การออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสม

Designed Framework of Multimedia Conference Architecture based on SIP

ณัฐดนัย วิทยาศิริกุล

Natdanai Wittayasirikul

เลขหมู่ TK5105.8865 N663 2552 ฉ. 2

Bib Key 322390

11 ก.ย. 2552

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Computer Engineering
Prince of Songkla University
2552

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์	การออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสม
ผู้เขียน	นายณัฐดนัย วิทยาศิริกุล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

.....

(รองศาสตราจารย์ ดร.สินชัย กมลภิวงศ์)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

.....

(รองศาสตราจารย์ทศพร กมลภิวงศ์)

คณะกรรมการสอบ

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.นิษฐิดา เอลซ์)

.....กรรมการ

(รองศาสตราจารย์ ดร.สินชัย กมลภิวงศ์)

.....กรรมการ

(รองศาสตราจารย์ทศพร กมลภิวงศ์)

.....กรรมการ

(ดร.กรรณิการ์ ศิริวงศ์ ณ อยุธยา)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....

(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ การออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสม
ผู้เขียน นายณัฐดนัย วิทยาศิริกุล
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2552

บทคัดย่อ

ในช่วงระยะเวลาไม่กี่ที่ผ่านมา จำนวนผู้ใช้งานในเครือข่ายอินเตอร์เน็ต ได้เพิ่มขึ้นอย่างรวดเร็วมาก ซึ่งมีผลทำให้การใช้งานในเครือข่ายเป็นไปอย่างหลากหลาย ซึ่งระบบการติดต่อแบบเทเลโฟนีซึ่งทำงานโดยใช้มาตรฐาน SIP เป็นระบบหนึ่งที่ได้รับการยอมรับอย่างแพร่หลายจากผู้ใช้ต่างและองค์กรต่างๆ แต่ยังพบการติดต่อที่ใช้ส่วนใหญ่ยังคงเป็นแบบจุดต่อจุด

ดังนั้นวิทยานิพนธ์นี้ ได้นำเสนอรูปแบบสถาปัตยกรรมการประชุมที่ทำงานโดยใช้มาตรฐาน SIP ซึ่งบางส่วนได้มีการนำเสนอจากหลายๆกลุ่มวิจัยเช่น IETF เป็นต้น เพื่อนำไปใช้ในการพัฒนา แต่อย่างไรก็ตาม จะพบว่ายังมีหลายๆส่วนที่ยังไม่ได้ระบุการทำงานอย่างชัดเจน เช่นรูปแบบของสัญญาณในการติดต่อ เป็นต้น ดังนั้นงานวิจัยจึงได้นำเสนอสถาปัตยกรรมกรประชุมแบบสื่อประสมที่มีการเพิ่มความสามารถต่างๆ และได้ทำงานระบุรายละเอียดต่างๆเข้าไปอย่างชัดเจนเช่น สัญญาณในการติดต่อ พารามิเตอร์ต่างๆที่จำเป็น และได้นำเสนอการนำระบบไปใช้ในการงานใช้จริง และเชื่อได้ว่าสถาปัตกรรมการประชุมที่ได้นำเสนอจะเป็นการเพิ่มความประสิทธิภาพของมาตรฐานได้มากยิ่งขึ้น

คำสำคัญ: การประชุมสื่อประสม, การติดต่อสื่อสาร, เทเลโฟนี

Thesis Title Designed Framework of Multimedia Conference Architecture based on SIP

Author Mr. Natdanai Wittayasirikul

Major Program Computer Engineering

Academic Year 2009

Abstract

The number of users and servers on the Internet has increased dramatically over past few years leading to increment of the variety of applications. SIP-based Internet Telephony or VoIP is an increasingly popular communication system that is used by various users and organizations. But it's still base on point to point communication.

So this thesis present some extended conference scenarios of SIP based VoIP. A number of recommendations, e.g. drafted by IETF, has been published, as well as some scenarios of deployment. However, some of them may have a number of engineering choices, even not clearly be described, e.g. signal flows or mechanisms. this research describes some enhancements of conference scenarios based on SIP. The system architecture to support such scenarios as well as signal flows are presented. Some flow parameters are given, to show real deployment. We believe that the proposed scenarios will make use of SIP in more effectively use. We are deploying these scenarios for SIP based elearning applications.

Keywords: SIP, VoIP, SIPPING, XCON, Conference

กิตติกรรมประกาศ

ขอขอบพระคุณรองศาสตราจารย์ ดร.สินชัย กมลภิวงศ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ผู้ช่วยศาสตราจารย์ทศพร กมลภิวงศ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ที่ได้ให้คำปรึกษา ชี้แนะแนวทาง และให้ความรู้ในด้านต่าง ๆ รวมถึงการให้การสนับสนุนในเรื่องอุปกรณ์ในการทำวิจัย ตลอดจนช่วยตรวจและแก้ไขวิทยานิพนธ์ให้เป็นไปอย่างสมบูรณ์

ขอขอบพระคุณ ดร.นิษฐิดา เอลซ์ ประธานกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ทศพร กมลภิวงศ์และดร.กรรณิการ์ ศิริวงศ์ ณ อยุธยา กรรมการสอบวิทยานิพนธ์ ที่ได้ให้คำแนะนำในการปรับปรุงวิทยานิพนธ์ให้สมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ และบุคลากรในภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ให้คำปรึกษาและความช่วยเหลือในระหว่างการทำวิทยานิพนธ์

ขอขอบพระคุณคณะวิศวกรรมศาสตร์ที่ได้สนับสนุนทุนการศึกษาในระดับปริญญาโทตลอดระยะเวลาการศึกษา

ขอขอบคุณนักศึกษาปริญญาโทสาขาวิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ได้ให้คำแนะนำและเป็นกำลังใจมาโดยตลอด

ณัฐดนัย วิทยาศิริกุล

สารบัญ

สารบัญ 6
สารบัญรูปภาพ 8
สารบัญตาราง 11
สัญลักษณ์คำย่อและตัวย่อ 12
บทที่ 1 บทนำ 13
1.1. ความสำคัญและที่มาของวิทยานิพนธ์ 13
1.2. วัตถุประสงค์ 14
1.3. ขอบเขตของการวิจัย 14
1.4. ขั้นตอนและวิธีการดำเนินการวิจัย 15
1.5. ประโยชน์ที่คาดว่าจะได้รับ 15
บทที่ 2 ทฤษฎีและหลักการ 16
2.1 SIP (Session Initiation Protocol) 16
2.2 การทำงานของ SIP Server ในระบบ SIP 18
2.3 การให้บริการเสริมของ SIP Server 22
2.4 รูปแบบลักษณะการสร้างการประชุมแบบต่าง ๆ 34
2.5 สรุป 40
บทที่ 3 การตรวจเอกสาร 41
3.1 รูปแบบข้อมูลของกฎข้อบังคับที่ใช้ในการควบคุมการประชุม (Conference Policy). 41
3.2 ความสามารถในการจัดการข้อมูลผู้ใช้งาน (User Management) 49
3.3 The Extensible Markup Language Configuration Access Protocol (XCAP) 57
3.4 สรุป 65
บทที่ 4 การออกแบบระบบการประชุมแบบสื่อประสม 66
4.1 เหตุใดจึงต้องออกแบบระบบการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP 66
4.2 การออกแบบองค์ประกอบเพิ่มเติมในระบบการประชุม 67

4.3 รูปแบบของสัญญาณ SIP ในระบบการประชุม 69
4.4 สรุป 94
บทที่ 5 การพัฒนาโปรแกรมประยุกต์เพื่อการทดสอบ 95
5.1 เหตุผลที่ต้องมีโปรแกรมประยุกต์เพื่อการทดสอบ 95
5.2 เครื่องมือที่ใช้ในการพัฒนา 95
5.3 ขอบเขตของการพัฒนาโปรแกรมประยุกต์ 98
5.4 การออกแบบโปรแกรมประยุกต์ 98
5.5 ตัวอย่างโปรแกรมประยุกต์ 101
5.6 ฟังก์ชันการทำงานของโปรแกรมประยุกต์ต้นแบบ 103
5.7 กระบวนการทดสอบ 103
5.8 ผลที่คาดว่าจะได้รับ 107
5.9 ผลการทดสอบ 107
5.10 สรุป 113
บทที่ 6 สรุปผลการวิจัย และข้อเสนอแนะ 114
6.1 สรุปผลการวิจัย 114
6.2 ข้อเสนอแนะ 115
เอกสารอ้างอิง 117
ประวัติ 119

สารบัญรูปภาพ

รูปที่ 2-1 องค์ประกอบสำคัญในระบบ SIP (SIP Component) 17
รูปที่ 2-2 แสดงการทำงานแบบ Registrar Server 19
รูปที่ 2-3 แสดงตัวอย่างการทำงานของ SIP Proxy Server.....20
รูปที่ 2-4 แสดงตัวอย่างการทำงานของ SIP Redirect Server 21
รูปที่ 2-5 แสดงรูปแบบการให้บริการ Instance Message ของ SIP Server 23
รูปที่ 2-6 แสดงลำดับสัญญาณ SIP ในการให้บริการ Instance Message 24
รูปที่ 2-7 แสดงรูปแบบการให้บริการ Call Forward ของ SIP Server 25
รูปที่ 2-8 แสดงลำดับสัญญาณ SIP ในการให้บริการ Call Forward 26
รูปที่ 2-9 รูปแบบบริการ Presence ของ SIP Server 27
รูปที่ 2-10 รูปแบบบริการ Presence ของ SIP Server เมื่อ SIP Server ส่งสัญญาณ NOTIFY .27
รูปที่ 2-11 รูปแบบบริการ Presence ของ SIP Server เมื่อ SIP Client B ทำการ Register 28
รูปที่ 2-12 แสดงลำดับสัญญาณ SIP ในการให้บริการ Presence 29
รูปที่ 2-13 แสดงรูปแบบการให้บริการ Forking ของ SIP Server 29
รูปที่ 2-14 แสดงลำดับสัญญาณ SIP ในการให้บริการ Forking 31
รูปที่ 2-15 Component ต่างๆในระบบ SIP Conference.....32
รูปที่ 2-16 Centralized Server 35
รูปที่ 2-17 Endpoint Server 36
รูปที่ 2-18 Media Server Component 37
รูปที่ 2-19 Distributed Mixing 38
รูปที่ 2-20 Cascaded Mixers 39
รูปที่ 3-1 Protocol stack of an end system in the SIP Conferencing Architecture.....49
รูปที่ 3-2 ตัวอย่างการใช้งาน SOAP เพื่อทำการควบคุมการประชุม 50
รูปที่ 3-3 โครงสร้างของระบบ XCCP 51
รูปที่ 3-4 XCCP protocol stack 51
รูปที่ 3-5 ผลจากการทดลองใช้ XCCP ในระบบการประชุม 52
รูปที่ 3-6 ผลจากการทดลองใช้ XCCP ในระบบการประชุม 52
รูปที่ 3-7 Distributed conferencing (full-mesh) 53
รูปที่ 3-8 Distributed conferencing (Tree) 54
รูปที่ 3-9 Average signaling delay for invitation 55

รูปที่ 3-10 Average stream transmission delay 56
รูปที่ 3-11 ตัวอย่างเอกสาร XML.....58
รูปที่ 3-12 โครงสร้างการอ้างถึงข้อมูลของโปรโตคอล XCAP 58
รูปที่ 3-13 รูปแบบการกำหนด URI ในโปรโตคอล XCAP 59
รูปที่ 3-14 การ Fetch Document.....60
รูปที่ 3-15 การ Fetch Element 60
รูปที่ 3-16 การ Fetch Attribute 61
รูปที่ 3-17 การ Delete Document 61
รูปที่ 3-18 การ Delete Elment.....62
รูปที่ 3-19 การเพิ่มข้อมูลลงในเอกสาร XML 63
รูปที่ 3-20 การแก้ไขข้อมูลในเอกสาร XML 63
รูปที่ 4-1 Service & Network Architecture 69
รูปที่ 4-2 INVITE: Joining a Conference Using the Conference URI – Dial-In.....71
รูปที่ 4-3 INVITE: Adding a Participant by the Focus – Dial-Out.....72
รูปที่ 4-4 INVITE: Manually Creating a Conference by Dialing In to a Conferencing Application.....73
รูปที่ 4-5 INVITE: Creating a Conference Using Ad-hoc SIP Methods 74
รูปที่ 4-6 REFER: Requesting a Focus to Add a New Resource to a Conference 75
รูปที่ 4-7 REFER with REFER: Requesting a Focus to Refer a Participant 76
รูปที่ 4-8 REFER with BYE: Requesting That the Focus Remove a Participant from a Conference 77
รูปที่ 4-9 Deleting a Conference.....78
รูปที่ 4-10 Discovery of URI Properties Using OPTIONS 79
รูปที่ 4-11 Just Conferencing Scenario 80
รูปที่ 4-12 Just Conference signaling flow 83
รูปที่ 4-13 Initial INVITE request received at the Conference Server.....85
รูปที่ 4-14 XCAP PUT request for specific conference information.....85
รูปที่ 4-15 Group Finding Scenario 86
รูปที่ 4-16 Group Finding (Via LDAP) signaling flow 87

รูปที่ 4-17 An example of the INVITE request F9 that contains MSCML data for specifying the Mixer. 88
รูปที่ 4-18 Group Finding (Via HTTP) signaling flow....89
รูปที่ 4-19 Scheduled Conference Scenario 89
รูปที่ 4-20 Schedule Conferencing signaling flow....92
รูปที่ 4-21 Sidebar Conference Call....93
รูปที่ 4-22 Voting in Conference....94
รูปที่ 5-1 การเชื่อม WeSIP กับ OpenSIPS ในรูปแบบต่าง ๆ 97
รูปที่ 5-2 โครงการระบบการประชุมแบบสื่อประสมที่ได้ทำการพัฒนา 99
รูปที่ 5-3 ตัวอย่างโปรแกรมประยุกต์ส่วนการสร้างการประชุม.... 101
รูปที่ 5-4 ตัวอย่างการติดต่อมายังประธานการประชุมเพื่อเริ่มต้นการประชุม 101
รูปที่ 5-5 ตัวอย่างเพิ่มผู้เข้าร่วมการประชุม.... 102
รูปที่ 5-6 รูปแบบทดสอบ 104
รูปที่ 5-7 รูปแบบสัญญาณที่ใช้ในการทดสอบ 106
รูปที่ 5-8 การใส่ค่าสำหรับเริ่มการประชุม.... 107
รูปที่ 5-9 ลำดับสัญญาณที่เกิดขึ้นในการติดต่อกับ Mixer.... 108
รูปที่ 5-10 ลำดับสัญญาณที่แสดงในระบบ WeSIP.... 108
รูปที่ 5-11 แสดงการเข้าร่วมการประชุมของประธานการประชุม.... 109
รูปที่ 5-12 แสดงลำดับสัญญาณการเข้าร่วมการประชุมของประธานการประชุม.... 109
รูปที่ 5-13 หน้าแสดงส่วนการเชิญผู้เข้าร่วมอื่นเข้าสู่การประชุม.... 110
รูปที่ 5-14 หน้าแสดงส่วนการเข้าสู่ระบบของผู้ร่วมเข้าประชุม.... 110
รูปที่ 5-15 หน้าแสดงส่วนการเชิญเข้าสู่ระบบของผู้ร่วมเข้าประชุม.... 111
รูปที่ 5-16 ลำดับสัญญาณในการเชิญผู้ร่วมการประชุมคนอื่นเข้าร่วมการประชุม 111
รูปที่ 5-17 ลำดับสัญญาณในการสิ้นสุดการประชุม.... 112
รูปที่ 5-18 หน้าจอผู้เข้าร่วมการประชุมเมื่อสิ้นสุดการประชุม.... 112

สารบัญตาราง

ตารางที่ 2-1 ตารางเปรียบเทียบความสามารถของสถาปัตยกรรมการประชุมแบบต่าง ๆ [10]...40
ตารางที่ 3-1 ตารางเปรียบเทียบโปรโตคอลต่าง ๆ 64

สัญลักษณ์คำย่อและตัวย่อ

HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
PSTN	Public Switching Telephony Network
RFC	Request for Comments
RTCP	Real-time Control Protocol
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
SIP	Session Initialtion Protocol
TCP	Transmission Control Protocol
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
XML	Extensible Markup Language
XCAP	The Extensible Markup Language Configuration Access Protocol

บทที่ 1
บทนำ

1.1. ความสำคัญและที่มาของวิทยานิพนธ์

อินเตอร์เน็ตเป็นเครือข่ายที่มีการใช้งานกันอย่างแพร่หลาย บริการบนเครือข่ายอินเตอร์เน็ตนั้นนอกจากบริการทางด้านข้อมูลอย่างเช่น จดหมายอิเล็กทรอนิกส์ เวิร์ลไวด์เว็บและการขนถ่ายไฟล์ ยังมีการให้บริการโดยใช้สื่อประเภทอื่นเช่น เสียง และวีดิทัศน์ เป็นต้น ดังนั้นจะเห็นได้ว่าปัจจุบันมีการพัฒนาโปรแกรมประยุกต์ที่สามารถใช้การสื่อสารหลายประเภทพร้อมกันได้ ซึ่งเรียกว่า การสื่อสารแบบสื่อประสม(Multimedia Communications) เช่น การสื่อสารที่ผู้ใช้สนทนากันโดยสามารถเห็นภาพของคู่สนทนา รวมทั้งอาจจะมีการส่งไฟล์ในระหว่างการสนทนาเป็นต้น โดยจะพบว่าการสื่อสารแบบสื่อประสมในอดีตนั้นเป็นแบบหนึ่งต่อหนึ่งซึ่งหากนำมาเปรียบเทียบกับการใช้งานปัจจุบันแล้วจะพบว่าไม่เพียงพอต่อการใช้งาน ยิ่งเมื่อนำมาใช้งานในเชิงธุรกิจ จะพบว่าการดำเนินธุรกิจในปัจจุบันนี้การติดต่อสื่อสารถือเป็นหัวใจสำคัญอย่างหนึ่งในองค์ประกอบที่จะนำไปสู่ความสำเร็จ ทำให้ระบบการติดต่อสื่อสารที่มีประสิทธิภาพและมีความรวดเร็วเป็นตัวแปรสำคัญอย่างหนึ่ง ในการเพิ่มโอกาสทางธุรกิจและในการติดต่อทางธุรกิจสิ่งหนึ่งที่จะขาดไปเสียมิได้คือ การประชุม แต่การประชุมก็เป็นสิ่งหนึ่งที่ทำให้เราต้องเสียเวลาและพลาดโอกาสต่าง ๆ ที่สำคัญ เนื่องจากต้องมีการเดินทางจากที่ต่าง ๆ เพื่อมาประชุมกัน ณ สถานที่ที่กำหนดที่ใดที่หนึ่ง ซึ่งในการเดินทางนั้นจะต้องเสียเวลา ค่าใช้จ่ายและยังเกิดความเสี่ยงในการเดินทางขึ้นอีกด้วย ดังนั้นเพื่อเป็นการแก้ปัญหาต่าง ๆ ที่เกิดขึ้นและเพื่อเป็นการสร้างโอกาสทางธุรกิจให้เพิ่มมากขึ้น จึงมีความจำเป็นในการนำอุปกรณ์สื่อสารเข้ามาใช้ประโยชน์ระบบการประชุมแบบสื่อประสมเป็น การติดต่อสื่อสารระหว่างบุคคลหรือกลุ่มบุคคลที่อยู่ต่างสถานที่กันได้ทั้งภาพเสียง และข้อมูลในเวลาเดียวกัน ซึ่งมาตรฐานที่มีการพัฒนาขึ้นเพื่อใช้งานในการสื่อสารแบบสื่อประสม(Multimedia Communications) ในปัจจุบันที่พบบ่อย ๆคือ H.323 [1]และมาตรฐาน SIP (Session Initiation Protocol)[2]ซึ่งมาตรฐานทั้งสองมาตรฐานนี้ได้ถูกออกแบบให้สามารถรองรับการทำงานในระบบการประชุมแบบสื่อประสมได้ทั้งคู่ โดยมาตรฐาน H.323 จะมีการทำงานที่เน้นไปในเรื่องของการทำ load balancing, conference management service ในขณะที่มาตรฐาน SIP จะให้ความสนใจในเรื่องความยืดหยุ่นของระบบ สามารถปรับปรุงเพิ่มเติมได้ จะพบว่ามาตรฐาน SIP กลับได้รับความนิยมมากกว่ามาตรฐาน H.323 เนื่องมาจากความง่ายและยืดหยุ่นในการพัฒนา[3][4]

แต่เมื่อมาศึกษาถึงระบบการประชุมแบบสื่อประสมของมาตรฐาน SIP จะพบว่ายังไม่มีสถาปัตยกรรมที่เป็นมาตรฐานแน่นอนและยังขาดในส่วนของมาตรฐานในการควบคุมการประชุม (Conference Control) ที่แน่นอน ซึ่งเป็นส่วนสำคัญในของระบบการประชุมแบบสื่อ

ประสมที่คอยประสานงานและจัดการข้อมูลต่าง ๆที่อยู่ในการประชุมเช่น สิทธิการเข้าถึงข้อมูลต่าง ๆของผู้ติดต่อ ชนิดของข้อมูลสื่อประสมที่ใช้ในการติดต่อ เป็นต้น หากระบบการประชุมขาดมาตรฐานการจัดการในส่วนนี้อาจจะมีผลทำให้การประชุมนั้น ๆมีประสิทธิภาพน้อยลงเป็นอย่างมาก วิทยานิพนธ์นี้จึงมุ่งที่จะออกแบบและพัฒนาสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมบนมาตรฐาน SIP ที่มีความสามารถทำงานได้ถูกต้องตามมาตรฐานมากที่สุด

1.2. วัตถุประสงค์

1. เพื่อศึกษารูปแบบการประชุมแบบสื่อประสมบนโปรโตคอล SIP
2. เพื่อออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมบนโปรโตคอล SIP
3. สถาปัตยกรรมที่ออกแบบจะต้องทำงานร่วมกับระบบการติดต่อสื่อสารเดิมโดยใช้โปรโตคอล SIP และไม่ส่งผลกระทบต่อระบบเดิม
4. ทดสอบการทำงานของระบบการประชุม เพื่อแสดงให้เห็นถึงความสามารถที่ได้เพิ่มเติมจากการออกแบบ และการทำงานร่วมกับระบบการติดต่อสื่อสารโดยใช้ SIP ผ่านเครือข่ายเดิม

1.3. ขอบเขตของการวิจัย

1. ออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมบนโปรโตคอล SIP ที่เหมาะสมกับสภาพการใช้งานในรูปแบบต่าง ๆ
2. พัฒนาระบบต้นแบบการประชุมแบบสื่อประสมบนโปรโตคอล SIP
3. ระบบการประชุมที่พัฒนา จะยังไม่ครอบคลุมในเรื่องของความปลอดภัย และการส่งข้อมูลสื่อประสมประเภทวิดีโอ
4. สัญญาณควบคุมในระบบการประชุมจะใช้ SIP ซึ่งเป็นโปรโตคอลมาตรฐานที่ถูกออกแบบโดยองค์กร IETF (Internet Engineering Task Force)
5. การทดสอบการทำงานของระบบการประชุมจะอยู่ในรูปของการพัฒนาโปรแกรมประยุกต์ขึ้นมาทดสอบ

1.4. ขั้นตอนและวิธีการดำเนินการวิจัย

1. ศึกษาทำความเข้าใจเกี่ยวกับ การออกแบบการทำการประชุมแบบสื่อประสมบนโปรโตคอล SIP ในรูปแบบต่าง ๆ และบทความที่เกี่ยวข้อง
2. ศึกษาทำความเข้าใจเกี่ยวกับ สัญญาณ SIP และโปรโตคอลอื่น ๆที่เกี่ยวข้องกับการทำการประชุมแบบสื่อประสมบนโปรโตคอล SIP รวมทั้งจัดทำโครงร่างวิทยานิพนธ์
3. สร้างออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมบนโปรโตคอล SIP
4. นำสถาปัตยกรรมที่ออกแบบมาทำการพัฒนาและทดสอบ
5. ปรับปรุงและทำสอบความถูกต้องของระบบที่ทำการออกแบบ
6. รวบรวมผลการทดสอบ สรุปผล จัดทำรายงานฉบับสมบูรณ์และคู่มือการใช้งาน

1.5. ประโยชน์ที่คาดว่าจะได้รับ

1. รูปแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมที่เหมาะสมกับสภาพเครือข่ายจะช่วยมีการใช้ทรัพยากรของระบบอย่างคุ้มค่า
2. สถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมที่ได้ทำการออกแบบจะช่วยให้นักพัฒนาสามารถที่จะพัฒนาโปรแกรมประเภทนี้ได้ง่ายยิ่งขึ้น
3. ผลงานในระหว่างที่ทำวิทยานิพนธ์สามารถนำไปตีพิมพ์ลงในบทความทางวิชาการ

บทที่ 2
ทฤษฎีและหลักการ

สำหรับการออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมที่จะกล่าวถึงในวิทยานิพนธ์นี้จะทำงานอยู่บนมาตรฐาน SIP ซึ่งในหัวข้อนี้จะกล่าวถึงหลักการทำงานของมาตรฐาน SIP และรูปแบบของสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมบางส่วนที่ได้มีการออกแบบไว้แล้ว

2.1 SIP (Session Initiation Protocol)

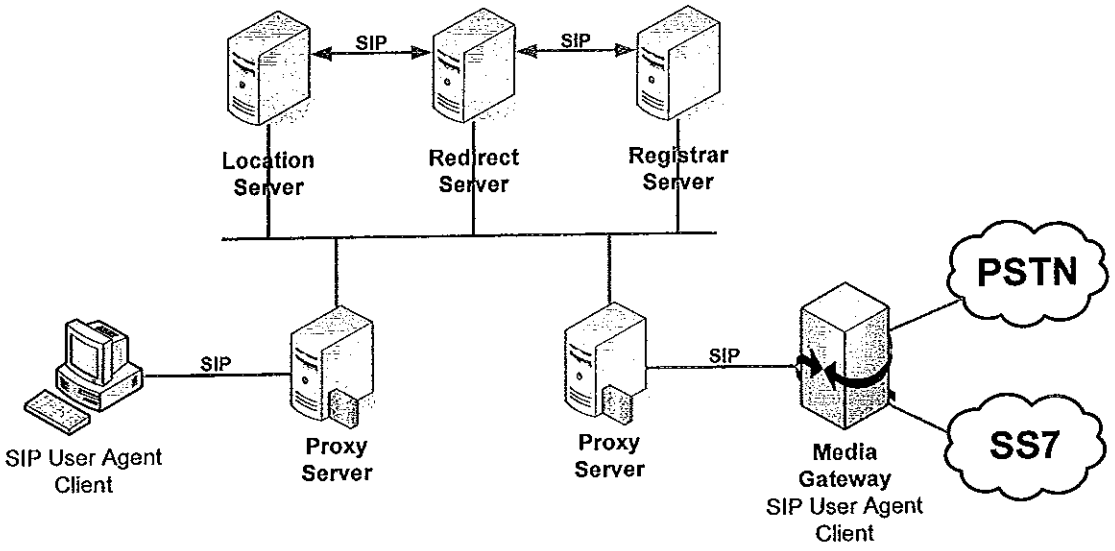
โปรโตคอล SIP เป็นโปรโตคอลระดับชั้นโปรแกรมประยุกต์ ถูกพัฒนาโดย IETF (Internet Engineering Task Force) โดยออกแบบให้ใช้ในการตกลงกันระหว่างคู่สนทนาหรือระหว่างผู้ใช้บริการเพื่อทำการเริ่มต้นการติดต่อ (Establish call), เปลี่ยนแปลงรูปแบบการติดต่อ (Modify) และสิ้นสุดการติดต่อหรือจบการติดต่อ (End call/Terminate call) ดังนั้นโปรโตคอล SIP มีหน้าที่รับผิดชอบในส่วนการส่งสัญญาณควบคุม (Control Signaling) ก่อนและจบการสนทนาเท่านั้น นอกจากนี้การทำงานอย่างเป็นระบบตามมาตรฐาน SIP ทำให้เกิดความยืดหยุ่นและความสะดวกในการใช้งานแก่ผู้ใช้บริการมากยิ่งขึ้น

2.1.1 คุณสมบัติของ SIP

1. เป็นโปรโตคอลระดับชั้นโปรแกรมประยุกต์ ซึ่งอยู่เหนือโปรโตคอลระดับชั้นสื่อสารนำส่งข้อมูล โดยอนุญาตให้สามารถทำการส่งสัญญาณโดยใช้โปรโตคอลระดับชั้นสื่อสารนำส่งข้อมูล ได้ทั้ง TCP (Transmission Control Protocol) และ UDP (User Datagram Protocol)
2. รูปแบบสัญญาณที่นิยามตามมาตรฐาน SIP มีลักษณะเป็นแบบข้อความ (Text-Based) ซึ่งถูกเรียกว่า SIP Message โดยรูปแบบและไวยากรณ์ของสัญญาณมีลักษณะคล้ายกับรูปแบบสัญญาณของโปรโตคอล HTTP (Hypertext Transfer Protocol) ทำให้ผู้พัฒนาสามารถพัฒนาได้ง่าย และเหตุที่มีการใช้สัญญาณ SIP Message ตลอดขั้นตอนการส่งสัญญาณควบคุมรวมถึงลำดับขั้นตอนการส่งสัญญาณตามมาตรฐาน SIP ไม่ซับซ้อนจึงทำให้โปรโตคอล SIP สามารถทำงานได้รวดเร็วกว่ามาตรฐาน H.323
3. ตามมาตรฐานของ SIP จะรับผิดชอบขั้นตอนการส่งสัญญาณควบคุมในส่วนก่อนและหลังการสนทนาเท่านั้น โดยในขั้นตอนการส่งข้อมูลมัลติมีเดีย (Media Stream) จะใช้โปรโตคอล RTP (Real Time Transfer Protocol)[5]และในขั้นตอนการแลกเปลี่ยนความสามารถในการส่งข้อมูลมัลติมีเดีย (Media Capabilities Exchange) จะใช้โปรโตคอล SDP (Session Description Protocol)[6]ทำงานร่วมด้วย

4. สถาปัตยกรรมตามมาตรฐาน SIP เป็นแบบ Client/Server โดยมีการอ้างอิงตัว SIP Client โดยใช้ชื่ออ้างอิงถึงที่อยู่ของตัว SIP Client นั้น ๆซึ่งถูกเรียกว่า SIP URL (Uniform Resource Locators) ทำให้เกิดความสะดวกและความยืดหยุ่นในการใช้บริการเพิ่มมากขึ้น

2.1.2 องค์ประกอบสำคัญในระบบ SIP (SIP Component)



รูปที่ 2-1 องค์ประกอบสำคัญในระบบ SIP (SIP Component)

จากรูปที่ 2-1 แสดงส่วนต่าง ๆ ที่ทำงานร่วมกันอย่างเป็นระบบ เนื่องจากโปรโตคอล SIP มีการทำงานแบบ Client/Server ดังนั้นจึงสามารถแบ่งออกเป็น 2 ส่วนหลัก คือ User Agent (SIP Client) และ Network Server (SIP Server)

2.1.3 User Agent

เป็นนิยามของ SIP Client ภายในระบบ SIP โดยสามารถแยกเป็น 2 รูปแบบการทำงาน คือ

- UAC (User Agent Client) ทำหน้าที่ในการสร้างหรือเริ่มต้นการติดต่อโดยส่งสัญญาณร้องขอ (Request Signal) ไปยังปลายทาง
- UAS (User Agent Server) ทำหน้าที่ในการตอบรับการโทรและส่งสัญญาณตอบกลับ (Response Signal)

โดยปกติแล้ว SIP Client จะทำหน้าที่ได้ทั้ง UAC และ UAS เพื่อให้สามารถทำงานได้ทั้งเป็นผู้โทร (Caller) และผู้ถูกเรียก (Callee)

2.1.4 Network Server

เป็นนิยามของ SIP Server ที่มีรูปแบบการทำงานแบบต่าง ๆ ได้แก่

- **Registrar Server** ทำหน้าที่รับการลงทะเบียนจากผู้ใช้บริการ (SIP Client) เพื่อเก็บข้อมูลที่อยู่ ชื่อ SIP URL และข้อมูลสำคัญ ๆของผู้ใช้บริการ
- **Proxy Server** ทำหน้าที่เป็นตัวกลางในการส่งสัญญาณระหว่าง Caller และ Callee โดยรับผิดชอบการส่งสัญญาณตลอดขั้นตอนการส่งสัญญาณตามมาตรฐาน SIP
- **Redirect Server** ทำหน้าที่ระบุที่อยู่ของ Callee ที่ได้ทำการลงทะเบียนไว้ โดยจะส่งที่อยู่ที่ค้นหาได้กลับไปยัง Caller

นอกจาก User Agent และ Network Server แล้วภายในระบบ SIP ยังมีส่วนการทำงานอื่น ๆ ได้แก่

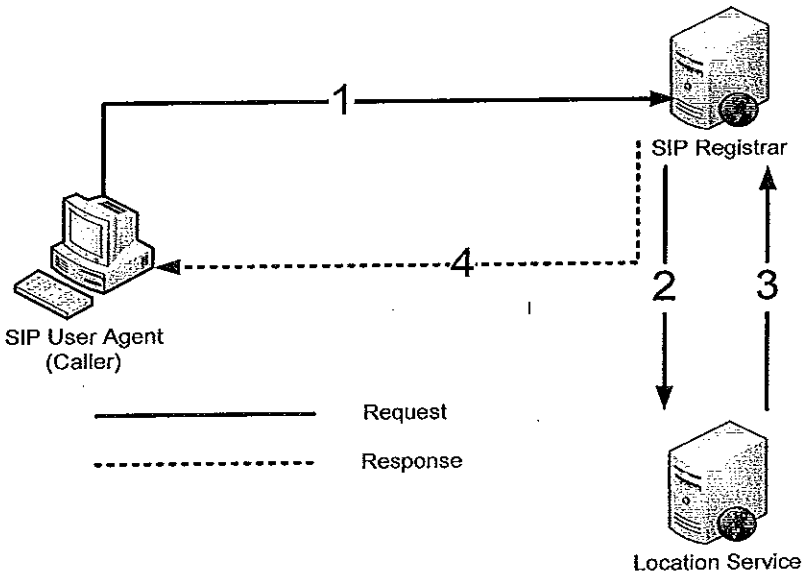
- **Location Server** ทำหน้าที่ในการเก็บข้อมูลหรือเป็นฐานข้อมูลผู้ใช้บริการให้กับตัว Network Server ได้ ตามมาตรฐาน SIP อนุญาตให้สามารถพัฒนาตัว Location Server ไว้เป็นตัวเดียวกันกับตัว SIP Server ได้
- **SIP Gateway** ทำหน้าที่ในการแปลงสัญญาณและข้อมูลมัลติพหุสื่อระหว่างระบบเครือข่ายที่แตกต่างกัน เช่น การเชื่อมต่อไปยังระบบ PSTN (Public Switching Telephone Network), ระบบ SS7 (Signaling System 7) เป็นต้น

2.2 การทำงานของ SIP Server ในระบบ SIP

ลักษณะการทำงานของ SIP Server สามารถทำงานได้หลายรูปแบบ ดังนี้

2.2.1 Registrar Server

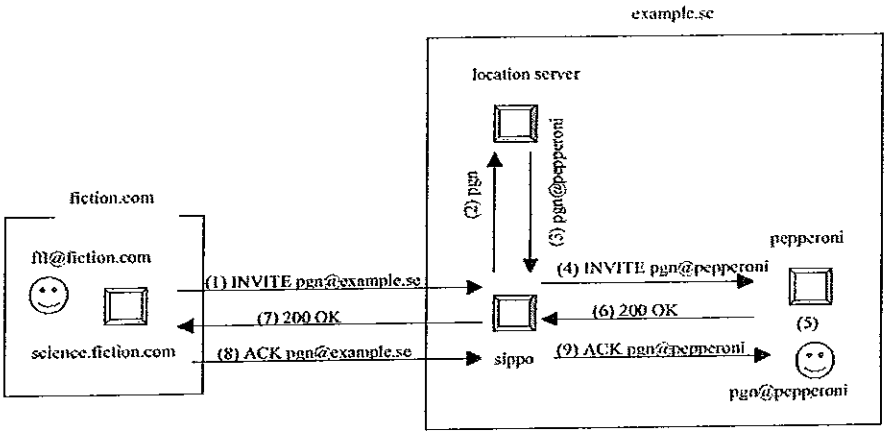
เป็น SIP Server ที่ทำหน้าที่ในการรับการลงทะเบียนจาก SIP Client โดยจะทำการเก็บข้อมูลทั้งหมดไว้ใน Location Server ซึ่งโดยทั่วไป Registrar Server ก็คือ SIP Sever เมื่อทำหน้าที่รับการลงทะเบียนนั่นเอง ดังแสดงในรูปที่ 2-2 เมื่อผู้ร้องขอการติดต่อทำการส่ง Request Message ไปยัง SIP Registrar Server ตัว SIP Registrar Server จะทำการติดต่อไปยัง Location Service หรือ Location Server เพื่อถามหาที่อยู่ของผู้ถูกร้องขอการติดต่อโดยสัญญาณที่ใช้นี้ตามมาตรฐานของ SIP ไม่มีการกำหนดไว้ ดังนั้น SIP Registrar Server สามารถกำหนดรูปแบบของสัญญาณที่ใช้ติดต่อระหว่าง SIP Registrar Server กับ Location Server ได้เอง ต่อมาเมื่อ Location Server สามารถหาที่อยู่ได้แล้วก็จะส่งมาบอกกับ SIP Registrar Server สุดท้าย SIP Registrar Server จะส่ง Response Message ไปให้ผู้ร้องขอการติดต่อเพื่อบอกที่อยู่ของผู้ถูกร้องขอการติดต่อตามที่ผู้ร้องขอการติดต่อต้องการดังรูปที่ 2-2



รูปที่ 2-2 แสดงการทำงานแบบ Registrar Server

2.2.2 Proxy Server

เป็น SIP Server ที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ร้องขอการติดต่อและผู้ถูกร้องขอการติดต่อ โดยการทำงานในขั้นตอน Call Setup, Media & Channel Control และ End Call การส่งสัญญาณทุกอย่างในขั้นตอนทั้งหมดนี้จะต้องทำผ่าน SIP Server โดย SIP Server จะทำการตรวจสอบ SIP Message ที่ได้รับทุกครั้งและจะทำการส่ง SIP Message นั้นไปยังปลายทางที่ต้องการ โดยอาจมีการเปลี่ยนแปลงรูปแบบของ SIP Message ตามความเหมาะสม สำหรับการทำงานในขั้นตอนการส่งข้อมูลเสียง (Media Stream) จะไม่ผ่าน SIP Server นั่นคือ SIP Client ทั้ง 2 ฝ่ายจะทำการส่งข้อมูลเสียงกันโดยตรง เมื่อมองในระดับ Logical แต่หากมองในระดับ Physical แล้ว SIP Client ทั้ง 2 ฝ่ายอาจมีการส่งข้อมูลผ่าน Server หรือ Gateway ก็ได้ เนื่องจากการทำงานทั้งหมดมีพื้นฐานการทำงานอยู่บนระบบเครือข่ายอินเตอร์เน็ต ดังนั้นการส่งข้อมูลสามารถส่งผ่านแต่ละโหนด (Node) เพื่อให้สามารถส่งข้อมูลไปยังปลายทางที่ต้องการได้ ดังรูปที่ 2-3



รูปที่ 2-3 แสดงตัวอย่างการทำงานของ SIP Proxy Server

การทำงานของ Proxy Server

สามารถอธิบาย โดยพิจารณาตัวอย่างการทำงาน จากรูปที่ 2-3 ผู้ร้องขอการติดต่อมี SIP URL คือ ffl@fiction.com โดยมีที่อยู่จริงที่ science.fiction.com ซึ่งอยู่ในเครือข่ายของ fiction.com สำหรับผู้ถูกร้องขอการติดต่อมี SIP URL คือ pgn@example.se และมีที่อยู่คือ pgn@pepperoni อยู่ในเครือข่ายของ example.se ซึ่งมี SIP Server เป็นแบบ Proxy Server โดยมีชื่อว่า คือ sippo เมื่อผู้ร้องขอการติดต่อ ffl@fiction.com ต้องการติดต่อไปยัง pgn@pepperoni โดยผู้ถูกร้องขอการติดต่อนี้ได้ทำการ Register ไว้กับ SIP Server sippo แล้ว

1. เริ่มต้นผู้ร้องขอการติดต่อจะทำการส่ง INVITE Message ไปยัง SIP Server เพื่อแจ้งว่าต้องการติดต่อไปยังผู้ถูกร้องขอการติดต่อใด โดยทำการระบุ SIP URL ของผู้ถูกร้องขอการติดต่อด้วย นั่นคือ pgn@example.se
2. เมื่อ SIP Server ได้รับ INVITE Message SIP Server จะทำการหาที่อยู่โดยถามจาก Location Server โดยส่งชื่อของผู้ถูกร้องขอการติดต่อไปให้จากนั้น Location Server จะหาที่อยู่แล้วตอบกลับมาในที่นี้คือ pgn@pepperoni
3. จากนั้น SIP Server จะทำการเปลี่ยนแปลง SIP Message เพื่อให้สามารถส่งไปยังปลายทางที่ pgn@pepperoni ได้
4. หลังจากนั้น SIP Server จะส่ง INVITE Message pgn@pepperoni ไปยังจุดหมายที่ถูกต้อง เ
5. มื่อ pgn ได้รับแล้วจึงตอบรับการติดต่อโดยส่ง Response Message เป็น Status Code 200 OK กลับมายัง SIP Server
6. จากนั้น SIP Server ก็ทำการส่งต่อ SIP Message ไปให้ ffl@fiction.com เพื่อให้รับรู้ว่าได้รับ Response Message นี้แล้วจึงต้องมีการส่ง ACK Message ไปแจ้ง SIP Server เมื่อ

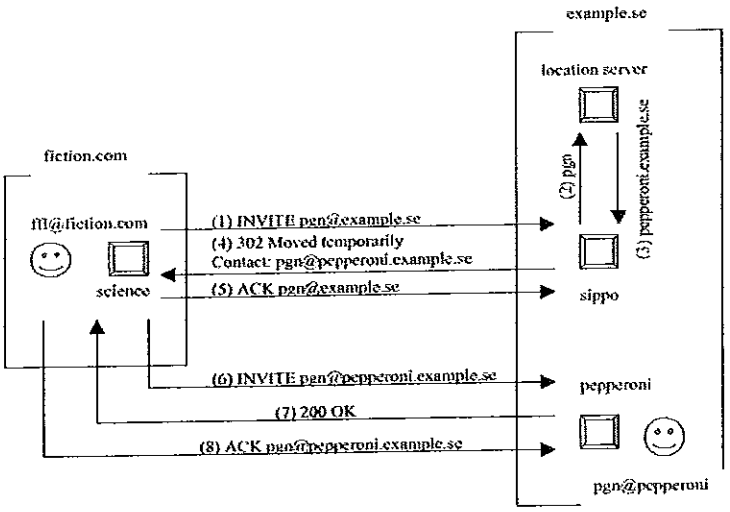
ได้รับ ACK Message ก็ทำการส่งต่อไปยัง pgn โดยทำการเปลี่ยนแปลง SIP Message เพื่อระบุที่อยู่ของ pgn นั่นคือ ACK pgn@pepperoni

7. หลังจากขั้นตอนนี้ SIP Client ทั้ง 2 ฝ่ายจะทำการส่งข้อมูลเสียงโดยไม่ผ่าน SIP Server

ดังจะเห็นได้ว่า Proxy Server จะทำหน้าที่ในการรับภาระหรือรับผิดชอบการติดต่อระหว่างผู้ร้องขอการติดต่อและผู้ถูกร้องขอการติดต่อ ทำให้ Proxy Server สามารถทำการเพิ่มเติมความสามารถต่าง ๆเพื่อให้สามารถให้บริการ SIP Client ได้ดียิ่งขึ้น แต่อย่างไรก็ตามการทำงานในลักษณะนี้ทำให้ต้องสูญเสียเวลาในช่วงการติดต่อที่ทำโดยตัว Proxy Server และ Proxy Server สามารถให้บริการ SIP Client ได้จำกัดขึ้นอยู่กับความสามารถของ Proxy Server นั้น ๆที่จะสามารถรองรับการบริการกับ SIP Client ที่ทำการติดต่อได้มากน้อยเพียงไร

2.2.3 Redirect Server

เป็น SIP Server ที่ทำหน้าที่ในการระบุที่อยู่ของผู้รับการติดต่อที่ได้ทำการลงทะเบียนไว้ โดยจะทำการหาที่อยู่จาก Location Server ซึ่งเมื่อหาที่อยู่ได้แล้วจะส่งที่อยู่ทั้งหมดที่หาได้ให้กับผู้ร้องขอการติดต่อ ดังนั้นหลังจากที่ SIP Client ได้ที่อยู่ในการติดต่อแล้วจะต้องทำการติดต่อไปยังปลายทางที่ต้องการตามที่อยู่ที่ได้นั้นเอง โดยหากผู้ร้องขอการติดต่อได้ที่อยู่มากกว่า 1 ที่อยู่ ผู้ร้องขอการติดต่อจะต้องทำการติดต่อไปยังทุก ๆที่อยู่เพื่อหาที่อยู่จริง ๆของผู้ถูกร้องขอการติดต่อเอง ดังจะเห็นได้ว่าการทำงานในลักษณะนี้จะทำให้ภาระในการติดต่อทั้งหมดตกอยู่ที่ตัว SIP Client แต่การทำงานลักษณะนี้จะทำให้ SIP Client สามารถทำการติดต่อได้อย่างรวดเร็วในกรณีที่ได้ที่อยู่เดียวจาก SIP Server และตัว SIP Server สามารถให้บริการกับ SIP Client ได้จำนวนมาก ดังรูปที่ 2-4



รูปที่ 2-4 แสดงตัวอย่างการทำงานของ SIP Redirect Server

การทำงานของ Redirect Server

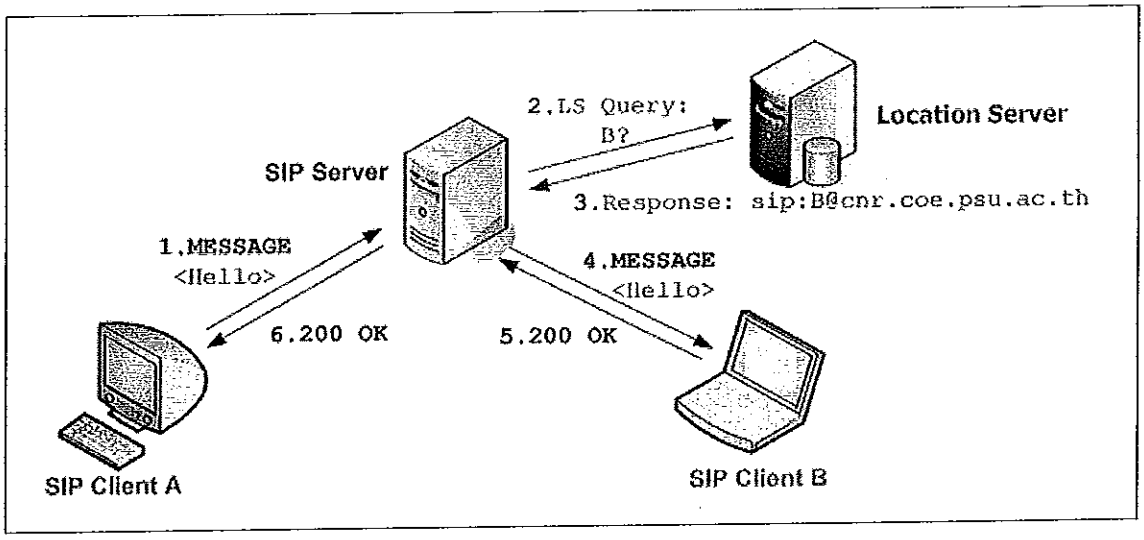
สามารถอธิบายโดยพิจารณาตัวอย่างการทำงาน ผู้ร้องขอการติดต่อมี SIP URL คือ ffl@fiction.com อยู่ในเครือข่ายของ fiction.com ผู้ถูกร้องขอการติดต่อ ชื่อ pgn มีที่อยู่คือ pgn@pepperoni โดยมี SIP URL คือ pgn@example.se อยู่ในเครือข่าย example.se ซึ่งมี SIP Server แบบ Redirect Server ชื่อ sippo เมื่อผู้ร้องขอการติดต่อ ffl@fiction.com ต้องการติดต่อไปยัง pgn ผู้ร้องขอการติดต่อจะทำการติดต่อไปยัง Redirect Server เพื่อถามหาที่อยู่ของ pgn ดังนั้นจึงส่ง INVITE Message (1) ไปยัง Redirect Server จากนั้น Server จะทำการถามที่อยู่จาก Location Server โดยส่งชื่อ pgn (2) ไป เมื่อ Location Server สามารถหาที่อยู่ได้แล้วจะส่งที่อยู่ที่ได้กลับมาคือ pepperoni.example.se (3) จากนั้น Redirect Server จะส่งที่อยู่นี้ไปให้ผู้ร้องขอการติดต่อ ffl@fiction.com โดยส่งเป็น Response Message ซึ่งมี Status Code คือ 302 Moved temporarily (4) ซึ่งจะใช้สำหรับ SIP Server แบบ Redirect Server และเพื่อเป็นการยืนยันว่าได้รับที่อยู่นี้แล้วผู้ร้องขอการติดต่อจึงทำการส่ง ACK Message (5) ไปยัง Redirect Server โดยขั้นตอนการติดต่อหลังจากนี้ผู้ร้องขอการติดต่อจะทำการติดต่อกับปลายทางที่ต้องการเอง โดยเริ่มตั้งแต่การส่ง INVITE Message (6) ไปยัง pgn ตามที่อยู่ที่ได้จาก Redirect Server หลังจากนั้นเมื่อ pgn ได้รับการติดต่อก็จะส่ง Response Message (7) กลับมา และเพื่อให้รับรู้ว่าได้รับการตอบรับแล้วจึงส่ง ACK Message (8) บอกไปยัง pgn หลังจากนี้ทั้ง 2 ฝ่ายก็จะทำการส่งข้อมูลเสียงตามลำดับ

2.3 การให้บริการเสริมของ SIP Server

การให้บริการเสริมของ SIP Server แต่ละรูปแบบของการให้บริการเสริมนั้น ๆ จะมีขั้นตอนและกระบวนการทำงานที่แตกต่างกันออกไป ซึ่งแต่ละบริการเสริมจะอ้างอิงจาก Draft-ietf-sipping-service-examples [7] เป็นหลักซึ่งมีรายละเอียดดังนี้

2.3.1 Instance Message Service

Instance Message Service คือการบริการที่ให้ SIP Client สามารถส่งข้อความระหว่างกันได้ โดยมี SIP Server เป็นตัวกลางในการติดต่อ โดยรูปแบบของการทำงานดังรูปที่ 2-5



รูปที่ 2-5 แสดงรูปแบบการให้บริการ Instance Message ของ SIP Server [7]

การให้บริการ Instance Message ของ SIP Server ซึ่งมีองค์ประกอบดังนี้

- **SIP Client A** คือผู้ที่ต้องการส่งข้อความไปยัง SIP Client B
- **SIP Client B** คือผู้ที่รับข้อความจาก SIP Client A
- **SIP Server** คือเครื่อง Server ที่ทำหน้าที่ในการให้บริการ Instance Message
- **Location Server** คือเครื่อง Server ที่เป็นที่เก็บข้อมูลของ SIP Client ที่ทำการติดต่อกับ SIP Server โดย Location Server สามารถอยู่รวมภายในเครื่องเดียวกับ SIP Server

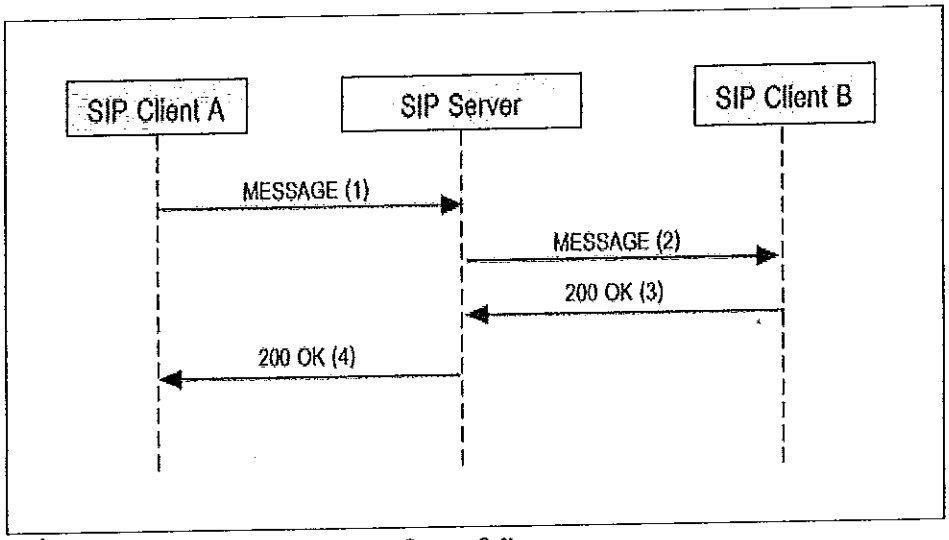
การให้บริการ Instance Message ของ SIP Server ซึ่งมีลำดับการทำงานดังนี้

1. MESSAGE <Hello> เป็นสัญญาณร้องขอการบริการ Instance Message จาก SIP Client A ที่ต้องการส่งข้อความไปยัง SIP Client B โดยสัญญาณในการร้องขอนั้นจะถูกส่งไปยัง SIP Server
2. LS Query: B? เมื่อ SIP Server ได้รับสัญญาณ MESSAGE จาก SIP Client A แล้ว SIP Server จะให้ Location Server ทำการค้นหาว่า SIP Client B ที่ SIP Client A ต้องการจะทำการติดต่อด้วยในขณะนั้น ได้ทำการ register ไว้ใน SIP Server หรือไม่
3. Response: sip:B@cnr.coe.psu.ac.th เมื่อ Location Server ทำการตรวจสอบแล้วพบว่า SIP Client B ทำการ register กับ SIP Server แล้ว Location Server จะทำการส่งสัญญาณตอบกลับไปยัง SIP Server พร้อมกับ SIP URL ของ SIP Client B
4. MESSAGE <Hello> เป็นสัญญาณ MESSAGE ที่ SIP Server ทำการส่งต่อไปยัง SIP Client B ตาม SIP URL ที่ Location Server ส่งมา

5. 200 OK เป็นสัญญาณตอบกลับจาก SIP Client B ที่ส่งไปยัง SIP Client A โดยผ่าน SIP Server เพื่อยืนยันว่า SIP Client B ได้รับข้อความเรียบร้อยแล้ว
6. 200 OK เป็นสัญญาณตอบกลับจาก SIP Client B ที่ SIP Server ส่งต่อไปยัง SIP Client A เมื่อ SIP Client A ได้รับสัญญาณตอบกลับนี้ เป็นการยืนยันว่า SIP Client B ได้รับข้อความที่ SIP Client A ส่งมาเรียบร้อยแล้ว

จากลำดับการทำงาน 1-6 เป็นตัวอย่างขั้นตอนการทำงานในการให้บริการ Instance Message ของ SIP Server ซึ่งในกรณีที่ SIP Client B ต้องการส่งข้อความไปยัง SIP Client A ก็มีลำดับการทำงานเช่นเดียวกัน โดยแทนที่สัญญาณ MESSAGE <Hello> จะถูกส่งจาก SIP Client A ก็จะถูกส่งจาก SIP Client B แทน ส่วนขั้นตอนการทำงานอื่น ๆ ก็มีความคล้ายคลึงกัน

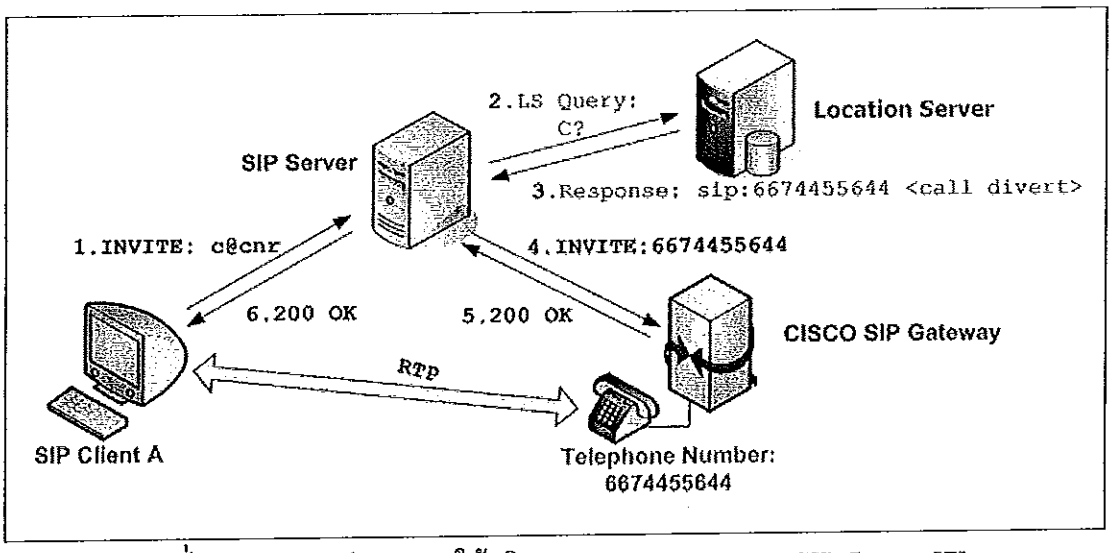
เมื่อนำสัญญาณ SIP ที่ใช้ในการติดต่อจากรูปที่ 2-5 มาเขียนเป็นแผนภาพแสดลำดับการส่งสัญญาณ ได้ดังรูปที่ 2-6



รูปที่ 2-6 แสดงลำดับสัญญาณ SIP ในการให้บริการ Instance Message [7]

2.3.2 Call Forward Service

Call Forward Service คือการบริการโอนสายอัตโนมัติ โดย SIP Client ที่ต้องการใช้บริการนี้ต้องมีการกำหนดหมายเลขโทรศัพท์หรือ SIP URL ปลายทางที่ต้องการให้ SIP Server ทำการโอนสายเมื่อมีการร้องขอการติดต่อเข้ามา โดยรูปแบบของการทำงานมีดังรูปที่ 2-7



รูปที่ 2-7 แสดงรูปแบบการให้บริการ Call Forward ของ SIP Server [7]

การให้บริการ Call Forward ของ SIP Server ซึ่งมีองค์ประกอบดังนี้

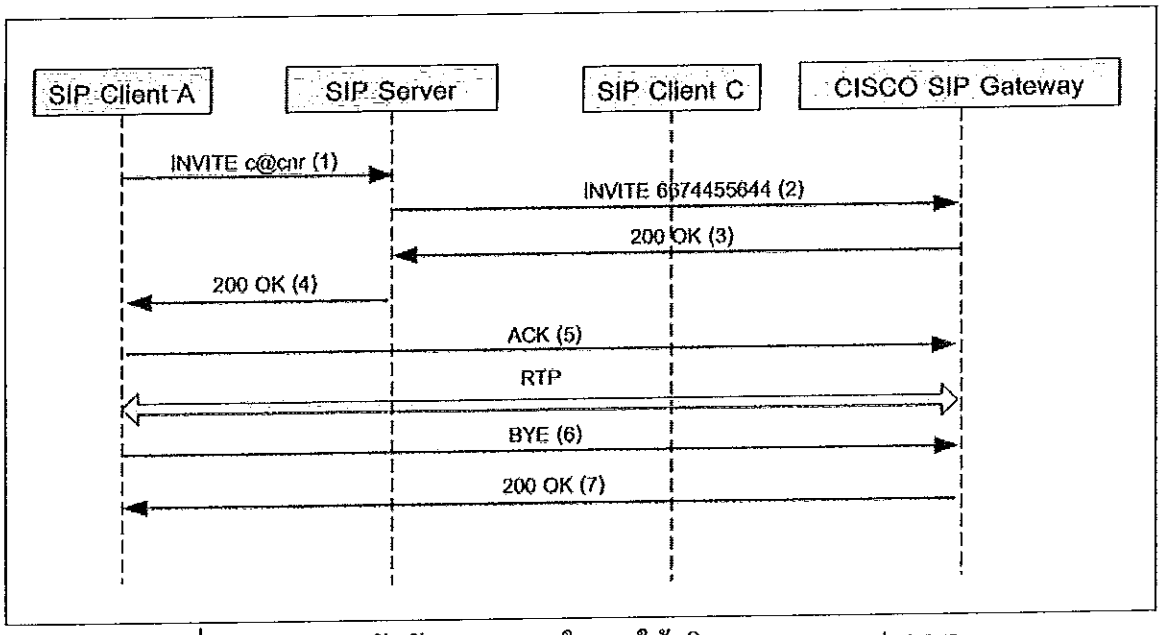
- **SIP Client A** คือผู้ที่ต้องการร้องขอการติดต่อไปยัง c@cnr
- **Telephone** คือโทรศัพท์ที่ c@cnr ได้ทำการกำหนดการโอนสายอัตโนมัติ ไปยังหมายเลขนี้
- **SIP Server** คือเครื่อง Server ที่ทำหน้าที่ในการให้บริการ Call Forward

การให้บริการ Call Forward ของ Server มีลำดับการทำงาน ดังนี้

1. INVITE: c@cnr เป็นสัญญาณร้องขอการติดต่อจาก SIP Client A ไปยัง c@cnr โดยสัญญาณในการร้องขอนั้น จะถูกส่งไปยัง SIP Server
2. LS Query: c? เมื่อ SIP Server ได้รับสัญญาณ INVITE จาก SIP Client A แล้ว SIP Server จะให้ Location Server ทำการค้นหาว่า c@cnr ที่ SIP Client A ต้องการจะทำการติดต่อด้วยในขณะนั้น ได้ทำการ register ไว้ใน SIP Server หรือไม่
3. Response: sip:6674455644 <Call Forward> เมื่อ Location Server ทำการตรวจสอบแล้วพบว่า c@cnr ได้กำหนดหมายเลขโทรศัพท์ปลายทางที่ต้องการให้ SIP Server ทำการโอนสายเมื่อมีการร้องขอการติดต่อเข้ามา แล้ว Location Server จะทำการส่งสัญญาณตอบกลับไปยัง SIP Server พร้อมกับหมายเลขโทรศัพท์ที่ต้องการให้มีการโอนสาย
4. INVITE: 66744556444 เป็นสัญญาณร้องขอการติดต่อที่ SIP Server ทำการส่งต่อไปยังหมายเลขโทรศัพท์ปลายทาง โดยผ่าน CISCO SIP Gateway เป็นตัวกลางในการแปลงสัญญาณระหว่าง SIP Protocol และ SS7
5. 200 OK เป็นสัญญาณตอบกลับจาก CISCO SIP Gateway ที่ส่งไปยัง SIP Client A โดยผ่าน SIP Server

6. 200 OK เป็นสัญญาณตอบกลับจาก CISCO SIP Gateway ที่ SIP Server ส่งต่อไปยัง SIP Client A เมื่อ SIP Client A ได้รับสัญญาณตอบกลับนี้ เป็นการยืนยันว่า CISCO SIP Gateway ได้รับสัญญาณ INVITE เรียบร้อยแล้ว

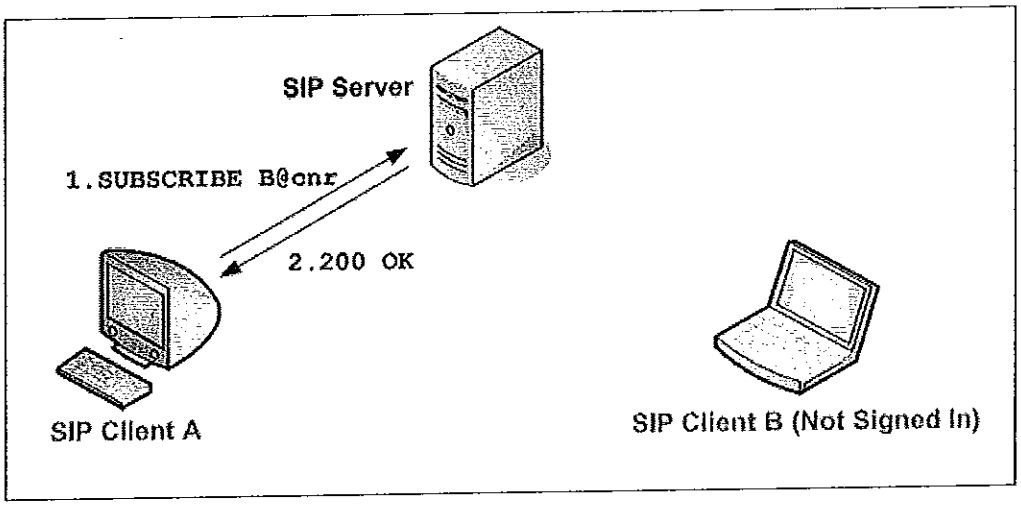
เมื่อนำสัญญาณ SIP ที่ใช้ในการติดต่อจากรูปที่ 2-7 มาเขียนเป็นแผนภาพแสดงลำดับการส่งสัญญาณได้ดังรูปที่ 2-8



รูปที่ 2-8 แสดงลำดับสัญญาณ SIP ในการให้บริการ Call Forward [7]

2.3.3 Presence Service

Presence Service คือการบริการตรวจสอบการสถานะอัตโนมัติของ SIP Client โดย SIP Client แต่ละคน จะมี Contact List ซึ่งมีลักษณะเป็น SIP URL ของผู้ที่เราต้องการจะทำการติดต่อด้วย โดยรูปแบบของการทำงานดังรูปที่ 2-9



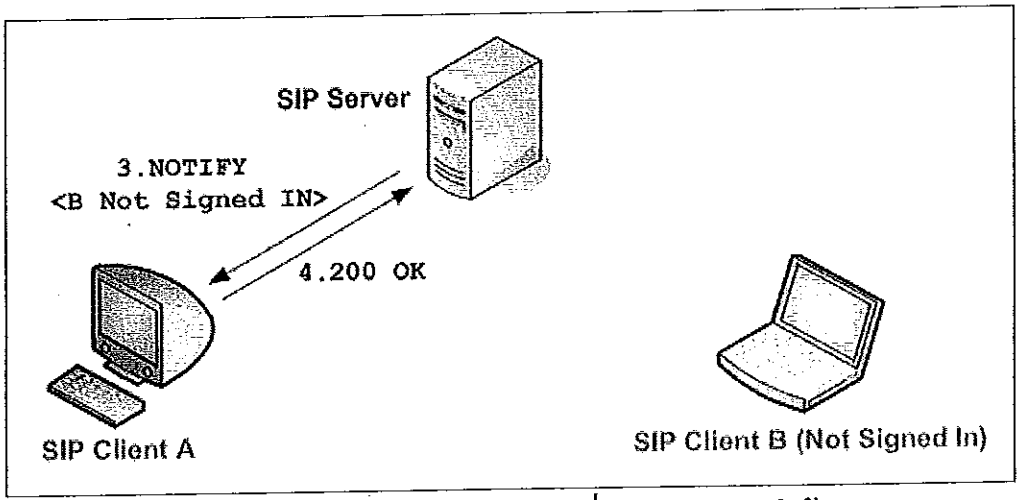
รูปที่ 2-9 รูปแบบบริการ Presence ของ SIP Server เมื่อ SIP Client B ยังไม่ทำการ Register [7]

การให้บริการ Presence ของ SIP Server ซึ่งมีองค์ประกอบดังนี้

- **SIP Client A** คือผู้ที่ต้องการทราบสถานะการออนไลน์ของ SIP Client B
- **SIP Client B** คือผู้ที่ SIP Client A ต้องการทราบสถานะการออนไลน์
- **SIP Server** คือเครื่อง Server ที่ทำหน้าที่ในการให้บริการ Presence

การให้บริการ Presence ของ SIP Server ซึ่งมีลำดับการทำงานดังนี้

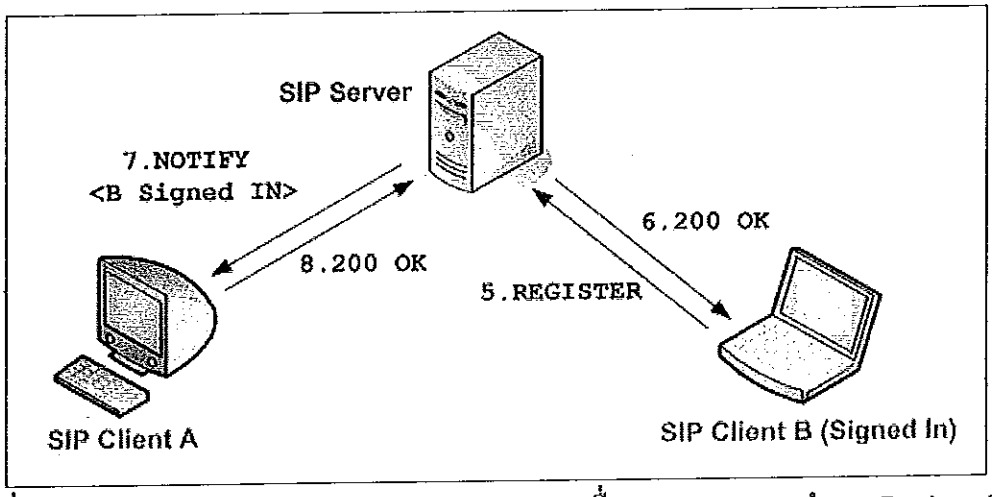
1. SUBSCRIBE B@cnr เป็นสัญญาณร้องขอการทราบสถานะการออนไลน์ของ SIP Client B ที่ SIP Client A ส่งไปยัง SIP Server
2. 200 OK เมื่อ SIP Server ได้รับสัญญาณร้องขอการทราบสถานะการออนไลน์แล้วก็จะทำการส่งสัญญาณตอบกลับ 200 OK ไปยัง SIP Client A



รูปที่ 2-10 รูปแบบบริการ Presence ของ SIP Server เมื่อ SIP Server ส่งสัญญาณ NOTIFY [7]

ขณะนั้น SIP Client B ยังไม่ได้ทำการ register กับ SIP Server ดังรูปที่ 2-10

1. NOTIFY <B Not Signed IN> เมื่อ SIP Server ทำการตรวจสอบสถานะการออนไลน์ของ SIP Client B แล้ว ก็ทำการส่งสัญญาณ NOTIFY เพื่อเป็นการแจ้งสถานะให้ SIP Client A ทราบว่าในขณะนี้ SIP Client B ยังไม่ได้ทำการออนไลน์
2. 200 OK เป็นสัญญาณตอบกลับจาก SIP Client A เพื่อเป็นการยืนยันว่าได้รับสัญญาณ NOTIFY

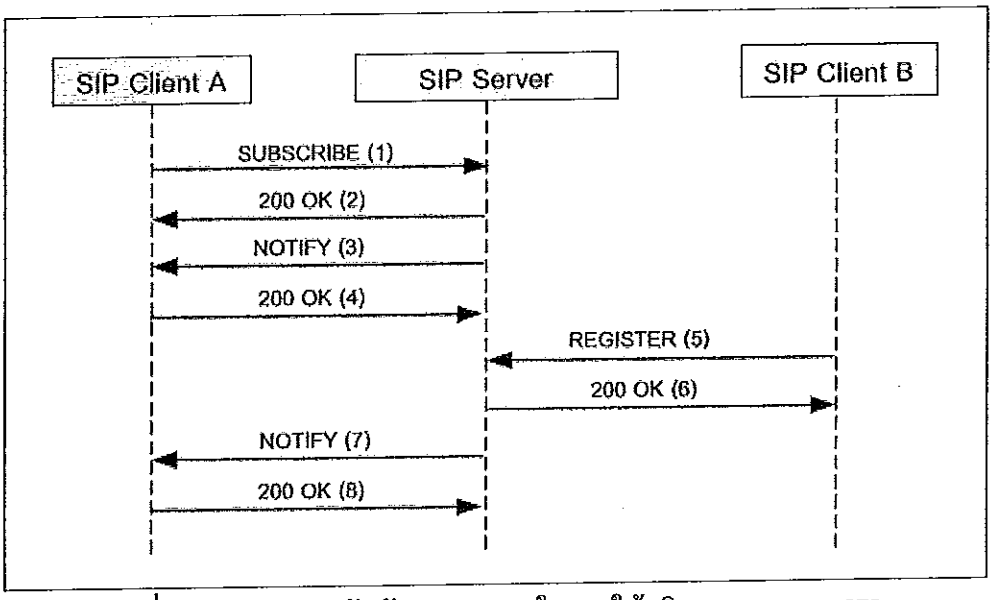


รูปที่ 2-11 รูปแบบบริการ Presence ของ SIP Server เมื่อ SIP Client B ทำการ Register [7]

ในขณะนั้น SIP Client B ทำการ register กับ SIP Server ดังรูปที่ 2-11

1. REGISTER เป็นสัญญาณร้องการลงทะเบียนกับ SIP Server
2. 200 OK เป็นสัญญาณตอบกลับจาก SIP Server ว่าการลงทะเบียนของ SIP Client Bนั้นเสร็จสมบูรณ์แล้ว
3. NOTIFY <B Signed IN> SIP Server ทำการการแจ้งสถานะการออนไลน์ของ SIP Client B ให้ SIP Client A ทราบอีกครั้งว่าในขณะนี้ SIP Client B ทำการออนไลน์
4. 200 OK เป็นสัญญาณตอบกลับจาก SIP Client A เพื่อเป็นการยืนยันว่าได้รับสัญญาณ NOTIFY แล้ว

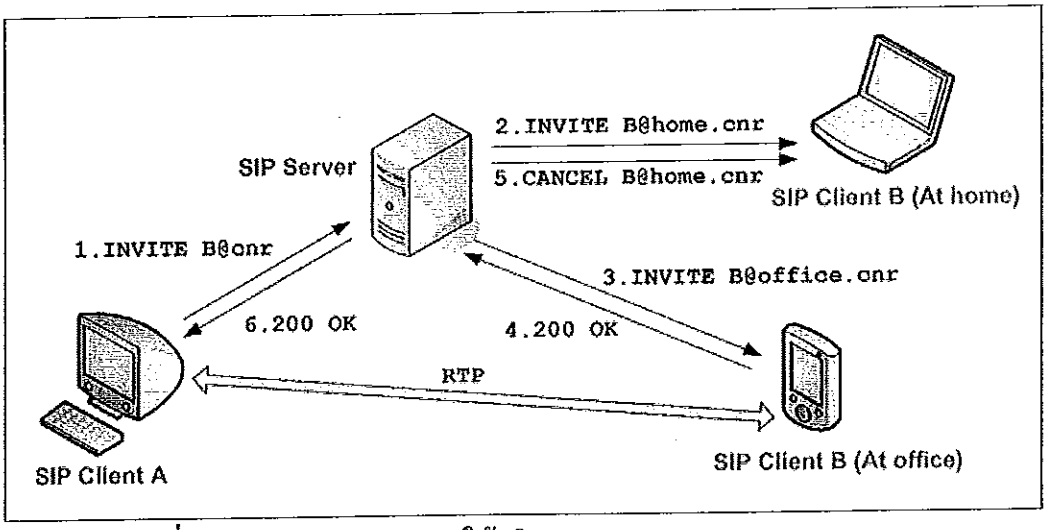
สัญญาณ NOTIFY เป็นสัญญาณแจ้งสถานะการออนไลน์ของ SIP Client B ที่ถูกส่งออกจาก SIP Server ทุกครั้งที่ SIP Client B ทำการเปลี่ยนแปลงสถานะการออนไลน์ดังรูปที่ 2-12



รูปที่ 2-12 แสดงลำดับสัญญาณ SIP ในการให้บริการ Presence [7]

2.3.4 Forking Service

Forking Service คือการให้บริการของ SIP Server ที่จะทำการส่งสัญญาณ INVITE ไปยังทุกๆ Host ปลายทางที่ SIP Client ทำการ Register ไว้ ซึ่งการบริการนี้ SIP Server จะต้องมีการรองรับให้ SIP Client สามารถทำการ register โดยใช้ SIP URL เดียวกัน บนหลายๆ Host ได้ โดยรูปแบบของการทำงานดังรูปที่ 2-13



รูปที่ 2-13 แสดงรูปแบบการให้บริการ Forking ของ SIP Server [7]

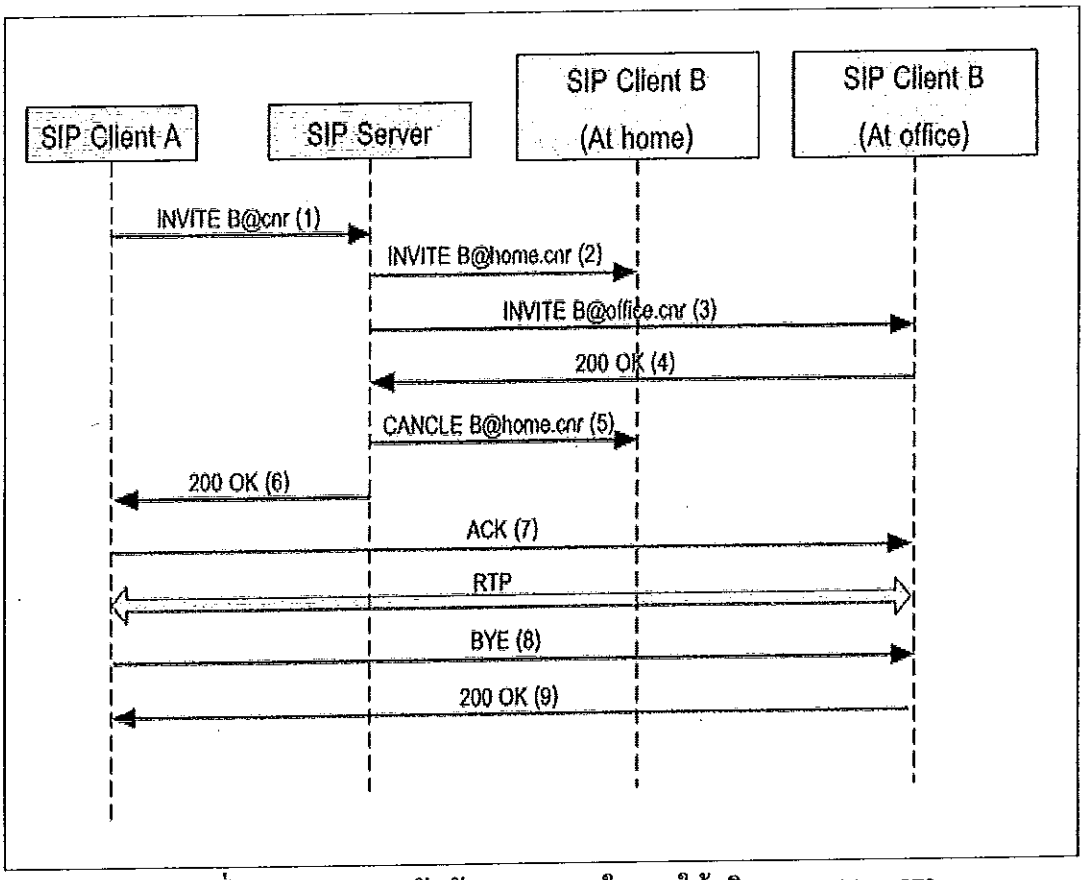
การให้บริการ Forking ของ SIP Server ซึ่งมีองค์ประกอบดังนี้

- **SIP Client A** คือผู้ที่ร้องขอการติดต่อไปยัง SIP Client B
- **SIP Client B** คือผู้ที่รับการติดต่อ โดย SIP Client B ทำการ register กับ SIP Server 2 แห่ง คือที่บ้านและที่ทำงาน
- **SIP Server** คือเครื่อง Server ที่ทำหน้าที่ในการให้บริการ Forking

การให้บริการ Forking ของ SIP Server มีลำดับการทำงาน ดังนี้

1. INVITE B@cnr เป็นสัญญาณที่ SIP Client A ร้องขอการติดต่อไปยัง SIP Client B โดยผ่าน SIP Server เมื่อ SIP Server ได้รับสัญญาณร้องขอการติดต่อจาก SIP Client A แล้ว SIP Server ทำการตรวจสอบว่าในขณะนั้น SIP Client B ทำการ register ไว้ที่ใดบ้าง จากนั้น SIP Server ทำการส่งสัญญาณ INVITE ไปยังทุกๆ Host ที่ SIP Client B ทำการ register เอาไว้
2. INVITE B@home.cnr เป็นสัญญาณที่ SIP Server ส่งสัญญาณร้องขอการติดต่อจาก SIP Client A ไปยัง SIP Client B ที่ทำการ register ไว้ที่บ้าน
3. INVITE B@office.cnr เป็นสัญญาณที่ SIP Server ส่งสัญญาณร้องขอการติดต่อจาก SIP Client A ไปยัง SIP Client B ที่ทำการ register ไว้ที่ทำงาน แต่ในขณะ SIP Client B อยู่ที่ทำงานและทำการตอบรับการติดต่อของ SIP Client A
4. 200 OK เป็นสัญญาณยืนยันการรับการติดต่อจาก SIP Client B โดยส่งผ่านไปยัง SIP Server
5. CANCEL B@home.cnr เมื่อ SIP Server ได้รับสัญญาณตอบกลับจาก B@office.cnr แล้ว ก็จะทำการส่งสัญญาณ CANCEL ไปยัง Host อื่นๆ ที่ SIP Client B ที่ทำการ register ไว้ เพื่อเป็นการยกเลิกสัญญาณ INVITE B@home.cnr ที่ทำการร้องขอการติดต่อไว้
6. 200 OK เป็นสัญญาณยืนยันการรับการติดต่อจาก SIP Client B ที่ SIP Server ส่งต่อไปยัง SIP Client A

หลังจากนั้น SIP Client A ทำการส่งสัญญาณ ACK ไปยัง SIP Client B และหลังจากที่ SIP Client B ได้รับสัญญาณ ACK แล้ว จะมีการส่ง Media Stream ระหว่าง SIP Client A และ SIP Client B ในกรณีที่ SIP Client ต้องการยกเลิกการติดต่อก็จะทำการส่งสัญญาณ BYE ไปยัง SIP Client ปลายทางได้ ดังรูปที่ 2-14



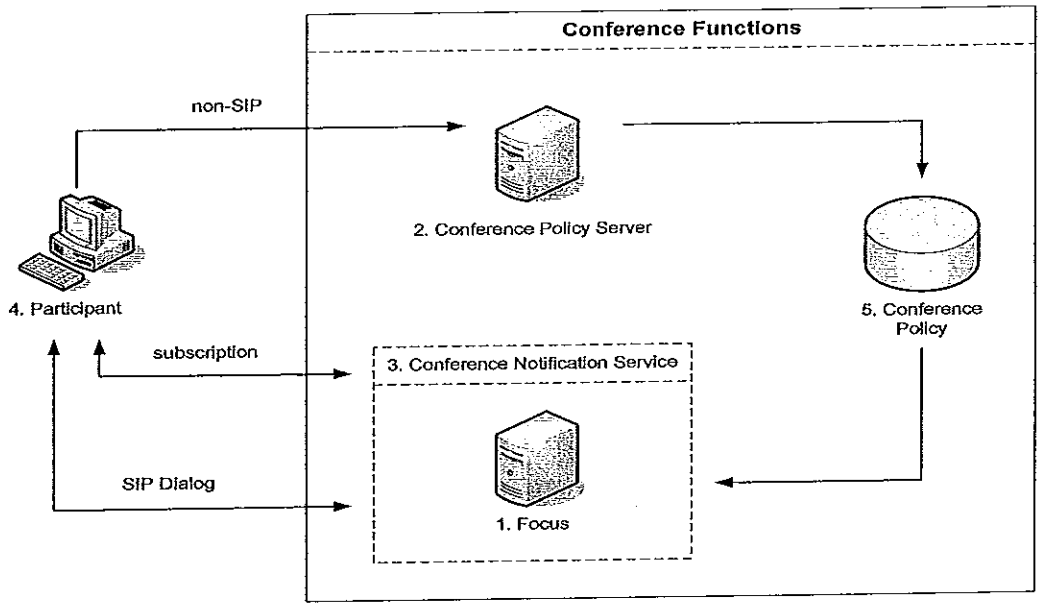
รูปที่ 2-14 แสดงลำดับสัญญาณ SIP ในการให้บริการ Forking [7]

จากรูปแบบการให้บริการทั้งหมดที่กล่าวมา พบว่าโปรโตคอลได้มีการออกแบบบริการเสริมเพื่อรองรับการใช้งานของผู้ใช้ที่หลากหลาย ซึ่งเราสามารถนำแนวคิดของการให้บริการเหล่านี้มาใช้ร่วมในระบบการประชุมแบบสื่อประสมเพื่อให้ผู้เข้าร่วมการประชุมได้รับความสะดวกในการใช้งานประชุม

โดยในการที่จะสร้างระบบการประชุมแบบสื่อประสมโดยใช้โปรโตคอล SIP จำเป็นที่จะต้องมีส่วนประกอบต่าง ๆ ที่เข้ามาใช้ในการควบคุมการติดต่อและสิทธิต่าง ๆของผู้เข้าร่วมการประชุมแต่ละคน ซึ่งจะประกอบไปด้วยการทำงาน 2 ส่วนหลัก ๆคือ [8]

- ส่วนของการติดต่อเพื่อเข้าหรือออกจากการประชุมโดยส่วนจะนี้เราสามารถใช้งานในส่วนของโปรโตคอลมาตรฐานต่าง ๆ มาควบคุมซึ่งในที่นี้คือโปรโตคอล SIP เป็นต้น
- ส่วนควบคุมการประชุมซึ่งทำหน้าที่ในการจัดการทรัพยากรต่าง ๆในการประชุม

ดังแสดงในรูปที่ 2-15



รูปที่ 2-15 Component ต่างๆในระบบ SIP Conference

2.3.5 Focus

เป็นส่วนที่ทำหน้าที่ในการควบคุมสัญญาณการติดต่อของเข้าร่วมการประชุมแต่ละคนที่จะเกิดขึ้นในการประชุม ซึ่งผู้เข้าร่วมการประชุมทุกคนจะต้องเชื่อมต่อไปยัง Focus ผ่านทาง SIP Dialog โดย Focus จะมีหน้าที่ในการดูแลควบคุมสถานะรวมถึงรูปแบบการรับส่งข้อมูลสื่อประสมต่างๆของผู้เข้าร่วมการประชุมแต่ละคน ซึ่งจะถูกกำหนดโดย Membership Policy นอกจากนี้ Focus จะเป็นผู้จัดการสร้าง Media Session ต่างๆ ในการประชุมเพื่อให้แน่ใจว่าผู้เข้าร่วมการประชุมทุกคนจะได้รับ ข้อมูลสื่อประสมที่เหมือนกัน

2.3.6 Conference Policy Server

จะเป็นส่วนที่เชื่อมต่ออยู่ระหว่างผู้เข้าร่วมการประชุมกับ Conference Policy ซึ่ง Conference Policy Server จะเป็นส่วนที่ช่วยให้ผู้ควบคุมการประชุม (Moderator) สามารถกำหนดข้อบังคับต่างๆ (Policy) ที่ใช้ในการประชุม โดยหลังจากผู้ควบคุมการประชุมทำการกำหนดข้อบังคับต่างๆ ผ่านทางConference Policy Server เรียบร้อยแล้ว Conference Policy Server จะทำการบันทึกค่าเหล่านั้นไว้ใน Conference Policy เพื่อให้ Focus สามารถนำข้อบังคับเหล่านั้นไปใช้ในการควบคุมการประชุมต่อไป

โดย Conference Policy Server อาจจะทำงานอยู่ในรูปแบบของ Web Application ซึ่งจะมีความสามารถในการกำหนดรายละเอียดต่าง ๆของผู้ใช้เพื่อเข้าใช้งานการประชุม เช่น สามารถกำหนดให้ผู้เข้าร่วมการประชุมมีสิทธิรับส่งข้อมูลภาพและเสียงต่าง ๆ ประเภทใดในการประชุม เป็นต้น นอกจากนี้ยังสามารถใช้กำหนดรูปแบบของการทำการประชุมว่าให้รูปแบบเป็นอย่างไร ซึ่งสามารถจำแนกได้ 2 รูปแบบ คือ

- **Open Conference** ซึ่งจะเป็นรูปแบบการประชุมที่เป็นแบบไม่เป็นทางการคือผู้ใช้คนใดก็ได้สามารถที่จะเข้ามาร่วมการประชุม เพื่อรับข้อมูลการประชุมจะไม่มีการจำกัดสิทธิการใช้งานของผู้เข้าร่วมการประชุมใด ๆทั้งสิ้น
- **Close Conference** จะเป็นรูปแบบการประชุมที่เป็นแบบทางการ คือผู้ใช้ที่สามารถเข้าประชุมได้ จำเป็นจะต้องได้รับการกำหนดสิทธิก่อนจึงจะสามารถเข้ารับข้อมูลการประชุมได้

โดยทั้งสองรูปแบบนี้จะต้องทำการกำหนดผู้ควบคุมการประชุม ขึ้นมา เพื่อใช้ในการควบคุมการประชุมเหล่านั้น ซึ่งโดยปกติจะเป็นผู้ติดต่อที่มีชื่ออยู่ในลำดับแรกของรายชื่อผู้มีสิทธิเข้าประชุมจะได้รับสิทธินี้ แต่ในกรณีที่ผู้ติดต่อนั้นไม่สามารถมาเข้าร่วมประชุมได้หรือออกจากการประชุมไปก่อน Conference Policy Server จะทำการกำหนดสิทธิของผู้จัดการการประชุมใหม่โดยจะกำหนดให้แก่ผู้ใช้ที่มีรายชื่ออยู่ในลำดับถัดไปแทน

2.3.7 Conference Notification Server

จะมีหน้าที่ในการแจ้งสถานะต่าง ๆของผู้เข้าร่วมการประชุมแต่ละคนที่เกิดขึ้นในการประชุม ซึ่ง Conference Notification Server จะมีการทำงานที่ความสัมพันธ์กับ Focus เนื่องจากในการติดต่อไปยังผู้เข้าร่วมการประชุมแต่ละคนจำเป็นจะต้องทำผ่านทาง SIP Dialog ดังนั้นจึงต้องให้ Focus เป็นผู้สร้างสัญญาณการติดต่อไปยังผู้เข้าร่วมการประชุม โดยจะมีรูปแบบการให้บริการที่มีลักษณะคล้ายคลึงกับการให้บริการ Presence ซึ่ง Conference Notification Server จะทำหน้าที่ในการรับ Subscription จากผู้เข้าร่วมการประชุมแต่ละคนที่อยู่ในการประชุมเพื่อใช้ในการแจ้งสถานะต่าง ๆของการประชุม ให้แก่ผู้เข้าร่วมการประชุมเหล่านั้น

โดยสถานะของการประชุมจะมีการแจ้งเมื่อมีเหตุการณ์ต่าง ๆเกิดขึ้น เช่น มีผู้เข้าร่วมการประชุมคนใหม่เข้ามาร่วมการประชุมหรือในกรณีที่มีผู้เข้าร่วมการประชุมได้ออกจากการประชุม เป็นต้น ซึ่งรูปแบบของข้อมูลสถานะที่จะใช้แจ้งให้กับผู้ติดต่อแต่ละคนได้ทราบ จะอยู่ในรูปแบบของ XML ซึ่งถูกกำหนดในมาตรฐาน RFC 4575

2.3.8 Participants

ส่วนนี้จะหมายถึงผู้เข้าร่วมการประชุมต่าง ๆ ซึ่งอีกความหมายหนึ่งจะหมายถึง SIP User agent ใด ๆที่สามารถติดต่อไปยัง Focus ผ่านทาง SIP Dialog โดย SIP User Agent เหล่านี้อาจจะอยู่ในรูปแบบของ PC Application โทรศัพท์ หรือ PSTN Gateway เป็นต้น

Participants จะต้องเป็นผู้ที่รับสิทธิที่จะสามารถเข้าร่วมการประชุม ซึ่ง Focus จะเป็นผู้ตรวจสอบผ่านทาง Conference Policy เมื่อ Participants ได้ทำการติดต่อเข้ามายังการประชุม โดยสามารถแบ่งประเภทของ Participant ออกได้เป็น 2 ประเภทคือ

- **Conference-Aware Participant** หมายถึงผู้เข้าร่วมการประชุมที่ติดต่อโดยใช้ SIP User Agent ที่มีความสามารถรองรับการทำงานเสริมต่าง ๆ ของระบบการประชุมแบบสื่อประสมได้อย่างสมบูรณ์ เช่น สามารถใช้งาน Conference Notification เป็นต้น
- **Conference-Unaware Participant** หมายถึงผู้เข้าร่วมการประชุมที่ติดต่อโดยใช้ SIP User Agent ที่รองรับการทำงานของโปรโตคอล SIP ตามมาตรฐาน RFC3261 เท่านั้น

2.3.9 Conference Policy

จะเป็นส่วนที่ใช้จัดเก็บกฎข้อบังคับต่าง ๆ เพื่อใช้ในการกำหนดสิทธิการทำงานให้กับ Focus และผู้เข้าร่วมการประชุมแต่ละคน โดย Conference Policy จะอยู่รูปแบบของฐานข้อมูลที่เป็นแบบ XML ซึ่ง Conference Policy จะถูกกำหนดค่าต่าง ๆผ่านทาง Conference Policy Server โดยผู้ที่กำหนดค่าเหล่านี้ต้องเป็นผู้สร้างการประชุมเท่านั้น

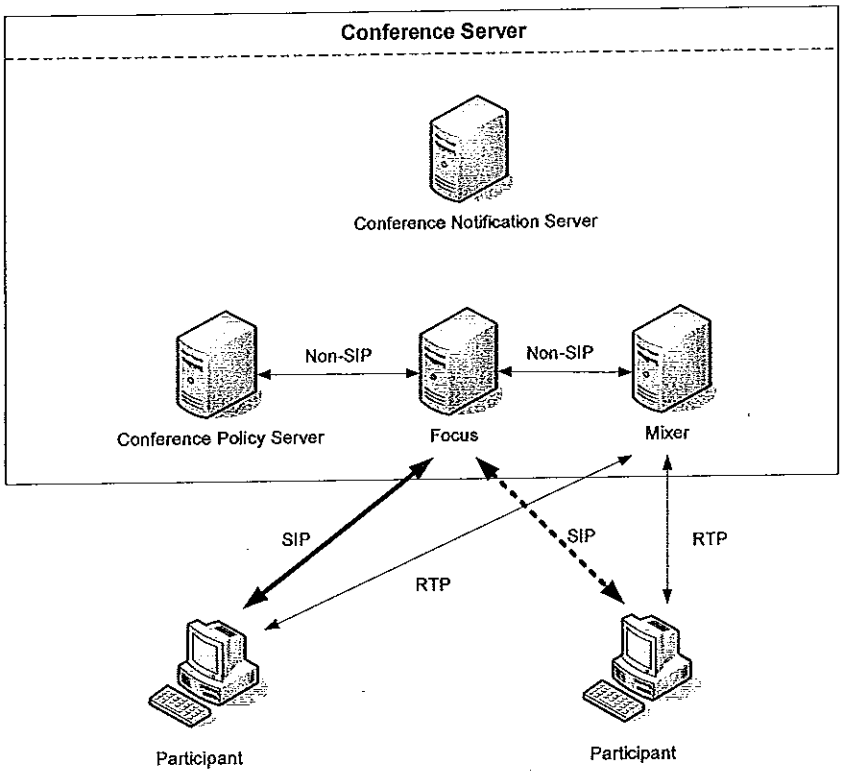
จากรายละเอียดส่วนประกอบต่าง ๆ ที่กล่าวมาทั้งหมด เราสามารถที่จะนำส่วนประกอบต่าง ๆเหล่านี้มาสร้างเป็นรูปแบบการประชุมแบบสื่อสมโดยใช้โปรโตคอล SIP โดยจะมีรูปแบบต่าง ๆ ดังจะกล่าวในหัวข้อต่อไป

2.4 รูปแบบลักษณะการสร้างการประชุมแบบต่าง ๆ

สำหรับรูปแบบการสร้างการประชุมแบบสื่อประสม ได้มีการนิยามรูปแบบการสร้างอยู่หลายรูปแบบ ซึ่งการประชุมแต่ละรูปแบบจะถูกออกแบบมาให้เหมาะสมกับสภาพการใช้งานที่แตกต่างกัน โดยวิเคราะห์จากตัวแปรต่าง ๆ เช่น จำนวนผู้เข้าร่วมการประชุม ฟังก์ชันการทำงานต่าง ๆที่จะต้องใช้ในการประชุม เป็นต้น ซึ่งสามารถแบ่งได้ดังนี้

2.4.1 Centralized Server

เป็นรูปแบบพื้นฐานที่ใช้ในการสร้างการประชุมแบบสื่อประสมโดยใช้โปรโตคอล SIP โดยส่วนประกอบทั้งหมดไม่ว่าจะเป็น Focus, Conference Notification Server, Conference Policy Server, Mixerเป็นต้นจะถูกรวมอยู่ที่เดียว ซึ่งเรียกว่า Conference Server โดยรูปแบบการประชุมลักษณะนี้ผู้เข้าร่วมการประชุมจะต้องสร้างการติดต่อทั้งหมดไปยัง Conference Server เพียงที่เดียวเท่านั้น ซึ่งทำให้สะดวกแก่การควบคุมการทำงานต่าง ๆ เนื่องจากระบบการทำงานจะทำที่ Conference Server ทั้งหมด แต่รูปแบบการติดต่อแบบนี้จะมีข้อเสียในเรื่องที่ Conference Server จะต้องรับภาระการทำงานทั้งหมด ซึ่งไม่เหมาะกับการประชุมที่ผู้เข้าร่วมการประชุมที่มีจำนวนมาก เพราะจะทำให้เกิดปัญหา Single point of failure ได้โดยรูปแบบการติดต่อจะเป็นดังแสดงในรูปที่ 2-16

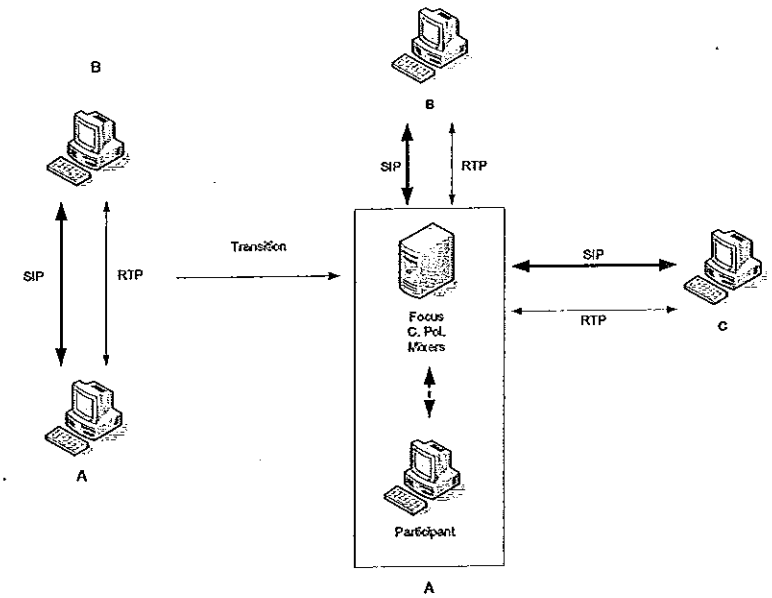


รูปที่ 2-16 Centralized Server [8]

2.4.2 Endpoint Server

สำหรับรูปแบบนี้จะเป็นการสร้างประชุมแบบทันทีทันใด โดยจะเป็นการสร้างการประชุมโดยเปลี่ยนการติดต่อระหว่างผู้ติดต่อสองคนไปเป็นการประชุมที่รองรับการติดต่อแบบหลายคน โดยรูปแบบการติดต่อแบบ Endpoint Server จะเป็นดังแสดงในรูปที่ 2-17

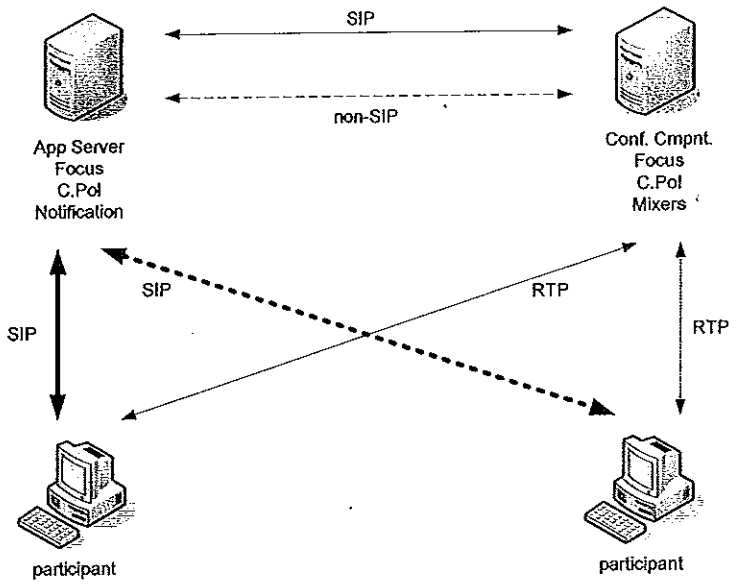
ซึ่งเริ่มจากการที่มีผู้ติดต่อสอง 2 คนกำลังทำการติดต่อแบบ Point To Point แบบปกติ หลังจากนั้น ผู้ติดต่อคนใดคนหนึ่งต้องการทำการเชิญชวน ผู้ติดต่ออีกคนหนึ่งเข้ามาเพื่อทำการประชุม จึงได้ทำการเปลี่ยนการทำงานของตนเองจาก SIP User Agent ให้เป็น Focus, Conference Policy Server และ Mixer และทำการสร้างการติดต่อใหม่เพื่อทำการแจ้งข้อมูลเกี่ยวกับการติดต่อใหม่ให้กับผู้ติดต่อคนอื่นว่าได้เปลี่ยนการติดต่อจากการติดต่อปกติไปเป็นการติดต่อแบบการประชุมแล้ว ซึ่งข้อดีของการสร้างการประชุมในรูปแบบนี้คือไม่จำเป็นต้องมี Conference Server ที่ซับซ้อนและสามารถสร้างการประชุมได้ทันทีทันใด แต่การสร้างการประชุมรูปแบบนี้จะมีข้อเสียคือในกรณีที่มีการติดต่อจำนวนมากภาระการทำงานทั้งหมดจะตกอยู่ที่ผู้ติดต่อที่ทำหน้าที่เป็น Focus เพียงผู้เดียว นอกจากนี้รูปแบบการประชุมแบบนี้จะไม่มีส่วนของ Conference Notification Service ซึ่งทำให้ผู้เข้าร่วมการประชุมแต่คนไม่สามารถทราบได้ว่ามีใครเข้าร่วมหรือออกจากการประชุมไปแล้วบ้าง ดังนั้นจึงสังเกตได้ว่ารูปแบบการประชุมนี้จึงเหมาะกับการประชุมที่มีผู้เข้าร่วมการประชุมที่มีจำนวนไม่มาก



รูปที่ 2-17 Endpoint Server [8]

2.4.3 Media Server Component

รูปแบบนี้จะเป็นการสร้างการประชุมโดยจะทำการแยกส่วนการทำงานของ Conference Server ออกเป็นสองส่วนคือส่วนของ Application Server และส่วนของ Media Server เพื่อลดภาระการทำงานให้กับ Conference Server โดยการทำงานของ Application Server จะมีหน้าที่ในการจัดการกฎข้อบังคับต่าง ๆที่ใช้ในการควบคุมการประชุมและยังมีหน้าที่ควบคุม SIP Dialog ต่าง ๆของผู้เข้าร่วมการประชุมแต่ละคนที่ติดต่อเข้ามายังการประชุม รวมไปถึงการให้บริการ Conference Notification แต่จะไม่มีการทำงานในส่วนของการจัดการ Media Stream ซึ่งหน้าที่ส่วนนี้จะเป็นถูกจัดการผ่านทาง Media Server โดย Media Server จะถูกกำหนดรายละเอียดการทำงานต่าง ๆ เพื่อใช้การควบคุม Media Stream จาก Application Server ซึ่งจะรูปแบบการติดต่อดังแสดงในรูปที่ 2-18 โดยการสร้างการประชุมในรูปแบบนี้จะมีข้อดีคือสามารถลดภาระการทำงานในส่วนของ Conference Server ที่ไม่ต้องมีหน้าที่ในการรวมข้อมูล Media Stream แต่ก็มีข้อเสียคือจำเป็นต้องเพิ่มการทำงานของ Server เป็นสองส่วนซึ่งจะกลายเป็นการเพิ่มความซับซ้อนให้ระบบมากยิ่งขึ้น

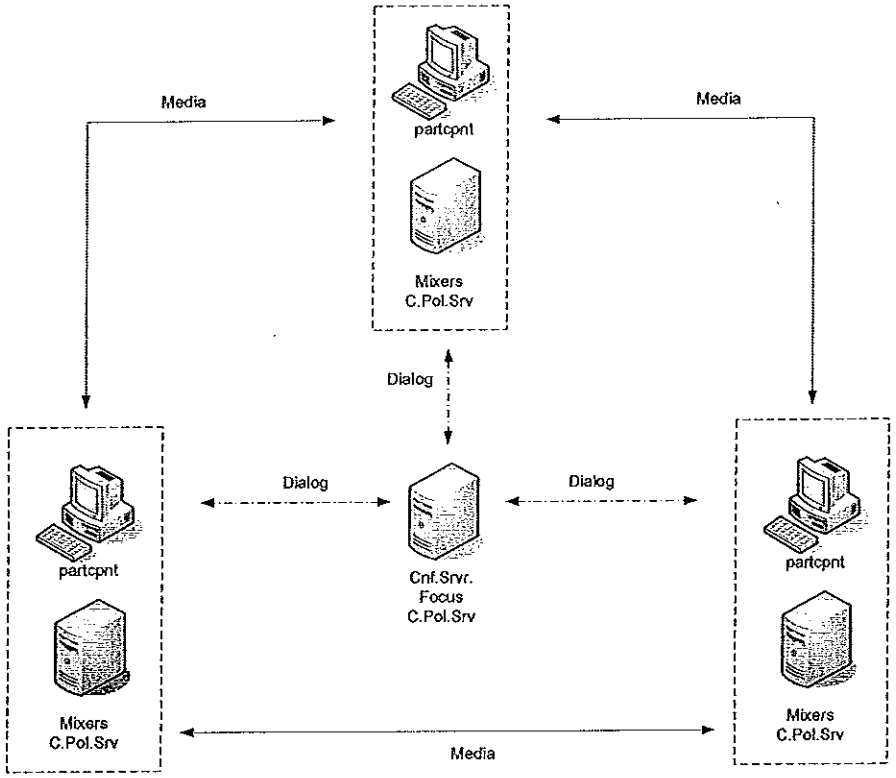


รูปที่ 2-18 Media Server Component [8]

2.4.4 Distributed Mixing

ในรูปแบบนี้จะมีรูปแบบที่คล้ายกับรูปแบบของ Centralized Server โดยมีการนำ Focus Conference Policy และ Media Policy Server เข้ามาใช้งาน แต่จะแตกต่างกันตรงที่จะไม่มีการใช้งานในส่วนของ Centralized Mixers ซึ่งสังเกตว่าส่วนของ Mixer จะถูกนำไปรวมอยู่กับ Endpoint แต่ละตัว ซึ่งการควบคุม Media Stream ต่าง ๆนั้น Focus จะทำการส่งสัญญาณในรูปแบบของ Third party Call Control เพื่อใช้ในการควบคุม โดยการส่งสัญญาณติดต่อระหว่างผู้เข้าร่วมการประชุมกับ Focus จะใช้เพียง SIP Dialog เดียวเท่านั้นแต่จะมีเพิ่มเติมในส่วนของ Session Description เพื่อระบุรายละเอียดของ Media Stream ให้ชัดเจนยิ่งขึ้น

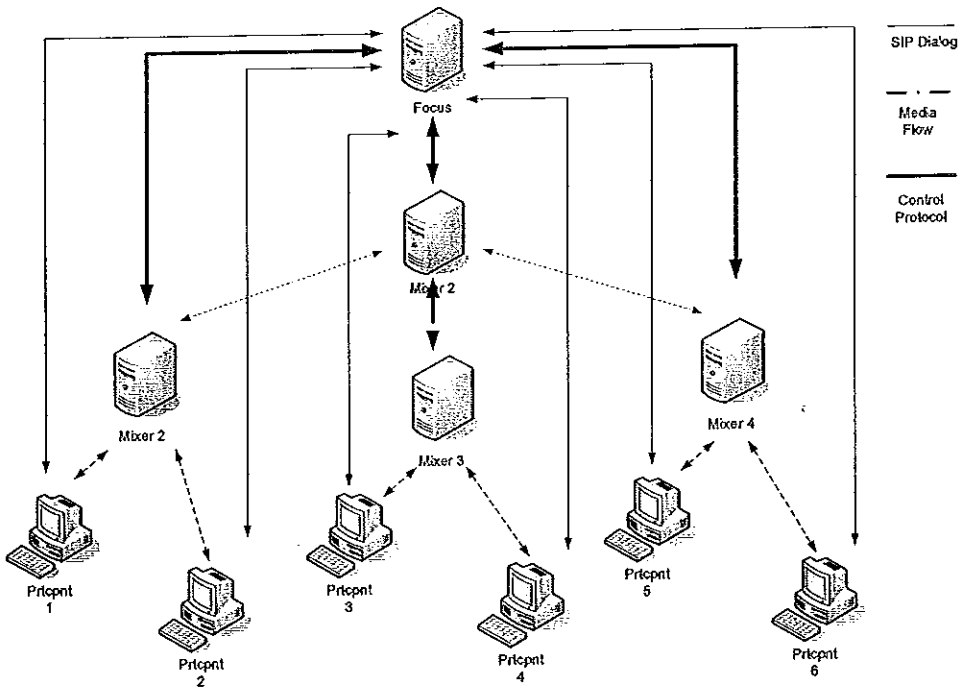
โดยในส่วนของการทำการส่งข้อมูล Media Stream มีหลายวิธีที่จะทำได้เช่นในรูปแบบ Multi-Unicast Model ซึ่งผู้เข้าร่วมการประชุมแต่ละคนจะทำการส่งข้อมูล Media Stream ไปยังผู้เข้าร่วมการประชุมคนอื่น ๆแบบหนึ่งต่อหนึ่ง หรือในแบบ Multicast Model โดยผู้เข้าร่วมการประชุมแต่ละคนจะทำการ Joint เข้าไปยัง Multicast Group จากนั้นก็ทำการส่งข้อมูล Media Stream ของตนเองไปยัง Multicast Group นั้น ๆ โดยรูปแบบการติดต่อจะแสดงในรูปที่ 2-19



รูปที่ 2-19 Distributed Mixing [8]

2.4.5 Cascaded Mixers

ในกรณีที่มีการใช้งานการประชุมที่มีผู้เข้าร่วมการประชุมจำนวนมาก จะเป็นไปได้ยากที่ใช้การรวมข้อมูล Media Stream ทั้งหมดโดยใช้ Mixer เพียงตัวเดียว ดังนั้นจึงได้มีการออกแบบการทำงานที่เป็นแบบ Cascaded Mixers เพื่อให้สามารถใช้ Mixer หลายตัว ๆมาทำงานร่วมกันได้ โดยในรูปแบบนี้จะยังคงมี Focus เป็นส่วนจัดการการติดต่อต่าง ๆเหมือนกับในรูปแบบของ Centralized Server แต่ในส่วนของ Mixing Function จะใช้การทำงานแบบ Multiplicity Mixers กล่าวคือในการ Mixing ข้อมูล Media Stream จะมีการใช้ Mixer มากกว่าหนึ่งตัวในการจัดการ ดังนั้น Focus จำเป็นที่จะมีต้องมีโปรโตคอลบางอย่างเพื่อมาจัดการและควบคุม Mixer เหล่านี้ดังแสดงในรูปที่ 2-20



รูปที่ 2-20 Cascaded Mixers [8]

จากรูปแบบแบบการสร้างการประชุมต่าง ๆที่กล่าวมา สามารถที่จะจำแนกคุณสมบัติต่าง ๆ ของแต่ละรูปแบบออกมาได้ดังตารางที่ 2-1 เพื่อจะใช้ในการกำหนดรูปแบบและความเหมาะสมกับผู้ติดต่อต่าง ๆ

	Location of Focus	Location of Mixer	Number of Servers	Relationship between Focus and Mixer	Scalability
Centralized Server Model	Central conferencing server	Central conferencing server	1	Co-located	Medium
Endpoint Server Model	One of participants	One of participants	0	Co-located	Small
Media Server Component Model	Central conferencing server	Central conferencing server	2	Separated	Medium
Distributed Mixing Model	Central conference server	Every participant	1	Separated	Medium
Cascade Mixers Model	Central conferencing server	Distributed conferencing server	Many	Separated	Large

ตารางที่ 2-1 ตารางเปรียบเทียบความสามารถของสถาปัตยกรรมการประชุมแบบต่าง ๆ [8]

จากตารางที่ 2-1 จะแสดงการเปรียบเทียบให้เห็นคุณลักษณะและความสามารถของสถาปัตยกรรมรูปแบบต่าง ๆ ซึ่งจะพบว่าจำนวนของ Server จะมีผลกับการรองรับผู้ติดต่อ (Scalability) ดังนั้นหากยิ่งมีจำนวน Server มากเท่าใดก็จะยิ่งทำให้รองรับผู้ที่เข้ามาติดต่อมากยิ่งขึ้นและยังสามารถแก้ปัญหา Single point of failure ได้อีกด้วย แต่ในทางกลับกันยิ่งมีจำนวน Server มากเท่าไรการจัดการควบคุมการทำงานของ Server แต่ละตัวก็จะซับซ้อนมากยิ่งขึ้น ซึ่งจะทำให้ขั้นตอนการพัฒนาจะยิ่งยากมากขึ้น นอกจากนี้การที่ใช้ Server ที่มีจำนวนมากจะทำให้ยิ่งสิ้นเปลืองค่าใช้จ่ายมากขึ้น ดังนั้นจึงมีความจำเป็นที่ต้องเลือกรูปแบบการประชุมที่เหมาะกับการใช้งานให้มากที่สุดเพื่อให้คุ้มค่ากับทรัพยากรที่ใช้ไปทั้งหมด

2.5 สรุป

จากทฤษฎีทั้งหมดที่กล่าวมาพบว่าโปรโตคอล SIP มีความสามารถรองรับในการระบบการประชุมแบบสื่อประสมได้หลายรูปแบบ แต่เนื่องจากโปรโตคอล SIP ได้ถูกออกแบบมาเพื่อใช้ในการสร้าง Session ต่าง ๆเท่านั้น แต่ไม่ได้มีความสามารถครอบคลุมถึงการควบสิทธิต่าง ๆในการประชุม รวมถึงการควบคุม Media Stream ต่าง ๆ ดังนั้นในการที่จะสร้างระบบการประชุมให้เกิดขึ้นได้นั้น จึงจำเป็นที่ต้องมีส่วนประกอบอื่น ๆเพิ่มเติมเพื่อช่วยให้ได้ระบบการประชุมที่มีประสิทธิภาพมากยิ่งขึ้น ซึ่งเรื่องนี้เป็นประเด็นหนึ่งที่น่าสนใจในการนำมาศึกษาค้นคว้าเพื่อสร้างระบบการประชุมแบบสื่อประสมที่มีความสามารถรองรับความต้องการของผู้ใช้งานได้อย่างมีประสิทธิภาพมากที่สุด

บทที่ 3
การตรวจเอกสาร

ในบทนี้จะกล่าวถึงบทความต่าง ๆที่เกี่ยวข้องกับการออกแบบระบบการประชุมแบบสื่อประสมซึ่งได้ทำการศึกษาค้นคว้าเพิ่มเติมโดยจะมีประเด็นต่าง ๆที่น่าสนใจ เช่น รูปแบบมาตรฐานที่ใช้ในการควบคุมการประชุม รูปแบบการเชื่อมต่อระหว่างส่วนประกอบต่าง ๆ เป็นต้น เพื่อใช้เป็นแนวทางที่จะนำมาใช้ในการออกแบบและพัฒนาสถาปัตยกรรมการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP

โดยจากเนื้อหาในบทที่ผ่านมาได้นำเสนอถึงองค์ประกอบสำหรับใช้ในระบบการประชุมแบบสื่อสมสำหรับโปรโตคอล SIP และรูปแบบการติดต่อและการสร้างการระบบการประชุมแต่พบว่ายังขาดในส่วนของมาตรฐานและโปรโตคอลที่จะใช้ในการควบคุมระบบการประชุม ซึ่งยังไม่ได้มีนิยามไว้อย่างชัดเจน ดังนั้นจึงเป็นประเด็นในการศึกษาค้นคว้าบทความทางวิชาการต่าง ๆ รวมถึงมาตรฐานต่าง ๆที่มีแนวทางที่จะนำมาประยุกต์ใช้สำหรับใช้ในระบบการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP ซึ่งจะมีบทความและหัวข้อที่น่าสนใจดังต่อไปนี้

3.1 รูปแบบข้อมูลของกฎข้อบังคับที่ใช้ในการควบคุมการประชุม (Conference Policy)

ปัญหาสำคัญอย่างหนึ่งที่ระบบการประชุมต่าง ๆ ยังขาดอยู่คือรูปแบบกฎข้อบังคับที่ใช้ในควบคุมผู้เข้าร่วมการประชุมแต่ละคน ที่จะถูกจัดเก็บไว้โดย Conference Policy ซึ่งจากการศึกษาได้มีการนำเสนอคิดที่จะจัดเก็บข้อมูลเหล่านี้ให้อยู่ในรูปแบบของ XML (Extensible Markup Language) โดยรูปแบบของข้อมูลที่ถูกจัดเก็บไว้ใน Conference Policy [9] จะมีรูปแบบที่เริ่มต้นด้วย Root ซึ่งกำหนดโดย Tag <conference> โดยจะประกอบไปด้วย ส่วนประกอบต่าง ๆ ดังต่อไปนี้

3.1.1 Conference Settings

ส่วนนี้จะกำหนดโดย Tag `<settings>` โดยจะเป็น Element หลักที่จำเป็นต้องมีโดยจะมี 2 sub-Element คือ

- `<conference-uri>` เป็น Element หลักโดยในการประชุมหนึ่งจะมีค่า conference-uri ได้เพียงค่าเดียวเท่านั้น โดยค่า conference-uri นี้จะไม่สามารถที่จะลบหรือแก้ไขได้ขณะที่กำลังทำการประชุมอยู่
- `<max-participant-count>` เป็น optional Element ซึ่งใช้สำหรับกำหนดจำนวนผู้เข้าร่วมการประชุมที่อนุญาตให้เข้าประชุมได้ ในกรณีที่จำนวนผู้เข้าร่วมประชุมมีจำนวนเท่ากับค่าที่กำหนดไว้จะไม่มีผู้ใดสามารถเข้าร่วมการประชุมได้ จนกว่าจะมีการ

ลดค่านี้อีกครั้ง โดยผู้เข้าร่วมประชุมที่เข้ามาทีหลังอาจจะได้รับสัญญาณตอบรับเป็น "480 Temporarily Unavailable" หรืออาจจะให้ทำการรออยู่ใน queue จนกว่าจะ server จะพร้อมให้เข้าประชุม

- <security-level> เป็น optional Element ซึ่งใช้บอกถึงระดับความปลอดภัยที่ใช้ในการรับ-ส่งข้อมูลสำหรับการประชุมนั้น ๆ โดยข้อมูลที่กล่าวถึงนี้จะหมายถึงสัญญาณและข้อมูล media ซึ่งจะมีการกำหนดระดับความปลอดภัยไว้ 4 ระดับคือ
- none ระดับนี้จะไม่มีการกำหนดระดับความปลอดภัยใด ๆทั้งสิ้น
- low ระดับนี้ข้อมูลที่ทำการรับส่งจะต้องมีความถูกต้องสมบูรณ์กันทั้งหมด
- medium ระดับนี้ข้อมูลที่ทำการรับส่งจะต้องเป็นความลับ
- high ระดับนี้ข้อมูลที่ทำการรับส่งจะต้องมีความถูกต้องสมบูรณ์กันทั้งหมดและเป็นความลับ
- <allow-sidebars> เป็น optional Element ในส่วนนี้จะใช้การกำหนดค่าเป็น Boolean เพื่อกำหนดว่าจะอนุญาตให้ทำการสร้าง conference ย่อย (sidebar) ได้หรือไม่
- <sidebar> เป็น Element ที่ใช้สำหรับระบุรายละเอียดของ sidebar ที่จะทำการสร้างขึ้น โดยจะมีพารามิเตอร์หลัก ๆคือ id และ uri ซึ่งใช้สำหรับอ้างอิงถึง sidebar นั้น ๆ

3.1.2 Conference Information

ส่วนนี้จะกำหนดโดยใช้ Tag <info> ซึ่งมีวัตถุประสงค์เพื่อใช้สำหรับอธิบายรายละเอียดของการประชุมที่จัดขึ้น หรืออาจจะนำข้อมูลเหล่าไปใช้สำหรับค้นหาการประชุมต่าง ๆ เป็นต้น โดยจะประกอบไปด้วยข้อมูลต่าง ๆดังนี้

- <subject> ใช้สำหรับระบุหัวข้อที่ใช้ในการประชุมครั้งนี้
- <display-name> ใช้สำหรับระบุชื่อการประชุมในครั้งนี้
- <free-text> ใช้สำหรับระบุรายละเอียดเพิ่มเติมที่เกี่ยวกับการประชุมนี้
- <keywords> ใช้สำหรับระบุ keywords ของการประชุมนี้
- <web-page> ใช้สำหรับระบุ web url ที่มีข้อมูลเพิ่มเติมที่เกี่ยวกับการประชุมนี้
- <host-info> ใช้สำหรับระบุถึงรายละเอียดของ host ที่ผู้ที่จัดการประชุมครั้งนี้
- <e-mail> ใช้ระบุ email ของที่ใช้สำหรับติดต่อสำหรับการประชุมนี้
- <uri> ใช้ระบุ uri ของการประชุมครั้งนี้

สำหรับข้อมูลเหล่านี้อาจจะนำไประบุในส่วนของ SDP ในขั้นตอนการ INVITE เพื่อระบุรายละเอียดของ session ได้อีกด้วย

3.1.3 Conference Time

เป็นข้อมูลที่เกี่ยวข้องกับการกำหนดเวลาต่าง ๆ ที่ใช้ในการประชุม ซึ่งในการประชุมอาจจะสามารถที่จะทำการกำหนดเวลาสำหรับการประชุมได้ โดยข้อมูลเหล่านี้จะระบุอยู่ภายใต้ Tag <time> ซึ่งจะประกอบไปด้วยข้อมูลต่าง ๆดังนี้

- <occurrence> ข้อมูลนี้อาจจะปรากฏได้หลายค่าในหนึ่งการประชุม โดยจะใช้ระบุรายละเอียดของเวลาต่าง ๆที่เกิดขึ้นในการประชุม ซึ่งแต่ละ <occurrence>จะมีข้อมูลรายละเอียดย่อยดังนี้
  - <mixing-start-time> ใช้สำหรับระบุเวลาที่เริ่มทำการ mixing โดยถ้าหากไม่ทำการระบุค่านี้จะหมายความว่าให้ทำการ mixing ข้อมูล media ทันทีที่เริ่มการประชุม
  - <mixing-stop-time> ใช้สำหรับระบุเวลาที่ให้ทำการหยุดการ mixing โดยถ้าหากไม่มีการระบุค่านี้จะหมายความว่าจะหยุดทำการ mixing ข้อมูล media เมื่อจบการประชุม

ในพารามิเตอร์ <mixing-start-time> และ <mixing-stop-time> จะประกอบไปด้วยส่วนหลักที่สำคัญคือ require-participant ซึ่งจะมีการระบุค่าได้ 3 ค่าดังนี้

- none จะหมายความว่าให้เริ่มทำการ mixing หรือหยุดทำการ mixing ได้ทันทีเมื่อถึงเวลาที่กำหนด
- key-participant จะหมายความว่าเมื่อถึงเวลาที่กำหนดแล้วจำเป็นจะต้องมี key-participant เข้าร่วมประชุมอย่างน้อยหนึ่งคนก่อนถึงจะเริ่มทำการ mixing ได้และจะต้องให้ key-participant คนสุดท้ายออกจากการประชุมก่อนจึงจะทำการหยุดทำการ mixing ได้
- participant จะหมายความว่าเมื่อถึงเวลาที่กำหนดแล้วจำเป็นจะต้องมี participant เข้าร่วมประชุมอย่างน้อยหนึ่งคนก่อนถึงจะเริ่มทำการ mixing ได้ และจะต้องให้ participant คนสุดท้ายออกจากการประชุมก่อนจึงจะทำการหยุดทำการ mixing ได้

3.1.4 Conference Dial-Out List

ส่วนนี้ใช้สำหรับกำหนด list ของ uri ที่จะให้ Focus ทำการ INVITE เพื่อให้เข้ามาร่วมการประชุม ซึ่งข้อมูลตรงส่วนนี้ อาจจะกำหนดได้ตั้งแต่เริ่มทำการประชุมหรืออาจจะทำการ update ระหว่างการประชุมก็ได้ซึ่งจะใช้ tag <dialout-list> โดยจะประกอบไปด้วยข้อมูลที่สำคัญดังนี้

- <target> ใช้สำหรับระบุ uri ที่จะทำการ dial-out
- <external> ข้อมูลรายละเอียดเพิ่มเติมอื่น ๆเช่น จำนวนครั้งในการ retransmit เป็นต้น

3.1.5 Conference Refer List

ส่วนนี้ใช้สำหรับกำหนด list ของ uri ที่จะให้ Focus ทำการ REFER เพื่อให้เข้ามาร่วมการประชุม โดยสามารถทำได้หลายรูปแบบเช่นการส่งสัญญาณ REFER หรืออาจจะทำการส่ง SMS หรือ E-mail แทนก็ได้ ซึ่งข้อมูลตรงส่วนนี้ อาจจะกำหนดได้ตั้งแต่เริ่มทำการประชุมหรืออาจจะทำการ update ระหว่างการประชุมก็ได้ซึ่งจะใช้ tag <refer-list> โดยจะประกอบไปด้วยข้อมูลที่สำคัญดังนี้

- <target> ใช้สำหรับระบุ uri ที่จะทำการ REFER
- <external> ข้อมูลรายละเอียดเพิ่มเติมอื่น ๆเช่น จำนวนครั้งในการ retransmit เป็นต้น

3.1.6 Conference Media Streams

รูปแบบการใช้งานจะใช้ tag <media-stream> ข้อมูลตรงส่วนนี้จะเป็นการกำหนดถึงรายละเอียดของ media stream ที่จะใช้สำหรับให้ Focus ทำการระบุในการ INVITE ไปยังผู้เข้าร่วมประชุมต่าง ๆ หรือเพื่อระบุให้ Focus รู้ว่าควรรับ media stream ที่ผู้ใช้ติดต่อมาหรือไม่ แต่จะไม่มีรายละเอียดเกี่ยวกับสิทธิต่าง ๆ เนื่องจากข้อมูลตรงส่วนนั้นจะถูกกำหนดอยู่ใน Media Policy โดยรายละเอียดของ media stream นี้สามารถที่จำกำหนดตั้งแต่ตอนเริ่มทำการประชุมหรืออาจจะกำหนดระหว่างทำการประชุมก็ได้ ซึ่งถ้าหากมีการกำหนดใหม่ระหว่างการประชุมจำเป็นที่จะต้องให้ Focus ทำการ re-INVITE เพื่อ modify session ใหม่อีกครั้ง ซึ่งจะมีรายละเอียดต่าง ๆ ดังนี้

- <audio> ใช้ระบุว่าข้อมูล media-stream ที่ใช้เป็นแบบ audio stream
- <video> ใช้ระบุว่าข้อมูล media-stream ที่ใช้เป็นแบบ video stream
- <text> ใช้ระบุว่าข้อมูล media-stream ที่ใช้เป็นแบบ text mode

3.1.7 Conference Authorization Rule

ส่วนนี้จะเป็นส่วนหนึ่งที่สำคัญอย่างมากในกำหนดค่าของ Conference Policy ซึ่งจะเป็นกำหนดข้อมูลเกี่ยวกับการอนุญาตหรือการให้สิทธิการใช้งานทรัพยากรต่าง ๆในการประชุม รวมถึงสิทธิในการเข้าร่วมการประชุมต่าง ๆ ซึ่งข้อมูลเหล่านี้จะอยู่ภายใต้ tag <ruleset> โดยใน <ruleset> นี้จะประกอบได้ด้วย <rule> ต่าง ๆซึ่งใช้กำหนดสิทธิให้แก่ผู้เข้าร่วมแต่ละคน โดยใน <rule> จะประกอบไปด้วย 3 ส่วนหลัก ๆคือ

- Conditions ใช้ในการตัดสินใจว่าจะให้คำร้องขอเพื่อจะใช้งาน rule นี้ได้หรือไม่
- Actions เป็นการกระทำต่าง ๆ ถ้าหากคำร้องขอใน conditions ได้รับการอนุญาต

- Transformations เป็นการกำหนดสิทธิการทำงานต่าง ๆในการประชุมของผู้ติดต่อ ถ้าหากคำร้องขอ conditions ได้รับการอนุญาต

สำหรับ Rule ต่าง ๆนั้นสามารถที่จะเพิ่มเข้าไปใหม่ได้ตลอดเวลา แต่จะไม่สามารถลบ rule เก่าที่เคยกำหนดไว้ก่อนหน้านี้ได้ แต่อาจจะใช้การแก้ไขแทนได้เพราะบางครั้งอาจจะมีบาง rule ไม่มีความจำเป็นที่ต้องใช้งานแล้ว

ในบางกรณีนี้อาจจะเกิดการกำหนด Rule ซ้ำซ้อนได้เช่นอาจจะทำการ block และ allow ผู้เข้าร่วมประชุมคนเดียวกัน หากเกิดการนี้ขึ้น common rule จะถูกนำมาใช้ซึ่งจะขึ้นอยู่กับสถานการณ์ต่าง ๆด้วย

3.1.8 Conference State Events

แทนด้วย Tag <allow-conference-state> ซึ่งกำหนดค่าเป็น Boolean ถ้ากำหนดค่าเป็น TRUE จะเป็นการกำหนดให้ Focus อนุญาตให้มีการใช้งาน Conference State Events แต่ถ้ากำหนดเป็น FALSE จะเป็นการทำให้ Focus ทำการ reject subscription ทั้งหมดที่ได้รับมา ถ้า Element ไม่ได้ถูกกำหนดไว้จะถือว่ามีค่า default เป็น TRUE

3.1.9 Floor Control Events

แทนด้วย Tag <allow-floor-events> ซึ่งกำหนดค่าเป็น Boolean ถ้ากำหนดค่าเป็น TRUE จะเป็นการกำหนดให้ Focus อนุญาตให้มีการใช้งาน Floor Control Events แต่ถ้ากำหนดเป็น FALSE จะเป็นการทำให้ Focus ทำการ reject subscription ทั้งหมดที่ได้รับมา ถ้า Element ไม่ได้ถูกกำหนดไว้จะถือว่ามีค่า default เป็น FALSE

3.1.10 Conference Join Handling

แทนด้วย Tag <join-handling> ซึ่งจะมีหน้าที่สำหรับกำหนดสิทธิในกาเข้าร่วมการประชุม โดยข้อมูลตรงส่วนนี้จะถูกเรียกใช้งานเมื่อมีผู้ติดต่อต้องการขอเข้าร่วมประชุม ซึ่ง Focus จะการตรวจสอบสิทธิของผู้ติดต่อนั้น ๆ โดยค่าที่กำหนดใน Element นี้จะอยู่ในลักษณะ integer ซึ่งมีอยู่ 3 ค่าคือ

- Block สำหรับ action นี้ Focus จะทำการระงับและยกเลิกการร้องขอของผู้ติดต่อคนนั้นเพื่อเข้ามาในการประชุม ค่านี้กำหนดเป็น 0

- Confirm สำหรับ action นี้ Focus จะกำหนดให้ผู้ติดต่ออยู่ในสถานะ pending เพื่อรอให้ moderator ทำการกำหนดให้เข้าร่วมประชุม ค่านี้จะกำหนดเป็น 1
- Allow สำหรับ action นี้ Focus จะทำการตอบรับการเข้าร่วมการประชุมและกำหนดสิทธิต่าง ๆให้ตามที่ rule กำหนดไว้ ค่านี้กำหนดเป็น 2

3.1.11 Dynamically Referring Users

แทนด้วย tag <allow-referring-users-dynamically> ซึ่งจะกำหนดค่าเป็น Boolean โดยถ้าหากกำหนดค่าเป็น TRUE จะมีความหมายว่าอนุญาตให้สามารถที่จะส่งให้ Focus ทำการ refer user ต่าง ๆเข้ามาในการประชุมโดยไม่จำเป็นต้องแก้ไข refer-list ถ้าหากมีการกำหนดค่าเป็น FALSE จะหมายความว่าให้ Focus จะไม่ทำการตอบรับสัญญาณ REFER ทุกสัญญาณส่งเข้ามาเหตุผลที่ต้องกำหนดเนื่องจากในโปรโตคอล SIP อนุญาตให้ user สามารถที่จะส่งสัญญาณ REFER ไปยัง Focus ได้จากนั้น Focus จะทำการส่งสัญญาณ REFER นั้นต่อไปยัง user ต่าง ๆ ถ้าหาก Element นี้ไม่ได้ถูกกำหนดไว้จะมีค่าเริ่มต้นเป็น FALSE

3.1.12 Dynamically Inviting Users

แทนด้วย tag <allow-invite-users-dynamically> ซึ่งจะกำหนดค่าเป็น Boolean โดยถ้าหากกำหนดค่าเป็น TRUE จะมีความหมายว่าอนุญาตให้สามารถที่จะส่งให้ Focus ทำการ invite user ต่าง ๆเข้ามาในการประชุมโดยไม่จำเป็นต้องแก้ไข dial-out list ถ้าหากมีการกำหนดค่าเป็น FALSE จะหมายความว่าให้ Focus จะไม่ทำการตอบรับสัญญาณ REFER ทุกสัญญาณส่งเข้ามาเหตุผลที่ต้องกำหนดเนื่องจากในโปรโตคอล SIP อนุญาตให้ user สามารถที่จะส่งสัญญาณ REFER ไปยัง Focus ได้จากนั้น Focus จะทำการส่งสัญญาณ INVITE นั้นต่อไปยัง user ต่าง ๆ เพื่อทำการเชิญเข้ามายังการประชุม ถ้าหาก Element นี้ไม่ได้ถูกกำหนดไว้จะมีค่าเริ่มต้นเป็น FALSE

3.1.13 Dynamically Removing Users

แทนด้วย tag <allow-remove-users-dynamically> ซึ่งจะกำหนดค่าเป็น Boolean โดยถ้าหากกำหนดค่าเป็น TRUE จะมีความหมายว่าอนุญาตให้สามารถที่จะส่งให้ Focus ทำการ invite user ต่าง ๆเข้ามาในการประชุมโดยไม่จำเป็นต้องแก้ไข ruleset ถ้าหากมีการกำหนดค่าเป็น FALSE จะหมายความว่าให้ Focus จะไม่ทำการตอบรับสัญญาณ REFER ทุกสัญญาณส่งเข้ามาเหตุผลที่ต้องกำหนดเนื่องจากในโปรโตคอล SIP อนุญาตให้ user สามารถที่จะส่งสัญญาณ

REFER ไปยัง Focus ได้จากนั้น Focus จะทำการส่งสัญญาณ BYE นั้นต่อไปยัง user ต่าง ๆเพื่อทำการสั่งให้ user นั้น ๆออกจากการประชุม ถ้าหาก Element นี้ไม่ได้ถูกกำหนดไว้จะมีค่าเริ่มต้นเป็น FALSE

3.1.14 Floor Request Handling

แทนด้วย Tag <floor-request-handling> ซึ่งจะเป็นการกำหนดการทำงานให้ Focus เพื่อใช้ในควบคุม Floor Requests ที่ผู้เข้าประชุมต่าง ๆร้องขอเข้ามา โดยจะค่าที่กำหนดจะอยู่รูปแบบของ integer

3.1.15 Key Participant

แทนด้วย Tag <is-key-participant> ซึ่งจะมีรูปแบบเป็น Boolean โดยถ้าหากมีการกำหนดค่าเป็น TRUE จะหมายความว่าผู้ติดต่อที่เข้ามาภายใต้ rule นี้จะถูกกำหนดให้เป็น key participant ถ้ากำหนดค่าเป็น FALSE จะหมายความว่าผู้ติดต่อจะไม่ถูกกำหนดเป็น key participant

ถ้าหากค่าของ Element ไม่ได้ถูกกำหนดไว้จะหมายความว่าได้กำหนดค่าของ Element เป็น FALSE

3.1.16 Floor Moderator

แทนด้วย Tag <is-floor-moderator> ซึ่งจะมีการกำหนดรูปแบบเป็น Boolean ถ้าหากกำหนดค่าเป็น TRUE จะหมายความว่าผู้ติดต่อที่เข้ามาภายใต้ rule นี้จะถูกกำหนดให้มีความสามารถเป็น floor moderator ซึ่งมีสิทธิในการควบคุม floor ต่าง ๆที่เกิดขึ้นใน conference ถ้าหากกำหนดเป็น FALSE จะหมายความว่าผู้ติดต่อจะไม่ถูกกำหนดเป็น floor moderator

ถ้าหากค่าของ Element ไม่ได้ถูกกำหนดไว้จะหมายความว่าได้กำหนดค่าของ Element เป็น FALSE

3.1.17 Conference Information

แทนด้วย Tag <show-conference-info> ซึ่งจะกำหนดรูปแบบเป็นแบบ Boolean ถ้าหากกำหนดค่าเป็น TRUE จะหมายความว่าข้อมูลรายละเอียดของ conference จะถูกแสดงไปยังผู้ที่

ติดต่อเข้ามายังการประชุม แต่ถ้าหากกำหนดค่าเป็น FALSE จะหมายความว่าข้อมูลรายละเอียดของ conference จะไม่ถูกแสดงให้ผู้เข้าร่วมการประชุมรับทราบ

ถ้าหากค่าของ Element ไม่ได้ถูกกำหนดไว้จะหมายความว่าได้กำหนดค่าของ Element เป็น FALSE

3.1.18 Floor Holder

แทนด้วย Tag <show-floor-holder> ซึ่งจะกำหนดรูปแบบของข้อมูลเป็นแบบ Boolean ถ้าหากกำหนดค่าเป็น TRUE จะหมายความว่าผู้เข้าร่วมการประชุมสามารถที่จะทราบได้ว่าขณะนั้นผู้ใดได้ใช้งาน floor อยู่ ถ้าหากกำหนดค่าเป็น FALSE จะความหมายว่าผู้เข้าร่วมการประชุมจะไม่สามารถที่จะทราบว่าขณะนั้นผู้ใดได้ใช้งาน floor อยู่

ถ้าหากค่าของ Element ไม่ได้ถูกกำหนดไว้จะหมายความว่าได้กำหนดค่าของ Element เป็น FALSE

3.1.19 Floor Request

แทนด้วย Tag <show-floor-requests> ซึ่งจะกำหนดรูปแบบของข้อมูลเป็นแบบ Boolean ถ้าหากกำหนดค่าเป็น TRUE จะหมายความว่าผู้เข้าร่วมการประชุมสามารถที่จะทราบได้ว่าขณะนั้นผู้ใดได้ทำการร้องขอใช้งาน floor อยู่ ถ้าหากกำหนดค่าเป็น FALSE จะความหมายว่าผู้เข้าร่วมการประชุมจะไม่สามารถที่จะทราบว่าขณะนั้นผู้ใดได้ทำการร้องขอใช้งาน floor อยู่

ถ้าหากค่าของ Element ไม่ได้ถูกกำหนดไว้จะหมายความว่าได้กำหนดค่าของ Element เป็น FALSE

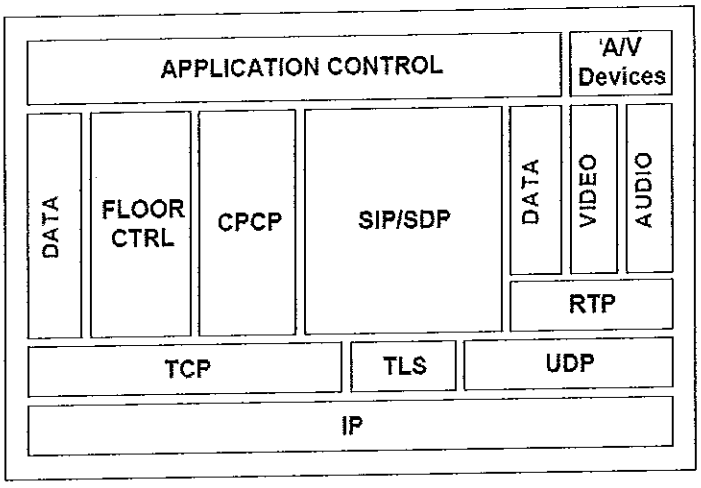
จากทั้งหมดที่กล่าวมาเป็นรูปแบบของกฎข้อบังคับต่าง ๆ ที่ได้ทำการศึกษาเพื่อจะนำมาใช้ร่วมกับระบบการประชุมที่ได้ทำการออกแบบไว้ โดยเนื่องจากรูปแบบข้อมูลที่จัดเก็บจะอยู่ในรูปแบบของ XML และข้อมูลเหล่านี้จะถูกจัดเก็บไว้ที่ Conference Policy Server ซึ่งในการใช้งานจริงนั้นผู้ควบคุมระบบจะต้องทำการกำหนดค่าของ Conference Policy เหล่านี้ได้ตลอดเวลา และค่าของ Confererence Policy ของแต่ละการประชุมก็จะต่างการกำหนดที่แตกต่างกัน ดังนั้นจึงมีความเป็นที่จะต้องมีโปรโตคอลเข้ามาควบคุมการทำงานตรงส่วนนี้ ซึ่งในหัวข้อต่อไปจะกล่าวถึงการศึกษาแนวคิดและโปรโตคอลที่จะใช้ในการควบคุม

3.2 ความสามารถในการจัดการข้อมูลผู้ใช้งาน (User Management)

สำหรับความต้องการในส่วนของ User Management สำหรับ CPCP (Conference Policy Control Protocol) มีข้อกำหนดขั้นพื้นฐานดังนี้ [9] มีดังนี้

- CPCP จะต้องให้ผู้ติดต่อคนใดคนหนึ่งมีความสามารถในการเข้าถึง Conference Policy
- CPCP จะต้องสามารถที่จะยกเลิกผู้ติดต่อคนใดคนหนึ่งในการเข้าถึง Conference Policy
- CPCP จะต้องสามารถที่จะกำหนดให้ผู้ติดต่อใด ๆสามารถที่จะ Subscribe ไปยัง Conference Event Package ได้

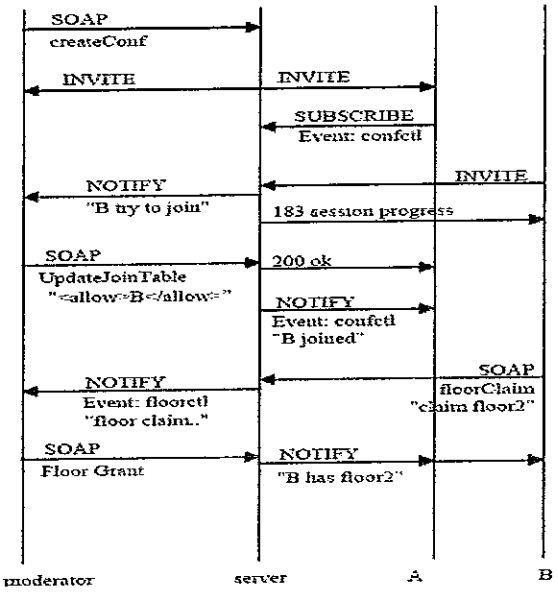
จากข้อมูลข้างต้นเราจะพบว่า Conference Policy Control Protocol จะมีการงานทำงานอยู่ในระดับชั้นเดียวกับโปรโตคอล SIP ดังแสดงในรูปที่ 3-1 ซึ่งอยู่ในชั้น Application Layer โดยเป็นระดับชั้นที่มีการทำงานในการควบคุมการติดต่อต่าง ๆ ของระบบ ซึ่งส่วนของ CPCP จะทำหน้าที่ในการควบคุมการทำงานต่าง ๆที่เกิดขึ้นในการประชุมร่วมถึงการกำหนดกฎข้อบังคับและข้อกำหนดต่าง ๆที่จะใช้ในการประชุม โดยรูปแบบของ CPCP ได้ถูกกำหนดให้อยู่ในรูปแบบ XML เผื่อความสะดวกในการทำความเข้าใจและยังรูปแบบข้อมูลที่ใช้กันอย่างแพร่หลายในปัจจุบัน



รูปที่ 3-1 Protocol stack of an end system in the SIP Conferencing Architecture[10]

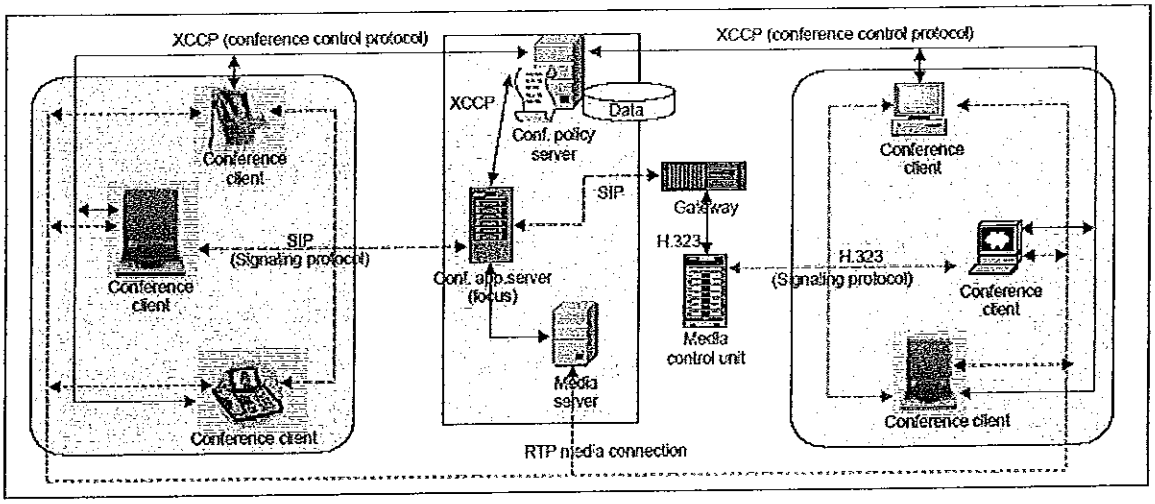
ทำให้มีแนวคิดที่นำเสนอรูปแบบของ Conference Policy Control Protocol ออกมาโดยใช้โปรโตคอล HTTP ในการส่งข้อมูลที่เป็นแบบ XML หรือเรียกอีกอย่างหนึ่งว่า SOAP (Simple Object Access Protocol) [11] ตัวอย่างการใช้งาน SOAP เพื่อทำการควบคุมการประชุมแสดงในรูปที่ 3-2 ซึ่งจากรูปจะสังเกตได้ว่าส่วนการทำงานของ SOAP จะถูกเรียกใช้งานตั้งแต่ขั้นตอนเริ่ม

สร้างการประชุม เพื่อใช้ในการกำหนดค่าเริ่มต้นสำหรับการประชุมอาทิเช่น จำนวนผู้เข้าร่วมการประชุม รูปแบบข้อมูลสื่อประสมที่ใช้ในการประชุม เป็นต้น หลังจากนั้นเมื่อการประชุมถูกเริ่มต้นขึ้นมาแล้ว SOAP อาจจะถูกเรียกใช้อีกครั้งระหว่างที่ทำการประชุม เพื่อเปลี่ยนแปลงค่าต่างๆ เช่น จากรูปตัวอย่างจะเป็นการกำหนดสิทธิอนุญาตให้ผู้ติดต่อ B สามารถเข้าร่วมการประชุมได้ เป็นต้น



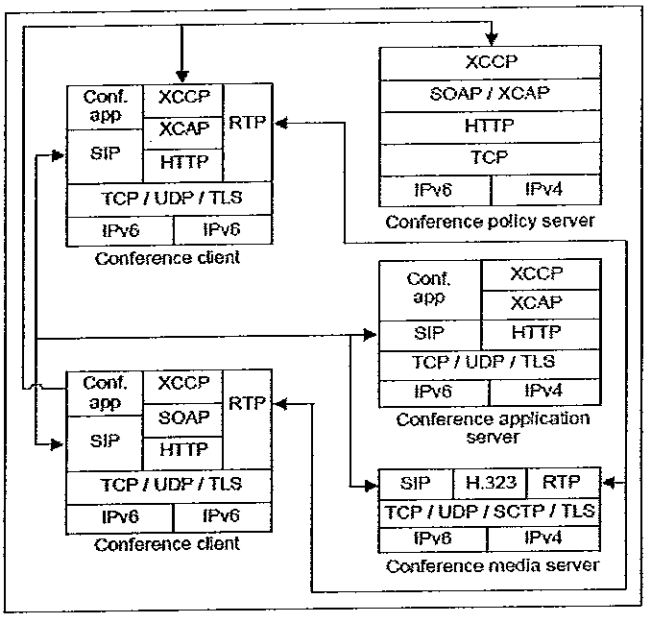
รูปที่ 3-2 ตัวอย่างการใช้งาน SOAP เพื่อทำการควบคุมการประชุม [11]

จะพบว่า SOAP เป็นรูปแบบหนึ่งที่ใช้ในการส่งข้อมูล Conference Policy Control Protocol แต่อาจจะยังไม่ใช่หนทางที่ดีที่สุดสำหรับการทำงานตรงส่วนนี้ ทำให้มีแนวคิดที่นำเสนอรูปแบบเพิ่มเติมขึ้นมาโดยนำเสนอในชื่อ XCCP (XML-based Conference Control Protocol) [12] โดยมีรูปแบบการทำงานดังแสดงในรูปที่ 3-3 โดยจะสังเกตได้ว่า XCCP จะเป็นส่วนที่เชื่อมต่อระหว่างผู้เข้าร่วมการประชุมกับ Conference PolicyServer เพื่อใช้ในการกำหนดค่าต่างๆ ซึ่งค่าที่ถูกกำหนดจะถูกส่งต่อไปยัง Focus เพื่อให้นำค่าที่กำหนดไปทำการควบคุมผู้เข้าร่วมการประชุมคนอื่นๆต่อไป

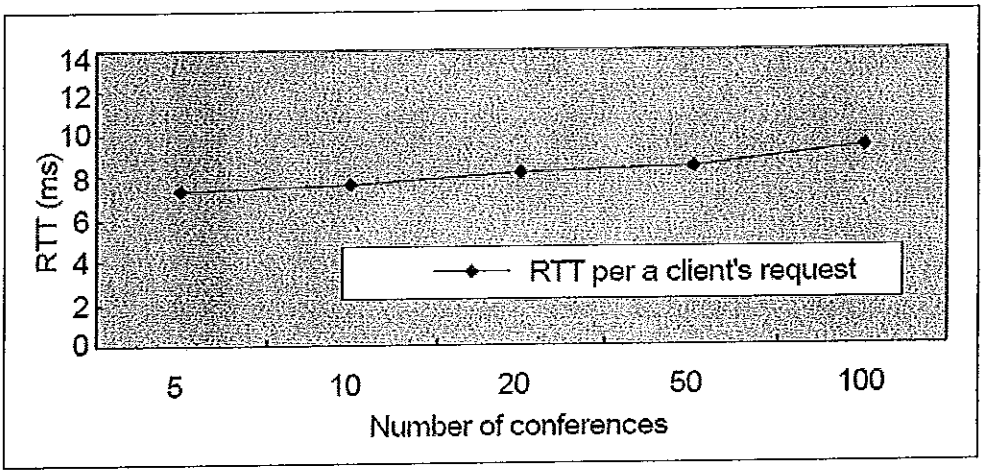


รูปที่ 3-3 โครงสร้างของระบบ XCCP [12]

โดยการทำงานของ XCCP จะอ้างอิงมาจาก XCON-WG [13] เป็นการผสมผสานระหว่าง SOAP และ XCAP (The Extensible Markup Language Configuration Access Protocol) [14] ซึ่งข้อดีของการทำงานบน XCAP คือ XCAP จะทำงานอยู่ HTTP โดยตรงดังนั้นจึงเป็นการง่ายที่จะนำมาผสมผสานการทำงานร่วมกับ SOAP ได้ซึ่งสามารถแสดงในรูปที่ 3-4 ซึ่งจะสังเกตได้ว่า XCCP จะถูกนำมาใช้งานกับ Component ที่ใช้สำหรับติดต่อการประชุมเท่านั้น แต่จะไม่ถูกนำไปใช้งานในส่วนของการควบคุมสื่อประสม



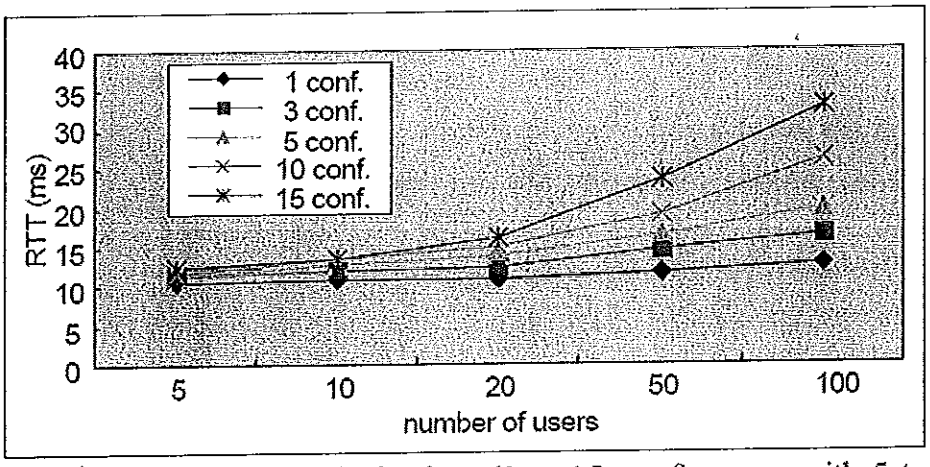
รูปที่ 3-4 XCCP protocol stack [12]



Simulated round trip time (5 to 100 conferences).

รูปที่ 3-5 ผลจากการทดลองใช้ XCCP ในระบบการประชุม [12]

จากรูปที่ 3-5 จะเป็นผลการประเมินคุณภาพในประเด็นของการรองรับการขยายตัวของระบบ (Scalability) ซึ่งจะเป็นการทดลองโดยใช้ XCCP ในขั้นตอนของการสร้างการประชุม (Create Conference) ซึ่งจะเห็นได้ว่าถึงแม้จำนวนของกลุ่มการประชุมจะเพิ่มมากขึ้น ค่าของ Round trip time (RTT) ก็ยังมีค่าที่ใกล้เคียงกัน



Simulated round trip time (1 to 15 conferences with 5 to 100 users in each).

รูปที่ 3-6 ผลจากการทดลองใช้ XCCP ในระบบการประชุม [12]

จากรูปที่ 3-6 จะเป็นการทดลองในประเด็นในการเปรียบเทียบในกรณีของจำนวนของผู้ติดต่อกับจำนวนของการประชุมว่ามีผลอย่างไรหากนำ XCCP เข้าไปใช้ ซึ่งจะพบว่าปริมาณของจำนวนการประชุมจะมีผลกระทบกับค่า RTT น้อยกว่าการที่มีผู้ติดต่อเข้ามาในการประชุม ซึ่ง

สามารถสรุปได้ว่า XCCP เป็นโปรโตคอลที่สามารถรองรับการประชุมได้ตั้งแต่ขนาดเล็กจนถึงขนาดกลาง

ซึ่งจากการทดลองข้างต้นจะพบว่าในการทดลองไม่ได้กล่าวถึงในประเด็นของกำลังที่ใช้ในการประมวลผล (CPU Computing Power) หรือความสามารถในการประมวลผลข้อมูลเลย ซึ่งประเด็นนี้ก็เป็นเรื่องสำคัญอีกเรื่องหนึ่งที่ควรจะนำมาทดลอง หากต้องการที่จะนำไปใช้งานในระบบ Centralized Conference

จากบทความทั้งสองบทความข้างต้นจะพบว่ามีหลักการทำงานที่คล้ายคลึงกันเป็นอย่างมาก กล่าวคือ ทำงานอยู่บนหลักการเดียวกันโดยรูปแบบของข้อมูลที่เป็นแบบ XML และทำงานอยู่บน SOAP เหมือนกันซึ่งมีผลดีผลเสียดังนี้

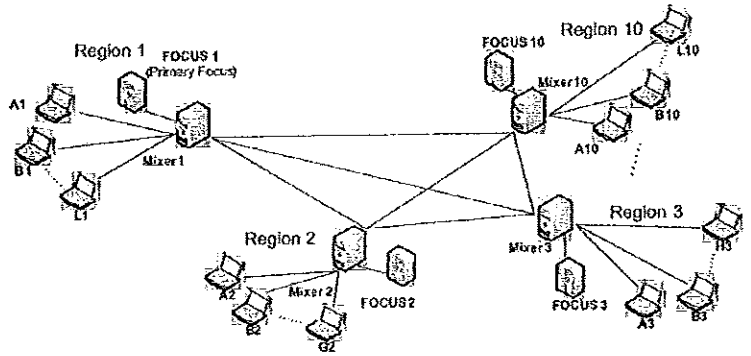
ผลดีของการใช้ SOAP และ XML ในการส่งข้อมูล

- เป็นมาตรฐานที่ได้รับการยอมรับ
- ทำงานอยู่บน HTTP ซึ่งเป็นโปรโตคอลที่ได้ความนิยมอย่างมาก
- โครงสร้างของข้อมูลสามารถเข้าใจได้ง่าย

ผลเสียของการใช้ SOAP และ XML ในการส่งข้อมูล

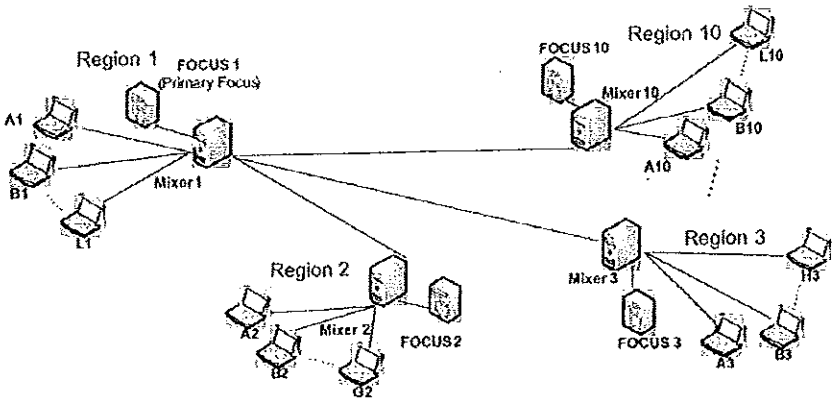
- มี Overhead ในการส่งข้อมูลค่อนข้างสูง
- กำลังที่ใช้ในการประมวลผลสูงในการส่งข้อมูล
- มีความซับซ้อนสูงหากจะนำไปประยุกต์ใช้

ได้มีการศึกษาถึงรูปแบบของการกำหนด Policy สำหรับการทำการประชุมที่เป็นแบบ Distributed Management Architecture [15] สำหรับประเด็นที่น่าสนใจคือในส่วนของการจัดการ Policy ต่างๆเพื่อให้รองรับการทำงานของ Focus และ Mixer ที่มีมากกว่าหนึ่งตัวเพื่อรองรับระบบที่มีขนาดให้ใหญ่ซึ่งจะแตกต่างจากเดิมที่อยู่ในรูปแบบของ Centralized โดยรูปแบบของการติดต่อแบบ Distributed นั้นสามารถแบ่งออกได้เป็นอีก 2 รูปแบบคือ แบบ Full-mesh และ Tree ดังแสดงในรูปที่ 3-7 และรูปที่ 3-8



รูปที่ 3-7 Distributed conferencing (full-mesh) [15]

ในรูปที่ 3-7 จะพบว่ารูปแบบการติดต่อนั้นจะยังคงต้องใช้ Focus หลักในการทำงานอยู่ 1 ตัวซึ่งเรียกว่า Primary Focus ส่วน Focus ตัวอื่น ๆจะถูกเรียกว่า Regional Focus ซึ่งจะคอยรับคำสั่งจาก Primary Focus อีกทีหนึ่งเพื่อใช้ในการควบคุมการทำงานต่าง ๆ โดยรูปแบบการติดต่อของ Focus แต่ละตัวจะเป็นแบบ Centralized ซึ่งหมายความว่า Regional Focus ทุกตัวจะติดต่อผ่านไปยัง Primary Focus เท่านั้นเพื่อรับข้อมูลของผู้ติดต่อต่าง ๆ แต่ในส่วนของ Mixer นั้นจะรูปแบบการติดต่อแบบ Full-mesh ซึ่งจะมีข้อดีในเรื่องของการลดค่าเวลาหน่วงของ Media Stream แต่ก็จะมีข้อเสียในส่วนของสิ้นเปลือง Bandwidth และในส่วนของการควบคุมที่จะไม่การควบคุมการทำงานของ Mixer ใด ๆทั้งสิ้น



รูปที่ 3-8 Distributed conferencing (Tree) [15]

ในรูปที่ 3-8 รูปแบบของการเชื่อมต่อของ Focus นั้นจะมีรูปแบบเหมือนกับในรูปที่ 3-7 แต่จะแตกต่างกันตรงในส่วนของการเชื่อมต่อ Mixer ที่เป็นแบบ Tree base โดยมี Mixer หลักหนึ่งตัวที่จะคอยทำหน้าจัดการและควบคุมการทำงานของ Mixer ตัวอื่น ๆที่อยู่ภายใต้ตัวมัน ซึ่งเมื่อดูถึงในส่วนของข้อดีและข้อเสียแล้วนั้นจะพบว่าการเชื่อมต่อแบบ Tree base จะมีค่าเวลาหน่วงของ Media Stream สูงกว่ารูปแบบของ Full-mesh แต่จะดีกว่าตรงในส่วนของการจัดการและควบคุมการทำงานของ Mixer ได้ดีกว่ารวมไปถึงยังสามารถลดในส่วนของ Bandwidth ได้อีกด้วย

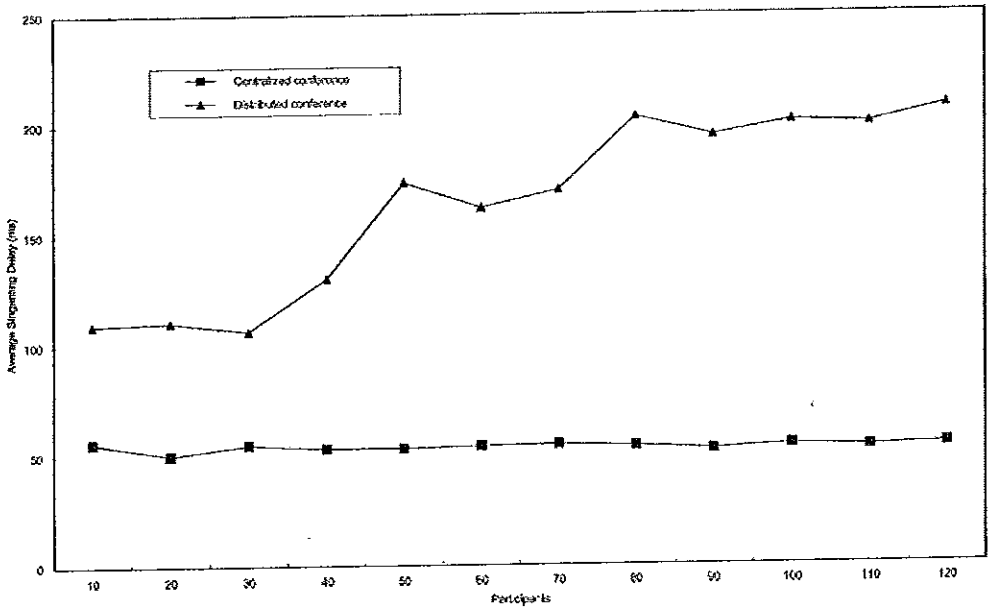
เมื่อพิจารณาในส่วนของการจัดการ Policy พบว่าได้ใช้งานมาตรฐาน CPCP ที่กำหนดโดย XCON แต่จะมีการเพิ่มเติม Policy เข้าไปอีกเพื่อใช้การจัดการการทำงานของ Mixer และ Regional Focus โดยได้เพิ่มหัวข้อ Policy ดังนี้

- Participant Management จะใช้ในการจัดการในส่วนของ Membership Policy การให้สิทธิต่าง ๆของ user ที่จะเข้ามาใช้งานในการประชุม
- Mixer Management จะใช้ในการจัดการทำงานของ Mixer เช่น กำหนดจำนวนของผู้ติดต่อที่สามารถเข้าใช้งานใน Mixer เป็นต้น

- Regional Focus Management จะใช้ในการจัดการทำงานของ Regional เช่น กำหนดจำนวนของผู้ติดต่อที่สามารถเข้าใช้งานใน Focus แต่ละตัว เป็นต้น

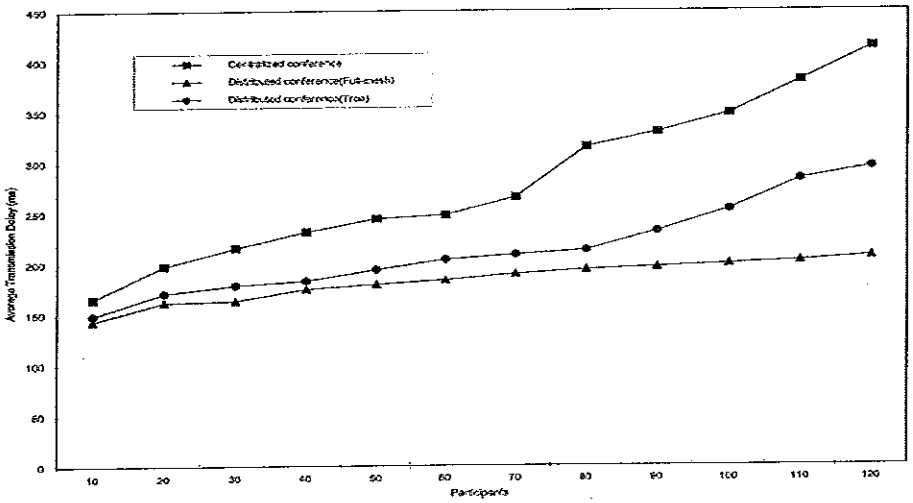
ในส่วนของติดต่อเพื่อใช้งานการประชุม ในขั้นตอนเริ่มต้นผู้ติดต่อทุกคนจะต้องติดต่อไปยัง Primary Focus ก่อนทุกครั้งเพื่อทำการยืนยันตัวตน จากนั้น Primary Focus จะทำการตรวจสอบค่า Policy ก่อนที่จะเลือกว่าจะให้ผู้ติดต่อนั้นไปเชื่อมต่อกับ Mixer และ Regional Focus ตัวใดอีกทีหนึ่ง

ดังนั้นหากพิจารณาถึงข้อดีและข้อเสียของวิธีการข้างต้นจะพบข้อดีในส่วนของการจัดการการประชุมที่มีขนาดใหญ่ (Large scale) ที่ระบบสามารถรองรับการติดต่อได้ค่อนข้างมากพอสมควรโดยที่ค่าเวลาหน่วงและ Processing load ไม่ได้สูงขึ้นมากเมื่อเปรียบเทียบกับระบบที่เป็นแบบ Centralized ดังแสดงในรูปที่ 3-9 และรูปที่ 3-10



รูปที่ 3-9 Average signaling delay for invitation [15]

โดยในรูปที่ 3-9 จะเป็นกราฟที่แสดงค่าเวลาหน่วง ของสัญญาณ SIP ใช้ในการเข้าร่วมการประชุม (Join Conference) ของผู้ติดต่อที่ส่งไปยัง Focus โดยการวัดค่าจะวัดตั้งแต่เริ่มส่งสัญญาณ INVITE จนถึงได้รับสัญญาณ 200 OK ซึ่งเมื่อสังเกตจะพบว่าในระบบที่เป็นแบบ Centralized จะค่าเวลาหน่วง เพิ่มขึ้นตามจำนวนของผู้ติดต่อที่เพิ่มขึ้นแต่ในส่วนที่เป็นแบบ Distributed ค่าเวลาหน่วงกลับไม่เพิ่มเท่าที่ควร



รูปที่ 3-10 Average stream transmission delay [15]

สำหรับในรูปที่ 3-10 จะเป็นกราฟที่แสดงถึงค่าเวลาหน่วงของ Media Stream ที่ถูกส่งระหว่าง Mixer แต่ละตัวก็จะพบว่าผลลัพธ์ที่ได้ในแบบ Centralized ยังมีค่าเวลาหน่วง ที่ค่อนข้างสูงเมื่อจำนวนผู้ติดต่อสูงขึ้นแต่ในส่วนของ Distributed ที่เป็นแบบ Full-mesh จะมีอัตราการหน่วงเวลาที่น้อยมากเนื่องจาก Media Stream จะถูกส่งถึงกันโดยตรงโดยปราศจากการควบคุมใด ๆ แต่ส่วนของรูปแบบ Tree base นั้นจะมีค่าเวลาหน่วง เพิ่มขึ้นมาเล็กน้อย โดยค่าเวลาหน่วงของการประชุมที่เชื่อมต่อแบบ Tree base นั้นจะขึ้นกับความลึกของจำนวนโหนดที่เชื่อมต่อ

จากนั้นเมื่อมาสังเกตถึงข้อเสียของแนวคิดนี้พบว่ามีกำหนดรายละเอียดของการติดต่อสื่อสารเข้าไปใหม่เพิ่มเติมหลายส่วน รวมถึงรูปแบบของสัญญาณใหม่ ซึ่งอาจจะทำให้ระบบที่ออกแบบมาอาจจะไม่สามารถใช้งานร่วมกับ SIP User Agent ที่มีอยู่ในปัจจุบันได้ อีกส่วนหนึ่งคือในส่วนของรูปแบบของช่องทางการติดต่อที่ใช้การกำหนดค่า Policy ต่าง ๆซึ่งไม่ได้ระบุไว้อย่างชัดเจนว่าจะนำมาตรฐานใด ๆ มาใช้ในการกำหนดค่าเหล่านี้

จากปัญหาดังกล่าวข้างต้นจึงน่าจะเป็นประเด็นที่สามารถไปทำการวิจัยเพื่อสรุปและหารูปแบบมาตรฐานของช่องทางการติดต่อที่ใช้ในการกำหนด Policy ต่าง ๆเพื่อใช้ในการทำ Multimedia Conference

แต่จากศึกษาเพิ่มเติมพบว่า ณ ปัจจุบันทาง XCON Workgroup ได้ทำการยกเลิกการใช้งานโปรโตคอล CPCP แต่ได้ทำการกำหนดรูปแบบของข้อมูลขึ้นมาใหม่โดยใช้ชื่อว่า “Common Conference Information Data Model” [16] ซึ่งยังคงมีรูปแบบของข้อมูลที่เป็นแบบ XML เช่นเดิม แต่จะมีการเพิ่มรายละเอียดต่าง ๆเพิ่มเติมมากกว่าโปรโตคอล CPCP ของเดิม โดยจะเพิ่มในส่วนของการกำหนดรายละเอียดของ Mixer และมีการนำข้อมูลบางส่วนของ SIP Event Package [17][18] เข้ามารวมเพื่อใช้ในการรายงานสถานะต่าง ๆของผู้เข้าร่วมการประชุม

ซึ่งจะพบว่าถึงแม้ทาง XCON Workgroup จะได้ทำการยกเลิกโปรโตคอล CPCP ไปแต่ก็ได้ทำการกำหนด Common Conference Information Data Model ขึ้นมาแทน ซึ่งยังคงมีรายละเอียดหลาย ๆส่วนที่มีความเหมือนและใกล้เคียงกับโปรโตคอล CPCP เดิม ดังนั้นรูปแบบของการจัดการกำหนดหรือแก้รายละเอียดข้อมูล Policy ต่าง ๆนั้นจึงยังรูปแบบที่คล้ายคลึงกัน ซึ่งจากการศึกษาพบว่ามีเทคนิคและโปรโตคอลที่น่าสนใจซึ่งมีชื่อว่า "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)" โดยมีรายละเอียดในหัวข้อต่อไป

3.3 The Extensible Markup Language Configuration Access Protocol (XCAP)

สำหรับในหัวข้อนี้จะกล่าวถึงรายละเอียดของโปรโตคอล XCAP ซึ่งกล่าวถึงคุณสมบัติ รูปแบบการทำงานและความเหมาะสมต่าง ๆที่จะนำมาใช้งานร่วมกับรูปแบบการกำหนดข้อมูล Policy ต่าง ๆในระบบการประชุม

3.3.1 คุณสมบัติของโปรโตคอล XCAP

สำหรับโปรโตคอล XCAP นั้นได้มีการออกแบบและพัฒนาโดยนำแนวคิดของรูปแบบเข้าถึงข้อมูลตามแบบโปรโตคอล HTTP ซึ่งโปรโตคอล XCAP จะมีการใช้งานเมธอด (Method) ต่าง ๆของโปรโตคอล HTTP เพื่อใช้ในการแก้ไขข้อมูลต่าง ๆที่อยู่ในรูปแบบของ XML ซึ่งจะมีเมธอดที่ใช้งานหลักอยู่ 3 เมธอดคือ

- HTTP-GET ใช้สำหรับดึงข้อมูล Document Element หรือ Attribute ต่าง ๆที่อยู่บนฝั่ง Server
- HTTP-PUT ใช้สำหรับเพิ่มเติมหรือแก้ไขข้อมูล Document Element หรือ Attribute ต่าง ๆที่อยู่บนฝั่ง Server
- HTTP-DELETE ใช้สำหรับลบข้อมูล Document Element หรือ Attribute ต่าง ๆที่อยู่บนฝั่ง Server

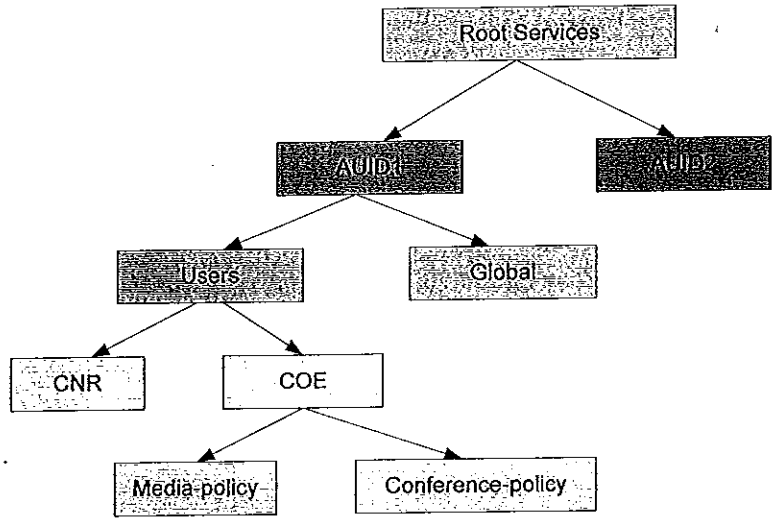
นอกจากนี้โปรโตคอลได้นำเทคโนโลยีที่เรียกว่า "XPath" ซึ่งเป็นเทคนิคที่ใช้สำหรับทำ XML Addressing กล่าวคือจะทำการแปลง Element หรือ Attribute ของเอกสาร XML ให้อยู่ในรูปแบบของ URI ตัวอย่างเช่น

```
<?xml version="1.0" encoding="UTF-8"?
xmlns:post="http://www.post.com"
xmlns:sync=http://www.sync.com
xmlns="http://www.post.com">
<address-book>
<!—This guy is a bozo --
<entry>
<name>Jonathan R<name>
```

```
<email>jr@dsoft.com</email>
<street paved="true">600 Lx Pl</street>
<city>Parsippany</city>
<state>NJ</state>
<country>USA</country>
<ietf-participant/>
<sync:state>synchronized</sync:state>
</entry>
</address-book>
```

รูปที่ 3-11 ตัวอย่างเอกสาร XML

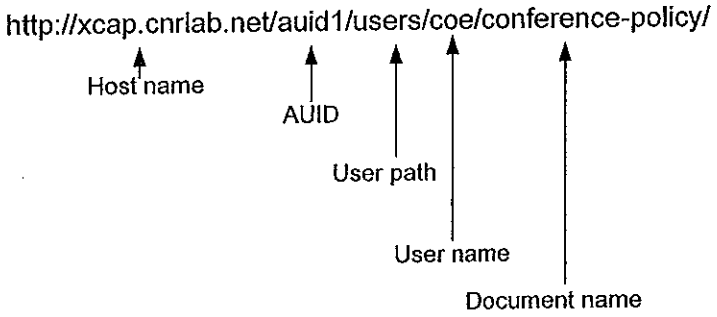
จากตัวอย่างเอกสาร XML ในรูปที่ 3-11หากผู้ใช้งานต้องการข้อมูลที่อยู่ใน Element ผู้ใช้งานสามารถที่จะอ้างถึงได้โดยกำหนด Path เป็นดังนี้ address-book/entry/name เป็นต้น สำหรับในโปรโตคอล XCAP นั้นได้มีการเพิ่มเติมส่วนของ Application Unique ID (AUID) เพื่อใช้ในการแยกประเภทของเอกสาร XML ของแต่ละ Application ออกจากกัน ซึ่งแต่ละ AUID จะประกอบด้วยส่วนของ User ที่มีสิทธิใช้งานต่าง ๆอีกชั้นหนึ่งดังแสดงในรูปที่ 3-12



รูปที่ 3-12 โครงสร้างการอ้างถึงข้อมูของโปรโตคอล XCAP

ซึ่งจะมีรูปแบบการใช้งานดังรูปที่ 3-13 โดยจะประกอบไปด้วยส่วนต่าง ๆ คือ

- Host name เป็นการระบุถึงชื่อเครื่องแม่ข่ายที่ใช้เก็บเอกสาร XML
- AUID เป็นการระบุถึงบริการที่จะใช้การเข้าถึงเอกสาร XML
- User Path เป็นส่วนที่ใช้ระบุระดับในการเข้าถึงเอกสาร โดยแบ่งออกได้เป็นสองแบบคือ ถ้ากำหนดเป็น users จะหมายถึงในการเข้าถึงเอกสาร XML จำเป็นต้องมีการระบุชื่อผู้ใช้งาน แต่ถ้าหากกำหนดเป็นแบบ global จะไม่ต้องระบุชื่อผู้ใช้ในการเข้าถึงเอกสาร XML
- User Name หมายถึงชื่อผู้ใช้งานที่ได้รับสิทธิในการเข้าถึงเอกสาร XML
- Document หมายถึงชื่อเอกสาร XML ที่ต้องการเข้าถึงข้อมูล



รูปที่ 3-13 รูปแบบการกำหนด URI ในโปรโตคอล XCAP

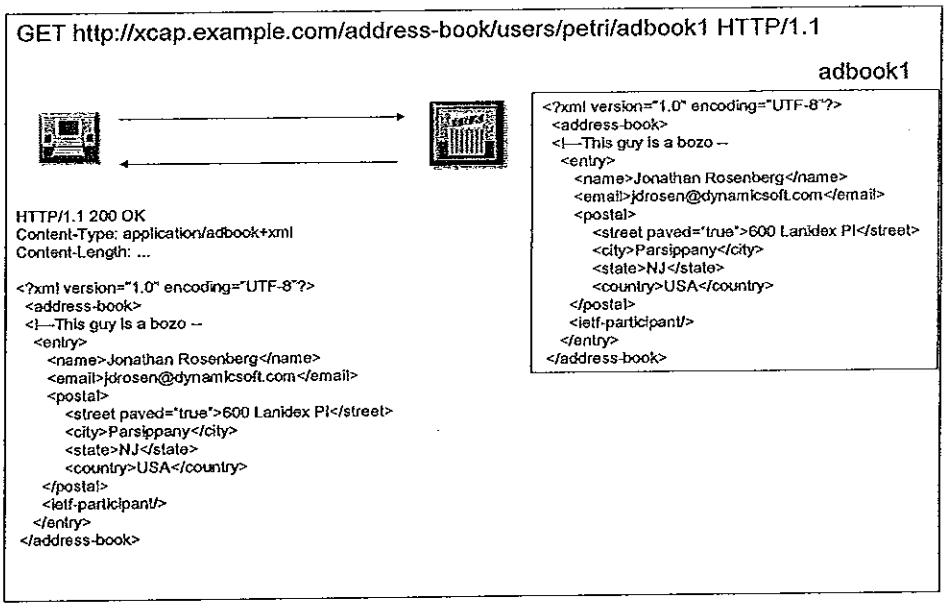
ซึ่งจากความสามารถในการเข้าถึงเอกสารในรูปแบบ XPath เมื่อนำมาใช้งานร่วมกับรูปแบบการส่งข้อมูลที่เป็นแบบ HTTP ก็จะทำให้สามารถที่จะแก้ไขข้อมูลที่เป็น XML ได้อย่างสมบูรณ์

3.3.2 รูปแบบการทำงานของโปรโตคอล XCAP

ในส่วนนี้จะกล่าวถึงรูปแบบของการนำโปรโตคอล XCAP มาใช้งานเพื่อทำการแก้ไขข้อมูลเอกสาร XML ต่าง ๆซึ่งจะแบ่งการทำงานออกเป็นส่วนต่าง ๆดังนี้

- Fetch Document

สำหรับรูปแบบและกระบวนการที่จะใช้ในการนำเอกสาร XML จาก Server เพื่อมาใช้งานนั้น โปรโตคอล XCAP ได้กำหนดให้ใช้งาน HTTP-GET เพื่อทำการดึงข้อมูลโดยจำกำหนด URI เป็นชื่อของเอกสาร XML ที่ต้องการดังรูปที่ 3-14

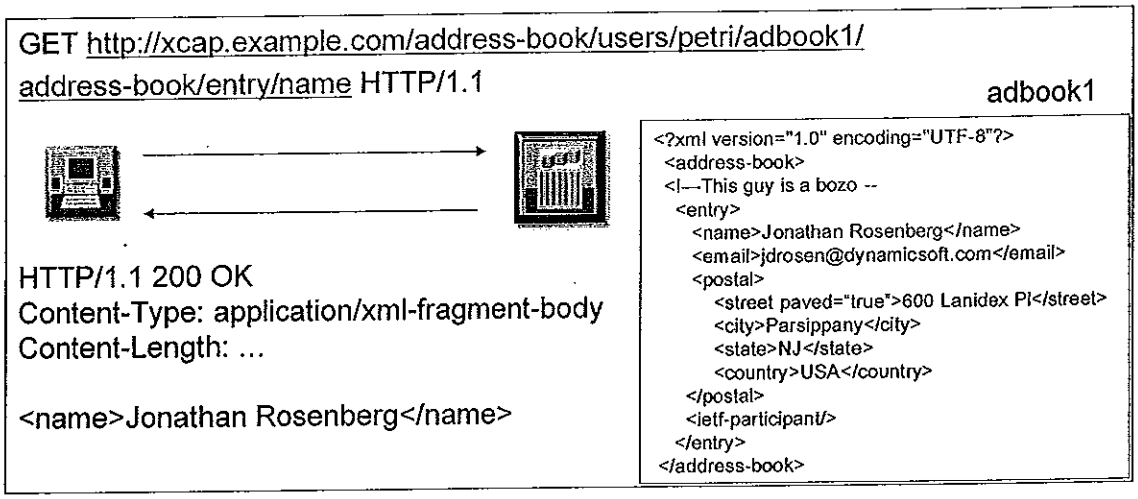


รูปที่ 3-14 การ Fetch Document

จากตัวอย่างจะเป็นการดึงข้อมูลที่ชื่อว่า adbook1 จาก XCAP Server โดยจะสังเกตได้จาก URI จะมีการระบุเป็น adbook1 ซึ่งจะเป็นการกำหนดเพื่อระบุว่าผู้ใช้ต้องเอกสาร XML ชื่ออะไร

- Fetch Element

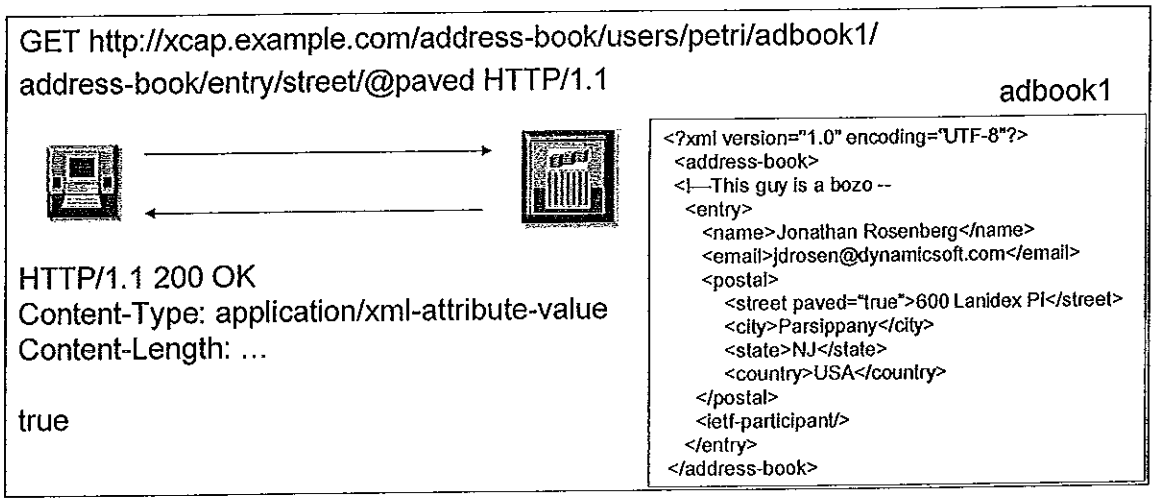
โปรโตคอล XCAP อนุญาตให้ผู้ใช้สามารถที่ดึงข้อมูลเฉพาะในส่วนของ Element ออกมาได้ซึ่งเป็นความสามารถที่เหนือกว่าโปรโตคอลอื่นๆ โดยรูปแบบการทำงานจะลักษณะที่ใกล้เคียงกับการ Fetch Document คือยังคงใช้งาน HTTP-GET เพื่อทำการดึงข้อมูลเช่นกันแต่จะต่างกันตรงที่จะต้องทำการระบุถึงชื่อ Element ที่ต้องการข้อมูลใน URI ด้วยดังแสดงในรูปที่ 3-15



รูปที่ 3-15 การ Fetch Element

- Fetch Attribute

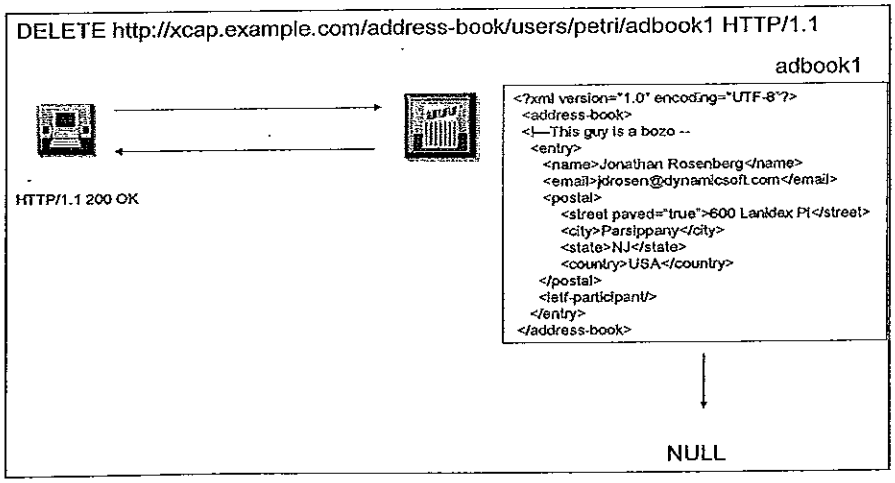
สำหรับในส่วนของการ Fetch Attribute เพื่อนำข้อมูลที่ถูกจัดเก็บอยู่ใน Attribute ของเอกสาร XML ซึ่งโปรโตคอล XCAP ได้มีการรองรับการทำงานตรงส่วนนี้ โดยการที่จะอ้างถึง Attribute ในเอกสาร XML นั้นการกำหนด URI จะมีความแตกต่างจากการ Fetch Document และ Element คือจะมีการใส่เครื่องหมาย "@" หน้าชื่อ Attribute ที่เราต้องการดึงข้อมูลดังแสดงในรูปที่ 3-16



รูปที่ 3-16 การ Fetch Attribute

- Delete Document

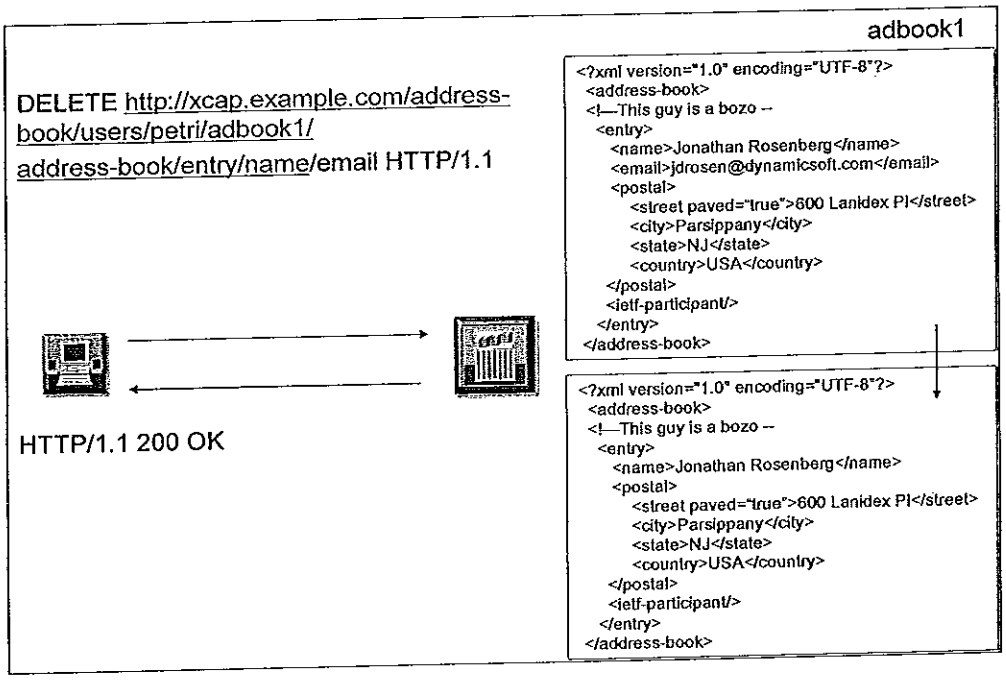
ในส่วนของการลบข้อมูลเอกสาร XML สำหรับโปรโตคอล XCAP ได้กำหนดให้ใช้งาน HTTP-DELETE ส่งไปยัง Server เพื่อให้ Server ทำการลบข้อมูลของเอกสาร XML ดังแสดงในรูปที่ 3-17



รูปที่ 3-17 การ Delete Document

- Delete Element

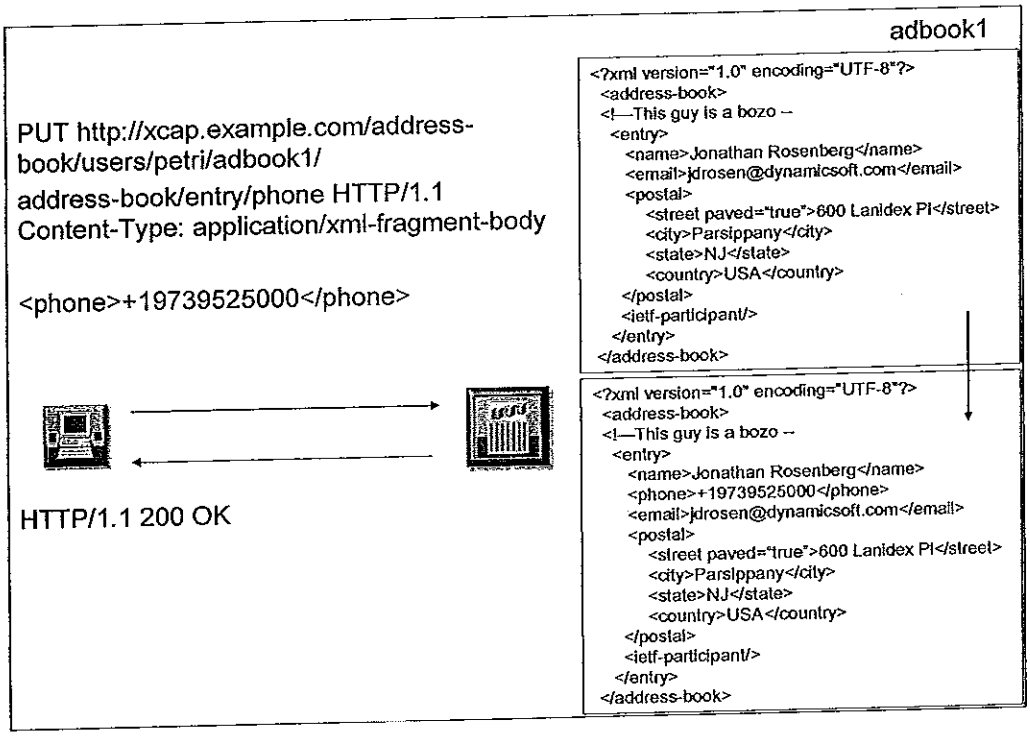
สำหรับในส่วนนี้จะลักษณะการทำงานเหมือนกับการ Delete document แต่จะแตกต่างกันในส่วนของการกำหนด URI เพื่อเข้าถึงข้อมูลที่ต้องการจะลบดังแสดงในรูปที่ 3-18



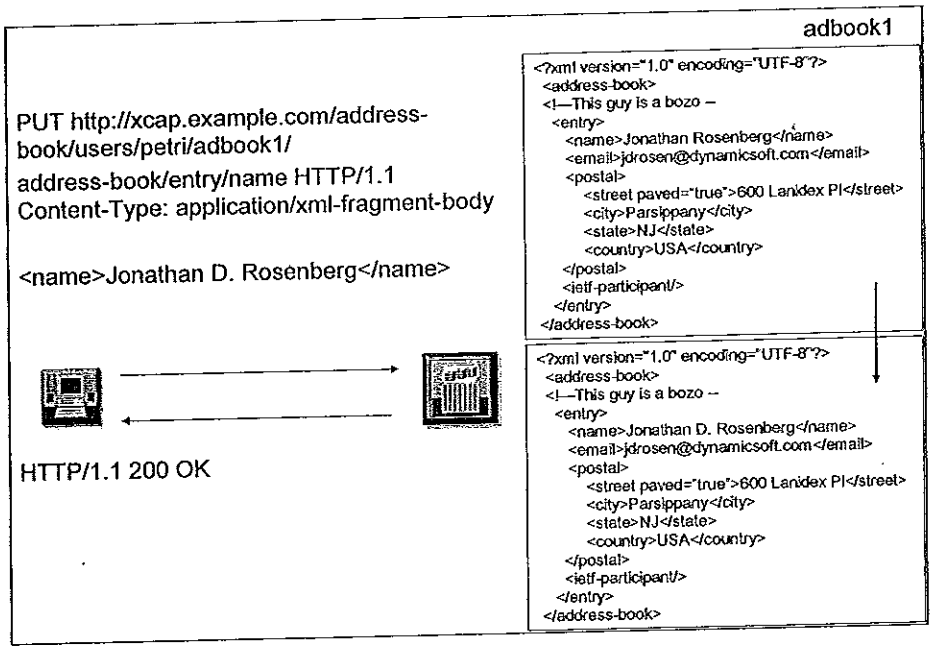
รูปที่ 3-18 การ Delete Elment

- Add and Modify Document, Element and Attribute

สำหรับในส่วนของการเพิ่มและแก้ไขสำหรับโปรโตคอล XCAP มีรูปแบบการทำงานที่เหมือนกันคือจะใช้งาน HTTP-PUT เพื่อทำการนำข้อมูลในฝั่งของผู้ใช้ไปจัดเก็บไว้บนฝั่ง Server ซึ่งจะแยกแยะการทำงานระหว่างการเพิ่มข้อมูลกับการแก้ไข โดยสังเกตจาก URI กล่าวคือ ถ้าหาก URI ที่กำหนดนั้น Server ได้ทำการตรวจสอบแล้วไม่พบว่าเป็น Document Element หรือ Attribute ที่ได้กำหนดไว้ก่อนหน้าแล้ว จะถือว่าผู้ใช้ได้ทำการร้องขอให้ทำการเพิ่มข้อมูลใหม่เข้าไปยังเอกสาร XML แต่ถ้าหากพบว่าได้มีการกำหนดข้อมูลไว้อยู่ก่อนหน้าแล้วนั้นจะถือว่าเป็นกระบวนการแก้ไขข้อมูล ซึ่งรูปแบบการทำงานจะแสดงในรูปที่ 3-19 และรูปที่ 3-20



รูปที่ 3-19 การเพิ่มข้อมูลลงในเอกสาร XML



รูปที่ 3-20 การแก้ไขข้อมูลในเอกสาร XML

3.3.3 การเปรียบความสามารถโปรโตคอล XCAP กับโปรโตคอลอื่น ๆ

จากการศึกษาคุณสมบัติและความสามารถของโปรโตคอล XCAP นั้นพบว่ามีความเหมาะสมอย่างยิ่งในการนำมาใช้เพื่อทำการกำหนดค่า Policy ต่างเพื่อใช้ในระบบการประชุม แต่ก็ยังมีโปรโตคอลอื่น ๆที่มีความสามารถหรือคุณสมบัติที่นำมาใช้ได้อีกมากมาย จึงได้มีแนวคิดในการศึกษาค้นคว้ารายละเอียดของโปรดตคอลเหล่านี้เพิ่มเติมเพื่อนำมาเปรียบเทียบความสามารถในด้านต่าง ๆเช่น ความเร็วในการรับส่งข้อมูล ความสามารถในการแก้ไขข้อมูล เป็นต้น กับโปรโตคอล XCAP เพื่อให้ได้โปรโตคอลที่เหมาะสมกับการทำงานมากที่สุด ซึ่งจะพบว่าแต่ละโปรโตคอลมีความเป็นไปได้ที่จะนำมาใช้งานในการกำหนด Conference Policy ต่าง ๆ แต่เมื่อสังเกตถึงข้อดีและข้อเสียของแต่ละโปรโตคอลจะพบว่าโปรโตคอล XCAP มีความสามารถและคุณสมบัติเหมาะสมที่สุดที่จะมาใช้ในการทำ Policy Exchange เนื่องจากคุณสมบัติเด่น ๆของโปรโตคอล XCAP ในเรื่องของการสืบทอดความสามารถมาจากโปรโตคอล HTTP ซึ่งทำให้ง่ายต่อการที่จะนำมาประยุกต์และพัฒนา เพื่อใช้งานกับระบบการประชุม นอกจากนี้ยังมีในส่วนของความสามารถที่จะแก้ไขข้อมูล XML เป็นส่วน ๆได้ ซึ่งเหมาะสมสำหรับใช้ในการกำหนดหรือแก้ไขค่า Policy ต่าง ๆสำหรับระบบการประชุมที่อาจจะมีการแก้ไขค่าเหล่านี้ได้ตลอดเวลาดังแสดงในตารางที่ 3-1

Features\Protocol	HTTP	SIP	SOAP	WebDAV	CCP	XML RPC	XCAP
Base on XML			X		X	X	X
Compact					X		
Easy to Implement	X	X		X			
Able to modify parts of document							
Security	X	X	X	X		X	

ตารางที่ 3-1 ตารางเปรียบเทียบโปรโตคอลต่าง ๆ

จากตารางที่ 3-1 จะพบว่าโปรโตคอล XCAP มีความสามารถเหนือกว่าโปรโตคอลอื่นในเรื่องของความสามารถในการแก้ไขข้อมูลในเอกสาร XML เป็นส่วน ๆได้ซึ่งโปรโตคอลอื่น ๆไม่มีความสามารถในส่วน จึงเป็นเหตุผลหลักที่ทำให้โปรโตคอล XCAP เหมาะสมที่จะนำมาใช้งานมากที่สุด นอกจากนี้โปรโตคอล XCAP ยังมีข้อได้เปรียบโปรโตคอลอื่น ๆในเรื่องของความง่ายต่อการนำมาพัฒนาเนื่องมาจากโปรโตคอล XCAP ได้ออกแบบโดยอยู่บนฐานของโปรโตคอล HTTP

3.4 สรุป

จากข้อมูลที่ได้ทำการค้นคว้ามาทั้งหมดพบว่าได้มีการนำเสนอเทคนิคและกระบวนการต่าง ๆที่ใช้ในควบคุมผู้ติดต่อในการประชุมแบบสื่อประสม ซึ่งเป็นประเด็นหนึ่งที่ยังคงต้องศึกษาเนื่องจากเป็นส่วนสำคัญที่จะทำให้ระบบการประชุมมีประสิทธิภาพและถูกต้องมากยิ่งขึ้น โดยในการศึกษาจะเริ่มต้นตั้งแต่รูปแบบข้อมูลที่จะต้องใช้ในการจัดเก็บกฎข้อบังคับต่าง ๆ ซึ่งแนวคิดที่นำเสนอมาส่วนใหญ่จะอยู่ในรูปแบบของ XML ที่สามารถเข้าได้ง่ายและยืดหยุ่นต่อการนำมาใช้งานในรูปแบบที่หลากหลาย

เมื่อได้รูปแบบของข้อมูลที่จะจัดเก็บแล้วส่วนต่อที่ต้องทำการศึกษาค้นคว้าคือรูปแบบของโปรโตคอลที่จะนำมาใช้ในการกำหนดและแก้ไขกฎข้อบังคับที่ใช้ในการประชุม ซึ่งมีแนวคิดรวมถึงเทคนิคที่น่าสนใจมากมาย แต่เมื่อทำการเปรียบเทียบและวิเคราะห์ถึงความเหมาะสมแล้วพบว่าโปรโตคอล XCAP เป็นโปรโตคอลที่มีความสามารถเหมาะสมที่จะนำมาใช้ในการทำหน้าที่เพื่อควบคุมการประชุมในสถาปัตยกรรมการประชุมแบบสื่อประสมดังที่กล่าวมาแล้วในบทนี้

ดังนั้นจึงเป็นไปได้ที่จะนำแนวคิดเหล่านี้มาทำการออกแบบสถาปัตยกรรมการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP ดังจะกล่าวถึงรายละเอียดในบทต่อไป

บทที่ 4
การออกแบบระบบการประชุมแบบสื่อประสม

ในบทนี้จะกล่าวถึงการออกแบบรูปแบบของสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP ซึ่งจากผลการศึกษาบทความและความสามารถของโปรโตคอลต่าง ๆที่กล่าวในบทที่แล้ว ทำให้สามารถที่จะวิเคราะห์และคัดเลือกโปรโตคอลที่เหมาะสมกับการนำมาใช้ในการควบคุมระบบการประชุมและสร้างการติดต่อต่าง ๆที่จะเกิดขึ้นในได้ โดยในการออกแบบจำเป็นที่จะต้องคำนึงถึงประสิทธิภาพในการทำงานของโปรโตคอลที่จะนำมาใช้ ซึ่งจะต้องให้ผลลัพธ์ที่รวดเร็วและถูกต้องซึ่งเป็นหัวใจหลักของการทำงานของระบบการประชุมแบบสื่อประสม โดยสิ่งที่จะนำเสนอในบทนี้จะกล่าวถึงโปรโตคอลที่จะเลือกมาใช้งาน รวมถึงองค์ประกอบอื่น ๆที่จะช่วยให้การทำงานและการติดต่อที่เกิดขึ้นในระบบการประชุมเพื่อช่วยอำนวยความสะดวกให้ผู้เข้าร่วมการประชุมสามารถสร้างการติดต่อได้ง่ายขึ้น เนื่องจากมีมาตรฐานที่ชัดเจน รวมถึงยังจะสามารถเพิ่มฟังก์ชันการทำงานต่าง ๆให้กับระบบการประชุมเพื่อตอบสนองความต้องการของผู้ใช้งานให้ได้มากที่สุด

4.1 เหตุใดจึงต้องออกแบบระบบการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP

จากการศึกษาข้อมูลในเชิงทฤษฎีซึ่งกล่าวในบทที่ 2 โดยได้นำเสนอการศึกษาถึงรูปแบบขององค์ประกอบต่าง ๆ ที่ต้องใช้สำหรับการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP ดังแสดงในรูปที่ 2-15 ซึ่งประกอบไปด้วยส่วนต่าง ๆ เหล่านี้เป็นส่วนประกอบพื้นฐานที่จำเป็นต้องมีหากต้องการสร้างระบบการประชุม แต่ทว่าหากพิจารณาถึงการที่จะนำรูปแบบของสถาปัตยกรรมดังกล่าวไปใช้พัฒนาเพื่อใช้งานจริงก็จะพบว่ายังคงขาดรายละเอียดอีกหลายส่วนที่ยังไม่ได้กล่าวถึงอย่างชัดเจนไม่ว่าจะเป็นประเด็นของรูปแบบในการที่จะควบคุมสิทธิของผู้เข้าร่วมการประชุมแต่ละคน รูปแบบของการเก็บข้อมูลรายละเอียดต่าง ๆในการประชุม เป็นต้น ดังนั้นในบทที่ 3 จึงได้กล่าวการศึกษาแนวคิดรวมถึงข้อมูลต่าง ๆที่เกี่ยวข้องไม่ว่าจะเป็น รูปแบบของโปรโตคอลต่าง ๆ ที่สามารถนำมาใช้เสริมการทำงานหรือรูปแบบโครงสร้างการเก็บข้อมูลต่าง ๆ เป็นต้น

ดังนั้นเมื่อนำข้อมูลและแนวคิดทั้งหมดที่มีมาทำการวิเคราะห์ก็จะสามารถที่จะทำการออกแบบและพัฒนาสถาปัตยกรรมการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP ขึ้นมาได้ โดยระบบจะต้องมีความสามารถในการควบคุมสิทธิต่าง ๆของผู้เข้าร่วมการประชุมต่าง ๆ ได้ รวมถึงความสามารถต่าง ๆ เพิ่มเติมซึ่งจะกล่าวถึงในหัวข้อต่อ ๆไป

4.2 การออกแบบองค์ประกอบเพิ่มเติมในระบบการประชุม

ในหัวข้อนี้ได้นำเสนอการศึกษาและออกแบบระบบเพื่อให้เกิดความชัดเจนมากยิ่งขึ้นในประเด็นของสัญญาณที่ใช้ในการติดต่อและได้มีการนำเสนอแนวคิดที่จะเพิ่มองค์ประกอบเพิ่มเติมเพื่อมาช่วยให้ระบบการประชุมสามารถทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น โดยในการออกแบบจะทำการระบุรายละเอียดของรูปแบบสัญญาณมาตรฐานที่ใช้ในการติดต่อระหว่างองค์ประกอบต่าง ๆ ทั้งหมด นอกจากนี้ได้ทำการเพิ่มองค์ประกอบพิเศษที่จะช่วยเพิ่มความสามารถให้กับการประชุม อันได้แก่

4.2.1.1 Conference Manager

จะทำหน้าที่ในการประสานงานระหว่างผู้เข้าร่วมการประชุมและ Conference Policy Server เพื่อทำการกำหนดค่าของ User Account, Policy หรือ Service ต่าง ๆ ตามที่ผู้เข้าร่วมการประชุมต้องการใช้งาน โดย Conference Manager นี้จะมีลักษณะการทำงานอยู่ในรูปแบบของ Application Server ที่ต้องรองรับการติดต่อจากองค์ประกอบต่าง ๆ ซึ่งจะสังเกตได้ว่าหากระบบไม่มีการทำงานในส่วนของ Conference Manager องค์ประกอบต่าง ๆในระบบจำเป็นต้องทำหน้าที่ติดต่อประสานงานกันเองทำให้เกิดความซับซ้อนในการพัฒนา เนื่องจากทุกองค์ประกอบต้องทำหน้าที่รับข้อมูลอื่น ๆเพิ่มเติมนอกเหนือจากข้อมูลปกติที่ต้องจัดการอยู่แล้ว นอกจากนี้ในกรณีที่ผู้ติดต่อที่ต้องการสร้างการประชุมหรือกำหนดค่าในการประชุมจำเป็นที่จะต้องติดต่อไปยัง Conference Policy โดยตรงทำให้จะต้องมีความสามารถในการติดต่อใช้โปรโตคอลอื่น ๆเพิ่มเติม

จะพบว่า Conference Manager ทำให้มีความสะดวกในการพัฒนาและแก้ไข ในกรณีที่มีการเปลี่ยนแปลงรูปแบบของข้อมูล Policy ซึ่งทำให้ผู้ติดต่อสามารถที่จะทำการเพิ่มเติมหรือแก้ไขข้อมูลต่าง ๆเหล่านี้ ได้อย่างสะดวก ซึ่งในระบบที่ออกแบบ Conference Manager จะติดต่อไปยังองค์ประกอบ 3 ส่วนที่สำคัญคือ

- **Participant**

  โดยแนวคิดสำหรับการติดต่อในส่วนนี้จะออกแบบให้ Conference Manager มีส่วนการทำงานของ Web Application ซึ่งมีข้อดีคือผู้ใช้สามารถเข้าถึงข้อมูลได้อย่างรวดเร็วและเป็นเทคโนโลยีที่ได้การยอมรับอย่างแพร่หลายในปัจจุบัน โดย Participant จะติดต่อมายัง Conference Manager เพื่อทำการกำหนดค่าการทำงานต่าง ๆของระบบการประชุม ผ่านทาง Web Browser ซึ่งหลังจาก Conference Manager ได้รับข้อมูลเหล่านี้ก็จะทำการประมวลผลและส่งต่อไปยังส่วนการทำงานอื่น ๆต่อไป

- **Focus**

  สำหรับแนวคิดสำหรับการติดต่อในส่วนนี้ Conference Manager จะมีหน้าที่ในการควบคุมการทำงานของ Focus ในการสร้างการติดต่อไปยังผู้เข้าร่วมการประชุมแต่ละคนตามที่กำหนดไว้ผ่านทางโปรโตคอล SIP ซึ่งจะเป็นการลดภาระการทำงานในการตัดสินใจต่าง ๆของ Focus ลงเนื่องจาก Conference Manager จะเป็นส่วนตัดสินใจแทน โดย Focus มีหน้าที่ในการจัดการ Session เท่านั้น

- **Conference Policy Server**

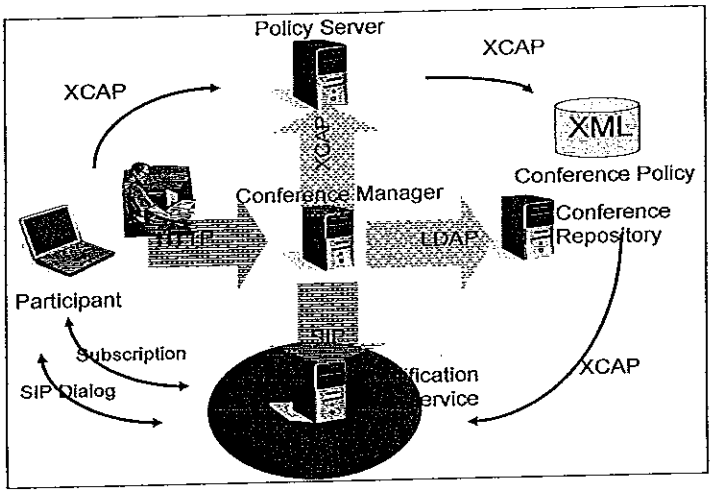
  ในการติดต่อระหว่าง Conference Manager และ Conference Policy Server จะกระทำผ่านทางโปรโตคอล XCAP เนื่องจากรูปแบบข้อมูลที่ถูกจัดเก็บใน Conference Policy Server อยู่ในรูปแบบ XML ซึ่งจากการศึกษาและเปรียบเทียบโปรโตคอลต่าง ๆในบทที่ 3 พบว่าโปรโตคอล XCAP เป็นโปรโตคอลที่เหมาะสมที่สุดที่จะนำมาใช้เพื่อให้ Conference Manager ทำหน้าที่กำหนดและแก้ไขค่าของ Confererence Policy

ดังนั้นจะสังเกตได้ว่าเมื่อได้มีการนำ Conference Manager เข้ามาใช้งานร่วมกับระบบการประชุมจะทำให้ผู้ติดต่อสามารถใช้งานได้สะดวกขึ้น เนื่องจากไม่จำเป็นต้องทำการเปลี่ยนแปลงในส่วนของผู้ติดต่อ และหากมีการเปลี่ยนส่วนของข้อมูลใด ๆ ก็จะเป็นการแก้ไขที่ฝั่งแม่ข่ายเท่านั้น ซึ่งจะไม่มีผลกระทบผู้ใช้แต่อย่างใด ทำให้ในการพัฒนาระบบการประชุมหากนำแนวคิดตรงส่วนนี้ไปใช้งานก็สามารถทำได้ง่ายและมีผลกระทบกับการใช้งานแบบเดิมน้อยที่สุด

4.2.1.2 Conference Repository

เป็นองค์ประกอบที่ทำหน้าที่เปรียบเสมือนฐานข้อมูล โดยทำการจัดเก็บรายละเอียดของการประชุมต่าง ๆเช่น Conference URI ชนิดของการประชุม เป็นต้น เพื่ออำนวยความสะดวกในการค้นหา Active Conference ที่ให้บริการอยู่ ณ เวลานั้น ซึ่งเหตุผลสำคัญที่ให้มีแนวคิดในการเสนอ Conference Repository ขึ้นมาเนื่องจากในการใช้งานจริงอาจจะมีกลุ่มการประชุมมากมาย ซึ่งระบบการประชุมแบบสื่อประสมแบบเดิมที่ทาง IETF ได้ออกแบบไว้ไม่ได้มีการนำเสนอแนวคิดในเรื่องของการค้นหาการประชุม ดังนั้นแนวคิดในการเพิ่ม Conference Repository จะช่วยให้ผู้ติดต่อสามารถที่จะค้นหาและเข้าร่วมการประชุมได้อย่างถูกต้องและรวดเร็วยิ่งขึ้น นอกจากนี้ในประเด็นเรื่องของความปลอดภัย Conference Repository สามารถที่จะช่วยป้องกันไม่ให้ผู้ไม่ประสงค์ดีสามารถเข้ามาร่วมการประชุมได้เนื่องมาจาก Conference URI จะถูกกำหนดให้อยู่ในรูปแบบของ Pseudo Random ทำให้ยากปลอมแปลงและรับรู้ โดยการทำงาน Confererence Repository จะติดต่อกับ Conference Manager เพื่อรับส่งข้อมูลหมายเลขห้องการ

ประชุม ทำให้ผู้ติดต่อสามารถที่จะค้นหาข้อมูลผ่านทาง Conference Manager ได้อีกทางหนึ่ง ซึ่งการติดต่อไปยัง Conference Repository นี้สามารถที่ทำได้โดยผ่านทาง LDAP [19]



รูปที่ 4-1 Service & Network Architecture

จากโครงสร้างของระบบทั้งหมดที่ได้ทำการออกแบบในรูปที่ 4-1 จะพบว่าได้มีการระบุถึงรายละเอียดของโปรโตคอลต่าง ๆไว้อย่างชัดเจนมากกว่าระบบเดิมที่ทาง IETF ได้ออกแบบไว้ นอกจากนี้ยังได้มีการเพิ่มองค์ประกอบที่ทำหน้าช่วยประสานการทำงานขององค์ประกอบอื่น ๆในระบบเข้าไว้ด้วยกันซึ่งทำให้ระบบที่ทำการออกแบบสามารถที่จะเพิ่มความสามารถอื่น ๆเข้าไปในระบบการประชุม เช่น การตั้งเวลาการประชุม การค้นหารายชื่อการประชุม เป็นต้น

จากที่กล่าวมาจึงสามารถสรุปได้ว่าแนวคิดของระบบการประชุมแบบสื่อประสมที่ได้ทำการออกแบบ ได้มีการเพิ่มเติมองค์ประกอบใหม่สองส่วนคือ Conference Manager และ Conference Repository และยังทำการระบุถึงรายละเอียดของโปรโตคอลที่ใช้ในการติดต่อไว้อย่างชัดเจน ดังนั้นการทำงานส่วนต่อไปที่ต้องคำนึงถึงคือรายละเอียดของสัญญาณการติดต่อต่าง ๆที่จะเกิดขึ้นได้ในระบบการประชุม ซึ่งจะกล่าวในหัวข้อต่อไป

4.3 รูปแบบของสัญญาณ SIP ในระบบการประชุม

ในสถาปัตยกรรมการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP ส่วนแรกที่ต้องคำนึงถึงคือรูปแบบของสัญญาณที่จะใช้ในการติดต่อเพื่อทำการเข้าร่วมการประชุมหรือออกจากการประชุม ซึ่งแนวคิดที่จะใช้ในการออกแบบรูปแบบของสัญญาณจะต้องยังคงยึดรูปแบบของมาตรฐานโปรโตคอล SIP ที่มีอยู่ในปัจจุบัน เนื่องจากหากมีระบบมีการสร้างรูปแบบสัญญาณใหม่

ที่แตกต่างนอกเหนือจากมาตรฐานเดิม ก็จะทำให้ผู้ติดต่อที่ใช้ SIP User Agent ที่มีอยู่อย่างแพร่หลายในปัจจุบัน ไม่สามารถที่จะเข้าใช้งานระบบการประชุมที่นำเสนอได้

ดังนั้นจึงสรุปได้ว่าสถาปัตยกรรมการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP ที่ได้ทำการออกแบบสามารถที่จะรองรับการทำงานผู้ติดต่อได้อย่างหลากหลาย ไม่ว่าผู้ติดต่อผู้นั้นจะใช้อุปกรณ์ใด ๆ ในการติดต่อก็ตาม โดยรูปแบบของการติดต่อและสัญญาณได้ถูกจำแนกออกเป็น 2 รูปแบบที่สำคัญคือ

- รูปแบบสัญญาณพื้นฐานสำหรับสร้างการประชุม (Basic Conference Scenarios)
- รูปแบบสัญญาณพิเศษ (Extended Conference Scenarios)

ซึ่งรูปแบบของสัญญาณที่ใช้และรูปแบบการติดต่อจะเริ่มตั้งแต่การสร้างการประชุม การเชิญผู้อื่นเข้าร่วมการประชุม เป็นต้น โดยในส่วนของรายละเอียดทั้งหมดของสัญญาณจะกล่าวในหัวข้อต่อไป

4.3.1 BASIC CONFERENCE SCENARIOS AND SIGNAL FLOWS

ในหัวข้อนี้จะกล่าวถึงรูปแบบการติดต่อเพื่อเข้าร่วมการประชุมและรูปแบบสัญญาณต่าง ๆ ที่เป็นมาตรฐานซึ่งถูกกำหนดโดย IETF โดยจะทำงานอยู่บนมาตรฐาน SIP ทั้งหมดซึ่งอ้างอิงจากมาตรฐาน RFC 4579 [20] และ RFC 4597 [21] โดยรูปแบบของสัญญาณที่กล่าวถึงจะแบ่งการทำงานออกเป็น 4 ประเภทคือ

- ทำการสร้างและเข้าร่วมการประชุม
- ทำการเชิญผู้อื่นเข้าร่วมการประชุม
- ทำการยกเลิกการประชุมหรือออกจากการประชุม
- ทำการค้นหาการประชุม

โดยการนำแนวคิดและรูปแบบสัญญาณมาตรฐานมาใช้ร่วมกับระบบการประชุมที่ได้ทำการออกแบบไว้ ซึ่งหากมีการระบุรูปแบบสัญญาณการติดต่อไว้อย่างชัดเจน ก็จะทำให้ในขั้นตอนพัฒนาระบบเป็นไปได้อย่างรวดเร็วและถูกต้อง สำหรับรูปแบบสัญญาณทั้งหมดที่ได้ทำการออกแบบจะมีรูปแบบดังต่อไปนี้

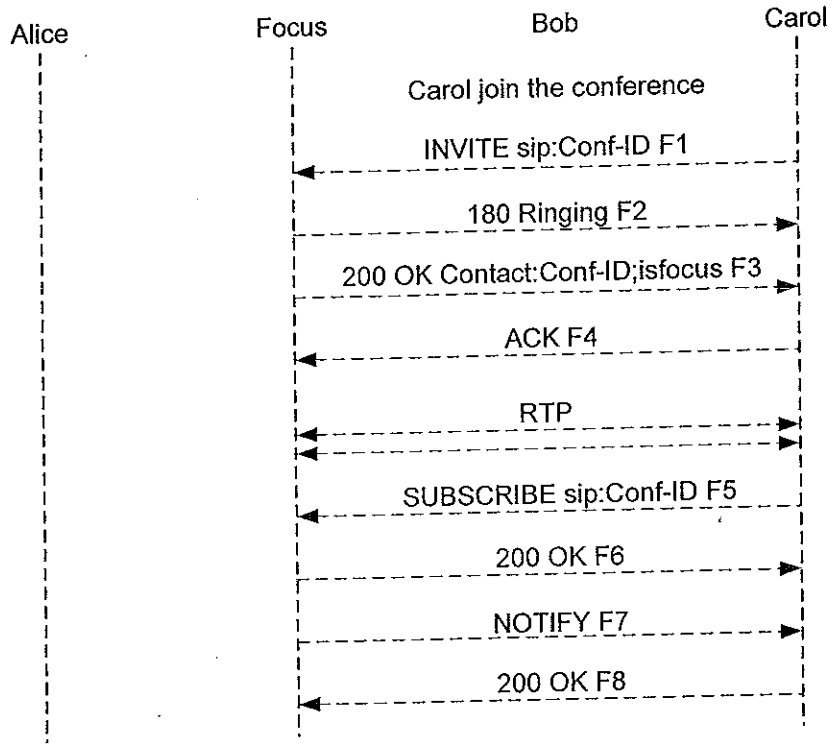
4.3.1.1 สัญญาณที่ใช้ในการสร้างและเข้าร่วมการประชุม

- **INVITE: Joining a Conference Using the Conference URI – Dial-In**

ในส่วนนี้ จะสมมุติว่าผู้ที่ต้องการเข้าร่วมการประชุมได้ทราบ Conference URI อยู่แล้วตั้งแต่แรก จึ่งได้ทำการติดต่อไปยัง Focus เพื่อขอเข้าร่วมการประชุม หลังจากนั้น Focus จะทำการตรวจสอบสิทธิต่าง ๆของผู้เข้าร่วมการประชุมคนนั้นก่อนที่อนุญาตให้เข้าร่วมการประชุม โดยตรวจสอบจากค่า Policy ต่าง ๆที่กำหนดใน Conference Policy

ซึ่งถ้าหากผู้เข้าร่วมการประชุมคนนั้นเป็นคนแรกที่เข้าร่วมการประชุม จะเปรียบเสมือนเป็นการกำหนดการเริ่มการทำงานของ Focus และเริ่มการประชุม แต่ถ้าหากการประชุมได้เริ่มทำงานอยู่ก่อนแล้ว ก็เปรียบเสมือนเป็นการเข้าร่วมการประชุมโดยทั่วไป

โดยสำหรับการเข้าร่วมการประชุมที่ได้ทำการประชุมอยู่แล้วนั้น ผู้ติดต่อสามารถที่จะทำการส่งสัญญาณ INVITE ที่มีการกำหนด Request-URI เป็น Conference URI ซึ่งเมื่อ Focus ได้รับสัญญาณจะทำการตอบกลับโดยมีการกำหนดพารามิเตอร์ isFocus เข้าไปใน Contact Header ของสัญญาณตอบกลับ 200 OK ดังแสดงในรูปที่ 4-2



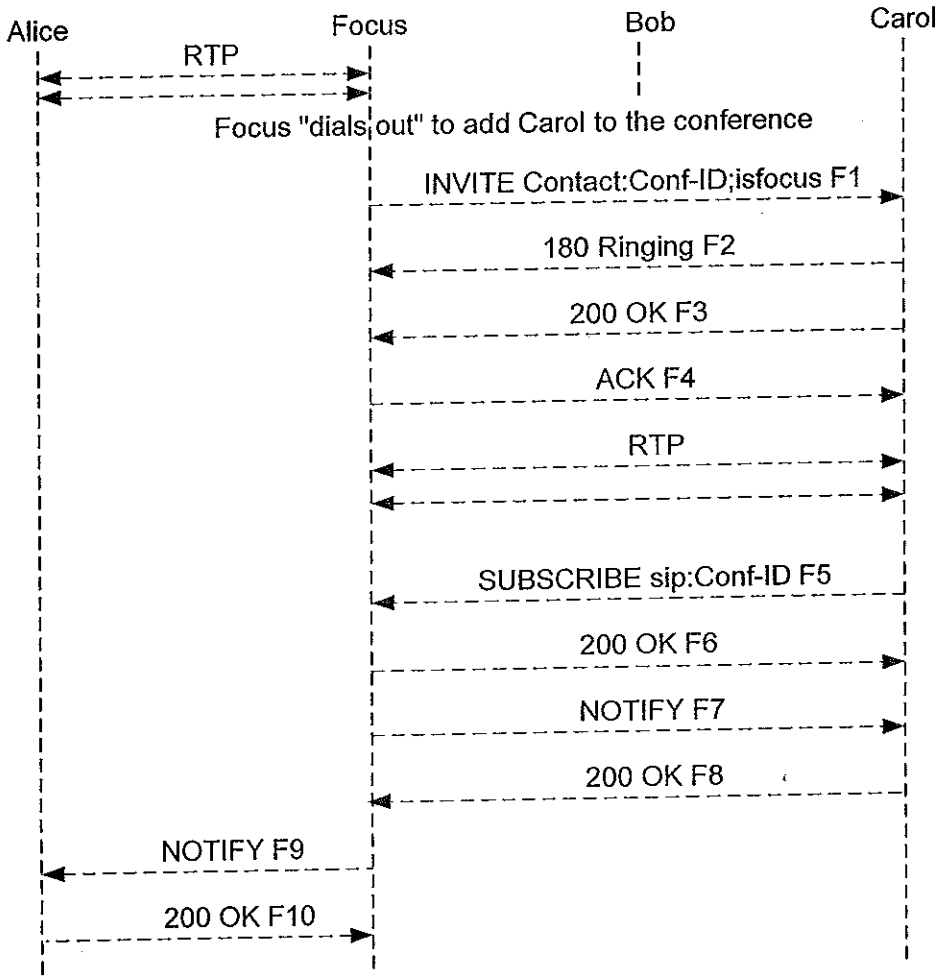
รูปที่ 4-2 INVITE: Joining a Conference Using the Conference URI – Dial-In

- **INVITE: Adding a Participant by the Focus – Dial-Out**

สำหรับวิธีการที่จะเชิญผู้อื่นเข้าร่วมการประชุมโดยตรงนั้น สามารถทำได้โดยการที่ Focus ทำการส่ง INVITE ไปยังผู้ที่ต้องการจะให้เข้าร่วมการประชุม โดยในสัญญาณ INVITE จะทำการระบุพารามิเตอร์ *isFocus* ไว้ใน Contact Header

ในส่วนของ Conference-Unaware UA ถ้าหากได้รับสัญญาณ INVITE ที่มีการระบุพารามิเตอร์ *isFocus* สามารถที่จะทำการเพิกเฉยต่อพารามิเตอร์ *isFocus* โดยจะมองการติดต่อเป็นแบบ point-to-point แทน สำหรับเหตุผลเนื่องมาจาก Conference-Unaware UA สามารถรองรับการติดต่อที่เป็นไปตามมาตรฐาน RFC 3261 เท่านั้น ซึ่งตามมาตรฐานได้มี

การระบุไว้ว่าให้ทำการเพิกเฉยกับพารามิเตอร์ที่ไม่รู้จักหากได้รับ ดังเช่นกรณีนี้คือพารามิเตอร์ *isFocus* ดังแสดงในรูปที่ 4-3

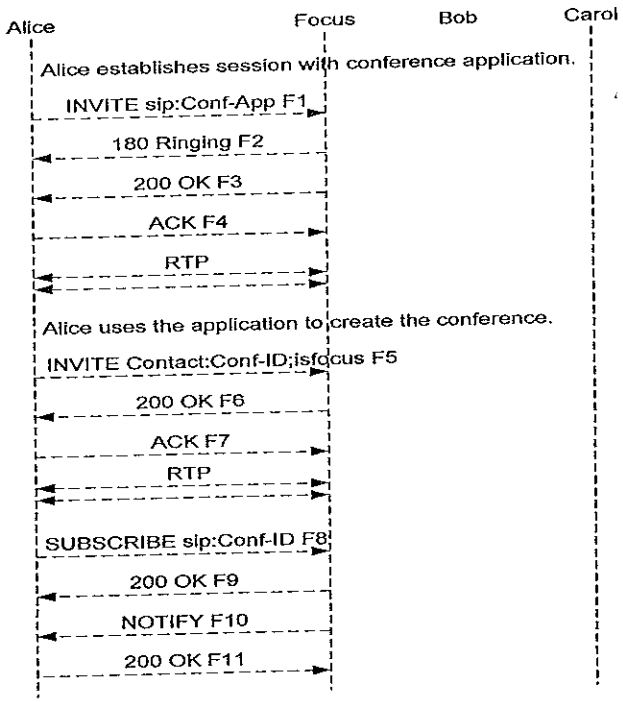


รูปที่ 4-3 INVITE: Adding a Participant by the Focus – Dial-Out

- **INVITE: Manually Creating a Conference by Dialing In to a Conferencing Application**

สำหรับในส่วนนี้ผู้ติดต่อจะทำการส่งสัญญาณ INVITE ไปยัง Conference Server Application เนื่องมาจากในบางกรณีการสำหรับการประชุมจำเป็นที่ต้องมีข้อมูลบางอย่างเพิ่มเติมก่อนที่เริ่มทำการประชุมได้ โดยอาจจะผ่านทางเว็บเพ็จหรือระบบตอบรับอัตโนมัติเป็นต้นซึ่งหลัง Conference Server ได้รับข้อมูลพิเศษเหล่านั้นแล้วระบบก็จะทำการเริ่มการประชุมและการทำงานของ Focus ได้ โดยที่ Focus จะทำการส่งสัญญาณ re-INVITE ไปยัง UA อีกครั้งโดยจะมีการเพิ่มพารามิเตอร์ *isFocus* เข้าไปยัง Contact Header เพื่อระบุว่าการติดต่อนี้เป็นทำการประชุม

ในอีกรูปแบบหนึ่ง สามารถที่จะจัดการส่งข้อมูลพิเศษเพื่อใช้ในการเริ่มการประชุมโดยผ่านทาง SIP Dialog ซึ่งสามารถทำได้ Conference Server จะทำการส่งสัญญาณตอบรับ 183 Session Progress เพื่อเป็นการแจ้งให้ผู้ติดต่อได้รับทราบว่าระบบกำลังจัดการประมวลผลข้อมูลได้รับ โดยหลังจาก Conference Server ทำการประมวลผลข้อมูลเรียบร้อยแล้ว Conference Server จะทำการส่งสัญญาณตอบรับ 200 OK พร้อมกับกำหนด Conference URI ให้ใน Contact Header จากนั้นรูปแบบการติดต่อจะมีลักษณะเหมือนการ Dial-In ดังแสดงในรูปที่ 4-4

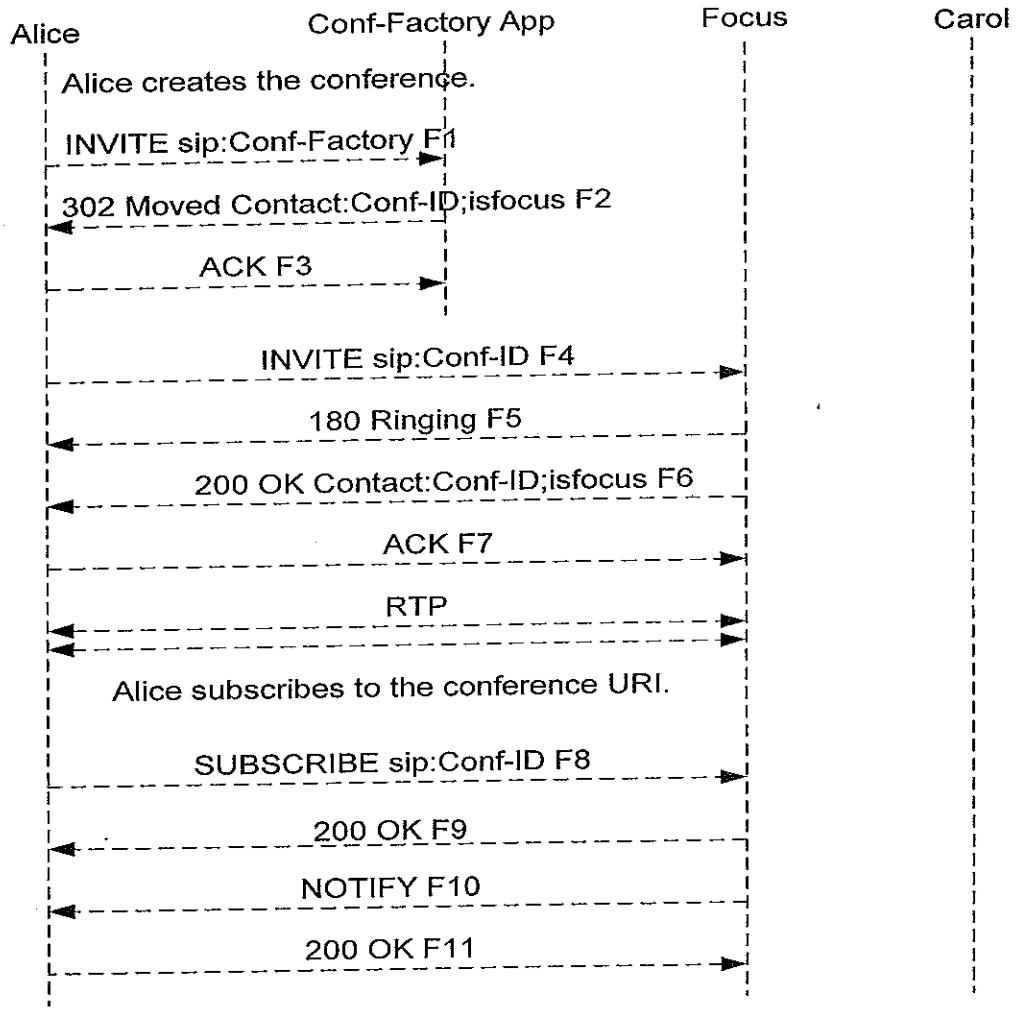


รูปที่ 4-4 INVITE: Manually Creating a Conference by Dialing In to a Conferencing Application

- **INVITE: Creating a Conference Using Ad-hoc SIP Methods**

สำหรับในหัวข้อนี้การทำงานจะแตกต่างจากหัวข้อที่แล้วตรงที่ Conference URI จะถูกกำหนดไว้ตั้งแต่แรกโดย Focus หรืออาจจะทำการกำหนดได้ผู้เข้าร่วมการประชุมทำการเป็นผู้กำหนดเอง

จากตัวอย่างสัญญาณในรูปที่ 4-5 จะสังเกตได้ว่าผู้เข้าร่วมการประชุมจะทำการสร้างการประชุมโดยการติดต่อไปยัง Conference Factory เพื่อร้องขอ Conference URI จะใช้ในการติดต่อไปยัง Focus จากนั้น Conference Factory จะทำการคืนค่าของ Conference URI มากับสัญญาณ 302 Moved Temporally หลังจากนั้นผู้เข้าร่วมการประชุมจะทำการ Dial-In ไปยัง Conference URI ที่ได้รับ

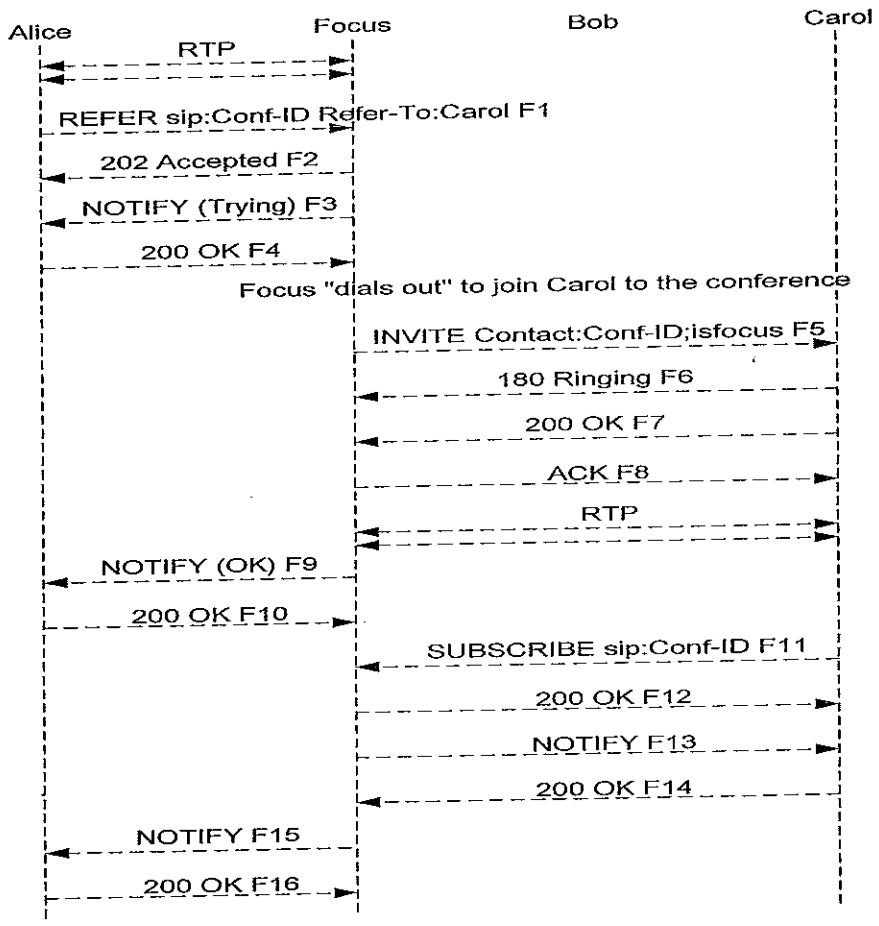


รูปที่ 4-5 INVITE: Creating a Conference Using Ad-hoc SIP Methods

4.3.1.2 สัญญาณที่ใช้ในเชิญผู้อื่นเข้าร่วมการประชุม

- **REFER: Requesting a Focus to Add a New Resource to a Conference (Dial Out to a New Participant)**

สำหรับในส่วนนี้จะมีรูปแบบการติดต่อโดยจะมีผู้เข้าร่วมการประชุมที่ได้ทำการติดต่อกับ Focus อยู่แล้วตั้งแต่แรก หลังจากนั้นผู้เข้าร่วมการประชุมต้องการที่จะทำการเชิญผู้อื่นติดต่อเข้ามายังการประชุม ซึ่งมีกระบวนการที่สามารถทำได้หลายวิธี แต่วิธีหนึ่งที่สามารถทำได้คือ การส่งสัญญาณ REFER [22] ไปยัง Focus โดยมีการระบุ Refer-To Header เป็น SIP URI ของผู้ที่ต้องการจะเชิญเข้ามาร่วมการประชุม หลังจากที่ Focus ได้รับสัญญาณ REFER แล้ว Focus จะทำการ Dial-Out ไปยังปลายทางเพื่อเชิญผู้ติดต่อนั้นเข้ามายังการประชุมดังแสดงในรูปที่ 4-6

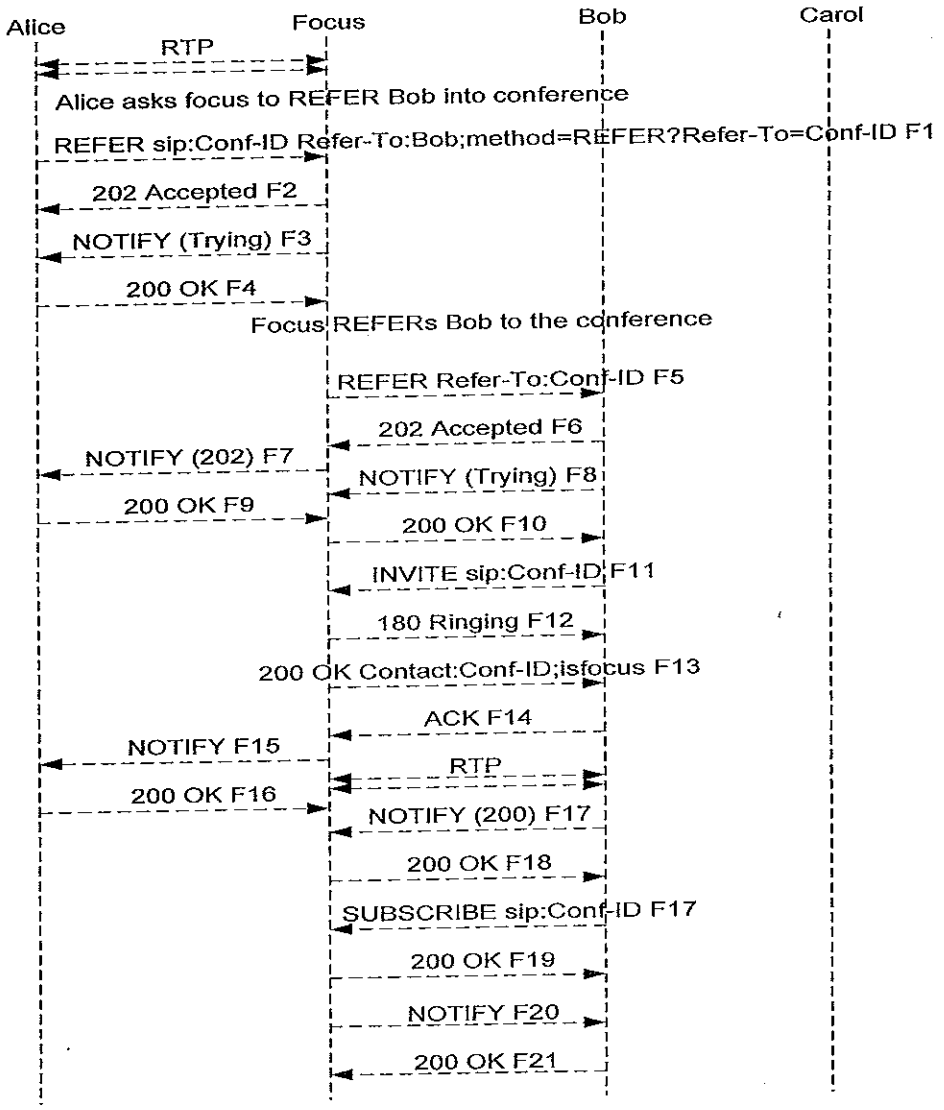


รูปที่ 4-6 REFER: Requesting a Focus to Add a New Resource to a Conference

(Dial Out to a New Participant)

- **REFER with REFER: Requesting a Focus to Refer a Participant to Dial in to the Conference**

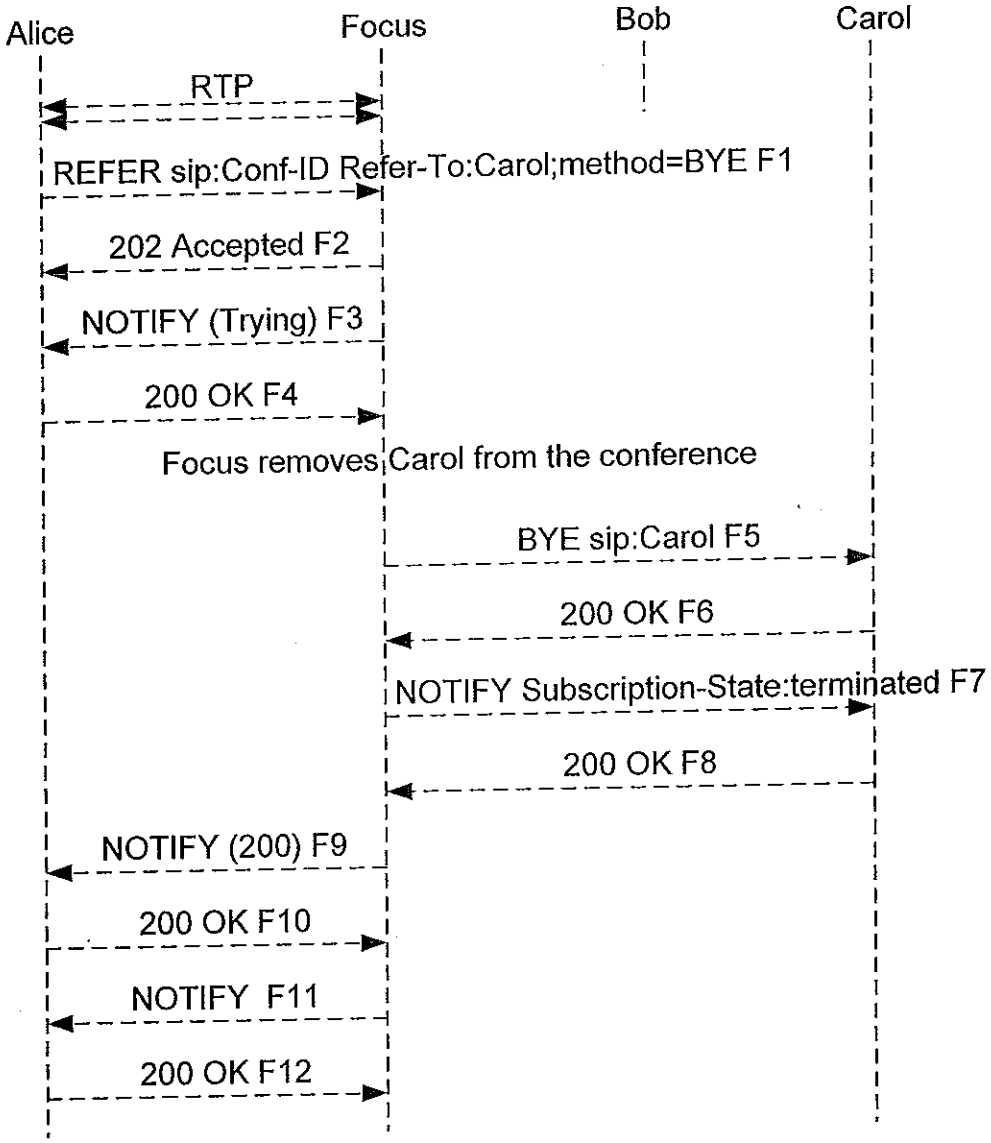
สำหรับส่วนนี้จะยังคงมีลักษณะที่ใกล้เคียงกับหัวข้อที่แล้วจะแตกต่างกันตรงที่ผู้เข้าร่วมการประชุมจะทำการส่งสัญญาณ REFER ไปยัง Focus เพื่อให้ Focus ทำการ REFER ต่ออีกครั้งหนึ่ง ซึ่งการทำเช่นนี้จะเป็นการทำให้ผู้ที่ถูกเชิญจะต้องทำการ Dial-In เข้ามายัง Focus เองเพื่อเข้าร่วมการประชุมดังแสดงในรูปที่ 4-7



รูปที่ 4-7 REFER with REFER: Requesting a Focus to Refer a Participant to Dial in to the Conference

- **REFER with BYE: Requesting That the Focus Remove a Participant from a Conference**

ในกระบวนการนี้จะเหตุการณ์ที่ผู้เข้าร่วมการประชุมบางคนต้องการที่จะขับไล่หรือยกเลิกการติดต่อผู้เข้าร่วมการประชุมคนออกจากการประชุม ซึ่งกระบวนการนี้สามารถทำได้โดยทำการส่งสัญญาณ REFER ที่มีการระบุ Refer-To Header เป็น SIP URI ของผู้ที่ต้องการให้ออกจากการประชุมและจะต้องทำการระบุพารามิเตอร์ method=BYE หลังจากนั้นเมื่อ Focus ได้รับสัญญาณ Focus จะทำการส่งสัญญาณ BYE ไปยัง SIP URI ที่ถูกระบุไว้เพื่อยกเลิกการติดต่อดังแสดงในรูปที่ 4-8

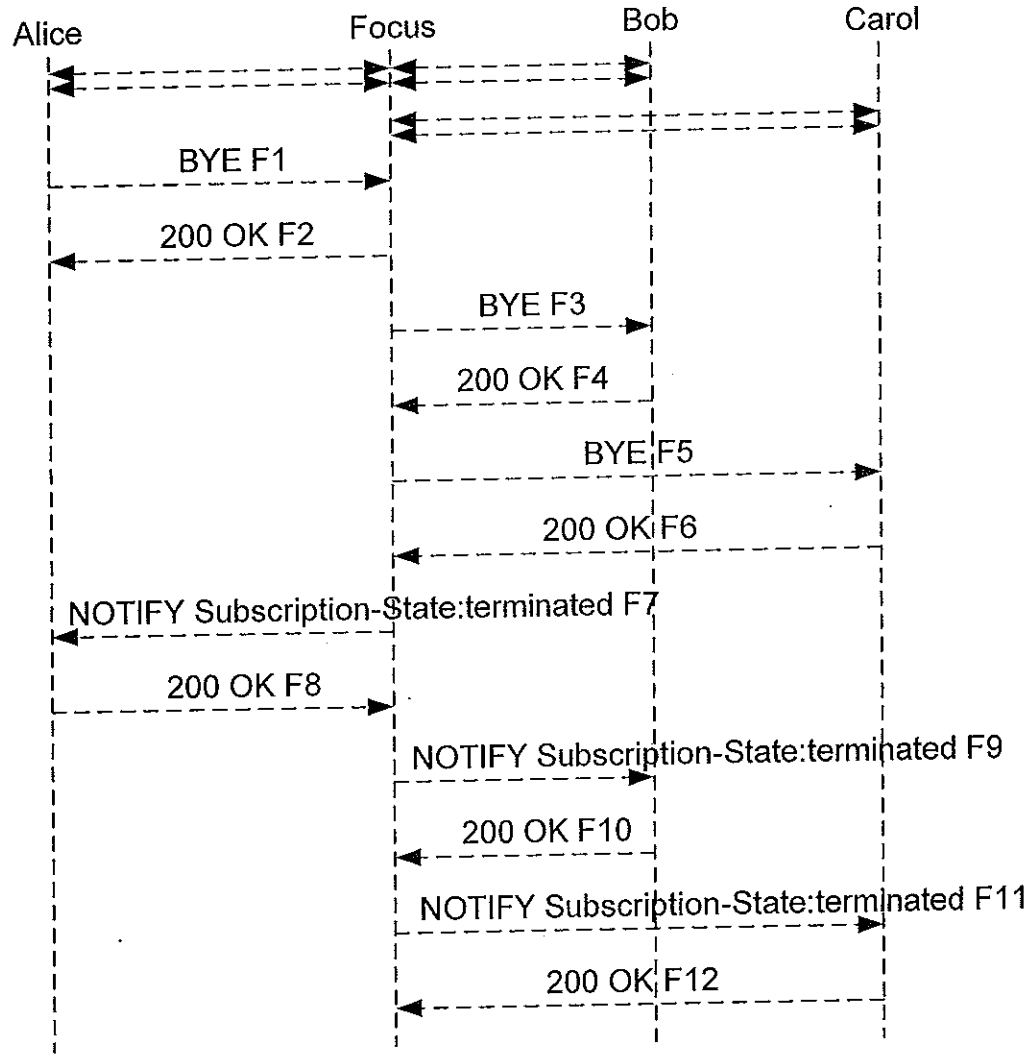


รูปที่ 4-8 REFER with BYE: Requesting That the Focus Remove a Participant from a Conference

4.3.1.3 สัญญาณที่ใช้ในยกเลิกการประชุม

- **Deleting a Conference**

สำหรับในหัวข้อนี้จะเป็นการยกเลิกการประชุมทั้งหมดโดยผู้ที่จะทำการยกเลิกได้นั้นจะต้องได้รับสิทธิที่กำหนดอยู่ใน Conference Policy ซึ่งถ้าหากผู้เข้าร่วมการประชุมคนนั้นมีสิทธิในการยกเลิกการประชุมทั้งหมด ก็จะสามารถยกเลิกการประชุมได้โดยการส่งสัญญาณ BYE ไปยัง Focus เมื่อ Focus ได้รับสัญญาณ BYE แล้ว Focus ก็จะทำการค้นหารายชื่อผู้เข้าร่วมการประชุมทั้งหมดเพื่อทำการส่งสัญญาณ BYE ไปยังผู้เข้าร่วมการประชุมเหล่านั้นเพื่อยกเลิกการติดต่อทั้งหมดดังแสดงในรูปที่ 4-9

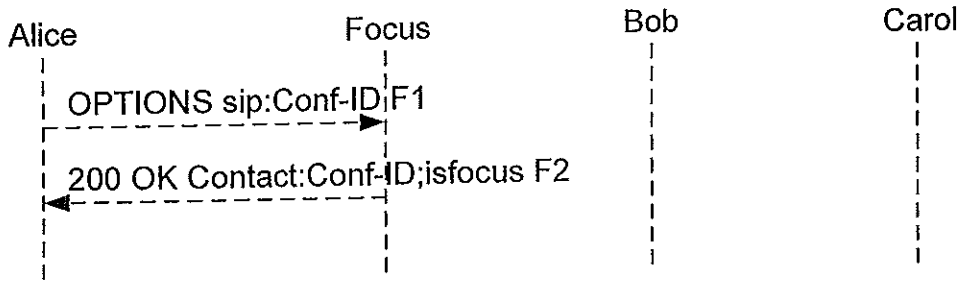


รูปที่ 4-9 Deleting a Conference

4.3.1.4 สัญญาณที่ใช้ในค้นหาการประชุม

- **Discovery of URI Properties Using OPTIONS**

สำหรับส่วนนี้จะเป็นรูปแบบสัญญาณ SIP เพื่อใช้ในการตรวจสอบว่า Focus มี Conference URI เป็นอะไร โดยผู้ติดต่อจะทำได้โดยส่งสัญญาณ OPTIONS ไปยัง SIP URI ต้องการจะค้นหา หากได้รับสัญญาณตอบรับเป็น 200 OK กลับมาแสดงว่า Focus พร้อมที่จะให้บริการทำการประชุมดังแสดงในรูปที่ 4-10



รูปที่ 4-10 Discovery of URI Properties Using OPTIONS

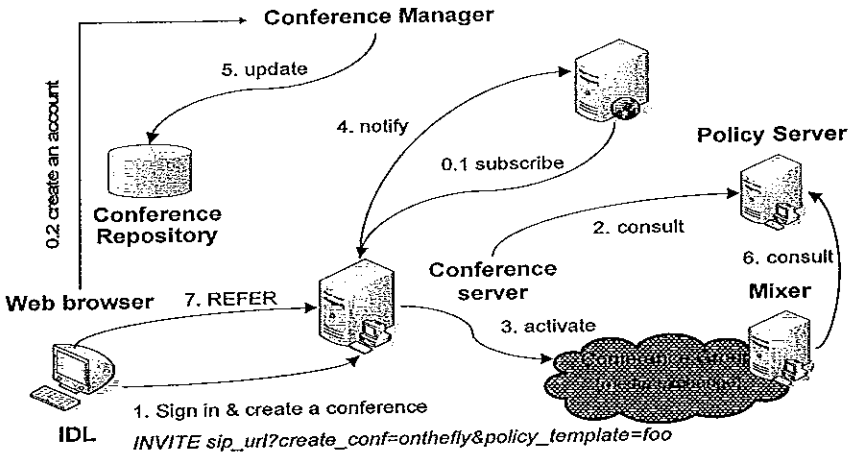
จากรูปแบบสัญญาณทั้งหมดที่กล่าวมาจะเน้นที่การติดต่อระหว่างผู้เข้าร่วมการประชุมกับ Focus เป็นหลัก โดยยังไม่ได้กล่าวถึงการติดต่อกับองค์ประกอบใหม่ที่นำเสนอ ดังนั้นในหัวข้อต่อไปจะกล่าวถึงรูปแบบการใช้งานหากได้มีการเพิ่มองค์ประกอบใหม่เข้าไปในระบบ

4.3.2 EXTENDED CONFERENCE SCENARIOS AND SIGNAL FLOWS

ในหัวข้อนี้จะกล่าวถึงรูปแบบการติดต่อที่สำคัญต่าง ๆ ซึ่งจะแสดงถึงความสามารถและประโยชน์ของระบบที่ได้ทำการออกแบบไว้ในหัวข้อที่ผ่านมา โดยประกอบไปด้วยรูปแบบต่างดังนี้

4.3.2.1 Just Conferencing Scenarios

สำหรับ Scenario แรกนี้จะเป็นการเริ่มสร้างและทำการประชุมแบบทันทีทันใด โดยผู้สร้างการประชุมไม่ต้องทำการกำหนด Policy ที่ซับซ้อน แล้วผู้สร้างการประชุมทราบถึงรายละเอียดของผู้เข้าร่วมการประชุมที่แน่นอน



รูปที่ 4-11 Just Conferencing Scenario

จากรูปที่ 4-11 เป็นรูปแบบการติดต่อที่แสดงการทำงานของระบบการสร้างการประชุม โดยมีลำดับการทำงานดังนี้

a. Conference Manager จะทำการ SUBSCRIBE ไปยัง Conference Server เพื่อตรวจว่า Conference Server พร้อมให้บริการหรือไม่

b. ในขณะเดียวกันผู้เข้าร่วมการประชุมก็จะทำการสร้าง Account ของตนเอง เพื่อที่จะใช้ในการเข้าร่วมการประชุมในกรณีที่ผู้เข้าร่วมคนนั้นยังไม่ได้มี Account อยู่ก่อน โดยจะทำผ่านทาง Conference Manager

1. หลังจากที่ได้ทำการสร้าง Account เป็นที่เรียบร้อยแล้ว ก็จะทำการส่งสัญญาณ INVITE ไปยัง Conference Server โดยมีการระบุพารามิเตอร์เพิ่มเติมเพื่อเป็นการแจ้งให้ Conference Server ทราบว่าเป็นการสร้าง Conference ซึ่งพารามิเตอร์ที่ใช้นั้นคือ
   - create_conf=onthefly เป็นการระบุว่าให้ Conference Server สร้าง Conference ทันทีได้รับสัญญาณนี้
   - policy_template=foo เป็นการระบุถึง Conference Policy ที่จะใช้สำหรับการประชุมนี้

2. หลังจากที่ Conference Server ได้รับสัญญาณแล้ว Conference Server จะทำการตรวจสอบสิทธิกับ Policy Server ว่าผู้ที่ร้องขอในการสร้างการประชุมนั้นมีสิทธิที่จะสร้างได้หรือไม่

3. หากผู้ที่ทำการสร้างการประชุมมีสิทธิที่จะทำการสร้างการประชุมได้นั้น Conference Server จะทำการ Activate การประชุมโดยติดต่อไปยัง Mixer เพื่อแจ้งให้ Mixer เตรียมเริ่มการทำงาน
4. จากนั้น Conference Server จะทำการส่ง Notify ไปแจ้งให้ Conference Manager ทราบถึงรายละเอียดของ Conference Group ใหม่ที่ได้สร้างขึ้นมา
5. Conference Manger จะทำการเก็บข้อมูลของ Conference Group ที่ได้รับไว้ที่ Conference Repository เพื่อให้ผู้เข้าร่วมการประชุมคนอื่นสามารถค้นหาได้
6. ในส่วนของ Mixer จะทำการติดต่อไปยัง Policy Server เพื่อนำข้อมูลของ Media Policy ซึ่งจำเป็นต่อการควบคุม Media Stream ที่ส่งกันระหว่างการประชุม
7. หลังจากทำการกำหนดค่าเริ่มต้นต่าง ๆหรับการประชุมเป็นที่เรียบร้อยแล้ว ผู้สร้างการประชุมจะทำการเชิญผู้เข้าร่วมการประชุมคนอื่นเข้ามาร่วมการประชุม โดยการส่งสัญญาณ REFER ไปยัง Conference Server

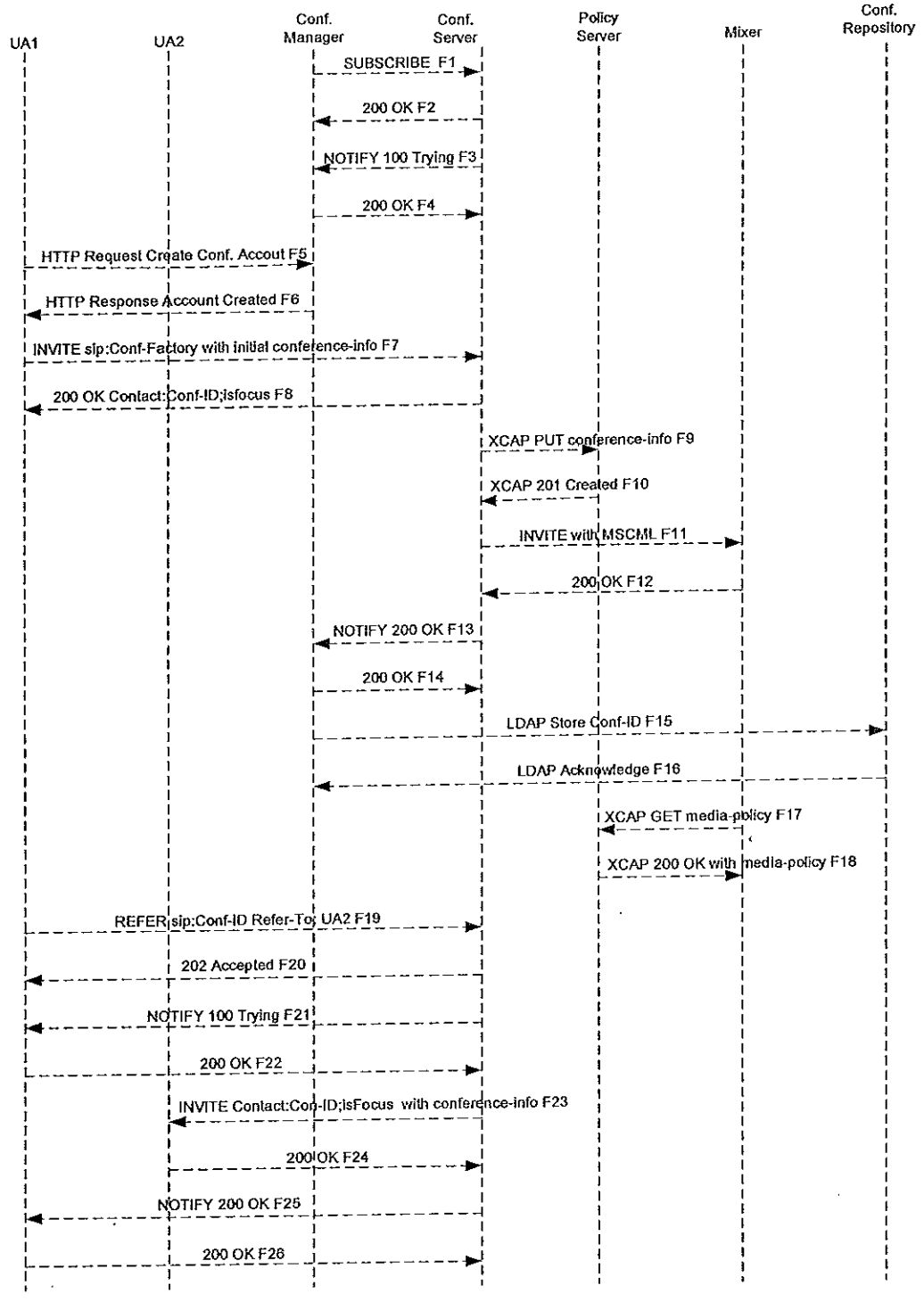
4.3.2.2 Just Conferencing signal flows

ส่วนนี้จะอธิบายถึงรูปแบบและลำดับของสัญญาณต่าง ๆที่ใช้ในการสร้างการประชุมแบบ Just Conferenceดังแสดงในรูปที่ 4-12

- เริ่มต้น Conference Manager ทำการส่งสัญญาณ SUBSCRIBE ไปยัง Conference Server หลังจากนั้น Conference Server จะทำการตอบรับด้วยสัญญาณ 200 OK และตามด้วยสัญญาณ NOTIFY เพื่อแจ้งว่ายังไม่มีการสร้างการประชุมเกิดขึ้น
- เมื่อ UA1 ทำการสร้าง Conference Account ไปยัง Conference Manager โดยผ่าน HTTP เป็นที่เรียบร้อยแล้ว UA1 สามารถที่จะทำการเริ่มการประชุมได้โดยการส่งสัญญาณ INVITE F7 ที่ประกอบไปด้วยข้อมูล SDP และข้อมูลเบื้องต้นของการประชุมนั้น ๆ ไปยัง Conference Server.
- หลังจากนั้น Conference Server จะทำการตอบรับด้วยสัญญาณ 200 OK ที่มีรายละเอียดของ SDP ซึ่งใช้สำหรับการติดต่อ
- Conference Server นำข้อมูล Conference Policy ที่ได้จาก UA1 ไปจัดเก็บไว้ที่ Policy Server ผ่านทาง XCAP PUT F9
- Policy Server ทำการตอบรับด้วยสัญญาณ 201 Created กลับไปยัง Conference Server หลังจากนั้น Conference Server ทำการติดต่อไปยัง Mixer เพื่อแจ้ง

เริ่มทำการประชุมโดยใช้สัญญาณ INVITE F11 ที่ประกอบไปด้วยข้อมูล MSCML [23]

- Conference Server ทำการส่งสัญญาณ NOTIFY F13 ไปยัง Conference Manager เพื่อแจ้งว่ามีการสร้างการประชุมขึ้นมาใหม่
- หลังจากนั้น Conference Manager ทำการจัดเก็บข้อมูลของ Conf-ID ไว้ที่ Conference Repository โดยใช้ LDAP F15
- Mixer ทำการร้องขอข้อมูล Media Policy จาก Policy Server เพื่อใช้ในการควบคุม media stream ระหว่างทำการประชุม โดยใช้ XCAP GET F17
- UA1 ทำการส่งสัญญาณ REFER F19 ไปยัง Conference Server เพื่อเชิญ UA2 เข้าร่วมการประชุม
- หลังจากนั้น Conference Server ทำการส่งสัญญาณ INVITE F23 ไปยัง UA2 ที่ประกอบไปด้วยข้อมูลเบื้องต้นของการประชุม



รูปที่ 4-12 Just Conference signaling flow

ในรูปที่ 4-13 แสดงถึงรูปแบบของสัญญาณที่ใช้ในการสร้างการประชุม โดยในส่วนของ Body message ประกอบด้วยข้อมูลสองส่วนคือ ข้อมูล SDP ที่ใช้ในการสร้าง session การติดต่อเพื่อใช้ในการรับส่ง media stream และส่วนของ conference information โดยในสัญญาณจะมีการกำหนด Content-Type header ให้เป็นแบบ "multipart/mixed"

```
INVITE sip:conf-id@sip.cnrlab.net SIP/2.0
Via: SIP/2.0/UDP coe.psu.ac.th;branch=z9hG4bKa38s
Max-Forwards: 70
To: sip:conf-id@sip.cnrlab.net
From: <sip:ua@sip.cnrlab.net>;tag=13323
Call-ID: skdj9898dffs89df
CSeq: 1 INVITE
Contact: <sip:ua@ua.cnrlab.net>
Allow: INVITE, ACK, CANCEL, BYE, REFER
Allow-Events: dialog
Accept: application/sdp, message/sipfrag
Require: conference-info-invite
Content-Type: multipart/mixed;boundary="boundary1"
Content-Length: ...

--boundary1
 Content-Type: application/sdp

 (SDP not shown)
 --boundary1
 Content-Type: application/conference-info+xml
 Content-Disposition: conference-info

 <?xml version="1.0" encoding="UTF-8"?>
 <conference-info
  xmlns="urn:ietf:params:xml:ns:conference-info">
  <conference-description>
   <subject>Teacher meeting</subject>
```

```
<service-uris>
 <entry>
  <uri>http://cnr.coe.psu.ac.th/sip/</uri>
  <purpose>web-page</purpose>
 </entry>
</service-uris>
<maximum-user-count>10</maximum-user-count>
</conference-description>
```

รูปที่ 4-13 Initial INVITE request received at the Conference Server

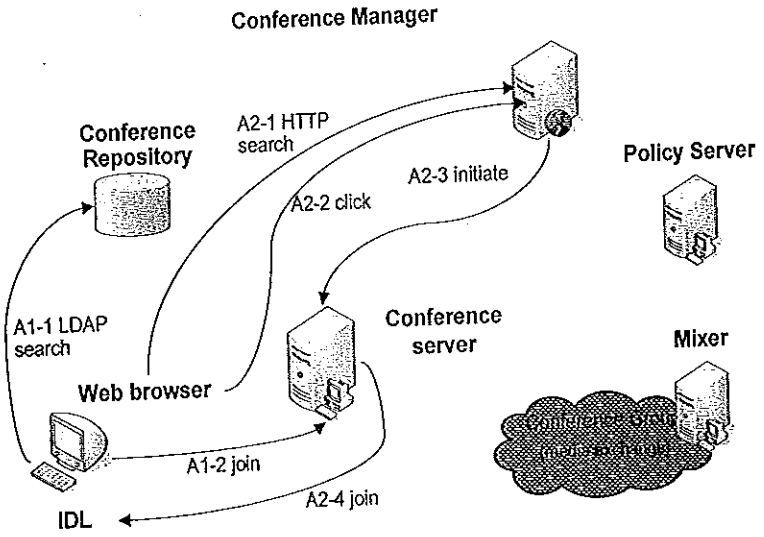
ในรูปที่ 4-14 จะแสดงรูปแบบสัญญาณของโปรโตคอล XCAP ที่ใช้ในการจัดเก็บข้อมูล conference information โดยรูปแบบสัญญาณอยู่ในรูปของ HTTP PUT

```
PUT http://xcap.cnrlab.net/xcap/global/confid/ HTTP/1.1
Content-Type: application/xml-fragment-body

 <?xml version="1.0" encoding="UTF-8"?>
 <conference-info
  xmlns="urn:ietf:params:xml:ns:conference-info">

  <conference-description>
   <subject>Teacher meeting</subject>
   <service-uris>
    <entry>
     <uri>http://cnr.coe.psu.ac.th/sip/</uri>
     <purpose>web-page</purpose>
    </entry>
   </service-uris>
   <maximum-user-count>10</maximum-user-count>
  </conference-description>
```

รูปที่ 4-14 XCAP PUT request for specific conference information

4.3.2.3 Group finding

สำหรับ Scenario นี้จะเป็นการแสดงถึงวิธีการค้นหาข้อมูลการประชุมเพื่อทำการเข้าร่วมการประชุม ซึ่งจากโครงสร้างของระบบที่ได้ทำการออกแบบไว้นั้นสามารถแบ่งได้เป็น 2 กรณี ดังแสดงในรูปที่ 4-15



รูปที่ 4-15 Group Finding Scenario

- ผ่านทาง LDAP

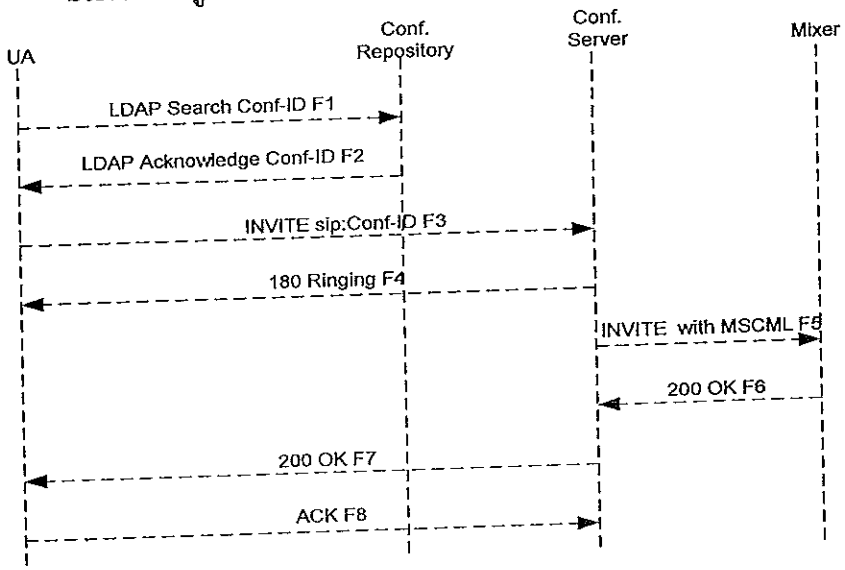
สำหรับกรณีแรกผู้ที่ต้องการจะเข้าร่วมการประชุมจะทำการค้นหาข้อมูลของ Conference URI จาก Conference Repository โดยผ่านทาง LDAP ผู้เข้าที่ต้องการเข้าร่วมการประชุมสามารถเข้าร่วมการประชุมโดย Join ไปยัง Conference URI ที่ได้รับมาจาก Conference Repository

Group finding via LDAP signal flows

ในรูปที่ 4-16 จะแสดงถึงรูปแบบของสัญญาณในการใช้ความสามารถของ Group finding โดยผ่านทาง LDAP เพื่อทำการเข้าร่วมการประชุม โดยมีขั้นตอนดังนี้

- เริ่มต้น UA จะทำการค้นหาข้อมูลของ Conf-ID จาก Conference Repository โดยใช้โปรโตคอล LDAP F1
- UA จะทำการส่งสัญญาณ INVITE F3 ไปยัง Conference Server ทันทีเมื่อได้รับสัญญาณตอบกลับจาก Conference Repository ที่ประกอบไปด้วย Conf-ID เพื่อทำการเริ่มการประชุม

- Conference Server ทำการตอบรับด้วยสัญญาณ 180 RINGING
- หลังจากนั้น Conference Server ทำการติดต่อไปยัง MIXER โดยใช้สัญญาณ INVITE F5 ที่ประกอบไปด้วยข้อมูลของ MSCML เพื่อเริ่มการทำงาน Mixer
- Mixer ทำการตอบรับด้วยสัญญาณ 200 OK หลังจากนั้น Conference Server เข้าสู่สถานะพร้อมที่จะทำการประชุม



รูปที่ 4-16 Group Finding (Via LDAP) signaling flow

```
INVITE sip:conf=123@mixer.cnrlab.net SIP/2.0
Via: SIP/2.0/UDP coe.psu.ac.th;branch=z9hG4bKa38s
Max-Forwards: 70
To: sip:conf=123@mixer.cnrlab.net
From: <sip:conf-id@sip.cnrlab.net>;tag=1fdf33
Call-ID: lk4343kl34
CSeq: 1 INVITE
Contact: <sip:conf-id@sip.cnrlab.net>
Content-Type: application/mediaservercontrol+xml
Content-Length: 211

<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
```

```
<request>
  <configure_conference reservedtalkers="120"
    reserveconfmedia="yes"/>
</request>
</MediaServerControl>
```

รูปที่ 4-17 An example of the INVITE request F9 that contains MSCML data for specifying the Mixer.

- ผ่านทาง HTTP

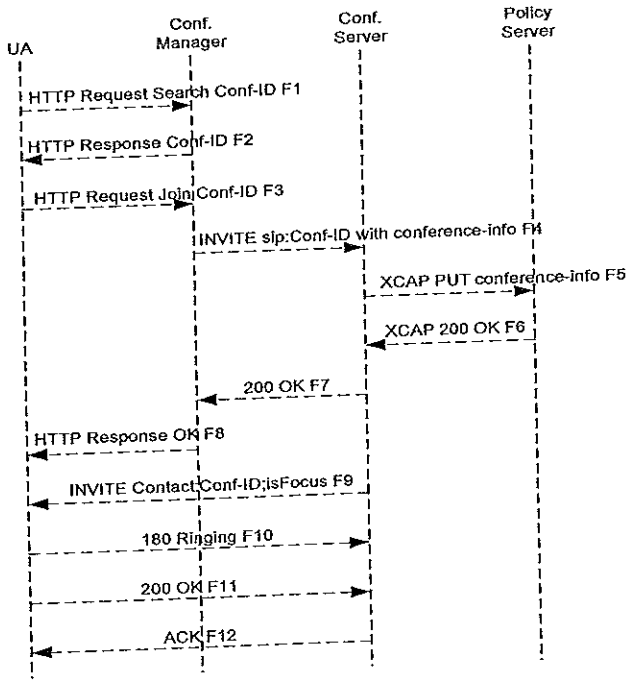
สำหรับกรณีนี้ผู้ที่ต้องการเข้าร่วมการประชุมจะทำการค้นหาข้อมูลของการประชุมผ่านทาง Web browser จากนั้น Conference Manager จะการคืนค่าเป็น link ของ Conference URI ผู้ที่ต้องการเข้าร่วมการประชุมสามารถที่จะเข้าร่วมการประชุมได้โดยการ Click ไปยัง link (Click & Talk) ที่ได้รับมาเพื่อ Join ไปยังการประชุม

Conference Manager จะทำการสร้างการติดต่อไปยัง Conference Server โดย Conference Server จะทำการตรวจสอบสิทธิการเข้าร่วมการประชุมกับ Conference Policy Server ถ้าผู้ติดต่อมีสิทธิเข้าร่วมการประชุมได้ Conference Server จะแจ้งกลับไปยังผู้ที่ต้องการเข้าร่วมการประชุม

Group finding via HTTP signal flows

ในรูปที่ 4-18 จะแสดงถึงรูปแบบของสัญญาณในการใช้ความสามารถของ Group finding โดยผ่านทาง HTTP เพื่อทำการเข้าร่วมการประชุม โดยมีขั้นตอนดังนี้

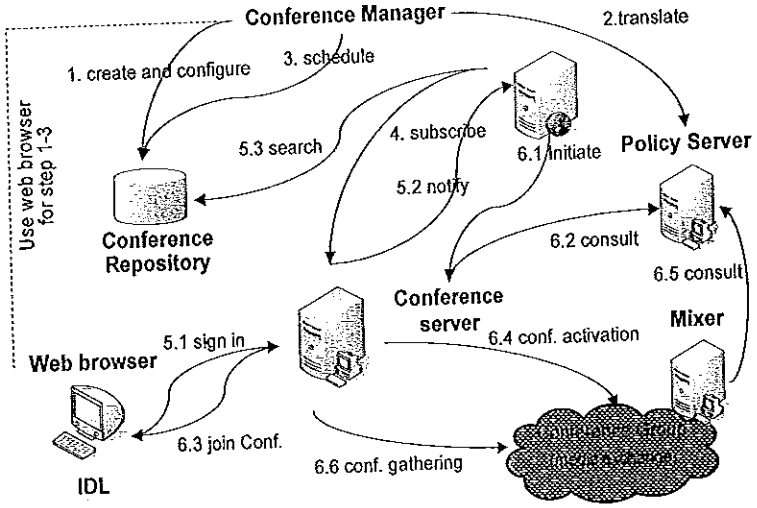
- เริ่มต้น UA จะทำการค้นหาข้อมูลของ Conf-ID จาก Conference Manager โดยใช้โปรโตคอล HTTP F1
- UA จะทำการส่งสัญญาณ HTTP F3 ไปยัง Conference Manager เพื่อร้องขอในการเข้าร่วมการประชุม
- Conference Manager ทำการส่งสัญญาณ INVITE F4 ที่ประกอบไปด้วยข้อมูล conference information ไปยัง Conference Server ทันทีเมื่อได้รับสัญญาณ HTTP F3 เพื่อเริ่มการประชุม
- หลังจากนั้น Conference Server ทำการจัดเก็บข้อมูล conference information ไปยัง Policy Server โดยใช้สัญญาณ XCAP PUT F5
- Conference Server ทำการส่งสัญญาณ INVITE F8 ไปยัง UA เพื่อทำการเริ่มการประชุม



รูปที่ 4-18 Group Finding (Via HTTP) signaling flow

4.3.2.4 Scheduled Conference

สำหรับกรณีนี้จะการกำหนดรายละเอียดต่าง ๆของการประชุมไว้ล่วงหน้าก่อนที่จะเริ่มการประชุม โดยจะมีการตั้งเวลาของการประชุมไว้ ซึ่งเมื่อถึงเวลาการประชุม Conference Server จะทำการเริ่มการติดต่อให้โดยอัตโนมัติดังแสดงในรูปที่ 4-19



รูปที่ 4-19 Scheduled Conference Scenario

สำหรับขั้นตอนการทำงานจะมีลำดับดังนี้

1. ขั้นแรกผู้สร้างการประชุมจะทำการกำหนดข้อมูลรายละเอียดต่าง ๆของการประชุมผ่าน Web browser ไปยัง Conference Manager
2. หลังจากที่ Conference Manager ได้รับข้อมูลรายละเอียดดังกล่าวแล้ว Conference Manager ก็จะทำการจัดเก็บข้อมูลเหล่านั้นไปยัง Conference Repository
3. Conference Manager จะทำการนำข้อมูลที่ผู้สร้างการประชุมได้กำหนดไว้ส่งต่อไปยัง Policy Server
4. จากนั้นข้อมูลของผู้เข้าร่วมการประชุมจะถูกจัดเก็บและรอเวลาที่จะเริ่มการประชุม
5. ก่อนเริ่มการประชุม Conference Manager จะทำการ Subscribe ไปยัง Conference Server เพื่อรับการสถานะของหัวหน้าการประชุม (Moderator) ว่าได้ติดต่อมายัง Conference Server แล้วหรือยัง เพื่อที่จะได้ทำการเริ่มการประชุม

6.1 ผู้ที่เป็นหัวหน้าการประชุมได้ทำการเข้าสู่ระบบ

6.2 Conference Server จะทำการแจ้งสถานการณ์เข้าสู่ระบบของ Moderator ไปยัง Conference Manager

6.3 Conference Manager จะทำการตรวจสอบข้อมูลของ Moderator กับ Conference Repository ว่ามีอยู่หรือไม่

7.1 ถ้าตรวจสอบแล้วพบว่ามีข้อมูลของ Moderator Conference Manager จะทำการสร้างการประชุมโดยติดต่อไปยัง Conference Server เพื่อเริ่มการประชุม

7.2 Conference Server จะติดต่อไปยัง Policy Server เพื่อตรวจสอบข้อมูล Policy ต่าง ๆที่จะต้องใช้ในการควบคุมการประชุม

7.3 Conference Server จะทำการส่งสัญญาณ INVITE ไปยัง Moderator เพื่อเริ่มการประชุม

7.4 ถ้าหาก Moderator ทำการตอบรับสัญญาณ INVITE การประชุมจะได้รับการ Activate ทันที

7.5 ขณะเดียวกัน Mixer จะทำการติดต่อไปยัง Policy Server เพื่อร้องขอ Media Policy

7.6 Conference Server จะทำการส่งสัญญาณไปเชิญผู้เข้าร่วมการประชุมคนอื่น ๆเพื่อเข้ามาร่วมการประชุม

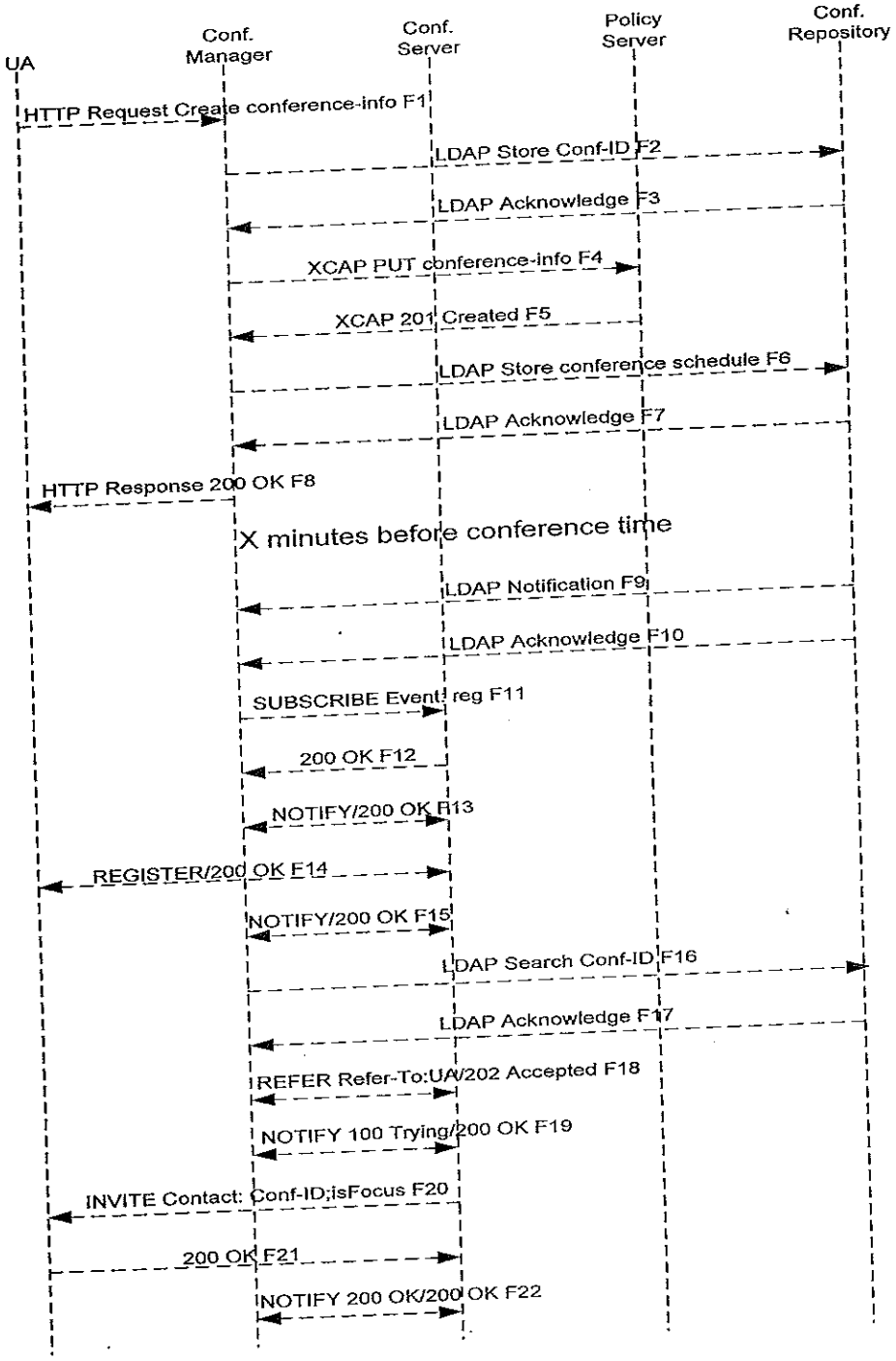
Schedule Conference signal flows

ในรูปที่ 4-20 จะแสดงถึงรูปแบบสัญญาณของ Schedule Conference โดยมีขั้นตอนดังนี้

- เริ่มต้น UA จะทำการส่งสัญญาณ HTTP F1 ไปยัง Conference Manager เพื่อทำการสร้างข้อมูล conference information
- หลังจากนั้น Conference Manager จะทำการจัดเก็บข้อมูลของ Conf-ID ไปยัง Conference Repository โดยใช้สัญญาณ LDAP F2 และทำการจัดเก็บ

ข้อมูล policy โดยส่งสัญญาณ XCAP PUT F4 พร้อมกับข้อมูล policy ไปยัง Conference Policy Server

- Conference Manager ทำการตั้งเวลาเริ่มการประชุม โดยกำหนดไปยัง Conference Repository ผ่านทางสัญญาณ LDAP F6
- เมื่อถึงเวลาที่กำหนด Conference Repository ทำการแจ้งเตือนไปยัง Conference Manager โดยใช้สัญญาณ LDAP F9 เพื่อเริ่มทำการประชุม
- Conference Server จะทำการส่งสัญญาณ NOTIFY F15 ไปยัง Conference Manager เมื่อ UA ทำการลงทะเบียนกับ Conference Server
- หลังจากนั้น Conference Manager ทำการค้นหา Conf-ID จาก Conference Repository และทำการส่งสัญญาณ REFER ไปยัง UA เพื่อเริ่มทำการประชุม

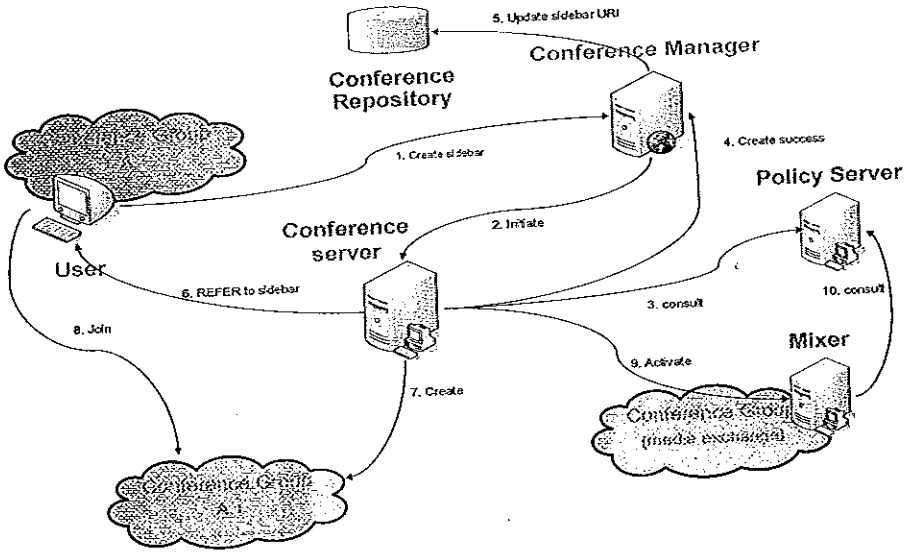


รูปที่ 4-20 Schedule Conferencing signaling flow

4.3.2.5 Sidebar Conference Call

สำหรับในการประชุมโดยทั่วไป บางครั้งอาจจะมีการแบ่งกลุ่มการประชุมย่อยเพื่อหารือกันเป็นการส่วนตัวหรือหารือในประเด็นย่อย นอกเหนือจากหัวข้อในการประชุมหลัก ซึ่งการประชุมย่อยนั้น

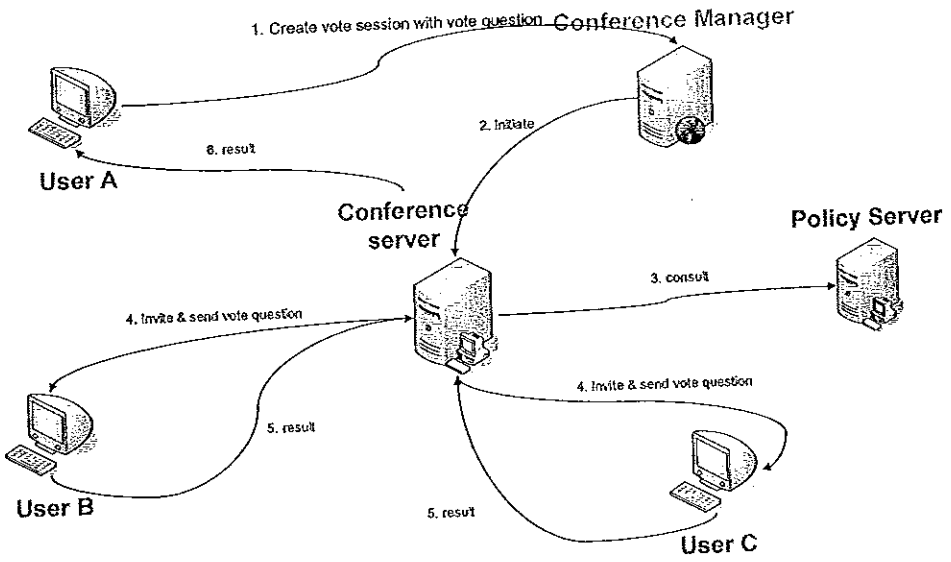
จำเป็นจะต้องมีผู้เข้าร่วมการประชุมคนใดคนหนึ่งเป็นสมาชิกของการประชุมหลักอยู่ก่อน เพื่อทำการสร้างและควบคุมการประชุมย่อย โดยในการประชุมย่อยนั้นเราสามารถที่จะชวนบุคคลอื่นที่ไม่ได้อยู่ในการประชุมหลักเข้ามาร่วมการประชุมย่อยได้ แต่ผู้ที่เข้าร่วมการประชุมใหม่นั้นจะไม่สามารถที่จะเข้าร่วมการประชุมหลักได้จนกว่าจะได้รับอนุญาตจากประธานการประชุมหลักก่อน ซึ่งรูปแบบการประชุมจะเป็นดังแสดงในรูปที่ 4-21



รูปที่ 4-21 Sidebar Conference Call

4.3.2.6 Voting in Conference

ในการประชุมส่วนใหญ่ ระหว่างการประชุมบางครั้งจำเป็นที่จะต้องให้สมาชิกที่อยู่ในการประชุมทำการโหวตลงมติความเห็นกันเพื่อใช้ในการตัดสินใจต่าง ๆ ซึ่งได้นำเสนอรูปแบบของการสร้างการติดต่อเพื่อทำการโหวต ซึ่งจะมีรูปแบบดังแสดงรูปที่ 4-22



รูปที่ 4-22 Voting in Conference

4.4 สรุป

ในบทนี้ได้นำเสนอแนวคิดและการออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP โดยในการออกแบบได้นำแนวคิดที่จากการศึกษาในบทที่ 2 และบทที่ 3 ซึ่งพบว่ามีปัญหาในเรื่องของรายละเอียดของโปรโตคอลที่ใช้ในการติดต่อระหว่างองค์ประกอบแต่ละส่วนที่ยังไม่ได้มีการระบุไว้ นอกจากนี้ส่วนของการทำงานของระบบการประชุมยังขาดในเรื่องของรายละเอียดสัญญาณที่ใช้ในการติดต่อทั้งหมด

ดังนั้นจึงเป็นประเด็นหนึ่งที่ทำให้เกิดแนวคิดที่จะออกแบบและพัฒนาระบบการประชุมแบบสื่อประสมสำหรับโปนโตคอล SIP โดยแนวคิดที่นำเสนอได้มีการเพิ่มองค์ประกอบเพิ่มเติมสองส่วนคือ Conference Manager และ Conference Repository ซึ่งเข้ามาช่วยประสานการทำงานและลดภาระการทำงานให้กับองค์ประกอบต่าง ๆ นอกจากนี้ยังได้นำเสนอถึงรายละเอียดของรูปแบบสัญญาณที่ใช้ในการติดต่อระหว่างองค์ประกอบแต่ละส่วน โดยการนำแนวคิดและข้อมูลของโปรโตคอลต่าง ๆที่ได้ศึกษาและเปรียบเทียบไว้ในบทที่ 3 มาใช้งานระบบการประชุมที่ได้ทำการออกแบบ ซึ่งทำให้สามารถที่จะออกแบบรูปแบบการติดต่อใหม่ ๆ นอกเหนือจากรูปแบบการติดต่อแบบปกติ

สำหรับในส่วนของการนำระบบการประชุมที่ออกแบบทั้งหมดมาพัฒนาเพื่อพิสูจน์ความถูกต้องของแนวคิดและวัดประสิทธิภาพการทำงานเมื่อเทียบกับระบบการประชุมแบบอื่น ซึ่งรายละเอียดจะขอกล่าวในบทต่อไป

บทที่ 5
การพัฒนาโปรแกรมประยุกต์เพื่อการทดสอบ

ในบทนี้จะกล่าวถึงการพัฒนาระบบการประชุมบนโปรโตคอล SIP ซึ่งจะเห็นได้ว่าในระบบการประชุมจำเป็นที่จะต้องประกอบไปด้วยองค์ประกอบต่าง ๆหลายส่วนที่เชื่อมต่อกัน ดังนั้นในการพัฒนาเราจึงจำเป็นที่จะต้องเลือกเครื่องมือที่เหมาะสมมาใช้ในการพัฒนา เพื่อให้เกิดประสิทธิภาพในการทำงานของระบบมากที่สุด โดยจะกล่าวถึงเครื่องมือที่จะนำมาใช้ในการพัฒนา ขอบเขตของการพัฒนา การออกแบบการพัฒนาโปรแกรมประยุกต์ และตัวอย่างของโปรแกรมประยุกต์ที่ได้พัฒนาขึ้นมา

5.1 เหตุผลที่ต้องมีโปรแกรมประยุกต์เพื่อการทดสอบ

จากสถาปัตยกรรมการประชุมที่ได้ทำการศึกษาและออกแบบไว้ในบทที่ 4 ซึ่งพบว่ามีหลายองค์ประกอบหลายส่วนอยู่ในระบบการประชุม ซึ่งเพื่อเป็นการพิสูจน์ว่าระบบที่ได้ทำการออกแบบนั้นสามารถรองรับการทำงานได้จริงตามที่ออกแบบไว้ จึงจะเป็นต้องพัฒนาระบบการประชุมที่ได้ทำการออกแบบไว้รวมถึงทดสอบการทำงานร่วมกันขององค์ประกอบต่าง ๆ ว่าสามารถทำงานร่วมกันได้อย่างถูกต้องตามที่ออกแบบไว้หรือไม่ โดยในการพัฒนาได้เลือกใช้เครื่องมือรวมถึงเทคโนโลยีที่มีความยืดหยุ่นในการที่จะนำระบบที่พัฒนาไปต่อยอดเพื่อขยายความสามารถต่อไป โดยจะต้องมีความเสถียรและรองรับการใช้งานจากผู้ติดต่อที่มีจำนวนมากได้ ซึ่งในหัวข้อต่อไปจะขอกล่าวถึงเครื่องที่จะนำมาใช้ในการพัฒนาโปรแกรมยุกต์ต่อไป

5.2 เครื่องมือที่ใช้ในการพัฒนา

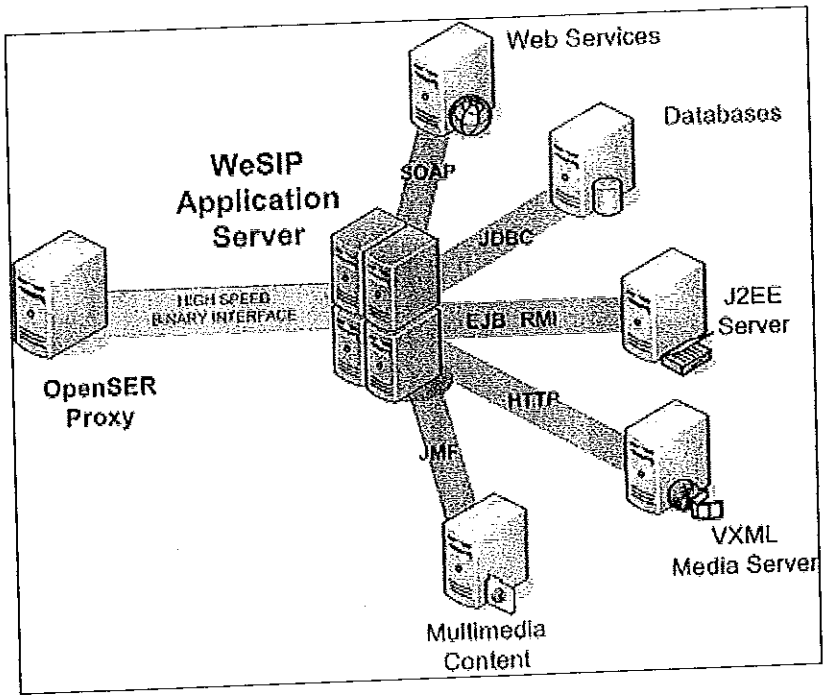
จุดประสงค์ของการพัฒนาโปรแกรมประยุกต์ต้นแบบนี้ก็เพื่อที่จะนำมาใช้ในการทดสอบการทำงานขั้นตอนต่าง ๆ ของโครงสร้างสถาปัตยกรรมการประชุมที่ได้ทำการออกแบบไว้จากบทที่ 4 โดยโปรแกรมจะถูกพัฒนาขึ้นโดยใช้ภาษาโปรแกรมจาวา ซึ่งภาษาจาวานี้เป็นภาษาโปรแกรมเชิงวัตถุที่ถูกพัฒนาขึ้นมาโดย Sun Microsystems ตั้งแต่ปี พ.ศ. 2534 จากข้อดีของภาษาจาวาที่เป็นภาษาโปรแกรมเชิงวัตถุ สามารถทำงานได้โดยไม่ขึ้นอยู่กับแพลตฟอร์ม (สถาปัตยกรรมของเครื่องคอมพิวเตอร์และระบบปฏิบัติการ) และมีไลบรารีที่สามารถนำมาใช้งานร่วมที่หลากหลาย ทำให้ภาษาจาวาถูกเลือกนำมาใช้ในการพัฒนาโปรแกรมประยุกต์ต้นแบบ

โดยในขั้นตอนของการพัฒนา เนื่องระบบการประชุมแบบสื่อประสมทำงานโดยใช้โปรโตคอล SIP และโปรโตคอล HTTP เป็นหลัก จึงได้ทำการศึกษาค้นคว้าและเลือกเครื่องมือในการพัฒนาที่สามารถใช้งานโปรโตคอลทั้งสองทำงานร่วมกันได้อย่างสมบูรณ์ และยังสามารถพัฒนาโดยใช้ภาษาจาวาได้ ซึ่งมีเครื่องมือต่าง ๆ ดังนี้

- OpenSIPS [24] เป็น Open Source SIP Server ที่ได้รับการพัฒนาต่อยอดมาจากโปรเจ็ค SER [25] (SIP Express Router) โดย OpenSIPS มีความสามารถหลักคือทำหน้าที่เป็น SIP Proxy Server เพื่อทำหน้าดูแลสัญญาณ SIP ต่าง ๆที่ถูกส่งมาจากผู้ติดต่อต่าง ๆ นอกจากนี้ OpenSIPS ยังมีโมดุลต่าง ๆ มากมายที่เข้ามาเสริมการทำงานให้หลากหลายมากยิ่งขึ้น โดย OpenSIPS ได้รับการออกแบบมีเสถียรภาพอย่างมากในการรับการเชื่อมจำนวนมากจากผู้ติดต่อทั้งหลาย โดยสามารถรับการทำ Call Setup ได้มากถึง 5000 call setup ต่อวินาที ซึ่งมีความจำเป็นอย่างมากในระบบการประชุม จึงทำให้ OpenSIPS ได้รับความนิยมอย่างกว้างขวางในปัจจุบัน

- J2EE [26] คือ กลุ่มของชุดคำสั่งภาษาจาวา ที่จำเป็นสำหรับการใช้ภาษาจาวาสร้างโปรแกรมระดับองค์กร ซึ่งภาษาจาวาเป็นภาษาที่รวมเทคโนโลยีที่จำเป็นทั้งหมดสำหรับการเขียนโปรแกรม ในระดับองค์กรเข้าไว้ในตัวมันเลยตั้งแต่ต้น ชุดคำสั่งที่อยู่ในชุดของ J2EE เรียกได้ว่า ครอบคลุมทุกสิ่งทุกอย่างที่โปรแกรมระดับองค์กรต้องการ ดังนั้นการเขียนโปรแกรมระดับองค์กรด้วย J2EE จึงไม่จำเป็นต้องมีซอฟท์แวร์สนับสนุนใด ๆอีก
- WeSIP [27] เป็นเทคโนโลยีที่จะทำให้ OpenSIPS สามารถใช้งานร่วมกับเทคโนโลยี J2EE โดยผนวกความสามารถของมาตรฐาน SIP เข้ากับมาตรฐาน HTTP ซึ่งทำให้ผู้พัฒนาสามารถใช้ความฟังก์ชันการทำงานเดิมที่หลากหลายของ OpenSIPS เช่น Registrar, Accounting, NAT Traversal เป็นต้น และในขณะเดียวกันก็สามารถที่จะพัฒนาความสามารถอื่น ๆเพิ่มเติมเข้าไปยังระบบได้ ซึ่งในที่นี่หมายถึงส่วนประกอบต่าง ๆของระบบการประชุม เช่น Focus, Notification Server, Mixer Control เป็นต้น

  เนื่องจาก WeSIP ได้ใช้เทคโนโลยี J2EE ในการพัฒนาจึงทำให้ผู้พัฒนาสามารถใช้ภาษา Java เพื่อเรียกใช้ความสามารถต่าง ๆของ J2EE ออกมาใช้งานได้ เช่น SOAP, EJB, JDBC เป็นต้น เพื่อนำมาสร้างเป็น Application Server ที่สามารถทำงานร่วมกับ OpenSIPS ได้ ดังแสดงในรูปที่ 5-1



รูปที่ 5-1 การเชื่อม WeSIP กับ OpenSIPS ในรูปแบบต่าง ๆ [27]

- Jain-SIP [28] เป็นไลบรารีที่ถูกพัฒนาขึ้นมาเพื่อใช้จัดการเกี่ยวกับสัญญาณ SIP โดยไลบรารีจะอ้างอิงตามมาตรฐาน SIP (RFC 3261) ที่องค์กร IETF ได้กำหนดไว้ การที่ Jain-SIP ถูกพัฒนาขึ้นมานั้นเพื่อเป็นมาตรฐานในการเชื่อมต่อของภาษาจาวาในการจัดการกับโปรโตคอล SIP ซึ่งจะไลบรารีจะจัดเตรียมตั้งแต่เรื่องของการสร้างสัญญาณ ช่องทางการติดต่อ การรับและส่งสัญญาณ การแปลความหมายของสัญญาณ รวมถึงการแจ้งไปยังโปรแกรมประยุกต์ถึงเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ทั้งนี้เพื่อให้ผู้พัฒนาโปรแกรมประยุกต์มีความสะดวกและง่ายต่อการพัฒนา
- SEMS (SIP Express Media Server) [29] เป็น Application Server ที่ทำงานอยู่บนมาตรฐาน SIP โดยมีความสามารถในการจัดการ Media Stream ต่างๆ เช่น IVR, Announcement, Conference เป็นต้น ซึ่งระบบของ SEMS ได้ถูกออกแบบให้สามารถจะนำไปพัฒนารวมกับระบบต่างๆเพื่อใช้สำหรับจัดการ Media Stream โดย SEMS รองรับการใช้งาน Media Stream ที่อยู่ในรูปแบบของเสียง ซึ่ง SEMS มีความสามารถในการที่จะทำการรวม Media Stream เข้าด้วยกัน (Mixing) ก่อนที่จะส่งไปยังผู้ติดต่อต่างๆ จึงเป็นทางเลือกหนึ่งที่น่าสนใจที่จะนำ SEMS มีพัฒนาเป็น Mixer ใช้กับระบบการประชุม นอกจากนี้ SEMS ยังรองรับ codec เสียงต่างๆ เช่น G711, GSM ,iLBC ,Speex ,ADPCM เป็นต้น
- MySQL เป็นระบบจัดการฐานข้อมูล มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่งเอสคิวแอล (SQL = Structured Query Language) เป็นเครื่องมือสำหรับเก็บข้อมูล เพื่อให้ได้ระบบงานที่

รองรับความต้องการของผู้ใช้ เช่นทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษาพีเอชพี ภาษาเอเอสพีดอทเน็ต หรือภาษาเจเอสพี เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษาวิชวลเบสิกดอทเน็ต ภาษาจาวา หรือภาษาซีชาร์ป เป็นต้น โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย และเป็นระบบฐานข้อมูลโอเพนซอร์ทที่ถูกนำไปใช้งานมากที่สุด

5.3 ขอบเขตของการพัฒนาโปรแกรมประยุกต์

การพัฒนาโปรแกรมประยุกต์เพื่อทดสอบการทำงานของสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสมจะมีขอบเขตในการพัฒนาดังต่อไปนี้

1. โปรแกรมประยุกต์ถูกพัฒนาขึ้นโดยใช้ภาษาโปรแกรมจาวาทั้งหมด
2. โปรแกรมประยุกต์ที่พัฒนาขึ้นจะมีการทำงานหลักอยู่ในส่วนของแม่ข่าย (Server side)
3. โปรแกรมประยุกต์ที่ถูกพัฒนาจะเป็นการทดสอบการทำงานของขั้นตอนต่าง ๆ ที่จะทำให้เกิดระบบการประชุมขึ้น ตามรูปแบบที่ได้ออกแบบไว้ในบทที่ 4
4. โปรแกรมประยุกต์จะฝั่งลูกข่ายที่ใช้ในการทดสอบระบบจะใช้ SIP User Agent ทั่วไปโดยจะทดสอบการร่วมการประชุมที่ใช้สื่อเป็นเสียงเท่านั้น

5.4 การออกแบบโปรแกรมประยุกต์

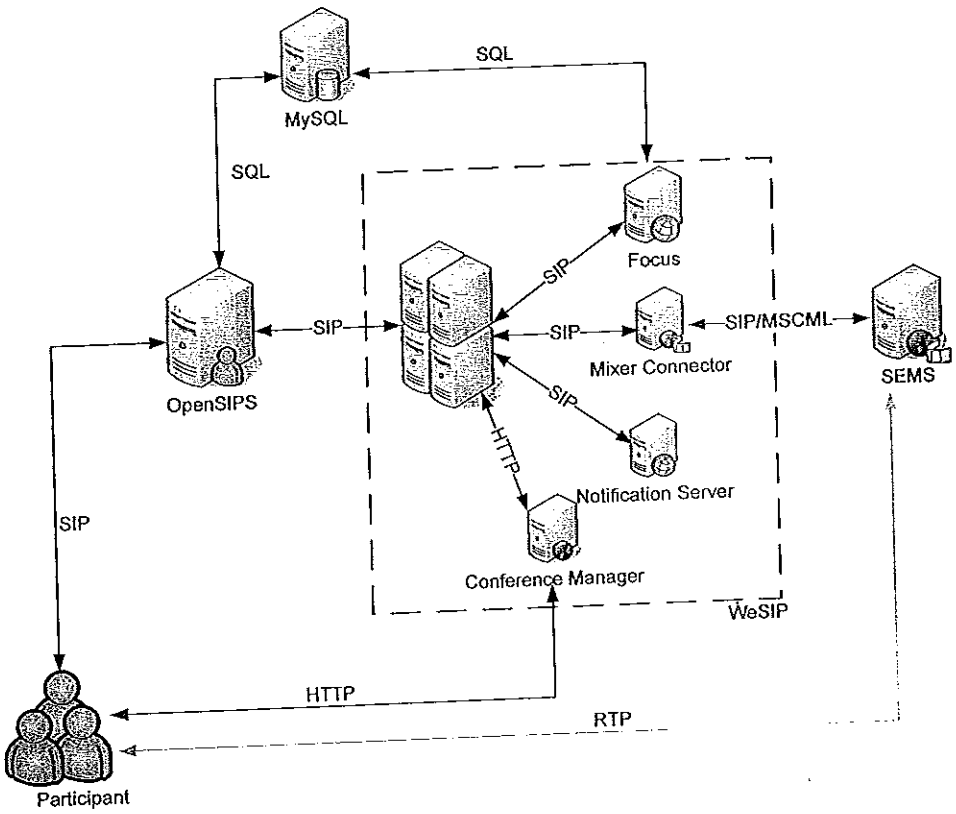
ในการออกแบบโปรแกรมประยุกต์เพื่อใช้ในการทดสอบระบบการประชุมที่ได้พัฒนาขึ้นมานั้น จำเป็นต้องคำนึงถึงรูปแบบการทำงานและสัญญาณของโปรแกรมประยุกต์ที่จะต้องสอดคล้องกับรูปแบบของระบบที่ได้ออกแบบไว้ในบทที่ 4 โดยในขั้นตอนการออกแบบระบบจะถูกแยกออกเป็นส่วน ๆตามองค์ประกอบที่ได้ออกแบบไว้ ซึ่งแต่ละองค์ประกอบจะยังคงทำงานร่วมกันได้อย่างได้ถูกต้องสอดตล้องกัน

ดังนั้นแนวคิดในการออกแบบจึงเป็นดังแสดงในรูปที่ 5-2 ซึ่งจะสังเกตได้ว่าในส่วนการติดต่อระหว่างผู้เข้าร่วมการประชุมจะถูกแบ่งเป็นสองวิธีคือ

- ใช้โปรโตคอล SIP เพื่อใช้สร้างการติดต่อในการเข้าร่วมระบบการประชุมโดยผ่านทาง OpenSIPS Server
- ใช้โปรโตคอล HTTP เพื่อใช้ในการกำหนดค่าการทำงานต่าง ๆในระบบการประชุม โดยผ่านทาง Conference Manager

ซึ่งการทำงานทั้งสองส่วนจะเป็นการทำงานที่มีความสัมพันธ์ระหว่าง OpenSIPS Server และเทคนิคของ WeSIPS Server โดยใช้แนวคิดที่ว่าจะให้ส่วนของ OpenSIPS Server ทำหน้าที่ใน

การจัดการเกี่ยวกับ SIP Dialog ทั้งหมดที่เกิดขึ้นเพื่อช่วยลดภาระการทำงานของส่วนอื่น ๆออกไป ซึ่งส่วนการทำงานอื่น ๆ ที่ได้ทำการพัฒนาจะมีแนวคิดในการออกแบบดังต่อไปนี้



รูปที่ 5-2 โครงการระบบการประชุมแบบสื่อประสมที่ได้ทำการพัฒนา

5.4.1 Focus

จะใช้แนวคิดและเทคนิคการพัฒนาของ SIP Servlet โดย Focus จะรับสัญญาณ SIP ที่ได้ถูกส่งมาจาก OpenSIPS ซึ่ง Focus จะทำการตรวจสอบกับข้อมูลที่อยู่จากฐานข้อมูลเพื่อตรวจสอบว่าผู้ติดต่อที่ติดต่อมานั้นมีสิทธิที่จะเข้าร่วมการประชุมได้หรือไม่ โดยการทำงานจะแบ่งออกเป็นโมดุลต่าง ๆดังนี้

5.4.1.1 Notification Server

ได้มีแนวคิดในการพัฒนารูปแบบการทำงานคล้ายกับ Presence Service ของมาตรฐาน SIP โดยการพัฒนาจะใช้เทคนิคในการสร้าง SIP Servlet เช่นเดียวการกับการสร้าง Focus แต่ในส่วนของ Notification Server จะตรวจสอบเฉพาะสัญญาณ Subscribe และ Notify เท่านั้น

สำหรับรูปแบบของข้อมูลที่จะถูกส่งในส่วนของ Notification Server จะเป็นไปตามรูปแบบของ Conference Event Package ดังแนวคิดที่นำเสนอในบทที่ 3

5.4.1.2 Mixer Connector

สำหรับส่วนของ Mixer Connector ได้ทำการพัฒนาโดยการใช้ SIP Servlet มาประยุกต์ร่วมกับโปรโตคอล MSCML ซึ่งทำงานอยู่โปรโตคอล SIP เช่นกัน โดยส่วนของ Mixer Connector จะถึงเรียกใช้งานทันทีเมื่อเริ่มทำการสร้างการประชุม ซึ่ง Conference Server จะทำการติดต่อไปยัง Mixer Server เพื่อร้องขอทรัพยากรสำหรับใช้ในสร้างการประชุม จากนั้นเมื่อมีผู้ติดต่อเข้ามายังการประชุม Focus จะทำการส่งข้อมูลผู้ที่เข้ามาติดต่อไปยัง Mixer Connector เพื่อสร้าง Session ระหว่างผู้ติดต่อนั้นกับการประชุม

5.4.1.3 Conference Manager

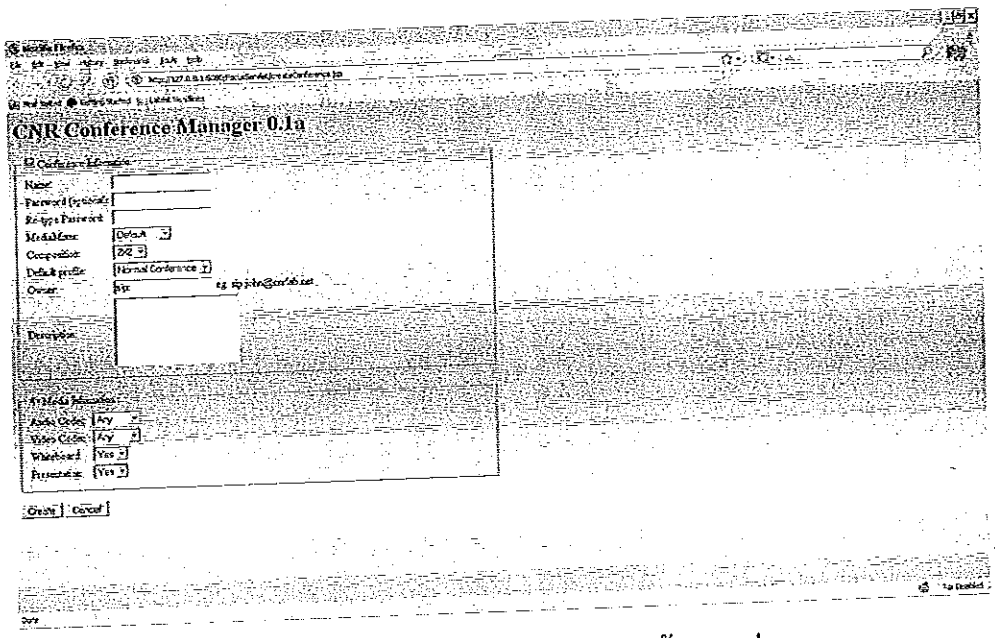
เป็นส่วนที่ติดต่อกับใช้อีกช่องทางหนึ่งเพื่อให้ผู้ใช้งานสะดวกในการสร้างการประชุมและกำหนดข้อบังคับ ๆต่างที่ใช้งานการประชุม โดยรูปแบบของ Conference Manager จะเป็น Web Application ซึ่งสามารถทำงานอยู่บนเทคโนโลยี WeSIP เช่นเดียวกับส่วนอื่น ๆ ทำให้ช่วยลดเวลาในการพัฒนารวมถึงขั้นตอนในการติดตั้งระบบ

5.4.2 Mixer

สำหรับในการพัฒนาส่วนของ Mixer ได้มีแนวคิดที่จะใช้โปรแกรมประยุกต์ที่ชื่อ SEMS เข้ามาใช้งานตรงส่วนเพื่อช่วยในการรวมข้อมูลสื่อประสมที่เป็นแบบชนิดข้อมูลเสียงซึ่งได้มีการแก้ไขการทำงานในส่วนของการติดต่อให้สามารถรองรับการทำงานในส่วนของโปรโตคอล MSCML ที่ถึงส่งมาโดย MixerConnector เพื่อใช้ในการควบคุมการส่งข้อมูลสื่อประสมที่เกิดขึ้นในการประชุม

จากแนวคิดในการออกแบบองค์ประกอบทั้งหมดที่จะมีในโปรแกรมประยุกต์ที่ได้กล่าวในหัวข้อนี้ ทำให้สามารถที่จะพัฒนาระบบการประชุมแบบสื่อประสมสำหรับโปรโตคอล SIP ได้ โดยส่วนต่อไปที่ต้องคำนึงถึงคือการทดสอบระบบที่ได้ทำการพัฒนาว่าได้ทำงานอย่างถูกต้องเป็นตามที่ได้ออกแบบไว้หรือซึ่งจะกล่าวในห้อข้อถัดไป

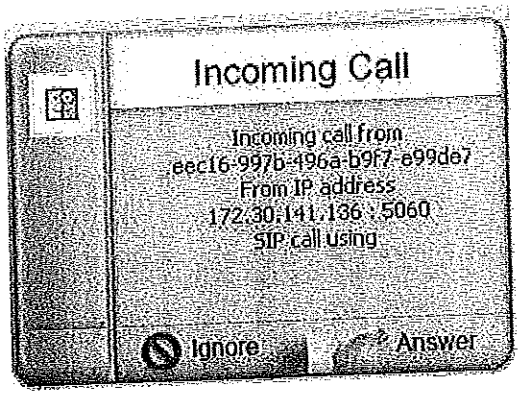
5.5 ตัวอย่างโปรแกรมประยุกต์



รูปที่ 5-3 ตัวอย่างโปรแกรมประยุกต์ส่วนการสร้างการประชุม

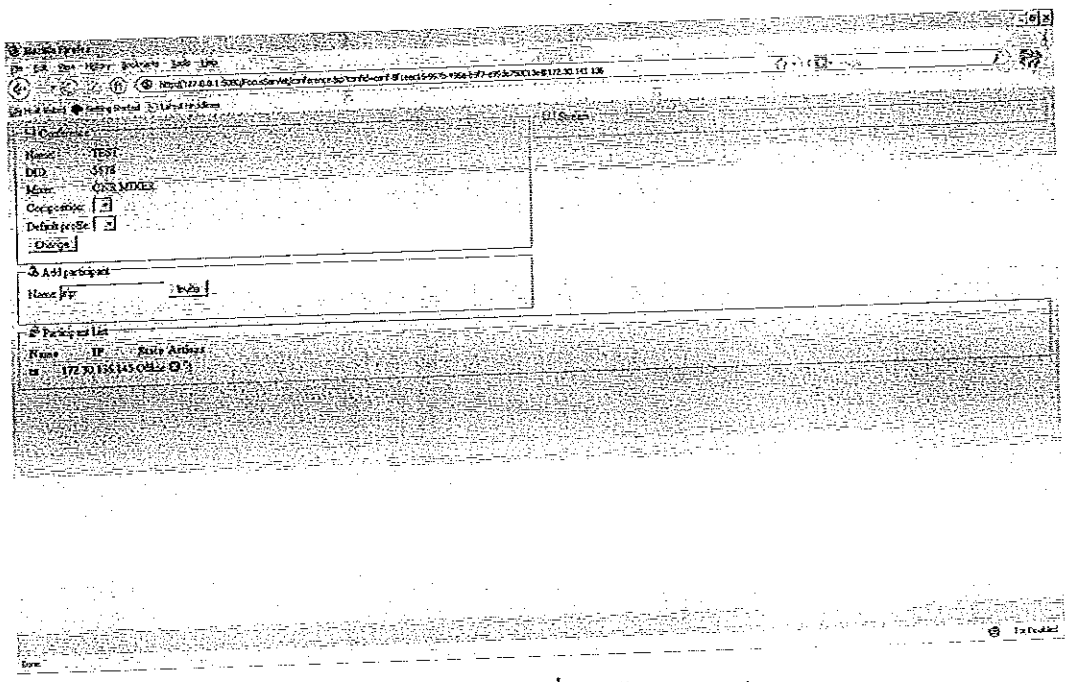
จากรูปที่ 5-3 เป็นตัวอย่างการทำงานในส่วนของ Conference Manager โดยจะมีข้อมูลที่ผู้ใช้สามารถกำหนดได้ต่าง ๆ ดังนี้ ชื่อที่การประชุม รหัสผ่านสำหรับเข้าร่วมการประชุม Mixer ที่ใช้ในการประชุม ประธานการประชุม รายละเอียดของสื่อประสมที่ใช้ในการประชุม

โดยผู้สร้างการประชุมได้ทำการสร้างการประชุมเป็นที่เรียบร้อยแล้วระบบจะทำการสร้างการติดต่อไปยังประธานการประชุมเพื่อสร้างการประชุมในขั้นต้นดังแสดงในรูปที่ 5-4



รูปที่ 5-4 ตัวอย่างการติดต่อมายังประธานการประชุมเพื่อเริ่มต้นการประชุม

หลังจากนั้นผู้สร้างการประชุมสามารถที่จะเพิ่มผู้ร่วมการประชุมอื่นเข้าไปได้โดยผ่านทาง Conference Manager ดังแสดงในรูปที่ 5-5



รูปที่ 5-5 ตัวอย่างเพิ่มผู้เข้าร่วมการประชุม

5.6 ฟังก์ชันการทำงานของโปรแกรมประยุกต์ต้นแบบ

โปรแกรมประยุกต์ต้นแบบที่ได้รับการพัฒนาขึ้นมา เพื่อใช้ในการทดสอบนี้จะมีฟังก์ชันการทำงานต่าง ๆ ดังนี้

1. ฟังก์ชันการทำงานในระบบการติดต่อสื่อสารแบบปกติบนโปรโตคอล SIP
2. ฟังก์ชันการทำงานในระบบสำหรับสร้างกาประชุมแบบสื่อประสม
3. ฟังก์ชันการทำงานในการติดต่อไปยัง Mixer
4. ฟังก์ชันการทำงานในรูปแบบของ Web Application สำหรับการควบคุมการประชุม
5. ฟังก์ชันการทำงานสำหรับติดต่อฐานข้อมูลเพื่อจัดเก็บ Conference Policy ต่าง ๆ
6. ฟังก์ชันการทำงานในส่วนของ Notification Server เพื่อแจ้งการทำงานสถานะการประชุมต่าง ๆ

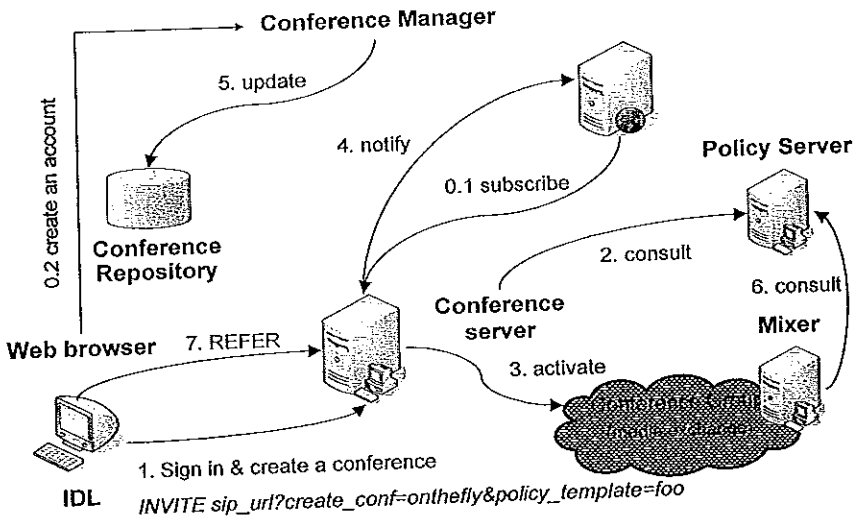
5.7 กระบวนการทดสอบ

ในการทดสอบระบบการประชุมแบบสื่อประสม จะทำการทดสอบการทำงานของส่วนประกอบต่าง ๆของระบบทั้งหมดว่าสามารถทำงานได้ตามกำหนดไว้หรือไม่ แต่เนื่องจากโปรแกรมประยุกต์ที่พัฒนาจะเน้นการทำงานในฝั่งของแม่ข่ายเป็นหลัก โดยไม่ได้มีมีการพัฒนาโปรแกรมประยุกต์ในฝั่งลูกข่าย

ดังนั้นผู้เขียนจึงเลือกโปรแกรมประยุกต์ที่ทำงานเป็นแบบ SIP User Agent รวมถึง SIP Device ต่าง ๆเข้ามาใช้ในการทดสอบระบบที่พัฒนาขึ้น โดยผู้เขียนได้เลือกโปรแกรมประยุกต์แล้วอุปกรต์ต่าง ๆ ซึ่งมีรายละเอียดดังต่อไปนี้

- โปรแกรมประยุกต์ SJPhone 1.65
- โปรแกรมประยุกต์ Windows Messenger 5.1

สำหรับในการทดสอบจะทำการทดสอบการเริ่มสร้างและทำการประชุมแบบทันทีทันใด โดยผู้สร้างการประชุมไม่ต้องทำการกำหนด Policy ที่ซับซ้อน แล้วผู้สร้างการประชุมทราบถึงรายละเอียดของผู้เข้าร่วมการประชุมที่แน่นอน



รูปที่ 5-6 รูปแบบทดสอบ

จากรูปที่ 5-6 เป็นรูปแบบการติดต่อที่แสดงการทำงานของระบบการสร้างการประชุม โดยมีลำดับการทำงานดังนี้

A. Conference Manager จะทำการ SUBSCRIBE ไปยัง Conference Server เพื่อตรวจว่า Conference Server พร้อมให้บริการหรือไม่

B. ในขณะเดียวกันผู้เข้าร่วมการประชุมก็จะทำการสร้าง Account ของตนเองเพื่อที่จะใช้ในการเข้าร่วมการประชุมในกรณีที่ผู้เข้าร่วมคนนั้นยังไม่ได้มี Account อยู่ก่อน โดยจะทำผ่านทาง Conference Manager

1. หลังจากที่ได้ทำการสร้าง Account เป็นที่เรียบร้อยแล้ว ก็จะทำการส่งสัญญาณ INVITE ไปยัง Conference Server โดยมีการระบุพารามิเตอร์เพิ่มเติมเพื่อเป็นการแจ้งให้ Conference Server ทราบว่าเป็นการสร้าง Conference ซึ่งพารามิเตอร์ที่ใช้นั้นคือ
   - create_conf=onthefly เป็นการระบุว่าให้ Conference Server สร้าง Conference ทันทีได้รับสัญญาณนี้
   - policy_template=foo เป็นการระบุถึง Conference Policy ที่จะใช้สำหรับการประชุมนี้

8. หลังจากที่ Conference Server ได้รับสัญญาณแล้ว Conference Server จะทำการตรวจสอบสิทธิกับ Policy Server ว่าผู้ที่ร้องขอในการสร้างการประชุมนั้นมีสิทธิที่จะสร้างได้หรือไม่

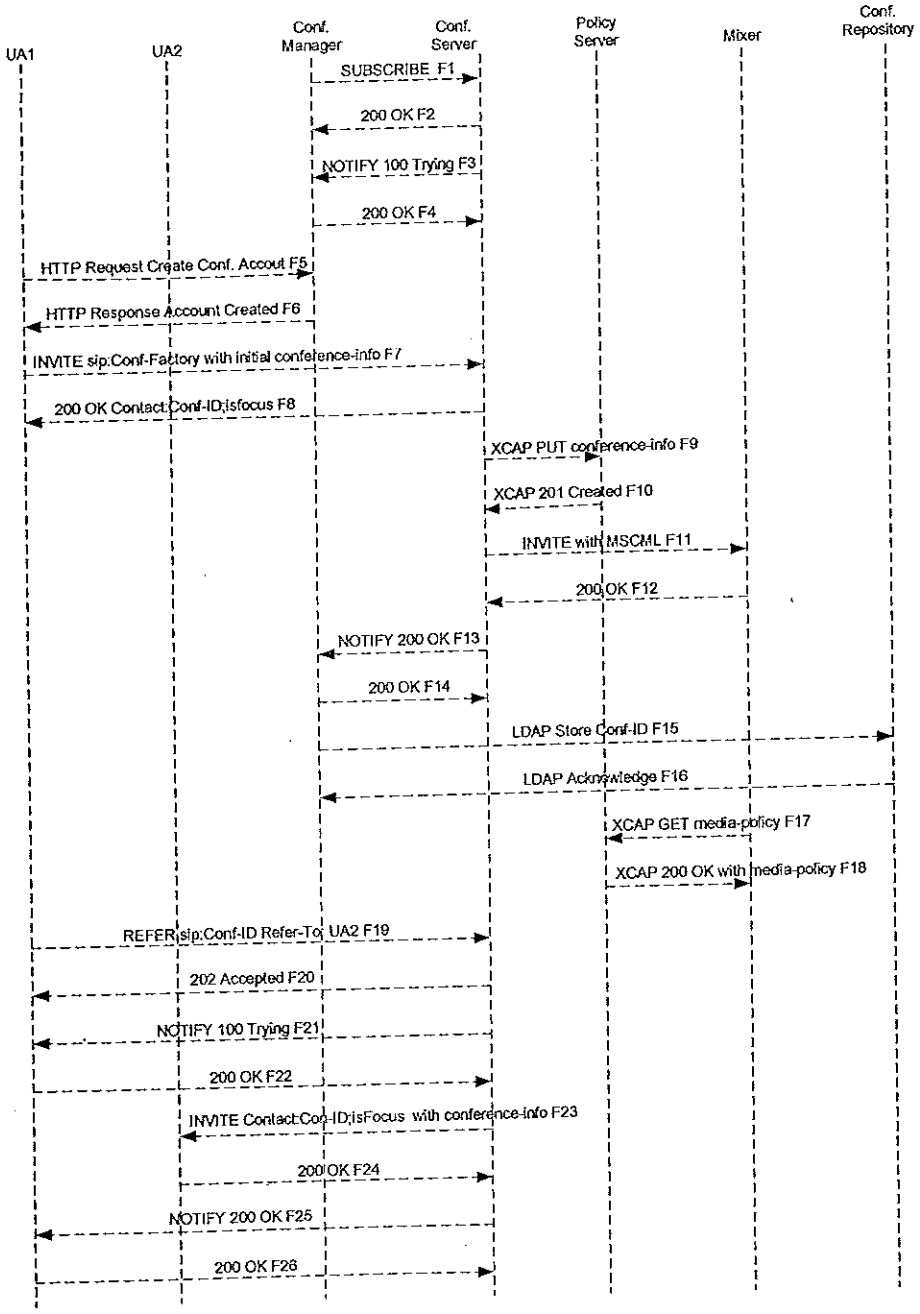
9. หากผู้ที่ทำการสร้างการประชุมมีสิทธิที่จะทำการสร้างการประชุมได้นั้น Conference Server จะทำการ Activate การประชุมโดยติดต่อไปยัง Mixer เพื่อแจ้งให้ Mixer เตรียมเริ่มการทำงาน

10. จากนั้น Conference Server จะทำการส่ง Notify ไปแจ้งให้ Conference Manager ทราบถึงรายละเอียดของ Conference Group ใหม่ที่ได้สร้างขึ้นมา
11. Conference Manger จะทำการเก็บข้อมูลของ Conference Group ที่ได้รับไว้ที่ Conference Repository เพื่อให้ผู้เข้าร่วมการประชุมคนอื่นสามารถค้นหาได้
12. ในส่วนของ Mixer จะทำการติดต่อไปยัง Policy Server เพื่อนำข้อมูลของ Media Policy ซึ่งจำเป็นต่อการควบคุม Media Stream ที่ส่งกันระหว่างการประชุม
13. หลังจากทำการกำหนดค่าเริ่มต้นต่าง ๆหรับการประชุมเป็นที่เรียบร้อยแล้ว ผู้สร้างการประชุมจะทำการเชิญผู้เข้าร่วมการประชุมคนอื่นเข้ามาร่วมการประชุม โดยการส่งสัญญาณ REFER ไปยัง Conference Server

ส่วนต่อไปจะอธิบายถึงรูปแบบและลำดับของสัญญาณต่าง ๆที่ใช้ในการสร้างการประชุมดังแสดงในรูปที่ 5-7

- เริ่มต้น Conference Manager ทำการส่งสัญญาณ SUBSCRIBE ไปยัง Conference Server หลังจากนั้น Conference Server จะทำการตอบรับด้วยสัญญาณ 200 OK และตามด้วยสัญญาณ NOTIFY เพื่อแจ้งว่ายังไม่มีการสร้างการประชุมเกิดขึ้น
- เมื่อ UA1 ทำการสร้าง Conference Account ไปยัง Conference Manager โดยผ่าน HTTP เป็นที่เรียบร้อยแล้ว UA1 สามารถที่จะทำการเริ่มการประชุมได้โดยการส่งสัญญาณ INVITE F7 ที่ประกอบไปด้วยข้อมูล SDP และข้อมูลเบื้องต้นของการประชุมนั้น ๆ ไปยัง Conference Server
- หลังจากนั้น Conference Server จะทำการตอบรับด้วยสัญญาณ 200 OK ที่มีรายละเอียดของ SDP ซึ่งใช้สำหรับการติดต่อ
- Conference Server นำข้อมูล Conference Policy ที่ได้จาก UA1 ไปจัดเก็บไว้ที่ Policy Server ผ่านทาง XCAP PUT F9
- Policy Server ทำการตอบรับด้วยสัญญาณ 201 Created กลับไปยัง Conference Server หลังจากนั้น Conference Server ทำการติดต่อไปยัง Mixer เพื่อแจ้งเริ่มทำการประชุมโดยใช้สัญญาณ INVITE F11 ที่ประกอบไปด้วยข้อมูล MSCML
- Conference Server ทำการส่งสัญญาณ NOTIFY F13 ไปยัง Conference Manager เพื่อแจ้งว่ามีการสร้างการประชุมขึ้นมาใหม่
- หลังจากนั้น Conference Manager ทำการจัดเก็บข้อมูลของ Conf-ID ไว้ที่ Conference Repository โดยใช้ LDAP F15

- Mixer ทำการร้องขอข้อมูล Media Policy จาก Policy Server เพื่อใช้ในการควบคุม media stream ระหว่างทำการประชุม โดยใช้ XCAP GET F17
- UA1 ทำการส่งสัญญาณ REFER F19 ไปยัง Conference Server เพื่อเชิญ UA2 เข้าร่วมการประชุม
- หลังจากนั้น Conference Server ทำการส่งสัญญาณ INVITE F23 ไปยัง UA2 ที่ประกอบไปด้วยข้อมูลเบื้องต้นของการประชุม



รูปที่ 5-7 รูปแบบสัญญาณที่ใช้ในการทดสอบ

5.8 ผลที่คาดว่าจะได้รับ

โปรแกรมประยุกต์ฝั่งลูกข่ายสามารถที่จะเข้าร่วมการประชุมได้อย่างสมบูรณ์ โดยจะได้รับการติดต่อจาก Focus เพื่อให้เข้าร่วมการประชุมและสามารถได้รับข้อมูลสื่อประสมที่เป็นแบบเสียง จาก Mixer ตามที่ได้กำหนดไว้อย่างถูกต้อง และเมื่อสิ้นสุดการประชุมผู้เข้าการประชุมจะจบการติดต่อได้อย่างถูกต้องตามมาตรฐาน SIP

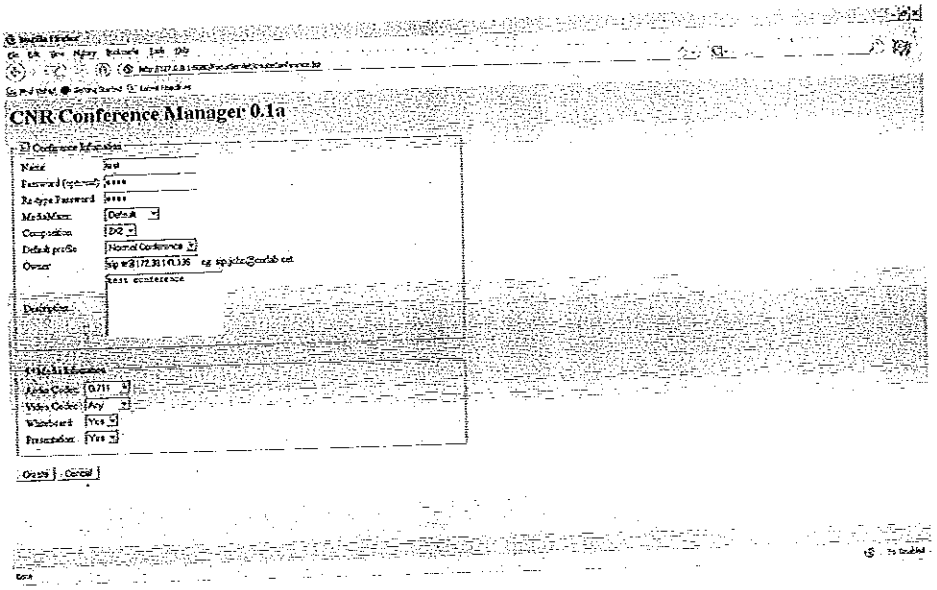
5.9 ผลการทดสอบ

จากกระบวนการทดสอบระบบจะพบว่าสามารถแบ่งการทำงานออกได้ 3 กระบวนการดังต่อไปนี้

5.9.1 การสร้างการประชุม

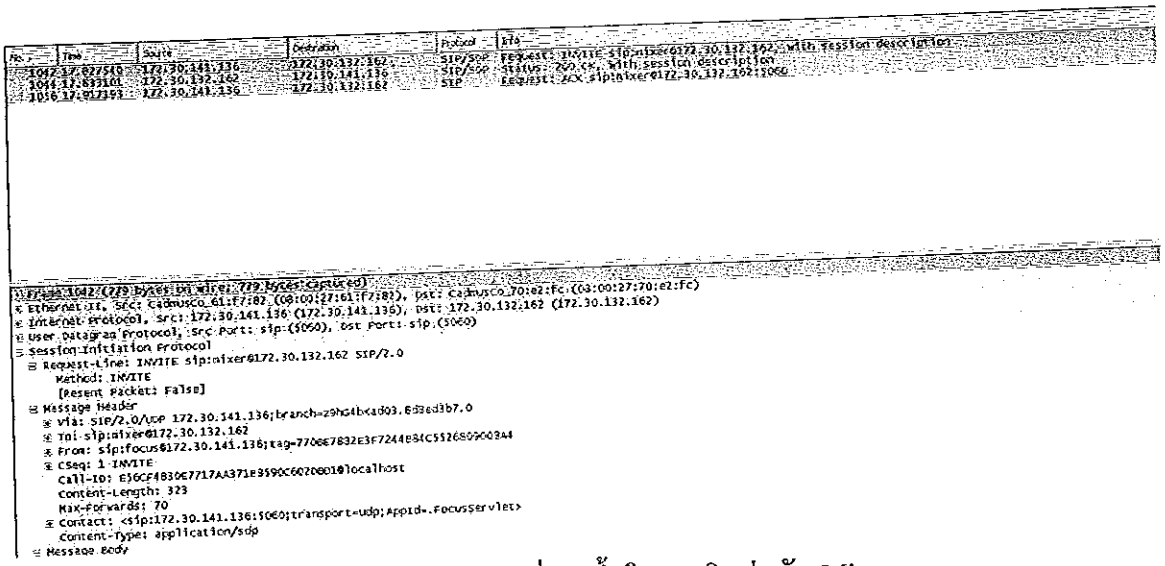
สำหรับการทดสอบสร้างการประชุมจะทดสอบในส่วนของ Web Application เพื่อตรวจสอบว่าสร้างทำการสร้างการประชุมและกำหนดค่าต่าง ๆสำหรับการประชุมได้

โดยจากผลการทดสอบจะเป็นดังแสดงในรูปที่ 5-8



รูปที่ 5-8 การใส่ค่าสำหรับเริ่มการประชุม

เมื่อทำการปุ่ม Create ระบบการประชุมจะถูกสร้างขึ้น โดยนำข้อมูลที่กำหนดไปเก็บไว้ในฐานข้อมูล และทำการติดต่อไปยัง Mixer เพื่อจองทรัพยากรสำรับที่ใช้ในการประชุม ซึ่งมีรูปแบบสัญญาณดังแสดงในรูปที่ 5-9 โดยในการดักจับสัญญาณได้ใช้งานโปรแกรม Wireshark เพื่อใช้ในการดักจับสัญญาณ ซึ่งเป็นโปรแกรมที่ได้รับการยอมรับในด้านประสิทธิภาพ การทำงานอย่างแพร่หลาย

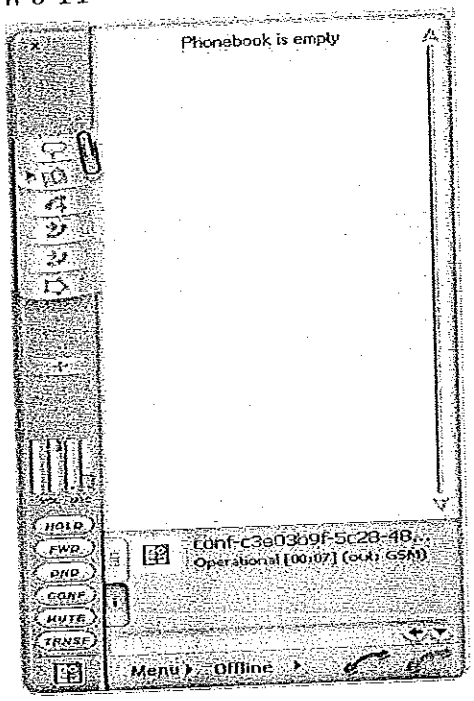


รูปที่ 5-9 ลำดับสัญญาณที่เกิดขึ้นในการติดต่อกับ Mixer

โดยในส่วนของโปรแกรมประยุกต์ก็จะแสดงการ Debug รูปแบบสัญญาณออกมาเช่นเดียวกันกับที่ดักจับกับโปรแกรม Wireshark ดังแสดงในรูปที่ 5-10 ซึ่งเป็นการพิสูจน์ได้ว่าระบบสามารถทำงานการทำงานในส่วนนี้ได้อย่างถูกต้อง

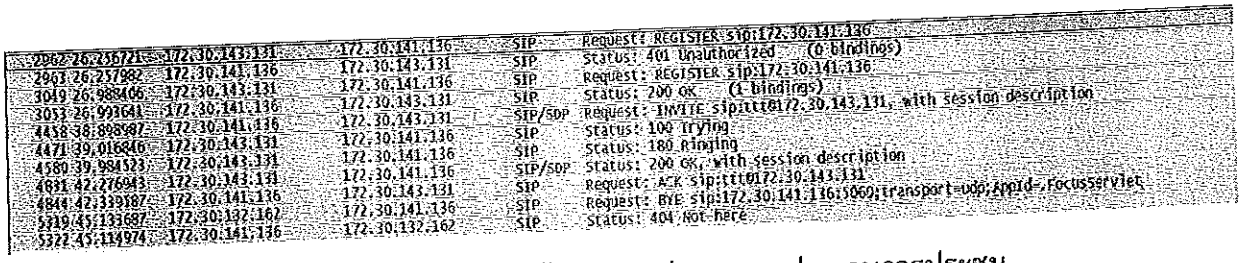
รูปที่ 5-10 ลำดับสัญญาณที่แสดงในระบบ WeSIP

จากนั้น ระบบจะทำการส่งสัญญาณ SIP Invite ไปยังประธานการประชุมตามที่กำหนดไว้ ซึ่งผลลัพธ์จะเป็นดังแสดงในรูปที่ 5-11



รูปที่ 5-11 แสดงการเข้าร่วมการประชุมของประธานการประชุม

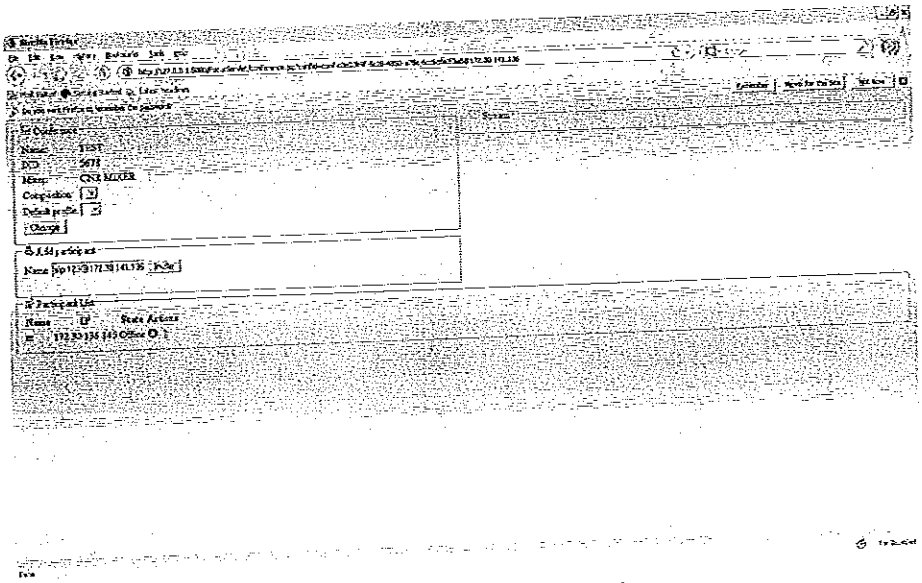
ซึ่งจะสังเกตได้ว่า SIP URI ที่แสดงจะเป็นอยู่ในรูปแบบของ Psudo random เพื่อความปลอดภัยในการป้องผู้แปลกปลอมเข้ามายังประชุม โดยรูปแบบสัญญาณ SIP ทั้งหมดที่เกิดขึ้นในขั้นตอนนี้จะแสดงในรูปที่ 5-12



รูปที่ 5-12 แสดงลำดับสัญญาณการเข้าร่วมการประชุมของประธานการประชุม

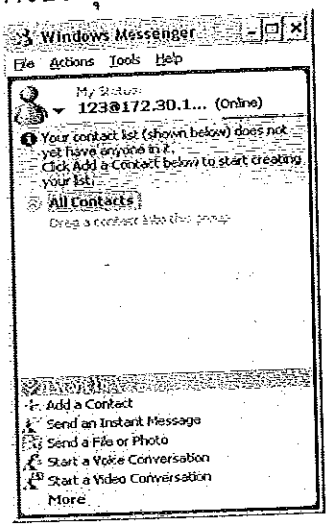
5.9.2 การเข้าร่วมประชุม

เมื่อทำกรสร้างการประชุมเสร็จเรียบร้อยแล้วในส่วนของ Web Application จะทำการเปลี่ยนเข้าสู่หน้าทำงานสำหรับเชิญผู้เข้าร่วมการประชุมคนอื่นเข้าสู่การประชุมดังแสดงในรูปที่ 5-13

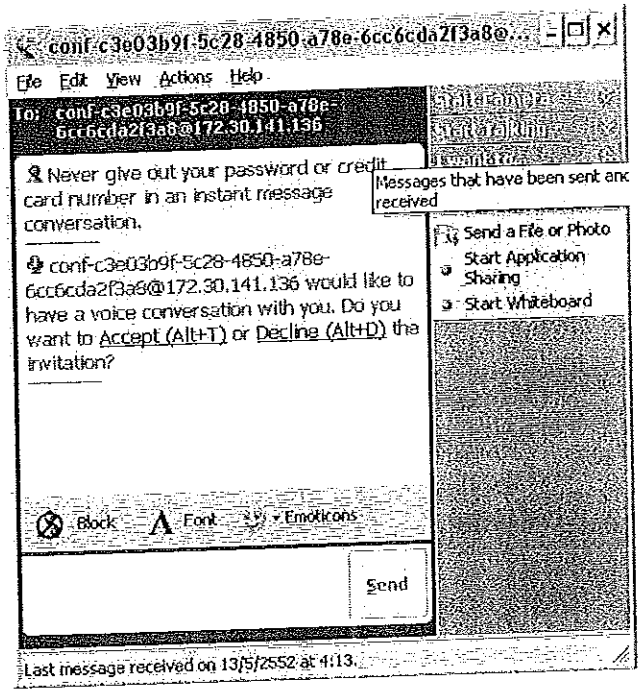


รูปที่ 5-13 หน้าแสดงส่วนการเชิญผู้เข้าร่วมอื่นเข้าสู่การประชุม

ซึ่งเมื่อผู้สร้างการประชุมทำการกดปุ่ม Invite ระบบจะทำการส่งสัญญาณไปยังผู้เข้าร่วมการประชุมตามที่กำหนดไว้ ดังในรูปที่ 5-13 ได้กำหนดทำให้การเชิญผู้ติดต่อที่มี SIP URI ชื่อว่า sip:123@172.30.141.136 เข้าร่วมการประชุม ซึ่งผลลัพธ์จะเป็นดังรูปที่ 5-14 และรูปที่ 5-15

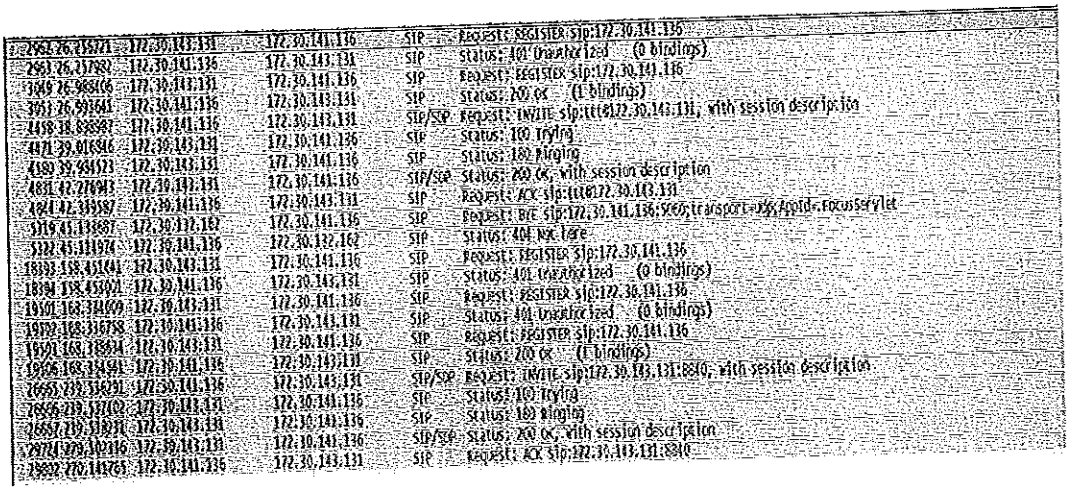


รูปที่ 5-14 หน้าแสดงส่วนการเข้าสู่ระบบของผู้ร่วมเข้าประชุม



รูปที่ 5-15 หน้าแสดงส่วนการเชิญเข้าสู่ระบบของผู้ร่วมเข้าประชุม

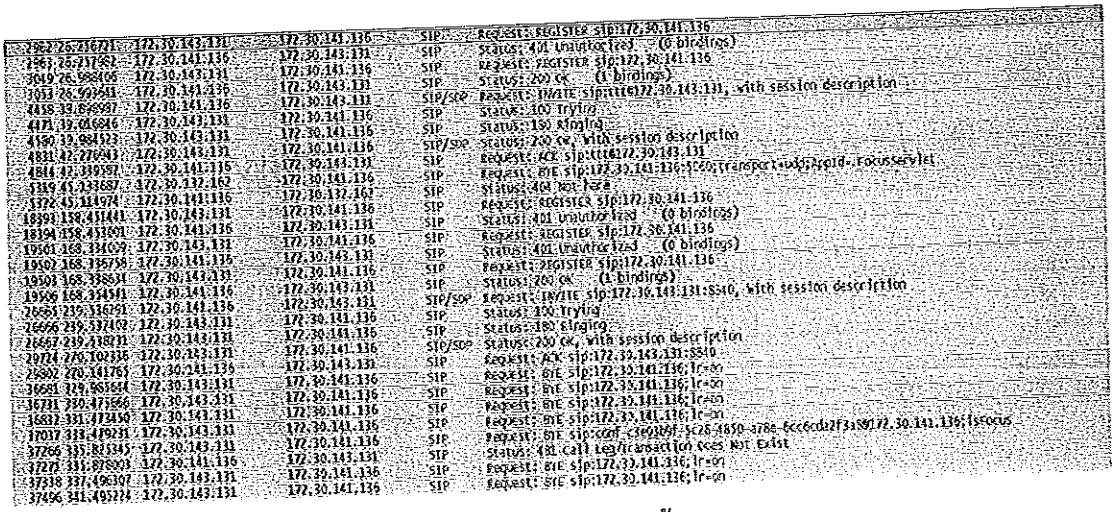
โดยลำดับสัญญาณที่เกิดขึ้นทั้งหมดจะแสดงอยู่ในรูปที่ 5-16 ซึ่งเป็นรูปแบบสัญญาณตามที่ออกแบบไว้ จึงพิสูจน์ได้ว่าระบบสามารถทำงานได้อย่างถูกต้องตามที่ออกแบบไว้ และสังเกตได้ว่าโปรแกรมฝั่งลูกข่ายสามารถเข้าร่วมการทำงานได้อย่างสมบูรณ์ถึงแม้จะไม่โปรแกรมชนิดเดียวกัน



รูปที่ 5-16 ลำดับสัญญาณในการเชิญผู้ร่วมการประชุมคนอื่นเข้าร่วมการประชุม

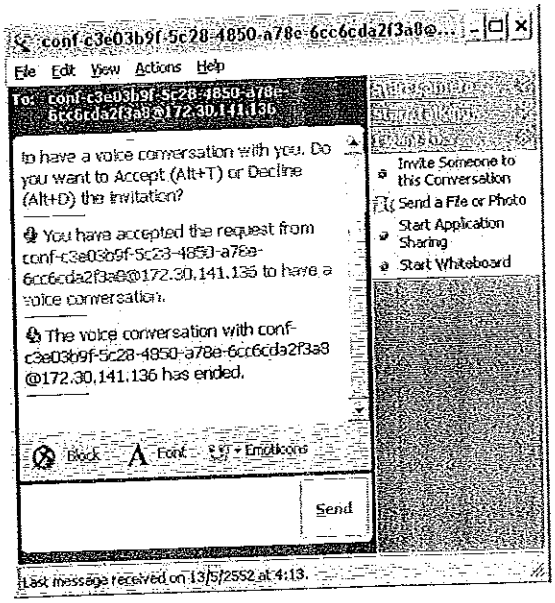
5.9.3 การสิ้นสุดการประชุม

เมื่อการประชุมสุดลงประธานการประชุมจะทำการออกจากการประชุมซึ่งมีผลทำให้ผู้ร่วมการประชุมคนอื่นของออกจากการประชุมเช่นกัน ซึ่งจากผลการทดสอบในรูปที่ 5-17 จะแสดงให้เห็นสัญญาณ SIP BYE ถูกส่งไปยังผู้เข้าร่วมการประชุมแต่ละคน ซึ่งเป็นการพิสูจน์ว่าระบบสามารถที่จะสิ้นสุดการประชุมได้อย่างสมบูรณ์



รูปที่ 5-17 ลำดับสัญญาณในการสิ้นสุดการประชุม

ในรูปที่ 5-18 จะแสดงหน้าจอโปรแกรมของผู้เข้าร่วมการประชุมเมื่อได้รับสัญญาณสิ้นสุดการประชุม



รูปที่ 5-18 หน้าจอผู้เข้าร่วมการประชุมเมื่อสิ้นสุดการประชุม

5.10 สรุป

ในบทนี้ได้นำเสนอการพัฒนาและการทดสอบการทำงานของโปรแกรมประยุกต์และส่วนประกอบต่าง ๆที่ได้พัฒนาขึ้นเพื่อใช้ในการสร้างการประชุมแบบสื่อประสม ซึ่งจากผลลัพธ์ได้พบว่าโปรแกรมประยุกต์ที่พัฒนาขึ้นมา ส่วนต่าง ๆในระบบสามารถทำงานเข้ากันได้อย่างถูกต้องสมบูรณ์ โดยพิสูจน์ได้จากรูปแบบสัญญาณได้ทำการบันทึกไว้ ซึ่งมีรูปแบบเป็นไปตามดังที่ออกแบบไว้ในบทที่ 4 นอกจากนี้ในการทดสอบการทำงานผู้เข้าร่วมการประชุมทุกทดนสามารถที่จะได้รับข้อมูลสื่อประสมอย่างครบถ้วน

จึงสรุปได้ว่าสถาปัตยกรรมการประชุมที่ได้ทำการออกแบบไว้ทั้งหมดสามารถที่จะนำมาใช้งานได้จริง แต่เพื่อประสิทธิภาพการทำงานที่ดีขึ้นจำเป็นที่จะต้องศึกษาส่วนของ Scalability ซึ่งจะทำให้ระบบสามารถรองรับการติดต่อได้มากยิ่งขึ้น ซึ่งข้อเสนอแนะต่าง ๆ จะขอกล่าวในบทต่อไป

บทที่ 6
สรุปผลการวิจัย และข้อเสนอแนะ

ในบทนี้จะทำการสรุปผลการทำวัยของวิทยานิพนธ์นี้ ทั้งหมดรวมไปถึงข้อเสนอแนะและแนวคิดต่าง ๆเพื่อให้ผู้ที่สนใจสามารถทำการพัฒนาต่อยอดความสามารถต่าง ๆ ของระบบการประชุมแบสื่อประสมโดยใช้โปรโตคอล SIP ให้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งมีหัวข้อต่าง ๆ ดังต่อไปนี้

6.1 สรุปผลการวิจัย

ในการทำวิจัยสำหรับวิทยานิพนธ์ ได้ทำการศึกษาค้นคว้าหาข้อมูลต่าง ๆ ในการออกแบบสถาปัตยกรรมการประชุมแบบสื่อประสม รวมไปถึงเทคนิคการพัฒนาโปรแกรมประยุกต์ต่าง ๆ เพื่อใช้ในการพัฒนาระบบการประชุมเพื่อให้ให้สามารถใช้งานได้ตามโครงสร้างที่ได้ทำการออกแบบไว้ โดยจากทดสอบให้ผู้ใช้ต่าง ๆทำการสร้างการประชุมและทำการเข้าร่วมการประชุมพร้อม ๆกัน ระบบสามารถจัดการผู้เข้าร่วมการประชุมต่าง ๆได้เป็นอย่างดี ซึ่งสามารถสรุปกระบวนการทำงานออกเป็นส่วนต่าง ๆ ดังนี้

6.1.1 การออกแบบสถาปัตยกรรมสำหรับการประชุมแบบสื่อประสม

ในส่วนของการออกแบบสถาปัตยกรรมการประชุมแบบสื่อประสม ได้ทำการศึกษาข้อมูลจากบทความที่เกี่ยวข้องรวมไปถึงข้อมูลจากกลุ่มวิจัยต่าง ๆ เพื่อให้ได้แนวคิดที่น่าสนใจและสามารถนำมาประยุกต์ใช้งานให้เกิดประสิทธิภาพมากที่สุด ซึ่งจากศึกษาค้นคว้าพบว่าโปรโตคอล SIP เป็นโปรโตคอลหนึ่งที่กำลังได้รับความนิยมเป็นอย่างมากในปัจจุบันในด้านการติดต่อสื่อสาน สังเกตได้จากมีการผลิตอุปกรณ์ต่าง ๆมากที่รองรับทำงานโปรโตคอล SIP จึงเลือกที่จะนำมาใช้ในส่วนสำหรับการสร้างการประชุมแต่ก็พบว่าโปรตคอล SIP อย่างเดียวไม่สามารถสร้างระบบการประชุมที่ยืดหยุ่นและรองรับความต้องการต่าง ๆผู้ใช้ได้อย่างสมบูรณ์ รวมไปถึงกลไลในการควบคุมการทำงานระหว่างทำการประชุม จึงได้ทำการศึกษาค้นคว้าหาเทคนิคที่สามารถนำมาประยุกต์ใช้กับระบบที่มีอยู่ ซึ่งมีเทคนิคที่น่าสนใจคือการเพิ่มองค์ประกอบเข้าไปเพื่อช่วยให้การทำงานของระบบมีฟังก์ชันการทำงานที่มากยิ่งขึ้น โดยได้ทำการเพิ่มองค์ประกอบใหม่สองส่วนคือ

- Conference Manager
- Conference Repository

ซึ่งในการเพิ่มองค์ประกอบสองส่วนนี้เข้าไปในระบบที่ออกแบบ ทำให้เกิดกระบวนติดต่อรูปแบบใหม่ต่าง ๆ เช่น การตั้งเวลาการประชุม การค้นหาการประชุม เป็นต้น โดยในการเพิ่มองค์ประกอบเหล่านี้เข้าไปในระบบผู้ติดต่อที่อยู่ในฝั่งของลูกข่ายไม่จำเป็นต้องเปลี่ยนรูปแบบการติดต่อใด ๆทั้งสิ้น เนื่องจากแนวคิดที่จะเพิ่มองค์ประกอบใหม่เข้าไปในระบบนั้นจะเน้นการทำงานที่ฝั่งของแม่ข่าย

นอกจากนี้ในการออกแบบยังได้เสนอรูปแบบและรายละเอียดของสัญญาณในการติดต่อทั้งในส่วนของผู้ติดต่อกับระบบการประชุมและการติดต่อภายในองค์ประกอบต่าง ๆที่ใช้ในระบบการประชุม โดยได้นำเสนอการใช้งานโปรโตคอล XCAP เพื่อนำมาใช้ในการกำหนดค่ากฎข้อบังคับต่าง ๆที่ใช้ในการประชุม

ดังนั้นจากข้อมูลทั้งหมดทำให้สามารถออกแบบสถาปัตยกรรมการประชุมแบบสื่อประสมที่มีความสามารถรองรับความต้องการของผู้ใช้ต่าง ๆได้เป็นอย่างดี นอกจากนี้ยังได้นำเสนอรูปแบบการติดต่อเพิ่มเติมจากการประชุมแบบทั่วไป ซึ่งทำให้ระบบมีความสามารถเหมาะกับการใช้งานที่หลากหลายได้

6.1.2 การพัฒนาโปรแกรมประยุกต์สำหรับใช้ในการควบคุมการประชุมแบบสื่อประสมและการทดสอบการทำงาน

ในส่วนของการพัฒนาโปรแกรมประยุกต์เพื่อนำมาใช้ในการทดสอบระบบการประชุมแบบสื่อประสมที่ได้ออกแบบไว้ ได้เลือกใช้เทคโนโลยีจาวามาใช้ในการพัฒนา ซึ่งเป็นเทคโนโลยีที่มีผู้ใช้งานอย่างแพร่หลาย และมีเครื่องต่าง ๆให้เลือกใช้มากมายในการพัฒนาซึ่งโปรแกรมประยุกต์ที่พัฒนาขึ้นมาจะทำงานอยู่ในส่วนของฝั่งแม่ข่าย โดยจะรอรับการติดต่อจะผู้ติดต่อที่ใช้ SIP User Agent ต่าง ๆ เพื่อทดสอบความสามารถของระบบที่สามารถรับการติดต่อได้ไม่ว่าผู้ใช้จะ SIP User Agent แบบใดก็ตาม พบว่าจากการทดสอบผู้ใช้สามารถที่จะสร้างห้องการประชุมจากระบบที่ทำการออกแบบและพัฒนาไว้ นอกจากนี้ผู้ใช้คนอื่น ๆสามารถที่จะเข้าร่วมการประชุมได้และสามารถจบการประชุมได้อย่างสมบูรณ์ ซึ่งเป็นการพิสูจน์ว่าสถาปัตยกรรมที่ออกแบบสามารถนำไปใช้งานได้จริง

6.2 ข้อเสนอแนะ

สำหรับข้อเสนอแนะในการนำงานวิจัยนี้ไปพัฒนาต่อ โดยมีข้อเสนอแนะที่น่าสนใจดังต่อไปนี้

6.2.1 รูปแบบข้อมูลสื่อประสม

เนื่องจากโปรแกรมประยุกต์ที่พัฒนาขึ้นระบบการประชุมจะรองรับข้อมูลสื่อประสมที่เป็นแบบเสียงเท่านั้น ยังไม่รองรับการใช้งานข้อมูลสื่อประสมในรูปแบบอื่นเช่น ภาพ เป็นต้น เพราะว่าติดข้อจำกัดในส่วนของ Mixer ที่จำเป็นต้องพัฒนาให้มีความสามารถในการรวมข้อมูลเหล่าเข้าด้วย ซึ่งยังขาดความรู้ในการพัฒนาตรงส่วนนี้ จึงอยากแนะนำให้ผู้ต้องการนำงานวิจัยนี้ไปพัฒนาต่อทำการศึกษาในประเด็นเหล่านี้ เพื่อที่จะพัฒนาระบบให้ดีต่อไป

6.2.2 โปรแกรมประยุกต์สำหรับลูกข่าย

สำหรับระบบที่พัฒนาขึ้น ถึงแม้ว่าจะสามารถใช้ SIP User Agent ใด ๆเข้ามาติดต่อก็ได้ แต่จะพบว่าหากต้องการใช้งานระบบอย่างสมบูรณ์จำเป็นที่ต้องพัฒนาโปรแกรมประยุกต์ในฝั่งลูกข่ายให้รองความสามารถต่าง ๆ ตามมาตรฐานที่ระบบได้กำหนดไว้ ดังนั้นจึงเป็นประเด็นหนึ่งที่น่าสนใจในการศึกษาและค้นคว้าต่อไป

เอกสารอ้างอิง

[1] V. Kumar, M. Korpi and S. Sengoodan, "IP Telephony with H.323: Architectures for Unified Networks and Integrated Services", USA: John Wiley & Sons, 2001.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, Internet Engineering Task Force, June 2002.

[3] K. Katrinis, G. Parissidis, and B. Plattner (Switzerland), "A Comparison of Framework for Multimedia Conferencing: SIP and H.323

[4] J. Glasmann,W. Kellerer, and H. Muller, "Service Architectures in H.323 and SIP: A Comparison, IEEE Communications Society Surveys and Tutorials,2003

[5] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications, " RFC 1889, January 1996.

[6] M. Handley, V. Jacobson, "SDP: Session Description Protocol",RFC 2327, April 1998.

[7] A. Johnston and SIPPING Working Group, "draft-ietf-sipping-service-examples-05", August 29, 2003.

[8] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, Internet Engineering Task Force, February 2006.

[9] H.Khartabil, P.Koskelainen,and A. Miemi, "The conference policy control protocol",IETF,Internet draft <draft-ietf-xcon-cpcp-01>,2004.

[10] Koskelainen, P., Schulzrinne, H. and X. Wu, "A sip-based conference control framework", Nossdav'2002 Miami Beach, May 2002.

[11] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman,Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer, Simple Object Access Protocol (SOAP) 1.1, W3C Note, May 2000.

[12] Y.-H. Cho, M.-S. Jeong, J.-W. Nah, W.-H. Lee, and J.-T. Park," Policy-Based Distributed Management Architecture for Large-Scale Enterprise Conferencing Service Using SIP", IEEE Journal, October 2005.

[13] Novo," Policy A Common Conference Information Data Model for Centralized Conferencing", Internet draft <draft-ietf-xcon-common-data-model-02>,2006.

[14] J. Rosenberg, The Extensible Markup Language (XML) Configuration Access Protocol (XCAP), IETF Internet Draft, June 2003.

[15] K. Katrinis, G. Parissidis and B.Plattner, "A Comparison of Frameworks for Multimedia Conferencing: SIP and H.323", IASTED Internet Multimedia Systems and Applications (IMSA 2004), Kauai, Hawaii, USA, August 16-18, 2004.

[16] J. Rosenberg, "A Session Initiation Protocol (SIP) Event Package for Conference State", Internet draft <draft-ietf-sipping-conference-package-12>,July 2005.

[17] P. Koskelainen, Requirements for Conference Policy Control Protocol, IETF Internet Draft, June 2003.

[18] J. Rosenberg, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, Internet Engineering Task Force, August 2006.

[19] K. Zeilenga, "Lightweight Directory Access Protocol (LDAP):Technical Specification Road Map", Internet Engineering Task Force, RFC 4510, June 2006.

[20] A. Johnston, O. Levin, " Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", Internet Engineering Task Force,RFC 4579, August 2006.

[21] Even, Ismail, "Conferencing Scenarios", RFC 4597, Internet Engineering Task Force, July 2006.

[22] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, Internet Engineering Task Force, April 2003.

[23] J. Van Dyke, E. Burger, "Media Server Control Markup Language (MSCML) and Protocol", Internet Engineering Task Force, RFC 4722, November 2006.

[24] OpenSIPS, Available from: http://www.opensips.org/ , Last Accessed: April 10, 2009.

[25] SER, Available from: http://www.iptel.org/ser , Last Accessed: April 10, 2009.

[26] J2EE, Available from: http://en.wikipedia.org/wiki/J2EE , Last Accessed: April 10, 2009.

[27] WeSIP, Available from: http://www.wesip.com/mediawiki/index.php/Main_Page , Last Accessed: April 10, 2009.

[28] Jain-SIP, Available from: https://jain-sip.dev.java.net/ , Last Accessed: April 10, 2009.

[29] SEMS, Available from: http://www.iptel.org/sems/ , Last Accessed: April 10, 2009.WebDev, Available from: http://en.wikipedia.org/wiki/WebDev , Last Accessed: March 5, 2009.

ประวัติผู้เขียน

ชื่อ สกุล นายณัฐดนัย วิทยาศิริกุล

รหัสประจำตัวนักศึกษา 4812111

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2548

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนศิษย์ก้นกุฏิ คณะวิศวกรรมศาสตร์

การตีพิมพ์เผยแพร่ผลงาน

N. Wittayasirikul, P. Boonlert, S. Kamolphiwong, S. Sae-Wong, T. Kamolphiwong, "Enhancements of IP Conference Scenarios", Proceedings of the International Conference on Mobile Technology Applications and Systems, , September 2007.