

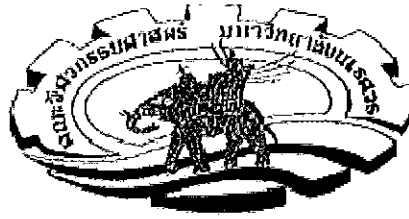
โปรแกรมควบคุมการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์  
SOFTWARE FOR TEACHING AND LEARNING CONTROL IN COMPUTER  
LAB

นายจักรวิทย์	กิจเจริญ	รหัส 51364699
นายประจต	จตุพันธ์	รหัส 51364835
นายอนุพงศ์	หอมรินทร์	รหัส 51365061

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2554
ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 20 ก.ค. 2558
เลขทะเบียน..... 1686 2329
เลขเรียกหนังสือ..... 45
มหาวิทยาลัยนเรศวร จ. 239

๒ 2554



## ใบรับรองปริญญาโท

ชื่อหัวข้อโครงการ โปรแกรมควบคุมการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์  
ผู้ดำเนินโครงการ นายจักรวิทย์ กิจเจริญ รหัส 51364699  
นายประณต จุลพันธ์ รหัส 51364835  
นายอนุพงศ์ หอมรัตน์ รหัส 51365061

ที่ปรึกษาโครงการ อาจารย์สิรภพ คชรัตน์  
สาขาวิชา วิศวกรรมคอมพิวเตอร์  
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
ปีการศึกษา 2554

.....  
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญาโทฉบับนี้เป็นส่วนหนึ่ง  
ของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

.....  
.....ประธานกรรมการ  
(ดร.สุรเดช จิตประไพกุลศาล)

.....  
.....กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ธิยะมงคล)

.....  
.....กรรมการ  
(ดร.วรลักษณ์ คงเด่นฟ้า)

.....  
.....กรรมการ  
(อาจารย์สิรภพ คชรัตน์)

ชื่อหัวข้อโครงการ	โปรแกรมควบคุมการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์	
ผู้ดำเนินโครงการ	นายจักราวิทย์ กิจเจริญ	รหัส 51364699
	นายประณต จุลพันธ์	รหัส 51364835
	นายอนุพงศ์ หอมรัตน์	รหัส 51365061
ที่ปรึกษาโครงการ	อ.สิรภพ ชรรัตน์	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2554	

#### บทคัดย่อ

โครงการนี้เป็นการพัฒนาโปรแกรม เพื่อใช้อำนวยความสะดวกและควบคุมการใช้งานคอมพิวเตอร์ในห้องปฏิบัติการคอมพิวเตอร์ โดยมีการพัฒนาโปรแกรมออกเป็นสองส่วน คือ ส่วนของโปรแกรมฝั่งแม่ข่าย ที่ใช้ในการเข้าถึง โปรแกรมเครื่องลูกข่ายที่อยู่ภายในเครือข่ายเดียวกัน และโปรแกรมฝั่งลูกข่าย ที่ตอบสนองต่อการทำงานของเครื่องแม่ข่าย

วัตถุประสงค์ของโปรแกรมที่พัฒนาขึ้น จะแบ่งออกเป็นสองส่วน คือ ส่วนแรกโปรแกรมสามารถอำนวยความสะดวกให้กับผู้เรียนและผู้สอนในห้องปฏิบัติการคอมพิวเตอร์ เพื่อให้การเรียนการสอนมีความสะดวกสบายยิ่งขึ้น เช่น การสนทนาและการรับส่งไฟล์ข้อมูลระหว่างผู้เรียนกับผู้สอนรวมทั้งการดูหน้าจอของเครื่องลูกข่าย และส่วนที่สองโปรแกรมสามารถควบคุมและจำกัดสิทธิ์การใช้งานของเครื่องลูกข่ายในห้องปฏิบัติการคอมพิวเตอร์ เช่น การจำกัดการเข้าถึงเว็บไซต์ การล็อกหน้าจอของเครื่องลูกข่าย เพื่อให้ผู้สอนสามารถควบคุมการใช้งานคอมพิวเตอร์ของผู้เรียนให้ตรงเป้าหมายการเรียนรู้มากที่สุดและนำไปสู่การเรียนการสอนที่มีประสิทธิภาพยิ่งขึ้น

**Project title** Software for Teaching and Learning Control in Computer Lab

**Name** Mr. Jakravit Kitcharoen ID. 51364699

Mr. Pranot Junlaphun ID. 51364835

Mr. Anupong Homruen ID. 51365061

**Project advisor** Mr. Sirapop Khotcharrat

**Major** Computer Engineering

---

**Department** Electrical and Computer Engineering

**Academic year** 2011

.....

**Abstract**

This project is developed to facilitate and control the use of computers in computer labs. The program has two parts : the server-side application used to access clients within the same network, and client-side applications that response to the server.

There are two main objectives of the program. The first objective is to facilitate the learning and teaching in computer labs, so the teaching is more comfortable, such as chatting, files transferring and screen monitoring. The second objective is to control and restrict the use of computers in the lab, such as blocking websites or locking screens of client computers. Using this program in a computer lab can help an instructor to control their students' activities in order to meet the specific learning goals, and can make the instruction more effective.



## กิตติกรรมประกาศ

โครงการวิศวกรรมคอมพิวเตอร์เล่มนี้สำเร็จลุล่วงมาได้ด้วยดีนั้น เนื่องจากอาจารย์ที่ปรึกษา คือ อาจารย์ศิริภพ กษรัตน์ ที่คอยให้คำปรึกษา คำแนะนำ และวิธีการแก้ปัญหาที่เกิดขึ้นตลอดระยะเวลาการทำโครงการ คณะผู้จัดทำโครงการจึงขอกราบขอบพระคุณอาจารย์ที่ปรึกษาเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณอาจารย์ซึ่งเป็นคณะกรรมการทั้ง 3 ท่าน ได้แก่ ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ธิยะมงคล ดร.สุรเดช จิตประไพกุลศาส และดร.วรลักษณ์ กงเด่นฟ้า ที่ช่วยแนะแนวทางการแก้ไขโครงการนี้ให้มีความสำเร็จลุล่วงไปด้วยดี

และสุดท้ายนี้ขอกราบขอบพระคุณ บิดา มารดา ที่คอยให้กำลังใจและคอยให้การสนับสนุนในทุกด้านจนทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี



นายจักราวิทย์ กิจเจริญ  
นายประณต จุลพันธ์  
นายอนุพงศ์ หอมริน

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
<hr/>	
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบข่ายของโครงการ.....	2
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 แผนการดำเนินงาน.....	3
1.6 ผลที่คาดว่าจะได้รับ.....	4
1.7 งบประมาณของโครงการ.....	4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ความรู้พื้นฐานในระบบเครือข่าย.....	5
2.2 สถาปัตยกรรมเครือข่าย.....	5
2.3 โปรโตคอล.....	9
2.4 โทโปโลยี.....	15
2.5 ประเภทของระบบเครือข่าย LAN แบ่งตามลักษณะการทำงาน.....	19
2.6 ความรู้เกี่ยวกับ Window API.....	21

## สารบัญ (ต่อ)

	หน้า
2.7 ความรู้เกี่ยวกับ Socket.....	22
2.8 ความรู้เกี่ยวกับพอร์ท.....	27
<b>บทที่ 3 วิธีการดำเนินงาน.....</b>	<b>32</b>
3.1 แนวคิดในการออกแบบ.....	32
3.2 ความสามารถของโปรแกรม.....	33
3.3 การออกแบบส่วนติดต่อกับผู้ใช้(Graphic User Interface).....	34
3.4 การออกแบบโปรแกรมด้วยภาษา UML(Unified Modeling Language).....	38
3.4.1 การออกแบบด้วย Use case Diagram.....	38
3.4.2 การออกแบบด้วย Activity Diagram ในภาพรวม.....	39
3.4.3 การออกแบบด้วย Activity Diagram ในแต่ละงาน.....	42
3.5 การออกแบบพอร์ตการใช้งาน.....	63
<b>บทที่ 4 การทดสอบและผลการทดสอบ.....</b>	<b>64</b>
4.1 แผนการทดสอบโปรแกรม.....	64
4.2 การทดสอบ โปรแกรม.....	66
4.2.1 การทดสอบรอร์รับที่อยู่ไอพีของเครื่องลูกข่าย.....	66
4.2.2 การทดสอบสนทนาระหว่างเครื่องแม่ข่ายกับลูกข่าย.....	66
4.2.3 การทดสอบสนทนาแบบห้องสนทนา.....	67
4.2.4 การทดสอบรับไฟล์แบบเลือกเครื่อง.....	68
4.2.5 การทดสอบส่งไฟล์แบบกระจาย.....	69
4.2.6 การทดสอบการส่งข้อความเตือน.....	70

## สารบัญ (ต่อ)

	หน้า
4.2.7 การทดสอบ Turn off , Restart , Log off, Lock work station .....	71
4.2.8 การทดสอบการจำกัดการเข้าเว็บไซต์.....	74
4.2.9 การทดสอบการดูหน้าจอเครื่องลูกข่ายแบบเลือกเครื่อง.....	75
4.2.10 การทดสอบการดูหน้าจอเครื่องลูกข่ายแบบหลายเครื่อง.....	76
4.2.11 การทดสอบการดูข้อมูลเบื้องต้นของเครื่องลูกข่าย.....	77
<b>บทที่ 5 สรุปผลการดำเนินงานและแนวทางการพัฒนา.....</b>	<b>79</b>
5.1 ผลการทดลอง.....	79
5.2 ปัญหาและอุปสรรค.....	80
5.3 ข้อจำกัดของโปรแกรม.....	81
5.4 ข้อเสนอแนะ.....	81
5.5 แนวทางการพัฒนาต่อ.....	82
5.6 สรุปการดำเนินงาน.....	82
เอกสารอ้างอิง.....	83

## สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน.....	3
ตารางที่ 2.1 โปรโตคอลของในลำดับชั้นแต่ละชั้น.....	8
ตารางที่ 2.2 หมายเลขพอร์ต.....	28
ตารางที่ 3.1 การใช้งานพอร์ตของโปรแกรม.....	63
ตารางที่ 4.1 แผนการทดสอบโปรแกรม.....	64
ตารางที่ 4.2 ผลการทดลองการรับส่งไฟล์.....	69
ตารางที่ 5.2 ปัญหาและอุปสรรค.....	80



## สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 การทำงานลำดับชั้นของ OSI Model.....	6
รูปที่ 2.2 การส่งแพ็กเก็ตจอสวิตช์ใน OSI Model.....	7
รูปที่ 2.3 ส่วนดำเนินการโดยผู้ใช้และส่วนดำเนินการ โดยเครือข่าย.....	7
รูปที่ 2.4 โครงสร้าง TCP/IP.....	9
รูปที่ 2.5 IP Header.....	11
รูปที่ 2.6 ICMP Header.....	13
รูปที่ 2.7 TCP Header.....	13
รูปที่ 2.8 UDP Header.....	14
รูปที่ 2.9 การทำงานของ FTP.....	15
รูปที่ 2.10 โทโปโลยี.....	16
รูปที่ 2.11 โทโปโลยี แบบวงแหวน.....	17
รูปที่ 2.12 โทโปโลยี แบบดาว.....	18
รูปที่ 2.13 ลักษณะการทำงานแบบ หนึ่งต่อหนึ่ง.....	19
รูปที่ 2.14 การติดต่อแบบ ลูกข่าย-แม่ข่าย.....	20
รูปที่ 2.15 ความสัมพันธ์ ระหว่าง Application , Socket และ โปรโตคอล.....	23
รูปที่ 2.16 ขั้นตอนการทำงานของ Stream Socket.....	24
รูปที่ 2.17 ขั้นตอนการทำงานของ Datagram Socket.....	25
รูปที่ 2.18 การใช้งานหมายเลขพอร์ต.....	29
รูปที่ 2.19 ลักษณะของ Hosts file .....	30
รูปที่ 2.20 ลักษณะของเว็บเบราว์เซอร์เข้าเว็บไม่ได้.....	31
รูปที่ 3.1 หน้าตาโปรแกรมเครื่องแม่ข่ายในส่วนติดต่อผู้ใช้.....	34
รูปที่ 3.2 หน้าจอการคลิกขวาที่ไอคอนคอมพิวเตอร์เครื่องลูกข่าย.....	36

## สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 3.3 หน้าตาโปรแกรมเครื่องลูกข่ายในส่วนติดต่อผู้ใช้.....	37
รูปที่ 3.4 Use Case Diagram ทั้งหมดของระบบ.....	38
รูปที่ 3.5 Activity Diagram ของระบบที่ไม่ต้องรอให้ลูกข่ายตอบรับการทำงาน.....	39
รูปที่ 3.6 Activity Diagram ของระบบที่ต้องรอให้ลูกข่ายตอบรับการทำงาน.....	40
รูปที่ 3.7 Activity Diagram ของการรอรับไอพีของเครื่องลูกข่าย.....	41
รูปที่ 3.8 Activity Diagram ของการรอรับไอพีของเครื่องลูกข่าย.....	42
รูปที่ 3.9 Activity Diagram การสนทนาแบบเลือกเครื่องของเครื่องแม่ข่าย.....	43
รูปที่ 3.10 Activity Diagram การสนทนาแบบเลือกเครื่องของเครื่องลูกข่าย.....	44
รูปที่ 3.11 Activity Diagram ของการสนทนาแบบห้องสนทนาของเครื่องแม่ข่าย.....	45
รูปที่ 3.12 Activity Diagram ของการสนทนาแบบห้องสนทนาของเครื่องลูกข่าย.....	46
รูปที่ 3.13 Activity Diagram การส่งไฟล์แบบเลือกเครื่องของเครื่องแม่ข่าย.....	47
รูปที่ 3.14 Activity Diagram การส่งไฟล์แบบเลือกเครื่องของเครื่องลูกข่าย.....	48
รูปที่ 3.15 Activity Diagram Activity Diagram ของการส่งไฟล์แบบกระจายของเครื่องแม่ข่าย....	49
รูปที่ 3.16 Activity Diagram ของการส่งไฟล์แบบกระจายของเครื่องลูกข่าย.....	50
รูปที่ 3.17 Activity Diagram ของการส่งข้อความเตือนของเครื่องแม่ข่าย.....	51
รูปที่ 3.18 Activity Diagram ของการส่งข้อความเตือนของเครื่องลูกข่าย.....	52
รูปที่ 3.19 Activity Diagram ของการควบคุมเครื่องในส่วนของเครื่องแม่ข่าย.....	53
รูปที่ 3.20 Activity Diagram ของการควบคุมเครื่องในส่วนของเครื่องลูกข่าย.....	54
รูปที่ 3.21 Activity Diagram ของการจำกัดการเข้าถึงเว็บไซต์ในส่วนของเครื่องแม่ข่าย.....	55
รูปที่ 3.22 Activity Diagram ของการจำกัดการเข้าถึงเว็บไซต์ในส่วนของ เครื่องลูกข่าย.....	57
รูปที่ 3.23 Activity Diagram การดูหน้าจอแบบจอเดียว (ดูจอใหญ่) ฝั่ง เครื่องแม่ข่าย.....	58
รูปที่ 3.24 Activity Diagram การดูหน้าจอแบบหลายจอ ฝั่ง เครื่องแม่ข่าย.....	59

## สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 3.25 Activity Diagram Activity Diagram การดูหน้าจอ ฟังก์ชันเครื่องลูกข่าย.....	61
รูปที่ 4.1 หน้าจอการทดสอบโปรแกรมฟังก์ชันเครื่องแม่ข่าย.....	65
รูปที่ 4.2 หน้าจอการทดสอบโปรแกรมฟังก์ชันเครื่องลูกข่าย.....	65
รูปที่ 4.3 การรองรับ ไอพีของเครื่องลูกข่าย.....	66
รูปที่ 4.4 การสนทนาเมื่อมีเครื่องแม่ข่ายติดต่อเข้ามา.....	67
รูปที่ 4.5 การเข้าห้องสนทนาของเครื่องแม่ข่ายและเครื่องลูกข่ายหลายๆ เครื่อง.....	68
รูปที่ 4.6 การรับส่งไฟล์แบบเลือกเครื่อง.....	68
รูปที่ 4.7 การส่งไฟล์แบบกระจาย.....	70
รูปที่ 4.8 หน้าจอทดสอบการเตือน.....	70
รูปที่ 4.9 หน้าจอการปิดเครื่อง รีสตาร์ท ออกจากระบบปฏิบัติการ หยุดการทำงาน 1.....	71
รูปที่ 4.10 หน้าจอการปิดเครื่อง รีสตาร์ท ออกจากระบบปฏิบัติการ หยุดการทำงาน 2.....	72
รูปที่ 4.11 หน้าจอเครื่องลูกข่ายถูกสั่งปิด.....	72
รูปที่ 4.12 หน้าจอเครื่องลูกข่ายถูกสั่งปิดแล้วเปิดเครื่องใหม่.....	73
รูปที่ 4.13 หน้าจอเครื่องลูกข่ายถูกสั่งล็อกออฟ (Log off) .....	73
รูปที่ 4.14 หน้าจอเครื่องลูกข่ายถูกสั่งล็อกหน้าจอ (Lock Screen).....	74
รูปที่ 4.15 หน้าจอการจำกัดการเข้าถึงเว็บไซต์ของเครื่องลูกข่าย.....	75
รูปที่ 4.16 หน้าจอการจำกัดการเข้าถึงเว็บไซต์ของเครื่องลูกข่ายแบบเปลี่ยนที่อยู่ไอพี.....	76
รูปที่ 4.17 การทดลองดูหน้าจอเครื่องลูกข่าย.....	76
รูปที่ 4.18 การทดลองดูหน้าจอหลายๆ เครื่องของเครื่องลูกข่าย.....	77
รูปที่ 4.19 การทดลองดูข้อมูลเบื้องต้นของเครื่องลูกข่าย.....	78



## บทที่ 1

### บทนำ

#### 1.1 ที่มาและความสำคัญ

การเรียนการสอนภายในห้องปฏิบัติการคอมพิวเตอร์ในปัจจุบันได้รับความนิยมอย่าง

แพร่หลาย เพราะช่วยอำนวยความสะดวกในการเรียนการสอนได้เป็นอย่างดี แต่การเรียนการสอนแบบนี้ผู้สอนกับผู้เรียนมีลักษณะมีอิสระต่อกันผู้สอนไม่สามารถทราบพฤติกรรมการใช้งานคอมพิวเตอร์ของผู้เรียนได้ ทำให้ผู้เรียนมีโอกาสที่จะใช้งานคอมพิวเตอร์ผิดเป้าหมาย และปัญหาที่พบได้บ่อยคือผู้เรียนไม่สามารถสื่อสารกับผู้สอนเมื่อมีข้อสงสัย อันเนื่องมาจากสภาพแวดล้อมอันเป็นอุปสรรคในบริเวณห้องปฏิบัติการ จากปัญหาเหล่านี้จึงทำให้การเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์ไม่ได้ประสิทธิภาพเท่าที่ควร

ปัจจุบันมีผู้ให้ความสนใจปัญหานี้เป็นอย่างมากและมีการพัฒนาเครื่องมืออย่างต่อเนื่องแต่อาจจะยังไม่ตรงกับความต้องการที่เฉพาะเจาะจงของลักษณะการเรียนการสอนของผู้สอน ผู้จัดทำโครงการนี้จึงเห็นความสำคัญของปัญหานี้และศึกษาต่อแนวทางของ โปรแกรมที่ใช้ควบคุมการเรียนการสอน เช่น โปรแกรมดูแลและควบคุมเครื่องลูกข่าย เป็นต้น เพื่อพัฒนาเครื่องมือที่ช่วยในการเรียนการสอนของผู้สอนและเพื่อเป็นการ อำนวยความสะดวกในด้านการควบคุม เช่น การจำกัดการใช้งานเว็บไซต์ การล็อกหน้าจอเครื่องลูกข่ายและการดูแลพฤติกรรมของผู้เรียน เช่น การดูหน้าจอเครื่องลูกข่าย เพื่อให้การเรียนการสอนมีประสิทธิภาพมากยิ่งขึ้น

#### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อพัฒนาและศึกษาหลักการทำงานของระบบช่วยการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์
2. เพื่อให้ผู้สอนสามารถควบคุมการใช้งานคอมพิวเตอร์ที่ไม่เหมาะสมของผู้เรียนได้
3. เพื่ออำนวยความสะดวกแก่ผู้สอนและผู้เรียนสำหรับการเรียนการสอนภายในห้องปฏิบัติการคอมพิวเตอร์

### 1.3 ขอบข่ายของโครงการงาน

1. ประยุกต์ใช้ภาษาซีชาร์ป (C#) เพื่อพัฒนาระบบช่วยการเรียนรู้การสอนผ่านระบบเครือข่าย  
แลน (Local Network Area)
2. โปรแกรมสามารถช่วยอำนวยความสะดวกการเรียนรู้การสอนในห้องปฏิบัติการ  
คอมพิวเตอร์ ดังนี้
  - 2.1 เครื่องแม่ข่ายสามารถสนทนากับเครื่องลูกข่ายที่เลือก และเครื่องลูกข่ายสามารถคุยกัน  
ในลักษณะห้องสนทนา รวมทั้งเครื่องแม่ข่ายสามารถแจ้งเตือนด้วยข้อความไปยัง  
เครื่องลูกข่ายได้
  - 2.2 เครื่องแม่ข่ายสามารถส่งไฟล์ไปยังเครื่องลูกข่ายที่เลือก และเครื่องลูกข่ายสามารถส่ง  
ไฟล์กลับมาให้แม่ข่ายได้ รวมทั้งเครื่องแม่ข่ายสามารถส่งไฟล์แบบกระจายไปหา  
เครื่องลูกข่ายได้
  - 2.3 เครื่องแม่ข่ายสามารถเลือกดูหน้าจอของเครื่องลูกข่าย และดูข้อมูลเบื้องต้นของเครื่อง  
ลูกข่ายนั้นได้
  - 2.4 เครื่องแม่ข่ายสามารถทราบที่อยู่ไอพี (IP Address) ของเครื่องลูกข่ายทุกเครื่องที่ติดต่อ  
เข้ามาได้
3. เครื่องแม่ข่ายสามารถควบคุมการทำงานของลูกข่ายที่อยู่ในห้องปฏิบัติการคอมพิวเตอร์ได้
  - 3.1 สามารถ จำกัดการเข้าถึงเว็บไซต์ เครื่องลูกข่ายได้ และสามารถจดจำ เครื่อง และ  
เว็บไซต์ที่เคยจำกัดได้
  - 3.2 สามารถ ปิดเครื่อง , ปิดแล้วเปิดเครื่องใหม่ , ล็อกหน้าจอเครื่องลูกข่ายได้

### 1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาทฤษฎีและหลักการที่เกี่ยวข้องกับ โครงการงาน
  - ศึกษาการเขียน network Programing
  - ศึกษาการเขียน System Programing
  - ศึกษารูปแบบการเขียน โปรแกรมภาษา C#



## 1.6 ผลที่คาดว่าจะได้รับ

1. ได้ความรู้ในหลักการสร้างระบบช่วยการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์
2. ผู้สอนสามารถควบคุมการใช้งานคอมพิวเตอร์ของผู้เรียนได้อย่างมีประสิทธิภาพ
3. ได้โปรแกรมที่สามารถอำนวยความสะดวกการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์

## 1.7 งบประมาณของโครงการ

- |                       |                           |
|-----------------------|---------------------------|
| 1. ค่าถ่ายเอกสาร      | 600 บาท                   |
| 2. ค่ากระดาษ          | 600 บาท                   |
| 3. ค่าหมึกพิมพ์       | 800 บาท                   |
| 4. ค่าเช่าเล่มโครงการ | 1,000 บาท                 |
| รวมเป็นเงิน           | 3,000 บาท (สามพันบาทถ้วน) |

หมายเหตุ ขอถัวเฉลี่ยทุกรายการ



## บทที่ 2

### หลักการและทฤษฎีที่เกี่ยวข้อง

การใช้เครือข่ายคอมพิวเตอร์จำเป็นต้องมีเครื่องคอมพิวเตอร์อย่างน้อย เครื่องข่ายคอมพิวเตอร์มีหลายแบบ แต่สำหรับโครงงานนี้ ใช้เครือข่ายแบบแลน ซึ่งมีเครื่องคอมพิวเตอร์ สองเครื่องขึ้นไป เชื่อมต่อโดยใช้ สายยูทีพี (Unshielded Twisted Pair) เข้าหัวแบบ อาร์เจ 45 เชื่อมต่อกันในโทโปโลยี ที่หลากหลายขึ้นอยู่กับความเหมาะสมในการนำไปใช้แต่ละสภาพแวดล้อม การนำเครือข่ายคอมพิวเตอร์มาใช้ทำให้การดำเนินงานต่างๆ มีความสะดวกยิ่งขึ้นและมีการลดต้นทุนอย่างมากไม่ว่าจะในเชิงธุรกิจหรือการศึกษา เพราะมันสามารถที่จะใช้ทรัพยากรที่มีอยู่อย่างจำกัด และยังช่วยส่งข้อมูลระหว่างกันได้อย่างรวดเร็ว

#### 2.1 ความรู้พื้นฐานในระบบเครือข่าย

ก่อนที่เราจะใช้งานระบบเครือข่ายเราต้องมีความเข้าใจเครือข่ายที่เราใช้งาน และ รู้ที่มาที่ไปของระบบเครือข่าย และทราบการทำงานว่าระบบเครือข่ายทำงานอย่างไร เพื่อให้การทำงานเป็นไปได้อย่างมีประสิทธิภาพ

โดยความรู้พื้นฐานในระบบเครือข่ายมีดังต่อไปนี้

1. ความรู้ด้านสถาปัตยกรรมเครือข่าย
2. ความรู้ของ โพรโทคอลที่ใช้ในระบบเครือข่าย
3. โทโปโลยี ที่ใช้ในระบบเครือข่าย
4. ลักษณะการทำงานของเครือข่าย
5. อุปกรณ์ที่ใช้ในระบบเครือข่าย

#### 2.2 สถาปัตยกรรมเครือข่าย OSI (OSI Model)

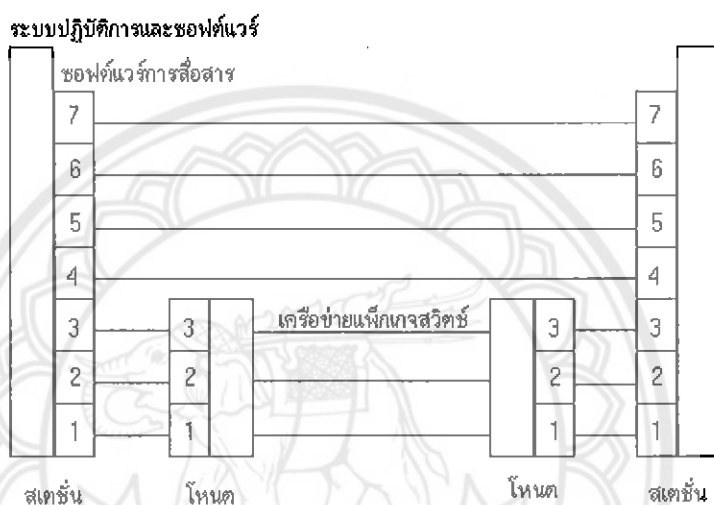
องค์กร ISO (international organization for Standard) ได้กำหนดรูปแบบสถาปัตยกรรมเครือข่ายซึ่งก็คือ "รูปแบบ OSI " (Open System Interconnection Model) เพื่อใช้เป็นรูปแบบมาตรฐานในการเชื่อมต่อระบบ คอมพิวเตอร์ เพื่อเป็นมาตรฐาน โดย OSI มีการ แบ่งโครงสร้างออกเป็น 7 ลำดับชั้น และในแต่ละ ลำดับชั้น ได้มีการกำหนดหน้าที่การทำงานไว้ ดังต่อไปนี้

7 Application	โปรโตคอล Application	7 Application
6 Presentation	โปรโตคอล Presentation	6 Presentation
5 Session	โปรโตคอล Session	5 Session
4 Transport	โปรโตคอล Transport	4 Transport
3 Network	โปรโตคอล Network	3 Network
2 Data Link	โปรโตคอล Data Link	2 Data Link
1 Physical	สายสื่อสาร	1 Physical

รูปที่ 2.1 การทำงานลำดับชั้นของ OSI Model

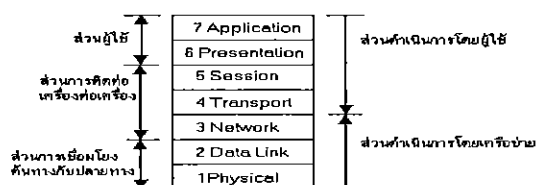
1. Physical layer เป็นชั้นล่างสุด จะทำหน้าที่รับส่งข้อมูลจริง จากสื่อ(media) ระหว่างคอมพิวเตอร์เครื่องต่างๆ มาตรฐานสำหรับลำดับชั้นในชั้นนี้จะกำหนดว่าแต่ละ Connector นั้น เช่น อาร์เอส-232-ซี มีกี่พิน (pin) แต่ละ พิน ทำหน้าที่อะไรบ้าง ใช้สัญญาณไฟกี่โวลต์ และเทคนิคของการมัลติเพล็กซ์แบบต่างๆ ก็จะถูกกำหนดอยู่ในส่วนลำดับชั้นในชั้นนี้
2. Data Link layer จะทำหน้าที่เหมือนผู้ตรวจสอบ และควบคุมความผิดพลาดของข้อมูล โดยจะทำการแบ่งข้อมูลออกเป็นแพ็กเกจหรือเฟรมแล้วส่งออก เมื่อผู้รับได้รับข้อมูลถูกต้องก็จะส่งสัญญาณยืนยันกลับมายาว ได้รับ ข้อมูลนั้นแล้ว เรียกว่า สัญญาณเอชเค (Acknowledge) ให้กับผู้ส่ง แต่ถ้าผู้ส่งไม่ได้รับสัญญาณเอชเค หรือได้รับ สัญญาณเอ็นเอชเค (Negative Acknowledge) กลับมา ซึ่งผู้ส่งนั้นอาจจะทำการส่งข้อมูลไปให้ใหม่ และอีกหน้า ของลำดับชั้นชั้นนี้คือป้องกันไม่ให้ส่งข้อมูลเร็วจนเกินขีดความสามารถของเครื่อง ผู้รับจะรับข้อมูลได้
3. Network layer ในชั้นนี้จะกำหนดและออกแบบเส้นทางในการรับ-ส่งข้อมูล ระหว่างเครื่องต้นทางกับเครื่องปลายทาง ซึ่งในการสื่อสารข้อมูลผ่านเครือข่ายนั้น จะมีอยู่มากกว่า 1 เส้นทาง ดังนั้นในชั้นนี้จะทำหน้าที่หาเส้นทางที่ดีที่สุด
4. Transport layer อาจเรียกว่า Host-to-Host หรือเครื่องต่อเครื่อง ซึ่งลำดับชั้นตั้งแต่ชั้นที่ 4 - 7 นี้รวมกันจะเรียกว่าลำดับชั้นEnd-to-End ในชั้น Transport นี้เป็นการสื่อสารกันระหว่างต้นทางและปลายทาง (คอมพิวเตอร์กับคอมพิวเตอร์) กันจริงๆ คือมีทำหน้าที่ตรวจสอบว่าข้อมูลที่ส่งมาจากลำดับชั้นชั้น Session นั้น ไปถึงปลายทางจริงๆ หรือไม่ ดังนั้นการกำหนดตำแหน่งของข้อมูล(address) จึงเป็นเรื่องสำคัญในชั้นนี้

5. Session layer ชั้นนี้ กับ ชั้น Transport อาจเป็นลำดับชั้น ชั้นเดียวกัน ลำดับชั้น ชั้นนี้ทำหน้าที่เชื่อมโยงระหว่างผู้ใช้กับเครื่องคอมพิวเตอร์เครื่องอื่นๆ โดยผู้ใช้คำสั่งหรือข้อความที่กำหนดไว้ป้อนเข้าไปในระบบ ในการสร้างการเชื่อมโยง
6. Presentation layer ทำหน้าที่แปลงรูปแบบข้อมูลให้มีรูปแบบการสื่อสารเดียวกัน เพื่อช่วยลดปัญหาต่าง ๆ ที่อาจจะเกิดขึ้นกับผู้ใช้งานในระบบ
7. Application layer เป็น ลำดับชั้น ชั้นบนสุดเป็นชั้นที่ใช้ติดต่อระหว่างผู้ใช้โดยตรง



รูปที่ 2.2 การส่งแพ็คเกจสวิตช์ใน OSI Model

ซึ่งเราสามารถแบ่งส่วนการทำงานของสถาปัตยกรรมรูปแบบ OSI ได้ง่าย ๆ โดยด้านนี้จะจัดแบ่งลำดับชั้นทั้ง 7 ชั้นออกเป็น 3 ส่วนคือส่วนของผู้ใช้งาน ส่วนการติดต่อระหว่างเครื่องต่อเครื่อง และส่วนการเชื่อมโยงต้นทางกับปลายทาง สำหรับในทางขวามือของรูปจะเป็นการจัดแบ่งลักษณะ การสื่อสารออกเป็น 2 ส่วนคือส่วนดำเนินการโดยผู้ใช้งาน และอีกส่วนหนึ่งเป็นการดำเนินการโดยเครือข่าย



รูปที่ 2.3 การดำเนินการ โดยผู้ใช้และส่วนดำเนินการโดยเครือข่าย

โดยโปรโตคอลของใน ลำดับชั้น แต่ละชั้นจะแตกต่างกันออกไป ดังตารางต่อไปนี้

ตารางที่ 2.1 โพรโทคอลของใน ลำดับชั้น แต่ละชั้น

Layer	มาตรฐาน	รายละเอียด
7	ISO 8571	การบริการ โอนถ่าย และการแลกเปลี่ยนข้อมูล
	ISO 8572	การบริการ โอนถ่าย และการแลกเปลี่ยนข้อมูล
	ISO 8831	การบริการ โอนถ่าย และการแลกเปลี่ยนข้อมูล
	ISO 8832	โพรโทคอลการบริการ โอนถ่าย และการแลกเปลี่ยนข้อมูล
	ISO 9040	การบริหารเทอร์มินัลแบบเสมือน
	ISO 9041	โพรโทคอลการบริหารเทอร์มินัลแบบเสมือน
	CCITT X.400	ไปรษณีย์อิเล็กทรอนิกส์ และกักเก็บข่าวสาร
6	ISO 8822	การบริหารแบบ Connection-oriented ใน Layer Presentation
	ISO 8823	โพรโทคอลการบริการแบบ Connection-oriented ใน Layer Presentation
5	ISO 8326	การบริการแบบ Connection-oriented ใน Layer Session
	ISO 8327	โพรโทคอลการบริการแบบ Connection-oriented ใน Layer Session
4	ISO 8072	การบริหารแบบ Connection-oriented ใน Layer Transport
	ISO 8073	โพรโทคอลการบริการแบบ Connection-oriented ใน Layer Transport
3	CCITT X.25	โพรโทคอล X.25 ใน Layer Network
2	ISO 8802 (IEEE 802)	โพรโทคอลสำหรับเครือข่ายท้องถิ่น (LAN)
	CCITT X.25	โพรโทคอล SDLC, HDLC ใน Layer Data Link
1	CCITT X.21	คิวิตอลอินเทอร์เฟซของ Layer Physical



## 2.3 โพรโทคอล (Protocol)

### 2.3.1 TCP/IP [1]

TCP/IP (Transmission control Protocol/Internet Protocol) เป็นชุดการทำงานที่ใช้ในการสื่อสารอินเทอร์เน็ต เพื่อจุดประสงค์ที่จะให้การสื่อสารระหว่างต้นทางไปยังปลายทาง และหาเส้นทางที่จะส่งข้อมูลได้อย่างเชี่ยวชาญและอัตโนมัติ

โดยโพรโทคอลชุดเหล่านี้ ได้รับการพัฒนามาโดยตลอด ซึ่งก็ได้ใช้เป็นที่ครั้งแรก ภายใต้เครือข่ายที่ชื่อว่า ARPANET จนเป็นที่ยอมรับจนเป็นสากลในปัจจุบัน

TCP/IP มีจุดประสงค์ในการทำงาน ดังนี้

1. เพื่อใช้ในการติดต่อสื่อสารในระบบที่แตกต่างกัน ให้ทำงานร่วมกันได้
  2. มีความสามารถในการแก้ไขปัญหาการติดต่อสื่อสารที่เกิดขึ้นในระบบเครือข่าย  
 อย่างเช่น ต้นทางและปลายทางยังติดต่อสื่อสารกันอยู่ ตัวกลางที่ใช้เป็นตัวช่วยในการรับส่งข้อมูล  
 เกิดชำรุดขึ้น ก็สามารถหาทางแก้ โดยจัดหาทางเลือกใหม่ เพื่อที่จะทำให้การสื่อสารดำเนินต่อไปได้  
 อย่างอัตโนมัติ
  3. การสื่อสารมีความคล่องตัว ทั้งงานเร่งด่วน และไม่เร่งด่วนก็ตาม
- โดยในแต่ละลำดับชั้นในโครงสร้างของ TCP/IP สามารถอธิบายได้ดังนี้

Application	FTP,SMTP
Transport	TCP,UDP
Internet	IP,ICMP,IGMP
Host-to-Network	Ether,SLIP

รูปที่ 2.4 โครงสร้าง TCP/IP

## 1. ชั้น โยส-เครือข่าย (Host-to-Network Layer)

กฎเกณฑ์ของระบบการทำงานของชั้นนี้ ไม่ได้กำหนดรายละเอียดอย่างเป็นทางการ แต่จะมีหน้าที่หลักๆคือ รับข้อมูลจากชั้นการสื่อสารไอพี แล้วส่งข้อมูลไปยัง โหนด ที่ระบุเส้นทางเดินไปยังผู้ส่ง และในทางกลับกันก็รับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นการสื่อสารอีกด้วย

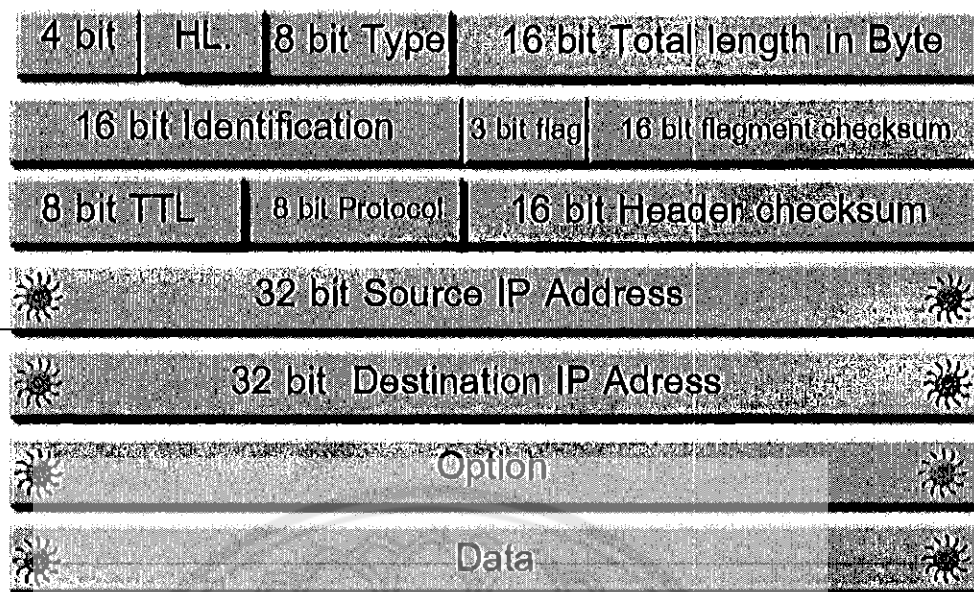
## 2. ชั้นการสื่อสารอินเทอร์เน็ต (The Internet Layer)

เป็นการสื่อสารที่เราเรียกกันว่า ระบบเครือข่ายแบบสลับช่องการสื่อสารในระดับ Packet ซึ่งเป็นการสื่อสารแบบไม่ต่อเนื่อง คือสามารถไหลข้อมูลจาก โหนด ผู้ส่งไปตาม โหนด ต่างๆได้ โดยอิสระ นั่นหมายความว่า แพ็กเก็ตแต่ละตัวเป็นอิสระต่อกัน ดังนั้น ข้อมูลจากต้นทางไปปลายทาง อาจไม่เป็นไปตามลำดับก็ได้ ซึ่งในชั้นการสื่อสารนี้จะมี 2 โพรโตคอล

### 2.3.2 IP (Internet Protocol)

IP เป็นโพรโตคอลในระดับ Network Layer ซึ่งจะทำหน้าที่หลักๆคือควบคุมการส่งข้อมูลที่ใช้ในการหาเส้นทาง ซึ่งจะต้องหาเส้นทางที่ดีที่สุด เพื่อการรับส่งข้อมูลที่มีประสิทธิภาพ และยังสามารถเปลี่ยนแปลงเส้นทางการรับส่งข้อมูล กอปรกับมีการแยกและประกอบ ดาต้าแกรม (Datagram) เพื่อรองรับข้อมูลในระดับ Data link ที่มี MTU (Maximum Transmission Unit) ต่างกัน จึงทำให้สามารถนำที่อยู่ไอพีไปใช้บนโพรโตคอลอื่นๆได้มากมาย

สำหรับ IP นั้น จะมีการเชื่อมต่อเพื่อทำการส่งข้อมูลแบบ Connectionless หรือเรียกอีกอย่างว่า เป็นการเกิดเส้นทางเชื่อมต่อในทุกๆครั้งที่มีการส่งข้อมูล 1 ดาต้าแกรม โดยเราจะไม่ทราบถึงข้อมูล ดาต้าแกรม ที่ส่งก่อนหน้าหรือที่ส่งตามมา แต่การส่งข้อมูล 1 ดาต้าแกรม นั้น ก็อาจจะส่งได้หลายครั้งก็ได้ หากเราแบ่งข้อมูลออกเป็นส่วนย่อยๆ (fragmentation) และถูกนำไปรวมเมื่อถึงปลายทาง ดังรูปด้านล่างนี้จะเป็น เสดเดอร์ ของ ไอพีจะเป็นการทำงานการหาช่องทางสื่อสารข้อมูล



รูปที่ 2.5 IP Header

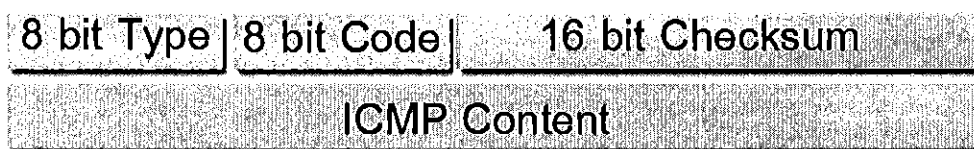
เฮดเดอร์ ของ ไอพี โดยปกติจะมีขนาด 20 ไบต์ ยกเว้นในกรณีที่มีการเพิ่มการเลือกบางอย่าง 필ด์ของเฮดเดอร์ ไอพีจะมีความหมายดังนี้

- Version** : หมายเลขเวอร์ชันของ โปรโตคอล ที่ใช้งานในปัจจุบันคือ ไอพีเวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)
- Header Length** : ความยาวของ Header โดยทั่วไปถ้าไม่มีการเลือก จะมีค่าเป็น 5 (5\*32 บิต)
- Type of Service (TOS)** : ใช้เป็นข้อมูลสำหรับเราเตอร์การตัดสินใจเลือกการเราเตอร์ข้อมูลในแต่ละคาต้าแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว
- Length** : ความยาวทั้งหมดเป็นจำนวน ไบต์ของคาต้าแกรมซึ่งด้วยขนาด 16 บิตของฟิลด์ จะหมายถึงความยาวสูงสุดของคาต้าแกรมคือ 65535 ไบต์ (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดใน Datalink layer และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทาง แอปพลิเคชันส่วนใหญ่จะมีขนาดของคาต้าแกรมไม่เกิน 512 ไบต์

- e. **Identification** : เป็นหมายเลขของดาต้าแกรมในกรณีที่มีการแยก ดาต้าแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี identification เดียวกันมารวมกัน
- f. **Flag** : ใช้ในกรณีที่มีการแยก Datagram
- g. **Fragment offset** : ใช้ในการกำหนดตำแหน่งข้อมูลในดาต้าแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง
- h. **Time to live (TTL)** : กำหนดจำนวนครั้งที่มากที่สุดที่ดาต้าแกรมจะถูกส่งระหว่างตำแหน่ง(การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 ตำแหน่งจะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิดการยกเลิกของเวลาระหว่างการส่งข้อมูล
- i. **Protocol** : ระบุโปรโตคอลที่ส่งในดาต้าแกรม TCP ,UDP หรือ ICMP
- j. **Header checksum** : ใช้ในการตรวจสอบความถูกต้องของข้อมูลใน Header
- k. **Source IP address** : หมายเลขไอพีของผู้ส่งข้อมูล
- l. **Destination IP address** : หมายเลขไอพีของผู้รับข้อมูล
- m. **Data** : ข้อมูลจากโปรโตคอลระดับบน

### 2.3.3 ICMP (Internet Control Message Protocol)

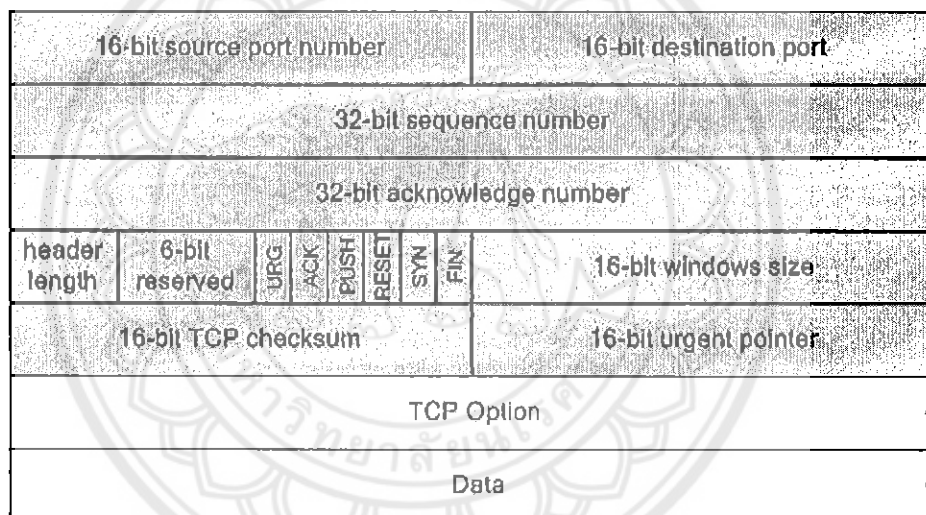
ทำหน้าที่เสริมการทำงานของไอพีให้สมบูรณ์ โดยจะเป็นโปรโตคอลที่คอยส่งข่าวสาร ตรวจสอบและรายงานสถานภาพของดาต้าแกรมแต่ก็ไม่มีอะไรรับประกันได้ว่าข้อความ ICMP ที่ส่งไปนั้นจะถึงผู้รับหรือไม่ หากแจ้งความผิดพลาดออกมา เราก็อาจจะแปลความหมายได้สองกรณีคือ ข้อมูลถูกส่งไปปลายทางเรียบร้อยแล้ว หรืออีกอย่างคือ การส่งข้อมูลมีปัญหา ดังนั้น ICMP จึงไม่ได้รับความเชื่อถือมากนัก ต้องให้ลำดับชั้นที่สูงกว่านั้น จัดการให้การสื่อสารนั้น น่าเชื่อถืออีกที



รูปที่ 2.6 ICMP Header

ชั้นการสื่อสารนำส่งข้อมูล (Transport Layer) จะแบ่งโปรโตคอลออกเป็น 2 ชนิดใหญ่ๆ

อย่างแรกคือ Transmission Control Protocol (TCP) เป็นการกำหนดช่วงระยะเวลาการสื่อสาร (Connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลแบบกระแสเป็นไบนารีที่ไว้วางใจได้ โดยไม่มีข้อผิดพลาด ปริมาณข้อมูลแต่ละชิ้น เราเรียกว่า ข้อความ นอกจากนั้น TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูล เพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกำหนดที่ผู้รับจะรับข้อมูลได้



รูปที่ 2.7 TCP Header

**Source Port Number:** หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้

**Destination Port Number:** หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

**Sequence Number:** 필ด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง

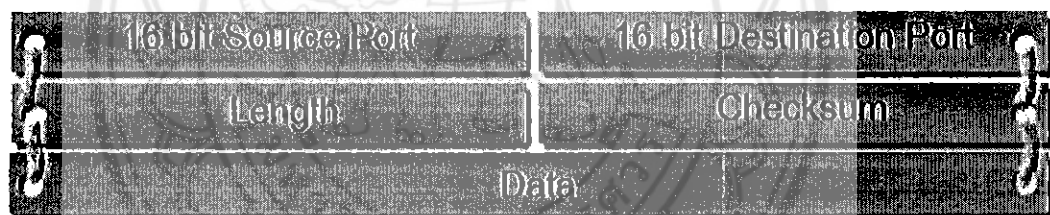
**Acknowledgment Number:** ทำหน้าที่เช่นเดียวกับลำดับหมายเลขแต่จะใช้ในการตอบรับ

**Header Length:** โดยปกติความยาวของ Header TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ตัวเลือก แต่ต้องไม่เกิน 60 ไบต์

**Flag:** เป็นข้อมูลระดับบิตที่อยู่ใน Header TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย

โปรโตคอลอีกชนิดที่เราเรียกว่า UDP เป็นการสื่อสารแบบไม่ต่อเนื่อง (Connectionless) โดย UDP จะส่งข้อมูลเพียงครั้งละ 1 ชุดข้อมูล เรียกว่า UDP Datagram ซึ่งไม่มีความสัมพันธ์กันระหว่างดาต้าแกรม และไม่มีกลไกการตรวจสอบ

แต่จะมีกลไกที่ป้องกันข้อมูลที่อาจถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง นั่นคือ UDP Checksum ซึ่งปลายทางจะรู้ว่าข้อมูลผิดพลาดเกิดขึ้น หากพบข้อผิดพลาด (Error) ในการเช็ค ผู้รับก็จะทำการทิ้งข้อมูลนั้น โดยไม่แจ้งกลับไปยังผู้ส่งแต่อย่างใด



รูปที่ 2.8 UDP Header

**Source Port Number:** หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้

**Destination Port Number:** หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

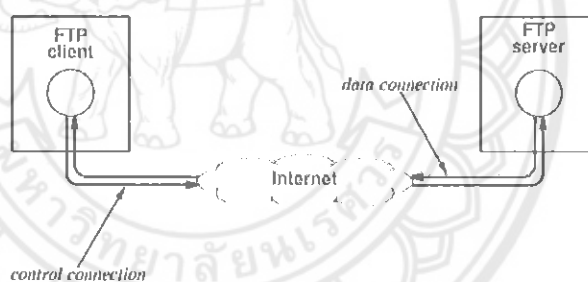
**UDP Length:** ความยาวของดาต้าแกรมทั้งส่วนเฮดเดอร์ และข้อมูลนั้นหมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของเฮดเดอร์

**Checksum:** เป็นตัวตรวจสอบความถูกต้องของ UDP ดาต้าแกรมและจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

### 2.3.4 FTP (File Transmission Protocol)

เอฟทีพี หรือ เคนท์วีธีถ่ายโอนแฟ้ม (อังกฤษ: FTP: File Transfer Protocol) เป็น โพรโทคอลเครือข่ายชนิดหนึ่ง ใช้สำหรับแลกเปลี่ยนและจัดการไฟล์บนเครือข่ายที่ซีพี/ไอพีเช่น อินเทอร์เน็ต เอฟทีพีถูกสร้างขึ้นด้วยสถาปัตยกรรมแบบระบบรับ-ให้บริการ (client-server) และใช้การเชื่อมต่อสำหรับส่วนข้อมูลและส่วนควบคุมแยกกันระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย โปรแกรมประยุกต์เอฟทีพีเริ่มแรกได้ต่อกันด้วยเครื่องมือรายการคำสั่ง สั่งการด้วยไวยากรณ์ที่เป็นมาตรฐาน แต่ก็มีการพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้ขึ้นมาสำหรับระบบปฏิบัติการเดสก์ท็อปที่ใช้กันทุกวันนี้ เอฟทีพียังถูกใช้เป็นส่วนประกอบของโปรแกรมประยุกต์อื่นเพื่อส่งผ่านไฟล์โดยอัตโนมัติสำหรับการทำงานภายใน โปรแกรม เราสามารถใช้เอฟทีพีผ่านทางกริพูจน์ตัวจริงด้วยชื่อผู้ใช้และรหัสผ่าน หรือเข้าถึงด้วยผู้ใช้นิรนาม

นอกจากนี้ยังมีทีเอฟทีพี (Trivial File Transfer Protocol) ซึ่งมีลักษณะคล้ายกับเอฟทีพีที่ลดความซับซ้อนลง แต่ไม่สามารถควบคุมให้ทำงานประสานกันได้ และไม่มีการพิสูจน์ตัวตนจริง



รูปที่ 2.9 การทำงานของ FTP

## 2.4 โทโปโลยี

โทโปโลยีคือลักษณะทางกายภาพ (ภายนอก) ของระบบเครือข่าย ซึ่งหมายถึง ลักษณะของการเชื่อมโยงสายสื่อสารเข้ากับอุปกรณ์ อิเล็กทรอนิกส์และเครื่องคอมพิวเตอร์ ภายในเครือข่ายด้วยกันนั่นเอง โทโปโลยีของเครือข่ายแลน แต่ละแบบมีความเหมาะสมในการใช้งาน แตกต่างกันไป การนำไปใช้จึงมีความจำเป็นที่เราจะต้องทำการศึกษาลักษณะและคุณสมบัติ ข้อดีและข้อเสียของโทโปโลยีแต่ละแบบ เพื่อนำไปใช้ในการออกแบบพิจารณาเครือข่าย ให้เหมาะสมกับการใช้งาน รูปแบบของโทโปโลยี ของเครือข่ายหลัก ๆ มีดังต่อไปนี้



รูปที่ 2.10 โทโปโลยีแบบบัส

#### 2.4.1 โทโปโลยีแบบบัส (BUS)

เป็นรูปแบบที่ เครื่องคอมพิวเตอร์จะถูกเชื่อมต่อกัน โดยผ่านสายสัญญาณแกนหลัก ที่เรียกว่าบัสหรือ แบล็ค โบน (Backbone) คือ สายรับส่งสัญญาณข้อมูลหลัก ใช้เป็นทางเดินข้อมูลของทุกเครื่องภายในระบบเครือข่าย และจะมีสายแยกย่อยออกไปในแต่ละจุด เพื่อเชื่อมต่อเข้ากับคอมพิวเตอร์เครื่องอื่น ๆ ซึ่งเรียกว่า โหนด (Node) ข้อมูลจากโหนดผู้ส่งจะถูกส่งเข้าสู่สายบัสในรูปของแพ็กเก็ต ซึ่งแต่ละแพ็กเก็ตจะประกอบไปด้วยข้อมูลของผู้ส่ง, ผู้รับ และข้อมูลที่ส่ง การสื่อสารภายในสายบัสจะเป็นแบบ 2 ทิศทางแยกไปยังปลายทั้ง 2 ด้านของ บัส โดยตรงปลายทั้ง 2 ด้านของบัส จะมีเทอร์มินเตอร์ (Terminator) ทำหน้าที่ลบล้างสัญญาณที่ส่งมาถึง เพื่อป้องกันไม่ให้สัญญาณข้อมูลนั้นสะท้อนกลับ เข้ามายังบัสอีก เพื่อเป็นการป้องกันการชนกันของข้อมูลอื่น ๆ ที่เดินทางอยู่บนบัสในขณะนั้น

สัญญาณข้อมูลจากโหนดผู้ส่งเมื่อเข้าสู่บัส ข้อมูลจะไหลผ่านไปยังปลายทั้ง 2 ด้านของบัส แต่ละโหนดที่เชื่อมต่อเข้ากับบัส จะคอยตรวจสอบว่า ตำแหน่งปลายทางที่มากับแพ็กเก็ตข้อมูลนั้นตรงกับตำแหน่งของตนหรือไม่ ถ้าตรง ก็จะรับข้อมูลนั้นเข้ามาสู่โหนด ตน แต่ถ้าไม่ใช่ ก็จะปล่อยให้สัญญาณข้อมูลนั้นผ่านไป จะเห็นว่าทุก ๆ โหนดภายในเครือข่ายแบบ BUS นั้นสามารถรับรู้สัญญาณข้อมูลได้ แต่จะมีเพียงโหนดปลายทางเพียง โหนดเดียวเท่านั้นที่จะรับข้อมูลนั้นไปได้

#### ข้อดี

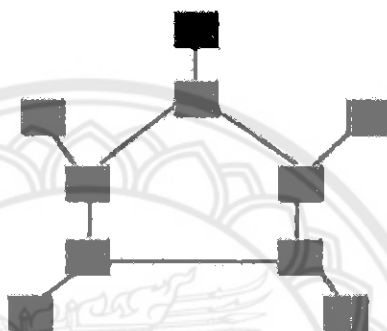
- ไม่ต้องเสียค่าใช้จ่ายในการวางสายสัญญาณมากนัก สามารถขยายระบบได้ง่าย เสียค่าใช้จ่ายน้อย ซึ่งถือว่าระบบบัสนี้เป็นแบบโทโปโลยีที่ได้รับความนิยมใช้กันมากที่สุดมา ตั้งแต่อดีตจนถึงปัจจุบัน เหตุผลอย่างหนึ่งก็คือสามารถติดตั้งระบบ ดูแลรักษา และติดตั้งอุปกรณ์เพิ่มเติมได้ง่าย ไม่ต้องใช้เทคนิคที่ยุ่งยากซับซ้อนมากนัก



### ข้อเสีย

- อาจเกิดข้อผิดพลาดง่าย เนื่องจากทุกเครื่องคอมพิวเตอร์ ต่ออยู่บนสายสัญญาณเพียงเส้นเดียว ดังนั้นหากมี สัญญาณขาดที่ตำแหน่งใดตำแหน่งหนึ่ง ก็จะทำให้เครื่องบางเครื่อง หรือทั้งหมดในระบบไม่สามารถใช้งานได้ตามไปด้วย

- การตรวจหาโหนดเสีย ทำได้ยาก เนื่องจากขณะใดขณะหนึ่ง จะมีคอมพิวเตอร์เพียงเครื่องเดียวเท่านั้น ที่สามารถส่งข้อความ ออกมาบนสายสัญญาณ ดังนั้นถ้ามีเครื่องคอมพิวเตอร์จำนวนมากๆ อาจทำให้เกิดการคับคั่งของเน็ตเวิร์ค ซึ่งจะทำให้ระบบช้าลงได้



รูปที่ 2.11 โทโปโลยี แบบวงแหวน

### 2.4.2 โทโปโลยีแบบวงแหวน (RING)

เป็นรูปแบบที่ เครื่องคอมพิวเตอร์ทุกเครื่องในระบบเครือข่าย ทั้งเครื่องที่เป็นผู้ให้บริการ (Server) และ เครื่องที่เป็นผู้ขอใช้บริการ (Client) ทุกเครื่องถูกเชื่อมต่อกันเป็นวงกลม ข้อมูลข่าวสารที่ส่งระหว่างกัน จะไหลวนอยู่ในเครือข่ายไปในทิศทางเดียวกัน โดยไม่มีจุดปลายหรือเทอร์มินเตอร์เช่นเดียวกับเครือข่ายแบบบัสในแต่ละโหนดหรือแต่ละเครื่อง จะมีรีพีตเตอร์ (Repeater) ประจำแต่ละเครื่อง 1 ตัว ซึ่งจะทำหน้าที่เพิ่มเติมข้อมูลที่จำเป็นต่อการติดต่อสื่อสารเข้าในส่วนหัวของแพ็กเก็ตที่ส่ง และตรวจสอบข้อมูลจากส่วนหัวของแพ็กเก็ตที่ส่งมาถึง ว่าเป็นข้อมูลของตนหรือไม่ แต่ถ้าไม่ใช่ก็จะปล่อยข้อมูลนั้นไปยังรีพีตเตอร์ของเครื่องถัดไป

### ข้อดี

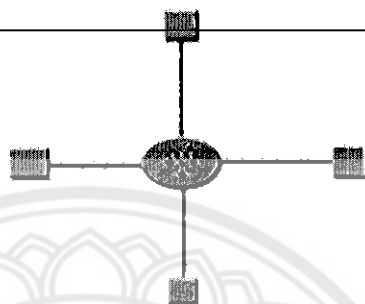
- ผู้ส่งสามารถส่งข้อมูลไปยังผู้รับได้หลาย ๆ เครื่องพร้อม ๆ กัน โดยกำหนดตำแหน่งปลายทางเหล่านั้นลงในส่วนหัวของแพ็กเก็ตข้อมูล รีพีตเตอร์ของแต่ละเครื่องจะทำการตรวจสอบเองว่า ข้อมูลที่ส่งมาให้นั้น เป็นตนเองหรือไม่

- การส่งผ่านข้อมูลในเครือข่ายแบบวงแหวน จะเป็นไปในทิศทางเดียวจากเครื่องสู่เครื่อง จึงไม่มีการชนกันของสัญญาณ ข้อมูลที่ส่งออกไป

- คอมพิวเตอร์ทุกเครื่องในเน็ตเวิร์คมีโอกาที่จะส่งข้อมูลได้อย่างทัดเทียมกัน

### ข้อเสีย

- ถ้ามีเครื่องใดเครื่องหนึ่งในเครือข่ายเสียหาย ข้อมูลจะไม่สามารถส่งผ่านไปยังเครื่องต่อ ๆ ไปได้ และจะทำให้เครือข่ายทั้งเครือข่าย หยุดชะงักได้
- ขณะที่ข้อมูลถูกส่งผ่านแต่ละเครื่อง เวลาส่วนหนึ่งจะสูญเสียไปกับการที่ทุก ๆ รีพีเทเตอร์ จะต้องทำการตรวจสอบตำแหน่งปลายทางของข้อมูลนั้น ๆ ทุก ข้อมูลที่ส่งผ่านมาถึง



รูปที่ 2.12 โทโปโลยี แบบดาว

### 2.4.3 โทโปโลยีแบบดาว (STAR)

เป็นรูปแบบที่ เครื่องคอมพิวเตอร์ทุกเครื่องที่เชื่อมต่อเข้าด้วยกันในเครือข่าย จะต้องเชื่อมต่อกับอุปกรณ์ตัวกลางตัวหนึ่งที่เรียกว่าเครื่องกระจายข้อมูล(HUB) หรือเครื่อง ๆ หนึ่ง ซึ่งทำหน้าที่เป็นศูนย์กลางของการเชื่อมต่อสายสัญญาณที่มาจากเครื่องต่าง ๆ ในเครือข่าย และควบคุมเส้นทางการสื่อสาร ทั้งหมด เมื่อมีเครื่องที่ต้องการส่งข้อมูลไปยังเครื่องอื่น ๆ ที่ต้องการในเครือข่าย เครื่องนั้นก็ต้องส่งข้อมูลมายังเครื่องกระจายข้อมูล หรือเครื่องศูนย์กลางก่อน แล้วเครื่องกระจายสัญญาณ ก็จะทำหน้าที่กระจายข้อมูลนั้น ไปในเครือข่ายต่อไป

### ข้อดี

- การติดตั้งเครือข่ายและการดูแลรักษาทำได้ง่าย หากมีเครื่องใดเกิดความเสียหาย ก็สามารถตรวจสอบได้ง่าย และศูนย์กลางสามารถตัดเครื่องที่เสียหายนั้นออกจากการสื่อสาร ในเครือข่ายได้เลย โดยไม่มีผลกระทบต่อระบบเครือข่าย

### ข้อเสีย

- เสียค่าใช้จ่ายมาก ทั้งในด้านของเครื่องที่จะใช้เป็น เครื่องศูนย์กลาง หรือตัวเครื่องกระจายข้อมูลเอง และค่าใช้จ่ายในการติดตั้งสายเคเบิลในเครื่องอื่น ๆ ทุกเครื่อง การขยายระบบให้ใหญ่ขึ้นทำได้ยาก เพราะการขยายแต่ละครั้ง จะต้องเกี่ยวเนื่องกับเครื่องอื่นๆ ทั้งระบบ

## 2.4.4 โทโปโลยีแบบ Hybrid

เป็นรูปแบบใหม่ ที่เกิดจากการผสมผสานกันของ โทโปโลยีแบบ ดาว , บัส, วงแหวนเข้าด้วยกัน เพื่อเป็นการลดข้อเสียของรูปแบบที่กล่าวมา และเพิ่มข้อดีขึ้นมา มักจะนำมาใช้กับระบบวง (Wide Area Network) มาก ซึ่งการเชื่อมต่อกันของแต่ละรูปแบบนั้น ต้องใช้ตัวเชื่อมต่อสัญญาณเข้ามาเป็นตัวเชื่อม ตัวนั้นก็คือเราเตอร์ เป็นตัวเชื่อมการติดต่อกัน

## 2.4.5 โทโปโลยีแบบ MESH

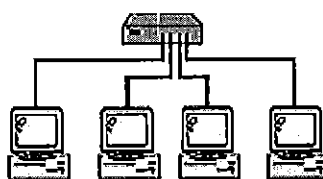
เป็นรูปแบบที่ถือว่า สามารถป้องกันการผิดพลาดที่อาจจะเกิดขึ้นกับระบบได้ดีที่สุด เป็นรูปแบบที่ใช้วิธีการเดินสายของแต่ละเครื่อง ไปเชื่อมการติดต่อกับทุกเครื่องในระบบเครือข่าย คือเครื่องทุกเครื่องในระบบเครือข่ายนี้ ต้องมีสายไปเชื่อมกับทุก ๆ เครื่อง ระบบนี้ยากต่อการเดินสาย และมีราคาแพง จึงมีค้อยมีผู้นิยมมากนัก

## 2.5 ประเภทของระบบเครือข่ายแลน ซึ่งแบ่งตามลักษณะการทำงาน

ในการแบ่งรูปแบบการเชื่อมต่อระบบเครือข่ายแลน นั้น สามารถแบ่งออกเป็น 2 ประเภทใหญ่ ๆ ได้แก่การเชื่อมต่อแบบ หนึ่งต่อหนึ่ง และแบบ ลูกข่าย/ แม่ข่าย

### 2.5.1 แบบ หนึ่งต่อหนึ่ง

เป็นการเชื่อมต่อเครื่องคอมพิวเตอร์เข้าด้วยกัน โดยเครื่องคอมพิวเตอร์ แต่ละเครื่อง จะสามารถแบ่งทรัพยากรต่างๆ ไม่ว่าจะเป็นไฟล์หรือเครื่องพิมพ์ซึ่งกันและกันภายในเครือข่ายได้ เครื่องแต่ละเครื่องจะทำงานในลักษณะที่ตัดเทียมกัน ไม่มีเครื่องใดเครื่องเครื่องหนึ่งเป็นเครื่องหลักเหมือนแบบ ลูกข่าย/ แม่ข่ายแต่ก็ยังคงคุณสมบัติพื้นฐานของระบบเครือข่ายไว้เหมือนเดิม การเชื่อมต่อแบบนี้มักทำในระบบที่มีขนาดเล็กๆ เช่น หน่วยงานขนาดเล็กที่มีเครื่องใช้ไม่เกิน 10 เครื่อง การเชื่อมต่อแบบนี้มีจุดอ่อนในเรื่องของระบบรักษาความปลอดภัย แต่ถ้าเป็นเครือข่ายขนาดเล็ก และเป็นงานที่ไม่มีข้อมูลที่เป็นความลับมากนัก เครือข่ายแบบนี้ ก็เป็นรูปแบบที่น่าเลือกนำมาใช้ได้เป็นอย่างดี

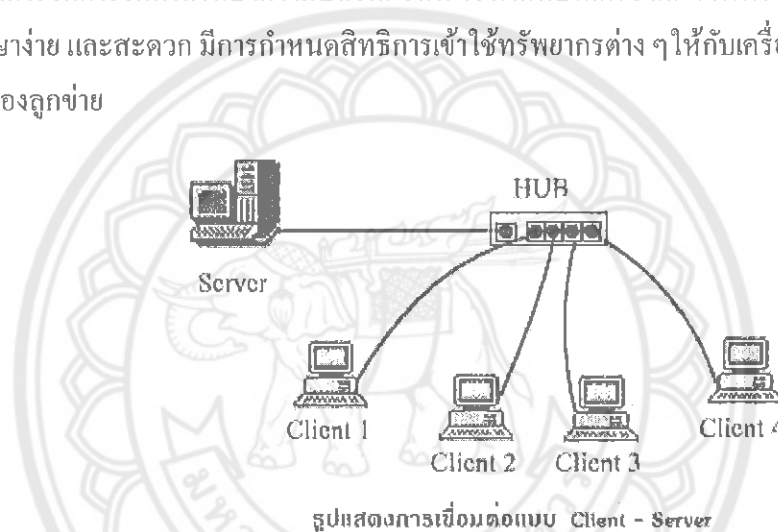


Peer to Peer

## รูปที่ 2.13 ลักษณะการทำงานแบบ หนึ่งต่อหนึ่ง

### 2.5.2 แบบ ถูกข่าย-แม่ข่าย

เป็นระบบที่มีเครื่องคอมพิวเตอร์ทุกเครื่องมีฐานะการทำงานที่เหมือน ๆ กัน เท่าเทียมกัน ภายในระบบ เครือข่าย แต่จะมีเครื่องคอมพิวเตอร์เครื่องหนึ่ง ที่ทำหน้าที่เป็นเครื่อง Server ที่ทำหน้าที่ให้บริการทรัพยากรต่าง ๆ ให้กับ เครื่องลูกข่ายหรือเครื่องที่ขอใช้บริการ ซึ่งอาจจะต้องเป็นเครื่องที่มีประสิทธิภาพที่ค่อนข้างสูง ถึงจะทำให้การให้บริการมีประสิทธิภาพตามไปด้วย ข้อดีของระบบเครือข่ายลูกข่าย-แม่ข่ายเป็นระบบที่มีการรักษาความปลอดภัยสูงกว่า ระบบแบบหนึ่งต่อหนึ่ง เพราะการจัดการในด้านรักษาความปลอดภัยนั้น จะทำกันบนเครื่องแม่ข่ายเพียงเครื่องเดียว ทำให้ดูแลรักษาง่าย และสะดวก มีการกำหนดสิทธิการเข้าใช้ทรัพยากรต่าง ๆ ให้กับเครื่องผู้ขอใช้บริการ หรือเครื่องลูกข่าย



รูปแสดงการเชื่อมต่อแบบ Client - Server

### รูปที่ 2.14 การติดต่อ ลูกข่าย-แม่ข่าย

ประเภทของระบบเครือข่ายมีอีกรูปแบบหนึ่งที่กำลังเป็นที่นิยมใช้กันในปัจจุบัน ก็คือ การเชื่อมต่อแลนแบบไร้สาย (Wireless LAN) เป็นเทคโนโลยีที่นำมาใช้ได้อย่างกว้างขวาง เหมาะที่จะใช้ได้ทั้งเครื่องพีซีตั้งโต๊ะธรรมดา และเครื่องคอมพิวเตอร์แบบพกพา ซึ่งการส่งสัญญาณติดต่อกันนั้น จะใช้สัญญาณวิทยุเป็นพาหะ ดังนั้นความเร็วในการส่งข้อมูลก็จำเป็นต้องขึ้นอยู่กับระยะทาง ยิ่งไกล ความเร็วในการส่งข้อมูลก็ทำให้ช้าลงไปด้วย แลนไร้สายเหมาะที่จะนำมาใช้กับงานที่ต้องการความคล่องตัวในการปฏิบัติงาน อย่างเช่นพวก เครื่องคอมพิวเตอร์แบบพกพา เพียงแต่มีอินเตอร์เฟสแลนแบบไร้สาย ก็สามารถเคลื่อนที่ไปที่ใดก็ได้ภายในขอบเขตของระยะทางที่กำหนด อย่างเช่นภายในตึกได้ทั่วตึกเลยทีเดียว จุดเด่น ๆ ของ เครือข่ายไร้สาย มีดังนี้

- การเคลื่อนที่ทำได้สะดวก สามารถใช้ระบบแลนจากที่ใดก็ได้ และสามารถเข้าถึงข้อมูลได้แบบต่อเนื่องได้อีกด้วย

- การติดตั้งใช้งานง่าย และรวดเร็ว ไม่ต้องเดินสายสัญญาณให้ยุ่งยาก
  - การติดตั้งและการขยายระบบ ทำได้อย่างกว้างขวาง เพราะสามารถขยายไปติดตั้งใช้งานในพื้นที่ที่สายสัญญาณเข้าไม่ถึง
  - เสียค่าใช้จ่ายลดน้อยลง เพราะในปัจจุบันการส่งสัญญาณของ เครื่องข่ายไร้สาย ทำได้ไกลมากยิ่งขึ้น สามารถส่งได้ไกลกว่า 10 กม. ทำให้ลดค่าใช้จ่ายในส่วนของการเข้าสายสัญญาณลงไปได้เป็นอย่างมาก
- 
- มีความยืดหยุ่นในการใช้งานและการติดตั้ง สามารถปรับแต่งระบบให้ใช้ได้กับทุกโทโปโลยีเครือข่าย การปรับแต่งทำได้ง่าย ไม่ว่าจะเป็นการติดตั้งเครือข่าย การติดตั้งแอปพลิเคชันต่างๆทำได้โดยง่าย

มาตรฐานของเครือข่ายไร้สาย นั้นตามมาตรฐานสากล 802.11 มีอัตราการส่งสัญญาณข้อมูลได้สูงสุด 11 เมกะบิตต่อวินาที ระยะทางการรับส่งสัญญาณขึ้นอยู่กับผู้ผลิตว่าออกแบบมาอย่างไร ถ้าเป็นการใช้ภายในอาคารสถานที่ ก็จะใช้สายอากาศแบบทุกทิศทาง จะได้ระยะทางประมาณ 50 เมตร แต่ถ้าเป็นการใช้กันแบบจุดต่อจุดหรือนอกสถานที่ ก็จะมีการออกแบบให้ใช้สายอากาศแบบกำหนดทิศทาง ให้ได้ระยะทางมากกว่า 10 กม.ได้

## 2.6 ความรู้เกี่ยวกับ Window API [4]

API มาจากคำว่า Application Programming Interface คือ กลุ่มของไลบรารี (Library) ซึ่งใช้เป็นส่วนติดต่อระหว่างแอปพลิเคชัน โดยเกือบทุกโปรแกรมในระบบปฏิบัติการแบบวินโดวส์ นั้นจะใช้ API ในการติดต่อ ซึ่งการทำงานของวินโดวส์ API สามารถแบ่งออกได้เป็น 8 ประเภท คือ

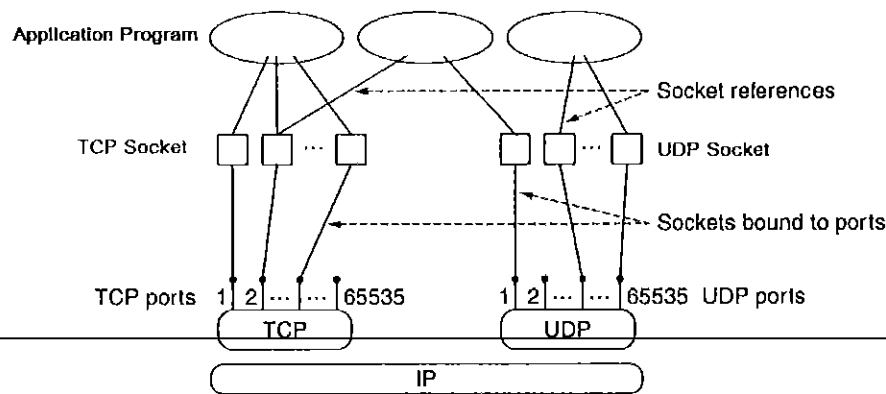
1. Base Services มีหน้าที่ในการเข้าถึงแหล่งข้อมูลพื้นฐานที่สามารถใช้ได้ในระบบวินโดวส์ รวมถึงที่ต้องการจากระบบแฟ้ม, อุปกรณ์, ตัวจัดการงาน กับ ตัวแบ่งงานและการจัดการข้อผิดพลาด ซึ่งฟังก์ชันเหล่านี้อยู่ใน kernel32.dll ของวินโดวส์ แบบ 32 บิต
2. Advanced Services มีหน้าที่เกี่ยวกับการเข้าถึงฟังก์ชันการทำงานเพิ่มเติมเคอร์เนล ปิด, รีสตาร์ทระบบ (หรือยกเลิก), เปิด, หยุด, สร้างบริการวินโดวส์, จัดการบัญชีผู้ใช้ ซึ่งฟังก์ชันเหล่านี้อยู่ใน Advapi32.dll ในวินโดวส์แบบ 32 บิต
3. Graphics Device Interface เป็นฟังก์ชันสำหรับการทำงานเพื่อแสดงผลเนื้อหาเป็นกราฟิก, เครื่องพิมพ์ และอื่น ๆ อุปกรณ์เอาท์พุทแบบอื่นๆ ซึ่งฟังก์ชันเหล่านี้อยู่ในGdi32.dll ในวินโดวส์แบบ 32 บิต

4. User Interface เป็นฟังก์ชันการที่มีหน้าที่ สร้างและจัดการหน้าจอของวินโดว์ และยังช่วยควบคุมสิ่งพื้นฐานต่างๆ ปุ่ม, เม้าส์ , คีย์บอร์ดและฟังก์ชันอื่น ๆ ที่เกี่ยวข้องกับระบบติดต่อกับผู้ใช้งานของวินโดว์ซึ่งฟังก์ชันเหล่านี้อยู่ใน user32.dll ในวินโดว์แบบ 32 บิต
5. Common Dialog Box Library มีหน้าที่เกี่ยวกับการประยุกต์กล่องโต้ตอบ สำหรับเปิดและบันทึกไฟล์, การเลือกสีและตัวอักษร ฯลฯ ซึ่งฟังก์ชันเหล่านี้อยู่ใน comdlg32.dll ของวินโดว์แบบ 32 บิต
6. Common Control Library มีหน้าที่ในการเข้าถึงการควบคุมขั้นสูงของระบบปฏิบัติการบางส่วน รวมถึง แถบสถานะ , แถบความคืบหน้า , แถบเครื่องมือและแท็บ ซึ่งฟังก์ชันเหล่านี้อยู่ในComctl32.dll บนวินโดว์แบบ 32 บิต
7. Windows Shell จะมีหน้าที่เกี่ยวกับเรื่องของการแสดง ซึ่งฟังก์ชันเหล่านี้อยู่ในshell32.dll บนวินโดว์แบบ 32 บิต
8. Network Services จะมีหน้าที่เกี่ยวกับเรื่องของเน็ตเวิร์ค

## 2.7 ความรู้เกี่ยวกับ Socket [3]

Windows Socket API(WSA) หรือเรียกสั้นๆ ว่า Winsock เป็น API ที่ใช้ในการเข้าถึงเน็ตเวิร์คที่ใช้โปรโตคอล TCP/IP มันจัดเตรียมไว้ในการเชื่อมต่อระหว่างแอปพลิเคชัน ของลูกข่ายกับ TCP/IP ไปยังเครื่องคอมพิวเตอร์เครื่องอื่นที่ทำการติดต่อกันทางโปรโตคอล TCP/IP เดียวกันในชั้น Transport Layer ซึ่งจะมี Socket ที่ประกอบด้วยที่อยู่ไอพี ซึ่งเปรียบเสมือนเลขที่บ้าน และพอร์ตที่เปลี่ยนเสมือนเลขห้อง เพื่อใช้ในการติดต่อกันระหว่างเครือข่าย ซึ่งก่อนที่เราจะส่งหรือรับข้อมูลเราต้องทราบที่อยู่ไอพี และพอร์ตที่ต้องการที่จะรับหรือส่ง การที่จะเชื่อมต่อไปยังปลายทางได้นั้น จะต้องให้เครื่องปลายทาง เปิดพอร์ตรอการเชื่อมต่อไว้ เรียกฝั่งที่เปิด Socket โดยแม่ข่ายจะต้องระบุว่า ผู้ที่จะเชื่อมต่อเข้ามา จะเชื่อมต่อเข้ามาทางพอร์ต ไหนส่วนฝั่งที่ทำการเชื่อมต่อเข้าไป เรียกว่าลูกข่าย

ความแตกต่างของ Socket จะขึ้นอยู่กับการใช้โปรโตคอล ชนิดของ socket ซึ่งในที่นี้เราจะพูดถึง จะเป็นโปรโตคอล TCP/IP จะมี Socket ที่สำคัญคือ stream socket ในโปรโตคอล TCP , datagram socket ในโปรโตคอล UDP และ Raw Socket



รูปที่ 2.15 ความสัมพันธ์ ระหว่าง Application, Socket และ โพรโตคอล

ในรูปแสดงความสัมพันธ์ระหว่างแอปพลิเคชัน, socket, โพรโตคอล, และพอร์ตภายในโฮสต์เดียว ซึ่งจากรูปจะเห็นได้ว่า socket เดียว สามารถที่จะอ้างอิงได้หลายแอปพลิเคชัน และแอปพลิเคชันสามารถที่จะติดต่อผ่าน socket ซึ่งมีพอร์ตของ socket แต่ละตัวอยู่ ถ้าแอปพลิเคชันมีความต้องการที่เหมือนกันก็สามารถที่จะใช้ socket เดียวกันในการติดต่อ โพรโตคอลได้เช่นเดียวกัน

### 2.7.1 Stream Socket

เป็นการจัดเตรียมการติดต่อแบบสองทาง มีความต่อเนื่อง นำเชื่อถือ และมีการจำลองเส้นทางการไหลของข้อมูล ไม่มีการกำหนดขอบเขต การดำเนินการของกระแสข้อมูลมีลักษณะการสื่อสารเหมือนการติดต่อทางโทรศัพท์ ซึ่งใช้ในโพรโตคอล TCP

### 2.7.2 การใช้งาน Stream Socket [2]

เมื่อทำการสร้าง Socket ระบบจะทำการจัดสรรหน่วยความจำเพื่อสร้างเป็นโครงสร้างของข้อมูล เพื่อเก็บข้อมูลที่ใช้ในการติดต่อสื่อสาร ซึ่งในโครงสร้างจะมีฟิลด์เมื่อ Socket มีการเรียกใช้งานข้อมูลต่างๆ จะถูกเก็บไว้ในฟิลด์ ที่ได้สร้างไว้

Socket จะมีหน้าที่ รอรับการเชื่อมต่อและเชื่อมต่อ ไปยังเครื่องที่เราต้องการติดต่อ ซึ่งการรอรับการเชื่อมต่อ ใช้ใน server Application ซึ่งจะเรียกว่า Passive Socket ส่วนทางด้านแอปพลิเคชันของลูกข่ายที่ทำการเชื่อมต่อไปยังปลายทางจะเรียกว่า Active Socket การสร้าง Socket ทั้งฝั่ง Passive Socket และ Active Socket

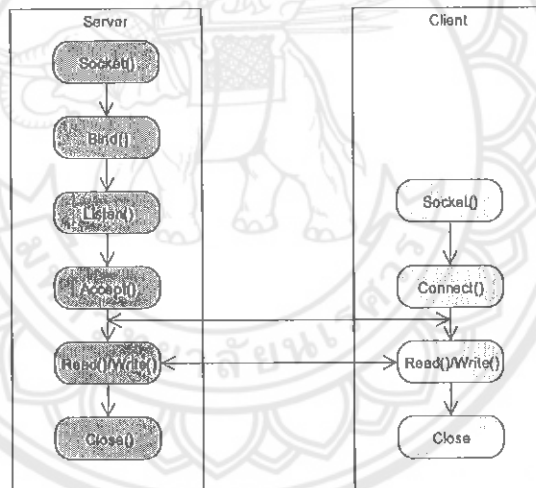
ขั้นตอนการสร้าง Passive Socket

1. ทำการเปิด Socket
2. คอยการติดต่อเข้ามา
3. รับการติดต่อจากลูกข่าย
4. ส่งและรับ ข้อมูล
5. ปิด Socket

#### ขั้นตอนการสร้าง Active Socket

1. ทำการ Open Socket
2. ติดต่อไปยัง Remote Host
3. ส่งหรือรับ ข้อมูล
4. ปิด Socket

ขั้นตอนการทำงานมีลักษณะดังรูป



รูปที่ 2.16 ขั้นตอนการทำงานของ Stream Socket

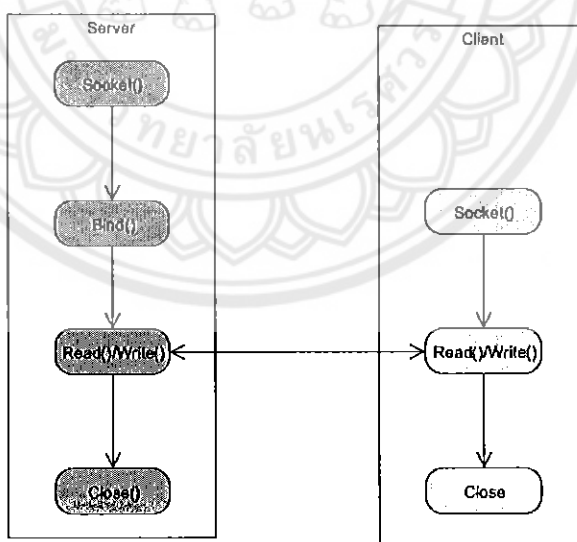


### 2.7.3 Datagram Socket

สนับสนุนการติดต่อแบบสองทิศทาง อาจมีการรับข้อความที่มีความแตกต่างกันของคำสั่ง มีการกำหนดขอบเขตของข้อมูลเพื่อปกป้องไว้ การทำงานของ Datagram socket มีลักษณะเหมือน การตอบจดหมาย หรือ ส่งจดหมายไปชื่อนอก ซึ่งใช้ใน โพรโตคอล UDPการใช้งาน Datagram Socket

### 2.7.4 การใช้งาน Datagram Socket

การทำงานจะทำงานระหว่าง แม่ข่ายและลูกข่าย เพราะฉะนั้นต้องทำการสร้าง Socket ไว้ ทั้งสองฝั่งเพื่อใช้ในการติดต่อระหว่างกันเมื่อทำการสร้าง เสร็จแล้วการทำงานจะเริ่มจาก ฝั่งแม่ข่าย ทำการผูกจุดหมายปลายทางที่เราจะรับการติดต่อ ถ้าจะรอเครื่องที่ไม่เจาะจงสามารถใช้ฟิลด์ IPAddress.Any เพื่อรอการติดต่อทุกเครื่องที่ติดต่อมาที่พอร์ตนี้โดยไม่สนใจที่อยู่ไอพีเมื่อลูกข่าย สร้าง socket แล้วสามารถส่งข้อมูลหากันได้เลยโดยไม่ต้องทำการติดต่อก่อน เหมือน Stream Socket เพราะมันจะส่งไปตามที่ Socket ทำการใส่ที่อยู่ไอพีและพอร์ต ไปแล้ว การส่งนี้จึงไม่รับประกัน ความสมบูรณ์ของข้อมูล ขั้นตอนการทำงานของ Socket เป็นดังรูป



รูปที่ 2.17 ขั้นตอนการทำงานของ Datagram Socket

### 2.7.5 Asynchronous Sockets

Asynchronous จะเหมาะกับงานที่ใช้เวลาทำที่นานและมีหลายขั้นตอน เพราะต้องทำงานที่ละขั้นตอนและต้องรอให้งานแรกทำงานเสร็จก่อนถึงจะสามารถทำงานต่อไปได้ เนื่องจากความสามารถของ Asynchronous คือการไม่ต้องรอให้การทำงานขั้นตอนแรกทำเสร็จก่อน ซึ่งมันสามารถที่จะข้ามไปทำขั้นตอนต่อไปได้โดยทำให้มีความรวดเร็วในการทำงานและสามารถสร้างงานได้มากกว่าการทำงานแบบเก่า ซึ่งในการใช้กับ Socket ก็สามารที่จะทำได้เช่นกัน

การทำงานของ Asynchronous Socket จะอยู่บนความสามารถของ .Net Framework ซึ่งการทำงานจะใช้หลักการทำงานของตัวแบ่งงานที่มีความสามารถในการแยกการทำงาน โดยไม่ให้ขึ้นกับการทำงานหลัก ซึ่งทำให้การทำงานย่อยสามารถที่จะทำงานไปพร้อมกับการทำงานหลักพร้อมกันได้ ซึ่งใน .Net Framework สามารถแบ่งการทำงานแบบการทำงาน เป็นสองแบบคือ แบบใช้วัตถุที่ชื่อ IAsyncResult และอีกแบบคือใช้ event driving ซึ่งการทำงานแบบ IAsyncResult จะมีความยืดหยุ่นสูง แต่การทำงานของมันค่อนข้างที่จะซับซ้อนพอสมควร ผู้พัฒนาจึงต้องมีความรู้เกี่ยวกับวิธีการใช้งานในระดับหนึ่ง

ซึ่งการความสามารถในการทำงานกับ Socket สามารถทำงานแบบ Socket IO ซึ่งการทำงาน จะต้องทำงานโดยมีรูปแบบการทำงานเป็นคู่ คือแบบเริ่มต้นด้วย BeginMethodname และจบด้วย EndMethodname ซึ่งจะเป็นการเริ่มต้นและสิ้นสุดการทำงานแบบ Asynchronous ซึ่งการทำงานแบบนี้หาก EndMethodname ถูกเรียกให้ทำงานก่อนที่งานจะเสร็จจะทำให้ Thread การทำงานที่ได้ทำนั้นถูกจำกัดไว้จนกว่าการทำงานนั้นจะสำเร็จ เพื่อหลีกเลี่ยงการทำงานนั้นไม่ให้ถูกจำกัด เราจึงต้องเขียนโปรแกรมในการทำงานดังนี้ คือการใช้คุณสมบัติที่ชื่อ IsCompleted ถ้าเป็น True จะแสดงว่าการทำงานเสร็จแล้วก็ให้หยุดการทำงานของ Asynchronous และอีกแบบคือการใช้ AsyncCallback delegate ซึ่งจะทำการเรียกเมื่อ มีการทำงานเสร็จสิ้น

## 2.8 ความรู้เกี่ยวกับพอร์ต [5]

ในโปรโตคอล TCP และ UTP ที่ใช้ในการติดต่อระหว่างคอมพิวเตอร์ เมื่อต้องการที่จะติดต่อกับบริการ ต้องรู้พอร์ตที่บริการที่ทำงานอยู่ หมายเลขพอร์ตจะเป็นเลขฐาน 16 บิต ตั้งแต่ 0 ถึง 65535 หมายเลขพอร์ต แต่ละหมายเลขจะถูกกำหนดโดยเฉพาะจาก OS (Operating Systems) ซึ่งมี IANA ( Internet Assigned Numbers Authority) เป็นหน่วยงานกลางในการประสานการใช้พอร์ต ซึ่งพอร์ตจะถูกกำหนดให้เหมาะกับ service ถ้าพอร์ตไหนถูกกำหนดโดย IANA จะเป็นพอร์ตแบบเป็นทางการ แต่ ถ้า IANA ไม่ได้กำหนดขึ้นมาจะเป็นแบบไม่เป็นทางการและสามารถแบ่งตามหมายเลขพอร์ต เป็นแบบที่รู้จักกันดี จะเป็นหมายเลขพอร์ตอยู่ที่ 0 – 1023 ถ้าเป็นแบบการบันทึกหมายเลขพอร์ตที่มีหมายเลขพอร์ต อยู่ที่ 1024 – 49151 และหมายเลขพอร์ตที่อยู่ระหว่าง 49152 – 65535 จะเป็นพอร์ตที่ไม่ได้ถูกกำหนดขึ้นจาก IANA มันจะถูกใช้ในการจัดสรรอัตโนมัติโดยผู้ใช้ใช้เป็นพอร์ตชั่วคราว จะเป็นพอร์ตเคลื่อนที่

### 2.8.1 Well known Ports

Well Known Ports ก็จะเป็นพอร์ตที่ระบบส่วนใหญ่ กำหนดให้ใช้โดย Privileged User (ผู้ใช้ที่มีสิทธิพิเศษ) โดยพอร์ตเหล่านี้ ใช้สำหรับการติดต่อระหว่างเครื่องที่มีระบบเวลาที่ยาวนาน วัตถุประสงค์เพื่อให้บริการแก่ผู้ใช้ (ที่ไม่รู้จักหรือคุ้นเคย) แปรลกหน้า จึงจำเป็นต้องกำหนดพอร์ตติดต่อสำหรับ Service นั้นๆ

### 2.8.2 Registered Ports

จะเป็นพอร์ตหมายเลข 1024 ขึ้นไป ซึ่ง IANA ไม่ได้กำหนดไว้

ตารางที่ 2.2 หมายเลขพอร์ต

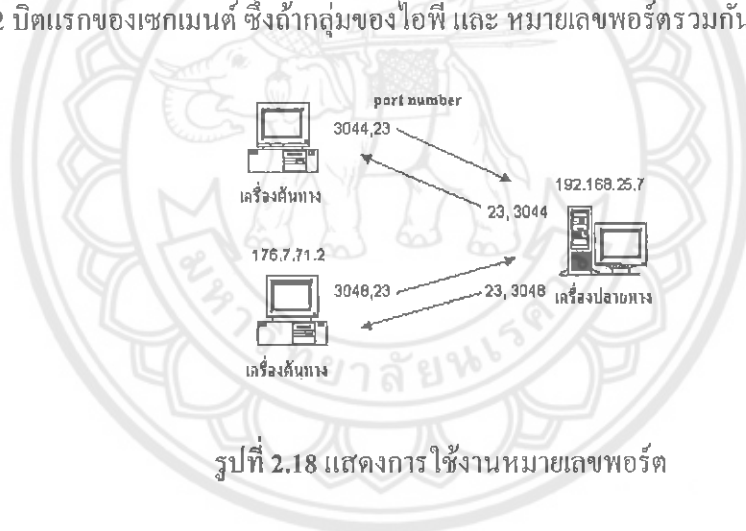
Port Number	TCP	UTP	Application Layer Services
5	TCP	UTP	Remote Job Entry
7	TCP	UDP	Echo
13	TCP	UDP	Day Time
18	TCP	UDP	Message Send Protocol
21	TCP	-	FTP Control
23	TCP	-	TELNET
25	TCP	-	SMTP
34	TCP	UDP	Remote File (RF)- used to transfer files between mail servers
37	TCP	UDP	Time
42	TCP	UDP	Name server , ARPA Host Name Server Protocol
50	TCP	UDP	Remote Mail Checking Protocol
51	TCP	UDP	IMP Logical Address Maintenance
53	TCP	UDP	Domain Name Server(DNS)
57	TCP		Mail Transfer Protocol(MTP)
66	TCP		Oracle SQL*NET
69		UDP	Day Time
70	TCP		Gopher
80	TCP	UDP	World Wide Web HTTP
107	TCP		Remote Telnet
110	TCP		Post Office Protocol(POP3)
137	TCP	UTP	NetBIOS NetBIOS Name Service

ตารางที่ 2.2 หมายเลขพอร์ต (ต่อ)

161		UDP	SNMP(Simple Network Management Protocol)
194	TCP	UDP	Internet Relay Chat (IRC)
545	TCP		OSIsoft PI(VMS), OSIsoft PI
			Server Client Access

2.8.3 การใช้งานพอร์ต

ในการส่งเซกเมนต์ ในชั้น Transport Layer จะใช้หมายเลขพอร์ตในการกำหนดเครื่องปลายทาง ในเซกเมนต์ จะประกอบไปด้วย หมายเลขพอร์ตของเครื่องต้นทางและเครื่องปลายทาง อยู่ใน 32 บิตแรกของเซกเมนต์ ซึ่งถ้ากลุ่มของไอพี และ หมายเลขพอร์ตรวมกันเราจะเรียกว่า socket



รูปที่ 2.18 แสดงการใช้งานหมายเลขพอร์ต

## 2.9 Hosts file

Hosts file เป็นไฟล์ที่ใช้เก็บค่าการจับคู่ของ ไอพี กับชื่อโฮส โดยเป็นการบ่งชี้ว่าถ้ามีการเรียกไปยังชื่อโฮสหนึ่ง ก็จะมีการวิ่งไปที่ไอพีที่มีการจับคู่ไว้ในไฟล์โฮสครับ ตัวอย่าง เช่น

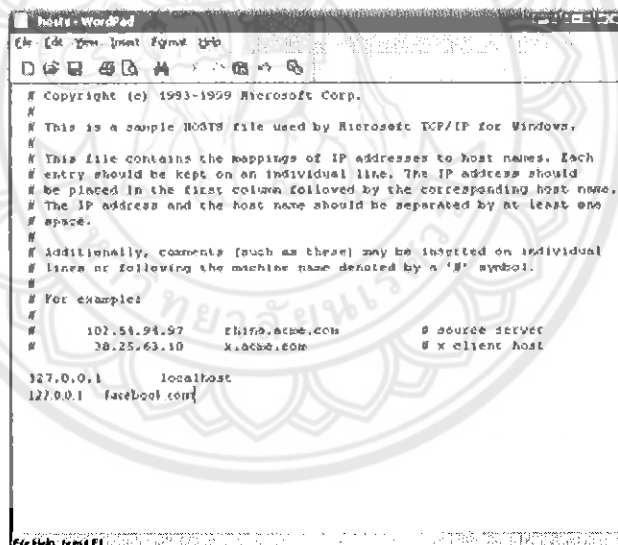
```
102.54.94.97 rhino.acme.com
```

หมายความว่าหากมีการเรียก rhino.acme.com ผ่านเว็บเบราว์เซอร์ หรือแม้แต่การ Ping ก็

ตาม ก็จะมีการวิ่งไปที่ไอพี 102.54.94.97 นั่นเอง

โดย Hosts file จะมีอยู่ที่เครื่องของเราอย่างแน่นอนที่ C:\WINDOWS\system32\drivers\etc หากเป็น ระบบปฏิบัติการ วินโดวส์อื่นๆ ที่อยู่ก็อาจจะแตกต่างกันเล็กน้อยครับ

โดยเราจะนำคุณสมบัติของ Hosts file มาใช้ประโยชน์ในการจำกัดการเข้าถึงเว็บไซต์ ตัวอย่างเช่น

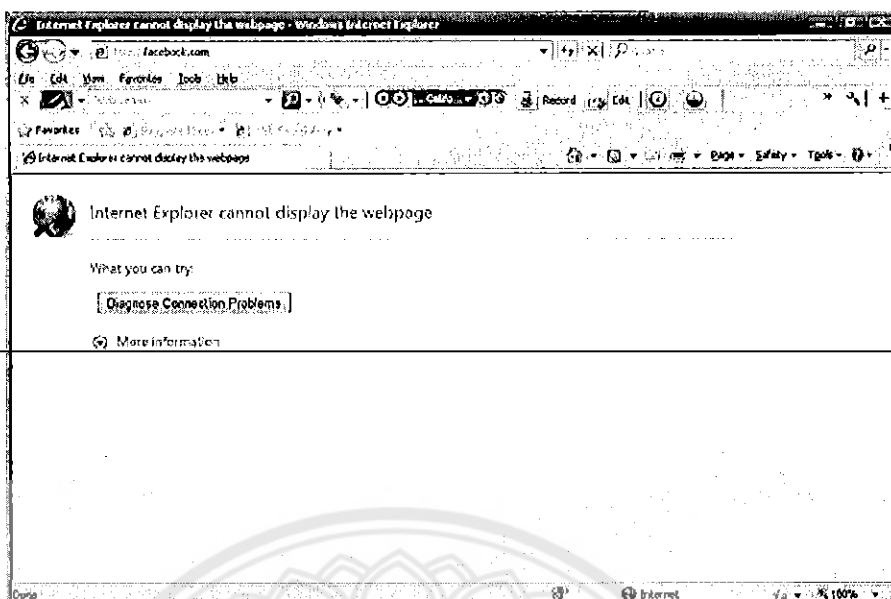


```
hosts - WordPad
File Edit View Format Help
D [Icons] [Font] [Color] [Style] [Align] [List] [View] [Print] [Exit]

# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name,
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 28.25.63.10 x.acme.com # x client host

127.0.0.1 localhost
127.0.0.1 facebook.com
```

รูปที่ 2.19 ลักษณะของ Hosts file



รูปที่ 2.20 ลักษณะของเว็บเบราว์เซอร์เข้าเว็บไม่ได้

จากตัวอย่างอธิบายได้ว่าเมื่อเราได้เพิ่มข้อความ "127.0.0.1 facebook.com" ลงไปใน Hosts file จะทำให้เมื่อเราเรียก facebook.com ผ่าน เว็บเบราว์เซอร์ ก็จะมีการเรียกไปที่ไอพีเครื่องเราแทนทำให้ไม่สามารถเข้า facebook.com ได้ครับ

## บทที่ 3

### วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงการการออกแบบและพัฒนาระบบซึ่งการพัฒนานี้ใช้ ภาษาซีชาร์ป(C#) ซึ่งมีเครื่องมือที่อำนวยความสะดวกไว้อย่างมากมายรวมไปถึง .Net Framework ซึ่งทำให้การพัฒนา มีความสะดวกเป็นอย่างมาก การพัฒนา โปรแกรมจะแบ่ง โปรแกรมเป็น 2 โปรแกรม โปรแกรมแรก จะเป็นในส่วนของแม่ข่าย และ อีกโปรแกรมจะเป็นในส่วนของลูกข่าย ซึ่งจะมีความสามารถที่ แตกต่างกันไปตามวัตถุประสงค์และขอบเขตของงาน

#### 3.1 แนวคิดในการออกแบบ

3.1.1 โปรแกรมที่เครื่องแม่ข่ายมีสิทธิที่จะเข้าถึงเครื่องลูกข่าย แต่โปรแกรมที่เครื่องลูกข่าย ไม่มีสิทธิที่จะเข้าถึงเครื่องแม่ข่าย

3.1.2 การทำงานจะแบ่งเป็น 2 โปรแกรมคือแม่ข่าย และลูกข่าย โปรแกรมต้องทำงานพร้อม กันทั้งคู่

3.1.3 การทำงานของลูกข่าย และแม่ข่าย ใช้หลักการของ Socket ซึ่งเครื่องแม่ข่ายจะต้องรอ การติดต่อเข้ามาของเครื่องลูกข่าย เมื่อมีเครื่องลูกข่ายติดต่อเข้ามาจึงจะสามารถติดต่อระหว่างกัน ได้ และการติดต่อจะทำผ่านที่อยู่ไอพี (IP Address) และพอร์ต เราสามารถสร้าง ได้หลาย Socket เพื่อ รองรับหลายงาน โดยใช้พอร์ตที่ต่างกัน

3.1.4 โปรแกรมจะปรากฏให้เห็นชัดเจนทั้งเครื่องแม่ข่ายและเครื่องลูกข่าย และจะจำกัดบาง ความสามารถที่เครื่องลูกข่าย

3.1.5 โปรแกรมเครื่องแม่ข่ายต้องทำเลือกเครื่องลูกข่ายที่ต้องการติดต่อก่อนจึงจะสามารถ จัดการกับเครื่องนั้นได้

3.1.6 การทำงานบางอย่าง ต้องรอให้โปรแกรมที่เครื่องแม่ข่ายทำงานก่อนเครื่องลูกถึงจะ ทำงานได้ เช่น การสนทนาแบบเลือกเครื่อง การรับส่งไฟล์



### 3.2 ความสามารถของโปรแกรม

#### ความสามารถของโปรแกรมเครื่องแม่ข่าย

1. โปรแกรมสามารถทราบที่อยู่ไอพี (IP Address) และ ชื่อเครื่องของลูกข่ายได้
2. โปรแกรมสามารถสนทนากับเครื่องลูกข่ายที่เลือกได้ และเครื่องลูกข่ายสามารถสนทนากับแม่ข่ายได้
3. โปรแกรมสามารถส่งข้อความเตือนไปหาเครื่องลูกข่ายทั้งหมดหรือเครื่องลูกข่ายที่เลือกได้
4. โปรแกรมสามารถสร้างห้องสนทนา และ ผู้ใช้สามารถเข้าไปสนทนาในห้องสนทนาได้
5. โปรแกรมสามารถส่งไฟล์และรับไฟล์จากเครื่องที่เลือกได้
6. โปรแกรมสามารถดูจอเครื่องลูกข่ายที่เลือกได้
7. โปรแกรมสามารถจำกัดการเข้าเว็บไซต์ที่เครื่องลูกข่ายเฉพาะที่เลือกหรือเครื่องลูกข่ายทั้งหมดได้
8. โปรแกรมสามารถสั่ง Turn off เครื่องลูกข่ายเฉพาะที่เลือกหรือเครื่องลูกข่ายทั้งหมดได้
9. โปรแกรมสามารถสั่ง Log off เครื่องลูกข่ายเฉพาะที่เลือกหรือเครื่องลูกข่ายทั้งหมดได้
10. โปรแกรมสามารถสั่ง Restart เครื่องลูกข่ายเฉพาะที่เลือกหรือเครื่องลูกข่ายได้
11. โปรแกรมสามารถสั่ง Lock Screen เครื่องลูกข่ายเฉพาะเครื่องลูกข่ายที่เลือกหรือเครื่องลูกข่ายทั้งหมดได้

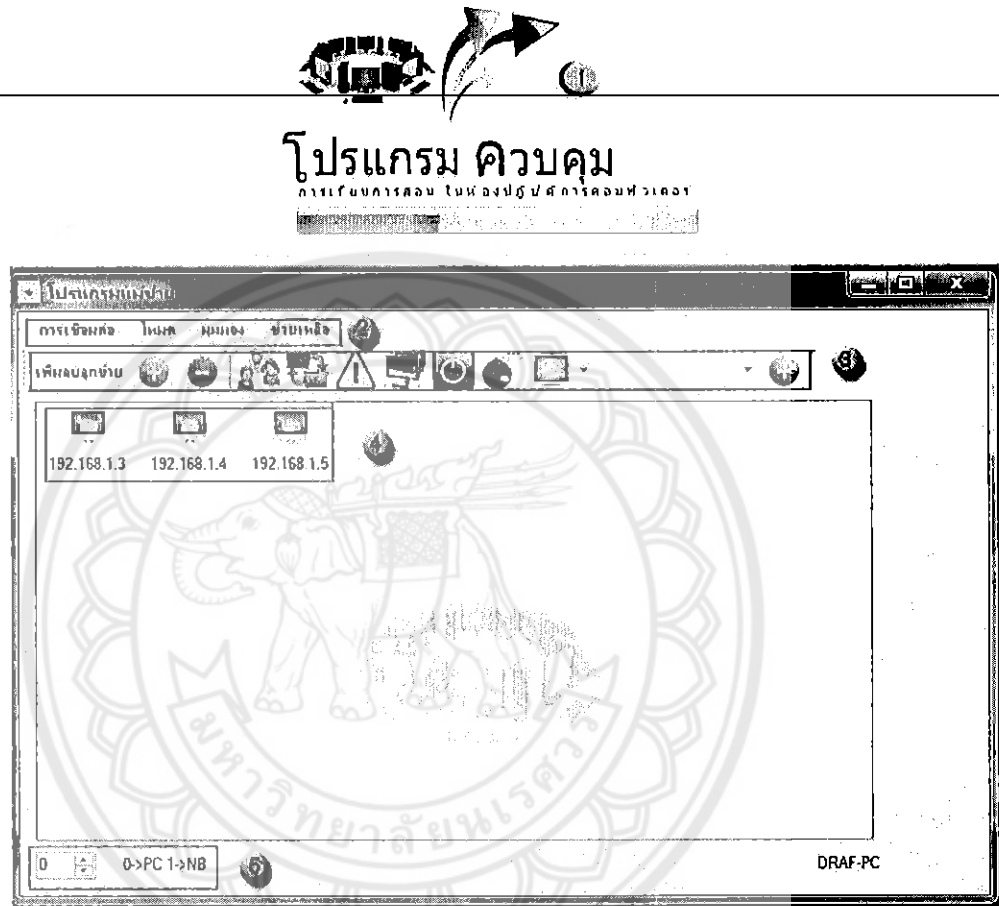
#### ความสามารถของโปรแกรมเครื่องลูกข่าย

1. โปรแกรมสามารถสนทนากับเครื่องแม่ข่ายได้ เมื่อเครื่องลูกข่ายต้องการสนทนากับเครื่องแม่ข่าย
2. โปรแกรมสามารถเข้าห้องสนทนาได้
3. โปรแกรมสามารถส่งไฟล์และรับไฟล์เมื่อเครื่องแม่ข่ายติดต่อเข้ามาได้

**หมายเหตุ** โปรแกรมเครื่องแม่ข่าย คือ โปรแกรมที่ผู้สอนใช้ในการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์ ส่วนโปรแกรมเครื่องลูกข่าย คือ โปรแกรมที่ผู้เรียนใช้ในห้องปฏิบัติการคอมพิวเตอร์

### 3.3 การออกแบบส่วนติดต่อกับผู้ใช้ (Graphic User Interface)

#### การออกแบบส่วนติดต่อกับผู้ใช้ทางฝั่งโปรแกรมแม่ข่าย



รูปที่ 3.1 หน้าตาโปรแกรมเครื่องแม่ข่ายในส่วนติดต่อกับผู้ใช้

หมายเลข 1 คือ เครื่องหมายโปรแกรมเครื่องแม่ข่าย (Logo) ของโปรแกรมควบคุมการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์ ซึ่งจะแสดงผลเมื่อเปิด โปรแกรมเครื่องแม่ข่ายขึ้นมา

หมายเลข 2 คือ ส่วนของแถบเมนู โดยมี ทั้ง การเชื่อมต่อ โหมด มุมมอง และการช่วยเหลือ ในแถบการเชื่อมต่อ ก็จะมี 2 ส่วน คือ

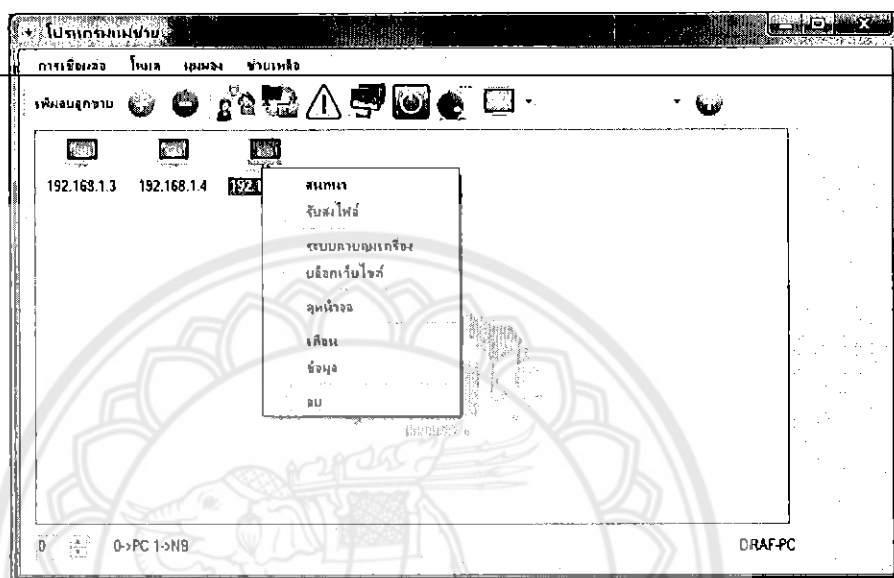
- การเพิ่มเครื่องลูกข่าย
- การลบเครื่องลูกข่าย

หมายเลข 3 แถบแจ้งข้อความเมนู ( Toolstripmenu) ซึ่งจะมีให้เลือกหลายรูปแบบ โดยที่เห็นในภาพเป็นรูปแบบในการกรอกข้อความ (text box) รูปแบบปุ่มกด (button) และรูปแบบการเลือกข้อความ (combo box) ซึ่งก็จะมีอยู่ 10 ตำแหน่ง เรียงจากซ้ายไปขวา ดังนี้

- ไอคอนเพิ่มเครื่องลูกข่าย เพื่อเพิ่มไอพีเครื่องลูกข่าย ด้วยการแปลงเป็นไอคอนรูปภาพคอมพิวเตอร์ที่ด้านล่าง เพื่อความสวยงามและอำนวยความสะดวกให้กับผู้ใช้

- 
- ไอคอนลบเครื่องลูกข่าย เพื่อลบไอคอนเครื่องลูกข่ายที่ไม่ต้องการให้ออกจาก โปรแกรม
  - ไอคอนห้องสนทนา เพื่อเข้าสู่ห้องสนทนากับเครื่องลูกข่ายหลายๆเครื่อง
  - ไอคอนรับส่งไฟล์แบบกระจาย เพื่อใช้รับส่งไฟล์ จากเครื่องหนึ่ง ไปสู่หลายๆเครื่อง
  - ไอคอนแจ้งเตือนเครื่องลูกข่าย โดยจะแจ้งเตือนเป็นกล่องข้อความแจ้งเครื่องลูกข่ายแบบกระจาย
  - ไอคอนดูหลายหน้าจอ เพื่อ ดูหน้าจอลูกข่ายได้ทีเดียหลายๆเครื่อง
  - ไอคอนระบบจัดการปิดเครื่อง เพื่อจัดการเครื่องลูกข่ายด้วยการปิดเครื่อง ปิดเครื่องแล้วเปิดใหม่ (Restart) ล็อกออฟ (Lock off) ล็อกหน้าจอ (Lock Screen) ของเครื่องลูกข่าย
  - ไอคอนแสดงการเลือกไอคอนคอมพิวเตอร์เครื่องลูกข่าย ซึ่งมีทั้ง ไอคอนแบบใหญ่ แบบเล็ก แบบรายการ
  - ไอคอนการตอบรับการสนทนาจากเครื่องลูกข่าย เมื่อเครื่องลูกข่ายต้องการสนทนากับเครื่องแม่ข่าย เครื่องแม่ข่ายต้องทำการตอบรับจากไอคอนนี้ก่อน
  - แถบการรอรับไอพีเครื่องลูกข่าย เมื่อโปรแกรมเครื่องลูกข่ายทำการรัน เราจะเจอไอพีเครื่องลูกข่ายที่แถบแสดงนี้
  - ปุ่มไอคอนเพิ่มไอพี เพื่อจะเพิ่มไอพีในส่วนที่เลือกเครื่องลูกข่ายในแถบข้อความด้านซ้าย
- หมายเลข 4 แถบไอคอนสัญลักษณ์แสดงถึงตัวแทนเครื่องลูกข่าย ที่มีการระบุไอพีกำกับไว้ในไอคอน ซึ่งสามารถเพิ่มลบได้

หมายเลข 5 ส่วนเลือกหมายเลขว่าจะใช้เลขใดในการแสดงไอคอน หากเป็นเลข 0 ก็จะแสดงรูปภาพเป็น คอมพิวเตอร์ส่วนบุคคล หากเป็นเลข 1 ก็จะแสดงรูปภาพเป็น คอมพิวเตอร์แบบพกพา



รูปที่ 3.2 รูปเมื่อคลิกขวาที่ไอคอนคอมพิวเตอร์เครื่องลูกข่าย

เมื่อเราจะส่งคำสั่งไปที่เครื่องลูกข่าย ก็จะมีอีกวิธี คือ วิธีคลิกขวาที่ไอคอนคอมพิวเตอร์เครื่องลูกข่าย ก็จะปรากฏ คำสั่งทั้งหมด 8 คำสั่ง คือ

- สถานที่, รับส่งไฟล์
- ระบบควบคุมเครื่องลูกข่าย, บล็อกเว็บไซต์
- ดูหน้าจอเครื่องลูกข่าย
- แจ้งเตือนเครื่องลูกข่าย, ดูข้อมูลเบื้องต้นของเครื่องลูกข่าย
- ลบเครื่องลูกข่าย

ซึ่งในการทดลองและการใช้งานเราจะกล่าวต่อไปในบทที่ 4

### การออกแบบส่วนติดต่อกับผู้ใช้ทางฝั่งโปรแกรมลูกข่าย

ในส่วนนี้จะเป็นส่วนที่ต้องทำหลังจากมีกล่องข้อความ (Message Box) แสดงความต้องการของเครื่องแม่ข่ายเข้ามาก่อนจึงจะสามารถทำงานได้ ยกเว้นการเข้าห้องสนทนาสามารถที่จะกดปุ่มได้เลย



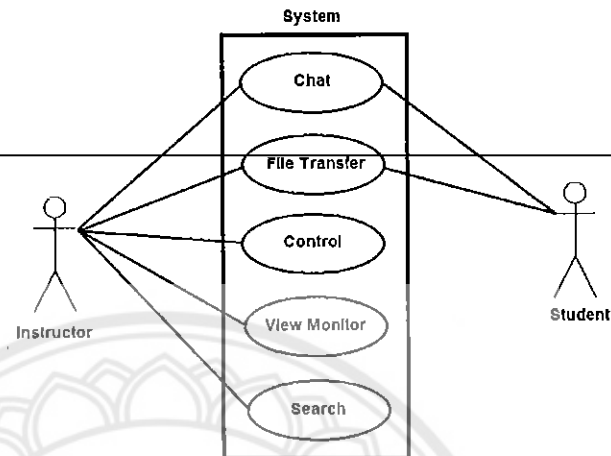
รูปที่ 3.3 หน้าตาโปรแกรมเครื่องลูกข่ายในส่วนติดต่อกับผู้ใช้

แบ่งออกเป็น 3 ปุ่ม ไอคอนการทำงาน คือ

1. ปุ่มสนทนา เมื่อเครื่องแม่ข่ายเรียกการสนทนาให้เครื่องลูกข่าย เครื่องลูกข่ายก็จะเข้าไปใช้งานในส่วนสนทนา
2. ปุ่มรับส่งไฟล์ เมื่อลูกข่ายจะทำการรับส่งไฟล์ก็จะเข้ามาใช้งานในส่วนนี้ เพื่อเลือกข้อมูลที่จะส่งและเลือกที่อยู่ที่จะรับ
3. ปุ่มห้องสนทนา ใช้งานเมื่อเครื่องลูกข่ายต้องการสนทนาแบบหลายๆเครื่อง
4. ปุ่มรับไฟล์แบบกระจาย ใช้งานเมื่อมีความต้องการของเครื่องแม่ข่ายที่จะส่งไฟล์แบบกระจายมาให้

### 3.4 การออกแบบโปรแกรมด้วยภาษา UML (Unified Modeling Language)

#### 3.4.1 การออกแบบด้วย Use case Diagram



รูปที่ 3.4 รูปแสดง Use Case Diagram ทั้งหมดของระบบ

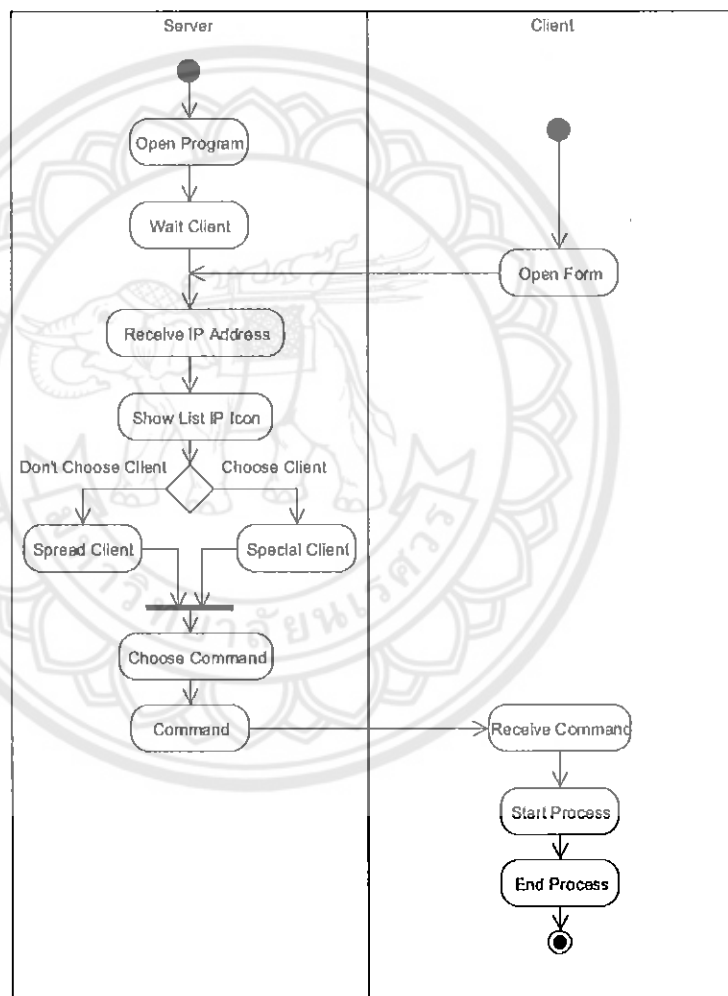
จากภาพ แสดงให้เห็นถึงการติดต่อของ ผู้ใช้กับระบบซึ่ง ผู้ใช้ทั้งสองฝั่งจะมีบทบาทที่แตกต่างกันไป จากการใช้งานของระบบจะเห็นได้ว่า ฝั่งเครื่องแม่ข่าย (Instructor) สามารถควบคุมการใช้งานพื้นฐานของ ฝั่งลูกข่าย (Student) ได้ ซึ่งแตกต่างกับฝั่งลูกข่าย จะใช้งานระบบได้เมื่อมีคำสั่งจาก ฝั่งแม่ข่าย เข้ามาและใช้งานได้เพียงบางส่วนเท่านั้น ดังการอธิบายดังนี้

1. Chat ในส่วนนี้จะมีลักษณะการทำงานที่เกี่ยวกับการสนทนาทั้งหมด การสนทนาจะเกิดขึ้นทั้งสองฝ่าย ดังนั้นการสนทนาจะทำงานกับระบบทั้งสองฝั่งของผู้ใช้งาน
2. File Transfer การทำงานของการรับส่งไฟล์จะมีลักษณะการทำงานเหมือนการสนทนา คือการทำงานของระบบจะกระทำทั้งสองฝ่าย ฝ่ายเครื่องแม่ข่ายสามารถที่จะทำการรับส่งไฟล์ ให้เครื่องลูกข่ายได้ในขณะเดียวกัน เครื่องลูกข่ายก็สามารถที่จะรับส่งไฟล์ให้เครื่องแม่ข่ายได้เช่นกัน
3. Control ในส่วนนี้การทำงานจะสั่งการ โดยเครื่องแม่ข่ายเพียงฝ่ายเดียวเท่านั้น โดยที่เครื่องลูกข่าย จะไม่สามารถที่จะสั่งการกับเครื่องแม่ข่ายได้จึงมีลักษณะการทำงานเป็นฝั่งเดียว
4. View Monitor การทำงานในส่วนนี้จะเหมือนกับ Control คือฝั่งเครื่องแม่ข่ายสามารถที่จะเห็นหน้าจอของเครื่องลูกข่ายได้เพียงฝ่ายเดียว เครื่องลูกข่ายไม่สามารถเห็นหน้าจอของเครื่องแม่ข่ายได้
5. Search การทำงานในส่วนนี้ส่วนของเครื่องแม่ข่ายสามารถรับการติดต่อ และ มองเห็นที่อยู่ไอพี ของเครื่องลูกข่ายที่ทำการติดต่อเข้ามาได้

### 3.4.2 การออกแบบด้วย Activity Diagram ในภาพรวม

การออกแบบด้วย Activity เราได้ทำการแยกการทำงานไว้ 2 แบบด้วยกันเนื่องจากมีความแตกต่างกันในการทำงานของระบบ อย่างแรกจะเป็นใช้งานระบบให้สั่งให้ โปรแกรมที่เครื่องลูกข่ายทำงานได้เลยโดยไม่ต้องมีการตอบรับการทำงานก่อน ซึ่งแตกต่างกับแบบที่ สอง ที่จะต้องรอให้สั่งโปรแกรมที่เครื่องลูกข่ายตอบรับการทำงานก่อนที่จะเริ่มทำงานได้

#### 1. การทำงานของระบบที่ไม่ต้องรอให้เครื่องลูกข่ายตอบรับการทำงาน

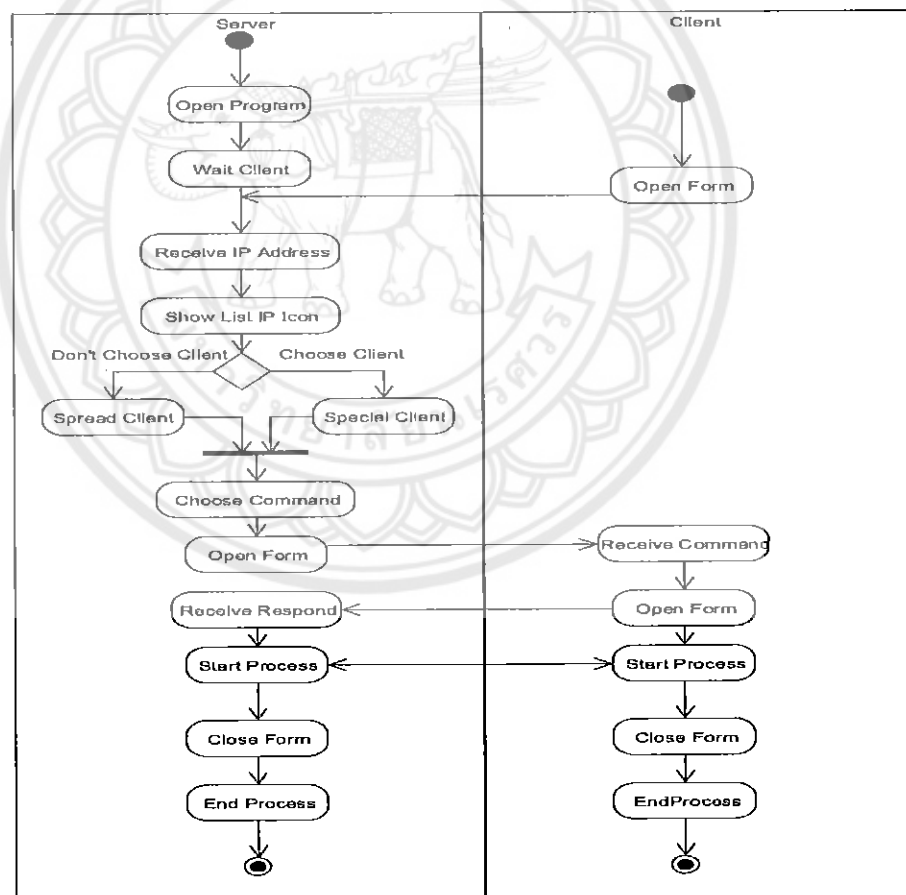


รูปที่ 3.5 Activity Diagram ของระบบที่ไม่ต้องรอให้ลูกข่ายตอบรับการทำงาน

## คำอธิบาย

จากรูปภาพแสดงให้เห็นการทำงานของเหตุการณ์ที่ไหลตามเวลา จะเห็นได้ว่าเมื่อผู้สอน (Instructor) ต้องทำการเปิดโปรแกรมก่อน จากนั้นจะรอการติดต่อของโปรแกรมที่ติดตั้งไว้ที่ลูกข่าย เมื่อโปรแกรมที่เครื่องลูกข่ายทำงานเครื่องแม่ข่ายก็จะทราบที่อยู่ไอพีของเครื่องลูกข่าย และทำการเลือกการทำงานว่าจะเลือกเป็นเครื่อง หรือจะให้ทำงานแบบทุกเครื่องเมื่อทำการเลือกเครื่องและเลือกการทำงานแล้วระบบก็จะทำงาน โดยการทำงานจะสั่งให้เครื่องลูกข่ายทำงานได้ทันทีเมื่อทราบที่อยู่ไอพีของเครื่องข่ายโดยไม่ต้องรอการติดต่อกลับมาของลูกข่ายการทำงานในลักษณะนี้ได้แก่ การปิดเครื่อง , บล็อกเว็บไซต์ , ส่งข้อความเตือน เป็นต้น

## 2. การทำงานของระบบที่ต้องรอให้เครื่องลูกข่ายตอบรับการทำงาน



รูปที่ 3.6 Activity Diagram ของระบบที่ต้องรอให้ลูกข่ายตอบรับการทำงาน



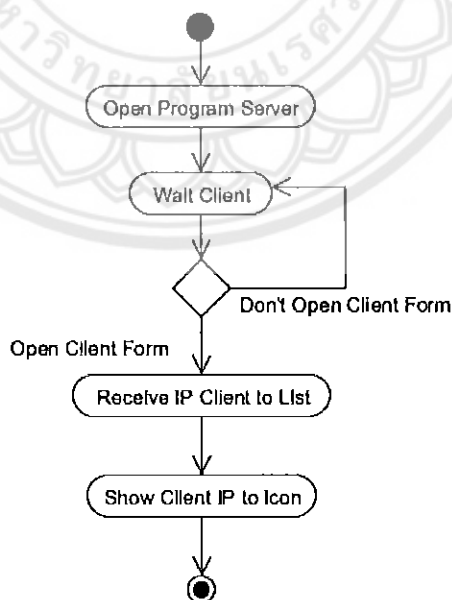
### คำอธิบาย

จะเห็นได้ว่าลักษณะการทำงานของทั้งสองระบบมีความแตกต่างกัน แต่ทั้งสองระบบต้องมีการติดต่อเข้ามาของเครื่องลูกข่ายเสมอเพื่อที่จะสามารถสั่งการทำงานต่างๆ ได้ จากแผนภาพจะเห็นได้ว่าการทำงานในช่วงแรกจะเหมือนกันแต่จะแตกต่างกัน ตรงการทำงานแบบต้องรอให้เครื่องลูกข่ายตอบรับก่อน โดยที่เครื่องแม่ข่ายจะแสดงความต้องการส่งกล่องข้อความเมื่อเครื่องลูกข่ายทราบ ว่าเครื่องแม่ข่ายต้องการจะทำกระบวนการใด จะทำการตอบรับโดยทำตามความต้องการกดปุ่มเมื่อลูกข่ายทำการตอบกลับแล้วระบบก็จะเริ่มทำกระบวนการ โดยการทำกระบวนการจะทำทั้งสองฝ่าย ไม่ว่าจะเป็นทางด้านแม่ข่ายและลูกข่าย การทำงานในลักษณะนี้เห็นได้จากการทำงาน ในการสนทนา รับส่งไฟล์ เป็นต้น

### 3.4.3 การออกแบบด้วย Activity Diagram ในแต่ละงาน

#### 1. Activity Diagram การสแกนไอพี

การทำงานของ โปรแกรมในส่วน ของ เครื่องแม่ข่าย

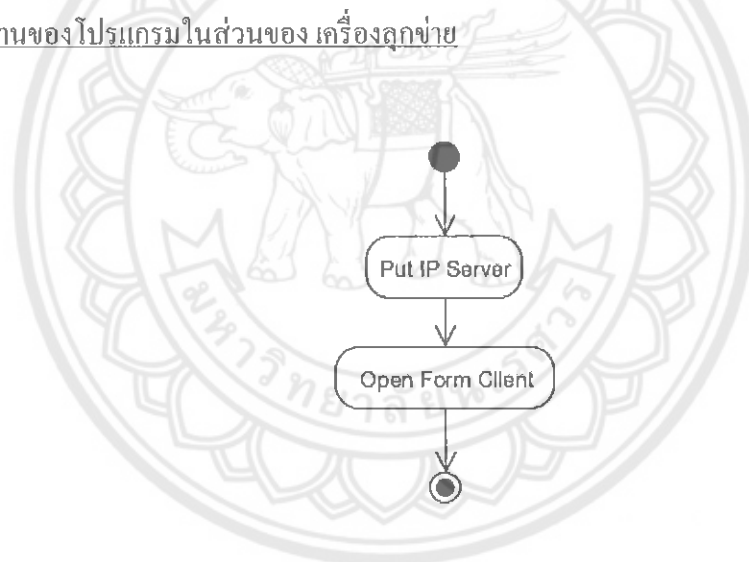


รูปที่ 3.7 Activity Diagram ของการรอรับ ไอพีของเครื่องแม่ข่าย

### คำอธิบาย

การทำงานจะเริ่มจากเปิดโปรแกรมที่เครื่องแม่ข่าย และโปรแกรมจะเปิด Socket เพื่อรอการติดต่อของเครื่องลูกข่ายเข้ามา ก่อนที่เราจะทำการรันเครื่องลูกข่ายเราต้องทำการกำหนด ไอพีเครื่องแม่ข่ายที่ลูกข่ายก่อนเพื่อให้สามารถติดต่อกับ Socket ที่ได้ทำการเปิดไว้ที่เครื่องแม่ข่ายได้ เมื่อเราใส่ ไอพีเครื่องลูกข่ายเสร็จ ก็ทำการเปิดโปรแกรมเครื่องลูกข่าย โปรแกรมก็จะทำการติดต่อไปที่ Socket ที่เปิดไว้ เมื่อการติดต่อเข้ามาเราจะสามารถทราบไอพีเครื่องที่ทำการติดต่อเข้ามาได้ เมื่อทราบไอพีของเครื่องลูกข่ายแล้วจะนำไอพีนั้นนำไปเก็บที่รายการในอาร์เรย์ เพื่อนำไปใช้งานในงานที่ต้องการส่งไปแบบกระจาย หรือนำไอพีไปเก็บในรายการแล้วจะนำไปโชว์เป็นรายการไอคอนเพื่อนำไปใช้งานกับงานที่ต้องการทำแบบเลือกเครื่อง ก็จะสิ้นสุดการหาไอพีทั้งหมดที่ทำการติดต่อเข้ามา

### การทำงานของโปรแกรมในส่วนของ เครื่องลูกข่าย



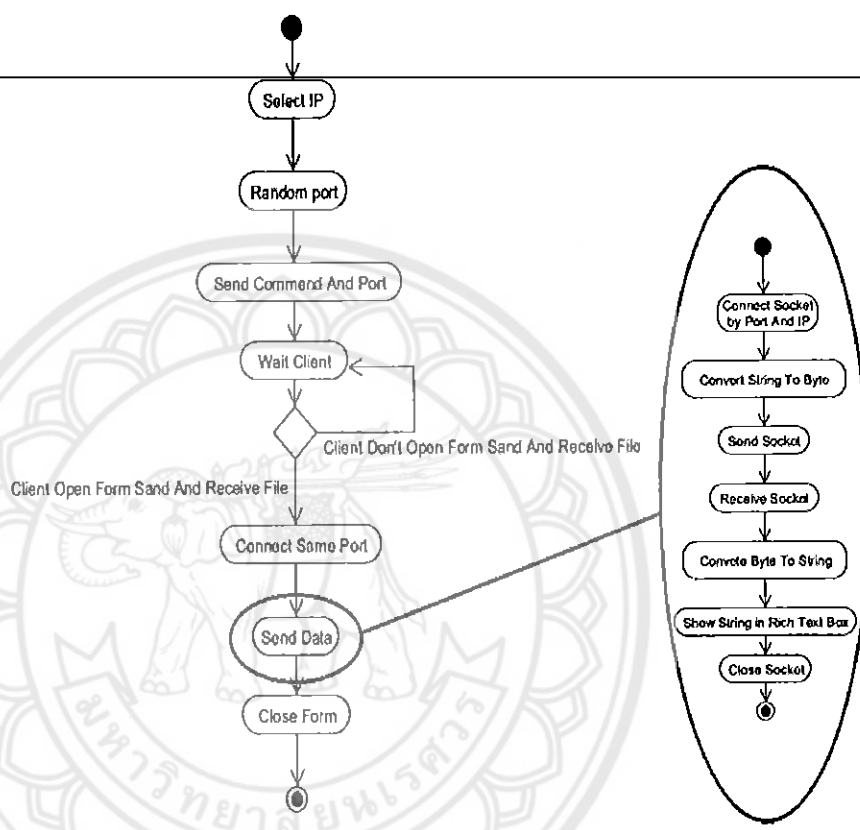
รูปที่ 3.8 Activity Diagram ของการรับ ไอพีของเครื่องลูกข่าย

### คำอธิบาย

การทำงานของเครื่องลูกข่ายในการส่งที่อยู่ไอพีไปหาเครื่องแม่ข่าย การทำงานเพียงแต่ทำการกรอกที่อยู่ไอพีของเครื่องแม่ข่าย ลงไปใน Socket เครื่องที่ต้องการติดต่อ และทำการติดต่อไปยังปลายทางนั้น เครื่องแม่ข่ายก็สามารถทราบที่อยู่ไอพีของเครื่องลูกข่ายได้ ซึ่งการกรอกไอพีในโปรแกรมจะใส่ไว้ที่ ฟอรั่ม เมื่อทำการ เปิด ฟอรั่ม ของโปรแกรมเครื่องลูกข่ายก็จะทำการติดต่อไปยัง เครื่องแม่ข่ายเป็นการสิ้นสุดการติดต่อเครื่องแม่ข่าย

## 2. Activity Diagram การสนทนาแบบทีละเครื่อง

การทำงานของโปรแกรมในส่วนของ เครื่องแม่ข่าย



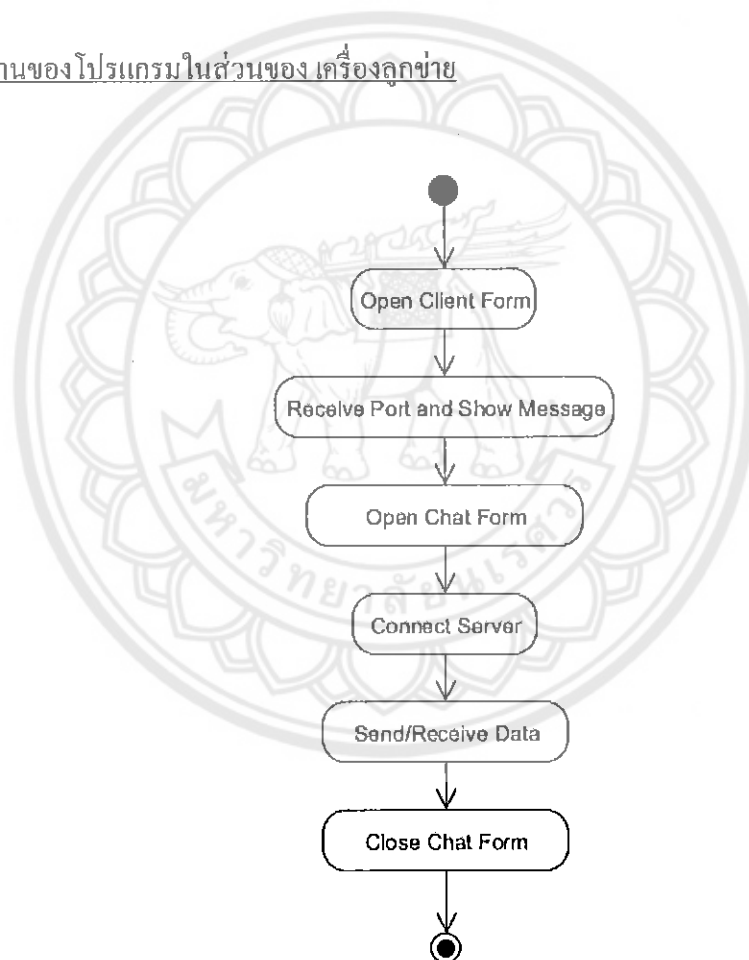
รูปที่ 3.9 Activity Diagram การสนทนาแบบทีละเครื่องของเครื่องแม่ข่าย

### คำอธิบาย

เมื่อเราทำการ เพิ่ม ไอคอนในรายการแล้วก็จะปรากฏ เครื่องที่ทำการติดต่อมาทั้งหมด เราสามารถที่จะคลิกขวาแล้วทำการ เลือกสนทนากับเครื่องนั้นได้ โดยการทำงานของมันจะเริ่มจากเราทำการเปิดฟอร์ม เมื่อเราทำการเปิดฟอร์มแล้ว เหมือนเราได้เลือกที่อยู่ไอพีของเครื่องนั้นเพื่อใช้ในการติดต่อ พร้อมกับทำการสุ่มพอร์ต ที่จะใช้ในการรับส่งข้อมูล เพื่อไม่ให้การรับส่งข้อมูลนั้นชนกันในกรณีที่มีการเลือกเครื่องอื่นด้วย เมื่อทำการเลือกไอพี และสุ่มพอร์ต แล้ว ก็จะนำพอร์ตนั้นไปสร้าง Socket เพื่อรอการติดต่อในการรับส่งข้อมูล เมื่อสร้างเสร็จก็จะส่งคำสั่งว่าต้องการที่จะ

สนทนา และ พอร์ต ไปที่เครื่องลูกข่ายนั้น โดยใช้ Socket ที่ใช้ Protocol UDP ก็จะปรากฏกล่องข้อความที่เครื่องลูกข่ายว่าเครื่องแม่ข่ายต้องการที่จะสนทนาด้วย เมื่อเครื่องลูกข่ายทราบความต้องการก็จะกดปุ่ม สนทนา เนื่องจากการติดต่อจะเริ่มเมื่อเปิดฟอร์มรับส่งไฟล์ของฝั่งลูกข่าย เมื่อเปิดฟอร์มนั้นแล้ว เราก็จะสามารถที่จะรับส่งข้อความได้ ผ่าน Socket ที่ได้สร้างใหม่ โดยการรับส่งข้อความนั้น เมื่อทำการติดต่อกับ Socket เรียบร้อยแล้ว ก็จะแปลงข้อความสตริงเป็นไบต์ เพื่อง่ายในการส่ง และในการรับก็เช่นกันเราจะได้ข้อมูลเป็นไบต์ เราต้องทำการแปลงเป็นข้อความสตริงกลับเช่นกันเมื่อได้ข้อความสตริงแล้ว เราจะนำไปแสดง เป็นการสิ้นสุดการรับส่งข้อความสตริงและการทำงานจะจบลงเมื่อ ปิดฟอร์ม

การทำงานของโปรแกรมในส่วนของ เครื่องลูกข่าย



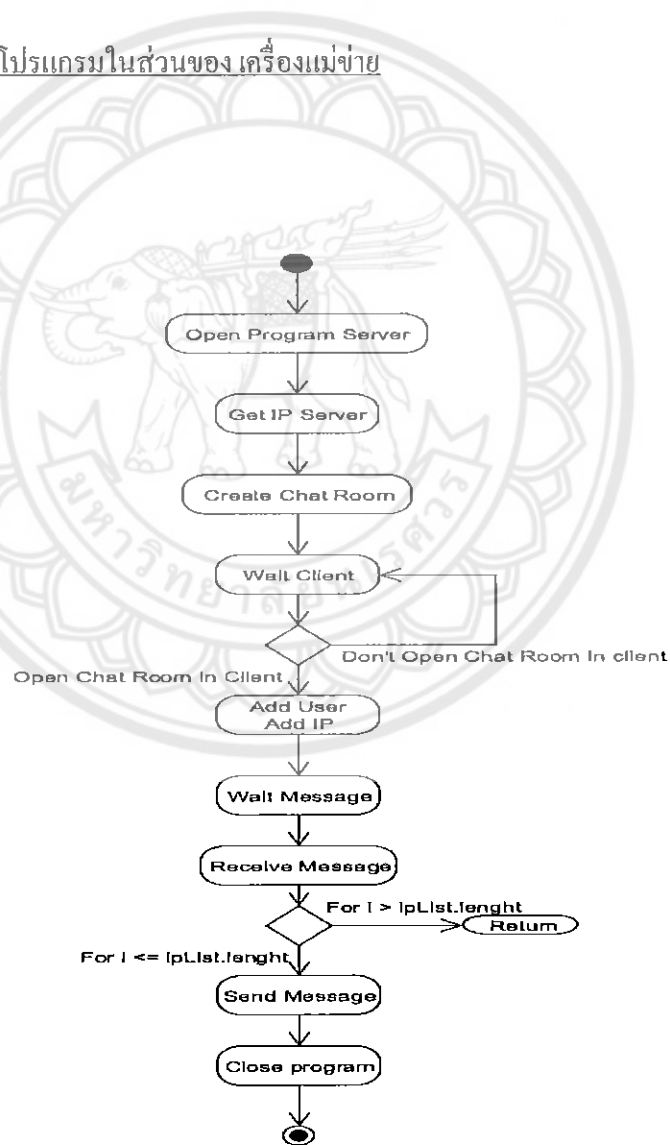
รูปที่ 3.10 Activity Diagram การสนทนาแบบทีละเครื่องของเครื่องลูกข่าย

### คำอธิบาย

การทำงานของเครื่องลูกข่ายในการสนทนา จะต้องรอให้มีการติดต่อของเครื่องแม่ข่ายเข้ามา ก่อน โดยการติดต่อเราจะสามารถทราบได้จาก มีกล่องข้อความ เข้ามาและในการติดต่อเข้ามานั้น จะมี พอร์ต แบนมาด้วยเพื่อใช้ในการสร้างการติดต่อสนทนาอีกครั้งหนึ่ง และเมื่อมี กล่องข้อความ ปรากฏข้อความแสดงความสนทนา เราก็จะทำการเปิด หน้าต่างสนทนา เป็นเสมือนการส่งการ ติดต่อกลับไป เมื่อทำการติดต่อเสร็จเราก็จะสนทนากับเครื่องแม่ข่ายได้

### 3. Activity Diagram การสนทนาแบบห้องสนทนา

การทำงานของโปรแกรมในส่วนของ เครื่องแม่ข่าย

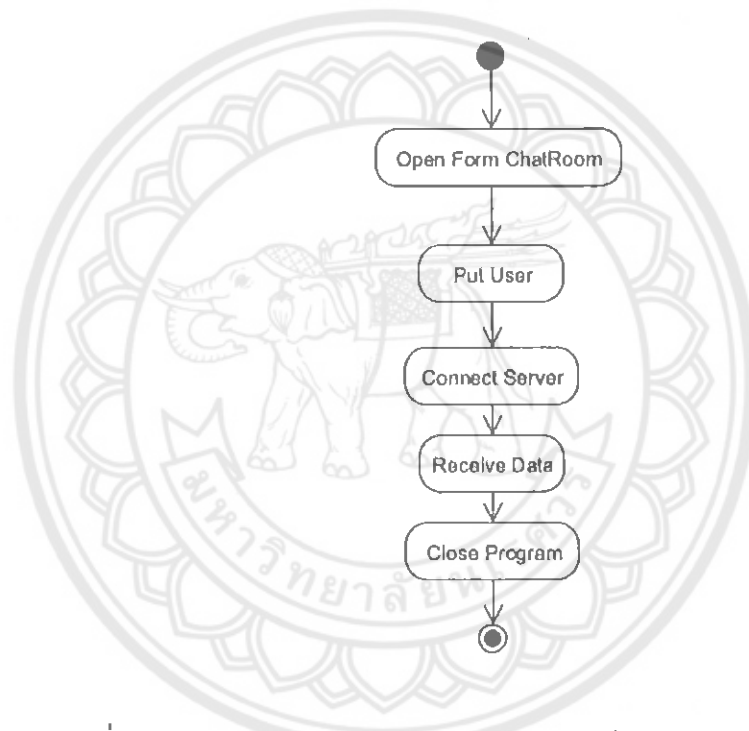


รูปที่ 3.11 Activity Diagram ของการสนทนาแบบห้องสนทนาของเครื่องแม่ข่าย

### คำอธิบาย

เมื่อเราทำการเปิดเครื่องแม่ข่ายแล้ว การสร้างห้องสนทนาจะถูกสร้างขึ้น เมื่อมีเครื่องลูกข่ายติดต่อเข้ามาก็จะทำการเพิ่มผู้ใช้ไว้ในรายการ เพื่อที่จะนำรายการนั้นมาใช้ในการรับและส่งข้อมูล ซึ่งการทำงานทั้งหมดจะทำผ่าน เครื่องแม่ข่ายแล้วให้เครื่องแม่ข่ายกระจายข้อมูลทั้งหมดไปยังเครื่องลูกข่ายที่ทำการติดต่อเข้ามาส่วนการรับและส่งข้อมูลจะมีลักษณะเหมือนการส่งไฟล์แบบกระจาย คือทำการวนในรายการจนกว่าจะหมด

### การทำงานของโปรแกรมในส่วนของ เครื่องลูกข่าย



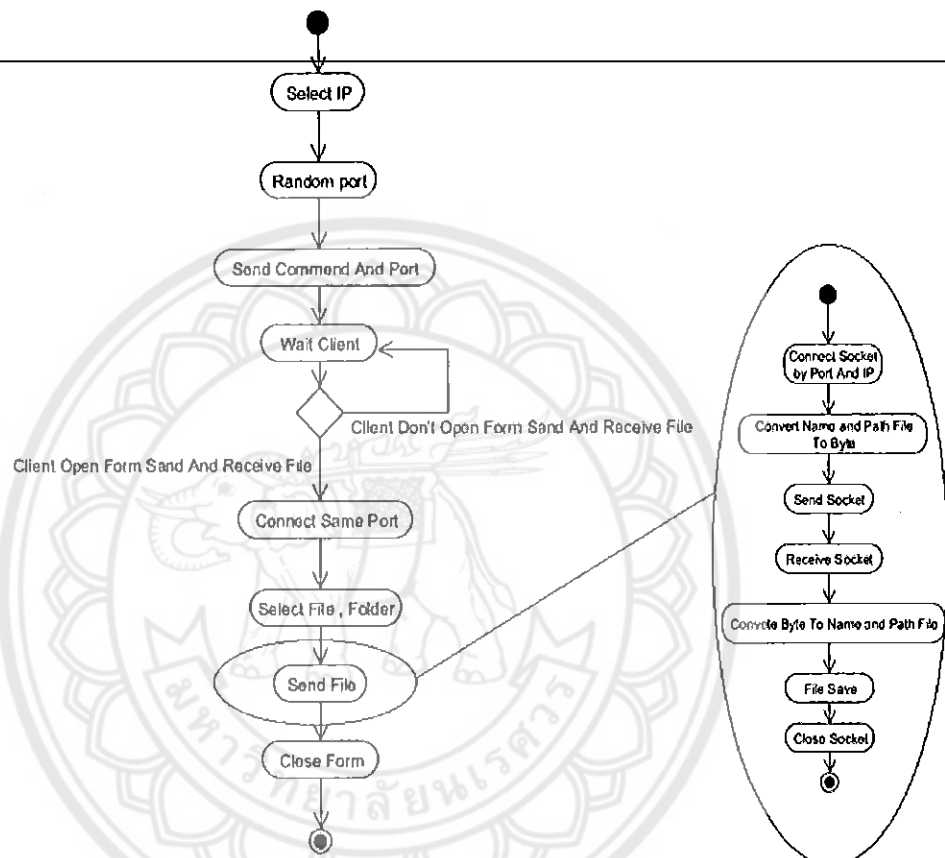
รูปที่ 3.12 Activity Diagram ของการสนทนาแบบห้องสนทนาของเครื่องลูกข่าย

### คำอธิบาย

การทำงานในการเข้าห้องสนทนาของเครื่องลูกข่ายจะเกิดขึ้นเมื่อ เปิดหน้าต่างเข้าห้องสนทนา ทำการใส่ ชื่อ และ ติดต่อไปหาเครื่องแม่ข่าย เมื่อเครื่องแม่ข่ายทราบว่าเครื่องลูกข่ายติดต่อเข้ามาก็จะทำการ ส่งข้อความไปให้เครื่องลูกข่ายว่ามีการเข้าห้องสนทนาสำเร็จ เครื่องลูกข่ายนั้นก็จะสามารถทำการสนทนาได้

#### 4. Activity Diagram การรับส่งไฟล์แบบเลือกเครื่อง

การทำงานของโปรแกรมในส่วนของ เครื่องแม่ข่าย



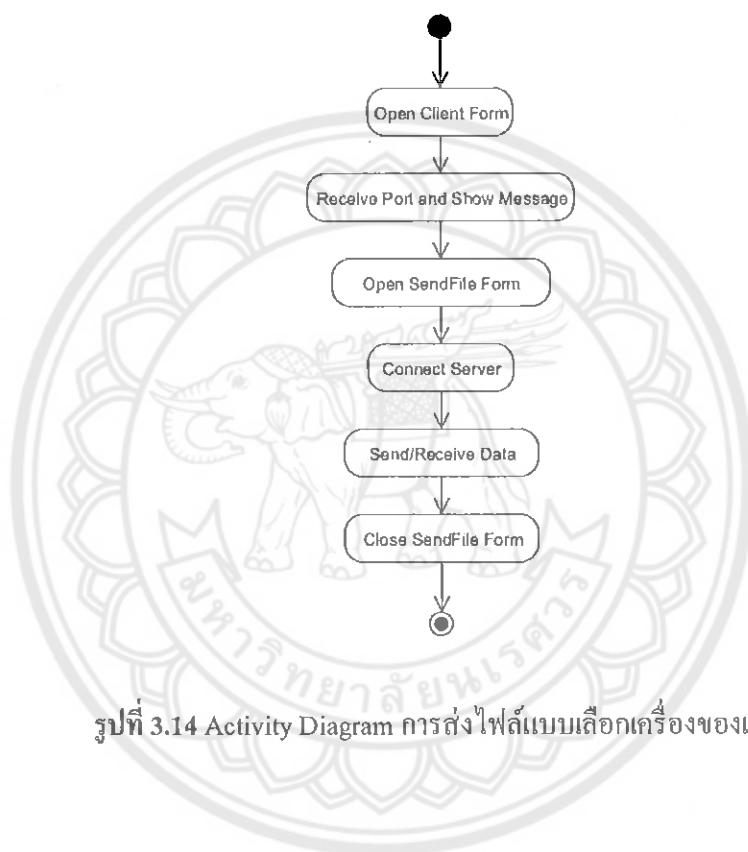
รูปที่ 3.13 Activity Diagram การส่งไฟล์แบบเลือกเครื่องของเครื่องแม่ข่าย

#### คำอธิบาย

ลักษณะการทำงานของ การรับส่งแบบเลือกเครื่องจะมีลักษณะเหมือนกับการสนทนาแบบเลือกเครื่อง ซึ่งการทำงานแบบเลือกเครื่องนี้จะนำที่อยู่ไอพีจากรายการไอคอนไปใช้งานและจะทำการติดต่อผ่านไอน์นั้น พร้อมส่งพอร์ต การใช้งานไปพร้อมกับคำสั่งว่าต้องการจะรับส่งไฟล์ และการติดต่อรับส่งไฟล์จะสำเร็จเมื่อ เครื่องลูกข่ายทำการเปิดฟอร์มรับส่งไฟล์ นั้นหมายถึง ก่อนจะส่งพอร์ตมาทางเครื่องลูกข่าย ด้านเครื่องแม่ข่ายจะทำการใช้พอร์ต นั้นรอการติดต่ออยู่แล้ว เมื่อเปิด

ฟอร์มรับส่งไฟล์ที่เครื่องลูกข่ายเหมือนเป็นการตอบรับการติดต่อ กับพอร์ต ที่เปิดรออยู่ทำให้การติดต่อสำเร็จ และจะสามารถรับส่งไฟล์กันได้ โดยการรับส่งไฟล์นี้จะเหมือนกับการรับส่งข้อมูลธรรมดาแต่จะแตกต่างกันตรงชนิดของไฟล์ ซึ่งจะต้องแปลงไฟล์ไปเป็นข้อมูลไบนารีก่อนที่จะส่ง และแปลงกลับมาเป็น ข้อมูลชนิดไฟล์ และทำการบันทึกตามเส้นทางของโฟลเดอร์ที่เลือก

### การทำงานของโปรแกรมในส่วนของ เครื่องลูกข่าย



รูปที่ 3.14 Activity Diagram การส่งไฟล์แบบเลือกเครื่องของเครื่องลูกข่าย

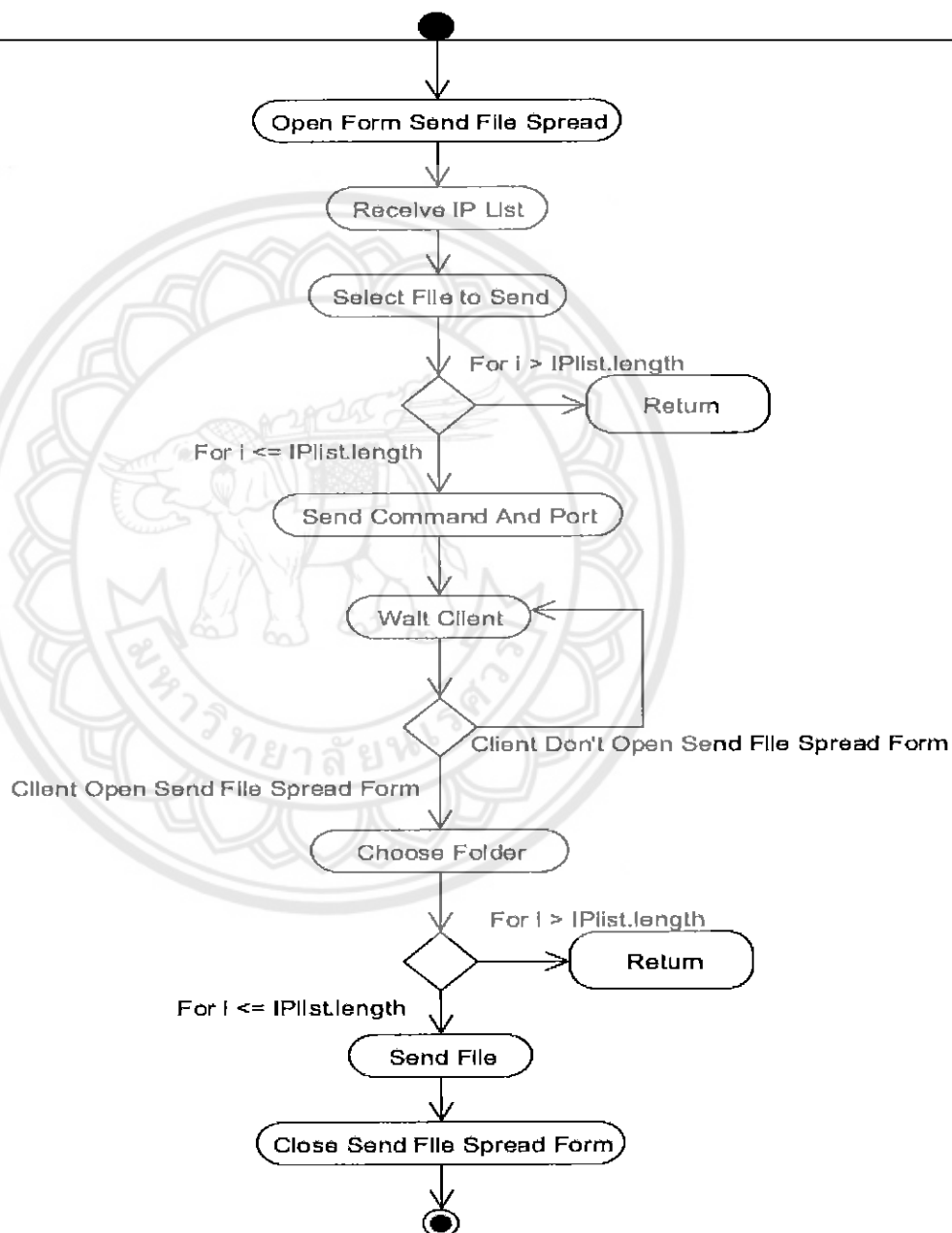
### คำอธิบาย

การทำงานของเครื่องลูกข่ายในการรับส่งไฟล์ จะต้องรอให้มีการติดต่อของเครื่องแม่ข่ายเข้ามาก่อน โดยการติดต่อเราจะสามารถทราบได้จาก มี กล่องข้อความ เข้ามาและในการติดต่อเข้ามานั้นจะมี พอร์ต แบนมาด้วยเพื่อใช้ในการสร้างการติดต่อรับส่งไฟล์อีกครั้งหนึ่ง และเมื่อมีกล่องข้อความ ปรากฏข้อความแสดงความต้องการรับส่งไฟล์ เราก็จะทำการเปิดฟอร์มรับส่งไฟล์ เป็นเสมือนการส่งการติดต่อกลับไปโดยพอร์ตที่ทำการส่งมา เมื่อทำการติดต่อเสร็จเราก็จะสามารถรับส่งไฟล์ได้



## 5. Activity Diagram ส่งไฟล์แบบกระจาย

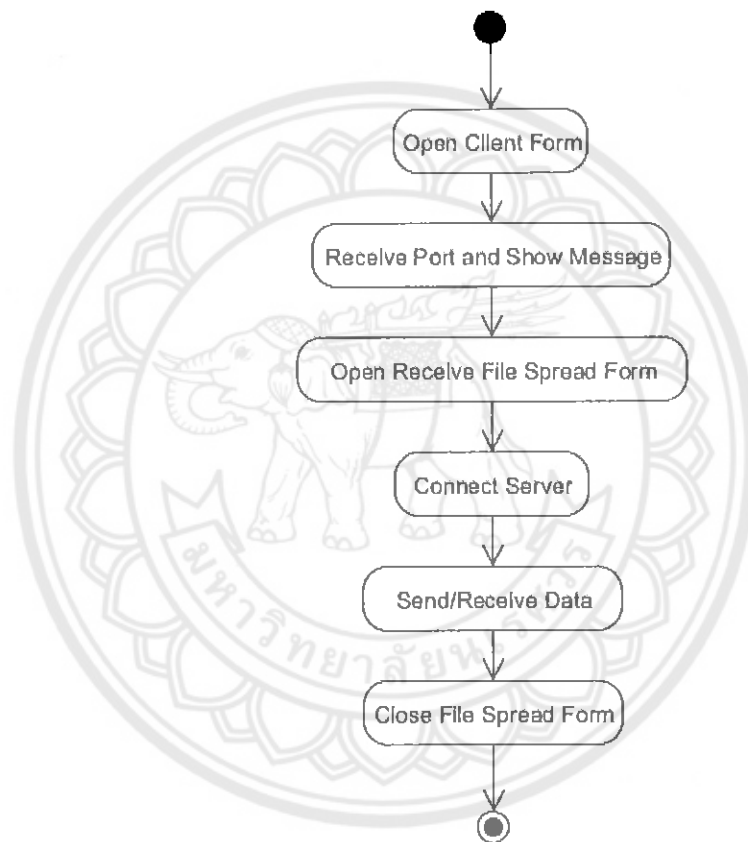
การทำงานของโปรแกรมในส่วนของ เครื่องแม่ข่าย



รูปที่ 3.15 Activity Diagram ของการส่งไฟล์แบบกระจายของเครื่องแม่ข่าย

### คำอธิบาย

การทำงานส่งไฟล์แบบกระจายจะมีลักษณะเหมือนกับส่งไฟล์แบบเลือกเครื่องแต่การส่งไฟล์แบบกระจายใช้ไอพีทั้งหมดที่ทำการติดต่อเข้า แล้วทำการวนส่งไฟล์จน ในรายการที่อยู่ของไอพีที่ว่าง ซึ่งในการส่งไฟล์แบบนี้จะทำการส่งพอร์ตให้กับทุกเครื่องก่อน โดยใช้วิธีวนลูบเช่นกันเพื่อป้องกันไม่ให้ Socket ซ้ำกัน อันนำไปสู่การทำงานที่ไม่ยืดหยุ่น เมื่อทุกเครื่องได้พอร์ต แล้วก็จะทำการเปิดพอร์ตเพื่อรอการส่งไฟล์เข้ามาและทางเครื่องแม่ข่ายจะทำการวนส่งจนครบทุกเครื่อง



รูปที่ 3.16 Activity Diagram ของการส่งไฟล์แบบกระจายของเครื่องลูกข่าย

### การทำงานของโปรแกรมในส่วนของ เครื่องลูกข่าย

การทำงานในส่วนเครื่องลูกข่ายในการส่งไฟล์แบบกระจายจะมีความทำงานเหมือนกับการรับส่งไฟล์แบบเลือกเครื่อง คือ เมื่อมีกล่องข้อความแสดงความต้องการรับส่งไฟล์ เราก็จะทำการกดปุ่ม รับไฟล์พร้อมกับนำพอร์ตที่แนบมาด้วยมาสร้างการติดต่อใหม่ และทำการติดต่อกลับไป ก็จะมารับไฟล์ที่ส่งมาจากเครื่องแม่ข่ายได้

## 6. Activity Diagram การทำงานของการส่งข้อความเตือน

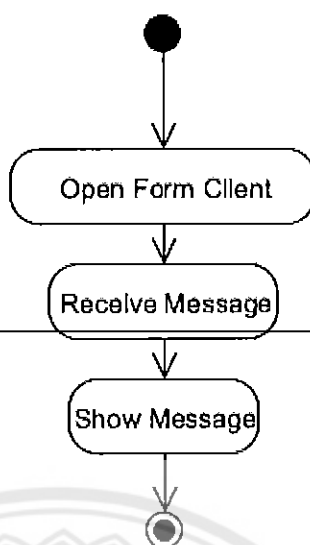
### การทำงานของโปรแกรมในส่วนของ เครื่องแม่ข่าย



รูปที่ 3.17 Activity Diagram ของการส่งข้อความเตือนของเครื่องแม่ข่าย

### คำอธิบาย

การส่งข้อความเตือนสามารถทำได้ง่ายเพราะเป็นการติดต่อทางเดียวและไม่จำเป็นต้องทำงานซับซ้อน ซึ่งเราจะใช้ Protocol UDP ในการส่ง Socket ซึ่งจะทำได้ง่ายต่อการส่ง การทำงานเริ่มจากการเลือกไอพีแล้วทำการส่งไปที่ ไอพีนั้น เป็นอันเสร็จขั้นตอนการส่ง เพราะเราได้ทำการสร้าง Socket รอรับไว้ที่เครื่องลูกข่ายแล้วเนื่องจากการใช้ UDP นี้ไม่ต้องรอให้มีการติดต่อกันก่อน จึงทำให้ไม่ต้องขั้นตอนยุ่งยากเหมือนการใช้ Protocol TCP ในการส่งข้อความเตือนทุกเครื่องก็ใช้หลักการเช่นเดียวกัน เพียงแค่วนส่งให้ครบทุกเครื่องเพียงเท่านั้นเอง



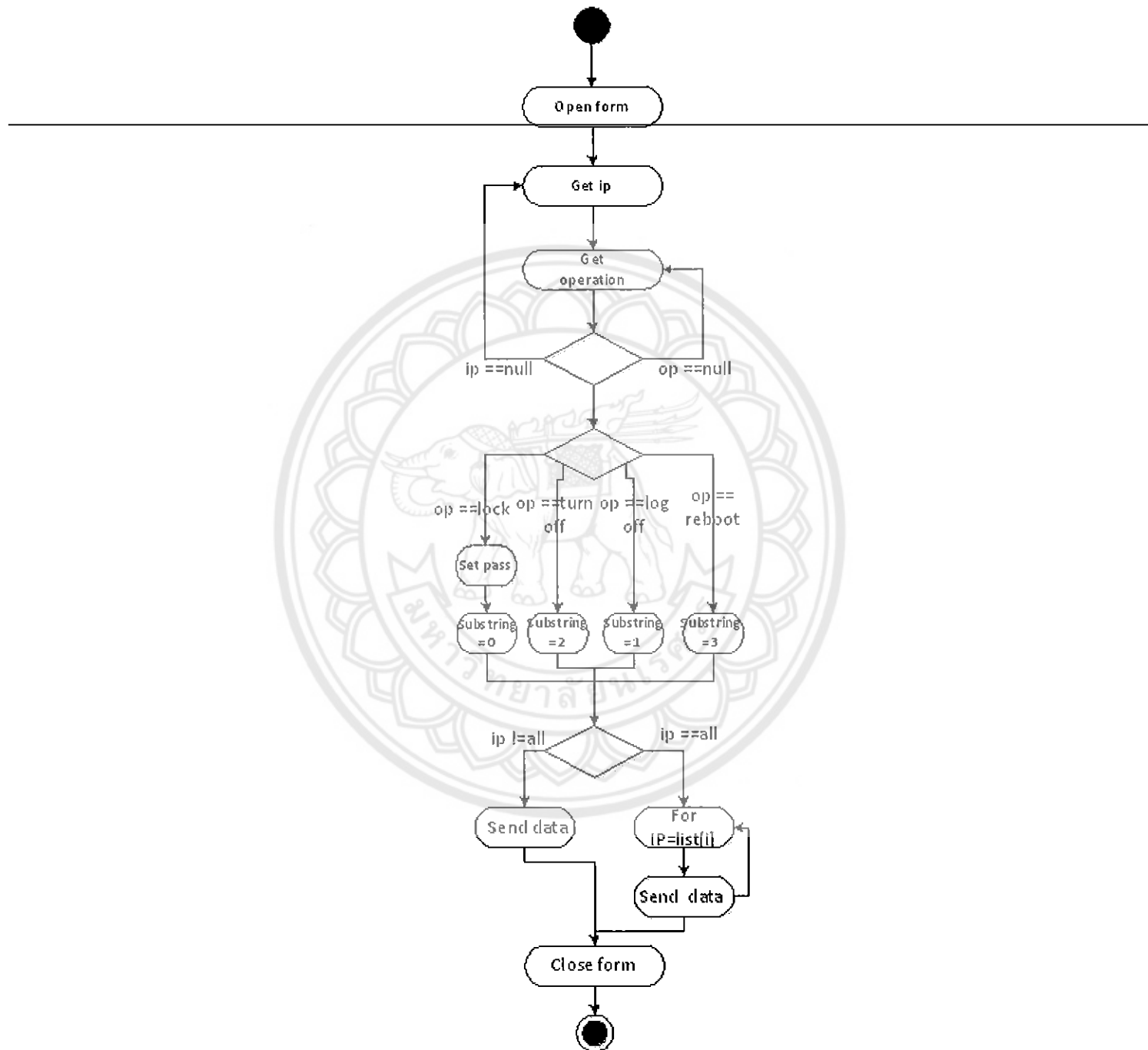
รูปที่ 3.18 Activity Diagram ของการส่งข้อความเตือนของเครื่องลูกข่าย

#### การทำงานของ โปรแกรมในส่วนของ เครื่องลูกข่าย

การแสดงข้อความเตือนในฝั่งเครื่องลูกข่าย จะทำเพียงแครับข้อมูลและนำมาแสดง ในการทำงานของการรับข้อความเตือนเพียงแค่เปิดพอร์ต และรอการส่งข้อมูลมาเพียงเท่านั้น

## 7. Activity Diagram ในส่วนของการควบคุมเครื่องถูกข่าย

การทำงานของโปรแกรมในส่วนของ เครื่องแม่ข่าย

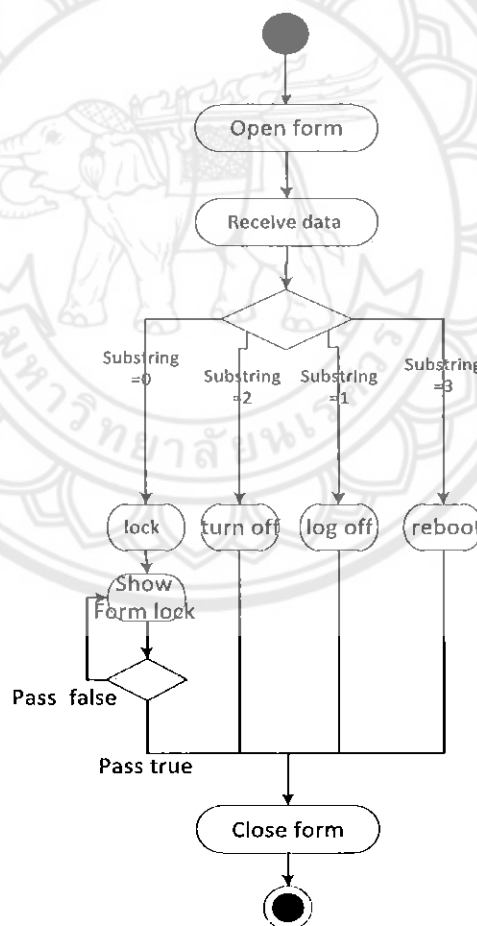


รูปที่ 3.19 Activity Diagram ของการควบคุมเครื่องในส่วนของเครื่องแม่ข่าย

### คำอธิบาย

หลักการทำงาน คือ เลือกไอพีเครื่องลูกข่ายเพื่อทำการติดต่อ ทั้งแบบเลือกติดต่อทีละเครื่อง หรือ แบบเลือกติดต่อทุกเครื่อง โดยแบบเลือกติดต่อทุกเครื่อง จะใช้การวนลูปไล่ติดต่อกับเครื่องลูกข่ายทีละเครื่อง จนครบทั้งหมดทั้งเครื่องข่าย และรับคำสั่งการทำงานจากผู้ใช้ โดยคำสั่งในส่วนของ การควบคุมเครื่องลูกข่าย จะมีอยู่คำสั่ง คือ 1. การล็อกหน้าจอ 2. การล็อกคอฟ 3. การปิดเครื่อง 4. การรีสตาร์ท เมื่อรับคำสั่งจากผู้ใช้งานแล้ว ก็จะทำการแปลงเป็นเลขฐาน แล้วทำการส่งคำสั่งไปยังเครื่องลูกข่าย โดยใช้ โพรโทคอล UDP ในการติดต่อกับเครื่องลูกข่าย แต่ในส่วนของล็อกหน้าจอ เครื่องลูกข่าย จะมีการเพิ่มในส่วนของ การตั้งรหัสผ่าน เพื่อใช้ในการปลดล็อกการทำงานเพิ่มเข้าไป

### การทำงานของ โปรแกรมส่วน เครื่องลูกข่าย



รูปที่ 3.20 Activity Diagram ของการควบคุมเครื่องในส่วนเครื่องลูกข่าย

## คำอธิบาย

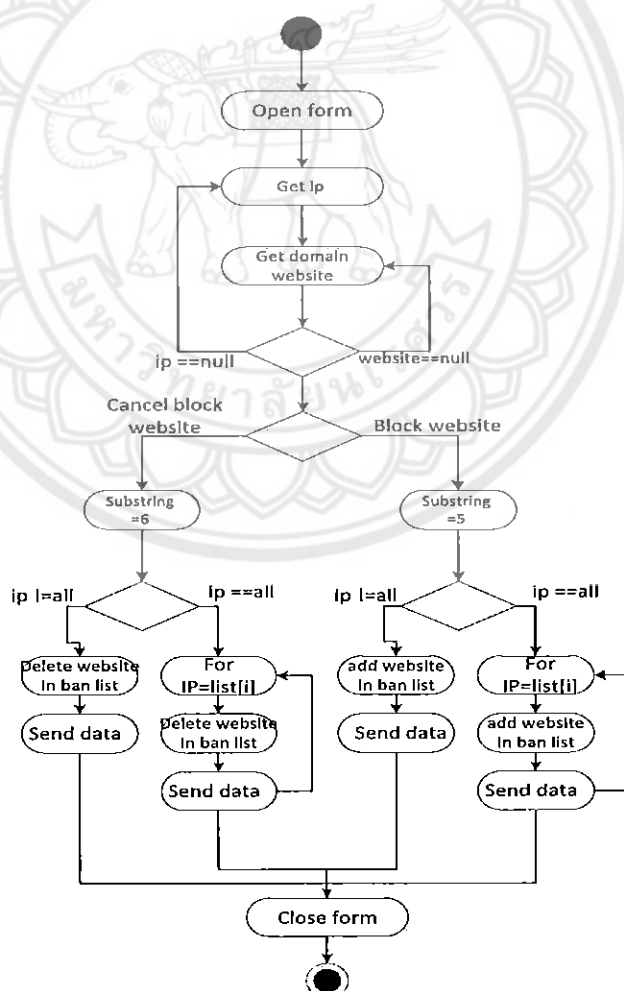
จะการใช้ API Windows ในการทำงาน โดยจะรอรับคำสั่งจากเครื่องแม่ข่าย โดยใช้โปรโตคอล UDP ในการติดต่อกับเครื่องแม่ข่าย แล้วทำการกรองคำสั่งว่าเป็นคำสั่งอะไรแล้วทำการเรียกใช้ API Windows ให้ทำงาน แต่ในส่วนของคำสั่งล๊อคหน้าจอของเครื่องลูกข่ายนั้น จะมีส่วนของการใส่รหัสผ่านเพิ่มเข้ามา คือเครื่องลูกข่ายต้องรอรับรหัสผ่านจากเครื่องแม่ข่าย หรือจาก

---

อาจารย์ผู้สอนในห้องปฏิบัติการคอมพิวเตอร์ ถึงจะสามารถกลับมาใช้งานคอมพิวเตอร์ได้อีกครั้ง

### 8. Activity Diagram ในส่วนของการจำกัดการเข้าถึงเว็บไซต์

การทำงานของโปรแกรมในส่วน เครื่องแม่ข่าย



รูปที่ 3.21 Activity Diagram ของการจำกัดการเข้าถึงเว็บไซต์ในส่วนของเครื่องแม่ข่าย

## คำอธิบาย

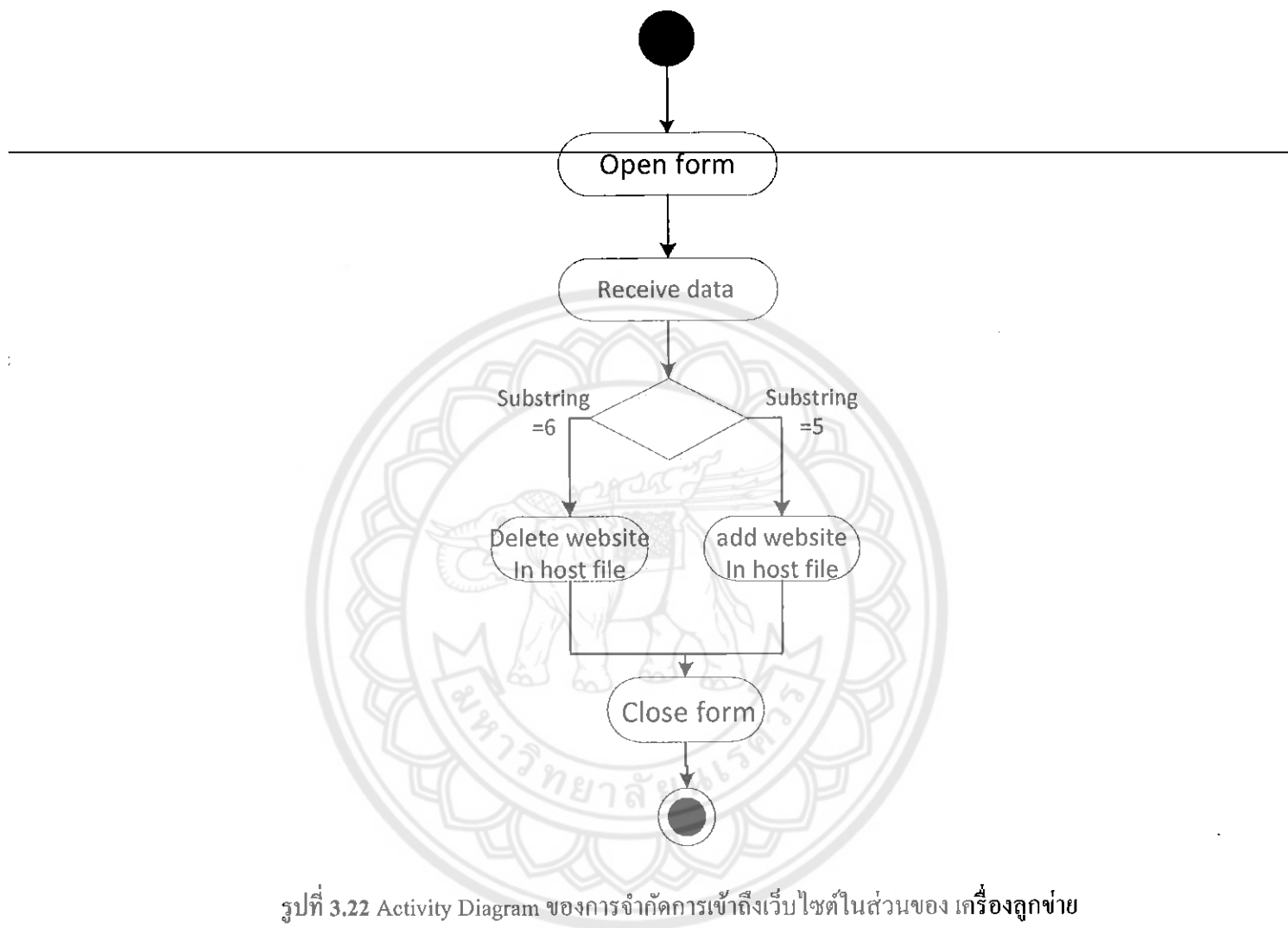
หลักการทํางาน คือ เลือกไอพีเครื่องลูกข่ายเพื่อทำการติดต่อ ทั้งแบบเลือกติดต่อทีละเครื่อง หรือ แบบเลือกติดต่อทุกเครื่อง โดยแบบเลือกติดต่อทุกเครื่อง จะใช้การวนรูปไล่ติดต่อกับเครื่องลูกข่ายทีละเครื่อง จนครบทั้งหมดทั้งเครื่องข่าย และรับชื่อ โดเมนเว็บไซต์ และรับคำสั่งทํางานจาก ผู้ใช้ โดยคำสั่งในส่วนของการจำกัดการเข้าถึงเว็บไซต์ จะมีคำสั่งอยู่ 2 คำสั่ง คือ 1. ทำการบล็อกเว็บไซต์ 2. ทำการยกเลิกการบล็อกเว็บไซต์

- การทำการบล็อกเว็บไซต์ นั้นมีการบันทึกชื่อ โดเมนเว็บไซต์ ที่ทำการบล็อกไว้ของเครื่องลูกข่ายแต่ละเครื่อง ซึ่งจะใช้แมกแอดเดรสของเครื่องลูกข่ายในการบันทึก ทำให้เมื่อลูกข่ายไอพีเปลี่ยนไปก็ยังสามารถจำเว็บไซต์ที่ได้อบล็อกไว้ได้อยู่
- การทำการยกเลิกการบล็อกเว็บไซต์ จะลบชื่อโดเมนเว็บไซต์ ออกจากรายการของโดเมนเว็บไซต์ที่ถูกบล็อก

เมื่อผู้ใช้ทำการเลือก ไอพีเครื่องลูกข่าย ชื่อโดเมนเว็บไซต์ และคำสั่งแล้ว โปรแกรมก็จะทำการส่งคำสั่งและชื่อโดเมนเว็บไซต์ไปยังเครื่องลูกข่าย โดยใช้ โปรโตคอล UDP ในการติดต่อกับเครื่องลูกข่าย



### การทำงานของโปรแกรมในส่วน เครื่องลูกข่าย



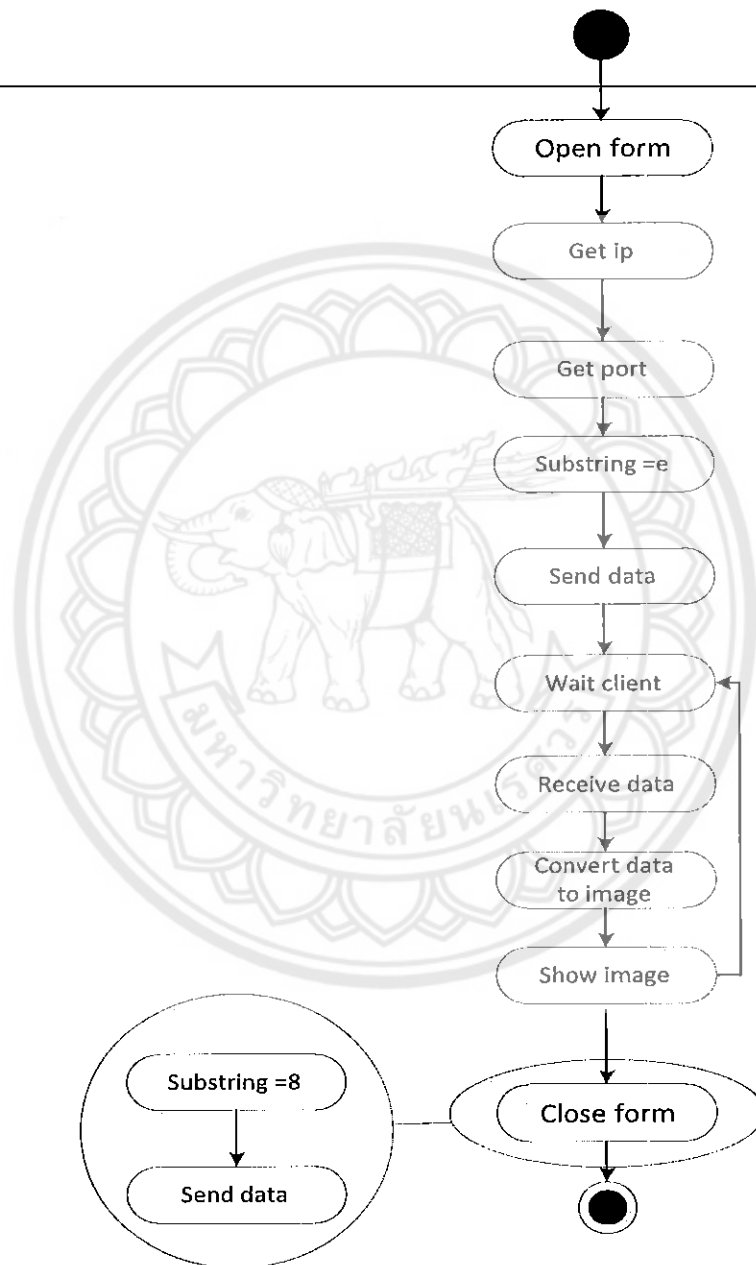
รูปที่ 3.22 Activity Diagram ของการจำกัดการเข้าถึงเว็บไซต์ในส่วนของ เครื่องลูกข่าย

### คำอธิบาย

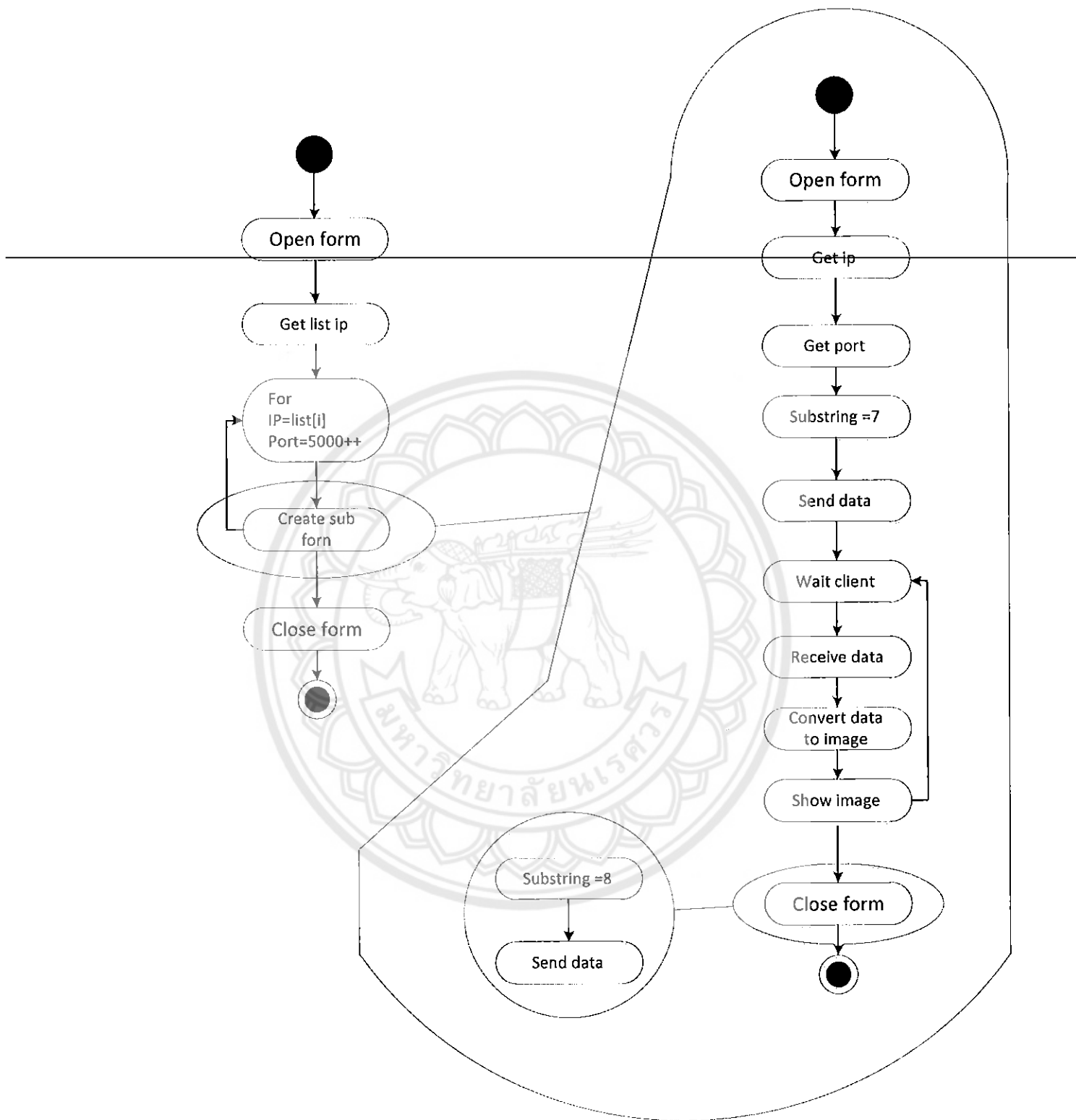
หลักการทำงาน จะรอรับคำสั่งจากเครื่องแม่ข่าย โดยใช้ โปรโตคอล UDP ในการติดต่อกับเครื่องแม่ข่าย เมื่อคำสั่งเข้ามาจะทำการกรองคำสั่งว่าเป็นคำสั่งอะไร ถ้าเป็นคำสั่งบล็อกเว็บไซต์ จะนำชื่อ โดเมนเว็บไซต์ที่ได้จะเครื่องแม่ข่าย ไปบันทึกในไฟล์โฮส ถ้าเป็นคำสั่งยกเลิกบล็อกเว็บไซต์ จะทำการลบ โดเมนเว็บไซต์ที่ได้จะเครื่องแม่ข่าย ออกจากไฟล์โฮส โดย ไฟล์โฮส ทำหน้าที่ แปลงโดเมนมายังเครื่องของตัวเอง ทำให้ไม่สามารถติดต่อไปยังที่อยู่จริงๆของเว็บไซต์ได้

## 9. Activity Diagram ในส่วนของการดูหน้าจอเครื่องถ่ายภาพ

การทำงานของโปรแกรมในส่วน เครื่องแม่ข่าย



รูปที่ 3.23 Activity Diagram การดูหน้าจอแบบจอกเดียว (จอกใหญ่) ฟัง เครื่องแม่ข่าย



รูปที่ 3.24 Activity Diagram การดูหน้าจอแบบหลายจอ ฟังก์ชันเครื่องแม่ข่าย

## คำอธิบาย

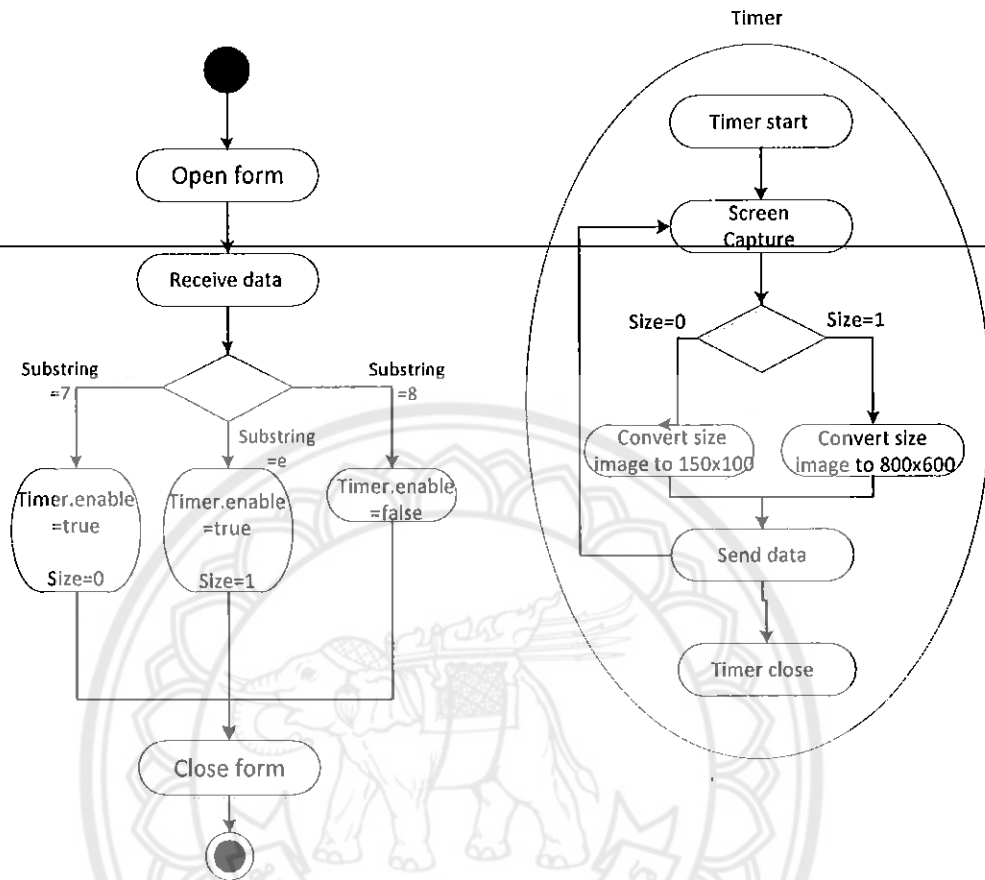
จะมีหลักการทำงานอยู่ 2 แบบ คือ 1. การดูหน้าจอบางจอเดียว ( ดูจอใหญ่ ) 2. การดูหน้าจอบางหลายเครื่อง

-การดูหน้าจอบางจอเดียว ( ดูจอใหญ่ ) หลักการทำงาน คือ จะส่งคำสั่งไปยังเครื่องลูกข่าย โดยส่วนนี้จะใช้โปรโตคอล UDP ในการติดต่อกับเครื่องลูกข่าย แล้วรอรับข้อมูลจากเครื่องลูกข่าย โดยส่วนนี้จะใช้โปรโตคอล TCP ในการติดต่อกับเครื่องลูกข่าย จากนั้นจะทำการแปลงข้อมูลเป็นรูปภาพแล้วทำการแสดงออกมา

-การดูหน้าจอบางหลายเครื่อง มีการใช้ หลักการฟอร์มในฟอร์ม โดยจะมี ฟอร์มหลัก ทำการรัน แจกไอพี และแจกหมายเลขพอร์ต ให้กับ ฟอร์มรอง ส่วนฟอร์มรอง จะมีหลักการทำงาน คล้ายกับการดูหน้าจอบางจอเดียว ( ดูจอใหญ่ )



### การทำงานของโปรแกรมในส่วน เครื่องลูกข่าย



รูปที่ 3.25 Activity Diagram การดูหน้าจอ ผัง เครื่องลูกข่าย

### คำอธิบาย

หลักการทำงาน จะรอรับคำสั่งจากเครื่องแม่ข่าย โดยใช้ โปรโตคอล UDP ในการติดต่อกับเครื่องแม่ข่าย เมื่อคำสั่งเข้ามาจะทำการกรองคำสั่งว่าเป็นคำสั่งอะไร โดยหลักการทำงานคือ จะให้ timer ทำงาน เมื่อครบ 2 วินาที จะให้ทำการจับหน้าจอ แล้วทำการแปลงขนาดรูปภาพ ซึ่ง ถ้าเป็นคำสั่งดูหน้าจอแบบจอเดียว ( จอจอใหญ่ ) จะให้ ขนาดเท่ากับ 800x600 ส่วนถ้าเป็นคำสั่งการดูหน้าจอแบบหลายเครื่อง จะให้ ขนาดเท่ากับ 150x100 จากนั้นทำการแปลงรูปภาพเป็นเลขฐานแล้วส่งออกไป ใช้ โปรโตคอล TCP ในการติดต่อกับเครื่องแม่ข่าย

### 3.5 การออกแบบพอร์ตการใช้งาน

เนื่องจากการใช้งาน ต้องมีการทำงานที่มีความแตกต่างกันซึ่งในงานแต่ละงานนั้นจะต้องทำผ่านพอร์ตแสดงความต้องการของแอปพลิเคชัน นั้นๆ ทำให้ต้องการใช้งานพอร์ตเป็นจำนวนมาก เราจึงต้องเห็นความสำคัญในการออกแบบการทำงานของพอร์ตเพื่อป้องกันไม่ให้งานของซ้ำกันหรือเพื่อลดความซับซ้อนในการใช้งาน เราจึงได้ทำการกำหนดพอร์ตของแต่ละงานไว้แต่ละงาน ดังตารางต่อไปนี้

ตารางที่ 3.1 การใช้งานพอร์ตของโปรแกรม

Feature	Protocol	Port	Substring	Work
สนทนาแบบเลือกเครื่อง	UDP	9000	9	ใช้ส่งข้อความต้องการในการสนทนา
	TCP	11000-11500	no	รับส่งข้อความ
สนทนาแบบลูกข่ายติดต่อมา	UDP	9005	1	ใช้ส่งข้อความต้องการในการสนทนา
	UDP	12000-12500	no	ใช้ส่งข้อความ
	UDP	12501-13000	no	ใช้รับข้อความ
รับส่งไฟล์	UDP	9000	\$	ใช้ส่งข้อความต้องการรับส่งไฟล์
	TCP	6000-6499	no	ใช้รับไฟล์
	TCP	6500-6999	no	ใช้ส่งไฟล์
ส่งไฟล์แบบกระจาย	UDP	9000	%	ใช้ส่งข้อความต้องการส่งไฟล์กระจาย
	TCP	10000-10500	no	ใช้ส่งไฟล์
ส่งเตือน	UDP	9000	^	ใช้ส่งเตือน
ควบคุมเครื่อง	UDP	9000	0	Lock Work Station
	UDP	9000	1	Log off
	UDP	9000	2	Turn off
	UDP	9000	3	Restart
บล็อกเว็บ	UDP	9000	5	บล็อกเว็บ
	UDP	9000	6	ยกเลิกบล็อกเว็บ

ตารางที่ 3.1 การใช้งานพอร์ตของโปรแกรม (ต่อ)

ดูจอลูกข่าย	UDP	9000	7	ใช้ดูเครื่องลูกข่าย
	UDP	9000	8	ใช้ดูเครื่องลูกข่ายแบบ Zoom
	TCP	ตั้งแต่ 5000 ขึ้นไป (ตาม จำนวนเครื่อง ลูกข่ายที่ดู)	no	ใช้ดูเครื่องลูกข่ายหลายเครื่อง
	TCP	4000	no	ใช้ดูเครื่องลูกข่ายทีละเครื่อง(จอใหญ่)
ห้องสนทนา	TCP	1986	no	เข้าห้องสนทนา
สแกน ไอพี	TCP	8003	no	ส่งการติดต่อ



## บทที่ 4

### การทดสอบและผลการทดสอบ

#### 4.1 แผนการทดสอบโปรแกรม

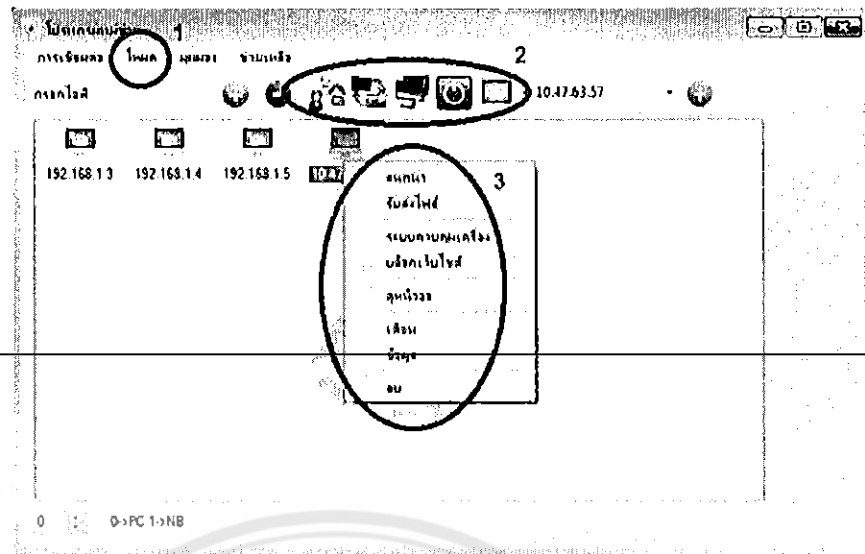
ตารางที่ 4.1 ตารางแผนการทดสอบโปรแกรม

ลำดับการทดสอบ	ชื่อการทดสอบ
1	การค้นหาที่อยู่ไอพีเครื่องลูกข่าย
2	การสนทนาระหว่างเครื่องแม่ข่ายกับเครื่องลูกข่ายแบบหนึ่งต่อหนึ่ง
3	การสนทนาในห้องสนทนา
4	การเตือนเครื่องลูกข่ายด้วยข้อความ
5	การรับและส่งไฟล์
6	การรับและส่งไฟล์แบบกระจาย
7	การปิดเครื่อง รีสดาร์ท ถี้อคออฟ ถี้อคหน้าจอ
8	การหยุดการทำงานของเว็บไซต์
9	การดูหน้าจอเครื่องลูกข่าย
10	การดูหน้าจอเครื่องลูกข่ายหลายๆเครื่อง
11	การดูข้อมูลเบื้องต้นของเครื่องลูกข่าย

ตามตารางนี้ มีการวางแผนการทดสอบกับ โปรแกรมเครื่องแม่ข่าย (Server) และ โปรแกรมเครื่องลูกข่าย (Client) ซึ่งเป็นการสื่อสารในวงแลนเดียวกัน เพื่อใช้ในการเรียนการสอนในห้องปฏิบัติการคอมพิวเตอร์ ในตารางแผนการทดสอบโปรแกรมนี้ได้นำไปใช้ทดสอบในห้องปฏิบัติการคอมพิวเตอร์จริง ใช้เครื่องคอมพิวเตอร์ลูกข่ายทั้งหมด 10 เครื่อง เครื่องแม่ข่าย 1 เครื่อง โดยมีการลงโปรแกรมที่ใช้กับเครื่องแม่ข่าย ที่เครื่องแม่ข่าย 1 เครื่อง และลงโปรแกรมที่ใช้เป็นเครื่องลูกข่าย ที่เครื่องลูกข่ายทั้งหมด 10 เครื่อง เพื่อทดสอบการทำงานของระบบ

ในการทดสอบโปรแกรมทางด้าน เครื่องแม่ข่ายจะต้องเปิดโปรแกรมก่อน และทำการเพิ่มเครื่องลูกข่ายเข้าไปในรายการ ถึงจะสามารถที่จะใช้คำสั่งต่างๆ ได้โดยการคลิกขวาหรือแถบด้านบน เพื่อทดสอบการทำงานต่างๆ ของ โปรแกรมดังรูปที่ 4.1





รูปที่ 4.1 หน้าจอการทดสอบ โปรแกรมฝังเครื่องแม่ข่าย

ส่วนทางด้านเครื่องลูกข่ายจะทำการเปิด โปรแกรมเครื่องลูกข่ายและรอความต้องการของเครื่องแม่ข่ายว่าต้องการจะทำงานอะไร โดยจะแสดงเป็นกล่องข้อความและเครื่องลูกข่ายจะตอบรับคำสั่ง โดยการกดปุ่มที่แสดงในรูปที่ 4.2

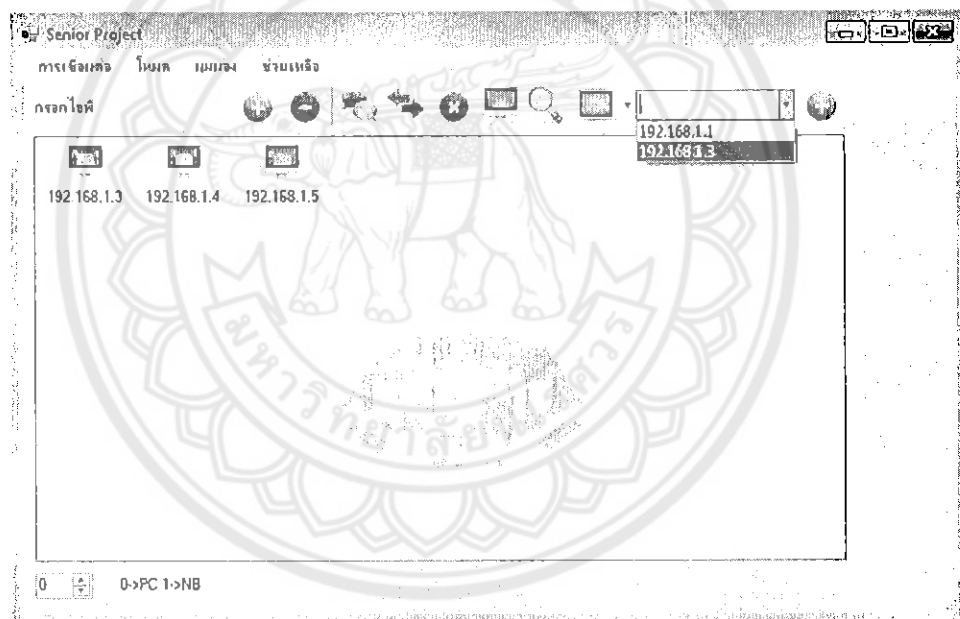


รูปที่ 4.2 หน้าจอการทดสอบ โปรแกรมฝังเครื่องลูกข่าย

## 4.2 การทดสอบโปรแกรม

### 4.2.1 การทดสอบรอรับ IP Address เครื่องลูกข่าย

การที่เราจะเห็นที่อยู่ไอพี (IP Address) ของเครื่องลูกข่าย เครื่องลูกข่ายจะต้องทำการเปิดโปรแกรมก่อน เพื่อให้เครื่องแม่ข่ายเห็นไอพี ในการทำงานนี้ ทางด้าน เครื่องลูกข่ายจะใช้ Socket ในการส่งข้อมูลต่างๆ ไปยัง เครื่องแม่ข่าย ทาง เครื่องแม่ข่าย จะสามารถดูไอพีของเครื่องลูกข่ายที่ทำการติดต่อมา และนำไปแสดงในรายการในกรณีที่มีการติดต่อเข้ามาของเครื่องลูกข่าย หลายๆ ตัว ทาง เครื่องแม่ข่าย จะต้องสร้างตัวแปรงานในการรองรับการติดต่อเหล่านั้นให้แยกการทำงานออกจากกันในแต่ละเครื่อง เพื่อให้สามารถรับการติดต่อได้พร้อมกันหลายเครื่อง จากการทดสอบจะปรากฏไอพีของเครื่องลูกข่าย ที่ทำการติดต่อมาแสดงเป็นรายการ ดังรูป 4.3



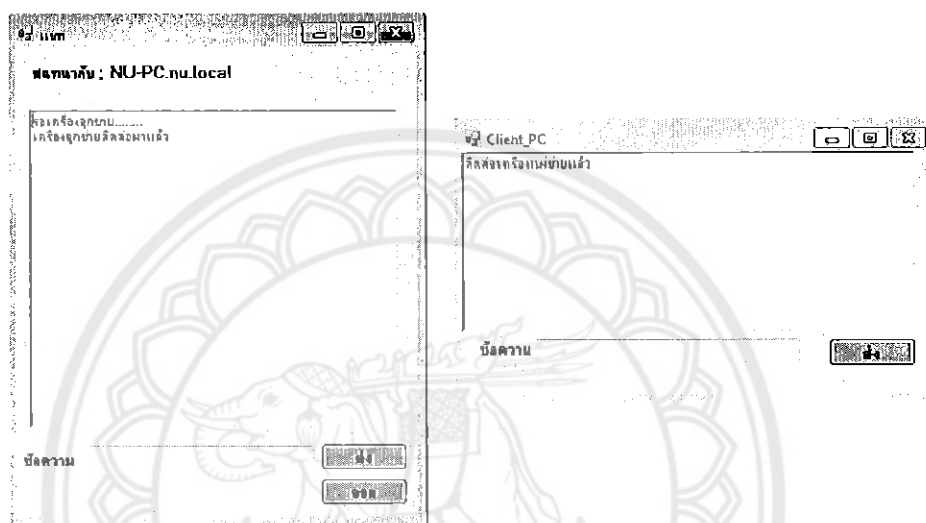
รูปที่ 4.3 หน้าจอการรอรับไอพีจากเครื่องลูกข่าย

### 4.2.2 การทดสอบสนทนาระหว่างเครื่องแม่ข่ายกับเครื่องลูกข่ายแบบหนึ่งต่อหนึ่ง

การส่งข้อมูลข้ามกันระหว่างเครื่องจำเป็นต้องมี Socket ทั้งสองฝั่งเพื่อที่จะใช้ในการรอรับข้อมูล และ ใช้ในการส่งข้อมูลระหว่างกัน ข้อมูลที่ใช้ส่งจะเป็นข้อมูลชนิดไบต์ดังนั้นเราต้องทำการแปลงจากสตริงเป็นไบต์เพื่อให้ส่งต่อการส่ง ส่วนในการทดสอบเครื่องแม่ข่ายต้องทำการเลือกเครื่องลูกข่ายที่ต้องการสนทนา และคลิกคำสั่งสนทนา เมื่อคลิกแล้ว จะปรากฏกล่องข้อความที่

เครื่องลูกข่าย ลูกข่ายทำการจะทำการตอบรับก่อน โดยการกดปุ่มสนทนา เพื่อตอบสนองความต้องการของเครื่องแม่ข่าย จากนั้นทั้งสองฝ่ายจะสามารถสนทนากันได้

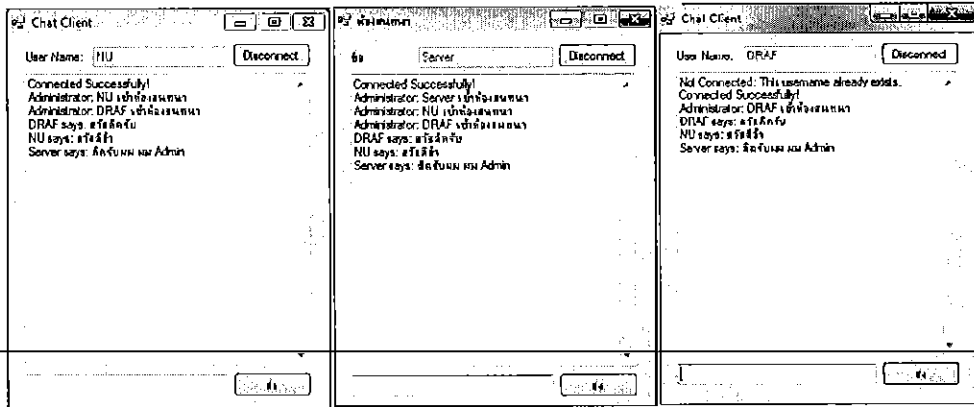
เมื่อทางลูกข่ายตอบรับจากที่มีกล่องข้อความเข้ามา แล้วกดปุ่มสนทนา จะแสดงหน้าต่างดังรูป 4.5 จากการทดสอบปรากฏว่าสามารถที่จะส่งข้อความหากันได้ทั้งสองฝ่าย และสามารถเลือกเครื่องอื่นสนทนาได้อีกโดยไม่ทำให้เสียการติดต่อจากการสนทนาครั้งแรก



รูปที่ 4.4 หน้าจอการสนทนาเมื่อมีเครื่องลูกข่ายติดต่อเข้ามา

#### 4.2.3 การทดสอบสนทนาแบบห้องสนทนา

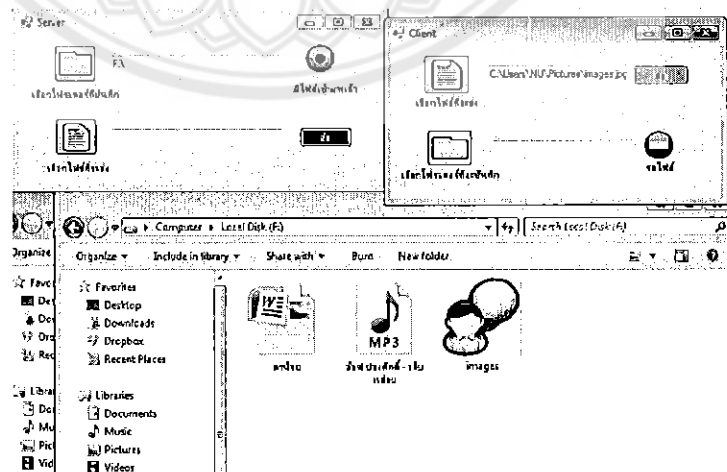
หลักการของการสนทนาแบบห้องสนทนา หน้าทีการรับการส่งข้อมูลจะสร้าง ที่เครื่องแม่ข่าย ไม่ว่าจะเป็น การใส่ผู้ใช้ การส่งข้อความจะกระทำที่เครื่องแม่ข่ายทั้งหมด เหมือนสร้างห้องสนทนาไว้ที่เครื่องแม่ข่าย และส่งข้อความทั้งหมด ไปให้เครื่องลูกข่าย การทดสอบสามารถทำได้ โดยการเข้าห้องสนทนา โดยคลิกปุ่มเข้าห้องสนทนา จะสามารถเข้าห้องสนทนาได้ดังรูป 4.6 โดยจากการทดสอบเครื่องที่อยู่ในเครือข่ายทั้งหมดสามารถที่จะเข้าห้องสนทนาได้ทุกเครื่อง รวมทั้งเครื่องแม่ข่ายด้วย และสามารถสนทนากัน ได้ทุกเครื่อง



รูปที่ 4.5 หน้าจอการเข้าห้องสนทนาของเครื่องแม่ข่ายและเครื่องลูกข่ายหลายๆเครื่อง

#### 4.2.4 การทดสอบรับส่งไฟล์แบบเลือกเครื่อง

การรับส่งไฟล์จะมีลักษณะเหมือนกับการส่งข้อความ แต่จะแตกต่างกันที่ชนิดของข้อมูลซึ่งในการรับส่งไฟล์จะแปลงข้อมูลเป็นไบนารีก่อนส่งเหมือนกับการส่งข้อความ การทดสอบจะรับส่งไฟล์กันได้ ทางเครื่องแม่ข่าย ต้องแสดงความต้องการว่าจะรับส่งไฟล์ โดยคลิกขวาที่ เครื่องลูกข่ายที่เลือก แล้วเลือกคำสั่ง รับส่งไฟล์ เมื่อทำการคลิกเสร็จจะ แสดงหน้าต่างรับส่งไฟล์ ในขณะที่เครื่องลูกข่าย จะแสดงกล่องข้อความว่าทางแม่ข่าย ต้องการที่จะรับส่งไฟล์ จากนั้น กดปุ่มรับส่งไฟล์ที่ เครื่องลูกข่าย ก็จะสามารถรับส่งไฟล์ระหว่างกันได้ เมื่อเราทำการเลือกไฟล์ที่ต้องการส่ง และคลิกที่ปุ่มส่งไฟล์จะมีเครื่องหมายแสดงสถานะสีเขียวที่เครื่องลูกข่ายที่ทำการเลือกไว้ แสดงว่ามีไฟล์เข้ามาแล้ว เมื่อเราดูในแฟ้ม ที่ได้ทำการบันทึกไว้ก็จะปรากฏไฟล์ที่ทำการส่งเข้ามา



รูปที่ 4.6 หน้าจอการรับส่งไฟล์แบบเลือกเครื่อง

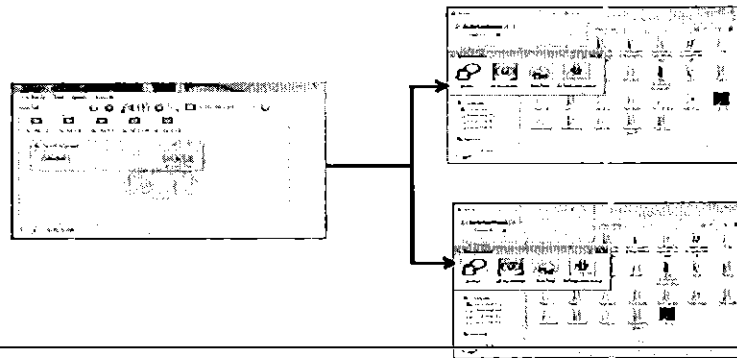
จากการทดสอบรับส่งไฟล์ระหว่างเครื่องสองเครื่อง ได้ลองทดสอบขนาดของไฟล์ ปรากฏว่าสามารถรับไฟล์สูงสุดประมาณ 200 MB โดยความเร็วของการรับส่งไฟล์จะขึ้นอยู่กับขนาดของไฟล์ดังตารางการทดสอบ

ตารางที่ 4.2 ตารางผลการทดลองการรับส่งไฟล์

ลำดับ	ขนาด/MB	เวลา/วินาที	การส่งไฟล์
1	0.01	1.60	ส่งได้
2	0.1	1.65	ส่งได้
3	1	1.70	ส่งได้
4	10	5.70	ส่งได้
5	50	24.72	ส่งได้
6	100	47.96	ส่งได้
7	200	133.44	ส่งได้
8	300	-	ส่งไม่ได้

#### 4.2.5 การทดสอบส่งไฟล์แบบกระจาย

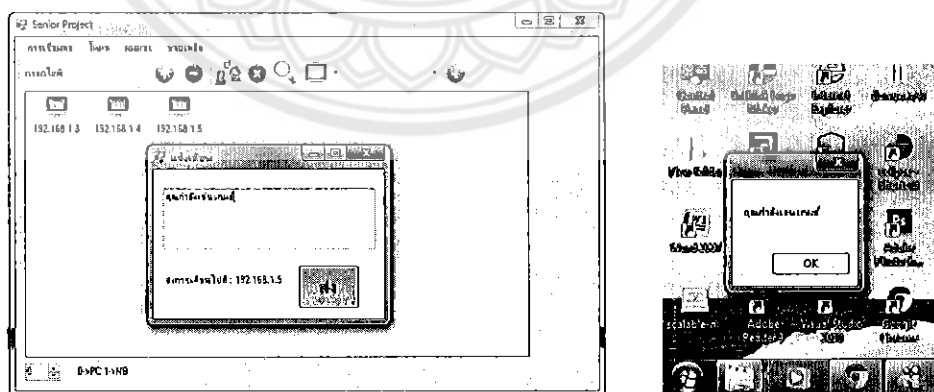
การส่งไฟล์แบบกระจาย ใช้หลักการเกี่ยวกับการส่งไฟล์แบบเลือกเครื่อง ซึ่งการส่งแบบกระจายจะมีลักษณะ คือ สามารถที่จะส่งไฟล์ให้กับทุกเครื่องที่อยู่ในเครือข่ายเดียวกัน โดยหลักการส่งให้กับทุกเครื่องจะใช้วิธีการวน รูป เพื่อส่งไฟล์ให้กับทุกเครื่องที่ทำการติดต่อเข้ามา การทดสอบทำได้โดยคลิก ไอคอนส่งไฟล์แบบกระจาย ที่เครื่องแม่ข่าย จากนั้นจะปรากฏกล่องข้อความที่ลูกข่ายเมื่อ เครื่องลูกข่าย รับทราบต้องทำการคลิกรับไฟล์แบบกระจาย จากนั้นทางเครื่องแม่ข่าย จึงจะสามารถส่งไฟล์ได้ เมื่อทำการคลิกส่งไฟล์ ไฟล์ที่ทำการส่งจะปรากฏที่เครื่องลูกข่ายทุกเครื่อง ดังรูป 4.8 จากการทดสอบ การส่งไฟล์แบบกระจายสามารถที่จะส่งไฟล์ให้กับทุกเครื่องได้ โดยขนาดและความเร็วจะมีความใกล้เคียงกับการรับส่งไฟล์แบบเลือกเครื่อง แต่อาจจะช้ากว่าเพราะจำเป็นต้องวนให้ส่งครบทุกเครื่อง



รูปที่ 4.7 การส่งไฟล์แบบกระจาย

#### 4.2.6 การทดสอบส่งข้อความเตือน

ลักษณะการส่งข้อความเตือนการส่งข้อมูลจะเหมือนกับการส่งข้อความสนทนาธรรมดา จะแตกต่างกันตรงที่ การส่งจะใช้โปรโตคอล UDP เพื่อให้มีความเร็วในการส่งมากขึ้น และการใช้งานจะง่ายกว่า TCP การส่งเตือนส่งลูกข่ายไม่จำเป็นต้องมีการส่งข้อมูลตอบกลับจึงเป็นการส่งข้อมูลแบบทางเดียว การทดสอบทำโดยการคลิกขวาที่เครื่องที่เลือกแล้วเลือกคำสั่งเตือน จากนั้นพิมพ์ข้อความเตือนและกดปุ่มส่ง ข้อความนั้นจะปรากฏที่เครื่องที่ทำการเลือกไว้ จากการทดสอบเครื่องแม่ข่ายสามารถที่จะส่งข้อความเตือนไปยังเครื่องลูกข่ายได้ดังรูป 4.7 และเห็นได้ชัดว่าการส่งทำได้อย่างรวดเร็วและสามารถเลือกเครื่องส่งพร้อมกันได้โดยที่ความเร็วยังไม่ลดลง แสดงให้เห็นถึงข้อดีของ โปรโตคอล UDP



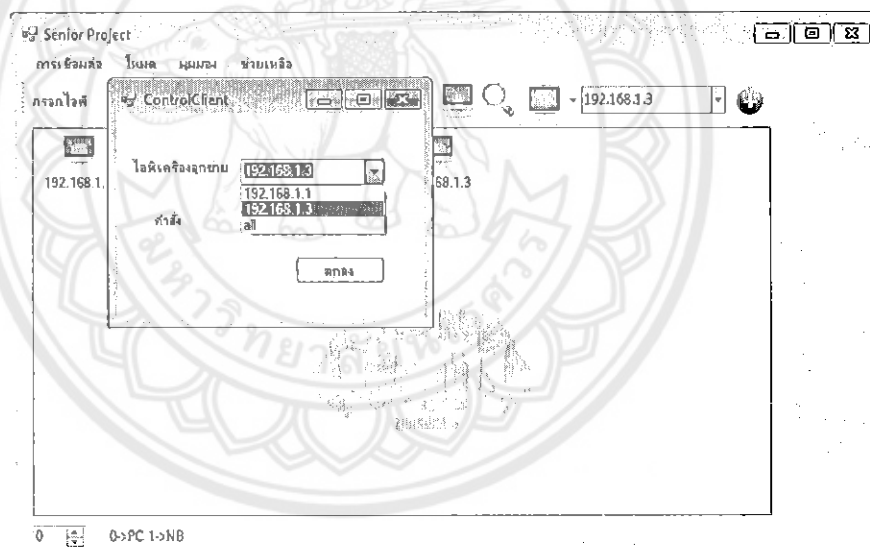
รูปที่ 4.8 หน้าจอทดสอบการเตือน

#### 4.2.7 การทดสอบ Turn off, Restart, Log off, Lock Screen

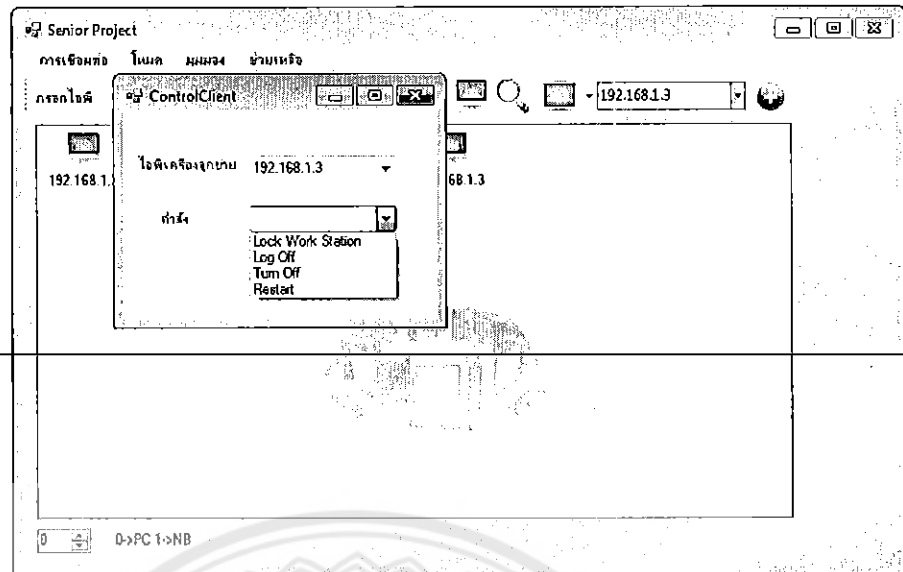
วิธีการทดลอง คือ คลิกที่ปุ่ม”ระบบจัดการปิดเครื่อง” ที่แถบ Toolstripmenu ก็จะปรากฏหน้าต่างการควบคุมเครื่องลูกข่าย ซึ่งจะสามารถจัดการได้ทั้งที่อยู่ไอพีเดียวและที่อยู่ไอพีรวมได้ทั้งหมด

โดยจะมีอยู่ 4 คำสั่งในการควบคุมเครื่องลูกข่ายให้ผู้ใช้เลือกใช้ คือ

1. ปิดเครื่อง (Turn off)
2. ปิดเครื่องแล้วเปิดใหม่ (Restart)
3. ล็อกออฟ (Log off)
4. ล็อกหน้าจอ(Lock Screen)

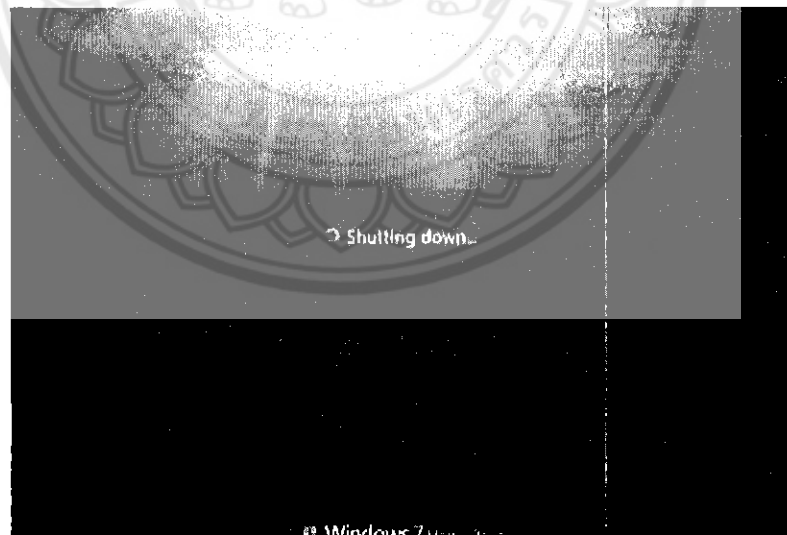


รูปที่ 4.9 หน้าจอการปิดเครื่อง รีสตาร์ท ล็อกออฟ ล็อกหน้าจอ



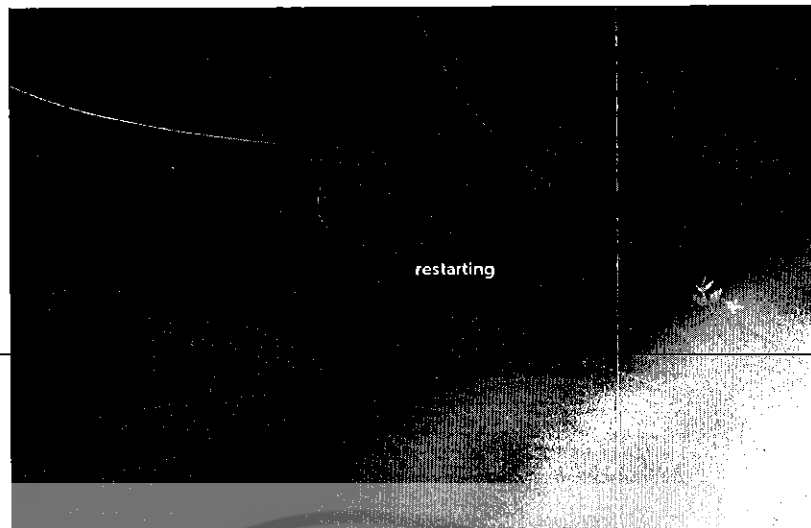
รูปที่ 4.10 รูปการปิดเครื่อง รีโมทรีท ล็อคออฟ ล็อคหน้าจอ 2

หลังจากที่เลือกคำสั่งควบคุมเครื่องลูกข่ายแล้ว เครื่องลูกข่ายก็จะปรากฏดังรูปที่ 4.14 ถึงรูปที่ 4.16

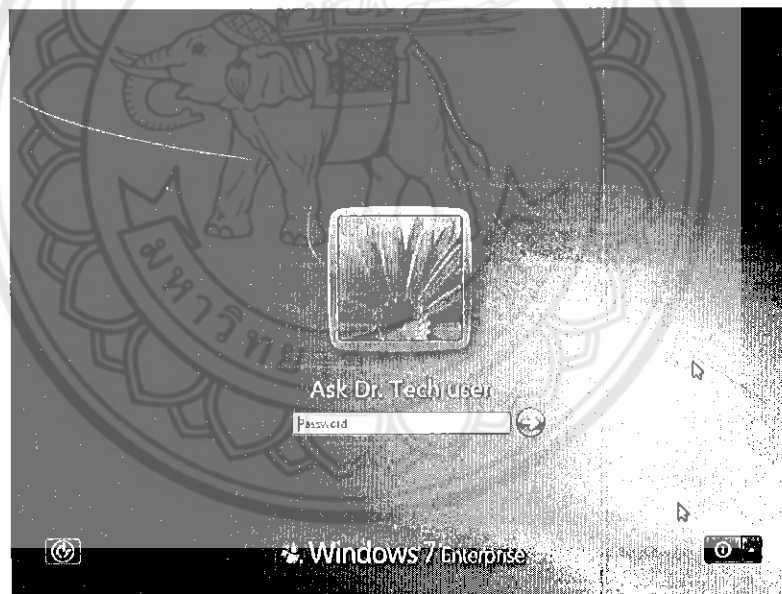


รูปที่ 4.11 หน้าจอเครื่องลูกข่ายถูกสั่งปิดเครื่อง (Turn off)

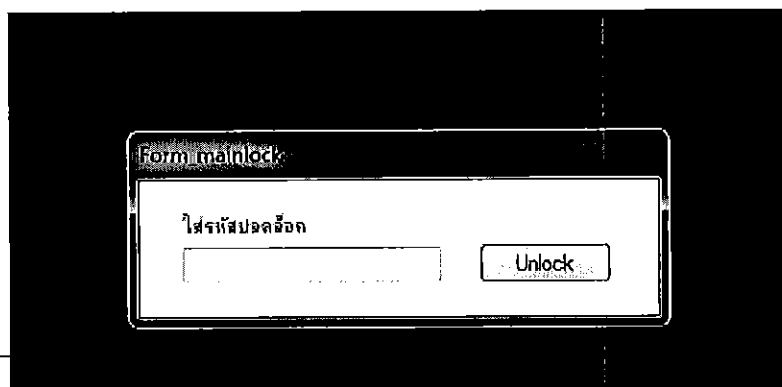




รูปที่ 4.12 หน้าจอเครื่องถูกข่ายถูกสั่งปิดเครื่องแล้วเปิดใหม่ (Restart)



รูปที่ 4.13 หน้าจอเครื่องถูกข่ายถูกสั่งล็อกออฟ (Log off)

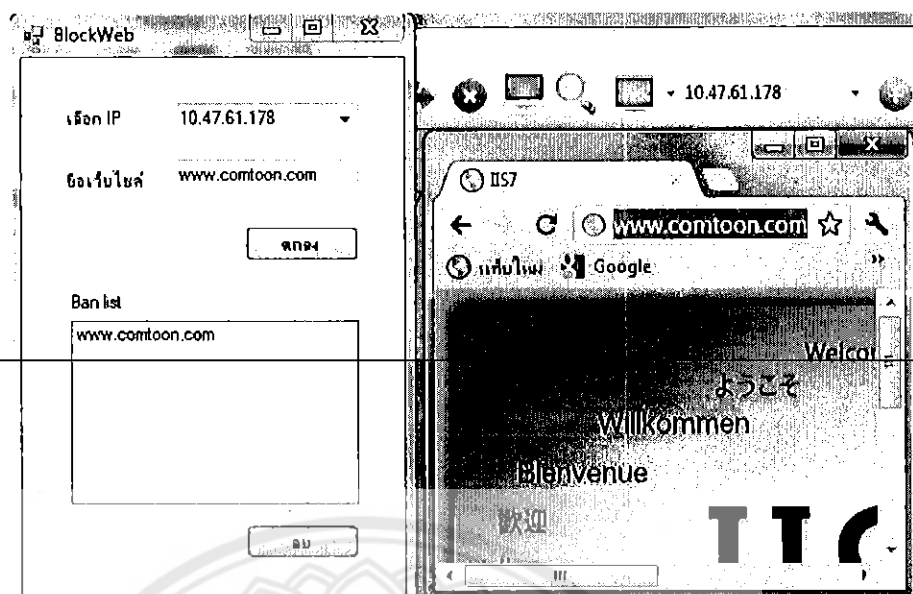


รูปที่ 4.14 หน้าจอเครื่องลูกข่ายถูกสั่งล็อกหน้าจอ (Lock Screen)

จากการที่เลือกควบคุมเครื่องลูกข่ายทั้ง 4 ประเภท จะเห็นได้ว่ามีความสำคัญในการที่ลูกข่ายลืมปิดเครื่อง หรือต้องการหยุดไม่ให้ผู้ใช้เครื่องลูกข่ายใช้งานคอมพิวเตอร์

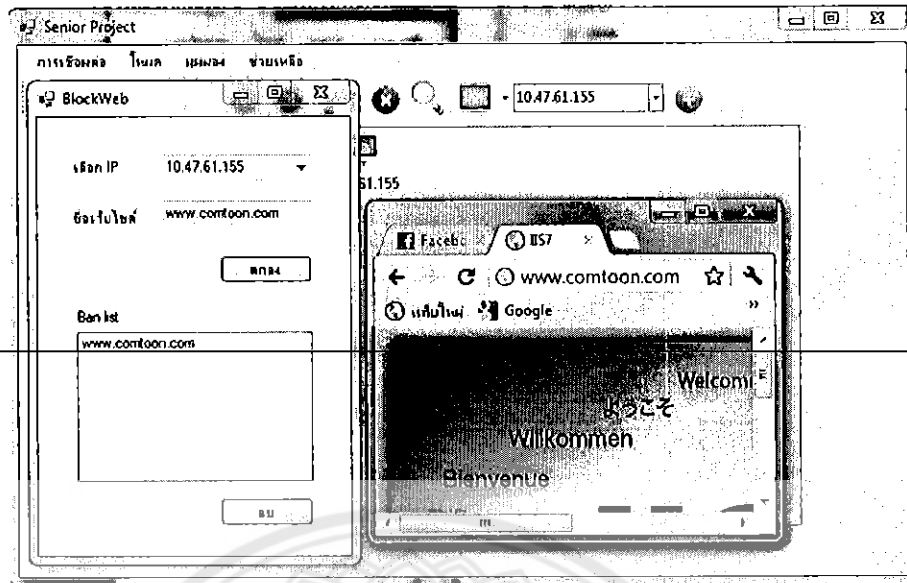
#### 4.2.8 การทดสอบจำกัดการเข้าเว็บไซต์

วิธีการทดลอง คือ คลิกขวาที่ปุ่ม “บล็อกเว็บไซต์” ที่ไอคอนคอมพิวเตอร์ของเครื่องลูกข่ายแต่ละที่อยู่ไอพี จากนั้นก็จะมีหน้าต่างขึ้นมา ให้ผู้ใช้เลือกที่อยู่ไอพีที่จะเลือกหยุดการเข้าถึงเว็บไซต์แบบเลือกที่อยู่ไอพีเครื่องลูกข่ายแบบเดียว หรือเลือกที่อยู่ไอพีเครื่องลูกข่ายแบบรวม จากนั้นก็มาพิมพ์ชื่อเว็บไซต์ที่เราจะทำการหยุดการทำงานไม่ให้เครื่องลูกข่ายเข้าถึงได้



รูปที่ 4.15 หน้าจอการจำกัดการเข้าถึงเว็บไซต์ของเครื่องลูกข่าย

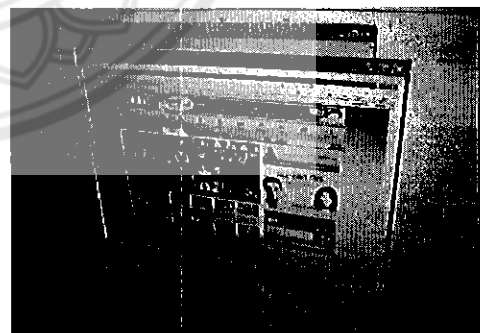
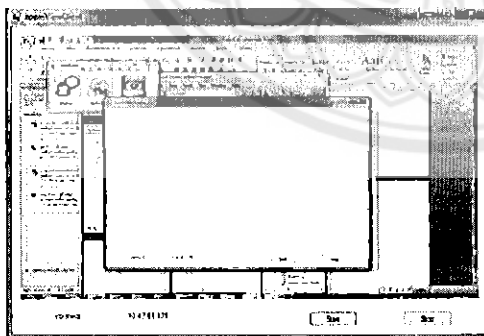
หลังจากที่กดปุ่มตกลงแล้ว เมื่อเครื่องลูกข่ายเปิดเว็บไซต์ในแต่ละเว็บเบราว์เซอร์แล้ว ก็จะพบว่าไม่สามารถเข้าถึง URL (Uniform Resource Locator) ได้ ไม่ว่าจะเครื่องลูกข่ายจะเปลี่ยนที่อยู่ไอพี หรือจะเปลี่ยน ISP (Internet Service Provider) ในการใช้งานอินเทอร์เน็ตก็ตาม ดังรูปที่ 4.18 เพราะโปรแกรมได้พัฒนาด้วยการฝัง URL (Uniform Resource Locator) ไว้ในระบบปฏิบัติการและแมกแอดเดรส (Mac Address) ซึ่งจะฝังอยู่ใน NIC (Network Interface Card) ของแต่ละเครื่อง ซึ่งจะมีหมายเลขไม่ซ้ำกัน



รูปที่ 4.16 หน้าจอการจำกัดการเข้าถึงเว็บไซต์ของเครื่องลูกข่าย แบบเปลี่ยน IP Address

#### 4.2.9 การดูหน้าจอเครื่องลูกข่ายแบบเลือกเครื่อง

วิธีการทดลอง คือ คลิกขวาที่ปุ่ม “ดูหน้าจอ” ที่ไอคอนคอมพิวเตอร์ของเครื่องลูกข่ายแต่ละที่อยู่ไอพี จากนั้นก็จะมีหน้าต่างขึ้นมา ให้กดคำว่า Connect จากนั้นก็จะปรากฏภาพเคลื่อนไหวของเครื่องลูกข่ายออกมา เมื่อต้องการหยุดหน้าจอก็คลิกคำว่า Disconnect หน้าจอก็จะดับไป



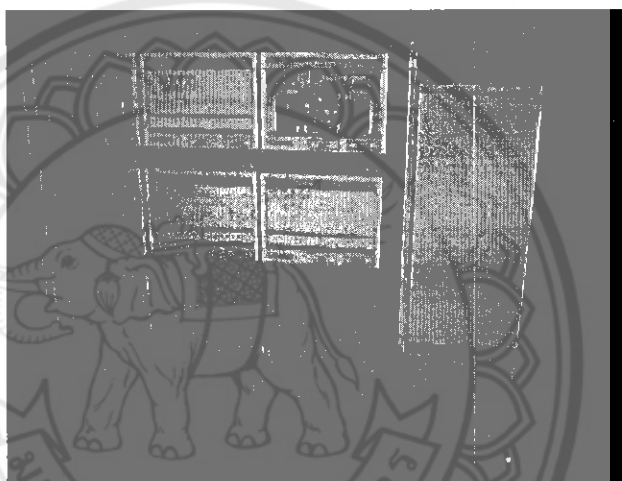
รูปที่ 4.17 การทดลองดูหน้าจอเครื่องลูกข่าย

ในการทดลองนี้ เป็นส่วนที่สำคัญ เนื่องจากเราสามารถตรวจสอบหน้าจอเครื่องลูกข่ายได้ตลอดเวลา ว่าผู้ใช้เครื่องลูกข่ายกำลังทำอะไรอยู่กับเครื่องตัวเอง โดย โปรแกรมจะส่งภาพให้เครื่องแม่ข่ายดูอย่างต่อเนื่อง

#### 4.2.10 การดูหน้าจอเครื่องลูกข่ายหลายๆเครื่อง

วิธีการทดลอง คือ คลิกที่ปุ่มไอคอนคำว่า ดูหลายจอ ในแถบแจ้งข้อความไอคอนและข้อความ (Toolstripmenu) ก็จะเห็นหน้าจอเครื่องลูกข่ายหลายๆเครื่อง ที่ติดต่อสื่อสารกับเครื่องแม่ข่ายอยู่ ซึ่งก็จะเรียงหน้าต่างจากซ้ายไปขวา บรรทัดละ 3 หน้าต่าง ซึ่งสามารถดูหน้าจอเครื่องลูกข่ายแบบองค์รวมได้ถึง 3 หน้าจอ หากจะขยายก็กดที่ปุ่ม Zoom เพื่อทำการขยายภาพหน้าจอเครื่องลูก

ข่าย

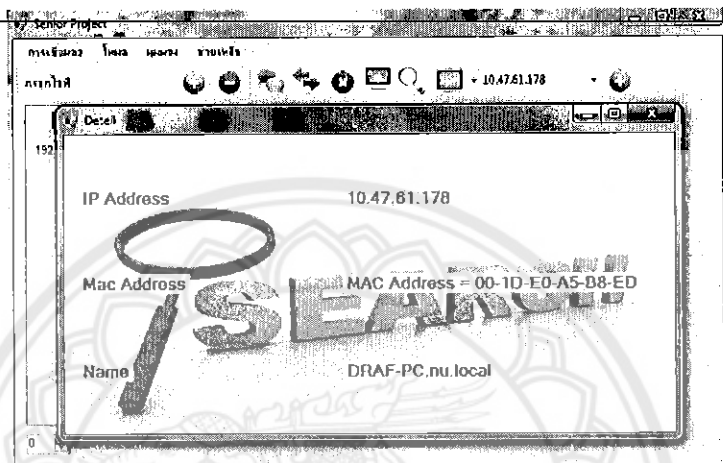


รูปที่ 4.18 การทดลองดูหน้าจอหลายๆเครื่องของเครื่องลูกข่าย

#### 4.2.11 การดูข้อมูลเบื้องต้นของเครื่องลูกข่าย

วิธีการทดลอง คือ คลิกขวาที่ปุ่ม “ข้อมูล” ที่ไอคอนคอมพิวเตอร์ของเครื่องลูกข่ายแต่ละ IP Address

ก็จะปรากฏหน้าต่างขึ้นมา ซึ่งจะมีข้อมูล 3 ชนิดคือ ที่อยู่ไอพี, ชื่อเครื่อง และ แมกแอดเดรส



รูปที่ 4.19 การทดลองดูข้อมูลเบื้องต้นของเครื่องลูกข่าย

## บทที่ 5

### สรุปผลการดำเนินงานและแนวทางการพัฒนา

โครงการนี้จัดทำขึ้นเพื่อพัฒนาโปรแกรมที่ใช้ในห้องปฏิบัติการคอมพิวเตอร์เพื่อให้นักเรียน การสอนมีประสิทธิภาพมากขึ้น อีกทั้งยังเป็นการศึกษาหาความรู้นอกตำราเรียนเพื่อให้ได้มาของ โปรแกรม จากการดำเนินงานทำให้ทราบว่า การพัฒนาโปรแกรมจำเป็นต้องมีความรู้พื้นฐานในการ พัฒนาพอสมควร สรุปได้ดังนี้

1. ความรู้ด้านเน็ตเวิร์ค ต้องมีความเข้าใจอุปกรณ์และการเชื่อมต่อกันผ่านเครือข่ายของแต่ละ เครือข่าย มีความรู้ด้าน ไอพีแอดเดรสและพอร์ตอันเป็นความรู้พื้นฐานในการติดต่อสื่อสาร ข้อมูลข้ามเครื่อง
2. ความรู้ด้าน Socket Programming เป็นสิ่งที่มีความสำคัญมากในการติดต่อสื่อสารผ่าน เครือข่าย จากเครื่องหนึ่งสู่อีกเครื่องหนึ่ง
3. ความรู้ด้าน C# Programming เป็นสิ่งพื้นฐานในการพัฒนาโปรแกรมและยังรวบรวม เครื่องมือต่างๆ ที่ทำให้การพัฒนาโปรแกรมสามารถทำได้ง่ายขึ้น อย่างเช่น .NET Framework
4. ความรู้ด้านระบบต้องมีความรู้เกี่ยวกับการทำงานภายในของระบบคอมพิวเตอร์ เช่น การ ทำงานของระบบวินโดวส์

#### 5.1 สรุปผลการทดลอง

สิ่งที่เราได้วางขอบเขตของงาน เราก็จะสามารถทำผลการทดลองว่าได้ทำสำเร็จและมี ประสิทธิภาพได้ดังนี้

1. โปรแกรมสามารถทราบที่อยู่ไอพี ( IP Address) และชื่อเครื่องของลูกข่าย เมื่อเครื่องลูกข่าย ติดต่อมา
2. โปรแกรมสามารถสนทนากับเครื่องลูกข่ายที่เลือกได้
3. โปรแกรมสามารถส่งข้อความเตือนไปหาเครื่องลูกข่ายที่เลือกได้

4. โปรแกรมสามารถสร้างห้องสนทนาและ ผู้ใช้สามารถเข้าไปสนทนาได้
  5. โปรแกรมสามารถรับส่งไฟล์จากเครื่องลูกข่ายที่เลือกและส่งไฟล์แบบกระจายไปที่เครื่องลูกข่ายทั้งหมดได้
  6. โปรแกรมสามารถดูหน้าจอเครื่องลูกข่ายที่เลือกได้
- 
7. โปรแกรมสามารถจำกัดการเข้าถึงเว็บไซต์ที่เครื่องลูกข่ายเฉพาะที่เลือกหรือทั้งหมดได้
  8. โปรแกรมสามารถสั่งเครื่องลูกข่ายเฉพาะที่เลือกหรือทั้งหมด ปิด (Turn off) ปิดและเปิดเครื่องใหม่ (Restart) ล็อกออฟ ( Log off ) ล็อกหน้าจอ( Lock Screen) ได้
  9. โปรแกรมสามารถส่งไฟล์แบบกระจายไปที่เครื่องลูกข่ายได้

## 5.2 ปัญหาและอุปสรรค

ตารางที่ 5.2 ปัญหาและอุปสรรค

ปัญหาและอุปสรรค	แนวทางแก้ไข
1. การทำงานของ Socket ถ้ากำลังส่งข้อมูลอยู่จะไม่สามารถรับข้อมูลที่เข้ามาได้ ดังนั้นจึงเกิดปัญหาไม่สามารถรับส่งข้อมูลได้ในเวลาเดียวกัน	1.ต้องทำการศึกษา Asynchronous Socket เพื่อไม่ต้องรอให้การส่งเสร็จก่อน ถึงจะรับได้
2. เนื่องจากมีการทำงานหลายงาน งานอื่นจะเริ่มได้ต้องรอการทำงานหนึ่งให้เสร็จก่อน ในขณะเดียวกันถ้าต้องการทำงานอื่น ในเวลาเดียวกันปัญหามีที่ว่าต้องใช้พอร์ตเดียวกัน ทำให้ไม่สามารถทำงานนั้นได้	2. ต้องทำการกำหนดพอร์ตของแต่ละงานให้แตกต่างกันเพื่อป้องกันการชนกันของพอร์ตเดียวกัน
3. เนื่องจากการทำงานระหว่างแม่ข่ายกับลูกข่ายต้องสอดคล้องกัน และต้องมีการ ใช้พอร์ตเดียวกันตลอดเวลา ปัญหาเกิดตรงที่ ถ้ามีการใช้งานเดียวกันนั้นในขณะที่งานแรกยังไม่เสร็จจะไม่สามารถทำงานได้เพราะพอร์ตนั้นถูกใช้ไปแล้ว	3. ให้ทางแม่ข่ายส่งพอร์ตที่ต้องการใช้งานไปให้ทางด้านลูกข่าย เพื่อให้การทำงานสอดคล้องกัน



### ตารางที่ 5.2 ปัญหาและอุปสรรค (ต่อ)

4. เมื่อมีการติดต่อพร้อมกันหลายเครื่อง ถ้ามีเครื่องหนึ่งได้ใช้งานนั้นไปแล้ว เครื่องที่เข้ามาทีหลังจะไม่สามารถทำงานนั้นได้	4. ต้องทำการศึกษาการทำงานแบบใช้ตัวแบ่งงาน (Thread) เพื่อให้รองรับกับการทำงานพร้อมกัน
5. ปัญหาการส่งรูปภาพไปแสดงแบบเคลื่อนไหว เนื่องจากขนาดภาพมีขนาดใหญ่ทำให้การส่งล่าช้า	5. ต้องทำการย่อภาพให้มีขนาดลดลงและกำหนดขนาดบัฟเฟอร์ให้พอดี

### 5.3 ข้อจำกัดของโปรแกรม

1. โปรแกรมที่เครื่องแม่ข่ายต้องรันก่อน โปรแกรมเครื่องลูกข่าย เพราะการทำงานทางด้านแม่ข่ายจะรอการติดต่อเข้ามาของโปรแกรมเครื่องลูกข่าย มิฉะนั้นจะไม่สามารถใช้งานได้
2. การรับส่งไฟล์ หรือ การสนทนาแบบเลือกเครื่องต้องรอให้เครื่องแม่ข่ายติดต่อมาก่อนถึงจะใช้งานได้
3. เนื่องจากโปรแกรมลูกข่ายต้องทำการฝังที่อยู่ไอพีของแม่ข่ายไว้ ดังนั้นจะทำการเปลี่ยนที่อยู่ไอพีของเครื่องแม่ข่ายไม่ได้ ถ้าเปลี่ยนไอพีของเครื่องแม่ข่ายต้องทำการเปลี่ยนไอพี ที่ฝังไว้ในโปรแกรมลูกข่ายด้วย
4. โปรแกรมสามารถรองรับการดูหน้าจอเครื่องลูกข่ายในห้องปฏิบัติการคอมพิวเตอร์ได้ 10 หน้าจอ จากการทดลองในห้องปฏิบัติการคอมพิวเตอร์ ถ้ามากกว่านี้การรับส่งไฟล์ซึ่งเป็นรูปภาพจะไม่สามารถส่งได้ เพราะการส่งข้อมูลในโปรโตคอลมีข้อจำกัด

### 5.4 ข้อเสนอแนะ

เนื่องจากการทดลองพัฒนาโปรแกรม มีการทำงานที่หลากหลายโดยสิ่งจำเป็นของการทำงานผ่านเครือข่ายนี้คือการใช้งาน Socket ซึ่งมีสองชนิดหลักด้วยกัน ชนิดแรกคือ Socket ที่ใช้โปรโตคอล TCP และอีกชนิดเป็น Socket ที่ใช้โปรโตคอล UDP ซึ่งในการนำ Socket นี้มาพัฒนาเราสามารถนำทั้ง

สองประเภทนี้มาประยุกต์หรืออาจนำมาใช้งานร่วมกัน เพื่อให้ได้งานที่หลากหลายและยืดหยุ่นมากขึ้น เนื่องจากหลักการทำงานของทั้งสองมีความแตกต่างกันและมีข้อดีข้อเสียที่แตกต่างกัน

### 5.5 แนวทางการพัฒนาต่อ

Auto Start up Program เป็นสิ่งที่ต้องพัฒนาต่อ เพราะมีความสำคัญในการติดต่อระหว่างเครื่องแม่ข่ายและเครื่องลูกข่าย โดยที่เครื่องลูกข่ายจะทำการรัน โปรแกรมเองอัตโนมัติ ทำให้ผู้สอนไม่ต้องสั่งให้ผู้เรียนเปิด โปรแกรมที่เครื่องลูกข่าย ในขณะที่กำลังจะส่งการทำงานไปที่เครื่องลูกข่าย ทั้งนี้ก็เพื่ออำนวยความสะดวกแก่ผู้สอนเป็นสำคัญ

### 5.6 สรุปผลการดำเนินงาน

จากการพัฒนาโปรแกรมทำให้ได้โปรแกรมที่สามารถใช้งานในห้องปฏิบัติการคอมพิวเตอร์ที่อยู่ในเครือข่ายเดียวกัน และต้องมีการติดตั้ง โปรแกรมทั้งสองฝ่าย ในการทำงานนั้น โปรแกรมฝั่งแม่ข่ายต้องเปิดทำงานก่อน และเมื่อเปิดการทำงานของ โปรแกรมที่เครื่องลูกข่าย โปรแกรมจึงจะสามารถทำงานได้ตามเป้าหมาย โดยโปรแกรมสามารถที่จะอำนวยความสะดวกต่อผู้สอน ไม่ว่าจะเป็นการทำงานในส่วนของการติดต่อสื่อสารผ่านการสนทนาของ โปรแกรม หรือการส่ง ไฟล์ระหว่างเครื่อง และยังสามารถดูพฤติกรรมการใช้งานคอมพิวเตอร์ของนักเรียนผ่าน โปรแกรม อีกทั้งยังรวมไปถึงการควบคุมการทำงานของเครื่องคอมพิวเตอร์ลูกข่าย ผู้สอนสามารถที่จะจัดการกับเครื่องลูกข่ายได้ทันที เมื่อผู้สอนมีความต้องการอย่างเช่น การจำกัดการเข้าถึงเว็บไซต์ หรือ ดับเครื่อง จะเห็นได้ว่า โปรแกรมสามารถจัดการกับเครื่องลูกข่ายได้ในระดับหนึ่ง จากการพัฒนาและทดลองหวังว่าจะเป็นประโยชน์และเป็นแนวทางในการพัฒนาของผู้ที่สนใจปัญหาในลักษณะคล้ายกัน อันนำไปสู่การพัฒนาเครื่องมือที่มีความทันสมัยและตรงต่อความต้องการในอนาคต