สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาโปรแกรมสร้างแผนที่เครือข่าย

A DEVELOPMENT OF NETWORK MAP PROGRAM

T039660

วิมลรัตน์ อริยชจร
สีตะลา นุ้ยขาว
สุนิดา โอภาคเจริญ

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

ปก.
QA76.9
.D5
V53
เลขหมู่.....
เลขทะเบียน 39660
วัน, เดือน, ปี 19 มิ.ย. 2544
.b.....
.i.....

A DEVELOPMENT OF NETWORK MAP PROGRAM

WIMONRAT ARIYAKAJORN

SRITALA NUYKHOA

SUNIDA OPARKCHAROEN

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCES

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2000

หัวข้อปัญหาพิเศษ	การพัฒนาโปรแกรมสร้างแผนที่เครือข่าย	
	A DEVELOPMENT OF NETWORK MAP PROGRAM	
ชื่อนักศึกษา	นางสาววิมลรัตน์ อริยขจร	40056080
	นางสาวสีตะลา นุ้ยขาว	40056094
	นางสาวสุนิดา โอภาคเจริญ	40056100
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์	
	อาจารย์ นรฤทธิ์ สุนทรศารทูล	

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2543

คณะกรรมการสอบ	ลายมือชื่อ
ประธานกรรมการ ดร.นันทิกา เบญจเทพานันท์	
กรรมการ รองศาสตราจารย์อุบลวรรณา เงินวิจิตร	
กรรมการและอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์	
กรรมการและอาจารย์ที่ปรึกษา อาจารย์นรฤทธิ์ สุนทรศารทูล	

(ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปัญหาพิเศษเรื่อง	การพัฒนาโปรแกรมสร้างแผนที่เครือข่าย
ชื่อนักศึกษา	นางสาววิมลรัตน์ อริยขจร 40056080
	นางสาวสีตะลา นุ้ยขาว 40056094
	นางสาวสุนิดา โอภาคเจริญ 40056100
ปริญญา	วิทยาศาสตรบัณฑิต
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2543
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์
	อาจารย์ นรฤทธิ์ สุนทรศารทูล

บทคัดย่อ

ปัญหาพิเศษเรื่องการพัฒนาโปรแกรมสร้างแผนที่เครือข่ายนี้ เป็นการพัฒนาโปรแกรมเพื่อทำการสร้างแผนที่เครือข่ายของอุปกรณ์ภายในเครือข่ายที่สนับสนุนการทำงานของโปรโตคอล SNMP โดยการพัฒนาโปรแกรมจะแบ่งเป็น 3 ส่วน คือการดึงค่าข้อมูลจากอุปกรณ์ในเครือข่าย, การเก็บค่าข้อมูลลงฐานข้อมูล และการสร้างแผนที่เครือข่าย ซึ่งจะมีประโยชน์มากในการจัดการเครือข่าย เพราะทำให้เจ้าหน้าที่ผู้ดูแลเครือข่ายสามารถมองเห็นและเข้าใจความสัมพันธ์ระหว่างองค์ประกอบต่างๆบนเครือข่ายได้ดียิ่งขึ้น

ปัญหาพิเศษนี้พัฒนาโดยใช้ Delphi version 5 และเก็บข้อมูลโดยใช้ Database Desktop

Special Project Tittle	A Development Of Network Map Program	
Students	Miss Wimonrat Ariyakajorn	40056080
	Miss Sritala Nuykhoa	40056094
	Miss Sunida Oparkcharoen	40056100
Degree	Bachelor's Degree of Science	
Department	Mathematics and Computer Sciences, Faculty of Science	
Programme	Computer Sciences	
Academic Year	2000	
Special Project Advisor	Assistant Professor Praiboon Pantaragphong	
	Lecturer Norarit Suntornsaratoon	

ABSTRACT

Special problem is to develop Network Map to support the function of SNMP Protocol. It is divided into 3 parts as following : Poll information from Network device, Collect database and Build Network Map

Network Map is very useful because Network Administrator can see and better understand the relationship among components in the Network

Delphi version 5.0 and Database Desktop are created to solve this special problem.

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องการพัฒนาโปรแกรมสร้างแผนที่เครือข่าย สามารถสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณบุคคลต่างๆ ดังต่อไปนี้

- ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์ ที่ให้คำปรึกษา คอยบอกแนวทางในการจัดทำปัญหาพิเศษครั้งนี้ ตลอดจนแนวคิดต่างๆ
- อาจารย์นรฤทธิ์ สุนทรศารทูล ที่ให้คำปรึกษา ชี้แนะแนวทางการทำงาน รวมทั้งอำนวยในเรื่องสถานที่
- อาจารย์ทุกท่านที่ให้ความรู้ตลอดระยะเวลาการศึกษา
- พี่ๆที่สำนักวิจัยและบริการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยดูแลเป็นอย่างดีมาก(โดยเฉพาะพี่ธีรกิต)ช่วยแก้ปัญหาหลายๆอย่าง รวมทั้งให้ความรู้และคำแนะนำที่ดี
- เพื่อนๆที่ให้กำลังใจเสมอมา

และขอขอบพระคุณบุคคลสำคัญอีกสองท่านคือบิดา มารดา ที่คอยให้ทั้งกำลังใจ กำลังทรัพย์ และเลี้ยงดูเป็นอย่างดีเสมอ

คณะผู้จัดทำ

มีนาคม 2544

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VII
สารบัญรูป.....VIII
บทที่1 บทนำ.....1
1.1 ความสำคัญและที่มาของปัญหา.....1
1.2 วัตถุประสงค์ของการศึกษา.....1
1.3 ขอบเขตของการศึกษา.....1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....1
1.5 ขั้นตอนในการดำเนินงาน.....2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....3
2.1 โปรโตคอล TCP/IP.....3
2.2 การอ้างอิงอุปกรณ์ในเครือข่าย.....3
2.3 IP ADDRESS.....4
2.3.1 การจัดลำดับชั้นของเครือข่าย (Network class).....5
2.3.2 Subnet.....7
2.3.3 Class C Subnet.....10
2.3.4 หมายเลขอ้างอิงเครือข่ายและ Broadcast Address.....11
2.3.5 Multicast Address.....12
2.3.6 IP Address ที่ออกสู่อินเตอร์เน็ตไม่ได้.....12
2.4 Simple Network Management Protocol (SNMP).....13
2.4.1 คำสั่ง SNMP.....15
2.4.2 Limitation of SNMP.....16
2.5 SNMP Management Information Base.....16
2.5.1 Structure of management information.....17
2.5.2 MIB Structure.....17
2.5.2.1 Object Syntax.....18

2.5.2.2 Application Types.....19
2.5.2.3 Defining Object.....20
2.5.2.4 Defining Tables.....20
2.5.2.5 Encoding.....21
2.5.3 Management Information Base (MIB) – II.....21
2.5.4 SNMP Formats.....23
2.5.5 SNMP PDUs.....25
บทที่ 3 การออกแบบโปรแกรม.....26
3.1 หลักการและการออกแบบระบบการทำงานของโปรแกรม.....26
3.1.1 การติดต่อระหว่าง Management Station กับ Agent.....26
3.1.1.1 การดึงค่าข้อมูลจาก Agent ในระบบเครือข่าย.....27
3.1.1.2 คำสั่งที่ Management Station ใช้ในการดึงค่าข้อมูลจาก Agent.....28
3.1.2 การเก็บค่าลงฐานข้อมูล.....28
3.1.3 Management Station แสดงผลที่ Interface.....28
3.2 การออกแบบฐานข้อมูล.....28
3.3 การออกแบบส่วนที่ติดต่อกับผู้ใช้.....32
3.3.1 จอภาพแสดงการ Poll โดยระบุ IP Address.....33
3.3.2 จอภาพแสดงการ Poll โดยกำหนด Network Address และ Subnet Address.....35
3.3.3 จอภาพแสดงข้อมูลในแต่ละ Port.....37
3.3.4 แผนภูมิแท่งแสดงจำนวนข้อมูลที่ผ่านแต่ละ Interface ของ Agent.....38
3.3.5 หน้าจอแสดงแผนที่ของเครือข่าย (แบบที่1).....39
3.3.6 หน้าจอแสดงแผนที่ของเครือข่าย (แบบที่2).....40
3.3.7 หน้าจอแสดงแผนที่ของเครือข่าย (แบบที่3).....41
บทที่ 4 การพัฒนาและทดสอบโปรแกรม.....42
4.1 ซอฟท์แวร์และฮาร์ดแวร์ที่ใช้ในการพัฒนาโปรแกรม.....42
4.2 สภาพแวดล้อมของการใช้โปรแกรม.....42
4.3 การทดสอบและขั้นตอนการดำเนินการพัฒนาโปรแกรม.....42
4.3.1 ระบุ IP Address ของอุปกรณ์โดยตรง.....43
4.3.1.1 ดึงข้อมูลจากอุปกรณ์เครือข่าย.....43

4.3.1.2 การเก็บข้อมูลลงฐานข้อมูล.....43
4.3.1.3 การแสดงข้อมูลทางจอภาพเพื่อแสดงแผนที่เครือข่าย....44
4.3.2 ระบุเป็นหมายเลขของเครือข่าย.....48
4.3.2.1 ดึงข้อมูลจากอุปกรณ์เครือข่าย.....48
4.3.2.2 การเก็บข้อมูลลงฐานข้อมูล.....51
4.3.2.3 การแสดงรูปแผนที่เครือข่าย.....53
บทที่ 5 สรุปผลการดำเนินงานปัญหาพิเศษ.....59
5.1 ผลการทดลองปัญหาพิเศษ.....59
5.2 สรุปผลปัญหาพิเศษ.....60
5.3 สรุปผลการดำเนินงานปัญหาพิเศษ.....61
5.3.1 ขั้นตอนการศึกษาและรวบรวมข้อมูล.....61
5.3.2 ขั้นตอนการออกแบบโครงสร้างของโปรแกรม.....61
5.3.3 ขั้นตอนการพัฒนาโปรแกรม.....61
5.4 ข้อจำกัดของปัญหาพิเศษ.....62
5.5 ข้อเสนอแนะและแนวทางในการแก้ปัญหา.....62
ภาคผนวก 63
บรรณานุกรม

สารบัญตาราง

ตารางที่ **หน้า**

2.1 network class....6
2.2 แสดงการแบ่ง Host address....9
2.3 แสดงการกำหนด Subnet mask....10
2.4 แสดง Subnet ทั้งหมดของคลาส C....10
2.5 ตารางตัวอย่าง multicast address....12
2.6 แสดงเครือข่ายที่เข้าสู่อินเตอร์เนตไม่ได้....13
2.7 SNMP Message Fields....24
3.1 แสดงรายละเอียดของ Agent.: SystemData.db....30
3.2 เก็บข้อมูลของเครือข่ายที่เชื่อมต่อ Router ของ Network Address : Router_table.db....30
3.3 แสดงชนิดของอุปกรณ์ในระบบเครือข่าย : Equipment_table.db....31
3.4 เก็บข้อมูลเข้าและออกของแต่ละ Interface ของ Agent : Dataport.db....31
4.1 ฐานข้อมูลแสดงรายละเอียดของ Agent(SyetemData.db) 43
4.2 ฐานข้อมูลรูปภาพอุปกรณ์ (Equipment.db)....43
4.3 ตัวอย่างฐานข้อมูลเก็บค่าข้อมูลเข้าและออกจาก Port....44
4.4 Router_table.db....44
4.5 ฐานข้อมูลแสดงรายละเอียดของ Agent (SystemData.db)....52
4.6 ฐานข้อมูลรูปภาพอุปกรณ์ (Equipment.db)....52
4.7 ตัวอย่างฐานข้อมูลเก็บค่าข้อมูลเข้าและออกจาก Port....52
4.8 Router_table.db....53

สารบัญรูป

รูปที่ **หน้า**

2.1 การแบ่งส่วนของหมายเลขเครือข่ายและหมายเลขเครื่องลูกข่ายในแต่ละ
ระดับชั้นของเครือข่ายในคลาส A , B และ C 6
2.2 แสดงเครือข่าย 3 เครือข่ายที่แต่ละเครือข่ายมี IP Addressเป็นของตัวเอง.....8
2.3 แสดงการแบ่ง Subnet ออกเป็นเครือข่ายย่อย โดยใช้ IP Address ตัวเดียว..... 9
2.4 แสดงวิธีรวบรวมข้อมูลของเน็ตเวิร์ค SNMP.....13
2.5 Configuration of SNMP.....14
2.6 Role of SNMP.....15
2.7 แสดง Structure of MIB.....18
2.8 ตัวอย่างตาราง MIB.....21
2.9 SNMP Formats.....24
3.1 ระบบการทำงานของโปรแกรม.....26
3.2 แสดงความสัมพันธ์ของตาราง.....29
3.3 แสดงการออกแบบหน้าจอสำหรับการ Poll โดยกำหนด IP Address.....33
3.4 แสดงการออกแบบหน้าจอสำหรับการ Poll โดยกำหนดหมายเลขเครือข่ายและ
Subnet mask.....35
3.5 แสดงการออกแบบหน้าจอสำหรับพิจารณาแต่ละ Interface ของ Agent.....37
3.6 แสดงการออกแบบรูปที่ได้จากการ Poll โดยใช้หมายเลขเครือข่าย.....39
3.7 แสดงการออกแบบรูปที่ได้จากความสัมพันธ์ระหว่าง Interface ของ Router
กับเครือข่ายที่เชื่อมต่อ.....40
3.8 แสดงการออกแบบภาพที่ได้จากการ Poll ใน 1 เครือข่าย.....41
4.1 แสดงหน้าจอ One Poll.....45
4.2 แสดงหน้าจอ Equipment.....46
4.3 แสดงหน้าจอกรณีที่ Agent ตัวนี้เคย Support SNMP.....47
4.4 แสดงจอภาพ Find Informaion.....48
4.5 แสดงจอภาพ Network Address Poll.....49
4.6 แสดงจอภาพ Poll Only Support.....51
4.7 แสดงจอภาพ Network Map.....53
4.8 แสดงจอภาพ Network Map (ต่อ).....54
4.9 แสดงจอภาพ Router.....55

4.10 แสดงหน้าจอ Port Data. 55
4.11 แสดงหน้าจอสรุปผลการไหลเข้าออกของข้อมูลของแต่ละ Port....57
4.12 แสดงหน้าจอแผนภูมิการไหลเข้าออกของข้อมูลแต่ละ Port....58
5.1 แสดงแผนที่ของเครือข่ายซึ่งมีหมายเลขของเครือข่ายเป็น 161.246.10....59
5.2 แสดงอุปกรณ์ในเครือข่ายที่สนับสนุน SNMP โปรโตคอลโดยมีหมายเลขเครือข่าย เป็น161.246.10.... 60
ก.1 แสดงการเข้าสู่การติดตั้ง SNMP Service ของ Window NT 63
ก.2 แสดงการเลือก TabSheet Service 64
ก.3 แสดงการเลือกติดตั้ง SNMP Service 64

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

เนื่องจากในปัจจุบันโปรแกรมทางด้านระบบเครือข่ายในประเทศไทยมีจำนวนน้อย ส่วนใหญ่จะเป็นโปรแกรมของต่างประเทศ การพัฒนาโปรแกรมด้านนี้ขึ้นมาจึงเป็นประโยชน์ในส่วนของการช่วยลดค่าใช้จ่ายและการสร้างแผนที่เครือข่าย ช่วยให้เจ้าหน้าที่ผู้ดูแลระบบเครือข่าย (Network Administrator) สามารถมองเห็นและเข้าใจความสัมพันธ์ระหว่างองค์ประกอบต่างๆบนเครือข่ายได้ดียิ่งขึ้น

ในการทำ Network Map จะใช้ SNMP (Simple Network Management Protocol) ในการตรวจสอบระบบ Network บนพื้นฐานโปรโตคอล TCP/IP ซึ่งโปรโตคอล TCP มีความสามารถที่จะกระทำการ Monitor ได้แต่มีความยุ่งยากในการโปรแกรมและการนำมาใช้ในการจัดการระบบเครือข่าย แต่ SNMP ง่ายในการใช้งาน ไม่ยุ่งยากซับซ้อน และมีข้อความคำสั่งสั้นๆ ง่ายๆ ไม่ต้องการทรัพยากรมากนัก ทำงานอยู่บนอุปกรณ์และสภาพแวดล้อมทั่วๆ ไปได้

1.2 วัตถุประสงค์ของการทำ

เป็นการพัฒนาโปรแกรมเพื่อดึงค่าข้อมูลจากอุปกรณ์ในเครือข่ายที่สนับสนุนการทำงานของโปรโตคอล SNMP มาจัดทำแผนที่เครือข่าย ให้เห็นถึงความสัมพันธ์และสถานภาพขององค์ประกอบต่างๆ ขณะระบบเครือข่ายกำลังปฏิบัติงานอยู่

1.3 ขอบเขตของปัญหา

1.3.1 ศึกษาและเข้าใจในทฤษฎีของการจัดการระบบเครือข่ายโดยใช้ SNMP Protocol

1.3.2 ทำการสืบค้นเครือข่าย (Network discovery) โดยใช้ SNMP และนำข้อมูลที่ได้มาสร้างเป็น Network Map

1.3.3 ทำการโปรแกรมด้วย Delphi 5 เพื่อดึงค่า Information ในการแสดงแผนที่เครือข่าย

1.3.4 โปรแกรมนี้จะทำงานบนระบบปฏิบัติการ Window NT Server เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้ศึกษาและเข้าใจในระบบเครือข่ายมากขึ้น

1.4.2 เรียนรู้เกี่ยวกับ SNMP Protocol

1.4.3 Network Map จะเป็นประโยชน์สำหรับงานจัดการเครือข่ายสำหรับผู้ดูแลระบบ

1.5 ขั้นตอนในการดำเนินงาน

1.5.1 ศึกษาเกี่ยวกับอุปกรณ์และการอ้างอิงอุปกรณ์ในเครือข่าย

1.5.2 ศึกษาการเชื่อมต่อของอุปกรณ์ในเครือข่าย

1.5.3 ศึกษาการจัดการระบบเครือข่ายโดยใช้ SNMP Protocol

1.5.4 ออกแบบแผนที่เครือข่าย และฐานข้อมูลที่ใช้ในการจัดเก็บ

1.5.5 เลือกตารางจากฐานข้อมูล Management Information Base (MIB) เพื่อนำมาใช้ในการดึงค่าข้อมูลจากอุปกรณ์ในเครือข่ายและนำมาสร้างแผนที่เครือข่าย

1.5.6 ทดสอบ code ของ tool ต่างๆ ที่น่าจะเป็นไปได้

1.5.7 ศึกษา Compiler ของ Borland Delphi 5.0 และ Library ของ SNMP ที่ใช้กับ Compiler Delphi

1.5.8 ทดสอบ code ในการส่งคำสั่งดึงค่าข้อมูลไปยังอุปกรณ์ในเครือข่ายโดยใช้ SNMP Protocol

1.5.9 ทำการสืบค้นระบบเครือข่าย

1.5.10 นำข้อมูลที่ได้เก็บลงฐานข้อมูล

1.5.11 จัดทำ Interface เกี่ยวกับการแสดงผลข้อมูล

1.5.12 นำข้อมูลจากฐานข้อมูลสร้างแผนที่เครือข่าย

1.5.13 วิเคราะห์ถึงความสัมพันธ์ขององค์ประกอบและการเชื่อมโยงระหว่างเครือข่าย

1.5.14 สรุปและจัดทำเอกสารการวิจัยการทำงาน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 โปรโตคอล TCP/IP

โปรโตคอล TCP/IP เป็นชื่อเรียกของชุดโปรโตคอลที่สำคัญ มีการใช้งานกันอย่างแพร่หลายตามการขยายตัวของอินเทอร์เน็ต/อินทราเน็ต ความจริงแล้วโปรโตคอล TCP/IP เป็นกลุ่มของโปรโตคอลหลายตัวที่ประกอบกันเป็นชุดให้ใช้งาน โดยมีคำเต็มว่า Transmission Control Protocol/Internet Protocol ซึ่งจากชื่อเต็มทำให้เราเห็นว่าอย่างน้อยก็มีโปรโตคอลประกอบกันทำงานร่วมกัน 2 โปรโตคอลคือ TCP และ IP

ตัวอย่างของกลุ่มโปรโตคอลในชุดของ TCP/IP ที่เราพบและใช้งานบ่อยๆ (ส่วนใหญ่เราจะไม่ได้ใช้งานตรงๆ แต่ใช้งานผ่านแอพพลิเคชันต่างๆ หรือใช้งานโดยทางอ้อม) เช่น Internet Protocol (IP) , Address Resolution Protocol (ARP) , Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP) , Transport Control Protocol (TCP) , Simple Mail Transfer Protocol (SMTP) และ Domain Name System (DNS) เป็นต้น

โปรโตคอลที่มีบทบาทสำคัญในการทำงานในเครือข่ายอินเตอร์เน็ต คือ Internet Protocol (โปรโตคอล IP) เนื่องจากเมื่อโปรโตคอลอื่นๆต้องการส่งผ่านข้อมูลข้ามเครือข่ายในอินเทอร์เน็ตนั้น จะต้องอาศัยการผนึกข้อมูล (Encapsulation) ไปกับโปรโตคอล IP ที่มีกลไกการระบุเส้นทาง (Route service) ผ่าน Gateway หรือ Router เพื่อนำข้อมูลไปยังเครือข่ายและเครื่องปลายทางที่ถูกต้อง เนื่องจากกลไกการระบุเส้นทางจะทำงานที่โปรโตคอล IP เท่านั้น และด้วยเหตุนี้เราจึงเรียก IP ว่าเป็นโปรโตคอลที่มีความสามารถระบุเส้นทางการส่งผ่านของข้อมูลได้ (Routable)

2.2 การอ้างอิงอุปกรณ์ในเครือข่าย

แนวความคิดหลักของระบบเครือข่ายคอมพิวเตอร์ก็คือ การเชื่อมโยงอุปกรณ์เข้าด้วยกัน ไม่ว่าจะเป็นเครื่องเซิร์ฟเวอร์ที่ให้บริการ (หรือบางทีเรียกว่า Host) และอุปกรณ์ในเครือข่ายอื่นๆ เช่น Router, เครื่องพิมพ์ เพื่อให้สามารถแชร์การใช้อุปกรณ์ร่วมกันได้ หรือสามารถส่งผ่านข้อมูลไปมาระหว่างกันได้ถูกต้อง เมื่อมีการเชื่อมต่อกันแล้วก็จำเป็นต้องมีการกำหนดหรือระบุเลขหมายของอุปกรณ์ทุกชิ้นทุกชนิดในเครือข่าย เพื่อให้อ้างอิงได้โดยไม่ซ้ำกัน เพราะถ้าซ้ำกันแล้วการรับส่งข้อมูลอาจจะไม่ถึงมือผู้รับปลายทางได้อย่างถูกต้อง เลขหมายดังกล่าวจะเรียกว่า แอดเดรส (Address) หรือเลขหมายประจำตัวที่มีข้อกำหนดเป็นมาตรฐาน ซึ่งในการใช้งานโปรโตคอล TCP/IP ที่เชื่อมโยงเครือข่ายอินเตอร์เน็ตนี้ เลขหมายที่ใช้อ้างอิงถึงกันจะใช้เป็นตัวเลขที่เรียกว่า IP address (Internet Protocol address) หรือ " หมายเลข IP"

2.3 IP ADDRESS

หมายเลข IP address ถูกกำหนดขึ้นมาให้เป็นหมายเลขอ้างอิงประจำตัวของอุปกรณ์ต่างๆที่เชื่อมต่อกันอยู่ในเครือข่ายอินเทอร์เน็ต โดยการกำหนดหมายเลข IP address ให้แต่ละเครื่องหรือแต่ละอุปกรณ์นี้จะต้องไม่ซ้ำกัน ซึ่งหมายเลข IP address นี้จะไม่ถูกผูกติดกับตัวฮาร์ดแวร์แต่อย่างใด ทั้งนี้เนื่องจากเป็นการกำหนดด้วยซอฟต์แวร์ แต่ต่างกับหมายเลข MAC address (Media Access Control address) ซึ่งเป็นหมายเลขประจำตัวของอุปกรณ์ที่ต่ออยู่ในเครือข่าย ค่า MAC address จะถูกกำหนดจากบริษัทผู้ผลิตอุปกรณ์ตั้งแต่เริ่มผลิต เช่น อุปกรณ์ Network Interface Card (NIC) จะมีค่า MAC address ที่ไม่ซ้ำกันและไม่สามารถแก้ไขได้ ค่า MAC address เป็นการระบุค่าอ้างอิงของอุปกรณ์ฮาร์ดแวร์ในระดับล่างสุด (Physical Layer) ของกลไกการรับส่งข้อมูลภายในเครือข่าย ถ้าจะใช้หมายเลข MAC address สำหรับระบุอ้างอิงกันในเครือข่ายแล้วจะเกิดปัญหามาก เมื่อมีการเปลี่ยนหรือย้ายเครื่องต้องทำการกำหนดระบบเครือข่ายใหม่ (Configuration) นอกจากนี้ยังจดจำได้ยากกว่า ตัวอย่างของหมายเลข MAC address คือ 08:0a:0e:12:b5:05 การที่หมายเลข IP address ถูกใช้อ้างอิงในการติดต่อกันด้วยโปรโตคอล TCP/IP เพราะการใช้หมายเลข IP address จะยืดหยุ่นและคล่องตัวกว่า

การทำงานของโปรโตคอล IP จำเป็นต้องอาศัยหมายเลข IP address นี้เพื่อระบุและอ้างอิงอุปกรณ์ต่างๆที่ต่ออยู่ในเครือข่าย ไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์, เมล์เซิร์ฟเวอร์, อุปกรณ์ Router ฯลฯ หมายเลข IP address จะเป็นค่าตัวเลขขนาด 32 บิต ถูกแบ่งออกเป็นส่วนละ 8 บิต รวมเป็น 4 ส่วน และคั่นแต่ละส่วนด้วยเครื่องหมายจุด (.) ดังนั้นค่าตัวเลขในแต่ละส่วนจะมีได้ตั้งแต่ 0 ถึง 255 (2^8) ตัวอย่างของ IP address เช่น 205.144.78.1 Loop back address เป็นแอดเดรสที่กำหนดขึ้นเพื่อใช้ในงานที่ต้องการให้ Process หนึ่งติดต่อกับ Process อื่นๆในเครื่องเดียวกันผ่าน IP address หรืออาจจะกล่าวโดยง่ายว่า Process ในเครื่องจะติดต่อกันระหว่าง Process เองได้โดยผ่านกลไก Loop back interface ซึ่งมีการกำหนดหมายเลข IP address พิเศษให้เป็น Loop back address ดังนี้

127.0.0.0 Loop back สำหรับเครือข่าย Class A

191.255.0.0 Loop back สำหรับเครือข่าย Class B

223.255.255.0 Loop back สำหรับเครือข่าย Class C

ดังนั้นถ้าเราพบค่า IP address เป็น 127.0.0.2 หรือ 191.255.0.1 หรือ 223.255.255.111 ให้คิดว่า เป็นการกำหนดค่า Loop back ในเครือข่าย class A, B, C ตามลำดับ

ค่า IP address ที่กำหนดเป็น Loop back นี้จะถูกผนึกหรือ Bind (กำหนดการอ้างอิงกับฮาร์ดแวร์) เข้ากับ Loop back network interface ของอุปกรณ์นั้นๆ เช่น ค่า IP address ที่ 127.0.0.1 สำหรับในระบบ Windows NT จะถือเป็นค่าที่ใช้เป็น Loop back ปกติที่ระบบรู้จักและกำหนดไว้ให้ใช้งาน ดังนั้นเมื่อมีการส่งผ่านข้อมูลไปที่หมายเลข IP 127.0.0.1 จะไม่มีการส่งข้อมูลออกไปที่เครือข่าย แต่จะย้อนกลับมาที่ IP ต้นทางโดยกลไกการ Loop back ของ Loop back driver นั่นเอง

ค่าของ IP address จะถูกกำหนดออกเป็น 2 ความหมายคือ ค่าของหมายเลขอุปกรณ์ในเครือข่าย (Host address) และค่าของหมายเลขเครือข่าย (Network address) ตัวอย่างเช่น มีเครื่องเซิร์ฟเวอร์เชื่อมต่ออยู่ในเครือข่าย 2 เครื่อง โดยแต่ละเครื่องมี IP address ประจำตัวคือ 205.144.78.2 และ 205.144.78.3 ตามลำดับ เครื่องทั้งสองมีค่าของหมายเลขเครือข่ายเหมือนกัน คือ 205.144.78 แสดงว่าเครื่องทั้งสองต่อเชื่อมอยู่ในเครือข่ายเดียวกัน บนสายสัญญาณที่เชื่อมโยงเส้นเดียวกันแต่มีหมายเลขประจำตัวเครื่องที่แตกต่างกันคือ 2 และ 3 ตามลำดับ

เพื่อไม่ให้การกำหนดแจกจ่ายค่า IP address ซ้ำซ้อนกัน จึงมีหน่วยงานกลางทำหน้าที่กำหนด IP address และแจกจ่ายเลขหมายให้แต่ละองค์กรได้ใช้งาน คือ หน่วยงาน InterNIC หรือ Internet Network Information Center เป็นผู้ดูแลฐานข้อมูลการแจกจ่าย IP address และการใช้งานให้เกิดประโยชน์สูงสุด บริษัทหรือองค์กรที่ต้องการ IP address ในการกำหนดใช้งานเพื่อเชื่อมระบบเครือข่ายของตนเข้าสู่เครือข่ายอินเตอร์เน็ต จะต้องสมัครเป็นสมาชิกขององค์กร InterNIC เสียก่อน โดยจะมีการเก็บค่าบำรุงเป็นรายปีและแบ่งระดับของสมาชิกเป็น องค์กรขนาดใหญ่, องค์กรขนาดกลางและองค์กรขนาดเล็ก ตามลำดับ ค่าบำรุงสมาชิกก็จะจ่ายไม่เท่ากัน เมื่อบริษัทหรือองค์กรได้เป็นสมาชิกแล้วก็สามารถแจ้งความจำนงขอ IP address มาใช้งานได้ต่อไป สำหรับในประเทศไทย บริษัทใดที่ต้องการเชื่อมต่อเครือข่ายของตนเข้าสู่อินเตอร์เน็ตจะต้องติดต่อกับบริษัทผู้ให้บริการอินเตอร์เน็ต (Internet Service Provider หรือ ISP) แทนและสามารถขอ IP address ที่ ISP ได้ทันที เพราะทาง ISP ได้ขอ IP address เอาไว้สำหรับแจกจ่ายให้กับลูกค้าของตนแล้ว

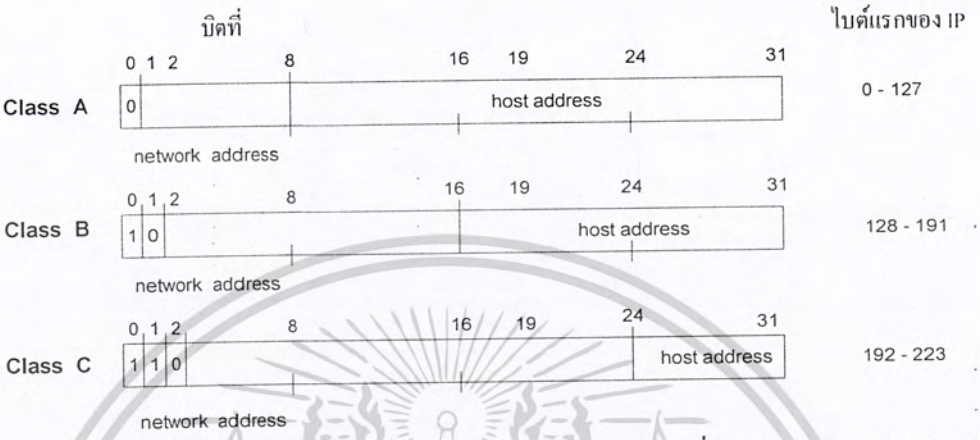
2.3.1 การจัดลำดับชั้นของเครือข่าย (Network Class)

การกำหนดค่า IP address ไม่สามารถกำหนดขึ้นได้ตามใจชอบ แต่มีระเบียบวิธีแบ่งและการกำหนดที่ชัดเจนเป็นมาตรฐาน ในหมายเลข IP address จะถูกแบ่งเป็น 4 ส่วน โดยคั่นด้วยเครื่องหมายจุด ซึ่งสามารถแบ่งแยกออกเป็น 2 ส่วนย่อยคือ ส่วนแรกเป็นหมายเลขของเครือข่าย (Network address) และส่วนที่สองเป็นหมายเลขของเครื่องลูกข่าย (Host address) ทั้งนี้การแบ่งส่วนจะเป็นไปตามการแบ่งระดับชั้นของเครือข่าย เรียกว่า Network class ซึ่งการกำหนดให้มี Network class นี้ก็เพื่อให้สามารถแจกจ่าย IP address ให้กับเครือข่ายต่างๆได้อย่างเหมาะสม เพราะในแต่ละเครือข่ายมักจะแตกต่างกัน บางเครือข่ายก็มีจำนวนเครื่องลูกข่ายมาก บางเครือข่ายมีน้อยแต่มีเครือข่ายย่อยๆในเครือข่ายหลักจำนวนมาก ฉะนั้นถ้าไม่มีการจัดลำดับของเครือข่ายให้ดี หมายเลข IP address ก็จะถูกใช้งานอย่างสิ้นเปลืองและใช้งานไม่ได้เต็มจำนวนที่มี ลำดับชั้นของเครือข่ายแบ่งได้เป็น5 ลำดับ คือ Class A, B, C, D และ E

ในแต่ละ Network class หมายเลข IP ทั้ง 32 บิตจะถูกกำหนดเป็นหมายเลขของเครือข่ายและหมายเลขของเครื่องลูกข่าย โดยมีเงื่อนไขดังนี้คือ

- Class A IP address บิตแรกของไบต์แรกสุดจะเป็น 0 เสมอ
- Class B IP address 2 บิตแรกของไบต์แรกสุดจะเป็น 1 และ 0 เสมอ
- Class C IP address 3 บิตแรกของไบต์แรกสุดจะเป็น 1,1 และ 0 เสมอ

การแบ่งหมายเลขเครือข่ายและหมายเลขเครื่องลูกข่ายจะเป็นดังรูป



รูปที่ 2.1 การแบ่งส่วนของหมายเลขเครือข่ายและหมายเลขเครื่องลูกข่าย ในแต่ละลำดับชั้นของเครือข่าย ใน Class A, B และ C

ดังนั้นแต่ละ Network classจะมีเครื่องลูกข่าย(Host address)ที่เป็นสมาชิกในเครือข่ายของตนได้ตามตาราง

ตารางที่ 2.1 network class

Class	จำนวนเครื่องลูกข่ายที่มีได้	IP address เริ่มต้น
A	2^{24} = 16,777,214	0-127
B	2^{16} = 65,536	128-191
C	2^{8} = 256	191-223
D	-	224-239
E	-	240-255

- Class A เป็นหมายเลข IP address ที่เริ่มตั้งแต่ 0-127 จะกำหนดให้กับเครือข่ายที่มีขนาดใหญ่ เพราะ 1 เครือข่ายสามารถมีเครื่องลูกข่ายได้กว่า 16 ล้านเครื่อง หมายเลข IP address จะเป็นลักษณะ net.host.host.host ตัวอย่างเช่น ค่า IP address ของ Class A เป็น 121.7.23.3 หมายถึงเครือข่ายเลขที่ 121 หมายเลขเครื่องคือ 7.23.3
- Class B เป็น IP address ที่เริ่มตั้งแต่ 128-191 จะกำหนดให้กับเครือข่ายที่มีขนาดใหญ่เช่นกันแต่เล็กกว่า Class A ในแต่ละเครือข่ายของ Class B สามารถมีเครื่องลูกข่ายได้ 2^{16}- จำนวน

แอดเดรสที่ใช้ควบคุมระบบเครือข่าย = 64,516 เครื่อง หมายเลข IP address จะเป็นลักษณะ net.net.host.host ตัวอย่างเช่น ค่า IP address ของ class B เป็น 137.103.210.2 หมายถึง เครือข่ายเลขที่ 137.103 หมายเลขเครื่องคือ 210.2

- Class C เป็น IP address ที่เริ่มตั้งแต่ 192-223 จะกำหนดให้กับเครือข่ายที่เป็นองค์กรทั่วไป ซึ่งส่วนใหญ่จะเป็นองค์กรขนาดกลางถึงเล็ก ในแต่ละเครือข่ายมีเครื่องลูกข่ายไม่เกิน 2^8- จำนวนแอดเดรสที่ใช้ควบคุมระบบเครือข่าย = 254 เครื่อง หมายเลข IP address จะเป็นลักษณะ net.net.net.host ตัวอย่างเช่น ค่า IP address ของ Class C เป็น 202.182.255.3 หมายถึง เครือข่ายเลขที่ 202.182.255 หมายเลขเครื่องคือ 3

- Class D เป็นการกำหนดหมายเลข IP address สำรองไว้สำหรับส่งข้อมูลแบบ Multicast ซึ่งจะไม่มีการแจกจ่ายให้ใช้งานทั่วไป

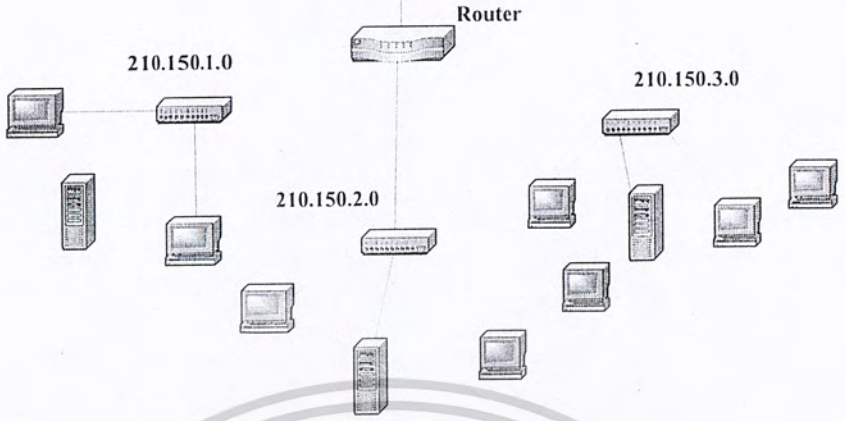
- Class E เป็นหมายเลข IP address พิเศษที่ใช้สำหรับงานทดสอบและพัฒนา ไม่มีการกำหนดให้ใช้งานทั่วไป

เราอาจจำหมายเลข IP address ได้ง่ายๆ คือ ในเครือข่าย Class A จะมีหมายเลข IP ตัวแรกอยู่ระหว่าง 0-127 ดังนั้นค่าที่ตามมาจะเป็นหมายเลขของเครื่อง ส่วนเครือข่าย Class B หมายเลข IP หน้าสุดจะเป็น 128-191 ดังนั้น ค่า IP สองตัวแรกจะเป็นหมายเลขเครือข่ายและค่า IP สองตัวหลังจะเป็นหมายเลขเครื่องสุดท้าย เครือข่ายใน Class C จะมีหมายเลข IP หน้าสุดเป็น 192-223 ดังนั้นค่า IP ตัวหลังสุดจะเป็นหมายเลขเครื่อง

2.3.2 Subnet

ปัจจุบันการกำหนดหมายเลข IP ของเครือข่ายประเภท Class A และ Class B นั้น ไม่มีการกำหนดให้แล้ว เนื่องจากแทบไม่มีเครือข่ายใดที่มีความจำเป็นต้องใช้ Address มากขนาดนั้น คงเหลือแต่ Class C เท่านั้นที่กำหนดให้แต่ละบริษัทหรือหน่วยงานต่างๆ และถึงแม้ว่าจะเป็นเครือข่ายที่ใช้หมายเลข IP ประเภท Class C ก็ตาม ส่วนมากก็ไม่มีใครเชื่อมต่อเครื่องลูกข่ายมากถึง 254 เครื่องในหนึ่งเครือข่าย หากทุกๆเครือข่ายมีอุปกรณ์หรือเครื่องติดตั้งใช้งานในเครือข่ายนั้นเพียงไม่กี่เครื่อง แต่จำเป็นต้องกำหนดหมายเลข IP ให้ใช้งานในเครือข่ายนั้น 254 เครื่อง จะเกิดปัญหามีหมายเลข IP ที่ไม่ได้ถูกใช้งานเป็นจำนวนมาก และไม่สามารถนำหมายเลข IP ที่ไม่ได้ใช้งานนี้ไปให้ผู้อื่นใช้ได้ ดังนั้นการทำ Subnet หรือ Sub Network จึงเกิดขึ้น เพื่อแบ่งเครือข่ายออกเป็นเครือข่ายย่อยๆและทำให้การกำหนดใช้งานหมายเลข IP ที่ได้รับมาสามารถแบ่งออกเป็นส่วนๆ เหมาะสมกับจำนวนอุปกรณ์ในแต่ละเครือข่ายได้ ทั้งยังแบ่งหมายเลข IP address ส่วนที่ไม่ได้ใช้ให้หน่วยงานหรือเครือข่ายอื่นได้อีกด้วย

ตัวอย่างเช่น หากหน่วยงานของเรามีเครือข่ายอยู่ 3 เครือข่าย แต่ละเครือข่ายก็มีหมายเลข IP Address Class C เป็นของตัวเองคือ 210.150.1.0 และ 210.150.2.0 และ 210.150.3.0 เครือข่ายทั้งหมดก็จะมีรูปร่างหน้าตาดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แสดงเครือข่าย 3 เครือข่ายที่แต่ละเครือข่ายมี IP Address ของตัวเอง

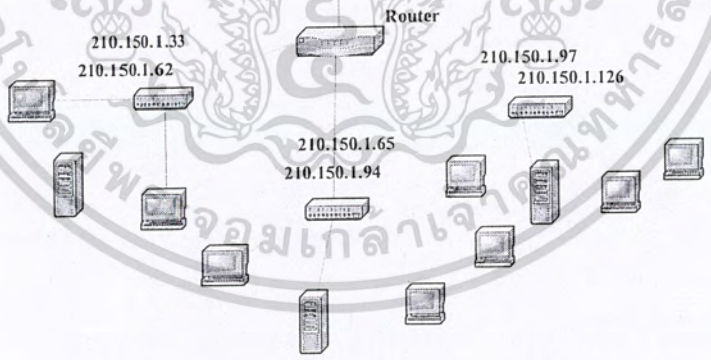
แต่ละเครือข่าย เช่น 210.150.1.0 จะสามารถเชื่อมต่อเครื่องลูกข่ายได้สูงสุดถึง 254 เครื่อง เนื่องจากใช้ 8 บิตสุดท้ายเป็นตัวกำหนด Host address ซึ่งจำนวนข้อมูล 8 บิตนี้จะอ้างอิงตัวเลขแตกต่างกันได้ 256 address หักออก 2 address คือ 0 กับ 255 ซึ่งจองไว้สำหรับควบคุมการทำงานของเครือข่าย ถ้าหากว่าแต่ละเครือข่ายเชื่อมต่อกับคอมพิวเตอร์ไม่กี่เครื่อง หมายเลข IP address ที่เหลือก็จะเสียไปเปล่าๆและจะแบ่งให้ผู้อื่นนำไปใช้ก็ไม่ได้เนื่องจาก อุปกรณ์ Router เมื่อพบ IP address 210.150.1.x จะไม่สนใจข้อมูลตัวสุดท้ายทางขวามือสุด แต่จะส่งข้อมูลให้ระบบเครือข่าย โดยดูที่ IP address 24 บิต ซ้ายมือคือดูที่หมายเลขประจำเครือข่าย (Network address) ในที่นี้คือ 210.150.1 เท่านั้น เนื่องจาก Class C แบ่งการอ้างอิง IP address 24 บิตซ้ายมือ สำหรับเป็น Network address และ 8 บิตขวามือเป็นหมายเลขเครื่อง (Host address) การทำ Subnet จะช่วยให้การกำหนดหมายเลข IP address ถูกแบ่งออกเป็นเครือข่ายย่อยๆที่ใช้ IP address ได้อย่างมีประสิทธิภาพ

จากตัวอย่างข้างต้นหากเรานำ Host address จำนวน 8 บิต มาแบ่งออกเป็น 2 ส่วน เพื่อแบ่งเครือข่ายออกเป็นเครือข่ายย่อยๆเช่นใช้ข้อมูล 3 บิตทางซ้ายมือเป็นตัวกำหนดเครือข่ายย่อย และใช้ 5 บิตทางขวามือเป็นตัวกำหนด Host address ดังนั้นในหมายเลข IP address 1 ชุด เราสามารถแบ่งเป็นเครือข่ายย่อยได้ 8 เครือข่ายโดยในแต่ละเครือข่ายจะมีเครื่องลูกข่ายได้ถึง 32 เครื่อง เราเรียกการแบ่งเครือข่ายย่อยนี้ว่า Subnet ดังแสดงในตารางต่อไปนี้

ตารางที่ 2.2 แสดงการแบ่ง Host address ออกเป็น 2 ส่วน คือ Subnet และ Host address

IP address ของ เครือข่าย 24 บิต	Host address เดิมขนาด 8 บิต	
(Network address)	Subnet address 3 บิต	Host address ใหม่ 5 บิต
210.150.1	000	00000 ถึง 11111
210.150.1	001	00000 ถึง 11111
210.150.1	010	00000 ถึง 11111
210.150.1	011	00000 ถึง 11111
210.150.1	100	00000 ถึง 11111
210.150.1	101	00000 ถึง 11111
210.150.1	110	00000 ถึง 11111
210.150.1	111	00000 ถึง 11111

แต่ในทางปฏิบัติ Subnet address 000 และ 111 กับ Host address 00000 และ 11111 จะถูกจองไว้สำหรับใช้ในการควบคุม ดังนั้น หากเราแบ่ง 3 บิตซ้ายมือเป็น Subnet และ 5 บิตขวามือเป็น Host address เราจะได้เครือข่ายย่อยทั้งหมด 6 เครือข่ายย่อย และแต่ละเครือข่ายจะมีเครื่องลูกข่ายได้สูงสุด 30 เครื่อง



รูปที่ 2.3 แสดงการแบ่ง Subnet ออกเป็นเครือข่ายย่อย โดยใช้ IP Address ตัวเดียว คือ 210.150.1.0 (ในรูปแสดงเพียง 3 เครือข่าย จากทั้งหมด 6 เครือข่าย)

ซึ่งจากตัวอย่างเราจะเห็นว่าเราใช้หมายเลข IP address เพียงค่าเดียวคือ 210.151.1.0 มาแบ่งออกเป็นเครือข่ายย่อยๆหรือ Subnet โดยในตัวอย่างนี้เราแบ่งออกเป็น 6 Subnet สูงสุด แต่ละ Subnet จะเชื่อมต่อเครื่องลูกข่ายได้สูงสุด 30 เครื่อง ทำให้เราสามารถกำหนดใช้งานหมายเลข IP

address ได้อย่างมีประสิทธิภาพ และไม่เสีย IP address โดยไม่จำเป็น แต่ในการแบ่งเครือข่ายย่อยนี้ อุปกรณ์ Router จะต้องทราบข้อมูลว่าการแบ่ง Subnet ของเรานั้น ใช้ข้อมูลกี่บิตเป็น Subnet และใช้ข้อมูลกี่บิตเป็น Host address จึงจะส่งข้อมูลให้เครือข่ายได้อย่างถูกต้อง เราเรียกข้อมูลการแบ่ง Subnet นี้ว่า " Subnet Mask" ซึ่งจะกำหนดให้ค่าของ Network address และ Subnet address มีค่าเป็น "1" ส่วน Subnet Mask ของ Host address จะมีค่าเป็น "0" ดังแสดงในตารางต่อไปนี้

ตารางที่ 2.3 แสดงการกำหนด Subnet Mask ของ Subnet ขนาด 3 บิต

	Network address	Subnet	Host address
IP address	11010010.10010110.00000001	xxx	xxxxx
Subnet Mask	11111111.11111111.11111111	111	00000
Subnet Mask เลขฐานสิบ	255.255.255	224	

เมื่อเรากำหนดค่าของ Subnet Mask ได้แล้วและใส่ค่านี้ลงในอุปกรณ์Router ตัวRouter ก็จะรับส่งข้อมูลในเครือข่ายได้อย่างถูกต้อง เนื่องจากอุปกรณ์ Router จะตรวจสอบหมายเลข IP address ว่าเป็นของเครือข่ายใดโดยใช้ข้อมูลถึง 27 บิต แทนที่จะเป็น 24 บิตตามเดิมโดยใช้ Subnet Mask นี้เข้าช่วย

2.3.3 Class C Subnet

เนื่องจากหมายเลข IP address ที่เราใช้งานกันอยู่ทุกวันนี้จะเป็นเครือข่าย Class C เกือบทั้งหมด ซึ่งหมายถึง เราจะใช้ 8 บิตขวาสุดของ IP address ในการกำหนด Host address หรือเครื่องลูกข่าย ดังนั้นการ Subnet ใน Class C ก็คือการแบ่งข้อมูล 8 บิตของ Host address นี้ออกเป็น Subnet address และ Host address ใหม่นั่นเอง ซึ่งใน Class C นี้เราจะแบ่ง Subnet ได้ทั้งหมด 5 แบบดังตารางต่อไปนี้

ตารางที่ 2.4 แสดง Subnet ทั้งหมดของ Class C ซึ่งหากไม่มีการทำ Subnet ค่า Subnet Mask จะเป็น 255.255.255.0

จำนวนเครือข่ายย่อย (Subnet)	จำนวนเครื่องลูกข่าย (Host address)	Subnet Mask เลขฐานสิบ	จำนวนบิตที่ใช้เป็น Subnet
2	62	255.255.255.192	2
6	30	255.255.255.224	3
14	14	255.255.255.240	4
30	6	255.255.255.248	5
62	2	255.255.255.252	6

สำหรับ Subnet 0 คือ ทุกบิตของ Subnet มีค่าเป็น "0 " ทั้งหมด และ Subnet 7 ที่ทุกบิตมีค่าเป็น "1 " ทั้งหมดจะถูกจองไว้ใช้สำหรับการควบคุมระบบเครือข่าย และ Subnet ที่มีจำนวนเครือข่ายย่อยเท่ากับหนึ่งจะใช้งานไม่ได้เนื่องจากไม่มีจำนวนบิตเหลือให้นำไปใช้ทำ Subnet ดังนั้นจาก Host address ขนาด 8 บิต จึงเหลือนำมาใช้ทำ Subnet ได้เพียง 5 แบบเท่านั้น เพราะถูกหักออกไป 3 Subnet จากเหตุผลดังกล่าว

ในแต่ละ Subnet ก็จะมีการจอง Host address ไว้ 2 ตำแหน่ง คือ Host address ที่มีค่าเป็น "0 " หมดทุกบิต เอาไว้ใช้สำหรับเป็นหมายเลขประจำเครือข่าย (Network address) และ Host address ที่มีค่าเป็น "1 " หมดทุกบิตจะถูกจองไว้ใช้สำหรับเป็น Broadcast address ดังนั้นในทุกๆ แบบของ Subnet จำนวนเครื่องลูกข่ายจึงถูกหักออก 2 address คือ address ที่เป็น "0" ทั้งหมดทุกบิต และที่เป็น "1 " หมดทุกบิต เหลือจำนวนเครื่องลูกข่ายสูงสุดดังแสดงในตารางข้างต้น

เราจะเห็นได้ว่าถ้าเราแบ่งจำนวนเครือข่ายย่อยออกไม่กี่เครือข่าย แต่ละเครือข่ายย่อยก็จะมีจำนวนเครื่องลูกข่ายได้มาก ในทางกลับกันหากเราแบ่งจำนวนเครือข่ายย่อยออกไปมากๆ แต่ละเครือข่ายย่อยก็จะมีเครื่องลูกข่ายได้ไม่กี่เครื่องเท่านั้น ในทางปฏิบัติจำนวนเครือข่ายย่อย 2 เครือข่าย หรือ 64 เครือข่ายจึงไม่เหมาะสมในการวางระบบ เนื่องจากเครือข่ายย่อย 2 เครือข่ายนั้นน้อยเกินไป และขยายเครือข่ายย่อยอีกไม่ได้ ส่วนจำนวนเครือข่ายย่อย 62 เครือข่ายนั้นก็มากเกินไปและแต่ละเครือข่ายย่อยก็มีเครื่องลูกข่ายได้เพียง 2 เครื่องเท่านั้น ในการใช้งานจริงก็น่าจะเป็นการแบ่งเครือข่ายย่อยออกเป็น 6 เครือข่าย โดยแต่ละเครือข่ายย่อยมีลูกข่าย 30 เครื่อง หรือแบ่งออกเป็นเครือข่ายย่อย 14 เครือข่าย และแต่ละเครือข่ายย่อยมีลูกข่าย 14 เครื่องเท่านั้นสำหรับการทำ Subnet ของ Class C นี้

2.3.4 หมายเลขอ้างอิงเครือข่ายและ Broadcast Address

ในการกำหนดหมายเลข IP address ให้กับอุปกรณ์และเครื่องเซิร์ฟเวอร์ให้บริการ จะต้องมีการกำหนดค่าหมายเลข IP address พิเศษให้กับเครือข่ายนั้นด้วย ดังนี้

- กำหนดหมายเลขประจำเครือข่าย (Network address)
- กำหนดหมายเลข IP สำหรับการส่งข้อมูล Broadcast ในเครือข่ายนั้น (Broadcast address) ใช้เพื่อส่งข้อมูลให้กับอุปกรณ์ทั้งหมดของเครือข่ายนั้น

หมายเลขประจำเครือข่าย คือ หมายเลข IP address ที่ใช้อ้างอิงเครือข่ายนั้นโดยไม่เจาะจงถึงเครื่องลูกข่ายภายใน ซึ่งโดยทั่วไปการกำหนดหมายเลขของเครือข่ายทำได้โดยกำหนดให้ค่าหมายเลข host เป็น 0 เช่น เครือข่าย 172.10.0.0

หมายเลข IP address สำหรับส่งข้อมูล broadcast ของเครือข่ายนั้นๆ คือ หมายเลข IP address ที่กำหนดค่าหมายเลข host เป็น 255 ทั้งหมด จากตัวอย่างหมายเลข IP address ที่ 172.10.255.255 จะเป็นหมายเลข Network broadcast address ของเครือข่ายหมายเลข 172.10.0.0 และเครือข่ายนี้มี IP address ที่จะจัดสรรให้เป็นหมายเลข IP address ประจำเครือข่ายได้ตั้งแต่หมายเลข IP ที่ 172.10.0.1 ถึง 172.10.255.254 เป็นต้น

นอกจากนี้ได้มีการกำหนดให้หมายเลข IP ที่ 255.255.255.255 เป็นหมายเลข IP สำหรับการ Broadcast ทั่วไปไม่เฉพาะเจาะจงกับเครือข่ายใดเครือข่ายหนึ่งอีกด้วย

2.3.5 Multicast Address

จากการแบ่งเครือข่ายด้วยหมายเลข IP Address เป็นชั้นต่างๆของเครือข่ายเป็น Class A, B, C, D และ E นั้น เครือข่ายใน Class D ถูกกำหนดให้เป็นหมายเลข IP Address สำรองเพื่อใช้งานในแบบ Multicast กล่าวคือหมายเลข IP ของ Class D ตั้งแต่ 224.0.0.0 ถึง 239.255.255.255 จะทำหน้าที่เป็น Multicast Address โดยแบ่งเป็นกลุ่มย่อยๆ ซึ่งสมาชิกในกลุ่มจะรับข้อมูลที่ถูกส่งออกมาให้ในระหว่างเครือข่าย ตัวอย่างเช่น โปรโตคอล RIPv2 ใช้หมายเลข IP ที่ 224.0.0.9 ทำหน้าที่เป็น Multicast Address ของกลุ่มตนเอง คือกลุ่มอุปกรณ์ Gateway หรือ Router ที่ใช้โปรโตคอล RIPv2 ในการติดต่อระหว่างกัน กลุ่มอุปกรณ์อื่นหรืออุปกรณ์ที่ใช้การติดต่อกันด้วยโปรโตคอลอื่นก็จะไม่สนใจข้อมูลที่ถูกส่งเป็น Broadcast ออกมา ดังนั้นในการส่งข้อมูลถึงกันในกลุ่มจะติดต่อด้วย IP Address นี้เท่านั้น ตัวอย่างของค่า Multicast Address ที่กำหนดไว้จะดูได้ในตารางต่อไปนี้

ตารางที่ 2.5 ตัวอย่าง multicastt address

โปรโตคอล	Multicast address	ความหมาย
RIPv2	224.0.0.9	กลุ่มอุปกรณ์ Router ที่ใช้โปรโตคอล RIPv2 รับข้อมูล broadcast
OSPF	224.0.0.5	กลุ่มอุปกรณ์ Router ที่ใช้โปรโตคอล OSPF รับข้อมูล broadcast
	224.0.0.6	ใช้เฉพาะบาง Router ที่ใช้โปรโตคอล OSPF รับข้อมูล broadcast
Multicast ทั่วไป	224.0.0.1	ทุกระบบในเครือข่ายนั้น รับข้อมูล broadcast

2.3.6 IP Address ที่ออกสู่อินเตอร์เน็ตไม่ได้

เนื่องจากความนิยมในการใช้โปรโตคอล TCP/IP มีมากขึ้นทั้งการใช้งานในเครือข่ายส่วนตัว (Private network/Intranet) และในเครือข่ายอินเตอร์เน็ต หน่วยงาน InterNIC ได้มีการกำหนดให้หมายเลขประจำเครือข่ายบางอันเป็นหมายเลขพิเศษสำรองเอาไว้ใช้งาน ในวัตถุประสงค์แบบเครือข่ายส่วนตัวที่ไม่สามารถใช้งานในอินเตอร์เน็ตได้ และกำหนดให้หมายเลขเครือข่ายนั้นเป็นแบบ Non-routable คือไม่สามารถรับส่งข้อมูลออกสู่อินเตอร์เน็ตได้ ซึ่งหมายเลขเครือข่ายเฉพาะนี้จะใช้ในการกำหนดหมายเลข IP address ให้กับเครือข่ายขององค์กรหรือหน่วยงานที่ไม่ต้องการเชื่อมต่อเข้าสู่อินเตอร์เน็ต โดยหมายเลขเครือข่ายดังกล่าวมีดังนี้

ตารางที่ 2.6 แสดงเครือข่ายที่เข้าสู่อินเตอร์เน็ตไม่ได้

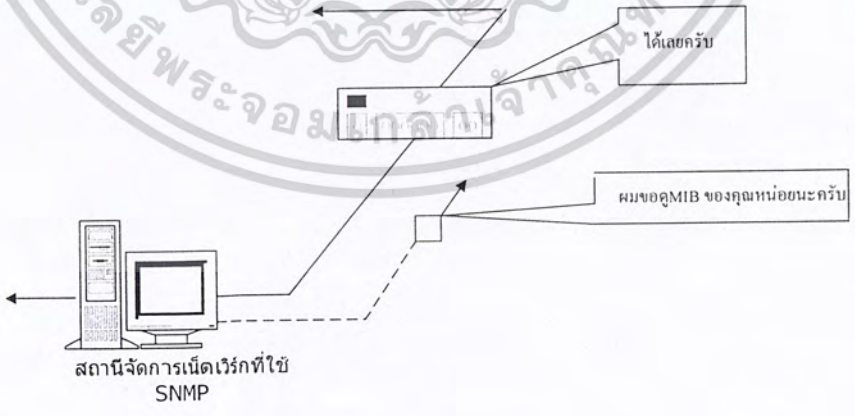
Network Class	จำนวนเครือข่ายที่กำหนดได้	หมายเลขเครือข่าย
A	1	10.0.0.0
B	16	172.16.0.0 – 173.31.0.0
C	256	192.168.0.0 – 192.168.255.0

ดังนั้นในเครือข่าย Class C ที่มีการกำหนดหมายเลขเครือข่ายเป็น 192.168.0.0 และมีเครื่องลูกข่ายเป็นหมายเลข IP ตั้งแต่ 192.168.0.1 ถึง 192.168.0.254 นั้น เราจะรู้ได้ทันทีว่าเครือข่ายนี้ใช้งานโดยไม่มีการเชื่อมต่อออกสู่อินเตอร์เน็ต ซึ่งการกำหนดหมายเลขเครือข่ายในกลุ่ม Non-Routable Network นี้ก็เพื่อให้แน่ใจได้ว่าเครือข่ายขององค์กรนั้นจะไม่สามารถส่งผ่านข้อมูลใดๆไปยังอินเตอร์เน็ตได้ และผู้ใช้อินเตอร์เน็ตจากภายนอกก็จะไม่สามารถติดต่อเรียกใช้ข้อมูลหรือเครื่องเซิร์ฟเวอร์ที่ให้บริการใดๆในเครือข่ายนี้ได้เช่นกัน

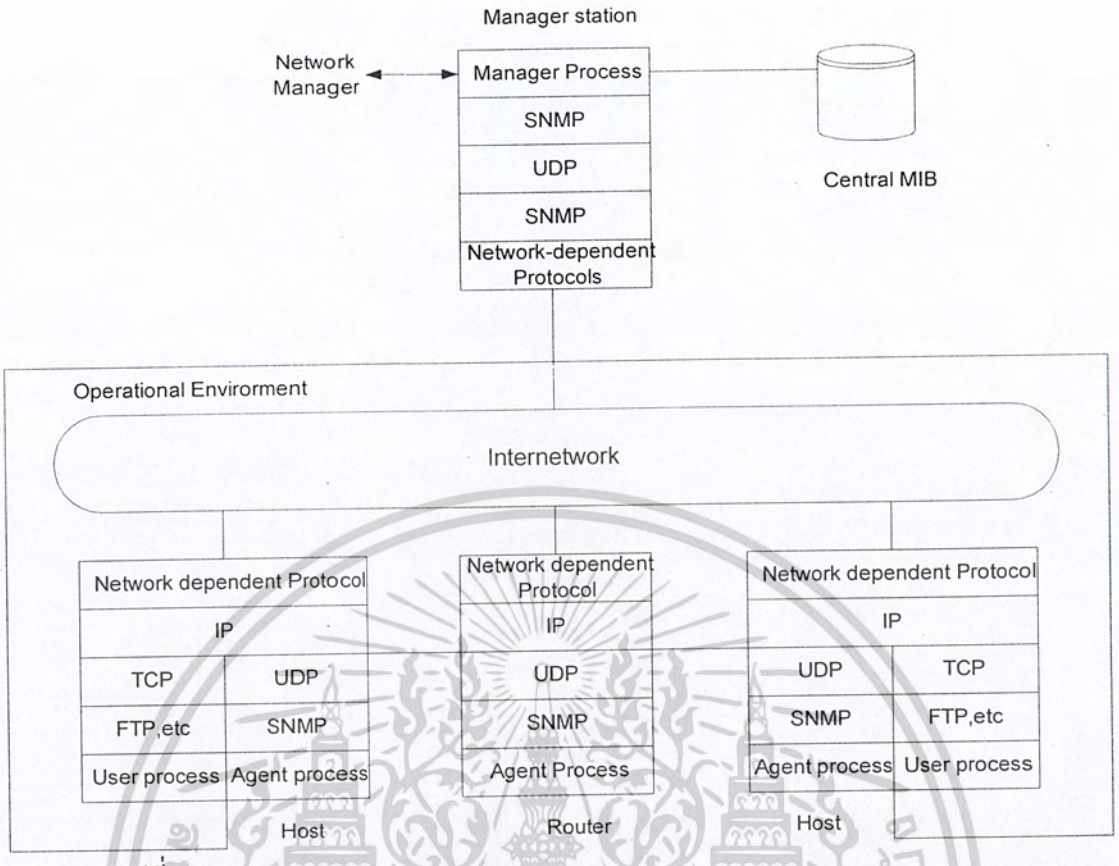
2.4 Simple Network Management Protocol (SNMP)

Simple Network Management Protocol หรือ SNMP เป็นโปรโตคอลที่รู้จักกันดีที่ใช้ในการดูแลเครือข่าย ทำงานโดยมีส่วนของ Manager คอยสั่งงานหรือขอข้อมูล Agent ที่คอยดูแล MIB ด้วยข้อความคำสั่งที่เรียกว่า Protocol Data Units หรือ PDU

วิธีรวบรวมข้อมูลของเน็ตเวิร์ค SNMP ใช้เทคนิคที่เรียกว่า MIB Collection เหมือนกับนักสำรวจสำมะโนครัวที่เข้าไปถามอุปกรณ์แต่ละชิ้นว่า "สบายดีไหม" แล้วเก็บข้อมูลลง MIB ข้อดีของ SNMP คือ ฮาร์ดแวร์ของเน็ตเวิร์กไม่ต้องฉลาดมากถึงขนาดที่ต้องคอยรายงานปัญหา เมื่อผู้จัดการ SNMP เข้าไปดูที่อุปกรณ์แต่ละตัวก็จะได้สถานะปัจจุบันและข้อมูลใน MIB ดังรูป



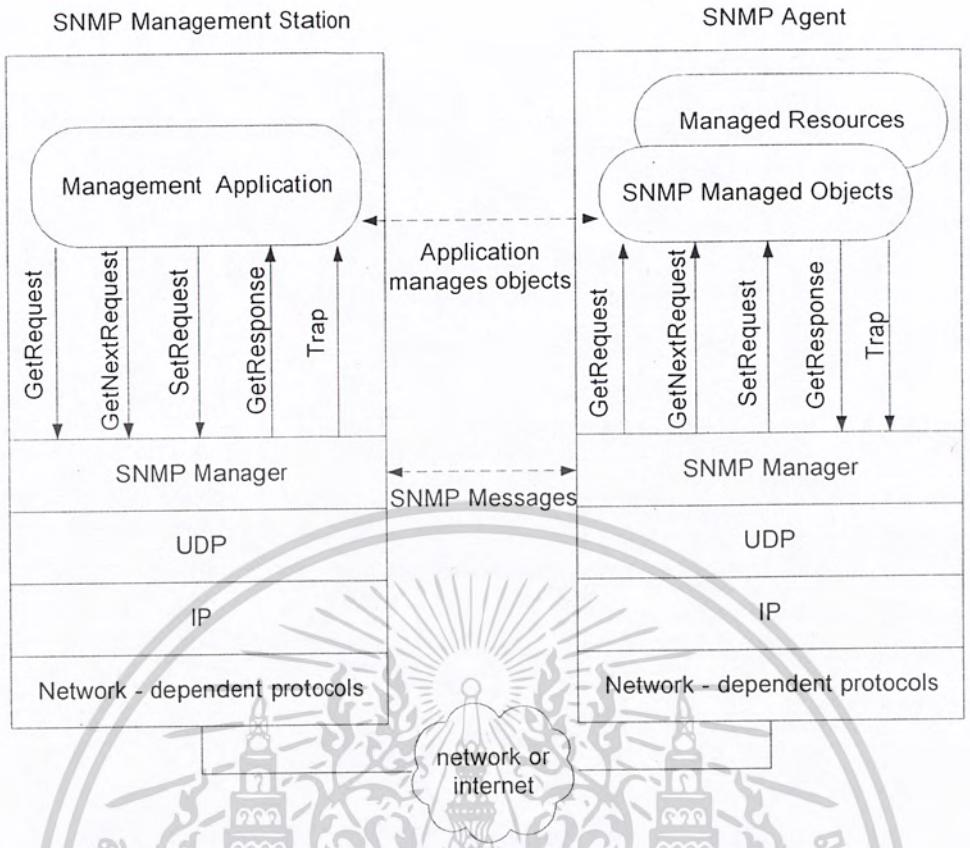
รูปที่ 2.4 แสดงวิธีรวบรวมข้อมูลของเน็ตเวิร์ค SNMP



รูปที่ 2.5 Configuration of SNMP

จากรูปที่ 2.5 Manager process จะควบคุมการติดต่อไปยัง MIB ที่ Manager station และติดต่อไปยังผู้จัดการเครือข่าย ผู้จัดการเครือข่ายจะจัดการเครือข่ายโดยใช้ SNMP ซึ่งถูกติดตั้งบน UDP, IP และโปรโตคอลอื่นๆที่เกี่ยวข้องกับเครือข่าย เช่น Ethernet, FDDI และ x.25

แต่ละ Agent จะต้องทำการติดตั้ง SNMP, UDPและ IP นอกจากนั้น Agent process เป็นตัวแปลความหมายของ SNMP message และควบคุม MIB ของ Agent สำหรับอุปกรณ์ของ Agent จะต้องสนับสนุนการใช้งานอื่นๆ เช่น FTP, UDP และ TCP



รูปที่ 2.6 Role of SNMP

จากรูปที่ 2.6 Management station มี SNMP message 3 ชนิด คือ GetRequest, GetNextRequest และ SetRequest โดย Message ทั้งสามจะถูกรับรู้โดย Agent ในรูปแบบของ GetResponse message นอกจากนี้ Agent อาจจะมี Trap message ในการตอบสนองต่อเหตุการณ์ที่มีผลกระทบต่อ MIB และทรัพยากร

2.4.1 คำสั่ง SNMP

- **Get request :** ตัว Manager จะร้องขอที่จะดึงข้อมูลมายัง Agent เพื่อขอข้อมูลของอุปกรณ์ตัวที่เก็บไว้ใน MIBs
- **Get-Next request :** จะถูกใช้โดย Manager เพื่อขอดึงข้อมูลทีเดียวหลายตัว เพื่อลดทราฟฟิกของเครือข่าย และก็สามารถดึงข้อมูลที่ไม่มีการระบุที่แน่นอนได้อีกด้วย
- **Set :** ตัว Manager จะเป็นตัวสั่งให้ Agent กำหนดหรือเปลี่ยนค่าข้อมูลใน MIB ให้เป็นไปตามที่ต้องการ

- **Get-Response :** จะส่งมาจาก Agent เป็นการตอบรับคำสั่ง Set, Get หรือ Get-Next ที่มาจาก Manager ว่ารับหรือไม่และเพื่อเป็นการตรวจความผิดพลาดด้วย และเป็นการส่งผลลัพธ์ให้พร้อมเลยได้ด้วย
- **Trap :** จะเป็นข้อความหรือข้อมูลที่ส่งจาก Agent ไปยัง Manager เพื่อบอกถึงเหตุการณ์ต่างๆ ไม่ว่าจะเป็นอุปกรณ์นั้นแฮงค์ รีบูต โอเวอร์โหลดหรือเครือข่ายขาดกันเป็นต้น

2.4.2 Limitation of SNMP

ผู้ใช้ SNMP สำหรับการจัดการเครือข่ายจำเป็นต้องรู้ถึงข้อจำกัดดังต่อไปนี้

1) SNMP อาจจะไม่เหมาะสำหรับการจัดการกับเครือข่ายขนาดใหญ่ เนื่องจากประสิทธิภาพที่มีจำกัดในการ Poll ตัวอย่างเช่น ถ้าคุณต้องการส่งหนึ่ง Packet ออกไปเพื่อรับข้อมูลของอีกหนึ่ง Packet กลับมา ด้วยการหยั่งเสียงนี้ทำให้ Routine Message มีปริมาณมากและอาจจะมีปัญหาเวลาตอบสนองกลับซึ่งไม่สามารถยอมรับได้
2) SNMP อาจไม่เหมาะสำหรับการดึงข้อมูลปริมาณมากๆ ตัวอย่างเช่น การ Routing ทั้งตาราง
3) SNMP trap ยังไม่เป็นที่ยอมรับ ในกรณีของ UDP/IP ใช้เพื่อการส่ง Trap Message ซึ่ง Agent ไม่อาจจะแน่ใจได้ว่า Message ไปถึง Management Station
4) SNMP เหมาะสำหรับการเฝ้าดูมากกว่าใช้ในการควบคุม
5) SNMP ไม่สนับสนุนคำสั่งที่บังคับโดยตรง เป็นเพียงการนำไป Trigger เหตุการณ์ที่ Agent ทางอ้อมเท่านั้น โดยการเซ็ตค่า Object
6) SNMP MIB Model ถูกจำกัดและไม่สามารถทำได้ง่ายในการสนับสนุน Application ซึ่งทำให้ต้องมีความสามารถในการจัดการ Query ค่าหรือชนิดของ Object
7) SNMP ไม่สนับสนุนการติดต่อสื่อสารระหว่าง Manager กับ Manager ตัวอย่างเช่น ไม่มีกลไกที่ทำให้ระบบจัดการเรียนรู้เกี่ยวกับอุปกรณ์และเครือข่ายที่จัดการโดยอีกระบบจัดการหนึ่ง

2.5 SNMP Management Information Base

ในระบบการจัดการ Network ซึ่งมีรากฐานอยู่บน TCP/IP (Transmission-Control Protocol/ Internet Protocol) ซึ่งการบริหารระบบนี้จะมี Database ซึ่งเก็บ Information ต่างๆ เกี่ยวกับสิ่งที่เราสนใจจะบริหารมัน ซึ่ง Database นี้จะถูกเสนอในรูปของ Management Information Base (MIB) ซึ่งแต่ละ Resource ที่เราจะบริหารนั้นจะถูกแสดงในรูปของ Object ซึ่ง MIB จะมีโครงสร้างที่เก็บแต่ละ Object เอาไว้ ซึ่งในการบริหาร Network นี้เราสามารถทำการ Monitor Resource ต่างๆ ได้ โดย

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การอ่านค่าของ Object ที่เราสนใจใน MIB ซึ่งโครงสร้างของ MIB จะมีรูปแบบตาม Structure of Management Information(SMI)

2.5.1 Structure of Management Information

SMI ได้ถูกกำหนดไว้ตามมาตรฐาน RFC1155 ซึ่งมีการประกาศโครงสร้างทั่วไปของ MIB รวมถึงการกำหนด Type ของ Data ที่ใช้ใน MIB รวมถึงการอ้างถึง Resource ใน MIB โดยชื่อได้อย่างไร ซึ่ง SMI นี้จะเป็นตัวสนับสนุนและกำหนดนโยบายทั้งหมดใน MIB ตัวอย่างเช่น MIB สามารถเก็บค่าได้เป็น Type ประเภทเดียวเท่านั้น หรือเก็บเป็น Scalars หรือเป็น Array 2 มิติของ Scalars ซึ่งเก็บค่า Type ก็จะแตกต่างกันตาม Object ต่างๆ ใน MIB ซึ่ง SMI จะไม่ Support โครงสร้างข้อมูลที่มีความซับซ้อน

เพื่อให้มีมาตรฐานในการนำเสนอข้อมูลที่จะต้องจัดการ ดังนั้น SMI ต้องมีเทคนิคที่เป็นมาตรฐานในเรื่องของ

- กำหนดโครงสร้างของ MIB
- การกำหนด Object ใดๆ ประกอบด้วย ไวยากรณ์และค่าของแต่ละ Object
- การเข้ารหัสให้กับค่าของ Object

2.5.2 MIB Structure

กลุ่มของแต่ละ Type ใน Object ต่างๆ ใน MIB จะถูกกำหนดในมาตรฐาน ASN.1 ซึ่งกำหนดเป็น Type " OBJECT INDENTIFIRE" ซึ่ง Identifier นี้จะใช้เพื่อตั้งชื่อให้กับ Object ซึ่งค่าของ OBJECT IDENTIFIER จะเป็นโครงสร้างลำดับชั้น (Hierarchical) ทำให้เป็นแบบแผนในการตั้งชื่อนี้ใช้ในการระบุโครงสร้างของ Object Type ได้ด้วย

Object identifier จะมีค่าของแต่ละ Object เป็นเอกลักษณ์ โดยค่าของมันจะประกอบด้วยลำดับของ Integer และ Object ต่างๆ จะถูกประกาศในรูปแบบต้นไม้ (Tree structure) ซึ่งค่าของมันประกอบด้วยชุดหนึ่งของเลขจำนวนเต็มคั่นด้วยจุดใช้เพื่อการอ้างอิงใน Node ใดๆ ในโครงสร้างต้นไม้ โดยจะเริ่มจาก Root ระดับลงมาเป็น CCITT , ISO และ Joint-CCITT-ISO โครงสร้างใน MIB นี้มีการกำหนดเป็นมาตรฐานของ RFC1155 เพื่อให้อุปกรณ์ต่างๆ ในเครือข่ายสามารถถูกรู้จักจาก Management station ได้แม้ว่า SNMP application ที่ผู้ขายต่างๆ สร้างขึ้นมาหลากหลายกันไป

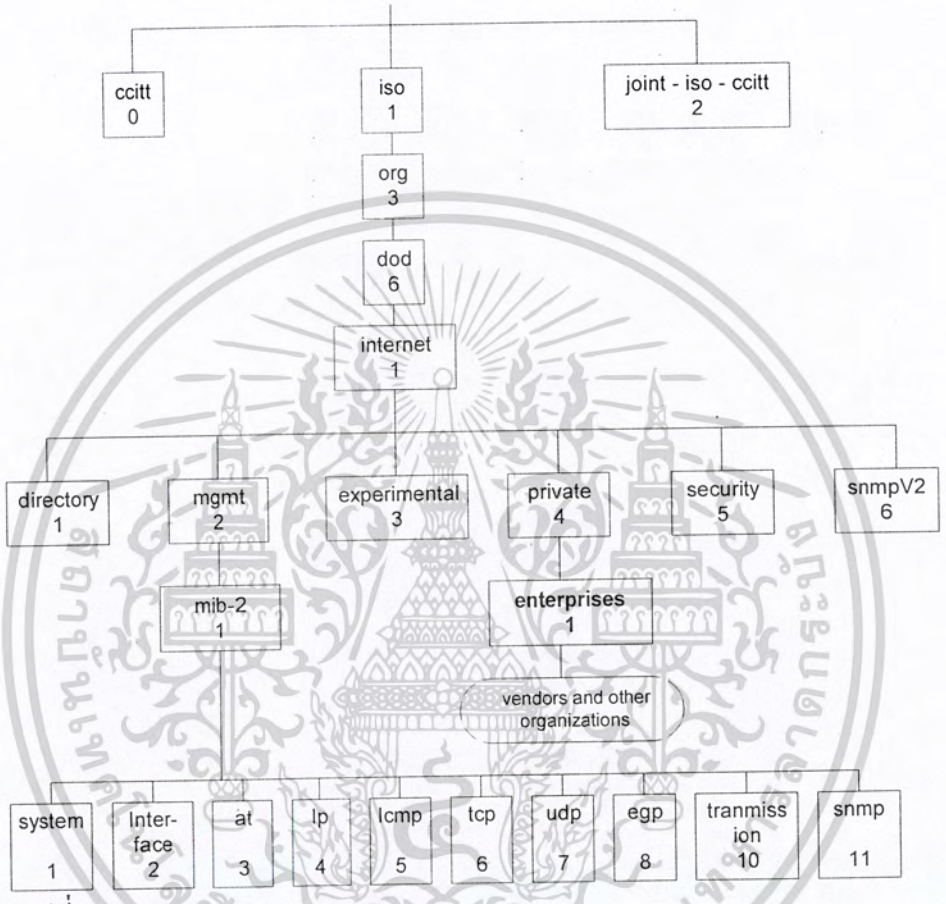
ตัวอย่าง internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }

โดยแสดงดังรูป 2.7 internet node จะมีค่า object identifier คือ 1.3.6.1 ซึ่งโครงสร้าง MIB ภายใต้ internet group มีกลุ่มย่อยทั้งหมด 6 group

1. directory (1) subtree นี้ถูกจองไว้สำหรับใช้งานในอนาคตกับ OSI directory
2. mgmt(2) subtree นี้ถูกใช้สำหรับการจัดการภายใต้ SNMPv1
3. experimental(3) subtree นี้ถูกใช้สำหรับการระบุ Object ที่ใช้ในการทดลอง Internet

39660

4. private(4) subtree นี้ถูกใช้สำหรับการระบุ Object ที่ถูกกำหนดโดยเฉพาะ (สำหรับให้ผู้ผลิตกำหนดตัวแปรเฉพาะอุปกรณ์)
5. security(5) ใช้ในระบบรักษาความปลอดภัย
6. SNMPv2(6) subtree นี้ถูกใช้สำหรับการจัดการภายใต้ SNMPv2



รูปที่ 2.7 แสดง Structure of MIB

2.5.2.1 Object Syntax

Object ภายใน SNMP MIB หนึ่งและ MIB structure ทั้งหมด ถูกกำหนดให้ใช้ ASN.1 เพื่อการรักษาไว้ซึ่งวัตถุประสงค์นั่นคือ สามารถเข้าใจและใช้งานได้ง่าย ASN.1 มี Type ดังนี้

1. Universal Types ภายใน UNIVERSAL class มีชนิดของข้อมูลเหล่านี้เท่านั้นที่ยอมรับให้ใช้ในการกำหนด MIB object ได้คือ

- INTERGER (UNIVERSAL 2)
- OCTET STRING (UNIVERSAL 4)
- NULL (UNIVERSAL 5)

- OBJECT IDENTIFIER (UNIVERSAL 6)
- SEQUENCE , SEQUENCE OF (UNIVERSAL 2)

ใน SNMPv2 เพิ่ม type ขึ้นมาอีกหนึ่งคือ

- BIT STRING

ซึ่ง 4 type แรก เป็น type basic ที่มีมาแต่ดั้งเดิม ส่วน type แบบ ENUMERATED นั้นยังไม่มีการสร้างเกิดขึ้น เมื่อมีการกำหนด ENUMERATED type ขึ้นมันจะมีการใช้ร่วมกับ INTERGER type ซึ่งจะมีระเบียบในการใช้ ENUMERATED type อยู่ 2 อย่างดังนี้

1. ค่าที่เป็น 0 จะไม่มีการใช้แต่จะใช้ในระดับการ check error
2. เฉพาะ enumerate integer ที่ถูกใช้ และจะมี type หนึ่งของ enumerated type "other" ซึ่งจะใช้เป็น label

ค่า object identifier ของแต่ละ object จะมีค่าไม่ซ้ำกัน ซึ่งจะประกอบด้วยลำดับของ integer ซึ่งเลขเหล่านี้ทำให้เราสามารถรู้ subidentifiers ได้ การอ่านค่า identifier จะอ่านจาก ซ้ายไปขวา ซึ่งจะถูกประกาศในแต่ละตำแหน่งของ Object ใน Structure ของ MIB ตัวอย่างนี้เป็น Object Identifier ของ object tcpConnTable แสดงได้ดังนี้

iso	org	dod	internet	mgmt	mib-2	tcp	tcpConntable
1	3	6	1	2	1	6	13

เราสามารถเขียน identifier ได้ดังนี้ 1.3.6.1.2.1.6.13

2.5.2.2 Application Types

RFC 1155 ได้ประกาศ data type ของ application type ไว้ดังนี้

- NetworkAddress : ชนิดของข้อมูลนี้ถูกกำหนดโดยการใช้การสร้างจาก CHOICE เพื่อใช้ในการเลือกรูปแบบ address จากหนึ่งในจำนวนหลายๆ protocol
- IpAddress : คือ 32-bit address ที่ใช้ในรูปแบบที่ได้มีการระบุไว้ใน ip
- Counter : จำนวนเต็มศูนย์และบวกซึ่งอาจทำให้เพิ่มขึ้นได้แต่ไม่มีการลดลง ค่าที่มากที่สุดที่ได้ กำหนดไว้คือ 2^{32} - 1(4,294,967,295) เมื่อ counter ถูกเพิ่มขึ้นจนถึงค่ามากที่สุดที่สามารถเป็นไปได้มันจะวนกลับมาเริ่มต้นที่ 0 ใหม่
- Gauge : จำนวนเต็มศูนย์และบวก ซึ่งอาจเพิ่มขึ้นหรือลดลงโดยค่าสูงสุดเป็น 2^{32} - 1 ถ้าค่าถูกใช้ไปจนถึงค่ามากที่สุดมันจะค้างอยู่ที่ค่านั้นจนกระทั่งมีการเริ่มต้นใหม่ (reset)
- TimeTicks : จำนวนเต็มศูนย์และบวกซึ่งนับเวลาใน 1/100 วินาที ตั้งแต่มีการเปลี่ยนแปลงเหตุการณ์บางอย่าง
- Opaque : ใช้ในการสนับสนุนความสามารถในการส่งข้อมูลที่ไม่มีหลักเกณฑ์

ซึ่ง counter เราจะรู้จักกันดีแบบ rollover counter มันจะเป็น type หนึ่งที่ใช้กันมากใน

object ต่างๆ เช่น application นี้จะใช้ type นี้ในการนับจำนวน packet หรือ octet ที่ส่งหรือที่ได้รับ ทางเลือกหนึ่งของ counter โดยจะพิจารณา counter ออกแบบโดย SMI ว่าเป็น latch counter ซึ่งจะมีการระบุค่า maximum และ กำหนดว่าเมื่อไรจะมีการ reset แต่การใช้ latch counter นี้ก็ก่อให้เกิดปัญหาเกิดขึ้นลองคิดดูถ้าเกิดระบบมีการจัดการจากหลาย station ซึ่งสามารถเข้าถึง counter เหล่านี้ได้อาจเกิดปัญหาจากการตั้งเวลาการ reset ได้ (หลายๆ station เข้ามา access พร้อมกัน) หรืออาจจะเป็นการตั้งค่า maximum ที่ไม่สอดคล้องกัน

2.5.2.3 Defining Object

MIB หนึ่งๆประกอบด้วยชุดของ object แต่ละ object จะมี type และค่าของ object โดย Type ของ Object จะกำหนดถึงประเภทของ object ที่ถูกจัดการความหมายของ type ของ Object ก็คือ Syntactic Description

ใน SNMP การใช้ Macro ในการกำหนดชุดของ Type ที่มีความสัมพันธ์กัน ได้นำมาใช้ในการกำหนด Object ที่จะต้องจัดการ ซึ่งระดับของ Definition มีดังนี้

- Macro Definition : ระบุไวยากรณ์ของชุดของ type ที่มีความสัมพันธ์กัน
- Macro Instance : instance ที่สร้างจาก macro definition ที่กำหนดโดยมีการใส่ argument ให้กับ parameter ใน macro definition เป็นการระบุถึงชนิดใดชนิดหนึ่งโดยเฉพาะ
- Macro instance value : ใส่ค่าหนึ่งค่าใดที่ระบุไว้ให้กับตัวแปร

2.5.2.4 Defining Tables

ในปัจจุบัน Structure of Management Information (SMI) สนับสนุนรูปแบบของการจัดโครงสร้างข้อมูลเพียงรูปแบบเดียวคือ Simple two-dimension table ที่ประกอบด้วย Scalar Values โดยความหมายของตารางทำให้เกิดการใช้ SEQUENCE, SEQUENCE OF ASN.1 Type และ IndexPart ของ OBJECT-TYPE macro

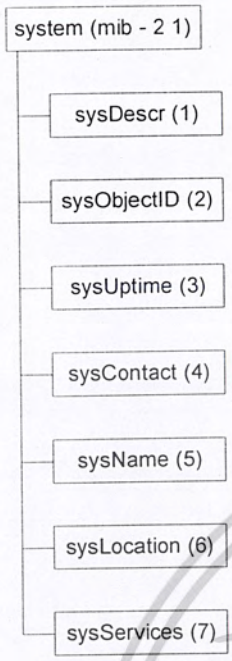


Table MIB - II System Group Objects

Object	Syntax	Access	Status	Description
SysDescr	DisplayString (SIZE (0..255))	RO	M	A description of the entity, such as hardware, operation system ,etc
SysObjectID	OBJECT IDENTIFIER	RO	M	The vendor's authoritative identification of the network-management sub-system contained in the entity
SysUptime	TimeTicks	RO	M	The time since the network-management portion of the system was last re
SysContact	DisplayString (SIZE (0..255))	RW	M	An administratively assigned name for this managed node
Sys Name	DisplayString (SIZE (0..255))	RW	M	The physical location of this node
SysLocation	DisplayString (SIZE (0..255))	RO	M	A value indicating the set of services that this entity primarily offers

รูปที่ 2.8 ตัวอย่างตาราง MIB

2.5.2.5 Encoding

Object ใน MIB ถูกเข้ารหัสโดยการใช้ Basic Encoding Rule(BER)ซึ่งเกี่ยวข้องกับ ASN.1

2.5.3 Management Information Base (MIB) - II

จากมาตรฐานและการนำมากล่าวของ RFC1213 จะกล่าวถึง object ใน MIB-II ดังนี้

- System Group

Group นี้สำคัญมากเป็นที่ต้องการของทุกๆอุปกรณ์ซึ่งรวม configuration information ว่ามันเป็นอย่างไรและอยู่ที่ไหน และใครเรียกมันเมื่อเกิดสิ่งผิดพลาด

- Interface Group

The interface group กลุ่มนี้จะเก็บข้อมูลทั่วไปเกี่ยวกับ physical interface ของ entity อุปกรณ์ติดต่อกับภายนอกผ่าน interface hardware และ software MIB-II มี 32 types

- configuration information เช่น ประเภทของ interface , ความเร็ว

- สถานะของ interface
- สถิติความหนาแน่นการเข้าออกของข้อมูล
- การนับของชนิดของ error

- Address – Translation group

การขนส่งข้อมูลจะเป็นเส้นทางจะมีการเลือกเส้นทางที่ดีที่สุดไปสู่ปลายทางก่อน Datagram จะถูกส่งมันต้องถูกหุ้มด้วยเฟรม The header frame จะรวม address information ที่จำเป็นสำหรับเฟรมที่ขนส่ง

กระบวนการของการ Map IP Address ของ The next-hop ไป The lower layer frame Address เรียก Address Translation ลิสต์ของ network layer addresses และการติดต่อทาง LAN หรือ packet net addresses จะถูกเก็บไว้ในตาราง translation ที่ระบบหนึ่งๆ at group ให้การเข้าถึงการจัดการ application ที่ตาราง information นี้

ตาราง AT นิยามด้วย MIB-I การออกแบบโดยทั่วไปเป็นจุดมุ่งหมายเพื่อแปลงประเภทของ Network Layer Addresses IP, IPX อะไรก็ตามที่เป็นการติดต่อใน lower layer address นิยามด้วย MIB-II จะมีการพิจารณาเรื่องความผิดพลาดที่เกิดขึ้นด้วย การ Tanslation เป็นความจำเป็นของโปรโตคอล เช่น TCP/IP , NetWare IPX, DECnet และ AppleTalk แตกต่างกันในรายละเอียด ตาราง IP address จะแยกนิยามไว้ใน IP Group

- IP Group

ครอบคลุมการ configuration และการจัดการที่จำเป็นของ host และ router

- ตัวแปรที่ config
- สถิติความหนาแน่นของการเข้าออกข้อมูล
- การนับของ error ที่เกิด
- ตาราง config แต่ละ interface กับหนึ่ง IP address , subnet mask และค่าพารามิเตอร์ที่เกี่ยวข้องกับ IP อื่นๆ
- A new IP – specific address translation table
- ตาราง IP routing ซึ่งเคยถูกเลือกที่ The next – hop destination เมื่อ routing diagram

กลุ่มนี้จะเป็นระบบ TCP/IP ทั้งหมด

- ICMP Group

Internet Control Massage Protocol error message จะเป็นการส่ง Message จากหลายๆ Router หรือ Host อื่นๆไปยัง Host หนึ่ง ICMP จะส่งกลับไปที่ IP datagram source เพื่อรายงานปัญหาที่นับได้ ขณะที่พยายามจะส่ง Datagram ไปปลายทาง ICMP จะสนับสนุนการใช้ Query message icmp group จะตรงไปตรงมา มันจะนับทุกประเภทของการเข้าและออก icmp message กลุ่มนี้จะเป็นระบบ TCP/IP ทั้งหมด

ข้อมูลใน Icmp Group จะเป็นความสัมพันธ์การ Implementation และ Operation ของ ICMP ที่โหนดหนึ่ง

- The TCP Group

ข้อมูลใน TCP group จะเป็นความสัมพันธ์การ implementation และ operation ของ TCP ที่โหนดหนึ่ง

กลุ่มนี้จะจัดการ Config ค่าของ TCP เช่นขีดจำกัดสูงสุดของจำนวนการติดต่อ TCP ที่มีอยู่ที่สามารถจะ handle ได้ MIB จะเก็บบันทึกสถิติความหนาแน่นของข้อมูลที่เข้าที่ออกและ Error แต่บางทีเครื่องมือที่ใช้วินิจฉัยส่วนใหญ่คือตาราง Connection ซึ่งอนุญาตให้ Manager ดูรายชื่อของ Active TCP Sessions

- The UDP Group

ข้อมูลใน UDP Group จะเป็นความสัมพันธ์การ Implementation และ Operation ของ UDPที่โหนดหนึ่ง จะเป็นข้อมูลที่เกี่ยวกับ Datagrams ที่ส่งและได้รับ UPD Group จะรวม UDP Table ตารางนี้จะเก็บข้อมูลที่เกี่ยวกับ Entity's UDP end-point สำหรับแต่ละ UDP user ตารางจะเก็บ IP address และ Port สำหรับ User

กลุ่มนี้ Track ความหนาแน่นและ Error แต่ UDP เป็นโปรโตคอลอย่างง่ายจึงไม่มีค่าตัวแปรที่จำเป็นมากๆ

UDP application จะส่ง stand-alone massage ไปไม่มีการเก็บบันทึกข้อมูลที่ทำการติดต่อกับ Partner แต่จะพบมันได้โดยการฟังจาก partner ที่จะคุยได้โดย UDP service

- The Transmission Group

The transmission Group จะเป็นการจัดการ function ใหม่และไม่ควรจะเรียกว่าเป็นกลุ่มเพราะมันอยู่ในโหนดที่อยู่ในทรี MIBs ส่วนมากบรรจุหลายๆกลุ่ม ซึ่ง Apply ตามความแตกต่าง Transmission Technologies ที่เป็นสิ่งที่ติดมากับโหนดนี้

- The SNMP Group

เราจะคาดหวังว่า Management Station จะ Poll ระบบและถามสำหรับข้อมูลนี้ เรายังคาดหวังจะได้ Trap traffic ที่เกิดบน Network จากเวลาต่อเวลา ถ้า SNMP ต้องการตรวจทุกๆสิ่ง สามารถนับ Traffic และ Error ของมันเอง สามารถตรวจสอบจำนวนของข้อมูลที่เข้าและออก Gets, Set, Response และ นับ Error ต่างๆ เช่น พยายามอ่านตัวแปร Nonexistent

2.5.4 SNMP Formats

ข้อมูลที่แลกเปลี่ยนกันระหว่าง Management station และ Agent อยู่ในรูปแบบของ SNMP Message แต่ละ Message ประกอบด้วย Version number ที่บอกถึง version ของ SNMP ชื่อของ community ที่ใช้ในการแลกเปลี่ยนและหนึ่งในห้าชนิดของ Protocol Data Unit โครงสร้างนี้แสดงดังรูป

version	community	SNMP PDU

(a) SNMP message

PDU type	request-id	0	0	variable - bindings

(b) GetRequest PDU, GetNextRequest PDU, and SetRequest PDU

PDU type	request-id	error-status	error-index	variable - bindings

(c) GetResponse PDU

PDU type	enterprise	agent-addr	generic-trap	specific-trap	time-stamp	variable - bindings

(d) Trap PDU

name 1	value 1	name 2	value 2	. . .	name n	value n

(e) variable - bindings

รูปที่ 2.9 SNMP Formats

ตารางที่ 2.7 SNMP MESSAGE FIELDS

Fields	ความหมาย
Version	SNMP version (RFC1157)
Community	การจับคู่กันของ SNMP agent กับกลุ่มของ SNMP application entity
Request-id	ชี้ให้เห็นข้อแตกต่างระหว่างความต้องการให้แต่ละความต้องการIDที่ไม่ซ้ำกัน
Error-status	ชี้ให้เห็นถึงสิ่งที่ผิดปกติขณะกำลังประมวลผลความต้องการ เช่น noError(0) , tooBig(1) , NoSuchName(2) , badValue(3) , readOnly(4) , genErr(5)
Error-index	เมื่อ Error-status ไม่เท่ากับศูนย์ อาจจะให้ข้อมูลเพิ่มเติมโดยชี้ว่าตัวแปรในรายชื่อเป็นสาเหตุให้เกิดสิ่งผิดปกติ(ตัวแปร คือ instance ของ object ที่ต้องจัดการ)
Variablebindings	รายชื่อของชื่อตัวแปรและค่าที่เกี่ยวข้อง
Enterprise	ชนิดของ object ที่สร้าง trap ขึ้นอยู่กับ sysObjectID
Agent-addr	ที่อยู่ของ object ที่สร้าง trap
Generic-trap	ชนิดของ trap ทั่วไป
Specific-trap	trap code ที่ต้องกำหนด
Time-stamp	เวลาที่ผ่านไประหว่างการเริ่มต้นใหม่ครั้งสุดท้ายของ network entity และการสร้าง trap ซึ่งจะมีค่าของ sysUpTime

ในทางปฏิบัติ Authentication Service มีหน้าที่เพียงแค่ตรวจสอบชื่อ Community ที่มีสิทธิ์รับ Message จาก SNMP entity ที่เป็นตัวส่ง

2.5.5 SNMP PDUs

ชนิดของ PDU มี 5 ชนิด ที่ใช้ใน SNMP มีดังต่อไปนี้

1. GetRequest PDU

โดย PDU นี้จะส่งจาก management station ไปยัง agent และใช้เพื่อดึงค่าของ object instance ซึ่ง OID จะถูกเก็บไว้ใน PDU

2. GetNextRequest PDU

PDU นี้ถูกส่งจาก management station ไปยัง agent และถูกใช้เพื่อดึงค่าของ object instance ซึ่ง OID จะถูกแบ่งจาก OID ที่เก็บใน PDU

3. GetResponse PDU

PDU นี้จะส่งจาก agent ไปยัง management station ในการตอบสนองต่อ GetRequest PDU , GetNextRequest PDU หรือ SetRequest PDU และประกอบด้วยค่าของ object instance

4. SetRequest PDU

PDU นี้ถูกส่งจาก management station ไปยัง agent และถูกใช้เพื่อตั้งค่าของ object instance ซึ่ง OID และค่าจะเก็บไว้ใน PDU

5. Trap PDU

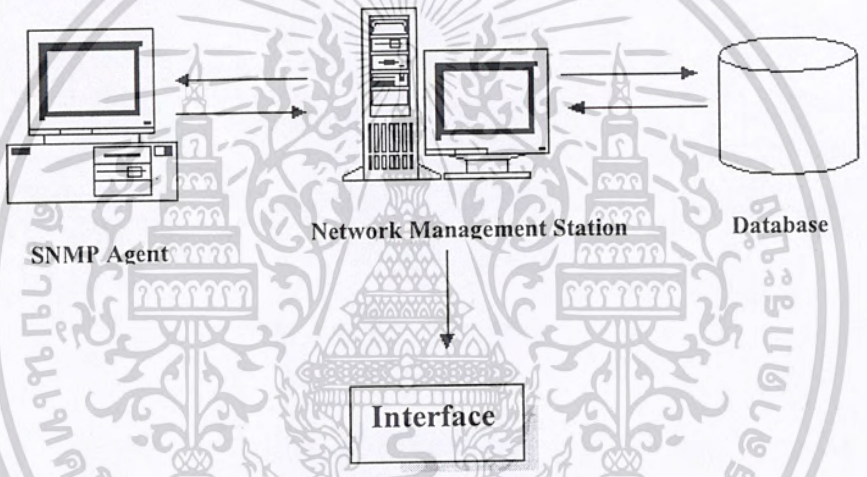
PDU นี้ถูกส่งจาก agent เพื่อเลือกกลุ่มของ management station เพื่อสังเกตการณ์เหตุการณ์ที่ผิดปกติ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่ 3
การออกแบบโปรแกรม

3.1 หลักการและการออกแบบระบบการทำงานของโปรแกรม

ระบบการทำงานของโปรแกรมเริ่มโดยเครื่อง Management Station ส่งคำสั่งเพื่อร้องขอข้อมูลจาก Agent ซึ่งเปรียบเสมือนเป็นตัวแทนของอุปกรณ์ในระบบเครือข่าย Agent ใดที่สนับสนุนการทำงานของโปรโตคอล SNMP ก็จะส่งข้อมูลกลับมาที่ Management Station หลังจากนั้น Management Station จะทำการเก็บข้อมูลที่ได้ไว้ในฐานข้อมูล เพื่อนำข้อมูลเหล่านี้มาใช้สร้างแผนที่เครือข่ายสำหรับอุปกรณ์ในเครือข่ายที่สนับสนุนการทำงานของโปรโตคอล SNMP เท่านั้น รายละเอียดขั้นตอนของโปรแกรมมีดังนี้



รูปที่ 3.1 ระบบการทำงานของโปรแกรม

3.1.1 การติดต่อระหว่าง Management Station กับ Agent

เป็นการทำงานส่วนแรกของโปรแกรมโดย Managemet Station ส่งคำสั่ง Poll ไปร้องขอข้อมูลจาก Agent ซึ่งการทำงานของโปรแกรมออกแบบให้มีการติดต่อกับ Agent ได้ 2 วิธี คือ

1. ระบุ IP Address ของ Agent โดยตรง เช่น IP Address 161.246.10.21
2. ระบุเป็นหมายเลขของเครือข่าย (Network Address) กรณีที่ต้องการส่งคำสั่ง Poll เก็บข้อมูลของอุปกรณ์ในเครือข่ายตามระดับชั้นของเครือข่าย แล้วโปรแกรมจะทำการวน Loop เพื่อส่งคำสั่ง Poll ไปยัง Agent ในลักษณะ Sequencial นอกจากนี้โปรแกรมกำหนดให้มีการ

ระบุ Subnet Address ของเครือข่ายด้วย โดยค่าของ Subnet Address ขึ้นอยู่กับการแบ่งเครือข่ายย่อยๆของแต่ละองค์กร เช่น

- ต้องการเก็บข้อมูลของเครือข่ายในคลาส B ซึ่งหมายเลข IP Address มีลักษณะเป็น net.net.host.host ค่า IP Address เริ่มตั้งแต่ 128-191 เช่น161.246.10.45 แทนเครือข่ายเลขที่ 161.246 และหมายเลขเครื่องคือ 10.45 สำหรับคลาส B นี้มี Subnet Address เป็น 255.255.0.0 ถ้ามีการกำหนดในลักษณะนี้โปรแกรมจะทำการส่งคำสั่ง Poll ไปยังแต่ละเครือข่ายเริ่มตั้งแต่ 161.246.0.0 จนถึง 161.246.255.0 ซึ่งแต่ละเครือข่ายโปรแกรมจะส่งคำสั่งไปยังเครื่องลูกข่ายเริ่มต้นตั้งแต่ 1 ถึง 254 (หมายเลขเครื่อง 0 แทนเครือข่ายและหมายเลขเครื่อง 255 แทนการ Broadcast)

- ต้องการเก็บข้อมูลของเครือข่ายในคลาส C ซึ่งหมายเลข IP Address มีลักษณะเป็น net.net.net.host ค่า IP Address เริ่มตั้งแต่ 192-223 เช่น 197.103.210.2 แทนเครือข่ายเลขที่ 197.103.210 และหมายเลขเครื่องคือ 2 และสำหรับคลาสนี้ถ้ากำหนด Subnet Address เป็น 255.255.255.0 โปรแกรมจะส่งคำสั่งไปยังเครือข่าย 197.103.210 โดยวนลูป Poll ตั้งแต่เครื่องลูกข่ายเป็น 1 จนถึง 254 (หมายเลขเครื่อง 0 แทนเครือข่าย และหมายเลขเครื่อง 255 แทน Boradcast Address ของแต่ละเครือข่าย)

3.1.1.1 การดึงค่าข้อมูลจาก Agent ในระบบเครือข่าย

สำหรับ Agent ที่สนับสนุนการทำงานของ SNMP โปรโตคอลจะมีฐานข้อมูลในรูปของ Management Information Base หรือ MIB ซึ่งมีโครงสร้างที่เก็บแต่ละ Object เอาไว้ โปรแกรมจะทำการส่งคำสั่งไปดึงค่าข้อมูลใน System Group , Interface Group และ IP Group จาก MIB โดยมีรายละเอียดของแต่ละ Group ดังนี้

- System Group : ดึงค่าข้อมูลเกี่ยวกับ Information ทั่วไปของ Agent นั้น ประกอบด้วย

1. sysDescr : รายละเอียดของ Agent เช่น Hardware , OS
2. sysObjectID : เป็นหมายเลขแทนผู้ผลิต
3. sysUpTime : เวลาตั้งแต่ส่วนของการจัดการเครือข่ายกลับไปเริ่มต้นใหม่ครั้งสุดท้าย
4. sysContact : บุคคลที่รับผิดชอบในการจัดการระบบเครือข่ายย่อย
5. sysName : ชื่อของระบบเครือข่ายย่อยที่จัดการ
6. sysLocation : ตำแหน่งทางกายภาพของระบบเครือข่ายย่อยที่จัดการ

- Interface Group : เก็บข้อมูลเกี่ยวกับ Interface ของ Agent นั้น เพื่อใช้ในการวิเคราะห์การไหลของข้อมูลที่เข้าและออกแต่ละ Interface ของ Agent

- IP Group : เก็บข้อมูลเกี่ยวกับ Gateway , Subnet Mask ที่เกาะอยู่กับแต่ละ Interface ของ Router ที่สนใจ

3.1.1.2 คำสั่งที่ Management Station ใช้ในการดึงค่าข้อมูลจาก Agent

คำสั่งที่ Management Station ส่งไปยัง Agent เพื่อใช้ในการดึงค่าข้อมูล คือคำสั่ง AgentGetNextRequest เป็นคำสั่งทีใช้ในการดึงค่าข้อมูลทั้งที่เป็นข้อมูลเดี่ยวและข้อมูลที่เป็นชุด ซึ่งพารามิเตอร์ที่ระบุประกอบด้วย Object Identifiler, Community Name, IP Address, หมายเลข Port

- AgentGetNextRequest (Object Identifiler , Community Name , IP Address , Port)

เช่น AgentGetNextRequest (1.3.6.1.2.1.1., public, 161.246.10.21, 161) เป็นการร้องขอข้อมูลจากตาราง System Group ใน MIB ของ Agent ที่มี Community name เป็น public มีหมายเลข IP Address เป็น 161.246.10.21 และ หมายเลข Port ของ Agent เป็น 161

- AgentGetNextRequest (1.3.6.1.2.1.2., public, 161.246.25.15, 161)

เป็นการร้องขอข้อมูลจากตาราง Interface Group ใน MIB ของ Agent มี Community name เป็น public มีหมายเลข IP Address เป็น 161.246.25.15 และ หมายเลข Port ของ Agent เป็น 161

ในแต่ละครั้งที่ Management Station ส่งคำสั่งไปดึงข้อมูลจาก Agent ถ้า Agent นั้นสนับสนุนการทำงานของโปรโตคอล SNMP ก็จะส่งข้อมูลมายัง Management Station แต่ถ้าไม่ได้รับการตอบกลับจาก Agent ภายในระยะเวลาที่กำหนด Management Station จะส่งคำสั่ง Polling ไปยัง Agent นั้นใหม่จนครบ 3 ครั้ง ถ้ายังไม่ได้รับการตอบกลับจะสรุปว่า Agent นั้นไม่สนับสนุนการทำงานของโปรโตคอล SNMP ณ ช่วงเวลานั้น

3.1.2 การเก็บค่าที่ได้ลงฐานข้อมูล

Management Station เก็บค่าที่ได้จากตาราง MIB II ของ Agent ลงฐานข้อมูล บอกว่าการ poll เก็บ Information

3.1.3 Management Station แสดงผลที่ Interface

Management Station นำค่าจากฐานข้อมูลมาสร้างเป็นแผนที่เครือข่าย และสามารถจัดเก็บภาพไว้ได้

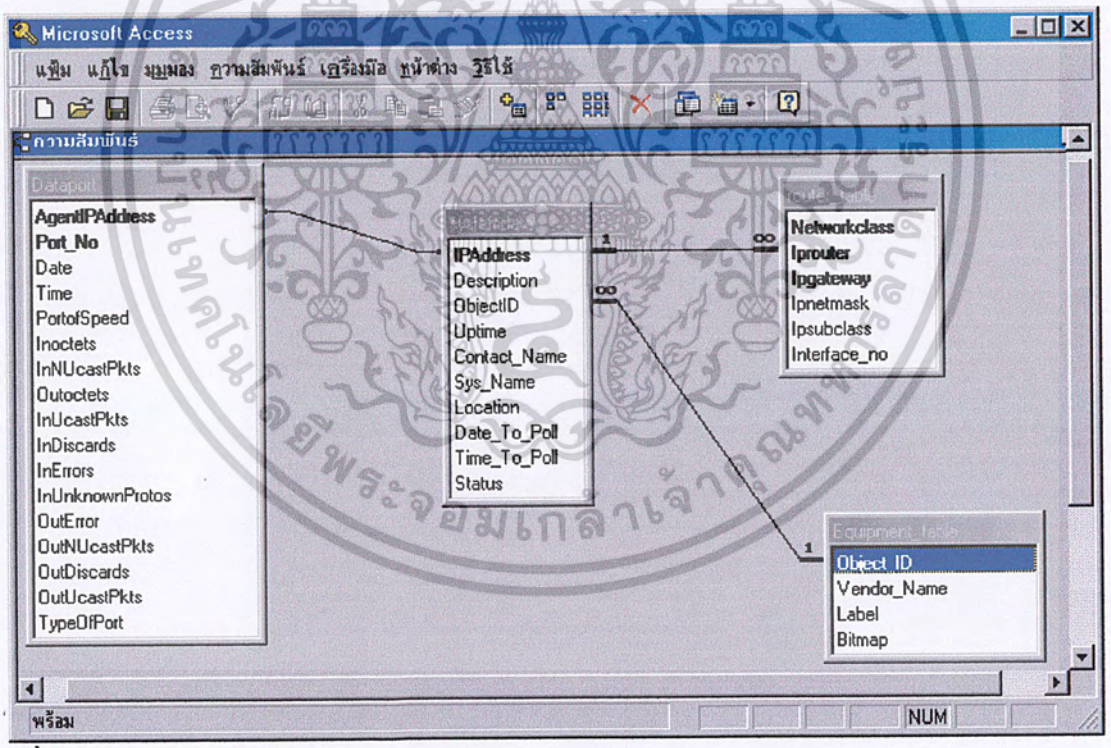
3.2 การออกแบบฐานข้อมูล

เนื่องจากโปรแกรมใช้ Application ของ Borland Delphi เป็นเครื่องมือในการพัฒนาโปรแกรม จึงใช้โปรแกรมสำหรับจัดการฐานข้อมูลของ Delphi ดังนี้

- ใช้ฐานข้อมูลมาตรฐานของ Delphi เป็นตัวเก็บข้อมูล คือ Paradox ซึ่งเป็นฐานข้อมูลภายในของ Delphi(ฐานข้อมูลภายในหมายถึง ฐานข้อมูลที่อยู่ในรูปแบบของไฟล์สำหรับเก็บข้อมูล ปกติจะอยู่ภายในเครื่องเดียวกันกับ Application ที่ใช้งาน)
- ใช้การเชื่อมต่อกับฐานข้อมูลผ่านระบบ Borland Database Engine (BDE) โดยกำหนดคุณสมบัติการเชื่อมต่อเป็น

Type	=	STANDARD
Default Driver	=	Patadox
Path	=	D:\Project Network\Network
Enable BCD	=	False

- สร้างตารางโดยใช้ Database Destop โปรแกรมประกอบด้วยตารางทั้งหมด 4 ตาราง โดยมี 3 ตารางที่นำมาใช้ในการสร้างแผนที่เครือข่าย คือ ตารางSystemData.db, Router_table.db และ Equipment_table.db และอีก 1 ตารางสำหรับเก็บข้อมูลเข้าและออกแต่ละ Interface ของ Agent



รูปที่ 3.2 แสดงความสัมพันธ์ของตาราง

ตารางที่3.1 แสดงรายละเอียดของ Agent : SystemData.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	IPAddress	IP Address ของ Agent	Char[15]	P.K.
2	Description	รายละเอียดของ Agent	Char[200]	
3	ObjectID	Object ID ของผู้ผลิต	Char[50]	
4	Uptime	ระยะเวลาการทำงานของระบบ	Char[10]	
5	Contact_Name	บุคคลที่รับผิดชอบระบบ	Char[100]	
6	Sys_Name	ชื่อระบบเครือข่ายย่อยที่จัดการ	Char[100]	
7	Location	ตำแหน่งทางกายภาพของระบบ	Char[100]	
8	Date_To_Poll	วันที่ทำการดึงค่า	Date	
9	Time_To_Poll	เวลาที่ทำการดึงค่า	Time	
10	Status	สถานะปัจจุบันของ Agent	Char[3]	

ตารางที่ 3.2 เก็บข้อมูลของเครือข่ายที่เชื่อมต่อ Router ของ Network Address : Router_table.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	Networkclass	Network Address ของ Agent	Char[15]	C.K.
2	Iprouter	IP Address ของ Router ใน Network Address	Char[15]	C.K.
3	Ipgateway	หมายเลข Gateway	Char[15]	C.K.
4	Ipnetmask	Subnet Mask	Char[15]	
5	Ipsubclass	เป็น Network Address ที่ได้จากการ And ทาง Logic ระหว่าง Ipgateway และ Ipnetmask	Char[15]	
6	Interface_no		Char[15]	

ตารางที่3.3 แสดงชนิดของอุปกรณ์ในระบบเครือข่าย : Equipment_table.db

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	Object_ID	Object ID ของผู้ผลิต	Char[50]	P.K.
2	Vendor_Name	ชื่อของผู้ผลิต	Char[100]	
3	Label	ชื่อที่ใช้แทนอุปกรณ์	Char[50]	
4	Bitmap	ชื่อ File ที่แทนอุปกรณ์	Char[50]	

ตารางที่3.4 เก็บข้อมูลเข้าและออกแต่ละ Interface ของ Agent : Dataport.db

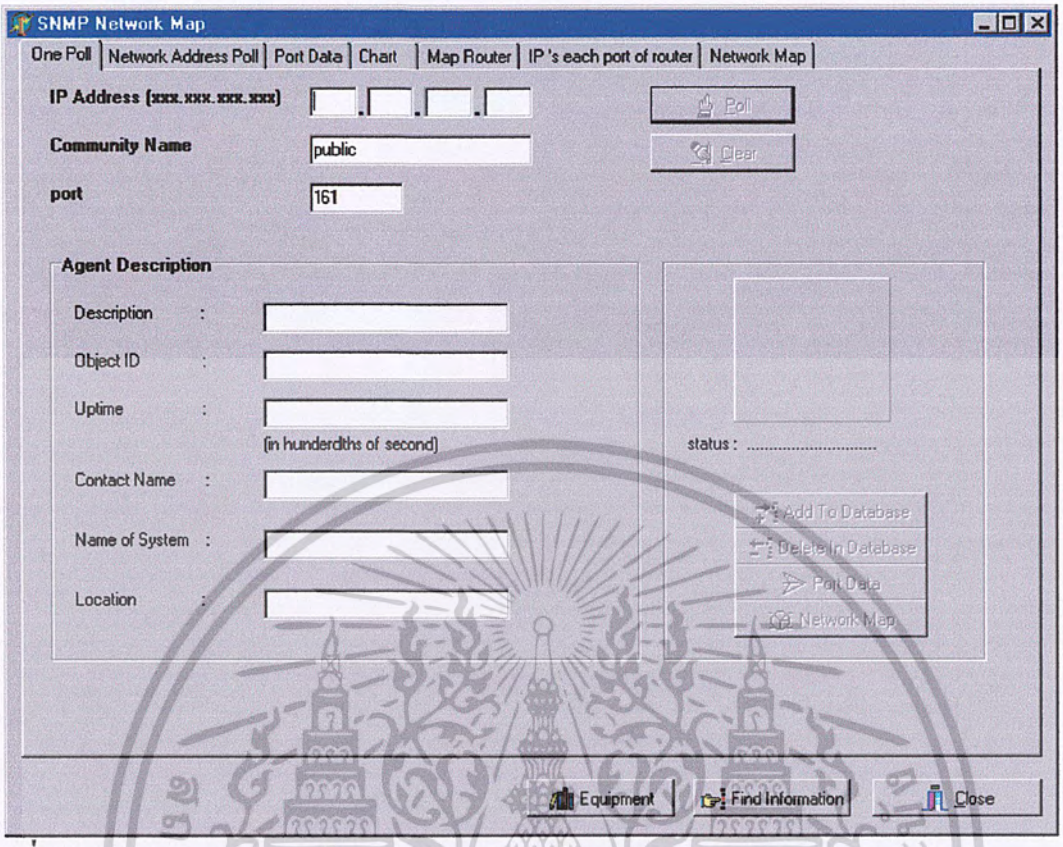
ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
1	AgentIPAddress	IP Address ของ Agent	Char[15]	C.K.
2	Port_No	หมายเลข Interface	Number	C.K.
3	Date	วันที่ทำการดึงค่า	Date	
4	Time	เวลาที่ทำการดึงค่า	Time	
5	PortofSpeed	ความเร็วในการรับส่งข้อมูลของ Interface นั้น	Char[50]	
6	Inoctets	จำนวนรวมของข้อมูลที่รับเข้ามาในแต่ละ Interface	Char[20]	
7	InNUcastPkts	จำนวนรวมของ Packet ที่ Nonunicast ไป Subnetwork ที่ Protocol layer ที่สูง	Char[20]	
8	Outoctets	จำนวนรวมของข้อมูลที่ส่งออกไปในแต่ละ Interface	Char[20]	
9	InUcastPkts	จำนวนรวมของ Packet ที่ Unicast ไป Subnetwork ที่ Protocol layer ที่สูงกว่า	Char[20]	
10	InDiscards	จำนวนของ Inbound Packets ที่ทิ้งถึงแม้ไม่มี Error เพื่อป้องกันไม่ให้ส่งไป Protocol layer ที่สูงกว่า	Char[20]	
11	InErrors	จำนวนของ Error ที่เกิดขึ้นเมื่อรับ Packet เข้ามาในแต่ละ Interface	Char[20]	
12	InUnknownProtos	จำนวนของ Inbound Packet ที่ถูกทิ้งเพราะไม่รู้หรือไม่สนับสนุนโปรโตคอล	Char[20]	

ลำดับ	ชื่อคอลัมน์	รายละเอียด	ชนิดข้อมูล	คีย์
13	OutError	จำนวน Error ที่เกิดขึ้นเมื่อส่ง Packet ออกไปในแต่ละ Interface	Char[20]	
14	OutNUcastPkts	จำนวนรวมของ Packet ซึ่ง Higher-level protocols ร้องขอที่ nonunicast ไป address ของ subnetwork รวมถึง	Char[20]	
15	OutDiscards	จำนวนรวมของ Outbound Packets ที่ทิ้ง ถึงแม้ไม่มี Error เพื่อป้องกันไม่ให้ส่งไป Protocol layer ที่สูงกว่า	Char[20]	
16	OutUcastPkts	จำนวนรวมของ Packets ซึ่ง Higher-level protocols ร้องขอที่ Unicast ไป Address ของ subnetwork รวมถึง Packets ที่ทิ้งหรือไม่ได้ส่งด้วย	Char[20]	
17	TypeOfPort	ชนิดของ Port	Char[50]	

3.3 การออกแบบส่วนที่ติดต่อกับผู้ใช้

การออกแบบส่วนที่ติดต่อกับผู้ใช้มีรายละเอียดดังต่อไปนี้

3.3.1 จอภาพแสดงการ Poll โดยระบุ IP Address



รูปที่ 3.3 แสดงการออกแบบหน้าจอสำหรับการ Poll โดยกำหนด IP Address

หน้าจอประกอบด้วยช่องสำหรับใส่ IP Address , Commuinty Name และหมายเลขของ Port เริ่มแรกโปรแกรมกำหนดค่า Community Name เป็น Public และกำหนดหมายเลข Port เป็น 161 (สำหรับ Port ของ SNMP Agent) ซึ่งทั้งสองค่ากำหนดเป็น Deflaut ไว้ เมื่อคลิกปุ่ม Poll โปรแกรมจะเริ่มส่งคำสั่งไปดึงข้อมูลจาก Agent ซึ่งเป็นการดึงข้อมูลใน System Group ประกอบด้วย Fields

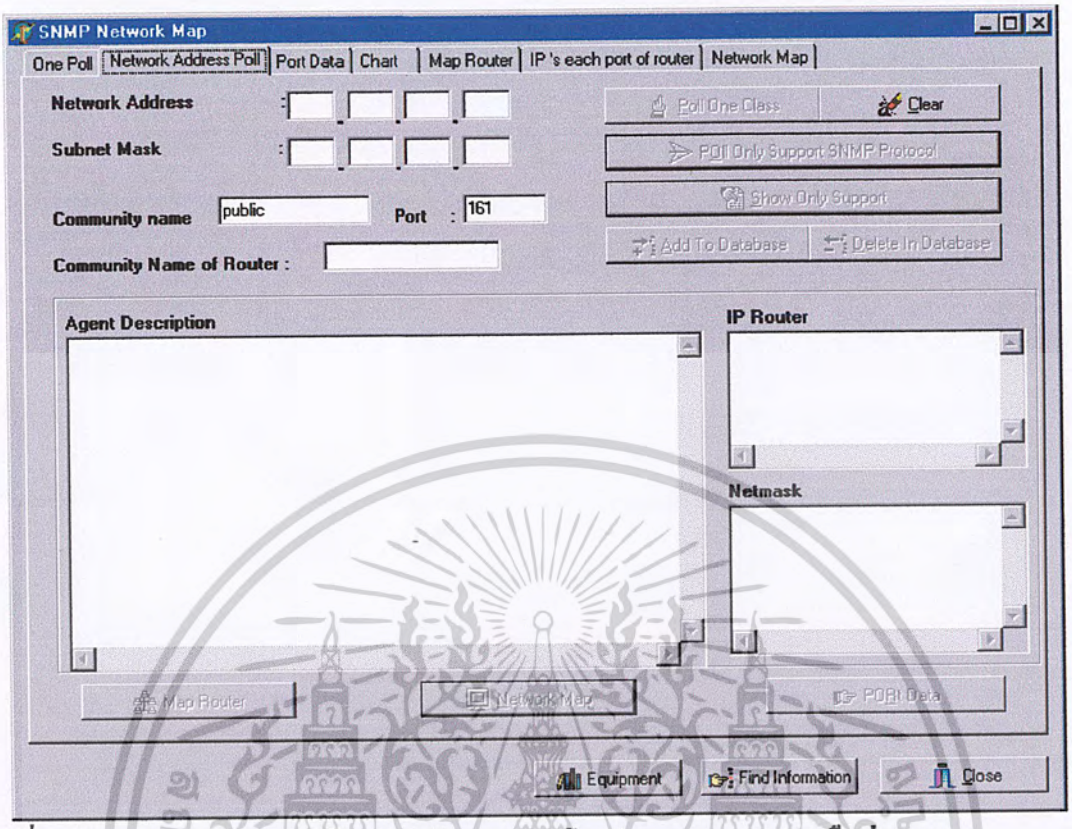
- Description ใช้ Object sysDescr : Object Identifier เป็น 1.3.6.1.2.1.1
- ObjectID ใช้ Object sysObjectID : Object Identifier เป็น 1.3.6.1.2.1.2
- Uptime ใช้ Object sysUptime : Object Identifier เป็น 1.3.6.1.2.1.3
- Contact Name ใช้ Object sysContact : Object Identifier เป็น 1.3.6.1.2.1.4
- Name of System ใช้ Object sysName : Object Identifier เป็น 1.3.6.1.2.1.5
- Location ใช้ Object sysLocation : Object Identifier เป็น 1.3.6.1.2.1.6

หลังจาก Management Station ส่งคำสั่งไปยัง Agent แล้ว Output ที่ได้รับจะเกิดขึ้นได้ 2 กรณีคือ

1. กรณีที่ Agent สนับสนุนการทำงานของ SNMP โปรโตคอล Agent ก็จะส่งข้อมูลกลับมายัง Management Station ภายในช่วงเวลาที่กำหนด โปรแกรมจะแสดงสถานะของการ Support SNMP Protocol ขึ้นที่หน้าจอ และรับค่าของ Object ID ที่ได้จากการ Poll ไว้เพื่อนำค่าที่ได้ไปตรวจสอบหาชนิดของอุปกรณ์ ในตาราง Equipment_table เพื่อใช้ในการแสดงภาพของ Agent นอกจากนี้โปรแกรมกำหนดให้มีการค้นหา IP Address ของAgent ตัวนี้ในตาราง SystemData.db เพื่อตรวจสอบสถานะในอดีต ถ้า Agent ตัวนี้มีอยู่ในฐานข้อมูล และสถานะในอดีตเคยสนับสนุนการทำงานของ SNMP โปรโตคอล โปรแกรมแสดงรูปของอุปกรณ์โดยมีกรอบของรูปเป็นสีเขียว ซึ่งภาพที่ได้จะเหมือนกับกรณีที่ตรวจสอบ Agent ในฐานข้อมูลแล้วไม่มีอยู่แต่สถานะในปัจจุบันสนับสนุนการทำงานของ SNMP โปรโตคอลก็จะแสดงกรอบของรูปที่เป็นสีเขียวเช่นเดียวกัน สำหรับกรณีที่ตรวจสอบค่าของ ObjectID แล้วไม่มีอยู่ในฐานข้อมูล โปรแกรมจะแสดงรูปที่เป็น Default ของอุปกรณ์ขึ้นมา และแสดงหน้าจอของการรับอุปกรณ์ชนิดนั้นเข้ามาไว้ในฐานข้อมูล นอกจากนี้ถ้า Agent สนับสนุนการทำงานของ SNMP โปรโตคอล โปรแกรมกำหนดให้มีการเข้าไปดูข้อมูลที่ไหลเข้าและออกจากแต่ละ Interface ของ Agent ได้ และในการดูแผนที่เครือข่ายของ Agent ตัวนี้โปรแกรมจะทำการดึงค่าของ IP Address ทั้งหมดที่อยู่ในฐานข้อมูล ซึ่ง IP Address เหล่านี้ต้องมีสถานะ Support SNMP Protocol โดยเลือกจาก Network Address ของ Agent ที่ระบุ

2. กรณีที่ Agent ไม่สนับสนุนการทำงานของโปรโตคอล SNMP ครั้งแรก Agent ก็อาจจะไม่ส่งผลลัพธ์ใดๆกลับมายัง Management Station ภายในช่วงเวลาที่โปรแกรมกำหนดไว้ เมื่อโปรแกรมรับรู้ในลักษณะนี้โปรแกรมก็จะส่งคำสั่งในการ Poll กลับไปยัง Agent ตัวนั้นใหม่ โดยโปรแกรมกำหนดจำนวนครั้งที่ส่งคำสั่ง Polling ไปสำหรับ Agent 1 ตัวไม่เกิน 3 ครั้ง ถ้าครบ 3 ครั้งแล้วโปรแกรมยังไม่ได้รับคำตอบกลับมาภายในช่วงเวลาที่กำหนด โปรแกรมจะถือว่า Agent ไม่สนับสนุนการทำงานของ SNMP Protocol ณ ช่วงเวลานั้น และโปรแกรมแสดงสถานะของการไม่ Support ขึ้นที่หน้า

3.3.2 จอภาพแสดงการ Poll โดยกำหนด Network Address และ Subnet Address



รูปที่ 3.4 แสดงการออกแบบหน้าจอสำหรับการ Poll โดยกำหนดหมายเลขเครือข่ายและ Subnetmask

หน้าจอนี้มีการออกแบบสำหรับกรณีที่ Management Station ต้องการส่งคำสั่ง Poll ไปยัง Agent ทุกตัวจากเครือข่ายที่ต้องการ โดยมีช่องสำหรับใส่ Network Address , Subnet Address , Community Name , Port และ Community Name Of Router โปรแกรมจะนำตัวเลข Network Address และ Subnet Address มาเข้า Operator And ทาง Logic เพื่อตรวจสอบว่าเป็นเครือข่ายในคลาสใด และ Management Station จะส่งคำสั่ง Polling ไปยัง Agent ทุกตัวในเครือข่ายนั้น โดยส่งคำสั่งในลักษณะ Sequencial คือเรียงลำดับก่อนหลัง คำสั่งที่ส่งไปเป็นการ Polling ขอข้อมูลในตาราง System Group ของฐานข้อมูล MIB ซึ่งประกอบด้วย Fields

- Description ใช้ Object sysDescr : Object Identifier เป็น 1.3.6.1.2.1.1
- ObjectID ใช้ Object sysObjectID : Object Identifier เป็น 1.3.6.1.2.1.2
- Uptime ใช้ Object sysUptime : Object Identifier เป็น 1.3.6.1.2.1.3
- Contact Name ใช้ Object sysContact : Object Identifier เป็น 1.3.6.1.2.1.4
- Name of System ใช้ Object sysName : Object Identifier เป็น 1.3.6.1.2.1.5
- Location ใช้ Object sysLocation : Object Identifier เป็น 1.3.6.1.2.1.6

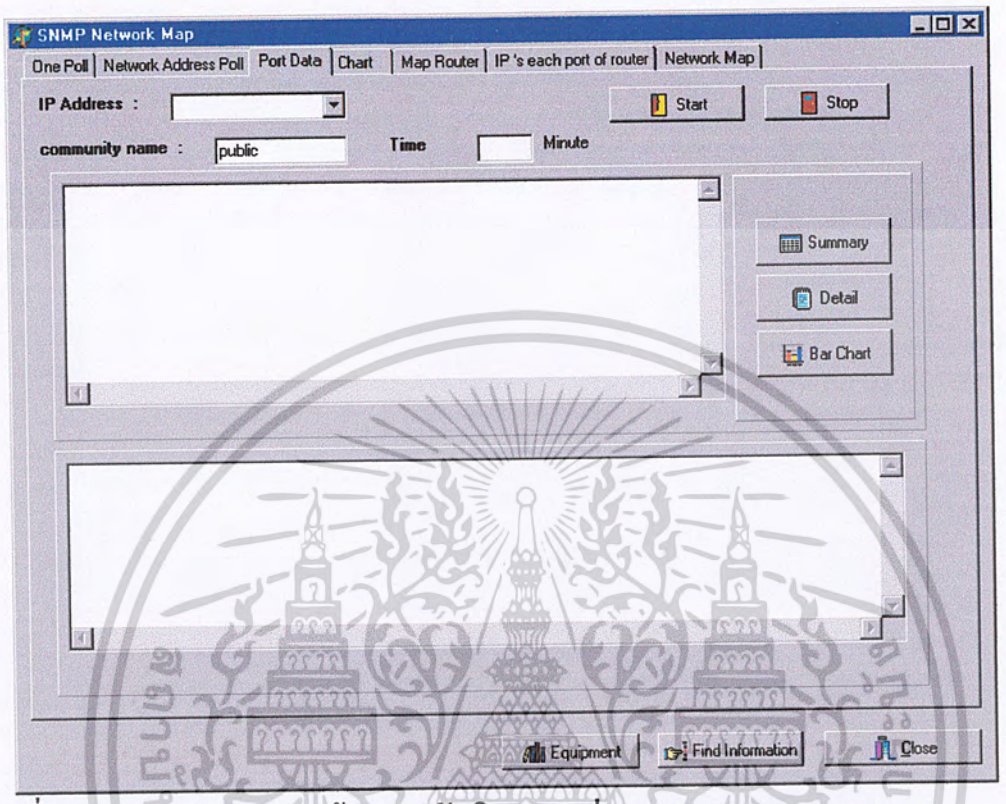
ในครั้งแรกที่ Management Station ส่งคำสั่ง Poll ไปยัง Agent จากเครือข่ายที่กำหนด จะใช้ Community Name เป็น Public (เป็นค่า Community Name ที่เป็น well-known) กรณีที่ไม่ได้รับการตอบรับจาก Agent ภายในช่วงเวลาที่กำหนด โปรแกรมก็จะส่งคำสั่งไปใหม่จนครบ 3 ครั้ง แต่ถ้าหาก Management Station ยังไม่ได้รับการตอบรับจาก Agent อีกโปรแกรมจะส่งคำสั่งกลับไปยัง Agent ใหม่โดยใช้ Community Name ของ Router ซึ่งเปรียบเสมือนเป็น Password เพื่อใช้ในการสืบค้นและรวบรวมข้อมูลของ Router และเครือข่ายย่อยทั้งหมดที่เชื่อมต่อกับ Router นั้น เพื่อใช้สร้างแผนที่เครือข่ายซึ่งอยู่ในรูปของการเชื่อมต่อระหว่างเครือข่ายกับแต่ละ Interface ของ Router โปรแกรมจะเก็บค่า IP Address ที่เป็น Router ไว้ Management Station จะส่งคำสั่ง Polling ไปยัง IP Address เหล่านี้เพื่อเก็บค่าของเครือข่ายที่เชื่อมต่ออยู่ พารามิเตอร์บน MIB ที่ถูกเลือกใช้ อยู่ใน IP Group ประกอบด้วย

- ipAdEntIfIndex เป็นค่าที่ชี้ลักษณะเฉพาะของแต่ละ Interface มีค่า Object Identifier เป็น 1.3.6.1.2.1.4.20.1.2
- ipAdEntNetMask เป็นค่า Subnet Mask และ IpGateway ที่มีส่วนเกี่ยวข้องกับแต่ละ Interface ของ Router มีค่า Object Identifier เป็น 1.3.6.1.2.1.4.20.1.3 เมื่อนำหมายเลข IPAddress และ Netmask มาคำนวณจะทำให้ได้หมายเลขไอพีของเครือข่ายย่อยที่ Router ได้เชื่อมต่ออยู่

หลังจากเสร็จสิ้นกระบวนการสืบค้นเครือข่ายแล้วข้อมูลที่ได้อยู่ในรูปของความสัมพันธ์ระหว่าง Router กับหมายเลขเครือข่ายย่อย ซึ่งสามารถนำมาสร้างเป็นแผนที่เครือข่ายที่แสดงความสัมพันธ์ระหว่าง Router กับเครือข่ายย่อยในแต่ละ Interface โดยแผนที่เครือข่ายที่ถูกสร้างขึ้นจะเป็นภาพ Bitmap ที่มีการเก็บข้อมูลของตำแหน่งพิกัดบนภาพของ Router และเครือข่ายย่อย แผนที่เครือข่ายและข้อมูลจะถูก Application บาง Application นำไปประมวลผลในลำดับต่อไป

หน้าจอนี้โปรแกรมออกแบบให้มีส่วนของการส่งคำสั่ง Polling ไปเฉพาะ Agent ที่สนับสนุนการทำงานของ SNMP โปรโตคอล โปรแกรมใช้วิธีการดึง IP Address ในฐานข้อมูลเฉพาะ IP Address ที่มีสถานะสนับสนุนการทำงานของ SNMP โปรโตคอลจากเครือข่ายที่กำหนดเท่านั้น แล้วส่งคำสั่ง Polling ไปเก็บข้อมูลใหม่ ซึ่งหลักการทำงานจะเหมือนกับการ Polling ปกติ

3.3.3 จอภาพแสดงข้อมูลในแต่ละ Port



รูปที่ 3.5 แสดงการออกแบบหน้าจอสำหรับพิจารณาแต่ละ Interface ของ Agent

หน้าจอนี้โปรแกรมออกแบบมาเพื่อตรวจสอบข้อมูลที่ไหลเข้าและออกจากแต่ละ Interface ของ Agent ณ ช่วงเวลาหนึ่ง โปรแกรมกำหนดให้มีช่องสำหรับใส่ IP Address ของ Agent ที่ต้องการตรวจสอบ และกำหนด ให้มีการใส่ Community Name และเวลาที่จะให้มีการเก็บค่าข้อมูล ส่วนของการพิจารณาการไหลของข้อมูล โปรแกรมใช้การส่งคำสั่ง Polling ไปเก็บข้อมูล จาก Interface Group ของ MIB ประกอบด้วย

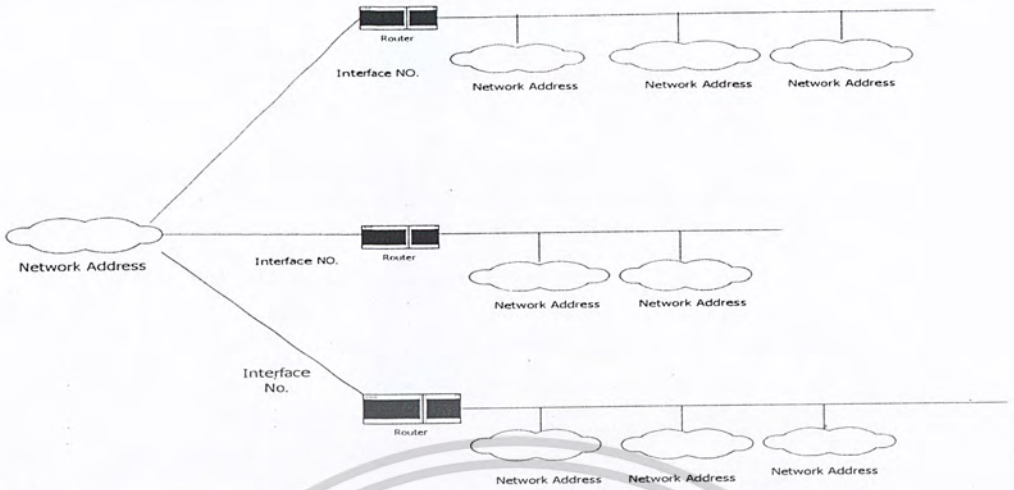
- Port_no : ค่าที่ชี้เฉพาะสำหรับแต่ละ Interface (หมายถึงหมายเลขของ Port) ใช้ Object ifIndex
- PortOfspeed : อัตราการส่งข้อมูลของแต่ละ Interface ใช้ Object ifSpeed
- TypeOfPort : ข้อมูลเกี่ยวกับ Interface รวมทั้งชื่อผู้ผลิต ชื่อผลิตภัณฑ์ และเวอร์ชั่นของแต่ละ Interfaceใช้ Object ifDescr
- Inoctets : จำนวนรวมของข้อมูลที่รับเข้ามาในแต่ละ Interface ใช้ Object IfInoctets
- Outoctets :จำนวนรวมของข้อมูลที่ส่งออกไปในแต่ละ Interface ใช้ Object ifOutoctets

- InErrors : จำนวน Error ที่เกิดขึ้นเมื่อรับ Packet เข้ามาในแต่ละ Interfaceใช้ Object ifInErrors
- OutErrors : จำนวน Error ที่เกิดขึ้นเมื่อส่ง Packet ออกไปในแต่ละ Interface ใช้ Object ifOutErrors
- InUcastPkts : จำนวนของ Packet ที่ Unicast ไป Subnetwork ที่ Protocol Layer ที่สูงกว่า ใช้ Object ifInUcastPkts
- InNUcastPkts : จำนวนของ Packet ที่ Nonunicast ไป Subnetwork ที่ Protocol Layer ที่สูงกว่า ใช้ Object ifNUcastPkts
- InDiscards : จำนวนของ Inbound Packet ที่ทิ้งถึงแม้ไม่มี Error เพื่อป้องกันไม่ให้ส่งไป Protocol Layer ที่สูงกว่า ใช้ Object ifDiscards
- OutDiscards: จำนวนของ Outbound Packet ที่ทิ้งถึงแม้ไม่มี Error เพื่อป้องกันไม่ให้ส่งไป Protocol Layer ที่สูงกว่า ใช้ Object ifOutDiscards
- OutUcastPkts : จำนวนรวมของ Packet ซึ่ง Higher-level Protocol ร้องขอเพื่อ Unicast ไป Address ของ Subnetwork รวมถึง Packet ที่ทิ้งหรือไม่ได้ส่งด้วย ใช้ Object ifOutUcastPkts
- OutNUcastPkts :จำนวนรวมของ Packet ซึ่ง Higher-level Protocol ร้องขอเพื่อNonunicast ไป Address ของ Subnetwork รวมถึง Packet ที่ทิ้งหรือไม่ได้ส่งด้วย ใช้ Object ifOutNUcastPkts
- InUnknownProtos:จำนวนของ Inbound Packets ที่ถูกทิ้งเพราะไม่รู้หรือไม่สนับสนุนโปรโตคอล ใช้ Object ifInUnknowProtos

3.3.4 แผนภูมิแท่งแสดงจำนวนข้อมูลที่ผ่านแต่ละ Interface ของ Agent

จากการที่โปรแกรมเก็บข้อมูลประเภทต่างๆที่ไหลเข้าและออกจากแต่ละ Port ของ Agent เอาไว้ โปรแกรมจะนำข้อมูลเหล่านี้มาสร้างเป็นแผนภูมิแท่งแสดงการไหลเข้าและออกของข้อมูลในแต่ละ Port โดยกราฟแต่ละแท่งแทนแต่ละ Port ของ Agent ซึ่งประกอบด้วยส่วนของข้อมูลเข้าและข้อมูลที่ออกจาก port โดยแสดงอยู่ในแท่งเดียวกัน

3.3.5 ภาพแสดงแผนที่ของเครือข่าย (แบบที่1)



รูปที่ 3.6 แสดงการออกแบบรูปที่ได้จากการ Poll โดยใช้หมายเลขเครือข่าย

สำหรับการสืบค้นเครือข่าย เมื่อมีการกำหนดเครือข่ายที่ต้องการแล้ว โปรแกรมส่งคำสั่ง Polling ไปเก็บข้อมูลที่จะนำมาเป็นส่วนประกอบในการสร้างแผนที่เครือข่ายย่อย เมื่อส่งคำสั่งไปยัง Agent ของเครือข่ายที่ต้องการแล้ว คำตอบที่ได้กลับมาจะทราบว่า IP Address ใดในเครือข่ายที่เป็น Router บ้าง ซึ่งเป็นจุดเริ่มต้นที่ใช้ในการวาดแผนที่ โปรแกรมออกแบบขั้นตอนสำหรับการสร้างแผนที่ดังนี้ คือ

1. Management Station ส่งคำสั่ง Polling ไปยัง Agent โดยใช้ Community Name ของ Router
2. เก็บ IP Address ที่เป็น Router ในเครือข่ายไว้
3. ส่งคำสั่ง GetNextRequest ไปยัง Router โดยใช้ ObjectID ipAdEntIfIndex เพื่อตรวจสอบว่าแต่ละ Port ของ Router มีค่า Index เป็นอะไร และแต่ละ Index มี IPGateway ใดอยู่
4. ส่งคำสั่ง GetNextRequest ไปยัง Router โดยใช้ ObjectID ipAdEntNetMask เพื่อตรวจสอบว่าแต่ละ Port ของ Router มี IP Gateway เป็นอะไร และมี Subnet Mask เป็นเท่าไร
5. เก็บข้อมูลไว้ในฐานข้อมูล เพื่อนำมาสร้างเป็นแผนที่เครือข่ายที่แสดงความสัมพันธ์ระหว่าง Router กับเครือข่ายย่อย รูปที่ได้จะแสดงให้เห็นว่า Router ที่มีหมายเลข IP Address เป็นค่าหนึ่ง อยู่ในเครือข่ายใด และแต่ละ Interface ของ Router มีเครือข่ายใดเชื่อมต่ออยู่บ้าง บาง Interface อาจจะมีเครือข่ายเชื่อมต่อกันอยู่มากกว่า 1 เครือข่าย รูปที่ใช้แทนเครือข่ายที่เชื่อมต่ออยู่ในแต่ละ Interface ของ Router แสดงรูปเหมือนกัน กรณีที่ใน 1 วงแทนเครือข่ายหลายเครือข่ายจะมี Network Address ของแต่ละเครือข่ายระบุไว้ด้วย และแต่ละ Interface ของ

Router จะมีหมายเลขของ Interface number กำกับไว้ ภาพที่ได้เป็นภาพ Bitmap ที่มีการเก็บข้อมูลของตำแหน่งพิกัดบนภาพของ Router และเครือข่ายย่อย

6. แต่ละรูปที่แสดงโปรแกรมกำหนดให้สามารถคลิกแต่ละตำแหน่งได้ เมื่อคลิกที่ Interface ใดของ Router โปรแกรมจะมี Information แสดง ซึ่งการแสดง Information ของอุปกรณ์ได้จากการตรวจสอบพิกัดของจุดที่ใช้เมาส์คลิก และโปรแกรมจะแสดงรูปสำหรับ Interface ที่เลือกชัดเจนขึ้น

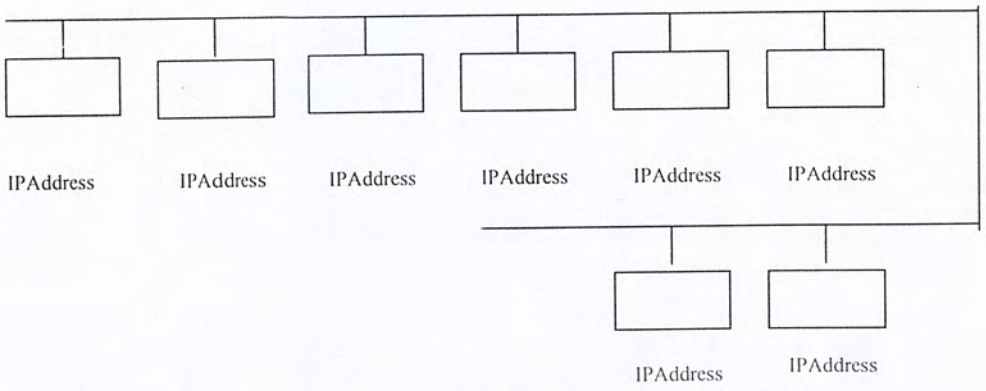
3.3.6 ภาพแสดงแผนที่ของเครือข่าย (แบบที่2)

รูปที่ 3.7 แสดงการออกแบบรูปที่ได้จากความสัมพันธ์ระหว่าง Interface ของ Router กับเครือข่ายที่เชื่อมต่ออยู่

ภาพนี้เป็นภาพที่เชื่อมมาจากภาพแบบที่ 1 โปรแกรมออกแบบในลักษณะที่เมื่อคลิกเลือก Interface ใดของ Router จากภาพที่ 1 แล้ว โปรแกรมจะแสดงภาพใน Interface นั้นที่ชัดเจนขึ้น สามารถดูได้ว่า ใน Interface เดียวกันมีเครือข่ายใดบ้างที่เกาะอยู่ ลักษณะของการวาดภาพจะคล้ายกับการวาดภาพแบบที่ 1 คือเก็บค่าพิกัดของตำแหน่งต่างๆของรูปไว้ เพื่อใช้ในการตรวจสอบหา Information อื่นๆ

รูปที่แสดงมีลักษณะว่า ใน Interface เดียวกันมีเครือข่ายใดที่เกาะอยู่ ถ้าคลิกเลือกเครือข่ายที่ต้องการแล้ว โปรแกรมออกแบบให้มีการไปตรวจสอบว่าเครือข่ายนั้นเคยมีการส่งคำสั่งจาก Management Station ไปเก็บข้อมูลจาก Agent แล้วเก็บค่าไว้ในฐานข้อมูลหรือไม่ ถ้ามีเครือข่ายนี้อยู่ในฐานข้อมูล โปรแกรมจะนำ IP Address ของ Agent ทั้งหมดที่มีสถานะสนับสนุนการทำงานของ SNMP โปรโตคอล จากเครือข่ายนี้มาวาดเป็นรูปที่แสดงให้เห็นว่าในเครือข่ายนี้มี IPAddress ใดบ้างที่สนับสนุนการทำงานของ SNMP โปรโตคอล

3.3.7 ภาพแสดงแผนที่ของเครือข่าย (แบบที่3)



รูปที่ 3.8 แสดงการออกแบบภาพที่ได้จากการ Poll ใน 1 เครือข่าย

ภาพนี้สำหรับแสดงให้เห็นว่าในเครือข่ายที่ Management Station ส่งคำสั่ง Poll ไปยัง Agent ในเครือข่ายเดียวกันมี IP Address ใดบ้างที่สนับสนุนการทำงานของ SNMP โปรโตคอล การแสดงรูปในหน้าจอนี้จะแสดงให้เห็นด้วยว่า IP Address ใดที่สถานะปัจจุบันไม่สนับสนุนการทำงานของ SNMP โปรโตคอล แต่อดีตสถานะที่เก็บไว้ในฐานข้อมูลเคยสนับสนุนการทำงานของ SNMP โปรโตคอล จะแสดงให้เห็นในลักษณะของสีของกรอบรูปที่แตกต่างกัน โดยรูปที่แสดงแต่ละรูปจะมีกรอบรูปแสดงสีไว้ ซึ่งสีของรูปมี 2 สี ใช้สีเหลืองแทน Agent ที่สนับสนุนการทำงานของ SNMP โปรโตคอล และใช้สีแดงแทน Agent ที่สถานะในอดีตสนับสนุนการทำงานของ SNMP โปรโตคอลแต่ปัจจุบันไม่สนับสนุนแล้ว การวาดรูปของโปรแกรมจะเก็บตำแหน่งพิกัดของแต่ละรูปไว้เช่นกัน เมื่อคลิกที่ IP Address ตัวใดก็จะมี Information ของ IP Address ตัวนั้นแสดงที่หน้าจอ

ภาพแต่ละภาพที่ได้จะแสดงในรูปของอุปกรณ์ในเครือข่าย เช่น Computer , Router เป็นต้น การพิจารณาชนิดของอุปกรณ์พิจารณาได้จากค่า ObjectID ของ Vendor ที่ผลิตอุปกรณ์ เมื่อส่งคำสั่ง Poll ไปยัง Agent โดยเลือกข้อมูลใน System Group นำค่า SysObjectID ที่ได้เปรียบเทียบกับค่าในตารางอุปกรณ์ที่เก็บไว้ ถ้าค่ามีอยู่ในฐานข้อมูล โปรแกรมจะแสดงรูป Bitmap ตามไฟล์ที่เก็บไว้ แต่ถ้าไม่มีค่าอยู่ในฐานข้อมูล โปรแกรมจะแสดงรูปที่เป็น Default ของอุปกรณ์ขึ้นมา ซึ่งการเพิ่มไฟล์รูปของอุปกรณ์เข้าสู่ระบบเป็นหน้าที่ของผู้ดูแลเครือข่าย

บทที่ 4
การพัฒนาและทดสอบโปรแกรม

4.1 ซอฟท์แวร์และฮาร์ดแวร์ที่ใช้ในการพัฒนาโปรแกรม

Software
- Borland Delphi 5.0
- BDE (Borland Database Engine)
- Database Desktop
- Component ของ PowerTCP คือ Component Agent, Manager, SNMPPlus
- Window NT Server

Personal Computer
- CPU Pentium III
- Memory 128 MB
- Harddisk 3 GB
- Card Land 10 MBps

4.2 สภาพแวดล้อมของการใช้โปรแกรม

File Name - ProjectMain.exe

Operating System - Window NT Server

Path - D:\Project Network\Network

External File - D:\Project_Nt\

Place - King Mongkut's Institute of Technology Ladkrabang

IP Address - 161.246.11.99

4.3 การทดสอบและขั้นตอนการดำเนินการพัฒนาโปรแกรม

การทดสอบผลการทำงานของโปรแกรมนี้ได้ทำ ณ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง ซึ่งมีหมายเลขเครือข่ายอยู่ในคลาส B (หมายเลข IP Address ที่เริ่มต้นตั้นแต่ 128 - 191) จะมีลักษณะ net.net.host.host คือ 161.246.host.host และสถาบันได้ทำการแบ่ง Subnet ให้เป็นคลาส C โดยใช้ Subnet Mask เป็น 255.255.255.0 และแจกจ่ายให้กับคณะและหน่วยงานต่างๆ ในสถาบัน โดยเครื่องที่ใช้เป็นผู้จัดการเครือข่าย (Management Station) นั้นมีหมายเลข IP คือ 161.246.11.99 อยู่ในความดูแลของสำนักวิจัยและบริการคอมพิวเตอร์

ดังนั้นการทดสอบโปรแกรมจะสนใจข้อมูลในแต่ละคลาส C ซึ่งมี IP Address ที่ใช้แทนอุปกรณ์ในเครือข่ายได้ 254 หมายเลข ที่ได้นำมาแสดงในที่นี้คือ ข้อมูลที่ได้จากการดึงค่าจากอุปกรณ์ในเครือข่าย 161.246.10.0

การทำงานของโปรแกรมจะแบ่งเป็น 2 ส่วนหลัก คือ

4.3.1 ระบุ IP Address ของอุปกรณ์โดยตรง เช่น IP Address 161.246.10.5

มีขั้นตอนในการดำเนินงาน 3 ขั้นตอนคือ

4.3.1.1 ดึงข้อมูลจากอุปกรณ์เครือข่าย

เช่น AgentGetNextRequest (1.3.6.1.2.1.1, public , 161.246.10.5, 161)

แสดงค่าข้อมูลที่ได้จากอุปกรณ์เครือข่ายที่มี IP Address 161.246.10.5 ดังรูปที่ 4.1

4.3.1.2 การเก็บข้อมูลลงฐานข้อมูล

หลังจากที่ดึงค่าจากอุปกรณ์เครือข่ายที่มี IP Address 161.246.10.5 ได้แล้ว จะทำการเก็บลงฐานข้อมูล ชื่อ SystemData.db แสดงดังตารางที่ 4.1 ในส่วนของการแสดงภาพจะได้จากนำค่าในฟิลด์ Object ID ไปตรวจสอบกับฟิลด์ Object_ID ในฐานข้อมูล ชื่อ Equipment.db ถ้าตรงกันจะดึงค่าจากฟิลด์ Bitmap ออกมาแสดงภาพ ถ้าไม่มีค่า Object_ID ในฐานข้อมูล Equipment.db ตรงกับในฐานข้อมูล SystemData.db แล้ว โปรแกรมจะแสดงรูปเป็นค่า Default ของอุปกรณ์

ตัวอย่างการเก็บลงฐานข้อมูล

ตารางที่ 4.1 ฐานข้อมูลแสดงรายละเอียดของ Agent (SystemData.db)

IP Address	Description	ObjectID	Uptime
161.246.10.5	Cisco Internetwork Operating System Software IOS (tm) 7200 Software (C7200-DS-M), Version 11.3(7)T, RELEASE SOFT WARE (fc1)Copyright (c) 1986-1998 by cisco Systems, Inc. Compiled Tue 01-Dec-98 2	1.3.6.1.4.1.9.1.108	1150192083

ตารางที่ 4.1 ฐานข้อมูลแสดงรายละเอียดของ Agent (SystemData.db) (ต่อ)

Contact_Name	Sys_Name	Location	Status	Date_To_Poll	Time_To_Poll
	CAR40		ON	1/3/01	19:00:08

ตารางที่ 4.2 ฐานข้อมูลรูปภาพอุปกรณ์ (Equipment.db)

Object_ID	Vendor_Name	Label	Bitmap
1.3.6.1.4.1.9.1.108	Cisco	Cisco Internetwork	Router.bmp

ตารางที่ 4.3 ตัวอย่างฐานข้อมูลเก็บค่าข้อมูลเข้าออก port

AgentIP Address	Port_ No	Date	Time	Portof Speed	In Octets	InNUcastPkts	Out octets	InUcast Pkts
161.246.10.5	1.00	1/3/01	19:08:09	155520000	3640032208	2412196689	0	0

ตารางที่ 4.3 (ต่อ)

InDis card	InErrors	InUnknown Pro	Out Errors	OutNUcast Pkts	OutDis Cards	OutUcast Pkts	Type of Port
8876549	2936287	2936287	0	0	0	0	ATM 1/0

ในที่นี้แสดงเป็นตัวอย่างเพียง record เดียว คือ ค่าข้อมูลเข้าออก port ที่ 1 ของ IP Address 161.246.10.5

ตารางที่ 4.4 Router_table.db

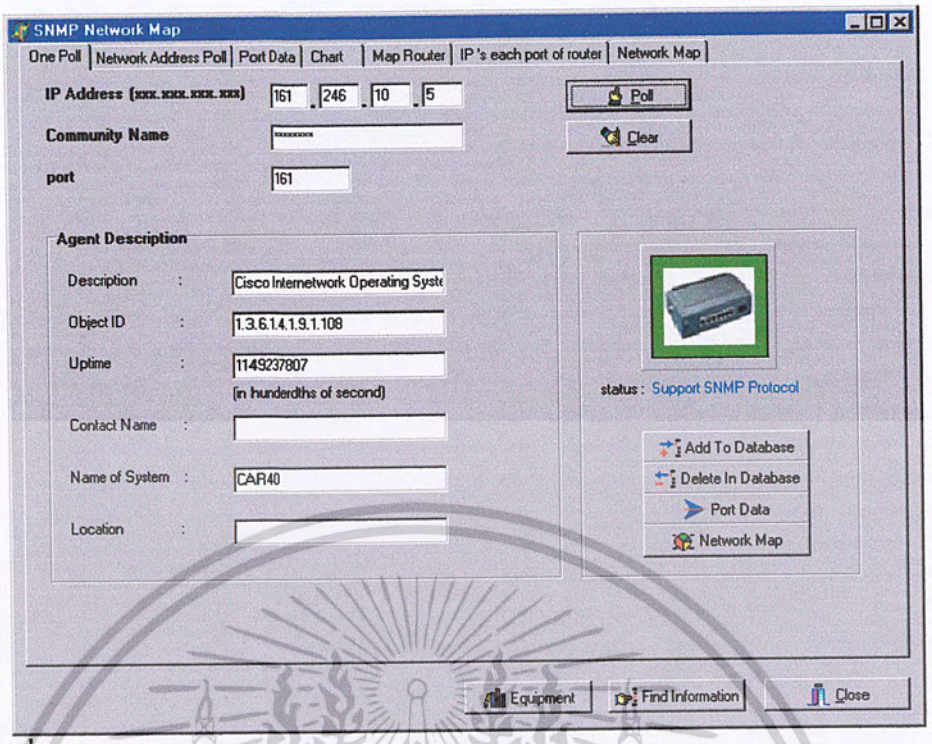
Networkclass	Iprouter	Ipgateway	Ipnetmask	Ipsubclass	Interface_no

4.3.1.3 การแสดงข้อมูลทางจอภาพเพื่อแสดงแผนที่เครือข่าย

การแสดงข้อมูลออกทางจอภาพจะแสดงออกมาด้วยโปรแกรม Delphi 5.0 มีวิธีการใช้งานของโปรแกรมพร้อมแสดงจอภาพดังต่อไปนี้

ตัวอย่างจอภาพของโปรแกรมและวิธีการใช้งานของโปรแกรม

จอภาพแรกเป็นการ Poll 1 Agent แสดงดังรูปที่ 4.1



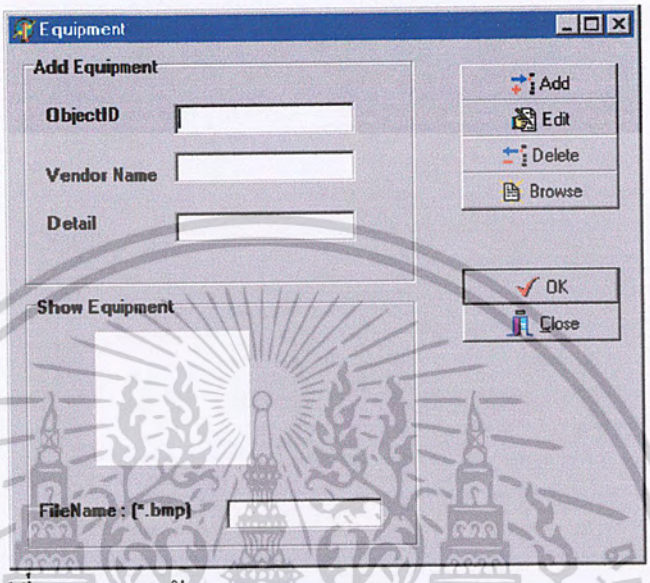
รูปที่ 4.1 แสดงหน้าจอ One Poll

แสดงรายละเอียดของ Agent โดยมีช่องสำหรับใส่ IP Address, Community Name และ Port ของ Agent ที่ต้องการเก็บค่า ในส่วนของ Community Name และ หมายเลข Port โปรแกรมจะกำหนดค่า Default ให้เป็น public และ 161 ตามลำดับ แต่ถ้า Agent ตัวนั้นถูกกำหนดโดย Community Name อื่นก็ต้องใส่ Community Name ของตัวนั้น จึงจะสามารถเก็บค่าข้อมูลได้ เมื่อคลิกปุ่ม Poll โปรแกรมจะเริ่มส่งคำสั่งไปเก็บข้อมูลของ Agent ที่ระบุ Output ที่ได้จากการเก็บค่าของข้อมูลจาก Agent จะเกิดขึ้นได้ 3 กรณีดังนี้

กรณีที่ 1 ถ้า Agent ตัวนี้สนับสนุนการทำงานของโปรโตคอล SNMP โปรแกรมก็จะสามารถเก็บค่าของข้อมูลได้ และมีข้อความแสดงสถานะของ Agent ว่า Support SNMP Protocol พร้อมทั้งมีรายละเอียดอื่นๆ ประกอบด้วย

- Description : รายละเอียดของ Agent เช่น Hardware , OS
- Object ID : Object ID ของ Agent ที่บริษัทผู้ผลิตเป็นผู้กำหนดขึ้น
- Uptime : เวลาตั้งแต่ส่วนของการจัดการเครือข่ายกลับไปเริ่มต้นใหม่ครั้งสุดท้าย
- Contact Name : ผู้ดูแลและรับผิดชอบ Agent
- Name Of System : ชื่อของระบบเครือข่ายย่อย
- Location : ตำแหน่งที่ตั้งของ Agent

เมื่อได้ Object ID ของ Agent แล้ว โปรแกรมจะตรวจสอบชนิดของอุปกรณ์ในฐานข้อมูล เพื่อแสดงรูปของอุปกรณ์โดยพื้น Background เป็นสีเขียว ถ้า Object ID ของ Agent ตัวนี้มีอยู่ในฐานข้อมูลแล้วโปรแกรมจะสามารถแสดงรูปของ Agent ออกมาได้ แต่ถ้าไม่มีค่า Object ID อยู่ในฐานข้อมูลโปรแกรมจะแสดงรูปที่เป็น Default ของอุปกรณ์ขึ้นมาพร้อมทั้งแสดงหน้าจอการเพิ่มข้อมูลของอุปกรณ์คือ



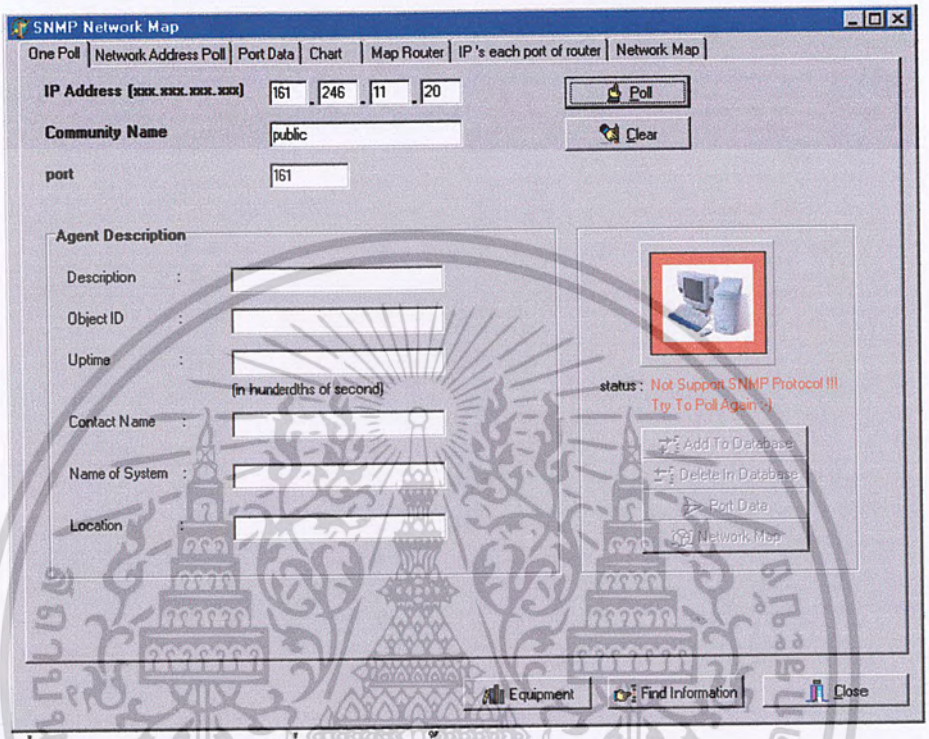
รูปที่ 4.2 แสดงหน้าจอ Equipment

ซึ่งหน้าจอการเพิ่มข้อมูลของอุปกรณ์สามารถที่จะดูข้อมูลของอุปกรณ์ที่จัดเก็บไว้ในฐานข้อมูลได้ โดยคลิกที่ปุ่ม Equipment โปรแกรมจะแสดงรายละเอียดดังต่อไปนี้

- Object ID : Object ID ของ Agent ที่บริษัทผู้ผลิต (Vendor) เป็นผู้กำหนดขึ้น
- Vendor Name : ชื่อของบริษัทผู้ผลิต (Vendor)
- Bitmap : ชื่อของ File รูปของอุปกรณ์
- Detail : ชื่อที่ผู้ใช้โปรแกรมใช้เรียกอุปกรณ์

โปรแกรมจะแสดงรูปของอุปกรณ์ตัวนั้นไว้ด้วย กรณีที่ต้องการเพิ่มข้อมูลของอุปกรณ์ตัวใหม่ คลิกปุ่ม Add และกรอกข้อมูล Object ID , Vendor Name, ชื่อ File รูป (File รูปที่ต้องการจะต้องจัดเก็บไว้ใน Folder เดียวกับโปรแกรม โดยสามารถเลือกดูรูปได้โดยคลิกปุ่ม Browse) หลังจากนั้นคลิกปุ่ม Post โปรแกรมจะทำการเก็บข้อมูลไว้ในฐานข้อมูล

กรณีที่ 2 ถ้าส่งคำสั่งไปเก็บค่าของข้อมูลจาก Agent ซึ่ง Agent ตัวนี้เคยเก็บค่าไว้ในฐานข้อมูลแล้ว โดยสถานะในอดีต Support SNMP Protocol แต่สถานะในปัจจุบันไม่สนับสนุนการทำงานของ SNMP Protocol โปรแกรมจะแสดงข้อมูลของ Agent ตัวนี้ แต่ในส่วนของรูปจะมีพื้น Background เป็นสีแดง และมีข้อความแสดงว่า Not Support SNMP Protocol ดังจอภาพ



รูปที่ 4.3 แสดงหน้าจอกรณีที่ Agent ตัวนี้เคย Support SNMP

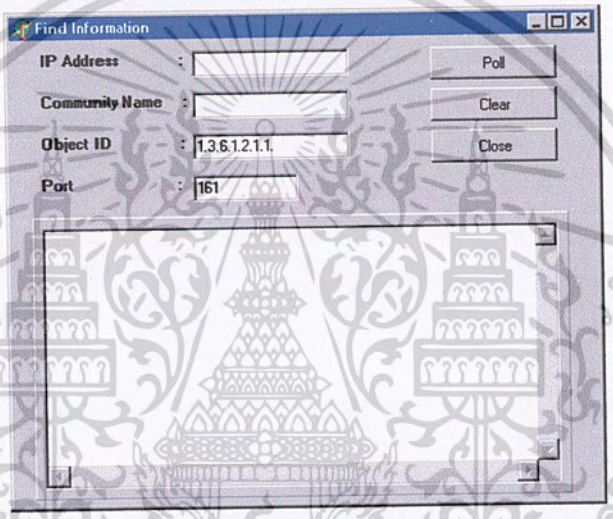
กรณีที่ 3 ถ้า Agent ที่ต้องการเก็บค่าของข้อมูล ไม่สนับสนุนการทำงานของโปรโตคอล SNMP โปรแกรมจะแสดงข้อความสถานะว่า Not Support SNMP Protocol และไม่แสดง Information ต่างๆ ไม่ว่า Output จะเป็นอย่างไรจากทั้งสามกรณี ก็สามารถเก็บ Information ของ Agent ตัวนี้ไว้ในฐานข้อมูลได้โดยคลิกปุ่ม Add To Database แต่ถ้าในฐานข้อมูลมี Information ของ Agent แล้ว โปรแกรมจะเตือนว่าต้องการจะบันทึกทับข้อมูลเดิมหรือไม่ ถ้า Agent ใดสนับสนุน SNMP Protocol สถานะในฐานข้อมูลจะเป็น "ON" แต่ถ้าไม่สนับสนุนการทำงานของ SNMP Protocol สถานะจะเป็น "OFF" และสามารถที่จะลบข้อมูลของ Agent จากฐานข้อมูลได้โดยคลิกปุ่ม Delete In Database

ในการทดสอบครั้งนี้จะเป็นกรณีที่1 คือใส่ IP Address 161.246.10.5 แล้วคลิกปุ่ม Poll จะได้ข้อมูลจากอุปกรณ์ตัวนั้น และจากข้อมูลที่ได้ทำให้ทราบว่าอุปกรณ์ทางเครือข่ายตัวนี้เป็นอุปกรณ์ประเภท Router

กรณีที่ต้องการดูรายละเอียดการไหลของข้อมูลเข้าและออกของ Port สามารถคลิกปุ่ม Port Data โปรแกรมจะแสดงหน้าจอ Port Data (จะกล่าวถึงในลำดับต่อไป)

กรณีที่ต้องการดูแผนที่เครือข่ายของ Network Address ที่ IP Address ตัวนี้เกาะอยู่ ให้คลิกปุ่ม Network Map โปรแกรมจะทำการตรวจสอบว่า Network Address ของคลาสนี้เคยเก็บค่าไว้ในฐานข้อมูลแล้วหรือไม่เพราะการแสดงรูป โปรแกรมจะดึงค่าจากฐานข้อมูลเพื่อนำมาแสดงรูป ถ้าคลาสนี้เคยเก็บ Information ไว้ในฐานข้อมูลแล้วจะสามารถแสดงรูปได้ แต่ถ้ายังไม่มี Information ของคลาสนี้ในฐานข้อมูล โปรแกรมจะแสดงข้อความให้ไปเก็บค่าของคลาสนี้และบันทึกลงฐานข้อมูลก่อน

ปุ่ม Clear ไว้สำหรับต้องการเคลียร์ข้อมูลต่างๆที่แสดงไว้ในช่องสำหรับกรอกข้อมูล



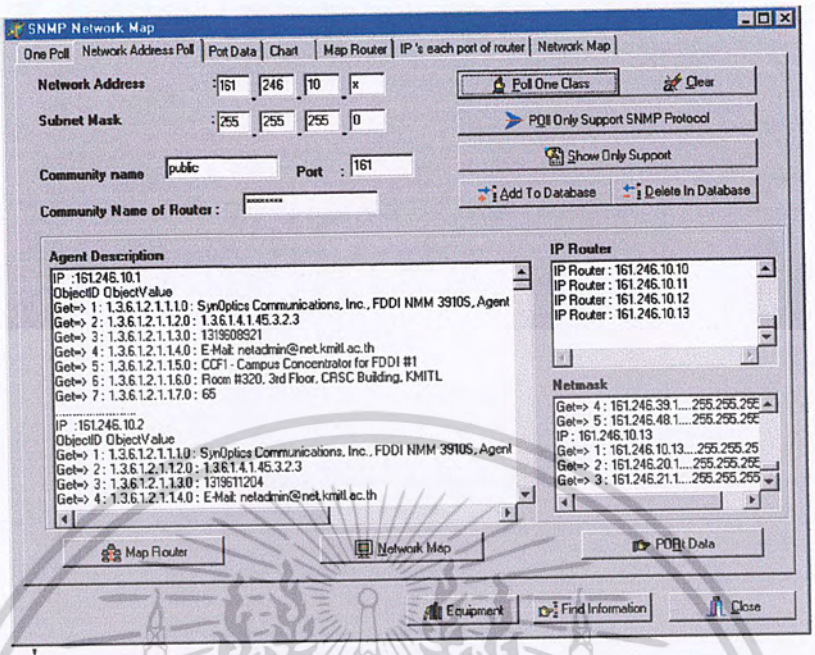
รูปที่ 4.4 แสดงจอภาพ Find Information

4.3.2 ระบุเป็นหมายเลขของเครือข่าย (Network Address) โปรแกรมจะทำการวนลูปเก็บข้อมูลจากอุปกรณ์ในเครือข่ายที่สนใจในลักษณะ Sequencial ในที่นี้ถ้าสนใจอุปกรณ์เครือข่ายที่อยู่ในคลาส C ซึ่งหมายเลข เครือข่ายเลขที่ 161.246.10 และหมายเลขเครื่องลูกข่ายตั้งแต่ 1 จนถึง 254 ซึ่งก็คือจะทำการดึงค่าจากอุปกรณ์ในเครือข่ายที่มี IP Address 161.246.10.1 จนถึง IP Address 161.246.10.254 โดยอุปกรณ์ที่ดึงค่าข้อมูลได้ต้องสนับสนุนการทำงานของ SNMP มีขั้นตอนการทำงาน 3 ขั้นตอนเหมือนแบบแรกคือ

4.3.2.1 ดึงข้อมูลจากอุปกรณ์เครือข่าย

ใช้ฟังก์ชันเดียวกับการระบุ IP Address แต่จะวนลูปดึงค่าข้อมูลให้ครบ 254 รอบ

หน้าจอแสดง Network Address Poll ดังแสดงในรูปที่ 4.5



รูปที่ 4.5 แสดงจอภาพ Network Address Poll

การแสดงรายละเอียดของ Agent แบ่งออกเป็น 2 กรณี ดังนี้

กรณีที่ 1 เมื่อต้องการเก็บข้อมูลของคลาส C

โดยมีช่องสำหรับใส่ Network Address ของคลาส C , Subnet Mask ซึ่งกำหนดเป็น 255.255.255.0 , Community Name (โปรแกรมกำหนดให้เป็น Public สำหรับอุปกรณ์โดยทั่วไป) , Port No., Community Name Of Router (เป็น Community Name เฉพาะสำหรับเครือข่าย) เมื่อใส่ข้อมูลเรียบร้อยแล้วคลิกปุ่ม **Poll One Class** โปรแกรมจะเริ่มเก็บค่า Information จาก Agent ต่างๆ โดยวน Loop เก็บค่าตั้งแต่ Host Address เป็น 1 จนถึง 254 (เนื่องจาก Host Address 0 แทนหมายเลขเครือข่าย และ Host Address 255 แทน Broadcast Address) Agent ใดที่สนับสนุนการทำงานของ SNMP Protocol โปรแกรมก็จะเก็บ Information ได้และมีข้อความแสดงรายละเอียดต่างๆของ Agent ไว้ใน Memo “Agent Description” แต่ถ้า Agent ใดไม่สนับสนุนการทำงานของ SNMP Protocol จะมีข้อความแสดงว่า “Not Support SNMP Protocol !!! Try To Poll Again :-) ”

รายละเอียดของข้อมูลที่แสดงใน Memo “Agent Description” ประกอบด้วย

- Description : รายละเอียดของ Agent เช่น Hardware , OS
- Object ID : Object ID ของ Agent ที่บริษัทผู้ผลิตเป็นผู้กำหนดขึ้น
- Uptime : เวลาตั้งแต่ส่วนของการจัดการเครือข่ายกลับไปเริ่มต้นใหม่ครั้งสุดท้าย

- Contact Name : ผู้ดูแลและรับผิดชอบ Agent
- Name Of System : ชื่อของระบบเครือข่ายย่อย
- Location : ตำแหน่งที่ตั้งของ Agent

โปรแกรมจะแสดง IP Address ของ Router ไว้ใน Memo "IP Router" ส่วนใน Memo "Netmask" ข้อมูลที่แสดงประกอบด้วย

- หมายเลข IP Address ของ Router
- IP Address ที่เกาะอยู่ในแต่ละ Port
- Subnet Mask ของแต่ละ Port

เมื่อทำการ Poll เก็บค่าของข้อมูลเรียบร้อยแล้ว ปุ่ม Show Only Support , Delete In Database , Add To Database , Map Router , Network Map, Port Data จึงจะสามารถทำงานได้

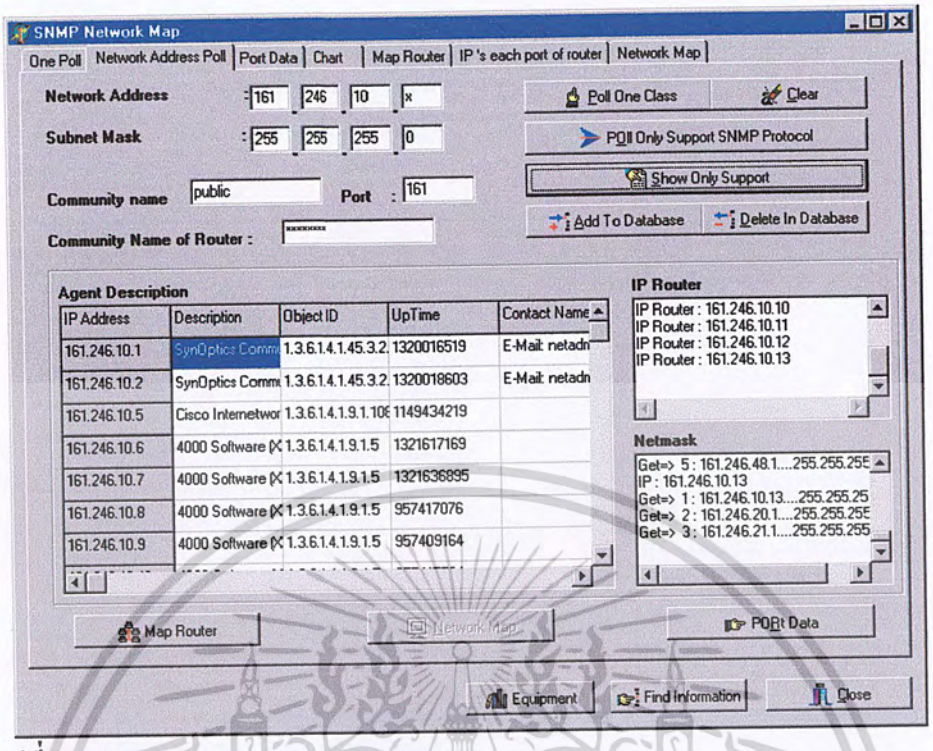
จากการทดสอบโดยใส่หมายเลขเครือข่าย 161.246.10 แล้วทำจากดึงค่าข้อมูลจะได้ว่ามีอุปกรณ์ในเครือข่ายที่สนับสนุนการทำงานของ SNMP Protocol 17 ตัว และเป็นอุปกรณ์ประเภท Router 9 ตัว

กรณีที่ 2 เมื่อต้องการเก็บข้อมูลของคลาส B

โดยมีช่องสำหรับใส่ Network Address ของคลาส C , Subnet Mask ซึ่งกำหนดเป็น 255.255.0.0 , Community Name (โปรแกรมกำหนดให้เป็น Public สำหรับอุปกรณ์โดยทั่วไป) , Port No., Community Name Of Router (เป็น Community Name เฉพาะสำหรับเครือข่าย) เมื่อใส่ข้อมูลเรียบร้อยแล้วคลิกปุ่ม **Poll One Class** โปรแกรมจะเริ่มเก็บค่า Information จาก Agent ต่างๆ โดยวน Loop เก็บค่าตั้งแต่ Network Address ของ Class B เป็น 1 และในแต่ละ Network Address นี้ โปรแกรมจะกำหนดให้มีการ Poll เก็บค่าของข้อมูลตั้งแต่ Host Address เป็น 1 จนถึง 254 เปรียบเสมือนเป็นการ Poll เก็บค่าของข้อมูลใน Class C และเมื่อ Poll จนครบ 1 คลาสแล้วก็จะเริ่ม Poll เก็บค่า Information จาก Network Address ของคลาสใหม่คือคลาสที่ 2 และPoll เก็บค่า Information ของ Host Address ตั้งแต่ 1 ถึง 254 ของคลาสที่ 2 วนอย่างนี้ไปเรื่อยๆจนกว่าจะครบคือถึง Network Address ที่ 254 (เนื่องจาก Host Address 0 แทนหมายเลขเครือข่าย และ Host Address 255 แทน Broadcast Address)

- **ปุ่ม Show Only Support**

โปรแกรมจะแสดง Agent ที่สนับสนุนการทำงานของ SNMP Protocol ที่ได้จากการเก็บค่า Information จาก Network Address ที่กำหนด โดยแสดงในรูปของตาราง เพื่อให้มองดูเข้าใจง่ายขึ้น แสดงดังรูปที่ 4.6



รูปที่ 4.6 แสดงจอภาพ Show Only Support

● ปุ่ม Poll Only Support SNMP Protocol

คลิกปุ่มนี้เมื่อต้องการ Poll เก็บค่า Information ของ Agent ใน Network Address ที่กำหนด โดยการ Poll จะ Poll เฉพาะ Agent ที่สนับสนุนการทำงานของ SNMP Protocol เท่านั้นโดยโปรแกรมจะเลือกเฉพาะ IP Address ที่มีสถานะเป็น "ON" จากฐานข้อมูลมา Poll เก็บ Information ใหม่อีกครั้ง ซึ่งเริ่มแรกโปรแกรมจะตรวจสอบก่อนว่า Network Address นี้มีเก็บอยู่ในฐานข้อมูลหรือไม่ ถ้าไม่มีค่าเก็บอยู่โปรแกรมจะแสดงข้อความว่าไม่มี Network Address ของคลาสนี้ในฐานข้อมูลให้ไปทำการ Poll เก็บค่าก่อน โดยใช้การ Poll แบบ Poll One Class และเก็บข้อมูลไว้ในฐานข้อมูล เพื่อใช้ในการตรวจสอบสถานะในครั้งต่อไป

4.3.2.2 การเก็บข้อมูลลงฐานข้อมูล

● ปุ่ม Add To Database

คลิกเมื่อต้องการเก็บ Information ของ Agent ที่ทำการ Poll ซึ่งเก็บทั้ง Agent ที่สนับสนุนการทำงานของ SNMP Protocol และไม่สนับสนุนการทำงานของ SNMP Protocol

● ปุ่ม Delete In Database

คลิกเมื่อต้องการลบ Information ของ Agent ที่ทำการ Poll ออกจากฐานข้อมูล ซึ่งการ Delete นี้โปรแกรมกำหนดให้เป็นการ Delete Information ของทั้งคลาส จากการกำหนด Network Address

ตัวอย่างการเก็บลงฐานข้อมูล

ตารางที่ 4.5 ฐานข้อมูลแสดงรายละเอียดของ Agent (SystemData.db)

IP Address	Description	ObjectID	Uptime
161.246.10.1	SynOptics Communications, Inc., FDDI NMM 3910S,Agent V1.3, Compiled Date: Oct 13 1992, Time: 12:25:45	1.3.6.1.4.1.45.3.2.3	1320773484
161.246.10.10	4000 Software (XX-K), Version 9.14(9), RELEASE SOFTWARE (fc1)Patchlevel = 9.1 (12.6)Compiled Wed 28-Sep-94 14:13 by chansen	1.3.6.1.4.1.9.1.5	958173723
161.246.10.100		xxx	

ตารางที่ 4.5 ฐานข้อมูลแสดงรายละเอียดของ Agent (SystemData.db) (ต่อ)

Contact_Name	Sys_Name	Location	Status	Date_To_Poll	Time_To_Poll
E-Mail: netadmin@net.kmitl.ac.th	CCF1 - Campus Concentrator for FDDI #1	Room #320, 3rd Floor, CRSC Building, KMITL	ON	1/3/01	19:30:08
	CAR23.net.kmitl.ac.th		ON	1/3/01	19:33:27
			OFF	1/3/01	19:46:07

ตารางที่ 4.6 ฐานข้อมูลรูปภาพอุปกรณ์ (Equipment.db)

Object_ID	Vendor_Name	Label	Bitmap
1.3.6.1.4.1.45.3.2.3	SynOptics	SynOptics Communication	concen.bmp
1.3.6.1.4.1.9.1.5	4000 Software	4000 Software CAR11.net.kmitl.ac.th	router_s.bmp

ตารางที่ 4.7 ฐานข้อมูลเก็บค่าข้อมูลเข้าออก port

AgentIP Address	Port_No	Date	Time	Portof Speed	In Octets	InNUcastPkts	Out octets	InUcast Pkts
161.246.10.5	1.00	1/3/01	19:08:09	155520000	3640032208	2412196689	0	0

ตารางที่ 4.7 ฐานข้อมูลเก็บค่าข้อมูลเข้าออก port (ต่อ)

InDiscard	InErrors	InUnknownPro	Out Errors	OutNUcastPkts	OutDis Cards	OutUcastPkts	Type of Port
8876549	2936287	2936287	0	0	0	0	ATM 1/0

ตารางที่ 4.7 เป็นลักษณะเดียวกับตารางที่ 4.3 คือแสดงข้อมูลเข้าออกแต่ละ port ของ IP Address ที่เลือกที่จะเก็บค่าข้อมูล เพื่อใช้ในการตรวจสอบความหนาแน่นการไหลของข้อมูลเข้า-ออก port ณ ช่วงเวลาหนึ่ง

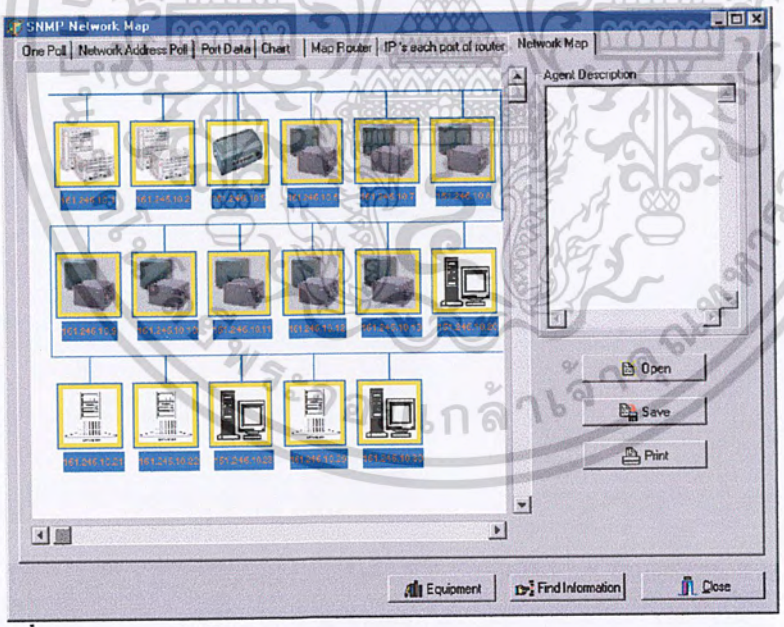
ตารางที่ 4.8 Router_table.db

NetworkClass	Iprouter	Ipgateway	Ipnetmask	Ipsubclass	Interface_no
161.246.10.0	161.246.10.0	161.246.18.1	255.255.255.0	161.246.18.0	2
161.246.10.0	161.246.10.0	161.246.37.1	255.255.255.0	161.246.37.0	3
161.246.10.0	161.246.10.11	161.246.10.11	255.255.255.0	161.246.10.0	4
161.246.10.0	161.246.10.11	161.246.4.5	255.255.255.0	161.246.4.0	2

4.3.2.3 การแสดงรูปแผนที่เครือข่าย

- **ปุ่ม Network Map**

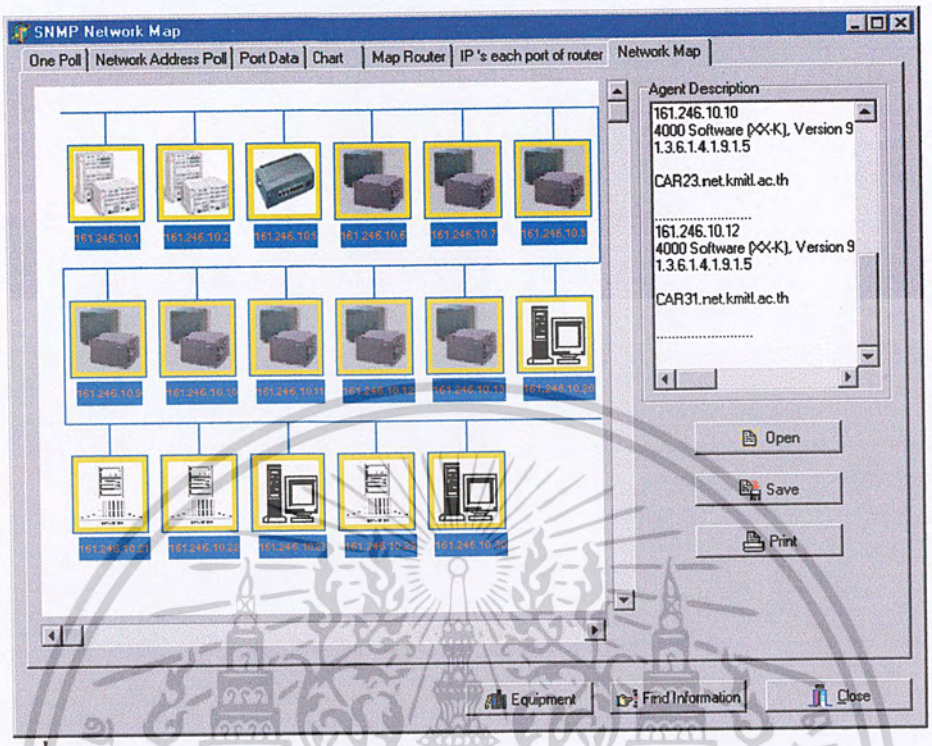
คลิกปุ่มนี้เมื่อต้องการดูว่าในเครือข่ายที่กำหนด Network Address มี IP Address ใดบ้างที่สนับสนุนการทำงานของโปรโตคอล SNMP เมื่อคลิกปุ่มนี้แล้วหน้าจอ Network Map จะแสดงขึ้นมา



รูปที่ 4.7 แสดงจอภาพ Network Map

ในการทดสอบนี้มีอุปกรณ์ที่สนับสนุนการทำงานของ SNMP Protocol 17 ตัว ดังแสดงในรูปที่ 4.7 ซึ่งรูปของอุปกรณ์ที่แสดงนั้นอาจไม่ใช่รูปลักษณะที่แท้จริงของอุปกรณ์นั้นๆ แต่สามารถเปลี่ยนแปลงรูปภาพได้ในภายหลัง

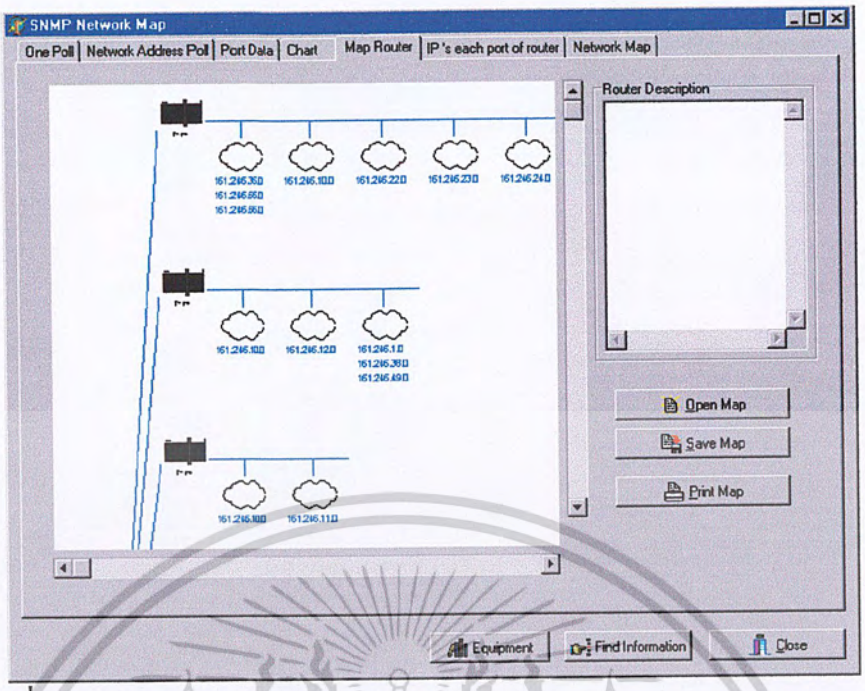
และเมื่อคลิกบริเวณรูปภาพอุปกรณ์ที่แสดงบนหน้าจอ จะแสดงข้อมูลของอุปกรณ์ตัวนั้นใน Memo Agent Description ทางด้านขวามือ ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 แสดงจอภาพ Network Map(2)

- **ปุ่ม Map Router**

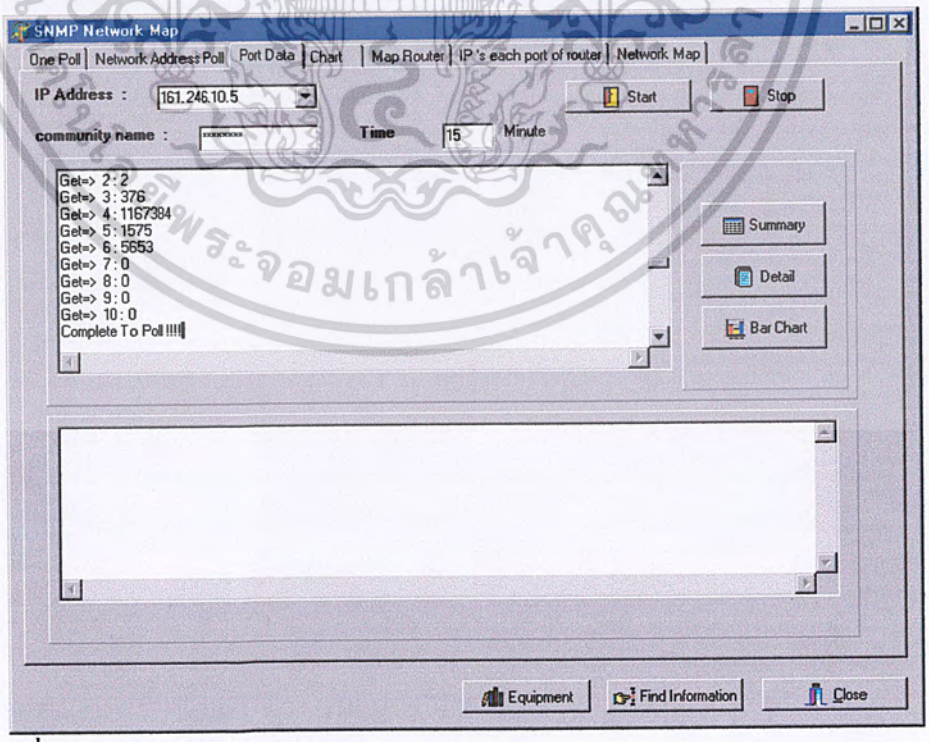
คลิกปุ่มนี้เมื่อต้องการดูแผนที่เครือข่ายของ Network Address ที่กำหนด เมื่อคลิกปุ่มนี้แล้วหน้าจอของ Map Router จะแสดงขึ้นมา



รูปที่ 4.9 แสดงจอภาพ Router Map

- **ปุ่ม Port Data**

สำหรับดูข้อมูลที่ไหลเข้าและออกจากแต่ละ Port ของอุปกรณ์ เมื่อคลิกปุ่มนี้แล้ว โปรแกรมจะแสดงหน้าจอ Port Data ขึ้นมา และส่ง IP Address ที่สนับสนุนการทำงานของ SNMP Protocol ไปสำหรับดูข้อมูล



รูปที่ 4.10 แสดงหน้าจอ Port Data

หน้าจอจะมีช่องสำหรับใส่ IP Address และ Community Name ของ Agent ที่ต้องการจะดูข้อมูล

ในกรณีนี้ถ้าเริ่มแรกอยู่ที่หน้าจอ One Poll เมื่อทำการ Poll ไปยัง Agent แล้วได้คำตอบกลับมาว่า Agent นี้สนับสนุนการทำงานของ SNMP Protocol โปรแกรมจะกำหนดให้สามารถเข้ามาดูการไหลเข้าออกของข้อมูลในแต่ละ Port ได้ แต่ถ้าตรวจสอบแล้วว่า Agent ตัวนี้ไม่สนับสนุนการทำงานของ SNMP Protocol ก็จะไม่สามารถเข้ามาดูรายละเอียดของแต่ละ Port ได้

กรณีที่เริ่มแรกอยู่ที่หน้าจอ Network Address Poll ซึ่งเป็นการ Poll เก็บ information ของเครือข่ายในคลาส B และคลาส C เมื่อทำการ Poll เก็บค่าแล้ว โปรแกรมจะเก็บ IP Address ที่สนับสนุนการทำงานของ SNMP Protocol ไว้ เมื่อคลิกเลือกปุ่ม Port Data โปรแกรมจะแสดงหน้าจอ Port Data และส่ง IP Address ที่เก็บไว้มาแสดงใน Combo Box เพื่อให้เลือกได้ว่าจะดูข้อมูลรายละเอียดของ Port ของ IP Address ใด ซึ่ง IP Address เหล่านี้การันตีได้ว่าสนับสนุนการทำงานของ SNMP Protocol

เมื่อเลือก IP Address ได้แล้ว (กรณีที่ไม่ได้มาจากหน้าจอ One Poll หรือ Network Class Poll สามารถกำหนด IP Address ที่ต้องการได้ โดยใส่ใน Combo Box) หลังจากนั้นกำหนดเวลาที่ใช้ในการเก็บข้อมูลในแต่ละครั้งโดยมีหน่วยเป็นนาที แล้วคลิกปุ่ม Start เพื่อเริ่มต้นการเก็บค่าของรายละเอียด Port เมื่อต้องการหยุดการเก็บค่าของข้อมูลให้คลิกปุ่ม Stop

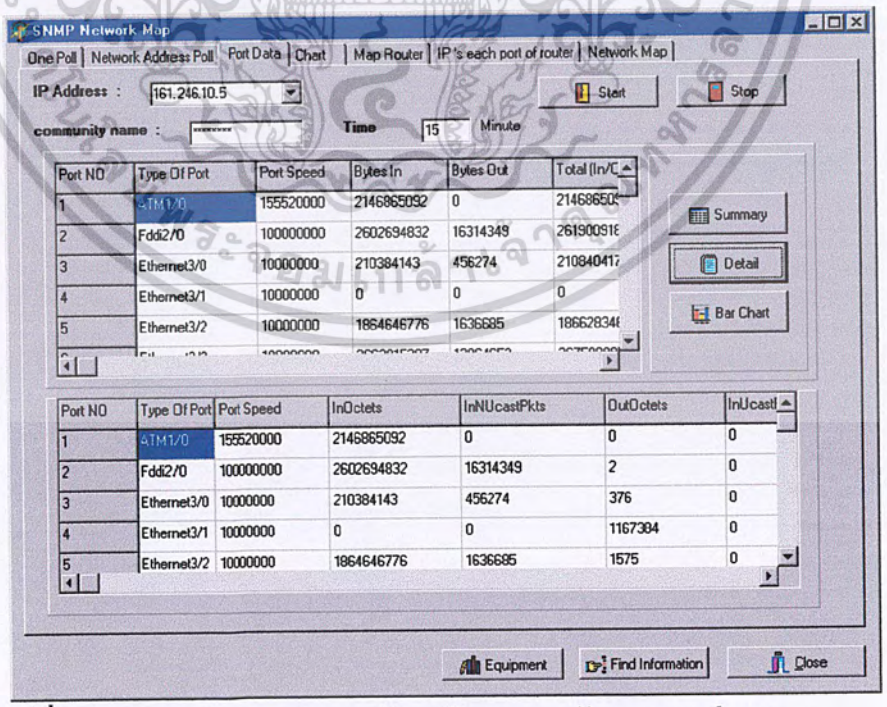
เมื่อสิ้นสุดการเก็บค่าแล้ว ถ้าต้องการดูรายละเอียดต่างๆของแต่ละ Port ให้คลิกปุ่มดังนี้

- **ปุ่ม Summary** เมื่อคลิกปุ่มนี้แล้วโปรแกรมจะแสดงตารางที่ประกอบด้วยข้อมูลดังต่อไปนี้แสดงดังรูปที่ 4.11

- Port No : แทนหมายเลขของ Port
- Type Of Port : แทนชนิดของ Port
- Port Of Speed : แทนความเร็วในการรับส่งข้อมูลของ Port นั้น
- Byte In : แทนจำนวนข้อมูลที่รับเข้ามาในแต่ละ Port
- Byte Oute : แทนจำนวนข้อมูลที่ส่งออกไปในแต่ละ Port
- Total On Port : แทนจำนวนรวมของข้อมูลที่รับเข้ามาและส่งออกไปใน Port นั้นๆ
- % On Device : แทนอัตราส่วนของการรับและส่งข้อมูลของ Port นั้นเมื่อเทียบกับการรับส่งข้อมูลทั้งหมดของทุก Port

- **ปุ่ม Detail** เมื่อคลิกปุ่มนี้แล้วโปรแกรมจะแสดงรายละเอียดของข้อมูลที่มีการไหลผ่านแต่ละ Port ดังต่อไปนี้ ดังแสดงรูปที่ 4.11

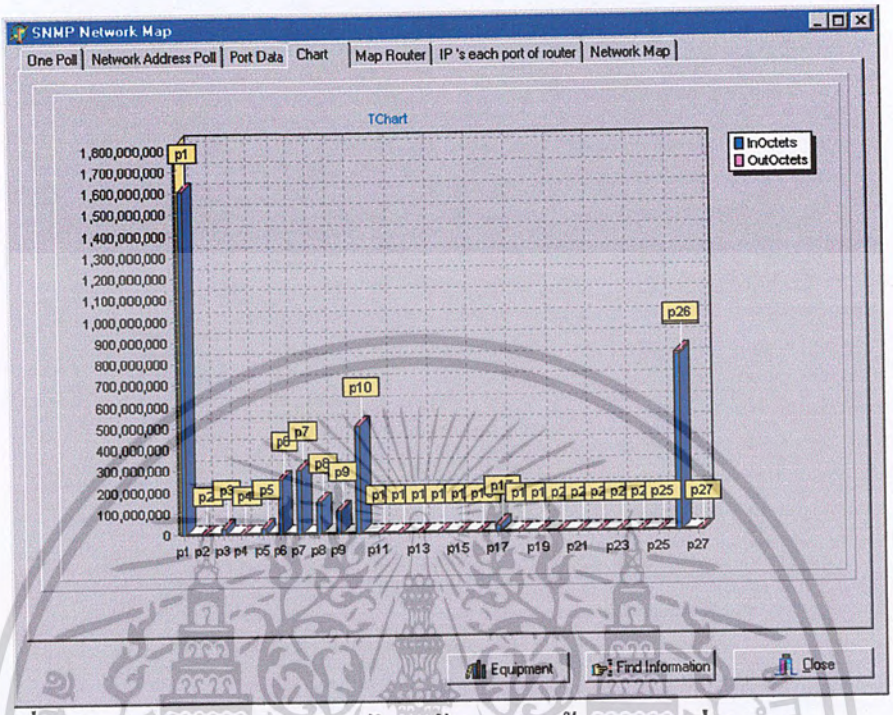
- IfInOctets : แทนจำนวนรวมของข้อมูลที่รับเข้ามาในแต่ละ port
- IfOutOctets : แทนจำนวนรวมของข้อมูลที่ส่งออกไปในแต่ละ port
- IfInErrors : แทนจำนวน error ที่เกิดขึ้นเมื่อรับ packet เข้ามาในแต่ละ port
- IfOutErrors : แทนจำนวน error ที่เกิดขึ้นเมื่อส่ง packet ออกไปในแต่ละ port
- IfInUcastPkts: แทนจำนวนของ packet ที่ unicast ไป subnetwork ที่ Protocol layer ที่สูงกว่า
- IfInNUcastPkts : แทนจำนวน packet ที่ nonunicast ไป subnetwork ที่ Protocol layer ที่สูงกว่า
- IfInDiscards: แทนจำนวนของ inbound packets ที่ทิ้งถึงแม้ไม่มี error เพื่อป้องกันไม่ให้ส่งไป Protocol layer ที่สูงกว่า
- IfInUnknowProtos : แทนจำนวนของ inbound packets ที่ถูกทิ้งเพราะไม่รู้หรือไม่สนับสนุนโปรโตคอล
- IfOutUcastPkts : แทนจำนวนรวมของ packet ซึ่ง higher-level protocols ร้องขอเพื่อ unicast ไป address ของ subnetwork รวมถึง packet ที่ทิ้งหรือไม่ได้ส่งด้วย
- IfOutNUcastPkts : แทนจำนวนรวมของ packet ซึ่ง higher-level protocols ร้องขอที่ nonunicast ไป address ของ subnetwork รวมถึง packet ที่ทิ้งหรือไม่ได้ส่งด้วย
- IfOutDiscards : แทนจำนวนของ outbound packets ที่ทิ้งถึงแม้ไม่มี error เพื่อป้องกันไม่ให้ส่งไป protocol layer ที่สูงกว่า



รูปที่ 4.11 แสดงหน้าจอสรุปผลการไหลเข้าออกของข้อมูลของแต่ละ Port

- **ปุ่ม Bar Chart**

คลิกปุ่มนี้เมื่อต้องการดูการไหลเข้าและออกของข้อมูลในแต่ละ port ของ agent ในลักษณะของกราฟแท่ง เมื่อคลิกปุ่มนี้แล้วโปรแกรมจะแสดงหน้าจอ Chart ขึ้นมา



รูปที่ 4.12 แสดงหน้าจอแผนภูมิการไหลเข้าออกของข้อมูลของแต่ละ Port

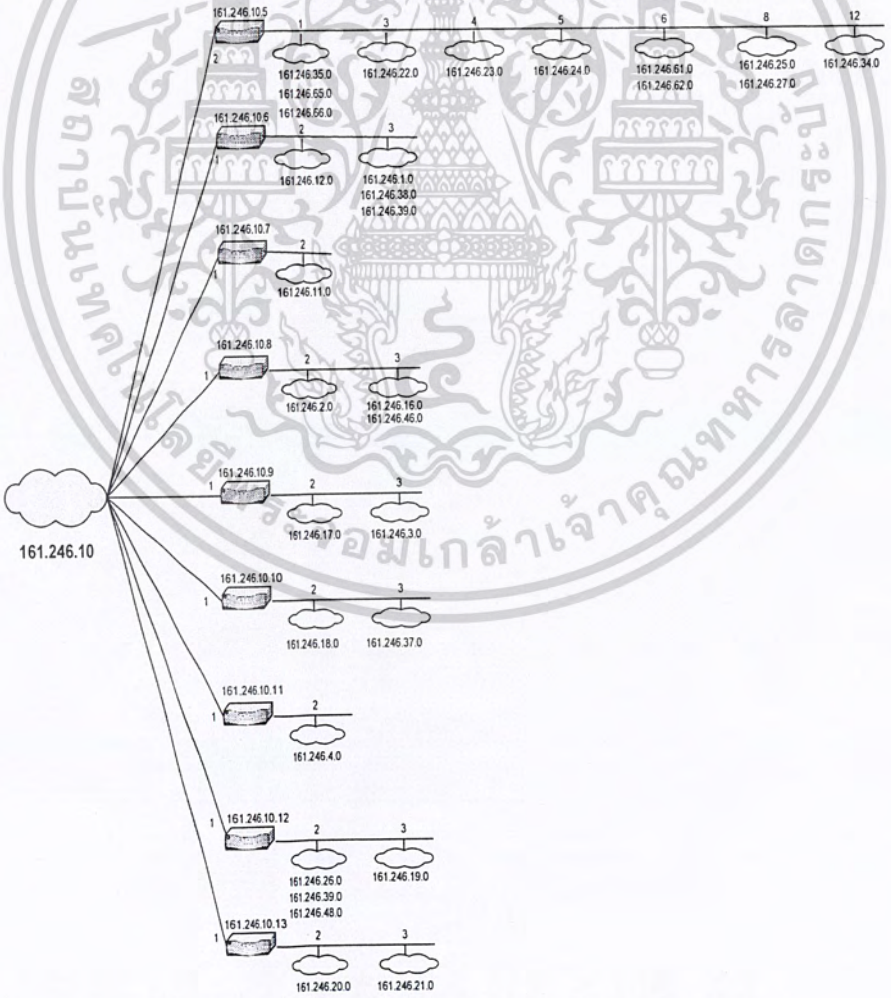
จอภาพแสดงการไหลเข้าและออกของแต่ละ Port ของ Agent แสดงดังรูปที่ 4.12 แสดงการไหลเข้าและออกของข้อมูลจากแต่ละ Port ในลักษณะของกราฟแท่ง สำหรับแต่ละแท่งของกราฟแทนแต่ละ Port ของ Agent ซึ่งประกอบด้วยส่วนของข้อมูลเข้า และข้อมูลที่ออกจาก port แสดงอยู่ในแท่งเดียวกัน

บทที่ 5
สรุปผลการดำเนินงานปัญหาพิเศษ

5.1 ผลการทดลองปัญหาพิเศษ

ข้อมูลที่ได้จากการทดลอง โดยการที่ Station ส่งคำสั่ง Poll ไปยังอุปกรณ์ในเครือข่ายที่มี Network address เป็น 161.246.10 และมี Subnet mask เป็น 255.255.255.0 เมื่อพิจารณาจากค่า Subnet mask แสดงว่าเป็นเครือข่ายในคลาส C ทำการ Poll เมื่อวันที่ 1 มีนาคม 2543 ตั้งแต่เวลา 19.00 น. ถึงเวลา 20.00 น.

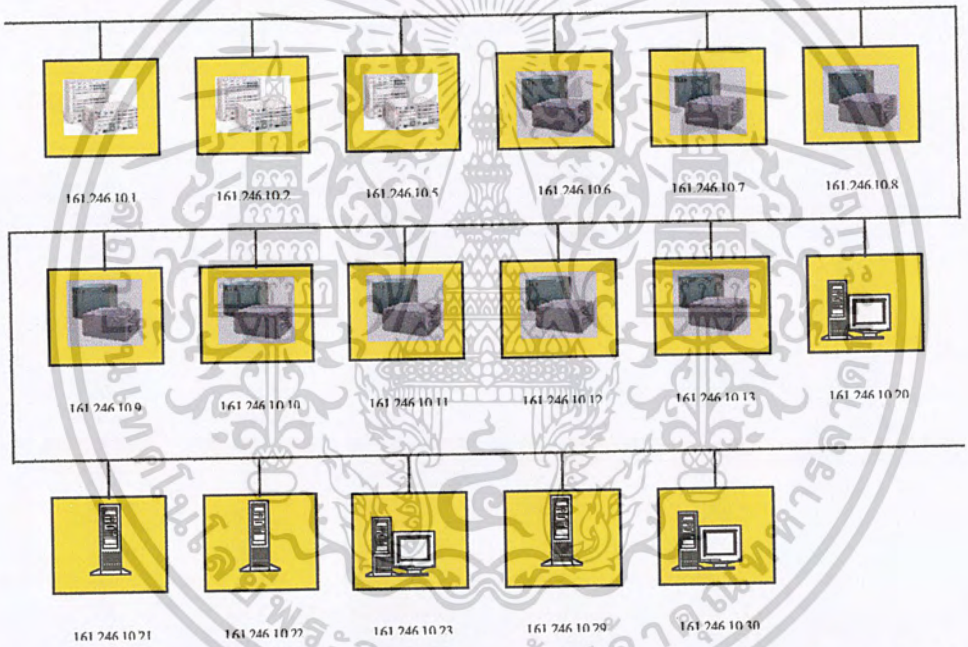
จากการ Poll เก็บค่า information ของอุปกรณ์ในเครือข่ายที่มีหมายเลขเครือข่ายเป็น 161.246.10 และใช้ Community name ค่าหนึ่ง ใช้ Port 161 มีอุปกรณ์ในเครือข่ายที่สนับสนุนการทำงานของโปรโตคอล SNMP ทั้งสิ้น 17 ตัว และมีอุปกรณ์ที่เป็น Router ในเครือข่ายอยู่ 9 ตัว ใช้เวลาในการ Poll เก็บค่าตลอดทั้งเครือข่ายเป็นเวลา 60 นาทีและได้แผนที่ของเครือข่าย ดังรูป



รูปที่ 5.1 แสดงแผนที่ของเครือข่ายซึ่งมีหมายเลขเครือข่ายเป็น 161.246.10

เครือข่ายที่ได้จากการ Poll เป็นเครือข่าย TCP/IP ที่อุปกรณ์ในเครือข่ายสนับสนุนการทำงานของโปรโตคอล SNMP จากรูปแสดงการเชื่อมต่อระหว่าง Router กับเครือข่ายที่เชื่อมอยู่ในแต่ละ Interface ของ Router โดยใน 1 Interface ของ Router อาจจะมีหลายเครือข่ายที่เชื่อมต่อกันอยู่ที่ Interface เดียวกัน หรือมีเพียงเครือข่ายเดียวเท่านั้นที่เชื่อมต่ออยู่ การแสดงรูปถ้าหากว่าใน 1 Interface มีหลายเครือข่ายเชื่อมต่ออยู่จะใช้รูป 1 วงแทนหลายเครือข่ายโดยมีหมายเลขเครือข่ายบอกด้านล่างของรูปนั้น และสำหรับแต่ละ Interface ของ Router จะมี Interface number บ่งบอก

การแสดงรูปอุปกรณ์ที่สนับสนุนการทำงานของโปรโตคอล SNMP สำหรับเครือข่ายเดียวกันจะแสดงออกมาดังรูปที่ 5.2 ซึ่งเป็นการ Poll ในเครือข่าย 161.246.10 มีอุปกรณ์ที่สนับสนุนการทำงานของโปรโตคอล SNMP ทั้งหมด 17 ตัว



รูปที่ 5.2 แสดงอุปกรณ์ในเครือข่ายที่สนับสนุน SNMP โปรโตคอล โดยมีหมายเลขเครือข่ายเป็น 161.246.10

5.2 สรุปผลปัญหาพิเศษ

หลังจากที่ Station ส่งคำสั่ง Poll ไปยังเครือข่ายที่ต้องการแล้ว ข้อมูลที่ได้จะอยู่ในรูปของความสัมพันธ์ระหว่าง Router กับ หมายเลข Gateway แต่ละ Interface ของ Router ก็จะมีเครือข่ายอื่นๆเชื่อมต่ออยู่ บาง Interface อาจจะมีหลายเครือข่าย แต่บาง Interface มีเพียงเครือข่ายเดียวเท่านั้น ข้อมูลที่ได้สามารถนำมาสร้างเป็นแผนที่เครือข่ายที่แสดงความสัมพันธ์ระหว่าง Router กับเครือข่ายย่อยได้

แผนที่เครือข่ายที่สร้างขึ้นถูกสร้างเป็นภาพบิทแมพมีการเก็บพิกัดของภาพทั้งของ Router และเครือข่ายย่อยที่เชื่อมต่อกันอยู่ สามารถเข้าไปดูรายละเอียดทางด้าน Hardware ผู้ดูแลระบบ ตำแหน่งทางกายภาพของอุปกรณ์ในเครือข่าย รวมทั้งปริมาณการไหลเข้าและออกของข้อมูลจากแต่ละ Interface ของอุปกรณ์ในเครือข่ายได้ โดยคลิกเลือกเครือข่ายที่ต้องการ

5.3 สรุปผลการดำเนินงานปัญหาพิเศษ

ในการดำเนินงานตามปัญหาพิเศษตั้งแต่ มิถุนายน 2543 จนถึง มีนาคม 2544 ได้มีการดำเนินงานหลายประเภท และหลายขั้นตอน เพื่อที่จะสามารถบรรลุจุดมุ่งหมายที่ตั้งไว้ได้ จึงได้แบ่งการดำเนินงานออกเป็นหลายขั้นตอน ซึ่งในปัญหาพิเศษนี้บางขั้นตอนสามารถทำควบคู่กันไปได้ แต่บางขั้นตอนจะต้องรอผลลัพธ์จากขั้นตอนอื่นก่อน

โดยขั้นตอนการทำงานของปัญหาพิเศษ สามารถแบ่งออกได้ดังนี้

5.3.1 ขั้นตอนการศึกษาและรวบรวมข้อมูล

การศึกษาข้อมูลที่เกี่ยวข้องทั้งหมด ซึ่งรวมถึงการติดต่อกับผู้ที่ศึกษาและพัฒนาโปรแกรมที่ใช้โปรโตคอลเดียวกันและที่เกี่ยวข้องกันกับที่ต้องใช้ในปัญหาพิเศษทั้งหมด รวมถึงการออกแบบการทำงานของโปรแกรมคร่าวๆ เพื่อเป็นแนวทางที่จะออกแบบรายละเอียดและเริ่มพัฒนาโปรแกรม โดยขั้นตอนนี้ใช้เวลาตั้งแต่เดือน มิถุนายน 2543 นับจากการเสนอโครงงานปัญหาพิเศษกับอาจารย์ที่ปรึกษาปัญหาพิเศษ จนถึงเดือน สิงหาคม 2543 รวมใช้เวลา 3 เดือน

5.3.2 ขั้นตอนการออกแบบโครงสร้างของโปรแกรม

การออกแบบส่วนโครงสร้างของโปรแกรม , ฐานข้อมูล และสรุปถึงเครื่องมือต่างๆที่ใช้ในการพัฒนาโปรแกรม โดยโครงสร้างและเครื่องมือที่ใช้ในปัญหาพิเศษนี้ ประกอบด้วย

- Borland Delphi 5.0
- Database Desktop
- Borland Database Engine Administrator

การทำงานในขั้นตอนนี้ต้องทดสอบเครื่องมือต่างๆที่จะใช้ดึงค่า Informaion จากอุปกรณ์ในเครือข่าย โดยทดลองใช้ Borland Delphi 5.0 แล้วสามารถดึงค่า Information ได้จึงเลือก Delphi เป็นเครื่องมือในการพัฒนาโปรแกรม

5.3.3 ขั้นตอนการพัฒนาโปรแกรม

สิ่งที่ต้องคำนึงในการพัฒนาโปรแกรมบางครั้งต้องคำนึงถึงปัจจัยที่มีผลกระทบหลายสิ่งหลายอย่าง แต่สิ่งสำคัญที่ต้องพิจารณาคือระยะเวลาในการพัฒนาโปรแกรมโดยจะต้องทำการพัฒนาโปรแกรมให้เสร็จทันตามที่กำหนดและตรงตามเป้าหมาย ซึ่งในบางครั้งอาจจะทำไม่ได้ตามที่ออกแบบไว้ทำให้ต้องมีการปรับเปลี่ยนโครงสร้างหรือเครื่องมือที่ใช้เพื่อให้เหมาะสมและได้ผลลัพธ์ตรงตามเป้าหมายที่ต้องการ

โปรแกรมใช้ Borland Delphi 5.0 เป็นเครื่องมือในการโปรแกรมในส่วนควบคุมหลัก เป็นส่วนที่ใช้ในการติดต่อกับอุปกรณ์ในเครือข่ายที่สนใจและสนับสนุนการทำงานของโปรโตคอล SNMP เหตุผลที่ใช้เครื่องมือนี้เนื่องมาจาก Delphi มีฟังก์ชันในการทำงานที่สามารถรองรับการรับส่งข้อมูลผ่านโปรโตคอล SNMP ได้ นอกจากนี้โปรแกรมใช้ Delphi ในส่วนของการติดต่อกับฐานข้อมูล และติดต่อกับผู้ใช้ และใช้ Database Desktop ในการสร้างฐานข้อมูล เพราะ Database Desktop เป็นเครื่องมือในกลุ่มของ Delphi ใช้สำหรับสร้างฐานข้อมูล สร้างตารางรวมทั้งการจัดการข้อมูลในตาราง โดยใช้โครงสร้างของตารางเป็นแบบ Paradox และใช้ Borland Database Engine (BDE) ซึ่งเป็นเครื่องมือในกลุ่มเดียวกับ Delphi เช่นกัน โดยใช้ BDE ในส่วนของการเชื่อมต่อกับฐานข้อมูล การใช้เครื่องมือในกลุ่มเดียวกันทำให้เข้าใจและใช้งานได้ง่ายขึ้น

5.4 ข้อจำกัดของปัญหาพิเศษ

- แผนที่เครือข่ายที่ได้จะมองเห็นเฉพาะอุปกรณ์ที่สนับสนุนการทำงานของโปรโตคอล SNMP เท่านั้น
- ในการเก็บข้อมูลทางสถิติจำเป็นต้องใช้เนื้อที่ที่พอเพียงกับจำนวนของข้อมูล
- ไม่สามารถรับประกันได้ว่า Agent จะสามารถส่งข้อมูลกลับมาได้ในเวลาที่กำหนด
- รูปของอุปกรณ์ในเครือข่ายที่แสดงถ้ายังไม่มีการจัดเก็บในฐานข้อมูล โปรแกรมจะกำหนดเป็นรูป Default ขึ้นมาซึ่งเป็นหน้าที่ของผู้ดูแลระบบเป็นผู้นำรูปนั้นจัดเก็บในฐานข้อมูล

5.5 ข้อเสนอแนะและแนวทางในการแก้ปัญหา

เนื่องจากโปรแกรมที่สร้างขึ้นเป็นการสร้างแผนที่เครือข่ายโดยใช้โปรโตคอล SNMP ยังมีความสามารถและประสิทธิภาพในการทำงานจำกัด ลักษณะในการทำงานบางโมดูลสามารถปรับปรุงให้ดีขึ้นได้ เช่น

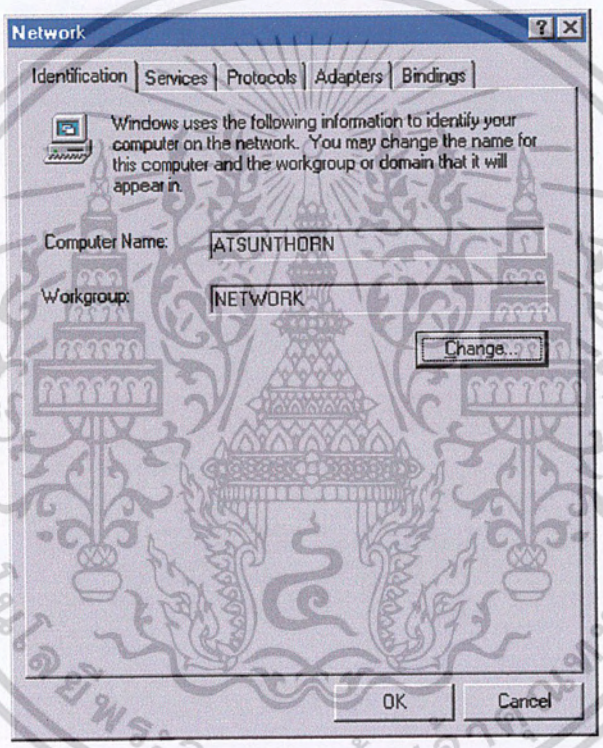
- ส่วนของการแสดงรูป แผนที่เครือข่ายที่ได้ลักษณะของรูปเป็นการแสดงรูปในแนวตั้งและแนวนอน อาจจะไม่ถูกต้องตามหลักการเชื่อมต่อจริงของเครือข่าย
- รูปที่แสดงเป็นการกำหนดพิกัดที่แน่นอนไม่สามารถเคลื่อนย้ายตำแหน่งได้ อาจมีการปรับปรุงให้สามารถเคลื่อนย้ายหรือจัดวางใหม่ได้

ภาคผนวก ก.

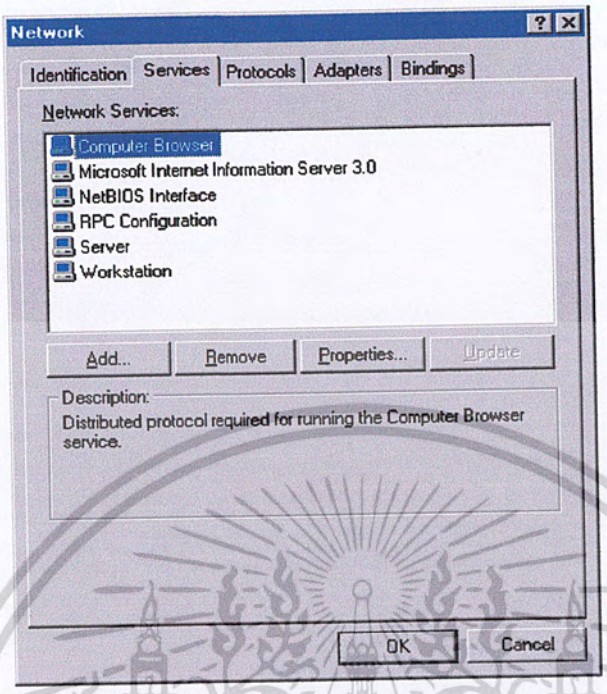
ก.1 การติดตั้ง

ก.1.1 การติดตั้ง SNMP Service

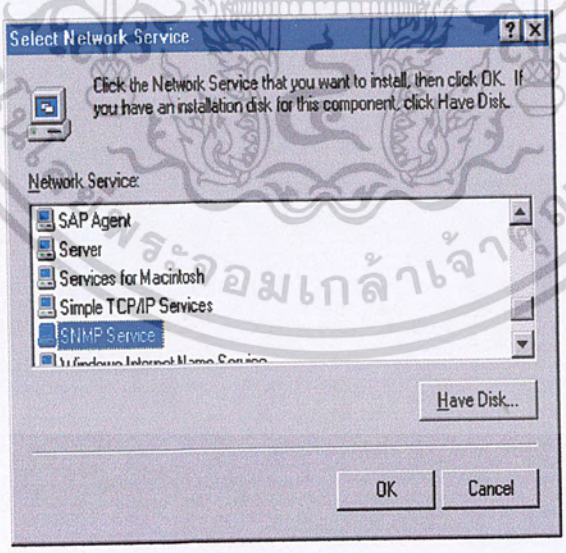
เนื่องจากการทำงานของโปรแกรมจะสามารถดึงค่า Informaion จากอุปกรณ์ในเครือข่ายได้ อุปกรณ์ตัวนั้นจะต้องสนับสนุนการทำงานของโปรโตคอล SNMP การที่จะสร้างแผนที่เครือข่าย ของอุปกรณ์ตัวนั้นได้จะต้องมีการติดตั้ง SNMP Service โดยการเลือก Control Panel และเลือก Network จากนั้นทำตามขั้นตอนดังรูปที่ ก.1 – ก.3



รูปที่ ก.1 แสดงการเข้าสู่การติดตั้ง SNMP Service ของ Window NT



รูปที่ ก.2 แสดงการเลือก Tabsheet Service โดยคลิกปุ่ม Add เมื่อต้องการเพิ่ม Service



รูปที่ ก. 3 แสดงการเลือกติดตั้ง SNMP Service

ภาคผนวก ข.

ข.1 ตัวอย่างฟังก์ชันการดึงค่าข้อมูลจากอุปกรณ์ในเครือข่ายที่สนับสนุนการทำงานของ SNMP Protocol

```
Function AgentGetRequest(LObjectID, LCommunity, LRemoteDotAddr :
str255;LRemotePort : word):GetRequestRec;
var  tObjectID : array [0..14] of Longint;
     pp : pchar;
     str : str255;
begin
  idxString := 0;
  RequestFlag := 1;   // This number indicates that it's AgentGetRequest.
  Resending := 0;
  StrPCopy(first13,LObjectID);  // assign string to First13 pchar global variable.
  first13len := StrLen(first13); // assign length of First13 to First13len(Global).
  GetMem(pp,first13len+1);
  strcopy(pp,first13);
  tObjectID[0] := longint(pp);
  GObjectID := LObjectID;
  GCommunity := LCommunity;
  GRemotePort := LRemotePort;
  GRemoteDotAddr := LRemoteDotAddr;
  strcopy(CurrentObjID,first13);  //copy current objectID.

SNMPFunctionForm.SnmpPlus1.SendGetRequest(LRemoteDotAddr,LRemotePort,LCommu
nity,1,1,@tObjectID);
  KeyFlag := 0;
  repeat
    Application.ProcessMessages;
  until KeyFlag<>0;   // KeyFlag should be Byte better, for many reasons of message.
```

```
Freemem(pp,first13len+1);
if KeyFlag = 1 then
begin
  result.cFlag := 1;
  result.Answer := GAnswer;
end
else
begin
  str := 'Error no.'+inttostr(KeyFlag)+' could not get '+strpas(CurrentObjID)+' from
'+LRemoteDotAddr;
  SNMPFunctionForm.SNMPMemo.Lines.Add(str);
  result.cFlag := KeyFlag;
 end;
end; // end of AgentGetRequest.
```

Function AgentGetNextRequest(LObjectID, LCommunity, LRemoteDotAddr : str255;LRequestID : integer; LRemotePort : word):DataRec;

```
var  tObjectID : array [0..14] of Longint;
    pp : pchar;  str : str255;
    ii : integer;
begin
  RequestFlag := 2;  // This number indicates that it's AgentGetNextRequest.
  Resending := 0;
  idxString := 0;       // inits global variable value.
  StrPCopy(first13,LObjectID);  // assign string to First13 pchar global variable.
  first13len := StrLen(first13); // assign length of First13 to First13len(Global).
  GetMem(pp,first13len+1);
  strcopy(pp,first13);
  tObjectID[0] := longint(pp);
  rim := 1;
  GObjectID := LObjectID;
  GCommunity := LCommunity;
```

```
GRemotePort := LRemotePort;
GRemoteDotAddr := LRemoteDotAddr;
GRequestID := LRequestID;
strcopy(CurrentObjID,first13);  //copy current objectID.

SNMPFunctionForm.SnmpPlus1.SendGetNextRequest(LRemoteDotAddr,LRemotePort,LCo
mmunity,LRequestID,1,@tObjectID);
KeyFlag := 0;
repeat
  Application.ProcessMessages;
until KeyFlag<>0;
Freemem(pp,first13len+1);
if KeyFlag = 1 then
begin
  for ii := 1 to idxstring do
  begin
    result.dObjValue[ii] := pObjValue[ii];
    result.dObjIndex[ii] := pObjIndex[ii];
    result.dObjID[ii] := pObjID[ii];
  end;
  result.didxstring := idxstring;
  result.dFlag := 1;
 end
 else
 begin
   result.dFlag := KeyFlag;
   str := 'Error no.'+inttostr(KeyFlag)+' could not get '+strpas(CurrentObjID)+ ' from
'+LRemoteDotAddr;
   SNMPFunctionForm.SNMPMemo.Lines.Add(str);
 end;
end;
```

```
procedure TForm1.btncpollClick(Sender: TObject);
var data :DataRec; i,j,k,h,l,j1,iii :integer; DataStr,DataStr2 : String[255];
   id,comm,ip : String[255];rp :word;
   check : string[50];
   found,check1 : boolean;
   edt_h1,edt_h2,edt_h3,edt_h4,edt_m4,edt_m3 : string[3];
   ip_network,ip_netmask,ip_str,ip_str_poll :str15;
begin
      memo2.Clear; memo3.Clear; memo_cpoll.Clear;
      Tabsheet1.Enabled := false;
      onepoll.enabled := false;
      btnclose.Enabled := false;
      btncsupport.Enabled := false;
      btncpoll.Enabled := false;
      btninputequipment.enabled := false;
      btnfindinformation.enabled := false;
   //.....clear data in stringgrid.....
      for i := 1 to total_ip do
        begin
          strGrid_SysDetail.Rows[i].Clear;
        end;
      StrGrid_sysdetail.RowCount := 2;
      qryequ.Open;
      qryequ.Active := true;
      qry_onepoll.Open;
      qry_onepoll.Active := true;
      timer1.Enabled := true;
   //.....
      StrGrid_SysDetail.Visible := false;
      memo_cpoll.visible := true;
      edt_host1.enabled := false;
      edt_host2.enabled := false;
```

```
edt_host3.enabled := false;
edt_host4.enabled := false;
edt_mask1.enabled := false;
edt_mask2.enabled := false;
edt_mask3.enabled := false;
edt_mask4.enabled := false;
edt_comm_cpoll.Enabled := false;
edt_cport.Enabled := false;
total_router := 1;
total_ip := 0;
k := 1;
l :=1;
id := '1.3.6.1.2.1.1.';
rp := Strtoint('161');
//..... Poll Information display in memo
if edt_host4.Text = 'x' then
    edt_h4 := '0'
else
    edt_h4 := edt_host4.Text;
if edt_host3.Text = 'x' then
    edt_h3 := '0'
else
    edt_h3 := edt_host3.Text;
if edt_host2.Text = 'x' then
    edt_h2 := '0'
else
    edt_h2 := edt_host2.Text;
edt_h1 := edt_host1.Text;
edt_m4 := edt_mask4.Text;
edt_m3 := edt_mask3.text;
//.....
if strtoint(edt_m3) <> 255 then
```

```
begin
   FindPointLoop(edt_h3,edt_m3);
   edt_h4 := inttostr(1);
   edt_m4 := inttostr(254);
end {case Poll class C}
else if (strtoint(edt_m3) = 255) and (strtoint(edt_m4) = 0 ) then
begin
   edt_h3 := edt_host3.Text;
   edt_m3 := edt_mask3.Text;
   edt_h4 := inttostr(1);
   edt_m4 := inttostr(254);
end
else  if (strtoint(edt_m3) = 255) and (strtoint(edt_m4) <> 0 ) then
   begin
      FindPointLoop(edt_h4,edt_m4);
      edt_h3 := edt_host3.text;
      edt_m3 := edt_host3.text;
      ip_netmask := edt_mask1.Text +
'.'+edt_mask2.Text+'.'+edt_mask3.Text+'.'+edt_mask4.Text;
   end;
//.....................................
   ip_network := edt_h1+'.'+edt_h2+'.';
   for h := strtoint(edt_h3) to strtoint(edt_m3) do
   begin
    for j := strtoint(edt_h4) to strtoint(edt_m4) do
      begin
        ip := ip_network + inttostr(h) + '.' + inttostr(j);
        memo_cpoll.lines.add('IP  :'+ip);
        comm := edt_comm_cpoll.Text;
        data := AgentGetNextRequest(id,comm,ip,1,rp);
        if data.dFlag = 1 Then
          begin
```

```
CPollData[k].IPAddress := ip;
arr_ipcheck[l].Ipaddress := ip;
arr_ipcheck[l].Status := 'ON';
l := l+1;
total_ip := total_ip+1;
memo_cpoll.Lines.Add('ObjectID ObjectValue');
for i := 1 to data.didxString do
 begin
  DataStr := data.dObjID[i]+' : '+data.dObjValue[i];
  DataStr2 :=  data.dObjValue[i];
  memo_cpoll.Lines.Add('Get=> '+IntToStr(i)+' : '+DataStr);
  with CPollData[k] do
  begin
   case i of
    1 : Description := DataStr2;
    2 : begin
        ObjectID := DataStr2;
        arr_objid[l-1] := DataStr2;
        end;
    3 : Uptime := DataStr2;
    4 : Contact_Name := DataStr2;
    5 : Sys_Name := DataStr2;
    6 : Location := DataStr2;
   end; {case}
  end; {with}
 end; {for}
  memo_cpoll.Lines.add('.....');
  CPollData[k].Date_To_Poll := date;
  CPollData[k].Time_to_Poll := time;
  CPollData[k].status := 'ON';
  k := k+1;
end {if}
```

```
else
  begin
//.....set community name for router
    comm := edt_router.Text;
 //.... Poll Information router display in memo
    data := AgentGetNextRequest(id,comm,ip,1,rp);
    if data.dFlag = 1 Then
     begin
       CPollData[k].IPAddress := ip;
       arr_ipcheck[l].Ipaddress := ip;
       arr_ipcheck[l].Status := 'ON';
       l := l+1;
       total_ip := total_ip+1;
       arr_router[total_router] := ip;
       total_router := total_router+1;
       memo_cpoll.Lines.Add('ObjectID ObjectValue');
       for i := 1 to data.didxString do
       begin
        DataStr := data.dObjID[i]+'   '+data.dObjValue[i];
        DataStr2 :=  data.dObjValue[i];
        memo_cpoll.Lines.Add('Get=> '+IntToStr(i)+' : '+DataStr2);
        with CPollData[k] do
        begin
         case i of
         1 : Description := DataStr2;
         2 : begin
          ObjectID := DataStr2;
          arr_objid[l-1] := DataStr2;
           end;
         3 : Uptime := DataStr2;
         4 : Contact_Name := DataStr2;
         5 : Sys_Name := DataStr2;
```

```
                6 : Location := DataStr2;
                end; {case}
              end; {with}
            end; {for}
            memo_cpoll.Lines.add('.....');
            CPollData[k].Date_To_Poll := date;
            CPollData[k].Time_to_Poll := time;
            CPollData[k].status := 'ON';
            k := k+1;
          end
          else
           begin
            memo_cpoll.Lines.Add('Not Support SNMP Protocol !!! ');
            memo_cpoll.lines.add('Try To Poll Again :-) ');
            memo_cpoll.lines.add('.....');
            check1 := qry_onepoll.Locate('IPAddress',ip,[loCaseInsensitive]);
            if check1 = true then
             begin
              if qry_onepoll['status'] = 'ON' then
               begin
                arr_ipcheck[l].Ipaddress := ip;
                arr_ipcheck[l].Status := 'OFF';
                arr_objid[l] := qry_onepoll['ObjectId'];
                l := l+1;
               end;
             end;
            end;
          end;
          total_ip1 := l-1;
      end; {case class c}
    end; {case class B}
//.....End loop for.....
```

```
//.....Keep Picture In Array.....
        for i := 1 to total_ip1 do
         begin
          found := QryEqu.Locate('Object_ID',arr_objid[i],[loCaseInsensitive]);
            if found = true then
              begin
                check :=  QryEqu['Bitmap'];
                arr_bitmap[i] := check;
              end {if found }
              else
              arr_bitmap[i] := 'default.bmp';
         end;
        //.....
        for i := 1 to total_router-1 do
         begin
          memo2.Lines.Add('IP Router : '+ arr_router[i] );
         end;

        poll_router;
        memo_cpoll.lines.add('Total SNMP Agent Support :  '+inttostr(total_ip));
        Btncshow.Enabled := true;
        btncadd.Enabled := true;
        btncdelete.Enabled := true;
        btnportdata.Enabled := true;
        qry_onepoll.Close;
        qry_onepoll.Active := false;
        qryequ.Close;
        qryequ.Active := false;
        edt_host3.Enabled := true;
        edt_host4.Enabled := true;
        edt_mask4.Enabled := true;
        btnclose.Enabled := true;
```

```
        btnonepoll.Enabled := true;
        btncsupport.Enabled := true;
        btncclear.Enabled := true;
        Tabsheet1.Enabled := true;
        onepoll.enabled := true;
        btncpoll.Enabled := true;
        btninputequipment.Enabled := true;
        btnfindinformation.Enabled := true;
end;
```

ข.2 ตัวอย่างฟังก์ชันการเก็บค่าข้อมูลจากอุปกรณ์ในเครือข่ายที่สนับสนุนการทำงานของ SNMP Protocol

```
procedure TForm1.btncaddClick(Sender: TObject);
var  i,j,h  : integer;
    check1,check2 : boolean;
    ip : string;
    edt_h1,edt_h2,edt_h3,edt_h4,edt_m4,edt_m3 : string[3];
```

```
  ip_network,ip_netmask :str15;
begin
      qry_onepoll.open;
      qry_onepoll.sql.clear;
      qry_onepoll.SQL.Add('Select * from systemdata.db ');
      qry_onepoll.open;
      i := 1;
      //.....
        if edt_host4.Text = 'x' then
        edt_h4 := '0'
      else
        edt_h4 := edt_host4.Text;
      if edt_host3.Text = 'x' then
        edt_h3 := '0'
      else
        edt_h3 := edt_host3.Text;
      if edt_host2.Text = 'x' then
        edt_h2 := '0'
      else
        edt_h2 := edt_host2.Text;
      edt_h1 := edt_host1.Text;
      edt_m4 := edt_mask4.Text;
      edt_m3 := edt_mask3.text;
  //.....
      if strtoint(edt_m3) <> 255 then
      begin
        FindPointLoop(edt_h3,edt_m3);
        edt_h4 := inttostr(1);
        edt_m4 := inttostr(254);
      end {case Poll class C}
      else if (strtoint(edt_m3) = 255) and (strtoint(edt_m4) = 0 ) then
      begin
```

```
        edt_h3 := edt_host3.Text;
        edt_m3 := edt_mask3.Text;
        edt_h4 := inttostr(1);
        edt_m4 := inttostr(254);
      end
      else  if (strtoint(edt_m3) = 255) and (strtoint(edt_m4) <> 0 ) then
        begin
          FindPointLoop(edt_h4,edt_m4);
          edt_h3 := edt_host3.text;
          edt_m3 := edt_host3.text;
          ip_netmask := edt_mask1.Text +
'.'+edt_mask2.Text+'.'+edt_mask3.Text+'.'+edt_mask4.Text;
        end;
    //.....
     ip_network := edt_h1+'.'+edt_h2+'.';
     for h := strtoint(edt_h3) to strtoint(edt_m3) do
     begin
      for j := strtoint(edt_h4) to strtoint(edt_m4) do
        begin
          ip := ip_network + inttostr(h) + '.' + inttostr(j);
    //.....case support SNMP.....
          if ip = CPollData[i].IPAddress then
           begin
            check1 := qry_onepoll.Locate('IPAddress',ip,[loCaseInsensitive]);
            if check1 = true then
             begin
    // Replace Information suport SNMP in qry_onepoll from qry_cpoll
              qry_onepoll.edit;
              qry_onepoll['IPAddress'] := ip;
              if CPollData[i].Description = '' then
                qry_onepoll['Description']  := '***'
              else qry_onepoll['Description'] := CPollData[i].Description;
```

```
if CPollData[i].ObjectID = '' then
   qry_onepoll['ObjectID']  := '***'
else qry_onepoll['ObjectID'] := CPollData[i].ObjectID;
if CPollData[i].Uptime = '' then
   qry_onepoll['Uptime']  := '***'
else qry_onepoll['Uptime'] := CPollData[i].Uptime;
if CPollData[i].Contact_Name = '' then
   qry_onepoll['Contact_Name']  := '***'
else qry_onepoll['Contact_Name'] := CPollData[i].Contact_Name;
if CPollData[i].Sys_Name = '' then
   qry_onepoll['Sys_Name']  := '***'
else qry_onepoll['Sys_Name'] := CPollData[i].Sys_Name;
if CPollData[i].Location = '' then
   qry_onepoll['Location']  := '***'
else qry_onepoll['Location'] := CPollData[i].Location;
qry_onepoll['Date_To_Poll'] := CPollData[i].Date_To_Poll;
qry_onepoll['Time_To_Poll'] := CPollData[i].Time_To_Poll;
qry_onepoll['status'] := 'ON';
i := i+1;
end {if check1 }
else
begin
 // Add Information support SNMP in qry_onepoll from qry_cpoll
 qry_onepoll.append;
 qry_onepoll['IPAddress'] := ip;
 if CPollData[i].Description = '' then
    qry_onepoll['Description']  := '***'
 else qry_onepoll['Description'] := CPollData[i].Description;
 if CPollData[i].ObjectID = '' then
    qry_onepoll['ObjectID']  := '***'
 else qry_onepoll['ObjectID'] := CPollData[i].ObjectID;
 if CPollData[i].Uptime = '' then
```

```
        qry_onepoll['Uptime'] := '***'
      else qry_onepoll['Uptime'] := CPollData[i].Uptime;
      if CPollData[i].Contact_Name = '' then
        qry_onepoll['Contact_Name'] := '***'
      else qry_onepoll['Contact_Name'] := CPollData[i].Contact_Name;
      if CPollData[i].Sys_Name = '' then
        qry_onepoll['Sys_Name'] := '***'
      else qry_onepoll['Sys_Name'] := CPollData[i].Sys_Name;
      if CPollData[i].Location = '' then
        qry_onepoll['Location'] := '***'
      else qry_onepoll['Location'] := CPollData[i].Location;
      qry_onepoll['Date_To_Poll'] := CPollData[i].Date_To_Poll;
      qry_onepoll['Time_To_Poll'] := CPollData[i].Time_To_Poll;
      qry_onepoll['status'] := 'ON';
       i := i+1;
      end; {else if check1}
    end {if support = true }
//.....case not support .....
    else if ip <> CPollData[i].IPAddress then
     begin
     check2 := qry_onepoll.Locate('IPAddress',ip,[loCaseInsensitive]);
     if check2 = true then
      begin
// Replace Information Not Support SNMP in qry_onepoll
       qry_onepoll.edit;
       qry_onepoll['IPAddress'] := ip;
       qry_onepoll['Date_To_Poll'] :=  date;
       qry_onepoll['Time_To_Poll'] := time;
       qry_onepoll['status'] := 'OFF';
      end {if check2}
      else
       begin
```

```
    // Add Information Not Support SNMP in qry_onepoll
                qry_onepoll.append;
                qry_onepoll['IPAddress'] := ip;
                qry_onepoll['ObjectID'] := 'xxx';
                qry_onepoll['Date_To_Poll'] :=  date;
                qry_onepoll['Time_To_Poll'] := time;
                qry_onepoll['status'] := 'OFF';
               end; {else if check2}
          end; {else Cpolldata}
            qry_onepoll.post;
            qry_onepoll.Close;
            qry_onepoll.Open;
     end;  {for j}
end;{for i}
showmessage('Add To Database Complete!!!!');
btncmap.Enabled := true;
btncpoll.Enabled := true;
btnmaprouter.Enabled := true;
end;
```

ข.3 ตัวอย่างฟังก์ชันการแสดงภาพอุปกรณ์ในเครือข่ายที่สนับสนุนการทำงานของ SNMP Protocol

```
procedure TForm1.NetworkMap(total_router : integer);
var   r : Trect;
      bitmap : Tbitmap;
      st_x,st_y,fin_x,fin_y : integer;
      st_lx,st_ly,fin_lx,fin_ly  : integer;
      r_x1,r_y1,r_x2,r_y2 : integer;
      i,j,i1,j1,total_router1,x1,p,x,p1 : integer;
      x_let,y_let : integer;
      host : string;
```

```
    check : boolean;
    ip_network,subnet_mask : str15;
begin
 total_router1 := total_router -1;
 Bitmap := TBitmap.Create;        { construct the bitmap object }
 Bitmap.Width := 3500;  { assign the initial width... }
 Bitmap.Height := 150 * total_router1; { ...and the initial height }
 img_router.Picture.Graphic := Bitmap;    { assign the bitmap to the image component }
 img_router.Canvas.Pen.Color := clblue;
 img_router.Canvas.Font.Name := 'cordia new ';
 img_router.Canvas.Font.Size := 7;
 img_router.Canvas.Font.Color := clgreen;
 x := 1;
//.....
    r_x1 := 10;
    r_y1 := ( ((40*total_router1)+(100* (total_router1))) div 2) - 10;
    r_x2 := r_x1 + 40;
    r_y2 := r_y1 + 40;
 //.....draw host address.....
    bitmap := Tbitmap.Create;
    bitmap.LoadFromFile('subnet1.bmp');
    r := rect(r_x1,r_y1,r_x2,r_y2);
    img_router.Canvas.StretchDraw(r,bitmap);
    ip_network := edt_host1.text + '.' + edt_host2.Text + '.' + edt_host3.Text
 +'.'+edt_host4.Text;
    subnet_mask := edt_mask1.Text + '.' + edt_mask2.Text + '.' + edt_mask3.Text
 +'.'+edt_mask4.Text;
    img_router.Canvas.Textout(r_x1,r_y1+40,ip_network + '/');
    img_router.Canvas.TextOut(r_x1,r_y1+52,subnet_mask);
    scrollreset;
 //.....
    st_x := 90;
```

```
st_y := 10;
fin_x := 130;
fin_y := 50;
st_lx := 130;
st_ly := 30;
fin_lx := 190;
fin_ly := 30;
x_let := st_lx;
y_let := st_ly;
//.....
for i1 := 1 to total_router1 do
 begin
    check := false;
 //.....draw line from host to router.....
    img_router.Canvas.MoveTo(r_x2,r_y1);
    img_router.Canvas.lineto(st_x,st_y+20);
 //.....draw router.....
    bitmap := Tbitmap.Create;
    bitmap.LoadFromFile('router_d.bmp');
    r := rect(st_x,st_y,fin_x,fin_y);
    img_router.Canvas.StretchDraw(r,bitmap);
    scrollreset;
    img_router.Canvas.TextOut(st_x,st_y-2,arr_router[i1]);

    host := edt_host1.text+'.'+edt_host2.text+'.'+edt_host3.Text+'.0';
    p1 := 0;
    qry_router.Active := true;
    qry_router.SQL.clear;
    qry_router.SQL.Add('select * from router_table.db where (Networkclass = "'+host+'"
and Iprouter = "'+arr_router[i1]+'" and Ipsubclass = "'+host+'") ');
    qry_router.Open;
    qry_router.First;
```

```
while p1 < qry_router.RecordCount do
 begin
 img_router.Canvas.TextOut(st_x-3,fin_y+3,'No: ' + qry_router['Interface_no']);
  qry_router.Next;
  p1 := p1+1;
 end;
qry_router.Active := false;
qry_router.close;
//.....
for j1 := 1 to arr_total_port[i1] do
 begin
  host := edt_host1.text+'.'+edt_host2.text+'.'+edt_host3.Text+'.0';
  p := 0;
  qry_router.Active := true;
  qry_router.SQL.clear;
  qry_router.SQL.Add('select * from router_table.db where (Networkclass =
'"+host+'" and Iprouter = '"+arr_router[i1]+'" and Interface_no = '"+arr_interface1[i1,j1]+'"
and Ipsubclass <> '"+host+'") ');
  qry_router.Open;
  qry_router.First;
  x1 := 48;
  if qry_router.RecordCount > 0 then
    check := true;
  while (check = true) do
  begin
  //.....draw subnet.....
    while p < qry_router.RecordCount do
      begin
        img_router.Canvas.MoveTo(st_lx,st_ly);
        img_router.Canvas.lineto(fin_lx,fin_ly);
       //.....draw subnet of port.....
        bitmap := Tbitmap.Create;
```

```
bitmap.LoadFromFile('subnet.bmp');
r := rect(st_lx+10,st_ly+10,st_lx+50,st_ly+50);
img_router.Canvas.StretchDraw(r,bitmap);
scrollreset;
//.....
SizeIpsubclass[x].IPRouter := arr_router[i1];
SizeIpsubclass[x].InterfaceNo := arr_interface1[i1,j1];
SizeIpsubclass[x].St_x := st_lx+10;
SizeIpsubclass[x].St_y := st_ly+10;
SizeIpsubclass[x].Fin_x := st_lx+50;
SizeIpsubclass[x].Fin_y := st_ly+50;
//.....draw line.....
img_router.Canvas.MoveTo(st_lx+30,st_ly);
img_router.Canvas.lineto(st_lx+30,st_ly+20);
//.....write interface no.....
img_router.Canvas.Textout(st_lx+33,st_ly+3,'No : '+
qry_router['Interface_no']);
img_router.Canvas.TextOut(st_lx+8,st_ly+x1,qry_router['Ipsubclass']);
qry_router.Next;

//.....
p := p+1;
x1 := x1 + 12;
end; {while loop}
check := false;
st_lx := fin_lx;
fin_lx := fin_lx + 60;
end; {while loop check}
//.....
x := x+ 1;
total_ipsub := x-1;
end;{for j1}
```

```
      //.....set position new.....
        r_y1 := r_y1 + (40 div (total_router1));
        st_lx := 130;
        st_ly := st_ly + 140;
        fin_lx := 190;
        fin_ly := fin_ly +140;
        st_y := st_y +140;
        fin_y := fin_y +140;
      end; {for i1 }
      scrollreset;
end;

procedure TForm1.btnmaprouterClick(Sender: TObject);
var i,j,n,m : integer;
   check,check1 : boolean;
   host : string;
begin
  //.....keep group ipsubclass in array arr_groupip .....
      check1 := true;
      for i := 1 to total_router-1 do
       begin
        m := 1;
        for n := 1 to 50 do
         begin
          for j := 1 to arr_port_no[i] do
           begin
            if (strtoint(arr_interface[i,j]) = n) and (check1 = true) then
              begin
               arr_interface1[i,m] := inttostr(n) ;
               m := m+1;
               check1 := false;
              end;
```

```
            end;{for j}
            check1 := true;
          end;{for n  }
          arr_total_port[i]  :=  m-1;
        end; {for i}
 //.....Add Information Router to Database
   qry_router.Active := true;
   table_router.open;
   table_router.Active := true;
    for i := 1 to total_router-1 do
      begin
       host := edt_host1.text+'.'+edt_host2.text+'.'+edt_host3.Text+'.0';
       qry_router.SQL.clear;
       qry_router.SQL.Add('select * from router_table.db where (Networkclass = '"+host+'"
and Iprouter = '"+arr_router[i]+'")');
       qry_router.Open;
       for j := 1 to arr_port_no[i] do
        begin
         check := table_router.Locate('Ipgateway',arr_gateway[i,j],[loCaseInsensitive]);
          if check = true then
           begin
              table_router.Edit;
              table_router['Networkclass'] := host;
              table_router['Iprouter'] := arr_router[i];
              table_router['Ipgateway'] := arr_gateway[i,j];
              table_router['Ipnetmask'] := arr_netmask[i,j];
              table_router['Ipsubclass'] := arr_ipsubclass[i,j];
              table_router['Interface_no'] := arr_interface[i,j];
              table_router.post;
           end
          else
           begin
```

```
                table_router.Append;
                table_router['Networkclass'] := host;
                table_router['Iprouter'] := arr_router[i];
                table_router['Ipgateway'] := arr_gateway[i,j];
                table_router['Ipnetmask'] := arr_netmask[i,j];
                table_router['Ipsubclass'] := arr_ipsubclass[i,j];
                table_router['Interface_no'] := arr_interface[i,j];
                table_router.Post;
              end;
        end; {for j}
    end; {for i }
    table_router.Close;
    table_router.Active := false;
    qry_router.Active := false;
    qry_router.Close;
    pagecontrol1.Pages[4].Show;
    img_router.Picture.LoadFromFile('screen.bmp');
    Networkmap(total_router);
    scrollreset;
end;

procedure TForm1.line(ip_count : integer);
var
  r,r1 : Trect;
  bitmap : Tbitmap;
  ip : string;
  st_x,st_y,fin_x,fin_y : integer;
  line_h,line_v,i,j,jj,jj1 : integer;
begin
      line_h := ip_count div 6;
      Bitmap := TBitmap.Create; { construct the bitmap object }
      Bitmap.Width := 650;      { assign the initial width... }
```

```
Bitmap.Height := 150 * (line_h+1);  { ...and the initial height }
img_map.Picture.Graphic := Bitmap;
if ip_count mod 6 <> 0 then
   line_h := line_h + 1;
line_v := line_h -1;
img_map.Canvas.Pen.Color := clblue;
st_y :=  20;
for i := 0 to line_h -1 do
 begin
  img_map.Canvas.MoveTo(20,st_y);
  img_map.Canvas.lineto(440,st_y);
  st_y := st_y + 120;
 end;
//.....
  st_y := 20;
  for i := 0 to line_v -1 do
  begin
  j := (i +1) mod 2;
  case j of
  1 : begin
       img_map.Canvas.MoveTo(440,st_y);
       img_map.Canvas.lineto(440,st_y+120);
      end;
  0 : begin
       img_map.Canvas.MoveTo(20,st_y);
       img_map.Canvas.lineto(20,st_y+120);
      end;
  end;
  st_y := st_y + 120;
  end;{for}
//.....
  jj := 1;
```

```
st_x := 55;
st_y := 20;
fin_x := 55;
fin_y := 50;
for i := 0 to line_h-1 do
begin  {line1}
   if (i= 0) and (ip_count <= 6) then
      begin
         for j:= 0 to ip_count-1 do
         begin
          img_map.Canvas.MoveTo(st_x,st_y);
          img_map.Canvas.lineto(fin_x,fin_y);
          if arr_ipcheck[jj].Status = 'ON' then
          begin
           //.....Draw Equipment That Suppot.....
            img_map.canvas.brush.color := clgreen;
            img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
            checksize[jj].st_x := fin_x-30;
            checksize[jj].st_y := fin_y-5;
            checksize[jj].fin_x := fin_x+30;
            checksize[jj].fin_y := fin_y+55;

          end
          else  //.....Case Status Not Support
          begin
              //.....Draw Equipment That Not Support.....
            img_map.canvas.brush.color := clred;
            img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
            checksize[jj].st_x := fin_x-30;
            checksize[jj].st_y := fin_y-5;
            checksize[jj].fin_x := fin_x+30;
            checksize[jj].fin_y := fin_y+55;
```

```
        end;
      bitmap := Tbitmap.Create;
      bitmap.LoadFromFile(arr_bitmap[jj]);
      r := rect(fin_x-25,fin_y,fin_x+25,fin_y+50);
      img_map.Canvas.StretchDraw(r,bitmap);
      img_map.canvas.brush.Style := bsclear;
      scrollreset;

      img_map.canvas.font.Name := 'cordia new';
      img_map.canvas.Font.Size := 10;
      img_map.canvas.Font.Color := clblue;
      img_map.canvas.TextOut(fin_x-25,fin_y+58,arr_ipcheck[jj].Ipaddress);
      st_x := st_x+70;
      fin_x := fin_x+70;
      jj := jj +1;
     end;
   end
 else if (i = 0) and (ip_count > 6) then
   begin
    for j:= 0 to 5 do
    begin
      img_map.Canvas.MoveTo(st_x,st_y);
      img_map.Canvas.lineto(fin_x,fin_y);
    //.....Draw Equipment That Support.....
      if arr_ipcheck[jj].Status = 'ON' then
        begin
          img_map.canvas.brush.color := clgreen;
          img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
          checksize[jj].st_x := fin_x-30;
          checksize[jj].st_y := fin_y-5;
          checksize[jj].fin_x := fin_x+30;
```

```
        checksize[jj].fin_y := fin_y+55;
      end
      else  //.....Case Status Not Support
       begin
        //.....Draw Equipment That Not Support.....
        img_map.canvas.brush.color := clred;
        img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
        checksize[jj].st_x := fin_x-30;
        checksize[jj].st_y := fin_y-5;
        checksize[jj].fin_x := fin_x+30;
        checksize[jj].fin_y := fin_y+55;
         end;
       bitmap := Tbitmap.Create;
       bitmap.LoadFromFile(arr_bitmap[jj]);
       r := rect(fin_x-25,fin_y,fin_x+25,fin_y+50);
       img_map.Canvas.StretchDraw(r,bitmap);
       scrollreset;
     //  img_map.canvas.brush.color := clblue;
       img_map.canvas.font.Name := 'cordia new';
       img_map.canvas.Font.Size := 10;
       img_map.canvas.brush.Style := bsclear;
       img_map.canvas.Font.Color := clblue;
       img_map.canvas.TextOut(fin_x-25,fin_y+58,arr_ipcheck[jj].Ipaddress);
       st_x := st_x+70;
       fin_x := fin_x+70;
       jj := jj + 1;
    end;
  end
else if ( i>0 ) and (i <> line_h -1) then
  begin
   for j:= 0 to 5 do
     begin
```

```
img_map.Canvas.MoveTo(st_x,st_y);
img_map.Canvas.lineto(fin_x,fin_y);
if arr_ipcheck[jj].Status = 'ON' then
 begin
//.....Draw Equipment That Suppot.....
  img_map.canvas.brush.color := clgreen;
  img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
  checksize[jj].st_x := fin_x-30;
  checksize[jj].st_y := fin_y-5;
  checksize[jj].fin_x := fin_x+30;
  checksize[jj].fin_y := fin_y+55;
 end
 else  //.....Case Status Not Support
  begin
    //.....Draw Equipment That Not Support.....
   img_map.canvas.brush.color := clred;
   img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
   checksize[jj].st_x := fin_x-30;
   checksize[jj].st_y := fin_y-5;
   checksize[jj].fin_x := fin_x+30;
   checksize[jj].fin_y := fin_y+55;
     end;
 bitmap := Tbitmap.Create;
 bitmap.LoadFromFile(arr_bitmap[jj]);
 r := rect(fin_x-25,fin_y,fin_x+25,fin_y+50);
 img_map.Canvas.StretchDraw(r,bitmap);
 scrollreset;
//   img_map.canvas.brush.color := clblue;
 img_map.canvas.font.Name := 'cordia new';
 img_map.canvas.Font.Size := 10;
 img_map.canvas.brush.Style := bsclear;
 img_map.canvas.Font.Color := clblue;
```

```
            img_map.canvas.TextOut(fin_x-25,fin_y+58,arr_ipcheck[jj].IPAddress);
            st_x := st_x+70;
            fin_x := fin_x+70;
            jj := jj + 1;
          end;
      end
{line end}
else if (i <> 0) and (i = line_h-1) then
 begin
  if ip_count mod 6 <> 0 then
  begin
    for j:= 0 to (ip_count mod 6)-1 do
    begin
      img_map.Canvas.MoveTo(st_x,st_y);
      img_map.Canvas.lineto(fin_x,fin_y);
      if arr_ipcheck[jj].Status = 'ON' then
       begin
       //.....Draw Equipment That Suppot.....
        img_map.canvas.brush.color := clgreen;
        img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);

        checksize[jj].st_x := fin_x-30;
        checksize[jj].st_y := fin_y-5;
        checksize[jj].fin_x := fin_x+30;
        checksize[jj].fin_y := fin_y+55;
       end
       else  //.....Case Status Not Support
       begin
         //.....Draw Equipment That Not Support.....
        img_map.canvas.brush.color := clred;
        img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
        checksize[jj].st_x := fin_x-30;
```

```
        checksize[jj].st_y := fin_y-5;
        checksize[jj].fin_x := fin_x+30;
        checksize[jj].fin_y := fin_y+55;
          end;
      bitmap := Tbitmap.Create;
      bitmap.LoadFromFile(arr_bitmap[jj]);
      r := rect(fin_x-25,fin_y,fin_x+25,fin_y+50);
      img_map.Canvas.StretchDraw(r,bitmap);
      scrollreset;
      img_map.canvas.font.Name := 'cordia new';
      img_map.canvas.Font.Size := 10;
      img_map.canvas.brush.Style := bsclear;
      img_map.canvas.Font.Color := clblue;
      img_map.canvas.TextOut(fin_x-25,fin_y+58,arr_ipcheck[jj].Ipaddress);
      st_x := st_x+70;
      fin_x := fin_x+70;
      jj := jj +1;
    end;
  end
  else
    for j:= 0 to 5 do
    begin
      img_map.Canvas.MoveTo(st_x,st_y);
      img_map.Canvas.lineto(fin_x,fin_y);
      if arr_ipcheck[jj].Status = 'ON' then
    begin
      //.....Draw Equipment That Suppot.....
        img_map.canvas.brush.color := clgreen;
        img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
        checksize[jj].st_x := fin_x-30;
        checksize[jj].st_y := fin_y-5;
        checksize[jj].fin_x := fin_x+30;
```

```
        checksize[jj].fin_y := fin_y+55;


      end
      else  //.....Case Status Not Support
      begin
          //.....Draw Equipment That Not Support.....
        img_map.canvas.brush.color := clred;
        img_map.canvas.Rectangle(fin_x-30,fin_y-5,fin_x+30,fin_y+55);
        checksize[jj].st_x := fin_x-30;
        checksize[jj].st_y := fin_y-5;
        checksize[jj].fin_x := fin_x+30;
        checksize[jj].fin_y := fin_y+55;
        end;
       bitmap := Tbitmap.Create;
       bitmap.LoadFromFile(arr_bitmap[jj]);
       r := rect(fin_x-25,fin_y,fin_x+25,fin_y+50);
       img_map.Canvas.StretchDraw(r,bitmap);
       scrollreset;
       img_map.canvas.font.Name := 'cordia new';
       img_map.canvas.Font.Size := 10;
       img_map.canvas.brush.Style := bsclear;
       img_map.canvas.Font.Color := clblue;
       img_map.canvas.TextOut(fin_x-25,fin_y+58,arr_ipcheck[jj].IPAddress);
       st_x := st_x+70;
       fin_x := fin_x+70;
       jj := jj +1;
      end;
 end;
st_x := 55;
fin_x := 55;
st_y := st_y+120;
fin_y := fin_y+120;
```

```
        end; {Loop Main For}
        scrollreset;
end;

procedure TForm1.btncmapClick(Sender: TObject);
var
  bitmap : Tbitmap;
begin
   //.....check picture in database equpiment.....
        qryequ.Open;
        qryequ.Active := true;
        qryequ.close;
        qryequ.Active := false;
        Bitmap := TBitmap.Create; { construct the bitmap object }
        Bitmap.Width := 800;      { assign the initial width... }
        Bitmap.Height := 1500;    { ...and the initial height }
        img_map.Picture.Graphic := Bitmap;          { assign the bitmap to the image
component }
        line(total_ip1);
        scrollreset;
        pagecontrol1.Pages[6].Show;
end;
```

บรรณานุกรม

[1] COHEN, Yorun. SNMP-Simple Network Management Protocol.[online]. Available : http://194.87.208.92/tech1/1995/snmp/snmp.htm

[2] Feit, S. 1995. SNMP a Guide to Network Management. McGRAW-HILL Book Co.,

สุวัฒน์ ปุณณชัยยะ , ต้น ตันฑ์สุทธิวงศ์ และ สุพจน์ ปุณณชัยยะ. เปิดโลกของ TCP/IP และโปรโตคอลของอินเตอร์เน็ต.บริษัทโปรวิชั่นจำกัด.

[3] จักรพงษ์ สุขประเสริฐ ,กนกพร ภาวศุทธิกุล และ สัจจะ จรัสรุ่งรวีวร. คู่มือการเขียนโปรแกรมด้วย Delphi 4.0 ฉบับสมบูรณ์.สำนักพิมพ์อินโฟเพรส.

[4] กนก กุสุมาลย์นุกูล และ ไกรวุฒิ มั่นเสถียรสิน . คู่มือการเขียนโปรแกรมด้วย Delphi 4.0 บริษัทซัคเซสมีเดียจำกัด

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง