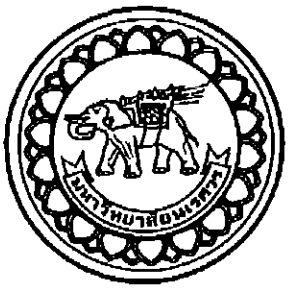


ส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับ ทอพอโลยีของระบบจำลองเครือข่าย

GRAPHIC USER INTERFACE FOR TOPOLOGY OF A NETWORK SIMULATOR

นายรุ่งโรจน์ ตั้งอิสราวุฒิกุล รหัส 49361614

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 30 ก.ย. 2558
เลขทะเบียน 1686288
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร ร 636 ส
2556

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2556

ใบรับรองปริญญานิพนธ์

หัวข้อโครงงาน	**ส่วนต่อประสานกราฟิกกับผู้ใช้ สำหรับ ทอพอโลยีของระบบจำลองเครือข่าย**
ผู้ดำเนินโครงงาน	นายรุ่งโรจน์ ตั้งอิสราวุฒิกุล รหัสนิสิต 49361614
อาจารย์ที่ปรึกษา	ดร. สุรเดช จิตประไพกุลศาล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2556

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....ที่ปรึกษาโครงการ
(ดร. สุรเดช จิตประไพกุลศาล)

.....กรรมการ
(ดร. วรลักษณ์ คงเด่นฟ้า)

.....กรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)

หัวข้อโครงงาน	ส่วนต่อประสานกราฟิกกับผู้ใช้ สำหรับทอพอโลยีของระบบจำลองเครือข่าย	
ผู้ดำเนินโครงงาน	นายรุ่งโรจน์ ตั้งอิสราวุฒิกุล	รหัสนิสิต 49361614
อาจารย์ที่ปรึกษา	ดร. สุรเดช จิตประไพกุลศาล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2556	

.....

บทคัดย่อ

หลักสูตรวิศวกรรมคอมพิวเตอร์มีวิชาเรียนเกี่ยวกับระบบเครือข่ายและการสื่อสารข้อมูล ซึ่งมีเนื้อหาที่ประกอบด้วยโพรโทคอลจำนวนมาก และมีการทำงานที่ซับซ้อน ทางผู้พัฒนาจึงเล็งเห็นว่าควรจะมีสื่อการเรียนการสอนที่จะใช้ประกอบการอธิบายเกี่ยวกับระบบเครือข่ายและการสื่อสารที่ดี สามารถทำให้เห็นรายละเอียดการทำงานของโพรโทคอลในระบบเครือข่าย เพื่อเพิ่มความเข้าใจแก่ผู้ศึกษายิ่งขึ้น

โครงงานนี้เป็นการพัฒนาต่อจากโครงงานโปรแกรมแสดงการทำงานเครือข่าย 4 ชั้นของ OSI โดยเพิ่มเติมดังนี้ เพิ่มและลดโพรโทคอลได้สะดวก มีส่วนแสดงผลที่หลากหลาย สามารถเก็บข้อมูลเพื่อใช้ในการศึกษา เพิ่มและลดส่วนแสดงผลได้สะดวก สามารถจำลองได้มากกว่าสองโหนด และมี GUI ออกแบบทอพอโลยี

สำหรับเครื่องมือที่ใช้ในการพัฒนานั้น โปรแกรมนี้ถูกพัฒนาด้วยภาษาจาวาและใช้เฉพาะไลบรารี่พื้นฐานที่มากับคอมไพเลอร์ JDK 1.6.0_20 โดยใช้โปรแกรมที่เขียนคือไฟล์ Text ทั่วไป ส่วนหลักการในการพัฒนานั้น ผู้พัฒนาได้พัฒนาตัวโปรแกรมตามหลักการของ Object Oriented เพราะวัตถุประสงค์ของโปรแกรมนี้ คือพัฒนาขึ้นเพื่อให้ง่ายต่อการเพิ่มเติมและพัฒนาต่อ

โปรแกรมที่พัฒนาขึ้นสามารถใช้เป็นสื่อในการเรียนการสอนวิชาที่สอนเกี่ยวกับระบบเครือข่ายและการสื่อสารข้อมูลซึ่งจะช่วยเพิ่มความเข้าใจการทำงานของโพรโทคอลได้ดียิ่งขึ้น

Project title	GRAPHIC USER INTERFACE FOR TOPOLOGY OF A NETWORK SIMULATOR
Name	Mr. Rungrode Tangissarawottikol ID. 4931614
Project advisor	Dr. Suradet Jitprapaikulsarn, Ph.D.
Major	Computer Engineering.
Department	Electrical and Computer Engineering.
Academic year	2013

.....

ABSTRACT

Department of Computer Engineering course is an introduction to networking and data communications. The body contains a large number of protocols. And a complex function. The developer recognizes that there should be the medium of instruction is used to describe the network and better communication. Can studing the details of the operation of the protocol in the network. To add even more insight to the study.

This project is developed from the project Network 4 Layers of ISO Operation Program further below. Add and remove protocols easily. Have output variety. To collect data for the study. Add and remove output convenience. Can simulate more than two nodes and have GUI design topology.

For tools used to develop it. This application is developed in Java and uses only libraries that come with JDK 1.6.0_20 compiler. Is written using text files general principles for developing it. The developers have developed a program based on the principles of Object Oriented The purpose of this program. Was developed to make it easier to develop more and more.

Program developed can be used as a medium of instruction taught about networking and data communications, which will help increase the understanding of the protocol's performance is even better.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ฉบับนี้สำเร็จลุล่วงมาได้นั้น เนื่องจากความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงงาน ดร.สุรเดช จิตประไพกุลศาล ที่ท่านให้โอกาสพร้อมทั้งกรุณาสละเวลาให้คำแนะนำในการทำงาน ตลอดจนการตรวจสอบการทำงานพร้อมทั้งเสนอแนวทางการแก้ไขปัญหาตลอดระยะเวลาการทำโครงงาน พร้อมทั้งให้คำแนะนำที่เป็นประโยชน์อย่างมากต่อการทำโครงงาน ทำให้การทำโครงงานเป็นไปอย่างราบรื่น

ทั้งนี้ต้องขอขอบพระคุณกรรมการทั้งสองท่านอันได้แก่ ดร. วรลักษณ์ คงเด่นฟ้า กับอาจารย์ภาณุพงศ์ สอนคม อาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ที่แต่ละท่านสละเวลาอันมีค่าให้คำปรึกษาและแนะแนวทางในการแก้ปัญหาต่างๆ

สุดท้ายนี้ผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา อาจารย์ทุกท่าน ที่คอยสั่งสอนให้ความรู้จนผู้จัดทำสำเร็จการศึกษา และขอขอบคุณพี่ๆน้องๆทุกคนที่ยังไม่ได้เอ่ยนามที่คอยให้กำลังใจ ช่วยให้คำปรึกษา ตลอดจนช่วยแบ่งเบาภาระค่าใช้จ่าย ทั้งในเรื่องส่วนตัวและเรื่องเรียนจนสำเร็จลุล่วงมาได้ด้วยดี

ขออำนาจสิ่งศักดิ์สิทธิ์ทั้งหลายบันดาลให้ บิดา มารดา และอาจารย์ทุกท่าน สุขภาพแข็งแรงและเป็นแรงผลักดันให้คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร มีความก้าวหน้าต่อไป

นายรุ่งโรจน์ ตั้งอิสราวุฒิกุล

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ
บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 2
1.3 ขอบข่ายของโครงงาน 2
1.4 ขั้นตอนการดำเนินงาน 2
1.5 แผนการดำเนินงาน 3
1.6 ผลที่คาดว่าจะได้รับ 4
1.7 งบประมาณที่ใช้ 4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 5
2.1 โพรโทคอลในระบบเครือข่ายการสื่อสารข้อมูล 5
2.2 อุปกรณ์สื่อสารในระบบเครือข่ายการสื่อสารข้อมูล 9
2.3 ทอพอโลยีของระบบเครือข่ายการสื่อสารข้อมูล 11
บทที่ 3 วิธีการดำเนินงานโครงงาน 15
3.1 ศึกษาโครงสร้าง การทำงาน ข้อดี และข้อจำกัดของโปรแกรมเดิม 15
3.2 แก้ไขโครงสร้างเพื่อให้โปรแกรมสะดวกต่อการพัฒนาต่อ 16
3.3 เพิ่มเติมส่วนแสดงผล 19
3.4 ออกแบบ และสร้าง GUI ที่แสดงโหนดแต่ละโหนด 21
3.5 พัฒนาให้โปรแกรมสะดวกต่อการเพิ่มหรือลดส่วนแสดงผล 23
3.6 พัฒนาให้โปรแกรมสะดวกต่อการเพิ่มหรือลดโพรโทคอลแต่ละชั้น 23
3.7 พัฒนาให้โปรแกรมสามารถจำลองได้มากกว่าสองโหนด 23

สารบัญ(ต่อ)

หน้า

3.8 พัฒนาให้โปรแกรมสามารถรอรับข้อมูลได้หลายทิศทาง 25
3.9 เพิ่มส่วน GUI ของทอพอโลยี 27
3.10 ตรวจสอบการใช้งานจริง 29
3.11 จัดทำคู่มือโครงงาน 29
บทที่ 4 ผลการทดลอง 30
4.1 การทดสอบโปรแกรม 30
บทที่ 5 สรุปผลและข้อเสนอแนะ 46
5.1 สรุปผลการทดลอง 47
5.2 ปัญหาและแนวทางแก้ไข 47
5.3 ข้อเสนอแนะ 48
เอกสารอ้างอิง 49
ภาคผนวก ก 50
ประวัติผู้เขียนโครงงาน 55

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงานโครงงานวิศวกรรม....3
5.1 ตารางสรุปผลการทดลอง....47
5.2 ปัญหาและแนวทางแก้ไขที่ได้รับการแก้ไข....47
5.3 ปัญหาและแนวทางแก้ไขที่ยังไม่ได้รับการแก้ไข....48

สารบัญรูป

รูปที่ หน้า

2.1 แบบจำลอง OSI และ แบบจำลอง TCP/IP 8
2.2 การส่งต่อข้อมูลระดับเครือข่าย 8
2.3 อุปกรณ์สื่อสารในระบบเครือข่ายการสื่อสารข้อมูล 9
2.4 ทอพอโลยีแบบบัส 11
2.5 ทอพอโลยีแบบวงแหวน 12
2.6 ทอพอโลยีแบบวงดาว 13
3.1 รูปแสดง GUI ของโปรแกรมเดิม 16
3.2 Interface Output ตอนเริ่มสร้าง 16
3.3 Interface Output ตอนสิ้นสุดการพัฒนา 16
3.4 Interface OutputUnit 17
3.5 แสดงโครงสร้างเดิมของโพรโทคอลแต่ละชั้นที่เชื่อมด้วยฟังค์ชัน nextLayer 17
3.6 แสดงโครงสร้างใหม่ของโพรโทคอลแต่ละชั้นที่เชื่อมด้วยฟังค์ชัน nextLayer 18
3.7 แสดงโครงสร้างกรณีโพรโทคอลชั้นที่2 มีทั้งส่วน LLC และ MAC 18
3.8 แสดงโครงสร้างของโพรโทคอลแต่ละชั้นที่เชื่อมด้วยฟังค์ชัน nextLayer โดยกรณีที่ชั้นที่ 2 ประกอบด้วย LLC และ MACแสดงผล 18
3.9 แสดงโครงสร้างการสืบทอดของโพรโทคอลแต่ละชั้น 19
3.10 แสดงโครงสร้างของส่วนแสดงผล 19
3.11 แสดงโครงสร้างของส่วนแสดงผลทั้งหมด 20
3.12 แสดง GUI ของโหนด ที่ออกแบบไว้ 20
3.13 Flowchart Diagram แสดงการทำงานของปุ่มเริ่มการทำงาน 21
3.14 Flowchart Diagram แสดงการทำงานของ NewConfigLayer 22
3.15 ความสัมพันธ์ของ MyFrame กับ NewConfigLayer 22
3.16 แสดงไลบรารี่ในส่วนของทอพอโลยี 24
3.17 แสดงไลบรารี่ในส่วนของเส้นทาง 24
3.18 Flowchart Diagram แสดงการทำงานของคลาส TopologyFromFile 24
3.19 Flowchart Diagram แสดงการทำงานของคลาส PathFromFile 24
3.20 Diagram แสดงการการติดต่อกันของ GUI กับทอพอโลยีและเส้นทาง 25
3.21 Flowchart Diagram แสดงการทำงานของปุ่มเริ่มทำงาน 26

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.22 Flowchart Diagram แสดงการทำงานของปุ่มใส่ข้อมูลการส่งเดิม....26
3.23 Flowchart Diagram แสดงการทำงานของปุ่มใส่ข้อมูลการส่งใหม่....27
3.24 GUI ของทอพอโลยีที่ออกแบบ....27
3.25 GUI ของทอพอโลยีที่ออกแบบ หลังทำการเพิ่มโหนด A....28
4.1 GUI ของทอพอโลยี....30
4.2 GUI ของทอพอโลยี หลังทำการเพิ่มโหนด....31
4.3 GUI ของทอพอโลยี หลังทำการแก้ไขชื่อโหนดจาก C เป็น K....31
4.4 GUI ของทอพอโลยี หลังทำการ Add NIC....32
4.5 GUI ของทอพอโลยี หลังทำการเพิ่ม Bus....32
4.6 GUI ของทอพอโลยี หลังทำการเพิ่มการติดต่อกับ Bus....33
4.7 แสดงไฟล์ Topology.xml หลังใช้คำสั่ง Build....34
4.8 GUI ของโหนดต่างๆจากคำสั่ง Run....35
4.9 แสดงหน้าต่างเมื่อเลือกคำสั่ง Save....35
4.10 แสดงไฟล์ที่ได้หลังใช้คำสั่ง Save....36
4.11 แสดงข้อมูลในไฟล์....36
4.12 GUI ของทอพอโลยี หลังใช้คำสั่ง New....37
4.13 GUIของทอพอโลยี แสดงทอพอโลยีหลังจากใช้คำสั่ง Open ไฟล์....37
4.14 รูปแสดงการติดต่อของแต่ละโหนด....38
4.15 รูปแสดงโหนดต่างๆ เมื่อเสร็จสิ้นการส่งข้อมูลจาก A ไป J....38
4.16 การแสดงผลทาง Console ของโหนด A....39
4.17 การแสดงผลทาง GUI ของโหนด A....40
4.18 การแสดงผลทาง Frame ของโหนด A....40
4.19 ไฟล์ที่ได้จากการแสดงผลทาง Web From (index.html)....41
4.20 ไฟล์ที่ได้จากการแสดงผลทาง Web From ในรายละเอียดแต่ละส่วน....41
4.21 ข้อมูลในส่วนเริ่มต้นของไฟล์ Output_A....42
4.22 ข้อมูลไฟล์ 15.txt ในโฟลเดอร์ Output_A....43

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.23 ข้อมูลไฟล์ 15.txt ในโฟลเดอร์ Output_A 43

4.24 ข้อความแจ้งเตือนการเพิ่มส่วนแสดงผล NewOutput 44

4.25 GUI ของโหนดที่มีการเพิ่มส่วนแสดงผล NewOutput 44

4.26 ข้อความแจ้งเตือนการเพิ่มโพรโทคอล ProtocolSample 45

4.27 GUI ของโหนดที่ทำการลบส่วนแสดงผล ProtocolSample 45

ก.1 ส่วนประกอบของ GUI ของโหนด 51

ก.2 ส่วนประกอบของ GUI ของทอพอโลยี 52

ก.3 แสดงการเชื่อมโยงของ Interface ของโพรโทคอลแต่ละชั้น 53

ก.4 แสดงการเชื่อมโยงของ ส่วนแสดงผล 54

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในยุคปัจจุบันการส่งข้อมูลในระบบเครือข่ายมีความสำคัญมาก ดังนั้นการศึกษาเกี่ยวกับการส่งข้อมูลระบบเครือข่ายจึงเป็นสิ่งที่หลีกเลี่ยงไม่ได้ ซึ่งการส่งข้อมูลระบบเครือข่ายมีเนื้อหาที่ซับซ้อนประกอบด้วยโพรโทคอลจำนวนมากที่ควรรู้ โดยแต่ละโพรโทคอลมีการทำงานที่แตกต่างกันออกไป ดังนั้นจึงควรมีโปรแกรมสำหรับจำลองการทำงานของระบบเครือข่าย เพื่อเป็นตัวช่วยในการเรียนรู้การส่งข้อมูลในระบบเครือข่าย

โปรแกรมสำหรับจำลองการทำงานของระบบเครือข่ายที่มีอยู่เดิมคือโปรแกรมแสดงการทำงานเครือข่าย 4 ชั้นของ OSI [2] โปรแกรมดังกล่าวสามารถแสดงการทำงานเครือข่าย 4 ชั้นของ OSIสามารถแสดงการส่งข้อมูลระหว่าง 2 เครื่องโดยมีข้อจำกัดดังนี้

1. ส่วนแสดงผลไม่สามารถแสดงการทำงานย้อนหลังได้ และไม่สามารถบันทึกไว้เพื่อใช้ในการศึกษาได้

2. ไม่สะดวกการเพิ่มโพรโทคอลและส่วนแสดงผลใหม่ๆ อันเนื่องมาจากโครงสร้างไม่รองรับ และยังมีฟังค์ชั่นจำนวนมากที่สามารถรวมกันได้

3. ไม่สะดวกการเพิ่มโพรโทคอลและส่วนแสดงผลใหม่ๆ อันเนื่องมาจาก GUI ไม่รองรับ

4. โปรแกรมยังมีโครงสร้าง และหลักการเขียนโปรแกรมที่ควรแก้ไข

อีกทั้งยังมีความคิดที่จะพัฒนาให้โปรแกรมสามารถทำการรับส่งจากเดิมที่ทำได้เพียงสองโหนด คือผู้รับหนึ่งโหนดและผู้ส่งอีกหนึ่งโหนด เป็นมากกว่าสองโหนด ซึ่งจะต้องเพิ่มส่วนของทอพอโลยี (Topology) ซึ่งเป็นตัวกำหนดการเชื่อมต่อกันของแต่ละโหนดและเส้นทาง (Path) ซึ่งเป็นตัวกำหนดเส้นทางข้อมูล

นอกจากนี้ยังมีความคิดที่จะพัฒนาตัว GUI ที่จำลองการเชื่อมต่อของแต่ละโหนด (GUI ของทอพอโลยี) ที่ง่ายต่อการใช้งาน สามารถสร้าง บันทึก โหลด และนำการติดต่อของแต่ละโหนดไปใช้ในโปรแกรมหลัก เพื่อเป็นประโยชน์แก่การออกแบบการติดต่อแต่ละโหนด

ซึ่งทั้งหมดที่กล่าวมานี้เพื่อจะช่วยให้การจำลองระบบเครือข่ายมีความสมบูรณ์ ง่ายต่อการเพิ่มและลดส่วนต่างๆโดยไม่ต้องเสียเวลาในการศึกษาโปรแกรมเดิม และง่ายต่อการใช้งานเพื่อเพิ่มความเข้าใจในระบบเครือข่ายแก่ผู้ใช้งานยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

เพื่อพัฒนาโปรแกรมสำหรับศึกษาการส่งข้อมูลเครือข่ายที่ดี และมีข้อผิดพลาดน้อยลง โดยมีรายละเอียดดังนี้

- โปรแกรมมีโครงสร้างและหลักการเขียนโปรแกรมที่ง่ายต่อการพัฒนา
- โปรแกรมสะดวกต่อการเพิ่ม หรือ ลด โพรโทคอล และส่วนแสดงผล
- โปรแกรมสามารถแสดงการส่งข้อมูลได้มากกว่าสองโหนด
- โปรแกรมสามารถแสดงการเชื่อมต่อของโหนดต่างๆ

1.3 ขอบข่ายของโครงงาน

1. โปรแกรมสามารถจำลองการแสดงผลการทำงานของโพรโทคอลทีซีพี/ไอพีในการส่งข้อมูลมากกว่าสองโหนด
2. โปรแกรมสามารถจำลองการแสดงผลการทำงานของเครือข่ายการส่งต่อข้อมูลผ่านไฟล์
3. GUI สามารถเพิ่มลดส่วนแสดงผลได้ เมื่อทำการเพิ่มหรือลดส่วนแสดงผลอย่างถูกหลัก
4. GUI สามารถเพิ่มลดโพรโทคอลได้เมื่อทำการเพิ่มหรือลดโพรโทคอลอย่างถูกหลัก

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาการทำงานของโปรแกรมที่สร้างมาก่อนหน้านี้
2. ศึกษาความรู้ทั่วไปเกี่ยวกับโพรโทคอล
3. แก้ไขข้อจำกัดของโปรแกรมเดิม
4. สร้างโปรแกรมที่สะดวกต่อการเพิ่มลดโพรโทคอลส่วนแสดงผล
5. ศึกษาการทำงานของระบบเครือข่ายที่มีจำนวนโหนดมากกว่าสองโหนด
6. สร้างโปรแกรมที่สะดวกต่อการจำลองการเชื่อมต่อของโหนด
7. ทดสอบความถูกต้องของโปรแกรม
8. จัดทำเอกสารและคู่มือการใช้

1.5 แผนการดำเนินงาน

ตารางที่1.1 แผนการดำเนินงานโครงงานวิศวกรรม

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. ศึกษาการทำงานของโปรแกรมที่สร้างมาก่อนหน้านี้										
2. ศึกษาความรู้ทั่วไปเกี่ยวกับ โพรโทคอล										
3. แก้ไขข้อจำกัดของโปรแกรมเดิม										
4. สร้างโปรแกรมที่สะดวกต่อการเพิ่มลดโพรโทคอล ส่วนแสดงผล										
5. ศึกษาการทำงานของระบบเครือข่ายที่มีจำนวนโหนดมากกว่าสองโหนด										
6. สร้างโปรแกรมที่สะดวกต่อการจำลองการเชื่อมต่อของโหนด										
7. ทดสอบความถูกต้องของโปรแกรม										
8. จัดทำเอกสารและคู่มือการใช้										

1.6 ผลที่คาดว่าจะได้รับ

1. โปรแกรมใช้ในการศึกษาการทำงานของโพรโทคอลได้ดียิ่งขึ้น
2. โปรแกรมมีทางเลือกในการแสดงผลที่หลากหลาย
3. โปรแกรมสะดวกต่อการเพิ่มลดโพรโทคอลที่ใช้ในการจำลองการทำงานของเครือข่าย
4. โปรแกรมสะดวกต่อการเพิ่มลดส่วนแสดงผลที่ใช้ในการจำลองการทำงานของเครือข่าย
5. โปรแกรมสามารถศึกษาระบบการทำงานของเครือข่ายที่มีการจำลองมากกว่าสองโหนดได้
6. โปรแกรมสามารถสร้าง บันทึก โหลด และแก้ไข แบบจำลองทอพอโลยีได้

1.7 งบประมาณที่ใช้

- ค่าหนังสือ	400	บาท
- ค่าวัสดุสำนักงาน	100	บาท
- ค่าถ่ายเอกสารและจัดทำรูปเล่ม	1200	บาท
รวม	1700	บาท

หมายเหตุ สามารถถัวเฉลี่ยได้ทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีต่างๆที่ต้องใช้โดยหลักการและทฤษฎีส่วนใหญ่จะเกี่ยวกับระบบเครือข่าย ซึ่งเนื้อหาในบทประกอบด้วย

- โพรโทคอลในระบบเครือข่ายการสื่อสารข้อมูล
- อุปกรณ์สื่อสารในระบบเครือข่ายการสื่อสารข้อมูล
- ทอพอโลยีของระบบเครือข่ายการสื่อสารข้อมูล

2.1 โพรโทคอลในระบบเครือข่ายการสื่อสารข้อมูล

2.1.1 ความหมายและความสำคัญของโพรโทคอล [7]

โพรโทคอล (Protocol) คือข้อตกลงอย่างเป็นทางการเกี่ยวกับวิธีที่อุปกรณ์จะจัดรูปแบบตอบรับข้อมูลระหว่างการสื่อสาร ซึ่งโพรโทคอลจะมีหลายมาตรฐาน และในแต่ละโพรโทคอลจะมีข้อดีข้อเสียต่างกันไป

2.1.2 ความหมายและความสำคัญของโพรโทคอล [7]

โพรโทคอลมีความสำคัญอย่างยิ่งต่อการติดต่อสื่อสาร เพราะโพรโทคอลเป็นส่วนประกอบในการติดต่อสื่อสาร การติดต่อสื่อสารจะเกิดขึ้นได้ต้องมี 5 องค์ประกอบด้วยกัน ประกอบด้วย ผู้รับ ผู้ส่ง สื่อ สาร และสุดท้ายคือโพรโทคอล

การติดต่อสื่อสารข้อมูลผ่านทางเครือข่ายนั้น จำเป็นต้องมีการกำหนดโพรโทคอลที่เป็นข้อตกลงในการสื่อสารขึ้น เพื่อช่วยให้ระบบสองระบบที่แตกต่างกันโดยสิ้นเชิง สามารถสื่อสารกันอย่างเข้าใจได้โพรโทคอลจึงรวมถึงวิธีการส่งและรับข้อมูล วิธีการตรวจสอบข้อผิดพลาดของการส่งและรับข้อมูล การแสดงผลข้อมูลเมื่อส่งและรับ

2.1.3 แบบจำลอง OSI [8]

แบบจำลองโอเอสไอ (Open Systems Interconnection model: OSI model) เป็นมาตรฐานการติดต่อสื่อสารและโพรโทคอลของระบบคอมพิวเตอร์ พัฒนาขึ้นโดยองค์การระหว่างประเทศว่าด้วยการมาตรฐาน (ISO)

โมเดลนี้ได้ถูกแบ่งย่อยออกเป็น 7 ชั้นอันได้แก่ Application, Presentation, Session, Transportation, Network, Data Link และ Physical ตามลำดับจากบนลงล่าง เหตุผลที่โมเดลนี้ถูกแบ่งออกเป็น 7 ชั้นก็เพื่อให้ง่ายต่อการทำความเข้าใจว่าแต่ละชั้นนั้นมีหน้าที่อะไร และสัมพันธ์กันอย่างไรกับชั้นอื่น ซึ่งโดยหลักๆแล้วแต่ละชั้นจะมีความสัมพันธ์โดยตรงกับชั้นที่อยู่ติดกันกับชั้นนั้นๆ

1. Application Layer ชั้นที่เจ็ดเป็นชั้นที่อยู่ใกล้ผู้ใช้มากที่สุดและเป็นชั้นที่ทำงานส่งและรับข้อมูลโดยตรงกับผู้ใช้ ตัวอย่างเช่น ซอร์ฟแวร์โปรแกรม ต่างๆที่อาศัยอยู่บนเลเยอร์นี้ เช่น DNS, HTTP, Browser เป็นต้น

2. Presentation Layer ชั้นที่หกเป็นชั้นที่รับผิดชอบเรื่องรูปแบบของการแสดงผลเพื่อโปรแกรมต่างๆที่ใช้งานระบบเครือข่ายทราบว่าข้อมูลที่ได้เป็นประเภทใด เช่น [รูปภาพ, เอกสาร, ไฟล์วีดีโอ]

3. Session Layer ชั้นที่ห้านี้ทำหน้าที่ในการจัดการกับเซสชั่นของโปรแกรม ชั้นนี้เองที่ทำให้ในหนึ่งโปรแกรมยกตัวอย่างเช่น โปรแกรมค้นดูเว็บ (Web browser) สามารถทำงานติดต่ออินเทอร์เน็ตได้พร้อมๆกันหลายหน้าต่าง

4. Transport Layer ชั้นนี้ทำหน้าที่ดูแลจัดการเรื่องของความผิดพลาดที่เกิดขึ้นจากการสื่อสาร ซึ่งการตรวจสอบความผิดพลาดนั้นจะพิจารณาจากข้อมูลส่วนที่เรียกว่า checksum และอาจมีการแก้ไขข้อผิดพลาดนั้นๆ โดยพิจารณาจาก ฝั่งต้นทางกับฝั่งปลายทาง (End-to-end) โดยหลักๆแล้วชั้นนี้จะอาศัยการพิจารณาจาก พอร์ต (Port) ของเครื่องต้นทางและปลายทาง

5. Network Layer ชั้นที่สามจะจัดการการติดต่อสื่อสารข้ามเน็ตเวิร์ค ซึ่งจะเป็นการทำงานติดต่อข้ามเน็ตเวิร์คแทนชั้นอื่นๆที่อยู่ข้างบน

6. Data Link Layer ชั้นนี้จัดเตรียมข้อมูลที่จะส่งผ่านไปบนสื่อตัวกลาง

7. Physical Layer เป็นชั้นแรกของสื่อที่ใช้ในการติดต่อสื่อสาร ซึ่งอาจจะเป็นทั้งแบบที่ใช้สายหรือไม่ใช้สาย ตัวอย่างของสื่อที่ใช้ได้แก่ Shield Twisted Pair (STP), Unshield Twisted Pair (UTP), Fibre Optic และอื่นๆ

2.1.3 แบบจำลอง TCP/IP

แบบจำลอง ทีซีพี/ไอพี (TCP/IP model : Transmitsion Control Protocol/Internet Protocol) เป็นมาตรฐานที่ทำให้คอมพิวเตอร์ภายในระบบเครือข่ายอินเทอร์เน็ต สามารถเชื่อมต่อเข้าหากันและติดต่อสื่อสารแลกเปลี่ยนข้อมูลกันได้ เป็นมาตรฐานที่ว่าด้วยการกำหนดวิธีการติดต่อสื่อสารระหว่างคอมพิวเตอร์ โดยใช้แนวคิดของการแบ่งลำดับชั้นโพรโทคอล

ลำดับชั้นของโพรโทคอลในระบบอินเทอร์เน็ต มีลำดับชั้นที่น้อยกว่าโครงสร้างลำดับชั้นของโอเอสไอ โดยในโอเอสไอมีลำดับชั้นของโพรโทคอลทั้งหมด 7 ชั้น แต่ในระบบอินเทอร์เน็ตมีทั้งหมดเพียง 4 ชั้นเท่านั้น

1. Link layer หรือชั้นการเชื่อมต่อสามารถเทียบได้กับชั้นที่ 1 และ 2 ในโครงสร้างแบบโอเอสไอ เป็นลำดับชั้นที่ไม่ได้เกี่ยวข้องกับระบบอินเทอร์เน็ตโดยตรง แต่เป็นระบบพื้นฐานของการเชื่อมต่อที่ระบบอินเทอร์เน็ตใช้ส่งข้อมูลภายในเครือข่าย

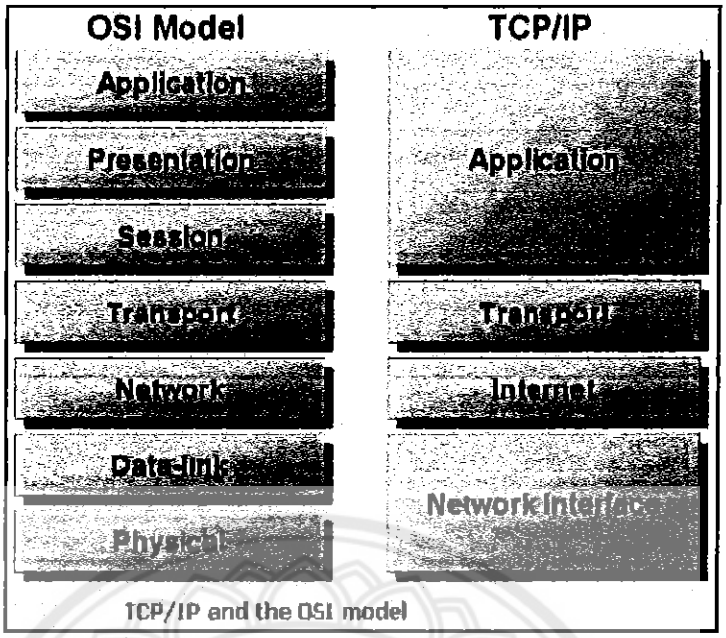
หน้าที่ของชั้นนี้สำหรับการส่งข้อมูล เนื่องจากแบบจำลอง TCP/IP ไม่ได้กำหนดมาตรฐานในข้อตอนนี้อย่างมากนัก กำหนดไว้เพียงว่าให้สามารถส่งข้อมูลสู่เครือข่ายได้เท่านั้น ทำให้ไม่สามารถระบุเนื้อหาหน้าที่ที่ชัดเจนได้ ดังนั้นจึงอาจจะยกระบบของ โครงสร้างแบบ โอเอสไอทั้งสองชั้นแรกมาซึ่งได้แก่การจัดเตรียมข้อมูลเพื่อให้เหมาะสมก่อนที่จะส่งไปยังที่หมายปลายทาง ซึ่งได้แก่การจัดเตรียม Packet Header การควบคุมระบบฮาร์ดแวร์และซอฟต์แวร์ที่จะใช้ในการจัดส่ง เช่น การเชื่อมต่อกับ Network card และการใช้งาน Device Driver หน้าที่สำหรับการรับข้อมูลคือคอยรับกรอบของข้อมูลที่ได้รับ นำข้อมูลส่วนหัวออกมา และจัดเตรียมข้อมูลเพื่อส่งต่อไปยังชั้นเครือข่ายตัวอย่าง Link layer เช่น Ethernet, Wi-Fi

2. Internet layer หรือ ชั้นอินเทอร์เน็ต เทียบได้กับชั้นที่ 3 ซึ่งคือ Network Layer ในโครงสร้างแบบ โอเอสไอ เป็นชั้นที่มีหน้าที่ส่งข้อมูลจากจุดเริ่มต้นไปยังปลายทาง โดยหาเส้นทางที่ข้อมูลจะใช้เดินทางผ่านเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่งจนกระทั่งถึงปลายทาง

โพรโทคอลที่ใช้ในชั้นนี้คือ อินเทอร์เน็ตโพรโทคอล หรือ ไอพี ทำหน้าที่เปรียบเสมือนซองจดหมายซึ่งระบุถึงที่อยู่ของต้นทางและปลายทาง โดยมีบุรุษไปรษณีย์ทำหน้าที่ส่งจดหมายนั้นผ่านกรมการไปรษณีย์ในพื้นที่ต่างๆ จนถึงจุดหมายปลางทาง ที่อยู่บนซองจดหมายในอินเทอร์เน็ตโพรโทคอลเรียกว่า หมายเลขไอพี ที่ทำการไปรษณีย์คือเราเตอร์ที่ทำหน้าที่ค้นหาเส้นทางที่เหมาะสมเพื่อส่งข้อมูลไปตามสายส่งจนกระทั่งถึงปลายทางตัวอย่าง Internet layer เช่น IP

3. Transport layer หรือ ชั้นขนส่งเทียบได้กับชั้นที่ 4 ในโครงสร้างแบบ โอเอสไอ เป็นชั้นที่มีหน้าที่ควบคุมการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองเครื่องที่ติดต่อกัน ซึ่งอาจแบ่งได้สองลักษณะคือ บริการการส่งข้อมูลแบบสร้างการเชื่อมต่อ และบริการการส่งข้อมูลแบบไม่สร้างการเชื่อมต่อ และจัดส่งข้อมูลไปยังapplicationที่ต้องการข้อมูลตัวอย่าง Transport layer ได้แก่ TCP, UDP, RTP

4. Application layer หรือ ชั้นการประยุกต์ใช้งาน เทียบได้กับชั้นที่ 5 ถึง 7 ในโครงสร้างแบบ โอเอสไอ จะครอบคลุมบริการที่เกี่ยวข้องกับการรักษาความปลอดภัย การเข้ารหัส การเชื่อมต่อระหว่างโปรแกรมประยุกต์ และเป็นชั้นที่โปรแกรมประยุกต์ใช้งานโดยตรง โดยโพรโทคอลที่อยู่บนชั้นนี้จะถูกออกแบบให้เหมาะสำหรับประเภทของโปรแกรมประยุกต์เฉพาะทาง เช่น โปรแกรมอีเมลใช้โพรโทคอล SMTP สำหรับส่งอีเมล ใช้โพรโทคอล POP3 สำหรับรับและเรียกดูอีเมล, ส่วนโปรแกรมเว็บเบราว์เซอร์ใช้โพรโทคอล HTTP สำหรับเรียกดูเว็บเพจ เป็นต้นตัวอย่าง Application layer เช่น HTTP, FTP, DNS เป็นต้น

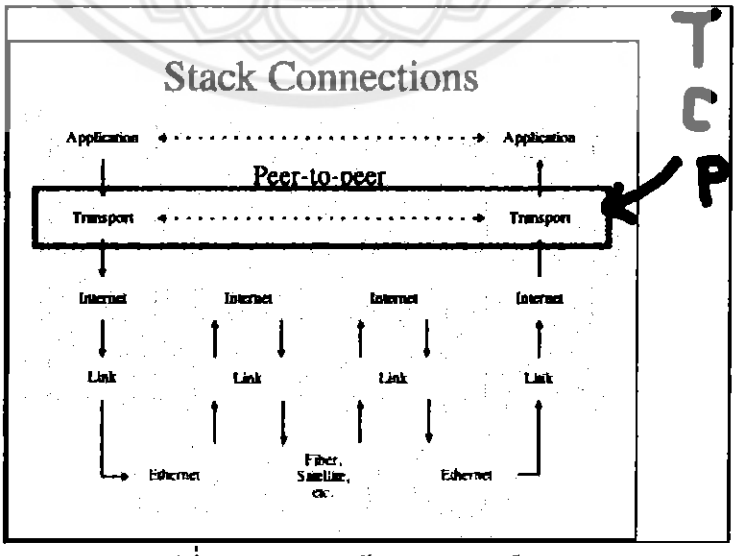


รูปที่ 2.1 แบบจำลอง OSI และ แบบจำลอง TCP/IP

ที่มา: [6]

2.1.4 เกณฑ์วิธีควบคุมการขนส่งข้อมูล(ทีซีพี) [1]

เกณฑ์วิธีควบคุมการขนส่งข้อมูล (Transmission Control Protocol) หรือ**ทีซีพีโดยทีซีพีเป็นหนึ่งในสองส่วนประกอบเดิมของชุดการส่งเสริมอินเตอร์เนตโพรโทคอล (IP) โดยที่ทั้งหมดจะเรียกรวมกันว่า ทีซีพี/ไอพี (TCP/IP) โดยการควบคุมการส่งโพรโทคอลถูกสร้างขึ้นบนอินเตอร์เน็ตโพรโทคอล (IP)**



รูปที่ 2.2การส่งต่อข้อมูลระดับเครือข่าย

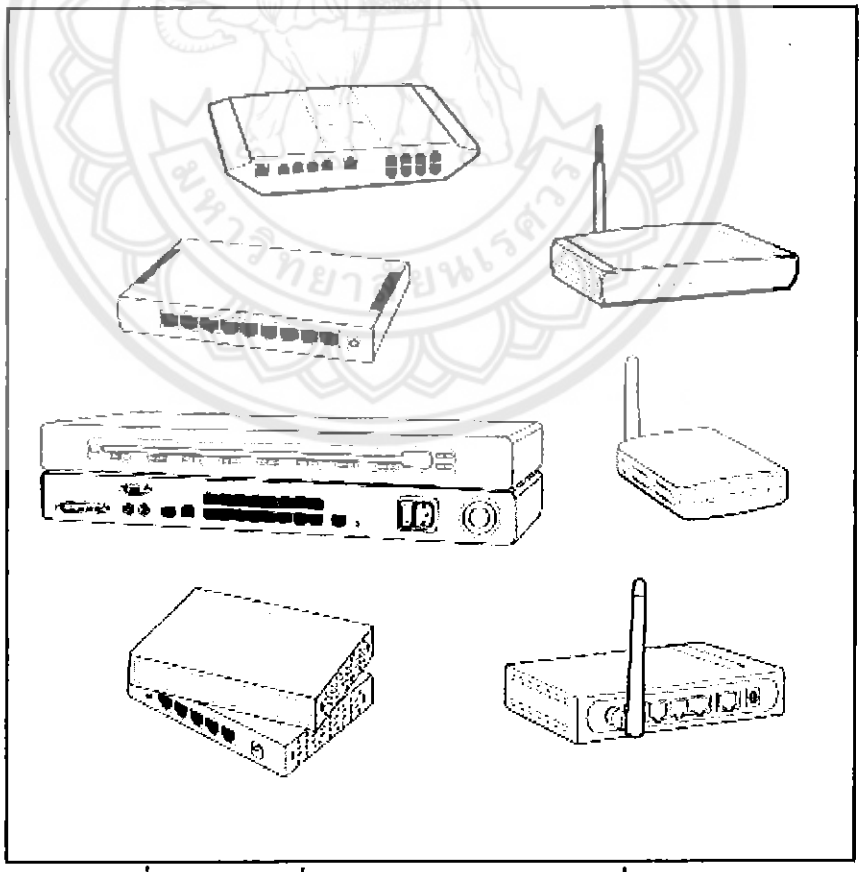
ที่มา: [1]

การทำงานของโพรโทคอลทีซีพีจะถือว่าไอพีที่รับอาจมีการสูญเสียข้อมูลบางส่วนได้ ดั้งนั้นจึงได้ทำการเก็บสำเนาของข้อมูลไว้ หากข้อมูลที่ได้รับการตอบกลับจากไอพีปลายทางไม่ตรงกับข้อมูลที่ต้องการส่ง จะทำการส่งข้อมูลไปอีกครั้ง หากส่วนรับไม่ทำการตอบกลับมาเป็นเวลานานก็เช่นกัน มันจะทำการส่งข้อมูลอีกครั้ง และหากข้อมูลที่ตอบกลับมาตรงกับข้อมูลที่ทำการส่งจะทำการส่งข้อมูลชุดถัดไป

ข้อมูลจะถูกส่งจากชั้นไอพีของเครื่องหนึ่งไปยังชั้นไอพีของอีกเครื่องหนึ่งจนถึงปลายทาง ข้อมูลจะถูกส่งจากชั้นของไอพีมายังชั้นทีซีพี โดยชั้นทีซีพีจะเก็บข้อมูลเป็นบัฟเฟอร์ เมื่อข้อมูลที่ได้มีความถูกต้องและสมบูรณ์แล้ว ทีซีพีจะทำการส่งข้อมูลไปยังชั้นที่สูงกว่าต่อไป

2.2 อุปกรณ์สื่อสารในระบบเครือข่ายการสื่อสารข้อมูล

การส่งข้อมูลในระบบเครือข่ายการสื่อสารข้อมูลนั้น จะต้องส่งข้อมูลจากโหนดหนึ่งไปยังอีกโหนดหนึ่งส่งจนข้อมูลถึงโหนดปลายทาง โดยโหนดต่างๆในระบบเครือข่ายการสื่อสารข้อมูลนั้นมีหลายชนิด และแต่ละชนิดทำงานในลำดับชั้นที่ต่างกัน ตัวอย่างเช่น



รูปที่ 2.3 อุปกรณ์สื่อสารในระบบเครือข่ายการสื่อสารข้อมูล

ที่มา: [8]

2.2.1 เครื่องทวนสัญญาณ (Repeater) [8]

เป็นอุปกรณ์ที่ทำหน้าที่รับสัญญาณดิจิทัล แล้วส่งต่อออกไปยังอุปกรณ์ตัวอื่น เหตุที่ต้องใช้อุปกรณ์ทวนสัญญาณ เนื่องจากการส่งสัญญาณไปในตัวกลางที่เป็นสัญญาณนั้น เมื่อระยะทางมากขึ้นแรงดันของสัญญาณจะลดลงเรื่อยๆ ทำให้ไม่สามารถส่งสัญญาณในระยะทางไกลๆได้ ดังนี้ การใช้อุปกรณ์ทวนสัญญาณจะทำให้สามารถส่งสัญญาณไปได้ไกลขึ้น โดยสัญญาณไม่สูญหาย

2.2.2 ฮับ (Hub) [8]

เป็นอุปกรณ์ที่ทำหน้าที่รวมสัญญาณที่มาจากอุปกรณ์รับส่ง หรือเครื่องคอมพิวเตอร์หลายๆ เครื่องเข้าด้วยกัน สัญญาณที่ส่งมาจากฮับจะกระจายไปยังทุกเครื่องที่ต่ออยู่กับฮับ ซึ่งแต่ละเครื่องจะเลือกรับเฉพาะข้อมูลที่ส่งมาถึงตนเองเท่านั้น

2.2.3 บริดจ์ (Bridge) [8]

ใช้ในการเชื่อมต่อเครือข่ายหลายเครือข่ายเข้าด้วยกัน โดยจะต้องเป็นเครือข่ายที่ใช้โพรโทคอลเดียวกัน ซึ่งมีความสามารถมากกว่าฮับและอุปกรณ์ทวนสัญญาณ คือ สามารถกรองข้อมูลที่ส่งต่อได้ โดยการตรวจสอบว่า ข้อมูลที่ส่งนั้นปลายทางอยู่ที่ใด หากเครื่องปลายทางอยู่ภายในเครือข่ายเดียวกันกับเครื่องส่ง ก็จะส่งข้อมูลนั้นไปในเครือข่ายเดียวกันเท่านั้น ไม่ส่งไปยังเครือข่ายอื่น แต่หากข้อมูลมีปลายทางอยู่ที่เครือข่ายอื่น ก็จะส่งข้อมูลไปในเครือข่ายที่มีเครื่องปลายทางอยู่เท่านั้น ทำให้สามารถจัดการกับความหนาแน่นของข้อมูลได้อย่างมีประสิทธิภาพมากขึ้น

2.2.4 อุปกรณ์จัดหาเส้นทาง (Router) [8]

สามารถกรองข้อมูลได้เช่นกับบริดจ์ แต่จะมีความสามารถมากกว่า โดยจะหาเส้นทางในการส่งกลุ่มข้อมูล (data packet) ไปยังเครื่องปลายทางในระยะทางที่สั้นที่สุดได้

2.2.5 สวิตช์ (Switch)[8]

นำความสามารถของฮับกับบริดจ์มารวมกัน แต่การส่งข้อมูลจากคอมพิวเตอร์ตัวหนึ่งจะไม่กระจายไปยังคอมพิวเตอร์ทุกเครื่องเหมือนกับฮับ เพราะสวิตช์จะทำหน้าที่รับกลุ่มข้อมูลมาตรวจสอบก่อนว่าเป็นของคอมพิวเตอร์เครื่องใด แล้วนำข้อมูลนั้นส่งต่อไปยังคอมพิวเตอร์เป้าหมาย ซึ่งช่วยลดปัญหาการชนกันหรือความคับคั่งของข้อมูล

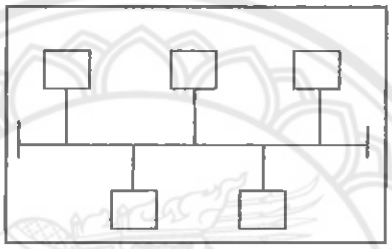
2.2.6 เกตเวย์ (Gateway) [8]

เป็นอุปกรณ์ที่ทำหน้าที่เชื่อมต่อเครือข่ายต่างๆ เข้าด้วยกันไม่ว่าเครือข่ายนั้นจะใช้โพรโทคอลตัวใดก็ตาม เนื่องจากเกตเวย์สามารถแปลงรูปแบบแพ็คเก็ตของโพรโทคอลหนึ่งไปเป็นรูปแบบของอีกโพรโทคอลหนึ่งได้ เพื่อให้เหมาะสมกับการใช้งานในเครือข่าย ทำให้สามารถเชื่อมต่อกับเครือข่ายอื่นๆ ได้อย่างไม่มีข้อจำกัด แต่ในปัจจุบันนี้ได้รวมการทำงานของเกตเวย์ไว้ในอุปกรณ์จัดเส้นทาง (Router) แล้วทำให้อุปกรณ์จัดเส้นทางสามารถทำงานเป็นเกตเวย์ได้ ดังนั้น จึงไม่จำเป็นต้องซื้อเกตเวย์อีก

2.3 ทอพอโลยี

2.3.1 ความหมายทอพอโลยี [5]

ทอพอโลยีคือลักษณะทางกายภาพของระบบเครือข่าย ซึ่งหมายถึงลักษณะของการเชื่อมโยงสายสื่อสารเข้ากับอุปกรณ์อิเล็กทรอนิกส์และเครื่องคอมพิวเตอร์ ภายในเครือข่ายด้วยกันนั่นเองทอพอโลยีของเครือข่าย LAN แต่ละแบบมีความเหมาะสมในการใช้งาน แตกต่างกันออกไป การนำไปใช้จึงมีความจำเป็นที่เราจะต้องทำการศึกษาลักษณะและคุณสมบัติข้อดีและข้อเสียของทอพอโลยีแต่ละแบบ เพื่อนำไปใช้ในการออกแบบพิจารณาเครือข่ายให้เหมาะสมกับการใช้งาน รูปแบบของทอพอโลยี ของเครือข่ายหลักๆ มีดังต่อไปนี้



รูปที่ 2.4 ทอพอโลยีแบบบัส

ที่มา: [5]

2.3.1.1 ทอพอโลยีแบบบัส (BUS)

เป็นรูปแบบที่เครื่องคอมพิวเตอร์จะถูกเชื่อมต่อกันโดยผ่ายสายสัญญาณแกนหลัก ที่เรียกว่า BUS หรือ แบ็คโบน (Backbone) คือ สายรับส่งสัญญาณข้อมูลหลักใช้เป็นทางเดินข้อมูลของทุกเครื่องภายในระบบเครือข่ายและจะมีสายแยกย่อยออกไปในแต่ละจุดเพื่อเชื่อมต่อเข้ากับคอมพิวเตอร์เครื่องอื่นๆ ซึ่งเรียกว่าโหนด (Node) ข้อมูลจากโหนดผู้ส่งจะถูกส่งเข้าสู่สายบัสในรูปของแพ็กเกจซึ่งแต่ละแพ็กเกจจะประกอบไปด้วยข้อมูลของผู้ส่ง, ผู้รับ และข้อมูลที่จะส่งการสื่อสารภายในสายบัสจะเป็นแบบ 2 ทิศทางแยกไปยังปลายทั้ง 2 ด้านของ บัสโดยตรงปลายทั้ง 2 ด้านของบัส จะมีเทอร์มิเนเตอร์ (Terminator) ทำหน้าที่ลบล้างสัญญาณที่ส่งมาถึงเพื่อป้องกันไม่ให้สัญญาณข้อมูลนั้นสะท้อนกลับ เข้ามายังบัสอีกเพื่อเป็นการป้องกันการชนกันของข้อมูลอื่นๆ ที่เดินทางอยู่บนบัสในขณะนั้น

สัญญาณข้อมูลจากโหนดผู้ส่งเมื่อเข้าสู่บัส ข้อมูลจะไหลผ่านไปยังปลายทั้ง 2 ด้านของบัส แต่ละโหนดที่เชื่อมต่อเข้ากับบัส จะคอยตรวจดูว่าตำแหน่งปลายทางที่มากับแพ็กเกจข้อมูลนั้นตรงกับตำแหน่งของตนหรือไม่ ถ้าตรงก็จะรับข้อมูลนั้นเข้ามาสู่โหนดตน แต่ถ้าไม่ใช่ก็จะปล่อยให้สัญญาณข้อมูลนั้นผ่านไป จะเห็นว่าทุกๆ โหนดภายในเครือข่ายแบบ BUS นั้นสามารถรับรู้สัญญาณข้อมูลได้แต่จะมีเพียงโหนดปลายทางเพียงโหนดเดียวเท่านั้นที่จะรับข้อมูลนั้นไปได้

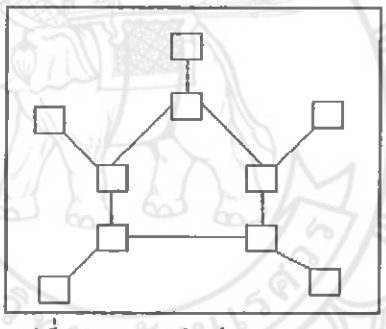
ก. ข้อดี

- ไม่ต้องเสียค่าใช้จ่ายในการวางสายสัญญาณมากนัก สามารถขยายระบบได้ง่ายเสียค่าใช้จ่ายน้อยซึ่งถือว่าระบบบัสนี้เป็นแบบทอพอโลยีที่ได้รับความนิยมใช้กันมากที่สุดมาตั้งแต่อดีตจนถึงปัจจุบัน เหตุผลอย่างหนึ่งก็คือสามารถติดตั้งระบบ ดูแลรักษาและติดตั้งอุปกรณ์เพิ่มเติมได้ง่ายไม่ต้องใช้เทคนิคที่ยุ่งยากซับซ้อนมากนัก

ข. ข้อเสีย

- อาจเกิดข้อผิดพลาดได้ง่ายเนื่องจากทุกเครื่องคอมพิวเตอร์ ต่ออยู่บนสายสัญญาณเพียงเส้นเดียวดังนั้นหากมี สัญญาณขาดที่ตำแหน่งใดตำแหน่งหนึ่ง ก็จะทำให้เครื่องบางเครื่องหรือทั้งหมดในระบบไม่สามารถใช้งานได้ตามไปด้วย

- การตรวจหาโหนดเสีย ทำได้ยากเนื่องจากขณะใดขณะหนึ่ง จะมีคอมพิวเตอร์เพียงเครื่องเดียวเท่านั้นที่สามารถส่งข้อความ ออกมาบนสายสัญญาณดังนั้นถ้ามีเครื่องคอมพิวเตอร์จำนวนมากๆอาจทำให้เกิดการคับคั่งของเน็ตเวิร์ค ซึ่งจะทำให้ระบบช้าลงได้



รูปที่ 2.5 ทอพอโลยีแบบวงแหวน

ที่มา: [5]

2.3.1.2 ทอพอโลยีแบบวงแหวน (RING)

เป็นรูปแบบที่ เครื่องคอมพิวเตอร์ทุกเครื่องในระบบเครือข่ายทั้งเครื่องที่เป็นผู้ให้บริการ(Server) และเครื่องที่เป็นผู้ขอใช้บริการ (Client) ทุกเครื่องถูกเชื่อมต่อกันเป็นวงกลมข้อมูลข่าวสารที่ส่งระหว่างกัน จะไหลวนอยู่ในเครือข่ายไปใน ทิศทางเดียวกันโดยไม่มีจุดปลายหรือเทอร์มิเนเตอร์เช่นเดียวกับเครือข่ายแบบ BUS ในแต่ละโหนดหรือแต่ละเครื่อง จะมีรีพีตเตอร์ (Repeater) ประจำแต่ละเครื่อง 1 ตัวซึ่งจะทำหน้าที่เพิ่มเติมข้อมูลที่จำเป็นต่อการติดต่อสื่อสารเข้าในส่วนหัวของแพ็กเกจที่ส่งและตรวจสอบข้อมูลจากส่วนหัวของ Packet ที่ส่งมาถึง ว่าเป็นข้อมูลของตนหรือไม่ แต่ถ้าไม่ใช่ก็จะปล่อยข้อมูลนั้นไปยัง Repeater ของเครื่องถัดไป

ก. ข้อดี

- ผู้ส่งสามารถส่งข้อมูลไปยังผู้รับได้หลายๆเครื่องพร้อมๆกัน โดยกำหนดตำแหน่งปลายทางเหล่านั้นลงในส่วนหัวของแพ็กเกจข้อมูล Repeaterของแต่ละเครื่องจะทำการตรวจสอบเองว่า ข้อมูลที่ส่งมาให้นั้นเป็นตนเองหรือไม่

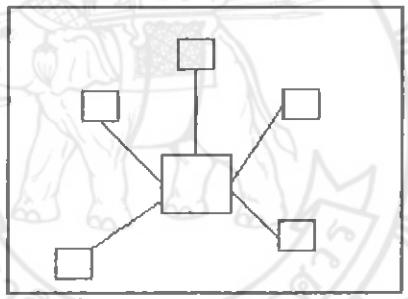
- การส่งผ่านข้อมูลในเครือข่ายแบบ RING จะเป็นไปในทิศทางเดียวจากเครื่องสู่เครื่อง จึงไม่มีการชนกันของสัญญาณข้อมูลที่ส่งออกไป

- คอมพิวเตอร์ทุกเครื่องในเน็ตเวิร์กมีโอกาสที่จะส่งข้อมูลได้อย่างทัดเทียมกัน

ข. ข้อเสีย

- ถ้ามีเครื่องใดเครื่องหนึ่งในเครือข่ายเสียหายข้อมูลจะไม่สามารถส่งผ่านไปยังเครื่องต่อๆไปได้และจะทำให้เครือข่ายทั้งเครือข่าย หยุดชะงักได้

- ขณะที่ข้อมูลถูกส่งผ่านแต่ละเครื่องเวลาส่วนหนึ่งจะสูญเสียไปกับการที่ทุกๆ Repeater จะต้องทำการตรวจสอบตำแหน่งปลายทางของข้อมูลนั้นๆ ทุกข้อมูลที่ส่งผ่านมาถึง



รูปที่ 2.6 ทอพอโลยีแบบวงดาว

ที่มา: [5]

2.3.1.3 ทอพอโลยีแบบดาว (STAR)

เป็นรูปแบบที่เครื่องคอมพิวเตอร์ทุกเครื่องที่เชื่อมต่อเข้าด้วยกันในเครือข่ายจะต้องเชื่อมต่อกับอุปกรณ์ตัวกลางตัวหนึ่งที่เรียกว่า ฮับ (HUB) หรือเครื่องๆ หนึ่งซึ่งทำหน้าที่เป็นศูนย์กลางของการเชื่อมต่อสายสัญญาณที่มาจากเครื่องต่างๆในเครือข่าย และควบคุมเส้นทางการสื่อสาร ทั้งหมด เมื่อมีเครื่องที่ต้องการส่งข้อมูลไปยังเครื่องอื่นๆ ที่ต้องการในเครือข่ายเครื่องนั้นก็จะต้องส่งข้อมูลมายัง HUB หรือเครื่องศูนย์กลางก่อน แล้ว HUB ก็จะทำหน้าที่กระจายข้อมูลนั้นไปในเครือข่ายต่อไป

ก. ข้อดี

- การติดตั้งเครือข่ายและการดูแลรักษาทำได้ง่าย หากมีเครื่องใดเกิดความเสียหาย ก็สามารถตรวจสอบได้ง่าย และศูนย์กลางสามารถตัดเครื่องที่เสียหายนั้นออกจากการสื่อสาร ในเครือข่ายได้เลยโดยไม่มีผลกระทบกับระบบเครือข่าย

ข. ข้อเสีย

- เสียค่าใช้จ่ายมากทั้งในด้านของเครื่องที่จะใช้เป็น เครื่องศูนย์กลาง หรือตัว HUB เองและค่าใช้จ่ายในการติดตั้งสายเคเบิลในเครื่องอื่นๆ ทุกเครื่องการขยายระบบให้ใหญ่ขึ้นทำได้ยาก เพราะการขยายแต่ละครั้งจะต้องเกี่ยวเนื่องกับเครื่องอื่นๆ ทั้งระบบ

2.3.1.4 ทอพอโลยีแบบ Hybrid

เป็นรูปแบบใหม่ ที่เกิดจากการผสมผสานกันของทอพอโลยีแบบ STAR , BUS , RING เข้าด้วยกัน เพื่อเป็นการลดข้อเสียของรูปแบบที่กล่าวมาและเพิ่มข้อดี ขึ้นมา มักจะนำมาใช้กับระบบ WAN (Wide Area Network) มากซึ่งการเชื่อมต่อกันของแต่ละรูปแบบนั้นต้องใช้ตัวเชื่อมสัญญาณเข้ามาเป็นตัวเชื่อม ตัวนั้นก็คือ Router เป็นตัวเชื่อมการติดต่อกัน

2.3.1.5 ทอพอโลยีแบบ MESH

เป็นรูปแบบที่ถือว่าสามารถป้องกันการผิดพลาดที่อาจจะเกิดขึ้นกับระบบได้ดีที่สุดเป็นรูปแบบที่ใช้วิธีการเดินสายของแต่เครื่องไปเชื่อมการติดต่อกับทุกเครื่องในระบบเครือข่ายคือเครื่องทุกเครื่องในระบบเครือข่ายนี้ ต้องมีสายไปเชื่อมกับทุกๆ เครื่องระบบนี้ยากต่อการเดินสายและมีราคาแพง จึงมีค่อยมีผู้นิยมมากนัก

บทที่ 3
วิธีการดำเนินงานโครงงาน

ในบทนี้จะกล่าวถึงขั้นตอนการดำเนินงานเพื่อพัฒนาโปรแกรม ตั้งแต่การแก้ไขโปรแกรมเดิมจนถึงการจัดทำคู่มือโครงงาน ซึ่งประกอบด้วย เนื้อหาดังนี้

- ศึกษาโครงสร้าง การทำงานข้อดี และข้อจำกัดของโปรแกรมเดิม
- แก้ไขโครงสร้างเพื่อให้โปรแกรมสะดวกต่อการพัฒนาต่อ
- เพิ่มเติมส่วนแสดงผล
- ออกแบบ และสร้าง GUI ที่แสดงโหนดแต่ละโหนด
- พัฒนาให้โปรแกรมสะดวกต่อการเพิ่มหรือลดส่วนแสดงผล
- พัฒนาให้โปรแกรมสะดวกต่อการเพิ่มหรือลดโพรโทคอลแต่ละชั้น
- พัฒนาให้โปรแกรมสามารถส่งได้หลายโหนด
- นำความรู้เรื่องเทรดมาใช้เพื่อพัฒนาโปรแกรม
- เพิ่มส่วน GUI ที่ใช้ในการจำลองการเชื่อมของโหนด
- ตรวจสอบการใช้งานจริง
- จัดทำคู่มือโครงงาน

3.1 ศึกษาโครงสร้าง การทำงาน ข้อดี และข้อจำกัดของโปรแกรมเดิม

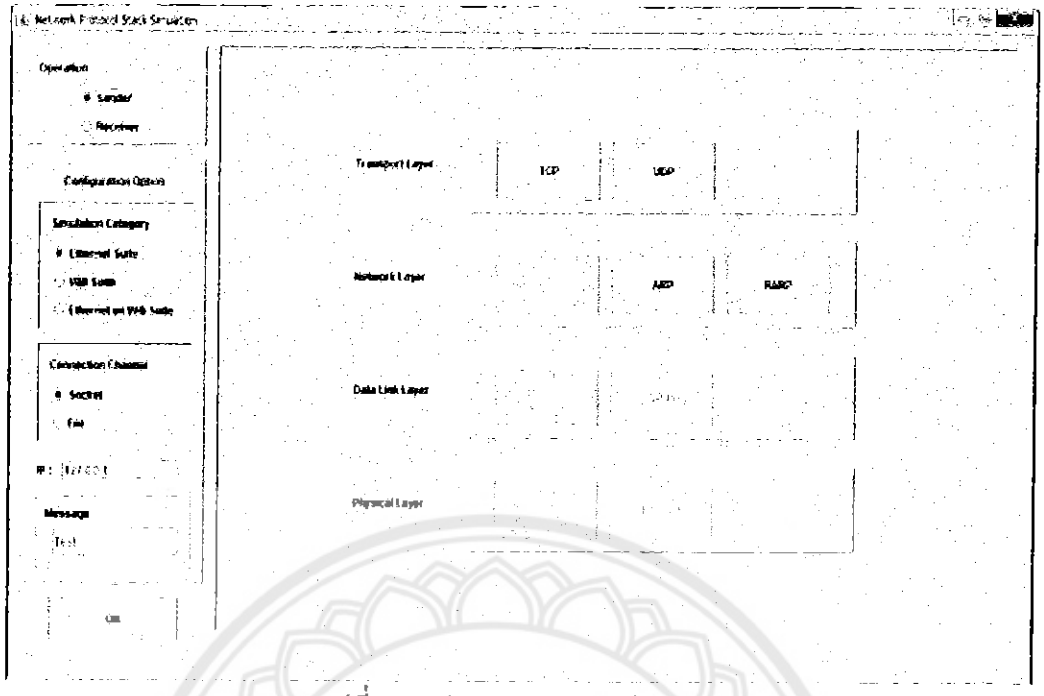
เริ่มต้นการทำโครงงาน โดยการศึกษา โปรแกรมเดิม ซึ่งโปรแกรมที่มีอยู่นั้น เป็นโปรแกรมที่ถูกเขียนโดยใช้ภาษาจาวา ซึ่งโปรแกรมที่มีอยู่นั้นมีข้อดี และข้อจำกัดดังนี้

3.1.1 ข้อดีของโปรแกรม

- สามารถจำลองการส่งข้อมูลระหว่างสองโหนดได้ โดยกำหนดโหนดรับและโหนดส่ง
- สามารถเลือกโพรโทรคอลที่จะใช้ในการส่งได้

3.1.2 ข้อจำกัดโปรแกรมเดิม

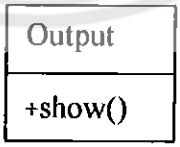
- ส่วนแสดงผล ไม่สามารถตรวจสอบข้อมูลการส่งย้อนหลังได้ไม่สามารถจัดเก็บข้อมูลการส่งแล้วนำไปศึกษาต่อได้ การเพิ่มส่วนแสดงผลใหม่ๆทำได้ยาก
- โพรโทคอลมีค่อนข้างจำกัด และทำการเพิ่มได้ยาก
- ไม่สามารถจำลองการส่งข้อมูลมากกว่าสองโหนดได้



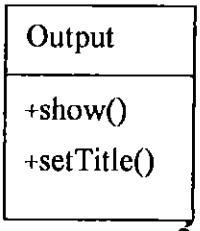
รูปที่ 3.1 รูปแสดง GUI ของโปรแกรมเดิม

3.2 แก้ไขโครงสร้างเพื่อให้โปรแกรมสะดวกต่อการพัฒนาต่อ

เนื่องจากโปรแกรมเดิมมีการทำงานบางส่วนที่ควรแก้ไข อีกทั้งโครงสร้างยังมีข้อจำกัดอยู่ พอสมควรควรจึงได้ทำการแก้ไขคลาสบางคลาส หรือทำการสร้างคลาสตัวใหม่มาใช้แทน ตัวอย่างเช่น



รูปที่ 3.2 Interface Output ตอนเริ่มสร้าง



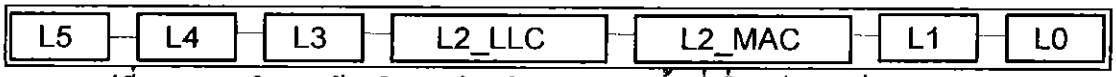
รูปที่ 3.3 Interface Output ตอนสิ้นสุดการพัฒนา

OutputUnit
+L5_Write() +L4_UDP_Write_Send() +L4_TCP_Write_Send() +L4_UDP_Write_Receive() +L4_TCP_Write_Receive() + L3_IP_Write_Send() + L3_IP_Write_Receive() +L3_ARP_RARP_Write() + L2_LLC_Ethernet_Write_Send() + L2_LLC_WIFI_Write_Send() + L2_MAC_Ethernet_Write_Send() + L2_MAC_WIFI_Write_Send() +L2_MAC_WIFI_Ack_Write_Send() +L1_PHY_CAT5_Write_Send() +L1_PHY_Infrared_Write_Send() +L2_LLC_Ethernet_Write_Receive() +L2_LLC_WIFI_Write_Receive() +L2_MAC_Ethernet_Write_Receive() +L2_MAC_WIFI_Write_Receive() +L2_MAC_WIFI_Ack_Write_Receive() +L1_PHY_CAT5_Write_Receive() +L1_PHY_Infrared_Write_Receive() +show()

รูปที่ 3.4 Interface OutputUnit

เหตุผลที่ OutputUnit ไม่เหมาะแก่การสร้าง ส่วนแสดงผลเพิ่มเติมเนื่องจาก OutputUnit มีฟังค์ชันเยอะเกิน ไม่เพียงเท่านี้ฟังค์ชันแต่ละตัวยังมีการรับค่าที่ซับซ้อนด้วย เช่น L2_MAC_WIFI_Write_Send (String version, String type, String subType, String toDS, String fromDS, String MF, String RT, String PM, String MD, String WEP, String order, String duration, String address1, String address2, String address3, String fragmentNumber, String sequenceNumber, String address4, String infoField, String CRC)

นอกจากการปรับเปลี่ยนส่วนแสดงผลแล้ว ยังมีการปรับเปลี่ยนโครงสร้างข้อมูลจากฟังค์ชัน nextLayer ที่เชื่อมแต่ละระดับชั้นเข้ากันดังนี้

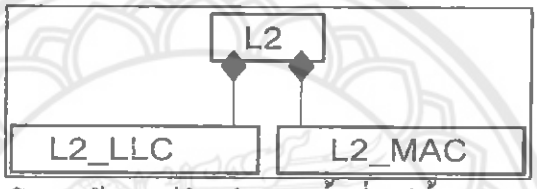


รูปที่ 3.5 แสดงโครงสร้างเดิมของโพรโทคอลแต่ละชั้นที่เชื่อมด้วยฟังค์ชัน nextLayer

จากเดิมจะเห็นว่าชั้นที่ 2 ถูกบังคับให้ต้องประกอบด้วย LLC และ MAC จึงไม่รองรับโพรโทคอลชั้นที่ 2 ที่มีเพียงส่วนเดียวจะ จึงได้ปรับเปลี่ยนโครงสร้างเป็น โครงสร้างที่มีโพรโทคอลชั้นที่ 2 เพียงส่วนเดียว

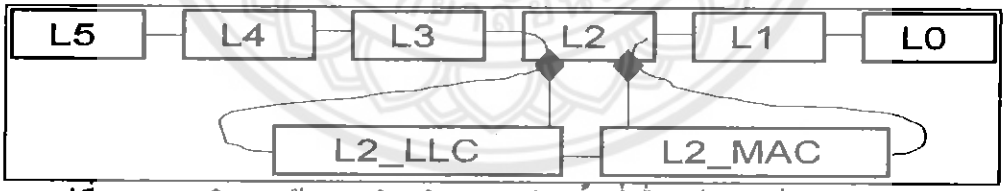
รูปที่ 3.6 แสดงโครงสร้างใหม่ของโพรโทคอลแต่ละชั้นที่เชื่อมด้วยฟังค์ชัน nextLayer

หาก โพรโทคอลชั้นที่2 มีทั้งส่วนที่เป็น LLC และ MAC จะได้ว่าโพรโทคอลชั้นที่ 2 มีส่วน LLC และ MAC อยู่ในตัว



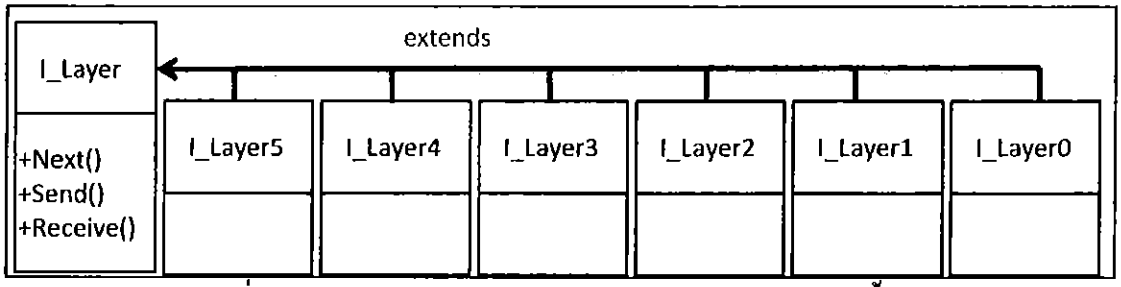
รูปที่ 3.7 แสดงโครงสร้างกรณีโพรโทคอลชั้นที่ 2 มีทั้งส่วน LLC และ MAC

กรณีที่โพรโทคอลระดับชั้นที่สอง มีทั้งส่วนที่เป็น LLC และ MAC ฟังค์ชัน nextLayer ของโพรโทคอล จะเป็นการเรียกใช้ฟังค์ชัน nextLayer ของส่วน LLC และ MAC เพื่อให้ได้การเชื่อมโยงกันด้วยฟังค์ชัน nextLayer ดังรูป



รูปที่ 3.8 แสดงโครงสร้างของโพรโทคอลแต่ละชั้นที่เชื่อมด้วยฟังค์ชัน nextLayer โดยกรณีที่ชั้นที่ 2 ประกอบด้วย LLC และ MAC

นอกจากนี้ยังมีการแก้ไขโดยมีการเพิ่ม Interface I_Layer1เพื่อให้เป็นตัวหลักในการสืบทอดของ Interface I_Layer0, I_Layer1, I_Layer2, I_Layer2_LLC, I_Layer2_MAC, I_Layer3, I_Layer4, I_Layer5 ทั้งนี้เนื่องจาก Interface ที่กล่าวมามีการใช้ฟังค์ชันเดียวกัน และเพื่อรองรับการพัฒนาต่อในอนาคตด้วย

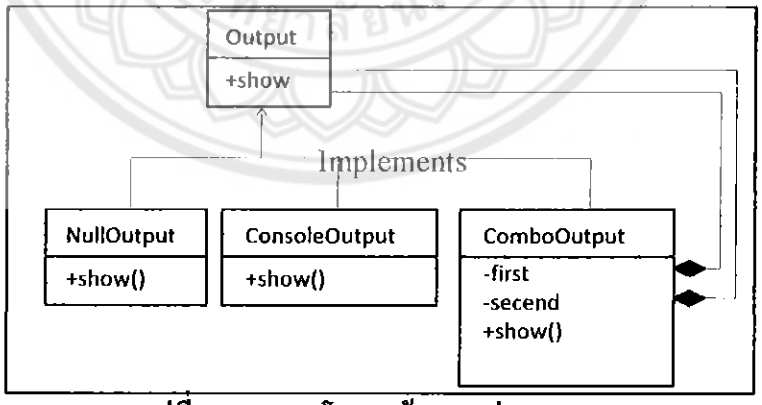


รูปที่ 3.9 แสดงโครงสร้างการสืบทอดของโพรโทคอลแต่ละชั้น

นอกจากการแก้ไขโครงสร้างตามที่กล่าวมาแล้วยังมีการแก้ไขความผิดพลาดของโปรแกรมในหลายๆจุด โดยได้ทำให้โปรแกรมมีความผิดพลาดน้อยลง

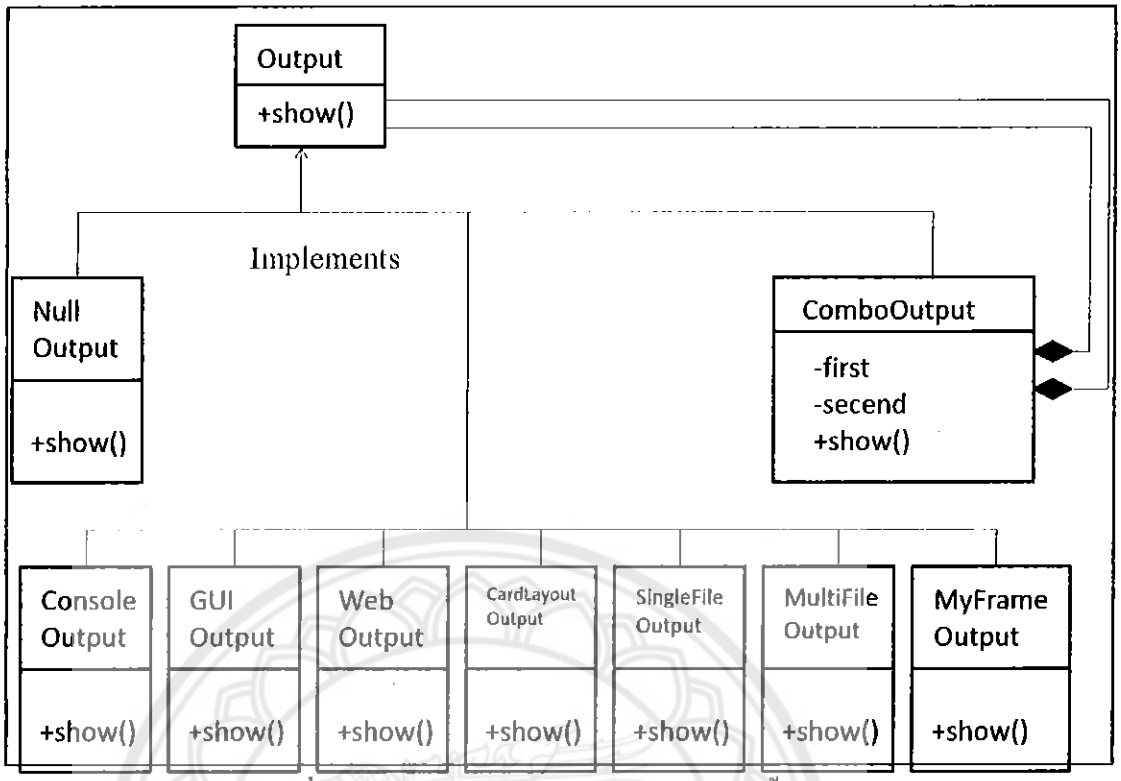
3.3 เพิ่มเติมส่วนแสดงผล

เนื่องจากการแสดงผลแต่เดิมประกอบด้วยส่วนแสดงผลเพียงตัวเดียว ซึ่งทำการแสดงผลในส่วน GUI ส่วนการแสดงผลทางคอลโซลจะปนอยู่กับฟังชันค์ต่างๆ จึงได้ดึงส่วนแสดผลนี้ออกจากฟังค์ชันต่างๆเพื่อเป็นการปรับให้แต่ละส่วนทำงานของตน เมื่อเสร็จจากการปรับโครงสร้างเป็นมาใช้ Interface Output แล้ว ได้ทำการสร้างส่วนแสดงผล NullOutput, ComboOutput, ConsoleOutput และยกเลิกการใช้ Output_MVC เนื่องจากฟังค์ชันที่มากเกินจากสืบทอดมาจาก Interface OutputUniและอีกประการคือไม่สามารถดูข้อมูลของการส่งข้อมูลย้อนหลังได้ โดยโครงสร้างของส่วนแสดงผลตัวใหม่เป็นดังนี้



รูปที่ 3.10 แสดงโครงสร้างของส่วนแสดงผล

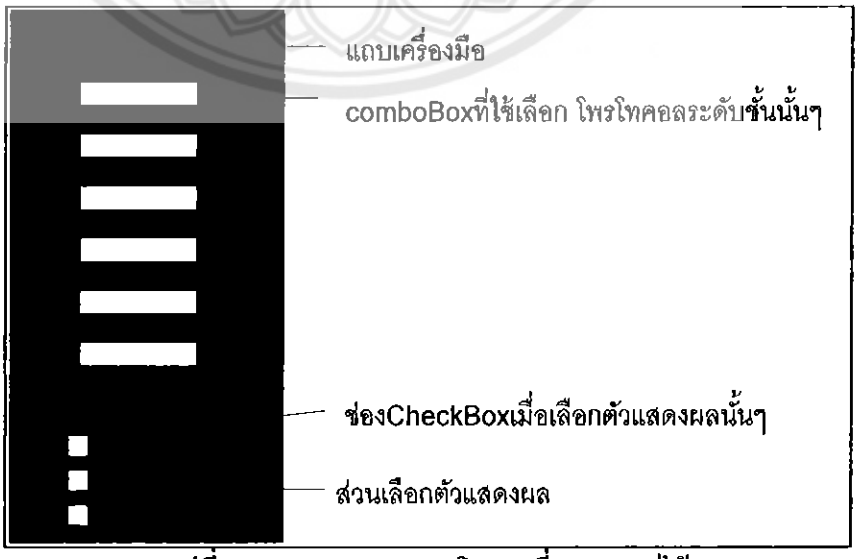
NullOutput และ ComboOutput เป็นเพียงตัวสร้างตัวแสดงผลที่หลากหลาย ตัวแสดงผลจริงในตอนนี้คือ ConsoleOutput เพียงตัวเดียว ดังนั้นจึงได้ทำการเพิ่มตัวแสดงผลอีกหลายตัวได้แก่ GUIOutput, WebOutput, MyFrameOutput, CardLayoutOutput, SingleFileOutput, MultiFileOutput



รูปที่ 3.11 แสดงโครงสร้างของส่วนแสดงผลทั้งหมด

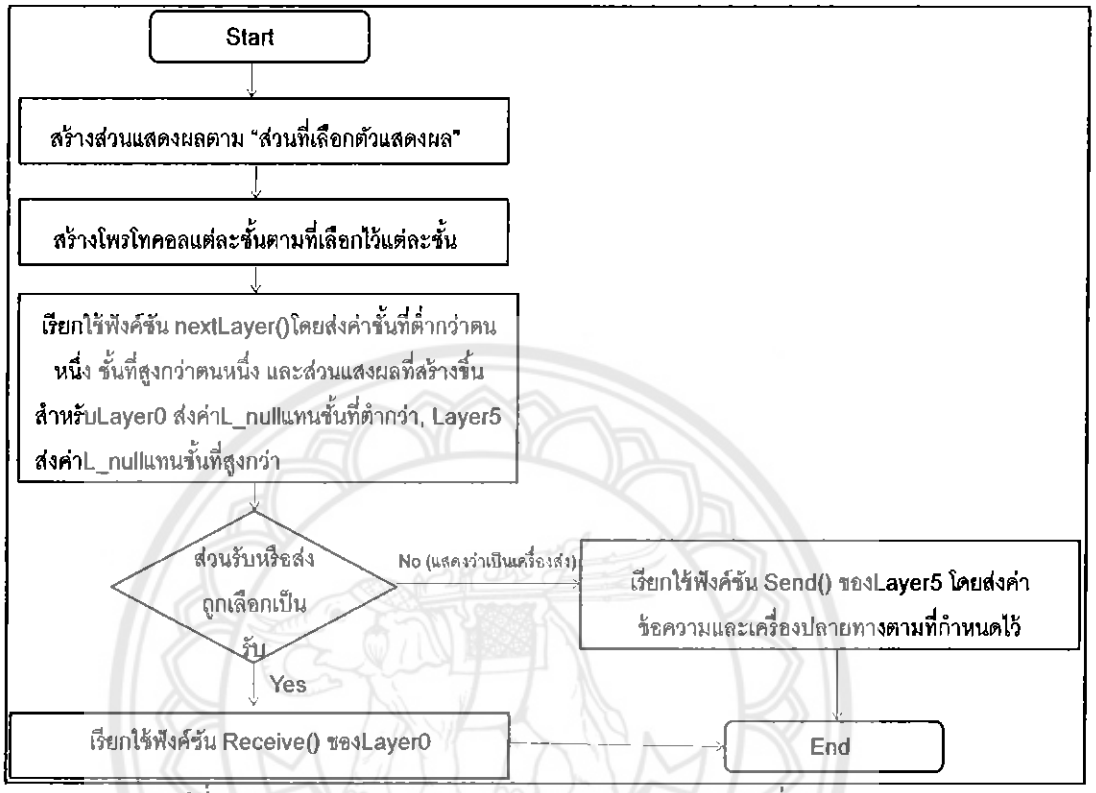
3.4 ออกแบบ และสร้าง GUI ที่แสดงโหนดแต่ละโหนด

เนื่องจาก GUI ของโปรแกรมเดิมไม่รองรับการเพิ่มโพรโทคอล และไม่รองรับการเพิ่มส่วนแสดงผลดังนั้นจึงได้ออกแบบ GUI ตัวใหม่ซึ่งรองรับการเพิ่มโพรโตคอล และ รองรับการเพิ่มส่วนแสดงผลโดยออกแบบไว้ดังนี้



รูปที่ 3.12 แสดง GUI ของโหนด ที่ออกแบบไว้

แถบเครื่องมือประกอบด้วย Combobox สำหรับเลือกว่าเป็นส่วนรับหรือส่ง ปุ่มที่ใส่ข้อความที่ต้องการส่งและโหนดปลายทาง ปุ่มเริ่มการทำงาน

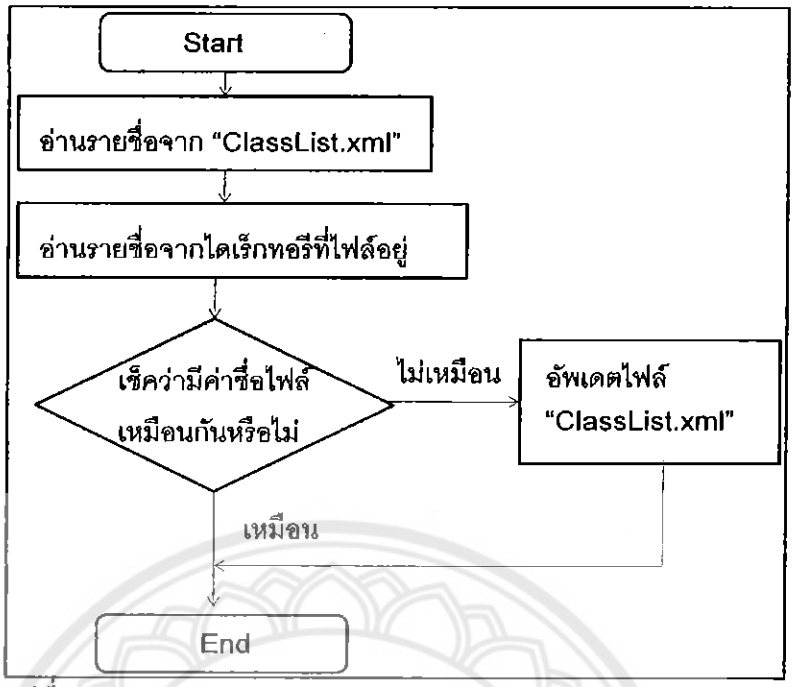


รูปที่ 3.13 Flowchart Diagram แสดงการทำงานของปุ่มเริ่มการทำงาน

หลังจากออกแบบเสร็จได้ทำการสร้าง GUI ตามที่ออกแบบไว้โดยการสร้างคลาส MyFrameซึ่งทำหน้าที่สร้าง GUI ตามที่ได้ออกแบบไว้

3.5 พัฒนาให้โปรแกรมสะดวกต่อการเพิ่มหรือลดส่วนแสดงผล

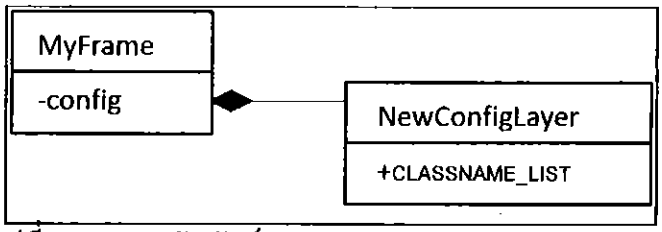
หลังจากออกแบบและสร้างตัว GUI เสร็จเรียบร้อยแล้ว ตอนนี้ตัว GUI รองรับการเพิ่มส่วนแสดงผล แต่การเพิ่มส่วนแสดงผลยังทำได้ยาก ดังนั้นจึงได้ทำการเพิ่มฟังค์ชั่นให้กับโปรแกรม โดยสร้าง NewConfigLayer ซึ่งทำหน้าที่หลักๆ อ่านชื่อคลาสจากไฟล์ ClassList.xml และทำการอ่านชื่อคลาสในสารบบที่ไฟล์อยู่ แล้วทำการเช็คกับว่ามีคลาสใหม่เพิ่มเข้ามาคลาสใดบ้าง และทำการสร้าง แก้ไข และเพิ่มเติมข้อมูลให้กับไฟล์ ClassList.xml



รูปที่ 3.14 Flowchart Diagram แสดงการทำงานของ NewConfigLayer

นอกจากนี้ ยังมีตัวแปร ArrayList<String> CLASSNAME_LIST[] ซึ่งถูกประกาศให้เป็น public โดยเป็นArray ที่มีขนาด 9 โดยแต่ละตัวเป็น ArrayList ที่ทำหน้าที่ดังนี้

- CLASSNAME_LIST[0] เป็น ArrayList ที่เก็บชื่อคลาสที่เป็นคลาสลูกของ I_Layer5
- CLASSNAME_LIST[1] เป็น ArrayList ที่เก็บชื่อคลาสที่เป็นคลาสลูกของ I_Layer4
- CLASSNAME_LIST[2] เป็น ArrayList ที่เก็บชื่อคลาสที่เป็นคลาสลูกของ I_Layer3
- CLASSNAME_LIST[3] เป็น ArrayList ที่เก็บชื่อคลาสที่เป็นคลาสลูกของ I_Layer2
- CLASSNAME_LIST[4] เป็น ArrayList ที่เก็บชื่อคลาสที่เป็นคลาสลูกของ I_Layer1
- CLASSNAME_LIST[5] เป็น ArrayList ที่เก็บชื่อคลาสที่เป็นคลาสลูกของ I_Layer0
- CLASSNAME_LIST[6] เป็น ArrayList ที่เก็บชื่อคลาสที่เป็นคลาสลูกของ Output
- CLASSNAME_LIST[7] เป็น ArrayList ที่เก็บชื่อคลาสที่นอกเหนือจากนั้นไม่รวม interfac
- CLASSNAME_LIST[8] เป็น ArrayList ที่เก็บชื่อคลาสที่เป็น interface



รูปที่ 3.15 ความสัมพันธ์ของ MyFrame กับ NewConfigLayer

จากคลาสดังกล่าวตัวโปรแกรมจะสามารถทราบได้ถึงชื่อของส่วนแสดงผลจาก CLASSNAME_LIST[6] นั้นเองโดย MyFrame ได้มี NewConfigLayer อยู่ และ MyFrame ก็ได้นำเอา CLASSNAME_LIST[6] ของ NewConfigLayer ไปสร้างให้มี checkbox ของ Output นั้นๆ และหากมีการเลือกใช้งานจะมาการแปลงจาก String เป็น คลาสโดยอาศัยคลาสที่ชื่อว่า Class ดังโค๊ดตัวอย่าง

```
Class theClass =Class.forName(OutputName);
OuputnewOuput = (Ouput)theClass.newInstance();
```

เมื่อOutputNameเป็นชื่อของคลาสเช่น "ConsoleOutput"เป็นต้น

3.6 พัฒนาให้โปรแกรมสะดวกต่อการเพิ่มหรือลดโพรโทคอลแต่ละชั้น

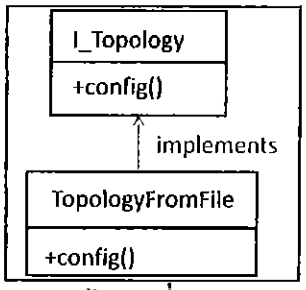
หลังจากการพัฒนาให้โปรแกรมสะดวกต่อการเพิ่มหรือลดส่วนแสดงผลจะเห็นได้มีการสร้างคลาส NewConfigLayer มีตัวแปร CLASSNAME_LIST ที่ใช้ในการเก็บชื่อคลาสที่สืบทอดมาจาก I_Layer5 – I_Layer0 ทำให้ทราบถึงถึงชื่อของโพรโทคอลต่างๆจาก CLASSNAME_LIST[0]-CLASSNAME_LIST[5] โดยคลาส Myframe ได้ใช้ในการสร้าง combobox ที่มีตัวเลือกเหล่านั้นเข้าไปและหากมีการเลือกใช้งานจะมาการแปลงจาก String เป็น คลาสโดยอาศัยคลาสที่ชื่อว่า Class ดังโค๊ดตัวอย่าง

```
Class theClass =Class.forName(protocalName);
I_LayernewProtocal=(I_Layer)theClass.newInstance();
```

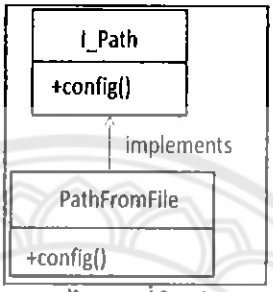
เมื่อ protocalName เป็นชื่อของคลาสเช่น "L5_HTTP" เป็นต้น

3.7 การพัฒนาให้โปรแกรมสามารถสามารถจำลองได้มากกว่าสองโหนด

หลังจากการพัฒนาให้โปรแกรมสะดวกต่อการเพิ่มหรือลดโพรโทคอลได้แล้ว และจากความรู้เกี่ยวกับเทรด (Tread) และความรู้เกี่ยวกับโพรโทคอลชั้นที่สองและสาม ทำให้มีการพัฒนาให้โปรแกรมสามารถส่งข้อมูลได้มากกว่าสองโหนด ซึ่งต้องมีการเพิ่มไลบรารี่ในส่วนทอพอโลยีและเส้นทางโดยได้มีการสร้าง Interface I_Topology, Interface I_Path, class TopologyFromFile ซึ่ง Implements I_Topology, class PathFromFile ซึ่ง Implements I_Path

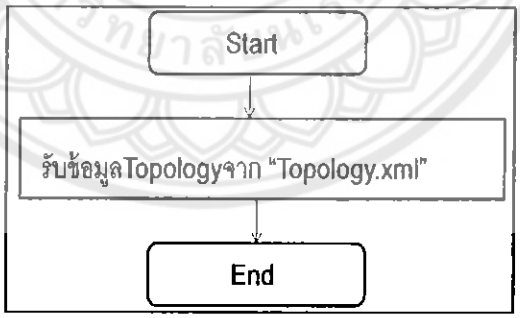


รูปที่ 3.16 แสดงไลบรารี่ในส่วนของทอพอโลยี

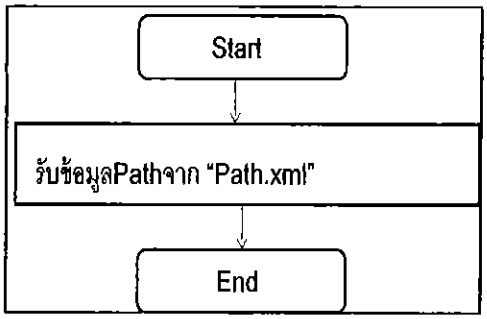


รูปที่ 3.17 แสดงไลบรารี่ในส่วนของเส้นทาง

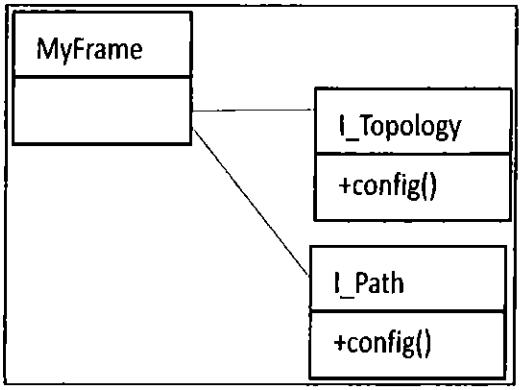
นอกจากไฟล์นามสกุลจาวาที่กล่าวมาแล้ว ยังสร้าง Topology.xml, Path.xml โดยที่ Topology.xml เป็นไฟล์นามสกุล xml ที่เก็บข้อมูลของทอพอโลยีและ Path.xml เป็นไฟล์นามสกุล xml ที่เก็บข้อมูลของเส้นทาง



รูปที่ 3.18 Flowchart Diagram แสดงการทำงานของคลาส TopologyFromFile



รูปที่ 3.19 Flowchart Diagram แสดงการทำงานของคลาส PathFromFile



รูปที่ 3.20 Diagram แสดงการการติดต่อกันของ GUI กับทอพอโลยีและเส้นทาง

3.8 พัฒนาให้โปรแกรมสามารถรอรับข้อมูลได้หลายทิศทาง

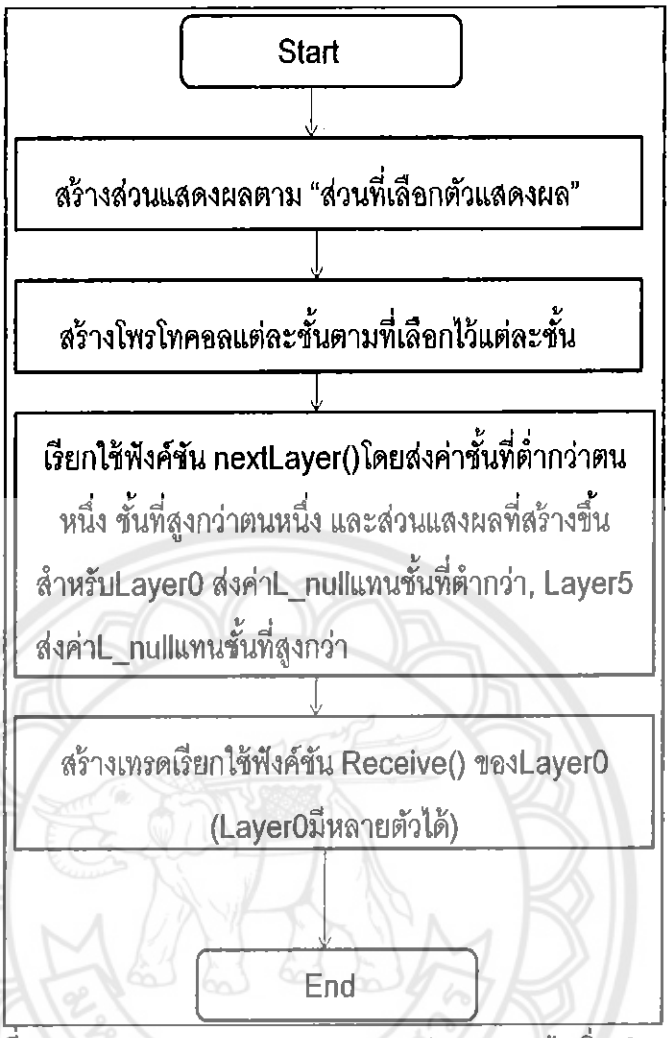
หลังจากพัฒนาโปรแกรมสามารถส่งได้มากกว่าสองโหนดแล้ว เราไม่สามารถรู้ได้ว่าข้อมูลจะถูงส่งมาจากช่องทางใด ดังนั้นใน Layer0 จึงควรรอรับข้อมูลจากทุกทางจากปัญหาดังกล่าว จึงได้นำความรู้เรื่องเทรดมาใช้ในการแยกตัว GUI และการส่งข้อมูลของโพรโทคอล อยู่คนละเทรดกัน

ปัญหาของโปรแกรมที่เป็นเทรดเชิงเดี่ยวยังทำให้ไม่สามารถรอรับข้อมูลจาก Layer0 ได้หลายตัว อันเนื่องมาจากหากมีการรอรับข้อมูลจาก Layer0 ตัวหนึ่งแล้วถึงแม้ Layer0 ตัวอื่นมีการรับข้อมูลเข้ามาโปรแกรมก็ไม่สามารถสลับมาทำงานให้ตัว Layer0 อีกตัวได้

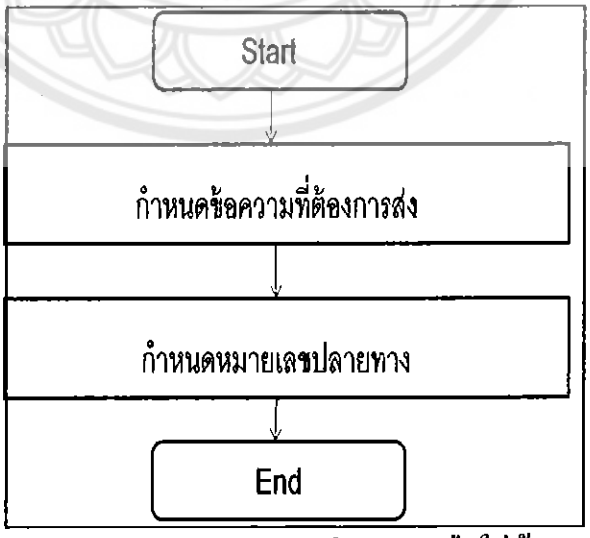
จากปัญหาดังกล่าว จึงได้นำความรู้เรื่องเทรดมาใช้ให้ Layer0 แต่ละตัวทำงานบนคนละเทรดกัน

เนื่องจากนำความรู้เรื่องเทรดมาใช้ทำให้ทุกโหนดสามารถส่งได้หมดโดยไม่เน้นว่าโหนดใดเป็นตัวรับหรือตัวส่ง จึงได้ทำการแก้ไข GUI จากเดิมที่แถบเครื่องมือมีส่วนกำหนดตัวรับหรือส่งโดยได้นำส่วนนี้ออกและแก้ไขการทำงานของปุ่มเริ่มทำงานใหม่

ปุ่มเริ่มทำงานมีการแก้ไขเกี่ยวกับตัวรับหรือส่ง และยังมีการนำส่วนส่งข้อมูลที่อยู่ในตัวปุ่มเริ่มทำงานซึ่งส่วนส่งข้อมูลได้ย้ายไปอยู่ในส่วนของปุ่มที่ใส่ข้อมูลที่ต้องการส่ง



รูปที่ 3.21 Flowchart Diagram แสดงการทำงานของปุ่มเริ่มทำงาน

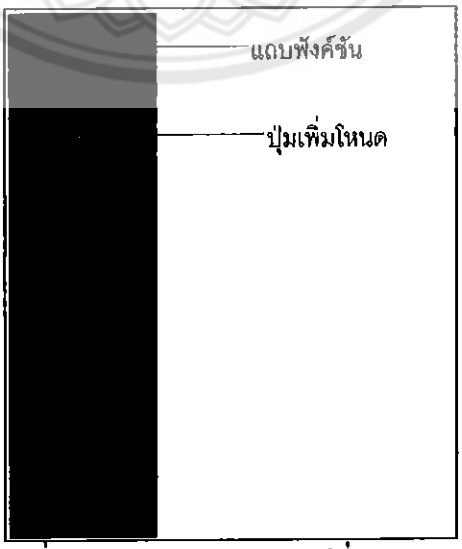


รูปที่ 3.22 Flowchart Diagram แสดงการทำงานของปุ่มใส่ข้อมูลการส่งเดิม

รูปที่ 3.23 Flowchart Diagram แสดงการทำงานของปุ่มใส่ข้อมูลการส่งใหม่

3.9 เพิ่มส่วน GUI ของทอพอโลยี

หลังจากพัฒนาโปรแกรมโดยใช้ความรู้เกี่ยวกับเทรดแล้ว ถึงตอนนี้โปรแกรมสามารถทำงานและแสดงผลได้ดีระดับหนึ่ง แต่ตัวทอพอโลยียังยุ่งยากต่อการใช้งาน เนื่องจาก ผู้ใช้ต้องเปิดไฟล์ Topology.xml มาทำการแก้ไข ผู้จัดทำเห็นว่าการนำเสนอทอพอโลยีในรูปแบบตารางเมตริกซ์จะทำให้ผู้ใช้เห็นภาพการเชื่อมโยงกันของทอพอโลยีมากกว่า อีกทั้งยังสามารถทำการพัฒนาต่อได้อีกด้วย ดังนั้นจึงทำการออกแบบและสร้าง GUI ที่คล้ายกับการเขียนทอพอโลยีในรูปแบบเมตริกซ์เพื่อใช้ในการสร้างทอพอโลยี



รูปที่ 3.24 GUI ของทอพอโลยีที่ออกแบบ

โดยแถบฟังค์ชันจะประกอบด้วยฟังค์ชันต่างๆ ไม่ว่าจะเป็น New (การสร้างทอพอโลยีใหม่) Load (การเปิดทอพอโลยี) Save (การบันทึกทอพอโลยี) Build (การสร้างทอพอโลยีลงในทอพอโลยีของโปรแกรม) Run (การเปิดและตั้งค่าโหนดตามทอพอโลยีของโปรแกรม) Exit (ออก)

และเมื่อกดปุ่มAdd Node จะมีหน้าต่างรับค่าข้อมูลของโหนดออกมาเพื่อรับค่าข้อมูลของโหนดที่ต้องการเพิ่ม คือชื่อโหนด

รูปที่ 3.25 GUI ของทอพอโลยี ที่ออกแบบหลังทำการเพิ่มโหนด A

พื้นที่ช่องทางที่ติดต่อหากไม่ใช่ของโหนดเดียวกันเช่น A กับ B จะต้องสามารถเพิ่มช่องทางได้ โดยต้องใส่ข้อมูลของช่องทางอันประกอบด้วยหมายเลขช่องทาง

หลังจากออกแบบ GUI ของทอพอโลยีโดยมีส่วนประกอบและรายละเอียดต่างๆดังที่กล่าวมาเสร็จจึงได้ทำการสร้าง GUI ของทอพอโลยี ตามที่ออกแบบไว้และได้พัฒนาเป็นแบบ GUI ที่มีด้านหลักเป็นโหนด และด้านแถวเป็นบัส

3.10 ตรวจสอบการใช้งานจริง

เมื่อปรับปรุงโปรแกรมเสร็จสิ้นแล้ว ได้ทำการตรวจสอบการใช้งานจริงโดยผู้จัดทำโปรแกรมเอง และได้ผู้ที่สนใจและเรียนเนื้อหาเกี่ยวกับโพรโทคอลได้ใช้งานโปรแกรม เพื่อนำข้อเสนอแนะไปสรุปผลต่อไป

3.11 จัดทำคู่มือโครงงาน

การจัดทำรายงาน โดยได้ศึกษาหาความรู้เกี่ยวกับรูปเล่มรายงานจาก รูปแบบรายงานของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวรและปริญญานิพนธ์ในห้องสมุด คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

มหาวิทยาลัยนเรศวร

บทที่ 4
ผลการทดลอง

หลังจากศึกษาและทำการออกแบบส่วนประกอบต่างๆ และดำเนินการการพัฒนาโปรแกรมเป็นอันเสร็จสิ้นแล้ว ได้นำโปรแกรมของโครงงานที่ทำมาทดสอบ เพื่อตรวจสอบว่าโปรแกรมเป็นไปตามเป้าหมายที่วางไว้ในโครงงานหรือไม่ โดยผลการทดสอบโปรแกรมเป็นส่วนหนึ่งของผลการทดลอง เพื่อปรับปรุงโปรแกรมให้มีข้อผิดพลาดน้อยลง

4.1 การทดสอบโปรแกรม

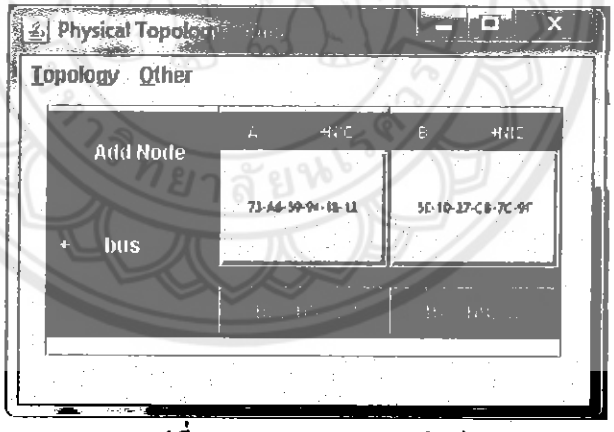
4.1.1 ทดสอบการแสดงผลของ GUI ของทอพอโลยี

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบการแสดงผลของ GUI ของทอพอโลยี

รายละเอียด เปิดหน้าต่าง cmd, คำสั่ง java MainPlanePhysicalTopology

ผลที่คาดหวัง GUI ของทอพอโลยี



รูปที่ 4. 1 GUI ของทอพอโลยี

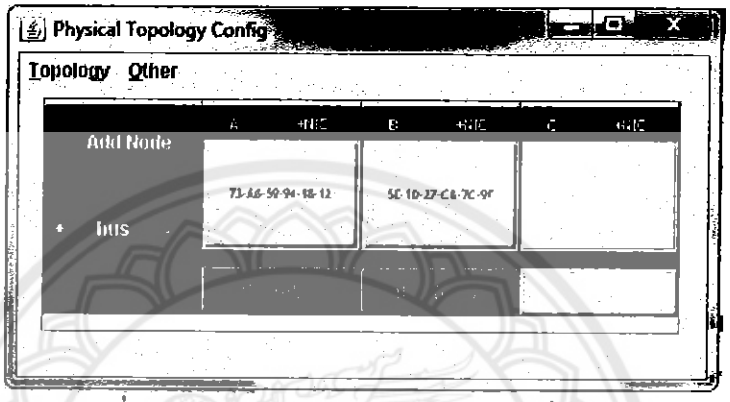
4.1.2 ทดสอบการแสดงผลของ GUI ของทอพอโลยี เมื่อเพิ่มโหนด

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบการแสดงผลของ GUI เมื่อเพิ่มโหนด

รายละเอียด ต่อจาก 4.1.1, กดปุ่ม Add Node

ผลที่คาดหวัง ทอพอโลยีมีการเพิ่มโหนดใหม่เข้าไป



รูปที่ 4.2 GUI ของทอพอโลยี หลังทำการเพิ่มโหนด

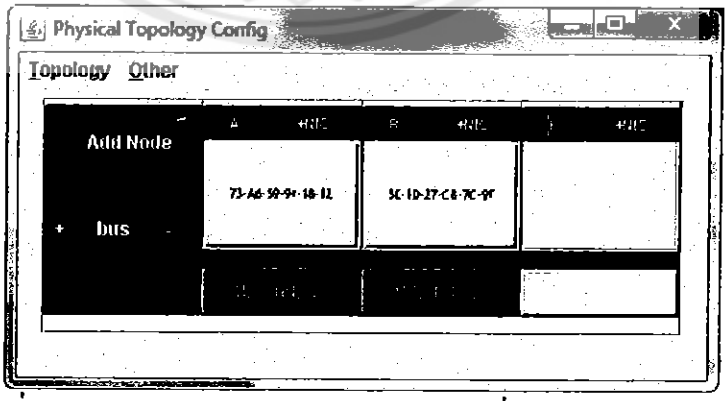
4.1.3 ทดสอบการแสดงผลของ GUI ของทอพอโลยี เมื่อแก้ไขชื่อโหนด

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบการแสดงผลของ GUI ของทอพอโลยี เมื่อแก้ไขชื่อโหนด

รายละเอียด ต่อจาก 4.1.2, กดปุ่ม C, ใส่ชื่อใหม่เป็น K

ผลที่คาดหวัง ทอพอโลยี มีการแก้ไขชื่อโหนด



รูปที่ 4.3 GUI ของทอพอโลยี หลังทำการแก้ไขชื่อโหนดจาก C เป็น K

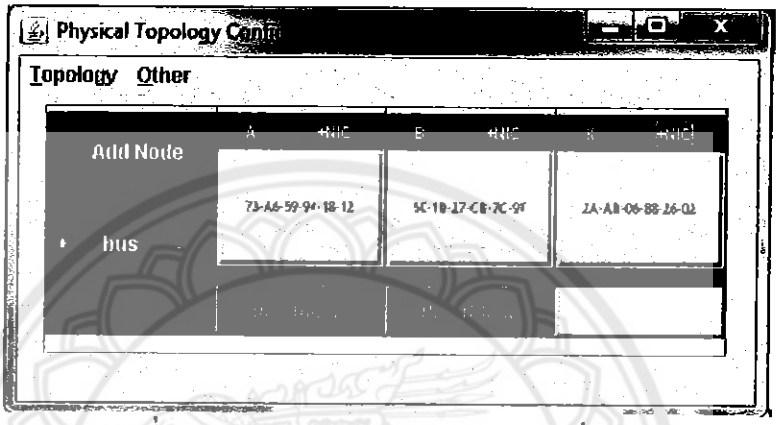
4.1.4 ทดสอบการแสดงผลของ GUI เมื่อเพิ่ม NIC Card

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบการแสดงผลของ GUI เมื่อเพิ่ม NIC

รายละเอียด ต่อจาก 4.1.3, ที่โหนด K กดปุ่ม +NIC

ผลที่คาดหวัง GUI ของทอพอโลยี ที่ K เพิ่ม NIC



รูปที่ 4.4 GUI ของทอพอโลยี หลังทำการเพิ่ม NIC

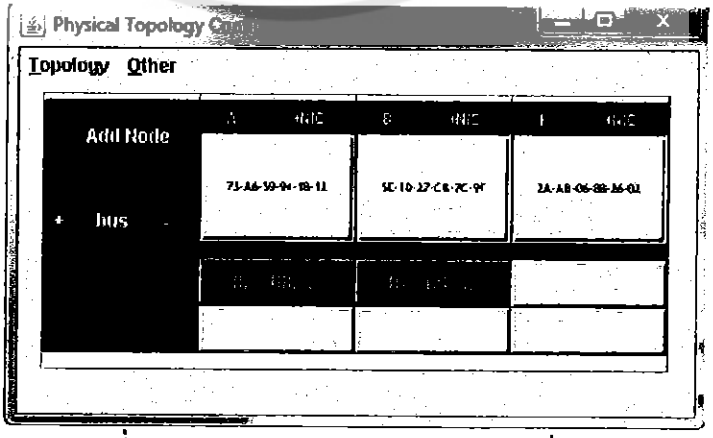
4.1.5 ทดสอบการแสดงผลของ GUI เมื่อเพิ่ม bus

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบการแสดงผลของ GUI เมื่อเพิ่ม Bus

รายละเอียด ต่อจาก 4.1.4, กดปุ่ม + ที่อยู่หน้าปุ่ม Bus

ผลที่คาดหวัง GUI ของทอพอโลยีมีการเพิ่ม Bus



รูปที่ 4.5 GUI ของทอพอโลยี หลังทำการเพิ่ม Bus

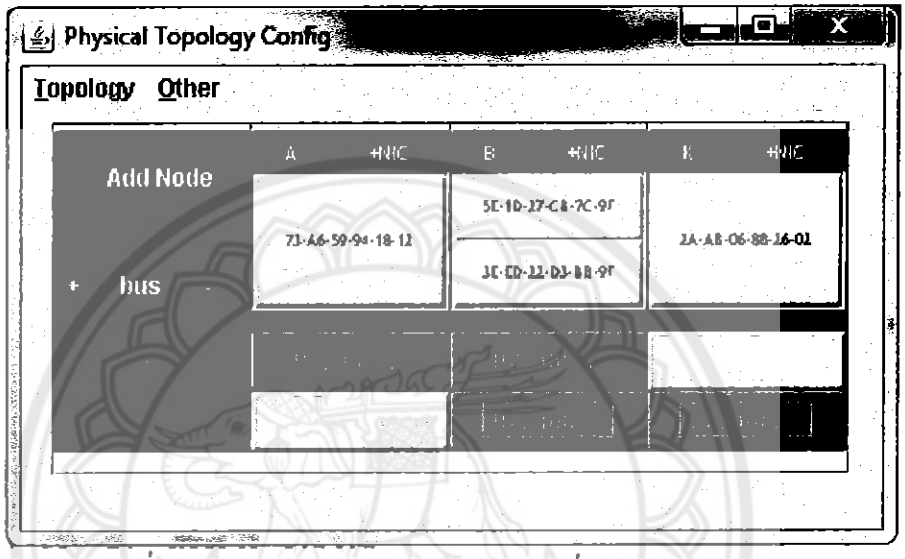
4.1.6 ทดสอบการแสดงผลของ GUI เมื่อเชื่อมกับ Bus

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบการแสดงผลของ GUI เมื่อเชื่อมต่อกับ Bus

รายละเอียด ต่อจาก 4.1.5, กดเพิ่ม +NIC ของ B กดที่บัสแถวที่สอง หลักที่ B กับ K

ผลที่คาดหวัง GUI ของทอพอโลยีที่มีการเพิ่มการติดต่อกับ bus



รูปที่ 4.6 GUI ของทอพอโลยี หลังทำการเพิ่มการติดต่อกับ Bus

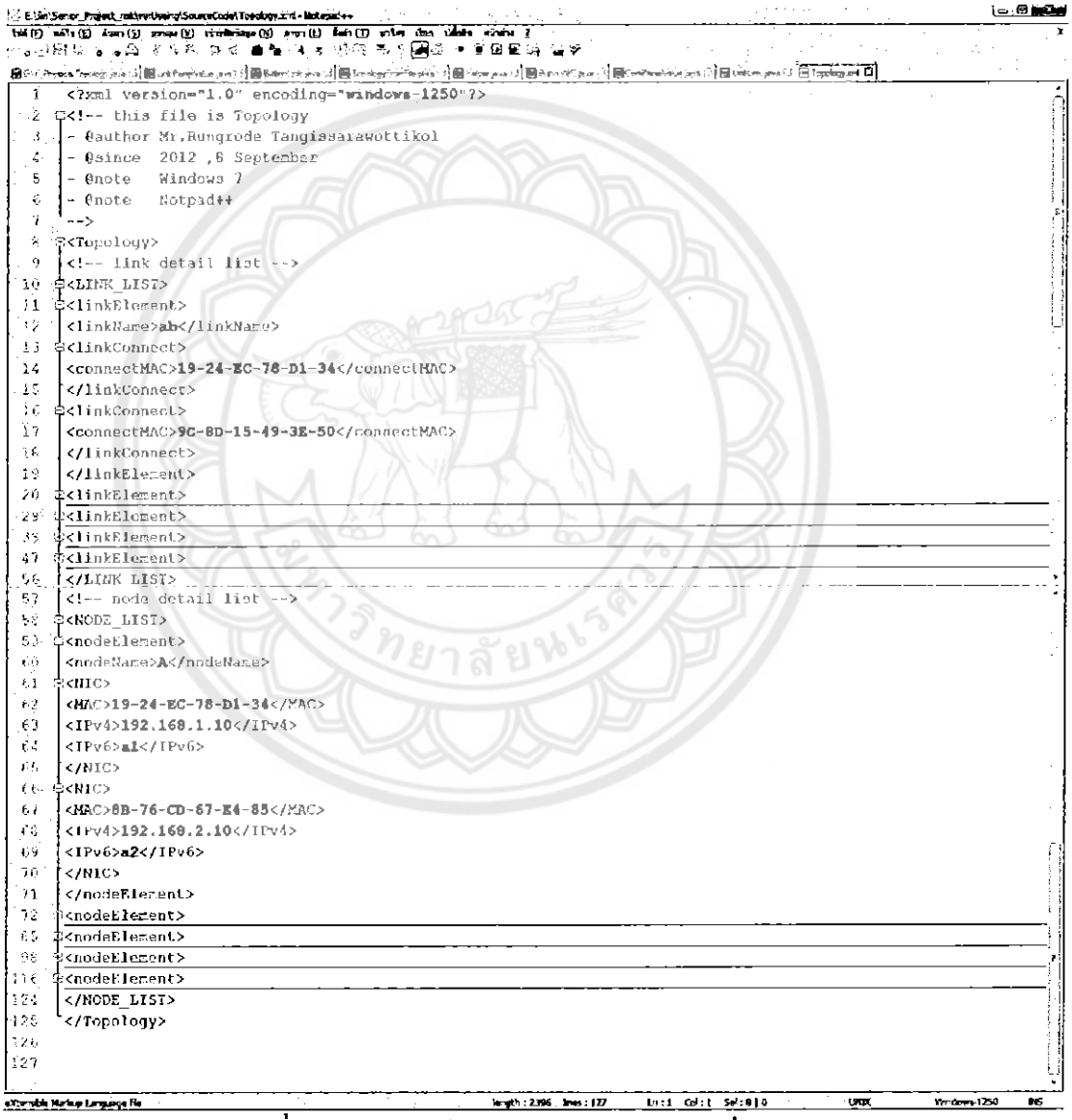
4.1.7 ทดสอบความถูกต้องของการทำงานคำสั่ง Build

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการทำงานคำสั่ง Build

รายละเอียด ลองสร้าง Topology, กดชุดคำสั่ง Other เลือกคำสั่ง Build

ผลที่คาดหวัง ไฟล์ Topology.xml ในตัวโปรแกรมมีการเปลี่ยนแปลงเป็นทอพอโลยีซึ่งได้ทำการออกแบบไว้



รูปที่ 4.7 แสดงไฟล์ Topology.xml หลังใช้คำสั่ง Build

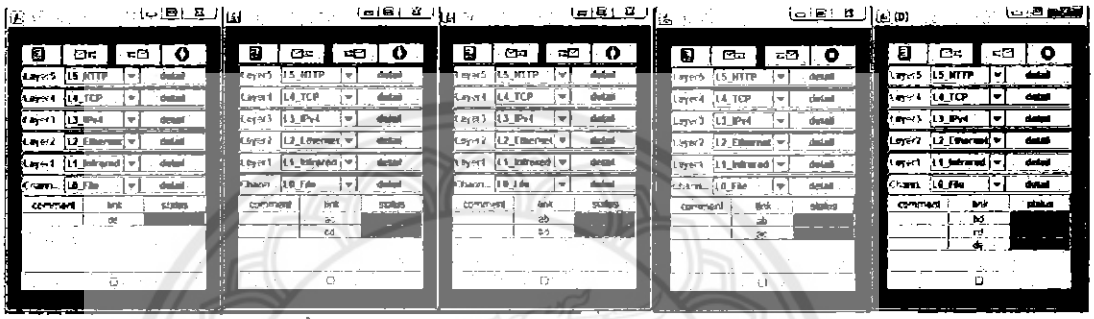
4.1.8 ทดสอบความถูกต้องของการทำงานคำสั่ง Run

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการทำงานคำสั่ง Run

รายละเอียด ต่อจาก 4.1.6, กดชุดคำสั่ง Other เลือกคำสั่ง Run

ผลที่คาดหวัง โปรแกรมเปิดตัวโหนดตามจำนวนโหนดที่ออกแบบไว้และมีช่องทางตามที่กำหนดไว้



รูปที่ 4.8 GUI ในของโหนดต่างๆ หลังใช้คำสั่ง Run

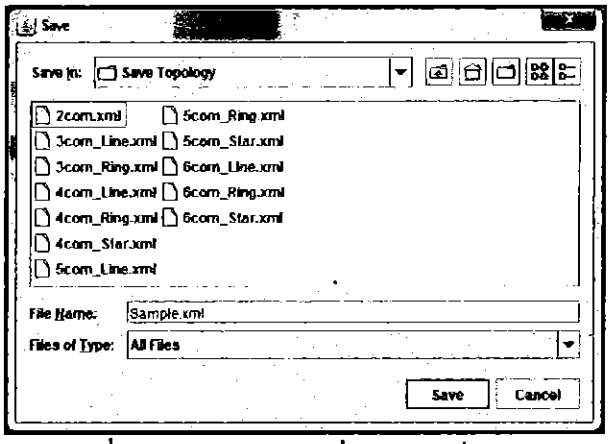
4.1.9 ทดสอบความถูกต้องของการทำงานคำสั่ง Save

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการทำงานคำสั่ง Save

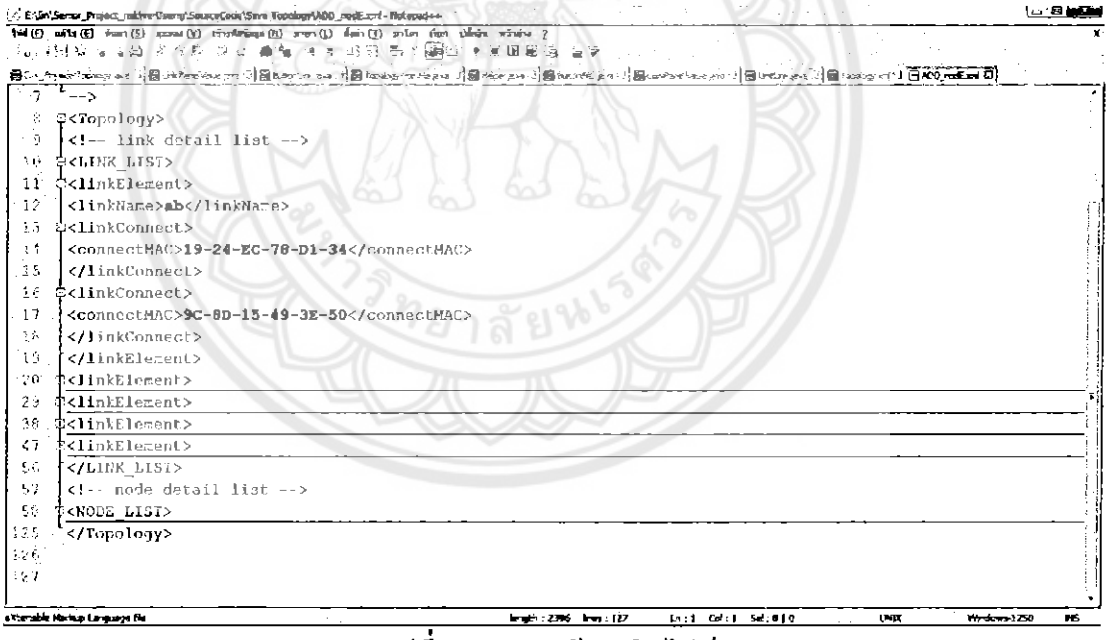
รายละเอียด ต่อจาก 4.1.7, กดชุดคำสั่ง Topology เลือกคำสั่ง Save เลือกที่อยู่ของไฟล์ (ที่เดิม) พิมพ์ชื่อไฟล์เป็น Sample.xml

ผลที่คาดหวัง ได้ไฟล์ตามที่อยู่และชื่อที่กำหนด และในตัวไฟล์มีข้อมูลของทอพอโลยีที่ออกแบบไว้



รูปที่ 4.9 แสดงหน้าต่างเมื่อเลือกคำสั่ง Save

รูปที่ 4.10 แสดงไฟล์ที่ได้ หลังใช้คำสั่ง Save



รูปที่ 4.11 แสดงข้อมูลในไฟล์

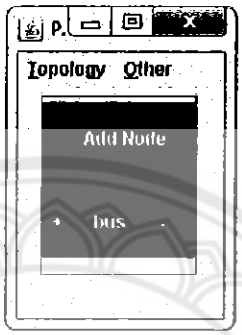
4.1.10 ทดสอบความถูกต้องของการทำงานคำสั่ง New

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการทำงานคำสั่ง New

รายละเอียด ต่อจาก 4.1.9, กดชุดคำสั่ง Topology เลือกคำสั่ง New

ผลที่คาดหวัง ได้ทอพอโลยีตัวใหม่ ยังไม่มีช่องทางและโหนด



รูปที่ 4.12 GUI ที่ใช้ออกแบบทอพอโลยี หลังใช้คำสั่ง New

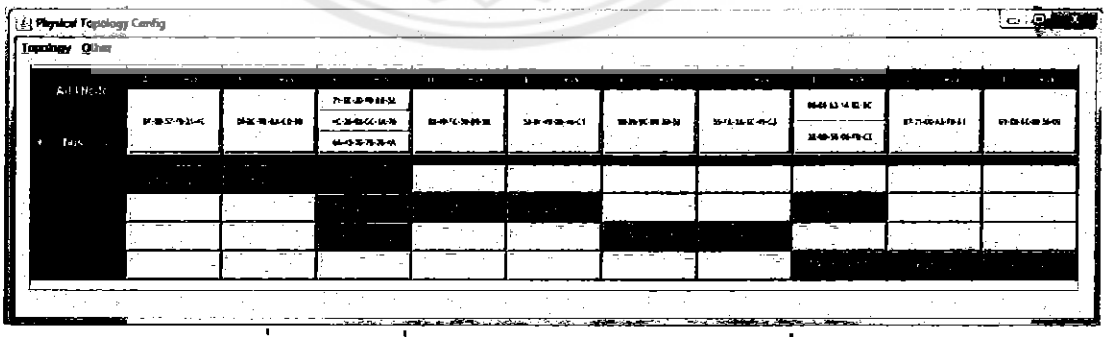
4.1.11 ทดสอบความถูกต้องของการทำงานคำสั่ง Open

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการทำงานคำสั่ง Open

รายละเอียด ต่อจาก 4.1.10, กดชุดคำสั่ง Topology เลือกคำสั่ง Open เลือกชื่อไฟล์ที่ต้องการ (TestTopology.xml)

ผลที่คาดหวัง GUI แสดงทอพอโลยีที่ออกแบบไว้ตามไฟล์ที่เลือก



รูปที่ 4.13 GUI ที่ใช้ออกแบบทอพอโลยี หลังใช้คำสั่ง Open

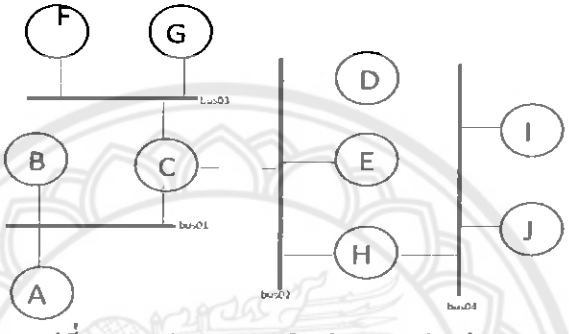
4.1.12 ทดสอบความถูกต้องการส่งข้อมูล

การทดสอบครั้งที่ 1

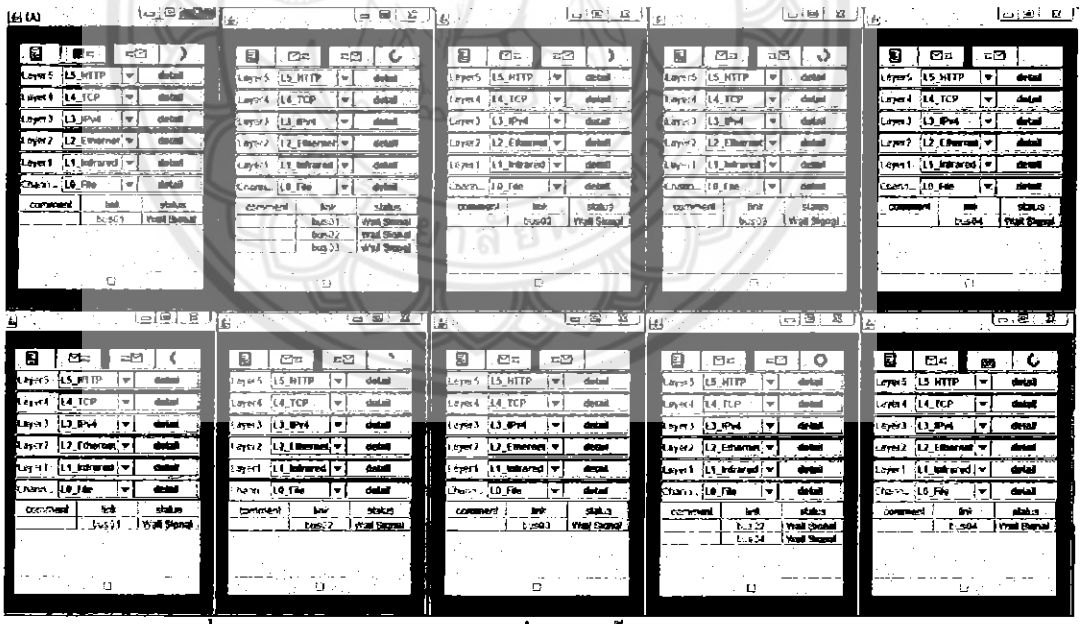
วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการส่งข้อมูล

รายละเอียด ต่อจาก 4.1.11, Build, Run, จากนั้นเซตทุกโหนดให้ทำงานโดยเลือกโหนด A แสดงผลทาง Console, Web_From, GUI, Frame และ file, ทำการส่งข้อมูลจาก A ไป J

ผลที่คาดหวัง สามารถทำการส่งข้อมูลได้ถูกต้องจาก A ไป J โดย J ได้รับข้อมูลที่ถูกต้อง



รูปที่ 4.14 รูปแสดงการติดต่อของแต่ละโหนด



รูปที่ 4.15 รูปแสดงโหนดต่างๆเมื่อเสร็จสิ้นการส่งข้อมูลจาก A ไป J

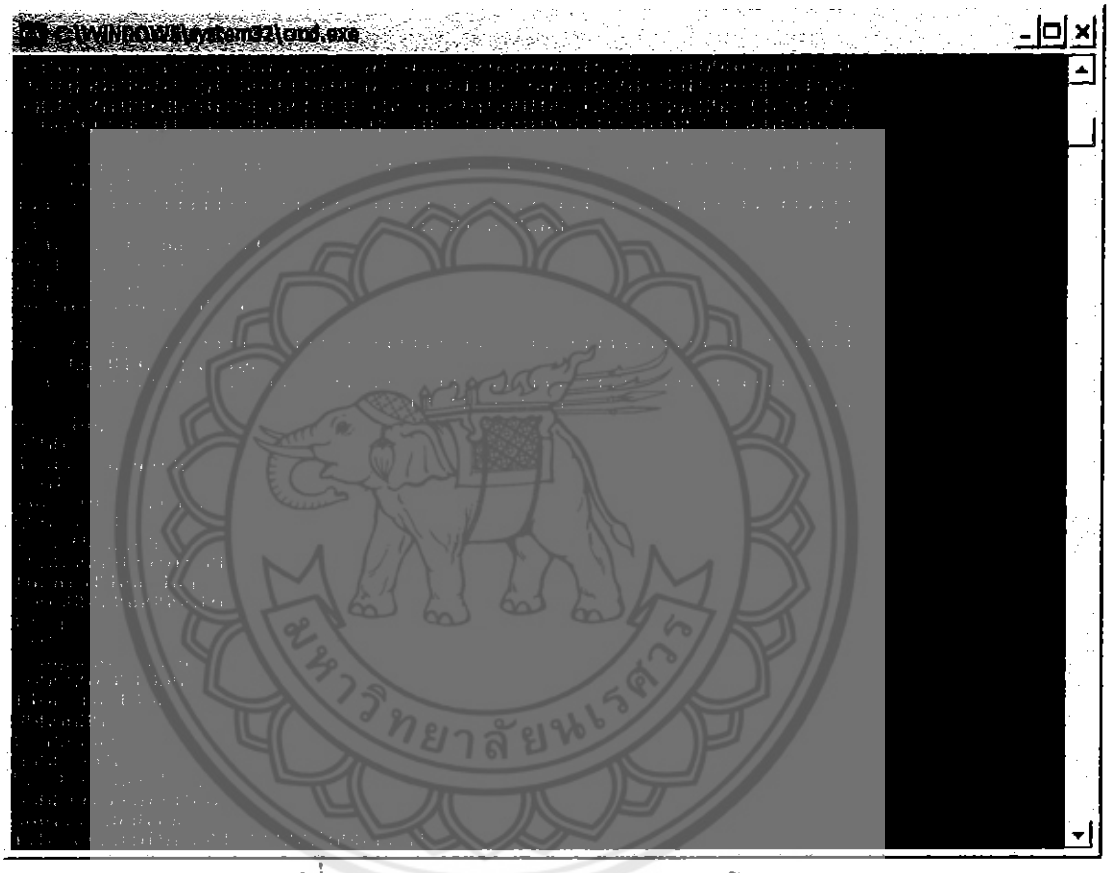
4.1.13 ทดสอบการแสดงผลทาง Console

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการแสดงผลทาง ConsoleOutput

รายละเอียด ต่อจาก 4.1.12, ตรวจสอบการแสดงผลทาง Console ของ A

ผลที่คาดหวัง โปรแกรมสามารถแสดงผลออกทาง Console ได้ถูกต้อง



รูปที่ 4.16 การแสดงผลทาง Console ของโหนด A

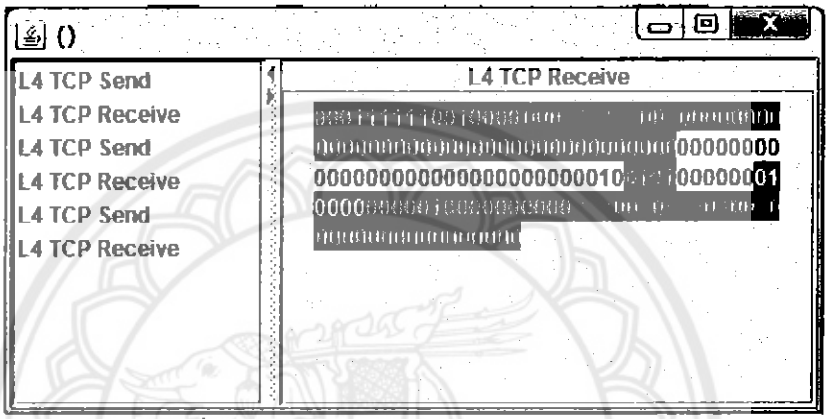
4.1.14 ทดสอบการแสดงผลทาง GUI

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการแสดงผลทาง GUI

รายละเอียด ต่อจาก 4.1.12, ตรวจสอบการแสดงผลทาง GUI ของ A โดยกดปุ่มหลังโพโทคอลโดยเลือกกด ชั้นที่ 4

ผลที่คาดหวัง โปรแกรมสามารถแสดงผลออกทาง GUI ได้ถูกต้อง



รูปที่ 4.17 การแสดงผลทาง GUI ของโหนด A

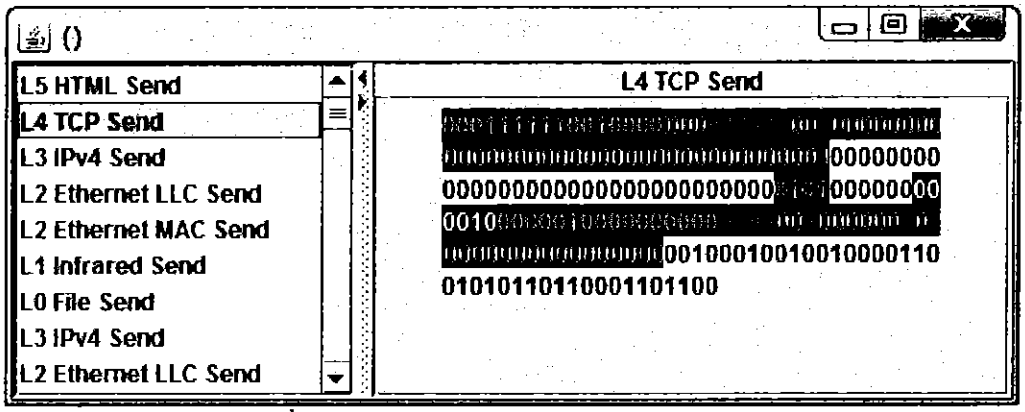
4.1.15 ทดสอบการแสดงผลทาง Frame

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการแสดงผลทาง Frame

รายละเอียด ต่อจาก 4.1.12, ตรวจสอบการแสดงผลทาง Frame โดยจะแสดงทุกชั้นของ A รวมกัน เพื่อที่จะได้เห็นถึงลำดับของเหตุการณ์

ผลที่คาดหวัง โปรแกรมสามารถแสดงผลออกทาง Frame ได้ถูกต้อง



รูปที่ 4.18 การแสดงผลทาง Frame ของโหนด A

4.1.16 ทดสอบการแสดงผลทาง WebForm

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการแสดงผลทาง WebForm

รายละเอียด ต่อจาก 4.1.12, ตรวจสอบการแสดงผลทาง WebForm

ผลที่คาดหวัง โปรแกรมสามารถแสดงผลออกทาง WebForm ซึ่งบันทึกข้อมูลการแสดงผลลงไฟล์นามสกุล html เพื่อเก็บไว้ศึกษาได้

รูปที่ 4.19 ไฟล์ที่ได้จากการแสดงผลทาง Web From (index.html)

รูปที่ 4.20 ไฟล์ที่ได้จากการแสดงผลทาง Web From ในรายละเอียดแต่ละส่วน

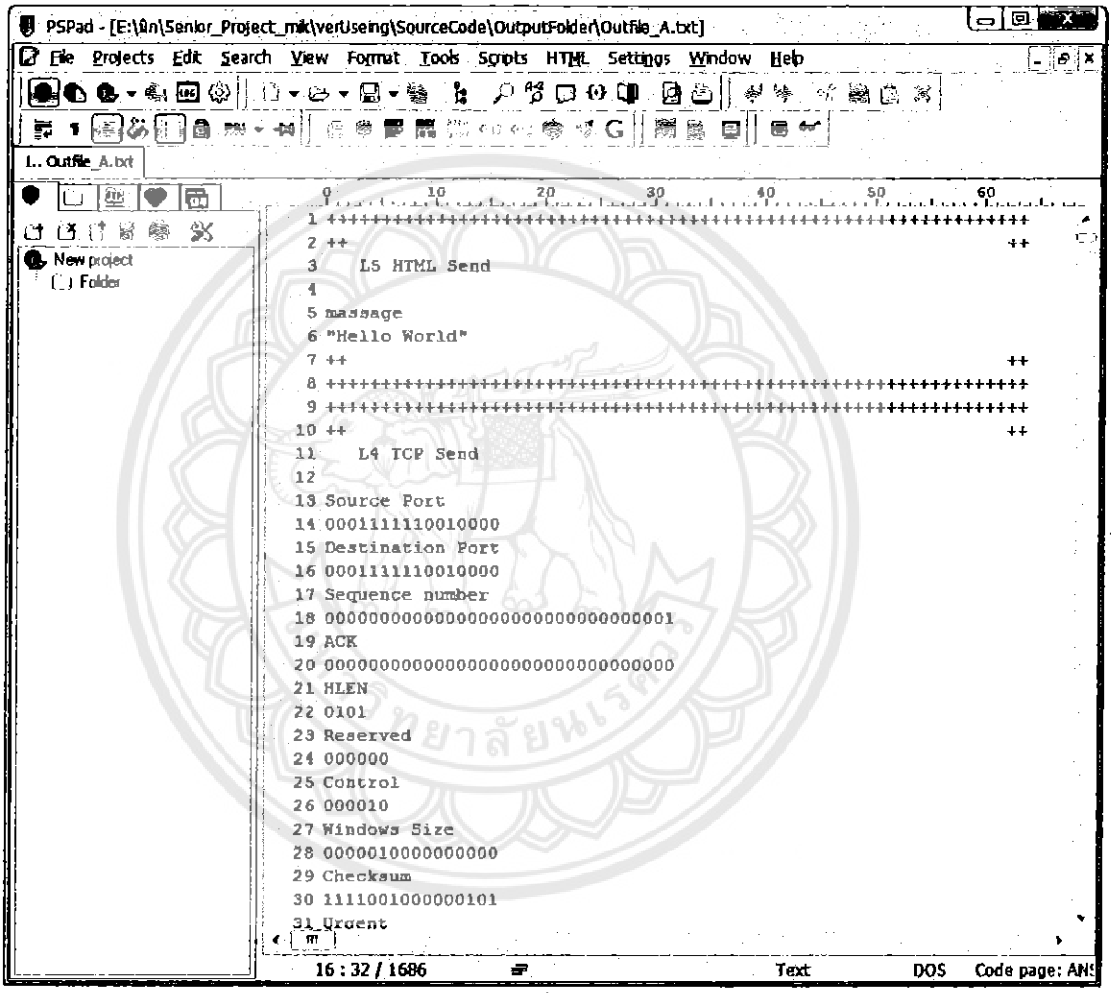
4.1.17 ทดสอบการแสดงผลทาง FileOutput ประเภท Single File

การทดสอบครั้งที่ 1

วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการแสดงผลทาง File

รายละเอียด ต่อจาก 4.1.12, ตรวจสอบการแสดงผลทาง File

ผลที่คาดหวัง โปรแกรมสามารถแสดงผลออกทาง File ซึ่งบันทึกข้อมูลการแสดงผลลงไฟล์นามสกุล txt เพื่อเก็บไว้ศึกษาได้



รูปที่ 4.21 ข้อมูลในส่วนเริ่มต้นของไฟล์ Output_A

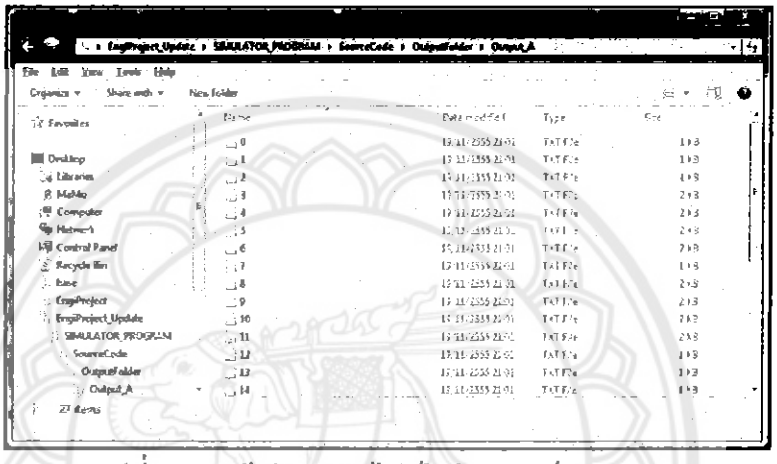
4.1.18 ทดสอบการแสดงผลทาง FileOutput ประเภท Multi File

การทดสอบครั้งที่ 1

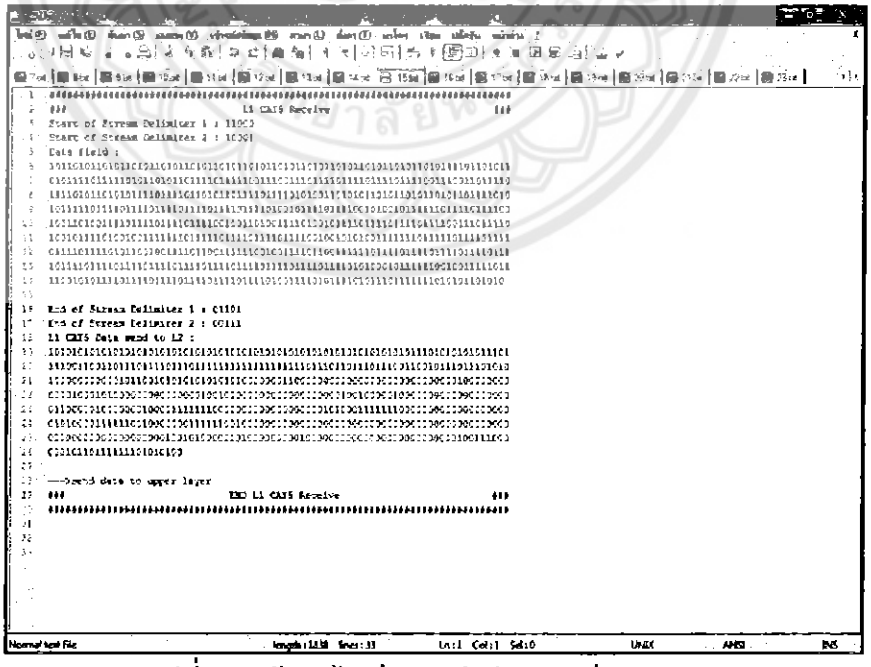
วัตถุประสงค์ เพื่อทดสอบความถูกต้องของการแสดงผลทาง File

รายละเอียด ต่อจาก 4.1.12, ตรวจสอบการแสดงผลทาง File

ผลที่คาดหวัง โปรแกรมสามารถแสดงผลออกทาง File ซึ่งบันทึกข้อมูลการแสดงผลลงไฟล์นามสกุล txt เพื่อเก็บไว้ศึกษาได้



รูปที่ 4.22 หน้าต่างแสดงไฟล์ในโฟลเดอร์ Output_A



รูปที่ 4.23 ข้อมูลไฟล์ 15.txt ในโฟลเดอร์ Output_A

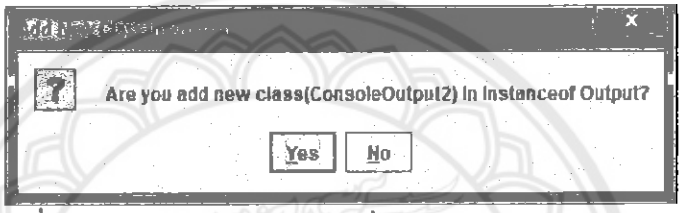
4.1.19 ทดสอบการแสดงผลของ GUI เมื่อทำการเพิ่มส่วนแสดงผล

การทดสอบครั้งที่ 1

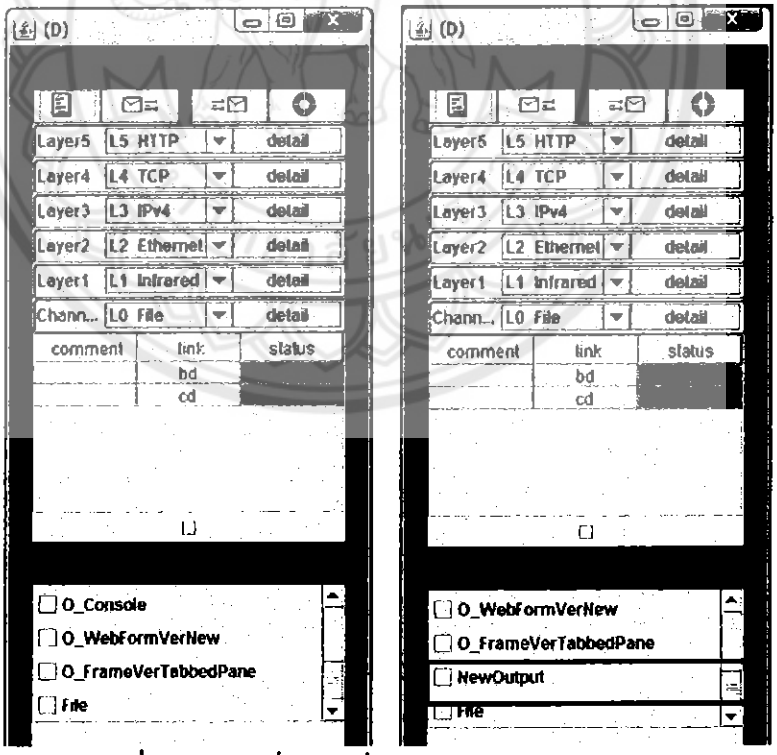
วัตถุประสงค์ เพื่อทดสอบความถูกต้องของ GUI เมื่อทำการเพิ่มส่วนแสดงผล

รายละเอียด สร้างคลาส NewOutput ซึ่งเป็นส่วนแสดงผลตัวใหม่, เปิดตัวออกแบบทอพอโลยี แล้วเลือกคำสั่ง Run สังเกตุตัวเลือกส่วนแสดงผลของแต่ละโหนด,

ผลที่คาดหวัง โปรแกรมสามารถสามารถทำการเพิ่มส่วนแสดงผลได้โดยมีข้อความแจ้งเตือนการเพิ่มส่วนแสดงผล เมื่อกดตกลง GUI ของโปรแกรมทำการเพิ่มตัวเลือก NewOutpu ที่ส่วนควบคุมส่วนแสดงผล



รูปที่ 4.24 ข้อความแจ้งเตือนการเพิ่มส่วนแสดงผล NewOutput



รูปที่ 4.25 GUI ที่มีการเพิ่มส่วนแสดงผล NewOutput

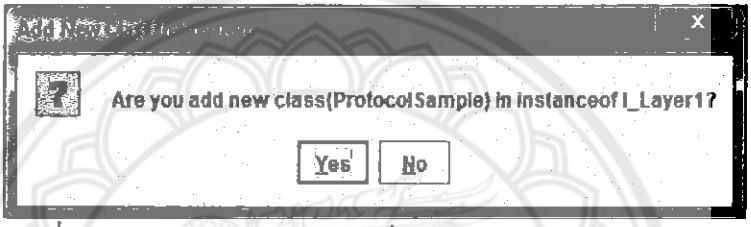
4.1.20 ทดสอบการแสดงผลของ GUI เมื่อทำการเพิ่มโพโทคอล

การทดสอบครั้งที่ 1

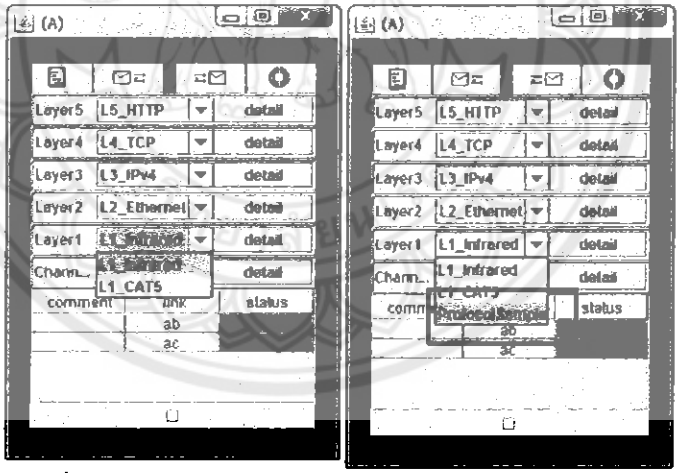
วัตถุประสงค์ เพื่อทดสอบความถูกต้องของ GUI เมื่อทำการเพิ่มโพโทคอล

รายละเอียด สร้างคลาส ProtocolSample ซึ่งเป็นโพโทคอลใหม่ อยู่ชั้นที่ 1, เปิดตัวออกแบบทอพอโลยี แล้วเลือกคำสั่ง Run สังเกตุตัวเลือกส่วนแสดงผลของแต่ละโหนด

ผลที่คาดหวัง โปรแกรมสามารถสามารถทำการเพิ่มโพโทคอลได้โดยมีข้อความแจ้งเตือนการเพิ่ม เมื่อกดตกลง GUI ของโปรแกรมจะทำการเพิ่มตัวเลือก ProtocolSample ที่ตัวเลือกโพโทคอลชั้นที่ 1



รูปที่ 4.26 ข้อความแจ้งเตือนการเพิ่มโพรโทคอล ProtocolSample



รูปที่ 4.27 GUI ที่ทำการลบส่วนแสดงผล ProtocolSample

บทที่ 5
สรุปผลและข้อเสนอแนะ

โครงงานนี้เป็นโครงการต่อยอด โดยใช้ภาษาจาวาในการพัฒนาโปรแกรม และใช้เฉพาะไลบรารี่พื้นฐานที่มากับคอมไพเลอร์ เนื่องจากเป็นภาษาที่ง่ายต่อการเรียนรู้ มีคุณสมบัติ Object Oriented และที่สำคัญคือมีคุณสมบัติ "Write Once, Run Anywhere" ซึ่งทำให้ไลบรารี่ของโครงงานนี้สามารถนำไปใช้ได้ทุกระบบปฏิบัติการ โดยโครงงานได้รับการพัฒนาจากโปรแกรมเดิมในเรื่องแก้ไขความผิดพลาดในส่วนต่างๆ จัดโครงสร้างให้สะดวกต่อการต่อยอด เพิ่มทางเลือกในส่วนแสดงผล สะดวกต่อการเพิ่มลดส่วนแสดงผลและโพรโทคอล และมีโปรแกรมสำหรับช่วยเหลือในการจำลองทอพอโลยี โดยทั้งหมดนี้เพื่อทำให้ง่ายต่อการใช้งานและพัฒนา

ในส่วนของหลักการและทฤษฎีนั้น ได้ศึกษาเกี่ยวกับการส่งข้อมูลระดับเครือข่ายโพรโทคอล เทรด การเพิ่มไลบรารี่ ทอพอโลยี การหาเส้นทางที่สั้นที่สุด โดยได้รับการช่วยเหลือความรู้และข้อมูลจากอาจารย์ที่ปรึกษาโครงงานเป็นสำคัญ กอปรกับการศึกษาเพิ่มเติมจากแหล่งต่างๆ ทั้งหนังสือ และเอกสารบนอินเตอร์เน็ต

วิธีการดำเนินงานเพื่อพัฒนาโปรแกรมนั้น ได้มีการดำเนินการอย่างเป็นขั้นตอนตั้งแต่โดยเริ่มจาก การแก้ไขความผิดพลาดและโครงสร้างของโปรแกรมเดิมจากนั้นพัฒนาโปรแกรมในส่วนต่างๆ อาทิ พัฒนาให้โปรแกรมสะดวกต่อการเพิ่มลดโพรโทคอล โดยได้รับการช่วยเหลือออกแบบโครงสร้างและข้อเสนอแนะในการดำเนินงานต่างๆจากอาจารย์ที่ปรึกษาโครงงานเป็นสำคัญ

หลังจากดำเนินการการพัฒนาโปรแกรมเป็นอันเสร็จสิ้นแล้ว ได้มีการตรวจสอบโครงงานโดยได้ทำการทดลองการใช้งานโปรแกรมเป็นไปตามที่คาดหวังไว้หรือไม่ โดยได้ทดสอบการใช้งานโปรแกรมโดยละเอียดตั้งแต่การเปิดโปรแกรม จนถึงการปิดโปรแกรม และนำผลการทดสอบมาปรับปรุงโปรแกรมให้มีข้อผิดพลาดน้อยลง

โปรแกรมที่พัฒนาขึ้น ได้ถูกนำมาใช้เป็นสื่อการเรียนการสอนในวิชาที่สอนเกี่ยวกับระบบเครือข่ายและการสื่อสารข้อมูล ซึ่งจะช่วยเพิ่มความเข้าใจเกี่ยวกับการทำงานของโพรโทคอล ทั้งยังเพิ่มความเข้าใจเกี่ยวกับการเขียนโปรแกรมด้วยภาษาจาวาอีกด้วย

5.1 สรุปผลการทดลอง

จากการทดสอบโปรแกรมสามารถสรุปการทดลองได้ดังตารางที่ 5.1

ตารางที่ 5.1 ตารางสรุปผลการทดลอง

	ผลการทดลอง
การทำงานของ GUI ของแต่ละโหนด	✓
การทำงานของส่วนแสดงผล	✓
การทำงานของส่วนโพรโทคอล	✓
สะดวกต่อการเพิ่มลดส่วนแสดงผล	✓
สะดวกต่อการเพิ่มลดส่วนโพรโทคอล	✓
การทำงานของ GUI ที่จำลองการเชื่อมต่อของแต่ละโหนด	✓

5.2 ปัญหาและแนวทางแก้ไข

5.2.1 ปัญหาและแนวทางแก้ไขที่ได้รับการแก้ไข

ตารางที่ 5.2 ปัญหาและแนวทางแก้ไขที่ได้รับการแก้ไข

ปัญหา	แนวทางแก้ไข
1. ใช้เวลาในการศึกษาระบบการทำงานของโปรแกรมชุดเดิมนาน	1. รันโปรแกรมเพื่อศึกษาการทำงานประกอบกับขอคำปรึกษาอาจารย์และหาข้อมูลจากคู่มือของโปรแกรมเดิม
2. ใช้เวลาศึกษาการทำงานของโพรโทคอลนานเนื่องจากไม่มีความรู้เกี่ยวกับโพรโทคอล	2. สืบค้นจากหนังสือที่เกี่ยวข้องในส่วนของโพรโทคอล
3. ใช้เวลาในการศึกษา แนวความคิดการทำให้โปรแกรม ที่ใช้งานง่าย ทั้งในส่วนของ GUI และ การทำให้การเพิ่มส่วนต่างๆทำได้ง่าย	3. ขอคำปรึกษาอาจารย์ หาความรู้เพิ่มเติมเกี่ยวกับการเขียนโปรแกรมภาษาจาวา

5.2.2 ปัญหาและแนวทางแก้ไขที่ยังไม่ได้รับการแก้ไข

ตารางที่ 5.3 ปัญหาและแนวทางแก้ไขที่ยังไม่ได้รับการแก้ไข

ปัญหา	แนวทางแก้ไข
1. ปัญหาต่างๆจากตัวโปรแกรมเดิมเช่น การติดต่อข้อมูลผ่านทาง Socket	1. ศึกษาการทำงานของคลาส L0_Socket จนรู้ถึงปัญหาที่เกิดขึ้น

5.3 ข้อเสนอแนะ

จากการศึกษาและทำโครงงาน ผู้จัดทำจึงมีข้อเสนอแนะสำหรับผู้ที่ต้องการนำไปพัฒนาต่อเกี่ยวความรู้พื้นฐานที่ควรมี ปัญหาที่จะเจอและแนวทางแก้ไข และแนวทางการพัฒนาโปรแกรมดังนี้

5.3.1 ความรู้พื้นฐานที่ควรมี

- มีความรู้พื้นฐานการเขียนโปรแกรมด้วยภาษาจาวาเป็นอย่างดี
- มีความรู้พื้นฐานเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุเป็นอย่างดี
- มีความรู้ความเข้าใจการสร้าง GUI โดยใช้ภาษาจาวา
- มีความรู้ความเข้าใจการสร้างเทรดโดยใช้ภาษาจาวา
- มีความรู้ความเข้าใจเกี่ยวกับ หลักการ MVC

5.3.2 ปัญหาที่จะเจอและแนวทางแก้ไข

นอกจากในข้อ 5.2.1 และข้อ 5.2.2 แล้ว ผู้ที่จะพัฒนาอาจจะเจอปัญหาเกี่ยวกับการใช้เทรดจึงควรหาหนังสือเกี่ยวกับการใช้เทรดในภาษาจาวา

5.3.3 แนวทางการพัฒนาโปรแกรม

เนื่องจากโปรแกรมที่จัดทำนี้สะดวกต่อการเพิ่มเติมส่วนต่างๆ แต่การเพิ่มนั้นยังต้องอาศัยการปิดแล้วเปิดโปรแกรมใหม่ ซึ่งหากมีความรู้ของ MVC มากพอ ผู้จัดทำคิดว่าจะสามารถทำการเพิ่มเติมส่วนต่างๆโดยปิดเปิดโปรแกรมใหม่ เพียงแต่นำคลาสใหม่มาไว้ในโฟลเดอร์ของโปรแกรมโดยสามารถเพิ่มได้ในขณะที่ใช้งานโปรแกรมอยู่ แนวทางการพัฒนาคือสร้างตัวโมเดล (Model) โดยใช้เทรด (Thread) ตรวจจับที่โฟลเดอร์ของโปรแกรมตลอดเวลา และให้ GUI เป็นวิว (View) คลาส NewConfigLayer เป็นคอนโทรลเลอร์ (Controller)

หรือหากการตรวจจับที่โฟลเดอร์ตลอดเวลาทำให้โปรแกรมทำงานหนักเกิน อาจจะเพิ่มปุ่มในโปรแกรม เมื่อกดปุ่มดังกล่าวจะทำการตรวจจับที่โฟลเดอร์ของโปรแกรมหนึ่งครั้ง แต่มีข้อเสียคือหากเราจำลองหลายโหนด จะต้องกดทุกโหนด

เอกสารอ้างอิง

[1] lys. (September 3, 2012). **Transmission Control Protocol (TCP).** Retrieved November 15 2012, from http://blackandwhitecomputer.blogspot.com/2012/09/transmission-control-protocol-tcp.html

[2] มัณฑนาลิมากิรักษ์, สราวุธสุขใจ. **โปรแกรมแสดงการทำงานเครือข่าย 4 ชั้นของ OSI.** มหาวิทยาลัยนเรศวร, พิษณุโลก, ปริญญานิพนธ์ 2010

[3] ชัยภัทร จารุชาติ, นทภพ เริงนภาพล. **โปรแกรมจำลองการทำงานสำหรับเน็ตเวิร์คโพรโทคอล** TCP/IP เลเยอร์1 และเลเยอร์2. มหาวิทยาลัยนเรศวร, พิษณุโลก, ปริญญานิพนธ์ 2008

[4] กิตติศักดิ์ ตุนาโป่ง, ณัฐเสฏฐ์ วัชรานุรักษ์. **ไลบรารีสำหรับจำลองเน็ตเวิร์คโพรโทคอลสแตก.** มหาวิทยาลัยนเรศวร, พิษณุโลก, ปริญญานิพนธ์ 2007

[5] ไม่ทราบผู้แต่ง. **รูปแบบของการเชื่อมโยงเครือข่าย หรือโทโปโลยี (LAN Topology). สืบค้นเมื่อ** 15 พฤศจิกายน 2555, จากhttp://www.yupparaj.ac.th/RoomNet2545/activity7/topology.htm

[6] ไม่ทราบผู้แต่ง. (17 สิงหาคม 2553). **Encapsulation and De-enapsulation Process. สืบค้นเมื่อ** 15 มีนาคม 2557, จากhttp://ciscoiseasy.blogspot.com/2010/08/lesson-5-encapsulation-and-de.html

[7] ไม่ทราบผู้แต่ง. (17 มิถุนายน พ.ศ. 2555). โพรโทคอลและอุปกรณ์สื่อสารในระบบเครือข่ายคอมพิวเตอร์.สืบค้นเมื่อ 15 มีนาคม2557, จากhttp://onatcha46203.blogspot.com/2012/06/blog-post_5181.html

[8] ไม่ทราบผู้แต่ง. (16 ตุลาคม พ.ศ. 2555). **OSI Model และ TCP/IP Model. สืบค้นเมื่อ** 15 มีนาคม 2557, จากhttp://genesistk2009.blogspot.com/2012/10/osi-model-tcpip-model.html

ภาคผนวก ก

ภาคผนวก ก

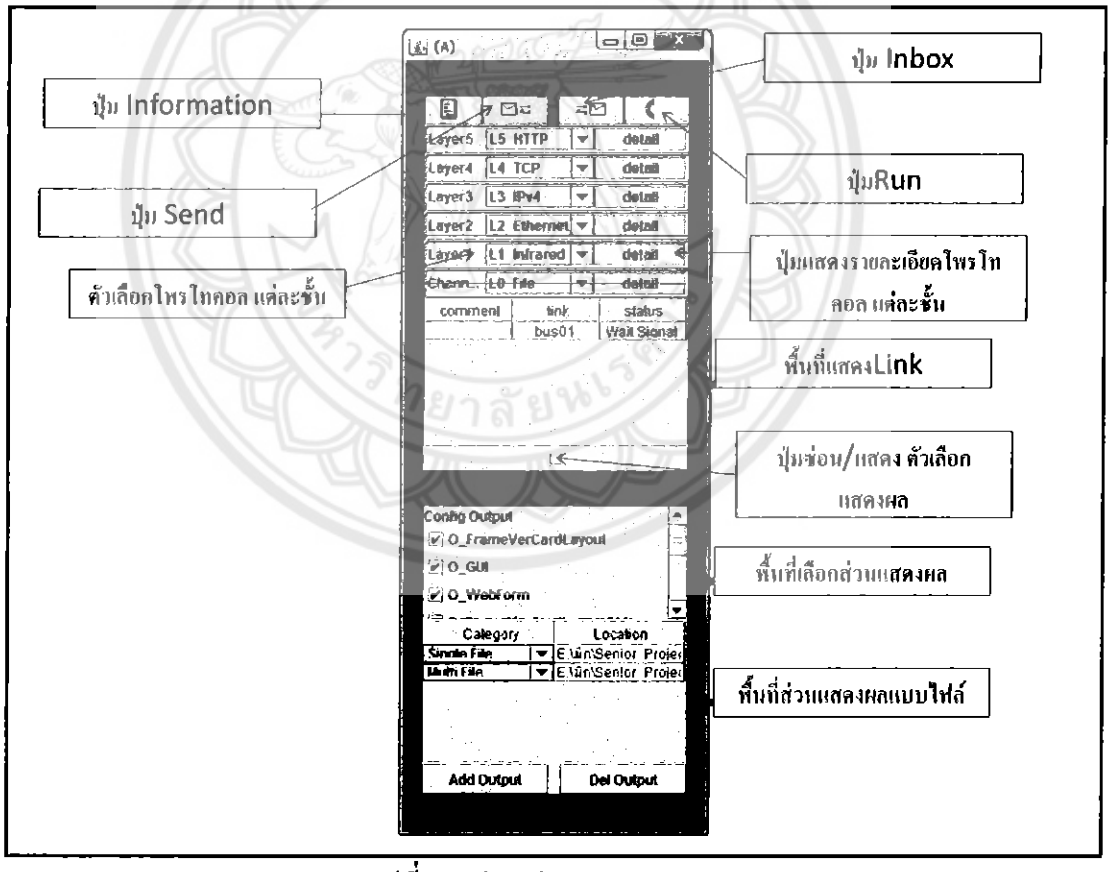
การใช้งานและเพิ่มลดคลาสใหม่ในส่วนแสดงผล หรือ โพรโทคอล

ก1. วิธีการใช้โปรแกรม

การรันโปรแกรมสามารถรันได้สามวิธีคือ

1. การรันจาก Main (เช่น java Main A 127.0.0.10)
2. การรันจาก MainNew (เช่น java MainNew แล้วค่อยใส่ค่า A และ 127.0.0.10)
3. จากการใช้คำสั่ง Run ของ GUI ที่ใช้ออกแบบทอพอโลยี

เมื่อรันโปรแกรมแล้วจะได้ GUI ดังนี้



รูปที่ ก.1 ส่วนประกอบของ GUI

- ปุ่ม Information ใช้แสดงเนื้อหาเกี่ยวกับการใช้โปรแกรม
- ปุ่ม Run ใช้กดเพื่อเริ่มการทำงาน
- ปุ่ม Send ใช้ส่งข้อมูล

- ปุ่ม Inbox แสดงข้อมูลที่ได้รับ

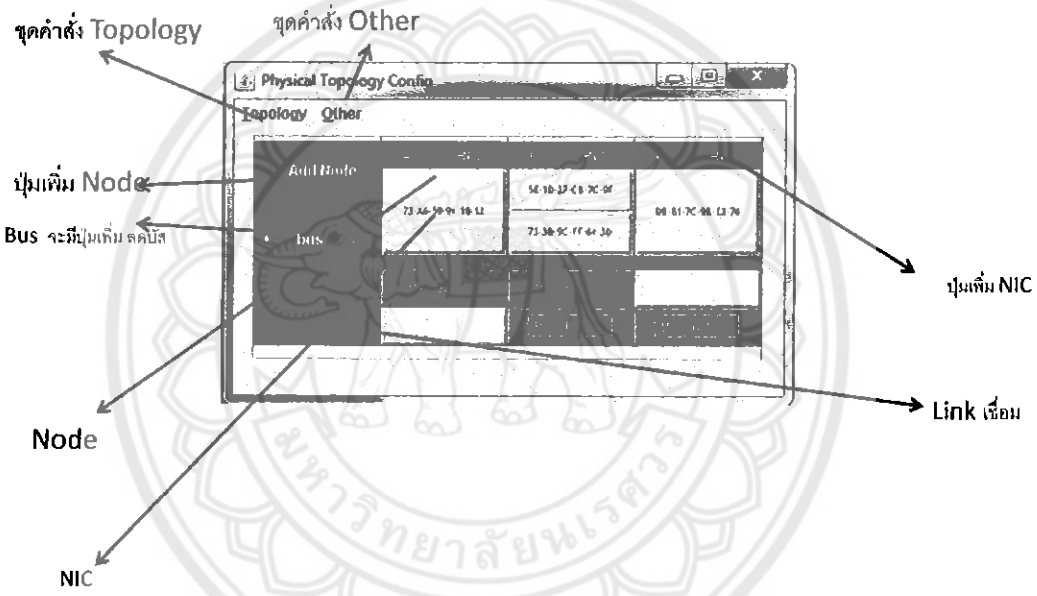
- พื้นที่โพรโทคอลแต่ละชั้น ประกอบด้วยชื่อชั้น ตัวเลือกโพรโทคอลในชั้นนั้นๆ และปุ่มแสดงการรับส่งโพรโทคอลในชั้นนั้นๆ (ปุ่ม + msg)

- พื้นที่แสดง port (ช่องทาง) ของโหนดนั้นๆ

- ปุ่มเพิ่ม, ปุ่มลด port ใช้เพิ่มและลบ port

- พื้นที่เลือกส่วนแสดงผล ใช้เลือกส่วนแสดงผล

ก2. วิธีการใช้โปรแกรมตัวออกแบบทอพอโลยี



รูปที่ ก.2 ส่วนประกอบของ GUI ของตัวออกแบบทอพอโลยี

ชุดคำสั่ง Topology ประกอบด้วยคำสั่งดังนี้

1. คำสั่ง New ใช้ออกแบบทอพอโลยีใหม่

2. คำสั่ง Save ใช้บันทึกทอพอโลยีที่ออกแบบไว้ในที่อยู่และชื่อไฟล์ที่ต้องการ โดยเก็บอยู่ในรูปของไฟล์นามสกุล เอ็กซ์เอ็มแอล

3. คำสั่ง Open ใช้เรียกทอพอโลยีที่ถูกเก็บบันทึกอยู่ในไฟล์เอ็กซ์เอ็มแอล มาแสดงผล

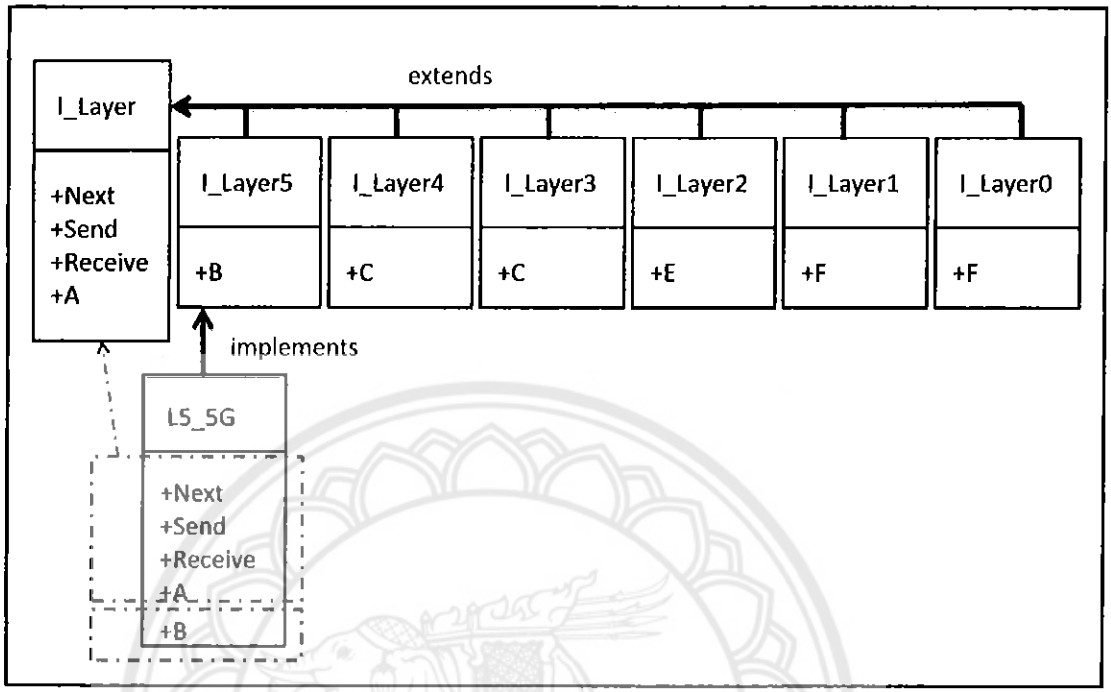
ชุดคำสั่ง Other ประกอบด้วยคำสั่งดังนี้

1. คำสั่ง Build ใช้สร้างทอพอโลยีลงในทอพอโลยีของโปรแกรมหลัก (บันทึกทอพอโลยีที่ออกแบบไว้ลงในไฟล์ Topology.xml ที่อยู่ในตัวโปรแกรม)

2. คำสั่ง Run ใช้ เปิดโหนดและกำหนดค่าตามทอพอโลยีที่ออกแบบไว้

3. คำสั่ง Exit ใช้ออกโปรแกรม

ก3. วิธีการเพิ่มและลดโพรโทคอล

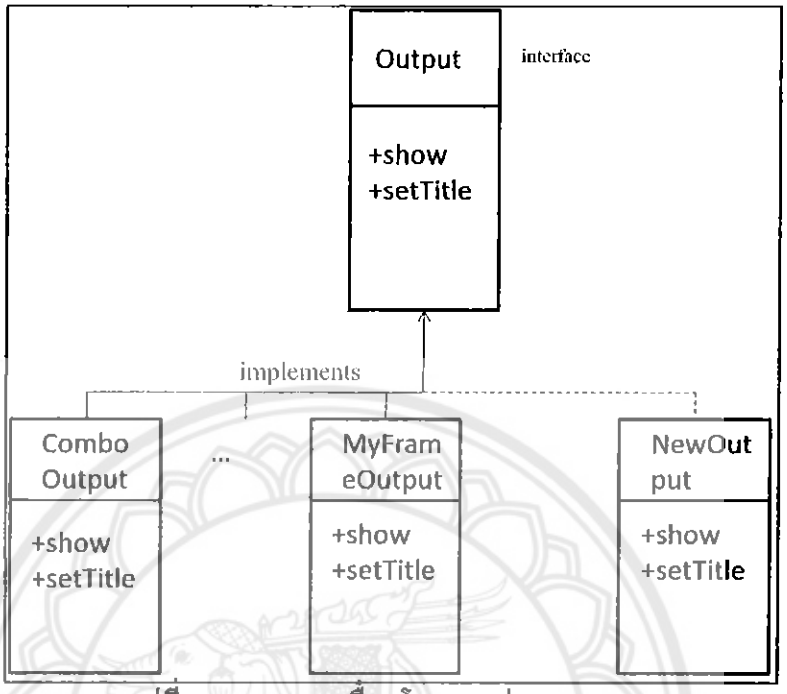


รูปที่ ก.3 แสดงการเชื่อมโยงของ Interface ของโพรโทคอลแต่ละชั้น

การจะเพิ่มโพรโทคอลที่ชั้นใด ก็ให้เขียนโพรโทคอล implements จาก I_Layer ชั้นนั้น เช่น ตัวอย่างต้องการเพิ่ม L5_HTML ก็สร้างคลาสที่ implements จาก I_Layer5 และที่สำคัญ ต้องมีฟังค์ชันอย่างน้อยดังตัวอย่างเพราะได้รับการสืบทอด abstract ฟังค์ชัน เพราะว่า I_Layer, I_Layer5 เป็นเพียง interface เท่านั้น หากทำถูกต้องเพียงแค่นี้ GUI ก็จะสร้าง โพรโทคอลของเรา เป็นตัวเลือกที่อยู่ที่ Combobox ของ Layer นั้นๆ ให้

การจะลดโพรโทคอลให้ทำการย้ายไฟล์ของโพรโทคอลนั้นๆ (ไฟล์นามสกุล .java และ .class ที่เป็นชื่อของโพรโทคอลนั้น) ออกจากตัวโปรแกรม

ก4. วิธีการเพิ่มและลดส่วนแสดงผล



รูปที่ ก.4 แสดงการเชื่อมโยงของ ส่วนแสดงผล

การเพิ่มส่วนแสดงผล ส่วนแสดงผลตัวใหม่ต้อง implements Output และมี ฟังค์ชันตามที่ Output มี เพียงแค่นี้ตัวโปรแกรมก็จะทำการเพิ่มส่วนแสดงผลใหม่ให้กับเรา

การจะลดส่วนแสดงผล ให้ทำการย้ายไฟล์ของส่วนแสดงผลนั้นๆ (ไฟล์นามสกุล .java และ .class ที่เป็นชื่อของส่วนแสดงผลนั้น) ออกจากตัวโปรแกรม

ก5. วิธีการเพิ่มและลดInformation

การเพิ่ม Information ทำได้โดยสร้าง ไฟล์นามสกุลที่รองรับ แล้วนำไฟล์มาใส่ในโฟลเดอร์ Information เท่านี้ ตัว Information ก็จะทำการเพิ่มไฟล์ดังกล่าวให้เอง

การลบ Information ทำได้โดยนำตัวไฟล์เนื้อหาที่ต้องการเอาออก ออกจากโฟลเดอร์ Information เท่านี้ตัว Information ก็จะทำการลบไฟล์ดังกล่าวให้เอง

เบื้องต้นรองรับไฟล์นามสกุล Doc, Docx, Ppt, Pptx หากต้องการแก้ไข แก้ไขที่ไฟล์ Informatiom.java โดยนามสกุลของไฟล์จะอยู่ในตัวแปล exts