

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง วิทยาเขตระยอง

ระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ต

INTERNET RADIO BROADCAST MANAGEMENT SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาดตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERNET RADIO BROADCAST MANAGEMENT SYSTEM



A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ต	
ชื่อนักศึกษา	นางสาวชลลดา เรืองศิลป์ประเสริฐ	รหัสประจำตัว 43010087
	นางสาวชุติมน ประวีณวรกุล	รหัสประจำตัว 43010107
	นางสาวหทัยชนก นวลประเสริฐ	รหัสประจำตัว 43010511
อาจารย์ที่ปรึกษา	อาจารย์ภูษงค์ หงษ์สุวรรณ	
ระดับการศึกษา	ปริญญาตรีวิศวกรรมศาสตรบัณฑิต	
	สาขาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2546	

วิทยานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

(อาจารย์ภูษงค์ หงษ์สุวรรณ)
 อาจารย์ผู้ควบคุมวิทยานิพนธ์

หัวข้อปริญญานิพนธ์	ระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ต	
ชื่อนักศึกษา	นางสาวชลลดา เรืองศิลป์ประเสริฐ	รหัสประจำตัว 43010087
	นางสาวชุติมน ประวีณวรกุล	รหัสประจำตัว 43010107
	นางสาวหทัยชนก นวลประเสริฐ	รหัสประจำตัว 43010511
อาจารย์ที่ปรึกษา	อาจารย์ภูษงค์ หงษ์สุวรรณ	
ระดับการศึกษา	ปริญญาตรีวิศวกรรมศาสตรบัณฑิต	
	สาขาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2546	



ปริญญานิพนธ์นี้ กล่าวถึงกรนำเทคโนโลยีอินเทอร์เน็ตมาประยุกต์ใช้ร่วมกับการสื่อสารแบบวิทยุกระจายเสียง มีจุดประสงค์หลักเพื่อสร้างแอปพลิเคชันสำหรับอำนวยความสะดวกให้แก่ผู้ดำเนินรายการ ในระบบวิทยุกระจายเสียงผ่านอินเทอร์เน็ต โดยการทำงานของแอปพลิเคชันจะแบ่งออกเป็นสองส่วนหลักคือ ส่วนฮาร์ดแวร์ จะทำหน้าที่รับการเชื่อมต่อจากไคลเอนท์ และทำการส่งข้อมูลเสียงไปยังไคลเอนต์ทุกเครื่องที่มีการเชื่อมต่อมายังเซิร์ฟเวอร์อยู่ในขณะนั้น โดยส่งข้อมูลเป็นแบบเรียลไทม์เพื่อให้มีรูปแบบการสื่อสารเป็นแบบเดียวกับระบบวิทยุกระจายเสียงแบบธรรมดา กล่าวคือ เซิร์ฟเวอร์จะส่งข้อมูลเสียงไปอย่างต่อเนื่องในขณะที่ไคลเอนท์จะทำการเล่นเสียงไปพร้อม ๆ กับการรับข้อมูลเสียงจากเซิร์ฟเวอร์ ไม่ต้องดาวน์โหลดข้อมูลเสียงมาทั้งหมดมาเก็บไว้ก่อนจึงทำการเล่นเสียง และส่วนที่สองคือ โปรแกรมจัดการการกระจายเสียง มีหน้าที่จัดการกับข้อมูลเสียงที่จะออกอากาศ และควบคุมการออกอากาศ โดยผู้ดำเนินรายการสามารถจัดการและควบคุมผ่านทางส่วนติดต่อกับผู้ใช้ซึ่งออกแบบมาเพื่อจุดประสงค์หลักคือ ให้ผู้ดำเนินรายการสามารถใช้งานได้ง่าย ประหยัดเวลา และช่วยลดการระระหว่างดำเนินรายการให้ได้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Internet Radio Broadcast Management System

Student Miss Chonlada Ruangsinsprasert ID 43010087
Miss Chutimon Praweenvorakul ID 43010107
Miss Hathaichanok Nuanprasert ID 43010511

Advisor Mr.Puchong Hongsuwan

Graduate Level Bachelor Degree of Information Engineering

Department Information Engineering

Academic Year 2003

ABSTRACT

This project describes applying the Internet technology to the radio broadcast communication system for reaching the very helpful application for the administrator of the internet radio broadcast communication system. The application works together two parts , the first one is the Audio Server which is always waits for the connection requests from the clients and then accept them before continuing broadcast the audio stream to all clients those are being connected to it at the time. As in the real time systems the server continuously broadcasts audio stream while the clients are also continuously reading and playing back the audio without downloading the whole audio before playing back so this behavior makes this system is the same type of communication as the ordinary radio broadcast system.

The second part is the Program Manager, the module for management and controlling the audio broadcasting, works through the easy using user interfaces which is designed for helping the administrator to save much more time and be easy to control the audio broadcasting when on air.

สารบัญ

เรื่อง	หน้า
สารบัญ	จ
สารบัญรูปภาพ	ฉ
สารบัญตาราง	ช
บทที่ 1 บทนำ	1
1.1 แนวคิดและที่มาของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการดำเนินโครงการ	3
1.5 ผลที่คาดว่าจะได้รับ	3
1.6 อุปกรณ์ที่ต้องใช้	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 การเข้ารหัสสัญญาณเสียง	4
2.1.1 ไซโคอะคูสติกโมเดล (Psychoacoustics Model)	5
2.1.2 การวิเคราะห์เชิงความถี่ (Spectral Analysis)	7
2.1.3 การลดสัญญาณรบกวน (Noise Allocations)	8
2.1.4 เทคนิคอื่น ๆ ในคารบีบอัด	8
2.2 การเข้ารหัสสัญญาณเสียงแบบเอ็มพีทีรี (MPEG-1 Audio Layer 3)	9
2.2.1 ลักษณะของสัญญาณเสียงแบบเอ็มพีทีรี	9
2.2.2 วิธีการเข้ารหัสสัญญาณเสียงแบบเอ็มพีทีรี	10
2.3 โพรโตคอลที่ใช้ในการส่งข้อมูลแบบเรียลไทม์	11
2.3.1 ลักษณะของการสื่อสารแบบเรียลไทม์	12
2.3.2 โพรโตคอลอาร์ทีพี (RTP Protocol)	15
2.3.3 โพรโตคอลอาร์ทีซีพี (RTCP Protocol)	17
2.3.4 โพรโตคอลอาร์ทีเอสพี (RTSP Protocol)	19
2.4 จาวามีเดียเฟรมเวิร์ก (Java Media Framework, JMF)	20
2.4.1 การจับข้อมูลมัลติมีเดีย	21
2.4.2 คาต้าซอร์ส	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง

หน้า

2.4.3	เพลย์เซอ์	25
2.4.4	โปรเซสเซอร์ (Processor)	27
2.4.5	เมนเจอร์ (Managers)	31
2.5	เครือข่ายมัลติคาสต์ (Multicast Network)	32
2.5.1	เอ็ม โบน (MBone หรือ Multicast Backbone)	34
2.5.2	กลุ่มของมัลติคาสต์ (Multicast Groups)	35
2.5.3	มัลติคาสต์เราท์ติ้งอัลกอริทึม (Multicast Routing Algorithms)	37
2.5.3.1	ฟลัดดิ้ง (Flooding)	37
2.5.3.2	สเปนนิงทรี (Spanning Trees)	37
2.5.3.3	อาร์พีบี (Reverse Path Broadcasting (RPB))	38
2.5.3.4	ทีอาร์พีบี (Truncated Reverse Path Broadcasting (TRPB))	39
2.5.3.5	อาร์พีเอ็ม (Reverse Path Multicasting (RPM))	40
2.5.3.6	เอสที (Steiner Tree (ST))	41
2.5.3.7	ซีบีที (Core-Based Tree (CBT))	41
2.5.4	มัลติคาสต์เราท์ติ้งโปรโตคอล (Multicast Routing Protocol)	42
2.5.4.1	ดีวีเอ็มอาร์พี (Distance Vector Multicast Routing Protocol (DVMRP))	42
2.5.4.2	เอ็ม โอเอสทีเอฟ (Multicast Extensions to OSPF (MOSPF))	43
2.6	การเขียนโปรแกรมจาวาด้วยเน็ตฟเมรอด	45
2.6.1	การเขียนโค้ดภาษาจาวา	47
2.6.2	การคอมไพล์โค้ดจาวา	49
2.6.3	การสร้างไฟล์เซคเตอร์	49
2.6.4	การสร้างเน็ตฟเมรอด	51
2.6.5	การสร้างแชร์ไลบรารี	52
2.6.6	การรันโปรแกรม	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 3 การออกแบบ	54
3.1 ข้อมูลเบื้องต้น	54
3.1.1 ฟังก์ชันสำหรับผู้ดูแลระบบ (Administrator)	54
3.1.2 ฟังก์ชันสำหรับผู้ดำเนินรายการ (DJ)	54
3.2 การออกแบบระบบ	55
3.2.1 การออกแบบสถาปัตยกรรมของระบบ (System Architecture Design)	55
3.2.2 โดเมนโมเดล (Domain Model)	56
3.2.3 ยูสเคสไดอะแกรม (Use Case Diagram)	58
3.2.4 แอกทิวิตีไดอะแกรม (Activity Diagram)	59
3.2.5 การออกแบบส่วนติดต่อผู้ใช้ (User Interface)	60
3.2.5.1 หน้าต่างหลัก	60
3.2.5.2 หน้าต่างสำหรับสร้างผังรายการล่วงหน้า	63
3.2.5.3 หน้าต่างสำหรับแก้ไขผังรายการล่วงหน้า	65
บทที่ 4 ผลการทดลอง	68
4.1 ทดลองเรียกแอปพลิเคชัน	68
4.2 ทดลองสร้างผังรายการปัจจุบันด้วยโหมคการทำงานแบบควบคุมด้วยมือ (Manual Manager)	69
4.3 ทดลองเล่นมีเดีย	72
4.4 ทดลองหยุดเล่นมีเดียชั่วคราว	74
4.5 ทดลองเล่นมีเดียถัดไปในผังรายการปัจจุบัน	76
4.6 ทดลองเล่นมีเดียก่อนหน้าในผังรายการปัจจุบัน	78
4.7 ทดลองหยุดเล่นมีเดีย	80
4.8 ทดลองลบมีเดียที่เลือกไว้ในผังรายการปัจจุบัน	82
4.9 ทดลองล้างผังรายการปัจจุบัน	84
4.10 ทดลองสร้างผังรายการล่วงหน้า	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.11 ทดลองเพิ่มเสียงเพลงหรือเสียงโฆษณาลงในผังรายการลွ่งหน้า (Auto Playlist)	88
4.12 ทดลองแทรกช่วงหยุดเล่นมีเดียชั่วคราวในผังรายการลွ่งหน้า	90
4.13 ทดลองลบมีเดียออกจากผังรายการ	93
4.14 ทดลองการตั้งค่าวันที่ และเวลาของผังรายการลွ่งหน้า	95
4.15 ทดลองบันทึกการสร้างผังรายการลွ่งหน้า	97
4.16 ทดลองแก้ไขผังรายการลွ่งหน้า	101
4.17 ทดลองการสร้างผังรายการลွ่งหน้าที่สร้างไว้	106
4.18 ทดลองลบผังรายการลွ่งหน้า	109
4.19 ทดลองสลับการทำงานเป็น โหมดอัตโนมัติ	111
4.20 ทดลองส่งข้อมูลเสียงไปยัง ไอศเอนท์	114
4.21 ทดลองปรับระดับเสียงของระบบ	119
4.22 ทดลองปรับระดับเสียงไมโครโฟน	120
บทที่ 5 สรุปผลการทดลอง	123
5.1 สรุปผลการทดลอง	123
5.1.1 ลักษณะของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ต โดยรวม	123
5.1.2 สรุปผลการทดลองแต่ละส่วน	124
5.2 ปัญหาที่เกิดขึ้นระหว่างการทดลอง	124
5.2.1 ปัญหาที่เกิดขึ้นระหว่างขั้นตอนการสร้างระบบ	124
5.2.2 ปัญหาที่เกิดขึ้นระหว่างขั้นตอนการทดลอง	126
5.2.3 ปัญหาอื่น ๆ	126
5.3 แนวทางการพัฒนาระบบต่อ	127
5.3.1 การพัฒนาส่วนจัดการการกระจายเสียง	127
5.3.2 การพัฒนา โมดูลทาง ไอศเอนท์เพื่อแก้ไขปัญหาลิขสิทธิ์เอ็มพีทีรี เอน โค้ดเคอร์	127

บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
๗

สารบัญรูปภาพ

เรื่อง	หน้า	
รูปที่ 2.1	การเข้ารหัสสัญญาณเสียงของแอน โคลด์เคอร์แบบทั่วไป	4
รูปที่ 2.2	เชรส โยลค์ของการ ได้ยินมีจนถึงที่ความถี่ 20 กิโลเฮิร์ตซ์	5
รูปที่ 2.3	การบดบังทางความถี่	6
รูปที่ 2.4	การบดบังแบบชั่วคราว	6
รูปที่ 2.5	บล็อกไดอะแกรมของแอน โคลด์เคอร์ของเอ็มพีที	10
รูปที่ 2.6	ความสัมพันธ์เชิงเวลาของการสื่อสารแบบเรียลไทม์	12
รูปที่ 2.7	ปัญหาจิดเคอร์	13
รูปที่ 2.8	การแก้ปัญหาคิดเคอร์ โดยใช้โทมัสแลมปี	13
รูปที่ 2.9	รูปแบบส่วนหัวของแพ็คเกจอาร์ทีพี	15
รูปที่ 2.10	รูปแบบส่วนหัวของแพ็คเกจอาร์ทีซีพี	18
รูปที่ 2.11	สถานะของเซิร์ฟเวอร์	20
รูปที่ 2.12	เปรียบเทียบการทำงานของกลองวีดี โอกับเจเอ็มเอฟ	21
รูปที่ 2.13	ค่าค่า โมเดลของเจเอ็มเอฟ	22
รูปที่ 2.14	ฟอร์แมตของบีบีโอในเจเอ็มเอฟ	24
รูปที่ 2.15	แบบจำลองของเพลย์เบอรัในเจเอ็มเอฟ	25
รูปที่ 2.16	คลาสไดอะแกรมของเพลย์เบอรัในเจเอ็มเอฟ	25
รูปที่ 2.17	สถานะของเพลย์เบอรั	26
รูปที่ 2.18	แบบจำลองของโปรเซสเซอร์ในเจเอ็มเอฟ	27
รูปที่ 2.19	คลาสไดอะแกรมของโปรเซสเซอร์ในเจเอ็มเอฟ	28
รูปที่ 2.20	ภาพรวมของสถานะต่าง ๆ ของโปรเซสเซอร์	28
รูปที่ 2.21	สถานะของโปรเซสเซอร์ในเจเอ็มเอฟ	30
รูปที่ 2.22	การส่งข้อมูลแบบยูนิคาสต์	33
รูปที่ 2.23	การส่งข้อมูลแบบมัลติคาสต์	33
รูปที่ 2.24	ตัวอย่างการส่งข้อมูลในแบบต่าง ๆ	34
รูปที่ 2.25	การทำอู โมงส์ผ่านเครือข่ายที่ไม่สนับสนุนการทำมัลติคาสต์	35
รูปที่ 2.26	รูปแบบของไอพีแอดเดรสคลาสสิค	36
รูปที่ 2.27	สเปนนิงทรี	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เรื่อง	หน้า	
รูปที่ 2.28	อาร์พีบีทีรี	39
รูปที่ 2.29	อาร์พีเอ็มทีรี	40
รูปที่ 2.30	เอสทีทีรี	41
รูปที่ 2.31	ซีบีที	42
รูปที่ 2.32	แสดงขั้นตอนการสร้างโปรแกรม Hello World	46
รูปที่ 2.33	แสดงให้เห็นถึงรูปแบบชื่อของเนตเวิร์กที่สร้างขึ้นที่ถูกสร้างจากเนตเวิร์กเฟรมเวิร์ค	50
รูปที่ 3.1	สถาปัตยกรรมของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ต	55
รูปที่ 3.2	โดเมน โมเดลของระบบจัดการการกระจายเสียงผ่านอินเทอร์เน็ต	57
รูปที่ 3.3	ยูสเคส ไดอะแกรมสำหรับการจับเสียงจากไมโครโฟน	58
รูปที่ 3.4	ยูสเคส ไดอะแกรมสำหรับการเข้ารหัสสัญญาณเสียงให้อยู่ในรูปแบบแพ็คเกจอาร์ทีพี	58
รูปที่ 3.5	ยูสเคส ไดอะแกรมสำหรับการเลือกสัญญาณเสียงสำหรับส่งมีเดียสตรีม	58
รูปที่ 3.6	ยูสเคส ไดอะแกรมสำหรับการส่งสตรีมเสียงไปยังไคลเอนท์	59
รูปที่ 3.7	แอคทีวิตี ไดอะแกรมสำหรับการจับ เสียงจากไมโครโฟน	59
รูปที่ 3.8	แอคทีวิตี ไดอะแกรมสำหรับการเข้ารหัสสัญญาณเสียงให้อยู่ในรูปแบบแพ็คเกจอาร์ทีพี	59
รูปที่ 3.9	แอคทีวิตี ไดอะแกรมสำหรับการเลือกสัญญาณเสียงสำหรับส่งมีเดียสตรีม	60
รูปที่ 3.10	แอคทีวิตี ไดอะแกรมสำหรับการการส่งสตรีมเสียง ไปยังไคลเอนท์	60
รูปที่ 3.11	หน้าต่างหลักสำหรับการจัดการการกระจายเสียงผ่านอินเทอร์เน็ต	61
รูปที่ 3.12	หน้าต่างสำหรับสร้างผังรายการล่วงหน้า	64
รูปที่ 3.13	หน้าต่างสำหรับแก้ไขผังรายการล่วงหน้า	67
รูปที่ 4.1	หน้าต่างหลัก	68
รูปที่ 4.2	วิธีทดลองเลือกแหล่งข้อมูลเสียง โดยใช้ตัวควบคุมด้วยมือ	70
รูปที่ 4.3	หน้าต่างเลือกแหล่งข้อมูลเสียง	71
รูปที่ 4.4	หน้าต่างหลักเมื่อทำการเลือกแหล่งข้อมูลเสียงแล้ว	72
รูปที่ 4.5	หน้าต่างหลักก่อนการเล่นเสียงเพลง หรือเสียงโฆษณา	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เรื่อง	หน้า	
รูปที่ 4.6	หน้าต่างหลักเมื่อทำการเล่นเสียงเพลง หรือเสียงโฆษณา	74
รูปที่ 4.7	หน้าต่างหลักก่อนการหยุดเล่นเสียงเพลง หรือเสียงโฆษณาชั่วคราว	75
รูปที่ 4.8	หน้าต่างหลักเมื่อทำการหยุดเล่นเสียงเพลง หรือเสียงโฆษณาชั่วคราว	76
รูปที่ 4.9	หน้าต่างหลักก่อนเริ่มเล่นมีเดียถัดไปในผังรายการปัจจุบัน	77
รูปที่ 4.10	หน้าต่างหลักเมื่อเริ่มเล่นมีเดียถัดไป	78
รูปที่ 4.11	หน้าต่างหลักก่อนเริ่มเล่นมีเดียก่อนหน้าในผังรายการปัจจุบัน	79
รูปที่ 4.12	หน้าต่างหลักเมื่อเริ่มเล่นมีเดียก่อนหน้าในผังรายการปัจจุบัน	80
รูปที่ 4.13	หน้าต่างหลักก่อนการหยุดเล่นมีเดีย	81
รูปที่ 4.14	หน้าต่างหลักเมื่อทำการหยุดเล่นมีเดีย	82
รูปที่ 4.15	หน้าต่างหลักก่อนสลับมีเดียที่เลือกไว้ในผังรายการปัจจุบัน	83
รูปที่ 4.16	หน้าต่างหลักเมื่อทำการสลับมีเดียที่เลือกไว้ออกจากผังรายการปัจจุบัน	84
รูปที่ 4.17	หน้าต่างหลักก่อนทำการล้างผังรายการปัจจุบัน	85
รูปที่ 4.18	หน้าต่างหลักเมื่อทำการล้างผังรายการปัจจุบัน	86
รูปที่ 4.19	หน้าต่างหลักก่อนสร้างผังรายการล่วงหน้า	87
รูปที่ 4.20	หน้าต่างสำหรับสร้างผังรายการล่วงหน้า	88
รูปที่ 4.21	วิธีเลือกแหล่งเสียงสำหรับสร้างผังรายการล่วงหน้า	89
รูปที่ 4.22	หน้าต่างสำหรับสร้างผังรายการล่วงหน้าหลังจากทำการเลือกมีเดียที่ต้องการ	90
รูปที่ 4.23	วิธีแทรกช่วงเวลาว่างในผังรายการล่วงหน้า	91
รูปที่ 4.24	ไดอะล็อกซ็อบ (Input Dialog Box) สำหรับรับค่าช่วงเวลาว่างหน่วยเป็นวินาที	92
รูปที่ 4.25	ทดลองป้อนค่าช่วงเวลาว่างในไดอะล็อกซ็อบ	92
รูปที่ 4.26	หน้าต่างควบคุมแบบอัตโนมัติ เมื่อทำการแทรกการหยุดเล่นมีเดียชั่วคราว	93
รูปที่ 4.27	วิธีลบมีเดียออกจากผังรายการล่วงหน้า	94
รูปที่ 4.28	หน้าต่างสำหรับสร้างผังรายการล่วงหน้าหลังจากทำการลบมีเดียออกจากผังรายการ	95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญรูปภาพ (ต่อ)

เรื่อง		หน้า
รูปที่ 4.29	วิธีการตั้งค่าวันที่และเวลาเริ่มต้นของผังรายการล่วงหน้า	96
รูปที่ 4.30	หน้าต่างสำหรับสร้างผังรายการล่วงหน้าเมื่อทำการตั้งค่าวันที่และเวลาสำหรับผังรายการล่วงหน้า	97
รูปที่ 4.31	วิธีการบันทึกการสร้างผังรายการล่วงหน้า	98
รูปที่ 4.32	ไดอะล็อกบ็อกซ์เตือนกรณีไม่มีรายการเสียงในผังรายการที่สร้าง	99
รูปที่ 4.33	ไฟล์แสดงรายละเอียดของผังรายการล่วงหน้า	99
รูปที่ 4.34	ข้อมูลของผังรายการล่วงหน้าในหน้าต่างหลักเพิ่มขึ้นหลังจากทำการบันทึกผังรายการ	100
รูปที่ 4.35	ไดอะล็อกบ็อกซ์เตือนการตั้งค่าวันที่และเวลาผิดพลาด	101
รูปที่ 4.36	วิธีการแก้ไขผังรายการล่วงหน้าที่สร้างไว้แล้ว	102
รูปที่ 4.37	หน้าต่างแก้ไขผังรายการล่วงหน้า	103
รูปที่ 4.38	ทดลองแก้ไขผังรายการล่วงหน้า	104
รูปที่ 4.39	ไฟล์นามสกุลเอ็นเอเอ็มหลังการแก้ไขผังรายการล่วงหน้า	105
รูปที่ 4.40	หน้าต่างหลักเมื่อทำการแก้ไขผังรายการล่วงหน้า	106
รูปที่ 4.41	หน้าต่างสำหรับสร้างผังรายการล่วงหน้าก่อนทำการล้างผังรายการที่สร้างไว้	107
รูปที่ 4.42	ไดอะล็อกบ็อกซ์ยืนยันการเคลียร์รายละเอียดผังรายการล่วงหน้า	108
รูปที่ 4.43	หน้าต่างสำหรับสร้างผังรายการล่วงหน้าเมื่อทำการล้างผังรายการที่สร้างไว้	109
รูปที่ 4.44	หน้าต่างหลัก ก่อนการลบผังรายการล่วงหน้า	110
รูปที่ 4.45	หน้าต่างหลักเมื่อทำการลบผังรายการล่วงหน้า	111
รูปที่ 4.46	หน้าต่างหลัก ก่อนการสลับเป็น โหมดอัตโนมัติ	112
รูปที่ 4.47	หน้าต่างหลักหลังจากสลับการทำงานเป็น โหมดอัตโนมัติ	113
รูปที่ 4.48	หน้าต่างหลักเมื่อทำงานใน โหมดอัตโนมัติตามเวลาเริ่มต้นของผังรายการ	114
รูปที่ 4.49	วิธีทดลองส่งข้อมูลเสียงไปยังไคลเอนท์	115
รูปที่ 4.50	วิธีการเชื่อมต่อของไคลเอนท์โปรแกรมมายังระบบ	116

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เรื่อง		หน้า
รูปที่ 4.51	หน้าต่างหลักเมื่อทำการส่งข้อมูลเสียงไปยัง โคลนเอนท์	117
รูปที่ 4.52 ก.	โปรแกรมโคลนเอนท์เมื่อเริ่มทำการเชื่อมต่อไปยังเซิร์ฟเวอร์	118
รูปที่ 4.52 ข.	โปรแกรมโคลนเอนท์เมื่อเริ่มการรับข้อมูลเสียง	118
รูปที่ 4.53	วิธีปรับเสียงระดับเสียงของระบบ	119
รูปที่ 4.54	โปรแกรมโคลนเอนท์เมื่อทำการปรับระดับเสียงของระบบ	120
รูปที่ 4.55	วิธีปรับเสียงไมโครโฟน	121
รูปที่ 4.56	โปรแกรมโคลนเอนท์เมื่อทำการปรับระดับเสียงไมโครโฟน	122



สารบัญตาราง

เรื่อง		หน้า
ตารางที่ 2.1	ตารางแสดงคุณสมบัติของมาตรฐานเอ็มเปกอดีโอแต่ละเลขเออร์	10
ตารางที่ 2.2	ชนิดของข้อมูลในแพคเกจอาร์ทีพี	16
ตารางที่ 2.3	ชนิดของแพคเกจอาร์ทีซีพี	18
ตารางที่ 2.4	เมธอดของโปรโตคอลอาร์ทีเอสพี	20
ตารางที่ 2.5	ค่าต่าง ๆ ที่ใช้ในการควบคุมขอบเขตที่ทีแอล	43
ตารางที่ 4.1	สถานะเริ่มต้นของแอปพลิเคชัน	69



บทที่ 1

บทนำ

1.1 แนวคิดและที่มาของปัญหา

จากการศึกษาปัญหาของระบบวิทยุกระจายเสียงผ่านอินเทอร์เน็ตที่มีอยู่ในปัจจุบันพบว่า ผู้ดำเนินรายการต้องรับภาระในการดำเนินรายการวิทยุและต้องคอยควบคุมการออกอากาศเอง ทำให้เสียเวลาและอาจเกิดความผิดพลาดระหว่างการควบคุมได้ง่าย จึงได้เกิดแนวคิดสร้างระบบจัดการการกระจายเสียงผ่านอินเทอร์เน็ต โดยโครงงานนี้มีจุดประสงค์หลักเพื่อสร้างแอปพลิเคชันสำหรับอำนวยความสะดวกให้แก่ผู้ดำเนินรายการในระบบวิทยุกระจายเสียงผ่านอินเทอร์เน็ต ซึ่งระบบนี้ประกอบด้วยสองส่วนหลัก ๆ คือ ส่วนบรอดคาสต์เซิร์ฟเวอร์ (Audio Broadcast Server) ทำหน้าที่กระจายข้อมูลเสียงแบบมัลติคาสต์ไปยังไคลเอนต์ และส่วนที่สองนั้นคือ โปรแกรมสำหรับจัดการการกระจายเสียง (Program Management) มีหน้าที่จัดการกับข้อมูลเสียงที่จะออกอากาศ และควบคุมการออกอากาศ โดยผู้ดำเนินรายการสามารถจัดการการกระจายเสียงผ่านทางส่วนติดต่อกับผู้ใช้ (Graphic User -Interface) โดยเลือกการควบคุมการทำงานของระบบได้สองแบบคือ แบบควบคุมด้วยมือ (Manual Mode) และแบบควบคุมอัตโนมัติ (Automatically Playing Mode) โดยการพัฒนาเลือกใช้ภาษาจาวา เนื่องจากตัวภาษามีความง่าย, ยืดหยุ่น และคงทน นอกจากนั้นยังทำให้แอปพลิเคชันสามารถทำงานได้กับทุกแพลตฟอร์ม

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการทำงานของระบบวิทยุกระจายเสียงผ่านอินเทอร์เน็ต
2. เพื่อพัฒนาแอปพลิเคชันสำหรับจัดการการกระจายเสียงผ่านอินเทอร์เน็ตโดยใช้ภาษาจาวาบนแพลตฟอร์มวินโดวส์ (Window Platform)

1.3 ขอบเขตของโครงงาน

1.3.1. พัฒนาโปรแกรมส่วนจัดการการส่งข้อมูล

โครงงานนี้มีเป้าหมายในการพัฒนาโปรแกรมส่วนจัดการการส่งข้อมูล หรือโปรแกรมจัดการการกระจายเสียง (Program Management Module) ให้มีความสามารถสนับสนุนการจัดการการกระจายเสียงได้สองแบบ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.1.1. แบบควบคุมด้วยมือ (Manual Mode)

โดยจะพัฒนาให้มีคุณสมบัติดังนี้

1. สามารถเลือกไฟล์เสียงเพลงหรือเสียงโฆษณาสำหรับรอกการออกอากาศได้
2. สามารถคำนวณเวลาสำหรับแต่ละไฟล์เสียงที่เลือกในข้อหนึ่งได้
3. สามารถเลือกลบแหล่งเสียงที่เลือกไว้ในข้อหนึ่งได้
4. สามารถควบคุมการเล่นและการหยุดเพลงได้
5. เมื่อเพลงจบลง สามารถเล่นเพลงต่อไปได้ทันที
6. สามารถควบคุมระดับความดังของเสียงจากไฟล์เสียงและเสียงจากไมโครโฟนได้
7. สามารถควบคุมการผสมเสียงระหว่างเสียงจากไฟล์เสียงกับเสียงจากไมโครโฟนได้สามารถส่งข้อมูลเสียงไปยังบรรดาคาสท์เซิร์ฟเวอร์ได้

1.3.1.2. แบบควบคุมอัตโนมัติ (Automatically Playing Mode)

โดยจะพัฒนาให้มีคุณสมบัติดังนี้

1. สามารถแสดงรายการของไฟล์เสียงแบ่งตามแขนแนลดังนี้ แขนแนลเสียงเพลง และแขนแนลเสียงโฆษณา
2. สามารถสร้างผังรายการล่วงหน้า โดยการเลือกแหล่งเสียงจากรายการข้างต้นได้
3. สามารถใส่ช่วงว่าง (Pause Time) ลงในผังรายการได้โดยกำหนดช่วงเวลาในหน่วยวินาที
4. สามารถแสดงเวลาที่ใช้ในแต่ละรายการได้
5. สามารถคำนวณเวลาทั้งหมดและเวลาสิ้นสุดรายการได้โดยกำหนดเวลาเริ่มรายการให้
6. สามารถสร้างรายการได้มากกว่าหนึ่งรายการ
7. สามารถกำหนดเวลาในการนำรายการขึ้นมาใช้งานได้
8. สามารถส่งข้อมูลเสียงไปยังบรรดาคาสท์เซิร์ฟเวอร์ตามที่ได้กำหนดไว้ในผังรายการได้อย่างถูกต้อง
9. ผังรายการที่สร้างไว้ สามารถนำมาแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.2. พัฒนาโปรแกรม broadcast สำหรับเซิร์ฟเวอร์

ให้มีคุณสมบัติดังนี้

1. สามารถเข้ารหัสข้อมูลเสียงให้สามารถส่งผ่าน โปรโตคอลในเน็ตเวิร์กแบบเรียลไทม์ได้
2. สามารถส่งสตรีมเสียงแบบมัลติคาสท์ไปยังมัลติคาสท์เราเตอร์ได้
3. สามารถให้บริการไคลเอนท์หลายๆ การเชื่อมต่อได้ในเวลาเดียวกัน

1.4 ขั้นตอนการดำเนินโครงการ

1. ศึกษาการทำงานของระบบวิทยุกระจายเสียงผ่านระบบอินเทอร์เน็ต
2. ศึกษาหลักการของจาวามีเดียเฟรมเวิร์ก (Java Media Framework)
3. ศึกษาการเขียนโปรแกรมเชิงวัตถุโดยใช้ภาษาจาวา
4. รวบรวมความต้องการของผู้ใช้
5. ออกแบบระบบ
6. ลงมือพัฒนาระบบแบบแยกส่วน
7. ทดสอบการใช้งานแบบแยกส่วน
8. รวบรวมส่วนต่างๆ เข้าด้วยกันและแก้ไขข้อผิดพลาด
9. คิดตั้งแอปพลิเคชันและทดสอบการใช้งาน
10. บันทึกลงและสรุปผลการทดลอง
11. จัดทำปริญญานิพนธ์

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถพัฒนาระบบจัดการการกระจายเสียงผ่านทางอินเทอร์เน็ต
2. สามารถพัฒนาแอปพลิเคชันบนระบบปฏิบัติการวินโดวส์ได้
3. สามารถพัฒนาแอปพลิเคชันด้วยภาษาจาวาได้

1.6 อุปกรณ์ที่ต้องใช้

1. คอมพิวเตอร์ที่สามารถรองรับระบบมัลติมีเดียและเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต จำนวน 2 เครื่อง
2. ซอฟต์แวร์ j2sdk 1.4.2
3. ซอฟต์แวร์ JMF 2.1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

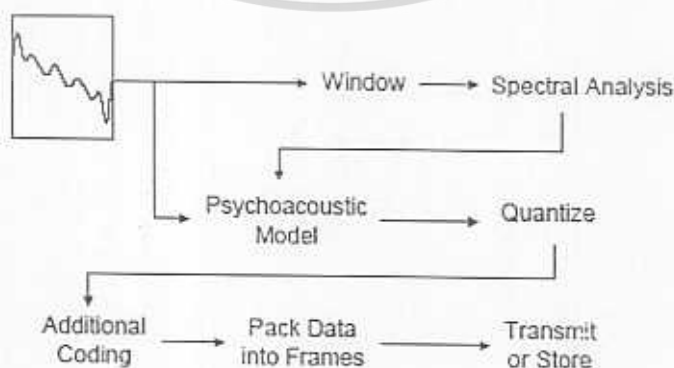
บทที่ 2

ทฤษฎีและหลักการ

2.1 การเข้ารหัสสัญญาณเสียง

เทคนิคในการประมวลผลสัญญาณ (Signal Processing Techniques) เป็นหนึ่งในเทคโนโลยีขั้นก้าวหน้าของเสียงดิจิทัล ตัวอย่างสำคัญที่มีการนำเทคนิคในการประมวลผลสัญญาณมาประยุกต์ใช้ ได้แก่ แอปพลิเคชันสำหรับบรอดคาสต์ผ่านอินเทอร์เน็ต ถ้าหากต้องการบรอดคาสต์ข้อมูลเสียงที่มีคุณภาพสูงผ่านระบบอินเทอร์เน็ต โดยข้อมูลเสียงถูกเข้ารหัสแบบพีซีเอ็ม (Pulse Code Modulation (PCM)) 16 บิต ที่ความถี่ 44.1 กิโลเฮิร์ตซ์ แอปพลิเคชันต้องใช้ช่องสัญญาณ (Channel) ที่มีแบนด์วิธ (Bandwidth) ไม่น้อยกว่า 1.4 เมกะบิตต่อวินาที ซึ่งปัจจุบันหากเชื่อมต่ออินเทอร์เน็ตด้วยโมเด็ม จะมีแบนด์วิธ 56 กิโลบิตต่อวินาทีเท่านั้น เทคนิคการโพรเซสสัญญาณจึงถูกนำมาใช้เพื่อลดขนาดของสัญญาณลง ให้สามารถส่งผ่านช่องสัญญาณที่มีแบนด์วิธน้อยลงได้ เทคนิคนี้เรียกว่าการบีบอัด (Compression) หรือการเข้ารหัส (Encoding) สัญญาณเสียง การบีบอัดสัญญาณเสียงดิจิทัลคือการตัดข้อมูลส่วนที่ซ้ำซ้อนหรือส่วนที่ไม่ต้องการออกไป

การเข้ารหัสกระทำโดยเอนโคเดอร์ (Encoder) ซึ่งมักจะเป็น โปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ที่มีวงจรเสียงติดตั้งอยู่ รูปที่ 2.1 แสดงถึงวิธีการเข้ารหัสของเอนโคเดอร์ทั่วไป ซึ่งจะรับข้อมูลอินพุตเป็นบล็อกของสัญญาณเสียงที่ถูกแซมเปิลลิง (Sampled Audio Signal) บล็อกหนึ่งจะมีค่าแซมเปิลลิง (Samples) ตั้งแต่ 500-1500 ค่าต่อหนึ่งช่องสัญญาณ แล้วแต่เอนโคเดอร์แต่ละตัว ตัวอย่างเช่น เอนโคเดอร์ของเอ็มพีทีรี (MPEG-1 audio layer-III) มี 576 ค่าแซมเปิลลิงต่อหนึ่งช่องสัญญาณต่อหนึ่งบล็อก ส่วนเอาต์พุตคือผลที่ได้จากการบีบอัดของข้อมูลแต่ละบล็อก (เรียกเป็น "เฟรม") ซึ่งสามารถส่งไปหรือเก็บไว้ทำการถอดรหัสในภายหลัง



รูปที่ 2.1 การเข้ารหัสสัญญาณเสียงของเอนโคเดอร์แบบทั่วไป

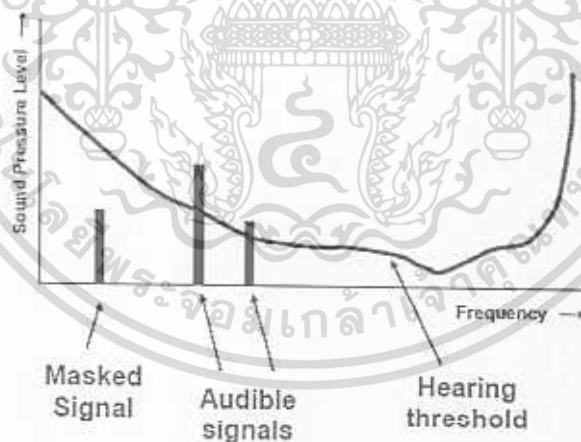
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 ไชโคอคูสติก โมเดล (Psychoacoustics Model)

แนวคิดพื้นฐานในการลดขนาดข้อมูลที่รับเข้ามาคือ ทำการตัดข้อมูลเสียงที่มนุษย์ไม่ได้ยินทิ้งไป การบีบอัดข้อมูลชนิดนี้ มักถูกเรียกว่า การเข้ารหัสแบบเพอร์เซปชวล (Perceptual Encoding) ซึ่งต้องศึกษาการได้ยินเสียงของมนุษย์ วิธีการในการบีบอัดเสียงยังนำหลักการที่เสียงมีลักษณะบดบังกันเอง (Masking) มาใช้ด้วย ลักษณะที่ว่ามีเกิดขึ้นได้สามแบบดังนี้ คือ การถูกตัดด้วยธรศโฮลด์ (Threshold Cut-Off), การบดบังทางความถี่ (Frequency Masking) และการบดบังชั่วคราว (Temporal Masking)

2.1.1.1 การถูกตัดด้วยธรศโฮลด์

มนุษย์สามารถได้ยินเสียงหลายระดับ วัดเป็นระดับความเข้มเสียง (Sound Pressure Level (SPL)) ถ้าระดับระดับของเสียงต่ำกว่าธรศโฮลด์การได้ยิน (Hearing Threshold) มนุษย์จะไม่ได้ยินเสียงนั้น รูปที่ 2.2 แสดงธรศโฮลด์ในฟังก์ชันของความถี่ของเสียง จากรูปสังเกตได้ว่า ส่วนประกอบของเสียงที่อยู่ใต้ธรศโฮลด์จะไม่ได้ยิน หรือเป็นสัญญาณถูกมาสก์ไว้ (Masked Signal) ขณะที่ส่วนประกอบของเสียงที่อยู่เหนือธรศโฮลด์จะเป็นเสียงที่มนุษย์ได้ยิน (Audible Signals)

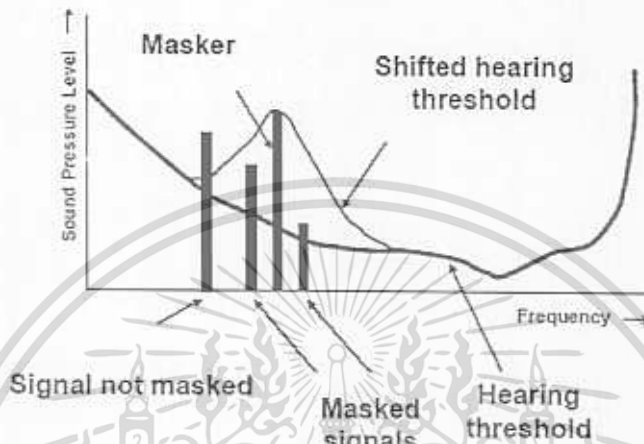


รูปที่ 2.2 ธรศโฮลด์ของการได้ยินมีจนถึงที่ความถี่ 20 กิโลเฮิรต์ซ์

2.1.1.2 การบดบังทางความถี่

แม้ว่าส่วนประกอบของเสียงบางอันจะอยู่เหนือธรศโฮลด์ แต่มันก็อาจถูกบดบังโดยเสียงที่ดังกว่าและมีความถี่ใกล้เคียงกันได้ ลักษณะที่เกิดขึ้นแบบนี้เรียกว่า การบดบังทางความถี่ (Frequency Masking) หรือการบดบังของเสียงที่เกิดขึ้นพร้อมกัน (Simultaneous Masking) ปรากฏการณ์นี้เกิดขึ้นเนื่องจากแต่ละส่วนประกอบของเสียงจะคล้ายมีเงาแผ่ไปยังความถี่ใกล้เคียง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

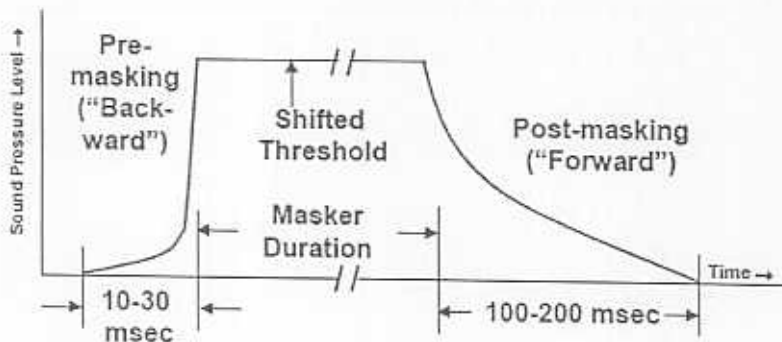
จากรูปที่ 2.3 สัญญาณที่ไปบดบังสัญญาณอื่น (Masker) จะมีแอมพลิจูดเกิดขึ้น เนื่องจากแอมพลิจูด (Amplitude) ของตัวมันมีความแรงแต่ไปยังความถี่ใกล้เคียง ทำให้ข้อมูลเสียงจริงที่ความถี่ใกล้เคียงที่อาจมีแอมพลิจูดน้อยกว่าซีฟท์เฮียร์ริงแอมพลิจูดถูกบดบัง เนื่องจากที่ความถี่หนึ่ง ๆ มนุษย์จะได้ยินเฉพาะเสียงที่ดังที่สุดเท่านั้น



รูปที่ 2.3 การบดบังทางความถี่

2.1.1.3 การบดบังแบบชั่วคราว

ลักษณะแบบนี้ เหมือนกับกรณีที่เสียงคล้ายมีเงาไปพาดทับเสียงข้างเคียง ในแกนความถี่ ในกรณีนี้จะเป็นการที่เสียงเต็มความดังอย่างรวดเร็วจนสามารถบดบังเสียงที่เงียบกว่าซึ่งฟังจะหยุดไปได้ เหตุการณ์ที่เกิดขึ้นนี้เรียกว่า การบดบังชั่วคราว (Temporal Masking) สิ่งที่น่าสนใจอีกอย่างหนึ่งคือ ทั้งก่อนและหลังจากที่เสียงดังขึ้นอย่างรวดเร็ว นั้น สัญญาณข้างเคียงเสียงนั้นอาจถูกบดบังได้ รูปที่ 2.4 จะแสดงถึงสถานการณ์ที่ถูกบดบังชั่วคราว สถานการณ์นี้เสียงที่อยู่ใต้แอมพลิจูดนี้จะไม่ได้ยิน สังเกตได้ว่าช่วงก่อนที่จะถูกบดบัง (Pre-Masking) นั้นสั้นกว่าช่วงหลังที่ถูกบดบัง (Post-Masking)



รูปที่ 2.4 การบดบังแบบชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 การวิเคราะห์เชิงความถี่ (Spectral Analysis)

สเปกตรัม (Spectrum) หรือสัญญาณในโดเมนความถี่ มีประโยชน์ในการวิเคราะห์ลักษณะทางความถี่และใช้ในการหาแอมพลิจูด เทคนิคในการแปลงสัญญาณจากโดเมนเวลาไปเป็นโดเมนความถี่ มีอยู่หลายเทคนิคด้วยกัน แบ่งได้เป็นสองประเภทคือ การแปลง (Transform) และการใช้ฟิลเตอร์ (Filter banks) โดยที่การแปลงจะแตกสเปกตรัมของสัญญาณอินพุตแต่ละตัวออกเป็นอนุกรมย่อย ตัวอย่างเช่น การแปลงฟูเรียร์ (Fourier Transform) จะแตกสเปกตรัมออกเป็นอนุกรมของเอ็กโปเนนเชียลที่ซับซ้อน ส่วนการใช้ฟิลเตอร์นั้น จะเป็นการป้อนสัญญาณอินพุตผ่านเข้าแบนด์พาสฟิลเตอร์หลาย ๆ ตัวที่มีช่วงความถี่ต่าง ๆ กัน ซึ่งจะได้ผลเป็นอนุกรมในแกนเวลาหลายอนุกรม ตามแต่ละความถี่

ก่อนจะกล่าวถึงโครงสร้างต้องไม่ลืมว่าการสร้างสเปกตรัมทำเพื่อจุดมุ่งหมาย 2 ประการคือ

- เพื่อหาแอมพลิจูดสำหรับอ้างอิงให้สัญญาณที่ต่ำกว่าแอมพลิจูดถูกตัดทิ้งไปได้
- เพื่อแปลงรูปแบบสัญญาณให้เป็นรูปแบบที่นำมาอ้างอิงกับแอมพลิจูดได้

การแปลงที่นิยมกันมากที่สุดคือ เอ็ฟเอ็ฟที (Fast Fourier Transform, FFT) เป็นการป้อนสัญญาณในโดเมนเวลา เอ็ฟเอ็ฟทีจะให้ผลในโดเมนความถี่ที่มีค่าเป็นจำนวนเชิงซ้อน เอนโค้ดเดอร์มักใช้เอ็ฟเอ็ฟทีเพื่อหาแอมพลิจูด การแปลงอีกแบบหนึ่งที่นิยมได้แก่ ดีซีที (Discrete Cosine Transform, DCT) ซึ่งให้ผลในโดเมนความถี่ที่มีค่าเป็นจำนวนจริง ทั้งเอ็ฟเอ็ฟทีและดีซีทีต่างก็ประสบปัญหาสัญญาณผิดเพี้ยนที่ขอบของสัญญาณในแต่ละบล็อก อันเนื่องมาจากกระบวนการแปลงข้อมูลเพื่อแก้ปัญหาสัญญาณอินพุตแต่ละบล็อกจึงต้องมีการซ้อนทับกัน (Overlapping and Windowing) ด้วยเหตุนี้เองที่เอนโค้ดเดอร์ส่วนใหญ่ที่ใช้เทคนิคการแปลง มักใช้ดีซีทีที่กำหนดให้สัญญาณอินพุตแต่ละบล็อกมีการซ้อนทับกันได้ ซึ่งเรียกว่าเอ็มดีซีที (Modified Discrete Cosine Transform, MDCT) เช่น การเข้ารหัสเอ็มพีทีที, เอ็มเปกทู เอ็ชซีซี (MPEG-2 AAC) และดอลบี เอ็ชทีที (Dolby AC-3)

การใช้ฟิลเตอร์จะป้อนบล็อกข้อมูลที่แซมปลิงแล้ว เข้าแบนด์พาสฟิลเตอร์ที่ย่านความถี่ต่างกัน สัญญาณอินพุตจึงถูกแยกเป็นสัญญาณที่ความถี่ย่อย ๆ หลังจากถูกกรองโดยฟิลเตอร์แล้วจะนำไปอ้างอิงกับแอมพลิจูดเพื่อตัดส่วนที่ถูกบดบังโดยแอมพลิจูดออก โครงสร้างที่นิยมของการใช้ฟิลเตอร์มีสองแบบคือ ฟิลเตอร์แบบโพลีเฟส (Poly-Phase Filter Bank) และฟิลเตอร์แบบเวฟเลต (Wavelet Filter Bank) แบบโพลีเฟสจะใช้แบนด์พาสฟิลเตอร์หลายตัวที่มีแบนด์วิธเท่ากันมาขนานกัน ได้ผลเป็นค่าแซมปลิงที่น้อยลงเนื่องจากถูกแบ่งที่หลายความถี่ต่างกัน

เทคนิคการบีบอัดแบบมีการสูญเสีย (Lossy Compression Techniques) ดีโค้ดเดอร์ (Decoder) สามารถนำสัญญาณกลับมาได้โดยใช้ฟิลเตอร์แบบย้อนกลับและรวมสัญญาณจากทุกแบนด์ย่อยเข้าด้วยกัน โครงสร้างดังกล่าวจะใช้ในเอนโค้ดเดอร์เอ็มเปกวันออดีโอ (MPEG-1 audio)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การลดสัญญาณรบกวน (Noise Allocations)

กลับไปทีบัสล็อกโคอะแกรม จากการทำวิเคราะห์สเปกตรัม (Spectrum Analysis) สัญญาณในแกนเวลาจะถูกแสดงในแกนความถี่ เป้าหมายต่อไปคือ การบีบอัดสัญญาณอินพุตโดยการตัดข้อมูลที่ไม่จำเป็นออกไป ซึ่งทำได้โดยใส่เรสโวลต์ลงในสัญญาณอินพุตนั้น เรสโวลต์การได้ยินจะเป็นตัวบ่งชี้ว่าเสียงที่ความถี่ไหนต้องแรงเท่าไรถึงจะได้ยิน อธิบายได้สองแง่ อย่างแรกคือเสียงที่อยู่ใต้เรสโวลต์ไม่จำเป็นต้องถูกส่งไป และอย่างที่สองสัญญาณรบกวน (Noise) ที่อยู่ใต้เรสโวลต์เป็นสิ่งที่เราไม่พึงปรารถนา

ในการเข้ารหัสสัญญาณแอนะล็อกเคอร์ต้องทำการควอนไทซ์ (Quantizing) คือ การแปลงสัญญาณเสียงที่แต่ละความถี่ให้เป็นเลขฐานสอง บิตแอนะล็อกเคอร์ใช้จำนวนบิตในการควอนไทซ์มาก ความผิดพลาดก็จะยิ่งน้อย แต่ก็ใช้พื้นที่ในการเก็บข้อมูลมากขึ้นด้วย ในทางกลับกันถ้าใช้บิตน้อย ความผิดพลาดมาก แต่ก็ใช้พื้นที่น้อย อย่างไรก็ตามหากสัญญาณรบกวนอยู่ใต้เรสโวลต์ เสียงนั้นก็จะไม่ได้ยิน เพื่อที่จะลดขนาดพื้นที่สำหรับเก็บข้อมูล (หรือลดแบนด์วิธที่ใช้ในการส่งข้อมูล) แอนะล็อกเคอร์จะใช้จำนวนบิตเท่าที่เพียงพอต่อการเข้ารหัสแต่ละความถี่ โดยไม่นำสัญญาณรบกวนที่มนุษย์ได้ยินมาทำการควอนไทซ์ เทคนิคนี้เรียกว่า เทคนิคการลดสัญญาณรบกวน

2.1.4 เทคนิคอื่นๆ ในการบีบอัด

แม้ว่าการบีบอัดมักจะทำโดยใช้เทคนิคการลดสัญญาณรบกวน อย่างที่กล่าวมาข้างต้น แต่ยังมีเทคนิคอื่นอีกหลายแบบที่ใช้ได้ ประกอบด้วย การทำนาย (Predicting) การคับปลิง (Cupling) และการเข้ารหัสโดยอิงทฤษฎีสารสนเทศ (Information-theory-based coding)

การทำนายเป็นเทคนิคหนึ่งที่มีประสิทธิภาพสูง ใช้เวลาเข้ารหัสเสียงพูดและภาพ แนวคิดพื้นฐานคือ เราสามารถประมาณค่าของค่าแซมปลิงถัดไปจากค่าแซมปลิงที่ไปแล้วได้ ดังนั้นแทนที่จะส่งค่าแซมปลิงถัดมา แอนะล็อกเคอร์จึงส่งเพียงแค่ส่วนต่างของค่าที่ประมาณกับค่าจริง ซึ่งจะใช้บิตจำนวนน้อยลง เมื่อสีโ้คเคอร์ทำการถอดรหัส จะสร้างค่าประมาณขึ้นก่อนด้วยวิธีการเดียวกันกับแอนะล็อกเคอร์แล้วทำการปรับ โดยใช้ส่วนต่างที่ได้รับจากแอนะล็อกเคอร์เอิมเพลกทูเอซีซี ใช้วิธีนี้เพื่อเพิ่มประสิทธิภาพในการบีบอัด

การคับปลิงจะถูกนำมาใช้ เมื่อสัญญาณเสียงจำนวนหลายสัญญาณต้องถูกเข้ารหัสและถอดรหัสแบบขนาน เช่นเสียงสเตอริโอ (Stereo) หรือเสียงรอบทิศทาง (Surround) ซึ่งประกอบด้วยหลายช่องสัญญาณและความต่างระหว่างช่องสัญญาณมักจะน้อย เวลาทำการรวมทุกช่องสัญญาณเข้าด้วยกันเพื่อทำการส่ง อาจใช้ความเหมือนกันนี้ให้เป็นประโยชน์ เช่น แอนะล็อกเคอร์อาจส่งผลรวมและผลต่างระหว่างสองช่องสัญญาณ ในกรณีที่สัญญาณเหมือนกันมาก ๆ ผลต่างจะมีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกือบเป็นศูนย์ อัลกอริทึมในการบีบอัดหนึ่งที่จะยกตัวอย่างต่อไปนี้ เป็นเทคนิคที่มีพื้นฐานการทฤษฎีการเข้ารหัสสารสนเทศชื่อ "ฮัฟแมน (Huffman)" ซึ่งเข้ารหัสโดยใช้จำนวนบิตน้อยที่สุดเป็นค่าสำหรับเข้ารหัสสัญญาณที่เกิดบ่อยครั้งที่สุด อย่างเช่น ถ้าเอาท์พุทของเอนโค้ดเดอร์ตัวหนึ่งมีค่า ๆ หนึ่งที่เกิดประมาณ 25 เปอร์เซ็นต์จากค่าที่เกิดขึ้นทั้งหมด เอนโค้ดเดอร์จะเข้ารหัสค่านี้นี้ด้วยโค้ดเวิร์ด (Codeword) ที่สั้นกว่าโค้ดเวิร์ดอื่น

2.2 การเข้ารหัสสัญญาณเสียงแบบเอ็มพีทีรี (MPEG-1 Audio Layer 3)

เอ็มเปก (Motion Picture Expert Group, MPEG) หรือกลุ่มผู้เชี่ยวชาญด้านภาพเคลื่อนไหว ก่อตั้งเมื่อปีค.ศ.1988 มีจุดประสงค์เพื่อสร้างมาตรฐานเปิดสำหรับการพัฒนาการเข้ารหัสสำหรับภาพเคลื่อนไหวและเสียง มาตรฐานการเข้ารหัสสัญญาณเสียงที่ถูกพัฒนาโดยกลุ่มเอ็มเปก ถูกใช้ในแอปพลิเคชันจำนวนมากประกอบด้วย การbroadcastเสียงดิจิทัล เสียงในระบบอินเทอร์เน็ต เสียงสำหรับเครื่องเล่นแบบพกพา ซีดีวีดี และเสียงที่เก็บในแหล่งเก็บเอ็มเปกรุ่นที่ได้รับความนิยมคือเอ็มพีทีรี ซึ่งถูกพัฒนาโดยห้องทดลองวิจัยเสียงของเยอรมันจากสถาบันฟรอนโฮเฟอร์ (Fraunhofer Institute)

เอ็มพีทีรีเป็นชื่อของเอ็มเปกวันออร์ทุ ออดิโอเลเยอร์ที่รี (MPEG-1/2, audio layer 3) ซึ่งเป็นชั้นเซตของการบีบอัดข้อมูลแบบเอ็มเปก มาตรฐานแบบเอ็มเปกวันออร์ทุ ออดิโอเลเยอร์ที่รีถูกพัฒนาขึ้นเพื่อเป็นมาตรฐานการลดขนาดข้อมูลในกรณีที่มีข้อมูลเป็นเสียงอย่างเดียว ไม่มีภาพ สามารถใช้ได้กับสัญญาณทั้งแบบ โมโนและสเตอริโอ และมีเทคนิคที่เรียกว่า จอยท์สเตอริโอ (Joint Stereo) สามารถเข้ารหัสแยกช่องสัญญาณซ้ายและขวาได้ ประกอบด้วยสามโหมดการทำงาน เรียกเป็นสามเลเยอร์ ซึ่งจะเพิ่มความซับซ้อนและความสามารถขึ้นเรื่อย ๆ ตั้งแต่เลเยอร์หนึ่งถึงเลเยอร์สาม

2.2.1 ลักษณะของสัญญาณเสียงแบบเอ็มพีทีรี

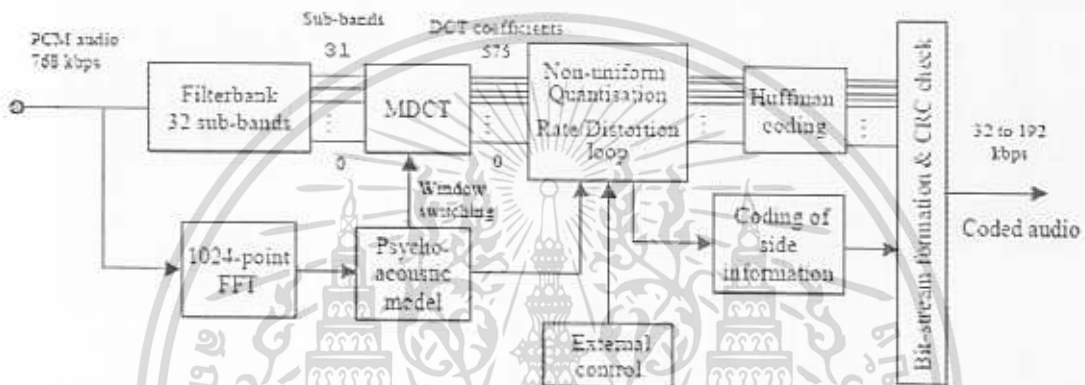
เอ็มพีทีรีได้รับความนิยมเป็นอย่างมาก เนื่องจากมันสามารถลดขนาดข้อมูลเสียงได้มากโดยที่ยังคงคุณภาพเสียงได้ไม่ต่างจากแผ่นคอมแพคดิสก์ จากตารางที่ 2.1 มาตรฐานเอ็มพีทีรีมีอัตราการลดขนาดข้อมูลมากที่สุดจากทั้งหมดสามเลเยอร์ คือ 1:10 ถึง 1:12 ทำให้ลดขนาดข้อมูลเสียงได้มากที่สุด ตัวอย่างเช่น เพลงจากซีดีความยาวสามนาทีจะมีขนาดประมาณ 20 เมกกะไบต์ มันสามารถลดขนาดลงเหลือประมาณ 2 เมกกะไบต์ และสามารถส่งข้อมูลได้ที่ระดับ 128 กิโลบิตต่อวินาทีสำหรับสัญญาณแบบสเตอริโอคือ ส่งได้ช่องสัญญาณละประมาณ 64 กิโลบิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน	อัตราการผลิตขนาดข้อมูล	จำนวนบิตต่อวินาทีของข้อมูล
MPEG Audio Layer 1 (MP1)	1:4	384 Kbps/Stereo
MPEG Audio Layer 2 (MP2)	1:6 ถึง 1:8	256 Kbps ถึง 192 Kbps/Stereo
MPEG Audio Layer 3 (MP3)	1:10 ถึง 1:12	128 Kbps ถึง 112 Kbps/Stereo

ตารางที่ 2.1 ตารางแสดงคุณสมบัติของมาตรฐานเอ็มเปกออกดีโอแต่ละเลเยอร์

2.2.2 วิธีการเข้ารหัสสัญญาณเสียงแบบเอ็มพีทีรี



รูปที่ 2.5 บล็อกไดอะแกรมของเอนโค้ดเดอร์ของเอ็มพีทีรี

จากรูปที่ 2.5 คือ บล็อกไดอะแกรมของเอนโค้ดเดอร์ของเอ็มพีทีรี ชั้นแรกผ่านสัญญาณเข้าฟิลเตอร์แบงก์แบบโพลีเฟส ซึ่งประกอบด้วยฟิลเตอร์จำนวน 32 ตัวต่อกันแบบขนานเพื่อแบ่งสัญญาณออกเป็นย่านความถี่ย่อยทั้งหมด 32 ความถี่ จากนั้นผ่านสัญญาณที่ออกจากฟิลเตอร์แต่ละตัวเข้าเอ็มดีซีที เพื่อทำการแปลงสัญญาณในแกนเวลาให้เป็นสัญญาณในแกนความถี่ที่แต่ละแบงก์จะได้เอาต์พุต 18 ช่องสัญญาณ จากทั้ง 32 แบงก์จะให้เอาต์พุตทั้งหมด 576 สัญญาณแซมปลิง การรวมเอาฟิลเตอร์แบงก์กับเอ็มดีซีทีเข้าด้วยกันนี้ เพื่อทำการกรองและแปลงสัญญาณ เรียกว่า “ไฮบริดฟิลเตอร์แบงก์” (Hybrid Filter Bank) หากมีการรบกวนสัญญาณ (Attack) เกิดขึ้น เอนโค้ดเดอร์จะทำการลดบิตของสัญญาณอินพุตลงจาก 576 เหลือ 192 แซมปลิง ซึ่งวิธีนี้จะช่วยลดการเกิดเอคโค่ (echo) ได้ จากนั้นเอาต์พุตที่ออกจากเอ็มดีซีที จะถูกใส่เฮรอสโกลด์ในแกนความถี่เพื่อมาส์กสัญญาณที่ไม่ได้ยินและสัญญาณรบกวนออกไป ทำการควอนไทซ์ เข้ารหัสสัฟแมน จากนั้นใส่ข้อมูลควบคุม (Side Information) สร้างเป็นแพ็คเกจที่มีรูปแบบพร้อมสำหรับขั้นตอนการส่ง ส่วนควบคุมจะทำหน้าที่ควบคุมการเข้ารหัสให้เหมาะสมกับสภาวะภายนอก เช่น เมื่อข้อมูลมีจำนวนมาก แบนด์วิธไม่เพียงพอ ส่วนควบคุมจะสั่งลดระดับการควอนไทซ์ให้หยาบขึ้น ทำให้จำนวนบิตข้อมูลน้อยลง เรียกว่าการควอนไทซ์แบบไม่ตายตัว (Non-Uniform Quantization)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากดีโคเดอร์ไม่มีการคำนวณค่าเรสโลต์ จึงมีความซับซ้อนน้อยกว่าเอนโคเดอร์ ซึ่งดีโคเดอร์ทุกตัวต้องทำคือ แกะแพ็คเกจ นำส่วนข้อมูลมาถอดรหัสฮัฟแมน ถอดสเกลแกน ความถี่ แล้วนำสัญญาณไปเข้าฟิลเตอร์แบงก์ย้อนกลับ (Inverse Filter Bank) ได้กลับเป็นสัญญาณเริ่มต้นอีกครั้ง

2.3 โพรโทคอลที่ใช้ในการส่งข้อมูลแบบเรียลไทม์

การส่งข้อมูลแบบเรียลไทม์ (Real Time) มีความแตกต่างจากการส่งข้อมูลทั่ว ๆ ไป ดังนั้นจึงต้องมีโปรโตคอลพิเศษที่สามารถรองรับการสื่อสารชนิดนี้ได้ ชั้นของโปรโตคอลที่มีความสำคัญอย่างมากก็คือ โปรโตคอลชั้นนำส่งข้อมูล (Transport Layer Protocol) โดยปกติแล้วโปรโตคอลชั้นนำส่งข้อมูลที่นิยมใช้ในระบบอินเทอร์เน็ตคือ โปรโตคอลทีซีพี (TCP) หรือยูดีพี (UDP) แต่ทีซีพีนั้นไม่เหมาะกับการสื่อสารแบบเรียลไทม์ด้วยเหตุผลหลาย ๆ ประการดังนี้

1. ไม่รองรับรูปแบบที่ใช้ไทม์สแตมป์ (Timestamp) และเพลย์แบ็ก (Playback)
2. ไม่ยอมให้เกิดการสูญหายของข้อมูลขึ้น คือ เมื่อข้อมูลบางส่วนหายไป จะสั่งให้ทำการส่งใหม่ทันที แต่ในการสื่อสารแบบเรียลไทม์ เราไม่สามารถสั่งให้ทำการส่งข้อมูลใหม่ได้ เพราะจำเป็นต้องใช้ความต่อเนื่องของข้อมูล
3. ไม่รองรับรูปแบบการสื่อสารแบบมัลติคาสต์ (Multicast)
4. ใช้เวลานานในการส่งข้อมูลเนื่องจากต้องมีการทำทรีเวย์แฮนด์เชก (Three Way Handshake) ทุกครั้งที่ส่งข้อมูลแต่ละแพ็คเกจ ในขณะที่การสื่อสารแบบเรียลไทม์ต้องการความเร็วในการรับส่งข้อมูล

จากการศึกษาโปรโตคอลยูดีพีทำให้ทราบว่า โปรโตคอลยูดีพีเหมาะสมสำหรับการสื่อสารแบบเรียลไทม์มากกว่า เนื่องจากยูดีพีรองรับการสื่อสารแบบมัลติคาสต์ และยอมให้ข้อมูลบางส่วนสูญหายได้ โดยไม่สั่งให้ทำการส่งข้อมูลนั้น ๆ ใหม่ อย่างไรก็ตาม ยูดีพียังไม่มีรองรับรูปแบบที่ใช้การไทม์สแตมป์, การเรียงลำดับข้อมูล (Sequencing) และการรวมสัญญาณจากหลาย ๆ แหล่งกำเนิด (Mixing)

ด้วยสาเหตุที่ยูดีพีไม่สามารถรองรับคุณสมบัติบางประการของการสื่อสารแบบเรียลไทม์ได้ทั้งหมด จึงต้องมีโปรโตคอลตัวอื่นที่สามารถรองรับสิ่งดังกล่าวขึ้น เพื่อใช้ร่วมกับโปรโตคอลยูดีพี นั่นก็คือ โปรโตคอลอาร์ทีที (Real-Time Transport Protocol)

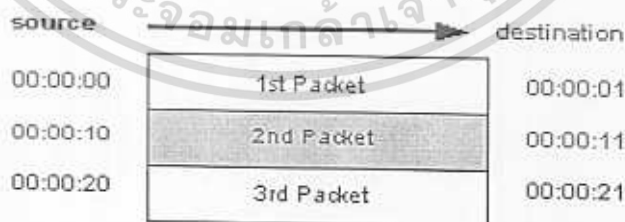
2.3.1 ลักษณะของการสื่อสารแบบเรียลไทม์

การสื่อสารแบบเรียลไทม์ มักจะเป็นการสื่อสารในรูปแบบของข้อมูลมัลติมีเดีย ซึ่งขนาดของข้อมูลจะใหญ่มากและต้องการความต่อเนื่องของข้อมูลในการส่ง การสื่อสารบนเครือข่ายอินเทอร์เน็ตทั่วไปไม่สามารถรองรับการสื่อสารดังกล่าวได้ ดังนั้นจำเป็นจะต้องมีรูปแบบที่รองรับการสื่อสารแบบเรียลไทม์ขึ้น

การสื่อสารแบบเรียลไทม์หมายความว่า ในการส่งข้อมูลเราไม่จำเป็นที่จะต้องดาวน์โหลดข้อมูลทั้งหมดจากเซิร์ฟเวอร์มาก่อนจึงจะสามารถอ่านข้อมูลนั้นได้ แต่สามารถดาวน์โหลดพร้อม ๆ กับการอ่านข้อมูลนั้นได้เลย และยังสามารถควบคุมการอ่านขณะที่กำลังอ่านได้ด้วย เช่น สามารถเล่นไปข้างหน้า (Forward Play) หรือย้อนกลับ (Rewind Play) หยุดเล่นชั่วคราว (Pause) และหยุดเล่น (stop) เป็นต้น

2.3.1.1 ความสัมพันธ์ทางเวลา (Time Relationship)

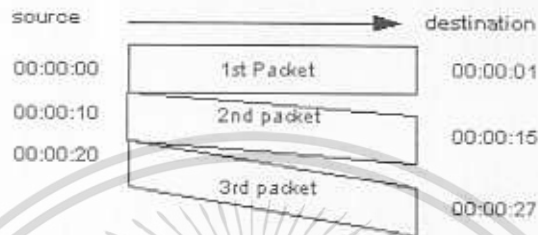
ในการส่งข้อมูลแบบเรียลไทม์ จำเป็นจะต้องคำนึงถึงความสัมพันธ์ของเวลาในการสื่อสารแต่ละครั้ง ยกตัวอย่างเช่น การส่งข้อมูลวิดีโอผ่านอินเทอร์เน็ต เซิร์ฟเวอร์จะแปลงสัญญาณวิดีโอให้อยู่ในรูปแบบของสัญญาณดิจิทัล จากนั้นทำการแบ่งข้อมูลดังกล่าวออกเป็นแพ็คเกจ (packet) ถ้ามีข้อมูลทั้งหมด 3 แพ็คเกจ แต่ละแพ็คเกจมีความยาวเท่ากับ 10 วินาที ดังนั้น ถ้าแพ็คเกจแรกเริ่มส่งที่เวลา 00.00.00 แพ็คเกจที่สองต้องส่งคือที่เวลา 00.00.10 และแพ็คเกจที่สามที่เวลา 00.00.20 เป็นต้น และในการรับข้อมูลใช้เวลา 1 วินาที ดังนั้นที่ฝั่งรับ จะรับแพ็คเกจแรกทีเวลา 00.00.01 แพ็คเกจที่สองที่เวลา 00.00.11 แพ็คเกจที่สามที่เวลา 00.00.21 ตามลำดับ ดังรูป



รูปที่ 2.6 ความสัมพันธ์เชิงเวลาของการสื่อสารแบบเรียลไทม์

แต่ในทางปฏิบัติการส่งจะต้องเกิดความล่าช้า (Delay) ขึ้น โดยในการส่งแต่ละครั้งจะเกิดความล่าช้าที่แตกต่างกันออกไปเช่น ในการส่งแพ็คเกจที่ 1 ฝั่งรับเริ่มรับข้อมูลที่เวลา 00.00.01 หมายความว่าเกิดความล่าช้า 1 วินาที แพ็คเกจที่ 2 รับข้อมูลที่เวลา 00.00.15 หมายความว่าเกิดความล่าช้า 5 วินาที แพ็คเกจที่ 3 รับข้อมูลที่เวลา 00.00.27 เกิดความล่าช้า 7 วินาที ถ้าที่ฝั่งรับเริ่มอ่านเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

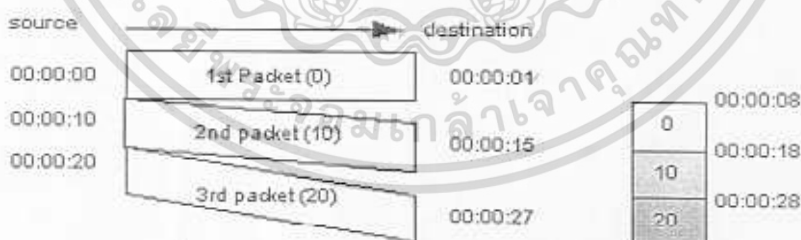
ข้อมูลของแพ็คเกจแรกใช้เวลา 00:00:01 เมื่อเวลา 00:00:11 ต้องอ่านแพ็คเกจที่สอง แต่ปรากฏว่าข้อมูลยังไม่ถึง ลักษณะเช่นนี้จะทำให้เกิดช่องว่างระหว่างการอ่านแพ็คเกจแรกกับแพ็คเกจต่อไป ช่องว่างดังกล่าวเรียกว่า “จิตเตอร์ (Jitter)” หรือความล่าช้าในระหว่างการรับข้อมูลแต่ละแพ็คเกจของรูปแบบการส่งข้อมูลแบบเรียลไทม์



รูปที่ 2.7 ปัญหาจิตเตอร์

2.3.1.2 ไทม์สแตมป์

การแก้ปัญหาที่เกิดจากจิตเตอร์แบบหนึ่งคือ การกำหนดไทม์สแตมป์หรือการกำหนดว่าจะเริ่มอ่านข้อมูลของแพ็คเกจนี้ที่เวลาใด หรือจะต้องอ่านข้อมูลของแพ็คเกจต่อไปอีกเมื่อใด เช่น แพ็คเกจแรกมีค่าไทม์สแตมป์เท่ากับ 0 แพ็คเกจที่สองมีค่าไทม์สแตมป์ 10 แพ็คเกจที่สามมีค่าไทม์สแตมป์ 20 ถ้าฝั่งรับเริ่มอ่านข้อมูลแรกเป็นเวลา 00:00:08 เพราะฉะนั้นอีก 10 วินาทีจึงเริ่มอ่านข้อมูลแพ็คเกจต่อไป คือเวลา 00:00:18 และแพ็คเกจที่สามที่เวลา 00:00:28 ดังรูป



รูปที่ 2.8 การแก้ปัญหาจิตเตอร์ โดยใช้ไทม์สแตมป์

2.3.1.3 เพลย์แบ็คบัฟเฟอร์ (Playback Buffer)

วิธีแก้ปัญหาที่เกิดจากจิตเตอร์อีกวิธีหนึ่งคือ การสร้างที่พักข้อมูลก่อนที่จะทำการอ่านข้อมูล (Playback Buffer) เมื่อเริ่มค้นกระบวนการส่งข้อมูล ฝั่งรับจะรอให้ข้อมูลของแพ็คเกจเข้ามาไว้ในบัฟเฟอร์จนเต็ม แล้วจึงเริ่มอ่านข้อมูลนั้น ๆ แพ็คเกจถัดไปก็จะถูกเก็บไว้ในบัฟเฟอร์เพื่อรอการเรียกอ่านต่อไป ขกตัวอย่างเช่น ทางฝั่งรับมีบัฟเฟอร์ที่สามารถจุแพ็คเกจความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยาว 7 วินาที เมื่อแพ็กเก็ตแรกที่มีความยาว 10 วินาทีมาถึง ข้อมูล 7 วินาทีแรกก็จะถูกเก็บไว้ในบัฟเฟอร์ ระบบก็จะเริ่มอ่านข้อมูลนั้น เมื่อเวลาผ่านไป 1 วินาที ข้อมูลของแพ็กเก็ตเดิมที่ช่วงเวลา 8 วินาทีก็จะถูกเติมมาในบัฟเฟอร์ ดังนั้นเมื่อเวลาผ่านไป 7 นาที ก็จะเหลือข้อมูลของแพ็กเก็ตอีก 3 วินาที ถ้าแพ็กเก็ตที่ 2 ล่าช้าไปไม่เกิน 3 วินาทีก็จะสามารถอ่านข้อมูลได้อย่างต่อเนื่อง

2.3.1.4 การเรียงลำดับ (Ordering)

การจะอ่านข้อมูลที่ถูกแบ่งและส่งมาเรื่อย ๆ บางครั้งข้อมูลอาจจะถูกส่งถึงในเวลาที่แตกต่างกัน ดังนั้นการอ่านจะต้องอ่านจากแพ็กเก็ตแรก และเรียงลำดับไปเรื่อย ๆ จึงต้องมีส่วนประกอบที่สำคัญอีกอย่างหนึ่งในระบบการสื่อสารแบบเรียลไทม์คือ เลขลำดับ (Sequence Number) ข้อดีอีกอย่างของเลขลำดับคือ ทำให้สามารถตรวจสอบได้ว่าข้อมูลที่ถูกส่งมาได้รับครบหรือไม่ ยกตัวอย่างเช่น ถ้าแพ็กเก็ตที่หนึ่ง สองและสามมีค่าไทม์สแตมป์เป็น 20 ,10 ,0 ถ้าแพ็กเก็ตที่สองสูญหายไป ทางฝั่งรับจะไม่สามารถตรวจสอบได้ว่าแพ็กเก็ตที่สองหายไป แล้วจะคิดว่าแพ็กเก็ตแรกมีความยาวเท่ากับ 20 หน่วยเวลา ดังนั้นจะต้องเพิ่มส่วนของเลขลำดับขึ้นมาเพื่อจะได้ทราบว่าแพ็กเก็ตที่สองได้สูญหายไปแล้ว

2.3.1.5 การแปลงสัญญาณ (Translation)

ความเร็วหรือความล่าช้าของการสื่อสารแบบเรียลไทม์นั้น ขึ้นอยู่กับปัจจัยหนึ่งคือ ขนาดของแบนด์วิธของช่องสัญญาณ หากข้อมูลจากแหล่งกำเนิดมีคุณภาพสูงต้องการอัตราการส่ง 5 เมกกะบิตต่อวินาที แต่แบนด์วิธในขณะที่ส่งสามารถส่งได้แค่ 1 เมกกะบิตต่อวินาทีหรือน้อยกว่า ดังนั้นจึงต้องมีส่วนของตัวแปลงสัญญาณ (Translator) ทำการแปลงสัญญาณให้มีคุณภาพลดลงไป เพื่อให้สามารถส่งข้อมูล ได้เมื่อแบนด์วิธของช่องสัญญาณลดลงเป็นต้น

2.3.1.6 การผสมสัญญาณ (Mixing)

ถ้ามีการส่งข้อมูลจากหลายแหล่งพร้อมกัน เช่น ส่งสัญญาณวิดีโอพร้อม กับสัญญาณเสียง หรือการประชุมทางไกลผ่านอินเทอร์เน็ต (Video Conference) ทำให้มีสตรีมข้อมูลจำนวนมากถูกส่งออกมาพร้อมกัน ทำให้ระบบเกิดความคับคั่ง วิธีการผสมสัญญาณ (Mixing) จึงเกิดขึ้นเพื่อลดความคับคั่งของข้อมูล โดยใช้วิธีรวมสตรีมหลาย ๆ สตรีมเข้าด้วยกัน แล้วส่งออกเป็นหนึ่งสตรีมเสมือนเป็นการส่งข้อมูลจากแหล่งกำเนิดข้อมูลตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 โพรโทคอลอาร์ทีพี (RTP Protocol)

โพรโทคอลอาร์ทีพีที่ถูกสร้างโดยองค์กรไอทีทีเอฟในอาร์เอฟซี1889 ขณะนี้อาร์ทีพีได้รับการยอมรับว่าเป็นมาตรฐานสำหรับการส่งข้อมูลมัลติมีเดียแบบเรียลไทม์ สามารถทำงานได้ทั้งยูนิคาสต์และมัลติคาสต์ ข้อมูลที่ส่งไปจะถูกดูแลโดยโพรโทคอลอีกชนิดหนึ่งที่ทำงานร่วมกับอาร์ทีพีที่มีย่อชื่อว่า โพรโทคอลอาร์ทีซีพี (RTCP Protocol) ซึ่งจะกล่าวถึงในหัวข้อ 2.3.3

ข้อมูลจากแพ็คเกจอาร์ทีพีที่มาถึงฝั่งผู้รับ จะถูกดูแลโดยโพรโทคอลอาร์ทีซีพี ซึ่งทำหน้าที่คอยแจ้งให้ชั้นอาร์ทีพีทราบ เพื่อปรับการเข้ารหัสและตัวแปรในการส่ง (Transmission Parameter) ให้เหมาะสม ตัวอย่างเช่น ถ้าตัวแปรในชั้นอาร์ทีซีพีตรวจเจอว่ามีแพ็คเกจสูญหายไปมาก มันอาจแจ้งให้ชั้นอาร์ทีพีปรับลดค่าอัตราการส่งข้อมูล ถึงแม้ว่าอาร์ทีพีจะเหมาะสมในการเล่นมีเดียแบบเรียลไทม์ แต่อาร์ทีพีไม่ได้มีกลไกในการรับประกันการส่งใด ๆ ทั้งสิ้น กลไกนี้จะขึ้นอยู่กับบริการในชั้นล่างลงมา มันถือว่าเครือข่ายนั้นเชื่อถือได้และแพ็คเกจที่ส่งนั้นมาถึงที่หมายถูกต้องตามลำดับ

อาร์ทีพีถูกออกแบบมาโดยจุดประสงค์หลักเพื่อการประชุมผ่านเครือข่าย และยังให้บริการสำหรับข้อมูลต่อเนื่อง การจำลองการโต้ตอบแบบกระจาย (Interactive Distributed Simulation) และ แอปพลิเคชันสำหรับการวัดคุม (Control and Measurement Application)

V	P	X	CC	M	Payload type	Sequence number
Time-stamp						
Synchronisation source identifier (SSRC)						
(First) Contributing Source Identifier (CSRC)						
...						
(Last) Contributing Source Identifier (CSRC)						

V : Version

P : Padding

CC : Contributor count

M : Marker

รูปที่ 2.9 รูปแบบส่วนหัวของแพ็คเกจอาร์ทีพี

จากรูปที่ 2.9 รูปแบบส่วนหัว (Header) ของแพ็คเกจอาร์ทีพีประกอบด้วยเขตข้อมูลต่าง ๆ

ดังนี้

1. "Version" (V) เวอร์ชันของโพรโทคอลอาร์ทีพีในปัจจุบันที่ใช้คือ เวอร์ชัน 2
2. "Padding" (P) หากบิตนี้ถูกเซตให้มีค่าเท่ากับ 1 นั้นหมายความว่า ในส่วนท้ายของแพ็คเกจจะมีส่วนของแพคคิงค่อท้ายมาด้วย และบิตสุดท้ายจะเป็นจำนวนบิตของแพคคิงทั้งหมด โดยปกติแล้วแพคคิงจะถูกใช้ในกรณีที่แพ็คเกจถูกเข้ารหัสมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. “Extension” (X) ถ้าบิตนี้ถูกเซตให้มีค่าเท่ากับ 1 หมายความว่า จะมีส่วนของ Extension header เพิ่มเข้ามาระหว่างส่วนของเฮดเดอร์และข้อมูล
4. “Contributor Count” (CC) คือจำนวนของแหล่งข้อมูลในแพ็คเกจ ในกรณีที่มีเดียสตรีมมีหลายแหล่งที่มาเช่น ภาพและเสียง ข้อมูลในฟิลด์นี้จะมีค่าได้สูงสุดเท่ากับ 15 เนื่องจากมีจำนวนบิตเท่ากับ 1 บิต
5. “Marker” (M) บิตนี้จะถูกเซตให้มีค่าเป็น 1 เมื่อบิตนั้นมีความจำเป็นที่จะต้องแสดงให้แอฟพลิเคชันเห็นเป็นพิเศษเช่น แพ็คเกจนี้คือข้อมูลชุดสุดท้ายของสตรีม
6. “Payload Type” เขตข้อมูลนี้มีความยาว 8 บิต จะแสดงชนิดข้อมูลของแพ็คเกจว่าถูกเข้ารหัสมาในรูปแบบใด และจะถูกส่งให้แอฟพลิเคชันใดทำการอ่านต่อไป

ชนิด	Application	ชนิด	Application	ชนิด	Application
0	PCM audio	7	LPC audio	15	G728 audio
1	1016	8	PCMA audio	16	Motion JPEG
2	G721 audio	9	G722 audio	31	H.261
3	GSM audio	10-11	L16 audio	32	MPEG1 video
5-6	DV14 audio	14	MPEG audio	33	MPEG2 video

ตารางที่ 2.2 ชนิดของข้อมูลในแพ็คเกจอาร์ทีพี

7. “Sequence Number” เขตข้อมูลนี้มีความยาว 16 บิต จะแสดงว่า แพ็คเกจอาร์ทีพีนี้มีหมายเลขใด โดยแพ็คเกจแรกเลขลำดับนี้จะถูกส่งขึ้นมาและจะเพิ่มขึ้นทีละ 1 สำหรับแพ็คเกจชุดเดียวกันที่ถูกส่งมาเป็นลำดับถัดไป ใช้สำหรับตรวจว่าแพ็คเกจที่ถูกส่งมาว่าสูญหายหรือเกินมาหรือไม่
8. “Timestamp” เขตข้อมูลนี้มีความยาว 32 บิต ใช้แสดงถึงความสัมพันธ์ในเชิงเวลาของแพ็คเกจชุดเดียวกัน ก็จะแสดงว่าแพ็คเกจนั้นอยู่ในช่วงเวลาใดของข้อมูลชุดเดียวกัน ยกตัวอย่างเช่น ถ้ามีแพ็คเกจ 3 แพ็คเกจ แต่ละแพ็คเกจมีความยาว 10 วินาที ดังนั้นถ้าค่าไทม์สแตมป์ของแพ็คเกจแรกมีค่าเท่ากับ 0 แพ็คเกจต่อมาจึงมีค่าเท่ากับ 10 และ 20 ตามลำดับ เมื่อรับแพ็คเกจแรกมาแล้วจะอ่านค่าไทม์สแตมป์ไว้ และเมื่อแพ็คเกจต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกส่งมา ก็จะอ่านค่าไทม์สแตมป์แล้วนำมาคำนวณว่า แพ็คเก็ตแรกมีความยาวเท่าใด ในที่นี้ แพ็คเก็ตที่สองมีค่าต่างจากแพ็คเก็ตแรก 10 เพราะฉะนั้น แพ็คเก็ตแรกมีความยาว 10 หน่วยเวลา และในขณะที่อ่านนี้อยู่ในช่วงใดของข้อมูล โดยค่าของช่วงเวลาดังกล่าว จะขึ้นอยู่กับแอฟพลิเคชันนั้น ๆ ว่าจะกำหนดช่วงเวลาให้มีความยาวเท่าใด โดยทั่วไปแล้วข้อมูลเสียงจะถูกสร้างขึ้นด้วยอัตราเร็ว 160 ไบต์ต่อ 1 ช่วงเวลา ดังนั้น การอ่านข้อมูลชนิดนี้จะต้องอ่านในช่วงเวลาเดียวกัน ดังนั้นค่าไทม์สแตมป์ของข้อมูลชนิดนี้ จึงต้องเพิ่มขึ้นทีละ 160 ไบต์ละแพ็คเก็ต

9. "Synchronization Source Identifier" ถ้าแพ็คเก็ตที่ถูกส่งมาจากแหล่งกำเนิดสัญญาณแหล่งเดียว ค่าของเขตข้อมูลนี้จะเป็นหมายเลขของแหล่งกำเนิดนั้น ๆ (ขนาด 32 บิต) แต่ถ้าข้อมูลมาจากหลาย ๆ แหล่ง แล้วทำการรวมสัญญาณนั้น ๆ ให้อยู่ในแพ็คเก็ตเดียวกัน ค่าของเขตข้อมูลจะเป็นหมายเลขของแหล่งกำเนิดใหม่ที่เกิดจากการรวมสัญญาณ โดยค่าดังกล่าวจะถูกส่งออกมาจากแหล่งกำเนิด และทางฝั่งรับจะทราบได้ว่า ข้อมูลนี้เกิดจากแหล่งกำเนิดมากกว่าหนึ่งแหล่งและต้องอ่านในเวลาเดียวกัน (ข้อมูลจากหลายแหล่งจะมีเลขลำดับเดียวกัน)
10. "Contributor Identifier" เขตข้อมูลนี้มีความยาว 32 บิต มีไว้เพื่อบอกว่าข้อมูลนี้เกิดจากแหล่งกำเนิดข้อมูลใดบ้าง ใช้ในกรณีที่ข้อมูลมาจากแหล่งกำเนิดมากกว่าหนึ่งและข้อมูลนั้นถูกรวมกับข้อมูลจากแหล่งอื่น ๆ

2.3.3 โพรโทคอลอาร์ทีซีพี (RTCP Protocol)

มีหน้าที่หลักคือ การควบคุมการส่งแพ็คเก็ตของการสื่อสารแบบเรียลไทม์ โดยใช้หลักการทำงานเช่นเดียวกับการส่งแพ็คเก็ตข้อมูลทั่วไป หน้าที่หลักของอาร์ทีซีพี มีดังนี้คือ

1. ตรวจสอบคุณภาพของการสื่อสารและควบคุมความคับคั่งของข้อมูล
2. ระบุข้อมูลที่จำเป็นสำหรับการส่งสตรีม
3. จัดการเกี่ยวกับขนาดของเซสชัน

อาร์ทีซีพีจะแสดงเฉพาะข้อมูลเกี่ยวกับคุณภาพของการสื่อสาร ผ่านรายงานสถานะของผู้ส่ง (Sender Reports (SR)) และรายงานสถานะของผู้รับ (Receiver Report (RR)) เพื่อตรวจสอบว่ามีแพ็คเก็ตใดสูญหายไปหรือไม่ เกิดความล่าช้าหรือไม่ และมีจิตเตอร์หรือไม่ การทำงานดังกล่าวคล้ายกับการทำงานของทีซีพี แตกต่างกันที่อาร์ทีซีพีจะทำงานร่วมกับยูดีพี และเป็นเพียงแค่การรายงานว่าการสื่อสารขณะนี้เป็นเช่นไร แต่จะไม่มีการสั่งให้ระบบส่งแพ็คเก็ตที่สูญหายไปใหม่อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Version	P	Count	Type	Length
Data				

รูปที่ 2.10 รูปแบบส่วนหัวของแพ็กเก็ตอาร์ทีซีพี

จากรูปที่ 2.10 รูปแบบส่วนหัว (Header) ของแพ็กเก็ต RTCP เขตข้อมูลที่ต่างจาก RTP มีดังนี้

1. “Count” เขตข้อมูลนี้มีขนาด 5 บิต โดยที่ค่าในเขตข้อมูลนี้ แสดงถึงจำนวนของรายงานการรับแพ็กเก็ต (reception report) ที่บรรจุอยู่ในแพ็กเก็ตนี้ หากชนิดของแพ็กเก็ต (เขตข้อมูล “Type”) เป็นรายงานสถานะของผู้รับ (RR, Receiver Report)
2. “Type” เขตข้อมูลนี้มีขนาด 8 บิต ค่าในเขตข้อมูลนี้แสดงถึงชนิดของแพ็กเก็ตอาร์ทีซีพี ซึ่ง สามารถดูชนิดของแพ็กเก็ตอาร์ทีซีพีทั้งหมดได้จากตารางที่ 2.3

ชนิด	คำอธิบาย
192	ร้องขอเฟรมใหม่ทั้งเฟรม (FIR, full INTRs-frame request)
193	แจ้งสถานะปฏิเสธ (NACK, negative acknowledgement)
200	รายงานสถานะของผู้ส่ง (SR, sender report)
201	รายงานสถานะของผู้รับ (RR, Receiver report)
202	คำอธิบายแหล่งกำเนิดข้อมูล (SDS, source description)
203	ข้อความจบการสื่อสาร (BYE, goodbye)
204	ประกาศแอปพลิเคชัน (APP, application defined)
207	ส่วนขยายของอาร์ทีซีพี (XR, RTCP extension)

ตารางที่ 2.3 ชนิดของแพ็กเก็ตอาร์ทีซีพี

3. “Length” เขตข้อมูลนี้มีขนาด 16 บิต ค่าในเขตข้อมูลนี้จะแสดงถึงขนาดของแพ็กเก็ตอาร์ทีซีพี รวมส่วนหัวและส่วนแพคคิงที่ต่อท้ายมาด้วย เพื่อป้องกันไม่ให้เกิดแพ็กเก็ตที่มีความยาวเท่ากับศูนย์ และยังป้องกันการเกิดการวนรอบจำกัด (Finite Loop) ใ้้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 โพรโทคอลอาร์ทีเอสพี (RTSP Protocol)

โพรโทคอลอาร์ทีเอสพีนี้เป็นโพรโทคอลที่ใช้ในการรับส่งข้อมูลมัลติมีเดียระหว่างเซิร์ฟเวอร์กับคอมพิวเตอร์ปลายทาง ซึ่งจะทำการรับส่งข้อมูลต่อเนื่องผ่านอินเทอร์เน็ต โดยทางเซิร์ฟเวอร์ด้านผู้ส่ง สามารถส่งข้อมูลไปให้ผู้รับปลายทางเพียงคนเดียว หรือส่งไปให้ผู้รับหลายคนในลักษณะเป็นกลุ่มก็ได้ ซึ่งอาร์ทีเอสพีถูกกำหนดให้เป็นโพรโทคอลที่นำไปใช้ในอินเทอร์เน็ตเอ็นจิเนียอร์ทาสก์ฟอร์ซ หรือไออีทีเอฟ (Internet Engineer Task Force, IETF) ในเดือนเมษายน ปี ค.ศ. 1998 เรียกว่า อาร์เอฟซี 2326 (RFC2326)

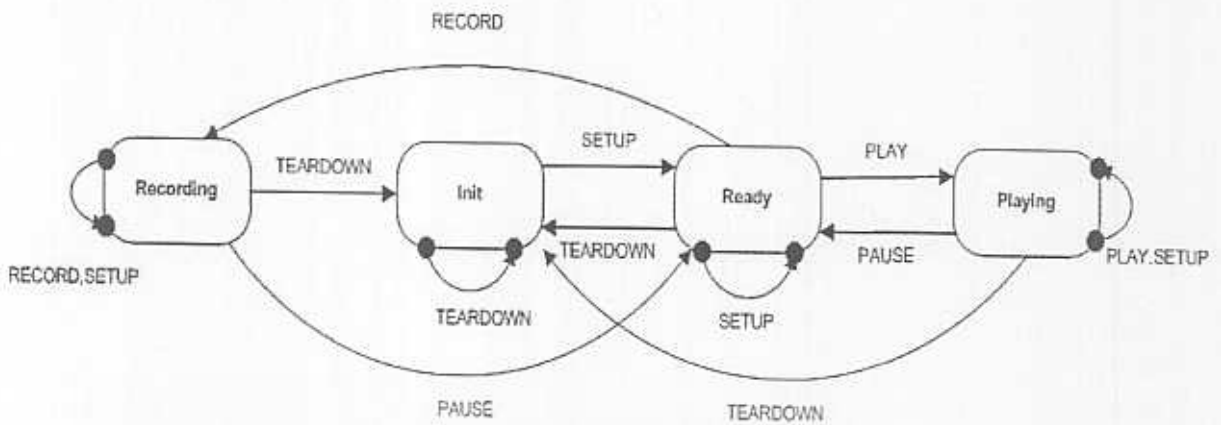
อาร์ทีเอสพีเป็นการกำหนดมาตรฐานโพรโทคอลที่สำคัญมากในการรับส่งข้อมูลมัลติมีเดียผ่านอินเทอร์เน็ต เนื่องจากการรับส่งข้อมูลมัลติมีเดียในแบบต่อเนื่องนั้น มีส่วนต่าง ๆ ที่เกี่ยวข้องกันถึง 3 ส่วน คือ เซิร์ฟเวอร์ที่เก็บข้อมูล, เอนโคเดออร์ที่ใช้เข้ารหัสข้อมูลและผู้รับข้อมูลหรือเพลย์เยอร์ (Player) ตัวเข้ารหัสข้อมูลนั้นจะต้องเข้ารหัสข้อมูลมัลติมีเดียเก็บลงในไฟล์ โดยมีฟอร์แมตที่เซิร์ฟเวอร์ที่เรียกใช้งานได้ และเมื่อเซิร์ฟเวอร์ต้องการส่งข้อมูลนี้ไปให้ผู้รับ ก็จะต้องใช้โพรโทคอลรับส่งข้อมูลอย่างต่อเนื่องที่ผู้รับเข้าใจ และสามารถรับข้อมูลได้อย่างถูกต้อง จากนั้นเมื่อได้รับข้อมูลมาแล้วก็จะต้องถอดรหัสข้อมูลออกแสดงผลได้ โดยใช้มาตรฐานเดียวกันกับตัวเข้ารหัส การทำงานทั้งหมดจึงผูกพันกัน ซึ่งอาร์ทีเอสพีจะอยู่ในส่วนโพรโทคอลที่ใช้รับส่งข้อมูลจากเซิร์ฟเวอร์ไปให้ผู้รับนั่นเอง

แม้ว่าอาร์ทีเอสพีจะมีความสำคัญในการรับส่งข้อมูลมัลติมีเดียผ่านอินเทอร์เน็ต ก็ตาม แต่ก็ไม่ใช่สิ่งเดียวที่ทำให้การรับส่งข้อมูลสมบูรณ์ได้ เรายังต้องการฟอร์แมตมาตรฐานของไฟล์ที่ใช้เก็บข้อมูลมัลติมีเดียอีกด้วยเช่น แอคทีฟสตรีมมิงฟอร์แมต (Active Streaming Format, ASF) ของไมโครซอฟต์ ลวิกโทม หรืออื่น ๆ เพื่อเก็บข้อมูล รวมถึงมาตรฐานการเข้ารหัสข้อมูลเช่น เอ็มเปก สำหรับใช้เข้ารหัสข้อมูลมัลติมีเดียเก็บลงไฟล์อีกด้วย

อาร์ทีเอสพีนี้จะทำหน้าที่ควบคุมการรับส่งข้อมูลมัลติมีเดียอย่างต่อเนื่องระหว่างเซิร์ฟเวอร์ที่เก็บข้อมูลกับคอมพิวเตอร์ผู้รับข้อมูล โดยมีไฟล์ฟอร์แมตสำหรับเก็บข้อมูล เช่น ลวิกโทม หรือไฟล์ฟอร์แมตอื่น ๆ ซึ่งการสร้างไฟล์ข้อมูลมัลติมีเดียนี้ ก็ได้มาจากการนำข้อมูลมาทำการเข้ารหัสโดยใช้ตัวเข้ารหัสอย่างเช่นเอ็มเปกนั่นเอง ซึ่งในปัจจุบันนี้ ซอฟต์แวร์ที่ใช้สำหรับเล่นข้อมูลมัลติมีเดียจะใช้อาร์ทีเอสพีในการรับส่งข้อมูลทั้งหมดทั้งนั้น ยกตัวอย่างเช่น ลวิกโทมเวอร์ชันโฟร์ (QuickTime Version 4), เรียลซิชเต็มจีทู (Real System G2) และวินโดวส์มีเดียเพลย์เยอร์ (Windows Media Player) เป็นต้น

รูปที่ 2.11 แสดงถึงสถานะต่าง ๆ ของเซิร์ฟเวอร์ ดูตารางที่ 2.4 แสดงเมธอดต่าง ๆ ของโพรโทคอลอาร์ทีเอสพีประกอบ

เนื่องจากมีความสัมพันธ์ในการเปลี่ยนสถานะของเซิร์ฟเวอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 สถานะของเซิร์ฟเวอร์

เมธอด	คำอธิบาย
Options	ขออนุมัติทั้งหมดที่มี
SETUP	สร้างช่องทางและจัดสรรทรัพยากรสำหรับการส่ง
ANNOUNCE	เปลี่ยนแปลงคำอธิบายของมัลติมีเดีย
DESCRIBE	ขอคำอธิบายของมัลติมีเดีย
PLAY	เริ่มเล่นมัลติมีเดีย
RECORD	เริ่มบันทึกมัลติมีเดีย
REDIRECT	ย้ายโกลบอลไปที่เซิร์ฟเวอร์ตัวใหม่
PAUSE	หยุดการส่งข้อมูลชั่วคราว แต่ยังคงหาสถานะไว้
SET-PARAMETER	ควบคุมการส่ง
TEARDOWN	ยกเลิกสถานะทั้งหมด

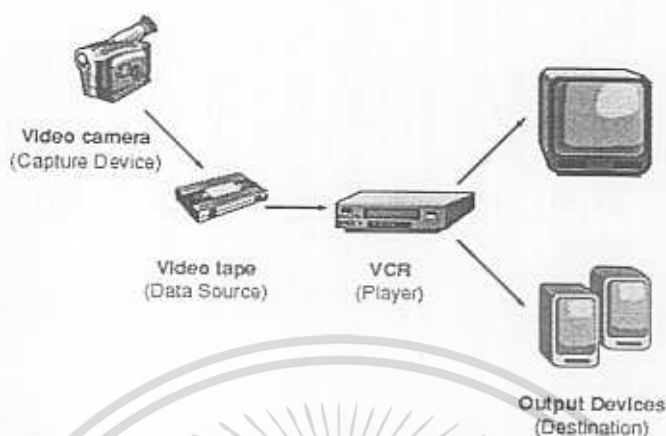
ตารางที่ 2.4 เมธอดของโปรโตคอลอาร์ทีเอสพี

2.4 จาวามีเดียเฟรมเวิร์ก (Java Media Framework, JMF)

การทำงานของจาวามีเดียเฟรมเวิร์กหรือเจเอ็มเอฟ มีหลักการทำงานคล้ายกับการทำงานของระบบวิดีโอ ตัวอย่างเช่น เมื่อเราต้องการดูหนังจากเครื่องเล่นวิดีโอ เราจะต้องใส่แผ่นเทปเข้าไปในเครื่องเล่นวิดีโอ จากนั้นเครื่องเล่นวิดีโอจะทำการอ่านข้อมูลจากเทป แล้วแสดงข้อมูลในรูปแบบของสัญญาณภาพออกทางจอโทรทัศน์และสัญญาณเสียงออกทางลำโพงตามลำดับ เจเอ็มเอฟก็ใช้หลักการเช่นเดียวกับระบบวิดีโอทั่วไป เริ่มตั้งแต่การจับภาพและเสียงโดยใช้กล้องวิดีโอ (Capture Device) และนำข้อมูลที่ได้อ่านเก็บไว้คาต้าซอร์ส ซึ่งเปรียบเสมือนม้วนวิดีโอเทป จากนั้นคาต้าซอร์สจะถูกนำไปประมวลผลโดยเพลย์เออร์ เพื่อให้สามารถแสดงผลหรือออกทางอุปกรณ์แสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Output Device) ต่าง ๆ ซึ่งการทำงานของระบบวีดีโอและการทำงานของเจเอ็มเอฟสามารถแสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 เปรียบเทียบการทำงานของกล้องวีดีโอกับเจเอ็มเอฟ

2.4.1 การจับข้อมูลมีเดีย

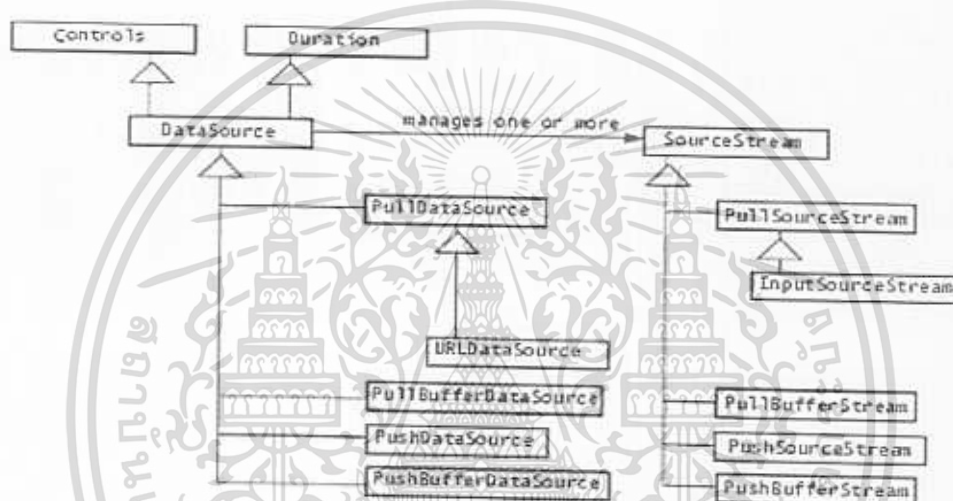
อุปกรณ์ที่ทำกรจับข้อมูลมีเดีย จะทำหน้าที่เป็นแหล่งของข้อมูลมีเดียสำหรับการทำงานของเจเอ็มเอฟ และข้อมูลที่ได้จากอุปกรณ์จับข้อมูลมีเดียในเจเอ็มเอฟจะแทนด้วยชื่อดาต้าซอร์ส (DataSources) ตัวอย่างของอุปกรณ์จับข้อมูลมีเดียเช่น การ์ดเสียงเป็นอุปกรณ์ที่ใช้ในการจับเสียงจากไมค์โครโฟนแล้วแปลงมาเป็นข้อมูล ในการจับข้อมูลมีเดียจากอุปกรณ์จับข้อมูลมีเดียที่เราต้องการใช้ เราจะต้องดึงเอามีเดียโลเคเตอร์ (MediaLocator) ของอุปกรณ์นั้น ๆ มาจากออบเจกต์ "CaptureDeviceInfo" ซึ่งได้มาจากคลาส "CaptureDeviceManager" จากนั้นจึงสามารถนำมีเดียโลเคเตอร์ที่ได้นี้ ไปสร้างเพลย์เยอร์ (Player) หรือโปรเซสเซอร์ (Processor) โดยตรง หรือใช้มีเดียโลเคเตอร์นี้สร้างดาต้าซอร์ส ซึ่งเราจะใช้เป็นอินพุตให้กับเพลย์เยอร์หรือโปรเซสเซอร์ และเราจะต้องเรียกใช้เมธอด "start()" ของเพลย์เยอร์หรือโปรเซสเซอร์ เพื่อเริ่มต้นทำการจับข้อมูล

2.4.2 ดาต้าซอร์ส

เอพีไอของเจเอ็มเอฟ (JMF API) จะใช้ดาต้าซอร์สในการจัดส่งข้อมูลมีเดีย โดยที่ดาต้าซอร์สนั้นจะทำการเอนแคพซูลตำแหน่งของข้อมูล รวมไปถึงโปรโตคอลและซอฟต์แวร์ที่ใช้ในการส่ง ซึ่งในการได้มาซึ่งข้อมูลแต่ละครั้งนั้น แหล่งข้อมูล (Resource) หนึ่ง ๆ จะสามารถส่งให้กับสื่อได้เพียงตัวใดตัวหนึ่งเท่านั้น ไม่สามารถนำมาส่งให้กับสื่อตัวอื่นได้

ดาต้าซอร์สนั้นจะถูกแสดงโดยเจเอ็มเอฟ มีเดียโลเคเตอร์หรือยูอาร์แอล (Universal Resource Locator, URL) ตัวใดตัวหนึ่ง มีเดียโลเคเตอร์นั้นมีความคล้ายคลึงกับยูอาร์แอล และยังคงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถสร้างได้จากยูอาร์แอลได้ด้วย ซึ่งจะถูกสร้างขึ้นก็ต่อเมื่อตัวจัดการ โปรโตคอลที่มีลักษณะ เช่นเดียวกัน (Corresponding Protocol Handler) นั้นไม่ถูกติดตั้งอยู่ในระบบ กล่าวคือ ในภาษาจาวา นั้นยูอาร์แอลถูกสร้างได้เพียงตัวเดียวเท่านั้น ถ้าตัวจัดการ โปรโตคอลที่มีลักษณะเช่นเดียวกันถูก ติดตั้งอยู่ในระบบ คาต้าซอร์สจะเป็นตัวจัดการกับกลุ่มของออบเจกต์ "SourceStream" ถ้าเป็นข้อมูล ตามมาตรฐาน (Standard Data Source) จะมีการเก็บข้อมูลที่เอาไว้ใช้ในการส่งเป็นแบบไบต์อาร์เรย์ ถ้าเป็นแบบข้อมูลที่มีการพักเก็บไว้ก่อน (Buffer Data Source) จะใช้ออบเจกต์ "Buffer" เป็นตัวส่ง ข้อมูล (unit of transfer) ซึ่งเจเอ็มเอฟได้กำหนดชนิดของออบเจกต์ "DataSource" ไว้ดังรูปที่ 2.13



รูปที่ 2.13 คาต้าโมเดลของเจเอ็มเอฟ

2.4.2.1 คาต้าซอร์สแบบพูลและแบบพุช

ข้อมูลที่เรารู้ได้มานั้นอาจจะได้มาจากแหล่งข้อมูลที่หลากหลาย อาทิเช่น ได้มาจากข้อมูลในเครื่องที่เราที่มีอยู่แล้วหรือจากระบบเครือข่ายหรือไม่ก็จากการ broadcast ซึ่ง คาต้าซอร์สในเจเอ็มเอฟ สามารถแบ่งออกเป็น 2 ประเภทตามวิธีการส่งข้อมูลคือ คาต้าซอร์สแบบ พูล (Pull Data Source) และคาต้าซอร์สแบบพุช (Push Data Source)

1. คาต้าซอร์สแบบพูล

วิธีการเริ่มส่งข้อมูลแบบนี้ ไคลเอนท์จะเป็นคนเริ่มสั่งให้มีการส่งข้อมูล และอัตรา ของการส่ง ไคลเอนท์สามารถที่จะควบคุมการส่งข้อมูลต่าง ๆ ได้ โปรโตคอลที่ใช้ในการส่งแบบนี้ ก็จะเป็นพวกไฮเปอร์เท็กซ์ทรานสเฟอร์โปรโตคอล (Hypertext Transfer Protocol, HTTP) และไฟล์ (File) ในเจเอ็มเอฟมีการกำหนดชนิดของคาต้าซอร์สแบบพูลไว้สองแบบคือ "PullDataSource" และ "PullBufferDataSource" โดยจะให้ออบเจกต์ของคลาส "Buffer" เป็นตัวที่ใช้ส่งข้อมูล ข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกต่างของทั้งสองแบบนี้ คือการส่งข้อมูลทาง “PullDataSource” จะเป็นการส่งแบบต่อเนื่องเรื่อยๆ แต่ถ้าเป็นแบบ “PullBufferDataSource” จะส่งข้อมูลมาทีละขนาดบัฟเฟอร์

2. คาด้าซอร์สแบบพุช

วิธีการเริ่มส่งข้อมูลแบบนี้ เซิร์ฟเวอร์จะเป็นผู้เริ่มสั่งให้มีการส่งข้อมูล และกำหนดอัตราของการส่ง ซึ่งคาด้าซอร์สแบบพุชนี้ จะประกอบไปด้วยบรอดคาสต์มีเดีย (Broadcast Media) มัลติคาสต์มีเดีย (Multicast Media) และวิดีโอออนดีมานด์ (video-on-demand ,VOD) สำหรับการส่งข้อมูลจำพวกบรอดคาสต์นี้ ก็จะมีโปรโตคอลที่เรียกว่า เรียลไทม์ทรานสปอร์ตโปรโตคอล (Real Time Transport Protocol, RTP) ซึ่งโปรโตคอลตัวนี้ถูกพัฒนาโดยไอทีทีเอฟ เจเอ็มเอฟได้ทำการกำหนดชนิดของคาด้าซอร์สแบบพุชเอาไว้สองแบบคือ แบบ “PushDataSource” และแบบ “PushBufferDataSource” โดยจะให้ ออบเจกต์ของคลาส “Buffer” เป็นตัวที่ใช้ส่งข้อมูล ตัวอย่างที่แสดงให้เห็นถึงความแตกต่างที่ชัดเจนของการส่งข้อมูลทั้งสองแบบนี้คือ ถ้าเรามีไฟล์เอ็มเปกที่เป็นคาด้าซอร์สแบบพุช จะเป็นการอนุญาตให้ผู้ใช้สามารถที่จะทำการเล่นซ้ำ (Replay) หรือทำการเลือกดูข้อมูลส่วนไหนก็ได้ของไฟล์ในทางตรงกันข้าม ถ้าเป็นไฟล์เอ็มเปกที่เป็นคาด้าซอร์สแบบพุชตัวนี้ จะถูกเก็บไว้ที่เซิร์ฟเวอร์ เซิร์ฟเวอร์จะบรอดคาสต์ไปให้ผู้ชมแต่ละคนดู ผู้ใช้ไม่สามารถทำการเล่นซ้ำหรือเลือกดูข้อมูลแบบสุ่มได้ ผู้ใช้ต้องดูไปเรื่อย ๆ ตามแต่เซิร์ฟเวอร์จะส่งมาให้ เหมือนกับการถ่ายทอดโทรทัศน์

2.4.2.2 คาด้าซอร์สชนิดพิเศษ

เจเอ็มเอฟได้กำหนดคาด้าซอร์สชนิดพิเศษเอาไว้สองชนิดคือ คาด้าซอร์สที่สามารถโคลนได้ (Cloneable Data Sources) และคาด้าซอร์สผสม (Merging Data Sources)

1. คาด้าซอร์สที่สามารถโคลนได้

มีไว้สำหรับการสร้างโคลน (Clone) ไม่ว่าจะเป็นคาด้าซอร์สแบบพุชหรือแบบพุช การสร้างคาด้าซอร์สที่สามารถโคลนได้ เราจะทำการเรียก เมธอด “createCloneableDataSource” จากคลาส “Manager” จากนั้นก็ส่งคาด้าซอร์สที่เราจะโคลนไปให้กับเมธอดนี้ เมื่อคาด้าซอร์สที่จะทำการโคลนถูกส่งไปเรียบร้อยแล้ว เราสามารถเรียกใช้งานคาด้าซอร์สที่ทำการโคลนกับตัวโคลนของมันได้โดยไม่ต้องเรียกใช้งานโดยตรงกับคาด้าซอร์สดั้งเดิม คาด้าซอร์สที่สามารถโคลนได้จะสนับสนุนอินเทอร์เฟซ “SourceCloneable” ซึ่งถูกกำหนดโดยเมธอด “createClone” เมื่อเราทำการเรียกเมธอดนี้ เราจะสามารถกำหนดได้ว่าจะสร้างโคลนขึ้นมาจำนวนเท่าไร ซึ่งโคลนที่สร้างขึ้นมานี้ เราสามารถควบคุมได้โดยผ่านคาด้าซอร์สที่สร้างตัวมันขึ้นมา (เราสามารถใช่เมธอด “connect”, “disconnect”, “start” หรือ “stop” เพื่อเป็นการควบคุมคาด้าซอร์สที่สามารถโคลนได้) โคลนที่เรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

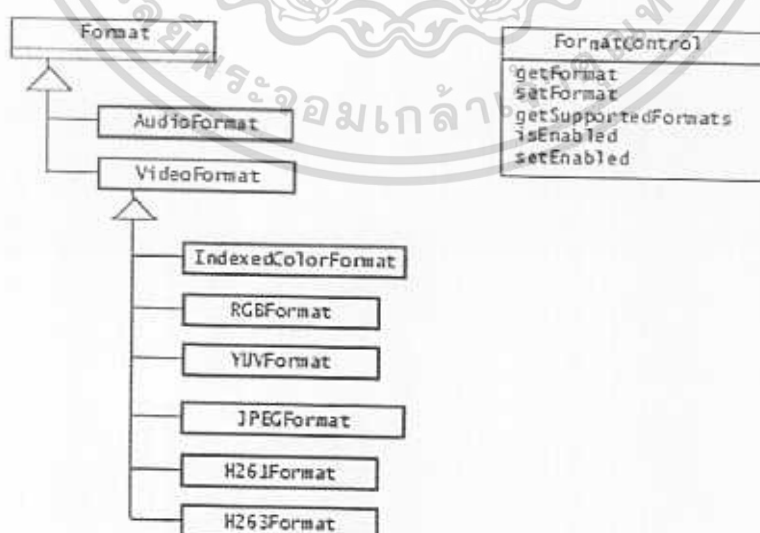
สร้างขึ้นมาไม่จำเป็นที่จะต้องมีความสัมพันธ์เหมือนกับคาล์ซอร์สต้นแบบที่ทำการโคลน กล่าวคือ สมมติว่าเราสร้าง โคลนสำหรับอุปกรณ์จับข้อมูลซึ่งมีฟังก์ชันเหมือนกับคาล์ซอร์สที่สามารถโคลน ได้ทุกประการ ในกรณีที่ไม่มีการเรียกใช้งานจากคาล์ซอร์สที่สามารถโคลนได้ ตัวโคลนของมันก็จะไม่ทำการผลิตข้อมูล แต่ถ้าเริ่มมีการใช้งานทั้งสองตัวคือ คาล์ซอร์สที่สามารถโคลนได้และตัว โคลนของมัน ตัวโคลนของมันก็จะทำการผลิตข้อมูลด้วยอัตราที่เทียบเท่ากับต้นแบบของมัน

2. คาล์ซอร์สแบบผสม

คาล์ซอร์สแบบผสม มีไว้เพื่อใช้สำหรับการรวม “SourceStream” จากคาล์ซอร์ส สองหรือสามตัวเข้าเป็นคาล์ซอร์สตัวเดียว โดยเราสามารถควบคุมกลุ่มของคาล์ซอร์สนี้ได้ เหมือนกับการควบคุมคาล์ซอร์สตัวเดียว ในการสร้างคาล์ซอร์สแบบผสมนี้ เราสามารถจะเรียกใช้ เมธอด “createMergingDataSource” จากคลาส “Manager” จากนั้นก็ส่งอาเรย์ที่ประกอบไปด้วย คาล์ซอร์สที่ที่ต้องการรวมเข้าด้วยกัน แต่มีข้อแม้ว่า คาล์ซอร์สที่จะทำการรวมกันนั้น จะต้องเป็น ชนิดเดียวกัน กล่าวคือเราไม่สามารถจะนำ “PullDataSource” มารวมกันกับ “PushDataSource” ได้

2.4.2.3 ฟอแมตของข้อมูล

รูปแบบของการเก็บข้อมูลของสื่อ นั้น จะถูกแสดงด้วยออบเจกต์หนึ่ง ที่เรียกว่าฟอแมต (Format) ซึ่งจะอธิบายถึงชื่อของการแปลงข้อมูลเป็นรหัสของรูปแบบของการ เก็บข้อมูล (format's encoding name) และชนิดของรูปแบบของการเก็บข้อมูลที่ต้องการ เจเอ็มเอฟ ใช้ออบเจกต์ “Format” ในการกำหนดรูปแบบของการเก็บข้อมูลของออดิโอและวิดีโอ ดังรูปที่ 2.14



รูปที่ 2.14 ฟอแมตของมีเดียในเจเอ็มเอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอททริบิวต์ “AudioFormat” จะอธิบายรูปแบบการเก็บข้อมูลของออดิโอ ซึ่งประกอบด้วย อัตราแซมปลิง จำนวนบิตต่อหนึ่งแซมเปิ้ล และจำนวนช่องสัญญาณเสียง ส่วน “VideoFormat” นั้นจะเป็นตัวเอนแคพซูเลทเกี่ยวกับข้อมูลที่มีความสัมพันธ์กันกับข้อมูลวิดีโอ ซึ่ง “VideoFormat” นี้จะเป็นตัวอธิบายเกี่ยวกับรูปแบบการเก็บข้อมูลพื้นฐานของวิดีโอ ดังนี้ “IndexedColorFormat”, “RGBFormat”, “YUVFormat”, “JPEGFormat”, “H261Format”, “H263Format”

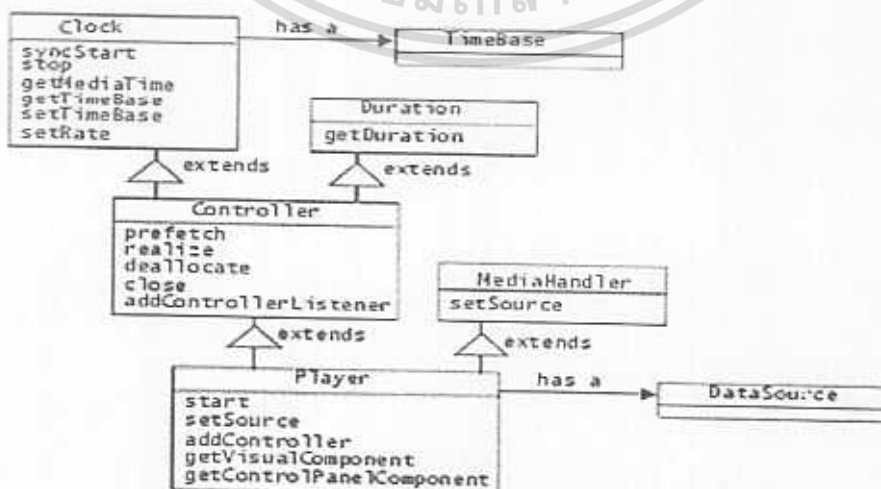
2.4.3 เพลย์เยอร์

เพลย์เยอร์ (Player) เป็นส่วนที่ใช้ในการประมวลผลอินพุตสตรีมของข้อมูล โดยจะแสดงผลได้ก็ต่อเมื่อคาล์บซอร์สนั้นได้จ่ายอินพุตสตรีมมายังเพลย์เยอร์ ในส่วนของการแสดงผลนั้นจะขึ้นอยู่กับชนิดของข้อมูลที่ใช้ในการแสดงผล



รูปที่ 2.15 แบบจำลองของเพลย์เยอร์ในเจเอ็มเอฟ

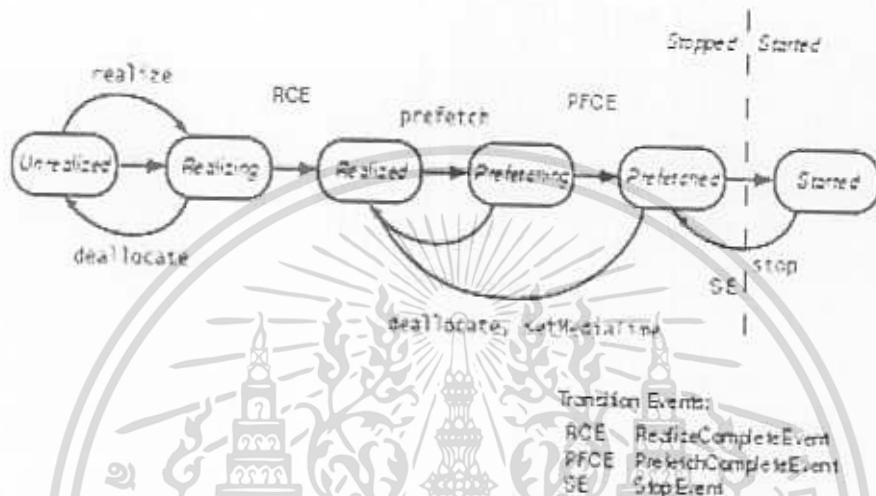
เพลย์เยอร์จะไม่สนใจว่า ข้อมูลจะถูกโพรเซสหรือเรนเดอร์ (Render) มาได้อย่างไร มันจะเป็นเพียงแค่ตัวควบคุมตามมาตรฐานทั่วไป ที่มีอยู่ในทุก ๆ อุปกรณ์ควบคุมข้อมูลต่าง ๆ ตัวอย่างเช่น เพลย์เยอร์จะมีแค่เพียงปุ่มเล่น (Play) และหยุด (Stop) เท่านั้น จะไม่มีอีควอไลเซอร์ (Equalizer) สำหรับปรับแต่งข้อมูล เราสามารถแสดงคลาสโคดอะแอมของเพลย์เยอร์ได้ดังรูปที่ 2.16



รูปที่ 2.16 คลาสโคดอะแอมของเพลย์เยอร์ในเจเอ็มเอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพลย์เยอร์ จะมีสถานะที่เป็นไปได้อยู่หกแบบด้วยกัน สถานะหลักสองสถานะ “Stopped” และ Started ซึ่งได้ถูกนิยามไว้ในอินเทอร์เน็ตเฟจ “Clock” แต่เพื่อให้ง่ายต่อการจัดการ ตัวควบคุมทรัพยากร (Resource Controller) จึงได้มีการแบ่งสถานะ “Stopped” ออกเป็น 5 สถานะย่อย คือ “Unrealized”, “Realizing”, “Realized”, “Prefetching”, และ “Prefetched” แสดงได้ดังรูปที่ 2.17



รูปที่ 2.17 สถานะของเพลย์เยอร์

ในการทำงานโดยทั่วไป เพลย์เยอร์จะต้องผ่านสถานะต่าง ๆ ตามขั้นตอนตั้งแต่ขั้นแรกไปจนถึง ขั้นตอนเริ่ม (Started) เพลย์เยอร์ ดังนี้

1. เมื่อเพลย์เยอร์ถูกสร้างขึ้นมาจะอยู่ในสถานะ “Unrealized” ซึ่งในตอนนี้ เพลย์เยอร์ยังไม่มีข้อมูลเกี่ยวกับมีเดียที่ตัวเองจะคิดต่อด้วย
2. เมื่อเมธอด “Realize” ถูกเรียก เพลย์เยอร์จะเปลี่ยนสถานะตัวเองจาก “Unrealized” เป็น “Realizing” ซึ่งในสถานะนี้ เพลย์เยอร์จะทำการตรวจสอบและพิจารณาความต้องการทรัพยากรของตนเอง ในกระบวนการนี้เพลย์เยอร์จะขอทรัพยากรที่ตนเองจะร้องขอเพียงแค่ครั้งเดียว เฉพาะทรัพยากรที่ไม่เป็นเอ็กซ์คลูซีฟยูส (Exclusive-use Resource) ซึ่งหมายถึง ทรัพยากรที่มีจำกัดที่สามารถถูกใช้ได้โดยเพลย์เยอร์เดียวในเวลาหนึ่ง ๆ เท่านั้น ซึ่งทรัพยากรเหล่านี้จะถูกขอในช่วงสถานะ “Prefetching”
3. เมื่อเพลย์เยอร์ผ่านสถานะ “Realizing” แล้ว ต่อมาคือสถานะ “Realized” ซึ่งในสถานะนี้ เพลย์เยอร์จะรู้ว่ามันต้องใช้ทรัพยากรอะไรบ้าง และรู้ว่าชนิดของข้อมูลที่จะใช้แสดงนั้นคืออะไร เนื่องจากเพลย์เยอร์ที่อยู่ในสถานะนี้ จะรู้วิธีการแสดงข้อมูลออกมา ดังนั้นมันจึงสามารถให้คอม โปเนนท์สำหรับแสดงผลและควบคุมได้เลย การเชื่อมต่อกับ

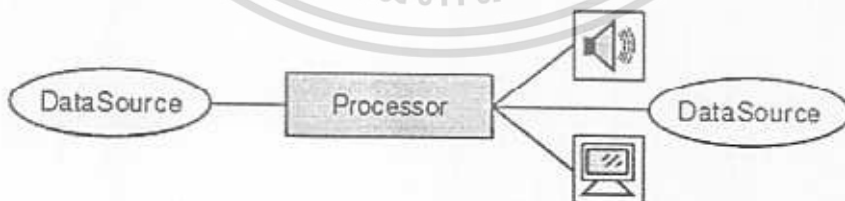
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจกต์อื่น ๆ ในระบบ ได้ถูกจัดเตรียมแล้ว แต่ว่าตัวเพลย์เยอร์เองยังไม่ได้ทำการจับจองทรัพยากรใด ๆ

4. เมื่อเมธอด “prefetch” ถูกเรียก เพลย์เยอร์จะเข้าสู่สถานะ “Prefetching” ซึ่งในสถานะนี้ เพลย์เยอร์จะเตรียมตัวที่จะนำเสนอข้อมูลออกมา กระบวนการนี้เพลย์เยอร์จะทำการอ่านข้อมูลมาเตรียมไว้ และทำการจับจองทรัพยากรที่เป็นอีกคลัสซีฟายด์และสิ่งต่าง ๆ ที่ต้องการในการนำเสนอสถานะ “Prefetching” นี้สามารถเกิดขึ้นได้อีก หากมีการเปลี่ยนแปลงใด ๆ ในตัวเพลย์เยอร์ ที่จะทำให้ต้องมีการเตรียมข้อมูลใหม่ เช่น การเปลี่ยนตำแหน่งที่จะอ่านข้อมูล การเปลี่ยนอัตราในการประมวลผลข้อมูล ทำให้ต้องอ่านบัพเฟอร์มากขึ้น หรือการเปลี่ยนวิธีการประมวลผลข้อมูล เป็นต้น
5. เมื่อเพลย์เยอร์ผ่านสถานะ “Prefetching” แล้ว ก็จะมาอยู่ในสถานะ “Prefetched” ซึ่งในสถานะนี้เพลย์เยอร์ พร้อมแล้วที่จะ ไปสู่สถานะ “Started”
6. การเรียกเมธอด “start” จะทำให้เพลย์เยอร์ไปอยู่ในสถานะ “Started” ซึ่งในสถานะนี้ เวลาจริงของระบบเวลาของข้อมูลจะถูกดึงให้ตรงกันและนาฬิกาจะเริ่มทำงาน แม้ว่าตัวเพลย์เยอร์เองอาจต้องจะรอให้ถึงเวลาหนึ่งก่อนที่จะเริ่มนำเสนอข้อมูลก็ตาม

2.4.4 โปรเซสเซอร์ (Processor)

โปรเซสเซอร์เป็นอีกตัวหนึ่งที่มีไว้สำหรับการแสดงผลอินพุตของข้อมูลเหมือนกับเพลย์เยอร์ แต่โปรเซสเซอร์มีความสามารถมากกว่า กล่าวคือ โปรเซสเซอร์สามารถทำการเปลี่ยนแปลง, ปรับปรุง หรือแก้ไขอินพุตของข้อมูลก่อนการแสดงผลได้ ซึ่งชนิดของข้อมูลที่เพลย์เยอร์สนับสนุนจะสามารถใช้กับ โปรเซสเซอร์ได้



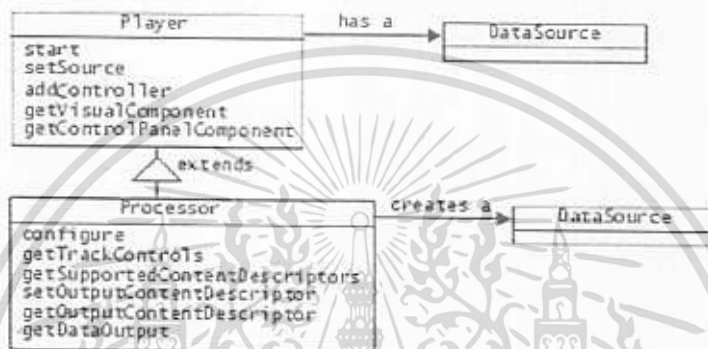
รูปที่ 2.18 แบบจำลองของ โปรเซสเซอร์ในเจเอ็มเอฟ

สำหรับการแสดงผลข้อมูลนั้น โปรเซสเซอร์สามารถทำได้หลายวิธี คือ สามารถทำการสร้างคาล์ซอร์สขึ้นมาใหม่จากนั้น จะนำคาล์ซอร์สตัวนี้ส่งไปยังเพลย์เยอร์หรือ โปรเซสเซอร์ตัวอื่น เพื่อแสดงผล หรือส่งไปยังเป้าหมายอื่น ๆ ในรูปของไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

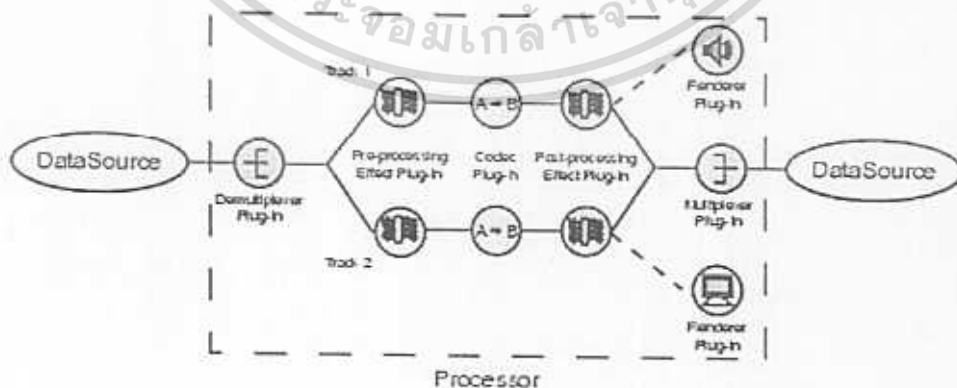
2.4.4.1 กระบวนการทำงานของ โปรเซสเซอร์

โปรเซสเซอร์คือเพลย์เบอร์ดึงเอาค้ำซอร์สมาปรับปรุงเปลี่ยนแปลงข้อมูล เพื่อสร้างเป็นค้ำซอร์สอีกตัวขึ้นมา และสามารถส่งข้อมูลไปให้อุปกรณ์แสดงผลอย่างอื่นได้ ซึ่งถ้าข้อมูลที่ส่งออกไปนั้นถูกส่งไปยังค้ำซอร์ส ค้ำซอร์สจะสามารถส่งข้อมูลที่อยู่ในตัวมันเอง ไปให้เพลย์เบอร์ดึง หรือโปรเซสเซอร์อื่น ๆ หรือไม่ก็ส่งไปให้ค้ำซิงค์ (DataSink) ได้อีกด้วย



รูปที่ 2.19 คลาสไดอะแกรมของโปรเซสเซอร์ในเจเอ็มเอฟ

ในขณะที่กระบวนการทำงานของเพลย์เบอร์ดึงนั้น ที่ถูกกำหนดขึ้น โดยนักพัฒนาโปรแกรม โปรเซสเซอร์ ได้มีการอนุญาตให้นักพัฒนาโปรแกรม สามารถกำหนดชนิดของกระบวนการทำงาน ที่เกี่ยวข้องกับข้อมูลได้เช่นกัน ตัวอย่างเช่นการใส่เอฟเฟกต์หรือการผสมมีเดีย (Mixing) การทำงานแบบเรียลไทม์ กระบวนการทำงานของ โปรเซสเซอร์ ถูกแบ่งออกเป็นขั้นตอนดังรูปที่ 2.20



รูปที่ 2.20 ภาพรวมของสถานะต่าง ๆ ของโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

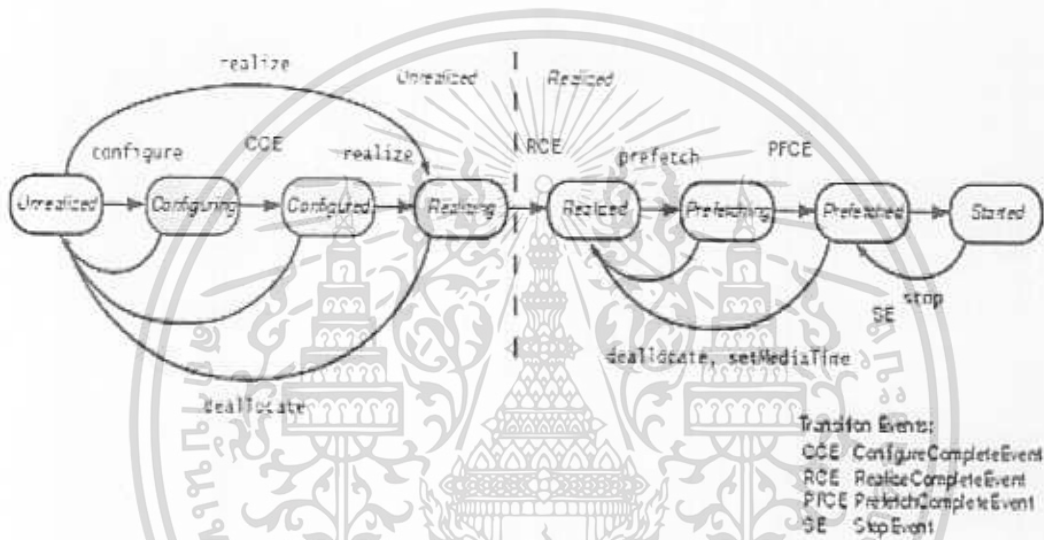
1. ดีมัลติเพล็กซ์ (Demultiplexing) คือกระบวนการที่กระจายข้อมูลออกเป็นแทร็ค (Tracks) ย่อย ๆ กล่าวคือ ถ้าข้อมูลที่เข้ามาประกอบไปด้วยหลายแทร็ครวมกัน เราสามารถทำการกระจายข้อมูลให้ออกมาเป็นแต่ละแทร็คได้ ยกตัวอย่างเช่น ไฟล์มัลติมีเดียแบบควิกไทม์ เราสามารถทำการดีมัลติเพล็กซ์ออกเป็นแทร็คออดิโอและแทร็ควิดีโอได้ ข้อมูลที่เข้ามายังโปรเซสเซอร์จะสามารถทำการดีมัลติเพล็กซ์ได้ก็ต่อเมื่อเป็นข้อมูลแบบมัลติเพล็กซ์เท่านั้น
2. พรีโพรเซสซิ่ง (Pre Processing) คือกระบวนการที่ตอบสนองต่อเอฟเฟกต์อัลกอริทึมของแต่ละแทร็คหลังจากการดีมัลติเพล็กซ์ข้อมูล
3. ทรานส์โคดีคิง (Transcoding) คือกระบวนการที่ทำการแปลงสัญญาณของแต่ละแทร็คจากรูปแบบหนึ่งให้ไปเป็นอีกรูปแบบหนึ่ง ถ้าทำการแปลงสัญญาณจากข้อมูลชนิดที่มีการบีบอัดไปเป็นข้อมูลชนิดที่ไม่มีการบีบอัดจะเรียกว่า “การถอดรหัส (decoding)” และในทางกลับกัน ถ้าทำการแปลงสัญญาณจากข้อมูลที่ไม่มีการบีบอัดไปเป็นข้อมูลชนิดที่มีการบีบอัดจะเรียกว่า “การเข้ารหัส (encoding)”
4. โพสต์โพรเซสซิ่ง (Post Processing) คือเป็นกระบวนการที่เกี่ยวกับการตอบสนองต่อเอฟเฟกต์อัลกอริทึมของแทร็คที่ถูกถอดรหัส
5. มัลติเพล็กซ์ (Multiplexing) คือกระบวนการรวมแทร็คแต่ละแทร็คที่จะออกจากโปรเซสเซอร์ให้เป็นข้อมูลเพียงชุดเดียว เช่น เราสามารถทำการรวมแทร็คของออดิโอและวิดีโอให้เป็นข้อมูลชนิดเอ็มเปกวันเทียงข้อมูลเดียวได้ ซึ่งเราสามารถกำหนดชนิดของข้อมูลที่จะออกจากโปรเซสเซอร์ได้จากเมธอด “setOutputContentDescriptor()”
6. เรนเดอร์ริง (Rendering) คือกระบวนการในการแสดงข้อมูลมายังอุปกรณ์แสดงผล ในกระบวนการทำงานของแต่ละชั้นคอน จะถูกจัดการ โดยคอมโพเนนท์ของเจเอ็มเอฟที่มีชื่อว่า เจเอ็มเอฟปลั๊กอิน (JMF plug-in) ซึ่งถ้าโปรเซสเซอร์ของเราสนับสนุน “TrackControls” เราสามารถที่จะเลือกปลั๊กอินที่เราต้องการใช้ในแต่ละแทร็คได้ ซึ่งมีปลั๊กอินให้เลือกทั้งหมดห้าชนิด ดังนี้
 - 1) ดีมัลติเพล็กซ์เซอร์ (Demultiplexer) เป็นการกระจายข้อมูลที่เป็นมีเดียสตรีมเช่นเอ็มเปกหรือควิกไทม์ที่ถูกมัลติเพล็กซ์มาออกเป็นหลายแทร็ค
 - 2) เอฟเฟกต์ (Effect) เป็นการเพิ่มลักษณะพิเศษลงไปในแทร็คของข้อมูล
 - 3) โคเดค (Codec) เป็นการทำการเข้ารหัสข้อมูลหรือการถอดรหัสข้อมูล
 - 4) มัลติเพล็กซ์เซอร์ (Multiplexer) เป็นการผสมแทร็คหลาย ๆ แทร็คให้เป็นข้อมูลเพียงชุดเดียว แล้วส่งข้อมูลนี้ไปยังค้ำซอร์ส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) เรนเดอร์เรอร์ (Renderer) เป็นการแสดงผลข้อมูลในแต่ละเฟรมไปยังอุปกรณ์แสดงภาพและเสียง

2.4.4.2 สถานะของ โปรเซสเซอร์

โปรเซสเซอร์จะมีสถานะเพิ่มขึ้นจากเพลย์เบ็คด้วยกันสองสถานะ คือ “Configuring” และ “Configured” โดยที่ทั้งสองสถานะนี้จะเกิดขึ้นก่อนที่โปรเซสเซอร์จะเข้าสู่สถานะ “Realizing”



รูปที่ 2.21 สถานะของ โปรเซสเซอร์ในเจเอ็มเอฟ

- โปรเซสเซอร์จะเข้าสู่สถานะ “Configuring” หลังจากมีการเรียกใช้เมธอด “configure” เมื่อโปรเซสเซอร์อยู่ในสถานะนี้ จะมีการติดต่อกับคาล์บเซิร์ฟ โดยการที่มีคาล์บเซิร์ฟข้อมูลที่เข้ามา และแจ้งให้คาล์บเซิร์ฟทราบถึงชนิดของข้อมูลนี้
- โปรเซสเซอร์จะเข้าสู่สถานะ “Configured” เมื่อกำลังติดต่อกับคาล์บเซิร์ฟและได้ทราบถึงชนิดของข้อมูลแล้ว เมื่อโปรเซสเซอร์อยู่ในสถานะ “Configured” จะได้คลาสที่ชื่อ “ConfigureCompleteEvent” ออกมา
- เมื่อมีการเรียกใช้งานเมธอด “realize” นั้น โปรเซสเซอร์จะเปลี่ยนแปลงไปเป็นสถานะ “Realized” ซึ่งเราจะถือว่าโปรเซสเซอร์ของเราได้มีการทำงานเสร็จสมบูรณ์ และในขณะที่โปรเซสเซอร์อยู่ในสถานะ “Configured” นั้น เราสามารถที่จะเรียกใช้งานเมธอด “getTrackControls” เพื่อเอาค่าในออบเจกต์ “TrackControl” สำหรับแต่ละเฟรมของข้อมูลมา เราสามารถระบุการทำงานของโปรเซสเซอร์ได้จากออบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

“TrackControl” นี้ ในการเรียกใช้เมธอด “realize” โดยตรง โปรเซสเซอร์จะเปลี่ยนจากสถานะจาก “Unrealized” ไปเป็นสถานะ “Realized” โดยอัตโนมัติ แต่เราจะไม่สามารถใช้ออบเจกต์ “TrackControl” เพื่อดูรายละเอียดของข้อมูลแต่ละแทร็คได้ และเราไม่สามารถเรียกใช้เมธอด “TrackControl” เมื่อ โปรเซสเซอร์อยู่ในสถานะ “Realized”

2.4.4.3 การควบคุมการประมวลผลเสียง (Processing controls)

เราสามารถที่จะเรียกใช้เมธอด “getTrackControls” นี้ โดยการดึงเอาข้อมูลในแต่ละแทร็คผ่านคลาส “TrackControl” เพื่อควบคุมการทำงานของโปรเซสเซอร์ที่กระทำต่อแทร็คแต่ละแทร็ค เราสามารถเรียกใช้คลาส “PlugInManager” เพื่อทำการตรวจสอบว่าเราได้มีการติดตั้งปลั๊กอินอะไรบ้าง การควบคุมการทำงานสไลด์ในแต่ละแทร็คโดยโกลบอลแต่ละตัวสามารถเรียกใช้ตัวช่วยจากเมธอด “getControls” ของ “TrackControl” ได้ ซึ่งเมธอดนี้จะให้ค่าโกลบอลคอนโทรลของแทร็ค เช่น “BitRateControl” และ “QualityControl” ในการกำหนดชนิดของเอาต์พุตเป็นไปตามที่เราต้องการ เราสามารถเรียกใช้เมธอด “setFormat” ในการระบุชนิดของฟอร์แมตที่เราต้องการ จากนั้น โปรเซสเซอร์จะเลือกโกลบอลและเรเนเจอร์ที่ที่เหมาะสมให้อีกวิธีหนึ่งที่เราสามารถทำได้คือ ทำการระบุชนิดของข้อมูลเมื่อมีการสร้างโปรเซสเซอร์ขึ้นมาโดยใช้คลาส “ProcessorModel” ซึ่งในคลาส “ProcessorModel” นี้จะทำการกำหนดอินพุตและเอาต์พุตที่ต้องการสำหรับโปรเซสเซอร์ที่เราสร้างขึ้น

2.4.4.4 ดาต้าเอาต์พุต (Data Output)

เมธอด “getDataOutput” จะทำการคืนค่าผลลัพธ์ที่ได้จากโปรเซสเซอร์ในรูปแบบของดาต้าซอร์ส ซึ่งดาต้าซอร์สนี้อาจอยู่ในรูปของ “PushDataSource”, “PushBufferDataSource”, “PullDataSource” และ “PullBufferDataSource” และยังสามารถเป็นข้อมูลที่ป้อนให้กับเพลย์เซอร์หรือโปรเซสเซอร์ตัวอื่น หรือเป็นข้อมูลที่ส่งไปยังดาต้าซิงก์ได้ ซึ่ง โปรเซสเซอร์ที่ไม่มีการส่งค่าผลลัพธ์ให้ดาต้าซอร์สจะถูกเปรียบเสมือนว่าเป็นเพลย์เซอร์

2.4.5 เมเนเจอร์ (Managers)

เอพีไอของเจเอ็มเอฟมีการทำงานของออบเจกต์ต่าง ๆ หลายส่วนประกอบกัน เช่น การจับเสียง การประมวลผล และการนำเสนอสื่อเชิงเวลา ซึ่งแต่ละส่วนนี้จะมีความสัมพันธ์ซึ่งกันและกัน เราจำเป็นต้องมีออบเจกต์ที่ใช้เป็นตัวกลาง (Intermediary Object) ในการทำให้ออบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่าง ๆ สามารถทำงานร่วมกันได้ ซึ่งเจเอ็มเอฟก็มีออบเจกต์ตัวกลางนี้เอาไว้สำหรับให้เรียกใช้อยู่ แล้ว มีชื่อว่า “Manager” โดยเจเอ็มเอฟจะมีเมเนเจอร์ที่เกี่ยวข้องทั้งหมดสี่ชนิดคือ

1. เมเนเจอร์ (Manager) ใช้จัดการเกี่ยวกับการสร้างออบเจกต์ของคลาส “Players”, “Processors”, “DataSources” และ “DataSinks”
2. แพคเกจเมเนเจอร์ (PackageManager) ใช้ในการเก็บรายการ (Registry) ของแพคเกจทั้งหมดที่เกี่ยวข้องกับ เจเอ็มเอฟ เช่นคลาส คลาส “Players”, “Processors”, “DataSources” และ “DataSinks” ที่เราสามารถเขียนขึ้นมาเองได้
3. แคปเจอร์ดีไวซ์เมเนเจอร์ (CaptureDeviceManager) ใช้ในการจัดการอุปกรณ์จับข้อมูล มีเดียต่าง ๆ
4. ปลั๊กอินเมเนเจอร์ (PlugInManager) จะใช้ในการเก็บรายการของปลั๊กอินต่าง ๆ ที่มีในเจเอ็มเอฟ ซึ่งในเจเอ็มเอฟมีปลั๊กอินด้วยกันทั้งสิ้นห้าชนิด คือ “Multiplexers”, “Demultiplexers”, “Codecs”, “Effects” และ “Renderers”

ในการพัฒนาโปรแกรมโดยใช้เจเอ็มเอฟ เราจำเป็นที่จะต้องใช้เมธอดของคลาส “Manager” เพื่อทำการสร้างเพลย์เบอร์ โปรเซสเซอร์ ค่ายอร์สและค้ายิ่งก์

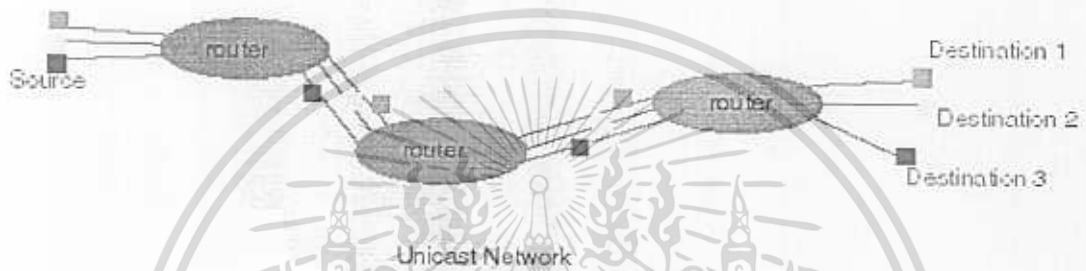
2.5 เครือข่ายมัลติคาสต์ (Multicast Network)

จำนวนของโปรแกรมใช้งานทั่วไปทั้งในอดีตและปัจจุบัน เช่น เว็บไซต์ (Website) อีเมลล์ (E-mail) จะเป็นการรับส่งข้อมูลระหว่างหนึ่งผู้ส่งและหนึ่งผู้รับ นอกจากนั้นยังมีโปรแกรมหลายสื่อ เช่น การประชุมระยะไกล การส่งภาพและเสียงผ่านเครือข่าย โปรแกรมอย่างหลังนี้จะเป็นการส่งที่เป็นกลุ่มจากแหล่งข้อมูลหนึ่งแหล่งไปยังผู้รับหลาย ๆ จุด ทำให้เกิดความต้องการใช้ทรัพยากรของเครือข่ายมากขึ้น

การใช้งานของเครือข่ายในปัจจุบัน ส่วนใหญ่จะเป็นแบบยูนิคาสต์ (Unicast) คือการส่งแบบหนึ่งผู้ส่งต่อหนึ่งผู้รับ ในกรณีที่เรากำลังส่งจากหนึ่งผู้ส่งไปยังหลาย ๆ ผู้รับ ถ้าเราใช้วิธีการแบบยูนิคาสต์ ก็จะต้องทำการส่งทีละครั้งไปเรื่อย ๆ ไปจนครบ ถ้าเราต้องการส่งครั้งเดียวให้ถึงผู้รับทั้งหมด ก็สามารถทำได้โดยวิธีที่เรียกว่าบรอดคาสต์ (Broadcast) ซึ่งทั้งสองแบบจะทำให้เกิดปัญหากับเครือข่ายในเรื่องของความเร็วของเครือข่ายไม่พอเพียง เนื่องจากมีการส่งข้อมูลซ้ำ ๆ กันจำนวนมากในเครือข่าย ทำให้เกิดปัญหาคอขวด และสำหรับกรณีหลัง ถ้าทางด้านผู้รับไม่ได้ต้องการข้อมูลดังกล่าวได้รับข้อมูลนั้น ๆ ไปด้วย ทำให้เสียเวลาที่จะต้องประมวลผลข้อมูลจากบรอดคาสต์อีกด้วย ปัจจุบันแนวโน้มความต้องการใช้งานหลายสื่อที่ประกอบด้วย ภาพ เสียง ข้อมูล ได้เพิ่มขึ้นอย่างรวดเร็วนั้น หมายถึงการที่จะต้องเพิ่มความเร็ว ความจุของเครือข่าย เพื่อที่จะรองรับกับการใช้

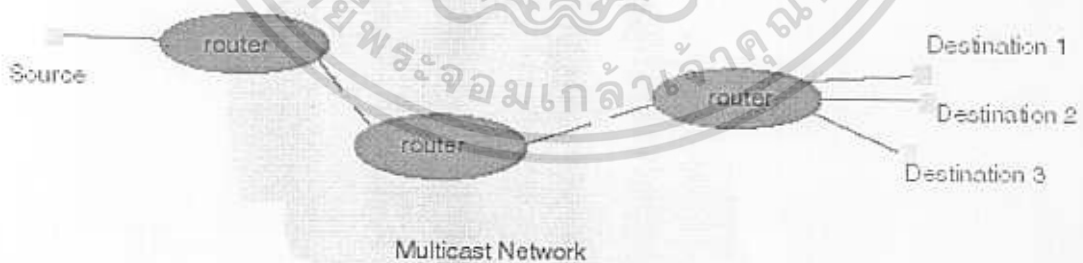
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานดังกล่าว และสามารถใช้ทรัพยากรที่มีอยู่ให้เกิดประสิทธิภาพ นอกเหนือจากการขยายในด้านของความเร็ว เราจำเป็นต้องมีวิธีการเพื่อที่จะสามารถใช้เครือข่ายให้เกิดประสิทธิภาพ การแก้ไขปัญหาดังกล่าว สามารถทำได้โดยการส่งข้อมูลแบบมัลติคาสต์ (Multicast) เนื่องจากการส่งข้อมูลแบบมัลติคาสต์นั้น สามารถที่จะส่งข้อมูลชุดเดียวกันจากหนึ่งผู้ส่งไปยังหลาย ๆ ผู้รับ (ที่ต้องการรับข้อมูล) โดยไม่จำเป็นต้องส่งข้อมูลดังกล่าวหลาย ๆ ครั้ง ทำให้สามารถใช้ประโยชน์จากแบนด์วิดท์ (Bandwidth) ของเครือข่ายได้อย่างมีประสิทธิภาพ



รูปที่ 2.22 การส่งข้อมูลแบบยูนิคาสต์

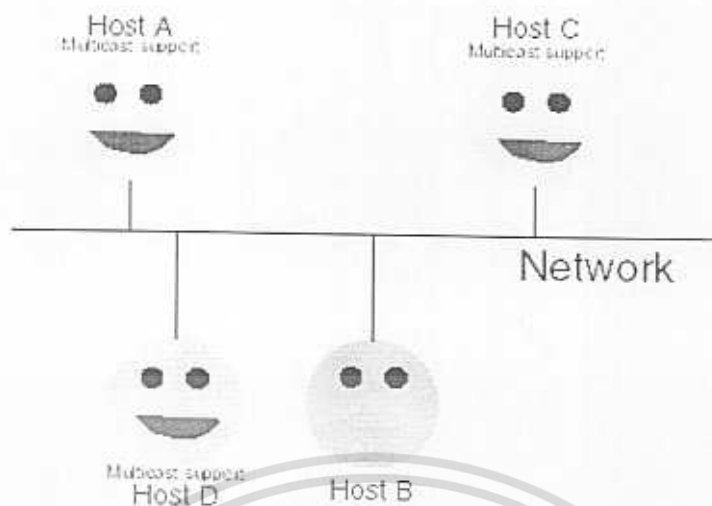
จากรูปที่ 2.22 เป็นการส่งแบบหนึ่งผู้ส่งไปยังหนึ่งผู้รับ ข้อมูลชุดเดียวกันถูกส่งแบบจุดต่อจุดไปยังผู้รับ 3 กลุ่ม ซึ่งต้องส่งข้อมูลถึง 3 ครั้ง



รูปที่ 2.23 การส่งข้อมูลแบบมัลติคาสต์

จากรูปที่ 2.23 เป็นการส่งแบบหนึ่งผู้ส่งไปยังหลาย ๆ ผู้รับ ข้อมูลชุดเดียวกันสามารถที่จะส่งถึงกลุ่มผู้รับเช่นเดียวกับแบบยูนิคาสต์ ซึ่งการส่งเพียงครั้งเดียว จะประหยัดการใช้ทรัพยากรของเครือข่ายมากกว่าแบบยูนิคาสต์อย่างมาก ในกรณีที่กลุ่มของผู้รับมีจำนวนเพิ่มมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 ตัวอย่างการส่งข้อมูลในแบบต่าง ๆ

จากรูปที่ 2.24 กรณีที่เป็นแบบยูนิคาสต์ เอ หรือ บี หรือ ซี จะได้รับข้อมูล

กรณีที่เป็นแบบบรอดคาสต์ เอ และ บี และ ซี จะได้รับข้อมูล

กรณีที่เป็นแบบมัลติคาสต์ เอ และ ซี จะได้รับข้อมูล

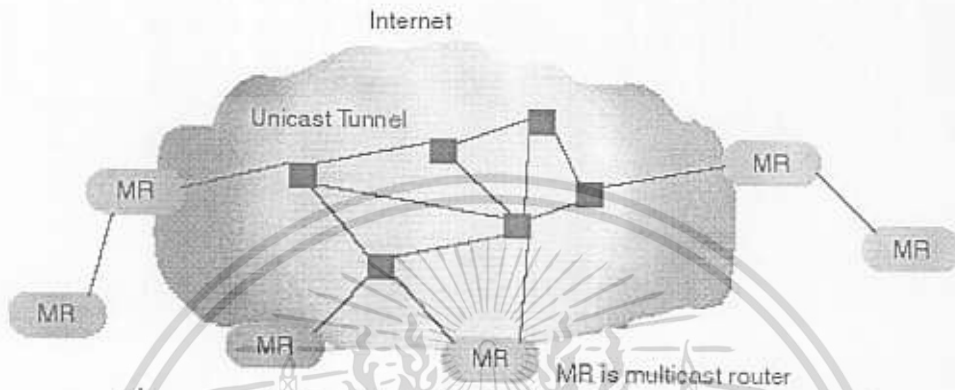
มัลติคาสต์เป็นเทคโนโลยีแบบที่ทางฝั่งผู้ส่งไม่จำเป็นต้องทราบ หรือจัดการทางด้านรับ (Receiver-Based Concept) เหมือนกับการส่งข้อมูลผ่านวิทยุโทรทัศน์นั่นเอง เมื่อผู้รับต้องการรับข้อมูลดังกล่าว ก็ปรับสถานีหรือช่องรับให้ตรงกับสถานีส่งที่ตนเองต้องการ

2.5.1 เอ็มโบน (MBone หรือ Multicast Backbone)

ปัจจุบันได้มีการนำวิธีการส่งข้อมูลแบบมัลติคาสต์ มาใช้ในเครือข่ายอินเทอร์เน็ต ซึ่งเป็นที่ทราบกันดีในชื่อของเอ็มโบน เกิดจากการทดลองที่จะส่งภาพและเสียงการประชุมขององค์กร ไออีทีเอฟ (Internet Engineering Task Force (IETF)) โดยเป็นการร่วมมือของอาสาสมัครในการทดลองและใช้งานไอพีมัลติคาสต์ (IP Multicast) และเป็นการสร้างเครือข่ายของไอพีมัลติคาสต์ เพื่อการพัฒนาการใช้งานของโปรแกรมประยุกต์ต่าง ๆ โดยไม่ต้องรอการแพร่หลายของเราท์เตอร์ที่สนับสนุนการทำมัลติคาสต์บนเครือข่ายอินเทอร์เน็ต ดังนั้นเอ็มโบนจึงเป็นเครือข่ายเสมือนที่ทำงานบนเครือข่ายอินเทอร์เน็ต โดยมีหน้าที่สนับสนุนการหาเส้นทางของไอพีมัลติคาสต์ เอ็มโบนนี้สามารถใช้ได้กับเครือข่ายทั้งที่สนับสนุนการทำมัลติคาสต์และไม่สนับสนุนการทำมัลติคาสต์ โดยเมื่อข้อมูลมัลติคาสต์ผ่านเครือข่ายแบบเดิม (ที่ไม่สนับสนุนการทำมัลติคาสต์) ข้อมูลของมัลติคาสต์ก็จะถูกส่งไปเป็นแบบยูนิคาสต์ โดยมีข้อมูลมัลติคาสต์อยู่ภายใน (Encapsulation) วิธีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังกล่าวจะเรียกว่าการทำอุโมงค์ (Tunneling) เพื่อให้ส่งข้อมูลมัลติคาสต์ผ่านยังเครือข่ายยูนิคาสต์ได้ ดังรูปที่ 2.25 โดยจุดที่เป็นมัลติคาสต์เราท์เตอร์ (Multicast Router) หรือจุดปลายของอุโมงค์ ก็จะเป็นอุปกรณ์ที่สนับสนุนวิธีการมัลติคาสต์ หรืออาจเป็นเครื่องคอมพิวเตอร์ที่ทำงานด้วยโปรแกรมที่ชื่อเอ็มเรท์ (Mroute) ซึ่งทำหน้าที่เป็นมัลติคาสต์เราท์ติ้ง (Multicast Routing)



รูปที่ 2.25 การทำอุโมงค์ผ่านเครือข่ายที่ไม่สนับสนุนการทำมัลติคาสต์

2.5.2 กลุ่มของมัลติคาสต์ (Multicast Groups)

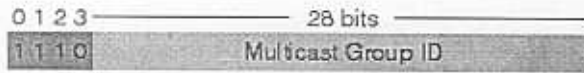
ในรูปแบบของไอพีวีโฟร์แอดเดรส (IPv4 addresses) จะถูกแบ่งออกเป็นสามชนิดด้วยกันคือ ยูนิคาสต์ บรอดคาสต์ และมัลติคาสต์ ซึ่งยูนิคาสต์แอดเดรสจะใช้สำหรับส่งข้อมูลแบบจุดต่อจุด ส่วนบรอดคาสต์แอดเดรสจะใช้สำหรับส่งข้อมูลไปยังทุก ๆ จุดที่อยู่ในเครือข่ายนั้น โดยที่ไอพีแอดเดรสคลาส เอ บี และ ซี (Class A, B and C IP address) สำหรับยูนิคาสต์ และคลาส ดี สำหรับมัลติคาสต์

2.5.2.1 มัลติคาสต์แอดเดรส (Multicast Address)

ไอพีแอดเดรสคลาส ดี (Class D IP address) นี้ จะใช้แทนสมาชิกของกลุ่มของมัลติคาสต์ ไอพีแอดเดรสคลาส ดี มีลักษณะเหมือนไอพีแอดเดรสทั่วไป คือ มีขนาด 32 บิต โดย 4 บิตแรกจะเป็น 1110 ส่วนอีก 28 บิตที่เหลือเราจะเรียกว่า มัลติคาสต์กรุปไอดี (Multicast Group ID) ซึ่งใช้แทนกลุ่มของมัลติคาสต์ ลักษณะการเขียนหมายเลขไอพีในแบบของคอตเดซิมีอล (Dot Decimal) จะเป็นลักษณะเดียวกันกับการเขียนหมายเลขไอพีทั่ว ๆ ไป ซึ่งจะได้หมายเลขตั้งแต่ 224.0.0.0 ถึง 239.255.255.255 โดยจะมีบางช่วงของแอดเดรสที่ถูกกันไว้ใช้งานบางอย่าง (ไม่สามารถนำไปใช้เป็นอย่างอื่นได้) เช่น 244.0.0.1 หมายถึง โฮสต์ทั้งหมดของกลุ่ม 224.0.0.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายถึง เราท์เตอร์ทั้งหมด เป็นต้น รายละเอียดของไอพีที่กันไว้ใช้งานนั้นสามารถหารายละเอียดดูได้จากอาร์เอฟซี1700 (RFC1700)



Multicast host groups 224.0.0.0 to 239.255.255.255

Permanent addresses Temporary addresses

224.0.0.1 "all-hosts-group"
 224.0.0.2 "all routers on a LAN"
 224.0.0.0 - 224.0.0.255
 is reserved for routing protocol and other low-level
 topology discovery or maintenance protocols.
 224.0.13.0 - 224.0.13.255 for Net News.

รูปที่ 2.26 รูปแบบของไอพีแอดเดรสคลาสต์

ทั้งนี้ มีลิตาสต์ค่าด้านกรม (Multicast Datagram) หรือแพ็กเกต (Packet) สามารถที่จะส่งไปยังกลุ่มผู้รับได้ด้วยวิธีเดียวกันกับยูนิคาสต์ไอพีปกติ (อาจเกิดการสูญหายและการผิดพลาดได้เช่นเดียวกันกับยูนิคาสต์ไอพีแพ็กเกต)

2.5.2.2 ไอจีเอ็มพี (Internet Group Management Protocol (IGMP))

ในการรับข้อมูลของมัลติคาสต์นั้น ผู้รับในกลุ่มมัลติคาสต์ จะต้องมีการแจ้งให้กับเราท์เตอร์ว่าต้องการที่จะรับข้อมูล ซึ่งในแต่ละจุดสามารถที่จะมีสมาชิกได้มากกว่าหนึ่งกลุ่มขึ้นไป โดยเราท์เตอร์จะทำการรับและส่งข้อมูลให้กับสมาชิกในกลุ่ม กลไกที่ใช้ในการทำงานงานดังกล่าว เราจะเรียกว่า "ไอจีเอ็มพี" วิธีการนี้ยังใช้ในการตรวจสอบสมาชิกในกลุ่มว่ายังมีอยู่หรือไม่อีกด้วย ในกรณีที่มียานวนเราท์เตอร์หลาย ๆ ตัวในเครือข่ายแลน (LAN) จะมีอยู่หนึ่งตัวที่จะถูกเลือกขึ้นมาเป็นตัวแทน เพื่อทำการสอบถาม (Query) และเก็บสถานะของสมาชิกในกลุ่มของมัลติคาสต์ จากข้อมูลที่ได้มา เราท์เตอร์จะใช้ข้อมูลเหล่านี้ในการตัดสินใจว่าจะส่งข้อมูลให้กับผู้รับในเครือข่ายหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 มัลติคาสต์เร้าท์ติงอัลกอริทึม (Multicast Routing Algorithms)

ปัจจุบันมีหลายวิธี ในการสร้างเส้นทางสำหรับส่งข้อมูล ไปยังผู้รับ หรือเรียกกันว่า มัลติคาสต์ทรี (Multicast Tree) โดยวิธีการเหล่านี้จะถูกนำไปใช้ในมัลติคาสต์เร้าท์ติงโพรโตคอล (Multicast Routing Protocol) เราจะมาดูสองวิธีการง่าย ๆ ก่อนที่เรียกกันว่า ฟลัดดิ้ง (Flooding) และ สเปนนิ่งทรี (Spanning Trees) จากนั้นก็จะกล่าวถึงวิธีที่ได้รับการปรับปรุงแก้ไขข้อบกพร่องของสองวิธีการแรกอันได้แก่ อาร์พีบี (Reverse Path Broadcasting (RPB)) ทีอาร์พีเอฟ (Truncated Reverse Path Forwarding (TRPF)) อาร์พีเอ็ม (Reverse Path Multicasting (RPM)) เอสที (Steiner Trees (ST)) ซีบีที (Core-Based Trees (CBT)) ส่วนในหัวข้อถัดไปจะเป็นการแสดงให้เห็นว่า เรา นำเอาวิธีการต่าง ๆ เหล่านี้ไปใช้ประโยชน์อย่างไร

2.5.3.1 ฟลัดดิ้ง (Flooding)

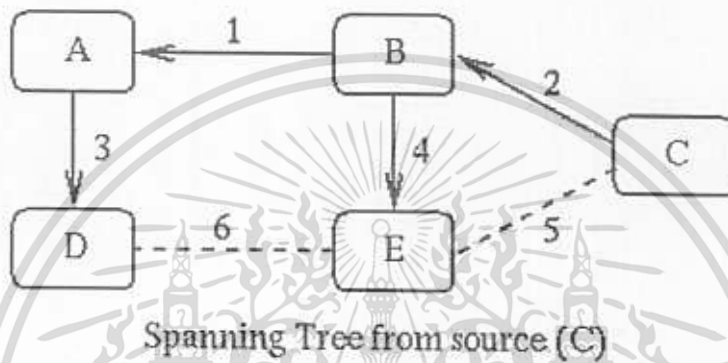
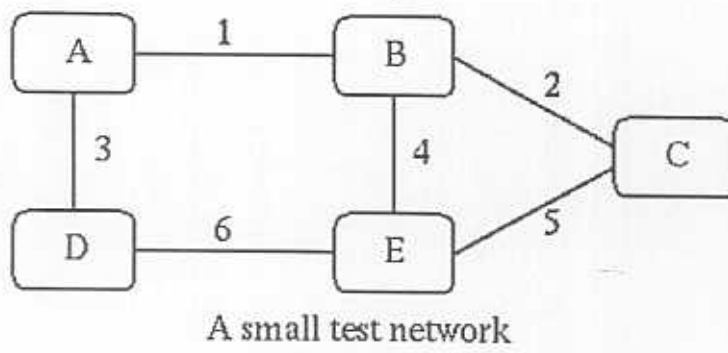
ฟลัดดิ้งอัลกอริทึม (Flooding Algorithm) นี้ เป็นวิธีการที่ใช้อยู่ในหลาย ๆ โพรโตคอล วิธีการนี้ใช้เทคนิคง่าย ๆ คือ เมื่อเราที่เคอร์ได้รับข้อมูลมัลติคาสต์ ก็จะทำการตรวจสอบข้อมูลนั้นว่าเป็นข้อมูลที่รับมาครั้งแรกหรือไม่ ถ้าใช่เราที่เคอร์ก็จะทำการส่งข้อมูลนั้นออกไปยังเราที่เคอร์ตัวอื่น ๆ ทุกตัวยกเว้นตัวที่มันได้รับข้อมูลนั้นมา ถ้าเป็นกรณีอื่นเราที่เคอร์ก็จะไม่สนใจข้อมูลดังกล่าว ในกรณีนี้เราแน่ใจได้เลยว่าเราที่เคอร์ทุกตัวในเครือข่ายจะได้รับข้อมูลอย่างน้อยหนึ่งครั้ง

แม้ว่าวิธีการนี้จะง่ายแต่มีข้อเสียคือ เปลืองเครือข่ายการสื่อสารมาก เนื่องจากการส่งข้อมูลออกจากเราที่เคอร์แต่ละตัว ไปยังเราที่เคอร์ตัวอื่น ๆ ที่ตัวมันเองเชื่อมต่อกับอยู่ ยิ่งไปกว่านั้น เราที่เคอร์ยังต้องเก็บสถานะของข้อมูลไว้เพื่อตรวจสอบดูว่าเป็นข้อมูลที่เคยได้รับมาแล้ว หรือได้รับมาเป็นครั้งแรก

2.5.3.2 สเปนนิ่งทรี (Spanning Trees)

วิธีการที่ดีกว่าฟลัดดิ้งก็คือสเปนนิ่งทรี วิธีนี้ในเครือข่ายจะถูกกำหนดให้เป็นลักษณะของโครงสร้างต้นไม้ กล่าวคือ เป็นโครงสร้างที่ไม่มีการวนรอบ (Loop-Less Graph) จะมีเส้นทางเดียวในการติดต่อระหว่างสองเราที่เคอร์ เมื่อใดก็ตามที่เราที่เคอร์ได้รับมัลติคาสต์แพ็กเกตมันจะส่งต่อไปยังทุกข่ายที่เชื่อมต่อกันในเส้นทางของผั่งต้นไม้อีก ยกเว้นเส้นทางที่ได้รับแพ็กเกตมา เมื่อพิจารณาจากรูปที่ 2.27 ประกอบ จะเห็นว่าเราที่เคอร์ทุกตัวในเครือข่ายได้รับข้อมูลครบทุกตัว โดยเราที่เคอร์จะเก็บข้อมูลที่แสดงให้รู้ว่าเส้นทางนั้นเป็นสเปนนิ่งทรีหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



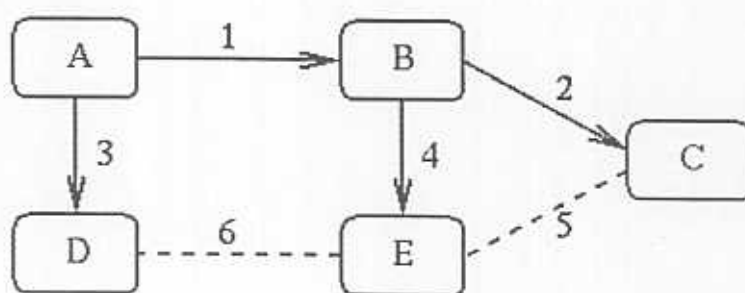
รูปที่ 2.27 สเปนนิ่งทรี

วิธีการแบบสเปนนิ่งทรีมีข้อเสียอยู่สองประการคือ ประการแรกมันจะพิจารณากราฟฟิคทั้งหมดถึงแม้ว่าจะเป็นจุดเล็ก ๆ ในการเชื่อมโยง อีกประการหนึ่งคือ มันไม่ได้ทำนายว่าจุดนั้นจะมีสมาชิกของมัลติคาสต์อยู่หรือไม่

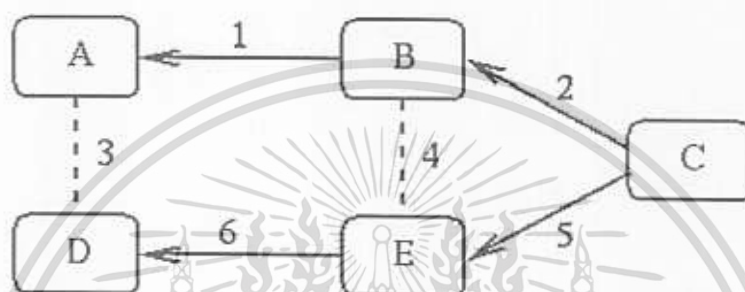
2.5.3.3 อาร์พีบี (Reverse Path Broadcasting (RPB))

วิธีการแบบอาร์พีบีที่ถูกนำมาใช้ในเอ็ม โบนนั้น เป็นการปรับปรุงวิธีการของสเปนนิ่งทรี โดยตอนเริ่มแรกจะใช้วิธีการของสเปนนิ่งทรีในการสร้างเส้นทาง เมื่อเราที่เตอร์ได้รับข้อมูลบนเส้นทางแอลจากต้นทางเอส เราที่เตอร์ก็จะทำการตรวจสอบดูว่าเส้นทางแอลเป็นเส้นทางที่สั้นที่สุดไปยังต้นทางเอสหรือไม่ ถ้าใช่เราที่เตอร์ก็จะส่งข้อมูล ไปยังทุกเส้นทางยกเว้นเส้นทางแอล ถ้าเป็นกรณีอื่น ๆ เราที่เตอร์ก็จะทิ้งข้อมูลนั้นไป ดูรูปที่ 2.28 ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



RPB tree from source (A)



RPB tree from source (C)

รูปที่ 2.28 ออร์พีบีทีรี

ตัวอย่างจาก ออร์พีบีทีรีจากแหล่งข้อมูล (RPB Tree From Source (C)) เราท์เตอร์ ดี จะไม่รับข้อมูลจาก ซี ผ่านทาง บี-เอ เพราะไม่ใช่เส้นทางที่ดีที่สุด เมื่อเทียบจาก ซี-อี ในกรณีนี้ก็คือ ไม่มีความจำเป็นที่จะต้องส่งข้อมูลผ่านมาทาง เอ-ดี หรือ บี-อี

วิธีการออร์พีบีทีรีนี้ง่ายและเร็ว เนื่องจากการส่งข้อมูลจากแหล่งกำเนิดไปยังปลายทาง จะใช้เส้นทางที่ดีที่สุดในการส่ง แต่ก็ยังมีข้อบกพร่องบางประการคือ วิธีนี้จะไม่ทราบข้อมูลเกี่ยวกับสมาชิกของกลุ่มมัลติคาสต์ เพื่อใช้สำหรับสร้างลิสทรีวิวชันทีรี (Distribution Tree) เลย

2.5.3.4 ทีอาร์พีบี (Truncated Reverse Path Broadcasting (TRPB))

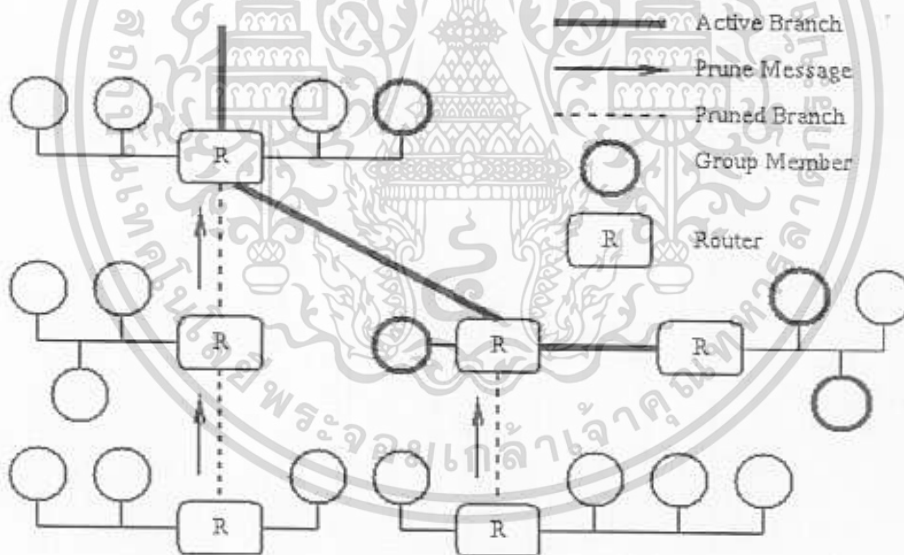
ทีอาร์พีบีเป็นวิธีการที่น่าเสนอขึ้นมา เพื่อแก้ไขข้อบกพร่องของออร์พีบี โดยจะใช้โปรโตคอล ไอจีเอ็มพีเข้ามาช่วย ทำให้เราท์เตอร์สามารถกำหนดได้ว่า มีสมาชิกมัลติคาสต์อยู่ในเครือข่ายของเราท์เตอร์ตัวนั้นหรือไม่ ในกรณีที่ไม่มีสมาชิกอยู่ในเครือข่ายเลย และถ้าเครือข่ายนั้นอยู่ปลายสุด (ไม่มีการติดต่อกับเราท์เตอร์ตัวอื่น ๆ) เราท์เตอร์ก็จะทำการตัดส่วนนี้ออกจากสเปนนิงทีรี นอกนั้นก็เหมือนกับวิธีการของออร์พีบี ถึงแม้ว่าเรานำเอาเงื่อนไขนี้มาใช้ แต่ถ้าเราท์เตอร์นั้นไม่ได้อยู่ปลายสุด แม้จะไม่มีสมาชิกอยู่ก็ไม่สามารถตัดออกไปจากระบบได้ เนื่องจากอาจจะยังมีการเชื่อมต่อกับตัวอื่น ๆ ที่ยังมีสมาชิกของมัลติคาสต์อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.5 อาร์พีเอ็ม (Reverse Path Multicasting (RPM))

อาร์พีเอ็ม (หรือที่รู้จักกันว่าเป็นอาร์พีบีที่ประกอบด้วยวิธีการตัดออกจากกลุ่มมัลติคาสต์) เป็นการเพิ่มประสิทธิภาพจากวิธีอาร์พีบี และทีอาร์พีบี โดยอาร์พีเอ็มจะสร้างเส้นทางการส่งข้อมูลเฉพาะสองกรณีคือ ส่งข้อมูลเฉพาะในเส้นทางที่มีสมาชิกของมัลติคาสต์อยู่ และส่งข้อมูลเฉพาะเส้นทางที่สั้นที่สุดเท่านั้น

ตัวอย่างเช่น กำหนดให้คู่ของแหล่งข้อมูลหรือกลุ่มสมาชิกแพ็คเกจแรกๆของมัลติคาสต์ถูกส่งไปด้วยวิธีแบบทีอาร์พีบี ถ้าเราท์เตอร์ที่อยู่ปลายทางสุดของทรีได้รับข้อมูลมัลติคาสต์ และไม่มีสมาชิกใดติดต่อกับเราท์เตอร์ดังกล่าว มันก็จะส่งข้อมูลที่เรียกว่า พรุน (Prune) ไปยังเราท์เตอร์ที่เป็นตัวส่งข้อมูลมัลติคาสต์นั้นมาให้ ซึ่งข้อมูลพรุนดังกล่าวทำให้เราท์เตอร์ทราบว่า ไม่ต้องทำการส่งข้อมูลมัลติคาสต์นั้นไปยังเส้นทางที่ได้รับข้อมูลพรุนแล้ว และที่สำคัญก็คือ พรุนจะถูกส่งภายในขอบเขตไม่เกิน 1 จุดในเครือข่าย (1 Hop) จะทำให้การส่งข้อมูลมัลติคาสต์ส่งในเฉพาะเส้นทางที่มีสมาชิกอยู่เท่านั้น รูปที่ 2.29 ประกอบ



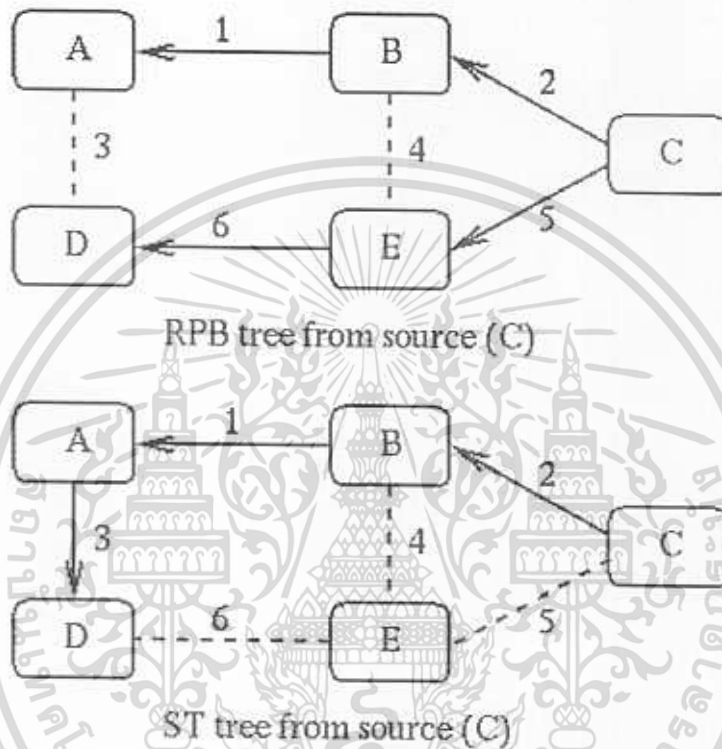
รูปที่ 2.29 อาร์พีเอ็มทรี

เนื่องจากกลุ่มของสมาชิกและโครงสร้างของเครือข่าย สามารถเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นจึงต้องมีการตรวจสอบสถานะอยู่เป็นระยะ ๆ เพื่อที่จะแจ้งให้เส้นทางทราบ ซึ่งทำให้จุดอ่อนของวิธีการนี้คือ จะต้องใช้หน่วยความจำจำนวนมาก เพื่อใช้ในการเก็บสถานะสำหรับในแต่ละคู่ของแหล่งข้อมูลและกลุ่มสมาชิก อีกอย่างหนึ่งก็คือ วิธีการดังกล่าวไม่เหมาะสมกับเครือข่ายขนาดใหญ่ ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.6 เอสที (Steiner Tree (ST))

วิธีการแบบอาร์พีบี ทีอาร์พีบี และอาร์ทีเอ็มเอ็นั้น เป็นวิธีการแบบเลือก ระยะทางที่สั้นที่สุด (Shortest Path) ระหว่างแหล่งกำเนิดและจุดรับข้อมูลปลายทาง ซึ่งรับประกันได้ว่าข้อมูลจะถูกส่งไปยังปลายทางได้อย่างรวดเร็วที่สุด แต่ถึงกระนั้นวิธีการดังกล่าวก็ยังไม่สามารถที่จะใช้ทรัพยากรของเครือข่ายได้อย่างมีประสิทธิภาพ



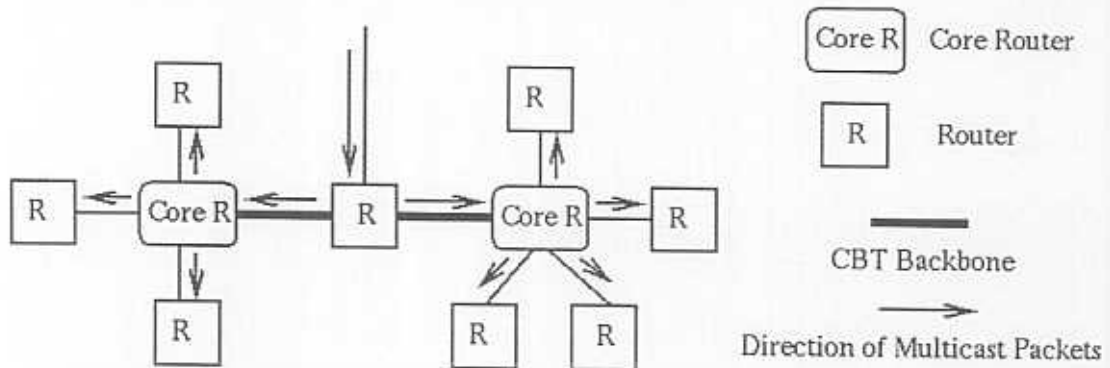
รูปที่ 2.30 เอสทีทรี

จากรูปที่ 2.30 จะเห็นได้ว่าวิธีเอสทีที่เริ่มจากแหล่งข้อมูล C จะดีกว่าในรูปแรก ถึงแม้ว่าวิธีการดังกล่าวจะทำให้เกิดเส้นทางที่ดีที่สุดก็ตาม แต่มันก็ยากในการที่จะคำนวณโครงสร้างให้เป็นโครงสร้างที่เล็ก ๆ ไม่ซับซ้อนได้ ดังนั้นรูปแบบของเอสทีทรีซึ่งมีการเข้าและออกของสมาชิกผู้ตลอดเวลา จึงทำให้วิธีนี้ไม่เสถียรเป็นอย่างมาก

2.5.3.7 ซีบีที (Core-Based Tree (CBT))

วิธีการสุดท้าย ณ ที่นี้ ก็คือ การสร้างโครงสร้างแบบที่เรียกว่า ซีบีที โดยที่ ซีบีทีจะทำการสร้างเส้นทางเดียวสำหรับส่งข้อมูลให้กลุ่มสมาชิก นั่นก็คือ มีการส่งข้อมูลไปยังจุด ๆ เดียวเท่านั้น ซึ่งจะเรียกจุด ๆ นั้นว่าคอร์ (Core) โดยกลุ่มของเราเตอร์จะทำการเลือกคอร์เราเตอร์ (Core Router) รูปที่ 2.31 ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 ซีบีที

เนื่องจากซีบีทีมีเส้นทางเดียวในการส่งข้อมูลให้กับกลุ่มสมาชิก จึงทำให้เราเตอร์ไม่ต้องเก็บข้อมูลจำนวนมาก เมื่อเทียบกับวิธีอื่น ๆ และไม่จำเป็นต้องทำวิธีการพลัดตั้ง ใดๆก็ตาม การที่มีเส้นทางเดียวก็อาจเกิดลักษณะที่เป็นคอขวดของเส้นทางในการส่งข้อมูลได้ วิธีการตั้งที่กล่าวมานี้ในแต่ละแบบก็จะมีข้อดีและข้อเสียแตกต่างกันไปในการที่จะนำมาพัฒนา เพื่อใช้สำหรับการส่งข้อมูล

2.5.4 มัลติคาสต์เราต์ติ้งโปรโตคอล (Multicast Routing Protocol)

ในหัวข้อที่ผ่านมา เราได้อธิบายถึงวิธีการที่สามารถจะนำมาใช้ได้ ในมัลติคาสต์เราต์ติ้งโปรโตคอล ต่อไปเราจะดูว่าปัจจุบันนี้มีมัลติคาสต์เราต์ติ้งโปรโตคอลแบบไหนบ้าง และในแต่ละวิธี ได้นำเอาวิธีการดังกล่าวมาใช้อย่างไร

2.5.4.1 ดีวีเอ็มอาร์พี (Distance Vector Multicast Routing Protocol (DVMRP))

ดีวีเอ็มอาร์พีถูกกำหนดขึ้นครั้งแรกโดยไออาร์เอฟซี1075 (RFC1075) โดยใช้หลักการพื้นฐานของวิธีที่เรียกว่า อาร์ไอพี (Routing Information Protocol (RIP)) แต่จะต่างกับที่อาร์ไอพีนั้นเป็นการส่งแพ็คเกจยูนิคาสต์ไปยังจุดถัดไปที่จะไปยังปลายทาง ในขณะที่ดีวีเอ็มอาร์พีเป็นการสร้างแผนผังต้นไม้บนข้อมูลของจุดก่อนหน้าที่จะไปยังคันทาง ดอนแรกดีวีเอ็มอาร์พีนั้นได้ใช้วิธีการแบบที่อาร์พีบี ต่อมาพัฒนามาใช้วิธีแบบอาร์พีเอ็ม ส่วนมาตรฐานรุ่นล่าสุดของดีวีเอ็มอาร์พีกำหนดโดยกลุ่มการทำงานไอดีเอ็มอาร์ (Inter-Domain Multicast Routing (IDMR)) ในองค์กรไออีทีเอฟ

ดีวีเอ็มอาร์พีได้นำเอาวิธีการของอาร์พีเอ็มมาใช้ โดยข้อมูลชุดแรกจะถูกส่งออกจากไปในลักษณะของการกระจายไปทั่วเครือข่าย จากนั้นพ-run message (Prune Message) จะถูกส่งออกไป เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัดในจุดที่ไม่มีกลุ่มสมาชิกปรากฏอยู่ และมีการกำหนดข้อมูลใหม่ขึ้นมาที่เรียกว่า กราฟท์ (Graft) สำหรับใช้ในกรณีที่มีสมาชิกเข้ามาในกลุ่มที่เคยถูกตัดไป โดยการส่งข้อมูลดังกล่าวไปยังจุดที่ทำการตัดออกไป

ในกรณีที่มีเราเตอร์มากกว่าหนึ่งตัวในกลุ่มของสมาชิก จะมีเพียงตัวเดียวที่ถูกกำหนดให้เป็นตัวส่งข้อมูลมัลติคาสต์ โดยจะเลือกมาจากตัวที่อยู่ใกล้กับแหล่งกำเนิดข้อมูลมากที่สุด โดยที่ เราเตอร์ตัวอื่น ๆ จะไม่สนใจข้อมูลที่ส่งมาจากแหล่งกำเนิดนี้ แต่ถ้าเราเตอร์ในกลุ่มดังกล่าวมีระยะทางไปยังแหล่งกำเนิดเท่ากัน ตัวที่มีไอพีแอดเดรสต่ำกว่าจะเป็นตัวที่ถูกเลือก

วิธีคิวเอ็มอาร์ทีนี้สามารถสนับสนุนการทำอุโมงค์ (Tunnel, การเชื่อมต่อระหว่างมัลติคาสต์เราเตอร์โดยมีการผ่านเราเตอร์ที่ไม่สามารถทำมัลติคาสต์) โดยที่ทุก ๆ อินเตอร์เฟซดังกล่าวนั้น จะมีการกำหนดไอพีแอดเดรสของเราเตอร์มัลติคาสต์ต้นทางและปลายทางสำหรับการทำอุโมงค์ดังกล่าว ส่วนขอบเขตของไอพีมัลติคาสต์สามารถกำหนดได้จากทีทีแอลฟิลต์ (TTL) ของไอพีซึ่งแสดงในตารางที่ 2.5

ทีทีแอลเทรชโฮลด์ (TTL Threshold)	ขอบเขต
0	ในโฮสต์เดียวกัน
1	กลุ่มเครือข่ายเดียวกัน
15	ท้องถิ่นเดียวกัน
63	ภูมิภาค
127	ทั่วโลก
191	ทั่วโลก (จำกัดบนนิวริ์)
255	ไม่จำกัด

ตารางที่ 2.5 ค่าต่าง ๆ ที่ใช้ในการควบคุมขอบเขตทีทีแอล

2.5.4.2 เอ็มโอเอสพีเอฟ (Multicast Extensions to OSPF (MOSPF))

เอ็มโอเอสพีเอฟได้ถูกกำหนดไว้ในอาร์เอฟซี1584 (RFC1584) โดยจะใช้หลักการพื้นฐานของวิธีที่เรียกว่าโอเอสพีเอฟ (Open Shortest Path First (OSPF)) รุ่นที่ 2 (อ่านรายละเอียดเพิ่มเติมได้ที่อาร์เอฟซี1583 (RFC1583)) เอ็มโอเอสพีเอฟจะใช้ข้อมูลของกลุ่มสมาชิกผ่านวิธีการของโอจีเอ็มที ในการสร้างโครงข่ายต้นไม้ สำหรับการส่งข้อมูล โดยโครงสร้างจะเป็นเส้นทางที่สั้นที่สุดในระหว่างจุดรับกับจุดส่ง โดยเอ็ม โอเอสพีเอฟจะไม่สนับสนุนการทำอุโมงค์ แต่สามารถติดต่อกับเราเตอร์ที่ไม่ใช่เอ็ม โอเอสพีเอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็ม ไอเอสพีเอฟจะสนับสนุนการทำเราท์ติ้งแบบลำดับชั้น โดยทุก ๆ โฮสต์ในอินเทอร์เน็ต จะถูกแบ่งออกเป็นเอเอส (Autonomous Systems (AS)) และแต่ละเอเอสจะมีการแบ่งเป็นกลุ่มย่อย ๆ ที่เราเรียกว่าเอเรีย (Areas) โดยจะมีการเราท์ติ้งสำหรับเอ็ม ไอเอสพีเอฟอยู่สามแบบ ดังต่อไปนี้

2.5.4.2.1 อินทราเอเรียเราท์ติ้ง (Intra-Area Routing)

เนื่องจากไอเอสพีเอฟนั้น เป็นเราท์ติ้งโปรโตคอลแบบลิงค์สเตท (Link-State, มีการรู้สถานะของการเชื่อมต่อ) ที่ยอมให้มีการแบ่งเอเอสออกเป็นเอเรีย โดยที่ฐานข้อมูลของลิงค์สเตทจะปรากฏรายละเอียดของเอเรียดังกล่าวในทุก ๆ เราท์เตอร์ การหาและเก็บข้อมูลเกี่ยวกับตำแหน่งของสมาชิกของมัลติคาสต์ จะทำโดยการเพิ่มประเภทของลิงค์สเตทที่เรียกว่ากรุปเมมเบอร์ชิปเอลเอส (Group-Membership-LSA (Group-Membership Link State Advertisement)) จากข้อมูลดังกล่าว เส้นทางที่ดีที่สุดจากต้นทาง (Root) จะถูกสร้างด้วยวิธีการของ Dijkstra Algorithm จากนั้นสมาชิกในกลุ่มจะตัดเส้นทางที่ไม่มีสมาชิกออกจากกลุ่ม

ด้วยวิธีนี้ เราท์เตอร์ทุกตัวจะมีข้อมูลต่าง ๆ อยู่ครบถ้วน ที่ประกอบด้วยโครงสร้างต้นไม้ในการส่งข้อมูลระหว่างต้นทางไปยังปลายทางของทุก ๆ เราท์เตอร์ในพื้นที่นั้น ๆ การสร้างโครงสร้างนี้จะเป็นการสร้างขึ้นตามความต้องการ (On Demand) คือ เมื่อเราท์เตอร์ได้รับมัลติคาสต์ค่าตัวแกรมระหว่างต้นกำเนิดกับกลุ่มสมาชิก ก็จะทำให้การสร้างโครงสร้างต้นไม้ดังกล่าวขึ้นมา จากข้อมูลนี้ทำให้เราท์เตอร์ทราบว่ามีอินเตอร์เฟซไหนจะเป็นตัวรับหรือส่งข้อมูล ซึ่งจะต่างจากดีวีเอ็มอาร์พีที่แพ็คเกตแรกจะไม่ได้ถูกส่งออกไปทุก ๆ อินเตอร์เฟซ

2.5.4.2.2 อินเทอร์เอเรียเราท์ติ้ง (Inter-Area Routing)

ถ้าแหล่งรับหรือส่งข้อมูลไม่ได้อยู่ในเอเรียเดียวกัน วิธีการที่ผ่านมาน่าจะไม่สามารถใช้ได้ การแก้ไขสามารถทำได้โดยเลือกเราท์เตอร์ที่เป็นเอเรียบอร์เดอร์เราท์เตอร์ (Area Border Router) สำหรับทำหน้าที่ส่งข้อมูลมัลติคาสต์ ระหว่างกลุ่มอินเทอร์เอเรียมัลติคาสต์ฟอเวอร์คเตอร์ (Inter Area Multicast Forwarders) ซึ่งจะมีหน้าที่ในการส่งข้อมูลของกลุ่มทั้งหมดไปยังแบคโบนเอเรีย (Backbone Area) และด้วยแนวคิดของไวลด์การ์ดมัลติคาสตรีซีฟเวอร์ (Wild-Card Multicast Receiver) ในเอ็ม ไอเอสพีเอฟ ทำให้มันรับทุก ๆ มัลติคาสต์เมจเสจในเอเรียนั้น ๆ และอินเทอร์เอเรียมัลติคาสต์ในเอเรียที่ไม่มีแบคโบนไว้ และเพื่อให้แน่ใจว่าทุก ๆ ข้อมูลในเอเรียที่ไม่มีแบคโบนสามารถที่จะส่งไปถึงอินเทอร์เอเรียมัลติคาสต์ฟอเวอร์คเตอร์ และถ้าจำเป็นมันจะส่งต่อไปยังแบคโบนเอเรีย ดังนั้นในแบคโบนก็จะมีข้อมูลของกลุ่มสมาชิกที่อยู่ในเอเรียต่าง ๆ อย่างครบถ้วน และข้อมูลมัลติคาสต์ก็สามารถที่จะถูกส่งไปยังเอเรียที่เหมาะสมในเอเอสนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4.2 อินเทอร์เน็ตเราท์ติ้ง (Inter-AS Routing)

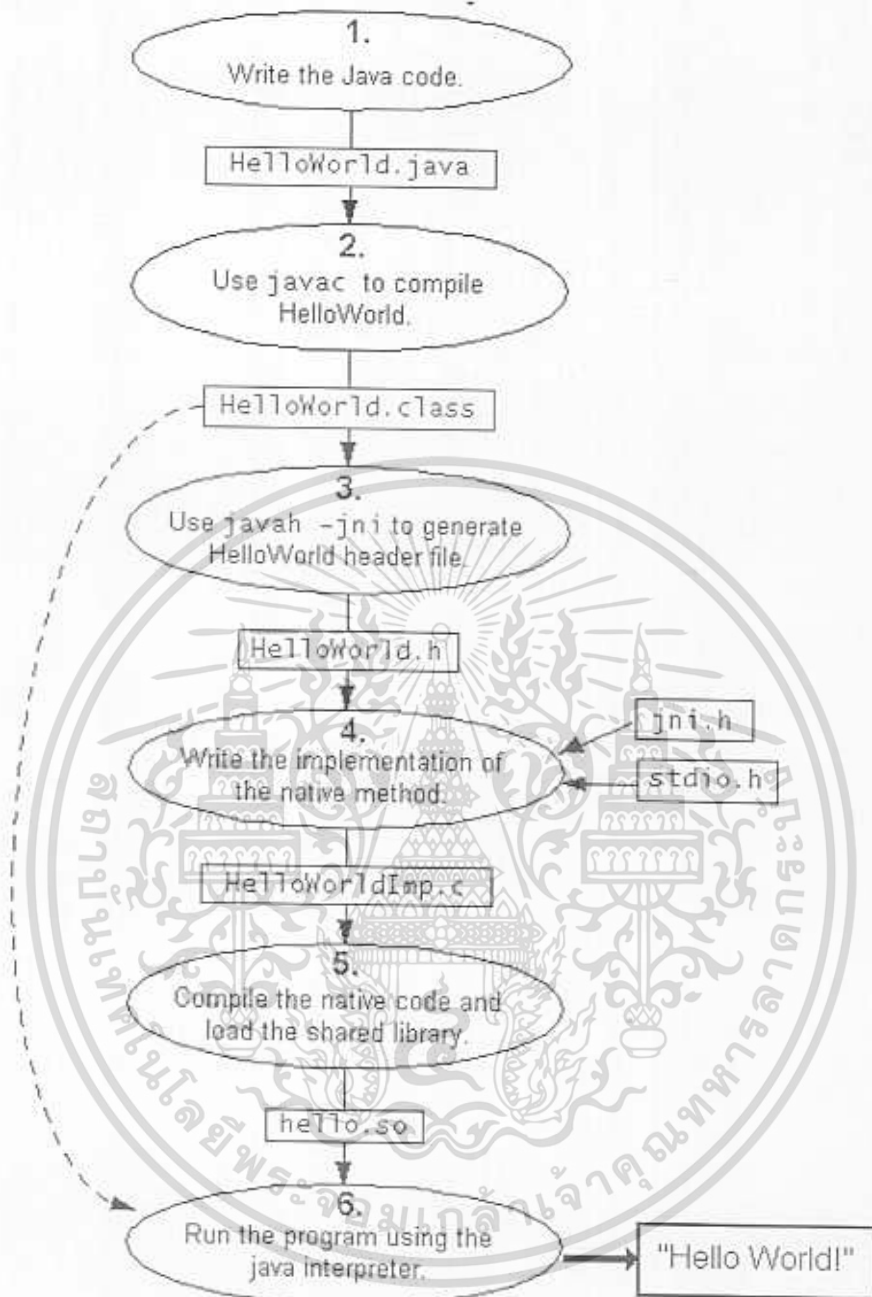
การรับส่งข้อมูลมัลติคาสต์ระหว่างเอเอสที่ต่างกัน จะคล้าย ๆ กับ อินเทอร์เน็ตเราท์ติ้ง โดย เอเอสบีอาร์ (AS Boundary Routers (ASBRs)) จะถูกกำหนดให้ทำหน้าที่เป็นอินเทอร์เน็ตเราท์ติ้งพอยต์ออฟเวอร์คเตอร์ โดยเอ็มไอเอสทีเอฟจะสร้างอาร์ทีพีทีพีสำหรับส่งข้อมูลมัลติคาสต์ ซึ่งอินเทอร์เน็ตเราท์ติ้งพอยต์ออฟเวอร์คเตอร์นี้จะเป็นไวลด์การ์ดมัลติคาสต์ของทางด้านรับในเอเอสที่ติดกัน และรับประกันว่าเราท์เตอร์ยังคงอยู่ในโครงสร้างของการส่งและรับข้อมูล ในขณะที่เส้นทางที่ส่งข้อมูลที่ใช้ภายในเอเอสไปยังแหล่งข้อมูลภายนอก จะใช้วิธีที่เราเรียกกันว่า รีเวอร์สพาร์ทซอร์สเบสทรี (Reverse-Path Source-Based Trees)

2.6 การเขียนโปรแกรมจาวาด้วยเนทีฟเมธอด

การเขียนเนทีฟเมธอดสำหรับโปรแกรมด้วยภาษาจาวามีขั้นตอนดังนี้

1. เริ่มต้น โดยการเขียนโปรแกรมด้วยภาษาจาวา เนื่องจากการสร้างกลาสจาวาที่มีการประกาศเนทีฟเมธอดนั้น จะต้องประกอบด้วยการประกาศเนทีฟเมธอด หรือระบุชนิดของตัวแปรของเนทีฟเมธอด รวมทั้งภายในเมธอดเมน (Main Method) ก็จะต้องเขียนให้มีการเรียกเนทีฟเมธอดด้วย
2. คอมไพล์กลาสจาวาที่มีการประกาศเนทีฟเมธอดและเมธอดเมน
3. สร้างไฟล์เฮดเดอร์ที่มีรูปแบบสำหรับเนทีฟเมธอด โดยใช้คำสั่ง javah -jni ซึ่งจะสร้างไฟล์เฮดเดอร์ที่มีชนิดของตัวแปรสำหรับเนทีฟเมธอดนั้น
4. เขียนเนทีฟเมธอดด้วยภาษาโปรแกรมที่เลือกไว้ เช่น ภาษาซี (C) หรือซีพลัสพลัส (C++)
5. คอมไพล์ไฟล์เฮดเดอร์และสร้างไฟล์เพื่อทำการแลกเปลี่ยนไฟล์ไบนารี
6. รันโปรแกรมจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า-
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.32 แสดงขั้นตอนการสร้าง โปรแกรม Hello World

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 การเขียนโค้ดภาษาจาวา

```
class HelloWorld {
    public native void displayHelloWorld();

    static {
        System.loadLibrary("hello");
    }

    public static void main(String[] args) {
        new HelloWorld().displayHelloWorld();
    }
}
```

จากตัวอย่างโค้ดภาษาจาวาด้านบน เป็นการสร้างคลาส HelloWorld โดยที่คลาสจะมีการประกาศเนทีฟเมธอด การสร้างเมธอดเม้น และส่วนของโค้ดสแตติก

2.6.1.1 การประกาศเนทีฟเมธอด

เราต้องประกาศเมธอดทั้งหมด ไม่ว่าจะเขียนจาวาเมธอดหรือเนทีฟเมธอด ภายในคลาสที่เขียนด้วยภาษาจาวา และเมื่อสร้างเมธอดจกภาษาอื่นที่ไม่ใช่ภาษาจาวา จะต้องเพิ่มคำประกาศ native เข้าไปในการเขียนเมธอดภายในภาษาจาวาด้วย โดยคำประกาศ native จะเป็นตัวบอกให้คอมไพเลอร์ทราบว่า ฟังก์ชันนั้นเป็นฟังก์ชันที่สร้างมาจากภาษาอื่น ซึ่งทำให้เราเข้าใจได้ง่ายว่า เมธอด displayHelloWorld ของ คลาส HelloWorld นั้น ถูกเขียนขึ้นมาด้วยภาษาอื่น เพราะคำประกาศ native บอกให้เราทราบถึงคำจำกัดความที่มาของเมธอดอยู่แล้ว

```
public native void displayHelloWorld();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประกาศเนทีฟเมธอดในคลาสภาษาจาวาตรงส่วนนี้ เราจะนำไปใช้ในเมธอด `displayHelloWorld` เท่านั้น ไม่ต้องสร้างเมธอดนั้นด้วยภาษาจาวา แต่จะสร้างเมธอดด้วยภาษาอื่น ตามที่ต้องการแทน ซึ่งการประกาศเมธอดสำหรับ `displayHelloWorld` จะประกาศเมธอดเป็น ประเภทสาธารณะ (Public) ไม่ยอมรับค่าใด ๆ และไม่มีารส่งค่ากลับ

2.6.1.2 การโหลดไลบรารี

เมื่อเราคอมไพล์โค้ดโปรแกรมภาษาอื่น มันจะสร้าง `displayHelloWorld` เป็นแชร์ไลบรารี ซึ่งในขณะที่รันระบบ จะมีการโหลดแชร์ไลบรารีนี้เข้ามาที่คลาสจาวา การโหลดไลบรารีเข้าไปยังจาวาคลาสจะทำการกำหนดตามการประกาศเนทีฟเมธอดที่ประกาศในคลาสจาวา

คลาส `HelloWorld` จะใช้เมธอด `System.loadLibrary` ซึ่งจะทำการโหลดแชร์ไลบรารี โดยแชร์ไลบรารีนี้จะถูกสร้างเมื่อทำการคอมไพล์ ซึ่งจะวางไว้ในตำแหน่งของส่วนสแตคเริ่มต้น ค่าที่ส่งไปยัง `System.loadLibrary` จะเป็นชื่อของแชร์ไลบรารีซึ่งจะเป็นชื่ออะไรก็ได้ โดยระบบจะเปลี่ยนชื่อของแชร์ไลบรารีไปเป็นชื่อเนทีฟไลบรารีตามมาตรฐานของแต่ละแพลตฟอร์ม ยกตัวอย่างเช่น ในระบบโซลาริส (Solaris system) มันจะเปลี่ยนชื่อไลบรารี "hello" ไปเป็น `libhello.so` ขณะที่ระบบของไมโครซอฟท์วินโดวส์ (Microsoft Windows system) จะเปลี่ยนเป็นชื่อเดียวกันคือ `hello.dll`

ค่าสแตคเริ่มต้นจากคลาส `HelloWorld` จะถูกโหลดไปยังไลบรารีที่เหมาะสม ซึ่งในที่นี้ก็คือ ชื่อ `hello` โดยระบบปฏิบัติการจะทำการรันค่าสแตคเริ่มต้นนี้เมื่อมัน โหลดคลาส

```
static {
    System.loadLibrary("hello");
}
```

2.6.1.3 การเขียนเมธอดเมน

คลาส `HelloWorld` นั้น ประกอบด้วยเมธอดเมนที่ใช้ระบุคลาสและเรียกเนทีฟเมธอด จากโค้ดตัวอย่างนี้ เมธอดเมนได้ระบุถึงคลาส `HelloWorld` และยังใช้เรียกเนทีฟเมธอด `displayHelloWorld` อีกด้วย

```
public static void main(String[] args) {
    new HelloWorld().displayHelloWorld();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างแสดงให้เห็นว่า สามารถเรียกเนทีฟเมธอดได้เหมือนกับการเรียกเมธอดทั่วไป โดยชื่อของเมธอดที่ถูกเรียกจะทำชื่อวัตถุ ซึ่งจะกันด้วยจุด (.) และต้องมียวงเล็บ () ตามหลังชื่อของเมธอดด้วย โดยจะมีการส่งค่าเข้าไปยังเมธอด ในที่นี้เมธอด `displayHelloWorld` ไม่มีการส่งค่าเข้าไป

2.6.2 การคอมไพล์โค้ดจาวา

ใช้ตัวคอมไพเลอร์ของภาษาจาวา ในการคอมไพล์คลาสที่สร้างขึ้นจากข้อ 2.6.2 โดยใช้รูปแบบคำสั่งนี้

```
javac HelloWorld.java
```

2.6.3 การสร้างไฟล์เฮดเดอร์

ในขั้นตอนนี้ เราจะใช้โปรแกรมช่วยเหลือ `javah` เพื่อทำการสร้างไฟล์เฮดเดอร์ (.h file) จากคลาส `HelloWorld` ซึ่งไฟล์เฮดเดอร์นี้จะเป็นการระบุถึงฟังก์ชันที่เขียนด้วยภาษา C ตามที่เนทีฟเมธอด `displayHelloWorld` ได้ประกาศไว้ภายในคลาสแล้ว

ชื่อของไฟล์เฮดเดอร์จะเป็นชื่อของคลาสจาวานั้นตามด้วย .h ข้างหลัง ยกตัวอย่างเช่น จากโค้ดคำสั่งด้านบน เราจะได้ไฟล์เฮดเดอร์ออกมาชื่อ `HelloWorld.h` ซึ่งตามปกติแล้ว `javah` จะสร้างไฟล์ .h ใหม่ไว้ที่ใคร่กทอริเดียวกับไฟล์ .class ซึ่งเราสามารถใช้คำสั่ง `-d` เพื่อบอกกับ `javah` ว่าต้องการจะวางเฮดเดอร์ไฟล์ไว้ที่ใคร่กทอริอื่น

2.6.3.1 การประกาศฟังก์ชัน

```
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class HelloWorld */

#ifdef _Included_HelloWorld
#define _Included_HelloWorld
#endif
#ifdef __cplusplus
extern "C" {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#endif
/*
 * Class: HelloWorld
 * Method: displayHelloWorld
 * Signature: ()V
 */

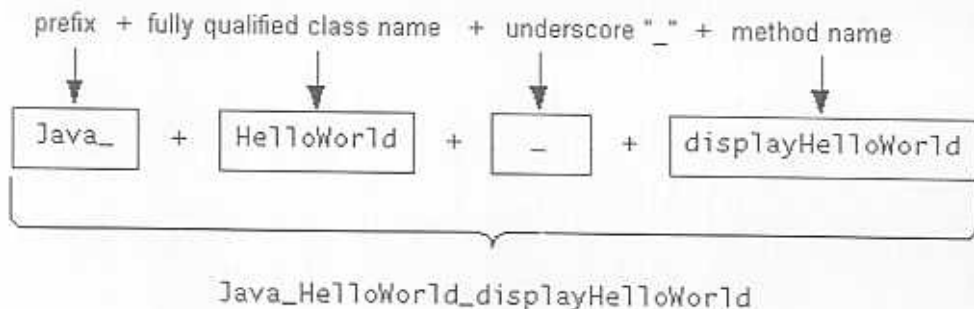
JNIEXPORT void JNICALL Java_HelloWorld_displayHelloWorld
(JNIEnv *, jobject);

#ifdef __cplusplus
}
#endif
#endif

```

จากตัวอย่างไฟล์เฮดเดอร์ HelloWorld.h ฟังก์ชัน Java_HelloWorld_displayHelloWorld จะระบุถึงเมธอด displayHelloWorld ของคลาส HelloWorld ซึ่งเราจะสร้างขึ้นในข้อ 2.6.4 ซึ่งสามารถใช้ลักษณะของฟังก์ชันแบบเดียวกันนี้ได้ เมื่อต้องการสร้างเมธอด ถ้า HelloWorld มีเมธอดอื่น ๆ อีก ฟังก์ชันนั้นก็จะมีปรากฏที่เหมือนกัน

ชื่อของเมธอดฟังก์ชันที่ถูกสร้างจากเมธอด จะมี Java_ นำหน้า ตามด้วยชื่อแพ็คเกจ, ชื่อคลาส และชื่อของเมธอด โดยระหว่างชื่อของแต่ละส่วนนั้น จะถูกคั่นด้วยขีดเส้นใต้ () ดังรูปที่ 2.33



รูปที่ 2.33 แสดงให้เห็นถึงรูปแบบชื่อของเมธอดฟังก์ชันที่ถูกสร้างจากเมธอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด `displayHelloWorld` ภายในคลาส `HelloWorld` นี้ จะถูกเปลี่ยนชื่อเป็น `Java_HelloWorld_displayHelloWorld` จากตัวอย่างจะไม่มีชื่อแพ็คเกจปรากฏอยู่ เนื่องจากคลาส `HelloWorld` อยู่ภายในดิฟอลท์แพ็คเกจ (Default package) จึงไม่มีชื่อ

การสร้างเมธอดฟังก์ชันอย่างที่ปรากฏในไฟล์เฮดเดอร์ จะมีการรับพารามิเตอร์ 2 ตัว โดยทางด้านจาวานั้น จะไม่มีการรับค่าพารามิเตอร์ แต่เจเอ็นไอ (JNI) จำเป็นที่จะต้องกำหนดให้ทุก ๆ เมธอด ต้องมีการรับค่าพารามิเตอร์ 2 ตัว

พารามิเตอร์แรกของทุก ๆ เมธอดคือ `JNIEnv` ซึ่งเป็นอินเทอร์เฟซพอยเตอร์ (Interface Pointer) โดยค่าต่าง ๆ จะถูกส่งผ่านพอยเตอร์ เมื่อเมธอดได้ทำการแอสเซส (access) มายังพารามิเตอร์นั้น ส่วนอีกตัวจะถูกส่งผ่านพอยเตอร์ไปยังจาวาแอปพลิเคชัน พารามิเตอร์สองคือ `jobject` ซึ่งจะอ้างอิงถึงอ็อบเจกต์ปัจจุบัน ซึ่งเราสามารถเทียบพารามิเตอร์ `jobject` เท่ากับตัวแปร "this" ในภาษาจาวา สำหรับเมธอดที่อินสแตนซ์เมธอด ซึ่งได้แก่ เมธอด `displayHelloWorld` ในตัวอย่าง `jobject` จะเป็นตัวแปรเพื่ออ้างถึงค่าปัจจุบันของอ็อบเจกต์ สำหรับเมธอดคลาสเมธอดตัวแปร `jobject` จะเป็นการอ้างอิงเมธอดของคลาสจาวานั้น ซึ่งจากตัวอย่างนั้น ได้ละเว้นตัวแปรทั้งสองไว้

2.6.4 การสร้างเมธอด

ในขั้นตอนที่เราจะทำการสร้างเมธอดจากภาษาอื่นที่ไม่ใช่ภาษาจาวา ซึ่งฟังก์ชันที่เขียนขึ้นนั้นเป็นที่จะต้องมีรูปแบบเดียวกับฟังก์ชันที่ได้จากการใช้คำสั่ง `javah` ในไฟล์ `HelloWorld.h` จากข้อ 2.6.3 โดยรูปแบบที่ได้จากการสร้างเมธอด `displayHelloWorld` ของคลาส `HelloWorld` จะมีรูปแบบดังนี้

```
JNIEXPORT void JNICALL
Java_HelloWorld_displayHelloWorld(JNIEnv *, jobject);
```

จากจุดนี้ เราสามารถที่จะใช้ภาษาซี (C) ในการเขียนโปรแกรมเพื่อทำการสร้างเมธอด `Java_HelloWorld_displayHelloWorld` ได้ โดยไฟล์นี้จะมีชื่อว่า `HelloWorldImp.c`

```
#include <jni.h>
#include "HelloWorld.h"
#include <stdio.h>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JNIEXPORT void JNICALL
Java_HelloWorld_displayHelloWorld(JNIEnv *env, jobject obj)
{
    printf("Hello world!\n");
    return;
}

```

โดยการสร้างฟังก์ชัน `Java_HelloWorld_displayHelloWorld` มีลักษณะทั่วไป โดยฟังก์ชันจะใช้คำสั่ง `printf` เพื่อทำการแสดงข้อความ "Hello World!" และจากนั้นจึงรีเทิร์น (Return) กลับมาไฟล์เฮดเดอร์ที่ประกอบใน `HelloWorldImp.c` มีสามไฟล์ ได้แก่

1. `jni.h` เป็นไฟล์เฮดเดอร์ที่ให้ข้อมูลว่า ต้องการเน็ตเวิร์กใดเมื่อต้องการจะติดต่อกับระบบปฏิบัติการจาวา เมื่อเราสร้างเน็ตเวิร์ก จะต้องรวมไฟล์นี้ใส่ลงไปเน็ตเวิร์กไฟล์ด้วย
2. `HelloWorld.h` เป็นไฟล์ `.h` ที่ได้จากข้อ 2.6.3
3. `stdio.h` เป็นส่วนที่ใช้เพื่อทำการเรียกฟังก์ชัน `printf`

2.6.5 การสร้างแชร์ไลบรารี

จากข้อ 2.6.1 เราได้ใช้เมธอดด้านล่างนี้ เพื่อทำการเรียกโหลดแชร์ไลบรารีที่ชื่อว่า `hello` มายังโปรแกรมเมื่อเริ่มทำงาน

```
System.loadLibrary("hello");
```

ในขั้นตอนที่ผ่านมา เราได้ทำการสร้างไฟล์ภาษาซี ซึ่งเราได้เขียนไว้แล้วในเน็ตเวิร์ก `displayHelloWorld` โดยจะเก็บเน็ตเวิร์กไว้ในไฟล์ `HelloWorldImp.c` ขั้นตอนนี้ เราจะต้องทำการคอมไพล์ไฟล์ `HelloWorldImp.c` ไปยังแชร์ไลบรารีที่มีชื่อว่า `hello` เพื่อให้สอดคล้องกับชื่อไลบรารีที่ใช้ในเมธอด `System.loadLibrary`

จากนั้นทำการคอมไพล์โค้ดเน็ตเวิร์กที่สร้างขึ้นในขั้นตอนก่อนไปยังแชร์ไลบรารี โดยในระบบโซลาริสจะเป็นการสร้างแชร์ไลบรารี ส่วนในระบบวินโดวส์จะทำการสร้างไฟล์ `dll` จากนั้นเราจะต้องระบุพาธลงไปในส่วนที่เกี่ยวข้องหรือระบุลงในไฟล์เฮดเดอร์ที่จำเป็นทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบโซลาริสจะใช้คำสั่งดังต่อไปนี้ในการสร้างแชร์ไลบรารี libhello.so

```
cc -G -I/usr/local/java/include -I/usr/local/java/include/solaris \
HelloWorldImp.c -o libhello.so
```

ในระบบวินโดวส์ จะใช้คำสั่งต่อไปนี้เพื่อสร้างไฟล์ hello.dll โดยใช้โปรแกรมไมโครซอฟต์
วิซวลซีพลัสพลัส 4.0 (Microsoft Visual C++ 4.0)

```
cl -Ic:\java\include -Ic:\java\include\win32  
-LD HelloWorldImp.c -Fehello.dll
```

2.6.6 การรันโปรแกรม

รันจาวาแอปพลิเคชัน (คลาส HelloWorld) ด้วยตัวแปลภาษาจาวา โดยใช้คำสั่งนี้

```
java HelloWorld
```

โดยจะได้ผลลัพธ์ดังนี้

```
Hello World!
```

แต่ถ้ามีเอ็กซ็เซ็ปชัน (Exception) ดังตัวอย่างนี้ แสดงว่ายังไม่มีการเชื่อมต่อให้ถูกต้อง

```
java.lang.UnsatisfiedLinkError: no hello in shared library path  
at java.lang.Runtime.loadLibrary(Runtime.java)  
at java.lang.System.loadLibrary(System.java)  
at  
at java.lang.Thread.init(Thread.java)
```

ไลบรารีพาทจะเป็นรายการของไดเรกทอรีทั้งหมด ที่ระบบปฏิบัติการจาวาจะทำการค้นหา
เมื่อต้องการโหลดไลบรารี โดยต้องแน่ใจว่าไลบรารี hello มีอยู่ในนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ

3.1 ข้อมูลเบื้องต้น

ทำการเก็บข้อมูลการใช้งานระบบวิทยุกระจายเสียงผ่านอินเทอร์เน็ตจากผู้ใช้ทั่วไป และจากการศึกษาแอปพลิเคชันตัวอย่าง ทำให้สามารถออกแบบฟังก์ชันต่าง ๆ ออกมาได้ ดังนี้

3.1.1 ฟังก์ชันสำหรับผู้ดูแลระบบ (Administrator)

1. แสดงสถานะของระบบ
2. แสดงรายการของแหล่งข้อมูลเสียงโฆษณา
3. เลือกเสียงโฆษณาไว้รอออกอากาศ
4. แสดงรายการ โฆษณาที่เลือกไว้
5. สร้างผังรายการล่วงหน้า
6. แก้ไขผังรายการล่วงหน้า
7. ลบผังรายการล่วงหน้า
8. แสดงรายการของผังรายการที่สร้างไว้ล่วงหน้า
9. แสดงรายละเอียดของผังรายการที่สร้างไว้
10. ควบคุมการเล่นเสียงโฆษณา
11. ควบคุมการผสมเสียง

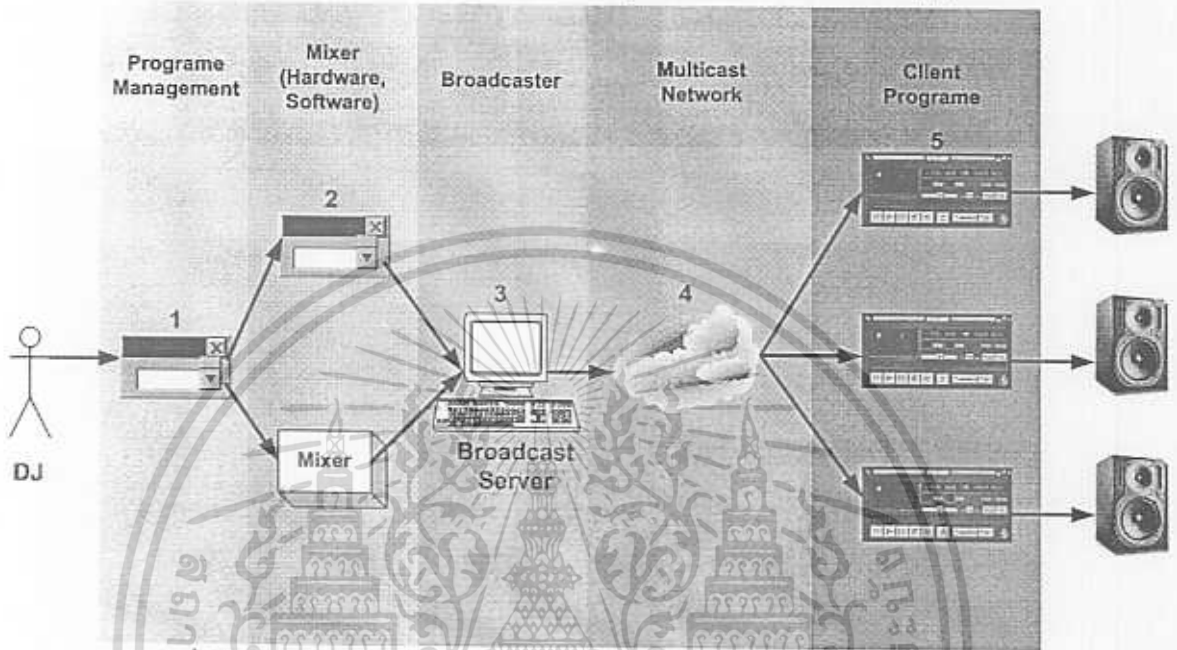
3.1.2 ฟังก์ชันสำหรับผู้ดำเนินรายการ (DJ)

1. แสดงรายการของแหล่งข้อมูลเสียงโฆษณา
2. เลือกเสียงเพลงไว้รอออกอากาศ
3. แสดงรายการของเพลงที่เลือกไว้
4. ควบคุมการเล่นเสียงเพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบระบบ

3.2.1 การออกแบบสถาปัตยกรรมของระบบ (System Architecture Design)



รูปที่ 3.1 สถาปัตยกรรมของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ต

3.2.1.1 ส่วนจัดการผังรายการ (Program Management Module)

หน้าที่หลักคือ จัดทำผังรายการและควบคุมการดำเนินรายการ มีสองโหมดการทำงานคือ แบบควบคุมด้วยมือ (Manual Mode) และแบบดำเนินรายการอัตโนมัติ (Auto Mode)

3.2.1.2 ส่วนมิกซ์เซอร์ (Mixer)

เป็นส่วนควบคุมระดับความดังค่อย และผสมเสียงที่จะส่งไปกระจายเสียงที่ส่วนบรอดคาสต์เซิร์ฟเวอร์ ในส่วนนี้ออกแบบให้สามารถใช้ได้ทั้งมิกซ์เซอร์ที่เป็นฮาร์ดแวร์ และมิกซ์เซอร์ที่เป็นซอฟต์แวร์

3.2.1.3 ส่วนบรอดคาสต์เซิร์ฟเวอร์ (Broadcast Server)

ทำหน้าที่รับข้อมูลเสียงจากมิกซ์เซอร์ และกระจายเสียงเหล่านั้นไปในเครือข่ายอินเทอร์เน็ต โดยมีรูปแบบการส่งแบบมัลติคาสต์ โดยระบุเป้าหมายเป็นมัลติคาสต์แอดเดรสหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.4 ส่วนเครือข่ายมัลติคาสต์ (Multicast Network)

ประกอบด้วยมัลติคาสต์เราเตอร์ (Multicast Router) ที่สนับสนุนไอพีมัลติคาสต์ และการเชื่อมต่อของไคลเอนท์มายังไอพีมัลติคาสต์

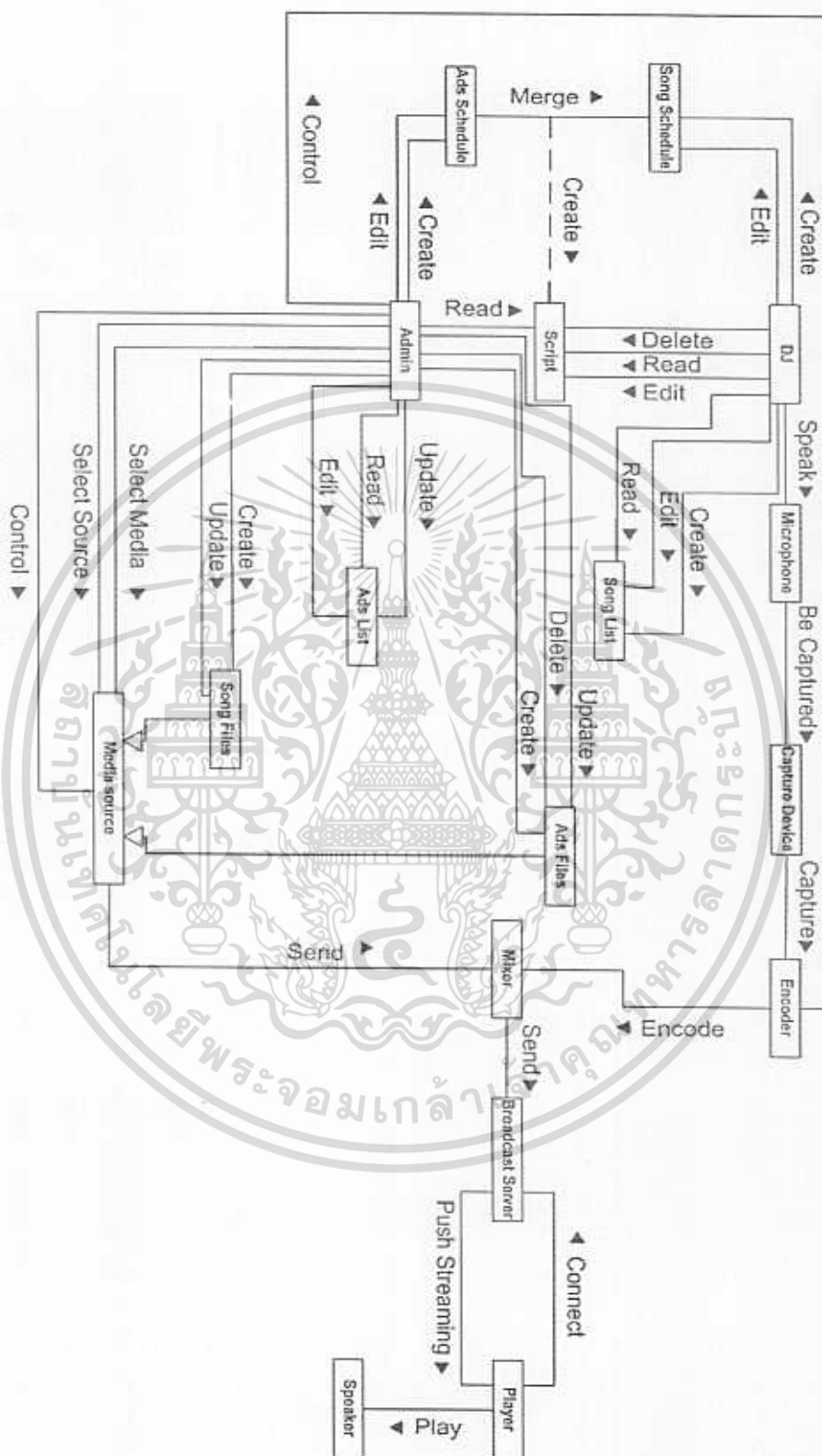
3.2.1.5 ส่วนไคลเอนท์โปรแกรม

ออกแบบให้ใช้โปรแกรมวินแอมป์ (Winamp) เนื่องจากเป็นโปรแกรมสำหรับฟังเพลงที่นิยมใช้กันทั่วไป และสามารถรองรับการเชื่อมต่อแบบมัลติคาสต์ และใช้โปรโตคอลฮาร์ทีบีได้

3.2.2 โดเมนโมเดล (Domain Model)



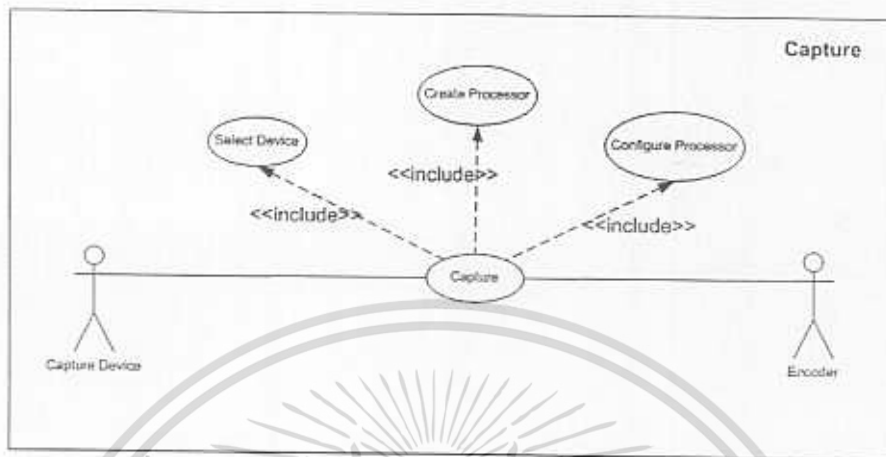
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



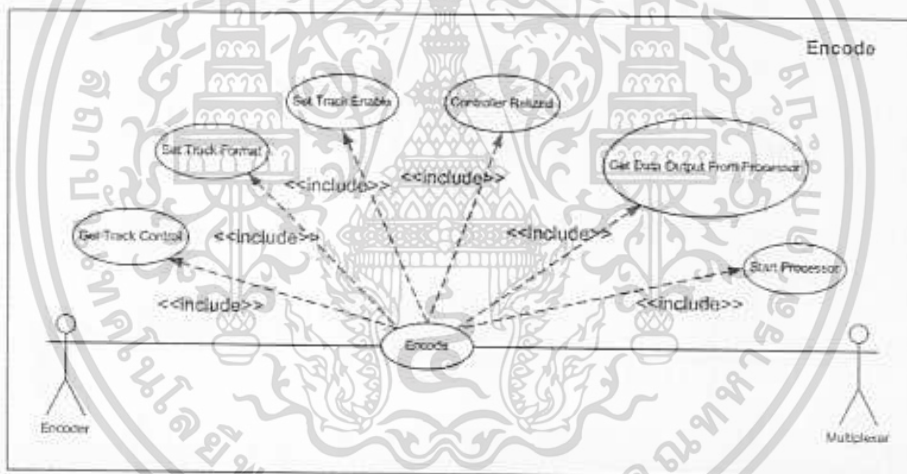
รูปที่ 3.2 โดเมนโมเดลของระบบจัดการการกระจายเสียงผ่านอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

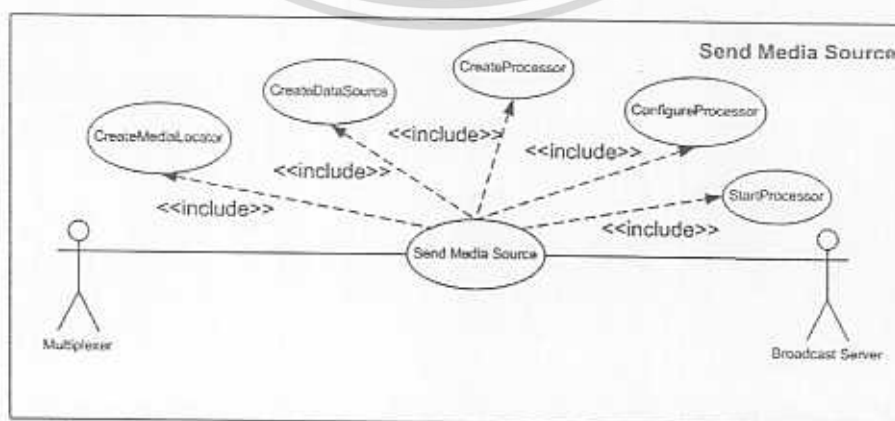
3.2.3 ยูสเคสไดอะแกรม (Use Case Diagram)



รูปที่ 3.3 ยูสเคสไดอะแกรมสำหรับการจับเสียงจากไมโครโฟน

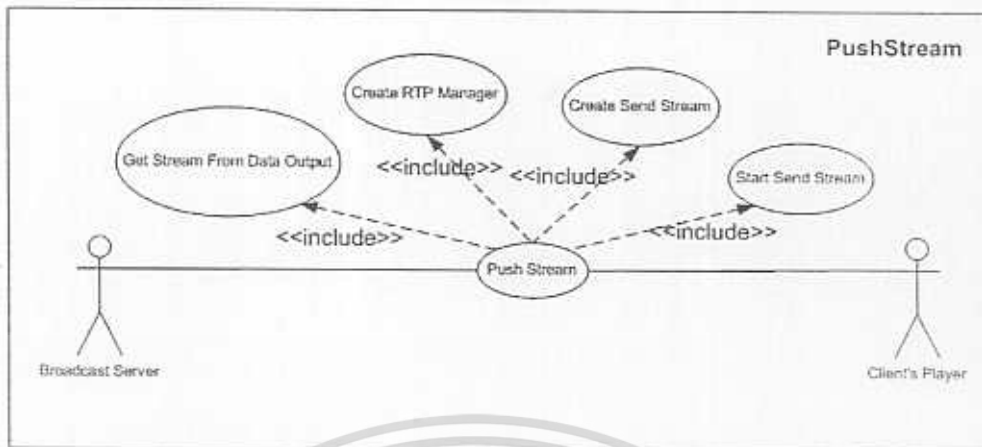


รูปที่ 3.4 ยูสเคสไดอะแกรมสำหรับการเข้ารหัสสัญญาณเสียงให้อยู่ในรูปแบบแพ็คเกจอาร์ทีพี



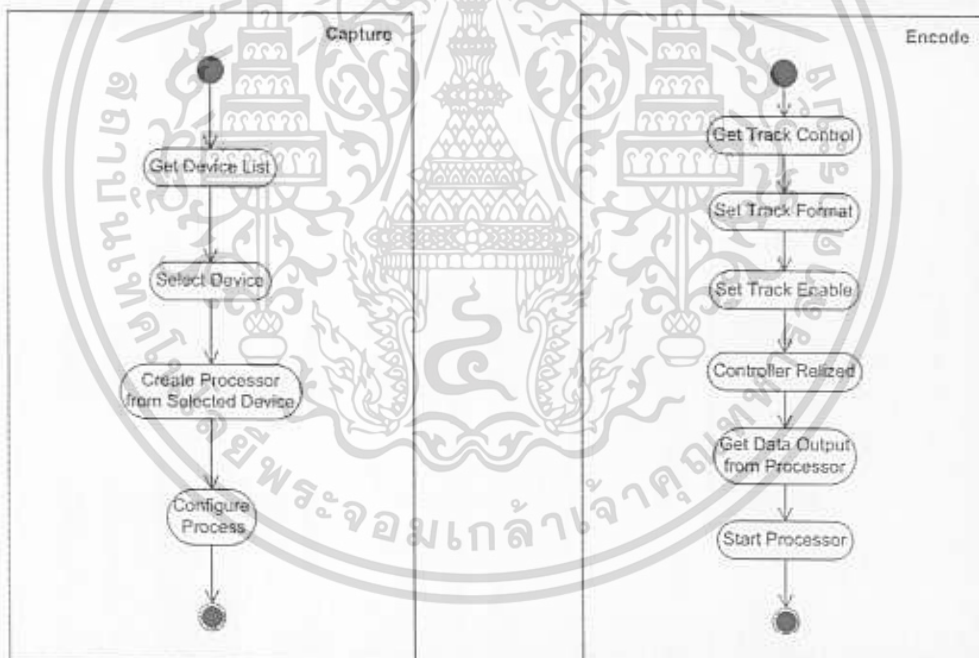
รูปที่ 3.5 ยูสเคสไดอะแกรมสำหรับการเลือกสัญญาณเสียงสำหรับส่งมีเดียสตรีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ยูสเคสไดอะแกรมสำหรับการส่งสตรีมเสียงไปยังไคลเอนท์

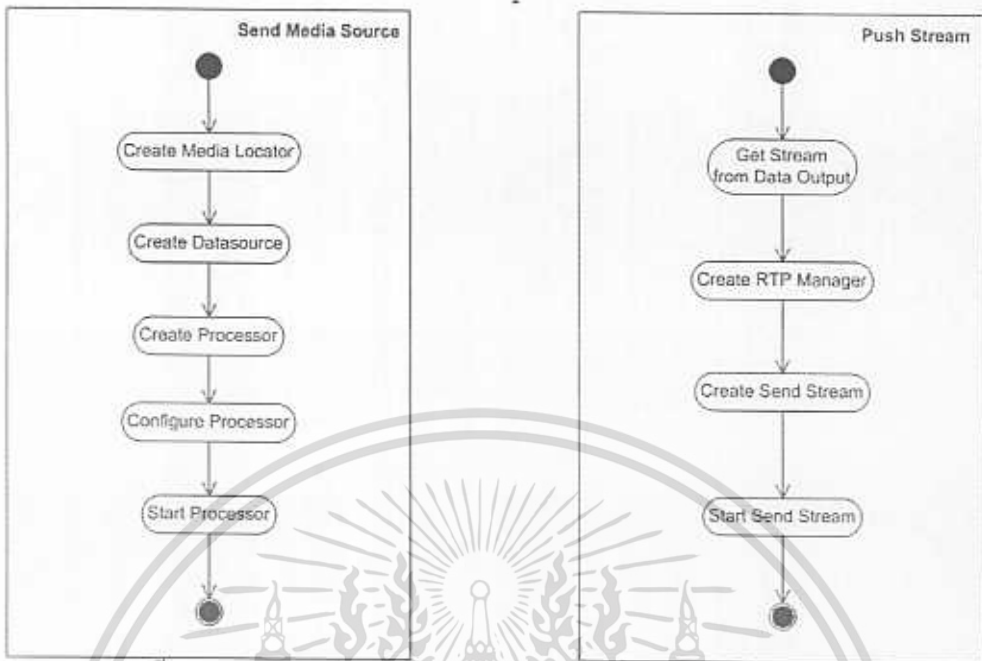
3.2.4 แอกทิวิตีไดอะแกรม (Activity Diagram)



รูปที่ 3.7 แอกทิวิตีไดอะแกรมสำหรับการจับเสียงจากไมโครโฟน

รูปที่ 3.8 แอกทิวิตีไดอะแกรมสำหรับการเข้ารหัสสัญญาณเสียงให้อยู่ในรูปแบบแพ็คเกจอาร์ทีพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



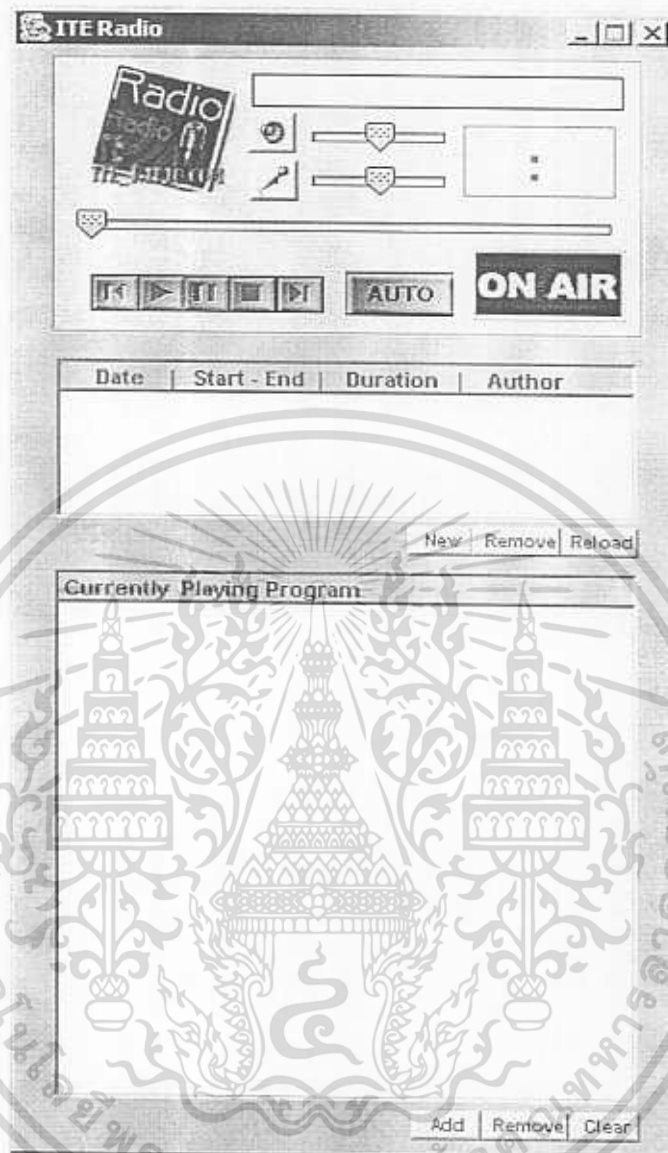
รูปที่ 3.9 แอคตวิตีไดอะแกรมสำหรับการเลือก รูปที่ 3.10 แอคตวิตีไดอะแกรมสำหรับการการ
สัญญาณเสียงสำหรับส่งมีเดียสตรีม ส่งสตรีมเสียงไปยังไคลเอนท์

3.2.5 การออกแบบส่วนติดต่อผู้ใช้ (User Interface)

3.2.5.1 หน้าต่างหลัก

หน้าต่างหลักมีหน้าที่เป็นเครื่องมือช่วยในการควบคุมการออกอากาศ และรายงานสถานะปัจจุบันของระบบ ประกอบด้วยสามส่วนดังนี้

1. ส่วนควบคุมการออกอากาศ
2. ส่วนแสดงผังรายการล่วงหน้า
3. ส่วนรายละเอียดของผังรายการปัจจุบัน



รูปที่ 3.11 หน้าต่างหลักสำหรับการจัดการกระจายเสียงผ่านอินเทอร์เน็ต

3.2.5.1.1 ส่วนควบคุมการออกอากาศ

1. ส่วนแสดงสถานการณ์ทำงานปัจจุบันของแอปพลิเคชัน ประกอบด้วย

- 1) ชื่อมีเดียที่กำลังเล่น (Now Playing)
- 2) แสดงตำแหน่งทางเวลาของมีเดียที่กำลังเล่น
- 3) สไลด์บาร์ (Slider Bar) แสดงตำแหน่งทางภาพของการเล่นเพลงในมีเดียหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนการผสมเสียง ประกอบด้วย

- 1) ปุ่มสำหรับเปิดและปิดเสียงไมโครโฟน
- 2) ปุ่มสำหรับเปิดและปิดเสียงเพลง
- 3) สไลด์บาร์สำหรับควบคุมระดับเสียงไมโครโฟน
- 4) สไลด์บาร์สำหรับควบคุมระดับเสียงเพลง

3. ส่วนควบคุมมีเดีย ประกอบด้วย

- 1) ปุ่มเพลย์ (Play Button)
- 2) ปุ่มพอส (Pause Button)
- 3) ปุ่มสต็อป (Stop Button)
- 4) ปุ่มพรีVIOUS (Previous Button)
- 5) ปุ่มเนกซ์ (Next Button)

4. ส่วนเลือกโหมดการทำงาน เลือกได้สอง โหมด คือ

- 1) โหมดการทำงานอัตโนมัติ (Auto Mode)
- 2) โหมดควบคุมด้วยมือ (Manual Mode)

5. ส่วนเปิดปิดการกระจายเสียง คือ

- 1) ปุ่มออนแอร์ (On Air Button)

3.2.5.1.2 ส่วนแสดงผังรายการล่วงหน้า

1. ลิสต์แสดงรายการของผังรายการล่วงหน้า แสดงรายละเอียดเบื้องต้นดังนี้

- 1) วันที่ที่กำหนดให้ออกอากาศ (Date)
- 2) ช่วงเวลาที่กำหนดให้ออกอากาศ (Start-End)
- 3) ความยาวของรายการที่สร้าง (Duration)
- 4) ชื่อผู้สร้างลิสต์ (Author)

2. ปุ่มนิวลิสต์ (New List Button)

3. ปุ่มรีมูฟ (Remove Button)

4. ปุ่มอีดิท (Edit Button)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

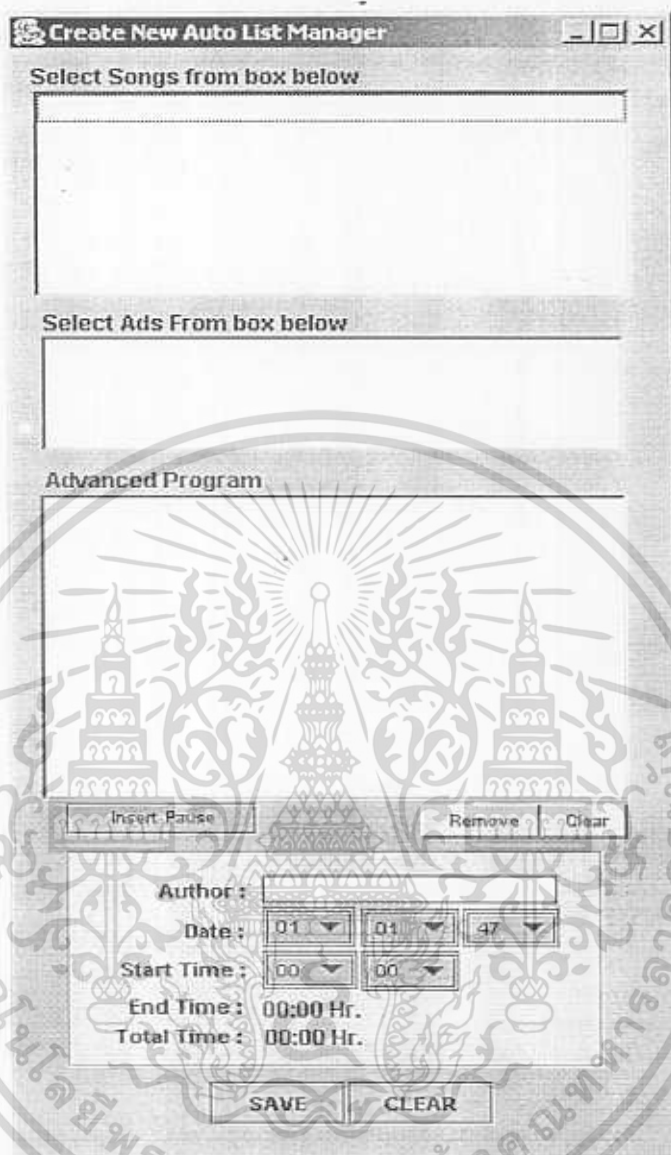
3.2.5.1.3 ส่วนแสดงผังรายการปัจจุบัน

1. ลิสต์แสดงรายการมีเดียที่เลือกไว้ แสดงรายละเอียดของมีเดีย ดังนี้
 - 1) ชื่อมีเดีย (Media Name)
 - 2) ความยาวของมีเดีย (Media Duration)
2. ปุ่มบราวซ์ (Browse Button)
3. ปุ่มรีมูฟ (Remove Button)
4. ปุ่มเคลียร์ (Clear Button)

3.2.5.2 หน้าต่างสำหรับสร้างผังรายการล่วงหน้า

หน้าต่างสำหรับสร้างผังรายการล่วงหน้า ดังรูปที่ 3.13 ประกอบด้วยสี่ส่วนหลักคือ

1. ส่วนแสดงแหล่งข้อมูลเสียง
2. ส่วนแสดงรายการที่สร้าง
3. ส่วนกำหนดการออกอากาศ
4. ส่วนบันทึกผังรายการลงไฟล์



รูปที่ 3.12 หน้าต่างสำหรับสร้างผังรายการล่วงหน้า

3.2.5.2.1 ส่วนแสดงรายชื่อแหล่งข้อมูลเสียง

1. ลิสต์สำหรับเลือกแหล่งข้อมูลเสียงเพลง (Song List) แสดง
 - 1) ชื่อมีเดีย (Media Name)
 - 2) ความยาวของมีเดีย (Media Duration)
2. ลิสต์สำหรับเลือกแหล่งข้อมูลเสียงโฆษณา (Ads List) แสดง
 - 1) ชื่อมีเดีย (Media Name)
 - 2) ความยาวของมีเดีย (Media Duration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5.2.2 ส่วนแสดงรายการที่สร้าง

1. ลิสต์แสดงรายการของข้อมูลเสียงที่เลือก (Advanced Program) แสดง
 - 1) ชื่อมีเดีย (Media Name)
 - 2) ความยาวของมีเดีย (Media Duration)
2. ปุ่มอินเสิร์ทพอส (Insert Pause Button)
3. ปุ่มรีมูฟ (Remove Button)
4. ปุ่มเคลียร์ (Clear Button)

3.2.5.2.3 ส่วนกำหนดการออกอากาศ

1. ก่อตั้งข้อความสำหรับกรอกชื่อผู้สร้างลิสต์ (Author)
2. กอมโบบ็อกซ์สำหรับเลือกวันที่ออกอากาศ (Date)
3. กอมโบบ็อกซ์สำหรับเลือกเวลาเริ่มออกอากาศ (Start Time)
4. ลาเบลแสดงผลการคำนวณเวลาสิ้นสุดการออกอากาศ (End-Time)
5. ลาเบลแสดงผลการคำนวณเวลารวมที่ใช้ ออกอากาศ (Total-Time)

3.2.5.2.4 ส่วนบันทึกผังรายการลงไฟล์

1. ปุ่มบันทึก (Save Button)
2. ปุ่มเคลียร์ (Clear Button)
3. ปุ่มยกเลิก (Cancel Button)

3.2.5.3 หน้าต่างสำหรับแก้ไขผังรายการล่วงหน้า

หน้าต่างสำหรับแก้ไขผังรายการล่วงหน้ามีรายละเอียดเหมือนกับหน้าต่างสำหรับสร้างผังรายการล่วงหน้าทุกประการ แบ่งเป็นส่วนหลักดังนี้

1. ส่วนแสดงรายชื่อแหล่งข้อมูลเสียง
2. ส่วนแสดงรายการที่สร้าง
3. ส่วนกำหนดการออกอากาศ
4. ส่วนบันทึกผังรายการลงไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5.3.1 ส่วนแสดงรายชื่อแหล่งข้อมูลเสียง

1. ลิสต์สำหรับเลือกแหล่งข้อมูลเสียงเพลง (Song List) แสดง
 - 1) ชื่อมีเดีย (Media Name)
 - 2) ความยาวของมีเดีย (Media Duration)
2. ลิสต์สำหรับเลือกแหล่งข้อมูลเสียงโฆษณา (Ads List) แสดง
 - 1) ชื่อมีเดีย (Media Name)
 - 2) ความยาวของมีเดีย (Media Duration)

3.2.5.3.2 ส่วนแสดงรายการที่สร้าง

1. ลิสต์แสดงรายการของข้อมูลเสียงที่เลือก (Advanced-Program) แสดง
 - 1) ชื่อมีเดีย (Media Name)
 - 2) ความยาวของมีเดีย (Media Duration)
2. ปุ่มอินเสิร์ทพอส (Insert Pause Button)
3. ปุ่มรีมูฟ (Remove Button)
4. ปุ่มเคลียร์ (Clear Button)

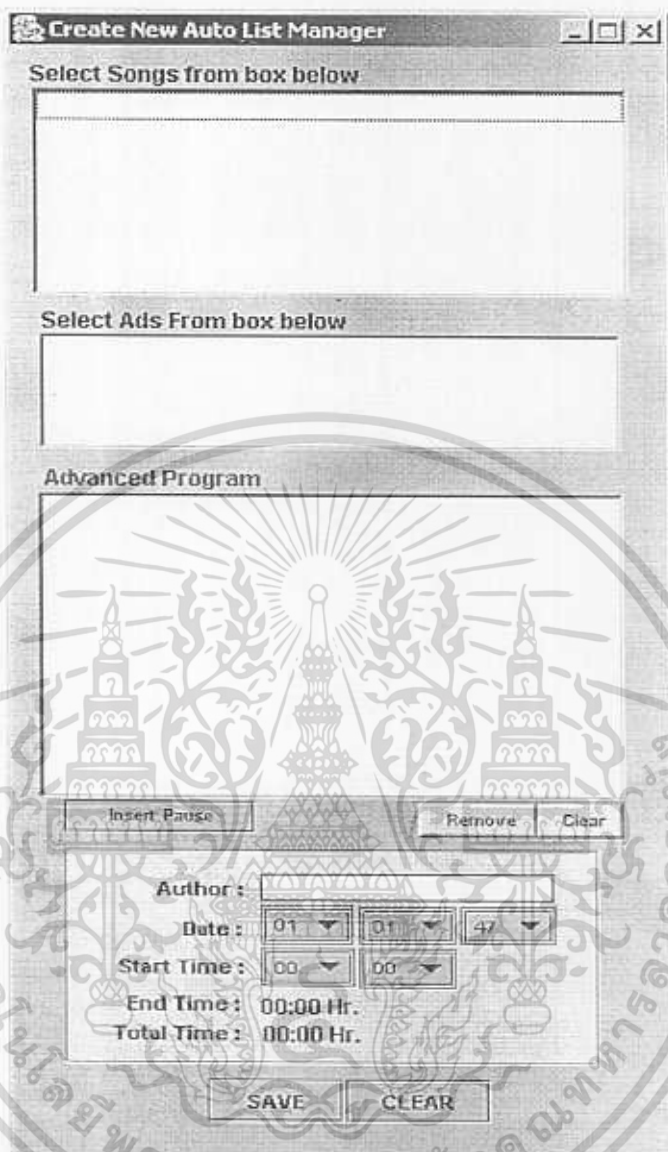
3.2.5.3.3 ส่วนกำหนดการออกอากาศ

1. กล่องข้อความสำหรับกรอกชื่อผู้สร้างลิสต์ (Author)
2. คอมโบบ็อกซ์สำหรับเลือกวันที่ออกอากาศ (Date)
3. คอมโบบ็อกซ์สำหรับเลือกเวลาเริ่มออกอากาศ (Start Time)
4. ลابلแสดงผลการคำนวณเวลาสิ้นสุดการออกอากาศ (End Time)
5. ลابلแสดงผลการคำนวณเวลารวมที่ใช้ออกอากาศ (Total Time)

3.2.5.3.4 ส่วนบันทึกผังรายการลงไฟล์

1. ปุ่มบันทึก (Save Button)
2. ปุ่มเคลียร์ (Clear Button)
3. ปุ่มยกเลิก (Cancel Button)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 หน้าต่างสำหรับแก้ไขผังรายการล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ทดลองเรียกแอปพลิเคชัน

เมื่อทำการเรียกแอปพลิเคชัน จะปรากฏหน้าต่างหลัก (Internet Radio Manager Window) ขึ้นดังรูปที่ 4.1



รูปที่ 4.1 หน้าต่างหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างหลักจะแสดงสถานะการทำงานจะแสดงข้อมูลที่เป็นอยู่จริง ณ เวลาใด ๆ ตารางที่ 4.1 แสดงสถานะเริ่มต้นถูกกำหนดไว้เมื่อเริ่มต้นแอปพลิเคชัน

ข้อมูล	ค่าเริ่มต้น
1. โหมดการทำงาน (Mode)	“Manual”
2. สถานะของไมโครโฟน (Microphone)	“On”
3. มีเดียที่กำลังเล่น (Now Playing)	-
4. โหมดการกระจายสัญญาณเสียง	“Not On-Air”

ตารางที่ 4.1 สถานะเริ่มต้นของแอปพลิเคชัน

4.2 ทดลองสร้างผังรายการปัจจุบันด้วยโหมดการทำงานแบบควบคุมด้วยมือ

4.2.1 วิธีทดลอง

กดปุ่มบราวซ์ (Browse Button) ที่อยู่ใต้ผังรายการปัจจุบันในหน้าต่าง จากนั้นทำการเลือกมีเดียซึ่งเป็นไฟล์เสียงเพลงหรือไฟล์เสียงโฆษณาก็ได้

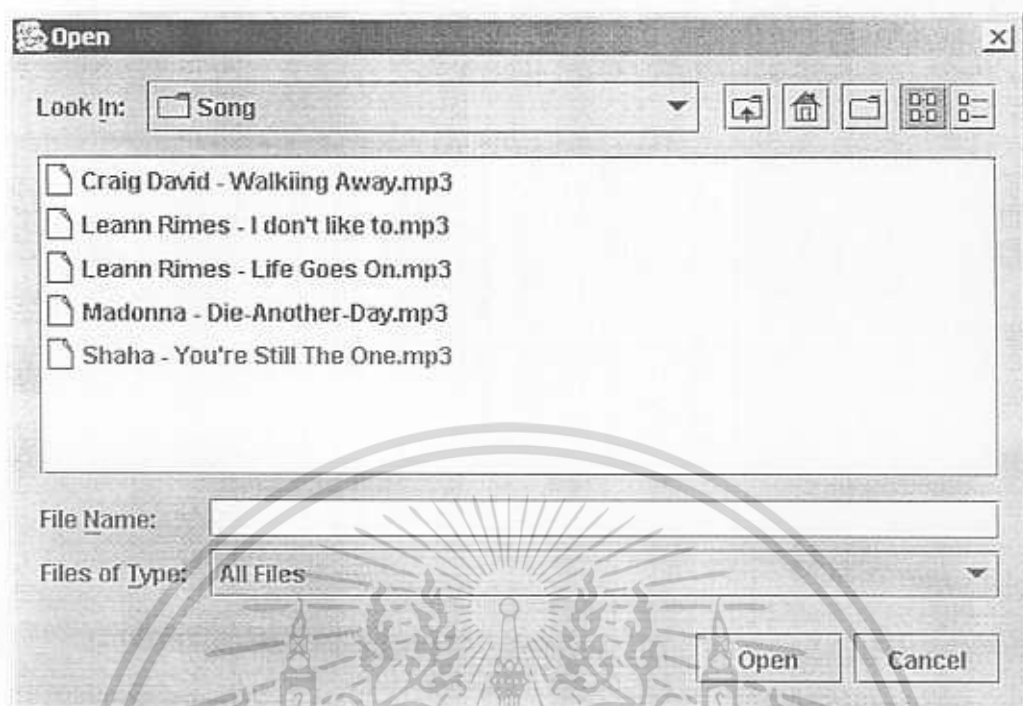


รูปที่ 4.2 วิธีทดลองเลือกแหล่งข้อมูลเสียงโดยใช้ตัวควบคุมด้วยมือ

4.2.2 ผลการทดลอง

ปรากฏหน้าต่างให้เลือกแหล่งข้อมูลเสียงดังรูปที่ 4.3 และเมื่อเลือกไฟล์เสียงที่ต้องการแล้ว รายการไฟล์เสียงที่ได้เลือกไว้จะถูกแสดงในผังรายการปัจจุบันดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 หน้าต่างเลือกแหล่งข้อมูลเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



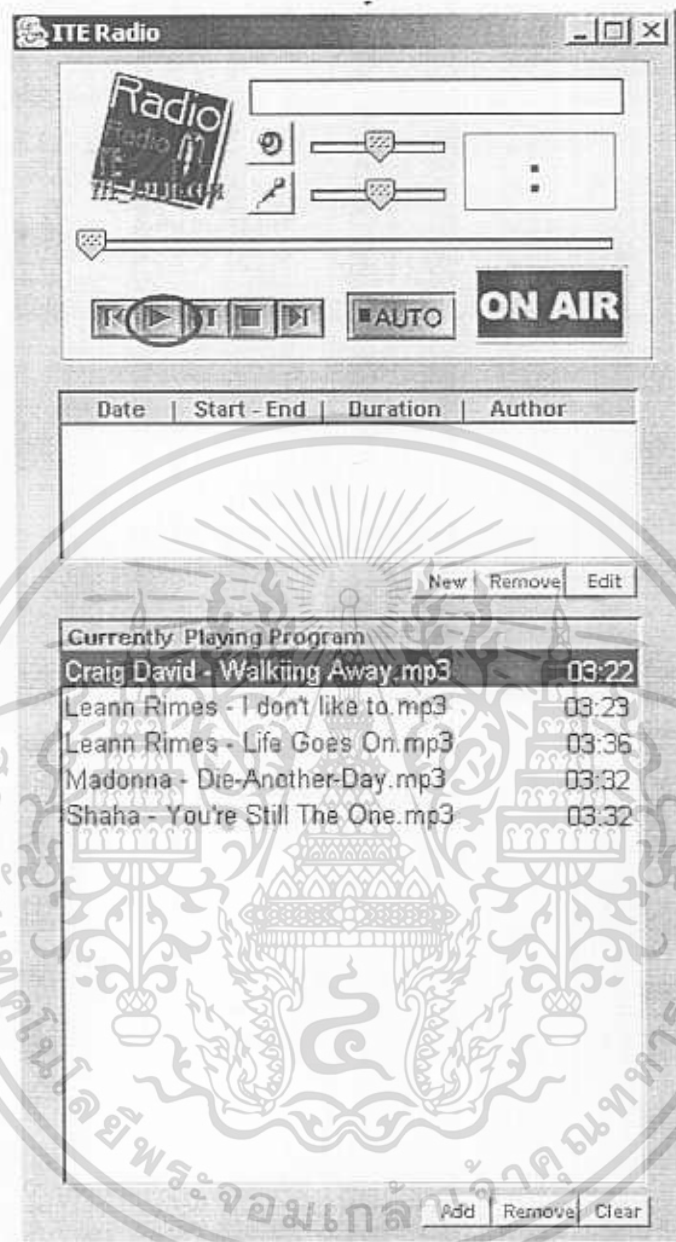
รูปที่ 4.4 หน้าต่างหลักเมื่อทำการเลือกแหล่งข้อมูลเสียงแล้ว

4.3 ทดลองเล่นมีเดีย

4.3.1 วิธีทดลอง

กดปุ่มเพลย์ (Play Button) ที่หน้าต่างควบคุมหลัก หรือดับเบิลคลิกที่มีเดียในรายการที่ต้องการเล่นในผังรายการปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 หน้าต่างหลักก่อนการเล่นเสียงเพลงหรือเสียงโฆษณา

4.3.2 ผลการทดลอง

หน้าต่างหลัก จะปรากฏชื่อของเสียงเพลงหรือเสียงโฆษณาที่แถบชื่อเพลง สไลด์บาร์จะเลื่อนไปข้างหน้าตามตำแหน่งของเสียงเพลงหรือเสียงโฆษณา และส่วนของแสดงเวลาจะแสดงเวลาที่เสียงเพลงหรือเสียงโฆษณาลงไปแล้ว หน่วยเป็นนาที ดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

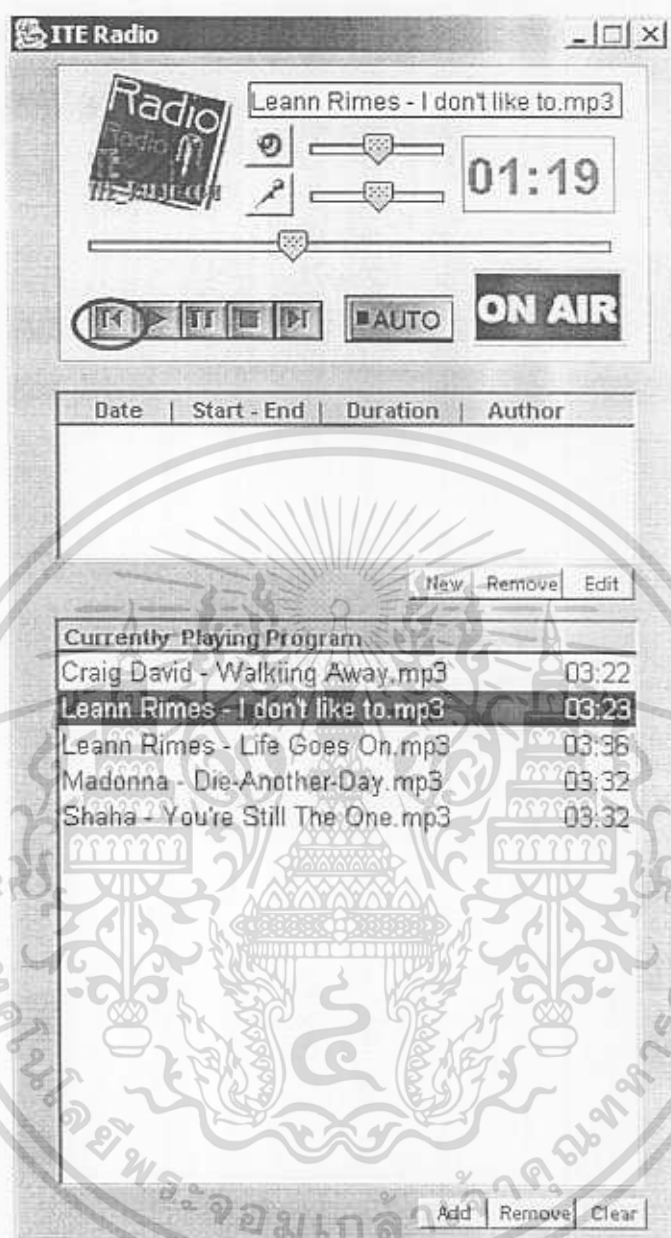


รูปที่ 4.7 หน้าต่างหลักก่อนการหยุดเล่นเสียงเพลง หรือเสียงโฆษณาชั่วคราว

4.4.2 ผลการทดลอง

จากรูปที่ 4.8 ส่วนของแสดงเวลาจะหยุดเวลาไว้ ณ ตำแหน่งที่ทำการหยุดเล่นเสียงเพลงหรือเสียงโฆษณาชั่วคราว เช่นเดียวกับสไลด์บาร์จะหยุดเลื่อนไปข้างหน้าที่ตำแหน่งที่ทำการหยุดเล่นเสียงเพลงหรือเสียงโฆษณาชั่วคราว

- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

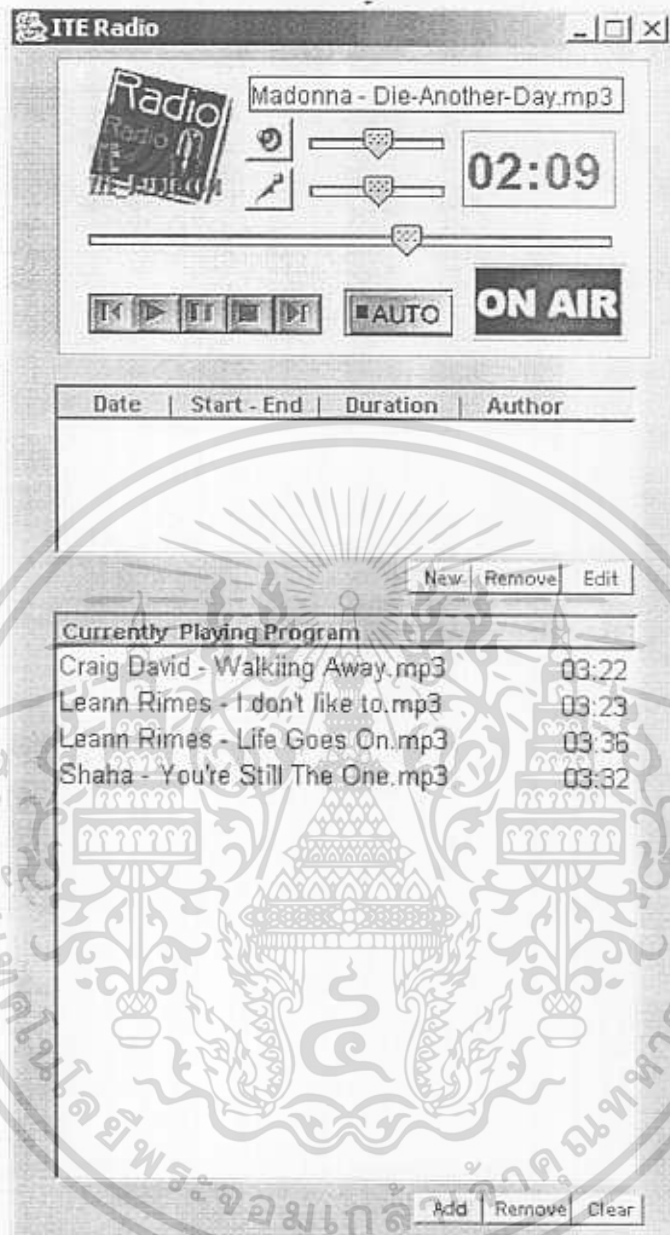


รูปที่ 4.11 หน้าต่างหลักก่อนเริ่มเล่นมีเดียก่อนหน้าในผังรายการปัจจุบัน

4.6.2 ผลการทดลอง

จากรูปที่ 4.12 โปรแกรมจะเริ่มเล่นมีเดียรายการก่อนหน้าของรายการปัจจุบันในผังรายการปัจจุบัน หากรายการปัจจุบันเป็นรายการแรกในผังรายการ โปรแกรมจะกลับไปเริ่มเล่นมีเดียในรายการสุดท้ายในผังรายการปัจจุบันแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 หน้าต่างหลักเมื่อทำการลบมีเดียที่เลือกไว้ออกจากผังรายการปัจจุบัน

4.9 ทดลองผังรายการปัจจุบัน

4.9.1 วิธีทดลอง

กดปุ่มเคลียร์ผังรายการปัจจุบัน (Clear Button) จากหน้าต่างหลักดังรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 หน้าต่างหลักก่อนทำการล้างผังรายการปัจจุบัน

4.9.2 ผลการทดลอง

รายการเสียงเพลง หรือเสียงโฆษณาทั้งหมดในลิสต์แสดงรายการมีเดียที่เลือกไว้ จะถูกลบออกทั้งหมด ดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 หน้าต่างหลักเมื่อทำการล้างผังรายการปัจจุบัน

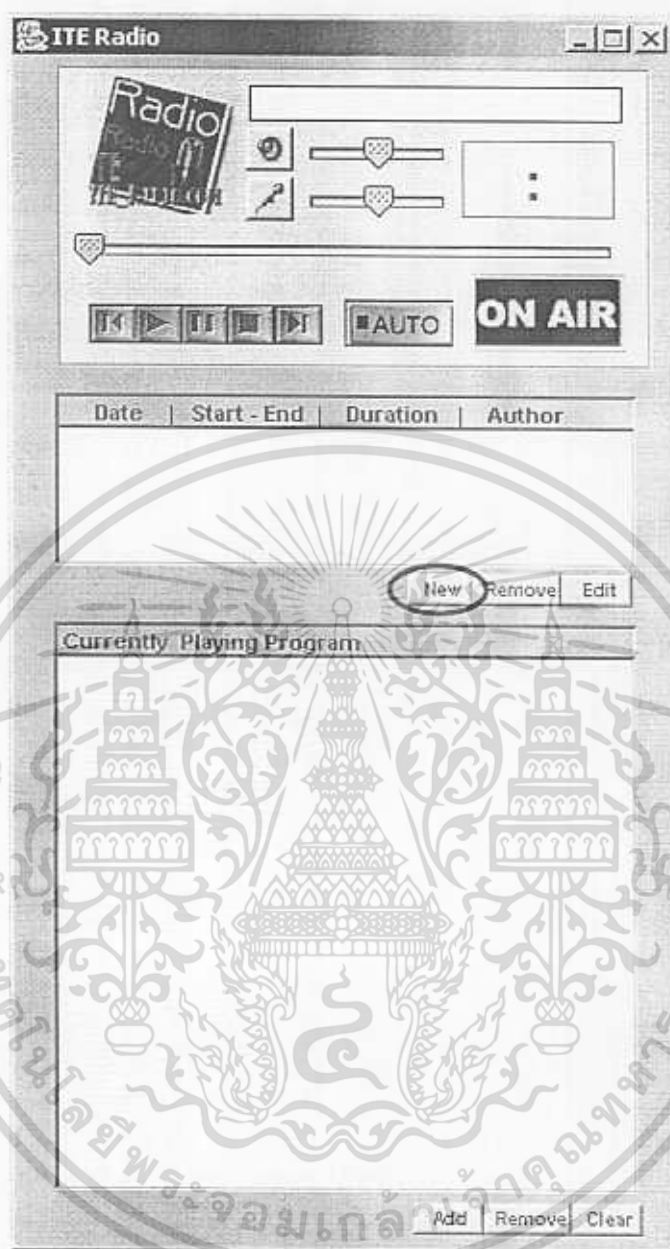
4.10 ทดลองสร้างผังรายการล่วงหน้า

4.10.1 วิธีทดลอง

กดปุ่มนิวลิสต์ (New List Button) ในส่วนคอนโทรลของหน้าต่างหลัก ดังรูปที่

4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 หน้าต่างหลักก่อนสร้างผังรายการลวงหน้า

4.10.2 ผลการทดลอง

เมื่อกดปุ่มนิวลิสท์เพื่อสร้างผังรายการลวงหน้า จะปรากฏหน้าต่างสำหรับสร้างผังรายการลวงหน้า (Create New Auto List Manager) ขึ้น มีลักษณะดังรูปที่ 4.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



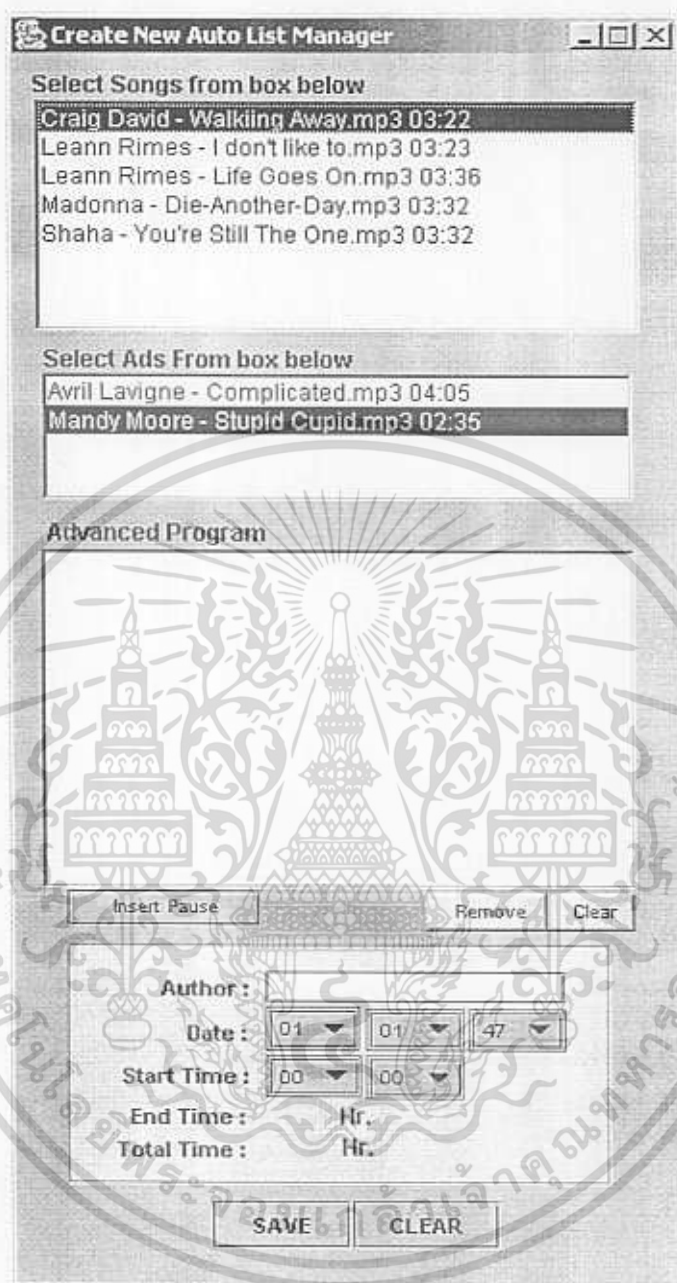
รูปที่ 4.20 หน้าต่างสำหรับสร้างผังรายการล่วงหน้า

4.11 ทดลองเพิ่มเสียงเพลงหรือเสียงโฆษณาลงในผังรายการล่วงหน้า

4.11.1 วิธีทดลอง

เลือกมีเดียจากลิสต์แสดงรายการแหล่งข้อมูลเสียง โดยการดับเบิลคลิกที่รายการมีเดียเสียงเพลง หรือดับเบิลคลิกที่รายการมีเดียเสียงโฆษณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

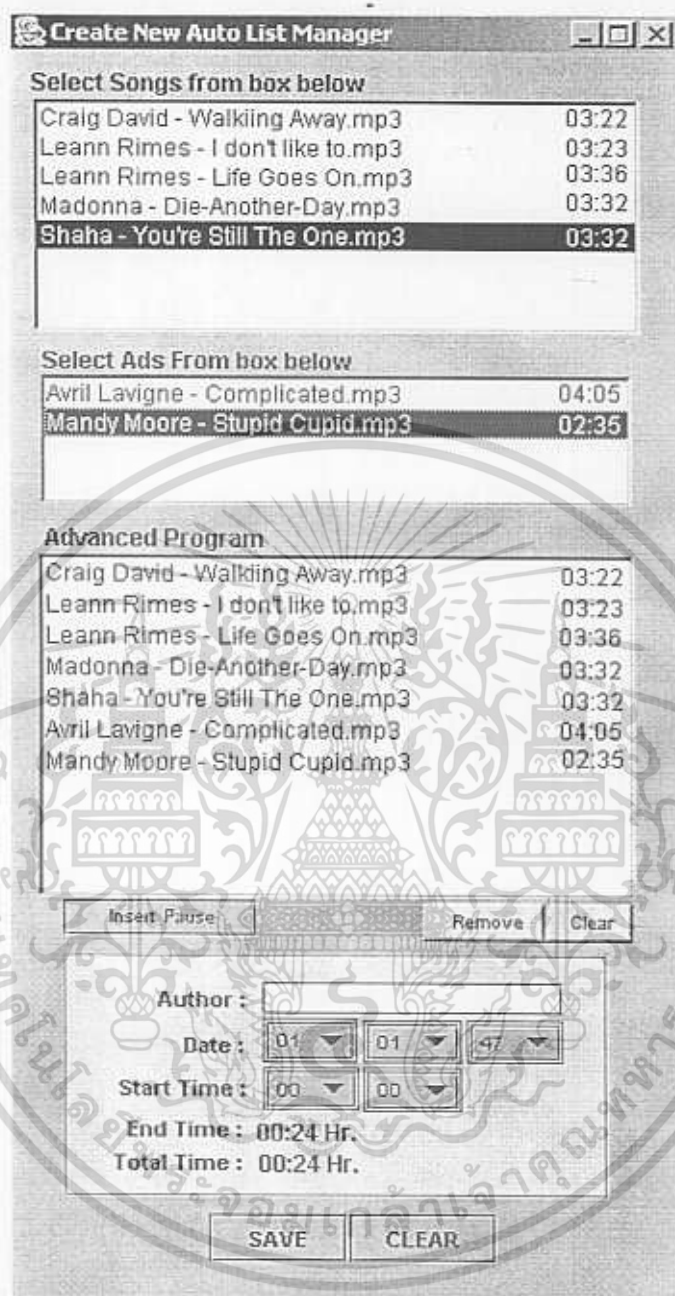


รูปที่ 4.21 วิธีเลือกแหล่งเสียงสำหรับสร้างผังรายการล่วงหน้า

4.11.2 ผลการทดลอง

หน้าต่างควบคุมแบบอัตโนมัติ ดังรูปที่ 4.22 รายการมีเดียที่ถูกเลือกจะเพิ่มเข้ามาในลิสต์รายการมีเดียที่เลือกไว้ ในส่วนแสดงเวลาสิ้นสุด (End Time) และในส่วนแสดงเวลาทั้งหมดที่ต้องใช้สำหรับผังรายการล่วงหน้า (Total Time) จะมีค่าเพิ่มขึ้น ตามเวลา มีเดีย (Media Time) ของรายการมีเดียที่เลือกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าหลังจากทำการเลือกมีเดียที่ต้องการ

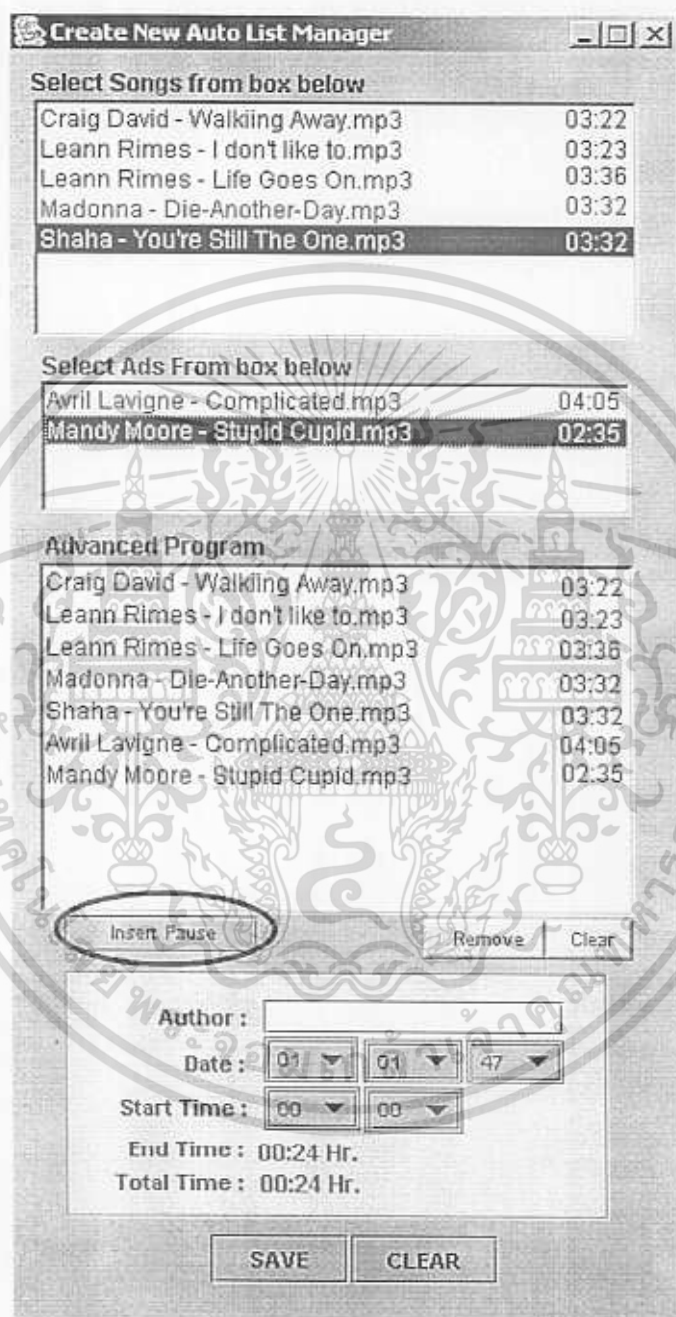
4.12 ทดลองแทรกช่วงหยุดเล่นมีเดียชั่วคราวในผังรายการล่วงหน้า

4.12.1 วิธีทดลอง

- กดปุ่มแทรกการหยุดเล่นมีเดียชั่วคราว (Insert Pause) ในหน้าต่างควบคุมแบบอัตโนมัติจะปรากฏไดอะล็อกบ็อกซ์อินพุต (Input Dialog Box) ดังรูปที่ 4.24

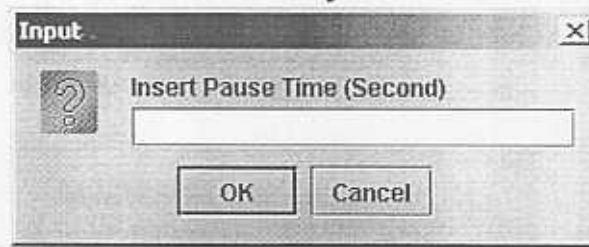
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ป้อนช่วงเวลาที่ต้องการแทรกการหยุดเล่นมีเดียชั่วคราว หน่วยเป็นวินาที ลงใน โดอะด็อกบ็อกซ์อินพุต แล้วกดปุ่มตกลง (OK Button)

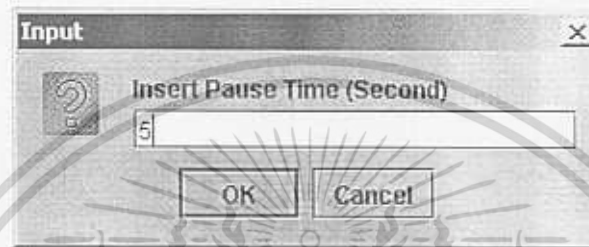


รูปที่ 4.23 วิธีแทรกช่วงเวลาวางในผังรายการล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



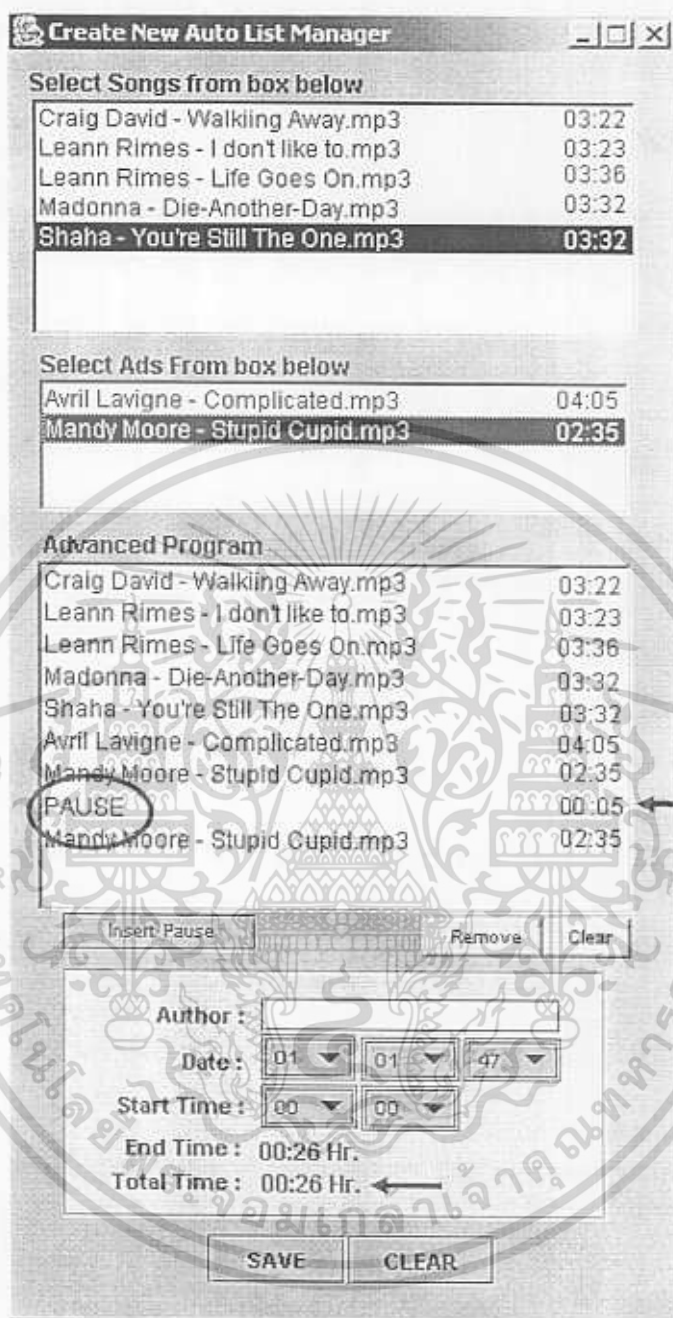
รูปที่ 4.24 โคอะล็อกซ์อินพุต (Input Dialog Box) สำหรับรับค่าช่วงเวลาวางหน่วยเป็นวินาที



รูปที่ 4.25 ทดลองป้อนค่าช่วงเวลาวางในโคอะล็อกซ์อินพุต

4.12.2 ผลการทดลอง

เมื่อกดปุ่มตกลง หลังจากใส่เวลาที่ต้องการแทรกการหยุดเล่นมีเดียชั่วคราวลงในโคอะล็อกซ์อินพุต หน้าต่างควบคุมแบบอัตโนมัติจะแทรกเวลาสำหรับการหยุดเล่นมีเดียชั่วคราวลงในลิสต์แสดงรายการมีเดียที่เลือกไว้ ถ้ากดปุ่มยกเลิกของโคอะล็อกซ์อินพุต ที่หน้าต่างควบคุมแบบอัตโนมัติจะไม่เกิดการเปลี่ยนแปลงใดๆ



รูปที่ 4.26 หน้าต่างควบคุมแบบอัตโนมัติ เมื่อทำการแทรกการหยุดเล่นมีเดียชั่วคราว

4.13 ทดลองลบมีเดียออกจากผังรายการ

4.13.1 วิธีทดลอง

เลือกรายการมีเดียที่ต้องการลบออกจากผังรายการล่วงหน้า จากนั้นคลิกปุ่มรีเฟรช ดัง

รูปที่ 4.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

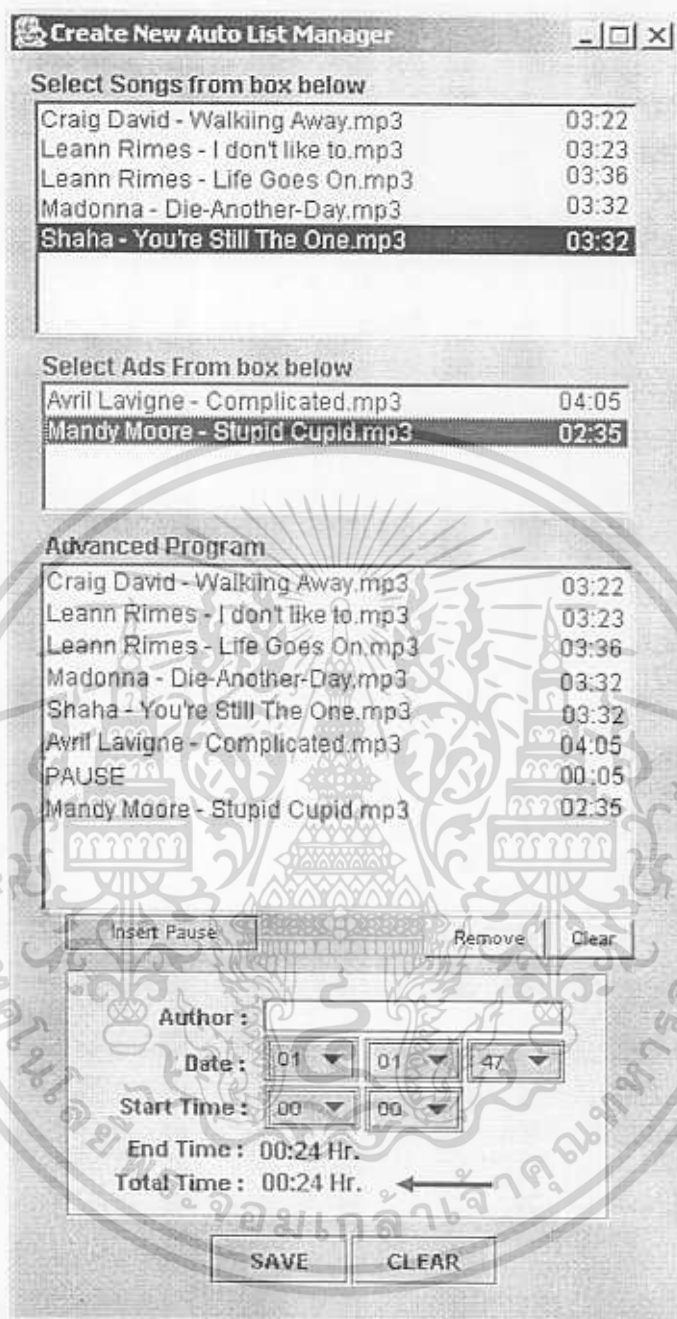


รูปที่ 4.27 วิธีลบมีเดียออกจากผังรายการล่วงหน้า

4.13.2 ผลการทดลอง

รายการมีเดียที่ถูกเลือก จะถูกลบออกจากผังรายการล่วงหน้าที่กำลังสร้างอยู่ ใน ส่วนแสดงเวลาสิ้นสุดของผังรายการ และส่วนแสดงเวลารวมของผังรายการจะลดลง ตาม เวลาของมีเดียที่ถูกลบไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



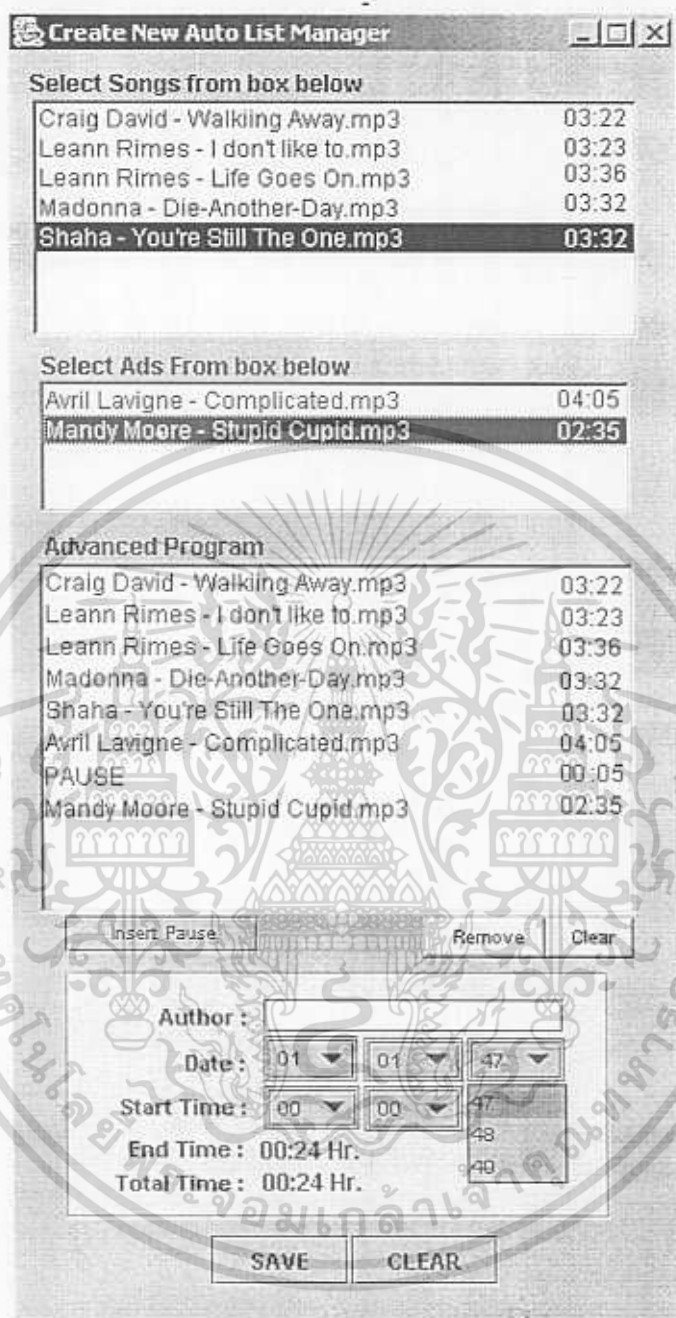
รูปที่ 4.28 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าหลังจากทำการลบมีเดียออกจากผังรายการ

4.14 ทดลองการตั้งค่าวันที่ และเวลาของผังรายการล่วงหน้า

4.14.1 วิธีทดลอง

เลือกค่าวันที่ และเวลาเริ่มต้นของผังรายการล่วงหน้า จากดริอปดาวน์เมนู (Drop down Menu) ในส่วนกำหนดเวลาออกอากาศ ดังรูปที่ 4.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

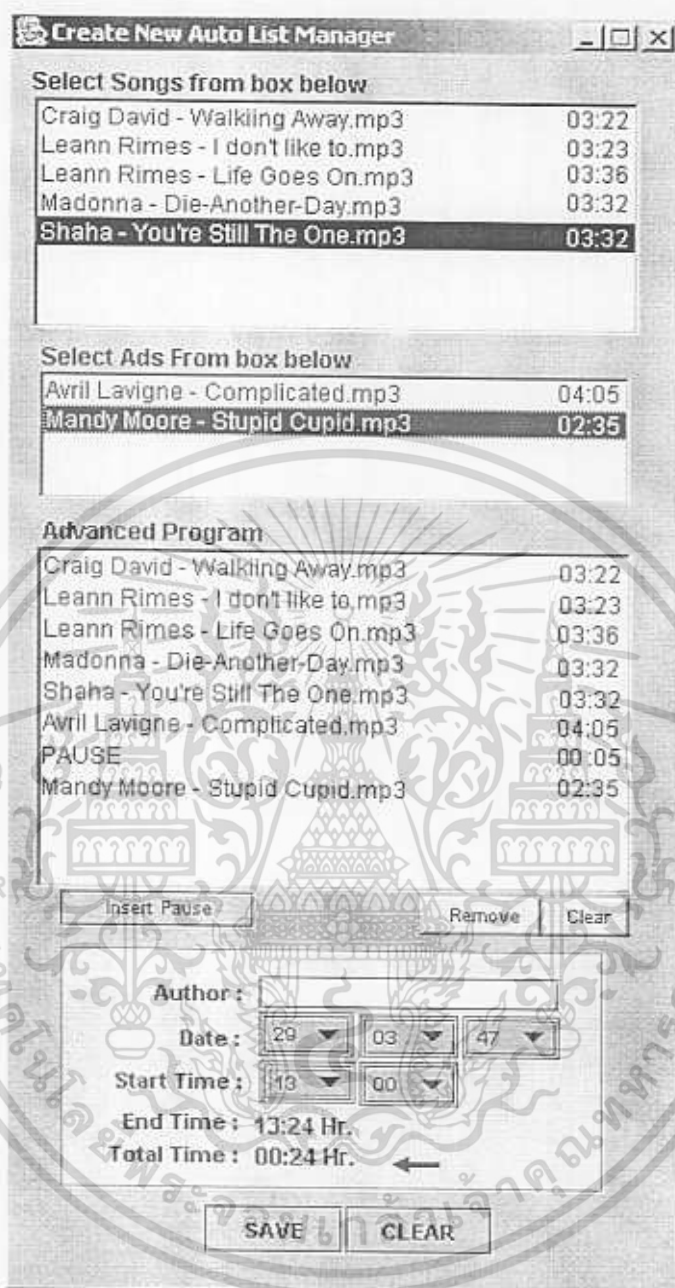


รูปที่ 4.29 วิธีการตั้งค่าวันที่และเวลาเริ่มต้นของผังรายการล่วงหน้า

4.14.2 ผลการทดลอง

ในส่วนของเวลาสิ้นสุดของผังรายการ (End Time) จะเปลี่ยนไปเป็นค่าเวลาที่ตั้งไว้รวมกับเวลาทั้งหมดที่ใช้เล่นเสียงเพลงหรือเสียงโฆษณา ที่เลือกไว้ในลิสต์แสดงรายการ มีเดียที่เลือกไว้ ดังรูปที่ 4.30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าเมื่อทำการตั้งค่าวันที่
และเวลาสำหรับผังรายการล่วงหน้า

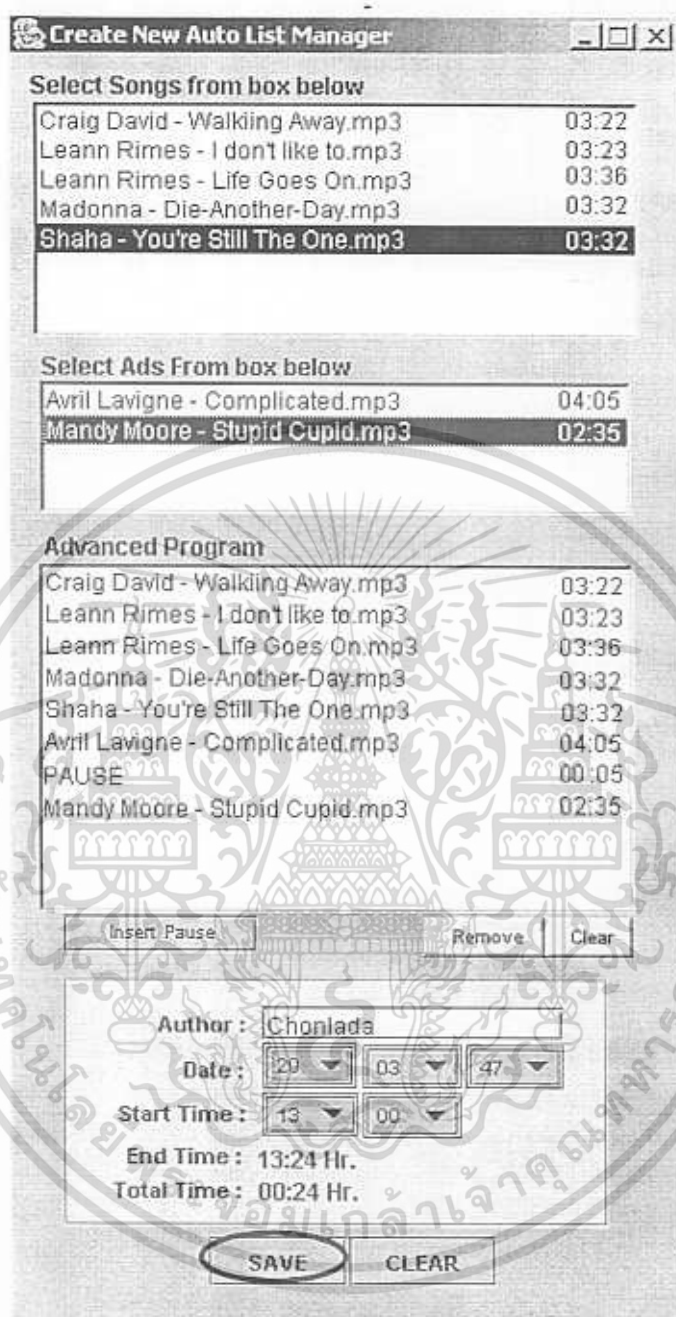
4.15 ทดลองบันทึกการสร้างผังรายการล่วงหน้า

4.15.1 วิธีทดลอง

ทดลองสร้างผังรายการล่วงหน้าโดยกำหนดค่าต่าง ๆ ให้ครบถ้วน จากนั้นกดปุ่ม

เซฟ (Save Button) บนหน้าต่างสำหรับสร้างผังรายการล่วงหน้า ดังรูปที่ 4.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.31 วิธีการบันทึกการสร้างผังรายการล่วงหน้า

4.15.2 ผลการทดลอง

มีได้ 3 กรณี คือ

1. ไม่มีรายการของมีเดียในผังรายการที่จะบันทึก
2. การตั้งค่าวันที่และเวลาถูกต้อง คือ วันที่ต้องเป็นวันที่สร้างผังรายการ หรือ วันที่หลังจากวันที่สร้างผังรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การตั้งค่าวันที่และเวลาผิดพลาด คือ วันที่เป็นวันก่อนวันจริงขณะสร้างผังรายการ หรือ วันที่เป็นวันจริงขณะสร้างผังรายการแต่เวลาเป็นเวลาก่อนเวลาจริงขณะสร้างผังรายการ

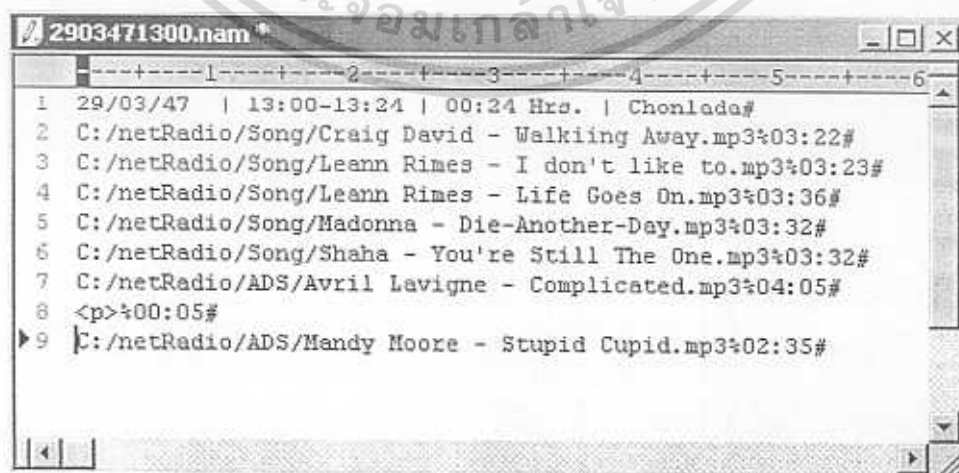
4.15.2.1 ผลการทดลอง กรณีแรก ไม่มีรายการของมีเดียในผังรายการที่จะบันทึก ระบบจะแสดงไดอะล็อกบ็อกซ์เตือน (ITE Radio Alert Dialog Box) ว่าในผังรายการล่วงหน้าที่เราสร้างไว้ ไม่มีรายการเสียงเพลง หรือเสียงโฆษณา ที่เลือกไว้ ดังรูปที่ 4.32



รูปที่ 4.32 ไดอะล็อกบ็อกซ์เตือนกรณีไม่มีรายการเสียงในผังรายการที่สร้าง

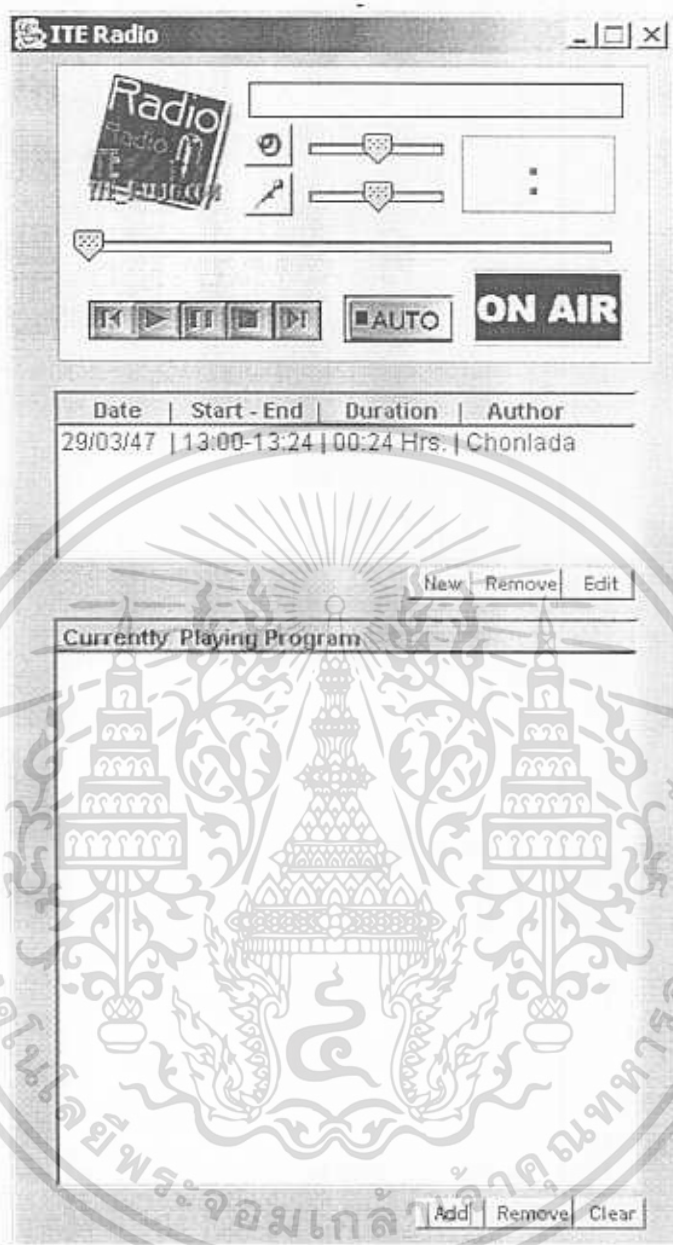
4.15.2.2 ผลการทดลอง กรณีที่สอง การตั้งค่าวันที่ และเวลาถูกต้อง

1. รายละเอียดของการสร้างผังรายการล่วงหน้าจะถูกบันทึกอยู่ในรูปแบบไฟล์นามสกุล เอ็นเอเอ็ม (.nam) มีรายละเอียด ดังรูปที่ 4.33
2. หน้าต่างหลัก รายละเอียดของผังรายการล่วงหน้าที่เราเพิ่งสร้างเสร็จจะถูกเพิ่มลงในลิสต์แสดงรายการผังรายการล่วงหน้า (Auto Playlist)



รูปที่ 4.33 ไฟล์แสดงรายละเอียดของผังรายการล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

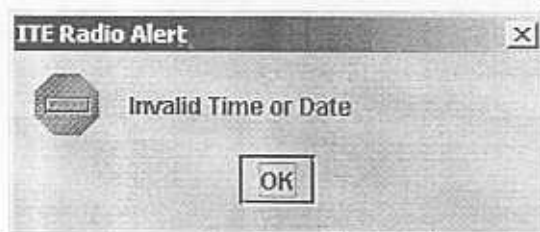


รูปที่ 4.34 ข้อมูลของผังรายการล่วงหน้าในหน้าต่างหลักเพิ่มขึ้นหลังจากทำการบันทึกผังรายการ

4.15.2.3 ผลการทดลอง กรณีที่สาม การตั้งค่าวันที่ และเวลาผิดพลาด

ระบบจะแสดง ไดอะล็อกบ็อกซ์เตือน (ITE Radio Alert Dialog Box) ว่า วันที่ หรือ เวลาที่ตั้งไว้ผิดพลาด ให้ทำการตั้งค่าวันที่ และเวลาใหม่อีกครั้ง ดังรูปที่ 4.35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.35 โดอะล๊อคบ็อกซ์เตือนการตั้งค่าวันที่และเวลาผิดพลาด

4.16 ทดลองแก้ไขผังรายการล่วงหน้า

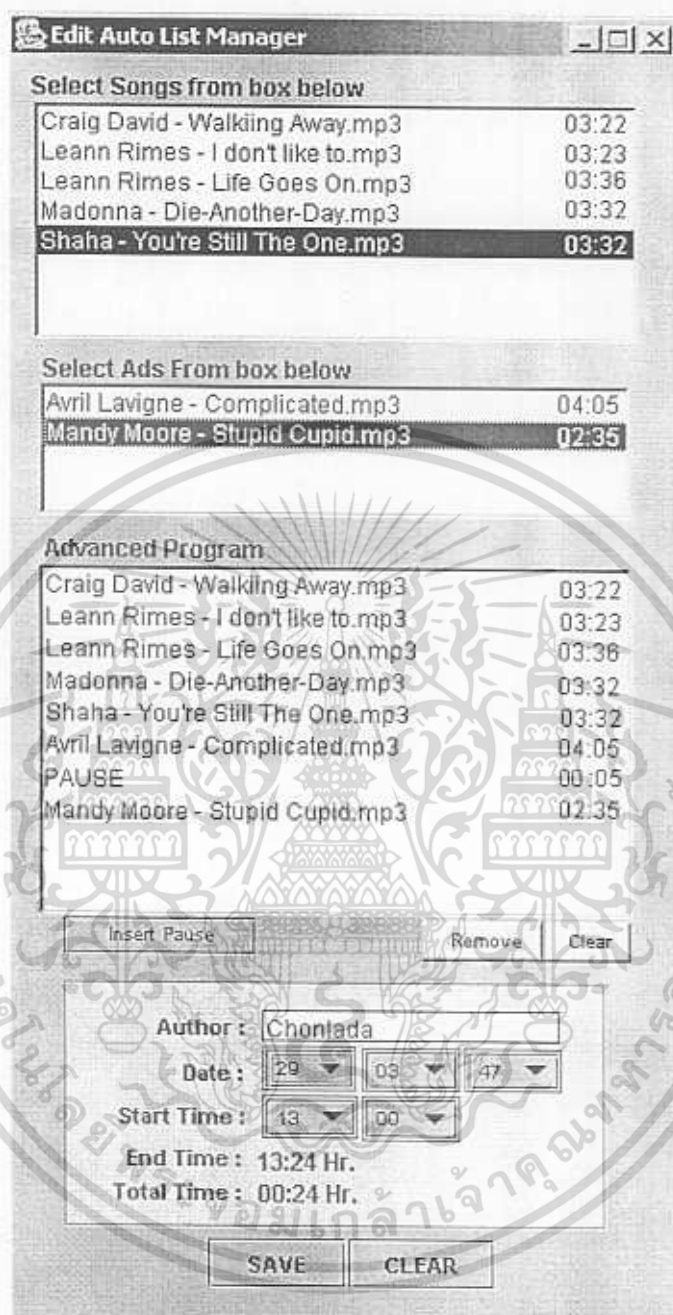
4.16.1 วิธีทดลอง

1. เลือกผังรายการล่วงหน้าที่สร้างไว้แล้วจากลิสต์แสดงผังรายการล่วงหน้า และกดปุ่มอีดิท (Edit Button) ดังรูปที่ 4.36
2. ปรากฏหน้าต่างแก้ไขผังรายการล่วงหน้า (Edit Auto List Manager) ดังรูปที่ 4.37
3. ทดลองแก้ไขรายละเอียดของผังรายการล่วงหน้า โดยเปลี่ยนวันออกอากาศของผังรายการล่วงหน้าจากวันที่ 29 เป็น 21 เดือน 03 ปี 2547 และลบข้อมูลมีเดียรายการสุดท้ายในผังรายการล่วงหน้าออกไปหนึ่งรายการ
4. กดปุ่มบันทึก (Save Button)



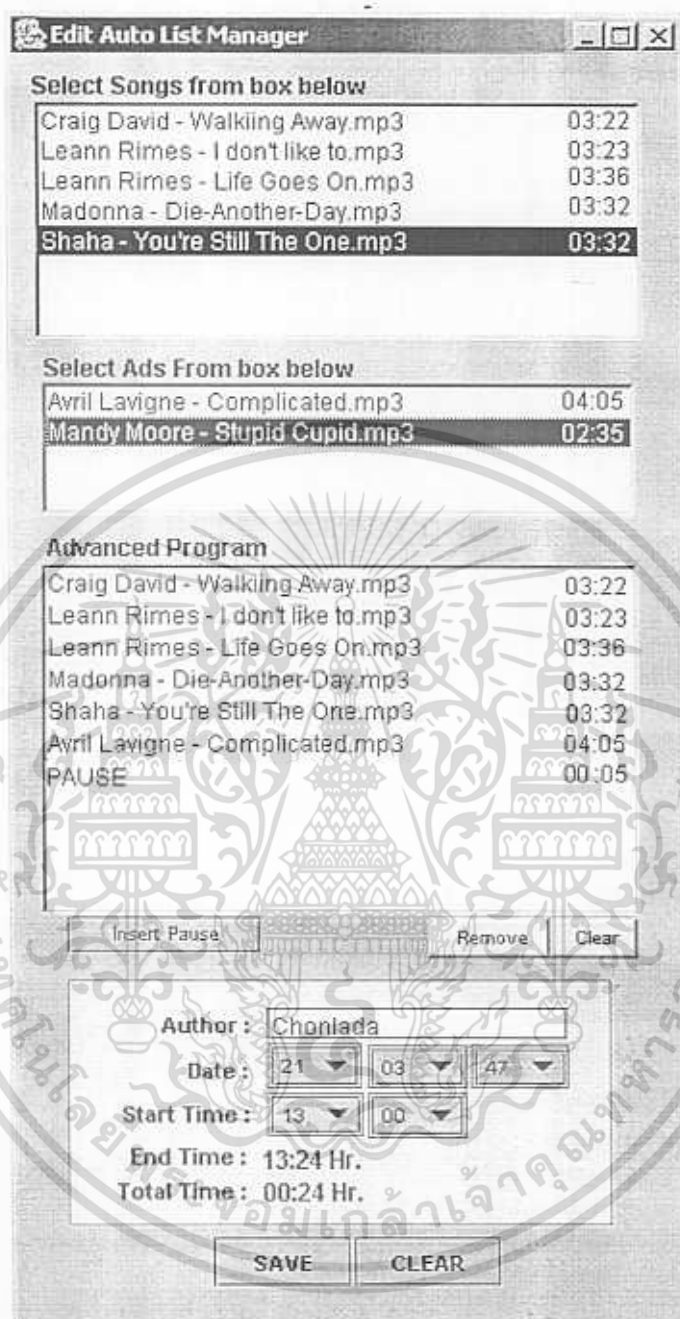
รูปที่ 4.36 วิธีการแก้ไขผังรายการลงหน้าที่สร้างไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.37 หน้าต่างแก้ไขผังรายการล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

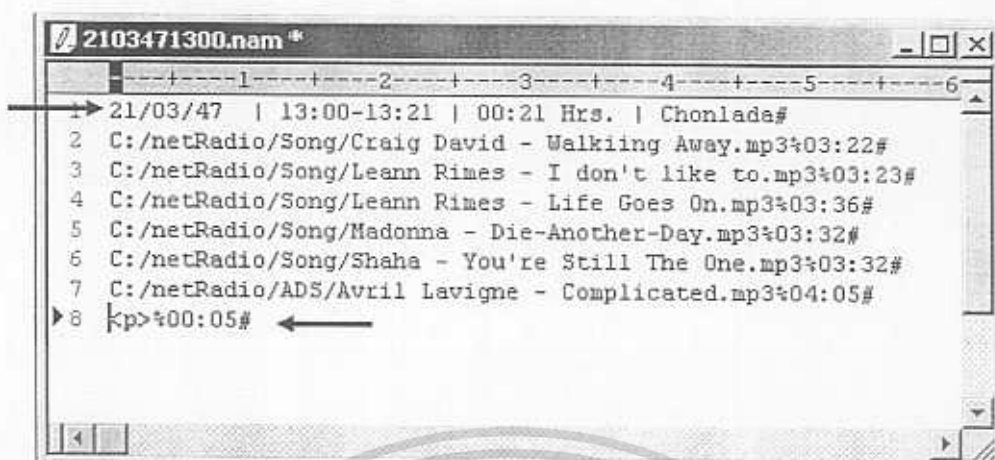


รูปที่ 4.38 ทดลองแก้ไขผังรายการล่วงหน้า

4.16.2 ผลการทดลอง

1. รายละเอียดของไฟล์นามสกุลเอ็มเอ็ม จะถูกเปลี่ยนแปลงไปจากเดิมจากรูปที่ 4.33 เป็นดังรูปที่ 4.39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.39 ไฟล์นามสกุลเอ็นเอ็มเอ็มหลังการแก้ไขผังรายการล่วงหน้า

- หน้าต่างหลัก ในส่วนแสดงรายการผังรายการล่วงหน้า จะทำการอัปเดตข้อมูลผังรายการที่แก้ไขแล้ว นำมาแสดงผล ดังรูป 4.40



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



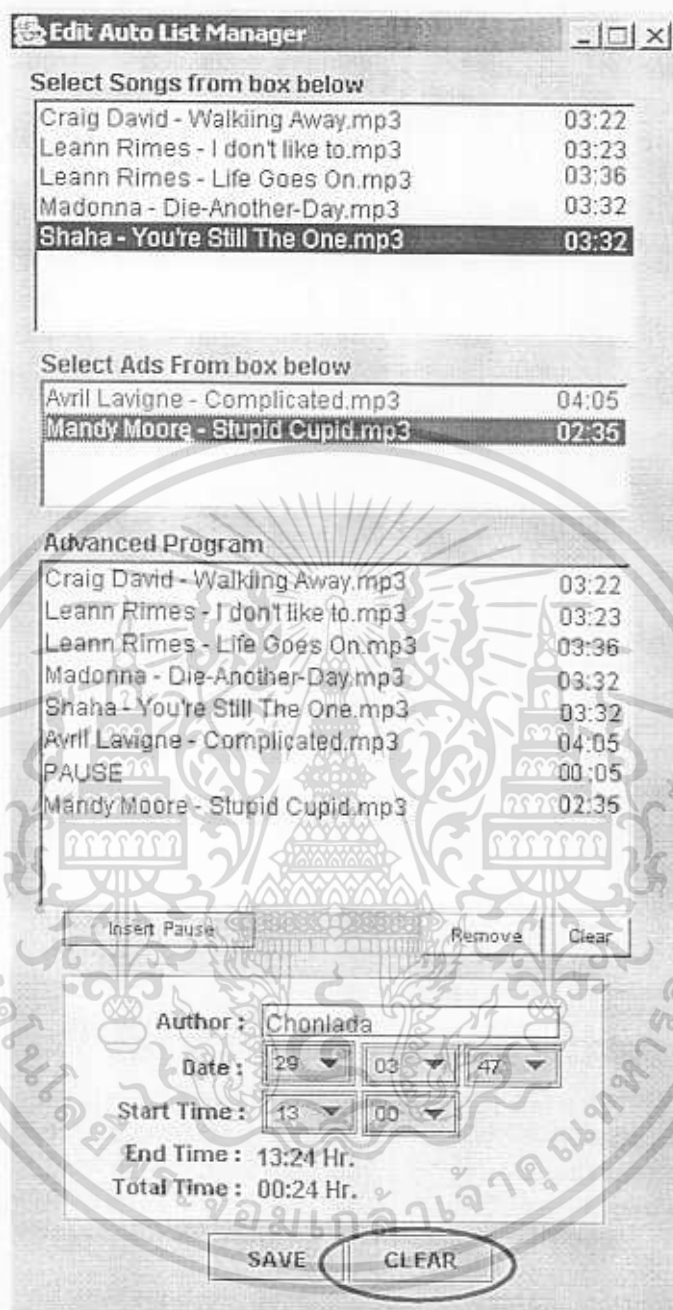
รูปที่ 4.40 หน้าต่างหลักเมื่อทำการแก้ไขผังรายการล่วงหน้า

4.17 ทดลองการดั่งผังรายการล่วงหน้าที่เราสร้างไว้

4.17.1 วิธีทดลอง

1. หน้าต่างควบคุมแบบอัตโนมัติ กดปุ่มเคลียร์ที่อยู่ล่างสุดของหน้าต่าง

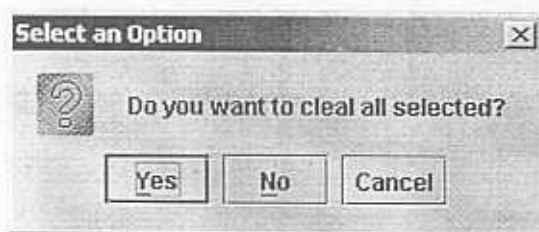
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าก่อนทำการล้างผังรายการที่สร้างไว้

- ปรากฏไดอะล็อกบ็อกซ์ยืนยัน (ITE Radio Alert Confirm Dialog Box) ถามว่าต้องการยืนยันการเคลียร์รายละเอียดผังรายการล่วงหน้า หรือไม่ ดังรูปที่ 4.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.42 ใคอะลื้อคบื้อกซ์ขึ้นชั้นการเคลียร์รายละเอียดผังรายการล่วงหน้า

4.17.2 ผลการทดลอง

1. เมื่อกดปุ่มเขส (Yes Button) จากใคอะลื้อคบื้อกซ์เพื่อยืนยันการล้างผังรายการที่สร้างไว้ รายการของมีเคือที่เลือกไว้ในผังรายการล่วงหน้า จะถูกลบออกทั้งหมด รวมถึงชื่อผู้สร้างผังรายการล่วงหน้า วันที่และเวลาที่ถูกลือกไว้จะกลับเป็นค่าเริ่มต้น ดังรูปที่ 4.43





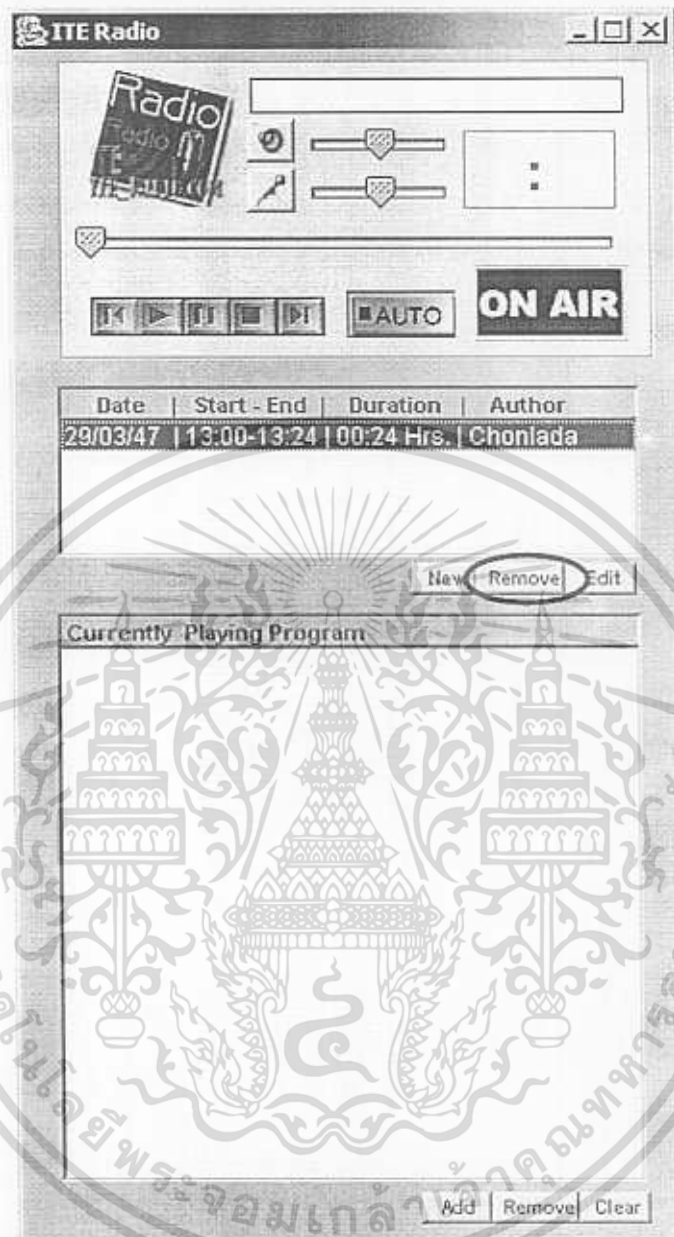
รูปที่ 4.43 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าเมื่อทำการล้างผังรายการที่สร้างไว้

4.18 ทดลองลบผังรายการล่วงหน้า

4.18.1 วิธีทดลอง

1. เลือกผังรายการล่วงหน้าจากลิสต์แสดงรายการผังรายการล่วงหน้า (Auto Playlist) กดปุ่มลบล้าง (Delete Button)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.44 หน้าต่างหลัก ก่อนการลบผังรายการล่วงหน้า

4.18.2 ผลการทดสอบ

1. หน้าต่างหลัก รายการผังรายการล่วงหน้าที่ต้องการลบ จะถูกลบออกจากลิสต์ แสดงรายการผังรายการล่วงหน้า ดังรูปที่ 4.45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



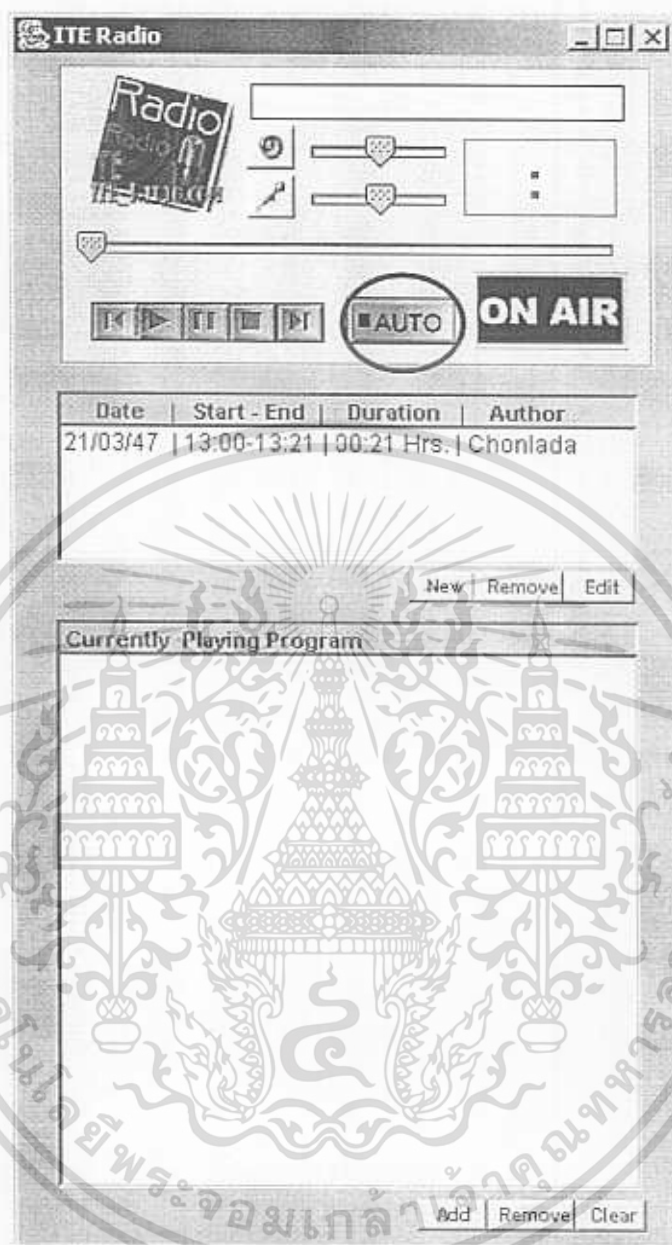
รูปที่ 4.45 หน้าต่างหลักเมื่อทำการลบผังรายการล่วงหน้า

4.19 ทดลองสลับการทำงานเป็นโหมดอัตโนมัติ

4.19.1 วิธีทดลอง

กดปุ่มออโต้ (Auto Button) ที่ส่วนควบคุมของหน้าต่างหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.46 หน้าต่างหลัก ก่อนการสลับเป็น โหมดอัตโนมัติ

4.19.2 ผลการทดลอง

1. ปุ่มออโต้บนหน้าต่างหลัก รูปสี่เหลี่ยมสีดำบนปุ่มจะเปลี่ยนเป็นสีแดง แสดงถึงสถานะการทำงานเป็นโหมดควบคุมอัตโนมัติ (Automatically Playing Mode) ดังรูปที่ 4.47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.47 หน้าต่างหลักหลังจากสลับการทำงานเป็นโหมดอัตโนมัติ

2. เมื่อถึงเวลาตามเวลาเริ่มต้นของผังรายการล่วงหน้า ระบบจะทำการเรียกผังรายการล่วงหน้าขึ้นมา และแสดงรายการเสียงเพลง หรือเสียงโฆษณา ลงในลิสต์แสดงรายการมีเดียที่เลือกไว้ (Currently Playing Program List) และทำการเล่นเสียงเพลง หรือเสียงโฆษณา ตามลำดับของผังรายการล่วงหน้าโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



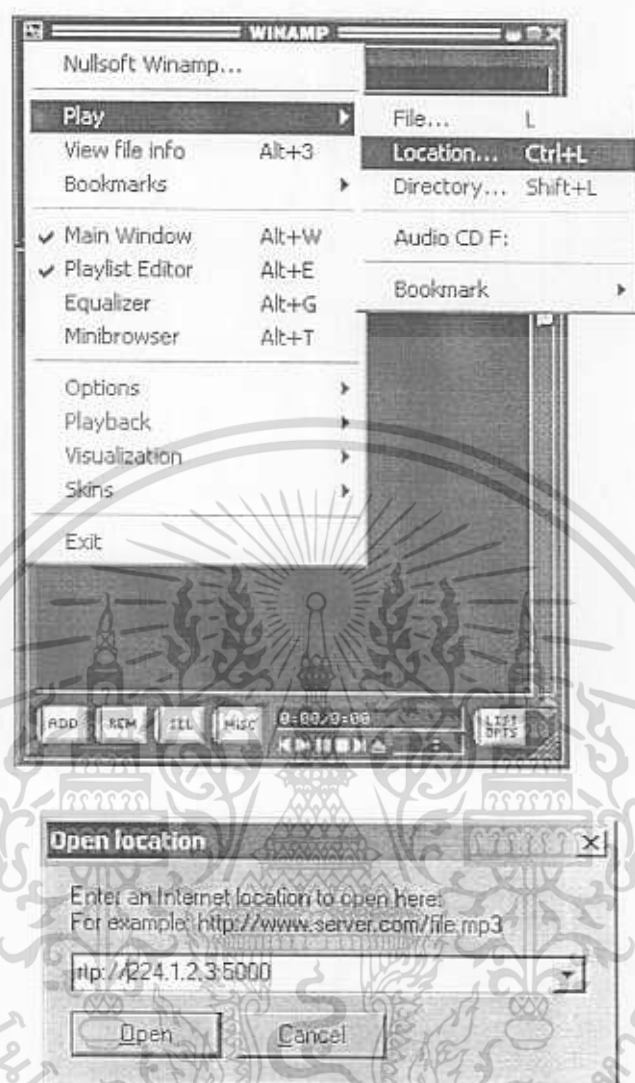
รูปที่ 4.48 หน้าต่างหลักเมื่อทำงานในโหมดอัตโนมัติตามเวลาเริ่มต้นของผังรายการ

4.20 ทดลองส่งข้อมูลเสียงไปยังโคลเอนท์

4.20.1 วิธีทดลอง

1. กดปุ่มออนแอร์ที่หน้าต่างควบคุมหลัก
2. เปิดโปรแกรมวินแอมป์ (Winamp) ดัดต่อไปยังมัลติคาสท์ไอทีแอดเดรส ผ่านโปรโตคอลอาร์ทีพี มีรูปแบบคอนเนคชั่นสตริงคือ "rtsp://224.1.2.3:5000"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.50 วิธีการเชื่อมต่อของ โคล์เอินท์โปรแกรมมายังระบบ

4.20.2 ผลการทดลอง

จากรูปที่ 4.51 รูปไอคอนแอร์ (ON AIR) ของปุ่มออนแอร์ที่หน้าต่างหลักเปลี่ยนเป็น สีสว่างขึ้น แสดงถึงสถานะออนแอร์ จากรูปที่ 4.52 ก. โคล์เอินท์โปรแกรมแสดงสถานะ กำลังเชื่อมต่อไปยังเซิร์ฟเวอร์ และรูปที่ 4.52 ข. โคล์เอินท์โปรแกรมเมื่อเริ่มการรับสตรีม เสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.52 ก. โปรแกรมไคลเอนท์เมื่อเริ่มทำการเชื่อมต่อไปยังเซิร์ฟเวอร์



รูปที่ 4.52 ข. โปรแกรมไคลเอนท์เมื่อเริ่มการรับข้อมูลเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.21.2 ผลการทดลอง

โปรแกรมโกลเอนท์ยังคงรับสตรีมเสียงได้อยู่ และมีระดับเสียงที่เปลี่ยนไปตาม เซอร์ฟเวอร์กำหนด โดยสถานะภายนอกไม่มีการเปลี่ยนแปลง



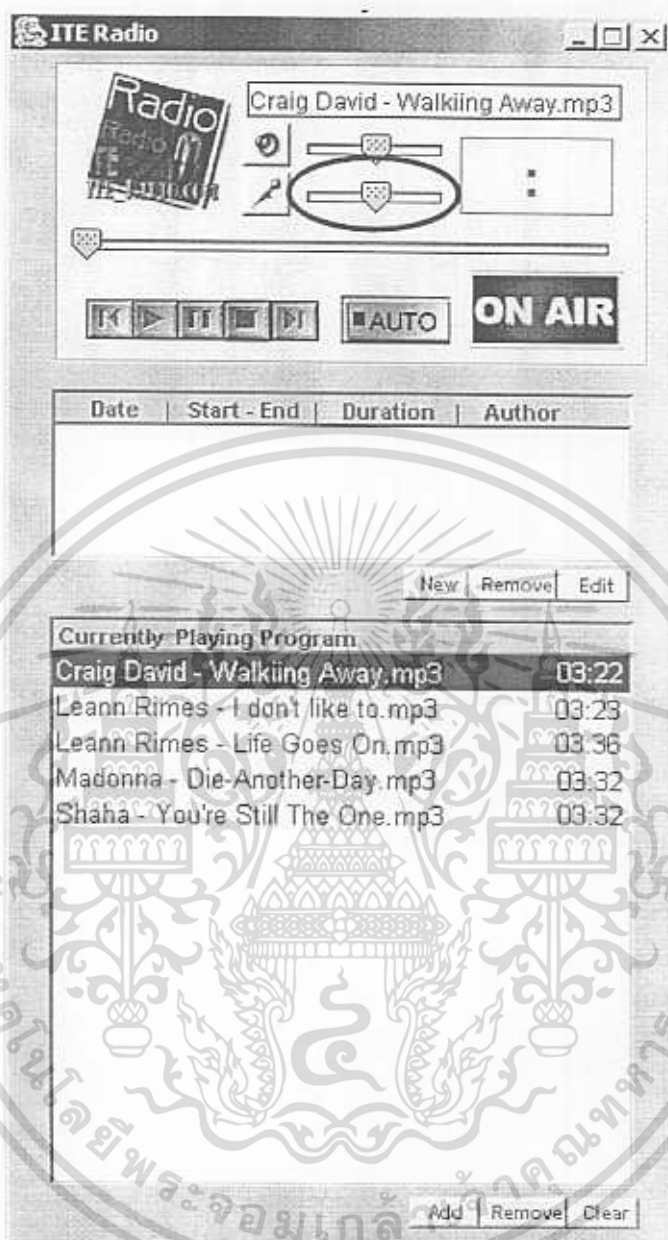
รูปที่ 4.54 โปรแกรมโกลเอนท์เมื่อทำการปรับระดับเสียงของระบบ

4.22 ทดลองปรับระดับเสียงไมโครโฟน

4.22.1 วิธีทดลอง

เลื่อนสไลด์บาร์ของไมโครโฟน (Microphone Slide Bar) ไปข้างหน้า หากต้องการเพิ่มเสียงไมโครโฟน เลื่อนย้อนกลับเมื่อต้องการลดความดังเสียงไมโครโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.55 วิธีปรับเสียงไมโครโฟน

4.22.2 ผลการทดลอง

โปรแกรมโคลนที่ยังคงรับสตรีมเสียงได้อยู่ และมีระดับเสียงไมโครโฟนที่เปลี่ยนไปตามเซิร์ฟเวอร์กำหนด โดยสถานะภายนอกไม่มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.56 โปรแกรมไลต์เอนท์เมื่อทำการปรับระดับเสียงไมโครโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

5.1.1 ลักษณะของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ตโดยรวม

5.1.1.1 ส่วนจัดการผังรายการ (Program Management Module)

ทำหน้าที่จัดทำผังรายการปัจจุบันที่กำลังออกอากาศ และสร้างผังรายการล่วงหน้าสำหรับออกอากาศอัตโนมัติตามเวลาที่กำหนด สามารถเลือกโหมดการทำงานได้ว่าต้องการเลือกใช้ผังรายการล่วงหน้าหรือไม่

5.1.1.2 ส่วนมิกซ์เซอร์ (Broadcast Server)

ทำหน้าที่ผสมเสียงและควบคุมระดับความดังของเสียงจากสองแหล่ง คือ เสียงจากไมโครโฟนและเสียงเพลงจากชาวดีการ์ด การผสมเสียงก็เป็นการที่สามารถเลือกเปิดหรือปิดเสียงใดเสียงหนึ่ง และการเปิดหรือปิดทั้งสองเสียงได้

5.1.1.3 ส่วนบรรณาธิการซีอาร์เอฟเวอร์ (Broadcast Server)

ทำหน้าที่รับข้อมูลเสียงจากแหล่งเสียงที่เป็นไฟล์เช่นไฟล์เสียงเพลงหรือไฟล์เสียงโฆษณา และแหล่งเสียงที่เป็นไมโครโฟน จากนั้นทำการจัดรูปแบบของข้อมูลให้สามารถส่งแบบเรียลไทม์ผ่านโปรโตคอลอาร์ทีพีได้ โดยการนำข้อมูลเสียงที่จับ (Capture) ได้ มาสร้างเป็นแพ็คเกจอาร์ทีพี (RTP Packets) และหน้าที่สุดท้ายนำเสียงที่จับได้ส่งไปยังไคลเอนท์ที่อยู่ในมัลติคาสท์กรุปแอดเดรสผ่านเครือข่ายที่สนับสนุนมัลติคาสท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 สรุปผลการทดลองแต่ละส่วน

5.1.2.1 ส่วนจัดการผังรายการ (Program Management Module)

สามารถจัดทำผังรายการปัจจุบันที่กำลังออกอากาศ และสร้างผังรายการล่วงหน้าสำหรับออกอากาศอัตโนมัติตามเวลาที่กำหนดได้ สามารถเลือกโหมดการทำงานได้ว่าต้องการเลือกใช้ผังรายการล่วงหน้าหรือไม่ สามารถควบคุมการเล่นเสียงระหว่างการออกอากาศได้อย่างถูกต้อง

5.1.2.2 ส่วนมิลซ์เซิร์ฟเวอร์ (Broadcast Server)

สามารถทำหน้าที่ผสมเสียง และควบคุมระดับความดังของเสียงจากสองแหล่งคือ เสียงจากไมโคร โฟนและเสียงเพลงจากฮาร์ดดิสก์ ได้ตามที่ออกแบบไว้

5.1.2.3 ส่วนบรอดคาสต์เซิร์ฟเวอร์ (Broadcast Server)

สามารถรับข้อมูลเสียง และทำการจัดรูปแบบของข้อมูลให้สามารถส่งแบบเรียลไทม์ผ่านโปรโตคอลอาร์ทีพีดี โดยสามารถสร้างเป็นแพ็คเกจอาร์ทีพี (RTP-Packets) และกระจายแพ็คเกจเหล่านั้น ไปยังโหนดเอ็นทีที่อยู่ในเครือข่ายแลน (LAN) เดียวกันได้ เนื่องจากไม่มีอุปกรณ์เครือข่ายที่สนับสนุนมัลติคาสท์ทำให้ไม่สามารถทำการทดลองข้ามเครือข่ายได้ แต่จากการทดลองในเครือข่ายเดียวกันโดยใช้หลักการเดียวกันปรากฏว่าสามารถทำได้ จึงสามารถสรุปว่าส่วนบรอดคาสต์เซิร์ฟเวอร์สามารถใช้งานได้ตามที่ออกแบบ

5.2 ปัญหาที่เกิดขึ้นระหว่างการทดลอง

5.2.1 ปัญหาที่เกิดขึ้นระหว่างขั้นตอนการสร้างระบบ

5.2.1.1 การไม่เข้มงวดในขั้นตอนการศึกษาและการออกแบบ

ในโมดูลย่อยนั้น การข้ามไปทำการเขียนโปรแกรมก่อนที่จะออกแบบโปรแกรมให้ดีทำให้ต้องเสียเวลาแก้โปรแกรมใหม่ในเวลาต่อมา ทำนองเดียวกันกับการมองข้ามความต้องการของผู้ใช้บางจุด ทำให้ต้องเสียเวลาแก้ส่วนที่ออกแบบไปแล้วใหม่เพื่อให้ตรงกับความต้องการที่แท้จริง นอกจากนี้ปัญหาใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่พบคือ การไม่เข้มงวดในขั้นตอนการศึกษาทฤษฎีและการศึกษาระบบเก่าไปทำขั้นตอนการออกแบบและขั้นตอนการเขียนโปรแกรมเลย ทำให้มีบางส่วนที่ออกแบบและลงมือทำไปแล้วทฤษฎีไม่รองรับ จึงต้องแก้ไขการออกแบบและโปรแกรมใหม่ทั้งหมดในภายหลัง

5.2.1.2 การใช้ชุดเครื่องมือภาษาจาวา (J2SDK) ต่างรุ่นกันของผู้พัฒนา

การใช้ชุดเครื่องมือภาษาจาวาที่ต่างรุ่นกันในการพัฒนาโปรแกรมเดียวกัน เป็นปัญหาสำคัญ เนื่องจากในรุ่นที่ใหม่กว่า จะมีบางคลาสที่ถูกออกแบบให้เปลี่ยนแปลงไป ทำให้โปรแกรมที่เขียนกับรุ่นหนึ่งซึ่งสามารถทำงานได้ปกติ กลายเป็นโปรแกรมที่มีความผิดพลาดในบางส่วนหากสลับชุดเครื่องมือต่างรุ่นกันทำงาน บางกรณีถึงขั้นคอมไพล์ไม่ผ่าน และไม่สามารถทำงานได้เลยทีเดียว

5.2.1.3 การใช้เครื่องมือสำหรับเขียนโปรแกรมค้างยี่ห้อยกัน

การใช้ชุดเครื่องมือสำหรับเขียนโปรแกรมที่ค้างยี่ห้อยกันในการพัฒนาตัวโปรแกรมเดียวกันเป็นปัญหาสำคัญ ปัญหาเกิดจากสองประเด็นหลักคือ ข้อหนึ่ง ชุดเครื่องมือแต่ละยี่ห้อยจะมีมาตรฐานค้างยี่ห้อยถึงแม้ว่าจะอยู่บนพื้นฐานของโครงสร้างภาษาจาวาก็ตาม เช่น ในช่วงแรกมีผู้พัฒนาสองคน คนหนึ่งใช้โปรแกรมอิดิทพลัสในการพัฒนาโปรแกรม ส่วนอีกคนใช้โปรแกรมบอร์แลน เจ บิลด์เดอร์ ส่วนที่ต่างกันคือ อิดิทพลัสจะไม่มีการใช้แพ็คเกจ (Package) เมื่อเรานำซอร์สโค้ด (Source Code) ของคนแรกมาส่งให้มันทำงานบนสิ่งแวดล้อม (Environment) ของคนที่สอง ซอร์สโค้ดนั้นจะมีปัญหาเกิดขึ้น ทำให้โปรแกรมไม่สามารถทำงานได้ตามปกติ ต้องทำการแก้ไขโค้ดให้ตรงกับมาตรฐานของอีกฝ่ายก่อน และข้อที่สอง สำหรับชุดเครื่องมือพัฒนาบางยี่ห้อยก็อาจมีแพ็คเกจพิเศษที่สร้างมาเฉพาะของยี่ห้อยนั้น ซึ่งถ้ามีการใช้งานแพ็คเกจพิเศษนี้ ผู้พัฒนาร่วมที่ใช้ชุดเครื่องมือพัฒนาต่างกัน จะต้องทำการเพิ่มแพ็คเกจพิเศษนี้ในสิ่งแวดล้อมของตนเองจึงจะทำให้โปรแกรมสามารถทำงานได้ตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ปัญหาที่เกิดระหว่างขั้นตอนการทดลอง

5.2.2.1 เครื่องข่ายและอุปกรณ์เครือข่ายไม่สนับสนุนการทดลองส่งข้อมูลแบบมัลติคาสท์

เป็นปัญหาที่ใหญ่ที่สุดในการทดลองครั้งนี้ เนื่องจากไม่มีอุปกรณ์เครือข่ายที่สนับสนุนมัลติคาสท์ ทำให้ไม่สามารถทำการทดลองข้ามเครือข่ายได้ หรือทำการทดลองผ่านโมเด็มได้ แม้ว่าการทดลองในเครือข่ายเดียวกันโดยใช้หลักการเดียวกันสามารถสรุปได้ว่า ส่วนบรอดคาสท์เซิร์ฟเวอร์สามารถใช้งานได้ตามที่ออกแบบ แต่การที่ไม่ได้ทำการทดลองจริง ทำให้ไม่ทราบรายละเอียดสำคัญที่แท้จริงได้ เช่น ค่าเฉลี่ย ทราฟฟิก ของการเชื่อมต่อผ่านเนตเวิร์กขนาดใหญ่

5.2.2.2 การขาดแคลนอุปกรณ์มิกซ์เซอร์ (ชนิดฮาร์ดแวร์) ซึ่งมีราคาแพง

มิกซ์เซอร์เป็นอุปกรณ์ที่มีราคาแพง และมีขนาดใหญ่ไม่สะดวกต่อการนำมาทดลองบ่อยครั้ง ทำให้การทดลองในส่วนมิกซ์เซอร์ที่เป็นฮาร์ดแวร์ต้องรอโอกาสที่เหมาะสม ซึ่งเป็นเวลาไม่มากนัก เป็นอีกปัญหาหนึ่งในการทำการทดลอง

5.2.3 ปัญหาอื่นๆ

5.2.3.1 ลิขสิทธิ์ของเอ็มทีทีเอ็นไอค็อคเคอร์

ปัจจุบันทางซันไมโครซิสเต็มส์ (Sun Microsystems) ได้ตัดส่วนเอ็มทีทีเอ็นไอค็อคเคอร์ออกจากจาวามีเดียเฟรมเวิร์กเรียบร้อยแล้ว เนื่องจากไม่ต้องการให้เกิดปัญหาละเมิดลิขสิทธิ์ของผู้คิดค้นเอ็มทีทีเอ็นไอค็อคเคอร์

5.3 แนวทางการพัฒนาระบบต่อ

5.3.1 การพัฒนาส่วนจัดการการกระจายเสียง

ส่วนจัดการการกระจายเสียง (Program Management Module) ของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ตนั้น นับเป็นจุดสำคัญจุดหนึ่งที่ทำให้ระบบนี้ต่างจากวิทยุกระจายเสียงผ่านอินเทอร์เน็ตโดยทั่วไป ระบบสามารถนำมาพัฒนาเพิ่มเติมให้สามารถอำนวยความสะดวกให้แก่ผู้ใช้ได้มากยิ่งขึ้น จากการวิเคราะห์สามารถพัฒนาเพิ่มเติมในส่วนต่าง ๆ ของระบบดังนี้

5.3.1.1 พัฒนารายการจัดเก็บไฟล์เสียงและแสดงรายละเอียดของไฟล์เสียง

พัฒนาให้สามารถเก็บรายละเอียดของไฟล์เสียงเพลง และเสียงโฆษณา ได้มากขึ้นเช่น ชื่อวงหรือชื่อศิลปิน ชื่ออัลบั้ม ปีที่ออกอัลบั้ม แนวเพลง ชื่อเพลง ผู้แต่ง เนื้อร้อง และสามารถแบ่งแสดงไฟล์ตามหมวดหมู่ได้

5.3.1.2 พัฒนารายการแสดงผลรายการล่วงหน้า

พัฒนาให้สามารถแสดงรายละเอียดได้มากขึ้นเช่น แฟล็ก (Flag) สำหรับระบุว่ารายการนี้มีการใช้ไปแล้วหรือยัง และอาจจะพัฒนาให้ระบบนั้นมีการเก็บล็อกไฟล์ (Log Files) ของการใช้ผังรายการล่วงหน้าเพื่อให้ผู้ควบคุมสามารถทราบรายละเอียดหากเกิดปัญหาระหว่างการออกอากาศ

5.3.2 การพัฒนาโมดูลทางไกลเอ็นทีเพื่อแก้ไขปัญหาลิขสิทธิ์เอ็มพีทีรีเอ็นโค้ดเดอร์

พัฒนาโมดูลหรือปลั๊กอินสำหรับโปรแกรมทางฝั่งไคลเอ็นท์ขึ้นมาเอง สำหรับการถอดรหัสสัญญาณสตรีมเสียงแบบอ็อก-วอร์บิส (Ogg-Vorbis) ซึ่งเป็นมาตรฐานการเข้ารหัสไฟล์เสียงชนิดใหม่ที่แจกฟรี ไม่เสียค่าลิขสิทธิ์ และมีแนวโน้มว่าจะเป็นที่ยอมรับตามมาตรฐานแบบเอ็มพีทีรีในอนาคตอันใกล้นี้ ในส่วนฝั่งเซิร์ฟเวอร์ซึ่งต้องใช้การเข้ารหัสแบบอ็อก-วอร์บิสนั้น ทางชันไมโครซิสเต็มส์ผู้ผลิตจาวาได้พัฒนาโมดูลไว้เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. B. Cavagnolo, J. Bier **Introduction to Digital Audio Compression**
(Canada: Berkeley Design Technology, Inc.)
2. H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund **Real Time Streaming Protocol (RTSP)**
(Columbia: U./Cisco/RealNetworks/Ericsson, November 01, 200)
3. Henning Schulzrinne **Internet Media-on-Demand: The Real-Time Streaming Protocol**
(New York: Dept. of Computer Science Columbia University New York)
4. Sun Microsystems **Java™ Media Framework API Guide**
(California, U.S.A. ; Sun Microsystems, Inc., 1999)
5. Sarom Ing, Steve Rudkin **Simplifying Real-Time Multimedia Application Development Using Session Descriptions** (BT Laboratories, Martlesham Heath Ipswich IP5 3RE)
6. H.M. Deitel, P.J. Deitel **JAVA HOWTO PROGRAM Fifth Edition**
(New Jersey: Pearson Education, Inc., 2003)
7. สุวัฒน์ ปู่อะชัยยะ, คัน ดันท์สุทธิวงศ์, สุพจน์ ปู่อะชัยยะ **เปิดโลก TCP/IP และโปรโตคอลของอินเทอร์เน็ต** (กรุงเทพมหานคร : บริษัท คำนวณสุทธาการพิมพ์, 2545)
8. ดร.วีระศักดิ์ ชิงถาวร, **JAVA Programming Volume I**
(กรุงเทพมหานคร : บริษัท ซีเอ็ดดูเลชั่น จำกัด (มหาชน), 2545)
9. ดร.วีระศักดิ์ ชิงถาวร, **JAVA Programming Volume II**
(กรุงเทพมหานคร : บริษัท ซีเอ็ดดูเลชั่น จำกัด (มหาชน), 2545)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้