การพัฒนาโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่ายผ่านบราวเซอร์ บนระบบปฏิบัติการวินโดวส์เอ็นที

นาย ชัยวัฒน์ ฉัตรศิริชัย

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2542

ISBN 974-333-474-2

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEVELOPMENT OF A BROWSER-BASED NETWORK DEVICES MANAGEMENT PROGRAM ON THE WINDOWS NT

Mr. Chaiwat Chatsirichai

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science

Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University

Academic Year 1999
ISBN 974-333-474-2

จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อวิทยานิพนธ์	การพัฒนาโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่ายผ่านบราวเซอร์บนระบบปฏิบัติการวินโดวส์เอ็นที
โดย	นาย ชัยวัฒน์ ฉัตรศิริชัย
ภาควิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ จารุมาตร ปิ่นทอง
อาจารย์ที่ปรึกษาร่วม	อาจารย์ เดชานุชิต กตัญญูทวีทิพย์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร. ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษา
(อาจารย์ จารุมาตร ปิ่นทอง)

..... อาจารย์ที่ปรึกษาร่วม
(อาจารย์ เดชานุชิต กตัญญูทวีทิพย์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. สมชาย ประสิทธิ์จูตระกูล)

..... กรรมการ
(อาจารย์ ชัยศิริ ปัณฑิตานนท์)

จุฬาลงกรณ์มหาวิทยาลัย

ชัยวัฒน์ ฉัตรศิริชัย: การพัฒนาโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่ายผ่านบราวเซอร์บนระบบปฏิบัติการวินโดวส์เอ็นที (DEVELOPMENT OF A BROWSER-BASED NETWORK DEVICES MANAGEMENT PROGRAM ON THE WINDOWS NT) อ. ที่ปรึกษา : อ. จารุมาตร ปิ่นทอง, อ. ที่ปรึกษาร่วม : อ. เดชานุชิต กตัญญูทวีทิพย์, 79 หน้า. ISBN 974-333-474-2

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อประยุกต์ใช้โพรโทคอลเอสเอ็นเอ็มพีในการจัดการอุปกรณ์ทางระบบเครือข่าย และพัฒนาโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่าย โดยทำงานอยู่บนเครื่องให้บริการมีหน้าที่เป็นตัวบริการเว็บ ซึ่งผู้ใช้งานสามารถติดต่อเพื่อจัดการอุปกรณ์ด้วยโปรแกรมค้นผ่านเว็บ หรือ บราวเซอร์จากเครื่องขอรับบริการ

การพัฒนาโปรแกรมแบ่งออกเป็นสองส่วน คือ ส่วนจัดการอุปกรณ์ซึ่งมีหน้าที่สอบถาม รายงาน และแก้ไขข้อมูลเกี่ยวกับอุปกรณ์ และ ส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี มีหน้าที่สนับสนุนส่วนจัดการอุปกรณ์ โดยอาศัยหลักการของโอเปอเรชันเอสเอ็นเอ็มพี ซึ่งประกอบไปด้วย โอเปอเรชันเก็ต โอเปอเรชันเก็ต-เน็กส์ โอเปอเรชันเซ็ต โอเปอเรชันแทร็ป และ โอเปอเรชันเก็ต-เรสปอนส์ ตามข้อกำหนดของอาร์เอฟซี 1157 ซึ่งพัฒนาโดยใช้โปรแกรมภาษาเพิร์ล 5 สำหรับ วิน 32

ในงานวิจัยได้นำอุปกรณ์ทางระบบเครือข่ายประกอบไปด้วย อุปกรณ์จัดเส้นทาง และ อีเทอร์เน็ตสวิช จากหลายผู้ผลิตมาใช้ทดสอบการทำงานของโปรแกรมที่ได้พัฒนาขึ้นนี้ ซึ่งสามารถรองรับงานจัดการอุปกรณ์ทางระบบเครือข่ายด้วยบราวเซอร์ได้เป็นอย่างดี ง่ายต่อการใช้งาน สามารถตรวจสอบ รายงานสถานะ และแก้ไขข้อมูลของอุปกรณ์ที่สามารถรองรับมิบ-ทู (อาร์เอฟซี 1213) และ อามอนมิบ (อาร์เอฟซี 1757) โดยไม่จำกัดสถานที่เรียกใช้งานในเครือข่าย และไม่ได้เปลี่ยนแปลงแก้ไขหรือเพิ่มเติมระบบฮาร์ดแวร์และซอฟต์แวร์ของอุปกรณ์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์	**ลายมือชื่อนิสิต** ชัยวัฒน์ ฉัตรศิริชัย
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์	**ลายมือชื่ออาจารย์ที่ปรึกษา**
ปีการศึกษา 2542	**ลายมือชื่ออาจารย์ที่ปรึกษาร่วม**

3970421721 : MAJOR COMPUTER SCIENCE
KEYWORD: SIMPLE NETWORK MANAGEMENT PROTOCOL / WEB BROWSER
CHAIWAT CHATSIRICHAI : DEVELOPMENT OF A BROWSER-BASED NETWORK DEVICES MANAGEMENT PROGRAM ON THE WINDOWS NT. THESIS ADVISOR : CHARUMATR PINTHONG, THESIS CO-ADVISOR : DECHANUCHIT KATANYUTAVEETIP, 79 pp. ISBN 974-333-474-2.

The objective of this thesis is to apply Simple Network Management Protocol (SNMP) to manage network devices and to develop a network management program which allows client to manage network devices using a web browser.

The program consists of two parts : network devices manager and SNMP manager. The first part performs network device polling, status reporting and information changes. The second part provides SNMP operations to support network management activities of the first part. These SNMP operations are Get Operation, Get-Next Operation, Set Operation, Trap Operation and Get-Response Operation. These operations are based on SNMP specification in RFC 1157. The program is developed by using Perl 5 for Win32.

The program is a useful tool that allows user to monitor and administer SNMP capable devices that support MIB-II (RFC 1213) and RMON MIB (RFC 1757) via a web browser from anywhere in the network. In this research, the program was successfully tested with network devices such as routers and ethernet switches from different manufacturers without any hardware and software modification on these devices.

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2542 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงไปได้โดยความช่วยเหลืออย่างดียิ่งจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ อาจารย์ จารุมาตร ปิ่นทอง และ อาจารย์ที่ปรึกษาร่วม อาจารย์ เดชานุชิต กตัญญูทวีทิพย์ ซึ่งทั้งสองท่านได้ให้แนวทาง และ ข้อเสนอแนะมาตลอด ขอกราบขอบพระคุณ อาจารย์ ดร. ยรรยง เต็งอำนวย ผู้ช่วยศาสตราจารย์ ดร. สมชาย ประสิทธิ์จูตระกูล และ อาจารย์ ชัยศิริ ปัณฑิตานนท์ ที่ได้เสียสละเวลาในการให้คำแนะนำ ตรวจสอบ และแก้ไขจุดบกพร่องต่างๆ ขอกราบขอบพระคุณอาจารย์ผู้สอนทุกท่านที่ได้ให้ความรู้ตลอดระยะเวลาที่ผู้วิจัยได้เข้ามาศึกษา ณ สถาบันแห่งนี้

ขอขอบคุณ บริษัท เทเลคอมเอเซีย คอร์ปอเรชั่น จำกัด (มหาชน) และ บริษัท ไนเน็กซ์ ไซเอ็นซ์ แอนด์ เทคโนโลยี่ เอเชีย จำกัด ที่ได้ให้ความอนุเคราะห์ให้ใช้อุปกรณ์ทางระบบเครือข่ายสำหรับใช้เป็นกรณีในการศึกษา

ท้ายสุดนี้ ผู้วิจัยขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ได้ให้การสนับสนุนและเป็นกำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 2
1.3 ขอบเขตของการวิจัย 2
1.4 ขั้นตอนและวิธีดำเนินการวิจัย 4
1.5 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย 4
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย 5
2.1 การจัดการระบบเครือข่าย (NETWORK MANAGEMENT) 5
2.2 โพรโทคอลเอสเอ็นเอ็มพี (SIMPLE NETWORK MANAGEMENT PROTOCOL) 6
2.3 เอสเอ็นเอ็มพี เมสเสจ (SNMP MESSAGE) 9
2.4 โอเปอเรชั่น เอสเอ็นเอ็มพี (SNMP OPERATION) 10
2.5 มิบ (MIB) 11
2.6 มิบ-ทู (MIB-II) 14
2.7 ไพร์เวทมิบ (PRIVATE MIB) 15
2.8 อามอนมิบ (RMON MIB) 15
2.9 องค์ประกอบของโปรแกรมจัดการระบบเครือข่าย 16
2.10 การเฝ้าดูและระบบเครือข่าย (NETWORK MANITORING) 17
2.11 กฎเบสิกเอ็นโค้ดดิ้งรูล (BASIC ENCODING RULE) 17
2.12 อินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์ (INTERNET INFORMATION SERVER) 21
บทที่ 3 การวิเคราะห์และออกแบบโปรแกรม 23
3.1 การออกแบบส่วนจัดการอุปกรณ์ 24
3.2 การออกแบบส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี 24
3.2.1 ส่วนจัดการโอเปอเรชั่นเอสเอ็นเอ็มพี 24

3.2.2 ส่วนจัดการเบสิกเอ็นโค้ดดิ้งรูล 24
3.3 การออกแบบและจัดรูปแบบเว็บเพจ 25
3.4 การออกแบบโครงสร้างข้อมูล 26
3.4.1 การออกแบบโครงสร้างข้อมูลของรายชื่อออบเจกต์ 26
3.4.2 การออกแบบโครงสร้างข้อมูลของค่าออบเจกต์ 27
3.4.3 การออกแบบโครงสร้างแฟ้มข้อมูล 27
บทที่ 4 การพัฒนาโปรแกรม 32
4.1 ส่วนจัดการอุปกรณ์ทางเครือข่าย 33
4.1.1 โปรแกรมสอบถามอุปกรณ์ (SnmpDevicePoll.cgi) 33
4.1.2 โปรแกรมจัดการแท็ปเมสเสจ (SnmpDeviceTrapListen.cgi) 37
4.1.3 โปรแกรมแก้ไขอุปกรณ์ (SnmpDeviceSet.cgi) 37
4.1.4 โปรแกรมจัดการแผนภูมิ (SnmpDeviceChartTimer.cgi) 42
4.2 ส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี 44
4.2.1 โปรแกรมจัดการโอเปอเรชั่นเอสเอ็นเอ็มพี 44
4.2.2 โปรแกรมจัดการเบสิกเอ็นโค้ดดิ้งรูล 52
4.3 การติดตั้งโปรแกรมสำหรับเครื่องให้บริการ 55
บทที่ 5 การทดสอบโปรแกรม 57
5.1 ขอบเขตการทดสอบ 57
5.2 การทดสอบการค้นหาอุปกรณ์ 59
5.2.1 ขั้นตอนในการทดสอบ 59
5.2.2 ผลการทดสอบ 59
5.3 การทดสอบการสอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์ 60
5.3.1 ขั้นตอนในการทดสอบ 61
5.3.2 ผลการทดสอบ 61
5.4 การทดสอบการสอบถามตามหมายเลขไอพี 62
5.4.1 ขั้นตอนในการทดสอบ 62
5.4.2 ผลการทดสอบ 63
5.5 การทดสอบการสอบถามตามรายชื่อออบเจกต์ 64
5.5.1 ขั้นตอนในการทดสอบ 64
5.5.2 ผลการทดสอบ 65

5.6 การทดสอบการรายงานข้อมูลในรูปของแผนภูมิ....66
5.6.1 ขั้นตอนในการทดสอบ....66
5.6.2 ผลการทดสอบ....68
5.7 การทดสอบการรายงานแทร็ปเมสเสจ....69
5.7.1 ขั้นตอนในการทดสอบ....69
5.7.2 ผลการทดสอบ....70
5.8 การทดสอบการแก้ไขอุปกรณ์....71
5.8.1 ขั้นตอนในการทดสอบ....71
5.8.2 ผลการทดสอบ....72
5.9 สรุปผลการทดสอบ....72
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ....75
6.1 สรุปผลการวิจัย....75
6.2 ข้อเสนอแนะ....76
รายการอ้างอิง....77
ประวัติผู้เขียน....79

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ 2.1 แสดงรายละเอียดของเอสเอ็นเอ็มพีพีดียู 10
ตารางที่ 2.2 แสดงรายละเอียดของเอสเอ็นเอ็มพี โอเปอเรชั่น 11
ตารางที่ 2.3 แสดงรายละเอียดของรหัสเอชทีเอ็มแอล FORM 22

ตารางที่ 4.1 แสดงตัวแปรและฟังก์ชัน เรียกใช้โดยฟังก์ชัน SnmpDeviceDiscovery 35
ตารางที่ 4.2 แสดงตัวแปร เรียกใช้โดยโปรแกรม SnmpDeviceTrapListen.cgi 37
ตารางที่ 4.3 แสดงออบเจกต์ภายใต้มิบ-ทู ที่สามารถแก้ไขค่าได้ 38
ตารางที่ 4.4 แสดงการกำหนดองค์ประกอบและชนิดของออบเจกต์ โดยค่าของตัวแปร $poll 50

ตารางที่ 5.1 แสดงผลการทดสอบการให้บริการประเภทต่างๆโปรแกรม 73
ตารางที่ 5.2 แสดงผลการทดสอบโปรแกรมในการรายงานค่ามิบ-ทู ของอุปกรณ์ 73
ตารางที่ 5.3 แสดงผลการทดสอบโปรแกรมในการรายงานค่าอามอนมิบ ของอุปกรณ์ 74
ตารางที่ 5.4 แสดงผลการทดสอบโปรแกรมในการรายงานค่าไพร์เวทมิบ ของอุปกรณ์ 74
ตารางที่ 5.5 แสดงผลการทดสอบโปรแกรมในการแก้ไขออบเจกต์ภายใต้มิบ-ทู ของอุปกรณ์ 74

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปที่ 2.1 แสดงโพรโทคอลเอสเอ็นเอ็มพีและชั้นที่เกี่ยวข้องในโอเอสไอโมเดล.....7
รูปที่ 2.2 แสดงองค์ประกอบของ สถาปัตยกรรมของการจัดการระบบเครือข่ายอินเทอร์เน็ต.....8
รูปที่ 2.3 แสดงการรับส่งเอสเอ็นเอ็มพีเมสเสจ.....8
รูปที่ 2.4 แสดงโครงสร้างของเอสเอ็นเอ็มพีเมสเสจ.....9
รูปที่ 2.5 แสดงโครงสร้างของแทร็ปพีดียู.....9
รูปที่ 2.6 แสดงรายละเอียดของเอสเอ็นเอ็มพี โอเปอเรชั่น.....11
รูปที่ 2.7 แสดงออบเจกต์ในต้นมิบ.....12
รูปที่ 2.8 แสดงออบเจกต์ ifNumber และออบเจกต์ ifSpeed.....12
รูปที่ 2.9 แสดง โอเปอเรชั่นเก็ตเพื่ออ่านออบเจกต์ประเภทค่าเดี่ยว.....13
รูปที่ 2.10 แสดง โอเปอเรชั่นเก็ต-เน็กส์ เพื่ออ่านค่าของออบเจกต์ประเภทตาราง.....14
รูปที่ 2.11 แสดงกลุ่มมิบ-ทู.....15
รูปที่ 2.12 แสดงรายละเอียดของ ทีแอลวีสตรัคเจอร์ (TLV Structure).....18
รูปที่ 2.13 แสดงการแปลงเลขจำนวนเต็ม 100 ให้อยู่ในรูปของไบต์.....19
รูปที่ 2.14 แสดงการแปลงสตริง “public” ให้อยู่ในรูปของไบต์.....19
รูปที่ 2.15 แสดงการแปลงค่าว่าง (NULL) ให้อยู่ในรูปของไบต์.....20
รูปที่ 2.16 แสดงการแปลงหมายเลขออบเจกต์ 1.3.6.1.2.1.1.1.1.0 ให้อยู่ในรูปของไบต์.....20
รูปที่ 2.17 แสดงการแปลงโครงสร้างเอเอสเอ็น.1 แบบลำดับ (SEQUENCE) ให้อยู่ในรูปของไบต์21
รูปที่ 2.18 แสดงการทำงานของซีจีไอผ่านเอชทีเอ็มแอลฟอร์ม.....22

รูปที่ 3.1 แสดงการทำงานโดยรวมของระบบ.....23
รูปที่ 3.2 แสดงการทำงานของส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี.....25
รูปที่ 3.3 แสดงหน้าจอหลัก.....26
รูปที่ 3.4 แสดงโครงสร้างข้อมูลแบบแฮชสำหรับข้อมูลของรายชื่อออบเจกต์.....27
รูปที่ 3.5 แสดงโครงสร้างข้อมูลแบบแฮชสำหรับข้อมูลของค่าออบเจกต์.....27
รูปที่ 3.6 แสดงรูปแบบของแฟ้มข้อมูลรายชื่ออุปกรณ์.....28
รูปที่ 3.7 แสดงรูปแบบของแฟ้มข้อมูลทางสถิติของอุปกรณ์.....31
รูปที่ 3.8 แสดงรูปแบบของแฟ้มข้อมูลรายชื่อแผนภูมิ.....31

รูปที่ 4.1 แสดงองค์ประกอบของโปรแกรมจัดการอุปกรณ์ทางเครือข่าย 33
รูปที่ 4.2 แสดงการทำงานของโปรแกรม SnmpDeviceChartTimer.cgi 43
รูปที่ 4.3 แสดงการทำงานของฟังก์ชัน SnmpGetScalarObject 44
รูปที่ 4.4 แสดงการทำงานของฟังก์ชัน SnmpGetTabularObject 46
รูปที่ 4.5 แสดงการทำงานของฟังก์ชัน SnmpSetObject 48
รูปที่ 4.6 แสดงการทำงานของฟังก์ชัน SnmpDevicePoll 51
รูปที่ 4.7 แสดงโครงสร้างของระบบแฟ้ม.....55

รูปที่ 5.1 แสดงสภาพแวดล้อมของการทดสอบโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่าย 59
รูปที่ 5.2 แสดงหน้าจอค้นหาอุปกรณ์.....60
รูปที่ 5.3 แสดงผลการทดสอบการค้นหาอุปกรณ์.....60
รูปที่ 5.4 แสดงตัวอย่างแฟ้มข้อมูลรายชื่ออุปกรณ์.....61
รูปที่ 5.5 แสดงหน้าจอสอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์.....61
รูปที่ 5.6 แสดงผลการทดสอบการสอบถาม ตามรายชื่อแฟ้มข้อมูลอุปกรณ์.....62
รูปที่ 5.7 แสดงหน้าจอสอบถามอุปกรณ์ด้วยหมายเลขไอพี.....63
รูปที่ 5.8 แสดงการรายงานข้อมูลของกลุ่มซิสเต็ม 63
รูปที่ 5.9 แสดงตัวอย่างการรายงานไพร์เวทมิบ.....64
รูปที่ 5.10 แสดงหน้าจอสอบถามตามรายชื่อออบเจกต์.....65
รูปที่ 5.11 แสดงตัวอย่างการรายงานค่าออบเจกต์ sysUpTime.....66
รูปที่ 5.12 แสดงตัวอย่างการรายงานค่าออบเจกต์ cardDescr.....66
รูปที่ 5.13 แสดงตัวอย่างแฟ้มข้อมูลรายชื่อแผนภูมิ.....67
รูปที่ 5.14 แสดงหน้าจอเริ่มงานรวบรวมข้อมูลสร้างแผนภูมิ.....67
รูปที่ 5.15 แสดงตัวอย่างหน้าจอเลือกอินเตอร์เฟสเพื่อแสดงแผนภูมิ 67
รูปที่ 5.16 แสดงตัวอย่างหน้าจอเลือกประเภทของแผนภูมิ 68
รูปที่ 5.17 แสดงตัวอย่างของปริมาณการใช้บริการ (Utilization) ของอินเตอร์เฟส.....68
รูปที่ 5.18 แสดงตัวอย่างของแพ็กเก็ตขนาดต่างๆที่ได้ผ่านเข้าออกจากอินเตอร์เฟส 69
รูปที่ 5.19 แสดงหน้าจอเริ่มการทำงานรับแทร็ปเมสเสจ.....70
รูปที่ 5.20 แสดงการรายงานแทร็ปเมสเสจ ที่ได้รับจากอุปกรณ์.....70
รูปที่ 5.21 แสดงหน้าจอแก้ไขอุปกรณ์.....71
รูปที่ 5.22 แสดงตัวอย่างหน้าจอแก้ไขออบเจกต์ ifAdminStatus ของกลุ่มอินเตอร์เฟส 72
รูปที่ 5.23 แสดงผลการทดสอบแก้ไขค่าออบเจกต์ 72

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบเครือข่ายคอมพิวเตอร์ในปัจจุบันประกอบด้วยอุปกรณ์หลากหลายต่างชนิดต่างแบบกันโดยเชื่อมต่อกันได้อย่างง่ายดาย ในทางปฏิบัติแล้วเราจะต้องใช้โปรแกรมประยุกต์สำหรับจัดการระบบเครือข่าย (Network Management Application) เป็นเครื่องมือช่วยในการควบคุม ติดตาม และจัดการข้อมูลเกี่ยวกับการทำงานของอุปกรณ์เหล่านี้ ถึงแม้ว่าในปัจจุบันอุปกรณ์ทางระบบเครือข่ายโดยส่วนใหญ่ใช้มาตรฐานการจัดการของโพรโทคอลเอสเอ็นเอ็มพี (SNMP ย่อมาจาก Simple Network Management Protocol) แต่เนื่องจากอุปกรณ์และโปรแกรมเหล่านี้ ได้ถูกพัฒนาขึ้นโดยผู้ผลิตอุปกรณ์ของแต่ละค่าย ดังนั้นจึงมีรูปแบบของวิธีการใช้งานแตกต่างกัน ไม่ว่าจะเป็นส่วนติดต่อผู้ใช้ (User Interface) และรูปแบบของการแสดงข้อมูล ผลที่ตามมาคือผู้มีหน้าที่ดูแลระบบเครือข่ายจะประสบความไม่สะดวกในการใช้งานโปรแกรมเหล่านี้หลายตัวพร้อมกัน

คอมพิวเตอร์ที่ต่อเชื่อมกันในระบบเครือข่ายอินเทอร์เน็ต (Internet) มีลักษณะการทำงานเป็นแบบรับ-ให้บริการ (Client-Server) โดยแต่ละเครื่องขอรับบริการ (Client) จะติดต่อกับเครื่องให้บริการ เรียกว่าตัวบริการเว็บ (Web Server) ด้วยโพรโทคอลเอชทีทีพี (HTTP ย่อมาจาก HyperText Transport Protocol) ในชั้นแอพพลิเคชั่น ของโอเอสไอโมเดล (OSI Model) ผ่านโพรโทคอลทีซีพี/ไอพี (TCP/IP Protocol) ข้อมูลที่เครื่องขอรับบริการได้รับจากเครื่องให้บริการจะอยู่ในรูปของแฟ้ม เอชทีเอ็มแอล (HTML ย่อมาจาก HyperText MarkUp Language) เรียกว่า เว็บเพจ (Web Page) ซึ่งสามารถแสดงผลผ่านโปรแกรมค้นผ่านเว็บ หรือเรียกอย่างง่ายว่า บราวเซอร์ (Browser) รูปแบบของการทำงานระหว่างเครื่องให้บริการและเครื่องขอรับบริการนี้ ทำให้เกิดแนวคิดในการนำวิธีนี้มาใช้ช่วยในการจัดการอุปกรณ์ทางระบบเครือข่าย ด้วยหลักการดังนี้

1) ตัวบริการเว็บสามารถกระจายข้อมูลได้ง่าย
2) ไม่จำกัดประเภทของบราวเซอร์
3) แฟ้มเอชทีเอ็มแอล สามารถสอดแทรกได้ทั้งรูปภาพและจาวาแอพเพล็ท (JAVA Applet โดยไม่ต้องใช้ซอฟต์แวร์และฮาร์ดแวร์เพิ่มเติม

ดังนั้น ถ้าได้มีการพัฒนาโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่าย โดยอาศัยบราวเซอร์เป็นส่วนติดต่อผู้ใช้ในการนำเสนอและรับส่งข้อมูล จะช่วยให้ผู้ที่มีหน้าที่ในการดูแลระบบเครือข่ายมีความสะดวกมากขึ้นในการจัดการอุปกรณ์ฯ โดยไม่จำกัดสถานที่เรียกใช้งานในเครือข่าย

ระบบปฏิบัติการวินโดวส์เอ็นที (Windows NT) เป็นระบบปฏิบัติการที่ใช้งานกันอย่างแพร่หลาย โดยใช้ โปรแกรมประยุกต์ ไอไอเอส (IIS ย่อมาจาก Internet Information Server) ทำหน้าที่เป็นตัวบริการเว็บซึ่งสามารถสนับสนุนมาตรฐานโปรแกรมตัวต่อประสานร่วมเกตเวย์ หรือ ซีจีไอ (CGI ย่อมาจาก Common Gateway Interface) ถ้าได้มีการพัฒนาโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่ายบนระบบปฏิบัติการวินโดวส์เอ็นที จะช่วยให้ผู้ที่เกี่ยวข้องสามารถลดปัญหาในการควบคุมการใช้งาน บำรุงรักษา และลดอุปสรรคในเรื่องสรรหาบุคลากรที่มีความรู้และประสบการณ์เหมือนระบบปฏิบัติการอื่น เช่น ระบบปฏิบัติการยูนิกส์

1.2 วัตถุประสงค์ของการวิจัย

เพื่อให้ได้โปรแกรมจัดการอุปกรณ์ทางระบบเครือข่ายผ่านบราวเซอร์ด้วยโพรโทคอลเอสเอ็นเอ็มพี บนระบบปฏิบัติการวินโดวส์เอ็นที

1.3 ขอบเขตของการวิจัย

1. โปรแกรมที่จะพัฒนาขึ้นนี้สามารถจัดการอุปกรณ์ทางระบบเครือข่าย ผ่านบราวเซอร์ โดยใช้โพรโทคอลเอสเอ็นเอ็มพีบนระบบปฏิบัติการวินโดวส์เอ็นที ประกอบด้วยตัวบริการเว็บโดยแสดงข้อมูลเกี่ยวกับสภาพการทำงาน และ ส่งคำสั่งควบคุมผ่านทางจอภาพได้ทีละอุปกรณ์
2. โปรแกรมสามารถทำงานกับอุปกรณ์ทางระบบเครือข่าย ที่สามารถรองรับประเภทของมิบภายใต้ โพรโทคอลเอสเอ็นเอ็มพี ตามอาร์เอฟซี 1157 ดังต่อไปนี้

    2.1 มิบ-ทู (MIB-II) ตามอาร์เอฟซี 1213

    2.2 อามอนมิบ (RMON MIB) ตามอาร์เอฟซี 1757

    2.3 ไพร์เวทมิบ (Private MIB) ของอุปกรณ์ ตามระบุในข้อที่ 3

3. ใช้ข้อมูลและสภาพแวดล้อมของระบบเครือข่ายซึ่งประกอบด้วยอุปกรณ์ ดังต่อไปนี้เป็นกรณีในการศึกษา

    3.1 อุปกรณ์จัดเส้นทาง (Router) ของซิสโก้ (Cisco) และ เบย์เน็ตเวิร์ก (Bay Networks)

    3.2 อีเทอร์เน็ตสวิช (Ethernet Switch) ของ ทรีคอม (3COM) และ ฟอร์ซิสเต็ม (FORE Systems)

    3.3 เอทีเอ็มสวิช (ATM Switch) ของ เบย์เน็ตเวิร์ก และ ฟอร์ซิสเต็ม

4. ไม่มีการเปลี่ยนแปลงแก้ไขหรือเพิ่มเติมระบบฮาร์ดแวร์และซอฟต์แวร์ ที่มีมาแต่เดิมกับอุปกรณ์ทางระบบเครือข่าย
5. ศึกษาในกรณีของอุปกรณ์ที่เชื่อมต่อกันอยู่ในระบบเครือข่ายภายในเท่านั้น มิได้รวมถึงระบบเครือข่ายอินเทอร์เน็ต
6. ระบบเครื่องที่ใช้

6.1 เครื่องแม่ข่าย

6.1.1 สภาพแวดล้อมฮาร์ดแวร์

6.1.1.1 หน่วยประมวลผลกลาง Pentium เป็นอย่างน้อย

6.1.1.2 หน่วยความจำหลัก 32 เมกกะไบต์ เป็นอย่างน้อย

6.1.1.3 ฮาร์ดดิสก์ ความจุ 800 เมกกะไบต์ เป็นอย่างน้อย

6.1.2 สภาพแวดล้อมซอฟต์แวร์

6.1.2.1 ระบบปฏิบัติการวินโดวส์เอ็นที รุ่น 4.0 เป็นอย่างน้อย

6.1.2.2 อินเทอร์เน็ตอินฟอร์เมชั่นเซิร์ฟเวอร์ (Internet Information Server) รุ่น 3.0 เป็นอย่างน้อย

6.1.2.3 ตัวแปรโปรแกรมภาษาเพิร์ล 5 สำหรับ วิน 32 (Perl 5 for Win32 Programming Language) เป็นอย่างน้อย

6.1.2.4 ตัวแปลโปรแกรมภาษาจาวา เวอร์ชั่น 1.02 (JAVA Development Kit Version 1.02 for Windows NT) เป็นอย่างน้อย

6.2 เครื่องลูกข่าย

6.2.1 สภาพแวดล้อมฮาร์ดแวร์

6.2.1.1 หน่วยประมวลผลกลาง 486 DX2-66 เป็นอย่างน้อย

6.2.1.2 หน่วยความจำหลัก (RAM) ความจุ 16 เมกกะไบต์ เป็นอย่างน้อย

6.2.1.3 ฮาร์ดดิสก์ ความจุ 200 เมกกะไบต์ เป็นอย่างน้อย

6.2.2 สภาพแวดล้อมซอฟต์แวร์

6.2.2.1 ระบบปฏิบัติการวินโดวส์ 95 เป็นอย่างน้อย

6.2.2.2 อินเทอร์เน็ตเอ็กซ์โพลเรอร์ (Internet Explorer) 3.0 หรือ เน็ตสเคปนาวิเกเตอร์ (Netscape Navigator) 3.0 เป็นอย่างน้อย

จุฬาลงกรณ์มหาวิทยาลัย

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษาและออกแแบบวิธีการจัดการอุปกรณ์ทางระบบเครือข่าย ดังนี้

    1.1 ศึกษาข้อกำหนดของโพรโทคอลเอสเอ็นเอ็มพีตามอาร์เอฟซี 1157

    1.2 ศึกษาโครงสร้างมิบ-ทูตามอาร์เอฟซี 1213 อามอนมิบตามอาร์เอฟซี 1757 และ ไพร์เวทมิบของอุปกรณ์ในระบบเครือข่ายที่ใช้เป็นกรณีศึกษา

    1.3 ศึกษารุ่น วิธีการเชื่อมต่อ ลักษณะการทำงาน วิธีใช้งาน และ โครงแบบ (Configuration) ของอุปกรณ์ในระบบเครือข่ายที่ใช้เป็นกรณีศึกษา

    1.4 วิเคราะห์ความหมายและความสัมพันธ์ระหว่างค่าออบเจกต์ ของมิบต่างๆ ที่รวบรวมได้จากอุปกรณ์

    1.5 ออกแบบวิธีการและแบบแผนในการคำนวณ และการนำเสนอรายงานข้อมูลเกี่ยวกับสภาพการทำงานของอุปกรณ์ และ กำหนดรูปแบบในการใช้คำสั่งควบคุมอุปกรณ์

2. วิเคราะห์ ออกแแบบ พัฒนา และทดสอบโปรแกรม
3. สรุปผลการทำวิทยานิพนธ์ และข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1. ได้โปรแกรมสำหรับใช้ในการจัดการอุปกรณ์ทางระบบเครือข่าย
2. เป็นแนวทางในการเปรียบเทียบและเลือกใช้ อุปกรณ์ทางระบบเครือข่าย และโปรแกรมประยุกต์สำหรับจัดการระบบเครือข่ายที่มีวิธีการ และ แบบแผนในการควบคุม และรายงานสภาพการทำงานของระบบเครือข่ายได้อย่างถูกต้องและเหมาะสม

จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

2.1 การจัดการระบบเครือข่าย (Network Management)

การจัดการระบบเครือข่าย สามารถจัดแบ่งออกเป็น 5 กิจกรรมขององค์กรมาตรฐาน (International Organization for Standardization หรือ ISO) เรียกว่าเอ็มเอฟเอ (MFA ย่อมาจาก Management Functional Areas) ด้วยวิธีการแบ่งงานที่ซับซ้อนในระบบเครือข่ายออกเป็นกิจกรรมย่อยจะช่วยทำให้การดำเนินงานควบคุมดูแลระบบเครือข่ายเป็นไปได้ด้วยความสะดวกและมีประสิทธิภาพ รายละเอียดของ 5 กิจกรรมมีดังนี้ [1] [2]

1. การแก้ไขข้อผิดพลาด (Fault Management) คือการแก้ไขปัญหาและกำหนดแนวทางป้องกันไม่ให้เกิดปัญหาหรือข้อผิดพลาดขึ้นในระบบเครือข่าย
2. การจัดการข้อมูลเกี่ยวกับการใช้งานของระบบเครือข่าย (Accounting Management) เป็นการควบคุมดูแล และรวบรวมข้อมูลเกี่ยวกับการใช้บริการของระบบเครือข่าย เช่น การจัดการบัญชีรายชื่อของผู้ใช้ การรวบรวมรายละเอียดเกี่ยวกับการใช้ทรัพยากรในเครือข่าย เป็นต้น
3. การกำหนดรูปแบบ และ วิธีการดำเนินงาน ของระบบเครือข่าย (Configuration Management) เป็นการวางแผนการติดตั้ง และ เปลี่ยนแปลงอุปกรณ์ทางระบบเครือข่าย การควบคุมและปรับปรุงแก้ไขการเชื่อมต่อของอุปกรณ์ ตลอดจนการติดตามรวบรวมข้อมูลเกี่ยวกับโครงสร้างและองค์ประกอบทางกายภาพ รวมถึงการจัดทำบัญชีรายชื่อของอุปกรณ์
4. การจัดการรักษาความปลอดภัยของระบบเครือข่าย (Security Management) เป็นการกำหนดมาตราการรักษาความปลอดภัยของระบบเครือข่าย เช่น การใช้อุปกรณ์จัดเส้นทาง (Router) และตัวป้องกันการบุกรุก (Firewall) เพื่อกลั่นกรองผู้ที่ต้องการติดต่อกับระบบเครือข่าย
5. การจัดการให้ระบบเครือข่ายดำเนินงานได้อย่างมีประสิทธิภาพ (Performance Management) คือการวัดประสิทธิภาพของระบบเครือข่าย โดยกำหนดการนำเสนอข้อมูลเกี่ยวกับประสิทธิภาพในรูปแบบทางสถิติต่างๆ เช่น ความสามารถในการให้บริการรับส่งข้อมูล (Throughput) ระยะเวลาที่ระบบสามารถให้บริการได้ (Availability) และปริมาณการใช้บริการของระบบเครือข่าย (Utilization)

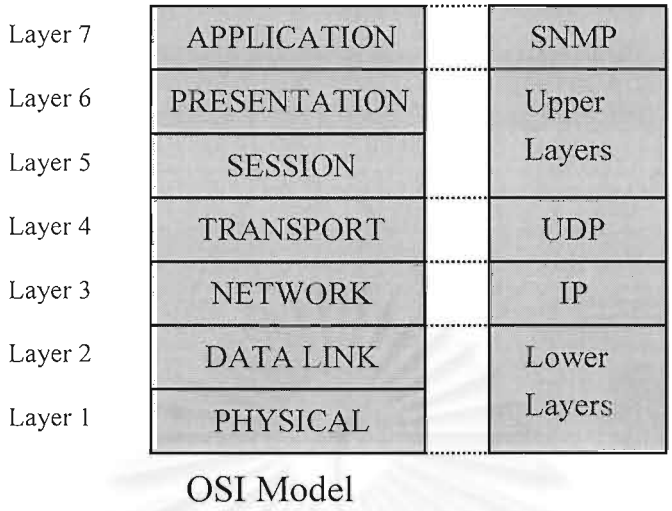
ในทางปฏิบัติ ขอบเขตในการดำเนินกิจกรรมทั้ง 5 ถูกจำกัดด้วยปัจจัยดังต่อไปนี้ [3]

1. องค์กรได้กำหนดนโยบายจัดการระบบเครือข่ายไว้อย่างไร ดังนี้
   1) ไม่มีการดูแลและแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย (Inactive)
   2) ไม่มีการดูแลระบบเครือข่าย แต่จะมีการดำเนินการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย (Reactive)
   3) มีการดูแลและแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย (Active)
   4) มีการดูแล แก้ไขและป้องกันไม่ให้เกิดข้อผิดพลาดขึ้นในระบบเครือข่าย (Proactive)
2. จำนวน และประสบการณ์ ความรู้ของบุคลากรควบคุม ดูแล และติดตามสภาพการทำงานของระบบเครือข่าย
3. มีการนำอุปกรณ์ฮาร์ดแวร์ และ ซอฟต์แวร์มาช่วยจัดการระบบเครือข่าย เช่น โปรแกรมประยุกต์จัดการระบบเครือข่าย และเครื่องมือทดสอบการทำงานของอุปกรณ์ทางระบบเครือข่าย เป็นต้น

2.2 โพรโทคอลเอสเอ็นเอ็มพี (Simple Network Management Protocol)

ไออีทีเอฟ (IETF ย่อมาจาก Internet Engineering Task Force) ภายใต้ไอเอบี (IABย่อมาจาก Internet Activities Board) ได้มีการศึกษาหาแนวทางจัดการอุปกรณ์ทางระบบเครือข่ายที่หลากหลาย ต่างชนิดและต่างผู้ผลิต ให้มีรูปแบบของการจัดการภายใต้ข้อกำหนดของมาตรฐานเดียวกัน ในปี ค.ศ.1990 ไออีทีเอฟ ได้เสนอโพรโทคอลเอสเอ็นเอ็มพี (SNMP ย่อมาจาก Simple Network management Protocol) เป็นมาตรฐานอินเทอร์เน็ต (Internet Standard) เพื่อใช้เป็นข้อกำหนดในการสร้างอุปกรณ์ทางระบบเครือข่าย รวมถึงการพัฒนาซอฟต์แวร์ที่เกี่ยวข้อง ให้สามารถรองรับวิธีจัดการภายในกรอบที่เป็นมาตรฐานเดียวกัน ในการจัดแบ่งชั้นต่างๆของโอเอสไอโมเดล (OSI Model) ขององค์กรมาตรฐาน โพรโทคอลเอสเอ็นเอ็มพีจะทำงานอยู่ในชั้นที่ 7 หรืออยู่ในชั้นของแอ็พพริเคชั่นเลเยอร์ (Application Layer) โดยใช้บริการของ ยูดีพีโพรโทคอล (UDP Protocol หรือ User Datagram Protocol) ในชั้นของทรานสปอตร์เลเยอร์ (Transport Layer) และใช้ ไอพีโพรโทคอล (IP Protocol) ในชั้นของเน็ตเวิร์กเลเยอร์ (Network Layer) เป็นวิธีการรับส่งข้อมูล [1] ดังแสดงในรูปที่ 2.1 ยูดีพีโพรโทคอล ได้ถูกเลือกใช้กับโพรโทคอลเอสเอ็นเอ็มพี เพื่อช่วยให้โปรแกรมจัดการระบบเครือข่ายทำงานได้อย่างมีประสิทธิภาพมากขึ้น เนื่องจากยูดีพีโพรโทคอลมีรูปแบบของการทำงานที่เป็นคอนเน็คชั่นเลส (Connectionless) ซึ่งมีลักษณะการรับส่งข้อมูลโดยไม่ต้องมีการยืนยันในการดำเนินการรับส่งข้อมูล ไม่ต้องรับผิดชอบในส่วนของการตรวจสอบข้อผิด

พลาดของข้อมูล และสามารถเปลี่ยนเส้นทางของการส่งข้อมูลได้ การตรวจสอบและแก้ไขข้อผิดพลาดของข้อมูลจะปล่อยให้เป็นหน้าที่ของชั้นที่สูงกว่า



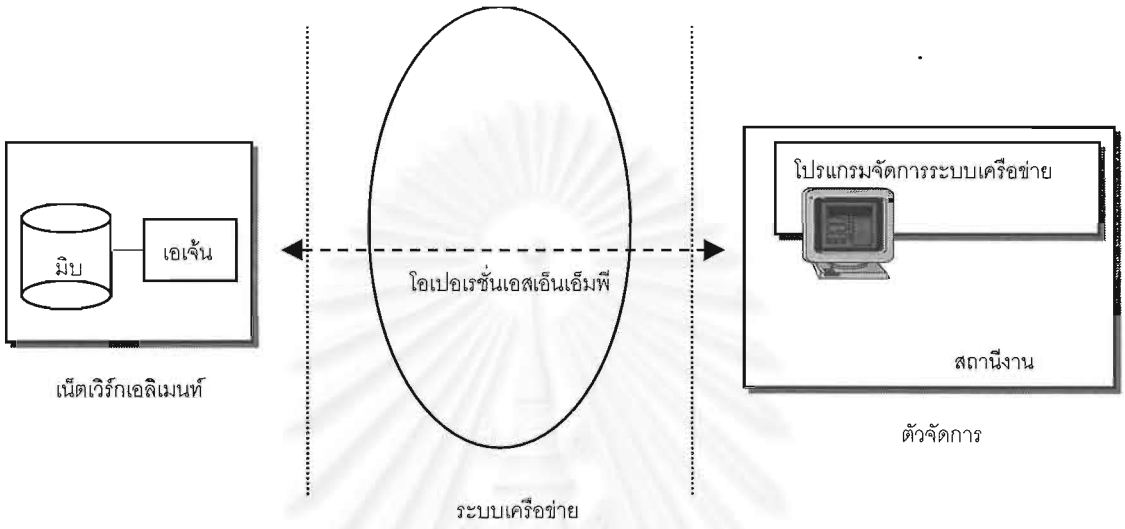
รูปที่ 2.1 แสดงโพรโทคอลเอสเอ็นเอ็มพีและชั้นที่เกี่ยวข้องในโอเอสไอโมเดล

โพรโทคอลเอสเอ็นเอ็มพี ถูกจัดเป็นส่วนหนึ่งของสถาปัตยกรรมของ การจัดการเครือข่ายอินเทอร์เน็ต (Internet Network Management Architecture) แสดงในรูปที่ 2.2 โดยมีรายละเอียดขององค์ประกอบที่สำคัญดังนี้ [4]

1. เอเจ้น (Agent) คือ ซอฟต์แวร์ที่ทำงานอยู่ในเน็ตเวิร์กเอลิเมนท์ (Network Element) เอเจ้นมีหน้าที่รายงานสถานะของอุปกรณ์ต่อตัวจัดการ (Manager) ในทางปฏิบัติ เน็ตเวิร์กเอลิเมนท์ คืออุปกรณ์ทางเครือข่าย (Network Device) เช่น อุปกรณ์จัดเส้นทาง (Router) อีเทอร์เน็ตสวิช (Switch) และ ฮับ (Hub) เป็นต้น
2. ตัวจัดการของเน็ตเวิร์กเอลิเมนท์ เรียกว่า เอ็นเอ็มเอส (NMS ย่อมาจาก Network Management Station) ในทางปฏิบัติ ตัวจัดการ คือ โปรแกรมจัดการระบบเครือข่าย (Network Management Program) ซึ่งทำงานอยู่บนสถานีงาน (Workstation) ที่เชื่อมต่อกับเครือข่ายของอุปกรณ์ต่างๆ ที่ต้องการดูแล โดยมีหน้าที่รวบรวมข้อมูลเกี่ยวกับสภาพการทำงานของอุปกรณ์จากเอเจ้น เพื่อนำไปประมวลผล จัดรูปแบบ และนำเสนอข้อมูลให้เป็นไปตามขอบเขตของความสามารถของโปรแกรม
3. มิบ (MIB หรือ Management Information Base) คือ ข้อมูลพื้นฐานของการจัดการอุปกรณ์ มิบมีโครงสร้างประกอบไปด้วย ออบเจกต์ (Object) ซึ่งเป็นรายละเอียดเกี่ยวกับคุณสมบัติ สถานะ สภาพการทำงาน หรือ ข้อมูลเกี่ยวกับความสามารถในการให้บริการรับส่งข้อมูลของเน็ตเวิร์กเอลิเมนท์ แต่ละออบเจกต์มีหมายเลขประจำตัวเรียกว่า

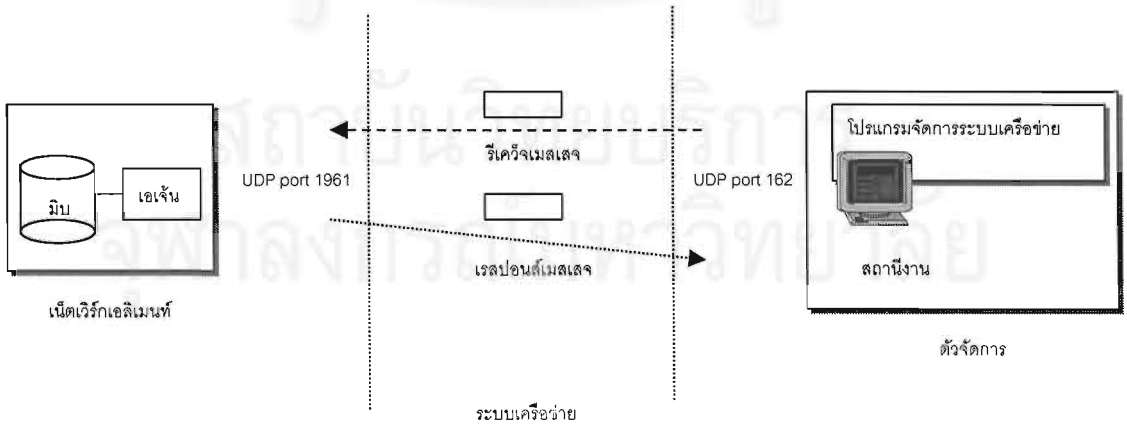
หมายเลขออบเจกต์ (Object Identifier หรือ OID) และค่า (Value) ของออบเจกต์เรียกว่า ออบเจกต์อินสแตนส์ (Object Instance)

4. เกณฑ์วิธีสำหรับการจัดการ (Management Protocol) เป็นข้อกำหนด หรือวิธีทำงานร่วมกันระหว่างเอเจ้น และตัวจัดการเรียกว่าโพรโทคอลเอสเอ็นเอ็มพี และวิธีการติดต่อรับส่งข้อมูลระหว่างตัวจัดการและเอเจ้น เรียกว่าโอเปอเรชั่น (Operation)



รูปที่ 2.2 แสดงองค์ประกอบของ สถาปัตยกรรมของการจัดการระบบเครือข่ายอินเทอร์เน็ต

ในการทำงานของโพรโทคอล เอสเอ็นเอ็มพี ตัวจัดการใช้หมายเลขช่องทางยูดีพี 1961 (UDP Port 1961) และเอเจ้นใช้หมายเลขช่องทางยูดีพี 161 (UDP Port 161) ในการสร้าง "การติดต่อแต่ละครั้ง" หรือ เซสชั่น (Session) เพื่อรับส่งข้อมูลระหว่างกัน เรียกว่าเอสเอ็นเอ็มพีเมสเสจ (SNMP Message) ดังแสดงในรูปที่ 2.3

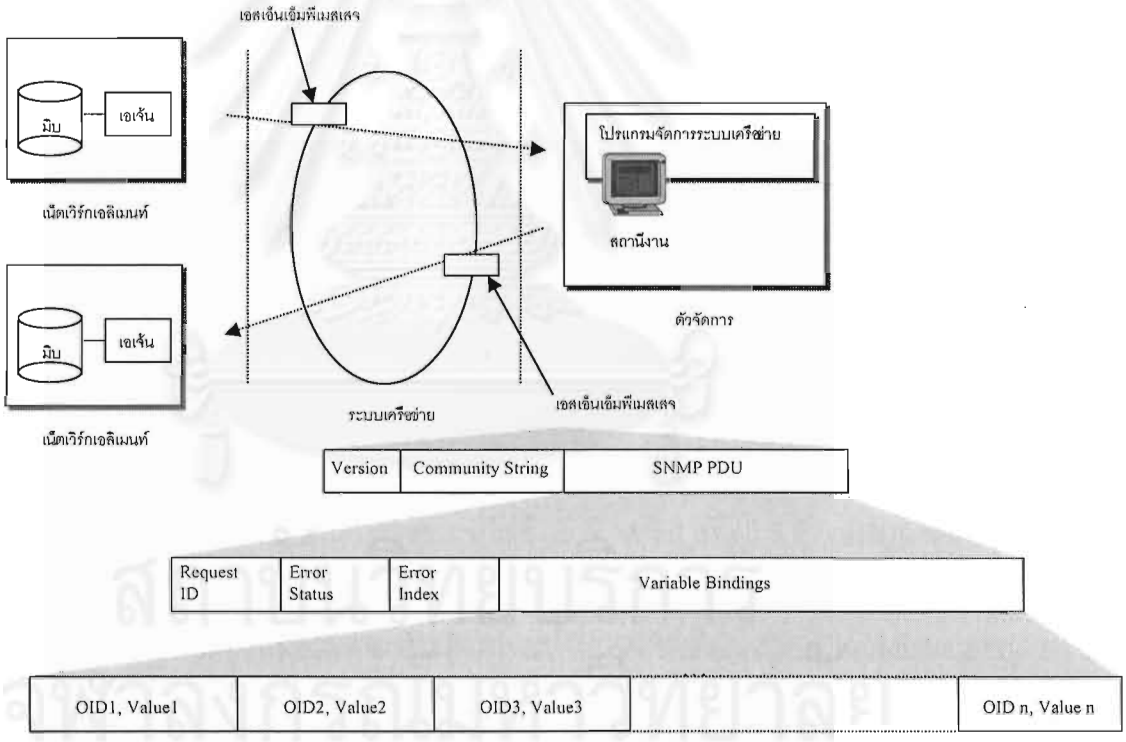


รูปที่ 2.3 แสดงการรับส่งเอสเอ็นเอ็มพีเมสเสจ

2.3 **เอสเอ็นเอ็มพี เมสเสจ** (SNMP Message)

กลุ่มข้อมูล หรือ แพ็กเก็ต (Packet) ของโพรโทคอลเอสเอ็นเอ็มพี เรียกว่า เอสเอ็นเอ็มพี เมสเสจ โดยมีโครงสร้างประกอบไปด้วย 2 ส่วน [5] [6] ส่วนแรกเป็นรายละเอียดของโพรโทคอลเอสเอ็นเอ็มพี ประกอบด้วยเขตข้อมูลเวอร์ชั่น (Version) ซึ่งจะมีค่าเป็น 1 สำหรับโพรโทคอลเอสเอ็นเอ็มพีเวอร์ชั่น 1 (SNMP Version 1) และ เขตข้อมูลคอมมิวนิตี้สตริง (Community String) ทำหน้าที่เป็นรหัสผ่านเพื่อป้องกันการติดต่อจากตัวจัดการที่ไม่มีหน้าที่เกี่ยวข้อง ส่วนที่สองเรียกว่าเอสเอ็นเอ็มพีพีดียู (SNMP PDU ย่อมาจาก SNMP Protocol Data Unit) มีอยู่ 5 ประเภท คือ

1. เก็ตพีดียู (Get PDU)
2. เก็ต-เน็กส์พีดียู (Get-Next PDU)
3. เก็ต-เรสปอนส์ พีดียู (Get-Response PDU)
4. เซ็ตพีดียู (Set PDU)
5. แทร็ปพีดียู (Trap PDU)



รูปที่ 2.4 แสดงโครงสร้างของเอสเอ็นเอ็มพีเมสเสจ

Enterprise	Agent Addr	Generic Trap	Specific Trap	Time Stamp	Variable Bindings

รูปที่ 2.5 แสดงโครงสร้างของแทร็ปพีดียู

เอสเอ็นเอ็มพีพีดียู ประกอบไปด้วยเขตข้อมูล (Field) แสดงในรูปที่ 2.4 และ 2.5 โดยมีรายละเอียดดังตารางที่ 2.1

เขตข้อมูล	รายละเอียด
1. รีเควัสไอดี (Request ID)	ใช้เรียงลำดับการรับส่งพีดียูให้ถูกต้อง
2. สถานะข้อผิดพลาด (Error Status)	ใช้ระบุประเภทของข้อผิดพลาดที่เกิดขึ้น
3. ดรรชนีข้อผิดพลาด (Error Index)	ใช้งานร่วมกับเขตข้อมูลสถานะข้อผิดพลาดเพื่อระบุออบเจกต์ที่เป็นตัวการทำให้เกิดข้อผิดพลาดขึ้น
4. แวเรียเบิลบายดิ้ง (Variable Bindings)	มีโครงสร้างแบบอธิบายตนเอง ประกอบด้วยหมายเลขออบเจกต์ และค่าของออบเจกต์
5. เอ็นเตอร์ไพร้ (Enterprise)	ใช้ระบุชนิดของเอเจ้น ที่ได้ส่งแทร็ปพีดียู
6. ที่อยู่เอเจ้น (Agent Addr)	ใช้ระบุหมายเลขไอพีของเอเจ้น ที่ได้ส่งแทร็ปพีดียู
7. จีนเนอริกแทร็ป (Generic Trap)	ใช้กำหนดชนิดของแทร็ปพีดียู
8. สเป็กสิฟิกแทร็ป (Specific Trap)	ใช้กำหนดข้อมูลจำเพาะของแทร็ปพีดียู
9. ไทม์สแต็มป์ (Time Stamp)	ระยะเวลาที่ตัวเอเจ้น สามารถให้บริการตั้งแต่อุปกรณ์ได้เริ่มทำงานครั้งล่าสุด

ตารางที่ 2.1 แสดงรายละเอียดของเอสเอ็นเอ็มพีพีดียู

เก็ตพีดียู เก็ต-เน็กส์พีดียูและเซ็ตพีดียู ถูกเรียกโดยรวมว่ารีเควัสพีดียู (Request PDU) ซึ่งตัวจัดการสามารถส่งรีเควัสพีดียูเหล่านี้ได้หมด ในขณะเดียวกันเอเจ้นสามารถส่งเรสปอนส์พีดียู และ แทร็ปพีดียูเท่านั้น

เก็ตพีดียู เก็ต-เน็กส์พีดียู เก็ต-เรสปอนส์พีดียู และ เซ็ตพีดียู ทั้งหมดมีโครงสร้างเหมือนกัน แต่เขตข้อมูลที่ 2 และ เขตข้อมูลที่ 3 จะมีค่าเป็น 0 เสมอสำหรับเก็ตพีดียู เก็ต-เน็กส์พีดียู และ เซ็ตพีดียู ส่วนแทร็ปพีดียูจะมีโครงสร้างแตกต่างจากพีดียูอื่นคือ เขตข้อมูลที่ 5 6 7 8 และ 9 จะมีใช้ในแทร็ปพีดียูเท่านั้น (ดูรูปที่ 2.5)

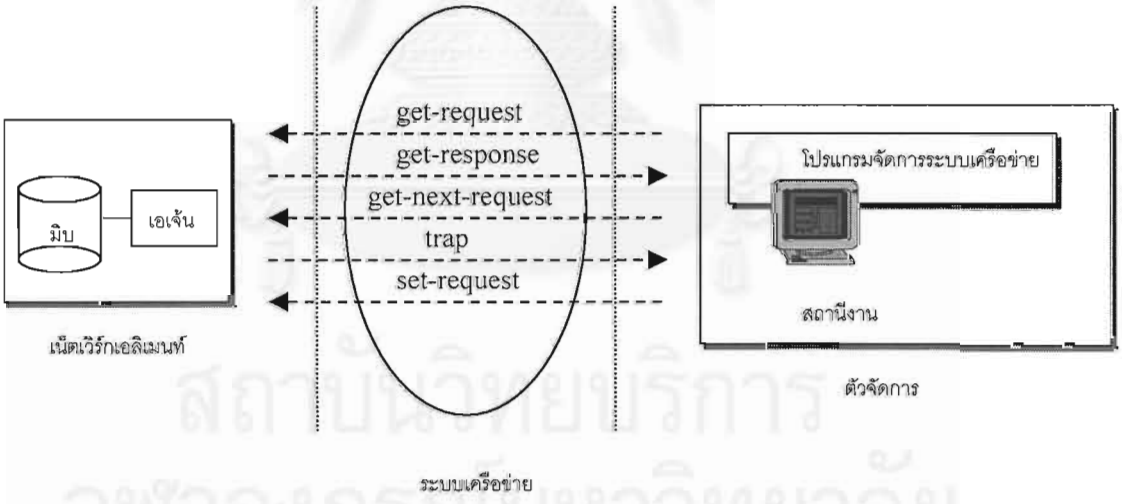
รูปแบบของเอสเอ็นเอ็มพีเมสเสจ (SNMP Message Formats) อยู่ในข้อกำหนดของไออีทีเอฟ เรียกว่าอาร์เอฟซี 1157 [7]

2.4 โอเปอเรชั่น เอสเอ็นเอ็มพี (SNMP Operation)

การติดต่อ รับส่งข้อมูลระหว่างตัวจัดการ และ เอเจ้นเรียกว่าโอเปอเรชั่น ตัวจัดการเรียกใช้เอเจ้นโดยการส่งเอสเอ็นเอ็มพีเมสเสจ ผ่าน 3 โอเปอเรชั่น ส่วนเอเจ้นสามารถติดต่อกับตัวจัดการด้วย 2 โอเปอเรชั่น รายละเอียดของ 5 โอเปอเรชั่น [5] [6] เหล่านี้แสดงดังตารางที่ 2.2

โอเปอเรชั่น	รายละเอียด
1. เก็ต (Get) คือ การเรียกให้เอเจ้นส่งค่าของออบเจกต์ประเภทค่าเดี่ยว (Scalar Object) ซึ่งมีค่าได้เพียง 1 ค่า	ตัวจัดการส่ง เก็ตพีดียู (Get PDU) โดยจะระบุออบเจกต์ที่ต้องการในเขตข้อมูลแวเรียเบิลบายดิ้ง และเอเจ้นจะตอบด้วย เก็ต-เรสปอนส์พีดียู (Get-Response PDU)
2. เก็ต-เน็กส์ (Get-Next) คือ วิธีการติดต่อให้เอเจ้นส่งค่าของออบเจกต์ประเภทตาราง (Tabular Object) ซึ่งมีค่าได้มากกว่า 1 ค่า	ตัวจัดการส่งเก็ต-เน็กส์พีดียู (Get-Next PDU) จนกว่าได้รับค่าของออบเจกต์ทั้งหมด
3. เก็ต-เรสปอนส์ (Get-Response) คือ การติดต่อตอบกลับของเอเจ้น	เอเจ้นส่งค่าของออบเจกต์ไปในเขตข้อมูลแวเรียเบิลบายดิ้ง ของเก็ต-เรสปอนส์พีดียู
4. เซ็ต (Set) เป็นการเปลี่ยนแปลงค่าของออบเจกต์	ตัวจัดการส่งเซ็ตพีดียู (Set PDU) เพื่อระบุค่าใหม่ของออบเจกต์ และ เอเจ้นจะตอบด้วยเก็ต-เรสปอนส์พีดียู (Get-Response PDU)
5. แทร็ป (Trap) คือวิธีการที่เอเจ้นส่งข้อมูลแจ้งตัวจัดการเมื่อเกิดเหตุการณ์ (Event) บางอย่างขึ้น ซึ่งทำให้ค่าของออบเจกต์ได้เปลี่ยนไป	เอเจ้นส่งแทร็ปพีดียู (Trap PDU) เพื่อรายงานค่าใหม่ของออบเจกต์ โดยจะถูกส่งมาที่หมายเลขช่องทางยูดีพี 162 (UDP Port 162) ของตัวจัดการ

ตารางที่ 2.2 แสดงรายละเอียดของเอสเอ็นเอ็มพี โอเปอเรชั่น

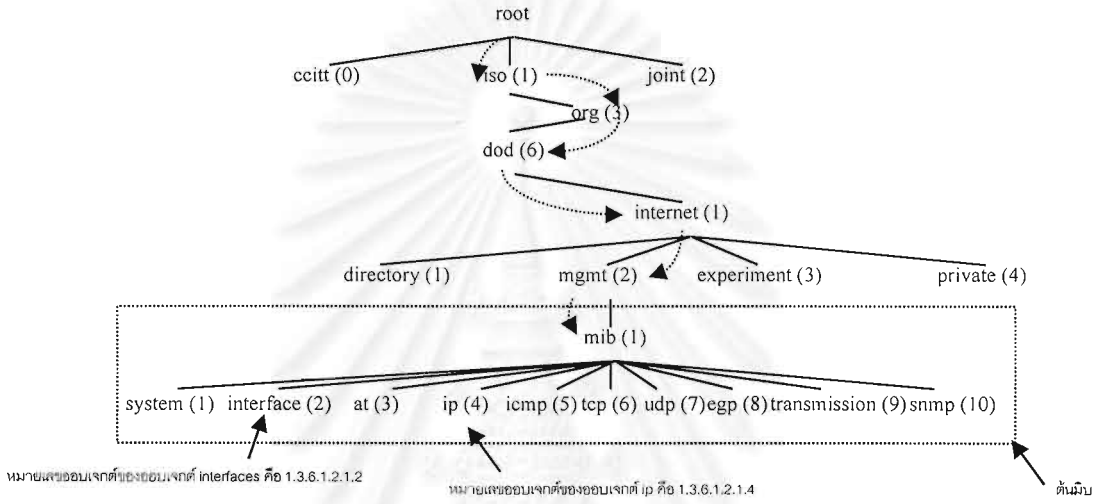


รูปที่ 2.6 แสดงรายละเอียดของเอสเอ็นเอ็มพี โอเปอเรชั่น

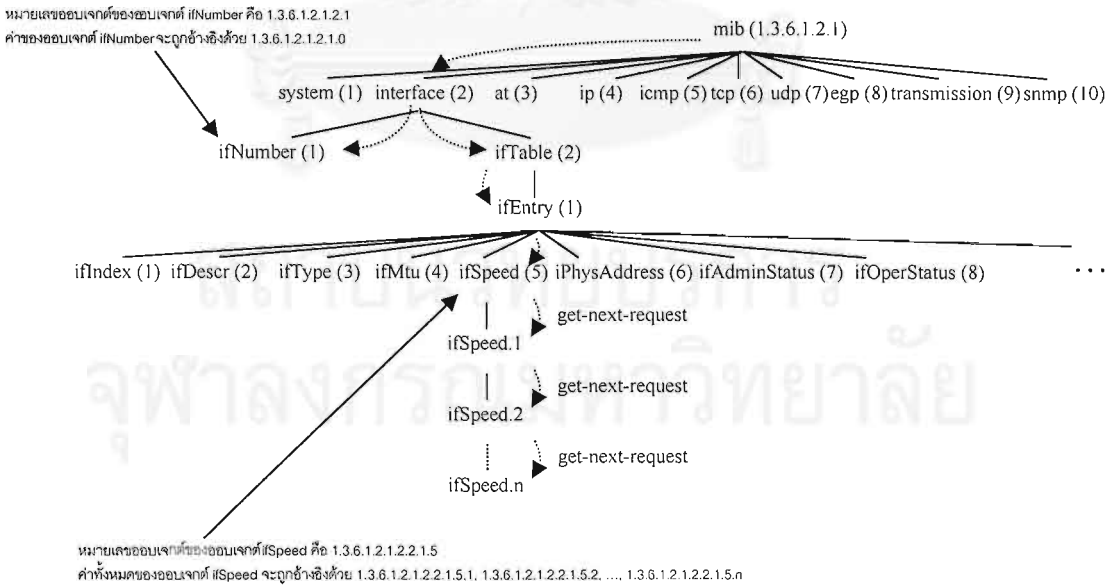
2.5 มิบ (MIB)

อุปกรณ์ทางระบบเครือข่าย ที่มีมาตรฐานการจัดการตามโพรโทคอลเอสเอ็นเอ็มพี ต้องใช้มิบในการอ้างอิงถึงออบเจกต์ ซึ่งเป็นชิ้นส่วนต่างๆของอุปกรณ์ที่ต้องการควบคุมดูแล โครงสร้างของมิบเรียกว่าต้นมิบ (MIB Tree) ถูกจัดเป็นส่วนหนึ่งของอินเทอร์เน็ตซับ-ทรี (Internet Sub-Tree)

ในโอเอสไอไดเร็คเตอรีทรี (OSI Directory Tree) ขององค์กรมาตรฐาน [2] [6] ต้นมิบมีคุณสมบัติเป็นแบบลำดับชั้น (Hierarchical) แต่ละโนด (node) ของต้นมิบเรียกว่าออบเจกต์ โดยมีหมายเลขประจำตำแหน่ง และถูกอ้างอิงด้วยหมายเลขประจำตัวเรียกว่า หมายเลขออบเจกต์ซึ่งประกอบด้วยตัวเลขชุดหนึ่งแบ่งออกเป็นหลายส่วน โดยคั่นแต่ละส่วนด้วยจุด เพื่อความง่ายในการอ่านและเขียน หมายเลขออบเจกต์จะสัมพันธ์กับตำแหน่งของออบเจกต์ในต้นมิบ เช่น ออบเจกต์ ip มีหมายเลขประจำตำแหน่งเป็น 4 และหมายเลขออบเจกต์เป็น 1.3.6.1.2.1.4 ออบเจกต์ interfaces มีหมายเลขประจำตำแหน่งเป็น 2 และหมายเลขออบเจกต์เป็น 1.3.6.1.2.1.2 ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 แสดงออบเจกต์ในต้นมิบ



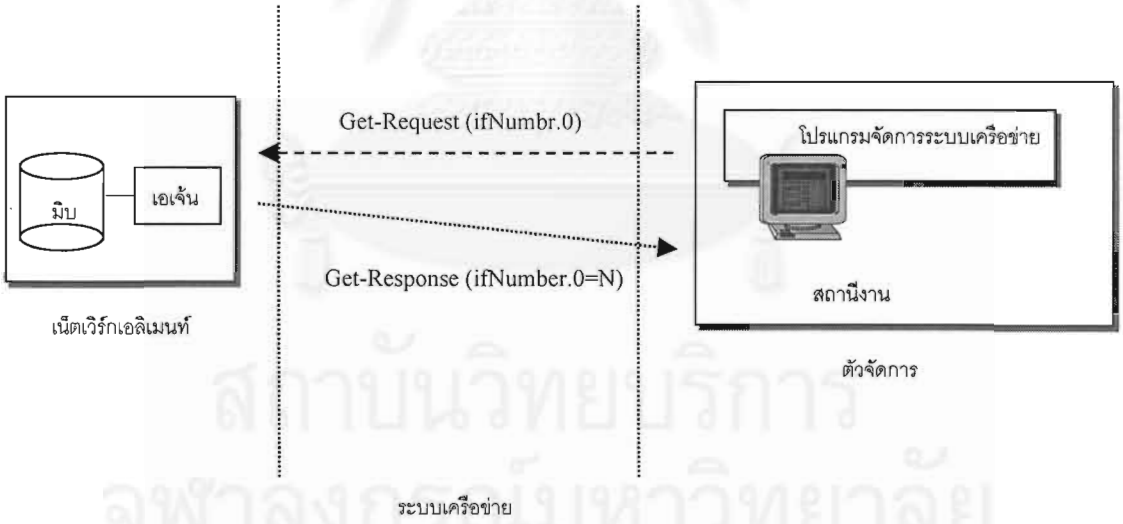
รูปที่ 2.8 แสดงออบเจกต์ ifNumber และออบเจกต์ ifSpeed

ออบเจกต์ที่ไม่อยู่ในตำแหน่งของใบในต้นมิบ จะไม่มีค่าออบเจกต์ เช่น ออบเจกต์ ip และออบเจกต์ interfaces รูปที่ 2.8 แสดงตัวอย่างของออบเจกต์ ifNumber ซึ่งเป็นใบของต้นมิบ และมีค่าได้เพียง 1 ค่า (Scalar Object) โดยมีหมายเลขออบเจกต์ 1.3.6.1.2.1.2.1 และ ค่าออบเจกต์จะถูกอ้างอิงด้วยหมายเลข 1.3.6.1.2.1.2.1.0 เพื่อความง่ายเราสามารถใช้ชื่อออบเจกต์ ifNumber แทนหมายเลขออบเจกต์ และอ้างอิงค่าออบเจกต์นี้ด้วย ifNumber.0 สำหรับออบเจกต์ ifSpeed เป็นตัวอย่างของออบเจกต์ที่เป็นใบ และเป็นออบเจกต์ที่มีโครงสร้างแบบตาราง (Tabular Object) ประกอบไปด้วยหมายเลขออบเจกต์ย่อยหลายค่า เพื่อใช้ในการอ้างอิงค่าออบเจกต์ ซึ่งมีมากกว่า 1 ค่า เช่น หมายเลขออบเจกต์ของออบเจกต์ ifSpeed คือ 1.3.6.1.2.1.2.2.1.5 และแต่ละค่าของออบเจกต์จะถูกอ้างอิงด้วยหมายเลขออบเจกต์ย่อย ดังนี้

1.3.6.1.2.1.2.2.1.5.1, 1.3.6.1.2.1.2.2.1.5.2, ... , 1.3.6.1.2.1.2.2.1.5.n

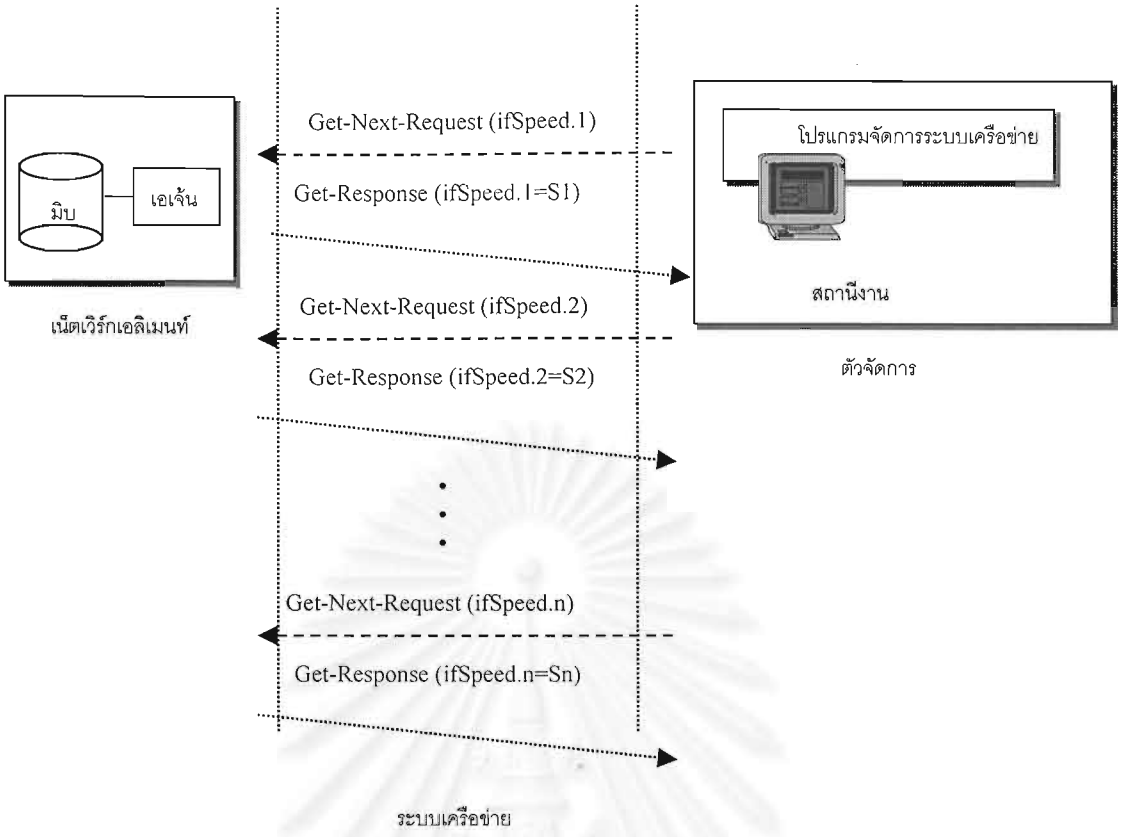
โดย n เป็นจำนวนของค่าในตารางของออบเจกต์ ifSpeed เพื่อความง่าย เราสามารถอ้างอิงค่าออบเจกต์เหล่านี้ด้วย ifSpeed.1, ifSpeed.2, ... , ifSpeed.n และใช้ชื่อออบเจกต์ ifSpeed แทนหมายเลขออบเจกต์

โอเปอเรชั่นสำหรับใช้ในการอ้างอิงถึงตำแหน่งของออบเจกต์ที่ต้องการ คือ เก็ต และเก็ต-เน็กส์ ซึ่งได้กล่าวมาแล้วข้างต้น เช่น ถ้าต้องการทราบค่าของออบเจกต์ประเภทค่าเดี่ยว เช่น ออบเจกต์ ifNumber เราจะต้องเรียกใช้โอเปอเรชั่นเก็ต แสดงในรูปที่ 2.9



รูปที่ 2.9 แสดง โอเปอเรชั่นเก็ตเพื่ออ่านออบเจกต์ประเภทค่าเดี่ยว

ในกรณีที่ต้องการทราบค่าต่างๆของ ออบเจกต์ประเภทตาราง เช่น ออบเจกต์ ifSpeed เราต้องใช้ โอเปอเรชั่นเก็ต-เน็กส์ เพื่ออ่านค่าเหล่านี้ โดยอ่านทีละค่าตามลำดับ ดังแสดงในรูปที่ 2.10



รูปที่ 2.10 แสดง โอเปอเรชั่นเก็ต-เน็กส์ เพื่ออ่านค่าของออบเจกต์ประเภทตาราง

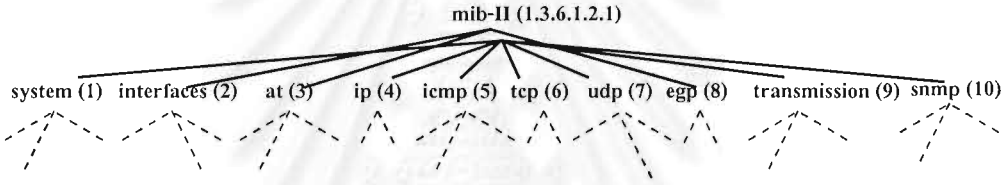
โครงสร้างของออบเจกต์ (Structure of Management Information หรือ SMI) อยู่ในข้อกำหนดของไออีทีเอฟ เรียกว่าอาร์เอฟซี 1155 [8]

2.6 มิบ-ทู (MIB-II)

มิบที่ได้กล่าวมาแล้วข้างต้น มีชื่อเรียกเป็นทางการว่า มิบ-ทู ซึ่งเป็นมิบมาตรฐานในข้อกำหนดของไออีทีเอฟ เรียกว่าอาร์เอฟซี 1213 [9] องค์ประกอบของมิบ-ทู ได้ถูกจัดแบ่งออกเป็น 10 กลุ่มของออบเจกต์ เรียกว่ากลุ่มมิบ-ทู (MIB-II Group) ตามคุณสมบัติของฮาร์ดแวร์ และซอฟต์แวร์ของเน็ตเวิร์กเอลิเมนท์ [5] ซึ่งได้แสดงในรูปที่ 2.11 ดังนี้

1. กลุ่มซิสเต็ม (System Group) คือ ข้อมูลโดยรวม ของเน็ตเวิร์กเอลิเมนท์ เช่น ชื่อ รุ่น
2. กลุ่มอินเตอร์เฟส (Interfaces Group) คือ ข้อมูลเกี่ยวกับส่วนที่ใช้เชื่อมต่อกับระบบเครือข่าย (Interfaces) เช่น จำนวน ชนิด และสถานะของส่วนเชื่อมต่อกับระบบเครือข่าย
3. กลุ่มเอที (AT Group) คือ ข้อมูลเกี่ยวกับโพรโทคอลเอที (Address Translation Protocol) โดยรวบรวมหมายเลขที่แท้จริง (MAC address) และ หมายเลขไอพี (IP Address) ของอุปกรณ์

4. กลุ่มไอพี (IP group) คือ ข้อมูลเกี่ยวกับการใช้โพรโทคอลไอพี (Internet Protocol)
5. กลุ่มไอซีเอ็มพี (ICMP Group) คือ ข้อมูลที่เกี่ยวกับการใช้โพรโทคอลไอซีเอ็มพี (Internet Control Message Protocol)
6. กลุ่มทีซีพี (TCP Group) คือ ข้อมูลเกี่ยวกับการใช้โพรโทคอลทีซีพี (Transmission Control Protocol)
7. กลุ่มยูดีพี (UDP Group) คือ ข้อมูลเกี่ยวกับการใช้โพรโทคอลยูดีพี (Datagram Protocol)
8. กลุ่มอีจีพี (EGP Group) คือ ข้อมูลเกี่ยวกับการใช้โพรโทคอลอีจีพี (EGP Protocol)
9. กลุ่มทรานสมิชชั่น (Transmission Group) คือ ข้อมูลในส่วนของวิธีการรับส่งข้อมูลในแต่ละอินเตอร์เฟสของอุปกรณ์
10. กลุ่มเอสเอ็นเอ็มพี (SNMP Group) คือ ข้อมูลเกี่ยวกับการทำงานของโพรโทคอลเอสเอ็นเอ็มพี เช่น สถิติของการรับส่งเอสเอ็นเอ็มพีแพ็กเก็ต



รูปที่ 2.11 แสดงกลุ่มมิบ-ทู

2.7 ไพร์เวทมิบ (Private MIB)

ไพร์เวทมิบคือมิบที่ไม่ได้ถูกสร้างขึ้นมาโดยไออีทีเอฟแต่จะถูกออกแบบและสร้างโดยผู้ผลิตอุปกรณ์ทางระบบเครือข่ายเพื่อเพิ่มรายละเอียดและความสามารถในการจัดการอุปกรณ์ของตน ในทางปฏิบัติอุปกรณ์ทางระบบเครือข่ายที่มีไพร์เวทมิบ จะต้องใช้โปรแกรมจัดการระบบเครือข่ายจากผู้ผลิตอุปกรณ์ตัวนั้น หรือต้องแก้ไขโปรแกรมจัดการระบบเครือข่ายที่มีอยู่ เพื่อจัดเตรียมให้สามารถนำไพร์เวทมิบมาประมวลผล จัดรูปแบบ และนำเสนอข้อมูลให้เป็นไปตามต้องการได้อย่างถูกต้อง

2.8 อามอนมิบ (RMON MIB)

มิบ-ทูให้ข้อมูลในรายละเอียดส่วนตัวของแต่ละอุปกรณ์เท่านั้น มิบ-ทูไม่สามารถให้ข้อมูลรวมเกี่ยวกับสถานภาพของเครือข่ายที่อุปกรณ์เชื่อมต่อกันอยู่ อามอนมิบ (RMON MIB หรือ Remote Network Monitoring MIB) เป็นมิบที่ได้ถูกออกแบบขึ้นเพื่อให้ข้อมูลทางสถิติของระบบ

เครือข่าย ซึ่งช่วยให้เราเห็นภาพรวมของสถานภาพของเครือข่าย อามอนมิบ ประกอบไปด้วย 10 กลุ่มของออบเจกต์ เรียกว่ากลุ่มอามอน (RMON Group) [5] ดังนี้

1. กลุ่มสถิติ (Statistics Group) คือปริมาณการใช้บริการของระบบเครือข่าย (Utilization) และสถิติของความผิดพลาด (Error Statistics) ที่เกิดขึ้นในเครือข่าย
2. กลุ่มประวัติ (History Group) รวบรวมและเก็บประวัติเกี่ยวกับข้อมูลของกลุ่มสถิติ
3. กลุ่มแจ้งเหตุ (Alarm Group) รวบรวม และจัดเก็บรายละเอียดเกี่ยวกับความผิดปกติของระบบเครือข่าย
4. กลุ่มโฮส (Host Group) รวบรวมและจัดเก็บรายละเอียดเกี่ยวกับปริมาณการรับส่งข้อมูลของอุปกรณ์ที่เชื่อมต่อกันอยู่ในเครือข่าย
5. กลุ่มโฮสท็อปเอ็น (HostTopN Group) รวบรวมและจัดอันดับอุปกรณ์ตามค่าทางสถิติต่างๆ
6. กลุ่มเมตริกซ์ (Matrix Group) รวบรวมและจัดเก็บรายละเอียดเกี่ยวกับปริมาณการรับส่งข้อมูลระหว่าง 2 อุปกรณ์
7. กลุ่มฟิวเตอร์ (Filter Group) รวบรวมและจัดเก็บรายละเอียดเกี่ยวกับแพ็กเก็ตประเภทต่างๆ ให้เป็นไปตามต้องการ
8. กลุ่มแพ็กเก็ตแค็ปเจอร์ (Packet Capture Group) ควบคุมการนำเสนอข้อมูลให้แก่ตัวจัดการ
9. กลุ่มเหตุการณ์ (Event Group) รวบรวมและจัดเก็บรายละเอียดเกี่ยวกับเหตุการณ์ที่ได้เกิดขึ้น
10. กลุ่มโทเค็นริง (TokenRing Group) รวบรวมและจัดเก็บรายละเอียดเกี่ยวกับเครือข่ายวงแหวนโทเค็น

อามอนมิบ อยู่ในข้อกำหนดของไออีทีเอฟ เรียกว่าอาร์เอฟซี 1757 (RFC 1757) [10]

2.9 องค์ประกอบของโปรแกรมจัดการระบบเครือข่าย

โปรแกรมจัดการระบบเครือข่ายที่ใช้งานอยู่ในปัจจุบันมีอยู่มากมายและแตกต่างกันไปตามผู้พัฒนาของแต่ละค่าย อย่างไรก็ตามโปรแกรมเหล่านี้มีโครงแบบที่สามารถจัดแบ่งองค์ระกอบออกเป็น 3 ส่วนหลักดังนี้ [5]

1. ส่วนนำเสนอผู้ใช้งาน (User Presentation Software) มีหน้าที่จัดรูปแบบและนำเสนอข้อมูลเกี่ยวกับการจัดการระบบเครือข่าย รวมถึงจัดการในเรื่องของการติดต่อเชื่อมโยงผู้ใช้ กับโปรแกรมจัดการระบบเครือข่าย

2. ส่วนจัดการระบบเครือข่าย (Network Management Software) มีหน้าที่ควบคุมวิธีการจัดการระบบเครือข่าย ให้เป็นไปตามรายละเอียดของแต่ละกิจกรรมในเอ็มเอฟเอขององค์กรมาตรฐาน โดยระดับความสามารถของโปรแกรมในกิจกรรมเหล่านี้จะขึ้นอยู่กับขอบเขตของการออกแบบและพัฒนาโปรแกรมจัดการระบบเครือข่าย
3. ส่วนสนับสนุนการจัดการระบบเครือข่าย (Network Management Support Software) มีหน้าที่จัดการข้อมูลเกี่ยวกับอุปกรณ์ให้เป็นไปตามข้อกำหนดของมิบประเภทต่างๆ

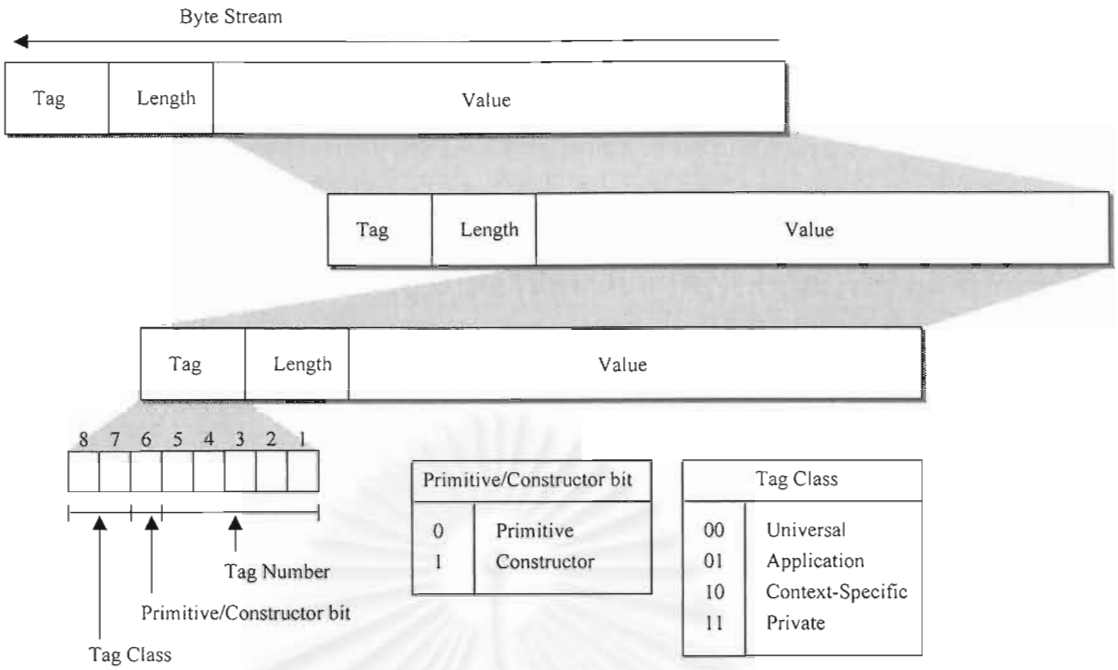
2.10 การเฝ้าดูแลระบบเครือข่าย (Network Manitoring)

การเฝ้าดูแลระบบเครือข่าย เป็นงานพื้นฐานของการจัดการระบบเครือข่าย เพื่อทำหน้าที่ติดตามสถานะของอุปกรณ์ที่เชื่อมต่อกันในระบบเครือข่าย รายละเอียดของการเฝ้าดูแลระบบเครือข่ายมีดังนี้

1. โพลลิ่ง (Polling) เป็นวิธีการรับส่งข้อมูลระหว่างตัวจัดการและเอเจ้น โดยตัวจัดการมีหน้าที่สั่งการให้เอเจ้นส่งข้อมูลเกี่ยวกับอุปกรณ์เพื่อนำไปประมวลผล จัดรูปแบบ และนำเสนอข้อมูลให้เป็นไปตามต้องการ
2. การรายงานเหตุการณ์ (Event Reporting) เป็นวิธีการรับส่งข้อมูลระหว่างตัวจัดการและเอเจ้นโดยเอเจ้นสามารถส่งข้อมูลเกี่ยวกับตัวอุปกรณ์ให้แก่ตัวจัดการโดยไม่ต้องรอรับคำสั่ง

2.11 กฎเบสิกเอ็นโค้ดดิ้งรูล (Basic Encoding Rule)

เนื่องจากข้อกำหนดต่างๆที่เกี่ยวข้องกับโพรโทคอลเอสเอ็นเอ็มพีจะอยู่ในรูปแบบของภาษาเอเอสเอ็น.1 (ASN.1 ย่อมาจาก Abstract Syntax Notation 1) ซึ่งเป็นภาษามาตรฐานที่ใช้ในการกำหนดโครงแบบ นิยามคุณสมบัติ ชนิดและค่าของออบเจกต์และเอสเอ็นเอ็มพีเมสเสจ เพื่อใช้ในการอ้างอิงเท่านั้น ในทางปฏิบัติอุปกรณ์ที่ทำงานภายใต้โพรโทคอลเอสเอ็นเอ็มพี จะรับส่งข้อมูลด้วยยูดีพีแพ็กเก็ตซึ่งเป็นกลุ่มข้อมูลในรูปของไบต์โค้ด (Byte Code) ดังนั้นในการพัฒนาโปรแกรมให้สามารถติดต่อกับอุปกรณ์ภายใต้โพรโทคอลเอสเอ็นเอ็มพี จำเป็นต้องใช้กฎเรียกว่าเบสิกเอ็นโค้ดดิ้งรูล (Basic Encoding Rule หรือ BER) เพื่อแปลรูปข้อมูลจากเอเอสเอ็น.1 ให้เป็นไบต์โค้ด [5] [6] ค่าเอเอสเอ็น.1 (ASN.1 Value) เมื่อถูกแทนค่าแล้วจะมีโครงสร้างเป็นกระแสข้อมูลไบต์ (Byte Stream) เรียกว่าทีแอลวีสตรัคเจอร์ (TLV Structure ย่อมาจาก Type-Length-Value Structure) ประกอบไปด้วย 3 ส่วนของเขตข้อมูล (Field) ดังแสดงในรูปที่ 2.12



ASN.1 Type	Tag Number
INTEGER	00010
OCTET STRING	00100
OBJECT IDENTIFIER	00110
NULL	00101
SEQUENCE, SEQUENCE OF	10000

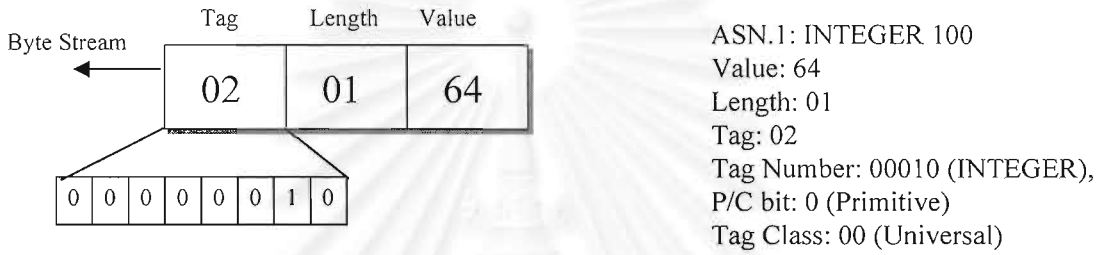
รูปที่ 2.12 แสดงรายละเอียดของ ทีแอลวีสตรัคเจอร์ (TLV Structure)

รายละเอียดของแต่ละเขตข้อมูล มีดังนี้

1. เขตข้อมูลสำหรับค่าเอเอสเอ็น.1 (Value) จากรูปที่ 2.12 เขตข้อมูลนี้อาจมีโครงสร้างแยกย่อยลงไปเป็น ทีแอลวีสตรัคเจอร์ ได้อีก (Constructor ASN.1 Type) หรือเป็นพียงโครงสร้างแบบง่าย (Simple ASN.1 Type) เช่น ทีแอลวีสตรัคเจอร์ ของ สายอักขระ (String) "public" มีค่าเอเอสเอ็น.1 ในเขตข้อมูลนี้ เป็น "70 75 62 6C 69 63"
2. เขตข้อมูลระบุขนาดของค่าเอเอสเอ็น.1 (Length) เช่น สายอักขระ "public" มีค่าในเขตข้อมูลนี้เป็น "06" เพื่อระบุว่ามีขนาด 6 ไบต์
3. เขตข้อมูลแท็ก (Tag) ใช้ 5 บิตแรกเรียกว่าหมายเลขแท็ก (Tag Number) สำหรับระบุชนิดของค่าเอเอสเอ็น.1 (ASN.1 Type) เช่น หมายเลขแท็กของสายอักขระคือ "00100" เลขจำนวนเต็มใช้หมายเลขแท็ก "00010" เป็นต้น บิตที่ 6 ของเขตข้อมูลแท็กเรียกว่า พริมิทีฟ/คอนสตรัคเตอร์บิต (Primitive/Constructor Bit) สำหรับใช้ระบุประเภทของ

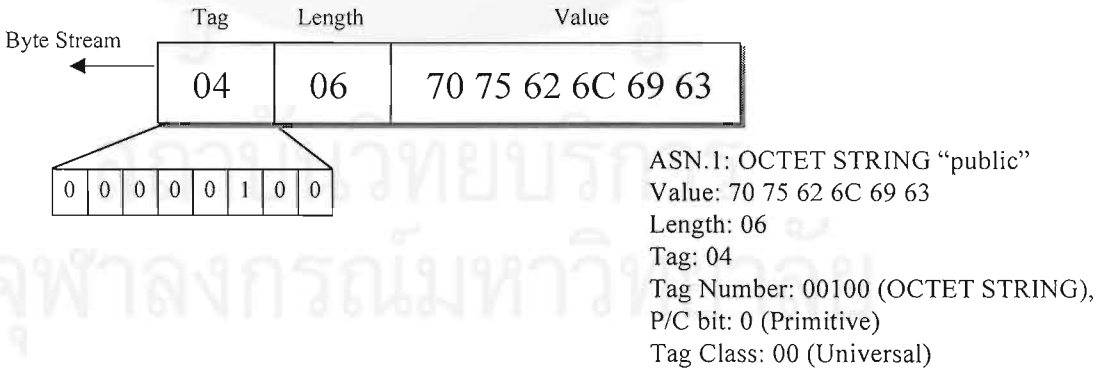
ทีแอลวีสตรัคเจอร์ เช่น เลขจำนวนเต็มและหมายเลขออบเจกต์จะมีทีแอลวีสตรัคเจอร์แบบง่ายเพราะค่าเอเอสเอ็น.1 เหล่านี้ไม่สามารถแยกย่อยลงไปเป็น ทีแอลวีสตรัคเจอร์อื่นได้อีก ส่วนบิตที่ 7 และบิตที่ 8 ของเขตข้อมูลแท็กเรียกว่าแท็กคลาส (Tag Class) ใช้ระบุประเภทของแท็กนัมเบอร์ โดยทั่วไปแล้วจะมีค่าเป็น "00"

รูปที่ 2.13 แสดงตัวอย่างการแปลงเลขจำนวนเต็ม INTEGER {100} จากเอเอสเอ็น.1 ให้อยู่ในรูปของไบต์ เริ่มจากแทนค่า 100 ด้วย 64 ซึ่งเป็นเลขฐาน 16 ขนาด 1 ไบต์ ถูกจัดอยู่ในประเภทพริมิทีฟ โดยหมายเลขแท็กมีค่าเป็น "00010" พริมิทีฟ/คอนสตรัคเตอร์บิตมีค่าเป็น "0" และแท็กคลาสมีค่าเป็น "00"



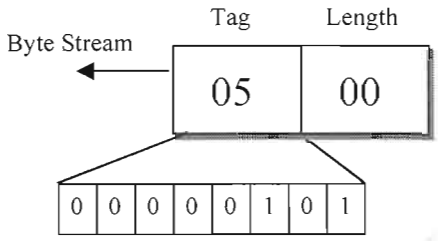
รูปที่ 2.13 แสดงการแปลงเลขจำนวนเต็ม 100 ให้อยู่ในรูปของไบต์

รูปที่ 2.14 แสดงการแปลงสายอักขระ OCTET STRING {"public"} จากเอเอสเอ็น.1 ให้อยู่ในรูปของไบต์ เริ่มจากแทนค่า "public" ด้วย "70 75 62 6C 69 63" ซึ่งเป็นเลขฐาน 16 ขนาด 6 ไบต์ ถูกจัดอยู่ในประเภทพริมิทีฟ โดยมีค่าหมายเลขแท็กเป็น "00100" พริมิทีฟ/คอนสตรัคเตอร์บิตมีค่าเป็น "0" และแท็กคลาสมีค่าเป็น "00"



รูปที่ 2.14 แสดงการแปลงสตริง "public" ให้อยู่ในรูปของไบต์

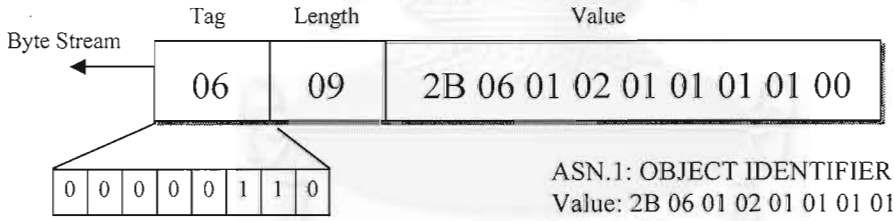
รูปที่ 2.15 แสดงตัวอย่างการแปลงค่าว่าง (NULL) จากเอเอสเอ็น.1 ให้อยู่ในรูปของไบต์ โดยถูกจัดอยู่ในประเภท พริมิทีฟ หมายเลขแท็กค่าเป็น "00101" พริมิทีฟ/คอนสตรัคเตอร์บิต มีค่าเป็น "0" และแท็กคลาสมีค่าเป็น "00"



ASN.1: NULL
Value: no field
Length: 00
Tag: 05
Tag Number: 00101 (NULL),
P/C bit: 0 (Primitive)
Tag Class: 00 (Universal)

รูปที่ 2.15 แสดงการแปลงค่าว่าง (NULL) ให้อยู่ในรูปของไบต์

รูปที่ 2.16 แสดงตัวอย่างการแปลงหมายเลขออบเจกต์ 1.3.6.1.2.1.1.1.1.0 จากเอเอสเอ็น.1 ให้อยู่ในรูปของไบต์ เริ่มจากแทนค่า 1.3.6.1.2.1.1.1.1.0 ด้วย "2B 06 01 02 01 01 01 01 00" ซึ่งเป็นเลขฐาน 16 ขนาด 9 ไบต์ ถูกจัดอยู่ในประเภทพริมิทีฟโดยมีค่าหมายเลขแท็กเป็น "00110" พริมิทีฟ/คอนสตรัคเตอร์บิต มีค่าเป็น "0" และแท็กคลาสมีค่าเป็น "00" โปรดสังเกต เลขจำนวนเต็ม 2 ตัวแรกของหมายเลขออบเจกต์ จะต้องแทนค่าด้วยสูตร 40 x oid1 + oid2 และค่าที่ได้คือ 43 หรือ 2B ของเลขฐาน16 [5]



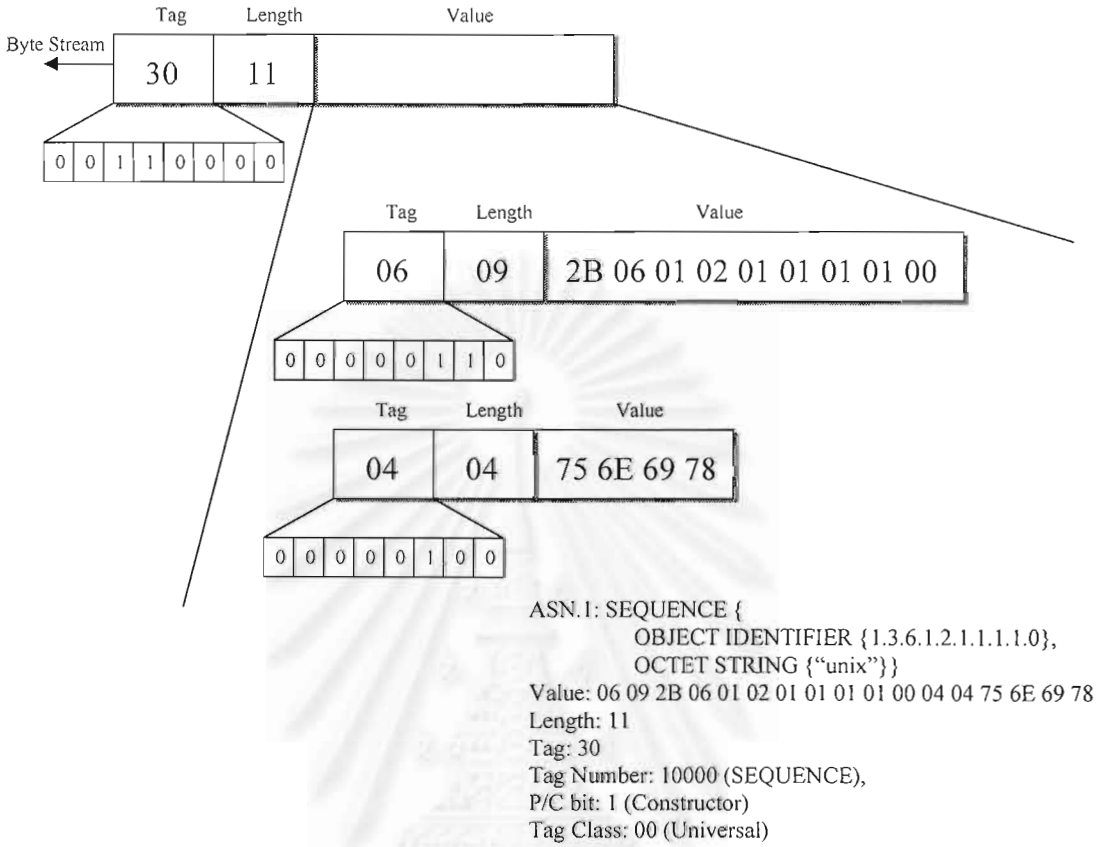
ASN.1: OBJECT IDENTIFIER 1.3.6.1.2.1.1.1.1.0
Value: 2B 06 01 02 01 01 01 01 00
Length: 09
Tag: 06
Tag Number: 00110 (OBJECT IDENTIFIER),
P/C bit: 0 (Primitive)
Tag Class: 00 (Universal)

รูปที่ 2.16 แสดงการแปลงหมายเลขออบเจกต์ 1.3.6.1.2.1.1.1.1.0 ให้อยู่ในรูปของไบต์

รูปที่ 2.17 แสดงตัวอย่างการแปลงโครงสร้างเอเอสเอ็น.1 แบบลำดับ (SEQUENCE) ให้อยู่ในรูปของไบต์ เราจะเห็นได้ว่าโครงสร้างแบบลำดับนี้ สามารถแยกย่อยลงไปเป็นที่แอลวีสตรัคเจอร์ได้อีก เริ่มจากแปลงหมายเลขออบเจกต์ 1.3.6.1.2.1.1.1.1.0 และสายอักขระ "unix" ให้เป็นไบต์โค้ด จากนั้นเราจะได้ขนาดของโครงสร้างแบบลำดับคือ 17 ไบต์ ถูกจัดเป็นประเภทคอนสตรัคเตอร์

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

โดยหมายเลขแท็กมีค่าเป็น “10000” พริมิทีฟ/คอนสตรัคเตอร์บิต มีค่าเป็น “1” และแท็กคลาสมีค่าเป็น “00”



รูปที่ 2.17 แสดงการแปลงโครงสร้างเอเอสเอ็น.1 แบบลำดับ (SEQUENCE) ให้อยู่ในรูปของไบต์

2.12 **อินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์** (Internet Information Server)

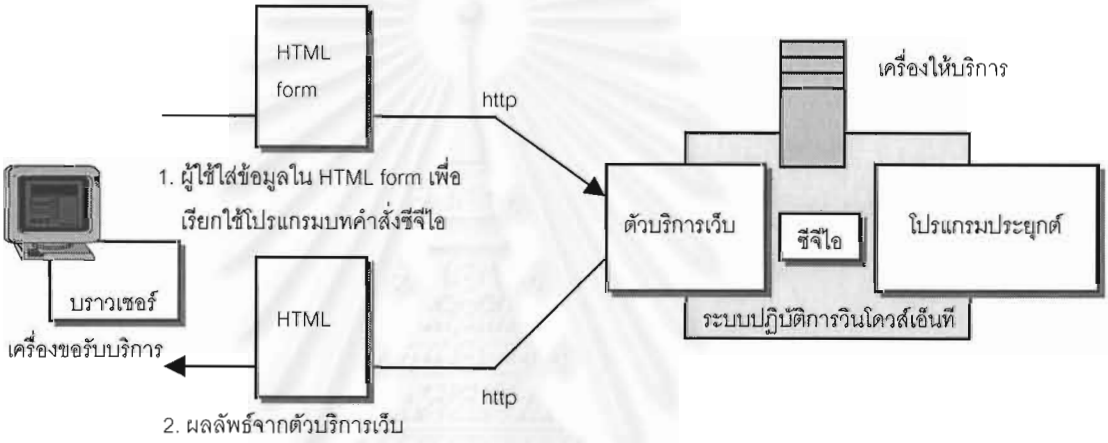
อินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์ คือ โปรแกรมประยุกต์บนระบบปฏิบัติการวินโดวส์เอ็นทีของไมโครซอฟต์ที่ได้รวมเอาเทคโนโลยีทางด้านการให้บริการข้อมูลบนเครือข่ายอินเทอร์เน็ตเพื่อทำหน้าที่เป็นตัวบริการเว็บ องค์ประกอบของอินเทอร์เน็ตอินฟอร์เมชันเซิร์ฟเวอร์ ที่ใช้ในงานวิจัยมีดังนี้

1. เอชทีทีพีเซิร์ฟเวอร์ (HTTP Server)

   เอชทีทีพี หรือ เกณฑ์วิธีขนส่งข้อความหลายมิติ คือเกณฑ์วิธีสำหรับตัวบริการเว็บใช้ในการกระจายข้อมูลข่าวสารบนบนเครือข่ายอินเทอร์เน็ต เอชทีทีพี มีลักษณะการทำงานแบบร้องขอและตอบรับ ข้อมูลที่เครื่องขอรับบริการได้รับจากเอชทีทีพีเซิร์ฟเวอร์จะอยู่ในรูปของแฟ้มเอชทีเอ็มแอลเรียกว่าเว็บเพจซึ่งสามารถแสดงผลผ่านโปรแกรมค้นผ่านเว็บเรียกว่าบราวเซอร์

2. ตัวต่อประสานร่วมสำหรับเกตเวย์ (Common Gateway Interface หรือ CGI)

ตัวต่อประสานร่วมสำหรับเกตเวย์ หรือ ซีจีไอ คือเกณฑ์วิธีสำหรับสื่อสาร 2 ทิศทาง ระหว่างเครื่องขอรับบริการและตัวบริการเว็บผ่านโปรแกรมตามมาตรฐานของซีจีไอเรียกว่า โปรแกรมบทคำสั่งซีจีไอ (CGI Program) เพื่อติดต่อกับระบบภายนอก เช่น โปรแกรมประยุกต์ ระบบฐานข้อมูล หรือ ควบคุมการทำงานขององค์ประกอบของเครื่องให้บริการที่กำลังปฏิบัติงานอยู่ ซีจีไอช่วยทำให้ให้เครื่องขอรับบริการและตัวบริการเว็บสามารถโต้ตอบกันได้ผ่านเว็บเพจที่อยู่ในรูปของฟอร์มเอชทีเอ็มแอล (HTML form) ซึ่งเป็นเว็บเพจที่ผู้ใช้สามารถใส่ข้อมูลที่ต้องการส่งให้ตัวบริการเว็บ ดังแสดงในรูปที่ 2.18 เราสามารถเขียนโปรแกรมซีจีไอจาก ภาษาเพิร์ล ภาษาซี วิช่วลเบสิก ฯลฯ



รูปที่ 2.18 แสดงการทำงานของซีจีไอผ่านเอชทีเอ็มแอลฟอร์ม

ภายในเอชทีเอ็มแอลฟอร์ม จะมีรหัสเอชทีเอ็มแอล (HTML MarkUp Code หรือ Tag) ชื่อ FORM เพื่อระบุการเริ่มต้นของส่วนที่เป็นเอชทีเอ็มแอลฟอร์มในเว็บเพจ ตารางที่ 2.2 แสดงรายละเอียดของรหัสเอชทีเอ็มแอล FORM

แอตตริบิวต์	คำอธิบาย
NAME	ใช้กำหนดชื่อของ เอชทีเอ็มแอลฟอร์ม
ACTION	ใช้ระบุชื่อของโปรแกรมซีจีไอที่ต้องการเรียกใช้
METHOD	ใช้กำหนดวิธีส่งข้อมูล ดังนี้ 1. GET คือวิธีส่งข้อมูลผ่านตัวแปรสภาพแวดล้อม ชื่อ QUERY_STRING 2. POST คือวิธีส่งข้อมูลผ่านหน่วยรับมาตรฐาน (STDIN)

ตารางที่ 2.3 แสดงรายละเอียดของรหัสเอชทีเอ็มแอล FORM

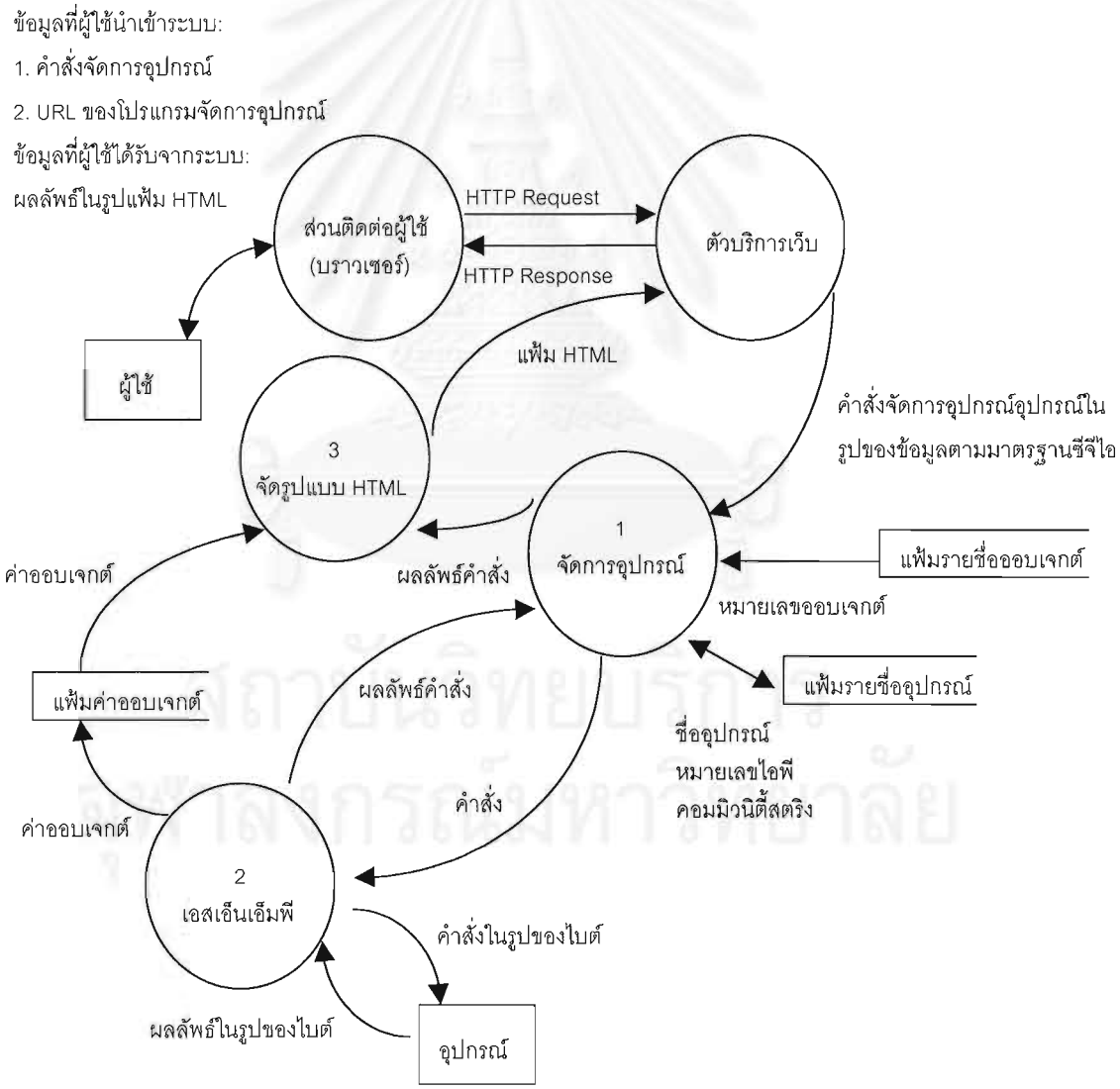
ตัวอย่าง

```
<FORM NAME="poll" ACTION="/cgi-bin/SnmpDevicePoll.cgi" METHOD="post">
```

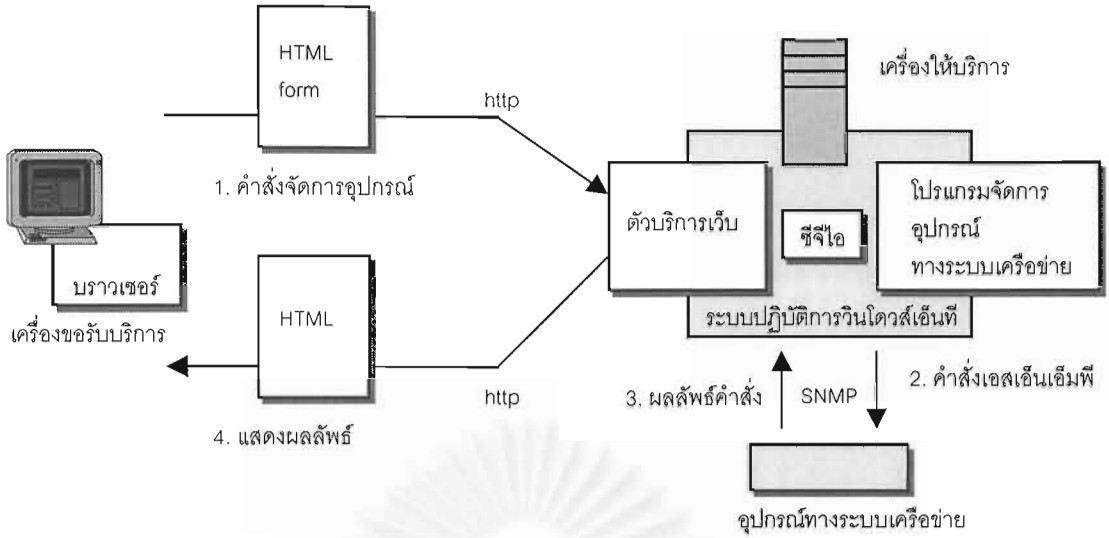
บทที่ 3
การวิเคราะห์และออกแบบโปรแกรม

แนวคิดในการออกแบบการทำงานของโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่ายผ่านบราวเซอร์บนระบบปฏิบัติการวินโดวส์เอ็นที สามารถแบ่งออกเป็น 4 ส่วนหลัก (รูปที่ 3.1) ดังนี้

1. การออกแบบส่วนจัดการอุปกรณ์
2. การออกแบบส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี
3. การออกแบบและจัดรูปแบบเว็บเพจ
4. การออกแบบโครงสร้างข้อมูล



รูปที่ 3.1 แสดงการทำงานโดยรวมของระบบ



รูปที่ 3.1 (ต่อ) แสดงการทำงานโดยรวมของระบบ

3.1 การออกแบบส่วนจัดการอุปกรณ์

เป็นการออกแบบโปรแกรมในส่วนทำหน้าที่จัดการอุปกรณ์ให้เป็นไปตามคำสั่งของผู้ใช้งาน โดยทำการรวบรวม และ ตรวจสอบองค์ประกอบต่างๆ ของคำสั่งจัดการอุปกรณ์จากข้อมูลในฟอร์มของเอชทีเอ็มแอล (HTML Form) ซึ่งผู้ใช้งานได้ส่งผ่านไปยังโปรแกรมบทคำสั่งซีจีไอที่ตัวบริการเว็บ หลังจากนั้นจะทำการจัดเตรียมคำสั่งจัดการอุปกรณ์ โดยใช้ข้อมูลเหล่านี้และเพิ่มเติมรายละเอียดที่จำเป็นต้องใช้ในการติดต่อกับอุปกรณ์ เช่น ชนิดของโอเปอเรชั่นเอสเอ็นเอ็มพี หมายเลขออบเจกต์ หมายเลขไอพี และ คอมมิวนิตี้สตริง เป็นต้น

3.2 การออกแบบส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี

ส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี ประกอบไปด้วย 2 ส่วน ตามรูปที่ 3.2 คือ

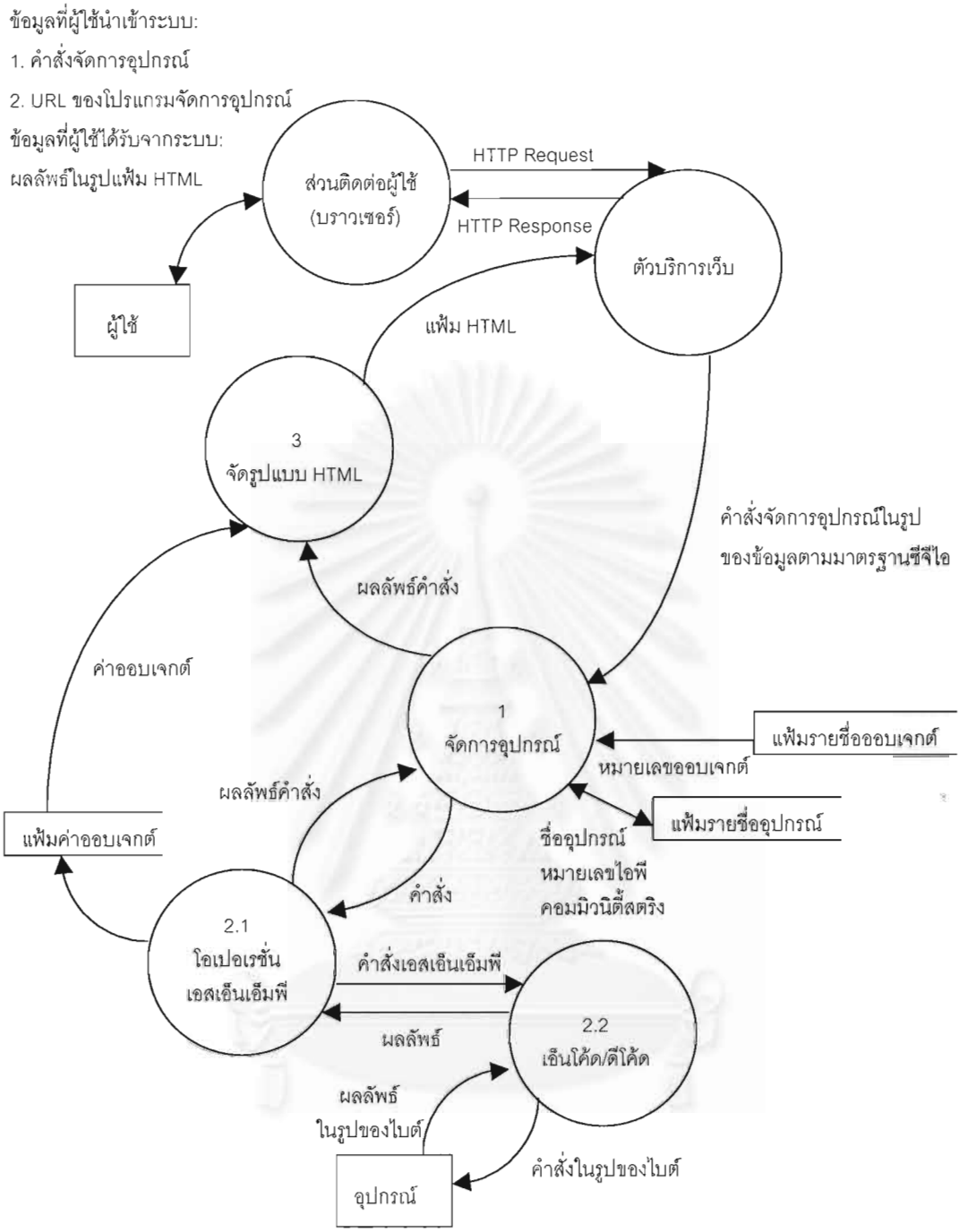
1. ส่วนจัดการโอเปอเรชั่นเอสเอ็นเอ็มพี
2. ส่วนจัดการเบสิกเอ็นโค้ดดิ้งรูล

3.2.1 ส่วนจัดการโอเปอเรชั่นเอสเอ็นเอ็มพี

เป็นการออกแบบโปรแกรมในส่วนทำหน้าที่ในรายละเอียดของ โอเปอเรชั่นเอสเอ็นเอ็มพี เพื่อติดต่อรับส่งข้อมูลกับอุปกรณ์ผ่าน เอสเอ็นเอ็มพีเมสเสจ ตามข้อกำหนดของ โพรโทคอล เอสเอ็นเอ็มพี

3.2.2 ส่วนจัดการเบสิกเอ็นโค้ดดิ้งรูล

เป็นส่วนทำหน้าที่เอ็นโค้ด (Encode) เพื่อแปลงเอสเอ็นเอ็มพีเมสเสจของ โอเปอเรชันเอสเอ็นเอ็มพีให้เป็นกลุ่มข้อมูลในรูปของไบต์โค้ดด้วยกฎเบสิกเอ็นโค้ดดิ้งรูล

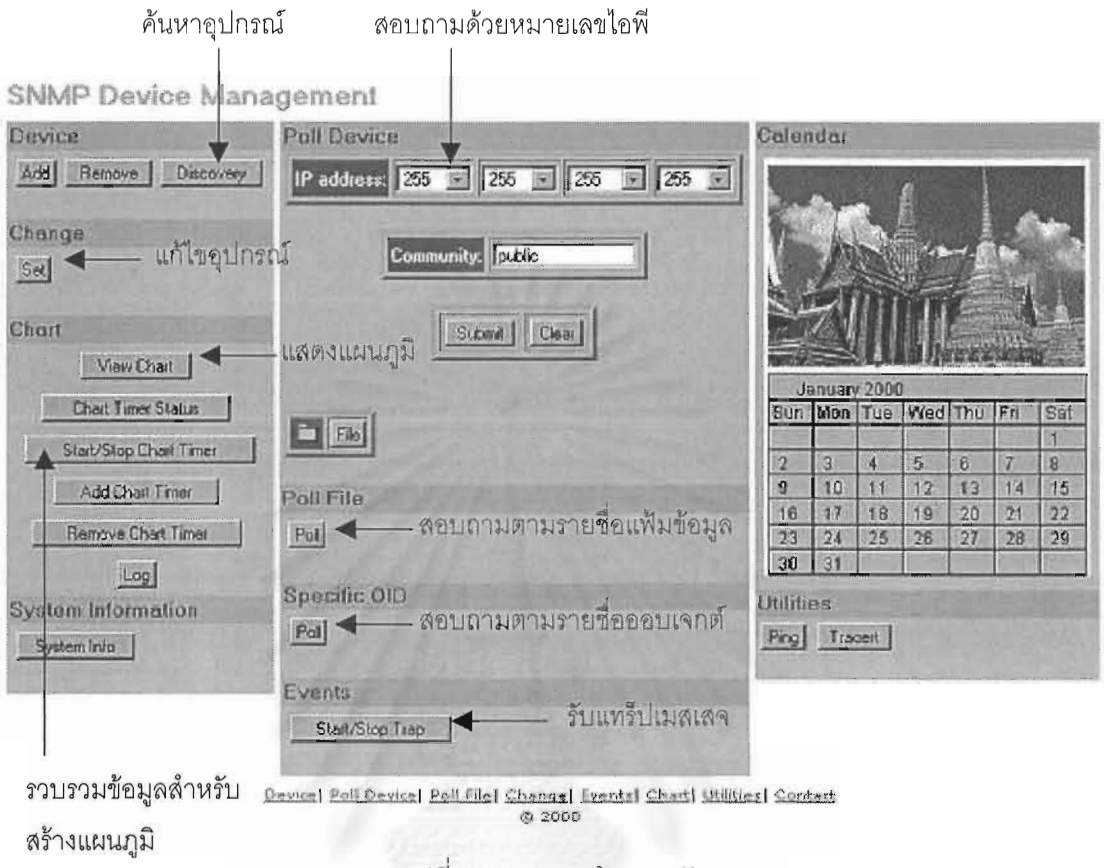


รูปที่ 3.2 แสดงการทำงานของส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี

3.3 การออกแบบและจัดรูปแบบเว็บเพจ

เป็นการออกแบบการแสดงผลผ่านหน้าจอเพื่อรับข้อมูล จัดรูปแบบ และ นำเสนอผลลัพธ์จากการประมวลผลของโปรแกรมในรูปของเว็บเพจให้เป็นไปตามประเภทของการเรียกใช้งานโดยมีหน้าจอหลัก (รูปที่ 3.3) ทำหน้าที่เป็นลำดับแรกของการติดต่อระหว่างเครื่องขอรับบริการและเครื่องให้บริการ จากรายการเลือกในหน้าจอหลักจะมีหน้าจออื่นๆประกอบไปด้วยเว็บเพจแบบคงที่

(Static Web Page) และ เว็บเพจแบบเปลี่ยนแปลงได้ (Dynamic Web Page) แยกย่อยตามประเภทของการเลือกรายการ

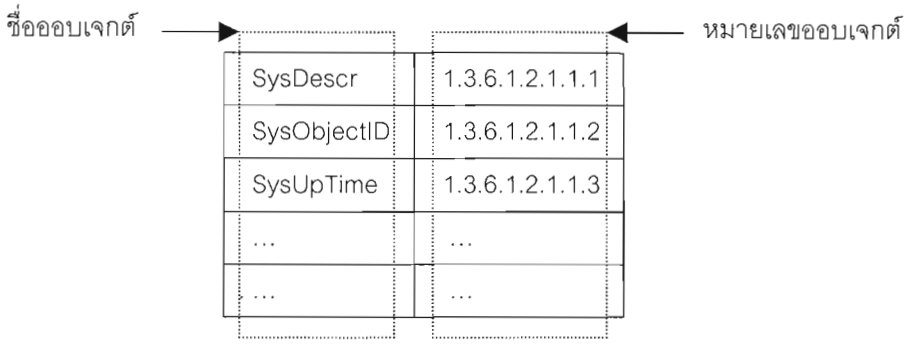


รูปที่ 3.3 แสดงหน้าจอหลัก

3.4 การออกแบบโครงสร้างข้อมูล

3.4.1 การออกแบบโครงสร้างข้อมูลของรายชื่อออบเจกต์

เนื่องจากหมายเลขออบเจกต์ภายใต้มิบเป็นชุดตัวเลขที่ยากต่อการจดจำและอ้างอิงโดยโปรแกรมและผู้ใช้งาน เช่น กรณีต้องการทราบระยะเวลาที่อุปกรณ์ได้ให้บริการ (System Up Time) ซึ่งเป็นค่าของออบเจกต์ในกลุ่มซิสเต็มภายใต้มิบ-ทู ชื่อ sysUpTime แต่ในการอ้างอิงถึงค่านี้จากอุปกรณ์ผ่านเอสเอ็นเอ็มพีเมสเสจ จำเป็นต้องใช้หมายเลขออบเจกต์ที่แท้จริง คือ 1.3.6.1.2.1.1.3.0 ดังนั้นเพื่อความสะดวกต่อการทำงานของ โปรแกรม และ ผู้ใช้งาน จึงมีแนวคิดรวบรวมหมายเลขออบเจกต์เหล่าไว้ด้วยกันในโครงสร้างข้อมูลแบบแฮช (Hash) และใช้ชื่อออบเจกต์เป็นเลขที่อยู่แบบแฮช (Hash Address) ของหมายเลขออบเจกต์ ดังแสดงในรูปที่ 3.4



h1(oidname) = oidnumber

โดย h1 คือ ฟังก์ชันแบบแฮช

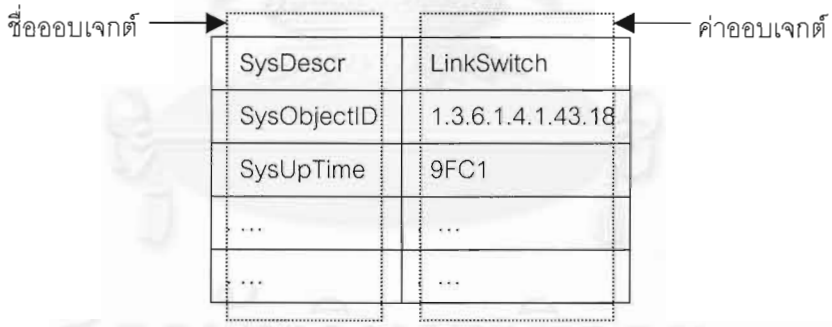
oidname คือชื่อออบเจกต์

oidnumber คือหมายเลขออบเจกต์

รูปที่ 3.4 แสดงโครงสร้างข้อมูลแบบแฮชสำหรับข้อมูลของรายชื่อออบเจกต์

3.4.2 การออกแบบโครงสร้างข้อมูลของค่าออบเจกต์

เพื่อให้สามารถจัดเก็บค่าออบเจกต์ที่รวบรวมได้จากอุปกรณ์ และนำมาอ้างอิงใช้งานได้อย่างสะดวก จึงมีแนวคิดใช้โครงสร้างข้อมูลแบบแฮชเพื่อเก็บค่าออบเจกต์โดยใช้ชื่อออบเจกต์เป็นเป็นเลขที่อยู่แบบแฮช ดังแสดงในรูปที่ 3.5



h2(oidname) = oidvalue

โดย h2 คือ ฟังก์ชันแบบแฮช

oidname คือชื่อออบเจกต์

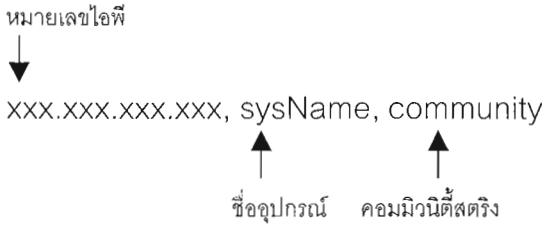
oidvalue คือค่าออบเจกต์

รูปที่ 3.5 แสดงโครงสร้างข้อมูลแบบแฮชสำหรับข้อมูลของค่าออบเจกต์

3.4.3 การออกแบบโครงสร้างแฟ้มข้อมูล

ขั้นตอนนี้คือการออกแบบแฟ้มข้อมูลสำหรับใช้จัดเก็บค่าออบเจกต์ของมิบในกลุ่มต่างๆที่รวบรวมได้จากอุปกรณ์ ประกอบไปด้วย

1. แฟ้มข้อมูลรายชื่ออุปกรณ์ ใช้จัดเก็บหมายเลขไอพี ชื่ออุปกรณ์ และคอมมิวนิตี้สตริง ภายในแฟ้ม จะเป็นข้อมูลที่จัดเก็บเป็นแบบตัวอักษร ประกอบไปด้วย หมายเลขไอพี ชื่ออุปกรณ์ซึ่งเป็นค่าของออบเจกต์ sysName และ คอมมิวนิตี้สตริง แต่ละเขตข้อมูลจะถูกคั่นด้วย ',' ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 แสดงรูปแบบของแฟ้มข้อมูลรายชื่ออุปกรณ์

2. แฟ้มข้อมูลทางสถิติของอุปกรณ์เพื่อใช้นำเสนอในรูปแบบของแผนภูมิ โดยการกำหนดให้โปรแกรมจัดการอุปกรณ์ ฯ ทำหน้าที่ติดต่อกับอุปกรณ์เพื่อรวบรวมข้อมูลให้เป็นไปตามระยะเวลาที่ได้กำหนดไว้ เช่น ทุก 1 นาที แฟ้มข้อมูลเหล่านี้แบ่งเป็น 13 ประเภท ดังนี้

    1) แฟ้มข้อมูลเก็บค่าออบเจกต์ ifInOctets และ ifOutOctets ซึ่งเป็นจำนวนไบต์ของข้อมูลที่ได้ผ่านเข้าออกในแต่ละอินเตอร์เฟสของอุปกรณ์
    2) แฟ้มข้อมูลเก็บค่าออบเจกต์ ifInUcastPkts ifOutUcastPkts ifInNUcastPkts และ ifOutNUcastPkts ซึ่งเป็นจำนวนแพ็กเก็ตของข้อมูลที่ได้ผ่านเข้าออกในแต่ละอินเตอร์เฟสของอุปกรณ์
    3) แฟ้มข้อมูลเก็บค่าออบเจกต์ ifInErrors ifOutErrors ifInDiscards ifOutDiscards และ ifInUnknownProtos ซึ่งเป็นจำนวนแพ็กเก็ตของข้อมูลที่ได้ผ่านเข้าออกในแต่ละอินเตอร์เฟสของอุปกรณ์แต่ได้เกิดข้อผิดพลาดขึ้น
    4) แฟ้มข้อมูลของปริมาณการใช้บริการ (Utilization) ของแต่ละอินเตอร์เฟสของอุปกรณ์ โดยคำนวณจาก

        utilization =

        100 * ((8 * (ifInOctets_diff + ifOutOctets_diff) / (t1 – t0)) / ifSpeed)

        โดยกำหนดให้

        ifInOctets_diff = ifInOctets(t1) – ifInOctets(t0)

        ifOutOctets_diff = ifOutOctets(t1) – ifOutOctets(t0)

        ifInOctets คือ จำนวนไบต์ของข้อมูลที่ได้ผ่านเข้ามาในอินเตอร์เฟส

ifOutOctets คือ จำนวนไบต์ของข้อมูลที่ได้ออกจากอินเตอร์เฟส

t0 และ t1 คือเวลา โดยที่ t1 > t0

ifSpeed คือความเร็วของอินเตอร์เฟส

สำหรับอินเตอร์เฟสแบบสื่อสารสองทางเต็มอัตรา (Full Duplex) ต้องคำนวณจาก

utilization =

100 * ((8 * max(ifInOctets_diff,ifOutOctets_diff) / (t1 – t0)) / ifSpeed)

5) แฟ้มข้อมูลของอัตราความผิดพลาดในการรับส่งข้อมูล (Error Percentages) ของแต่ละอินเตอร์เฟสของอุปกรณ์ ซึ่งได้คำนวณจาก

InError = (ifInErrors(t1) – ifInErrors(t0)) /
(ifInUcastPkts(t1) – ifInUcastPkts(t0))

OutError = (ifOutErrors(t1) – ifOutErrors(t0)) /
(ifOutUcastPkts(t1) – ifOutUcastPkts(t0))

โดยกำหนดให้

InEror	คือ อัตราควาผิดพลาดในการรับข้อมูล
OutError	คือ อัตราความผิดพลาดในการส่งข้อมูล
ifInErrors	คือจำนวนแพ็กเก็ตของข้อมูลที่ได้ผ่านเข้ามาในอินเตอร์เฟสแต่ได้เกิดข้อผิด พลาดขึ้น
ifOutErrors	คือจำนวนแพ็กเก็ตของข้อมูลที่ส่งออกจากอินเตอร์เฟสแต่ได้เกิดข้อผิดพลาดขึ้น
ifInUcastPkts	คือจำนวนแพ็กเก็ตของข้อมูลที่ได้ผ่านเข้ามาในอินเตอร์เฟส
ifOutUcastPkts	คือจำนวนแพ็กเก็ตของข้อมูลที่ได้ออกจากอินเตอร์เฟส

t0 และ t1 คือ เวลา โดยที่ t1 > t0

6) แฟ้มข้อมูลเก็บค่าออบเจกต์ ipInReceives และ ipOutRequests ซึ่งเป็นจำนวนของไอพีแพ็กเก็ตที่ได้ผ่านเข้าออกอุปกรณ์

7) แฟ้มข้อมูลเก็บค่าออบเจกต์ ipInReceives ipInDelivers ipForwDatagrams ipHdrErrors ipInAddrErrors ipInUnknownProtos และ ipInDiscards ซึ่งเป็นจำนวนของไอพีแพ็กเก็ตที่ได้ผ่านเข้ามาในอุปกรณ์ และ เกิดข้อผิดพลาดประเภทต่างๆขึ้น

8) แฟ้มข้อมูลเก็บค่าออบเจกต์ ipOutRequests ipOutDiscards และ ipOutNoRoutes ซึ่งเป็นจำนวนของไอพีแพ็กเก็ตที่ถูกส่งออกจากอุปกรณ์ และ

เกิดข้อผิดพลาดประเภทต่างๆขึ้น

9) แฟ้มข้อมูลเก็บค่าออบเจกต์ ciscolocIfInBitsSec และ ciscolocIfOutBitsSec ซึ่งเป็นเป็นอัตราการรับส่งข้อมูลในแต่ละอินเตอร์เฟสของอุปกรณ์ ข้อมูลเหล่านี้เป็นไพร์เวทมิบภายใต้กลุ่มอินเตอร์เฟสซึ่งจะรวบรวมได้จากอุปกรณ์ที่เป็นของซิสโกเท่านั้น

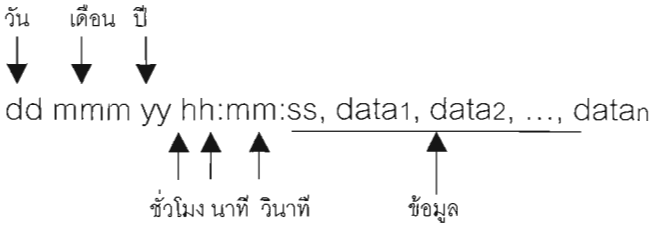
10) แฟ้มข้อมูลเก็บค่าออบเจกต์ ciscolocIfInPktsSec และ ciscolocIfOutPktsSec ซึ่งเป็นเป็นอัตราของแพ็กเก็ตของข้อมูลที่ได้ผ่านเข้าออกในแต่ละอินเตอร์เฟสของอุปกรณ์ ข้อมูลเหล่านี้เป็นไพร์เวทมิบภายใต้กลุ่มอินเตอร์เฟสซึ่งจะรวบรวมได้จากอุปกรณ์ที่เป็นของซิสโกเท่านั้น

11) แฟ้มข้อมูลเก็บค่าออบเจกต์ ciscolocIfipInPkts ciscolocIfipOutPkts ciscolocIfnovellInPkts ciscolocIfnovellOutPkts ciscolocIfdecnetInPkts และ ciscolocIfdecnetOutPkts ซึ่งเป็นจำนวนของข้อมูลในรูปแบบของแพ็กเก็ตของโพรโตคอลต่างๆ ที่ได้ผ่านเข้าออกในแต่ละอินเตอร์เฟสของอุปกรณ์ ข้อมูลเหล่านี้เป็นไพร์เวทมิบภายใต้กลุ่มอินเตอร์เฟสซึ่งจะรวบรวมได้จากอุปกรณ์ที่เป็นของซิสโกเท่านั้น

12) แฟ้มข้อมูลเก็บค่าออบเจกต์ ciscolocIfInRunts ciscolocIfInGiants ciscolocIfInCRC ciscolocIfInOverrun ciscolocIfInIgnored ciscolocIfInAbort และ ciscolocIfCollisions ซึ่งเป็นจำนวนแพ็กเก็ตของข้อมูลที่ได้ผ่านเข้าออกในแต่ละอินเตอร์เฟสของอุปกรณ์ แต่ได้เกิดข้อผิดพลาดขึ้น ข้อมูลเหล่านี้เป็นไพร์เวทมิบภายใต้กลุ่มอินเตอร์เฟสซึ่งจะรวบรวมได้จากอุปกรณ์ที่เป็นของซิสโกเท่านั้น

13) แฟ้มข้อมูลเก็บค่าออบเจกต์ภายใต้กลุ่มสถิติของอามอนมิบ ข้อมูลเหล่านี้จะรวบรวมได้จากอุปกรณ์ที่สามารถรองรับอามอนมิบเท่านั้น

ภายในแฟ้มข้อมูลเหล่านี้ จะเป็นข้อมูลที่จัดเก็บเป็นแบบตัวอักษร ประกอบไปด้วยเวลาที่รับข้อมูลจากอุปกรณ์ และค่าของออบเจกต์ที่รวบรวมได้ แต่ละเขตข้อมูลจะถูกคั่นด้วย ',' ดังรูปที่ 3.7



รูปที่ 3.7 แสดงรูปแบบของแฟ้มข้อมูลทางสถิติของอุปกรณ์

3. แฟ้มข้อมูลรายชื่อแผนภูมิใช้จัดเก็บหมายเลขไอพี ชื่ออุปกรณ์ และ คอมมิวนิตี้สตริง เพื่อใช้ในการรวบรวมข้อมูลสำหรับสร้างแผนภูมิ ภายในแฟ้ม จะเป็นข้อมูลที่จัดเก็บเป็นแบบตัวอักษร ประกอบไปด้วย หมายเลขไอพี ชื่ออุปกรณ์ซึ่งเป็นค่าของออบเจกต์ sysName และ คอมมิวนิตี้สตริง แต่ละเขตข้อมูลจะถูกคั่นด้วย ',' ดังแสดงในรูปที่ 3.8

รูปที่ 3.8 แสดงรูปแบบของแฟ้มข้อมูลรายชื่อแผนภูมิ

จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
การพัฒนาโปรแกรม

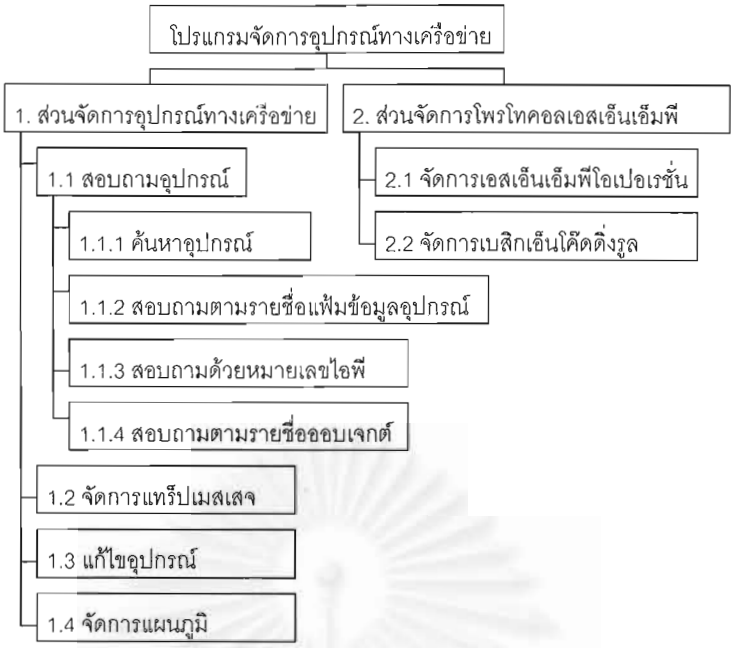
การพัฒนาโปรแกรมเป็นการพัฒนาบนเครื่องให้บริการ 1 เครื่อง ประกอบไปด้วยโปรแกรมประยุกต์ ไมโครซอฟต์อินเทอร์เน็ต อินฟอร์เมชั่น เซิร์ฟเวอร์ เวอร์ชั่น 4.0 ภายใต้ระบบปฏิบัติการวินโดวส์เอ็นที เวอร์ชั่น 4.0 และได้กำหนดให้เครื่องให้บริการได้ติดตั้งตัวแปลโปรแกรมภาษาเพิร์ล 5 ของแอคทีฟสเตท (ActiveState Perl 5) [11] เพื่อใช้ในการพัฒนาโปรแกรม ในส่วนของการใช้กฎเบสิกเอ็นโค้ดดิ้งรูลเพื่อทำหน้าที่เอ็นโค้ดเอสเอ็นเอ็มพีเมสเสจให้อยู่ในรูปของไบต์ของยูดีพีแพ็กเก็ต ได้เตรียมโมดูลแบบเพิร์ล BER.pm และ SNMP_Session.pm [12] และได้เตรียมโมดูลแบบเพิร์ล CGI.pm [13] ประกอบด้วยฟังก์ชันของชุดคำสั่งแบบซีจีไอสำหรับใช้ประกอบการพัฒนาโปรแกรมในส่วนที่ต้องอาศัยโค้ดสำหรับงานเกี่ยวกับซีจีไอ เช่น การใช้เอชทีเอ็มแอลฟอร์มเพื่อรับข้อมูลจากผู้ใช้งาน การสร้างเว็บเพจแบบเปลี่ยนแปลงได้ เป็นต้น

ผู้พัฒนาได้ใช้โปรแกรมประยุกต์ ไมโครซอฟต์ฟร้อนท์เพจ (Microsoft Front Page) และเพ้นท์ช้อพโพร (Paint Shop Pro) เป็นเครื่องมือช่วยในการสร้างเว็บเพจแบบคงที่ ส่วนเว็บเพจแบบเปลี่ยนแปลงได้จะถูกสร้างขึ้นโดยโปรแกรมที่ได้พัฒนาขึ้นตามประเภทของการเรียกใช้งาน สำหรับหน้าจอเว็บเพจที่แสดงผลลัพธ์ในรูปแบบของแผนภูมิ ทำได้โดยการใช้โปรแกรมประยุกต์จาวาชาท (JavaChart) ของ วิชวลเอ็นจิเนียริ่ง (Visual Engineering) [14]

จากการวิเคราะห์และออกแบบระบบ ฯ ในบทที่ 3 ทำให้สามารถแบ่งการพัฒนาโปรแกรมออกเป็น 2 ส่วนหลักดังรายละเอียดในรูปที่ 4.1 ดังนี้

1. การพัฒนาส่วนจัดการอุปกรณ์ทางเครือข่าย
2. การพัฒนาส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี

การแยกโปรแกรมจัดการอุปกรณ์ ฯ ออกเป็น 2 ส่วนนี้ทำให้สามารถรองรับการปรับปรุงเพิ่มเติมให้สามารถทำงานรองรับโพรโทคอลเอสเอ็นเอ็มพีเวอร์ชั่นใหม่ๆในอนาคตได้ตามต้องการ โดยการย้ายจากโมดูลแบบเพิร์ล BER.pm และ SNMP_Session.pm [12] ไปใช้โมดูลแบบเพิร์ลอื่นๆที่สามารถใช้กฎเบสิกเอ็นโค้ดดิ้งรูล ของโพรโทคอลเอสเอ็นเอ็มพี เวอร์ชั่นใหม่ได้ โดยไม่กระทบส่วนอื่นของโปรแกรม



รูปที่ 4.1 แสดงองค์ประกอบของโปรแกรมจัดการอุปกรณ์ทางเครือข่าย

4.1 ส่วนจัดการอุปกรณ์ทางเครือข่าย

วัตถุประสงค์เพื่อให้สามารถจัดการอุปกรณ์ โดยแบ่งการพัฒนาเป็น 4 ส่วนคือ

1. โปรแกรมสอบถามอุปกรณ์
2. โปรแกรมจัดการแทร็ปเมสเสจ
3. โปรแกรมแก้ไขอุปกรณ์
4. โปรแกรมจัดการแผนภูมิ

4.1.1 โปรแกรมสอบถามอุปกรณ์ (SnmpDevicePoll.cgi)

มีหน้าที่รวบรวมข้อมูลจากอุปกรณ์ โดยสามารถอธิบายตามประเภทของการใช้งานผ่านการเรียกใช้ฟังก์ชันดังต่อไปนี้

4.1.1.1 ฟังก์ชันค้นหาอุปกรณ์ (SnmpDeviceDiscovery)

มีหน้าที่ค้นหาอุปกรณ์ที่เชื่อมต่อกันในเครือข่าย โดยสามารถค้นพบอุปกรณ์ที่สามารถรองรับโพรโตคอลเอสเอ็นเอ็มพีได้เท่านั้น วิธีการค้นหาอุปกรณ์ทำได้โดยกำหนดค่าคอมมิวนิตี้สตริง ค่าเริ่มต้นและค่าสุดท้ายของหมายเลขไอพี และทำการส่งเอสเอ็นเอ็มพีเมสเสจออกไปยังเครือข่ายเพื่อให้อุปกรณ์ที่มีค่าคอมมิวนิตี้สตริงและหมายเลขไอพีเหล่านี้ตอบกลับมาด้วยเอสเอ็นเอ็มพีเมสเสจเพื่อบอกชื่อ ชนิด และสถานที่ตั้งของอุปกรณ์ ฟังก์ชันค้นหาอุปกรณ์ มีรูปแบบการเรียกใช้ดังนี้

```
&SnmpDeviceDiscovery($ip1,$ip2,$ip3,$ip4,$scan,$range,$community);
```

$ip1 คือ ส่วนแรก (8 บิตแรก) ของหมายเลขไอพี

$ip2 คือ ส่วนที่สองของหมายเลขไอพี

$ip3 คือ ส่วนที่สามของหมายเลขไอพี

$ip4 คือ ส่วนที่สี่ (8 บิตสุดท้าย) ของหมายเลขไอพี

$scan กำหนดวิธีการใช้หมายเลขไอพีในการค้นหาอุปกรณ์

$range กำหนดจำนวนหมายเลขไอพี ที่ต้องการใช้ในการค้นหาอุปกรณ์

$community คือ ค่าคอมมิวนิตี้สตริง

ขั้นตอนการทำงานของฟังก์ชันค้นหาอุปกรณ์

1. ตั้งค่า $poll = "discovery"
2. if $scan == "4"
   $ip = $ip4
   else if $scan == "3"
   $ip = $ip3
   else if $scan == "2"
   $ip = $ip2
   else $ip = $ip2
3. while $ip <= 255 and $range > 0 ทำข้อ 3.1 ถึงข้อ 3.5
   3.1. if $scan == "4"
   $poll_ip = "$ip1.$ip2.$ip3.$ip"
   else if $scan == "3"
   $poll_ip = "$ip1.$ip2.$ip.$ip4"
   else if $scan == "2"
   $poll_ip = "$ip1.$ip.$ip3.$ip4"
   else $poll_ip = "$ip.$ip2.$ip3.$ip4"
   3.2. ส่งค่าในตัวแปร $poll_ip $community และ $poll ไปประมวลผลที่ฟังก์ชัน SnmpDevicePoll ของโปรแกรมจัดการโอเปอเรชั่นเอสเอ็นเอ็มพี ผลลัพธ์คือค่าออบเจกต์ในตัวแปร $oidvaluescalar ของโครงสร้างข้อมูลแบบแฮช โดยใช้ชื่อออบเจกต์ในตัวแปร $oidname เป็นเลขที่อยู่แบบแฮช ดังแสดงในตารางที่ 4.1

3.3. แสดงผลออกเว็บเพจ พร้อมกับแทรกฟอร์มของเอชทีเอ็มแอล เพื่อให้ผู้ใช้งานสามารถเรียกแสดงข้อมูลเพิ่มเติมเกี่ยวกับอุปกรณ์ ผ่านฟังก์ชันดังแสดงในตารางที่ 4.1

3.4. $ip = $ip + 1

3.5. $range = $range – 1

ตัวแปร และฟังก์ชัน	คำอธิบาย
$oidvaluescalar{'sysObjectID'}	ค่าออบเจกต์ sysObjectID
$oidvaluescalar{'sysName'}	ค่าออบเจกต์ sysName
$oidvaluescalar{'sysDescr'}	ค่าออบเจกต์ sysDescr
$oidvaluescalar{'sysLocation'}	ค่าออบเจกต์ sysLocation
SnmpDevicePollSystem	ฟังก์ชันทำหน้าที่รวบรวมออบเจกต์ของกลุ่ม ซิสเต็ม
SnmpDevicePollInterfaces	ฟังก์ชันทำหน้าที่รวบรวมออบเจกต์ของกลุ่ม อินเตอร์เฟส
SnmpDevicePollIP	ฟังก์ชันทำหน้าที่รวบรวมออบเจกต์ของกลุ่ม ไอพี
SnmpDevicePollTCP	ฟังก์ชันทำหน้าที่รวบรวมออบเจกต์ของกลุ่ม ทีซีพี
SnmpDevicePollUDP	ฟังก์ชันทำหน้าที่รวบรวมออบเจกต์ของกลุ่ม ยูดีพี
SnmpDevicePollCisco	ฟังก์ชันทำหน้าที่รวบรวมออบเจกต์ของกลุ่มโพรดักส์ (Products) กลุ่มแชสซี (Chassis) และกลุ่มอินเตอร์เฟส (Interfaces) ของไพร์เวทมิบของซิสโก้
SnmpDevicePollBay	ฟังก์ชันทำหน้าที่รวบรวมออบเจกต์ของกลุ่มฮาร์ดแวร์ (Hardware) ของไพร์เวทมิบของเบย์เน็ตเวิร์ก
SnmpDevicePoll3Com	ฟังก์ชันทำหน้าที่รวบรวมออบเจกต์ของกลุ่มลิ้งค์สวิช (LinkSwitch) ของไพร์เวทมิบของทรีคอม (3Com)

ตารางที่ 4.1 แสดงตัวแปรและฟังก์ชัน เรียกใช้โดยฟังก์ชัน SnmpDeviceDiscovery

4.1.1.2 ฟังก์ชันสอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์ (SnmpDevicePollFile)

วัตถุประสงค์เพื่อให้สามารถตรวจสอบสถานะของอุปกรณ์ตามรายชื่อในแฟ้มข้อมูลรายชื่ออุปกรณ์ ฟังก์ชันสอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์ มีรูปแบบการเรียกใช้ดังนี้

```
&SnmpDevicPollFile($poll_file);
```

$poll_file คือ ชื่อของแฟ้มข้อมูลสำหรับเก็บรายชื่อของอุปกรณ์ที่ต้องการรวบรวมข้อมูล

ขั้นตอนการทำงานของฟังก์ชันสอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์

1. ตั้งค่า $poll = “file”
2. เปิดแฟ้มข้อมูลที่มีชื่อตามค่าในตัวแปร $poll_file
3. while not EOF ทำข้อ 3.1 ถึงข้อ 3.3
    3.1. อ่านแฟ้มข้อมูลเข้ามาทีละระเบียนแต่ละระเบียนประกอบด้วยชื่ออุปกรณ์ หมายเลขไอพี และ คอมมิวนิตี้สตริง สำหรับตัวแปร $sysName $poll_ip และ $community ตามลำดับ
    3.2. ส่งค่าในตัวแปร $poll_ip $community และ $poll ไปประมวลผลที่ฟังก์ชัน SnmpDevicePoll ผลลัพธ์ คือ ค่าในตัวแปรต่างๆดังแสดงในตารางที่ 4.1
    3.3. แสดงผลออกเว็บเพจ พร้อมกับแทรกฟอร์มของเอชทีเอ็มแอล เพื่อให้ผู้ใช้งานสามารถเรียกแสดงข้อมูลเพิ่มเติมเกี่ยวกับอุปกรณ์ ผ่านฟังก์ชันต่างๆ ดังแสดงในตารางที่ 4.1
4. ปิดแฟ้มข้อมูลที่มีชื่อตามค่าของตัวแปร $poll_file

4.1.1.3 ฟังก์ชันสอบถามด้วยหมายเลขไอพี (SnmpDevicePollAddress)

วัตถุประสงค์เพื่อให้ผู้ใช้สามารถตรวจสอบสถานะของอุปกรณ์ด้วยการระบุหมายเลขไอพีของอุปกรณ์ได้ตามต้องการ ฟังก์ชันนี้มีขั้นตอนการทำงานเหมือนฟังก์ชันสอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์ (SnmpDevicePollFile) แต่หมายเลขไอพีและคอมมิวนิตี้สตริงที่ส่งไปประมวลผลที่ฟังก์ชัน SnmpDevicePoll จะเป็นข้อมูลที่ได้จากฟอร์มของเอชทีเอ็มแอลซึ่งผู้ใช้งานได้ส่งผ่านไปยังโปรแกรม SnmpDevicePoll.cgi

4.1.1.4 ฟังก์ชันสอบถามตามรายชื่อออบเจกต์ (SnmpDevicePollSpecificOID)

มีหน้าที่ติดต่อกับอุปกรณ์เพื่ออ่านค่าออบเจกต์ โดยสามารถอ่านครั้งละ 1 ออบเจกต์ วัตถุประสงค์เพื่อให้ผู้ใช้สามารถระบุชื่อออบเจกต์ได้ตามต้องการ ฟังก์ชันนี้สามารถอ่านค่าออบเจกต์ของมิบ ดังต่อไปนี้ได้ทั้งหมด

1. มิบ-ทู
2. อามอนมิบ
3. ไพร์เวทมิบ
    - กลุ่มโพรดักส์ กลุ่มแชสซี และกลุ่มอินเตอร์เฟสของซิสโก้
    - กลุ่มฮาร์ดแวร์ของของเบย์เน็ตเวิร์ก
    - กลุ่มลิ้งค์สวิช ของทรีคอม

4.1.2 โปรแกรมจัดการแท็ปเมสเสจ (SnmpDeviceTrapListen.cgi)

มีหน้าที่รับเอสเอ็นเอ็มพีเมสเสจประเภทแทร็ป (SNMP Trap) จากอุปกรณ์ การส่งแทร็ป คือ วิธีการที่อุปกรณ์ส่งข้อมูลแจ้งตัวจัดการเพื่อรายงานสถานะเกี่ยวกับอุปกรณ์เมื่อค่าของออบเจกต์ได้เปลี่ยนไป วัตถุประสงค์ของโปรแกรมนี้คือทำการแยกเขตข้อมูลต่างๆในแทร็ปเมสเสจเพื่อนำเสนอในรูปแบบที่ผู้ใช้เข้าใจ

ขั้นตอนการทำงานของโปรแกรมจัดการแท็ปเมสเสจจะมีการเรียกใช้ฟังก์ชัน receive_trap เพื่อรับแทร็ปเมสเสจ และ ฟังก์ชัน decode_trap_request เพื่อดีโค้ด (Decode) แทร็ปเมสเสจ โดยฟังก์ชันทั้งสองนี้เป็นฟังก์ชัน ของโมดูลแบบเพิร์ล SNMP_Session.pm [10]

ขั้นตอนการทำงานของโปรแกรมจัดการแท็ปเมสเสจ

1. เปิดใช้หมายเลขช่องทางยูดีพีเบอร์ 162
2. while 1
3. $trap = receive_trap()
4. if $trap ทำข้อ 4.1 ถึง ข้อ 4.2
   - 4.1. ส่งค่าในตัวแปร $trap ไปประมวลผลที่ฟังก์ชัน decode_trap_request เพื่อแปลงจากไบต์โค้ดเป็นข้อมูลที่แสดงผลได้ ผลลัพธ์คือค่าต่างๆในตัวแปรดังแสดงในตารางที่ 4.2 โดยค่าเหล่านี้คือเขตข้อมูลต่างๆของแทร็ปเมสเสจตามข้อกำหนดของโพรโทคอลเอสเอ็นเอ็มพี
   - 4.2. แสดงผลออกทางเว็บเพจ

ตัวแปร	คำอธิบาย
$community	ค่าคอมมิวนิตี้สตริง
$enterprise	ค่าออบเจกต์ sysObjectID เพื่อระบุชนิดของอุปกรณ์
$agentaddr	หมายเลขไอพีของอุปกรณ์ที่เป็นตัวส่งแทร็ปเมสเสจ
$generictrap	ใช้ระบุชนิดของแทร็ปเมสเสจ
$specifictrap	ข้อมูลจำเพาะของแทร็ปเมสเสจ
$timestamp	ระยะเวลาตั้งแต่อุปกรณ์ได้เริ่มทำงาน
$VarBindList	ออบเจกต์ที่ค่าได้ถูกเปลี่ยนไป

ตารางที่ 4.2 แสดงตัวแปร เรียกใช้โดยโปรแกรม SnmpDeviceTrapListen.cgi

4.1.3 โปรแกรมแก้ไขอุปกรณ์ (SnmpDeviceSet.cgi)

มีหน้าที่รวบรวมและตรวจสอบองค์ประกอบต่างๆของคำสั่งแก้ไขอุปกรณ์จากข้อมูลใน

ฟอร์มของเอชทีเอ็มแอลที่ผู้ใช้งานได้ส่งมาที่โปรแกรมบทคำสั่งซีจีไอที่ตัวบริการเว็บ หลังจากนั้นจะทำการจัดเตรียมคำสั่งเพื่อแก้ไขอุปกรณ์โดยใช้ข้อมูลเหล่านี้และเพิ่มเติมรายละเอียดให้เป็นไปตามชนิดของค่าออบเจกต์ที่ต้องการแก้ไข โดยโปรแกรมสามารถแก้ไขออบเจกต์ภายใต้มิบ-ทู เท่านั้น และออบเจกต์ต้องมีคุณสมบัติที่สามารถแก้ไขได้ (READ WRITE) ตามข้อกำหนดของมิบ-ทู [5] [9] ดังแสดงในตารางที่ 4.3

จากตารางที่ 4.3 ออบเจกต์ ipRouteDest เป็นดรรชนีสำหรับอ้างอิงถึงค่าออบเจกต์ ipRouteIfIndex ipRouteNextHop ipRouteType ipRouteAge และ ipRouteMask เช่น ในกรณีออบเจกต์ ipRouteDest มีค่าเป็น 203.144.253.203 ค่าของออบเจกต์ ipRouteType จะต้องอ้างอิงด้วยค่าดังนี้

1.3.6.1.2.1.4.21.1.8.203.144.253.203

โดยส่วนแรกจะเป็นหมายเลขออบเจกต์ของ ipRouteType (1.3.6.1.2.1.4.21.1.8) และ ส่วนที่ 2 จะเป็นค่าของออบเจกต์ ipRouteDest (203.144.253.203)

สำหรับออบเจกต์ tcpConnState จะต้องอ้างอิงด้วยดรรชนีถึง 4 ตัว ดังนี้

1.3.6.1.2.1.6.13.1.1.203.144.253.203.137.0.0.0.0.10491

โดยส่วนแรกเป็นหมายเลขออบเจกต์ของ tcpConnState (1.3.6.1.2.1.6.13.1.1) ส่วนที่ 2 เป็นค่าของออบเจกต์ tcpConnLocalAddress (203.144.253.203) ส่วนที่ 3 คือค่าของออบเจกต์ tcpConnLocalPort (137) ส่วนที่ 4 เป็นค่าของออบเจกต์ tcpConnRemAddress (0.0.0.0) และส่วนที่ 5 คือค่าของออบเจกต์ tcpConnRemPort (10491)

ชื่อออบเจกต์	ประเภท	กลุ่ม	ดรรชนี
SysName	ค่าเดี่ยว	ซิสเต็ม	-
SysLocation	ค่าเดี่ยว	ซิสเต็ม	-
SysContact	ค่าเดี่ยว	ซิสเต็ม	-
IfAdminStatus	ค่าเดี่ยว	อินเตอร์เฟส	ออบเจกต์ ifIndex
IpForwarding	ค่าเดี่ยว	ไอพี	-
IpDefaultTTL	ค่าเดี่ยว	ไอพี	-
IpRouteDest	ตาราง	ไอพี	ออบเจกต์ ipRouteDest
IpRouteIfIndex	ตาราง	ไอพี	ออบเจกต์ ipRouteDest
IpRouteNextHop	ตาราง	ไอพี	ออบเจกต์ ipRouteDest
IpRouteType	ตาราง	ไอพี	ออบเจกต์ ipRouteDest

ตารางที่ 4.3 แสดงออบเจกต์ภายใต้มิบ-ทู ที่สามารถแก้ไขค่าได้

ชื่อออบเจกต์	ประเภท	กลุ่ม	ดรรชนี
IpRouteAge	ตาราง	ไอพี	ออบเจกต์ ipRouteDest
IpRouteMask	ตาราง	ไอพี	ออบเจกต์ ipRouteDest
IpRouteMetric1	ตาราง	ไอพี	ออบเจกต์ ipRouteDest
TcpConnState	ตาราง	ทีซีพี	ออบเจกต์ tcpConnLocalAddress ออบเจกต์ tcpConnLocalPort ออบเจกต์ tcpConnRemAddress ออบเจกต์ tcpConnRemPort

ตารางที่ 4.3 (ต่อ) แสดงออบเจกต์ภายใต้มิบ-ทู ที่สามารถแก้ไขค่าได้

ขั้นตอนการทำงานของโปรแกรมแก้ไขอุปกรณ์

1. รวบรวมข้อมูลในฟอร์มของเอชทีเอ็มแอลซึ่งได้รับจากผู้ใช้งานผ่านตัวบริการเว็บ เพื่อตั้งค่าตัวแปรดังนี้

   $poll_ip คือ หมายเลขไอพี ของอุปกรณ์ที่ต้องการแก้ไข

   $community คือ ค่าคอมมิวนิตี้สตริง

   $oid คือ สายอักขระสำหรับใช้ในการแก้ไขค่าออบเจกต์

   $oidvaluenew คือ ค่าใหม่ของออบเจกต์

   $action ใช้ระบุชนิดของออบเจกต์

2. if $action == "ip" ทำข้อ 2.1 ถึงข้อ 2.2

   2.1. แยกค่าสายอักขระในตัวแปร $oid ผลลัพธ์ คือ ตัวแปร $oidname จะเก็บชื่อของออบเจกต์ที่ต้องการแก้ไข และ ตัวแปร $ipRouteDest จะเก็บค่าออบเจกต์ ipRouteDest ซึ่งทำหน้าที่เป็นดรรชนีเพื่ออ้างอิงถึงตำแหน่งของค่าออบเจกต์ที่ต้องการแก้ไข ในกรณี ไม่สามารถแยกค่าสายอักขระในตัวแปร $oid ให้ไปทำข้อ 2.7 ถึงข้อ 2.8

   2.2. ตั้งค่าสายอักขระ $oidnum =

   "$SnmpMIB::OID{$oidname}.$ipRouteDest"

   โดย $SnmpMIB::OID{$oidname} เป็นการหาค่าหมายเลขออบเจกต์จากโครงสร้างข้อมูลแบบแฮช %SnmpMIB::OID

   else if $action == "tcp" ทำข้อ 2.3 ถึงข้อ 2.4

   2.3. แยกค่าสายอักขระในตัวแปร $oid ผลลัพธ์ คือ

   - ตัวแปร $oidname มีค่าเป็น "tcpConnState"

- ตัวแปร $tcpConnLocalAddress เก็บค่าออบเจกต์ tcpConnLocalAddress
- ตัวแปร $tcpConnLocalPort เก็บค่าออบเจกต์ tcpConnLocalPort
- ตัวแปร $tcpConnRemAddress เก็บค่าออบเจกต์ tcpConnRemAddress
- ตัวแปร tcpConnRemPort เก็บค่าออบเจกต์ tcpConnRemPort

2.4. ตั้งค่าสายอักขระ $oidnum = "$SnmpMIB::OID{$oidname}. $tcpConnLocalAddress.$tcpConnLocalPort. $tcpConnRemAddress.$tcpConnRemPort"

else if $action == "interfaces" ทำข้อ 2.5 ถึงข้อ 2.6

2.5. แยกค่าสายอักขระในตัวแปร $oid ผลลัพธ์ คือ ตัวแปร $oidname มีค่าเป็น "ifAdminStatus" และ ตัวแปร $ifIndex จะเก็บค่าออบเจกต์ ifIndex ซึ่งทำหน้าที่เป็นดรรชนีเพื่ออ้างอิงถึงตำแหน่งของค่าออบเจกต์ ifAdminStatus ซึ่งต้องการแก้ไข

2.6. ตั้งค่าสายอักขระ $oidnum = "$SnmpMIB::OID{$oidname}.$ifIndex"

else ทำข้อ 2.7 ถึงข้อ 2.8

2.7. ตั้งค่า $oidname = $oid

2.8. ตั้งค่าสายอักขระ $oidnum = "$SnmpMIB::OID{$oidname}"

3. ส่งค่าในตัวแปร $poll_ip $community $oidname $oidnum และ $oidvaluenew ไปประมวลผลที่ฟังก์ชัน SnmpDeviceSetObject ผลลัพธ์คือการแก้ไขค่าในออบเจกต์ ที่มีชื่อตามค่าในตัวแปร $oidname โดยมีค่าใหม่ตามค่าในตัวแปร $oidvaluenew

ตัวอย่างที่ 1

ข้อมูลในฟอร์มของเอชทีเอ็มแอลที่ได้รับจากผู้ใช้งานผ่านตัวบริการเว็บ เพื่อแก้ไขออบเจกต์ ifAdminStatus ของกลุ่มอินเตอร์เฟส ด้วยค่าใหม่ 2

```
<select name="ifAdminStatus_3">
        <option value="1">UP(1)
        <option value="2">DOWN(2)
        <option value="3">TESTING MODE(3)
</select>
```

```
<input type="radio" name="set" value="ifAdminStatus_3">
<input type="hidden" name="poll_ip" value="200.200.1.55">
<input type="hidden" name="community" value="nynexnet">
<input type="hidden" name="action" value="interfaces">
```

ผลลัพธ์

```
$poll_ip = "200.200.1.55"
$community = "nynexnet"
$action = "interfaces"
$oid = "ifAdminStatus_3"
$oidname = "ifAdminStatus"
$ifIndex = "3"
$oidnum = "1.3.6.1.2.1.2.2.1.7.3"
$oidvaluenew = "2"
```

ตัวอย่างที่ 2

ข้อมูลในฟอร์มของเอชทีเอ็มแอลที่ได้รับจากผู้ใช้งานผ่านตัวบริการเว็บ เพื่อแก้ไขออบเจกต์ ipRouteAge ของกลุ่มไอพี ด้วยค่าใหม่ 0

```
<input type="radio" name="set" value="ipRouteAge_2_203.144.253.203">
<input type="text" name="ipRouteAge_2_203.144.253.203" value="0">
<input type="hidden" name="poll_ip" value="200.200.1.55">
<input type="hidden" name="community" value="nynexnet">
<input type="hidden" name="action" value="ip">
```

ผลลัพธ์

```
$poll_ip = "200.200.1.55"
$community = "nynexnet"
$action = "ip"
$oid = "ipRouteAge_2_203.144.253.203"
$oidname = "ipRouteAge"
$ipRouteDest = "203.144.253.203"
$oidnum = "1.3.6.1.2.1.4.21.1.1.203.144.253.203"
```

$oidvaluenew = “0”

ตัวอย่างที่ 3

ข้อมูลในฟอร์มของเอชทีเอ็มแอลที่ได้รับจากผู้ใช้งานผ่านตัวบริการเว็บ เพื่อแก้ไขออบเจกต์ ipDefaultTTL ของกลุ่มไอพี ด้วยค่าใหม่ 128

```
<input type=”radio” name=”set” value=”ipDefaultTTL”>
<input type=”text” name=”ipDefaultTTL” value=”128”>
<input type="hidden" name="poll_ip" value="200.200.1.55">
<input type="hidden" name="community" value="nynexnet">
<input type="hidden" name="action" value="ip">
```

ผลลัพธ์

$poll_ip = “200.200.1.55”

$community = “nynexnet”

$action = “ip”

$oid = “ipDefaultTTL”

$oidname = “ipDefaultTTL”

$oidnum = “1.3.6.1.2.1.4.2.0”

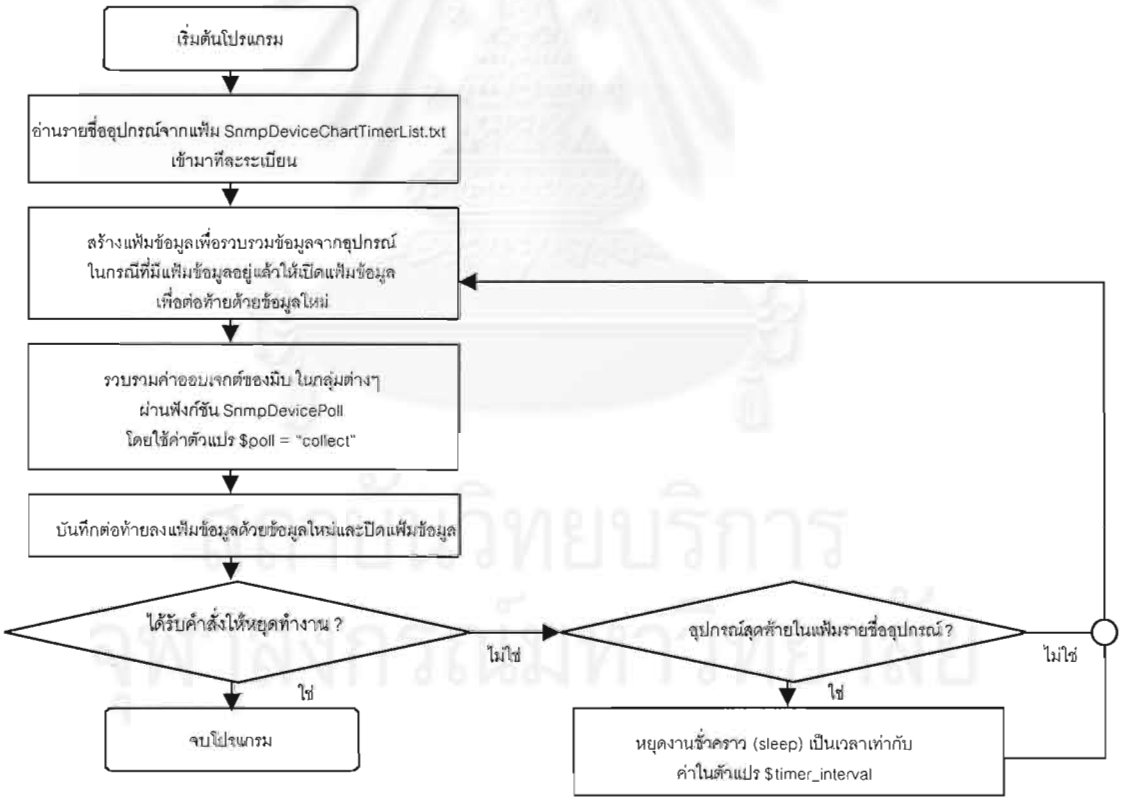
$oidvaluenew = “128”

4.1.4 โปรแกรมจัดการแผนภูมิ (SnmpDeviceChartTimer.cgi)

วัตถุประสงค์เพื่อให้สามารถรวบรวมข้อมูลทางสถิติของอุปกรณ์และบันทึกไว้ในแฟ้มข้อมูลสำหรับใช้นำเสนอในรูปแบบของแผนภูมิ ฟังก์ชัน SnmpDeviceChartTimer.cgi มีขั้นตอนการทำงานแสดงดังรูปที่ 4.2 โดยสามารถกำหนดการเรียกใช้ฟังก์ชัน SnmpDevicePoll เพื่อรวบรวมข้อมูลทางสถิติเหล่านี้ ตั้งแต่ทุก 1 นาที ไปจนถึง ทุก 60 นาที ผ่านตัวแปร $timer_interval จนกว่าได้รับคำสั่งให้หยุดการทำงาน วิธีการเรียกใช้ฟังก์ชัน SnmpDevicePoll จะเป็นการเริ่มต้นกระบวนการ Win32::Spawn Process ของวินโดวส์เอ็นที [15] เพื่อให้สามารถรวบรวมข้อมูลจากอุปกรณ์ได้พร้อมกันหลายตัว

โปรแกรม SnmpDeviceChartTimer.cgi มีรายละเอียดของการใช้กระบวนการ Win32::Spawn Process ดังนี้

```
### เปิดแฟ้มข้อมูลรายชื่ออุปกรณ์
open(SnmpDeviceChartTimerListTxt, $SnmpDeviceChartTimerListTxt);
### อ่านรายชื่ออุปกรณ์เข้ามาทีละระเบียน
($poll_ip,$sysName,$community) = <SnmpDeviceChartTimerListTxt>;
while ($poll_ip) {
### สร้างแฟ้มข้อมูลชั่วคราวเพื่อบันทึกคำสั่งเรียกใช้ฟังก์ชัน SnmpDevicePoll
        $SnmpDeviceChartTimerCmd=">$drive\\SNMPtemp\\$poll_ip.cmd";
        open(SnmpDeviceChartTimerCmd, $SnmpDeviceChartTimerCmd);
        print SnmpDeviceChartTimerCmd
"$perldrive\\perl\.exe $cgidrive\\SnmpDevicePoll $poll_ip $community $timer_interval\n";
        close(SnmpDeviceChartTimerCmd);
        Win32::Spawn("$drive\\SNMPtemp\\$poll_ip.cmd","",$pid);
        ($poll_ip,$sysName,$community) = <SnmpDeviceChartTimerListTxt>;
}
close(SnmpDeviceChartTimerListTxt); ### ปิดแฟ้มข้อมูลรายชื่ออุปกรณ์
```



รูปที่ 4.2 แสดงการทำงานของโปรแกรม SnmpDeviceChartTimer.cgi

4.2 ส่วนจัดการโพรโทคอลเอสเอ็นเอ็มพี

วัตถุประสงค์เพื่อให้สามารถติดต่อจัดการอุปกรณ์ด้วยโอเปอเรชั่นต่างๆตามข้อกำหนดของโพรโทคอลเอสเอ็นเอ็มพีเพื่อสนับสนุนการทำงานของโปรแกรมในส่วนจัดการอุปกรณ์ทางเครือข่าย โดยแบ่งการพัฒนาเป็น 2 ส่วนคือ

1. โปรแกรมจัดการโอเปอเรชั่นเอสเอ็นเอ็มพี
2. โปรแกรมจัดการเบสิกเอ็นโค้ดดิ้งรูล

4.2.1 โปรแกรมจัดการโอเปอเรชั่นเอสเอ็นเอ็มพี

มีหน้าที่ติดต่อรวบรวมข้อมูลจากอุปกรณ์ โดยสามารถอธิบายตามประเภทของการใช้งานผ่านการเรียกใช้ฟังก์ชันดังต่อไปนี้

4.2.1.1 ฟังก์ชัน SnmpGetScalarObject

มีหน้าที่ใช้ โอเปอเรชั่น เก็ต เพื่ออ่านค่าของออบเจกต์ประเภทค่าเดี่ยว ขั้นตอนการทำงานของฟังก์ชัน SnmpGetScalarObject แสดงดังรูปที่ 4.3

รูปที่ 4.3 แสดงการทำงานของฟังก์ชัน SnmpGetScalarObject

ฟังก์ชัน SnmpGetScalarObject มีรูปแบบการเรียกใช้ดังนี้

```
&SnmpDeviceGetScalarObject($poll_ip,$community,@oidname);
```

$poll_ip คือ หมายเลขไอพีของอุปกรณ์ที่ต้องการรวบรวมข้อมูล

$community คือ ค่าคอมมิวนิตี้สตริง

@oidname คือ รายชื่อออบเจกต์สำหรับรวบรวมข้อมูล

รายละเอียดของฟังก์ชัน SnmpGetScalarObject

```
sub SnmpGetScalarObject {
        my($poll_ip,$community,@oidname) = @_;
        my($oidname,@encode_oid,$SnmpSession,$VarBind,$VarBindList,$oid_next,$oidvalue);
### แปลงหมายเลขออบเจกต์ให้เป็นไบต์โค้ด
        @encode_oid = &SnmpEncodeOIDList(@oidname);
### เปิดเอสเอ็นเอ็มพีเซสชั่นกับอุปกรณ์
        $SnmpSession = &SnmpSessionOpen($poll_ip,$community,161);
### เรียกใช้โอเปอเรชั่นเก็ต เพื่อส่งเก็ตเมสเสจ
        &SnmpGetOperation($SnmpSession,@encode_oid);
### รับและดีโค้ด เก็ต-เรสปอนส์ เมสเสจ
        $VarBindList = &SnmpRetrieveVarBindList($SnmpSession);
### ปิดเอสเอ็นเอ็มพีเซสชั่นกับอุปกรณ์
        &SnmpSessionClose($SnmpSession);
### รวบรวมค่าออบเจกต์ จาก เก็ต-เรสปอนส์ เมสเสจ
        while ($VarBindList) {
                ($oidvalue,$oid_next,$VarBindList) =
                        &SnmpRetrieveOIDValue($VarBindList);
                $oidname = shift(@oidname);
### รวบรวมค่าออบเจกต์ ในตัวแปร $oidvaluescalar{$oidname} ของ
### โครงสร้างข้อมูลแบบแฮช โดยใช้ชื่อออบเจกต์เป็นเลขที่อยู่แบบแฮช
                $oidvaluescalar{$oidname} = $oidvalue;
        } # end while loop
}
```

4.2.1.2 ฟังก์ชัน SnmpGetTabularObject

มีหน้าที่ใช้โอเปอเรชั่น เก็ต-เน็กส์เพื่ออ่านค่าออบเจกต์ประเภทโครงสร้างแบบตาราง ขั้นตอนการทำงานของฟังก์ชัน SnmpGetTabularObject แสดงดังรูปที่ 4.4

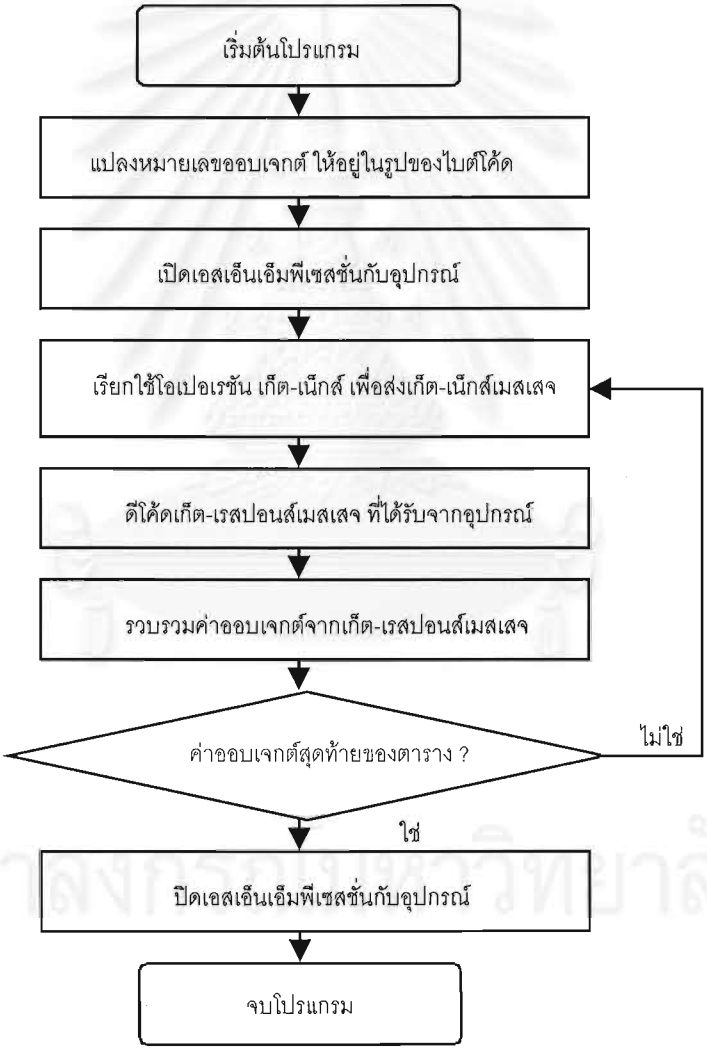
ฟังก์ชัน SnmpGetTabularObject มีรูปแบบการเรียกใช้ดังนี้

```
&SnmpDeviceGetTabularObject($poll_ip,$community,@oidname);
```

$poll_ip คือ หมายเลขไอพีของอุปกรณ์ที่ต้องการรวบรวมข้อมูล

$community คือ ค่าคอมมิวนิตี้สตริง

@oidname คือ รายชื่อออบเจกต์สำหรับรวบรวมข้อมูล



รูปที่ 4.4 แสดงการทำงานของฟังก์ชัน SnmpGetTabularObject

รายละเอียดของฟังก์ชัน SnmpGetTabularObject

```
sub SnmpGetTabularObject {
 my($poll_ip,$community,@oidname) = @_;
 my($oidname,$encode_oid,@encode_oid,$oid,$SnmpSession,$VarBindList,$oid_next,$oidvalue);
 my($Index1,$Index2,$Index3,$Index4);
 @encode_oid = &SnmpEncodeOIDList(@oidname); ###  แปลงหมายเลขออบเจกต์ให้เป็นไบต์โค้ด
 $SnmpSession = &SnmpSessionOpen($poll_ip,$community,161); ### เปิดเอสเอ็นเอ็มพีเซสชั่น
 foreach $encode_oid (@encode_oid) {
        $oid = $encode_oid;
        $oidname = shift(@oidname);
        for(;;) {   ### เรียกใช้โอเปอเรชั่นเก็ต เพื่อส่งเก็ตเมสเสจ
                &SnmpGetNextOperation($SnmpSession,$encode_oid);
### รับและดีโค้ด เก็ต-เรสปอนส์ เมสเสจ
                $VarBindList = &SnmpRetrieveVarBindList($SnmpSession);
### รวบรวมค่าออบเจกต์ จาก เก็ต-เรสปอนส์ เมสเสจ
                ($oidvalue,$oid_next,$VarBindList) = &SnmpRetrieveOIDValue($VarBindList);
                last unless BER::encoded_oid_prefix_p($oid,$oid_next); ### หยุดเมื่อได้ค่าสุดท้าย
### รวบรวมค่าออบเจกต์ ในตัวแปร $oidvaluetabular{$oidname} ของโครงสร้าง
### ข้อมูลแแบบแฮช โดยใช้ $oidname $Index1 $Index2 $Index3 และ Index4 เป็นเลขที่อยู่แบบแฮช
                $Index1 = shift(@Index1);
                $Index2 = shift(@Index2);
                if (@Index4) {
                        $Index3 = shift(@Index3);
                        $Index4 = shift(@Index4);
                $oidvaluetabular{$oidname,$Index1,$Index2,$Index3,$Index4} = $oidvalue;
                } elsif (@TableIndex2) {
                        $oidvaluetabular{$oidname,$Index1,$Index2} = $oidvalue;
                } else {
                        $oidvaluetabular{$oidname,$Index1} = $oidvalue;
                }
                $encode_oid = $oid_next;
        } # end for loop
 } # end foreach loop
 &SnmpSessionClose($SnmpSession);   ### ปิดเอสเอ็นเอ็มพีเซสชั่นกับอุปกรณ์
} # end sub SnmpGetTabularObject
```

4.2.1.3 ฟังก์ชัน SnmpSetObject

มีหน้าที่ใช้ โอเปอเรชั่น เซ็ต เพื่อแก้ไขค่าออบเจกต์ ขั้นตอนการทำงานของฟังก์ชัน SnmpSetObject แสดงดังรูปที่ 4.5

รูปที่ 4.5 แสดงการทำงานของฟังก์ชัน SnmpSetObject

ฟังก์ชัน SnmpSetObject มีรูปแบบการเรียกใช้ดังนี้

```
&SnmpDeviceSetObject($poll_ip,$community,$oidname,$oidnum,$oidvaluenew);
```

$poll_ip คือ หมายเลขไอพีของอุปกรณ์ที่ต้องการแก้ไขข้อมูล

$community คือ ค่าคอมมิวนิตี้สตริง

$oidname คือ ชื่อของออบเจกต์ที่ต้องการแก้ไขข้อมูล

$oidnum คือ หมายเลขออบเจกต์ที่ต้องการแก้ไขข้อมูล

$oidvaluenew คือ ค่าใหม่ของออบเจกต์ที่ต้องการแก้ไข

รายละเอียดของฟังก์ชัน SnmpSetObject

```
sub SnmpSetObject {
        my($poll_ip,$community,$oidname,$oidnum,$oidvaluenew) = @_;
        my($encode_oid,$encode_oidvaluenew,@encode_oidnew,$mibtype);
        my($SnmpSession,$VarBindList,$oid_next,$oidvalue,@oidvalue);
### แปลงหมายเลขออบเจกต์ให้เป็นไบต์โค้ด
        ($encode_oid) = &SnmpEncodeOIDList(($oidnum));
### แปลงค่าใหม่ของออบเจกต์ให้เป็นไบต์โค้ด ตามชนิดของค่าออบเจกต์
        $mibtype = $SnmpMIBvalue::Type{$oidname};
        if ($mibtype eq "string") {
                $encode_oidvaluenew = encode_string($oidvaluenew);
        } elsif ($mibtype eq "integer") {
                $encode_oidvaluenew = encode_int($oidvaluenew);
        } elsif ($mibtype eq "poll_ip") {
                $encode_oidvaluenew = encode_ip_address($oidvaluenew);
        } else {
                $encode_oidvaluenew = encode_string($oidvaluenew);
        }
        push(@encode_oidnew, [$encode_oid,$encode_oidvaluenew]);
### เปิดเอสเอ็นเอ็มพีเซสชั่นกับอุปกรณ์
        $SnmpSession = &SnmpSessionOpen($poll_ip,$community,161);
### เรียกใช้โอเปอเรชั่นเก็ต เพื่อส่งเซ็ตเมสเสจ แก้ไขค่าออบเจกต์
        &SnmpSetOperation($SnmpSession,@encode_oidnew);
### ปิดเอสเอ็นเอ็มพีเซสชั่นกับอุปกรณ์
        &SnmpSessionClose($SnmpSession);
}
```

4.2.1.4 ฟังก์ชัน SnmpDevicePoll

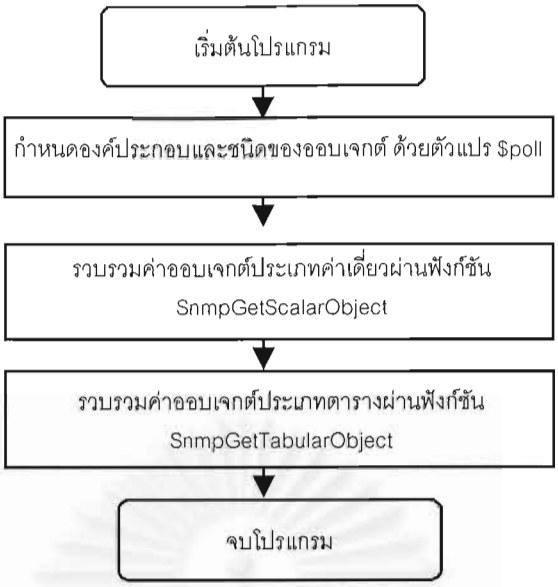
มีหน้าที่ติดต่อกับอุปกรณ์โดยการเรียกใช้ฟังก์ชัน SnmpGetScalarObject และ ฟังก์ชัน SnmpGetTabularObject ซึ่งได้กล่าวมาแล้วข้างต้น เพื่ออ่านค่าออบเจกต์ ขั้นตอนการทำงานของฟังก์ชัน SnmpDevicePoll แสดงดังรูปที่ 4.6 โดยค่าของตัวแปร $poll จะกำหนดองค์ประกอบและชนิดของออบเจกต์ ที่ต้องการอ่านจากอุปกรณ์ ดังแสดงในตารางที่ 4.4 ค่าออบเจกต์ประเภทค่าเดี่ยวจะรวบรวมไว้ในโครงสร้างข้อมูลแบบแฮช %oidvaluescalar และค่าออบเจกต์ประเภทตาราง

จะรวบรวมใน %oidvaluetabular โดยใช้ชื่อออบเจกต์เป็นเลขที่อยู่แบบแฮช เช่น ค่าออบเจกต์ sysName จะถูกเก็บไว้ในตัวแปร $oidvaluescalar{'sysName'}

ค่าตัวแปร $poll	ออบเจกต์	เรียกใช้โดยฟังก์ชัน
"discovery"	sysObjectID sysName sysDescr และ sysLocation	SnmpDeviceDiscovery
"file"	sysObjectID sysName sysDescr และ sysLocation	SnmpDevicePollFile
"address"	กลุ่มซิสเต็มของมิบ-ทู	SnmpDevicePollAddress
"specific"	อ่านได้ครั้งละ 1 ออบเจกต์	SnmpDevicePollSpecificOID
"system"	กลุ่มซิสเต็มของมิบ-ทู	SnmpDevicePollSystem
"interfaces"	กลุ่มอินเตอร์เฟสของมิบ-ทู	SnmpDevicePollInterfaces
"ip"	กลุ่มไอพีของมิบ-ทู	SnmpDevicePollIP
"tcp"	กลุ่มทีซีพีของมิบ-ทู	SnmpDevicePollTCP
"udp"	กลุ่มยูดีพีของมิบ-ทู	SnmpDevicePollUDP
"cisco"	กลุ่มโพรดักส์ กลุ่มแซสซี และกลุ่มอินเตอร์เฟสของซิสโก้	SnmpDevicePollCisco
"bay"	กลุ่มฮาร์ดแวร์ของของเบย์เน็ตเวิร์ก	SnmpDevicePollBay
"3com"	กลุ่มลิ้งค์สวิชของทรีคอม	SnmpDevicePoll3Com
"collect"	กลุ่มอินเตอร์เฟสและกลุ่มไอพีของมิบ-ทู กลุ่มอามอน กลุ่มอินเตอร์เฟสของซิสโก้	SnmpDeviceChart

ตารางที่ 4.4 แสดงการกำหนดองค์ประกอบและชนิดของออบเจกต์ โดยค่าของตัวแปร $poll

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.6 แสดงการทำงานของฟังก์ชัน SnmpDevicePoll

รูปแบบการเรียกใช้ ฟังก์ชัน SnmpDevicePoll

```
&SnmpDevicePoll($poll_ip,$community,$poll);
```

$poll_ip คือ หมายเลขไอพีของอุปกรณ์ที่ต้องการค้นหา

$community คือ ค่าคอมมิวนิตี้สตริง

$poll ใช้กำหนดออบเจกต์ประเภทต่างๆที่ต้องการอ่านจากอุปกรณ์

ตัวอย่าง การเรียกใช้ฟังก์ชัน SnmpDevicePoll จากฟังก์ชัน SnmpDevicePollBay

```
sub SnmpDevicePollBay {
        my($success,$poll);
### โครงสร้างข้อมูลแบบแฮชสำหรับรวบรวมค่าออบเจกต์ต่างๆ
        local(%oidvaluescalar,%oidvaluetabular);
### ต้องการไพร์เวทมิบของเบย์เน็ตเวิร์ก
        $poll = "bay";
        $success = &SnmpDevicePoll($poll_ip,$community,$poll);
        if ($success) {
### นำค่าออบเจกต์ที่รวบรวมได้มาแสดงผลในเว็บเพจ ผ่านฟังก์ชัน SnmpDevicePollBayHtml
                &SnmpDevicePollBayHtml;
        } else {
### แจ้งข้อผิดพลาดผ่านฟังก์ชัน SnmpDevicePollFailHtml
                &SnmpDevicePollFailHtml($SnmpDeviceMessage::Error{'Poll'});
        }
}
```

4.2.2 โปรแกรมจัดการเบสิกเอ็นโค้ดดิ้งรูล

เป็นส่วนทำหน้าที่ส่ง เอสเอ็นเอ็มพีเมสเสจ ของโอเปอเรชั่นเอสเอ็นเอ็มพีในรูปของไบต์โค้ด ด้วยกฎเบสิกเอ็นโค้ดดิ้งรูล โดยอาศัยเพิร์ลโมดูล BER.pm และ SNMP_Session.pm [10] ซึ่งประกอบไปด้วยฟังก์ชันที่สำคัญดังนี้

1. ฟังก์ชัน get_request_response มีหน้าที่เอ็นโค้ด (Encode) เก็ต เมสเสจ
2. ฟังก์ชัน getnext_request_response มีหน้าที่เอ็นโค้ดเก็ต-เน็กส์ เมสเสจ
3. ฟังก์ชัน set_request_response มีหน้าที่เอ็นโค้ดเซ็ต เมสเสจ
4. ฟังก์ชัน decode_get_response ใช้ดีโค้ด (Decode) เก็ต-เรสปอนส์ เมสเสจ ซึ่งเป็นข้อมูลที่ส่งมาจากอุปกรณ์ และยังอยู่ในรูปแบบของไบต์โค้ด

เพื่อให้สามารถเรียกใช้ฟังก์ชันเหล่านี้ได้อย่างเหมาะสมกับการทำงานของโปรแกรมจัดการอุปกรณ์ฯ จึงต้องมีการพัฒนาฟังก์ชันเพิ่มเติม ดังต่อไปนี้

4.2.2.1 ฟังก์ชัน SnmpGetOperation

วัตถุประสงค์เพื่อส่งเก็ตเมสเสจ ผ่านฟังก์ชัน get_request_response

รูปแบบการเรียกใช้ ฟังก์ชัน SnmpGetOperation

```
&SnmpGetOperation($SnmpSession,@encode_oid);
```

$SnmpSession คือ ตัวชี้ (Pointer) ของ เอสเอ็นเอ็มพีเซสชั่น

@encode_oid คือ รายชื่อหมายเลขออบเจกต์ สำหรับส่งไปในเก็ตเมสเสจ

รายละเอียดของฟังก์ชัน SnmpGetOperation

```
sub SnmpGetOperation {
    my($SnmpSession,@encode_oid) = @_;
    $SnmpSession->get_request_response(@encode_oid);
}
```

4.2.2.2 ฟังก์ชัน SnmpGetNextOperation

วัตถุประสงค์เพื่อส่งเก็ต-เน็กส์เมสเสจ ผ่านฟังก์ชัน getnext_request_response

รูปแบบการเรียกใช้ ฟังก์ชัน SnmpGetNextOperation

```
&SnmpGetNextOperation($SnmpSession,@encode_oid);
```

$SnmpSession คือ ตัวชี้ของเอสเอ็นเอ็มพีเซสชั่น

@encode_oid คือ รายชื่อหมายเลขออบเจกต์ สำหรับส่งไปในเก็ต-เน็กส์เมสเสจ

รายละเอียดของฟังก์ชัน SnmpGetNextOperation

```
sub SnmpGetNextOperation {
        my($SnmpSession,$encode_oid) = @_;
        $SnmpSession->getnext_request_response($encode_oid);
}
```

4.2.2.3 ฟังก์ชัน SnmpSetOperation

วัตถุประสงค์เพื่อส่งเซ็ตเมสเสจ ผ่านฟังก์ชัน set_request_response

รูปแบบการเรียกใช้ ฟังก์ชัน SnmpSetOperation

```
&SnmpSetOperation($SnmpSession,@encode_oid);
```

$SnmpSession คือ ตัวชี้ของ เอสเอ็นเอ็มพีเซสชั่น

@encode_oid คือ หมายเลขออบเจกต์ พร้อมค่าใหม่สำหรับส่งไปในเซ็ตเมสเสจ

รายละเอียดของฟังก์ชัน SnmpSetOperation

```
sub SnmpSetOperation {
        my($SnmpSession,@encode_oid) = @_;
        $SnmpSession->set_request_response(@encode_oid);
}
```

4.2.2.4 ฟังก์ชัน SnmpRetrieveVarBindList

วัตถุประสงค์เพื่อดีโค้ด เก็ต-เรสปอนส์ เมสเสจ ผ่านฟังก์ชัน decode_get_response

รูปแบบการเรียกใช้ ฟังก์ชัน SnmpRetrieveVarBindList

```
&SnmpRetrieveVarBindList($SnmpSession);
```

$SnmpSession คือตัวชี้ของ เอสเอ็นเอ็มพีเซสชั่น

รายละเอียดของฟังก์ชัน SnmpRetrieveVarBindList

```
sub SnmpRetrieveVarBindList {
        my($SnmpSession) = @_;
        my($SnmpGetResponsePDU,$VarBindList,$VarBind,$oid,$oidvalue,@oidvalue);
### รับเก็ต-เรสปอนส์ พีดียู
        $SnmpGetResponsePDU = $SnmpSession->pdu_buffer;
###  แยกเขตข้อมูล Variable Bindings จาก เก็ต-เรสปอนส์ พีดียู
        ($VarBindList) = $SnmpSession->decode_get_response($SnmpGetResponsePDU);
        return($VarBindList);
}
```

4.2.2.5 ฟังก์ชัน SnmpRetrieveOIDValue

วัตถุประสงค์เพื่อรวบรวมค่าออบเจกต์ และ ดีโค้ดตามชนิดข้อมูลของแต่ละออบเจกต์ ในเขตข้อมูลแวเรียเบิลบายดิ้ง (Variable Bindings) ของ เก็ต-เรสปอนส์ พีดียู

รูปแบบการเรียกใช้ ฟังก์ชัน SnmpRetrieveOIDValue

```
& SnmpRetrieveOIDValue($VarBindList);
```

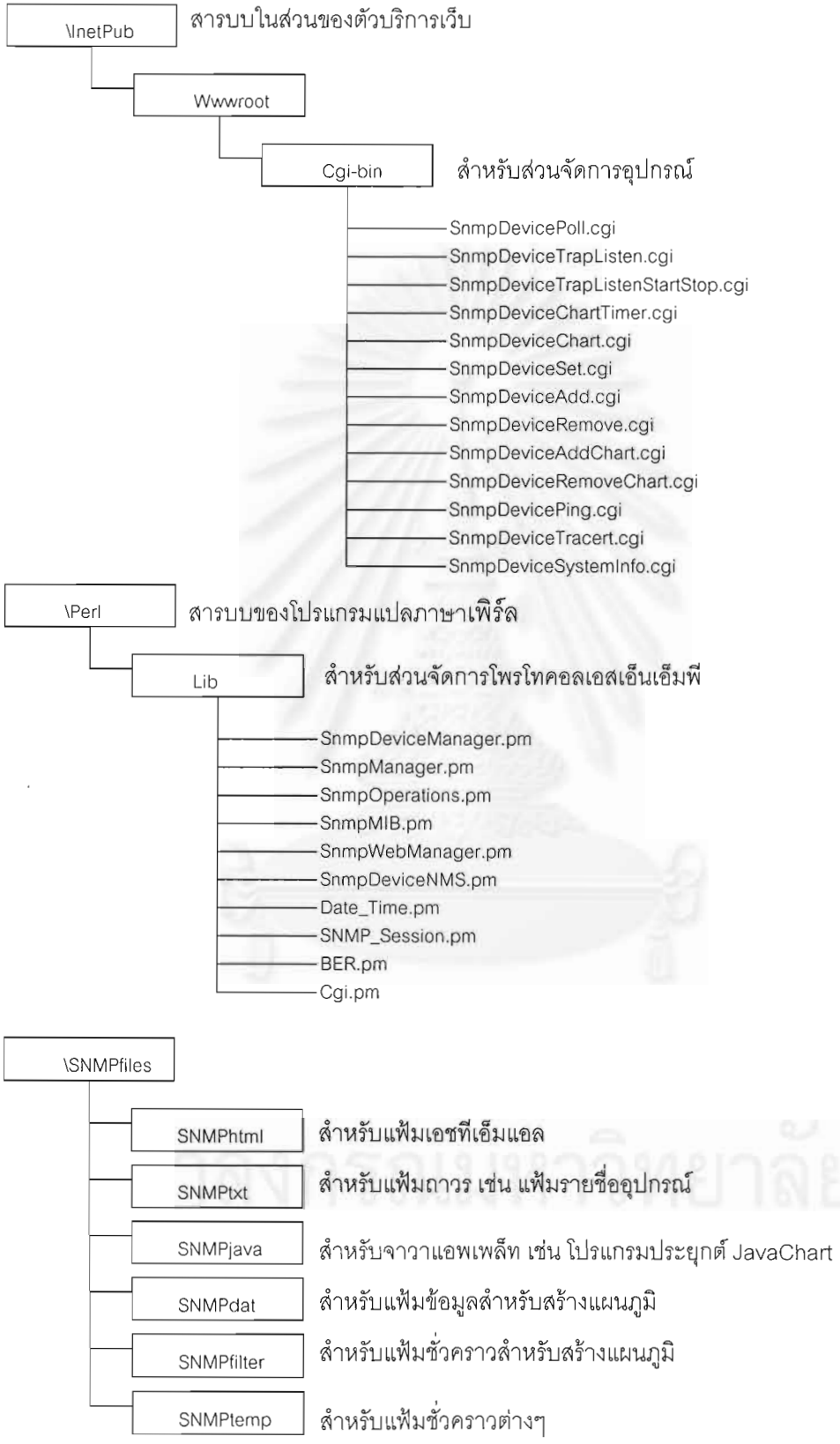
$VarBindList คือ ค่าออบเจกต์ต่างๆในเขตข้อมูลแวเรียเบิลบายดิ้งของ เก็ต-เรสปอนส์ พีดียู

รายละเอียดของฟังก์ชัน SnmpRetrieveOIDValue

```
sub SnmpRetrieveOIDValue {
        my($VarBindList) = @_;
        my($oid_next,$oidvalue,$VarBind);
### แยกค่าออบเจกต์ จาก เขตข้อมูล Varaible Bindings
        ($VarBind,$VarBindList) = decode_sequence($VarBindList);
### รวบรวมค่าออบเจกต์ และ ดีโค้ดตามชนิดของข้อมูลในออบเจกต์
        ($oid_next,$oidvalue) = decode_by_template($VarBind,"%O%@");
        if ($SnmpMIBvalue::Type{$oidname} eq "octet") {
                $oidvalue = hex_string($oidvalue); ### จัดรูปแบบเป็นไบต์
        } else {
                $oidvalue = pretty_print($oidvalue); ### จัดรูปแบบเป็นสายอักขระ
        }
### ตัวแปร $oidvalue คือ ค่าออบเจกต์ และ ตัวแปร $oid_next คือ หมายเลขออบเจกต์
        return($oidvalue,$oid_next,$VarBindList);
}
```

4.3 การติดตั้งโปรแกรมสำหรับเครื่องให้บริการ

โครงสร้างของระบบแฟ้มสำหรับโปรแกรมจัดการอุปกรณ์ ฯ ประกอบด้วย



รูปที่ 4.7 แสดงโครงสร้างของระบบแฟ้ม

โดยระบบแฟ้มทั้งหมดจะต้องสร้างในสารบบหลักของหน่วยขับ (Drive) และต้องสร้างสารบบเสมือน (Virtual Directory) ในส่วนของตัวบริการเว็บของเครื่องให้บริการ ดังนี้

/SNMPhtml = c:\SNMPfiles\SNMPhtml

/SNMPtxt = c:\SNMPfiles\SNMPtxt

/SNMPjava = c:\SNMPfiles\SNMPjava

/SNMPdat = c:\SNMPfiles\SNMPdat

/SNMPfilter = c:\SNMPfiles\SNMPfilter

/SNMPtemp = c:\SNMPfiles\SNMPtemp

จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5
การทดสอบโปรแกรม

การทดสอบโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่าย เป็นการทดสอบการใช้งานของโปรแกรมในการจัดการอุปกรณ์ชนิดต่างๆ โดยอาศัยโอเปอเรชั่นเอสเอ็นเอ็มพี ดังนี้

1. ค้นหาอุปกรณ์
2. สอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์
3. สอบถามตามหมายเลขไอพี
4. สอบถามตามรายชื่อออบเจกต์
5. รายงานข้อมูลในรูปแผนภูมิ
6. รายงานแทร็ปเมสเสจ
7. แก้ไขอุปกรณ์

5.1 ขอบเขตการทดสอบ

1. เครื่องให้บริการ

สภาพแวดล้อมฮาร์ดแวร์

- หน่วยประมวลผลกลาง Pentium 200
- หน่วยความจำหลัก (RAM) ความจุ 64 MB
- ฮาร์ดดิสก์ ความจุ 800 MB
- แผ่นวงจรต่อประสานข่ายงาน (Network Interface Card) ของเอสเอ็มซี (SMC) รุ่น PCI 10/100 โดยใช้งานที่ความเร็ว 10 Mbps และกำหนดหมายเลขไอพี 150.2.24.101

สภาพแวดล้อมซอฟต์แวร์

- ระบบปฏิบัติการวินโดวส์ เอ็นที เวอร์ชั่น 4.0
- อินเทอร์เน็ตอินฟอร์เมชั่นเซิร์ฟเวอร์ เวอร์ชั่น 4.0
- แอคทีฟสเตท เพิร์ล 5 (ActiveState Perl 5) เวอร์ชั่น 517
- โปรแกรมสร้างแผนภูมิ จาวาชาท (JavaChart)

2. เครื่องขอรับบริการ

สภาพแวดล้อมฮาร์ดแวร์

- หน่วยประมวลผลกลาง 486 DX4-100
- หน่วยความจำหลัก ความจุ 32 MB
- ฮาร์ดดิสก์ ความจุ 200 MB
- แผ่นวงจรต่อประสานข่ายงาน (Network Interface Card) ของเอสเอ็มซี (SMC) รุ่น PCI 10/100 โดยใช้งานที่ความเร็ว 10 Mbps และกำหนดหมายเลขไอพี 150.2.24.102

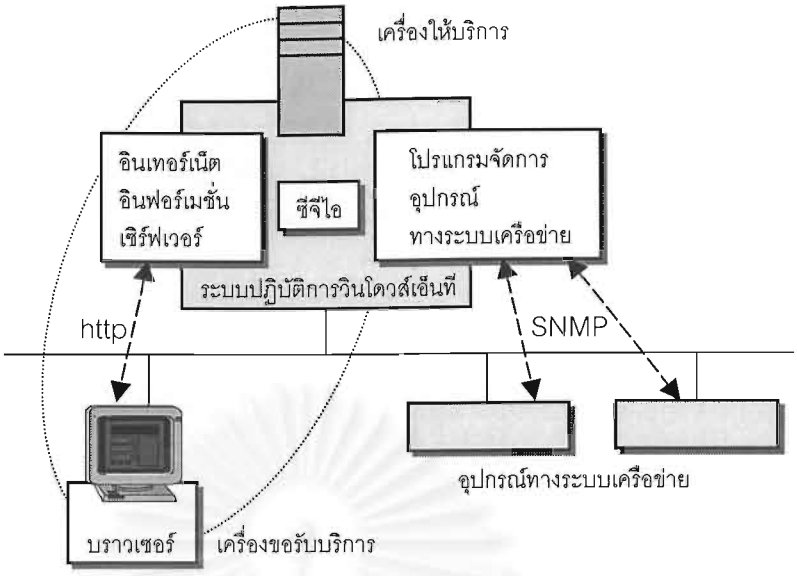
สภาพแวดล้อมซอฟต์แวร์

- ระบบปฏิบัติการวินโดวส์ 95
- เน็ตสเคปนาวิเกเตอร์ เวอร์ชั่น 4.6

3. อุปกรณ์ทางระบบเครือข่าย ประกอบไปด้วย

1) อุปกรณ์จัดเส้นทาง ของ เบย์เน็ตเวิร์ก รุ่น BLN หมายเลขไอพี 150.2.24.1
2) เครื่องให้บริการ บนระบบปฏิบัติการไมโครซอฟต์วินโดวส์เอ็นที เวอร์ชั่น 4.0 (ไม่ใช่ตัวเดียวกันกับเครื่องให้บริการของโปรแกรมจัดการอุปกรณ์ฯ) หมายเลขไอพี 150.2.24.10
3) อีเทอร์เน็ตสวิช ของ ทรีคอม รุ่น LinkSwitch 1000 หมายเลขไอพี 150.2.24.13
4) อุปกรณ์จัดเส้นทางของซิสโก้ รุ่น 4500 หมายเลขไอพี 150.2.24.16
5) อีเทอร์เน็ตสวิช ของฟอร์ซิสเต็ม รุ่น ES 3850 หมายเลขไอพี 150.2.24.64
6) เอทีเอ็มสวิช ของเบย์เน็ตเวิร์ก รุ่น Centillion 100 หมายเลขไอพี 150.50.254.250
7) เอทีเอ็มสวิช ของฟอร์ซิสเต็ม รุ่น ASX 200 หมายเลขไอพี 200.200.1.2

4. เครือข่ายที่ใช้ทดสอบโปรแกรม เป็นระบบเครือข่ายภายใน เชื่อมโยงแบบอีเทอร์เน็ต ประกอบไปด้วย เครื่องให้บริการ เครื่องขอรับบริการ และ อุปกรณ์ทางระบบเครือข่าย ตามระบุ ในข้อ 1 ถึง ข้อ 3 โดยมีสภาพแวดล้อมที่สามารถแสดงได้ดังแสดงรูปที่ 5.1



รูปที่ 5.1 แสดงสภาพแวดล้อมของการทดสอบโปรแกรมจัดการอุปกรณ์ทางระบบเครือข่าย

5.2 การทดสอบการค้นหาอุปกรณ์

เป็นการทดสอบโปรแกรมในการค้นหาอุปกรณ์ที่เชื่อมต่อกันอยู่ในระบบเครือข่ายตามที่ได้ระบุไว้ในขอบเขตของการทดสอบ

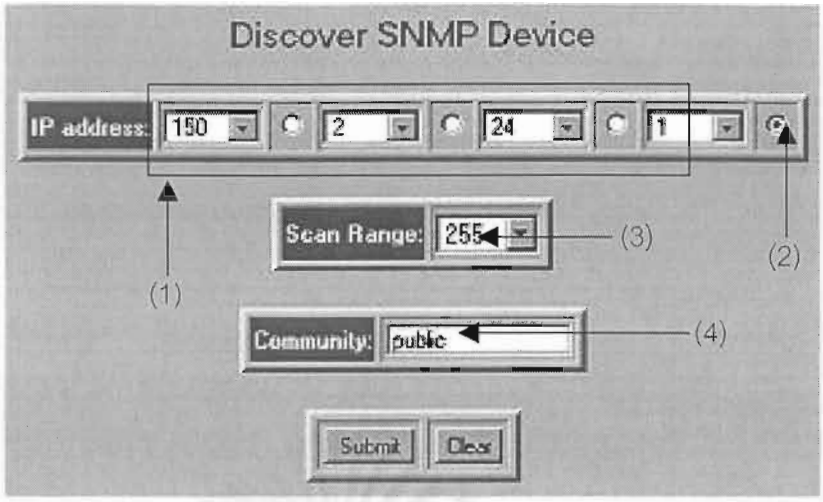
5.2.1 ขั้นตอนในการทดสอบ

1. จากหน้าจอหลัก เลือกรายการค้นหาอุปกรณ์
2. กำหนดข้อคำสั่ง (รูปที่ 5.2) ประกอบด้วย

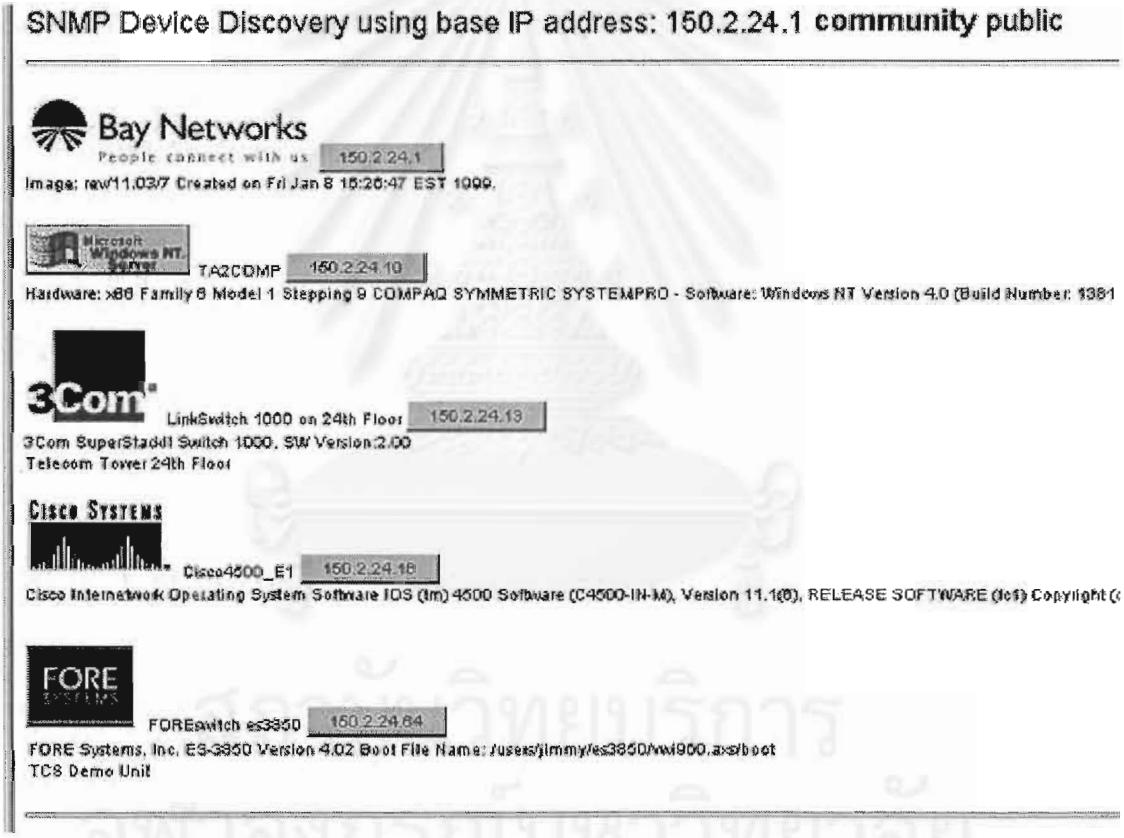
    1) กำหนดหมายเลขไอพีไอพีเริ่มต้น ทดสอบโดยใช้ค่า “150.2.24.1”
    2) กำหนดวิธีใช้หมายเลขไอพีในการค้นหาอุปกรณ์ โดยเลือกให้ส่วนที่ 4 ของหมายเลขไอพี เป็นตัวนับ
    3) กำหนดจำนวนหมายเลขไอพีที่ต้องการค้นหา ทดสอบโดยใช้ค่า “255”
    4) กำหนดค่าคอมมิวนิตี้สตริง ทดสอบโดยใช้ค่าโดยปริยาย “public”

5.2.2 ผลการทดสอบ

โปรแกรมสามารถค้นพบอุปกรณ์ได้ทั้งหมด ดังแสดงในรูปที่ 5.3 ยกเว้นเอทีเอ็มสวิช ของเบย์เน็ตเวิร์กหมายเลขไอพี 150.50.254.250 และ เอทีเอ็มสวิช ของฟอร์ซิสเต็ม หมายเลขไอพี 200.200.1.2 เพราะหมายเลขไอพีของอุปกรณ์ทั้ง 2 ไม่ได้อยู่ภายในกลุ่มของหมายเลขไอพีตามที่ได้ระบุไว้ในข้อคำสั่งของการทดสอบ คือ ตั้งแต่ 150.2.24.1 ถึง 150.2.24.255



รูปที่ 5.2 แสดงหน้าจอค้นหาอุปกรณ์



รูปที่ 5.3 แสดงผลการทดสอบการค้นหาอุปกรณ์

5.3 การทดสอบการสอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์

เป็นการทดสอบโปรแกรมในการสอบถามอุปกรณ์ ตามรายชื่อในแฟ้มข้อมูลรายชื่ออุปกรณ์

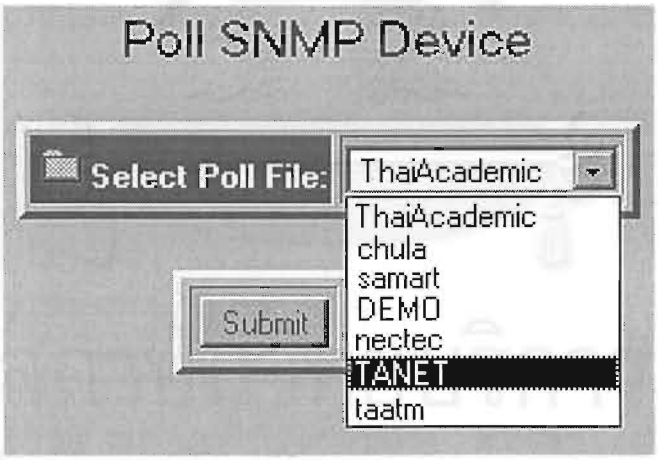
5.3.1 ขั้นตอนในการทดสอบ

1. สร้างแฟ้มข้อมูลรายชื่ออุปกรณ์ ชื่อ TANET ภายในแฟ้มประกอบไปด้วยหมายเลขไอพี และ ค่าคอมมิวนิตี้สตริง ของ อุปกรณ์ทั้งหมดที่เชื่อมต่อกันอยู่ในระบบเครือข่ายตามที่ได้ระบุไว้ในขอบเขตของการทดสอบดังนี้

```
150.2.24.1, BLN, public
150.2.24.10, TA2COMP, public
150.2.24.13, LinkSwitch1000 on 24th Floor, public
150.2.24.16, Cisco4500_E1, public
150.2.24.64, FOREswitch es3850, public
150.50.254.250, C100-TT17, public
200.200.1.2, FOREatm asx200bx, public
```

รูปที่ 5.4 แสดงตัวอย่างแฟ้มข้อมูลรายชื่ออุปกรณ์

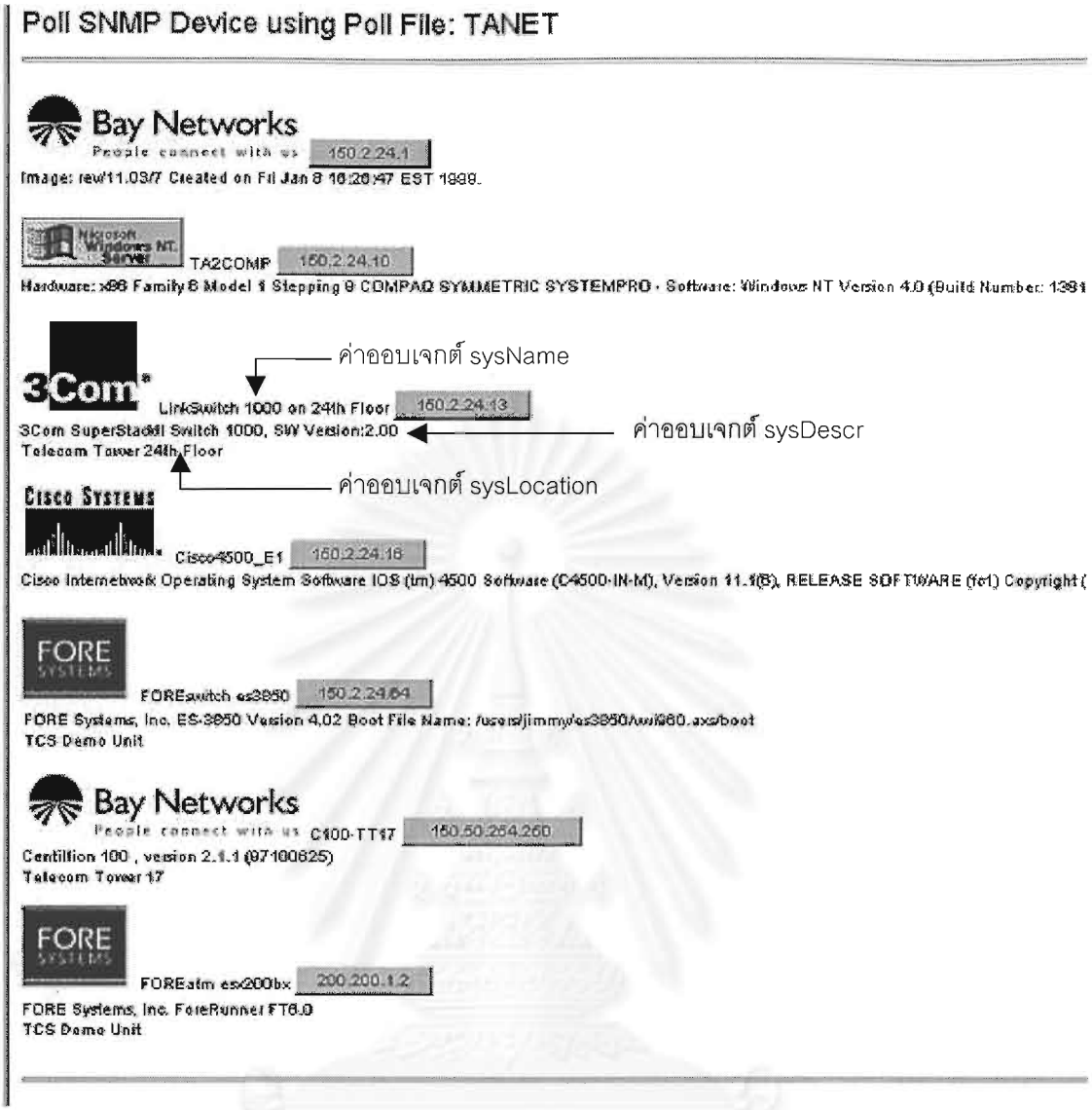
2. จากหน้าจอหลัก เลือกรายการสอบถาม ตามรายชื่อแฟ้มข้อมูลอุปกรณ์
3. เลือกแฟ้มข้อมูลรายชื่ออุปกรณ์ ชื่อ TANET (ดูรูปที่ 5.5)



รูปที่ 5.5 แสดงหน้าจอสอบถามตามรายชื่อแฟ้มข้อมูลอุปกรณ์

5.3.2 ผลการทดสอบ

โปรแกรมสามารถค้นพบอุปกรณ์ตามรายชื่อที่ปรากฏอยู่ในแฟ้มข้อมูลชื่อ TANET ได้ทั้งหมดดังแสดงในรูปที่ 5.6 โดยรวบรวมชื่อ (ออบเจกต์ sysName) รุ่น (ออบเจกต์ sysDescr) และสถานที่ตั้ง (ออบเจกต์ sysLocation) ของอุปกรณ์



รูปที่ 5.6 แสดงผลการทดสอบการสอบถาม ตามรายชื่อแฟ้มข้อมูลอุปกรณ์

5.4 การทดสอบการสอบถามตามหมายเลขไอพี

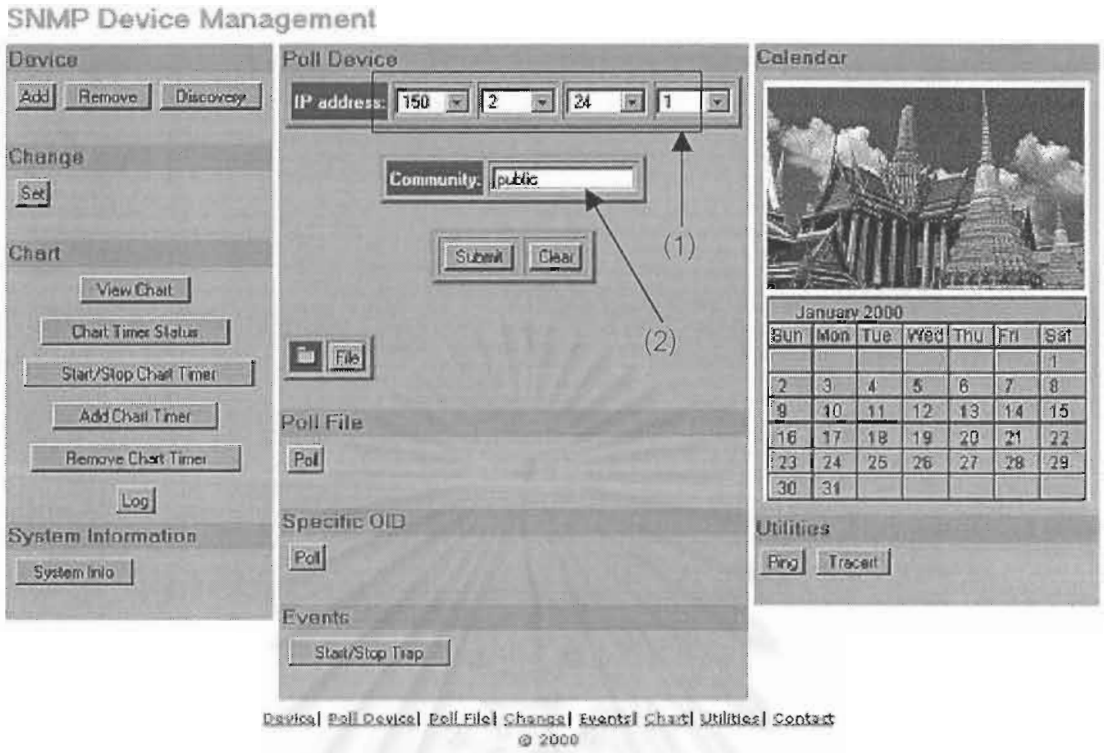
เป็นการทดสอบโปรแกรมในการสอบถามข้อมูลเกี่ยวกับอุปกรณ์ ด้วยหมายเลขไอพีตามต้องการ

5.4.1 ขั้นตอนในการทดสอบ

1. จากหน้าจอหลัก เลือกรายการสอบถามอุปกรณ์ตามหมายเลขไอพี
2. กำหนดข้อคำสั่ง (รูปที่ 5.7) ประกอบด้วย

    1) กำหนดหมายเลขไอพีของอุปกรณ์ที่ต้องการสอบถาม ทดสอบโดยใช้ค่า "150.2.24.1" ซึ่งเป็นหมายเลขไอพีของอุปกรณ์จัดเส้นทางของเบย์เน็ตเวิร์ก รุ่น BLN

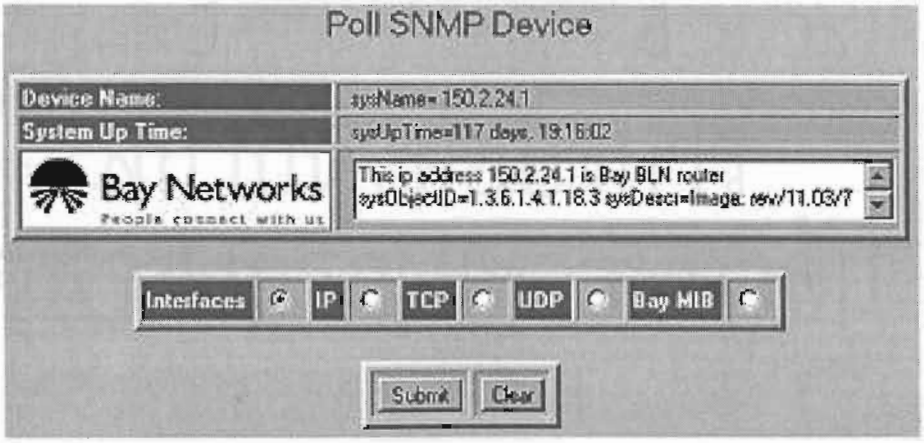
2) กำหนดค่าคอมมิวนิตี้สตริง ทดสอบโดยใช้ค่า "public"



รูปที่ 5.7 แสดงหน้าจอสอบถามอุปกรณ์ด้วยหมายเลขไอพี

5.4.2 ผลการทดสอบ

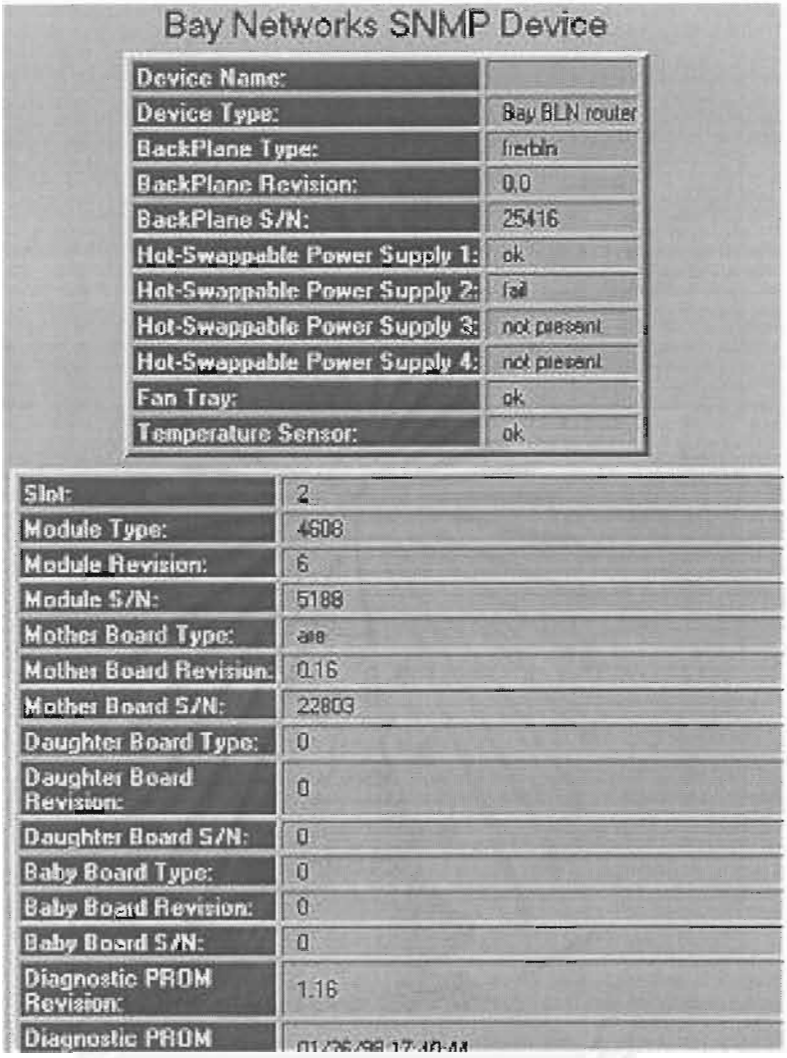
โปรแกรมแสดงข้อมูลเกี่ยวกับอุปกรณ์จัดเส้นทางของเบย์เน็ตเวิร์ก รุ่น BLN (ดังแสดงในรูปที่ 5.8) ซึ่งเป็นข้อมูลของออบเจกต์ในกลุ่มซิสเต็มภายใต้มิบ-ทู



รูปที่ 5.8 แสดงการรายงานข้อมูลของกลุ่มซิสเต็ม

จากหน้าจอนี้สามารถเรียกให้แสดงรายละเอียดของมิบในกลุ่มอื่น เช่น กลุ่มฮาร์ดแวร์ ซึ่งเป็น

ไพร์เวทมิบของ เบย์เน็ตเวิร์ก ดังแสดงในรูปที่ 5.9



Bay Networks SNMP Device

Device Name:	
Device Type:	Bay BLN router
BackPlane Type:	frerbln
BackPlane Revision:	0.0
BackPlane S/N:	25416
Hot-Swappable Power Supply 1:	ok
Hot-Swappable Power Supply 2:	fail
Hot-Swappable Power Supply 3:	not present
Hot-Swappable Power Supply 4:	not present
Fan Tray:	ok
Temperature Sensor:	ok

Slot:	2
Module Type:	4608
Module Revision:	6
Module S/N:	5188
Mother Board Type:	ase
Mother Board Revision:	0.16
Mother Board S/N:	22803
Daughter Board Type:	0
Daughter Board Revision:	0
Daughter Board S/N:	0
Baby Board Type:	0
Baby Board Revision:	0
Baby Board S/N:	0
Diagnostic PROM Revision:	1.16
Diagnostic PROM	01/26/98 17:40:44

รูปที่ 5.9 แสดงตัวอย่างการรายงานไพร์เวทมิบ

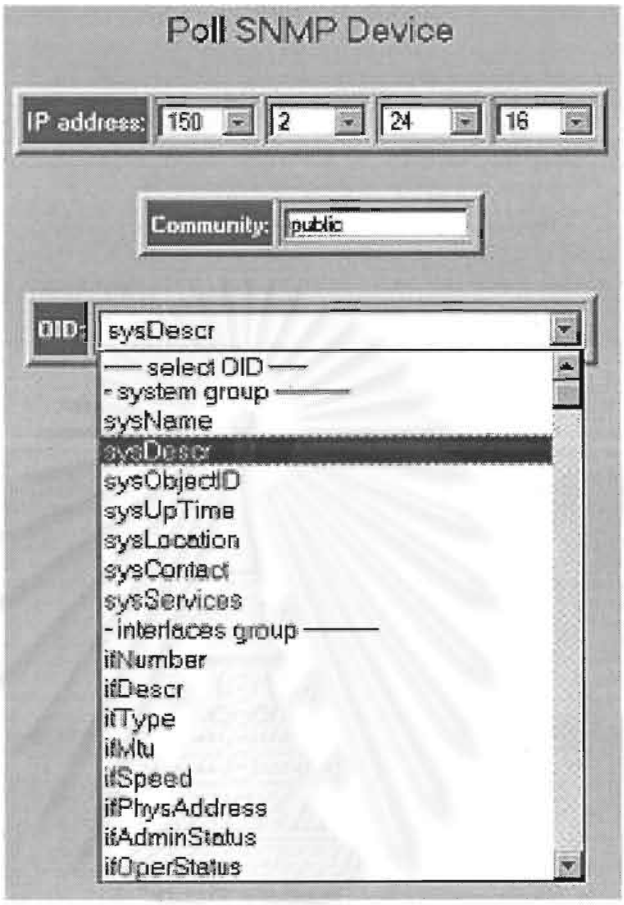
5.5 การทดสอบการสอบถามตามรายชื่อออบเจกต์

เป็นการทดสอบโปรแกรมในการสอบถามค่าของออบเจกต์ต่างๆตามต้องการของผู้ใช้งาน

5.5.1 ขั้นตอนในการทดสอบ

1. จากหน้าจอหลัก เลือกรายการสอบถามอุปกรณ์ตามรายชื่อออบเจกต์
2. กำหนดข้อคำสั่ง (รูปที่ 5.10) ประกอบด้วย
    1) กำหนดหมายเลขไอพีของอุปกรณ์ที่ต้องการสอบถาม ทดสอบโดยใช้ค่า “150.2.24.16” ซึ่งเป็นหมายเลขไอพีของอุปกรณ์จัดเส้นทาง ซิสโก้ รุ่น 4500
    2) กำหนดค่าคอมมิวนิตี้สตริง ทดสอบโดยใช้ค่า “public”
    3) เลือกออบเจกต์ที่ต้องการทราบค่า ทดสอบโดยเลือกออบเจกต์ sysUpTime

ของกลุ่มซิสเต็ม และ ออบเจกต์ cardDescr ของกลุ่มแชสซี ซึ่งเป็นไพร์เวทมิบของซิสโก้

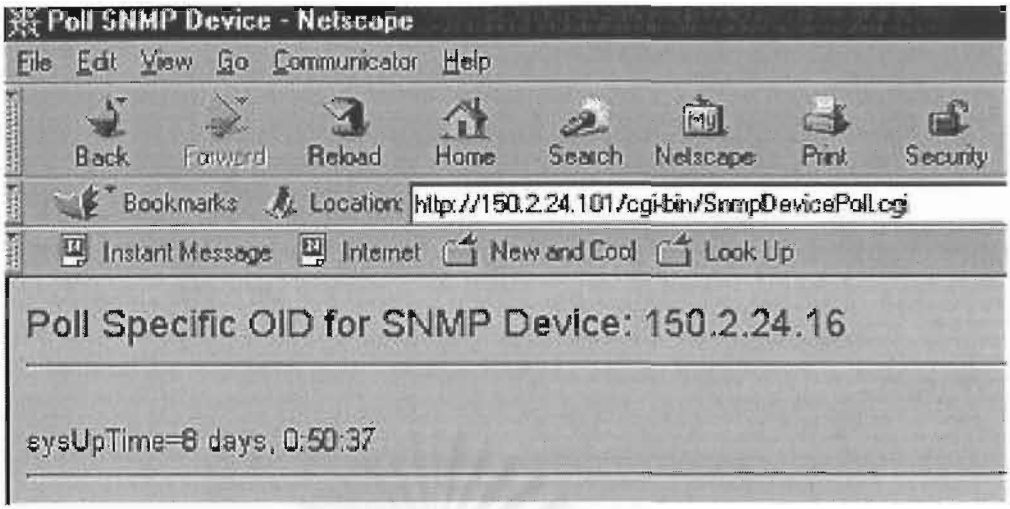


รูปที่ 5.10 แสดงหน้าจอสอบถามตามรายชื่อออบเจกต์

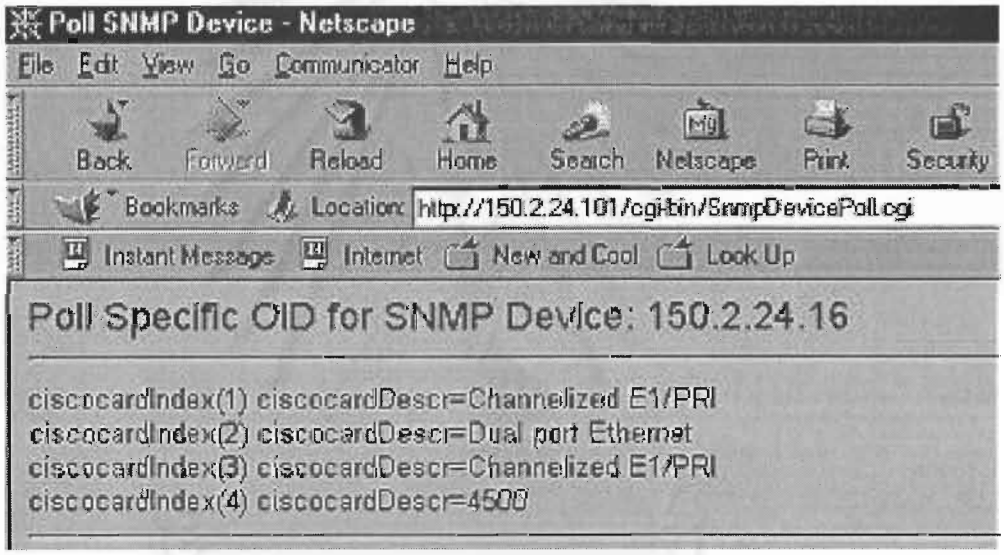
5.5.2 ผลการทดสอบ

โปรแกรมแสดงค่าออบเจกต์ ของอุปกรณ์จัดเส้นทางซิสโก้ รุ่น 4500 ดังแสดงในรูปที่ 5.11 และ รูปที่ 5.12

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.11 แสดงตัวอย่างการรายงานค่าออบเจกต์ sysUpTime



รูปที่ 5.12 แสดงตัวอย่างการรายงานค่าออบเจกต์ cardDescr

5.6 การทดสอบการรายงานข้อมูลในรูปของแผนภูมิ

เป็นการทดสอบโปรแกรมในการรายงานข้อมูลทางสถิติเกี่ยวกับอุปกรณ์ ซึ่งได้รวบรวมและบันทึกไว้ในแฟ้มข้อมูลต่างๆ เพื่อนำเสนอในรูปของแผนภูมิ

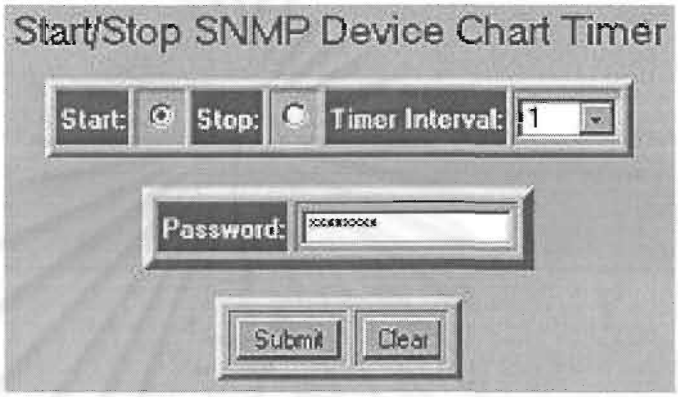
5.6.1 ขั้นตอนในการทดสอบ

1. กำหนดหมายเลขไอพีของอุปกรณ์ที่ต้องการรวบรวมข้อมูล ทดสอบโดยใช้ค่า "150.2.24.13" ซึ่งเป็นหมายเลขไอพีของ อีเทอร์เน็ตสวิช ของ ทรีคอม รุ่น LinkSwitch 1000
2. สร้างแฟ้มข้อมูลรายชื่อแผนภูมิ ภายในแฟ้มข้อมูลประกอบไปด้วยหมายเลขไอพี และค่าคอมมิวนิตี้สตริง ของอุปกรณ์ ดังนี้

150.2.24.13, LinkSwitch1000 on 24th Floor, public

รูปที่ 5.13 แสดงตัวอย่างแฟ้มข้อมูลรายชื่อแผนภูมิ

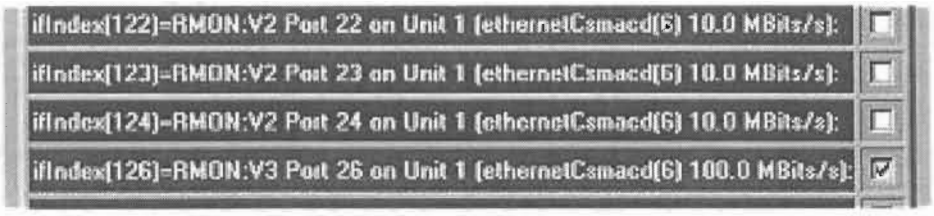
3. จากหน้าจอหลัก เลือกรายการ รวบรวมข้อมูลสร้างแผนภูมิจากอุปกรณ์ (รูปที่ 5.14) โดยกำหนดให้ติดต่อกับอุปกรณ์ทุก 1 นาที



รูปที่ 5.14 แสดงหน้าจอเริ่มงานรวบรวมข้อมูลสร้างแผนภูมิ

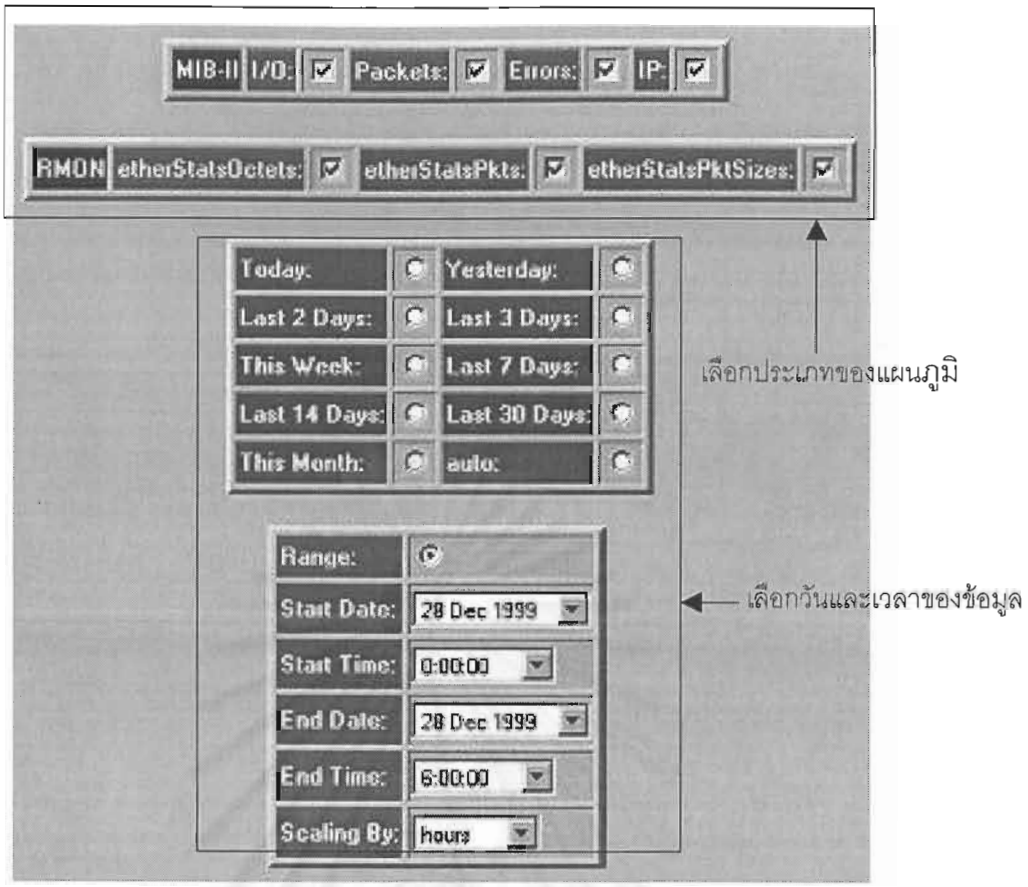
4. หลังจากโปรแกรมได้รวบรวมข้อมูลจากอุปกรณ์เป็นเวลา 6 ชั่วโมง จากหน้าจอหลัก เลือกรายการ แสดงแผนภูมิ และกำหนดข้อคำสั่ง ประกอบด้วย

1) เลือกอินเตอร์เฟสที่ต้องการ ทดสอบโดยการเลือกอินเตอร์เฟสหมายเลข 126 ดังแสดงในรูปที่ 5.15



รูปที่ 5.15 แสดงตัวอย่างหน้าจอเลือกอินเตอร์เฟสเพื่อแสดงแผนภูมิ

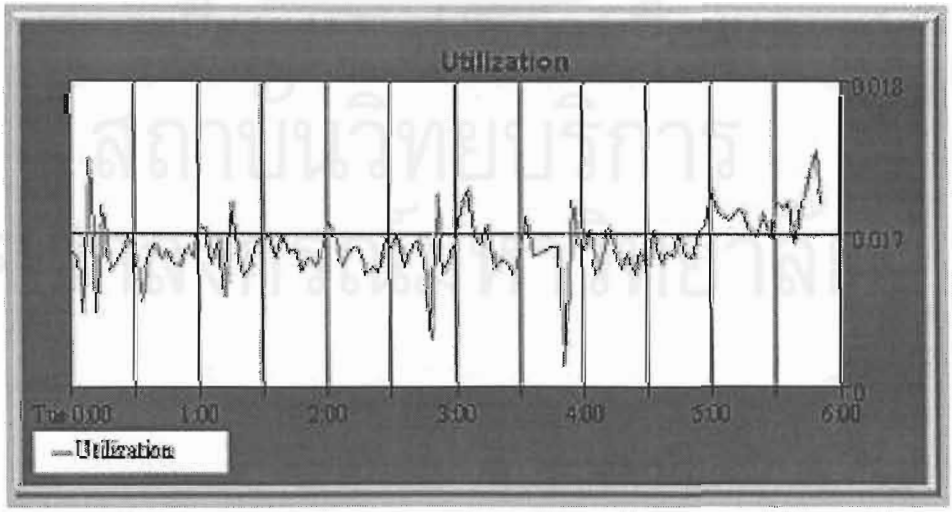
2) เลือกวัน เวลา และเลือกประเภทของแผนภูมิที่ต้องการ ดังแสดงในรูปที่ 5.16



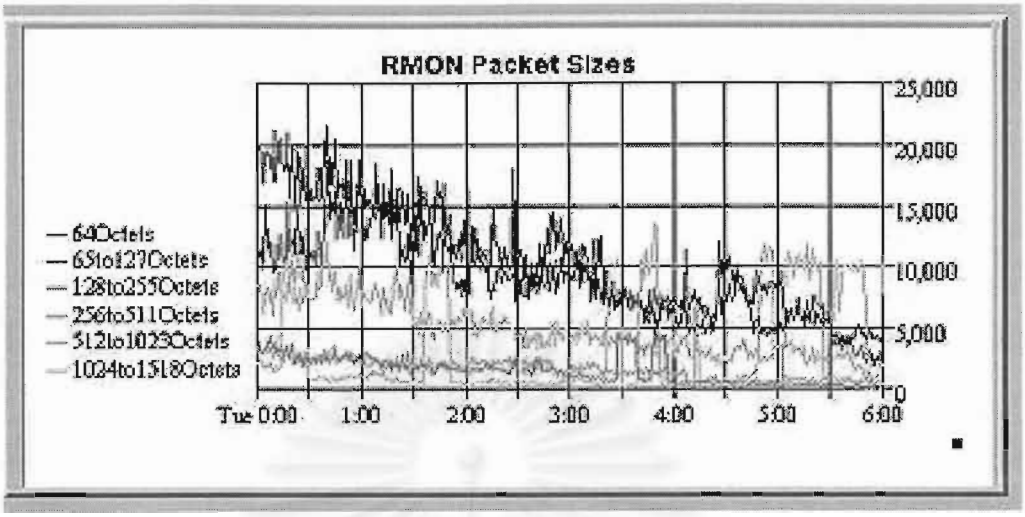
รูปที่ 5.16 แสดงตัวอย่างหน้าจอเลือกประเภทของแผนภูมิ

5.6.2 ผลการทดสอบ

โปรแกรมสามารถแสดงข้อมูลที่รวบรวมได้จากอุปกรณ์ในรูปของแผนภูมิ ดังตัวอย่างแสดงในรูปที่ 5.17 และ รูปที่ 5.18



รูปที่ 5.17 แสดงตัวอย่างของปริมาณการใช้บริการ (Utilization) ของอินเตอร์เฟส



รูปที่ 5.18 แสดงตัวอย่างของแพ็กเก็ตขนาดต่างๆที่ได้ผ่านเข้าออกจากอินเตอร์เฟส

5.7 การทดสอบการรายงานแทร็ปเมสเสจ

เป็นการทดสอบโปรแกรมในการรับรายงานเกี่ยวกับเหตุการณ์หรือสถานะของอุปกรณ์ ที่ได้ส่งมาในรูปของแทร็ปเมสเสจ ตามข้อกำหนดของโพรโทคอลเอสเอ็นเอ็มพี

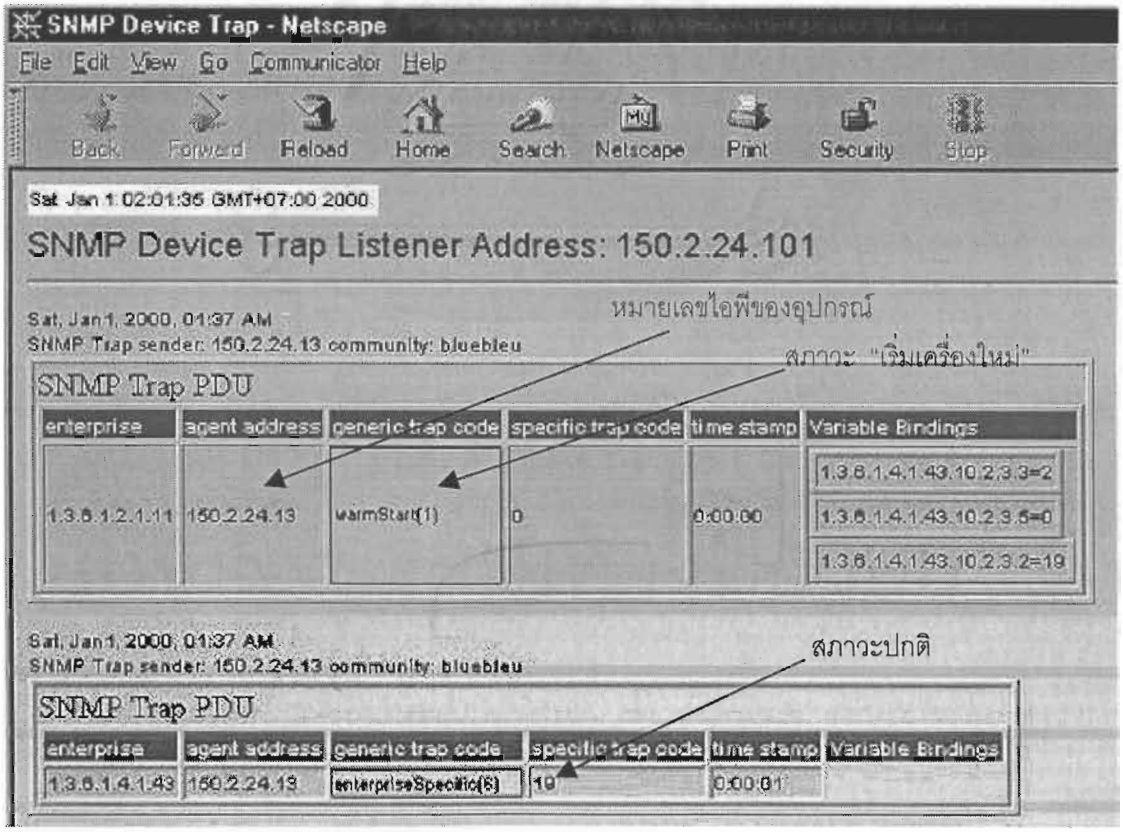
5.7.1 ขั้นตอนในการทดสอบ

1. กำหนดให้ อีเทอร์เน็ตสวิช ของ ทรีคอม รุ่น LinkSwitch 1000 เป็นอุปกรณ์สำหรับใช้ทดสอบการรายงานแทร็ปเมสเสจ
2. จัดเตรียมให้อีเทอร์เน็ตสวิชส่งแทร็ปเมสเสจมาที่โปรแกรมจัดการอุปกรณ์ฯ โดยกำหนดที่อยู่ของตัวรับแทร็ปเมสเสจ (Trap Listener Address) เป็นหมายเลขไอพีของเครื่องให้บริการ คือ 150.2.24.101
3. จากหน้าจอหลัก เลือกรายการ รับแทร็ปเมสเสจ (รูปที่ 5.19)
4. กำหนดให้อีเทอร์เน็ตสวิช ทำการ "ปลุกเครื่อง" (Reboot) อีกครั้ง

รูปที่ 5.19 แสดงหน้าจอเริ่มการทำงานรับแทร็ปเมสเสจ

5.7.2 ผลการทดสอบ

โปรแกรม สามารถรับแทร็ปเมสเสจ จากอีเทอร์เน็ตสวิช ของ ทรีคอม รุ่น LinkSwitch 1000 โดยเมสเสจแรกรายงานเหตุการณ์ เรียกว่า "เริ่มเครื่องใหม่" (Warm Start) และเมสเสจที่ 2 แจ้งว่าอุปกรณ์ได้กลับเข้าสู่สภาวะปกติ ดังแสดงในรูปที่ 5.20



รูปที่ 5.20 แสดงการรายงานแทร็ปเมสเสจ ที่ได้รับจากอุปกรณ์

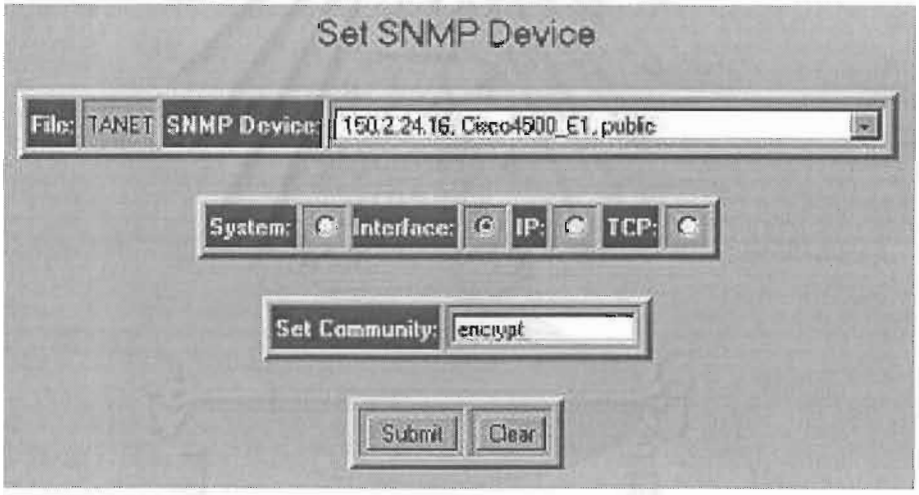
5.8 การทดสอบการแก้ไขอุปกรณ์

เป็นการทดสอบโปรแกรมในการแก้ไขค่าออบเจกต์ต่างๆ ภายใต้มิบ-ทู

5.8.1 ขั้นตอนในการทดสอบ

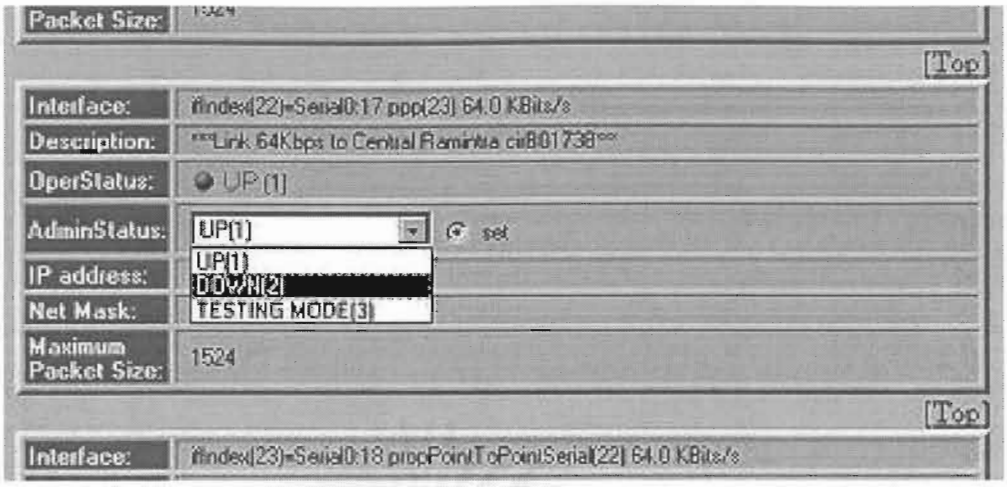
1. กำหนดให้ อุปกรณ์จัดเส้นทางของซิสโก้ รุ่น 4500 เป็นอุปกรณ์ที่ใช้ทดสอบการแก้ไขค่าออบเจกต์ต่างๆ
2. จากหน้าจอหลัก เลือกรายการแก้ไขอุปกรณ์
3. กำหนดข้อคำสั่ง (รูปที่ 5.21) ประกอบด้วย

    1) กำหนดหมายเลขไอพีของอุปกรณ์ที่ต้องการแก้ไข ทดสอบโดยการเลือกอุปกรณ์จัดเส้นทางของซิสโก้ รุ่น 4500
    2) เลือกแก้ไขออบเจกต์ ของกลุ่มอินเตอร์เฟส
    3) กำหนดค่าคอมมิวนิตี้สตริง ทดสอบโดยใช้ค่า “encrypt”



รูปที่ 5.21 แสดงหน้าจอแก้ไขอุปกรณ์

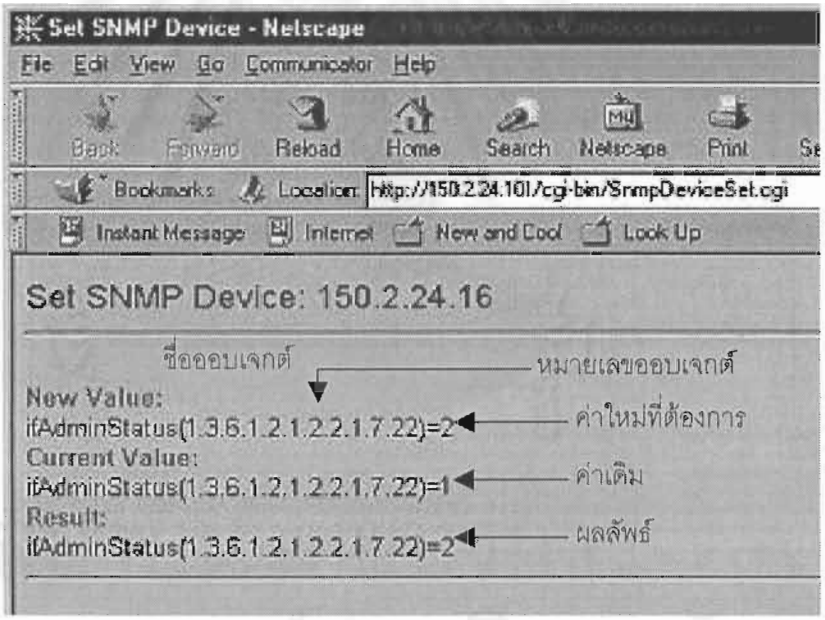
4. จากหน้าจอแก้ไขอุปกรณ์ (รูปที่ 5.22) เลือกออบเจกต์ที่ต้องการแก้ไข ทดสอบโดยการเปลี่ยนสถานะของ อินเตอร์เฟสที่ 22 จากสถานะ “UP” เป็น สถานะ “DOWN” ซึ่งเป็นการเปลี่ยนค่าของออบเจกต์ ifAdminStatus จาก “1” เป็น “2”



รูปที่ 5.22 แสดงตัวอย่างหน้าจอแก้ไขออบเจกต์ ifAdminStatus ของกลุ่มอินเตอร์เฟส

5.8.2 ผลการทดสอบ

ผลลัพธ์ของการแก้ไขอุปกรณ์ โดยการเปลี่ยนสถานะของอินเตอร์เฟสที่ 22 ด้วยการเปลี่ยนค่าของออบเจกต์ ifAdminStatus จาก "1" เป็น "2" สามารถแสดงในรูปที่ 5.23



รูปที่ 5.23 แสดงผลการทดสอบแก้ไขค่าออบเจกต์

5.9 สรุปผลการทดสอบ

จากการทดสอบโปรแกรมจัดการอุปกรณ์ฯ กับอุปกรณ์ทางเครือข่าย ที่ได้ระบุไว้ในขอบเขตของการทดสอบ สามารถสรุปผลการทดสอบเป็นไปตามตารางที่ 5.1 ถึง ตารางที่ 5.5

ลำดับที่	รายการทดสอบ	ผลการทดสอบ
1	ค้นหาอุปกรณ์	✓
2	สอบถามตามรายชื่อแฟ้มข้อมูล	✓
3	สอบถามตามหมายเลขไอพี	✓
4	สอบถามตามรายชื่อออบเจกต์	✓
5	รายงานผลในรูปแผนภูมิ	✓
6	รายงานแทร็ปเมสเสจ	✓
7	แก้ไขอุปกรณ์	✓

ตารางที่ 5.1 แสดงผลการทดสอบการให้บริการประเภทต่างๆโปรแกรม

✓ สามารถทำงานได้

ลำดับที่	อุปกรณ์	ผลการทดสอบ
1	อุปกรณ์จัดเส้นทางของซิสโก้ รุ่น 4500	✓
2	อุปกรณ์จัดเส้นทาง ของ เบย์เน็ตเวิร์ก รุ่น BLN	✓
3	เอทีเอ็มสวิช ของเบย์เน็ตเวิร์ก รุ่น Centillion 100	✓
4	อีเทอร์เน็ตสวิช ของ ทรีคอม รุ่น LinkSwitch 1000	✓
5	อีเทอร์เน็ตสวิช ของฟอร์ซิสเต็ม รุ่น ES 3850	✓
6	เอทีเอ็มสวิช ของฟอร์ซิสเต็ม รุ่น ASX 200 BX	✓
7	เครื่องให้บริการ บนระบบปฏิบัติการไมโครซอฟต์ วินโดวส์เอ็นที เวอร์ชั่น 4.0 (ไม่ใช่ตัวเดียวกันกับเครื่องให้บริการของโปรแกรมจัดการอุปกรณ์ฯ)	✓

ตารางที่ 5.2 แสดงผลการทดสอบโปรแกรมในการรายงานค่ามิบ-ทู ของอุปกรณ์

✓ โปรแกรมสามารถรายงานค่ามิบ-ทูของอุปกรณ์ได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ลำดับที่	อุปกรณ์	ผลการทดสอบ	หมายเหตุ
1	อุปกรณ์จัดเส้นทางของซิสโก้ รุ่น 4500	-	อุปกรณ์ไม่มีฟังก์ชันนี้
2	อุปกรณ์จัดเส้นทาง ของ เบย์เน็ตเวิร์กรุ่น BLN	-	อุปกรณ์ไม่มีฟังก์ชันนี้
3	เอทีเอ็มสวิช ของเบย์เน็ตเวิร์ก รุ่น Centillion 100	-	อุปกรณ์ไม่มีฟังก์ชันนี้
4	อีเทอร์เน็ตสวิช ของ ทรีคอม รุ่น LinkSwitch 1000	✓	-
5	อีเทอร์เน็ตสวิช ของฟอร์ซิสเต็ม รุ่น ES 3850	-	อุปกรณ์ไม่มีฟังก์ชันนี้
6	เอทีเอ็มสวิช ของฟอร์ซิสเต็ม รุ่น ASX 200 BX	-	อุปกรณ์ไม่มีฟังก์ชันนี้
7	เครื่องให้บริการ บนระบบปฏิบัติการไมโครซอฟต์ วินโดวส์เอ็นที เวอร์ชั่น 4.0 (ไม่ใช่ตัวเดียวกันกับเครื่องให้บริการของโปรแกรมจัดการอุปกรณ์ฯ)	-	อุปกรณ์ไม่มีฟังก์ชันนี้

ตารางที่ 5.3 แสดงผลการทดสอบโปรแกรมในการรายงานค่าอามอนมิบ ของอุปกรณ์

✓ โปรแกรมสามารถรายงานค่าอามอนมิบของอุปกรณ์ได้

ลำดับที่	อุปกรณ์	ผลการทดสอบ	หมายเหตุ
1	อุปกรณ์จัดเส้นทางซิสโก้ รุ่น 4500	✓	กลุ่มโพรดักส์ กลุ่มอินเตอร์เฟส และกลุ่มแชสซี ของซิสโก้
2	อุปกรณ์จัดเส้นทางเบย์เน็ตเวิร์ก รุ่น BLN	✓	กลุ่มฮาร์ดแวร์ ของเบย์เน็ตเวิร์ก
3	เอทีเอ็มสวิชของเบย์เน็ตเวิร์ก รุ่น Centillion 100	✓	กลุ่มฮาร์ดแวร์ ของเบย์เน็ตเวิร์ก
4	อีเทอร์เน็ตสวิชของทรีคอม รุ่น LinkSwitch 1000	✓	กลุ่มลิ้งค์สวิช ของทรีคอม

ตารางที่ 5.4 แสดงผลการทดสอบโปรแกรมในการรายงานค่าไพร์เวทมิบ ของอุปกรณ์

✓ โปรแกรมสามารถรายงานค่าไพร์เวทมิบของอุปกรณ์ได้

ลำดับที่	อุปกรณ์	ผลการทดสอบ
1	อุปกรณ์จัดเส้นทางของซิสโก้ รุ่น 4500	✓
2	อีเทอร์เน็ตสวิช ของ ทรีคอม รุ่น LinkSwitch 1000	✓
3	เครื่องให้บริการ บนระบบปฏิบัติการไมโครซอฟต์ วินโดวส์เอ็นที เวอร์ชั่น 4.0 (ไม่ใช่ตัวเดียวกันกับเครื่องให้บริการของโปรแกรมจัดการอุปกรณ์ฯ)	✓

ตารางที่ 5.5 แสดงผลการทดสอบโปรแกรมในการแก้ไขออบเจกต์ภายใต้มิบ-ทู ของอุปกรณ์

✓ โปรแกรมสามารถแก้ไขค่ามิบ-ทูของอุปกรณ์ได้

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

โปรแกรมจัดการอุปกรณ์ ฯ สามารถนำมาใช้งานเพื่อจัดการอุปกรณ์ด้วยโปรแกรมค้นผ่านเว็บเรียกว่าบราวเซอร์บนเครื่องขอรับบริการ ตามข้อกำหนดของโพรโทคอล เอสเอ็นเอ็มพี ตามอาร์เอฟซี 1157 มิบ-ทู (MIB-II) ตามอาร์เอฟซี 1213 อามอนมิบ (RMON MIB) ตามอาร์เอฟซี 1757 และ ไพร์เวทมิบของอุปกรณ์ที่ได้ถูกเลือกใช้ในงานวิจัย โดยมีเครื่องให้บริการ ทำหน้าที่ดังนี้

1. ติดต่อกับอุปกรณ์ ด้วยโอเปอเรชั่นเก็ต และเก็ต-เน็กส์
2. รับข้อมูลจากอุปกรณ์ ด้วยโอเปอเรชั่นเก็ต-เรสปอนส์ และ โอเปอเรชั่นแทร็ป
3. แก้ไขข้อมูลในอุปกรณ์ ด้วยโอเปอเรชั่นเซ็ต
4. เป็นตัวให้บริการเว็บ บนระบบปฏิบัติการวินโดวส์เอ็นที เพื่อรองรับการจัดการอุปกรณ์ด้วยบราวเซอร์จากเครื่องขอรับบริการ

การวิจัยได้พบปัญหาและข้อจำกัดที่เกิดขึ้นกับการทำงานของโปรแกรมจัดการอุปกรณ์ฯ ดังต่อไปนี้

1. โปรแกรมจัดการอุปกรณ์ ฯ ใช้เวลาค่อนข้างนานในการติดต่อ และ รวบรวมข้อมูลจากอุปกรณ์ เช่น เราท์เตอร์ของซิสโก้ รุ่น 4500 ซึ่งมีจำนวนอินเตอร์เฟส (Channelized E1 Interface) อยู่ถึง 66 อินเตอร์เฟส ทำให้ต้องใช้เวลาประมาณ 1 นาที
2. ปัญหาเรื่องความสามารถในการรองรับโพรโทคอล เอสเอ็นเอ็มพี ของอุปกรณ์ ในการวิจัยนี้ มีเพียงอุปกรณ์ชนิดเดียวคืออีเทอร์เน็ตสวิช ของทรีคอม รุ่น LinkSwitch 1000 ที่ได้ผลิตมาให้สามารถทำงานรองรับอามอนมิบได้ ดังนั้นงานวิจัยนี้ได้ให้แนวทางในการเปรียบเทียบ และเลือกใช้อุปกรณ์ คือ โปรแกรมจัดการระบบเครือข่ายมีความสามารถเพียงใดก็ไม่ได้ประโยชน์สูงสุด ถ้าอุปกรณ์ไม่ได้ถูกพัฒนาขึ้นมาให้สามารถรองรับข้อกำหนดของโพรโทคอลเอสเอ็นเอ็มพีได้เท่าเทียมกันทั้งหมด
3. ปัญหาเรื่องการแสดงผลในรูปของแผนภูมิ โดยเครื่องขอรับบริการ อาจจะหยุดทำงานเมื่อต้องแสดงแผนภูมิของทุกอินเตอร์เฟสภายในหน้าเว็บเพจเดียวกัน เพราะต้องบรรจุจาวาแอพเพล็ท (Java Applet) เป็นจำนวนมากในเวลาเดียวกัน

4. ในการกำหนดระยะเวลาในการรวบรวมข้อมูลจากอุปกรณ์ สำหรับบันทึกไว้สร้างแผนภูมิ ต้องคำนึงถึงอุปกรณ์ที่มีจำนวนอินเตอร์เฟส อยู่เป็นจำนวนมาก เช่น ต้องใช้เวลาไม่น้อยกว่า 1 นาทีในการรวบรวมข้อมูลจาก อุปกรณ์จัดเส้นทางของซิสโก้ รุ่น 4500 ที่ใช้ในการทดสอบ ซึ่งมีจำนวนอินเตอร์เฟสอยู่ถึง 66 อินเตอร์เฟส ดังนั้นจึงจำเป็นต้องกำหนดเวลาให้โปรแกรมติดต่อกับอุปกรณ์ ทุก 2 นาที หรือนานกว่านั้น

6.2 ข้อเสนอแนะ

จากงานวิจัยนี้เรื่องที่น่าสนใจที่สามารถทำต่อได้คือการจัดการอุปกรณ์ประเภทที่ไม่สามารถรองรับโพรโทคอลเอสเอ็นเอ็มพี โดยการพัฒนาส่วนเพิ่มเติมเพื่อจัดเตรียมเครื่องให้บริการเป็นพร็อกซี่เซิร์ฟเวอร์ (Proxy Server) สำหรับทำหน้าที่รายงานสถานะของอุปกรณ์ และรับส่งข้อมูลภายใต้โพรโทคอลเอสเอ็นเอ็มพี แทนอุปกรณ์เหล่านี้

จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Allen Leinwand and Karen Fang. Network Management: A Practical Perspective. Addison-Wesley, 1993.
2. Uyless Black. Network Management Standards: SNMP, CMIP, TMN, MIBs, and Object Libraries. McGraw-Hill, 1995.
3. Douglas W. Stevenson. Network Management: What it is and what it isn't, (URL http://smurfland.cit.buffalo.edu/NetMan/Doc/Dstevenson/).
4. Simple Network Management Protocol (SNMP). Cisco Systems Inc., (URL http://www.cisco.com/warp/public/535/3.html).
5. William Stallings. SNMP SNMPv2 and RMON Practical Network Management.. Addison-Wesley, 1996.
6. Phil County. Introduction to SNMP: The Simple Network Management Protocol. AARNET Networkshop, La Tropbe University, December 1992.
7. Jeffrey D. Case, Mark S. Fedor, Martin L. Schoffstall, and James R. Davin. A Simple Network Management Protocol (SNMP), RFC 1157 (previously RFC 1098), Network Working Group, 1990
8. K. McCloghrie and Marshall T. Rose. Structure and Identification of Management Information for TCP/IP-based Internets, RFC 1155 (previously RFC 1065), Network Working Group, 1990
9. K. McCloghrie and Marshall T. Rose. Management Information Base for Network Management of TCP/IP-based Internets: MIB-II, RFC 1213 (previously RFC 1158), Network Working Group, 1991
10. S. Waldbusser. Remote Network Monitoring Management Information Base, RFC 1757 (previously RFC 1271), Network Working Group, 1995
11. Perl for Win32, ActiveState Tool Corp., (URL http://www.ActiveState.com).
12. Simon Leinen. SNMP Support for Perl 5, (URL ftp://ftp.switch.ch/software/sources/network/snmp/perl/).
13. Lincoln Stein. CGI.pm – A Perl5 CGI Library, (URL http://stein.cshl.org/WWW/software/CGI/).

14. JavaChart, Visual Engineering, Inc., (URL http://www.ve.com).

15. Tony Zhang. David Till. Teach Yourself Perl 5 FOR Windows NT in 21 days, SAMS Publishing, 1997.

จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

นาย ชัยวัฒน์ ฉัตรศิริชัย สำเร็จการศึกษาวิทยาศาสตรบัณฑิต จาก Simon Fraser University ประเทศแคนาดา เมื่อ พ.ศ. 2531 และได้เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2539

จุฬาลงกรณ์มหาวิทยาลัย