

โปรแกรมการสื่อสารด้วยเสียงบนพีซีและพีดีเอ

Voice Instant Messaging on PC and PDA



นาย กฤตธี

ศิริสุนทร

นาย กิตติรัช

ศรีอรทัยวรรณ

นางสาว ชฎาทิพย์

กิตติสาร

เลขหมู่.....
เลขทะเบียน **61761**
วัน,เดือน,ปี **2 1 ก.ค. 2549**

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการสื่อสารด้วยเสียงบนพีซีและพีดีเอ

Voice Instant Messaging on PC and PDA



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมการสื่อสารบนพีซี

VOICE INSTANT MESSAGING ON PC

ผู้จัดทำ

- | | | |
|-----------------------------|--------------|----------|
| 1. นายกฤตธี ศิริสุนทร | รหัสประจำตัว | 44010009 |
| 2. นายกิตติรัช ศรีอรรถยวรรณ | รหัสประจำตัว | 44010025 |
| 3. นางสาวชฎาทิพย์ กิตติสาร | รหัสประจำตัว | 44010084 |



อาจารย์ที่ปรึกษา

(ดร.วรวัฒน์ ลิ้มโกศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการสื่อสารด้วยเสียง

นายกฤตธี ศิริสุนทร 44010009
 นายกิตติรัช ศรีอรทัยวรรณ 44010025
 นางสาวภาทิพย์ กิตติสาร 44010084
 ดร.วรวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษา
 ปีการศึกษา 2547

บทคัดย่อ

เมื่ออินเทอร์เน็ตมีการใช้งานอย่างกว้างขวางขึ้นความต้องการประยุกต์ใช้สิ่งใหม่ ๆ บนอินเทอร์เน็ตจึงได้รับการพัฒนา โดยเฉพาะอย่างยิ่งการใช้อินเทอร์เน็ตเป็นโครงสร้างพื้นฐานเพื่อรองรับการสื่อสารรูปแบบต่างๆ เช่น การติดต่อด้วยเสียง การกระจายสัญญาณเสียงหรือภาพบนเครือข่าย

การติดต่อสื่อสารแบบเรียลไทม์กลายเป็นส่วนสำคัญของธุรกิจในปัจจุบัน มีการประยุกต์ใช้ไอพีซึ่งเป็นรูปแบบในการสื่อสารข้อมูลในรูปแบบแพ็คเกจผ่านเครือข่าย เพื่อการลดค่าโทรศัพท์และค่าโทรสาร และยังเป็นการสร้างฐานสำหรับการใช้มัลติมีเดียในระดับที่สูงขึ้นเพื่อความมีประสิทธิภาพในการสื่อสารภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Voice Instant Messaging

Kritatee Sirisoonthorn 44010009

Kittithouch Sriortaiwan 44010025

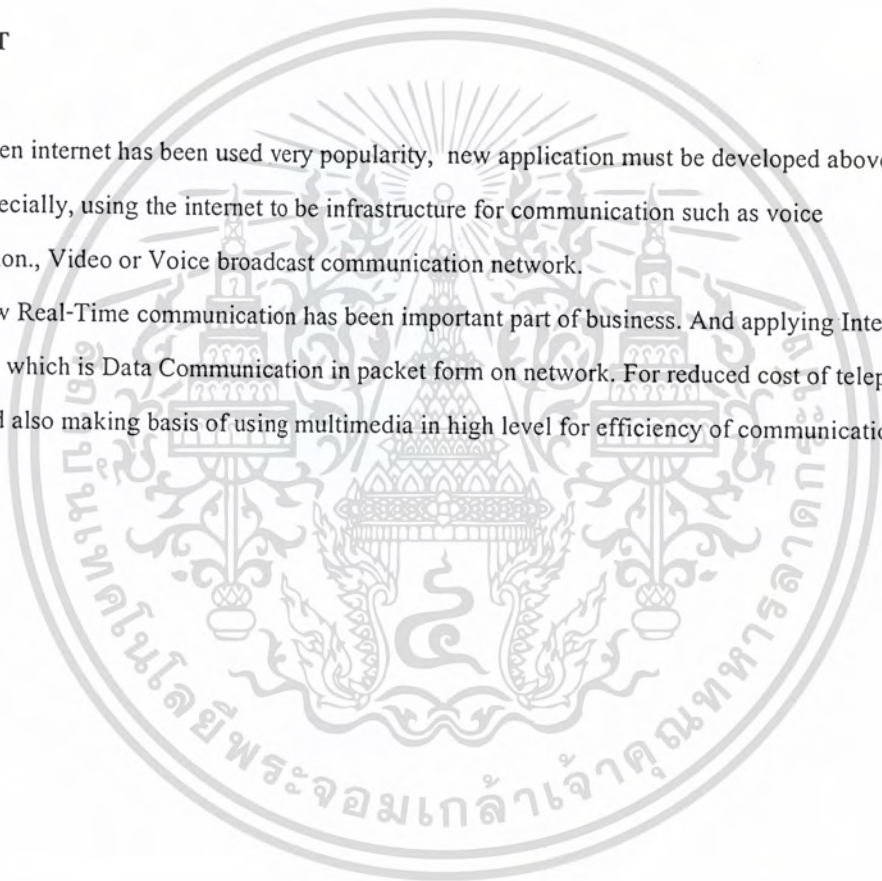
Chadatip Kitisarn 44010084

Dr. Worawat Limphokha Advisor

ABSTRACT

When internet has been used very popularity, new application must be developed above internet. Especially, using the internet to be infrastructure for communication such as voice communication., Video or Voice broadcast communication network.

Now Real-Time communication has been important part of business. And applying Internet Protocol (IP) which is Data Communication in packet form on network. For reduced cost of telephone and Fax. And also making basis of using multimedia in high level for efficiency of communication in organization.



กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงจะไม่สำเร็จได้ด้วยดี หากไม่รับคำแนะนำและคำปรึกษาจากบุคคลเหล่านี้ ซึ่งบุคคลแรกที่มีความสำคัญที่สุดก็คือ ดร.วรวัฒน์ ลิ้มโกศา ซึ่งเป็นอาจารย์ที่ปรึกษาที่ให้ความสนใจ คอยแนะนำเมื่อเกิดปัญหาในขณะทำวิทยานิพนธ์ ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

บุคคลต่อไปก็คือ คุณ วุฒินัย กาญจนสร ขอขอบคุณที่ให้คำปรึกษาเกี่ยวข้องกับส่วนหนึ่งของวิทยานิพนธ์

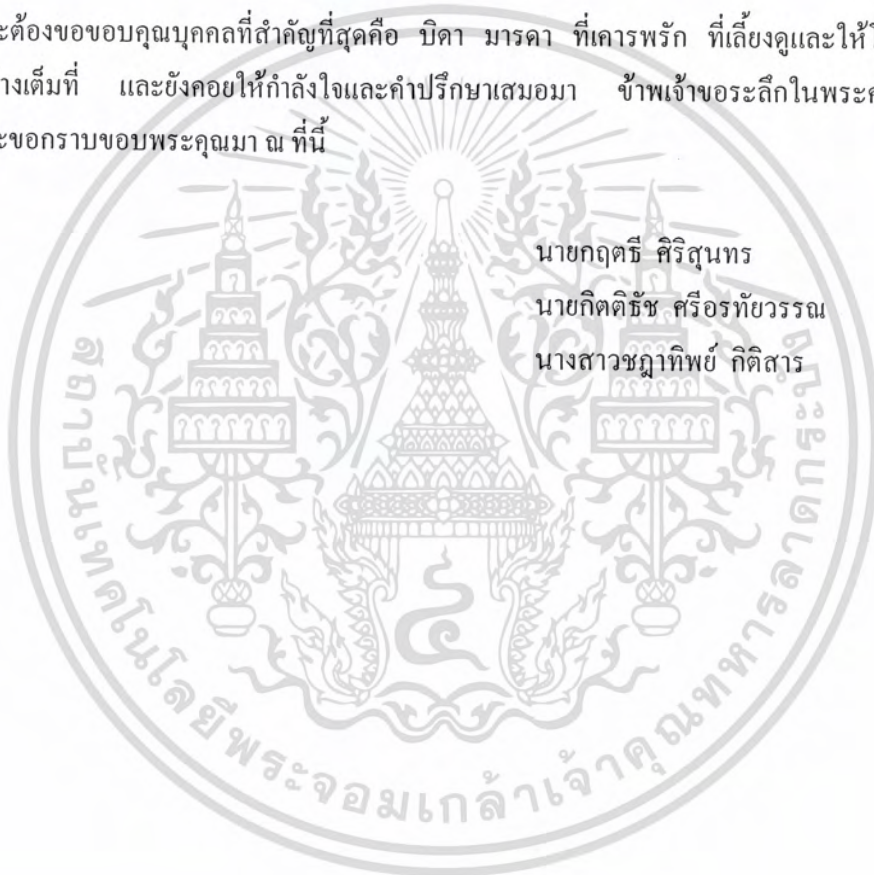
ขอขอบคุณ บริษัท คิจิตอล แอ็คเซ็ส จำกัด ที่ให้คำปรึกษาเกี่ยวกับอุปกรณ์ที่เกี่ยวข้องกับวอยซ์ โอเวอร์ไอพี

และต้องขอขอบคุณบุคคลที่สำคัญที่สุดคือ บิดา มารดา ที่เคารพรัก ที่เลี้ยงดูและให้โอกาสในการศึกษาอย่างเต็มที่ และยังคอยให้กำลังใจและคำปรึกษาเสมอมา ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

นายกฤตธี ศิริสุนทร

นายกิตติรัช ศรีอรทัยวรรณ

นางสาวชฎาทิพย์ กิตติสาร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	IX
สารบัญภาพ	X
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 เป้าหมายและขอบเขตของงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
บทที่ 2 วีโอไอพี (VOIP)	3
2.1 ลักษณะของโครงข่ายคอนเวิร์จ	3
2.2 องค์ประกอบโครงข่ายวีโอไอพี	5
2.3 ผลลัพธ์ของวีโอไอพี	6
2.4 ปัจจัยที่มีผลต่อคุณภาพเสียง	7
2.4.1 Delay	7
2.4.2 จิตเตอร์ (jitter)	7
2.4.3 แพ็คเก็ต	8
2.5 รูปแบบการส่งสัญญาณเสียงบนเครือข่ายไอพี	8
2.5.1 PC-to-PC	8
2.5.2 PC-to-telephone	9
2.5.3 เทเลโฟนนี่ (Phone - Phone)	9
2.6 ขั้นตอนการทำงานของวีโอไอพี	10
2.7 องค์ประกอบของวีโอไอพี	10
2.7.1 Software Client	10
2.7.2 วีโอไอพี เกตเวย์	10
2.7.3 เกททีปเปอร์	10
บทที่ 3 ซิปโพรโตคอล (SIP Protocol)	12
3.1 บทนำ	12
3.2 SIP Network Elements	13
3.2.1 User Agent	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

3.2.2	พรีอ็อกซีเซิร์ฟเวอร์ (Proxy Server)	14
3.2.3	Registrar	14
3.2.4	Redirect เซิร์ฟเวอร์	15
3.3	การสร้างชีพเสชชั้นในโดเมนเดียวกัน	16
3.4	การสร้างชีพเสชชั้นที่ต่างโดเมนกัน	16
3.5	ชื่อและแอดเดรส (Addressing & Naming)	18
3.6	SIP Message	18
3.7	SIP Request	20
3.8	SIP Response	20
3.9	SIP Transaction	21
3.10	SIP Dialogs	22
3.11	Dialogs Facilitate Routing	23
3.12	Dialog Identifiers	24
3.13	Typical SIP Scenarios	25
3.13.1	Registration	25
3.13.2	Session Invitation	25
3.13.3	Session Termination	26
3.13.4	Record Routing	26
3.13.5	Instant Message	27
บทที่ 4	Real Time Communication API	28
4.1	Real-time Communication Client Application Programming Interface	28
4.2	RTC Version 1.2 Object and Interface	28
4.2.1	Client Object	29
4.2.2	BuddyGroup Object	30
4.2.3	User Search Query Object	30
4.2.4	Presence Device Object	30
4.2.5	Session Object	30
4.2.6	Participant Object	30
4.2.7	Buddy Object	30
4.2.8	Watcher Object	30
4.2.9	Profile Object	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

4.3 ความต้องการพื้นฐานของ RTC (RTC General Requirement)	31
4.3.1 แนวคิดพื้นฐาน	31
4.3.1.1 Session	31
4.3.1.2 ซิปเซิร์ฟเวอร์	31
4.3.1.3 Contact list	31
4.3.1.4 Roaming	31
4.3.2 ชนิดของการเชื่อมต่อที่ใช้โดย RTC Client API	31
4.3.3 Session information	32
4.3.4 ความต้องการพื้นฐาน	32
4.4 Server requirements	33
4.4.1 การติดต่อจากของเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์	33
4.4.2 การติดต่อจากระหว่างเครื่องคอมพิวเตอร์กับโทรศัพท์	33
4.4.3 การติดต่อของเครื่องโทรศัพท์กับเครื่องโทรศัพท์	34
4.4.4 Presence information	34
4.4.5 Profile Requirements	34
4.4.6 SIP และ URL โทรศัพท์	35
4.5 การพิสูจน์ตัวตน	36
4.5.1 Basic Authentication	36
4.5.2 Digest Authentication	36
4.5.3 NTLM Authentication	37
4.5.4 Kerberos Autentication	37
4.6 Transport Protocol	37
4.7 ข้อดีของ RTC Client API	38
4.7.1 การสร้างแอปพลิเคชันโดยการใช้ RTC Client API	38
4.7.2 Desktop Application	38
4.7.3 Scalable Application	38
บทที่ 5 J2ME	39
5.1 configuration	39
5.2 profile	40
5.3 KVM	41
5.4 สถาปัตยกรรมของ J2ME	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

บทที่ 6 SIP API สำหรับ J2ME	42
6.1 การใช้ SIP API สำหรับ J2ME	42
6.2 ฟังก์ชันของ SIP ที่สนับสนุน	43
6.3 system requirement	44
บทที่ 7 การออกแบบและพัฒนาโปรแกรม	45
7.1 หลักการ	45
7.1.1 หลักการ	45
7.1.2 ขอบเขตการทำงาน	45
7.1.3 ฟังก์ชันการทำงานต่าง ๆ	45
7.1.4 การล็อกอินเข้าสู่ระบบ	46
7.1.5 การพิสูจน์ตน	46
7.1.6 แสดงรายชื่อผู้ติดต่อและสถานะของผู้ที่ติดต่อ	46
7.1.7 เพิ่ม/ลบ รายชื่อผู้ติดต่อ	46
7.1.8 ป้องกัน/อนุญาต การติดต่อ	46
7.1.9 การรับส่งข้อความแบบหนึ่งต่อหนึ่ง	46
7.1.10 การรับส่งข้อความแบบเป็นกลุ่ม	46
7.1.11 การติดต่อสื่อสารด้วยเสียงและภาพวีดีโอ	47
7.2 การออกแบบโปรแกรม	47
7.2.1 CRTCTWin	48
7.2.2 CRTCIMSession	48
7.2.3 CRTCAVSession	48
7.2.4 CRTCTWatcher	48
7.2.5 CRTCTSession	48
7.2.6 CRTCTEvent	48
7.3 สถาปัตยกรรมของระบบ	49
7.3.1 ซิพเซิร์ฟเวอร์(SIP Server)	49
7.3.2 ไคลเอนต์ (Client)	50
7.4 รูปแบบการรับส่งแพ็คเกจระหว่างเอนทิตีของฟังก์ชันการทำงาน	50
7.4.1 การล็อกอินเข้าสู่ระบบ (Log in)	50
7.4.2 การพิสูจน์ตน (Authentication)	51
7.4.3 การแสดงรายชื่อผู้ติดต่อและสถานะของผู้ที่ติดต่อ	51
(Contact List and thier Status)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

7.4.4	เพิ่ม/ลบ รายชื่อผู้ติดต่อ (Add/Remove Contact List)	52
7.4.5	ป้องกัน / อนุญาต การติดต่อ (Block / Unblock)	54
7.4.6	การรับส่งข้อความแบบหนึ่งต่อหนึ่ง (Chat)	55
7.4.7	การรับส่งข้อความแบบเป็นกลุ่ม (Multi Chat)	56
7.4.8	การติดต่อสื่อสารด้วยเสียงและภาพวิดีโอ (Audio and Video Communication)	57
7.4.9	การล็อกเอาต์ออกจากระบบ (Log out)	57
7.5	เครื่องมือที่ใช้ในการพัฒนา	58
บทที่ 8 การใช้งานโปรแกรม		59
8.1	การใช้โปรแกรมบนพีซี	59
8.1.1	การ login	59
8.1.2	การเพิ่มรายชื่อ	60
8.1.3	การ block/unblock	61
8.1.4	การเปลี่ยนสถานะของผู้ใช้	62
8.1.5	การส่ง Instant Message	63
8.1.6	การติดต่อโดยการส่งเสียง	64
8.1.7	การติดต่อโดยการส่งวิดีโอ	66
8.2	การใช้โปรแกรมบนมือถือ	67
8.2.1	การ login	67
8.2.2	การร้องขอการติดต่อ	68
8.2.3	การส่งข้อความแบบหนึ่งต่อหนึ่ง	70
บทที่ 9 บทวิจารณ์และสรุป		72
9.1	สรุปและวิจารณ์ผลของการทำโครงการ	72
9.2	แนวทางการพัฒนาต่อ	72
ภาคผนวก		
ภาคผนวก ก	การติดตั้ง Vocal เวอร์ชัน 1.5	73
ภาคผนวก ข	การใช้โปรแกรม Vocal	78
ภาคผนวก ค	การติดตั้ง JSR 180 RI package	84
ภาคผนวก ง	การติดตั้ง J2ME Wireless Toolkit 2.2	88
ภาคผนวก จ	Codec	93
บรรณานุกรม		95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 4-1 แสดงความหมายของการเชื่อมต่อแต่ละชนิด	32
ตารางที่ 4-2 แสดงชนิดของ media และจำนวนคนที่เข้าร่วมได้	32
ตารางที่ 4-3 แสดง profile ในแต่ละชนิดของการเชื่อมต่อ	35
ตารางที่ 4-4 แสดงรูปแบบของ TEL URL	35
ตารางที่ 4-5 แสดงรูปแบบของ SIP URL	36
ตารางที่ 7-1 แสดงฟังก์ชันการทำงานของโปรแกรม	45
ตารางที่ จ-1 แสดงคุณสมบัติของ Codec ภาพแต่ละชนิด	93
ตารางที่ จ-2 แสดงคุณสมบัติของ Codec เสียงแต่ละชนิด	94



สารบัญภาพ

หน้า

รูปที่ 2-1 แสดงเครือข่ายคอนเน็กชันโอเรียนเต็ล	4
รูปที่ 2-2 โครงข่ายคอนเน็กชันเลส	4
รูปที่ 2-3 โครงข่ายคอนเวิร์จ	5
รูปที่ 2-4 แสดงอุปกรณ์โครงข่ายวีโอไอพี	6
รูปที่ 2-5 แสดงถึง delay และ jitter	8
รูปที่ 2-6 แสดงการติดต่อระหว่าง PC-to-PC	9
รูปที่ 2-7 แสดงการติดต่อระหว่าง PC-to-phone	9
รูปที่ 2-8 แสดงการติดต่อระหว่าง phone-to-phone	10
รูปที่ 3-1 The Internet Multimedia Protocol stack	12
รูปที่ 3-2 การติดต่อระหว่าง UAC กับ UAS	13
รูปที่ 3-3 การติดต่อระหว่าง user agent โดยผ่าน proxy เซิร์ฟเวอร์	14
รูปที่ 3-4 จะแสดง ขั้นตอนของการ registration	15
รูปที่ 3-5 การทำงานของ ชิป Redirection	15
รูปที่ 3-6 การสร้างเซสชันใน โดเมนเดียวกัน	16
รูปที่ 3-7 การสร้างเซสชันที่ต่าง โดเมนกัน	17
รูปที่ 3-8 การส่งแมสเสจหากันในการสื่อสารระหว่าง 2 user agent	22
รูปที่ 3-9 แสดงถึงแมสเสจของ 1 ไคอะล็อก	23
รูปที่ 3-10 แสดงตัวอย่างของแมสเสจภายใน ไคอะล็อกที่ถูกส่งโดย proxy	24
รูปที่ 3-11 ลำดับการส่ง REGISTER message	25
รูปที่ 3-12 ลำดับการส่ง INVITE message	25
รูปที่ 3-13 ลำดับการส่ง BYE message	26
รูปที่ 3-14 ลำดับการส่งแมสเสจติดต่อกัน	27
รูปที่ 4-1 แสดงถึง object และ interface ของ RTC Version 1.2	29
รูปที่ 4-2 แสดงรูปเครือข่ายการติดต่อของ PC-to-PC	33
รูปที่ 4-3 แสดงรูปเครือข่ายการติดต่อของ PC-to-phone	33
รูปที่ 4-4 แสดงรูปเครือข่ายการติดต่อของ phone-to-phone	34
รูปที่ 5-1 แสดงความสัมพันธ์ระหว่าง J2ME Configuration และ J2SE	40
รูปที่ 5-2 โครงสร้างของ MIDP	41
รูปที่ 6-1 แสดง class diagram ของ API และความสัมพันธ์ระหว่างคลาส	42
รูปที่ 6-2 แสดงการเชื่อมต่อระหว่าง UAC กับ UAS ซึ่งเป็น terminal แบบเดียวกัน	43
รูปที่ 6-3 แสดงการเชื่อมต่อระหว่าง UAC กับ UAS ซึ่งเป็น terminal ต่างชนิดกัน	43
รูปที่ 7 – 1 โครงสร้างของโปรแกรมบนเครื่องคอมพิวเตอร์พีซี	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7-2 โครงสร้างของโปรแกรมบนอุปกรณ์มือถือ	49
รูปที่ 7-3 สถาปัตยกรรมของระบบ	49
รูปที่ 7-4 รูปแบบการส่งแพ็คเกจเมื่อไคลเอนต์ทำการล็อกอินเข้าสู่ระบบ	50
รูปที่ 7-5 รูปแบบการส่งแพ็คเกจในการล็อกอินเข้าสู่ระบบสำหรับสมาชิกที่ต้องมีการพิสูจน์ตน	51
รูปที่ 7-6 รูปแบบการส่งแพ็คเกจในการแสดงรายชื่อผู้ติดต่อ	51
รูปที่ 7-7 รูปแบบการส่งแพ็คเกจในการเพิ่มรายชื่อผู้ติดต่อ	53
รูปที่ 7-8 รูปแบบการส่งแพ็คเกจในการลบรายชื่อผู้ติดต่อ	53
รูปที่ 7-9 รูปแบบการส่งแพ็คเกจในการป้องกันการติดต่อ	54
รูปที่ 7-10 รูปแบบการส่งแพ็คเกจในการอนุญาตการติดต่อ	54
รูปที่ 7-11 รูปแบบการส่งแพ็คเกจในการรับส่งข้อความแบบหนึ่งต่อหนึ่ง	55
รูปที่ 7-12 รูปแบบการส่งแพ็คเกจในการรับส่งข้อความแบบเป็นกลุ่ม	56
รูปที่ 7-13 รูปแบบการส่งแพ็คเกจในการติดต่อสื่อสารด้วยเสียงและภาพวิดีโอ	57
รูปที่ 7-14 รูปแบบการส่งแพ็คเกจในการล็อกเอาต์ออกจากระบบ	58
รูปที่ 8-1 แสดงหน้าตาการล็อกอิน	59
รูปที่ 8-2 แสดงหน้าตาการพิสูจน์ตน	59
รูปที่ 8-3 แสดงหน้าตาหลักสำหรับแสดงชื่อผู้ใช้อื่นในระบบ	60
รูปที่ 8-4 แสดงหน้าตาที่ใช้ในการเพิ่มรายชื่อผู้ที่ต้องติดต่อด้วย	60
รูปที่ 8-5 แสดงรายชื่อที่ถูกเพิ่มเข้าไป	61
รูปที่ 8-6 แสดงหน้าตาที่แสดงการตอบรับคำร้องขอการติดต่อ	61
รูปที่ 8-7 แสดงรูปการทำการป้องกันการติดต่อ	61
รูปที่ 8-8 แสดงรูป หลังจากทำการป้องกันการติดต่อแล้ว	62
รูปที่ 8-9 การเปลี่ยนสถานะของผู้ใช้	62
รูปที่ 8-10 แสดงสถานะของผู้ใช้ที่ได้เปลี่ยนแล้ว	62
รูปที่ 8-11 แสดงวิธีการติดต่อสื่อสารด้วยการส่งข้อความ	63
รูปที่ 8-12 แสดงหน้าตาสำหรับใส่ URI ของผู้ที่ต้องการจะติดต่อด้วย	63
รูปที่ 8-13 แสดงหน้าตาแสดงการส่งข้อความ	63
รูปที่ 8-14 แสดงการเพิ่มผู้ที่อยากจะสนทนาด้วยเข้าไปอีก	64
รูปที่ 8-15 แสดงหน้าตาสำหรับใส่ URI ของคนที่ต้องการให้มาสนทนาด้วย	64
รูปที่ 8-16 แสดงหน้าตาหลังจากเพิ่มคู่สนทนาเข้ามา	64
รูปที่ 8-17 แสดงหน้าตาในการร้องขอการติดต่อเพื่อสื่อสารด้วยเสียง	65
รูปที่ 8-18 แสดงหน้าตาสำหรับใส่ URI ของคนที่เราต้องการติดต่อด้วย	65
รูปที่ 8-19 แสดงหน้าตาที่ใช้ในขณะที่ทำการติดต่อโดยใช้เสียง	65
รูปที่ 8-20 แสดงเลือกการติดต่อผ่านทางชื่อของผู้ใช้	66
รูปที่ 8-21 แสดงหน้าตาการติดต่อผ่านวิดีโอ	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8-22 แสดงหน้าจอเริ่มต้นก่อนการล็อกอิน	67
รูปที่ 8-23 แสดงหน้าจอการทำการล็อกอิน	67
รูปที่ 8-24 แสดงหน้าจอยืนยันข้อมูลที่ใช้ในการล็อกอิน	68
รูปที่ 8-25 แสดงหน้าจอสถานะออนไลน์	68
รูปที่ 8-26 แสดงหน้าก่อนการทำการร้องขอการติดต่อ	68
รูปที่ 8-27 แสดงหน้าจอยืนยันการร้องขอการติดต่อ	69
รูปที่ 8-28 แสดงหน้าจอขณะทำการร้องขอการติดต่อ	69
รูปที่ 8-29 แสดงหน้าหน้าจอเมื่อมีคนทำการร้องขอการติดต่อมา	69
รูปที่ 8-30 แสดงหน้าจอการยกเลิกการร้องขอการติดต่อ	70
รูปที่ 8-31 แสดงหน้าจอการติดต่อสื่อสาร โดยการส่งข้อความ	70
รูปที่ 8-32 แสดงหน้าจอเมื่อผู้สนทนาด้วยออกจากระบบแล้ว	71
รูปที่ ก-1 ผลลัพธ์ของคำสั่ง <code>ifconfig -a eth0</code>	73
รูปที่ ก-2 ผลลัพธ์ของคำสั่ง <code>cat /etc/resolv.conf</code>	74
รูปที่ ก-3 ผลลัพธ์ของคำสั่ง <code>hostname</code>	74
รูปที่ ก-4 ผลลัพธ์ของคำสั่ง <code>make install</code>	75
รูปที่ ก-5 ผลลัพธ์ของคำสั่ง <code>/usr/local/vocal/bin/allinoneconfigure/allinoneconfigure/</code>	75
รูปที่ ก-6 ผลลัพธ์เมื่อทำการคอนฟิกเสร็จ	76
รูปที่ ก-7 IMPORTANT INFORMATION	76
รูปที่ ก-8 ผลลัพธ์ของการใช้คำสั่ง <code>/etc/local/vocal/bin/verifysip -a</code>	77
รูปที่ ข-1 หน้าแรกของโปรแกรม Vocal	78
รูปที่ ข-2 หน้าจอการล็อกอินเข้าสู่ระบบ	79
รูปที่ ข-3 ผลลัพธ์เมื่อทำการล็อกเอาต์ออกจากระบบ	80
รูปที่ ข-4 หน้าจอการเพิ่มผู้ใช้	80
รูปที่ ข-5 หน้าจอเลือกผู้ใช้	81
รูปที่ ข-6 หน้าจอการเปลี่ยนแปลงข้อมูลผู้ใช้	81
รูปที่ ข-7 หน้าจอการลบผู้ใช้	82
รูปที่ ข-8 หน้าจอการเปลี่ยน ip ของเครือข่าย	82
รูปที่ ข-9 หน้าจอแสดงสถานะของระบบ	83
รูปที่ ค-1 แสดงการเลือก api ที่ได้รับการสนับสนุน	86
รูปที่ ง-1 ไดรกทอรีของ J2SDK	88
รูปที่ ง-2 กำหนด PATH และ CLASSPATH ในไฟล์ <code>autoexec.bat</code>	89
รูปที่ ง-3 รูปการแสดงผลของการ SET PATH และ CLASSPATH	89
รูปที่ ง-4 แสดงการตรวจ version ของ java	89
รูปที่ ง-5 หน้าแรกของการติดตั้ง	90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ง-6 ตัวติดตั้ง WTK จะหา Java Virtual Machine ให้กด Next	90
รูปที่ ง-7 หน้าจอหลักของโปรแกรม KToolbar	91
รูปที่ ง-8 สร้างโปรเจกต์ใหม่	91
รูปที่ ง-9 เมื่อสร้างโปรเจกต์จะมีรายละเอียดปรากฏที่คอนโซล	92
รูปที่ ง-10 โครงสร้างไฟล์เดอร์ภายในโปรเจกต์	92



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันเครือข่ายอินเทอร์เน็ตได้มีจำนวนผู้ใช้เพิ่มขึ้นอย่างรวดเร็ว เนื่องจากผู้ใช้สามารถค้นหาข้อมูลที่ต้องการจากแหล่งข้อมูลขนาดใหญ่และสามารถติดต่อสื่อสารกับบุคคลจากที่ต่างๆ ได้อย่างสะดวก นอกจากนี้การพัฒนาการให้บริการในรูปแบบใหม่เพื่อรองรับความต้องการของผู้ใช้ได้เพิ่มขึ้นอย่างรวดเร็วเนื่องมาจากความยืดหยุ่นของเครือข่ายอินเทอร์เน็ตทำให้สามารถพัฒนาการให้บริการได้ง่ายกว่าเครือข่ายอื่นๆ

การให้บริการอย่างหนึ่งที่กำลังได้รับความนิยมอย่างมากคือการส่งข้อมูลเสียงผ่านเครือข่าย รวมถึงการส่งข้อมูลแบบเรียลไทม์ (Real-Time) ในรูปแบบอื่นๆ การให้บริการแบบเรียลไทม์นี้ ช่วยอำนวยความสะดวกต่อผู้ใช้เป็นอย่างมาก เนื่องจากการประหยัดค่าใช้จ่ายในการติดต่อสื่อสาร ช่วยประหยัดค่าโทรศัพท์ทางไกล รวมทั้งยังสามารถส่งข้อมูลเหล่านี้รวมกันไปในเครือข่ายอินเทอร์เน็ตเพียงเครือข่ายเดียว และเมื่อพิจารณาในแง่ของผู้ให้บริการหรือผู้ดูแลเครือข่าย การพัฒนาปรับปรุงเครือข่ายหรือการให้บริการ สามารถทำได้ง่ายและสะดวกกว่า เพราะไม่จำเป็นต้องแยกเครือข่ายสำหรับข้อมูล เสียง และวิดีโอ ออกจากกัน

อย่างไรก็ตามเนื่องจากเครือข่ายอินเทอร์เน็ตเป็นเครือข่ายที่ไม่รับประกันในเรื่องคุณภาพของการให้บริการ เครือข่ายอินเทอร์เน็ตให้บริการการส่งข้อมูลแบบทั่วไปซึ่งไม่ต้องการคุณสมบัติเรียลไทม์แต่ในการส่งข้อมูลชนิดเสียง ซึ่งต้องการคุณสมบัติแบบเรียลไทม์ดังนั้นจึงจำเป็นต้องมีโปรโตคอลที่สามารถรองรับคุณสมบัติดังกล่าวในการส่งข้อมูล ซึ่งได้แก่ โปรโตคอลสซิป (SIP) เป็นโปรโตคอลในชั้นแอปพลิเคชัน และใช้บริการของโปรโตคอลในชั้นที่ต่ำกว่าซิป สามารถขอรับบริการได้ทั้ง UDP และ TCP โปรโตคอลซิป มีหน้าที่ในการสร้าง สิ้นสุดการเชื่อมต่อ หาดำแหน่งของเครื่องปลายทาง แต่เนื่องจากฟังก์ชันของโปรโตคอลซิปก็ยังมีขอบเขตจำกัดอยู่ ดังนั้นจึงได้นำโปรโตคอลอื่นมาช่วยในการทำงานด้วย ได้แก่ RTP/RTCP เพื่อให้สามารถให้บริการได้อย่างมีประสิทธิภาพมากขึ้น

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษาการพัฒนาโปรแกรมบนแพลตฟอร์มวินโดวส์ (พีซี) และอุปกรณ์มือถือ
- 1.2.2 เพื่อศึกษาการทำงานของโปรโตคอลซิป
- 1.2.3 เพื่อศึกษาการทำงานการส่งข้อมูลแบบมัลติมีเดีย
- 1.2.4 เพื่อพัฒนาโปรแกรมติดต่อสื่อสารด้วยเสียงโดยใช้โปรโตคอลซิปผ่านซิปเซิร์ฟเวอร์
- 1.2.5 เพื่อสร้างโปรแกรมต้นแบบในการพัฒนาซิปแอปพลิเคชันบนอุปกรณ์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เป้าหมายและขอบเขตของโครงการ

พัฒนาโปรแกรมการติดต่อสื่อสารด้วยเสียงบนแพลตฟอร์มพีซีและอุปกรณ์มือถือ ซึ่งสามารถติดต่อสื่อสารกันได้โดยผ่านทางซีพซีเอฟเวอร์ โดยมีรายละเอียดดังนี้

- 1.3.1 สามารถทำการติดต่อระหว่างไคลเอนท์ได้ ผ่านทางซีพซีเอฟเวอร์
- 1.3.2 สามารถทำการส่งข้อความระหว่างไคลเอนท์ได้
- 1.3.3 สามารถติดต่อสื่อสารกันด้วยเสียงและภาพวีดีโอได้

1.4 ขั้นตอนการดำเนินงาน

- 1.4.1 ศึกษาหาข้อมูลเกี่ยวกับการส่งข้อมูลมัลติมีเดีย (ข้อมูลเสียง วีดีโอ)
- 1.4.2 ศึกษาหาข้อมูลหลักการทำงานของ โปรโตคอลซีพ
- 1.4.3 ทดลองติดตั้งซีพซีเอฟเวอร์และทดสอบการทำงาน
- 1.4.4 ศึกษาหาข้อมูลการ โปรแกรมซีพแอพลิเคชัน (SIP APIs)
- 1.4.5 ศึกษาการเขียน โปรแกรมด้วย Visual C++
- 1.4.6 พัฒนาโปรแกรมบนพีซีให้สามารถติดต่อกับซีพซีเอฟเวอร์ได้
- 1.4.7 พัฒนาโปรแกรมบนพีซีให้สามารถส่งข้อความระหว่างกันผ่านทางซีพซีเอฟเวอร์
- 1.4.8 พัฒนาโปรแกรมบนพีซีให้สามารถติดต่อสื่อสารด้วยเสียงผ่านทางซีพซีเอฟเวอร์
- 1.4.9 พัฒนาโปรแกรมบนพีซีให้สามารถส่งภาพวีดีโอด้วยกล้องเว็บแคมผ่านทางซีพซีเอฟเวอร์
- 1.4.10.ศึกษาการเขียนโปรแกรมด้วยภาษา J2ME
- 1.4.11. พัฒนาโปรแกรมบนอุปกรณ์มือถือให้สามารถทำการติดต่อกับซีพซีเอฟเวอร์ได้
- 1.4.12.พัฒนาโปรแกรมบนอุปกรณ์มือถือให้สามารถส่งข้อความระหว่างกันผ่านทางซีพซีเอฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

วีโอไอพี (VOIP)

โครงข่ายคอนเวอร์จ (Converged Network) คือ โครงข่ายที่มีการส่งทั้งข้อมูลที่เป็นเสียง (Voice) และข้อมูลที่ไม่ใช่เสียง (Non voice) นั้นเอง ซึ่งจริง ๆ แล้วแนวคิดนี้มีมาตั้งแต่การพัฒนาระบบ ไอเอสดีเอ็น (ISDN : Integrated Services Digital Network) ในปี 1980 โดยองค์กร ITU-T (International Telecommunication Union – Telecommunications Standards Sector) แล้ว แต่ว่าในปัจจุบันมีการนำมาประยุกต์ใช้ค่อนข้างมาก เนื่องจากเทคโนโลยีหลาย ๆ ด้านได้รับการพัฒนาไปมาก เช่น เทคนิคในการส่งข้อมูลด้วยอัตราที่สูงขึ้น การประมวลผลทางด้านคอมพิวเตอร์ทำได้เร็วยิ่งขึ้น และการใช้เครือข่ายอินเทอร์เน็ตเป็นที่นิยมอย่างมากขึ้น นอกจากนี้รูปแบบของอุปกรณ์สื่อสารยังมีความหลากหลาย เป็นต้น

วีโอไอพี (VOIP : Voice over IP) เป็นมาตรฐานในการส่งข้อมูลเสียงผ่านโครงข่ายที่ใช้ไอพี (IP : Internet Protocol) ซึ่งเป็นโปรโตคอลที่อยู่ในชั้นเน็ตเวิร์ค เช่น เครือข่ายอินเทอร์เน็ต ซึ่งการที่จะทำให้ออมูลเสียงผ่านเครือข่ายอินเทอร์เน็ตนั้นจำเป็นต้องแปลงข้อมูลเสียงให้เป็นแพ็คเก็ตก่อน ซึ่งหลังจากข้อมูลเสียงถูกแปลงเป็นแพ็คเก็ตแล้วก็ใช้วิธีเดียวกันกับการส่งข้อมูลที่ไม่ใช่เสียง ผ่านเครือข่ายอินเทอร์เน็ตนั่นเอง โดยที่โครงข่ายในที่นี้หมายถึงการเชื่อมต่อของอุปกรณ์สื่อสารตั้งแต่สองอุปกรณ์ขึ้นไปเข้าด้วยกัน ส่วนเครือข่ายในที่นี้หมายถึงเฉพาะการเชื่อมต่อคอมพิวเตอร์ตั้งแต่สองเครื่องขึ้นไปเข้าด้วยกัน

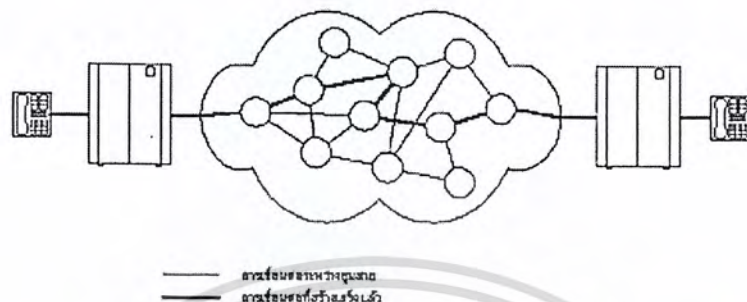
ระบบวีโอไอพีเป็นระบบที่นำสัญญาณเสียงที่ผ่านการดิจิไทซ์ โดยหนึ่งช่องเสียงเมื่อแปลงเป็นข้อมูลจะมีขนาด 64 Kbps การนำข้อมูลเสียงขนาด 64 Kbps นี้ ต้องนำมาบีบอัดก่อน โดยทั่วไปจะเหลือประมาณ 10 Kbps ต่อช่องสัญญาณเสียง แล้วจึงบรรจุลงในไอพีแพ็คเก็ต เพื่อส่งผ่านทางเครือข่ายไอพี การสื่อสารผ่านทางเครือข่ายไอพีต้องมีเราท์เตอร์ที่ทำหน้าที่พิเศษเพื่อประกันคุณภาพของช่องสัญญาณไอพี เพื่อให้ข้อมูลไปถึงปลายทางหรือกลับมาได้อย่างถูกต้อง และอาจมีการให้สิทธิพิเศษก่อนแพ็คเก็ตไอพีอื่น เพื่อการให้บริการที่ทำให้เสียงมีคุณภาพ นอกจากนี้วีโอไอพียังถูกนิยามให้เป็นความสามารถในการใช้โทรศัพท์ (ทำทุกอย่างที่เราสามารถทำได้ในปัจจุบันด้วย PSTN-Public Switched Telephone Network) และส่งสำเนาเอกสารบนเครือข่ายข้อมูลที่มีพื้นฐานอยู่บนไอพีด้วย QoS (Quality of Service-การวัดประสิทธิภาพของระบบสื่อสารซึ่งแสดงถึงคุณภาพในการส่งและความสะดวกในการใช้บริการ) ที่เหมาะสมและราคากับผลประโยชน์ที่คุ้มค่ากว่า

2.1 ลักษณะของโครงข่ายคอนเวอร์จ

เป็นที่ทราบกันโดยทั่วไปว่าโครงข่ายที่รองรับข้อมูลเสียงมีลักษณะเป็นแบบคอนเน็กชันโอเรียนเต็ด (Connection Oriented) ดังแสดงในรูปที่ 1 คือ ต้องมีการสร้างเส้นทางก่อนการสนทนา เช่น ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโทรศัพท์ที่นั่นเราต้องหมุนหรือกดหมายเลขผู้ใช้ปลายทาง หลังจากนั้นระบบจะหาเส้นทางให้ และเมื่อได้เส้นทางการเชื่อมต่อจากทางจากผู้โทรไปยังผู้รับแล้ว จึงสามารถสนทนาได้



รูปที่ 2-1 แสดงโครงข่ายคอนเน็กชันโอเรียนเต็ด

ส่วนโครงข่ายที่รองรับข้อมูลที่ไม่ใช่เสียงจะมีลักษณะเป็นแบบคอนเน็กชันเลส (Connectionless) ดังแสดงในรูปที่ 2 คือระบบไม่จำเป็นต้องสร้างเส้นทางก่อนการสื่อสารระหว่างผู้ส่งไปยังผู้รับ เช่น การรับส่งอีเมล ข้อมูลของอีเมลจะถูกแบ่งให้เป็นแพ็กเก็ตแล้วส่งไปตามโหนด (Node) ต่าง ๆ ในโครงข่ายจนกระทั่งถึงปลายทาง โดยอาจมีหลายเส้นทางที่รองรับแพ็กเก็ตของอีเมลฉบับนี้

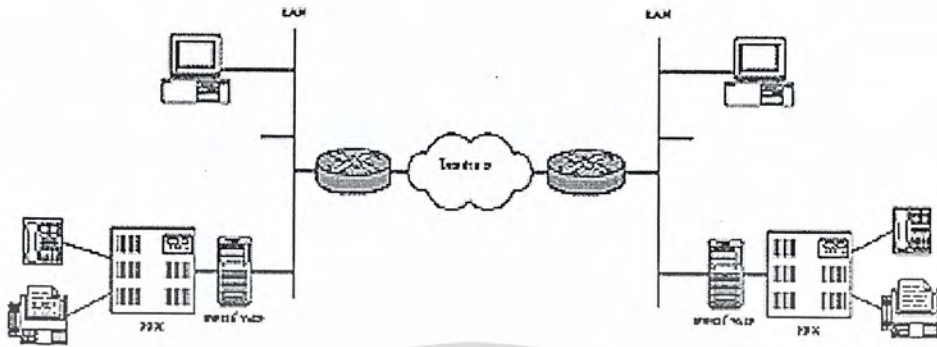


รูปที่ 2-2 โครงข่ายคอนเน็กชันเลส

สำหรับโครงข่ายที่รวมทั้งข้อมูลเสียงและข้อมูลที่ไม่ใช่เสียงหรือโครงข่ายคอนเน็กชันเลสสามารถแสดงตัวอย่างได้ในรูปที่ 3 ซึ่งใช้โครงข่ายไอพีเป็นตัวกลางระหว่างผู้ส่งไปยังผู้รับ โดยข้อมูลที่ไม่ใช่เสียงสามารถส่งผ่าน LAN (Local Area Network) ผ่านเราเตอร์โครงข่ายไอพีไปยังเราเตอร์ฝั่งผู้รับเพื่อที่จะส่งข้อมูลไปยังผู้รับอีกทีหนึ่ง ในขณะที่ข้อมูลเสียงจากพีบีเอ็กซ์ (PBX : Private Branch Exchange) จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงเฟรมข่าวสารจากรูปแบบอะนาล็อกให้เป็นแพ็คเก็ตโดยเกตเวย์ของวีโอไอพีเพื่อให้สามารถส่งข้อมูลผ่านโครงข่ายอินเทอร์เน็ตได้

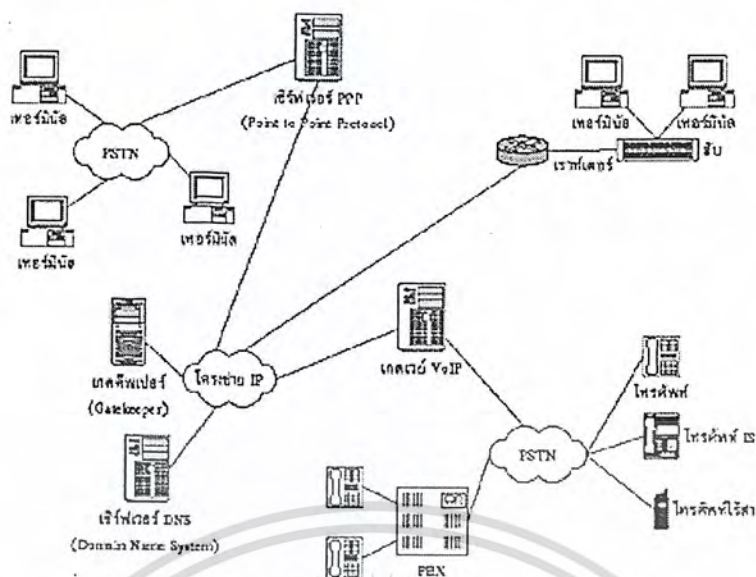


รูปที่ 2-3 โครงข่ายคอนเวอริ่ง

2.2 องค์ประกอบโครงข่ายวีโอไอพี

ในรูปที่ 4 แสดงตัวอย่างขององค์ประกอบต่าง ๆ ในโครงข่ายโทรศัพท์ไอพี ซึ่งประกอบด้วยโครงข่ายข้อมูลเสียงแบบเดิม เช่น เครื่องโทรศัพท์ แฟกซ์ ทีบีเอ็กซ์ ส่วนโครงข่ายข้อมูลแบบเดิมก็มี เช่น คอมพิวเตอร์ที่ทำหน้าที่ต่าง ๆ ได้แก่ เซิร์ฟเวอร์ โฮสต์ หรือ เทอร์มินัล เป็นต้น ส่วนอุปกรณ์ที่เพิ่มเข้ามาใหม่คือ เกตเวย์และเกตคีพเปอร์ (Gatekeeper) ซึ่งสร้างมาเพื่อทำให้อุปกรณ์ของข้อมูลเสียงและข้อมูลที่ไม่ใช่เสียงสามารถทำงานร่วมกันได้ ในรูปที่ 4 ยังแสดงถึงโครงข่ายโทรศัพท์สาธารณะ (PSTN : Public Switched Telephone Network) การเชื่อมต่อ WAN (Wide Area Network) เช่น การใช้งานระบบไอเอสดีเอ็น การเชื่อมต่อเราท์เตอร์ ไอพี (IP Router) และการเชื่อมต่อกับฮับด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 แสดงอุปกรณ์โครงข่ายวีโอไอพี

โครงข่ายในรูปที่ 4 เริ่มต้นจากการใช้โทรศัพท์ที่จากโครงข่ายทีเอสทีเอ็น สัญญาณของข้อมูลเสียงซึ่งเป็นสัญญาณแอนะล็อกจะถูกเปลี่ยนให้เป็นสัญญาณดิจิทัลที่พีบีเอ็กซ์หรือเกตเวย์ จากนั้นเกตเวย์จะติดต่อกับเกตเวย์เพื่อหาข้อมูลของอุปกรณ์ปลายทาง เช่น ไอพีแอดเดรส หรือความสามารถในการรับข้อมูล นั่นคือสัญญาณเสียงจะถูกแปลงให้เป็นแพ็กเก็ตแล้วส่งไปบนโครงข่ายไอพีเพื่อไปยังเกตเวย์ของโครงข่ายปลายทาง จากนั้นแพ็กเก็ตต่าง ๆ จะได้รับการจัดเรียงใหม่ แล้วแปลงกลับให้เป็นเสียงแล้วส่งไปยังอุปกรณ์ปลายทาง เป็นอันสิ้นสุดกระบวนการของการส่งและรับข้อมูล

2.3 ผลดีของวีโอไอพี

ข้อดีของวีโอไอพีที่เห็นได้ชัดและทำให้วีโอไอพีประสบความสำเร็จได้รับความสนใจจากคนทั่วโลกคือทำให้ประหยัดค่าโทรศัพท์ทางไกลและมีต้นทุนต่ำอันเนื่องมาจากอัตราค่าบริการของอินเทอร์เน็ตที่มีค่าคงที่ ข้อได้เปรียบดังกล่าวทำให้วีโอไอพี เป็นคู่แข่งที่น่ากลัวของศูนย์บริการโทรศัพท์ดั้งเดิม ส่งผลในแง่ดีคือกระตุ้นให้ศูนย์บริการโทรศัพท์ดังกล่าวเกิดการพัฒนาด้านเรื่องค่าบริการและฟังก์ชันการทำงาน

ผลดีของวีโอไอพี ได้แก่

1. ลดค่าโทรศัพท์ทางไกลประหยัดค่าส่งแฟกซ์และข้อความเสียงเนื่องจากอัตราค่าบริการของอินเทอร์เน็ตคงที่ไม่ขึ้นกับระยะเวลาและระยะทางลดต้นทุนค่าอุปกรณ์เนื่องจากอุปกรณ์ที่ใช้ส่งสัญญาณข้อมูลและเสียงเป็นตัวเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จำนวนอุปกรณ์ที่ต้องใช้ลดลงเพราะโครงสร้างพื้นฐานของวีโอไอพีสนับสนุนการสื่อสารทุกรูปแบบ การรวมโครงสร้างพื้นฐานย่อยเหล่านี้เข้าด้วยกันสามารถรองรับ dynamic bandwidth ได้ดีที่สุด และยังออกแบบให้ทนต่อความผิดพลาดด้วย

3. SNMP-based management สามารถให้บริการได้ทั้งเสียงและข้อมูลโดยใช้วีโอไอพีการที่อินเทอร์เน็ต โพรโตคอล มีการใช้งานอย่างกว้างขวางจะช่วยลดความยุ่งยาก เพิ่มความยืดหยุ่นและลดการเกิดความผิดพลาด ส่งผลให้ระบบมีความมั่นคงมากขึ้น

4. ประโยชน์ในระยะแรกของการใช้วีโอไอพี คือการใช้โทรศัพท์ แฟกซ์ ส่วนประโยชน์ในระยะยาวของการใช้วีโอไอพี คือ การประยุกต์ใช้มีลติมีเดียและมัลติเซอร์วิท

2.4 ปัจจัยที่มีผลต่อคุณภาพเสียง

ในการจะส่งสัญญาณเสียงไปนั้นก็มีหลายปัจจัยที่มีผลต่อคุณภาพของเสียงที่ส่งไปซึ่งอาจจะสรุปถึงปัจจัยที่มีผลต่อคุณภาพเสียงได้ดังนี้

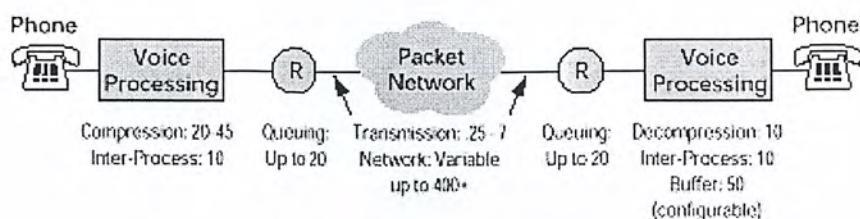
2.4.1. ดีเลย์ (Delay)

ความล่าช้าในการส่งสัญญาณข้อมูลเสียงแบบแพ็กเก็ตเกิดจากการรวบรวมสัญญาณที่สุ่มตัวอย่างจากสัญญาณเสียง เวลาในการเข้ารหัส/ถอดรหัส เวลาในการเข้าแพ็กเก็ต jitter buffer delay และความล่าช้าของเน็ตเวิร์ค ปัญหาที่เกิดจากความล่าช้าของสัญญาณจากปลายสายหนึ่งถึงอีกปลายสายหนึ่งในเครือข่ายสัญญาณเสียง คือ เสียงสะท้อนและผู้พูดพูดซ้อนกันจะเกิดเสียงสะท้อน (echo) ขึ้นเมื่อความล่าช้าของเสียงที่เดินทาง 1 รอบมีค่ามากกว่า 50 ms ระบบวีโอไอพีจำเป็นจะต้องมีการควบคุมเสียงสะท้อนและโปรแกรมกำจัดสัญญาณสะท้อน การที่ผู้พูดพูดซ้อนกัน (talker overlap) จะเป็นปัญหาลำคัญเมื่อความล่าช้าของสัญญาณเสียงทิศทางเดียว (one way delay) มีค่ามากกว่า 250 ms

2.4.2. จิตเตอร์ (Jitter)

เสียงกระตุกเกิดจากความไม่แน่นอนของระยะเวลาระหว่างแพ็กเก็ตที่มาถึง ซึ่งเกิดจากความล่าช้าในการส่งสัญญาณบนเครือข่าย เราสามารถกำจัดเสียงกระตุกนี้ได้โดยการรวบรวมแพ็กเก็ตและเก็บมันไว้นานพอจนแพ็กเก็ตที่ช้าที่สุดสามารถมาถึงทันเวลาที่จะเล่นเสียงออกมาได้ถูกต้องตามลำดับ ซึ่งวิธีนี้จะเพิ่มดีเลย์แต่ลดความผันแปรของดีเลย์ของแต่ละแพ็กเก็ตที่ส่งผ่านเครือข่ายแพ็กเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-5 แสดงถึงดีเลย์ และจิตเตอร์

2.4.3. แพ็คเก็ต (Packet)

เครือข่ายไอพีไม่สามารถรับประกันได้ว่าแพ็คเก็ตที่ส่งไปครบและถูกต้องตามลำดับ แพ็คเก็ตส่วนใหญ่สูญหายไปในช่วงที่มีการใช้งานมาก (peak load) ซึ่งเกิดจากความจุไม่เพียงพอ และการถ่ายทอดเสียงต้องการความต่อเนื่องของเวลา วิธีส่งข้อมูลซ้ำใหม่บนพื้นฐานของ TCP เดิมจึงไม่เหมาะสมสำหรับสัญญาณเสียง ต้องใช้วิธีแทรกคำพูดโดยข้อนเล่นแพ็คเก็ตก่อนหน้าและส่งข่าวสารเดิมซ้ำกันหลายครั้งเพื่อชดเชยแพ็คเก็ตที่สูญหายไป โดยทั่วไประบบสามารถทนแพ็คเก็ตสูญหายได้ไม่เกิน 10%

2.5 รูปแบบการส่งสัญญาณเสียงบนเครือข่ายไอพี

เราสามารถแบ่งลักษณะการส่งสัญญาณเสียงได้เป็น 3 ลักษณะ คือ

2.5.1 PC-to-PC

พีซี ในกรณีนี้จะต้องเป็นมัลติมีเดียพีซี คือ มีระบบเสียงและไมโครโฟนต่อกับเครือข่ายไอพี และไม่ได้ต่อกับพีเอสทีเอ็น การประยุกต์ใช้พีซีและ IP-enabled telephones สามารถสื่อสารกันแบบจุดต่อจุดหรือมัลติพอยท์ ระบบแบบนี้อาจคล้ายกับการสนทนาเป็นกลุ่มและยังสามารถรวมเอาระบบแลกเปลี่ยนข้อมูล เช่น whiteboard ไว้ด้วย โดยวิธีการนี้จำเป็นต้องอาศัยเครื่องคอมพิวเตอร์ทั้งต้นทางและปลายทาง พร้อมทั้งติดตั้งโปรแกรมเดียวกัน หรือติดตั้งโปรแกรมที่สามารถใช้งานร่วมกันได้ ซึ่งรูปแบบนี้เป็นวิธีการสื่อสารที่ไม่ต้องเสียค่าบริการโทรศัพท์แต่อย่างใดเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Scenario 1: PC to PC



- ☒ Need a PC with sound card
- ☒ IP Telephony software: Cuseeme, Internet Phone, ...
- ☒ Video optional

รูปที่ 2-6 แสดงการติดต่อระหว่าง PC-to-PC

2.5.2 PC-to-telephone

Voice trunks สามารถแทนที่วงจรสัญญาณอะนาล็อก หรือดิจิทัลที่ทำหน้าที่เหมือน Voice trunks (เช่น การเชื่อมโยงแบบส่วนตัวระหว่าง PBXs ของบริษัท) หรือ PSTN-access trunks (การเชื่อมโยงระหว่าง พีบีเอ็กซ์ และ ตัวนำ) เพื่อหลีกเลี่ยงจะถูกส่งระหว่างไอพีแอดเดรสที่ถูกกำหนดล่วงหน้า การกลับไปใช้ชุมสายโทรศัพท์ หรือวงจรสัญญาณเสียงส่วนตัวอื่นๆ สามารถทำได้ในกรณีนี้

Scenario 2: PC to Phone



- ☒ Need a gateway that connects IP network to phone network (Router to PBX)

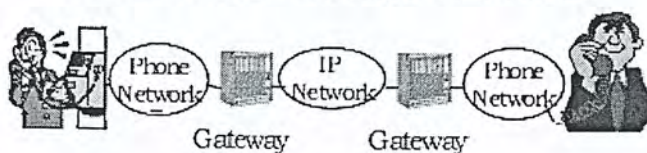
รูปที่ 2-7 แสดงการติดต่อระหว่าง PC-to-phone

2.5.3 เทเลโฟนี่ (Phone - Phone)

กรณีนี้คนพูดโทรศัพท์จะรู้สึกเหมือนกับการใช้โทรศัพท์ธรรมดา แต่จริงๆแล้วประกอบด้วยขั้นตอนการส่งเสียงบนเครือข่ายเพื่อเกิดประเภทต่างๆ ซึ่งทั้งหมดติดต่อกันระหว่างชุมสายโทรศัพท์ การติดต่อกับพีเอสทีเออน หรือการใช้โทรศัพท์ร่วมกับเครือข่ายข้อมูลจำเป็นต้องใช้เกตเวย์ในอนาคคโทรศัพท์แบบ IP-enabled จะสามารถติดต่อกันโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Scenario 3: Phone to Phone



- ☞ Need more gateways that connect IP network to phone networks
- ☞ The IP network could be dedicated intra-net or the Internet
- ☞ The phone networks could be intra-company PBXs or the carrier switches

รูปที่ 2-8 แสดงการติดต่อระหว่าง phone-to-phone

2.6 ขั้นตอนการทำงานของวีโอไอพี

2.6.1. เมื่อผู้พูดโทรศัพท์จากเครื่องโทรศัพท์ธรรมดา หรือพูดผ่านไมโครโฟนที่ถูกต้องเข้ากับการ์ดเสียงของเครื่องคอมพิวเตอร์ กลิ่นสัญญาณเสียงแบบอะนาล็อก ก็จะได้รับแปลงเป็นสัญญาณดิจิทัล จากนั้นจะถูกบีบอัดด้วยตัวถอดรหัสผ่านอุปกรณ์พีซีแอกซ์หรือวีโอไอพีเกตเวย์

2.6.2. เมื่อผ่านวีโอไอพี เกตเวย์ แล้วก็จะถูกส่งต่อไปยังเกตวิปเปอร์เพื่อค้นหาเครื่องปลายทางที่จะรับการติดต่อ เช่นหมายเลขไอพี หมายเลขโทรศัพท์ เป็นต้น แล้วแปลงเป็นแพ็คเกจข้อมูล ส่งออกไปบนระบบเครือข่ายอินเทอร์เน็ตนั่นเอง

2.6.3. จะผ่านมาที่วีโอไอพี เกตเวย์ปลายทาง แล้วจึงทำการย้อนกระบวนการทั้งหมดเพื่อส่งให้กับผู้รับปลายทางต่อไป

2.7 องค์ประกอบของวีโอไอพี

2.7.1. ซอฟต์แวร์ไคลเอนท์ หรือ ไอพีเทเล โฟนี่ อาจจะเป็นเครื่องคอมพิวเตอร์ที่ได้รับการติดตั้งโปรแกรมสื่อสารไอพี หรืออุปกรณ์ที่ได้รับการออกแบบขึ้นมาสำหรับการใช้งานโทรศัพท์ผ่านระบบอินเทอร์เน็ตโดยเฉพาะ

2.7.2. วีโอไอพี เกตเวย์ เป็นเครื่องเซิร์ฟเวอร์ที่ใช้งานสำหรับให้บริการโทรศัพท์ผ่านระบบอินเทอร์เน็ต เป็นตัวกลางในการเชื่อมต่อเข้ากับเครื่องโทรศัพท์ผู้ชุมสายโทรศัพท์สาธารณะ พีเอสทีเอ็นกับระบบเครือข่ายอินเทอร์เน็ตอย่างเครือข่ายไอพี ซึ่งการจะใช้งานระบบโทรศัพท์ไอพีต้องอาศัยอุปกรณ์นี้เป็นตัวกลางก่อน

2.7.3. เกตวิปเปอร์เป็นเครื่องเซิร์ฟเวอร์ที่เชื่อมต่อเข้ากับระบบอินเทอร์เน็ต เป็นตัวกลางที่ใช้บริหารจัดการ และควบคุมการให้บริการของ วีโอไอพี เกตเวย์ กับเครื่องคอมพิวเตอร์ที่ติดตั้งซอฟต์แวร์สำหรับใช้งาน วีโอไอพี หรือเครื่องโทรศัพท์แบบไอพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วีโอไอพี เกตเวย์ เป็นอุปกรณ์ในรูปแบบเราท์เตอร์ ที่มีคุณสมบัติเช่นเดียวกับเราท์เตอร์ที่ใช้งานกันอยู่ แต่มีคุณสมบัติที่ถูกเพิ่มเติม ให้สามารถรองรับโปรโตคอลการสื่อสารของ วีโอไอพี นั่นก็คือ โปรโตคอล H.323 SIP หรือ MGCP เป็นต้น ซึ่ง วีโอไอพี เกตเวย์ เป็นอุปกรณ์ที่ใช้เชื่อมต่อกับระบบ อินเทอร์เน็ต เพื่อให้สามารถโทรศัพท์ผ่านเครือข่ายเน็ตเวิร์กได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ซิปโปรโตคอล (SIP Protocol)

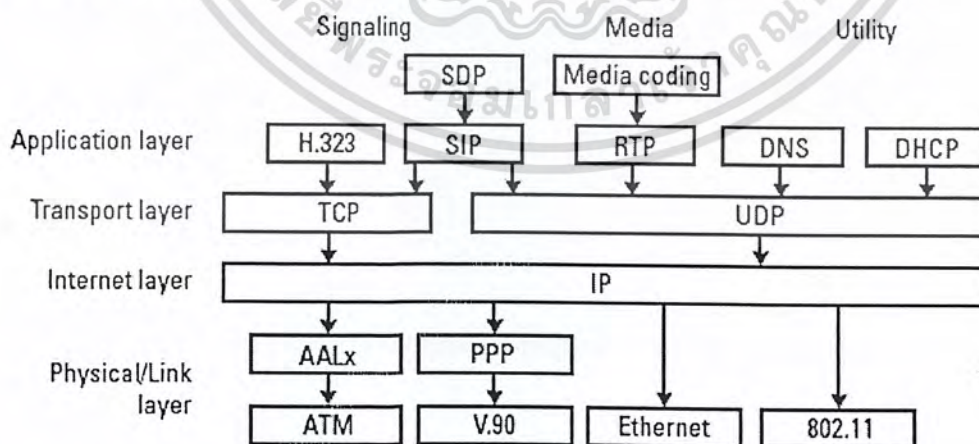
3.1 บทนำ

ซิป (SIP) ย่อมาจาก Session Initiation Protocol เป็นโปรโตคอล ที่ใช้ควบคุมในชั้น แอปพลิเคชันเลเยอร์ซึ่งถูกพัฒนาและออกแบบภายใน IETF โดยที่โปรโตคอลนี้ถูกออกแบบให้ง่ายต่อการนำไปใช้ และมีความยืดหยุ่นสามารถขยายขอบเขตได้ดี

RFC3261 จะประกอบไปด้วยข้อกำหนดหลักเพื่อใช้ในการสร้างการเชื่อมต่อ การแก้ไขสถานะของการเชื่อมต่อ การสิ้นสุดการเชื่อมต่อ ซึ่งสามารถที่จะมีผู้รับได้มากกว่า 1 คน โดยจากเซสชันนั้นเราสามารถที่จะเข้าใจถึงหลักของผู้รับ ผู้ส่งของการสื่อสาร และเข้าใจถึงสถานะของผู้รับและผู้ส่งในระหว่างการสื่อสาร

ซิปไม่ได้เป็นแค่โปรโตคอลที่อุปกรณ์สื่อสารต้องการ จุดประสงค์ของซิปนั้นก็คือทำให้การสื่อสารเกิดขึ้นได้ ซึ่งมี 2 โปรโตคอลที่สำคัญคือ RTP และ SDP โดยที่โปรโตคอล RTP นั้นจะถูกใช้ในการส่งข้อมูลมัลติมีเดียแบบเรียลไทม์ ซึ่งโปรโตคอลนี้ทำให้มีความเป็นไปได้ที่จะทำการเข้ารหัสและกระจาย (split) ข้อมูลไปในแพ็คเกจ และทำการส่งเหมือนการส่งแพ็คเกจบนอินเทอร์เน็ต ส่วนโปรโตคอล SDP นั้นจะถูกใช้ในการอธิบายและเข้ารหัสข้อมูลของเซสชัน ซึ่งการอธิบายนี้จะใช้ในการ negotiate คุณสมบัติของเซสชันซึ่งอุปกรณ์ทุกตัวสามารถที่จะมีส่วนร่วมได้

ซิปเป็น end-to-end oriented signaling protocol ซึ่งหมายความว่าลอจิกนั้นจะถูกเก็บเอาไว้ที่อุปกรณ์ปลายทาง ส่วนสถานะนั้นก็จะถูกเก็บเอาไว้เฉพาะที่อุปกรณ์ปลายทาง โดยจะไม่ได้รับผลของปัญหา single point of failure



รูปที่ 3-1 The Internet Multimedia Protocol stack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

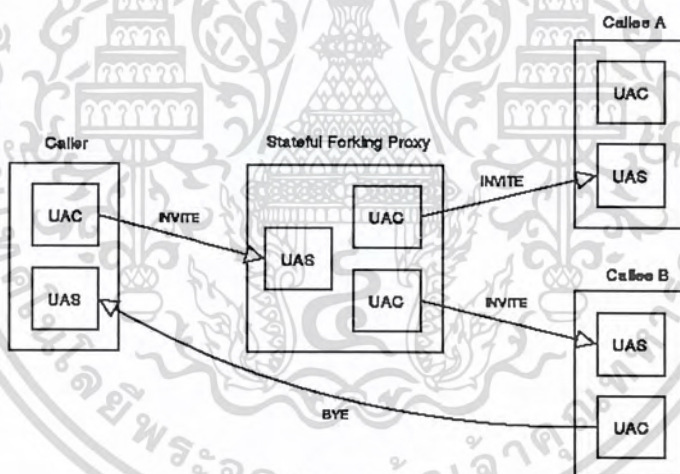
แนวคิด end-to-end ของ SIP มีความแตกต่างไปจากเครือข่ายพีเอสทีเอ็น ซึ่งสถานะและลोजิกทั้งหมดจะถูกเก็บในเครือข่ายและอุปกรณ์ เช่น โทรศัพท์ จุดประสงค์ของ SIP คือ การทำงานทุกอย่างที่พีเอสทีเอ็นทำได้ แต่แนวคิด end-to-end ทำให้เครือข่าย SIP มีความสามารถมากกว่าและเป็นการเปิดโอกาสให้สร้างบริการใหม่ๆขึ้นมา ซึ่งไม่มีในพีเอสทีเอ็น

ชีพอยู่บนพื้นฐานของโปรโตคอล HTTP ซึ่งโปรโตคอล HTTP เป็นโปรโตคอลที่เป็นที่นิยมใช้กันอย่างมากในเครือข่ายอินเทอร์เน็ต ที่จริงแล้ว HTTP จัดเป็น signaling protocol ตัวหนึ่งด้วยเหมือนกัน เนื่องจากว่า user agent ใช้โปรโตคอลนี้ในการบอกให้เว็บเซิร์ฟเวอร์ส่งเอกสารที่ต้องการให้ชีพ ใช้ในการบรรยายเซสชันพารามิเตอร์ซึ่งรายละเอียดเหล่านั้นจะถูกเข้ารหัสอยู่ในเอกสารด้วย SDP

3.2 SIP Network Elements

3.2.1 User Agent

User Agent คือ ส่วนปลายทางที่ใช้ชีพ ในการหาผู้ใช้คนอื่นและกำหนดคุณสมบัติของเซสชัน ซึ่งไม่จำเป็นที่จะต้องเป็นคอมพิวเตอร์เสมอไป อาจจะเป็นโทรศัพท์เคลื่อนที่ พีเอสทีเอ็น เกตเวย์ หรือเครื่องพีดีเอ



รูปที่ 3-2 การติดต่อระหว่าง UAC กับ UAS

User Agent มักจะกล่าวถึง User Agent Server (UAS) และ User Agent Client (UAC) โดยที่ UAC จะเป็นส่วนที่ใช้ในการส่ง Request (หรือ BYE เมื่อต้องการตัดการเชื่อมต่อ) และรับ Response ที่ส่งกลับมา ส่วน UAS จะเป็นส่วนที่รับ Request และส่ง Response กลับไป

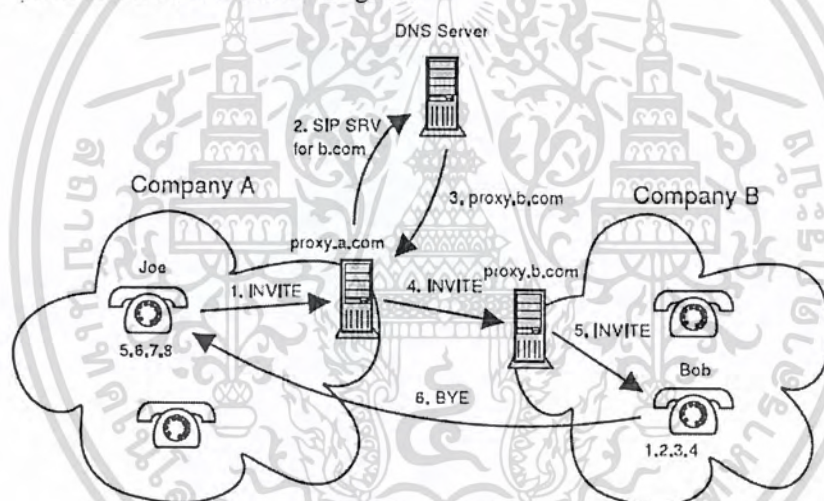
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Proxy Server

จะเป็นตัวที่อนุญาตให้สร้างโครงสร้างพื้นฐานของเครือข่ายของโฮสต์ พร็อกซีเซิร์ฟเวอร์เป็นส่วนที่มีความสำคัญต่อโครงสร้างของซิมมาก User Agent จะส่งข้อความมาให้พร็อกซีเซิร์ฟเวอร์ โดยที่จะทำการเชื่อมต่อ session invitation โดยที่ดูจากตำแหน่งปัจจุบันของผู้รับ

หน้าที่สำคัญที่สุดของพร็อกซีเซิร์ฟเวอร์ คือ การเชื่อม session invitation ให้เข้าใกล้ผู้รับมากขึ้น โดยที่ พร็อกซีเซิร์ฟเวอร์จะส่ง invitation ไปให้กับผู้รับโดยตรงเลย เมื่อมันรู้ตำแหน่งที่แน่นอนแล้วของผู้รับ แล้วผู้รับก็จะทำการตอบรับหรือปฏิเสธ invitation นั้น

การใช้พร็อกซีเซิร์ฟเวอร์นั้นจะเริ่มต้นโดยใช้แอดเดรส เช่น SIP:bob@b.com เพื่อติดต่อกับ Bob แต่ user agent ไม่รู้ว่าจะส่งไปยังไงเลยส่งไปที่พร็อกซีเซิร์ฟเวอร์ของบริษัท a พร็อกซีเซิร์ฟเวอร์จะรู้ว่าแอดเดรสที่ให้มาอยู่คนละบริษัท มันจะทำการค้นหาไปที่พร็อกซีเซิร์ฟเวอร์ของ b ซึ่งมันจะรู้ว่าแอดเดรสที่ส่งมาอยู่ที่บริษัทมัน หรือยัง โดยใช้เรคคอร์ดของดีเอ็นเอสเซิร์ฟเวอร์ในการค้นหา พร็อกซีเซิร์ฟเวอร์ของ b ก็ได้ ซึ่งมันจะรู้ว่าสามารถติดต่อได้ทางโทรศัพท์ IP 1.2.3.4 จากนั้นก็จะส่ง invitation มาที่นี้ โดยที่ proxy agent จะไปหาตำแหน่งของ Bob จาก Registrar

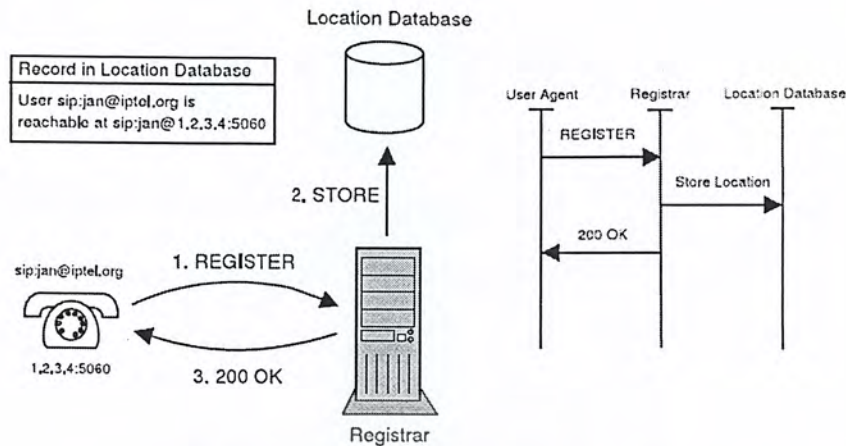


รูปที่ 3-3 การติดต่อระหว่าง user agent โดยผ่าน proxy เซิร์ฟเวอร์

3.2.4 Registrar Server

Registrar จะเป็นส่วนที่เก็บข้อมูลเกี่ยวกับตำแหน่งของผู้ใช้ ได้แก่ ไอพีแอดเดรส หมายเลขพอร์ต Username และเก็บข้อมูลเอาไว้ในฐานข้อมูล ซึ่งพร็อกซีเซิร์ฟเวอร์จะทำการหาข้อมูลปลายทางจาก registrar เมื่อได้รับ invitation โดยที่ registrar จะทำงานร่วมกับ proxy เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

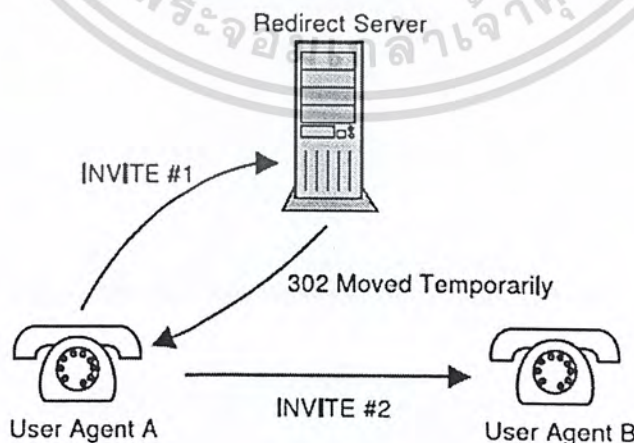


รูปที่ 3-4 จะแสดง ขั้นตอนของการ registration

ในข้อความ REGISTER จะประกอบไปด้วยที่อยู่ที่เกี่ยวข้องในเรคคอร์ด (Address of record) คือ SIP:jan@iptel.org และ ที่อยู่ที่ใช้ในการติดต่อ (contact address) คือ SIP:jan@1.2.3.4:5060 โดยที่ 1.2.3.4 คือ ไอพีแอดเดรสของโทรศัพท์ ซึ่งจะถูส่งไปยัง registrar จากนั้นจะทำการส่งข้อมูลนี้ไปยังฐานข้อมูล เมื่อเสร็จเรียบร้อย registrar จะส่งข้อความ response 200 ok กลับไปยังโทรศัพท์ โดยที่การรีจิสเตอร์แต่ละครั้งจะมีอายุของมัน โดยที่ user agent จะต้องทำการรีจิสเตอร์ใหม่ก่อนที่มันจะหมดอายุ

3.2.5 Redirect Server

เป็นส่วนที่จะได้รับ request และทำการตอบกลับไปด้วยตำแหน่งของ user ในปัจจุบัน เมื่อมันได้รับ request มันจะไปดูข้อมูลในฐานข้อมูลที่ถูกสร้างโดย registrar แล้วทำการสร้างรายการของตำแหน่งของ user และส่งกลับไปยังคนที่ส่ง Request มา ด้วย 3xx class โดยที่ตัวที่ส่ง request ไปจะทำการคัดลอกรายการของปลายทางไว้และให้ request อื่นอื่นส่งไปหามันโดยตรงเลย

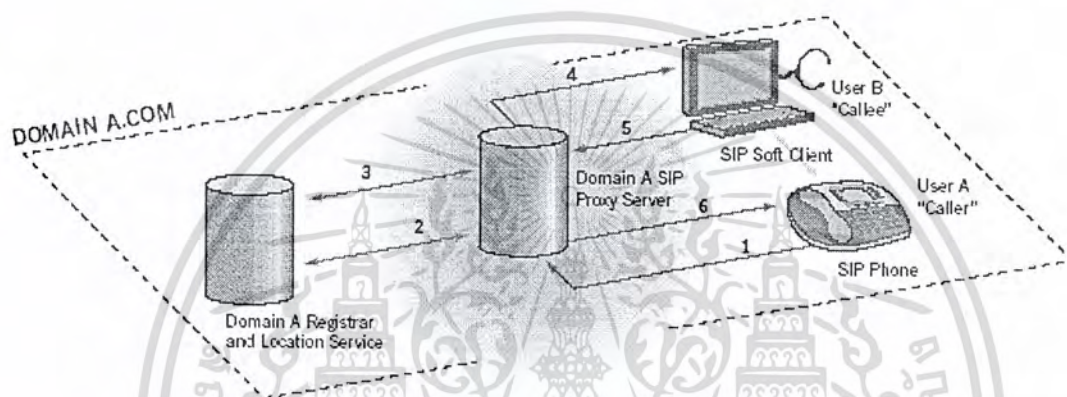


รูปที่ 3-5 การทำงานของ ซิป Redirection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การสร้างเซสชันในโดเมนเดียวกัน

ให้ User A ใช้ SIP Phone ส่วน User B ใช้แอปพลิเคชันบนเครื่องคอมพิวเตอร์ที่รองรับการส่งข้อมูลเสียง วิดีโอ ซึ่งทั้งคู่ได้บอกไอพีแอดเดรสและขนาดข้อมูลที่จะเพียงพอกับความต้องการกับพร็อกซีเซิร์ฟเวอร์ โดยเริ่มแรก User A จะบอกกับพร็อกซีเซิร์ฟเวอร์ว่าต้องการติดต่อกับ User B โดยตัวมันจะไปถามไอพีแอดเดรสของ B จากรีจิสทราเซิร์ฟเวอร์ โดยที่พร็อกซีเซิร์ฟเวอร์ จะทำการถาม B เกี่ยวกับการใช้ SDP รวมถึงตัวกลางที่ A ต้องการที่จะใช้ ซึ่ง B จะบอกกับพร็อกซีเซิร์ฟเวอร์ว่าได้รับคำเชิญ (Invitation) ของ A แล้ว ซึ่งพร็อกซีเซิร์ฟเวอร์ก็จะส่งกลับไปบอก A แล้วทำการสร้างการเชื่อมต่อเซสชันระหว่าง A กับ B ซึ่งผู้ใช้จะสร้างการเชื่อมต่อแบบจุดต่อจุดในการสื่อสาร



รูปที่ 3-6 การสร้างเซสชันในโดเมนเดียวกัน

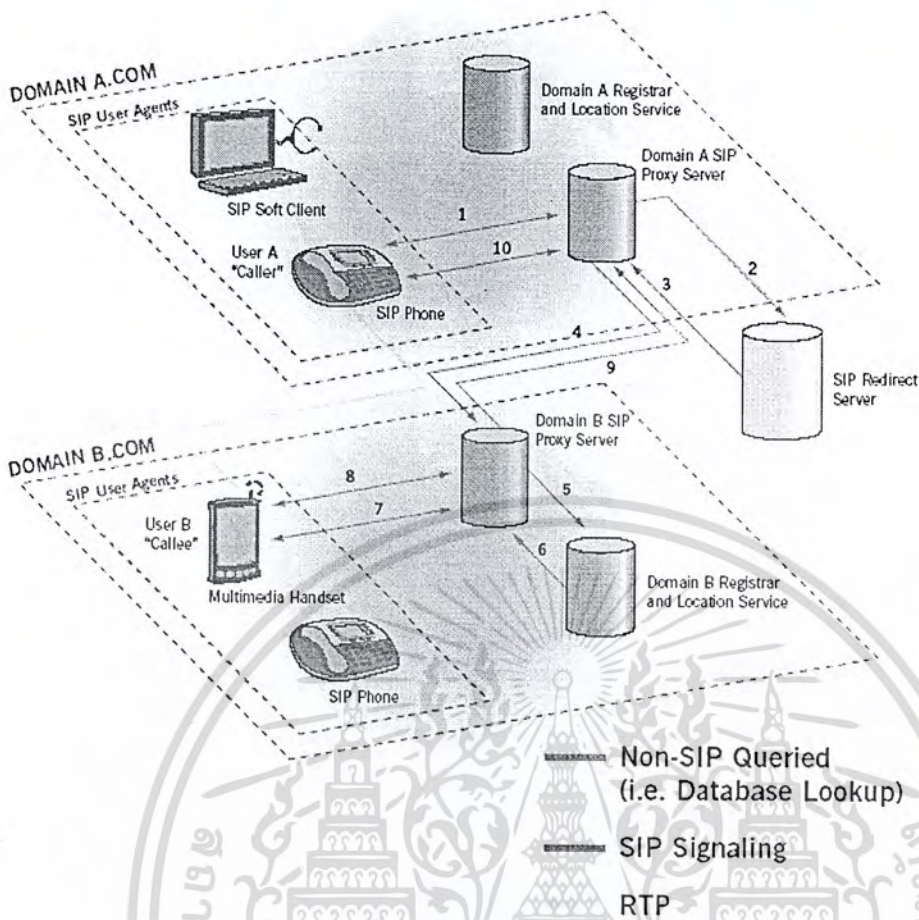
ขั้นตอนการทำงาน

1. user A ทำการติดต่อไปยัง user B โดยจะบอกผ่านไปทางพร็อกซีเซิร์ฟเวอร์
2. พร็อกซีเซิร์ฟเวอร์จะทำการถามแอดเดรสของ B จาก รีจิสทราเซิร์ฟเวอร์
3. รีจิสทราเซิร์ฟเวอร์ ก็จะทำการบอกแอดเดรสของ B ตอบกลับไปให้กับพร็อกซีเซิร์ฟเวอร์
4. พร็อกซีเซิร์ฟเวอร์ก็จะส่งไปบอกกับ User B ว่า User A ต้องการที่จะติดต่อด้วย
5. User B ก็จะตอบรับกลับไปบอกกับพร็อกซีเซิร์ฟเวอร์
6. พร็อกซีเซิร์ฟเวอร์ก็จะตอบกลับไปบอก User A
7. User A จะทำการเชื่อมต่อกับ User B โดยจะเป็นการเชื่อมต่อแบบ RTP

3.4 การสร้างเซสชันที่ต่างโดเมนกัน

User A เชิญ (Invite) B ซึ่งพร็อกซีเซิร์ฟเวอร์ใน โดเมน A รู้ได้ว่า B อยู่นอกโดเมนของมัน มันจะทำการสอบถามไปยังรีไจเรกเซิร์ฟเวอร์ (ซึ่งอยู่ได้ทั้งใน A และ B) เพื่อถามไอพีแอดเดรสของ B จากนั้นรีไจเรกเซิร์ฟเวอร์ จะส่งข้อมูลของ B กลับไปยัง พร็อกซีเซิร์ฟเวอร์ แล้วจะทำการส่งคำเชิญไปยัง พร็อกซีเซิร์ฟเวอร์ในโดเมน B ซึ่งจะส่งคำเชิญของ A ไปให้ยัง B และ B จะตอบรับกลับไปตามเส้นทางเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-7 การสร้างเซสชันที่ต่างโดเมนกัน

ขั้นตอนในการทำงาน

1. User A จะทำการติดต่อไปหา User โดยติดต่อไปทางพร็อกซีเซิร์ฟเวอร์
2. พร็อกซีเซิร์ฟเวอร์จะทำการถามรีไจสตรเซิร์ฟเวอร์ว่าจะไปยังโดเมน B ยังไง
3. รีไจสตรเซิร์ฟเวอร์จะบอกแอดเดรสของพร็อกซีเซิร์ฟเวอร์ที่คุมโดเมน B มา
4. พร็อกซีเซิร์ฟเวอร์ของโดเมน A จะบอกว่าต้องการที่จะติดต่อกับ User B ไปพร็อกซีเซิร์ฟเวอร์ของโดเมน B
5. พร็อกซีเซิร์ฟเวอร์ของโดเมน B จะไปตามแอดเดรสของ B จากรีจิสตราของโดเมน B
6. รีจิสตราจะบอกแอดเดรสของ B ไปให้กับพร็อกซีเซิร์ฟเวอร์
7. พร็อกซีเซิร์ฟเวอร์บอก User B ว่า User A ต้องการจะติดต่อด้วย
8. User B จะตอบกลับไปบอกพร็อกซีเซิร์ฟเวอร์ในโดเมน B
9. พร็อกซีเซิร์ฟเวอร์ในโดเมน B จะตอบกลับไปบอก พร็อกซีเซิร์ฟเวอร์ในโดเมน A
10. พร็อกซีเซิร์ฟเวอร์ในโดเมน A จะตอบกลับไปบอก User A
11. User A จะทำการเชื่อมต่อกับ User B โดยการเชื่อมต่อจะเป็นแบบ RTP

3.5 ชื่อและแอดเดรส (Addressing & Naming)

ในระบบขีพการส่งข้อมูลระหว่างกันจะต้องระบุ SIP URL เพื่อใช้อ้างอิงถึงผู้ใช้ SIP URL จะประกอบด้วยแอดเดรส รูปแบบของแอดเดรสจะอยู่ในรูปของ name@domain โดยอาจจะเป็น user@domain user@address phone-number@gateway และ user@host แอดเดรสนี้จะถูกใช้อ้างอิงถึงผู้ใช้ทั้งผู้เรียกและผู้ถูกเรียกในการส่งข้อความ ตัวอย่างของ SIP URL เช่น SIP://j.doe@example.com โดยที่ URL นี้จะอยู่ในส่วนของเฮดเดอร์ในการส่งข้อความไปยัง SIP URL ที่ระบุไว้จะต้องมีการแปลงแอดเดรสที่อยู่ในของ user@host โดยอาจจะเป็นการแปลงมากกว่าหนึ่งครั้งจนกระทั่งได้ตำแหน่งที่อยู่ของผู้ใช้ ในการแปลงแอดเดรสอาจใช้ DNS (Domain Name Service) หรือ LDAP (Lightweight Directory Access Protocol)

3.6 SIP Message

การสื่อสารโดยใช้ SIP จะเป็นการส่งข้อความที่เป็นชุดๆ ซึ่งปกติแล้วจะแยกส่งเป็นแต่ละ UDP datagram โดยที่แต่ละข้อความจะประกอบไปด้วย first line, message header, message body โดยที่ first line จะระบุชนิดของข้อความว่าเป็น request หรือ response โดยที่ request จะใช้เพื่อเป็นการแสดงการเริ่มต้นของเหตุการณ์บางอย่าง ส่วน reply จะเป็นการยืนยันว่า request นั้น ได้รับและได้ถูก process แล้ว

ตัวอย่าง Code ที่ใช้ในการส่ง request

```
INVITE SIP:7170@iptel.org SIP/2.0
Via: SIP/2.0/UDP 195.37.77.100:5040;rport
Max-Forwards: 10
From: "jiri" <SIP:jiri@iptel.org>;tag=76ff7a07-c091-4192-84a0-d56e91fe104f
To: <SIP:jiri@bat.iptel.org>
Call-ID: d10815e0-bf17-4afa-8412-d9130a793d96@213.20.128.35
CSeq: 2 INVITE
Contact: <SIP:213.20.128.35:9315>
User-Agent: Windows RTC/1.0
Proxy-Authorization: Digest username="jiri", realm="iptel.org",
algorithm="MD5", uri="SIP:jiri@bat.iptel.org",
nonce="3cef75390000001771328f5ae1b8b7f0d742da1feb5753c",
response="53fe98db10e1074
b03b3e06438bda70f"
Content-Type: application/sdp
Content-Length: 451
v=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

o=jku2 0 0 IN IP4 213.20.128.35
s=session
c=IN IP4 213.20.128.35
b=CT:1000
t=0 0
m=audio 54742 RTP/AVP 97 111 112 6 0 8 4 5 3 101
a=rtpmap:97 red/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:6 DVI4/16000
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=rtpmap: 3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16

```

- บรรทัดแรกแสดงถึง invite message ที่ถูกใช้ในการสร้างเซสชันโดยที่ SIP:7170@iptel.org เรียกว่า Request URI ซึ่งจะบอกถึง URI ของฮอปต่อไป (net hop) ของข้อความ
- request สามารถมี via header (ที่เก็บ path ของ request) มากกว่า 1 อันได้ โดยเอาไว้ใช้ให้ response สามารถกลับไปทางเดิมได้ INVITE message จะมี via header แค่อันเดียว ซึ่งถูกสร้างขึ้นจาก user agent ที่ส่ง request มา โดยจากตัวอย่าง เราจะรู้ว่ามันรันอยู่บน host 195.37.77.100 port 504D
- from และ to header จะระบุถึงผู้ส่งและผู้รับข้อความโดยที่ส่วนของ From header จะมีส่วนของ tag parameter ซึ่งจะทำหน้าที่เป็น dialog identifier
- call id จะเป็น dialog identifier ซึ่งจะเป็นตัวจำแนกข้อความว่ามาจากการติดต่อครั้งเดียวกัน โดยจะมีค่าไม่เหมือนกัน ยกเว้น registration request ที่จะมีค่าเหมือนกัน
- CSeq ใช้สำหรับเก็บลำดับของคำสั่งเอาไว้ เพราะอาจจะต้องทำการส่งใหม่ในตัวกลางที่มีความไม่แน่นอนได้ ซึ่งต้องระบุในข้อความซึ่งผู้รับสามารถที่จะให้ทำการส่งข้อความนั้นใหม่หรือ out of order request ได้ โดยมันจะมีค่าเพิ่มขึ้นในแต่ละทุกๆ request ยกเว้น ACK และ CANCEL
- contact header จะระบุไปที่แอดเดรสและ port ของผู้ส่งที่ผู้รับจะต้องตอบกลับมา
- message header จะถูกกำหนดโดย message body โดย empty line โดยที่ body message ของ INVITE request จะประกอบไปด้วยรายละเอียดชนิดของมีเดียที่ยอมรับโดยผู้ส่งและเข้ารหัสใน SDP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 SIP Request

SIP Request นั้นมี method ต่างๆซึ่งอยู่ใน header ดังนี้

ACK - เป็นข้อความสุดท้ายของการ INVITE เพื่อเป็นการบอกผู้ถูกเรียกได้ว่า ผู้เรียกได้รับ response จากผู้ถูกเรียกแล้ว

BYE - ใช้ในการสิ้นสุดเซสชัน โดยที่มันจะทำการส่งไปบอกสมาชิกคนอื่นในการเชื่อมต่อนั้น

CANCEL - จะใช้ในการยกเลิกการสร้างเซสชันที่ยังไม่เสร็จสมบูรณ์ โดยจะใช้เมื่อผู้รับยังไม่ได้ทำการตอบ final response และผู้ส่งต้องการที่จะยกเลิกการสื่อสาร อาจถูกเรียกใช้โดย User agent หรือ Proxy server

REGISTER - เพื่อเอาไว้ใช้ให้ registrar รู้ตำแหน่งปัจจุบันของ user

OPTION - ใช้ในการสอบถามความสามารถและสถานะของ user agent หรือ server

REFER - User agent ใช้ในการร้องขอให้ user agent อื่นติดต่อไปยัง URL ที่

SUBSCRIBE - User agent ใช้เมธอด SUBSCRIBE ในการร้องขอการเป็นสมาชิกของการตรวจสอบการรับข้อมูลในเหตุการณ์หนึ่งๆ (โดยเมธอด NOTIFY)

NOTIFY - User agent ใช้เมธอด NOTIFY ในการตรวจจับเหตุการณ์ที่จะเกิดขึ้น

MESSAGE - ใช้ในการส่งข้อความ (IM : Instant Message) โดย SIP

INFO - User agent ใช้เมธอดนี้ในการส่งข้อมูลของสัญญาณการเรียก (call signaling information) ถึง user agent อีกฝั่งหนึ่งที่ทำกรติดต่อด้วย

PRACK - ใช้ในการตอบกลับต่อ response ที่มีความน่าเชื่อถือ (reliably transported provisional responses)

UPDATE - ใช้ในการปรับเปลี่ยนสถานะของการติดต่อสื่อสาร โดยที่ไม่ต้องเปลี่ยนสถานะของไดอะล็อก

3.8 SIP Response

เมื่อ User agent หรือ พร็อกซีเซิร์ฟเวอร์ ได้รับ request มาจะต้องทำการ reply กลับไปทุก request ยกเว้น ACK request

ตัวอย่างของ Reply

SIP/2.0 200 OK

Via: SIP/2.0/UDP 192.168.1.30:5060;received=66.87.48.68

From: SIP:SIP2@iptel.org

To: SIP:SIP2@iptel.org;tag=794fe65c16edfdf45da4fc39a5d2867c.b713

Call-ID: 2443936363@192.168.1.30

CSeq: 63629 REGISTER

Contact: <SIP:SIP2@66.87.48.68:5060;transport=udp>;q=0.00;expires=120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Server: SIP Express router (0.8.11pre21xrc (i386/linux))
Content-Length: 0
Warning: 392 195.37.77.101:5060 "Noisy feedback tells:
pid=5110 req_src_ip=66.87.48.68 req_src_port=5060 in_uri=SIP:iptel.org
out_uri=SIP:iptel.org via_cnt==1"

```

response จะคล้ายกับ request ยกเว้นบรรทัดแรกที่ประกอบไปด้วย protocol version, reply code, reason phrase

reply code จะเป็นตัวเลขตั้งแต่ 100-699 ที่เป็นตัวเลขระบุ ชนิดของ response มีด้วยกันทั้งหมด 6 class

- 1xx จะเป็น provisional response ซึ่งจะเป็นตัวบอกว่า request ที่ส่งไปได้รับแล้ว แต่ยังไม่ทราบผลของกระบวนการ ซึ่ง response นี้จะส่งก็ต่อเมื่อ เป็นกระบวนการที่ไม่เสร็จในทันที ซึ่งปกติแล้ว proxy เซิร์ฟเวอร์ จะส่ง response เป็น 100 เมื่อเริ่มส่ง INVITE ละ user agent จะส่งกลับด้วย code 180 ซึ่งหมายความว่า โทรศัพท์ของผู้รับกำลังดัง
- 2xx จะเป็น positive final response จะเป็น response สุดท้ายที่ผู้ส่ง request จะได้รับ ซึ่ง final response ก็จะเป็นการ terminate transaction ด้วย โดยที่ จะมีความหมายว่า request ได้จัดการเสร็จและเป็นที่ยอมรับแล้ว เช่น 200 OK จะส่งเมื่อ user ได้ตกลงรับ INVITE request
- 3xx จะถูกใช้เมื่อต้องการ redirect ไปยังผู้ส่ง โดย redirect response จะบอกเกี่ยวกับตำแหน่งใหม่ของผู้ หรือบริการที่ผู้ส่งอาจเพื่อความพอใจในการติดต่อ ปกติแล้ว redirect response จะถูกส่งจากพร็อกซีเซิร์ฟเวอร์ เมื่อมันไม่สามารถจัดการได้ มันจะส่ง response ไปพร้อมกับตำแหน่งใหม่ ซึ่งอาจจะเป็นตำแหน่งของพร็อกซีตัวอื่น หรือตำแหน่งของผู้รับในปัจจุบันที่ได้จากฐานข้อมูลของ registrar ซึ่งผู้ส่งก็จะทำการส่งไปยังตำแหน่งใหม่อีกครั้ง
- 4xx เป็น negative final response ซึ่งแสดงว่ามีปัญหาทางฝั่งผู้ส่งทำให้ไม่สามารถที่จะทำตาม request ได้ อาจเป็นเพราะผิดไวยากรณ์ (syntax) หรือไม่สามารถทำให้เสร็จได้ที่เซิร์ฟเวอร์นั้น
- 5xx คือมีปัญหาที่ฝั่งเซิร์ฟเวอร์ โดยที่เซิร์ฟเวอร์ไม่สามารถทำให้สำเร็จได้ ซึ่งไคลเอนต์ควรจะส่ง request ใหม่อีกครั้ง
- 6xx คือ request ไม่สามารถที่จะทำได้ไม่ว่าจะเป็นเซิร์ฟเวอร์ไหนก็ตาม ซึ่งมักถูกส่งโดยเซิร์ฟเวอร์ ที่มีข้อมูลที่แน่นอนของ user นั้นๆ เช่น user จะส่ง 603 decline response เมื่อไม่ต้องการเข้าร่วมกับเซสชันนั้น

ซึ่งในบรรทัดแรกของ Response class ควรจะมี reason phrase ซึ่งประกอบด้วยข้อความที่มนุษย์เข้าใจที่จะอธิบายผลของกระบวนการนั้น ซึ่ง user agent ควรจะแสดง reason phrase ให้กับ user ด้วย

3.9 SIP Transaction

ทรานแซกชันเป็นลำดับของการแลกเปลี่ยนข้อความระหว่าง SIP Network Element โดยที่ 1 ทรานแซกชันจะประกอบไปด้วย 1 request และ response ทั้งหมดของ request นั้น ซึ่งรวมไปถึง response

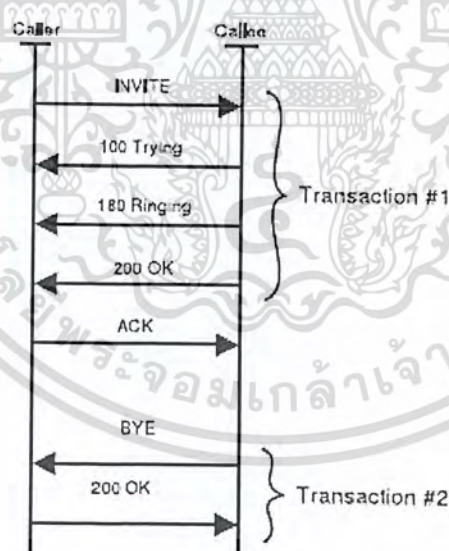
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั่วคราวที่จะมีหรือไม่ก็ได้ และ final response ที่อาจมีมากกว่า 1 ได้ (INVITE อาจถูกตอบกลับมาได้หลาย final response เมื่อพรีอ็อกซี เซิร์ฟเวอร์ เป็นตัว fork request ออกมา

ทรานแซกชันเริ่มต้นด้วย INVITE request ทรานแซกชันนั้นจะรวม ACK ด้วยก็ต่อเมื่อ final response ไม่เท่ากับ 2xx response ถ้าเท่ากับ 2xx response ACK จะไม่ถูกพิจารณาในส่วนของทรานแซกชันนั้น

ACK จะเป็นส่วนของถ้าทรานแซกชันที่มี final response เป็น negative แต่มันจะไม่ได้เป็นส่วนหนึ่งของทรานแซกชันที่มี final response เป็น positive สาเหตุที่ต้องแยกเช่นนี้ก็เพราะว่าการส่ง 200 OK ในทุกๆ ข้อความเป็นสิ่งจำเป็น เพราะ 200 OK ไม่ได้ใช้เฉพาะในการสร้างทรานแซกชันเท่านั้น แต่ 200 OK ยังถูกสร้างจากหลายๆส่วนเมื่อพรีอ็อกซีทำการ fork request ออกไป และผู้ที่รับนั้นจะส่งกลับไปยังผู้ส่ง ดังนั้น ในกรณีนี้ user agent จะต้องทำการ retransmit 200 OK จนกว่าผู้รับจะได้รับ ACK โดยจะมีเฉพาะ response ของ invite เท่านั้น ที่จะถูกส่งใหม่

ส่วนของ SIP ที่มีลักษณะแบบนี้จะเรียกว่า stateful โดยส่วนนี้มักจะสร้าง state ที่เอาไปใช้ในทรานแซกชันซึ่งจะถูกเก็บไว้ในหน่วยความจำสำหรับใช้ในระหว่างการสื่อสาร เมื่อ request หรือ response ได้ถูกส่งมา ส่วนที่เป็น stateful จะพยายามที่จะนำ request หรือ response นั้น เข้าไปในทรานแซกชันที่มีอยู่ โดยจะนำตัวระบุทรานแซกชันจาก response หรือ request ที่ได้มาไปเปรียบเทียบกับตัวระบุของทรานแซกชันที่มีอยู่ทั้งหมด ถ้าทรานแซกชันนั้นมีจริงก็จะทำการอัปเดตจากข้อความนั้น



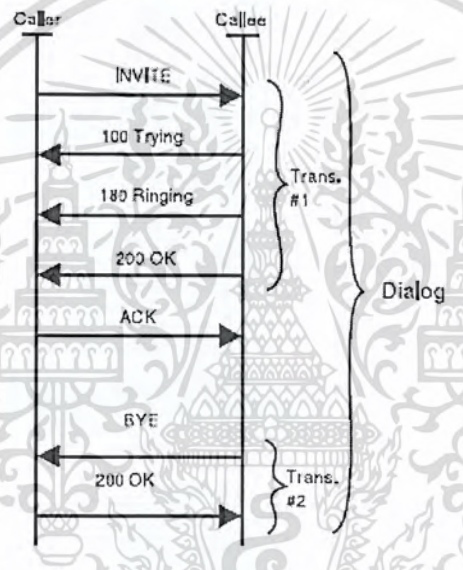
รูปที่ 3-8 การส่งข้อความหากันในการสื่อสารระหว่าง 2 user agent

3.10 SIP Dialogs

ก่อนหน้านั้นได้แสดงให้เห็นว่าทรานแซกชันคืออะไร เช่น ทรานแซกชันที่มี INVITE หรือ ทรานแซกชันที่มี BYE และ response ของมันเมื่อมันทำการตัดขาดการเชื่อมต่อ ซึ่งเรารู้ว่าทรานแซกชันทั้ง 2 อันนี้มีความสัมพันธ์กันบางอย่างซึ่งก็คือ ทรานแซกชันทั้ง 2 นั้นอยู่ในไดอะล็อกเดียวกัน ซึ่งไดอะล็อกนั้นแสดงความสัมพันธ์ของ SIP แบบ peer-to-peer ระหว่าง 2 user agent ซึ่งไดอะล็อกจะคงอยู่ในช่วงเวลาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนึ่ง และเป็นหลักการที่สำคัญระหว่าง user agent ทั้ง 2 ซึ่งไคอะล็อกนั้นจะทำให้ลำดับและการส่งข้อความระหว่าง 2 user agent นั้นง่ายขึ้น

ไคอะล็อกนั้นจะถูกจำแนกโดยการใช้ Call-ID, From tag และ To tag ซึ่งข้อความที่มีตัวระบุเหมือนกันทั้ง 3 อัน จะอยู่ในไคอะล็อกเดียวกัน เราได้แสดงให้เห็นว่า CSeq header field นั้นจะถูกใช้เพื่อที่จะจัดการกับแมสเสจซึ่งในความเป็นจริงแล้ว มันใช้เพื่อที่จะจัดการกับข้อความที่อยู่ภายในไคอะล็อกนั้น ซึ่งจะต้องทำการเพิ่ม CSeq ขึ้นทีละ 1 ในแต่ละข้อความที่ส่งภายในไคอะล็อกมีเช่นนั้น request หรือ retransmission จะถือว่าใช้ไม่ได้ ในความเป็นจริงแล้ว CSeq number จะเป็นตัวจำแนก transaction ภายใน ไคอะล็อกเพราะว่า เรบอกว่ request และ response ที่เกี่ยวข้องจะถูกเรียกว่า ทรานแซกชันซึ่งหมายความว่า จะมีเพียง 1 ทรานแซกชันในแต่ละ direction ที่สามารถใช้ได้ในไคอะล็อก ซึ่งเราสามารถพูดได้ว่าไคอะล็อกนั้นคือลำดับของทรานแซกชัน



รูปที่ 3-9 แสดงถึงแมสเสจของ 1 ไคอะล็อก

บางข้อความนั้นก็สร้างไคอะล็อกขึ้นมา บางข้อความก็จะไม่สร้าง ซึ่งมันแบ่งแยกความสัมพันธ์ของข้อความอย่างชัดเจน และ การส่งข้อความมันจะไม่มีความสัมพันธ์กับข้อความนอกไคอะล็อกซึ่งมันจะต่อการใช้เพราะว่า user agent ไม่ต้องเก็บไคอะล็อก state

ตัวอย่างเช่น INVITE message จะสร้างไคอะล็อกขึ้นมา เพราะว่ามันจะต้องทำการส่ง BYE request เพื่อที่จะยกเลิกเซสชันที่ถูกสร้างขึ้นจาก INVITE โดย BYE จะถูกส่งภายในไคอะล็อกที่ถูกสร้างขึ้นจาก INVITE แต่ถ้า User agent ส่ง MESSAGE request เช่น บอกให้ไม่ต้องสร้างไคอะล็อกจะทำให้ข้อความที่จะส่งต่อไปส่งได้อิสระโดยไม่ต้องสนใจข้อความที่ส่งไปก่อนหน้านี้

3.11 Dialogs Facilitate Routing

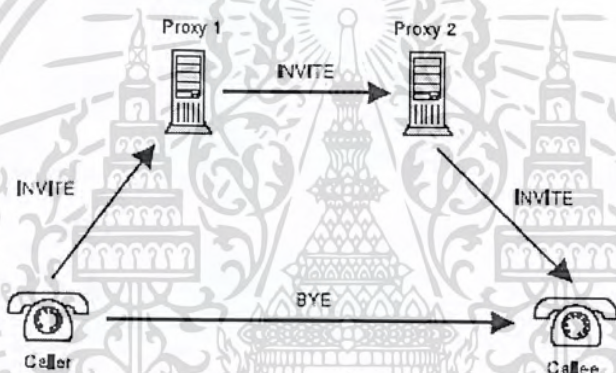
User SIP:bob@a.com ต้องการจะพูดกับ user :pete@b.com ซึ่งเขาจะรู้ว่าแอดเดรสของผู้รับคือ user :pete@b.com แต่แอดเดรสนี้ไม่ได้บอกอะไรเกี่ยวกับ current location ของ user

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ Request จะถูกส่งจากพร็อกซี่ ไปอีกพร็อกซี่ จนกระทั่งเจอพร็อกซี่ที่รู้ตำแหน่งปัจจุบันของผู้รับ ซึ่งกระบวนการนี้เรียกว่า routing เมื่อ request ค้นหาผู้รับเจอ user agent ของผู้รับก็จะทำการส่ง response กลับไปหาผู้ส่ง ซึ่ง user agent ของผู้รับจะทำการใส่ Contact header field ไปใน response ซึ่งจะบอกถึงตำแหน่งปัจจุบันของ user ซึ่ง request ที่ส่งไปครั้งแรกก็จะทำการใส่ Contact header field ซึ่งมีความหมายว่า user agent ของทั้งคู่ ก็จะรู้ตำแหน่งปัจจุบันของกันและกันโดยตรง

เพราะว่า User agent รู้ตำแหน่งของกันและกัน ทำให้ไม่จำเป็นที่จะต้องส่ง request ไปยัง proxy ก่อน ซึ่งมันสามารถที่จะส่งไปจาก user agent ไปหา user agent ได้โดยตรงเลย

ข้อความที่อยู่ในไคอะล็อกจะถูกส่งจาก user agent ไปยัง user agent ได้โดยตรง นี่เป็นการพัฒนาประสิทธิภาพที่น่าสนใจ เพราะ proxy จะไม่เห็นข้อความทุกอันภายในไคอะล็อกซึ่ง proxy จะถูกใช้ในการหาเส้นทางเฉพาะ request ครั้งแรกที่เป็นการสร้างไคอะล็อกซึ่งข้อความที่ส่งโดยตรงเลยจะมี smaller latency เพราะ proxy มักจะใช้ implement โดยใช้ routing logic ที่มีความซับซ้อน



รูปที่ 3-10 แสดงตัวอย่างของข้อความภายในไคอะล็อกที่ถูกส่งโดย proxy

3.12 Dialog Identifiers

Dialog identifier นั้นประกอบไปด้วย 3 ส่วน คือ Call-Id, From tag และ To tag ซึ่ง Call-ID นั้นถูกเรียกว่า call identifier ซึ่งควรเป็น string ที่มีความ unique เพื่อจำแนกการสื่อสาร โดยที่ 1 call อาจจะมีได้ มากกว่า 1 ไคอะล็อกซึ่ง multiple user agent อาจจะทำ response เมื่อ request ได้ถูกพร็อกซี่ทำการ fork โดยที่ user agent แต่ละตัว จะส่ง 2xx สร้างไคอะล็อกไปยังผู้ส่ง ซึ่งทุกไคอะล็อกนั้นจะเป็นส่วนหนึ่งของการสื่อสารเดียวกัน และมี Call-ID อันเดียวกัน

From tag จะถูกสร้าง โดยผู้ส่ง และมันจะจำแนกไคอะล็อกใน user agent ของผู้ส่ง

To tag จะถูกสร้างโดยผู้รับ และมันจะจำแนกไคอะล็อกใน user agent ของผู้รับ

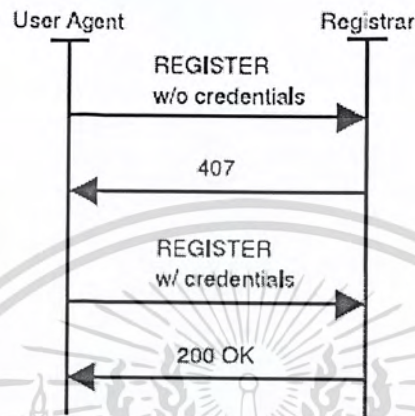
Hierarchical dialog identifier นั้นมีความจำเป็นเพราะ single call invitation นั้นสามารถที่จะสร้างไคอะล็อกขึ้นมาได้หลายอันและผู้ส่งก็จำเป็นที่จะต้องรู้จักพวกมันทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.13 Typical SIP Scenarios

3.13.1 Registration

User จำเป็นที่จะต้องลงทะเบียนตัวเองเพื่อให้ user คนอื่นสามารถที่จะเข้าถึงได้ ซึ่งเมื่อทำการ registration แล้วสำเร็จรีจิสตราจะส่งข้อความตอบกลับมาว่า 200 OK แต่การลงทะเบียนอาจได้รับ reply 407 ได้ถ้า user ไม่มีหลักฐานที่มีอยู่จริงในการใช้อ้าง

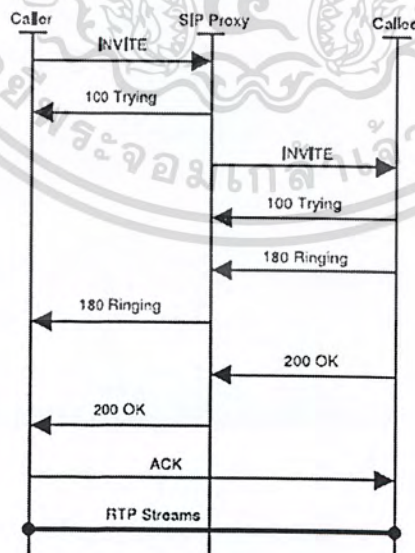


รูปที่ 3-11 ลำดับการส่ง REGISTER message

3.13.2 Session Invitation

INVITE request ปกติจะส่งไปยังพร็อกซี่ซึ่งจะตอบกลับมาด้วย 100 Trying ทันที เพื่อที่จะหยุดการ retransmission และส่ง request ต่อไป

Provisional response จะถูกสร้างโดยผู้รับเพื่อจะส่งกลับไปหาผู้ส่ง เช่น 180 Ringing response ซึ่งมันจะถูกสร้างขึ้นเมื่อโทรศัพท์ของผู้รับเริ่มที่จะดัง



รูปที่ 3-12 ลำดับการส่ง INVITE message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

200 OK จะถูกสร้างเมื่อผู้รับได้รับสายและ user agent จะทำการส่งข้อความใหม่จนกระทั่งมันได้รับ ACKจากผู้ส่ง ซึ่งเซสชันจะถูกสร้างขึ้นตรงนี้

3.13.3 Session Termination

Session termination จะสำเร็จโดยการส่ง BYE request ไปให้กับ user คนอื่นๆโดยตรง โดยส่งจาก user agent ยกเว้นพร็อกซีที่อยู่บนเส้นทาง (path) ของ INVITE request

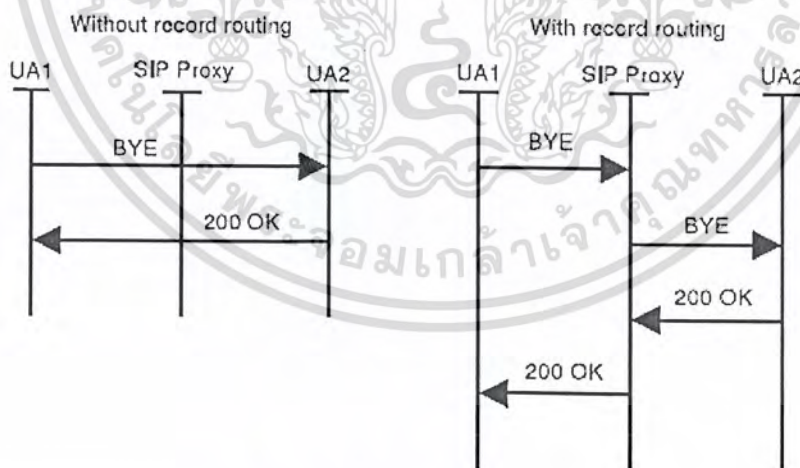
ผู้ที่ต้องการจะยกเลิกเซสชัน จะส่ง BYE request ไปยังเครื่องอื่นที่อยู่ในเซสชันเดียวกัน ซึ่งเครื่องอื่นก็จะส่ง 200 OK เพื่อยืนยัน BYE และทำการยกเลิกเซสชัน

3.13.4 Record Routing

Request ทั้งหมดที่อยู่ในไดอะล็อกจะถูกส่งไปโดยตรงจาก user agent หนึ่ง ไปยังตัวอื่นๆ ยกเว้น request ที่อยู่นอกไดอะล็อกจะเดินทางผ่านพร็อกซีด้วยวิธีการนี้ทำให้เครือข่าย SIP สามารถขยายขนาดเครือข่ายได้มากขึ้นเพราะว่าไม่จำเป็นว่าทุกข้อความจะต้องผ่านพร็อกซี

Record routing จะเป็นตัวคอยที่จะแจ้งให้ user ทราบว่าจำเป็นที่จะต้องผ่านทางพร็อกซีโดยที่พร็อกซีจะทำการใส่ Record-Route header ลงไปในข้อความ ดังนั้นข้อความที่ถูกส่งในไดอะล็อกก็จะส่งผ่าน พร็อกซีที่ได้กำหนดเอาไว้ใน Record-Route header

ผู้รับเมื่อได้รับ Record-route header แล้วผู้รับจะต้องทำการส่ง Record-route header field ลงไปใน response ด้วยเพราะผู้ส่งจำเป็นที่จะต้องทราบข้อมูลนั้นเช่นกัน



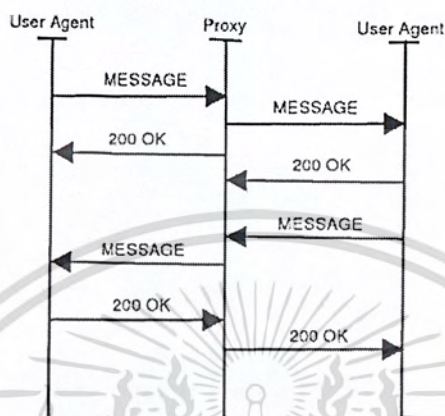
รูปที่ 3-13 ลำดับการส่ง BYE message

รูปซ้ายมือจะเป็นการส่ง BYE ไปให้ User agent คนอื่นโดยที่ไม่มี Record-route header อยู่ในข้อความ ส่วนรูปขวามือจะเป็นการส่งที่พร็อกซี ได้ทำการใส่ Record-route header ลงไปในข้อความแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.13.5 Instant Message

จะถูกใช้โดยการส่งข้อความ request โดยมันจะไม่มีโครงสร้างใดอะล็อกขึ้นมา ดังนั้น มันจะใช้พรีอ็อกซี ตัวเดียวกันตลอด โดยรูปนี้จะเป็นรูปแบบทั่วไปในการส่ง instant message โดยที่ข้อความของ instant message จะถูกส่งไปใน ส่วน body ของ request



รูปที่ 3-14 ลำดับการส่งข้อความติดต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

Real Time Communication API

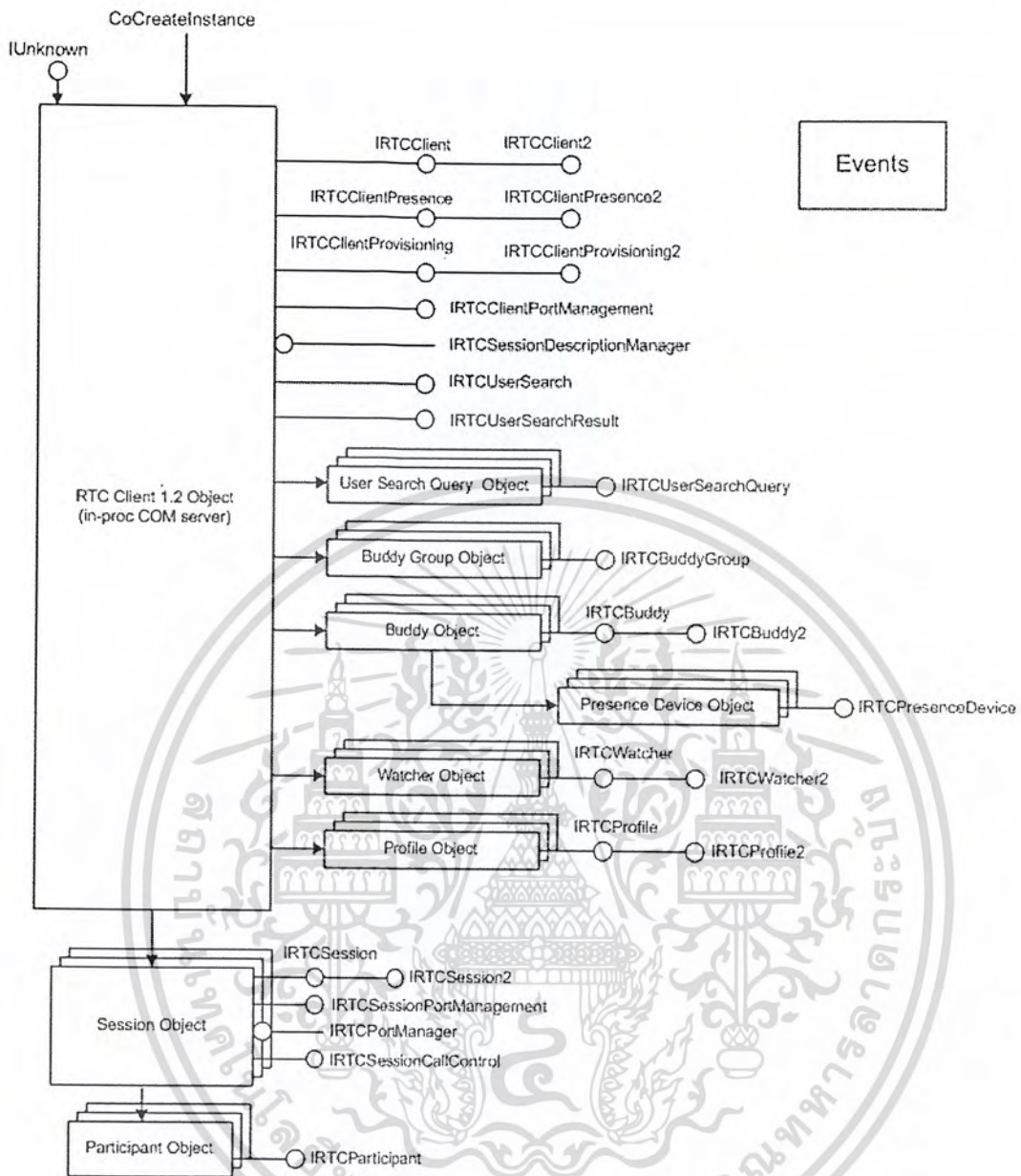
4.1 Real-time Communication Client Application Programming Interface

RTC (Real-time Communication) Client API สามารถให้คนพัฒนาเพื่อสร้างแอปพลิเคชัน เพื่อที่จะสร้างการติดต่อสื่อสาร โดยแอปพลิเคชันนั้นถูกพัฒนาให้เข้าถึง PC เพื่อให้กลายเป็นศูนย์กลาง สำหรับการติดต่อสื่อสารทางธุรกิจด้วยเสียง วีดีโอ และ ข้อความ ได้รวมกันเข้าเป็นการเชื่อมต่อเพียงอันเดียวบนพีซี การเชื่อมต่อระหว่าง PC กับ PC ผู้ใช้สามารถที่จะสร้างการติดต่อแบบ PC-to-phone, phone-to-phone สามารถส่งข้อความที่เป็นตัวอักษรได้ การแชร์แอปพลิเคชัน (application sharing) และ whiteboard สามารถใช้ได้บนการเชื่อมต่อระหว่าง PC กับ PC ส่วน Presence information นั้นเป็นบริการในการติดต่อสื่อสารของคอมพิวเตอร์ซึ่งอนุญาตผู้ใช้สามารถติดต่อกับ contact ผ่านทาง registrar server ซึ่งเก็บข้อมูลของตำแหน่งปัจจุบันบน contact ไว้ ซึ่งตำแหน่งนั้นอาจจะเป็น PC หรือโทรศัพท์ก็ได้โดยในอนาคตอาจจะเป็นมือถือก็ได้ ยกตัวอย่างเช่น ถ้าเราติดต่อเพื่อนโดยที่ presence information ได้ระบุว่าเป็น PC ที่บ้านของเธอมันก็จะทำการติดต่อไปให้เองเลย โดยผู้ใช้เก็บรักษาข้อมูลส่วนตัวโดยการ block การติดต่อจาก presence information ได้

การเชื่อมต่อโดยใช้ whiteboard สามารถที่จะเพิ่มการเชื่อมต่อเพื่อที่จะเอาไว้แชร์ภาพหรือข้อมูลที่มองเห็นได้ในงานสนทนา การติดต่อโดยใช้เสียงไปยังช่างถ่ายภาพเพื่อที่จะขอรูปภาพสามารถทำได้ขึ้นได้โดยการใช้ application sharing โดยช่างถ่ายภาพจะเอาภาพและตัวลูกค้ารวมเข้าไปในการเชื่อมต่อที่เกี่ยวข้องกันโดยการติดต่อที่ผ่านทางคอมพิวเตอร์ด้วย presence information สามารถเป็นการเชื่อมต่อที่ผิดพลาดน้อย ในขณะที่ทำการแชร์แอปพลิเคชันและ whiteboard สามารถช่วยให้ประหยัดเวลาและประสิทธิภาพของการเชื่อมต่อ ส่วนบริการด้านการส่งข้อความ (IM service) ปัจจุบันได้ถูกใช้โดย MSN และ AOL

4.2 RTC Version 1.2 Object and Interface

โคแอสแกรมนี้แสดงความสัมพันธ์ระหว่าง RTC Client API version 1.0 และ 1.2 object และ interface โดยสิ่งที่มีใหม่ใน version 1.2 นั้นคือ BuddyGroup, Presence Device, User Search Query object



รูปที่ 4-1 แสดงถึง object และ interface ของ RTC Version 1.2

4.2.1 Client Object

อ็อบเจกต์นี้ สำหรับ IRTCCClient interface เป็น RTC version 1.0 ซึ่งถูกสร้างโดยการเรียกใช้ CoCreateInstance ด้วย CLSID_IRTCCClient (GUID = {7a42ea29-a2b7-40c4-b091-f6f024aa89be}) ส่วน RTC Client2 ใน version 1.2 API ซึ่งถูกสร้างโดยการเรียก QueryInterface บน Client object โดยที่ใน RTC Client2 นั้นได้เพิ่มอินเทอร์เฟซและเมธอดเพื่อที่จะระบุว่าการเชื่อมต่อที่เข้ามานั้นจะถูกปฏิบัติอย่างไรในแต่ละชนิดของการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 BuddyGroup Object

โดยที่อ็อบเจกต์นี้เป็นอ็อบเจกต์หลักในการสร้างกลุ่มของผู้ที่ติดต่อ ซึ่ง IRTCBuddyGroup interface นั้นจะบน Buddy Object และสามารถได้มาจาก IRTCCliantPresence2::AddGroup

4.2.3 User Search Query Object

อ็อบเจกต์นี้จะอนุญาตให้ผู้ใช้ทำการค้นหาพารามิเตอร์เพื่อค้นหาผู้ใช้คนอื่นบนอินเทอร์เน็ตโดยที่อ็อบเจกต์นี้มี IRTCUserSearchQuery interface ที่ถูกสร้างโดย IRTCUserSearch::CreateQuery

4.2.4 Presence Device Object

อ็อบเจกต์นี้จะดึงข้อมูลบน presence device เพื่อที่จะระบุผู้ที่เราติดต่อ ซึ่ง status, notes, presence property, device data นั้นจะอยู่บน presence device โดยที่ IRTCPresenceDevice interface สามารถได้จาก IRTCEnumPresenceDevices::Next

4.2.5 Session Object

อ็อบเจกต์นี้จะสนับสนุนชนิดการเชื่อมต่อทุกรูปแบบและเป็นองค์ประกอบที่ใช้สร้างหรือรับการติดต่อโดย เสียง วิดีโอ และข้อความ ซึ่งข้อมูลการเชื่อมต่อเช่น state media type participant information สามารถเอามาได้จาก session interface โดยที่เมธอด AddParticipant นั้นสร้าง Participant object และสร้างการเชื่อมต่อโดย เสียง วิดีโอ หรือข้อความ โดย IRTCSession interface มีรายการเมธอด ที่เหมาะที่จะใช้อยู่

4.2.6 Participant Object

ผู้ที่มีส่วนร่วมจะถูกเพิ่มเข้าไปในการเชื่อมต่อโดย IRTCSession::AssParticipant โดยที่ multiple participant สามารถถูกเพิ่มเข้าไปได้เฉพาะรูปแบบการเชื่อมต่อระหว่างโทรศัพท์กับโทรศัพท์ และ multiparty IM session

4.2.7 Buddy Object

มันจะเก็บหรือดึงเอาข้อมูลของ contact ออกมา โดยที่รายการของผู้ที่เราติดต่อด้วยจะถูกสร้างโดย IRTCCliantPresence::AddBuddy โดย IRTCBuddy2 นั้นจะเพิ่มเมธอดเพื่อ refresh buddy subscriptions retrieve buddy groups และ ส่งค่า presence profile

4.2.8 Watcher Object

มันจะเก็บข้อมูลและนำ watcher state แสดงขึ้นมา โดย watcher นั้นจะถูกสร้างโดยการใช้ IRTCCliantPresence::AddWatcher ซึ่ง IRTCWatcher2 interface นั้นจะเพิ่มเมธอดเพื่อ refresh buddy subscriptions retrieve buddy groups และ ส่งค่า presence profile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.9 Profile Object

มันจะแสดง interface เพื่อที่ค้นหาพารามิเตอร์เพื่อเก็บเอาไว้ในโพรไฟล์ โดย IRTCPProfile interface นั้นจะเก็บรายการของเมธอดที่เหมาะสมเอาไว้ โดย profile2 object นั้นจะเพิ่มเมธอดเพื่อแสดงการยืนยันตน และ realm สำหรับโพรไฟล์เอาไว้

4.3 ความต้องการพื้นฐานของ RTC (RTC General Requirement)

4.3.1 แนวคิดพื้นฐาน

4.3.1.1 Session การเชื่อมต่อการสื่อสารแบบ real-time จะสร้างการสนทนาระหว่างผู้ใช้อย่างน้อย 2 คนรวมถึงชนิดของตัวกลางอย่างน้อย 1 ตัว โดย ซิพเซิร์ฟเวอร์ ในการค้นหาเส้นทางและต้องใช้ การเชื่อมต่อแบบ peer-to-peer ในตอนที่ไม่ได้ใช้ซิพเซิร์ฟเวอร์

4.3.1.2 ซิพเซิร์ฟเวอร์ นั้นจะช่วยในการค้นหาผู้ใช้คนอื่นๆและสร้างการเชื่อมต่อแบบเรียลไทม์ และยังช่วยให้ผู้ใช้ได้รับความปลอดภัยมากขึ้น จัดการกับ presence information และ contact list สำหรับผู้ใช้ โดย ซิพเซิร์ฟเวอร์สามารถทำตัวเป็น SIP proxy ซึ่งจะทำหน้าที่ค้นหาเส้นทางและทำการส่งข้อความกลับไปยัง SIP client มันยังทำหน้าที่เก็บฐานข้อมูลของตำแหน่งของผู้ใช้ในเครือข่าย และมันทำตัวเหมือนเป็น presence Agent Server (PAS) ซึ่งจะเก็บข้อมูลของผู้ใช้ทั้งหมดที่เข้ามา และจะประกาศให้กับผู้ใช้คนอื่นๆทราบเมื่อมีการเปลี่ยนแปลง presence information

4.3.1.3 Contact list คือรายชื่อของคนที่ต้องการจะติดต่อด้วย

4.3.1.4 Roaming ผู้ใช้แต่ละคนควรที่จะเข้าถึงข้อมูล เช่น contact list กฎของการพิสูจน์ตน เป็นต้น โดย RTC Client API นั้น อนุญาตให้อุปกรณ์เก็บข้อมูล account ของผู้ใช้เอาไว้ที่เซิร์ฟเวอร์และอนุญาตให้ผู้ใช้สามารถที่จะเข้าถึงข้อมูลเหล่านั้นได้

4.3.2 ชนิดของการเชื่อมต่อที่ใช้โดย RTC Client API

การเชื่อมนั้นสามารถที่จะแบ่งได้ดังนี้

- Instant Message สิ่งหลัก IM ก็เป็นการติดต่อสื่อสารระหว่างบุคคลซึ่งเราสามารถที่จะข้อความสั้นๆ ซึ่งอาจส่งให้กับคนอื่นได้มากกว่า 1 คน ซึ่งเป็นวิธีการที่มีประสิทธิภาพมากกว่าอีเมลอีก
- Audio และ video ถึงแม้จะไม่ถูกรบกวนโดยเซิร์ฟเวอร์ แต่มันก็ยังให้รายละเอียดในการติดต่อสื่อสารไม่มากพอ ซึ่ง RTC Client API จะมีความสามารถที่จะใช้การเชื่อมต่อแบบเสียงหรือวีดีโอ ซึ่งผู้ใช้จะสามารถได้ยินเสียงของกันและกันได้
- Telephony โทรศัพท์ส่วนใหญ่ติดต่อกับ PSTN มันสามารถเพิ่มประสิทธิภาพได้ด้วยการใช้ IP-base telephone network ซึ่งการทำแบบนี้จำเป็นที่จะต้องโทรศัพท์หรือตัวแปลงแบบพิเศษที่อนุญาตให้โทรศัพท์ต่อเข้ากับระบบเครือข่าย TCP/IP ได้ โดย RTC Client API นั้นจะสร้างการเชื่อมต่อระหว่างโทรศัพท์กับ PSTN แล้ว ITSP นั้นจะทำการหาเกตเวย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างเครือข่ายไอพีกับ PSTN ซึ่ง API นี้อนุญาตให้มีการเชื่อมต่อโดยใช้เสียงระหว่างเครื่องคอมพิวเตอร์กับระบบโทรศัพท์ได้

- Application Sharing โดยที่มันจะสามารถอนุญาตให้ผู้ใช้ทำการแชร์แอปพลิเคชันกันได้ ทำให้คน 2 คนสามารถใช้โปรแกรมเดียวกันที่เป็นมาตรฐานของ Windows ได้ เช่น Microsoft Internet Explorer
- Application Specific Sessions เช่นผู้ใช้ต้องการเล่นเกมกันหลายคนก็อาจจะใช้ API นี้ในการสร้างการเชื่อมต่อแบบเรียลไทม์เพื่อควบคุม gaming media

4.3.3 Session information

ในการสร้างการเชื่อมต่อ นั้น เราจะใช้เมธอดของ IRTCClient::CreateSession เพื่อที่จะให้การเชื่อมต่อสามารถใช้งานได้ต้องใช้เมธอดของ IRTCClient::AddParticipant โดยที่การเชื่อมต่อจะถูกสร้างโดย SIP proxy server ถ้า URL นั้นเป็น อีเมล หรือ เบอร์โทรศัพท์ ซึ่งถ้ามันเป็นไอพีแอดเดรส หรือ ชื่อของเครื่องใช้ นั้น ก็จะสร้างช่องทางการสื่อสารโดยตรงเลขโดยไม่จำเป็นต้องใช้ SIP proxy

Session type	Definition
PC-PC	การสื่อสารแบบ audio หรือ วิดีโอ ระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง
PC-phone	การสื่อสารแบบ audio จากคอมพิวเตอร์ไปยังอุปกรณ์ที่เอสทีเอน
Phone-phone	การติดต่อโดยใช้เสียงผ่านทางอุปกรณ์ที่เอสทีเอนตั้งแต่ 2 ตัวขึ้นไป
IM	การติดต่อระหว่างผู้ใช้ 2 คนผ่านทางข้อความ text แบบ real time
MIM	การติดต่อข้อความ text แบบ real time ระหว่างผู้ใช้ตั้งแต่ 2 คนขึ้นไปขึ้นไป
Application	การใช้งาน application ร่วมกัน

ตารางที่ 4-1 แสดงความหมายของการเชื่อมต่อแต่ละชนิด

4.3.4 ความต้องการพื้นฐาน

โดยที่การเชื่อมต่อในแต่ละแบบก็จะมีชนิดของ media และจำนวนของผู้ที่จะใช้ร่วมกันแตกต่างกันไปโดยสามารถแบ่งออกได้ดังนี้

Session type	Media types	Participants
PC-PC	รับส่งเสียง วิดีโอ Whiteboard(T120) Application sharing (T120)	2
PC-phone	รับส่ง audio	2
Phone-phone	รับส่ง audio	2 คนขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

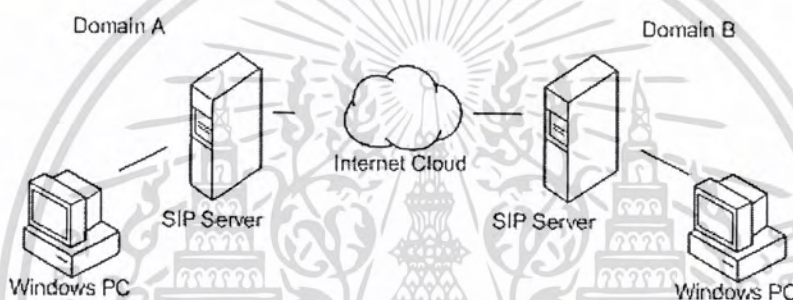
IM	เฉพาะข้อความแบบ text	2
Multiparty IM	เฉพาะข้อความแบบ text	2 คนขึ้นไป
APPLICATION	ส่งสื่อข้อมูลโดยใช้ application	2

ตารางที่ 4-2 แสดงชนิดของ media และจำนวนคนที่เข้าร่วมได้

4.4 Server requirements

4.4.1 การติดต่อจากของเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์

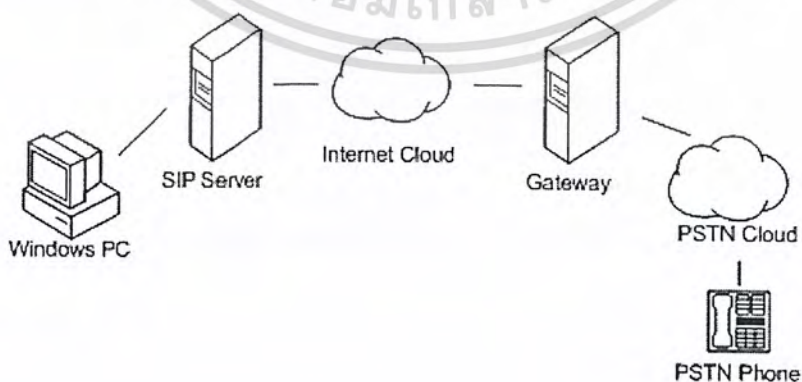
ถ้า SIP URL เป็นอีเมล SIP proxy server จะถูกใช้เป็นตัวที่ใช้ในการสร้างการเชื่อมต่อ ซึ่งถ้า SIP proxy server ไม่สามารถใช้ได้ผู้ใช้สามารถที่จะแสดง DNS lookup เพื่อที่จะหาตำแหน่ง



รูปที่ 4-2 แสดงรูปเครือข่ายการติดต่อของเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์

4.4.2 การติดต่อจากระหว่างเครื่องคอมพิวเตอร์กับโทรศัพท์

การจะแสดง URL ของโทรศัพท์ได้นั้นจำเป็นต้องใช้ SIP-PSTN gateway server โดย ITSP สามารถกำหนด gateway ใน PSTN หรือ gateway information อยู่ใน provisioning profile

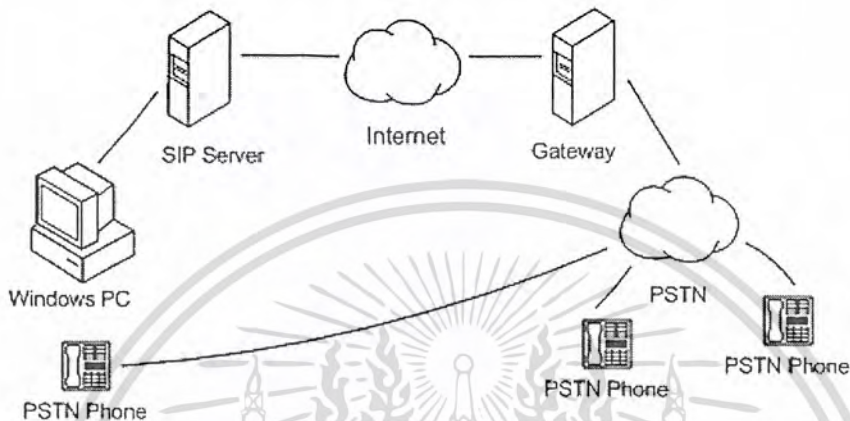


รูปที่ 4-3 แสดงรูปเครือข่ายการติดต่อระหว่างเครื่องคอมพิวเตอร์กับโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 การติดต่อของเครื่องโทรศัพท์กับเครื่องโทรศัพท์

การติดต่อแบบ third-party ซึ่งมีคอมพิวเตอร์เป็นตัวสร้างการติดต่อ จะใช้ได้เฉพาะการเชื่อมต่อแบบนี้ ซึ่ง ITSP นั้นเชื่อมต่ออยู่ระหว่างอุปกรณ์ PSTN อย่างน้อย 2 ตัว ซึ่งการติดต่อชนิดนี้สามารถหาเส้นทางบน PSTN หรือบางส่วนเป็น PSTN และบางส่วนเป็นอินเทอร์เน็ตก็ได้



รูปที่ 4-4 แสดงรูปเครือข่ายการติดต่อของเครื่องโทรศัพท์กับเครื่องโทรศัพท์

4.4.4 Presence information

การเชื่อมต่อสามารถใช้ชื่อของ presence information โดยที่ SIP registrar นั้นจะถูกใช้เมื่อมีความต้องการที่จะใช้ presence information นั้น โดยผู้ใช้สามารถที่จะเก็บข้อมูลของตัวมันหรือทำการดึงข้อมูลของตัวอื่นออกมาได้ ซึ่งเป็นตัวระบุกิจกรรมการติดต่อสื่อสารแบบเรียลไทม์ของผู้ใช้และยังระบุระดับของกิจกรรมด้วยเช่น การระบุว่าผู้ใช้อยู่ในสถานะออนไลน์หรือออฟไลน์

4.4.5 Profile Requirements

Internet telephony service provider (ITSP) หรือ third-party corporate-deployed server สามารถนำมาใช้ในบริการโทรศัพท์บนอินเทอร์เน็ต โดยผู้ใช้บริการอินเทอร์เน็ตนั้นจะเป็นผู้ใช้ gateway ติดต่อกับเครือข่ายพีเอสทีเอ็นสำหรับการติดต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องโทรศัพท์หรือสร้าง third-party สำหรับการเชื่อมต่อระหว่างเครื่องโทรศัพท์กับเครื่องโทรศัพท์โดยที่ XML profile นั้นจะถูกร้องขอสำหรับการเชื่อมต่อที่มีการใช้ service provider หรือ third-party ในกรณีที่ตัว client นั้นมี multiple service provider สำหรับ PC-to-PC PC-phone IM session ซึ่งมี profile ที่แตกต่างกัน โดย IRTCCClient::CreateSession สามารถใช้เพื่อระบุโพรไฟล์ได้ third-party corporate-deployed server จะถูกเก็บไว้ในโพรไฟล์ภายใต้เซิร์ฟเวอร์ องค์ประกอบส่วน ITSP นั้นจะถูกแสดงภายใต้ provider element

โพรไฟล์นั้นควรถูกเข้าถึงและจดทะเบียนบนเซิร์ฟเวอร์ด้วย IRTCCClientProvisioning::EnableProfile ก่อนที่จะถูกใช้ โดยที่ XML profile และ ITSP requirement ในแต่ละการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Session type	Profile
PC-PC	profile จะถูกใช้ในการระบุ proxy server ที่ใช้เป็นตัวส่งออกไป หรือ ITSP ที่ใช้ในการ set up การเชื่อมต่อ โดยถ้ายูอาร์แอลมันเป็นไอพีแอดเดรส หรือ machine name ก็จะทำให้การสร้างการติดต่อกันโดยตรงเลย
PC-phone	profile เอาไว้หาที่ตั้งของ SIP proxy server หรือ gateway server สำหรับการเชื่อมต่อแบบนี้ โดย ITSP สามารถที่ใช้การระบุ gateway หรือ gateway อาจจะถูกระบุโดยตรงเลยใน profile โดยที่ถ้า profile ไม่ได้ถูกใช้ SIP URI ของปลายทางควรจะประกอบด้วย แอดเดรส ของ gateway ไปพร้อมกับเบอร์โทรศัพท์.
Phone-phone	profile จะถูกใช้ในการระบุ ITST เพื่อที่จะสร้างการเชื่อมต่อและหาข้อมูลของ gateway
IM	profile จะถูกใช้ในการระบุ proxy server ที่ใช้เป็นตัวส่งออกไป หรือ ITSP ที่ใช้ในการ set up การเชื่อมต่อ โดยถ้ายูอาร์แอลมันเป็นไอพีแอดเดรส หรือ machine name ก็จะทำให้การสร้างการติดต่อกันโดยตรงเลย

ตารางที่ 4-3 แสดง profile ในแต่ละชนิดของการเชื่อมต่อ

4.4.6 SIP และ URL โทรศัพท์

ถ้าอุปกรณ์ต้องการที่จะเชื่อมต่อกับโทรศัพท์หรือคอมพิวเตอร์ RTC จะสร้าง SIP invite โดย URL จะถูกระบุอยู่ใน IRTCSession::AddParticipant โดย URL อาจจะเป็นของโทรศัพท์หรือของ SIP ก็ได้ โดยจะประกอบไปด้วย อีเมล แอดเดรส ไอพีแอดเดรส ดีเอ็นเอสแอดเดรส เบอร์โทรศัพท์ ถ้ามีการร้องขอ TEL URL ตัว RTC จะสร้าง SIP invite ที่มีเฮลเดอร์เป็นเบอร์โทรศัพท์ โดยสามารถดูตัวอย่างของ TEL URL ได้จากตาราง

TEL URL	Explanation
TEL: +1-425-555-0123	รูปแบบของ TEL URL ที่เป็น global phone number
TEL: +###-###-#####	รูปแบบของ TEL URL ที่เป็น global phone number format ('#' represents the digits in the actual phone number)

ตารางที่ 4-4 แสดงรูปแบบของ TEL URL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรูปแบบสำหรับ SIP URL จะอยู่ใน RFC 2543 ซึ่งแสดงได้ดังตาราง

SIP URL	Explanation
SIP:someone@10.1.2.3	SIP URL ที่ระบุด้วยไอพีแอดเดรส
SIP:someone@example.com;transport=TCP	SIP URL จะถูกระบุเหมือนการส่งแบบ TCP โดย default การติดต่อคือ UDP
SIP:someone@example.com	รูปแบบของ SIP URL พื้นฐาน
SIP:+12225550189@example.com;user=phone	SIP URL ที่ใช้เบอร์โทรศัพท์ที่ระบุโดย gateway
SIP:+1 206 555-0146@10.2.3.4;user=phone	รูปแบบ SIP URL ที่ใช้เบอร์โทรศัพท์ที่ระบุโดย gateway

ตารางที่ 4-5 แสดงรูปแบบของ SIP URL

4.5 การพิสูจน์ตัวตน

การพิสูจน์ตัวตนนั้นจะอนุญาตให้ RTC Client ตรวจสอบข้อความที่มาจากไคลเอนท์หรือเซิร์ฟเวอร์อื่นว่าข้อความที่ส่งมานั้นได้ถูกลอกเลียนแบบมาหรือไม่

4.5.1 Basic Authentication

รูปแบบการใช้นั้นขั้นการระบุตัวคือการ ใช้ username และ password โดย RTC client API จะไม่รับการตอบสนองเซิร์ฟเวอร์ ถ้า TLS ไม่ได้ระบุในโพรไฟล์สำหรับการสร้าง session แต่ข้อเสียของ basic authentication คือ username และ password นั้นจะถูกส่งไปบนเครือข่ายเป็น clear text โดยอาจจะถูกคนอื่นดูได้ TLS โพรโทคอล นั้นจะถูกใช้โดย RTC เพื่อที่จะเข้ารหัสข้อมูลอย่างไรก็ตามถ้า TLS ไม่มีตลอดเส้นทางการสื่อสาร segment นั้นก็สามารถที่จะถูกเห็นได้

4.5.2 Digest Authentication

เป็นการแสดงตัวใน SIP protocol ซึ่งไม่ต้องการหลักฐานที่จะส่งไปบนเครือข่ายในรูปแบบ clear text โดย password นั้นจะถูกส่งไปในขอบเขตของ client ในรูปแบบของ hashed form การแสดงตัวนั้นจะเกิดขึ้นเมื่อเซิร์ฟเวอร์ติดต่อกับไคลเอนท์ โดยที่ไคลเอนท์นั้นจะใช้การเรียกร่องของเซิร์ฟเวอร์ในการสร้าง password hash ซึ่งจะใส่เข้าไปในการตอบกลับไปให้เซิร์ฟเวอร์ โดยมีข้อดีที่เหนือกว่า basic authentication คือ hash password จะถูกส่งไปบนเครือข่าย โดยจะช่วยแนบไปกับเครือข่ายถ้า password ที่มียังไม่ดี ช่วยให้คุรหัสเรายากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3 NTLM Authentication

NTLM เป็นระบบแสดงตัวที่ถูกพัฒนาขึ้นสำหรับ Microsoft Window NT-based network โดยที่ โดเมนของผู้ใช้จะใช้ในการแสดงตัวกับ โคลเอนท์โดย window domain controller จะเรียกร่องไปยัง client เพื่อที่จะคำนวณ password โดเมนของผู้ใช้ และส่งไปเข้ารหัสแล้วส่งไปที่ domain controller ซึ่ง domain controller จะใช้การคำนวณเดียวกันเพื่อที่จะถอดรหัสเอา password มา ซึ่งถ้ารหัสที่ได้จากการถอดรหัสเหมือนกับที่ได้มาจาก เอสเอเอ็ม (Security Account Manager) database ซึ่ง client ก็จะได้ยืนยันการแสดงผลเรียบร้อย ซึ่งโคลเอนท์ก็สามารถที่จะเข้าไปในระบบเครือข่ายได้ และเข้าไปยังเครือข่ายได้ อุปกรณ์ RTC อาจที่จะเลือกเพื่อระบุ RTCAU_NTLM ในส่วนของแท็ก <Allowed Auth> ที่ได้รับการยอมรับจากโคลเอนท์ เพื่อที่จะใช้ เมื่อมีการร้องขอจากเซิร์ฟเวอร์

4.5.4 Kerberos Autentication

Kerberos เวอร์ชัน 5 นั้นใช้เพื่อที่จะระบุผู้ใช้และเซิร์ฟเวอร์ได้ โดยที่มันจะไม่ส่งหลักฐานรหัสไปบนสายทำให้มันมีความปลอดภัยมากที่สุดที่ถูกใช้โดย RTC client API คดยที่แบบอื่นนั้นจะใช้การเข้ารหัสหรือไม่เข้ารหัสแล้วส่งไปบนเครือข่าย โดยที่มันจะใช้ integrity feature ในการส่งข้อความไปแต่ละเซิร์ฟเวอร์ โดยมันจะในการยืนยันการแสดงผลของโคลเอนท์ เท่านั้น โดยที่ RTC Client API จะใช้ TLS ในการยืนยันการแสดงผลของ server

แอปพลิเคชันจะเป็นตัวที่ตัดสินใจว่าระดับของความปลอดภัยนั้นอยู่ที่ระดับไหน โดยจะระบุวิธีการยืนยันการแสดงผลในแท็ก <allowedauth> ใน XML profile เมื่อมีการร้องขอการแสดงผลมาที่ RTC Client API การยืนยันคนที่ปลอดภัยที่สุดจะถูกนำมาใช้ โดยลำดับความสำคัญในการยืนยันคน เรียงจากมากไปน้อยคือ Kerberos, NTLM, Digest, Basic

4.6 Transport Protocol

เมื่อ IM session ได้ถูกสร้างขึ้นอุปกรณ์สามารถที่จะระบุ provisioning profile เพื่อใช้สำหรับการเชื่อมต่อ SIP server element ใน profile นั้นประกอบไปด้วย transport protocol attribute โดยใน application สามารถตั้งได้ว่าจะเป็น UDP, TCP, TLS โดยถ้ากำหนดเป็น TLS สัญญาณจะถูกทำให้ปลอดภัยโดย TLS protocol

TLS นั้นเป็นส่วนที่อยู่บนสุดของ TCP โดยมันจะใช้การเข้ารหัสโดย TLS handshake protocol ในการเข้ารหัสข้อความ โดย key ที่ถูกเข้ารหัสนั้นจะถูกใช้ระหว่าง client กับ เซิร์ฟเวอร์ การยืนยันคนของโคลเอนท์ กับเซิร์ฟเวอร์โดยการใช้ TLS handshake protocol ก่อนการเชื่อมต่อสัญญาณจะถูกสร้างขึ้น

RTC Client API สร้างการเชื่อมต่อที่มีการเข้ารหัสสัญญาณไปยังเซิร์ฟเวอร์โดยใช้ TLS แต่ถ้าไม่ได้ระบุว่าจะใช้ TLS สัญญาณที่ส่งไปก็จะไม่ได้รับการเข้ารหัส ซึ่งโดยทั่วไปแล้ว server ที่เป็นทางผ่านของ packet นั้นไม่ได้มีเข้ารหัส packet จะสามารถที่จะถูกลอบฟังบนเครือข่ายได้

RTC Client API จะสนับสนุนการเชื่อมต่อแบบ peer-to-peer ซึ่งโดยทั่วไปแล้วการเชื่อมต่อแบบ peer-to-peer นั้นจะไม่มีเข้ารหัส เพราะ TC Client API นั้นจะไม่รับการเชื่อมต่อแบบ TLS ที่เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ข้อดีของ RTC Client API

API ตัวนี้เป็นได้มากกว่าการสร้างแอปพลิเคชันสำหรับการติดต่อสื่อสารด้วยข้อความ แต่มันยังสามารถสนับสนุนการเชื่อมต่อที่ใช้เสียง หรือ วิดีโอ ข้อความ การแชร์แอปพลิเคชันและ whiteboard โดยที่ API ตัวนี้ทำงานร่วมกับ registrar server เพื่อที่จะจัดการกับ presence information

4.7.1 การสร้างแอปพลิเคชัน โดยการใช้ RTC Client API

เพื่อที่จะสนับสนุนการสื่อสารแบบเรียลไทม์ระหว่างคอมพิวเตอร์ API นี้จะต้องสนับสนุนการติดต่อโดยใช้เสียงระหว่าง คอมพิวเตอร์กับโทรศัพท์ และกับโทรศัพท์ตั้งแต่ 2 สายขึ้นไป แล้วแต่ชนิดของแอปพลิเคชันซึ่งการสื่อสารทั้งหมดเกิดขึ้นบนระบบเครือข่ายไอพี หรือผ่านอุปกรณ์ที่แนบไปกับระบบเครือข่ายไอพี

ถ้าอุปกรณ์ 2 ฟังจะสามารถติดต่อกันได้จำเป็นต้องมีชนิดของการสื่อสารที่เข้ากันได้ซึ่งหมายความว่าจำเป็นที่จะต้องมีไมโครโฟนและลำโพงในการติดต่อ ถ้าเราทำการติดต่อแบบวิดีโอให้สามารถเห็นกันได้จำเป็นต้องมีกล้องเพิ่มเข้าไปอีก

4.7.2 Desktop Application

Windows Messenger เป็นตัวอย่างที่ดีซึ่งมันอนุญาตให้มีการติดต่อสื่อสารได้ตั้งแต่ 2 คนขึ้นไป โดยติดต่อกันแบบเรียลไทม์ ปกติแล้วแอปพลิเคชันที่เป็น Instant Message จะสนับสนุนการติดต่อโดยเสียง และวิดีโอ โดย Windows Messenger 5.0 นั้นได้เพิ่มความสามารถในการติดต่อด้วยโทรศัพท์และการแชร์แอปพลิเคชันระหว่างผู้ใช้เข้าไป

4.7.3 Scalable Application

API ตัวนี้สามารถที่จะสร้างแอปพลิเคชันเพิ่มเติมได้เช่น notification bot โดยเอาใช้หา presence information ของผู้ใช้ที่เราสนใจแล้วทำการส่งข้อความไปบอกผู้ใช้เมื่อมีอะไรเกิดขึ้น อีกอันคือ interactive bot ซึ่งแอปพลิเคชันชนิดนี้จะทำการขอรับการร้องขอการสร้างการเชื่อมต่อแบบเรียลไทม์ที่เข้ามา จากผู้ใช้ที่ได้รับการพิสูจน์ตนแล้วมันจะทำโดยการเข้าถึง knowledge base หรือ โดยให้มันติดต่อกับ customer service representative

บทที่ 5

J2ME

ภาษาจาวาเป็นภาษาเชิงอ็อบเจกต์ที่มีจุดเด่นตรงที่สามารถทำงานได้ทุกระบบปฏิบัติการ โดยไม่ต้องคอมไพล์โปรแกรมใหม่โดยโปรแกรมที่เขียนด้วยภาษาจาวานั้นจะถูกนำไปคอมไพล์โดยจาวาคอมไพล์เลอร์ ให้เป็นกลุ่มคำสั่งที่เรียกว่า คลาสโดยไบต์โค้ดเหล่านี้จะถูกนำไปใช้โดย Java Virtual Machine เพื่อแปลความหมาย เป็นภาษาเครื่องของแต่ละระบบปฏิบัติการอีกที การพัฒนาโปรแกรมโดยจาวาจึงไม่ต้องคำนึงถึงระบบปฏิบัติการที่จะมารองรับการทำงานของโปรแกรม ซึ่ง J2ME นั้นเป็นภาษาจาวาเวอร์ชันเล็กสุดทำงานบนอุปกรณ์ประเภทมือถือ เช่น โทรศัพท์มือถือและคอมพิวเตอร์พกพา ซึ่งอุปกรณ์ประเภทมือถืออื่นจะมีขนาดเล็กและมีทรัพยากรจำกัด เช่น ขนาดหน้าจอ หน่วยความจำ จึงได้มีการพัฒนา J2ME เพื่อให้สามารถติดตั้งบนอุปกรณ์เหล่านี้ได้ ซึ่งใน J2ME คลาสบางคลาสนั้นเป็นส่วนของ J2SE แต่ไม่สามารถใช้ทั้งหมดได้เพราะบางคลาสใช้ทรัพยากรของระบบเยอะและมีรูปแบบรูปแบบการทำงานที่ไม่เหมาะสมกับอุปกรณ์

นอกจากนี้แอปพลิเคชันที่พัฒนาด้วย J2ME ยังสามารถติดตั้งเพิ่มบนอุปกรณ์ประเภทมือถือได้อีกด้วยซึ่งแต่เดิมแอปพลิเคชันของอุปกรณ์เหล่านี้จะต้องติดตั้งจากโรงงานเท่านั้น เพื่อให้ความหลากหลายของเทคโนโลยีอยู่ในขอบเขตของ J2ME ทาง Sun จึงใช้หลักการที่เรียกว่า Configuration และ profile

5.1 Configuration

คือกลุ่มคลาสไลบรารีพื้นฐานสำหรับอุปกรณ์ประเภทมือถือทุกชนิดการที่จะให้กลุ่มคลาสเหล่านี้ครอบคลุมทุกอุปกรณ์นั้นจำเป็นต้องนำอุปกรณ์มาจัดกลุ่มโดยใช้คุณสมบัติพื้นฐานของอุปกรณ์คือ หน่วยความจำ การแสดงผล การเชื่อมต่อสัญญาณและการใช้พลังงาน แบ่งเป็น 2 กลุ่มได้ดังนี้

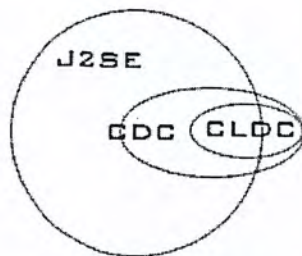
Connect Device Configuration (CDC)

- ใช้หน่วยความจำอย่างต่ำ 512 กิโลไบต์ สำหรับจาวา
- ใช้หน่วยความจำ 256 กิโลไบต์ ขณะรันไทม์
- การเชื่อมต่อสัญญาณมีแบนด์วิธสูง

Connected,Limited Device Configuration (CLDC)

- ใช้หน่วยความจำอย่างต่ำ 128 กิโลไบต์ สำหรับจาวา
- ใช้หน่วยความจำ 32 กิโลไบต์ ขณะรันไทม์
- มี User Interface ที่จำกัด
- ใช้พลังงานน้อย
- เชื่อมต่อสัญญาณด้วยแบนด์วิธต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-1 แสดงความสัมพันธ์ระหว่าง J2ME Configuration และ J2SE

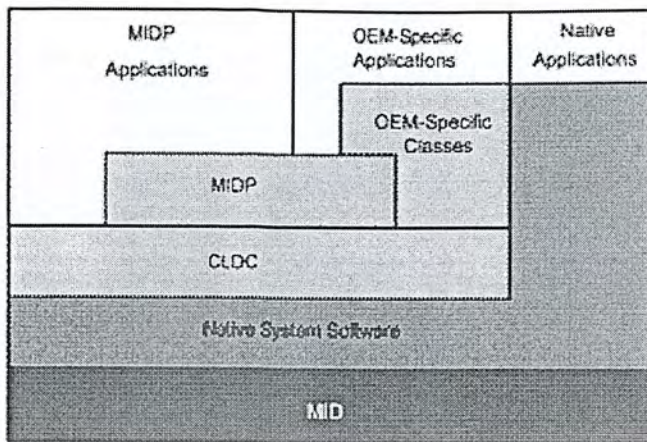
อุปกรณ์ในกลุ่ม CDC นั้นมีทรัพยากรของระบบมาก ทำให้สามารถใช้กลุ่มคลาสไลบรารีเหมือนกับ J2SE ได้ แต่สำหรับอุปกรณ์ในกลุ่ม CLDC เช่น โทรศัพท์ และ PDAs จะไม่สามารถใช้คลาสจาก J2SE ได้ทั้งหมดเพราะข้อจำกัดที่กล่าวมาจึงต้องพัฒนาคลาสไลบรารีขึ้นมาใหม่โดยมีข้อจำกัดคือ

- ไม่รองรับการทำงานกับข้อมูลชนิด Floating Point ทำให้ไม่มีคลาส `java.lang.Float` และ `java.lang.Double` รวมถึงเมธอดที่ส่งผลลัพธ์กลับมาเป็นข้อมูลชนิดนี้ด้วย
- ไม่มีเมธอด Finalization ในคลาส `java.lang.Object` เพื่อให้ขั้นตอนการคืนทรัพยากรให้กับระบบทำได้ง่ายขึ้น
- การจับข้อผิดพลาดในขณะรันไทม์มีเพียงคลาส `java.lang.Error`, `java.lang.OutOfMemoryError` และ `java.lang.VirtualMachineError`

เมื่อกำหนดกลุ่มคลาสไลบรารีพื้นฐานสำหรับอุปกรณ์ทั้งสองแล้ว จะต้องกำหนดกลุ่มคลาสเพื่อรองรับลักษณะที่แตกต่างกันของแต่ละอุปกรณ์ที่เรียกว่า Profile

5.2 Profile

คือส่วนเพิ่มเติมจาก Configuration เป็นกลุ่มคลาสไลบรารีที่รองรับข้อแตกต่างของอุปกรณ์ เช่น Profile ของอุปกรณ์ประเภทมือถือคือ MIDP จะเป็นกลุ่มคลาสไลบรารีต่างๆที่รองรับการพัฒนาแอปพลิเคชันบนอุปกรณ์ประเภทมือถือ กลุ่มคลาสไลบรารีที่กล่าวมาประกอบด้วย UI, Input/Output, Event Handling, Persistent Storage, Networking และ Timers



รูปที่ 5-2 โครงสร้างของ MIDP

5.3 KVM

โปรแกรมภาษาจาวาทุกตัวจะต้องทำงานภายใต้ JVM (Java Virtual Machine) เสมอไม่ว่าจะเป็น Servlet/JSP หรือ Applet ก็ตาม เมื่อเรากอมไพล์โปรแกรมจาวาเป็นคลาสแล้ว JVM จะทำหน้าที่แปลงคลาสเหล่านั้นเป็นภาษาเครื่องทำงานตามคำสั่งต่อไป

ใน J2ME จะใช้ Configuration เป็นตัวกำหนด JVM โดยในกลุ่ม CDC ซึ่งใช้กลุ่มคลาสชุดเดียวกับ J2SE ทำให้สามารถใช้ JVM ตัวเดียวกันได้ แต่กลุ่ม CLDC ไม่ได้ใช้คลาสเดียวกันทั้งหมดทำให้ต้องพัฒนา JVM ขึ้นมาใหม่ เรียกว่า K Virtual Machine หรือ KVM ซึ่งมีคุณสมบัติดังนี้

- ทำงานที่หน่วยความจำเพียง 40 และ 80 กิโลไบต์
- ทำงานที่โปรเซสเซอร์ 16 บิต และมีความเร็วในการประมวลผล 25 MHz

5.4 สถาปัตยกรรมของ J2ME

สถาปัตยกรรมของ J2ME นั้นแบ่งออกเป็น 4 ส่วนคือ

- ระบบปฏิบัติการของอุปกรณ์ของแต่ละตัว
- JVM บนอุปกรณ์ประเภท CDC จะใช้ JVM ของ J2SE ส่วนอุปกรณ์ประเภท CLDC จะใช้ KVM
- Configuration กลุ่มคลาสพื้นฐานของอุปกรณ์
- Profile กลุ่มคลาสเพิ่มเติม

สำหรับสถาปัตยกรรมอุปกรณ์ประเภทมือถืออื่นนั้น จะใช้ Profile คือ MIDP และ Virtual Machine คือ KVM

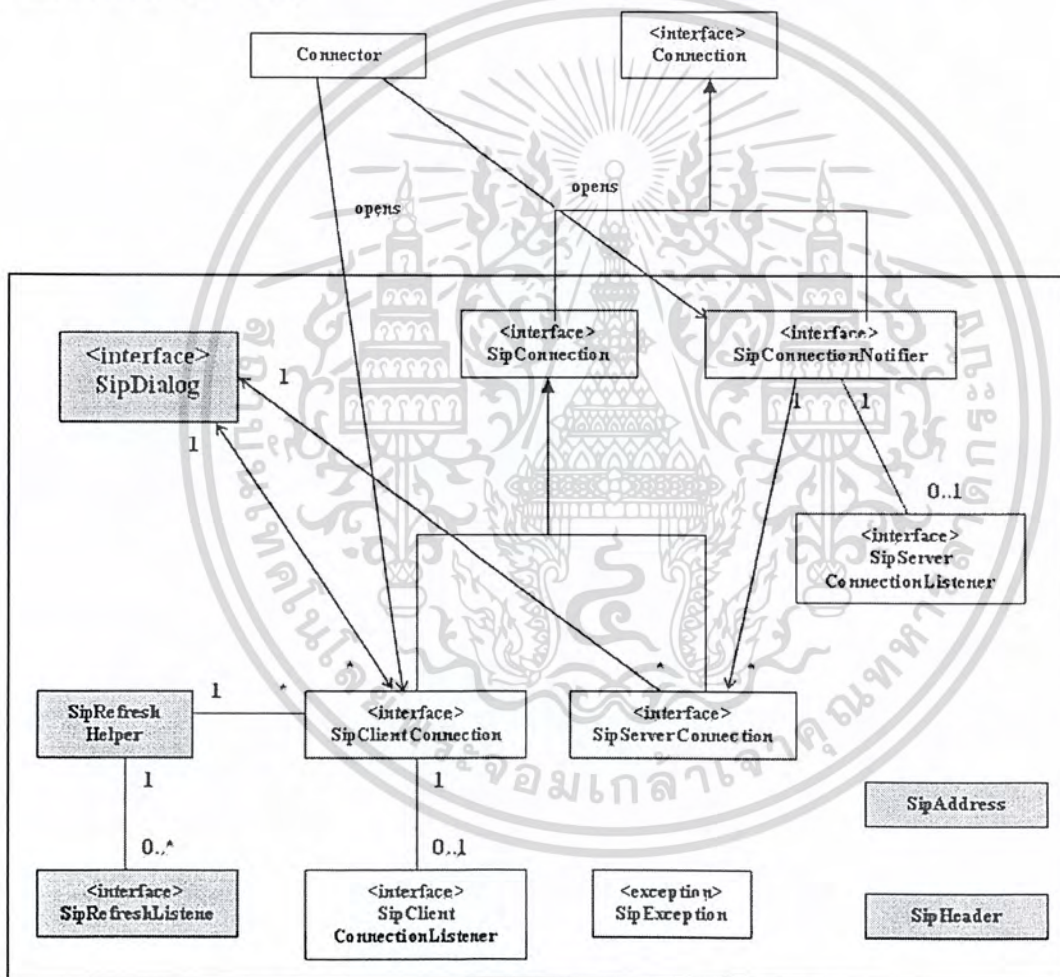
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

SIP API สำหรับ J2ME

SIP API สำหรับ J2ME (JSR180) ได้ถูกระบุให้สามารถเข้าถึง resource เพื่อที่จะส่งและรับ SIP message โดยที่ API นี้ได้ถูกออกแบบเพื่อให้เข้ากันได้กับ SIP API ซึ่ง API จะถูกรวมเข้ากับ Generic Connection Framework ที่ถูกกำหนดใน Connected, Limited Device Configuration (CLDC) SIPA ได้ถูกออกแบบมาเพื่อ run บน J2ME™ configuration ในการรวมของ J2ME profile

SIPA นั้นเป็น optional package ซึ่งสามารถที่จะ implement และใช้ได้บน Java™ enabled devices ที่อย่างน้อย CLDC 1.0.

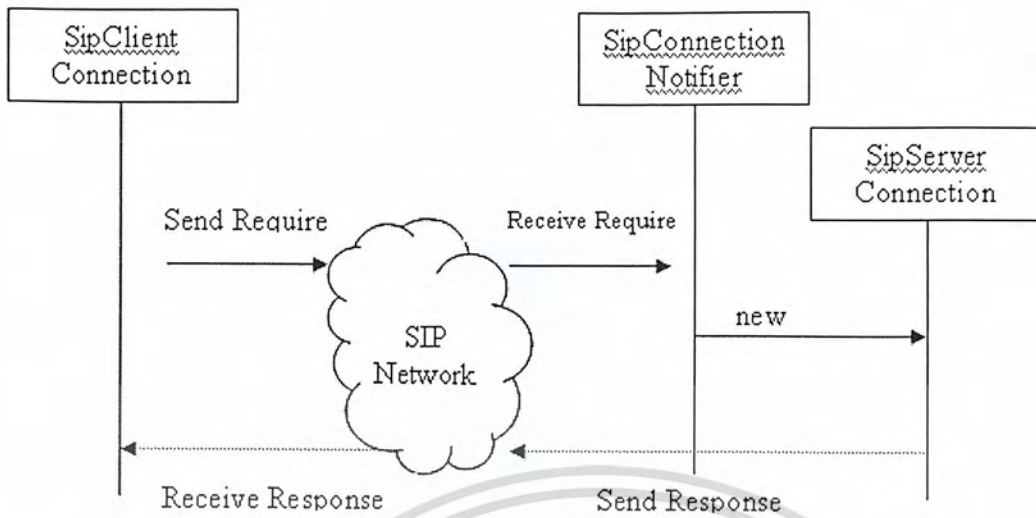


รูปที่ 6-1 แสดง class diagram ของ API และความสัมพันธ์ระหว่างคลาส

6.1 การใช้ SIP API สำหรับ J2ME

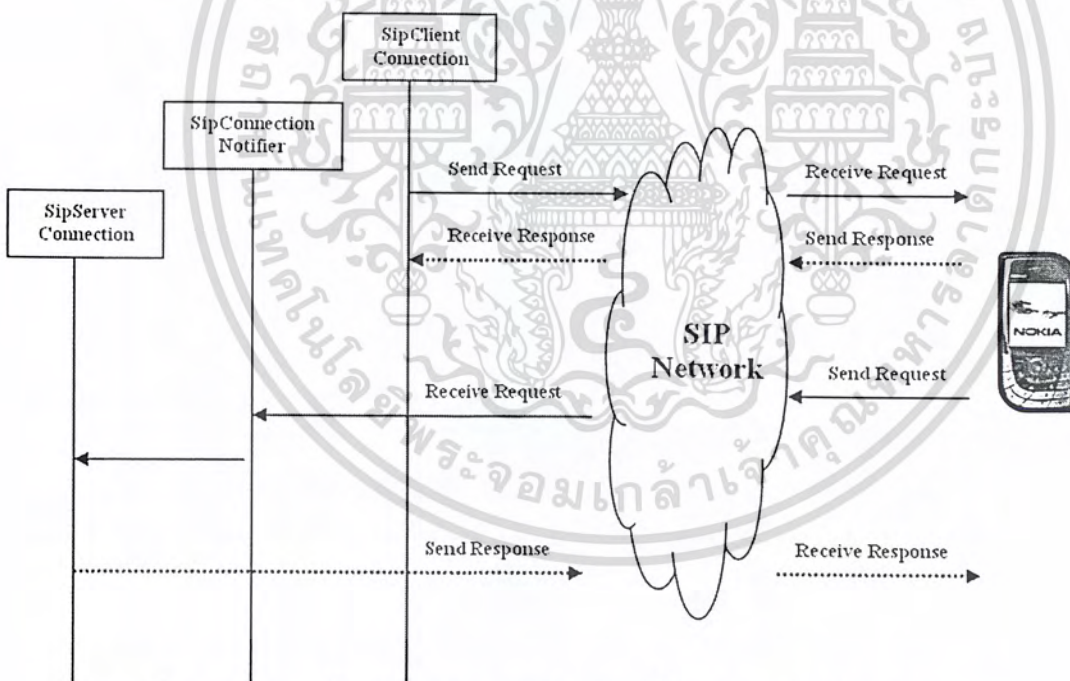
SIP API นั้นถูกใช้โดย application ที่ประยุกต์ใช้กับการทำงานของ SIP User Agent (UA) รูปแสดงว่า interface อะไรที่ terminal A และ B ใช้ในการใช้ SIP User Agent Client (UAC) และ User Agent Server (UAS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-2 แสดงการเชื่อมต่อระหว่าง UAC กับ UAS ซึ่งเป็น terminal แบบเดียวกัน

ในอุปกรณ์จริงนั้นจะใช้การเชื่อมต่อกับทั้ง UAS และ UAC ใน terminal เดียวกัน



รูปที่ 6-3 แสดงการเชื่อมต่อระหว่าง UAC กับ UAS ซึ่งเป็น terminal ต่างชนิดกัน

6.2 ฟังก์ชันของ SIP ที่สนับสนุน

SIPA reference implementation (RI) นั้นถูกสนับสนุนด้วย SIP messaging บน local area network ซึ่ง RI จะอนุญาตให้ application สามารถที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่ง SIP message
- รับ SIP message ในเวลาเดียวกัน
- รับ SIP message listener notification ในเวลาเดียวกัน

RI ใช้ส่วนที่จำเป็นของ RFC3261 โดยที่ SIP feature นั้นถูกสนับสนุนโดย

- วิธีการ Pre-initialized ด้วย header ที่เกี่ยวกับคำสั่ง นั้นถูกติดตั้งโดยอัตโนมัติ
- ได้รับการสนับสนุนโดย Method: REGISTER, INVITE, ACK, CANCEL, BYE, OPTIONS, REFER และ method ทั่วไปของ SIP พวก MESSAGE, SUBSCRIBE, NOTIFY, PRACK
- สนับสนุนโดย แผนผังโครงสร้างของ SIP (sips ไม่ได้ถูกสนับสนุน)
- Pre-initialized นั้นจะตอบสนองโดยคำที่มีเหตุผลที่เป็นมาตรฐาน
- เป็นไปได้ที่จะทำการเพิ่ม message body ลงไปทั้งในการร้องขอและการตอบรับ
- Client/server transaction จะมี timer ตัวพื้นฐานที่ใช้ในการส่งคำร้องขอ
- คำร้องขอ SipRefreshHelper นั้นสนับสนุน ทั้ง REGISTER และ SUBSCRIBE
- Configurable(internal.config) จะใช้ address, display name, outbound proxy, loco. System port
- Multiple listening points (SipConnectionNotifier) ต่อ 1 MIDlet
- SipConnectionNotifier สามารถที่สร้างด้วยการระบุ (sip:5080) หรือ arbitrary port (sip:) หรือ การทำ shared system port (sip;;type="application/test")
- มี Local-loop routing สำหรับ TCK test
- การพิสูจน์ตัวตนของ SIP จะไม่ได้รับการสนับสนุน

6.3 System Requirement

MIDP 2.0 รวมถึง SIPA RI นั้นถูกสนับสนุนบน Microsoft Window 2000/XP ซึ่งระบบพื้นฐานที่ต้องการในการ run ตัว phone emulator ด้วย SIP API คือ

Hardware Requirement

- 200 MHz Pentium processor
- 64 MB RAM
- TCP/IP network adapter

Software Requirement

- Windows 2000/XP operating system

Java Heap Memory Requirement

SIP RI จะแสดง I/O operation และ data processing ใน Java layer ปริมาณของหน่วยความจำที่แน่นอนจาก Java heap นั้นขึ้นอยู่กับว่าส่วนไหนของ API ถูกใช้และถูกใช้โดย application อย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การออกแบบและพัฒนาโปรแกรม

7.1 ความต้องการของระบบ

7.1.1 หลักการ

เป็นโปรแกรมที่ใช้ในการติดต่อสื่อสารบนเครือข่ายอินเทอร์เน็ตโดยใช้โปรโตคอลซีพี โดยโปรแกรมที่พัฒนาจะแบ่งออกเป็น 2 ส่วนคือ ส่วนที่ทำงานบนเครื่องคอมพิวเตอร์พีซีและส่วนที่ทำงานบนอุปกรณ์มือถือ โปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์พีซีนั้นสามารถติดต่อสื่อสารได้ทั้ง ข้อความ ภาพ วีดีโอ และเสียง ส่วนโปรแกรมที่ทำงานบนอุปกรณ์มือถือสามารถติดต่อสื่อสารด้วยข้อความเท่านั้น

วิธีการทำงานทำโดยเครื่อง โคลเอนท์จะทำการติดต่อไปยังเซิร์ฟเวอร์เพื่อทำการร้องขอในการเข้าระบบเพื่อขอใช้บริการการติดต่อสื่อสาร จากนั้น โคลเอนท์จะทำการเลือกผู้ที่ต้องการที่จะติดต่อด้วย แล้วส่งข้อมูลไปบอกความต้องการที่จะติดต่อกับคู่สนทนา

7.1.2 ขอบเขตการทำงาน

สำหรับการใช้งานการติดต่อสื่อสารโดยใช้โปรโตคอลซีพีนั้น จะเป็นการให้บริการกับบุคคลภายในองค์กรที่ต้องการติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีผู้ดูแลระบบเป็นผู้ที่ทำการสร้างสมาชิกของระบบและกำหนดรูปแบบการใช้งานของสมาชิกแต่ละคน การติดต่อสื่อสารในลักษณะนี้จะเป็นการติดต่อสื่อสารในรูปแบบที่เป็นส่วนตัว คือ ผู้ใช้งานระบบจำเป็นต้องเป็นคนภายในองค์กรเท่านั้น

โดยโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์สามารถติดต่อกับโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ได้เท่านั้น และโปรแกรมที่ทำงานของอุปกรณ์มือถือสามารถติดต่อกับโปรแกรมที่ทำงานบนอุปกรณ์มือถือได้เท่านั้นเช่นกัน

7.1.3 ฟังก์ชันการทำงานต่าง ๆ

ฟังก์ชันการทำงาน	คอมพิวเตอร์พีซี	อุปกรณ์มือถือ
การล็อกอินเข้าสู่ระบบ	/	/
การพิสูจน์คน	/	
แสดงรายชื่อผู้ติดต่อและสถานะของผู้ติดต่อ	/	
เพิ่ม/ลบ รายชื่อผู้ติดต่อ	/	
ป้องกัน/อนุญาต การติดต่อ	/	
การรับส่งข้อความแบบหนึ่งต่อหนึ่ง	/	/
การรับส่งข้อความแบบกลุ่ม	/	
การติดต่อสื่อสารด้วยเสียงและภาพวีดีโอ	/	
การล็อกเอาต์ออกจากระบบ	/	/

ตารางที่ 7-1 แสดงฟังก์ชันการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1.4 การล็อกอินเข้าสู่ระบบ

ผู้ที่ เป็นสมาชิกภายในองค์กรและมีความประสงค์ในการใช้งานระบบ สามารถทำการล็อกอินเข้าสู่เซิร์ฟเวอร์ที่ให้บริการที่ตนเองเป็นสมาชิกอยู่ โดยทำการป้อน ซิพแอดเดรสและเซิร์ฟเวอร์แอดเดรส เข้าไปก็สามารถเข้าสู่ระบบได้ โดยเซิร์ฟเวอร์จะทำการสร้างรักษาเซสชันของการเชื่อมต่อไว้

7.1.5 การพิสูจน์ตน

เมื่อทำการล็อกอินเข้าสู่ระบบ สำหรับสมาชิกคนใดที่ ผู้ดูแลระบบ ได้ทำการตั้งรหัสผ่านไว้ ก็จะมี โดอะล็อกขึ้นมาให้ใส่ข้อมูลเพิ่มเติม ซึ่งได้แก่ ซิพแอดเดรสและรหัสผ่าน

7.1.6 แสดงรายชื่อผู้ติดต่อและสถานะของผู้ที่ติดต่อ

เมื่อการล็อกอินเข้าสู่ระบบสำเร็จ โปรแกรมจะทำการดึงข้อมูลของรายชื่อผู้ติดต่อที่เก็บไว้ในตัวเครื่องแล้วส่งไปยังเซิร์ฟเวอร์ เพื่อทำการตรวจสอบว่า สมาชิกคนใด ที่ได้ที่ออนไลน์อยู่ และสมาชิกคนใดที่ไม่ได้ออนไลน์ พร้อมทั้งบอกสถานะของสมาชิกที่ออนไลน์อยู่ของแต่ละคนด้วย โดยสถานะที่จะได้นั้นได้แก่ ออนไลน์ (online), ไม่ได้ออนไลน์ (offline), ไม่อยู่ (Away), ไม่ได้ทำงาน (Idle), ไม่ว่าง (Busy), จะกลับมา (Be Right Back), คุยโทรศัพท์อยู่ (On The Phone), รับประทานอาหาร (Out To Lunch)

7.1.7 เพิ่ม/ลบ รายชื่อผู้ติดต่อ

ในการติดต่อสื่อสารกับสมาชิกคนใด จำเป็นต้องมีรายชื่อของสมาชิกคนนั้นอยู่ในรายชื่อผู้ติดต่อด้วย ดังนั้นเมื่อเราต้องการที่จะทำการติดต่อกับสมาชิกคนใดที่ไม่ได้มีรายชื่ออยู่ในรายชื่อผู้ติดต่อ จำเป็นต้องทำการเพิ่มรายชื่อของสมาชิกคนนั้นลงไป โดยการใส่ซิพแอดเดรสของสมาชิกคนดังกล่าว และยังสามารถทำการลบรายชื่อของสมาชิกคนอื่น ๆ ได้

7.1.8 ป้องกัน/อนุญาต การติดต่อ

สำหรับสมาชิกคนใดที่ไม่ต้องการให้สมาชิกอีกคนหนึ่งเห็นว่าตนเองได้ทำการออนไลน์อยู่ สมาชิกผู้นั้นสามารถทำการป้องกันไม่ให้สมาชิกคนดังกล่าวทำการติดต่อได้กับตนเองได้

7.1.9 การรับส่งข้อความแบบหนึ่งต่อหนึ่ง

สมาชิกที่ได้ทำการล็อกอินเข้าสู่ระบบแล้วสามารถทำการรับส่งข้อความกันได้ กับสมาชิกอีกคนหนึ่งซึ่งได้ทำการใช้งานในระบบอยู่ด้วยเช่นกัน โดยผู้ที่ต้องการจะส่งข้อความสามารถทำได้โดยเลือกชื่อของผู้ที่จะติดต่อด้วยในรายชื่อผู้ติดต่อหรือทำการใส่ ซิพแอดเดรส ของสมาชิกคนดังกล่าว

7.1.10 การรับส่งข้อความแบบเป็นกลุ่ม

เมื่อเกิดการรับส่งข้อความแบบหนึ่งต่อหนึ่งแล้ว ถ้าสมาชิกคนใดคนหนึ่งต้องการที่จะทำการเรียกสมาชิกอีกคนหนึ่งที่ได้ทำการใช้งานในระบบอยู่เข้ามาร่วมการสนทนา ก็สามารถทำได้โดยทำการใส่ ซิพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

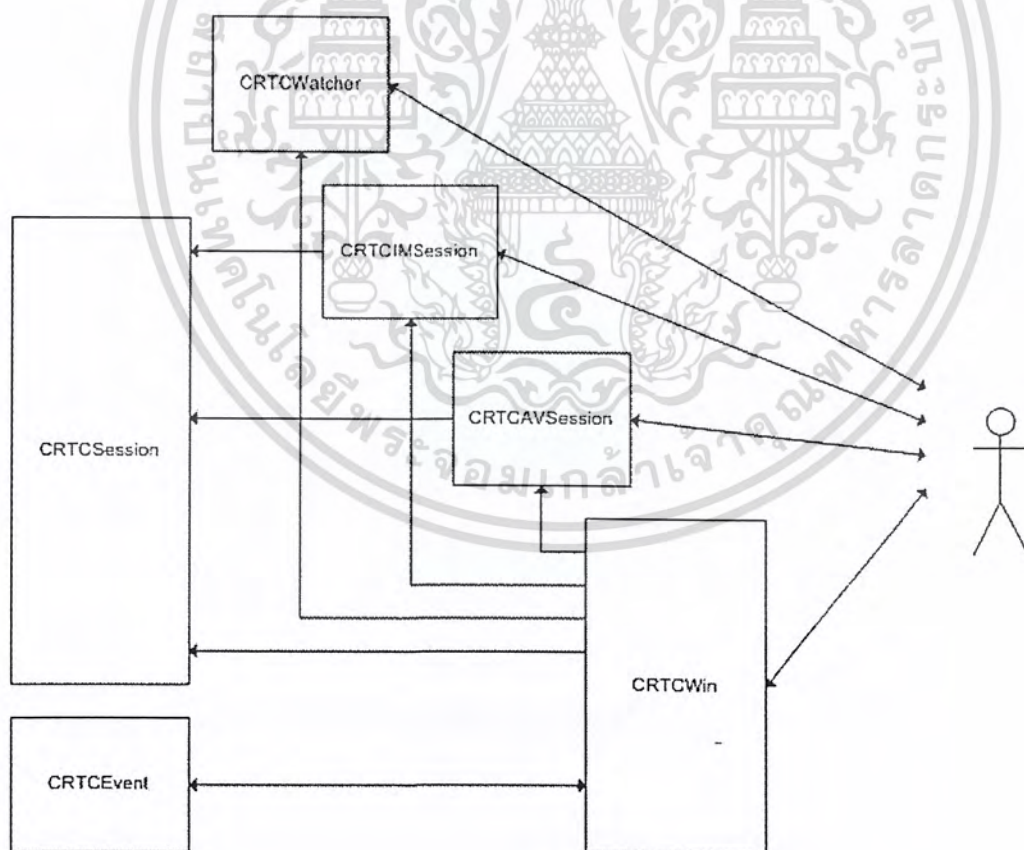
แอดเดรสของสมาชิกคนดังกล่าว โดยในการรับส่งข้อความแบบกลุ่มนั้นสามารถมีสมาชิกภายในกลุ่มการสนทนาได้มากถึง 32 คน

7.1.11 การติดต่อสื่อสารด้วยเสียงและภาพวิดีโอ

สมาชิกที่ได้ทำการล็อกอินเข้าสู่ระบบแล้วสามารถทำการติดต่อสื่อสารด้วยและภาพวิดีโอ กับสมาชิกอีกคนหนึ่งซึ่งได้ทำการใช้งานในระบบอยู่ด้วยเช่นกัน โดยผู้ที่ต้องการจะเริ่มทำการติดต่อสื่อสารสามารถทำได้โดยเลือกชื่อของผู้ที่จะติดต่อด้วยในรายชื่อผู้ติดต่อหรือทำการใส่ อีเมลแอดเดรส ของสมาชิกคนดังกล่าว โดยหน้าต่างของการติดต่อสื่อสารด้วยเสียงและภาพวิดีโอนั้นจะทำการตรวจสอบว่าเครื่องของตนเองและเครื่องของผู้ที่กำลังติดต่ออยู่นั้นมีกล้องเว็บแคมติดอยู่ด้วยหรือไม่ ถ้าฝั่งใดฝั่งหนึ่งมีกล้องเว็บแคมติดไว้ที่เครื่องคอมพิวเตอร์ฝั่งตรงกันข้ามก็สามารถมองเห็นได้

7.2 การออกแบบโปรแกรม

โครงสร้างการทำงานของโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์พีซี



รูปที่ 7-1 โครงสร้างของโปรแกรมบนเครื่องคอมพิวเตอร์พีซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของโปรแกรมประกอบไปด้วยคลาสหลัก 7 คลาส ได้แก่

7.2.1 CRTCTWin

เป็นส่วนหลักของโปรแกรมทำหน้าที่เกี่ยวข้องกับการทำงานต่าง ๆ ได้แก่

- การล็อกอิน - ล็อกเอาท์
- การพิสูจน์ตน
- การจัดการเกี่ยวกับรายชื่อผู้ติดต่อ
- เรียกฟังก์ชันการทำงานอื่น ๆ
- เป็นตัวกลางจัดการการทำงานต่าง ๆ
- รอร์บีอีเวนต์ต่าง ๆ ที่ได้รับมาจากคลาส CRTCEvent

7.2.2 CRTCIMSession

ทำงานเกี่ยวกับยูสเซอร์อินเทอร์เฟซติดต่อกับผู้ใช้งานในส่วนของการรับส่งข้อความ ทั้งการรับส่งข้อความแบบหนึ่งต่อหนึ่งและการรับส่งข้อความแบบกลุ่ม

7.2.3 CRTCAVSession

ทำงานเกี่ยวกับยูสเซอร์อินเทอร์เฟซติดต่อกับผู้ใช้งานในส่วนของการติดต่อสื่อสารด้วยเสียงและภาพวิดีโอ และการควบคุมคอนโทรลต่าง ๆ ที่ใช้ในการรับส่งข้อความเสียงและภาพวิดีโอ

7.2.4 CRTCTWatcher

ทำหน้าที่เกี่ยวกับยูสเซอร์อินเทอร์เฟซติดต่อกับผู้ใช้ที่เกี่ยวข้องกับการจัดการป้องกันและอนุญาตจากสมาชิกคนอื่น

7.2.5 CRTCSession

ทำหน้าที่ดูแลการเชื่อมต่อต่าง ๆ และเรียกใช้งานฟังก์ชันต่าง ๆ ที่เกี่ยวข้องกับการเชื่อมต่อ

7.2.6 CRTCEvent

ทำหน้าที่จัดการเกี่ยวกับอีเวนต์ต่าง ๆ และทำการเรียกการทำงานคอมโพเนนต์อ็อบเจกต์โมเดล เพื่อตรวจจับเหตุการณ์ต่าง ๆ ที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างการทำงานของโปรแกรมที่ทำงานบนอุปกรณ์มือถือ

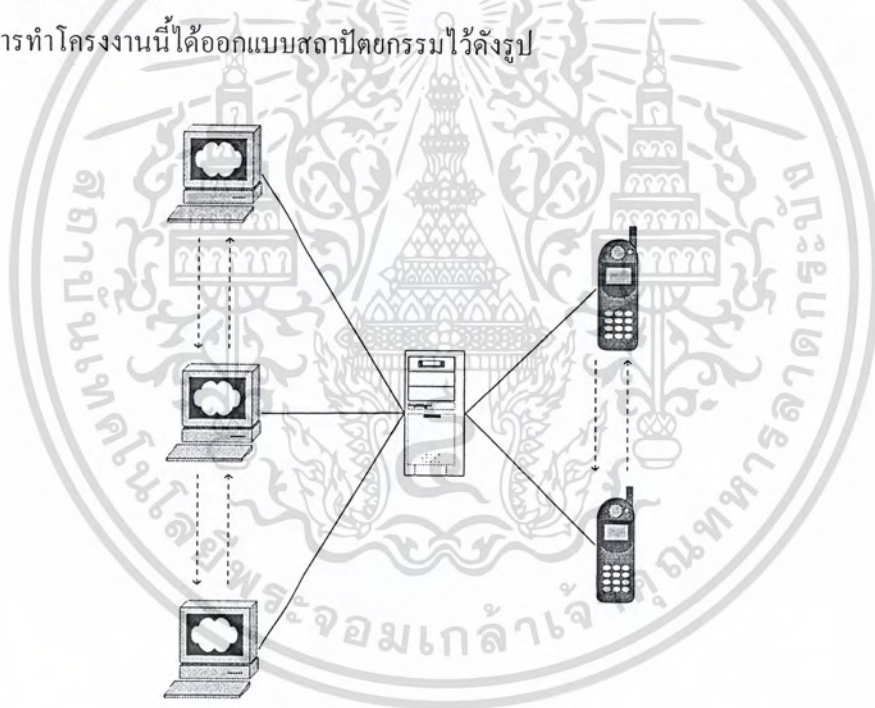


รูปที่ 7-2 โครงสร้างของโปรแกรมบนอุปกรณ์มือถือ

โครงสร้างของโปรแกรมส่วนนี้มี 1 คลาสซึ่งรองรับเหตุการณ์ต่าง ๆ ที่เกิดขึ้นจากเมธอด notifyResponse และ notifyRequest โดยเมธอด notifyResponse จะรองรับ Response ที่ได้รับกลับมาและเมธอด notifyRequest จะรองรับ Request จากไคลเอนท์อีกฝั่งหนึ่งผ่านทางซีพเซิร์ฟเวอร์

7.3 สถาปัตยกรรมของระบบ

ในการทำโครงงานนี้ได้ออกแบบสถาปัตยกรรมไว้ดังรูป



รูปที่ 7-3 สถาปัตยกรรมของระบบ

7.3.1 ซีพเซิร์ฟเวอร์(SIP Server) -

1. ระบบปฏิบัติการ (Operating System)

ใช้ระบบปฏิบัติการ Linux Red Hat เวอร์ชัน 9

2. โปรแกรมซีพเซิร์ฟเวอร์

ใช้ VOCAL เวอร์ชัน 1.5 เพื่อให้บริการการติดต่อสื่อสารข้ามเครือข่ายไอพีโดยใช้โปรโตคอลซีพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.2 ไคลเอนท์ (Client)

แบ่งออกเป็น 2 ส่วน คือ

- ส่วนที่ทำงานบนเครื่องคอมพิวเตอร์พีซี

1. ระบบปฏิบัติการ (Operating System)

ใช้ระบบปฏิบัติการ Windows XP Service Pack 2

2. RTC API SDK

เป็นซอฟต์แวร์ที่ช่วยในการพัฒนาโปรแกรมการติดต่อสื่อสารผ่านทางโปรโตคอลซิป
เมื่อทำการติดตั้งซอฟต์แวร์ตัวนี้ จะทำการติดตั้ง ไลบรารีต่าง ๆ ที่สำคัญที่ใช้ในการรัน โปรแกรม

- ส่วนที่ทำงานบนอุปกรณ์มือถือ

1. โปรแกรมที่รองรับการทำงาน

JAVA Runtime Environment 1.4

2. RI Binary For JSR-180 SIP API For J2ME

เป็นซอฟต์แวร์ที่ใช้ในการรันอิมูเลเตอร์ที่เขียนขึ้นมาโดยใช้ SIP API For J2ME

7.4 รูปแบบการรับส่งแพ็คเกจระหว่างเอนทิตีของฟังก์ชันการทำงาน

7.4.1 การล็อกอินเข้าสู่ระบบ (Log in)

เมื่อไคลเอนท์ต้องการที่จะล็อกอินเข้าสู่ระบบ จะทำการส่ง REGISTER ไปยังเซิร์ฟเวอร์และรอ
รับ 200 OK ตอบกลับมาจากเซิร์ฟเวอร์

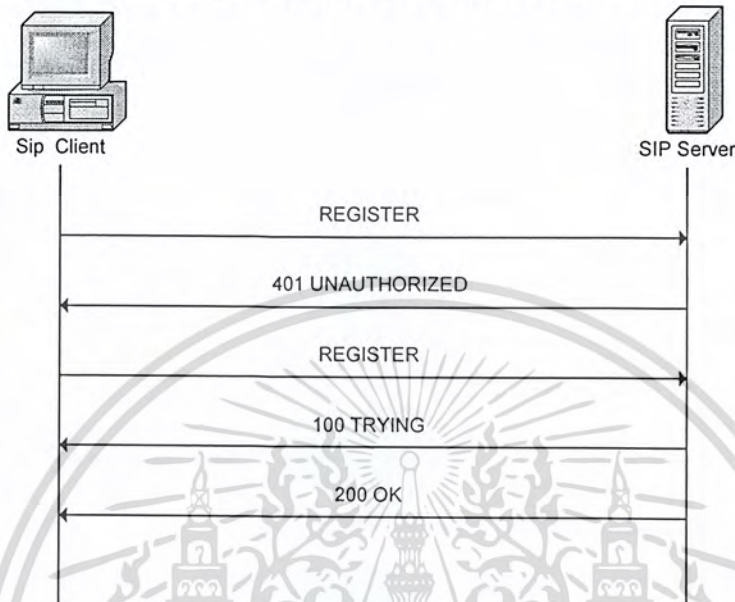


รูปที่ 7-4 รูปแบบการส่งแพ็คเกจเมื่อไคลเอนท์ทำการล็อกอินเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4.2 การพิสูจน์ตน (Authentication)

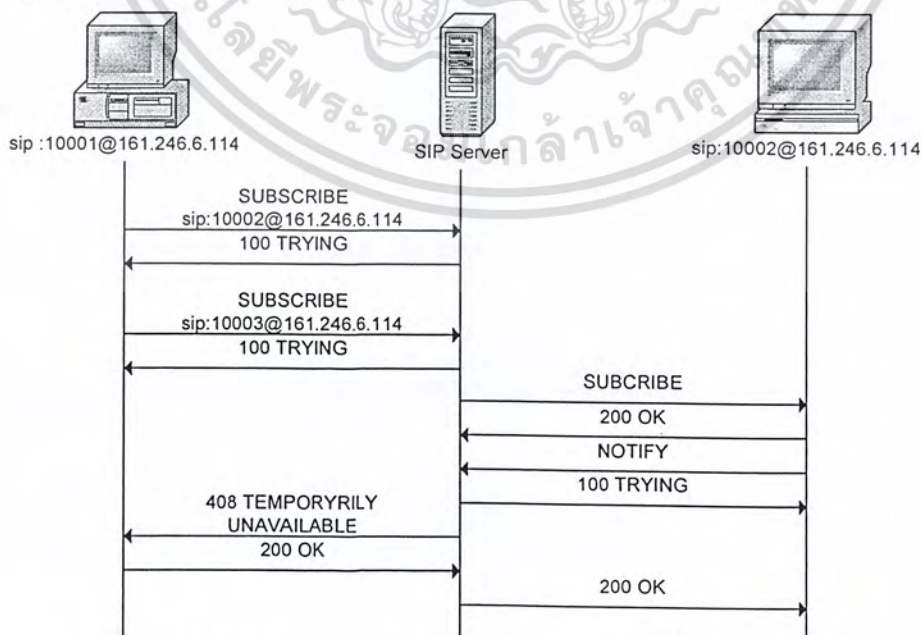
สำหรับสมาชิกที่ต้องมีการพิสูจน์ตนนั้น เมื่อไคลเอนท์ทำการล็อกอินแล้ว ไคลเอนท์จะได้รับ 401 Unauthorized ตอบกลับมาจากเซิร์ฟเวอร์ ซึ่งเป็นการบอกว่าต้องมีการใส่รหัสผ่านก่อน จากนั้นไคลเอนท์จะทำการล็อกอินอีกครั้งหนึ่ง โดยในการส่งครั้งนี้จะทำการส่งรหัสผ่านไปด้วย



รูปที่ 7-5 รูปแบบการส่งแพ็คเกจในการล็อกอินเข้าสู่ระบบสำหรับสมาชิกที่ต้องมีการพิสูจน์ตน

7.4.3 การแสดงรายชื่อผู้ติดต่อและสถานะของผู้ติดต่อ (Contact List and thier Status)

ในกรณีที่ sip:10001@161.246.6.114 ได้ทำการล็อกอินเข้าสู่ระบบเรียบร้อยแล้ว จากนั้น ถ้าในรายชื่อผู้ติดต่อของ sip:10001@161.246.6.114 มีอยู่ 2 คนคือ sip:10002@161.246.6.114 และ sip:10003@161.246.6.114



รูปที่ 7-6 รูปแบบการส่งแพ็คเกจในการแสดงรายชื่อผู้ติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

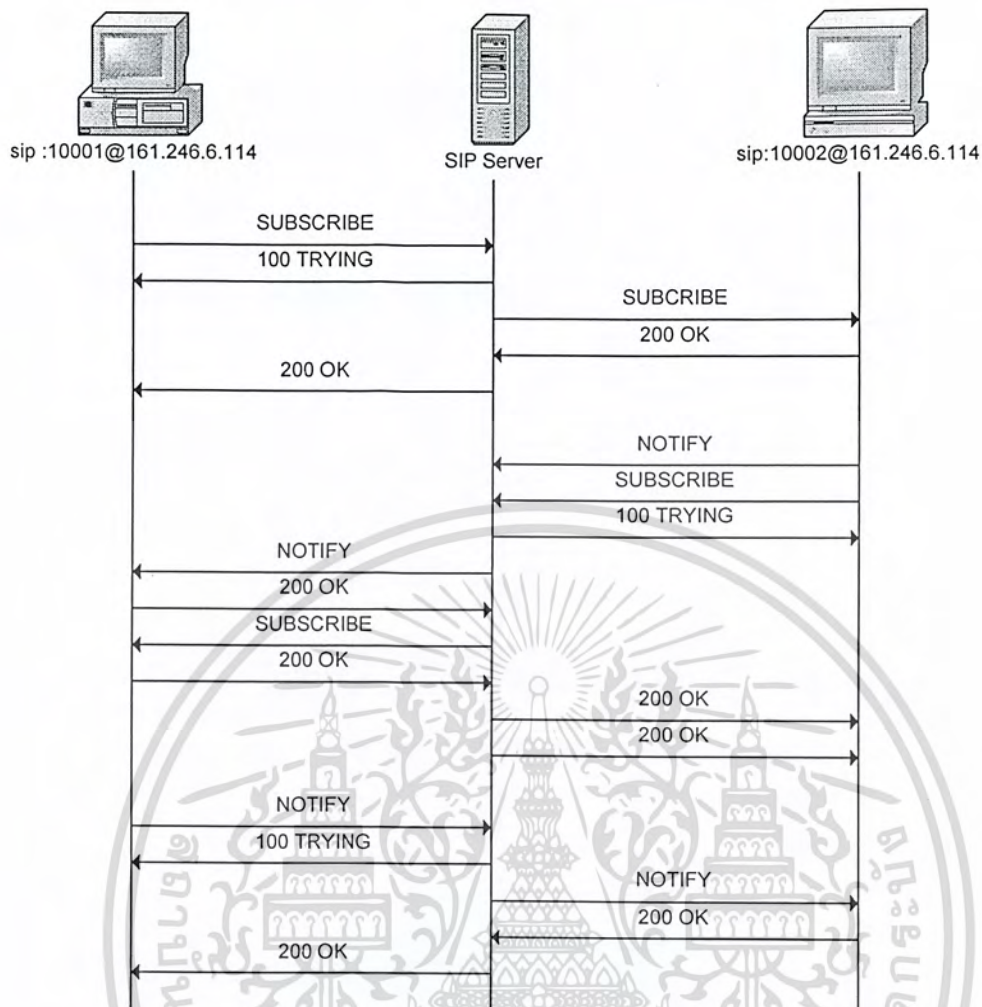
ลักษณะการรับส่งแพ็คเกจคือ sip:10001@161.246.6.114 จะทำการส่ง SUBSCRIBE sip:10002@161.246.6.114 และ sip:10003@161.246.6.114 ไปยังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์จะทำการส่ง SUBSCRIBE ไปยัง sip:10002@161.246.6.114 และ sip:10003@161.246.6.114 ในกรณีที่ sip:10002@161.246.6.114 ทำการออนไลน์อยู่ ก็จะมีการส่ง NOTIFY กลับไปบอกที่เซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์จะทำการส่ง 408 Temporarily Unavailable ให้กับ sip:10001@161.246.6.114 เพื่อบอกว่า sip:10002@161.246.6.114 นั้นมิได้ทำการออนไลน์อยู่ซึ่ง sip:10001@161.246.6.114 นั้นสามารถดูรายละเอียดได้จากแมสเสจจบบอดี

7.4.4 เพิ่ม/ลบ รายชื่อผู้ติดต่อ (Add/Remove Contact List)

การเพิ่มรายชื่อผู้ติดต่อ

เมื่อ sip:10001@161.246.6.114 ต้องการที่จะทำการเพิ่ม sip:10002@161.246.6.114 เข้าไปในรายชื่อผู้ติดต่อ sip:10001@161.246.6.114 จะทำการส่ง SUBSCRIBE ไปยัง sip:10002@161.246.6.114 ผ่านทางเซิร์ฟเวอร์

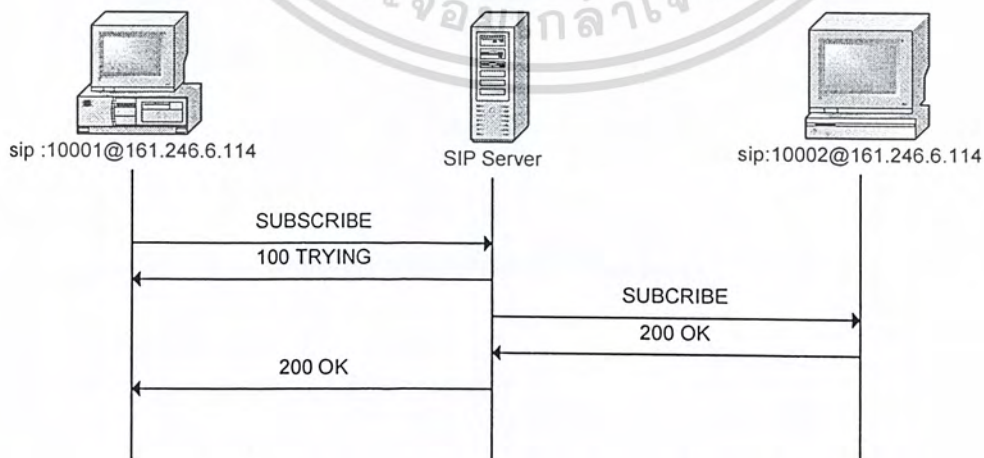
จากนั้น ถ้า sip:10002@161.246.6.114 ยอมรับคำร้องขอจาก sip:10001@161.246.6.114 sip:10002@161.246.6.114 จะทำการส่ง NOTIFY และ SUBSCRIBE กลับไป ยัง sip:10001@161.246.6.114 เมื่อข้อมูลส่งมาถึง sip:10001@161.246.6.114 ถ้า sip:10001@161.246.6.114 ยอมรับคำตอบก็จะทำการส่ง NOTIFY ไปให้ sip:10002@161.246.6.114 ผ่านทางเซิร์ฟเวอร์และรอรับ 200 OK จาก sip:10002@161.246.6.114



รูปที่ 7-7 รูปแบบการส่งแพ็คเกจในการเพิ่มรายชื่อผู้ติดต่อ

การลบรายชื่อผู้ติดต่อ

เมื่อ sip:10001@161.246.6.114 ทำการลบรายชื่อผู้ติดต่อ sip:10001@161.246.6.114 จะทำการส่ง SUBSCRIBE ไปยังผู้รับ จากนั้นจะทำการรอรับ 200 OK

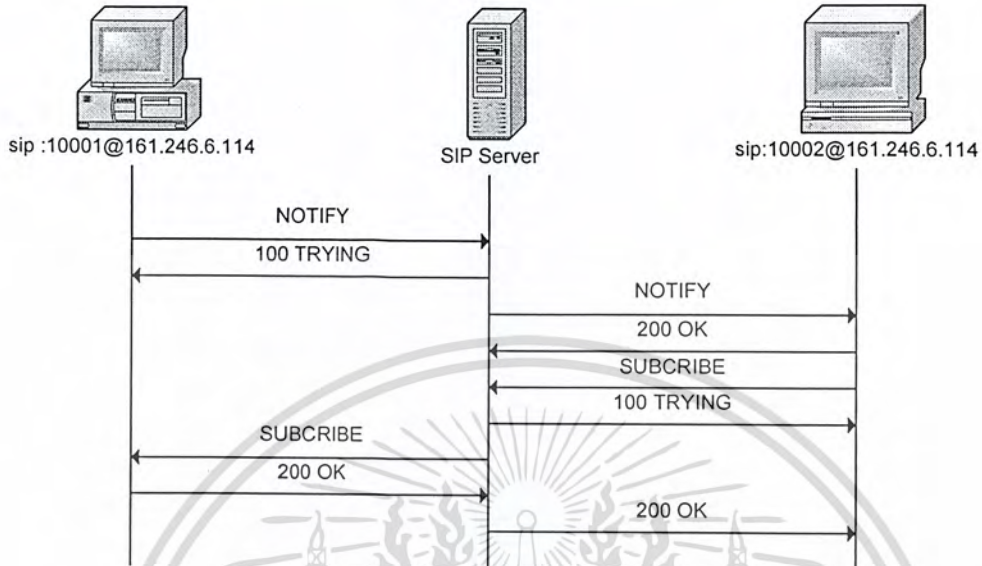


รูปที่ 7-8 รูปแบบการส่งแพ็คเกจในการลบรายชื่อผู้ติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4.5 ปกกัน / อนุญาต การติดต่อ (Block / Unblock)

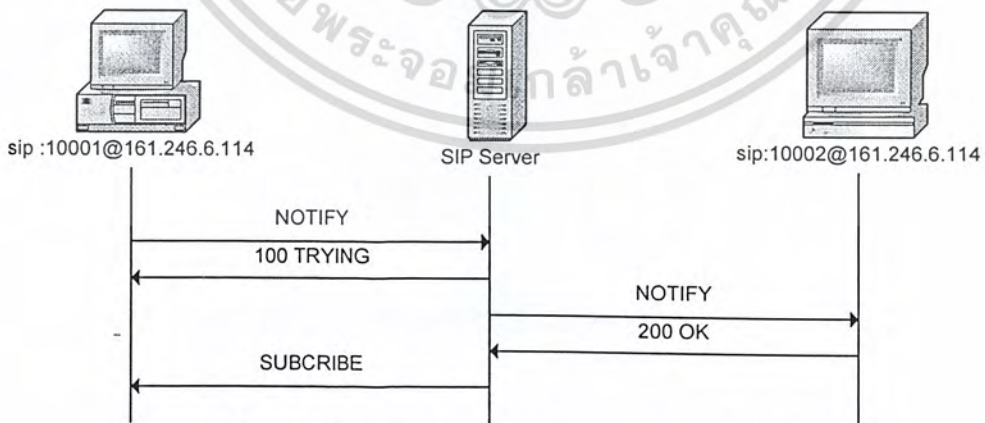
การป้องกันการติดต่อ



รูปที่ 7-9 รูปแบบการส่งแพ็คเกจในการป้องกันการติดต่อ

เมื่อ sip:10001@161.246.6.114 ได้ทำการป้องกันการติดต่อจาก sip:10002@161.246.6.114 นั้น sip:10001@161.246.6.114 ก็จะทำการส่ง NOTIFY ไปยังเซิร์ฟเวอร์เพื่อให้เซิร์ฟเวอร์ ส่ง NOTIFY ไปบอก sip:10002@161.246.6.114 ว่า sip:10001@161.246.6.114 นั้นได้ทำการออกจากระบบแล้ว จากนั้น sip:10002@161.246.6.114 จะทำการส่ง SUBSCRIBE ตอบกลับไปที่ sip:10001@161.246.6.114

การอนุญาตการติดต่อ

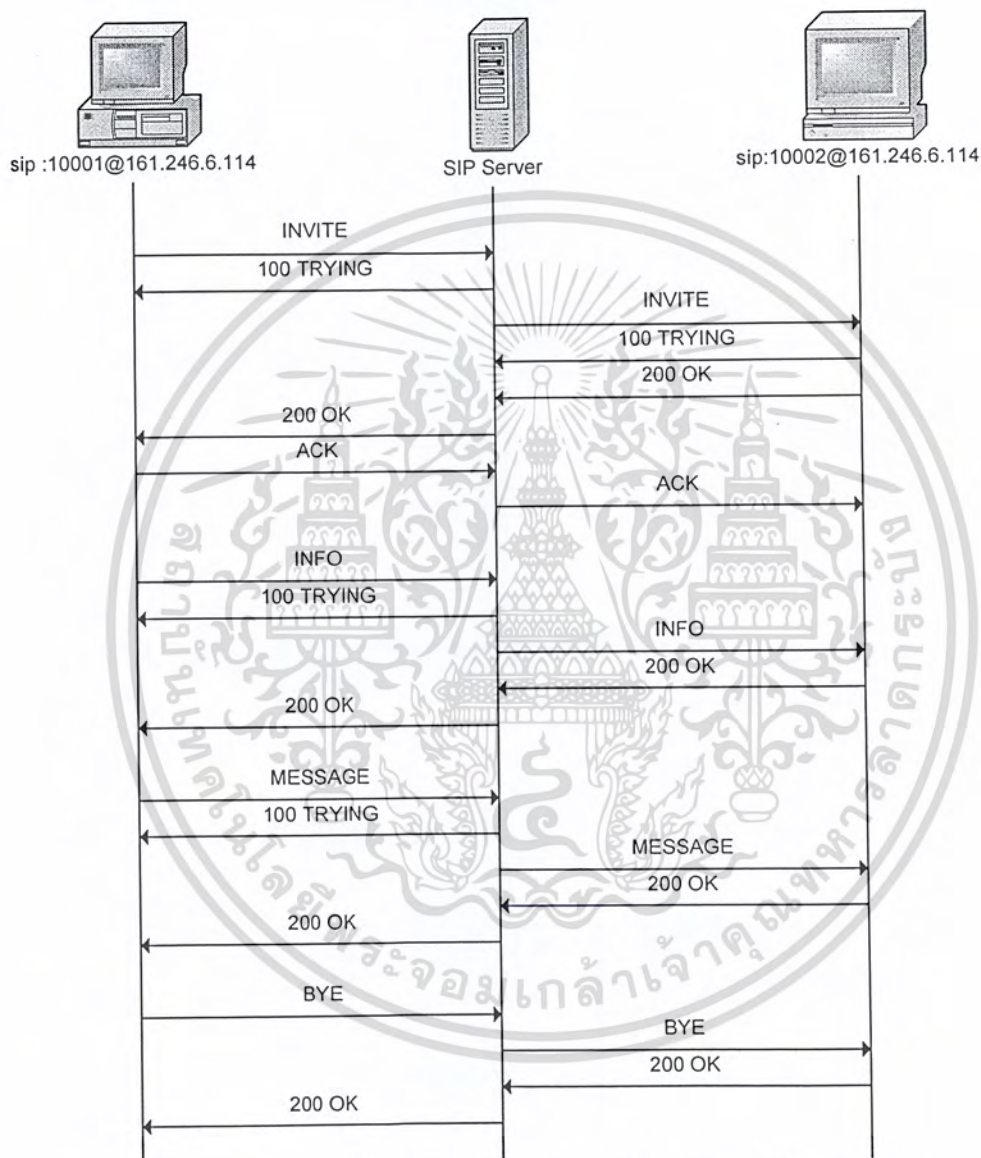


รูปที่ 7-10 รูปแบบการส่งแพ็คเกจในการอนุญาตการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ sip:10001@161.246.6.114 ต้องการที่จะอนุญาต sip:10002@161.246.6.114 หลังจากที่ทำการป้องกันการติดต่อแล้ว sip:10001@161.246.6.114 จะทำการส่ง NOTIFY ไปยัง sip:10002@161.246.6.114 ผ่านทางเซิร์ฟเวอร์ เพื่อบอกว่าตนเองนั้นมีสถานะออนไลน์แล้ว

7.4.6 การรับส่งข้อความแบบหนึ่งต่อหนึ่ง (Chat)



รูปที่ 7-11 รูปแบบการส่งแพ็คเกจในการรับส่งข้อความแบบหนึ่งต่อหนึ่ง

เมื่อ sip:10001@161.246.6.114 ต้องการที่จะส่งข้อความไปหา sip:10002@161.246.6.114 sip:10001@161.246.6.114 ก็จะทำการส่ง INVITE ไปยัง sip:10002@161.246.6.114 ผ่านทางเซิร์ฟเวอร์ จากนั้น sip:10002@161.246.6.114 จะทำการตอบตกลง 200 OK ไปยัง sip:10001@161.246.6.114 เมื่อ

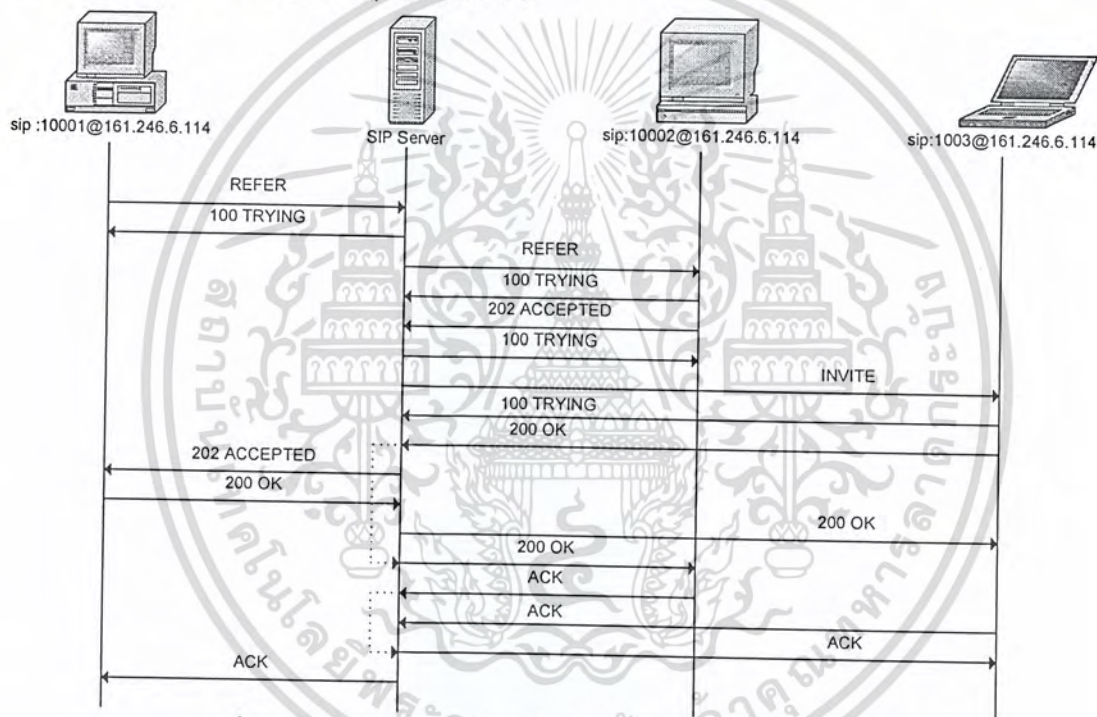
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sip:10001@161.246.6.114 ได้รับ 200 OK จะทำการส่ง ACK กลับไปให้ sip:10002@161.246.6.114 ผ่านทางเซิร์ฟเวอร์เช่นเดิม

เมื่อทั้งสองฝั่งทำการเชื่อมต่อกันผ่านทางเซิร์ฟเวอร์แล้ว เมื่อ sip:10001@161.246.6.114 ทำการส่งข้อความไปหา sip:10002@161.246.6.114 sip:10001@161.246.6.114 จะทำการส่ง INFO ไปบอก sip:10002@161.246.6.114 ก่อนเพื่อเป็นการบอกถึงข้อมูลต่าง ๆ ที่เกี่ยวข้องกับการติดต่อ จากนั้นจะทำการส่ง MESSAGE ซึ่งเก็บข้อความที่ไว้สนทนา ไปยัง sip:10002@161.246.6.114

และเมื่อ sip:10001@161.246.6.114 ทำการปิดหน้าต่างสนทนา ก็จะทำการส่ง BYE ไปบอก sip:10002@161.246.6.114 ผ่านทางเซิร์ฟเวอร์ว่าตนเองนั้นได้ตัดการเชื่อมต่อแล้ว

7.4.7 การรับส่งข้อความแบบเป็นกลุ่ม (Multi Chat)



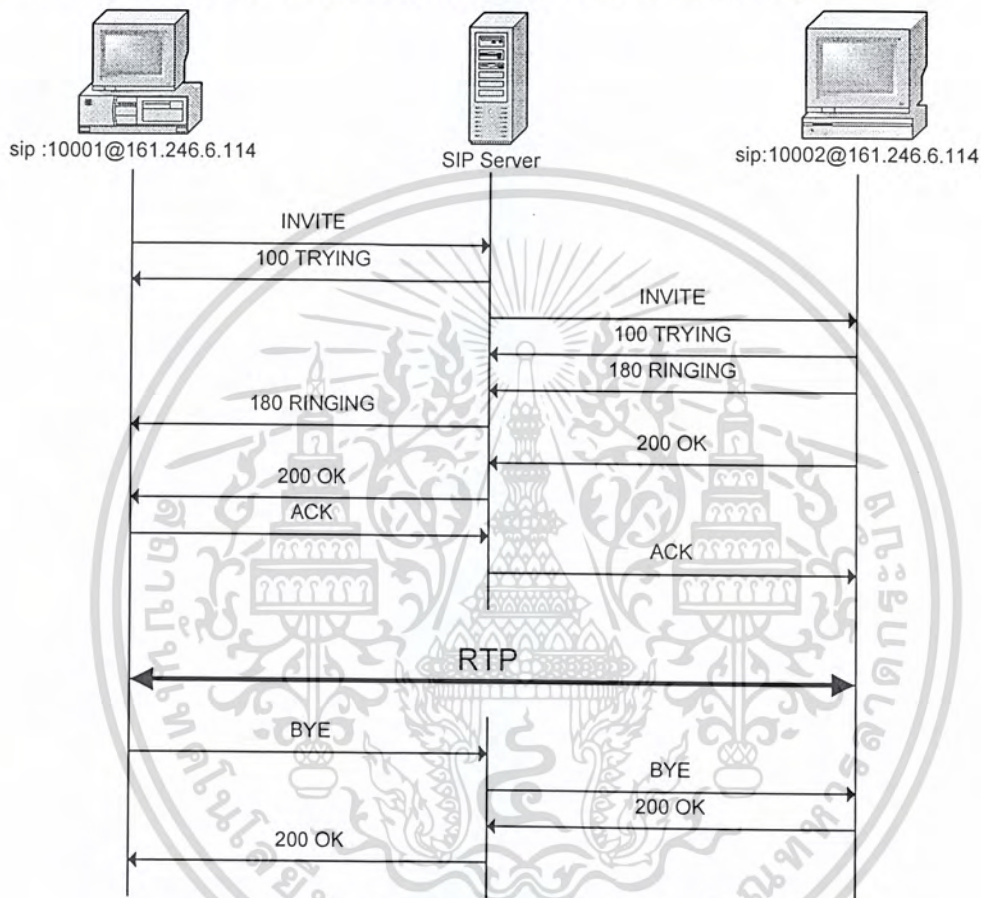
รูปที่ 7-12 รูปแบบการส่งแพ็คเกจในการรับส่งข้อความแบบเป็นกลุ่ม

ขณะที่ sip:10001@161.246.6.114 และ sip:10002@161.246.6.114 ทำการรับส่งข้อความอยู่ เมื่อ sip:10001@161.246.6.114 ต้องการที่จะทำการสนทนาเป็นกลุ่มกับ sip:10003@161.246.6.114 sip:10001@161.246.6.114 จะทำการส่ง REFER ไปหา sip:10002@161.246.6.114 ผ่านทางเซิร์ฟเวอร์ เพื่อบอกว่าจะทำการโอนการติดต่อไปยัง sip:10003@161.246.6.114 จากนั้น sip:10002@161.246.6.114 จะทำการส่ง 202 Accepted เพื่อบอกว่ายอมรับการร้องขอนี้ ไปยัง sip:10001@161.246.6.114 ผ่านทางเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์รับทราบจะทำการส่ง INVITE ไปยัง sip:10003@161.246.6.114 และเมื่อ sip:10003@161.246.6.114 ตอบตกลง (200 OK) เซิร์ฟเวอร์จะทำการส่ง 202 Accepted (INVITE จาก sip:10003@161.246.6.114) ไปยัง sip:10001@161.246.6.114 และทำการส่ง 200 OK จาก sip:10003@161.246.6.114 ไปยัง sip:10002@161.246.6.114 จากนั้น sip:10002@161.246.6.114 จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์ การค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่ง ACK ไปยัง sip:10003@161.246.6.114 ผ่านทางเซิร์ฟเวอร์ เพื่อให้ sip:10002@161.246.6.114 กับ sip:10003@161.246.6.114 จากนั้นทำการเชื่อมต่อกัน และ sip:10003@161.246.6.114 และ sip:10003@161.246.6.114 จะทำการส่ง ACK ไปให้ sip:10001@161.246.6.114 ผ่านทางเซิร์ฟเวอร์ เพื่อให้ sip:10003@161.246.6.114 และ sip:10001@161.246.6.114 ทำการเชื่อมต่อกัน

7.4.8 การติดต่อสื่อสารด้วยเสียงและภาพวิดีโอ (Audio and Video Communication)

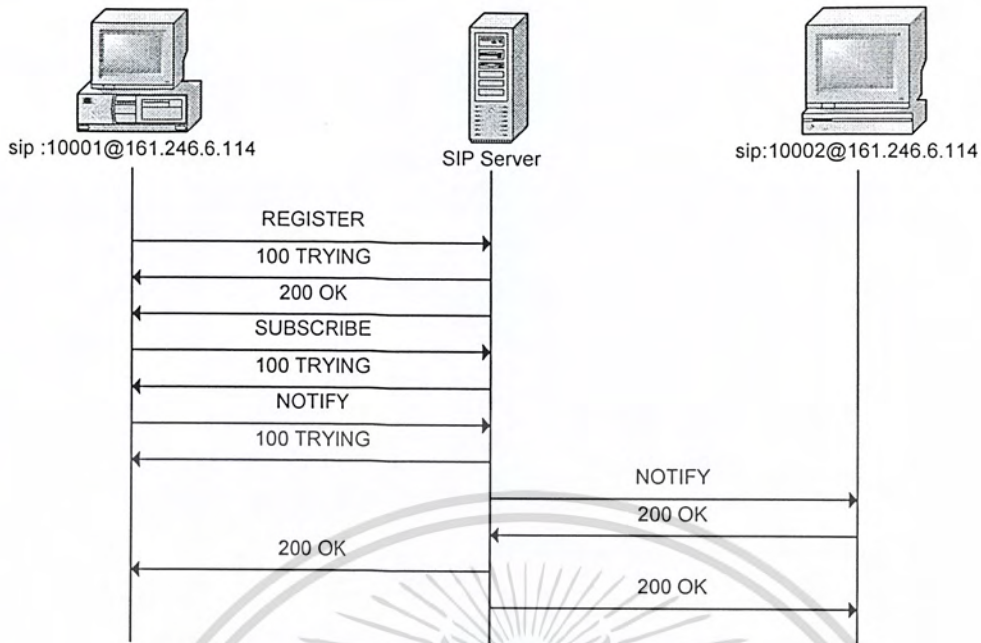


รูปที่ 7-13 รูปแบบการส่งแพ็คเกจในการติดต่อสื่อสารด้วยเสียงและภาพวิดีโอ

7.4.9 การล็อกเอาต์ออกจากระบบ (Log out)

การล็อกเอาต์ออกจากระบบจะทำโดย ทำการส่ง REGISTER ไปยังเซิร์ฟเวอร์ จากนั้นจะทำการส่ง SUBSCRIBE และ NOTIFY ไปบอกทุกคนในรายชื่อผู้ติดต่อที่ออนไลน์อยู่ จากนั้นทำการรอ รับ 200 OK จากนั้นก็จะทำการส่ง 200 OK กลับอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-14 รูปแบบการส่งแพ็คเกจในการลือกเอาที่ออกจากระบบ

7.5 เครื่องมือที่ใช้ในการพัฒนา

- C++ Language
- Microsoft Visual Studio 6.0
- RTC (Real Time Communication) API SDK v 1.2
- Microsoft Platform SDK for Windows XP SP 2
- Ethereal
- j2sdk 1.4
- Wireless Toolkit 2.2
- JSR-180 SIP API For J2ME
- Webcam
- Microphone/Speaker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

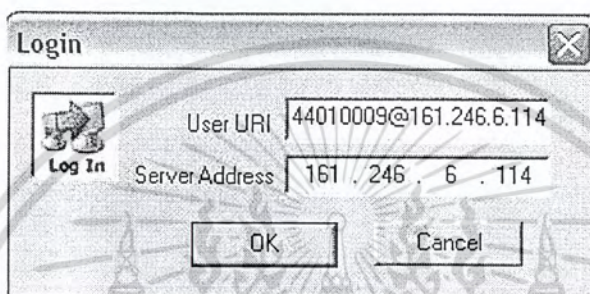
บทที่ 8

การใช้งานโปรแกรม

8.1 การใช้โปรแกรมบนพีซี

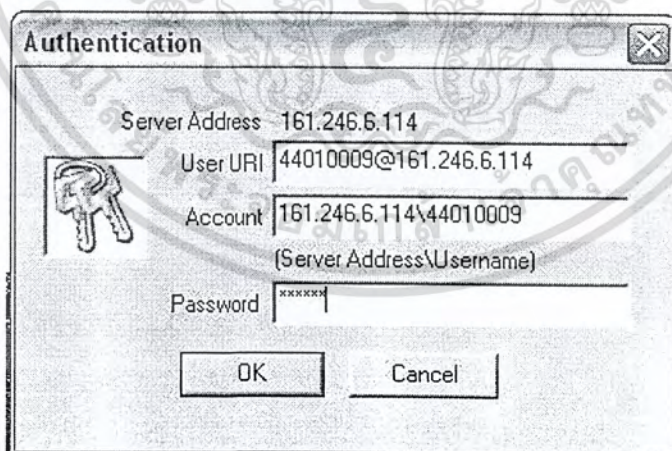
8.1.1 การ login

เมื่อทำการเข้าโปรแกรมจะพบกับหน้าต่างล็อกอินซึ่งเราจะต้องทำการใส่ URI ของตัวเรา และ แอดเดรสของเซิร์ฟเวอร์เพื่อใช้ในการล็อกอินเพื่อเข้าสู่ระบบ



รูปที่ 8-1 แสดงหน้าต่างการล็อกอิน

การล็อกอินอาจจะสามารถล็อกอินเข้าไปได้เลย หรืออาจจะต้องมีการพิสูจน์ตัวตนก่อนที่จะสามารถเข้าสู่ระบบได้

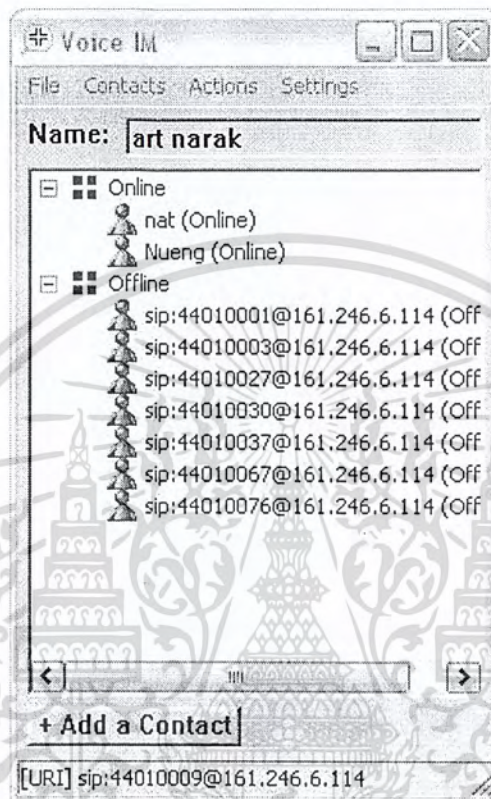


รูปที่ 8-2 แสดงหน้าต่างการพิสูจน์ตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

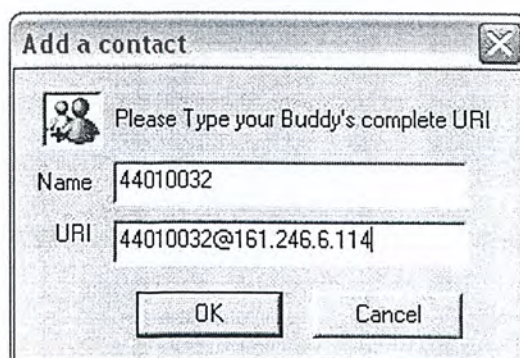
8.1.2 การเพิ่มรายชื่อ

เมื่อเราทำการพิสูจน์ตนเรียบร้อยแล้วเราก็จะสามารถเข้าสู่ระบบได้ โดยเราจะเห็นหน้าต่างหลักที่ใช้แสดงรายชื่อของผู้ใช้คนอื่นทั้งที่ออนไลน์อยู่และที่ออฟไลน์อยู่



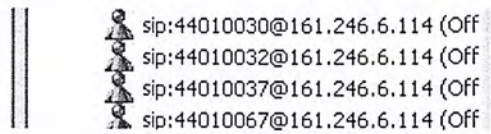
รูปที่ 8-3 แสดงหน้าต่างหลักสำหรับแสดงชื่อผู้ใช้อื่นในระบบ

เราจะสามารถทำการเพิ่มรายชื่อผู้ที่ต้องการจะติดต่อได้ โดยการเลือก Add a Contact ที่เมนูบาร์ หรือใช้ปุ่ม + Add a Contact ในหน้าต่างหลักก็จะสามารถเพิ่มรายชื่อคนอื่นที่เราต้องการจะติดต่อด้วยได้



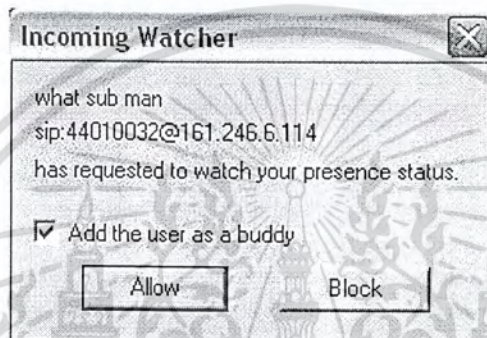
รูปที่ 8-4 แสดงหน้าต่างใช้ในการเพิ่มรายชื่อผู้ที่ต้องติดต่อด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-5 แสดงรายชื่อที่ถูกเพิ่มเข้าไป

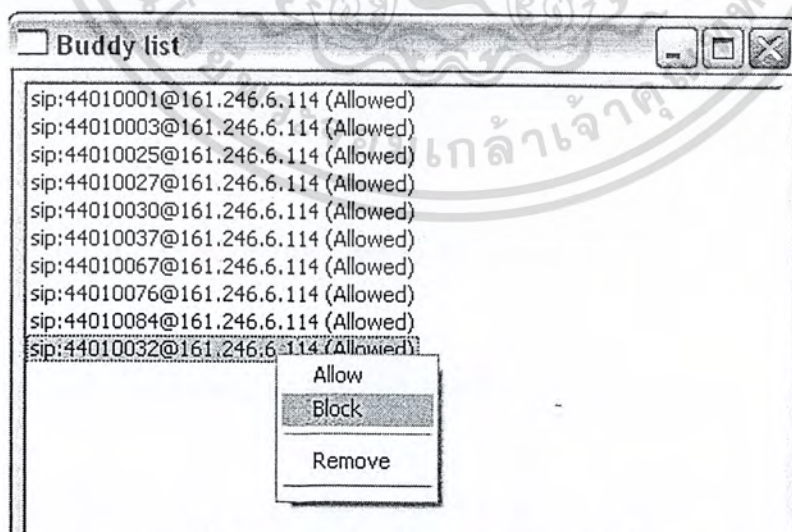
เมื่อคนที่เราต้องการที่จะเพิ่มชื่อ ได้ทำการตอบรับคำเชิญของเราก็จะมีข้อความส่งกลับมาบอกเรา



รูปที่ 8-6 แสดงหน้าต่างที่แสดงการตอบรับคำร้องขอการติดต่อ

8.1.3 การ block/unblock

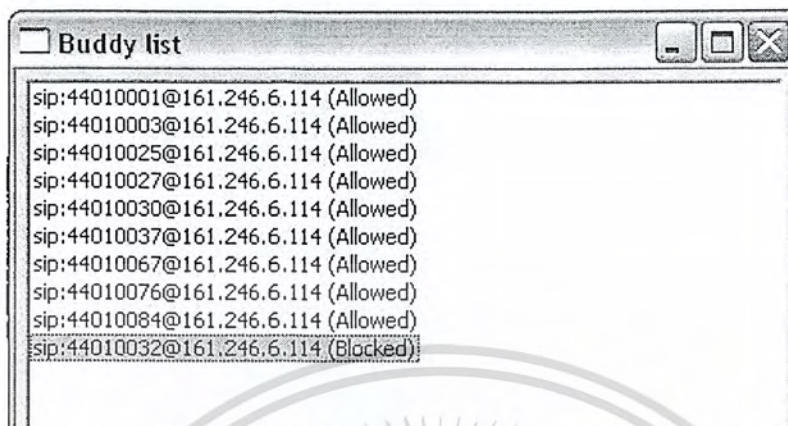
เราสามารถที่จะ block บุคคลที่เราไม่ต้องการจะคุยด้วยได้ โดยเลือกฟังก์ชันได้ที่เมนูบาร์



รูปที่ 8-7 แสดงรูปการทำการป้องกันการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

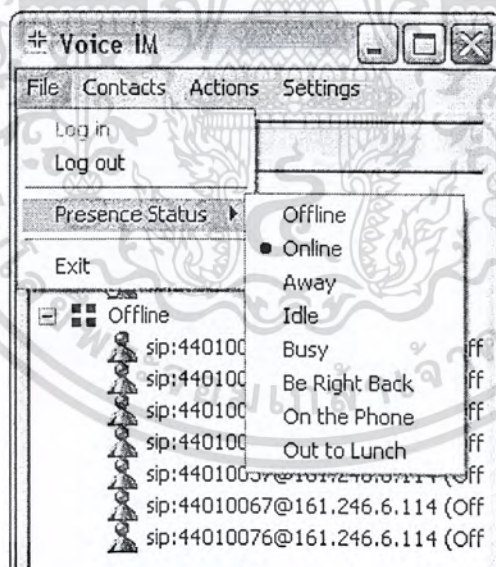
เมื่อเราทำการ block แล้ว รายชื่อในรายการจะเปลี่ยนไป ซึ่งเราสามารถที่จะทำการ unblock คนที่ถูก block อยู่ได้ด้วยวิธีเดียวกันแต่ให้ทำการเลือก Allow แทนที่จะเป็น Block



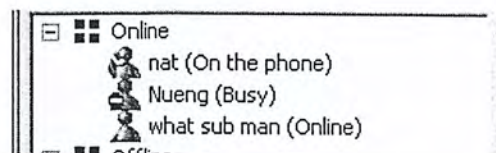
รูปที่ 8-8 แสดงรูป หลังจากทำการป้องกันการติดต่อแล้ว

8.1.4 การเปลี่ยนสถานะของผู้ใช้

ผู้ใช้สามารถที่จะเปลี่ยนสถานะของตัวเองเพื่อเป็นการบอกให้ผู้อื่นทราบได้ว่าในขณะนั้นผู้ใช้ทำอะไรอยู่ อย่างคร่าวๆ เราสามารถที่จะเปลี่ยนสถานะได้โดยการเลือก ที่เมนู file แล้วเลือกไปที่ Presence Status



รูปที่ 8-9 การเปลี่ยนสถานะของผู้ใช้

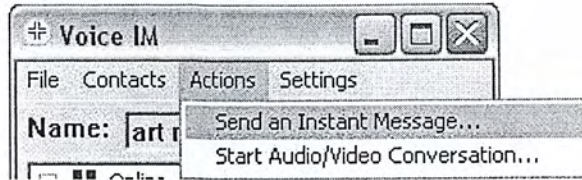


รูปที่ 8-10 แสดงสถานะของผู้ใช้ที่ได้เปลี่ยนแล้ว

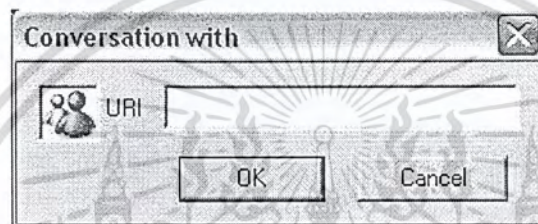
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.5 การส่ง Instant Message

เราสามารถที่จะส่ง Instant Message ไปคุยกับคนอื่นได้ ซึ่งทำได้โดยการเลือก menu Actions แล้วทำการเลือก Send an Instant Message แล้วก็ใส่ URI ของผู้ใช้ที่เราจะต้องการติดต่อด้วย ก็จะสามารถติดต่อกับผู้ใช้คนอื่นได้

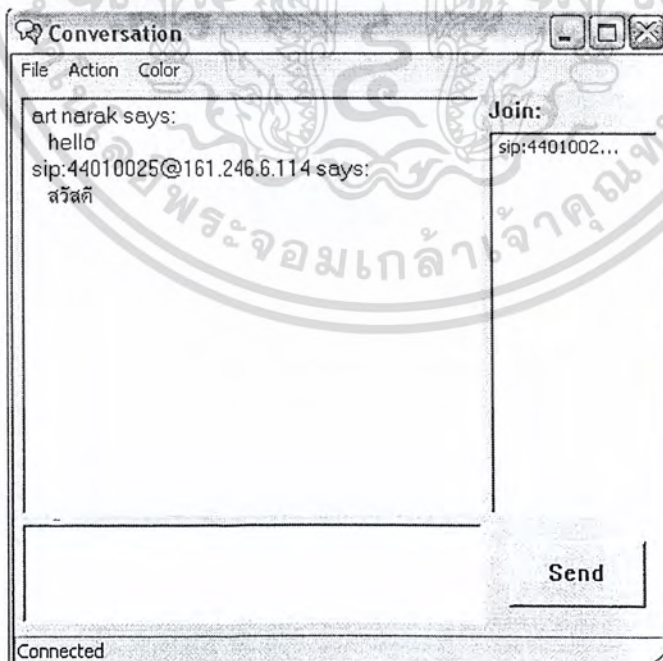


รูปที่ 8-11 แสดงวิธีการติดต่อด้วยการส่งข้อความ



รูปที่ 8-12 แสดงหน้าต่างสำหรับใส่ URI ของผู้ที่ต้องการจะติดต่อด้วย

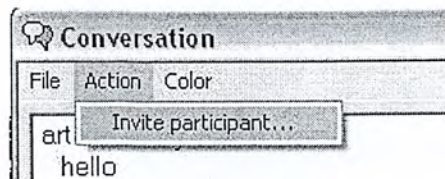
เมื่อใส่ URI แล้วจะสามารถทำการติดต่อกับผู้อื่นโดยการส่ง Instant Message ได้ โดยจะเกิดหน้าต่างที่ใช้สำหรับส่งข้อความขึ้นมา



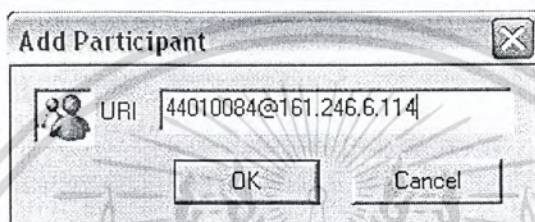
รูปที่ 8-13 แสดงหน้าต่างแสดงการส่งข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

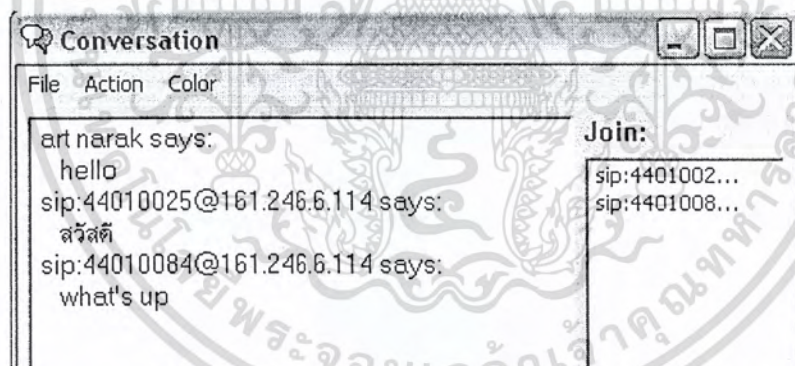
เราสามารถที่จะทำการเพิ่มสมาชิกภายในหน้าต่างนั้นได้โดยการเลือกที่ menu Action แล้วเลือก Invite participant ซึ่งจะเกิดหน้าต่างสำหรับให้ใส่ URI ของผู้ที่ต้องการเพิ่มให้เข้ามาสนทนาด้วย



รูปที่ 8-14 แสดงการเพิ่มผู้ที่อยากจะสนทนาด้วยเข้าไปอีก



รูปที่ 8-15 แสดงหน้าต่างสำหรับใส่ URI ของคนที่ต้องการให้มาสนทนาด้วย

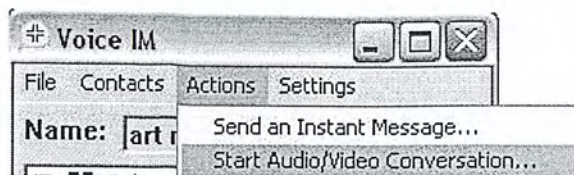


รูปที่ 8-16 แสดงหน้าต่างหลังจากเพิ่มผู้สนทนาเข้ามา

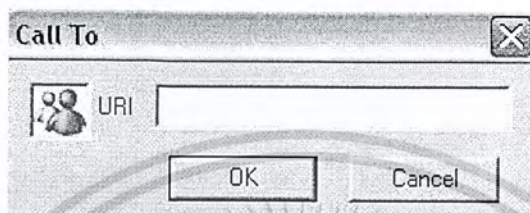
8.1.6 การติดต่อโดยการส่งเสียง

เราสามารถที่จะทำการติดต่อสื่อสารโดยการใช้เสียงได้ ซึ่งจะสร้างการเชื่อมต่อได้โดยการเลือกที่ menu Actions ในหน้าต่างหลัก แล้วทำการเลือก Start Audio/Video Conversation ซึ่งจะเกิดหน้าต่างสำหรับให้ใส่ URI ของคนที่เราต้องการที่จะ call ไปหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

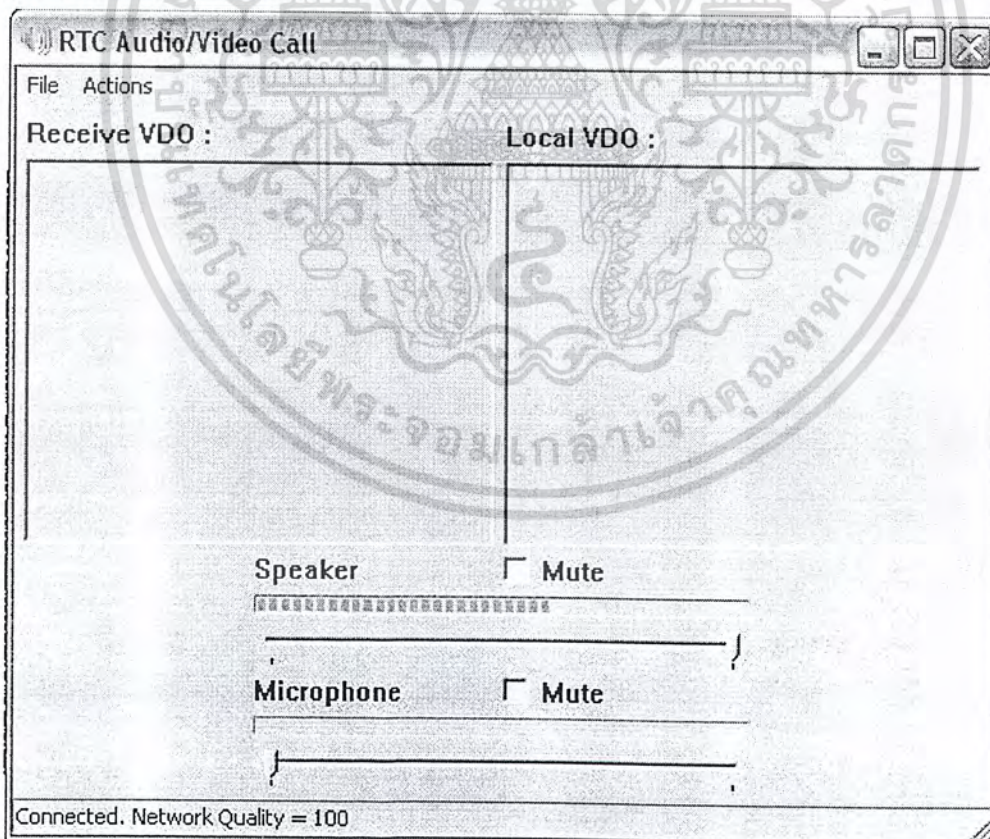


รูปที่ 8-17 แสดงหน้าต่างในการร้องขอการติดต่อ เพื่อสื่อสารด้วยเสียง



รูปที่ 8-18 แสดงหน้าต่างสำหรับใส่ URI ของคนที่เราต้องการติดต่อด้วย

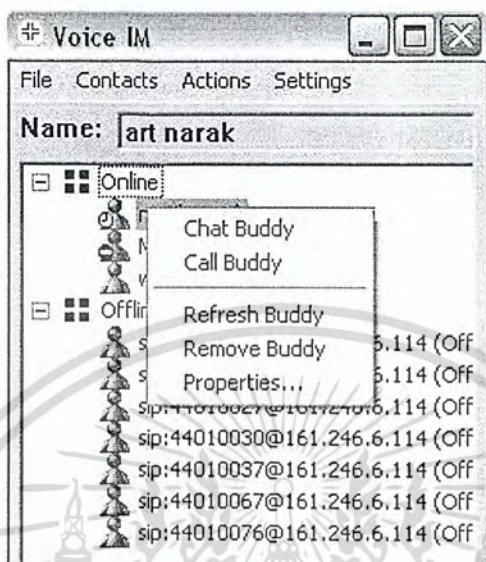
เมื่อเราทำการกำหนด URI ของคนที่เราต้องการจะ call ไปหาเสร็จเรียบร้อยแล้ว จะเกิดหน้าต่างที่ใช้ในการสื่อสาร โดยใช้เสียงขึ้นมา



รูปที่ 8-19 แสดงหน้าต่างที่ใช้ในขณะที่ทำการติดต่อโดยใช้เสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

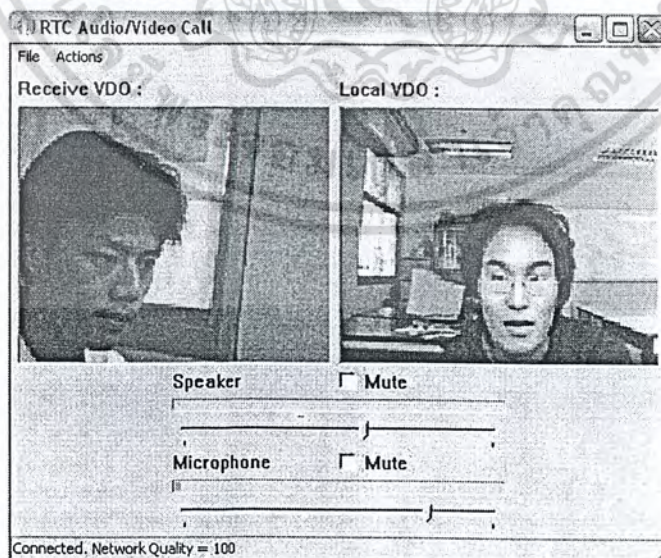
ทั้งการติดต่อโดยใช้ Instant Message หรือ เสียง นั้นเราสามารถทำการเชื่อมต่อกับบุคคลคนนั้นได้โดยตรงผ่านทางชื่อที่อยู่ใน contact list โดยทำการคลิกขวาที่ชื่อของคนที่เราต้องการจะติดต่อก็จะสามารถเลือกได้ว่าจะทำการสร้างการเชื่อมต่อโดยใช้ Instant Message หรือ เสียงได้



รูปที่ 8-20 แสดงเลือกการติดต่อผ่านทางชื่อของผู้ใช้

8.1.7 การติดต่อโดยการส่งวิดีโอ

การติดต่อโดยผ่านวิดีโอสามารถทำได้เหมือนกับการติดต่อโดยผ่านเสียง คือทำการเลือก Start Audio/Video Conversation ซึ่งจะเกิดหน้าต่างสำหรับให้ใส่ URI ของคนที่เราต้องการที่จะ ติดต่อกับซึ่งเมื่อใส่ URI แล้วก็จะเกิดหน้าต่างที่ทำการติดต่อผ่านวิดีโอขึ้นมา



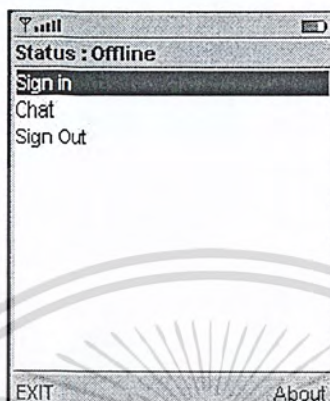
รูปที่ 8-21 แสดงหน้าต่างการติดต่อผ่านวิดีโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2 การใช้โปรแกรมบนมือถือ

8.2.1 การ login

เมื่อเราเริ่มต้นใช้โปรแกรมเราจะอยู่ในสถานะ Offline เราสามารถที่จะใช้โปรแกรมได้โดยทำการเลือกที่ Sign in



รูปที่ 8-22 แสดงหน้าจอเริ่มต้นก่อนการล็อกอิน

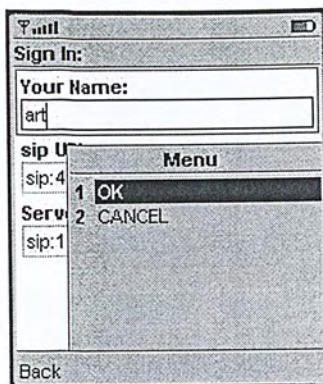
เมื่อเราทำการ Sign in เข้าไปแล้วเราจะต้องทำการใส่ URI และ Address ของเซิร์ฟเวอร์ และทำการกำหนดชื่อของเราที่ใช้ในการติดต่อ



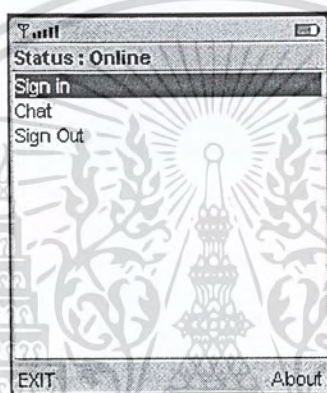
รูปที่ 8-23 แสดงหน้าจอการทำการล็อกอิน

เมื่อเราใส่ข้อมูลที่ต้องใช้แล้วก็ทำการเลือก Menu ซึ่งถ้าเลือก OK ก็จะเข้าสู่สถานะ Online แต่ถ้าเลือก CANCEL ก็จะทำการ Clear ข้อมูลที่เราได้ใส่ไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-24 แสดงหน้าจอยืนยันข้อมูลที่ใช้ในการล็อกอิน



รูปที่ 8-25 แสดงหน้าจอสถานะออนไลน์

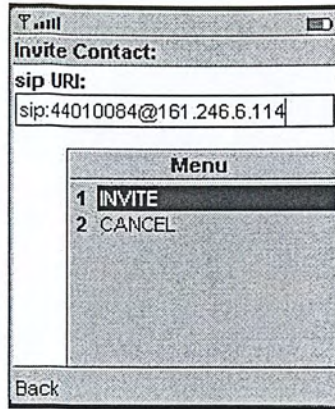
8.2.2 การร้องขอการติดต่อ

เมื่อเราต้องการที่จะทำการติดต่อเราสามารถติดต่อกับบุคคลเหล่านั้นได้โดยการทำการใส่ URI ของบุคคลที่เราอยากจะติดต่อด้วยแล้วทำการเลือกที่ปุ่ม Menu แล้วเลือก INVITE เพื่อทำการร้องขอการติดต่อไป



รูปที่ 8-26 แสดงหน้าก่อนการทำการร้องขอการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8-27 แสดงหน้าจอยืนยันการร้องขอการติดต่อ



รูปที่ 8-28 แสดงหน้าจอขณะทำการร้องขอการติดต่อ

เมื่อเราทำการ Invite แล้วทางฝั่งของผู้ที่ถูก Invite ก็จะมีข้อความแสดงการร้องขอมาโดยแสดงเป็น URI ของผู้ที่ติดต่อมา



รูปที่ 8-29 แสดงหน้าจอเมื่อมีคนทำการร้องขอการติดต่อมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

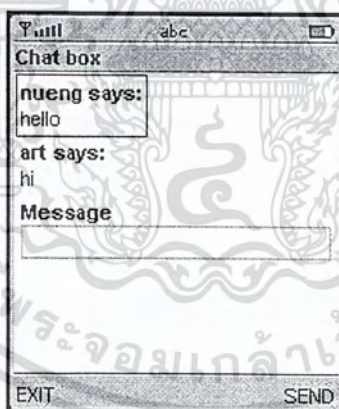
เมื่อผู้ที่ทำการ Invite มายกเลิกการร้องขอทำการเชื่อมต่อก็จะเกิดข้อความทางฝั่งผู้รับดังรูป 8-30



รูปที่ 8-30 แสดงหน้าจอการยกเลิกการร้องขอการติดต่อ

8.2.3 การส่งข้อความแบบหนึ่งต่อหนึ่ง

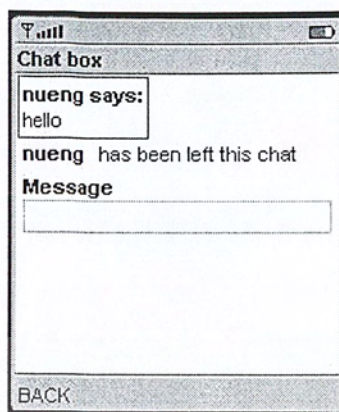
เมื่อเราทำการ Invite แล้ว เราจะสามารถทำการส่งข้อความเพื่อติดต่อกับบุคคลที่เราทำการ Invite ไปแล้วด้วยได้ โดยทำการพิมพ์ข้อความที่ต้องการจะส่งไป เมื่อต้องการที่จะส่งให้กลุ่ม SEND ก็จะเป็นการส่งข้อความไปให้ผู้ที่เราติดต่อด้วย



รูปที่ 8-31 แสดงหน้าจอการติดต่อสื่อสารโดยการส่งข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าผู้ที่เราทำการสนทนาด้วยได้ทำการออกจากระบบจะมีข้อความบอกให้กับอีกฝ่ายให้ทราบด้วย



รูปที่ 8-32 แสดงหน้าจอเมื่อผู้สนทนาด้วยออกจากระบบแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

บทวิจารณ์และสรุป

9.1 สรุปและวิจารณ์ผลของการทำโครงการ

การทำงานของโปรแกรมบนส่วนของคอมพิวเตอร์ที่ใช้นั้น สามารถที่จะทำการส่งข้อความติดต่อระหว่างกันโดยสามารถที่จะเพิ่มผู้ร่วมการสนทนาเข้าไปได้ทำให้สามารถที่จะทำส่งข้อความหาผู้ใช้คนอื่นได้หลายคนในการส่งข้อความเพียงครั้งเดียว และยังสามารถทำการติดต่อแบบมัลติมีเดียได้ ไม่ว่าจะเป็นการติดต่อโดยการใช้เสียงหรือการติดต่อผ่านทางวิดีโอ ทำให้ผู้ใช้สามารถที่จะมองเห็นคู่สนทนาได้มีผลทำให้การติดต่อสื่อสารเหมือนจริงมากยิ่งขึ้น และยังมีฟังก์ชันการทำงานอื่นๆที่ช่วยอำนวยความสะดวกต่างๆเช่น การเพิ่มหรือลบรายชื่อผู้ที่เราต้องการติดต่อด้วยได้ สามารถที่จะทำการกำหนดสถานะของตัวผู้ใช้ทำให้ทราบได้ว่าคู่สนทนาที่เราต้องการจะคุยด้วยนั้นพร้อมที่จะติดต่อกับเราหรือไม่ หรือสามารถทำการ block บุคคลที่เราไม่ต้องการจะติดต่อด้วยได้

ส่วนการทำงาน โปรแกรมบนอุปกรณ์มือถือนั้นสามารถทำการส่งข้อความติดต่อระหว่างกันโดยทำการใส่แอดเดรสของผู้ที่เราต้องการที่จะติดต่อด้วยและรอการอนุญาตจากผู้ที่ถูกเรียก โดยการทำงานทั้งหมดจะทำการติดต่อกันผ่านทางซีพียูเซอร์เวอร์

จากการพัฒนาโปรแกรมในโครงการนั้นประสบปัญหาค่อนข้างมากในเรื่องที่เกี่ยวข้องกับการเลือกนำเทคโนโลยีมาประยุกต์ใช้ เพราะเนื่องจากว่าการพัฒนาโปรแกรมโดยใช้ซีพียูโปรโตคอลนั้นยังไม่เป็นที่แพร่หลายมาก การสนับสนุนการพัฒนาจึงค่อนข้างมาก เช่น การพัฒนาโปรแกรมในส่วนของอุปกรณ์มือถือนั้นเราไม่สามารถพัฒนาโปรแกรมให้สามารถทำการติดต่อสื่อสารด้วยเสียงได้ เนื่องจากว่าการติดต่อสื่อสารด้วยเสียงนั้นจำเป็นต้องใช้โปรโตคอล RTP ในการรับส่งข้อมูล แต่อิมูเลเตอร์ไม่สนับสนุนการทำงานทำให้ไม่สามารถพัฒนาฟังก์ชันการทำงานในส่วนนี้ได้

9.2 แนวทางการพัฒนาต่อ

สิ่งที่ควรจะทำการพัฒนาต่อไปในอนาคตนั้น คือ ควรทำการพัฒนาในส่วนของซีพียูเซอร์ให้สามารถทำการติดต่อกับส่วนของไคลเอนท์ ได้ดียิ่งขึ้น เช่น สามารถเก็บรายชื่อผู้ที่ติดต่อด้วย สามารถเก็บข้อมูลของผู้ใช้ได้ สามารถทำการเปลี่ยนรหัสผ่านด้วยตนเองได้ ขึ้นต่อไปขึ้นสามารถที่จะทำการติดต่อข้ามแพลตฟอร์มได้ และสิ่งที่น่าสนใจอีกอย่างหนึ่ง คือ สามารถทำการติดต่อแบบมัลติมีเดีย แบบ 3 สายได้ ซึ่งจะเพิ่มความสะดวกสบายในการติดต่อสื่อสารให้เหมือนจริงมากยิ่งขึ้น

ภาคผนวก ก

การติดตั้ง vocal เวอร์ชัน 1.5

Vocal เวอร์ชัน 1.5 นี้ สนับสนุนแพลตฟอร์มต่าง ๆ มากมาย รวมถึง ลินุกซ์ และ โซลาริสโดยเนื้อหาในเอกสารนี้จะกล่าวถึงรายละเอียดและวิธีการติดตั้งบนระบบปฏิบัติการลินุกซ์เท่านั้น ซึ่งความต้องการของระบบในการติดตั้งมีรายละเอียดดังนี้

ความต้องการทางด้านฮาร์ดแวร์

- CPU 533 MHz , Intel Pentium II PC processor
- RAM 128 MB
- 2 GB of hard disk space

ความต้องการทางด้านซอฟต์แวร์

- Linux, Red Hat Version 6.2 เป็นต้นไป
- Apache Server
- Java Run-time Environment เวอร์ชัน 1.3 ขึ้นไป
- g++ Compiler เวอร์ชัน 2.91.66 เป็นต้นไป

ความต้องการทางด้านเน็ตเวิร์ค

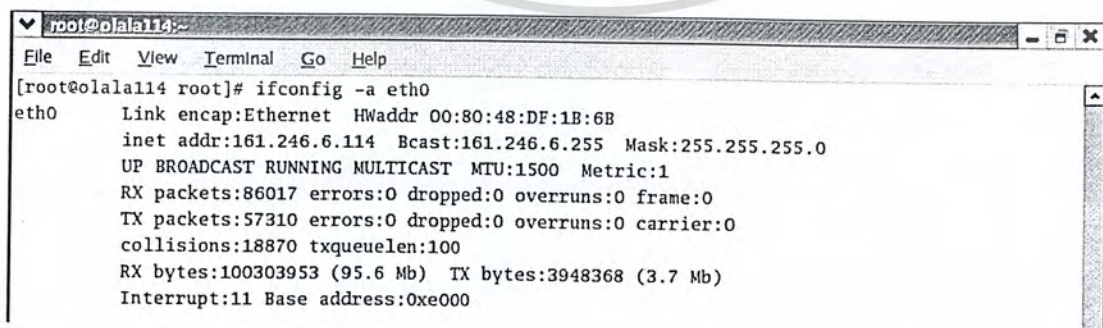
- มี ip ที่ไม่ใช่ loopback ip (127.0.0.1)

วิธีการตรวจสอบ

ที่ Terminal ใช้คำสั่ง

```
ifconfig -a eth0
```

จะได้ผลดังนี้



```

root@olala114:~
File Edit View Terminal Go Help
[root@olala114 root]# ifconfig -a eth0
eth0      Link encap:Ethernet  HWaddr 00:80:48:DF:1B:6B
          inet addr:161.246.6.114  Bcast:161.246.6.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:86017 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57310 errors:0 dropped:0 overruns:0 carrier:0
          collisions:18870 txqueuelen:100
          RX bytes:100303953 (95.6 Mb)  TX bytes:3948368 (3.7 Mb)
          Interrupt:11 Base address:0xe000
  
```

รูปที่ ก-1 ผลลัพธ์ของคำสั่ง `ifconfig -a eth0`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ : ถ้า inet addr: เป็น 127.0.0.1 ให้ทำการเซต ip ของเครื่อง

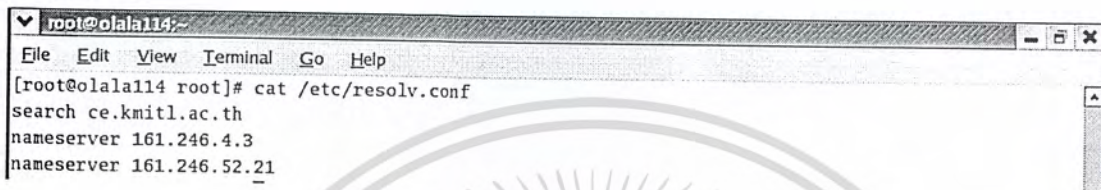
- ติดตั้งค่า DNS ไว้แล้ว

วิธีการตรวจสอบ

ที่ Terminal ใช้คำสั่ง

cat /etc/resolv.conf

จะได้ผลดังนี้



```

root@olala114:~# cat /etc/resolv.conf
search ce.kmitl.ac.th
nameserver 161.246.4.3
nameserver 161.246.52.21
  
```

รูปที่ ก-2 ผลลัพธ์ของคำสั่ง cat /etc/resolv.conf

หมายเหตุ : ถ้าไม่ปรากฏค่า nameserver ให้ทำการติดตั้งค่า DNS Server

- ติดตั้งค่า hostname ไว้แล้ว

วิธีการตรวจสอบ

ที่ Terminal ใช้คำสั่ง

hostname

จะได้ผลดังนี้



```

root@olala114:~# hostname
olala114.ce.kmitl.ac.th
  
```

รูปที่ ก-3 ผลลัพธ์ของคำสั่ง hostname

การติดตั้ง Vocal และปรับแต่งค่าต่าง ๆ ของระบบ

1. ทำการล็อกอินด้วยยูสเซอร์ root
2. ทำการดาวน์โหลด Vocal เวอร์ชัน 1.5 จาก

<http://www.vovida.org/applications/downloads/vocal/>

3. ไปที่ไดเรกทอรีที่ดาวน์โหลดมาแล้วใช้คำสั่ง tar ดังนี้

tar -vxzf vocal-1.5.0.tar.gz

4. จากนั้นจะได้ไฟล์เดอร์ที่ชื่อว่า vocal ให้เข้าไปในไฟล์เดอร์นี้แล้วทำการรันไฟล์ configure ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cd vocal
```

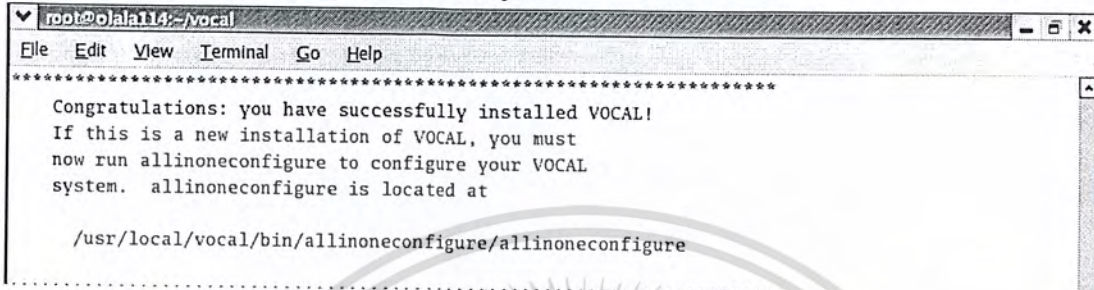
```
./configure
```

5. จากนั้นใช้คำสั่ง

```
make
```

```
make install
```

เมื่อโปรแกรมทำการติดตั้งเสร็จจะขึ้นหน้าจอดังรูป



```
root@olala114:~/vocal
File Edit View Terminal Go Help
*****
Congratulations: you have successfully installed VOCAL!
If this is a new installation of VOCAL, you must
now run allinoneconfigure to configure your VOCAL
system. allinoneconfigure is located at

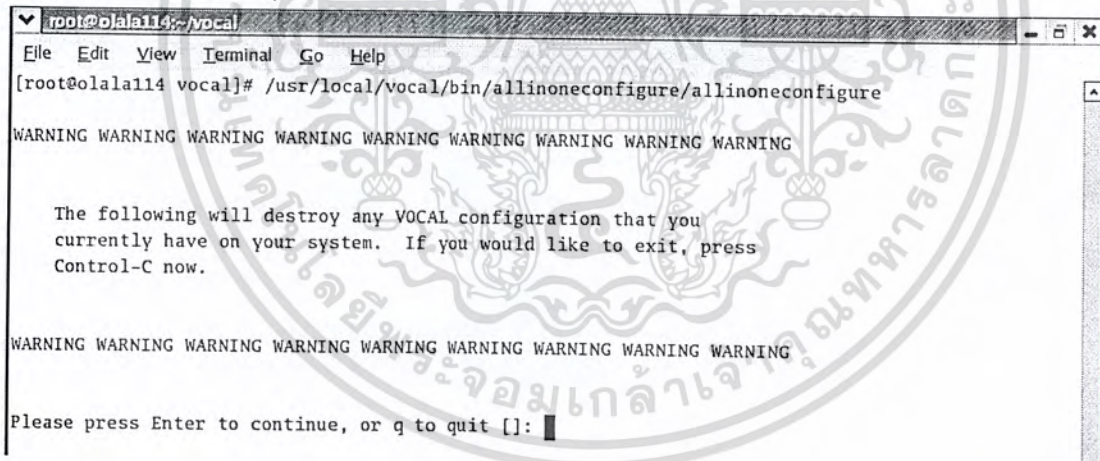
/usr/local/vocal/bin/allinoneconfigure/allinoneconfigure
```

รูปที่ ก-4 ผลลัพธ์ของคำสั่ง `make install`

6. ขั้นตอนต่อไปเป็นการคอนฟิกโปรแกรมโดยการใช้คำสั่ง

```
/usr/local/vocal/bin/allinoneconfigure/allinoneconfigure/
```

เมื่อพิมพ์คำสั่งนี้จะปรากฏหน้าจอดังกล่าว



```
root@olala114:~/vocal
File Edit View Terminal Go Help
[root@olala114 vocal]# /usr/local/vocal/bin/allinoneconfigure/allinoneconfigure

WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING

The following will destroy any VOCAL configuration that you
currently have on your system. If you would like to exit, press
Control-C now.

WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING WARNING

Please press Enter to continue, or q to quit []:
```

รูปที่ ก-5 ผลลัพธ์ของคำสั่ง `/usr/local/vocal/bin/allinoneconfigure/allinoneconfigure/`

7. กด Enter เพื่อเริ่มทำการคอนฟิกโปรแกรม

8. จากนั้นจะมีคำถามขึ้นมา ให้ใส่ค่าดีฟอลต์ของการคอนฟิกเข้าไป หรือทำการกด Enter (ค่า ดีฟอลต์ คือค่าที่อยู่ในวงเล็บ) เมื่อทำการตอบคำถามครบทุกคำถาม จะได้นหน้าจอดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

root@olala114:~/vocal
File Edit View Terminal Go Help

*****

Configuration:
Host IP Address:          161.246.6.114
Remote Contact Address:  161.246.6.114
Multicast Heartbeat IP Address: 244.0.0.100
Multicast Heartbeat Port: 0
Log Level:               LOG_NOTICE
User to run as:          nobody
HTML directory:          /usr/local/vocal/html
CGI directory:           /usr/local/vocal/cgi-bin
Apache will run as:      apache
Openssl:                 /usr/bin/openssl
Add alias to:            /etc/httpd/conf/httpd.conf
Add CGI to:              /etc/httpd/conf/httpd.conf

*****

Continue []: █

```

รูปที่ ก-6 ผลลัพธ์เมื่อทำการคอนฟิกเสร็จ

9. และจะมีคำถามว่าต้องการทำงานต่อหรือไม่ ให้พิมพ์ yes แล้วจะขึ้นหน้าจอบอกถึงข้อมูลที่สำคัญดังรูป

```

root@olala114:~/vocal
File Edit View Terminal Go Help

*****
IMPORTANT INFORMATION
*****

Your VOCAL system is just about ready to run. Please be aware of
the following information:

* You must first restart your HTTP server, as its configuration has
  changed. If you do not, you will not see the website below.

* To configure your system, go to

    http://olala114.ce.kmitl.ac.th/vocal/

And select one of the links there. Here is your information to
log into the Web Based Provisioning system:

User Name: vocal
Password: <entered by user>

* If you lose this information, you can reset the password using

    allinoneconfigure -r

[root@olala114 vocal]# █

```

รูปที่ ก-7 IMPORTANT INFORMATION

10. ทำการรีสตาร์ท Apache ด้วยคำสั่ง

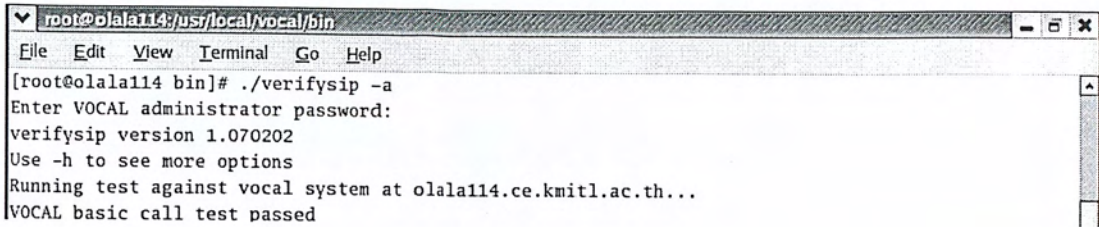
```
/etc/rc.d/init.d/httpd restart
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. จากนั้นทำการตรวจสอบโดยใช้คำสั่ง

```
/etc/local/vocal/bin/verifysip -a
```

ซึ่งจะได้หน้าจอคังรูป



```

root@olala114:/usr/local/vocal/bin
File Edit View Terminal Go Help
[root@olala114 bin]# ./verifysip -a
Enter VOCAL administrator password:
verifysip version 1.070202
Use -h to see more options
Running test against vocal system at olala114.ce.kmitl.ac.th...
VOCAL basic call test passed
  
```

รูปที่ ก-8 ผลลัพธ์ของการใช้คำสั่ง `/etc/local/vocal/bin/verifysip -a`

หมายเหตุ : ถ้าการตรวจสอบไม่ผ่านให้พิมพ์คำสั่งอีกครั้ง

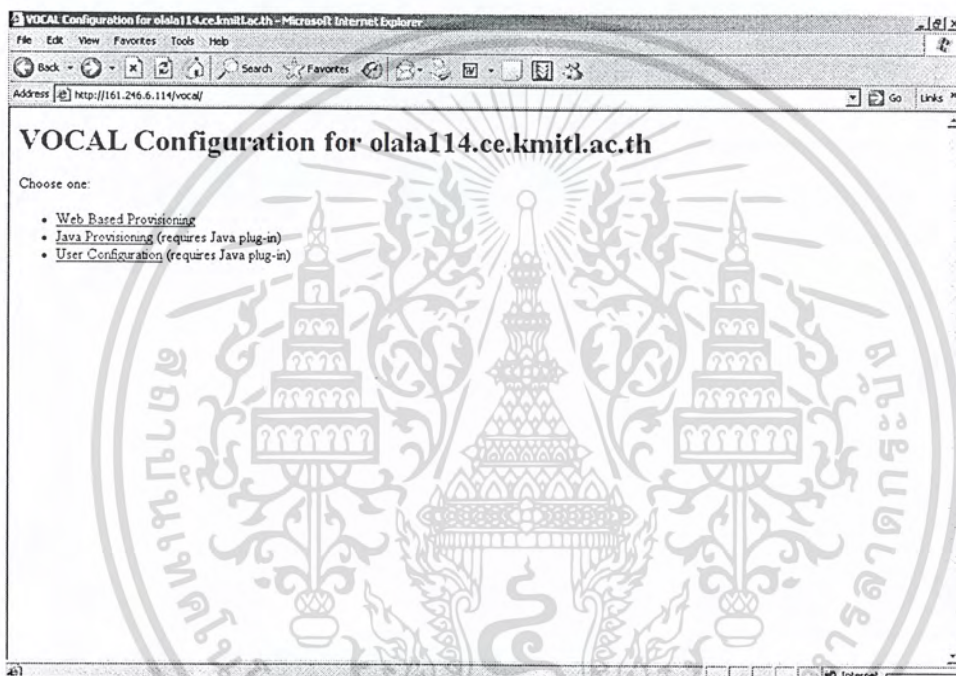


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

การใช้โปรแกรม Vocal

โปรแกรม Vocal สามารถใช้งานได้ผ่านทางบราวเซอร์ โดยการเข้าผ่านทาง url ของเครื่องที่ทำการติดตั้งโปรแกรม เช่น เครื่องที่ติดตั้งมี ip คือ 161.246.6.114 ซึ่งมี hostname คือ olala114.ce.kmitl.ac.th ก็ทำการพิมพ์ url ไปที่ <http://161.246.6.114/vocal> หรือ <http://olala114.ce.kmitl.ac.th> จะได้ผลลัพธ์ดังรูป



รูปที่ ข-1 หน้าแรกของโปรแกรม Vocal

จากนั้นทำการคลิกที่ Web Based Provisioning เพื่อเข้าสู่ระบบการจัดการผู้ใช้

การใช้งาน Web Based Provisioning

Web Based Provisioning นั้นมีฟังก์ชันการทำงานทั้งหมด 7 อย่างได้แก่

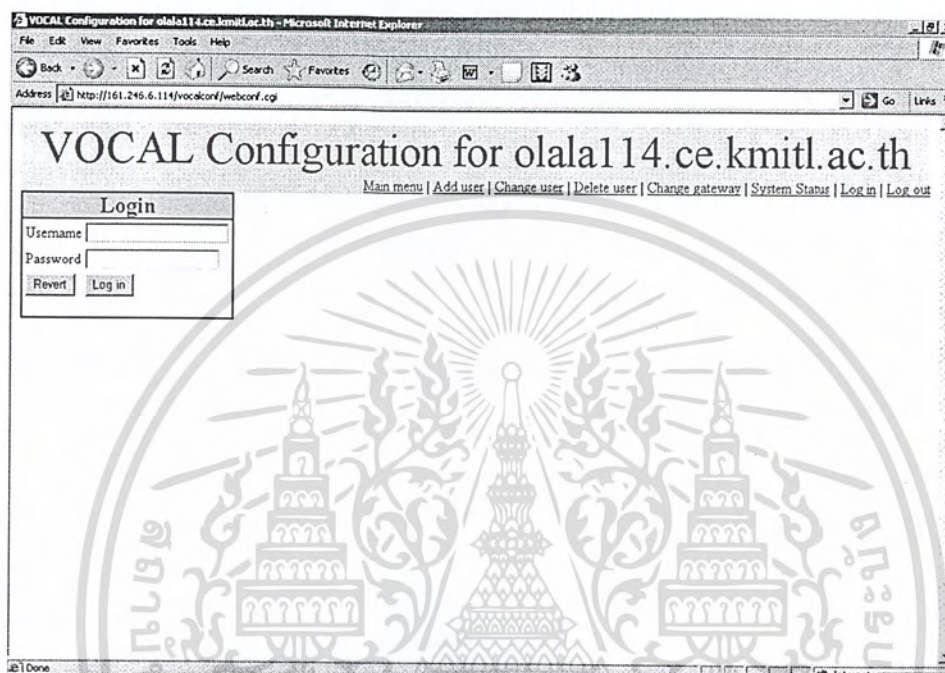
1. Log in ใช้สำหรับการล็อกอินเข้าสู่ระบบ
2. Logout ใช้สำหรับการล็อกเอาต์ออกจากระบบ
3. Add user ใช้สำหรับการเพิ่มจำนวนผู้ใช้
4. Change user ใช้สำหรับการเปลี่ยนแปลงข้อมูลของผู้ใช้
5. Delete user ใช้สำหรับลบผู้ใช้ออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Change gateway ใช้สำหรับเปลี่ยนแปลงเกตเวย์ที่ใช้ในการติดต่อ
7. System status ใช้สำหรับแสดงสถานะของระบบ

1. Login

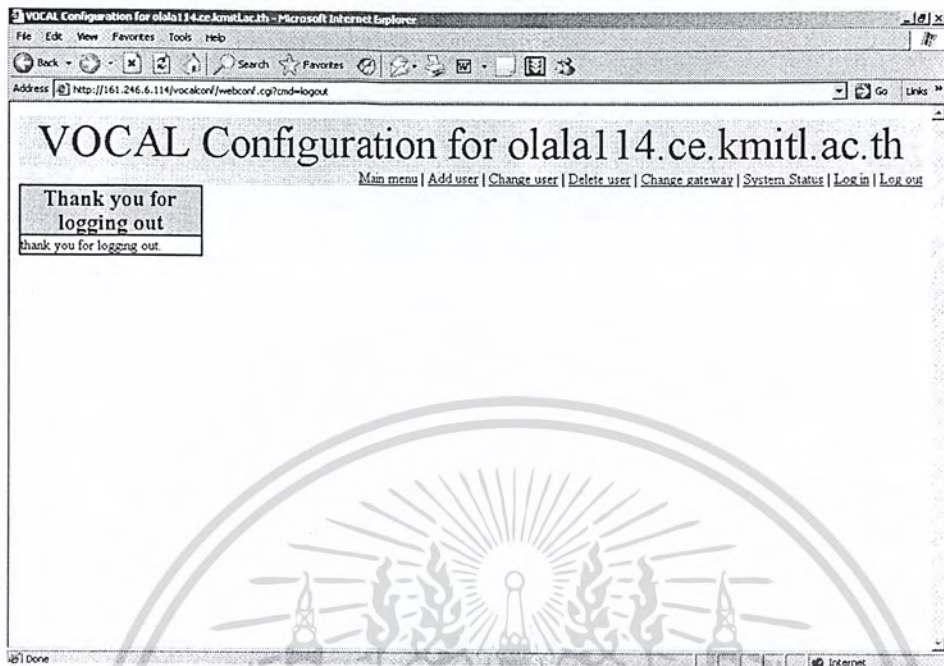
ให้ทำการใส่ username และ password ที่ทำการใส่ไว้เมื่อตอนติดตั้งระบบ



รูปที่ ข-2 หน้าจอการล็อกอินเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

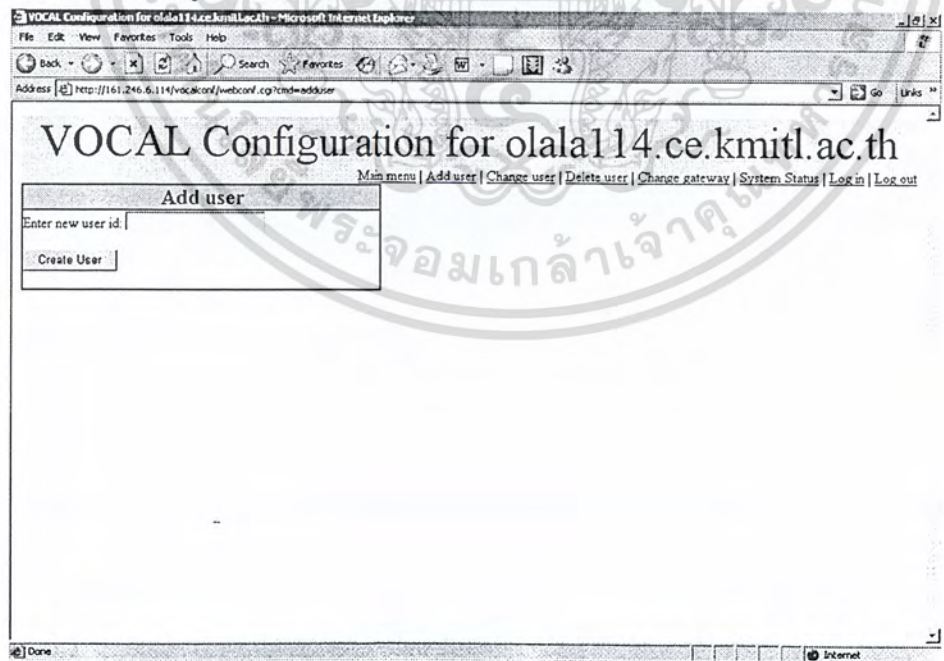
2. Log out



รูปที่ ข-3 ผลลัพธ์เมื่อทำการล็อกเอาต์ออกจากระบบ

3. Add user

ให้ทำการใส่ชื่อผู้ใช้ (เป็นตัวเลขมากกว่า 5 หลักขึ้นไป)



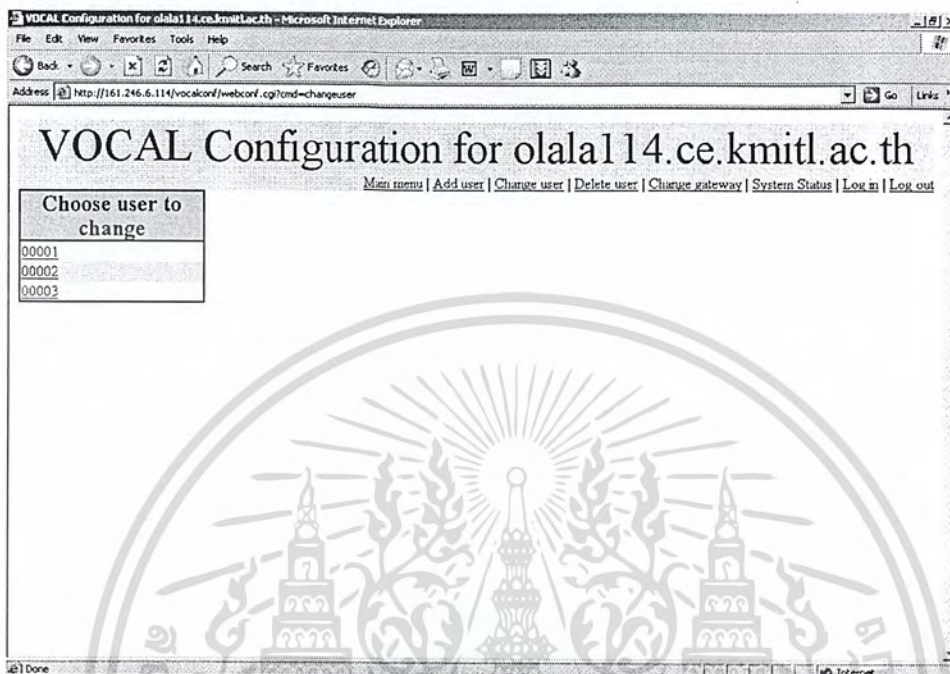
รูปที่ ข-4 หน้าจอการเพิ่มผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

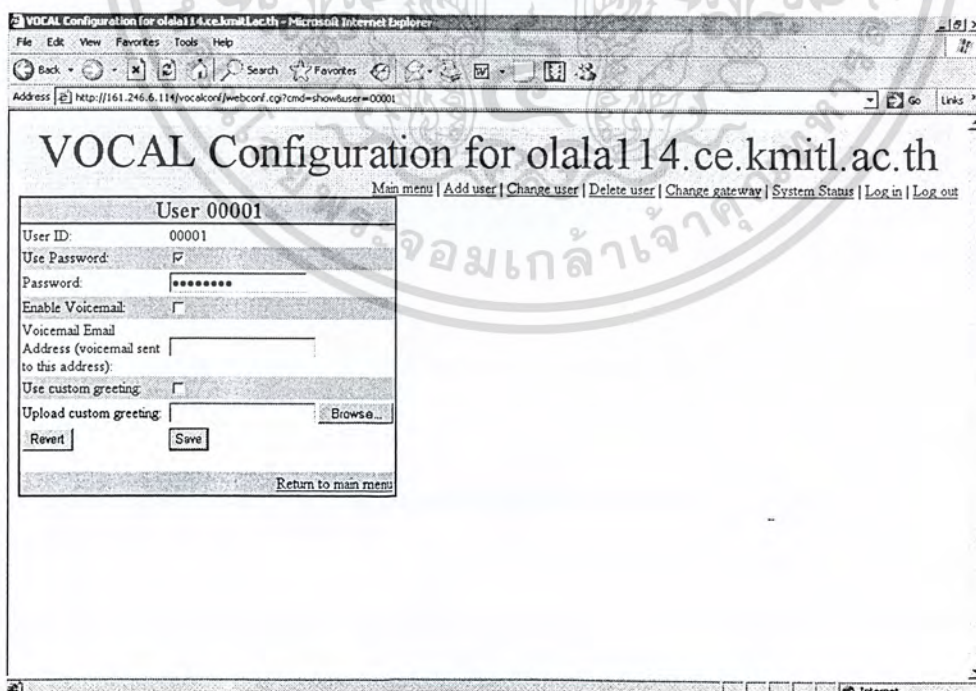
4. Change user

ให้ทำการเลือกผู้ใช้ที่ต้องการจะเปลี่ยนข้อมูล จากนั้นทำการเปลี่ยนข้อมูลตามต้องการ แล้วกดปุ่ม

Save



รูปที่ ข-5 หน้าจอเลือกผู้ใช้

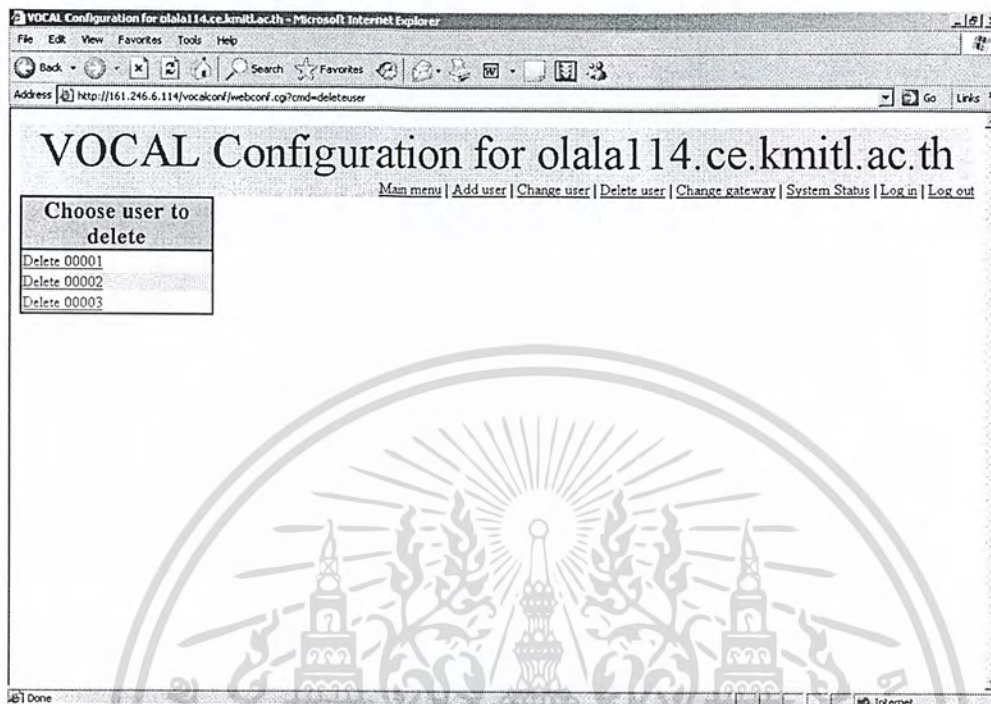


รูปที่ ข-6 หน้าจอการเปลี่ยนแปลงข้อมูลผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Delete user

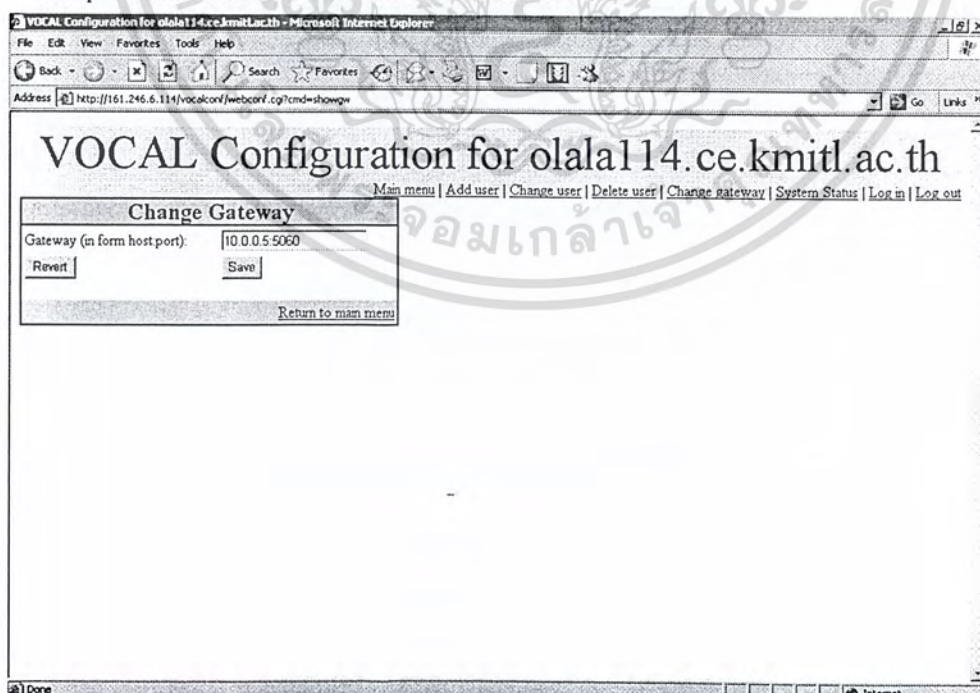
ทำการคลิกชื่อผู้ใช้ที่ต้องการจะลบ



รูปที่ ข-7 หน้าจอการลบผู้ใช้

6. Change gateway

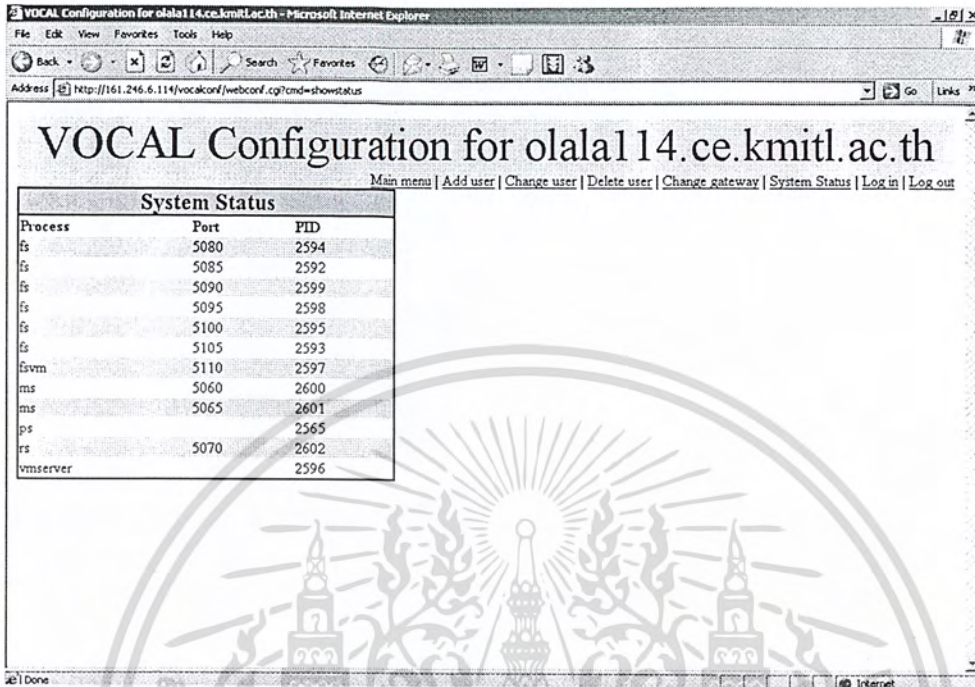
ใส่ ip-ของเกตเวย์ที่ต้องการจะให้เซิร์ฟเวอร์ทำการติดต่อ



รูปที่ ข-8 หน้าจอการเปลี่ยน ip ของเกตเวย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. System Status



VOCAL Configuration for olala114.ce.kmitl.ac.th

Main menu | Add user | Change user | Delete user | Change gateway | System Status | Log in | Log out

Process	Port	PID
fs	5080	2594
fs	5085	2592
fs	5090	2599
fs	5095	2598
fs	5100	2595
fs	5105	2593
fsvm	5110	2597
ms	5060	2600
ms	5065	2601
ps		2565
rs	5070	2602
vmserver		2596

รูปที่ ข-9 หน้าจอแสดงสถานะของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การติดตั้ง JSR 180 RI package

Installing the JSR180 RI package and running the emulator

1. ทำการ Unzip JSR180 RI เข้าไปในไดเรกทอรีที่เราต้องการ
2. การทำ Unzip นั้นจะสร้าง subdirectory, sipa1_0_1-ri-bin ซึ่งประกอบไปด้วยรายการของ subdirectory ใน README-sipa1.0.1-ri-bin.html
3. เอกสาร JSR180 RI นี้อ้างอิงถึง full path name ของไดเรกทอรีซึ่งครอบคลุมการ installation ของ JSR 180 RI เอาไว้ เช่นเดียวกับ <SIPA_ROOT>
4. การทำการ installation ไม่ต้องการ environment หรือ การทำ path setting แต่มีสิ่งหนึ่งที่ต้องตรวจสอบ MISP_HOME environment variable ซึ่ง variable นั้น บางครั้งจะถูกใช้โดย Wireless Toolkit ซึ่ง MIDP_HOME ควรจะไม่ทำการ setup หรือ set เป็นค่า <SIPA_ROOT>\midp
5. ตอนนี้เราสามารถรัน emulator ใน path <SIPA_ROOT>\midp\bin โดยทำการพิมพ์

<SIPA_ROOT>\midp\bin\sipa-midp.exe

สำหรับการข้อมูลบน command line parameter ให้พิมพ์

<SIPA_ROOT>\bin\sipa-amidp.exe -help

Compiling MIDlets

ก่อนที่จะสร้าง MIDlets ของเราเอง ควรรู้ว่า JSR180 RI emulator นั้นสนับสนุน J2ME APIs:

- MIDP2.0
- CLDC1.1
- JSR180

ส่วน software ที่ต้องการสำหรับการคอมไพล์ MIDlets คือ

- Java 2 SDK, Standard Edition (J2SE SDK), version 1.4.2 ขึ้นไป
- MIDP2.0 libraries

ก่อนที่จะทำการคอมไพล์เราควรมี Java 2 SDK และ MIDP2.0 FCS ติดตั้งไว้ เพื่อที่จะคอมไพล์ MIDlet SIP API เราควรที่จะใช้ javac โดยการให้ -d กับ javac ทำให้ file .class ทั้งหมดนั้นถูกเก็บไว้ใน temporary directory ทำให้การสร้างที่สำหรับการ build นั้นง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น การสร้าง output directory, <SIPA_ROOT>tmpclasses, และการคอมไพล์ MIDlets ตัวอย่าง ไปในไดเรกทอรีนั้น ควรจะพิมพ์คำสั่งตามนี้

```
c:\>cd <SIPA_ROOT>
c:\><SIPA_ROOT> mkdir tmpclasses
c:\><SIPA_ROOT> javac -classpath c:\midp2.0fcs\classes;\lib\sipa1_0_1.jar -d
tmpclasses examples/src/com/nokia/phone/ri/sip/examples/*.java
```

Preverifying a MIDlet

Preverifier อยู่ในไดเรกทอรี <SIPA_ROOT>\midp\bin โดยชื่อที่ใช้ในการ execute คือ preverify แสดงตัวอย่างของคำสั่ง preverify ซึ่งถูกใช้ในการตรวจสอบ file .class ทั้งหมดในไดเรกทอรี tmpclasses และ subdirectory ของมัน

```
c:\><SIPA_ROOT>\midp\bin\preverify -classpath
c:\midp2.0fcs\classes;\lib\sipa1_0_1.jar;tmpclasses tmpclasses
```

โดย default แล้วคำสั่งจะเขียน preverified classes ไว้ในไดเรกทอรีชื่อว่า output

Running a preverified MIDlet

ตอนนี้เราสามารถที่จะรัน preverified MIDlets ได้โดยตรงจาก command line โดยจะแสดงตัวอย่างการใช้ ReceiveMessage MIDlet

```
c:\><SIPA_ROOT>\midp\bin\sipa-midp -classpath output
com.nokia.phone.ri.sip.examples.ReceiveMessage
```

Packaging a MIDlet

ในการสร้างกลุ่มของ MIDlet นั้น (.jad และ .jar) ควรจะพิจารณาเอกสารใน MIDP2.0 FCS

การใช้ Wireless toolkit

ขณะนี้ SUN Wireless Toolkit ไม่สนับสนุน JSR180 API มันยังสามารถใช้การ compile และ package MIDlets utilizing SIP API ได้

RI emulator ไม่ได้สนับสนุน Wireless Toolkit และ MIDlets การใช้ JSR180 API ไม่สามารถที่จะรันภายใน Wireless Toolkit ได้

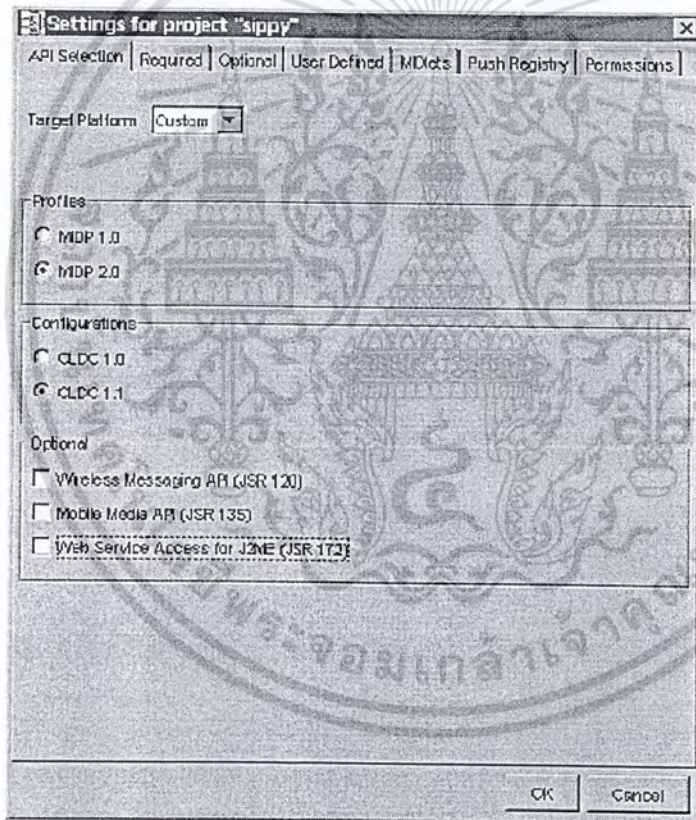
ตามขั้นตอนของ SUN ที่จะทำให้ Wireless Toolkit สามารถที่ดัดแปลงให้สนับสนุน การ compile ของ MIDlets โดยการใช้ JSR180 โดย Wireless Toolkit นั้นจะถูกสมมติให้ถูกติดตั้งลงในไดเรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทอริ C:\WIK21 ซึ่ง key ต่างๆจะถูกรวมไปใน JSR180 API classes ใน WTK ไลบรารี midpapi20.jar โดยสามารถทำได้หลายวิธีเช่น

1. เปิด c:\WTK21\lib\midpapi20.jar ใน Winzip
2. เปิด c:\<SIPA_ROOT>\lib\sipa1_0_1.jar ในอีก Winzip
3. ทำการลาก classes จาก sipa1_0_1.jar ไป midpapi20.jar
4. ทำการปิดทั้ง 2 หน้าต่าง

ก่อนที่จะทำการคอมไพล์ MIDlets ซึ่งสนับสนุน FTWI APIs ใน WTK เพราะฉะนั้นให้แน่ใจว่าไม่มีการใช้ API ที่ไม่ได้รับการสนับสนุน ซึ่งสามารถที่จะดัดแปลงแก้ไขโดย Project ->Settings...->API Selection ดังในรูปซึ่งสามารถใช้ได้เฉพาะ MIDP2.0 และ CLDC1.1 (หรือ 1.0) APIs จาก WTK



รูปที่ ก-1 แสดงการเลือก api ที่ได้รับการสนับสนุน

ตอนนี้เราสามารถที่จะใช้ อุปกรณ์ WTK 'Build' และ 'Project->Package->create package' ได้ ซึ่งการอนุญาต javax.microedition.io.Connector.sip ยังคงต้องถูกเพิ่มเข้าไปในไฟล์ JAD

แพ็คเกจ MIDlet สามารถจะถูกรันโดยตรงใน JSR180 RI emulator จาก ไคเรททอริ project/bin ตามคำสั่งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

c:\<SIPA_ROOT>\midp\bin\sipa-midp -classpath c:\WTK21\apps\my_project\bin
my_project.jar MyMidlet



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

การติดตั้ง J2ME Wireless Toolkit 2.2

ดาวน์โหลดและติดตั้ง J2SDK

สำหรับ J2SDK สามารถดาวน์โหลดได้จากเว็บไซต์ของ sun คือ <http://java.sun.com/j2se> โดย J2SDK ที่จะนำมาใช้พัฒนา MIDlet ต้องเป็นเวอร์ชัน 1.3 หรือสูงกว่า ซึ่งได้ดาวน์โหลด J2SDK เวอร์ชัน 1.4.0_01 ที่มีขนาด 37 เมกะไบต์มาติดตั้ง

หลังจากติดตั้ง J2SDK แล้ว ภายในไดเรกทอรีของ J2SDK จะมีไดเรกทอรีต่างๆ เช่น /bin, /demo, /lib ดังตัวอย่างในภาพที่ ง-1 ซึ่งได้ติดตั้ง J2SDK ไว้ที่ C:\jdk1.4.0_01

```

C:\WINDOWS\system32\cmd.exe
C:\jdk1.5.0_01>dir
Volume in drive C has no label.
Volume Serial Number is 100C-399F

Directory of C:\jdk1.5.0_01

03/21/2005  05:37 PM    <DIR>          .
03/21/2005  05:37 PM    <DIR>          ..
03/21/2005  05:37 PM    <DIR>          bin
12/06/2004  11:05 PM             2,539  COPYRIGHT
03/21/2005  05:37 PM    <DIR>          demo
03/21/2005  05:36 PM    <DIR>          include
03/21/2005  05:37 PM    <DIR>          jre
03/21/2005  05:37 PM    <DIR>          lib
12/06/2004  09:14 PM             15,848  LICENSE
03/21/2005  05:36 PM             21,223  LICENSE.rtf
12/06/2004  09:14 PM             19,367  README.html
03/21/2005  05:36 PM    <DIR>          sample
12/06/2004  09:14 PM            18,081,134  src.zip
12/06/2004  09:14 PM             69,740  THIRDPARTYLICENSEREADME.txt
        6 File(s)          18,209,851 bytes
        8 Dir(s)         4,573,818,880 bytes free

C:\jdk1.5.0_01>

```

รูปที่ ง-1 ไดเรกทอรีของ J2SDK

กำหนด PATH และ CLASSPATH

ไดเรกทอรี /bin ของ J2SDK มีไว้สำหรับเก็บตัวแปลภาษาหรือคอมไพเลอร์ของ java และเก็บโปรแกรมอื่นอีกจำนวนหนึ่ง เราจะต้องนำพาธ หรือชื่อไดเรกทอรี /bin ของ J2SDK (C:\jdk1.4.0_01\bin) ไปเขียนเพิ่มให้แก่ตัวแปล PATH เพื่อให้สามารถเรียกรันไฟล์ต่างๆใน /bin จากไดเรกทอรีไหนก็ได้ และจะต้องนำพาธหรือชื่อไดเรกทอรี /lib ของ J2SDK (C:\jdk1.4.0_01\lib) ไปกำหนดให้แก่ตัวแปล CLASSPATH ด้วย ซึ่งทำได้โดยการเปิดไฟล์ autoexec.bat ที่อยู่ใน C:\ ด้วยโปรแกรม Notepad แล้วเขียนคำสั่ง SET PATH และ SETCLASSPATH เพิ่มต่อท้าย ดังรูป ง-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AUTOEXEC - Notepad
File Edit Format View Help
SET PATH=%PATH%;c:\jdk1.5.0_01\bin
SET CLASSPATH=.;c:\jdk1.5.0_01\lib

```

รูปที่ ๓-2 กำหนด PATH และ CLASSPATH ในไฟล์ autoexec.bat

เมื่อเพิ่มค่าตัวแปร PATH และ CLASSPATH ลงในไฟล์ autoexec.bat เสร็จแล้ว ให้ออกจากการทำงานของระบบ Windows ไปที่ DOS Prompt แล้วรันไฟล์ autoexec.bat จากนั้นใช้คำสั่ง set Path ว่ามีข้อมูลที่เรามาเพิ่มเข้าไปหรือยัง ซึ่งควรได้ผลออกมาดังภาพที่ ๓-3

```

C:\WINDOWS\system32\cmd.exe
3 File(s)          70 bytes
17 Dir(s)         4,573,732,864 bytes free

C:\>autoexec.bat

C:\>SET PATH=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\PROGRAM~1\SecureFX;c:\j2sdk1.4.1_02\bin;C:\Program Files\Microsoft Visual Studio\Common\Tools\WinNI;C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin;C:\Program Files\Microsoft Visual Studio\Common\Tools;C:\Program Files\Microsoft Visual Studio\VC98\bin;c:\jdk1.5.0_01\bin

C:\>SET CLASSPATH=.;c:\jdk1.5.0_01\lib

C:\>set path
Path=C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\PROGRAM~1\SecureFX;c:\j2sdk1.4.1_02\bin;C:\Program Files\Microsoft Visual Studio\Common\Tools\WinNI;C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin;C:\Program Files\Microsoft Visual Studio\Common\Tools;C:\Program Files\Microsoft Visual Studio\VC98\bin;c:\jdk1.5.0_01\bin
PATHEXT=.COM;.EXE;.BAT;.CMD;.UBS;.UBE;.JS;.JSE;.WSF;.WSH

C:\>set classpath
CLASSPATH=.;c:\jdk1.5.0_01\lib

C:\>

```

รูปที่ ๓-3 รูปการแสดงผลของการ SET PATH และ CLASSPATH

หากค่าของ PATH ถูกต้องแล้วเมื่อเรียกตรวจสอบเวอร์ชันของ java ด้วยการป้อนคำสั่ง java -version ผ่านคอมมานไลน์จะได้ผลดังรูปที่ 2-7

```

C:\>java -version
java version "1.5.0_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_01-b08)
Java HotSpot(TM) Client VM (build 1.5.0_01-b08, mixed mode, sharing)

C:\>

```

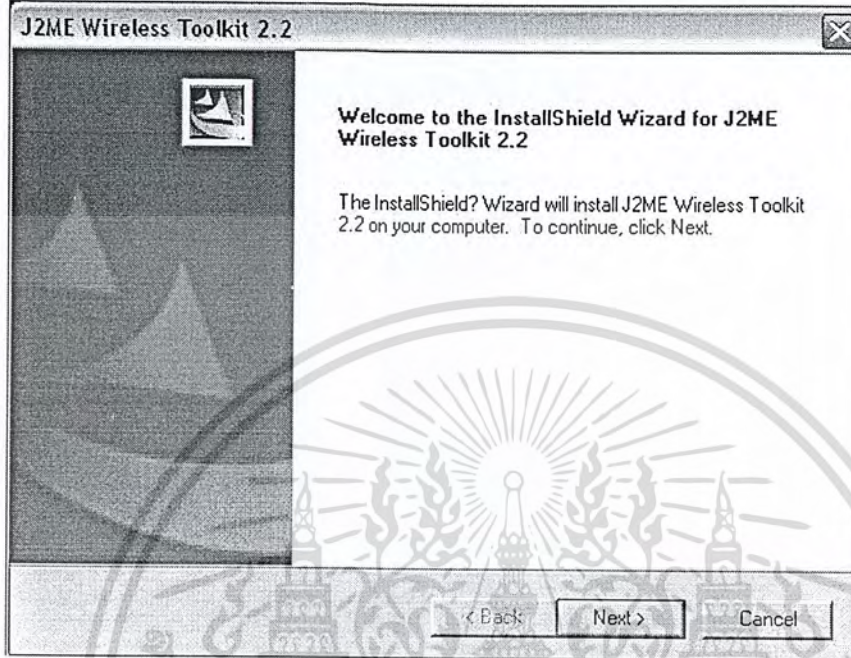
รูปที่ ๓-4 แสดงการตรวจ version ของ java

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

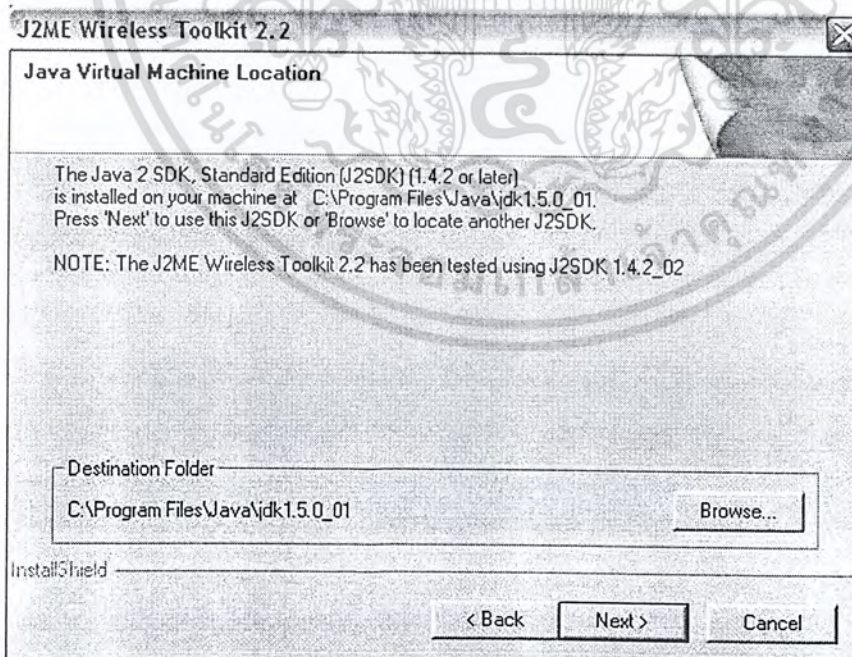
การติดตั้ง J2ME Wireless Toolkit 2.2

เมื่อเริ่มติดตั้งโปรแกรมจะถามถึงโฟลเดอร์ของ J2SE เพื่อหา JVM หากมี J2SE อยู่แล้วตัวติดตั้งจะชี้ไปยังโฟลเดอร์นั้นให้เราโดยอัตโนมัติ สำหรับไฟล์ต่างๆ ของ WTK จะติดตั้งอยู่ที่โฟลเดอร์

C:\\WTK22



รูปที่ 3-5 หน้าแรกของการติดตั้ง

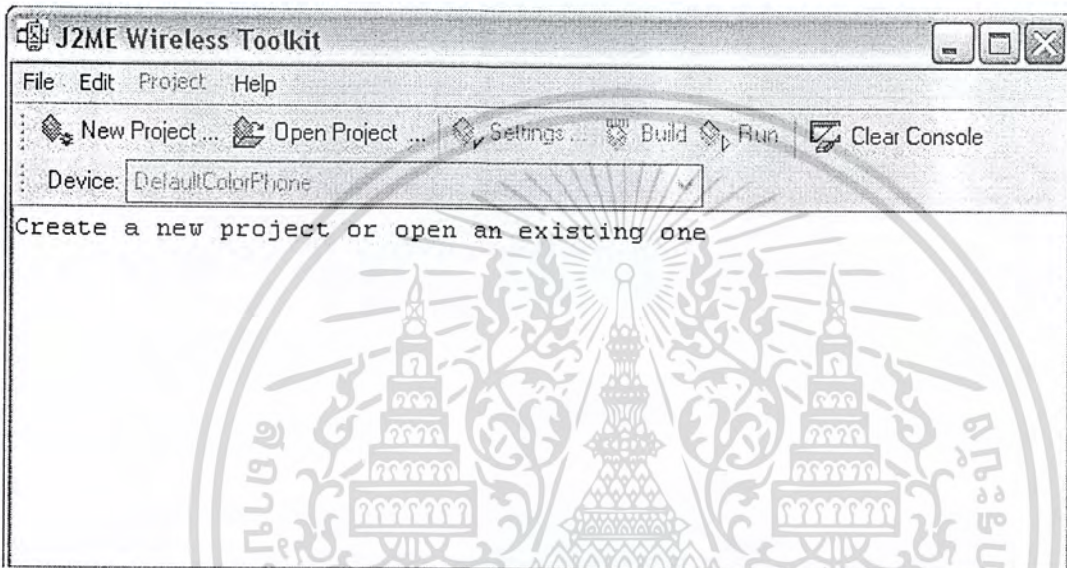


รูปที่ 3-6 ตัวติดตั้ง WTK จะหา Java Virtual Machine ให้กด Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน J2ME Wireless Toolkit 2.2

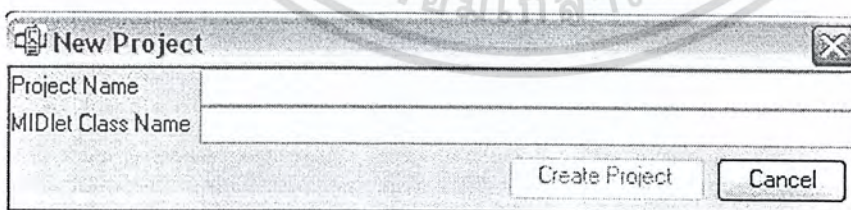
WTK เป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันของ J2ME (MIDlet) โดยมีรูปแบบการพัฒนาเป็นโปรเจกต์ ซึ่งภายในหนึ่งโปรเจกต์จะประกอบไปด้วยขั้นตอนการสร้างโปรเจกต์ การคอมไพล์ การทดสอบการทำงานและการทำแพ็คเกจ แต่สิ่งที่ WTK ไม่มีคือ เอดิเตอร์เพื่อเขียนโค้ดจาวา การเขียนโค้ดจึงต้องไปทำในเอดิเตอร์ตัวอื่นแล้วนำมาใส่ไว้ในไฟล์เตอร์ src ของโปรเจกต์เอง ขั้นตอนทั้งหมดจะผ่านโปรแกรมที่ชื่อ KToolBar (Programs > J2ME Wireless Toolkit 2.2 > KToolBar) เมื่อเรียกโปรแกรมขึ้นมาแล้วจะปรากฏจอร์นดังรูป



รูปที่ ๓-7 หน้าจอหลักของโปรแกรม KToolBar

สร้างโปรเจกต์ใหม่

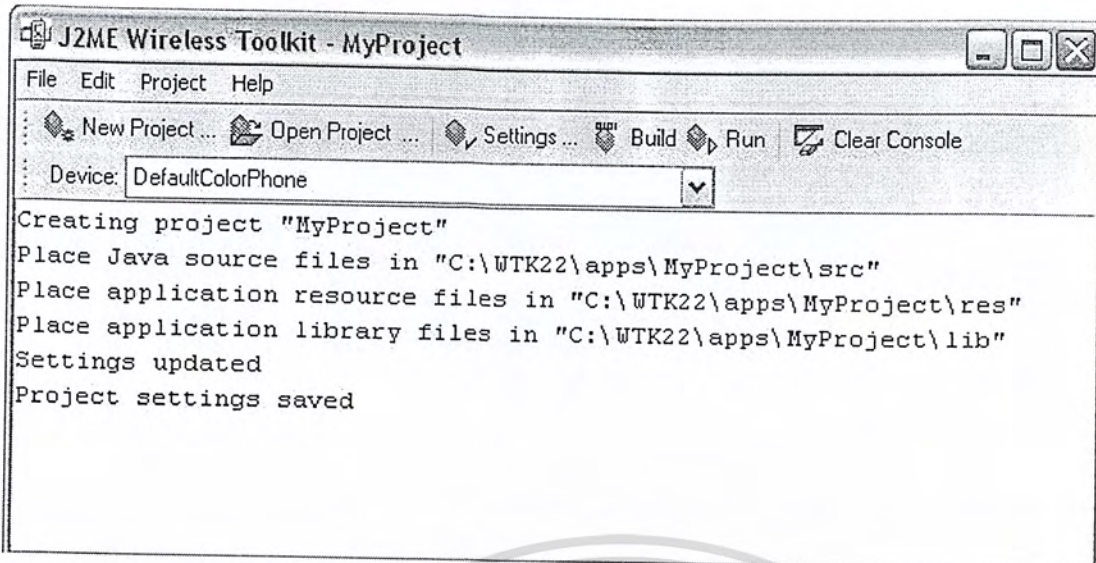
1. เลือกเมนู File > New Project หรือคลิกปุ่ม New Project บนทูลบาร์ จะปรากฏหน้าจอจอร์นดังรูป



รูปที่ ๓-8 สร้างโปรเจกต์ใหม่

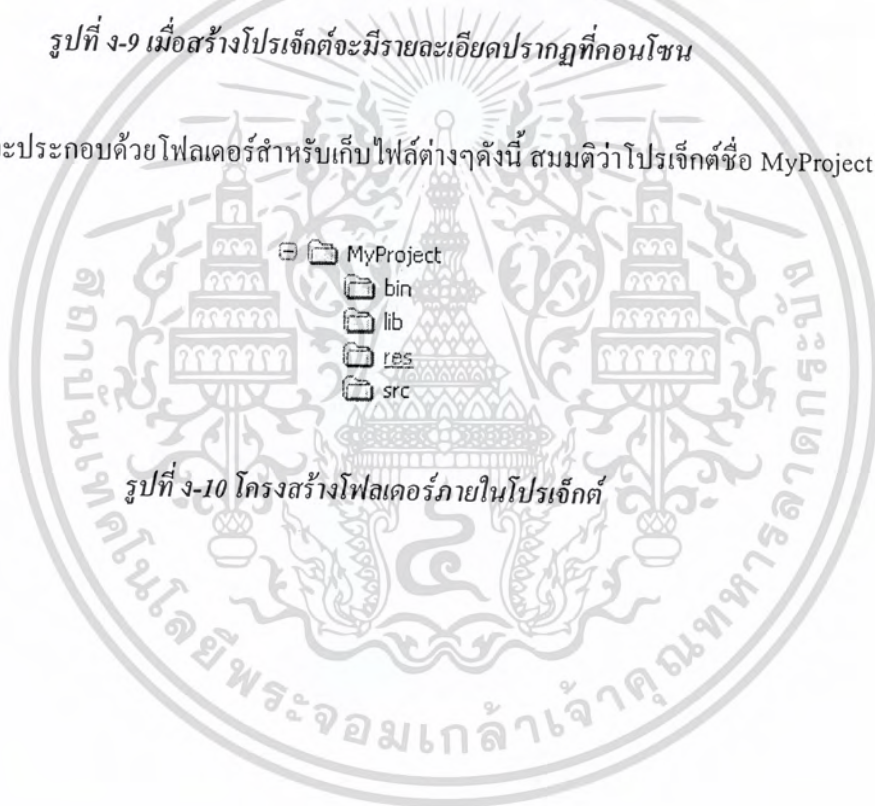
2. ในช่อง Project Name ให้พิมพ์ชื่อของโปรเจกต์เช่น myProject และใส่ชื่อคลาส MIDlet ที่ช่อง MIDlet Class Name เช่น myTest
3. กดปุ่ม Create Project จะปรากฏข้อความแสดงที่คอนโซลดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ๙-๙ เมื่อสร้างโปรเจกต์จะมีรายละเอียดปรากฏที่คอนโซล

ภายในโปรเจกต์จะประกอบด้วยโฟลเดอร์สำหรับเก็บไฟล์ต่างๆดังนี้ สมมติว่าโปรเจกต์ชื่อ MyProject



รูปที่ ๙-๑๐ โครงสร้างโฟลเดอร์ภายในโปรเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

Video/Audio Codec

ในโครงการนี้ได้ใช้ Microsoft's streaming media architecture เป็นสถาปัตยกรรมที่ใช้ในการส่งข้อมูลภาพวิดีโอและเสียง ซึ่งมีรายละเอียดของ Codec ประเภทต่าง ๆ ที่สนับสนุน ดังนี้

Video Codec

Codec	คุณสมบัติ
MPEG-4	<ul style="list-style-type: none"> - บีบอัด ได้คุณภาพดีมากโดยใช้อัตราการส่งข้อมูลที่ต่ำ - ใช้อัตราการส่งข้อมูลมากกว่า H.263
Vivo V263 (H.263)	<ul style="list-style-type: none"> - บีบอัด ได้คุณภาพดีโดยใช้อัตราการส่งข้อมูลที่ต่ำ - ไม่เหมาะสมกับการใช้อัตราการส่งข้อมูลที่สูง - คุณภาพอาจจะต่ำกว่า MPEG-4
VDOnet VDOwave	<ul style="list-style-type: none"> - บีบอัด ได้คุณภาพดีโดยใช้อัตราการส่งข้อมูลที่ต่ำ - ไม่เหมาะสมกับการใช้อัตราการส่งข้อมูลที่สูง - คุณภาพอาจจะต่ำกว่า MPEG-4
ClearVideo 1.3	<ul style="list-style-type: none"> - คุณภาพดีกับการเคลื่อนไหวที่ช้าและใช้ได้กับอัตราการส่งข้อมูลที่สูง - ไม่เหมาะสมกับการเคลื่อนไหวที่เร็วและ อัตราการส่งข้อมูลที่ต่ำ
Duck TrueMotion RT	<ul style="list-style-type: none"> - ได้คุณภาพของภาพที่ดีเยี่ยมที่อัตราการส่งข้อมูลที่สูง - เหมาะสมกับการใช้ในอินทราเน็ต - ไม่เหมาะสมกับการใช้งานบนอินเทอร์เน็ต
Indeo 3.2	<ul style="list-style-type: none"> - ใช้อัตราการส่งข้อมูลที่สูง - คุณภาพของภาพต่ำ
Indeo Video Interactive 4.1	<ul style="list-style-type: none"> - ใช้อัตราการส่งข้อมูลที่สูง - คุณภาพของภาพสูง(ใกล้เคียง MPEG)

ตารางที่ จ-1 แสดงคุณสมบัติของ Codec ภาพแต่ละชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Audio Codec

Codec	อัตราการส่งข้อมูล	คุณสมบัติ
MPEG layer III audio music	8-128 Kbps	<ul style="list-style-type: none"> - คุณภาพสูงเหมาะสำหรับเสียงของวงดนตรี - ไม่เหมาะกับเสียงพูดกับการส่งข้อมูลในอัตราการส่งที่ต่ำกว่านี้
Voxware MetaSound general	8-48 Kbps	<ul style="list-style-type: none"> - คุณภาพเสียงสูง - มีความยืดหยุ่นของช่วงของอัตราการส่งข้อมูล - ต้องการอัตราการส่งข้อมูลที่สูงกว่า MetaVoice
Voxware MetaVoice voice	2.4 Kbps	<ul style="list-style-type: none"> - เหมาะสมกับเสียงพูดในการอัตราการส่งข้อมูลที่ต่ำมาก ๆ - ไม่เหมาะสมกับเสียงดนตรี - ทำงานกับเสียงพูดได้ดี
Voxware RT29 voice	2.9 Kbps	<ul style="list-style-type: none"> - เหมาะสมกับเสียงพูดในการอัตราการส่งข้อมูลที่ต่ำมาก ๆ - ไม่เหมาะสมกับเสียงดนตรี - ทำงานกับเสียงพูดได้ดี
Lernout & Hauspie CELP voice	4.8 Kbps	<ul style="list-style-type: none"> - เหมาะสมกับเสียงพูดในการอัตราการส่งข้อมูลที่ต่ำ
Vivo Siren general	16 Kbps	<ul style="list-style-type: none"> - คุณภาพเสียงดีเหมาะกับเสียงพูดและเสียงดนตรี

ตารางที่ จ-2 แสดงคุณสมบัติของ *Codec* เสียงแต่ละชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ผศ.ดร.บงการ หอมมาน : วีโอไอพี เทคโนโลยีการสื่อสารไร้พรหมแดน : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)
- [2] สาริตพงษ์ พุทธิประเสริฐ, สิ้นชัย กมลภวิงส์, ลัญจนกร วุฒิสัทธิกุลกิจ : โพรโตคอลมาตรฐานสำหรับอินเทอร์เน็ตเทเลโฟนนี่ : เนคเทค เจอนเนล
- [3] ทรงเกียรติ ภาวดี : เก่ง J2ME ให้ครบสูตร : สำนักพิมพ์ วิตตี้กรุ๊ป จำกัด 2546
- [4] ศิวณัฏฐ์ มาศสุรางค์ : JAVA 2 MICRO EDITION (J2ME) คู่มือสำหรับเริ่มต้นพัฒนาจาวาบนมือถือ : บริษัท เออาร์ อินฟอร์เมชั่น แอนด์ พับลิเคชั่น จำกัด 2546
- [5] BUR GOODE : Voice Over Internet Protocol (VoIP): PROCEEDINGS OF THE IEEE
- [6] Jan Janak : SIP Introduction : Fhg FOKUS
- [7] Alan B. Johnston : SIP Understanding the Session Initiation Protocol Second Edition : ARTECH HOUSE, INC. 2004
- [8] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg : SIP: Session Initiation Protocol : RFC2543 1999
- [9] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler : SIP: Session Initiation Protocol : RFC3261 2002
- [10] Charles Petzold : Programming Windows : Microsoft Press 1998
- [11] Vovida.org : Vocal Installation Guide : Cisco System, Inc 2002
- [12] Ubiquity : Understanding SIP
- [13] "Microsoft Corporation", <http://www.microsoft.com>
- [14] "Real-Time Communications (RTC) Client", <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnrtccclnt/html/usertccInt.asp>
- [15] "The SIP Center - A portal for the commercial development of SIP Session Initiation Protocol", <http://www.sipcenter.com>
- [16] "SIP: Session Initiation Protocol", <http://cs.columbia.edu/sip>
- [17] "Codec Central", <http://www.siggraph.org>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้