โปรแกรมแสดงการทำงานเครือข่าย 4 ชั้นของ OSI

A program for displaying how protocols in 4 layers of OSI work

นางสาวมัณฑนา ลิมาภิรักษ์ รหัส 50362016

นายสราวุธ สุขใจ รหัส 50362580

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 11 มี.ค. 2555
เลขทะเบียน 15733868
เลขเรียกหนังสือ ป5
มหาวิทยาลัยนเรศวร ม336 ป
2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2553

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	โปรแกรมแสดงการทำงานเครือข่าย 4 ชั้นของ OSI		
ผู้ดำเนินโครงงาน	นางสาวมัณฑนา	ลิมาภิรักษ์	รหัส 50362016
	นายสราวุธ	สุขใจ	รหัส 50362580
ที่ปรึกษาโครงงาน	ดร.สุรเดช จิตประไพกุลศาล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2553		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

..... ที่ปรึกษาโครงงาน
(ดร.สุรเดช จิตประไพกุลศาล)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. พนมขวัญ ริยะมงคล)

..... กรรมการ
(อาจารย์เศรษฐา ตั้งค้าวานิช)

ชื่อหัวข้อโครงงาน	โปรแกรมแสดงการทำงานเครือข่าย 4 ชั้นของ OSI		
ผู้ดำเนินโครงงาน	นางสาวมัณฑนา	ลิมาภิรักษ์	รหัส 50362016
	นายสราวุธ	สุขใจ	รหัส 50362580
ที่ปรึกษาโครงงาน	ดร. สุรเดช จิตประไพกุลศาล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2553		

บทคัดย่อ

โครงงานนี้เป็นการเพิ่มเติมส่วนแสดงผลของไลบรารีสำหรับจำลองการทำงานของเครือข่าย การแสดงการทำงานในลักษณะ loop back การส่งต่อข้อมูลผ่านไฟล์ และเพิ่มเติมการแสดงการทำงานของโพรโทคอล UDP, ARP และ RARP ในส่วนของการแสดงผลที่เพิ่มเติมนั้นเดิมเป็นการแสดงผลโดยไม่แยกส่วน โดยในโครงงานนี้เพิ่มการแสดงผลแยกเป็นส่วน ๆ ตามลักษณะของโพรโทคอล ซึ่งในขณะนี้สามารถรองรับการแสดงผลโพรโทคอล TCP, UDP, ARP, RARP และ IPv4 ได้ โครงงานนี้ใช้ภาษา Java ในการพัฒนาโปรแกรมโดยใช้เฉพาะไลบรารีพื้นฐานที่มากับคอมไพเลอร์

Project title	A program for displaying how protocols in 4 layers of OSI work		
Name	Miss Munthana	Limapirak	ID. 50362016
	Mr. Sarawut	Sukjai	ID. 50362580
Project advisor	Dr. Suradet	Jitprapaikulsarn	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2010		

Abstract

This project enhances the library for simulating the TCP / IP Network Protocol Layer 1 and Layer 2 by adding organized and segmented display, showing loop back communication, transmitting data via files, and simulating the protocols UDP, ARP, and RARP. Originally the library supported only single page display with scrolling capability. Our project enhanced this capability by separating the display into segments according to the protocol specification. The supported protocols include TCP, UDP, ARP, RARP, and IPv4. In this project the Java programming language was used as implementation language without additional libraries beside standard libraries bundled with the compiler.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ฉบับนี้สำเร็จลุล่วงมาได้นั้น เนื่องจากความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงงาน ดร.สุรเดช จิตประไพกุลศาล ที่กรุณาสละเวลาให้คำแนะนำในการทำงาน ตลอดถึงการตรวจสอบการทำงานพร้อมทั้งเสนอแนวทางการแก้ไขปัญหาตลอดระยะเวลาการทำโครงงาน พร้อมทั้งให้คำแนะนำที่เป็นประโยชน์ทำให้การทำโครงงานเป็นไปอย่างราบรื่น

ทั้งนี้ต้องขอขอบพระคุณกรรมการทั้งสองท่านอันได้แก่ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล และอาจารย์เศรษฐา ตั้งก้าวานิช อาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ที่เสียสละเวลาอันมีค่าให้ปรึกษาและแนะแนวทางในการแก้ปัญหาต่างๆ

สุดท้ายนี้ผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา และอาจารย์ทุกท่าน ที่คอยสั่งสอนให้ความรู้จนผู้จัดทำสำเร็จการศึกษา และขอขอบคุณเพื่อน ๆ ที่คอยให้กำลังใจ ช่วยให้คำปรึกษาทั้งในเรื่องเรียนเรื่องส่วนตัวจนสำเร็จลุล่วงมาได้ด้วยดี

ขออำนาจคุณพระศรีรัตนตรัย และสิ่งศักดิ์สิทธิ์ทั้งหลายบันดาลให้บิดามารดาและอาจารย์ทุกท่านสุขภาพแข็งแรงและเป็นแรงผลักดันให้คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร มีความก้าวหน้าต่อไป

นางสาวมัณฑนา ลิมาภิรักษ์

นายสราวุธ สุขใจ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่ออังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่ 1 บทนำ

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 รายละเอียดงบประมาณของโครงงาน 3

บทที่ 2 หลักการและทฤษฎี

2.1 Open System Interconnection Basic Reference Model (OSI Model) 4

2.2 TCP/IP Protocol Suites 5

2.3 User Datagram Protocol (UDP) 6

2.3.1 UDP Header 8

2.3.2 UDP Checksum 8

2.3.3 Encapsulation และ Decapsulation 10

2.4 Address Mapping 10

2.5 Address Resolution Protocol (ARP) 11

2.5.1 การทำงานของโพรโทคอล ARP 14

2.5.1 กลไกการทำงานของโพรโทคอล ARP 15

2.6 Reverse Address Resolution Protocol (RARP) 17

สารบัญ (ต่อ)

หน้า

2.7 Model View Controller (MVC) 18

บทที่ 3 วิธีการดำเนินงาน

3.1 คำอธิบายของระบบ(System Description) 21

3.2 การออกแบบโปรแกรม 21

3.2.1 Overview ของโปรแกรม 22

3.2.2 ขั้นตอนการทำงานของโพรโทคอล UDP 23

3.2.3 ขั้นตอนการทำงานของโพรโทคอล ARP 25

3.2.4 ขั้นตอนการทำงานของโพรโทคอล RARP 26

3.2.5 การออกแบบโปรแกรมให้ทำงานตามรูปแบบของ Model View Controller 28

บทที่ 4 ผลการทดลอง

4.1 แผนการทดสอบโปรแกรม 30

4.2 การทดสอบโปรแกรม 31

4.2.1 Unit Test 31

4.2.2 การทดสอบการส่งของโพรโทคอล TCP 32

4.2.3 การทดสอบการรับของโพรโทคอล TCP 33

4.2.4 การทดสอบการส่งของโพรโทคอล UDP 34

4.2.5 การทดสอบการรับของโพรโทคอล UDP 34

4.2.6 การทดสอบการส่งของโพรโทคอล ARP 35

4.2.7 การทดสอบการรับของโพรโทคอล ARP 36

4.2.8 การทดสอบการส่งของโพรโทคอล RARP 37

4.2.9 การทดสอบการรับของโพรโทคอล RARP 38

4.2.10 การทดสอบการส่งของโพรโทคอล IPv4 39

4.2.11 การทดสอบการรับของโพรโทคอล IPv4 40

4.2.12 การแสดงผลของโปรแกรมแบบ GUI 41

4.3 สรุปการทดสอบโปรแกรม 43

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลการดำเนินงานและแนวทางการพัฒนา

5.1 ผลการทดลอง 44

5.2 ปัญหาและอุปสรรค 45

5.2.1 ปัญหาด้านการอิมพลีเมนท์โพรโทคอล 45

5.2.2 ปัญหาด้านการแสดงผลโดยแยกเป็นส่วน ๆ ตามลักษณะโพรโทคอล 45

5.3 ข้อเสนอแนะ 46

5.4 สรุปผลการดำเนินงาน 46

เอกสารอ้างอิง 47

ภาคผนวก 48

ประวัติผู้เขียนโครงงาน 51

สารบัญตาราง

ตารางที่ หน้า

2.1 รายละเอียดของ User Datagram Format 8

2.2 รายละเอียดของ Address Resolution Protocol Format 12

3.1 Mapping UML and PSP Views 21

4.1 ตารางแผนการทดลอง 30

4.2 ตารางการทดสอบโปรแกรม Unit Test 31

4.3 ตารางสรุปการทดสอบ 43

5.1 ตารางสรุปผลการทดลอง 44

5.2 ปัญหา อุปสรรค และแนวทางแก้ไข ด้านการอิมพลีเมนท์โพรโทคอล 45

5.3 ปัญหา อุปสรรค และแนวทางแก้ไข ด้านการแสดงผลโดยแยกเป็นส่วน ๆ 45

สารบัญรูป

รูปที่ หน้า

2.1 OSI Model 4

2.2 TCP/IP and OSI Model 5

2.3 ตำแหน่งของ UDP ใน TCP/IP Protocol Suite 7

2.4 User Datagram Format 8

2.5 Pseudo Header สำหรับการคำนวณ Checksum 9

2.6 ตัวอย่างของการคำนวณ Checksum 9

2.7 การ Encapsulation และการ Decapsulation 10

2.8 ARP และ RARP Frame Format 12

2.9 การส่ง Packet ของโพรโทคอล ARP 14

2.10 การตอบกลับของโพรโทคอล ARP 15

2.11 ARP Request จะถูกส่งไปยังเครื่องคอมพิวเตอร์ทุกเครื่องในเครือข่าย 16

2.12 ARP Reply จะถูกตอบกลับมา เพื่อบอก MAC Address ของตนเอง 16

2.13 RARP Operation 17

2.14 RARP Packet 18

2.15 โครงสร้างของ MVC 19

2.16 ตัวอย่างการเขียนโปรแกรมโครงสร้าง MVC 20

3.1 Use case Diagram ของระบบ 22

3.2 Activity Diagram ของระบบ 23

3.3 Class Diagram ของโพรโทคอล UDP 24

3.4 Sequence Diagram ของโพรโทคอล UDP 24

3.5 Class Diagram ของโพรโทคอล ARP 25

3.6 Sequence Diagram ของโพรโทคอล ARP 26

3.7 Class Diagram ของโพรโทคอล RARP 27

3.8 Diagram ของโพรโทคอล RARP 27

3.9 Class Diagram แสดงโครงสร้างของ MVC ของภาษา Java 28

3.10 Class Diagram แสดงการอิมพลีเมนท์ MVC 29

4.1 รายละเอียดของโพรโทคอล TCP ฝั่ง Sender 32

4.2 รายละเอียดของ Acknowledge ของโพรโทคอล TCP ฝั่ง Sender 33

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.3 รายละเอียดของ Acknowledge ของโพรโทคอล TCP ฝั่ง Receiver 33
4.4 รายละเอียดของโพรโทคอล TCP ฝั่ง Receiver 34
4.5 รายละเอียดของโพรโทคอล UDP ฝั่ง Sender 34
4.6 รายละเอียดของโพรโทคอล UDP ฝั่ง Receiver 35
4.7 รายละเอียดของ ARP Request Packet 35
4.8 รายละเอียดของ ARP Reply Packet 36
4.9 รายละเอียดของ RARP Request Packet 36
4.10 รายละเอียดของ RARP Reply Packet 37
4.11 รายละเอียดของโพรโทคอล IPv4 ฝั่ง Sender 38
4.12 รายละเอียดของโพรโทคอล IPv4 ฝั่ง Receiver 39
4.13 หน้าโปรแกรมส่วนของ Configuration 40
4.14 หน้าโปรแกรม Configuration แบบ GUI ของฝั่ง TCP Sender 41
4.15 หน้าโปรแกรม Configuration แบบ GUI ของฝั่ง TCP Receiver 42
ก.1 แสดงหน้าเริ่มต้นของโปรแกรม 48
ก.2 แสดงส่วนประกอบต่างๆ ของโปรแกรม 49

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ไลบรารีสำหรับจำลองการทำงานของระบบเครือข่ายที่มีอยู่เดิมนั้น มีส่วนของการแสดงผลที่ซับซ้อนยากแก่การเข้าใจ ซึ่งทำให้ไม่สามารถเห็นลำดับการทำงานของโพรโทคอลสแตกได้อย่างชัดเจน จึงมีความคิดที่จะสร้างส่วนของการแสดงผลที่ทำให้ผู้ใช้สามารถเห็นลำดับการทำงานได้อย่างชัดเจนยิ่งขึ้น

นอกจากนี้ยังมีความคิดที่จะทำการอิมพลีเมนท์ที่ระดับชั้น 3 และ 4 โดยได้ทำการเลือกโพรโทคอลที่ระดับชั้น 3 คือ ARP, RARP และที่ระดับชั้น 4 คือ UDP ซึ่งจะช่วยให้การจำลองระบบเครือข่ายมีความสมบูรณ์และเพิ่มความเข้าใจในระบบเครือข่ายให้มากยิ่งขึ้น

1.2 วัตถุประสงค์โครงงาน

1. เพื่อสร้างซอฟต์แวร์และไลบรารี แสดงการทำงานของเครือข่ายโพรโทคอลสแตกสำหรับโพรโตคอลในระดับชั้น 3 และ 4
2. เพื่อให้ผู้ใช้สามารถนำไลบรารีไปพัฒนาเพิ่มเติมได้ในอนาคต

1.3 ขอบเขตของโครงงาน

1. สามารถจำลองการแสดงผลการทำงานของโพรโทคอลระดับชั้น 3 โดยใช้โพรโทคอล ARP และ RARP
2. สามารถจำลองการแสดงผลการทำงานของโพรโทคอลระดับชั้น 4 โดยใช้โพรโทคอล UDP
3. สามารถเชื่อมต่อโพรโทคอลอื่นๆ กับระดับชั้นเดิมที่สร้างมาก่อนหน้านี้ได้
4. สามารถแสดงผลการทำงานของโพรโทคอลในระดับชั้น 3 และ 4 โดยผ่านการติดต่อกับผู้ใช้แบบกราฟิก

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาความรู้เกี่ยวกับโพรโทคอลในระดับชั้น 3 (ARP, RARP) และ 4 (UDP)
2. ศึกษาการทำงานของโปรแกรมที่สร้างมาก่อนหน้านี้
3. สร้างไลบรารีให้ได้ตามวัตถุประสงค์
4. ศึกษาการทำงานของ Model–View–Controller เพื่อนำมาใช้กับโปรแกรมภาษา Java
5. สร้างส่วนแสดงผลการทำงานของเครือข่ายโพรโทคอลระดับชั้น 3 และ 4
6. ทดสอบความถูกต้องของไลบรารี
7. จัดทำเอกสารและคู่มือการใช้งาน

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 โพรโทคอล										
3. ศึกษาการทำงานของโปรแกรมที่สร้างมาก่อนหน้านี้										
4. ศึกษาการทำงานของการทำงานโพรโทคอล UDP, ARP และ RARP										
5. สร้างไลบรารีให้ได้ตามวัตถุประสงค์										
6. ทดสอบความถูกต้องของไลบรารี										
7. จัดทำเอกสารและคู่มือการใช้										

1.6 ผลที่คาดว่าจะได้รับ

1. เพื่อใช้ในการจำลองการทำงานของเครือข่ายโพรโทคอลสแตกได้ง่ายขึ้น
2. ได้ตัวอย่างการทำงานของการรับส่งข้อมูลผ่านโพรโทคอลระดับชั้น 3
3. ได้ตัวอย่างการทำงานของการรับส่งข้อมูลผ่านโพรโทคอลระดับชั้น 4
4. แสดงให้เห็นถึงลำดับการทำงานของเครือข่ายโพรโทคอลสแตก
5. เพื่อให้ผู้ใช้สามารถใช้ซอฟต์แวร์ประกอบการเรียนรู้ ระบบการทำงานของเครือข่ายโพรโทคอลสแตกในระดับชั้น 3 และ 4

1.7 รายละเอียดงบประมาณของโครงงาน

1. ค่าเอกสารประกอบการศึกษาค้นคว้า	500	บาท
2. ค่าถ่ายสำเนาเอกสาร	500	บาท
3. ค่าจัดทำรูปเล่มปริญญานิพนธ์	1,000	บาท
รวมเป็นเงินทั้งสิ้น	2,000	บาท (สองพันบาทถ้วน)

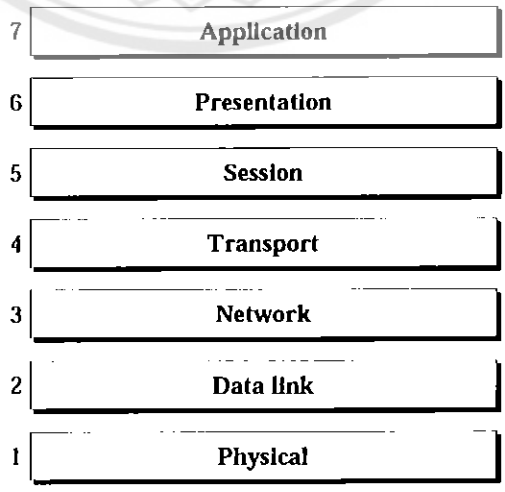
หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องในการพัฒนาโปรแกรมเพื่อแสดงผลของการทำงานเครือข่าย 4 ชั้นของ OSI โดยจะเน้นในส่วนของระดับชั้น Network และระดับชั้น Transport เป็นหลัก โดยได้ทำการเลือกโพรโทคอล ARP และโพรโทคอล RARP ในระดับชั้น Network และได้เลือกโพรโทคอล UDP ในระดับชั้น Transport เพื่อทำการจำลองและแสดงผลการทำงานของโปรแกรม รวมไปถึงการเขียนโปรแกรมภาษา Java โดยใช้ Interface

2.1 Open System Interconnection Basic Reference Model (OSI Model) [1, 2]

OSI Model หรือ OSI Reference Model เป็นมาตรฐานการอธิบายการติดต่อสื่อสารและโพรโทคอลของระบบคอมพิวเตอร์โดยมีวัตถุประสงค์เพื่อให้ผู้ผลิตฮาร์ดแวร์และซอฟต์แวร์ใด ๆ ใช้เป็นโครงสร้างอ้างอิงในการสร้างอุปกรณ์ให้สามารถทำงานร่วมกันได้อย่างดีบนระบบเครือข่าย โดยถูกพัฒนาขึ้นในปี 1984 โดยองค์กรที่ชื่อว่า International Organization for Standardization (ISO) และนอกจากนี้ยังเป็นสถาปัตยกรรมโมเดลหลักที่ใช้อ้างอิงในการสื่อสารระหว่าง computer โดยมีการจัดแบ่งระดับชั้นออกเป็น 7 ระดับชั้น แต่ละระดับชั้นจะมีการโต้ตอบหรือรับส่งข้อมูลกับระดับชั้นที่อยู่ข้างเคียงเท่านั้น โดยระดับชั้นที่อยู่ในระดับล่างจะกำหนดลักษณะของ interface เพื่อให้บริการกับระดับชั้นที่อยู่เหนือขึ้นไปตามลำดับขั้น เริ่มตั้งแต่ระดับล่างสุดซึ่งเป็นการจัดการลักษณะทางกายภาพของฮาร์ดแวร์และการส่งข้อมูลในระดับบิตจนไปสิ้นสุดที่ระดับ application ในระดับบนสุด



รูปที่ 2.1 OSI Model

ที่มา: [2]

จากรูปที่ 2.1 แสดงระดับชั้นทั้ง 7 ระดับชั้นของ OSI Model ซึ่งแต่ละระดับชั้นมีรายละเอียดดังต่อไปนี้

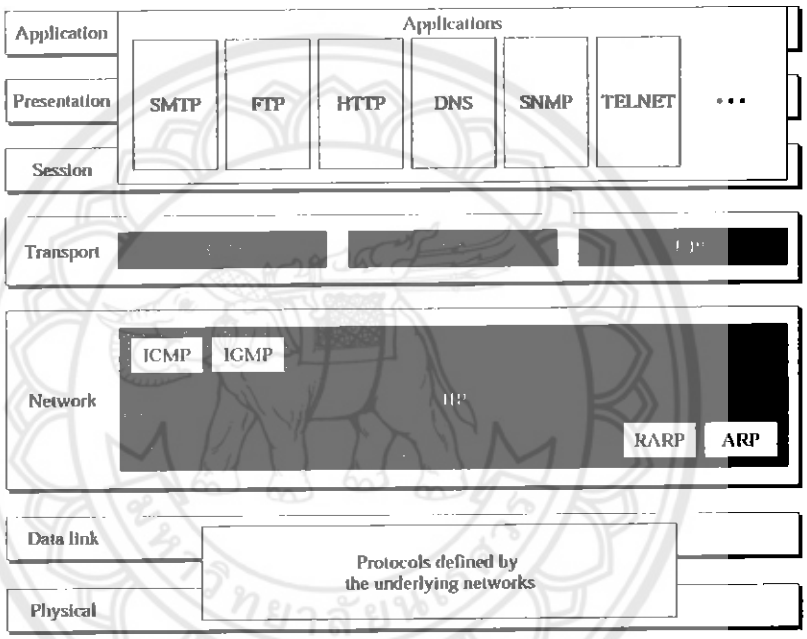
- **Application Layer:** เป็นระดับชั้นที่อยู่ในระดับบนสุดที่มีการทำงานใกล้ชิดกับผู้ใช้ (User) การทำงานในระดับชั้นนี้จะเกี่ยวข้องกับโพรโทคอลต่าง ๆ มากมาย ซึ่งจะมีการใช้งานที่แตกต่างกันและมีการบริการทางด้านโปรแกรมประยุกต์ต่าง ๆ ได้แก่ email, file transfer, remote job entry และ directory services นอกจากนี้ยังมีการจัดเตรียมฟังก์ชันในการเข้าถึงไฟล์และเครื่องพิมพ์ ซึ่งเป็นการแบ่งปันการใช้ทรัพยากรบนระบบเครือข่าย
- **Presentation Layer:** เป็นระดับชั้นที่ทำหน้าที่ในการแปลงรหัสข้อมูลที่ส่งระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องให้เป็นอักขระแบบเดียวกัน เครื่องคอมพิวเตอร์ส่วนใหญ่จะใช้รหัส ASCII (American Standard Code for Information Interchange) แต่ในบางกรณีมีการใช้รหัสแบบ EBCDIC (Extended Binary Coded Decimal Interchange Code) ดังนั้นระดับชั้นนี้จะทำหน้าที่แปลงรหัสเหล่านี้ให้เครื่องคอมพิวเตอร์เข้าใจได้ตรงกัน
- **Session Layer:** เป็นระดับชั้นที่มีการให้บริการสำหรับการใช้งานเครื่องที่อยู่ห่างไกลออกไป (remote login) การถ่ายโอนไฟล์ระหว่างเครื่อง โดยจะมีการจัดตั้งการสื่อสารระหว่าง 2 ฝ่าย เรียกว่า Application Entities หรือ AE โดยระดับชั้นนี้จะมีหน้าที่จัดการให้การสื่อสารเป็นไปอย่างราบรื่น
- **Transport Layer:** เป็นระดับชั้นที่ทำหน้าที่ในการตรวจสอบและแก้ไขความผิดพลาดที่เกิดขึ้นกับข้อมูล คอยแยกแยะและจัดระเบียบของ packet ข้อมูลในจัดเรียงอย่างถูกต้องเหมาะสม
- **Network Layer:** เป็นระดับชั้นที่ทำหน้าที่เกี่ยวข้องกับการหาเส้นทาง (routing) ในการส่ง packet จากต้นทางไปยังปลายทาง โดยจะมีการสลับช่องทางในการส่งข้อมูลที่เรียกว่า packet switching โดยจะมีการสร้างวงจรเสมือน (virtual circuit) ซึ่งคล้ายกับว่ามีเส้นทางเชื่อมโยงกันระหว่างคอมพิวเตอร์ 2 เครื่อง ให้ติดต่อสื่อสารกันได้โดยตรง
- **Data Link Layer:** เป็นระดับชั้นที่จะทำหน้าที่ในการจัดเตรียมข้อมูลที่จะทำการส่งผ่านไปยังระดับชั้นถัดไป
- **Physical Layer:** เป็นระดับชั้นที่ใช้ในการติดต่อสื่อสาร

2.2 TCP/IP Protocol Suite [2]

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโพรโทคอลที่ใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางไปยังปลายทางได้ และสามารถหาเส้นทางในการที่จะส่งข้อมูลได้โดยอัตโนมัติ โดยชุดโพรโทคอล

TCP/IP นี้ได้ถูกพัฒนาขึ้นมาตั้งแต่ปี 1960 โดยถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET และได้มีการขยายการเชื่อมต่อไปทั่วโลกกลายเป็นเครือข่ายอินเทอร์เน็ต

TCP/IP มีหน้าที่ในการติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน มีความคล่องตัวในการสื่อสารข้อมูลได้หลายชนิด นอกจากนี้ยังมีความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับมีการติดต่อสื่อสารกัน แต่สื่อกลางที่เป็นผู้ช่วยในการรับส่งข้อมูลได้รับความเสียหาย การสื่อสารนี้จะต้องสามารถจัดหาเส้นทางอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ



รูปที่ 2.2 TCP/IP and OSI Model

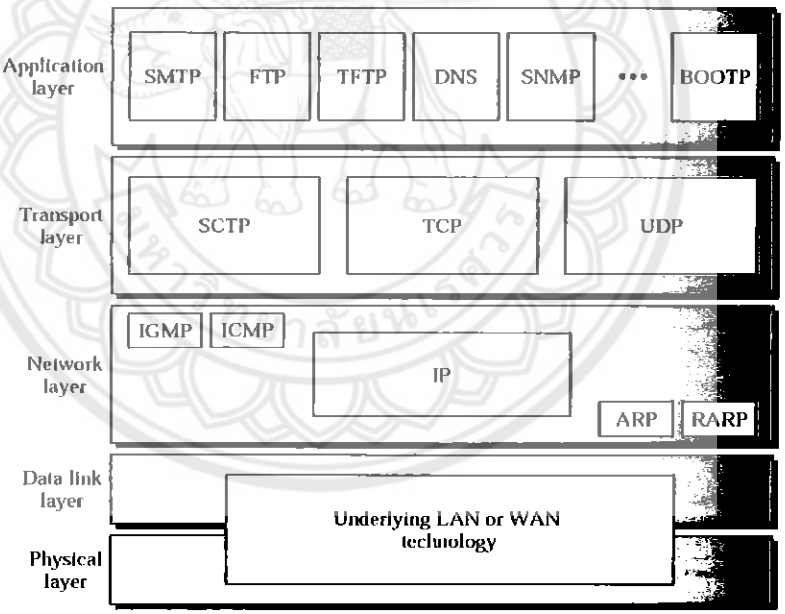
ที่มา: [2]

2.3 User Datagram Protocol (UDP) [1]

User Datagram Protocol (UDP) เป็นวิธีการสื่อสารหรือโพรโทคอลที่จำกัดจำนวนการบริการ เมื่อมีการแลกเปลี่ยนข่าวสารข้อมูลระหว่างคอมพิวเตอร์ในเครือข่ายที่ใช้ Internet Protocol (IP) โดย UDP เป็นโพรโทคอลตัวหนึ่งของ Transmission Control Protocol (TCP) ซึ่งเป็นโพรโทคอลในระดับชั้นที่ 4 ซึ่งเป็นระดับชั้น Transpor และใช้ร่วมกับโพรโทคอล IP ซึ่งในบางครั้งเรียกว่า UDP/IP หรืออาจกล่าวได้ว่า UDP เหมือนกับ TCP ในการใช้ IP สำหรับการดึงหน่วยข้อมูลหรือที่เรียกว่า Datagram จากคอมพิวเตอร์เครื่องหนึ่งไปยังอีกเครื่องหนึ่ง

UDP เป็นการรับส่งข้อมูลที่ไม่มีการยืนยันการรับส่งข้อมูลนั่นคือ ผู้ส่งไม่สามารถรับรู้ได้ว่าข้อมูลที่ส่งไปนั้นถึงผู้รับแล้วหรือไม่ แต่ UDP นั้นสามารถที่จะรับส่งข้อมูลได้เร็วกว่า TCP และจะไม่มีการสร้างการเชื่อมต่อขึ้นมาทำให้มีข้อมูลที่วิ่งในเครือข่ายน้อยลง โพรโทคอล UDP เป็นการสื่อสารแบบ Connectionless คือ ข้อมูลจะถูกแบ่งออกเป็นชิ้นๆ ตามที่อยู่ปลายทาง แล้วมีการส่งผ่านตัวกลางไปถึงยังปลายทาง อาจจะใช้เส้นทางคนละเส้นทางกันก็ได้ รวมไปถึงข้อมูลแต่ละชิ้นนั้นอาจจะมีการส่งถึงผู้รับก่อนหรือหลังแตกต่างกันไปทำให้การเริ่มต้นการส่งได้รวดเร็วโดยไม่ต้องเสียเวลาในการสร้างการเชื่อมต่อ ดังนั้น UDP จึงเหมาะสำหรับการรับส่งข้อมูลแบบ Real time ซึ่งข้อมูลที่เกิดการสูญหายหรือ Delay จะถูกละความสนใจไป

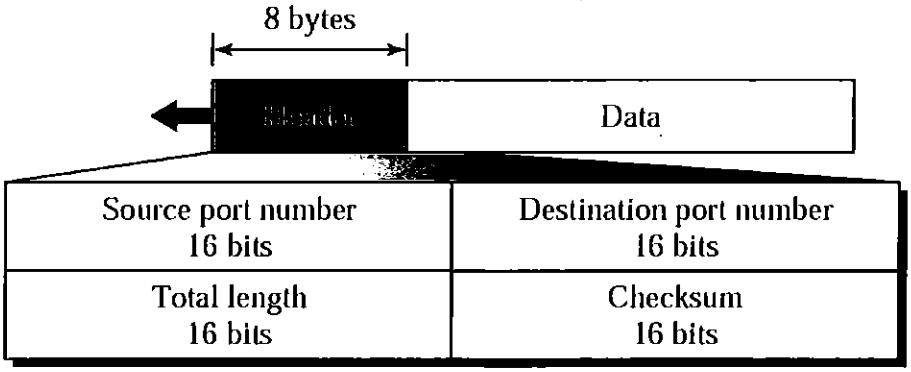
UDP เป็นโพรโทคอลที่อยู่ในระดับชั้น Transport เมื่อเทียบกับแบบจำลอง OSI โดยการส่งข้อมูลของ UDP จะเป็นการส่งครั้งละ 1 ชุดข้อมูล ซึ่ง UDP จะไม่มีความสัมพันธ์กันระหว่าง datagram และจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล ตำแหน่งของโพรโทคอล UDP ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ตำแหน่งของ UDP ใน TCP/IP Protocol Suite

ที่มา: [2]

2.3.1 UDP Header [1]



รูปที่ 2.4 User Datagram Format

ที่มา: [2]

จากรูปที่ 2.4 แสดงองค์ประกอบต่าง ๆ ของ UDP Header ซึ่งรายละเอียดต่าง ๆ นั้นดังแสดงในตารางที่ 2.1

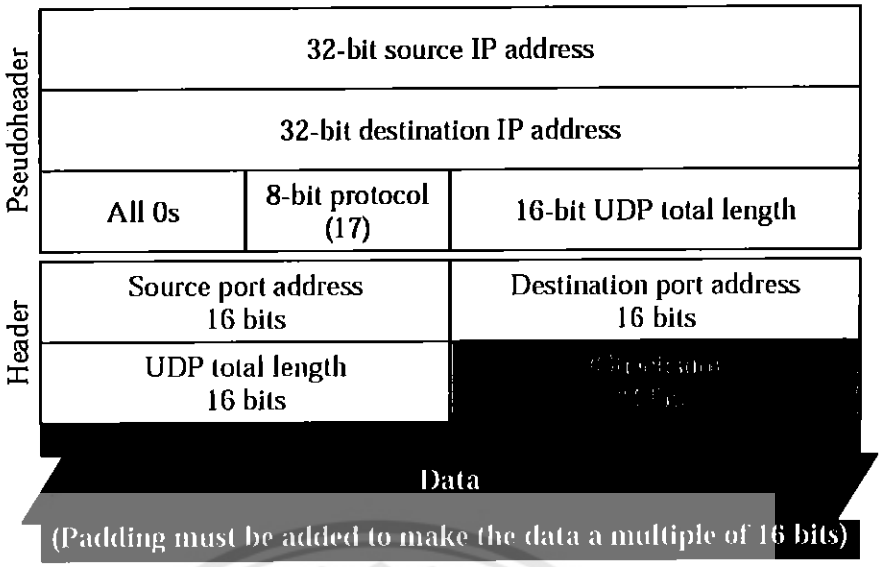
ตารางที่ 2.1 รายละเอียดของ User Datagram Format

ตำแหน่ง	ชื่อ	คำอธิบาย
บิต 0 – 15	Source port number	หมายเลข port ต้นทางที่ส่ง datagram นี้ มีความยาว 16 บิต
บิต 16 – 31	Destination port number	หมายเลข port ปลายทางที่จะเป็นผู้รับ datagram มีความยาว 16 บิต
บิต 32 – 47	UDP length	ความยาวของ datagram ทั้งส่วน Header และ data
บิต 48 - 63	Checksum	เป็นตัวตรวจสอบความถูกต้องของ UDP Datagram

ที่มา: [2]

2.3.2 UDP Checksum [1]

Checksum เป็นเลขขนาด 16 บิตซึ่งถูกคำนวณด้วยวิธี One's complement โดยนำ Pseudo Header และข้อมูลทั้งหมดใน UDP Datagram มาคำนวณ Pseudo Header เป็นข้อมูลที่อยู่ในส่วนของ IP Header ดังแสดงในรูปที่ 2.5

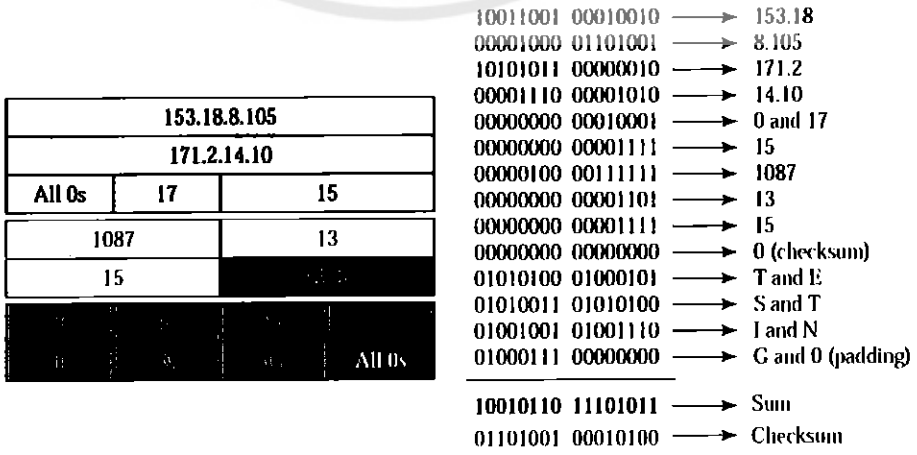


รูปที่ 2.5 Pseudo Header สำหรับการคำนวณ Checksum

ที่มา: [1]

ส่วนของ UDP Pseudo Header จะประกอบไปด้วย Source IP Address ซึ่งมีความยาว 32 บิต ส่วนของ Destination IP Address ซึ่งมีความยาว 32 บิต ส่วนที่ไม่ได้ใช้งานจะถูกปรับเป็น 0 นั่นคือ ส่วนของ All 0s ส่วนของโพรโทคอลซึ่งมีความยาว 8 บิต และส่วนของความยาวของ UDP ซึ่งมีขนาด 16 บิต โดยเมื่อ UDP มีการรับส่งข้อมูลออกไปแล้ว โพรโทคอล UDP จะทราบถึงค่าเหล่านี้โดยอัตโนมัติ

ถ้าหากค่า Checksum ที่คำนวณออกมาเป็น 0 ค่า Checksum จะถูกกำหนดให้เป็น 1 ทั้งหมดแทน (ซึ่งมีค่าเท่ากันกับในระบบ 1's complement) ทั้งนี้เพราะในบางแอพพลิเคชั่นที่ไม่ต้องการการตรวจสอบค่า Checksum ในระดับ UDP จะกำหนดค่านี้ให้เป็น 0 (Disable Checksum)

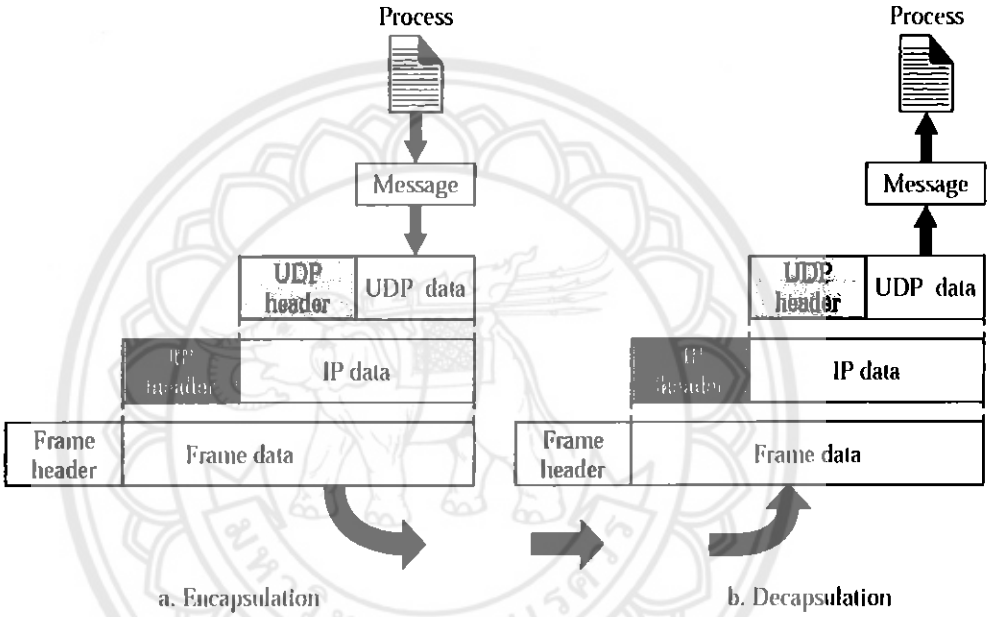


รูปที่ 2.6 ตัวอย่างของการคำนวณ Checksum

ที่มา: [2]

2.3.3 Encapsulation และ Decapsulation [2]

การส่งข้อมูลผ่านแต่ละระดับชั้นนั้น ในแต่ละระดับชั้นจะทำการประกอบข้อมูลที่ได้รับมากับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูล เรียกว่า Header ซึ่งภายใน Header จะบรรจุข้อมูลที่สำคัญของโพรโทคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูลมาก็จะเกิดกระบวนการทำงานย้อนกลับ คือโพรโทคอลเดียวกันในทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลผลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการทำงานย้อนกลับนี้ เรียกว่า Decapsulation



รูปที่ 2.7 การ Encapsulation และการ Decapsulation

ที่มา: [2]

2.4 Address Mapping

ในเครือข่ายระดับชั้น 3 นี้จะใช้ Logical Address หรือ IP Address[1] ในการกำหนดที่อยู่เพื่อรับส่งข้อมูล แต่เมื่อข้อมูลถูกส่งไปยังระดับชั้น Data Link แล้วจำเป็นต้องใช้ Physical Address ดังนั้นจึงต้องมีการค้นหา Physical Address หรือ MAC Address ของเครื่องลูกในเครือข่ายที่ต้องการส่งจาก Logical Address หรือ IP Address โดยวิธีการ Address Mapping นั่นคือก่อนที่จะมีการส่งข้อมูลนั้น จะมีการส่ง Packet ออกไปแบบ Broadcast ถ้า IP Address ตรงกับเครื่องลูกเครื่องใดใน

---

[1] IP Address คือ หมายเลขประจำเครื่องคอมพิวเตอร์ ซึ่งประกอบด้วยตัวเลข 4 ชุด มีเครื่องหมายจุดขั้นระหว่างชุด เช่น 192.168.100.1 หรือ 172.16.10.1 เป็นต้น มาตรฐานของ IP Address ปัจจุบันเป็นมาตรฐานเวอร์ชัน 4 หรือเรียกว่า IPv4 ซึ่งกำหนดให้ IP Address มีทั้งหมด 32 บิตหรือ 4 ไบต์

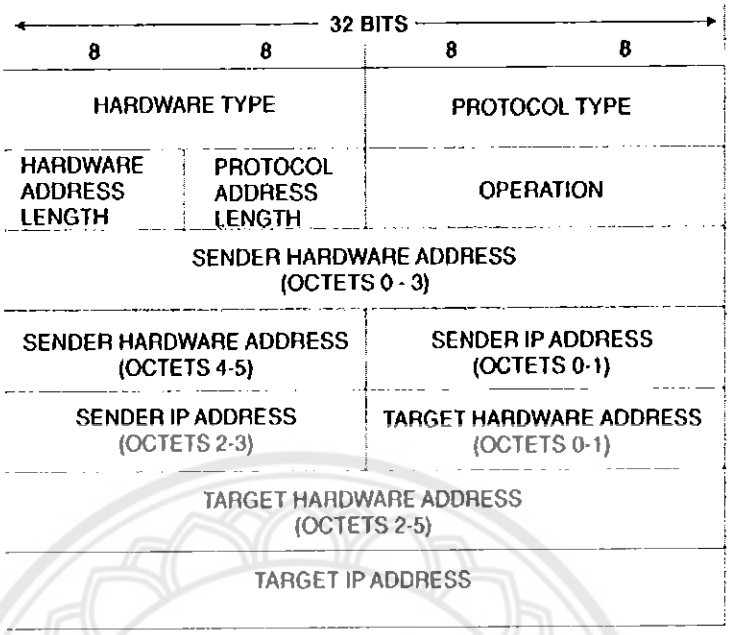
เครือข่าย เครื่องลูกจะมีการส่ง Packet ตอบกลับมาเป็น MAC Address ซึ่งจะใช้โพรโทคอล ARP ในการทำ Address Mapping

2.5 Address Resolution Protocol (ARP) [2]

Address Resolution Protocol (ARP) เป็นโพรโทคอลที่ทำหน้าที่เป็นสื่อกลางในการสื่อสาร ซึ่งมีหน้าที่ค้นหาที่อยู่และจับคู่ระหว่าง IP Address ซึ่งเป็น Logical Address กับ Hardware Address ซึ่งเป็น Physical Address ทั้งนี้เนื่องจากระบบของการรับส่งข้อมูลในระบบ IP นั้นเป็นระบบที่ไม่ขึ้นกับฮาร์ดแวร์ใด ๆ นั่นหมายความว่า ระบบ IP ไม่มีความสามารถในการเรียกใช้ฮาร์ดแวร์ต่าง ๆ ในการส่งข้อมูลด้วยตนเอง ทำให้เมื่อระบบ IP มีความต้องการที่จะส่งข้อมูลจะต้องมีการร้องขอการบริการจากระบบในระดับชั้น Data Link หากเนื่องจากระดับชั้น Data Link ไม่รู้จักที่อยู่ในระบบ IP ดังนั้นระบบ IP จึงต้องทำการค้นหาที่อยู่ที่ระดับชั้น Data Link รู้จัก ซึ่งก็คือ Hardware Address เพื่อที่จะทำการสร้างเฟรมข้อมูลในระดับชั้น Data Link ได้ ซึ่งกระบวนการนี้จะกระทำโดยโพรโทคอล ARP การทำงานของโพรโทคอล ARP คือ เมื่อ Packet นำเข้าที่เครื่อง Host ในระบบเครือข่ายมาถึงเกตเวย์ของคอมพิวเตอร์ที่เกตเวย์จะเรียกโพรโทคอล ARP เพื่อทำการหาเครื่องคอมพิวเตอร์ที่เป็น Host หรือ MAC Address ที่ตรงกับ IP Address โพรโทคอล ARP จะทำการค้นหาใน ARP Cache เมื่อพบเครื่องคอมพิวเตอร์ที่ต้องการแล้วจะทำการแปลง Packet ให้มีความยาวและรูปแบบที่ถูกต้อง เพื่อส่งต่อไปยังเครื่องคอมพิวเตอร์ที่ระบุไว้ แต่ถ้าหากว่าไม่พบโพรโทคอล ARP จะกระจาย Packet ในรูปแบบพิเศษไปยังเครื่องคอมพิวเตอร์ทุกเครื่องในระบบ และถ้าเครื่องคอมพิวเตอร์เครื่องใดเครื่องหนึ่งทราบว่ามี IP Address ตรงกันก็จะทำการตอบกลับมาที่โพรโทคอล ARP โพรโทคอล ARP จะทำการปรับปรุง ARP Cache และทำการส่ง Packet ไปยัง MAC Address หรือเครื่องคอมพิวเตอร์ที่ตอบกลับมา

โพรโทคอล ARP ได้ถูกกำหนดไว้เป็นมาตรฐานภายใต้ RFC 826 โดยการทำงานของโพรโทคอล ARP จะมีรูปแบบการทำงานในแบบ Broadcast ดังนั้นเครือข่ายที่สามารถใช้งานกับโพรโทคอล ARP ได้จึงต้องเป็นเครือข่ายที่มีการทำงานในแบบ Broadcast ซึ่งระบบ LAN ส่วนใหญ่แล้วจะมีการทำงานเป็นแบบ Broadcast จึงทำให้สามารถทำงานร่วมกับโพรโทคอล ARP ได้เป็นอย่างดี

นอกเหนือไปจากนี้แล้วยังมีอีกโพรโทคอลหนึ่งที่ถือว่าเป็นโพรโทคอลคู่แฝดของโพรโทคอล ARP โดยจะมีการทำงานที่ตรงข้ามกับโพรโทคอล ARP ดังนั้นจึงมีชื่อเรียกว่า RARP (Reverse Address Resolution Protocol) โดย RARP ได้ถูกกำหนดไว้ภายใต้ RFC 903 โดยรูปแบบเฟรมข้อมูลของ ARP และ RARP จะมีลักษณะเหมือนกัน ดังรูปที่ 2.8



รูปที่ 2.8 ARP และ RARP Frame Format

ที่มา: [2]

จากรูปที่ 2.8 แสดงรายละเอียดของเฟรมข้อมูลของโพรโทคอล ARP ได้ดังตารางที่ 2.2

ตารางที่ 2.2 รายละเอียดของ Address Resolution Protocol Format

ชื่อ	คำอธิบาย
Hardware Type	ระบุชนิดของฮาร์ดแวร์ที่ ARP ทำงานอยู่ มีความยาว 16 บิต
Protocol Type	ระบุว่าเฟรมข้อมูล ARP นี้ถูกเรียกใช้โดยโพรโทคอลใด มีความยาว 16 บิต
Hardware Address Length	ระบุความยาวของ Hardware Address มีความยาว 8 บิต
Protocol Address Length	ระบุความยาวของ Address ของโพรโทคอลที่เรียกใช้ มีความยาว 8 บิต

ที่มา: [2]

ตารางที่ 2.2 (ต่อ) รายละเอียดของ Address Resolution Protocol Format

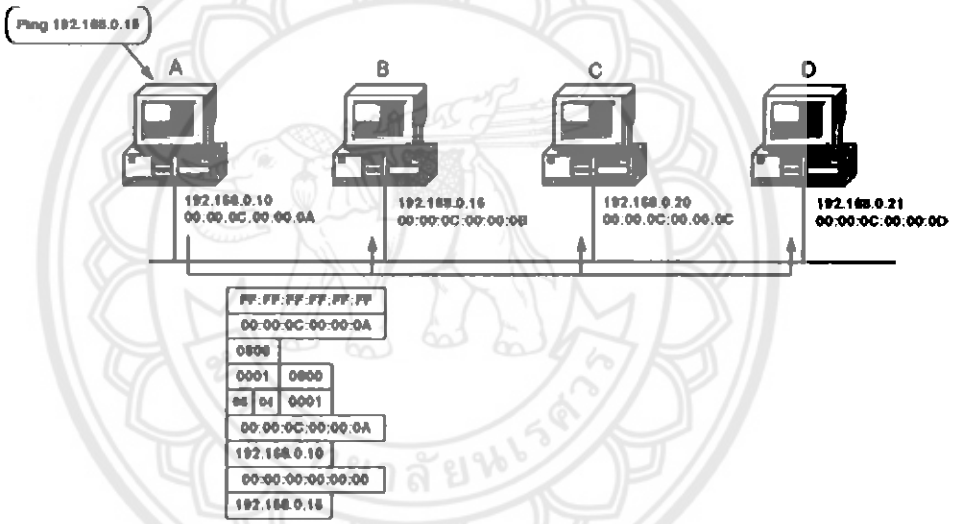
ชื่อ	คำอธิบาย
Operation	ระบุหน้าที่การทำงานของโพรโทคอล ARP มีความยาว 16 บิต ซึ่งจะมี 4 ค่าดังนี้ - 1 หมายถึง ARP Request ใช้ในการค้นหาหมายเลข Hardware Address - 2 หมายถึง ARP Replay ใช้ในการตอบกลับเพื่อบอกหมายเลข Hardware Address - 3 หมายถึง RARP Request ใช้ในการค้นหาหมายเลขของโพรโทคอล - 4 หมายถึง RARP Replay ใช้ในการตอบกลับเพื่อบอกหมายเลขของโพรโทคอล
Sender Hardware Address (OCTETS 0-3)	ใช้ในการเก็บค่า Hardware Address ของผู้ส่ง ไม่จำกัดความยาว
Sender Hardware Address (OCTETS 4-5)	ใช้ในการเก็บค่า Hardware Address ของผู้ส่ง ไม่จำกัดความยาว
Sender IP Address (OCTETS 0-1)	ใช้ในการเก็บค่า IP Address ของผู้ส่ง
Sender IP Address (OCTETS 2-3)	ใช้ในการเก็บค่า IP Address ของผู้ส่ง
Target Hardware Address (OCTETS 0-1)	ระบุ Hardware Address ของผู้รับ
Target Hardware Address (OCTETS 2-5)	ระบุ Hardware Address ของผู้รับ
Target IP Address	ระบุ IP Address ของผู้รับ

ที่มา: [2]

2.5.1 การทำงานของโพรโทคอล ARP [2]

การทำงานของโพรโทคอล ARP จะเริ่มการทำงานเมื่อระบบ IP ต้องการที่จะส่ง Packet ข้อมูลออกไปยังปลายทาง ตัวอย่างเช่น ทำงานกับเครือข่ายอินเทอร์เน็ต ระบบ IP จะต้องอาศัยเฟรมของอินเทอร์เน็ตในการส่งข้อมูล แต่เนื่องจากกระบวนการในการส่งเฟรมอินเทอร์เน็ตจากต้นทางไปยังปลายทางนั้นจำเป็นต้องทราบ Hardware Address ของอุปกรณ์ต้นทางและอุปกรณ์ปลายทาง

สำหรับหมายเลข Hardware Address ของอุปกรณ์ต้นทางนั้น อุปกรณ์ที่จะส่งย่อมทราบหมายเลข Hardware Address อยู่แล้ว หากยังไม่ทราบหมายเลข Hardware Address ของอุปกรณ์ปลายทาง ซึ่งระบบ IP ก็จะมีการใช้บริการของโพรโทคอล ARP ในการหาหมายเลข Hardware Address ของอุปกรณ์ปลายทางโดยการ Broadcast ไปในระบบ ดังตัวอย่างในรูปที่ 2.9

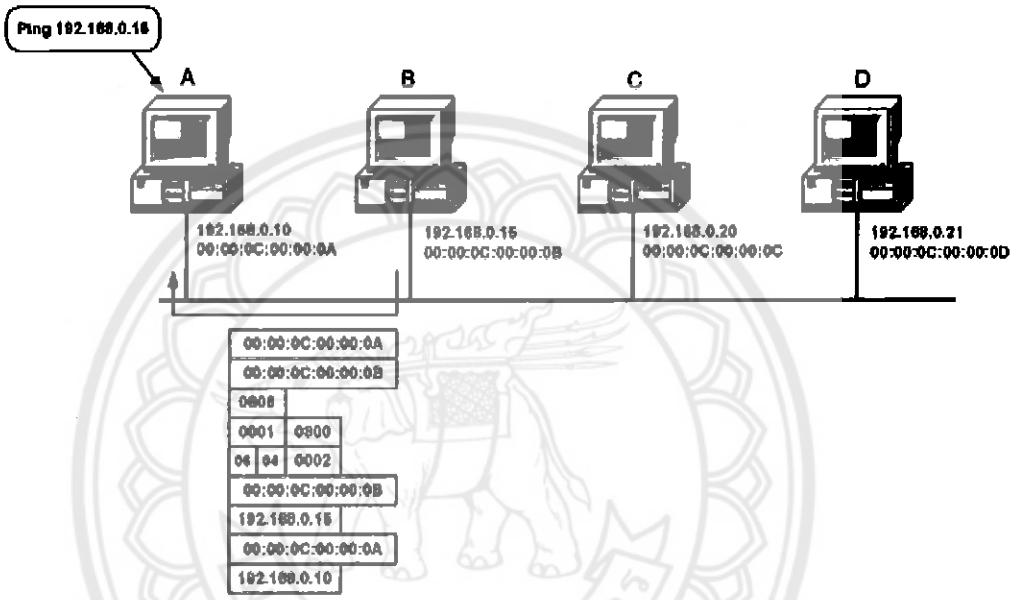


รูปที่ 2.9 การส่ง Packet ของโพรโทคอล ARP

ที่มา: [3]

จากรูปที่ 2.9 คอมพิวเตอร์ A ต้องการติดต่อกับคอมพิวเตอร์ B จึงทำการส่งPacket ARP เข้าไปในเครือข่ายแบบ Broadcast ซึ่งเครื่องคอมพิวเตอร์ทุกเครื่องที่อยู่ในเครือข่ายก็จะได้รับ Packet นี้ทั้งหมดโดยกำหนดให้ Hardware มีค่าเท่ากับ 1 ซึ่งหมายความว่าเครือข่ายนี้เป็นเครือข่ายอินเทอร์เน็ต และโพรโทคอลมีค่าเท่ากับ 0800h หมายถึงถูกเรียกใช้งานจากโพรโทคอล IP ส่วนของ Hardware Length มีค่าเท่ากับ 6 หมายถึงว่า Hardware Address ของเครือข่ายอินเทอร์เน็ตมีความยาว 6 ไบต์ ส่วนของ Protocol Length เท่ากับ 4 หมายถึงความยาวของ Address ของโพรโทคอลที่เรียกใช้ ซึ่งก็คือ IP Address มีความยาวเท่ากับ 4 ไบต์ ในส่วนของ Operation เป็น 1 หมายความว่าเป็นการส่งแบบ ARP Request สำหรับ Hardware ของผู้รับนั้นจะมีค่าเป็น

00:00:00:00:00:00 เนื่องจากยังไม่ทราบว่าผู้รับมี Hardware Address เป็นเท่าใด คอมพิวเตอร์ทุกเครื่องในเครือข่ายจะได้รับ Packet นี้ แต่จะมีเพียงเครื่องที่มีหมายเลข IP Address ตรงกับที่ระบุไว้เท่านั้นที่จะรับ Packet นี้ไปประมวลผลและทำการตอบกลับ ดังนั้นเมื่อเครื่องคอมพิวเตอร์ได้รับ Packet ดังกล่าวแล้ว จะต้องทำการเปรียบเทียบกับหมายเลข IP Address ของตนเองว่าตรงหรือไม่ หากค่าไม่ตรงกันก็จะทำการทิ้ง Packet นั้น ๆ ไป หากตรงกันก็จะทำการรับเข้ามาแล้วทำการตอบกลับไป ดังรูปที่ 2.10



รูปที่ 2.10 การตอบกลับของโพรโทคอล ARP

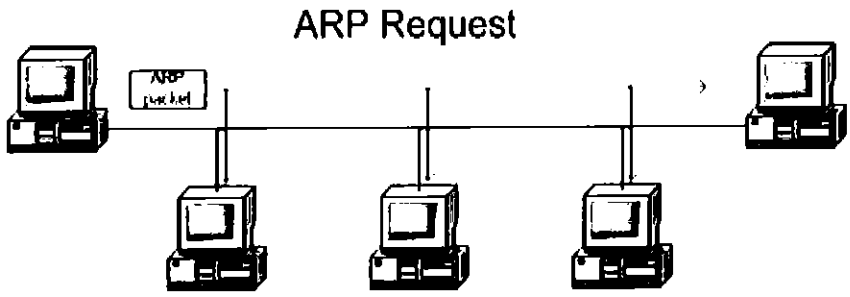
ที่มา: [3]

จากรูปที่ 2.10 จะเห็นว่าเครื่องคอมพิวเตอร์ B จะทำการส่ง Packet ARP กลับมายังเครื่องคอมพิวเตอร์ A ในแบบ Unicast เนื่องจากเครื่องคอมพิวเตอร์ B ทราบอยู่แล้วว่าเครื่องคอมพิวเตอร์ A มีหมายเลข Hardware Address เป็นเท่าใด จาก Packet ARP ที่ส่งมาก่อนหน้านี้

2.5.2 กลไกการทำงานของโพรโทคอล ARP

กลไกการทำงานของโพรโทคอล ARP จะมีอยู่เพียง 2 ขั้นตอนเท่านั้น ได้แก่

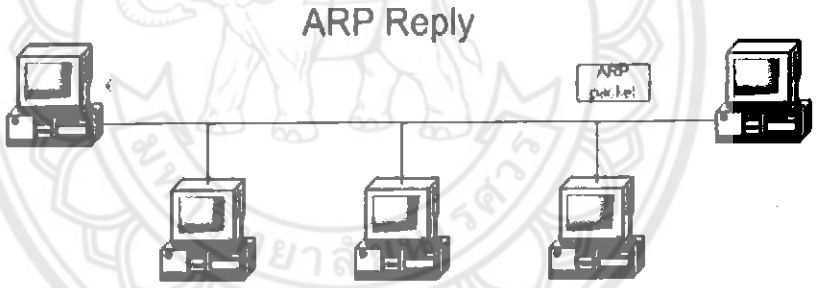
1. เครื่องที่ต้องการสอบถาม MAC Address จะทำการส่ง ARP Packet ซึ่งเรียกว่า ARP Request ภายในบรรจุ IP Address, MAC Address ของตนเองและ IP Address ของเครื่องที่ต้องการทราบ MAC Address ส่วน MAC Address ปลายทางนั้นจะถูกกำหนดเป็น FF:FF:FF:FF:FF:FF ซึ่งเป็น Broadcast address เพื่อให้ ARP Packet ถูกส่งไปยังเครื่องคอมพิวเตอร์ทุกเครื่องที่อยู่ในภายในเครือข่ายเดียวกัน



รูปที่ 2.11 ARP Request จะถูกส่งไปยังเครื่องคอมพิวเตอร์ทุกเครื่องในเครือข่าย

ที่มา: [2]

2. เฉพาะเครื่องที่มี IP Address ตรงกับที่ระบุไว้ใน ARP Packet จะตอบกลับมาด้วย ARP Packet เช่นเดียวกัน โดยใส่ MAC Address และ IP Address ของตนเองเป็นผู้ส่งและใส่ MAC Address และ IP Address ของเครื่องคอมพิวเตอร์ที่ส่งมาให้เป็นผู้รับ Packet ที่ตอบกลับนี้เรียกว่า ARP Reply

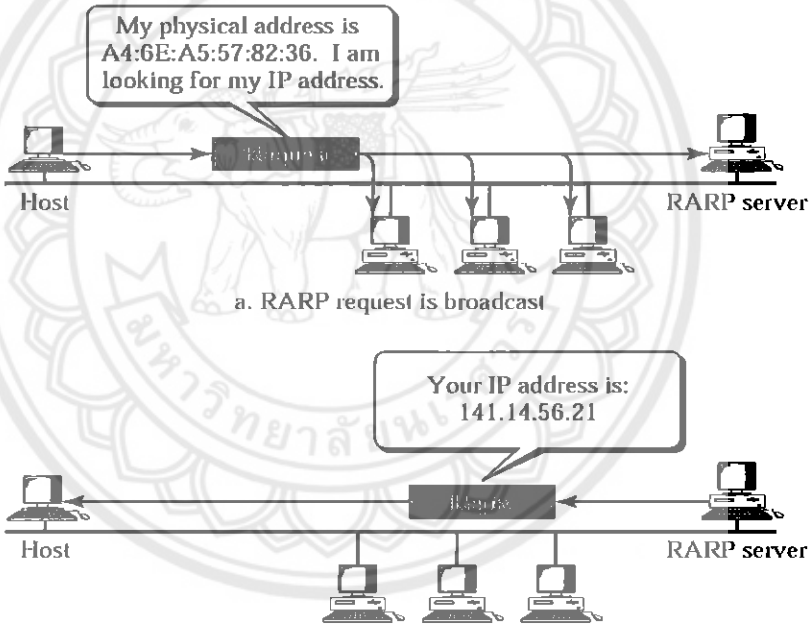


รูปที่ 2.12 ARP Reply จะถูกตอบกลับมาเพื่อบอก MAC Address ของตนเอง

ที่มา: [2]

2.6 Reverse Address Resolution Protocol (RARP) [2]

โพรโทคอล RARP จะมีการทำงานที่คล้ายคลึงกับโพรโทคอล ARP โดยจะทำงานในลักษณะตรงกันข้ามด้วยการแปลงหมายเลข MAC Address[2] ให้เป็นหมายเลข IP Address ซึ่งโพรโทคอล RARP นี้ได้รับการออกแบบมาเพื่อใช้งานกับเครื่องคอมพิวเตอร์ที่ปราศจากดิสก์หรือฮาร์ดดิสก์ (Diskless Compurter) ดังนั้นเวลาที่มีการบูทเครื่องคอมพิวเตอร์ประเภทนี้จึงจำเป็นต้อง boot จากระบบปฏิบัติการเครือข่ายของเซิร์ฟเวอร์บนเครือข่าย โดยเซิร์ฟเวอร์บนเครือข่ายจะจัดเก็บตารางความสัมพันธ์ระหว่าง MAC Address และ IP Address โดยเครื่อง Host ที่ต้องการหมายเลข IP Address จะทำการส่ง RARP Query Packet แบบ Broadcast ที่ภายในประกอบด้วย Physical Address เข้าไปยัง ทุก ๆ เครื่องบนเครือข่าย จากนั้นเครื่องเซิร์ฟเวอร์บนเครือข่ายก็จะทำการจัดการกับ RARP Packet ด้วยการตอบกลับไปด้วยหมายเลข IP Address ไปยัง Host นั้น ๆ



a. RARP request is broadcast

b. RARP reply is unicast

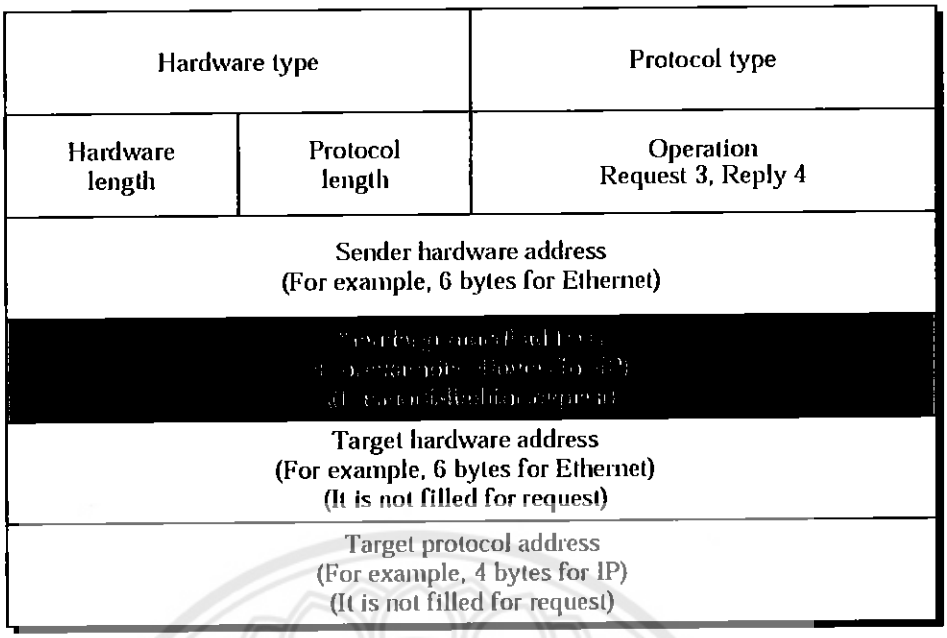
รูปที่ 2.13 RARP Operation

ที่มา: [2]

จากรูปที่ 2.13(a) จะเป็น RARP Request Packet ซึ่งจะถูกส่งออกไปยังเครื่อง Host ทุก ๆ เครื่องในเครือข่ายเพื่อร้องขอ IP Address และในส่วนของรูปที่ 2.13(b) จะเป็น RARP Reply Packet ที่ถูกส่งกลับมาและมีการส่ง IP Address ที่ถูกร้องขอกลับมายังเครื่อง Host นั้น ๆ

---

[2] MAC Address คือ หมายเลขเฉพาะที่ใช้อ้างถึงอุปกรณ์ที่ต่อเข้ากับเครือข่าย หมายเลขนี้จะมากับการ์ดแลน โดยการ์ดแลนแต่ละการ์ดจะมีหมายเลขที่ไม่ซ้ำกัน



รูปที่ 2.14 RARP Packet

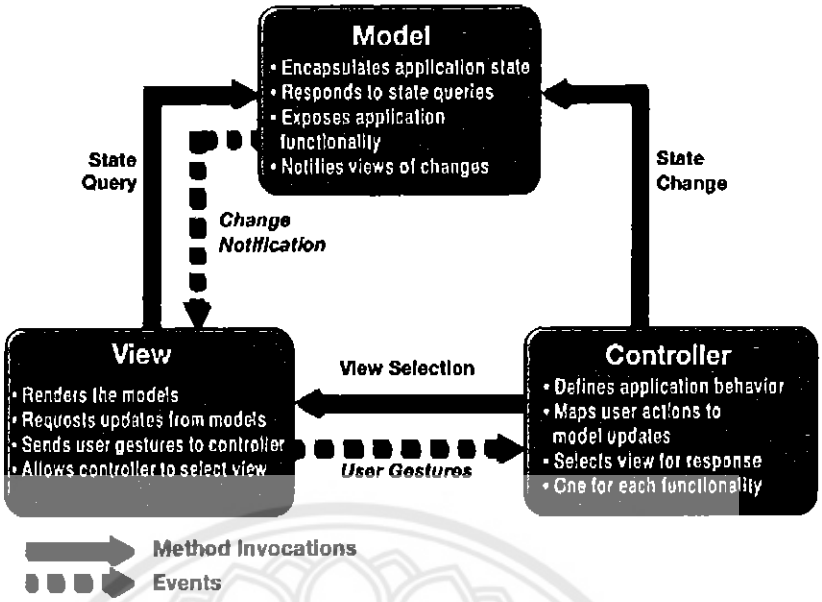
ที่มา: [2]

2.7 Model View Controller (MVC)

MVC เป็นหนึ่งในหลาย ๆ Pattern ของ Design Pattern ซึ่งเป็นแบบจำลองหรือรูปแบบในการวางโครงสร้างโปรแกรมทำให้แยกออกจากกันระหว่าง Model, View และ Controller

MVC เป็น framework ที่เกิดขึ้นมาในสมัย Smalltalk-80 เมื่อกว่าสิบปีมาแล้ว โดยแยก Object ที่เก็บข้อมูล (Model) Object ที่แสดงข้อมูล (View) และ Object ที่ติดต่อกับผู้ใช้ (Controller) ออกจากกันอย่างชัดเจน การสื่อสารระหว่าง Object จะใช้ Observer pattern นั่นหมายความว่า Object ที่ถูก Observe จะแจ้งการเปลี่ยนแปลงที่เกิดขึ้นในตัวมันให้แก่ตัว Observer ทุกตัวโดยไม่จำเป็นต้องรู้ว่าตัว Observer แต่ละตัวเป็นใครหรือเป็น Object ของคลาสไหน ตราบใดที่คลาสนั้นอิมพลีเมนท์ Observer Interface หรือ Listener

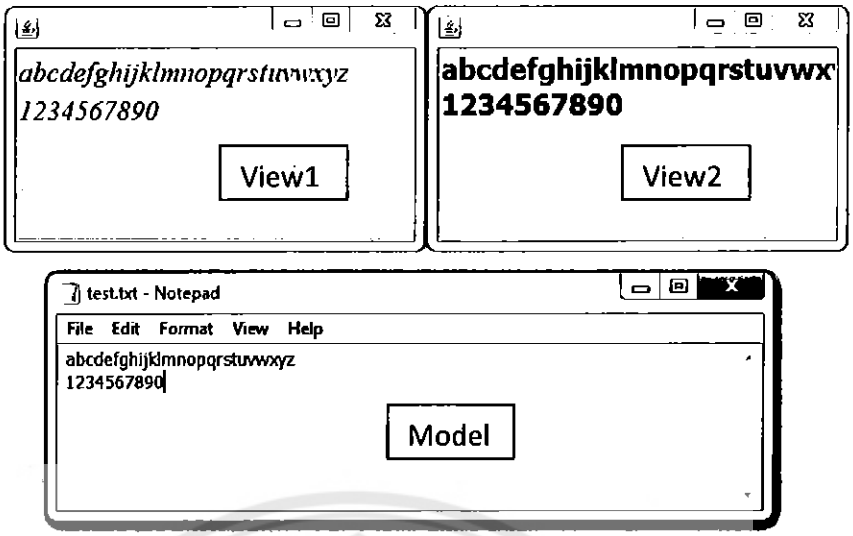
1. Model เป็นการตัดสินใจในการเข้าถึงและใช้งานข้อมูล (Data)
2. View เป็นส่วนของการนำข้อมูลที่ได้จาก Model มาแสดงผลให้ผู้ใช้ได้ทราบข้อมูลผ่านทางส่วนติดต่อกับผู้ใช้งาน (User Interface)
3. Controller เป็นส่วนที่ตอบรับและโต้ตอบการทำงานของผู้ใช้ (Client) โดยจะเป็นตัวกระตุ้นให้ Model และ View ทำงานไปในทิศทางเดียวกัน



รูปที่ 2.15 โครงสร้างของ MVC

การแยก Object เหล่านี้ออกจากกันจะทำให้ง่ายต่อการแก้ไขโปรแกรม เช่น เราสามารถเปลี่ยน View จาก GUI เป็น HTML รวมทั้ง JSP, ASP, PHP หรือ Text Mode ได้โดยไม่ต้องแก้ไขแบบจำลอง หรือเราจะสนับสนุนหลายๆ View พร้อมๆกัน เช่น ลูกค้าตลาดหุ้นแต่ละคนจะมีหน้าจอดูราคาหุ้นในตลาดหลักทรัพย์ แต่ละคนจะมี View ของตัวเองและจะแสดงข้อมูลหลักทรัพย์เฉพาะที่ตัวเองสนใจ ในขณะที่ Object Model คือหลักทรัพย์ทั้งหมดในตลาด เป็นต้น

หลักการสำคัญ ก็คือ Model จะไม่สามารถเรียกฟังก์ชันของ View หรือ Controllerโดยตรง เนื่องจากมันจะไม่มีตัวแปรเก็บ Object ที่เป็น View หรือ Controller ไว้ Model มีเพียงแต่รายการของ Object ชนิด Observer (View หรือ Controller ที่อิมพลีเมนท์ Observer Interface) ที่เฝ้าสังเกตการณ์เปลี่ยนแปลงอยู่ตัว Model จะแจ้งการเปลี่ยนแปลงไปให้ Observer ทุก ๆ ตัว ถ้า Observer นั้นเป็น View ก็จะ Update หน้าจอด้วยข้อมูลใหม่ เป็นต้น ในทำนองเดียวกันตัว View เองก็มองไม่เห็น Controller และไม่สามารถเรียกฟังก์ชันของ Controller ได้โดยตรง นอกจากจะส่ง Event ไปให้ Controller ที่เป็น Observer ของ View นั้น ๆ



รูปที่ 2.16 ตัวอย่างการเขียนโปรแกรมโครงสร้าง MVC

จากรูปที่ 2.16 เป็นการเขียนโปรแกรมเพื่อทดสอบการทำงานของโครงสร้าง MVC โดยจะมี View ที่ใช้การแสดงผล 2 รูปแบบโดยเมื่อข้อมูลใน Model (ในที่นี้จะถูกผูกไว้กับไฟล์ "test.txt") ถูกเปลี่ยนก็จะทำการเรียกใช้เมธอด Notify ที่อยู่ในคลาส Model เพื่อแจ้งให้ View ทั้งหมดที่ได้ทำการลงทะเบียนไว้ทราบ (ในที่นี้ก็คือ View1 และ View2) ต่อมาคลาสเหล่านี้จะทำการเรียกใช้เมธอด Update เพื่อทำการเปลี่ยนข้อมูลตามข้อมูลที่มีอยู่ใน Model โดยที่การทำงานทั้งหมดนี้จะถูกควบคุมโดยคลาส Controller

บทที่ 3

วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงวิธีการดำเนินงานเพื่อพัฒนาโปรแกรมเพื่อแสดงผลการทำงานเครือข่าย 4 ชั้นของ OSI โดยจะเน้นในส่วนของระดับชั้น Network และระดับชั้น Transport เป็นหลัก โดยได้ทำการเลือกโพรโทคอล ARP และโพรโทคอล RARP ในระดับชั้น Network และเลือกโพรโทคอล UDP ในระดับชั้น Transport

3.1 คำอธิบายของระบบ (System Description)

โปรแกรมจำลองการทำงานสำหรับเครือข่ายโพรโทคอล ในระดับชั้น 3 และระดับชั้น 4 นั้นระบบได้ทำการจำลองการเชื่อมต่อระหว่างระดับชั้น 3 และ 4 กับระดับชั้นอื่น ๆ ของเครือข่ายโพรโทคอลที่ได้ทำการสร้างขึ้นมาก่อนหน้านี้ ทำให้การจำลองมีความสมบูรณ์มากยิ่งขึ้น และมีการแสดงค่าต่าง ๆ ภายในโพรโทคอลนั้น ๆ

3.2 การออกแบบโปรแกรม

การออกแบบโปรแกรมนั้นจะใช้มาตรฐานของ UML (Unified Modeling Language) ที่เป็นภาษาที่ใช้ในการแสดงแบบจำลองการทำงานของระบบ ซึ่งอธิบายพฤติกรรมของซอฟต์แวร์ด้วยแผนภาพตาม View ต่าง ๆ ดังตารางต่อไปนี้

ตารางที่ 3.1 Mapping UML and PSP[3] Views

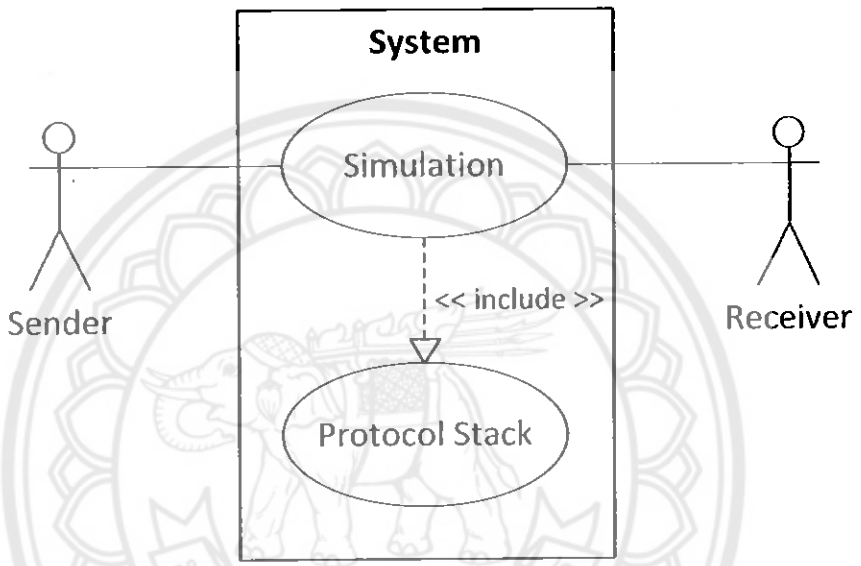
	Dynamic	Static
External	Use cases Diagrams Sequence Diagrams	Class Diagrams
Internal	Activity Diagrams	

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

3.2.1 Overview ของโปรแกรม

3.2.1.1 Use case Diagram

แสดงให้เห็นถึงวิธีการที่ผู้ใช้มีการติดต่อกับระบบ โดยผู้ใช้เลือกชุดโพรโทคอล IP ที่ต้องการส่ง ข้อความที่ต้องการส่ง เลือกว่าจะแสดงข้อมูลเป็นแบบข้อความ หรือแบบกราฟิก (GUI) ผู้ใช้เลือกว่าจะเป็นฝั่งรับหรือฝั่งส่งผ่าน Configuration Layer ส่วนการจำลองการทำงาน (Simulate) ของระบบนั้นกระทำผ่าน Configuration Layer เช่นเดียวกัน

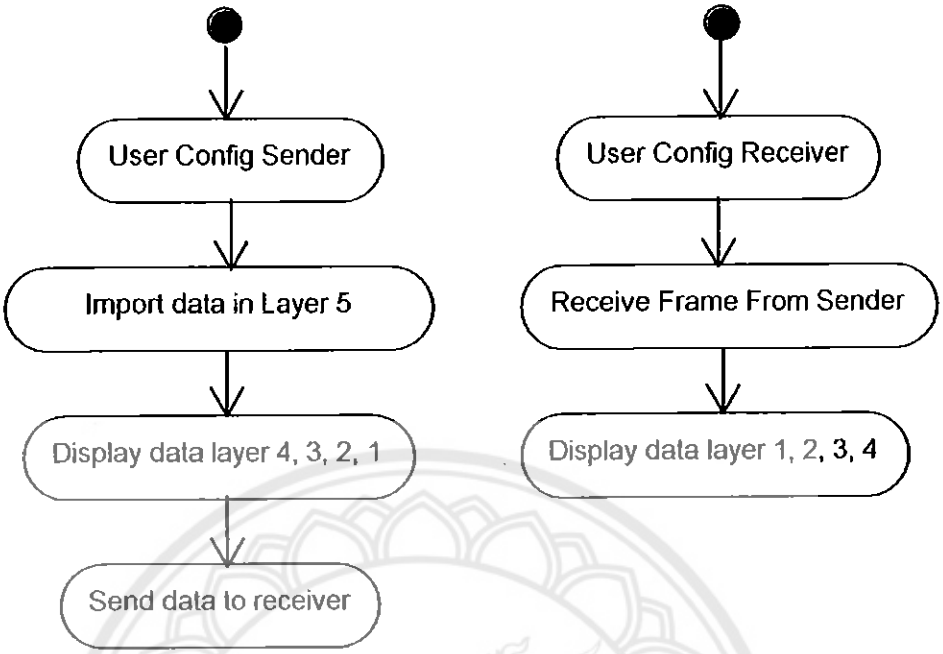


รูปที่ 3.1 Use case Diagram ของระบบ

3.2.1.2 Activity Diagram

แสดงให้เห็นถึงวิธีการทำงานภายในคลาสที่สร้างขึ้นมาการไหลของเหตุการณ์ (Flow of Events) ของโปรแกรม

1. เมื่อเข้าสู่โปรแกรมแล้วให้ผู้ใช้ทำการเลือกว่าจะเป็นฝ่ายผู้รับหรือฝ่ายผู้ส่งข้อความ ถ้าเป็นฝ่ายผู้ส่งก็จะมีการรับข้อมูลมาจากระดับชั้น 5 ถ้าเป็นฝ่ายผู้รับก็จะมีการรอจนกว่าจะได้รับข้อมูลจากฝ่ายผู้ส่ง
2. โปรแกรมจะมีการแสดงการทำงานกับข้อมูลในระดับชั้นที่ 4, 3, 2 และ 1 จากนั้นจะส่งข้อมูลไปยังฝ่ายผู้รับ
3. โปรแกรมจะมีการรับเฟรมข้อมูลจากฝ่ายผู้ส่งมาและแสดงการทำงานกับข้อมูลในระดับชั้นที่ 1, 2, 3 และ 4



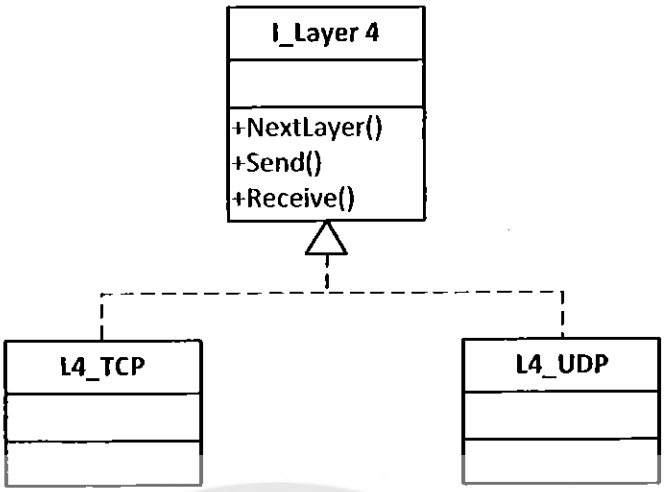
รูปที่ 3.2 Activity Diagram ของระบบ

3.2.2 ขั้นตอนการทำงานของโพรโทคอล UDP

ในส่วนนี้จะแสดงโครงสร้าง ลำดับขั้นตอนการทำงานของโพรโทคอล UDP ในระดับชั้น 4 ซึ่งจะแสดงในรูปของ Class Diagram และ Sequence Diagram

3.2.2.1 Class Diagram ของโพรโทคอล UDP

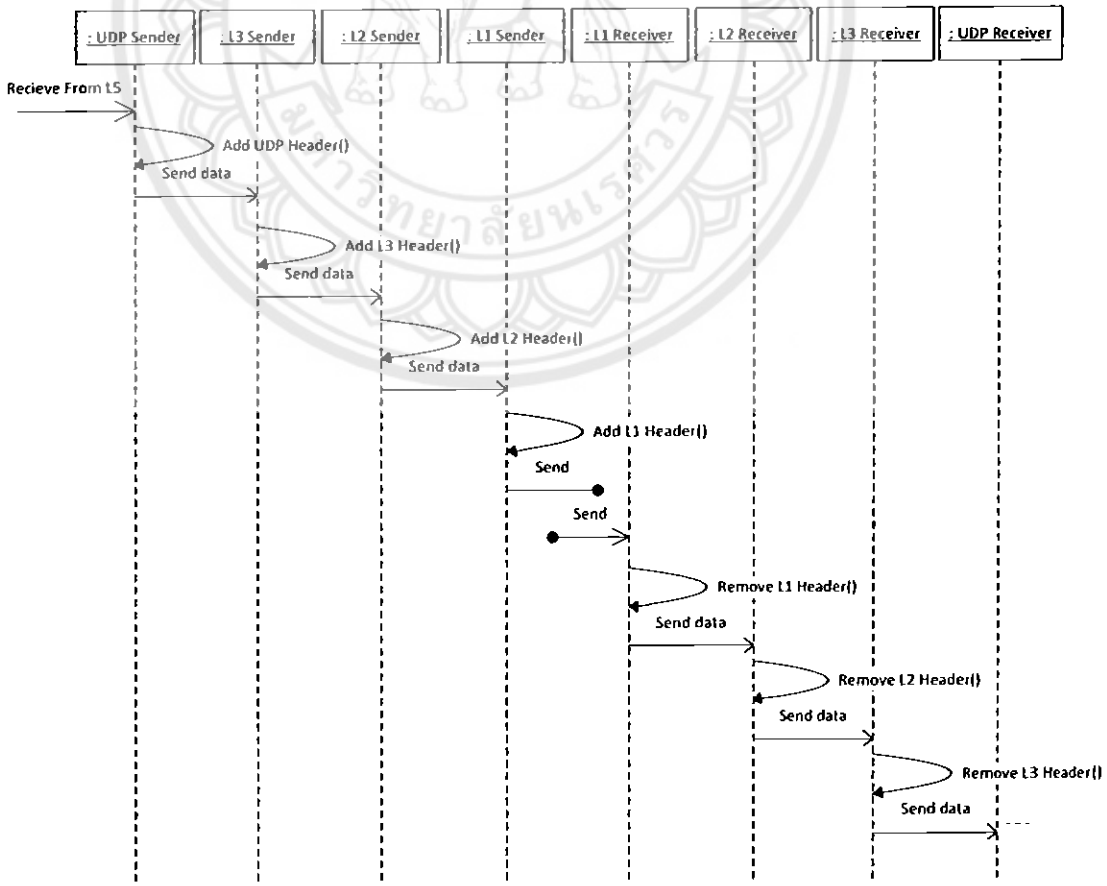
สร้างคลาสเพื่อรองรับการทำงานของโพรโทคอล UDP ซึ่งเดิมนั้นโปรแกรมมีการอิมพลีเมนท์การทำงานของโพรโทคอล TCP อยู่แล้วซึ่งในส่วนนี้ได้ทำการเพิ่มการอิมพลีเมนท์โพรโทคอล UDP เพิ่มลงไปในระดับชั้น 4 แสดงได้ดัง Class Diagram ดังรูปที่ 3.3



รูปที่ 3.3 Class Diagram ของโพรโทคอล UDP

3.2.2.2 Sequence Diagram ของโพรโทคอล UDP

Sequence Diagram แสดงให้เห็นถึงวิธีการที่โพรโทคอล UDP ติดต่อกับระดับชั้นอื่น ๆ ในเครือข่ายโพรโทคอลโดยจะแสดงเป็นลำดับการทำงาน ดังรูปที่ 3.4



รูปที่ 3.4 Sequence Diagram ของโพรโทคอล UDP

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

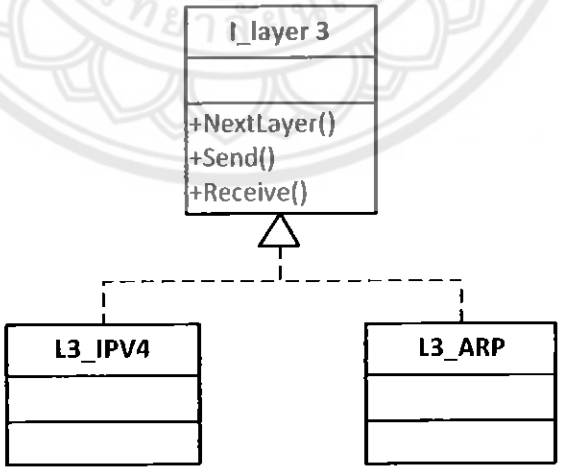
จากรูปที่ 3.4 จะเห็นว่าเมื่อมีการรับข้อมูลมาจากระดับชั้น 5 แล้วในชั้นของโพรโทคอล UDP จะมีการเพิ่ม UDP Header เข้าไปที่ข้อมูลนั้นและทำการส่งข้อมูลซึ่งเรียกว่า Datagram ไปยังระดับชั้นถัดไปซึ่งก็คือระดับชั้น 3 ที่ระดับชั้น 3 จะมีการเพิ่ม L3 Header และทำการส่งข้อมูลไปยังระดับชั้น 2 ที่ระดับชั้น 2 ก็จะมีการเพิ่ม L2 Header เข้าไปเช่นเดียวกัน จากนั้นจะทำการส่งข้อมูลชุดนั้นไปยังระดับชั้น 1 ที่ระดับชั้น 1 จะมีการเพิ่ม L1 Header จากนั้นระบบจะทำการส่งข้อมูลไปยังฝั่งผู้รับโดยเริ่มจากส่งไปยังผู้รับระดับชั้น 1 ที่ระดับชั้น นี้จะมีการถอด L1 Header ออกก่อนที่จะส่งไปยังระดับชั้น 2 และที่ระดับชั้น 2 ก็จะมีการถอด L2 Header ออก จากนั้นจะส่งไปยังผู้รับที่ระดับชั้น 3 ที่ระดับชั้น 3 นี้ ก็จะมีการถอด L3 Header ออกเช่นเดียวกับระดับชั้นที่ผ่าน ๆ มา และทำการส่งข้อมูลไปยังระดับชั้นถัดไป

3.2.3 ขั้นตอนการทำงานของโพรโทคอล ARP

ในส่วนนี้จะแสดงโครงสร้าง ลำดับขั้นตอนการทำงานของโพรโทคอล ARP ในระดับชั้น 3 ซึ่งจะแสดงในรูปของ class Diagram และ sequence Diagram

3.2.3.1 Class Diagram ของโพรโทคอล ARP

สร้างคลาสเพื่อรองรับการทำงานของโพรโทคอล ARP ซึ่งเดิมนั้น โปรแกรมมีการอิมพลีเมนท์ การทำงานของโพรโทคอล IPv4 อยู่แล้วซึ่งในส่วนนี้ได้ทำการเพิ่มการอิมพลีเมนท์โพรโทคอล ARP เพิ่มลงไปในระดับชั้น 3 แสดงได้ดัง class Diagram ดังรูปที่ 3.4

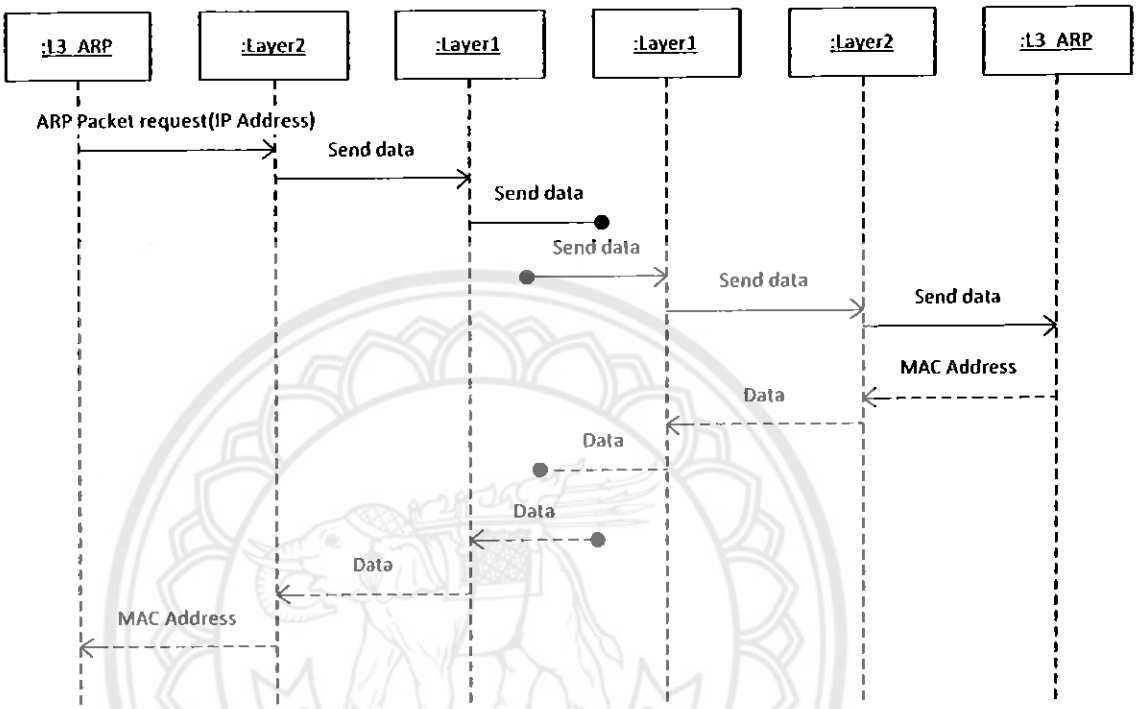


รูปที่ 3.5 Class Diagram ของโพรโทคอล ARP

167338
ป.ร.
ว335 ป
2553

3.3.3.2 Sequence Diagram ของโพรโทคอล ARP

Sequence Diagram แสดงให้เห็นถึงวิธีการที่โพรโทคอล ARP ติดต่อกับระดับชั้นอื่น ๆ ในเครือข่ายโพรโทคอลโดยจะแสดงเป็นลำดับการทำงาน ดังรูปที่ 3.5



รูปที่ 3.6 Sequence Diagram ของโพรโทคอล ARP

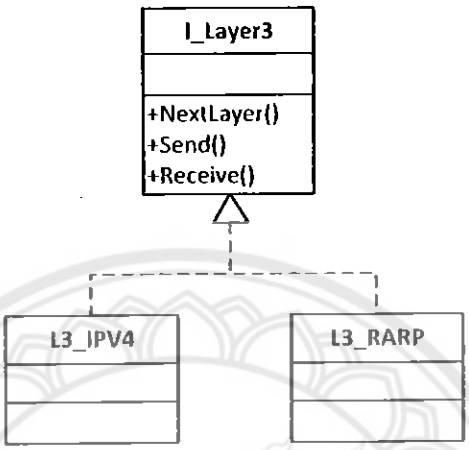
จากรูปที่ 3.5 Sequence Diagram ของโพรโทคอล ARP จะมีการทำงานเป็นดังนี้ เมื่อมีการส่งข้อมูลโพรโทคอล ARP จะทำการส่ง ARP Packet Request ออกไปพร้อมกับ IP Address ไปยังระดับชั้นถัดไป และจะมีการส่งข้อมูลชุดนั้น ๆ ไปยังฝั่งผู้รับ เมื่อฝั่งผู้รับมีการรับข้อมูลจนถึงระดับชั้นที่ 3 แล้วโพรโทคอล ARP ซึ่งอยู่ที่ฝั่งผู้รับจะทำการส่ง ARP Packet Reply กลับไปเพื่อระบุ MAC Address และส่งข้อมูลกลับไปยังผู้ส่งอีกครั้งหนึ่ง

3.2.4 ขั้นตอนการทำงานของโพรโทคอล RARP

ในส่วนนี้จะแสดงโครงสร้าง ลำดับขั้นตอนการทำงานของโพรโทคอล RARP ในระดับชั้น 3 ซึ่งจะแสดงในรูปของ class Diagram และ sequence Diagram

3.2.4.1 Class Diagram ของโพรโทคอล RARP

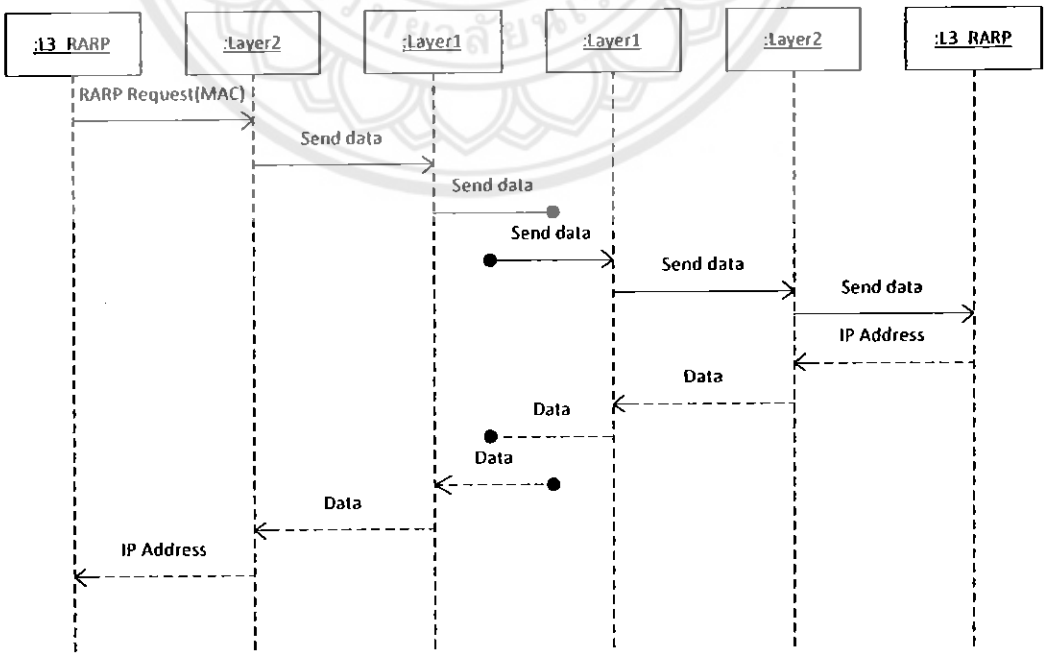
สร้างคลาสเพื่อรองรับการทำงานของโพรโทคอล RARP ซึ่งเดิมนั้นโปรแกรมมีการอิมพลีเมนท์การทำงานของโพรโทคอล IPv4 อยู่แล้วซึ่งในส่วนนี้ได้ทำการเพิ่มการอิมพลีเมนท์โพรโทคอล RARP เพิ่มลงไปในระดับชั้น 3 แสดงได้ดัง class Diagram ดังรูปที่ 3.6



รูปที่ 3.7 Class Diagram ของโพรโทคอล RARP

3.2.4.2 Sequence Diagram ของโพรโทคอล RARP

Sequence Diagram แสดงให้เห็นถึงวิธีการที่โพรโทคอล RARP ติดต่อกับระดับชั้น อื่น ๆ ในเครือข่ายโพรโทคอล โดยจะแสดงเป็นลำดับการทำงาน ดังรูปที่ 3.7

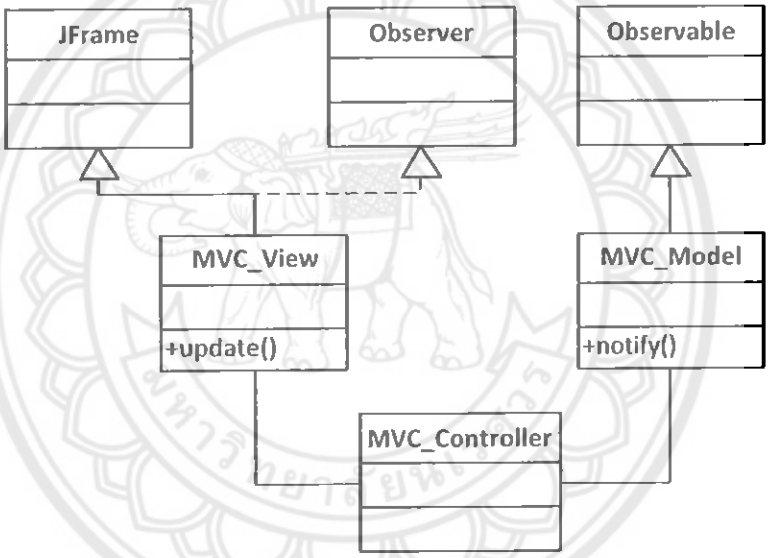


รูปที่ 3.8 Sequence Diagram ของโพรโทคอล RARP

จากรูปที่ 3.7 Sequence Diagram ของโพรโทคอล RARP จะมีการทำงานเป็นดังนี้ เมื่อมีการส่งข้อมูลโพรโทคอล RARP จะทำการส่ง RARP Packet Request ออกไปพร้อมกับ MAC Address ไปยังระดับชั้นถัดไป และจะมีการส่งข้อมูลชุดนั้น ๆ ไปยังฝั่งผู้รับ เมื่อฝั่งผู้รับมีการรับข้อมูลจนถึงระดับชั้น ที่ 3 แล้วโพรโทคอล RARP ซึ่งอยู่ที่ฝั่งผู้รับจะทำการส่ง RARP Packet Reply กลับไปเพื่อระบุ IP Address และส่งข้อมูลกลับไปยังผู้ส่งอีกครั้งหนึ่ง

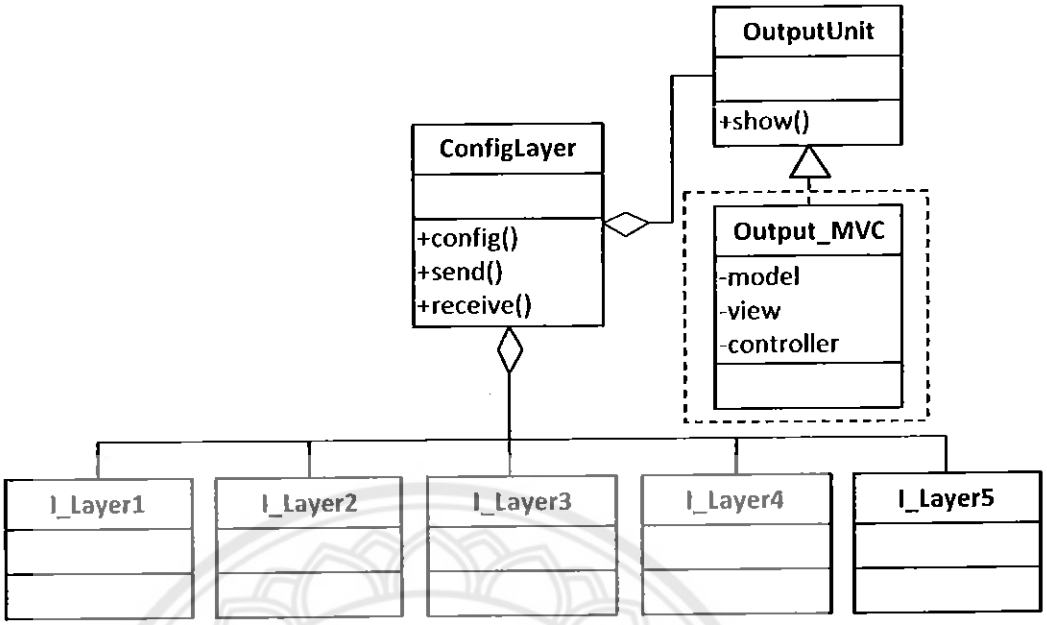
3.2.5 การออกแบบโปรแกรมให้ทำงานตามรูปแบบของ Model View Controller

การออกแบบโครงสร้างของ Model View Controller ทำได้โดยอาศัยคลาสที่ทางJava เตรียมไว้ให้คือ Observer และ Observable ซึ่งสามารถอธิบายความสัมพันธ์ที่จะนำมาทำเป็นโครงสร้างแบบ Model View Controller โดยการแสดงเป็น Class Diagram ได้ดังรูปที่ 3.8



รูปที่ 3.9 Class Diagram แสดงโครงสร้างของ MVC ของภาษา Java

ซึ่งในส่วนนี้เมื่อคลาส Model มีการเปลี่ยนแปลงของข้อมูลจะมีการเรียกใช้เมธอด Notify เพื่อแจ้งไปให้คลาส View ผ่านเมธอด Update เพื่อที่จะทำการปรับเปลี่ยนรูปแบบการแสดงผลให้ตรงตามข้อมูลที่เปลี่ยนแปลงไป



รูปที่ 3.10 Class Diagram แสดงการอิมพลีเมนท์ MVC

จากเดิมทางผู้จัดทำได้เพิ่มการอิมพลีเมนท์คลาส "Output_MVC" ที่คลาส "OutputUnit" โดยมี Attribute เป็น Model View และ Controller

โดยเมื่อเริ่มทำงาน ในแต่ละ Layer จะมีการเรียกใช้เมธอด Show ซึ่งอยู่ในคลาส "Output_MVC" เพื่อแสดงผลลัพธ์ในแต่ละเลเยอร์ ผู้จัดทำจึงได้นำแนวคิด MVC มาใช้ในส่วนนี้คือ นำส่วนของ Model มาทำการ Notify ที่เมธอด Show ในคลาส "Output_MVC" เพื่อทำการแจ้งให้ View ที่ได้ทำการลงทะเบียนเพื่อรับข้อมูลจาก Model จากนั้นจึงทำการ Update เพื่อเปลี่ยนแปลงข้อมูลที่ตัว Model ได้ทำการ Notify ไปให้

บทที่ 4

ผลการทดลอง

4.1 แผนการทดสอบโปรแกรม

ตารางที่ 4.1 ตารางแผนการทดสอบ

Test	รายการ
System Test	ทดสอบการแสดงผลของหน้าผู้ใช้แบบ Graphic User Interface (GUI) ทั้งฝั่ง Sender และฝั่ง Receiver
Integration Test	ทดสอบการแสดงผลการทำงานของโพรโทคอล TCP
	ทดสอบการแสดงผลการทำงานของโพรโทคอล UDP
	ทดสอบการแสดงผลการทำงานของโพรโทคอล ARP
	ทดสอบการแสดงผลการทำงานของโพรโทคอล RARP
	ทดสอบการแสดงผลการทำงานของโพรโทคอล IPv4
Unit Test	ทดสอบการสร้าง Header ของโพรโทคอล TCP
	ทดสอบการสร้าง Header ของโพรโทคอล UDP
	ทดสอบการสร้าง Header ของโพรโทคอล ARP
	ทดสอบการสร้าง Header ของโพรโทคอล RARP

4.2 การทดสอบโปรแกรม

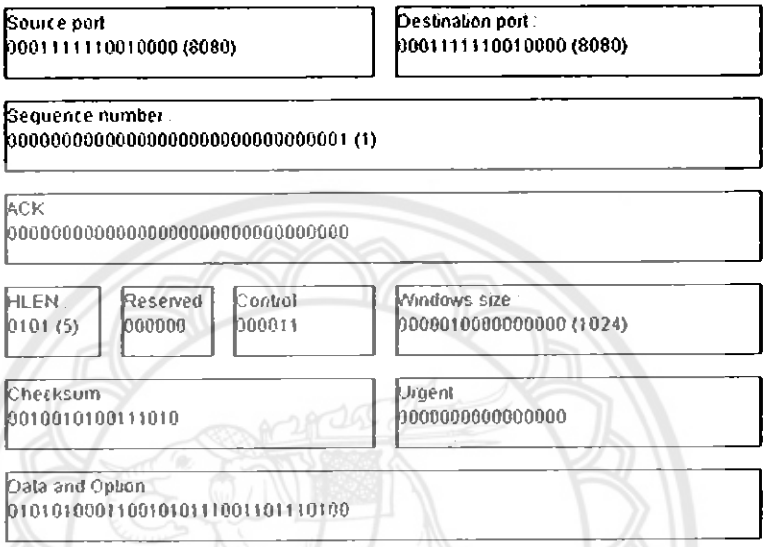
4.2.1 Unit Test

ตารางที่ 4.2 ตารางการทดสอบโปรแกรม Unit Test

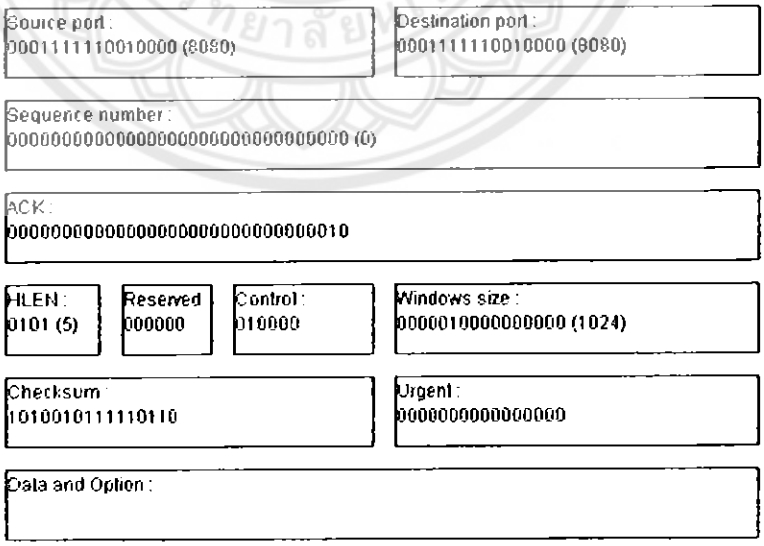
Class	Method	Input	Expected Result
L4_TCP	Send	Message "Test"	รูปที่ 4.1 และ 4.2
	Receive	รูปที่ 4.1	รูปที่ 4.3 และ 4.4
L4_UDP	Send	Message "Test"	รูปที่ 4.5
	Receive	รูปที่ 4.5	รูปที่ 4.6
L3_ARP	Send	ARP Request	รูปที่ 4.7
	Receive	รูปที่ 4.7	รูปที่ 4.8
L3_RARP	Send	RARP Request	รูปที่ 4.9
	Receive	รูปที่ 4.9	รูปที่ 4.10

4.2.2 การทดสอบการส่งของโพรโทคอล TCP

ทดสอบการแสดงผลการทำงานของโพรโทคอล TCP ฝั่ง Sender โดยกำหนดให้ส่ง Message คำว่า "Test" ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.1 และรูปที่ 4.2 จะเป็นการแสดงผลการทำงานของ Acknowledge ของโพรโทคอล TCP ที่ฝั่ง Sender ได้รับกลับมา



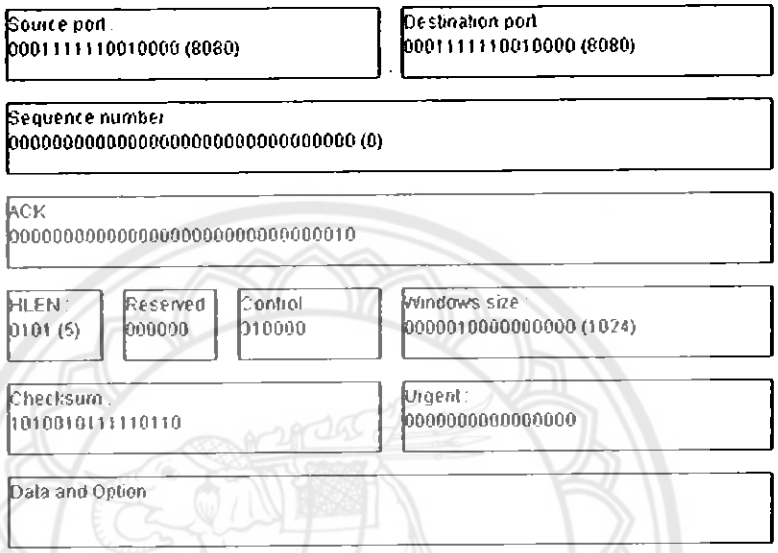
รูปที่ 4.1 รายละเอียดของโพรโทคอล TCP ฝั่ง Sender



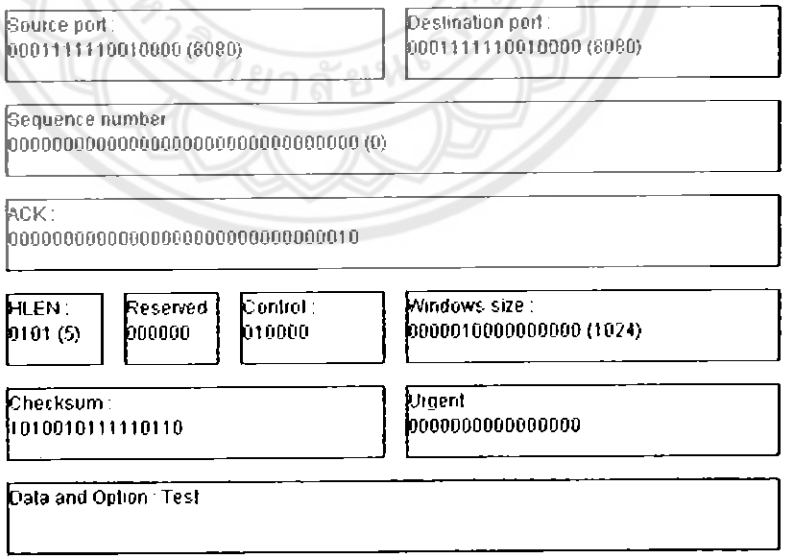
รูปที่ 4.2 รายละเอียดของ Acknowledge ของโพรโทคอล TCP ฝั่ง Sender

4.2.3 การทดสอบการรับของโพรโทคอล TCP

ทดสอบการแสดงผลการทำงานของโพรโทคอล TCP ฝั่ง Receiver โดยจะรับ Message คำว่า "Test" มาจากฝั่ง Sender ซึ่งจะได้รับ Acknowledge ตอบรับจากฝั่ง Sender ดังปรากฏในรูปที่ 4.3 และในรูปที่ 4.4 จะแสดงผลการทำงานของโพรโทคอล TCP ฝั่ง Receiver



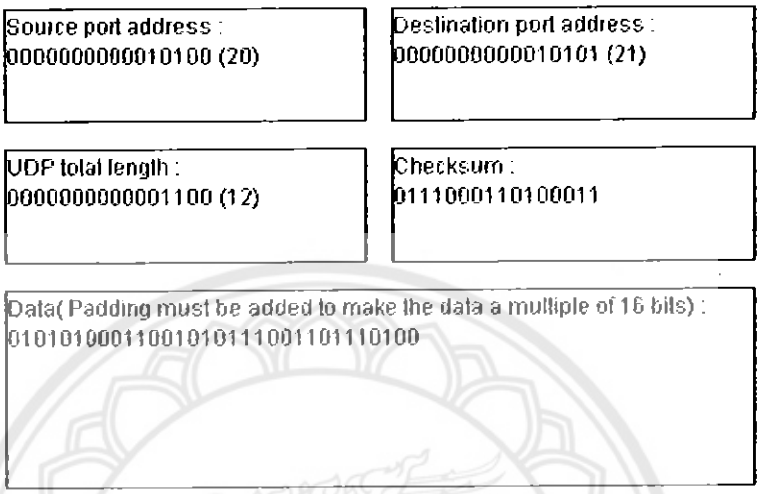
รูปที่ 4.3 รายละเอียดของ Acknowledge ของโพรโทคอล TCP ฝั่ง Receiver



รูปที่ 4.4 รายละเอียดของโพรโทคอล TCP ฝั่ง Receiver

4.2.4 การทดสอบการส่งของโพรโทคอล UDP

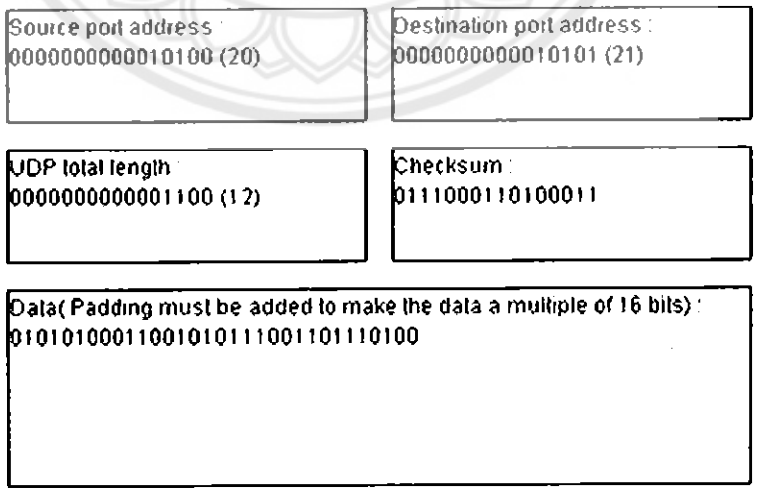
ทดสอบการแสดงผลการทำงานของโพรโทคอล UDP ฝั่ง Sender โดยกำหนดให้ส่ง Message คำว่า "Test" ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.5



รูปที่ 4.5 รายละเอียดของโพรโทคอล UDP ฝั่ง Sender

4.2.5 การทดสอบการรับของโพรโทคอล UDP

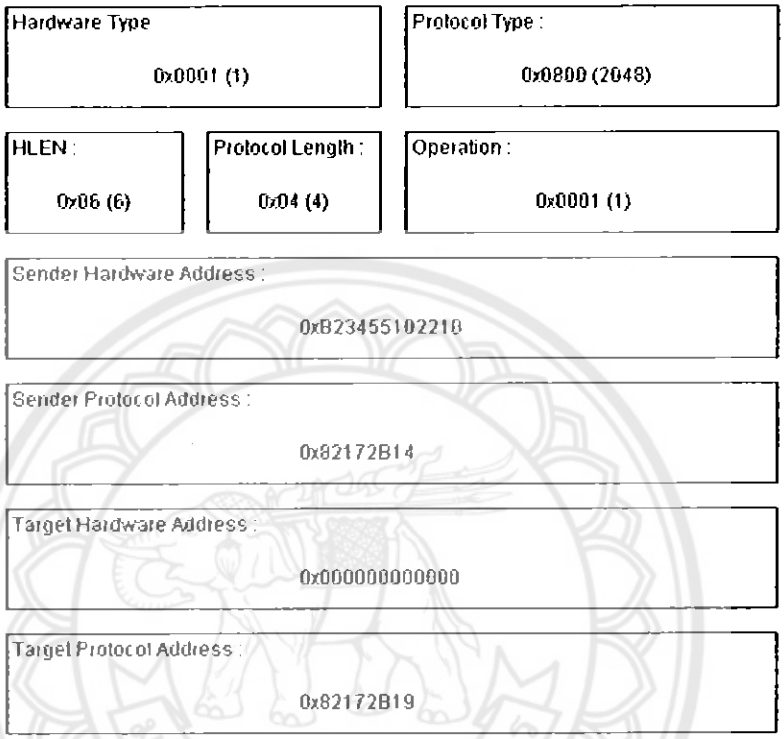
ทดสอบการแสดงผลการทำงานของโพรโทคอล TCP ฝั่ง Receiver โดยจะรับ Message คำว่า "Test" มาจากฝั่ง Sender ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.6



รูปที่ 4.6 รายละเอียดของโพรโทคอล UDP ฝั่ง Receiver

4.2.6 การทดสอบการส่งของโพรโทคอล ARP

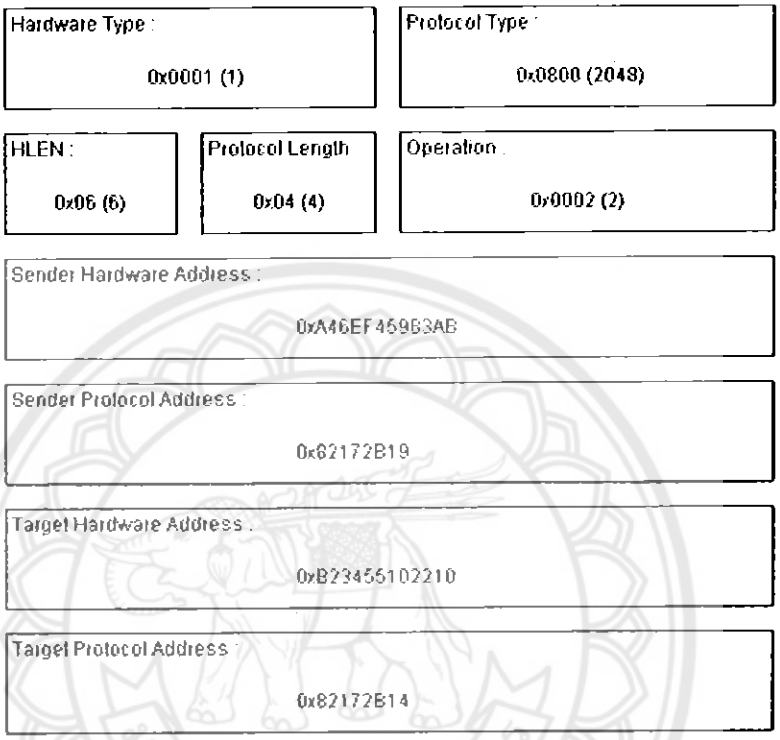
ทดสอบการแสดงผลการทำงานของโพรโทคอล ARP ฝั่ง Sender โดยกำหนดให้ส่ง Message คำว่า "Test" ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.7



รูปที่ 4.7 รายละเอียดของ ARP Request Packet

4.2.7 การทดสอบการรับของโพรโทคอล ARP

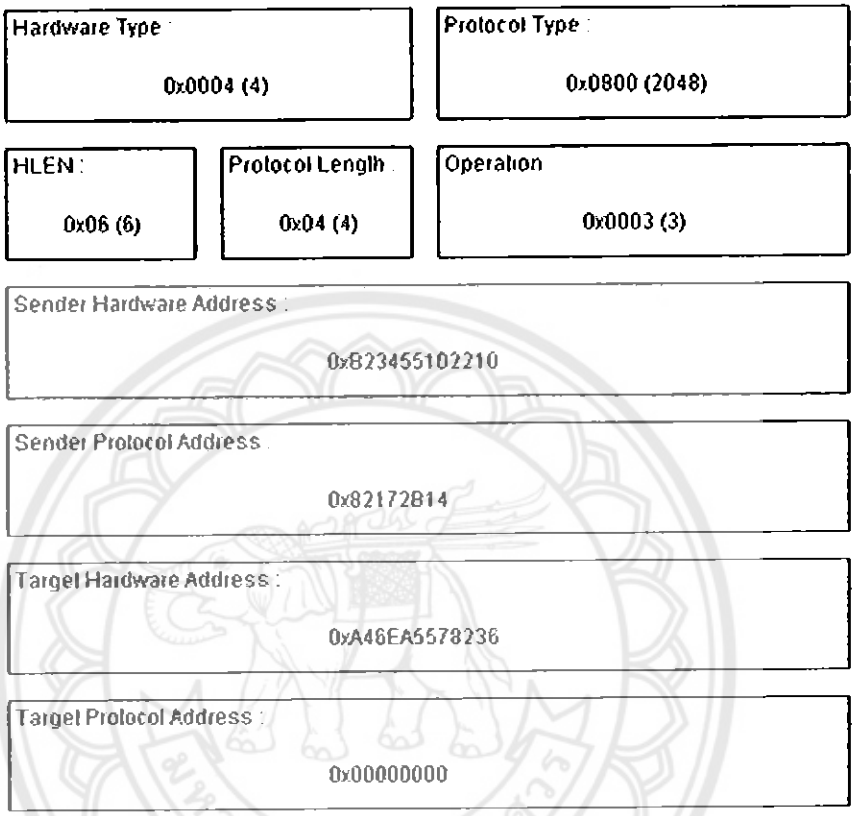
ทดสอบการแสดงผลการทำงานของโพรโทคอล ARP ฝั่ง Receiver โดยจะรับ Message คำว่า "Test" มาจากฝั่ง Sender ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.8



รูปที่ 4.8 รายละเอียดของ ARP Reply Packet

4.2.8 การทดสอบการส่งของโพรโทคอล RARP

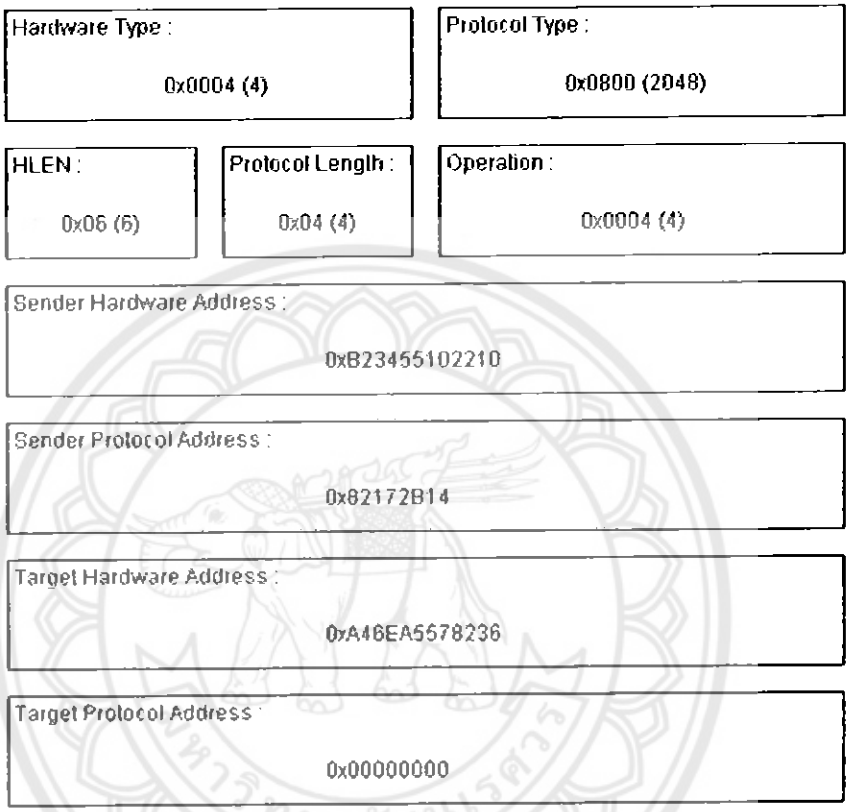
ทดสอบการแสดงผลการทำงานของโพรโทคอล RARP ฝั่ง Sender โดยกำหนดให้ส่ง Message คำว่า "Test" ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.9



รูปที่ 4.9 รายละเอียดของ RARP Request Packet

4.2.9 การทดสอบการรับของโพรโทคอล RARP

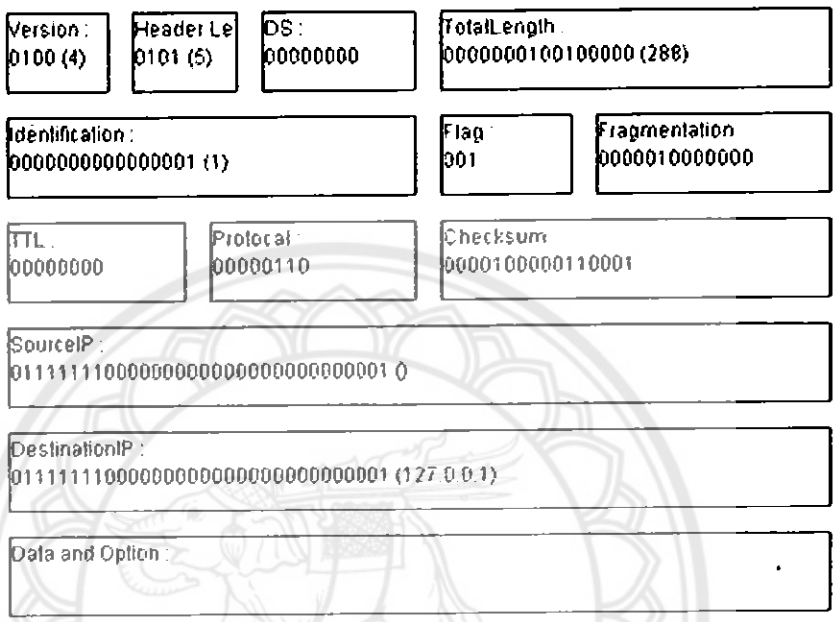
ทดสอบการแสดงผลการทำงานของโพรโทคอล RARP ฝั่ง Receiver โดยจะรับ Message คำว่า "Test" มาจากฝั่ง Sender ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.10



รูปที่ 4.10 รายละเอียดของ RARP Reply Packet

4.2.10 การทดสอบการส่งของโพรโทคอล IPv4

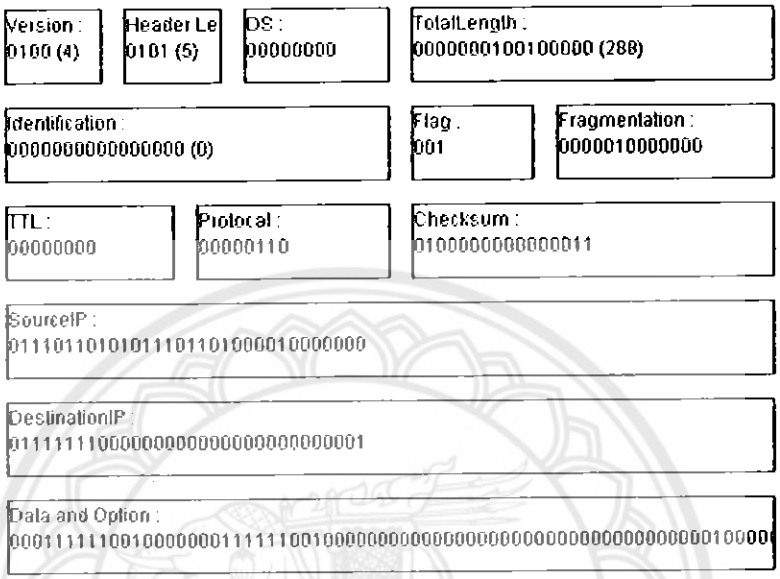
ทดสอบการแสดงผลการทำงานของโพรโทคอล IPv4 ฝั่ง Sender โดยกำหนดให้ส่ง Message คำว่า "Test" ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.11



รูปที่ 4.11 รายละเอียดของโพรโทคอล IPv4 ฝั่ง Sender

4.2.11 การทดสอบการรับของโพรโทคอล IPv4

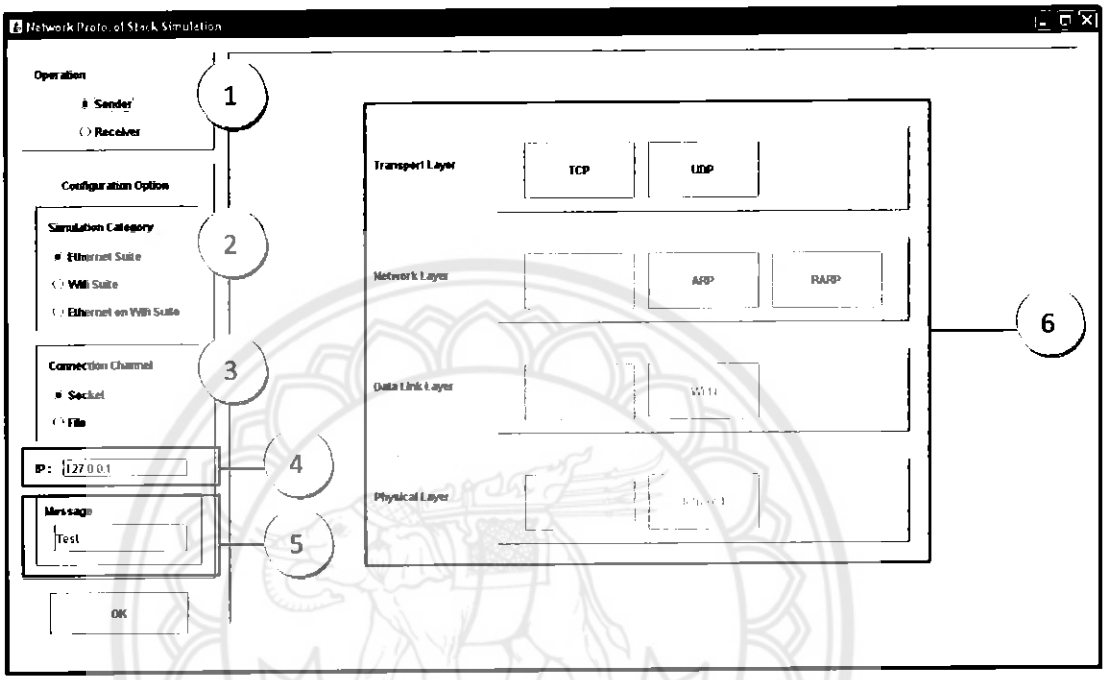
ทดสอบการแสดงผลการทำงานของโพรโทคอล IPv4 ฝั่ง Receiver โดยจะรับ Message คำว่า "Test" มาจากฝั่ง Sender ซึ่งการแสดงผลการทำงานจะปรากฏดังรูปที่ 4.12



รูปที่ 4.12 รายละเอียดของโพรโทคอล IPv4 ฝั่ง Receiver

4.2.12 การแสดงผลของโปรแกรมแบบ GUI

ทดสอบการแสดงผลของหน้าผู้ใช้แบบ Graphic User Interface หรือ GUI ซึ่งจะมีรูปแบบหน้าโปรแกรมเป็นดังที่แสดงในรูปที่ 4.13



รูปที่ 4.13 หน้าโปรแกรมส่วนของ Configuration

1 ส่วนของ Operation ซึ่งเป็นส่วนที่กำหนดว่าเป็นฝั่ง Sender หรือ Receiver

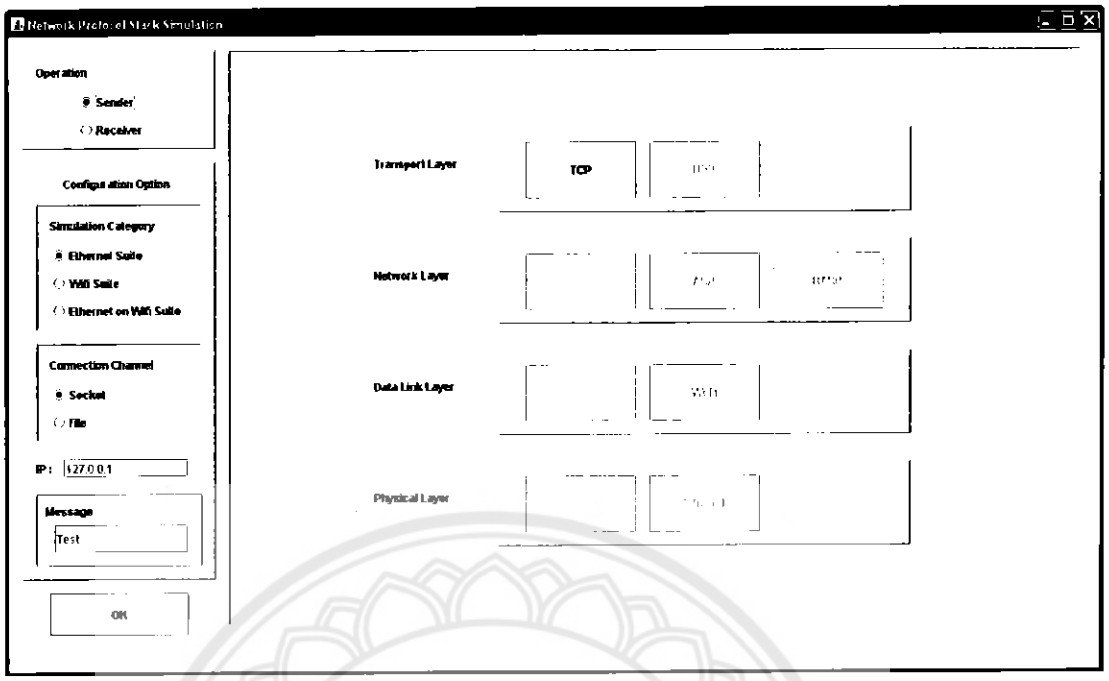
2 ส่วนของ Configuration Option ซึ่งเป็นส่วนที่กำหนดชุดของโพรโทคอล

3 ส่วนของ Connection Channel ซึ่งเป็นส่วนที่กำหนดช่องทางการเชื่อมต่อ

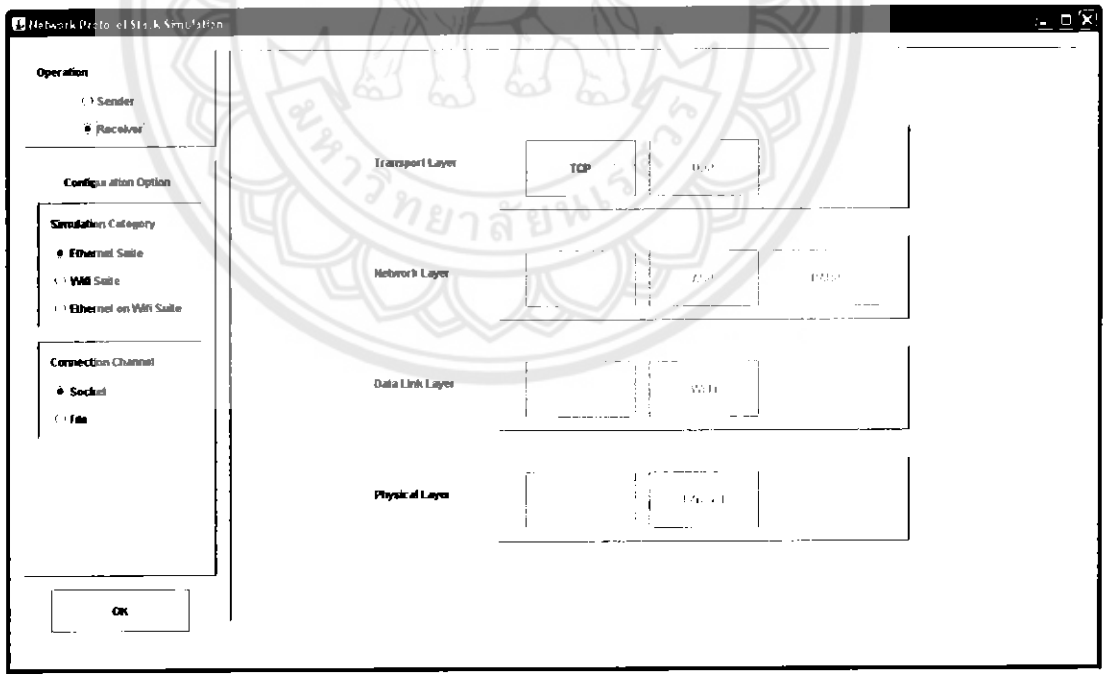
4 ส่วนของ IP Address ซึ่งเป็นส่วนที่กำหนด IP Address ที่ต้องการ

5 ส่วนของ Message ซึ่งเป็นส่วนที่กำหนดข้อความที่ต้องการส่ง

6 ส่วนของโพรโทคอลที่ทำการเลือก



รูปที่ 4.14 หน้าโปรแกรม Configuration แบบ GUI ของฝั่ง Sender โดยในที่นี้เลือกเป็นโพรโทคอล TCP



รูปที่ 4.15 หน้าโปรแกรม Configuration แบบ GUI ของฝั่ง Receiver โดยในที่นี้เลือกเป็นโพรโทคอล TCP

4.3 สรุปการทดสอบโปรแกรม

ตารางที่ 4.3 ตารางสรุปการทดสอบ

Test	รายการ	ผลการทดสอบ
System Test	ทดสอบการแสดงผลของหน้าผู้ใช้แบบ Graphic User Interface (GUI) ทั้งฝั่ง Sender และฝั่ง Receiver	✓
Integration Test	ทดสอบการแสดงผลการทำงานของโพรโทคอล TCP	✓
	ทดสอบการแสดงผลการทำงานของโพรโทคอล UDP	✓
	ทดสอบการแสดงผลการทำงานของโพรโทคอล ARP	✓
	ทดสอบการแสดงผลการทำงานของโพรโทคอล RARP	✓
	ทดสอบการแสดงผลการทำงานของโพรโทคอล IPv4	✓
Unit Test	ทดสอบการสร้าง Header ของโพรโทคอล TCP	✓
	ทดสอบการสร้าง Header ของโพรโทคอล UDP	✓
	ทดสอบการสร้าง Header ของโพรโทคอล ARP	✓
	ทดสอบการสร้าง Header ของโพรโทคอล RARP	✓

บทที่ 5
สรุปผลการดำเนินงานและแนวทางการพัฒนา

โครงงานนี้เป็นการเพิ่มเติมส่วนแสดงผลของไลบรารีสำหรับจำลองการทำงานของเครือข่าย [4, 5] ซึ่งโปรแกรมจะมีการแสดงผลการทำงานแบบ GUI ทำให้สามารถเห็นถึงลำดับการทำงานได้อย่างชัดเจนโดยที่โปรแกรมจะมีการแสดงผลการทำงานแยกเป็นส่วน ๆ ตามลักษณะของโพรโทคอลนั้น ๆ ได้ โปรแกรมสามารถแสดงการทำงานในลักษณะ loop back ซึ่งเป็นการทำงานภายในเครื่องคอมพิวเตอร์เพียงเครื่องเดียวโดยที่เป็นทั้งผู้ส่งและผู้รับในเครื่องเดียวกันและสามารถแสดงการทำงานบนเครื่องคอมพิวเตอร์ 2 เครื่องพร้อม ๆ กันได้ โดยที่เครื่องหนึ่งเป็นฝั่งผู้ส่งและอีกเครื่องหนึ่งเป็นฝั่งผู้รับ ซึ่งโปรแกรมมีการส่งต่อข้อมูลผ่านไฟล์ได้ นอกจากนี้โปรแกรมสามารถแสดงการทำงานของโพรโทคอล UDP ในระดับชั้นที่ 4 และโพรโทคอล ARP และ RARP ในระดับชั้นที่ 3 และในขณะนี้การแสดงผลสามารถรองรับโพรโทคอล TCP และ IPv4 ได้

โครงงานนี้ใช้ภาษา Java ในการพัฒนาโปรแกรมโดยใช้เฉพาะไลบรารีพื้นฐานที่มากับคอมไพเลอร์เนื่องจากเป็นภาษาที่ง่ายต่อการพัฒนาโปรแกรมและการออกแบบหน้าตาของโปรแกรมแบบกราฟิก (Graphic User Interface) เพื่อให้ง่ายต่อการใช้งาน

5.1 ผลการทดลอง

จากการทดลองโปรแกรมจะได้ผลสรุปการทดลองดังที่ปรากฏในตารางที่ 5.1

ตารางที่ 5.1 ตารางสรุปผลการทดลอง

การทดลอง	ผลการทดลอง
การสื่อสารระหว่างโพรโทคอล UDP	✓
การสื่อสารระหว่างโพรโทคอล ARP	✓
การสื่อสารระหว่างโพรโทคอล RARP	✓
การแสดงผลโดยแยกเป็นส่วน ๆ ตามลักษณะของโพรโทคอล	✓

5.2 ปัญหาและอุปสรรค

5.2.1 ปัญหาด้านการอิมพลีเมนท์โพรโทคอล

ตารางที่ 5.2 ปัญหา อุปสรรค และแนวทางแก้ไข ด้านการอิมพลีเมนท์โพร โทคอล

ปัญหาและอุปสรรค	แนวทางแก้ไข
1. ใช้เวลาในการศึกษาระบบการทำงานของไลบรารีชุดนี้ค่อนข้างนาน	1. ศึกษาโดยการทดลองรันโปรแกรมเพื่อดูผลการทำงานและเน้นเฉพาะเจาะจงในส่วนที่เกี่ยวข้องเท่านั้น
2. ต้องใช้เวลาในการศึกษาการทำงานในส่วนที่ต้องการเพิ่มเติมเข้าไปค่อนข้างนาน เนื่องจากข้อมูลที่ต้องการนั้นจะต้องนำมาจากแหล่งความรู้หลายแหล่งมาประกอบกันเพื่อใช้ในการอิมพลีเมนท์โพรโทคอล	2. สืบค้นจากหนังสือที่เกี่ยวข้องในส่วนของโพรโทคอลที่ต้องการเพิ่มเข้าไปโดยตรง เช่น โพรโทคอล UDP เน้นสืบค้นจากหนังสือหรืออินเทอร์เน็ตที่เกี่ยวข้องกับ TCP/IP Protocol เนื่องจาก UDP เป็นโพร โตคอลที่รวมอยู่ใน TCP/IP Protocol Suite

5.2.2 ปัญหาด้านการแสดงผลโดยแยกเป็นส่วน ๆ ตามลักษณะโพรโทคอล

ตารางที่ 5.3 ปัญหา อุปสรรค และแนวทางแก้ไข ด้านการแสดงผลโดยแยกเป็นส่วน ๆ

ปัญหา อุปสรรค	แนวทางแก้ไข
1. เนื่องจากการแสดงผลการทำงานของโปรแกรมเป็นไปอย่างอัตโนมัติทำให้ไม่สามารถย้อนดูการทำงานที่เกิดขึ้นไปแล้วได้	2. ศึกษาทฤษฎีและหลักการทำงานจากแหล่งข้อมูลต่างๆเช่น http://www.oracle.com/technetwork/articles/javase/mvc-136693.html

5.3 ข้อเสนอแนะ

จากปัญหาและอุปสรรคที่พบจากหัวข้อ 5.2 คณะผู้จัดทำจึงมีข้อเสนอแนะสำหรับผู้ที่ต้องการนำไปพัฒนาต่อ ควรจะต้องมีความรู้ในสิ่งต่อไปนี้เพื่อที่จะสามารถนำไปพัฒนาเพิ่มเติมได้ คือ

1. การเขียนโปรแกรมโครงสร้างแบบ Model View Controller ซึ่งสามารถศึกษาหลักการทำงานได้จาก http://www.oracle.com/technetwork/articles/javase/mvc-136693.html

2. ศึกษาการทำงานของเครือข่ายโพรโทคอลสแตก และลำดับการทำงานรวมไปถึงไลบรารีที่จำเป็นในการเรียกใช้งานต่าง ๆ โดยศึกษาจากหนังสือ TCP/IP Protocol Suite ซึ่งเขียนโดย Behrouz A. Forouzan และหนังสือ Data Communications and Networking ซึ่งเขียนโดย Behrouz A. Forouzan เช่นเดียวกัน

3. ศึกษาการสร้าง Graphic User Interface และ Thread ในภาษา Java เพื่อนำไปใช้ในการแสดงผลที่สมบูรณ์แบบยิ่งขึ้น

5.4 สรุปผลการดำเนินงาน

จากการทดลองโปรแกรมสามารถแสดงผลการทำงานแบบ GUI โดยที่โปรแกรมจะมีการแสดงผลการทำงานแยกเป็นส่วน ๆ ตามลักษณะของโพรโทคอลนั้น ๆ ซึ่งจะทำให้สามารถเห็นลำดับของการทำงานในแต่ละโพรโทคอลได้อย่างชัดเจนและสามารถเข้าใจการทำงานได้ง่าย และสามารถแสดงการทำงานในลักษณะ loop back ซึ่งเป็นการทำงานภายในเครื่องคอมพิวเตอร์เพียงเครื่องเดียวโดยที่เป็นทั้งผู้ส่งและผู้รับในเวลาเดียวกันได้ ซึ่งจะทำให้ผู้ใช้สามารถใช้งานโปรแกรมได้แม้มีเครื่องคอมพิวเตอร์เพียงเครื่องเดียว นอกจากนี้โปรแกรมสามารถแสดงการทำงานบนเครื่องคอมพิวเตอร์ 2 เครื่องพร้อม ๆ กันได้เช่นเดียวกัน โดยที่เครื่องหนึ่งเป็นผู้ส่งและอีกเครื่องหนึ่งเป็นผู้รับ โดยโปรแกรมสามารถที่จะมีการส่งต่อข้อมูลผ่านไฟล์ได้ และโปรแกรมสามารถที่จะแสดงผลการทำงานของโพรโทคอล UDP ในระดับชั้นที่ 4 และโพรโทคอล ARP และ RARP ในระดับชั้นที่ 3 ได้ซึ่งจะทำให้มีความเข้าใจและเห็นการทำงานสำหรับเครือข่ายโพรโทคอลสแตกได้ชัดเจนและหลากหลายมากยิ่งขึ้น และในขณะนี้การแสดงผลสามารถรองรับการแสดงผลการทำงานของโพรโทคอล TCP และ IPv4 ได้อีกด้วย

เอกสารอ้างอิง

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

[2] Behrouz A.Forouzan, **TCP/IP Protocol Suite**, 3rd ed. . Singapore: McGraw-Hill, 2007.

[3] James F. Kurose and Keith W. Ross, Computer Networking, Addison Wesley, 2001.

[4] ชัยภัทร จารุชาติ และนทภพ เริงนภาพล, โปรแกรมจำลองการทำงานสำหรับเน็ตเวิร์คโพรโทคอล TCP/IP เลเยอร์ 1 และเลเยอร์ 2, มหาวิทยาลัยนเรศวร, พิษณุโลก, ปริญญานิพนธ์ 2008.

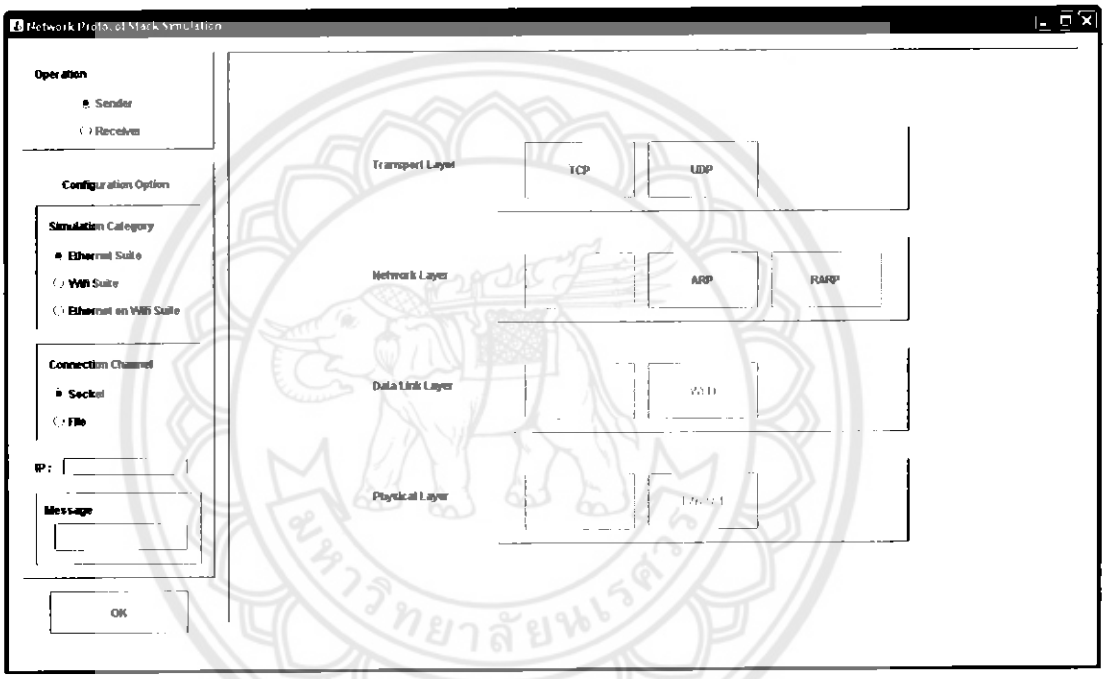
[5] กิตติศักดิ์ ตุนาโป่ง และณัฐเสฏฐ์ วัชรานุรักษ์, ไลบรารีสำหรับจำลองเน็ตเวิร์คโพรโทคอลสแตก, มหาวิทยาลัยนเรศวร, พิษณุโลก, ปริญญานิพนธ์ 2007.

ภาคผนวก

วิธีการใช้โปรแกรม

ก1. วิธีการใช้โปรแกรม

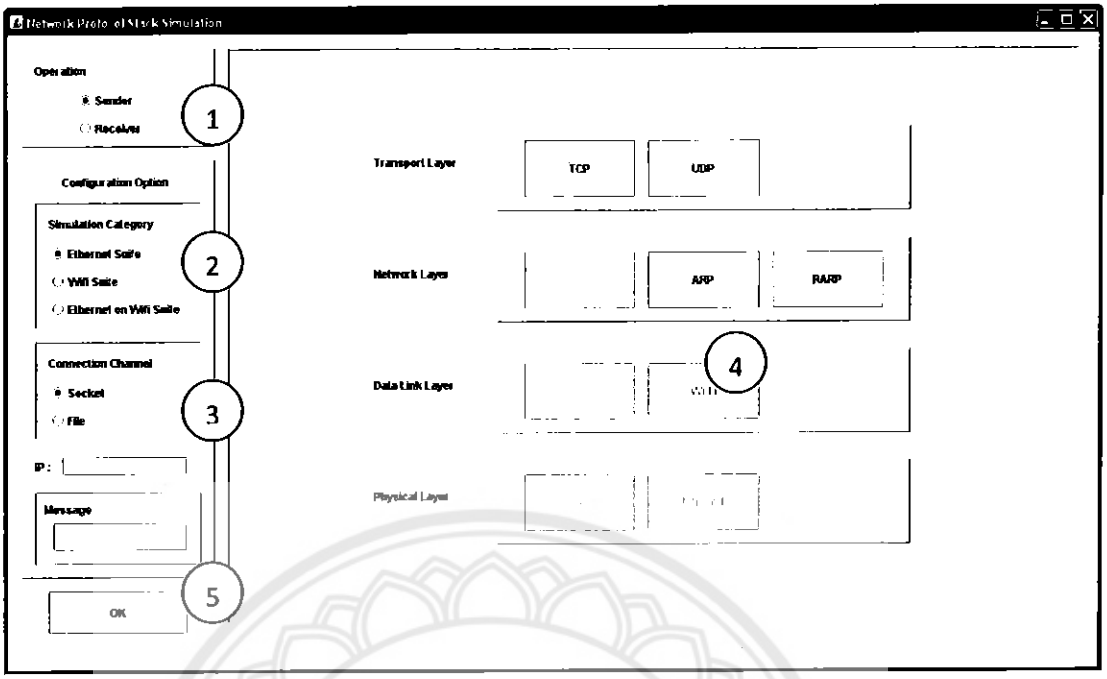
เมื่อผู้ใช้ทำการรันโปรแกรมเป็นที่เรียบร้อยแล้ว โปรแกรมจะปรากฏหน้าต่าง ดังรูปที่ ก.1



รูปที่ ก.1 แสดงหน้าเริ่มต้นของโปรแกรม

ในหน้าโปรแกรมนี้จะเป็นส่วนของ Configuration ซึ่งจะเป็นส่วนที่ให้ผู้ใช้ทำการกำหนดค่าต่าง ๆ ให้กับโปรแกรม ผู้ใช้สามารถเลือก Operation เป็น "Receiver" เนื่องจากในการทำงานของระบบเครือข่ายโพรโทคอลสแตก นั้นฝั่งรับจะรอรับข้อมูลที่จะถูกส่งมาจากฝั่ง Sender

การเลือกดูการทำงานของโปรแกรมนั้นจะสามารถดูรายละเอียดของโพรโตคอลได้เพียงชั้น 3 และ 4 นั่นก็คือ โพรโตคอล TCP UDP ARP RARP และ IPV4 โดยที่ IPV4 จะถูกตั้งค่าให้เลือกโดยอัตโนมัติเมื่อต้องการดูการทำงานของ TCP UDP และที่ระดับชั้นเลเยอร์ 1 และ 2 จะถูกเลือกโดยการกำหนดชุดโพรโตคอลในส่วนที่เป็น Simulation Category



รูปที่ ก.2 แสดงส่วนประกอบต่างๆ ของโปรแกรม

การตั้งค่าจะเลือกตามลำดับหมายเลขดังรูปโดยที่แต่ละหมายเลขที่ได้กำกับไว้อธิบายได้ดังนี้

1. ในส่วนของ Operation จะเป็นส่วนที่ให้เลือกว่าเป็น Sender หรือ Receiver
2. ในส่วนของ Simulation Category จะเป็นส่วนของการเลือกชุดโพรโตคอลในระดับเลเยอร์ 1 และ 2 โดยสามารถอธิบายได้ดังนี้
    a. Ethernet Suite จะเป็นการเลือกโพรโตคอลในเลเยอร์ 1 กับ 2 เป็น CAT5 และ Ethernet ตามลำดับ
    b. Wifi Suite จะเป็นการเลือกโพรโตคอลในเลเยอร์ 1 กับ 2 เป็น Infrared และ Wifi ตามลำดับ
    c. Ethernet on Wifi Suite จะเป็นการเลือกโพรโตคอลในเลเยอร์ 1 กับ 2 เป็น Infrared และ Ethernet ตามลำดับ
3. ในส่วนของ Connection Channel จะเป็นส่วนที่ทางผู้รับกับผู้ส่งจะทำการเลือกเส้นทางที่ใช้การติดต่อกันโดยจะมีอยู่ 2 ช่องทางคือ Socket และ File โดยที่จะอธิบายในแต่ละส่วนได้ดังนี้
    a. Socket จะเป็นการติดต่อผ่าน Network จริงโดยใช้ไลบรารีของทาง Java ในการติดต่อโดยจะต้องระบุ IP Address ปลายทางเป็นของฝั่ง Receiver

b. File จะเป็นการติดต่อสื่อสารกันระหว่าง Sender และ Receiver โดยผ่านทาง File System โดยเมื่อเลือกตัวเลือกนี้จะสามารถติดต่อได้เพียงในเครื่องตัวเองเท่านั้น

4. ในส่วนนี้จะเป็นส่วนที่เลือกดูการทำงานของโพรโตคอล หมายความว่าถ้าต้องการเลือกดูโพรโตคอลใดบ้างใน Network ก็จะต้องมาตั้งค่าในส่วนนี้
5. เป็นส่วนของ Message ที่ต้องการส่ง จะปรากฏในกรณีที่เป็นฝั่งผู้ส่งเท่านั้น

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวมัณฑนา ลิมาภิรักษ์

ภูมิลำเนา 441 ถนนอ่อนนุช ซอยอ่อนนุช 66 ซอย 6
เขตสวนหลวง กรุงเทพมหานคร 10250

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนนครสวรรค์ จังหวัดนครสวรรค์
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail: munthana.limapirak@gmail.com,
nalika_hanada@hotmail.com

ชื่อ นายสราวุธ สุขใจ

ภูมิลำเนา 196/4 หมู่ 7 ต.บางระกำ อ.บางระกำ
จังหวัดพิษณุโลก 65140

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนพิษณุโลกพิทยาคม จังหวัดพิษณุโลก
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail: zealousV_1@hotmail.com