช้ในการทำงานของแอพพลิเคชั่นและเซอร์วิส ดังนั้นเพื่อที่จะให้แอพพลิเคชั่นและเซอร์วิสสามารถเข้าใจและแยกแยะพารามิเตอร์ต่างๆได้ชัดเจนถูกต้อง จึงต้องมีรูปแบบของข้อมูลเหล่านั้น โดยรูปแบบจะมี 2 แบบคือ ข้อความที่เซอร์วิสได้รับ และ ข้อความที่แอพพลิเคชั่นได้รับ

สำหรับข้อความที่เซอร์วิสได้รับส่วนใหญ่จะเป็นข้อความสำหรับร้องขอใช้บริการจากเซอร์วิส รูปแบบของข้อความจะประกอบด้วย อักขระส่วนหัว ชื่อฟังก์ชัน ทิกเก็ตหรือเซอร์วิสคีย์ และ อินพุตของฟังก์ชัน โดยอักขระส่วนหัวจะเป็นค่าคงที่ซึ่งใช้สำหรับแยกแยะข้อความของระบบจากข้อความแปลกปลอมอื่นๆ ชื่อฟังก์ชันจะเป็นตัวระบุฟังก์ชันที่ข้อความต้องการใช้บริการจากเซอร์วิส สำหรับทิกเก็ตหรือเซอร์วิสคีย์ จะถูกนำมาใช้ตรวจสอบความถูกต้องของข้อความ และสุดท้ายคืออินพุตของฟังก์ชัน จะเป็นพารามิเตอร์ที่ฟังก์ชันนำไปใช้ในการทำงานซึ่งจะมีจำนวนและชนิดของพารามิเตอร์แตกต่างกันออกไปตามฟังก์ชันที่ถูกเรียกใช้บริการ

ในส่วนของข้อความที่แอพพลิเคชั่นได้รับจะประกอบด้วย อักขระส่วนหัวเช่นเหมือนกับข้อความของเซอร์วิส แฟล็กซึ่งใช้แสดงสถานภาพของคำตอบ และคำตอบซึ่งเป็นผลลัพธ์ที่ได้จากการทำงานของเซอร์วิสที่แอพพลิเคชั่นขอใช้บริการ สำหรับในกรณีที่เซอร์วิสขอใช้บริการจากเซอร์วิส

ด้วยกันเอง รูปแบบของข้อความก็จะมีลักษณะเดียวกับข้อความของแอพพลิเคชั่นเนื่องจากเป็นการขอใช้บริการจากเซอร์วิสเช่นเดียวกับแอพพลิเคชั่นนั่นเอง

3.7 การทำงานของระบบ

ในหัวข้อที่ผ่านมาได้กล่าวถึงหน้าที่ของโมดูลต่างๆบนระบบ โดยการจัดแบ่งหน้าที่ให้กับโมดูลด้วยความสอดคล้องถือเป็นส่วนสำคัญส่วนหนึ่งที่จะทำให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพ เพราะระบบมีลักษณะการทำงานเป็นแบบเลเยอร์ซึ่งต้องอาศัยการทำงานร่วมกันของโมดูล ดังนั้นส่วนสำคัญอีกส่วนหนึ่งคือขั้นตอนการทำงานของระบบซึ่งเป็นการจัดลำดับและจังหวะในการทำงานของโมดูลให้สามารถทำงานร่วมกันได้อย่างถูกต้องและมีประสิทธิภาพ การทำงานของระบบมีขั้นตอนที่แตกต่างกันไปขึ้นอยู่วัตถุประสงค์ของงาน เช่น การสตาร์ทอัพเซอร์วิสและแอพพลิเคชั่นของคอนเน็กเตอร์ซึ่งเป็นขั้นตอนหนึ่งของการสตาร์ทอัพระบบ การล็อกอินเข้าสู่ระบบของผู้ใช้ และการทำงานทั่วไปของระบบเพื่อบริการต่อรีเควสท์ของผู้ใช้ ซึ่งจะถูกกล่าวถึงตามลำดับในหัวข้อนี้

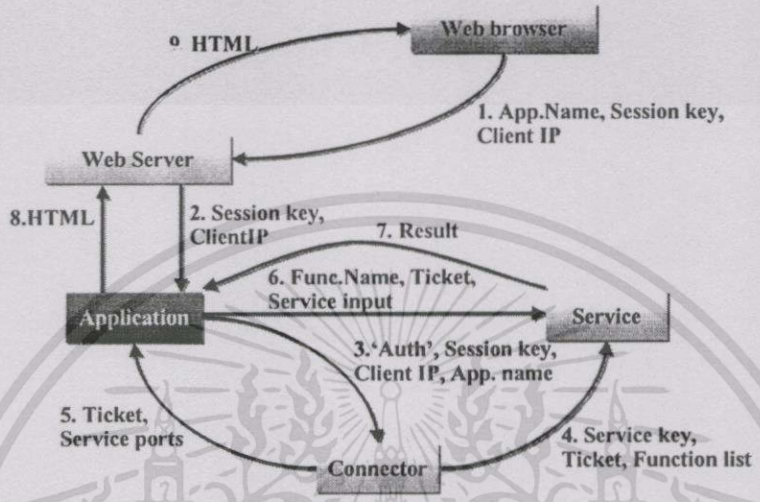
สำหรับขั้นตอนในการติดตั้งระบบได้แก่ การติดตั้งโมดูลของแอพพลิเคชั่น เซอร์วิส และไลบาร์รี สามารถติดตามรายละเอียดได้ในภาคผนวก ข. เรื่องการทำงานกับโปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบ

3.7.1 การล็อกอินเข้าสู่ระบบของผู้ใช้

การล็อกอินเข้าสู่ระบบเป็นขั้นตอนของการตรวจสอบสิทธิของผู้ใช้ โดยผู้ใช้ที่มีผ่านการตรวจสอบจะได้รับเซสชั่นคีย์สำหรับใช้ในการอ้างอิงต่อระบบเมื่อต้องการใช้บริการจากระบบ เซสชั่นคีย์จะถูกสร้างขึ้นโดยคอนเน็กเตอร์และถูกเก็บไว้ที่เวบเบราเซอร์ในรูปของคุ๊กกี้และจะถูกลบออกจากเวบเบราเซอร์เมื่อผู้ใช้ออกจากระบบ การล็อกเอ๊าท์จากระบบมีลักษณะเป็นแบบการทำงานทั่วไปของระบบซึ่งจะถูกกล่าวถึงในหัวข้อถัดไป สำหรับขั้นตอนการล็อกอินมีดังนี้

1. ผู้ใช้ใช้เวบเบราเซอร์ทำการล็อกอินเข้าสู่ระบบที่เวบเพจ Identify.html
2. Identify.html จะทำการส่งชื่อ รหัสผ่าน เลขหมายไอพีของผู้ใช้ให้กับแอพพลิเคชั่น UserSignOn โดยผ่านทางเวบเซิร์ฟเวอร์
3. เมื่อโปรเซสของ UserSignOn เริ่มทำงานก็จะส่งข้อมูลดังกล่าวให้กับคอนเน็กเตอร์เพื่อทำการตรวจสอบสิทธิของผู้ใช้ หลังจากการตรวจสอบคอนเน็กเตอร์จะสร้างเซสชั่นคีย์และทำการบันทึกเซสชั่นคีย์ วันเวลา เลขหมายไอพีของผู้ใช้การเข้าสู่ระบบของผู้ใช้เข้าสู่ฐานข้อมูลของระบบ
4. คอนเน็กเตอร์ทำการส่งเซสชั่นคีย์กลับไปยัง UserSignOn เพื่อให้ UserSignOn นำเซสชั่นคีย์ไปใช้ในการขอใช้บริการจากระบบโดยบริการดังกล่าวถูกจัดอยู่ในการทำงานทั่วไปซึ่งจะถูกกล่าวถึงในหัวข้อถัดไป

5. UserSignOn จะส่งเซสชั่นคีย์ที่ได้รับจากคอนเน็กเตอร์กลับไปยังเวบเบราเซอร์โดยผ่านทางเวบเซิร์ฟเวอร์โดยขอให้เวบเบราเซอร์เก็บค่าดังกล่าวในรูปแบบของคุ๊กกี้ และทำการสลายโปรเซสของตัวมันเองจากระบบ



รูปที่ 3.8 ขั้นตอนการทำงานโดยทั่วไปของระบบ

3.7.2 การทำงานทั่วไป

การทำงานทั่วไปของระบบเมื่อผู้ใช้ทำการล็อกอินเข้าสู่ระบบจนได้รับเซสชั่นคีย์ และทำการเลือกรายการจากเมนูผู้ใช้ที่ปรากฏบนเวบเบราเซอร์เพื่อขอใช้บริการจากแอพพลิเคชั่นของระบบ และสิ้นสุดลงเมื่อแอพพลิเคชั่นส่งผลลัพธ์ทั้งหมดกลับไปยังเวบเบราเซอร์ พิจารณาจากรูปที่ 3.8 ขั้นตอนดังกล่าวสามารถอธิบายได้ดังต่อไปนี้

1. ผู้ใช้ขอใช้บริการจากแอพพลิเคชั่นโดยการเลือกรายการจากเมนูในเวบเบราเซอร์ แล้วเวบเบราเซอร์จึงทำการส่งรีเควสท์จากผู้ใช้ไปยังเวบเซิร์ฟเวอร์ โดยในรีเควสท์นั้นประกอบด้วย ชื่อ อินพุตของแอพพลิเคชั่น เซสชั่นคีย์ และเลขหมายไอพีของผู้ใช้

2. เวบเซิร์ฟเวอร์ทำการพิจารณาตรวจสอบรีเควทส์ที่ได้รับว่าแอพพลิเคชั่นที่รีเควสท์อ้างถึงมีอยู่บนระบบหรือไม่แล้วจึงทำการส่งข้อมูลที่ได้รับจากเวบเบราเซอร์ไปยังแอพพลิเคชั่น ข้อมูลดังกล่าวประกอบด้วยอินพุตของแอพพลิเคชั่น เซสชั่นคีย์และเลขหมายไอพีของผู้ใช้

3. แอพพลิเคชั่นเริ่มทำงานโดยการขอให้คอนเน็กเตอร์ตรวจสอบสิทธิของผู้ใช้จากเซสชั่นคีย์รวมทั้งตรวจสอบสถานะภาพของแอพพลิเคชั่น เมื่อการตรวจสอบเสร็จสิ้นคอนเน็กเตอร์จะทำการสร้างและบันทึกทิกเก็ตคีย์ซึ่งเป็นค่าเฉพาะสำหรับแอพพลิเคชั่นและเซสชั่นคีย์ลงในฐานข้อมูลของคอนเน็กเตอร์ รวมทั้งจัดเตรียมเลขหมายเซิร์ฟเวอร์พอร์ทและรายชื่อฟังก์ชันของเซอร์วิสที่แอพพลิเคชั่นต้องการใช้บริการสำหรับการทำงานในขั้นต่อไปของคอนเน็กเตอร์ ค่าทิกเก็ตคีย์ที่ได้

รับการบันทึกนี้จะถูกนำไปใช้ตรวจสอบในภายหลังที่แอพพลิเคชั่นและเซสชั่นคีย์เดิมได้ร้องขอเพื่อใช้บริการจากระบบอีกทำให้คอนเน็กเตอร์ไม่ต้องสร้างทิกเก็ตคีย์ขึ้นมาใหม่และไม่ต้องส่งทิกเก็ตให้กับเซอร์วิสที่แอพพลิเคชั่นต้องการใช้บริการอีกและกระบวนการอินเตอร์โปรเซสซึ่งถูกลดลงด้วย

4. คอนเน็กเตอร์ทำการส่งรีเควสท์ซึ่งประกอบด้วยทิกเก็ตคีย์ของแอพพลิเคชั่น เซอร์วิสคีย์และรายชื่อฟังก์ชันของเซอร์วิสที่แอพพลิเคชั่นต้องการใช้บริการไปยังเซอร์วิสที่ตำแหน่งเลขหมายพอร์ทที่ได้เตรียมไว้ในข้อ 3 เพื่อแจ้งและขอให้เซอร์วิสบันทึกทิกเก็ตคีย์และรายชื่อฟังก์ชันนี้ไว้เป็นข้อมูลเพื่อใช้ตรวจสอบเมื่อมีรีเควสท์จากแอพพลิเคชั่นมาขอใช้บริการ แต่ก่อนที่จะมีการบันทึกเซอร์วิสจะทำการเปรียบเทียบเซอร์วิสคีย์ที่ได้รับจากคอนเน็กเตอร์ว่ามีค่าตรงกันกับเซอร์วิสคีย์ประจำของเซอร์วิสหรือไม่ เพื่อความแน่นอนว่ารีเควสท์ที่ถูกส่งมาเป็นรีเควสท์ที่มาจากโมดูลของระบบจริงๆ

5. หลังจากคอนเน็กเตอร์ทำการส่งทิกเก็ตให้กับเซอร์วิสที่แอพพลิเคชั่นต้องการใช้บริการเสร็จสิ้นแล้ว คอนเน็กเตอร์ก็จะทำการส่งค่าทิกเก็ตนี้พร้อมด้วยเลขหมายพอร์ทของเซอร์วิสไปให้แอพพลิเคชั่น เพื่อให้แอพพลิเคชั่นนำพอร์ทไปใช้ในการติดต่อโดยตรงกับเซอร์วิสและใช้ทิกเก็ตที่ได้ในการอ้างอิงเพื่อรับบริการจากเซอร์วิส จะสังเกตุได้ว่าการแจ้งเลขหมายพอร์ทของเซอร์วิสให้กับแอพพลิเคชั่นในลักษณะนี้ทำให้คอนเน็กเตอร์สามารถควบคุมและจัดการเลขหมายของพอร์ทได้อย่างเต็มที่

6. ขณะที่แอพพลิเคชั่นทำงานตามหน้าที่ที่ถูกสร้างขึ้นนั้น เมื่อถึงขั้นตอนที่จะต้องมีการใช้บริการจากเซอร์วิส แอพพลิเคชั่นจะส่งรีเควสท์ซึ่งประกอบด้วยชื่อและอินพุตของฟังก์ชันรวมทั้งทิกเก็ตคีย์ไปยังเซอร์วิสด้วยเลขหมายของพอร์ทที่ได้รับจากคอนเน็กเตอร์

7. เซอร์วิสที่ได้รับรีเควสท์จากแอพพลิเคชั่นจะทำการตรวจสอบด้วยการเปรียบเทียบค่าของทิกเก็ตคีย์และชื่อของฟังก์ชันที่แอพพลิเคชั่นต้องการใช้บริการกับข้อมูลที่ได้รับแจ้งจากคอนเน็กเตอร์ในข้อ 5

8. เมื่อเซอร์วิสทำการประมวลผลรีเควสท์ที่ได้รับจากแอพพลิเคชั่นเสร็จสิ้น เซอร์วิสจะทำการส่งผลลัพธ์กลับไปยังแอพพลิเคชั่น

9. เมื่อแอพพลิเคชั่นได้รับผลลัพธ์จากเซอร์วิสแล้วหากต้องการขอใช้บริการจากเซอร์วิสเดิมหรืออื่นๆ แอพพลิเคชั่นก็จะมีการทำงานเหมือนขั้นตอนในข้อ 6-8 จนกว่าการทำงานของแอพพลิเคชั่นทั้งหมดจะเสร็จสิ้น แอพพลิเคชั่นจะนำผลลัพธ์ที่ได้จากการทำงานของแอพพลิเคชั่นมาแปลงให้อยู่ในรูปของเอกสาร HTML แล้วส่งกลับไปยังเวบเซิร์ฟเวอร์

10. เวบเซิร์ฟเวอร์ทำการนำข้อมูลจากแอพพลิเคชั่นที่ได้ส่งกลับไปให้เวบเบราเซอร์ด้วยเซสชั่นเดิมของการเชื่อมต่อระหว่างเวบเซิร์ฟเวอร์และเวบเบราเซอร์ เมื่อการส่งข้อมูลเสร็จสิ้นการเชื่อมต่อของเซสชั่นก็จะยุติลง เวบเบราเซอร์จะนำผลลัพธ์ที่ได้ไปแสดงแก่ผู้ใช้ในที่สุด

3.6 สรุป

จากหลักการต่างๆที่ได้นำเสนอไปสามารถนำไปสร้างต้นแบบของโปรแกรมจัดการเครือข่ายที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับอุปกรณ์เครือข่าย มีโครงสร้างที่สนับสนุนการเปลี่ยนแปลงโมดูลโดยไม่กระทบต่อโมดูลที่ไม่เกี่ยวข้อง และมีระบบรักษาความปลอดภัยที่สามารถควบคุมการทำงานของโมดูลที่อยู่เลเยอร์ทุกเลเยอร์

บทที่ 4

การทดลองและผลการทดลอง

หลักและวิธีการที่ได้กล่าวไว้ในบทที่ผ่านมา ในบทต่อไปนี้จะกล่าวถึงการเครื่องมือที่ใช้ในการพัฒนาระบบและการนำระบบมาทดสอบเพื่อพิสูจน์ว่าโครงสร้างและขั้นตอนการทำงานของระบบสามารถทำงานได้จริง

4.1 การพัฒนาระบบ

ในการพัฒนาระบบผู้ทำวิจัยได้เลือกให้ระบบทำงานบนระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 95 เนื่องจากมีความคุ้นเคยกับระบบปฏิบัติการชนิดนี้มาก่อนรวมทั้งมีการใช้งานกันอย่างแพร่หลายและเลือกใช้ภาษาปาสคาลและคอมไพเลอร์ของบอร์แลนด์เดลไฟ เวอร์ชัน 5.0 ด้วยคุณสมบัติที่สนับสนุนการสร้างเวบเซิร์ฟเวอร์แอพพลิเคชั่นแบบ ISAPI ความสะดวกในการทำโปรแกรมที่ต้องมียูสเซอร์อินเตอร์เฟสเป็นแบบกราฟฟิก ประกอบกับชุดคำสั่งของโปรโตคอล SNMP ของบริษัท Dart Communications ที่นำมาใช้ก็สามารถใช้ร่วมกับคอมไพเลอร์ได้เป็นอย่างดี ในส่วนของฐานข้อมูลเนื่องจากในงานวิจัยนี้ไม่ได้เน้นที่การออกแบบและจัดการฐานข้อมูล ดังนั้นการเลือกใช้ชนิดของฐานข้อมูลจึงใช้ชนิดที่คอมไพเลอร์สนับสนุนคือ พาราด็อกส์ เวอร์ชัน 7.0 สำหรับเวบเซิร์ฟเวอร์ได้เลือกใช้ Personel Web Server 1.0a ของไมโครซอฟต์เนื่องจากสนับสนุนการทำงานของเวบเซิร์ฟเวอร์แอพพลิเคชั่นแบบ ISAPI ส่วนเวบเบราเซอร์ซึ่งจะต้องใช้เป็นส่วนติดต่อกับผู้ใช้ สามารถใช้ได้ทั้ง Internet Explorer และ Netscape Navigator

การพัฒนาระบบเมื่อทำตามหลักการที่ได้กล่าวไปแล้วในบทที่ 3 ลักษณะของระบบจะประกอบด้วยแอพพลิเคชั่นหลายชนิดตั้งแต่วินโดว์แอพพลิเคชั่น เวบเซิร์ฟเวอร์แอพพลิเคชั่น และเอกสาร HTML โดยส่วนของโมดูลที่อยู่ในชั้นแอพพลิเคชั่นจะเป็นโมดูลพวกเวบเซิร์ฟเวอร์แอพพลิเคชั่นซึ่งจะมีบางโมดูลที่ใช้เอกสาร HTML ในการรับอินพุตข้อมูลจากผู้ใช้ ในขณะที่โมดูลในชั้นเซอร์วิสจะเป็นวินโดว์แอพพลิเคชั่นซึ่งเป็นเอ็กซิคิวชั่นไฟล์ และส่วนของการติดต่อสื่อสารระหว่างโปรเซส ผู้ทำวิจัยได้เลือกใช้วินโดว์ซ็อกเก็ตเป็นช่องทางในการสื่อสาร ทั้งนี้เนื่องจากมีความคุ้นเคยกับการทำงานในลักษณะนี้อยู่ก่อนแล้ว สำหรับการจัดการข้อมูลเริ่มต้นของระบบสามารถใช้โปรแกรม Database Desktop ซึ่งมาพร้อมกับบอร์แลนด์เดลไฟ ข้อมูลดังกล่าวได้แก่ ข้อมูลของผู้ใช้ ข้อมูลของแอพพลิเคชั่นและเซอร์วิส เป็นต้น

4.2 หลักการทดลอง

เนื่องจากงานวิจัยนี้ถูกสร้างขึ้นด้วยแนวคิดที่ต้องการนำเสนอต้นแบบของโปรแกรมจัดการเครือข่ายที่มีแพล็ตฟอร์มที่มีความยืดหยุ่นในการปรับเปลี่ยนโมดูลของระบบโดยไม่กระทบต่อโมดูลอื่นที่ไม่เกี่ยวข้อง มีการรักษาความปลอดภัยให้กับระบบโดยควบคุมการทำงานของผู้ใช้และการทำงานภายในของระบบ และควบคุมการใช้โปรโตคอล SNMP ของผู้ใช้ในการเข้าถึงอุปกรณ์เครือข่าย ดังนั้นแนวคิดดังกล่าวจะถูกนำมาใช้เป็นเป้าหมายในการพิสูจน์และตัดสินว่าระบบสามารถทำงานได้ตามเป้าหมายหรือไม่ ผลการทำงานออกมาเป็นอย่างไร และมีข้อบกพร่องที่ควรปรับปรุงแก้ไขประการใดบ้าง

4.3 วิธีการและขั้นตอนการทดลอง

วิธีการคือจะสมมุติสถานการณ์ของการทดลองให้เริ่มจากการติดตั้งโปรแกรมซึ่งจะประกอบด้วยขั้นตอนต่างๆหลายขั้นตอนเช่น การติดตั้งโมดูลเข้าสู่ระบบ การใส่รายชื่อผู้ใช้ลงในฐานข้อมูลของระบบ และการถอดถอนโปรแกรม ในการทดลองจะควบคุมสภาพแวดล้อมได้แก่ เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอ็นต์ เครื่องคอมพิวเตอร์ที่ทำหน้าเป็นผู้จัดการเครือข่าย โปรแกรมเวบเบราเซอร์ โปรแกรมเวบเซิร์ฟเวอร์ และระบบปฏิบัติการ เครื่องมือดังกล่าวมีรายละเอียดดังนี้

4.3.1 เครื่องมือที่ใช้ในการทดลอง

- เวบเบราเซอร์ : อินเตอร์เนตเอ็กพลอเรอร์ 4.0
- เวบเซิร์ฟเวอร์ : ไมโครซอฟท์ เพอร์ซันเนล เวบเซิร์ฟเวอร์ 1.0a
- เครื่องผู้จัดการเครือข่าย : ระบบปฏิบัติการ ไมโครซอฟท์วินโดว์ 95 หน่วยประมวลผลกลาง เอเอ็มดี 230 MHz หน่วยความจำหลัก 64 เมกกะไบท์ ความจุฮาร์ดดิส 3 กิกะไบต์
- เครื่องลูกข่าย : ระบบปฏิบัติการ ไมโครซอฟท์วินโดว์ 95 หน่วยประมวลผลกลาง เอเอ็มดี 230 MHz หน่วยความจำหลัก 64 เมกกะไบท์ ความจุฮาร์ดดิส 3 จิกะไบต์

4.3.2 ขั้นตอนการทดลอง

1. ทดลองการติดตั้งเซอร์วิสและแอพพลิเคชั่นเข้าสู่ระบบ
2. ทดลองการถอดถอนแอพพลิเคชั่นออกจากระบบ
3. ทดลองการถอดถอนเซอร์วิสออกจากระบบ
4. ทดลองการรักษาความปลอดภัยของระบบ
5. ทดลองการทำงานทั่วไปของระบบ

4.4 การทดลอง

4.4.1 การติดตั้งเซอร์วิสและแอพพลิเคชั่น

เป็นการทดลองเพื่อต้องการแสดงตัวอย่างขั้นตอนการติดตั้งแอพพลิเคชั่นและเซอร์วิสเข้าสู่ระบบ และแสดงให้เห็นว่าเซอร์วิสและแอพพลิเคชั่นที่ผ่านการติดตั้งสามารถทำงานได้จริง ในการทดลองจะสมมุติสถานการณ์ให้ติดตั้งแอพพลิเคชั่น SNMPMIBBrowser และเซอร์วิสเน็ตเซอร์วิส สาเหตุที่เลือกติดตั้งพร้อมกันเนื่องมาจาก SNMPMIBBrowser ทำงานโดยเรียกใช้บริการจากเน็ตเซอร์วิส ดังนั้นจึงเป็นการสะดวกที่จะแสดงให้เห็นถึงการทำงานร่วมกันของแอพพลิเคชั่นและเซอร์วิสหลังจากการติดตั้ง ระบบก่อนการทดลองประกอบด้วย

เซอร์วิส : คอนเนกเตอร์และเมสเสจเมเนเจอร์

แอพพลิเคชั่น : UserSignOn UserSignOff และ MesgBoard

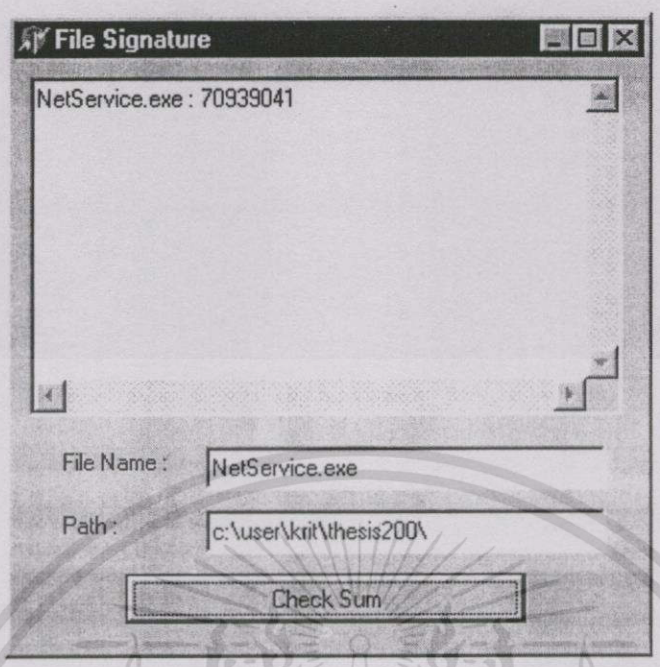
เมนูของผู้ใช้ : Sign Off

ผู้ใช้ : Admin ระดับสิทธิ 0 และ UserA ระดับสิทธิ 100

เซอร์วิสที่ต้องการติดตั้งเข้าสู่ระบบคือ NetService เป็นเวอร์ชัน 1.0 มีค่าไฟล์ซิกเนเจอร์เป็น 70939041 มีเซอร์วิสพอร์ทหมายเลข 6000 และเป็นเซอร์วิสที่ประกอบด้วยฟังก์ชันต่างๆ ดังนี้

ตารางที่ 4.1 ฟังก์ชันของ NetService

ลำดับ	ชื่อฟังก์ชัน	ระดับสิทธิ
1	Dead	0
2	GetServiceKey	0
3	InsertTicket	0
4	GetRequest	100
5	GetNextRequest	100
6	SetRequest	100
7	SNMPDiscovery	0
8	MapObject	100
9	NodeInfo	100
10	SubnetInfo	100



รูปที่ 4.1 โปรแกรม CheckSum

แอพพลิเคชั่นที่ติดตั้งคือ SNMPMIBBrowser เวอร์ชัน 1.0 เรียกใช้บริการของฟังก์ชันจากเน็ตเซอร์วิสดังนี้ GetNextRequest, GetRequest และ SetRequest เมื่อพิจารณาจากหลักเกณฑ์การให้ระดับสิทธิของแอพพลิเคชั่น SNMPMIBBrowser จะมีระดับสิทธิเท่ากับ 100

ขั้นตอนการทดลอง

1. ทำการชัทดาวน์ระบบ
2. ใช้โปรแกรม Database Desktop ในการป้อนข้อมูลลงสู่ตารางที่เกี่ยวข้อง ดังรูป

ตารางที่ 4.2 ข้อมูลที่ถูกป้อนลงในตาราง ServiceTable ด้วยโปรแกรม Database Desktop

ลำดับ	ชื่อฟิลด์	ค่าที่ป้อน
1.	SerName	NetService
2.	Version	1.0
3.	FileSignature	70939041
4.	Port	6000
5.	Detail	Add at 20th February 23, 2001 By Teerakrit.

ตารางที่ 4.3 ข้อมูลที่ถูกป้อนลงในตาราง FunctionTable ด้วยโปรแกรม Database Desktop

ลำดับ	ชื่อเซอร์วิส	ชื่อฟังก์ชัน	ระดับสิทธิ
1	NetService	Dead	0
2	NetService	GetServiceKey	0
3	NetService	InsertTicket	0
4	NetService	GetRequest	100
5	NetService	GetNextRequest	100
6	NetService	SetRequest	100
7	NetService	SNMPDiscovery	0
8	NetService	MapObject	100
9	NetService	NodeInfo	100
10	NetService	SubnetInfo	100

ตารางที่ 4.4 ข้อมูลของแอพพลิเคชั่น SNMPMIBBrowser ที่ถูกป้อนสู่ตาราง AppTable

ลำดับ	ชื่อฟิลด์	ค่าที่ป้อน
1.	AppName	SNMPMIBBrowser
2.	Label	<none>
3.	Version	1.0
4.	Level	100
5.	ServiceInfo	1!NetService#1.0#3#GetNextRequest#GetRequest#SetRequest

ตารางที่ 4.5 ข้อมูลของแอพพลิเคชั่น SNMPMIBBrowser ที่ถูกป้อนสู่ตาราง MenuList

ลำดับ	ชื่อฟิลด์	ค่าที่ป้อน
1.	ServiceName	MIB Browser
2.	WebPath	MIBBrowserInf.html
3.	AppName	SNMPMIBBrowser

<จำนวนเซอร์วิส>!<ข้อมูลเซอร์วิส 1>!<ข้อมูลเซอร์วิส 2>!...!<ข้อมูลเซอร์วิส N>

รูปที่ 4.2 ก. รูปแบบ ServiceInfo ของตาราง AppTable

<ชื่อเซอร์วิส>#<เวอร์ชัน>#<จำนวนฟังก์ชัน>#<ชื่อฟังก์ชัน 1>#...#<ชื่อฟังก์ชัน N>

รูปที่ 4.2 ข. รูปแบบข้อมูลเซอร์วิสของ ServiceInfo ของตาราง AppTable

ในการป้อนข้อมูลของฟิลด์ ServiceInfo ของตาราง AppTable จะต้องใส่ให้อยู่ในรูปแบบดังรูปที่ 4.2 ก. และ ข. ดังนั้นในกรณีของแอพพลิเคชั่น SNMPMIBBrowser ฟิลด์ ServiceInfo จะมีค่า 1!NetService#1.0#3#GetNextRequest#GetRequest#SerRequest ซึ่งแปลว่าแอพพลิเคชั่นจะทำการเรียกใช้บริการของ NetService เวอร์ชัน 1.0 เพียงเซอร์วิสเดียว และเรียกใช้ฟังก์ชัน 3 ฟังก์ชันได้แก่ GetNextRequest, GetRequest และ SetReuqest หลังการติดตั้ง NetService และ SNMPMIBBrowser ระบบประกอบด้วย

เซอร์วิส : คอนเนกเตอร์ เมสเสจเมเนเจอร์ และเน็ทเซอร์วิส

แอพพลิเคชั่น : UserSignOn UserSignOff MesgBoard และ SNMPMIBBrowser

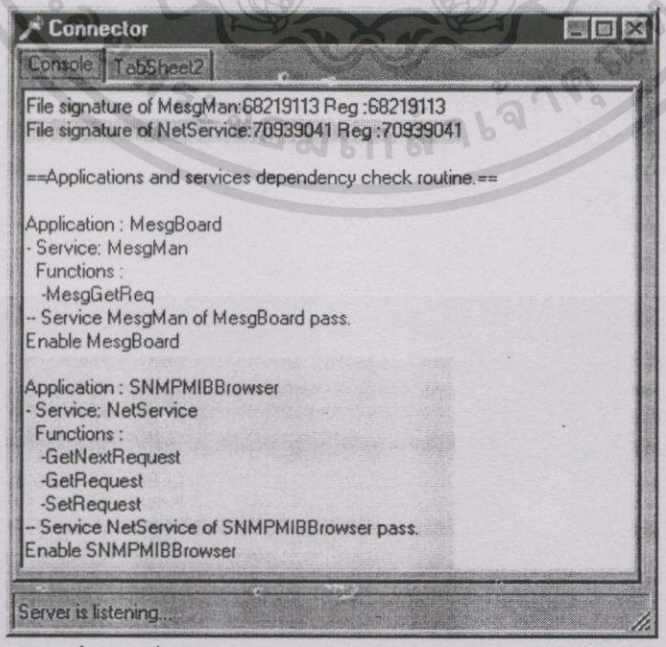
เมนูของผู้ใช้ : Sign Off และ MIB Browser

ผู้ใช้ : Admin ระดับสิทธิ 0 และ UserA ระดับสิทธิ 100

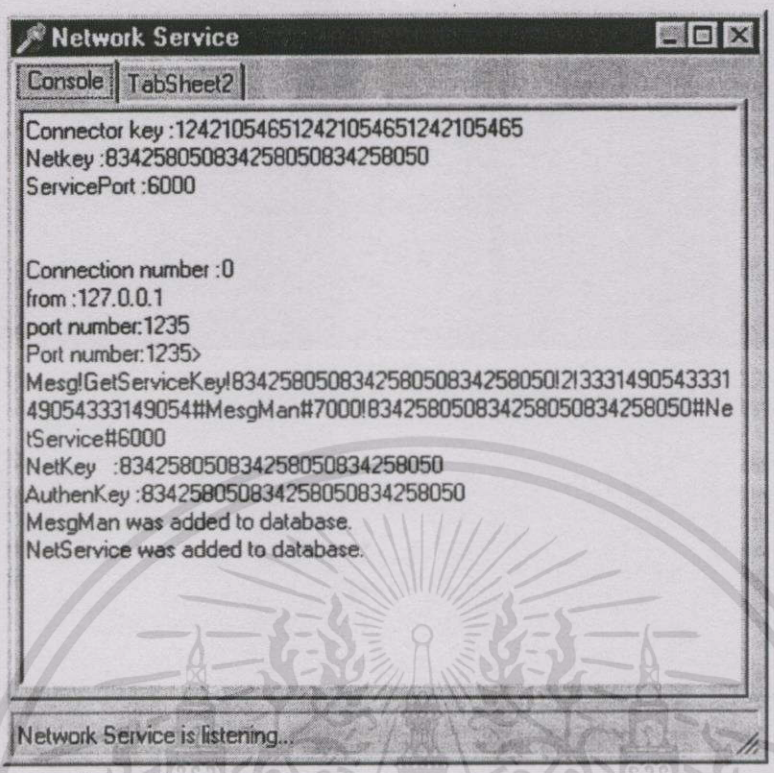
3. ทำการสตาร์ทอัพระบบ

3.1 คอนเนกเตอร์ทำการสตาร์ทอัพเซอร์วิสให้กับระบบ เซอร์วิสดังกล่าวได้แก่ แมสเสจเมเนเจอร์ และเน็ตเซอร์วิสพิจารณาจากไฮไลท์ด้านบนของรูปที่ 4.3 โดยรูปที่ 4.4 จะเป็นคอนโซลของเน็ตเซอร์วิสหลังจากถูกสตาร์ทอัพ

3.2 เมื่อสตาร์ทอัพเซอร์วิสเสร็จสิ้น คอนเนกเตอร์ทำการอีเนเบิลแอพพลิเคชั่น SNMPMIBBrowser พิจารณาจากไฮไลท์ล่างของรูปที่ 4.3

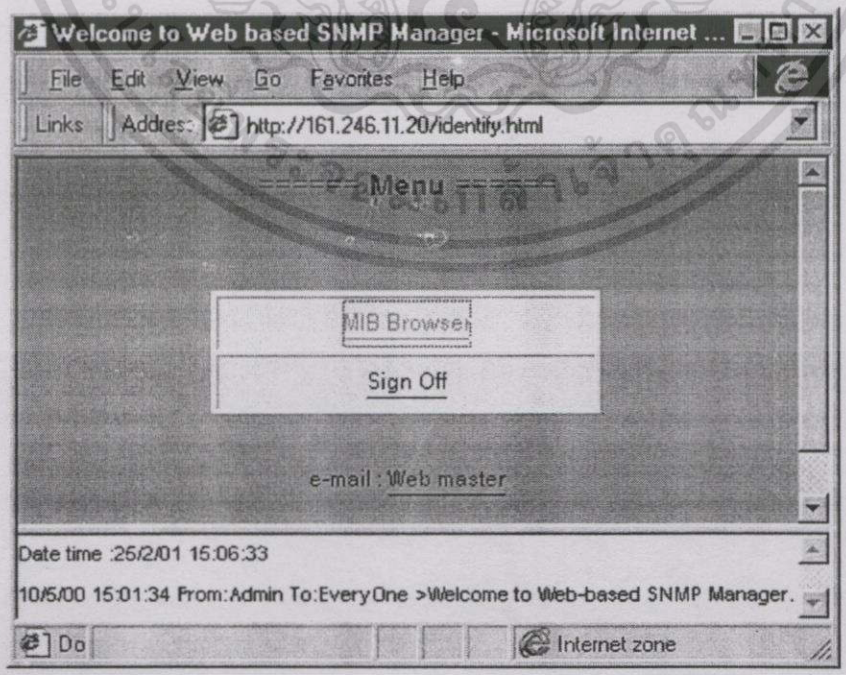


รูปที่ 4.3 คอนโซลของคอนเน็กเตอร์

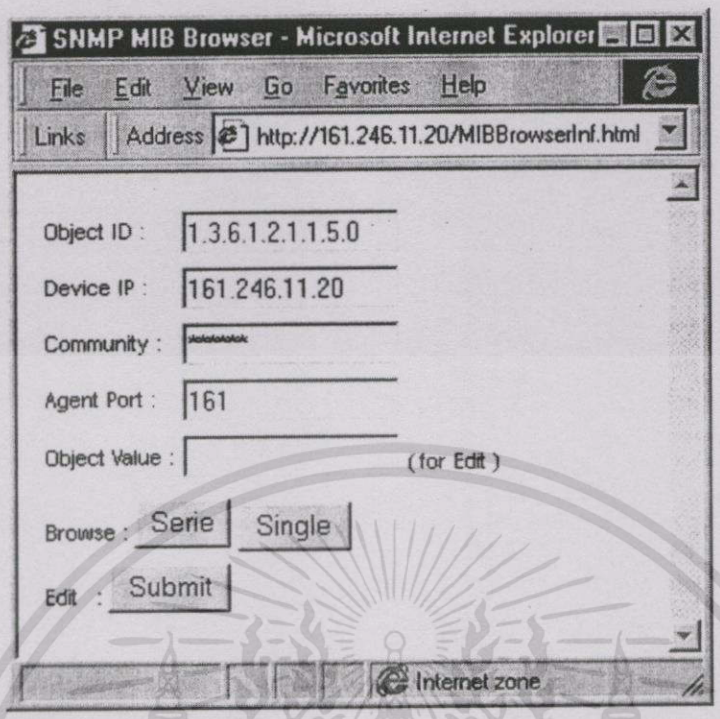


รูปที่ 4.4 คอนโซลของเน็ตเซอร์วิส

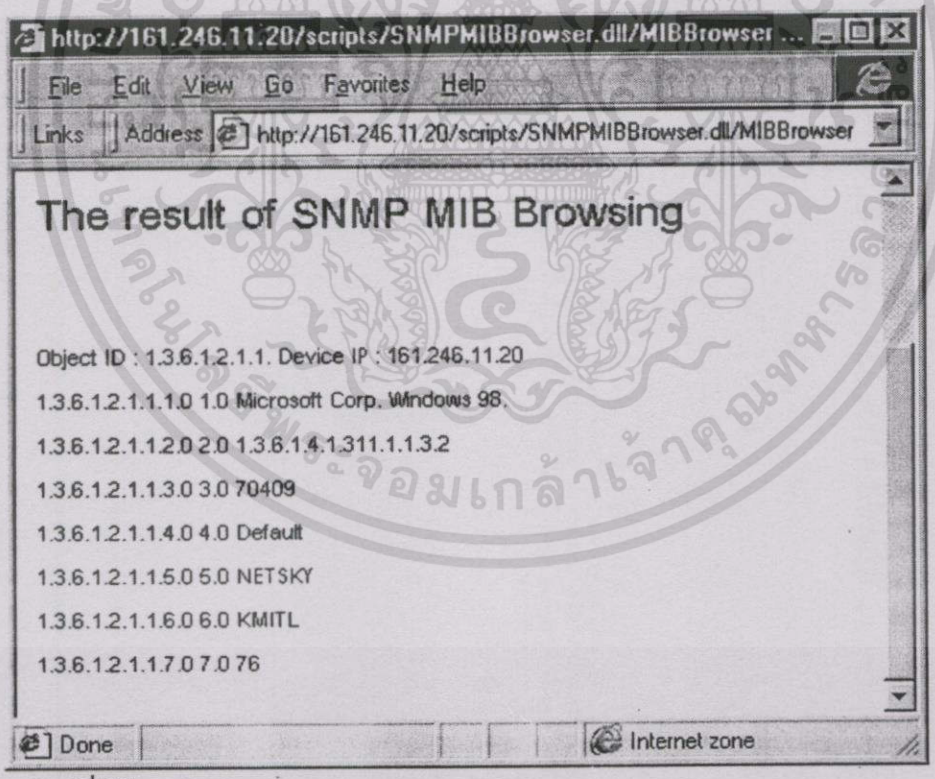
4.ทำการล็อกอินเข้าสู่ระบบ โดยใช้ UserA และลองเรียกใช้แอพพลิเคชั่น MIB Browser และสั่งให้ทำงาน จากรูปที่ 4.5 จะเป็นเมนูของ UserA โดยรูปที่ 4.6 และ 4.7 จะเป็นรูปการเรียกใช้แอพพลิเคชั่น SNMPMIBBrowser ของ UserA



รูปที่ 4.5 หน้าจอเมนูผู้ใช้ของ UserA



รูปที่ 4.6 หน้าจอของรายการ MIB Browser



รูปที่ 4.7 ผลลัพธ์ที่ได้จากการทำงานของ MIB Browser

4.4.2 การถอดถอนแอพพลิเคชั่น

เพื่อแสดงตัวอย่างของขั้นตอนและผลของการถอดถอนแอพพลิเคชั่น โดยสมมุติสถานการณ์ให้ผู้ดูแลระบบทำการถอดถอนแอพพลิเคชั่น MIB Browser ออกจากระบบ ระบบก่อนการทดลองประกอบด้วย

เซอร์วิส : คอนเนกเตอร์ เมสเสจเมเนเจอร์ และเน็ทเซอร์วิส

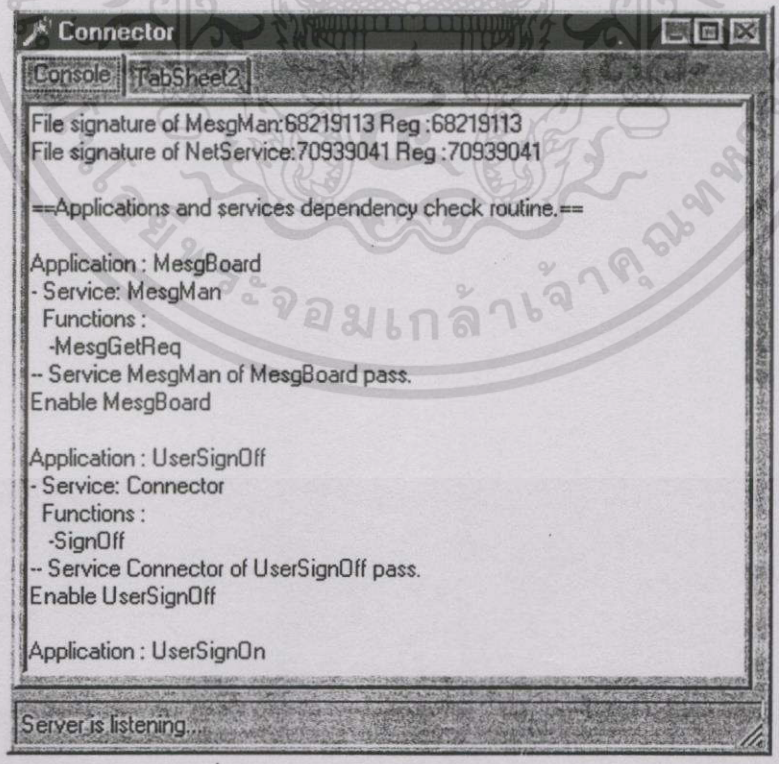
แอพพลิเคชั่น : UserSignOn UserSignOff MesgBoard และ SNMPMIBBrowser

เมนูของผู้ใช้ : Sign Off และ MIB Browser

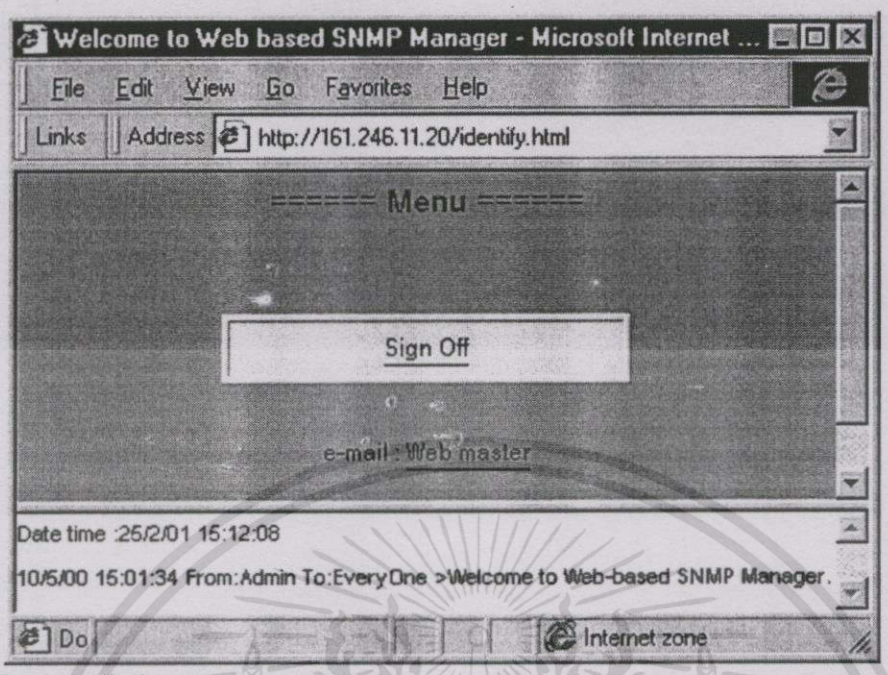
ผู้ใช้ : Admin ระดับสิทธิ 0 และ UserA ระดับสิทธิ 100

ขั้นตอนการทดลอง

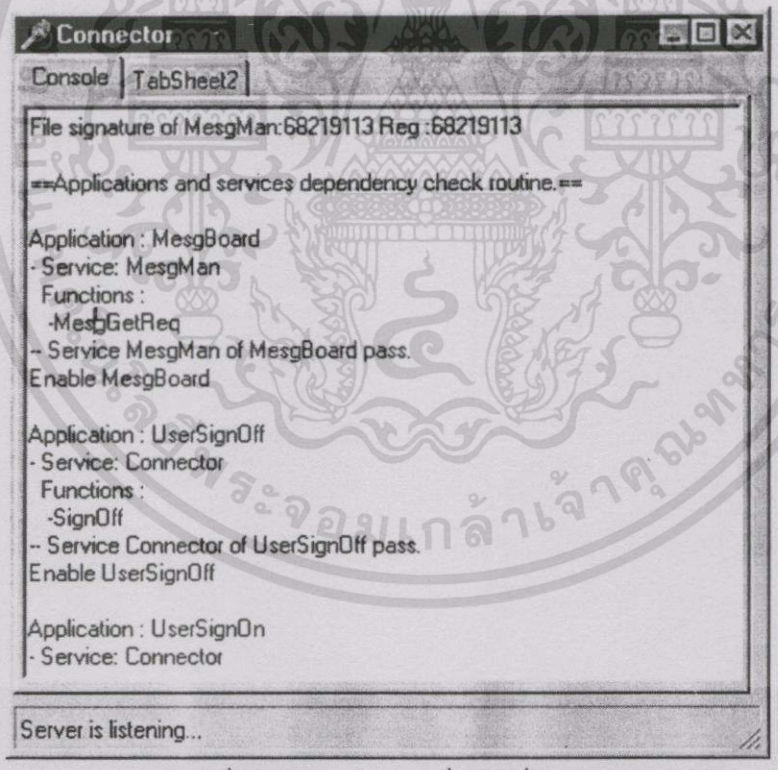
1. ทำการชัทดาวน์ระบบ
2. ทำการลบข้อมูลของ SNMPMIBBrowser ที่ได้ป้อนให้กับระบบในการทดลอง 4.3.1 จากตาราง AppTable และ MenuList
3. เมื่อการสตาร์ทอัพระบบเข้าสู่ขั้นตอนการอีเนเบิลแอพพลิเคชั่น คอนเนกเตอร์จะไม่อีเนเบิล SNMPMIBBrowser พิจารณาเปรียบเทียบรูปที่ 4.3 กับ 4.8
4. ทำการล็อกอินเข้าสู่ระบบโดยใช้ UserA จะสังเกตได้ว่าในเมนูของผู้ใช้ไม่ปรากฏรายการของ MIB Browser พิจารณาจากรูปที่ 4.9



รูปที่ 4.8 คอนโซลของคอนเนกเตอร์



รูปที่ 4.9 หน้าจอเมนูผู้ใช้ของ UserA



รูปที่ 4.10 คอนโซลของคอนเนกเตอร์หลังจากถอดถอนเน็ตเซอร์วิสออกจากระบบ

4.4.3 การถอดถอนเซอร์วิส

เพื่อแสดงตัวอย่างของขั้นตอนและผลของการถอดถอนเซอร์วิส โดยสมมุติสถานการณ์ให้ผู้ดูแลระบบทำการถอดถอนเซอร์วิส NetService ออกจากระบบ

1. ทำการชัทดาวน์ระบบ

2. ทำการลบข้อมูลของเน็ตเซอร์วิสที่ได้จากการติดตั้งเซอร์วิสในการทดลอง 4.3.1 จากตาราง ServiceTable และ FunctionList

3. ทำการสตาร์ทอัพระบบขึ้น จะสังเกตได้ว่าไม่ปรากฏเซอร์วิสเน็ตเซอร์วิสถูกสตาร์ทอัพ พิจารณาเปรียบรูปที่ 4.10 คอนโซลของคอนเนกเตอร์หลังจากการถอดถอนเน็ตเซอร์วิสกับรูปที่ 4.3 คอนโซลของคอนเนกเตอร์หลังการติดตั้งเน็ตเซอร์วิส

4.4.4 การทดสอบความปลอดภัยของระบบ

เพื่อแสดงวิธีการรักษาความปลอดภัยในลักษณะต่างๆของระบบ ซึ่งประกอบด้วยการรักษาความปลอดภัยในระดับผู้ใช้ ระดับแอพพลิเคชั่น ระดับเซอร์วิส ดังนั้นในการทดลองจะสมมุติเหตุการณ์ขึ้นเพื่อให้เหมาะสมกับแต่ละลักษณะของการรักษาความปลอดภัย

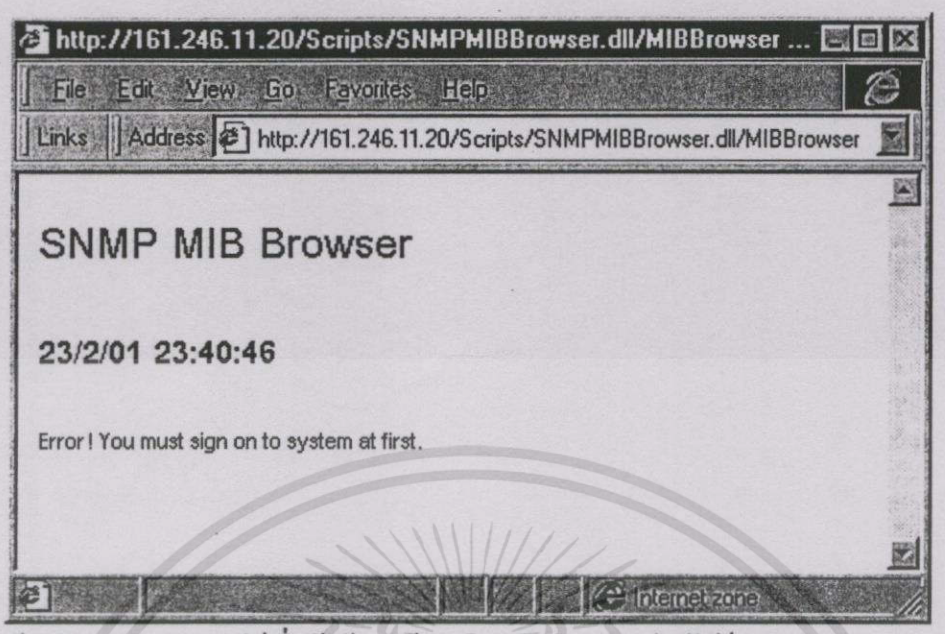
ระดับผู้ใช้

1. ทำการทดลองโดยสมมุติเหตุการณ์ให้ผู้ใช้ขอใช้บริการ SNMPMIBBrowser จากระบบโดยไม่ได้ล็อกอินเข้าสู่ระบบ โดยใช้ URL: http://161.246.11.20/Scripts/SNMPMIBBrowser.dll/MIBBrowser

ผลของการทดลองคือระบบจะขอเซสชั่นคีย์จากเวบเบราเซอร์ซึ่งในกรณีนี้เมื่อยังไม่ได้ล็อกอินก็จะไม่มีเซสชั่นคีย์ที่ถูกต้องให้กับระบบ คำตอบที่ได้คือระบบจะปฏิเสธการให้บริการดังรูปที่ 4.11

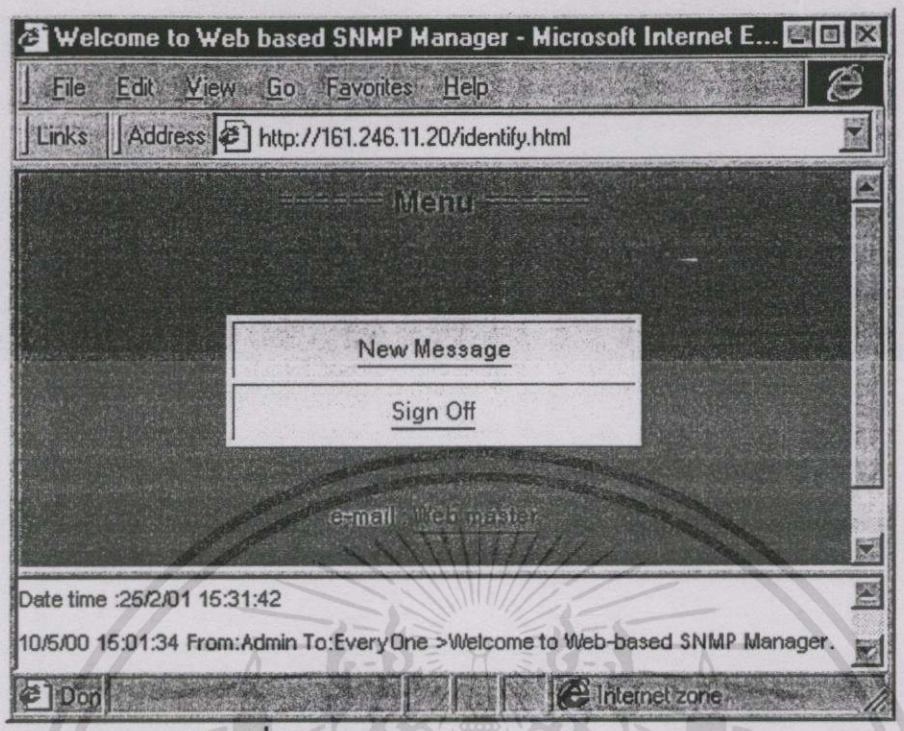
2. สมมุติสถานการณ์ให้ UserB ที่มีระดับสิทธิเท่ากับ 200 ทำการล็อกอินเข้าสู่ระบบและขอใช้บริการแอพพลิเคชั่น NetworkMap ที่มีระดับสิทธิเท่ากับ 100 (รูปที่ 4.12 แสดงแอพพลิเคชั่น NetworkMap บนระบบ) ซึ่งผู้ใช้มีสิทธิไม่เพียงพอด้วยการเข้าถึงโดยตรงโดยใช้ URL เช่นเดียวกับในข้อที่ 1 แต่แตกต่างกันตรงที่การทดลองนี้เวบเบราเซอร์ของผู้ใช้จะได้รับเซสชั่นคีย์จากการล็อกอินและใช้เซสชั่นคีย์ในการติดต่อขอใช้บริการจากระบบ

จากการทดลองจะพบว่าเมื่อ UserB ล็อกอินเข้าสู่ระบบเมนูของผู้ใช้จะปรากฏรายการที่ผู้ใช้มีสิทธิในการใช้บริการคือ New Message กับ Sign Off (จากรูปที่ 4.13) แต่ผู้ใช้กลับใช้ URL : http://161.246.11.20/script/NetworkMap.dll/NetworkMap ในการเข้าถึงแอพพลิเคชั่น NetworkMap ซึ่งผลการทดลองจะพบว่าระบบปฏิเสธการให้บริการเนื่องจากตรวจพบจากเซสชั่นคีย์ว่าผู้ใช้มีสิทธิไม่เพียงพอที่จะใช้บริการแอพพลิเคชั่น NetworkMap พิจารณาจากรูปที่ 4.14 ทำให้ทราบว่าระบบป้องกันการใช้บริการระบบจากผู้ใช้ที่มีสิทธิไม่เพียงพอ

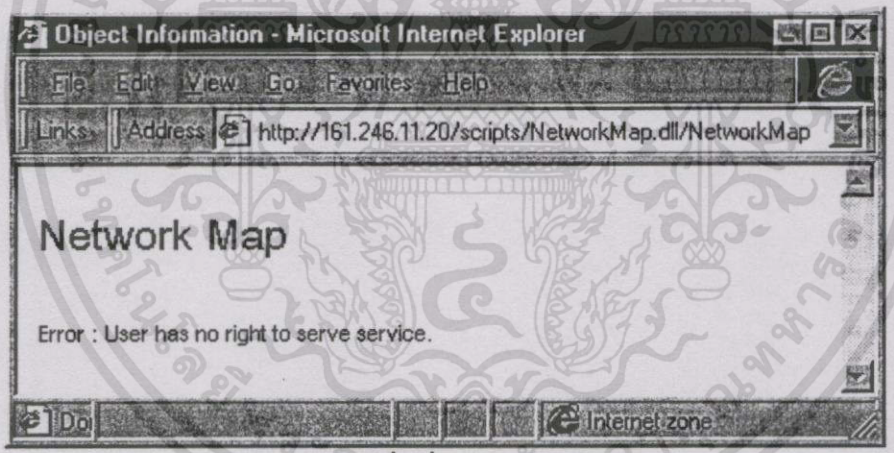


รูปที่ 4.11 หน้าจอของเวบเบราเซอร์เมื่อผู้ใช้ขอใช้บริการจากระบบโดยไม่ได้ล็อกอิน

รูปที่ 4.12 คอนโซลของคอนเนกเตอร์เมื่อเปิดบริการแอพพลิเคชั่น NetworkMap



รูปที่ 4.13 หน้าจอเมนูของ UserB ที่มีระดับสิทธิ 200

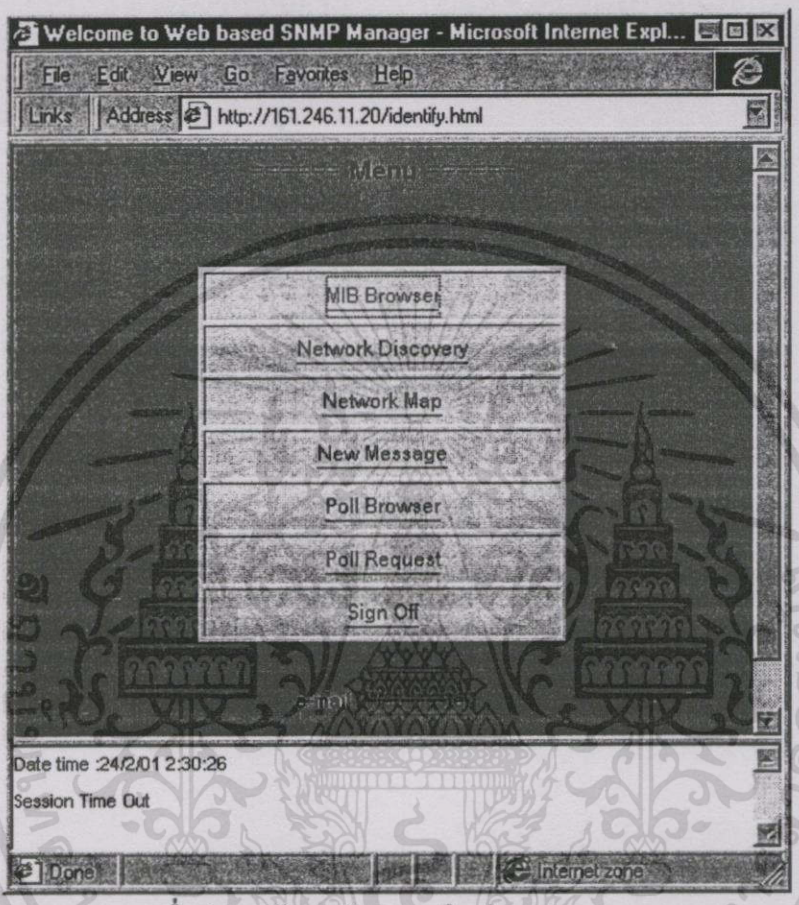


รูปที่ 4.14 ผลลัพธ์จากการขอใช้งานแอพพลิเคชั่นที่ UserB ไม่มีสิทธิใช้

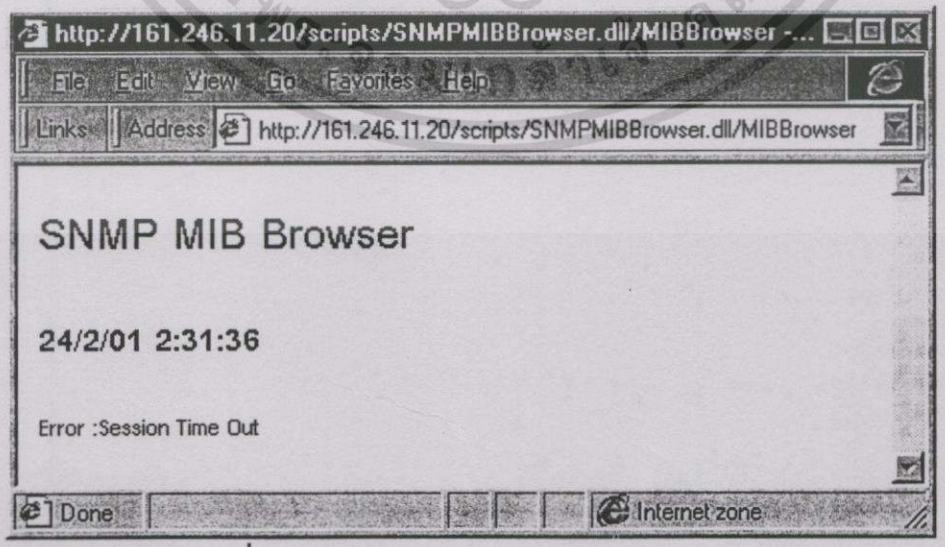
3. เป็นการทดลองการตรวจสอบไทม์เอ๊าท์ของเซสชั่นของผู้ใช้ ไทม์เอาท์เกิดจากการที่ผู้ใช้ทิ้งระยะเวลาในการติดต่อกับระบบนานเกินกว่าที่ได้กำหนดไว้ ระบบจะปฏิเสธการให้บริการต่อผู้ใช้ที่อยู่ในสภาวะไทม์เอาท์ เพื่อป้องกันไม่ให้ผู้อื่นสามารถสวมรอยขอใช้บริการจากระบบได้ ส่วนใหญ่จะเกิดขึ้นเมื่อผู้ใช้ของระบบออกจากระบบโดยไม่ได้ทำการล็อกอ๊อฟ ทำให้เซสชั่นคีย์ยังคงติดอยู่ในฐานข้อมูลของเวบเบราเซอร์ การใช้ไทม์เอ๊าท์สามารถจำกัดข้อผิดพลาดในลักษณะนี้ได้ การทดลองจะสมมุติให้ UserA ทำการล็อกอินเข้าสู่ระบบแล้วทิ้งช่วงไว้เกินกว่าเวลาที่กำหนดไว้คือ 5 นาที

ผลการทดลองคือ เมื่อทิ้งระยะเวลาไว้เกิน 5 นาทีหลังจากล็อกอิน สิ่งที่สามารถสังเกตุได้คือ เมสเสจบอร์ดที่อยู่ใต้เมนูของผู้ใช้ซึ่งต้องใช้เซสชั่นคีย์เพื่อส่งรีเควสท์ขอข้อมูลที่เป็นข่าวสารมาแสดงไว้

บนบอร์ด แต่การส่งรีเควสท์ของเมสเสจบอร์ดไม่นับว่าเป็นการติดต่อระบบของผู้ใช้ ดังนั้นเมื่อเกิดไทม์เอ๊าท์ขึ้นเมสเสจบอร์ดจะได้รับการปฏิเสธบริการข่าวสาร ผลลัพธ์ที่แสดงจึงเป็นเป็นดังรูปที่ 4.15 และ ในขณะที่ UserA เมื่อทำการขอใช้บริการ MIB Browser ก็จะได้รับการปฏิเสธเช่นกันดังรูปที่ 4.16



รูปที่ 4.15 หน้าจอของ UserA เมื่อทำการติดต่อกับระบบในช่วง Time out

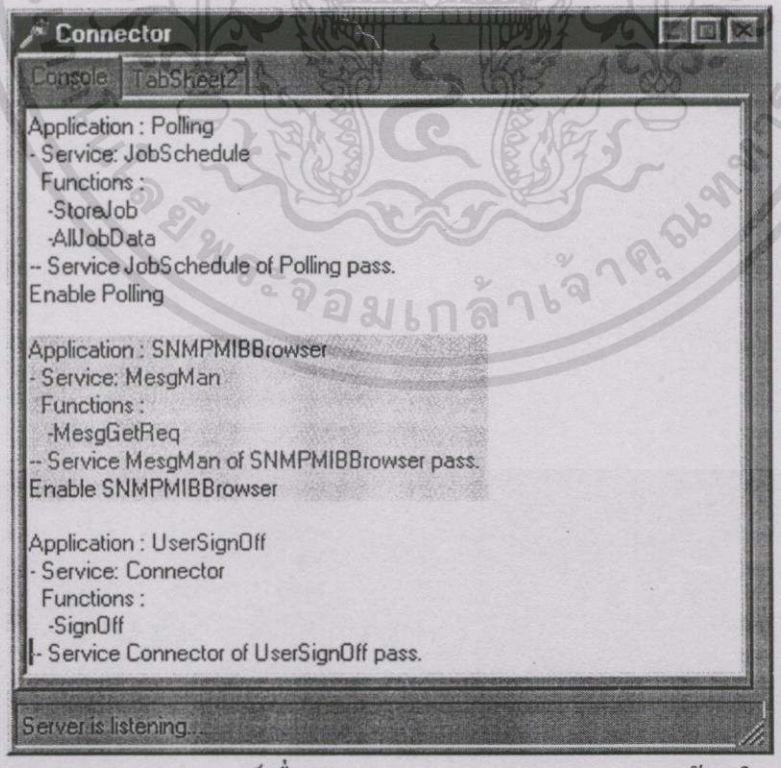


รูปที่ 4.16 หน้าจอของ UserA เมื่อเรียกใช้ MIB Browser ในช่วง Time out

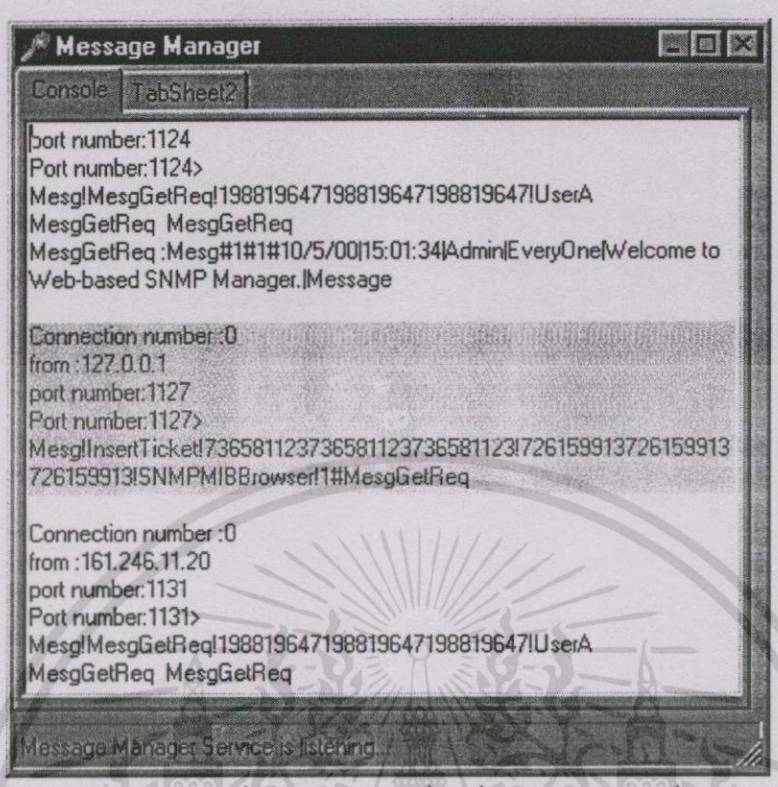
ระดับแอพพลิเคชั่น

ในระดับแอพพลิเคชั่นมีพารามิเตอร์ที่ใช้ควบคุมการรักษาความปลอดภัยคือฟิลด์ ServiceInfo ที่อยู่ในตาราง AppTable และทิกเก็ตคีย์ สำหรับทิกเก็ตคีย์จะอยู่ในการทดลองเรื่องการทำงานทั่วไปซึ่งเป็นหัวข้อถัดไป ในหัวข้อนี้จะทำการทดลองเพื่อแสดงให้เห็นว่าระบบสามารถควบคุมให้แอพพลิเคชั่นใช้บริการฟังก์ชันของเซอร์วิสตามข้อมูลที่ได้แจ้งไว้ตอนติดตั้งแอพพลิเคชั่นเข้าสู่ระบบ โดยฟิลด์ ServiceInfo จะเป็นตัวบอกให้ระบบทราบว่าแอพพลิเคชั่นต้องการใช้ฟังก์ชันจากเซอร์วิสใดบ้าง ในการทดลองสมมุติให้แอพพลิคชั่น SNMPMIBBrowser ทำการขอใช้บริการจากฟังก์ชัน MesgGetReq ของเมสเสจเมเนเจอร์ซึ่งไม่ตรงกับโปรแกรมของ SNMPMIBBrowser ที่ถูกสร้างขึ้น

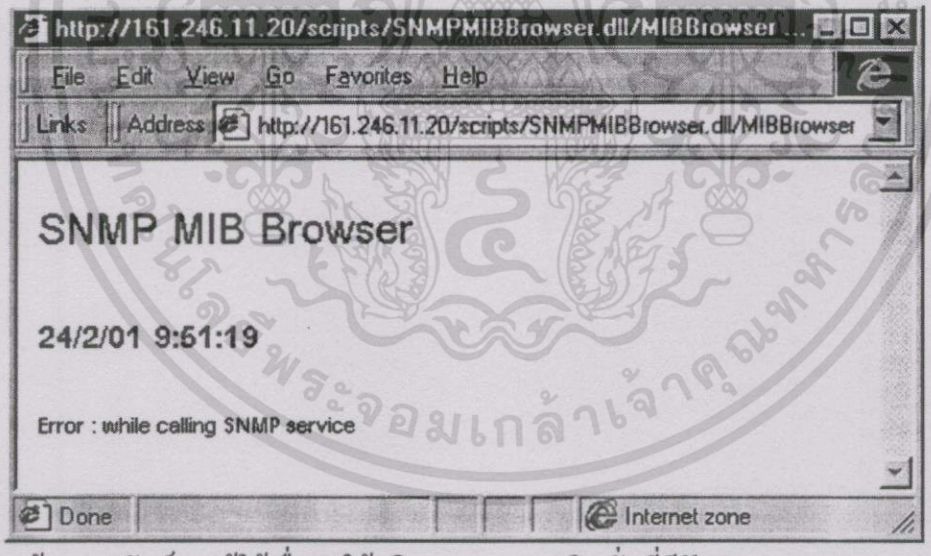
เมื่อพิจารณาผลการทดลองจะพบว่าคอนเนกเตอร์ได้อนุญาติให้ SNMPMIBBrowser ที่มีการขอใช้บริการที่บิดเบือนจากระบบผลลัพธ์สามารถรับบริการจากผู้ใช้ได้ พิจารณาจากรูปที่ 4.17 ไฮไลท์แสดงให้เห็นว่าคอนเนกเตอร์ได้อีเนเบิล SNMPMIBBrowser และเข้าใจว่าแอพพลิเคชั่นต้องการใช้บริการจากเซอร์วิสเมสเสจเมเนเจอร์ ดังนั้นเมื่อถึงขั้นตอนที่คอนเนกเตอร์ต้องส่งทิกเก็ตให้กับเซอร์วิสที่ SNMPMIBBrowser ต้องการใช้บริการ คอนเนกเตอร์ก็จะส่งให้กับเมสเสจเมเนเจอร์แทนดังรูปที่ 4.18 ทำให้ SNMPMIBBrowser ได้รับการปฏิเสธการให้บริการจากเน็ตเซอร์วิสเนื่องจากเน็ทเซอร์วิสไม่ได้รับทิกเก็ตจากคอนเนกเตอร์นั่นเอง (พิจารณาจากรูปที่ 4.19) ผลการทดลองแสดงให้เห็นว่าระบบสามารถควบคุมการขอใช้บริการเซอร์วิสจากแอพพลิเคชั่นได้



รูปที่ 4.17 หน้าจอของคอนเนกเตอร์เมื่อทำการตรวจสอบความสอดคล้องในการทำงานของ SNMPMIBBrowser



รูปที่ 4.18 หน้าจอของเมสเสจเมเนเจอร์หลังจากรับทิกเก็ตคีย์จากคอนเนกเตอร์



รูปที่ 4.19 หน้าจอผลลัพธ์ของผู้ใช้เมื่อขอใช้บริการจากแอพพลิเคชั่นที่มีปัญหา

ระดับเซอร์วิส

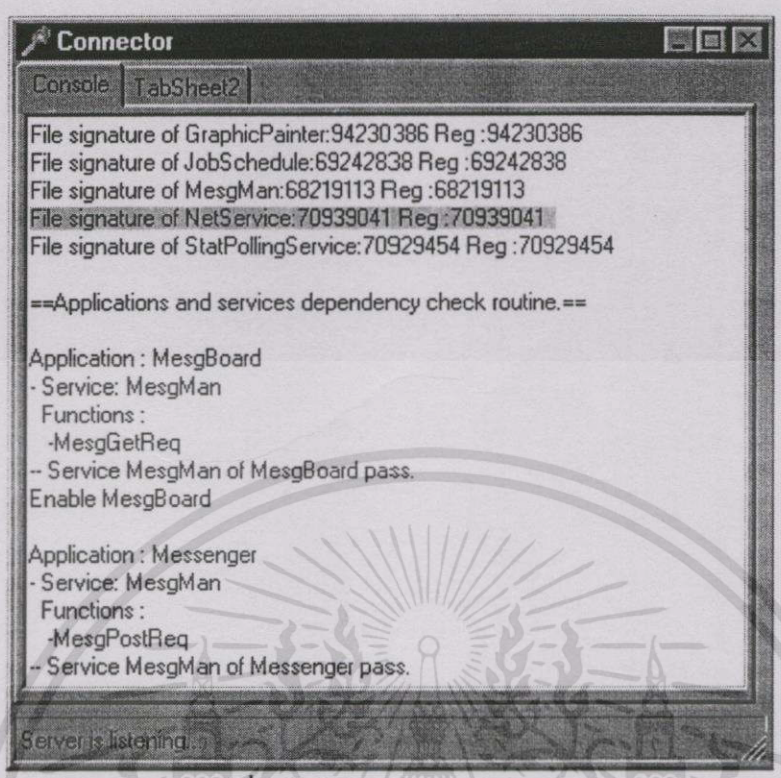
มีการตรวจสอบ 2 ขั้นคือ การตรวจสอบโมดูลปลอมและการตรวจสอบเซอร์วิสคีย์เมื่อมีการร้องขอใช้บริการจากเซอร์วิสด้วยกันเอง

1.เป็นการทดลองเพื่อแสดงว่าระบบสามารถตรวจสอบว่าเซอร์วิสที่ระบบกำลังจะสั่งให้ทำงานเป็นเซอร์วิสเดียวกับที่ได้ถูกติดตั้งหรือไม่ โดยสมมุติให้มีโมดูลเน็ตเซอร์วิส-1ซึ่งมีชื่อว่า

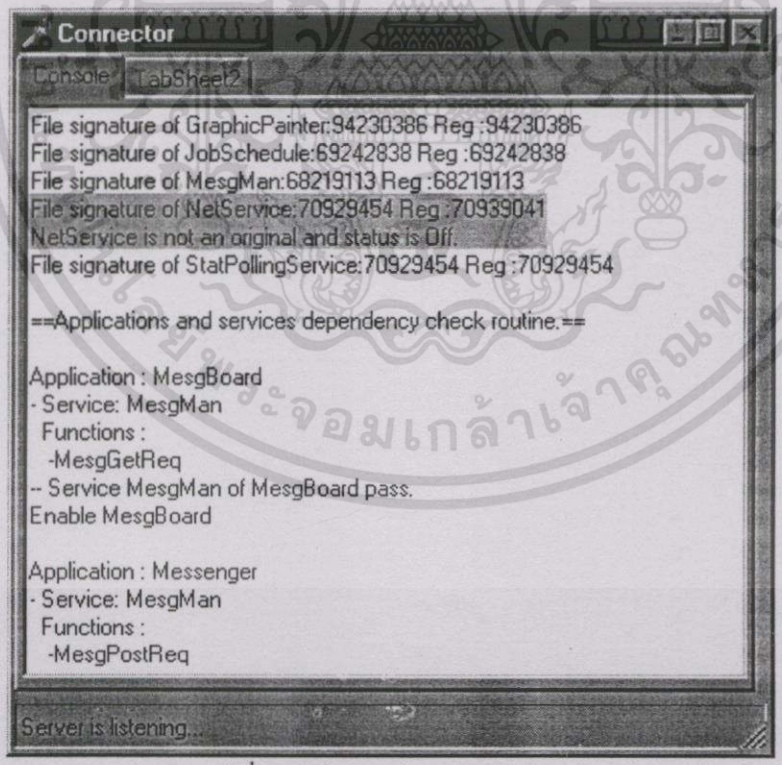
NetService.exe เป็นโมดูลที่ได้รับการติดตั้งเข้าสู่ระบบโดยมีค่าไฟล์ซิกเนเจอร์เท่ากับ 70939041 มีขนาดของไฟล์ 640 KB ในขณะที่เน็ตเซอร์วิส-2 เป็นไฟล์ที่แปลกปลอมที่ถูกสมมุติขึ้นมีขนาดของไฟล์ 640 KB เช่นกัน

การทดลองจากการชัทดาวน์ระบบและทำการแทนที่ไฟล์เน็ตเซอร์วิส-1ด้วยเน็ตเซอร์วิส-2 โดยใช้ชื่อเหมือนกันคือ NetService.exe เมื่อการสตาร์ทอัพระบบมาถึงขั้นตอนการสตาร์ทอัพเซอร์วิส จะพบว่าคอนเนกเตอร์ตรวจสอบค่าไฟลซิกเนเจอร์ของเน็ตเซอร์วิส-2ซึ่งมีค่าเท่ากับ 70929454 ซึ่งแตกต่างจากค่าไฟล์ซิกเนเจอร์เดิมที่ได้ติดตั้งไว้ ดังนั้นจึงคอนเนกเตอร์จึงรู้ได้ว่าโมดูลของเน็ตเซอร์วิสไม่ใช่โมดูลเดิมที่ได้ทำการติดตั้งไว้ตั้งแต่ตอนแรก คอนเนกเตอร์จึงไม่สตาร์ทอัพเน็ตเซอร์วิสซึ่ง เน็ตเซอร์วิส-1จะถูกสตาร์ทอัพขึ้นมาทำงานเป็นปกติ แต่เมื่อลองเอาโมดูลเน็ตเซอร์วิส-2 มาแทนที่โดยใช้ชื่อไฟล์ที่เหมือนกันคือ 'NetService.exe' และคงค่าไฟล์ซิกเนเจอร์เดิมไว้จะปรากฏว่าระบบจะไม่ทำการสตาร์ทอัพเน็ตเซอร์วิส-2 ขึ้นมาทำงาน ดังรูปที่ 4.20 และ 4.21 เป็นรูปแสดงคอนโซลของคอนเน็กเตอร์ขณะสตาร์ทอัพเซอร์วิสเมื่อใช้โมดูลของเน็ตเซอร์วิส-1 กับการสตาร์ทอัพขณะใช้โมดูลของเน็ตเซอร์วิส-2 คอนเนกเตอร์ตรวจพบว่าโมดูลเน็ตเซอร์วิส-2ที่มีอยู่บนระบบปัจจุบันเป็นคนละโมดูลกับเน็ตเซอร์วิส-1 ที่ได้ถูกติดตั้งไว้โดยค่าของไฟล์ซิกเนเจอร์ที่แตกต่างกัน และผลจากการที่เน็ตเซอร์วิสไม่สามารถทำงานได้นั้นแอพพลิเคชั่นที่เกี่ยวข้องเช่น SNMPMIBBrowser, NetworkDiscovery ซึ่งเรียกใช้บริการของเน็ตเซอร์วิสก็ไม่สามารถให้บริการแก่ผู้ใช้ได้ดังรูปที่ 4.22 และ 4.23 ซึ่งแสดงเมนูของ Admin ซึ่งมีระดับสิทธิเท่ากับ 0 ซึ่งสามารถใช้ทุกบริการบนระบบขณะที่ใช้เน็ตเซอร์วิส-1 และเน็ตเซอร์วิส-2 ซึ่งในรูปที่ 4.23 จะไม่ปรากฏรายการของ MIB Browser, Network Discovery และ Network Map ซึ่งเป็นอินเตอร์เฟสของแอพพลิเคชั่น SNMPMIBBrowser, NetworkDiscovery และ NetworkMap ปรากฏอยู่

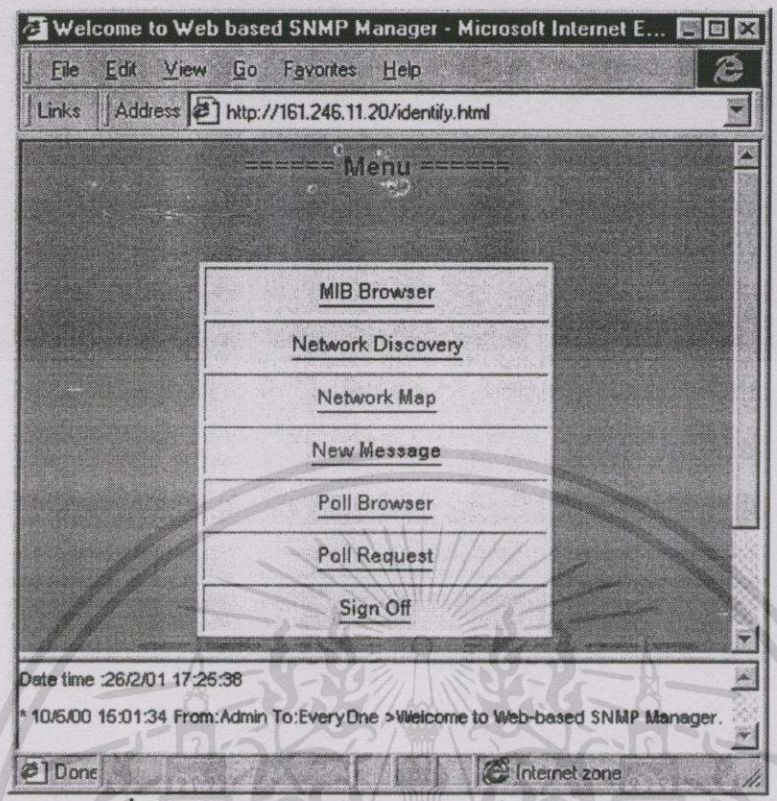
ดังนั้นผลการทดลองทำให้เห็นว่าระบบสามารถตรวจสอบโมดูลแปลกปลอมบนระบบได้ด้วยวิธีไฟล์ซิกเนเจอร์ และสามารถป้องกันแอพพลิเคชั่นในการใช้บริการจากเซอร์วิสที่ไม่ได้ทำงานบนระบบ



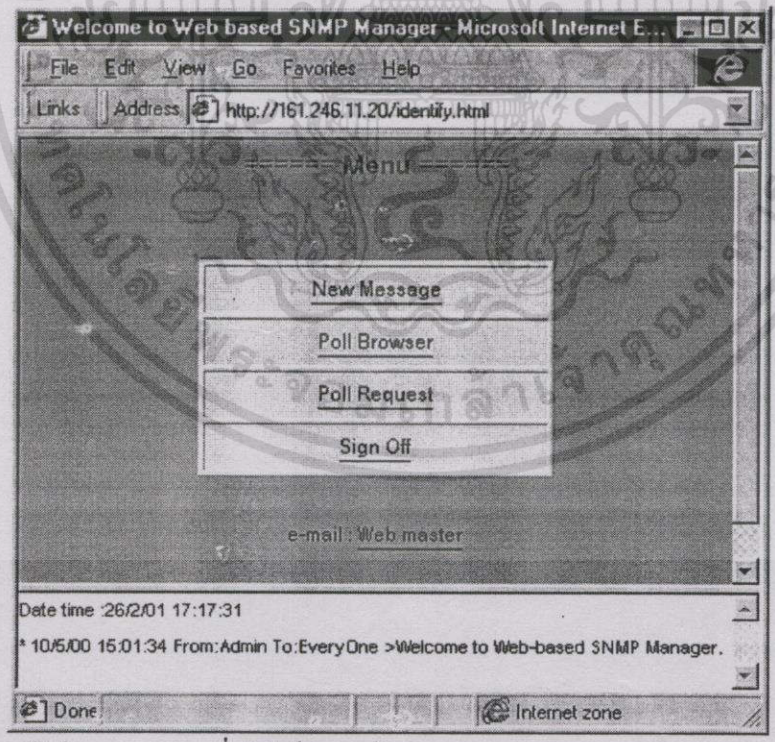
รูปที่ 4.20 คอนโซลของคอนเนกเตอร์เมื่อใช้เน็ตเซอร์วิส-1



รูปที่ 4.21 คอนโซลของคอนเนกเตอร์เมื่อใช้เน็ตเซอร์วิส-2



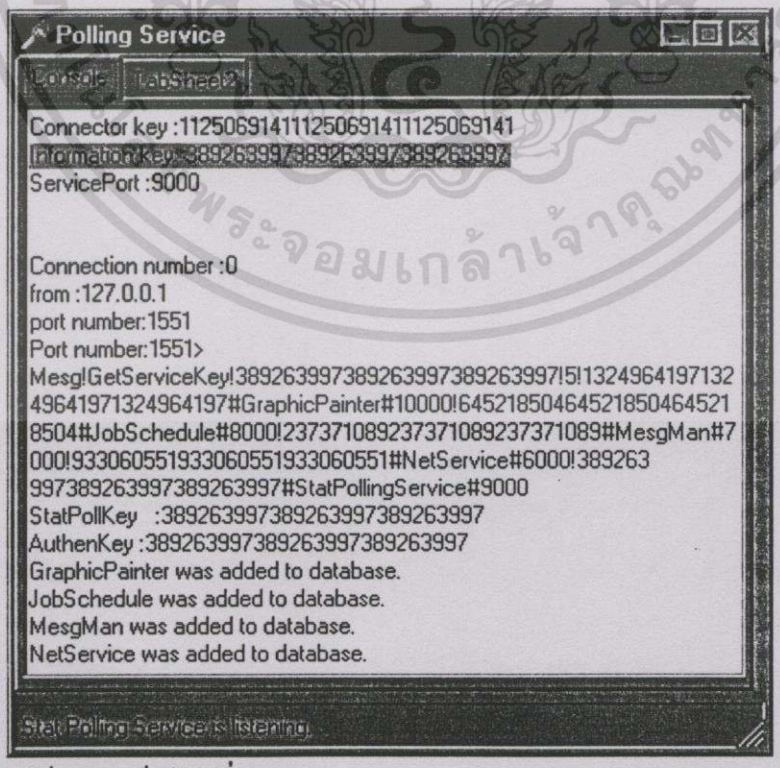
รูปที่ 4.22 เมนูของ Admin เมื่อใช้เน็ตเซอร์วิส-1



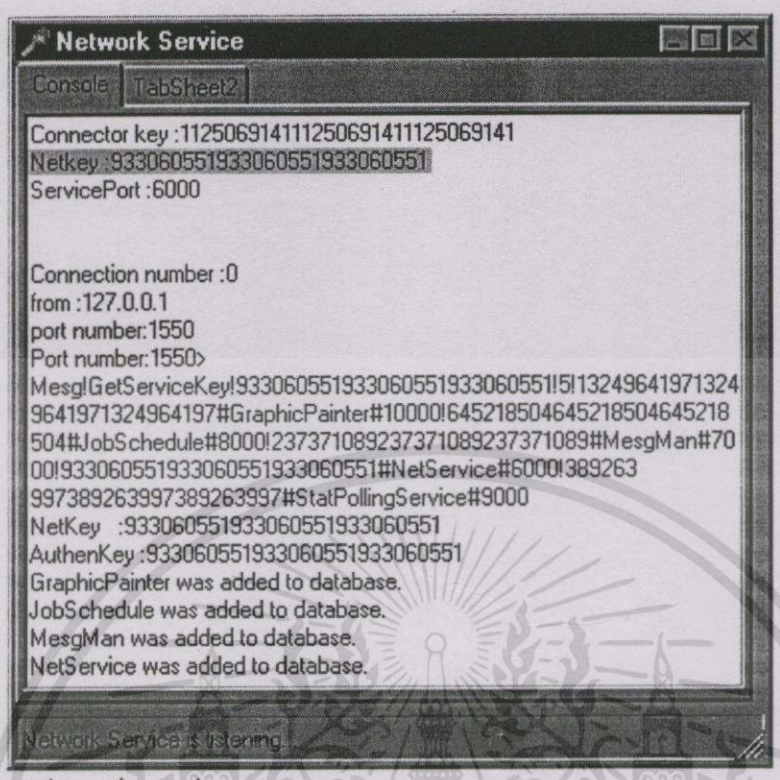
รูปที่ 4.23 ของเมนูผู้ใช้ของ Admin เมื่อใช้เน็ตเซอร์วิส-2

2.เป็นการทดลองเพื่อแสดงให้เห็นว่าระบบมีการตรวจสอบเพื่อรักษาความปลอดภัยสำหรับกรณีการติดต่อสื่อสารระหว่างเซอร์วิสด้วยกัน โดยใช้เหตุการณ์ที่เซอร์วิสสแต็ตโพลลิ่ง (StatPolling) ทำการขอใช้บริการฟังก์ชันชื่อ GetRequest ของเน็ตเซอร์วิส

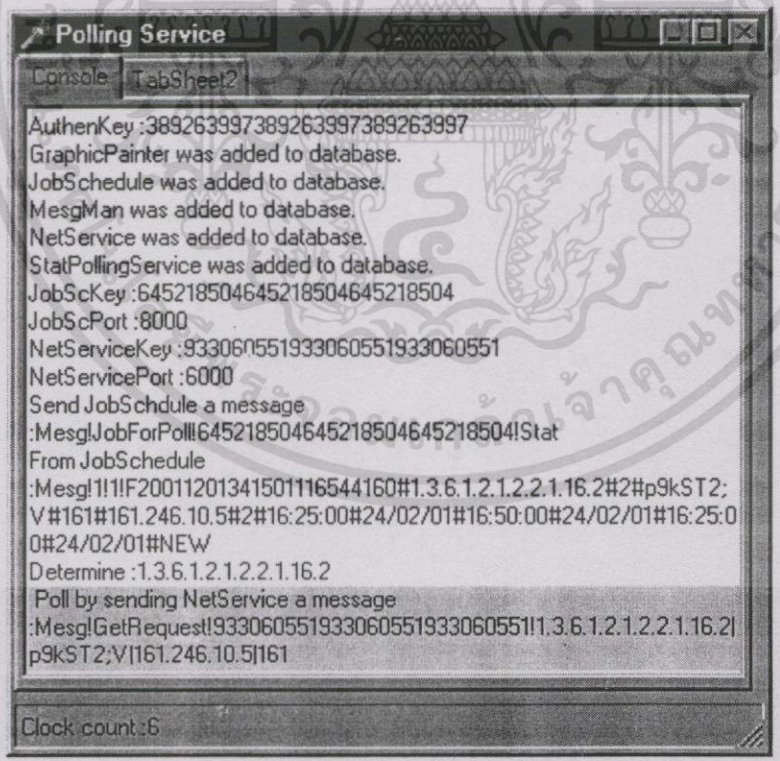
ผลการทดลองที่ได้คือ หลังจากเซอร์วิสสแต็ตโพลลิ่งถูกสตาร์ทอัพ เซอร์วิสจะได้รับเซอร์วิสคีย์ของเซอร์วิสทั้งหมดที่ถูกสตาร์ทอัพดังในรูปที่ 4.24 คอนเนเตอร์จะส่งเซอร์วิสคีย์ทั้งหมดให้กับสแต็ตโพลลิ่งโดยผ่านทางคำสั่ง GetServiceKey ของสแต็ตโพลลิ่ง ซึ่งหนึ่งในเซอร์วิสคีย์ทั้งหมดที่สแต็ตโพลลิ่งได้รับคือเน็ตเซอร์วิสคีย์ (พิจารณารูปที่ 4.25 ซึ่งแสดงเซอร์วิสคีย์ของเน็ตเซอร์วิส) เมื่อสแต็ตโพลลิ่งต้องการติดต่อกับเน็ตเซอร์วิส สแต็ตโพลลิ่งก็จะนำเน็ตเซอร์วิสคีย์มาใช้ในการอ้างอิงเพื่อขอบริการจากเน็ตเซอร์วิสดังรูปที่ 4.26 ซึ่งแสดงรีเควสท์ที่สแต็ตโพลลิ่งส่งไปยังเน็ตเซอร์วิส รูปที่ 4.27 แสดงรีเควสท์ที่เน็ตเซอร์วิสได้รับจากแสต็ตโพลลิ่งและคำตอบที่เน็ตเซอร์วิสส่งกลับไปหาแสต็ตโพลลิ่ง โดยรีเควสท์ดังกล่าวเป็นการขอใช้บริการจากฟังก์ชัน GetRequest ของเน็ตเซอร์วิสให้หาคำตอบของเลขหมายอ๊อปเจ็กท์ 1.3.6.1.2.1.2.2.1.16.2 จาก MIB ของอุปกรณ์เครือข่ายที่มีเลขหมายไอพีเท่ากับ 161.246.10.5 มีค่าคอมมิวนิวตี้เนมเป็น p9kST2;V และเน็ตเซอร์วิสสามารถใช้โปรโตคอล SNMP ติดต่อไปยังพอร์ทหมายเลข 161 และคำตอบที่เน็ตเซอร์วิสส่งกลับไปยังสแต็ตโพลลิ่งคือ 398953901 ดังนั้นจะพบว่าหากเซอร์วิสไม่ได้รับการสตาร์ทอัพจากคอนเนกเตอร์ เซอร์วิสจะไม่ทราบเซอร์วิสคีย์ที่จะใช้ในการอ้างอิงเพื่อขอบริการจากเซอร์วิสอื่นๆบนระบบได้ ขั้นตอนการทำงานในลักษณะนี้ช่วยให้ระบบเกิดความปลอดภัยมากขึ้น



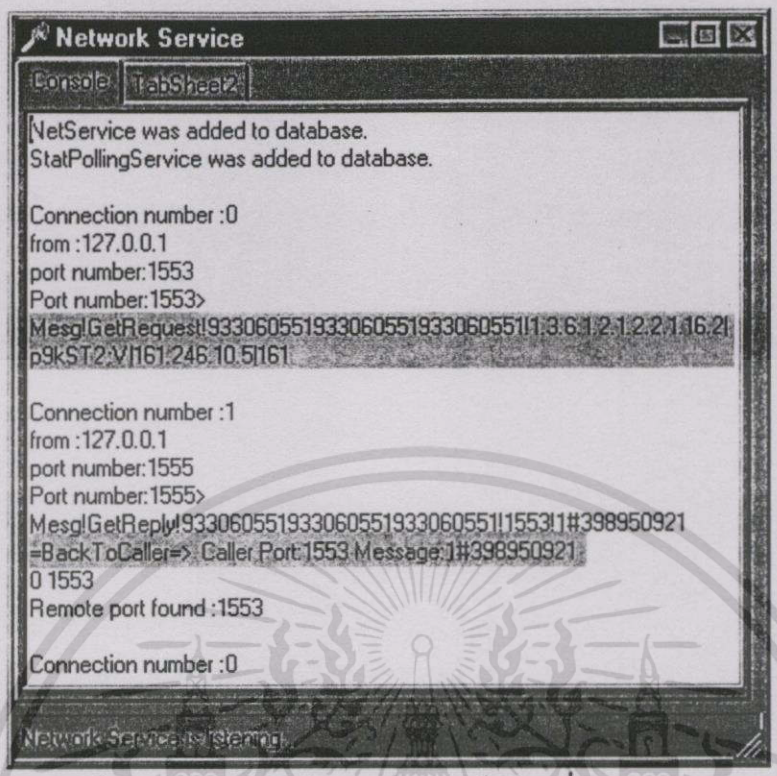
รูปที่ 4.24 เซอร์วิสคีย์ของแสต็ตโพลลิ่ง



รูปที่ 4.25 เซอร์วิสคีย์ของเน็ตเซอร์วิส



รูปที่ 4.26 คอนโซลของสแต็ตโพลลิ่งขณะขอส่งรีเควสท์ไปยังเน็ตเซอร์วิส



รูปที่ 4.27 คอนโซลของเน็ตเซอร์วิสขณะรับรีเควสท์จากสแต็ตโพลลิ่ง

4.4.5 การทำงานทั่วไปของระบบ

การทำงานทั่วไปของระบบดังที่ได้กล่าวถึงในหัวข้อ 3.5.2 เป็นภาพรวมของการทำงานของระบบซึ่งประกอบด้วยการติดต่อสื่อสารของส่วนต่างๆภายในระบบ ในการทดลองนี้จะแสดงให้เห็นถึงข้อมูลที่ถูกรับส่งระหว่างส่วนต่างๆของระบบ โดยจะสมมุติเหตุการณ์ UserA ซึ่งกำลังทำงานอยู่บนระบบขอใช้บริการ MIB Browser จากระบบ เพื่อขอข้อมูลของหมายเลขอ๊อปเจ็กท์ 1.3.6.1.2.1.1.5.0 จากอุปกรณ์เครือข่ายเลขหมายไอพี 161.246.11.20 ที่พอร์ท 161 โดยมีคอมมิวนิตี้เนมคือ public โดยใช้วิธี Single ของ MIB Browser ซึ่งเป็นการเรียกใช้คำสั่ง GetRequest ของโปรโตคอล SNMP ของเน็ตเซอร์วิส

พารามิเตอร์ของระบบที่มีเกี่ยวข้องกับการปฏิบัติการในการทดลองนี้คือ คอนเนกเตอร์คีย์ และ เน็ตเซอร์วิสคีย์ ซึ่งค่าของคีย์ทั้งสองคือ 123060239812306023981230602398 และ 121174497412117449741211744974 ตามลำดับพิจารณาจากรูปที่ 4.28

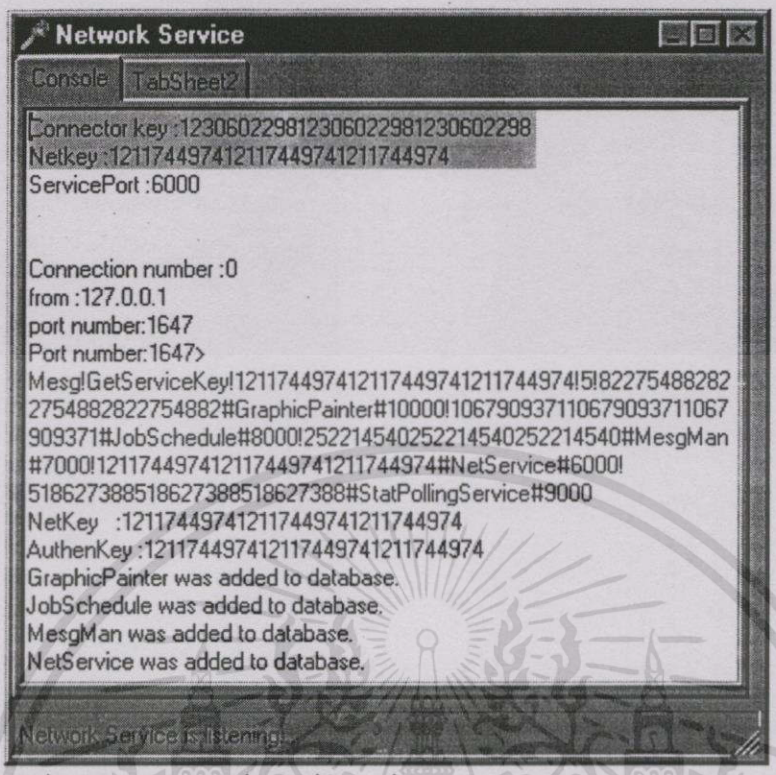
ต่อไปนี้เป็นการรับส่งข้อมูลที่เกิดขึ้นระหว่างการทำงานของระบบโดยเริ่มจากผู้ใช้ทำการเลือกรายการ MIB Browser ซึ่งเป็นอินเตอร์เฟสของแอพพลิเคชั่นSNMPMIBBrowser ดังรูปที่ 4.29 และป้อนข้อมูลและเลือกวิธีการซึ่งเป็นอินพุตสำหรับแอพพลิเคชั่นดังรูปที่ 4.30 หลังจากนั้น SNMPMIBBrowser จะใช้เซสชั่นคีย์สำหรับอ้างอิงเพื่อขอทิกเก็ตคีย์จากคอนเนกเตอร์ รีเควสท์ที่ SNMPMIBBrowser ส่งไปยังคอนเนกเตอร์เพื่อขอทิกเก็ตคีย์คือ “Mesg!Auth!SNMPMIBBrowser !161.246.11.20!24/2/01-2:59:33-940908864940908864 ” รีเควสท์นี้เป็นการขอใช้บริการจากฟังก์ชัน

Auth ของคอนเนกเตอร์ โดยแอพพลิเคชั่นที่ทำการขอใช้บริการคือ SNMPMIBBrowser ผู้ใช้มีเลขหมายไอพีคือ 161.246.11.20 และมีเซสชั่นคีย์เท่ากับ 24/2/01-2:59:33-9409088649 40908864

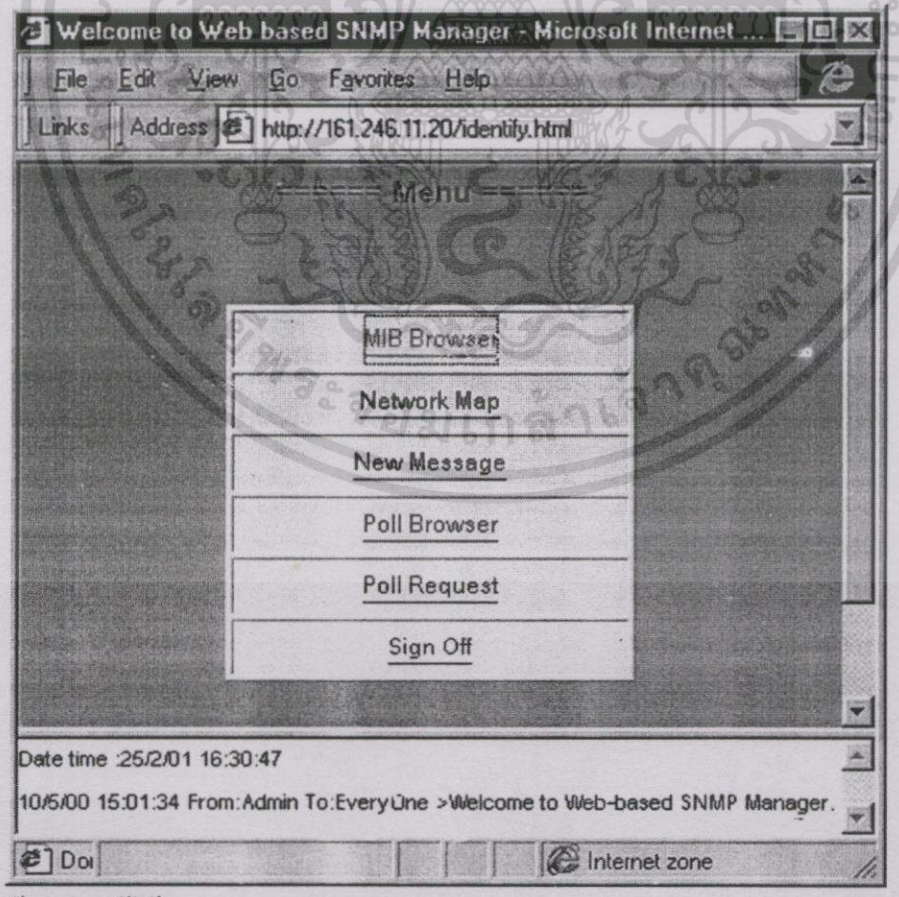
หลังจากคอนเนกเตอร์ตรวจสอบเซสชั่นคีย์แล้ว คอนเนกเตอร์จะส่งทิกเก็ตคีย์ให้กับเน็ตเซอร์วิสคีย์ด้วยการเรียกใช้ฟังก์ชัน InsertTicket ของเน็ตเซอร์วิส (พิจารณาจากไฮไลท์บนสุดในรูปที่ 4.32) "Mesg!InsertTicket!121174497412117449741211744974!127987458212798745821279874582!SNMPMIBBrowser!3#GetNextRequest#GetRequest#SetRequest" รีเควสท์ที่คอนเนกเตอร์ส่งให้กับเน็ตเซอร์วิสเป็นการเรียกใช้ฟังก์ชัน InsertTicket ของเน็ตเซอร์วิสโดยใช้เน็ตเซอร์วิสคีย์คือ 121174497412117449741211744974 สำหรับอ้างอิง ทิกเก็ตที่ส่งมาให้กับเน็ตเซอร์วิสมีค่าเท่ากับ 127987458212798745821279874 5 และฟังก์ชันของเน็ตเซอร์วิสที่แอพพลิเคชั่น SNMPMIBBrowser จะขอใช้บริการมี 3 ฟังก์ชันประกอบด้วย GetNextRequest, GetRequest และ SetRequest

หลังจากคอนเนกเตอร์ส่งทิกเก็ตคีย์ให้กับเน็ตเซอร์วิส คอนเนกเตอร์ก็จะส่งทิกเก็ตกับหมายเลขพอร์ทคือหมายเลข 6000 ของเน็ตเซอร์วิสเพื่อให้ SNMPMIBBrowser ใช้ในการติดต่อขอใช้บริการจากเน็ตเซอร์วิสโดยตรง "Mesg#1#127987458212798745821279874582#1!NetService!6000#UserA"

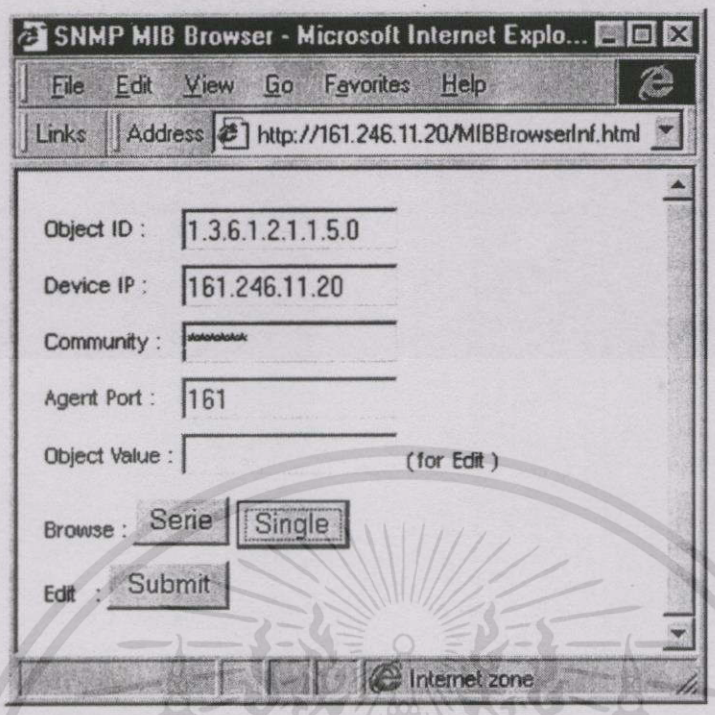
SNMPMIBBrowser ส่งรีเควสท์เพื่อขอใช้บริการ GetRequest จากเน็ตเซอร์วิสโดยใช้ทิกเก็ตคีย์สำหรับอ้างสิทธิ ดังในไฮไลท์ที่สองในรูปที่ 4.32 ซึ่งเป็นข้อความแสดงรีเควสท์ที่เน็ตเซอร์วิสได้รับจาก SNMPMIBBrowser "Mesg!GetRequest!127987458212798745821279874582!1.3.6.1.2.1.1.5.0|public|161.246.11.20|161" และข้อมูลที่เน็ตเซอร์วิสส่งกลับยัง SNMPMIBBrowser คือ 1#NETSKY ซึ่งเป็นคำตอบของหมายเลขอ๊อปเจ็กท์ 1.3.6.1.2.1.1.5.0 ของอุปกรณ์เครือข่ายที่มีเลขหมายไอพีเท่ากับ 161.246.11.20 ดังไฮไลท์ด้านล่างในรูปที่ 4.32 และ SNMPMIBBrowser ก็จะแสดงผลลัพธ์ที่ได้บนเวบเบราเซอร์ดังรูปที่ 4.31



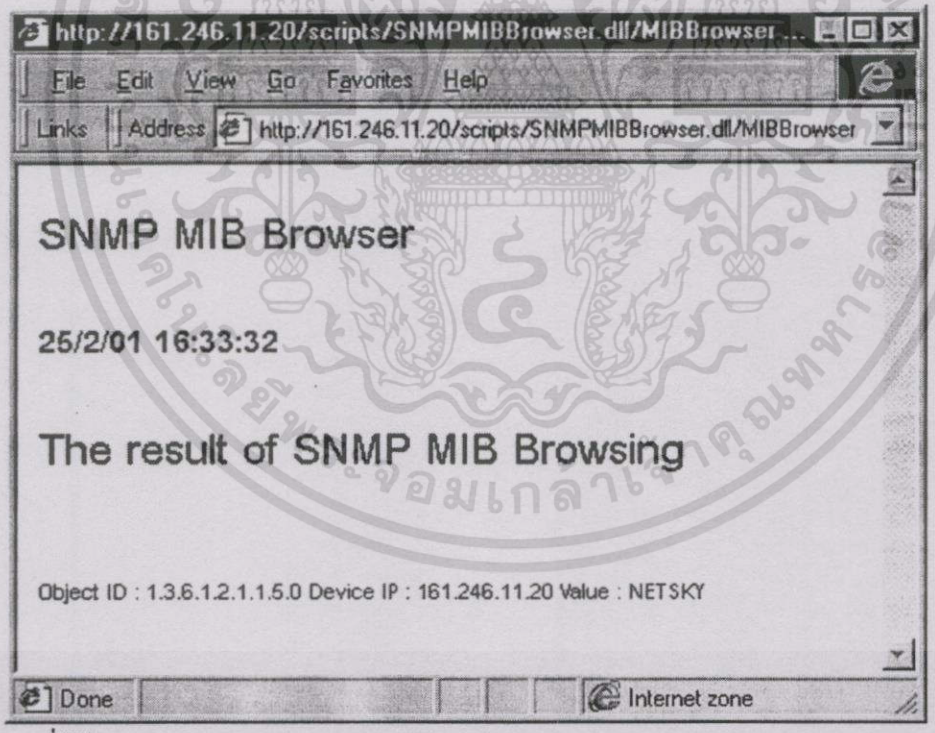
รูปที่ 4.28 เซอร์วิสคีย์ของคอนเนกเตอร์และเน็ตเซอร์วิส



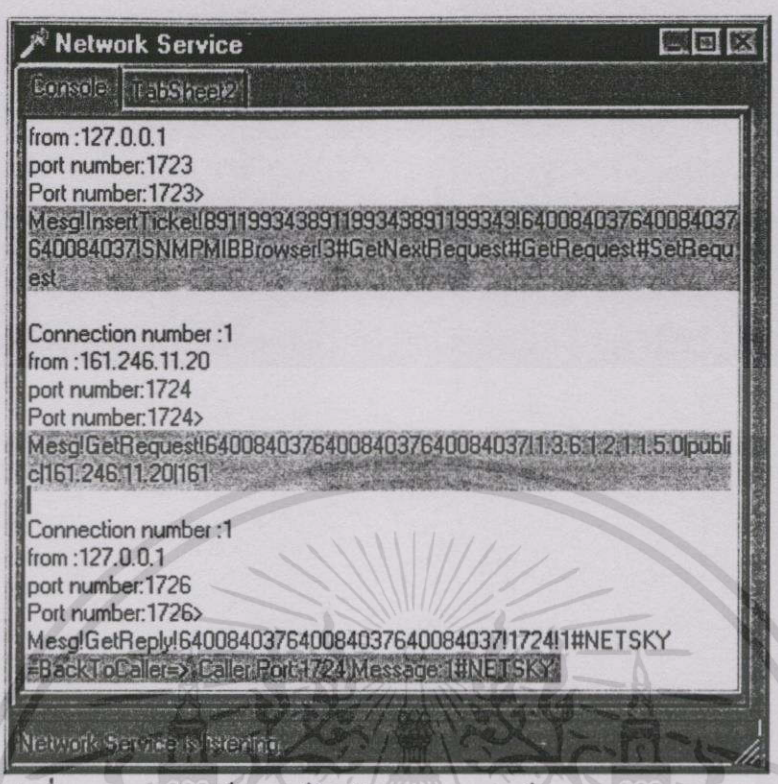
รูปที่ 4.29 หน้าจอเมนูผู้ใช้ของ UserA



รูปที่ 4.30 หน้าจอของ MIB Browser



รูปที่ 4.31 ผลที่ได้จากการทำงานของ MIB Browser



รูปที่ 4.32 การติดต่อสื่อสารระหว่างเน็ตเซอร์วิสกับคอนเนกเตอร์และ SNMPMIBBrowser

4.5 การวิเคราะห์ผลการทดลอง

ในการติดตั้งและถอดถอนโมดูลของเซอร์วิส NetService และแอพพลิเคชั่น SNMPMIBBrowser ในหัวข้อ 4.3.1 4.3.2 และ 4.3.3 แสดงให้เห็นว่าข้อมูลของเซอร์วิสที่มีความสำคัญต่อการให้บริการแก่แอพพลิเคชั่นคือ ระดับสิทธิของฟังก์ชันและเวอร์ชันของเซอร์วิส โดยสิทธิที่มีระดับต่ำสุดของฟังก์ชันในเซอร์วิสที่แอพพลิเคชั่นขอใช้บริการจะเป็นตัวกำหนดสิทธิของแอพพลิเคชั่นและสิทธิของผู้ใช้ ดังตัวอย่างที่เห็นจากการทดลอง SNMPMIBBrowser มีการขอใช้บริการจาก NetService ทั้งหมด 3 ฟังก์ชัน ได้แก่ GetRequest, GetNextRequest และ SetRequest ซึ่งแต่ละฟังก์ชันมีระดับสิทธิเท่ากับ 100 ทำให้สิทธิที่ต่ำสุดที่จะนำมาเป็นสิทธิของแอพพลิเคชั่นมีค่าเท่ากับ 100 นั่นแสดงว่าผู้ใช้ที่ต้องการใช้บริการแอพพลิเคชั่น SNMPMIBBrowser ต้องมีระดับสิทธิไม่ต่ำกว่า 100 พารามิเตอร์อีกตัวหนึ่งที่มีความสำคัญเช่นกันคือ เวอร์ชันของเซอร์วิส เนื่องจากระบบมีลักษณะเป็นกลุ่มของโมดูลที่ทำงานร่วมกันทำให้มีอิสระในการพัฒนาโมดูล แต่โมดูลที่จะทำงานด้วยกันได้ก็ต้องอยู่ภายใต้ข้อตกลงเดียวกันจึงจำเป็นต้องใช้เวอร์ชันของเซอร์วิสเพื่อบอกถึงรุ่นของโมดูล สำหรับข้อมูลของแอพพลิเคชั่นที่มีความสำคัญต่อการขอใช้บริการจากเซอร์วิสก็คือ ServiceInfo โดยระบบจะทราบหมายเลขเซอร์วิสพอร์ทที่แอพพลิเคชั่นต้องการจากรายชื่อเซอร์วิสที่อยู่ในพารามิเตอร์ตัวนี้ และจะทำการส่งหมายเลขทั้งหมดให้กับแอพพลิเคชั่นหลังจากที่แอพพลิเคชั่นผ่านการตรวจสอบสิทธิเรียบร้อยแล้ว ดังนั้น การกำหนดระดับสิทธิให้กับบริการต่างๆเพื่อใช้ควบคุมผู้ใช้จึงมีประโยชน์สำหรับระบบที่สนับสนุนการทำงานแบบหลายผู้ใช้ การกำหนดรายชื่อของฟังก์ชันของเซอร์วิสที่แอพพลิเคชั่นต้องการใช้บริการจะ

เหมาะสมกับระบบที่มีโครงสร้างแบบเลเยอร์เพราะทำให้เลเยอร์ที่ให้บริการสามารถควบคุมการให้บริการแก่เลเยอร์ที่มาขอใช้บริการได้

สำหรับการรักษาความปลอดภัยของระบบซึ่งแบ่งออกเป็น 3 ระดับคือระดับผู้ใช้ ระดับแอพพลิเคชั่น และระดับเซอร์วิส จากการทดลองจะพบว่าการรักษาความปลอดภัยในระดับผู้ใช้พารามิเตอร์สำคัญก็คือเซสชั่นคีย์กับระดับสิทธิของผู้ใช้ โดยผู้ใช้ที่ขอใช้บริการจากระบบด้วยการเรียกใช้แอพพลิเคชั่นผ่านทาง URL ของเวบเบราเซอร์โดยตรงจะไม่สามารถเข้าใช้บริการจากระบบได้ทั้งนี้เนื่องจากว่าเวบเบราเซอร์ไม่ได้ส่งเซสชั่นคีย์ให้กับระบบนั่นเอง ดังนั้นเมื่อแอพพลิเคชั่นที่ถูกเรียกโดยตรงทำการติดต่อกับคอนเนกเตอร์เพื่อขอตรวจสอบสิทธิของแอพพลิเคชั่น แอพพลิเคชั่นจะไม่มีเซสชั่นคีย์ให้กับคอนเนกเตอร์ทำให้แอพพลิเคชั่นถูกปฏิเสธการให้บริการจากระบบ ในขณะที่สถานการณ์ถัดมาที่ผู้ใช้ทำการล็อกอินเข้าสู่ระบบอย่างถูกต้องทำให้เวบเบราเซอร์ของผู้ใช้ได้รับเซสชั่นคีย์จากระบบและพร้อมที่จะขอใช้บริการอื่นๆจากรายการต่างๆในเมนูผู้ใช้ แต่ผู้ใช้กลับขอใช้บริการจากแอพพลิเคชั่นอื่นที่ไม่อยู่ในเมนูและมีระดับสิทธิที่สูงกว่าสิทธิของผู้ใช้โดยผ่านทาง URL ของเวบเบราเซอร์เหมือนกับกรณีก่อนหน้านี้ สำหรับกรณีนี้เมื่อระบบตรวจสอบพบว่าเซสชั่นคีย์ของผู้ใช้ถูกต้องสิ่งที่จะตรวจสอบถัดมาก็คือระดับสิทธิของผู้ใช้ถึงแม้ว่าจะมีการจัดรายการของบริการที่ผู้ใช้มีสิทธิใช้งานไว้บนเมนูผู้ใช้แล้วก็ตามและจะปฏิเสธการให้บริการถ้าหากพบว่าระดับสิทธิของผู้ใช้ต่ำกว่าสิทธิของแอพพลิเคชั่นที่ผู้ใช้ต้องการขอใช้บริการ ทั้งนี้เพื่อป้องกันเหตุการณ์ที่จะเกิดขึ้นเช่นเดียวกับกรณีนี้นั่นเอง ในกรณีสุดท้ายของการรักษาความปลอดภัยในระดับผู้ใช้เป็นกรณีที่เป็นผลพวงมาจากการที่ผู้ใช้ยุติการใช้งานโดยไม่ได้ล็อกเอ๊าท์ออกจากระบบทำให้ระบบไม่รับทราบว่าผู้ใช้ได้ออกจากระบบไปแล้ว ดังนั้นเซสชั่นคีย์ของผู้ใช้จึงยังคงมีอยู่ในระบบ ในกรณีนี้ทำให้ผู้ใช้คนอื่นที่มาใช้เวบเบราเซอร์ลำดับถัดมาสามารถใช้เวบเบราเซอร์ซึ่งมีเซสชั่นคีย์ของผู้ใช้คนก่อนมาใช้ในการขอใช้บริการจากระบบได้ ทำให้ต้องมีการตรวจสอบระยะเวลาที่ผู้ใช้มีปฏิสัมพันธ์กับระบบหรือที่เรียกว่าไทม์เอ๊าท์ โดยระบบถูกตั้งเวลาไทม์เอ๊าท์ไว้ที่ 5 นาที และจากการทดลองทำให้เห็นว่าวิธีการนี้สามารถช่วยลดปัญหาที่เกิดขึ้นจากการออกจากระบบโดยไม่ล็อกเอ๊าท์ของผู้ใช้ได้ การรักษาความปลอดภัยในระดับถัดมาคือระดับแอพพลิเคชั่นซึ่งจะเป็นลักษณะของการควบคุมการใช้บริการของแอพพลิเคชั่นและการตรวจสอบสิทธิโดยระบบ ระบบจะควบคุมการใช้บริการของแอพพลิเคชั่นจากพารามิเตอร์ ServiceInfo ดังที่ได้กล่าวมาบ้างแล้วในย่อหน้าแรกของหัวข้อนี้รวมทั้งทิกเก็ตคีย์เป็นตัวยืนยันสิทธิของแอพพลิเคชั่นหลังผ่านการตรวจสอบสิทธิแล้ว โดยเมื่อพิจารณาจากผลการทดลองซึ่งผู้ใช้ไม่สามารถขอให้แอพพลิเคชั่นที่มีการลงทะเบียนขอใช้ฟังก์ชันจากเซอร์วิสไม่ตรงกับฟังก์ชันของเซอร์วิสที่แอพพลิเคชั่นได้ขอใช้บริการจริงเกิดจากกระบวนการของระบบ 2 กระบวนการคือ กระบวนแจกจ่ายทิกเก็ตคีย์ไปยังเซอร์วิสต่างๆโดยใช้ข้อมูลจาก ServiceInfo และกระบวนการตรวจสอบทิกเก็ตคีย์ของเซอร์วิสเมื่อแอพพลิเคชั่นติดต่อกับเซอร์วิสที่ให้บริการ โดยกระบวนการแรกคอนเนกเตอร์ซึ่งเป็นผู้ทำหน้าที่แจกจ่ายทิกเก็ตคีย์ไปยังเซอร์วิสต่างๆ ได้ส่งทิกเก็ตคีย์ไปยังเซอร์วิสที่อยู่ใน ServiceInfo ทำให้เซอร์วิสที่แอพพลิเคชั่นต้องการใช้บริการจริงๆไม่ได้รับ Ticket

ต่อมาในกระบวนการหลังซึ่งเป็นปกติอยู่แล้วที่เซอร์วิสทุกเซอร์วิสก่อนที่จะมีการให้บริการต่อคำร้องขอที่ได้รับเข้าก็ต้องมีการตรวจสอบความถูกต้องของทิกเก็ตคีย์และชื่อของฟังก์ชันที่ต้องให้บริการก่อนเสมอ ดังนั้นจากกระบวนการแรกที่เซอร์วิสที่แท้จริงไม่ได้รับทิกเก็ตหรือได้รับแต่ชื่อฟังก์ชันไม่ตรงกับที่คอนเนกเตอร์แจ้งมา เซอร์วิสก็จะปฏิเสธที่จะให้บริการต่อคำร้องขอนี้ ซึ่งเป็นผลให้ผู้ใช้ไม่สามารถใช้บริการจากระบบได้ ดังนั้นในระดับแอพพลิเคชั่นระบบจึงสามารถควบคุมการทำงานของแอพพลิเคชั่นได้ ระดับสุดท้ายคือระดับเซอร์วิสซึ่งมีการรักษาความปลอดภัย 2 ขั้นตอน โดยขั้นตอนแรกเกิดขึ้นขณะที่เซอร์วิสถูกสตาร์ทอัพเป็นการตรวจสอบไฟล์ซิกเนเจอร์ของเซอร์วิส ส่วนขั้นตอนหลังจะเกิดขึ้นเมื่อเซอร์วิสติดต่อเพื่อขอใช้บริการซึ่งกันและกัน จากการทดลองในขั้นตอนแรกนั้นสามารถตรวจสอบได้ว่าเซอร์วิสที่ถูกติดตั้งกับเซอร์วิสที่มีอยู่ในขณะนั้นเป็นโมดูลที่แตกต่างกัน สาเหตุอาจจะเนื่องจากมีการแก้ไขเซอร์วิสแต่ไม่ได้ทำการแก้ไขค่าไฟล์ซิกเนเจอร์ให้ตรงกับโมดูลใหม่ หรืออาจจะมีการเข้าแทรกแซงระบบจากภายนอกด้วยการเปลี่ยนโมดูลของเซอร์วิส ขั้นตอนถัดมาคือการรักษาความปลอดภัยเมื่อเซอร์วิสขอใช้บริการจากเซอร์วิสด้วยกันเอง จากการทดลองจะพบว่าขั้นตอนการทำงานจะน้อยกว่าเมื่อแอพพลิเคชั่นขอใช้บริการจากเซอร์วิสทั้งนี้เนื่องจากเซอร์วิสแต่ละเซอร์วิสก่อนที่จะถูกสตาร์ทอัพให้เริ่มทำงานได้มีการตรวจสอบมาจากขั้นตอนแรกแล้วนั่นเองจึงทำให้ไม่จำเป็นที่จะต้องตรวจสอบอีก และเพื่อลดขั้นตอนของการติดต่อระหว่างโปรเซสจึงได้มีการแจกจ่ายเซอร์วิสโค๊ดของแต่ละเซอร์วิสให้กับเซอร์วิสทุกเซอร์วิสที่ทำงานอยู่บนระบบทำให้เซอร์วิสสามารถใช้เซอร์วิสโค๊ดของเซอร์วิสที่ต้องการติดต่อด้วยในการอ้างอิงเพื่อขอใช้บริการได้ทันที จากการทดลองเรื่องการรักษาความปลอดภัยที่ผ่านมานั้นพบว่าแต่ละเลเยอร์ของระบบสามารถดูแลควบคุมเลเยอร์ที่มาขอใช้บริการได้ทำให้เกิดความปลอดภัยและเสถียรภาพในการทำงานของระบบขึ้น

สำหรับการทดลองในหัวข้อสุดท้ายคือหัวข้อที่ 4.3.5 เป็นการทดลองเกี่ยวกับการทำงานทั่วไปของระบบซึ่งทำให้เห็นภาพรวมของการทำงานของระบบโดยเฉพาะการรับส่งข้อมูลระหว่างโปรเซสของโมดูลต่างๆบนระบบ ขั้นตอนการทำงานจะทำให้เห็นถึงความสัมพันธ์ระหว่างโมดูลในเลเยอร์แอพพลิเคชั่นและเซอร์วิส การทำงานโดยใช้โปรโตคอล SNMP และการรักษาความปลอดภัยของระบบ จากการทดลองพบว่าตั้งแต่ผู้ใช้เริ่มเข้าสู่ระบบจนกระทั่งขอใช้บริการจากระบบเซอร์วิสที่สำคัญที่สุดเพราะเกี่ยวข้องกับการทำงานของระบบมากที่สุดคือคอนเนกเตอร์ซึ่งทำหน้าที่ตั้งแต่ตรวจสอบสิทธิของผู้ใช้ ตรวจสอบสิทธิของแอพพลิเคชั่น ให้ข้อมูลของเมนูผู้ใช้ และแจกจ่ายทิกเก็ตคีย์ให้แก่เซอร์วิสที่ถูกแอพพลิเคชั่นร้องขอเพื่อใช้บริการ เมื่อสังเกตุจะพบว่าการทำงานของระบบที่ใช้เซสชั่นคีย์จะเป็นส่วนระหว่างผู้ใช้กับแอพพลิเคชั่นเท่านั้นและทิกเก็ตคีย์ก็จะเป็นการเกี่ยวข้องกันระหว่างแอพพลิเคชั่นกับเซอร์วิส โดยผู้ใช้จะไม่มีวันรู้ข้อมูลเกี่ยวกับทิกเก็ตคีย์และเช่นกันเซอร์วิสอื่นๆยกเว้นคอนเน็กเตอร์ก็จะไม่มีทางรู้ข้อมูลของเซสชั่นคีย์ ทำให้มองเห็นได้ว่าการทำงานของแต่ละเลเยอร์เป็นอิสระต่อกัน

4.6 สรุป

การทดลองนี้แสดงให้เห็นว่า ด้วยหลักการและวิธีการทำงานของระบบทำให้ระบบสามารถทำงานตามวัตถุประสงค์ที่ตั้งไว้ได้คือ ระบบสามารถเป็นตัวกลางในการติดต่อกับอุปกรณ์เครือข่ายด้วยโปรโตคอล SNMP และผู้ใช้ได้ ทำให้สามารถควบคุมการเข้าถึงอุปกรณ์เครือข่ายด้วยโปรโตคอล SNMP โดยตรงได้ ระบบมีการรักษาความปลอดภัยทั้งในระบบของผู้ใช้และโมดูลที่ทำงานอยู่บนระบบ และระบบสนับสนุนการปรับเปลี่ยนโมดูลได้โดยไม่กระทบต่อโมดูลอื่นที่ไม่เกี่ยวข้อง

บทที่ 5
สรุปการวิจัยและข้อเสนอแนะ

5.1 สรุปการวิจัยของโปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบ

โปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบเป็นระบบต้นแบบซึ่งมีองค์ประกอบหลักคือ SNMP เซิร์ฟเวอร์เพื่อใช้โปรโตคอล SNMP ในการติดต่อกับอุปกรณ์เครือข่าย มีเวบเซิร์ฟเวอร์เพื่อสนับสนุนให้ผู้ใช้สามารถติดต่อใช้งานระบบผ่านทางระบบเวบ และมีออเทนทิเคชั่นเซิร์ฟเวอร์เพื่อใช้ตรวจสอบและควบคุมการทำงานของระบบ องค์ประกอบเหล่านี้จะถูกจัดเป็นเซอร์วิสซึ่งเป็นส่วนหนึ่งของระบบ โดยระบบจะมีโครงสร้างเป็นแบบเลเยอร์ 3 เลเยอร์คือแอพพลิเคชั่น เซอร์วิสและไลบารี่เลเยอร์ ผู้ดูแลระบบสามารถปรับเปลี่ยนโมดูลที่ต้องการในแต่ละเลเยอร์ได้โดยไม่กระทบต่อโมดูลอื่นที่ไม่เกี่ยวข้องตราบใดที่โมดูลดังกล่าวไม่ได้เปลี่ยนแปลงจุดเชื่อมต่อระหว่างเลเยอร์ ความแตกต่างระหว่างเลเยอร์แอพพลิเคชั่นและเลเยอร์เซอร์วิสคือลักษณะการทำงานของโปรเซสของโมดูลในเลเยอร์แอพพลิเคชั่นจะทำงานเมื่อได้รับคำสั่งจากเวบเซิร์ฟเวอร์ซึ่งเป็นรีเควสท์ที่เวบเซิร์ฟเวอร์ได้รับมาจากเวบเบราเซอร์ของผู้ใช้ เมื่อโปรเซสเสร็จสิ้นการทำงาน โปรเซสก็จะสลายตัวไป ดังนั้นโปรเซส 1 โปรเซสของโมดูลในเลเยอร์แอพพลิเคชั่นจะทำงานต่อรีเควสท์ของผู้ใช้ 1 รีเควสท์ ซึ่งแตกต่างจากโปรเซสการทำงานของโมดูลในเลเยอร์เซอร์วิสซึ่งจะประพฤติตัวเป็นเซิร์ฟเวอร์โปรเซสคือจะเริ่มทำงานพร้อมกับระบบและคอยรอรับรีเควสท์ที่จะมีเข้ามายังพอร์ทประจำของเซอร์วิส โปรเซสของเซอร์วิสจะสลายตัวไปเมื่อได้รับคำสั่งจากคอนเนกเตอร์เมื่อต้องการชัทดาวน์ระบบ ดังนั้นโปรเซสของเซอร์วิส 1 โปรเซสสามารถให้บริการได้ไม่จำกัดจำนวนรีเควสท์ ระบบจะทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้และอุปกรณ์เครือข่าย โดยโมดูลที่ให้บริการจะอยู่ในเซอร์วิสเลเยอร์ซึ่งจะมีหน้าที่ให้บริการการติดต่อกับอุปกรณ์เครือข่ายด้วยโปรโตคอล SNMP การนำระบบไปใช้งานจะต้องเริ่มจากการติดตั้งเซอร์วิสและแอพพลิเคชั่นให้กับระบบตามลำดับทั้งนี้เนื่องแอพพลิเคชั่นมีการเรียกใช้บริการจากเซอร์วิส ดังนั้นจึงควรติดตั้งเซอร์วิสก่อนเพื่อสามารถตรวจสอบบริการต่างๆของเซอร์วิสได้ เมื่อทำการสตาร์ทอัพระบบ เซอร์วิสต่างๆที่มีอยู่บนระบบจะถูกคอนเนกเตอร์ซึ่งเป็นเซอร์วิสตัวหนึ่งในระบบสั่งให้ระบบปฏิบัติการสร้างเซอร์วิสขึ้นมาทำงานในลักษณะของเซิร์ฟเวอร์โปรเซสซึ่งจะทำงานกับระบบจนกระทั่งระบบถูกชัทดาวน์หรือถูกผู้ดูแลระบบสั่งให้ยุติการทำงาน เซอร์วิสที่สามารถเป็นเซิร์ฟเวอร์โปรเซสจะต้องผ่านการตรวจสอบตัวจริงด้วยวิธีไฟล์ซิกเนเจอร์และได้รับเซอร์วิสโค๊ดของเซอร์วิสอื่นๆเพื่อใช้ในการอ้างอิงเมื่อต้องมีการติดต่อระหว่างเซอร์วิสด้วยกันในภายหลัง

สำหรับการทำงานทั่วไปของผู้ใช้ ผู้ใช้จะเริ่มติดต่อและขอใช้บริการจากระบบได้จากการใช้เวบเบราเซอร์ติดต่อมายังเวบเซิร์ฟเวอร์ซึ่งเป็นส่วนหนึ่งของระบบและเวบเซิร์ฟเวอร์จะทำการแปลงคำร้องขอของผู้ใช้ที่ได้รับจากเวบเบราเซอร์และส่งผ่านทางมาตรฐาน ISAPI ไปยังแอพพลิเคชั่นซึ่งมี

ลักษณะการทำงานเป็นแบบเวบเซิร์ฟเวอร์แอพพลิเคชั่นคือเวบเซิร์ฟเวอร์จะขอให้ระบบปฏิบัติการสร้างโปรเซสของแอพพลิเคชั่นขึ้นมาเมื่อได้รับคำร้องขอที่ต้องการใช้แอพพลิเคชั่นจากเวบเบราเซอร์ ต่อจากนั้นแอพพลิเคชั่นก็จะเริ่มทำงานตามหน้าที่ที่ได้รับมอบหมายด้วยคำร้องขอที่มาจากเวบเบราเซอร์ แอพพลิเคชั่น UserSignOn ซึ่งมีหน้าที่จัดการเรื่องการล็อกอินเข้าสู่ระบบของผู้ใช้จะเป็นแอพพลิเคชั่นแรกที่ผู้ใช้ต้องขอใช้บริการเพื่อแลกกับเซสชั่นคีย์และเมนูของผู้ใช้ ขณะเริ่มทำงานแอพพลิเคชั่นส่วนใหญ่จะร้องขอให้ระบบทำการตรวจสอบสิทธิของผู้ใช้จากเซสชั่นคีย์ที่ได้รับมาจากเวบเบราเซอร์เพื่อขอทิกเก็ตสำหรับนำไปใช้อ้างอิงในการขอรับบริการจากเซอร์วิสที่ต้องการ ระบบจะควบคุมการขอใช้บริการของแอพพลิเคชั่นด้วยการส่งรายชื่อฟังก์ชันไปยังเซอร์วิสที่แอพพลิเคชั่นต้องการใช้บริการเพื่อให้เซอร์วิสใช้รายชื่อตรวจสอบฟังก์ชันจากรีเควสท์ของแอพพลิเคชั่น ขั้นตอนดังกล่าวสามารถควบคุมให้แอพพลิเคชั่นใช้บริการจากระบบตรงตามที่ถูกระบุไว้ตั้งแต่ติดตั้งแอพพลิเคชั่น และหลังจากที่แอพพลิเคชั่นได้รับทิกเก็ตและขอใช้บริการจากเซอร์วิสตลอดจนทำงานตามหน้าที่จนเสร็จสิ้นแล้ว แอพพลิเคชั่นจะส่งผลลัพธ์ที่ได้จากการทำงานย้อนกลับไปยังเวบเบราเซอร์ด้วยเซสชั่นเดียวกับเซสชั่นที่เวบเบราเซอร์ใช้ติดต่อกับเวบเซิร์ฟเวอร์แล้วแอพพลิเคชั่นจึงจะสลายตัวไปจากระบบ และนอกจากไฟล์ซิกเนเจอร์ เซอร์วิสโค๊ด ระดับสิทธิ เซสชั่นคีย์และทิกเก็ตซึ่งเป็นเครื่องมือที่ถูกใช้ในการรักษาความปลอดภัยของระบบแล้วระบบยังมีการตรวจสอบช่วงระยะเวลาที่ผู้ใช้มีปฏิสัมพันธ์กับระบบหรือไทม์เอาท์ โดยระบบจะปฏิเสธที่จะให้บริการแก่ผู้ใช้ที่ขาดการติดต่อกับระบบนานเกินกว่าช่วงระยะเวลาที่ได้ตั้งไว้ ผู้ใช้จำเป็นที่จะต้องทำการล็อกอินเข้าสู่ระบบใหม่ครั้งถ้าหากต้องการที่จะใช้บริการจากระบบอีกครั้ง วิธีการนี้เป็นการป้องกันการออกจากระบบที่ไม่ถูกต้องของผู้ใช้ซึ่งจะทำให้มีเซสชั่นคีย์ติดค้างอยู่ในเวบเบราเซอร์และเปิดโอกาสให้ผู้ใช้อื่นเข้าใช้บริการของระบบได้โดยไม่ต้องล็อกอินเข้าสู่ระบบ สำหรับการออกจากระบบที่ถูกต้องนั้น ผู้ใช้จะต้องเรียกใช้แอพพลิเคชั่น UserSignOff ผ่านทางรายการ Sign Off ในเมนูผู้ใช้ แอพพลิเคชั่นจะทำการขอให้เวบเบราเซอร์ลบเซสชั่นคีย์ออกฐานข้อมูลของเวบเบราเซอร์เอง ทำให้ผู้ใช้อื่นในลำดับถัดไปไม่สามารถนำเซสชั่นคีย์ไปใช้ได้

เซอร์วิสที่ทำงานอยู่บนระบบจะมีเซอร์วิสประเภทหนึ่งที่ทำงานเป็นระยะเวลาคือเมื่อครบช่วงเวลาที่กำหนดก็จะทำงานครั้งหนึ่งแล้วรอจนถึงช่วงเวลาอีกครั้งหนึ่งแล้วจึงทำต่ออีกครั้งเป็นวัฏจักร เซอร์วิสเหล่านี้ได้แก่ StatPolling และ GraphPainter เซอร์วิสประเภทนี้หากตั้งค่าของระยะเวลาในแต่ละรอบการทำงานสั้นเกินไปจะทำให้ความถี่ในการทำงานเพิ่มมากขึ้นซึ่งจะทำให้ภาระของเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นผู้จัดการเครือข่ายเพิ่มมากขึ้นและส่งผลกระทบไปยังการทำงานอื่นๆของระบบได้ สำหรับระบบในปัจจุบันได้ตั้งค่าของระยะเวลาการทำงานแต่ละรอบของ StatPolling ไว้ที่ 15 วินาที และ GraphPainter ที่ 1 นาที ซึ่งเครื่องคอมพิวเตอร์ที่เป็นผู้จัดการเครือข่ายตามคุณสมบัติที่ได้กล่าวไปแล้วในบทที่ 4 สามารถรองรับได้

โปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบนอกจากจะสามารถทำงานด้านการจัดการเครือข่ายแล้ว ระบบยังสามารถเพิ่มโมดูลที่ทำงานด้านอื่นๆเข้าสู่ระบบได้เช่นกันทั้งนี้เนื่องจากความ

สามารถในการปรับเปลี่ยนหรือเพิ่มเติมโมดูลของระบบนั่นเอง สำหรับข้อเปรียบเทียบระหว่างโปรแกรม Web-based SNMP Manager กับโปรแกรมอื่นๆ มีดังนี้

ตารางที่ 5.1 เปรียบเทียบคุณสมบัติของ Web-based SNMP Manager กับโปรแกรมจัดการเครือข่ายรูปแบบอื่นๆ

คุณสมบัติ	Web-based SNMP Manager	โปรแกรมจัดการเครือข่ายรูปแบบอื่นๆ		
		X-Window	แบบกระจาย	แบบรวมศูนย์
1. การทำงานแบบกราฟฟิก	สนับสนุน	สนับสนุน	สนับสนุน	สนับสนุน
2. การรักษาความปลอดภัยในระดับผู้ใช้	สนับสนุน	ขึ้นกับระบบปฏิบัติการ	ขึ้นกับอุปกรณ์เครือข่ายและโปรโตคอลที่ใช้ในการจัดการเครือข่าย	เช่นเดียวกับแบบกระจาย
3. การรักษาความปลอดภัยในระดับโมดูล	สนับสนุน	ไม่สนับสนุน	ไม่สนับสนุน	ไม่สนับสนุน
4. ภาระทราฟฟิกที่เกิดจากการทำงาน	ต่ำ	สูง	ต่ำ	ต่ำ
5. การขยายและเพิ่มเติมโมดูล	สนับสนุน	ขึ้นกับโครงสร้าง	ไม่สนับสนุน	ขึ้นกับโครงสร้าง
6. ไฟร์วอลล์คอนฟิกกูเรชั่น	สะดวก	สะดวก	ไม่สะดวก	สะดวก

เมื่อพิจารณาจากตารางจะพบว่าคุณสมบัติที่โปรแกรมจัดการเครือข่ายทุกแบบมีคือการสนับสนุนการทำงานแบบกราฟฟิก ในขณะที่การรักษาความปลอดภัยในระดับผู้ใช้นั้นจะมีเพียง Web-based SNMP Manager ซึ่งใช้เซสชั่นคีย์ในการรักษาความปลอดภัยเท่านั้นที่สนับสนุน โดยแบบ X-Window จะต้องอาศัยระบบปฏิบัติการในการรักษาความปลอดภัยในระดับผู้ใช้ ส่วนโปรแกรมแบบกระจายและแบบรวมศูนย์จะเป็นภาระของอุปกรณ์เครือข่ายหรือโปรโตคอลจัดการเครือข่ายที่จะทำหน้าที่รักษาความปลอดภัยในระดับผู้ใช้ ยกตัวอย่างเช่น การให้อุปกรณ์เครือข่ายเป็นฝ่ายกลั่นกรองเลขหมายไอพีของเวบเบราเซอร์ หรือการใช้คอมมิวนิตี้เนมเป็นรหัสผ่านในโปรโตคอล SNMP เป็นต้น สำหรับการรักษาความปลอดภัยในระดับโมดูลนั้นพบได้เพียงใน Web-based SNMP Manager โดยโปรแกรมจะควบคุมการทำงานของโมดูลด้วยทิกเก็ต เซอร์วิสอินโฟ และเซอร์วิสคีย์ ในขณะที่โปรแกรมอื่นๆที่มีลักษณะเป็นแพล็ตฟอร์มและสามารถติดตั้งโมดูลเข้าสู่โปรแกรมได้จะมีการไว้วางใจกันระหว่างโมดูล โดยหลังจากที่โมดูลถูกติดตั้งเข้าสู่โปรแกรมแล้ว โปรแกรมก็จะไม่มีการตรวจสอบโมดูลอีกเมื่อโมดูลเริ่มทำงาน ในส่วนของภาระทราฟฟิกที่เกิดขึ้นจากการทำงาน แบบ X-Window จะทำให้เกิดปริมาณข้อมูลบนระบบเครือข่ายมากกว่าแบบอื่นๆ ทั้งนี้เนื่องจากโปรโตคอล X นั่นเอง ส่วนคุณสมบัติการขยายและเพิ่มเติมโมดูลเพื่อให้ระบบมีความสามารถมากขึ้นซึ่งมีอยู่ใน Web-based SNMP Manager แต่ใน X-Window และแบบรวมศูนย์จะขึ้นอยู่กับโครงสร้างที่โปรแกรมนั้นๆถูกสร้างขึ้นมา การเปรียบเทียบสุดท้ายคือไฟร์วอลล์คอนฟิกกูเรชั่น โปรแกรมแบบกระจายจะทำให้ผู้ดูแลระบบมีภาระงานมากขึ้นเนื่องจากเลขหมายไอพีและจำนวนเครื่องที่ต้องการติดต่อกับอุปกรณ์เครือข่ายที่อยู่ภายใต้ไฟร์วอลล์นั่นเอง

จากคุณสมบัติต่างๆที่กล่าวมาจะพบว่า Web-based SNMP Manager ได้รวบรวมข้อดีของโปรแกรมแต่ละรูปแบบเข้ามารวมกันไว้ในรูปแบบเดียว

5.2 ข้อเสนอแนะสำหรับโปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบ

จากคุณสมบัติของโปรแกรมผู้จัดการเครือข่ายด้วย SNMP ที่ได้กล่าวมาแล้วสามารถนำโปรแกรมไปพัฒนาต่อเพื่อให้มีขีดความสามารถในการทำงานสูงขึ้นหรือนำไปประยุกต์ใช้งานด้านอื่นๆ ได้ตามข้อเสนอแนะดังนี้

1. สามารถนำโครงสร้างการทำงานของ Web-based SNMP Manager ในงานวิจัยนี้ซึ่งทำงานอยู่บนระบบปฏิบัติการวินโดว์ 95 ไปสร้างบนระบบปฏิบัติการอื่นๆ เช่น UNIX โดยอาศัยหลักการเดียวกันได้

2. เซอร์วิสเช่น NetService, JobSchedule, StatPolling และ GraphPainter ต่างถูกสร้างขึ้นมาเพื่อสนับการทำงานด้านการจัดการเครือข่าย แต่ด้วยโครงสร้างแบบโมดูลาร์และคุณสมบัติการจัดการฟังก์ชันและเซอร์วิสของระบบ ทำให้ผู้พัฒนาสามารถสร้างเซอร์วิสอื่นๆที่ไม่เกี่ยวข้องกับการจัดการเครือข่ายและทำงานอยู่บนระบบได้ เช่น เซอร์วิสที่ทำหน้าที่เป็นตัวกลางในการติดต่อกับดาต้าเบสเซิร์ฟเวอร์ต่างๆบนระบบเครือข่ายเป็นต้น

3. เนื่องจากการติดต่อระหว่างแอพพลิเคชั่นกับเซอร์วิสหรือเซอร์วิสกับเซอร์วิสเองต่างก็สื่อสารกันด้วยเน็ตเวิร์กโปรโตคอล ทำให้การพัฒนาในโอกาสต่อไปสามารถกระจายการทำงานของเซอร์วิสต่างๆไปสู่เซิร์ฟเวอร์อื่นๆบนระบบเครือข่ายเพื่อกระจายภาระงานที่จะเกิดขึ้น

4. สำหรับระบบ Web-based SNMP Manager ในงานวิจัยนี้เป็นการนำเสนอรูปแบบของตัวกลางระหว่างเวบเบราเซอร์กับอุปกรณ์เครือข่ายเพื่อให้ผู้ดูแลระบบสามารถใช้ประโยชน์จากรูปแบบดังกล่าวนี้ในการควบคุมการเข้าถึงอุปกรณ์เครือข่ายโดยตรงจากผู้ใช้ แต่ทั้งนี้และทั้งนั้นในการนำไปใช้งานควรนำระบบนี้ใช้ร่วมกับอุปกรณ์รักษาความปลอดภัยเช่นไฟร์วอลล์หรือพร็อกซี่เซิร์ฟเวอร์ เพื่อให้อุปกรณ์เหล่านี้ทำหน้าที่เสมือนเป็นกำแพงป้องกันการเข้าถึงอุปกรณ์เครือข่ายโดยตรงของผู้ใช้และให้ Web-based SNMP Manager ทำหน้าที่เสมือนเป็นทางเข้าหรือประตูสำหรับผู้ใช้ที่มีสิทธิที่ถูกต้องสามารถติดต่อกับอุปกรณ์เครือข่ายโดยผ่านระบบที่ได้สร้างขึ้น

5. การทำงานของฟังก์ชัน SNMPNetworkDiscovery ซึ่งทำหน้าที่ในการสืบค้นเครือข่ายมีความสามารถในการจัดการหน่วยความจำที่จำกัดทำให้ไม่สามารถทำงานกับตารางเราท์ติ้งในเราท์เตอร์ที่มีขนาดใหญ่ได้ และเทคนิกในการสืบค้นในบางส่วนเช่นการดึงข้อมูลที่เป็นคุณสมบัติทั่วไปที่ไม่เกี่ยวกับเส้นทางการสืบค้นซึ่งสามารถที่จะใช้เทคนิกแบบ Multi Thread ได้ยังเป็นแบบ Single Thread ทำให้การสืบค้นต้องใช้เวลามากกว่า นอกจากนี้ยังสนับสนุนการสืบค้นข้อมูลของอุปกรณ์ที่มีคอมมิวนิตี้เนมเหมือนกันเท่านั้น ซึ่งในความเป็นจริงอุปกรณ์บนระบบเครือข่ายหนึ่งๆอาจจะมีคอมมิวนิตี้เนมที่แตกต่างกันได้

6.การวาดรูปแผนที่เครือข่ายซึ่งอยู่ในสามารถปรับปรุงให้มีการจัดวางวัตถุบนแผนที่ให้ดีขึ้น โดยแอลกอริทึมที่ใช้อยู่ในปัจจุบันเป็นการวางวัตถุในแนวตั้งและแนวนอนโดยไม่ได้พิจารณาความสมดุลของรูปภาพ

7.เซอร์วิสที่คอยรับสัญญาณแทรปจากอุปกรณ์เครือข่ายเมื่อมีเหตุการณ์เกิดขึ้นกับอุปกรณ์นั้นๆ โดยอาจจะทำการส่งเมล์ให้กับผู้บริหารเครือข่ายได้รับทราบ หรือแจ้งเหตุการณ์ที่เกิดขึ้นบนแผนที่เครือข่ายหรือแจ้งแก่เมสเสจเมเนเจอร์เพื่อประกาศลงบนกระดานข่าวสารอีกทอดหนึ่ง

8.ปรับปรุงระบบยูสเซอร์อินเตอร์เฟสให้ดีขึ้นเพื่อการใช้งานที่สะดวกขึ้นโดยให้มีการใช้ประโยชน์จากกราฟฟิกให้มากขึ้นเช่นแผนที่เครือข่ายที่ผู้ใช้สามารถปรับเปลี่ยนตำแหน่งที่ตั้งบนแผนที่ของอ๊อปเจ็กท์เพื่อความสะดวกในการทำงาน เป็นต้น

เอกสารอ้างอิง

[1] Adrew S. Tanenbaum. **Computer Networks 3rd Edition.** New Jersey: Prentice-Hall Interantional, Inc. 1996.

[2] Alex Berson. **Client/Server Architecture International Edition**, New York: McGraw-Hill, 1994.

[3] Cisco. "CiscoView - GUI Monitoring For All Cisco Devices" [Online] Available: http://www.cisco.com/warp/public/cc/pd/wr2k/view/index.shtml

[4] D. Brent Chapman and Elizabeth D.Zwicky. Building Internet Firewalls, California: O'Reilly & Associates, Inc. 1995.

[5] D. Edgar Taylor. TCP/IP Complete, New York: McGraw-Hill, 1998.

[6] Inprise Corporation. **Borland Delphi 5 Developer's Guide for Windows 98, Windows 95, & Windows NT.** Carifornia: Inprise Corporation. 1999.

[7] Kris Jamsa, Suleiman Lalani, Steve Weakley. Web Programming, Las Vegas: Jamsa Press, 1996.

[8] Robert Orfali, Dan Harkey, Jeri Edwards. **Client/Server Survival Guide 3rd Edition**, New York: Wiley, John Wiley & Sons, Inc., 1999.

[9] Student Workbook. **HP Open View Network Node Manager Fundamentals for Network Managers**, Network and System Management Division, HEWLETT PACKARD, 1997.

[10] SunNet Manager 2.0 User's Guide. **SunConnect.** A Sun Microsystems Inc. 1992.

[11] Thomas A. Powell, **The Complete Reference HTML**: Osborne/McGraw-Hill, 1998.

[12] William Stallings. **SNMP, SNMPv2 and RMON.** Massachusetts: Addison-Wesley Publishing Company, Inc. 1996.

[13] Xavier Pacheco and Steve Teixeria. **Delphi 4 Developer's Guide.** Indianapolis: Sams Publishing. 1998.

ภาคผนวก

ภาคผนวก ก.

วิธีการใช้งานโปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบ

โปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบเป็นโปรแกรมที่พัฒนาขึ้นโดยใช้ภาษาปาสคาลและโปรแกรมบอร์แลนด์เดลไฟ มีความสามารถในการจัดการระบบเครือข่ายเช่น การสืบค้นเครือข่าย การตรวจสอบสถานะของทรัพยากรและแก้ไขค่าพารามิเตอร์บนอุปกรณ์เครือข่ายโดยใช้คำสั่งพื้นฐานจากโปรโตคอล SNMP จุดประสงค์ของโปรแกรมคือเป็นตัวกลางในการติดต่อระหว่างผู้ใช้กับอุปกรณ์เครือข่ายโดยผู้ใช้สามารถใช้งานโปรแกรมผ่านทางระบบเวบ ทำให้เกิดความสะดวกแก่ผู้บริหารเครือข่ายในการดูแลระบบเครือข่ายจากที่ต่างๆที่มีระบบเครือข่ายที่เชื่อมต่อกับระบบอินเตอร์เน็ต

โปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบเป็นโปรแกรมที่สนับสนุนการทำงานแบบไทม์แชร์ลิ่ง โปรแกรมสามารถรองรับรีเควสท์จากผู้ใช้ได้มากกว่า 1 รีเควสท์ในเวลาเดียวกัน ผู้ใช้แต่ละคนบนระบบจะได้รับการกำหนดระดับของสิทธิในการใช้บริการของระบบ ในการทำงานกับระบบผู้ใช้สามารถใช้เวบเบราเซอร์สำหรับล็อกอินเข้าสู่ระบบโดยข้อมูลที่ผู้ใช้ต้องแจ้งแก่ระบบเพื่อใช้ในการตรวจสอบสิทธิคือ ชื่อ รหัสผ่านของผู้ใช้และเลขหมายไอพีของเครื่องคอมพิวเตอร์ที่เวบเบราเซอร์ทำงานอยู่ เมื่อผ่านการตรวจสอบเวบเบราเซอร์จะได้รับข้อมูลเพื่อนำมาแสดงเป็นเมนูของบริการที่ระบบสามารถมีให้แก่ผู้ใช้ตามระดับสิทธิที่ผู้ใช้มีและเซสชั่นคีย์เพื่อให้เวบเบราเซอร์ใช้เป็นเสมือนรหัสในการติดต่อกับระบบ จากเมนูผู้ใช้สามารถเรียกใช้บริการต่างๆ ที่เกี่ยวข้องกับการจัดการเครือข่ายเช่น MIB Browser ซึ่งผู้ใช้สามารถใช้บริการนี้ในการขอหรือแก้ไขข้อมูลที่อยู่บน MIB ของอุปกรณ์เครือข่าย หรือ Network Map เพื่อดูแผนที่เครือข่าย เป็นต้น เมื่อผู้ใช้ออกจากระบบโดยการล็อกเอ๊าท์ ระบบจะสั่งให้เวบเบราเซอร์ลบเซสชั่นคีย์ออกจากแฟ้มข้อมูลของเวบเบราเซอร์ซึ่งจะเป็นผลให้ผู้ใช้คนอื่นไม่สามารถใช้เวบเบราเซอร์เดียวกันนี้เพื่อขอใช้บริการจากระบบโดยปราศจากการล็อกอิน

ก.1 ความต้องการของระบบ

เนื่องจากระบบเป็นโปรแกรมที่ถูกพัฒนาขึ้นภายใต้การทำงานของระบบปฏิบัติการไมโครซอฟต์วินโดวส์ ดังนั้นการกำหนดความต้องการของระบบจะพิจารณาตามระบบปฏิบัติการที่โปรแกรมทำงานอยู่ ซึ่งมีดังต่อไปนี้

1. ส่วนของโปรแกรมผู้จัดการเครือข่าย

1.1 คุณสมบัติทางด้านฮาร์ดแวร์

หน่วยประมวลผลกลาง : AMD 230 MHz

ขนาดหน่วยความจำหลัก : 48 MB

ความจุฮาร์ดดิส : 4 GB

1.2 ระบบปฏิบัติการ

Microsoft Windows 95

1.3 เวบเซิร์ฟเวอร์

Microsoft Personal Web Server 1.0a

1.4 SNMP Library

Dart's PowerTCP Tool kit

2. ส่วนของไคลเอ็นต์

2.1 คุณสมบัติทางด้านฮาร์ดแวร์

หน่วยประมวลผลกลาง : Pentium 133 MHz

ขนาดหน่วยความจำหลัก : 32 MB

ความจุฮาร์ดดิส : 1.2 GB

2.2 ระบบปฏิบัติการ

Microsoft Windows 95

2.3 เวบเบราเซอร์

Microsoft Internet Explorer หรือ Netscape Communicator

3. เครื่องมือที่ใช้ในการพัฒนา

3.1 Borland Delphi version 5.0

3.2 โปรแกรม Database Desktop ของ Borland Delphi version 5.0

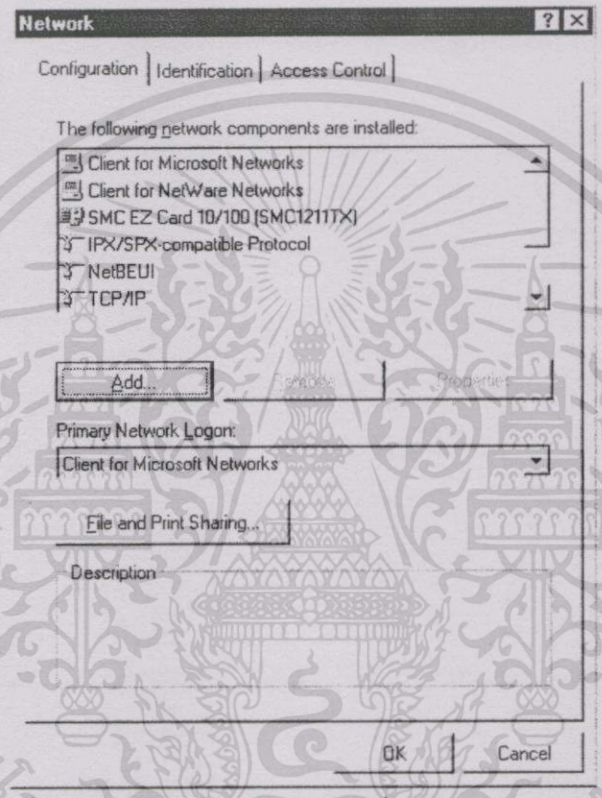
ก.2 การติดตั้ง

ก.2.1 การติดตั้ง PowerTCP

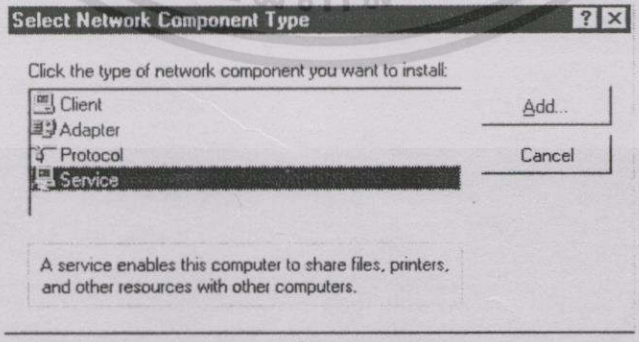
เนื่องจากไลบาร์รี่ของ PowerTCP มีลักษณะเป็นไลบาร์รี่แบบ DLL (Dynamic Link Library) จึงทำให้เมื่อมีการติดตั้งโปรแกรมผู้จัดการเครือข่ายจำเป็นที่จะต้องติดตั้งไลบาร์รี่ของ PowerTCP ลงในเครื่องที่ทำหน้าที่เป็นผู้จัดการเครือข่ายด้วย การติดตั้งทำได้โดยการสั่งรันโปรแกรม PT-105T และใส่ไลเซ่นทางการค้าเมื่อโปรแกรมต้องการ และสามารถเลือกตำแหน่งที่จะติดตั้งบนระบบปฏิบัติการ เมื่อติดตั้งสำเร็จจะมองเห็นโปรแกรมที่อยู่ในชุด PowerTCP ปรากฏในเมนูของวินโดวส์

ก.2.2 การติดตั้งโปรแกรมเวบเซิร์ฟเวอร์ Microsoft Personel Web Server 1.0a

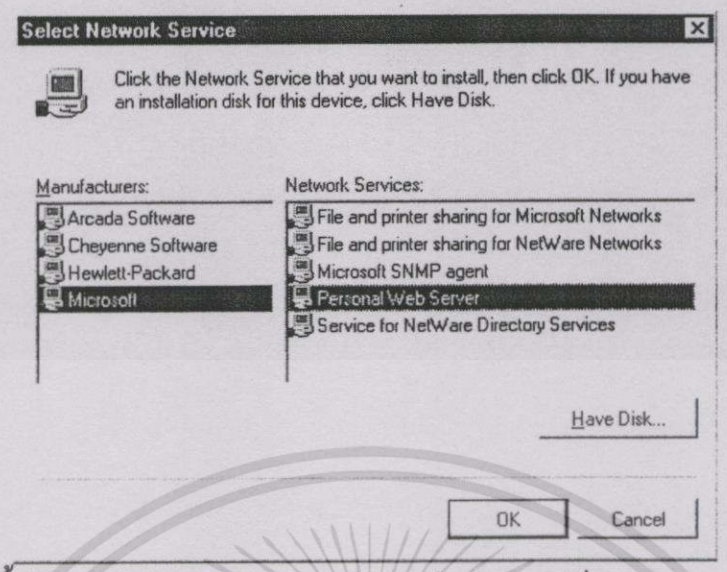
การติดตั้งโปรแกรม Personel Web Server 1.0a ทำโดยการสั่งรันเอ็กสิคิวส์ไฟล์ Pws10a โปรแกรมก็จะติดตั้งตัวเองเข้าสู่ระบบปฏิบัติการวินโดวส์ เนื่องจากโปรแกรมถูกสร้างให้ทำหน้าที่เป็นเซอร์วิสในโปรโตคอลสแต็กของระบบปฏิบัติการ ดังนั้นจึงต้องมีการเพิ่มเซอร์วิสของเวบเซิร์ฟเวอร์ดังกล่าวนี้เข้าไปสู่โปรโตคอลสแต็ก ดังรูปที่ ก.1-ก.4



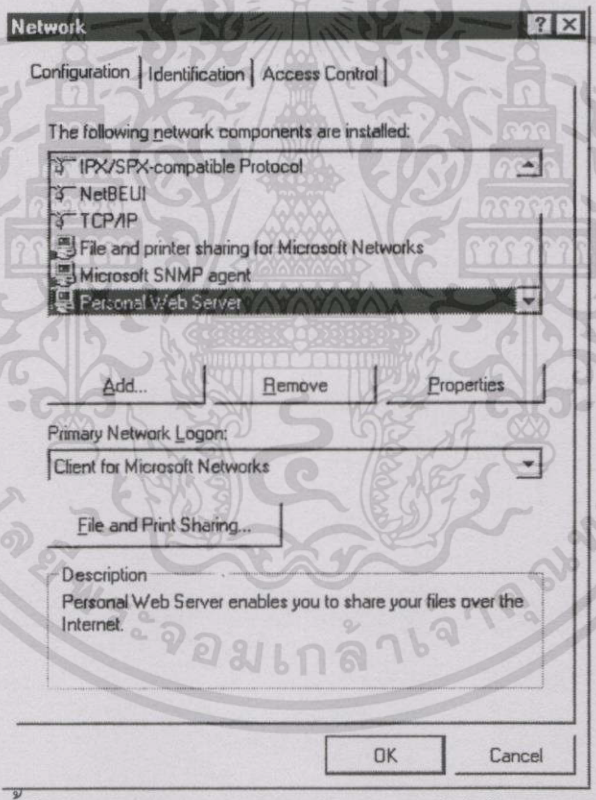
รูปที่ ก.1 โปรโตคอลสแต็กของระบบปฏิบัติการวินโดวส์



รูปที่ ก.2 การติดตั้งเซอร์วิสเข้าสู่โปรโตคอลสแต็ก



รูปที่ ก.3 การติดตั้งโปรแกรม Personel Web Server ของไมโครซอฟต์



รูปที่ ก.4 ผลจากการติดตั้งโปรแกรม Personel Web Server บนโปรโตคอลสแต็ก

ก.2.3 การติดตั้งโปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบ

โปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบมีการกำหนดระบบไฟล์และตำแหน่งที่ตั้งของโปรแกรมที่เกี่ยวข้องดังนี้ดังตารางที่ ก.1

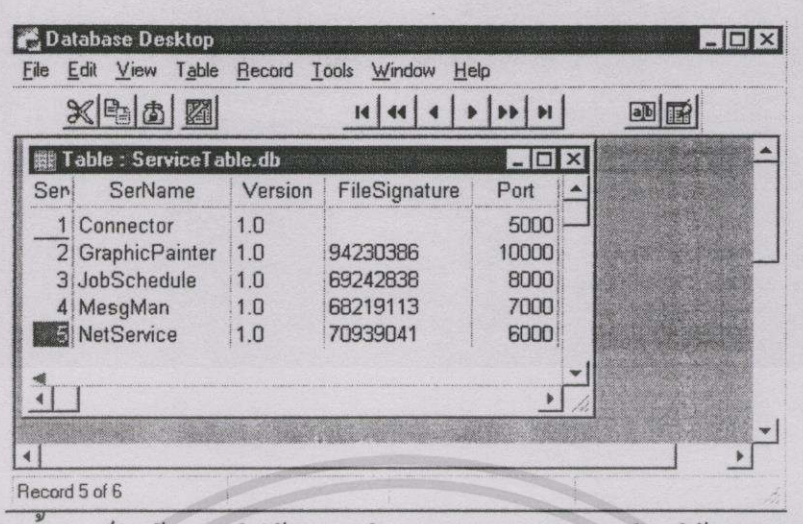
ตารางที่ ก.1 ไฟล์สำคัญต่างๆที่ใช้ในระบบของโปรแกรมผู้จัดการเครือข่ายด้วย SNMP โดยใช้เวบ

ชนิดของไฟล์	คำอธิบาย	ตำแหน่งที่ตั้ง
*.exe	โปรแกรมของเซอร์วิสโมดูล	<Window-Path>
*.db	ไฟล์ตารางข้อมูล	<Window-Path>/Database/Table
*.dll	ฟังก์ชันขยายของโมดูล	<Window-Path>/Functions
*.jpg	ไฟล์รูปภาพ	<Window-Path>/Output
*.html	ไฟล์เอกสาร HTML	/wwwroot/
*.dll	ไฟล์เวบเซิร์ฟเวอร์แอพพลิเคชั่น	/scripts/

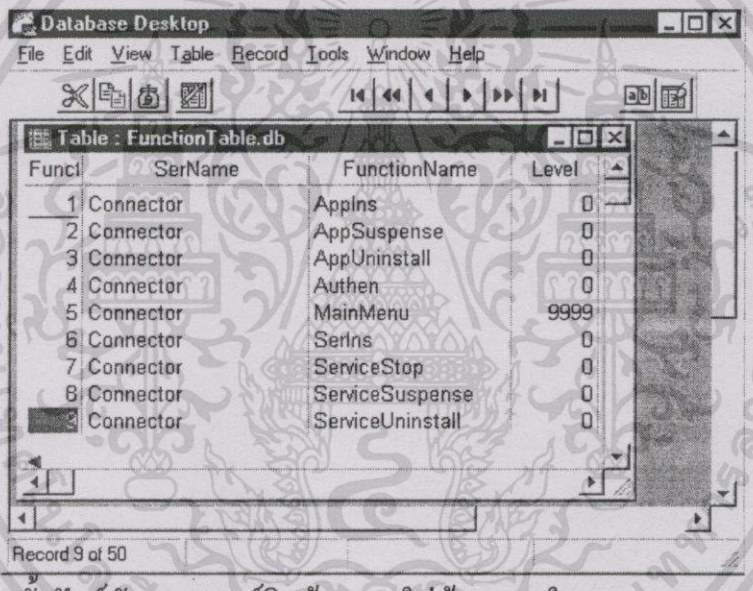
เนื่องจากระบบประกอบด้วยกลุ่มของโมดูลหลายโมดูลทำงานร่วมกันจึงทำให้ในการติดตั้งระบบเพื่อให้สามารถทำงานได้ครบตามฟังก์ชันที่มีอยู่จำเป็นที่จะต้องมีการใส่ข้อมูลของแต่ละโมดูลลงในตารางข้อมูลที่เกี่ยวข้องกับโมดูลนั้นๆ โปรแกรมที่ใช้ในการป้อนข้อมูลเพื่อติดตั้งระบบคือ Database Desktop ซึ่งเป็นยูทิลิตี้ที่มากับโปรแกรม Borland Delphi 5.0 การติดตั้งโมดูลบนระบบต้องใส่ข้อมูลของโมดูลลงในตารางที่เกี่ยวข้องโดยใช้ Database Desktop โมดูลที่เป็นเซอร์วิสจะถูกป้อนข้อมูลลงในตาราง ServiceTable และ FunctionTable ดังรูปที่ ก.5 และ ก.6 ส่วนโมดูลที่เป็นแอพพลิเคชั่นจะถูกป้อนข้อมูลลงในตาราง AppTable และ MenuList ดังรูปที่ ก.7 และ ก.8 ตามลำดับ ข้อมูลของโมดูลทั้งหมดมีดังนี้

ตารางที่ ก.2 ตารางข้อมูลของเซอร์วิสที่จะใช้ป้อนลงในตาราง ServiceTable

ลำดับ	ชื่อเซอร์วิส	เวอร์ชัน	พอร์ท	รายละเอียด
1.	Connector	1.0	5000	Add at 7th Sep. 2000
2.	GraphPainter	1.0	10000	Add at 12th Jan. 2001
3.	JobSchedule	1.0	8000	Add at 12th Jan. 2001
4.	MesgMan	1.0	7000	Add at 14th Dec. 2000
5.	NetService	1.0	6000	Add at 1st Dec. 2000
6.	StatPollingService	1.0	9000	Add at 12th Jan. 2001



รูปที่ ก.5 การติดตั้งเซอร์วิสด้วยการใส่ข้อมูลลงในตาราง ServiceTable โดยใช้ Database Desktop



รูปที่ ก.6 การติดตั้งฟังก์ชันของเซอร์วิสด้วยการใส่ข้อมูลลงในตาราง FunctionTable โดยใช้ Database Desktop

ตารางที่ ก.3 ฟังก์ชันของเซอร์วิสที่จะถูกป้อนลงสู่ตาราง FunctionTable

ลำดับ	ชื่อเซอร์วิส	ชื่อฟังก์ชัน	ระดับสิทธิ
1.	Connector	Auth	0
2.	Connector	MainMenu	9999
3.	Connector	SignOff	9999
4.	Connector	SignOn	9999
5.	GraphPainter	Dead	0
6.	GraphPainter	GetServiceKey	0

ตารางที่ ก.3 (ต่อ)

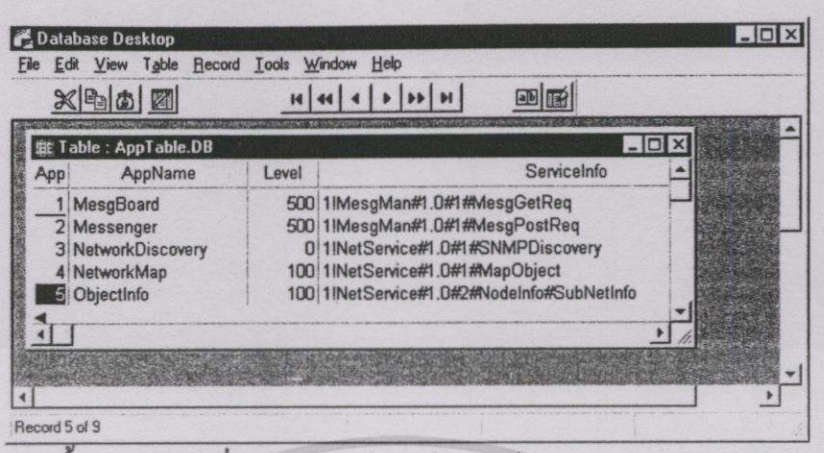
ลำดับ	ชื่อเซอร์วิส	ชื่อฟังก์ชัน	ระดับสิทธิ
7.	GraphPainter	InsertTicket	0
8.	JobSchedule	AllJobData	100
9.	JobSchedule	Dead	0
10.	JobSchedule	GetServiceKey	0
11.	JobSchedule	InsertTicket	0
12.	JobSchedule	JobForPoll	100
13.	JobSchedule	LinkTableData	100
14.	JobSchedule	StoreJob	100
15.	JobSchedule	UpdateJobStatus	100
16.	JobSchedule	UpdateLinkTable	100
17.	MesgMan	Dead	0
18.	MesgMan	GetServiceKey	0
19.	MesgMan	InsertTicket	0
20.	MesgMan	MesgGetReq	500
21.	MesgMan	MesgPostReq	500
22.	NetService	Dead	0
23.	NetService	GetNextRequest	100
24.	NetService	GetRequest	100
25.	NetService	GetServiceKey	0
26.	NetService	InsertTicket	0
27.	NetService	MapObject	100
28.	NetService	NodeInfo	100
29.	NetService	SNMPDiscovery	0
21.	NetService	SetRequest	100
22.	NetService	SubNetInfo	100
23.	StatPollingService	Dead	0
24.	StatPollingService	GetServiceKey	0
25.	StatPollingService	InsertTicket	0

ตารางที่ ก.4 ข้อมูลของแอพพลิเคชั่นที่จะต้องถูกป้อนลงสู่ตาราง AppTable

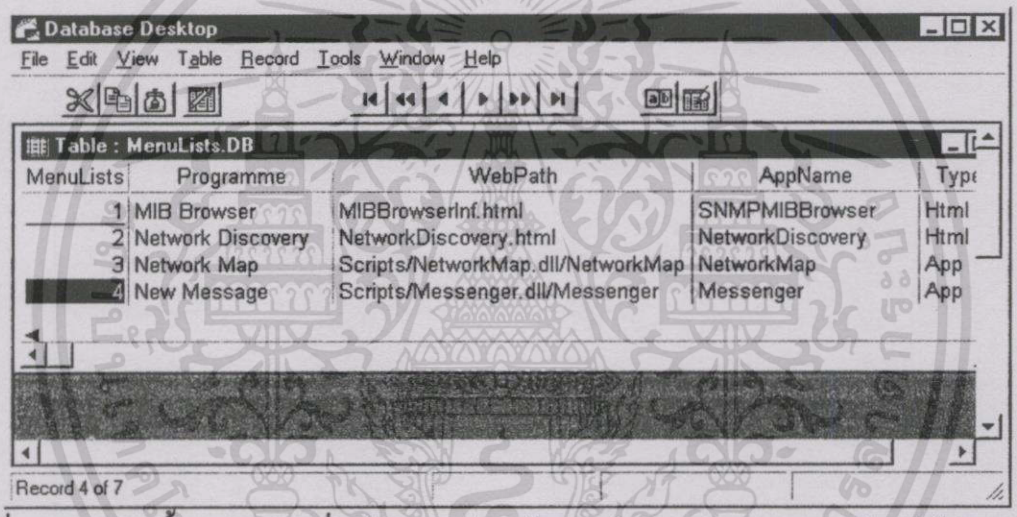
ลำดับ	ชื่อแอพพลิเคชั่น	ระดับสิทธิ	เซอร์วิสอินโฟ	เวอร์ชัน
1.	MesgBoard	500	1!MesgMan#1.0#1#MesgGetReq	1.0
2.	Messenger	500	1!MesgMan#1.0#1#MesgPostReq	1.0
3.	NetworkDiscovery	0	1!NetService#1.0#1#-SNMPDiscovery	1.0
4.	NetworkMap	100	1!NetService#1.0#1#MapObject	1.0
5.	ObjectInfo	100	1!NetService#1.0#2#NodeInfo#Sub-NetInfo	1.0
6.	Polling	100	1!JobSchedule#1.0#2#StoreJob#All-JobData	1.0
7.	SNMPMIBBrowser	100	1!NetService#1.0#3#GetNextRe-quest#GetRequest#SetRequest	1.0
8.	UserSignOff	9999	1!Connector#1.0#1#SignOff	1.0
9.	UserSignOn	9999	1!Connector#1.0#2#SignOn#-MainMenu	1.0

ตารางที่ ก.5 ข้อมูลของรายการเมนูที่จะถูกป้อนลงสู่ตาราง MenuList

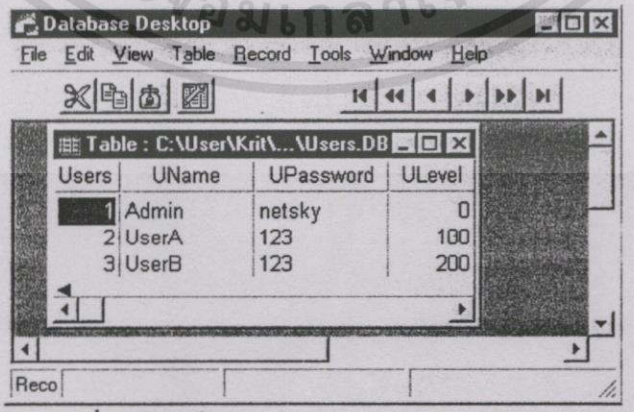
ลำดับ	ชื่อรายการ	เวบพาร์ท	ประเภท	ชื่อแอพพลิเคชั่น
1.	MIB Browser	MIBBrowserInf.html	HTML	SNMPMIBBrowser
2.	Network Discovery	NetworkDiscovery.html	HTML	NetworkDiscovery
3.	NetworkMap	Scripts/NetworkMap.dll/NetworkMap	App	NetworkMap
4.	New Message	Scripts/Messenger.dll/Messenger	App	Messenger
5.	Poll Browser	Scripts/Polling.dll/PollBrowser	App	Polling
6.	Poll Request	PollReqForm.html	HTML	Polling
7.	Sign Off	SignOff.html	HTML	UserSignOff



รูปที่ ก.7 การติดตั้งแอพพลิเคชั่นด้วยการป้อนข้อมูลเข้าสู่ตาราง AppTable โดยใช้ Database Desktop



รูปที่ ก.8 การติดตั้งแอพพลิเคชั่นด้วยการป้อนข้อมูลเข้าสู่ตาราง MenuList โดยใช้ Database Desktop



รูปที่ ก.9 การจัดการบัญชีรายชื่อผู้ใช้ในตารางข้อมูล Users โดยใช้ Database Desktop

สำหรับไฟล์ที่ทำการเก็บค่าเริ่มต้นของระบบคือ Config.txt เป็นไฟล์แบบตัวอักษรถูกกำหนดให้อยู่ในไดร์ฟซี (C:\Config.txt) เก็บค่าเลขหมายไอพีของเมเนเจอร์ไว้ในบรรทัดแรก และหมายเลขพอร์ทของคอนเน็กเตอร์ในบรรทัดที่สอง นอกจากนี้ในส่วนของการลงทะเบียนผู้ใช้ก็จัดเป็นส่วนหนึ่งของการติดตั้งเช่นกัน โดยผู้ดูแลระบบสามารถจัดการบัญชีผู้ใช้บนตารางข้อมูล Users ได้ด้วย Database Desktop ดังรูปที่ ก.9 ซึ่งข้อมูลสำคัญสำหรับผู้ใช้คือ ชื่อ รหัสผ่าน และ ระดับของสิทธิของผู้ใช้

ก.3 การสตาร์ทอัพระบบ

การทำให้ระบบเริ่มทำงานจำเป็นต้องมีการตรวจสอบความพร้อมของระบบเสียก่อน สิ่งที่จะต้องพิจารณาได้แก่

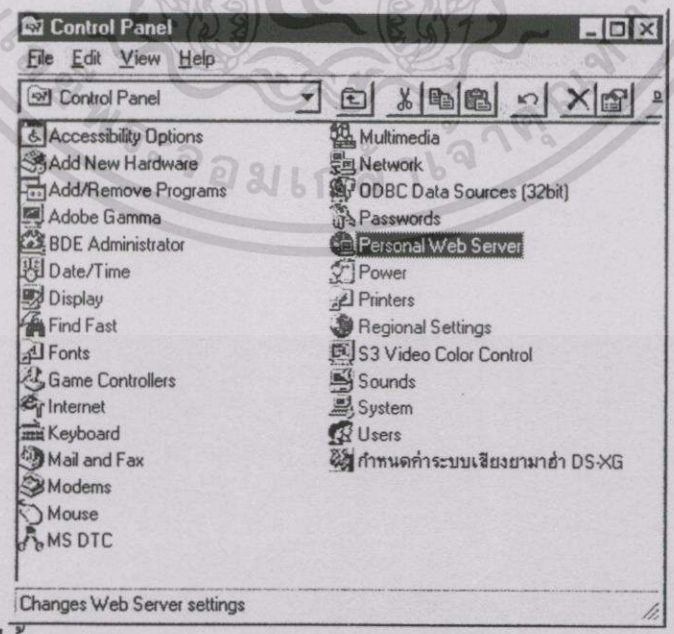
1. ต้องแน่ใจว่าไม่มีเซอร์วิสของระบบทำงานค้างอยู่บนเครื่องผู้จัดการเครือข่าย

2. เซิร์ฟเวอร์พอร์ทที่ถูกใช้เป็นพอร์ทของเซอร์วิสบนระบบ (พิจารณาจากพอร์ทในตาราง ก.2) ต้องไม่กำลังถูกโปรเซสอื่นนำไปใช้งาน

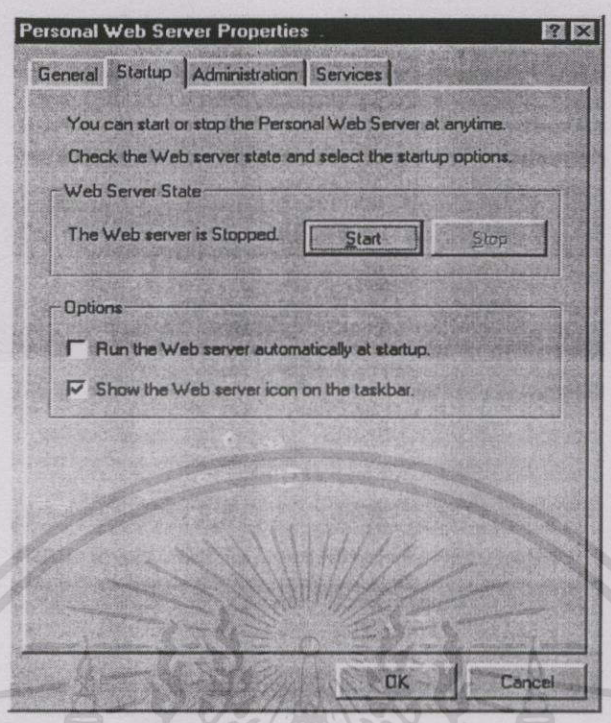
3. เนื่องจากไลบาร์รี่ SNMP ของ PowerTCP ที่ใช้เป็นไลเซ่นแบบผู้ใช้คนเดียวสำหรับเครือข่ายท้องถิ่นหรือแลนหนึ่งเครือข่าย ดังนั้นก่อนที่จะให้ระบบเริ่มทำงานจึงควรตรวจสอบเสียก่อนว่าในเครือข่ายท้องถิ่นไม่มีการใช้งานโปรแกรม PowerTCP อยู่ก่อน

หลังจากตรวจสอบความพร้อม ผู้บริหารระบบจึงสามารถสั่งให้ระบบทำงานโดยเริ่มจาก

1. เรียกใช้โปรแกรม Personel Web Server จาก Control Panel ดังรูปที่ ก.10 และสั่งให้โปรแกรมเริ่มทำงานดังรูปที่ ก.11

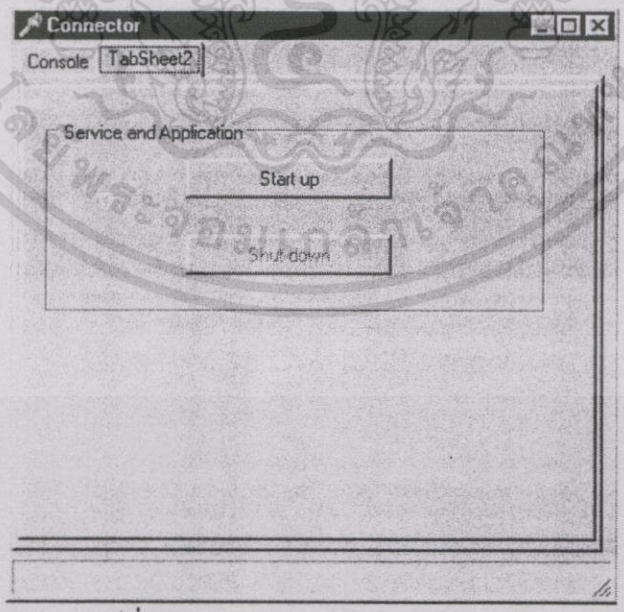


รูปที่ ก.10 ตำแหน่งที่ตั้งของโปรแกรม Personel Web Server



รูปที่ ก.11 การสั่งให้โปรแกรม Personel Web Server

2. สั่งให้โปรแกรมผู้จัดการเครือข่ายเริ่มทำงานโดยการรันไฟล์ Connector.exe ซึ่งอยู่ในไดเรกเทอรี่ของโปรแกรมผู้จัดการเครือข่าย โปรแกรมตัวนี้ก็คือคอนเนกเตอร์นั่นเอง ดังรูปที่ ก.12 (รายละเอียดการทำงานของคอนเนกเตอร์ในขั้นตอนนี้สามารถดูประกอบได้จากหัวข้อ Service and Application Start up ของคอนเนกเตอร์ในบทที่ 3)



รูปที่ ก.12 การสั่งให้คอนเนกเตอร์เริ่มทำงาน

ก.4 การชัทดาวน์ระบบ

เนื่องจากการทำงานของระบบจะประกอบด้วยเซิร์ฟเวอร์โปรเซสของเซอร์วิสคอยให้บริการต่างๆต่อรีเควสท์จากแอพพลิเคชั่นและงานภายในต่างๆของระบบด้วย ดังนั้นการที่จะชัทดาวน์ระบบจึงจะเป็นที่จะต้องมีความระมัดระวังงานที่เซอร์วิสกำลังประมวลผลอยู่ด้วย วิธีการชัทดาวน์จึงต้องการทำให้ระบบหยุดรับงานจากภายนอกเสียก่อน แล้วจึงให้ระบบหยุดการทำงานภายใน ขั้นตอนของการชัทดาวน์ระบบจึงมีดังนี้

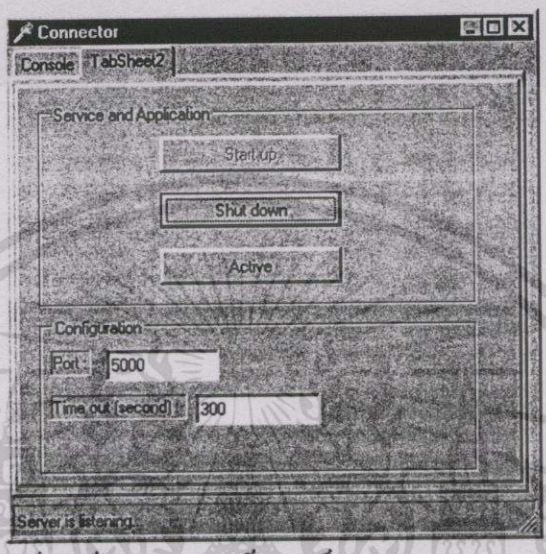
1. สั่งให้คอนเน็กเตอร์หยุดรับงานภายนอกโดยเลือกปุ่มรายการ 'InActive' ดังรูป ก.13 โดยในสถานะนี้คอนเน็กเตอร์จะไม่อนุญาตให้รีเควสท์จากแอพพลิเคชั่นใดๆใช้บริการจากเซอร์วิสด้วยการไม่สร้างทิกเก็ตให้แอพพลิเคชั่นนั่นเอง แต่สำหรับแอพพลิเคชั่นที่ได้รับทิกเก็ตไปก่อนหน้าที่คอนเน็กเตอร์จะถูกสั่ง InActive จะยังสามารถทำงานได้ตามปกติ

รูปที่ ก.13 ปุ่มรายการ InActive ของคอนเน็กเตอร์

2. เซอร์วิสไม่ควรจะมีการประมวลผลติดค้างอยู่ ผู้ดูแลระบบสามารถพิจาณาการทำงานของเซอร์วิสโดยลักษณะแรกคือ ให้ผู้ดูแลระบบหยุดการทำงานของเซอร์วิสที่มีการทำงานแบบรอบระยะเวลา โดยเลือกปุ่ม 'Stop' ที่อยู่บนคอนโซลของเซอร์วิส เซอร์วิสเหล่านี้ได้แก่ StatPolling และ GraphPainter และอีกลักษณะหนึ่งคือผู้ดูแลระบบควรจะรอให้เซอร์วิสว่างจากการประมวลผลงานใดๆ โดยสังเกตจากแถบแสดงสถานะภาพที่อยู่ด้านล่างของคอนโซลของเซอร์วิสซึ่งจะแสดงข้อความว่า 'Server is listening.' เซอร์วิสจะต้องอยู่ในสถานะดังกล่าวจึงจะแน่ใจได้ว่าการชัทดาวน์จะไม่ไปกระทบกระเทือนการทำงานของระบบ

3. สั่งให้เซอร์วิสสลายตัวจากระบบด้วยการเลือกปุ่มรายการ 'Shut down' ที่คอนโซลของคอนเน็กเตอร์ ดังรูปที่ ก.14 การสลายตัวของเซอร์วิสนี้จะยกเว้นเฉพาะคอนเน็กเตอร์ที่จะยังคงอยู่ทั้งนี้เพื่อผู้ดูแลระบบจะสามารถสั่งสตาร์ทอัพเซอร์วิสขึ้นมาอีกครั้งหากมีความต้องการสตาร์ทอัพระบบ

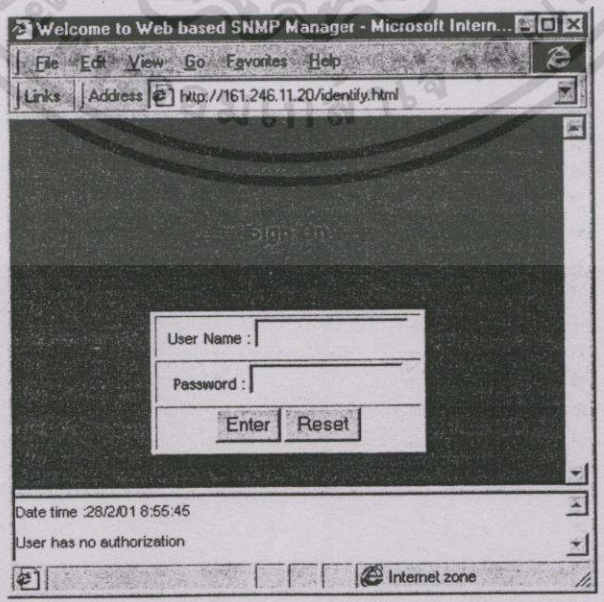
4. สั่งชัทดาวน์โปรเซสของเวบเซิร์ฟเวอร์



รูปที่ ก.14 ปุ่มรายการชัทดาวน์เซอร์วิสของคอนเน็กเตอร์

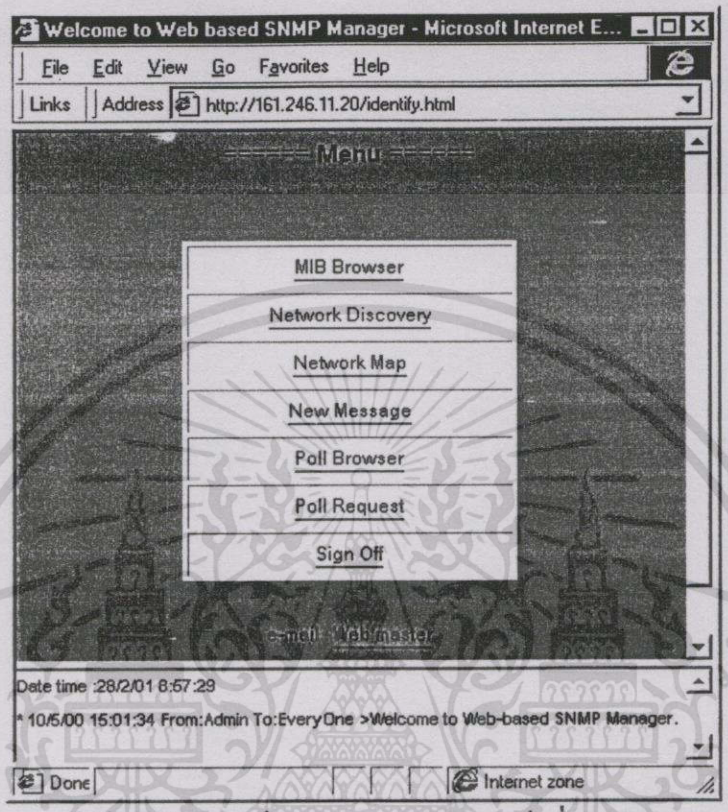
ก.5 การใช้งานระบบ

1. ก่อนที่จะขอใช้บริการจากระบบได้นั้น ผู้ใช้จำเป็นที่จะต้องล็อกอินเพื่อเข้าสู่ระบบเสียก่อนด้วยเวบเบราเซอร์โดยระบุ URL ไปที่ http://161.246.11.20/Identify.html และจะปรากฏหน้าจอดังรูปที่ ก.15 ซึ่งใช้สำหรับใส่ชื่อและรหัสผ่านของผู้ใช้เพื่อแสดงตนเข้าสู่ระบบ



รูปที่ ก.15 หน้าจอที่ใช้ในการล็อกอินเข้าสู่ระบบของผู้ใช้

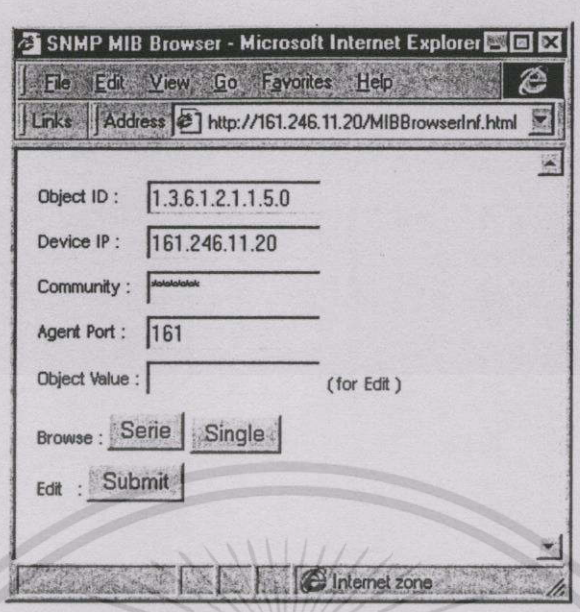
2.ในขณะที่ทำงานอยู่บนระบบหน้าจอบนเวบเบราเซอร์ของผู้ใช้จะแบ่งเป็น 2 ส่วนคือ ส่วนที่เป็นเมนูกับส่วนที่เป็นแมสเสจบอร์ดดังรูปที่ ก.16



รูปที่ ก.16 หน้าจอของผู้ใช้โดยส่วนบนเป็นเมนูและแมสเสจบอร์ดซึ่งอยู่ส่วนล่างของเวบเบราเซอร์

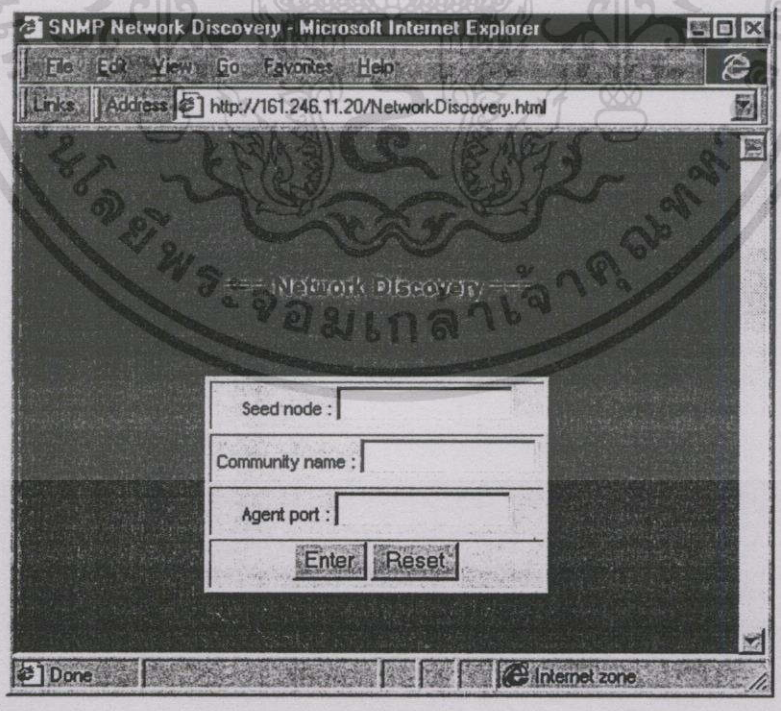
3. เมื่อผู้ใช้เลือกรายการที่ปรากฏอยู่บนเมนู รายการต่างๆจะมีหน้าจอแตกต่างกันตามการใช้งานดังรูปต่อไปนี้

3.1 ผู้ใช้สามารถจัดการกับข้อมูลใน MIB ของอุปกรณ์เครือข่ายจากการเลือกรายการ MIB Browser ดังรูปที่ ก.17 โดยสามารถเลือกประเภทของการทำงานได้ 3 ประเภท คือ Serie สำหรับการเรียกค้นข้อมูลแบบเป็นกลุ่ม ยกตัวอย่างเช่น หากใช้หมายเลขอ๊อปเจ็กท์เท่ากับ 1.3.6.1.2.1.1. ซึ่งเป็นการระบุหมายเลขในระดับกลุ่มโดย "1" ตัวสุดท้ายเป็นตัวบอกกลุ่ม System จึงต้องใช้คำสั่ง Serie ในการขอข้อมูลของกลุ่ม System ออกมา ในขณะที่หมายเลขอ๊อปเจ็กท์เท่ากับ 1.3.6.1.2.1.1.5.0 เป็นการขอดูชื่อของผู้บริหารอุปกรณ์เพียงอย่างเดียวจึงสามารถใช้คำสั่ง Single เพื่อขอข้อมูลได้ ส่วนคำสั่ง Edit จะเป็นการแก้ไขข้อมูลของอุปกรณ์เครือข่ายซึ่งสามารถแก้ได้ครั้งละ 1 อ๊อปเจ็กต์เท่านั้นและการอ้างถึงโดยใช้หมายเลขอ๊อปเจ็กท์จำเป็นต้องเป็นอ้างแบบเฉพาะเจาะจงเช่น 1.3.6.1.2.1.1.5.0 เป็นต้น



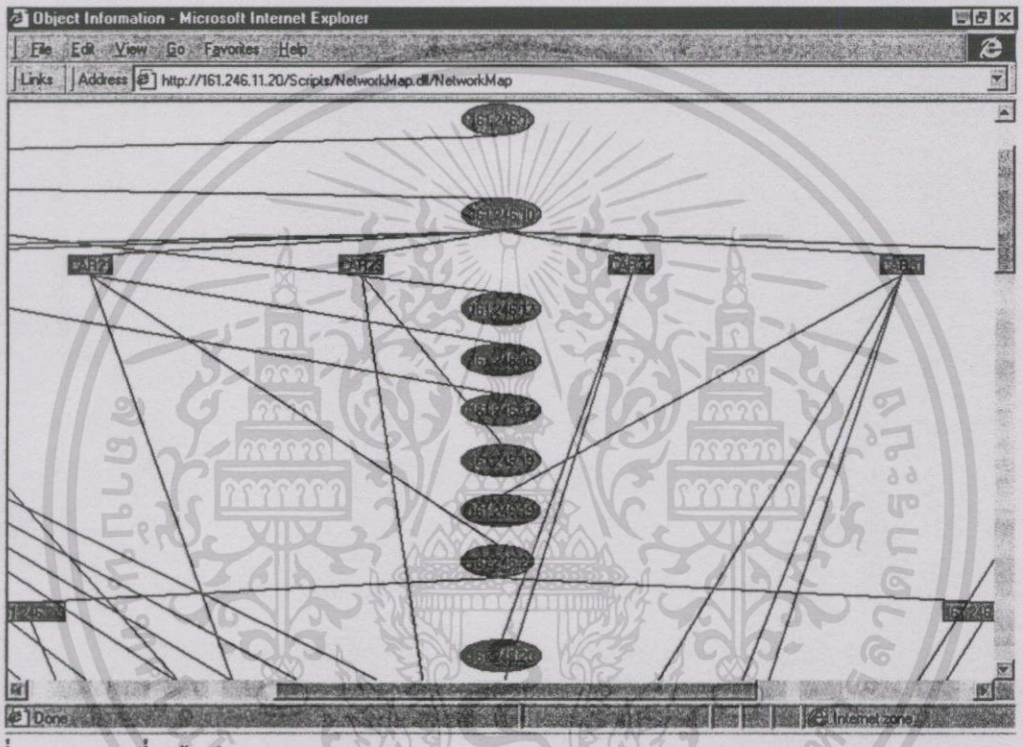
รูปที่ ก.17 หน้าจอของรายการ MIB Browser

3.2 เมื่อผู้ใช้เลือกรายการ Network Discovery ดังรูปที่ ก.18 ซึ่งเป็นการสั่งให้ระบบทำการสืบค้นข้อมูลของเราท์เตอร์ที่อยู่บนระบบเครือข่ายเพื่อนำมาสร้างเป็นแผนที่เครือข่าย ผู้ใช้จำเป็นที่จะต้องทราบเลขหมายไอพีของเราท์เตอร์เพื่อที่จะนำมาใช้เป็นโหนดเริ่มต้นของการสืบค้น โดยเราท์เตอร์ทั้งหมดที่สืบค้นได้จะเป็นเราท์เตอร์ที่มีชื่อคอมมิวนิตี้เหมือนกัน



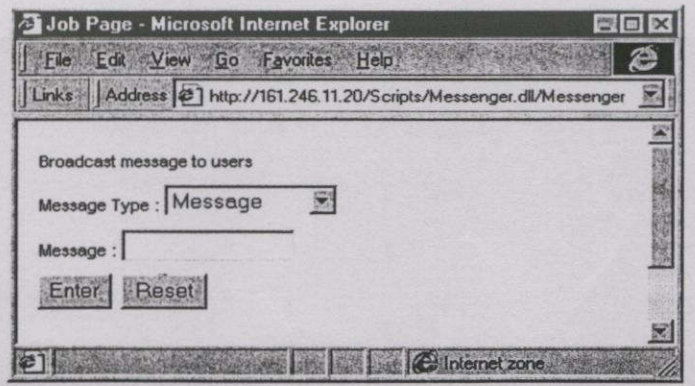
รูปที่ ก.18 หน้าจอของรายการ Network Discovery

3.3 เมื่อผู้ใช้ทำการเลือกรายการ Network Map ระบบจะให้บริการแผนที่เครือข่ายรวมทั้งข้อมูลของโหนดต่างที่ปรากฏอยู่บนแผนที่เครือข่าย แผนที่เครือข่ายดังกล่าวนี้เป็นผลลัพธ์ที่ได้มาจากการสืบค้นเครือข่ายของรายการ Network Discovery โหนดที่ปรากฏอยู่บนแผนที่จะแบ่งได้เป็น 2 ประเภทคือ เราท์เตอร์ และเครือข่ายย่อย ข้อมูลของเราท์เตอร์ที่ผู้ใช้สามารถขอดูจากระบบได้แก่ ชื่อ จำนวน และประเภทอินเตอร์เฟสของเราท์เตอร์ เป็นต้น ส่วนข้อมูลของเครือข่ายย่อยที่ผู้ใช้สามารถขอดูก็คือ เลขหมายไอพีของเราท์เตอร์ที่เชื่อมต่ออยู่เครือข่ายย่อย พิจารณารูปที่ ก.19



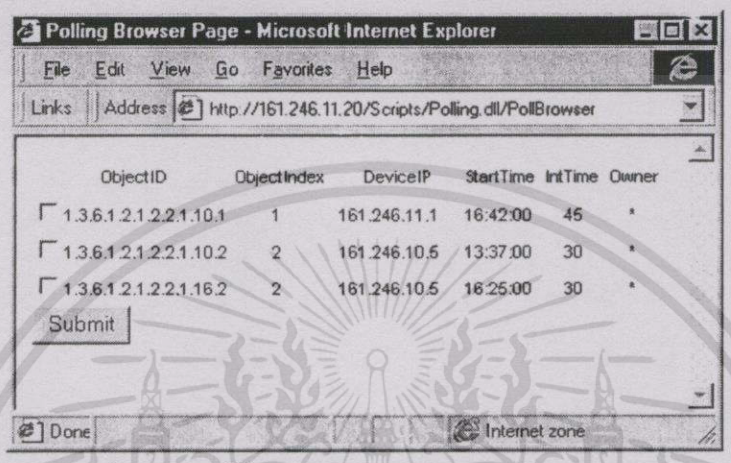
รูปที่ ก.19 แผนที่เครือข่ายของรายการ Network Map

3.4 ผู้ใช้สามารถเลือกรายการ New Message เพื่อใช้ในการส่งข้อความลงไปสู่แมสเสจบอร์ดเพื่อสื่อสารกับผู้ใช้คนอื่นๆที่อยู่บนระบบ ดังรูปที่ ก.20



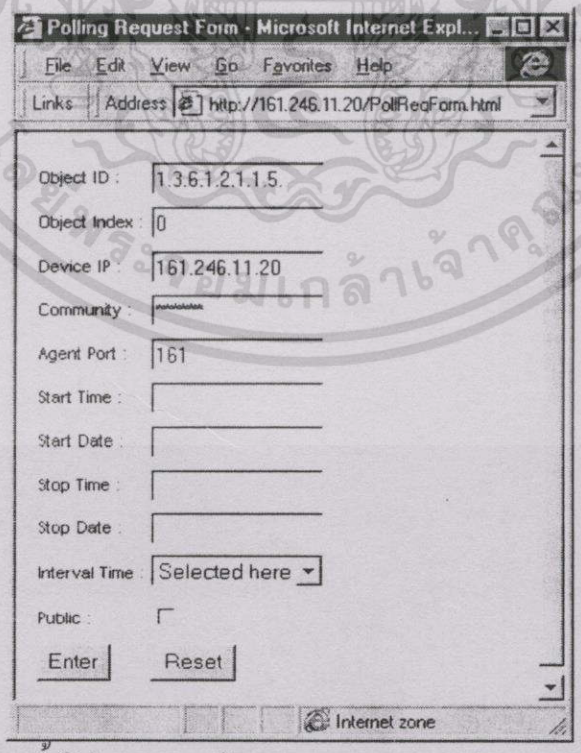
รูปที่ ก.20 หน้าจอของรายการ New Message

3.5 รายการ Poll Browser เป็นการขอดูรายการที่ผู้ใช้ได้ตั้งโปรแกรมขอให้ระบบทำการเก็บรวบรวมข้อมูลทางสถิติให้ ดังรูปที่ ก.21 นอกจากรายการของผู้ใช้แล้วยังมีรายการที่เป็นของผู้ใช้คนอื่นที่กำหนดค่าของรายการให้เป็นรายการสาธารณะซึ่งอนุญาตให้ผู้ใช้ที่ไม่ใช่เจ้าของสามารถขอดูข้อมูลที่ได้จากการโพลได้ ผลลัพธ์ของการโพลจะอยู่ในรูปของกราฟทางสถิติ



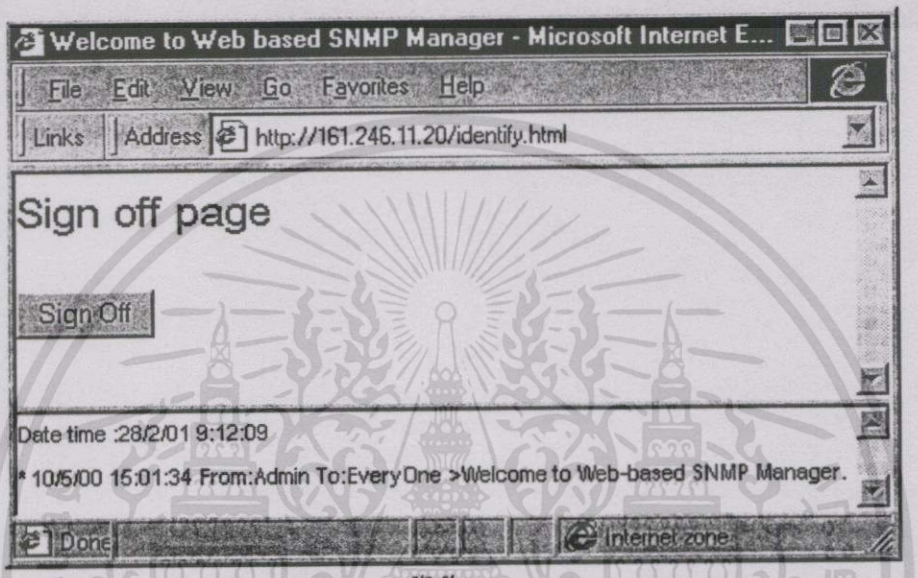
รูปที่ ก.21 หน้าจอของการโพลทางสถิติจากรายการ Poll Browser

3.6 รายการ Poll Request ทำงานร่วมกับ Poll Browser โดยเป็นฝ่ายรับรายการที่ผู้ใช้ต้องการให้มีการโพลส่งไปยังระบบให้ทำการเก็บเข้าสู่ตารางเวลา ดังรูปที่ ก.22



รูปที่ ก.22 หน้าจอของการตั้งโปรแกรมการโพลของรายการ Poll Request

3.7 เมื่อผู้ใช้ต้องการยุติการทำงานกับระบบ ผู้ใช้สามารถออกจากระบบได้โดยการเลือกรายการ Sign Off ซึ่งถือว่าเป็นการออกจากระบบที่ถูกวิธี เนื่องจากระบบจะขอให้เวบเบราเซอร์ทำการลบเซสชั่นคีย์ออกจากแฟ้มข้อมูลของเวบเบราเซอร์เอง เพื่อป้องกันการสวมรอยเข้าใช้บริการจากของผู้ใช้ที่ไม่มีสิทธิ ดังรูปที่ ก.23



รูปที่ ก.23 หน้าจอของการออกจากระบบของผู้ใช้ของรายการ Sign Off

จากรายละเอียดที่ได้กล่าวมาแล้วในข้างต้น ผู้ใช้จะสามารถใช้บริการของระบบเพื่อดูแลและตรวจสอบเครือข่าย พิจารณาโครงสร้างของระบบเครือข่ายจากแผนที่เครือข่ายได้ด้วยการทำงานผ่านระบบเวบทำให้เกิดความสะดวกในการทำงานเป็นอย่างดี

ภาคผนวก ข.
บทความที่ตีพิมพ์ในวารสาร

1. ธีรกฤษณ์ จันทเบญจมภัทร และอัครินทร์ คุณกิตติ. "การศึกษาองค์ประกอบของเครือข่ายที่มีผลต่อการสืบค้นเครือข่าย (A Study of Influential Factors to A Network Discovery)." สารสนเทศลาดกระบัง, ปีที่ 5, ฉบับที่ 1, 2543.

วารสาร

กรกฎาคม 2543 ปีที่ 5 ฉบับที่ 1

สารสนเทศลาดกระบัง

Ladkrabang Information Journal

ISSN 0859 - 5208

July 2000 Vol.5 No.1

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Faculty of Information Technology, King Mongkut's Institute of Technology, Ladkrabang Bangkok 10520

บทความวิจัย

บทความวิชาการ

บทความทั่วไป

การศึกษาองค์ประกอบของเครือข่ายที่มีผลต่อการสืบค้นเครือข่าย

A Study of Influential Factors to A Network Discovery

ธีรกฤษณ์ จันทเบญจม์ภัทร * อัครินทร์ คุณกิตติ** สุรสิทธิ์ วรรณไกรโรจน์**

บทคัดย่อ

การสืบค้นเครือข่ายเป็นหน้าที่ที่มีบทบาทสำคัญในโปรแกรมจัดการเครือข่าย โดยทุกครั้งที่มีการสืบค้นจะทำให้เกิดปริมาณข้อมูลขึ้นบนเครือข่ายซึ่งจะมีผลกระทบต่อประสิทธิภาพการทำงานของระบบเครือข่าย ในบทความนี้จะวิเคราะห์ว่ามีองค์ประกอบใดบ้างของระบบเครือข่ายที่เป็นตัวแปรสำคัญต่อปริมาณข้อมูลที่เกิดขึ้นจากแอลกอริทึมของการสืบค้นที่นำมาใช้ แอลกอริทึมจะทำการสืบค้นเครือข่ายโดยใช้โปรโตคอล SNMPv1 เพื่อรวบรวมหมายเลข IP และ Next hop จาก MIB-II ของเราท์เตอร์ในระบบ การวิเคราะห์จะอยู่ในรูปความสัมพันธ์ขององค์ประกอบที่เกี่ยวข้องในการสืบค้นกับค่าอย่างน้อยสุดของจำนวนแพ็กเก็ต ปริมาณข้อมูลในหน่วยไบต์ที่เกิดขึ้นบนระบบเครือข่าย และเวลาที่ใช้ไปขณะทำการสืบค้น ผลลัพธ์ที่ได้จากการวิเคราะห์จะถูกคำนวณและนำไปเปรียบเทียบกับผลลัพธ์ที่ได้จากการทดลอง จากการวิจัยพบว่าองค์ประกอบที่มีผลต่อปริมาณข้อมูลที่เกิดจากการสืบค้นคือ จำนวนเราท์เตอร์ จำนวนเครือข่ายย่อย และจำนวนอินเตอร์เฟสของเราท์เตอร์ในระบบเครือข่าย ในขณะที่ขีดความสามารถในการเข้าถึงและประมวลผลข้อมูลของฮาร์ดแวร์ทั้งในเราท์เตอร์และระบบจัดการเครือข่ายจะมีผลต่อเวลาที่ใช้ไปในการสืบค้นมากกว่าปริมาณข้อมูลที่เกิดจากการสืบค้นเครือข่าย

Abstract

Network discovery plays an important role of network management. Once it's in operation, it always produces data traffic that effect to network performance. In this paper, we will analyze a discovery algorithm and find relations of network parameters that affect to traffic. This algorithm collects information of IP interface and Next hop from routers' MIB-II, using SNMPv1 protocol. The analysis will show relations of minimum number of packets and bytes, minimum time to discover the network, and finally the analytic and experimental results are compared. The numbers of routers, sub-networks or interfaces, influence network traffic and discovery time, but the experimental results surprisingly show that the processing time of network management station itself is a significant time in discovery process.

1 บทนำ

หน้าที่สำคัญประการหนึ่งในการจัดการระบบเครือข่าย (Network management) คือการสืบค้นเครือข่าย (Network discovery) ซึ่งเป็นการค้นหาและรวบรวมข้อมูลขององค์ประกอบ (Parameter) ต่างๆที่เชื่อมต่ออยู่บนระบบเครือข่ายโดยโปรแกรมหรือระบบที่ใช้สำหรับจัดการเครือข่าย (Network management manager) หรือที่เรียกกันสั้นๆว่า manager โดยข้อมูลส่วนใหญ่จะถูกนำไปใช้ในการสร้างแผนที่เครือข่าย (Network map) ซึ่งมีประโยชน์มากสำหรับงานจัดการเครือข่าย เพราะนอกจากจะทำให้เจ้าหน้าที่ผู้ดูแลระบบเครือข่าย (Network administrator) สามารถมองเห็นและเข้าใจความสัมพันธ์ระหว่างองค์ประกอบต่างๆ

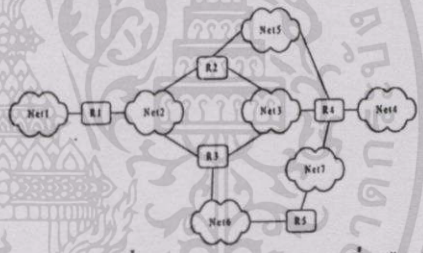
* นักศึกษาปริญญาโท หลักสูตรวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

** อาจารย์ประจำคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บนเครือข่ายได้ดียิ่งขึ้นแล้วยังมีประโยชน์กับงานส่วนอื่นอีกเช่น งานด้านการจัดการเครือข่ายแบบรูปลักษณ์ (Configuration management)[1] ซึ่งได้แก่ การแสดงสถานะภาพ (Status) ขององค์ประกอบต่างๆขณะระบบเครือข่ายกำลังปฏิบัติงานอยู่ เป็นต้น

จากการศึกษาทฤษฎีการจัดการเครือข่ายและโปรโตคอล SNMP จากเอกสารต่างๆ เช่น [2] และ [3] รวมถึงงานวิจัยในบทความของ Glenn Mansfield [4] ซึ่งกล่าวได้ถึงวิธีการและข้อมูลที่สามารถนำมาใช้ในการสืบค้นเพื่อสร้างแผนที่เครือข่าย แต่ไม่ได้กล่าวถึงปัจจัยหรือองค์ประกอบที่มีผลกระทบต่อการสืบค้น ดังนั้นงานวิจัยในบทความนี้จะวิเคราะห์ว่ามีองค์ประกอบใดบ้างที่เป็นตัวแปรสำคัญต่อปริมาณข้อมูลที่เกิดขึ้นจากการสืบค้นเครือข่าย เนื่องจากปริมาณข้อมูลที่เกิดขึ้นนี้จะมีผลต่อประสิทธิภาพการทำงานของระบบเครือข่าย ยิ่งระบบเครือข่ายมีองค์ประกอบอยู่มากเท่าไรปริมาณข้อมูลที่เกิดขึ้นก็ย่อมมีมากขึ้นเท่านั้น เพื่อเป็นประโยชน์ต่อไปในการออกแบบระบบเครือข่าย เช่น การหาตำแหน่งที่ตั้งที่มีแบนด์วิดท์ (Bandwidth) ที่เหมาะสมบนระบบเครือข่ายสำหรับ manager เป็นต้น โดยการวิเคราะห์จะอยู่ในรูปของความสัมพันธ์ทางคณิตศาสตร์ของปริมาณข้อมูลที่เกิดขึ้นที่ผลลัพธ์มีหน่วยเป็นจำนวนแพ็กเก็ต (Packet) จำนวนไบต์ (Byte) และความสัมพันธ์ของเวลาที่ถูกใช้ไปในการสืบค้น ความสัมพันธ์ทั้งสามรูปแบบที่ได้จากการวิเคราะห์ในบทความนี้จะสามารถนำไปใช้ในการประมาณค่าน้อยสุด (Minimum) ของปริมาณข้อมูลที่คาดว่าน่าจะเกิดขึ้นบนระบบเครือข่ายและเวลาที่ใช้ไปในสภาวะแวดล้อมที่แตกต่างกันออกไป โดยอ้างอิงจากวิธีการสืบค้นที่นำมาใช้ ซึ่งลักษณะการทำงานเป็นการค้นหาและรวบรวมข้อมูลของเราท์เตอร์และเครือข่ายย่อยทั้งหมดที่มีอยู่ในระบบเพื่อนำข้อมูลที่ได้มาใช้ในการสร้างแผนที่เครือข่าย ผลลัพธ์ของการคำนวณที่ได้จากความสัมพันธ์จะถูกนำไปเปรียบเทียบกับผลลัพธ์ที่ได้จากการทดลองเพื่อยืนยันความถูกต้องของสมการที่ได้จากการวิเคราะห์

บทความนี้จะเริ่มด้วยการอธิบายวิธีการและขั้นตอนการทำงานของการสืบค้นที่นำมาใช้ จากนั้นนำวิธีการดังกล่าวมาวิเคราะห์หาความสัมพันธ์ของจำนวนแพ็กเก็ตและจำนวนไบต์ของข้อมูลที่เกิดขึ้นบนระบบเครือข่ายและวิเคราะห์เวลาที่ถูกใช้ไปในการสืบค้น จากนั้นนำผลลัพธ์ที่ได้จากการคำนวณไปเปรียบเทียบกับผลลัพธ์ที่ได้จากการทดลอง และเราจะจบบทความนี้ด้วยการสรุปข้อมูลที่ได้จากการวิเคราะห์และทดลอง



รูปที่ 1 แสดงภาพของแผนที่เครือข่าย

2 แอลกอริทึมของการสืบค้น

การสืบค้นที่ใช้อ้างอิงในบทความนี้จะทำงานโดยการค้นหาและรวบรวมข้อมูลของเราท์เตอร์และเครือข่ายย่อยทั้งหมดที่มีอยู่ในระบบเพื่อนำมาใช้ในการสร้างแผนที่เครือข่ายซึ่งจะอยู่ในรูปของการเชื่อมต่อระหว่างเครือข่ายย่อยกับเราท์เตอร์ ดังรูปที่ 1 ซึ่งโปรโตคอลที่ใช้ในการทำงานคือ SNMPv1 (Simple Network Management Protocol version 1)[2] เพื่อเข้าถึงข้อมูลที่อยู่ใน MIB-II บนเราท์เตอร์ซึ่งกระจายอยู่บนระบบเครือข่าย และเนื่องจากแผนที่เครือข่ายที่ถูกสร้างจะอยู่ในรูปของการเชื่อมต่อระหว่างเครือข่ายย่อยกับเราท์เตอร์ ทำให้ข้อมูลของเราท์เตอร์ที่นำมาใช้จะต้องสามารถบอกให้ทราบได้ว่าเราท์เตอร์ได้เชื่อมต่ออยู่กับเครือข่ายย่อยใดบ้างและ

ต้องทำให้การสืบค้นข้อมูลดังกล่าวสามารถขยายออกไปได้ครอบคลุมทุกเครือข่ายย่อยในระบบ ข้อมูลที่ถูกเลือกมาใช้ในการสืบค้นจึงมีดังต่อไปนี้ :-

1. IP-interface เป็นข้อมูลที่เป็นหมายเลข IP (IP address) ของอินเตอร์เฟส และ net mask ที่อยู่ในตาราง IP อินเตอร์เฟส (IP interface table) ของเราท์เตอร์ และเมื่อนำหมายเลข IP และ net mask ของมันมาคำนวณ [5] จะทำให้ได้หมายเลข IP ของเครือข่ายย่อยที่เราท์เตอร์ได้เชื่อมต่ออยู่ ข้อมูลทั้งสองชนิดนี้จะอยู่ใน ipAdEntAddr และ ipAdEntNetMask ซึ่งเป็นตัวแปรที่อยู่ภายใต้ตาราง ipAddrTable [6] บน MIB-II ตามลำดับ

2. Next-hop เป็นข้อมูลของหมายเลข IP ที่เป็น next hop ที่อยู่ในตารางเราท์ติ้ง (Routing table) ของเราท์เตอร์ โดยหมายเลข IP เหล่านี้ก็คืออินเตอร์เฟสของเราท์เตอร์ตัวอื่นๆที่อยู่ในระบบเครือข่ายนั่นเอง ทำให้สามารถใช้หมายเลข IP เหล่านี้ขยายการสืบค้นไปทั่วทั้งระบบเครือข่ายได้ ข้อมูลชนิดนี้จะอยู่ใน ipRouteNextHop ซึ่งเป็นตัวแปรที่อยู่ภายใต้ตาราง ipRouteTable [6] บน MIB-II

การทำงานจะเริ่มจากสีดเราท์เตอร์ (Seed router) คือเราท์เตอร์ที่ถูกกำหนดมาให้เป็นจุดเริ่มต้นของการทำงาน ขั้นตอนการทำงานมีดังต่อไปนี้

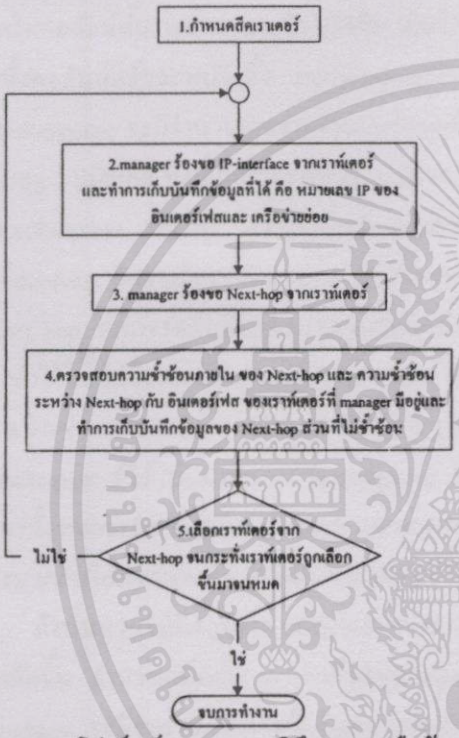
1. กำหนดสีดเราท์เตอร์ให้กับ manager เพื่อเป็นจุดเริ่มต้นของการสืบค้น

2. จากนั้น manager จะทำการร้องขอข้อมูลของ IP-interface จากเราท์เตอร์ เพื่อนำข้อมูลเหล่านี้มาคำนวณหาเครือข่ายย่อยที่กำลังเชื่อมต่ออยู่กับเราท์เตอร์จากที่ได้กล่าวมาแล้วในเบื้องต้น ดังนั้นเมื่อเสร็จสิ้นกระบวนการในขั้นตอนนี้ manager จะทราบว่าเราท์เตอร์ประกอบไปด้วยอินเตอร์เฟสที่มีหมายเลข IP อะไรบ้างและกำลังเชื่อมต่ออยู่เครือข่ายย่อยใด

3. ถัดมา manager จะทำการร้องขอข้อมูลจากเราท์เตอร์ตัวเดิม โดยในคราวนี้ข้อมูลที่ต้องการคือ Next-hop ซึ่งจะทำให้ manager ทราบเพิ่มเติมว่ามีเราท์เตอร์ใดอยู่ในระบบอีกบ้าง จากนั้นจะทำการเก็บหมายเลข IP ของเราท์เตอร์เหล่านี้ไว้สำหรับร้องขอข้อมูลในส่วนของ IP-interface ในโอกาสต่อไป เมื่อมาถึงจุดนี้เท่ากับว่า manager ได้รับข้อมูลทั้งหมดที่ต้องการจากเราท์เตอร์ที่ได้รับมาในเบื้องต้นครบแล้ว

4. เมื่อพิจารณาให้ดีจะพบว่ากระบวนการในข้อ 3. คือส่วนสำคัญที่ทำให้เกิดการสืบค้นข้อมูลที่ขยายออกไปเป็นทอดๆ จากเราท์เตอร์หนึ่งไปยังเราท์เตอร์อีกตัวหนึ่ง จึงจำเป็นอย่างยิ่งที่ต้องมีขั้นตอนการตรวจสอบความซ้ำซ้อน โดยเราท์เตอร์ที่เคยถูก manager ร้องขอข้อมูลไปแล้ว (สมมุติว่าการร้องขอข้อมูลในแต่ละครั้งประสบความสำเร็จคือไม่มีความผิดพลาดที่ก่อให้เกิดเหตุการณ์ที่ manager ต้องเก็บเราท์เตอร์ตัวนี้ไว้เพื่อร้องขอข้อมูลในโอกาสถัดไป) จะต้องไม่ถูกนำกลับมาใช้ในการร้องขอข้อมูลชนิดนี้อีก ทั้งนี้เพื่อป้องกันมิให้การวนซ้ำที่ไม่มีจุดสิ้นสุดเกิดขึ้น ดังนั้นในขั้นตอนนี้จึงต้องนำหมายเลข IP ของ Next-hop ที่ได้มาตรวจสอบความซ้ำซ้อนกับหมายเลข IP ทั้งหมดของ IP-interface ที่ manager มีอยู่ แล้วคัดเอาเฉพาะข้อมูลของหมายเลข IP ของ Next-hop ที่ไม่ซ้ำซ้อนเก็บไว้ อย่างไรก็ตามขั้นตอนการตรวจสอบก็ยังไม่สิ้นสุดที่จุดนี้ ยังต้องมีการตรวจสอบความซ้ำซ้อนภายในระหว่างหมายเลข IP ของ Next-hop ด้วยกันอีกครั้งหนึ่ง สาเหตุที่ต้องทำเช่นนี้เนื่องจากโดยปกติแล้วเราท์เตอร์จะใช้หมายเลข IP หนึ่งๆใน Next-hop เป็นช่องทางออกไปยังเครือข่ายปลายทาง (Destination network) ได้มากกว่า 1 เครือข่าย ทำให้ข้อมูลของ Next-hop มักจะมีหมายเลขซ้ำกันเกิดขึ้น ดังนั้นจึงต้องมีการตัดส่วนของหมายเลข IP ที่ซ้ำออกไป

5. เลือกเราท์เตอร์จากหมายเลข IP ของ Next-hop ที่ manager มีอยู่ มาใช้ในการทำงานต่อ โดยทำซ้ำในข้อ 2.-4. จนกระทั่งเราท์เตอร์ที่มีอยู่ถูกนำมาใช้ในการทำงานจนหมดจึงจบการทำงาน



รูปที่ 2 แสดงการโฟลว์ชาร์ทของแอลกอริทึมของการสืบค้น

รูปที่ 2 เป็นภาพแสดงการทำงานของแอลกอริทึมในรูปแบบของโฟลว์ชาร์ท (Flow chart) จากแอลกอริทึมที่ได้กล่าวมาแล้วนี้ จะสังเกตุได้ว่าการทำงานดังกล่าวจะมีจุดสำคัญอยู่ที่ขั้นตอนการตรวจสอบความซ้ำซ้อน ซึ่งเป็นตัวป้องกันไม่ให้การทำงานกลายเป็นการวนซ้ำโดยปราศจากการสิ้นสุดและมีผลให้ manager รับเราท์เตอร์แต่ละตัวเพื่อนำไปใช้ในการร้องขอชุดข้อมูล IP-interface และ Next-hop เพียงตัวละครั้งเท่านั้น คุณสมบัตินี้เองที่จะถูกนำไปใช้ในการวิเคราะห์ซึ่งอยู่ในหัวข้อถัดไป

3 การวิเคราะห์

การวิเคราะห์ในบทความนี้จะเริ่มจากจำนวนแพ็กเก็ตที่เกิดขึ้นขณะทำการสืบค้นและนำผลที่ได้มาวิเคราะห์ปริมาณข้อมูลที่เกิดขึ้นในหน่วยไบต์ ต่อจากนั้นจะนำปริมาณข้อมูลที่ได้ไปวิเคราะห์เวลาที่จะถูกใช้ไปในการรวบรวมข้อมูลขณะทำการสืบค้น การวิเคราะห์ทั้งหมดจะอยู่ภายใต้เงื่อนไขคือ กำหนดให้การรับส่งข้อมูลขณะทำการสืบค้นปราศจากความผิดพลาด

3.1 จำนวนแพ็กเก็ต

เพื่อความเข้าใจที่ดีขึ้น จะขอกล่าวถึงชนิดและคุณสมบัติของคำร้องขอข้อมูล[2] (SNMP Protocol Data Unit) ของโปรโตคอล SNMPv1 ที่ manager ใช้ไปในการร้องขอข้อมูลที่ต้องการจากเราท์เตอร์ในส่วนที่เกี่ยวข้องกับแอลกอริทึมของสืบค้นที่ใช้อ้างอิงในบทความนี้

1. GetRequest ถูกใช้เพื่อร้องขอข้อมูลทั้งที่อยู่และไม่อยู่ภายใต้ตารางใดๆใน MIB-II โดยหากต้องการเข้าถึงข้อมูลที่อยู่ภายใต้ตารางนั้นจะต้องทราบดรรชนีที่จะใช้ในการเข้าถึงข้อมูลตัวนั้นล่วงหน้า ในการร้องขอข้อมูลด้วย GetRequest 1 ครั้งหรือ 1 แพ็กเก็ต จะได้รับคำตอบกลับมาด้วย GetResponse 1 คำตอบหรือ 1 แพ็กเก็ต ยกตัวอย่างเช่น การใช้ GetRequest ด้วยดรรชนีคือ 0.0.0.0 ในการร้องขอหมายเลข IP ของเราท์เตอร์ที่เป็นค่าโดยปริยายของเกตเวย์ (Default gateway) ในตารางเราท์ติ้งของเราท์เตอร์ตัวหนึ่ง

2. GetNextRequest ใช้สำหรับร้องขอข้อมูลที่อยู่ถัดไปจากข้อมูลที่ดรรชนีอ้างอิงถึงใน MIB-II และด้วยกลไกการทำงานของ SNMPv1 ทำให้สามารถใช้ GetNextRequest ในการร้องขอข้อมูลที่อยู่ในคอลัมน์ต่างๆของตารางใน MIB-II ได้โดยไม่ต้องทราบดรรชนีล่วงหน้า โดยทั่วไปมักถูกใช้สำหรับร้องขอข้อมูลทั้งคอลัมน์ในตาราง ในการส่ง GetNextRequest เพื่อร้องขอข้อมูลแต่ละครั้งจะได้รับคำตอบกลับมาเป็น GetResponse 1 คำตอบหรือ 1

แพ็กเก็ตเช่นเดียวกับ GetRequest ดังนั้นหากต้องการข้อมูลทั้งหมดในคอลัมน์ใดๆ จำนวนแพ็กเก็ตของ GetNextRequest ที่ใช้ไปกับจำนวนแพ็กเก็ตของ GetResponse ที่ได้รับควรจะมีค่าเท่ากับจำนวนข้อมูลทั้งหมดในคอลัมน์นั้น แต่ในความเป็นจริงคือ เมื่อได้ข้อมูลทั้งคอลัมน์แล้วจะพบว่าทั้ง GetNextRequest และ GetResponse จะมีจำนวนเกินจากจำนวนข้อมูลที่มีอยู่จริงอยู่ 1 แพ็กเก็ต ทั้งนี้เนื่องมาจากคุณสมบัติของ GetNextRequest นั่นเอง ตัวอย่างเช่น การใช้ GetNextRequest ในการร้องขอหมายเลข IP ทั้งหมดของ next hop ในตารางเราท์ติ้งของเราท์เตอร์ตัวหนึ่งสามารถทำได้โดยไม่ต้องทราบค่าดรรชนีคือ หมายเลข IP ของเครือข่ายปลายทาง ในกรณีนี้จำนวน GetNextRequest ที่ใช้ไปและจำนวน GetResponse ที่ได้รับมาทั้งหมดจะมีค่าเท่ากับจำนวนแถวของตารางเราท์ติ้งบวกกับโอเวอร์เฮด (Overhead) ที่เกินมา 1 ครั้ง

ดังนั้นหากไม่มีความผิดพลาดในการรับส่งข้อมูลเกิดขึ้น จำนวนแพ็กเก็ตของการร้องขอข้อมูลของ manager จะมีจำนวนเท่ากับแพ็กเก็ตของข้อมูลที่ manager ได้รับจากเราท์เตอร์ การวิเคราะห์จำนวนแพ็กเก็ตที่เกิดจากการสืบค้นต่อไปนี้จะทำโดยใช้จำนวนแพ็กเก็ตที่เกิดจากการร้องขอข้อมูลของ manager มาพิจารณาและเพื่อง่ายต่อการเข้าใจการวิเคราะห์จะดำเนินตามแอลกอริทึมของการสืบค้นที่ได้กล่าวถึงไว้ก่อนหน้านี้แล้วในหัวข้อที่ 2

เมื่อ manager ได้เลือกเราท์เตอร์ขึ้นมา โดยในที่นี้ขอกำหนดให้เป็นเราท์เตอร์ตัวที่ r เมื่อ r คือ $1,2,3,\ldots,N_R$ และ N_R คือจำนวนเราท์เตอร์ทั้งหมดที่มีอยู่ในระบบ ข้อมูลส่วนแรกที่ต้องการจากเราท์เตอร์คือ IP-interface ประกอบด้วย หมายเลข IP และ net mask ซึ่งอยู่ในตาราง IP อินเตอร์เฟส โดยมีหมายเลข IP เป็นข้อมูลที่ต้องเป็นดรรชนี แต่เนื่องจากไม่ทราบค่าของดรรชนีในตารางหรือกล่าวอีกนัยหนึ่งว่าไม่ทราบว่าเราท์เตอร์ r มีหมายเลข IP ใดบ้าง ดังนั้นในการร้องขอหมายเลข IP จึงต้องใช้ GetNextRequest ทำให้จำนวนแพ็กเก็ตในการร้องขอข้อมูลจึงเท่ากับจำนวนหมายเลข IP ทั้งหมดที่เป็นอินเตอร์เฟสของเราท์เตอร์ (โดยนับรวมหมายเลข IP ของลูปแบ็ค (Loop back) คือ 127.0.0.1 ด้วย) รวมกับโอเวอร์เฮดที่เกินมา 1 แพ็กเก็ต โดยกำหนดให้ $N_{IP}(r)$ เป็นจำนวนหมายเลข IP ที่เป็นอินเตอร์เฟสของเราท์เตอร์ r สำหรับการร้องขอ net mask จึงสามารถใช้หมายเลข IP ที่ได้มาเป็นดรรชนีในการร้องขอข้อมูลด้วย GetRequest ยกเว้นดรรชนีที่ไม่สนใจซึ่งจะไม่ถูกใช้ในการร้องขอข้อมูล ในที่นี้คือ หมายเลข IP ของลูปแบ็ค โดย $N_{IPOther}(r)$ เป็นจำนวนดรรชนีที่ไม่ถูกใช้ในการร้องขอข้อมูล หลังจากที่ได้ข้อมูลของ IP-interface ที่ต้องการทั้งหมด จำนวนแพ็กเก็ตที่ถูกใช้ไปคือ

$$N_{getIP}(r) = N_{IP}(r) + 1 \quad \ldots\ldots(1)$$

$$N_{getNM}(r) = N_{IP}(r) - N_{IPOther}(r) \quad \ldots\ldots(2)$$

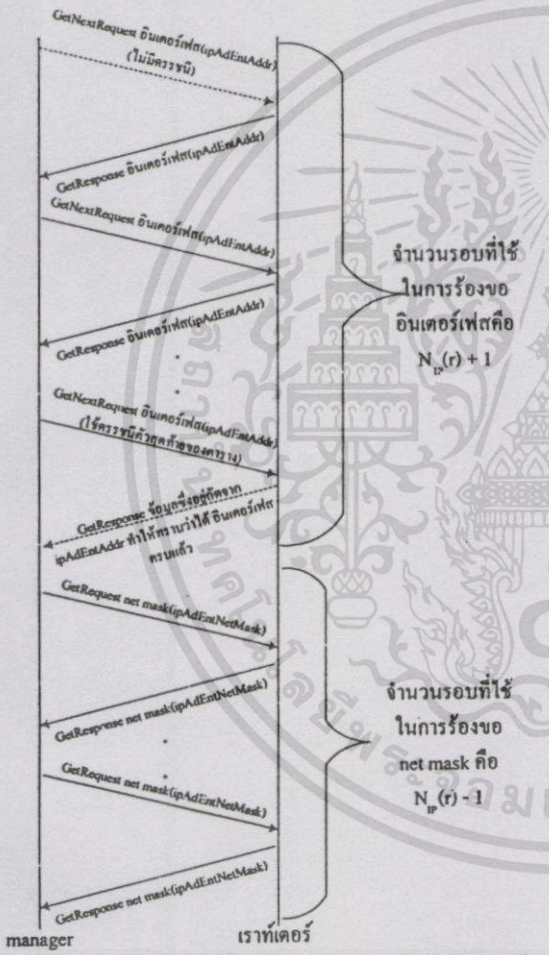
โดยที่ $N_{getIP}(r)$, $N_{getNM}(r)$ คือจำนวนแพ็กเก็ตที่ manager ใช้ไปในการร้องขอเพื่อรับข้อมูลของหมายเลข IP และ net mask ตามลำดับ เนื่องจาก $N_{IPOther}(r)$ ในบทความนี้หมายถึง หมายเลข IP ที่เป็นลูปแบ็คเพียงอย่างเดียวทำให้ $N_{IPOther}(r)$ มีค่าเท่ากับ 1 ดังนั้นเมื่อนำสมการที่ (1) และ (2) มาหาจำนวนแพ็กเก็ตที่ใช้ไปในการร้องขอข้อมูลของเราท์เตอร์ r จะได้

$$N_{reqIP}(r) = N_{getIP}(r) + N_{getNM}(r) \quad \ldots\ldots(3)$$

$$N_{reqIP}(r) = 2N_{IP}(r) \quad \ldots\ldots(4)$$

$N_{reqIP}(r)$ คือจำนวนแพ็กเก็ตที่ manager ใช้ไปในการร้องขอเพื่อรับข้อมูลของ IP-interface ของเราท์เตอร์ r รูปที่ 3 จะเป็นภาพแสดงการร้องขอข้อมูล

IP-interface ของ manager จากแต่ละเราท์เตอร์ โดยส่วนที่เป็น GetNextRequest ของอินเตอร์เฟส (ipAdEntAddr) ที่ไม่มีครรชนีและ GetResponse ที่ข้อมูลของมันบอก manager ให้ทราบว่าได้อินเตอร์เฟสครบแล้วจะถือเป็นส่วนของโอเวอร์เฮด



รูปที่ 3 แสดงการร้องขอข้อมูล IP-interface แต่ละเราท์เตอร์

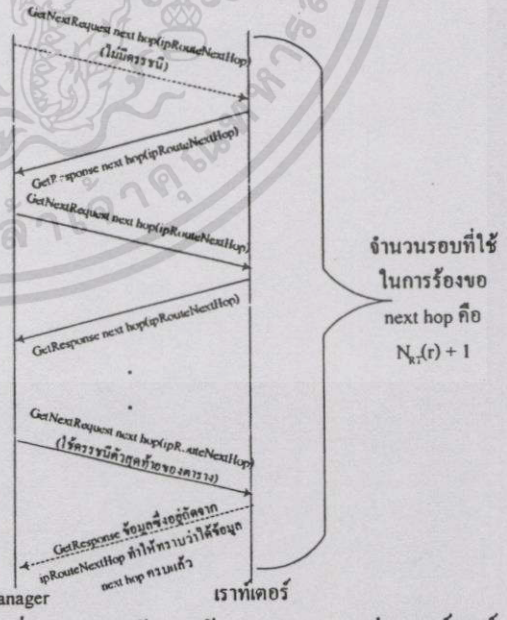
ต่อมา manager จะทำการร้องขอข้อมูล Next-hop จากเราท์เตอร์ตัวเดิม ซึ่งจะเป็นกรณีเดียวกับตัวอย่างในการอธิบายถึง GetNextRequest ที่ผ่านมา ดังนั้น

$$N_{reqNH}(r) = N_{RT}(r) + 1 \quad \text{.....(5)}$$

$$N_{RT}(r) = N_N + N_{OtherRow}(r) \quad \text{.....(6)}$$

$$N_{reqNH}(r) = N_N + N_{OtherRow}(r) + 1 \quad \text{.....(7)}$$

โดยที่ $N_{reqNH}(r)$ คือจำนวนแพ็กเก็ตที่ manager ใช้ไปในการร้องขอข้อมูล Next-hop ของเราท์เตอร์ r ในขณะที่ N_{RT} คือจำนวนแถวของตารางเราท์ติ้ง N_N คือจำนวนเครือข่ายย่อยทั้งหมดที่มีอยู่บนระบบก็ได้ $N_{OtherRow}(r)$ คือจำนวนแถวอื่นๆที่หมายเลข IP ของเครือข่ายปลายทาง ไม่ใช่หมายเลข IP ของเครือข่ายย่อย ยกตัวอย่างเช่น 0.0.0.0 127.0.0.1 224.0.0.0 255.255.255.255 และหมายเลข IP ที่เป็นอินเตอร์เฟสของตัวเราท์เตอร์เอง ซึ่งจะพบได้ใน WindowsNT เป็นต้น รูปที่ 4 เป็นภาพแสดงการร้องขอข้อมูล Next-hop ของ manger จากแต่ละเราท์เตอร์ โดยส่วนที่เป็น GetNextRequest ของ next hop (ipRouteNextHop) ที่ไม่มีครรชนีและ GetResponse ที่ข้อมูลของมันบอก manager ให้ทราบว่าได้ next hop ครบแล้วจะถือเป็นส่วนของโอเวอร์เฮด



รูปที่ 4 แสดงการร้องขอข้อมูล Next-hop แต่ละเราท์เตอร์

เมื่อมาถึงจุดนี้ manager จะได้ข้อมูลที่ต้องการทั้งหมดจากเราท์เตอร์ r ดังนั้น

$$N_{req}(r) = N_{reqIP}(r) + N_{reqNH}(r) + N_{Other}(r) \quad \ldots\ldots(8)$$

$$N_{req}(r) = 2N_{IP} + N_N + N_{OtherRow}(r) + N_{Other}(r) + 1 \quad \ldots.(9)$$

โดยที่ $N_{req}(r)$ คือจำนวนแพ็กเก็ตที่ manager ใช้ไปในการร้องขอข้อมูลทั้งหมดของเราท์เตอร์ r ส่วน $N_{Other}(r)$ เป็นข้อมูลเพิ่มเติมอื่นๆที่ต้องการนำมาใช้เพื่อเพิ่มความสมบูรณ์ให้กับการสืบค้น เช่น ชื่อของเราท์เตอร์ (System name) จำนวนเน็ทเวิร์คการ์ดของเราท์เตอร์ และชื่อผู้ดูแลรับผิดชอบเครื่อง เป็นต้น แต่ถ้าไม่มี $N_{Other}(r)$ จะมีค่าเป็นศูนย์

ดังนั้นเมื่อเราท์เตอร์ในระบบถูกเลือกขึ้นมาเพื่อใช้ในการร้องขอข้อมูลจนหมด จะได้

$$N_{req} = \sum_{r=1}^{N_R} [2N_{IP} + N_N + N_{OtherRow}(r) + N_{Other}(r) + 1] \quad \ldots(10)$$

โดยที่ N_{req} คือจำนวนแพ็กเก็ตที่ manager ใช้ไปในการร้องขอข้อมูลจากเราท์เตอร์ทุกตัวในระบบ ดังนั้นสมการ (10) จะเป็นสมการที่สามารถนำไปใช้ในการคำนวนหาจำนวนแพ็กเก็ตที่เกิดขึ้นจากการสืบค้นได้ โดยผลลัพธ์ที่ได้จะสามารถใช้บอกได้ทั้งจำนวนแพ็กเก็ตที่เกิดจากการร้องขอข้อมูลของ manager และจำนวนแพ็กเก็ตที่ manager ได้รับจากเราท์เตอร์ทั้งหมดในระบบ พารามิเตอร์บางตัวเช่น $N_{OtherRow}(r)$ ที่ปรากฏอยู่ในสมการในความเป็นจริงแล้วเป็นส่วนเติมเต็มให้ผลลัพธ์ที่จะเกิดขึ้นจากการคำนวณถูกต้องมากยิ่งขึ้นเท่านั้น ไม่มีการปรับแต่งโดยเพิ่มจำนวนเราท์ในตารางเราท์ติ้งของเราท์เตอร์แต่ละตัวมากนักก็สามารถที่จะละเลยตัวแปรตัวนี้ไปได้ ในตารางเราท์ติ้งของเราท์เตอร์โดยทั่วไปจะมีจำนวนแถวประเภทนี้ไม่มากนัก แต่จะมีเราท์เตอร์ที่ใช้ระบบปฏิบัติการบางระบบเช่น WindowsNT ตามที่ได้เคยยกตัวอย่างไปแล้วที่ $N_{OtherRow}(r)$ นอกจากจะประกอบด้วยจำนวนแถวอื่นๆแล้วยังประกอบด้วยหมายเลข IP ของเครือข่ายปลายทางที่เป็นหมายเลข IP ที่เป็นอินเตอร์เฟสของตัวเราท์เตอร์เองด้วย ตัวอย่างเช่น เราท์เตอร์ที่มีหมายเลข IP เป็น 161.246.253.1 และ 161.246.254.1 จะมีหมายเลข IP ทั้งสองนี้ปรากฏเป็นหมายเลข IP ของเครือข่ายปลายทางอยู่ในตารางเราท์ติ้งด้วย ซึ่งจะทำให้จำนวนแถวอื่นๆ ในตารางเราท์ติ้งเพิ่มขึ้นเมื่อจำนวนเครือข่ายย่อยที่เราท์เตอร์เชื่อมต่อได้เพิ่มขึ้นด้วย แต่อย่างไรก็ตามระบบปฏิบัติการที่ได้รับความนิยมเช่น NetWare ของ Novell หรือ IOS (Internetworking Operating System) ของ Cisco จะไม่มีลักษณะเช่นที่พบใน WindowsNT สมการของจำนวนแพ็กเก็ตที่ได้จากการวิเคราะห์ในหัวข้อนี้จะถูกนำไปใช้ในการวิเคราะห์ปริมาณข้อมูลในหน่วยไบต์ที่เกิดขึ้นในหัวข้อถัดไป

3.2 ปริมาณข้อมูลในหน่วยไบต์

การวิเคราะห์ปริมาณข้อมูลในหัวข้อนี้จะพิจารณาจากขนาดของแพ็กเก็ต โดยแพ็กเก็ตที่เกิดจากการสืบค้นจะมีขนาดที่แตกต่างกันขึ้นอยู่โปรโตคอลที่อยู่ภายในแพ็กเก็ต โดยในบทความนี้จะวิเคราะห์เฉพาะโปรโตคอล Ethernet ดังนั้นแพ็กเก็ตที่เกิดขึ้นจะเป็น Ethernet PDU[7] โดยแต่ละแพ็กเก็ตจะประกอบด้วย Ethernet header TCP/IP header และส่วนที่เป็นข้อมูลคือ SNMP PDU โดยส่วนที่เป็น Ethernet header และ TCP/IP header ซึ่งประกอบไปด้วย UDP header และ IP header จะมีค่าคงที่ คือ Ethernet header และ trailer ซึ่งมีขนาดเท่ากับ 26 ไบต์ IP header เท่ากับ 20 ไบต์ UDP header เท่ากับ 8 ไบต์ ดังนั้น header ของแพ็กเก็ตมีค่าเท่ากับ 54 ไบต์ ในขณะที่ขนาดของ SNMP PDU จะมีค่าไม่คงที่ขึ้นอยู่กับขนาดและชนิดของข้อมูลใน PDU ต่อไปนี้เป็นตารางแสดงขนาดน้อยสุดของ SNMP PDU และขนาด Ethernet PDU ของแพ็กเก็ตที่เกิดจากการร้องขอข้อมูลของ manger (Request

PDU) และแพ็กเก็ตที่ manager ได้รับจากเราท์เตอร์ (Response PDU) โดยขนาดของ SNMP PDU แต่ละชนิดในตารางได้มาจากการประมาณค่าน้อยสุดของแต่ละฟิลด์ (Field) ที่อยู่ใน SNMP PDU โดยอ้างอิงจาก ASN.1 (Abstract Syntax Notation 1) และ รูปแบบ (Format) ของ SNMP[2]

ตารางที่ 1 แสดงขนาดน้อยสุดของแพ็กเก็ตชนิดต่างๆ ของ IP-interface

IP-interface	Request PDU		Response PDU	
	SNMP	Ethernet	SNMP	Ethernet
หมายเลข IP	39	93	43	97
net mask	39	93	43	97
โอเวอร์เฮด	35	89	40	94

ตารางที่ 2 แสดงขนาดน้อยสุดของแพ็กเก็ตชนิดต่างๆ ของ Next-hop

Next-hop	Request PDU		Response PDU	
	SNMP	Ethernet	SNMP	Ethernet
next hop	39	93	43	97
โอเวอร์เฮด	35	89	40	94
ส่วนเพิ่มเติม	34	88	34	88

การวิเคราะห์จะพิจารณาจากการร้องขอข้อมูลของ IP-interface และ Next-hop ที่ได้เคยกล่าวมาแล้วในหัวข้อ 3.1 จากสมการ (1) (2) และ (3) จะได้สมการ (11) และ (12) จากสมการ (7) จะได้สมการ (13) และ (14) :-

$$P_{reqIP}(r)=[P_{IP}N_{IP}(r)+P_{OhIP}]+P_{NM}[N_{IP}(r)-1] \quad ...(11)$$

$$P_{respIP}(r)=[P_{rIP}N_{IP}(r)+P_{rOhIP}]+P_{rNM}[N_{IP}(r)-1] \quad ...(12)$$

$$P_{reqNH}(r)=P_{NH}[N_N+N_{OtherRow}(r)]+P_{OhNH} \quad ....(13)$$

$$P_{respNH}(r)=P_{rNH}[N_N+N_{OtherRow}(r)]+P_{rOhNH} \quad .....(14)$$

โดยที่ $P_{reqIP}(r)$ $P_{reqNH}(r)$ คือปริมาณของข้อมูลในหน่วยไบต์ที่ manager ใช้ไปในการร้องขอข้อมูลของ IP-interface และ Next-hop จากเราท์เตอร์ r ส่วน $P_{respIP}(r)$ $P_{respNH}(r)$ คือปริมาณข้อมูลของ IP-interface และ Next-hop ในหน่วยไบต์ที่ manager ได้รับจากเราท์เตอร์ r ส่วนขนาดแพ็กเก็ตของ Ethernet PDU ที่เกิดขึ้นขณะร้องขอข้อมูลของ IP-interface ซึ่งได้แก่ P_{IP} P_{rIP} จะเป็นขนาดแพ็กเก็ตของ Request และ Response PDU ที่เกิดขึ้นจากการร้องขอข้อมูลหมายเลข IP และ P_{OhIP} P_{rOhIP} จะเป็นขนาดแพ็กเก็ตของโอเวอร์เฮดของ Request และ Response PDU ที่เกิดจากการร้องขอข้อมูลของหมายเลข IP จากเราท์เตอร์ ส่วน P_{NM} P_{rNM} เป็นขนาดแพ็กเก็ตของ Request และ Response PDU ที่เกิดขึ้นจากการร้องขอข้อมูลของ net mask ในขณะที่ขนาดแพ็กเก็ตของ Ethernet PDU ที่เกิดขึ้นขณะร้องขอข้อมูลของ Next-hop ได้แก่ P_{NH} P_{rNH} เป็นขนาดแพ็กเก็ตของ Request และ Response PDU ที่เกิดขึ้นจากการร้องขอข้อมูลของ next hop ส่วน P_{OhNH} P_{rOhNH} เป็นขนาดแพ็กเก็ตของโอเวอร์เฮดของ Request และ Response PDU ที่เกิดจากการร้องขอข้อมูลของ next hop และ P_{Ext} P_{rExt} เป็นขนาดแพ็กเก็ตของ Request และ Response PDU ที่เกิดจากการร้องขอส่วนที่เป็นข้อมูลเพิ่มเติม ดังนั้นปริมาณข้อมูลที่ manager ใช้ไปในการร้องขอข้อมูลและปริมาณข้อมูลที่ manager ได้รับจากเราท์เตอร์ r คือ

$$P_{req}(r)=P_{reqIP}(r)+P_{reqNH}(r)+P_{reqOther}(r) \quad ....(15)$$

$$P_{resp}(r)=P_{respIP}(r)+P_{respNH}(r)+P_{respOther}(r) \quad .....(16)$$

$$P_{reqOther}(r)=P_{Ext}N_{Other}(r) \quad .....(17)$$

$$P_{respOther}(r)=P_{Ext}N_{Other}(r) \quad .....(18)$$

โดยที่ $P_{reqOther}$ และ $P_{respOther}$ คือปริมาณข้อมูลที่ใช้เกิดขึ้นจากการร้องขอข้อมูลเพิ่มเติม และเมื่อ manager ทำการร้องขอข้อมูลจากเราท์เตอร์ทุกตัวในระบบ จะทำให้ได้สมการดังนี้

$$P_{req} = \sum_{r=1}^{N_R} P_{reqIP}(r) + \sum_{r=1}^{N_R} P_{reqNH}(r) \quad \text{.....(19)}$$

$$P_{resp} = \sum_{r=1}^{N_R} P_{respIP}(r) + \sum_{r=1}^{N_K} P_{respNH}(r) \quad \text{.....(20)}$$

$$P_{req} = \sum_{r=1}^{N_R} [P_{IP}N_{IP}(r) + P_{NM}N_{IP}(r)] + (P_{OhIP} - P_{NM})N_R + \sum_{r=1}^{N_R} P_{reqOther}(r)$$
$$+ \sum_{r=1}^{N_R} P_{NH}N_{OtherRow}(r) + P_{NH}N_N N_R + P_{OhNH}N_R \quad \text{.....(21)}$$

$$P_{resp} = \sum_{r=1}^{N_R} [P_{rIP}N_{IP}(r) + P_{rNM}N_{IP}(r)] + (P_{rOhIP} - P_{rNM})N_R + \sum_{r=1}^{N_K} P_{respOther}(r)$$
$$+ \sum_{r=1}^{N_R} P_{rNH}N_{OtherRow}(r) + P_{rNH}N_N N_R + P_{rOhNH}N_R \quad \text{...(22)}$$

โดยที่ P_{req} คือปริมาณข้อมูลในหน่วยไบต์ที่เกิดจากการร้องขอข้อมูลของ manager จากเราท์เตอร์ทุกตัวในระบบ ส่วน P_{resp} คือปริมาณข้อมูลในหน่วยไบต์ที่ manager ได้รับมาจากเราท์เตอร์ทุกตัวในระบบ เมื่อแทนค่า Ethernet PDU จากตารางที่ 1 และ 2 ลงในสมการ (21) และ (22) แล้วจะได้สมการสำหรับใช้คำนวณปริมาณข้อมูลอย่างน้อยที่สุดที่เกิดขึ้นจากการสืบค้นดังนี้

$$P_{req} = 186\sum_{r=1}^{N_R} N_{IP}(r) + 93N_R N_N + 93\sum_{r=1}^{N_R} N_{OtherRow}(r)$$
$$+ 85N_R + 88\sum_{r=1}^{N_R} P_{reqOther}(r) \quad \text{.....(23)}$$

$$P_{resp} = 194\sum_{r=1}^{N_K} N_{IP}(r) + 97N_R N_N + 97\sum_{r=1}^{N_R} N_{OtherRow}(r)$$
$$+ 91N_R + 88\sum_{r=1}^{N_R} P_{respOther}(r) \quad \text{.....(24)}$$

$$P = P_{req} + P_{resp} \quad \text{.....(25)}$$

โดยที่ P คือปริมาณข้อมูลทั้งหมดในหน่วยไบต์ที่เกิดขึ้นบนระบบเครือข่ายจากการสืบค้นซึ่งจะถูกนำไปใช้ในการวิเคราะห์เวลาที่ใช้ไปในการสืบค้นในหัวข้อถัดไป

3.3 เวลา

เวลาที่ใช้ไปในการรับส่งข้อมูลระหว่าง manager และเราท์เตอร์โดยผ่านทางระบบเครือข่ายในแต่ละรอบนั้นประกอบด้วย เวลาที่เราท์เตอร์รับคำร้องขอจาก manager แล้วเริ่มทำการประมวลผลและทำการเข้าถึงข้อมูลจนกระทั่งข้อมูลถูกเตรียมพร้อมที่จะส่ง เวลาที่ manager ใช้ประมวลผลข้อมูลและเตรียมพร้อมที่จะส่งคำร้องขอข้อมูลไปยังเราท์เตอร์ และเวลาที่ใช้ในการเดินทางของข้อมูลจากต้นทางไปยังปลายทาง เป็นต้น การวิเคราะห์เวลาในบทความนี้จะวิเคราะห์เฉพาะเวลาที่ข้อมูลใช้เดินทางระหว่างmanager และเราท์เตอร์ซึ่งเป็นเวลาน้อยสุดที่ใช้ในการรับส่งข้อมูลเท่านั้น โดยจะไม่วิเคราะห์เวลาที่เกิดขึ้นในส่วนอื่นๆ เมื่อพิจารณาจากปริมาณข้อมูลในหน่วยไบต์ที่เกิดขึ้นจากสมการ (25) จะได้

$$Time = 8P/BW \quad \text{.....(26)}$$

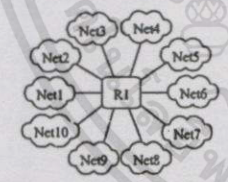
โดยที่ Time คือเวลาน้อยสุดที่ใช้ไปในการสืบค้น ส่วน BW คือขนาดของแบนด์วิดท์ของระบบเครือข่าย มีหน่วยเป็นบิตต่อวินาที

4 การทดลอง

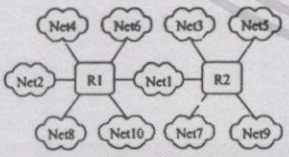
องค์ประกอบที่สำคัญในการทดลองสำหรับบทความนี้คือ manager และเราท์เตอร์ โดย manager ประกอบด้วยโปรแกรมที่ถูกเขียนขึ้นและคอมไพล์โดยใช้ Borland Delphi 4.0 ทำงานอยู่บนระบบปฏิบัติการ Mircrosoft Windows 95 และคอมพิวเตอร์พีซี (Personel Computer) ที่ใช้ Intel Pentium 133 MHz เป็นหน่วยประมวลกลาง พร้อมด้วยหน่วยความจำ 48 เมกกะไบต์ และเชื่อมต่อเข้ากับเครือข่าย Ethernet ด้วยความเร็ว 10 เมกกะบิตต่อวินาที ในขณะที่เราท์เตอร์ที่ใช้ในการทดลองทั้ง 3 ตัวจะทำงานโดยใช้ระบบปฏิบัติการ Microsoft WindowsNT 4.0 และ

คอมพิวเตอร์พีซีที่ใช้ Intel Pentium II 350 MHz เป็นหน่วยประมวลผลกลาง พร้อมด้วยหน่วยความจำ 48 เมกกะไบต์ และเชื่อมต่อเข้ากับเครือข่าย Ethernet ด้วยความเร็ว 10 เมกกะบิตต่อวินาที เราท์เตอร์แต่ละตัวจะมีเน็ทเวิร์คการ์ดแบบ Ethernet 2 การ์ด

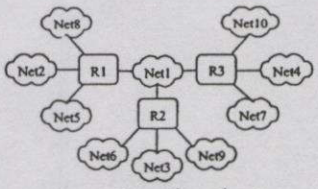
การทดลองจะแบ่งออกเป็น 3 กรณีคือ กรณีที่ระบบมี เราท์เตอร์ 1 ตัว 2 ตัว และ 3 ตัว โดยแต่ละกรณีจะเริ่มต้นด้วยจำนวนเครือข่ายย่อยในระบบจาก 4 จนถึง 10 เครือข่าย การเพิ่มจำนวนเครือข่ายย่อยแต่ละครั้งจะทำโดยการเพิ่มหมายเลข IP เข้าไปในเราท์เตอร์ โดยหมายเลข IP นั้นจะต้องเป็นหมายเลข IP ที่อยู่ในเครือข่ายย่อยอันใหม่จากรูปที่ 5 (ก-ค) ดังนั้นเมื่อหมายเลข IP ของเราท์เตอร์เพิ่มขึ้นจำนวนเครือข่ายย่อยก็จะเพิ่มขึ้นตามด้วย นอกจากนี้ยังให้ manager ทำการร้องขอข้อมูลเพิ่มเติมคือชื่อของเราท์เตอร์ (System name) สำหรับการทดลองในครั้งนี้ด้วย ผลลัพธ์ที่ได้จากการคำนวณจะถูกนำมาเปรียบเทียบกับผลลัพธ์ที่ได้จากการทดลอง ต่อไปนี้ผลที่ได้จากการทดลอง



(ก) กรณีเราท์เตอร์ 1 ตัว



(ข) กรณีเราท์เตอร์ 2 ตัว

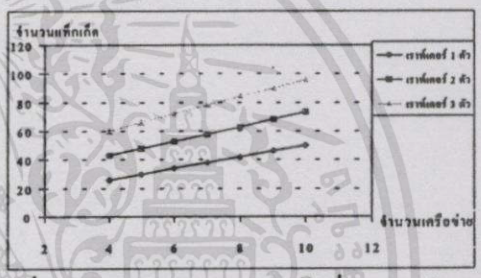


(ค) กรณีเราท์เตอร์ 3 ตัว

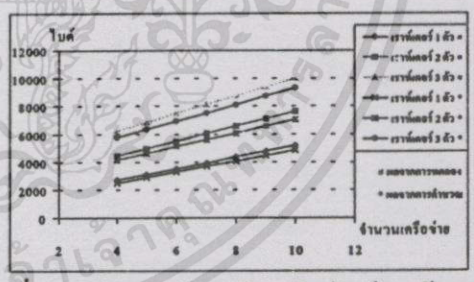
รูปที่ 5 แสดงระบบเครือข่ายที่ใช้ในการทดลอง

ตารางที่ 3 แสดงจำนวนแพ็กเก็ตที่เกิดจากการร้องขอข้อมูลของ manager

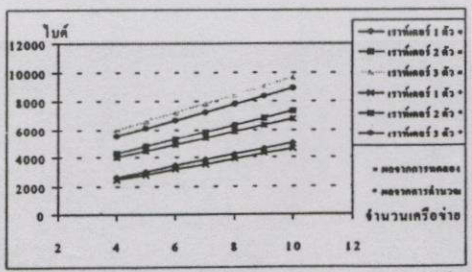
จำนวนเครือข่าย	เราท์เตอร์ 1 ตัว		เราท์เตอร์ 2 ตัว		เราท์เตอร์ 3 ตัว	
	คำนวณ	ทดลอง	คำนวณ	ทดลอง	คำนวณ	ทดลอง
4	26	26	43	43	60	60
5	30	30	48	48	66	66
6	34	34	53	53	72	72
7	38	38	58	58	78	78
8	42	42	63	63	84	84
9	46	46	68	68	90	90
10	50	50	73	73	96	96



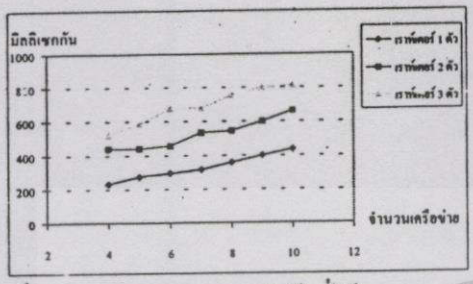
รูปที่ 6 กราฟแสดงจำนวนแพ็กเก็ตที่จากการร้องขอข้อมูลของ manager



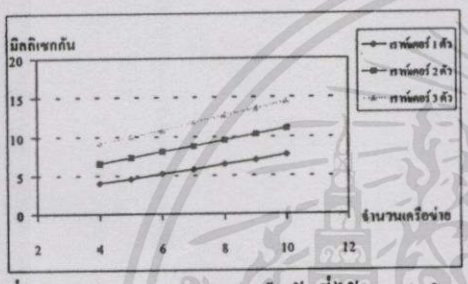
รูปที่ 7 กราฟแสดงปริมาณในหน่วยไบต์ของข้อมูลที่ manager ได้รับจากเราท์เตอร์



รูปที่ 8 กราฟแสดงปริมาณข้อมูลในหน่วยไบต์ที่เกิดจากการร้องข้อมูลของ manager



รูปที่ 9 กราฟแสดงเวลาของการสืบค้นที่ได้จากการทดลอง



รูปที่ 10 กราฟแสดงเวลาของการสืบค้นที่ได้จากการคำนวณ

เนื่องจากจำนวนแถวอื่นๆ ($N_{OtherRow}(r)$) ในตารางเราท์ติ้งของเราท์เตอร์ที่ใช้ทำการทดลองครั้งนี้มีจำนวนค่อนข้างมากเมื่อเปรียบเทียบจำนวนแถวทั้งหมดในตารางเราท์ติ้ง ดังนั้นผู้ทำวิจัยจึงได้นำค่าของตัวแปรในส่วนนี้มาใช้ในการคำนวณด้วย และเมื่อนำมาเปรียบเทียบกับผลลัพธ์ที่ได้จากการทดลองจะพบว่าในกรณีของจำนวนแพ็กเก็ตผลลัพธ์ทั้งสอง (จากตารางที่ 3 และรูปที่ 6) มีค่าเท่ากันนั่นแสดงว่าสมการที่ได้มาจากการวิเคราะห์มีความถูกต้อง ในขณะที่ผลลัพธ์ที่ได้จากการคำนวณและการทดลองของปริมาณข้อมูลในหน่วยไบต์ที่เกิดจากการสืบค้น (ในรูปที่ 7-8) จะมีค่าที่ใกล้เคียงกันโดยผลลัพธ์ที่ได้จากการคำนวณจะมีค่าที่ต่ำกว่าในทุกกรณี นั่นย่อมแสดงให้เห็นว่าสมการที่ได้จะเป็นสมการที่ใช้ในการประมาณค่าขั้นต่ำของปริมาณข้อมูลที่เกิดขึ้นจริงได้ ส่วนเวลาของการสืบค้นซึ่งผลที่ได้จากการทดลองและการคำนวณ (รูปที่ 9-10) จะมีค่าที่แตกต่างกันค่อนข้างมาก ทั้งนี้เนื่องจากเวลาที่ได้จากการทดลองจะประกอบด้วยเวลาที่เราท์เตอร์และ manager ใช้ไปในการประมวลผลข้อมูลต่างๆกับเวลาที่ใช้ไปในการรับส่งข้อมูลบนเครือข่าย แต่เวลาที่ได้จากการคำนวณจะเป็นเวลาที่ใช้ไปในการรับส่งข้อมูลบนเครือข่ายเท่านั้น ทำให้ทราบว่าเวลาส่วนใหญ่ที่ใช้ไปในการสืบค้นนั้นคือเวลาที่ใช้ไปในการเข้าถึงและประมวลผลข้อมูลนั่นเอง และจากผลลัพธ์ทั้งหมดทั้งที่ได้จากการทดลองและการคำนวณพบว่าเมื่อเครือข่ายและเราท์เตอร์มีจำนวนเพิ่มขึ้นจะทำให้จำนวนแพ็กเก็ตและปริมาณข้อมูลที่เกิดขึ้นและเวลาที่ใช้ไปในการสืบค้นมีค่าเพิ่มขึ้นตามไปด้วย

5 สรุป

จากการศึกษา วิเคราะห์ และทำการทดลอง พบว่ามีตัวแปรหลายตัวที่มีผลต่อปริมาณข้อมูลและเวลาที่ใช้ไปในการสืบค้น โดยตัวแปรสำคัญที่มีผลต่อปริมาณข้อมูลที่เกิดขึ้นบนเครือข่ายคือ จำนวนอินเตอร์เฟส (หมายเลข IP) ของเราท์เตอร์ จำนวนเราท์เตอร์และจำนวนเครือข่ายย่อยของระบบ ส่วนจำนวนแถวอื่นๆในตารางเราท์ติ้งของเราท์เตอร์ซึ่งเป็นตัวแปรอีกตัวหนึ่งนั้นจะขึ้นอยู่กับระบบปฏิบัติการของเราท์เตอร์

สมการที่ได้จากบทความนี้สามารถใช้ในการคำนวณหาปริมาณข้อมูลที่เกิดขึ้นเมื่อมีการสืบค้น โดยสมการ (10) ใช้สำหรับคำนวณหาจำนวนแพ็กเก็ตที่เกิดขึ้นจากการสืบค้น ส่วนสมการ (23)-(25) ใช้สำหรับคำนวณหาปริมาณข้อมูลในหน่วยไบต์ที่เกิดขึ้นจากการสืบค้นซึ่งสามารถนำไปใช้ในการคิดภาระ (Load) ของเครือข่ายร่วมกับปริมาณข้อมูลที่เกิดจากอุปกรณ์และแอพพลิเคชั่นอื่นๆในเครือข่ายย่อย ซึ่งจะทำให้ทราบว่าเครือข่ายย่อยใดที่มีแบนด์วิดท์ที่เหมาะสมที่จะติดตั้ง manager ในขณะที่ตัวแปรที่มีผลต่อเวลาที่ใช้ไปในการสืบค้นนอกจากจำนวนอินเตอร์เฟสของเราท์เตอร์ จำนวนเราท์เตอร์และจำนวนเครือข่ายย่อยของระบบ

แล้ว ตัวแปรที่สำคัญอีกตัวหนึ่งก็คือขีดความสามารถในการเข้าถึงและประมวลผลข้อมูลของเครื่องคอมพิวเตอร์หรือฮาร์ดแวร์ที่ manger และเราท์เตอร์ทำงานอยู่ เพราะฉะนั้นในการติดตั้งระบบการจัดการเครือข่ายควรที่จะคำนึงถึงขีดความสามารถของฮาร์ดแวร์เพื่อประสิทธิภาพในการทำงานที่ดีของ manager ด้วย

6 เอกสารอ้างอิง

[1] Stalling, William. **Local and metropolitan area networks, Forth Edition**, Macmillan Publishing Company, Macmillan Inc. 1993.

[2] Stallings, William. **SNMP SNMPv2 and RMON : Practical Network Management, Second Edition**, Addison-Wesley Publishing Company, Inc. 1996.

[3] Feit, Sidnie. **SNMP a guide to network management**, McGraw-Hill, Inc. 1995

[4] Glenn Manfield, M. Ouchi, K. Jayanthi, Y. Kimura. "Techniques for automated Network Map Generation using SNMP", Infocom96, San Francisco, USA, March 26-28, 1996.

[5] Merilee Ford, H. Kim Lew, Steve Spanier, Tim Steven. **Internetworking Technologies Handbook.** New Riders Publishing, 1997.

[5] K. McCloghrie and M. Rose. "Management Interface Base for Network Management of TCP/IP-based internets : MIB-II.", Request For Comment 1213, Hughes LAN Systems, Performance Systems International, March 1991

[7] Black, Uyless. **TCP/IP and Related Protocols, Second Edition**, McGraw-Hill, Inc. 1995

ประวัติผู้เขียน

ชื่อผู้เขียน	นายธีรกฤษณ์ จันทเบญจม์ภัทร
วันเดือนปีเกิด	วันที่ 28 มีนาคม พ.ศ. 2514
สถานที่เกิด	อำเภอเมือง จังหวัดตรัง
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2535
ผลงานทางวิชาการที่ได้รับการตีพิมพ์	การศึกษาองค์ประกอบของเครือข่ายที่มีผลต่อการสืบค้นเครือข่าย