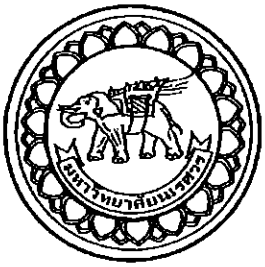


โปรแกรมจำลองการทำงานเครือข่ายของโพรโทคอลระดับ Application Layer

A PROGRAM FOR SIMULATING AN APPLICATION LAYER PROTOCOL

นายชานนท์ อ่วมเพชร รหัส 50365031

นายทวีสิน ลาภานุวัฒน์ รหัส 50365079

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 19 มิ.ย. 2555
เลขทะเบียน 15756972
เลขเรียกหนังสือ วิ
มหาวิทยาลัยนเรศวร ต 521 ป 2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2553

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน โปรแกรมจำลองการทำงานเครือข่ายของโพรโทคอลระดับ Application Layer

ผู้ดำเนินโครงงาน นายชานนท์ อ่วมเพชร รหัส 50365031

นายทวีสิน ลาภานุวัฒน์ รหัส 50365079

อาจารย์ที่ปรึกษา ดร.สุรเดช จิตประไพกุลศาล

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2553

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....ที่ปรึกษาโครงการ

(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ

(อาจารย์สิรภพ คชรัตน์)

.....กรรมการ

(อาจารย์เศรษฐา ตั้งค้าวานิช)

ชื่อหัวข้อโครงงาน	โปรแกรมจำลองการทำงานเครือข่ายของโพรโทคอลระดับ Application Layer
ผู้ดำเนินโครงงาน	นายชานนท์ อ่วมเพชร รหัส 50365031
	นายทวีสิน ลาภานุวัฒน์ รหัส 50365079
อาจารย์ที่ปรึกษา	ดร.สุรเดช จิตประไพกุลศาล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

บทคัดย่อ

โครงงานนี้ได้พัฒนาไลบรารีภาษา Java สำหรับจำลองการทำงานของเครือข่ายโพรโทคอลแบบ TCP /IP Model ในระดับชั้น Application Layer เพื่อใช้แสดงการทำงานของระบบเครือข่าย โดยเป็นการพัฒนาต่อยอดจากไลบรารีเดิมที่แสดงการทำงานในระดับชั้น Physical Layer, Data Link Layer, Internet Layer และ Transport Layer โพรโทคอลที่เลือกจำลองในโครงงานนี้คือ โพรโทคอล Simple Network Management Protocol (SNMP) อันเป็นโพรโทคอลที่ใช้กันอย่างแพร่หลาย เพื่อจัดการอุปกรณ์ภายในระบบเครือข่าย

Project Title	A program for simulating an Application Layer protocol		
Name	Mr.Chanon	Aumphet	ID. 50365031
	Mr.Taweesin	Lapanuwat	ID. 50365079
Project Advisor	Dr.Suradet Jitprapaikulsarn		
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2010		

Abstract

In this project, we developed a Java library for simulating a protocol in the Application Layer for the TCP/IP model. This library is an enhancement of an existing library which was originally developed for simulating 4 layers, including Physical Layer, Data Link Layer, Internet Layer and Transport Layer. The protocol selected for this project was Simple Network Management Protocol (SNMP) that was very popular for managing network devices.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์สำเร็จได้ด้วยดี ก็เนื่องจากความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงงาน ดร.สุรเดช จิตประไพกุลศาลที่กรุณาสละเวลาให้คำแนะนำในการทำงานตลอดถึงการตรวจสอบการทำงานพร้อมทั้งเสนอแนวทางการแก้ไขปัญหาตลอดระยะเวลาการทำโครงงาน ขอขอบคุณรุ่นพี่ซึ่งได้ริเริ่มโครงงานนี้ไว้ และขอขอบพระคุณอาจารย์ทุกท่านและเพื่อนๆ ทุกคนที่ยังไม่ได้เอ่ยนามที่คอยสนับสนุนในการทำโครงงานครั้งนี้

ขอขอบคุณอาจารย์สิรภพ กชรัตน์ และ อาจารย์เศรษฐา ตั้งค้าวานิช ที่กรุณาสละเวลามาเป็นกรรมการสอบโครงงาน พร้อมให้คำแนะนำที่เป็นประโยชน์

และขอขอบคุณบุพการีของคณะผู้จัดทำ ที่มอบความรักความเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่เยาว์จวบจนถึงปัจจุบัน คอยให้กำลังใจให้ผ่านอุปสรรคมาจนถึงทุกวันนี้

สารบัญ

หน้า

บทคัดย่อ ..... ก

Abstract ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูปภาพ ..... ช

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์โครงงาน 1

1.3 ขอบเขตของโครงงาน 1

1.4 ขั้นตอนการดำเนินงาน 2

1.5 แผนการดำเนินงาน 2

1.6 ผลที่คาดว่าจะได้รับ 3

1.7 รายละเอียดงบประมาณของโครงการ 3

บทที่ 2 หลักการและทฤษฎี 4

2.1 แบบจำลอง ทีซีพี/ไอพี (TCP/IP Model) 6

2.2 โพรโทคอล Simple Network Management Protocol (SNMP) 7

บทที่ 3 วิธีการดำเนินงาน 15

3.1 คำอธิบายของระบบ (System Description) 15

3.2 ความต้องการ (Requirement) 16

3.3 สมมติฐานของการออกแบบ (Design Assumption) 16

3.4 การออกแบบโปรแกรม 17

3.4.1 การออกแบบโดยใช้ Use Case Diagram 17

3.4.2 การออกแบบโดยใช้ Activity Diagram 18

3.4.2 การออกแบบโดยใช้ Sequence Diagram 20

สารบัญ (ต่อ)

หน้า

3.4.3 การออกแบบโดยใช้ คลาส Diagram 22

บทที่ 4 ผลการทดลอง 24

4.1 แผนการทดสอบ (Test Plan) 24

4.1.1 Unit Test 25

4.1.2 Integration Test 27

4.1.3 System Test 36

4.2 สรุปผลการทดลอง 43

บทที่ 5 สรุปผล 44

5.1 ผลการทดลอง 44

5.2 ปัญหาและอุปสรรค 45

5.3 ข้อเสนอแนะ 46

5.4 แนวทางในการพัฒนาต่อยอด 46

5.5 สรุป 47

ภาคผนวก ก คู่มือการใช้งาน 48

ภาคผนวก ข คู่มือการพัฒนา 54

เอกสารอ้างอิง 57

ประวัติผู้เขียนโครงการ 58

สารบัญตาราง

หน้า

ตารางที่ 1.1 แผนการดำเนินงาน 2
ตารางที่ 2.1 ตารางรหัสชนิดของข้อมูล 9
ตารางที่ 4.1 ตารางแผนการทดลอง 24
ตารางที่ 4.2 ผลการทดสอบจาก Unit Test จาก JUnit 26
ตารางที่ 4.3 ผลการทดสอบจาก Integration Test จากการ Run Program 27
ตารางที่ 4.4 การเข้ารหัส Version 30
ตารางที่ 4.5 การเข้ารหัส Community String 30
ตารางที่ 4.6 การเข้ารหัส Varbind 30
ตารางที่ 4.7 การเข้ารหัส RequestID 30
ตารางที่ 4.8 การเข้ารหัส ErrorStatus 31
ตารางที่ 4.9 การเข้ารหัส ErrorIndex 31
ตารางที่ 4.10 ตารางโครงสร้างข้อความ 31
ตารางที่ 4.11 การเข้ารหัส PDUType 32
ตารางที่ 4.12 การเข้ารหัส Header 32
ตารางที่ 4.13 แสดงผลการทดสอบจาก System Test จากการ Run 36
ตารางที่ 4.14 สรุปผลการทดลอง 43
ตารางที่ 5.1 ตารางปัญหาและอุปสรรค 45
ตารางที่ 5.2 ตารางข้อเสนอแนะ 46

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 OSI Model 4
รูปที่ 2.2 Application Layer ใน TCP/IP Model และเมื่อเทียบกับ OSI Model 6
รูปที่ 2.3 โพรโทคอล SNMP ใน Application Layer 7
รูปที่ 2.4 Manager และ Agent ใน SNMP 8
รูปที่ 2.5 โครงสร้างการจัดการ และโพรโทคอลที่เกี่ยวข้อง 8
รูปที่ 2.6 ชนิด Simple data type 9
รูปที่ 2.7 ชนิด Structured data type 9
รูปที่ 2.8 ตัวอย่างการถอดรหัส 10
รูปที่ 2.9 PDU ที่รับส่งระหว่าง Manager และ Agent 10
รูปที่ 2.10 แสดง field ต่างๆ ใน PDU 12
รูปที่ 2.11 แสดง field ต่างๆ ใน SNMP Message 13
รูปที่ 2.12 แสดง field ต่างๆ ใน SNMP Message 13
รูปที่ 2.13 แสดง field ต่างๆ ใน SNMP Message หลังจากเข้ารหัสข้อความ 14
รูปที่ 2.14 การทำงานของโพรโทคอล 14
รูปที่ 3.1 โครงสร้างของโปรแกรม 15
รูปที่ 3.2 การทำงานและการเชื่อมโยงของโปรแกรม 15
รูปที่ 3.3 การทำงานของระดับชั้นที่ 5 16
รูปที่ 3.4 Mapping UML and PSP Views 17
รูปที่ 3.5 การออกแบบโดยใช้ Use Case Diagram 17
รูปที่ 3.6 การออกแบบโดยใช้ Activity Diagram (Request) 18
รูปที่ 3.7 การออกแบบโดยใช้ Activity Diagram (Response) 19
รูปที่ 3.8 การออกแบบโดยใช้ Sequence Diagram 20
รูปที่ 3.9 การออกแบบโดยใช้คลาส Diagram 22
รูปที่ 3.10 ส่วนที่ออกแบบเพิ่มเติมโดยใช้ Class Diagram 23
รูปที่ 4.1 Console ของ Manager ที่ส่งข้อมูลไปในแต่ละระดับชั้น 28
รูปที่ 4.2 Console ของ Agent ในการรับข้อมูลไปในแต่ละระดับชั้น 28
รูปที่ 4.3 GUI ของ Manager ที่ส่งข้อความไปให้กับ Agent 29

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.4 GUI ของ Agent ที่รับข้อความจาก Manager 29
รูปที่ 4.5 Console ของ Agent ในการตอบกลับกลับไปยัง Manager 33
รูปที่ 4.6 Console ของ Manager ในการรับข้อความตอบกลับจาก Agent 33
รูปที่ 4.7 GUI ของ Manager ส่ง Request 34
รูปที่ 4.8 Agent รับ Request 34
รูปที่ 4.9 Agent ส่ง Response กลับ 35
รูปที่ 4.10 Manager รับ Response 35
รูปที่ 4.11 GUI ของ Agent เลือก File Connection 37
รูปที่ 4.12 GUI ของ Manager เลือก File Connection และระบุ IP Address ของ Agent 37
รูปที่ 4.13 Console ของทั้ง Manager และ Agent โต้ตอบกับในเครื่องเดียวกัน 38
รูปที่ 4.14 GUI ของ Manager ที่ส่งข้อความไปยัง IP 127.0.0.1 38
รูปที่ 4.15 GUI ของ Agent เลือก Socket Connection 39
รูปที่ 4.16 GUI ของ Agent เลือก Socket Connection และระบุ IP ในวง LAN 39
รูปที่ 4.17 การส่งข้อความผ่านเครือข่าย LAN 40
รูปที่ 4.18 การตอบกลับข้อความผ่านเครือข่าย LAN 40
รูปที่ 4.19 GUI ของ Manager 41
รูปที่ 4.20 GUI ของ Agent 41
รูปที่ 4.21 GUI ของ Manager ที่ส่งข้อความไปยัง IP 192.168.1.3 42
รูปที่ ก.1 หน้าต่างเลือก Client หรือ Server 48
รูปที่ ก.2 หน้าต่างเลือก Client หรือ Server 48
รูปที่ ก.3 GUI และ Console แสดงสำหรับพร้อมรับข้อความ 49
รูปที่ ก.4 หน้าต่างเลือก Client หรือ Server 50
รูปที่ ก.5 หน้าต่างการนำเข้าข้อมูล 50
รูปที่ ก.6 แสดงการใส่ข้อมูลของตัวแปร SNMP 51
รูปที่ ก.7 หน้าต่างการเลือกวิธีการส่งและรูปแบบการแสดงผล 52
รูปที่ ก.8 ผลลัพท์ GUI 53
รูปที่ ก.9 ผลลัพท์ File 53
รูปที่ ข.1 คลาสของโพรโทคอลสแตก 54

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ ข.2 Output ของไลบรารี 55
รูปที่ ข.3 คลาสของแต่ละโพรโทคอล 55
รูปที่ ข.4 Config การใช้โพรโทคอลในแต่ละระดับชั้น 56

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในการจำลองเครือข่ายโพรโทคอลสแตกที่ได้สร้างไลบรารี สำหรับจำลองการทำงานของระบบเครือข่ายขึ้นมาก่อนหน้านี้นั้นได้ทำการพัฒนาระดับชั้นที่ 1 ถึงระดับชั้นที่ 4 (Physical Layer, Data Link Layer, Internet Layer, Transport Layer) แต่ขาดระดับชั้นที่ 5 (Application Layer) เพื่อช่วยให้การจำลองเครือข่ายมีความสมบูรณ์และเข้าใจในระบบเครือข่ายให้มากยิ่งขึ้นจึงมีความคิดที่พัฒนาในส่วนของระดับชั้นที่ 5 เพิ่มเติม ซึ่งได้ทำการเลือกโพรโทคอล Simple Network Management Protocol (SNMP) ซึ่งใช้ในการบริหารและจัดการอุปกรณ์ต่าง ๆในเครือข่าย มาเป็นตัวอย่างในการจำลองการทำงาน ซึ่งโปรแกรมจำลองที่ทำการสร้างขึ้นนี้ง่ายต่อการพัฒนาสามารถเพิ่มโพรโทคอลใหม่ๆที่ต้องการศึกษาเข้าไปได้

1.2 วัตถุประสงค์โครงงาน

1. เพื่อสร้างไลบรารีสำหรับจำลองการทำงานของระบบเครือข่ายในระดับ Application Layer
2. เพื่อให้ผู้ใช้สามารถนำไลบรารีนี้ไปใช้จำลองการทำงานของโพรโทคอลในระดับ Application Layer ได้
3. เพื่อให้ผู้ใช้สามารถนำไลบรารีนี้ไปพัฒนาโพรโทคอลใหม่ๆเพิ่มเติมได้

1.3 ขอบเขตของโครงงาน

1. สามารถจำลองการทำงานของโพรโทคอล SNMP ได้
2. สามารถจำลองการทำงานเชื่อมต่อกับโพรโทคอล 4 ชั้นแรกที่สร้างมาก่อนหน้านี้
3. สามารถจำลองการทำงานภายในเครื่องเดียวกันและผ่านเครือข่าย

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาความรู้ทั่วไปเกี่ยวกับ TCP/IP Model
2. ศึกษาการทำงานของ TCP/IP Protocol
3. ศึกษาการทำงานของไลบรารีก่อนหน้านี้
4. ศึกษาการทำงานโพรโทคอล SNMP
5. สร้างโปรแกรมให้ได้ตามวัตถุประสงค์
6. ทดสอบความถูกต้องของโปรแกรม
7. จัดทำเอกสารและคู่มือการใช้

1.5 แผนการดำเนินงาน

กิจกรรม	ปี 2553						ปี 2554			
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาความรู้ทั่วไปเกี่ยวกับ TCP/IP Model										
2. ศึกษาการทำงานของ TCP/IP Protocol										
3. ศึกษาการทำงานของไลบรารีก่อนหน้านี้										
4. ศึกษาการทำงานของโพรโทคอล SNMP										
5. สร้างโปรแกรมให้ได้ตามวัตถุประสงค์										
6.ทดสอบความถูกต้องของโปรแกรม										
7.จัดทำเอกสารและคู่มือการใช้										

ตารางที่ 1.1 แผนการดำเนินงาน

1.6 ผลที่คาดว่าจะได้รับ

1. โปรแกรมที่พัฒนาสามารถแสดงให้เห็นการทำงานในโพรโทคอล SNMP ได้
2. โปรแกรมที่พัฒนาสามารถแสดงให้เห็นภาพการทำงานของการจำลองโพรโทคอลสแตกครบทุกระดับชั้น
3. โปรแกรมที่พัฒนาสามารถใช้เป็นไลบรารีในของพัฒนาโพรโทคอลใหม่ๆในอนาคตได้

1.7 รายละเอียดงบประมาณของโครงการ

1.	หนังสือประกอบการทำโคร	1,000 บาท
2.	ค่าเอกสาร	500 บาท
3.	อื่นๆ	500 บาท
	รวมเป็นเงินทั้งสิ้น	2,000 บาท (สองพันบาทถ้วน)

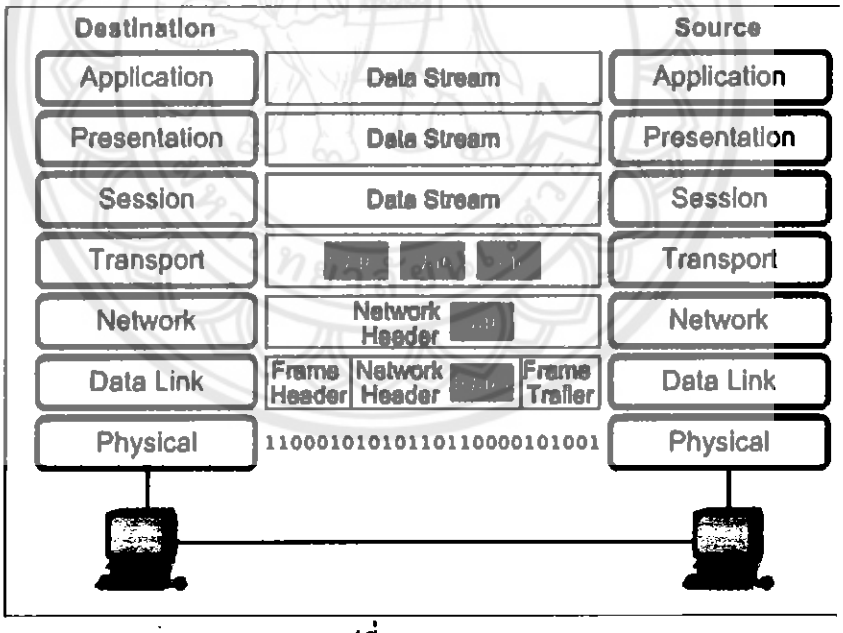
มหาวิทยาลัยนเรศวร

บทที่ 2
หลักการและทฤษฎี

บทนี้จะกล่าวถึงทฤษฎี และหลักการในการพัฒนาโปรแกรมเพื่อจำลองการทำงานโพรโทคอลสแตกชั้นการประยุกต์ผ่านการพัฒนาโพรโทคอล Simple Network Management Protocol (SNMP) โดยมีหลักการและทฤษฎีที่เกี่ยวข้องดังนี้ คือ แบบจำลอง ทีซีพี/ไอพี (TCP/IP model) และโพรโทคอล SNMP

ระบบเครือข่ายคอมพิวเตอร์ คือ กลุ่มของคอมพิวเตอร์หรืออุปกรณ์ต่างๆ ที่ถูกนำมาเชื่อมต่อกันเพื่อให้ผู้ใช้ในเครือข่ายติดต่อสื่อสาร แลกเปลี่ยนข้อมูล และใช้อุปกรณ์ต่างๆภายในเครือข่ายร่วมกัน ระบบเครือข่ายนั้นมีหลายขนาด ตั้งแต่ขนาดเล็กที่เชื่อมต่อกันด้วยคอมพิวเตอร์หรืออุปกรณ์เพียงสองสามเครื่องไปจนถึงเครือข่ายขนาดใหญ่ที่เชื่อมต่อกันทั่วโลก

OSI model คือ แบบจำลองการทำงานของระบบเครือข่าย ถูกแบ่งออกเป็น 7 ชั้น การส่งข้อมูลจะทำการไล่จาก Application Layer ลงมาจนถึง Physical Layer



รูปที่ 2.1 OSI Model

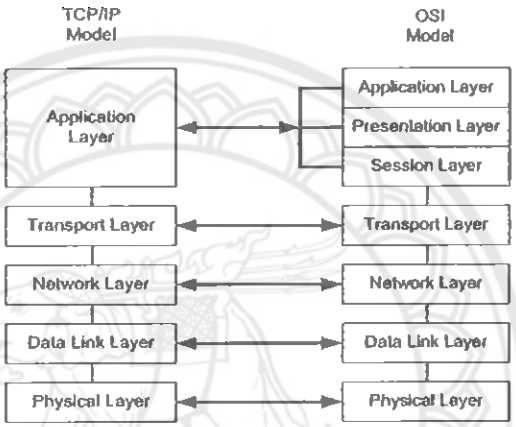
ที่มา: [1]

รายละเอียดของ OSI model แต่ละชั้น

- **Application Layer** เป็นชั้นบนสุดที่อยู่ใกล้ผู้ใกล้ผู้ใช้มากที่สุดและเป็นชั้นที่ทำงานส่งและรับข้อมูลโดยตรงกับผู้ใช้
- **Presentation Layer** เป็นชั้นที่รับผิดชอบเรื่องรูปแบบของการแสดงผลเพื่อโปรแกรมต่างๆที่ใช้งานระบบเครือข่ายทำให้ทราบว่าข้อมูลที่ได้เป็นประเภทใด
- **Session Layer** นี้ทำหน้าที่ในการจัดการกับเซสชั่นของโปรแกรม ชั้นนี้เองที่ทำให้ในหนึ่งโปรแกรมยกตัวอย่างเช่น Web browser สามารถทำงานติดต่ออินเตอร์เน็ตได้พร้อมๆกันหลายหน้าต่าง
- **Transport Layer** ชั้นนี้ทำหน้าที่ดูแลจัดการเรื่องของความผิดพลาดที่เกิดขึ้นจากการสื่อสาร ซึ่งการตรวจสอบความผิดพลาดนั้นจะพิจารณาจากข้อมูลส่วนที่เรียกว่า Checksum และอาจมีการแก้ไขข้อผิดพลาดนั้นๆ โดยพิจารณาจาก ฝั่งต้นทางกับฝั่งปลายทาง (End-to-end) โดยหลักๆแล้วชั้นนี้จะอาศัยการพิจารณาจาก พอร์ต (Port) ของเครื่องต้นทางและปลายทาง
- **Network Layer** จะจัดการการติดต่อสื่อสารข้ามเน็ตเวิร์ก ซึ่งจะเป็นการทำงานติดต่อข้ามเน็ตเวิร์คแทนชั้นอื่นๆที่อยู่ข้างบน
- **Data Link Layer** ชั้นนี้จัดเตรียมข้อมูลที่จะส่งผ่านไปบนสื่อตัวกลาง
- **Physical Layer** - ชั้นล่างสุดเป็นชั้นของสื่อที่ใช้ในการติดต่อสื่อสาร ซึ่งอาจจะเป็นทั้งแบบที่ใช้สายหรือไม่ใช้สาย ตัวอย่างของสื่อที่ใช้ เช่น Shielded Twisted Pair (STP), Unshielded Twisted Pair (UTP), Fiber Optic และอื่นๆ

2.1 แบบจำลอง ทีซีพี/ไอพี (TCP/IP Model)

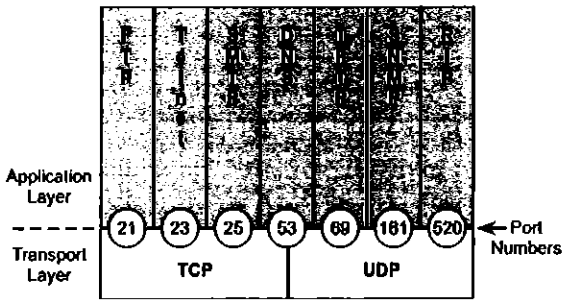
เป็นมาตรฐานที่ทำให้คอมพิวเตอร์ภายในระบบเครือข่ายอินเทอร์เน็ต สามารถเชื่อมต่อเข้าหากัน และติดต่อสื่อสารแลกเปลี่ยนข้อมูลกันได้ เป็นมาตรฐานที่ว่าด้วยการกำหนดวิธีการติดต่อสื่อสารระหว่างคอมพิวเตอร์โดยใช้แนวคิดของการแบ่งลำดับชั้นโพรโทคอลลำดับชั้นของโพรโทคอลในระบบอินเทอร์เน็ตมีลำดับชั้นที่น้อยกว่าโครงสร้างลำดับชั้นของโอเอสไอ (OSI Model) โดยในโอเอสไอมีลำดับชั้นของโพรโทคอลทั้งหมด 7 ชั้น แต่ในระบบอินเทอร์เน็ตมีทั้งหมดเพียง 5 ระดับชั้นเท่านั้น (The five-layer TCP/IP model)



รูปที่ 2.2 Application Layer ใน TCP/IP Model และเมื่อเทียบกับ OSI Model

ที่มา: [2]

ชั้นการประยุกต์ใช้งานหรือ Application Layer เทียบได้กับระดับชั้นที่ 5 ถึง 7 ในโครงสร้างแบบ โอเอสไอ จะครอบคลุมบริการที่เกี่ยวข้องกับการรักษาความปลอดภัย การเข้ารหัส การเชื่อมต่อระหว่างโปรแกรมประยุกต์ และเป็นชั้นที่โปรแกรมประยุกต์ใช้งานโดยตรง โดยโพรโทคอลที่อยู่บนชั้นนี้จะถูกออกแบบให้เหมาะสำหรับประเภทของโปรแกรมประยุกต์เฉพาะทาง เช่น โปรแกรมอีเมลใช้โปรโตคอล SMTP สำหรับส่งอีเมล ใช้โปรโตคอล POP3 สำหรับรับและเรียกดูอีเมล ส่วนโปรแกรมเว็บเบราว์เซอร์ใช้โปรโตคอล HTTP สำหรับเรียกดูเว็บเพจ เป็นต้น



รูปที่ 2.3 โพรโทคอล SNMP ใน Application Layer

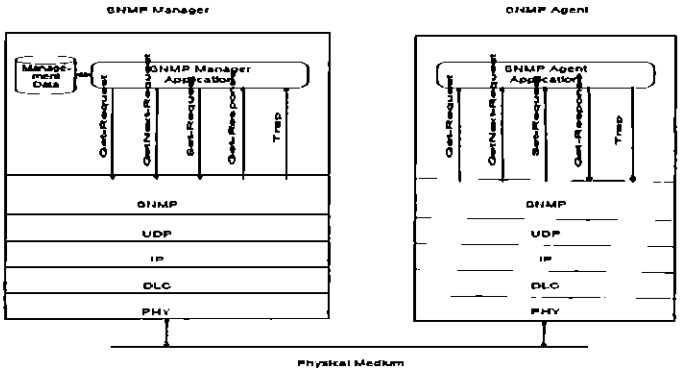
ที่มา: [3]

2.2 โพรโทคอล Simple Network Management Protocol (SNMP)

SNMP เป็นโพรโทคอลที่อยู่ระดับบนในชั้นการประยุกต์ และเป็นส่วนหนึ่งของชุดโพรโทคอล TCP/IP ซึ่งเป็นมาตรฐานการจัดการเครือข่ายที่ใช้งานได้ผลดี ในการบริการและจัดการเครือข่ายต้องใช้อุปกรณ์ต่าง ๆ มีส่วนของการทำงานร่วมกับระบบจัดการเครือข่าย ซึ่งเราเรียกว่า ตัวกระทำ (Agent) ตัวกระทำเป็นส่วนของซอฟต์แวร์ที่อยู่ในอุปกรณ์ต่างๆ ที่เชื่อมอยู่ในเครือข่ายโดยมีคอมพิวเตอร์หลักในระบบหนึ่งเครื่องเป็น ผู้จัดการ (Manager) และบริหารเครือข่ายหรือเรียกว่า NMS-Network Management System โดยมุ่งการทำงานกับ User Data Protocol (UDP[1]) การทำงานของ UDP เป็นลักษณะแบบ Connectionless คือ ไม่ต้องมีการสร้าง Connection ก่อนที่จะส่งข้อมูล UDP เหมาะสำหรับส่ง Message สั้นๆ

---

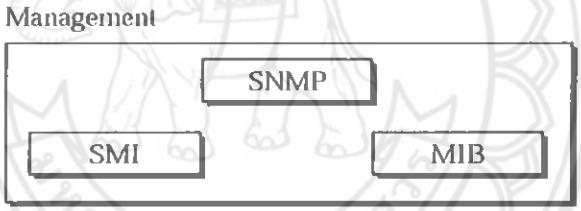
[1] User Datagram Protocol (UDP) เป็นโพรโทคอลหลักในชุดโพรโทคอลอินเตอร์เน็ต การส่งข้อมูลผ่าน UDP นั้น คอมพิวเตอร์จะส่งข้อมูลขนาดเล็กที่เรียกว่า ดาต้าแกรม (Datagram) ผ่านเครือข่ายไปยังเครื่องปลายทาง โดย UDP จะไม่รับประกันความน่าเชื่อถือและลำดับของเดต้าแกรม อย่างที่ TCP รับประกัน ซึ่งหมายความว่าเดต้าแกรมอาจมาถึงไม่เรียงลำดับ หรือสูญหายระหว่างทางได้



รูปที่ 2.4 Manager และ Agent ใน SNMP

ที่มา: [3]

ซึ่งในการส่วนประกอบของการจัดการเครือข่ายนี้จะมีโพรโทคอล องค์ประกอบหลักอยู่ 3 อย่างคือ SNMP, Structure of Management Information (SMI) และ Management Information Base (MIB) โดยที่ SMI คือโครงสร้างของข้อมูลที่จะจัดการ และ MIB คือการจัดการข้อมูล



รูปที่ 2.5 โครงสร้างการจัดการ และโพรโทคอลที่เกี่ยวข้อง

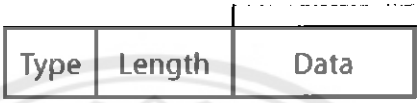
ที่มา: [4]

SNMP เป็นโพรโทคอลนั้นชั้นประยุกต์ที่ถูกออกแบบมาเพื่อใช้สังเกตการณ์อุปกรณ์เครือข่ายชนิดต่างๆได้ไม่ว่าจะถูกผลิตมาจากโรงงานใดๆก็ตาม หรือ ลักษณะทางกายภาพของเครือข่ายมีความแตกต่างกัน

หลักการและแนวคิดของโพรโทคอล อาจจะเปรียบได้ว่าตัว SNMP เป็น Manager และ Agent เป็น Routers รูปแบบการทำงานคือ SNMP เป็นตัวกำหนดรูปแบบของ Packet ในการแลกเปลี่ยนระหว่าง Manager และ Agent สามารถอ่านและเปลี่ยนค่าต่างๆภายใน Packet ได้

SMI เป็นโครงสร้างการกำหนดกฎทั่วไปของการตั้งชื่อ (Name) สิ่งของ (Object), ชนิด (Type) ของ Object วิธีการเข้ารหัสต่างๆ (Encoding Method) ต่างๆของการจัดการที่ใช้ในการสื่อสารภายในระบบ

MIB เป็นส่วนหนึ่งของโครงสร้าง SMI โดยที่จะเน้นการจัดการที่ใช้ในการ เก็บ Object ทั้งหมดที่ Manager สามารถจัดการได้ โดยทำหน้าที่สร้างชุดของชื่อ ชนิด และความสัมพันธ์ของ Objects ที่จะต้องถูกจัดการภาพรวมในการทำงานพื้นฐานการถอดรหัส SNMP Message การถอดข้อความ กฎพื้นฐานที่สุดของ Message ก็คือแต่ละ Field จะประกอบด้วย 3 ส่วน คือ Type, Length และ Data โดยที่ Type จะแบ่งชนิดของข้อมูลออกเป็น 2 ชนิด คือ Simple data type และ Structured data type ดังในรูปที่ 2.5 และรูปที่ 2.7 และ Length เป็นตัวกำหนดขนาดของ Data ว่าจะมีค่ากี่ Byte และ Data คือข้อมูลของ Message นั้น



รูปที่ 2.6 ชนิด Simple data type

ที่มา: [5]

รูปที่ 2.7 ชนิด Structured data type

ที่มา: [5]

Data Type	*Class*	*Format*	*Number*	*Tag (Binary)*	*Tag (Hex)*
INTEGER	00	0	00010	00000010	**02**
OCTET STRING	00	0	00100	00000100	**04**
OBJECT IDENTIFIER	00	0	00110	00000110	**06**
NULL	00	0	00101	00000101	**05**
Sequence, sequence of	00	1	10000	00110000	**30**
IPAddress	01	0	00000	01000000	**40**
Counter	01	0	00001	01000001	**41**
Gauge	01	0	00010	01000010	**42**
TimeTicks	01	0	00011	01000011	**43**
Opaque	01	0	00100	01000100	**44**

ตารางที่ 2.1 ตารางรหัสชนิดของข้อมูล

ที่มา: [4]

04	02	48	49
00000100	00000010	01001000	01001001
Tag (String)	Length (2 bytes)	Value (H)	Value (I)

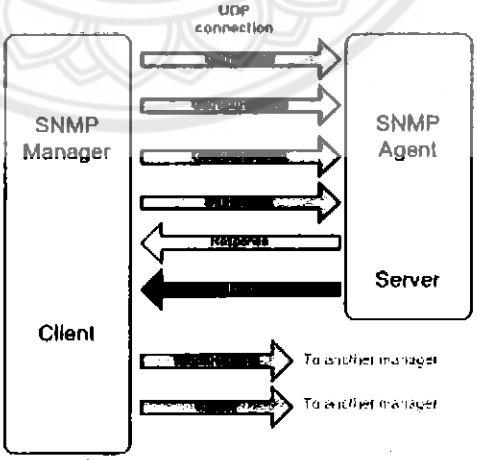
รูปที่ 2.8 ตัวอย่างการถอดรหัส

ที่มา: [4]

- ตัวเลขชุดแรก คือ 00000100 เป็น Tag ของข้อมูลเมื่อนำไปเทียบกับ ตารางรหัสชนิดของข้อมูลจะได้ว่าข้อมูลในชุดแรกนี้มีข้อมูลเป็นประเภท OCTET STRING ซึ่งข้อมูลภายในคือ HI
- ตัวเลขชุดที่สอง คือ 00000010 เป็น Length ของข้อมูลเมื่อแปลงเป็นเลขฐานสองแล้วจะมีค่าเท่ากับ 2 นั้นหมายความว่า Value จะมีค่า 2 Byte (16 bit)
- ตัวเลขชุดที่สามและสี่ คือข้อมูล String ขนาด 2 byte มีค่าคือ HI

SNMP ในการจัดการนี้ใช้ทั้ง SMI และ MIB เป็นตัวช่วยในการทำงานนี้ หลังจาก Manager ได้รับค่าคืนกลับมาจาก Agent แล้ว Manager จะเก็บค่าที่ได้รับลงใน Object ที่ถูกกำหนดโดย Agent เมื่อเกิดเหตุการณ์ผิดพลาด Agent ส่งสัญญาณเตือนไปยัง Manager

ใน SNMPv3 ได้กำหนดประเภทของ Packet หรือ Protocol Data Unit (PDU) ไว้ 8 ประเภทด้วยกันคือ GetRequest, GetNextRequest, GetBulkRequest, SetRequest, Response, Trap และ Report



รูปที่ 2.9 PDU ที่รับส่งระหว่าง Manager และ Agent

ที่มา: [4]

GetRequest

ในการ GetRequest นั้น Manager จะส่ง PDU ไปยัง Agent เพื่อบอกว่า Manager ต้องการทราบข้อมูลอะไรจาก Agent ซึ่งกำหนดโดย Object Identifier ที่ส่งไปพร้อมกับ Message

GetNextRequest

ในการ GetNextRequest นั้น Manager จะส่ง PDU ไปยัง Agent เพื่อที่จะเอาค่าตัวแปร โดยค่าที่ได้จะเป็นค่าของ Object ตัวถัดจาก Object ID ตัวที่กำหนดใน PDU ส่วนใหญ่มักจะใช้ในการเอาค่าของการเข้าไปตาราง ถ้า Manager ไม่รู้ index ของการเข้ามันจะไม่สามารถดึงค่าออกมาได้แต่เราใช้ GetNextRequest และกำหนด Object ID ของตารางเพราะ Entry ตัวแรกจะต่อจาก Object ID ของตารางทันที ก็จะได้ค่าของ First entry ตัวแรกกลับมา

GetbulkRequest

ในการ GetbulkRequest นั้น Manager จะส่ง PDU ไปยัง Agent เพื่อที่จะเอาค่าของข้อมูลจำนวนมาก สามารถใช้แทนกรณีมีหลาย GetRequest

SetRequest

ในการ SetRequest นั้น Manager จะส่ง PDU ไปที่ Agent เพื่อที่จะ Set (Store) ค่าลงในตัวแปร หรือเปลี่ยนแปลงค่า Configuration ต่าง ๆของข้อมูลใน MIB ของอุปกรณ์นั้น ๆ

Response

ในการ Response นั้น Agent จะส่ง PDU ไปที่ Manager เพื่อที่จะตอบสนอง GetRequest หรือ GetNextRequest ซึ่งจะบรรจุค่าของตัวแปรที่ร้องขอโดย Manager

Trap

ในการ Trap นั้น Agent จะส่ง PDU ไปยัง Manager เพื่อรายงานเหตุการณ์ที่เกิดขึ้น เช่น ถ้า Agent ทำการรีบูต (Reboot) ก็จะแจ้ง Manager และรายงานเวลาที่ทำการรีบูต

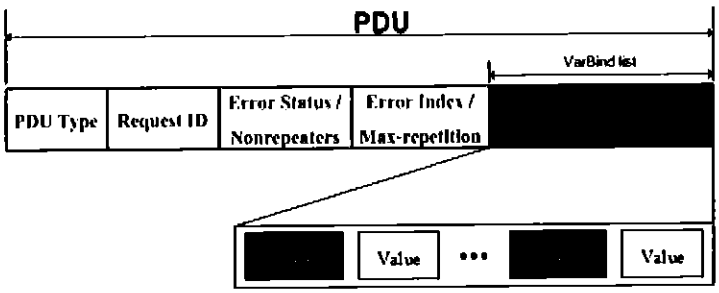
InformRequest

ในการ InformRequest นั้น Manager จะส่ง PDU ไปยัง Manager ที่ห่างไกล ซึ่ง Remote เข้ามารับค่าบางอย่างจาก Agents ซึ่งจะตอบกลับด้วย Response PDU

Report

ใน Report นั้น PDU จะถูกออกแบบให้รายงานข้อผิดพลาดบางอย่างระหว่าง Manager ซึ่งในตอนนี้ยังไม่ถูกใช้

รูปแบบของ SNMP PDU



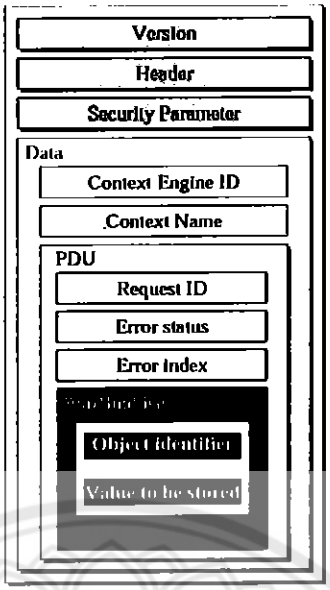
รูปที่ 2.10 แสดง field ต่างๆ ใน PDU

ที่มา: [4]

ประกอบไปด้วย

- PDU Type: กำหนดประเภทของ PDU
- Request - ID:หมายเลขการร้องขอที่ใช้โดย Manager ใน Request PDU และใช้ซ้ำโดย Agent ในการตอบสนอง
- Error Status:ใช้เพียงแค่ใน Response PDU เท่านั้น โดยใช้ในการแสดงประเภทของข้อผิดพลาดที่รายงานโดย Agent ซึ่งค่าในช่องนี้เป็น 0 เมื่อ PDU เป็นชนิด Request
- Nonrepeaters: ใช้เมื่อ PDU เป็นชนิด GetBulkRequest
- Error Index: ใช้บอกกับ Manager ว่าตัวแปรไหนเป็นสาเหตุของข้อผิดพลาด
- Max-repetition: ใช้เมื่อ PDU เป็นชนิด GetBulkRequest
- VarBind list: เซตของตัวแปรกับค่าที่ Manager ต้องการเอามาหรือต้องการกำหนด

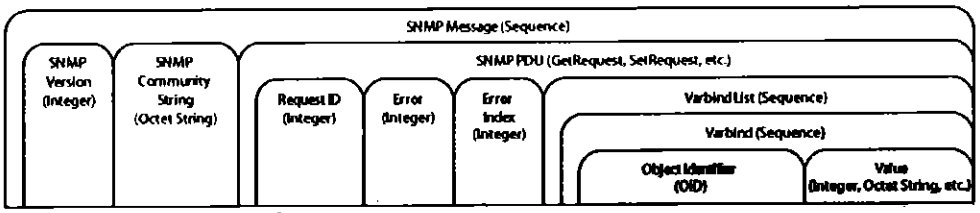
การทำงานของโพรโทคอล SNMP นั้นไม่ได้ส่งแค่ Packets ที่เรียกว่า PDU เพียงอย่างเดียว แต่ส่ง PDU ไปพร้อมกับข้อความ ซึ่งใน Version 3 นี้ประกอบไปด้วย Version, Header, Security Parameter, Data (PDU ที่ได้กล่าวถึงมา เป็นส่วนประกอบใน Data)



รูปที่ 2.11 แสดง field ต่างๆ ใน SNMP Message

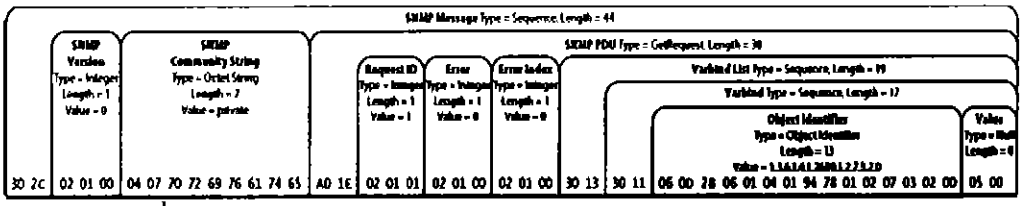
ที่มา: [4]

- Version กำหนดให้ใช้รุ่นปัจจุบันคือ Version 3
- Header จะบรรจุค่า Message Identifier, Maximum message size, Message flag (1 octet ของข้อมูลแบบ Octet String ซึ่งแต่ละบิตกำหนด Security type เช่น Privacy หรือ Authentication หรือข้อมูลอื่นๆ) และ Message security model
- Message Security Parameter ใช้ในการสร้าง message digest
- Data จะบรรจุ PDU ถ้าข้อมูลได้รับการเข้ารหัสตรงนี้ก็จะเป็นข้อมูลเกี่ยวกับ Encrypting engine และ Encrypting context (ประเภทของการเข้ารหัส) ตามมาด้วยการเข้ารหัสของ PDU ถ้าข้อมูลไม่ได้ทำการเข้ารหัส ส่วนของ Data จะมีแค่ PDU



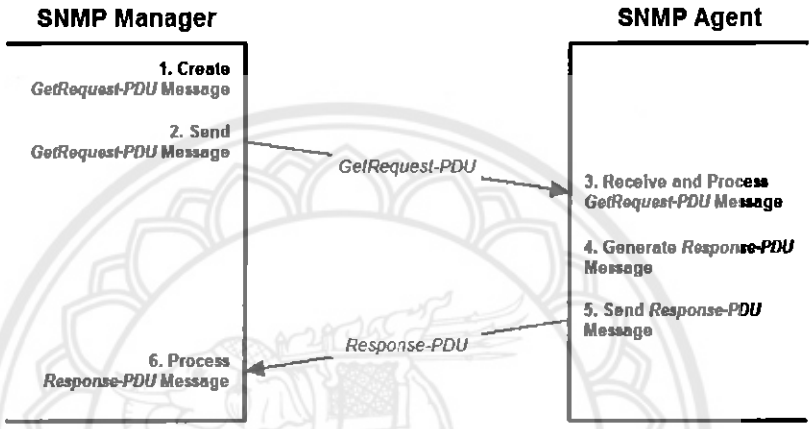
รูปที่ 2.12 แสดง field ต่างๆ ใน SNMP Message

ที่มา: [5]



รูปที่ 2.13 แสดง field ต่างๆ ใน SNMP Message หลังจากเข้ารหัสข้อความ

ที่มา: [5]



รูปที่ 2.14 การทำงานของโพรโทคอล

ที่มา: [5]

ขั้นตอนการทำงาน

1. Manager สร้าง PDU Message เพื่อส่งไปขอข้อมูลจาก Agent
2. Manager ส่ง PDU Message ที่สร้างนั้นไปให้กับ Agent
3. Agent รับ PDU Message แล้วประมวลผล
4. Agent สร้าง PDU Response ตอบกลับ
5. Agent ส่ง Response PDU กลับไปให้กับ Manager
6. Manager รับข้อความ Response PDU แล้วประมวลผลต่อไป

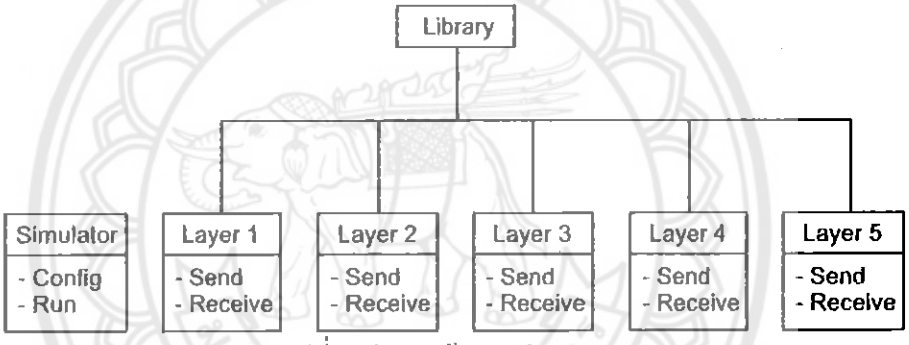
บทที่ 3
วิธีการดำเนินงาน

วิธีดำเนินนั้นเริ่มการศึกษาระบบของโครงงานเดิม[6,7] ว่าถูกออกแบบใด แล้วจึงเพิ่มเติมงานในส่วนของคณะผู้จัดทำเข้าไป ซึ่งในบทนี้จะกล่าวถึงการออกแบบและพัฒนาระบบต่อไป

3.1 คำอธิบายของระบบ (System Description)

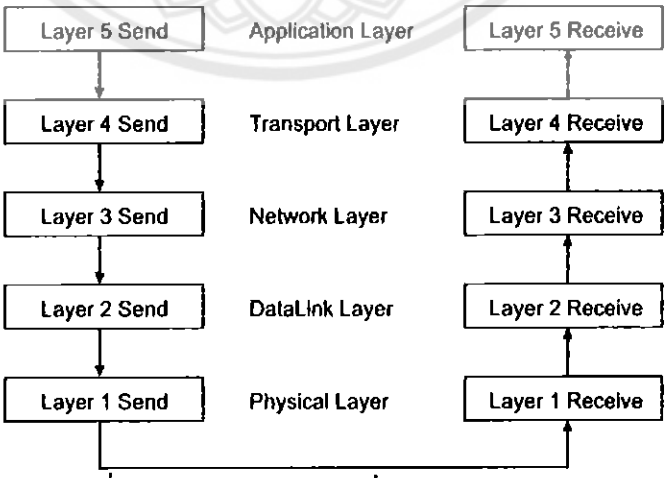
โปรแกรมนี้ประกอบไปด้วยส่วนหลักๆ คือ

- Simulator ซึ่งเป็นตัวตั้งค่าการใช้งาน และตัวควบคุมการทำงาน
- Layer 1 – Layer 5 ซึ่งในแต่ละระดับชั้นจะมีตัวส่งและตัวรับ เป็นตัวดำเนินงาน



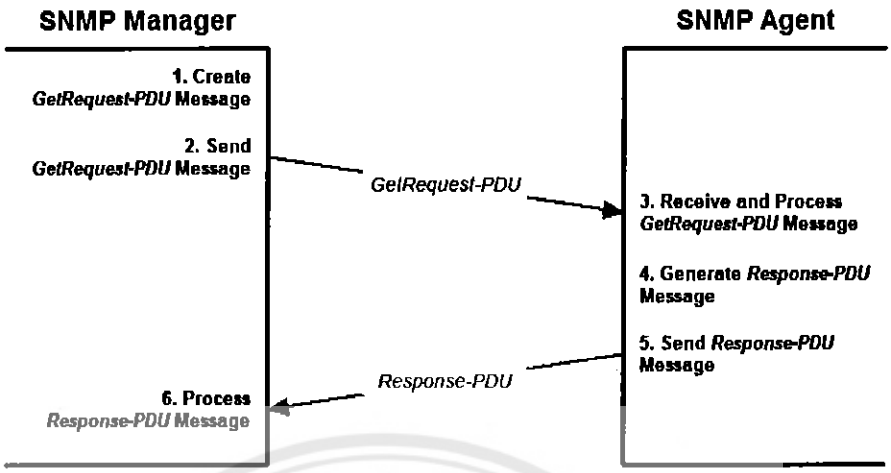
รูปที่ 3.1 โครงสร้างของโปรแกรม

ที่มา: [6,7]



รูปที่ 3.2 การทำงานและการเชื่อมโยงของโปรแกรม

ที่มา : [6,7]



รูปที่ 3.3 การทำงานของระดับชั้นที่ 5

ที่มา: [3]

3.2 ความต้องการ (Requirement)

- สามารถขยายความสามารถเพิ่มเติมต่อจากไลบรารี[6,7]ก่อนหน้านี้
- สามารถแสดงโครงสร้างของข้อความที่ออกมาจากโพรโทคอลได้ถูกต้อง
- สามารถแสดงการโต้ตอบการทำงานภายในในโพรโทคอลได้ถูกต้อง

3.3 สมมติฐานของการออกแบบ (Design Assumption)

การส่งข้อมูลระหว่างระดับชั้นเป็นลักษณะการ Encapsulate โดยแต่ละชั้นจะมีการเพิ่ม Header ของแต่ละระดับชั้น และการโต้ตอบภายในโพรโทคอล SNMP เป็นการโต้ตอบแบบ Client – Server

3.4 การออกแบบโปรแกรม

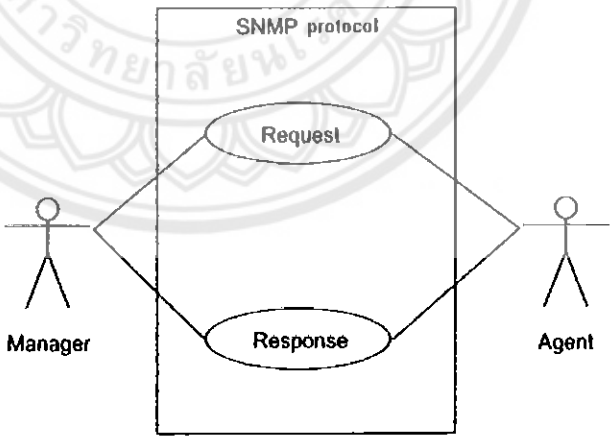
การออกแบบโปรแกรมใช้มาตรฐาน UML (Unified Modeling Language) ที่เป็นภาษาที่ใช้แสดงแบบการทำงานของระบบ ซึ่งอธิบายพฤติกรรมของระบบมุมมองต่างๆด้วยแผนภาพ ดังกล่าวนี้

	Dynamic	Static
External	Use Case Diagram Sequence Diagram	Class Diagram
Internal	Activity Diagram	

รูปที่ 3.4 Mapping UML and PSP[2] Views

3.4.1 การออกแบบโดยใช้ Use Case Diagram

การออกแบบนี้แสดงให้เห็นถึงวิธีการของระบบเครือข่ายที่ Manager กระทำการส่งการร้องขอ (Request) ไปยัง Agent เพื่อขอข้อมูลจาก Agent และ Agent ตอบกลับข้อมูลที่ร้องขอ (Response) กลับมายัง Manager นั้น สามารถอธิบายได้ตามภาพด้านล่างนี้



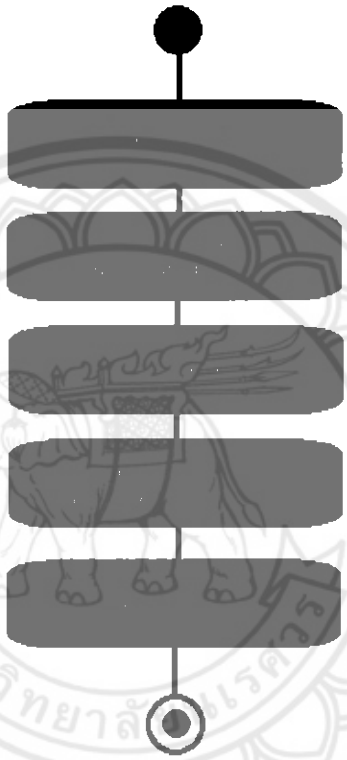
รูปที่ 3.5 การออกแบบโดยใช้ Use Case Diagram

---

[2] PSP (The Personal Software Process) กระบวนการที่ช่วยจัดการคุณภาพของโครงการ ช่วยปรับปรุงการประเมินและวางแผน ช่วยลดข้อผิดพลาดของผลิตภัณฑ์

3.4.2 การออกแบบโดยใช้ Activity Diagram

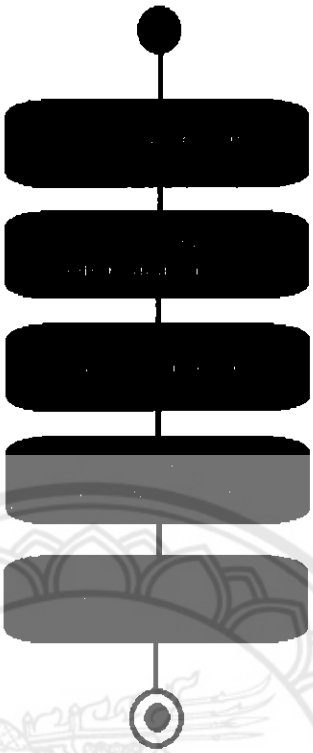
การออกแบบโดยใช้ Activity Diagram เป็นการบอกกิจกรรมการทำงานของระดับชั้น โดยมีวงกลมสีดำเป็นจุดเริ่มต้น และ มีวงกลมที่ดำที่มีวงกลมล้อมไว้อีกชั้น คือ จุดสิ้นสุดของกิจกรรม โดยในที่นี้จะอธิบายถึงสองกิจกรรมที่มีทั้งจากส่งการร้องขอ และ การส่งการตอบสนอง อธิบายได้ตามภาพด้านล่าง



รูปที่ 3.6 การออกแบบโดยใช้ Activity Diagram (Request)

ลำดับของเหตุการณ์ (Flow of Events) ของการส่งการร้องขอไปยัง Server

1. Client ส่งการร้องขอไปยัง Server ที่ต้องการ
2. Server ได้รับข้อมูล และใช้ MIBv2 แยกแยะได้ว่า ข้อมูลที่ได้รับเป็น UDP datagram
3. Server ใช้ SMI จำแนกและถอดรหัสว่าเป็นข้อมูลว่าเป็นวัตถุชนิดใด ชื่ออะไร และมีขนาดข้อมูลเท่าไร Server ใช้ SNMP ในการถอดรหัสข้อความว่าเป็นข้อความชนิดใด มี PDU ประเภทใดส่งเข้ามาส่งข้อความตอบกลับไปยัง Client



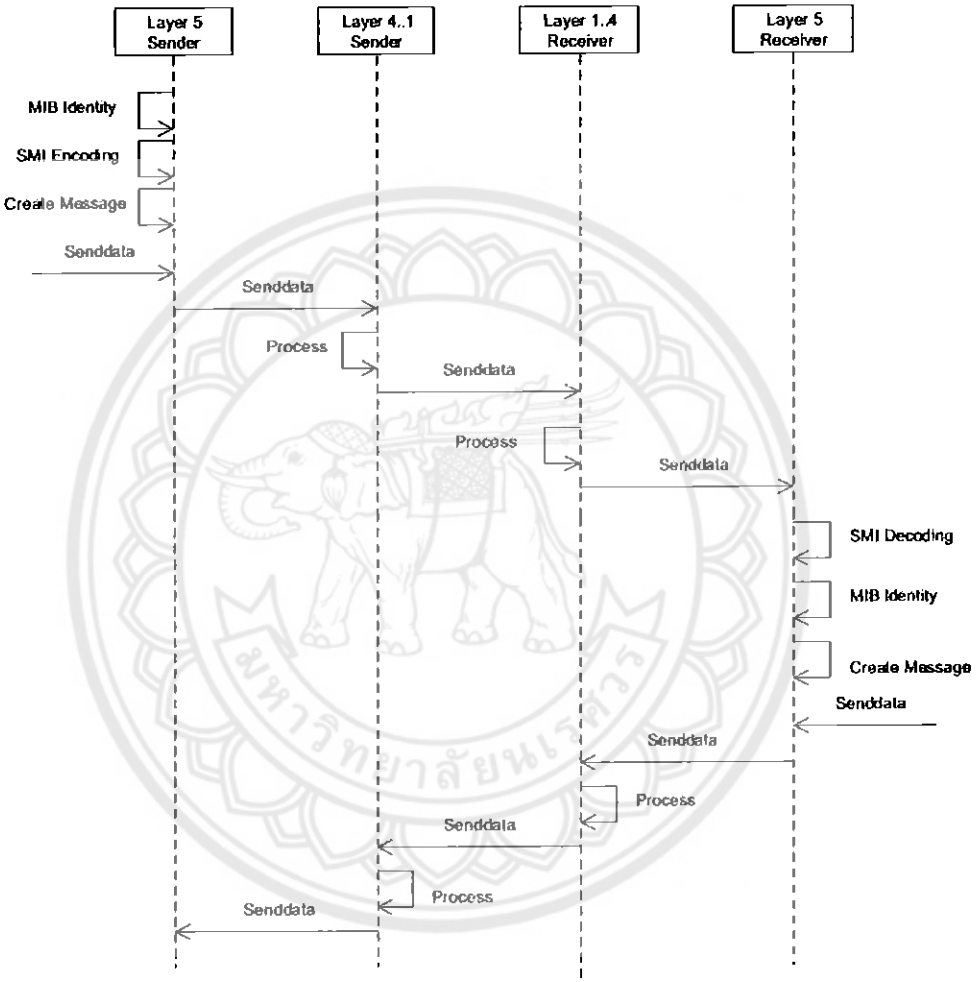
รูปที่ 3.7 การออกแบบโดยใช้ Activity Diagram (Response)

ลำดับของเหตุการณ์ (Flow of Events) ของการตอบกลับไปยัง Client

1. Server ส่งข้อความตอบกลับไปยัง Client
2. Client ได้รับข้อมูล และใช้ MIBv2 แยกแยะได้ว่า ข้อมูลที่ได้รับเป็น UDP datagram
3. Client ใช้ SMI จำแนกและถอดรหัสว่าเป็นข้อมูลว่าเป็นวัตถุชนิดใด ชื่ออะไร และ มีขนาดข้อมูลเท่าไร
4. Client ใช้ SNMP ในการถอดรหัสข้อความว่าเป็นข้อความชนิดใด มี PDU ประเภทใดส่งเข้ามาได้รับการตอบสนองจาก Server แล้ว

3.4.2 การออกแบบโดยใช้ Sequence Diagram

Sequence Diagram คือ Diagram ที่ใช้แสดงลำดับการกระทำระหว่างวัตถุ โดยแสดงลำดับการทำงานของระบบเครือข่ายโดยใช้โพรโทคอล SNMP ซึ่งสามารถเขียนอธิบายได้เป็นภาพตามภาพด้านล่างนี้

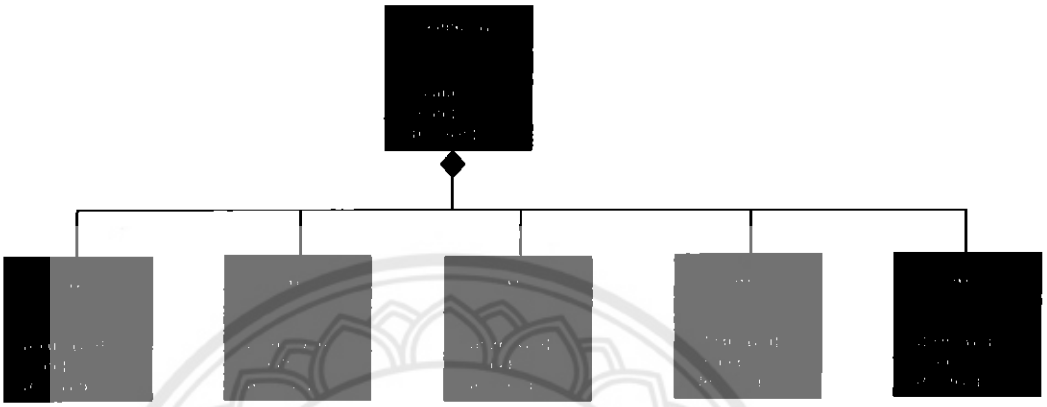


รูปที่ 3.8 การออกแบบโดยใช้ Sequence Diagram

1. ผู้ส่งจะทำการเลือก MIB ที่ต้องการทราบค่า
2. ทำการ Encoding ตามโพรโทคอล SNMP
3. สร้างข้อความ SNMP PDU
4. ส่งข้อความไปยังระดับชั้นที่ 4
5. ในระดับชั้นที่ 4 ถึงระดับชั้นที่ 1 ก็มีโพรโทคอลของแต่ละระดับชั้นทำงาน (Process) เช่น TCP, IP, Ethernet, UTP
6. ฝั่งผู้รับได้รับข้อความก็ทำการ Decoding ตามโพรโทคอลเพื่อให้รู้ว่าผู้ส่งต้องการทราบค่าอะไร
7. ระบุค่า MIB ที่ผู้ส่งต้องการทราบค่า
8. สร้างข้อความจาก MIB Response กลับไป

3.4.3 การออกแบบโดยใช้ คลาส Diagram

คลาส Diagram เป็น Diagram ที่แสดงความสัมพันธ์ระหว่าง Object โดยเขียนเพื่ออธิบายระบบเครือข่ายที่ได้สร้างก่อนหน้านี้ ได้ตามภาพด้านล่างนี้



รูปที่ 3.9 การออกแบบ โดยใช้คลาส Diagram

ที่มา: [6,7]

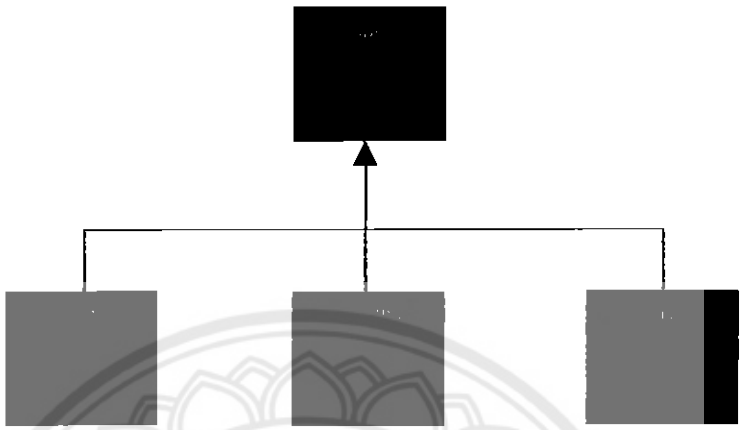
ในส่วนของคลาส ConfigLayer มีเมท็อดประกอบอยู่ภายในทั้งหมด 3 เมท็อด คือ

1. Method Config ทำหน้าที่เป็นตัวเลือกว่าจะเลือกเป็นฝ่ายรับหรือฝ่ายส่ง
2. Method Send ทำหน้าที่เป็นตัวส่งข้อมูล
3. Method Receive ทำหน้าที่เป็นตัวรับข้อมูล

ในส่วนของคลาส Layer ทั้ง 5 มีเมท็อดประกอบอยู่ภายในทั้งหมด 3 เมท็อด คือ

1. Method NextLayer ทำหน้าที่เป็นตัวเชื่อมต่อระหว่าง Layer
2. Method Send ทำหน้าที่เป็นตัวส่งข้อมูล
3. Method Receive ทำหน้าที่เป็นตัวรับข้อมูล

ในส่วนของที่โครงงานนี้จะทำขึ้นจะเป็นส่วนของ Layer 5 ที่เชื่อมต่ออยู่กับ Layer 4 นั้น แบ่งออกเป็น 3 คลาส ย่อยคือ คลาส L5_SMI, L5_SNMPv3 และ L5_MIBv2



รูปที่ 3.10 ส่วนที่ออกแบบเพิ่มเติมโดยใช้ Class Diagram

- คลาส L5_SMI จะทำในส่วนของการ Encode ข้อความจาก MIB ที่ต้องการ
- คลาส L5_SNMPv3 จะทำในเรื่องของการ Create Message หลังจากการ Encoding ใน L5_SMI แล้ว
- คลาส MIBv2 จะทำในส่วนของการตอบกลับข้อความ MIB ที่ต้องการ

บทที่ 4

ผลการทดลอง

ในบทนี้เราจะกล่าวถึงแผนการทดสอบ ซึ่งแบ่งเป็นการทดสอบใหญ่ๆสามหัวข้อ คือ

- Unit Test คือการทดสอบคลาสและเมท็อดว่าสามารถทำงานได้ถูกต้อง
- Integration Test คือทดสอบการใช้งานที่เกิดจากทำงานร่วมกันของคลาสและเมท็อด
- System Test คือทดสอบระบบโดยรวมทั้งหมด

4.1 แผนการทดสอบ (Test Plan)

ตารางที่ 4.1 ตารางแผนการทดลอง

ลำดับ	รายการ
1	Unit Test • ทดสอบการ Encode Header • ทดสอบการ Encode Length • ทดสอบการ Encode Data • ทดสอบการ Encapsulate Data • ทดสอบการ Service • ทดสอบการส่งที่ Layer 5 • ทดสอบการรับที่ Layer 5
2	Integration Test • ทดสอบการส่ง Request • ทดสอบการตอบกลับ Response
3	System Test • ทดสอบส่งภายในเครื่อง • ทดสอบส่งภายในเครือข่าย

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

4.1.1 Unit Test

จุดประสงค์ของการทดสอบ คือเพื่อตรวจสอบผลการทำงานของมอดูลย่อยทั้งหมดของระบบว่าทำงานได้อย่างถูกต้อง

การถอดข้อความ PDU Message ซึ่งจากทฤษฎีในการถอดข้อความในบทที่ 2 Message จะมีอยู่ 3 ส่วนคือ Header, Length และ Data ซึ่ง Header จะได้ค่ามาจากชนิดของ Data type ของโพรโทคอล Length จะได้ค่ามาจากความยาวของข้อมูล และ Data คือ ข้อมูล

หน้าที่การทำงานของเมท็อด

- Headertype ทำหน้าที่บอกว่าข้อความนี้เป็นชนิดข้อมูลประเภทใดใน Type ของโพรโทคอล
- Lengofdata ทำหน้าที่บอกว่าข้อความนี้มีความยาวข้อมูลเท่าไร
- CharDataEncode ทำหน้าที่บอกว่าข้อมูลคืออะไร
- PDUEncapsuting ทำหน้าที่ Encapsulate PDU Message
- MessageEncapsutingII ทำหน้าที่ Encapsulate SNMP Message (Version, PDUtype, PDU)
- MessageEncapsutingIII ทำหน้าที่ Encapsulate SNMP Message (Header, Version, PDUType, PDU)
- MsgL5binary ทำหน้าที่แปลข้อความเป็นเลขฐานสอง
- NextLayer ทำหน้าที่ระบุระดับชั้นที่ทำการส่งต่อไป
- Send ทำหน้าที่ส่งข้อความ
- Receive ทำหน้าที่รับข้อความ
- SnmpServiceCheck ทำหน้าที่คอยตรวจ Community String ซึ่งเหมือนกับการคอยตรวจสอบรหัสผ่าน
- SnmpServiceResponse ทำหน้าที่คอยตรวจสอบ Input ว่าเป็นคำใด แล้วจะแสดงกลับด้วยข้อความที่สอดคล้อง

ในส่วนของการทดสอบคลาส ซึ่งโดยใช้ JUnit เป็นเครื่องมือช่วยในการทดสอบ โดยตั้งชื่อมีคำว่า test นำหน้า Method ที่เราทำการทดสอบ และจะมีคำว่า test ตามหลังถ้าเป็นการทดสอบคลาสเพื่อให้เป็นการสอดคล้องกับเครื่องมือที่ใช้ ตัวอย่างเช่น คลาส L5_SNMP เป็น L5_SNMPTest และ Method Headertype เป็น testHeadertype

1525972
ว5.
ร521ว
2553

ตารางที่ 4.2 ผลการทดสอบจาก Unit Test จาก JUnit

Class	Method	Input	Expected Result
Encode_L5Test	testHeadertype	showos	30
Encode_L5Test	testLengofdata	showos	06
Encode_L5Test	testCharDataEncode	showos	73686f776f73
MessageL5EncoderTest	PDUEncapsuting	showos	0201010201000201003006736 86f776f73
MessageL5EncoderTest	testMessageEncapsutingII	Version = 1 Community = y PDUType = GetRequest Varbindlist = showos	020131040179A011020101020100020100300673686f776f73
MessageL5EncoderTest	testMessageEncapsutingIII	Version = 1 Community = y PDUType = GetRequest Varbindlist = showos	3019020131040179A01102010102010 0020100300673686f776f73
MessageL5EncoderTest	testMsgL5binary	11	00010001
L5_SNMPTest	testNextLayer	L4 = TCP OutputUnit =TextGUI	0
L5_SNMPTest	testSend	Version = 1 Community = y PDUType = GetRequest Varbindlist = showos	3019020131040179A01102010102010 0020100300673686f776f73
L5_SNMPTest	testReceive	3019020131040179A011020101020100020100300673686f776f73	Version = 1 Community = y PDUType = GetRequest Varbindlist = showos
SnmpServiceCheckTest	testaction	y	"true"

ตารางที่ 4.2 (ต่อ) ผลการทดสอบจาก Unit Test จาก JUnit

Class	Method	Input	Expected Result
SnmpServiceResponse Test	testaction	showos	Window7
SnmpServiceResponse Test	testaction	username	Ei8ht
SnmpServiceResponse Test	testaction	showip	127.0.0.1

4.1.2 Integration Test

จุดประสงค์ของการทดสอบ คือ เพื่อตรวจสอบความถูกต้องของ Function การทำงานต่างๆ เมื่อมีการ Integrate unit / module เข้าร่วมกัน

ตารางที่ 4.3 ผลการทดสอบจาก Integration Test จากการ Run Program

Function	Input	Expected Result	Test Result
Request	Version = 1 Community = y PDUType = GetRequest Varbindlist = showos	Agent สามารถรับข้อความที่ Manager ส่งไปได้	รูปที่ 4.1 -4.4
Response	Version = 1 Community = y PDUType = GetRequest Varbindlist = showos	Manager สามารถรับข้อความที่ Agent สร้างและตอบกลับไปได้	รูปที่ 4.5-4.10

4.1.2.1 ทดสอบการส่ง Request

```
########################## L5 SNMP Send      ##########################
SNMP Packet : 30190201310401 79A011020101020100020100300673686f776f73
########################## L4 TCP Send       ##########################
########################## L3 IPv4 Send      ##########################
L3 Send First Package
########################## L2 LLC Ethernet Send  ##########################
########################## L2 Ethernet Send      ##########################
########################## L1 CAT5 Send          ##########################
########################## L1 UTP Send           ##########################
L3 Send Second Package
########################## L2 LLC Ethernet Send  ##########################
########################## L2 Ethernet Send      ##########################
########################## L1 CAT5 Send          ##########################
```

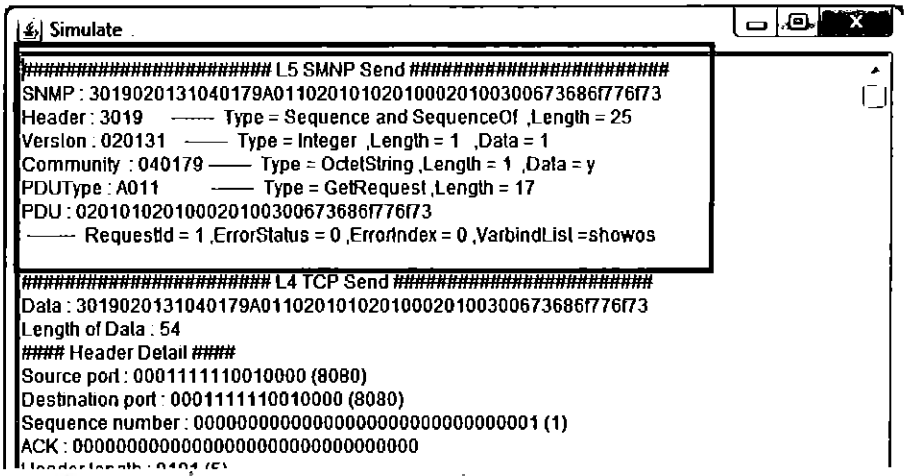
รูปที่ 4.1 Console ของ Manager ที่ส่งข้อมูลไปในแต่ละระกับชั้น

หลังจากที่ได้เห็นข้อมูลที่ส่งจาก Manager ในระดับชั้นที่ 5 แล้วจะเห็นว่า**ข้อความเดียวกันนี้ ถูก**รับโดย Agent ซึ่งอยู่ในรูป 4.2

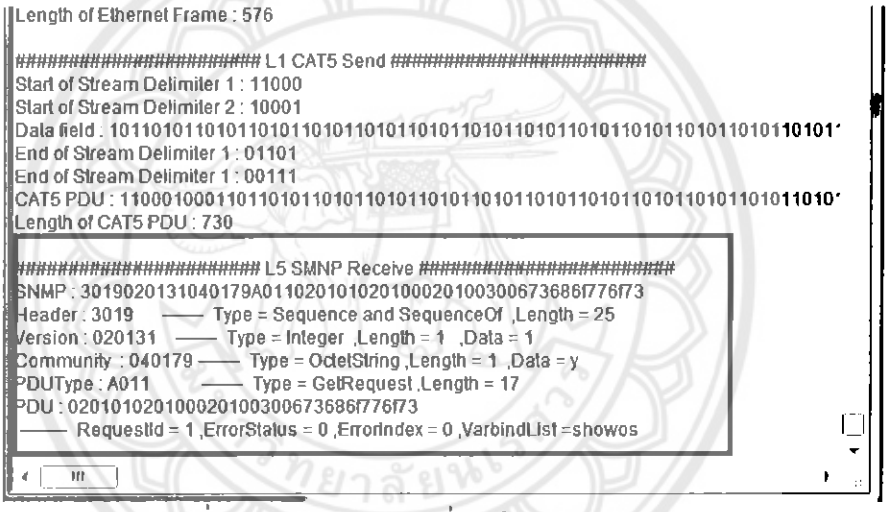
```
L3 Send Second Package
########################## L2 LLC Ethernet Send  ##########################
########################## L2 Ethernet Send      ##########################
########################## L1 CAT5 Send          ##########################
########################## L1 UTP Send           ##########################
##############################################################################
########################## L4 Receive finish     ##########################
##############################################################################

------------------------------------------------------------------------------
---------------------- Receive All Data : 30190201310401 79A011020101020100020 1
------------------------------------------------------------------------------
########################## L5 SNMP Receive   ##########################
L5 Receive : 30190201310401 79A011020101020100020100300673686f776f73
```

รูปที่ 4.2 Console ของ Agent ในการรับข้อมูลไปในแต่ละระดับชั้น

จะเห็นว่าในรูป 4.2 Agent สามารถรับข้อความที่ Manager ส่งมาได้โดยการรับส่งข้อความจะทำงานผ่านชุดโพรโทคอลสแตกที่มีอยู่แล้ว[6,7]



รูปที่ 4.3 GUI ของ Manager ที่ส่งข้อความไปให้กับ Agent



รูปที่ 4.4 GUI ของ Agent ที่รับข้อความจาก Manager

โดยใส่ค่าอินพุตดังนี้

- Version เท่ากับ 1
- Community เท่ากับ y
- PDUType เท่ากับ GetRequest
- Varbind เท่ากับ showos

- เข้ารหัส Version เท่ากับ 1

ตารางที่ 4.4 การเข้ารหัส Version

Structure / Version	Simple Data Type			
	Headertype	LengofData	CharDataEncode	PDUEncapsuting
1	02	01	31	020131

- เข้ารหัส Community เท่ากับ y

ตารางที่ 4.5 การเข้ารหัส Community String

Structure / Community	Simple Data Type			
	Headertype	LengofData	CharDataEncode	PDUEncapsuting
1	40	01	79	040179

- เข้ารหัส Varbind เท่ากับ showos

ตารางที่ 4.6 การเข้ารหัส Varbind

Structure / Varbind	Simple Data Type			
	Headertype	LengofData	CharDataEncode	PDUEncapsuting
showos	30	06	73686f776f73	300673686f776f73

- เข้ารหัส RequestID เท่ากับ 1 (default)

ตารางที่ 4.7 การเข้ารหัส RequestID

Structure / Varbind	Simple Data Type			
	Headertype	LengofData	CharDataEncode	PDUEncapsuting
RequestID	02	01	01	020101

- เข้ารหัส ErrorStatus เท่ากับ 0 (no error)

ตารางที่ 4.8 การเข้ารหัส ErrorStatus

Structure / Varbind	Simple Data Type			
	Headertype	LengofData	CharDataEncode	PDUEncapsuting
ErrorStatus	02	01	00	020100

- เข้ารหัส ErrorIndex เท่ากับ 0 (no error)

ตารางที่ 4.9 การเข้ารหัส ErrorIndex

Structure / Varbind	Simple Data Type			
	Headertype	LengofData	CharDataEncode	PDUEncapsuting
ErrorIndex	02	01	00	020100

เมื่อเข้ารหัสทั้ง 6 ส่วนที่เป็น Simple Data Type แล้วจะได้ Message คือ

ตารางที่ 4.10 ตารางโครงสร้างข้อความ

Header	version	Community	PDUType	PDU			
				Request ID	ErrorStatus	ErrorIndex	Varbind
-	020131	040179	-	020101	020100	020100	3006736 86f776f7 3

แล้วเข้ารหัสอีก 2 ส่วนที่เหลือ(Header, PDUType) ที่เป็นโครงสร้างแบบ Structured Data Type

- เข้ารหัส PDUType เท่ากับ GetRequest

ตารางที่ 4.11 การเข้ารหัส PDUType

Structure / PDUType	Structured Data Type		
	Headertype	LengofData	testMessageEncapsutingII
GetRequest	A0	11	A011020101020100020100300673686f776f73

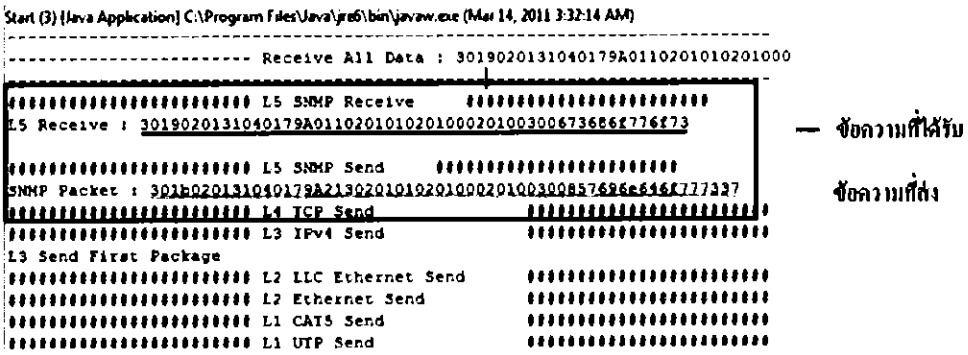
- เข้ารหัส Header ของข้อความ

ตารางที่ 4.12 การเข้ารหัส Header

Structure / Header	Structured Data Type		
	Headertype	LengofData	testMessageEncapsutingIII
Header	30	19	30190201310401 79A0110201010201000 20100300673686f776f73

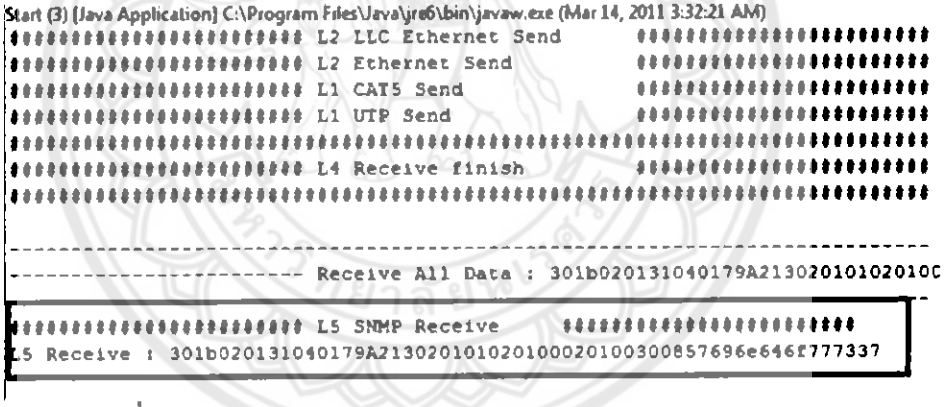
จะได้ข้อความที่ส่งคือ 30190201310401 79A0110201010201000201003006736 86f776f73 คือ Version เท่ากับ 1, CommunityString เท่ากับ y, PDUType เท่ากับ GetRequest และVarbind เท่ากับ showos

4.1.2.2 ทดสอบการตอบกลับ Response



รูปที่ 4.5 Console ของ Agent ในการตอบกลับกลับไปยัง Manager

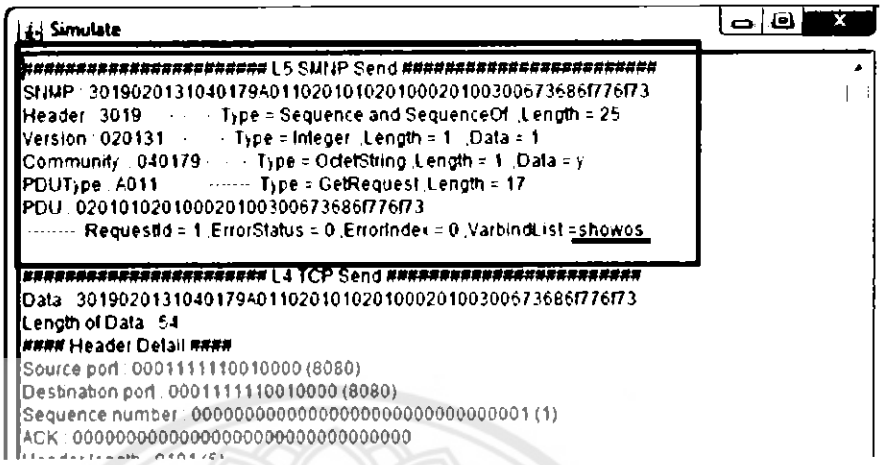
จะเห็นว่าหลังจาก Agent ได้รับข้อความมาแล้ว จะทำการเปลี่ยนข้อความโดยใช้คลาส SnmpServiceRespond และเมทอด action เพื่อให้เกิดข้อความชุดใหม่ เพื่อตอบกลับไปยังฝั่งของ Manager



รูปที่ 4.6 Console ของ Manager ในการรับข้อความตอบกลับจาก Agent

จะเห็นว่าได้รับข้อมูลได้ตรงตามที่ข้อความที่ Agent ส่งกลับไปยัง Manager

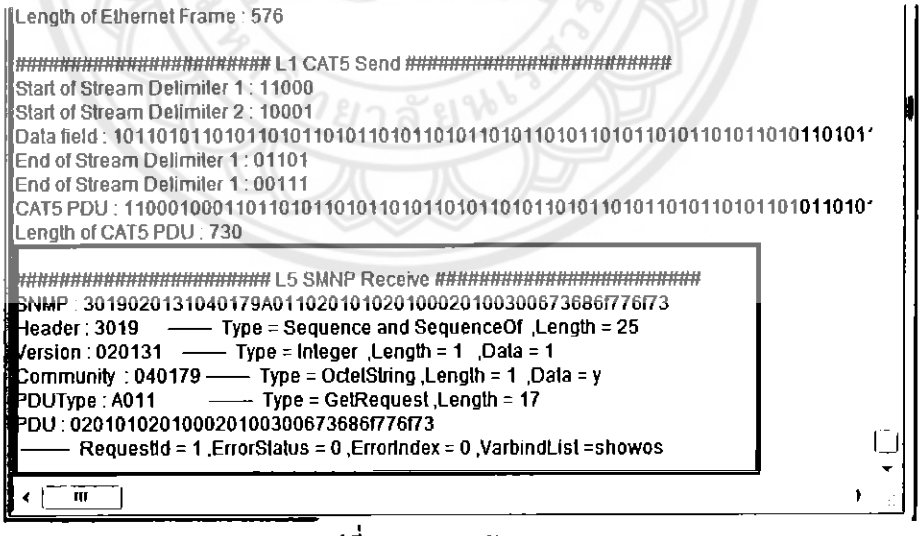
- **Manager Send (Request)**



รูปที่ 4.7 GUI ของ Manager ส่ง Request

Manager ส่งข้อมูลที่เข้ารหัสโดยใช้ Method ที่เห็นในช่วงของ Unit Test โดยส่งคำว่า showos ไปยัง Agent

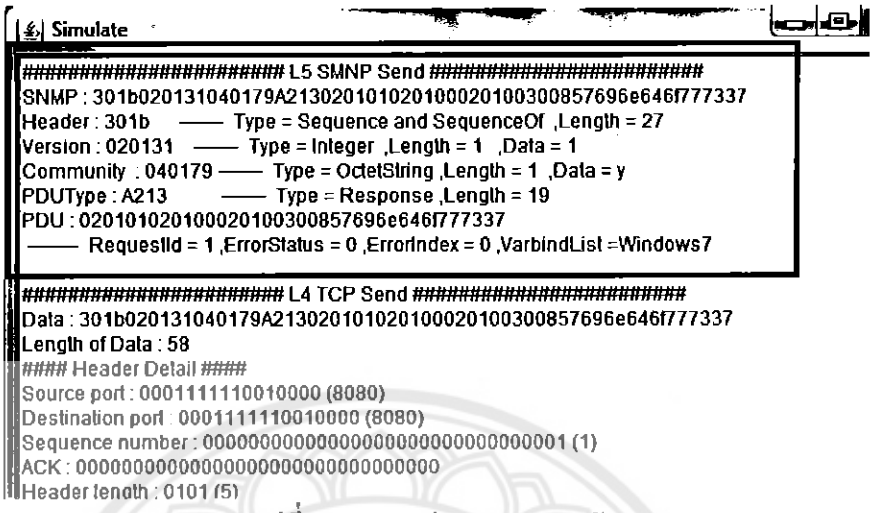
Agent Receive (Request)



รูปที่ 4.8 Agent รับ Request

Agent ได้รับข้อความแล้วนำไปถอดรหัสต่าง ซึ่งจะได้ข้อความเหมือนกับในรูปที่ 4.7

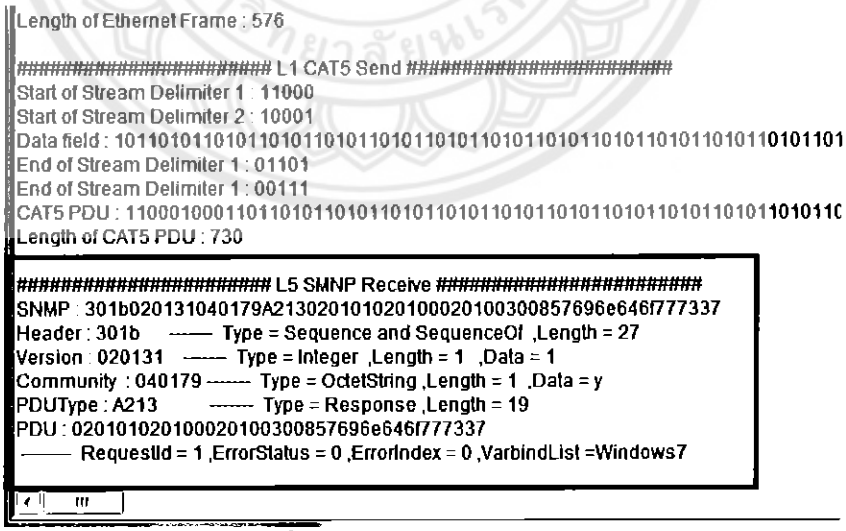
- **Agent Send (Response)**



รูปที่ 4.9 Agent ส่ง Response กลับ

หลังจาก Agent ได้รับข้อความ จะทำการส่งข้อความใหม่ใช้คลาส SnmpServiceResponse และเมท็อด action เพื่อให้เกิดข้อความชุดใหม่ เพื่อตอบกลับไปยังฝั่งของ Manager โดยในตัวอย่างส่งคำว่า "Windows7" กลับไป

- **Manager Receive (Response)**



รูปที่ 4.10 Manager รับ Response

จะเห็นว่า Manager ในรูปนี้จะได้รับข้อความตามที่ต้องการเหมือนกันกับ Agent ที่ส่งมาแล้ว

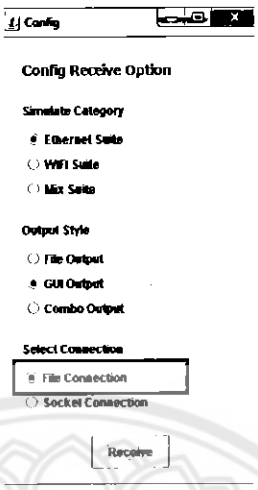
4.1.3 System Test

จุดประสงค์ของการทดสอบ คือเพื่อตรวจสอบระบบว่าระบบทำงานได้ถูกต้องและได้ผลลัพธ์ตรงตามวัตถุประสงค์และขอบเขตของโครงการ

ตารางที่ 4.13 แสดงผลการทดสอบจาก System Test จากการ Run

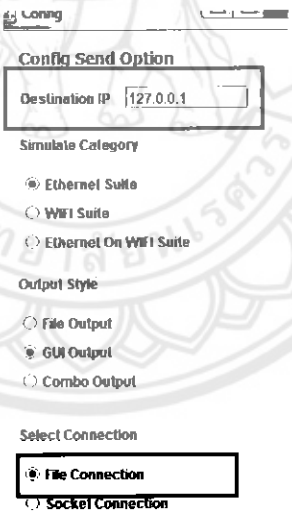
Connection	Input	Expected Result	Test Result
Loopback	Version = 1 Community = y PDUType = GetRequest Varbindlist = showos	สามารถทำงานบน IP 127.0.0.1	รูปที่ 4.11 - 4.14
LAN	Version = 1 Community = y PDUType = GetRequest Varbindlist = showos	สามารถทำงานบน Local Area Network เดียวกันได้	รูปที่ 4.15 - 4.20

4.1.3.1 ทดสอบส่งภายในเครื่อง

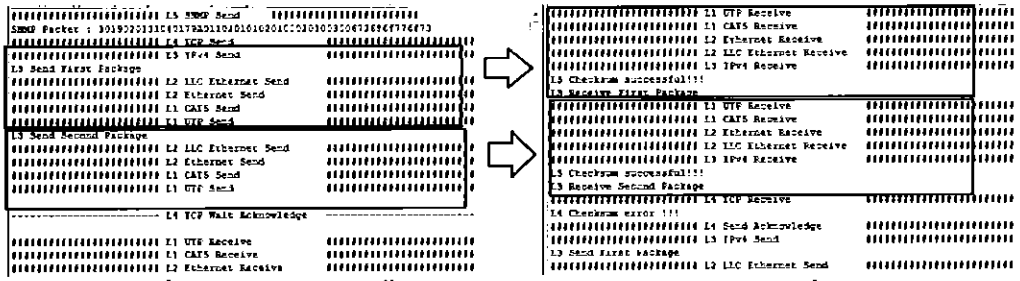


รูปที่ 4.11 GUI ของ Agent เลือก File Connection

เลือก File Connection จะทำการเขียนข้อมูลลงไฟล์แล้วอ่านข้อมูลผ่านไฟล์ในการส่งข้อความกัน



รูปที่ 4.12 GUI ของ Manager เลือก File Connection และระบุ IP Address ของ Agent



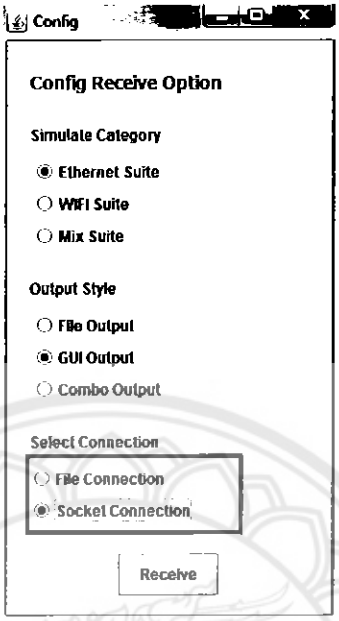
รูปที่ 4.13 Console ของทั้ง Manager และ Agent โต้ตอบกับในเครื่องเดียวกัน

Console ฝั่งซ้ายจะเป็นของ Sender และ Console ฝั่งขวาจะเป็นของ Receiver ซึ่งจะเห็นการรับส่งในแต่ละชั้น ตัวอย่างเช่น Layer3 ทำการส่ง first Package และ Second Package ไปให้ฝั่ง Receiver

```
########################## L3 IPv4 Send ##############################
Version : 0100 (4)
Header Length : 0101 (5)
DS : 00000000
TotalLength : 0000000011101000 (232)
Identification : 0000000000000011 (3)
Flag : 010
Fragmentation : 0000001001000
TTL : 00000000
Protocal : 00000110
Checksum : 0100110000001110
SourceIP : 11000000101010000000000100000011 ()
DestinationIP : 01111111000000000000000000000001 (127.0.0.1)
Option :
Layer 3 packet Send to L2 : 00011111100100000001111110010000000000000000000
Length of Packet : 200
```

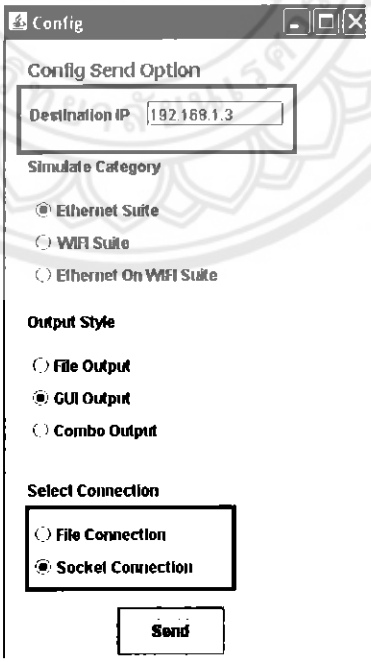
รูปที่ 4.14 GUI ของ Manager ที่ส่งข้อความไปยัง IP 127.0.0.1

4.1.3.2 การทดสอบภายในเครือข่าย

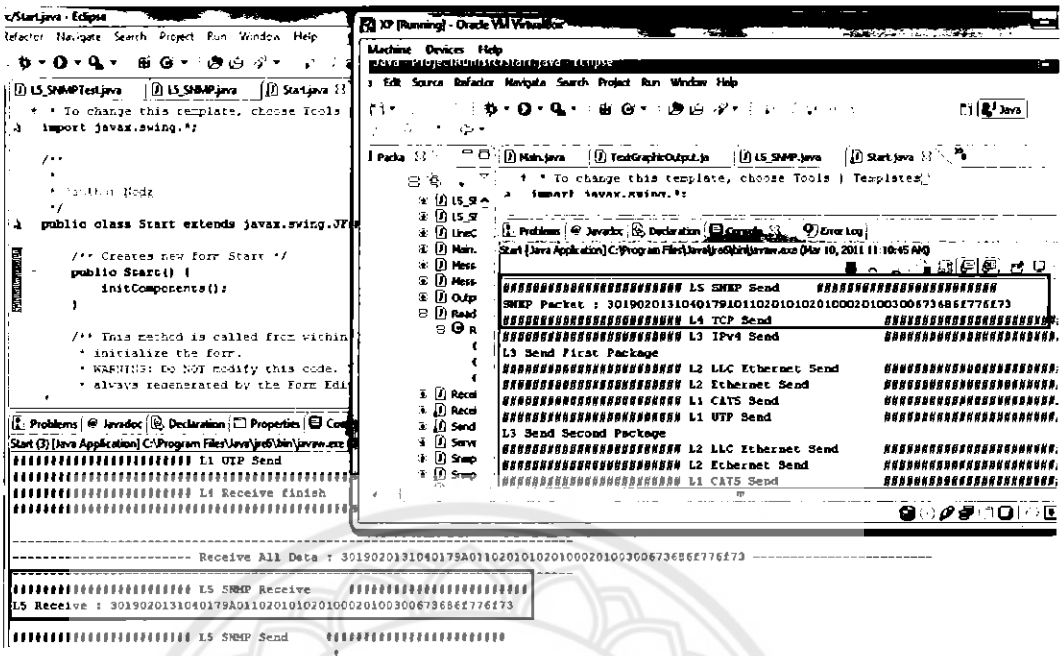


รูปที่ 4.15 GUI ของ Agent เลือก Socket Connection

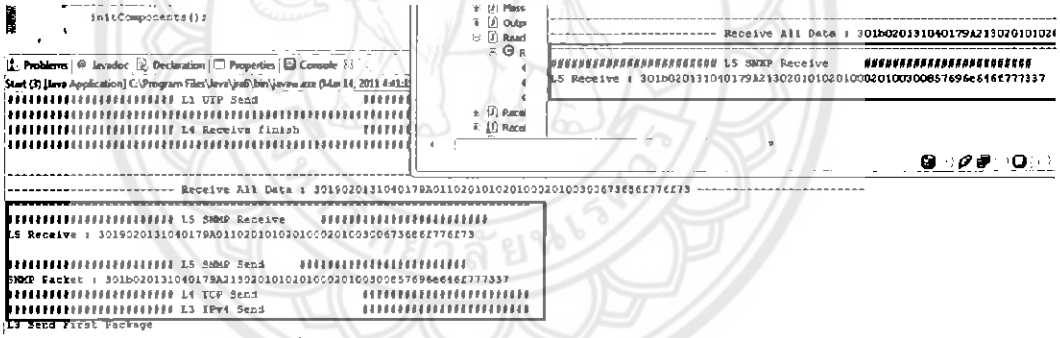
เลือก Socket Connection จะทำการเขียนรับส่งข้อมูลผ่าน Socket



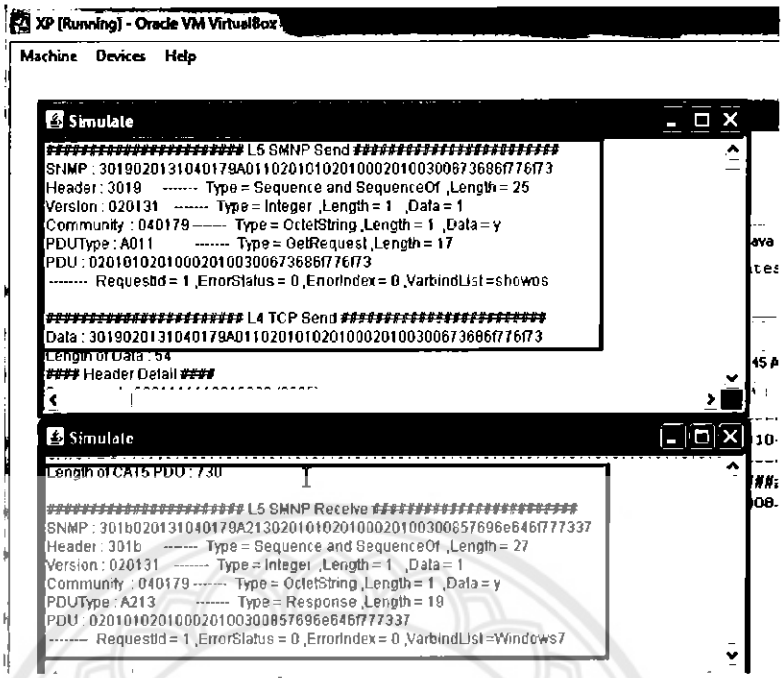
รูปที่ 4.16 GUI ของ Agent เลือก Socket Connection และระบุ IP ในวง LAN



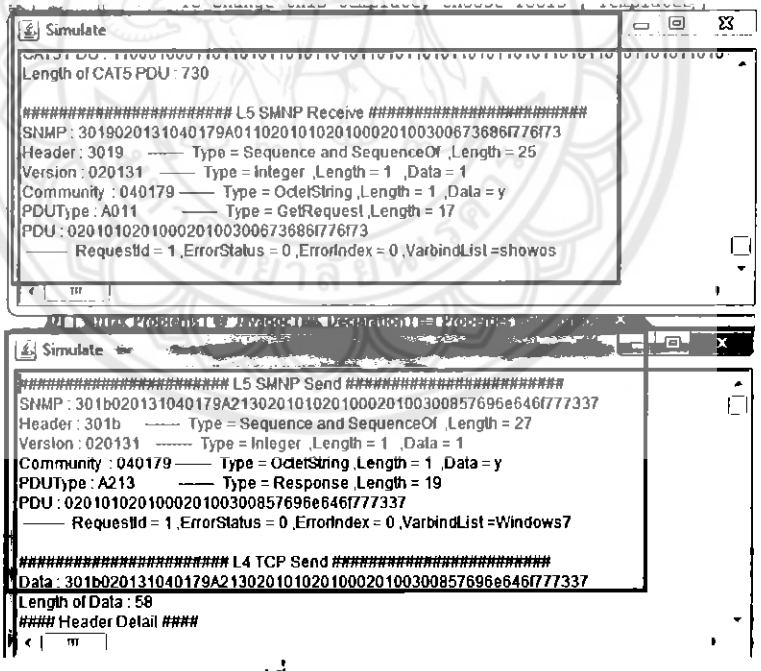
รูปที่ 4.17 การส่งข้อความผ่านเครือข่าย LAN



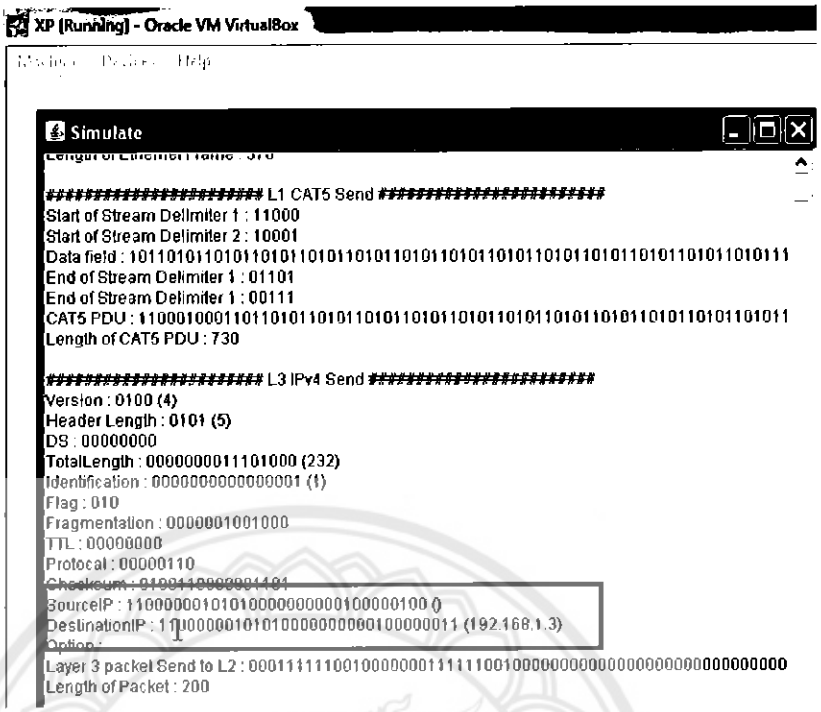
รูปที่ 4.18 การตอบกลับข้อความผ่านเครือข่าย LAN



รูปที่ 4.19 GUI ของ Manager



รูปที่ 4.20 GUI ของ Agent



รูปที่ 4.21 GUI ของ Manager ที่ส่งข้อความไปยัง IP 192.168.1.3

4.2 สรุปผลการทดลอง

ตารางที่ 4.14 สรุปผลการทดลอง

ลำดับ	รายการ	ผลการทดสอบ
1	Unit Test ● ทดสอบการ Encode Header ● ทดสอบการ Encode Length ● ทดสอบการ Encode Data ● ทดสอบการ Encapsulate Data ● ทดสอบการ Service ● ทดสอบการส่งที่ Layer 5 ● ทดสอบการรับที่ Layer 5	 ✓ ✓ ✓ ✓ ✓ ✓ ✓
2	Integration Test ● ทดสอบการส่ง Request ● ทดสอบการตอบกลับ Response	 ✓ ✓
3	System Test ● ทดสอบส่งภายในเครื่อง ● ทดสอบส่งภายในเครือข่าย	 ✓ ✓

บทที่ 5
สรุปผล

โครงงานนี้ได้ศึกษาและพัฒนาไลบรารีเพิ่มเติมจากโครงงานเดิม[6,7] โดยใช้ภาษา Java ในการพัฒนา ซึ่งได้สร้างตัวอย่างสำหรับการจำลองการทำงานของทั้ง 5 ระดับชั้น (The five-layer TCP/IP model) เพื่อให้เห็นภาพการทำงานของระบบเครือข่ายครบทุกระดับชั้น ซึ่งสามารถจำลองผ่านส่วนต่อผสาน (Interface) ภายในเครื่องตนเอง (Loopback) หรือผ่านเครือข่าย (LAN) และนอกจากนี้ผู้ใช้ยังสามารถนำโครงร่างนี้ไปพัฒนาโพรโทคอลอื่นๆ ต่อได้ไม่ว่าจะเป็นระดับชั้นไหนก็ตาม ซึ่งสามารถช่วยให้การจำลองเครือข่ายโพรโทคอลสแตกมีความหลากหลายมากยิ่งขึ้น

5.1 ผลการทดลอง

การทดลองแรกเริ่มจากการทดลองส่งข้อความธรรมดาจากระดับชั้นที่ 5 ไปให้กับระดับชั้นที่ 4 เพื่อเป็นการทดสอบว่าสามารถส่งต่อกันได้ ซึ่งการทดลองในส่วนของ Unit Test ในคลาส L5_SNMP แสดงให้เห็นว่าสามารถส่งข้อความหากันได้

การทดลองที่สองทำการทดลองถอดข้อความตามโพรโทคอล Simple Network Management Protocol (SNMP) ซึ่งจาก Unit Test ในคลาส Encode_L5Test และ MessageL5 EncoderTest แสดงว่าการถอดข้อความตามโพรโทคอล SNMP นั้นทำได้อย่างถูกต้อง

การทดลองที่สามนำสองการทดลองแรกมารวมกันนั้นสามารถแสดงได้ว่าสามารถส่งข้อความที่ถอดตามโพรโทคอล SNMP ไปให้กับระดับชั้นที่ 4 ได้ตามการทดลองในส่วน Integration Test

การทดลองที่สี่การส่งข้อความตอบกลับ ซึ่งใน Unit Test คลาส SnmpServiceCheckTest และ SnmpServiceResponseTest สามารถแสดงถึงการตรวจสอบข้อมูลที่ส่งแล้วสามารถสร้างคำตอบให้เหมาะสมได้แบ่งออกเป็น 4 กรณี คือ

1. กรณีที่ใส่ CommunityString ถูกต้องและ Varbind ถูกต้อง
2. กรณีที่ใส่ CommunityString ถูกต้องและ Varbind ผิด
3. กรณีที่ใส่ CommunityString ผิดและ Varbind ถูกต้อง
4. กรณีที่ใส่ CommunityString ผิดและ Varbind ผิด

ผลจากกรณีแรกจะได้ว่า Agent ตอบกลับค่าที่ Manager ต้องการตัวอย่างเช่น username คือ Ei8ht หรือ showos จะได้ค่าคือ window7

ผลจากกรณีที่สองคือจะได้ว่า Agent ตอบกลับว่า false เนื่องจาก Agent ไม่มี Service นี้
ผลจากกรณีที่สามคือ WrongCommunityString เนื่องจาก CommunityString ไม่ถูกต้อง
ผลจากกรณีที่สี่คือ WrongCommunityString เนื่องจาก CommunityString ไม่ถูกต้อง

กล่าวโดยสรุป โปรแกรมสามารถแสดงการทำงานในระดับชั้นที่ 5 ที่สามารถตอบกลับข้อความของฝั่ง Agent ไปยัง Manager ได้ซึ่งหมายถึงอุปกรณ์ในเครือข่ายทั่วไปตอบกลับสถานะไปยังซอฟต์แวร์จัดการอุปกรณ์ได้ ซึ่งเป็นหลักการทำงานของโพรโทคอลนี้

5.2 ปัญหาและอุปสรรค

ตารางที่ 5.1 ตารางปัญหาและอุปสรรค

ปัญหาและอุปสรรค	แนวทางแก้ไข
1. ขาดความเข้าใจในการเชื่อมต่อกับไลบรารีจากโครงงานเดิม[6,7]	1. เน้นศึกษาเฉพาะส่วนที่ต้องทำงานร่วมกัน 2. เน้นศึกษาโครงสร้างจาก Class diagram ของโครงการ
2. ขาดความชำนาญในการเขียนภาษา Java ทำให้เกิดความล่าช้า	1. ศึกษาการเขียนภาษา Java เฉพาะทักษะที่เกี่ยวข้องกับโครงการ 2. หาตัวอย่างการเขียนโปรแกรมที่เกี่ยวข้องจากอินเทอร์เน็ต
3. การศึกษาหาข้อมูลเกี่ยวกับโพรโทคอล SNMP หาได้ยาก และ เนื้อหาที่ได้จากแต่ละแหล่งน้อยและส่วนใหญ่เป็นภาษาต่างประเทศ	1. พยายามค้นหาจากแหล่งข้อมูลที่เกี่ยวกับของโพรโทคอลเช่น Power Point ที่เป็นภาษาไทย
4. มีปัญหาในการที่จะเขียนโปรแกรมให้เชื่อมต่อกับของเดิม ทำให้ต้องมีการเข้าไปแก้ในตัวโปรแกรมเดิมบ้าง	1. ไม่ไปแก้ไขของเดิม แต่ให้สร้างคลาสใหม่ที่มีอินเทอร์เฟสเหมือนคลาสเดิมแทน
5. ใช้เวลาในการดำเนินการ (RUN) นานมาก เนื่องจากข้อมูลในการนำเข้ามีความยาวมาก ทำให้การทดสอบแต่ละครั้งใช้เวลานานมาก	1. เน้นการทำ Unit Test และ Integration Test ก่อนทำ System Test

5.3 ข้อเสนอแนะ

ตารางที่ 5.2 ตารางข้อเสนอแนะ

ผู้ใช้โปรแกรม	ผู้ที่จะพัฒนาต่อยอด
1. ควรมีความรู้พื้นฐานเกี่ยวกับโพรโทคอลสแตก TCP/IP Model และโพรโทคอล SNMP ซึ่งอาจหาข้อมูลได้จากแหล่งต่อไปนี้ - http://www.thaicert.org/paper/basic/tcp-ip.php	1. ผู้พัฒนาต่อควรมีทักษะในการเขียนโปรแกรมด้วยภาษาจาวา ที่เกี่ยวกับ Encapsulation & Class และ Inheritance & Polymorphism และ Graphic User Interface
2. ควรมีความรู้พื้นฐานการใช้งานโปรแกรมภาษา Java	2. ผู้พัฒนาต่อควรเขียนโปรแกรมให้ไปรบกวนกับไลบรารีน้อยที่สุด โดยใช้เทคนิคจำพวก Design Pattern
	3. หากจะพัฒนาโพรโทคอลเพิ่มหรือเพิ่มความสามารถของไลบรารีนี้อาจจะขอคำปรึกษาจากอาจารย์ที่ปรึกษาหรืออาจารย์ท่านอื่นๆในภาควิชาหรือคณะผู้จัดทำได้

5.4 แนวทางในการพัฒนาต่อยอด

1. เพิ่มโพรโทคอลใหม่ๆ ที่สนใจ (Add Protocol)
2. เพิ่มความสามารถให้สามารถรองรับหลายๆไคลเอนต์ได้ (Multiple Server)
3. เพิ่มความสามารถให้จำลองบนเว็บ (On Web)
4. เพิ่มความสามารถให้จำลองบนมือถือ (On Mobile)
5. เพิ่มความสามารถให้เพิ่มโพรโทคอลผ่าน GUI (GUI Add Protocol)

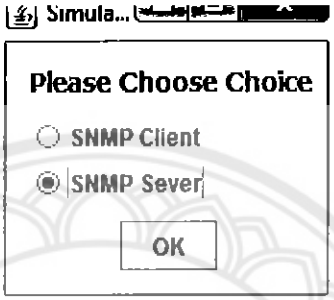
5.5 สรุป

จากผลการทดลองโปรแกรมจำลองการทำงานเครือข่ายของโพรโทคอลระดับ Application Layer โดยใช้ภาษา Java ซึ่งสามารถจำลองการทำงานของระบบเครือข่ายได้ครบถ้วนทั้ง 5 ระดับชั้น (The five-layer TCP/IP model) โดยเรียกใช้ไลบารี่เดิมที่มีอยู่[6,7]และเพิ่มเติมบางส่วน โดยโปรแกรมสามารถจำลองการทำงานในเครื่องตัวเองหรือผ่านเครือได้ และสามารถแสดงการทำงานของโพรโทคอล SNMP ซึ่งจากการทดลองแสดงให้เห็นว่าโปรแกรมสามารถทำงานได้ตามวัตถุประสงค์

ภาคผนวก ก

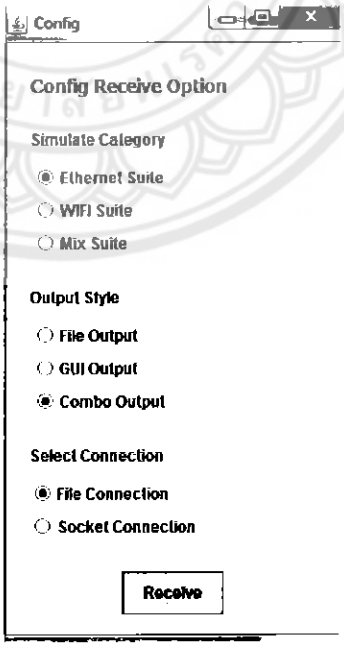
คู่มือการใช้งาน

1. เปิดการทำงานของโปรแกรมโดยการรันไฟล์ชื่อว่า "Start.java" เมื่อรันโปรแกรมขึ้นมาแล้วจะมีหน้าตาดังนี้



รูปที่ ก.1 หน้าต่างเลือก Client หรือ Server

2. ให้เลือกเปิด SNMP Server ก่อนแล้วจะมีหน้าต่างให้ Config สำหรับการรับข้อความ

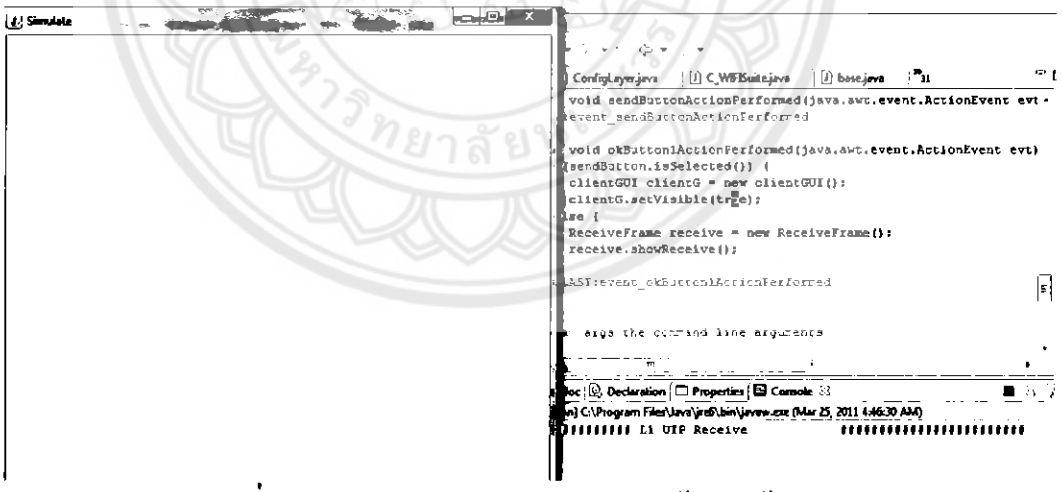


รูปที่ ก.2 หน้าต่างเลือก Client หรือ Server

อธิบายรูปที่ ก.2

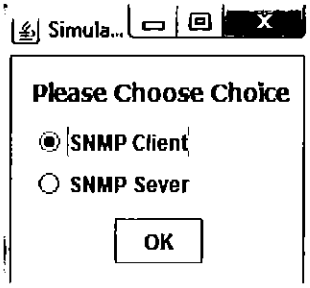
1. ปุ่ม Simulate Category คือเลือกวิธีการส่งว่าจะส่งแบบใด
   - Ethernet Suite คือการส่งผ่านสาย เช่น สาย UTP, สาย Fiber optical เป็นต้น
   - WIFI Suite คือการส่งแบบไร้สาย เช่น เช่น IEEE 802.11 b/g/n
   - Mix Suite คือการส่งทั้ง 2 แบบคือ Ethernet Suite และ WIFI Suite
2. ปุ่ม Simulate Category คือเลือกรูปแบบในการแสดงผล
   - File Output คือให้ผลของการทดสอบออกมาเป็น File
   - GUI Output คือให้ผลของการทดสอบออกมาเป็น Graphic User Interface (GUI)
   - Combo Output คือให้ผลออกทั้ง 2 output คือ File และ GUI
3. ปุ่ม Select Connection คือเลือกรูปแบบทดสอบว่าจะผ่าน File (Loopback) หรือ Socket(Local Area Network หรือ LAN)
   - File Connection คือทดสอบในเครื่องตัวเอง (Loopback)
   - Socket Connection คือทดสอบในเครือข่ายเดียวกัน (LAN)

3. กดปุ่ม Receive จะเห็นว่า มีข้อความขึ้น พร้อมรับข้อความแล้ว



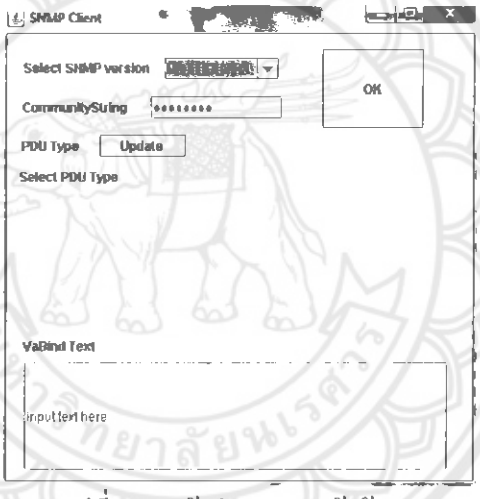
รูปที่ ก.3 GUI และ Console แสดงสำหรับพร้อมรับข้อความ

4. รันการทำงานอีกครั้งหนึ่งเลือกเป็น SNMP Client จะมี User interface ดังในรูปที่ ก.4



รูปที่ ก.4 หน้าต่างเลือก Client หรือ Server

5. กด OK แล้วจะมี GUI ดังในรูปที่ ก.5



รูปที่ ก.5 หน้าต่างการนำเข้าข้อมูล

Select SNMP Version คือเลือก Version ของโพรโทคอล SNMP โดย

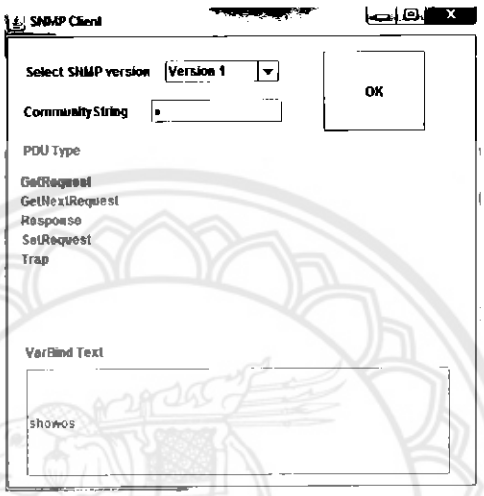
- Version 1 จะมี PDU Type อยู่ 5 ประเภท
- Version 2 จะมี PDU Type อยู่ 7 ประเภท
- Version 3 จะมี PDU Type อยู่ 8 ประเภท

CommunityString คือใส่รหัสผ่านของ Agent

VaBlind Text คือใส่ Object ที่ต้องการทราบ

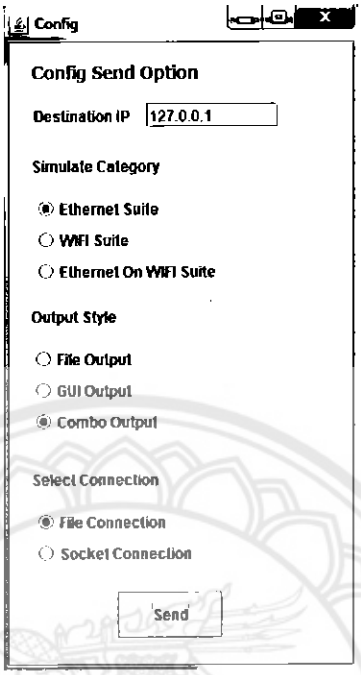
สามารถใส่ Input ดังนี้ (ตัวอย่าง) ซึ่งจะได้ผลดังรูปที่ ก.6

- Version เท่ากับ 1
- Community String เท่ากับ y
- PDUType เท่ากับ GetRequest
- Varbind text เท่ากับ showos



รูปที่ ก.6 แสดงการใส่ข้อมูลของตัวแปร SNMP

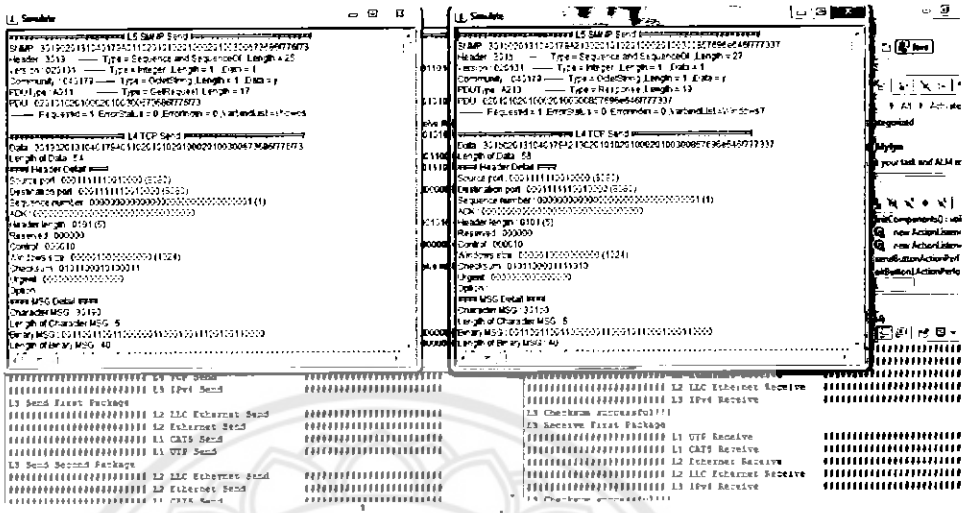
6. กด OK แล้วจะได้ Config สำหรับการส่งข้อความ



รูปที่ ก.7 หน้าต่างการเลือกวิธีการส่งและรูปแบบการแสดงผล

ในช่อง Destination IP คือการระบุ IP Address ของ Agent ที่ต้องการส่ง

7. กด Send และรอดูผลลัพท์



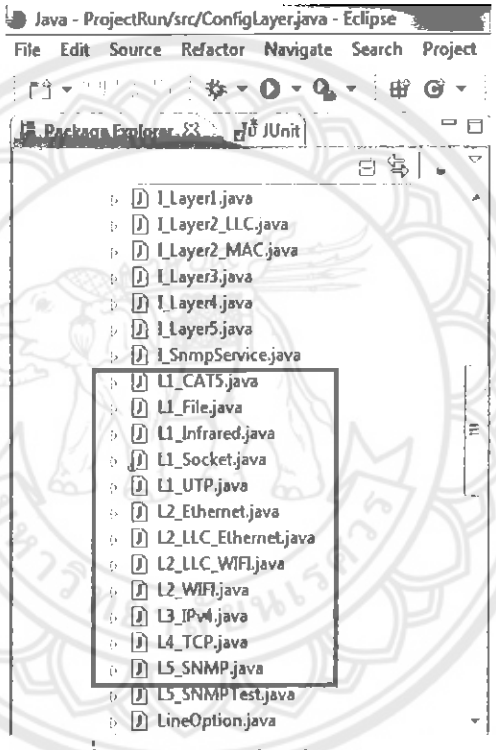
รูปที่ ก.8 ผลลัพท์ GUI

รูปที่ ก.9 ผลลัพท์ File

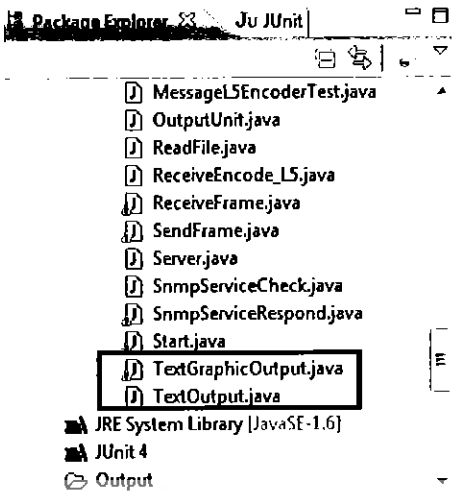
ภาคผนวก ข

คู่มือการพัฒนา

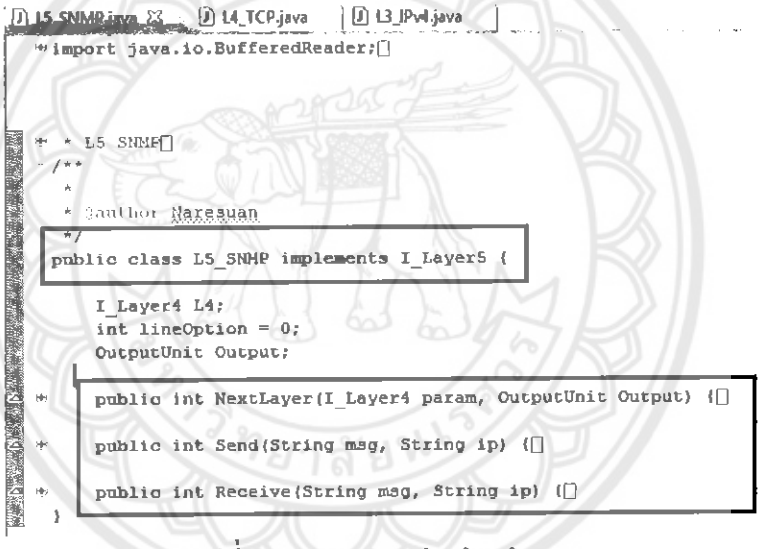
ตามโครงสร้างที่ออกแบบไว้ ซึ่งจะเห็นได้ว่าผู้ที่ต้องการพัฒนาเพิ่มสามารถทำการพัฒนาโพรโทคอลใหม่ๆ ได้ง่ายเพราะการสร้างไลบรารีเป็นในลักษณะ Extract Interface โดยการเพิ่มคลาสของโพรโทคอลที่เราสนใจหรือความสามารถใหม่ๆ



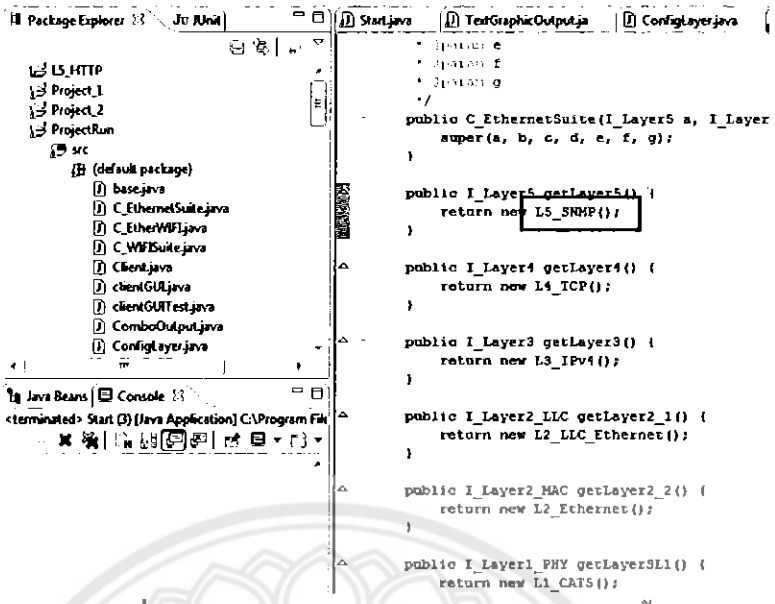
รูปที่ ข.1 คลาสของโพรโทคอลสแตก



รูปที่ ข.2 Output ของไลบรารี



รูปที่ ข.3 คลาสของแต่ละโพรโทคอล



รูปที่ ข.4 Config การใช้โพรโทคอลในแต่ละระดับชั้น

เอกสารอ้างอิง

[1] MaY- Leangsanam. (2011, February) OSI Model. [Online]. http://maymommom.blogspot.com/2011/02/7-osi-model.html

[2] Charles M. Kozierok. (2005, Sep.) The TCP/IP Guide. [Online]. http://www.TCPIPGuide.com

[3] Douglas R.Mauro and Kevin J.Schmidt, *Essential SNMP*, 2nd ed., Darren Kelly, Ed.: O'Really, 2005.

[4] Behrouz A.Forouzan, *Data Communications and Networking*, 4th ed.: McGraw-Hill, 2007.

[5] RaneNote 161. (2005, December) Douglas Bruey, Rane Corporation. [Online]. http://www.rane.com/note161.html

[6] ชัยภัทร จารุชาต และ นทภพ เริงนภาพล, "โปรแกรมจำลองการทำงานสำหรับเน็ตเวิร์คโพรโตคอล TCP/IP," มหาวิทยาลัยนเรศวร, วิทยาพิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ 2551.

[7] กิตติศักดิ์ ตุนาโป่ง และ ณัฐเสฏฐ์ วัชรานุรักษ์, "ไลบรารีสำหรับจำนองเน็ตเวิร์คโพรโตคอลสแตก," มหาวิทยาลัยนเรศวร, วิทยาพิพนธ์วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ 2550.

ประวัติผู้เขียนโครงการ

ชื่อ นายชานนท์ อ่วมเพชร

ภูมิลำเนา 23 หมู่ 10 ต.โคกสลุด อ.บางกระทุ่ม จ.พิษณุโลก 65110

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนพิษณุโลกพิทยาคม จ.พิษณุโลก
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: chanon_champ@hotmail.com

ชื่อ นายทวีสิน ลาภานุวัฒน์

ภูมิลำเนา 812/3 ถ.บรมไตรโลกนารถ2 ต.ในเมือง อ.เมือง จ.พิษณุโลก 65000

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนพิษณุโลกพิทยาคม จ.พิษณุโลก
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: nu_leng@hotmail.com