ระบบค้นหาเครือข่ายโดยใช้โปรโตคอล SNMP ผ่านเว็บ

WEB-BASED NETWORK DISCOVERY SYSTEM

โดย

ณัฐวิชช์ ว่องสิทธิโรจน์

สุรีย์นารถ เกียรติสาโรจน์

อาจารย์ที่ปรึกษา

อาจารย์ลภัส ประดิษฐ์ทัศนีย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2551

WEB-BASED NETWORK DISCOVERY SYSTEM

NUTTAWIT WONGSITTIROJ

SUREENART KIETSAROCH

A PROJECT SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2008

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2551

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบค้นหาเครือข่ายโดยใช้โปรโตคอลเอสเอ็นเอ็มพีผ่านเว็บ

WEB-BASED NETWORK DISCOVERY SYSTEM

ผู้จัดทำ

1. นายณัฐวิชช์ ว่องสิทธิโรจน์ รหัสนักศึกษา 48070057
2. นางสาวสุรีย์นารถ เกียรติสาโรจน์ รหัสนักศึกษา 48070083

.....อาจารย์ที่ปรึกษา

(อาจารย์ลภัส ประดิษฐ์ทัศนีย์)

หัวข้อวิทยานิพนธ์	ระบบค้นหาเครือข่ายโดยใช้โปรโตคอล SNMP ผ่านเว็บ
นักศึกษา	นายณัฐวิชช์ ว่องสิทธิโรจน์
	นางสาวสุรีย์นารถ เกียรติสาโรจน์
รหัสนักศึกษา	48070057
	48070083
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2551
อาจารย์ที่ปรึกษา	อาจารย์ลภัส ประดิษฐ์ทัศนีย์

บทคัดย่อ

ในปัจจุบันมีการใช้เครือข่ายการสื่อสารข้อมูลภายในองค์กรต่างๆอย่างแพร่หลาย ทั้งองค์กรของรัฐและองค์กรของเอกชน การใช้งานเครือข่ายการสื่อสารข้อมูลไม่ได้จำกัดอยู่แค่ภายในองค์กรเท่านั้น แต่ยังจะมีการสื่อสารข้อมูลระหว่างองค์กรด้วยโดยผ่านอินเตอร์เน็ต ดังนั้นเครือข่ายการสื่อสารข้อมูลจึงกลายเป็นส่วนประกอบสำคัญขององค์กรที่ขาดไม่ได้ ทำให้จำเป็นที่จะต้องมีเครื่อง -มือที่ช่วยสนับสนุนการจัดการเครือข่ายซึ่งจะทำให้การบริหารงานและจัดการเครือข่ายสะดวกรวดเร็วและมีประสิทธิภาพยิ่งขึ้น

ซึ่งปัจจุบันมีโปรแกรมมากมายที่ช่วยสนับสนุนการจัดการเครือข่ายซึ่งจะทำให้การบริหารงานและจัดการเครือข่ายสะดวกรวดเร็วและมีประสิทธิภาพยิ่งขึ้น โดยทั่วไปโปรแกรมจะมีการทำงานที่ต้องมีการใส่หมายเลขไอพีแอดเดรสทำให้เกิดปัญหากับผู้ดูแลระบบที่ไม่ทราบข้อมูลของเครือข่ายมาก่อน จึงมีการศึกษาและการทดลองกระบวนการค้นหาหมายเลขไอพีแอดเดรส แบบต่างๆ จนได้กระบวนการค้นหาหมายเลข ไอพีแอดเดรสแบบอัตโนมัติและในการดึงข้อมูลของอุปกรณ์ที่สามารถค้นหามาได้ ที่อาศัยหลักการทำงานของโปรโตคอลซิมเปิลเนตเวิร์กเมเนจเมนท์เป็นหลัก นำมาพัฒนาระบบโดยใช้เทคโนโลยีจาวาให้มีการทำงานผ่านเว็บแอปพลิเคชั่น เพื่อได้แผนภาพของเครือข่ายจากการใช้ระบบและข้อมูลพื้นฐานของอุปกรณ์ที่ทำงานอยู่ได้อย่างถูกต้อง เพื่อให้ผู้ดูแลไม่ต้องเสียเวลาและสามารถใช้งานกับระบบได้ง่าย

Thesis Title	WEB-BASED NETWORK DISCOVERY SYSTEM
Student	Mr. Nuttawit Wongsittiroj
	Ms. Sureenart Kietsaroch
Student ID.	48070057
	48070083
Degree	Bachelor of Science
Program	Information Science
Academic Year	2008
Advisor	Mr. Lapas Pradittasnee

ABSTRACT

Currently, Every enterprise are using data communication network and absolute to using internal enterprise but have communication channel on internet. Also, data communication network is important part of enterprise make to do with network management tool help quickly network management and approve efficiency of network.

Nowadays, Have many network management program which support administrator. Generally, Network management program have to input into program make problem with administrator who unknown network's data. After that, We are learning and do experiment about discovery function which using vary protocols. Result after experiment and learning is Auto discovery function and getting data from database (Management Information Base:MIB) in device using simple network management protocol (SNMP) in system .And then , system by developed with JAVA language which is web-based system. Benefit's system is get network's topology include basic data's device make save time and easy to use for administrator.

กิตติกรรมประกาศ

การทำวิทยานิพนธ์นี้จะสำเร็จลุล่วงไม่ได้ถ้าปราศจากการสนับสนุนจากบุคคลเหล่านี้ จึงได้ใคร่ขอกล่าวคำแสดงความขอบคุณมา ณ โอกาสนี้

ขอขอบพระคุณครอบครัวและบุพการีผู้ให้สติปัญญา ความคิดอ่านและคอยส่งเสริมทั้งทางด้านทุนทรัพย์และกำลังใจในการศึกษา

ขอขอบพระคุณท่านอาจารย์ยลภัส ประดิษฐ์ทัศนีย์ อาจารย์ที่ปรึกษาซึ่งคอยให้คำชี้แนะและแนวทางในการทำวิทยานิพนธ์ในครั้งนี้ผ่านพ้นไปได้ด้วยดี และขอขอบคุณอาจารย์ โชติพัชร์ ภรณวลัย ที่เอื้อเฟื้อสถานที่ในการทำโครงงานครั้งนี้

ขอขอบคุณคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้การสนับสนุนในเรื่องสถานที่ในการเรียนรู้และการทำงาน ซึ่งทำให้เกิดความผูกพันอย่างมาก

ขอขอบคุณพี่ๆที่อยู่ในห้อง I2R (537) ที่คอยช่วยเหลือ ทั้งในเรื่องคำแนะนำในการจัดทำโครงงาน อาหารการกิน และความสะดวกสบายในห้องโปรเจค ขอขอบคุณ พี่เอก ที่เอื้อเฟื้ออุปกรณ์ในการทำการทดลองของโครงงานนี้ให้สำเร็จลุล่วงไปได้ด้วยดี ขอขอบคุณพี่หมู ที่ช่วยเหลือด้านอาหารการกินและจัดการเรื่องสถานที่ การดูแลเอาใจใส่และกำลังใจจากพวกพี่ๆที่มีเสมอมา

ขอขอบคุณผู้เขียนหนังสือและเว็บไซต์ต่างๆ ที่เป็นแหล่งข้อมูลที่ดีให้ค้นคว้าหาความรู้ เพื่อความใช้ในการทำโครงงาน

ท้ายสุดนี้ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกคนที่คอยให้คำปรึกษาคอยเป็นกำลังใจอยู่เสมอ คุณค่าใดที่จะเกิดขึ้นจากวิทยานิพนธ์ฉบับนี้ขอมอบแด่ผู้มีพระคุณทุกท่าน

ณัฐวิชช์ ว่องสิทธิโรจน์
สุรีย์นารถ เกียรติสาโรจน์

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VIII

สารบัญรูป IX

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบเขตของงานวิจัย 2

1.4 วิธีการดำเนินงาน 2

บทที่ 2 หลักการจัดการระบบเครือข่าย 3

2.1 หลักการจัดการระบบเครือข่าย (Network Management Principle) 3

2.2 สถาปัตยกรรมของการจัดการระบบเครือข่าย (Network Management Architecture) 3

2.3 รูปแบบการจัดการเครือข่ายตามมาตรฐาน ISO (ISO Network Management Model) 4

2.3.1 Performance Management 4

2.3.2 Configuration Management 5

2.3.3 Accounting Management 6

2.3.4 Fault Management 6

2.3.5 Security Management 8

2.4 ทีซีพี/ไอพีโปรโตคอล 9

2.4.1 อินเตอร์เนตโปรโตคอล (Internet Protocol:IP) 12

2.4.2 ทรานมิสชั่นคอนโทรลโปรโตคอล(Transmitsion Control Protocol :TCP) 14

2.4.3 ยูสเซอร์ไดอะแกรมโปรโตคอล(User Datagram Protocol :UDP) 15

สารบัญ (ต่อ)

หน้า

บทที่ 3 ซิมเพิลเน็ตเวิร์กเมเนจเมนต์โปรโตคอล (Simple Network Management Protocol:SNMP) 17

3.1 ซิมเพิลเน็ตเวิร์กเมเนจเมนต์โปรโตคอล(Simple Network Management Protocol) 17

3.1.1 โครงสร้างของ SNMP 18

3.1.2 SNMP Version 1 (SNMPv1) 21

3.1.2.1 รูปแบบSNMPv1 PDU 22

3.1.2.2 รูปแบบของ SNMPv1 Trap-PDU 23

3.1.3 SNMP Version 2 (SNMPv2) 24

3.1.3.1 รูปแบบเมสเสจของ SNMP Version 2 (SNMPv2p) 27

3.1.3.2 รูปแบบเมสเสจของ SNMP Version 2 ที่มีรูปแบบทำงานที่ใช้คอมมิวนิตี้ เป็นเกณฑ์ Community-Based SNMP Version 2 (SNMPv2c) 28

3.1.3.3 รูปแบบเมสเสจของ SNMP Version 2 ที่มีรูปแบบทำงานที่ใช้ผู้ใช้เป็นเกณฑ์ User-Based SNMP Version 2 (SNMPv2u) 29

3.1.4 SNMP Version 3 (SNMPv3) 31

3.2 ฐานข้อมูลการจัดการ (Management Information Bases หรือ MIBs) 33

3.2.1 โครงสร้างของ MIBs 33

3.2.1.1 ชนิดของข้อมูลของเมเนจออปเจ็กต์ 35

3.2.1.2 การเข้าถึงข้อมูล 36

3.2.1.3 สถานะของตัวแปรข้อมูล 36

บทที่ 4 การออกแบบระบบ 37

4.1 การวิเคราะห์ 37

4.2 ภาพรวมของการทำงานของระบบ (WEB-BASED NETWORK DISCOVERY SYSTEM) 48

4.2.1 การทำงานของระบบ 48

4.2.2 ฐานข้อมูลที่ใช้ในระบบ 49

สารบัญ (ต่อ)

หน้า

บทที่ 5 การพัฒนาระบบ 53
5.1 ฟังก์ชั่นการทำงานของระบบ SNMP Network Discovery 53
5.1.1 ฟังก์ชั่นค้นหา IP Address ของ Default gateway 53
5.1.2 ฟังก์ชั่นค้นหา subnet address 54
5.1.3 ฟังก์ชั่นค้นหา IP Address ทั้งหมดที่เป็นไปได้ จาก subnet ทั้งหมด 55
5.1.4 ฟังก์ชั่นค้นหาหมายเลข IP Address ที่มีการนำไปใช้งานจริง 56
5.1.5 ฟังก์ชั่นจับกลุ่มหมายเลข IP Address กับอุปกรณ์ router 57
5.1.6 ฟังก์ชั่นค้นหา Next hop ของ router 58
5.1.7 ฟังก์ชั่นวาดรูปแผนภาพเครือข่าย 58

บทที่ 6 การทดลอง 60
6.1 การทดลองที่ 1 60
6.1.1 วิธีการทดลอง 60
6.1.2 ผลการทดลองที่ 1 60
6.2 การทดลองที่ 2 61
6.2.1 วิธีการทดลอง 61
6.2.2 ผลการทดลองที่ 2 61

บทที่ 7 สรุปการทำงานของระบบ 63
7.1 ผลจากการดำเนินการ 63
7.2 ประโยชน์ที่ได้จากระบบ 63
7.3 ข้อจำกัดของระบบ 64
7.4 แนวทางในการดาเนินงานในอนาคต 64

บรรณานุกรม 65
ภาคผนวก 66
คู่มือการติดตั้งระบบ 66
คู่มือการใช้งานระบบ 69
การพัฒนาระบบ 72

สารบัญตาราง

ตารางที่ หน้า

2.1 ประเภทของ Flag ในเฮดเดอร์ของ TCP 15

3.1 รูปแบบเมสเสจทั้งหมดของ SNMPv1 21

3.2 รูปแบบทั่วไปของ SNMPv1 PDUs 22

3.3 แสดงรูปแบบพิเศษเกี่ยวกับ SNMPv1 Trap-PDU 23

3.4 รูปแบบทั่วไปของ SNMPv2 PDU 26

3.5 แสดงค่าในฟิลด์ของ Error Status ในSNMPv2 PDU 26

3.6 รูปแบบทั่วไปของ SNMP Version 2 (SNMPv2p) 27

3.7 รูปแบบทั่วไปของCommunity-Based SNMP Version 2 (SNMPv2c) 28

3.8 รูปแบบทั่วไปของUser-Based SNMP Version 2 (SNMPv2u) 29

3.9 รูปแบบทั่วไปของ SNMP Version 3 (SNMPv3) 33

4.1 ผลการทดลองเปรียบเทียบทราฟฟิกของการค้นหาหมายเลข IP Address โดยใช้ระบบ SNMP Network Discovery ที่มีวิธีการค้นหาที่แตกต่างกันในเครือข่าย 43

4.2 ผลการทดลองเปรียบเทียบทราฟฟิกของการค้นหาหมายเลข IP Address โดยใช้ระบบ SNMP Network Discovery แบบ AUTO กับแบบมีการใส่ช่วง IP Address 46

4.3 ตารางอุปกรณ์ (Table Device) 50

4.4 ตารางอินเตอร์เฟซ (Table Interface) 51

4.5 ตารางการค้นหา (Table Discovery) 52

สารบัญรูป

รูปที่ หน้า

2.1 บรรยายตัวอย่างของสถาปัตยกรรมการจัดการเครือข่าย 5
2.2 ขั้นตอนการทำ Security Management 9
2.3 ขั้นตอนการ Encapsulation และ Demultiplexing 10
2.4 การเปรียบเทียบระหว่าง OSI กับ TCP/IP 11
2.5 ส่วนเฮดเดอร์ของไอพี (IP header) 12
2.6 แพ็กเกต TCP เฮดเดอร์ 14
2.7 แพ็กเกต UDP เฮดเดอร์ 16
3.1 การทำงานของ SNMP 18
3.2 โครงสร้างของ SNMP message 20
3.3 รูปแบบเมสเสจทั้งหมดของ SNMPv1 22
3.4 รูปแบบทั่วไปของ SNMPv1 PDUs 23
3.5 แสดงรูปแบบพิเศษเกี่ยวกับ SNMPv1 Trap-PDU 24
3.6 รูปแบบทั่วไปของ SNMPv2 PDU 27
3.7 รูปแบบทั่วไปของ SNMP Version 2 (SNMPv2p) 28
3.8 รูปแบบทั่วไปของCommunity-Based SNMP Version 2 (SNMPv2c) 29
3.9 รูปแบบทั่วไปของUser-Based SNMP Version 2 (SNMPv2u) 30
3.10 รูปแบบทั่วไปของ SNMP Version 3 (SNMPv3) 32
3.11 โครงสร้างของฐานข้อมูลการจัดการ 34
4.1 แสดงแผนภาพเครือข่ายที่มี router 3 เครื่อง 40
4.2 แสดงแผนภาพเครือข่ายที่มี router 5 เครื่อง 40
4.3 แสดงแผนภาพเครือข่ายที่มี router 8เครื่อง 40
4.4 การค้นหาหมายเลข IP Address ที่มีการใช้งานโดยใช้โปรโตคอล ICMP 41
4.5 การค้นหาหมายเลข IP Address ที่มีการใช้งานโดยใช้โปรโตคอล SNMP 42
4.6 การค้นหาหมายเลข IP Address แบบอัตโนมัติ 45
4.7 กระบวนการวาดรูป Topology 47
4.8 กระบวนการทำงานของระบบทั้งหมด 48

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.9 Entity Relation Diagram ของฐานข้อมูล 49

6.1 แผนภาพเครือข่ายการทดลองที่ 1 60

6.2 ผลการทดลองที่1 60

6.3 แผนภาพเครือข่ายการทดลองที่ 2 61

6.4 ผลการทดลองที่ 2 61

รูปภาคผนวก-1 แสดงการติดตั้งระบบขั้นตอนที่1 66

รูปภาคผนวก-2 แสดงการติดตั้งระบบขั้นตอนที่2 67

รูปภาคผนวก-3 แสดงการติดตั้งระบบขั้นตอนที่3 68

รูปภาคผนวก-4 แสดงหน้าจอเริ่มต้นของระบบ 69

รูปภาคผนวก-5 แสดงหน้าจอผลลัพธ์ของระบบครั้งที่1 70

รูปภาคผนวก-6 แสดงหน้าจอผลลัพธ์ของระบบครั้งที่2 70

รูปภาคผนวก-7 แสดงสถานที่เก็บรูป Topology ที่ได้จากระบบ 71

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันมีการใช้เครือข่ายการสื่อสารข้อมูลภายในองค์กรต่างๆอย่างแพร่หลาย ทั้งองค์กรของรัฐและองค์กรของเอกชน ทั้งในด้านธุรกิจ ด้านการศึกษาและด้านการเมืองการปกครอง การใช้งานเครือข่ายการสื่อสารข้อมูลไม่ได้จำกัดอยู่แค่ภายในองค์กรเท่านั้น แต่ยังจะมีการสื่อสารข้อมูลระหว่างองค์กรด้วยโดยผ่านตัวกลางแบบต่างๆ เช่น อินเตอร์เน็ต (Internet) ดังนั้นเครือข่ายการสื่อสารข้อมูลจึงกลายเป็นส่วนประกอบสำคัญขององค์กรที่ขาดไม่ได้ในปัจจุบัน

การใช้เครือข่ายการสื่อสารข้อมูลจึงจำเป็นที่จะต้องมีประสิทธิภาพในการทำงานที่สูงเพื่อเป็นการสนับสนุนให้ประสิทธิภาพในการทำงานขององค์กรสูงขึ้นด้วย การใช้งานเครือข่ายการสื่อสารข้อมูลให้มีประสิทธิภาพ องค์กรจำเป็นที่จะต้องมีเครื่องมือที่ช่วยสนับสนุนการจัดการเครือข่ายซึ่งจะทำให้การบริหารงานและจัดการเครือข่ายสะดวกรวดเร็วและมีประสิทธิภาพยิ่งขึ้น

ระบบที่ใช้ในการสนับสนุนการบริหารงานและจัดการเครือข่าย (Network Management Software) จัดเป็นเครื่องมือที่มีความจำเป็นอย่างมากในการบริหารงานและจัดการเครือข่ายให้มีประสิทธิภาพ ในเครือข่ายการสื่อสารข้อมูลประกอบไปด้วยส่วนประกอบต่างๆมากมาย ทั้งทางด้านฮาร์ดแวร์ (Hardware) ด้านระบบ (Software) และตัวกลางการสื่อสารข้อมูลต่างๆ (Media) การบริหารงานและจัดการเครือข่ายให้มีประสิทธิภาพจึงต้องพิจารณาในรายละเอียดของส่วนประกอบต่างๆเหล่านี้ รวมทั้งจะต้องพิจารณาถึงการทำงานร่วมกันของอุปกรณ์ต่างๆเหล่านี้ด้วย

เทคโนโลยีที่ใช้ในการพัฒนาระบบในการบริหารงานและจัดการเครือข่ายมีด้วยกันหลายเทคโนโลยี เช่น Simple Network Management Protocol (SNMP), Common Management Information Services /Common Management Information Protocol (CMIS/CMIP), Remote Monitor (RMON) เป็นต้น

ซึ่งแต่ละเทคโนโลยีก็มีความเหมาะสมกับรูปแบบและสถานการณ์ที่แตกต่างกันไปแต่ในโครงงานนี้จะใช้เทคโนโลยี Simple Network Management Protocol (SNMP) ในการพัฒนาระบบ

ระบบจัดการเครือข่ายที่กล่าวนี้จะอาศัยหลักการทำงานของโปรโตคอล SNMP (Simple Network Management Protocol) เป็นหลัก โดยนำหลักการที่ศึกษาการทำงานของโปรโตคอลดังกล่าวมาประยุกต์ใช้ในการค้นหาอุปกรณ์บนเครือข่ายโดยอาศัยหมายเลข IP Address เป็นเกณฑ์ เพื่อให้หาผลลัพธ์แบบอัตโนมัติจากการสั่งงานจากเครื่องเมเนเจอร์ที่อยู่ห่างไกลออกมาในรูปแบบโทโพโลยีของเครือข่าย ซึ่งทำให้เห็นภาพรวมและตำแหน่งอุปกรณ์นั้นๆ บนเครือข่ายได้ รวมทั้งข้อมูลพื้นฐานของอุปกรณ์ เพื่อผู้ดูแลไม่ต้องเสียเวลาเนื่องจากมีการจัดการแบบศูนย์กลาง และมีการทำงานผ่านอินเตอร์เน็ตซึ่งเรียกการทำงานแบบนี้ว่า "web-based"

ทั้งนี้ระบบจัดการเครือข่ายนั้นเป็นตัวช่วยในการทำงานของผู้ดูแลระบบ ทำให้สะดวกและมีความเข้าใจกับภาพรวมของโครงสร้างของระบบเพื่อนำไปศึกษาหรือวิเคราะห์ต่อภายหลังได้

1.2 วัตถุประสงค์

1.2.1 ศึกษาหลักการในการบริหารงานและจัดการเครือข่ายการสื่อสารข้อมูล

1.2.2 ศึกษา Simple Network Management Protocol (SNMP) ซึ่งเป็นโปรโตคอลที่ใช้ในการบริหารงานและจัดการเครือข่ายการสื่อสารข้อมูล และศึกษา Management Information Base (MIB) ซึ่งเป็นส่วนของข้อมูลรายละเอียดต่างๆของอุปกรณ์ที่ถูกบริหาร

1.2.3 วิเคราะห์และออกแบบระบบที่จะทำการพัฒนาโดยใช้ Object-Oriented Technology (JAVA) และมีการทำงานแบบ web-based

1.2.4 สามารถทำให้มองภาพรวมของเครือข่ายโดยได้ผลลัพธ์ออกมาเป็นโครงสร้างเชิงตรรกะของระบบทั้งหมดที่ต้องการในรูปแบบของโทโพโลยี(Topology)และทราบข้อมูลเกี่ยวกับอุปกรณ์บนเครือข่ายที่ได้มาจากการค้นหาอุปกรณ์ที่ใช้โปรโตคอล SNMP

1.2.5 ระบบสามารถทำงานที่อาศัยการค้นหาแบบอัตโนมัติโดยไม่ต้องให้ข้อมูลใดๆต่อระบบทำให้เป็นประโยชน์ต่อผู้ดูแลระบบ

1.3 ขอบเขตของงานวิจัย

1.3.1 ระบบสามารถค้นหาอุปกรณ์เครือข่ายโดยไม่ต้องให้ค่าเริ่มต้นใดๆทั้งสิ้น

1.3.2 ระบบสามารถสร้างการเชื่อมต่อไปยังอุปกรณ์เครือข่ายได้โดยทำการวาดเป็นโทโพโลยีไดอะแกรมรูปภาพอุปกรณ์เครือข่ายที่เชื่อมต่อในเครือข่ายได้ถูกต้อง

1.4 วิธีการดำเนินงาน

1.4.1 ศึกษาทฤษฎีต่างๆที่เกี่ยวข้องกับโครงงานนี้ ซึ่งมีหัวข้อหลักๆอยู่ดังนี้

- ทฤษฎีและหลักการในการจัดการเครือข่าย
- Transmitsion Control Protocol/Internet Protocol:TCP/IP
- Simple Network Management Protocol (SNMP)
- หลักการค้นหาอุปกรณ์บนเครือข่าย(Discovery)
- หลักการวาดโทโพโลยีของเครือข่ายที่ Discovery

1.4.2 ศึกษาหลักการและเมธอดต่างๆใน Library SNMP4j ที่ใช้ภาษา JAVA ในการพัฒนาระบบแบบ web-base อย่างไร

1.4.3 ทำการวิเคราะห์และออกแบบขั้นตอนการทำงานในระบบที่กำลังพัฒนา

1.4.4 ทำการทดสอบและการวิเคราะห์ผลการทำงานของระบบ

1.4.5 สรุปผลการทดสอบระบบและแก้ไขข้อบกพร่องที่เกิดขึ้น

บทที่ 2
หลักการจัดการระบบเครือข่าย

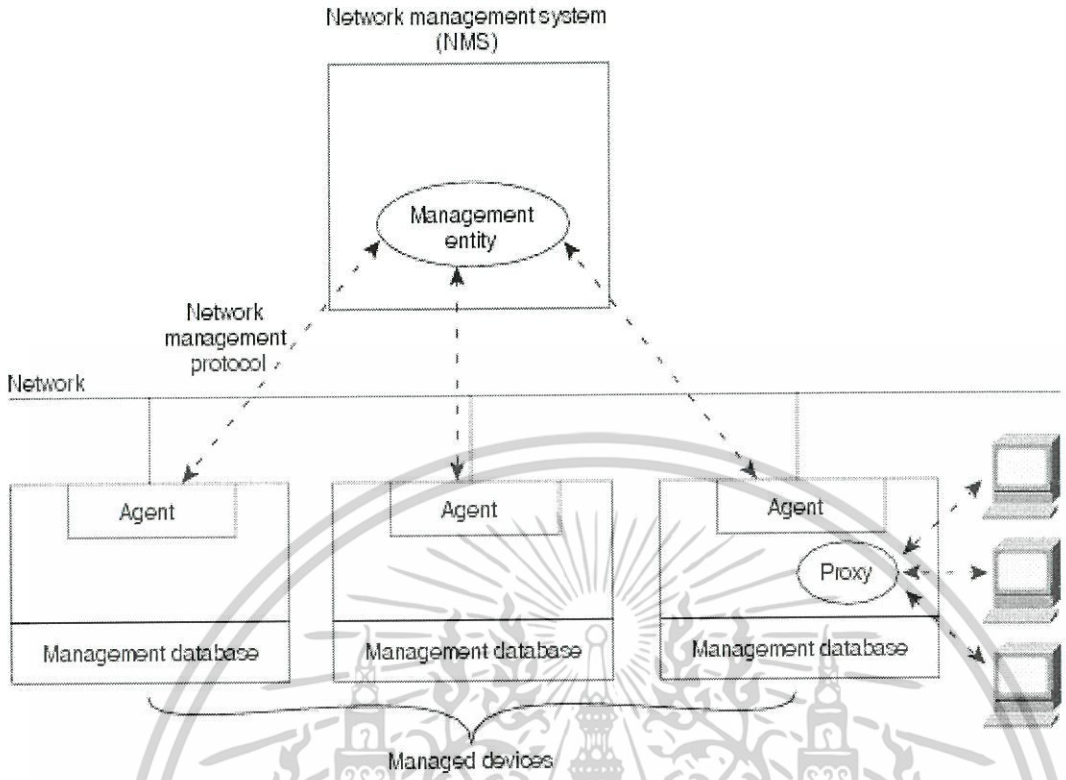
2.1 หลักการจัดการระบบเครือข่าย (Network Management Principle)

เป็นการที่ให้ความช่วยเหลือแก่ผู้ดูแลระบบเครือข่ายโดยสังเกตและควบคุมพฤติกรรมของเครือข่ายด้วยโปรโตคอลที่ใช้ในการวิเคราะห์ การจัดการระบบเครือข่ายรวมทั้งการกระจายฐานข้อมูล มีการสำรวจอุปกรณ์เครือข่ายแบบอัตโนมัติและมีความสามารถสูงในการสร้างมุมมองกราฟฟิกแบบเรียลไทม์ของโทโพโลยีของเครือข่ายที่เปลี่ยนแปลงและทราฟฟิกของการส่งข้อมูล โดยทั่วไปการจัดการระบบเครือข่ายคือการบริการที่มีเครื่องมือที่เข้ามาช่วยอย่างหลากหลาย,แอฟพลิเคชั่น และอุปกรณ์เพื่อช่วยผู้จัดการเครือข่ายในการสังเกต,ควบคุมและการรักษาเครือข่าย ให้เกิดการทำงานที่มีประสิทธิภาพและได้ผลลัพธ์ในการแก้ไขปัญหาต่างๆ รวมถึงมีการจัดเก็บสถานการณ์ต่างๆที่เกิดขึ้นในเครือข่าย

2.2 สถาปัตยกรรมของการจัดการระบบเครือข่าย (Network Management Architecture)

ส่วนใหญ่สถาปัตยกรรมของการจัดการระบบเครือข่ายใช้โครงสร้างพื้นฐานที่เหมือนๆกันและกลุ่มความสัมพันธ์ของอุปกรณ์ที่ต้องการจัดการ เช่นระบบคอมพิวเตอร์ และอุปกรณ์เครือข่ายประเภทอื่นๆ การที่ใช้ระบบที่สามารถมีการแจ้งเตือนเมื่อเกิดปัญหาที่รู้จักหรือเกิดขึ้นโดยทั่วไป ในเวลาที่ได้รับการแจ้งเตือน การจัดการที่มีอยู่ทำการโปรแกรมเพื่อที่จะโต้ตอบโดยการดำเนินการ ซึ่งประกอบด้วยการทำงานที่มีการเตือนล่วงหน้า,บันทึกเหตุการณ์ที่เกิดขึ้น,ปิดระบบ และ พยายามที่จะซ่อมแซมระบบโดยอัตโนมัติ

การจัดการที่มีอยู่นั้นสามารถสำรวจอุปกรณ์เพื่อตรวจสอบค่าเธรสโฮล์ด (Threshold) การสำรวจสามารถทำได้แบบอัตโนมัติหรือผู้ใช้เป็นผู้เริ่มการทำงาน แต่เอเจนต์ในอุปกรณ์ทั้งหมดที่จัดการจะตอบสนองการสำรวจ เอเจนต์คือส่วนระบบที่มีขั้นตอนการทำงานโดยในอันดับแรกมีการรวบรวมข้อมูลที่เกี่ยวกับอุปกรณ์ที่มีอยู่ แล้วเก็บข้อมูลเหล่านี้ในฐานข้อมูลการจัดการและสุดท้ายมีการจัดหาให้มีการจัดการภายในระบบการจัดการเครือข่าย network management systems (NMSs) โดยผ่านโปรโตคอลการจัดการเครือข่ายโปรโตคอลที่รู้จักกันดีประกอบด้วย Simple Network Management Protocol (SNMP) ตัวแทนที่ได้รับหน้าที่ในการจัดการนั้นจัดหาข้อมูลบนตัวแทนที่นอกเหนือจากนี้



รูปที่ 2.1 บรรยายตัวอย่างของสถาปัตยกรรมการจัดการเครือข่าย

2.3 รูปแบบการจัดการเครือข่ายตามมาตรฐาน ISO (ISO Network Management Model)

International Organization for Standardization (IOS) มีการพัฒนาโครงสร้างสำหรับการจัดการของเครือข่ายในพื้นฐานของ Structure of Management Information (SMI) โดยโครงสร้างมีการแบ่งแยกกระบวนการการจัดการเครือข่ายออกเป็น 5 ฟังก์ชั่นหลักๆ แต่ละฟังก์ชั่นจะมีความสัมพันธ์กับกระบวนการการจัดการ IT ระดับสูง คือ

2.3.1 Performance Management

เป็นตัวชี้วัดและหลักเกณฑ์ของประสิทธิภาพของเครือข่ายที่มีอยู่อย่างหลากหลาย ดังนั้นประสิทธิภาพระหว่างเครือข่ายสามารถถูกรักษาในระดับที่ยอมรับได้ ตัวอย่างของตัวแปรประสิทธิภาพ เช่น ค่าเปอร์เซ็นต์การใช้งานบนเครือข่าย (Utilization), จำนวนข้อมูลเข้าและออกในช่วงเวลาหนึ่งๆ ของเครือข่าย (Byte), แพ็คเก็ต(Packet) เป็นต้น

การจัดการประสิทธิภาพรวมถึงขั้นตอนหลัก 3 ขั้นตอน

1. ข้อมูลเชิงประสิทธิภาพถูกรวบรวมบนตัวแปรที่ผู้ดูแลเครือข่ายสนใจ
2. ข้อมูลถูกวิเคราะห์เพื่อกำหนดระดับปกติ เป็นเกณฑ์มาตรฐาน
3. ค่าเธรสโฮล์ ที่เหมาะสมที่เกี่ยวข้องกับเครือข่ายถูกกำหนดสำหรับแต่ละตัวแปรที่สำคัญ ดังนั้นเมื่อค่าตัวแปรมากกว่า ค่าเธรสโฮล์ จะชี้บอกถึงปัญหาที่เกิดขึ้นกับเครือข่าย

การจัดการที่มีอยู่สามารถสังเกตเฝ้าระวังตัวแปรของประสิทธิภาพได้อย่างต่อเนื่อง เมื่อค่าเชิงประสิทธิภาพมากกว่าค่าเธรสโฮล์ จะมีการแจ้งเตือนและส่งไปให้ระบบการจัดการเครือข่าย

แต่ละขั้นตอนได้อธิบายในส่วนกระบวนการตั้งค่าของระบบโต้ตอบ เมื่อประสิทธิภาพเริ่มไม่เป็นที่ยอมรับเพราะเกินกว่าค่าเธรสโฮล์ที่ผู้ใช้กำหนดไว้ ระบบจะโต้ตอบโดยการส่งข้อความแล้วการจัดการประสิทธิภาพจึงยอมรับให้มีการเริ่มการทำงาน ตัวอย่างการจำลองเครือข่ายที่วางแผนจะทำให้เครือข่ายที่ขยายเติบโตนั้นส่งผลต่อระบบเมตริกที่เกี่ยวข้องกับประสิทธิภาพของเครือข่ายอย่างไร ดังนั้นการจำลองสามารถแจ้งเตือนผู้ดูแลระบบให้ทราบปัญหาที่ใกล้จะเกิดขึ้น ทำให้เป็นตัววัดเพื่อให้ขัดขวางพฤติกรรมนั้นล่วงหน้า

2.3.2 Configuration Management

เป็นกระบวนการเก็บข้อมูลจากเครือข่ายและใช้ข้อมูลในการจัดการตั้งค่าอุปกรณ์เครือข่ายทั้งหมด ซึ่งประกอบด้วย

1. การเก็บข้อมูลเกี่ยวกับองค์ประกอบของเครือข่าย(net work configuration)ปัจจุบัน
2. การใช้ข้อมูลทำการแก้ไของค์ประกอบของเครือข่าย(net work configuration) อุปกรณ์
3. การเก็บข้อมูล,การดูแลรักษาข้อมูลที่มีให้นั้นใหม่อยู่เสมอ
4. การสร้างรายงานจากข้อมูลที่ได้

การทำ Configuration management ประกอบด้วยขั้นตอนการทำงานดังนี้

1. การรวบรวมข้อมูลเกี่ยวกับสภาวะแวดล้อมของเครือข่ายล่าสุด ข้อผิดพลาดในการเก็บข้อมูลจะทำให้ผู้ดูแลเสียเวลาในการแก้ไขปัญหาเครือข่าย ที่เกิดจากองค์ประกอบ ที่ผิดพลาดแบบง่ายๆ การเก็บข้อมูลสามารถทำได้แบบ manual โดยผู้ดูแล และแบบอัตโนมัติ โดยระบบ

2. การใช้ข้อมูลเพื่อแก้ไของค์ประกอบของอุปกรณ์เครือข่าย จากการที่สภาวะแวดล้อมของอุปกรณ์เครือข่าย มีการเปลี่ยนแปลงอยู่เสมอ ดังนั้นความสามารถในการแก้ไของค์ประกอบแบบ real time เป็นสิ่งที่จำเป็น การแก้ไขอาจทำได้แบบ manual หรืออัตโนมัติ ขึ้นอยู่กับรูปแบบการเก็บข้อมูลว่าเป็นแบบ manual หรืออัตโนมัติ

3. การเก็บข้อมูล, ดูแลรักษาข้อมูล รายการของอุปกรณ์ให้ใหม่อยู่เสมอ ในทุกๆส่วนของเครือข่าย และการสร้างรายงานแบบ ต่างๆ

ประโยชน์ที่ได้จากการทำ Configuration Management

ผลที่ได้อย่างแรกคือการเพิ่มความสามารถของผู้ดูแลที่จะควบคุมองค์ประกอบของอุปกรณ์เครือข่าย ซึ่งทำได้โดยการเสนอการเข้าถึงข้อมูลองค์ประกอบที่สำคัญของแต่ละอุปกรณ์ ในระบบที่ซับซ้อนขึ้นจะช่วยให้ผู้ดูแลเปรียบเทียบองค์ประกอบที่ใช้งานอยู่ (running configuration) กับองค์ประกอบที่เก็บไว้ในระบบ และทำการเปลี่ยนองค์ประกอบได้ง่ายตามต้องการ

ในบางกรณีที่อุปกรณ์มีการแก้ไข เช่นการที่ต้องแก้ไขอินเตอร์เฟซที่ทำให้เกิดข้อผิดพลาดบนส่วนของ LAN โดยการใช้เครื่องมือ configuration management สามารถทำ remote configuration มายังอุปกรณ์เพื่อยกเลิกการใช้งาน interface นั้น แล้วทำการตรวจสอบ configuration ของอินเตอร์เฟซและสังเกตว่ามีการตั้งค่า configure ที่ผิดพลาดทำให้เกิดข้อผิดพลาดขึ้น การใช้เครื่องมือ configuration management ทำให้สามารถแก้ไขค่า configure ที่ผิดพลาดให้ถูกต้องและทำการ active interface นั้นขึ้นมาใหม่

2.3.3 Accounting Management

เป็นตัววัดประสิทธิภาพที่ได้จากปัจจัยของเครือข่ายทำให้ผู้ใช้หรือกลุ่มผู้ใช้บนเครือข่ายโดยถูกควบคุม ดูแลได้อย่างเหมาะสม ดังนั้นการควบคุมดูแลทำให้เกิดปัญหาน้อยลง และ มีการเข้าถึงเครือข่ายได้อย่างถูกต้องของผู้ใช้ทุกคนให้ได้มากที่สุด เหมือนกับ performance management โดยขั้นตอนแรก accounting management เป็นการหาค่าการใช้ประโยชน์ของทรัพยากรที่สำคัญในเครือข่ายทั้งหมด การวิเคราะห์ของผลลัพธ์ให้เข้าใจในรูปแบบที่ใช้ในปัจจุบัน และส่วนแบ่งที่ใช้ให้เกิดประโยชน์สามารถถูกตั้งค่าได้ ข้อเท็จจริงบางส่วนคือมีความต้องการให้มีการเข้าถึงที่ดีที่สุด จากจุดนี้และไปเรื่อยๆ การประมาณการใช้ทรัพยากรสามารถให้ทำรายการของข้อมูลที่ดีเท่ากับข้อมูลที่ใช้เป็นทรัพย์สินที่ถูกต้องต่อไปและเป็นการใช้ทรัพยากรให้เกิดประโยชน์สูงสุด

2.3.4 Fault Management

เป็นกระบวนการที่ใช้กำหนดตำแหน่งและแก้ไขปัญหาเครือข่าย ที่เรียกว่า Fault (ข้อผิดพลาด) ซึ่งจากการที่ Network management มีการทำงานย่อยรวมอยู่หลายงาน Fault management เป็นงานที่จัดว่ามีความสำคัญสูงสุด ซึ่งประกอบด้วย

1. การระบุการเกิดของข้อผิดพลาดบน data network
2. การหาสาเหตุของข้อผิดพลาด
3. การแก้ไขข้อผิดพลาด(ถ้าทำได้)

การเก็บรวบรวมข้อมูลที่ใช้ในการระบุปัญหา ต้องทำการรวบรวมข้อมูล ที่เกี่ยวข้องกับสถานะของเครือข่าย ซึ่งจะมีวิธี 2วิธี

1. ข้อมูลที่เกี่ยวข้องกับ สถานการณ์วิกฤติของเครือข่าย (Critical network event) ที่ถูกส่งมาให้โดยอุปกรณ์เครือข่าย ในขณะที่เกิด ข้อผิดพลาดขึ้น เช่น Link fail, การที่อุปกรณ์ restart หรือการที่โฮสต์ทำการตอบสนองช้า โดยส่วนใหญ่การเชื่อถือข้อมูลเพียงบางเหตุการณ์จะไม่เพียงพอที่นำมาใช้สำหรับการทำFault management ที่มีประสิทธิภาพ ตัวอย่าง เช่น ถ้าอุปกรณ์เครือข่ายไม่สามารถทำงานต่อได้อย่างสมบรูณ์ ก็ไม่สามารถส่ง event ต่างๆได้ดังนั้น Fault management tool ที่ใช้เพียงบาง สถานการณ์วิกฤติของเครือข่าย (Critical network event) ก็อาจจะไม่มีการอัพเดตกับสถานะของอุปกรณ์เครือข่าย

2. การ polling ไปยังอุปกรณ์เครือข่ายเป็นช่วงๆ จะช่วยทำให้พบปัญหาที่เกิดขึ้นได้ขึ้นอยู่กับช่วงเวลาที่ใช้ polling อย่างไรก็ตามต้องยอมรับผลของการใช้วิธีนี้ ความเร็วของการตรวจพบขึ้นอยู่กับความถี่ของการ polling ซึ่งขึ้นกับการเปรียบเทียบความถี่ของการ polling แล้วทำให้พบปัญหาได้เร็วกับแบนด์วิดท์ (bandwidth) ที่ถูกใช้ไป ดังนั้นถ้าต้องการให้พบปัญหาได้เร็วที่สุดต้องใช้แบนด์วิดท์ที่มาก ซึ่งปัจจัยอื่นที่ใช้พิจารณาเมื่อทำการตัดสินในการกำหนดค่าช่วงเวลา polling time คือจำนวนของอุปกรณ์ที่ทำการ poll และแบนด์วิดท์ของ link นั้นๆ

การกำหนดว่าจะใช้ข้อผิดพลาดค่าใดเพื่อนำมาจัดการระบบ data network ซึ่งควรถูกกำหนดโดยปัจจัยดังต่อไปนี้

1. ขอบเขตของการดูแลเครือข่ายซึ่งมีผลต่อจำนวนของข้อมูลที่จะเก็บจากอุปกรณ์เครือข่าย
2. ขนาดของเครือข่าย

Fault management tools ต่างๆที่ใช้ตั้งแต่ใช้ง่ายๆจนถึงเครื่องมือที่พิเศษที่ออกแบบมาทำงานFault management โดยเครื่องมือที่ง่ายๆสามารถใช้หาจุดที่เกิดปัญหาได้แต่ไม่สามารถบอกสาเหตุได้ ส่วนเครื่องมือที่มีความซับซ้อนมากกว่าจะใช้ข้อได้เปรียบของโฮสต์ (host) และอุปกรณ์เครือข่ายเพื่อส่งสถานการณ์วิกฤติของเครือข่าย (Critical network event) ซึ่งสามารถใช้หาสาเหตุของปัญหาได้ ส่วนAdvance tool สามารถทำได้เหนือกว่าอีกระดับโดยสามารถแก้ปัญหาที่เกิดขึ้นให้ทันที

<u>รูปแบบของการรายงาน Fault</u>

รูปแบบที่ใช้มีความสำคัญเช่นกัน โดยปกติจะแสดงได้ 3 รูปแบบดังนี้

1. ข้อความ (Text)
2. รูป Graphic
3. เสียง

โดยแบบข้อความเป็นแบบที่ควรเลือกใช้ในเบื้องต้นเพราะสามารถทำงานได้บนจอหรือ terminal ได้ทุกแบบ แต่อย่างไร ก็ตามแบบรูปภาพจะดูแล้วสื่อความหมายได้ดีที่สุด โดยการแสดงผลแบบนี้ต้องใช้หน้าจอสีที่ปกติใช้กับเครื่องมือที่ใช้บน Network management system อยู่แล้ว โดยถ้าไม่มีสีก็สามารถทำให้รูปภาพกระพริบได้ ส่วนแบบที่ใช้เสียงมีจุดเด่น ในการเตือนให้ผู้ดูแลทราบได้เร็ว

ที่สุดในการแสดงผลถ้าสามารถระบุได้ถึงจุดที่มีผลกระทบด้วยก็จะทำให้ สามารถแยกแยะปัญหาได้เร็วขึ้น

2.3.5 Security Management

จุดประสงค์ของ Security Management คือการควบคุมการเข้าถึงทรัพยากรบนเครือข่าย ซึ่งสอดคล้องกับนโยบายที่เฉพาะ ดังนั้นเครือข่ายไม่สามารถถูกก่อกวนหรือทำลายได้ทั้งแบบเจตนาหรือไม่เจตนา และ มีการรับรู้ทันทีเมื่อมีข้อมูลที่ไม่สามารถเข้าถึงโดยที่ไม่ได้รับการอนุญาติที่เหมาะสม ระบบย่อยใน Security Management ตัวอย่าง เช่น สามารถเฝ้าระวังหรือสังเกตจากบันทึกพฤติกรรมของผู้ใช้ที่เข้ามาใช้ทรัพยากรบนเครือข่าย และ สามารถปฏิเสธการเข้าถึงของรหัสของผู้ใช้ในการเข้าถึงที่ไม่เหมาะสม

ระบบย่อยใน Security Management ทำงานโดยใช้หลักการในการแบ่งทรัพยากรบนเครือข่ายให้กับผู้ที่ได้รับอนุญาติและผู้ที่ไม่ได้รับอนุญาติ สำหรับผู้ใช้บางคนเข้าถึงทรัพยากรบางส่วนที่ไม่เหมาะสม โดยทั่วไปจะเป็นเพราะผู้ใช้เป็นคนภายนอกบริษัท สำหรับเครือข่ายอื่นๆภายในผู้ใช้สามารถเข้าถึงข้อมูลที่สร้างจากแผนกงานต่างๆก็เป็นสิ่งที่ไม่เหมาะสม อย่างการเข้าถึงไฟล์ที่เกี่ยวกับทรัพยากรบุคคล

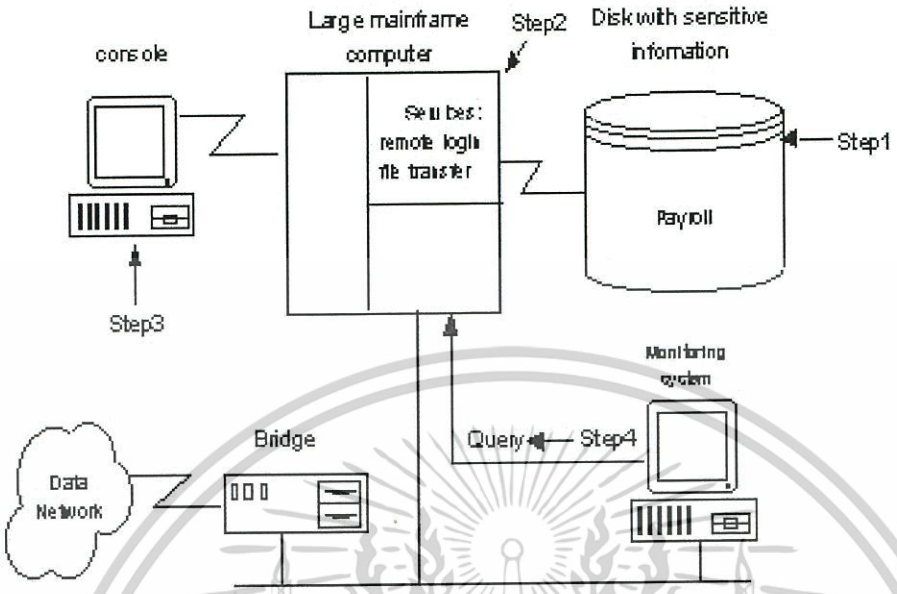
ระบบย่อยใน Security Management แสดงการทำงานหลายวิธี มีการกำหนดทรัพยากรบนเครือข่ายและให้มีการจับกลุ่มระหว่างทรัพยากรกับกลุ่มผู้ใช้ที่อนุญาติ และมีการเฝ้าสังเกตที่จุดที่มีการเข้าถึงทรัพยากรและบันทึกการเข้าถึงทรัพยากรที่ไม่เหมาะสม

<u>ประโยชน์ของการทำ Security Management</u>

สิ่งที่ต้องคำนึงอย่างแรกสำหรับผู้ใช้หลายๆคนเกี่ยวกับการต่อเครื่องแม่ข่ายไปยัง data network คือแนวโน้มของการขาด Security ของข้อมูลที่มีความสำคัญที่อยู่บน host เพื่อหลีกเลี่ยงปัญหานี้การที่เครื่องแม่ข่าย ทำงานกับข้อมูลที่มีความสำคัญสามารถหลีกเลี่ยงการเชื่อมต่อเครือข่ายและส่งผ่านข้อมูลได้โดยใช้ movable media เช่น เทปแม่เหล็ก,Optical disc และวิธีอื่นๆ โดยวิธีนี้ถ้าผู้ใช้ที่มี physical security access ไปยัง host สามารถเข้าถึงข้อมูลที่มีความสำคัญได้ อย่างไรก็ตามแม้ว่าวิธีนี้จะปลอดภัย แต่ก็ไม่สะดวกที่จะใช้งาน

การตั้งค่าที่เหมาะสมและการบำรุงรักษา Security management ที่ดี ทำให้สามารถที่จะเสนอแนวทางปฏิบัติได้หลายอย่าง เพื่อบรรเทาความกังวลเรื่องความปลอดภัย ของผู้ใช้และเพิ่มความเชื่อมั่นในประสิทธิภาพของเครือข่าย และ security การสร้างความเชื่อมั่นและการป้องกันข้อมูลที่สำคัญเป็นผลประโยชน์หลักที่ได้จากการทำ Security management

ผลเสียของการที่ไม่มี Security management ในเครือข่ายสามารถแสดงให้เห็นได้ไม่ยาก โดยสมมติว่า private data network ขององค์กร เชื่อมต่อไปยัง public data network และถ้าคอมพิวเตอร์ภายใน เครือข่ายของบริษัทมีข้อมูลเงินเดือนอยู่ให้บริการข้อมูลกับใครก็ได้ที่มาร้องขอ ซึ่งผลของการที่ไม่จำกัดสิทธิการ เข้าถึงไปยังข้อมูลที่สำคัญอาจทำให้เกิดความเสียหายกับองค์กรได้



รูปที่ 2.2 ขั้นตอนการทำ Security Management

เหตุผลในการบริหารและจัดการเครือข่ายการสื่อสาร

- เพื่อตรวจหาสถานะการทำงานของอุปกรณ์และการสื่อสารภายในเครือข่าย
- เพื่อการบริหารจัดการเครือข่ายขนาดใหญ่ที่ซับซ้อนได้ดียิ่งขึ้น
- เพื่อการตรวจสอบสถานะของ threshold ที่ถูกกำหนดไว้
- เพื่อจัดเตรียมเครือข่ายไว้สำหรับอุปกรณ์ชนิดใหม่ๆ
- เพื่อความสะดวกในการเปลี่ยนแปลงคุณลักษณะต่างๆของระบบเครือข่าย
- เพื่อการใช้งานเครือข่ายการสื่อสารของผู้ใช้ให้มีประสิทธิภาพมากยิ่งขึ้น
- เพื่อทำการปรับประสิทธิภาพและความสามารถในการทำงานของเครือข่ายให้ สมดุลกัน
- เพื่อรักษาต้นทุนในการจัดการเครือข่าย

2.4 ทีซีพี/ไอพีโปรโตคอล (Transmitsion Control Protocol/Internet Protocol:TCP/IP)

เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเตอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

ชุดโปรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้ TCP/IP เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน 3 ประการคือ

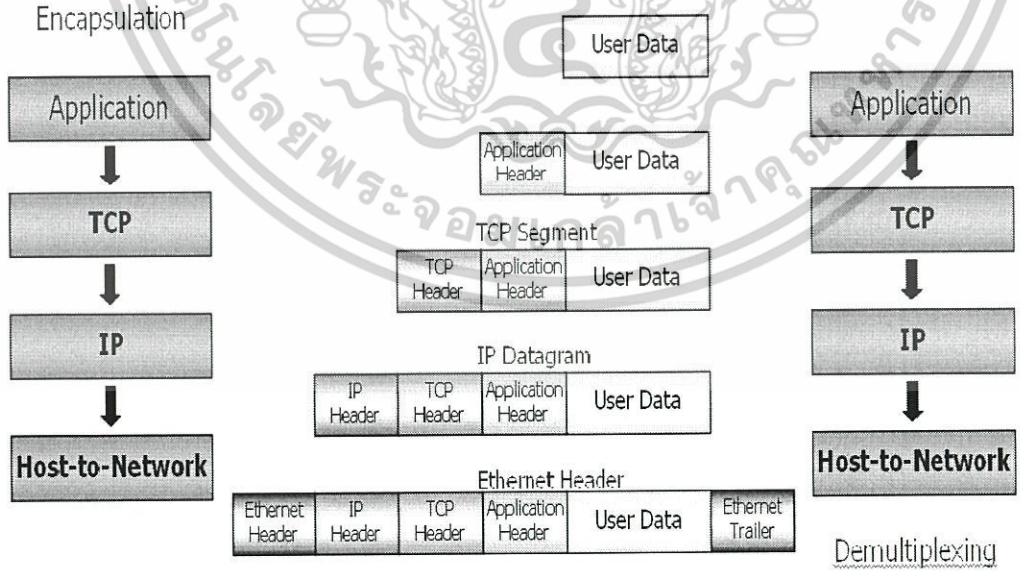
1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน

2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ

3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

Encapsulation/Demultiplexing

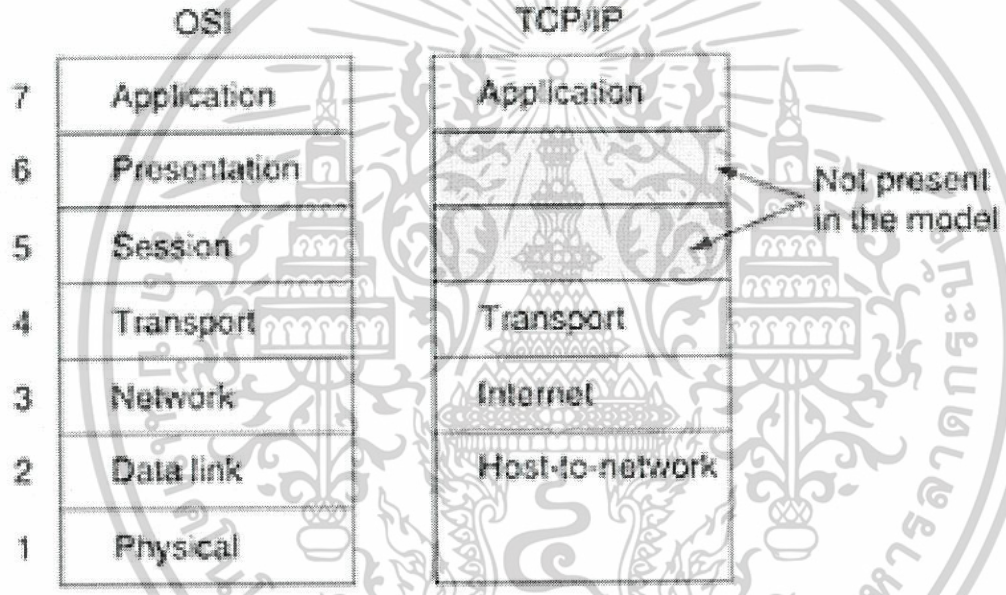
การส่งข้อมูลผ่านในแต่ละเลเยอร์ แต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมา กับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือ โปรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า “Demultiplexing”



รูปที่ 2.3 ขั้นตอนการ Encapsulation และ Demultiplexing

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละเลเยอร์มีชื่อเรียกแตกต่างกัน ดังนี้

- ข้อมูลที่มาจาก User หรือก็คือข้อมูลที่ User เป็นผู้ป้อนให้กับ Application เรียกว่า User Data
- เมื่อแอพพลิเคชั่นได้รับข้อมูลจาก user ก็จะนำมาประกอบกับส่วนหัวของแอพพลิเคชั่น เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP
- เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมารวมกับ Header ของ โปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยังโปรโตคอล IP
- เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมารวมกับ Header ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปยังเลเยอร์ Host-to-Network Layer
- ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป



รูปที่ 2.4 การเปรียบเทียบระหว่าง OSI กับ TCP/IP

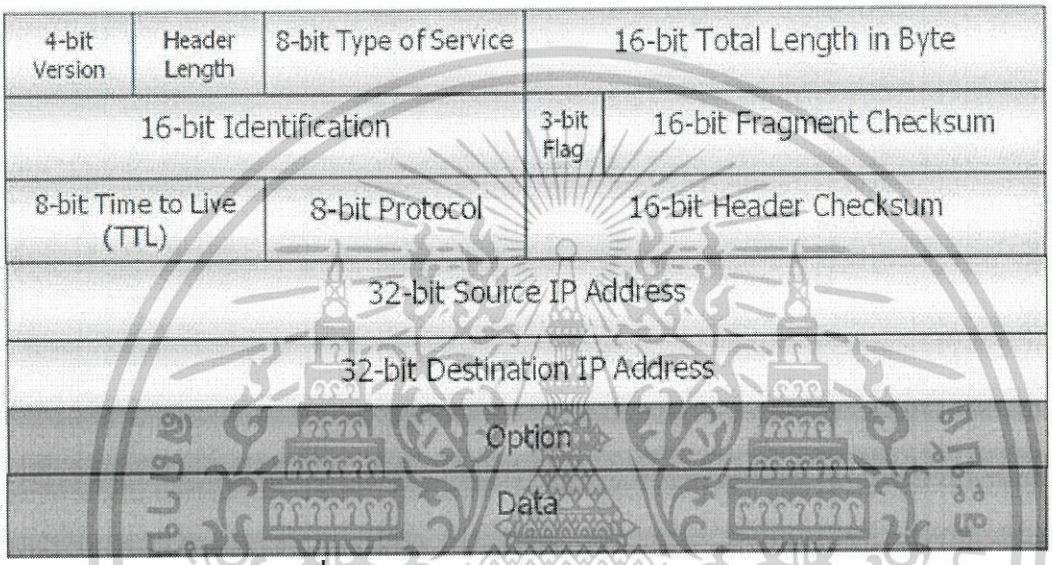
ในแต่ละเลเยอร์ของโครงสร้าง TCP/IP สามารถอธิบายได้ดังนี้

ชั้นที่ 1 : ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer) จะครอบคลุมการทำงานในส่วน 2 ชั้นล่างของ OSI คือชั้น Physical และชั้น Datalink เป็นโปรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูลทางด้านผู้รับก็จะทำงานในทางกลับกัน คือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร

ชั้นที่ 2 : ชั้นสื่อสารอินเตอร์เน็ต (The Internet Layer) ใช้ประเภทของระบบการสื่อสารที่เรียกว่า ระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็คเก็ต (packet-switching network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่า แพ็ค

เก็ต (Packet) สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่างๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ หากว่ามีการส่งแพ็คเก็ตออกมาเป็นชุดโดยมีจุดหมายปลายทางเดียวกันในระหว่างการเดินทางในเครือข่าย แพ็คเก็ตแต่ละตัวในชุดนี้ก็จะเป็นอิสระแก่กันและกัน ดังนั้น แพ็คเก็ตที่ส่งไปถึงปลายทางอาจจะไม่เป็นไปตามลำดับก็ได้

2.4.1 อินเตอร์เนตโปรโตคอล (Internet Protocol:IP)



รูปที่ 2.5 ส่วนเฮดเดอร์ของไอพี (IP header)

เฮดเดอร์ของ IP โดยปกติจะมีขนาด 20 bytes ยกเว้นในกรณีที่มีการเพิ่ม option บางอย่าง ฟิลด์ของเฮดเดอร์ IP จะมีความหมายดังนี้

- Version : หมายเลขเวอร์ชันของโปรโตคอล ที่ใช้งานในปัจจุบันคือ เวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)
- Header Length : ความยาวของเฮดเดอร์ โดยทั่วไปถ้าไม่มีส่วน option จะมีค่าเป็น (5*32 bit)
- Type of Service (TOS) : ใช้เป็นข้อมูลสำหรับเราเตอร์ในการตัดสินใจเลือกการเราต์ข้อมูลในแต่ละดาต้าแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว
- Length : ความยาวทั้งหมดเป็นจำนวนไบต์ของดาต้าแกรม ซึ่งด้วยขนาด 16 บิตของฟิลด์ จะหมายถึงความยาวสูงสุดของดาต้าแกรม คือ 65535 byte (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดในลิงค์เลเยอร์ และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทางแอพพลิเคชั่นส่วนใหญ่จะมีขนาดของดาต้าแกรมไม่เกิน 512 byte
- Identification : เป็นหมายเลขของดาต้าแกรมในกรณีที่มีการแยกดาต้าแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี identification เดียวกันมารวมกัน
- Flag : ใช้ในกรณีที่มีการแยกดาต้าแกรม

- Fragment offset : ใช้ในการกำหนดตำแหน่งข้อมูลในดาต้าแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง

- Time to live (TTL) : กำหนดจำนวนครั้งที่มากที่สุดที่ดาต้าแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิด time out ในระหว่างการส่งข้อมูล

- Protocol : ระบุโปรโตคอลที่ส่งในดาต้าแกรม เช่น TCP ,UDP หรือ ICMP
- Header checksum : ใช้ในการตรวจสอบความถูกต้องของข้อมูลในเฮดเดอร์
- Source IP address : หมายเลข IP ของผู้ส่งข้อมูล
- Destination IP address : หมายเลข IP ของผู้รับข้อมูล
- Data : ข้อมูลจากโปรโตคอลระดับบน

เป็นโปรโตคอลที่อยู่ในชั้น Network ของ OSI ถูกพัฒนาขึ้นเพื่อส่งข้อมูลในรูปกลุ่มของบิตจากต้นทางไปยังปลายทางผ่านระบบการเชื่อมต่อของเครือข่าย ทำหน้าที่รับผิดชอบเกี่ยวกับการค้นหาเส้นทาง (Routing) บนระบบเครือข่าย และหากในกรณีแพ็คเก็ตข้อมูลมีขนาดใหญ่เกินไปก็จะทำการแบ่งแพ็คเก็ตนั้นออกเป็นชิ้นส่วนย่อยๆหรือทำการ Fragment แล้วจึงทำการส่งผ่านไปยังระบบเครือข่ายต่อไป และเมื่อถึงปลายทางก็จะทำการจัดเรียงลำดับที่ถูกต้องใหม่ เนื่องจากแต่ละชิ้นส่วนนั้นสามารถถูกส่งมาโดยคนละเส้นทางกันได้ และในการส่งข้อมูลไปบนเครือข่ายแต่ละประเภทจะมีขนาดของข้อมูลไม่เท่ากัน จึงต้องทำการแบ่งแพ็คเก็ตเป็นส่วนๆเพื่อให้พอดีกับขนาดของข้อมูลของเครือข่ายที่จะทำการส่งข้อมูลผ่านไป โดยในส่งหัว (Header) ของแพ็คเก็ตนั้น จะมีข้อมูลของลำดับของชิ้นของแพ็คเก็ตอยู่ด้วย เพื่อใช้ในการจัดเรียงลำดับข้อมูลในการประกอบชิ้นส่วนกับรูปเดิม

การค้นหาเส้นทางของแพ็คเก็ตจะต้องใช้ข้อมูลจากตารางการค้นหาเส้นทาง (Routing Table) ซึ่งในการเก็บข้อมูลเส้นทางของเครือข่ายจะต้องใช้โปรโตคอลชนิดอื่นๆเข้ามาช่วย เช่น Routing Information Protocol (RIP)

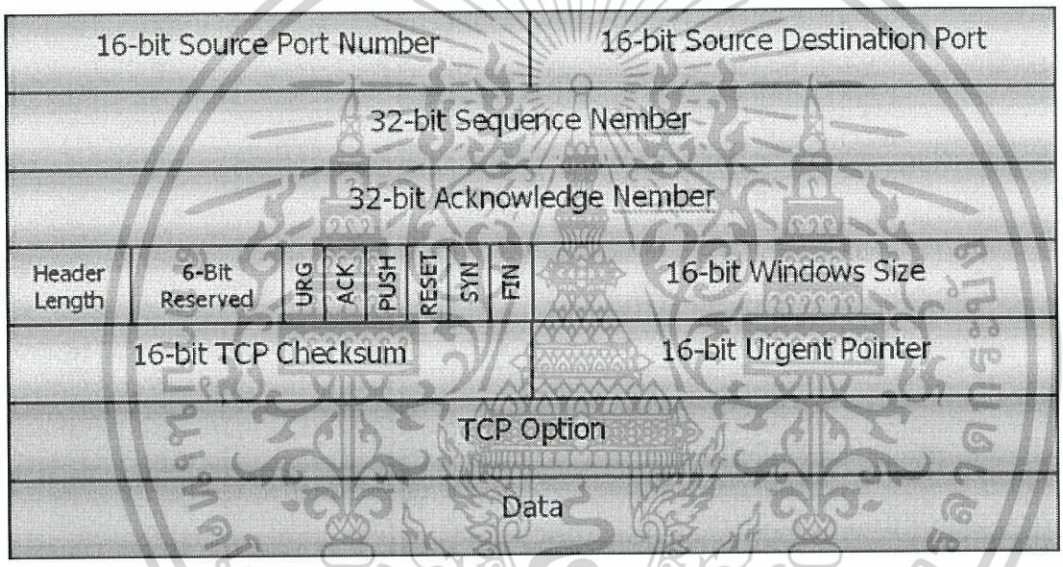
ชั้นที่ 3 : ชั้นสื่อสารนำส่งข้อมูล (Transport Layer) แบ่งเป็นโปรโตคอล 2 ชนิดตามลักษณะ ลักษณะแรกเรียกว่า Transmission Control Protocol (TCP) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte stream ที่ไว้ใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเตอร์เน็ต ทางฝ่ายผู้รับจะนำเมสเสจ (message) มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

โปรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้ง

กลับไปยังผู้ส่ง จึงถือได้ว่าไม่มีการตรวจสอบความถูกต้องของข้อมูล อย่างไรก็ตาม วิธีการนี้มีข้อดีในด้านความรวดเร็วในการส่งข้อมูล จึงนิยมใช้ในระบบผู้ให้และผู้ใช้บริการ (client/server system) ซึ่งมีการสื่อสารแบบ ถาม/ตอบ (request/reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่งเสียง (voice) ทางอินเตอร์เน็ต

2.4.2 ทรานมิสชั่นคอนโทรลโปรโตคอล (Transmitsion Control Protocol :TCP)

เป็นโปรโตคอลที่ทำหน้าที่อยู่ในชั้น Transport โดยมีการทำงานเป็นแบบ Connection Oriented โดยมีหน้าที่รับผิดชอบดูแลความน่าเชื่อถือของการติดต่อสื่อสารระหว่างงาน 2 งานผ่านระบบเครือข่าย โดยกลุ่มของข้อมูลที่ถูกส่งชั้นนี้จะถูกเรียกว่า สตรีม (Stream)



รูปที่ 2.6 แพ็กเกต TCP เฮดเดอร์

มีรายละเอียด ดังนี้

- Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้
- Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม
- Sequence Number : ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง
- Acknowledgment Number : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ
- Header Length : โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ option แต่ต้องไม่เกิน 60 ไบต์

- Flag : เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็คเก็ต TCP ขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังนี้

-

Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent pointer)
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (reset) เนื่องจากในกรณีที่เกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น โฮสต์มีปัญหา ให้เริ่มต้นสื่อสารกันใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

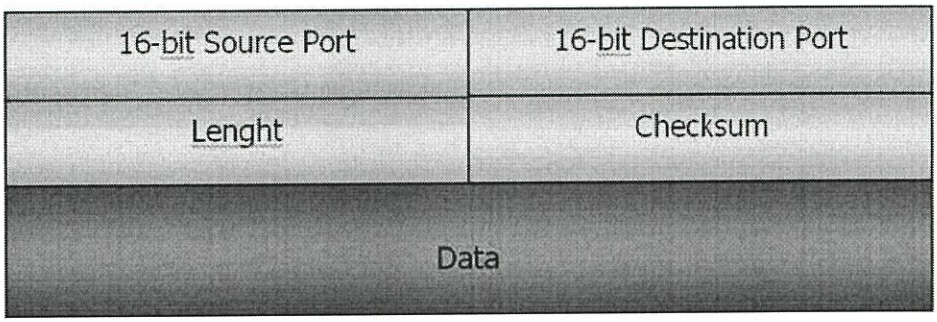
ตารางที่ 2.1 ประเภทของ Flag ในเฮดเดอร์ของ TCP

Flag ในเฮดเดอร์ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP segment เนื่องจากข้อมูลในเฮดเดอร์ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการสทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน flag จะเป็นตัวกำหนดว่าให้ใช้งานฟิลด์ไหน เช่น ฟิลด์ Acknowledgment number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลในฟิลด์ ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ ซึ่งถ้าไม่มี flag เป็นตัวกำหนดก็อาจจะมีการนำข้อมูลมาใช้ และก่อให้เกิดความผิดพลาดได้

2.4.3 ยูสเซอร์ไดอะแกรมโปรโตคอล (User Datagram Protocol :UDP)

เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล

กลไกการตรวจสอบโดย checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง, หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของ UDP เอง จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด



รูปที่ 2.7 แพ็กเกต UDP เฮดเดอร์

มีรายละเอียด ดังนี้

- Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้
- Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม
- UDP Length : ความยาวของดาต้าแกรม ทั้งส่วน Header และ data นั่นหมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
- Checksum : เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

ชั้นที่ 4 : ชั้นสื่อสารการประยุกต์ (Application Layer) มีโปรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือน เรียกว่า TELNET โปรโตคอลสำหรับการจัดการแฟ้มข้อมูล เรียกว่า File Transfer Protocol (FTP) และโปรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า Simple Mail Transport Protocol (SMTP) โดยโปรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่องโฮสต์ที่อยู่ไกลออกไปโดยผ่านอินเตอร์เน็ต และสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่องโฮสต์นั้น โปรโตคอลสำหรับการจัดการแฟ้มข้อมูลช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใดๆก็ได้ โปรโตคอลสำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

บทที่ 3

ซิมเพิลเนตเวิร์คเมเนจเมนต์โปรโตคอล

(Simple Network Management Protocol: SNMP)

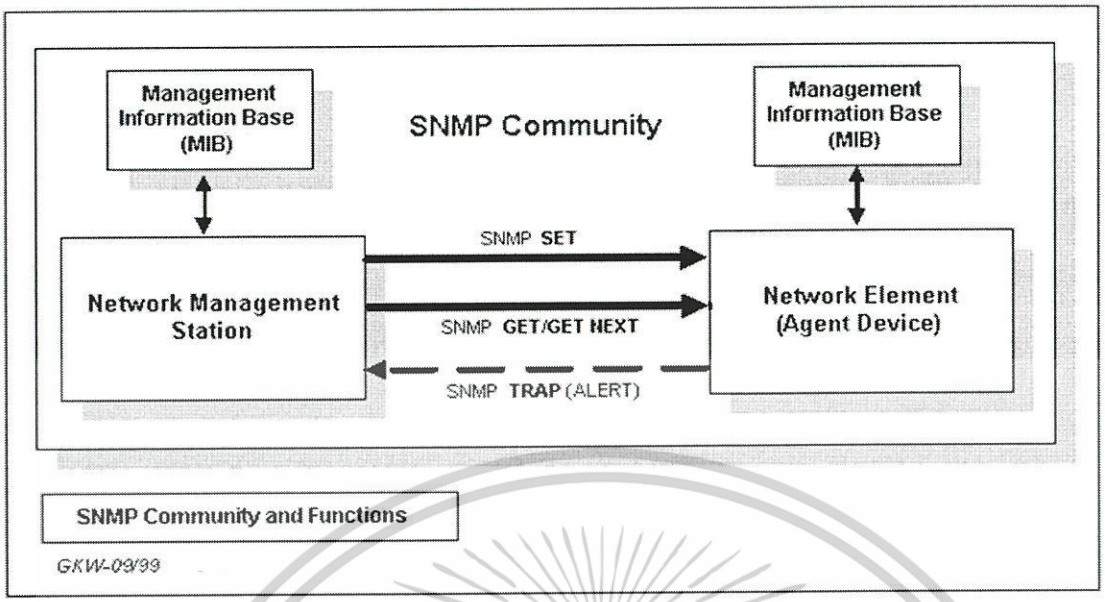
3.1 ซิมเพิลเน็ตเวิร์คเมเนจเมนต์โปรโตคอล (Simple Network Management Protocol)

เนื่องจากในปัจจุบันเครือข่ายมีขนาดใหญ่และมีความซับซ้อนมากขึ้น ทำให้การจัดการกับระบบเป็นไปอย่างลำบาก จึงมีการพัฒนาโปรโตคอลที่ใช้ในการบริหารเครือข่ายขึ้น (Network Management Protocol)

SNMP เป็นโปรโตคอลที่ใช้ในการบริหารเครือข่ายประเภทหนึ่ง โดยสนับสนุนอุปกรณ์ตั้งแต่รีพีทเตอร์ (Repeater) ไปจนถึงเครื่องคอมพิวเตอร์ความสามารถระดับสูง (Supercomputer) โดยในตอนเริ่มแรกได้พัฒนาโดยใช้ TCP/IP โปรโตคอลและต่อมาได้ขยายไปยังโปรโตคอลอื่นๆด้วย

ในอุปกรณ์ที่สนับสนุน SNMP จะมีตัวสนับสนุนระบบเอเจนต์ติดตั้งอยู่ ซึ่งเอเจนต์นี้จะรับข้อความจากตัวเครื่องที่ใช้ในการควบคุมระบบซึ่งจะอ่านค่า หรือการกำหนดค่าของอุปกรณ์ ซึ่งจะส่งข้อความตอบ สนองกลับไป เอเจนต์นอกจากจะคอยการตอบคำถามข้อมูลจากตัวเครื่องที่ใช้ในการควบคุมระบบแล้ว เมื่อมีปัญหาเกิดขึ้นยังสามารถส่งข้อความไปเตือนตัวเครื่องที่ใช้ในการควบคุมระบบได้ ซึ่งเรียกว่าแทรป (Trap) และยังมีความสามารถในการควบคุมการทำงานเครือข่ายอีกด้วย โดยสามารถดูปริมาณการไหลเวียนของข้อมูลในเครือข่าย, ระดับประสิทธิภาพของการทำงาน และข้อมูลพื้นฐานต่างๆ

โดยข้อมูลต่างๆในเครือข่ายส่วนมากจะถูกเก็บอยู่ในอุปกรณ์ต่างๆในเครือข่าย ข้อมูลที่นำมาบริหารเครือข่ายได้มีการกำหนดให้เก็บในมาตรฐานเดียวกัน (Standard MIBs) และผู้ผลิตสามารถเก็บข้อมูลที่มีเฉพาะอุปกรณ์ของตนได้เพิ่มเข้าไปในมาตรฐาน (Enterprise Specific MIBs) วิธีในการอ้างถึงข้อมูลต่างๆไม่ว่าจะเป็นการอ่านหรือเขียนจะต้องมีการระบุถึงชื่อของสิ่งที่ต้องการ



รูปที่ 3.1 การทำงานของ SNMP

3.1.1 โครงสร้างของ SNMP

ระบบที่ทำการจัดการหรือบริหารเครือข่ายด้วย SNMP จะประกอบด้วยส่วนประกอบต่างๆ ดังนี้

1. เครื่องที่ใช้ในการบริหารระบบ (Management Station) เป็นเครื่องที่มี แอพพลิเคชันที่ช่วยให้ผู้ใช้หรือผู้บริหารทำการบริหารเครือข่ายได้ โดยแอพพลิเคชันที่ใช้ในการบริหารนี้จะไม่มีมาตรฐานแน่นอนขึ้นอยู่กับผู้ผลิต ส่วนใหญ่จะช่วยในการดึงข้อมูลจากอุปกรณ์ต่างๆให้แก่ผู้ใช้ หรือช่วยให้ผู้ใช้สามารถแก้ไขข้อมูลบางอย่างได้ หรือช่วยให้ผู้ใช้สามารถสร้างแอพพลิเคชันได้เอง

2. อุปกรณ์ต่างๆในเครือข่าย โดยอุปกรณ์เหล่านี้จะต้องมีการติดตั้งระบบเอเจนต์เข้าไป โดยเอเจนต์จะทำการรับข้อความ (Message) ทำการรับข้อความที่ส่งมาจากเครื่องที่ทำการบริหารระบบ แล้วตอบสนองตามการร้องข้อนั้นๆ หรือเมื่อเกิดเหตุการณ์สำคัญหรือเหตุกาณ์ผิดปกติ เอเจนต์ก็จะส่งข้อความไปยังเครื่องที่ใช้ในการบริหารระบบเองโดยไม่ต้องมีการร้องขอ

โดยข้อมูลต่างๆที่เอเจนต์ที่ส่งกลับไปให้ผู้ที่ทำการร้องขอข้อมูลได้มาจากฐานข้อมูลทาง Logical ที่เก็บข้อมูลต่างๆของอุปกรณ์นั้นๆเรียกว่า MIBs

3. ข้อความต่างๆที่โต้ตอบระหว่างเครื่องที่บริหารระบบกับอุปกรณ์ในเครือข่าย ได้แก่

- Get Request เป็นการร้องขอข้อมูลต่างๆจากผู้บริหารระบบไปยังเอเจนต์เพื่ออ่านค่าต่างๆของอุปกรณ์

- Get Response เป็นการตอบสนองของเอเจนต์ต่อการร้องขอของผู้บริหารเครือข่าย

- Get Next Request เป็นการร้องขอข้อมูลต่างๆจากผู้บริหารเครือข่ายไปยังเอ-เจนต์เพื่ออ่านค่าต่างๆของอุปกรณ์โดยจะเป็นข้อมูลตามลำดับถัดไปจากข้อมูลอันเดิม

- Set Request เป็นการเปลี่ยนแปลงค่าของ MIBs โดยการร้องขอของผู้บริหารเครือข่าย

- Trap เป็นการส่งข้อความเตือนแก่ผู้บริหารเครือข่ายโดยเอเจนต์

- โดยใน Get Request และ Get Response ประกอบด้วยข้อมูลต่างๆคือ

- Version เป็นเวอร์ชันของ SNMP Protocol

- Community Name เป็นเหมือนระบบรักษาความปลอดภัยคือเป็นการกำหนดสิทธิในการจะเข้าไปกระทำการใดๆต่อเมเนจออปเจ็กต์ได้บ้าง เช่น สามารถอ่านได้อย่างเดียว สามารถอ่านและเขียนได้ สามารถเขียนได้ หรือ ไม่สามารถทำอะไรได้

- Command เป็นชนิดของข้อความที่ส่งไปเช่น Get Request, Get Response, Get Next Request, Set Request และ Trap

- Request ID เป็นหมายเลขสำหรับทำการจับคู่ระหว่างข้อความที่ส่งไปกับข้อความที่ได้รับกลับเข้ามา โดยถ้ามีค่าเท่ากันแสดงว่าเป็นการตอบกลับมาโดยข้อความที่ส่งไป

- Error Status ใช้ในการบอกว่าเกิดข้อผิดพลาดขึ้นมา โดยในตอนที่ทำการร้องขอจะใส่ค่า 0 ลงไปเพื่อบอกว่ายังไม่มีข้อผิดพลาดเกิดขึ้นแต่ถ้าค่าที่ตอบสนองกลับมาไม่เท่ากับ 0 แสดงว่ามีข้อผิดพลาดเกิดขึ้นโดยมีค่าดังนี้

- 0 แสดงว่าไม่มีความผิดพลาดเกิดขึ้น

- 1 เป็นการตอบสนองต่อการร้องขอที่มีจำนวนมากจนเกินกว่าขนาดสูงสุดของแพ็คเก็ตตามที่โปรโตคอลกำหนด

- 2 เกิดข้อผิดพลาดเนื่องจากไม่มี Object Identifier ตามที่ได้ร้องขอไป

- 3 เป็นการใช้ประเภทของข้อมูลที่จะทำการเขียนลงไปผิดพลาด

- 4 มีความต้องการเขียนข้อมูลที่สามารถอ่านได้อย่างเดียว

- 5 ไม่สามารถตอบสนองต่อการร้องขอได้โดยเหตุผลอื่นๆนอกจากข้างต้น

- Error Index เมื่อเกิดข้อผิดพลาดโดนรายงานใน Error Status ส่วน Error Index จะบอกถึง Object Identifier ที่ทำให้เกิดความผิดพลาดขึ้น

- Object Identifier แสดงถึง Manage Object ที่ต้องการจะเข้าถึง

- Value จะเป็นค่าที่ตอบสนองกลับมาโดยในตอนเริ่มแรกที่ส่งออกไปจะใส่ค่าเป็น NULL

โดยใน Get Next Request จะใช้ในการเข้าถึงข้อมูลแบบลำดับโดยค่าใน Object Identifier จะเป็น Object Identifier ของข้อมูลที่อยู่ก่อนข้อมูลที่ต้องการ

ส่วนข้อความแทรป(Trap)จะประกอบด้วยข้อมูลดังนี้

- Version เป็นเวอร์ชันของ SNMP Protocol

- Community Name เป็นเหมือนระบบรักษาความปลอดภัยคือเป็นการกำหนดสิทธิในการจะเข้าไปกระทำการใดๆต่อเมเนจออปเจ็กต์ได้บ้าง เช่น สามารถอ่านได้อย่างเดียว สามารถอ่านและเขียนได้ สามารถเขียนได้ หรือ ไม่สามารถทำอะไรได้เลย

- Command เป็นชนิดของข้อความที่ส่งไปเช่น Get Request, Get Response, Get Next Request, Set Request และ Trap

- Enterprise Field เป็นค่า Object Identifier ที่บอกถึงชื่อผลิตภัณฑ์ที่ส่งข้อความแทรปนั้น

- Network Address จะบอกถึง ไอพีแอดเดรสของอุปกรณ์ที่ส่งข้อความแทรปนั้น

- Generic Trap เป็นค่าของการแทรปซึ่งประกอบด้วย

- Cold Start (0) แสดงว่าอุปกรณ์ที่ส่งข้อความแทรปมีการเริ่มต้นการทำงานใหม่และอาจมีการเปลี่ยนแปลง Configuration

- Warm Start (1) แสดงว่าอุปกรณ์ที่ส่งข้อความแทรปมีการเริ่มต้นการทำงานใหม่แต่จะไม่มีการเปลี่ยนแปลง Configuration

- Link Down (2) แสดงว่าเกิดความล้มเหลวที่ลิงค์ของอุปกรณ์ที่ส่งข้อความแทรป

- Link Up (3) ลิงค์ของอุปกรณ์ที่ส่งข้อความแทรปสามารถ กลับมาใช้งานได้เป็นปกติ

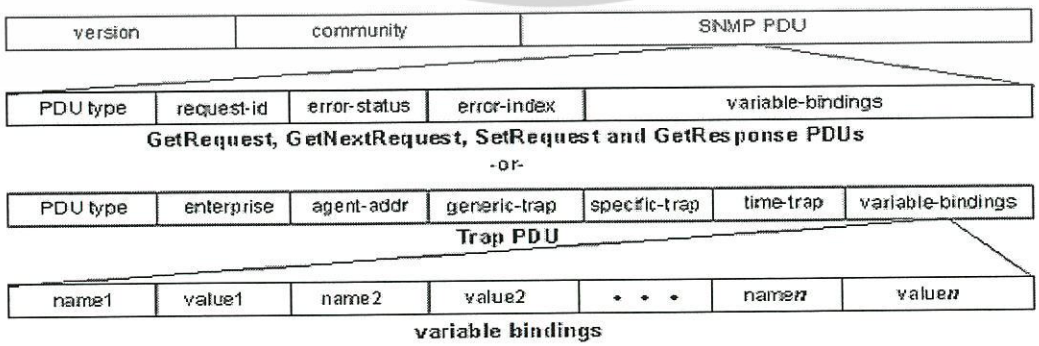
- Authentication Failure (4) รายงานถึงความผิดพลาดที่ได้รับข้อความที่มีชื่อกลุ่ม (Community Name) ไม่ถูกต้อง

- EgpNeighbor Lost (5) รายงานถึง EGP โปรโตคอลของอุปกรณ์ข้างเคียงไม่สามารถทำงานได้

- Enterprise Specific (6) เป็นแทรปที่เจ้าของผลิตภัณฑ์เป็นผู้กำหนดขึ้นเอง

- Specification Trap เป็นชนิดของแทรปที่ผู้ผลิตกำหนดขึ้นเองตามแต่ผลิตภัณฑ์

- Time Trap บอกถึงช่วงเวลาที่เกิดเหตุการณ์ที่ทำให้เกิดการแทรปขึ้นจนกระทั่งมีการสร้างข้อความแทรป



รูปที่3.2 โครงสร้างของ SNMP message

รายละเอียดของ SNMP ในเวอร์ชันต่างๆ ดังนี้

3.1.2 SNMP Version 1 (SNMPv1)

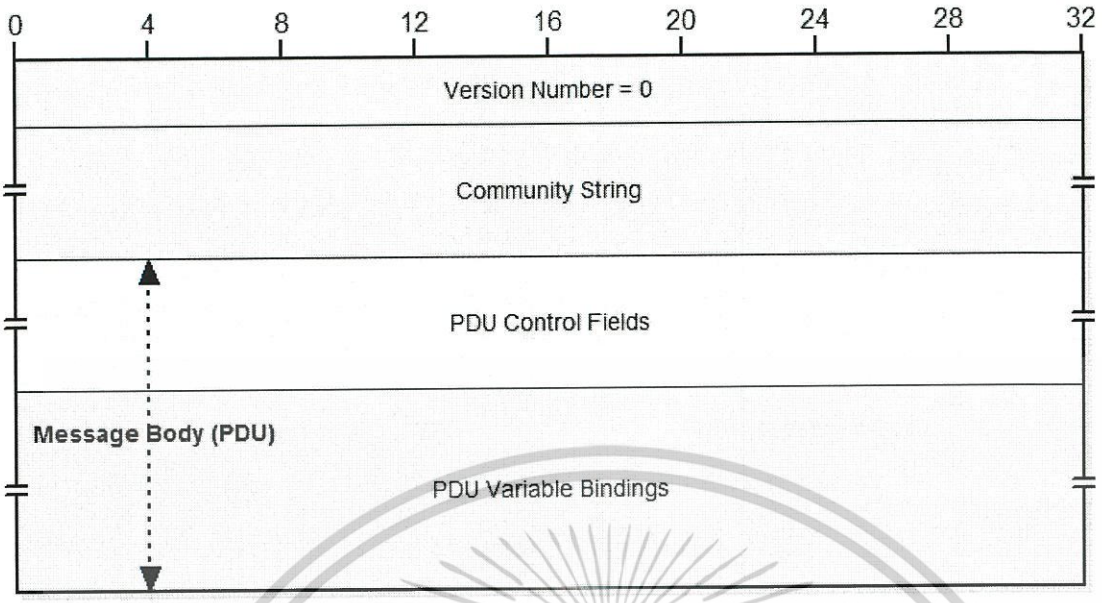
รูปแบบเมสเสจของ SNMP version1

รูปแบบเมสเสจจะเป็นเหมือน SNMP ปกติ อันดับแรกการใช้ต้องกำหนดรูปแบบของเมส-เสจในโปรโตคอล Simple Network Management Protocol (SNMP) SNMP version 1 (SNMPv1) เวอร์ชันแรกของ SNMP ซึ่งเป็นที่รู้จักกันดีมากที่สุดในเรื่องความสัมพันธ์ที่เข้าใจได้ง่ายเปรียบเทียบกับเวอร์ชันอื่นๆ เมื่อไตร่ตรองในรูปแบบของเมสเสจ ซึ่งค่อนข้างจะใช้แบบทั่วๆไป

ทั่วไปรูปแบบเมสเสจของ SNMPv1 เป็นสิ่งที่ห่อหุ้ม (wrapper) ซึ่งประกอบด้วยส่วนเฮดเดอร์และส่วนที่ครอบคลุมของ PDU (Protocol Data Unit) SNMPv1 ไม่มีความจำเป็นต้องมีส่วนเฮดเดอร์ เพราะมีวิธีตรวจสอบใน SNMPv1 ที่ใช้community เป็นเกณฑ์อันดับแรก รูปแบบเมสเสจทั้งหมดของ SNMPv1 แสดงดังตารางที่ 3.1 และรูป 3.3

Field Name	Syntax	Size (bytes)
Version	*Integer*	4
Community	*Octet String*	Variable
PDU	—	Variable

ตารางที่ 3.1 รูปแบบเมสเสจทั้งหมดของ SNMPv1



รูปที่3.3 รูปแบบเมสเสจทั้งหมดของ SNMPv1

3.1.2.1 รูปแบบSNMPv1 PDU

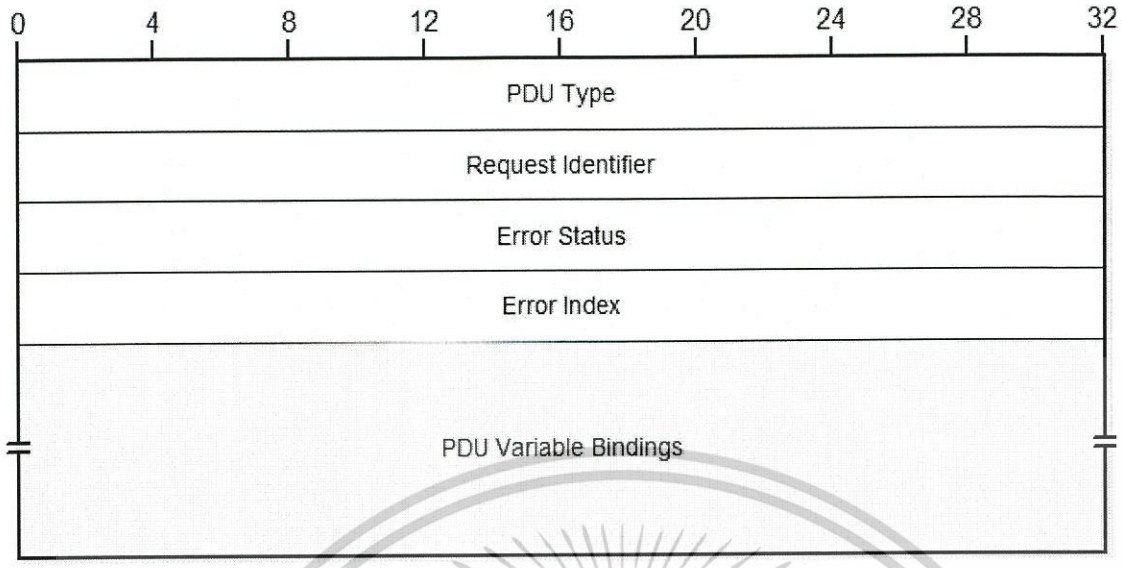
รูปแบบ PDUs ใน SNMPv1 จะเหมือนกัน และมีข้อยกเว้นในกรณีพิเศษคือ Trap-PDU การศึกษาความหมายของแต่ละฟิลด์ (field) ใน PDU จะขึ้นอยู่กับเมสเสจที่เจาะจง ยกตัวอย่าง ในฟิลด์ ErrorStatus เท่านั้นที่มีการตอบกลับ โดยไม่มีการร้องขอ และค่าออปเจ็กต์ที่เพิ่มเติมใช้แตกต่างกันในการร้องขอและการตอบกลับ

รูปแบบทั่วไปของ SNMPv1 PDU

ตารางที่3.2 และรูปที่ 3.4 แสดงรูปแบบทั่วไปที่เกี่ยวกับ SNMPv1 PDUs GetRequest-PDU, GetNextRequest-PDU, SetRequest-PDU and GetResponse-PDU:

Field Name	Syntax	Size (bytes)
PDU Type	*Integer (Enumerated)*	4
Request ID	*Integer*	4
Error Status	*Integer (Enumerated)*	4
Error Index	*Integer*	4
Variable Bindings	Variable	Variable

ตารางที่ 3.2 รูปแบบทั่วไปของ SNMPv1 PDU



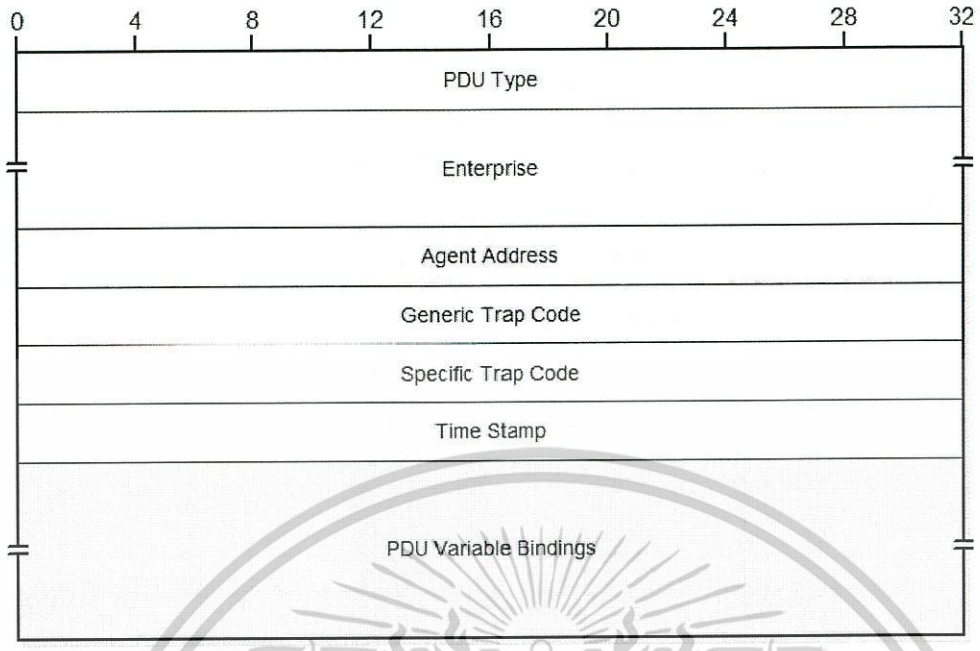
รูปที่3.4 รูปแบบทั่วไปของ SNMPv1 PDUs

3.1.2.2 รูปแบบของ SNMPv1 Trap-PDU

ตารางที่ 3.3 และรูปภาพ 3.5 แสดงรูปแบบพิเศษของ SNMPv1 Trap-PDU

Field Name	Syntax	Size (bytes)
PDU Type	*Integer (Enumerated)*	4
Enterprise	Sequence of *Integer*	Variable
Agent Addr	*NetworkAddress*	4
Generic Trap	*Integer (Enumerated)*	4
Specific Trap	*Integer*	4
Time Stamp	*TimeTicks*	4
Variable Bindings	Variable	Variable

ตารางที่ 3.3 แสดงรูปแบบพิเศษเกี่ยวกับ SNMPv1 Trap-PDU



รูปที่ 3.5 แสดงรูปแบบพิเศษเกี่ยวกับ SNMPv1 Trap-PDU

3.1.3 SNMP Version 2 (SNMPv2)

รูปแบบเมสเสจของ SNMP version2

หลังจาก SNMP เวอร์ชันแรกถูกใช้เป็นเวลาหลายปี แน่นอนว่าสิ่งที่เวอร์ชันใหม่ต้องน่าสนใจและมีการปรับปรุงประสิทธิภาพการกำหนดออปเจ็กต์ให้ดีขึ้น สิ่งนี้ทำให้เกิดการพัฒนาของ SNMP version 2 ในตอนแรกเริ่ม ซึ่งเป็นจุดมุ่งหมายที่จะเพิ่มการทำงาน SNMPv1 ในหลายๆจุดและประกอบด้วย การกำหนดออปเจ็กต์แบบ MIB (management information base) และมีการทำงานของโปรโตคอลและการรักษาความปลอดภัย ในขอบเขตการทำงานด้านการรักษาความปลอดภัยทำให้เกิดการแพร่หลายของSNMPv2ที่มีความหลากหลาย (variants) ดังจะได้อธิบายต่อไป

ตั้งแต่มีความความแตกต่างของ SNMPv2 และมีรูปแบบของหลายแบบที่เกี่ยวกับ SNMPv2 การทำงานของโปรโตคอล SNMPv2 ที่เปลี่ยนแปลงจาก SNMPv1 ซึ่งจำเป็นที่ต้องดัดแปลงรูปแบบบางส่วนของ SNMPv2 PDUs อย่างไรก็ตาม การทำงานของโปรโตคอลจึงเป็นการเปลี่ยนแปลงในส่วนของ SNMPv2 ทั้งหมด ความแตกต่างระหว่างความหลากหลายของ SNMPv2 คือส่วนของวิธีการการรักษาความปลอดภัย ดังนั้นผลที่ตามมาคือรูปแบบของ PDU จะเป็นประเภท SNMPv2 เหมือนกันทั้งหมด ในขณะที่รูปแบบเมสเสจทั้งหมดแตกต่างในแต่ละความหลากหลาย

ในความหลากหลายของ SNMPv2 นั้นมีการกำหนดอยู่ 4 แบบ

1. the original SNMPv2 (SNMPv2p); แบบแรกเริ่ม
2. community-based SNMPv2 (SNMPv2c) แบบทำงานที่ใช้คอมมิวนิตี้เป็นเกณฑ์
3. user-based SNMPv2 (SNMPv2u) แบบทำงานที่ใช้ผู้ใช้เป็นเกณฑ์
4. SNMPv2 star (SNMPv2*)

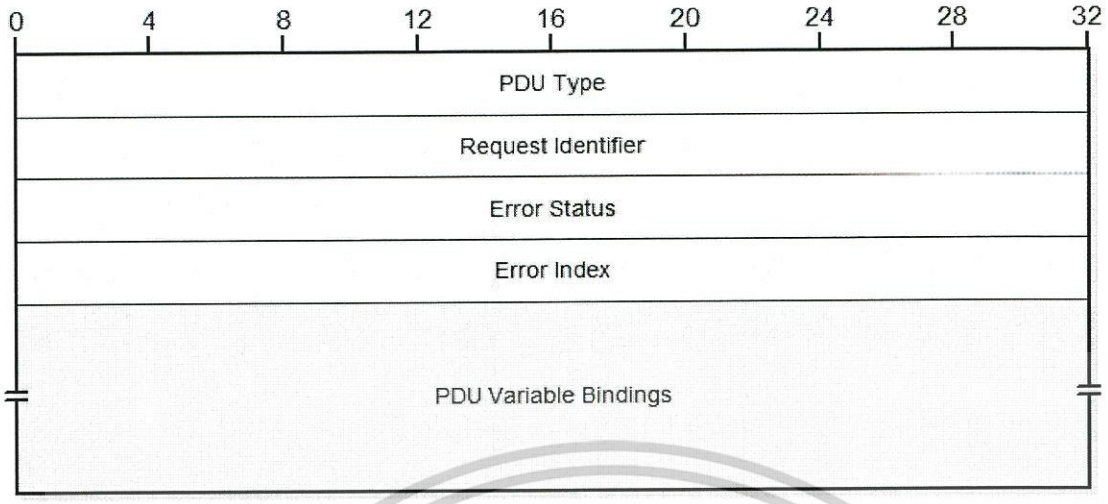
ใน 3 แบบแรกมีในเอกสารประกอบในกลุ่มของ SNMP RFC ที่เป็นมาตรฐาน และส่วนแบบที่ 4 ไม่มี โดยโครงสร้างรูปแบบของเมสเสจทั้งหมดในแต่ละแบบถูกอภิปรายเกี่ยวกับเรื่องการจัดการหรือมาตรฐานการรักษาความปลอดภัย ซึ่งทำการอ้างอิงร่วมกันกับมาตราฐานSNMPv2ที่เกี่ยวกับรูปแบบของ PDU (RFC 1905)

Field Name	Syntax	Size (bytes)
PDU Type	*Integer* (Enumerated)	4
Request ID	*Integer*	4
Error Status	*Integer* (Enumerated)	4
Error Index	*Integer*	4
Variable Bindings	Variable	Variable

ตารางที่ 3.4 รูปแบบทั่วไปของ SNMPv2 PDU

Error Status Value	Error Code
0	noError
1	tooBig
2	noSuchName
3	badValue
4	readOnly
5	genErr
6	noAccess
7	wrongType
8	wrongLength
9	wrongEncoding
10	wrongValue
11	noCreation
12	inconsistentValue
13	resourceUnavailable
14	commitFailed
15	undoFailed
16	authorizationError
17	notWritable
18	inconsistentName

ตารางที่ 3.5 แสดงค่าในฟิลด์ของ Error Status ใน SNMPv2 PDU



รูปที่ 3.6 รูปแบบทั่วไปของ SNMPv2 PDU

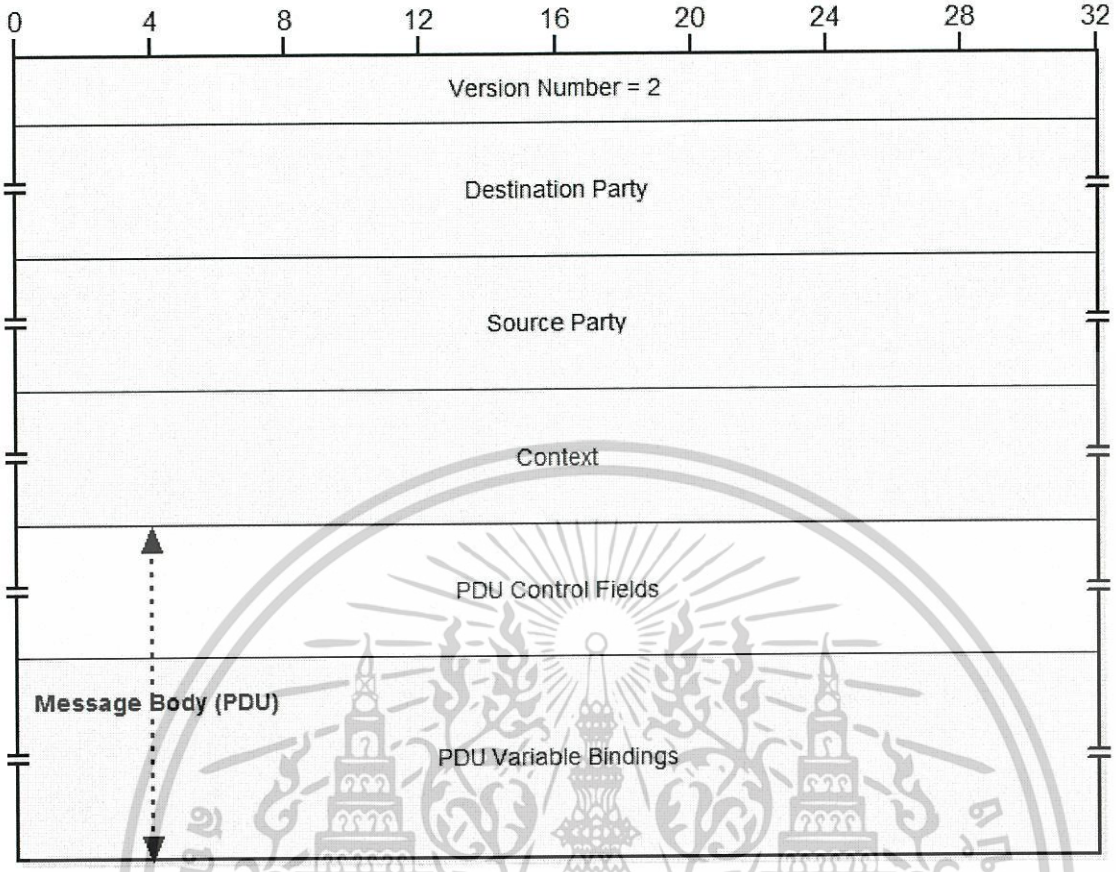
รูปแบบเมสเสจทั้งหมดที่เกี่ยวกับ SNMPv2p, SNMPv2c และ SNMPv2u มีรายละเอียดดังนี้

3.1.3.1 รูปแบบเมสเสจของ SNMP Version 2 (SNMPv2p)

รูปแบบการรักษาความปลอดภัยที่ใช้กลุ่ม (party) เป็นเกณฑ์นั้นค่อนข้างมีความซับซ้อน แต่พื้นฐานของการส่งเมสเสจในเวอร์ชันนี้ได้อธิบายผ่านการนิยามของการจัดการการสื่อสาร ซึ่งได้ระบุกลุ่มของต้นทางและกลุ่มของปลายทางและทำการอ้างอิงถึงบริบทที่เกี่ยวกับการสื่อสาร รูปแบบเมสเสจทั้งหมดนั้นมีการอธิบายรายละเอียด ใน RFC 1445 ข้อมูลที่รวบรวมได้ตามตารางที่ 3.6 และแสดงในรูป 3.7

Field Name	Syntax	Size (bytes)
Version	*Integer*	4
Dst Party	Sequence of *Integer*	Variable
Src Party	Sequence of *Integer*	Variable
Context	Sequence of *Integer*	Variable
PDU	—	Variable

ตารางที่ 3.6 รูปแบบทั่วไปของ SNMP Version 2 (SNMPv2p)



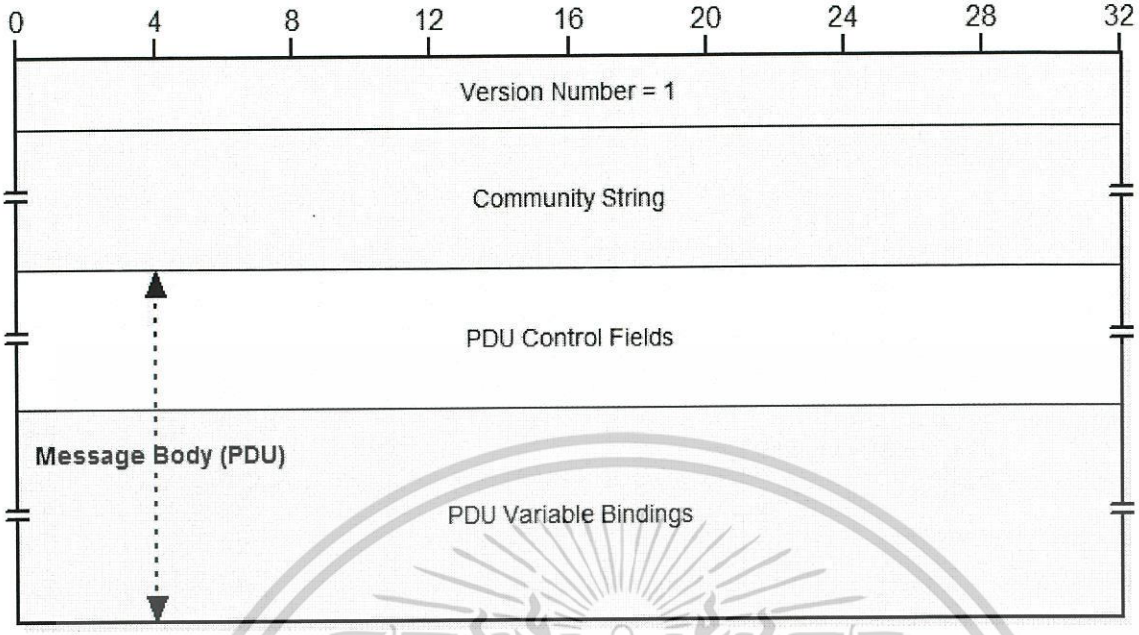
รูปที่ 3.7 รูปแบบทั่วไปของ SNMP Version 2 (SNMPv2p)

3.1.3.2 รูปแบบเมสเสจของ SNMP Version 2 ที่มีรูปแบบทำงานที่ใช้คอมมิวนิตี้เป็นเกณฑ์ Community-Based SNMP Version 2 (SNMPv2c)

รูปแบบนี้ของ SNMPv2 มีจุดประสงค์ในการดำเนินการให้มีการเพิ่มโปรโตคอลใหม่ โดยนำ SNMPv2p มาใช้ แต่กลับไปใช้รูปแบบการรักษาความปลอดภัยแบบง่ายของ SNMPv1 อย่างเช่น ที่ระบุในเอกสารที่เกี่ยวกับ SNMPv2c, RFC 1901 ที่อธิบายในเรื่องรูปแบบเมสเสจทั้งหมดนั้นมีลักษณะเหมือนกับ SNMPv1 ยกเว้นในส่วนหมายเลขเวอร์ชัน (version number) ที่เปลี่ยนไป ดังที่แสดงในตารางที่ 3.7 และรูปที่ 3.8

Field Name	Syntax	Size (bytes)
Version	*Integer*	4
Community	*Octet String*	Variable
PDU	—	Variable

ตารางที่ 3.7 รูปแบบทั่วไปของCommunity-Based SNMP Version 2 (SNMPv2c)



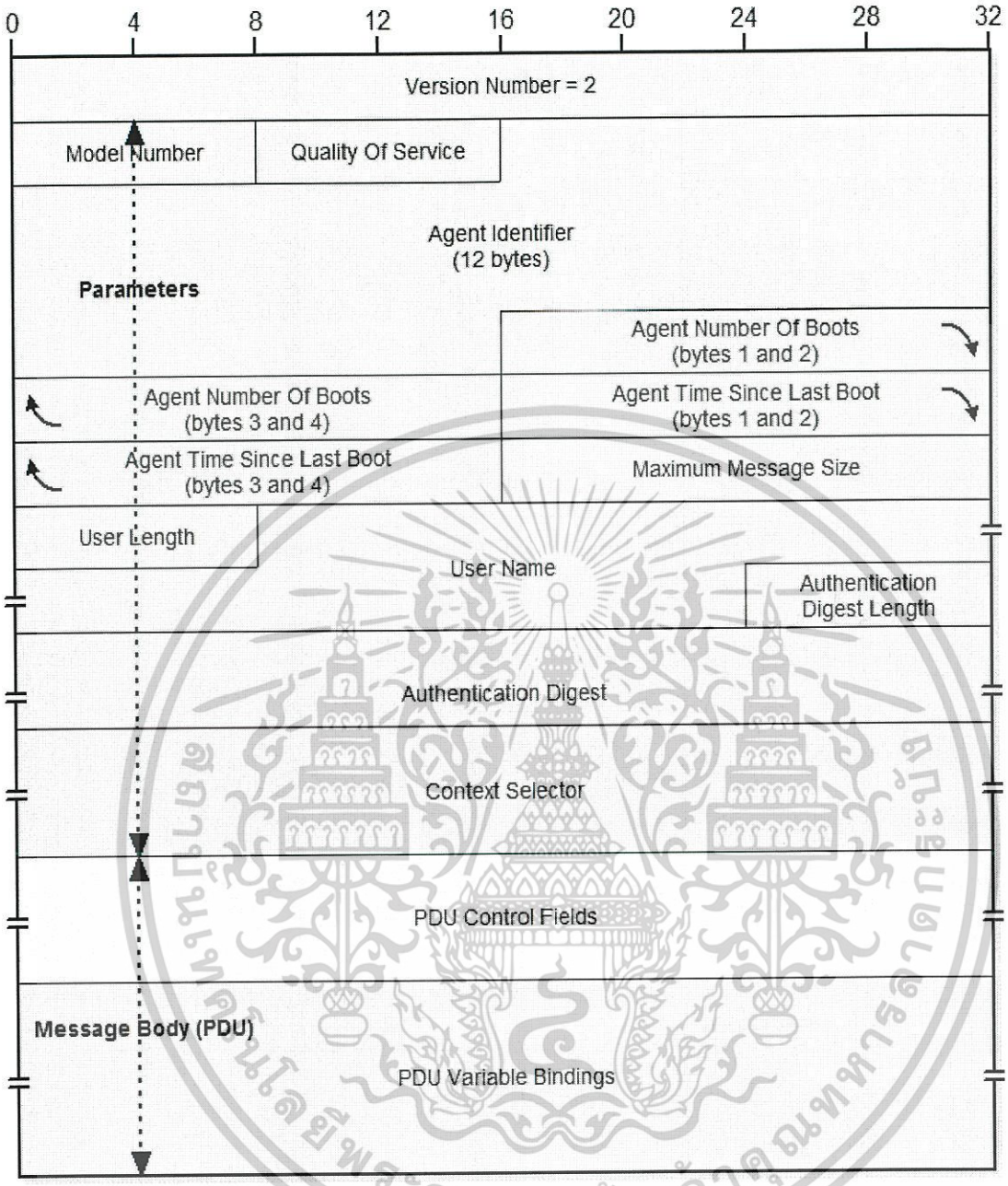
รูปที่ 3.8 รูปแบบทั่วไปของCommunity-Based SNMP Version 2 (SNMPv2c)

3.1.3.3 รูปแบบเมสเสจของ SNMP Version 2 ที่มีรูปแบบทำงานที่ใช้ผู้ใช้เป็นเกณฑ์ User-Based SNMP Version 2 (SNMPv2u)

SNMPv2u มีสามารถกำหนดรูปแบบการรักษาความปลอดภัยได้ ในขณะที่ SNMPv2c ถูกทำให้ได้มาตรฐาน RFC 1910 ได้ระบุรูปแบบการรักษาความปลอดภัยของ SNMPv2u และอธิบายรูปแบบของเมสเสจในตารางที่ 3.8 และรูปที่ 3.9

Field Name	Syntax	Size (bytes)
Version	*Integer*	4
Parameters	*Octet String*	Variable
PDU	—	Variable

ตารางที่ 3.8 รูปแบบทั่วไปของUser-Based SNMP Version 2 (SNMPv2u)



รูปที่ 3.9 รูปแบบทั่วไปของUser-Based SNMP Version 2 (SNMPv2u)

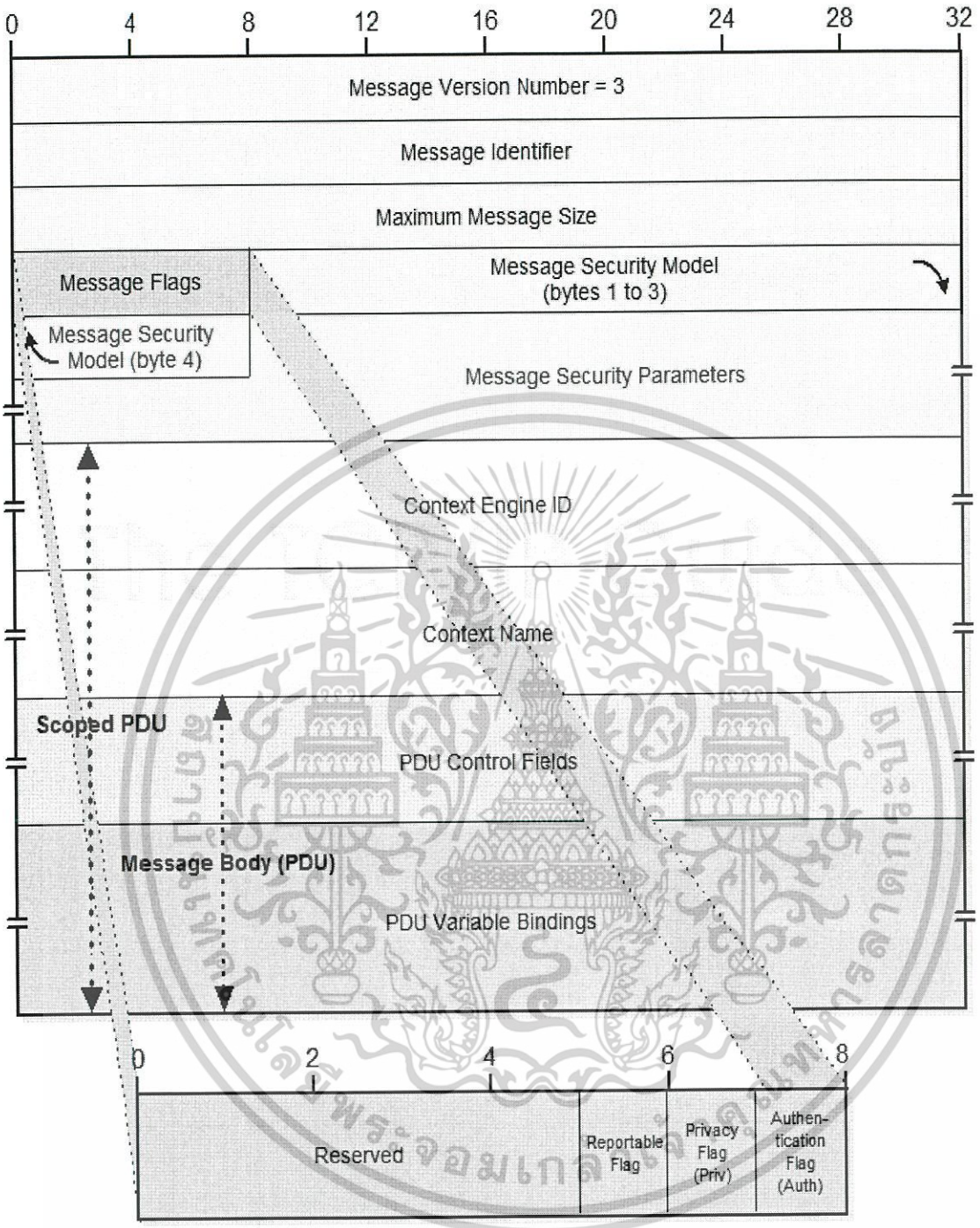
3.1.4 SNMP Version 3 (SNMPv3)

รูปแบบเมสเสจ SNMP Version 3 (SNMPv3)

ภายหลังปี 1990 SNMP Version 3 ถูกสร้างเพื่อใช้แก้ปัญหาต่างๆที่เกิดขึ้นจากความหลากหลายของ SNMPv2 โครงร่างของ SNMPv3 นำส่วนประกอบต่างๆที่ถูกสร้างใน SNMPv2 มาใช้ ซึ่งจะประกอบด้วยการทำงาน ประเภทของPDU และรูปแบบของเหมือน SNMPv2 ในระหว่างที่ทำการเปลี่ยนแปลงใน SNMPv3 ที่ประกอบด้วยการกำหนดวิธีการรักษาความปลอดภัยที่สามารถเปลี่ยนแปลงได้ง่ายจึงมีความยืดหยุ่นมากขึ้นและปัจจัย ที่อนุญาตให้ใช้เทคนิคการรักษาความปลอดภัยได้หลายรูปแบบ

รูปแบบเมสเสจทั่วไปของ SNMPv3 ยังทำตามวิธีเดิมของเมสเสจทั้งหมด คือสิ่งที่ห่อหุ้ม ซึ่งมีส่วนเฮดเดอร์และส่วนที่บรรจุ PDU อย่างไรก็ตาม แนวคิดในเวอร์ชัน 3 มีการกลั่นกรองเพิ่มขึ้น ฟิลด์ในส่วนเฮดเดอร์จะถูกแบ่งแยกเป็นส่วนๆด้วยตัวเอง มีอยู่ 2 แบบคือมีการจัดการด้วยความปลอดภัยและส่วนที่ไม่มีการจัดการกับสภาวะที่ต้องการความปลอดภัย ฟิลด์ที่ไม่มีความปลอดภัยธรรมดาจะมีการดำเนินงานใน SNMPv3 ในขณะที่ใช้ฟิลด์ที่มีความปลอดภัยสามารถทำให้ได้ข้อมูลใหม่ๆ โดยขึ้นกับแต่ละรูปแบบของการรักษาความปลอดภัยของ SNMPv3 และผ่านกระบวนการโมดูลใน SNMP เอนทิตีซึ่งจัดการกับความปลอดภัย การหาทางแก้ไขปัญหามีจำนวนมากซึ่งสามารถเปลี่ยนแปลงได้ ในเวลาเดียวกันก็ทำการหลีกเลี่ยงปัญหาที่เกิดขึ้นเหมือน SNMPv2

รูปแบบเมสเสจทั้งหมดของ SNMPv3 ถูกอธิบายใน RFC 3412 ซึ่งอธิบายการประมวลผลและส่งอย่างรวดเร็วของเมสเสจเวอร์ชัน 3 ดังที่แสดงในตารางที่ 3.9 และรูปที่ 3.10



รูปที่ 3.10 รูปแบบทั่วไปของ SNMP Version 3 (SNMPv3)

Field Name	Syntax	Size (bytes)
Msg Version	*Integer*	4
Msg ID	*Integer*	4
Msg Max Size	*Integer*	4
Msg Flags	*Octet String*	1
Msg Security Model	*Integer*	4
Msg Security Parameters	—	Variable
Scoped PDU	—	Variable

ตารางที่ 3.9 รูปแบบทั่วไปของ SNMP Version 3 (SNMPv3)

3.2 ฐานข้อมูลการจัดการ (Management Information Bases หรือ MIBs)

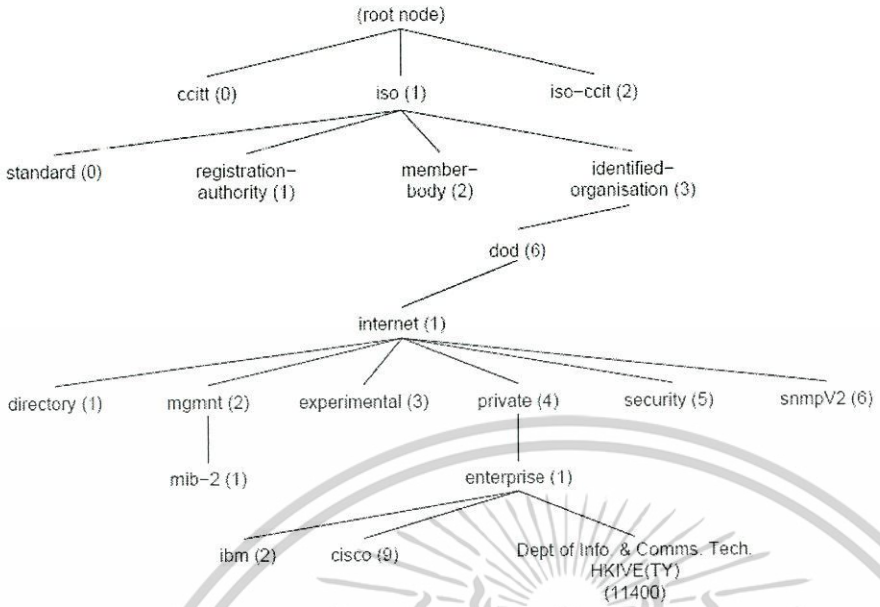
อุปกรณ์ต่างๆที่สนับสนุน SNMP โปรโตคอลจะมีการเก็บข้อมูลที่เกี่ยวข้องลงในฐานข้อมูลเสมือน โดยเรียกว่า ฐานข้อมูลการจัดการ (Management Information Bases หรือ MIBs)เป็นฐานข้อมูลที่ใช้เก็บค่าข้อมูลต่างๆของ เมเนจออปเจ็กต์ (Manage Object) เพื่อใช้สำหรับตั้งค่าหรือรายงานผลของตัวอุปกรณ์ต่างๆ

เมเนจออปเจ็กต์ คือชนิดหรือคลาสของข้อมูลที่เก็บในอุปกรณ์ เช่น ข้อมูลที่เป็นรายละเอียดของอุปกรณ์ (System Description) หรือ สถานะของอินเตอร์เฟซ (Interface Status)

อินสแตนท์ (Instance) คือค่าต่างๆของเมเนจออปเจ็กต์ โดยเมเนจออปเจ็กต์อาจมีอินสแตนท์เพียงค่าเดียว เช่น รายละเอียดของอุปกรณ์ (System Description) หรือ อาจมีหลายค่าเช่น อุปกรณ์ที่มีหลายอินเตอร์เฟซ ดังนั้น MIBs ออปเจ็กต์ที่แสดงสถานะของอินเตอร์เฟซ จะมีอินสแตนท์เท่ากับจำนวนอินเตอร์เฟซที่มี

3.2.1 โครงสร้างของ MIBs

โครงสร้างของ MIBs จะมีลักษณะเหมือนต้นไม้ (Hierarchical Tree Structure) โดยแต่ละโหนดในต้นไม้ (Tree) จะมีชื่อกำกับ เช่น iso,org,dod และเป็นตัวเลขกำกับ (Numeric Identifier) ที่ไม่ซ้ำกันในโหนดลูกที่แตกของมาจากโหนดแม่เดียวกัน เช่นเลข 1,2



รูปที่ 3.11 โครงสร้างของฐานข้อมูลการจัดการ

การอ้างอิงเมเนจออปเจ็กต์จะต้องอ้างอิงถึง Object Identifier โดยสามารถอ้างอิงได้ 2 แบบ คือ แบบสัญลักษณ์ตัวอักษร (Symbolic Identifier) และแบบตัวเลข (Numeric Identifier) เช่นถ้าเราต้องการอ้างถึงเมเนจ ออปเจ็กต์ที่ชื่อ sysdesc เราสามารถอ้างได้ 2 แบบคือ 1.3.6.1.2.1.1.1 หรือ iso.org.dod.internet.mgmt.mib-2.system.sysDescr ซึ่งการอ้างถึงทั้ง 2 แบบจะอ้างอิงไปยังเมเนจออปเจ็กต์เดียวกัน และต้องมีการกำหนดว่าจะอ้างอิงถึงอินสแตนท์ไหนของเมเนจออปเจ็กต์ โดยถ้ามีเพียง 1 อินสแตนท์จะต้องเติม ".0" ต่อท้าย Object Identifier เช่น 1.3.6.1.2.1.1.1.0 หรือ iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 แต่ถ้าเมเนจออปเจ็กต์มีหลายอินสแตนท์ เช่น ต้องการทราบสถานะของอินเตอร์เฟซที่ 3 ของเครื่องสามารถอ้างได้โดย เติมหมายเลขของอินสแตนท์ต่อท้ายเข้าไปใน Object Identifier เช่น 1.3.6.1.2.1.2.2.1.8.3 ซึ่งเป็นสถานะของอินเตอร์เฟซที่ 3 ของเครื่อง

ในกลุ่มฐานข้อมูลการจัดการที่เป็นมาตรฐานจะอยู่ในกลุ่ม Mib-2 ซึ่งจะมีทั้งหมด 11 กลุ่ม โดยแต่ละกลุ่มจะมีเมเนจออปเจ็กต์ที่เกี่ยวข้องอยู่ภายใต้นั้นอีก กลุ่มฐานข้อมูลการจัดการมาตรฐานได้แก่

- System Group ข้อมูลในกลุ่มนี้จะเป็นข้อมูลเกี่ยวกับคอนฟิกกูเรชันต่างๆของอุปกรณ์ เช่น System service จะบอกว่าอุปกรณ์นี้มีการบริการอยู่ในเลเยอร์ใดบ้าง

- Interface Group จะประกอบด้วยข้อมูลที่กับอินเตอร์เฟซของอุปกรณ์ เช่นสถานะปริมาณข้อมูลเข้าและออก

- IfExtension Group เป็นข้อมูลของอินเตอร์เฟซเพิ่มเติมขึ้นมาจากกลุ่มอินเตอร์เฟซ

- Address Translation Group ข้อมูลในกลุ่มนี้จะเกี่ยวกับการแปลงเนตเวิร์คเลเยอร์ Address เช่นแปลงไอพีแอดเดรสไปเป็น Address ที่อยู่ในชั้นต่ำกว่า

- Internet Protocol Group ข้อมูลในกลุ่มนี้จะเกี่ยวกับอินเตอร์เน็ตโปรโตคอล เช่นค่าคอนฟิกกูเรชันเกี่ยวกับอินเตอร์เน็ตโปรโตคอล

- ICMP Group ข้อมูลในกลุ่มนี้จะเกี่ยวกับ ICMPโปรโตคอลเช่น จำนวน ICMPแพ็คเก็ตที่เกิดความผิดพลาด

- TCP Group ข้อมูลในกลุ่มนี้จะเกี่ยวกับ TCPโปรโตคอล เช่น จำนวน TCPแพ็ค-เก็ตที่เกิดความผิดพลาด

- UDP Group ข้อมูลในกลุ่มนี้จะเกี่ยวกับUDPโปรโตคอล เช่น จำนวนUDPแพ็ค-เก็ตที่เกิดความผิดพลาด

- EGP Group ข้อมูลในกลุ่มนี้จะเกี่ยวกับอีจีพีโปรโตคอล เช่น จำนวนEGPแพ็ค-เก็ตที่ได้รับ

- Transmission Groupโดยข้อมูลในกลุ่มนี้จะประกอบด้วยกลุ่มต่างๆของทรานมิสชันเทคโนโลยี

3.2.1.1 ชนิดของข้อมูลของเมเนจออปเจ็กต์

ชนิดของข้อมูลที่ใช้ในฐานข้อมูลการจัดการมีดังนี้

- Integer เป็นชนิดข้อมูลที่เก็บจำนวนเต็ม ที่ขนาดของข้อมูลขึ้นอยู่กับสถาปัตยกรรมของอุปกรณ์

- Octet String เป็นชนิดข้อมูลที่เก็บตัวอักษร โดยค่าออคเทด จะมีค่าอยู่ระหว่าง 0 ถึง 255 โดยหนึ่งออคเทดใช้เก็บหนึ่งตัวอักษร

- Sequence เป็นชนิดข้อมูลใช้สำหรับการกำหนดเมเนจออปเจ็กต์เป็นแบบรายการ (List)

- Sequence of เป็นชนิดข้อมูลใช้สำหรับการกำหนดเมเนจออปเจ็กต์เป็นแบบอาร์เรย์ (Array)

- IP Address เป็นชนิดข้อมูลที่ใช้เก็บหมายเลขไอพีแอดเดรส ขนาด 32 บิต

- Counter เป็นชนิดข้อมูลตัวนับที่จะเพิ่มขึ้นเรื่อยๆ โดยเริ่มจาก 0 ไปจนถึง 4,294,967,295 เมื่อขึ้นไปถึงสูงสุดจะกลับไปเริ่มที่ 0 ใหม่

- Gauge เป็นชนิดข้อมูลตัวนับที่สามารถเพิ่มหรือลดค่าได้ โดยจะมีค่าตั้งแต่ 0 ไปจนถึง 4,294,967,295 เมื่อนับไปจนสูงสุดแล้วจะคงสภาพไว้ไม่กลับมาเริ่มนับ 0 ใหม่

- Time Ticks เป็นชนิดข้อมูลตัวนับเวลาที่มีหน่วยเป็น 1/100 วินาทีจะนับตั้งแต่ 0 ไปจนถึง 4,294,967,295

- Opaque เป็นชนิดข้อมูลแบบพิเศษ โดยค่าที่จะใส่เป็น ออคเทดสตริง (octet string)

3.2.1.2 การเข้าถึงข้อมูล

ในการเข้าถึงข้อมูลจะมีการแบ่งระดับการเข้าถึงข้อมูลดังนี้

- Read Only (RO) เป็นการกำหนดว่าเมเนจออปเจ็กต์สามารถอ่านค่าได้อย่างเดียว
- Read Write (RW) เป็นการกำหนดให้เมเนจออปเจ็กต์สามารถอ่านและเขียนค่าได้
- Write Only (WO) เป็นการกำหนดให้เมเนจออปเจ็กต์สามารถเขียนค่าได้อย่างเดียว
- Not Accessible (NA) เป็นการกำหนดให้เมเนจออปเจ็กต์ไม่สามารถอ่านและเขียนค่าได้

3.2.1.3 สถานะของตัวแปรข้อมูล

จะบ่งบอกถึงสถานะของตัวแปรของข้อมูลโดยมีสถานะดังนี้

- Mandatory (M) บอกถึงสถานะว่าเมเนจออปเจ็กต์ยังมีการใช้งานอยู่
- Optional (O) บอกถึงสถานะว่าเมเนจออปเจ็กต์อาจจะยังมีการใช้งานอยู่
- Obsolete (B) บอกถึงสถานะว่าเมเนจออปเจ็กต์เลิกใช้งานอยู่
- Deprecated (D) บอกถึงสถานะว่าเมเนจออปเจ็กต์กำลังจะไปสู่สถานะออบโซลีท

บทที่ 4

การออกแบบระบบ

4.1 การวิเคราะห์ระบบ

การออกแบบระบบในส่วนของฟังก์ชั่นการค้นหาอุปกรณ์ต้องมีการศึกษาการทำงานของโปรแกรมที่มีการใช้งานกันในปัจจุบันมาศึกษาฟังก์ชั่นการค้นหาอุปกรณ์ที่มีการทำงานที่มีความคล้ายคลึงกันจึงต้องมีการทดลอง

ซึ่งการศึกษาในครั้งนี้ทำเพื่อนำมาประกอบในการออกแบบระบบ เพราะในฟังก์ชั่นการค้นหาหมายเลข IP Address ของอุปกรณ์ที่ทำงานอยู่นั้นสามารถทำได้โดยมี 2 วิธี ซึ่งแต่ละวิธีนั้นจะแตกต่างกันตรงที่กระบวนการการค้นหาอุปกรณ์ที่ใช้คือ

1. ใช้โปรโตคอลไอซีเอ็มพี (Internet Control Message Protocol : ICMP) ซึ่งเรียกว่ากระบวนการนั้นว่า "ping" ในการค้นหาหมายเลข IP Address ของอุปกรณ์ที่มีการทำงานอยู่ แล้วค่อยส่งแพ็คเก็ต SNMP get-Request ซึ่งมีข้อดีดังนี้

- ประหยัดแบนด์วิดธ์ของทราฟฟิกในเครือข่ายเนื่องจากแพ็คเก็ตของ ICMP นั้นมีขนาดน้อยกว่าแพ็คเก็ตของ SNMP
- โปรโตคอล ICMP เป็นที่นิยมใช้กับโปรแกรมที่ใช้กันอย่างแพร่หลายในปัจจุบัน

2. ใช้โปรโตคอลเอสเอ็นเอมพี (Simple Network Management Protocol: SNMP) ที่ให้มีการส่งเมสเสจ Get Request โดยให้มีการตอบ Get Response กลับมาแสดงว่ามีหมายเลข IP Address นั้นอยู่จริง การประยุกต์ใช้แบบนี้มีข้อดี

- เนื่องจากอุปกรณ์ทั่วไปที่ใช้ในเครือข่ายนั้นมีการทำงานของโปรโตคอลเอสเอ็นเอ็มพี (SNMP) ซึ่งเป็นพื้นฐานปกติของทุกอุปกรณ์
- เนื่องจากระบบนี้ได้อ้างอิงถึงโปรโตคอล SNMP เป็นหลัก ในการดึงข้อมูลของอุปกรณ์ จึงนำมาใช้ในฟังก์ชั่นนี้ ทำให้มีมุมมองถึงระบบได้ชัดเจนมากยิ่งขึ้น
- ป้องกันการบล็อคพอร์ตของโปรโตคอล ICMP ซึ่งสามารถทำได้ในrouter

จึงมีการทดลองโปรแกรมที่มีใช้กันทั่วไปนี้ ได้นำโปรแกรม What's up gold กับ โปรแกรม Solarwind มาศึกษาโดยมีการใช้โปรแกรม Wireshark ดักจับแพ็คเก็ตเพื่อให้เข้าใจกลไกการทำงานหรืออัลกอริทึมแล้วสรุปได้ดังนี้

โปรแกรม What's up gold

ฟังก์ชั่นสแกนแบบ SNMP smart scan

1. ส่งแพ็คเก็ต ARP เพื่อถามหา IP Address ของ router ที่ต้องต่อภายใน subnet เดียวกัน แต่ถ้าอยู่ต่าง subnet กัน จะทำการ ping ที่ใช้โปรโตคอล ICMP ค้นหาหมายเลขที่มีการใช้งานอยู่
2. เมื่อได้หมายเลข IP Address ของ router ที่ต้องต่อภายใน subnet เดียวกัน จะทำการส่ง ICMP request ไปยังหมายเลขเป้าหมาย (เป็นหมายเลข IP Address ทั้งหมดที่อยู่ภายใน subnet เดียวกัน)
3. เมื่อที่หมายเลข IP Address ใดตอบ ICMP response กลับมา
4. โปรแกรมจะทำการส่ง SNMP get Request โดยร้องขอ object ที่มีหมายเลข OID 1.3.6.1.2.1.1.2.0 ซึ่งเป็นค่า sysObjectID ที่บ่งบอกถึงหมายเลข OID ที่อ้างอิงไปถึง enterprise ใน subtree (เป็นจุดเริ่มต้นของ MIB)
5. รับ SNMP get Response ที่ดึงข้อมูลมาจากอุปกรณ์ใน MIB

โปรแกรม Solarwind

ฟังก์ชั่นสแกนแบบ SNMP sweep

1. ส่งแพ็คเก็ต ARP เพื่อถามหา IP Address ของ router ที่ต้องต่อภายใน subnet เดียวกัน แต่ถ้าอยู่ต่าง subnet กัน จะทำการ ping ที่ใช้โปรโตคอล ICMP ค้นหาหมายเลขที่มีการใช้งานอยู่
2. เมื่อได้หมายเลข IP Address ของ router ที่ต้องต่อภายใน subnet เดียวกัน จะทำการส่ง ICMP request ไปยังหมายเลขเป้าหมาย (เป็นหมายเลข IP Address ทั้งหมดที่อยู่ภายใน subnet เดียวกัน)
3. เมื่อที่หมายเลข IP Address ใดตอบ ICMP response กลับมา
4. โปรแกรมจะทำการส่ง SNMP get Request โดยร้องขอ object ที่มีหมายเลข OID 1.3.6.1.2.1.1.2.0 ซึ่งเป็นค่า sysObjectID ที่บ่งบอกถึงหมายเลข OID ที่อ้างอิงไปถึง enterprise ใน subtree (เป็นจุดเริ่มต้นของ MIB)
5. รับ SNMP get Response ที่ดึงข้อมูลมาจากอุปกรณ์ใน MIB

จากการทดลองทั้ง 2 โปรแกรมนั้นมีการทำงานที่คล้ายคลึงกัน คือ มีการกำหนดช่วง IP Address ที่ต้องการเพื่อนำไปดึงข้อมูลใน MIB และผลการทดลองที่ได้จากโปรแกรม Wireshark ทำให้เห็นถึงกระบวนการทำงานการค้นหาอุปกรณ์ โดยมีการส่งแพ็คเก็ต ARP แบบ broadcast เพื่อขอดูค่า MAC Address ของ IP Address ที่กำหนดไปในช่วง เพราะไม่มีใน ARP Table ของเครื่องเมเนเจอร์ และเมื่อได้รับค่า Mac Address ตอบกลับมาก็จะทำการ ping (ส่งแพ็คเก็ต ICMP Echo Request ออกไปตาม IP Address นั้นๆ) หากมีการใช้งานของอุปกรณ์ที่มี IP Address นั้นๆ จะมีแพ็คเก็ต ICMP Echo Response ตอบกลับมาแสดงว่ามี IP Address นั้นอยู่จริง แล้วส่งแพ็คเก็ต SNMP get-Request ที่มีปลายทางเป็นหมายเลข IP Address ที่ได้จากกระบวนการ ping เพื่อดึงข้อมูลของอุปกรณ์นั้นๆ ออกมา โดยส่งข้อมูลมาในรูปของแพ็คเก็ต SNMP get-Response กลับมายังเครื่องเมเนเจอร์แล้วแสดงผลลัพธ์ ซึ่งทั้งโปรแกรม What's up gold กับ โปรแกรม Solarwind มีการทำงานที่เหมือนกัน คือเป็นแบบวิธีที่ 1

เมื่อได้ทดลองทั้ง 2 โปรแกรมเพื่อนำมาอ้างอิงในการทำระบบ เพื่อหาเหตุผลมาเปรียบเทียบของทั้ง 2 วิธี จึงมีการทดลองเปรียบเทียบขนาดของแพ็คเก็ตของการค้นหาหมายเลข IP Addressโดยใช้ระบบ SNMP Network Management ที่มีฟังก์ชั่นการค้นหาทั้ง 2 วิธีคือใช้การ ping กับการใช้แพ็คเก็ต SNMP มาทำการทดลอง

การทดลองที่ 1 เปรียบเทียบทราฟฟิกของการค้นหาหมายเลข IP Address โดยใช้ระบบ SNMP Network Discovery ที่มีวิธีการค้นหาที่แตกต่างกันในเครือข่าย

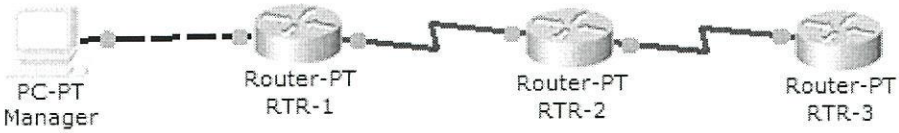
จุดประสงค์การทดลอง

- ศึกษาการเปรียบเทียบของทราฟฟิกที่เกิดขึ้นในกระบวนการการค้นหาหมายเลข IP Address ที่มีอยู่จริงโดยใช้โปรโตคอลแตกต่างระหว่างโปรโตคอล SNMP กับโปรโตคอล ICMP เพื่อศึกษาโปรโตคอลที่เหมาะสมกับระบบของโครงงาน

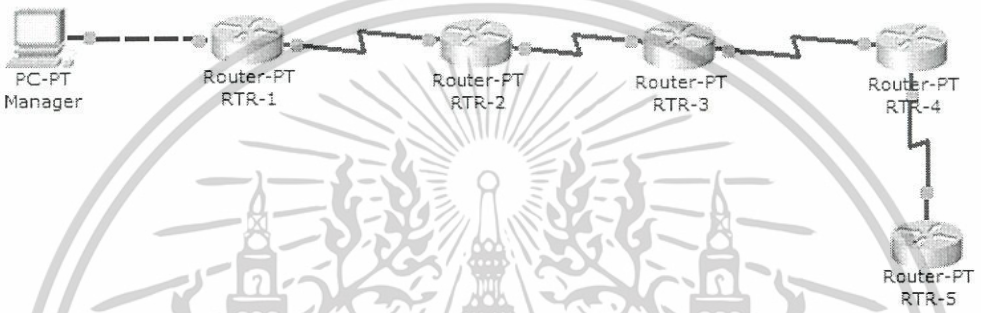
วิธีการทดลอง

การทดลองที่ 1 จะทำการเปรียบเทียบปริมาณทราฟฟิกของแต่ละวิธีโดยใช้โปรแกรมดักจับ แพ็คเก็ต ของ Wireshark ที่อินเตอร์เฟซของฝั่งเมเนเจอร์ จึงออกแบบการทดลองโดยมีจำนวนอุปกรณ์ที่แตกต่างกันเพื่อเปรียบเทียบขนาดของแพ็คเก็ตของแต่ละวิธีที่เกิดขึ้นตอนมีการเรียกใช้ฟังก์ชั่นการค้นหาหมายเลข IP Address ที่มีการใช้งานอยู่ในเครือข่าย แล้วนำผลการทดลองมาศึกษาเพื่อหาข้อสรุปในการทำฟังก์ชั่นในการค้นหาต่อไป

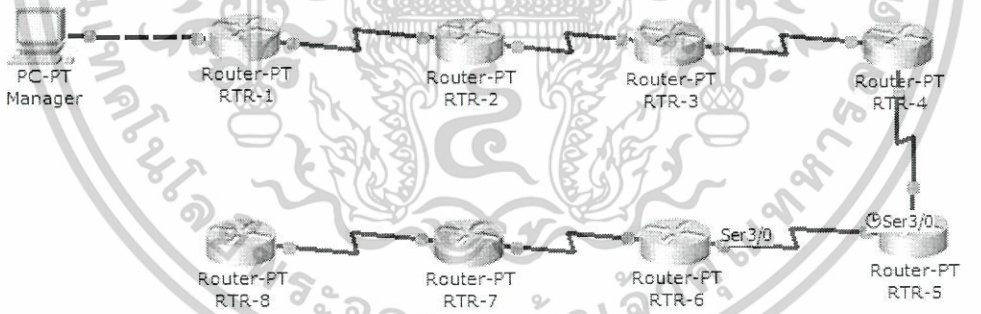
ซึ่งจำนวนอุปกรณ์(Router)ที่ทำการทดลองคือ 3,5 และ 8 เครื่องเชื่อมต่อไปยังเครื่องฝั่งเมเนเจอร์ ตามรูปที่ 4.1-4.3 จากนั้นเปิดโปรแกรม Wireshark ดักจับที่อินเตอร์เฟซ แล้วให้มีการทำงานของระบบ SNMP Network Discovery โดยมีการค้นหาอุปกรณ์ที่มีวิธีที่แตกต่างกัน คือแบบใช้โปรโตคอล ICMP กับแบบใช้โปรโตคอล SNMP ซึ่งมีการทำงานตามรูปที่ 4.4-4.5



รูปที่ 4.1 แสดงแผนภาพเครือข่ายที่มี router 3 เครื่อง

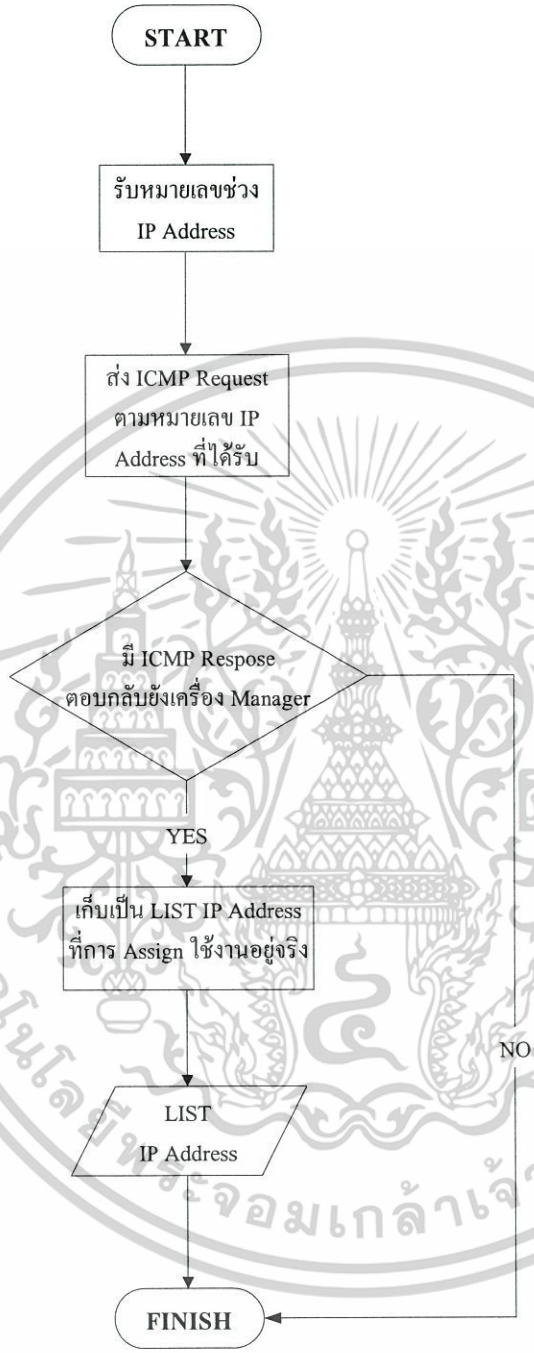


รูปที่ 4.2 แสดงแผนภาพเครือข่ายที่มี router 5 เครื่อง



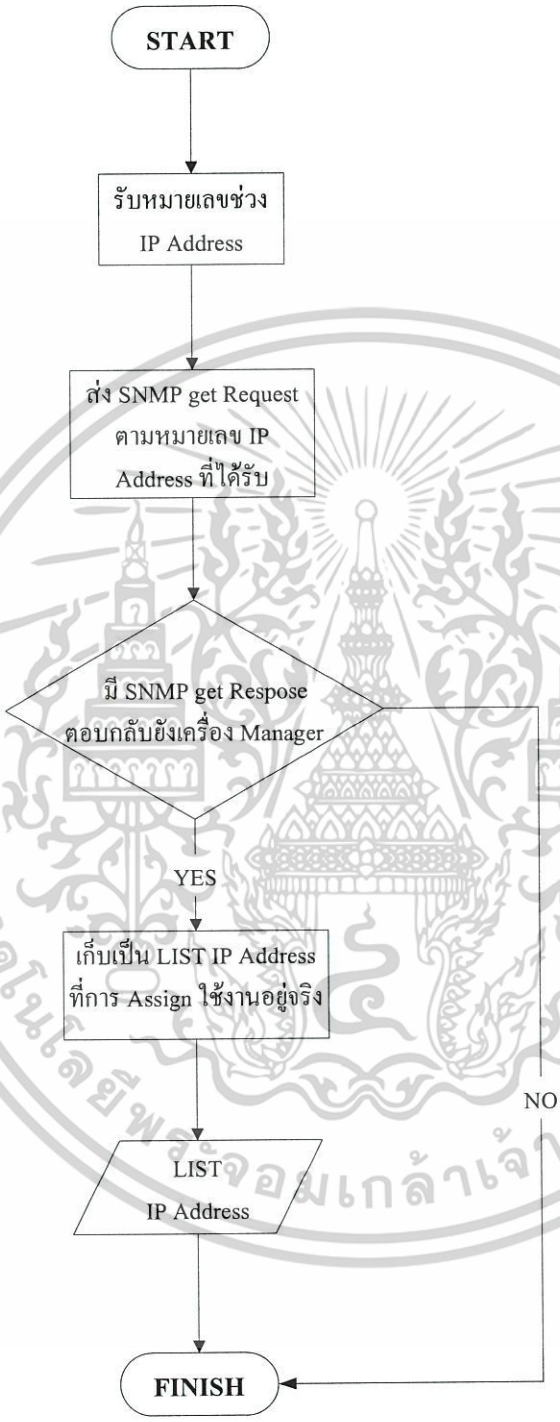
รูปที่ 4.3 แสดงแผนภาพเครือข่ายที่มี router 8 เครื่อง

ICMP DISCOVERY FUNCTION



รูปที่ 4.4 การค้นหาหมายเลข IP Address ที่มีการใช้งานโดยใช้โปรโตคอล ICMP

SNMP DISCOVERY FUNCTION



รูปที่ 4.5 การค้นหาหมายเลข IP Address ที่มีการใช้งานโดยใช้โปรโตคอล SNMP

ผลการทดลอง

ตารางที่ 4.1 ผลการทดลองเปรียบเทียบทราฟฟิกของการค้นหาหมายเลข IP Address โดยใช้ระบบ SNMP Network Discovery ที่มีวิธีการค้นหาที่แตกต่างกันในเครือข่าย

จำนวนrouter	ใช้วิธีที่ 1แบบ ICMP มีขนาดของแพ็คเก็ต (bytes)	ใช้วิธีที่ 2 แบบ SNMP มีขนาดของแพ็คเก็ต (bytes)
3	444	525
5	740	875
8	1184	1400

<u>หมายเหตุ</u> โดยผลการทดลองนี้จะรวมขนาดของแพ็คเก็ตทั้ง request และ response ในกระบวนการการค้นหาหมายเลข IP Address

สรุปผลการทดลอง

จากตารางที่ 1 จะเห็นได้ว่าทราฟฟิกที่ได้จากวิธีที่ใช้โปรโตคอล ICMP จะมีทราฟฟิกน้อยกว่าทราฟฟิกที่ได้จากวิธีที่ใช้โปรโตคอล SNMP ความแตกต่างนั้นขึ้นอยู่กับจำนวนอุปกรณ์ที่ต้องการค้นหา แต่ความแตกต่างที่เกิดขึ้นนั้นเป็นค่าที่น้อยมากทำให้ไม่สามารถระบุได้ว่าใช้โปรโตคอลแบบใดเหมาะสมที่สุด

เมื่อได้ศึกษาทฤษฎีและจากการทดลองจึงเลือกวิธีที่ใช้โปรโตคอลเอสเอ็นเอ็มพี (Simple Network Management Protocol: SNMP) มีข้อดีดังนี้

- อุปกรณ์ทั่วไปมีการทำงานของโปรโตคอลเอสเอ็นเอ็มพี (SNMP) ซึ่งเป็นพื้นฐานปกติของทุกอุปกรณ์
- ระบบของโครงงานนี้ได้อ้างอิงถึงโปรโตคอล SNMP เป็นหลัก ในการดึงข้อมูลของอุปกรณ์จึงนำมาใช้ในเพื่อทำให้มีมุมมองถึงระบบได้ชัดเจนมากยิ่งขึ้น
- ป้องกันการบล็อคพอร์ตของโปรโตคอล ICMP

การทดลองที่ 2 เปรียบเทียบทราฟฟิกของการค้นหาหมายเลข IP Address โดยใช้ระบบ SNMP Network Discovery แบบ AUTO กับแบบมีการใส่ช่วง IP Address

จุดประสงค์การทดลอง

- ศึกษาการเปรียบเทียบของทราฟฟิกที่เกิดขึ้นในกระบวนการการค้นหาหมายเลข IP Address ที่มีอยู่จริงโดยใช้ระบบค้นหาเครือข่ายโดยใช้โปรโตคอล SNMP ผ่านเว็บที่เป็นแบบกำหนดช่วงหมายเลข IP Address กับ แบบ Auto (ไม่มีการกำหนดค่าใดๆ) เพื่อศึกษาว่าถ้าระบบมีการทำงานแบบ Auto จะมีทราฟฟิกแตกต่างกับแบบมีการกำหนดช่วงหมายเลข IP Address อย่างไร และควรใช้ระบบแบบใดจึงจะเหมาะสมที่สุด

วิธีการทดลอง

การทดลองที่ 2 จะทำการเปรียบเทียบปริมาณทราฟฟิกที่เกิดจากการทำงานในฟังก์ชั่นการค้นหาที่แตกต่างกันระหว่างการค้นหาแบบอัตโนมัติกับการค้นหาที่มีการใส่ช่วงหมายเลขของ IP Address แล้วใช้โปรแกรมดักจับแพ็คเก็ตของ Wireshark ที่อินเตอร์เฟซของฝั่งเมเนเจอร์ จึงออกแบบการทดลองโดยมีจำนวนอุปกรณ์ที่แตกต่างกันเพื่อเปรียบเทียบขนาดของแพ็คเก็ตของแต่ละวิธีตามรูปที่ 4.1-4.3แล้วนำผลการทดลองมาศึกษาเพื่อหาข้อสรุปในการทำฟังก์ชั่นในการค้นหาต่อไป

กระบวนการการค้นหาหมายเลข IP Address ที่มีการใช้งานอยู่จริงใน Web-based Network Discovery System ทั้ง 2 แบบมีการทำงานตามรูปที่ 4.5 แบบใส่ช่วง IP Address และรูปที่ 4.6 แบบอัตโนมัติ

AUTO DISCOVERY FUNCTION

รูปที่ 4.6 การค้นหาหมายเลข IP Address แบบอัตโนมัติ

ผลการทดลอง

ตารางที่ 4.2 ผลการทดลองเปรียบเทียบทราฟฟิกของการค้นหาหมายเลข IP Address โดยใช้ระบบ SNMP Network Discovery แบบ AUTO กับแบบมีการใส่ช่วง IP Address

จำนวนrouter	SNMP Network Discovery แบบ AUTO มีทราฟฟิก (bytes)	SNMP Network Discovery แบบมีการใส่ช่วง IP Address มีทราฟฟิก (bytes)
3	129312	128778
5	215520	214630
8	344832	343408

หมายเหตุ โดยผลการทดลองนี้จะรวมทราฟฟิกทั้ง request และ response ในกระบวนการการค้นหาหมายเลข IP Address ซึ่งในกรณีนี้มีการกำหนดค่าให้ทุก IP Address ใน subnet มีการใช้งานอยู่จริงเพื่อศึกษากรณีที่มี IP Address มากที่สุด (ทราฟฟิกมากที่สุด)

สรุปผลการทดลอง

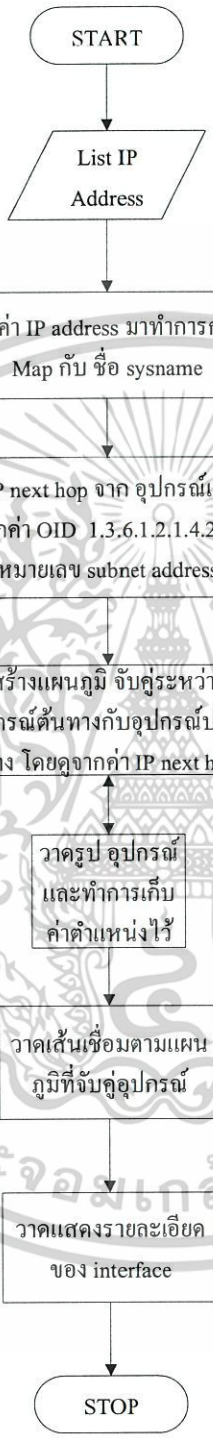
จากตารางที่ 4.2 จะพบได้ว่าทราฟฟิกระหว่าง 2 แบบเมื่อนำมาเปรียบเทียบกันแสดงให้เห็นว่ามีความแตกต่างกันน้อยมาก ปริมาณทราฟฟิกที่เกิดจะขึ้นอยู่กับจำนวนของอุปกรณ์ ความแตกต่างของทั้ง 2 แบบ เกิดขึ้นเพราะแบบมีการใส่ช่วง IP Address ไม่จำเป็นจะต้องมีการกำหนด IP Address ของ default gateway ของเครื่อง manager เพื่อนำไปหาค่าช่วง IP Address เนื่องจากจะให้ผู้ใช้งานระบบเป็นผู้กำหนดค่าช่วง IP Address เอง แต่ถ้าเป็นแบบ Auto ผู้ใช้งานระบบไม่จำเป็นจะต้องกำหนดค่า ช่วง IP Address ระบบจะดึงค่า IP Address ของ default gateway จากเครื่อง manager ดังนั้นเครื่อง manager จำเป็นจะต้องกำหนดค่า IP Address ของ default gateway ระบบจะนำค่า IP Address ของ default gateway ไปหาค่าช่วง IP Address ที่มีความเป็นไปได้

ดังนั้นได้ศึกษาจากการทดลองและสรุปข้อดี- ข้อเสียของแต่ละแบบอย่างละเอียด จึงเลือกใช้ระบบแบบ Auto จะดีกว่าแบบมีการใส่ช่วง IP Address เพราะ

- ผู้ใช้ระบบจะมีความสะดวกมากเพราะไม่จำเป็นจะต้องทราบค่าช่วง IP Address ทั้งหมดของเครือข่าย ซึ่งระบบจะทำการค้นหาและหาค่า IP Address ให้โดยอัตโนมัติ
- มีการใช้งานกับระบบทำได้ง่ายขึ้น

เมื่อเราตัดสินแล้วว่าจะทำการ Discovery แบบ auto เราจึงดำเนินการส่วนของการวาดรูปแผนภาพเครือข่ายโดยออกแบบการทำงานดังนี้

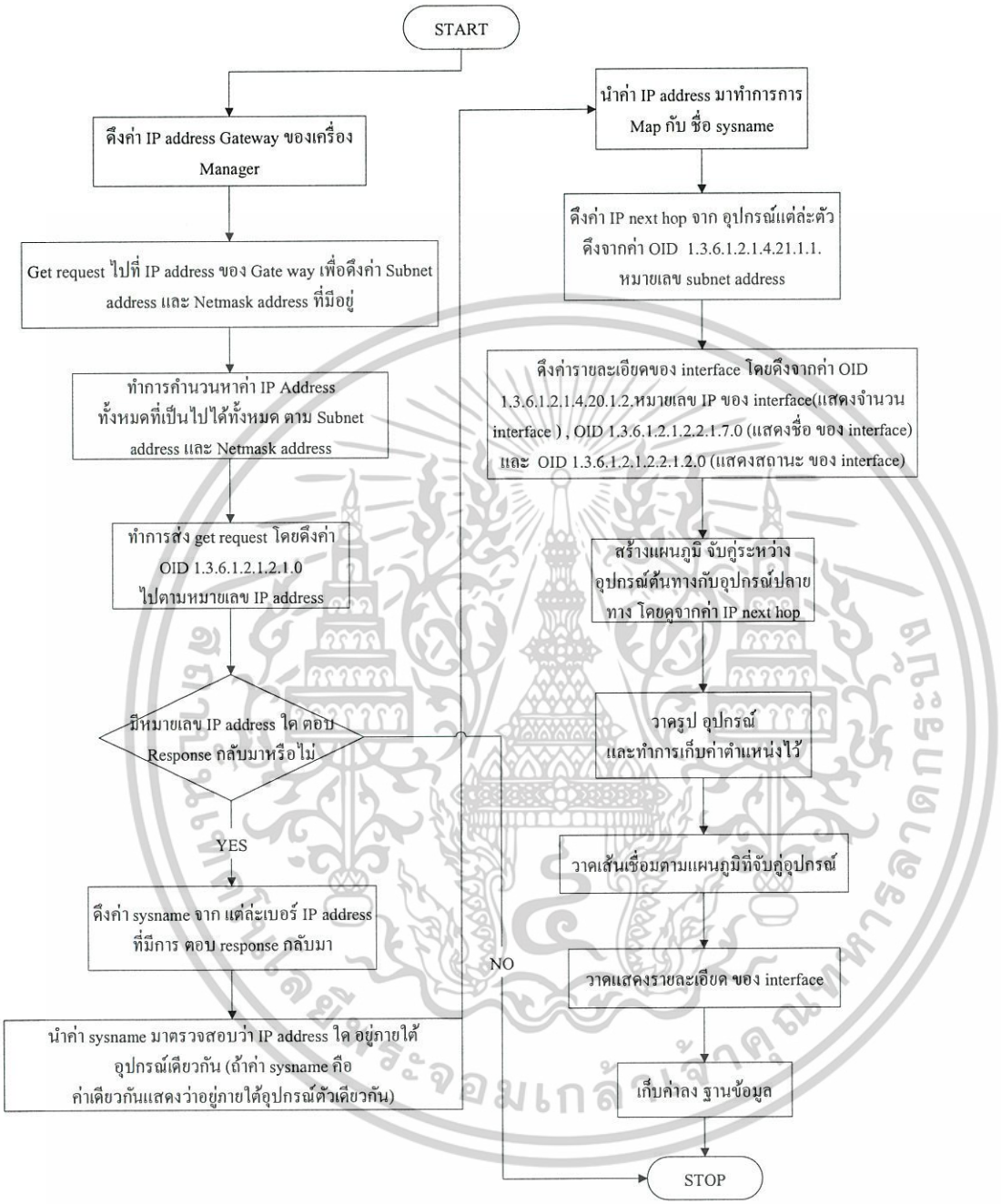
DRAWTOPOLOGYFUNTION



รูปที่ 4.7 กระบวนการวาดรูป Topology

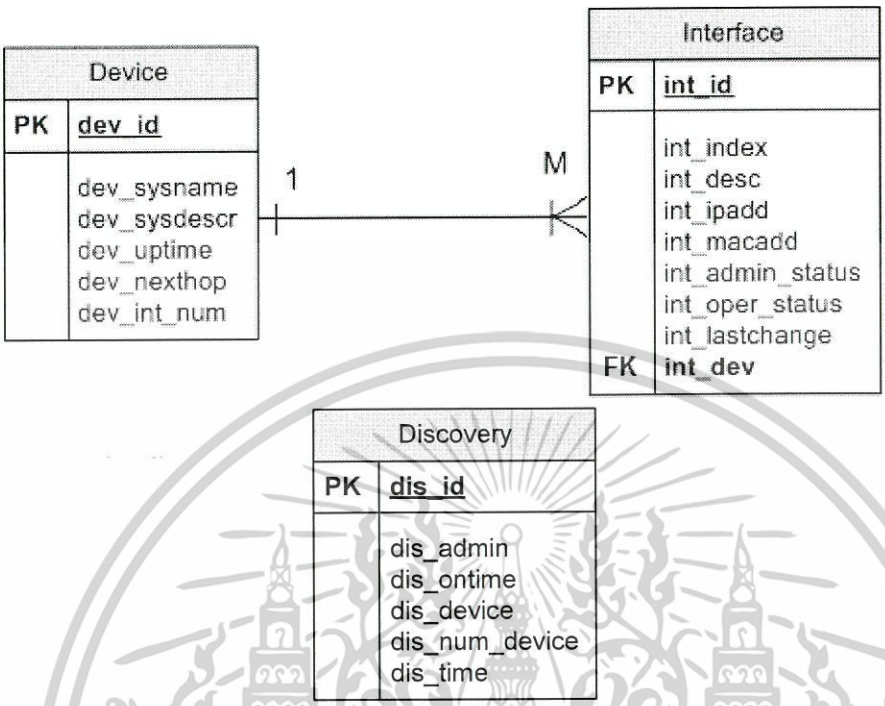
4.2 ภาพรวมของการทำงานของระบบ WEB-BASED NETWORK DISCOVERY SYSTEM

4.2.1การทำงานของระบบ



รูปที่ 4.8 กระบวนการทำงานของระบบทั้งหมด

4.2.2 ฐานข้อมูลที่ใช้ในระบบ



รูปที่ 4.9 ER Diagram ของฐานข้อมูล

Data Dictionary (พจนานุกรมข้อมูล)

ตารางอุปกรณ์ (Table Device)

ชื่อคอลัมภ์ (Colunm name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
dev_id	integer	PK	รหัสประจำตัวของ device
dev_sysname	varchar(40)		ชื่อของอุปกรณ์
dev_type	integer		ค่าตัวเลขแสดงว่าเป็นอุปกรณ์ใด
dev_sysdescr	varchar(40)		รายละเอียดของอุปกรณ์
dev_uptime	number		เวลาที่อุปกรณ์นี้ได้เปิดการทำงานล่าสุดจนถึงปัจจุบัน
dev_int_num	integer		จำนวน interface ที่มีการใช้งานอยู่
dev_nexthop	varchar(20)		หมายเลข IP Address next hop ของอุปกรณ์

ตารางที่ 4.3 ตารางอุปกรณ์ (Table Device)

ตารางอินเตอร์เฟซ (Table Interface)

ชื่อคอลัมภ์ (Colunm name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
int_id	integer	PK	รหัสของ interface
int_index	integer		ลำดับของ interface
int_desc	varchar(40)		รายละเอียดของ interaface
int_ipadd	varchar(15)		หมายเลข IP Address ที่อยู่บน interface นั้น
int_macadd	varchar(20)		หมายเลข MAC Address ที่อยู่บน interface นั้น
int_admin_status	integer		เลขแสดงสถานะ admin status ของ interface นั้น
int_oper_status	integer		เลขแสดงสถานะ operation status ของ interface นั้น
int_lastchange	number		เลขแสดงสถานะที่ interface มีการเปลี่ยนแปลง
int_dev	integer	FK (dev_id of Table device)	รหัสของ device ที่มี interface นี้อยู่

ตารางที่ 4.4 ตารางอินเตอร์เฟซ (Table Interface)

ตารางการค้นหา (Table Discovery)

ชื่อคอลัมภ์ (Colunm name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
dis_id	Integer	PK	รหัสของการ discovery
dis_admin	varchar(20)		รายชื่อ admin ที่ทำการ discovery
dis_ontime	datetime		เวลาที่ทำการ discovery
dis_time	double		เวลาที่ใช้ในการ discovery
dis_num_device	Integer		จำนวนอุปกรณ์ที่ค้นพบ

ตารางที่ 4.5 ตารางการค้นหา (Table Discovery)

บทที่ 5

การพัฒนาระบบ

5.1 ฟังก์ชั่นการทำงานของระบบ SNMP Network Discovery

5.1.1 ฟังก์ชั่นค้นหา IP Address ของ Default gateway

ออกแบบฟังก์ชั่นค้นหาหมายเลข IP Address ของ Default gateway มีเพื่อหาหมาย Subnet address ที่มีใช้งานอยู่จริง

จะเริ่มค้นหา IP Address ของ Default gateway จาก คำสั่ง ipconfig และนำค่าที่ได้ไปทำงานต่อ ซึ่งอาศัยการทำงานของ class ipconfig สิ่งที่ได้จากฟังก์ชั่นนี้คือ หมายเลข IP Address ของ gateway ที่เครื่อง manager ต่ออยู่

```
public class ipconfig {

public String[] getdefaultgateway() throws IOException {
        Process process = Runtime.getRuntime().exec("ipconfig");
        InputStream iStream = new BufferedInputStream(process.getInputStream());
        StringBuffer buffer = new StringBuffer();
        while (true) {
        int c = iStream.read();
        if (c == -1) {
        break;
        }
        buffer.append((char) c);
        }
        iStream.close();
        String[] defaultaddress = buffer.toString().substring(289, 299).split("\\.");
         return defaultaddress ;
}
```

5.1.2 ฟังก์ชั่นค้นหา subnet address

ออกแบบฟังก์ชั่นค้นหา subnet address เพื่อหาหมายเลข IP address ทั้งหมดที่เป็นไปได้

เมื่อได้รับค่าหมายเลข IP Address ของ gateway แล้วจะนำหมายเลข IP Address นี้ ส่ง SNMP frame เพื่อรับค่า get Response เป็น ค่าหมายเลข subnet ทั้งหมดที่routerตัวนี้ติดต่อไปได้ โดยส่ง frame ที่มีค่าหมายเลข OID เป็น 1.3.6.1.2.1.4.21.1.1. เบอร์ IP subnet สิ่งที่ได้จาก ฟังก์ชันนี้คือ หมายเลข subnet address ทั้งหมดที่มี

```
request = new PDU();
response = new PDU();
        request.add(new VariableBinding(new OID("1.3.6.1.2.1.4.21.1.0")));
        request.setType(PDU.GETNEXT);
        response = snmp.send(request, target).getResponse();
while(response.get(0).getOid().toString().substring(0,21).equals("1.3.6.1.2.1.4.21.1.1.")) {
        resultlist.add(response.get(0).getVariable().toString());
        request.clear();
        request.add(new VariableBinding(new OID(response.get(0).getOid().toString())));
        request.setType(PDU.GETNEXT);
        response.clear();
        response = snmp.send(request, target).getResponse();
}
```

5.1.3 ฟังก์ชั่นค้นหา IP Address ทั้งหมดที่เป็นไปได้ จาก subnet ทั้งหมด

ออกแบบฟังก์ชั่นค้นหาหมายเลข IP Address ทั้งหมดที่เป็นไปได้จาก subnet ทั้งหมด เพื่อจะได้ค้นหา IP address ทั้งหมด โดยไม่ตกหล่นเราจึงต้องค้นหาหมายเลขทั้งหมดที่เป็นไปได้

เมื่อได้รับหมายเลข subnet address จะนำค่ามาค้นหาหมายเลข IP Address ที่สามารถถูกนำไปใช้งานได้ โดยไล่ค่าหมายเลขตั้งแต่ 1 – 255 ในทุกๆ subnet สิ่งที่ได้รับจากฟังก์ชั่นนี้ คือ หมายเลข IP Address ทั้งที่เป็นไปได้ที่จะมีการนำไปใช้งาน

```
public class iprange {

        public ArrayList getiprange(ArrayList netaddress) {

        ArrayList iprange = new ArrayList();

        for (int i = 0; i < netaddress.size(); i++) {

        String[] ip4 = new String[4];

        ip4 = netaddress.get(i).toString().split("\\.");

        int y = Integer.parseInt(ip4[3].toString());

        for (int j = y; j < 15; j++) {

        iprange.add(ip4[0].toString() + "." + ip4[1].toString() + "." + ip4[2].toString()
+ "." +(j+1));
} }

        return iprange; }
}
```

5.1.4 ฟังก์ชั่นค้นหาหมายเลข IP Address ที่มีการนำไปใช้งานจริง

ออกแบบฟังก์ชั่นค้นหาหมายเลข IP Address ที่มีการนำไปใช้งานจริง เพื่อจะนำไปค้นหาอุปกรณ์ทั้งหมดที่เครือข่ายมี(layer3)

เมื่อได้หมายเลข IP Address ทั้งหมดที่อาจจะมีการนำไปใช้งานจริง แล้วจะทำการส่ง snmp frame เพื่อรอการตอบ get Response ถ้ามี get Response กลับมาจากหมายเลข IP Address นั้นแสดงว่าหมายเลข IP Address นั้นถูกนำไปใช้งาน สิ่งที่ได้จากการทำงานของฟังก์ชันนี้ คือ หมายเลข IP Address ทั้งหมดที่มีการนำไปใช้งานเป็นหมายเลข IP Address ของอินเตอร์เฟซของrouter

```
public void docheck() throws Exception {
        UdpAddress addr = new UdpAddress(UdpAddress+"/161");
        CommunityTarget target = new CommunityTarget(addr, new OctetString("kmitl"));
        target.setRetries(1);
        target.setTimeout(1500);
        target.setVersion(SnmpConstants.version2c);
        PDU pdu = new PDU();
        pdu.add(new VariableBinding(new OID("1.3.6.1.2.1.1.2.0")));
        pdu.setType(PDU.GET);
        TransportMapping transport = new DefaultUdpTransportMapping();
        transport.listen();
        try {
        Snmp snmp = new Snmp(transport);
        ResponseEvent res = snmp.send(pdu, target);
        this.status = true ;
        if (res.getError() != null) {
        throw res.getError();
} } finally {
        transport.close();
}}
```

5.1.5 ฟังก์ชั่นจับกลุ่มหมายเลข IP Address กับอุปกรณ์ router

ออกแบบฟังก์ชั่นจับกลุ่มหมายเลข IP Address กับอุปกรณ์ router เพื่อนำหมายเลข IP address ของแต่ล่ะ interface มาจัดกลุ่มตาม อุปกรณ์ เพื่อนำไปวาด topology

เมื่อได้หมายเลข IP Address ทั้งหมดที่มีการนำไปใช้งานจริง สิ่งที่ต้องการทราบต่อมาคือ อุปกรณ์ใดมีหมายเลข IP Address ของอินเตอร์เฟซใดอยู่บ้าง โดยการตรวจสอบว่ามีค่า sysName ตรงกันหรือไม่ ถ้ามีค่าตรงกันแสดงว่าอยู่ภายใต้อุปกรณ์เดียวกันถ้าไม่ตรงกันแสดงว่าไม่ได้อยู่ภายใต้อุปกรณ์เดียวกัน สิ่งที่ได้จากฟังก์ชั่นนี้คือ กลุ่มของการจับคู่ระหว่าง ชื่อrouterกับกลุ่มหมายเลข IP Address ที่อยู่ภายใต้router

```
public class groupIP {
        private Getsnmp snmp;
        public ArrayList[] dogrouping(ArrayList listip) {
        ArrayList listhostname = new ArrayList();
        ArrayList listIOS = new ArrayList();
        Set set = new HashSet();
        Iterator it = set.iterator();
        ArrayList router = new ArrayList();
        for (int i = 0; i < listip.size(); i++) {
        snmp = new Getsnmp(listip.get(i).toString(), "161", "kmitl",
"1.3.6.1.2.1.1.5.0");
        listhostname.add(snmp.getValue());
}
        for (int i = 0; i < listhostname.size(); i++) {
        set.add(listhostname.get(i).toString());
}
        it = set.iterator();
        for (int i = 0; i < set.size(); i++) {
        router.add(it.next().toString());
}
        ArrayList routerlist[] = new ArrayList[router.size()];
        for (int i = 0; i < router.size(); i++) {
        }       }
```

5.1.6 ฟังก์ชั่นค้นหา Next hop ของ router

ออกแบบฟังก์ชั่นค้นหา next hop ของ router เพื่อนำไปวาด link เชื่อมต่อระหว่างอุปกรณ์

เมื่อได้กลุ่มของการจับคู่ระหว่าง ชื่อrouterกับกลุ่มหมายเลข IP Address ที่อยู่ภายใต้router จำเป็นจะต้องการทราบว่าอุปกรณ์ใดต่ออยู่กลับอุปกรณ์ใด จึงต้องดึงค่า Next hop จากrouterโดยดึงค่าจาก MIB โดยใช้หมาเลย OID เป็น 1.3.6.1.2.1.4.21.1.7.subnet ที่จะคิดต่อไป สิ่งที่ได้จาก ฟังก์ชั่นนี้คือหมายเลข IP Address Next hop ของแต่ละrouter

```
request = new PDU();
        response = new PDU();
        requestIP = new PDU();
        responseIP = new PDU();
        for (int i = 0; i < netadd.size(); i++) {
        request.add(new VariableBinding(new OID("1.3.6.1.2.1.4.21.1.8." + netadd.get(i).toString())))
        request.setType(PDU.GET);
        response = snmp.send(request, target).getResponse();
        if (response.get(0).getVariable().toString().equals("4")) {
        requestIP.add(new VariableBinding(new OID("1.3.6.1.2.1.4.21.1.7." + netadd.get(i).toString())));
        requestIP.setType(PDU.GET);
```

5.1.7 ฟังก์ชั่นวาดรูปแผนภาพเครือข่าย

ออกแบบฟังก์ชั่นวาดรูปแผนภาพเครือข่ายหรือ Topology เพื่อแสดงรูป Topology ทั้งหมด และรายละเอียดของแต่ล่ะ interface

เมื่อได้หมายเลข IP Address Next hop ของแต่ล่ะ routerแล้ว สิ่งต่อมาคือจะนำค่าที่ได้รับมาวาดเป็นรูปแผนโดยตรวจสอบว่าหมายเลข IP Address นี้อยู่ภายใต้อุปกรณ์ใด สิ่งที่ได้จากฟังก์ชั่นนี้ คือ รูปแผนภาพเครือข่าย

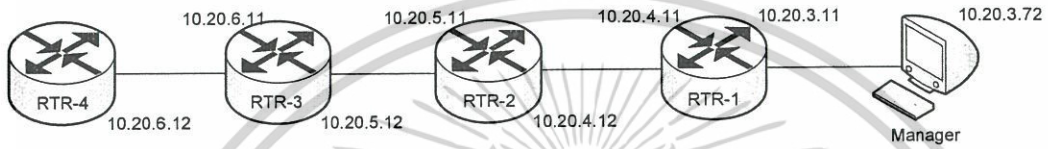
```
    Toolkit tk = Toolkit.getDefaultToolkit();
     routerpic = tk.getImage(" /image/router.png");
     pcpic = tk.getImage("/image/pc.png");
     Map showmap = new HashMap();
     for (int i = 0; i < routerlist.size(); i++) {
        ArrayList target = new ArrayList();
        target.add(100 * (i + 1));
        target.add(50);
        showmap.put(routerlist.get(i).toString(), target);
     }
         for (int i = 0; i < map.size(); i++) {
         ArrayList source = new ArrayList();
        ArrayList dest = new ArrayList();
        source = (ArrayList) showmap.get(routerlist.get(i).toString());
        dest = (ArrayList) showmap.get(map.get(routerlist.get(i).toString()).toString());
        x1 = Integer.parseInt(source.get(0).toString());
        y1 = Integer.parseInt(source.get(1).toString());
}
```

บทที่ 6

การทดลอง

6.1 การทดลองที่ 1

Topology 1



รูปที่ 6.1 แผนภาพเครือข่ายการทดลองที่ 1

6.1.1 วิธีการทดลอง

กำหนดทุกrouterconfigค่าต่างๆดังนี้

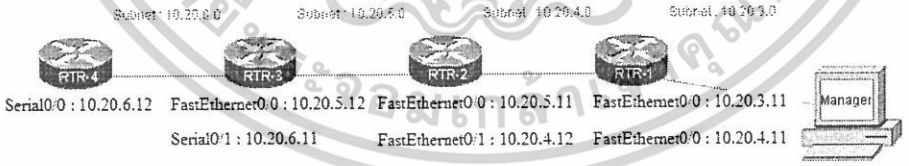
router rip version 2

network 10.20.0.0

snmp-server community string public

6.1.2 ผลการทดลองที่ 1

Topology



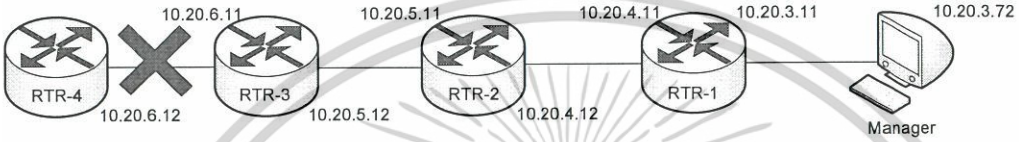
Discovery on Time : Fri Mar 20 11:55:58 ICT 2009

รูปที่ 6.2 ผลการทดลองที่1

จากรูปที่ 6.2 แสดง ผลลัพธ์ระบบตามการทดลองที่ 1 แสดงการเชื่อมต่อจาก RTR-1 ไปยัง RTR-2 , RTR-2 ไปยัง RTR-3 , RTR-3 ไปยัง RTR-4 (เป็นไปตาม Topology ที่มีการเปลี่ยนแปลง) และแสดงรายละเอียดของแต่ละ interface ทั้ง ชื่อ , IP Address และ Subnet Address

6.2 การทดลองที่ 2

Topology 2



รูปที่ 6.3 แผนภาพเครือข่ายการทดลองที่ 2

6.2.1 วิธีการทดลอง

กำหนดทุก router config ค่าต่างๆดังนี้

router rip version2

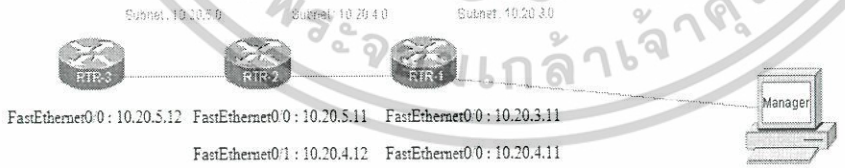
network 10.20.0.0

snmp-server community string public

ทำการใช้คำสั่ง shutdown ที่ interface หมายเลข 10.20.6.11

6.2.2 ผลการทดลองที่ 2

Topology



Discovery on Time : Fri Mar 20 11:57:17 ICT 2009

รูปที่ 6.4 ผลการทดลองที่ 2

จากรูปที่ 6.4 แสดงผลลัพธ์ระบบตามการทดลองที่ 2 แสดงการเชื่อมต่อจาก RTR-1 ไปยัง RTR-2 ,RTR-2 ไปยัง RTR-3 แต่จะไม่มี RTR-3 ไปยัง RTR-4 (เป็นไปตาม Topology ที่มีการเปลี่ยนแปลง) และแสดงรายละเอียดของแต่ล่ะ interface ทั้ง ชื่อ , ip address และ subnet address

บทที่ 7
สรุปการทำงานของระบบ

7.1 ผลจากการดำเนินการ

จากการพัฒนาระบบค้นหาเครือข่ายโดยใช้โปรโตคอล SNMP ผ่านเว็บเพื่อช่วยสนับสนุนการทำงานของผู้ดูแลระบบในเครือข่ายโดยผ่านเว็บ แสดงแผนภาพเครือข่ายและรายละเอียดสำคัญที่เกี่ยวข้องกับอุปกรณ์ให้ผู้ดูแลระบบเพื่อเป็นประโยชน์ในการตรวจสอบเครือข่าย

เมื่อระบบค้นหาเครือข่ายโดยใช้โปรโตคอล SNMP ผ่านเว็บและได้ผลลัพธ์จากการทำงานจะรวมเอาความสามารถของระบบค้นหาเครือข่ายโดยใช้โปรโตคอล SNMP ผ่านเว็บเหล่านี้มาทำงานร่วมกัน

ความสามารถของระบบค้นหาเครือข่ายโดยใช้โปรโตคอล SNMP ผ่านเว็บ

ฟังก์ชั่นการทำงาน SNMP Network Discovery

- ฟังก์ชั่นค้นหาหมายเลข IP Address ของ Default gateway
- ฟังก์ชั่นค้นหา subnet address
- ฟังก์ชั่นค้นหาหมายเลข IP Address ทั้งหมดที่เป็นไปได้จาก subnet ทั้งหมด
- ฟังก์ชั่นค้นหาหมายเลข IP Address ที่มีการนำไปใช้งานจริง
- ฟังก์ชั่นจับกลุ่มหมายเลข IP Address กับอุปกรณ์ router
- ฟังก์ชั่นค้นหา next hop ของ router
- ฟังก์ชั่นวาดรูปแผนภาพเครือข่ายหรือ Topology

7.2 ประโยชน์ที่ได้จากระบบ

- ผู้ดูแลสามารถทราบแผนภาพเครือข่ายแบบอัตโนมัติ เพื่อช่วยสนับสนุนการทำงาน
- ประหยัดเอกสารของแผนภาพเครือข่ายหรือเอกสารอ้างอิงหมายเลข IP Address ของเครือข่าย
- ประหยัดเวลาในการทำงานของผู้ดูแลระบบ
- สามารถให้ผลลัพธ์โดยไม่ต้องมีการกำหนดค่าใดๆให้กับระบบ ทำให้ผู้ใช้ไม่ต้องทราบข้อมูลใดๆของเครือข่ายมาก่อนที่จะใช้ระบบนี้
- สามารถแสดงสถานะของอินเตอร์เฟซในแต่ละอุปกรณ์ของเครือข่าย
- ระบบสามารถทำงานผ่านเว็บ ซึ่งหมายความว่าผู้ดูแลสามารถทำงานได้ทุกที่ มีความสะดวกสบายในการทำงาน
- ผู้ดูแลระบบสามารถเรียนรู้และเข้าใจเครือข่ายได้ชัดเจนยิ่งขึ้น

7.3 ข้อจำกัดของระบบ

1. เครื่องเมเนเจอร์จำเป็นต้องทำการกำหนดค่า default gateway เนื่องจากระบบมีการค้นหาแบบอัตโนมัติ
2. สามารถทำการค้นหาอุปกรณ์ที่มีการตั้งค่า IP Address เท่านั้นหรืออุปกรณ์ที่ทำงานบนเลเยอร์ 3 ของ OSI Model
3. ทุกอุปกรณ์ต้องกำหนดค่า Community Name เป็น Public

7.4 แนวทางในการดำเนินงานในอนาคต

เมื่อโครงงานนี้เสร็จสมบูรณ์แล้ว สามารถนำมาพัฒนาในการทำงานอื่นๆ ได้มากมาย เช่น เพิ่มความสามารถในการตั้งค่าอุปกรณ์เครือข่ายผ่านระบบได้,ค้นหาอุปกรณ์เลเยอร์ 2 ได้ รวมถึงการควบคุมปริมาณข้อมูลในเครือข่าย และการปิดและเปิดอุปกรณ์เครือข่าย อัตโนมัติตามช่วงเวลาที่ใช้งานได้และสามารถที่จะเพิ่มหมายเลข OID ในระบบเพื่อไปดึงข้อมูลนอกเหนือจากที่ระบบนี้สามารถทำได้จาก MIB ของอุปกรณ์เครือข่ายได้ อีกทั้งยังสามารถเก็บข้อมูลเป็นฐานข้อมูลเพื่อให้ปัญญาประดิษฐ์คิดวิเคราะห์ได้ว่าจะต้องปรับปรุงประสิทธิภาพด้านใดของเครือข่าย

บรรณานุกรม

Mellquist, Peter Erik, **SNMP++ : an object-oriented approach to developing network management applications** .Peter Erik Mellquist, Upper Saddle River, NJ : Prentice Hall PTR, 1998

Simoneau, Paul, **SNMP network management** . Paul Simoneau, New York : McGraw-Hill, 1999

Stallings, William, **SNMP, SNMPv2, SNMPv3, and RMON 1 and 2** .William Stallings, Reading, Mass. : Addison-Wesley, 1999

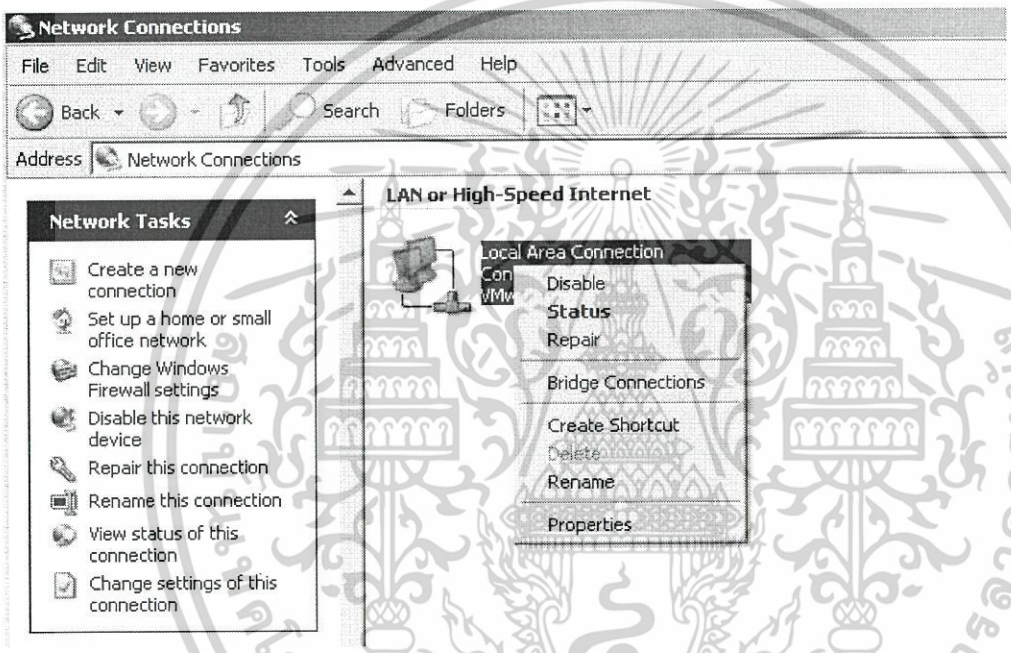
ภาคผนวก

คู่มือการติดตั้งระบบ

ระบบค้นหาเครือข่ายโดยใช้โปรโตคอล SNMP ผ่านเว็บ (WEB BASED NETWORK DISCOVERY SYSTEM)

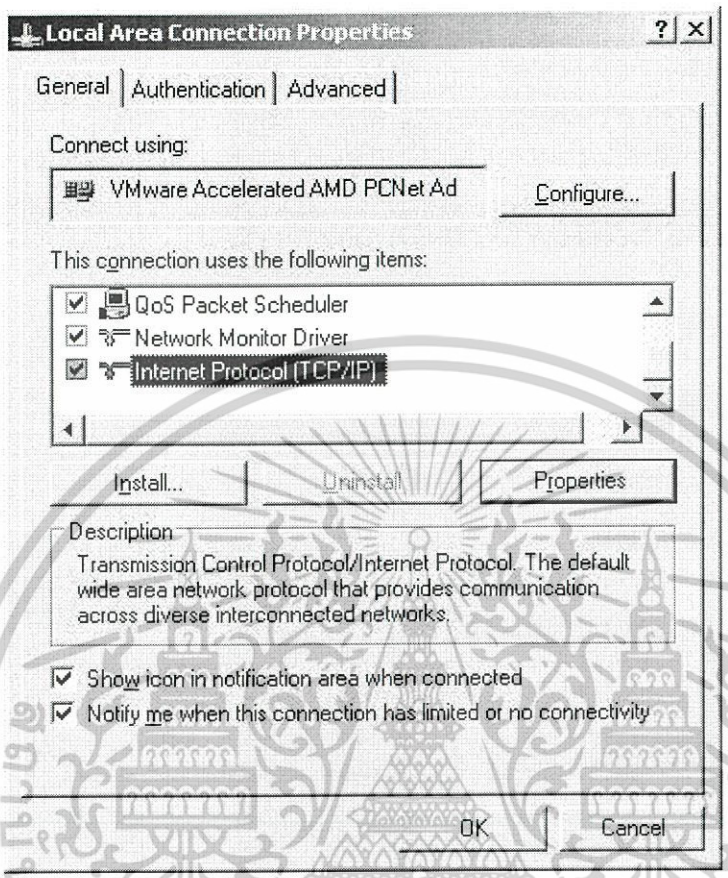
- เครื่อง Manager

1. เข้าไปที่ My Network Places > เลือก view network connection > คลิกขวาที่ Local Area Connection > เลือก properties



รูปภาคผนวก-1 แสดงการติดตั้งระบบขั้นตอนที่ 1

2. เลือก internet Protocol(TCP/IP) กดปุ่ม Properties



รูปภาคผนวก-2 แสดงการติดตั้งระบบขั้นตอนที่ 2

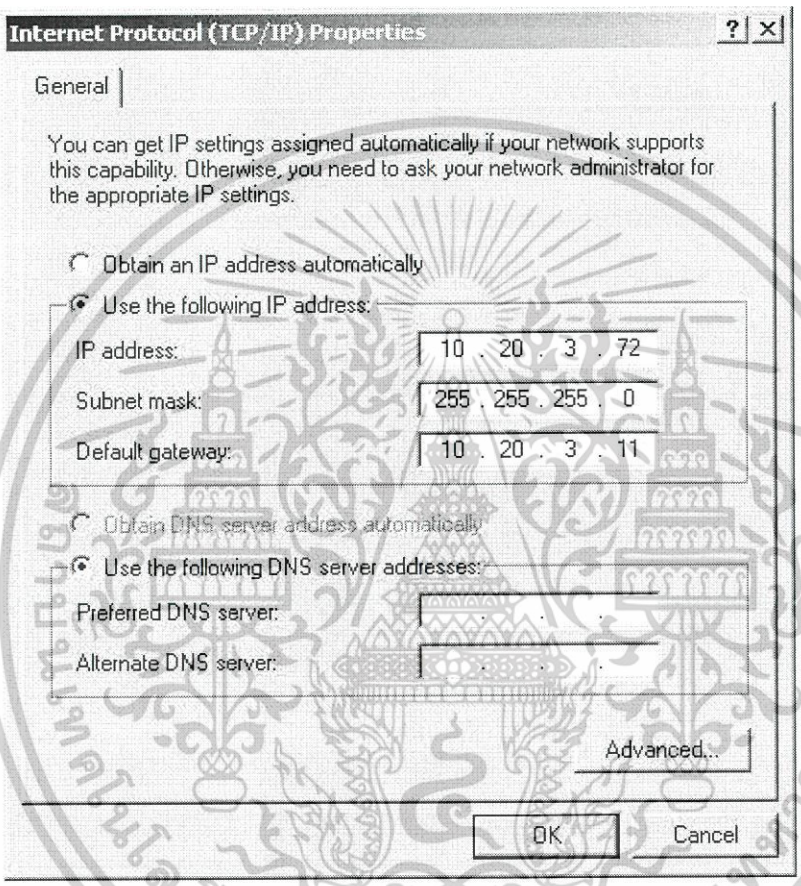
3. ตัวอย่างการกำหนดค่าต่างๆ ดังนี้

IP address: 10.20.3.72

Subnet mask: 255.255.255.0

Default gateway: 10.20.3.11

กดปุ่ม OK

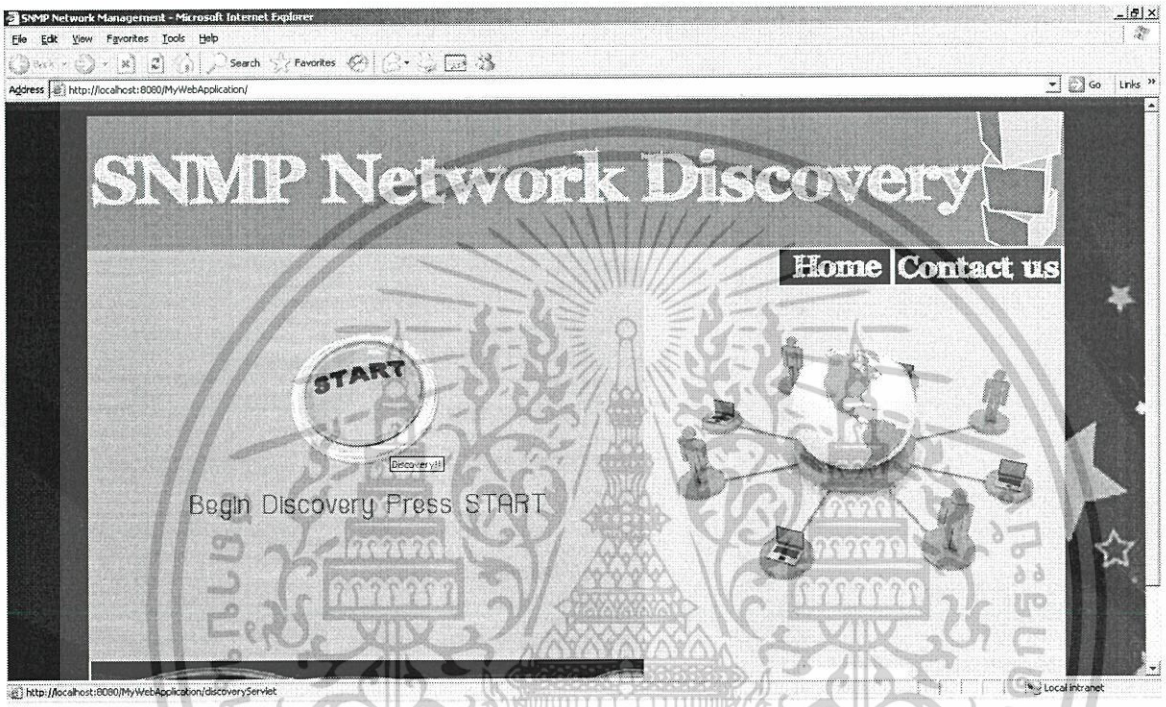


รูปภาคผนวก-3 แสดงการติดตั้งระบบขั้นตอนที่ 3

<u>หมายเหตุ</u> หมายเลข IP Address ของ Default gateway จะเป็นหมายเลข IP Address ของอุปกรณ์ที่ต่อเชื่อมกับเครื่องเมเนเจอร์โดยตรง

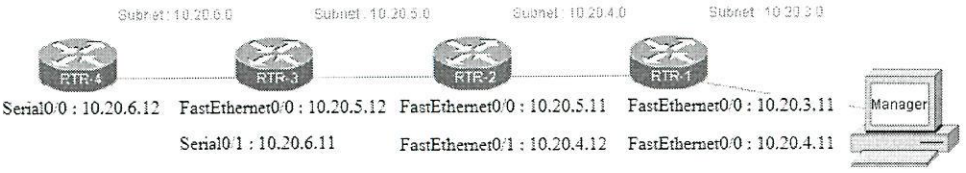
คู่มือการใช้งานระบบ

เมื่อผู้ใช้งานเข้ามาที่หน้าแรกของระบบ จะพบว่าในหน้าเว็บจะมีปุ่มกดเพียงหนึ่งปุ่มเนื่องจากระบบจะทำการค้นหาเครือข่ายแบบอัตโนมัติ ตามรูปภาคผนวก-1



รูปภาคผนวก-4 แสดงหน้าจอเริ่มต้นของระบบ

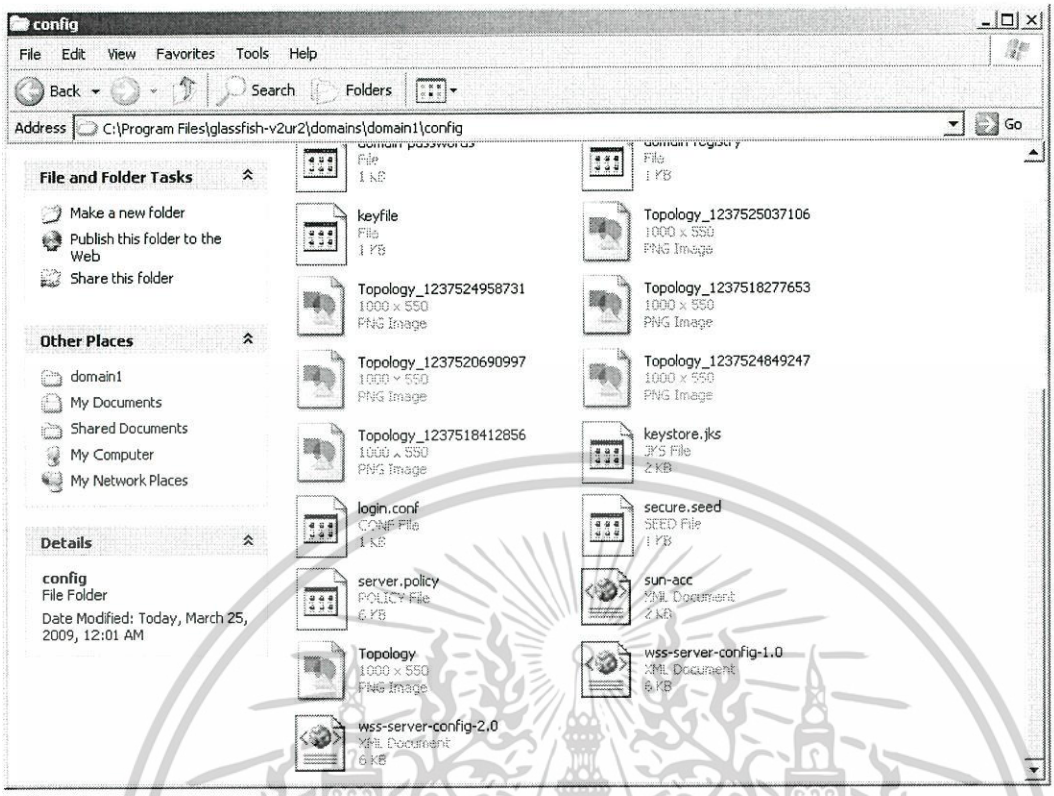
ผลลัพธ์จากการสั่งงานจากหน้าแรก ระบบจะทำการประมวลผลแล้วแสดงผลลัพธ์ออกมาในรูปของแผนภาพเครือข่าย (topology) และแสดงสถานะของอินเตอร์เฟซในแต่ละอุปกรณ์เพื่อช่วยให้ผู้ดูแลระบบสามารถทราบได้ว่าสถานะของแต่ละอินเตอร์เฟซเป็นอย่างไร



รูปภาคผนวก-5 แสดงหน้าจอผลลัพธ์ของระบบครั้งที่ 1

ผลลัพธ์เมื่อลองตั้งค่าให้สถานะของอินเตอร์เฟซใน router (RTR-3) ให้มีสถานะ down เพื่อตรวจสอบผลลัพธ์ที่ถูกต้อง ซึ่งได้ผลลัพธ์ได้ถูกต้อง

รูปภาคผนวก-6 แสดงหน้าจอผลลัพธ์ของระบบครั้งที่ 2



รูปภาคผนวก-7 แสดงสถานที่เก็บรูป Topology ที่ได้จากระบบ

หมายเหตุ ตำแหน่งที่เก็บอยู่ที่ “C:\Program Files\glassfish-v2ur2\domains\domain1\config” ผู้ใช้สามารถนำรูป Topology ไปใช้งานได้ โดยมีเวลากำกับอยู่ในรูปเพื่อให้ทราบว่ามีการวาด Topology เมื่อใด

การพัฒนาระบบ

index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>SNMP Network Management</title>
<style type="text/css">
</style>
<script type="text/javascript">
<!--
function MM_swapImgRestore() { //v3.0
var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++)
x.src=x.oSrc;
}
function MM_preloadImages() { //v3.0
var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}
var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
}
```

การพัฒนาระบบ(ต่อ)

ImageServlet.java

```
import java.io.*;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import javax.servlet.http.*;
import javax.servlet.*;

public class ImageServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
      try {
        ArrayList routerlist = (ArrayList) request.getSession().getAttribute("routerlist");
        Map map = (Map) request.getSession().getAttribute("map");
        ArrayList liststatus = (ArrayList) request.getSession().getAttribute("liststatus");
        Date date =  (Date) request.getSession().getAttribute("date");
        ArrayList subnet =  (ArrayList) request.getSession().getAttribute("subnet");
        drawTopology imageProducer = new drawTopology();
        imageProducer.drawTopology(routerlist, map,liststatus,date,subnet);

        String type = imageProducer.createImage(response.getOutputStream());
        response.setContentType(type);
         } catch (Exception e) {
        throw new ServletException(e);
             }
           }
```

การพัฒนาระบบ(ต่อ)

drawTopology.java

```
import java.awt.image.ImageConsumer;
public class drawTopology implements ImageProducer {
   public void drawTopology(ArrayList routerlist, Map map, ArrayList liststatus, Date date,
ArrayList subnet) throws Exception {
      image = f.createImage(1000, 550);
      graphics = image.getGraphics();
      f.setVisible(false);
      Toolkit tk = Toolkit.getDefaultToolkit();
      routerpic = tk.getImage("C:/Documents and
Settings/Support/Desktop/MyWebApplication/build/web/images/router.png");
      pcpic = tk.getImage("C:/Documents and
Settings/Support/Desktop/MyWebApplication/build/web/images/pc.png");
      Map showmap = new HashMap();
      int countx = 0, county = 0;
         graphics.setColor(Color.GREEN);
         graphics.drawLine((x1 + 35), (y1 + 35), (x2 + 35), (y2 + 35));
      }
         for (int j = 0; j < (list1.size()); j++) {
            if (list1.get(j).equals(1)) {
               if (!(list3.get(j).equals("No ip"))) {
                  graphics.setColor(Color.GREEN);
      }
```

การพัฒนาระบบ(ต่อ)

Getsnmp.java

```
package javabean;
import org.snmp4j.smi.*;
public class Getsnmp{

      CommunityTarget target = new CommunityTarget();
       target.setCommunity(new OctetString(this.community));
       Address targetAddress = GenericAddress.parse("udp:" + this.targetIpAddress +
"/" + this.targetPort);
       target.setAddress(targetAddress);
       target.setRetries(2);
       target.setTimeout(1500);
       target.setVersion(SnmpConstants.version2c);
       ResponseEvent response = null;
       try {
          TransportMapping transport = new DefaultUdpTransportMapping();
          transport.listen();
          Snmp snmp = new Snmp(transport);
          response = snmp.send(pdu, target);
          return response.getResponse().get(0).getVariable().toString();
```

การพัฒนาระบบ(ต่อ)

ifconfig.java

```
package javabean;
import java.io.*
public class ipconfig {
   public String[] getdefaultgateway() throws IOException {
      Process process = Runtime.getRuntime().exec("ipconfig");
      InputStream iStream = new BufferedInputStream(process.getInputStream());
      StringBuffer buffer = new StringBuffer();
      while (true) {
         int c = iStream.read();
         if (c == -1) {
            break;
         }
         buffer.append((char) c);
      }
      iStream.close();
      String[] defaultaddress = buffer.toString().substring(289, 299).split("\\.");
      return defaultaddress ;
   }
   public String[] getiphost() throws Exception {
      InetAddress localhost = null;
      localhost = InetAddress.getLocalHost();
      String[] ipaddress = localhost.getHostAddress().split("\\.");
      return ipaddress;
   }
}
```

การพัฒนาระบบ(ต่อ)

groupIP.java

```
package javabean;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;

public class groupIP {
   private Getsnmp snmp;
   public ArrayList[] dogrouping(ArrayList listip) {
      ArrayList listhostname = new ArrayList();
      ArrayList listIOS = new ArrayList();
      Set set = new HashSet();
      Iterator it = set.iterator();
      ArrayList router = new ArrayList();
      for (int i = 0; i < listip.size(); i++) {
      }
      return routerlist;
   }
}
```

f

การพัฒนาระบบ(ต่อ)

snmpcheck.java

```
package javabean;import java.util.logging.Level;
import java.util.logging.Logger;
import org.snmp4j.*;
   public void docheck() throws Exception {
      CommunityTarget target = new CommunityTarget(addr, new
OctetString("public"));
      target.setTimeout(1500);
      target.setVersion(SnmpConstants.version2c);
      PDU pdu = new PDU();
      pdu.add(new VariableBinding(new OID("1.3.6.1.2.1.2.1.0")));
      pdu.setType(PDU.GET);
      TransportMapping transport = new DefaultUdpTransportMapping();
      transport.listen();
      try {
         Snmp snmp = new Snmp(transport);
         ResponseEvent res = snmp.send(pdu, target);
         this.status = true ;

   }
}
```

การพัฒนาระบบ(ต่อ)

snmpmap.java

```
package javabean;
import java.util.ArrayList;
public class snmpmap {
   public String lookupfindname(ArrayList[] devicelist, String iptarget) {
      String target = null;
      for (int d = 0; d < devicelist.length; d++) {
         for (int i = 0; i < devicelist[d].size(); i++) {
            if (devicelist[d].get(i).toString().equals(iptarget)) {
               target = devicelist[d].get(0).toString();
            }
         }
      }
      return target;
   }
   public ArrayList domap(ArrayList[] devicelist, String namesource, String
namedest) {
      ArrayList list = new ArrayList();
      list.add(lookupfindname(devicelist, namesource));
      list.add(lookupfindname(devicelist, namedest));
      return list;
   }
}
```

การพัฒนาระบบ(ต่อ)

snmpnexthop.java

```
package javabean;
import org.snmp4j.*;
public class snmpnexthop {
      for (int i = 0; i < netadd.size(); i++) {
         request.add(new VariableBinding(new OID("1.3.6.1.2.1.4.21.1.8." +
netadd.get(i).toString())));
         request.setType(PDU.GET);
         response = snmp.send(request, target).getResponse();
         if (response.get(0).getVariable().toString().equals("4")) {
            requestIP.add(new VariableBinding(new OID("1.3.6.1.2.1.4.21.1.7." +
netadd.get(i).toString())));
            requestIP.setType(PDU.GET);
            responseIP = snmp.send(requestIP, target).getResponse();
            resultlist.add(responseIP.get(0).getVariable().toString());
            requestIP.clear();
            responseIP.clear();
         }
         request.clear();
         response.clear();
      }
      return resultlist;
   }
}
```

การพัฒนาระบบ(ต่อ)

snmpstatusport.java

```
package javabean;
import java.io.IOException;
import java.util.ArrayList;
import org.snmp4j.*
      respdu_intdes = new PDU();
      reqpdu_intdes.add(new VariableBinding(new OID("1.3.6.1.2.1.2.1.0")));
      reqpdu_intdes.add(new VariableBinding(new OID("1.3.6.1.2.1.2.2.1.2.0")));
      respdu_intdes = snmp.send(reqpdu_intdes, target).getResponse();
      for (int i = 0; i < numberofinterface; i++) {
         reqpdu_intdes.add(new VariableBinding(new
OID(respdu_intdes.get(0).getOid())));
         reqpdu_intdes.setType(PDU.GETNEXT);
         respdu_intdes = snmp.send(reqpdu_intdes, target).getResponse();
         list_interface_status[1].add(respdu_intdes.get(0).getVariable().toString());
      }
      for (int i = 0; i < numberofinterface; i++) {
         list_interface_status[2].add("No ip");
      }
      for (int j = 1; j < list[rank].size(); j++) {
         list_interface_status[2].set((dogetipinterface(list[rank].get(j).toString()) - 1),
list[rank].get(j).toString());
      }
      return list_interface_status;
   }
}
```

การพัฒนาระบบ(ต่อ)

snmpwalker.java

```
package javabean;
import java.io.IOException;
import java.util.ArrayList;
import org.snmp4j.*;
public class snmpwalker {
   private static Snmp snmp;
   private static PDU request , response;
   public ArrayList dowalk(String targetdevice) throws IOException {
      request.add(new VariableBinding(new OID("1.3.6.1.2.1.4.21.1.0")));
      request.setType(PDU.GETNEXT);
      response = snmp.send(request, target).getResponse();
      while (response.get(0).getOid().toString().substring(0,
21).equals("1.3.6.1.2.1.4.21.1.1.")) {
         resultlist.add(response.get(0).getVariable().toString());
         request.clear();
         request.add(new VariableBinding(new
OID(response.get(0).getOid().toString())));
         request.setType(PDU.GETNEXT);
         response.clear();
         response = snmp.send(request, target).getResponse();
      }
      return resultlist;
   }
}
```

ประวัติผู้จัดทำ

ชื่อ-นามสกุล	นายณัฐวิชช์ ว่องสิทธิโรจน์
วัน เดือน ปีเกิด	4 สิงหาคม 2530
ที่อยู่	69 หมู่ 19 ต. ทุ่งลูกนก อ. กำแพงแสน จ. นครปฐม
ประวัติการศึกษา	2551 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อ-นามสกุล	นางสาวสุรีย์นารถ เกียรติสาโรจน์
วัน เดือน ปีเกิด	25 กันยายน 2528
ที่อยู่	108 แขวง สำเหร่ อ. ธนบุรี จ.กรุงเทพ 10600
ประวัติการศึกษา	2551 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง