

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

โปรแกรมสนทนาด้วยเสียงผ่านเครือข่าย

IP SOFTPHONE

นายสุเมธ เอียดเกลี้ยง  
นายสุรียัน ศรีมาตย์

เลขหมู่.....  
เลขทะเบียน.....62375  
วัน,เดือน,ปี...1.6. ส.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสนทนาด้วยเสียงผ่านเครือข่าย  
IP SOFTPHONE

โดย

นายสุเมธ เอียดเกลี้ยง

นายสุริยัน ศรีมาตย์

อาจารย์ที่ปรึกษา

พศ. ธนา หงษ์สุวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมสนทนาด้วยเสียงผ่านเครือข่าย

IP SOFTPHONE

ผู้จัดทำ

1. นายสุเมธ เอียดเกลี้ยง รหัสนักศึกษา 46015384

2. นายสุริยัน ศรีมาตย์ รหัสนักศึกษา 46015387



อาจารย์ที่ปรึกษา

(ผศ.ธนา หงษ์สุวรรณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โปรแกรมสนทนาด้วยเสียงผ่านเครือข่าย

สุเมธ ใต้ดเกลี้ยง	46015384
สุวิทย์ ศรีมาตย์	46015387
ผศ. ธนา หงษ์สุวรรณ	อาจารย์ที่ปรึกษา
ปีการศึกษา 2548	

## บทคัดย่อ

เมื่ออินเทอร์เน็ตมีการใช้งานอย่างกว้างขวางขึ้นความต้องการประยุกต์ใช้สิ่งใหม่ๆ บนอินเทอร์เน็ตจึงได้รับการพัฒนา โดยเฉพาะอย่างยิ่งการใช้อินเทอร์เน็ตเป็นโครงสร้างพื้นฐานเพื่อรองรับการสื่อสารรูปแบบต่างๆ เช่น การติดต่อด้วยเสียง การกระจายสัญญาณเสียงหรือภาพบนเครือข่าย

การติดต่อสื่อสารแบบเรียลไทม์กลายเป็นส่วนสำคัญของธุรกิจในปัจจุบัน เพื่อลดค่าโทรศัพท์และค่าโทรสารและยังเป็นการสร้างฐานสำหรับการใช้มัลติมีเดียในระดับที่สูงขึ้นเพื่อควมมีประสิทธิภาพในการสื่อสารภายในองค์กร เช่น การประชุมแบบ คอนเฟอร์เรนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# IP SOFTPHONE

Sumate Ieadkleang 46015384

Suriyan Srimat 46015387

Asst.Prof. Thana Hongsuwan Advisor

Academic Year 2004

## ABSTRACT

Now the internet has more using and more interest to make new application to use with it, Especial use internet for basic communication and using many ways. for example voice communication, audio/voice in network communication

Currently real time communication is important part of business for reduce expenses of telephone and fax, it make base for using in higher multimedia for efficiency of communication in organize. For example vedio multimedia conference

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก ผศ ธนา หงษ์สุวรรณ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ที่ให้คำปรึกษาและการช่วยเหลือข้าพเจ้าขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า ขอขอบคุณเพื่อนๆ และรุ่นพี่ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบอบแด่ผู้มีพระคุณทุกท่าน

สุเมธ เอียดเกลี้ยง  
สุรียัน ศรีมาตย์

# สารบัญ

	หน้า	
บทคัดย่อภาษาไทย.....	I	
บทคัดย่อภาษาอังกฤษ.....	II	
กิตติกรรมประกาศ.....	III	
สารบัญ.....	IV	
สารบัญตาราง.....	VI	
สารบัญรูป.....	VII	
บทที่ 1 บทนำ		
1.1 ความสำคัญและที่มา.....	1	
1.2 วัตถุประสงค์ของโครงการ.....	2	
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2	
1.4 ขอบเขตของโครงการ.....	2	
บทที่ 2 พื้นฐาน โปรโตคอลอวอยซ์โอเวอร์ไอพี : VoIP.....		3
2.1 หลักการพื้นฐานของเครือข่ายไอพี.....	3	
2.2 ขั้นตอนการทำงานของ VoIP.....	5	
2.2.1 ขั้นตอนการทำงานเบื้องต้น.....	5	
2.2.2 กระบวนการทำงานของเทคโนโลยี VoIP มีรายละเอียดการทำงานดังนี้.....	6	
2.3 Standard of VoIP Technology.....	8	
2.3.1 SIP: Session Initial Protocol.....	10	
2.3.2 Real Time Protocol.....	19	
บทที่ 3 Enhancing Rich Client Communications with the Microsoft* Real-Time Communications API.....		21
3.1 Introduction.....	21	
3.1.1 RTC Client Initialization & Platform Capabilities.....	22	
3.2 Get Ready.....	24	
3.3 Session Initiation Protocol Server.....	24	
3.3.1 การเริ่มต้นสำหรับ Presence และ Buddy List Notifications.....	25	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

3.3.2 Handling RTC Events.....	25
3.3.3 Enabling and Detecting Presence .....	27
3.3.4 SIP Server Registration .....	28
3.3.5 การสร้าง XML Profile .....	38
3.3.10 Creating the Buddy List และ Watcher Object .....	39
3.3.11 Deregistering และ Disabling a Profile .....	41
3.3.12 Optimizing for Performance .....	41
3.3.13 Conclusion.....	42
<b>บทที่ 4 การออกแบบโครงงานและพัฒนาโครงงาน.....</b>	<b>43</b>
4.1 กำหนด requirement .....	43
4.1.1 function requirement.....	43
4.1.2.nonfunction requirement .....	43
4.1.3 Use Case Diagram.....	43
4.2 การวิเคราะห์.....	44
4.2.1 วิเคราะห์ระบบในส่วนที่เป็น se Case Diagram แบบละเอียด.....	44
4.2.2 วิเคราะห์ Use Case.....	45
4.2.3 sequence diagram .....	46
4.2.4 วางแผนการสร้าง .....	49
4.3 การออกแบบ.....	50
4.3.1 User Interface.....	50
4.4 การทดลองและผลการทดลอง.....	56
<b>บทที่ 5 บทวิจารณ์และสรุป</b>	
5.1. บทสรุป .....	59
5.2. วิจารณ์สิ่งที่ได้จากโครงงาน .....	59
5.3. ปัญหาอุปสรรคและแนวทางแก้ไข .....	59
5.4. แนวทางการพัฒนาต่อ .....	59
<b>บรรณานุกรม .....</b>	<b>60</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

	หน้า
ตารางที่ 1 แสดง Payload Type Number กับ Video Formate .....	20
ตารางที่ 2 audio codec ที่รองรับ โดย Windows RTC client .....	23
ตารางที่ 3 แสดงรายละเอียดของ use case .....	44
ตารางที่ 4 การวางแผนดำเนินงาน .....	49



# สารบัญรูป

หน้า

รูปที่ 2.1.1	ภาพการส่ง แพ็กเก็ตระหว่างเครือข่าย.....	3
รูปที่ 2.1.2	ภาพการเชื่อมต่อเครือข่ายโทรศัพท์ผ่านอินเทอร์เน็ต โดยนำ VoIP มาใช้ .....	4
รูปที่ 2.1.3	ภาพการเชื่อมต่อระบบโทรศัพท์โดยใช้ VoIP กับ PBX มาใช้ .....	5
รูปที่ 2.2.1	ภาพขั้นตอนแรกจะเป็นการแปลงสัญญาณ .....	6
รูปที่ 2.2.2	ภาพขั้นตอนการแยกสัญญาณออกเป็นส่วนๆ เพื่อทำการตัดสัญญาณ Echo .....	6
รูปที่ 2.2.3	ภาพการจัดแบ่งและจัดรูปแบบขึ้นมาใหม่ในรูปของ Frame .....	6
รูปที่ 2.2.4	ภาพการแปลง Frame ของสัญญาณให้มาอยู่ในรูปของ Packet .....	7
รูปที่ 2.2.5	ภาพการใส่ค่า IP Address ปลายทาง .....	7
รูปที่ 2.2.6	ภาพการแปลงสัญญาณ Digital PCM ให้กลับมาเป็น สัญญาณรูปแบบ ที่เราได้ยินกันอีกครั้งหนึ่ง .....	7
รูปที่ 2.2.7	ภาพขั้นตอนการตรวจสอบความผิดพลาด .....	8
รูปที่ 2.3.1	VOIP Protocol Stack .....	9
รูปที่ 2.3.2	ภาพแสดงสถาปัตยกรรมของ SIP .....	11
รูปที่ 2.3.3	ภาพแสดงการส่งข่าวสารซ้ำ .....	16
รูปที่ 2.3.4	ภาพแสดงตัวอย่างการโต้ตอบข่าวสารการเรียกที่เกี่ยวข้อง กับทั้งพรีอ็อกซีและการตรวจสอบสิทธิ์ SUA .....	18
รูปที่ 2.3.5	แสดงลักษณะการส่งข้อมูล .....	19
รูปที่ 2.3.6	แสดงตำแหน่งของ RTP ใน แพ็กเก็ต .....	19
รูปที่ 2.3.7	แสดง RTP packet header .....	19
รูปที่ 3.1.1	แสดง wizard เมื่อมีการเรียกใช้ method InvokeTuningWizard().....	23
รูปที่ 3.3.1	แสดงขั้นตอนการทำ Enable Profile และ Presence service .....	27
รูปที่ 3.3.2	แสดงขั้นตอนการทำ Enable Presence หลังจาก registration .....	34
รูปที่ 4.1.1	Use Case Diagrм แสดงรายการความต้องการทั้งหมด ที่เป็นแบบฟังก์ชันการใช้งาน .....	43
รูปที่ 4.2.1	แสดง State Diagrame ในส่วนการติดต่อของ Use Case Diagram.....	44
รูปที่ 4.2.2	แสดง Class Diagram เพื่อกำหนดคลาสที่ต้องสร้าง เพื่อการใช้งานคร่าวๆว่ามีอะไรบ้าง.....	45

## สารบัญรูป(ต่อ)

รูปที่ 4.2.3 แสดงลำดับขั้นตอนการทำงานของการทำงานของการ Login .....	46
รูปที่ 4.2.4 แสดงลำดับการทำงานของการทำงานของการ addBuddy .....	46
รูปที่ 4.2.5 แสดงลำดับขั้นตอนการทำงานของการทำงานของการ removeBuddy .....	47
รูปที่ 4.2.6 แสดงลำดับขั้นตอนการทำงานของการทำงานของการเปลี่ยนสถานะของ user .....	47
รูปที่ 4.2.7 แสดงลำดับขั้นตอนการทำงานของการทำงานของการ ติดต่อบน AppCall .....	48
รูปที่ 4.2.8 แสดงลำดับขั้นตอนการทำงานของการทำงานของการ ติดต่อบน PhoneCall .....	48
รูปที่ 4.2.9 แสดงลำดับขั้นตอนการทำงานของการทำงานของการ ติดต่อบน Messaging .....	49
รูปที่ 4.3.1 แสดงหน้าเริ่มต้นของ โปรแกรม .....	50
รูปที่ 4.3.2 แสดงหน้าของการใส่ Login .....	51
รูปที่ 4.3.3 แสดงหน้าเริ่มต้นของ โปรแกรมเมื่อทำการ Login เสร็จแล้ว .....	52
รูปที่ 4.3.4 ไอคอนเมื่อต้องการ Add Buddy .....	52
รูปที่ 4.3.5 แสดงหน้าของการติดต่อบน Messaging .....	53
รูปที่ 4.3.6 แสดงไอคอนเพื่อระบุปลายทาง เพื่อทำการติดต่อบน ไอพีแอดเดรส .....	53
รูปที่ 4.3.7 แสดงหน้าค้างให้เลือกเมื่อกดปุ่มการติดต่อบน AppCall .....	54
รูปที่ 4.3.8 แสดง Message Box เมื่อมีผู้ต้องการติดต่อเข้ามา .....	54
รูปที่ 4.3.9 แสดงรูปแบบการติดต่อบน PhoneCall .....	55
รูปที่ 4.3.10 แสดงการติดต่อบน whiteboard .....	55
รูปที่ 4.3.11 แสดง ไอคอนของการทำ Phone Tuning .....	56
รูปที่ 4.4.1a แสดงการทดลอง PhoneCall ฟังก์ชัน A .....	56
รูปที่ 4.4.1b แสดงการทดลอง PhoneCall ฟังก์ชัน B .....	56
รูปที่ 4.4.2 แสดงการทดลอง AppCall .....	57
รูปที่ 4.4.3 แสดงการทดลอง Messaging .....	57
รูปที่ 4.4.4 แสดงการทดลอง Application Sharing .....	58

# บทที่ 1

## บทนำ

### 1.1. ความสำคัญและที่มา

ในปัจจุบันเครือข่ายอินเทอร์เน็ตได้มีจำนวนผู้ใช้เพิ่มขึ้นอย่างรวดเร็ว เนื่องจากผู้ใช้สามารถค้นหาข้อมูลที่ต้องการจากแหล่งข้อมูลขนาดใหญ่และสามารถติดต่อสื่อสารกับบุคคลจากที่ต่างๆ ได้อย่างสะดวกนอกจากนี้การพัฒนาการให้บริการในรูปแบบใหม่เพื่อรองรับความต้องการของผู้ใช้ได้เพิ่มขึ้นอย่างรวดเร็วเนื่องมาจากความยืดหยุ่นของเครือข่ายอินเทอร์เน็ตทำให้สามารถพัฒนาการให้บริการได้ง่ายกว่าเครือข่ายอื่นๆ

การให้บริการอย่างหนึ่งที่กำลังได้รับความนิยมอย่างมากคือการส่งข้อมูลเสียงผ่านเครือข่าย รวมไปถึงการส่งข้อมูลแบบเรียลไทม์ (Real-Time) ในรูปแบบอื่นๆ การให้บริการแบบเรียลไทม์นี้ ช่วยอำนวยความสะดวกต่อผู้ใช้อย่างมาก เนื่องจากการประหยัดค่าใช้จ่ายในการติดต่อสื่อสาร ช่วยประหยัดค่าโทรศัพท์ที่ทางไกล รวมทั้งยังสามารถส่งข้อมูล เสียง, ภาพ, วิดีโอ และข้อความ รวมไปถึงในเครือข่ายอินเทอร์เน็ตเพียงเครือข่ายเดียว และเมื่อพิจารณาในแง่ของผู้ให้บริการหรือผู้ดูแลเครือข่าย การพัฒนาปรับปรุงเครือข่ายหรือการให้บริการ สามารถทำได้ง่ายและสะดวกกว่า การติดต่อสื่อสารในรูปแบบอื่น เนื่องจากไม่จำเป็นต้องแยกเครือข่ายสำหรับข้อมูลเสียงและวิดีโอ ออกจากกัน

อย่างไรก็ตามเนื่องจากเครือข่ายอินเทอร์เน็ตเป็นเครือข่ายที่ไม่รับประกันในเรื่องคุณภาพของการให้บริการ เครือข่ายอินเทอร์เน็ตให้บริการการส่งข้อมูลแบบทั่วไปซึ่งไม่ต้องการคุณสมบัติเรียลไทม์แต่ในการส่งข้อมูลที่อยู่ในรูปแบบของสัญญาณเสียงนั้นจำเป็นต้องใช้คุณสมบัติแบบเรียลไทม์ ดังนั้นจึงจำเป็นต้องมีโปรโตคอลที่สามารถรองรับคุณสมบัติดังกล่าวในการส่งข้อมูล ซึ่งได้แก่ โปรโตคอลซิป (SIP) เป็นโปรโตคอลในชั้นแอปพลิเคชัน และให้บริการของโปรโตคอลชั้นที่ต่ำกว่า สามารถขอรับบริการได้ทั้ง UDP และ TCP โปรโตคอลซิป มีหน้าที่ในการสร้าง, สิ้นสุด, การเชื่อมต่อ, หาตำแหน่งของเครื่องปลายทาง แต่เนื่องจากฟังก์ชันของโปรโตคอลซิปยังมีขอบเขตที่จำกัดอยู่ ดังนั้นจึงได้นำโปรโตคอลอื่นมาช่วยในการทำงานด้วยได้แก่ RTP/RTCP เพื่อให้สามารถให้บริการได้อย่างมีประสิทธิภาพมากขึ้น

## 1.2. วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษาการพัฒนาโปรแกรมบนแพลตฟอร์มวินโดวส์(พีซี)
- 1.2.2 เพื่อศึกษาการทำงานของโปรโตคอลซีพ
- 1.2.3 เพื่อศึกษาการส่งข้อมูลแบบมัลติมีเดีย
- 1.2.4 เพื่อการพัฒนาโปรแกรมติดต่อสื่อสารด้วยเสียงโดยใช้โปรโตคอลซีพผ่านซีพเซิร์ฟเวอร์

## 1.3. ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 ได้ศึกษาการพัฒนาโปรแกรมบนแพลตฟอร์มวินโดวส์
- 1.3.2 ได้ศึกษาการส่งข้อมูลแบบมัลติมีเดีย
- 1.3.4 ได้พัฒนาโปรแกรมติดต่อสื่อสารด้วยเสียงโดยใช้โปรโตคอลซีพผ่านซีพเซิร์ฟเวอร์
- 1.3.5 ได้สร้างโปรแกรมต้นแบบในการพัฒนาซีพแอปพลิเคชันบนเครื่องพีซี

## 1.4. ขอบเขตของโครงการ

พัฒนาโปรแกรมการติดต่อสื่อสารด้วยเสียงบนแพลตฟอร์มพีซี ซึ่งสามารถติดต่อสื่อสารกันได้โดยผ่านทางซีพเซิร์ฟเวอร์ โดยมีรายละเอียดดังนี้

- 1.4.1 สามารถทำการติดต่อระหว่างไคลเอนท์ได้ ผ่านทางซีพเซิร์ฟเวอร์
- 1.4.2 สามารถติดต่อสื่อสารกันด้วยเสียงและภาพวิดีโอได้
- 1.4.3 สามารถส่งข้อความระหว่างไคลเอนท์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### Voice over the Internet Protocol : VOIP

#### 2.1 หลักการพื้นฐานของเครือข่ายไอพี

เครือข่ายไอพี (Internet Protocol) มีพัฒนามาจากรากฐานระบบการสื่อสารแบบแพ็กเก็ต โดยระบบมีการกำหนดแอดเดรส ที่เรียกว่า ไอพีแอดเดรส จากไอพีแอดเดรสหนึ่งไปยัง อีกไอพีแอดเดรสหนึ่ง ใช้หลักการบรรจุข้อมูลใส่ใน แพ็กเก็ต แล้วส่งไปในเครือข่าย ระบบการจัดส่งแพ็กเก็ตกระทำด้วยอุปกรณ์สื่อสารจำพวกเราเตอร์ มีหลักพื้นฐานการส่งแบบ ไปรษณีย์สมัยเก่า บางที่เราจึงเรียกการส่งแบบนี้ว่า คาด้าแกรม

การสื่อสารแบบ ไอพีแพ็กเก็ต จะเป็นการส่งแพ็กเก็ตเข้าไปในเครือข่าย โดยไม่มีการประกันว่า แพ็กเก็ตนั้นจะถึงปลายทาง เมื่อไร ดังนั้นรูปแบบของเครือข่ายไอพีจึงไม่เหมาะสมกับการสื่อสารแบบต่อเนื่อง เช่น ส่งเสียง หรือวิดีโอ

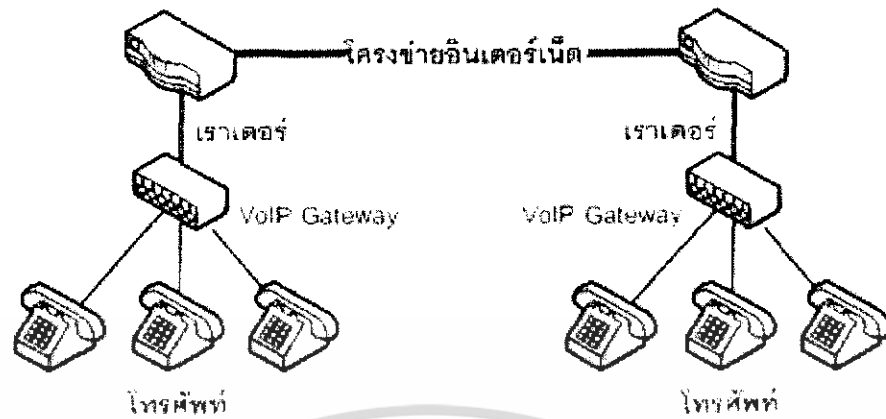


รูปที่ 2.1.1 ภาพการส่ง แพ็กเก็ตระหว่างเครือข่าย

ครั้งเมื่อมีเครือข่ายไอพีกว้างขวางและเชื่อมโยงกันมากขึ้น ความต้องการส่งสัญญาณข้อมูลเสียงที่ได้คุณภาพก็เกิดขึ้น สิ่งที่สำคัญ คือระบบประกันคุณภาพการสื่อสาร โดยจัดลำดับความสำคัญ หรือจองช่องสัญญาณไว้ให้ก่อน ระบบการสื่อสารในรูปแบบใหม่นี้ จะต้องกระทำโดยเราท์เตอร์

การส่งเสียงบนเครือข่ายไอพี หรือเรียกว่า VoIP-Voice Over IP เป็นระบบที่นำสัญญาณข้อมูลเสียงมาบรรจุลงเป็นแพ็กเก็ต ไอพี แล้วส่งไปโดยที่เร้าเตอร์มีวิธีการปรับตัวเพื่อรับสัญญาณแพ็กเก็ต และยังแก้ปัญหาบางอย่างให้ เช่น การบีบอัดสัญญาณเสียง ให้มีขนาดเล็กลง การแก้ปัญหาเมื่อมีบางแพ็กเก็ตสูญหาย หรือ ได้มალ่าช้า

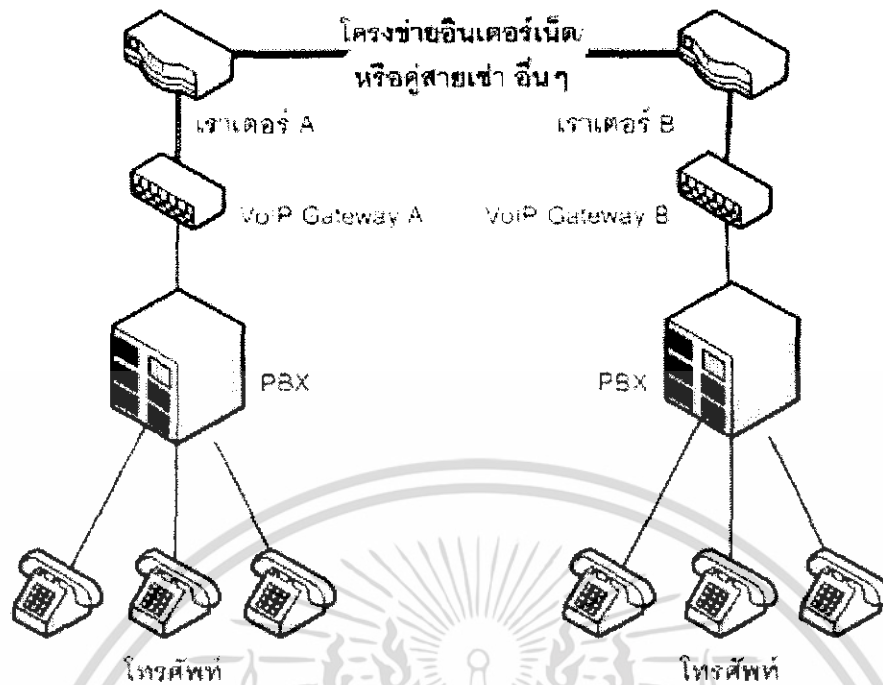
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1.2 ภาพการเชื่อมต่อเครือข่ายโทรศัพท์ผ่านอินเทอร์เน็ตโดยนำ VoIP มาใช้

ระบบ VoIP เป็นระบบที่นำสัญญาณเสียงที่ผ่านการดิจิไทซ์ โดยหนึ่งช่องเสียงเมื่อแปลงเป็นข้อมูลจะมีขนาด 64 กิโลบิตต่อ วินาที การนำข้อมูลเสียงขนาด 64 Kbps นี้ ต้องนำมาบีบอัด โดยทั่วไปจะเหลือประมาณ 10 Kbps ต่อช่องสัญญาณเสียงแล้วจึง บรรจุลงใน ไอพีแพ็กเก็ต เพื่อส่งผ่านทางเครือข่ายไอพี การสื่อสารผ่านทางเครือข่ายไอพีต้องมีเราเตอร์ที่ทำหน้าที่พิเศษเพื่อประกันคุณภาพช่องสัญญาณ ไอพีนี้ เพื่อให้ข้อมูล ไปถึง ปลายทางหรือกลับมาได้อย่างถูกต้อง และอาจมีการให้สิทธิพิเศษก่อนแพ็กเก็ต ไอพีอื่น เพื่อการให้บริการที่ทำให้เสียงมีคุณภาพ

ช่องสัญญาณ ไอพีนี้ เพื่อให้ข้อมูล ไปถึง ปลายทางหรือกลับมาได้อย่างถูกต้อง และอาจมีการให้สิทธิพิเศษก่อนแพ็กเก็ต ไอพีอื่น (Quality of Service : QoS) เพื่อการให้บริการที่ทำให้เสียงมีคุณภาพจากระบบดังกล่าวนี้เอง จึงสามารถนำมาประยุกต์ใช้กับระบบเชื่อมโยงเครือข่ายโทรศัพท์ระหว่างสำนักงาน โดยแต่ละสำนักงานสามารถใช้ระบบสื่อสาร โทรศัพท์ผ่านทางเครือข่ายไอพี (VoIP) รวมถึงยังสามารถรับส่งข้อมูล (data) ไปพร้อมๆ กันได้



รูปที่ 2.1.3 ภาพการเชื่อมต่อระบบโทรศัพท์โดยใช้ VoIP กับ PBX มาใช้

ด้วยวิธีการสื่อสารแบบ VoIP จึงทำให้ระบบโทรศัพท์ที่เป็นตู้ชุมสายภายในขององค์กรสามารถเชื่อมถึงกันผ่านทางเครือข่าย ไอพี การสื่อสารแบบนี้ทำให้สามารถใช้โทรศัพท์ข้ามถึงกันได้ ในลักษณะ PABX กับ PABX และทำให้ประหยัดค่าใช้จ่ายได้มาก

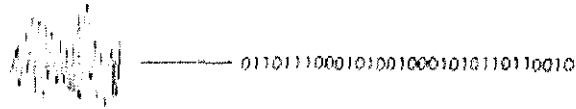
## 2.2 ขั้นตอนการทำงานของ VoIP

### 2.2.1 ขั้นตอนการทำงานเบื้องต้น

1. เมื่อผู้พูด โทรศัพท์จากเครื่องโทรศัพท์ธรรมดา หรือพูดผ่านไมโครโฟนที่ถูกต่อเข้ากับการ์ด เสียงของเครื่องคอมพิวเตอร์คลื่นสัญญาณเสียง แบบ อนาล็อกก็จะได้รับการแปลงเป็นสัญญาณดิจิทัล จากนั้นจะถูกบีบอัดด้วยตัวถอดรหัสผ่านอุปกรณ์ PBX (Private Box Exchange) หรือ VoIP Gateway
2. เมื่อผ่าน VoIP Gateway แล้วก็จะถูกส่งต่อไปยัง Gatekeeper เพื่อค้นหาเครื่องปลายทางที่จะรับการติดต่อ เช่น หมายเลขไอพี หมายเลข โทรศัพท์ เป็นต้น แล้วแปลงเป็นแพ็กเกจข้อมูลส่งออกไปบนระบบ เครือข่ายอินเทอร์เน็ตนั่นเอง
3. จะผ่านมาที่ VoIP Gateway ปลายทาง แล้วจะจึงทำการย่อนกระบวนการทั้งหมดเพื่อส่งให้กับผู้รับปลายทางต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.2.2. กระบวนการทำงานของเทคโนโลยี VoIP มีรายละเอียดการทำงานดังนี้  
Conversion to PCM (Pulse Code Modulation) จะเป็นการแปลงสัญญาณ Analog ให้ไปอยู่ในรูปแบบสัญญาณ Digital หรือที่เรียกว่า PCM



### PCM (Pulse Code Modulation)

รูปที่ 2.2.1 ภาพขั้นตอนแรกจะเป็นการแปลงสัญญาณ

Removal of Echo จะเป็นการมีการแยกสัญญาณออกเป็นส่วนๆ เพื่อทำการตัดสัญญาณ Echo ออก ซึ่งกระบวนการนี้จะถูกจัดการ โดย DSP (Digital Signal Processors)

0110111000101001000101011011001001101001001011

### Removal of Echo

รูปที่ 2.2.2 ภาพขั้นตอนการแยกสัญญาณออกเป็นส่วนๆ เพื่อทำการตัดสัญญาณ

Echo ออก

Framing ในส่วนของสัญญาณที่เหลือนั้น ก็จะถูกแบ่งและจัดรูปแบบขึ้นมาใหม่ในรูปของ Frame ซึ่งกระบวนการนี้จะถูกจัดการ โดยรูปแบบการบีบอัดที่เรียกว่า CODEC หลังจากกระบวนการนี้แล้ว Frame ของสัญญาณเสียงจะถูกสร้างขึ้น

0110111000101001000101011011001001101001001

### Framing Process

รูปที่ 2.2.3 ภาพการจัดแบ่งและจัดรูปแบบขึ้นมาใหม่ในรูปของ Frame

Packetisation ในกระบวนการนี้จะเป็นการแปลง Frame ของสัญญาณให้มาอยู่ในรูปของ Packet ซึ่งจะมีการเพิ่ม Header เข้าไปใน Packet โดยในส่วนของ Header นั้น ก็จะประกอบไปด้วยข้อมูลที่เรียกว่า Sequence Number และ Time Stamp หลังจากนั้น Packet นี้จะถูกส่งต่อไปที่ Host Processor

**RTP:** 0110111000101001000101011011001001101001001

## Packetisation Process

### รูปที่ 2.2.4 ภาพการแปลง Frame ของสัญญาณให้มาอยู่ในรูปของ Packet

Address and Delivery หลังจากที่ได้แปลงสัญญาณให้อยู่ในรูปของ Packet แล้ว ข้อมูลนั้น จะถูกนำมาวิเคราะห์และใส่ค่า IP Address ปลายทาง

IP:UDP **RTP:** 0110111000101001000101011011001001101001001

## Address and Delivery

### รูปที่ 2.2.5 ภาพการใส่ค่า IP Address ปลายทาง

Conversion to Analog หลังจากที่ได้ทำการใส่ค่าของ IP Address ปลายทางไปใน Header ของ Packet แล้วนั้น เมื่อ Packet เหล่านั้น ไปถึงด้านปลายทาง ข้อมูล Header เหล่านี้จะถูกแยกออกเพื่อให้เหลือแค่ Voice Frame หลังจากนั้นก็จะทำการแปลงสัญญาณ Digital PCM ให้กลับมาเป็นสัญญาณรูปแบบ Analog ที่เป็นสัญญาณเสียงที่เราได้ยินกันอีกครั้งหนึ่ง

011011100010100100010101101 \_\_\_\_\_



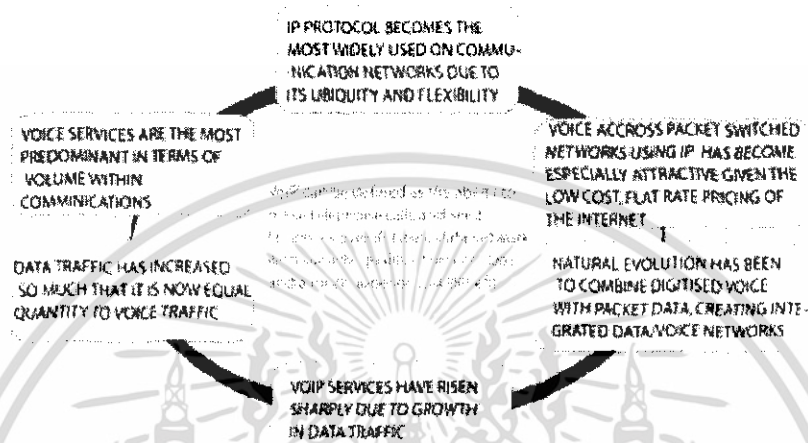
## Conversion to Analog

### รูปที่ 2.2.6 ภาพการแปลงสัญญาณ Digital PCM ให้กลับมาเป็นสัญญาณรูปแบบ ที่เราได้ยินกันอีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Error Correction กระบวนการนี้จะเป็กระบวนการที่ใช้ในการตรวจสอบและแก้ไขข้อผิดพลาดซึ่งอาจเกิดขึ้นระหว่างการส่งสัญญาณและนำมาซึ่งความผิดเพี้ยนหรือความเสียหายของสัญญาณจนทำให้เราไม่สามารถทำการสื่อสารอย่างถูกต้องได้

## Reasons that Encourage VoIP

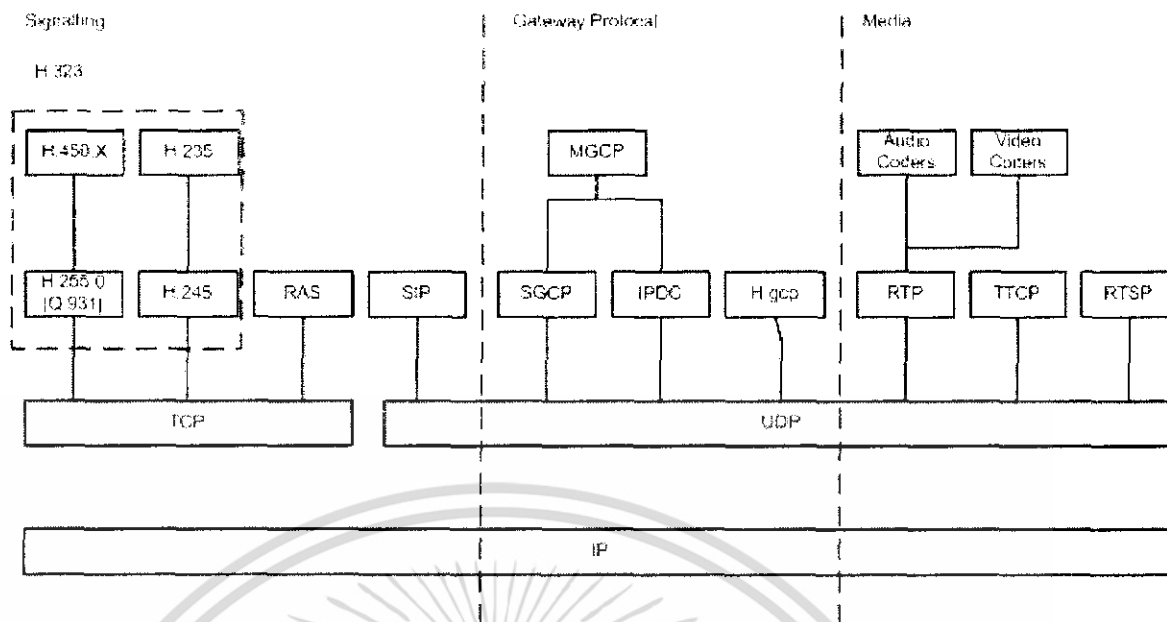


รูปที่ 2.2.7 ภาพขั้นตอนการตรวจสอบความผิดพลาด

### 2.3. Standard of VoIP Technology

โพรโตคอลสำหรับ VoIP มีอยู่ 2 มาตรฐานคือ H.323[2] และ SIP(Session Initial Protocol)[1] โพรโตคอล H.323 เป็นโพรโตคอลที่พัฒนาโดย ITU-T (International Telecommunications Union- Telecommunications section) ส่วน SIP ถูกพัฒนาโดย IETF (Internet Engineering Task Force) มาตรฐานเหล่านี้ เราสามารถเรียกได้คืออย่างหนึ่งว่า “Call Control Technologies” โพรโตคอลทั้งสองมีหน้าที่หลักในการสร้าง สิ้นสุด และการเปลี่ยนแปลงรายละเอียดของการเรียก ระหว่างผู้ใช้ VoIP รวมทั้งยังสามารถให้ฟังก์ชันเพิ่มเติมอื่นๆ โพรโตคอลทั้งสองเป็นโพรโตคอลสำหรับ VoIP ซึ่งใช้บริการชุดโพรโตคอล TCP/IP ในชั้นต่ำกว่า และสามารถใช้งานร่วมกับโพรโตคอลอื่น เพื่อให้เกิดการบริการที่มีคุณภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3.1 VOIP Protocol Stack

โพรโทคอล H.323 และ SIP เป็นโพรโทคอลในชั้นแอปพลิเคชัน (application layer) และใช้บริการของโพรโทคอลในชั้นที่ต่ำกว่า SIP สามารถใช้ได้ทั้ง UDP และ TCP ส่วน H.323 ใช้ TCP เท่านั้น แต่เนื่องจากว่าฟังก์ชันของ H.323 และ SIP มีขอบเขตจำกัด ดังนั้นจึงได้นำโพรโทคอลอื่นมาช่วยในการทำงาน ซึ่งได้แก่ RTSP (Real-time Streaming Protocol) [3] RSVP (Resource Reservation Protocol) [7] และ RTP/RTCP [4] ซึ่งทำให้ VoIP สามารถให้บริการได้อย่างมีประสิทธิภาพมากขึ้น โพรโทคอลดังกล่าวเป็นโพรโทคอลในชั้นแอปพลิเคชันซึ่งทำงานอยู่บนชุดโพรโทคอล TCP/IP โพรโทคอลเหล่านี้ไม่ได้เป็นโพรโทคอลเฉพาะสำหรับ VoIP ดังนั้นจะกล่าวถึงอย่างคร่าวๆ ในส่วนนี้ แต่สำหรับโพรโทคอล SIP ซึ่งเป็นโพรโทคอลหลักสำหรับ VoIP จะกล่าวถึงรายละเอียดในหัวข้อถัดไป

### 2.3.1 SIP: Session Initial Protocol)

มาตรฐาน SIP นั้นถือเป็นมาตรฐานใหม่ในการใช้งานเทคโนโลยี VoIP โดยที่มาตรฐาน SIP นั้น ได้ถูกออกแบบมาให้ใช้งานกับระบบ IP โดยเฉพาะ ซึ่งโดยปกติแล้วเราจะแนะนำให้ลูกค้าใหม่ที่จะมีการใช้งาน VoIP ให้มีการใช้งานอยู่บนมาตรฐาน SIP โดยมีเนื้อหาสำคัญดังนี้

1.มาตรฐาน SIP นั้นเป็นมาตรฐานภายใต้ IETF Standard ซึ่งถูกออกแบบมาสำหรับการเชื่อมต่อ VoIP

2.มาตรฐาน SIP นั้นจะเป็นมาตรฐาน Application Layer Control Protocol สำหรับการเริ่มต้น (Creating), การปรับเปลี่ยน (Modifying) และการสิ้นสุด (Terminating) ของ Session หรือการติดต่อสื่อสารหนึ่งครั้ง

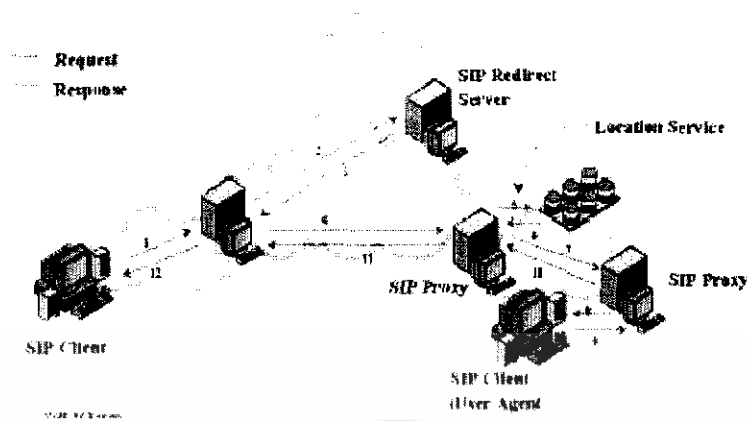
3.มาตรฐาน SIP จะมีสถาปัตยกรรมการทำงานคล้ายคลึงการทำงานแบบ Client-Server Protocol

4.เป็นมาตรฐานที่มี Reliability ที่ค่อนข้างสูง

SIP เป็นโปรโตคอลใช้งานสำหรับ IP Telephony ที่กำหนดโดย IETF (Internet Engineering Task Force) SIP เป็นโปรโตคอลในชั้นแอปพลิเคชันซึ่งทำหน้าที่ในการสร้าง สิ้นสุด และเปลี่ยนแปลงแก้ไข เซสชันของพหุสื่อ (multimedia session) หรือ การเรียก ซึ่งรวมถึง Internet telephony การประชุมแบบพหุสื่อ (multimedia conference) และแอปพลิเคชันอื่นที่คล้ายคลึงกัน SIP เป็นโปรโตคอลไคลเอ็นท์-เซิร์ฟเวอร์ (client-server) โดยใช้การส่งข้อมูลในรูปของตัวอักษร (text based) เช่นเดียวกับโปรโตคอล HTTP (Hypertext Transfer Protocol) รวมทั้งยังมีกลไกที่คล้ายคลึงกัน ทำให้สามารถใช้เซดเตอร์และกลไกที่มีอยู่บางอย่างของ HTTP ได้ สำหรับฟังก์ชันที่ SIP สนับสนุนมีดังนี้

- User location การกำหนด endpoint ที่ใช้ในเซสชันการสื่อสาร
- User capabilities การกำหนดขีดความสามารถของมีเดียที่ใช้ในการสื่อสาร
- User availability การกำหนดความต้องการของผู้ถูกเรียกว่าต้องการเข้าร่วมในเซสชันหรือไม่
- Call setup การสร้าง การเรียก และกำหนดพารามิเตอร์ของการเรียก
- Call handling การจัดการกับ การเรียก รวมทั้งการ โอนย้าย การเรียก และการสิ้นสุดการเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3.2 ภาพแสดงสถาปัตยกรรมของ SIP

SIP ถูกพัฒนาโดย IETF โดยเป็นส่วนหนึ่งของสถาปัตยกรรมควบคุมและข้อมูลพหุสื่อ (multimedia data and control architecture) ซึ่งรวมถึง โปรโตคอล เช่น RSVP RTP RTSP และ SDP (Session Data Protocol) เป็นต้น โดย SIP สามารถใช้งานหรือทำงานร่วมกับโปรโตคอลเหล่านี้เพื่อประสิทธิภาพที่ดีขึ้น แต่ทั้งนี้ขั้นตอนการทำงานของ SIP ไม่ขึ้นอยู่กับโปรโตคอลเหล่านี้ สำหรับใน SIP ได้ถูกพัฒนาอยู่ในเวอร์ชัน 2

#### 2.3.1.1. สถาปัตยกรรมและองค์ประกอบของ SIP (SIP architecture & Components)

SIP เป็น โปรโตคอลไคลเอนต์-เซิร์ฟเวอร์ ไคลเอนต์จะทำหน้าที่ส่งคำร้องขอให้ เซิร์ฟเวอร์เพื่อทำการประมวลผลแล้วจึงตอบสนองกลับมายัง ไคลเอนต์ ในการส่งแมสเสจร้องขอแมสเสจอาจจะถูกส่งผ่านเซิร์ฟเวอร์หลายตัวจนกระทั่งถึงเซิร์ฟเวอร์ที่สามารถตอบสนองคำร้องขอของไคลเอนต์ได้ ในระบบ SIP จะมีองค์ประกอบที่ทำหน้าที่ของไคลเอนต์และเซิร์ฟเวอร์ องค์ประกอบเหล่านี้จะทำการติดต่อสื่อสารกัน โดยใช้แมสเสจ SIP ซึ่งมีสถาปัตยกรรมดังรูปที่ 2.3.4

ใน SIP จะแบ่งองค์ประกอบเป็น 2 ชนิดหลักคือ user agent และ network server ดังรายละเอียดต่อไปนี้

1. User agent เป็น endpoint ที่ทำหน้าที่แทนผู้ใช้ในการติดต่อสื่อสารเนื่องจากว่าผู้ใช้ต้องสามารถเริ่ม การเรียก หรือตอบสนองต่อการเรียก ที่เข้ามา ดังนั้น user agent ควรจะสามารถทำหน้าที่เป็นได้ทั้งไคลเอนต์และเซิร์ฟเวอร์ในกรณีที่มีการเริ่ม การเรียก ผู้ใช้จะทำหน้าที่เป็น ไคลเอนต์เพื่อทำการร้องขอไปยังผู้ถูกเรียกซึ่งจะทำหน้าที่เป็นเซิร์ฟเวอร์ในการตอบสนองการร้องขอ โดยทั่วไป user agent จึงประกอบด้วยส่วนที่ทำหน้าที่เป็น ไคลเอนต์และเซิร์ฟเวอร์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

User agent client (UAC) จะทำหน้าที่ในการเริ่ม การเรียก โดยการส่งส่งแมสเสจร้องขอไปยังผู้ถูกเรียก โดยผ่านทาง network server

2. User agent server (UAS) จะทำหน้าที่ในการรับคำร้องขอ และตอบสนองต่อคำร้องขอ โดยจะรอการตอบสนองจากผู้ใช้ ซึ่งการตอบสนองอาจจะเป็นการยอมรับหรือปฏิเสธ การเรียก ในกรณีที่ผู้ใช้มีการใช้งานเทอร์มินัลหลายตัว ผู้ใช้ยังอาจจะกำหนดให้ UAS ทำการ redirect ไปยังที่ UAS อื่นที่ผู้ใช้ใช้งานอยู่จริง

### 2.3.1.2 Network server

เป็นเซิร์ฟเวอร์ภายในเครือข่ายซึ่งจะทำหน้าที่ในการจัดการกับแมสเสจที่ได้รับ โดยอาจจะได้รับจาก user agent หรือ network server อื่นๆ การจัดการกับแมสเสจจะขึ้นกับชนิดของเซิร์ฟเวอร์ ซึ่งมี 2 ชนิดคือ

1.Proxy server เซิร์ฟเวอร์จะทำการกำหนดเอนทิตีที่จะรับ ข้อมูลต่อไป โดย อาจจะเป็น UAS หรือ network server ก็ได้ จากนั้นเซิร์ฟเวอร์จะเป็นผู้ทำการร้องขอ ไปยังเอนทิตีนั้น พร้อมกับข้อมูลตอบสนองให้กับ UAC (หรืออาจจะเป็น network server อื่นที่ส่งข้อมูลร้องขอมา) เพื่อระบุว่ากำลังรอการตอบสนองจากผู้ถูกเรียก เมื่อเซิร์ฟเวอร์ได้รับการตอบสนองจากผู้ถูกเรียก หรือ UAS เซิร์ฟเวอร์จึงจะส่งแมสเสจตอบสนองต่อกลับไปให้กับ UAC ดังรูปที่ 22. เซิร์ฟเวอร์ชนิดนี้จะทำหน้าที่เป็นทั้ง โคลเอนท์และเซิร์ฟเวอร์ ในกรณีที่ส่งแมสเสจร้องขอจะเป็น โคลเอนท์ส่วน ในกรณีที่ส่งข้อมูลตอบสนองจะเป็นเซิร์ฟเวอร์

2. Redirect server เมื่อเซิร์ฟเวอร์ได้รับแมสเสจร้องขอแล้วจะกำหนดเอนทิตีที่จะรับข้อมูลต่อไป จากนั้นเซิร์ฟเวอร์จะส่งแอดเดรสของเอนทิตีนั้น ไปให้กับ UAC หรือ network server ที่ส่งข้อมูลร้องขอมา เมื่อ UAC (หรือ network server) ได้รับแอดเดรส แล้วจึงจะทำการส่งคำร้องไปยังเซิร์ฟเวอร์นั้นด้วยตนเอง

เนื่องจากว่าผู้ใช้จะมีการเปลี่ยนเทอร์มินัลที่ใช้งานได้ network server จึงจะต้องสามารถ กำหนดเอนทิตีที่รับข้อมูลเพื่อให้สามารถส่งแมสเสจให้กับผู้ถูกเรียกได้ โดย network server จะทำการติดต่อกับ location server เพื่อกำหนดเอนทิตีต่อไปที่จะรับแมสเสจ location server จะทำหน้าที่ ในการหาตำแหน่งปัจจุบันของผู้ถูกเรียก โดยการกำหนดเอนทิตีที่จะรับแมสเสจต่อไปแล้วส่งแอดเดรสของเอนทิตีนี้ให้กับ network server ข้อมูลของ location server จะได้รับจาก registrar ซึ่งทำหน้าที่ในการรับข้อมูลเกี่ยวกับตำแหน่งของผู้ใช้แล้วส่งข้อมูลนี้ให้กับ location server ในการให้ข้อมูลของผู้ใช้กับ registrar จะทำได้โดยใช้แมสเสจ REGISTER เพื่อบอกตำแหน่งที่อยู่ของผู้ใช้ โดยทั่วไปแล้ว registrar จะถูกรวมเข้ากับ network server

### 2.3.1.3. SIP Respond

โพรโทคอล ซิป มีข้อความตอบกลับมีความหมายดังนี้

1XX Provisional 100 Trying

ตั้งแต่ 100 ถึง 199 เป็น Provisional message หมายถึง กำลังทำการ จะมีการส่งการตอบกลับประเภทนี้เมื่ออยู่ในขณะการรอ Successful message เพื่อบอกว่ากำลังทำการอยู่(เช่น กำลัง Register อยู่) 2XX Successful 200 OK

ตั้งแต่ 200 ถึง 299 เป็น Successful หมายถึง ได้ทำการเรียบร้อยแล้ว

3XX Redirection 302 Moved Temporarily

ตั้งแต่ 300 ถึง 399 เป็น Redirection message หมายถึง ทำการปรับทิศทางใหม่

4XX Client Error 404 Not Found

ตั้งแต่ 300 ถึง 399 เป็น Message error ที่เกิดจากไคลเอนท์ เช่น ไม่พบไคลเอนท์ที่ตามถึง

5XX Server Error 504 Server Time-Out

ตั้งแต่ 500 ถึง 599 เป็น Message error ที่เกิดจากเซิร์ฟเวอร์ เช่น เกิดการหมดเวลา (Time out)

6XX Global Failure 603 Decline

ตั้งแต่ 600 ถึง 699 เป็น Message error ที่เกิดจาก Global เช่น เกิดการ ปฏิเสธการร้องขอ

### 2.3.1.4 Header Field

Accept-Contact	a	Refer-to	r
Allow-Event	u	Referred-by	b
Call-ID	I	Reject-Contact	j
Contact	m	To	t
Content-Encoding	e	Via	v
Content-length	l		
Content-Type	c		
Event	o		
From	f		
Subject	s		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.1.5. SIP Application

ซิปโพรโตคอล (SIP Protocol) สนับสนุนการติดต่อสื่อสารแบบบุคคลต่อบุคคล โดยรวมสื่อที่หลากหลายไว้ด้วยกัน เช่น ตัวหนังสือ,รูปภาพ,เสียง และวิดีโอ เป็นต้น นอกจากนี้ยังใช้ในการบอกตำแหน่งของจุดหมายปลายทาง และแสดงสถานะต่างๆ (Presence information) ของผู้ใช้ ดังนั้นจึงเกิดแอปพลิเคชันต่างๆ เช่น เรียลไทม์วิดีโอแชร์ริง (Real-Time Video Sharing), พุชทูทอล์กโอเวอร์เซลลูลาร์ (PoC : Push-to-talk Over Cellular), โพรแกรมรับส่งสารด่วน (Instant Messaging) และ เกมออนไลน์แบบจุดต่อจุด (Peer to Peer Game Online) เป็นต้น โดยในที่นี้จะยกตัวอย่าง 2 แอปพลิเคชัน คือ เกมออนไลน์แบบจุดต่อจุดบนโทรศัพท์มือถือ และ โพรแกรมรับส่งสารด่วนผ่านวินโดวส์เมสเซนเจอร์ (Windows Messenger) บนเครื่องคอมพิวเตอร์

### 2.3.1.6. ชื่อและแอดเดรส (Addressing & Naming)

ในระบบ SIP การส่งแอสเสจระหว่างเอนทิตีจะต้องการระบุ SIP URL เพื่อใช้อ้างอิงถึงผู้ใช้ SIP URL จะประกอบด้วย SIP แอดเดรส รูปแบบของแอดเดรสจะอยู่ในรูปของ name@domain โดยอาจจะเป็น user@domain user@address phone-number@gateway และ user@host แอดเดรสนี้จะถูกใช้อ้างอิงถึงผู้ใช้ทั้งผู้เรียกและผู้ถูกเรียกในการส่งแอสเสจ ตัวอย่างของ SIP URL เช่น SIP://j.doe@example.com โดยที่ URL นี้จะอยู่ในส่วนเฮดเดอร์ของแอสเสจ ในการส่งแอสเสจไปยัง SIP URL ที่ระบุไว้จะต้องมีการแปลง SIP แอดเดรสให้อยู่ในของ User@host โดยอาจจะผ่านการแปลงมากกว่าหนึ่งครั้งจนกระทั่งได้ตำแหน่งที่อยู่ของผู้ใช้ ในการแปลงแอดเดรสอาจจะใช้ DNS (Domain Name Service) หรือ LDAP (Lightweight Directory Access Protocol)

### 2.3.1.7 Locating Server

ในการส่งแอสเสจจะใช้ SIP URL อ้างอิงถึงในการส่ง โดยจะต้องมีการแปลงส่วน domain ของ SIP แอดเดรสไปเป็นหมายเลข IP ซึ่งเป็น แอดเดรสของ SIP server ที่สามารถค้นหาตำแหน่งของผู้ใช้ออกไปได้ การแปลง SIP แอดเดรสอาจจะทำโดย UAC หรือ UAC จะส่งแอสเสจให้กับเซิร์ฟเวอร์ที่กำหนดซึ่งเซิร์ฟเวอร์จะเป็นผู้ที่ทำหน้าที่ในการแปลง SIP แอดเดรสแทน ในการแปลง SIP แอดเดรสสามารถใช้ DNS เข้ามาช่วยได้

### 2.3.1.8 Locate User

จากข้างต้นเมื่อได้ตำแหน่งของเซิร์ฟเวอร์ที่สามารถส่งข้อมูลให้กับผู้ถูกเรียกแล้วต่อไปจะเป็นการค้นหาตำแหน่งของผู้ถูกเรียก เมื่อ SIP server ได้รับแอสเสจร้องขอแล้ว เซิร์ฟเวอร์จะดำเนินการค้นหาผู้ใช้ที่อ้างอิงถึงใน SIP แอดเดรส โดยการร้องขอข้อมูลไปยัง location server ซึ่งจะตอบกลับด้วยรายการตำแหน่งที่เป็นไปได้ของผู้ถูกเรียก เมื่อ SIP server ได้ข้อมูลเกี่ยวกับตำแหน่งของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

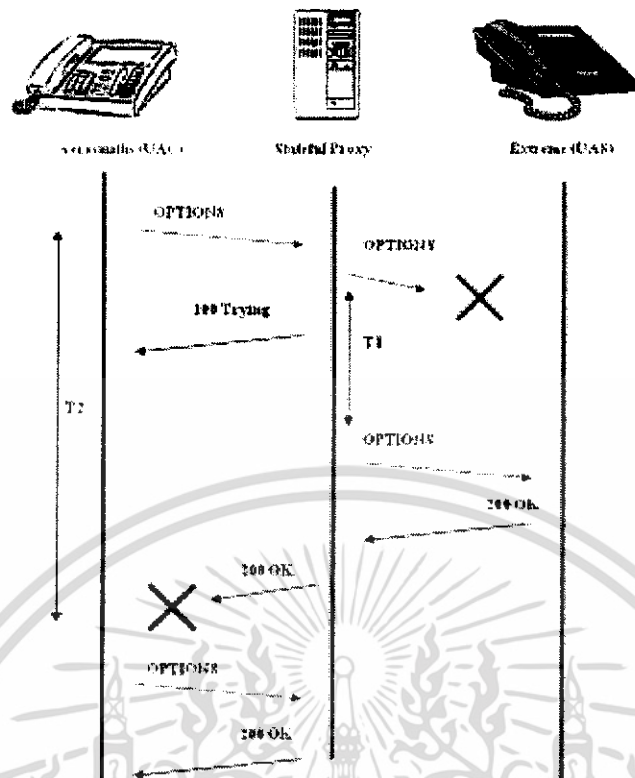
องผู้ ถ้าเป็น proxy server จะทำส่งแมสเสจร้องขอต่อไปยังตำแหน่งต่างๆ ตามรายการที่ได้รับจาก location server ไว้ โดยอาจจะส่งแบบ sequential หรือ parallel ส่วนถ้าเป็น redirect server จะส่งรายการตำแหน่งของผู้ถูกเรียกไปให้ผู้เรียกผ่าน โดยใช้เฮดเดอร์ contact เพื่อให้ผู้เรียกส่งแมสเสจร้องขอไปเอง สำหรับตำแหน่งของผู้ใช้จะต้องทำการลงทะเบียนกับ registrar โดยใช้เฮดเดอร์ REGISTER รวมทั้งยังอาจจะอัปเดต script ของผู้ใช้เองเพื่อเก็บไว้ที่เซิร์ฟเวอร์สำหรับจัดการกับการเรียก ตามความต้องการของผู้ใช้

### 2.3.1.9 ความเชื่อถือได้ (Reliability)

SIP มีกลไกหนึ่งที่ทำให้การรับส่งข้อมูลน่าเชื่อถือในขณะที่ใช้โปรโตคอล UDP (User Datagram Protocol) ซึ่งมีการเชื่อมต่อแบบ Connectionless ส่วนการรับส่งข้อมูลโดยใช้โปรโตคอล TCP (Transmission Control Protocol) นั้นมีความเชื่อถือที่ติดอยู่แล้ว จึงไม่จำเป็นต้องใช้กลไกดังกล่าว แต่แก้ไข โดยการส่งข้อมูลซ้ำเมื่อระบบแจ้งว่ามีการสูญหายของข้อมูลกลไกในการรับส่งข้อมูลให้มีความน่าเชื่อถือมีองค์ประกอบดังนี้

- ตัวจับเวลาส่งข้อมูลซ้ำ
- หมายเลขลำดับคำสั่ง
- ข่าวสารการตอบรับ

ตัวจับเวลาจะเริ่มจับเวลาเมื่อมีการส่งข่าวสารออกไป หากเกินเวลาที่กำหนดไว้ (T1) จะส่งข่าวสารเดิมซ้ำ และเริ่มจับเวลาใหม่ แต่คราวนี้หากเวลาที่จับไว้เกินค่าเวลาที่กำหนด (T2) จึงค่อยส่งข่าวสารซ้ำ ทั้งนี้การส่งข่าวสารซ้ำมีได้จำนวนจำกัด เช่น มากที่สุด 10 ครั้ง เป็นต้น



รูปที่ 2.3.3 ภาพแสดงการส่งข่าวสารซ้ำ

รูปที่ 2.3.3 แสดงตัวอย่างการส่งข่าวสารซ้ำ โดยข่าวสาร OPTIONS ที่ส่งโดย Stateful Proxy Server ไปยัง UAS (User Agent Server) สูญหาย ดังนั้นเมื่อใช้เวลานาน  $T_1$  (เช่น 500 มิลลิวินาที) จึงมีการส่งข่าวสารซ้ำ และในขณะที่ตัวจับเวลาที่ UAC เกินค่า  $T_2$  (เช่น 4 วินาที) จึงต้องส่งข่าวสาร OPTIONS ใหม่อีกครั้ง แต่พรีอ็อกซีทราบข่าวสาร 200 OK ได้สูญหายไป จึงทำการส่งซ้ำ ทั้งนี้ช่วงเวลาระหว่างหมายเลขลำดับคำสั่งไม่ได้แสดงถึงข่าวสารที่สูญหายเสมอไป แต่อาจเกิดจากการที่ไม่ได้ผ่านขั้นตอนการตรวจสอบสิทธิ์

#### 2.3.1.10 ความสามารถในการขยาย (Protocol xtension)

SIP สามารถรองรับคุณลักษณะใหม่ que เพิ่มเติมขึ้นสำหรับ เมธอด เซคเตอร์ และ status code ได้ดังนี้

- เมธอด เซิร์ฟเวอร์จะส่งเมสเสจแสดงความผิดพลาด (error message) กลับมาให้ไคลเอนท์ เมธอดที่ร้องขอมาเซิร์ฟเวอร์ไม่เข้าใจ และจะบอกเมธอดที่เซิร์ฟเวอร์เข้าใจ โดยใช้เซคเตอร์ Public และ Allow ไคลเอนท์อาจจะส่งเมสเสจร้องขอเพื่อขอทราบเมธอดที่เซิร์ฟเวอร์สนับสนุน โดยใช้ตัวเลือกที่เซคเตอร์ (header option)

- เซคเตอร์ เมื่อเอนท์ตี้ได้รับเซคเตอร์ที่ไม่เข้าใจ ก็จะละทิ้งเซคเตอร์นั้น ในกรณี que ไคลเอนท์จำเป็นต้องการใช้เซคเตอร์บางเซคเตอร์ ไคลเอนท์จะส่งเมสเสจเพื่อร้องขอเซคเตอร์ที่จำเป็นต้องใช้

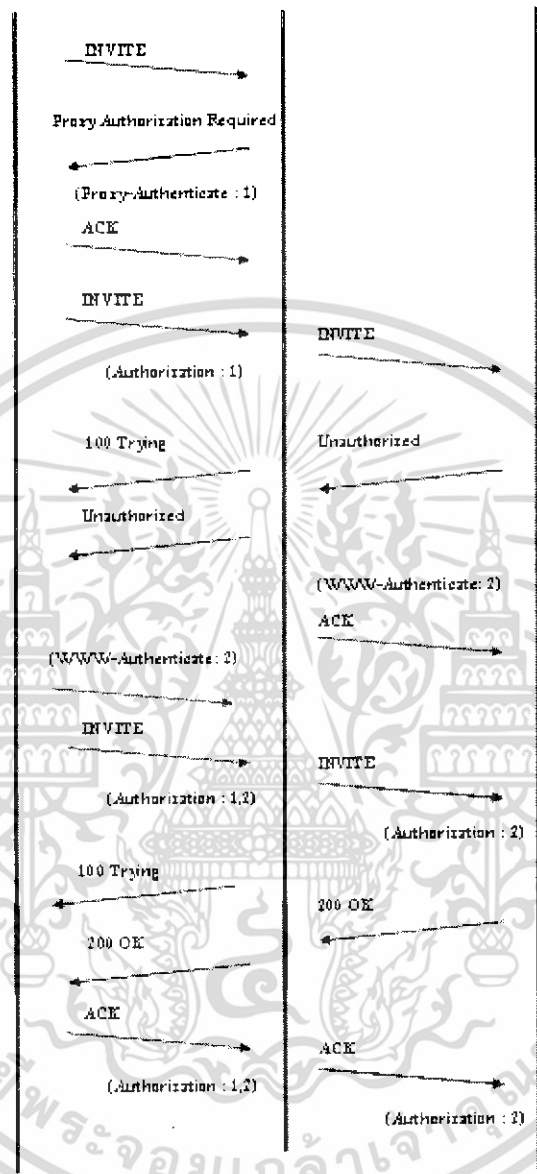
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปโดยระบุในเฮดเดอร์ Require หากมีเฮดเดอร์ที่เซิร์ฟเวอร์ไม่สามารถให้การสนับสนุนได้เซิร์ฟเวอร์จะตอบปฏิเสธกลับมา

- status code ได้แบ่งเป็นคลาสต่างๆ เช่นเดียวกับ response code ของโปรโตคอล HTTP ซึ่งไคลเอนต์ต้องเข้าใจในความหมายในแต่ละคลาสเพื่อที่จะได้ทราบผลของการร้องขอว่าสำเร็จหรือไม่ สำหรับ status code ในเมสเสจตอบจะมีข้อความต่อหลังซึ่งจะเป็นความหมายของ code ซึ่งสามารถอ่านเข้าใจได้ โดยถ้าไคลเอนต์ไม่เข้าใจในรายละเอียดของ code ทั้งหมด ไคลเอนต์จะตีความหมายเป็น X00 เมื่อ X เป็นตัวเลขตัวแรกของ status code และนอกจากนั้นอาจจะนำ PEP (protocol extension protocol) มาปรับปรุงใช้งานกับ SIP ได้ ในกรณีมีการส่งเมสเสจผ่านหลายเซิร์ฟเวอร์ จะใช้เฮดเดอร์ Via เพื่อระบุเซิร์ฟเวอร์ที่เป็นทางผ่านของเมสเสจทั้งหมด สำหรับใช้ในการส่งเมสเสจตอบสนองกลับไปให้ผู้เรียก ในระหว่างการส่งเมสเสจร้องขอและเมสเสจตอบสนองจะมีการตกลงเกี่ยวกับพารามิเตอร์ของเซสชันด้วย ซึ่งรายละเอียดจะอยู่ในส่วนของ message body เช่นในกรณีของการสื่อสารโดยใช้เสียง พารามิเตอร์จะเป็น IP แอดเดรส พอร์ตสำหรับ RTP และการเข้ารหัสเสียง หลังการสร้าง การเรียกเสร็จสมบูรณ์ ช่องสัญญาณสำหรับ RTP จะถูกสร้างขึ้นทำให้ทั้งสองฝ่ายสามารถสื่อสารกันได้ รวมทั้งยังอาจจะเชิญผู้อื่นมาเข้าร่วมในเซสชันนี้ได้ ในกรณีที่ต้องการเปลี่ยนพารามิเตอร์ของเซสชัน สามารถทำได้โดยส่งเมสเสจร้องขอใหม่อีกครั้งโดยใช้เมทอด Invite ซึ่งมี call-id เดิม ไปยังผู้ร่วมเซสชันพร้อมทั้งค่าพารามิเตอร์ของเซสชันใหม่ที่ต้องการใช้ รายละเอียดในส่วนนี้จะอยู่ในส่วนของ message body ซึ่งโดยทั่วไปจะใช้โปรโตคอล SDP ในการอธิบายความหมาย

### 2.3.1.11 การตรวจสอบสิทธิ์ผู้ใช้

รูปแบบในการตรวจสอบสิทธิ์ของผู้ใช้ (Authentication) ในโครงข่าย SIP มี 2 แบบหลัก ๆ คือการตรวจสอบสิทธิ์ SUA โดย Proxy Server, Redirect Server หรือ Registration Server และการตรวจสอบโดย SUA ตัวอื่น ปกติแล้วพร็อกซีหรือ Redirect Server ต้องการการตรวจสอบสิทธิ์ SUA เพื่ออนุญาตให้เข้าใช้บริการ ยกตัวอย่าง Proxy Server อาจต้องการการตรวจสอบสิทธิ์ของ SUA ก่อนส่งต่อข่าวสาร INVITE ไปยังเกตเวย์หรือ Registration Server อาจต้องการการตรวจสอบสิทธิ์เพื่อป้องกันการเรียกของผู้บุกรุก อย่างไรก็ตาม SUA สามารถตรวจสอบสิทธิ์ซึ่งกันและกันเพื่อตรวจสอบว่ากำลังสนทนากับใคร



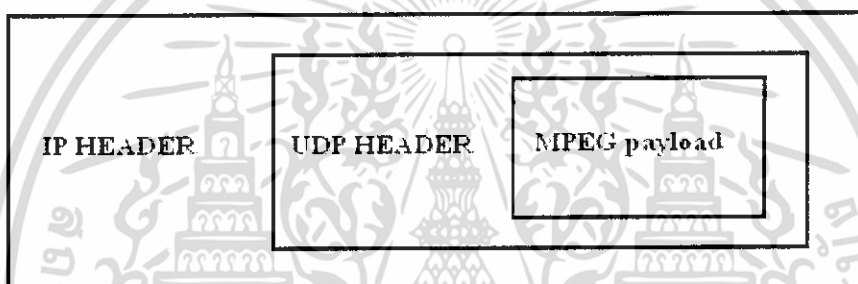
รูปที่ 2.3.4 ภาพแสดงตัวอย่างการโต้ตอบข่าวสารการเรียกที่เกี่ยวข้องกับทั้งพร็อกซีและการตรวจสอบสิทธิ์ SUA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2. Real Time Protocol

1. ใช้ในการส่งผ่านข้อมูลผ่าน เครือข่ายสำหรับการส่งข้อมูลพวกใช้เวลาจริง เช่น วิดีโอ
2. สามารถใช้งานร่วมกับ โปรแกรมประยุกต์โครงข่ายมัลติมีเดียอื่นได้
3. อาร์ทีพีไม่เป็นแบบ Connection-Oriented
4. ไม่มีความผิดพลาดในการเรียงข้อมูล ซึ่งแตกต่างจากโปรโตคอล ยูดีพี เมื่อส่งแล้วมีปัญหาในการลำดับก่อนหลัง
5. ข้อมูลที่จะส่งจะถูกควบคุมด้วย RTCP :Real Time Control Protocol
6. RTP ไม่มีการรับประกันคุณภาพของการส่งข้อมูลที่ หมายความว่าไม่มีกลไกใดๆยืนยันว่าการส่งสำเร็จหรือไม่

#### 2.3.2.1 สถาปัตยกรรมของ RTP



รูปที่ 2.3.5 แสดงลักษณะการส่งข้อมูล



รูปที่ 2.3.6 แสดงตำแหน่งของ RTP ใน แพ็คเก็ต

Payload Type	Sequence Number	Timestamp	Synchronization Source Identifier	Miscellaneous Fields
--------------	-----------------	-----------	-----------------------------------	----------------------

รูปที่ 2.3.7 แสดง RTP packet header

ซึ่งภายใน RTP packet header นั้นจะประกอบด้วยส่วนต่างๆดังนี้

-Payload Type จะเป็นตัวบอกรูปแบบการเข้ารหัสทาง วิดีทัศน์ ดังตารางที่แสดงดังนี้

ตารางที่ 1 แสดง Payload Type Number กับ Video Formate

Payload Type Number	Video Formate
26	Motion JPEG
31	H.261
32	MPEG 1 Video
33	MPEG 2 Video

-Sequence Number

-Timestamp ทำให้สามารถนำมาจัดเรียงข้อมูลได้อย่างถูกต้อง

-Synchronization Soucre Identifer (SSRC)

### 2.3.2.2 RTP Control Protocol

แพ็คเก็ต RTP จะถูกส่งออกไปเป็นช่วงๆ เพื่อเก็บข้อมูลรายงานทางสถิติของ ทางฝ่ายส่ง และฝ่ายรับเพื่อการนำไปใช้ประโยชน์ ซึ่งสถิตินี้ประกอบด้วย จำนวนของแพ็คเก็ตที่ส่งออกไป และจำนวนของแพ็คเก็ตที่สูญเสีย รายงานการรับนี้ประกอบด้วย

-SSRC (Synchronization Soucre Identifer )ของสตรีม RTP ที่รายงานเกี่ยวกับแหล่งกำเนิด

-เศษส่วนของแพ็คเก็ตที่สูญเสียในสายข้อมูล RTP นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

# Enhancing Rich Client Communications with the Microsoft\* Real-Time Communications API

### 3.1 Introduction

ในส่วนแรก “การรวมการสื่อสารของ client หลายๆตัวด้วย Microsoft Real-Time Communications API” และอธิบายถึงตัวอย่างว่าทำอะไรและ ความสามารถโดยรวมของมัน เหมือน audio/video conferencing, instant messaging และ application sharing ใน application ที่คุณใช้ Real-Time Communications (RTC) Client API ในส่วนนี้อธิบายเพิ่มเติมถึง เหตุผลต่างๆถึง การสร้างการสื่อสารว่าทำอะไร โดยใช้ RTC Client API ในการเพิ่มแต่ละความสามารถที่แสดง ข้อมูล profile และ buddy list คุณจะได้เรียนรู้ขั้นตอนการใช้งานนี้ โดยใช้ Microsoft\* Windows\* XP application created ในส่วนแรก

Microsoft's real-time communication (RTC) API ได้ทำการ encapsulates การสื่อสารแบบใหม่ของ PC based ที่น่าสนใจเอาไว้ RTC API enables พัฒนาให้ ความสามารถในการสื่อสาร ของ Microsoft Windows XP\* based application ใดก็ตามดีขึ้น มันทำให้การพัฒนาด้วยการสร้างบล็อก ในการ add instant messaging, voice และ video-conferencing, และ application sharing/collaboration capabilities to their application สำหรับ โปรแกรม ที่เพิ่มเข้าไปจำนวนมากของ ความสามารถที่ใช้ในการสื่อสารในการที่ แอปพลิเคชัน มีให้ใช้ของ RTC API โดยตรง อันดับแรก แอปพลิเคชันต้องกำหนดความสามารถของ platform การสื่อสารของไคลเอนท์อีกความสามารหนึ่ง ของ platform ที่รู้คือ แอปพลิเคชันสามารถเสื่อกว่า ชอบ audio และ video devices ที่ใช้ระหว่างส่วน ของการสื่อสาร แอปพลิเคชันที่พร้อมจะเริ่มส่วนของสื่อสารด้วยตัวอื่นหลังจากนั้น ส่วนที่สร้าง และเพิ่มเข้าไปในส่วนของการสื่อสารและ client application coordinates จะทำการ capture, compression, และ transmission ข้อมูลจนกระทั่งฝั่งรับและผู้รับ decompresses, และ replays the transmitted data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.1 RTC Client Initialization & Platform Capabilities

ขั้นตอนแรก RTC client interface ต้องเพิ่มไปยังแอปพลิเคชันซึ่งทำการอินเตอร์เฟซเข้ามาได้โดย CoCreateInstance() using the CLSID\_RTCCClient (GUID = {7a42ea29-a2b7-40c4-b091-f6f024aa89be}) ซึ่ง อินเตอร์เฟซหนึ่งที่ยอมให้ใช้และเป็นไปได้ ที่ใช้ในการกำหนด communication session capabilities ของแพลตฟอร์ม

```
// Initialize the RTC COM object
hr = CoCreateInstance (CLSID_RTCCClient, NULL,
    CLSCTX_INPROC_SERVER, IID_IRTCCClient,
    (LPVOID *)&m_pClient);
// Initialize the client interface
hr = m_pClient->Initialize();
```

เมื่อความสารณของแพลตฟอร์มที่รู้คือ ชนิดที่อ้างถึงของการสื่อสาร และ อุปกรณ์เสริม(camera, microphone) ที่ต้องการเลือก มันเป็นสิ่งสำคัญที่ต้องจำถ้าต้องการ communication modes ที่มันไม่ยอมให้ใช้บน participant's platform, และ RTC client dynamically กำหนดโดย ชนิดของการสื่อสารที่ใช้สำหรับตัวอย่างทุกชนิดของการสื่อสารคือทางเลือกที่ดีที่สุด นี่คือนักที่บอกว่าผู้ที่อยู่ใน communication session สามารถ share applications, transmit instant messages, audio, or video เมื่อยอมให้ใช้ฟังก์ชันนั้น

```
m_pClient->SetPreferredMediaTypes ( RTCMT_ALL, VARIANT_TRUE );
```

คงยังอีกนานที่ codec จะใช้ได้โดย window RTC client ในการส่ง audio และ video data ระหว่าง session participants ที่ขึ้นอยู่กับความสารณของแพลตฟอร์ม ของคู่ที่เราทำการติดต่อใน session และ bandwidth ที่ใช้ระหว่างการสื่อสาร สำหรับ video ซึ่ง Windows RTC client รองรับ H.261 และ H.263 codecs ที่ QCIF (176x144) resolution ซึ่ง bit rate ที่ใช้ในการส่ง video data อยู่ในช่วงระหว่าง 6 และ 125-KBps มันเป็นไปได้ที่จะเกิดจากระยะทางและ วิธีการในทางปฏิบัติของ การส่ง video ของ IRTCCClient interface methods MaxBitRate และ put\_TemporalSpatialTradeOff.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 3.2 Get Ready

อธิบายถึงโครงสร้างของตระกูล XML Schema ที่อ้างถึงในบทแรก สำหรับ information เริ่มต้นของ RTC object สำหรับ source code ที่มากับบทนี้มีสองเวอร์ชันคือ: *RTCSample.exe\** เป็นตัวอย่าง code ที่แสดงทุกการทำงานของ ฟังก์ชัน ที่มีให้ของ enabling audio/video conferencing, instant messaging, presence และ buddy list *AVDCConf2.exe* ตัวอย่าง code ที่มีอยู่ใน capabilities ที่ เป็น *RTCSample.exe* ตัวอย่าง code ด้วยการรวม features แต่ละตัวเช่น whiteboard sharing, application sharing, Intel® Pentium® 4 processor ด้วย Hyper-Threading Technology<sup>1</sup> detection, และ CPU utilization meter

#### 3.2.1 สิ่งที่ต้องการ:

Microsoft\* Visual Studio\* 6.0 with SP5

Microsoft Platform SDK

SIP Server or Microsoft Windows\* .NET Server Beta 3

### 3.3 Session Initiation Protocol Server

เป็นการรวมเอา presence , profile และ buddy list เข้าไปในแอปพลิเคชันของคุณว่าคุณ ต้องการ SIP server หรือ Microsoft Windows .NET Server Beta 3 สำหรับ SIP registrar server จะ ต้องการเมื่อประสงค์จะแสดงข้อมูลและต้องการใช้งานส่วนของแต่ละ Buddy List ผู้ใช้สามารถรีจิสต์ แต่ละ presence information ด้วยเซิร์ฟเวอร์ และ ค้นคืน presence information ด้วยบริการนี้ เซิร์ฟเวอร์จะใช้ Session Initiation Protocol (SIP) และเข้าถึง protocol SIMPLE ที่อยู่ภายใต้ โพรโตคอลการสื่อสาร ซึ่ง SIP จะเตรียมการรองรับได้ดีมากสำหรับการสื่อสารแบบ multi-modal ซึ่ง SIP และ SIMPLE คือ โพรโตคอลที่ทำได้มากกว่า text messaging sharing มันสามารถจัดการ manage voice, video, application sharing และอีกหลายอย่าง

SIP protocol initiates sessions ในอินเทอร์เน็ตเวิร์คและ registers presence information สามารถทำได้ สองทางคือ PC-to-PC communication หรือ collaborative multi-media conference session ซึ่ง SIP คือ Internet Engineering Task Force (IETF) signaling protocol สำหรับ establishing, manipulating, และ tearing down sessions ซึ่ง จุดประสงค์หลักของ SIP คือช่วยรับการเริ่มต้นของ potential session participants เมื่อมันทำการ SIP มีรายละเอียดที่ "simple, extensible" IP telephony signaling protocol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.1 การเริ่มต้นสำหรับ Presence และ Buddy List Notifications

การรับ event notifications ใหม่สำหรับ presence, buddy list, และ profiling จะกำหนดตาม event ตาม event filter masks เพื่อให้แอปพลิเคชันของคุณสามารถรับ event notification จาก RTC เลเยอร์ ได้

```

RTCEF_REGISTRATION_STATE_CHANGE 0x00000002
#define RTCEF_BUDDY 0x00000100
#define RTCEF_WATCHER 0x00000200
#define RTCEF_PROFILE 0x00000400

```

ตัวอย่างอีกอันหนึ่งในการ register ทุก event คือการใช้ RTCEF\_ALL macro notify ซึ่ง RTC เลเยอร์ จะส่งทุก event ไปยังแอปพลิเคชัน

```
long lEventMask = RTCEF_ALL;
```

### 3.3.2 Handling RTC Events

ตาม code ด้านล่างแสดงให้เห็น additional events ในการโปรเซสเมื่อ registering ไปยัง SIP server ด้วย user profile ,ส่งและรับ presence information และ buddy list events ที่แต่ละ event คือ received , event filter method จะใช้การแยก RTC interface ไปยังโปรเซสในการทำ received event

```

HRESULT CAVDConfDlg::OnRTCEvent(UINT message, WPARAM wParam,
LPARAM lParam)
{
    IDispatch * pDisp = (IDispatch *)lParam;
    RTC_EVENT enEvent = (RTC_EVENT)wParam;
    HRESULT hr;

    // Based on the RTC_EVENT type, query for the
    // appropriate event interface and call a helper
    // method to handle the event
    switch ( wParam )
    {
        ...

        case RTCEF_REGISTRATION_STATE_CHANGE:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    IRTCRegistrationStateChangeEvent * pEvent = NULL;

    // Get the event handle associated with the current session.
    hr = pDisp->QueryInterface(
IID_IRTCSessionStateChangeEvent,
    (void *)&pEvent );
    if (SUCCEEDED(hr))
    {
        OnRTCSessionStateChangeEvent(pEvent);
        SAFE_RELEASE(pEvent);
    }
}

break;

case RTCE_BUDDY:
{
    IRTCBuddyEvent * pEvent = NULL;
    hr = pDisp->QueryInterface(
IID_IRTCBuddyEvent,
    (void *)&pEvent );
    if (SUCCEEDED(hr))
    {
        OnRTCBuddyEvent(pEvent);
        SAFE_RELEASE(pEvent);
    }
}

break;

case RTCE_WATCHER:
{
    IRTCWatcherEvent * pEvent = NULL;
    hr = pDisp->QueryInterface( IID_IRTCWatcherEvent,
    (void *)&pEvent );
    if (SUCCEEDED(hr))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

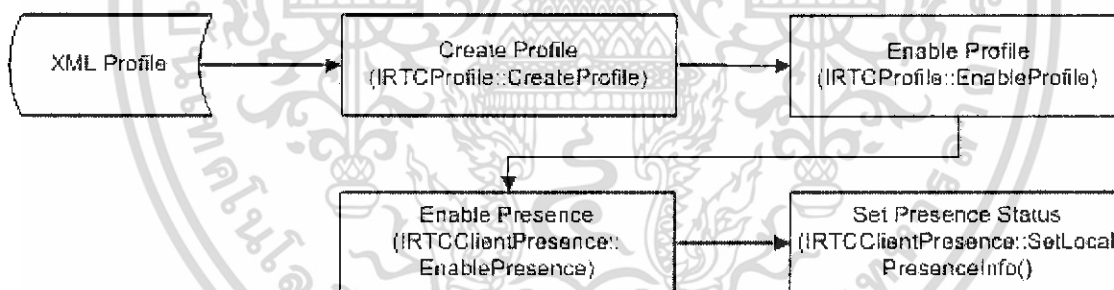
```

{
OnRTCWatcherEvent(pEvent);
SAFE_RELEASE(pEvent);
}}
break;
}
...
}

```

### 3.3.3 Enabling and Detecting Presence

Presence information service ที่มีให้ผู้ใช้แทรกไปยัง contact's presence status , บอกร contacts เหล่านั้นหรือ สถานะ, และ เรียกคู่สนทนาเหล่านั้น(or contacts) เพื่อส่งไปยัง registrar server ที่รองรับ location information ปัจจุบันของcontacts นั้นซึ่ง location สามารถเป็นได้ทั้ง PC หรือ โทรศัพท์และในอนาคตจะเป็น เซล โฟน ,เพจเจอร์ หรือ handheld device ตามไดอะแกรมแสดง high-level steps ในการ register ของไคลเอนท์บนซีพีเอวอร์และ enable Presence



รูปที่ 3.3.1 แสดงขั้นตอนการทำ Enable Profile และ Presence service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.4 SIP Server Registration

ในการทำให้ Presence service ใช้งานได้คุณต้องสร้าง Profile object สร้างโดยใช้

`IRTCClientProvisioning::CreateProfile` \*method ในการสร้าง Profile object, client application จะต้องสร้าง XML string ที่ตรงกับข้อกำหนดของ schema สำหรับ Attributes ของ XML schema ประกอบด้วย:

Provision settings – unique identifier สำหรับ Profile

User settings – User's URI, realm, และ login account information

Client settings – Information เกี่ยวกับ client application ที่ไม่ได้อ้างอิงถึง communications link คือ ส่วนของ information ที่เป็นเลือกได้

Provider settings – Information เกี่ยวกับ Internet telephony service provider (ITSP).

SIP server settings – ข้อระบุถึง SIP servers available ที่มีให้, หน้าที่ของ SIP server และชนิดของ session server ที่รองรับ

Register ที่ใช้บน SIP server และ client ต้องสร้าง XML profile string เริ่มต้นให้กับ instruction ของ RTC client API ในการสื่อสารไปยัง SIP server สำหรับตัว XML string ที่สร้าง เรียก `IRTCClientProvisioning::CreateProfile()` method ไปยังการสร้าง profile object ขึ้นตอน ต่อมาเรียก `IRTCClientProvisioning::EnableProfile()` method เป็นถาวร register ให้ผู้ใช้บน RTC server และกำหนดชนิดของ session ที่ profile ทำการ register บน server

ชนิด registration คือ: การยอมให้เข้ามาของ PC-to-PC sessions

(`RTCRF_REGISTER_INVITE_SESSIONS`), ยอมให้ instant messaging session เข้ามา

(`RTCRF_REGISTER_MESSAGE_SESSIONS`), ยอมให้ watchers เข้ามา

(`RTCRF_REGISTER_PRESENCE`), หรือยอมให้ทุก registration types

(`RTCRF_REGISTER_ALL`). Setting `RTCRF_REGISTER_PRESENCE` หรือ

`RTCRF_REGISTER_ALL` จะแจ้ง registrar server ให้ client accepts SIP SUBSCRIBE method.

นี่คือสิ่งที่ทำให้ผู้ใช้สำหรับเปลี่ยน presence status, get notifications ของการเปลี่ยน contact's presence status และคนอื่นที่เพิ่มเข้ามาใน Watcher list

watcher objects จะให้ status ของ user ใน buddy list ซึ่ง watcher สามารถเรียงลำดับ สำหรับ status ของ Presentity และ notified of changes ใน status ของ user ใน buddy list ซึ่ง watcher ทำได้ทั้ง block หรือ allow จาก status ของ Presentity เหล่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง code ที่แสดงให้เห็นขั้นตอน register user และ enable presence และ buddy list.

```

HRESULT CAVDConfDlg::DoSIPLogin(BSTR bXMLObj)
{
    HRESULT hr;
    ...
    // Get the RTC client provisioning interface
    IRTCClientProvisioning * pProv = NULL;
    hr = m_pClient->QueryInterface(
        IID_IRTCClientProvisioning,
        (void *)&pProv);
    if (FAILED(hr))
    {
        // Query Interface
        failed
        return hr;
    }
    // Create a RTC profile object from the XML
    // provisioning document
    hr = pProv->CreateProfile(bXMLObj,
    &m_pProfile);
    SAFE_FREE_STRING(bXMLObj);
    if (FAILED(hr))
    {
        // CreateProfile failed
        SAFE_RELEASE(pProv);
        return hr;
    }
    // Enable the RTC profile object
    hr = pProv->EnableProfile(m_pProfile,
    RTCRF_REGISTER_ALL);
    SAFE_RELEASE(pProv);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (FAILED(hr))
{
// EnableProfile failed
return hr;
}

// Enable presence
// It is best to enable presence immediately after enabling
the
// profile so that incoming watchers will not be lost.
// See below for EnablePresence() method
below for function definition
hr = EnablePresence(TRUE);
if (FAILED(hr))
{
// DoEnablePresence failed
DoSIPLogoff(); // See code sample
AVDConf2.exe for function sample
return hr;
}
return S_OK;
}

```

นี่คือ profile ที่ enable และคุณต้อง enable presence ป้องกันการสูญเสียการเข้ามาของ watcher event ซึ่ง EnablePresence() method แสดงว่าสร้าง buddy list อย่างไร

```

HRESULT CAVDConfDlg::EnablePresence(BOOL bEnable)
{
IRTCClientPresence * pPresence = NULL;
HRESULT hr;
// Cleanup the buddy list
ClearBuddyList();
// Get the RTC client presence interface

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hr = m_pClient->QueryInterface(
IID_IRTCCClientPresence,
(void *)&pPresence);
if (FAILED(hr))
{
// QueryInterface failed
return hr;
}

// Get the location of the presence storage
VARIANT varStorage;
VariantInit(&varStorage);
varStorage.vt = VT_BSTR;
varStorage.bstrVal = SysAllocString(L"presence.xml");
// If we are disabling presence, save the latest
// copy of the presence data to presence.xml file.
if (!bEnable &&
m_bPresenceEnabled)
{
hr = pPresence->Export(varStorage);
if (FAILED(hr))
{
// Export failed
SAFE_RELEASE(pPresence);
VariantClear(&varStorage);
return hr;
}
}

// Enable presence
hr = pPresence->EnablePresence(
bEnable ? VARIANT_TRUE : VARIANT_FALSE, varStorage);
VariantClear(&varStorage);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (FAILED(hr))
{
// EnablePresence failed
SAFE_RELEASE(pPresence);
return hr;
}

// Set the enabled flag
m_bPresenceEnabled = bEnable;

// If we are disabling presence, cleanup the
// presence data
if (!bEnable)
{
// Cleanup buddies
IRTCEnumBuddies * pEnumBuddy = NULL;
IRTCBuddy * pBuddy = NULL;
    if (FAILED(hr))
    {
// Enumerate buddies failed
SAFE_RELEASE(pPresence);
return hr;
    }
    // Enumerate user's buddy list.
    hr =
pPresence->EnumerateBuddies(&pEnumBuddy);
if (FAILED(hr))
{
// Enumerate buddies failed
SAFE_RELEASE(pPresence);
return hr;
}
while (pEnumBuddy->Next(1, &pBuddy, NULL)
== S_OK)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

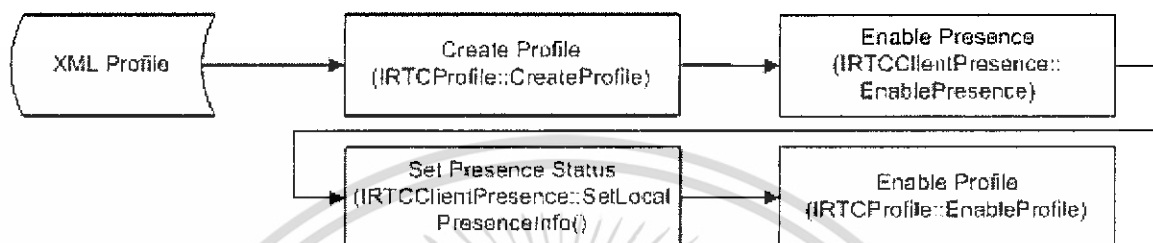
```

{
pPresence->RemoveBuddy(pBuddy);
SAFE_RELEASE(pBuddy);
}
SAFE_RELEASE(pEnumBuddy);
// Cleanup watchers
IRTCEnumWatchers * pEnumWatcher = NULL;
IRTCWatcher * pWatcher = NULL;
hr = pPresence->EnumerateWatchers(&pEnumWatcher);
if (FAILED(hr))
{
// Enumerate watchers failed
SAFE_RELEASE(pPresence);
return hr;
}
while (pEnumWatcher->Next(1, &pWatcher,
NULL) == S_OK)
{
pPresence->RemoveWatcher(pWatcher);
SAFE_RELEASE(pWatcher);
}
SAFE_RELEASE(pEnumWatcher);
}
SAFE_RELEASE(pPresence);
return S_OK;
}

```

ประกาศใน parameter ที่สองของ pPresence->EnablePresence() function และร้องขอ variant storage ไปเก็บใน buddy list และ allowed and blocked watcher lists ซึ่ง variant storage สามารถอยู่ในรูปของ file name (BSTR), DOMDocument\* object, หรือ object ใดๆ ที่supports IStream\*, ISequentialStream\*, หรือ IPersistStream\* หลังจาก register ด้วย RTC server และแอปพลิเคชันรับ RTCE\_REGISTRATION\_STATE\_CHANGE event. ด้วยการแสดงว่าการ ลงทะเบียนสำเร็จ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ในสื่อใดๆ ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ ล้มเหลว Presence service สามารถ enabled ก่อน registering Profile ของคุณบนเซิร์ฟเวอร์ซึ่งไคอะแกรมข้างล่างแสดงว่าทำอะไร Presence service จึงสามารถ enabled ก่อน register ได้ นั่นคือ enable ที่ผู้ set status ก่อน register บน server ซึ่ง user สามารถ register ใหม่บน SIP server แต่สามารถแสดง "Off-line" ที่ Watchers และเห็น contacts' status ของเขาได้ซึ่ง feature นี้เป็นประโยชน์มากเมื่อ user ต้องการ monitor group's presence status โดยไม่มี intrusive



### รูปที่ 3.3.2 แสดงขั้นตอนการทำ Enable Presence หลังจาก registration

Note the CreateProfile() method สามารถเรียกหลังจาก setting the Presence status.

ตาม XML Schema

```

<?xml version="1.0" ?>
<Schema name="ProvisioningSchema.xml"
xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
<comments>Schema Version MajorRevisionNumber = 1
MinorRevisionNumber = 0
</comments>
<ElementType name="provision" model="closed">
<AttributeType name="key" required="yes" />
<attribute type="key" />
<AttributeType name="name" required="yes" />
<attribute type="name" />
<AttributeType name="expires" dt:type="dateTime.tz"
/>
<attribute type="expires" />
<element type="user" minOccurs="1" maxOccurs="1"
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/>
<element type="sipsrv" minOccurs="1" maxOccurs="*"
/>
<element type="client" minOccurs="0" maxOccurs="1"
/>
<element type="provider" minOccurs="0" maxOccurs="1"
/>
</ElementType>
<ElementType name="user" model="closed">
<AttributeType name="uri" required="yes" />
<attribute type="uri" />
<AttributeType name="account" required="no" />
<attribute type="account" />
<AttributeType name="name" required="no" />
<attribute type="name" />
<AttributeType name="password" required="no" />
<attribute type="password" />
<AttributeType name="realm" required="no" />
<attribute type="realm" />
</ElementType>
<ElementType name="client" model="closed">
<AttributeType name="name" required="yes" />
<attribute type="name" />
<AttributeType name="banner"
dt:type="bool"
required="no" />
<attribute type="banner" />
<AttributeType name="updates"
dt:type="bool"
required="no" />
<attribute type="updates" />
<AttributeType name="minver"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dt:type="fixed.14.4"
required="no"/>
<attribute type="minver" />
<AttributeType name="curver"

dt:type="fixed.14.4"
required="no"/>
<attribute type="curver" />
<AttributeType name="updateuri"
dt:type="uri"
required="no" />
<attribute type="updateuri" />
<element type="data" minOccurs="0" maxOccurs="1"
/>
</ElementType>
<ElementType name="data" model="open" />
<ElementType name="provider" model="closed"
content="mixed">
<AttributeType name="name" />
<attribute type="name" />
<AttributeType name="homepage" dt:type="uri" required="no"
/>
<attribute type="homepage" />
<AttributeType name="helpdesk" dt:type="uri" required="no"
/>
<attribute type="helpdesk" />
<AttributeType name="personal" dt:type="uri" required="no"
/>
<attribute type="personal" />
<AttributeType name="calldisplay" dt:type="uri"
required="no" />
<attribute type="calldisplay" />

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<AttributeType name="idledisplay" dt:type="uri"
required="no" />
<attribute type="idledisplay" />
<element type="data" />

</ElementType>
<ElementType name="sipsrv" model="closed"
content="mixed">
<AttributeType name="addr" required="yes" />
<attribute type="addr" />
<AttributeType name="protocol"
dt:type="enumeration"
dt:values="TCP UDP TLS"
required="yes" />
<attribute type="protocol" />
<AttributeType name="auth"
dt:type="enumeration"
dt:values="basic digest"
required="no" />
<attribute type="auth" />
<AttributeType name="role"
dt:type="enumeration"
dt:values="proxy registrar"
required="yes" />
<attribute type="role" />
<element type="session" minOccurs="0" maxOccurs="*"
/>
</ElementType>
<ElementType name="session" model="closed">
<AttributeType name="party"
dt:type="enumeration"
dt:values="first third" />

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<attribute type="party" />
<AttributeType name="type"
dt:type="enumeration"
dt:values="pc2pc pc2ph ph2ph im" />

<attribute type="type" />
</ElementType>
</Schema>

```

### 3.3.5 การสร้าง XML Profile

นี่คือ code ที่แสดงการ create XML profile string.

```

HRESULT CSIPLogin::CreateXMLProvision(LPSTR
szURI, LPSTR szSIPIP,
LPSTR
szTransport, BSTR
*bstrBuf)
{
...
// Generate the XML provisioning document
wprintf(szBuf, "<provision key=\"AVDConf_2\"
name=\"AVDConf_2\">"
"<user uri=\"%s\" account=\"\" password=\"\" realm=\"%s\" />"
"<sipsrv addr=\"%s\" protocol=\"%s\" %s role=\"proxy\">"
"<session party=\"first\" type=\"pc2pc\" />"
"<session party=\"first\" type=\"pc2ph\" />"
"<session party=\"first\" type=\"im\" />"
"</sipsrv>"
"<sipsrv addr=\"%s\" protocol=\"%s\" %s role=\"registrar\"/>"
"</provision>".
szURIBuf, szRealm,
szSIPIP, szTransport, bBasicAuth ?

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

"auth=\"
basic\":" : "",
    szSIPIP, szTransport, bBasicAuth ?
"auth=\"
basic\":" : ""
);
...
return S_OK;
}

```

นี่คือตัวอย่างแอปพลิเคชันและ code สั้นๆ ในส่วนที่แสดงข้างบน มันไม่จำเป็นต้องรวม account userid และ password ที่ไม่ต้องการที่ SIP server อย่างไรก็ตามถ้า SIP server ร้องขอ login account , userid และ password ที่สามารถรวมอยู่ใน Profile string ในการ login ไปยัง SIP server มีเพียงข้อมูลข่าวสารเท่านั้นที่ต้องการรีจิสเตอร์ที่ user Uniform Resource Identifier (URI), realm หรือ domain, RTC server IP, authentication method, และ transport ที่ใช้ในการสื่อสารไปยัง เซิร์ฟเวอร์ที่รองรับการส่งข้อมูล คือ TCP, UDP, และ TLS ซึ่ง SIP server ต้องใช้รองรับทั้ง Basic และ Digest authentication ถ้า authentication คือ Basic และ transport ต้องมี TLS สำหรับความปลอดภัย

### 3.3.10 Creating the Buddy List และ Watcher Object

Profile ตัวหนึ่งคือ registered และ presence ที่ enabled, adding ให้กับผู้ใช้ใหม่ไปยัง buddy list คือ ตัวอย่าง ใช้ IRTCCClientPresence interface, ด้วยการ provides methods ให้กับ enable presence, add a buddy, remove a buddy, enumerate watchers, set local presence status, กำหนดอย่างไรใน application handles subscriptions จาก new Watchers, และ set the privacy mode

นี่คือตัวอย่าง code ที่ไม่ใช่ demonstrate , ว่ามีเครื่องมืออย่างไรบ้างใน privacy mode, แต่อย่างไรก็ตามมันคุ้มนี้คือ feature ถ้า user create a discrete list ของสำหรับเรียก people permitted

```

// Get the RTC client presence interface
IRTCCClientPresence * pPresence = NULL;
hr = m_pClient->QueryInterface(
    IID_IRTCCClientPresence,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(void *)&pPresence);
if (FAILED(hr))
{
// QueryInterface failed
char szBuf[256];
wsprintf (szBuf, "Failed to
Query
Presence Interface\nErr = 0x%x", hr );
MessageBox ( szBuf );
return hr;
}
// Add the buddy
IRTCBuddy * pBuddy = NULL;
hr = pPresence->AddBuddy(
bstrURI,
bstrName,
NULL,
VARIANT_TRUE,
NULL,
0,&pBuddy);
SAFE_RELEASE(pPresence);
if (FAILED(hr))
{
// Addbuddy failed
SAFE_RELEASE(pBuddy);
char szBuf[256];
wsprintf (szBuf, "Failed to
Add
Buddy to List.\nErr = 0x%x", hr );
MessageBox ( szBuf );
return hr;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Update the buddy list entry
UpdateBuddyList(pBuddy);
SAFE_RELEASE(pBuddy);
```

ถ้า new buddy สร้างสมบูรณ์ AddBuddy() method จะทำการรีเทิร์นพอยน์เตอร์ไปยัง IRTCBuddy\* interface บน created Buddy object ตัวใหม่ ใช้ IRTCBuddy\* interface และ client application จะทำการ able ในการ get buddy's presence URI, name ของ buddy, buddy's status, persistent type, และ private data รวมกับ buddy's presence.

### 3.3.11 Deregistering และ Disabling a Profile

การเรียก IRTCClientProvisioning::DisableProfile() method ในการ deregister user จาก SIP server เพื่อให้แน่ใจว่า profile object มีการแก้ไขหลังจาก calling DisableProfile() method.

### 3.3.12 Optimizing for Performance

ใน article แรกเราแสดง CPU utilization เมื่อ running RTC application บน Pentium 4 processor-based system และยังคงจำนวนมากอยู่ของ headroom สำหรับ background tasks อย่างไรก็ตามการตอบรับของ application สามารถ hindered(ล่าช้า) โดยสถาปัตยกรรมของมัน สำหรับปัญหา address นี้, create threads ในการ run แบบ parallel และ service อื่นๆ ที่ต้องการกระทำที่เป็น immediate attention ซึ่ง Running multi-threaded application บน Pentium 4 processor ด้วย Hyper-Threading Technology1 และ Windows XP ด้วย Service Pack 1 (SP1) จะแสดง application's responsiveness ที่ดีขึ้นและ enable application multitask ที่เพิ่ม efficiently.

Windows XP รุ่น SP1 ตรวจสอบที่ Pentium 4 processor ด้วย Hyper-Threading Technology1 ที่ 2 logical processors, ซึ่ง Windows XP สามารถทำ schedule twice รวมของงานที่ทำที่ CPU ที่ใช้ เปรียบเทียบกับ single logical CPU

### 3.3.8 Conclusion

Real-Time Communications (RTC) API จะทำให้คุณสร้าง functioning conferencing ได้เต็มที่และ collaboration tools, เมื่อเป็นการสื่อสารแบบ PC-to-PC, PC-to-Phone, หรือ Phone-to-Phone เราทดสอบให้เห็นใน article แรกกว่ารวดเร็วอย่างไรคุณสามารถพัฒนา audio และ video conferencing application ด้วย instant messaging และ application sharing ใน article นี้เรานำเพิ่ม capabilities ในการรวม presence และ buddy list functions ในการสร้างการสื่อสาร และ track people's availability ทั้งคู่ต่อกันด้วย RTC API ด้วย Microsoft Real-Time Communications server, และคุณสามารถสร้าง sophisticated, effective corporate communication tools ที่จะเพิ่ม cross-site team productivity.

Communication applications ที่พัฒนาโดยใช้ RTC API และ run บน Pentium 4 Processor ด้วย Hyper-Threading Technology' และ Windows XP ที่เป็น SPI จะได้ higher level ของ peer communications power ด้วยการตอบรับที่ดียิ่งขึ้นจนกระทั่ง performing multiple tasks concurrently



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4 การออกแบบโครงงานและพัฒนาโครงงาน

### 4.1 กำหนด requirement

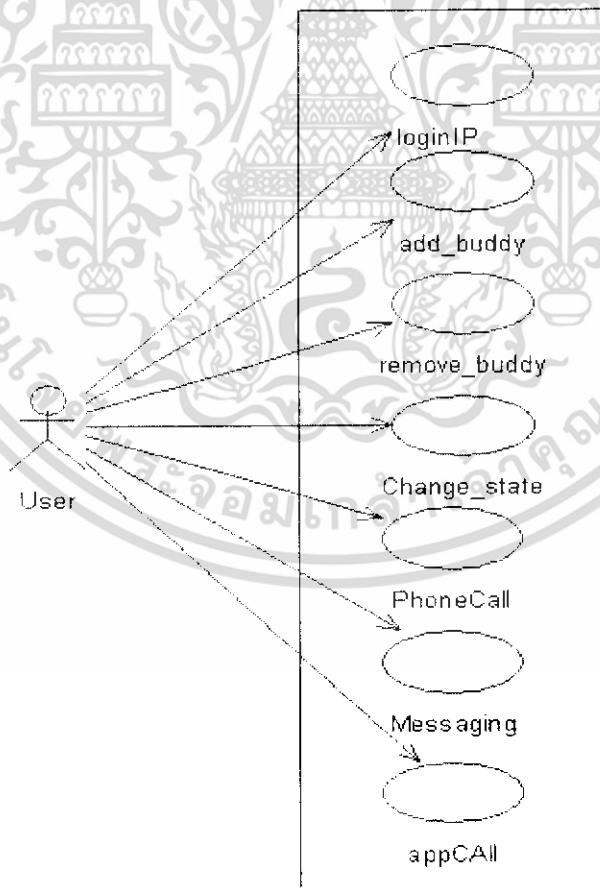
#### 4.1.1 function requirement

- 1.ผู้ใช้สามารถลือคอินผ่าน ไอพี ได้
- 2.สามารถ add buddy list ได้
- 3.สามารถ remove buddy list ได้
- 4.สามารถ เปลี่ยนสถานะ ของผู้ใช้ได้
- 5.เมื่อลือคอินแล้วผู้ใช้สามารถติดต่อกันแบบ text message ได้
- 6.เมื่อลือคอินแล้วผู้ใช้สามารถติดต่อกันแบบ video/audio ได้
- 7.สามารถที่จะทำการ sharing application ได้

#### 4.1.2.nonfunction requirement

- 1.สามารถ login จากที่ใดก็ได้ที่มีการเชื่อมต่ออินเทอร์เน็ต
- 2.ใช้โปรโตคอล SIP server

#### 4.1.3 Use Case Diagram



รูปที่ 4.1.1 Use Case Diagram แสดงรายการความต้องการทั้งหมดที่เป็นแบบฟังก์ชันการใช้งาน

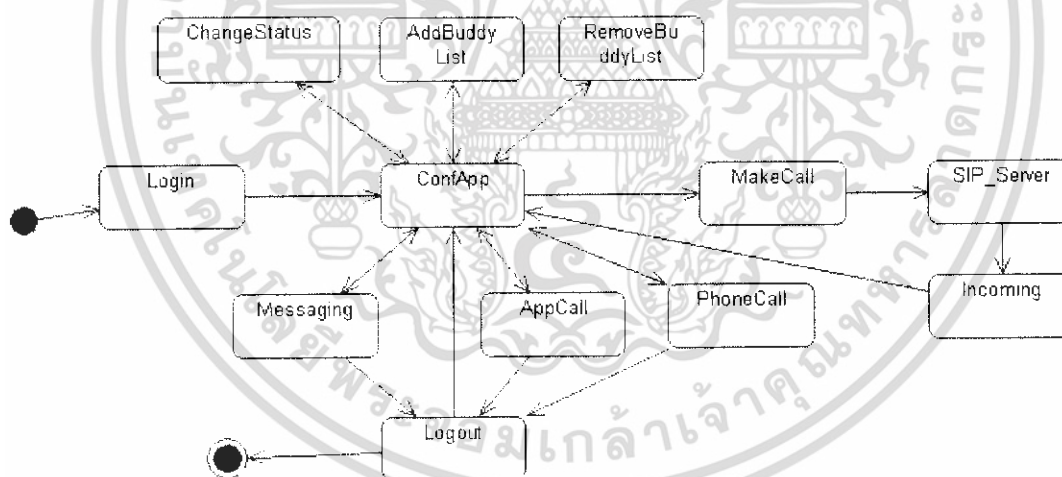
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Use Caser	Actor	Short Description
1.loginIP	user	ตอน login สามารถใช้ user name เป็นไอพีได้
2.add buddy list	user	สามารถเพิ่ม บุคคลที่จะสนทนาได้ต่อเนื่อง
3.remove buddy list	user	สามารถลบรายการบุคคลที่สนทนาออกไปได้
4.Change status	user	สามารถเปลี่ยนสถานะ เพื่อแสดงว่าตอนนี้เรา online,off line,busy
5.PhoneCall	user	สามารถติดต่อกันแบบ audio/vedio
6.Messaging	User	สามารถติดต่อกันด้วยข้อความ
7.appCall	user	สามารถ share application ได้

ตารางที่3 แสดงรายละเอียดของ use case

## 4.2 การวิเคราะห์

### 4.2.1. วิเคราะห์ระบบในส่วนที่เป็น se Case Diagram แบบละเอียด



รูปที่ 4.2.1 แสดง State Diagrame ในส่วนการติดต่อของ Use Case Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 วิเคราะห์ Use Case

จากการทำงานของ User Diagram มีการแสดงคลาสต่างๆดังนี้

คลาส Login เพื่อรับค่า เป็นเบอร์โทรศัพท์เพื่อส่งให้ ConfApp อีกที

คลาส MakeCallDialog เพื่อใช้ทำการเชื่อมต่อไปยัง SIPServer

คลาส ConfDialog เพื่อใช้ในการสร้างการเชื่อมต่อไปยัง RTC client ยกเลิกการเชื่อมต่อ ควบคุม อุปกรณ์ รับค่าพารามิเตอร์ต่างๆของผู้ใช้ และเป็นตัวที่จะเลือกว่าเป็นการติดต่อแบบใด

คลาส IncomingDialog เพื่อใช้ทำหน้าที่ในการรับ ตอรับ สายเรียกเข้าของ client

คลาส ChangStatus เพื่อใช้ในการเปลี่ยนสถานะผู้ใช้งานเป็น online,busy,offline

เป็นต้น

คลาส AddBuddylist เพื่อเพิ่มรายชื่อในการเชื่อมต่อ

คลาส RemoveBuddyList เพื่อลบรายชื่อการเชื่อมต่อออกจากรายการ

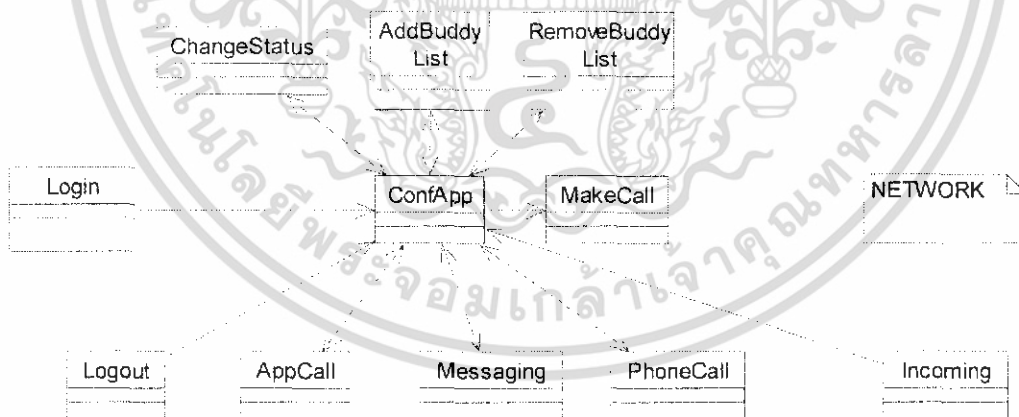
คลาส Messaging เพื่อใช้ในการเชื่อมต่อแบบข้อความ

คลาส AppCall เพื่อใช้จัดการการเชื่อมต่อที่เป็นการ Share application

คลาส PhoneCall เพื่อการสื่อสารแบบ audio/vedio data

คลาส Logout เพื่อออกจากการเชื่อมต่อ

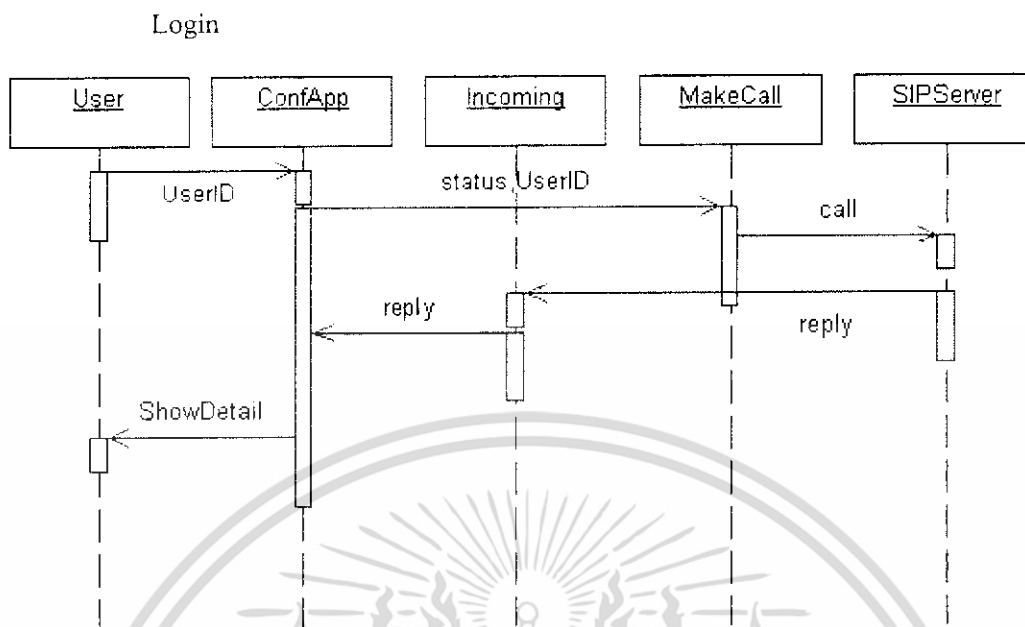
class diagram



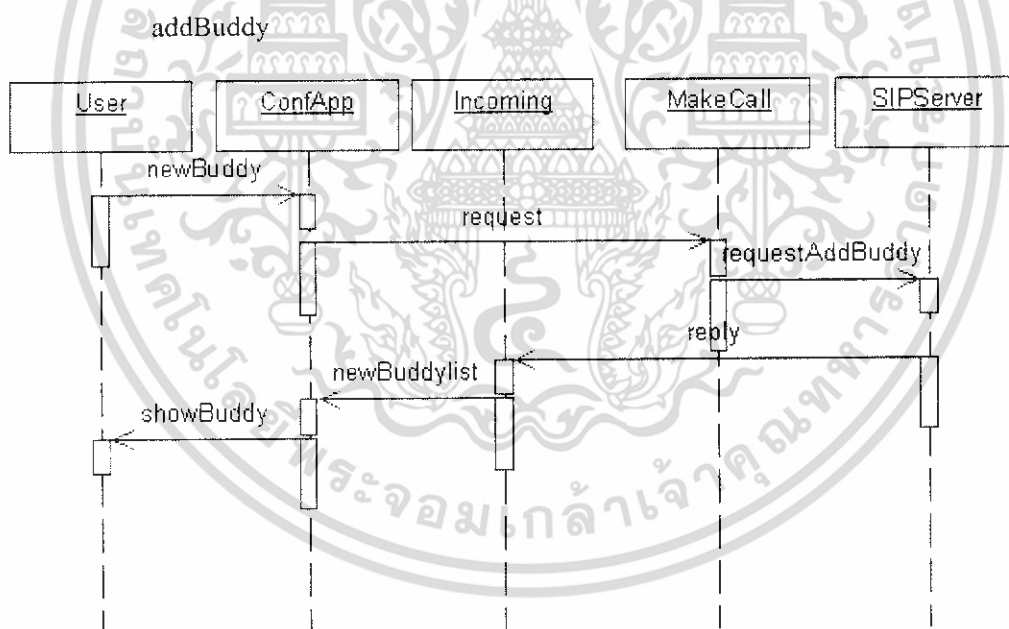
รูปที่ 4.2.2 แสดง Class Diagram เพื่อกำหนดคลาสที่ต้องสร้างเพื่อการใช้งานคร่าวๆว่ามีอะไรบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2.3 sequence diagram

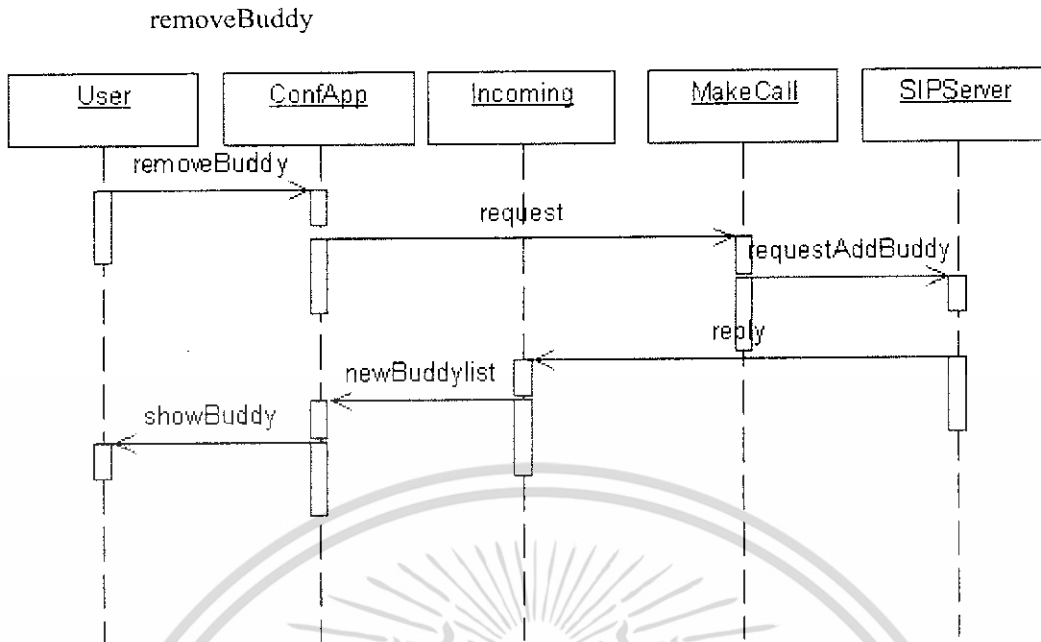


รูปที่ 4.2.3 แสดงลำดับขั้นตอนการทำงานของกร Login

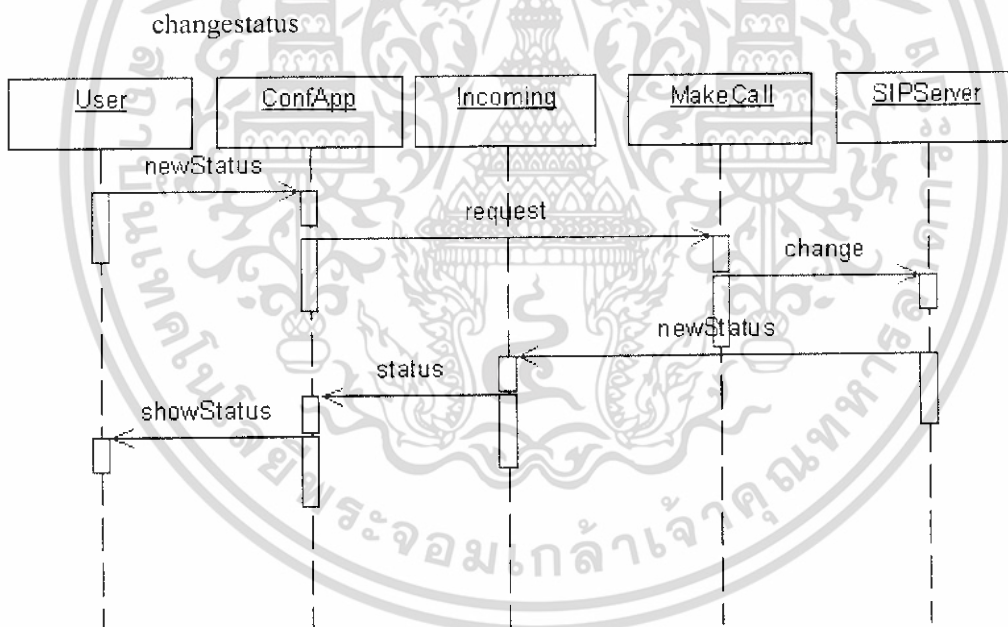


รูปที่ 4.2.4 แสดงลำดับการทำงานของกร addBuddy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

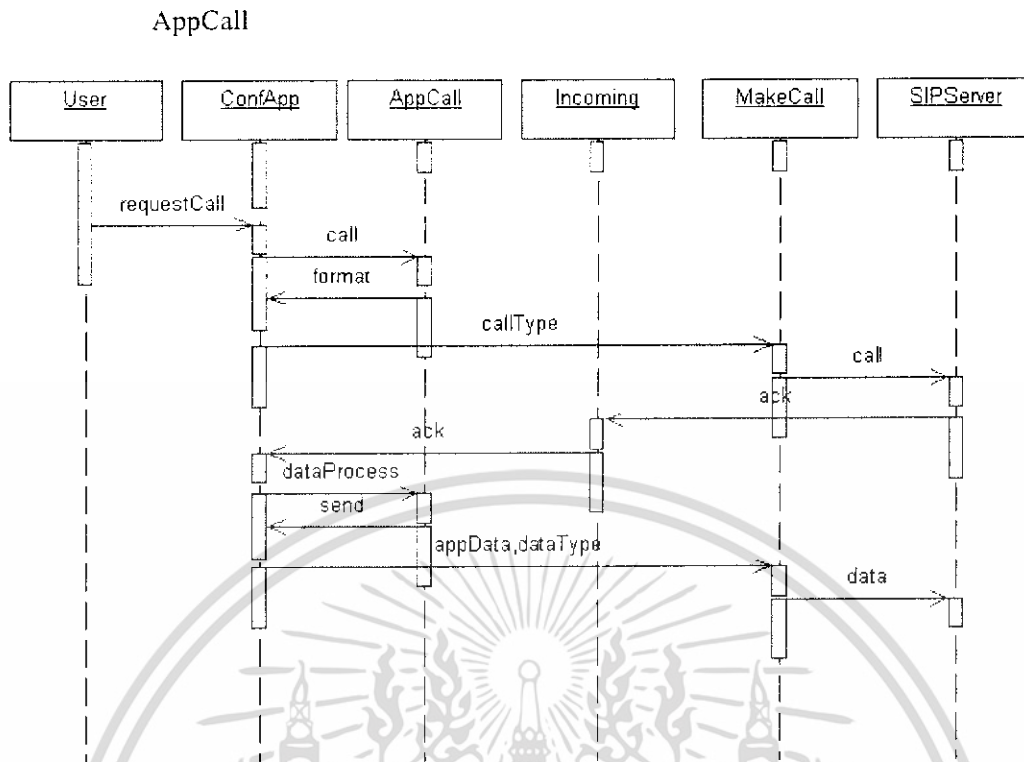


รูปที่ 4.2.5 แสดงลำดับขั้นตอนการทำงานของการทำงานของการ `removeBuddy`

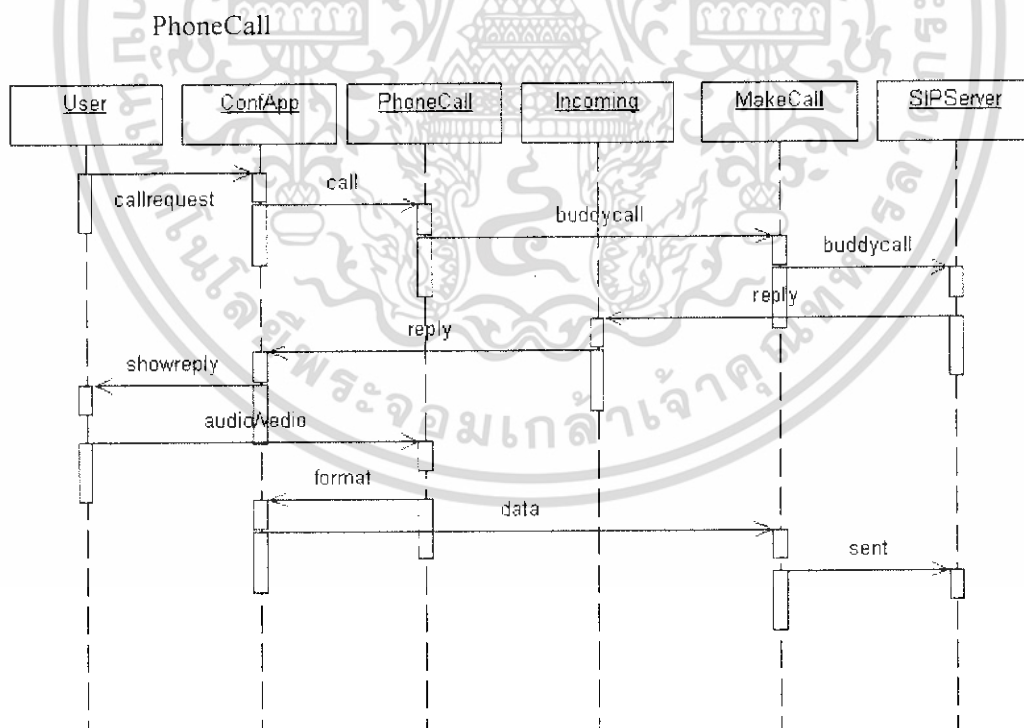


รูปที่ 4.2.6 แสดงลำดับขั้นตอนการทำงานของการทำงานของการเปลี่ยนสถานะของ user

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



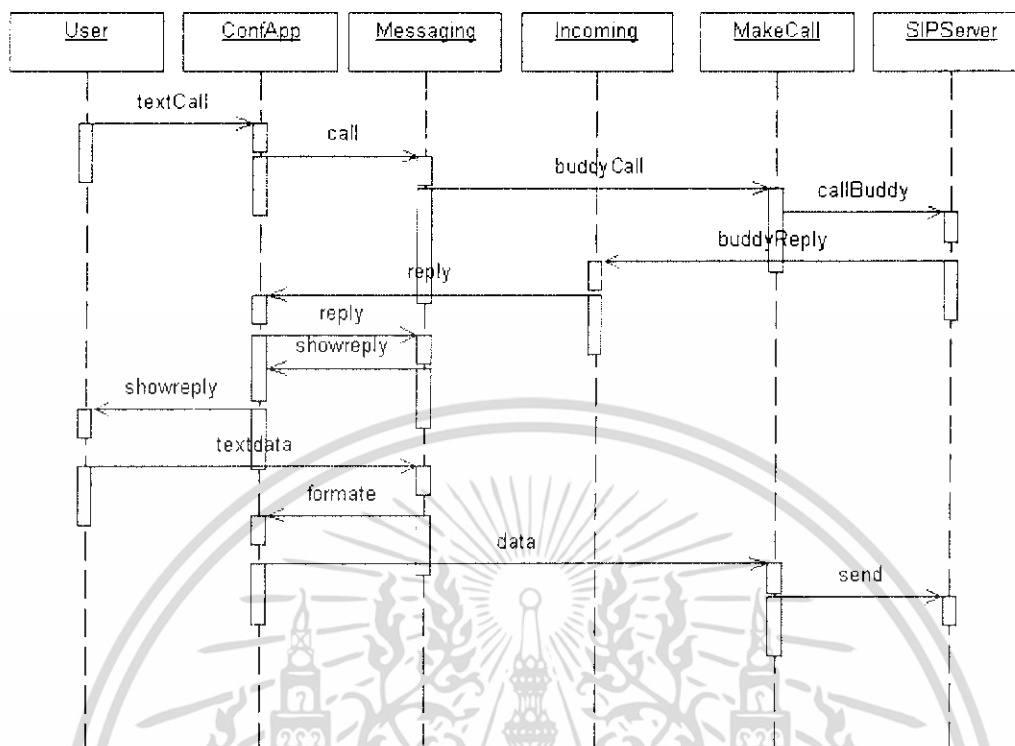
รูปที่ 4.2.7 แสดงลำดับขั้นตอนการทำงานของ การติดต่อแบบ AppCall



รูปที่ 4.2.8 แสดงลำดับขั้นตอนการทำงานของ การติดต่อแบบ PhoneCall

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Messaging



รูปที่ 4.2.9 แสดงลำดับขั้นตอนการทำงานของ การติดต่อแบบ Messaging

4.2.4 วางแผนการสร้าง

จะมีอยู่สองส่วนด้วยกันเนื่องจากตอนที่แล้วเราทำการกำหนด ความต้องการและศึกษา โพรโตคอลและรูปแบบที่ต้องใช้มาบ้างแล้ว ในส่วนนี้จึงเน้นการสร้างและพัฒนาโปรแกรมเป็น หลัก ซึ่งมีอยู่ ส่วนหลักๆดังนี้

ลำดับการ ทำงาน	รายการดำเนินงาน	วันเริ่มต้น	กำหนด	หมายเหตุ
1	การสร้าง application สำหรับใช้งาน	20/10/48	30/11/48	
2	การ setup SIP server เพื่อใช้งาน	01/12/48	01/01/49	
3	การติดตั้งและประกอบส่วนของ component เพื่อ ใช้งาน	02/01/49	08/01/49	
4	การทดสอบ	10/01/49	15/01/49	
5	การแก้ไขเพิ่มเติม	16/01/49	15/02/49	
6	การจัดทำเอกสาร	16/01/49	20/02/49	

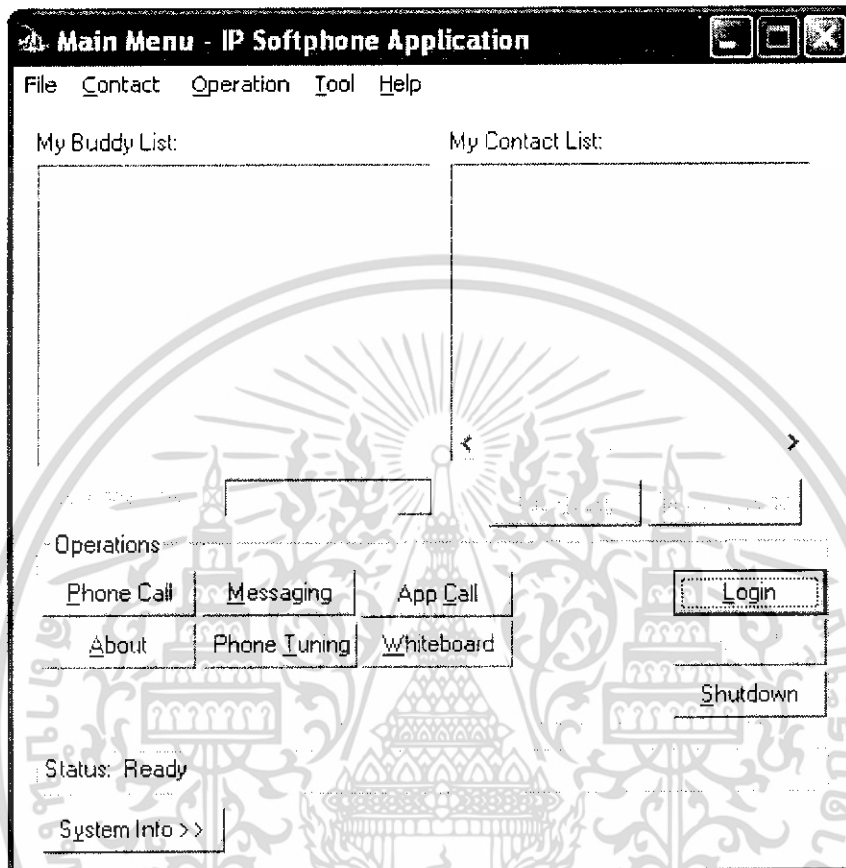
ตารางที่ 4 การวางแผนดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.3 การออกแบบ

### 4.3.1 User Interface

สำหรับในส่วนของการติดต่อผู้ใช้ ซึ่งเมื่อเปิดโปรแกรมขึ้นมาจะก็จะได้ หน้าต่างดังรูป ซึ่งประกอบด้วย menu และส่วนต่างๆดังนี้



รูปที่ 4.3.1 แสดงหน้าเริ่มต้นของโปรแกรม

Menu File จะประกอบด้วยเมนูย่อย 3 เมนูคือ

Login จะไว้สำหรับส่วน login เพื่อเข้าใช้งาน

Logoff สำหรับออกจากการเชื่อมต่อ

Shutdown เพื่อไว้สำหรับปิดโปรแกรม

Menu Contact จะประกอบด้วย

Add buddy สำหรับเพิ่มรายชื่อผู้ที่เราต้องการติดต่อด้วยเข้ามายังรายการ

Remove buddy สำหรับลบรายชื่อที่เราไม่ต้องการติดต่อออกจากรายชื่อ

Menu Operation

Phone call ไว้สำหรับการติดต่อแบบ audio/vedio

Messaging ไว้สำหรับการติดต่อแบบข้อความ

AppCall ไว้สำหรับการ share application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Whiteboard เปรียบเสมือนกระดานที่สามารถเขียนหรือวาดอะไรลงไปก็ได้

Menu Tool มีเมนูย่อยเพียงตัวเดียวคือ Device Tuning Help ไว้สำหรับติดตั้งหรือเชื่อมต่ออุปกรณ์ เช่น ไมโครโฟน หรือ ลำโพง

Menu Help ไว้สำหรับช่วยเหลือ เป็นคู่มือให้ผู้ใช้

ส่วนของการแสดงผลนั้นประกอบด้วยสองส่วนคือ My buddy list และ My contact list

My buddy list จะไว้สำหรับแสดงรายชื่อของ buddy list

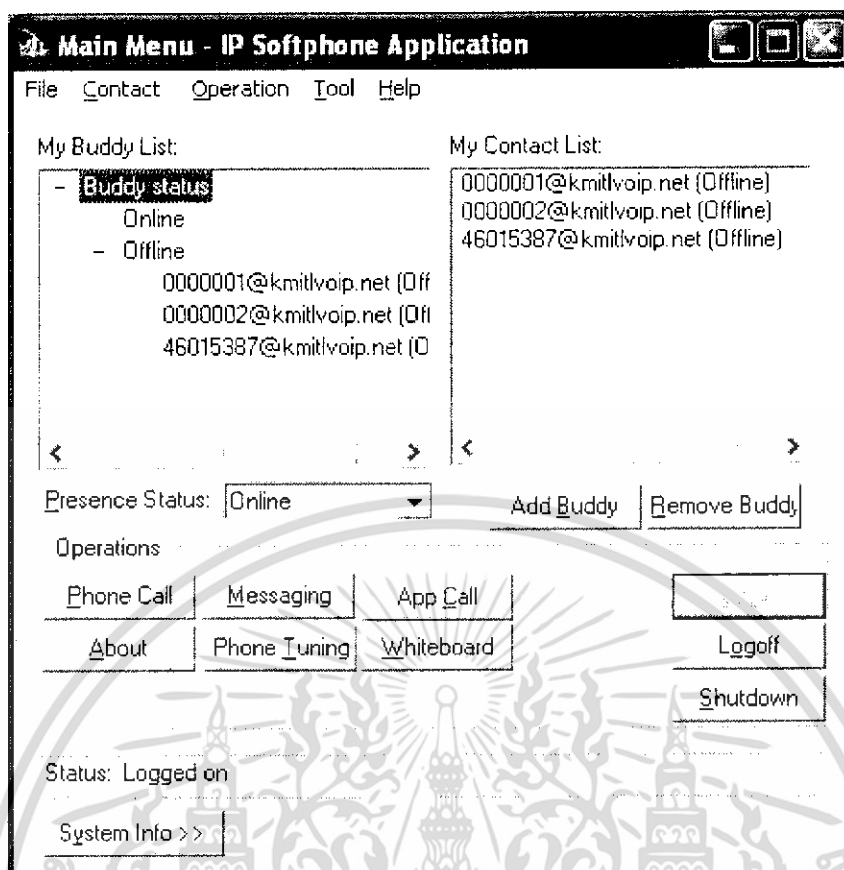
My contact list แสดงรายชื่อ buddy list และไว้สำหรับเลือกเมื่อต้องการติดต่อ

ส่วนของปุ่มใช้งานนั้นก็เหมือนเมนูข้างต้นที่กล่าวมาแล้ว เพียงแต่เปลี่ยนจากการเลือกเมนู ก็มาใช้ปุ่มกดแทน ซึ่งตัวแรกที่จะกล่าวถึงคือ login ซึ่งจะเป็นตัวแรกที่เราต้องทำเพื่อจะเป็นการเริ่มต้นสู่การสื่อสารด้วยรูปแบบต่างๆ และคนที่มียาชื่อจะได้รู้ว่าเราอยู่ในระบบหากเราพร้อมจะติดต่อ

Field	Value
URI	46015384@kmitlvoip.net
Server IP	161 . 246 . 5 . 129
Transport	TCP

รูปที่ 4.3.2 แสดงหน้าของการใส่ Login

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



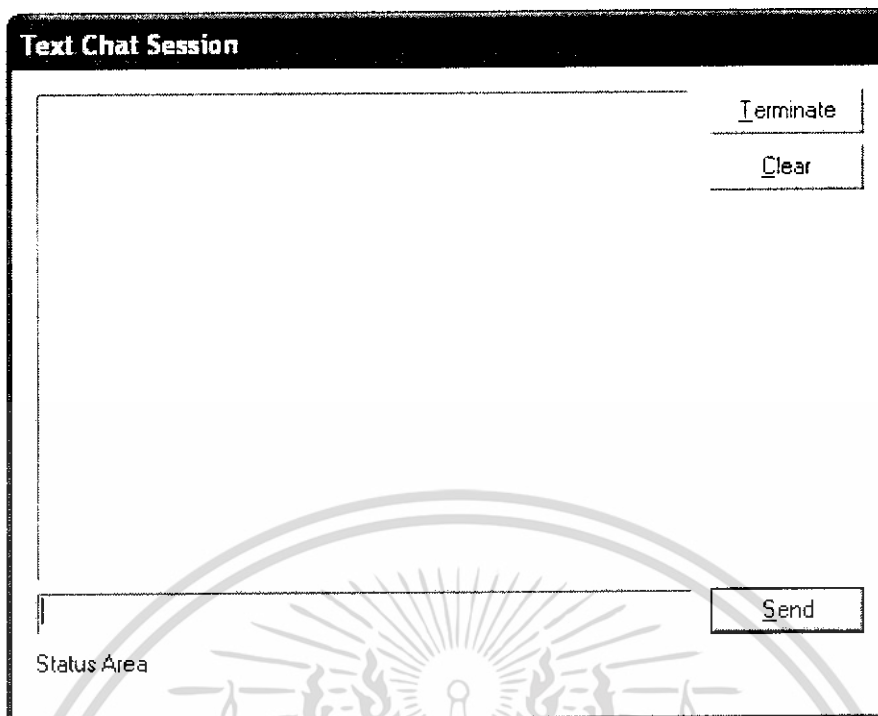
รูปที่ 4.3.3 แสดงหน้าเริ่มต้นของโปรแกรมเมื่อทำการ Login เสร็จแล้ว

ซึ่งภาพด้านล่างนี้จะแสดงสองส่วนเมื่อเราทำการ ล็อกอินเข้ามา นั่นก็คือ My Buddy list ซึ่งแสดง ส่วนของรายชื่อทั้งหมดที่มี และส่วนที่สองคือ My contact list ซึ่งจะแสดงเฉพาะรายชื่อของเรา

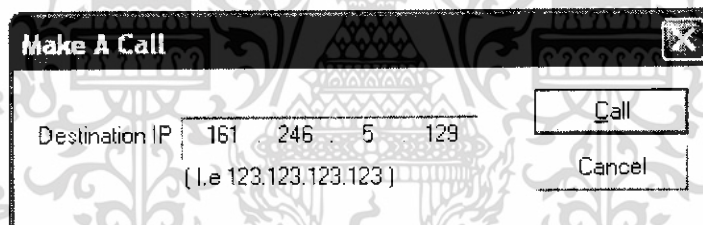


รูปที่ 4.3.4 โดอะล็อกเมื่อต้องการ Add Buddy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

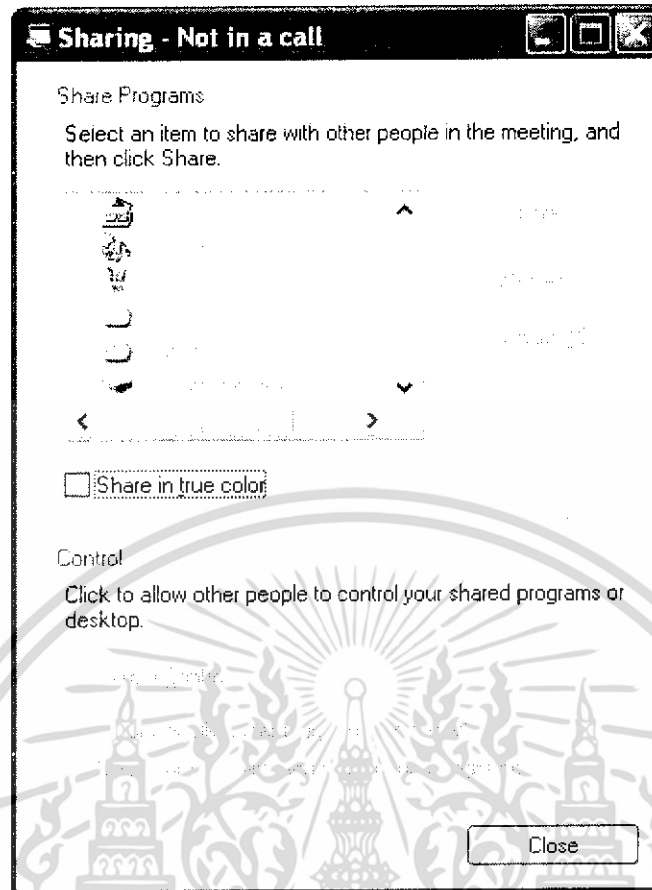


รูปที่ 4.3.5 แสดงหน้าของการติดต่อแบบ Messaging

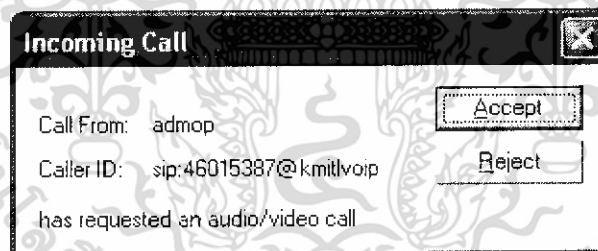


รูปที่ 4.3.6 แสดงไดอะล็อกเพื่อระบุปลายทางเพื่อทำการติดต่อแบบไอพีแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

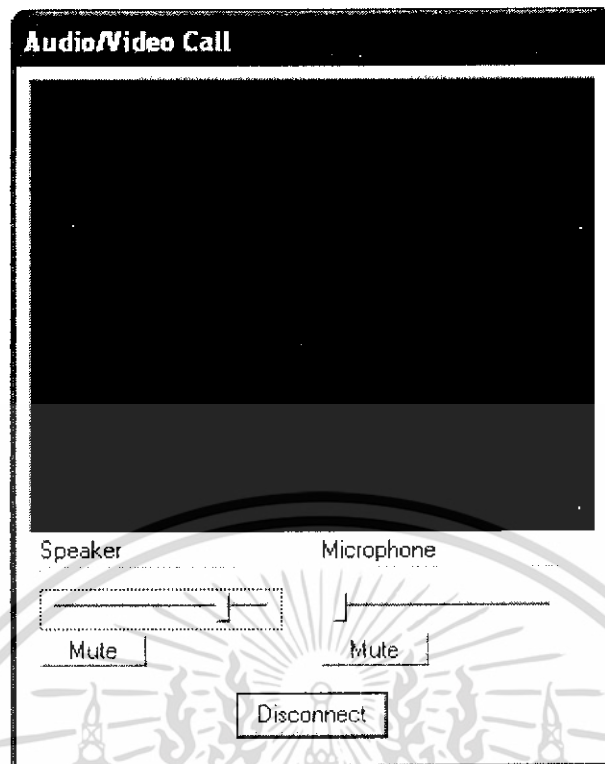


รูปที่ 4.3.7 แสดงหน้าต่างให้เลือกเมื่อกดปุ่มการติดต่อแบบ AppCall

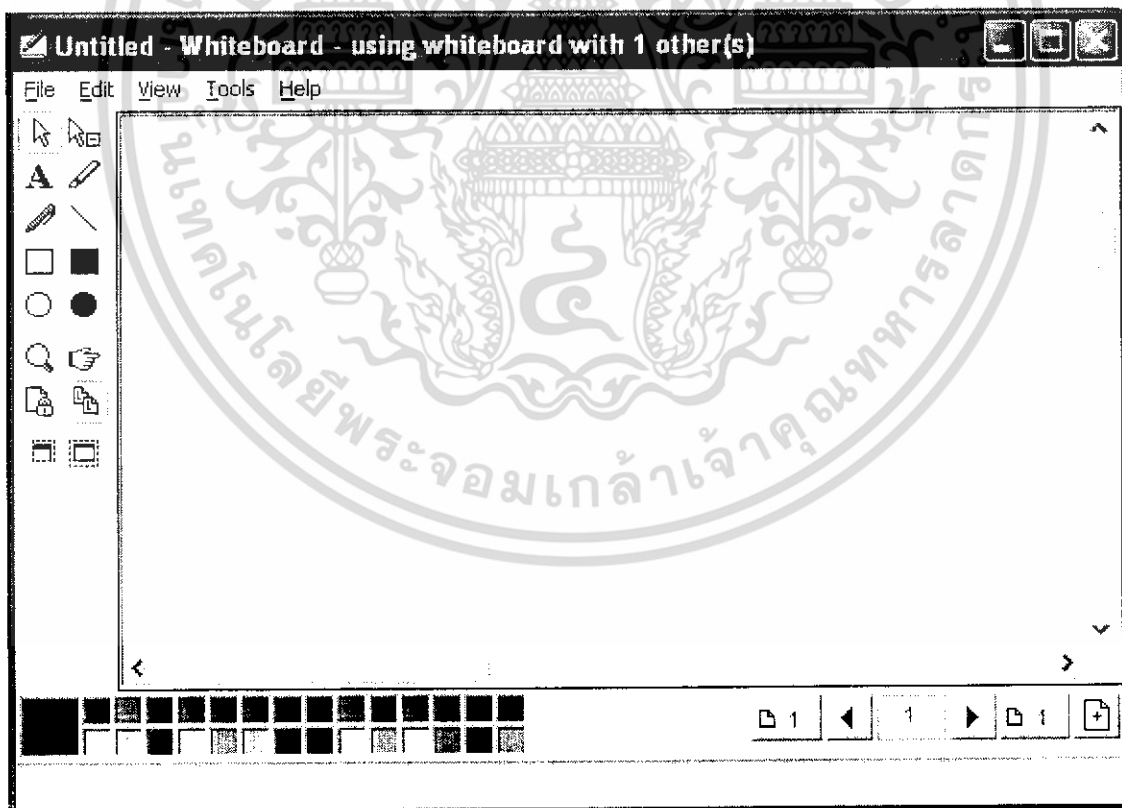


รูปที่ 4.3.8 แสดง Message Box เมื่อมีผู้ต้องการติดต่อเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

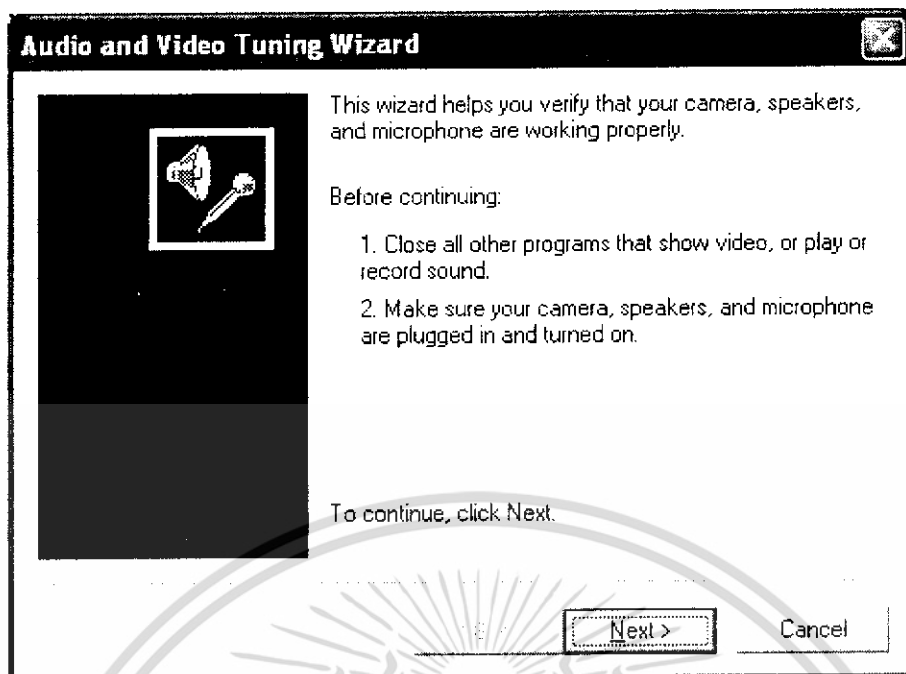


รูปที่ 4.3.9 แสดงรูปแบบการติดต่อแบบ PhoneCall



รูปที่ 4.3.10 แสดงการติดต่อแบบ whiteboard

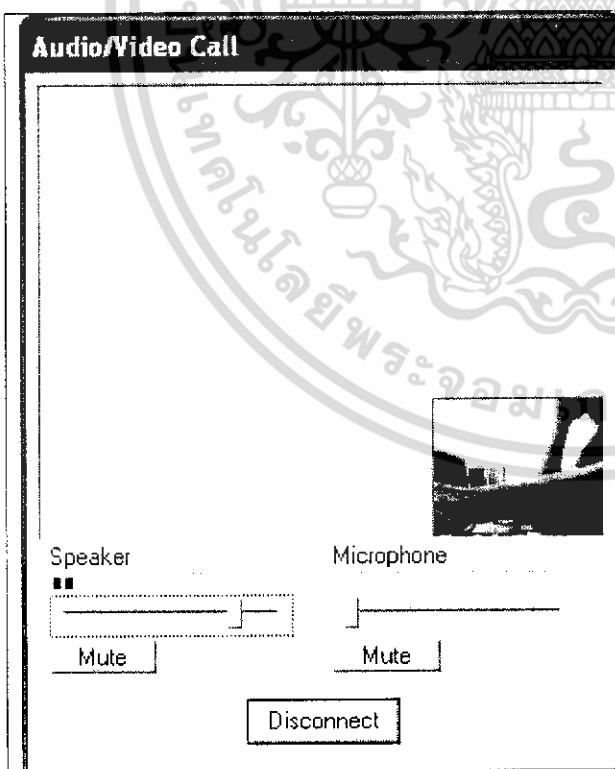
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



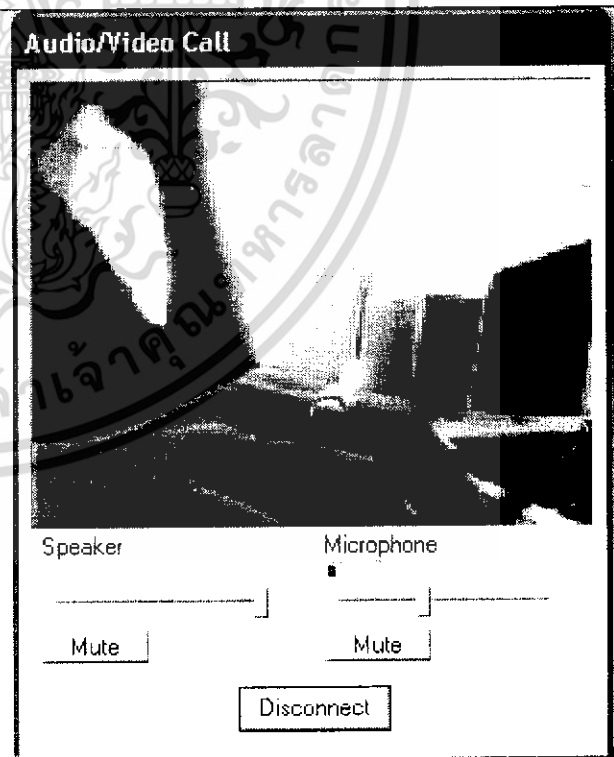
รูปที่ 4.3.11 แสดงไดอะล็อกของการทำ PhoneTuning

#### 4.4 การทดลองและผลการทดลอง

จากภาพด้านล่างจะเป็นการติดต่อระหว่างฝั่งผู้ส่งและผู้รับ เหมือนการคุยโทรศัพท์แต่จะมีภาพของทั้งสองฝั่งแสดงเท่านั้นเอง



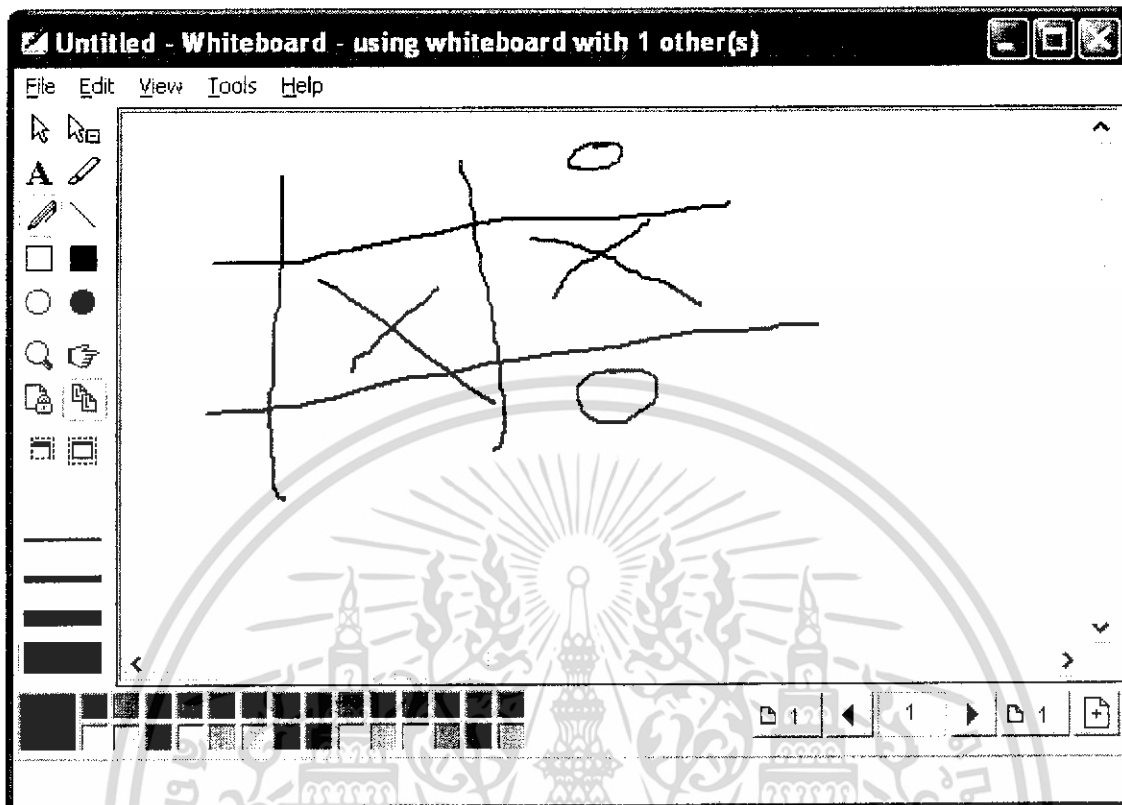
รูปที่ 4.4.1a แสดงการทดลอง PhoneCall ฝั่งผู้ส่ง



รูปที่ 4.4.1b แสดงการทดลอง PhoneCall ฝั่งผู้รับ

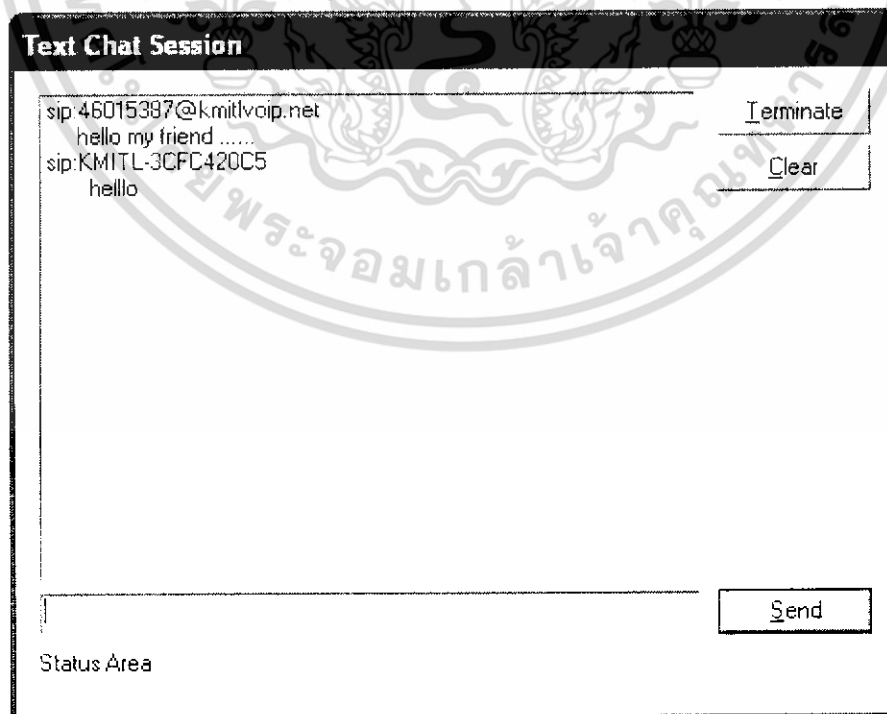
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพด้านล่างจะเป็นการติดต่อกันแบบ WhiteBoard ซึ่งจะเป็นการใช้ Application ร่วมกัน  
ดังแสดงในภาพ



รูปที่ 4.4.2 แสดงการทดลอง Whiteboard

ส่วนการติดต่ออีกแบบคือแบบ Messaging คือการถามตอบกันด้วยข้อความนั่นเอง



รูปที่ 4.4.3 แสดงการทดลอง Messaging

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## บทที่ 5 บทวิจารณ์และสรุป

### 5.1. บทสรุป

ปัจจุบันการติดต่อสื่อสารกระทำกันบนอินเทอร์เน็ตมากขึ้น เนื่องจากสะดวก รวดเร็วและประหยัด แต่การจะสร้างส่วนของการติดต่อได้นั้นต้องมีความรู้เรื่องโปรโตคอลเพื่อใช้กำหนดแนวทางในการเชื่อมต่อเพื่อให้ไปในแนวทางเดียวกัน โปรโตคอลซีฟซึ่งอยู่ในระดับแอปพลิเคชันนั้นก็เป็นอีกตัวที่สร้างขึ้นมาเพื่อใช้งานสำหรับ VoIP โดยเฉพาะ แต่ก็ยังอาศัยโปรโตคอลตัวอื่นๆ ด้วยเช่น TCP ,RTP เป็นต้น

### 5.2. วิจารณ์สิ่งที่ได้จากโครงการ

สิ่งที่ได้จากการทำโครงการนี้ นอกจากตัวโปรแกรม และความรู้ความเข้าใจในโปรโตคอลที่เกี่ยวข้องกับ VoIP แล้วยังมี API ต่างๆ ที่ช่วยในการสร้างแอปพลิเคชันอีกด้วย ซึ่งอย่างหลังสำคัญมากเพราะหากมองแค่โปรแกรมเราอาจหาความรู้ได้จากที่ไหนก็ได้และใช้งานได้ดีกว่าด้วย แต่สิ่งที่เราทำมาจะเป็นสิ่งที่ทำให้เราสร้าง และนำไปประยุกต์ใช้งานสำหรับส่วนอื่นๆ ได้อีกต่อไป

### 5.3. ปัญหาอุปสรรคและแนวทางแก้ไข

ปัญหาที่สำคัญคือเราไม่สามารถที่จะหา เนื้อหาที่เกี่ยวข้องได้ง่ายและหลากหลายมากนัก เนื่องจากข้อจำกัดด้านภาษา และอีกอย่างหนึ่งคือ การวางแผนการทำงานไม่ค่อยดีนักในเรื่องเวลา เพราะเมื่อทำไปปัญหาและความต้องการบางอย่างอาจเกิดขึ้นได้ เราจึงจำเป็นต้องวางแผนเพื่อเวลาส่วนนี้ไว้ด้วย เพื่อที่ต่อไปหากเกิดปัญหาที่ไม่คาดคิดเราจะได้มีเวลาในการแก้ไข

### 5.4 แนวทางการพัฒนาต่อ

สิ่งสำคัญที่ใช้ในการพัฒนาที่สำคัญคือ API ซึ่งเราจำเป็นต้องมีความเข้าใจที่ตีพอสมควรจึงจะสามารถสร้างได้ และที่สำคัญส่วนที่ใช้ในการทำ เซิร์ฟเวอร์นั้นส่วนมากจะเป็นลินุกซ์เสียส่วนนี้อาจเป็นปัญหา เพราะความไม่คุ้นเคยเราควรศึกษาส่วนนี้ไว้เพื่อจะได้พัฒนาและนำไปใช้งานได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] บทความเกี่ยวกับ VoIP. “การสื่อสารด้วยระบบ Voice-over-IP (VoIP).” [Online]. Available :[http://www.voiphailand.com/voip/voip\\_providers.php](http://www.voiphailand.com/voip/voip_providers.php)
- [2] บทความน่าสนใจ. “เทคโนโลยี VoIP ภาคที่1,2.” [Online]. Available :<http://www.marketathome.com>
- [3] VoIP เทคโนโลยีสื่อสารไร้พรมแดน. “ตอนความปลอดภัยบนโครงข่าย SIP.” ศศ.ดร.บงการ หอมนาน.[Online]. Available :<http://micro.se-ed.com>
- [4] Engineer Forums. “Voice over IP (VoIP) Technology.” [Online]. Available :<http://eng.bu.ac.th/forum>
- [5] DCOM IT FOCUS. “Voice over IP (VoIP) Technology.” (ตอนที่ 1,2). [Online]. Available :[http://www.dcomputer.com/proinfo/support/TipTrick/techno\\_VoIP02.asp](http://www.dcomputer.com/proinfo/support/TipTrick/techno_VoIP02.asp)
- [6] Business and Industry Development. “วอยซ์ โอเวอร์ไอพี (VoIP).” [Online]. Available :<http://www.nectec.or.th>
- [7] สารนารัฐประจำสัปดาห์. “VoIP การสื่อสารทางเสียงผ่านเครือข่าย IP.” [Online]. Available :[http://www.ku.ac.th/magazine\\_online/index.html](http://www.ku.ac.th/magazine_online/index.html)
- [8] Enhancing Rich Client Communications with the Microsoft\* Real-Time Communications API.” Jim Huang Sr. Technical Marketing Engineer Intel Corporation” [Online]. Available : <http://www.intel.com/cd/ids/developer/asmo-na/eng/newsletter/43672.htm>
- [9] สาธิตพงศ์ พุทธิประเสริฐ,สินชัย กมลภิวงศ, ดัชนีกร วุฒิสัทติกุลกิจ “โปรโตคอลมาตรฐานสำหรับอินเทอร์เน็ตเทเลโฟนนี่” วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย
- [10] จารุณี มีเงิน ,จิรายุ เจริญวิริยะภาพ “การพัฒนาการติดต่อระหว่างคอมพิวเตอร์กับโทรศัพท์โดย DirectX และ TAPI ” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2546
- [11] ธนภัทร์ ลิขิตพัฒน์,อดิสร สิริสมบุญณ์ “โปรแกรมสนทนาด้วยมาตรฐานซิปพบนซิมเบียนสมาร์ตโฟน” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้