สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมอินสแตนซ์เมจเสจจิ้งสำหรับองค์กร

INSTANT MESSAGING FOR ORGANIZATION

T 051778

ฉัตรดนัย ช้างพันธ์
อภิรดี ศักดิ์บรรพต

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 51778
วัน,เดือน,ปี 29 ก.ค. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b.....
i.....

INSTANT MESSAGING FOR ORGANIZATION

CHATDANAI CHANGPHAN

APIRADEE SAKBANPOT

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	โปรแกรมอินสแตนซ์เมสเสจจิ้งสำหรับองค์กร INSTANT MESSAGING FOR ORGANIZATION
ชื่อนักศึกษา	นายฉัตรดนัย ช้างพันธ์ 43050363 นางสาวอภิรดี ศักดิ์บรรพต 43050434
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผศ.ดร.จีรพร ศรีสวัสดิ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2546

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	อ.สิริลักษณ์ อนันต์สถิตย์สิน	
กรรมการ	ผศ.ดร.ศรัณย์ อินทโกสุม	
กรรมการและอาจารย์ที่ปรึกษา	ผศ.ดร.จีรพร ศรีสวัสดิ์	

(ผู้ช่วยศาสตราจารย์ ดร.วีระ บุญจริง)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ โปรแกรมอินสแตนซ์เมสเสจจิ้งสำหรับองค์กร
INSTANT MESSAGING FOR ORGANIZATION

ชื่อนักศึกษา นายฉัตรดนัย ช้างพันธ์ 43050363
นางสาวอภิรดี ศักดิ์บรรพต 43050434

ปริญญา วิทยาศาสตรบัณฑิต
ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2546
อาจารย์ที่ปรึกษา ผศ.ดร.จีรพร ศรีสวัสดิ์

บทคัดย่อ

ปัญหาพิเศษเรื่องโปรแกรมอินสแตนซ์เมจเสจจิ้งสำหรับองค์กรนี้เป็นการนำเอาลักษณะเด่นของโปรแกรมการสื่อสารข้อมูลบนอินเทอร์เน็ตที่มีอยู่ในปัจจุบันมาพัฒนาให้มีความปลอดภัยเหมาะสำหรับใช้ในการส่งข้อมูลภายในองค์กร โดยใช้หลักการของ Socket และ Thread พร้อมทั้งมีการออกแบบลักษณะการสั่งคำสั่งขึ้นมาใช้กับโปรแกรมอินสแตนซ์เมจเสจจิ้งที่พัฒนาขึ้นโดยเฉพาะ ใช้ภาษา Java ในการสร้างพัฒนาโปรแกรม โดยใช้ JBuilder เป็นเครื่องมือในการเขียนโปรแกรม

โปรแกรมระบบงานที่พัฒนาขึ้นนี้ สามารถรองรับการใช้งานได้บนเครือข่ายภายในขององค์กร โดยที่โปรแกรมมีความสามารถในการส่งข้อมูล และแลกเปลี่ยนไฟล์ข้อมูลระหว่างบุคลากรภายในองค์กร ซึ่งเป็นการช่วยประหยัดเวลาและเพิ่มประสิทธิภาพในการสื่อสารข้อมูลระหว่างบุคลากรที่อยู่ในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	INSTANT MESSAGING FOR ORGANIZATION
Students	Mr. Chatdanai Changphan 43050363
	Miss Apiradee Sakbanpot 43050434
Degree	Bachelor of Science
Department	Mathematics and Computer Science, Faculty of Science
Programme	Computer Science
Academic Year	2003
Special Project Advisor	Asst.Prof.Dr.Jeeraporn Srisawat

ABSTRACT

This special problem " Instant Messaging for Organization " is developed based upon the recent instant messaging over Internets to be more secure for using in an organization. It uses principle of socket and thread. Also, the Instant Messaging for Organization uses specific commands and protocols. Java language and JBuilder are used as the important tools in developing the program.

This Instant Messaging for Organization supports for applying over any local area network. It is able to send messages, chat, and transfer files between personnel. The main objective of the Instant Messaging for Organization is to save time and to yield efficient communication between personnel of the organization.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการปัญหาพิเศษเรื่องโปรแกรมอินสแตนซ์เมสเสจจิ้งสำหรับองค์กรสามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.จีรพร ศรีสวัสดิ์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานปัญหาพิเศษนี้ที่กรุณาให้คำแนะนำในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องของโครงงานปัญหาพิเศษฉบับนี้

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ได้ให้ความสนับสนุนทางด้านกำลังใจและทุนทรัพย์ จนการทำปัญหาพิเศษนี้สำเร็จลุล่วงไปด้วยดี ขอขอบพระคุณคณาจารย์ทุกท่านที่ได้ทำการอบรมสั่งสอนวิชาความรู้ รวมทั้งเพื่อนๆ และพี่ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาพิเศษ

Instant messaging เป็นโปรแกรมส่งข้อความระหว่างผู้ใช้อินเทอร์เน็ตที่ได้รับความนิยมอย่างมาก เนื่องจากข้อความที่ส่งไปจะถึงผู้ใช้ทันทีเมื่อผู้ใช้กำลัง online อยู่ ถ้าผู้ใช้ไม่ได้ online อยู่ ผู้ใช้ก็จะได้รับข้อความเมื่อ onlineครั้งต่อไป นอกจากนี้ยังมีความสามารถในการรับ-ส่งแฟ้มข้อมูลระหว่างผู้ใช้ได้โดยตรง จากคุณสมบัติดังกล่าว ถ้าเราสามารถนำเอา Instant messaging มาประยุกต์ใช้ในองค์กรก็จะสามารถเพิ่มประสิทธิภาพในการติดต่อสื่อสารระหว่างบุคลากรภายในองค์กรได้ แต่ระบบ Instant messaging ที่มีบริการในอินเตอร์เน็ตในขณะนี้จะอนุญาติให้ผู้ใช้ซึ่งอยู่ในส่วนใดของโลกติดต่อสื่อสารกัน ดังนั้นถ้านำ Instant messaging ดังกล่าวมาใช้ในองค์กรโดยตรง ก็อาจก่อให้เกิดการรั่วไหลของข้อมูลออกจากองค์กร

จากข้อดี และปัญหาของ Instant messaging ที่ให้บริการอยู่ดังที่กล่าวมา จึงมีแนวคิดที่จะพัฒนาโปรแกรม Instant messaging เพื่อใช้ในการสื่อสารข้อมูลเฉพาะบุคลากรภายในองค์กร ซึ่งโปรแกรมดังกล่าวจะช่วยเพิ่มประสิทธิภาพในการสื่อสารภายในองค์กร และข้อมูลที่สำคัญก็จะไม่รั่วไหลไปสู่ภายนอกด้วย

1.2 วัตถุประสงค์ของปัญหาพิเศษ

1. ศึกษาและทำความเข้าใจ พื้นฐานการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ภายในเครือข่าย
2. พัฒนาโปรแกรมเพื่อเพิ่มประสิทธิภาพในการประสานงานระหว่างบุคลากรในองค์กรโดยข้อมูลภายในองค์กรไม่รั่วไหล

1.3 ขอบเขตของปัญหา

โปรแกรม Instant messaging ที่ทำการพัฒนาขึ้นมีความสามารถ ดังนี้

1. ส่งข้อความระหว่างบุคคลภายในองค์กรไม่ว่าบุคลากรนั้นจะ online อยู่หรือไม่
2. ส่งข้อความถึงกลุ่มบุคคลภายในองค์กร
3. สืบค้นแฟ้มข้อมูลว่าอยู่ในเครื่องใดในเครือข่าย
4. ส่งแฟ้มข้อมูลระหว่าง 2 คน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ซอฟต์แวร์ที่ช่วยเพิ่มประสิทธิภาพในการติดต่อกันภายในองค์กรโดยข้อมูลภายในไม่รั่วไหล
2. ได้นำเนื้อหาวิชาที่เรียนมาทั้งในด้านการสื่อสารข้อมูล และออกแบบและพัฒนาระบบมาประยุกต์เพื่อใช้งานจริง

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาความรู้พื้นฐานเกี่ยวกับการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่าย
2. ศึกษาการทำงานของ server เกี่ยวกับการทำงานของ Instant messaging, การเขียนโปรแกรมบนเครือข่าย, การเขียนโปรแกรมติดต่อ database, thread, graphic user interface
3. ศึกษาเครื่องมือที่ใช้ในการเขียนโปรแกรม
4. วิเคราะห์และออกแบบระบบ
5. สร้างและพัฒนาโปรแกรม Server และ Client
6. ทดสอบการทำงานของโปรแกรม
7. สรุปผลการดำเนินงานของปัญหาและแนวทางในการปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 แผนการดำเนินงาน

ลำดับ	ชื่องาน	ภาคเรียนที่ 1					ภาคเรียนที่ 2				
		มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1	ระบุปัญหา และขอบเขตของโครงงาน										
2	ศึกษาหลักการ และเครื่องมือที่ใช้ในการทำโครงงาน										
3	รวบรวมข้อมูล และจัดทำเอกสารประกอบโครงงาน										
4	วิเคราะห์และออกแบบระบบ										
5	สร้างและพัฒนาโปรแกรม										
6	ทดสอบการทำงานของโปรแกรม										
7	สรุปผลการดำเนินงานของปัญหาและแนวทางในการปรับปรุง										

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 TCP/IP

2.1.1 ลักษณะโดยรวมของ protocol

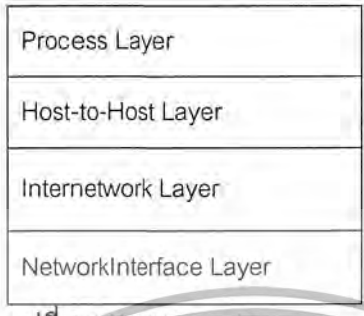
TCP/IP เป็นมาตรฐานของการรับส่งข้อมูลระหว่างคอมพิวเตอร์สองเครื่องที่เกิดการค้นคิดของกระทรวงกลาโหมของประเทศสหรัฐอเมริกาที่ต้องการเชื่อมโยงคอมพิวเตอร์ทางการทหารของแต่ละหน่วย ซึ่งเป็นคอมพิวเตอร์ต่างชนิดกันให้สามารถติดต่อรับส่งข้อมูลกัน protocol TCP/IP ประกอบด้วยส่วนหลักๆ 2 ส่วนคือ Transmission Control Protocol หรือ TCP และ Internet Protocol หรือ IP ซึ่ง TCP มีหน้าที่ตรวจสอบการรับส่งข้อมูลระหว่างคอมพิวเตอร์ผู้รับและผู้ส่งให้ได้ข้อมูลครบถ้วน หากข้อมูลเสียหายก็จะแจ้งให้ต้นทางส่งข้อมูลใหม่ส่วน IP จะมีหน้าที่เลือกเส้นทางที่ใช้รับส่งข้อมูลผ่านระบบเครือข่าย และตรวจสอบ address ของผู้รับ อย่างไรก็ตามการเชื่อมต่อคอมพิวเตอร์ต่างระบบกันทำได้ยากในยุคแรกๆของการสื่อสารข้อมูล จึงมีการกำหนดมาตรฐานส่วนกลางที่จำเป็นต้องใช้ในการรับส่งข้อมูล เรียกว่า Open System Interconnection หรือ OSI

Application Layer เชื่อมต่อกับผู้ใช้ และแปลคำสั่งต่างๆให้กับคอมพิวเตอร์อย่างถูกต้องตามกฎ
Presentation Layer แปลงคำสั่งตามกฎที่ได้รับออกเป็นขั้นตอนย่อยๆแต่ละขั้นตอน
Session Layer ควบคุมจังหวะการรับส่งข้อมูลของคอมพิวเตอร์ทั้งสองด้านให้โต้ตอบกันตามวิธีที่กำหนด
Transport Layer เชื่อมต่อรับส่งข้อมูลระหว่างปลายทาง ควบคุมข้อผิดพลาดและตัดข้อมูลออกเป็นส่วนย่อย
Network Layer ติดต่อกำหนดเส้นทางการรับส่งข้อมูลผ่านเครือข่ายและตรวจสอบ address ของผู้รับ
Datalink Layer ควบคุมการรับส่งข้อมูลในระดับฮาร์ดแวร์ และตรวจสอบข้อผิดพลาดในการรับส่งข้อมูล
Physical Layer กำหนดคุณสมบัติการเชื่อมต่อรับส่งข้อมูลทางฮาร์ดแวร์ ความเร็วและการเชื่อมต่อเข้ากับสายการรับส่งข้อมูล

รูปที่ 2.1 โครงสร้างและหน้าที่ของ OSI Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OSI กำหนดให้การสื่อสารข้อมูลจากระบบคอมพิวเตอร์หนึ่งไปยังอีกระบบคอมพิวเตอร์หนึ่ง แบ่งออกเป็น 7 ขั้นย่อยๆ ดังรูปที่ 2.1



รูปที่ 2.2 TCP/IP stack

2.1.2 การทำงานในแต่ละระดับชั้นลักษณะ

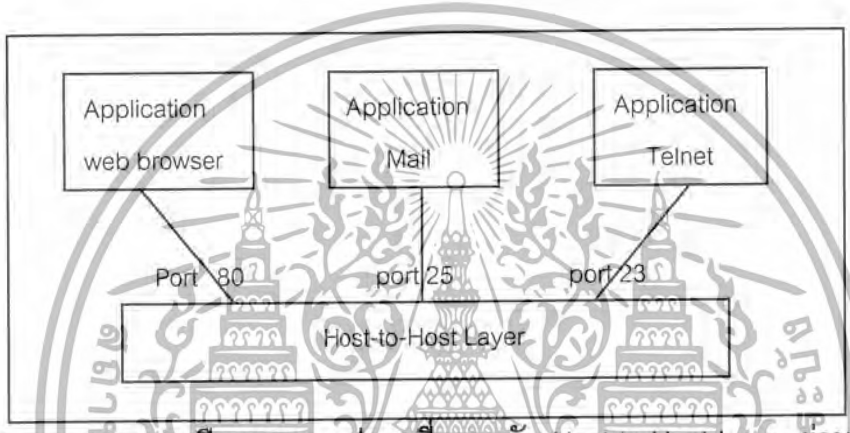
2.1.2.1 Process Layer ในชั้นนี้จะทำงาน 2 หน้าที่เทียบได้กับ Application Layer และ Presentation Layer ในชั้นนี้จะรองรับการทำงานของ application ต่างๆที่ทำงานเป็น process อยู่ในเครื่อง server ให้บริการและเครื่องที่ขอใช้บริการ หรือ client ซึ่งจะติดต่อกันผ่าน protocol เฉพาะ application อีกทีหนึ่ง การทำงานของ application ต่างๆจะอยู่ที่ Process Layer นี้ และมีการติดต่อกันตามแต่ละ protocol เฉพาะแล้วแต่ application ที่ใช้งาน จากการที่ Process Layer ของ TCP/IP รองรับให้ protocol อื่นทำงานได้หลาย process และหลาย protocol ได้พร้อมกันนั้น ทำให้ผู้ใช้สามารถเปิดโปรแกรมใช้งานได้หลายๆอย่างพร้อมกัน

protocol หลักๆที่ทำงานใน Process Layer ซึ่งผู้ใช้คุ้นเคยกันเป็นอย่างดีได้แก่ FTP (File Transfer Protocol) , Telnet , HTTP (HyperText Transfer Protocol) และ SMTP (Simple Mail Transfer Protocol) นอกจากนี้ยังมี protocol อื่นที่อยู่เบื้องหลัก ซึ่งทำงานโดยที่ผู้ใช้ไม่สามารถมองเห็นได้จากโปรแกรมหรือไม่ได้มีการใช้งานโดยตรง เช่น

- protocol DNS (Domain Name System) ที่ทำหน้าที่แปลงข้อมูลชื่อ domain name หรือชื่อเว็บไซต์ทั้งหลายให้เป็นหมายเลข IP address
- protocol SNMP (Simple Network Management Protocol) ใช้ในการควบคุมและการตรวจสอบอุปกรณ์ที่อยู่ในเครือข่าย
- protocol DHCP (Dynamic Host Configuration Protocol) ทำหน้าที่แจกจ่ายข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายที่เชื่อมต่ออยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.2 Host-to-Host Layer ชั้นนี้จะมีบทบาทในการจัดการต่อจาก Process Layer บางครั้งเรียกชั้นนี้ว่าเป็น Transport Layer ซึ่งไม่ใช่ชั้น Transport Layer ในมาตรฐาน OSI Model การทำงานของชั้นนี้จะมีการสร้าง connection หรือการเชื่อมต่อกันระหว่าง application กับ Host-to-Host Layer โดยจุดที่เชื่อมกันเพื่อรับส่งข้อมูลนี้เรียกว่า port (ไม่ได้หมายถึง port ทางฮาร์ดแวร์)หรือ socket และในแต่ละ application ก็จะสร้างการเชื่อมต่อผ่าน port ได้พร้อมกันหลาย application ซึ่งการใช้งาน port ของแต่ละ application ที่อยู่ในชั้น Process Layer จะแตกต่างกันตามหมายเลขที่กำหนดไว้ และแต่ละ protocol จะมีการใช้งาน port หมายเลขต่างๆไม่ซ้ำกัน ดังรูป 2.3

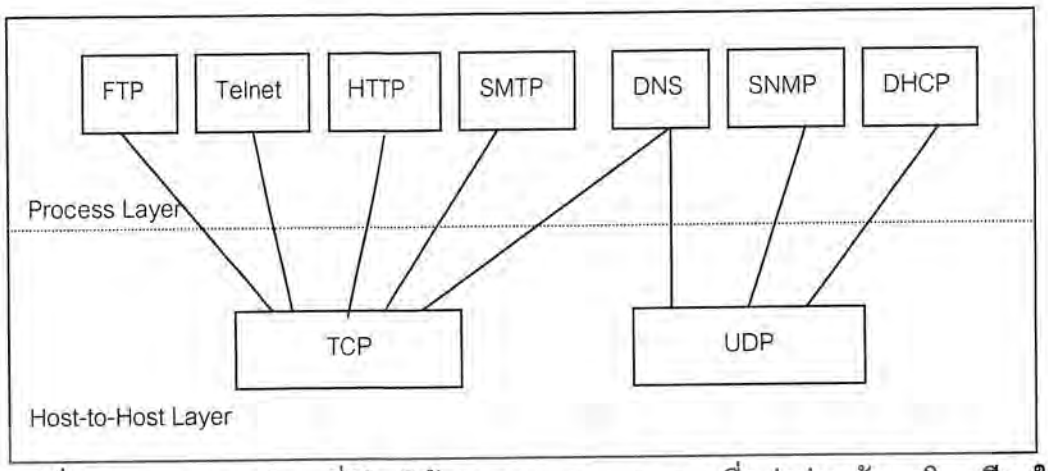


รูปที่ 2.3 applicationหรือ process ต่าง ๆสื่อสารกับ Host-to-Host Layer ผ่านทาง port

เมื่อ application ทำงานผ่าน protocol ในชั้น Process Layer จะมีการส่งผ่านข้อมูลไปยัง Host-to-Host Layer ที่ชั้นนี้จะมีการเชื่อมต่อผ่าน port ที่กำหนด ทำให้การรับส่งข้อมูลในแต่ละ protocol ทำได้ถูกต้อง ถึงแม้ว่าในเครื่อง server ที่ให้บริการจะมีการทำงานอยู่หลาย process ที่แตกต่างกันก็ตามหรือมีผู้ใช้บริการเข้ามาใช้งานพร้อมกันจำนวนมากและหลาย application ในเวลาเดียวกัน ในชั้น Host-to-Host Layer จะมี protocol ที่ทำงานอยู่ 2 protocol ที่แตกต่างกัน คือ protocol TCP และ protocol UDP(User Datagram Protocol) ในการส่งผ่านข้อมูลไปชั้นถัดๆไป เราจะเห็นว่า protocol TCP และ UDP จะถูกผนึกเข้าไปในprotocol IP อีกทีหนึ่งและส่งต่อไปยังเครือข่ายอินเตอร์เน็ตต่อไป

ตัว protocol TCP และ protocol UDP จะมี application เฉพาะเพื่อเรียกใช้งานแยกกันคือ application ที่ใช้ protocol FTP , Telnet , HTTP และ SMTP จะมีการส่งผ่านข้อมูลโดยเรียกใช้ protocol UDP และสำหรับ protocol DNS นั้นจะสามารถเรียกใช้งานได้ทั้ง TCP และ UDP ดังรูปที่ 2.4 ซึ่งเหตุผลที่มีการเรียกใช้ protocol TCP และ UDP แตกต่างกัน ก็เนื่องจากวิธีการทำงานของทั้งสอง protocolต่างกันนั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



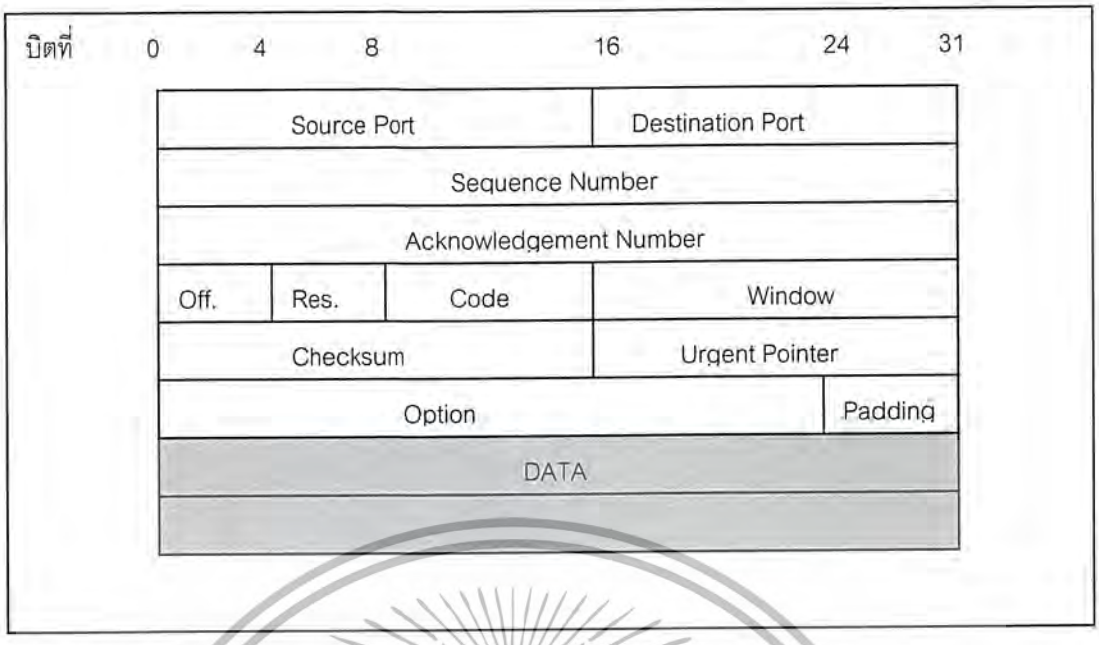
รูปที่ 2.4 process ต่างๆที่เรียกใช้ Host-to-Host layer เพื่อส่งผ่านข้อมูลโดยเรียกใช้ port ต่างกัน

1) protocol TCP protocol TCP (Transmission Control Protocol) เป็น protocol ที่มีการรับส่งข้อมูลแบบ steam oriented protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อยๆก่อน แล้วจึงจะส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป ก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูล datagram ใหม่ให้ต่อเนื่องกันและประกอบกลับเป็นข้อมูลทั้งหมดได้ ซึ่งจะแยกข้อมูลที่ไม่ถูกต้องออก ดังนั้น application หรือ process ใดที่อาศัยการส่งผ่านข้อมูลด้วย protocol TCP จะต้องใช้หน่วยความจำและขนาดของช่องสัญญาณ (bandwidth) มากกว่า UDP

การติดต่อระหว่างกันจะต้องเป็นแบบ connection-oriented คือต้องมีการสร้างการการติดต่อกันเป็นsession ทั้ง 2 ด้านเสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมกัน (full duplex) เหมือนกับการใช้โทรศัพท์ติดต่อกัน เมื่อผู้ติดต่อต้นทางเรียกให้ฝ่ายตรงข้ามรับสายแล้วจึงเริ่มการสนทนา การติดต่อกันด้วยกลไก protocol TCP เมื่อ application ต้องการส่งผ่านข้อมูลจะใช้ protocol ที่เหมาะสมในชั้น Process Layer ติดต่อไปและมีการสร้างช่องส่งข้อมูลผ่าน port ที่กำหนดเพื่อส่งผ่านข้อมูลไปยัง protocol TCP ในระหว่างการรับส่งข้อมูลนี้ protocol TCP จะเพิ่มขบวนการสอบทานข้อมูลเพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณสอบทานข้อมูล (acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น

ความน่าเชื่อถือของการส่งผ่านข้อมูลโดย protocol TCP จะมีมากกว่า แต่ก็ต้องอาศัยทรัพยากรของระบบมากกว่าในการทำงานเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 รูปแบบของ TCP packet

จากรูปที่ 2.5 รูปแบบของ TCP packet จะเห็นว่ามี field Acknowledgement Number และข้อมูล Checksum เพื่อใช้ตรวจสอบการเดินทางของข้อมูล ส่วน header มีข้อมูลมากทำให้ต้องอาศัยทรัพยากรของระบบทำงานมาก

2) Protocol UDP ใน Host-to-Host layer นอกจากจะมี protocol TCP ทำงานแล้ว ก็ยังมี protocol UDP (User Datagram Protocol) ที่มีคุณสมบัติแตกต่างกันอยู่ด้วย ในการรับส่งข้อมูลผ่าน protocol UDP จะเป็นแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมกัน (connectionless) ระหว่างเครื่อง server ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเตรียมรับข้อมูลเหมือน protocol TCP และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการรับส่งข้อมูลนั้นๆ ด้วย เนื่องจาก protocol UDP ไม่มีสัญญาณสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้ application หรือ process ใดที่ต้องอาศัย protocol UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการตรวจสอบข้อมูลขึ้นมาเอง

จากรูปที่ 2.4 จะเห็นว่า protocol ชั้นบนขึ้นไป ที่ใช้การส่งผ่านข้อมูลโดย protocol UDP เช่น protocol SNMP (ใช้ควบคุมและจัดการอุปกรณ์ในเครือข่าย) หรือ protocol DHCP (ใช้ส่งข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายได้ใช้งาน) การส่งข้อมูลเหล่านั้นไม่ต้องรับทราบหรือตรวจสอบว่าข้อมูลไปถึงปลายทางถูกต้องหรือไม่ แต่กลไกการตรวจสอบข้อมูลที่มีการรับส่งจะไปทำในขั้นตอนของ protocol ชั้นที่สูงกว่าแทน

ตัวอย่างขั้นตอนและกลไกการทำงานโดยใช้ protocol UDP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

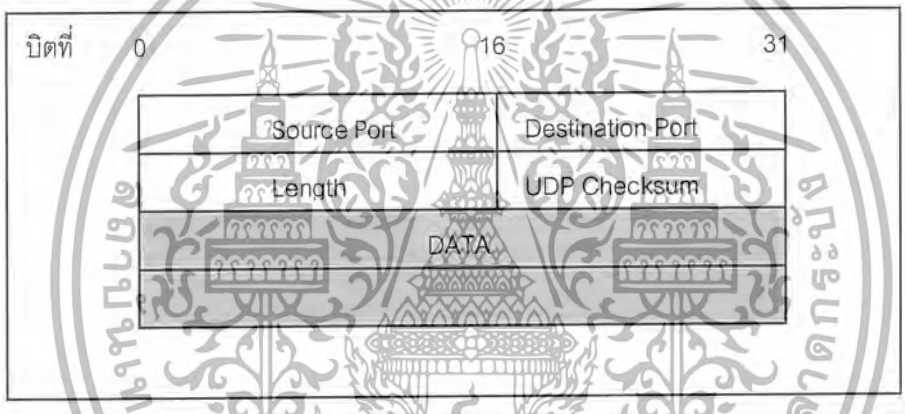
1. ในชั้นของ Process Layer เมื่ออุปกรณ์ควบคุมเครือข่ายเช่น โปรแกรม Network management ต้องการส่งข้อมูลไปยังอุปกรณ์ที่ต้องการ application นั้นจะติดต่อผ่าน protocol SNMP ในชั้น Process layer

2. Protocol SNMP จะติดต่อกับ protocol UDP ในชั้นถัดไป เพื่อขอติดต่อผ่านทาง port ที่กำหนดไว้

3. Protocol SNMP เตรียมข้อมูลที่จะส่ง รวมทั้งที่อยู่ปลายทาง

4. Protocol SNMP ส่งผ่านข้อมูลให้ protocol UDP ที่อยู่ในชั้น Host-to-Host layer

5. Protocol UDP ทำหน้าที่ผนึกข้อมูลหรือ datagram นั้นไปกับ protocolIP ในชั้นถัดลงไป เพื่อส่งข้อมูลออกจากเครื่อง



รูปที่ 2.6 รูปแบบของ UDP packet

2.1.2.3 Internetwork Layer ในระดับนี้มีหน้าที่ส่งผ่านข้อมูลในระหว่างเครือข่าย โดยมี protocol ที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆบนอินเตอร์เน็ต คือ protocol IP (Internet Protocol) นอกจากนี้ในชั้น Internetwork layer ยังมี protocol ทำงานอยู่ด้วยอีก 2 ชนิดคือ protocol Internet Control Message Protocol (ICMP) และ protocol Address Resolution Protocol (ARP)

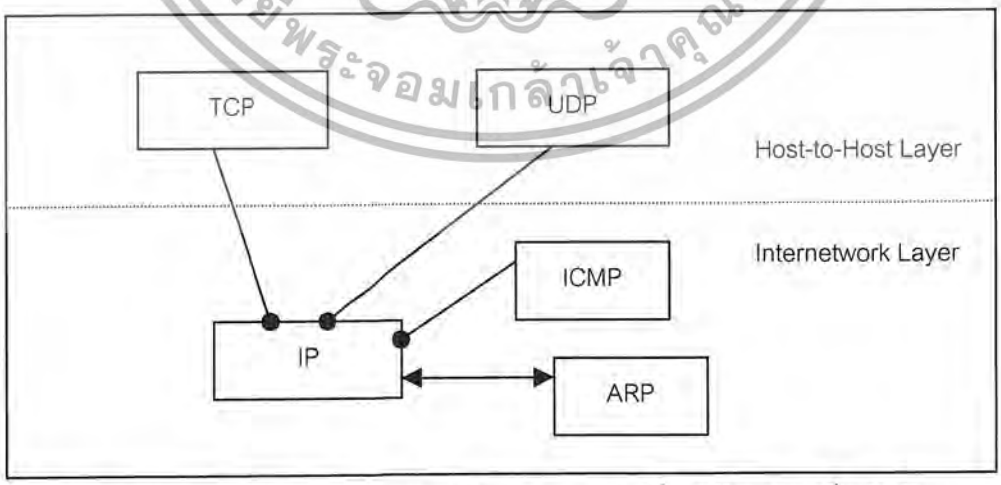
1) **protocol IP** ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host layer เพื่อส่งข้ามไปยังเครือข่ายใดๆ ได้อย่างถูกต้อง แม้ว่าจะมีเครือข่ายเชื่อมต่อกันอยู่ในอินเตอร์เน็ตเป็นล้านๆ เครือข่ายก็ตาม เนื่องจาก protocol IP มีข้อมูลตำแหน่ง IP ปลายทางที่จะส่งข้อมูลไปให้ โดยทำงานร่วมกับอุปกรณ์ Router เพื่อส่งข้อมูลข้ามเครือข่ายออกไปได้ ตัว protocol IP จะทำงานแบบ packet switching คือมีการส่งข้อมูลผ่านสวิทช์ (switch) ไปยังปลายทาง โดยข้อมูลจะเดินทางไปยังเครือข่ายต่างๆผ่านสวิทช์ไปเรื่อยๆ จนกว่าจะถึงปลายทาง ตัววงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านหรือ switch นี้อาจเป็น Gateway หรือ Router ในระบบเครือข่ายก็ได้ ซึ่งในข้อมูลของ protocol IP จะมีข้อมูลของหมายเลข IP ปลายทางที่จะส่งข้อมูลไปและเมื่อถึงเครือข่ายปลายทางแล้ว จะมีกลไกแปลงหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์ประจำเครื่องที่ถูกต้องอีกทีหนึ่งด้วย protocol ARP ตามรูปที่ 2.7 ที่จะแสดงการติดต่อกันระหว่าง protocol ในชั้นของ Host-to-Host layer และ Internetwork layer

2) protocol ICMP หน้าที่หลัก protocol ICMP (Internet Control Message Protocol) คือการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งปัญหาส่วนมากที่พบคือส่งไปไม่ได้ หรือปลายทางรับข้อมูลไม่ได้ เป็นต้น นอกจากนี้ protocol ICMP ยังถูกเรียกใช้งานจากเครื่อง server และ Router อีกด้วย เพื่อแลกเปลี่ยนข้อมูลที่ใช้ควบคุม ส่วนรูปแบบการทำงานของ protocol ICMP นั้นจะทำควบคู่กับ protocol IP ในระดับเดียวกัน และข้อความต่างๆ ที่แจ้งให้ทราบจะถูกผนึกอยู่ภายใต้ข้อมูลของ IP (IP datagram) อีกทีหนึ่ง

ข้อความที่ protocol ICMP ส่งนั้นแบ่งออกได้ 2 แบบคือ ICMP error message หรือข้อความแจ้งข้อผิดพลาด และ ICMP query หรือข้อความเรียกขอข้อมูลเพิ่ม ตัวอย่างกลไกการทำงานของ protocol ICMP เช่น เมื่อมีการส่งผ่านข้อมูลจากผู้ใช้ไปยังปลายทางที่ไม่ถูกต้องหรือขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ที่ Router จะส่งข้อความแจ้งเป็น ICMP message ที่ชื่อ destination unreachable ให้กับผู้ส่งข้อมูล นอกจากนี้ตัวข้อมูลที่แจ้งข้อความก็จะมีส่วนข้อมูล IP datagram ที่เกิดปัญหาด้วย ดังนั้นเมื่อผู้ส่งข้อมูลได้รับข้อความแจ้งแล้วก็จะทราบได้ว่าจุดที่เกิดปัญหานั้นอยู่ที่ใด



รูปที่ 2.7 protocol TCP และ UDP อาศัย protocol IP ที่อยู่ด้านล่างเพื่อส่งข้อมูลระหว่างเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น protocol ICMP จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง ping ที่เรามักใช้ทดสอบว่าเครื่อง server ที่ให้บริการหรืออุปกรณีที่ต่ออยู่ในเครือข่ายอินเทอร์เน็ตนั้นยังทำงานเป็นปกติหรือไม่ แล้วคำสั่ง ping มีการเรียกใช้งาน protocol ICMP แจ้งเป็นข้อความให้ทราบอีกต่อหนึ่ง

3) protocol ARP (Address Resolution Protocol) ถูกเรียกใช้งานโดย protocol IP เพื่อช่วยแปลงหมายเลข IP ไปเป็นหมายเลขฮาร์ดแวร์ปลายทาง ตัวอย่างเช่น เว็บ server เครื่องหนึ่งเชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ต และในการเชื่อมต่อนี้ต้องอาศัย Network Interface Card (NIC) หรือ LAN card ติดตั้งอยู่ที่ LAN card นี้เองจะมีหมายเลขเฉพาะประจำฮาร์ดแวร์ที่ไม่ซ้ำกับใคร เพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งาน protocol TCP/IP ก็ต้องมีการกำหนดหมายเลข IP address ประจำตัวเพื่อใช้อ้างอิงกัน และ protocol ARP จะทำหน้าที่แปลงค่าหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์จริงให้ในระดับการทำงานที่ Interwork layer นี้ ซึ่งกลไกการแปลงนี้เรียกว่า address resolution

2.1.2.4 Network Interface Layer เนื่องจากในด้านกายภาพของเครือข่ายนั้น มีหลายวิธีการและหลายรูปแบบในการเชื่อมต่อระบบให้เป็นเครือข่าย แต่อย่างไรก็ตามในเครือข่ายอินเทอร์เน็ตนี้ ข้อมูลหรือ IP datagram จะถูกถ่ายทอดและส่งผ่านไปยังปลายทางโดยไม่คำนึงถึงรูปแบบการเชื่อมต่อทางกายภาพ ไม่ว่าจะเป็นการใช้เครือข่ายใยแก้วนำแสงหรือเครือข่ายสาย Unshielded Twist Pair (UTP) เชื่อมต่อเป็นแบบเครือข่าย Ethernet ธรรมดาหรือเครือข่าย Token Ring, ATM, ISDN ฯลฯ ก็ตาม

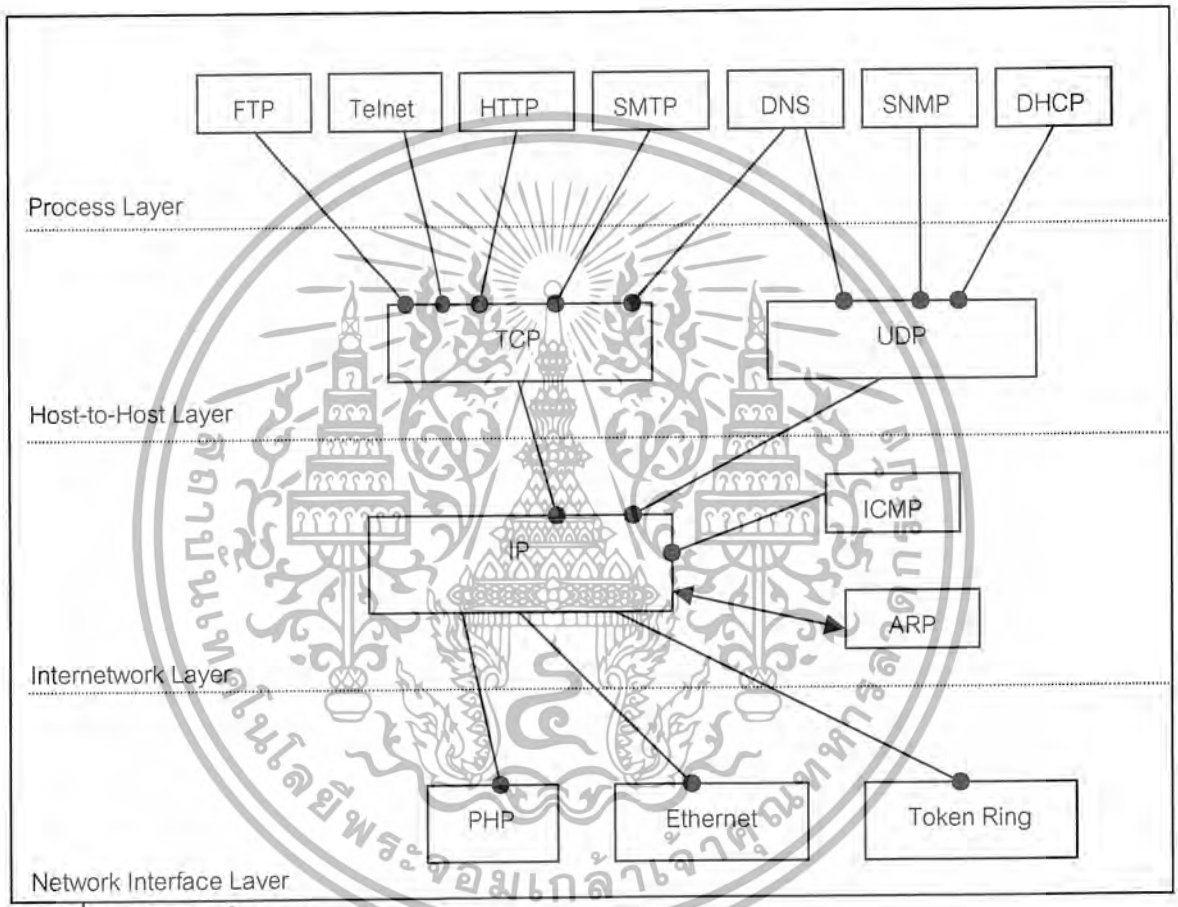
การทำงานระดับล่างสุดต่อจาก Internetwork layer จะเป็นการแปลงข้อมูล IP datagram ให้อยู่ในรูปที่เหมาะสม และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่ายต่อไปซึ่งในชั้น Network Interface layer นี้เมื่อเทียบกับมาตรฐาน OSI model แล้วจะเป็นการรวม 2 layer เข้าด้วยกันคือ Data link layer และ Physical layer กล่าวโดยสรุปคือ การทำงานในชั้นต่างๆตามโครงสร้างของ protocol TCP/IP จะมีลักษณะดังรูปที่ 2.8

กล่าวโดยสรุปก็คือ protocol TCP/IP ทำงานโดยแบ่งเป็นชั้นเทียบกับ OSI model ได้ กลไกในการทำงานของ protocol TCP/IP มี 4 ชั้น ซึ่งในชั้นแรก คือ Process layer ทำหน้าที่ติดต่อกับ application และ protocol ที่ application นั้นๆ ใช้งาน และส่งต่อมาให้ชั้น Host-to-Host layer เพื่อติดต่อกันระหว่างเครื่อง

server ให้บริการกับเครื่องผู้ขอใช้บริการ ในชั้นนี้จะมีการสร้าง session หรือการเชื่อมต่อระหว่างระบบขึ้นตามแต่ละ protocol ที่ต้องการ ต่อมาเป็นการผนึกข้อมูลไปเป็น IP datagram ที่ชั้น Internetwork layer โดยอาศัยprotocol IP เพื่อให้สามารถติดต่อส่งข้อมูลข้ามเครือข่ายปังเครือข่ายและเครื่องที่ถูกต้องได้ และสุดท้ายการส่งข้อมูลออกสู่โลกภายนอกต้องอาศัยกลไกในชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Network Interface layer เพื่อแปลงข้อมูลใหม่ เพิ่มข้อมูลที่จำเป็นในการอ้างอิงตำแหน่งและแปลงข้อมูลเป็นสัญญาณไฟฟ้าส่งออกไปยังเครือข่าย และอาจจะออกไปยัง Gateway หรือ Router เพื่อข้ามเครือข่ายออกไปยังเส้นทางที่กำหนดไว้ในอินเทอร์เน็ตต่อไป เราจะเห็นว่าในแต่ละชั้นของโครงสร้าง TCP/IP Stack มีการใช้งาน protocol ต่างๆ อยู่หนึ่ง protocol หรือมากกว่า ในแต่ละ protocol เหล่านี้ก็จะรับผิดชอบทำหน้าที่ของตน เพื่อส่งผ่านข้อมูลลงไปยังระดับล่าง และออกสู่เครือข่ายอินเทอร์เน็ตในที่สุด



รูปที่ 2.8 **โครงสร้างของ** protocol TCP/IP

2.2 Socket

socket คือ วิธีการสำหรับการสื่อสารกันระหว่างโปรแกรม client กับโปรแกรม server ในเครือข่าย socket ถูกกำหนดให้อยู่ในฐานะ end point ของการติดต่อ (connection) socket ถูกสร้างและใช้กับชุดของโปรแกรมร้องขอ หรือ socket API socket สามารถถูกใช้เพื่อสื่อสารกันระหว่าง process ที่อยู่ในเครื่องคอมพิวเตอร์เดียวกัน ในระบบ connectionless ของอินเทอร์เน็ต ที่ server ต้องจัดการกับการร้องขอจาก client หลาย ๆ ที่ และไม่มีการรักษาการเชื่อมต่อไว้นาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไป socket จะสื่อสารข้อมูลกับ socket ตัวอื่น ๆ ที่กำลังทำงานอยู่ในโดเมนของการติดต่อสื่อสารเดียวกันโดยใช้ IP ช่วย

2.2.1 **ชนิดของ socket** socket สามารถแบ่งได้เป็น 2 ชนิด คือ

1) socket แบบ stream ทำงานกับข้อมูลที่ส่งไปโดยไม่กำหนดของเขตของ record หรือไม่เป็น record และข้อมูลที่ส่งเรียกว่า a stream of bytes ซึ่งสามารถส่งได้ทั้ง 2 ทาง นั่นคือ ส่งและรับใน socket เดียวกัน และ stream จะรับประกันว่าข้อมูลถูกส่งถึงปลายทางอย่างถูกต้องโดยข้อมูลมาสลับกันและไม่มีการส่งซ้ำซ้อน โดย stream ยังสามารถจัดการส่งและรับข้อมูลขนาดใหญ่ได้ดีอีกด้วยเพราะ protocol ในชั้น Internetwork อาจแตกข้อมูลให้อยู่ในชั้นที่สมควร

การสื่อสารข้อมูลแบบ stream นั้นเป็นการสื่อสารข้อมูลแบบ connection oriented stream นั้นจะต้องมีการเชื่อมต่อกันอย่างชัดเจน ถ้า socket A ต้องการเชื่อต่อกับ socket B แล้ว socket A จะต้องมีการร้องขอเชื่อมต่อไปยัง socket B แล้ว socket B จะต้องยอมรับหรือปฏิเสธการร้องขอนั้น ๆ เหมือนกับโทรศัพท์ ซึ่งจะต้องมีผู้เรียก แล้วผู้ที่รับสายจึงจะคุยได้ และการสนทนาจะไม่มีเสียงซ้ำกันถ้าผู้พูดไม่ได้พูดซ้ำ ลำดับของเสียงที่พูดจะตรงกับลำดับของเสียงที่ผู้ฟังได้ยิน และที่สำคัญที่สุด คือ เสียงที่ผู้พูด พูดไปนั้นจะถูกส่งไปถึงผู้รับแน่นอน ด้วยเหตุนี้จึงทำให้ socket แบบ stream นี้เป็นที่นิยมมากกว่า socket แบบ datagram

2) socket แบบ datagram socket แบบนี้สนับสนุนการส่งข้อมูลสองทาง โดยไม่รับประกันข้อมูลจะถูกส่งไปถึงอย่างถูกต้องหรือไม่และข้อมูลอาจมีการสลับหรือซ้ำซ้อน แต่ขนาดของ record จะได้รับการปกป้องรักษา แม้ว่าขนาดของ record จะใหญ่กว่าขนาดที่ผู้รับกำหนด

การสื่อสารข้อมูลแบบ datagram เป็นการสื่อสารข้อมูลแบบ connectionless ไม่จำเป็นต้องสร้างการเชื่อมต่อโดยชัดเจน เพียงส่ง datagram ไปยัง socket ที่กำหนดก็จะสามารถสื่อสารข้อมูลกันได้แล้ว และ socket แบบ datagram ยังมีความสามารถในการส่งข้อมูลแบบกระจาย (Broadcasting Message)

2.2.2 **port และ address ของ socket**

1) port เป็นตัวเลขที่ใช้กำหนดค่าให้กับ socket แต่ละตัว โดยจะไม่มีเลขที่ซ้ำกันเกิดขึ้นพร้อมกัน ดังนั้น socket ของ application จะมีค่าของ port ที่ไม่ซ้ำกันเลย ทำให้สามารถ application ที่ทำงานกับ socket ได้หลายตัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) address ของ socket socket แต่ละตัวจะติดต่อกับ IP Address ในเครือข่าย โดยทั่วไป address จะเปรียบเสมือนชื่อเครื่องเช่น kmitl.ac.th หรือ " 161.246.10.21 "

2.2.3 การทำงานระหว่าง socket และ application layer

ดังที่ได้อธิบายไปแล้วว่า stream ข้อมูลถูกแบ่งเป็น datagram อย่างไร ส่งไปคอมพิวเตอร์อีกเครื่องอย่างไร และ datagram ถูกนำไปรวมกันอย่างไร อย่างไรก็ตามต้องมีวิธีที่จะเปิดการเชื่อมต่อไปยังคอมพิวเตอร์ที่ระบุ และบอกคอมพิวเตอร์ว่า คุณต้องการไฟล์ใด และควบคุมการถ่ายโอนไฟล์ งานเหล่านี้จะถูกทำโดยแอพพลิ-เคชัน protocol ซึ่งรันอยู่บน protocol TCP/IP นั่นก็คือ เมื่อ application protocol ต้องการส่งข้อความ ก็จะให้ข้อความนั้นแก่ TCP เพื่อให้ TCP ตรวจสอบให้แน่ใจว่าข้อความนั้นถึงจุดหมายปลายทาง เนื่องจาก TCP และIP ดูแลในเรื่องรายละเอียดในเรื่องเครือข่ายทั้งหมด application protocol สามารถปฏิบัติต่อเครือข่ายการเชื่อมต่อ (network connection) เสมือนว่าเป็นเพียงไบต์ stream ธรรมดา ๆ เหมือนกับเทอร์มินอลหรือสายโทรศัพท์

2.3 Threads

2.3.1 process คือ โปรแกรมที่กำลังทำงานอยู่ องค์ประกอบหลักของ process คือ

1) Data ข้อมูลที่โปรแกรมต้องการซึ่งอยู่บนหน่วยความจำ ข้อมูลนี้อาจอยู่ใน process ใด process หนึ่งหรืออาจใช้ร่วมกันกับ process อื่นๆก็ได้

2) Program counter ทำหน้าที่เก็บ address ของ machine instruction ที่จะถูกประมวลผลเป็นคำสั่งถัดไป

3) Stack เนื้อที่หน่วยความจำที่ถูกใช้งานโดย process

2.3.1.1 สถานะของ process

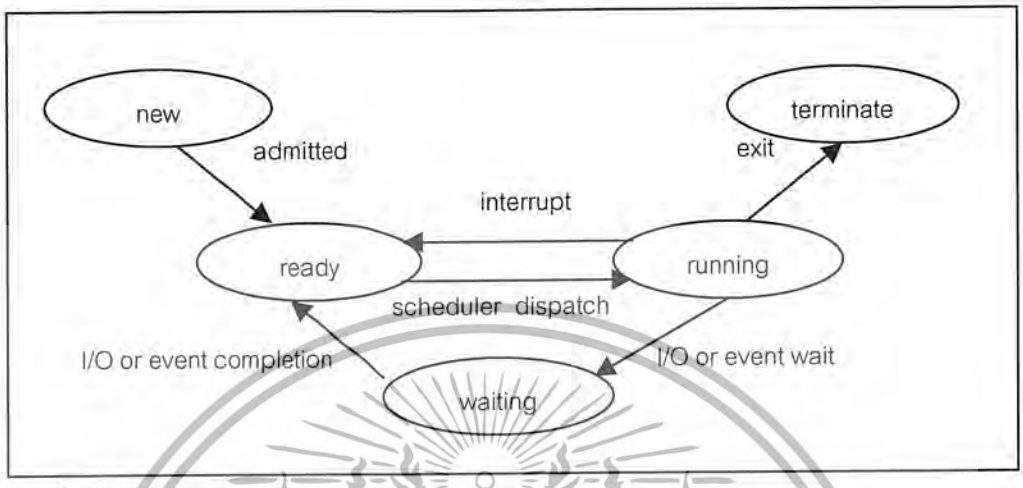
ในขณะที่แต่ละ process กำลังทำงานอยู่ จะมีการเปลี่ยนแปลงสถานะของ process

โดยแต่ละ process จะอยู่ในสถานะใดสถานะหนึ่ง ดังต่อไปนี้

- New process ใหม่กำลังถูกสร้างขึ้น
- Running process กำลังทำงานตามคำสั่งในโปรแกรม (ในหน่วยประมวลผล- กลาง)
- Waiting process กำลังรอคอยให้เหตุการณ์บางอย่างเกิดขึ้น เช่น รอให้การรับหรือส่งข้อมูลเสร็จ
- Ready process กำลังรอที่จะเข้าใช้หน่วยประมวลผล
- Terminate process เสร็จสิ้นการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขณะหนึ่งจะมี process 1 process ที่กำลังทำงานอยู่ แต่จะมีหลาย process ที่อยู่ในสถานะพร้อม (ready) หรือสถานะกำลังรอคอย (waiting) สามารถแสดงสถานะต่างๆของ process ได้ดังรูปที่ 2.9



รูปที่ 2.9 แสดงแผนภาพสถานะของ process

2.3.1.2 Process Control Block (PCB)

เป็นเนื้อที่ในหน่วยความจำที่ระบบปฏิบัติการกำหนดขึ้นเพื่อใช้เก็บข้อมูลที่สำคัญของ process หนึ่ง ๆ โดยข้อมูลเหล่านี้ได้แก่

- สถานะของ process ที่เป็นอยู่ในปัจจุบัน
- หมายเลขประจำตัวของ process
- ลำดับความสำคัญของ process
- pointer ที่ชี้ไปยังตำแหน่งที่อยู่ของ process ในหน่วยความจำ
- pointer ที่ชี้ไปยังทรัพยากรต่างๆที่ process ครอบครองอยู่
- พื้นที่ที่เก็บค่าของรีจิสเตอร์ (register save area)

PCB เป็นศูนย์กลางในการเก็บข้อมูลที่สำคัญของ process ที่ระบบปฏิบัติการต้องการ เมื่อระบบปฏิบัติการต้องการให้ process อื่นได้เข้าใช้งานซีพียูบ้าง ระบบปฏิบัติการจะเก็บข้อมูลต่าง ๆ ของ process เดิมไว้ใน PCB เพื่อที่จะนำกลับมาใช้ใหม่อีกครั้ง เมื่อ process นี้ได้กลับมาใช้หรือครอบครองซีพียู

2.3.1.3 Context Switching

จากการศึกษาในเรื่องสถานะของ process ถ้ามีหลาย process อยู่ในสถานะ ready เพื่อรอใช้ซีพียู เมื่อซีพียูว่าง process ที่อยู่ต้นคิวในสถานะ ready จะได้ครอบครองซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วเปลี่ยนสถานะไปเป็นสถานะ running process อยู่ในสถานะนี้ได้ไม่เกินเวลาที่ระบบปฏิบัติการกำหนด (quantum time) ถ้า process ต้องรอเหตุการณ์บางอย่างโดยไม่มีความจำเป็นต้องใช้ซีพียู process นั้นจะถูกย้ายไปอยู่ในสถานะ waiting แต่ถ้า process อยู่ในสถานะ running จนครบเวลา process นั้นต้องไปรอใช้งานซีพียูที่สถานะ ready เพื่อเปิดโอกาสให้ process อื่นได้ใช้งานซีพียู ไม่ว่า process จะปล่อยซีพียูเนื่องจากหมดเวลา หรือมีการรอเหตุการณ์บางอย่างซีพียูจะต้องถูกครอบครองโดย process อื่นที่อยู่ในสถานะ ready นั่นหมายความว่าซีพียูต้องเปลี่ยนการทำงานจากงานหนึ่งไปเป็นอีกงานหนึ่ง ซึ่งเหตุการณ์นี้เรียกว่า การเปลี่ยนสถานะของซีพียู (Context Switching)

การเปลี่ยนสถานะของซีพียูถือเป็น overhead ของระบบ overhead ในที่นี้หมายถึง งานที่ระบบจำเป็นต้องกระทำเพื่อจุดประสงค์บางอย่าง ระบบต้องใช้เวลาในการทำงานเหล่านี้ทำให้เสียเวลาในการทำงานของ process ของผู้ใช้ไปด้วย ดังนั้นในระบบจึงไม่ควรมี overhead ของระบบมากเกินไป สำหรับการเปลี่ยนสถานะของซีพียูจะเสีย overhead สำหรับงานหลัก 3 งาน ดังนี้

1) เก็บค่า register และสถานะของ process ไว้ใน PCB ของ process ที่กำลังปล่อยซีพียู
2) เลือก process ที่อยู่ในสถานะ ready ให้เข้าใช้งานซีพียู
3) โหลดค่า register และสถานะของ process จาก PCB ของ process ใหม่

2.3.2 Threads

Threads มีหลายประเภท ซึ่งแต่ละประเภทก็จะมีคุณสมบัติและวิธีการใช้งานที่แตกต่างกัน ในหัวข้อนี้จะอธิบายถึง Threads ใน 3 ประเภทที่สำคัญ

1) Kernel Threads เป็น Thread ของตัว kernel เองซึ่งไม่เกี่ยวข้องกับ process ของผู้ใช้ จะถูกสร้างและถูกทำลายโดยตัว kernel เอง

2) Lightweight Processes (LWP) เป็น thread ของ process ที่มี kernel ช่วยในการทำงาน โดยทำงานในระดับที่สูงกว่า kernel thread ดังนั้นระบบที่รองรับ LWP ต้องรองรับการทำงานของ kernel thread เสียก่อน ทุก ๆ process อาจมี LWP มากกว่า 1 ตัวก็ได้ โดยแต่ละ LWP จะมี kernel thread ช่วยในการทำงาน

3) User Threads เป็น thread ที่อยู่ในระดับผู้ใช้ทั้งหมด ตัว kernel จะไม่รับรู้เกี่ยวกับ user thread ใด ๆ ทั้งสิ้น ซึ่งการทำงานของ thread ในระดับนี้ ต้องมี thread library เข้ามาเกี่ยวข้อง โดยใน library จะเตรียมฟังก์ชันสำหรับการสร้างการเข้าจังหวะการทำงาน การจัดตารางการทำงานของ thread และการจัดการอื่น ๆ เกี่ยวกับ thread โดยปราศจากการช่วยเหลือใด ๆ จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

kernel การโต้ตอบกันระหว่าง thread จะไม่เกี่ยวข้องกับ kernel ดังนั้น user thread จึงทำงานได้เร็วมาก ซึ่งการทำ thread นี้เรียกว่า synchronous programming paradigm นั่นคือ เมื่อถึงงานที่ต้องการแล้วค่อยสร้าง thread ขึ้นมา

ข้อแตกต่างระหว่าง thread กับ process

1) Dependency แต่ละ thread ของ process เดียวกันจะทำงานที่แตกต่างกัน แต่มีความเกี่ยวข้องกันบางอย่างและต้องทำงานอยู่ภายใน environment เดียวกัน จึงอาจใช้ตัวแปรบางตัวร่วมกัน หรืออาจใช้โปรแกรมบางส่วนร่วมกัน แต่สำหรับ process จะไม่มีความเกี่ยวข้องกันเลย เพราะแต่ละ process จะมี environment ของตัวเอง

2) การติดต่อสื่อสาร การติดต่อระหว่าง thread ทำได้ง่าย โดยอาจใช้ตัวแปรร่วมกัน แต่สำหรับ process จะไม่มีการใช้ตัวแปรร่วมกัน เนื่องจากเมื่อมีการทำ context switching จะเปลี่ยน environment ของ process ด้วย วิธีเดียวที่ process จะส่งค่าให้แก่กันได้ คือทำผ่านทางไฟล์ (เรียกว่า pipe) ซึ่งเป็นวิธีที่สิ้นเปลืองมากและช้ากว่าการอ่านหรือกำหนดค่าตัวแปรอย่างมาก

3) Context Switching เมื่อ process หนึ่งถูกนำเข้าทำงาน หาก process นั้นมีมากกว่าหนึ่งthread ก็จะมีการสลับกันทำงานระหว่าง thread เหล่านั้น โดยไม่ต้องเปลี่ยน environment และ PCB ของ process ทำให้การสลับการทำงานระหว่าง thread เป็นภาระต่อเครื่องน้อยกว่าการทำ context switching ของการเปลี่ยน process ส่งผลให้ thread เหล่านั้นมีเวลาทำงานมากขึ้น

2.4 XML

XML (the Extensible Markup Language) เป็นภาษา Markup ที่เป็น text-based ที่ใช้เป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเตอร์เน็ตในปัจจุบัน ผู้ที่ทำหน้าที่รับผิดชอบและกำหนดมาตรฐานของ XMLคือ World Wide Web Consortium (W3C) ทุกวันนี้ ภาษา XML เป็นแนวทางหนึ่งที่จะสร้างคุณค่าให้กับผู้ใช้อิน-เทอร์เน็ต และเป็นหนทางที่จะสร้างบทบาทการประยุกต์ใช้งานบนเครือข่ายและเป็นภาษาที่กำลังเป็นที่นิยมมากเพราะกำลังกลายมาเป็นภาษามาตรฐานสำหรับการพัฒนาระบบการแลกเปลี่ยนเอกสารข้อมูลอิเล็กทรอนิกส์ซึ่งจะนำศักยภาพใหม่ของโทคโนโลยีที่มีการพัฒนาอย่างไม่หยุดยั้ง เพื่อเข้ามารองรับในการประมวลผลบน อินเทอร์เน็ต โดยเป็นเทคโนโลยีระบบเปิดที่เน้นการสื่อสารระหว่างผู้คนบนอินเทอร์เน็ตและ application ซึ่งมีความสามารถทำงานได้ระหว่าง platform ที่แตกต่างกัน XML (eXtensible Markup Language) ถูกออกแบบมาเพื่อให้ผู้สร้างเอกสารสามารถนำไปใช้งานในรูปแบบวิธีการที่ง่าย มีความชัดเจนและเป็นเซตย่อยของ SGML (Standard Generalized Markup Language)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

51778

ซึ่งเป็นภาษาที่นิยมใช้และได้รับการพัฒนาให้มีประสิทธิภาพสูงสุดในการทำงานบนเว็บ โดย XML จะประกอบด้วย 3 ส่วนพื้นฐานด้วยกัน คือ เอกสารข้อมูล (Data document) เอกสารนิยามความหมาย (definition document) และนิยามภาษา (definition language)

XML ใช้มาตรฐานเปิดจึงทำงานได้ในทุก platform และใช้ภาษาคอมพิวเตอร์ใดเขียนก็ได้ XML เป็นโครงสร้างแทนข้อมูล หัวใจหลักของ XML คือ การแยกข้อมูลออกจาก process XML เป็นการแสดงมาตรฐานเปิดสำหรับการแลกเปลี่ยนข้อมูลจากอุปกรณ์และหน่วยผู้ควบคุมผ่านเครือข่ายอินทราเน็ต ซึ่งมีแนวโน้มว่าในระหว่างที่รวมตัวกันอุปกรณ์คงต้องมีปฏิกิริยาต่อกันน้อย โดยพื้นฐานแล้วถ้าข้อมูลเป็น web-based สามารถยินยอมให้เทคโนโลยี Java เป็นตัว process บน browser ได้ เพราะนักพัฒนาสามารถสร้าง application แบบไม่ขึ้นกับ platform ด้วยภาษา Java และให้ XML แลกเปลี่ยนข้อมูลระหว่างผู้ใช้เครื่องรับโทรศัพท์ระบบเซลล์ลูล่าร์วันนี้ในทางปฏิบัติเครื่องมือบางชนิดสามารถแลกเปลี่ยนข้อมูลโดยใช้ XML พร้อมกับเครือข่ายควบคุมฐานข้อมูล โดยมีความสามารถในแง่ของการนำไปใช้ในการสร้างธุรกรรมการค้าทางอิเล็กทรอนิกส์ เพื่อช่วยลดต้นทุน

2.4.1 ข้อแตกต่างระหว่าง XML กับ HTML

XML ใช้ในการอธิบายข้อมูล ว่าข้อมูลนั้นคืออะไร สำหรับ html นั้นใช้อธิบายการแสดงผล คือบอกว่าข้อมูลนั้นจะแสดงผลในหน้าตาแบบไหน ทั้งคู่จึงมีลักษณะใช้เสริมกัน

สำหรับ tag ใน XML นั้นจะมีรูปแบบคล้ายกับ html แต่ต่างกันตรงที่ tag ของ html นั้นเกิดจาก elementที่มีการนิยามสำเร็จรูปจากองค์กรที่วางมาตรฐานแล้วเราสามารถนำเอาไปใช้งานได้เลย ในขณะที่ tag ของ XML เราต้องทำการนิยาม element ก่อนถึงจะนำไปใช้ได้

2.4.2 กฎเกณฑ์และการใช้ XML

ตัวอย่างเอกสาร XML

```
1.<?xml version="1.0" encoding="windows874"?>
2.<note>
3.<to>Mr. kridsada</to>
4.<from>p'big</from>
5.<heading>Hello !! What's up mate</heading>
6.<body>It's nice to meet you</body>
7.</note>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรทัดที่1. เป็นการประกาศว่าเป็นเอกสาร XML เวอร์ชันที่ 1

บรรทัดที่ 2. คือ element ตัวแรกหรือตัวแม่ของเอกสาร หรือที่เราเรียกว่า root element นั่นเอง

บรรทัดที่ 3- 6. มี element 4 ตัวคือ to,from,heading,body ซึ่งเป็น element ลูกของ Note ซึ่งเป็น element แม่

บรรทัดที่ 7. คือ tag ปิดของ element แม่นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 รูปแบบในการสื่อสารของโปรแกรม Instant Message for Organization (protocol)

รูปแบบการติดต่อของโปรแกรม Instant Message for Organization ที่ทำการพัฒนาบน application layer ของ OSI Model จะทำงานลักษณะไม่มีการเชื่อมต่อถาวร (connectionless) เมื่อต้องการติดต่อกันครั้งหนึ่งก็เชื่อมต่อครั้งหนึ่ง เมื่อทำงานตามที่ต้องการสำเร็จก็จะตัดการเชื่อมต่อ เมื่อต้องการติดต่อกันอีกก็ทำการเชื่อมต่อใหม่ โดย protocol จะทำงานอยู่บน protocol TCP/IP ซึ่งจะให้เกิดความเชื่อถือในการส่งข้อมูลเพราะว่า TCP ทำงานลักษณะ connection oriented โดยรูปแบบของคำสั่งของโปรแกรม Instant Message for Organization ที่จะทำการพัฒนาจะมีรูปแบบ ดังนี้

1. คำสั่งที่ส่งจาก client ไปยัง server
2. คำสั่งที่ส่งจาก server มายัง client
3. คำสั่งที่ส่งกันระหว่าง client

โดยแต่ละประเภทมีรูปแบบของคำสั่ง ดังนี้

1) คำสั่งที่ส่งจาก client ไปยัง server

ทุกคำสั่งที่ส่งจาก client ไปยัง server จะมีโครงสร้างของข้อมูลเริ่มต้น(header) ดังนี้

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	VERSION	เวอร์ชันของ protocol โดยไบต์สูงจะบอกเวอร์ชันย่อย ส่วนไบต์ต่ำจะบอกเวอร์ชันหลัก โดยในที่นี้เป็นเวอร์ชัน 1.0 จะมีค่าเป็น 0x0001
4	SESSION_ID	เป็นตัวเลขที่ใช้ยืนยันว่าผู้ใช้ได้ผ่านการพิสูจน์ตัวบุคคลแล้ว โดยจะใช้ในการอ้างอิงในการติดต่อครั้งต่อไปด้วย
2	COMMAND	ประเภทของคำสั่งที่ส่งมา
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่เป็นผู้ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CMD_ACK

เป็นคำสั่งที่จะส่งเพื่อเป็นการยืนยันว่าได้รับคำสั่งก่อนหน้าที่ส่งมาแล้วและใช้ในการตอบคำถาม ใช่/ไม่ใช่ ด้วย

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	VALUE	ตัวเลขสุ่ม ในกรณีที่ใช้ในการตอบคำถาม ใช่/ไม่ใช่ ค่านี้จะเป็น 0 ถ้าเป็นการปฏิเสธ และเป็นค่าอื่นที่ไม่ใช่ 0 ถ้าตอบตกลง

CMD_SEND_OFFLINE_MESSAGE

เป็นคำสั่งที่ใช้สำหรับส่งข้อความไปยังผู้ใช้ที่ไม่ได้เชื่อมต่อกับเครือข่าย ผู้รับจะได้รับเมื่อมีการเชื่อมต่อกับเครือข่ายครั้งต่อไป โดยข้อความที่ส่งมีความยาวสูงสุดได้ 255 ตัวอักษร

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้
1	MESSAGE_LENGTH	ความยาวของข้อความที่ส่งมา
Variation	MESSAGE	ข้อความที่ส่ง

CMD_LOGIN

เป็นคำสั่งที่สำหรับการ log-in เข้าสู่เครือข่าย

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	PORT	หมายเลขของ port TCP ที่ client เปิดเพื่อรอการเชื่อมต่อ
4	IP_ADDRESS	IP address ของ client
1	PASSWORD_LENGTH	ความยาวของรหัสผ่าน
Variation	PASSWORD	รหัสผ่าน

CMD_INFO_REQUEST

เป็นคำสั่งสำหรับขอดูรายละเอียดของผู้ใช้คนอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่ต้องการขอดูรายละเอียด

CMD_CHANGE_PASSWORD

เป็นคำสั่งที่ใช้สำหรับเปลี่ยนรหัสผ่าน

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	PASSWORD_LENGTH	ความยาวของรหัสผ่านเก่า
Variation	PASSWORD	รหัสผ่านเก่า
1	NEW_PASSWORD_LENGTH	ความยาวของรหัสผ่านใหม่
Variation	NEW_PASSWORD	รหัสผ่าน ใหม่

CMD_STATUS_CHANGE

เป็นคำสั่งที่ใช้สำหรับแจ้งสถานะการเชื่อมต่อให้กับ server ทราบ

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	STUTUS	ใช้บอกสถานะของผู้ใช้

STATUS ใช้บอกสถานะของผู้ใช้ โดยสถานะที่เป็นไปได้ทั้งมี ดังนี้

- 0x00 ใช้แทนสถานะ Offline
- 0x01 ใช้แทนสถานะ Online

CMD_FILE

เป็นคำสั่งที่ใช้สำหรับแจ้งรายชื่อไฟล์ทั้งหมดที่เปิดให้ใช้ร่วมกันให้ server ทราบ โดยจะส่งอยู่ในรูปของ XML

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	FILE_LIST_LENGTH	ความยาวของ XML stream
Variable	FILE_LIST	XML stream ของรายชื่อไฟล์พร้อมที่อยู่ของไฟล์ที่อนุญาตให้มีการใช้ร่วมกัน

โดยมี XML DTD ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT file (name, size)>
<!ELEMENT files (file*)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT size (#PCDATA)>
```

CMD_SEARCH_FILE

เป็นคำสั่งที่ใช้สำหรับการค้นหาไฟล์ที่ต้องการว่าตั้งอยู่ที่ใด

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	FILE_NAME_LENGTH	ความยาวของชื่อไฟล์ที่ต้องการค้นหา
Variable	FILE_NAME	ชื่อไฟล์ที่ต้องการค้นหา

CMD_REQUEST_ADD_USER

เป็นคำสั่งที่ใช้สำหรับการร้องขอเพิ่มชื่อผู้ใช้คนอื่นเข้ามาใน Contact List

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่ต้องการขอเพิ่มเข้ามาใน Contact List

CMD_EDIT_CONTACT_LIST

เป็นคำสั่งการจัดรายชื่อของผู้ใช้ในแต่ละกลุ่ม ใน Contact List

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่ต้องการ
1	GROUP_LENGTH	บอกความยาวของชื่อกลุ่ม
Variable	GROUP	ชื่อกลุ่มที่ต้องการให้ผู้ใช้อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CMD_REMOVE_USER

เป็นคำสั่งที่ใช้สำหรับลบผู้ใช้ออกจาก Contact List

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้
Variable	UESR_NAME	ชื่อผู้ใช้ที่ต้องการ

CMD_ADD_GROUP

เป็นคำสั่งที่ใช้สำหรับการสร้างกลุ่มใน Contact List ขึ้นมาใหม่

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	GROUP_LENGTH	บอกความยาวของชื่อกลุ่ม
Variable	GROUP	ชื่อกลุ่มที่ต้องการ

CMD_REMOVE_GROUP

เป็นคำสั่งที่ใช้สำหรับลบกลุ่มออกจาก Contact List

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	GROUP_LENGTH	บอกความยาวของชื่อกลุ่ม
Variable	GROUP	ชื่อกลุ่มที่ต้องการ

CMD_ACK_REQUEST

เป็นคำสั่งที่ใช้ตอบอนุญาต ในกรณีที่มีผู้ขอเพิ่มรายชื่อผู้ใช้เข้าไปใน contact list

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวชื่อผู้ใช้ที่ขอเพิ่มรายชื่อ
Variable	USER_NAME	ชื่อผู้ใช้ที่ขอเพิ่มรายชื่อ

CMD_NACK_REQUEST

เป็นคำสั่งที่ใช้ตอบไม่อนุญาต ในกรณีที่มีผู้ขอเพิ่มรายชื่อผู้ใช้เข้าไปใน contact list

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวชื่อผู้ใช้ที่ขอเพิ่ม
Variable	USER_NAME	ชื่อผู้ใช้ที่ขอเพิ่มรายชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **คำสั่งที่ส่งจาก server มายัง client**

ทุกคำสั่งที่ส่งจาก server มายัง client จะมีโครงสร้างของข้อมูลเริ่มต้น(header) ดังนี้

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	VERSION	เวอร์ชันของ protocol โดยไบต์สูงจะบอกเวอร์ชันย่อย ส่วนไบต์ต่ำจะบอกเวอร์ชันหลัก โดยในที่นี้เป็นเวอร์ชัน 1.0 จะมีค่าเป็น 0x0001
4	SESSION_ID	เป็นตัวเลขที่ใช้ยืนยันว่าผู้ใช้ได้ผ่านการพิสูจน์ตัวบุคคลแล้ว โดยจะใช้ในการอ้างอิงในการติดต่อครั้งต่อไปด้วย
2	COMMAND	ประเภทของคำสั่งที่ส่งมา
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่ติดต่อด้วย

SRV_ACK

เป็นคำสั่งที่จะส่งเพื่อเป็นการยืนยันว่าได้รับคำสั่งก่อนหน้าที่ส่งมาแล้วและใช้ในการตอบคำถามใช่/ไม่ใช่ ด้วย

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	VALUE	ตัวเลขสุ่ม ในกรณีที่ใช้ในการตอบคำถามใช่/ไม่ใช่ ค่านี้จะเป็น 0 ถ้าเป็นการปฏิเสธ และเป็นค่าอื่นที่ไม่ใช่ 0 ถ้าตอบตกลง

SRV_INFO_RESPONSE

เป็นคำสั่งที่ตอบรับคำสั่ง CMD_INFO_REQUEST โดยจะมีรายละเอียดของผู้ใช้ที่ร้องขอตอบกลับมาอยู่ในรูป XML

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	INFO_LENGTH	ความยาวของ XML stream
Variable	INFO	XML stream ของรายละเอียดข้อมูลของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยมี XML DTD ดังนี้

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT address (#PCDATA)>
<!ELEMENT info (username, name, surname, address, tel)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT surname (#PCDATA)>
<!ELEMENT tel (#PCDATA)>
<!ELEMENT username (#PCDATA)>
```

SRV_LOGIN_REPLY

เป็นคำสั่งที่ server จะตอบกลับมาเมื่อ client ส่งคำสั่ง CMD_LOGIN แล้วชื่อผู้ใช้และรหัสผ่านถูกต้อง คำสั่งนี้จะส่ง SESSION_ID ที่ใช้อ้างอิงในการติดต่อครั้งต่อไปมาให้ด้วย โดยอยู่ที่ส่วน โครงสร้างของข้อมูลเริ่มต้น (header)

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	KEEP_ALIVE_TIME	บอกระยะเวลาที่ client จะต้องส่งคำสั่ง CMD_KEEP_ALIVE ไปให้กับ server หน่วยเป็นวินาที

SRV_BAD_PASSWORD

เป็นคำสั่งที่ server จะตอบกลับมาเมื่อ client ส่งคำสั่ง CMD_LOGIN แล้วชื่อผู้ใช้และรหัสผ่านไม่ถูกต้อง

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	VALUE	ตัวเลขสุ่มหรือจะเป็นค่าอะไรก็ได้

SRV_USER_STATUS

เป็นคำสั่งที่ใช้สำหรับแจ้งสถานะของผู้ใช้อื่นที่อยู่ใน Contact List ของผู้ใช้ให้ผู้ใช้ทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่เปลี่ยนสถานะ
2	PORT	หมายเลขของ port TCP ที่ client เปิดเพื่อรอการเชื่อมต่อ
4	IP_ADDRESS	IP address ของ client
1	STUTUS	ใช้บอกสถานะของผู้ใช้

STATUS ใช้บอกสถานะของผู้ใช้ โดยสถานะที่เป็นไปได้ทั้งมี ดังนี้

- 0x00 ใช้แทนสถานะ Offline
- 0x01 ใช้แทนสถานะ Online

SRV_RECEIVE_OFFLINE_MESSAGE

เป็นคำสั่งที่ใช้ส่ง offline message ที่ผู้ใช้ได้รับ

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ของผู้ส่ง
4	RECEIVE_TIME	เวลาที่ได้รับข้อความมีหน่วยเป็นวินาที เริ่มตั้งแต่ 1 มกราคม ค.ศ. 1970
1	MESSAGE_LENGTH	ความยาวของข้อความ
Variation	MESSAGE	ข้อความที่ได้รับ

SRV_FILE_RESULT

เป็นคำสั่งที่ server ตอบกลับเมื่อไคลเอ็นส่งคำสั่ง CMD_SEARCH_FILE โดยตอบกลับมาอยู่ในรูปของ XML ที่จะบอกที่ตั้งของไฟล์

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	SEARCH_RESULT_LENGTH	ความยาวของ XML stream
Variable	SEARCH_RESULT	XML stream ของผลลัพธ์ในการค้นหาไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยมี XML DTD ดังนี้

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT results (result*)>
<!ELEMENT result (username, filename, filesize, ip, port)>
<!ELEMENT username (#PCDATA)>
<!ELEMENT filename (#PCDATA)>
<!ELEMENT filesize (#PCDATA)>
<!ELEMENT ip (#PCDATA)>
<!ELEMENT port (#PCDATA)>
```

SRV_SEND_CONTACT

เป็นคำสั่งที่ใช้แจ้งรายละเอียดของ Contact List ให้ผู้ใช้ทราบ อยู่ในรูปของ XML

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
Variable	CONTACT	XML stream ของ Contact List

โดยมี XML DTD ดังนี้

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT contact (#PCDATA)>
<!ATTLIST contact name CDATA #REQUIRED>
<!ELEMENT contacts (group+)>
<!ELEMENT group (contact*)>
<!ATTLIST group name CDATA #REQUIRED>
```

SRV_REQUEST_TO_ADD

เป็นคำสั่งที่จะแจ้งให้ผู้ใช้ทราบว่ามีการร้องขอในการเพิ่มชื่อของผู้ใช้ โดยผู้ใช้คนอื่น

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ของผู้ที่ร้องขอ

3. คำสั่งที่ส่งกันระหว่าง client

ทุกคำสั่งที่ส่งกันระหว่าง client จะมีโครงสร้างของข้อมูลเริ่มต้น(header) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	VERSION	เวอร์ชันของ protocol โดยไบต์สูงจะบอกเวอร์ชันย่อย ส่วนไบต์ต่ำจะบอกเวอร์ชันหลัก โดยในที่นี้เป็นเวอร์ชัน 1.0 จะมีค่าเป็น 0x0001
2	COMMAND	ประเภทของคำสั่งที่ส่งมา
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่เป็นผู้ส่ง

PEER_SEND_MESSAGE

เป็นคำสั่งสำหรับการส่งข้อความหากันของผู้ใช้ที่เชื่อมต่ออยู่กับเครือข่าย

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	MESSAGE_LENGTH	ความยาวของข้อความที่ส่งมา
Variation	MESSAGE	ข้อความที่ส่ง

PEER_REQUEST_FILE

เป็นคำสั่งสำหรับการร้องขอไฟล์ โดยตำแหน่งไฟล์ที่ร้องขอได้มาจากการผลลัพธ์ของคำสั่ง

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
1	FILE_NAME_LENGTH	ความยาวของชื่อไฟล์ที่ต้องการ
Variable	PATH	ชื่อไฟล์ที่ต้องการ

PEER_RESPONSE_FILE

เป็นคำสั่งที่ใช้ตอบกลับคำสั่ง PEER_REQUEST_FILE ว่าอนุญาตให้มีการ download ไฟล์หรือไม่

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
2	Port	ถ้าค่านี้ไม่เท่ากับ 0 จะเป็นค่าพอร์ตที่จะใช้ในการส่งไฟล์ ถ้าค่าเท่ากับ 0 แสดงว่าไม่อนุญาตให้มีการดาวโหลดไฟล์
1	FILE_NAME_LENGTH	ความยาวของชื่อไฟล์ที่ต้องการ
Variable	PATH	ชื่อไฟล์ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ เมื่อเจ้าของไฟล์อนุญาตให้ดาวโหลดไฟล์ เจ้าของไฟล์จะทำการเปิดพอร์ตสื่อสารเพื่อรอให้ผู้ต้องการดาวโหลดไฟล์เชื่อมต่อเข้ามา เมื่อมีการติดต่อเข้าก็จะเริ่มการส่งข้อมูลโดยฝ่ายของผู้ที่ต้องการดาวโหลดจะทำการส่งข้อความพร้อมในการรับข้อมูล ต่อจากนั้นเจ้าของไฟล์จะทำการแจ้งให้ทราบถึงจำนวนของ block ข้อมูลที่จะส่งมาทั้งหมดโดยแต่ละ block มีขนาด 4,096 ไบต์ ผู้ขอดาวโหลดส่งข้อความพร้อมอีกครั้ง เจ้าของไฟล์ส่งข้อมูลทีละ block ผู้ดาวโหลดจะแจ้งข้อความได้รับข้อมูลแล้วทุกครั้ง ทำขั้นตอนนี้ไปเรื่อยๆ จนได้รับข้อมูลสมบูรณ์

PEER_CHAT_INVITE

เป็นคำสั่งที่ผู้เริ่มการสนทนาจะส่งคำเชิญชวนไปยังผู้ใช้คนอื่นเพื่อเข้ามาร่วมการสนทนา โดยผู้ที่สร้างการสนทนาขึ้นมาเท่านั้นถึงจะสามารถใช้คำสั่งนี้ได้

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	CHAT_ID	เป็นเลขอ้างอิงในการสนทนาว่าเป็นการสนทนาอันใดโดยมีค่าตั้งแต่ 1 ขึ้นไป

PEER_CHAT_RESPONSE

เป็นคำสั่งที่ใช้ตอบกลับคำเชิญชวนในการสนทนา

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	CHAT_ID	เป็นเลขอ้างอิงในการสนทนาว่าเป็นการสนทนาอันใด ถ้าเป็น 0 แสดงว่าปฏิเสธการสนทนา

PEER_CHAT_MESSAGE

เป็นคำสั่งที่ใช้สำหรับการส่งข้อความเข้าไปในห้องสนทนา

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	CHAT_ID	เป็นเลขอ้างอิงในการสนทนาว่าเป็นการสนทนาอันใด
1	MESSAGE_LENGTH	ความยาวของข้อความที่ส่ง
Variable	MESSAGE	ข้อความที่ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PEER_CHAT_EXIT

เป็นคำสั่งสำหรับที่ผู้เข้าร่วมสนทนาส่งไปยังเครื่องเจ้าของการสนทนาว่าออกจากการสนทนา

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	CHAT_ID	เป็นเลขอ้างอิงในการสนทนาว่าเป็นการสนทนาอันใด

PEER_CHAT_INFO_ADD_USER

เป็นคำสั่งที่ใช้กระจายข้อมูลการเข้าร่วมการสนทนาของผู้ใช้รายใหม่ไปยังผู้ใช้ที่อยู่ในห้องสนทนาทุกคน โดยเจ้าของห้องสนทนาจะเป็นผู้ส่งคำสั่งนี้

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	CHAT_ID	เป็นเลขอ้างอิงในการสนทนาว่าเป็นการสนทนาอันใด
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่เข้าร่วมการสนทนา
1	NAME_LENGTH	บอกความยาวของชื่อจริงของผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อจริงของผู้ใช้ที่เข้าร่วมการสนทนา

PEER_CHAT_INFO_REMOVE_USER

เป็นคำสั่งที่ใช้กระจายข้อมูลการออกจากการสนทนาของผู้ใช้รายใหม่ไปยังผู้ใช้ที่อยู่ในห้องสนทนาทุกคน โดยเจ้าของห้องสนทนาจะเป็นผู้ส่งคำสั่งนี้

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	CHAT_ID	เป็นเลขอ้างอิงในการสนทนาว่าเป็นการสนทนาอันใด
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่ออกจากการสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PEER_CHAT_INFO_MESSAGE

เป็นคำสั่งที่ใช้กระจายข้อความที่ผู้ใช้ที่เข้าร่วมการสนทนาส่งเข้ามา ไปยังผู้ใช้คนอื่นไปยังผู้ใช้คนอื่นที่อยู่ในห้องสนทนา

ความยาวของข้อมูล (ไบต์)	ชื่อ	คำอธิบาย
4	CHAT_ID	เป็นเลขอ้างอิงในการสนทนาว่าเป็นการสนทนาอันใด
1	USER_NAME_LENGTH	บอกความยาวของชื่อผู้ใช้ที่จะส่งตามมา
Variation	USER_NAME	ชื่อผู้ใช้ที่เป็นผู้ส่งข้อความ
1	MESSAGE_LENGTH	ความยาวของข้อความที่ส่ง
Variable	MESSAGE	ข้อความที่ส่ง

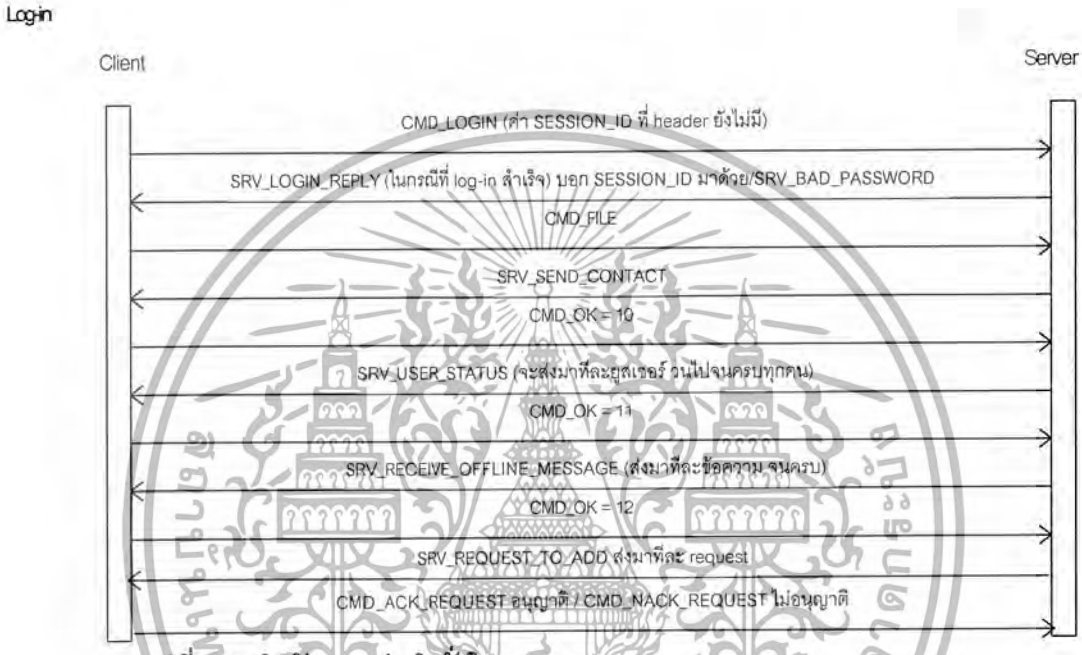
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ลำดับการส่งคำสั่ง

การทำงานของโปรแกรมจะมีการทำงานสองลักษณะคือการทำงานแบบ client/server และการทำงานแบบ Peer to Peer โดยลำดับการส่งคำสั่งแยกตามการทำงานหน้าที่ต่างๆ ดังนี้

- **การ log-in เข้าระบบ**

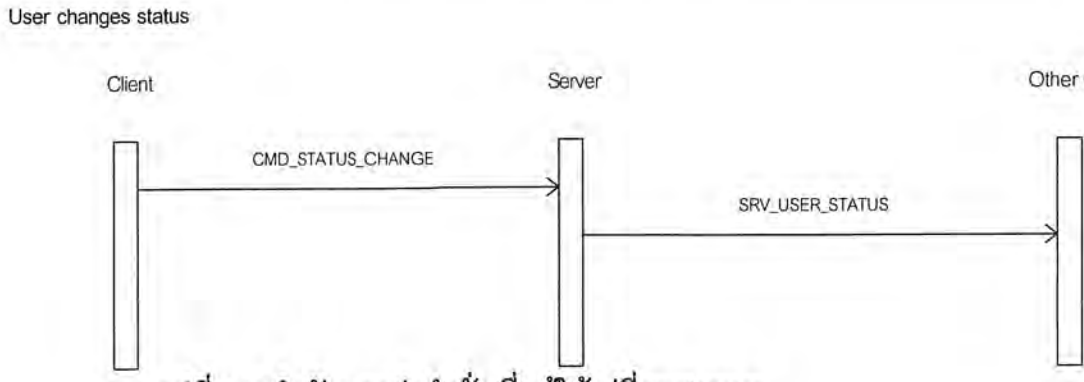
เมื่อผู้ใช้เริ่มเข้าสู่ระบบจะทำการ log in ซึ่งเป็นการทำงานในลักษณะ client/server โดยมีลำดับการส่งคำสั่ง ดังรูปที่ 3.1



รูปที่ 3.1 ลำดับการส่งคำสั่งในการ log-in

- User changes status

การเปลี่ยนสถานะของผู้ใช้จะเกิดขึ้นเมื่อมีผู้ใช้ทำการ sign in หรือ sign out ในการเปลี่ยนสถานะของผู้ใช้ระบบจะทำการแจ้งสถานะของผู้ใช้คนนั้นให้แก่ผู้ใช้คนอื่นๆทราบ การทำงานจะเป็นในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.2

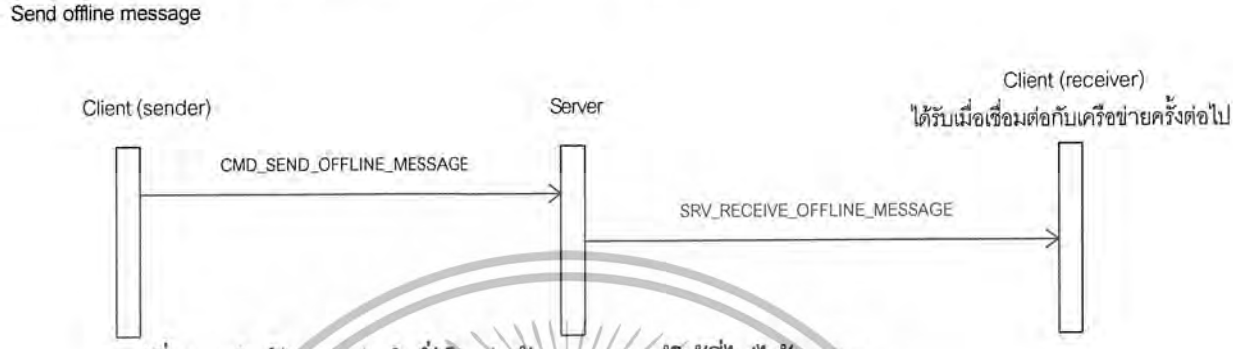


รูปที่ 3.2 ลำดับการส่งคำสั่งเมื่อผู้ใช้เปลี่ยนสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Send offline message

การส่ง offline message ผู้ใช้ที่เป็นผู้ส่งจะทำการส่งไปเก็บไว้ที่server จากนั้นเมื่อผู้ใช้ที่เป็นผู้รับข้อความ online server จะทำการส่งข้อความนั้นไปให้กับผู้ใช้คนนั้น การทำงานจะเป็นในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.3



รูปที่ 3.3 ลำดับการส่งคำสั่งในส่งข้อความหาผู้ใช้ที่ไม่ได้ online

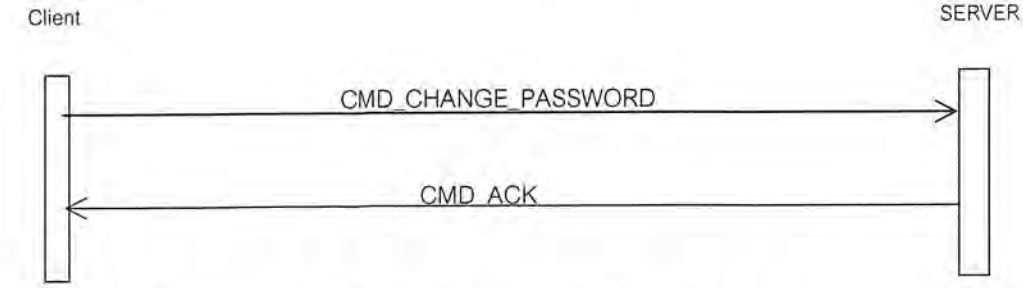
- User information request

การร้องขอการดูข้อมูลของผู้ใช้คนอื่นทำโดยผู้ใช้ร้องขอไปที่ server แล้วserver จะทำการส่งข้อมูลของผู้ใช้ที่ร้องขอไปกลับมา การทำงานจะเป็นในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.4

รูปที่ 3.4 ลำดับการส่งคำสั่งเมื่อผู้ใช้ขอดูข้อมูลผู้ใช้คนอื่น

- User changes password

ผู้ใช้ทำการเปลี่ยนรหัสผ่าน เป็นการทำงานในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.5

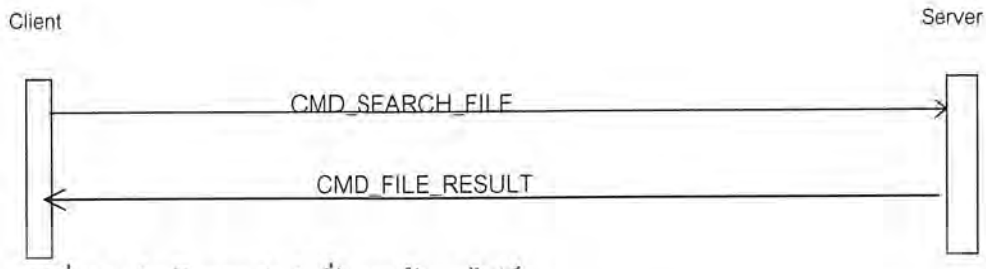


รูปที่3.5 ลำดับการส่งคำสั่งเมื่อผู้ใช้ร้องขอการเปลี่ยนรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- User searches file

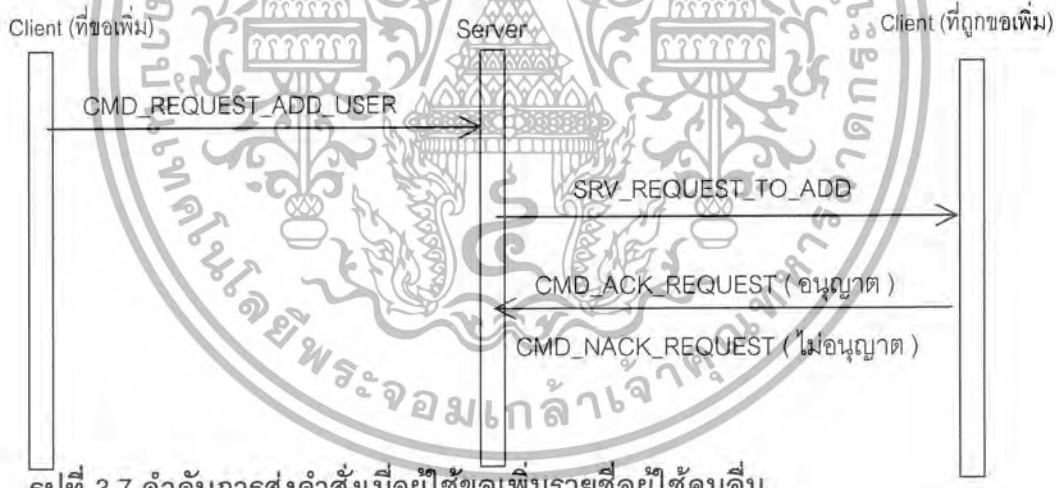
การทำการค้นหาไฟล์ข้อมูลว่าอยู่บนเครื่องใด เป็นการทำงานในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.6



รูปที่ 3.6 ลำดับการส่งคำสั่งการค้นหาไฟล์

- Request to add contact

การขอเพิ่มผู้ใช้คนอื่นๆเข้า contact list ผู้ใช้ทำการร้องขอผ่าน server ว่าต้องการเพิ่มรายชื่อผู้ใช้คนใดเข้า contact list จากนั้น server จะทำการส่งคำสั่งไปหาผู้ใช้คนที่ถูกเพิ่มว่าจะอนุญาตให้เพิ่มชื่อของเขาได้หรือไม่ การทำงานเป็นในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.7



รูปที่ 3.7 ลำดับการส่งคำสั่งเมื่อผู้ใช้ขอเพิ่มรายชื่อผู้ใช้คนอื่น

- User edits contact

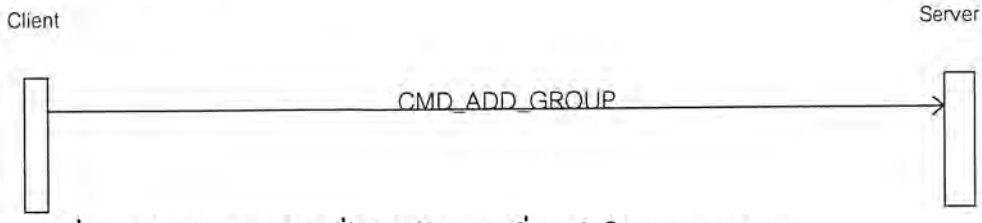
การจัดกลุ่มผู้ใช้ใน contact list เป็นการทำงานในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.8

รูปที่ 3.8 ลำดับการส่งคำสั่งสำหรับการจัดกลุ่มใน contact list

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- User adds contact list's group

การเพิ่มกลุ่มผู้ใช้ใน contact list เป็นการทำงานในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.9



รูปที่ 3.9 ลำดับการส่งคำสั่งสำหรับการเพิ่มกลุ่มใน contact list

- User removes contact list's group

การลบกลุ่มผู้ใช้ใน contact list เป็นการทำงานในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.10

รูปที่ 3.10 ลำดับการส่งคำสั่งสำหรับการลบกลุ่มใน contact list

- User removes contact list

การลบผู้ใช้ใน contact list เป็นการทำงานในลักษณะ client/server โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.11

รูปที่ 3.11 ลำดับการส่งคำสั่งสำหรับการลบผู้ใช้ออกจาก contact list

- Send message

การส่งข้อความหาผู้ใช้ที่กำลัง online อยู่ จะเป็นการส่งไปหาผู้ใช้คนนั้นโดยตรง ซึ่งเป็นการทำงานในลักษณะ peer to peer โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 ลำดับการส่งคำสั่งเมื่อผู้ใช้ส่งข้อความ online

- Download file

การ download ไฟล์ที่อยู่บนเครื่องของผู้ใช้คนอื่น เป็นการทำงานในลักษณะ peer to peer โดยที่การdownload ไฟล์ที่อยู่บนเครื่องของผู้ใช้คนใดจะต้องได้รับการอนุญาตจากผู้ใช้คนนั้นเสียก่อน ลักษณะการส่งคำสั่งเป็นดังรูปที่ 3.13

รูปที่ 3.13 ลำดับการส่งคำสั่งเมื่อผู้ใช้ทำการ download ไฟล์

- Chat

ในการสนทนาจะเริ่มต้นที่ผู้ใช้คนหนึ่งทำการเชิญผู้ใช้คนอื่นๆเข้าสู่ห้องสนทนาโดยผู้ใช้ที่มีสิทธิในการเพิ่มผู้ใช้คนอื่นเข้าในการสนทนาคือผู้ใช้ที่เป็นผู้สร้างการสนทนานั้นขึ้น ซึ่งผู้ใช้ที่เป็นผู้เริ่มการสนทนาจะมีลักษณะการทำงานเหมือนเป็น server ผู้ใช้คนอื่นๆจะทำการส่งข้อความมาที่เจ้าของการสนทนาแล้วผู้ใช้ที่เป็นผู้เริ่มต้นการสนทนาจะทำการส่งให้กับผู้ใช้คนอื่นๆที่เข้าร่วมการสนทนานั้น เช่นเดียวกันเมื่อมีผู้ใช้ออกจากการสนทนาผู้เริ่มการสนทนาก็จะทำการแจ้งให้กับผู้ใช้คนอื่นๆที่เข้าร่วมการสนทนาทราบ โดยลำดับการส่งคำสั่งเป็นดังรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.14 ลำดับการส่งคำสั่งเมื่อผู้ใช้ทำการสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 **ระบบฐานข้อมูลของระบบ** ประกอบด้วย

ตาราง UserData

attribute	Data type	คำอธิบาย	หมายเหตุ
UserName	char(255)	e-mail ของผู้ใช้	PK
Password	char(10)	รหัสผ่านของผู้ใช้	Not null
Name	char(20)	ชื่อผู้ใช้	
Surname	char(30)	นามสกุลผู้ใช้	
Address	char(50)	ที่อยู่ของผู้ใช้	
Tel	char(9)	เบอร์โทรศัพท์ของผู้ใช้	
StatusID	integer	สถานะของผู้ใช้	FK จาก Status (StatusID)
IP	long	หมายเลข IP ที่ใช้ในการติดต่อกัน	
Port	integer	หมายเลข Port ที่เปิดรอการเชื่อมต่อ	
Session	long	เลขที่ใช้อ้างอิงสำหรับการเชื่อมต่อ	

ตารางที่ 3.1 ตารางข้อมูลผู้ใช้

ตาราง Status

attribute	Data type	คำอธิบาย	หมายเหตุ
StatusID	integer	รหัสสถานะ	PK
StatusType	char(20)	คำอธิบายสถานะ	

ตารางที่ 3.2 ตารางสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง FileInfo

attribute	Data type	คำอธิบาย	หมายเหตุ
UserName	char(255)	e-mail เจ้าของไฟล์	PK, FK จาก UserData (UserName)
FileName	char(255)	ชื่อไฟล์	PK
FileSize	long	ขนาดไฟล์	

ตารางที่ 3.3 ตารางข้อมูลไฟล์ที่ผู้ใช้เปิดให้ผู้อื่น download ได้

ตาราง Contact

attribute	Data type	คำอธิบาย	หมายเหตุ
UserName	char(255)	e-mail ของผู้ใช้	PK, FK จาก UserData (UserName)
UserNameContact	char(255)	e-mail ของผู้ใช้คนอื่นๆ ที่ถูกร้องขอการติดต่อ	PK, FK จาก UserData (UserName)
GroupName	char(50)	กลุ่มของรายชื่อในlist	PK, FK จาก GroupContact (GroupName)
Allow	boolean	ผู้ใช้ที่ถูกร้องขอการติดต่ออนุญาตให้ติดต่อด้วยหรือไม่	
Reply	boolean	ผู้ใช้ที่ถูกร้องขอตอบกลับคำร้องขอแล้วหรือไม่	

ตารางที่ 3.4 ตารางรายชื่อผู้ใช้ที่ติดต่อกับผู้ใช้แต่ละคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง GroupContact

attribute	Data type	คำอธิบาย	หมายเหตุ
UserName	char(255)	e-mail ของผู้ใช้	PK, FK จาก UserData (UserName)
GroupName	char(50)	ชื่อกลุ่มของรายชื่อในlist	PK

ตารางที่ 3.5 ตารางกลุ่มผู้ใช้

ตาราง OfflineMessage

attribute	Data type	คำอธิบาย	หมายเหตุ
MessageID	long	ลำดับของข้อความ	PK
UserNameSend	char(255)	e-mail ผู้ส่งข้อความ	FK จาก UserData (UserName)
UserNameReceive	char(255)	e-mail ผู้รับข้อความ	FK จาก UserData (UserName)
Message	char(255)	ข้อความ	
DateTimeReceive	date	เวลาที่ได้รับข้อความ	
Send	boolean	ผู้รับได้รับข้อความแล้วหรือยัง	

ตารางที่ 3.6 ตารางข้อความที่ส่งถึงผู้ใช้ที่ offline

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การทำงานของโปรแกรม Instant Message for Oraganization (IMO)

การทำงานของโปรแกรมจะมีการทำงานสองลักษณะคือการทำงานแบบ client/server (ทำงานในลักษณะของผู้ร้องขอและผู้ให้บริการที่ชัดเจน) และการทำงานแบบ Peer to Peer (ทำงานในลักษณะการติดต่อกันโดยตรงระหว่างโหนดกับโหนดโดยที่แต่ละโหนดเป็นได้ทั้งผู้ให้บริการและผู้ขอรับบริการ) การทำงานทั้งสองลักษณะของโปรแกรมจะทำงานผ่านโพรโตคอล TCP/IP และในการอ้างอิงถึงผู้ใช้แต่ละคนจะใช้ e-mail addressในการอ้างอิง

การทำงานแบบ client/server

จะมีโปรแกรมส่วนที่เป็น server ทำหน้าที่เป็นตัวกลางในการที่จะทำให้ client ในเครือข่าย IMO สามารถติดต่อกันได้ เพราะการที่ client จะติดต่อกันบนโพรโตคอล TCP/IP นั้น จะต้องทราบ IP address ของอีกฝ่ายที่ต้องการจะติดต่อด้วย โดยขั้นตอนการทำงานจะมีลักษณะดังนี้

1. เริ่มต้น server จะเปิดให้บริการโดยรอรับการเชื่อมต่อจากผู้ใช้ (ใช้โปรแกรม IMO ฝั่ง client)
2. เมื่อผู้ใช้ต้องการเข้ามาใช้เครือข่าย ผู้ใช้จะต้องทำการเชื่อมต่อมาที่ server เพื่อทำการพิสูจน์ตัวบุคคลก่อน (Authenticate) โดยใช้ชื่อผู้ใช้และรหัสผ่านของแต่ละคน
3. ถ้าการพิสูจน์ตัวบุคคลถูกต้อง ผู้ใช้จะทำการส่งรายละเอียดเกี่ยวกับการเชื่อมต่อรวมถึง IP address ให้ server ทราบ server จะเก็บข้อมูลส่วนนี้ไว้ที่ฐานข้อมูลของ server
4. server จะทำการแจ้งสถานะการเชื่อมต่อและ IP address ของผู้ใช้ที่เพิ่งเข้ามาในเครือข่าย IMO ให้ผู้ใช้คนอื่นที่มีผู้ใช้คนนี้อยู่ใน Contact List และมีสถานะ Online ทราบ
5. เมื่อผู้ใช้ต้องการติดต่อกัน (ส่งข้อความ, chat หรือส่งไฟล์) จะติดต่อกันโดยตรงเลยโดยไม่ต้องผ่าน server

นอกจากนี้ยังมีการทำงานอื่นอีกที่ทำงานในลักษณะ client/server ดังนี้

1. ส่งข้อความในกรณีที่ผู้รับไม่ได้ทำการเชื่อมต่ออยู่กับเครือข่าย (Offline message) เป็นการส่งข้อความไปยังผู้ใช้ที่ไม่ได้ทำการเชื่อมต่อยู่กับเครือข่ายโดยจะส่งข้อความไปยัง server, server จะเก็บข้อความไว้ในฐานข้อมูลของ server และครั้งต่อไปที่ผู้รับทำการเชื่อมต่อเข้ามาในเครือข่าย server จะส่งข้อความนี้ให้
2. เมื่อผู้ใช้ทำการเปลี่ยนสถานะการเชื่อมต่อจะมีการแจ้งให้ server ทราบ เพื่อที่ server จะได้แจ้งให้ผู้ใช้คนอื่นที่มีผู้ใช้คนนี้อยู่ใน Contact List และมีสถานะ Online ทราบ โดยสถานะที่เป็นไปได้ทั้งหมดมีดังนี้

   - Offline คือ ผู้ใช้ไม่ได้เชื่อมต่ออยู่กับเครือข่าย IMO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Online คือ ผู้ใช้เชื่อมต่ออยู่กับเครือข่าย

3. การจัดการเกี่ยวกับการใช้ไฟล์ร่วมกัน โดย server จะมีหน้าที่ ดังนี้
   - หน้าที่ในการเก็บรายชื่อไฟล์ คือ เมื่อผู้ใช้ทำการเชื่อมต่อเข้ามาในเครือข่ายทุกๆ ครั้ง ผู้ใช้จะต้องทำการแจ้งรายชื่อไฟล์ทั้งหมดที่อนุญาตให้มีการใช้ร่วมกันได้ให้ server ทราบ server จะเก็บข้อมูลนี้ไว้ในฐานข้อมูลของ server
   - หน้าที่ในการทำการค้นหาไฟล์ คือ เมื่อผู้ใช้ต้องการค้นหาไฟล์ที่มีอยู่ทั้งหมดในเครือข่าย ผู้ใช้จะส่งชื่อไฟล์ที่ต้องการค้นหามาให้ server, server จะทำการหาไฟล์นั้นจากรายชื่อไฟล์ทีมีอยู่ในฐานข้อมูล แล้วแจ้งผลลัพธ์ให้ผู้ใช้ทราบถึงตำแหน่งที่ตั้งของไฟล์ว่าอยู่ที่ผู้ใช้คนใด
4. การเปลี่ยนรหัสผ่าน ในกรณีที่ผู้ใช้ต้องการเปลี่ยนรหัสผ่าน
5. การจัดการ Contact List โดย Contact List คือรายชื่อผู้ใช้อื่นที่ผู้ใช้จะสามารถติดต่อได้ ผู้ใช้แต่ละคนจะมี Contact List เป็นของตนเอง โดย Contact List ของผู้ใช้ทุกคนจะเก็บอยู่ที่ฐานข้อมูลของ server โดย server มีหน้าที่ ดังนี้
   - เมื่อผู้ใช้ทำการเชื่อมต่อเข้ามาในเครือข่ายทุกๆ ครั้ง server จะต้องแจ้ง Contact List ของผู้ใช้คนนั้น ให้ผู้ใช้ทราบ
   - จัดการการจัดการกลุ่มของรายชื่อของผู้ใช้ โดยผู้ใช้จะสามารถเพิ่ม ลด รายชื่อกลุ่ม ย้ายรายชื่อผู้ใช้ไปยังกลุ่มต่างได้
     หมายเหตุ: ผู้ใช้ไม่ทราบสร้างชื่อกลุ่มที่มีชื่อว่า Other ได้เพราะจะเป็นชื่อกลุ่มชั่วคราวที่โปรแกรมจะใช้ในการจัดการ Contact List
   - จัดการเพิ่ม ลด รายชื่อผู้ใช้คนอื่นใน Contact List โดยขั้นตอนการเพิ่มรายชื่อเข้ามาใน Contact List ของผู้ใช้แต่ละคนจะเป็น ดังนี้
     1. ผู้ใช้ทำการร้องขอการเพิ่มผู้ใช้คนอื่นเข้ามาใน Contact List ไปยัง server ต่อจากนั้นผู้ใช้ต้องทำการรอการอนุญาตจากผู้ใช้ที่ถูกเพิ่ม โดยผู้ใช้ที่อยู่ระหว่างการรอเพิ่มจะอยู่ในกลุ่มที่ชื่อว่า Other ใน Contact List ของผู้ใช้ชั่วคราว
     2. server ทำการแจ้งให้ผู้ใช้ที่ถูกร้องขอการเพิ่มรายชื่อเข้าไปใน Contact List ทราบเพื่อขอนุญาต ถ้าผู้ใช้คนนั้นไม่ได้เชื่อมต่ออยู่กับเครือข่าย server จะแจ้งให้ทราบเมื่อผู้ใช้คนนั้นเชื่อมต่อเข้ามาในเครือข่ายครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ถ้าผู้ใช้ที่ถูกร้องขอการเพิ่มรายชื่อเข้าไปใน Contact List อนุญาต server จะทำการปรับปรุง Contact List ให้ถูกต้อง แต่ถ้าไม่อนุญาต จะทำการลบรายชื่อออกจากกลุ่ม Other ที่ได้ทำการเพิ่มไว้ชั่วคราวไว้ก่อนหน้านี้

การทำงานแบบ Peer to Peer

เมื่อ client สามารถทราบ IP address ของ client อื่นได้จากการทำงานแบบ client/server แล้ว client ก็สามารถติดต่อกันโดยตรงโดยไม่ต้องผ่าน server ได้ โดยแต่ละ client สามารถเป็นได้ทั้งผู้ให้บริการและผู้ขอใช้บริการ โดยทุกๆ client จะเปิด port การเชื่อมต่อไว้เพื่อรอให้บริการ client เครื่องอื่นที่จะเข้าใช้บริการ โดยบริการมี 3 บริการ ดังนี้

1. การส่งข้อความถึงกัน ผู้ใช้ฝ่ายส่งข้อความจะทำการเลือกรายชื่อผู้ใช้ที่ต้องการติดต่อด้วยจาก Contact List ของผู้ใช้เอง โดยจะเลือกได้เฉพาะผู้ใช้ที่เชื่อมต่ออยู่กับเครือข่ายเท่านั้น จากนั้นจะส่งข้อความที่ต้องการไปให้ผู้ใช้ที่ได้ทำการเลือกไว้ โดยส่งไปตามข้อมูลการเชื่อมต่อและ IP address ของผู้ใช้คนนั้นที่ได้รับจาก server
2. การสนทนา (Chat) คือ การพูดคุยกันในห้องสนทนาของผู้ใช้ตั้งแต่ 2 คนขึ้นไป โดยมีขั้นตอนการทำงาน ดังนี้
   2.1 ผู้ใช้ที่ต้องการให้มีการสนทนาจะต้องทำการเปิดห้องสนทนาขึ้นมา แล้วเชิญผู้ใช้ที่เชื่อมต่ออยู่กับเครือข่ายจากใน Contact List ของผู้ใช้คนนั้น เข้ามาให้ห้องสนทนาอย่างน้อย 1 คน
   2.2 ผู้ใช้ที่ได้รับการเชิญเข้าร่วมสนทนาจะได้รับข้อความแจ้ง ถ้าผู้ใช้ตอบตกลงก็จะเริ่มการสนทนาได้ โดยขณะที่มีการสนทนาจะสามารถเพิ่ม, ลด หรือเลิกการสนทนาของผู้ใช้ในห้องสนทนาได้ โดยผู้ใช้ที่เปิดการสนทนาจะต้องเป็นผู้ดำเนินการและมีอำนาจเด็ดขาด
   2.3 เมื่อมีผู้ใช้ส่งข้อความมายังห้องสนทนา ข้อความจะถูกส่งมายังเครื่องของผู้ใช้เจ้าของการสนทนานี้ แล้วจากนั้นข้อความจะถูกส่งกระจายจากเครื่องของผู้ใช้เจ้าของการสนทนาไปยังผู้ใช้คนอื่นที่ร่วมการสนทนา ยกเว้นผู้ใช้ที่เป็นผู้ส่งข้อความ
   2.4 เมื่อมีการเพิ่ม หรือลดผู้ใช้เข้ามาในห้องสนทนา เครื่องที่เป็นเจ้าของการสนทนาจะแจ้งให้ผู้ใช้คนอื่นทราบ
3. การใช้ไฟล์ร่วมกัน ในทุกๆ client จะมีการตั้ง directory หนึ่งไว้ โดยให้ทุกไฟล์ที่อยู่ใน directory นี้จะอนุญาตให้ผู้อื่นสามารถทำสำเนานำไปใช้ได้ โดยมีขั้นตอนการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ผู้ใช้ที่ต้องการไฟล์ทำการร้องขอการอนุญาตในการคัดลอกไฟล์ไปที่ผู้ใช้ที่เป็นเจ้าของไฟล์นั้น (ข้อมูลนี้ได้มาจากการค้นหาไฟล์จากการทำงานแบบ client/server)

3.2 ถ้าผู้ใช้ที่เป็นเจ้าของไฟล์อนุญาตจะเริ่มทำการส่งข้อมูลของไฟล์นั้น เช่น ขนาดไฟล์ให้ผู้ใช้ผู้ร้องขอทราบ

3.3 เมื่อผู้ใช้ที่ร้องขอได้รับข้อมูลเกี่ยวกับรายละเอียดไฟล์ ต่อจากนั้นผู้ใช้เจ้าของไฟล์จะเริ่มส่งเนื้อหาไฟล์ไปให้ผู้ใช้ที่ร้องขอโดยส่งเป็นลักษณะของบล็อกข้อมูล ไปเรื่อยๆ จนกว่าจะครบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการดำเนินงาน

จากการดำเนินการศึกษาปัญหาพิเศษเรื่อง Instant Messaging for Organization และจากการพัฒนาโปรแกรมขึ้นพบว่าระบบมีความต้องการขั้นพื้นฐานของคอมพิวเตอร์และวิธีการดำเนินงาน ดังต่อไปนี้

4.1 ความต้องการขั้นพื้นฐานของคอมพิวเตอร์

4.1.1 ความต้องการทางด้านฮาร์ดแวร์

- คอมพิวเตอร์รุ่น Pentium 100 ขึ้นไป
- หน่วยความจำขนาด 32 MB ขึ้นไป
- ฮาร์ดดิสก์ 10 MB ขึ้นไป
- สามารถเชื่อมต่อกับเครือข่าย TCP/IP ได้

4.1.2 ความต้องการทางด้านซอฟต์แวร์

- ระบบปฏิบัติการ Window 95 , 98 , 2000 , XP

4.2 ขั้นตอนการติดตั้งโปรแกรม

1. ลงโปรแกรม server กับเครื่องที่เป็น server
2. ลงโปรแกรม client กับเครื่องในเครือข่าย
3. ทำการรันโปรแกรม

4.3 การทำงานของโปรแกรม Instant Messaging for Organization

เมื่อทำการรันโปรแกรมแล้วจะแสดงหน้าจอดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

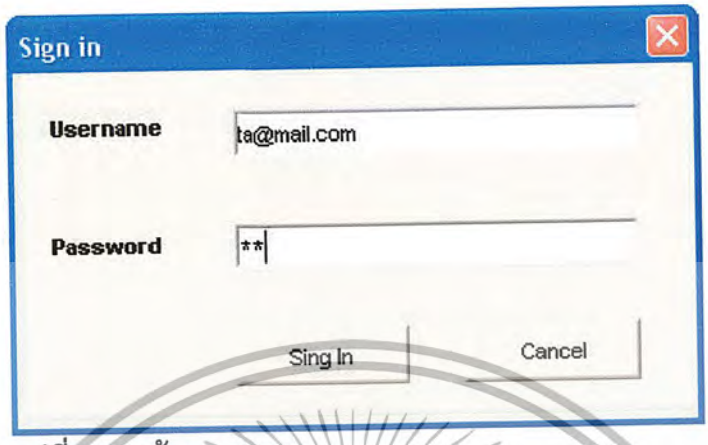
รูปที่ 4.1 **หน้าจอแรกของโปรแกรม** Instant Messaging for Organization

ผู้ใช้ทำการ sign in เข้าสู่ระบบโดยเลือก File > Sign In ดังรูปที่ 4.2

รูปที่ 4.2 **หน้าจอแสดงการเลือกเมนู** sign in

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเลือก Sign in จะปรากฏหน้าจอให้ผู้ใช้กรอก username และรหัสผ่านของตนเพื่อทำการ sign in ดังรูปที่ 4.3

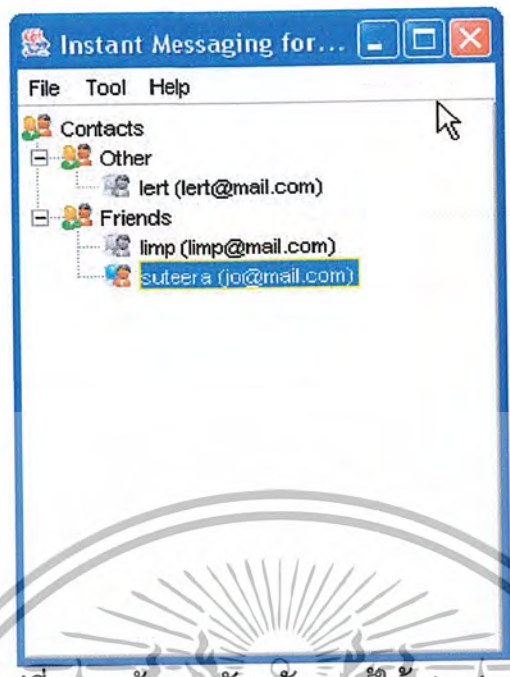


รูปที่ 4.3 หน้าจอ sign in

เมื่อกดปุ่ม Sign In จะมีการพิสูจน์ตัวบุคคลก่อน ถ้าไม่ถูกต้องจะแสดงหน้าจอเตือนให้ตรวจสอบรหัสผ่านอีกครั้ง ดังรูปที่ 4.4 และถ้าการพิสูจน์ตัวบุคคลถูกต้องจะแสดง Contact List ที่หน้าจอหลัก ดังรูปที่ 4.5

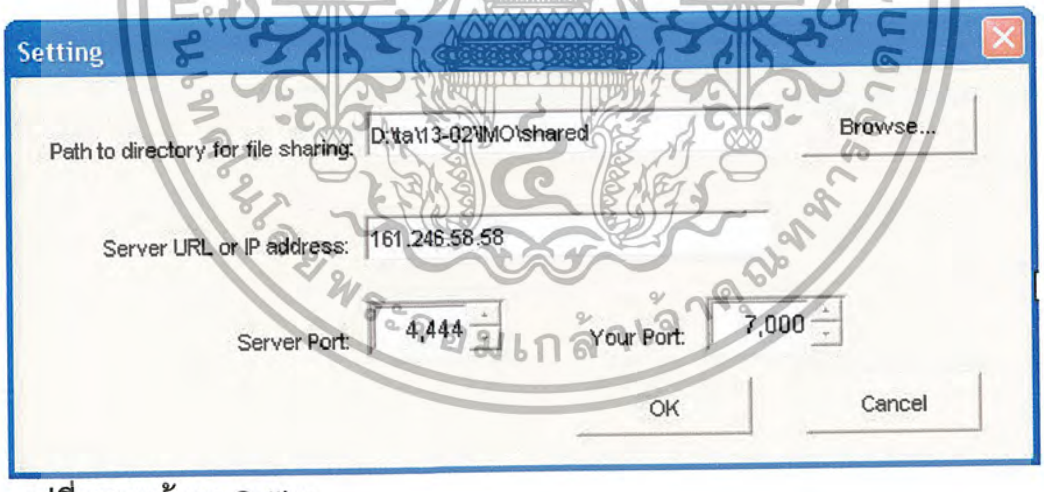
รูปที่ 4.4 หน้าจอแสดงการพิสูจน์ตัวบุคคลไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 หน้าจอหลักหลังจากผู้ใช้ sign in เข้าสู่ระบบ

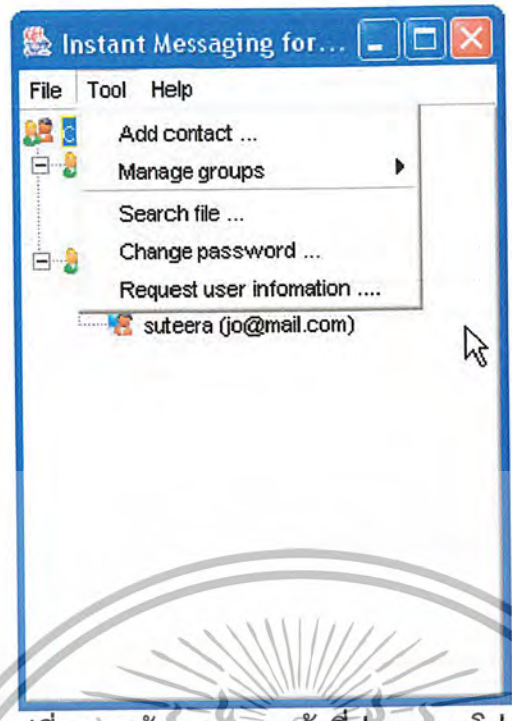
ผู้ใช้จะทำการกำหนดรายละเอียดการเชื่อมต่อ (IP Address และ Port) และไฟล์ที่อนุญาตให้มีการ share ได้ โดยการเลือก File > My setting จะแสดงหน้าจอดังนี้



รูปที่ 4.6 **หน้าจอ** Setting

เมื่อเลือก Tool จะแสดงหน้าที่ต่างๆที่โปรแกรมทำงานได้ ดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



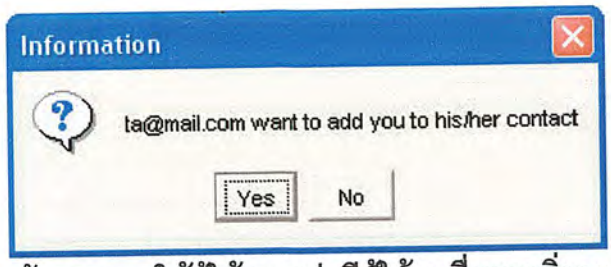
รูปที่ 4.7 หน้าจอแสดงหน้าที่ต่าง ๆของโปรแกรม

ผู้ใช้สามารถเพิ่มรายชื่อผู้ใช้คนอื่นใน Contact List โดยทำการเลือก Tool > Add contact จากนั้นจะแสดงหน้าจอให้ผู้ใช้กรอก username ที่ต้องการเพิ่ม ดังรูปที่ 4.8

รูปที่ 4.8 หน้าจอการเพิ่มรายชื่อผู้ใช้คนอื่นเข้าใน Contact List

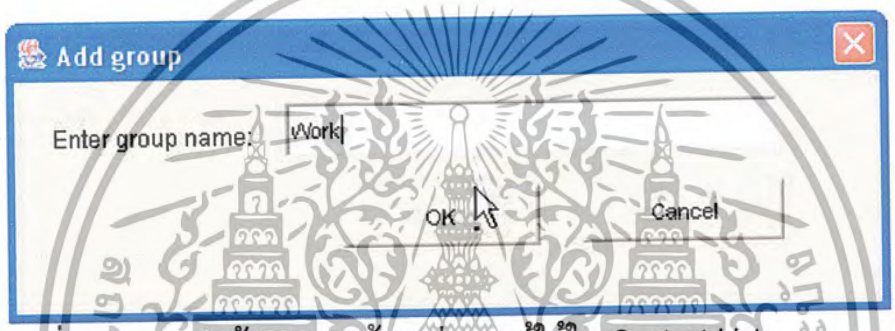
เมื่อทำการขอเพิ่มรายชื่อผู้ใช้คนอื่นเข้า Contact List แล้ว ที่เครื่องของผู้ใช้คนที่ถูกเพิ่มจะมีข้อความแจ้งให้ผู้ใช้คนนั้นทราบว่ามีผู้ใช้คนอื่นขอเพิ่มรายชื่อของเขาดังรูปที่ 4.9 ซึ่งผู้ใช้ที่ถูกเพิ่มจะทำการอนุญาตให้เพิ่มหรือไม่ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

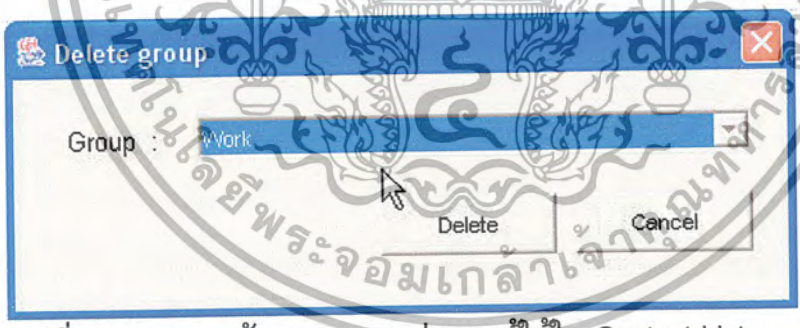


รูปที่ 4.9 หน้าจอแสดงให้ผู้ใช้ทราบว่ามีผู้ใช้คนอื่นขอเพิ่มรายชื่อเขาเข้า Contact List

ผู้ใช้สามารถแบ่งกลุ่มผู้ใช้ใน Contact List ของตนเป็นกลุ่มต่างๆได้ โดยทำการสร้างกลุ่มได้โดยเลือก Tool > Manage groups > Add group จะขึ้นหน้าจอดังรูปที่ 4.10 และการลบกลุ่มได้โดยเลือก Tool > Manage groups > Add group จะขึ้นหน้าจอดังรูปที่ 4.11



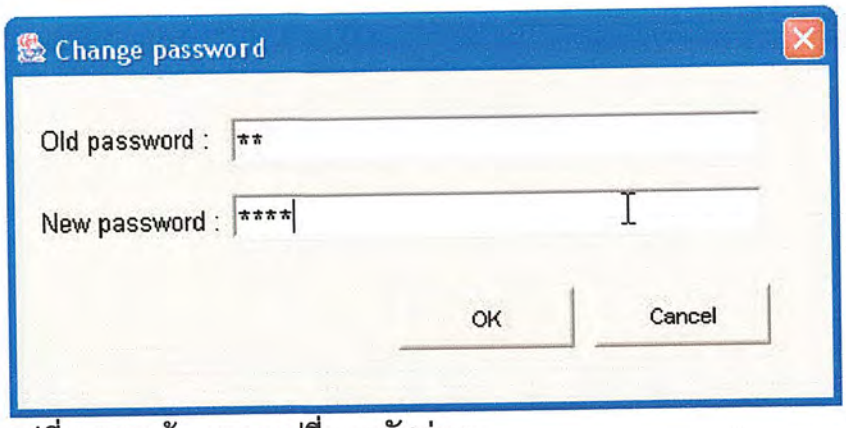
รูปที่ 4.10 แสดงหน้าจอการสร้างกลุ่มของผู้ใช้ใน Contact List



รูปที่ 4.11 แสดงหน้าจอการลบกลุ่มของผู้ใช้ใน Contact List

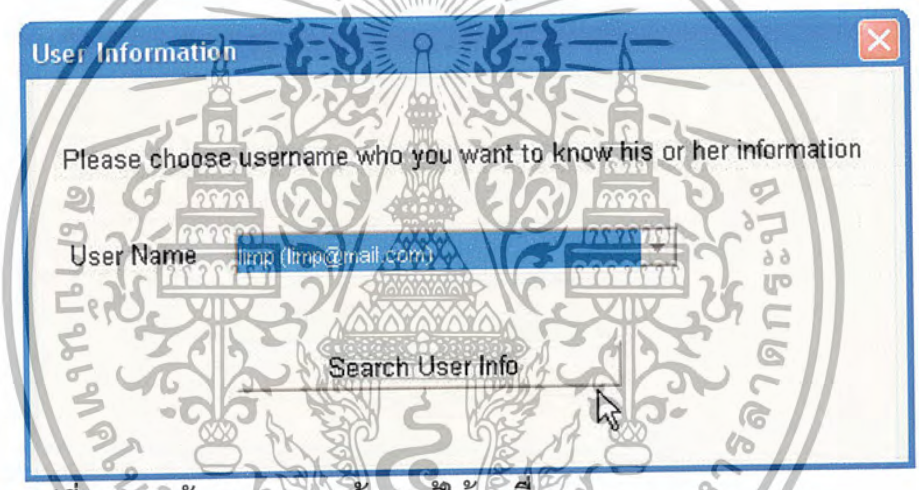
การเปลี่ยนรหัสผ่านของผู้ใช้ทำได้โดยเลือก Tool > Change password จะแสดงหน้าจอให้ใส่รหัสผ่านเดิมและรหัสผ่านใหม่ดังรูปที่ 4.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 หน้าจอการเปลี่ยนรหัสผ่าน

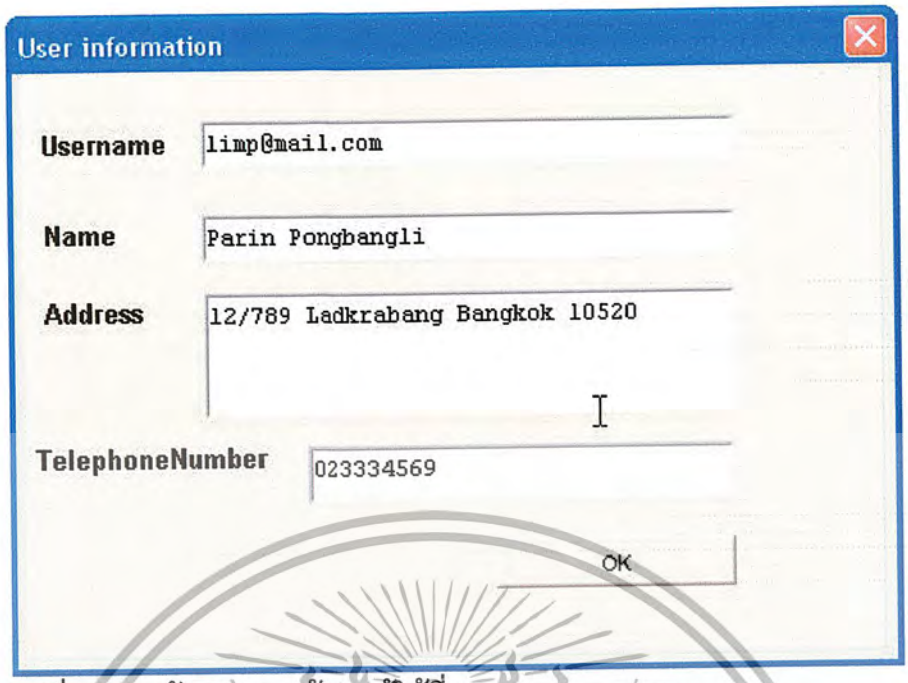
ผู้ใช้สามารถขอดูข้อมูลของผู้ใช้คนอื่นที่อยู่ใน Contact List ได้โดยเลือก Tool > Request user information เมื่อเลือกแล้วจะแสดงหน้าจอให้เลือกผู้ใช้ที่ต้องการขอดูข้อมูล ดังรูปที่ 4.13



รูปที่ 4.13 หน้าจอการขอดูข้อมูลผู้ใช้คนอื่น

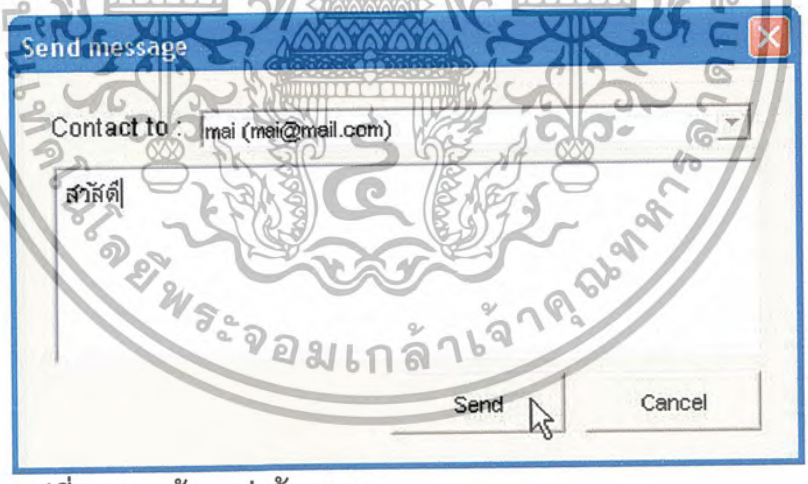
เมื่อกด Search User Info จะแสดงข้อมูลผู้ใช้ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 หน้าจอแสดงข้อมูลผู้ใช้ที่ขอดู

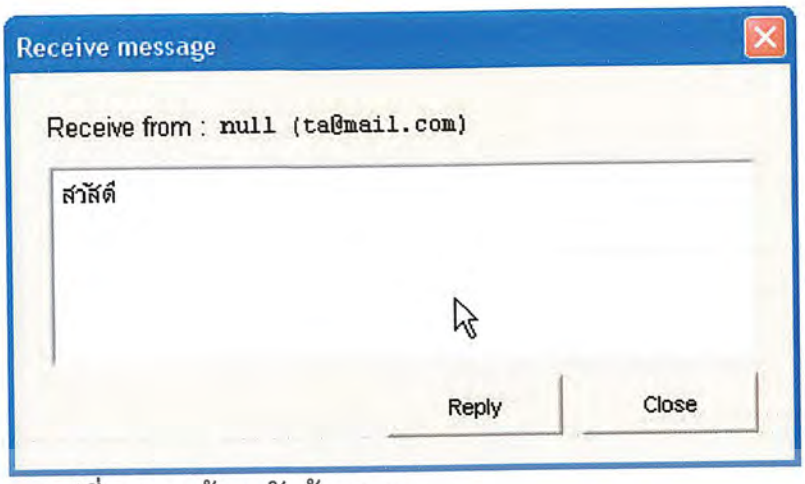
การสนทนาระหว่างผู้ใช้ 2 คน ทำโดยการกดเลือกที่ผู้ใช้คนที่ต้องการสนทนาด้วยแล้วจะแสดงหน้าจอสำหรับการส่งข้อความ ดังรูปที่ 4.15



รูปที่ 4.15 หน้าจอส่งข้อความ

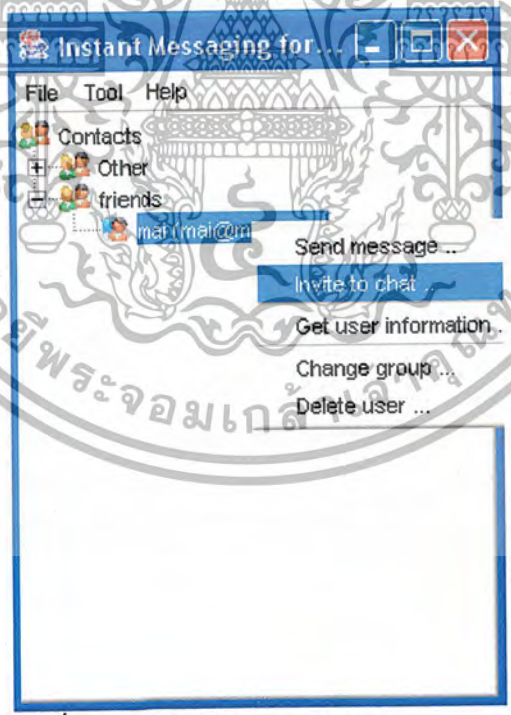
ผู้ใช้ทำการเลือกผู้ใช้อื่นที่ต้องการสนทนาด้วย และพิมพ์ข้อความที่ต้องการส่งจากนั้นกดส่ง เมื่อทำการส่งข้อความไปแล้วที่เครื่องผู้รับจะมีการแสดงหน้าจอแสดงข้อความที่ได้รับ ดังรูปที่ 4.16 และถ้าต้องการส่งข้อความต่อให้ทำการกด Reply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



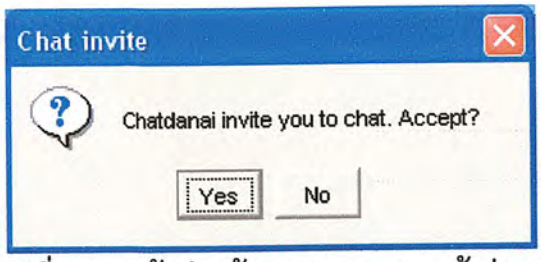
รูปที่ 4.16 หน้าจอรับข้อความ

การเข้าสู่เมนูการสนทนากับกลุ่มบุคคลทำได้จะต้องมีผู้ใช้ที่เป็นผู้เริ่มต้นการสนทนา ผู้ใช้ที่เป็นผู้เริ่มการสนทนาจะทำการเชิญผู้ใช้คนอื่นโดย เลือกที่ชื่อของผู้ใช้ที่ต้องการจะสนทนาด้วย จากนั้นคลิกขวาเลือก Invite to chat.. ดังรูปที่ 4.17 และที่หน้าจอของผู้ใช้ที่ถูกเชิญเข้าร่วมการสนทนา จะมีข้อความขึ้นถามว่าต้องการเข้าร่วมการสนทนาหรือไม่ ดังรูปที่ 4.18



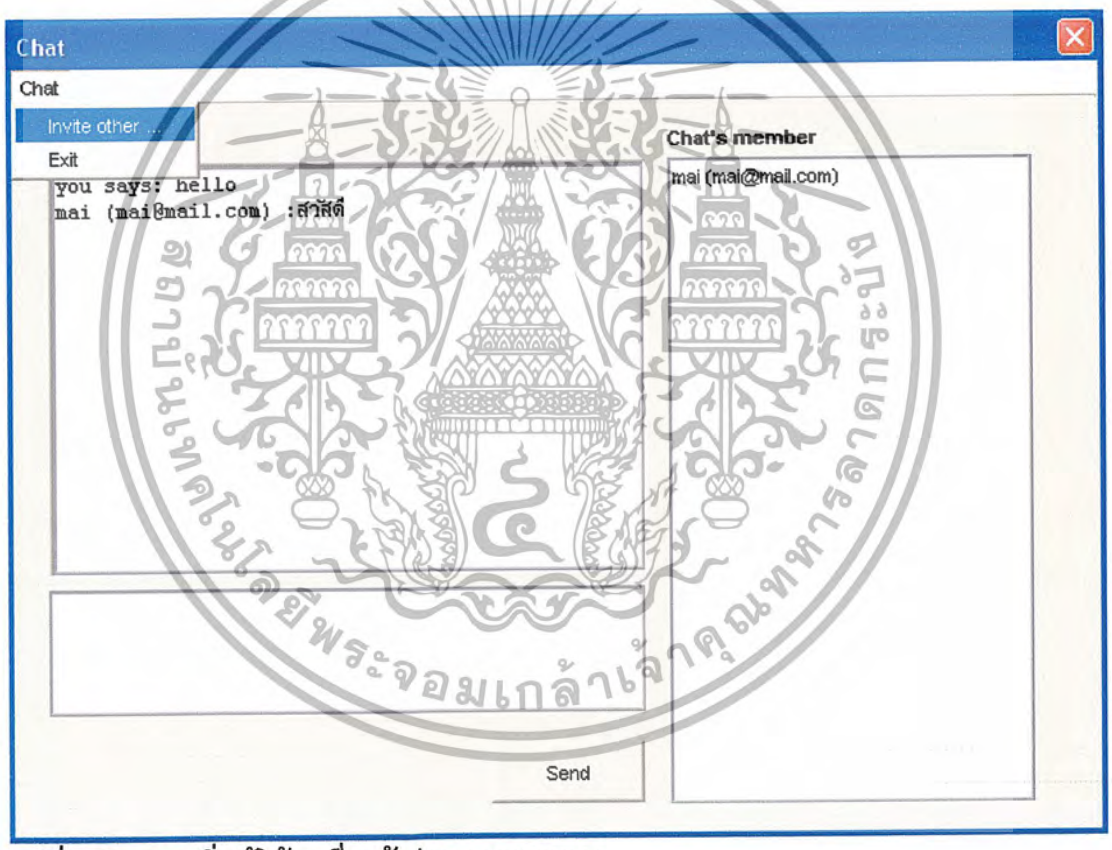
รูปที่ 4.17 การเลือกเมนูสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



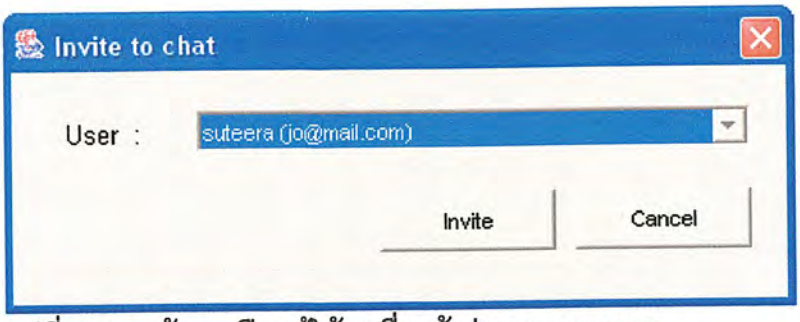
รูปที่ 4.18 หน้าต่างข้อความถามการเข้าร่วมสนทนา

การเพิ่มผู้ใช้คนอื่นเข้าร่วมการสนทนาเป็นสิทธิ์ของผู้ใช้ที่เป็นผู้เริ่มการสนทนานั้นเพียงผู้เดียว โดยการเลือกที่เมนู Chat > Invite other.. ดังรูป 4.19 จากนั้นจะขึ้นหน้าจอชื่อผู้ใช้ที่อยู่ใน list และ online อยู่ให้เลือกเชิญ ดังรูป 4.20



รูปที่ 4.19 การเพิ่มผู้ใช้คนอื่นเข้าร่วมการสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 หน้าจอเลือกผู้ใช้คนอื่นเข้าร่วมการสนทนา

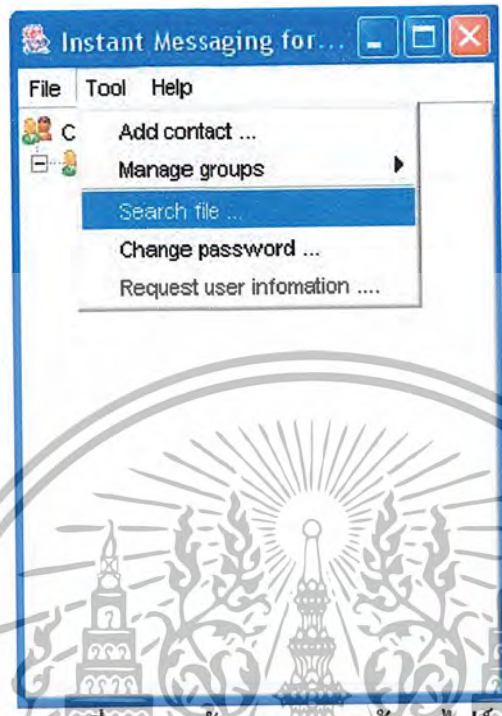
รูปที่ 4.21 แสดงหน้าจอการสนทนาโดยที่ช่อง Chat's member จะขึ้นรายชื่อของผู้ใช้ทุกคนที่ร่วมสนทนาอยู่ ช่อง Chat จะแสดงข้อความที่สนทนาทั้งหมด ผู้ที่ต้องการสนทนาจะทำการพิมพ์ข้อความที่ต้องการสนทนาลงที่ช่องด้านล่าง จากนั้น กด Send ข้อความนั้นจะไปแสดงที่ช่อง Chat ด้านบน การจบการสนทนาทำโดยการเลือก Chat > Exit

รูปที่ 4.21 แสดงหน้าจอการสนทนา

การ download ไฟล์ที่อยู่บนเครื่องของผู้ใช้คนอื่นทำได้โดยการเลือกเมนูการค้นหาไฟล์ที่ต้องการจะdownload ก่อนว่าอยู่ที่เครื่องของผู้ใช้คนใด โดยเลือก Tool > Search file... ดังรูปที่

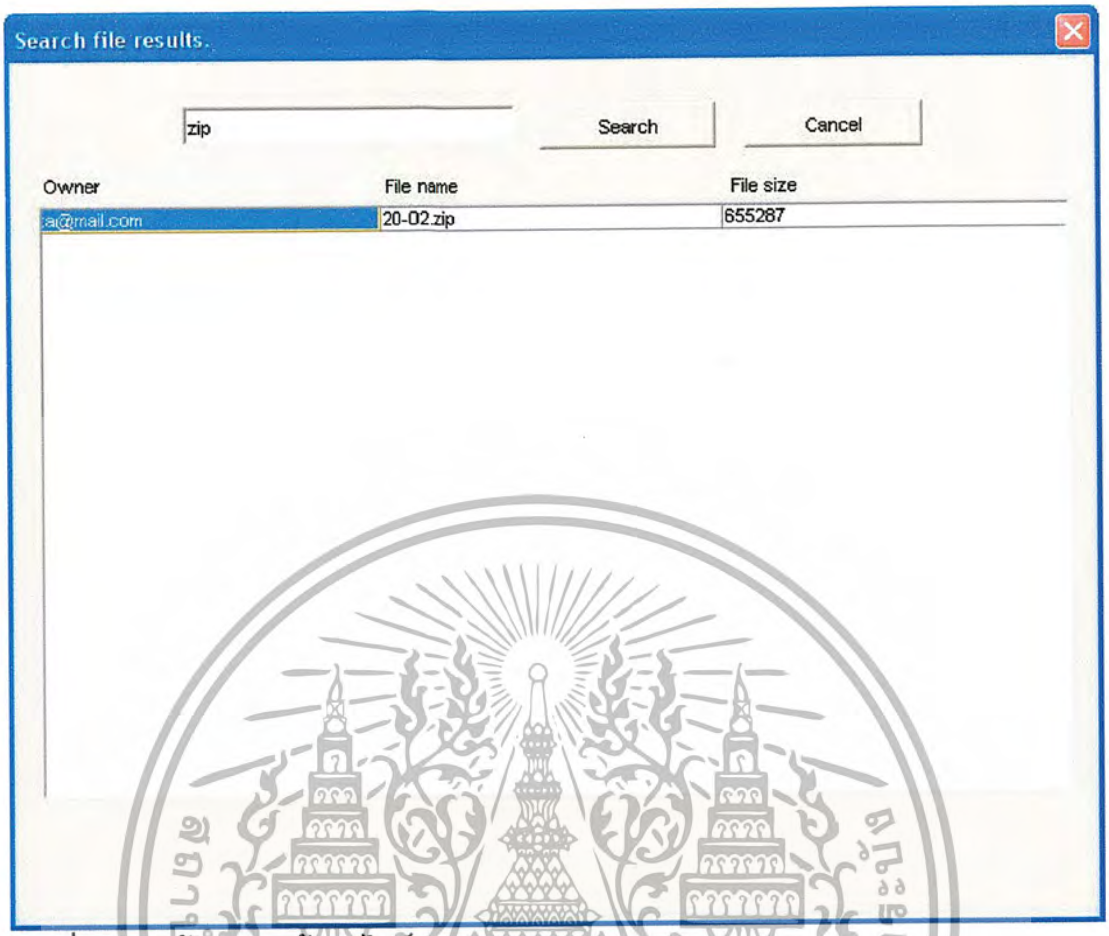
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.22 ในการค้นหาไฟล์ ให้ใส่ชื่อไฟล์หรือประเภทไฟล์ที่ต้องการค้นหาลงในช่องด้านบนแล้วทำการกด Search จะขึ้นไฟล์ทั้งหมดที่ค้นหาเจอ เครื่องเจ้าของไฟล์ และขนาดของไฟล์ ดังรูปที่ 4.24



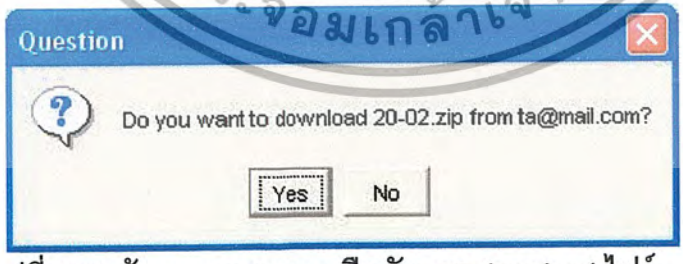
รูปที่ 4.22 หน้าจอเมนูการค้นหาไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 **หน้าจอการค้นหาไฟล์**

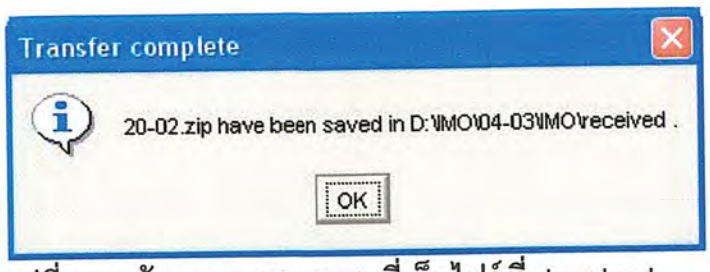
ถ้าต้องการจะ download ให้ทำการคลิกที่ไฟล์ที่ต้องการจะขึ้นข้อความถามยืนยันการ download ไฟล์อีกครั้ง ดังรูปที่ 4.24



รูปที่ 4.24 **ข้อความแสดงการยืนยันการ** download **ไฟล์**

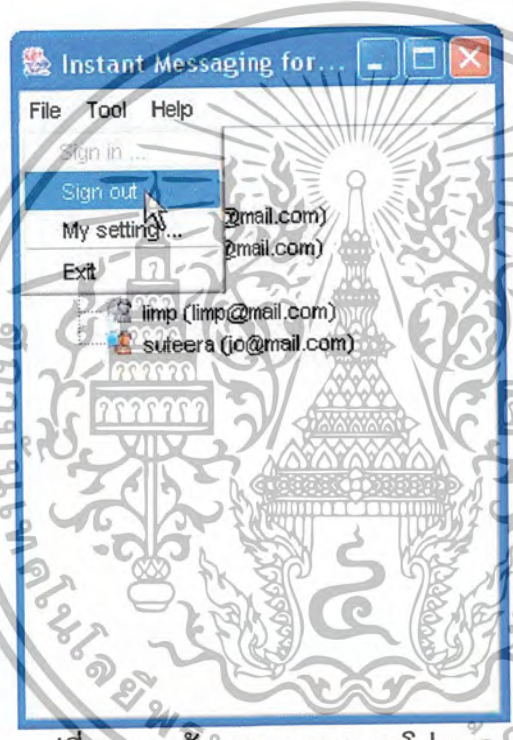
เมื่อทำการกด Yes จะขึ้นข้อความแสดงให้ผู้ใช้ทราบว่าไฟล์ที่ download จะถูกเก็บอยู่ที่ path ใด ดังรูปที่4.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 ข้อความแสดง path ที่เก็บไฟล์ที่ download

การออกจากโปรแกรมทำได้โดยการเลือก File > Sign out หรือ การเลือก File > Exit ดังนี้



รูปที่ 4.26 หน้าจอการออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การทำปัญหาพิเศษเรื่อง Instant Messaging for Organization มีวัตถุประสงค์เพื่อศึกษาและทำความเข้าใจ พื้นฐานการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ภายในเครือข่าย และเครื่องมือหรือโปรแกรมที่ใช้ในการพัฒนา เพื่อนำความรู้มาสร้างเป็นโปรแกรมที่สามารถใช้งานได้จริง และคาดหวังว่าจะสามารถนำไปประยุกต์ใช้ได้ในอนาคต

โปรแกรม Instant Messaging for Organization ที่ทำการพัฒนาขึ้นนี้สามารถใช้ในการติดต่อสื่อสารภายในองค์กร รวมทั้งการแลกเปลี่ยนไฟล์ข้อมูลระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง ความสามารถของโปรแกรมมีดังนี้

- ส่งข้อความถึงบุคคลโดยที่บุคคลนั้นจะ online หรือ offline อยู่ก็ได้
- สนทนา (chat)
- แลกเปลี่ยนไฟล์ข้อมูลระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง
- ดูข้อมูลของผู้ใช้คนอื่น
- เพิ่มและลบรายชื่อผู้ใช้
- สร้างและลบกลุ่มรายชื่อผู้ใช้
- ผู้ใช้แต่ละคนสามารถเปลี่ยนรหัสผ่านได้

ซอฟต์แวร์ที่ใช้ในการพัฒนาโปรแกรม Instant Messaging for Organization (IMO) มีดังนี้

1. JBuilder Personal 9.0
2. ฐานข้อมูลที่รองรับการเชื่อมต่อ ODBC

5.2 ข้อจำกัดของโปรแกรม

- เรื่องของการเชื่อมต่อ เนื่องจากการทำงานของโปรแกรมมีการทำงานในลักษณะของ Peer to Peer บางสถานการณ์ต้องการกลไกที่จะระบุตำแหน่งของโหนดที่ไม่ได้ใช้ Public IP Address และต้องการการติดต่อผ่าน firewall ซึ่งทั้ง NATs และ firewall ต่างก็มีหลายประเภท ปัญหานี้ไม่สามารถแก้ไขได้ทุกกรณี ซึ่งในขณะนี้โปรแกรมยังไม่รองรับการเชื่อมต่อผ่าน NATs และ firewall

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เรื่องของความปลอดภัยและความเป็นส่วนตัว มีความเสี่ยงเพิ่มขึ้นในเรื่องของความปลอดภัยจากการใช้งานเครือข่าย เนื่องจากการทำงานแบบ Peer to Peer มีการเข้าถึงทรัพยากรต่างๆ ร่วมกัน
- การเชื่อมต่อที่ไม่ต่อเนื่อง เนื่องจากแต่ละโหนดต่างสามารถเข้าหรือออกเครือข่ายได้อย่างอิสระ ทำให้ไม่สามารถรับรองได้ว่าในขณะเวลาหนึ่งจะสามารถสื่อสารกับโหนดที่ต้องการได้
- ผู้ใช้งาน แต่ละโหนดจะมีบริการให้กับเครื่องคนอื่นซึ่งทำงานในลักษณะเครื่องแม่ข่าย แต่ผู้ใช้ไม่มีประสบการณ์ในการจัดการเครื่องแม่ข่าย ทำให้อาจเป็นปัญหาในการใช้งานได้

5.3 ข้อเสนอแนะ

โปรแกรมควรมีการพัฒนาต่อเพื่อให้สามารถรองรับการทำงานในหัวข้อต่อไปนี้

1. เรื่องระยะเวลาการตอบอนุญาตให้ download ไฟล์ข้อมูล คือโปรแกรมไม่ได้กำหนดระยะเวลาว่าเจ้าของไฟล์ต้องตอบกลับภายในเวลาเท่าใด เป็นผลให้ผู้ใช้ที่เป็นผู้ขอ download ไฟล์จะต้องรอคอยการตอบกลับไปเรื่อยๆ
2. ในการ download ไฟล์ ไฟล์ที่ download มานั้นจะถูกเก็บไว้ใน directory หนึ่งที่ระบบกำหนดไว้ ซึ่งผู้ใช้ไม่สามารถกำหนดเองได้ว่าจะให้เก็บไว้ที่ใด
3. ในการส่งข้อความ ควรที่จะมีการเก็บข้อความที่ถูกส่งไปแล้ว เพื่อที่จะได้รู้ว่าข้อความนี้ได้ถูกส่งไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

วีระศักดิ์ ซึงถาวร. 2545. Java Programming Volume 1. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

วีระศักดิ์ ซึงถาวร. 2545. Java Programming Volume 2. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

Philip M. Lewis et. al. 2002. Database and Transaction Processing. New York : Caroline Fell.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้