สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ต

INTERNET RADIO BROADCAST MANAGEMENT SYSTEM

T055720

โดย

นางสาวชลลดา เรืองศิลป์ประเสริฐ

นางสาวชุติมน ประวีณวรกุล

นางสาวหทัยชนก นาลประเสริฐ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 55720
วัน,เดือน,ปี 25 พ.ค. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

INTERNET RADIO BROADCAST MANAGEMENT SYSTEM

BY

MISS CHONLADA RUANGSINPRASERT

MISS CHUTIMON PRAWEENVORAKUL

MISS HATHAICHANOK NUANPRASERT

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

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

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....

(อาจารย์ภุชงค์ หงษ์สุวรรณ)

อาจารย์ผู้ควบคุมวิทยานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

บทคัดย่อ

ปริญญานิพนธ์นี้ กล่าวถึงการนำเทคโนโลยีอินเทอร์เนตมาประยุกต์ใช้ร่วมกับการสื่อสารแบบวิทยุกระจายเสียง มีจุดประสงค์หลักเพื่อสร้างแอพพลิเคชันสำหรับอำนวยความสะดวกให้แก่ผู้ดำเนินรายการในระบบวิทยุกระจายเสียงผ่านอินเทอร์เนต โดยการทำงานของแอพพลิเคชันจะแบ่งออกเป็นสองส่วนหลักคือ ส่วนออดิโอเซิร์ฟเวอร์ จะทำหน้าที่รับการเชื่อมต่อจากไคลเอนท์ และทำการส่งข้อมูลเสียงไปยังไคลเอนต์ทุกเครื่องที่มีการเชื่อมต่อมายังเซิร์ฟเวอร์อยู่ในขณะนั้น โดยส่งข้อมูลเป็นแบบเรียลไทม์เพื่อให้มีรูปแบบการสื่อสารเป็นแบบเดียวกับระบบวิทยุกระจายเสียงแบบธรรมดา กล่าวคือ เซิร์ฟเวอร์จะส่งข้อมูลเสียงไปอย่างต่อเนื่องในขณะที่ไคลเอนท์จะทำการเล่นเสียงไปพร้อม ๆ กับการรับข้อมูลเสียงจากเซิร์ฟเวอร์ ไม่ต้องดาวน์โหลดข้อมูลเสียงมาทั้งหมดมาเก็บไว้ก่อนจึงทำการเล่นเสียง และส่วนที่สองคือ โปรแกรมจัดการการกระจายเสียง มีหน้าที่จัดการกับข้อมูลเสียงที่จะออกอากาศ และควบคุมการออกอากาศ โดยผู้ดำเนินรายการสามารถจัดการและควบคุมผ่านทางส่วนติดต่อกับผู้ใช้ซึ่งออกแบบมาเพื่อจุดประสงค์หลักคือ ให้ผู้ดำเนินรายการสามารถใช้งานได้ง่าย ประหยัดเวลา และช่วยลดภาระระหว่างดำเนินรายการให้ได้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Internet Radio Broadcast Management System	
Student	Miss Chonlada Ruangsinprasert	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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ หากไม่มีอาจารย์ภุชงค์ หงษ์สุวรรณ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ผู้คอยช่วยเหลือ ให้คำปรึกษา แนะนำ ตรวจแก้ไข และเอาใจใส่ตลอดระยะเวลาทั้งหมดที่ทำปริญญานิพนธ์ ซึ่งคณะผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอาจารย์เป็นอย่างยิ่ง

ขอขอบคุณอาจารย์หลาย ๆ ท่านประจำภาควิชาวิศวกรรมสารสนเทศ ที่เป็นผู้ริเริ่มคิดโครงงานนี้ และคอยห่วงใยความเป็นไปของโครงงานมาโดยตลอด ได้แก่ อาจารย์กฤษดากร กล่อมการ อาจารย์ดลชัย สุขเจริญผล และขอขอบพระคุณอาจารย์พนารัตน์ ระวิวรรณ ที่ช่วยให้คำแนะนำและตอบข้อซักถามของพวกเราอย่างเต็มใจ ขอบคุณเพื่อน ๆ ในภาควิชาวิศวกรรมสารสนเทศทุกคน โดยเฉพาะอย่างยิ่งเพื่อน ๆ ในห้องโปรเจคซอฟท์แวร์ สำหรับมิตรภาพ กำลังใจ และน้ำใจที่มีให้กันเสมอมา

สุดท้ายนี้คณะผู้จัดทำต้องขอกราบขอบพระคุณ บุคคลที่สำคัญที่สุดที่ทำให้มีวันนี้ นั่นคือ บิดา มารดา ที่เคารพรักของคณะผู้จัดทำ และคุณยายทองสุข รัตนชัย สำหรับความห่วงใย กำลังใจ ความเข้าใจ และคอยสนับสนุนในทุกด้านตลอดมา

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
สารบัญ	จ
สารบัญรูปภาพ	ฌ
สารบัญตาราง	ฑ
บทที่ 1 บทนำ	1
1.1 แนวคิดและที่มาของปัญหา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 ขั้นตอนการดำเนินโครงงาน	3
1.5 ผลที่คาดว่าจะได้รับ	3
1.6 อุปกรณ์ที่ต้องใช้	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 การเข้ารหัสสัญญาณเสียง	4
2.1.1 ไซโคอคูสติกโมเดล (Psychoacoustics Model)	5
2.1.2 การวิเคราะห์เชิงความถี่ (Spectral Analysis)	7
2.1.3 การลดสัญญาณรบกวน (Noise Allocations)	8
2.1.4 เทคนิคอื่น ๆ ในการบีบอัด	8
2.2 การเข้ารหัสสัญญาณเสียงแบบเอ็มพีทรี (MPEG-1 Audio Layer 3)	9
2.2.1 ลักษณะของสัญญาณเสียงแบบเอ็มพีทรี	9
2.2.2 วิธีการเข้ารหัสสัญญาณเสียงแบบเอ็มพีทรี	10
2.3 โปรโตคอลที่ใช้ในการส่งข้อมูลแบบเรียลไทม์	11
2.3.1 ลักษณะของการสื่อสารแบบเรียลไทม์	12
2.3.2 โปรโตคอลอาร์ทีพี (RTP Protocol)	15
2.3.3 โปรโตคอลอาร์ทีซีพี (RTCP Protocol)	17
2.3.4 โปรโตคอลอาร์ทีเอสพี (RTSP Protocol)	19
2.4 จาวามีเดียเฟรมเวิร์ก (Java Media Framework, JMF)	20
2.4.1 การจับข้อมูลมัลติมีเดีย	21
2.4.2 ดาต้าซอร์ส	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4.3 เพลย์เยอร์	25
2.4.4 โปรเซสเซอร์ (Processor)	27
2.4.5 เมเนเจอร์ (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 สแปนนิ่งทรี (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 เอ็มโอเอสพีเอฟ (Multicast Extensions to OSPF (MOSPF))	43
2.6 การเขียนโปรแกรมจาวาด้วยเนทีฟเมธอด	45
2.6.1 การเขียนโค้ดภาษาจาวา	47
2.6.2 การคอมไพล์โค้ดจาวา	49
2.6.3 การสร้างไฟล์เฮดเดอร์	49
2.6.4 การสร้างเนทีฟเมธอด	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 โดเมน โมเดล (Domain Model)	56
3.2.3 ยูสเคส ไดอะแกรม (Use Case Diagram)	58
3.2.4 แอคทิวิตี้ไดอะแกรม (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 ทดลองเรียกแอพพลิเคชั่น	68
4.2 ทดลองสร้างผังรายการปัจจุบันด้วยโหมดการทำงานแบบควบคุมด้วยมือ (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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.11 ทดลองเพิ่มเสียงเพลงหรือเสียงโฆษณาลงในผังรายการล่วงหน้า (Auto Playlist)	88
4.12 ทดลองแทรกช่วงหยุดเล่นมีเดียชั่วคราวในผังรายการล่วงหน้า	90
4.13 ทดลองลบมีเดียออกจากผังรายการ	93
4.14 ทดลองการตั้งค่าวันที่ และเวลาของผังรายการล่วงหน้า	95
4.15 ทดลองบันทึกการสร้างผังรายการล่วงหน้า	97
4.16 ทดลองแก้ไขผังรายการล่วงหน้า	101
4.17 ทดลองการล้างผังรายการล่วงหน้าที่สร้างไว้	106
4.18 ทดลองลบผังรายการล่วงหน้า	109
4.19 ทดลองสลับการทำงานเป็นโหมดอัตโนมัติ	111
4.20 ทดลองส่งข้อมูลเสียงไปยังไคลเอนท์	114
4.21 ทดลองปรับระดับเสียงของระบบ	119
4.22 ทดลองปรับระดับเสียงไมโครโฟน	120
บทที่ 5 สรุปผลการทดลอง	123
5.1 สรุปผลการทดลอง	123
5.1.1 ลักษณะของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เน็ตโดยรวม	123
5.1.2 สรุปผลการทดลองแต่ละส่วน	124
5.2 ปัญหาที่เกิดขึ้นระหว่างการทดลอง	124
5.2.1 ปัญหาที่เกิดระหว่างขั้นตอนการสร้างระบบ	124
5.2.2 ปัญหาที่เกิดระหว่างขั้นตอนการทดลอง	126
5.2.3 ปัญหาอื่น ๆ	126
5.3 แนวทางการพัฒนาระบบต่อ	127
5.3.1 การพัฒนาส่วนจัดการการกระจายเสียง	127
5.3.2 การพัฒนาโมดูลทางไคล์เอนท์เพื่อแก้ไขปัญหาลิขสิทธิ์เอ็มพีทรีเอนโค้ดเดอร์	127
บรรณานุกรม	128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

เรื่อง		หน้า
รูปที่ 2.1	การเข้ารหัสสัญญาณเสียงของเอนโค้ดเดอร์แบบทั่วไป	4
รูปที่ 2.2	เธรสโฮลด์ของการได้ยินมีจนถึงที่ความถี่ 20 กิโลเฮิร์ตซ์	5
รูปที่ 2.3	การบดบังทางความถี่	6
รูปที่ 2.4	การบดบังแบบชั่วคราว	6
รูปที่ 2.5	บล็อกไดอะแกรมของเอนโค้ดเดอร์ของเอ็มพีทรี	10
รูปที่ 2.6	ความสัมพันธ์เชิงเวลาของการสื่อสารแบบเรียลไทม์	12
รูปที่ 2.7	ปัญหาจิตเตอร์	13
รูปที่ 2.8	การแก้ปัญหาจิตเตอร์โดยใช้ไทม์สแตมป์	13
รูปที่ 2.9	รูปแบบส่วนหัวของแพคเกตอาร์ทีพี	15
รูปที่ 2.10	รูปแบบส่วนหัวของแพคเกตอาร์ทีซีพี	18
รูปที่ 2.11	สถานะของเซิร์ฟเวอร์	20
รูปที่ 2.12	เปรียบเทียบการทำงานของกล้องวีดีโอกับเจเอ็มเอฟ	21
รูปที่ 2.13	ดาต้าโมเดลของเจเอ็มเอฟ	22
รูปที่ 2.14	ฟอร์แมตของมีเดียในเจเอ็มเอฟ	24
รูปที่ 2.15	แบบจำลองของเพลย์เยอร์ในเจเอ็มเอฟ	25
รูปที่ 2.16	คลาสไดอะแกรมของเพลย์เยอร์ในเจเอ็มเอฟ	25
รูปที่ 2.17	สถานะของเพลย์เยอร์	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	การทำอุโมงค์ผ่านเครือข่ายที่ไม่สนับสนุนการทำมัลติคาสต์	35
รูปที่ 2.26	รูปแบบของไอพีแอดเดรสคลาสดี	36
รูปที่ 2.27	สแปนนิ่งทรี	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เรื่อง		หน้า
รูปที่ 2.28	อาร์พีบีทรี	39
รูปที่ 2.29	อาร์พีเอ็มทรี	40
รูปที่ 2.30	เอสทีทรี	41
รูปที่ 2.31	ซีบีที	42
รูปที่ 2.32	แสดงขั้นตอนการสร้างโปรแกรม Hello World	46
รูปที่ 2.33	แสดงให้เห็นถึงรูปแบบชื่อของเนทีฟฟังก์ชันที่ถูกสร้างจากเนทีฟเมธอด	50
รูปที่ 3.1	สถาปัตยกรรมของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เนต	55
รูปที่ 3.2	โดเมนโมเดลของระบบจัดการการกระจายเสียงผ่านอินเทอร์เนต	57
รูปที่ 3.3	ยูสเคสไดอะแกรมสำหรับการจับเสียงจากไมโครโฟน	58
รูปที่ 3.4	ยูสเคสไดอะแกรมสำหรับการเข้ารหัสสัญญาณเสียงให้อยู่ในรูปแบบแพ็กเกตอาร์ทีพี	58
รูปที่ 3.5	ยูสเคสไดอะแกรมสำหรับการเลือกสัญญาณเสียงสำหรับส่งมีเดียสตรีม	58
รูปที่ 3.6	ยูสเคสไดอะแกรมสำหรับการส่งสตรีมเสียงไปยังไคลเอนท์	59
รูปที่ 3.7	แอคติวิตี้ไดอะแกรมสำหรับการจับ เสียงจากไมโครโฟน	59
รูปที่ 3.8	แอคติวิตี้ไดอะแกรมสำหรับการเข้ารหัสสัญญาณเสียงให้อยู่ในรูปแบบแพ็กเกตอาร์ทีพี	59
รูปที่ 3.9	แอคติวิตี้ไดอะแกรมสำหรับการเลือกสัญญาณเสียงสำหรับส่งมีเดียสตรีม	60
รูปที่ 3.10	แอคติวิตี้ไดอะแกรมสำหรับการการส่งสตรีมเสียงไปยังไคลเอนท์	60
รูปที่ 3.11	หน้าต่างหลักสำหรับการจัดการการจายเสียงผ่านอินเทอร์เนต	61
รูปที่ 3.12	หน้าต่างสำหรับสร้างผังรายการล่วงหน้า	64
รูปที่ 3.13	หน้าต่างสำหรับแก้ไขผังรายการล่วงหน้า	67
รูปที่ 4.1	หน้าต่างหลัก	68
รูปที่ 4.2	วิธีทดลองเลือกแหล่งข้อมูลเสียงโดยใช้ตัวควบคุมด้วยมือ	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	หน้าต่างหลักก่อนลบมีเดียที่เลือกไว้ในผังรายการปัจจุบัน	83
รูปที่ 4.16	หน้าต่างหลักเมื่อทำการลบมีเดียที่เลือกไว้ออกจากผังรายการปัจจุบัน	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	ไดอะล้อกซ์อินพุต (Input Dialog Box) สำหรับรับค่าช่วงเวลาว่าง หน่วยเป็นวินาที	92
รูปที่ 4.25	ทดลองป้อนค่าช่วงเวลาว่างในไดอะล้อกซ์อินพุต	92
รูปที่ 4.26	หน้าต่างควบคุมแบบอัตโนมัติ เมื่อทำการแทรกการหยุดเล่นมีเดียชั่วคราว	93
รูปที่ 4.27	วิธีลบมีเดียออกจากผังรายการล่วงหน้า	94
รูปที่ 4.28	หน้าต่างสำหรับสร้างผังรายการล่วงหน้าหลังจากทำการลบมีเดียออกจากผังรายการ	95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เรื่อง		หน้า
รูปที่ 4.29	วิธีการตั้งค่าวันที่และเวลาเริ่มต้นของผังรายการล่วงหน้า	96
รูปที่ 4.30	หน้าต่างสำหรับสร้างผังรายการล่วงหน้าเมื่อทำการตั้งค่าวันที่และเวลาสำหรับผังรายการล่วงหน้า	97
รูปที่ 4.31	วิธีการบันทึกการสร้างผังรายการล่วงหน้า	98
รูปที่ 4.32	ไดอะล้อคบ๊อกซ์เตือนกรณีไม่มีรายการเสียงในผังรายการที่สร้าง	99
รูปที่ 4.33	ไฟล์แสดงรายละเอียดของผังรายการล่วงหน้า	99
รูปที่ 4.34	ข้อมูลของผังรายการล่วงหน้าในหน้าต่างหลักเพิ่มขึ้นหลังจากทำการบันทึกผังรายการ	100
รูปที่ 4.35	ไดอะล้อคบ๊อกซ์เตือนการตั้งค่าวันที่และเวลาผิดพลาด	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	ไดอะล้อคบ๊อกซ์ยืนยันการเคลียร์รายละเอียดผังรายการล่วงหน้า	108
รูปที่ 4.43	หน้าต่างสำหรับสร้างผังรรายการล่วงหน้าเมื่อทำการล้างผังรายการที่สร้างไว้	109
รูปที่ 4.44	หน้าต่างหลัก ก่อนการลบผังรายการล่วงหน้า	110
รูปที่ 4.45	หน้าต่างหลักเมื่อทำการลบผังรายการล่วงหน้า	111
รูปที่ 4.46	หน้าต่างหลัก ก่อนการสลับเป็นโหมดอัตโนมัติ	112
รูปที่ 4.47	หน้าต่างหลักหลังจากสลับการทำงานเป็นโหมดอัตโนมัติ	113
รูปที่ 4.48	หน้าต่างหลักเมื่อทำงานในโหมดอัตโนมัติตามเวลาเริ่มต้นของผังรายการ	114
รูปที่ 4.49	วิธีทดลองส่งข้อมูลเสียงไปยังไคลเอนท์	115
รูปที่ 4.50	วิธีการเชื่อมต่อของไคล์เอนท์โปรแกรมมายังระบบ	116

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เรื่อง		หน้า
รูปที่ 4.51	หน้าต่างหลักเมื่อทำการส่งข้อมูลเสียงไปยังไคลเอนท์	117
รูปที่ 4.52 ก.	โปรแกรมไคล์เอนท์เมื่อเริ่มทำการเชื่อมต่อไปยังเซิร์ฟเวอร์	118
รูปที่ 4.52 ข.	โปรแกรมไคล์เอนท์เมื่อเริ่มการรับข้อมูลเสียง	118
รูปที่ 4.53	วิธีปรับเสียงระดับเสียงของระบบ	119
รูปที่ 4.54	โปรแกรมไคล์เอนท์เมื่อทำการปรับระดับเสียงของระบบ	120
รูปที่ 4.55	วิธีปรับเสียงไมโครโฟน	121
รูปที่ 4.56	โปรแกรมไคล์เอนท์เมื่อทำการปรับระดับเสียงไมโครโฟน	122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

เรื่อง		หน้า
ตารางที่ 2.1	ตารางแสดงคุณสมบัติของมาตรฐานเอ็มเปกออดิโอแต่ละเลเยอร์	10
ตารางที่ 2.2	ชนิดของข้อมูลในแพคเกตอาร์ทีพี	16
ตารางที่ 2.3	ชนิดของแพคเกตอาร์ทีซีพี	18
ตารางที่ 2.4	เมธอดของโปรโตคอลอาร์ทีเอสพี	20
ตารางที่ 2.5	ค่าต่าง ๆ ที่ใช้ในการควบคุมขอบเขตทีทีแอล	43
ตารางที่ 4.1	สถานะเริ่มต้นของแอพพลิเคชัน	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 แนวคิดและที่มาของปัญหา

จากการศึกษาปัญหาของระบบวิทยุกระจายเสียงผ่านอินเทอร์เนตที่มีอยู่ในปัจจุบันพบว่า ผู้ดำเนินรายการต้องรับภาระในการดำเนินรายการวิทยุและต้องคอยควบคุมการออกอากาศเอง ทำให้เสียเวลาและอาจเกิดความผิดพลาดระหว่างการควบคุมได้ง่าย จึงได้เกิดแนวคิดสร้างระบบจัดการการกระจายเสียงผ่านอินเทอร์เนต โดยโครงงานนี้มีจุดประสงค์หลักเพื่อสร้างแอพพลิเคชันสำหรับอำนวยความสะดวกให้แก่ผู้ดำเนินรายการในระบบวิทยุกระจายเสียงผ่านอินเทอร์เนต ซึ่งระบบนี้ประกอบด้วยสองส่วนหลัก ๆ คือ ส่วนบรอดคาสท์เซิร์ฟเวอร์ (Audio Broadcast Server) ทำหน้าที่กระจายข้อมูลเสียงแบบมัลติคาสท์ไปยังไคลเอนต์ และส่วนที่สองนั้นคือ โปรแกรมสำหรับจัดการการกระจายเสียง (Program Management) มีหน้าที่จัดการกับข้อมูลเสียงที่จะออกอากาศ และควบคุมการออกอากาศ โดยผู้ดำเนินรายการสามารถจัดการการกระจายเสียงผ่านทางส่วนติดต่อกับผู้ใช้ (Graphic User -Interface) โดยเลือกการควบคุมการทำงานของระบบได้สองแบบคือ แบบควบคุมด้วยมือ (Manual Mode) และแบบควบคุมอัตโนมัติ (Automatically Playing Mode) โดยการพัฒนาระบบเลือกใช้ภาษาจาวา เนื่องจากตัวภาษามีความง่าย, ยืดหยุ่น และคงทน นอกจากนั้นยังทำให้แอพพลิเคชันสามารถทำงานได้กับทุกแพล็ตฟอร์ม

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการทำงานของระบบวิทยุกระจายเสียงผ่านอินเทอร์เนต
2. เพื่อพัฒนาแอพพลิเคชันสำหรับจัดการการกระจายเสียงผ่านอินเทอร์เนตโดยใช้ภาษาจาวาบนแพล็ตฟอร์มวินโดวส์ (Window Platform)

1.3 ขอบเขตของโครงงาน

1.3.1. พัฒนาโปรแกรมส่วนจัดการการส่งข้อมูล

โครงงานนี้มีเป้าหมายในการพัฒนาโปรแกรมส่วนจัดการการส่งข้อมูล หรือโปรแกรมจัดการการกระจายเสียง (Program Management Module) ให้มีความสามารถสนับสนุนการจัดการการกระจายเสียงได้สองแบบ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.1.1. แบบควบคุมด้วยมือ (Manual Mode)

โดยจะพัฒนาให้มีคุณสมบัติดังนี้

1. สามารถเลือกไฟล์เสียงเพลงหรือเสียงโฆษณาสำหรับรอการออกอากาศได้
2. สามารถคำนวณเวลาสำหรับแต่ละไฟล์เสียงที่เลือกในข้อหนึ่งได้
3. สามารถเลือกลบแหล่งเสียงที่เลือกไว้ในข้อหนึ่งได้
4. สามารถควบคุมการเล่นและการหยุดเพลงได้
5. เมื่อเพลงจบลง สามารถเล่นเพลงต่อไปได้ทันที
6. สามารถควบคุมระดับความดังของเสียงจากไฟล์เสียงและเสียงจากไมโครโฟนได้
7. สามารถควบคุมการผสมเสียงระหว่างเสียงจากไฟล์เสียงกับเสียงจากไมโครโฟนได้สามารถส่งข้อมูลเสียงไปยังบรอดคาสท์เซิร์ฟเวอร์ได้

1.3.1.2. แบบควบคุมอัตโนมัติ (Automatically Playing Mode)

โดยจะพัฒนาให้มีคุณสมบัติดังนี้

1. สามารถแสดงรายการของไฟล์เสียงแบ่งตามแชนแนลดังนี้ แชนแนลเสียงเพลง และแชนแนลเสียงโฆษณา
2. สามารถสร้างผังรายการล่วงหน้า โดยการเลือกแหล่งเสียงจากรายการข้างต้นได้
3. สามารถใส่ช่วงว่าง (Pause Time) ลงในผังรายการได้โดยกำหนดช่วงเวลาในหน่วยวินาที
4. สามารถแสดงเวลาที่ใช้ในแต่ละรายการได้
5. สามารถคำนวณเวลาทั้งหมดและเวลาสิ้นสุดรายการได้โดยกำหนดเวลาเริ่มรายการให้
6. สามารถสร้างรายการได้มากกว่าหนึ่งรายการ
7. สามารถกำหนดเวลาในการนำรายการขึ้นมาใช้งานได้
8. สามารถส่งข้อมูลเสียงไปยังบรอดคาสท์เซิร์ฟเวอร์ตามที่ได้กำหนดไว้ในผังรายการได้อย่างถูกต้อง
9. ผังรายการที่สร้างไว้ สามารถนำมาแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.2. พัฒนาโปรแกรมบรอดคาสท์เซิร์ฟเวอร์

ให้มีคุณสมบัติดังนี้

1. สามารถเข้ารหัสข้อมูลเสียงให้สามารถส่งผ่านโปรโตคอลในเนตเวิร์กแบบเรียลไทม์ได้
2. สามารถส่งสตรีมเสียงแบบมัลติคาสท์ไปยังมัลติคาสท์เราเตอร์ได้
3. สามารถให้บริการไคล์เอนท์หลาย ๆ การเชื่อมต่อได้ในเวลาเดียวกัน

1.4 ขั้นตอนการดำเนินโครงงาน

1. ศึกษาการทำงานของระบบวิทยุกระจายเสียงผ่านระบบอินเทอร์เนต
2. ศึกษาหลักการของจาวามีเดียเฟรมเวิร์ก (Java Media Framework)
3. ศึกษาการเขียนโปรแกรมเชิงวัตถุโดยใช้ภาษาจาวา
4. รวบรวมความต้องการของผู้ใช้
5. ออกแบบระบบ
6. ลงมือพัฒนาระบบแบบแยกส่วน
7. ทดสอบการใช้งานแบบแยกส่วน
8. รวบรวมส่วนต่าง ๆ เข้าด้วยกันและแก้ไขข้อผิดพลาด
9. ติดตั้งแอพพลิเคชันและทดสอบการใช้งาน
10. บันทึกและสรุปผลการทดลอง
11. จัดทำปริญญานิพนธ์

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถพัฒนาระบบจัดการการกระจายเสียงผ่านทางอินเทอร์เนต
2. สามารถพัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการวินโดวส์ได้
3. สามารถพัฒนาแอพพลิเคชั่นด้วยภาษาจาวาได้

1.6 อุปกรณ์ที่ต้องใช้

1. คอมพิวเตอร์ที่สามารถรองรับระบบมัลติมีเดียและเชื่อมต่อกับเครือข่ายอินเทอร์เนต จำนวน 2 เครื่อง
2. ซอฟแวร์ j2sdk 1.4.2
3. ซอฟแวร์ JMF 2.1.1

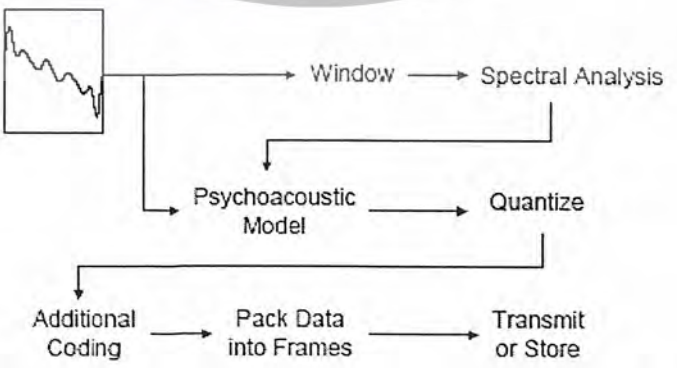
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 การเข้ารหัสสัญญาณเสียง

เทคนิคในการประมวลผลสัญญาณ (Signal Processing Techniques) เป็นหนึ่งในเทคโนโลยีขั้นก้าวหน้าของเสียงดิจิตอล ตัวอย่างสำคัญที่มีการนำเทคนิคในการประมลผลสัญญาณมาประยุกต์ใช้ ได้แก่แอพพลิเคชันสำหรับบรอดคาสท์ผ่านอินเทอร์เนต ถ้าหากต้องการบรอดคาสท์ข้อมูลเสียงที่มีคุณภาพสูงผ่านระบบอินเทอร์เนต โดยข้อมูลเสียงถูกเข้ารหัสแบบพีซีเอ็ม (Pulse Code Moduration (PCM)) 16 บิต ที่ความถี่ 44.1 กิโลเฮิร์ตซ์ แอพพลิเคชันต้องใช้ช่องสัญญาณ (Channel) ที่มีแบนด์วิธ (Bandwidth) ไม่น้อยกว่า 1.4 เมกกะบิตต่อวินาที ซึ่งปัจจุบันหากเชื่อมต่ออินเทอร์เนตด้วยโมเด็ม จะมีแบนด์วิธ 56 กิโลบิตต่อวินาทีเท่านั้น เทคนิคการโพรเซสสัญญาณจึงถูกนำมาใช้เพื่อลดขนาดของสัญญาณลง ให้สามารถส่งผ่านช่องสัญญาณที่มีแบนด์วิธน้อยลงได้ เทคนิคนี้เรียกว่าการบีบอัด (Compression) หรือการเข้ารหัส (Encoding) สัญญาณเสียง การบีบอัดสัญญาณเสียงดิจิตอลคือการตัดข้อมูลส่วนที่ซ้ำซ้อนหรือส่วนที่ไม่ต้องการออกไป

การเข้ารหัสกระทำโดยเอนโค้ดเดอร์ (Encoder) ซึ่งมักจะเป็นโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ที่มีการ์ดเสียงติดตั้งอยู่ รูปที่ 2.1 แสดงถึงวิธีการเข้ารหัสของเอนโค้ดเดอร์ทั่วไป ซึ่งจะรับข้อมูลอินพุตเป็นบล็อกของสัญญาณเสียงที่ถูกแซมปลิ้ง (Sampled Audio Signal) บล็อกหนึ่งจะมีค่าแซมปลิ้ง (Samples) ตั้งแต่ 500-1500 ค่าต่อหนึ่งช่องสัญญาณ แล้วแต่เอนโค้ดเดอร์แต่ละตัว ตัวอย่างเช่น เอนโค้ดเดอร์ของเอ็มพีทรี (MPEG-1 audio layer-III) มี 576 ค่าแซมปลิ้งต่อหนึ่งช่องสัญญาณต่อหนึ่งบล็อก ส่วนเอาท์พุตคือผลที่ได้จากการบีบอัดของข้อมูลแต่ละบล็อก (เรียกเป็น "เฟรม") ซึ่งสามารถส่งไปหรือเก็บไว้ทำการถอดรหัสในภายหลัง



รูปที่ 2.1 การเข้ารหัสสัญญาณเสียงของเอนโค้ดเดอร์แบบทั่วไป

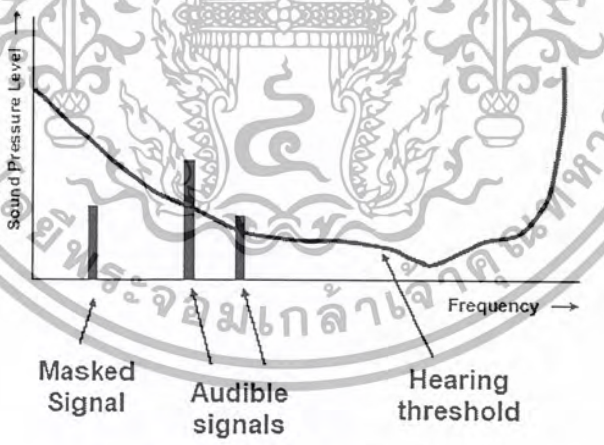
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 ไซโคอคูสติก โมเดล (Psychoacoustics Model)

แนวคิดพื้นฐานในการลดขนาดข้อมูลที่รับเข้ามาคือ ทำการตัดข้อมูลเสียงที่มนุษย์ไม่ได้ยินทิ้งไป การบีบอัดข้อมูลชนิดนี้ มักถูกเรียกว่า การเข้ารหัสแบบเพอร์เซปช่วล (Perceptual Encoding) ซึ่งต้องศึกษาการได้ยินเสียงของมนุษย์ วิธีการในการบีบอัดเสียงยังนำหลักการที่เสียงมีลักษณะบดบังกันเอง (Masking) มาใช้ด้วย ลักษณะที่ว่านี้เกิดขึ้นได้สามแบบดังนี้ คือ การถูกตัดด้วยเธรดโฮลด์ (Threshold Cut-Off), การบดบังทางความถี่ (Frequency Masking) และการบดบังชั่วคราว (Temporal Masking)

2.1.1.1 การถูกตัดด้วยเธรสโฮลด์

มนุษย์สามารถได้ยินเสียงหลายระดับ วัดเป็นระดับความเข้มเสียง (Sound Pressure Level (SPL)) ถ้าระดับระดับของเสียงต่ำกว่าเธรสโฮลด์การได้ยิน (Hearing Threshold) มนุษย์จะไม่ได้ยินเสียงนั้น รูปที่ 2.2 แสดงเธรสโฮลด์ในฟังก์ชันของความถี่ของเสียง จากรูปสังเกตได้ว่า ส่วนประกอบของเสียงที่อยู่ใต้เธรสโฮลด์จะไม่ได้ยิน หรือเป็นสัญญาณถูกมาสก์ไว้ (Masked Signal) ขณะที่ส่วนประกอบของเสียงที่อยู่เหนือเธรสโฮลด์จะเป็นเสียงที่มนุษย์ได้ยิน (Audible Signals)



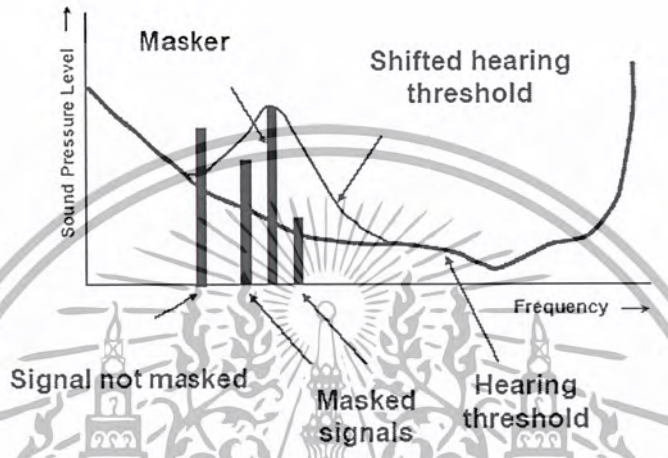
รูปที่ 2.2 เธรสโฮลด์ของการได้ยินมีจนถึงที่ความถี่ 20 กิโลเฮิร์ตซ์

2.1.1.2 การบดบังทางความถี่

แม้ว่าส่วนประกอบของเสียงบางอันจะอยู่เหนือเธรสโฮลด์ แต่มันก็อาจถูกบดบังโดยเสียงที่ดังกว่าและมีความถี่ใกล้เคียงกันได้ ลักษณะที่เกิดขึ้นแบบนี้เรียกว่า การบดบังทางความถี่ (Frequency Masking) หรือการบดบังของเสียงที่เกิดขึ้นพร้อมกัน (Simultaneous Masking) ปรากฏการณ์นี้เกิดขึ้นเนื่องจากแต่ละส่วนประกอบของเสียงจะคล้ายมีเงาแผ่ไปยังความถี่ใกล้เคียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

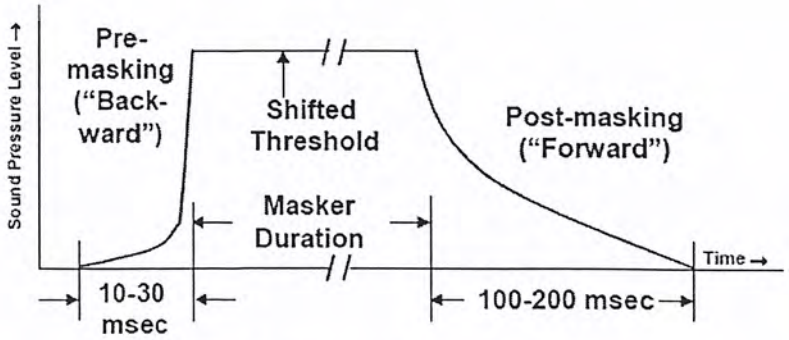
จากรูปที่ 2.3 สัญญาณที่ไปบดบังสัญญาณอื่น (Masker) จะมีเธรสโฮลด์เกิดขึ้น เนื่องจากแอมปลิจูด (Amplitude) ของตัวมันมีความแรงแผ่ไปยังความถี่ใกล้เคียง ทำให้ข้อมูลเสียงจริงที่ความถี่ใกล้เคียงที่อาจมีแอมปลิจูดน้อยกว่าชิฟท์เฮียริ่งเธรสโฮลด์ถูกมาสก์ เนื่องจากที่ความถี่หนึ่ง ๆ มนุษย์จะได้ยินเฉพาะเสียงที่ดังที่สุดเท่านั้น



รูปที่ 2.3 การบดบังทางความถี่

2.1.1.3 การบดบังแบบชั่วคราว

ลักษณะแบบนี้ เหมือนกับกรณีที่เสียงคล้ายมีเงาไปพาดทับเสียงข้างเคียงในแกนความถี่ ในกรณีนี้จะเป็นการที่เสียงเพิ่มความดังอย่างรวดเร็วจนสามารถบดบังเสียงที่เงียบกว่าซึ่งพึ่งจะหยุดไปได้ เหตุการณ์ที่เกิดขึ้นนี้เรียกว่า การบดบังชั่วคราว (Temporal Masking) สิ่งที่น่าสนใจอีกอย่างหนึ่งคือ ทั้งก่อนและหลังจากที่เสียงดังขึ้นอย่างรวดเร็วนั้น สัญญาณข้างเคียงเสียงนั้นอาจถูกบดบังได้ รูปที่ 2.4 จะแสดงถึงสถานการณ์ที่ถูกบดบังชั่วคราว สถานการณ์นี้เสียงที่อยู่ใต้เธรสโฮลด์นี้จะไม่ได้ยิน สังเกตได้ว่าช่วงก่อนที่จะถูกบดบัง (Pre-Masking) นั้นสั้นกว่าช่วงหลังถูกบดบัง (Post-Masking)



รูปที่ 2.4 การบดบังแบบชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 การวิเคราะห์เชิงความถี่ (Spectral Analysis)

สเปกตรัม (Spectrum) หรือสัญญาณในโดเมนความถี่ มีประโยชน์ในการวิเคราะห์ลักษณะทางความถี่และใช้ในการหาเธรสโฮลด์ 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

- เพื่อหาเธรสโฮลด์สำหรับอ้างอิงให้สัญญาณที่ต่ำกว่าเธรสโฮลด์ถูกตัดทิ้งไปได้
- เพื่อแปลงรูปแบบสัญญาณให้เป็นรูปแบบที่นำมาอ้างอิงกับเธรสโฮลด์ได้

การแปลงที่นิยมกันมากที่สุดคือ เอ็ฟเอ็ฟที (Fast Fourier Transform, FFT) เป็นการป้อนสัญญาณในโดเมนเวลา เอ็ฟเอ็ฟทีจะให้ผลในโดเมนความถี่ที่มีค่าเป็นจำนวนเชิงซ้อน เอนโค้ดเดอร์มักใช้เอ็ฟเอ็ฟทีเพื่อหาเธรสโฮลด์ การแปลงอีกแบบหนึ่งที่นิยมได้แก่ ดีซีที (Discrete Cosine Transform, DCT) ซึ่งให้ผลในโดเมนความถี่มีค่าเป็นจำนวนจริง ทั้งเอ็ฟเอ็ฟทีและดีซีทีต่างก็ประสบปัญหาสัญญาณผิดเพี้ยนที่ขอบของสัญญาณในแต่ละบล็อก อันเนื่องมาจากกระบวนการแปลงข้อมูล เพื่อแก้ปัญหานี้ สัญญาณอินพุตแต่ละบล็อกจึงต้องมีการซ้อนทับกัน (Overlapping and Windowing) ด้วยเหตุนี้เองที่เอนโค้ดเดอร์ส่วนใหญ่ที่ใช้เทคนิคการแปลง มักใช้ดีซีทีที่กำหนดให้สัญญาณอินพุตแต่ละบล็อกมีการซ้อนทับกันได้ ซึ่งเรียกว่าเอ็มดีซีที (Modified Discrete Cosine Transform, MDCT) เช่น การเข้ารหัสเอ็มพีทรี, เอ็มเปกทู เอซีซี (MPEG-2 AAC) และดอลบี้ เอซีทรี (Dolby AC-3)

การใช้ฟิลเตอร์จะป้อนบล็อกข้อมูลที่แซมปลิ้งแล้ว เข้าแบนด์พาสฟิลเตอร์ที่ย่านความถี่ต่างกัน สัญญาณอินพุตจึงถูกแยกเป็นสัญญาณที่ความถี่ย่อย ๆ หลังจากถูกกรองโดยฟิลเตอร์แล้วจะนำไปอ้างอิงกับเธรสโฮลด์เพื่อตัดส่วนที่ถูกบดบังโดยเธรสโฮลด์ออก โครงสร้างที่นิยมของการใช้ฟิลเตอร์มีสองแบบคือ ฟิลเตอร์แบบโพลีเฟส (Poly-Phase Filter Bank) และฟิลเตอร์แบบเวฟเล็ต (Wavelet Filter Bank) แบบโพลีเฟสจะใช้แบนด์พาสฟิลเตอร์หลายตัวที่มีแบนด์วิธเท่ากันมาขนานกันได้ผลเป็นค่าแซมปลิ้งที่น้อยลงเนื่องจากถูกแบ่งที่หลายความถี่ต่างกัน

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การลดสัญญาณรบกวน (Noise Allocations)

กลับไปที่บล็อกไดอะแกรม จากการทำวิเคราะห์สเปกตรัม (Spectrum Analysis) สัญญาณในแกนเวลาจะถูกแสดงในแกนความถี่ เป้าหมายต่อไปคือ การบีบอัดสัญญาณอินพุตโดยการตัดข้อมูลที่ไม่จำเป็นออกไป ซึ่งทำได้โดยใส่เธรสโฮลด์ลงในสัญญาณอินพุตนั้น เธรสโฮลด์การได้ยินจะเป็นตัวบ่งชี้ว่าเสียงที่ความถี่ไหนต้องแรงเท่าไรถึงจะได้ยิน อธิบายได้สองแง่ อย่างแรกคือเสียงที่อยู่ใต้เธรสโฮลด์ไม่จำเป็นต้องถูกส่งไป และอย่างที่สองสัญญาณรบกวน (Noise) ที่อยู่ใต้เธรสโฮลด์เป็นสิ่งที่เราไม่พึงปรารถนา

ในการเข้ารหัสสัญญาณเอนโค้ดเดอร์ต้องทำการควอนไทซ์ (Quantizing) คือ การแปลงสัญญาณเสียงที่แต่ละความถี่ให้เป็นเลขฐานสอง ยิ่งเอนโค้ดเดอร์ใช้จำนวนบิตในการควอนไทซ์มาก ความผิดพลาดก็จะยิ่งน้อย แต่ก็ใช้พื้นที่ในการเก็บข้อมูลมากขึ้นด้วย ในทางกลับกันถ้าใช้บิตน้อย ความผิดพลาดมาก แต่ก็ใช้พื้นที่น้อย อย่างไรก็ตามหากสัญญาณรบกวนอยู่ใต้เธรสโฮลด์ เสียงนั้นก็จะไม่ได้ยิน เพื่อที่จะลดขนาดพื้นที่สำหรับเก็บข้อมูล (หรือลดแบนด์วิธที่ใช้ในการส่งข้อมูล) เอนโค้ดเดอร์จะใช้จำนวนบิตเท่าที่เพียงพอต่อการเข้ารหัสแต่ละความถี่โดยไม่นำสัญญาณรบกวนที่มนุษย์ได้ยินมาทำการควอนไทซ์ เทคนิคนี้เรียกว่า เทคนิคการลดสัญญาณรบกวน

2.1.4 เทคนิคอื่น ๆ ในการบีบอัด

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

การทำนายเป็นเทคนิคหนึ่งที่นิยมใช้เวลาเข้ารหัสเสียงพูดและภาพ แนวคิดพื้นฐานคือ เราสามารถประมาณค่าของค่าแซมปลิ้งถัดไปจากค่าแซมปลิ้งที่แล้วได้ ดังนั้นแทนที่จะส่งค่าแซมปลิ้งถัดมา เอนโค้ดเดอร์จึงส่งเพียงแค่ส่วนต่างของค่าที่ประมาณกับค่าจริง ซึ่งจะใช้บิตจำนวนน้อยลง เมื่อดีโค้ดเดอร์ทำการถอดรหัส จะสร้างค่าประมาณขึ้นก่อนด้วยวิธีการเดียวกันกับเอนโค้ดเดอร์แล้วทำการปรับโดยใช้ส่วนต่างที่ได้รับจากเอนโค้ดเดอร์เอ็มเปกทูเอซีซี ใช้วิธีนี้เพื่อเพิ่มประสิทธิภาพในการบีบอัด

การคับปลิ้งจะถูกนำมาใช้ เมื่อสัญญาณเสียงจำนวนหลายสัญญาณต้องถูกเข้ารหัสและถอดรหัสแบบขนาน เช่นเสียงสเตอริโอ (Stereo) หรือเสียงรอบทิศทาง (Surround) ซึ่งประกอบด้วยหลายช่องสัญญาณและความต่างระหว่างช่องสัญญาณมักจะน้อย เวลาทำการรวมทุกช่องสัญญาณเข้าด้วยกันเพื่อทำการส่ง อาจใช้ความเหมือนกันนี้ให้เป็นประโยชน์ เช่น เอนโค้ดเดอร์อาจส่งผลรวมและผลต่างระหว่างสองช่องสัญญาณ ในกรณีที่สัญญาณเหมือนกันมาก ๆ ผลต่างจะมีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกือบเป็นศูนย์ อัลกอริธึมในการบีบอัดหนึ่งที่จะยกตัวอย่างต่อไปนี้ เป็นเทคนิคที่มีพื้นฐานการทฤษฎีการเข้ารหัสสารสนเทศชื่อ "ฮัฟแมน (Huffman)" ซึ่งเข้ารหัสโดยใช้จำนวนบิตน้อยที่สุดเป็นค่าสำหรับเข้ารหัสสัญญาณที่เกิดบ่อยครั้งที่สุด อย่างเช่น ถ้าเอาท์พุตของเอนโค้ดเดอร์ตัวหนึ่งมีค่า ๆ หนึ่งที่เกิดประมาณ 25 เปอร์เซ็นต์จากค่าที่เกิดขึ้นทั้งหมด เอนโค้ดเดอร์จะเข้ารหัสค่านี้ด้วยโค้ดเวิร์ด (Codeword) ที่สั้นกว่าโค้ดเวิร์ดอื่น

2.2 การเข้ารหัสสัญญาณเสียงแบบเอ็มพีทรี (MPEG-1 Audio Layer 3)

เอ็มเปก (Motion Picture Expert Group, MPEG) หรือกลุ่มผู้เชี่ยวชาญด้านภาพเคลื่อนไหว ก่อตั้งเมื่อปีค.ศ.1988 มีจุดประสงค์เพื่อสร้างมาตรฐานเปิดสำหรับการพัฒนาการเข้ารหัสสำหรับภาพเคลื่อนไหวและเสียง มาตรฐานการเข้ารหัสสัญญาณเสียงที่ถูกพัฒนาโดยกลุ่มเอ็มเปก ถูกใช้ในแอพพลิเคชันจำนวนมากประกอบด้วย การบรอดคาสท์เสียงดิจิตอล เสียงในระบบอินเทอร์เน็ต เสียงสำหรับเครื่องเล่นแบบพกพา ดีวีดี และเสียงที่เก็บในแหล่งเก็บเอ็มเปกรุ่นที่ได้รับความนิยมคือ เอ็มพีทรี ซึ่งถูกพัฒนาโดยห้องทดลองวิจัยเสียงของเยอรมันจากสถาบันฟรอนโฮเฟอร์ (Fraunhofer Institue)

เอ็มพีทรีเป็นชื่อย่อของเอ็มเปกวันออร์ทู ออดิโอเลเยอร์ทรี (MPEG-1/2, audio layer 3) ซึ่งเป็นซับเซตของการบีบอัดข้อมูลแบบเอ็มเปก มาตรฐานแบบเอ็มเปกวันออร์ทู ออดิโอเลเยอร์ทรีถูกพัฒนาขึ้นเพื่อเป็นมาตรฐานการลดขนาดข้อมูลในกรณีที่ข้อมูลเป็นเสียงอย่างเดียวไม่มีภาพ สามารถใช้ได้กับสัญญาณทั้งแบบโมโนและสเตอริโอ และมีเทคนิคที่เรียกว่า จอยท์สเตอริโอ (Joint Stereo) สามารถเข้ารหัสแยกช่องสัญญาณซ้ายและขวาได้ ประกอบด้วยสามโหมดการทำงาน เรียกเป็นสามเลเยอร์ ซึ่งจะเพิ่มความซับซ้อนและความสามารถขึ้นเรื่อย ๆ ตั้งแต่เลเยอร์หนึ่งถึงเลเยอร์สาม

2.2.1 ลักษณะของสัญญาณเสียงแบบเอ็มพีทรี

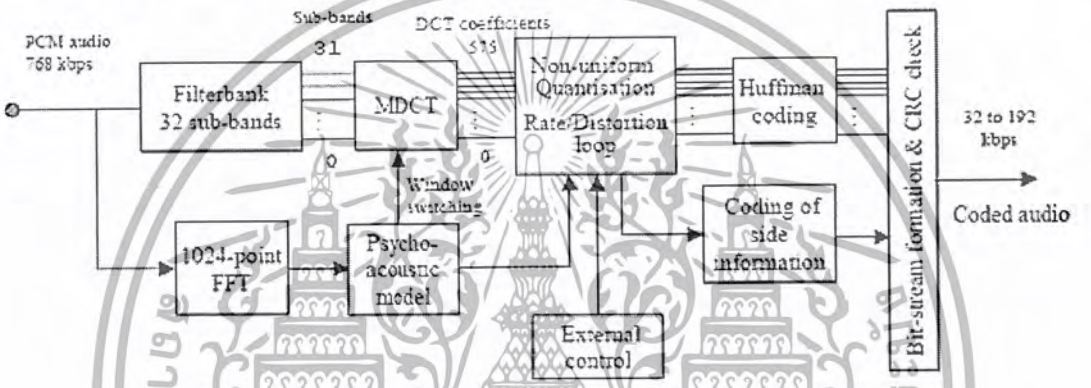
เอ็มพีทรีได้รับความนิยมเป็นอย่างมาก เนื่องจากมันสามารถลดขนาดข้อมูลเสียงได้มากโดยที่ยังคงคุณภาพเสียงได้ไม่ต่างจากแผ่นคอมแพคดิสก์ จากตารางที่ 2.1 มาตรฐานเอ็มพีทรีมีอัตราการลดขนาดข้อมูลมากที่สุดจากทั้งหมดสามเลเยอร์ คือ 1:10 ถึง 1:12 ทำให้ลดขนาดข้อมูลเสียงได้มากที่สุด ตัวอย่างเช่น เพลงจากซีดีความยาวสามนาทีจะมีขนาดประมาณ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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน	อัตราการลดขนาดข้อมูล	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 ตารางแสดงคุณสมบัติของมาตรฐานเอ็มเปกออดิโอแต่ละเลเยอร์

2.2.2 วิธีการเข้ารหัสสัญญาณเสียงแบบเอ็มพีทรี



รูปที่ 2.5 บล็อกไดอะแกรมของเอนโค้ดเดอร์ของเอ็มพีทรี

จากรูปที่ 2.5 คือ บล็อกไดอะแกรมของเอนโค้ดเดอร์ของเอ็มพีทรี ขั้นแรกผ่านสัญญาณเข้าฟิลเตอร์แบงค์แบบโพลีเฟส ซึ่งประกอบด้วยฟิลเตอร์จำนวน 32 ตัวต่อกันแบบขนาน เพื่อแบ่งสัญญาณออกเป็นย่านความถี่ย่อยทั้งหมด 32 ความถี่ จากนั้นผ่านสัญญาณที่ออกจากฟิลเตอร์แต่ละตัวเข้าเอ็มดีซีที เพื่อทำการแปลงสัญญาณในแกนเวลาให้เป็นสัญญาณในแกนความถี่ ที่แต่ละแบงค์จะได้เอาท์พุต 18 ช่องสัญญาณ จากทั้ง 32 แบงค์จะให้เอาท์พุตทั้งหมด 576 สัญญาณแซมปลิ้ง การรวมเอาฟิลเตอร์แบงค์กับเอ็มดีซีทีเข้าด้วยกันนี้ เพื่อทำการกรองและแปลงสัญญาณเรียกว่า "ไฮบริดจ์ฟิลเตอร์แบงค์" (Hybrid Filter Bank) หากมีการรบกวนสัญญาณ (Attack) เกิดขึ้น เอนโค้ดเดอร์จะทำการลดบล็อกของสัญญาณอินพุตลงจาก 576 เหลือ 192 แซมเปิ้ล ซึ่งวิธีนี้จะช่วยลดการเกิดเอคโค่ (echo) ได้ จากนั้นเอาท์พุตที่ออกจากเอ็มดีซีที จะถูกใส่เธรสโฮลด์ในแกนความถี่ เพื่อมาสก์สัญญาณที่ไม่ได้ยินและสัญญาณรบกวนออกไป ทำการควอนไทซ์ เข้ารหัสฮัฟแมน จากนั้นใส่ข้อมูลควบคุม (Side Information) สร้างเป็นแพ็คเกตที่มีรูปแบบพร้อมสำหรับขั้นตอนการส่ง ส่วนควบคุมจะทำหน้าที่ควบคุมการเข้ารหัสให้เหมาะสมกับสภาวะภายนอก เช่น เมื่อข้อมูลมีจำนวนมาก แบนด์วิธไม่เพียงพอ ส่วนควบคุมจะสั่งลดระดับการควอนไทซ์ให้หยาบขึ้น ทำให้จำนวนบิตข้อมูลน้อยลง เรียกว่าการควอนไทซ์แบบไม่ตายตัว (Non-Uniform Quantization)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากดีโค้ดเดอร์ไม่มีการคำนวณค่าเธรสโฮลด์ จึงมีความซับซ้อนน้อยกว่าเอนโค้ดเดอร์ สิ่งที่ดีโค้ดเดอร์ทุกตัวต้องทำคือ แกะแพ็คเกต นำส่วนข้อมูลมาถอดรหัสฮัฟแมน ถอดสเกลแกนความถี่ แล้วนำสัญญาณไปเข้าฟิลเตอร์แบงค์ย้อนกลับ (Inverse Filter Bank) ได้กลับเป็นสัญญาณเริ่มต้นอีกครั้ง

2.3 โปรโตคอลที่ใช้ในการส่งข้อมูลแบบเรียลไทม์

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

1. ไม่รองรับรูปแบบที่ใช้ไทม์สแตมป์ (Timestamp) และเพลย์แบ็ค (Playback)
2. ไม่ยอมให้เกิดการสูญหายของข้อมูลขึ้น คือ เมื่อข้อมูลบางส่วนหายไป จะสั่งให้ทำการส่งใหม่ทันที แต่ในการสื่อสารแบบเรียลไทม์ เราไม่สามารถสั่งให้ทำการส่งข้อมูลใหม่ได้ เพราะจำเป็นต้องใช้ความต่อเนื่องของข้อมูล
3. ไม่รองรับรูปแบบการสื่อสารแบบมัลติคาสท์ (Multicast)
4. ใช้เวลานานในการส่งข้อมูลเนื่องจากต้องมีการทำทรีเวย์แฮนด์เชค (Three Way Handshake) ทุกครั้งที่ส่งข้อมูลแต่ละแพ็คเกต ในขณะที่การสื่อสารแบบเรียลไทม์ต้องการความเร็วในการรับส่งข้อมูล

จากการศึกษาโปรโตคอลยูดีพีทำให้ทราบว่า โปรโตคอลยูดีพีเหมาะสมสำหรับการสื่อสารแบบเรียลไทม์มากกว่า เนื่องจากยูดีพีรองรับการสื่อสารแบบมัลติคาสท์ และยอมให้ข้อมูลบางส่วนสูญหายได้ โดยไม่สั่งให้ทำการส่งข้อมูลนั้น ๆ ใหม่ อย่างไรก็ตาม ยูดีพียังไม่รองรับรูปแบบที่ใช้การไทม์สแตมป์, การเรียงลำดับข้อมูล (Sequencing) และการรวมสัญญาณจากหลาย ๆ แหล่งกำเนิด (Mixing)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

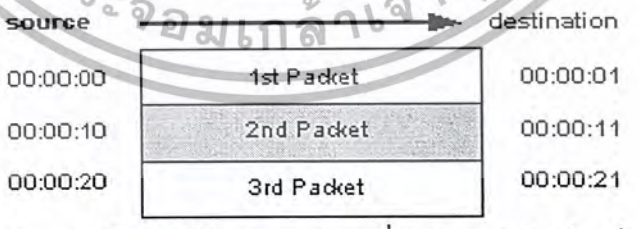
2.3.1 ลักษณะของการสื่อสารแบบเรียลไทม์

การสื่อสารแบบเรียลไทม์ มักจะเป็นการสื่อสารในรูปแบบของข้อมูลมัลติมีเดีย ซึ่งขนาดของข้อมูลจะใหญ่มากและต้องการความต่อเนื่องของข้อมูลในการส่ง การสื่อสารบนเครือข่ายอินเทอร์เนตทั่วไปไม่สามารถรองรับการสื่อสารดังกล่าวได้ ดังนั้นจำเป็นจะต้องมีรูปแบบที่รองรับการสื่อสารแบบเรียลไทม์ขึ้น

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

2.3.1.1 ความสัมพันธ์ทางเวลา (Time Relationship)

ในการส่งข้อมูลแบบเรียลไทม์ จำเป็นจะต้องคำนึงถึงความสัมพันธ์ของเวลาในการสื่อสารแต่ละครั้ง ยกตัวอย่างเช่น การส่งข้อมูลวิดีโอผ่านอินเทอร์เนต เซิร์ฟเวอร์จะแปลงสัญญาณวิดีโอให้อยู่ในรูปแบบของสัญญาณดิจิตอล จากนั้นทำการแบ่งข้อมูลดังกล่าวออกเป็นแพ็คเกต (packet) ถ้ามีข้อมูลทั้งหมด 3 แพ็คเกต แต่ละแพ็คเกตมีความยาวเท่ากับ 10 วินาที ดังนั้น ถ้าแพ็คเกตแรกเริ่มส่งที่เวลา 00.00.00 แพ็คเกตที่สองต้องส่งต่อที่เวลา 00.00.10 และแพ็คเกตที่สามที่เวลา 00.00.20 เป็นต้น และในการข้อมูลใช้เวลา 1 วินาที ดังนั้นที่ฝั่งรับ จะรับแพ็คเกตแรกที่เวลา 00.00.01 แพตเก็ตที่สองที่เวลา 00.00.11 แพ็คเกตที่สามที่เวลา 00.00.21 ตามลำดับ ดังรูป

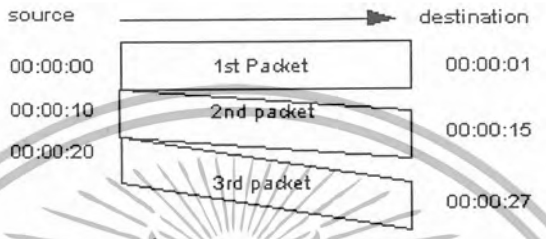


รูปที่ 2.6 ความสัมพันธ์เชิงเวลาของการสื่อสารแบบเรียลไทม์

แต่ในทางปฏิบัติการส่งจะต้องเกิดความล่าช้า (Delay) ขึ้น โดยในการส่งแต่ละครั้งจะเกิดความล่าช้าที่แตกต่างกันออกไปเช่น ในการส่งแพ็คเกตที่ 1 ฝั่งรับเริ่มรับข้อมูลที่เวลา 00.00.01 หมายความว่าเกิดความล่าช้า 1 วินาที แพ็คเกตที่ 2 รับข้อมูลที่เวลา 00.00.15 หมายความว่าเกิดความล่าช้า 5 วินาที แพ็คเกตที่ 3 รับข้อมูลที่เวลา 00.00.27 เกิดความล่าช้า 7 วินาที ถ้าที่ฝั่งรับเริ่มอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

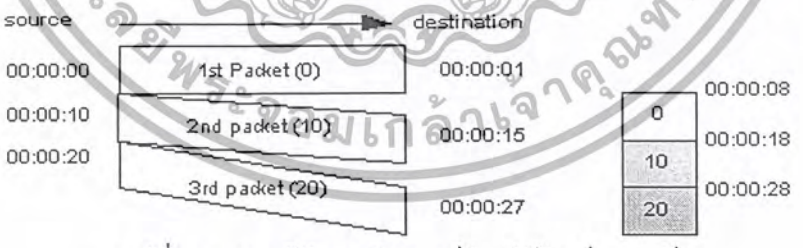
ข้อมูลของแพ็คเกตแรกที่เวลา 00.00.01 เมื่อเวลา 00.00.11 ต้องอ่านแพ็คเกตที่สอง แต่ปรากฏว่าข้อมูลยังมาไม่ถึง ลักษณะเช่นนี้จะทำให้เกิดช่องว่างระหว่างการอ่านแพ็คเกตแรกกับแพ็คเกตต่อไป ช่องว่างดังกล่าวเรียกว่า "จิตเตอร์ (Jtter)" หรือความล่าช้าในระหว่างการรับข้อมูลแต่ละแพ็คเกตของรูปแบบการส่งข้อมูลแบบเรียลไทม์



รูปที่ 2.7 ปัญหาจิตเตอร์

2.3.1.2 ไทม์แสตมป์

การแก้ปัญหาที่เกิดจากจิตเตอร์แบบหนึ่งคือ การกำหนดไทม์สแตมป์หรือการกำหนดว่าจะเริ่มอ่านข้อมูลของแพ็คเกตนี้ที่เวลาใด หรือจะต้องอ่านข้อมูลของแพ็คเกตต่อไปอีกเมื่อใด เช่น แพ็คเกตแรกมีค่าไทม์สแตมป์เท่ากับ 0 แพ็คเกตที่สองมีค่าไทม์สแตมป์ 10 แพ็คเกตที่สามมีค่าไทม์สแตมป์ 20 ถ้าฝั่งรับเริ่มอ่านข้อมูลแรกที่เวลา 00.00.08 เพราะฉะนั้นอีก 10 วินาทีจึงเริ่มอ่านข้อมูลแพ็คเกตต่อไป คือเวลา 00.00.18 และแพ็คเกตที่สามที่เวลา 00.00.28 ดังรูป



รูปที่ 2.8 การแก้ปัญหาจิตเตอร์โดยใช้ไทม์สแตมป์

2.3.1.3 เพลย์แบ็คบัฟเฟอร์ (Playback Buffer)

วิธีแก้ปัญหาที่เกิดจากจิตเตอร์อีกวิธีหนึ่งคือ การสร้างที่พักข้อมูลก่อนที่จะทำการอ่านข้อมูล (Playback Buffer) เมื่อเริ่มต้นกระบวนการส่งข้อมูล ฝั่งรับจะรอให้ข้อมูลของแพ็คเกตเข้ามาไว้ในบัฟเฟอร์จนเต็ม แล้วจึงเริ่มอ่านข้อมูลนั้น ๆ แพ็คเกตถัดไปก็จะถูกเก็บไว้ในบัฟเฟอร์เพื่อรอการเรียกอ่านต่อ ๆ ไป ยกตัวอย่างเช่น ทางฝั่งรับมีบัฟเฟอร์ที่สามารถจุแพ็คเกตความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยาว 7 วินาที เมื่อแพ็คเกตแรกที่มีความยาว 10 วินาทีมาถึง ข้อมูล 7 วินาทีแรกก็จะถูกเก็บไว้ในบัฟเฟอร์ ระบบก็จะเริ่มอ่านข้อมูลนั้น เมื่อเวลาผ่านไป 1 วินาที ข้อมูลของแพ็คเกตเดิมที่ช่วงเวลา 8 วินาทีก็จะถูกเติมมาในบัฟเฟอร์ ดังนั้นเมื่อเวลาผ่านไป 7 นาที ก็จะเหลือข้อมูลของแพ็คเกตอีก 3 วินาที ถ้าแพ็คเกตที่ 2 ล่าช้าไปไม่เกิน 3 วินาทีก็จะสามารถอ่านข้อมูลได้อย่างต่อเนื่อง

2.3.1.4 การเรียงลำดับ (Ordering)

การจะอ่านข้อมูลที่ถูกแบ่งและส่งมาเรื่อย ๆ บางครั้งข้อมูลอาจจะถูกส่งถึงในเวลาที่แตกต่างกัน ดังนั้นการอ่านจะต้องอ่านจากแพ็คเกตแรก และเรียงลำดับไปเรื่อย ๆ จึงต้องมีส่วนประกอบที่สำคัญอีกอย่างหนึ่งในระบบการสื่อสารแบบเรียลไทม์คือ เลขลำดับ (Sequence Number) ข้อดีอีกอย่างของเลขลำดับคือ ทำให้สามารถตรวจสอบได้ว่าข้อมูลที่ถูกส่งมาได้รับครบหรือไม่ ยกตัวอย่างเช่น ถ้าแพ็คเกตที่หนึ่ง สองและสามมีค่าไทม์สแตมป์เป็น 20 ,10 ,0 ถ้าแพ็คเกตที่สองสูญหายไป ทางฝั่งรับจะไม่สามารถตรวจสอบได้ว่าแพ็คเกตที่สองหายไปแล้ว แต่จะคิดว่าแพ็คเกตแรกมีความยาวเท่ากับ 20 หน่วยเวลา ดังนั้นจะต้องเพิ่มส่วนของเลขลำดับขึ้นมาเพื่อจะได้ทราบว่าแพ็คเกตที่สองได้สูญหายไปแล้ว

2.3.1.5 การแปลงสัญญาณ (Translation)

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

2.3.1.6 การผสมสัญญาณ (Mixing)

ถ้ามีการส่งข้อมูลจากหลายแหล่งพร้อมกัน เช่น ส่งสัญญาณวิดีโอพร้อมกับสัญญาณเสียง หรือการประชุมทางไกลผ่านอินเทอร์เนต (Video Conference) ทำให้มีสตรีมข้อมูลจำนวนมากถูกส่งออกมาพร้อมกัน ทำให้ระบบเกิดความคับคั่ง วิธีการผสมสัญญาณ (Mixing) จึงเกิดขึ้นเพื่อลดความคับคั่งของข้อมูล โดยใช้วิธีรวมสตรีมหลาย ๆ สตรีมเข้าด้วยกัน แล้วส่งออกเป็นหนึ่งสตรีมเสมือนเป็นการส่งข้อมูลจากแหล่งกำเนิดข้อมูลตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 โปรโตคอลอาร์ทีพี (RTP Protocol)

โปรโตคอลอาร์ทีพีถูกสร้างโดยองค์กรไออีทีเอฟในอาร์เอฟซี1889 ขณะนี้อาร์ทีพีได้รับการยอมรับว่าเป็นมาตรฐานสำหรับการส่งข้อมูลมัลติมีเดียแบบเรียลไทม์ สามารถทำงานได้ทั้งยูนิคาสท์และมัลติคาสท์ ข้อมูลที่ส่งไปจะถูกดูแลโดยโปรโตคอลอีกชนิดหนึ่งที่ทำงานร่วมกับอาร์ทีพีที่มีชื่อว่า โปรโตคอลอาร์ทีซีพี (RTCP Protocol) ซึ่งจะกล่าวถึงในหัวข้อ 2.3.3

ข้อมูลจากแพ็คเกตอาร์ทีพีที่มาถึงฝั่งผู้รับ จะถูกดูแลโดยโปรโตคอลอาร์ทีซีพี ซึ่งทำหน้าที่คอยแจ้งให้ชั้นอาร์ทีพีทราบ เพื่อปรับการเข้ารหัสและตัวแปรในการส่ง (Transmission Parameter) ให้เหมาะสม ตัวอย่างเช่น ถ้าตัวแปรในชั้นอาร์ทีซีพีตรวจเจอว่ามีแพ็คเกตสูญหายไปมาก มันอาจแจ้งให้ชั้นอาร์ทีพีปรับลดค่าอัตราการส่งข้อมูล ถึงแม้ว่าอาร์ทีพีจะเหมาะสมในการเล่นมีเดียแบบเรียลไทม์ แต่อาร์ทีพีไม่ได้มีกลไกในการรับประกันการส่งใด ๆ ทั้งสิ้น กลไกนี้จะขึ้นอยู่กับบริการในชั้นล่างลงมา มันถือว่าเครือข่ายนั้นเชื่อถือได้และแพ็คเกตที่ส่งนั้นมาถึงที่หมายถูกต้องตามลำดับ

อาร์ทีพีถูกออกแบบมาโดยจุดประสงค์หลักเพื่อการประชุมผ่านเครือข่าย และยังให้บริการสำหรับข้อมูลต่อเนื่อง การจำลองการโต้ตอบแบบกระจาย (Interactive Distributed Simulation) และแอพพลิเคชันสำหรับการวัดคุม (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 CC : Contributor count
P : Padding M : Marker

รูปที่ 2.9 รูปแบบส่วนหัวของแพ็คเกตอาร์ทีพี

จากรูปที่ 2.9 รูปแบบส่วนหัว (Header) ของแพ็คเกตอาร์ทีพีประกอบด้วยเขตข้อมูลต่าง ๆ ดังนี้

1. "Version" (V) เวอร์ชันของโปรโตคอลอาร์ทีพีในปัจจุบันที่ใช้อยู่คือ เวอร์ชัน2
2. "Padding" (P) หากบิตนี้ถูกเซ็ตให้มีค่าเท่ากับ 1 นั่นหมายความว่า ในส่วนท้ายของแพ็คเกตจะมีส่วนของแพดดิ้งต่อท้ายมาด้วย และไบต์สุดท้ายจะเป็นจำนวนไบต์ของแพดดิ้งทั้งหมด โดยปกติแล้วแพดดิ้งจะถูกใช้ในกรณีที่แพ็คเกตถูกเข้ารหัสมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. "Extension" (X) ถ้าบิตนี้ถูกเซ็ตให้มีค่าเท่ากับ 1 หมายความว่า จะมีส่วนของ Extension header เพิ่มเข้ามาระหว่างส่วนของเฮดเดอร์และข้อมูล
4. "Contributor Count" (CC) คือจำนวนของแหล่งข้อมูลในแพ็คเกต ในกรณีที่มีเดียสตรีมมีหลายแหล่งที่มาเช่น ภาพและเสียง ข้อมูลในฟิลด์นี้จะมีค่าได้สูงสุดเท่ากับ 15 เนื่องจากมีจำนวนบิตเท่ากับ 1 บิต
5. "Marker" (M) บิตนี้จะถูกเซ็ตให้มีค่าเป็น 1 เมื่อบิตนั้นมีความจำเป็นที่จะต้องแสดงให้แอพพลิเคชันเห็นเป็นพิเศษเช่น แพ็คเกตนี้คือข้อมูลชุดสุดท้ายของสตรีม
6. "Payload Type" เขตข้อมูลนี้มีความยาว 8 บิต จะแสดงชนิดข้อมูลของแพ็คเกตว่าถูกเข้ารหัสมาในรูปแบบใด และจะถูกส่งให้แอพพลิเคชันใดทำการอ่านต่อไป

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 ชนิดของข้อมูลในแพ็คเกตอาร์ทีพี

7. "Sequence Number" เขตข้อมูลนี้มีความยาว 16 บิต จะแสดงว่า แพ็คเกตอาร์ทีพีนี้มีหมายเลขใด โดยแพ็คเกตแรกเลขลำดับนี้จะถูกสุ่มขึ้นมาและจะเพิ่มขึ้นทีละ 1 สำหรับแพ็คเกตชุดเดียวกันที่ถูกส่งมาเป็นลำดับถัดไป ใช้สำหรับตรวจว่าแพ็คเกตที่ถูกส่งมาว่าสูญหายหรือเกินมาหรือไม่
8. "Timestamp" เขตข้อมูลนี้มีความยาว 32 บิต ใช้แสดงถึงความสัมพันธ์ในเชิงเวลาของแพ็คเกตชุดเดียวกัน คือจะแสดงว่าแพ็คเกตนั้นอยู่ในช่วงเวลาใดของข้อมูลชุดเดียวกัน ยกตัวอย่างเช่น ถ้ามีแพ็คเกต 3 แพ็คเกต แต่ละแพ็คเกตมีความยาว 10 วินาที ดังนั้นถ้าค่าไทม์สแตมป์ของแพ็คเกตแรกมีค่าเท่ากับ 0 แพ็คเกตต่อมาจึงมีค่าเท่ากับ 10 และ 20 ตามลำดับ เมื่อรับแพ็คเกตแรกมาแล้วจะอ่านค่าไทม์สแตมป์ไว้ และเมื่อแพ็คเกตต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ถูกส่งมา ก็จะอ่านค่าไทม์สแตมป์แล้วนำมาคำนวณว่า แพ็คเกตแรกมีความยาวเท่าใด ในที่นี้ แพ็คเกตที่สองมีค่าต่างจากแพ็คเกตแรก 10 เพราะฉะนั้น แพ็คเกตแรกมีความยาว 10 หน่วยเวลา และในขณะที่อ่านนี้อยู่ในช่วงใดของข้อมูล โดยค่าของช่วงเวลาดังกล่าว จะขึ้นอยู่กับแอพพลิเคชันนั้น ๆ ว่าจะกำหนดช่วงเวลาให้มีความยาวเท่าใด โดยทั่วไปแล้วข้อมูลเสียงจะถูกสร้างขึ้นด้วยอัตราเร็ว 160 ไบต์ต่อ 1 ช่วงเวลา ดังนั้น การอ่านข้อมูลชนิดนี้จะต้องอ่านในช่วงเวลาเดียวกัน ดังนั้นค่าไทม์สแตมป์ของข้อมูลชนิดนี้ จึงต้องเพิ่มขึ้นทีละ 160 ในแต่ละแพ็คเกต

9. "Synchronization Source Identifier" ถ้าแพ็คเกตที่ถูกส่งมาจากแหล่งกำเนิดสัญญาณแหล่งเดียว ค่าของเขตข้อมูลนี้จะเป็นหมายเลขของแหล่งกำเนิดนั้น ๆ (ขนาด 32 บิต) แต่ถ้าข้อมูลมาจากหลาย ๆ แหล่ง แล้วทำการรวมสัญญาณนั้น ๆ ให้อยู่ในแพ็คเกตเดียวกัน ค่าของเขตข้อมูลจะเป็นหมายเลขของแหล่งกำเนิดใหม่ที่เกิดจากการรวมสัญญาณ โดยค่าดังกล่าวจะถูกสุ่มออกมาจากแหล่งกำเนิด และทางฝั่งรับจะทราบได้ว่าข้อมูลนี้เกิดจากแหล่งกำเนิดมากกว่าหนึ่งแหล่งและต้องอ่านในเวลาเดียวกัน (ข้อมูลจากหลายแหล่งจะมีเลขลำดับเดียวกัน)
10. "Contributor Identifier" เขตข้อมูลนี้มีความยาว 32 บิต มีไว้เพื่อบอกว่าข้อมูลนี้เกิดจากแหล่งกำเนิดข้อมูลใดบ้าง ใช้ในกรณีที่ข้อมูลมาจากแหล่งกำเนิดมากกว่าหนึ่งและข้อมูลนั้นถูกรวมกับข้อมูลจากแหล่งอื่น ๆ

2.3.3 โปรโตคอลอาร์ทีซีพี (RTCP Protocol)

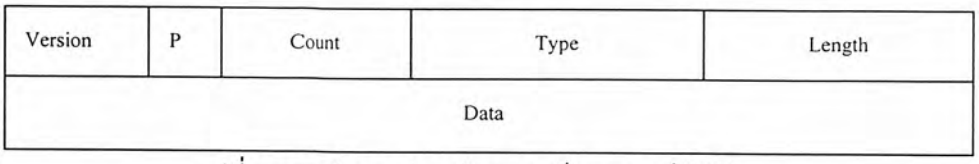
มีหน้าที่หลักคือ การควบคุมการส่งแพ็คเกตของการสื่อสารแบบเรียลไทม์ โดยใช้หลักการทำงานเช่นเดียวกับการส่งแพ็คเกตข้อมูลทั่ว ๆ ไป หน้าที่หลักของอาร์ทีซีพี มีดังนี้คือ

1. ตรวจสอบคุณภาพของการสื่อสารและควบคุมความคับคั่งของข้อมูล
2. ระบุข้อมูลที่จำเป็นสำหรับการส่งสตรีม
3. จัดการเกี่ยวกับขนาดของเซสชัน

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

55720



รูปที่ 2.10 รูปแบบส่วนหัวของแพ็คเกตอาร์ทีซีพี

จากรูปที่ 2.10 รูปแบบส่วนหัว (Header) ของแพ็คเกต RTCP เขตข้อมูลที่ต่างจาก RTP มีดังนี้

1. "Count" เขตข้อมูลนี้มีขนาด 5 บิต โดยที่ค่าในเขตข้อมูลนี้ แสดงถึงจำนวนของรายงานการรับแพ็คเกต (reception report) ที่บรรจุอยู่ในแพ็คเกตนี้ หากชนิดของแพ็คเกต (เขตข้อมูล "Type") เป็นรายงานสถานะของผู้รับ (RR, Receiver Report)
2. "Type" เขตข้อมูลนี้มีขนาด 8 บิต ค่าในเขตข้อมูลนี้แสดงถึงชนิดของแพ็คเกตอาร์ทีซีพี ซึ่ง สามารถดูชนิดของแพ็คเกตอาร์ทีซีพีทั้งหมดได้จากตารางที่ 2.3

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

ตารางที่ 2.3 ชนิดของแพ็คเกตอาร์ทีซีพี

3. "Length" เขตข้อมูลนี้มีขนาด 16 บิต ค่าในเขตข้อมูลนี้จะแสดงถึงขนาดของแพ็คเกตอาร์ทีซีพี รวมส่วนหัวและส่วนแพดดิ้งที่ต่อท้ายมากับแพ็คเกตด้วย เพื่อป้องกันไม่ให้เกิดแพ็คเกตที่มีความยาวเท่ากับศูนย์ และยังป้องกันการเกิดการวนรอบจำกัด (Finite Loop) ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 โปรโตคอลอาร์ทีเอสพี (RTSP Protocol)

โปรโตคอลอาร์ทีเอสพีนี้เป็นโปรโตคอลที่ใช้ในการรับส่งข้อมูลมัลติมีเดียระหว่างเซิร์ฟเวอร์กับคอมพิวเตอร์ปลายทาง ซึ่งจะทำให้การรับส่งข้อมูลต่อเนื่องผ่านอินเทอร์เนต โดยทางเซิร์ฟเวอร์ด้านผู้ส่ง สามารถส่งข้อมูลไปให้ผู้รับปลายทางเพียงคนเดียว หรือส่งไปให้ผู้รับหลาย ๆ คนในลักษณะเป็นกลุ่มก็ได้ ซึ่งอาร์ทีเอสพีถูกกำหนดให้เป็นโปรโตคอลที่นำไปใช้ในอินเทอร์เนตเอ็นจิเนียร์ทาสก์ฟอร์ซ หรือไออีทีเอฟ (Internet Engineer Task Force, IETF) ในเดือนเมษายน ปี ค.ศ. 1998 เรียกว่า อาร์เอฟซี2326 (RFC2326)

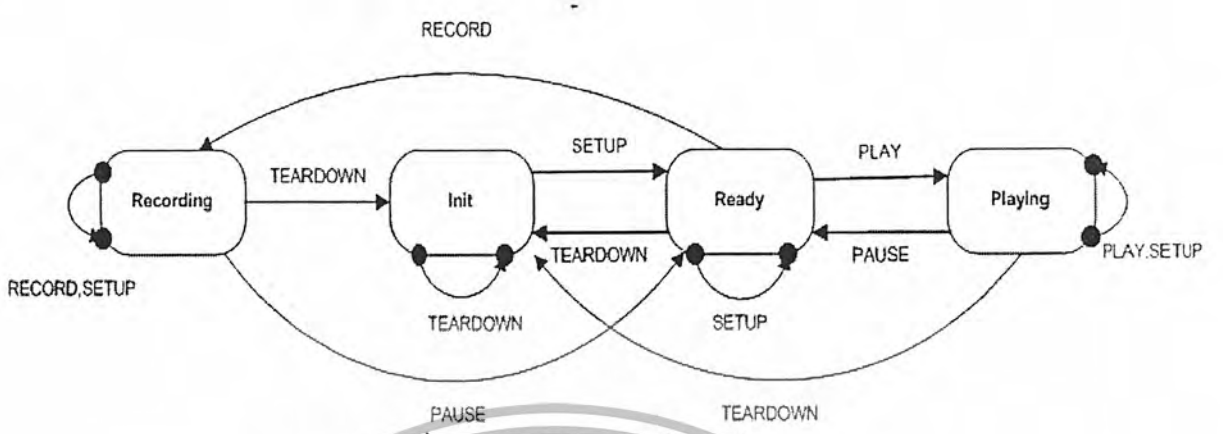
อาร์ทีเอสพีเป็นการกำหนดมาตรฐานโปรโตคอลที่สำคัญมากในการรับส่งข้อมูลมัลติมีเดียผ่านอินเทอร์เนต เนื่องจากการรับส่งข้อมูลมัลติมีเดียในแบบต่อเนื่องนั้น มีส่วนต่าง ๆ ที่เกี่ยวโยงกันถึง 3 ส่วน คือ เซิร์ฟเวอร์ที่เก็บข้อมูล, เอนโค้ดเดอร์ที่ใช้เข้ารหัสข้อมูลและผู้รับข้อมูลหรือเพลย์เยอร์ (Player) ตัวเข้ารหัสข้อมูลนั้นจะต้องเข้ารหัสข้อมูลมัลติมีเดียเก็บลงในไฟล์ โดยมีฟอร์แมตที่เซิร์ฟเวอร์ที่เรียกใช้งานได้ และเมื่อเซิร์ฟเวอร์ต้องการส่งข้อมูลนี้ไปให้ผู้รับ ก็จะต้องใช้โปรโตคอลรับส่งข้อมูลอย่างต่อเนื่องที่ผู้รับเข้าใจ และสามารถรับข้อมูลได้อย่างถูกต้อง จากนั้นเมื่อได้รับข้อมูลมาแล้วก็จะต้องถอดรหัสข้อมูลออกแสดงผลได้ โดยใช้มาตรฐานเดียวกันกับตัวเข้ารหัส การทำงานทั้งหมดจึงผูกพันธุ์กัน ซึ่งอาร์ทีเอสพีจะอยู่ในส่วนโปรโตคอลที่ใช้รับส่งข้อมูลจากเซิร์ฟเวอร์ไปให้ผู้รับนั่นเอง

แม้ว่าอาร์ทีเอสพีจะมีความสำคัญในการรับส่งข้อมูลมัลติมีเดียผ่านอินเทอร์เนต ก็ตาม แต่ก็ไม่ใช่สิ่งเดียวที่ทำให้การรับส่งข้อมูลสมบูรณ์ได้ เรายังต้องการฟอร์แมตมาตรฐานของไฟล์ที่ใช้เก็บข้อมูลมัลติมีเดียอีกด้วยเช่น แอคทีฟสตรีมมิ่งฟอร์แมต (Active Streaming Format, ASF) ของไมโครซอฟต์ ควิกไทม์ หรืออื่น ๆ เพื่อเก็บข้อมูล รวมถึงมาตรฐานการเข้ารหัสข้อมูลเช่น เอ็มเปกสำหรับใช้เข้ารหัสข้อมูลมัลติมีเดียเก็บลงไฟล์อีกด้วย

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

รูปที่ 2.11 แสดงถึงสถานะต่าง ๆ ของเซิร์ฟเวอร์ ดูตารางที่ 2.4 แสดงเมธอดต่าง ๆ ของโปรโตคอลอาร์ทีเอสพีประกอบ เนื่องจากมีความสัมพันธ์ในการเปลี่ยนสถานะของเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 สถานะของเซิร์ฟเวอร์

เมธอด	คำอธิบาย
Options	ขอดูเมธอดทั้งหมดที่มี
SETUP	สร้างช่องทางและจัดสรรทรัพยากรสำหรับการส่ง
ANNOUNCE	เปลี่ยนแปลงคำอธิบายของมัลติมีเดีย
DESCRIBE	ขอดูคำอธิบายของมัลติมีเดีย
PLAY	เริ่มเล่นมัลติมีเดีย
RECORD	เริ่มบันทึกมัลติมีเดีย
REDIRECT	ย้ายไคล์เอนท์ไปที่เซิร์ฟเวอร์ตัวใหม่
PAUSE	หยุดการส่งข้อมูลชั่วคราว แต่ยังรักษาสเตทไว้
SET-PARAMETER	ควบคุมการส่ง
TEARDOWN	ยกเลิกสเตททั้งหมด

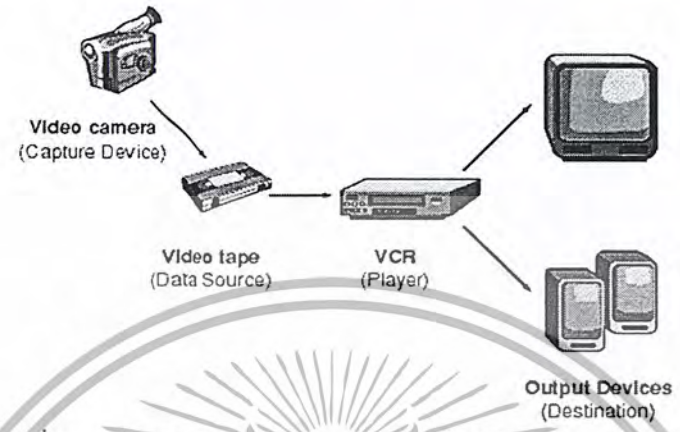
ตารางที่ 2.4 เมธอดของโปรโตคอลอาร์ทีเอสพี

2.4 จาวามีเดียเฟรมเวิร์ก (Java Media Framework, JMF)

การทำงานของจาวามีเดียเฟรมเวิร์กหรือเจเอ็มเอฟ มีหลักการทำงานคล้ายกับการทำงานของระบบวีดีโอ ตัวอย่างเช่น เมื่อเราต้องการดูหนังจากเครื่องเล่นวีดีโอ เราจะต้องใส่ม้วนเทปเข้าไปในเครื่องเล่นวีดีโอ จากนั้นเครื่องเล่นวีดีโอจะทำการอ่านข้อมูลจากเทป แล้วแสดงข้อมูลในรูปของสัญญาณภาพออกทางจอโทรทัศน์และสัญญาณเสียงออกทางลำโพงตามลำดับ เจเอ็มเอฟก็ใช้หลักการเช่นเดียวกับระบบวีดีโอทั่วไป เริ่มตั้งแต่การจับภาพและเสียงโดยใช้กล้องวีดีโอ (Capture Device) และนำข้อมูลที่ได้มาเก็บไว้ดาต้าซอร์ส ซึ่งเปรียบเสมือนม้วนวีดีโอเทป จากนั้นดาต้าซอร์สจะถูกนำไปประมวลผลโดยเพลย์เยอร์ เพื่อให้สามารถแสดงผลลัพธ์ออกทางอุปกรณ์แสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Output Device) ต่าง ๆ ซึ่งการทำงานของระบบวีดีโอและการทำงานของเจเอ็มเอฟสามารถแสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 เปรียบเทียบการทำงานของกล้องวีดีโอกับเจเอ็มเอฟ

2.4.1 การจับข้อมูลมัลติมีเดีย

อุปกรณ์ที่ทำการจับข้อมูลมัลติมีเดีย จะทำหน้าที่เป็นแหล่งของข้อมูลมีเดียสำหรับการทำงานของเจเอ็มเอฟ และข้อมูลที่ได้จากอุปกรณ์จับข้อมูลมัลติมีเดียในเจเอ็มเอฟจะแทนด้วยซื่อดาต้าซอร์ส (DataSources) ตัวอย่างของอุปกรณ์จับข้อมูลมัลติมีเดียเช่น การ์ดเสียงเป็นอุปกรณ์ที่ใช้ในการจับเสียงจากไมค์โครโฟนแล้วแปลงมาเป็นข้อมูล ในการจับข้อมูลมีเดียจากอุปกรณ์จับข้อมูลมัลติมีเดียที่เราต้องการใช้ เราจะต้องดึงเอามีเดียโลเคเตอร์ (MediaLocator) ของอุปกรณ์นั้น ๆ มาจากออบเจคท์ "CaptureDeviceInfo" ซึ่งได้มาจากคลาส "CaptureDeviceManager" จากนั้นจึงสามารถนำมีเดียโลเคเตอร์ที่ได้นี้ ไปสร้างเพลย์เยอร์ (Player) หรือโปรเซสเซอร์ (Processor) โดยตรง หรือใช้มีเดียโลเคเตอร์นี้สร้างดาต้าซอร์ส ซึ่งเราจะใช้เป็นอินพุตให้กับเพลย์เยอร์หรือโปรเซสเซอร์ และเราจะต้องเรียกใช้เมธอด "start()" ของเพลย์เยอร์หรือโปรเซสเซอร์ เพื่อเริ่มต้นทำการจับข้อมูล

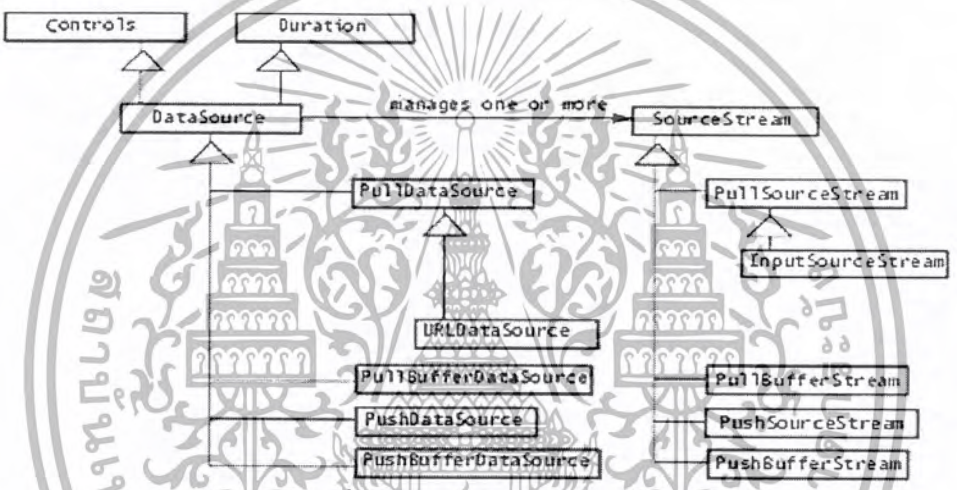
2.4.2 ดาต้าซอร์ส

เอพีไอของเจเอ็มเอฟ (JMF API) จะใช้ดาต้าซอร์สในการจัดส่งข้อมูลมีเดีย โดยที่ดาต้าซอร์สนั้นจะทำการเอนแคพซูเลทตำแหน่งของข้อมูล รวมไปถึงโปรโตคอลและซอฟแวร์ที่ใช้ในการส่ง ซึ่งในการได้มาซึ่งข้อมูลแต่ละครั้งนั้น แหล่งข้อมูล (Resource) หนึ่ง ๆ จะสามารถส่งให้กับสื่อได้เพียงตัวใดตัวหนึ่งเท่านั้น ไม่สามารถนำมาส่งให้กับสื่อตัวอื่นได้

ดาต้าซอร์สนั้นจะถูกแสดงโดยเจเอ็มเอฟ มีเดียโลเคเตอร์หรือยูอาร์แอล (Universal Resource Locator, URL) ตัวใดตัวหนึ่ง มีเดียโลเคเตอร์นั้นมีความคล้ายคลึงกันกับยูอาร์แอล และยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถสร้างได้จากยูอาร์แอลได้ด้วย ซึ่งจะถูกสร้างขึ้นก็ต่อเมื่อตัวจัดการโปรโตคอลที่มีลักษณะเช่นเดียวกัน (Corresponding Protocol Handler) นั้นไม่ถูกติดตั้งอยู่ในระบบ กล่าวคือ ในภาษาจาวานั้นยูอาร์แอลถูกสร้างได้เพียงตัวเดียวเท่านั้น ถ้าตัวจัดการโปรโตคอลที่มีลักษณะเช่นเดียวกันถูกติดตั้งอยู่ในระบบ ดาต้าซอร์สจะเป็นตัวจัดการกับกลุ่มของออบเจคท์ "SourceStream" ถ้าเป็นข้อมูลตามมาตรฐาน (Standard Data Source) จะมีการเก็บข้อมูลที่เอาไว้ใช้ในการส่งเป็นแบบไบต์อาเรย์ ถ้าเป็นแบบข้อมูลที่มีการพักเก็บไว้ก่อน (Buffer Data Source) จะใช้ออบเจคท์ "Buffer" เป็นตัวส่งข้อมูล (unit of transfer) ซึ่งเจเอ็มเอฟได้กำหนดชนิดของออบเจคท์ "DataSource" ไว้ดังรูปที่ 2.13



รูปที่ 2.13 ดาต้าโมเดลของเจเอ็มเอฟ

2.4.2.1 ดาต้าซอร์สแบบพุลและแบบพุช

ข้อมูลที่เราได้มานั้นอาจจะได้มาจากแหล่งข้อมูลที่หลากหลาย อาทิเช่น ได้มาจากข้อมูลในเครื่องที่เราที่มีอยู่แล้วหรือจากระบบเครือข่ายหรือไม่ก็จากการบรอดคาสท์ ซึ่งดาต้าซอร์สในเจเอ็มเอฟ สามารถแบ่งออกเป็น 2 ประเภทตามวิธีการส่งข้อมูลคือ ดาต้าซอร์สแบบพุล (Pull Data Source) และดาต้าซอร์สแบบพุช (Push Data Source)

1. ดาต้าซอร์สแบบพุล

วิธีการเริ่มส่งข้อมูลแบบนี้ ไคล์เอนท์จะเป็นคนเริ่มสั่งให้มีการส่งข้อมูล และอัตราของการส่ง ไคลเอนท์สามารถที่จะควบคุมการส่งข้อมูลต่าง ๆ ได้ โปรโตคอลที่ใช้ในการส่งแบบนี้ก็จะเป็นพวกไฮเปอร์เท็กซ์ทรานสเฟอร์โปรโตคอล (Hypertext Transfer Protocol, HTTP) และไฟล์ (File) ในเจเอ็มเอฟมีการกำหนดชนิดของดาต้าซอร์สแบบพุลไว้สองแบบคือ "PullDataSource" และ "PullBufferDataSource" โดยจะให้ออบเจคท์ของคลาส "Buffer" เป็นตัวที่ใช้ส่งข้อมูล ข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกต่างของทั้งสองแบบนี้ คือการส่งข้อมูลทาง "PullDataSource" จะเป็นการส่งแบบต่อเนื่องเรื่อย ๆ แต่ถ้าเป็นแบบ "PullBufferDataSource" จะส่งข้อมูลมาทีละขนาดบัฟเฟอร์

2. ดาต้าซอร์สแบบพุช

วิธีการเริ่มส่งข้อมูลแบบนี้ เซิร์ฟเวอร์จะเป็นผู้เริ่มสั่งให้มีการส่งข้อมูล และกำหนดอัตราของการส่ง ซึ่งดาต้าซอร์สแบบพุชนี้ จะประกอบไปด้วยบรอดคาสท์มีเดีย (Broadcast Media) มัลติคาสท์มีเดีย (Multicast Media) และวิดีโอออนดีมานด์ (video-on-demand ,VOD) สำหรับการส่งข้อมูลจำพวกบรอดคาสท์นี้ ก็จะมีโปรโตคอลที่เรียกว่า เรียลไทม์ทรานสปอร์ตโปรโตคอล (Real Time Transport Protocol, RTP) ซึ่งโปรโตคอลตัวนี้ถูกพัฒนาโดยไออีทีเอฟ เจเอ็มเอฟได้ทำการกำหนดชนิดของดาต้าซอร์สแบบพุชเอาไว้สองแบบคือ แบบ "PushDataSource" และแบบ "PushBufferDataSource" โดยจะให้ ออบเจคท์ของคลาส "Buffer" เป็นตัวที่ใช้ส่งข้อมูล ตัวอย่างที่แสดงให้เห็นถึงความแตกต่างที่ชัดเจนของการส่งข้อมูลทั้งสองแบบนี้คือ ถ้าเรามีไฟล์เอ็มเปกที่เป็นดาต้าซอร์สแบบพุล จะเป็นการอนุญาตให้ผู้ใช้สามารถที่จะทำการเล่นซ้ำ (Replay) หรือทำการเลือกดูข้อมูลส่วนไหนก็ได้ของไฟล์ในทางตรงกันข้าม ถ้าเป็นไฟล์เอ็มเปกที่เป็นดาต้าซอร์สแบบพุชตัวนี้จะถูกเก็บไว้ที่เซิร์ฟเวอร์ เซิร์ฟเวอร์จะบรอดคาสท์ไปให้ผู้ใช้แต่ละคนดู ผู้ใช้ไม่สามารถทำการเล่นซ้ำหรือเลือกดูข้อมูลแบบสุ่มได้ ผู้ใช้ต้องดูไปเรื่อย ๆ ตามแต่เซิร์ฟเวอร์จะส่งมาให้ เหมือนกับการถ่ายทอดโทรทัศน์

2.4.2.2 ดาต้าซอร์สชนิดพิเศษ

เจเอ็มเอฟได้กำหนดดาต้าซอร์สชนิดพิเศษเอาไว้สองชนิดคือ ดาต้าซอร์สที่สามารถโคลนได้ (Cloneable Data Sources) และดาต้าซอร์สผสม (Merging Data Sources)

1. ดาต้าซอร์สที่สามารถโคลนได้

มีไว้สำหรับการสร้างโคลน (Clone) ไม่ว่าจะเป็นดาต้าซอร์สแบบพุลหรือแบบพุช การสร้างดาต้าซอร์สที่สามารถโคลนได้ เราจะทำการเรียก เมธอด "createCloneableDataSource" จากคลาส "Manager" จากนั้นก็ส่งดาต้าซอร์สที่เราจะโคลนไปให้กับเมธอดนี้ เมื่อดาต้าซอร์สที่จะทำการโคลนถูกส่งไปเรียบร้อยแล้ว เราสามารถเรียกใช้งานดาต้าซอร์สที่ทำการโคลนกับตัวโคลนของมันได้โดยไม่ต้องเรียกใช้งานโดยตรงกับดาต้าซอร์สต้นแบบ ดาต้าซอร์สที่สามารถโคลนได้จะสนับสนุนอินเทอร์เฟซ "SourceCloneable" ซึ่งถูกกำหนดโดยเมธอด "createClone" เมื่อเราทำการเรียกเมธอดนี้ เราจะสามารถกำหนดได้ว่าจะสร้างโคลนขึ้นมาจำนวนเท่าไร ซึ่งโคลนที่สร้างขึ้นมานี้เราสามารถควบคุมได้โดยผ่านดาต้าซอร์สที่สร้างตัวมันขึ้นมา (เราสามารถใช้เมธอด "connect", "disconnect", "start" หรือ "stop" เพื่อเป็นการควบคุมดาต้าซอร์สที่สามารถโคลนได้) โคลนที่เรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

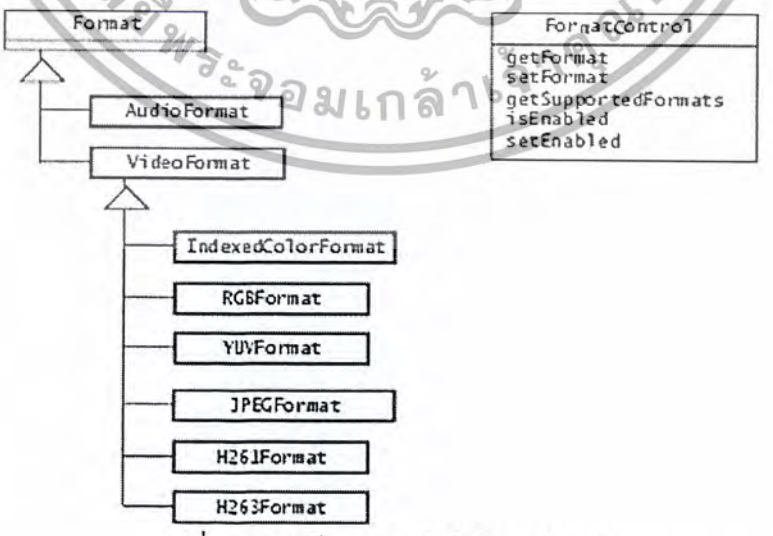
สร้างขึ้นมานี้ไม่จำเป็นที่จะต้องมีคุณสมบัติเหมือนกับดาต้าซอร์สต้นแบบที่ทำการโคลน กล่าวคือ สมมติว่าเราสร้างโคลนสำหรับอุปกรณ์จับข้อมูลซึ่งมีฟังก์ชันเหมือนกับดาต้าซอร์สที่สามารถโคลนได้ทุกประการ ในกรณีที่ไม่มีการเรียกใช้งานจากดาต้าซอร์สที่สามารถโคลนได้ ตัวโคลนของมันก็จะไม่ทำการผลิตข้อมูล แต่ถ้าเรามีการใช้งานทั้งสองตัวคือ ดาต้าซอร์สที่สามารถโคลนได้และตัวโคลนของมัน ตัวโคลนของมันก็จะทำการผลิตข้อมูลด้วยอัตราที่เทียบเท่ากับต้นแบบของมัน

2. ดาต้าซอร์สแบบผสม

ดาต้าซอร์สแบบผสม มีไว้เพื่อใช้สำหรับการรวม "SourceStream" จากดาต้าซอร์สสองหรือสามตัวเข้าเป็นดาต้าซอร์สตัวเดียว โดยเราสามารถควบคุมกลุ่มของดาต้าซอร์สนี้ได้เหมือนกับการควบคุมดาต้าซอร์สตัวเดียว ในการสร้างดาต้าซอร์สแบบผสมนี้ เราสามารถจะเรียกใช้เมธอด "createMergingDataSource" จากคลาส "Manager" จากนั้นก็ส่งอาเรย์ที่ประกอบไปด้วยดาต้าซอร์ทที่ต้องการรวมเข้าด้วยกัน แต่มีข้อแม้อยู่ว่า ดาต้าซอร์สที่จะทำการรวมกันนั้น จะต้องเป็นชนิดเดียวกัน กล่าวคือเราไม่สามารถจะนำ "PullDataSource" มารวมกันกับ "PushDataSource" ได้

2.4.2.3 ฟอร์แมตของข้อมูล

รูปแบบของการเก็บข้อมูลของสื่อนั้น จะถูกแสดงด้วยออบเจคท์หนึ่งที่เรียกว่าฟอร์แมต (Format) ซึ่งจะอธิบายถึงชื่อของการแปลงข้อมูลเป็นรหัสของรูปแบบของการเก็บข้อมูล (format's encoding name) และชนิดของรูปแบบของการเก็บข้อมูลที่ต้องการ เจเอ็มเอฟใช้ออบเจคท์ "Format" ในการกำหนดรูปแบบของการเก็บข้อมูลของออดิโอและวิดีโอดังรูปที่ 2.14



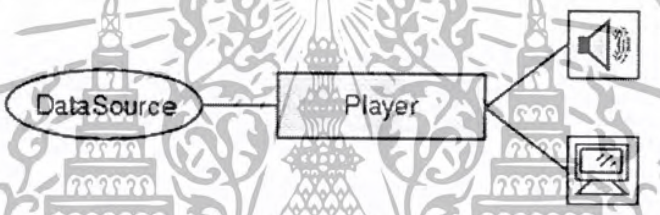
รูปที่ 2.14 ฟอร์แมตของมีเดียในเจเอ็มเอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอททริบิวท์ "AudioFormat" จะอธิบายรูปแบบการเก็บข้อมูลของออดิโอ ซึ่งประกอบด้วย อัตราแซมปลิ้ง จำนวนบิตต่อหนึ่งแซมเปิ้ล และจำนวนช่องสัญญาณเสียง ส่วน "VideoFormat" นั้น จะเป็นตัวเอนแคพซูลเลทเกี่ยวกับข้อมูลที่มีความสัมพันธ์กันกับข้อมูลวิดีโอ ซึ่ง "VideoFormat" นี้ จะเป็นตัวอธิบายเกี่ยวกับรูปแบบการเก็บข้อมูลพื้นฐานของวิดีโอ ดังนี้ "IndexedColorFormat", "RGBFormat", "YUVFormat", "JPEGFormat", "H261Format", "H263Format"

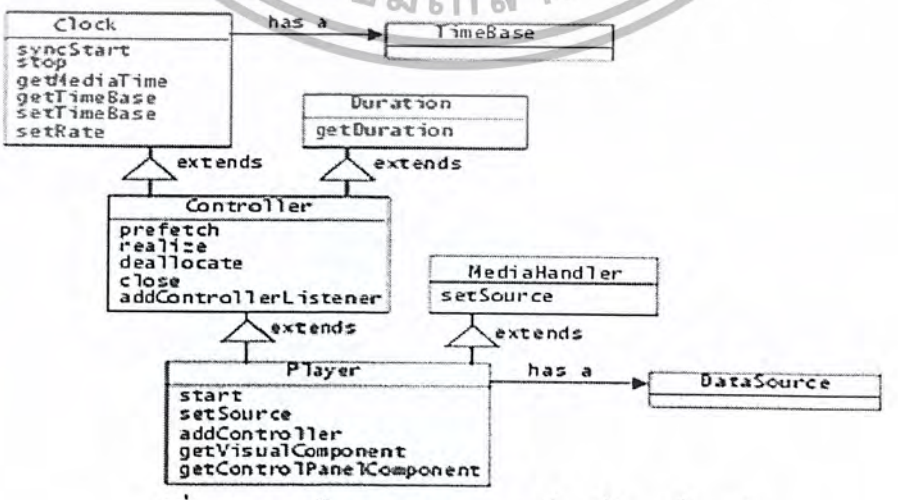
2.4.3 เพลย์เยอร์

เพลย์เยอร์ (Player) เป็นส่วนที่ใช้ในการประมวลผลอินพุตสตรีมของข้อมูล โดยจะแสดงผลได้ก็ต่อเมื่อดาต้าซอร์สนั้นได้จ่ายอินพุตสตรีมมายังเพลย์เยอร์ ในส่วนของการแสดงผลนั้นจะขึ้นอยู่กับชนิดของข้อมูลที่ใช้ในการแสดงผล



รูปที่ 2.15 แบบจำลองของเพลย์เยอร์ในเจเอ็มเอฟ

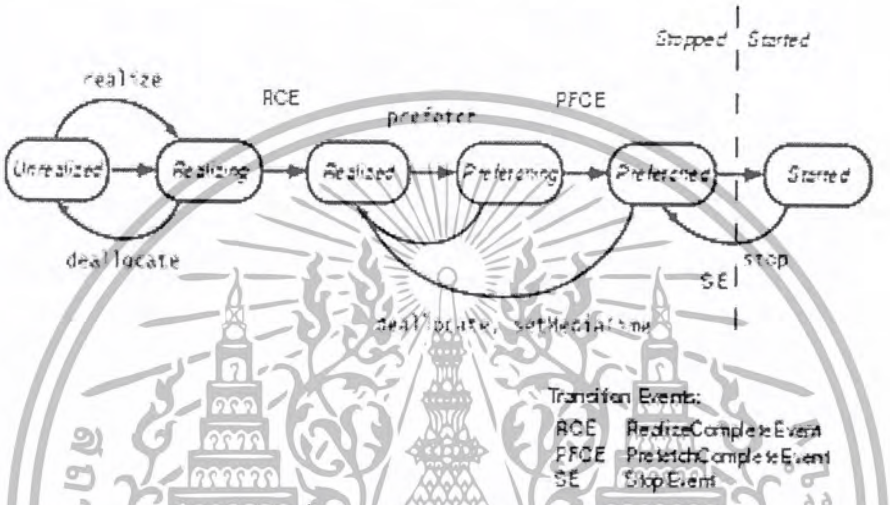
เพลย์เยอร์จะไม่สนใจว่า ข้อมูลจะถูกโพรเซสหรือเรนเดอร์ (Render) มาได้อย่างไร มันจะเป็นเพียงแค่ตัวควบคุมแบบมาตรฐานทั่วไป ที่มีอยู่ในทุก ๆ อุปกรณ์ควบคุมข้อมูลต่าง ๆ ตัวอย่าง เช่น เพลย์เยอร์จะมีแค่เพียงปุ่มเล่น (Play) และหยุด (Stop) เท่านั้น จะไม่มีอีควอไลเซอร์ (Equalizer) สำหรับปรับแต่งข้อมูล เราสามารถแสดงคลาสไดอะแกรมของเพลย์เยอร์ ได้ดังรูปที่ 2.16



รูปที่ 2.16 คลาสไดอะแกรมของเพลย์เยอร์ในเจเอ็มเอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพลย์เยอร์ จะมีสถานะที่เป็นไปได้อยู่หกแบบด้วยกัน สถานะหลักสองสถานะ “Stopped” และ Started ซึ่งได้ถูกนิยามไว้ในอินเทอร์เฟซ “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 เพลย์เยอร์ยังไม่มีข้อมูลเกี่ยวกับมีเดียที่ตัวเองจะติดต่อด้วย
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 ดังนั้นมันจึงสามารถให้คอมโพเนนท์สำหรับแสดงผลและควบคุมได้เลย การเชื่อมต่อกับ

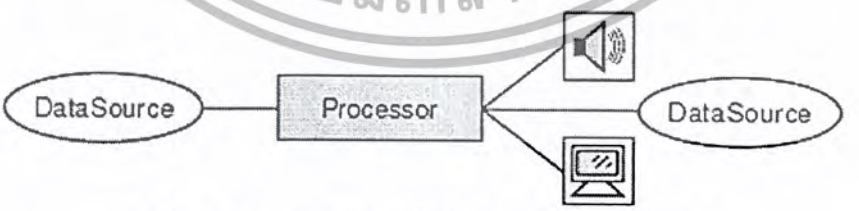
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจคท์อื่น ๆ ในระบบ ได้ถูกจัดเตรียมแล้ว แต่ว่าตัวเพลย์เยอร์เองยังไม่ได้ทำการจับจองทรัพยากรใด ๆ

4. เมื่อเมธอด "prefetch" ถูกเรียก เพลย์เยอร์จะเข้าสู่สถานะ "Prefetching" ซึ่งในสถานะนี้เพลย์เยอร์จะเตรียมตัวที่จะนำเสนอข้อมูลออกมา กระบวนการนี้เพลย์เยอร์จะทำการอ่านข้อมูลมาเตรียมไว้ และทำการจับจองทรัพยากรที่เป็นเอ็กคลูซีฟยูสและสิ่งต่าง ๆ ที่ต้องการในการนำเสนอสถานะ "Prefetching" นี้สามารถเกิดขึ้นได้อีก หากมีการเปลี่ยนแปลงใด ๆ ในตัวเพลย์เยอร์ ที่จะทำให้ต้องมีการเตรียมข้อมูลใหม่ เช่น การเปลี่ยนตำแหน่งที่จะอ่านข้อมูล การเปลี่ยนอัตราในการประมวลผลข้อมูล ทำให้ต้องอ่านบัฟเฟอร์มากขึ้น หรือการเปลี่ยนวิธีการประมวลผลข้อมูลเป็นต้น
5. เมื่อเพลย์เยอร์ผ่านสถานะ "Prefetching" แล้ว ก็จะมาอยู่ในสถานะ "Prefetched" ซึ่งในสถานะนี้เพลย์เยอร์ พร้อมแล้วที่จะไปสู่สถานะ "Started"
6. การเรียกเมธอด "start" จะทำให้เพลย์เยอร์ไปอยู่ในสถานะ "Started" ซึ่งในสถานะนี้เวลาจริงของระบบเวลาของข้อมูลจะถูกตั้งให้ตรงกันและนาฬิกาจะเริ่มทำงาน แม้ว่าตัวเพลย์เยอร์เองอาจต้องจะรอให้ถึงเวลาหนึ่งก่อนที่จะเริ่มนำเสนอข้อมูลก็ตาม

2.4.4 โปรเซสเซอร์ (Processor)

โปรเซสเซอร์เป็นอีกตัวหนึ่งที่มีไว้สำหรับการแสดงผลอินพุตของข้อมูลหมือนกับเพลย์เยอร์ แต่โปรเซสเซอร์มีความสามารถมากกว่า กล่าวคือ โปรเซสเซอร์สามารถทำการเปลี่ยนแปลง, ปรับปรุง หรือแก้ไขอินพุตของข้อมูลก่อนการแสดงผลได้ ซึ่งชนิดของข้อมูลที่เพลย์เยอร์สนับสนุนจะสามารถใช้กับโปรเซสเซอร์ได้



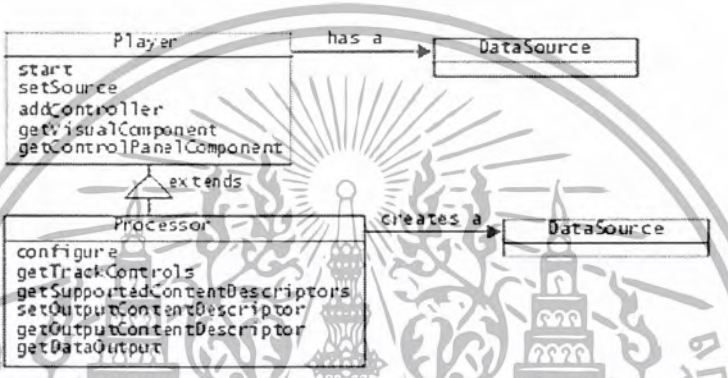
รูปที่ 2.18 แบบจำลองของโปรเซสเซอร์ในเจเอ็มเอฟ

สำหรับการแสดงผลข้อมูลนั้น โปรเซสเซอร์สามารถทำได้หลายวิธี คือ สามารถทำการสร้างดาต้าซอร์สขึ้นมาใหม่จากนั้น จะนำดาต้าซอร์สตัวนี้ส่งไปยังเพลย์เยอร์หรือโปรเซสเซอร์ตัวอื่น เพื่อแสดงผล หรือส่งไปยังเป้าหมายอื่น ๆ ในรูปของไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

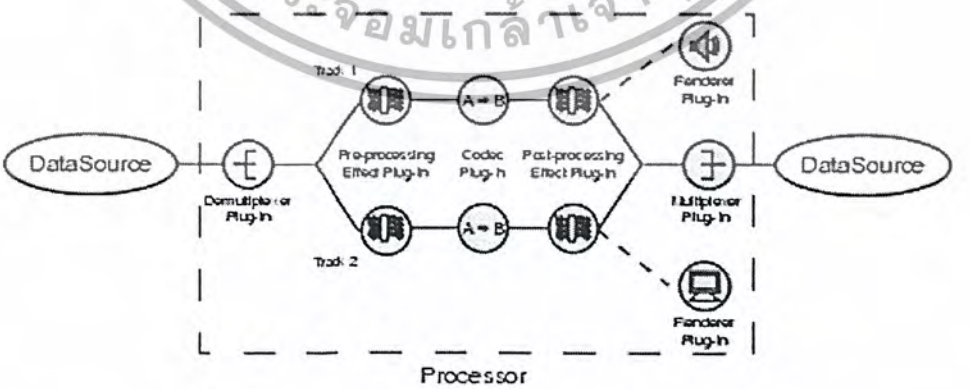
2.4.4.1 กระบวนการทำงานของ โปรเซสเซอร์

โปรเซสเซอร์คือเพลย์เยอร์ที่นำดาต้าซอร์สมาปรับปรุงเปลี่ยนแปลงข้อมูลเพื่อสร้างเป็นดาต้าซอร์สอีกตัวขึ้นมา และสามารถส่งข้อมูลไปให้อุปกรณ์แสดงผลอย่างอื่นได้ ซึ่งถ้าข้อมูลที่ส่งออกไปนั้นถูกส่งไปยังดาต้าซอร์ส ดาต้าซอร์สจะสามารถส่งข้อมูลที่อยู่ในตัวมันเองไปให้เพลย์เยอร์ หรือโปรเซสเซอร์อื่น ๆ หรือไม่ก็ส่งไปให้ดาต้าซิงค์ (DataSink) ได้อีกด้วย



รูปที่ 2.19 คลาสไดอะแกรมของโปรเซสเซอร์ในเจเอ็มเอฟ

ในขณะที่กระบวนการทำงานของเพลย์เยอร์นั้น ที่ถูกกำหนดขึ้นโดยนักพัฒนาโปรแกรม โปรเซสเซอร์ ได้มีการอนุญาตให้นักพัฒนาโปรแกรม สามารถกำหนดชนิดของกระบวนการทำงานที่เกี่ยวกับข้อมูลได้เช่นกัน ตัวอย่างเช่นการใส่เอฟเฟคท์หรือการผสมมีเดีย (Mixing) การทำงานแบบเรียลไทม์ กระบวนการทำงานของ โปรเซสเซอร์ ถูกแบ่งออกเป็นขั้นตอนดังรูปที่ 2.20



รูปที่ 2.20 ภาพรวมของสถานะต่าง ๆ ของโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

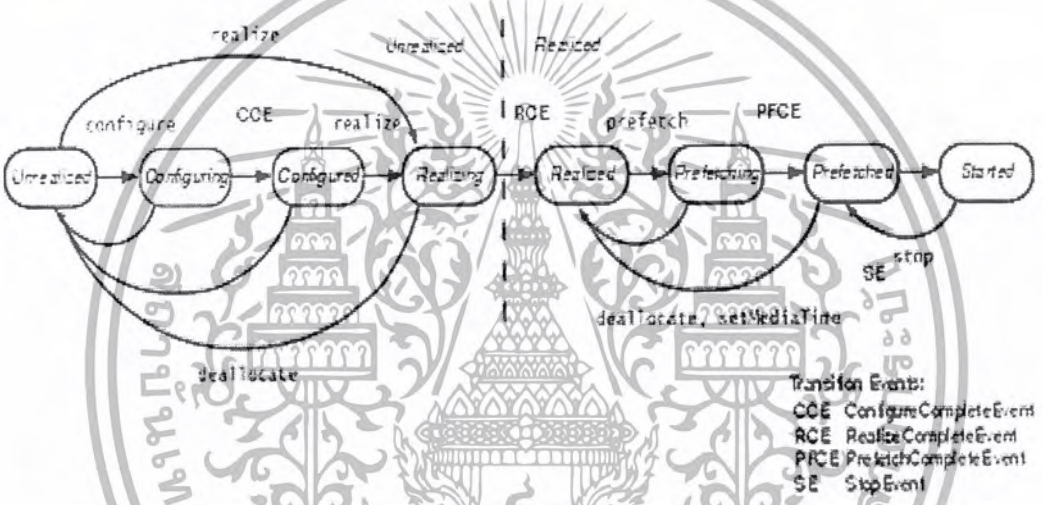
1. ดีมัลติเพล็กซิ่ง (Demultiplexing) คือกระบวนการที่กระจายข้อมูลออกเป็นแทร็ค (Tracks) ย่อย ๆ กล่าวคือ ถ้าข้อมูลที่เข้ามาประกอบไปด้วยหลายแทร็ครวมกัน เราสามารถทำการกระจายข้อมูลให้ออกมาเป็นแต่ละแทร็คได้ ยกตัวอย่างเช่น ไฟล์มัลติมีเดียแบบควิกไทม์ เราสามารถทำการดีมัลติเพล็กซ์ออกเป็นแทร็คออดิโอและแทร็ควิดีโอได้ ข้อมูลที่เข้ามายังโปรเซสเซอร์จะสามารถทำการดีมัลติเพล็กซ์ได้ก็ต่อเมื่อเป็นข้อมูลแบบมัลติเพล็กซ์เท่านั้น
2. พรีโพรเซสซิ่ง (Pre Processing) คือกระบวนการที่ตอบสนองต่อเอฟเฟ็คท์อัลกอริธึมของแต่ละแทร็คหลังจากการดีมัลติเพล็กซ์ข้อมูล
3. ทรานสโค้ดดิ้ง (Transcoding) คือกระบวนการที่ทำการแปลงสัญญาณของแต่ละแทร็คจากรูปแบบหนึ่งให้ไปเป็นอีกรูปแบบหนึ่ง ถ้าทำการแปลงสัญญาณจากข้อมูลชนิดที่มีการบีบอัดไปเป็นข้อมูลชนิดที่ไม่มีการบีบอัดจะเรียกว่า "การถอดรหัส (decoding)" และในทางกลับกัน ถ้าทำการแปลงสัญญาณจากข้อมูลที่ไม่มีการบีบอัดไปเป็นข้อมูลชนิดที่มีการบีบอัดจะเรียกว่า "การเข้ารหัส (encoding)"
4. โพสโพรเซสซิ่ง (Post Processing) คือเป็นกระบวนการที่เกี่ยวกับการตอบสนองต่อเอฟเฟคท์อัลกอริธึมของแทร็คที่ถูกถอดรหัส
5. มัลติเพล็กซิ่ง (Multiplexing) คือกระบวนการรวมแทร็คแต่ละแทร็คที่จะออกจากโปรเซสเซอร์ให้เป็นข้อมูลเพียงชุดเดียว เช่น เราสามารถทำการรวมแทร็คของออดิโอและวิดีโอให้เป็นข้อมูลชนิดเอ็มเปกวันเพียงข้อมูลเดียวได้ ซึ่งเราสามารถกำหนดชนิดของข้อมูลที่จะออกจากโปรเซสเซอร์ได้จากเมธอด "setOutputContentDescriptor()"
6. เรนเดอร์ริ่ง (Rendering) คือกระบวนการในการแสดงข้อมูลมายังอุปกรณ์แสดงผล ในกระบวนการทำงานของแต่ละขั้นตอน จะถูกจัดการโดยคอมโพเนนท์ของเจเอ็มเอฟที่มีชื่อว่า เจเอ็มเอฟปลั๊กอิน (JMF plug-in) ซึ่งถ้าโปรเซสเซอร์ของเราสนับสนุน "TrackControls" เราสามารถที่จะเลือกปลั๊กอินที่เราต้องการใช้ในแต่ละแทร็คได้ ซึ่งมีปลั๊กอินให้เลือกทั้งหมดห้าชนิด ดังนี้
    1) ดีมัลติเพล็กเซอร์ (Demultiplexer) เป็นการกระจายข้อมูลที่เป็นมีเดียสตรีมเช่น เอ็มเปกหรือควิกไทม์ที่ถูกมัลติเพล็กซ์มาออกเป็นหลายแทร็ค
    2) เอฟเฟคท์ (Effect) เป็นการเพิ่มลักษณะพิเศษลงไปในแทร็คของข้อมูล
    3) โคเดค (Codec) เป็นการทำการเข้ารหัสข้อมูลหรือการถอดรหัสข้อมูล
    4) มัลติเพล็กเซอร์ (Multiplexer) เป็นการผสมแทร็คหลาย ๆ แทร็คให้เป็นข้อมูลเพียงชุดเดียว แล้วส่งข้อมูลนี้ไปยังดาต้าซอร์ส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เรนเดอร์เรอร์ (Renderer) เป็นการแสดงผลข้อมูลในแต่ละแทร็คไปยังอุปกรณ์แสดงภาพและเสียง

2.4.4.2 สถานะของ โปรเซสเซอร์

โปรเซสเซอร์จะมีสถานะเพิ่มขึ้นจากเพลย์เยอร์ด้วยกันสองสถานะ คือ "Configuring" และ "Configured" โดยที่ทั้งสองสถานะนี้จะเกิดขึ้นก่อนที่โปรเซสเซอร์จะเข้าสู่สถานะ "Realizing"



รูปที่ 2.21 สถานะของโปรเซสเซอร์ในเจเอ็มเอฟ

- โปรเซสเซอร์จะเข้าสู่สถานะ "Configuring" หลังจากมีการเรียกใช้เมธอด "configure" เมื่อโปรเซสเซอร์อยู่ในสถานะนี้ จะมีการติดต่อกับดาต้าซอร์ส โดยการดีมัลติเพล็กซ์ข้อมูลที่เข้ามา และแจ้งให้ดาต้าซอร์สทราบถึงชนิดของข้อมูลนี้
- โปรเซสเซอร์จะเข้าสู่สถานะ "Configured" เมื่อกำลังติดต่อกับดาต้าซอร์สและได้ทราบถึงชนิดของข้อมูลแล้ว เมื่อโปรเซสเซอร์อยู่ในสถานะ "Configured" จะได้คลาสที่ชื่อ "ConfigureCompleteEvent" ออกมา
- เมื่อมีการเรียกใช้งานเมธอด "realize" นั้น โปรเซสเซอร์จะเปลี่ยนแปลงไปเป็นสถานะ "Realized" ซึ่งเราจะถือว่าโปรเซสเซอร์ของเราได้มีการทำงานเสร็จสมบูรณ์ และในขณะที่โปรเซสเซอร์อยู่ในสถานะ "Configured" นั้น เราสามารถที่จะเรียกใช้งานเมธอด "getTrackControls" เพื่อเอาค่าในออบเจคท์ "TrackControl" สำหรับแต่ละแทร็คของข้อมูลมา เราสามารถระบุการทำงานของโปรเซสเซอร์ได้จากออบเจคท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"TrackControl" นี้ ในการเรียกใช้เมธอด "realize" โดยตรง โปรเซสเซอร์จะเปลี่ยนจากสถานะจาก "Unrealized" ไปเป็นสถานะ "Realized" โดยอัตโนมัติ แต่เราจะไม่สามารถใช้ออบชั่น "TrackControl" เพื่อดูรายละเอียดของข้อมูลแต่ละแทร็คได้ และเราไม่สามารถเรียกใช้เมธอด "TrackControl" เมื่อโปรเซสเซอร์อยู่ในสถานะ "Realized"

2.4.4.3 การควบคุมการประมวลผลเสียง (Processing controls)

เราสามารถที่จะเรียกใช้เมธอด "getTrackControls" นี้โดยการดึงเอาข้อมูลในแต่ละแทรคผ่านคลาส "TrackControl" เพื่อควบคุมการทำงานของโปรเซสเซอร์ที่กระทำต่อแทรคแต่ละแทร็ค เราสามารถเรียกใช้คลาส "PlugInManager" เพื่อทำการตรวจสอบว่าเราได้มีการติดตั้งปลั๊กอินอะไรบ้าง การควบคุมการทรานสโค้ดในแต่ละแทรคโดยโคเดคแต่ละตัวสามารถเรียกใช้ตัวช่วยจากเมธอด "getControls" ของ "TrackControl" ได้ ซึ่งเมธอดนี้จะให้ค่าโคเดคคอนโทรลของแทรค เช่น "BitRateControl" และ "QualityControl" ในการกำหนดชนิดของเอาท์พุตเป็นไปตามที่เราต้องการ เราสามารถเรียกใช้เมธอด "setFormat" ในการระบุชนิดของฟอร์แมตที่เราต้องการ จากนั้นโปรเซสเซอร์จะเลือกโคเดคและเรนเดอร์เรอร์ที่เหมาะสมให้ อีกวิธีหนึ่งที่เราสามารถทำได้คือ ทำการระบุชนิดของข้อมูลเมื่อมีการสร้างโปรเซสเซอร์ขึ้นมาโดยใช้คลาส "ProcessorModel" ซึ่งในคลาส "ProcessorModel" นี้จะทำการกำหนดอินพุตและเอาท์พุตที่ต้องการสำหรับโปรเซสเซอร์ที่เราสร้างขึ้นมา

2.4.4.4 ดาต้าเอาท์พุต (Data Output)

เมธอด "getDataOutput" จะทำการคืนค่าผลลัพธ์ที่ได้จากโปรเซสเซอร์ในรูปของดาต้าซอร์ส ซึ่งดาต้าซอร์สนี้อาจอยู่ในรูปของ "PushDataSource", "PushBufferDataSource", "PullDataSource" และ "PullBufferDataSource" และยังสามารถเป็นข้อมูลที่ป้อนให้กับเพลย์เยอร์หรือโปรเซสเซอร์ตัวอื่น หรือเป็นข้อมูลที่ส่งไปยังดาต้าซิงค์ได้ ซึ่งโปรเซสเซอร์ที่ไม่มีการส่งค่าผลลัพธ์ให้ดาต้าซอร์สจะถูกเปรียบเสมือนว่าเป็นเพลย์เยอร์

2.4.5 เมเนเจอร์ (Managers)

เอพีไอของเจเอ็มเอฟมีการทำงานของออบเจคท์ต่าง ๆ หลายส่วนประกอบกัน เช่น การจับเสียง การประมวลผล และการนำเสนอสื่อเชิงเวลา ซึ่งแต่ละส่วนนี้จะมีความสัมพันธ์ซึ่งกันและกัน เราจำเป็นต้องมีออบเจคท์ที่ใช้เป็นตัวกลาง (Intermediary Object) ในการทำให้ออบเจคท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่าง ๆ สามารถทำงานร่วมกันได้ ซึ่งเจเอ็มเอฟก็มีออบเจคท์ตัวกลางนี้เอาไว้สำหรับให้เรียกใช้อยู่แล้ว มีชื่อว่า "Manager" โดยเจเอ็มเอฟจะมีเมเนเจอร์ที่เกี่ยวข้องทั้งหมดสี่ชนิดคือ

1. เมเนเจอร์ (Manager) ใช้จัดการเกี่ยวกับการสร้างออบเจคท์ของคลาส "Players", "Processors", "DataSources" และ "DataSinks"
2. แพ็คเกตเมเนเจอร์ (PackageManager) ใช้ในการเก็บรายการ (Registry) ของแพคเก็จทั้งหมดที่เกี่ยวกับ เจเอ็มเอฟ เช่นคลาส คลาส "Players", "Processors", "DataSources" และ "DataSinks" ที่เราสามารถเขียนขึ้นมาเองได้
3. แคพเจอร์ดีไวซ์เมเนเจอร์ (CaptureDeviceManager) ใช้ในการจัดการอุปกรณ์จับข้อมูลมีเดียต่าง ๆ
4. ปลั๊กอินเมเนเจอร์ (PlugInManager) จะใช้ในการเก็บรายการของปลั๊กอินต่าง ๆ ที่มีในเจเอ็มเอฟ ซึ่งในเจเอ็มเอฟมีปลั๊กอินด้วยกันทั้งสิ้นห้าชนิด คือ "Multiplexers", "Demultiplexers", "Codecs", "Effects" และ "Renderers"

ในการพัฒนาโปรแกรมโดยใช้เจเอ็มเอฟ เราจำเป็นที่จะต้องใช้เมธอดของคลาส "Manager" เพื่อทำการสร้างเพลย์เยอร์ โปรเซสเซอร์ ดาต้าซอร์สและดาต้าซิงค์

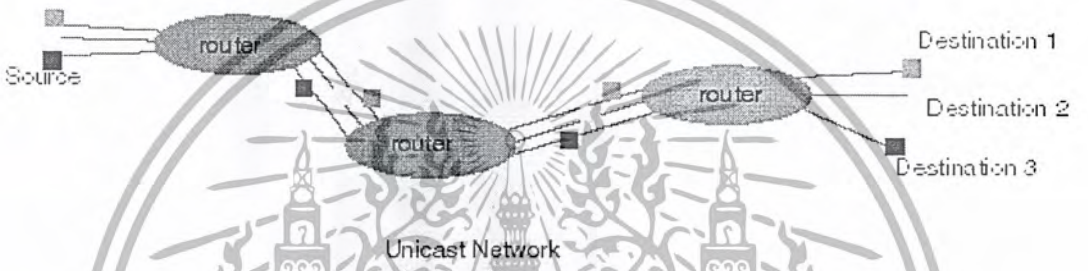
2.5 เครือข่ายมัลติคาสต์ (Multicast Network)

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

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

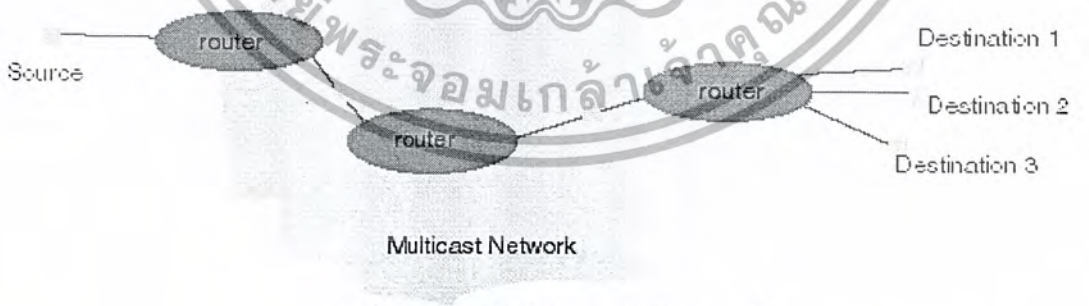
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



รูปที่ 2.22 การส่งข้อมูลแบบยูนิคาสต์

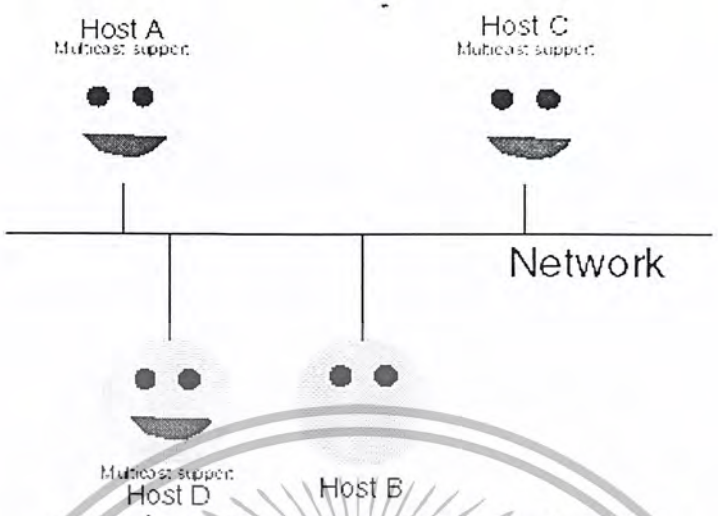
จากรูปที่ 2.22 เป็นการส่งแบบหนึ่งผู้ส่งไปยังหนึ่งผู้รับ ข้อมูลชุดเดียวกันถูกส่งแบบจุดต่อจุดไปยังผู้รับ 3 กลุ่ม ซึ่งต้องส่งข้อมูลถึง 3 ครั้ง



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

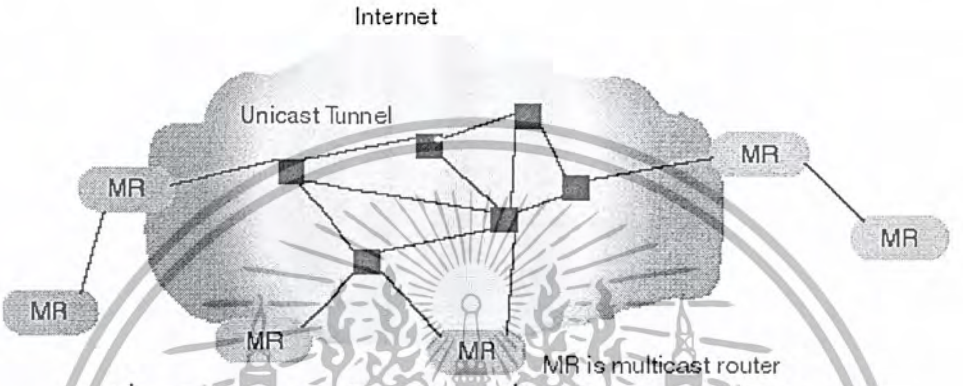
มัลติคาสต์เป็นเทคโนโลยีแบบที่ทางด้านผู้ส่งไม่จำเป็นต้องทราบ หรือจัดการทางด้านรับ (Receiver-Based Concept) เหมือนกับการส่งข้อมูลผ่านวิทยุโทรทัศน์นั่นเอง เมื่อผู้รับต้องการรับข้อมูลดังกล่าว ก็ปรับสถานีหรือช่องรับให้ตรงกับสถานีส่งที่ตนเองต้องการ

2.5.1 เอ็มโบน (MBone หรือ Multicast Backbone)

ปัจจุบันได้มีการนำเอาการส่งข้อมูลแบบมัลติคาสต์ มาใช้ในเครือข่ายอินเทอร์เนต ซึ่งเป็นที่ทราบกันดีในชื่อของเอ็มโบน เกิดจากการทดลองที่จะส่งภาพและเสียงการประชุมขององค์กรไออีทีเอฟ (Internet Engineering Task Force (IETF)) โดยเป็นการร่วมมือของอาสาสมัครในการทดลองและใช้งานไอพีมัลติคาสต์ (IP Multicast) และเป็นการสร้างเครือข่ายของไอพีมัลติคาสต์เพื่อการพัฒนาการใช้งานของโปรแกรมประยุกต์ต่าง ๆ โดยไม่ต้องรอการแพร่หลายของเราท์เตอร์ที่สนับสนุนการทำมัลติคาสต์บนเครือข่ายอินเทอร์เนต ดังนั้นเอ็มโบนจึงเป็นเครือข่ายเสมือนที่ทำงานบนเครือข่ายอินเทอร์เนต โดยมีหน้าที่สนับสนุนการหาเส้นทางของไอพีมัลติคาสต์ เอ็มโบนนี้สามารถที่จะใช้ได้กับเครือข่ายทั้งที่สนับสนุนการทำมัลติคาสต์และไม่สนับสนุนการทำมัลติคาสต์ โดยเมื่อข้อมูลมัลติคาสต์ผ่านเครือข่ายแบบเดิม (ที่ไม่สนับสนุนการทำมัลติคาสต์) ข้อมูลของมัลติคาสต์ก็จะถูกส่งไปเป็นแบบยูนิคาสต์ โดยมีข้อมูลมัลติคาสต์อยู่ภายใน (Encapsulation) วิธีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังกล่าวจะเรียกว่าการทำอุโมงค์ (Tunneling) เพื่อให้ส่งข้อมูลมัลติคาสต์ผ่านยังเครือข่ายยูนิคาสต์ได้ ดังรูปที่ 2.25 โดยจุดที่เป็นมัลติคาสต์เราท์เตอร์ (Multicast Router) หรือจุดปลายของอุโมงค์ ก็จะเป็นอุปกรณ์ที่สนับสนุนวิธีการมัลติคาสต์ หรืออาจเป็นเครื่องคอมพิวเตอร์ที่ทำงานด้วยโปรแกรมที่ชื่อเอ็มเราท์ (Mrouted) ซึ่งทำหน้าที่เป็นมัลติคาสต์เราท์ติ้ง (Multicast Routing)



รูปที่ 2.25 การทำอุโมงค์ผ่านเครือข่ายที่ไม่สนับสนุนการทำมัลติคาสต์

2.5.2 กลุ่มของมัลติคาสต์ (Multicast Groups)

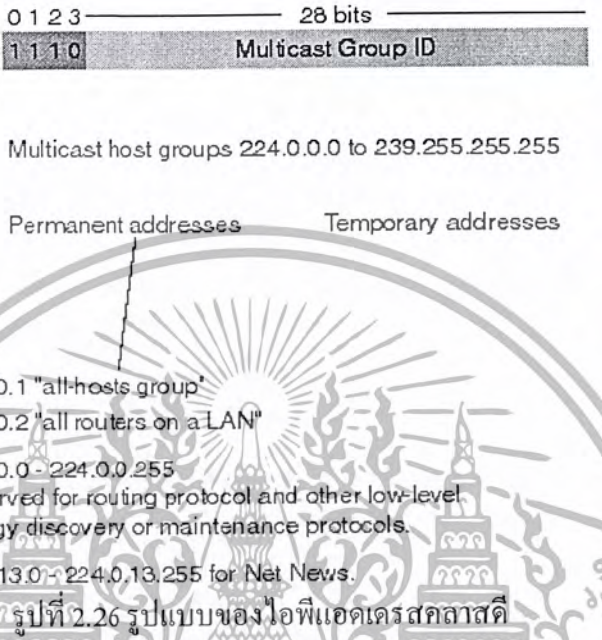
ในรูปแบบของไอพีวีโฟร์แอดเดรส (IPv4 addresses) จะถูกแบ่งออกเป็นสามชนิดด้วยกันคือ ยูนิคาสต์ บรอดคาสต์ และมัลติคาสต์ ซึ่งยูนิคาสต์แอดเดรสจะใช้สำหรับส่งข้อมูลแบบจุดต่อจุด ส่วนบรอดคาสต์แอดเดรสจะใช้สำหรับส่งข้อมูลไปยังทุก ๆ จุดที่อยู่ในเครือข่ายนั้น โดยที่ไอพีแอดเดรสคลาส เอ บี และ ซี (Class A, B and C IP address) 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 บิตโดย 4 บิตแรกจะเป็น 1110 ส่วนอีก 28 บิตที่เหลือเราจะเรียกว่า มัลติคาสต์กรุ๊ปไอดี (Multicast Group ID) ซึ่งใช้แทนกลุ่มของมัลติคาสต์ ลักษณะการเขียนหมายเลขไอพีในแบบของดอทเดซิมอล (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 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายถึง เราท์เตอร์ทั้งหมด เป็นต้น รายละเอียดของไอพีที่กันไว้ใช้งานนั้นสามารถหารายละเอียดดูได้จากอาร์เอฟซี1700 (RFC1700)



รูปที่ 2.26 รูปแบบของไอพีแอดเดรสคลาสดี

ทั้งนี้ มัลติคาสต์ดาต้าแกรม (Multicast Datagram) หรือแพ็คเกต (Packet) สามารถที่จะส่งไปยังกลุ่มผู้รับได้ด้วยวิธีเดียวกันกับยูนิคาสต์ไอพีปกติ (อาจเกิดการสูญหายและการผิดลำดับได้เช่นเดียวกันกับยูนิคาสต์ไอพีแพ็คเกต)

2.5.2.2 ไอจีเอ็มพี (Internet Group Management Protocol (IGMP))

ในการรับข้อมูลของมัลติคาสต์นั้น ผู้รับในกลุ่มมัลติคาสต์ จะต้องมีการแจ้งให้กับเราท์เตอร์ทราบว่าต้องการที่จะรับข้อมูล ซึ่งในแต่ละจุดสามารถที่จะมีสมาชิกได้มากกว่าหนึ่งกลุ่มขึ้นไป โดยเราท์เตอร์จะทำการรับและส่งข้อมูลให้กับสมาชิกในกลุ่ม กลไกที่ใช้ในการทำงานงานดังกล่าว เราจะเรียกว่า "ไอจีเอ็มพี" วิธีการนี้ยังใช้ในการตรวจสอบสมาชิกในกลุ่มว่ายังมีอยู่หรือไม่อีกด้วย ในกรณีที่มีจำนวนเราท์เตอร์หลาย ๆ ตัวในเครือข่ายแลน (LAN) จะมีอยู่หนึ่งตัวที่จะถูกเลือกขึ้นมาเป็นตัวแทน เพื่อทำการสอบถาม (Query) และเก็บสถานะของสมาชิกในกลุ่มของมัลติคาสต์ จากข้อมูลที่ได้มา เราท์เตอร์จะใช้ข้อมูลเหล่านี้ในการตัดสินว่า จะส่งข้อมูลให้กับผู้รับในเครือข่ายหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 มัลติคาสต์เราท์ติ้งอัลกอริทึม (Multicast Routing Algorithms)

ปัจจุบันมีหลายวิธี ในการสร้างเส้นทางสำหรับส่งข้อมูลไปยังผู้รับ หรือเรียกกันว่า มัลติคาสต์ทรี (Multicast Tree) โดยวิธีการเหล่านี้จะถูกนำไปใช้ในมัลติคาสต์เราท์ติ้งโปรโตคอล (Multicast Routing Protocol) เราจะมาดูสองวิธีการง่าย ๆ ก่อนที่เรียกกันว่า ฟลัดดิ้ง (Flooding) และสแปนนิ่งทรี (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 เรานำเอาวิธีการต่าง ๆ เหล่านี้ไปใช้ประโยชน์อย่างไร

2.5.3.1 ฟลัดดิ้ง (Flooding)

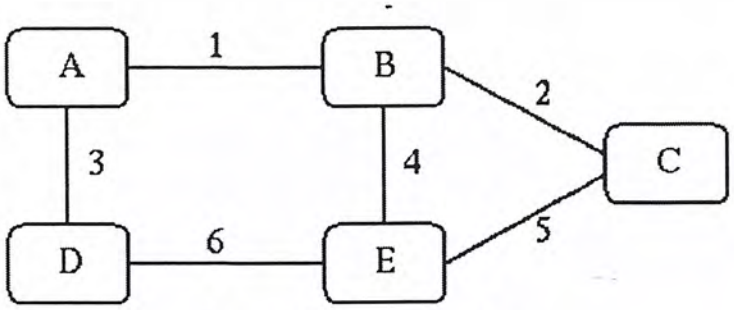
ฟลัดดิ้งอัลกอริทึม (Flooding Algorithm) นี้ เป็นวิธีการที่ใช้อยู่ในหลาย ๆ โปรโตคอล วิธีการนี้ใช้เทคนิคง่าย ๆ คือ เมื่อเราท์เตอร์ได้รับข้อมูลมัลติคาสต์ ก็จะทำการตรวจสอบข้อมูลนั้นว่าเป็นข้อมูลที่ได้รับมาครั้งแรกหรือไม่ ถ้าใช่เราท์เตอร์ก็จะทำการส่งข้อมูลนั้นออกไปยังเราท์เตอร์ตัวอื่น ๆ ทุกตัวยกเว้นตัวที่มันได้รับข้อมูลนั้นมา ถ้าเป็นกรณีอื่นเราท์เตอร์ก็จะไม่สนใจข้อมูลดังกล่าว ในกรณีนี้เราแน่ใจได้เลยว่าเราท์เตอร์ทุกตัวในเครือข่ายจะได้รับข้อมูลอย่างน้อยหนึ่งครั้ง

แม้ว่าวิธีการนี้จะง่ายแต่มีข้อเสียคือ เปลืองเครือข่ายการสื่อสารมาก เนื่องจากมีการส่งข้อมูลออกจากเราท์เตอร์แต่ละตัวไปยังเราท์เตอร์ตัวอื่น ๆ ที่ตัวมันเองเชื่อมต่ออยู่ ยิ่งไปกว่านั้น เราท์เตอร์ยังต้องเก็บสถานะของข้อมูลไว้ เพื่อตรวจสอบด้วยว่าเป็นข้อมูลที่เคยได้รับมาแล้ว หรือได้รับมาเป็นครั้งแรก

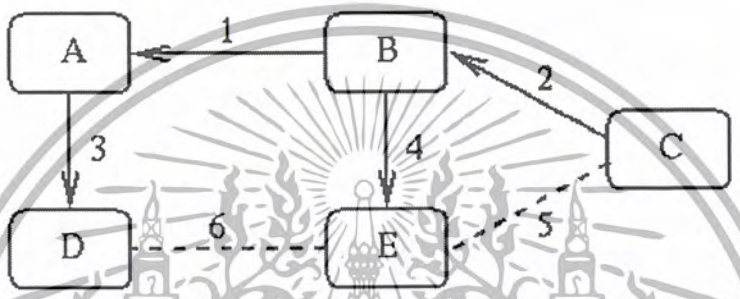
2.5.3.2 สแปนนิ่งทรี (Spanning Trees)

วิธีการที่ดีกว่าฟลัดดิ้งก็คือสแปนนิ่งทรี วิธีนี้ในเครือข่ายจะถูกกำหนดให้เป็นลักษณะของโครงสร้างต้นไม้ กล่าวคือ เป็นโครงสร้างที่ไม่มีการวนรอบ (Loop-Less Graph) จะมีเส้นทางเดียวในการติดต่อระหว่างสองเราท์เตอร์ เมื่อใดก็ตามที่เราท์เตอร์ได้รับมัลติคาสต์แพ็คเกตมันจะส่งต่อไปยังทุกข่ายที่เชื่อมต่อในเส้นทางของผังต้นไม้ ยกเว้นเส้นทางที่ได้รับแพ็คเกตมา เมื่อพิจารณาจากรูปที่ 2.27 ประกอบ จะเห็นว่าเราท์เตอร์ทุกตัวในเครือข่ายได้รับข้อมูลครบทุกตัว โดยเราท์เตอร์จะเก็บข้อมูลที่แสดงให้รู้ว่าเส้นทางนั้นเป็นสแปนนิ่งทรีหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



A small test network



Spanning Tree from source (C)

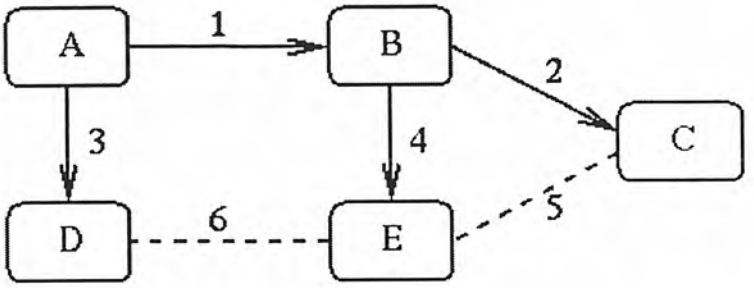
รูปที่ 2.27 สแปนนิ่งทรี

วิธีการแบบสแปนนิ่งทรีมีข้อเสียอยู่สองประการคือ ประการแรกมันจะพิจารณาทราฟฟิคทั้งหมดถึงแม้ว่าจะเป็นจุดเล็ก ๆ ในการเชื่อมโยง อีกประการหนึ่งคือ มันไม่ได้คำนึงว่าจุดนั้นจะมีสมาชิกของมัลติคาสต์อยู่หรือไม่

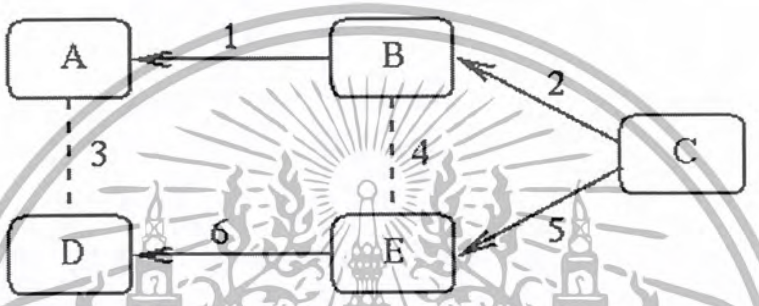
2.5.3.3 อาร์พีบี (Reverse Path Broadcasting (RPB))

วิธีการแบบอาร์พีบีที่ถูกนำมาใช้ในเอ็มโบนนั้น เป็นการปรับปรุงวิธีการของสแปนนิ่งทรี โดยตอนเริ่มแรกจะใช้วิธีการของสแปนนิ่งทรีในการสร้างเส้นทาง เมื่อเราท์เตอร์ได้รับข้อมูลบนเส้นทางแอลจากต้นทางเอส เราท์เตอร์ก็จะทำการตรวจสอบดูว่าเส้นทางแอลเป็นเส้นทางที่สั้นที่สุดไปยังต้นทางเอสหรือไม่ ถ้าใช่เราท์เตอร์ก็จะส่งข้อมูลไปยังทุกเส้นทางยกเว้นเส้นทางแอล ถ้าเป็นกรณีอื่น ๆ เราท์เตอร์ก็จะทิ้งข้อมูลนั่นไป ดูรูปที่ 2.28 ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



RPB tree from source (A)



RPB tree from source (C)

รูปที่ 2.28 อาร์พีบีทรี

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

วิธีการอาร์พีบีนี้ง่ายและเร็ว เนื่องจากการส่งข้อมูลจากแหล่งกำเนิดไปยังปลายทาง จะใช้เส้นทางที่ดีที่สุดในการส่ง แต่ก็ยังมีข้อบกพร่องบางประการคือ วิธีนี้จะไม่ทราบข้อมูลเกี่ยวกับสมาชิกของกลุ่มมัลติคาสต์ เพื่อใช้สำหรับสร้างดีสทริบิวชันทรี (Distribution Tree) เลย

2.5.3.4 ทีอาร์พีบี (Truncated Reverse Path Broadcasting (TRPB))

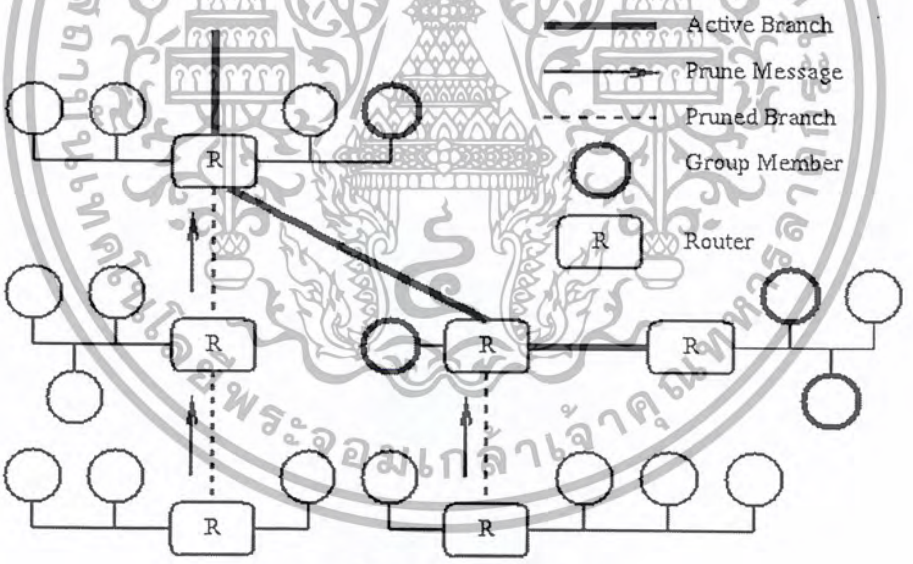
ทีอาร์พีบีเป็นวิธีการที่นำเสนอขึ้นมา เพื่อแก้ไขข้อบกพร่องของอาร์พีบี โดยจะใช้โปรโตคอลไอจีเอ็มพีเข้ามาช่วย ทำให้เราท์เตอร์สามารถกำหนดได้ว่า มีสมาชิกมัลติคาสต์อยู่ในเครือข่ายของเราท์เตอร์ตัวนั้นหรือไม่ ในกรณีที่ไม่มีสมาชิกอยู่ภายในเครือข่ายเลย และถ้าเครือข่ายนั้นอยู่ปลายสุด (ไม่มีการติดต่อกับเราท์เตอร์ตัวอื่น ๆ) เราท์เตอร์ก็จะทำการตัดส่วนนี้ออกจากสแปนนิ่งทรี นอกนั้นก็จะเหมือนกับวิธีการของอาร์พีบี ถึงแม้ว่าเรานำเอาเงื่อนไขนี้มาใช้ แต่ถ้าเราท์เตอร์นั้นไม่ได้อยู่ปลายสุด แม้จะไม่มีสมาชิกอยู่ก็ไม่สามารถตัดออกไปจากระบบได้ เนื่องจากอาจจะยังมีการเชื่อมต่ออยู่กับตัวอื่น ๆ ที่ยังมีสมาชิกของมัลติคาสต์อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.5 อาร์พีเอ็ม (Reverse Path Multicasting (RPM))

อาร์พีเอ็ม (หรือที่รู้จักกันว่าเป็นอาร์พีบีที่ประกอบด้วยวิธีการตัดออกจากกลุ่มมัลติคาสต์) เป็นการเพิ่มประสิทธิภาพจากวิธีอาร์พีบี และทีอาร์พีบี โดยอาร์พีเอ็มจะสร้างเส้นทางการส่งข้อมูลเฉพาะสองกรณีคือ ส่งข้อมูลเฉพาะในเส้นทางที่มีสมาชิกของมัลติคาสต์อยู่ และส่งข้อมูลเฉพาะเส้นทางที่สั้นที่สุดเท่านั้น

ตัวอย่างเช่น กำหนดให้คู่ของแหล่งข้อมูลหรือกลุ่มสมาชิกแพ็คเกตแรกของมัลติคาสต์ถูกส่งไปด้วยวิธีแบบทีอาร์พีบี ถ้าเราท์เตอร์ที่อยู่ปลายสุดของทรีได้รับข้อมูลมัลติคาสต์ และไม่มีสมาชิกใดติดต่ออยู่กับเราท์เตอร์ดังกล่าว มันก็จะส่งข้อมูลที่เรียกว่า พรุน (Prune) ไปยังเราท์เตอร์ที่เป็นตัวส่งข้อมูลมัลติคาสต์นั้นมาให้ ซึ่งข้อมูลพรุนดังกล่าวทำให้เราท์เตอร์ทราบว่า ไม่ต้องทำการส่งข้อมูลมัลติคาสต์นั้นไปยังเส้นทางที่ได้รับข้อมูลพรุนแล้ว และที่สำคัญก็คือ พรุนจะถูกส่งภายในขอบเขตไม่เกิน 1 จุดในเครือข่าย (1 Hop) จะทำให้การส่งข้อมูลมัลติคาสต์ส่งในเฉพาะเส้นทางที่มีสมาชิกอยู่เท่านั้น ดูรูปที่ 2.29ประกอบ



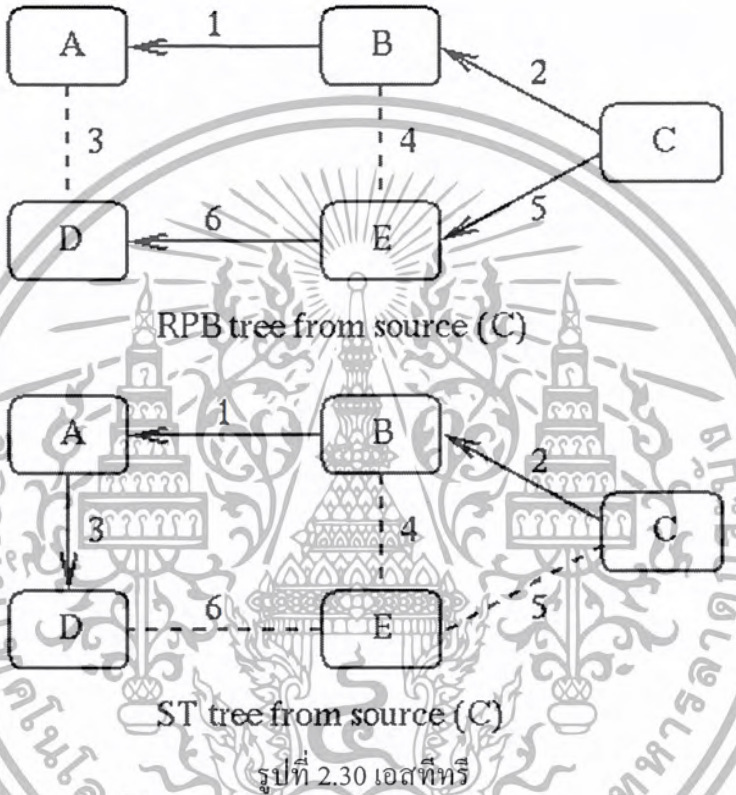
รูปที่ 2.29 อาร์พีเอ็มทรี

เนื่องจากกลุ่มของสมาชิกและโครงสร้างของเครือข่าย สามารถเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นจึงต้องมีการตรวจสอบสถานะอยู่เป็นระยะ ๆ เพื่อที่จะแจ้งให้ต้นทางทราบ ซึ่งทำให้จุดอ่อนของวิธีการนี้คือ จะต้องใช้หน่วยความจำจำนวนมาก เพื่อใช้ในการเก็บสถานะสำหรับในแต่ละคู่ของแหล่งข้อมูลและกลุ่มสมาชิก อีกอย่างหนึ่งก็คือ วิธีการดังกล่าวไม่เหมาะสมกับเครือข่ายขนาดใหญ่ ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.6 เอสที (Steiner Tree (ST))

วิธีการแบบอาร์พีบี ทีอาร์พีบี และอาร์พีเอ็มนั้น เป็นวิธีการแบบเลือกระยะทางที่สั้นที่สุด (Shortest Path) ระหว่างแหล่งกำเนิดและจุดรับข้อมูลปลายทาง ซึ่งรับประกันได้ว่าข้อมูลจะถูกส่งไปยังปลายทางได้อย่างรวดเร็วที่สุด แต่ถึงกระนั้นวิธีการดังกล่าวก็ยังไม่สามารถที่จะใช้ทรัพยากรของเครือข่ายได้อย่างมีประสิทธิภาพ



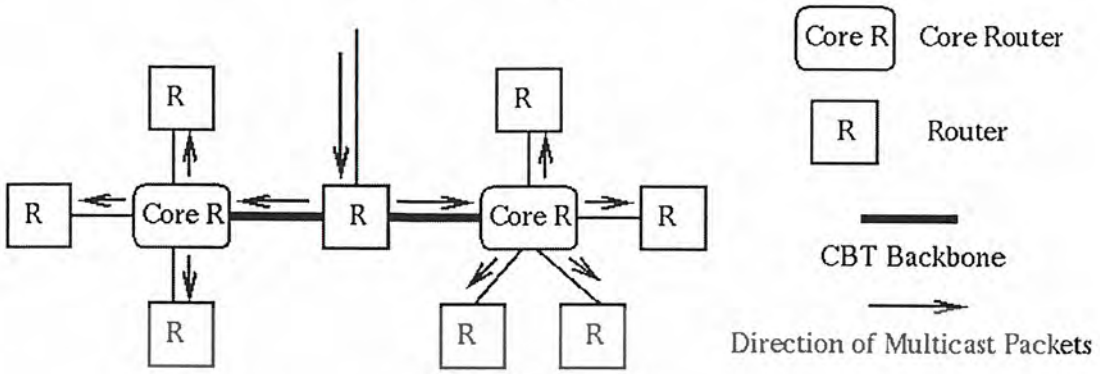
รูปที่ 2.30 เอสทีทรี

จากรูปที่ 2.30 จะเห็นได้ว่า วิธีเอสทีทรีจากแหล่งข้อมูล C จะดีกว่าในรูปแรก ถึงแม้ว่าวิธีการดังกล่าวจะทำให้เกิดเส้นทางที่ดีที่สุดก็ตาม แต่มันก็ยากในการที่จะคำนวณโครงสร้างให้เป็นโครงสร้างที่เล็ก ๆ ไม่ซับซ้อนได้ ดังนั้นรูปแบบของเอสทีทรีซึ่งมีการเข้าและออกของสมาชิกอยู่ตลอดเวลา จึงทำให้วิธีนี้ไม่เสถียรเป็นอย่างมาก

2.5.3.7 ซีบีที (Core-Based Tree (CBT))

วิธีการสุดท้าย ณ ที่นี้ ก็คือ การสร้างโครงสร้างแบบที่เรียกว่า ซีบีที โดยที่ซีบีทีจะทำการสร้างเส้นทางเดียวสำหรับส่งข้อมูลให้กลุ่มสมาชิก นั่นก็คือ มีการส่งข้อมูลไปยังจุด ๆ เดียวเท่านั้น ซึ่งจะเรียกจุด ๆ นั้นว่าคอร์ (Core) โดยกลุ่มของเราท์เตอร์จะทำการเลือกคอร์เราท์เตอร์ (Core Router) ดูรูปที่ 2.31 ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 ซีบีที

เนื่องจากซีบีทีมีเส้นทางเดียวในการส่งข้อมูลให้กับกลุ่มสมาชิก จึงทำให้เราท์เตอร์ไม่ต้องเก็บข้อมูลจำนวนมาก เมื่อเทียบกับวิธีอื่น ๆ และไม่จำเป็นต้องทำวิธีการฟลัดดิ้ง อย่างไรก็ตาม การที่มีเส้นทางเดียวก็อาจเกิดลักษณะที่เป็นคอขวดของเส้นทางในการส่งข้อมูลได้ วิธีการดังที่กล่าวมานี้ในแต่ละแบบก็จะมีข้อดีและข้อเสียแตกต่างกันไปในการที่จะนำมาพัฒนา เพื่อใช้สำหรับการส่งข้อมูล

2.5.4 มัลติคาสต์เราท์ติ้งโปรโตคอล (Multicast Routing Protocol)

ในหัวข้อที่ผ่านมา เราได้อธิบายถึงวิธีการที่สามารถจะนำมาใช้ได้ในมัลติคาสต์เราท์ติ้งโปรโตคอล ต่อไปเราจะดูว่าปัจจุบันมีมัลติคาสต์เราท์ติ้งโปรโตคอลแบบไหนบ้าง และในแต่ละวิธีได้นำเอาวิธีการดังกล่าวมาใช้อย่างไร

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

ดีวีเอ็มอาร์พีถูกกำหนดขึ้นครั้งแรกในอาร์เอฟซี1075 (RFC1075) โดยใช้หลักการพื้นฐานของวิธีที่เรียกว่า อาร์ไอพี (Routing Information Protocol (RIP)) แต่จะต่างกันที่อาร์ไอพีนั้นเป็นการส่งแพ็คเกตยูนิคาสต์ไปยังจุดถัดไปที่จะไปยังปลายทาง ในขณะที่ดีวีเอ็มอาร์พีเป็นการสร้างแผนผังต้นไม้บนข้อมูลของจุดก่อนหน้าที่จะไปยังต้นทาง ตอนแรกดีวีเอ็มอาร์พีนั้นได้ใช้วิธีการแบบทีอาร์พีบี ต่อมาพัฒนามาใช้วิธีแบบอาร์พีเอ็ม ส่วนมาตรฐานรุ่นล่าสุดของดีวีเอ็มอาร์พีกำหนดโดยกลุ่มการทำงานไอดีเอ็มอาร์ (Inter-Domain Multicast Routing (IDMR)) ในองค์กรไออีทีเอฟ

ดีวีเอ็มอาร์พีได้นำเอาวิธีการของอาร์พีเอ็มมาใช้ โดยข้อมูลชุดแรกจะถูกส่งออกจากไปในลักษณะของการกระจายไปทั่วเครือข่าย จากนั้นพรุนเมจเสจ (Prune Message) จะถูกส่งออกไป เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัดในจุดที่ไม่มีกลุ่มสมาชิกปรากฏอยู่ และมีการกำหนดข้อมูลใหม่ขึ้นมาที่เรียกว่า กราฟท์ (Graft) สำหรับใช้ในกรณีที่มีสมาชิกเข้ามาในกลุ่มที่เคยถูกตัดไป โดยการส่งข้อมูลดังกล่าวไปยังจุดที่ทำการตัดออกไป

ในกรณีที่มีเราท์เตอร์มากกว่าหนึ่งตัวในกลุ่มของสมาชิก จะมีเพียงตัวเดียวที่ถูกกำหนดให้เป็นตัวส่งข้อมูลมัลติคาสต์ โดยจะเลือกมาจากตัวที่อยู่ใกล้กับแหล่งกำเนิดข้อมูลมากที่สุด โดยที่เราท์เตอร์ตัวอื่น ๆ จะไม่สนใจข้อมูลที่ส่งมาจากแหล่งกำเนิดนี้ แต่ถ้าเราท์เตอร์ในกลุ่มดังกล่าวมีระยะทางไปยังแหล่งกำเนิดเท่ากัน ตัวที่มีไอพีแอดเดรสต่ำกว่าจะเป็นตัวที่ถูกเลือก

วิธีดีวีเอ็มอาร์พีนี้สามารถสนับสนุนการทำอุโมงค์ (Tunnel, การเชื่อมต่อระหว่างมัลติคาสต์เราท์เตอร์โดยมีการผ่านเราท์เตอร์ที่ไม่สามารถทำมัลติคาสต์) โดยที่ทุก ๆ อินเตอร์เฟสดังกล่าวนั้นจะมีการกำหนดไอพีแอดเดรสของเราท์เตอร์มัลติคาสต์ต้นทางและปลายทางสำหรับการทำอุโมงค์ดังกล่าว ส่วนขอบเขตของไอพีมัลติคาสต์สามารถกำหนดได้จากทีทีแอลฟิลด์ (TTL) ของไอพีซึ่งแสดงในตารางที่ 2.5

ทีทีแอลเทรสโฮล (TTL Threshold)	ขอบเขต
0	ในโฮสต์เดียวกัน
1	กลุ่มเครือข่ายเดียวกัน
15	ท้องถิ่นเดียวกัน
63	ภูมิภาค
127	ทั่วโลก
191	ทั่วโลก (จำกัดแบนวิดธ์)
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)) เอ็มโอเอสพีเอฟจะใช้ข้อมูลของกลุ่มสมาชิกผ่านวิธีการของไอจีเอ็มพี ในการสร้างโครงข่ายต้นไม้ สำหรับการส่งข้อมูล โดยโครงสร้างจะเป็นเส้นทางที่สั้นที่สุดในระหว่างจุดรับกับจุดส่ง โดยเอ็มโอเอสพีเอฟจะไม่สนับสนุนการทำอุโมงค์ แต่สามารถติดต่อกับเราท์เตอร์ที่ไม่ใช่เอ็มโอเอสพีเอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอ็มโอเอสพีเอฟจะสนับสนุนการทำเราท์ติ้งแบบลำดับชั้น โดยทุก ๆ โฮสต์ในอินเทอร์เนต จะถูกแบ่งออกเป็นเอเอส (Autonomous Systems (AS)) และแต่ละเอเอสจะมีการแบ่งเป็นกลุ่มย่อย ๆ ที่เราเรียกว่าเอเรีย (Areas) โดยจะมีการเราท์ติ้งสำหรับเอ็มโอเอสพีเอฟอยู่สามแบบ ดังต่อไปนี้

2.5.4.2.1 อินทราเอเรียเราท์ติ้ง (Intra-Area Routing)

เนื่องจากโอเอสพีเอฟนั้น เป็นเราท์ติ้งโปรโตคอลแบบลิ้งค์สเตท (Link-State, มีการดูสถานะของการเชื่อมต่อ) ที่ยอมให้มีการแบ่งเอเอสออกเป็นเอเรีย โดยที่ฐานข้อมูลของลิ้งค์สเตทจะปรากฏรายละเอียดของเอเรียดังกล่าวในทุก ๆ เราท์เตอร์ การหาและเก็บข้อมูลเกี่ยวกับตำแหน่งของสมาชิกของมัลติคาสต์ จะทำโดยการเพิ่มประเภทของลิ้งค์สเตทที่เรียกว่ากรุ๊ปเมมเบอร์ชิปแอลเอสเอ (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 เมื่อเราท์เตอร์ได้รับมัลติคาสต์ดาต้าแกรมระหว่างต้นกำเนิดกับกลุ่มสมาชิก ก็จะทำการสร้างโครงสร้างต้นไม้ดังกล่าวขึ้นมา จากข้อมูลนี้ทำให้เราท์เตอร์ทราบว่า อินเตอร์เฟสไหนจะเป็นตัวรับหรือส่งข้อมูล ซึ่งจะต่างจากดีวีเอ็มอาร์พีที่แพ็คเกตแรกจะไม่ได้ถูกส่งออกไปทุก ๆ อินเตอร์เฟส

2.5.4.2.2 อินเทอร์เอเรียเราท์ติ้ง (Inter-Area Routing)

ถ้าแหล่งรับหรือส่งข้อมูลไม่ได้อยู่ในเอเรียเดียวกัน วิธีการที่ผ่านมาจะไม่สามารถใช้ได้ การแก้ไขสามารถทำได้โดยเลือกเราท์เตอร์ที่เป็นเอเรียบอร์เดอร์เราท์เตอร์ (Area Border Router) สำหรับทำหน้าที่ส่งข้อมูลมัลติคาสต์ ระหว่างกลุ่มอินเทอร์เอเรียมัลติคาสต์ฟอเวอร์ดเดอร์ (Inter Area Multicast Forwarders) ซึ่งจะมีหน้าที่ในการส่งข้อมูลของกลุ่มทั้งหมดไปยังแบคโบนเอเรีย (Backbone Area) และด้วยแนวคิดของไวล์ดการ์ดมัลติคาสต์รีซีฟเวอร์ (Wild-Card Multicast Receiver) ในเอ็มโอเอสพีเอฟ ทำให้มันรับทุก ๆ มัลติคาสต์เมจเสจในเอเรียนั้น ๆ และอินเทอร์เอเรียมัลติคาสต์ในเอเรียที่ไม่มีแบคโบนไว้ และเพื่อให้แน่ใจว่าทุก ๆ ข้อมูลในเอเรียที่ไม่มีแบคโบนสามารถที่จะส่งไปถึงอินเทอร์เอเรียมัลติคาสต์ฟอเวอร์ดเดอร์ และถ้าจำเป็นมันจะส่งต่อไปยังแบคโบนเอเรีย ดังนั้นในแบคโบนก็จะมีข้อมูลของกลุ่มสมาชิกที่อยู่ในเอเรียต่าง ๆ อย่างครบถ้วน และข้อมูลมัลติคาสต์ก็สามารถที่จะถูกส่งไปยังเอเรียที่เหมาะสมในเอเอสนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4.2.3อินเทอร์เอเอสเราท์ติ้ง (Inter-AS Routing)

การรับส่งข้อมูลมัลติคาสต์ระหว่างเอเอสที่ต่างกัน จะคล้าย ๆ กับอินเทอร์เอเรียเราท์ติ้ง โดย เอเอสบีอาร์ (AS Boundary Routers (ASBRs)) จะถูกกำหนดให้ทำหน้าที่เป็นอินเทอร์เอเอสมัลติคาสต์ฟอเวอร์ดเดอร์ โดยเอ็มโอเอสพีเอฟจะสร้างอาร์พีบีทรีสำหรับส่งข้อมูลมัลติคาสต์ ซึ่งอินเทอร์เอเอสมัลติคาสต์ฟอเวอร์ดเดอร์นี้จะเป็นไวล์ดการ์ดมัลติคาสต์ของทางด้านรับในเอเรียที่ติดกัน และรับประกันว่าเราท์เตอร์ยังคงอยู่ในโครงสร้างของการส่งและรับข้อมูล ในขณะที่เส้นทางการส่งข้อมูลที่ใช้ภายในเอเอสไปยังแหล่งข้อมูลภายนอก จะใช้วิธีที่เราเรียกกันว่า รีเวอร์สพาร์ทซอร์สเบสทรี (Reverse-Path Source-Based Trees)

2.6 การเขียนโปรแกรมจาวาด้วยเนทีฟเมธอด

การเขียนเนทีฟเมธอดสำหรับโปรแกรมด้วยภาษาจาวามีขั้นตอนดังนี้

1. เริ่มต้นโดยการเขียนโปรแกรมด้วยภาษาจาวา เนื่องจากการสร้างคลาสจาวาที่มีการประกาศเนทีฟเมธอดนั้น จะต้องประกอบด้วยการประกาศเนทีฟเมธอด หรือระบุชนิดของตัวแปรของเนทีฟเมธอด รวมทั้งภายในเมธอดเมน (Main Method) ก็จะต้องเขียนให้มีการเรียกเนทีฟเมธอดด้วย
2. คอมไพล์คลาสจาวาที่มีการประกาศเนทีฟเมธอดและเมธอดเมน
3. สร้างไฟล์เฮดเดอร์ที่มีรูปแบบสำหรับเนทีฟเมธอด โดยใช้คำสั่ง javah -jni ซึ่งจะสร้างไฟล์เฮดเดอร์ที่มีชนิดของตัวแปรสำหรับเนทีฟเมธอดนั้น
4. เขียนเนทีฟเมธอดด้วยภาษาโปรแกรมที่เลือกไว้ เช่น ภาษาซี (C) หรือซีพลัซพลัซ (C++)
5. คอมไพล์ไฟล์เฮดเดอร์และสร้างไฟล์เพื่อทำการแลกเปลี่ยนไฟล์ไลบรารี
6. รันโปรแกรมจาวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.32 แสดงขั้นตอนการสร้างโปรแกรม 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 การสร้างเมธอดเมน และส่วนของโค้ดสแตติก

2.6.1.1 การประกาศเนทีฟเมธอด

เราต้องประกาศเมธอดทั้งหมด ไม่ว่าจะเป็นจาวาเมธอดหรือเนทีฟเมธอดภายในคลาสที่เขียนด้วยภาษาจาวา และเมื่อสร้างเมธอดจากภาษาอื่นที่ไม่ใช่ภาษาจาวา จะต้องเพิ่มคำประกาศ native เข้าไปในการเขียนเมธอดภายในคลาสจาวาด้วย โดยคำประกาศ native จะเป็นตัวบอกให้คอมไพเลอร์ทราบว่า ฟังก์ชันนั้นเป็นฟังก์ชันที่สร้างมาจากภาษาอื่น ซึ่งทำให้เราเข้าใจได้ง่ายว่า เมธอด displayHelloWorld ของ คลาส HelloWorld นั้น ถูกเขียนขึ้นมาด้วยภาษาอื่น เพราะคำประกาศ native บอกให้เราทราบถึงคำจำกัดความที่มาของเมธอดอยู่แล้ว

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประกาศเนทีฟเมธอดในคลาสภาษาจาวาตรงส่วนนี้ เราจะนำไปใช้ในเมธอด displayHelloWorld เท่านั้น ไม่ต้องสร้างเมธอดนั้นด้วยภาษาจาวา แต่จะสร้างเมธอดด้วยภาษาอื่นตามที่ต้องการแทน ซึ่งการประกาศเมธอดสำหรับ displayHelloWorld จะประกาศเมธอดเป็นประเภทสาธารณะ (Public) ไม่ยอมรับค่าใด ๆ และไม่มีการส่งค่ากลับ

2.6.1.2 การโหลดไลบรารี

เมื่อเราคอมไพล์โค้ดโปรแกรมภาษาอื่น มันจะสร้าง displayHelloWorld เป็นแชร์ไลบรารี ซึ่งในขณะรันระบบ จะมีการโหลดแชร์ไลบรารีนี้เข้ามาที่คลาสจาวา การโหลดไลบรารีเข้าไปยังจาวาคลาสจะทำการกำหนดตามการประกาศเนทีฟเมธอดที่ประกาศในคลาสจาวา

คลาส Helloworld จะใช้เมธอด System.loadLibrary ซึ่งจะทำการโหลดแชร์ไลบารี่ โดยแชร์ไลบารี่นี้จะถูกสร้างเมื่อทำการคอมไพล์ ซึ่งจะวางไว้ในตำแหน่งของส่วนสแตติกเริ่มต้น ค่าที่ส่งไปยัง System.loadLibrary จะเป็นชื่อของแชร์ไลบรารีซึ่งจะเป็นชื่ออะไรก็ได้ โดยระบบจะเปลี่ยนชื่อของแชร์ไลบารีไปเป็นชื่อเนทีฟไลบรารีตามมาตรฐานของแต่ละแพลตฟอร์ม ยกตัวอย่างเช่น ในระบบโซลาริส (Solaris system) มันจะเปลี่ยนชื่อไลบรารี "hello" ไปเป็น libhello.so ขณะที่ระบบของไมโครซอฟท์วินโดว์ (Microsoft Windows system) จะเปลี่ยนเป็นชื่อเดียวกันคือ hello.dll

ค่าสแตติกเริ่มต้นจากคลาส Helloworld จะถูกโหลดไปยังไลบรารีที่เหมาะสม ซึ่งในที่นี้ก็คือ ชื่อ hello โดยระบบปฏิบัติการจะทำการรันค่าสแตติกเริ่มต้นนี้เมื่อมันโหลดคลาส

```
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 โดยชื่อของเมธอดที่ถูกเรียกจะต่อท้ายชื่อวัตถุ ซึ่งจะคั่นด้วยจุด (.) และต้องมีวงเล็บ () 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 ใหม่ไว้ที่ไดเรกทอรีเดียวกับไฟล์ .class ซึ่งเราสามารถใช้คำสั่ง -d เพื่อบอกกับ javah ว่าต้องการจะวางเฮดเดอร์ไฟล์ไว้ที่ไดเรกทอรีอื่น

2.6.3.1 การประกาศฟังก์ชัน

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

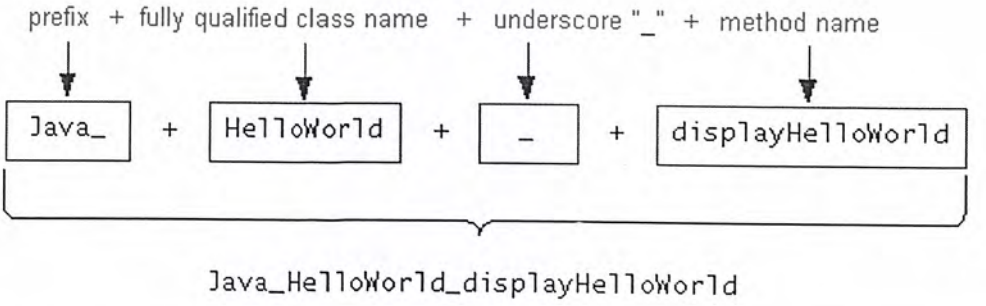
```
#endif
/*
 * Class:     HelloWorld
 * Method:    displayHelloWorld
 * Signature: ()V
 */

JNIEXPORT void JNICALL Java_HelloWorld_displayHelloWorld
  (JNIEnv *, jobject);

#ifdef __cplusplus
}
#endif
#endif
```

จากตัวอย่างไฟล์เฮดเดอร์ HelloWorld.h ฟังก์ชัน Java_HelloWorld_displayHelloWorld จะระบุถึงเนทีฟเมธอด displayHelloWorld ของคลาส HelloWorld ซึ่งเราจะสร้างขึ้นในข้อ 2.6.4 ซึ่งสามารถใช้ลักษณะของฟังก์ชันแบบเดียวกันนี้ได้ เมื่อต้องการสร้างเนทีฟเมธอด ถ้า HelloWorld มีเนทีฟเมธอดอื่น ๆ อีก ฟังก์ชันนั้นก็จะปรากฏที่นี่เช่นเดียวกัน

ชื่อของเนทีฟฟังก์ชันที่ถูกสร้างจากเนทีฟเมธอด จะมี Java_ นำหน้า ตามด้วยชื่อแพ็กเกต, ชื่อคลาส และชื่อของเนทีฟเมธอด โดยระหว่างชื่อของแต่ละส่วนนั้น จะถูกคั่นด้วยขีดเส้นใต้ (_) ดังรูปที่ 2.33



รูปที่ 2.33 แสดงให้เห็นถึงรูปแบบชื่อของเนทีฟฟังก์ชันที่ถูกสร้างจากเนทีฟเมธอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนทีฟเมธอด displayHelloWorld ภายในคลาส HelloWorld นี้ จะถูกเปลี่ยนชื่อเป็น Java_HelloWorld_displayHelloWorld จากตัวอย่างจะไม่มีชื่อแพคเก็จปรากฏอยู่ เนื่องจากคลาส HelloWorld อยู่ภายในดิฟอลท์แพคเก็จ (Default package) จึงไม่มีชื่อ

การสร้างเนทีฟฟังก์ชันอย่างที่ปรากฏในไฟล์เฮดเดอร์ จะมีการรับพารามิเตอร์ 2 ตัว โดยทางด้านจาวานั้น จะไม่มีการรับค่าพารามิเตอร์ แต่เจเอ็นไอ (JNI) จำเป็นที่จะต้องกำหนดให้ทุก ๆ เนทีฟเมธอด ต้องมีการรับค่าพารามิเตอร์ 2 ตัว

พารามิเตอร์แรกของทุก ๆ เนทีฟเมธอดคือ JNIEnv ซึ่งเป็นอินเทอร์เฟสพอยเตอร์ (Interface Pointer) โดยค่าต่าง ๆ จะถูกส่งผ่านพอยเตอร์ เมื่อเนทีฟโค้ดทำการแอสเซส (access) มายังพารามิเตอร์นั้น ส่วนอ๊อบเจคจะถูกส่งผ่านพอยเตอร์ไปยังจาวาแอพพลิเคชั่น พารามิเตอร์สอง คือ jobject ซึ่งจะอ้างอิงถึงอ๊อบเจคปัจจุบัน ซึ่งเราสามารถเทียบพารามิเตอร์ jobject เท่ากับตัวแปร "this" ในภาษาจาวา สำหรับเนทีฟอินสแตนเมธอด ซึ่งได้แก่ เมธอด displayHelloWorld ในตัวอย่าง jobject จะเป็นตัวแปรเพื่ออ้างถึงค่าปัจจุบันของอ๊อบเจค สำหรับเนทีฟคลาสเมธอดตัวแปร jobject จะเป็นการอ้างอิงเมธอดของคลาสจาวานั้น ซึ่งจากตัวอย่างนั้นได้ละเว้นตัวแปรทั้งสองไว้

2.6.4 การสร้างเนทีฟเมธอด

ในขั้นตอนนี้เราจะทำการสร้างเนทีฟเมธอดจากภาษาอื่นที่ไม่ใช่ภาษาจาวา ซึ่งฟังก์ชันที่เขียนขึ้นจำเป็นที่จะต้องมีรูปแบบเดียวกับฟังก์ชันที่ได้จากการใช้คำสั่ง javah ในไฟล์ HelloWorld.h จากข้อ 2.6.3 โดยรูปแบบที่ได้จากการสร้างเนทีฟเมธอด displayHelloWorld ของคลาส HelloWorld จะมีรูปแบบดังนี้

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

จากจุดนี้ เราสามารถที่จะใช้ภาษาซี (C) ในการเขียนโปรแกรมเพื่อทำการสร้างเนทีฟเมธอด 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) กลับมา

ไฟล์เฮดเดอร์ที่ประกอบใน HelloWorldImp.c มีสามไฟล์ ได้แก่

1. jni.h เป็นไฟล์เฮดเดอร์ที่ให้ข้อมูลว่า ต้องการเนทีฟโค้ดตัวใดเมื่อต้องการจะติดต่อกับระบบปฏิบัติการจาวา เมื่อเราสร้างเนทีฟเมธอด จะต้องรวมไฟล์นี้ใส่ลงไปในเนทีฟซอร์สไฟล์ด้วย
2. HelloWorld.h เป็นไฟล์ .h ที่ได้มาจากข้อ 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 ซึ่งเราได้เขียนไว้แล้วในเนทีฟเมธอด displayHelloWorld โดยจะเก็บเนทีฟเมธอดไว้ในไฟล์ HelloWorldImp.c ขั้นตอนนี้ เราจะต้องทำการคอมไพล์ไฟล์ HelloWorldImp.c ไปยังแชร์ไลบรารีที่มีชื่อว่า hello เพื่อให้สอดคล้องกับชื่อไลบรารีที่ใช้ในเมธอด System.loadLibrary

จากนั้นทำการคอมไพล์โค้ดเนทีฟที่สร้างขึ้นในขั้นตอนก่อนไปยังแชร์ไลบรารี โดยในระบบโซราลิซจะเป็นการสร้างแชร์ไลบรารี ส่วนในระบบวินโดว์จะทำการสร้างไฟล์ dll จากนั้นเราจะต้องระบุพาธลงไปในส่วนที่เกียวข้องหรือระบุลงในไฟล์เฮดเดอร์ที่จำเป็นทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบโซราลิซจะใช้คำสั่งดังต่อไปนี้ในการสร้างแชร์ไลบรารี libhello.so

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

ในระบบวินโดว์ จะใช้คำสั่งต่อไปนี้เพื่อสร้างไฟล์ hello.dll โดยใช้โปรแกรมไมโครซอฟต์วิชวลซีพลัสพลัส 4.0 (Microsoft Visual C++ 4.0)

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

2.6.6 การรันโปรแกรม

รันจาวาแอพพลิเคชั่น (คลาส HelloWorld) ด้วยตัวแปรภาษาจาวา โดยใช้คำสั่งนี้

```
java HelloWorld
```

โดยจะได้ผลลัพธ์ดังนี้

```
Hello World!
```

แต่ถ้ามีเอ็กเซ็บชัน (Exception) ดังตัวอย่างนี้ แสดงว่ายังไม่ได้มีการเซ็ตพาธให้ถูกต้อง

```
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)
```

ไลบรารีพาธจะเป็นรายการของไดเรกทอรีทั้งหมด ที่ระบบปฏิบัติการจาวาจะทำการค้นหาเมื่อต้องการโหลดไลบรารี โดยต้องแน่ใจว่าไลบรารี hello มีอยู่ในนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

3.1 ข้อมูลเบื้องต้น

ทำการเก็บข้อมูลการใช้งานระบบวิทยุกระจายเสียงผ่านอินเทอร์เนตจากผู้ใช้ทั่วไป และจากการศึกษาแอพพลิเคชั่นตัวอย่าง ทำให้สามารถออกแบบฟังก์ชันต่าง ๆ ออกมาได้ ดังนี้

3.1.1 ฟังก์ชันสำหรับผู้ดูแลระบบ (Administrator)

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

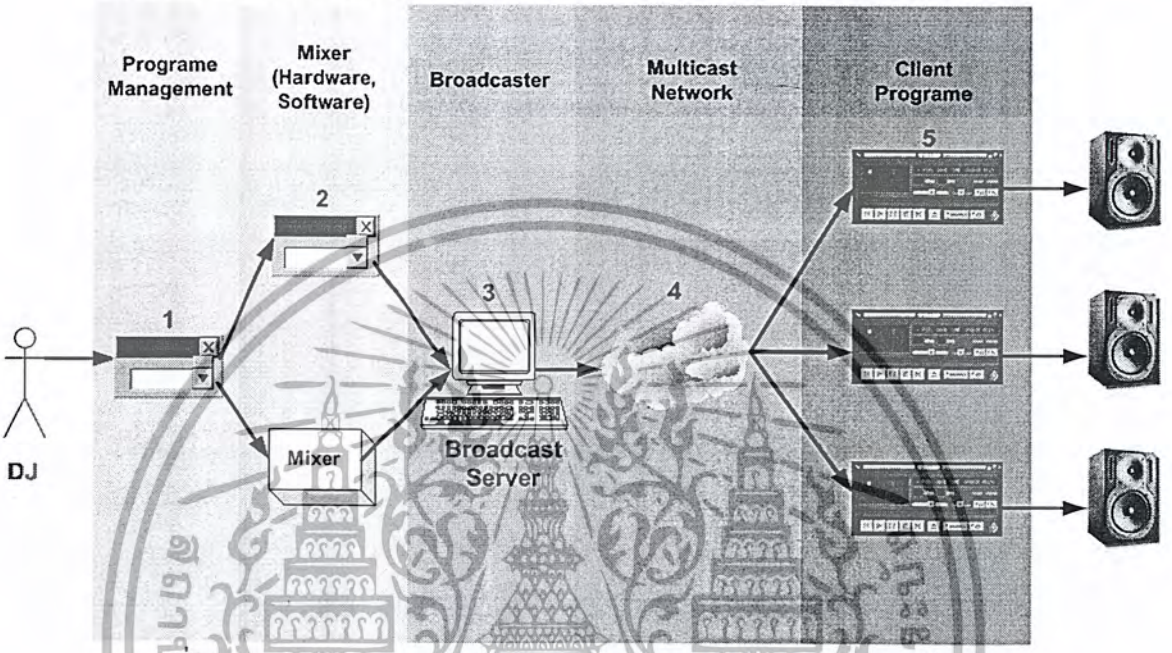
3.1.2 ฟังก์ชันสำหรับผู้ดำเนินรายการ (DJ)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบระบบ

3.2.1 การออกแบบสถาปัตยกรรมของระบบ (System Architecture Design)



รูปที่ 3.1 สถาปัตยกรรมของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เนต

3.2.1.1 ส่วนจัดการผังรายการ (Program Management Module)

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

3.2.1.2 ส่วนมิกซ์เซอร์ (Mixer)

เป็นส่วนควบคุมระดับความดังค่อย และผสมเสียงที่จะส่งไปกระจายเสียงที่ส่วนบรอดคาสท์เซิร์ฟเวอร์ ในส่วนนี้ออกแบบให้สามารถใช้ได้ทั้งมิกซ์เซอร์ที่เป็นฮาร์ดแวร์ และมิกซ์เซอร์ที่เป็นซอฟแวร์

3.2.1.3 ส่วนบรอดคาสซ์เซิร์ฟเวอร์ (Broadcast Server)

ทำหน้าที่รับข้อมูลเสียงจากมิกซ์เซอร์ และกระจายเสียงเหล่านั้นไปในเครือข่ายอินเทอร์เนตโดยมีรูปแบบการส่งแบบมัลติคาสท์ โดยระบุเป้าหมายเป็นมัลติคาสท์แอดเดรสหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.4 ส่วนเครือข่ายมัลติคาสท์ (Multicast Network)

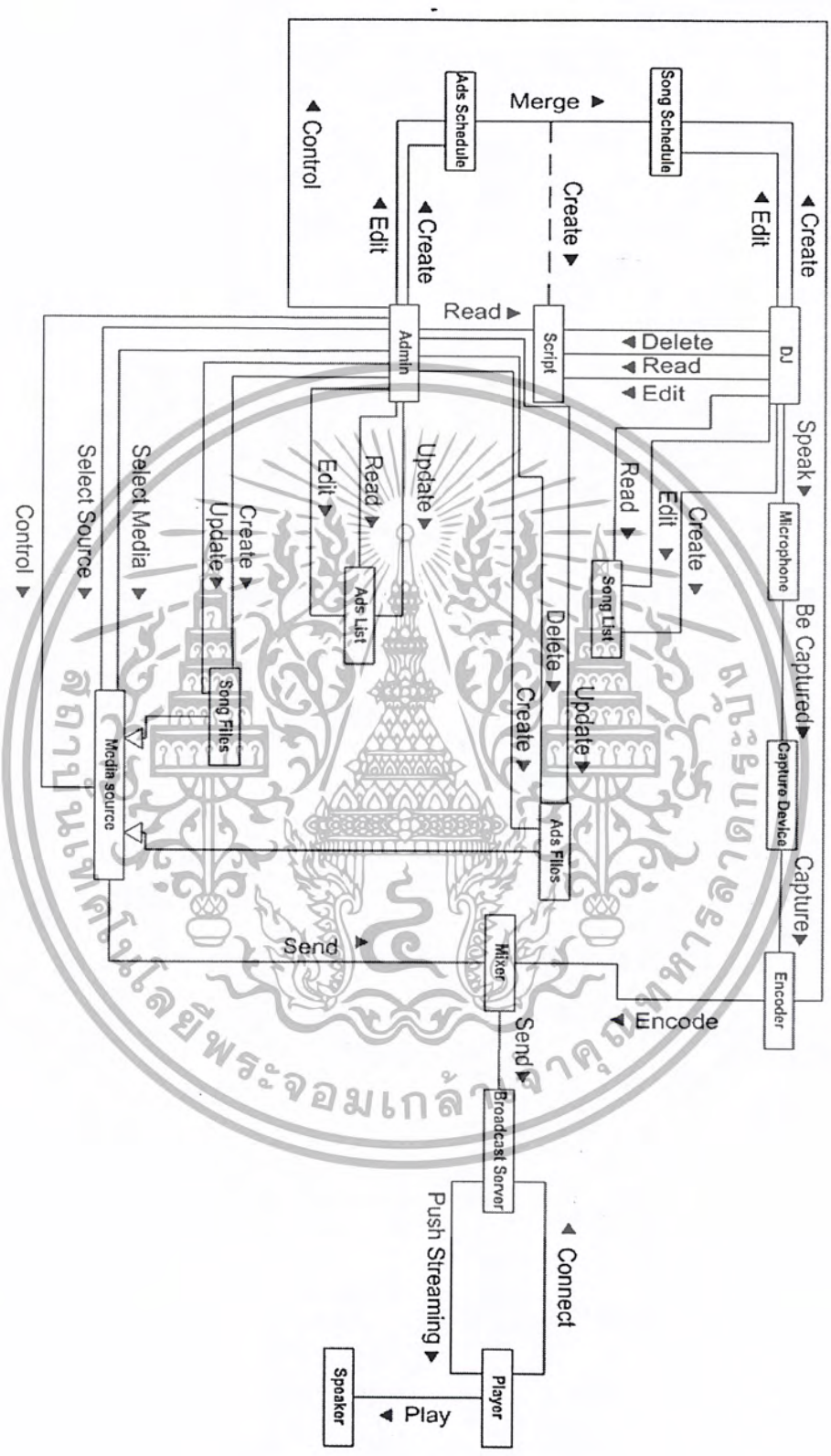
ประกอบด้วยมัลติคาสท์เราเตอร์ (Multicast Router) ที่สนับสนุนไอพีมัลติคาสท์ และการเชื่อมต่อของไคล์เอนท์มายังไอพีมัลติคาสท์

3.2.1.5 ส่วนไคล์เอนท์โปรแกรม

ออกแบบให้ใช้โปรแกรมวินแอมป์ (Winamp) เนื่องจากเป็นโปรแกรมสำหรับฟังเพลงที่นิยมใช้กันทั่วไป และสามารถรองรับการเชื่อมต่อแบบติคาสท์และใช้โปรโตคอลอาร์ทีพีได้

3.2.2 โดเมนโมเดล (Domain Model)

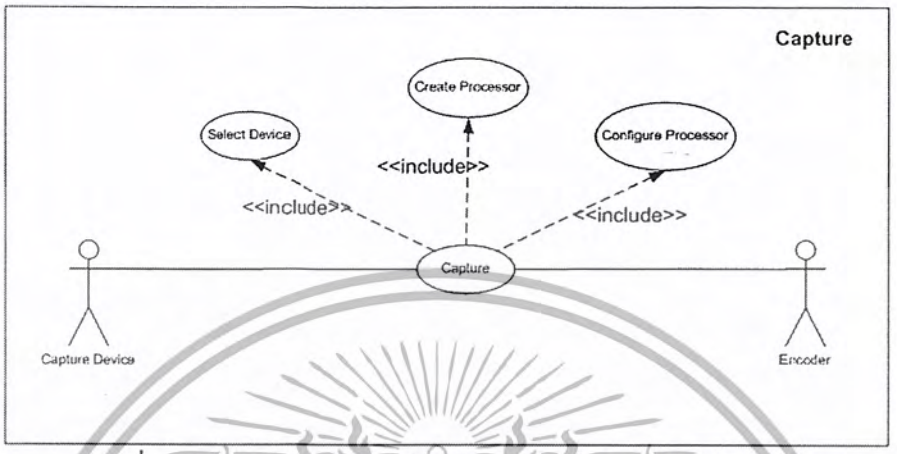
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



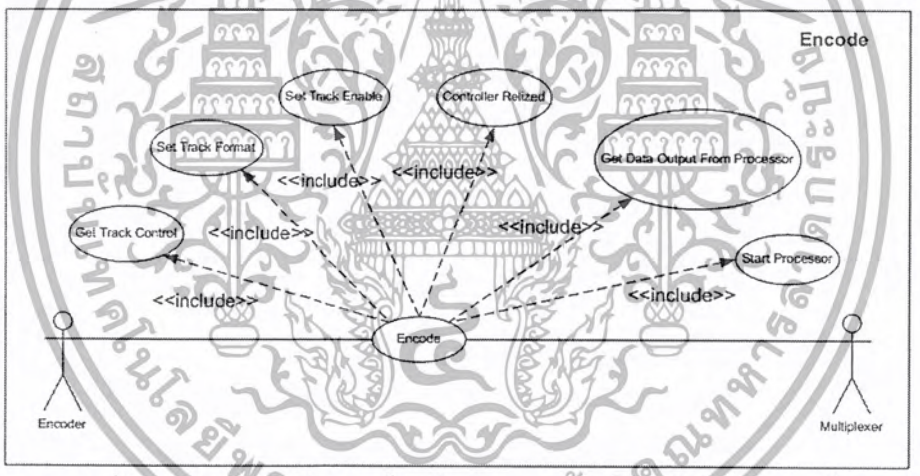
รูปที่ 3.2 โดเมนโมเดลของระบบจัดการการกระจายเสียงผ่านอินเทอร์เนต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

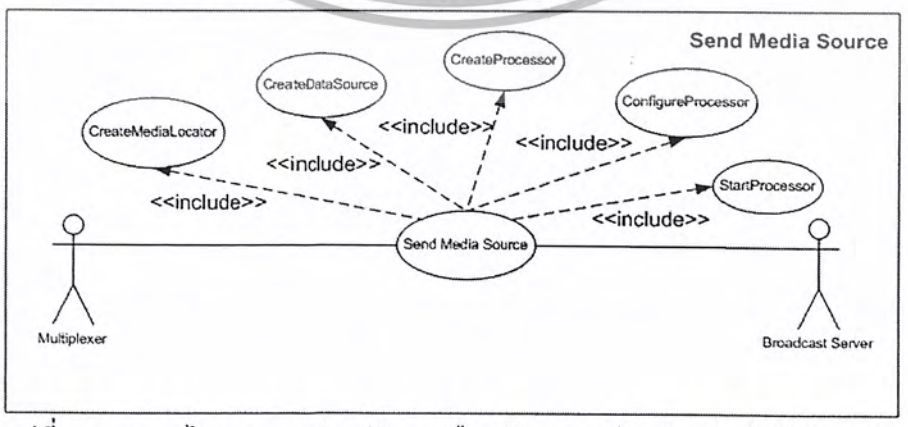
3.2.3 ยูสเคสไดอะแกรม (Use Case Diagram)



รูปที่ 3.3 ยูสเคสไดอะแกรมสำหรับการจับเสียงจากไมโครโฟน

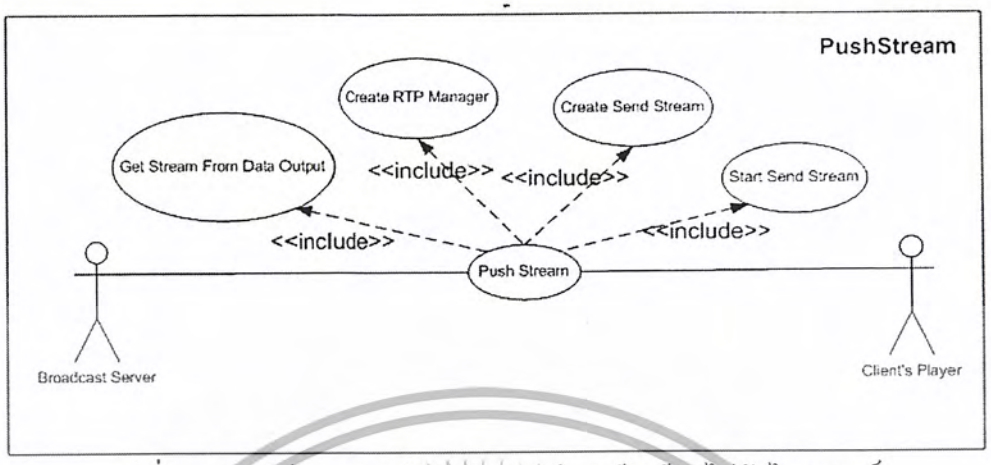


รูปที่ 3.4 ยูสเคสไดอะแกรมสำหรับการเข้ารหัสสัญญาณเสียงให้อยู่ในรูปแบบแพ็กเกตอาร์ทีพี



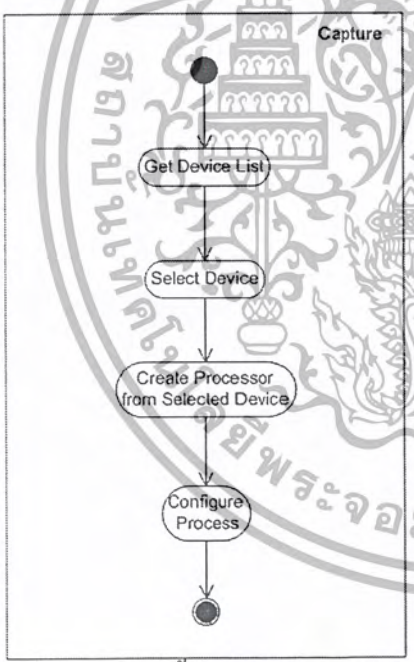
รูปที่ 3.5 ยูสเคสไดอะแกรมสำหรับการเลือกสัญญาณเสียงสำหรับส่งมีเดียสตรีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

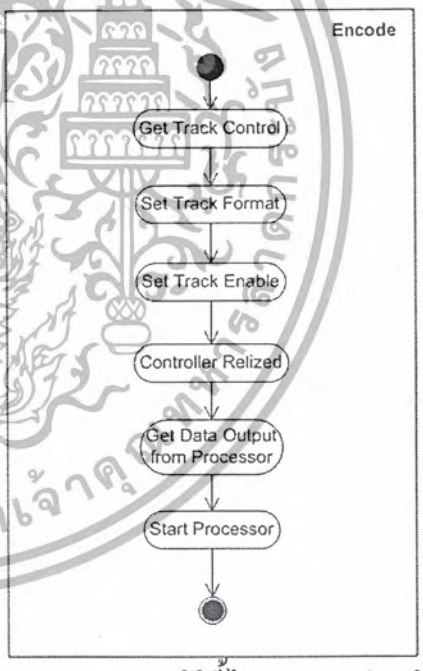


รูปที่ 3.6 ยูสเคส ไดอะแกรมสำหรับการส่งสตรีมเสียงไปยังไคลเอนท์

3.2.4 แอคทิวิตี้ไดอะแกรม (Activity Diagram)

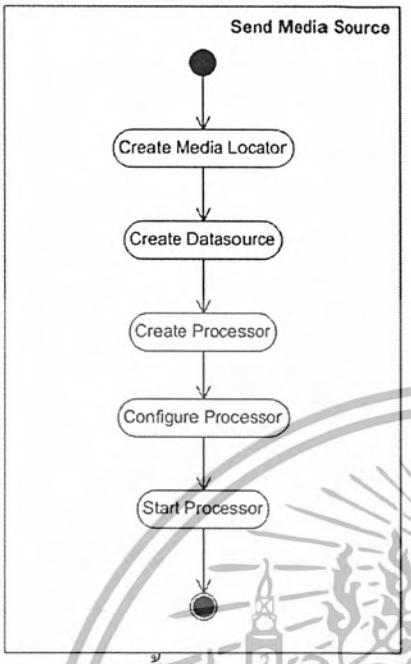


รูปที่ 3.7 แอคติวิตี้ไดอะแกรมสำหรับการจับเสียงจากไมโครโฟน

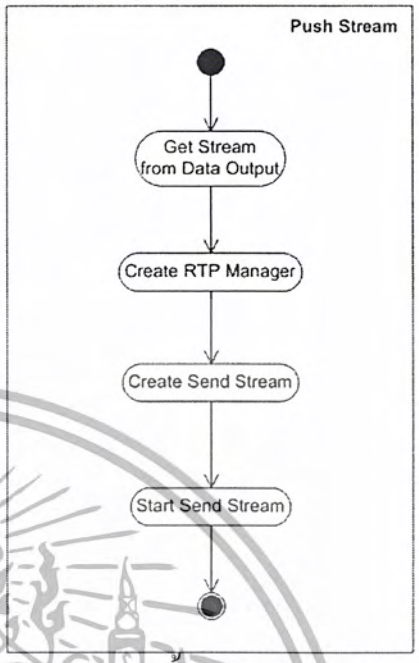


รูปที่ 3.8 แอคติวิตี้ไดอะแกรมสำหรับการเข้ารหัสสัญญาณเสียงให้อยู่ในรูปแบบแพ็กเกตอาร์ทีพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แอคติวิตี้ไดอะแกรมสำหรับการเลือกสัญญาณเสียงสำหรับส่งมีเดียสตรีม



รูปที่ 3.10 แอคติวิตี้ไดอะแกรมสำหรับการการส่งสตรีมเสียงไปยังไคลเอนท์

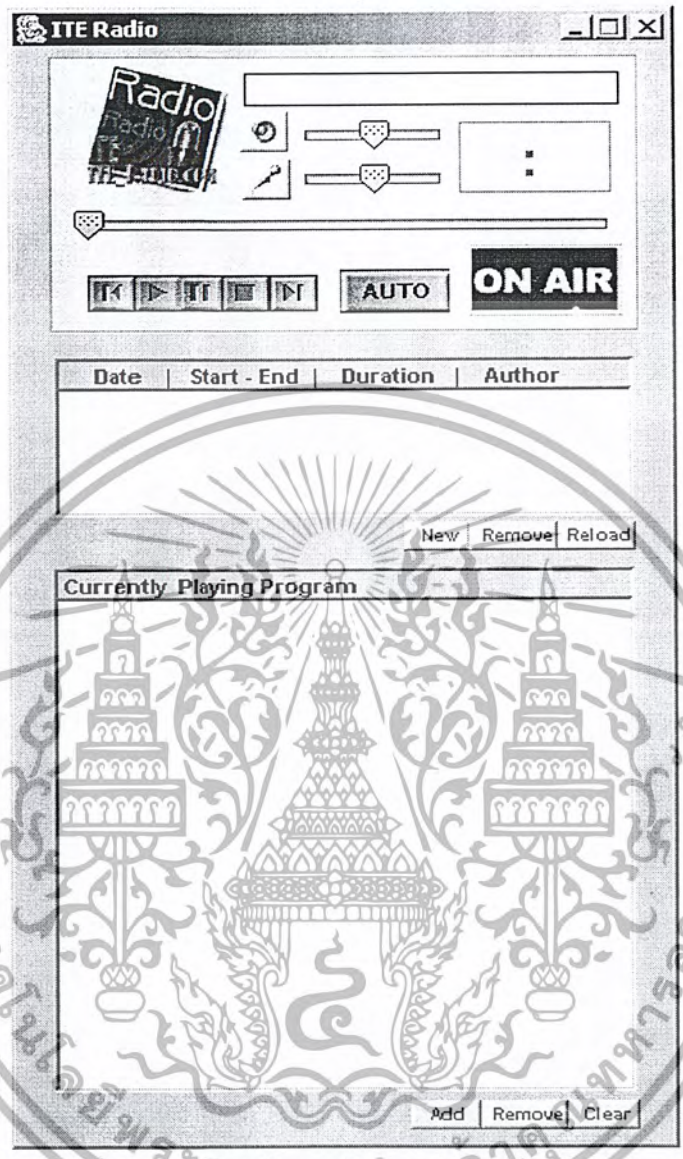
3.2.5 การออกแบบส่วนติดต่อผู้ใช้ (User Interface)

3.2.5.1 หน้าต่างหลัก

หน้าต่างหลักมีหน้าที่เป็นเครื่องมือช่วยในการควบคุมการออกอากาศและรายงานสถานะปัจจุบันของระบบ ประกอบด้วยสามส่วนดังนี้

1. ส่วนควบคุมการออกอากาศ
2. ส่วนแสดงผังรายการล่วงหน้า
3. ส่วนรายละเอียดของผังรายการปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 หน้าต่างหลักสำหรับการจัดการการจายเสียงผ่านอินเทอร์เนต

3.2.5.1.1 ส่วนควบคุมการออกอากาศ

1. ส่วนแสดงสถานการณ์ทำงานปัจจุบันของแอพพลิเคชัน ประกอบด้วย
   1) ชื่อมีเดียที่กำลังเล่น (Now Playing)
   2) แสดงตำแหน่งทางเวลาของมีเดียที่กำลังเล่น
   3) สไลด์บาร์ (Slider Bar) แสดงตำแหน่งทางกายภาพของการเล่นเพลงในมีเดียหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ 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) ปุ่มพรีเวียส (Previous Button)
    5) ปุ่มเนกซ์ (Next Button)
4. ส่วนเลือกโหมดการทำงาน เลือกได้สองโหมด คือ
    1) โหมดการทำงานอัตโนมัติ (Auto Mode)
    2) โหมดควบคุมด้วยมือ (Manual Mode)
5. ส่วนเปิดปิดการกระจายเสียง คือ
    1) ปุ่มออนแอร์ (On Air Button)

3.2.5.1.2 ส่วนแสดงผังรายการล่วงหน้า

1. ลิสท์แสดงรายการของผังรายการล่วงหน้า แสดงรายละเอียดเบื้องต้นดังนี้
    1) วันที่ที่กำหนดให้ออกอากาศ (Date)
    2) ช่วงเวลาที่กำหนดให้ออกอากาศ (Start-End)
    3) ความยาวของรายการที่สร้าง (Duration)
    4) ชื่อผู้สร้างลิสท์ (Author)
2. ปุ่มนิวลิสท์ (New List Button)
3. ปุ่มรีมูฟ (Remove Button)
4. ปุ่มอิดิท (Edit Button)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-

3.2.5.1.3 ส่วนแสดงผังรายการปัจจุบัน

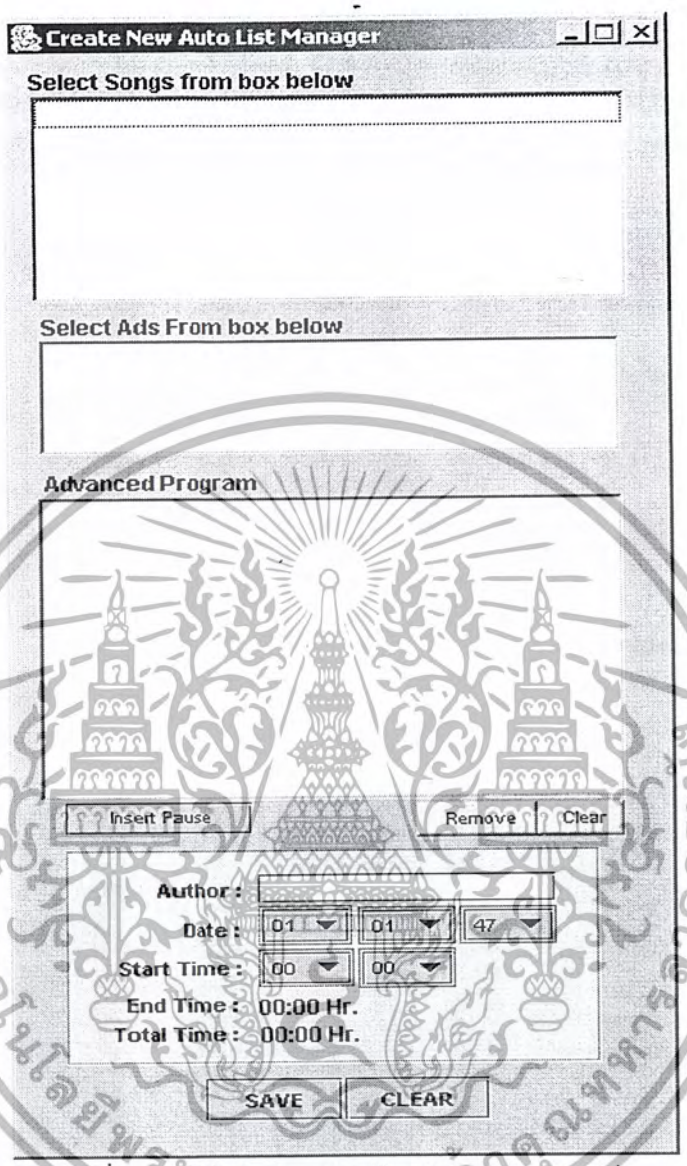
1. ลิสท์แสดงรายการมีเดียที่เลือกไว้ แสดงรายละเอียดของมีเดีย ดังนี้
    1) ชื่อมีเดีย (Media Name)
    2) ความยาวของมีเดีย (Media Duration)
2. ปุ่มบราวซ์ (Browse Button)
3. ปุ่มรีมูฟ (Remove Button)
4. ปุ่มเคลียร์ (Clear Button)

3.2.5.2 หน้าต่างสำหรับสร้างผังรายการล่วงหน้า

หน้าต่างสำหรับสร้างผังรายการล่วงหน้า ดังรูปที่ 3.13 ประกอบด้วยสี่ส่วนหลักคือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 หน้าต่างสำหรับสร้างผังรายการล่วงหน้า

3.2.5.2.1 ส่วนแสดงรายชื่อแหล่งข้อมูลเสียง

1. ลิสท์สำหรับเลือกแหล่งข้อมูลเสียงเพลง (Song List) แสดง
   1) ชื่อมีเดีย (Media Name)
   2) ความยาวของมีเดีย (Media Duration)
2. ลิสท์สำหรับเลือกแหล่งข้อมูลเสียงโฆษณา (Ads List) แสดง
   1) ชื่อมีเดีย (Media Name)
   2) ความยาวของมีเดีย (Media Duration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5.2.2 ส่วนแสดงรายการที่สร้าง

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

3.2.5.2.3 ส่วนกำหนดการออกอากาศ

1. กล่องข้อความสำหรับกรอกชื่อผู้สร้างลิสท์ (Author)
2. คอมโบบ็อกซ์สำหรับเลือกวันที่ออกอากาศ (Date)
3. คอมโบบ็อกซ์สำหรับเลือกเวลาเริ่มออกอากาศ (Start Time)
4. ลาเบลแสดงผลการคำนวณเวลาสิ้นสุดการออกอากาศ (End-Time)
5. ลาเบลแสดงผลการคำนวณเวลารวมที่ใช้ออกอากาศ (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 แบ่งเป็นสี่ส่วนหลักดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5.3.1 ส่วนแสดงรายชื่อแหล่งข้อมูลเสียง

1. ลิสท์สำหรับเลือกแหล่งข้อมูลเสียงเพลง (Song List) แสดง
    1) ชื่อมีเดีย (Media Name)
    2) ความยาวของมีเดีย (Media Duration)
2. ลิสท์สำหรับเลือกแหล่งข้อมูลเสียงโฆษณา (Ads List) แสดง
    1) ชื่อมีเดีย (Media Name)
    2) ความยาวของมีเดีย (Media Duration)

3.2.5.3.2 ส่วนแสดงรายการที่สร้าง

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

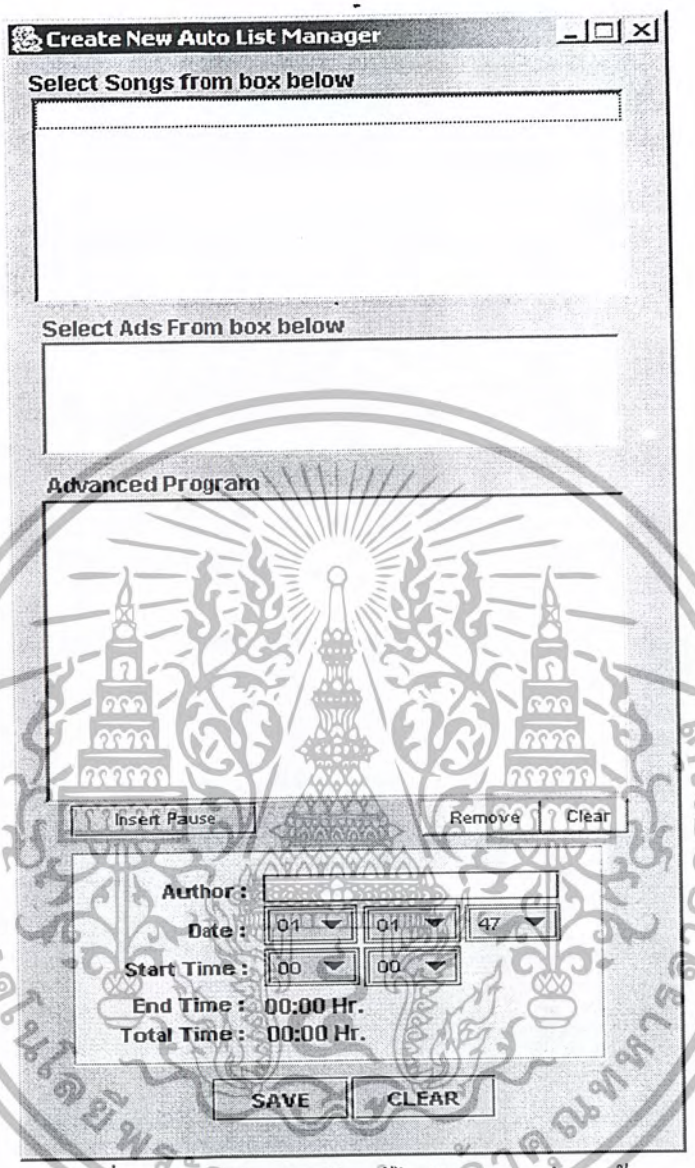
3.2.5.3.3 ส่วนกำหนดการออกอากาศ

1. กล่องข้อความสำหรับกรอกชื่อผู้สร้างลิสท์ (Author)
2. คอมโบบ็อกซ์สำหรับเลือกวันที่ออกอากาศ (Date)
3. คอมโบบ็อกซ์สำหรับเลือกเวลาเริ่มออกอากาศ (Start Time)
4. ลาเบลแสดงผลการคำนวณเวลาสิ้นสุดการออกอากาศ (End Time)
5. ลาเบลแสดงผลการคำนวณเวลารวมที่ใช้ออกอากาศ (Total Time)

3.2.5.3.4 ส่วนบันทึกผังรายการลงไฟล์

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 หน้าต่างสำหรับแก้ไขผังรายการล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ทดลองเรียกแอพพลิเคชั่น

เมื่อทำการเรียกแอพพลิเคชัน จะปรากฏหน้าต่างหลัก (Internet Radio Manager Window) ขึ้นดังรูปที่ 4.1

รูปที่ 4.1 หน้าต่างหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างหลักจะแสดงสถานะการทำงานจะแสดงข้อมูลที่เป็นอยู่จริง ณ เวลาใด ๆ ตารางที่ 4.1 แสดงสถานะเริ่มต้นถูกกำหนดไว้เมื่อเริ่มต้นแอพพลิเคชัน

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

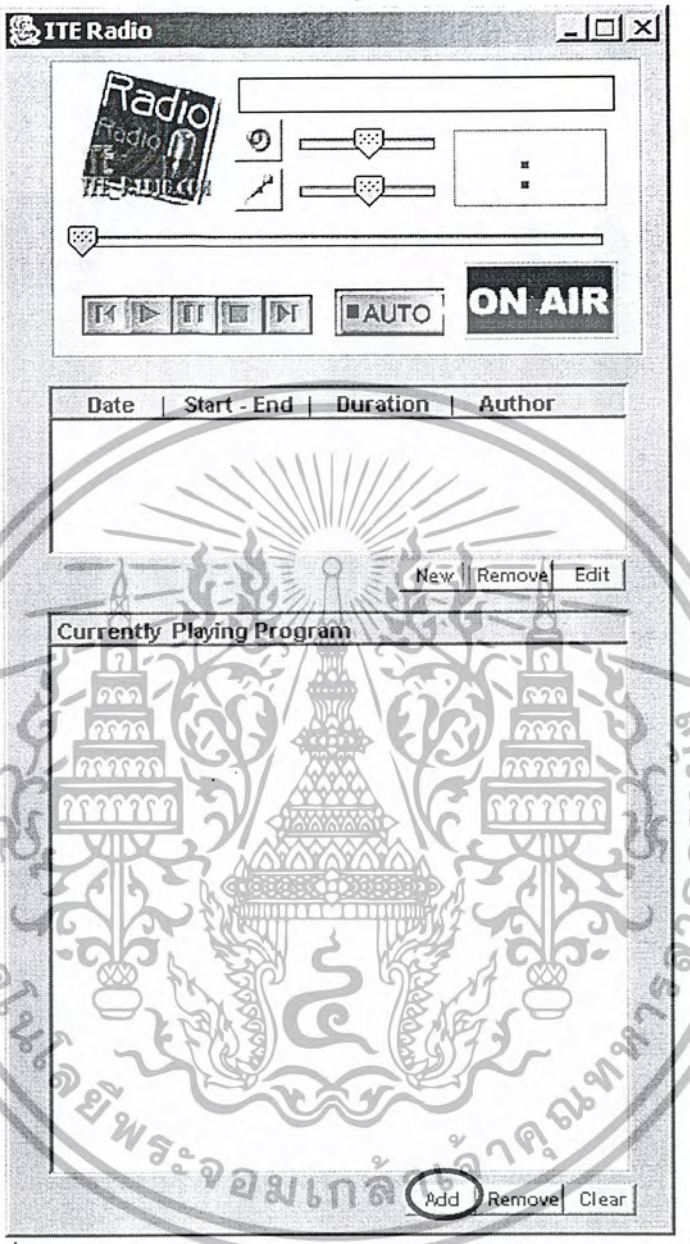
ตารางที่ 4.1 สถานะเริ่มต้นของแอพพลิเคชัน

4.2 ทดลองสร้างผังรายการปัจจุบันด้วยโหมดการทำงานแบบควบคุมด้วยมือ

4.2.1 วิธีทดลอง

กดปุ่มบราวซ์ (Browse Button) ที่อยู่ใต้ผังรายการปัจจุบันในหน้าต่าง จากนั้นทำการเลือกมีเดียซึ่งอาจเป็นไฟล์เสียงเพลงหรือไฟล์เสียงโฆษณาก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

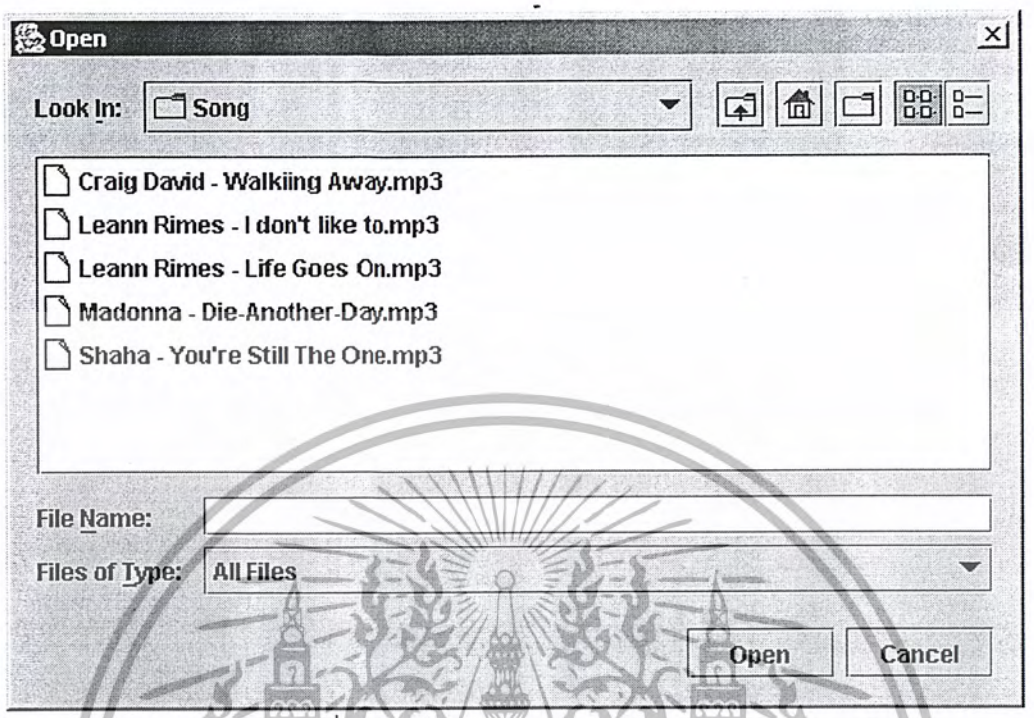


รูปที่ 4.2 วิธีทดลองเลือกแหล่งข้อมูลเสียงโดยใช้ตัวควบคุมด้วยมือ

4.2.2 ผลการทดลอง

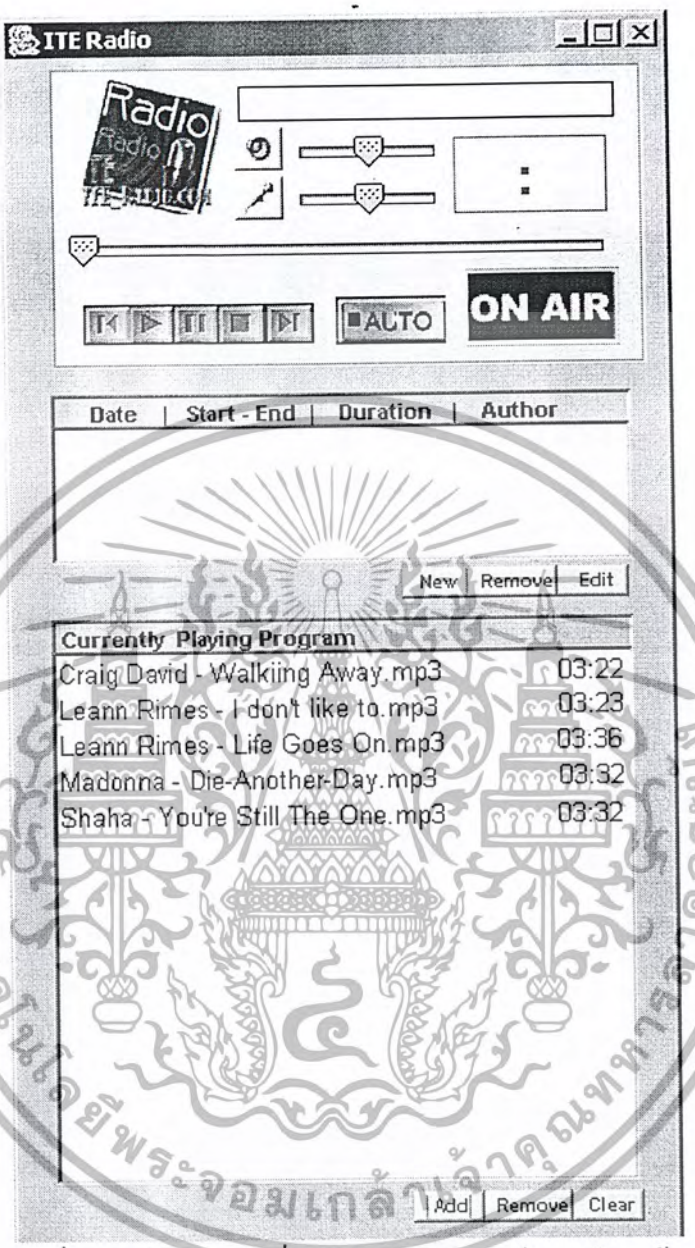
ปรากฏหน้าต่างให้เลือกแหล่งข้อมูลเสียงดังรูปที่ 4.3 และเมื่อเลือกไฟล์เสียงที่ต้องการแล้ว รายการไฟล์เสียงที่ได้เลือกไว้จะถูกแสดงในผังรายการปัจจุบันดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 หน้าต่างเลือกแหล่งข้อมูลเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



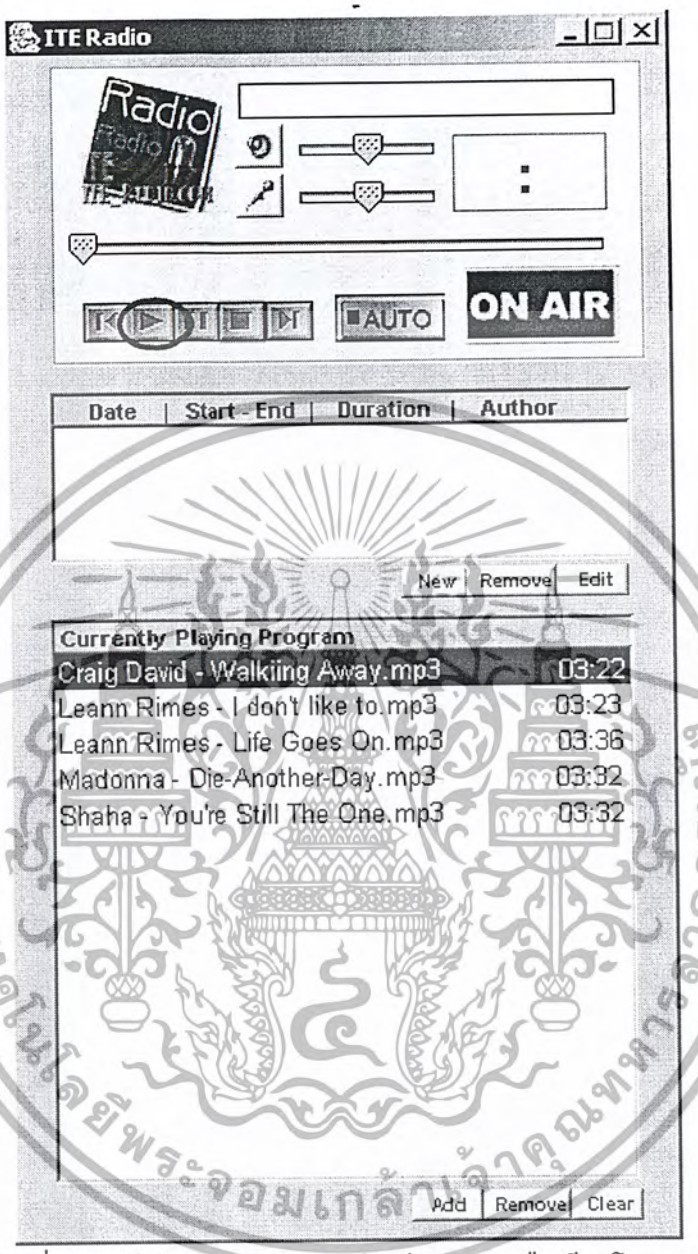
รูปที่ 4.4 หน้าต่างหลักเมื่อทำการเลือกแหล่งข้อมูลเสียงแล้ว

4.3 ทดลองเล่นมีเดีย

4.3.1 วิธีทดลอง

กดปุ่มเพลย์ (Play Button) ที่หน้าต่างควบคุมหลัก หรือดับเบิ้ลคลิกที่มีเดียในรายการที่ต้องการเล่นในผังรายการปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

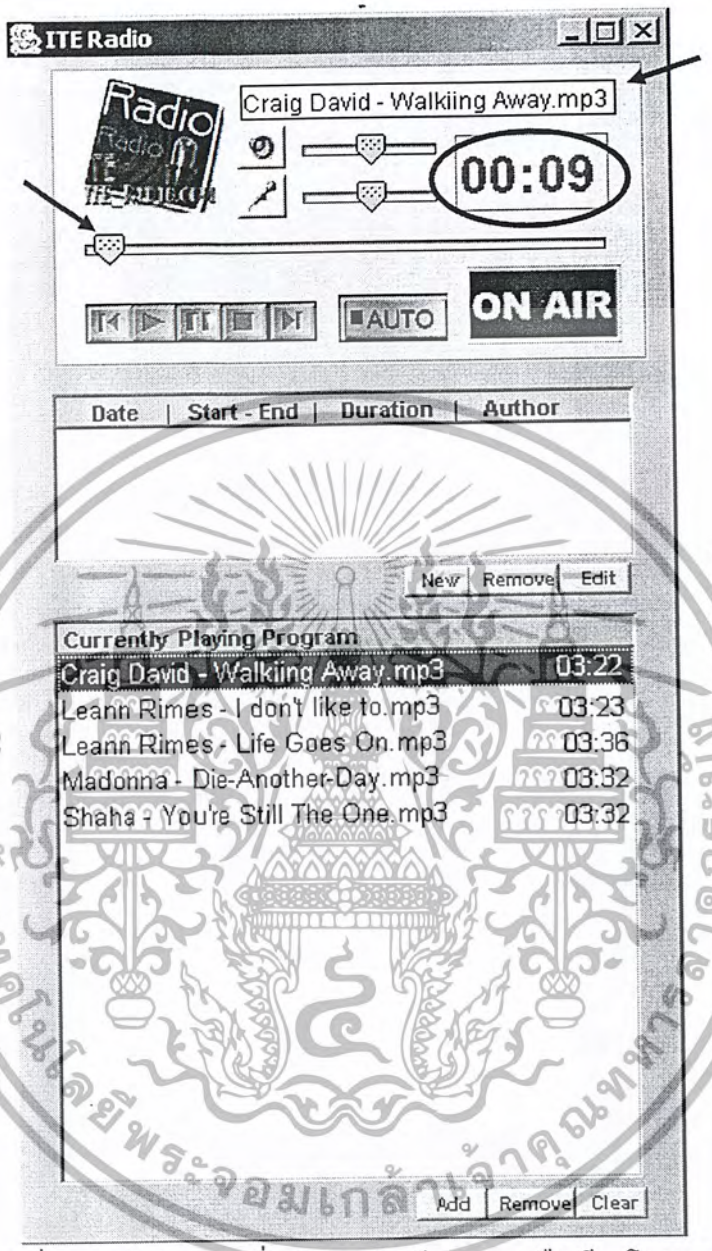


รูปที่ 4.5 หน้าต่างหลักก่อนการเล่นเสียงเพลงหรือเสียงโฆษณา

4.3.2 ผลการทดลอง

หน้าต่างหลัก จะปรากฏชื่อของเสียงเพลงหรือเสียงโฆษณาที่แถบชื่อเพลง สไลด์บาร์จะเลื่อนไปข้างหน้าตามตำแหน่งของเสียงเพลงหรือเสียงโฆษณา และส่วนของแสดงเวลาจะแสดงเวลาที่เสียงเพลงหรือเสียงโฆษณาเล่นไปแล้ว หน่วยเป็นนาที ดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



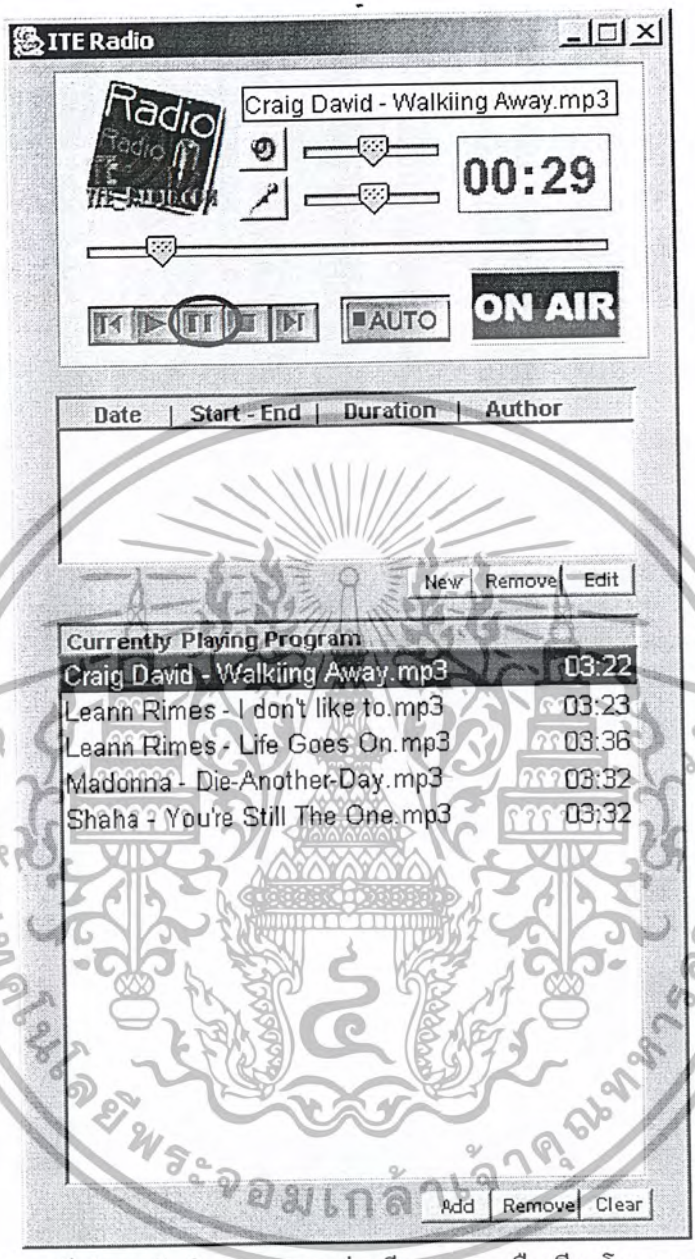
รูปที่ 4.6 หน้าต่างหลักเมื่อทำการเล่นเสียงเพลงหรือเสียงโฆษณา

4.4 ทดลองหยุดเล่นมีเดียชั่วคราว

4.4.1 วิธีทดลอง

กดปุ่มพอส (Pause Button) ที่หน้าต่างควบคุมหลักดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

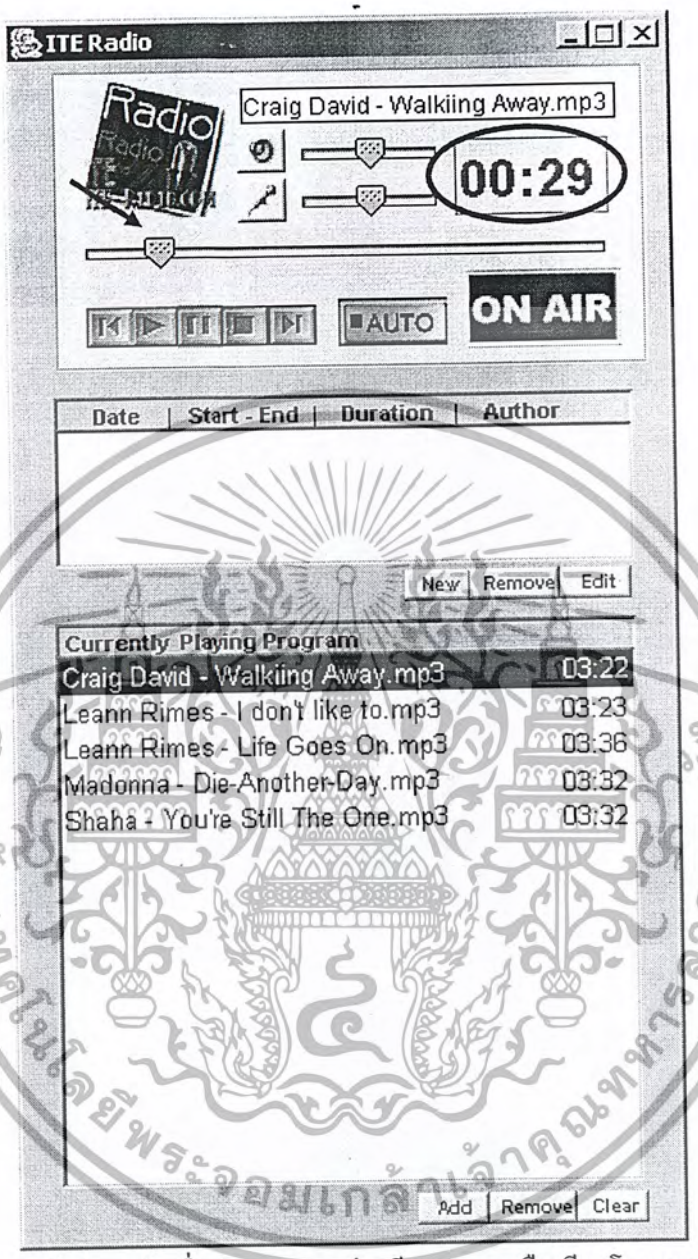


รูปที่ 4.7 หน้าต่างหลักก่อนการหยุดเล่นเสียงเพลง หรือเสียงโฆษณาชั่วคราว

4.4.2 ผลการทดลอง

จากรูปที่ 4.8 ส่วนของแสดงเวลาจะหยุดเวลาไว้ ณ ตำแหน่งที่ทำการหยุดเล่นเสียงเพลงหรือเสียงโฆษณาชั่วคราว เช่นเดียวกับสไลด์บาร์จะหยุดเลื่อนไปข้างหน้าที่ตำแหน่งที่ทำการหยุดเล่นเสียงเพลงหรือเสียงโฆษณาชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



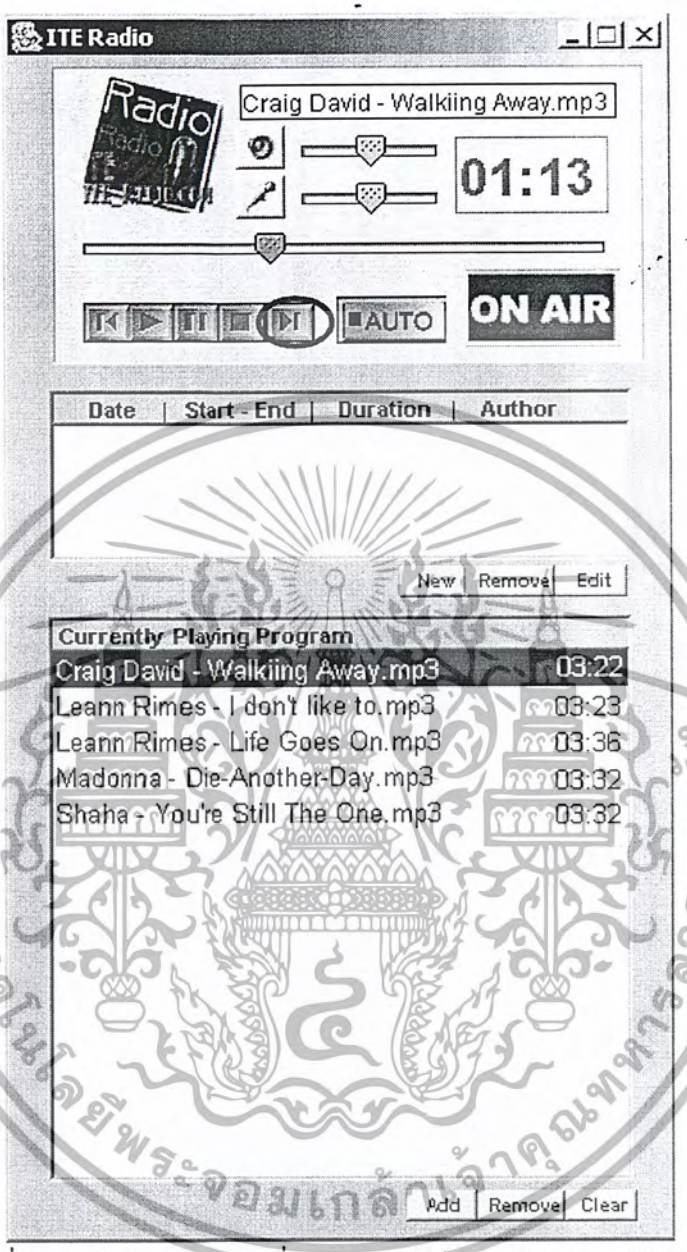
รูปที่ 4.8 หน้าต่างหลักเมื่อทำการหยุดเล่นเสียงเพลง หรือเสียงโฆษณาชั่วคราว

4.5 ทดลองเล่นมีเดียถัดไปในผังรายการปัจจุบัน

4.5.1 วิธีทดลอง

กดปุ่มเนกซ์ (Next Button) ที่ส่วนคอนโทรลของหน้าต่างหลัก ดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

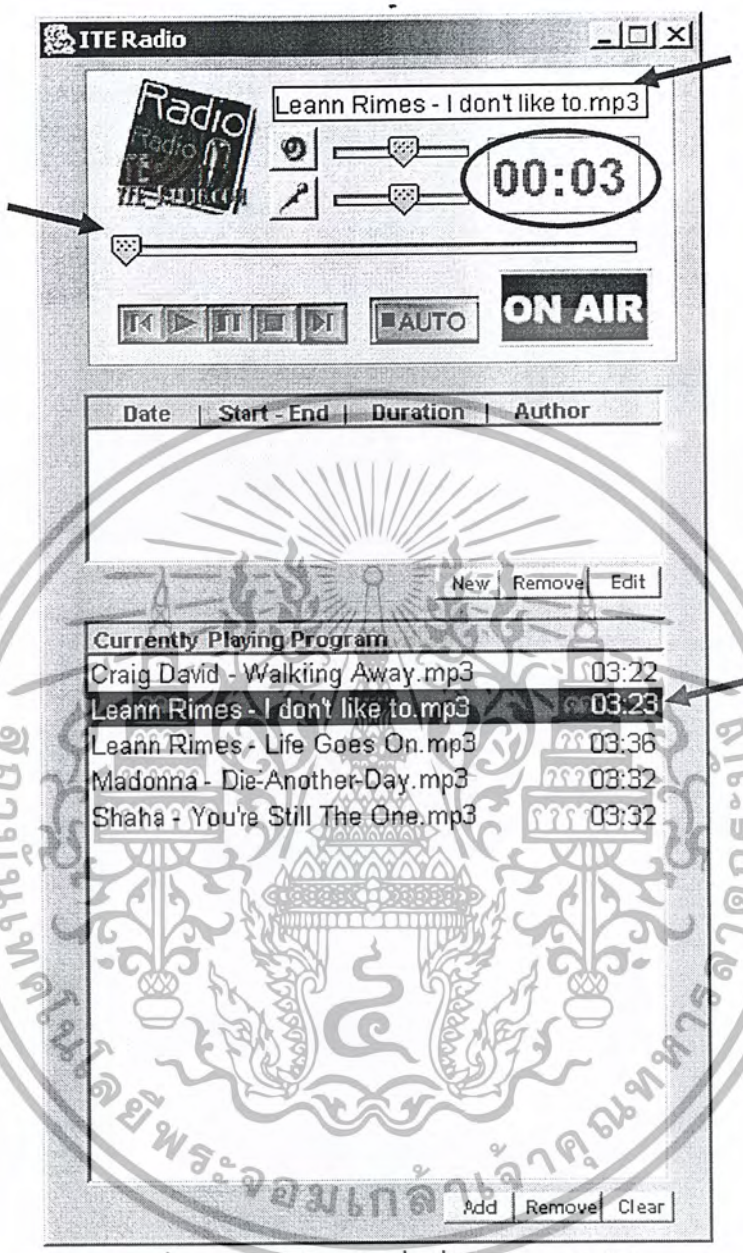


รูปที่ 4.9 หน้าต่างหลักก่อนเริ่มเล่นมีเดียถัดไปในผังรายการปัจจุบัน

4.5.2 ผลการทดลอง

จากรูปที่ 4.10 โปรแกรมจะเริ่มเล่นเพลงถัดไปในผังรายการปัจจุบันหากรายการปัจจุบันเป็นรายการสุดท้ายในผังรายการ จะเริ่มไปเล่นมีเดียในรายการแรกในผังรายการปัจจุบันแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 หน้าต่างหลักเมื่อเริ่มเล่นมีเดียถัดไป

4.6 ทดลองเล่นมีเดียก่อนหน้าในผังรายการปัจจุบัน

4.6.1 วิธีทดลอง

กดปุ่มพรีเวียส (Previous Button) ที่หน้าต่างหลัก ดังรูปที่ 4.11

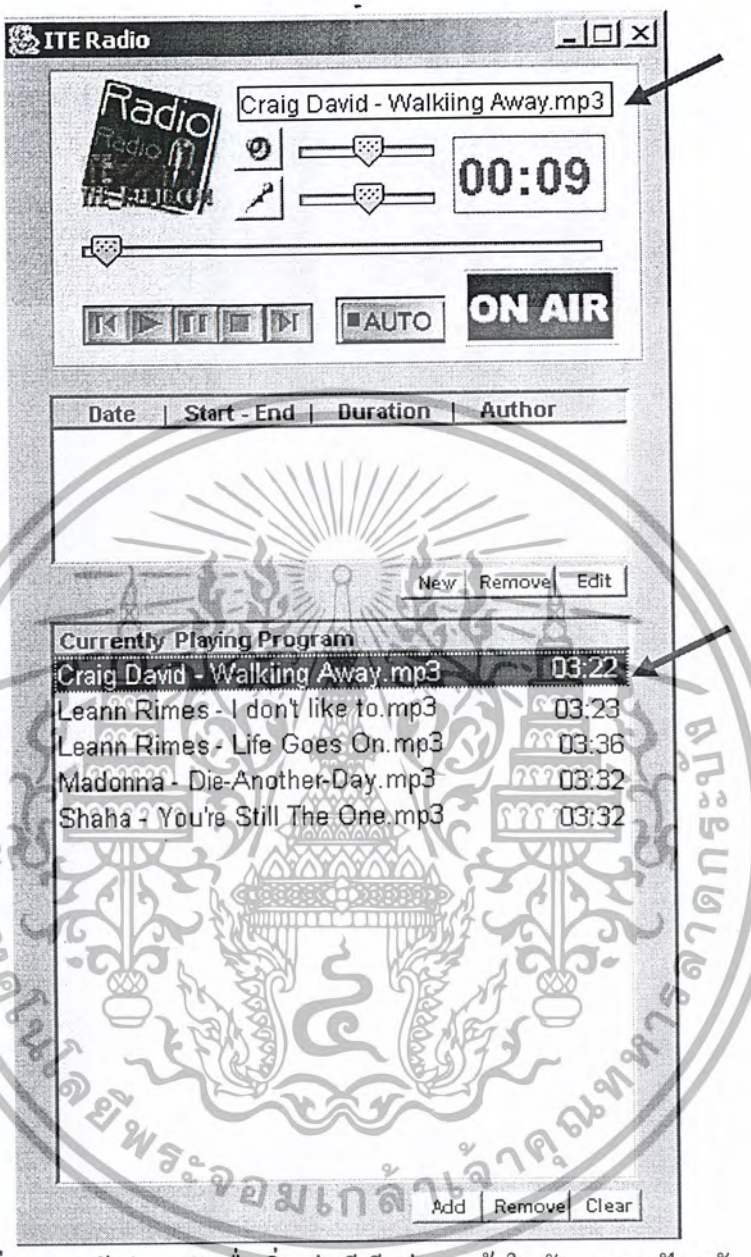
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.11 หน้าต่างหลักก่อนเริ่มเล่นมีเดียก่อนหน้าในผังรายการปัจจุบัน

4.6.2 ผลการทดลอง

จากรูปที่ 4.12 โปรแกรมจะเริ่มเล่นมีเดียรายการก่อนหน้าของรายการปัจจุบันในผังรายการปัจจุบัน หากรายการปัจจุบันเป็นรายการแรกในผังรายการ โปรแกรมจะกลับไปเริ่มเล่นมีเดียในรายการสุดท้ายในผังรายการปัจจุบันแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



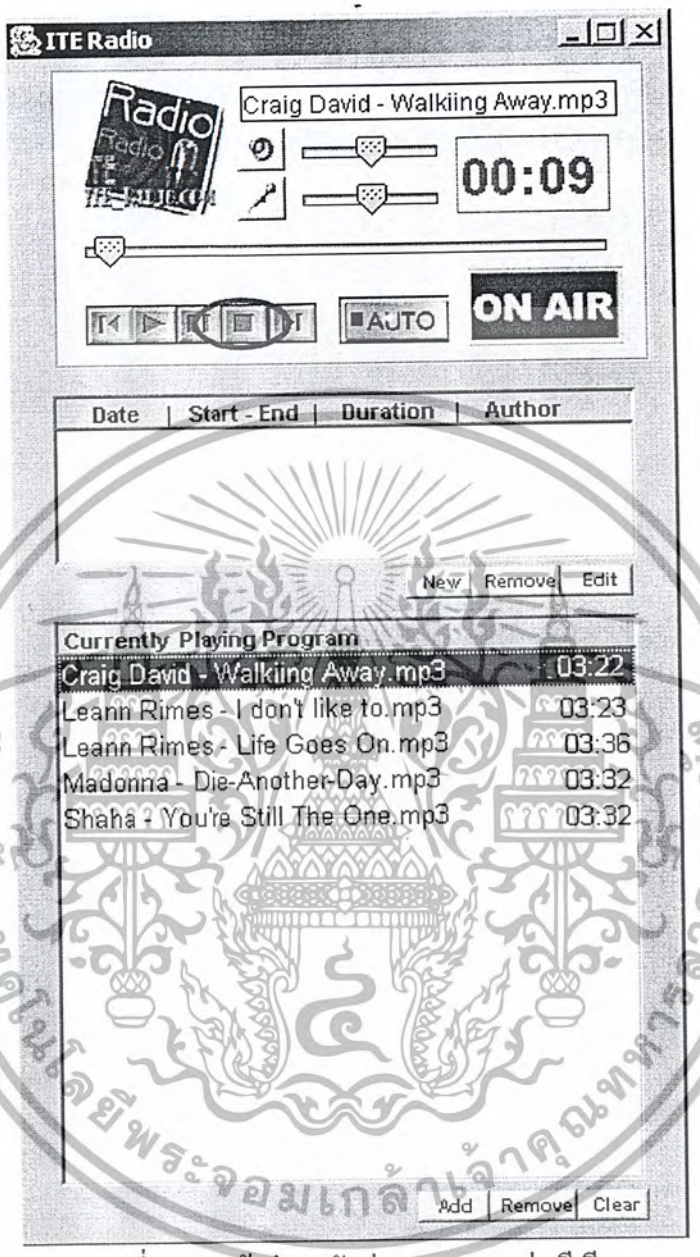
รูปที่ 4.12 หน้าต่างหลักเมื่อเริ่มเล่นมีเดียก่อนหน้าในผังรายการปัจจุบัน

4.7 ทดลองหยุดเล่นมีเดีย

4.7.1 วิธีทดลอง

กดปุ่มสต็อป (Stop Button) ที่หน้าต่างหลัก ดังรูปที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

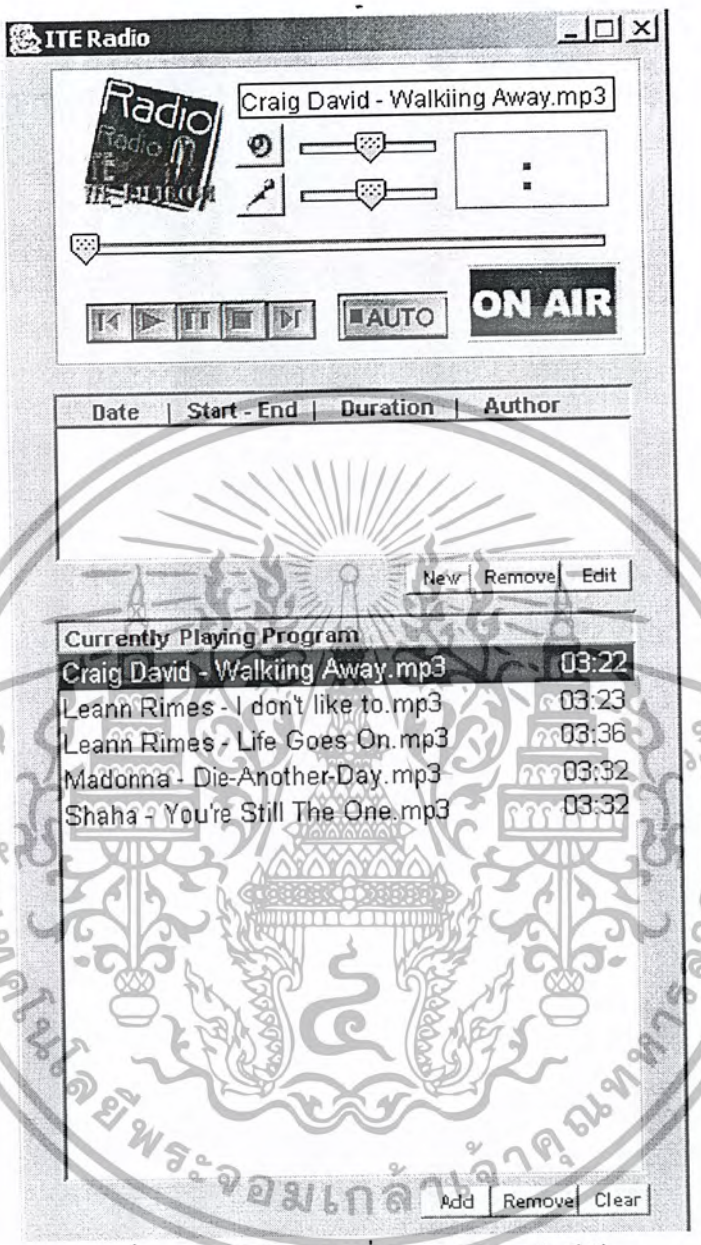


รูปที่ 4.13 หน้าต่างหลักก่อนการหยุดเล่นมีเดีย

4.7.2 ผลการทดลอง

จากรูปที่ 4.14 ส่วนแสดงเวลาของมีเดีย ค่าเวลาจะกลับเป็นค่าเริ่มต้น และสไลด์บาร์จะเลื่อนไปย้อนกลับไปยังจุดเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 หน้าต่างหลักเมื่อทำการหยุดเล่นมีเดีย

4.8 ทดลองลบมีเดียที่เลือกไว้ในผังรายการปัจจุบัน

4.8.1 วิธีทดลอง

เลือกรายการเสียงเพลง หรือเสียงโฆษณาที่ต้องการลบจากลิสท์แสดงรายการมีเดียที่เลือกไว้จากหน้าต่างหลัก กดปุ่มรีมูฟ (Remove Button) ดังรูปที่ 4.15

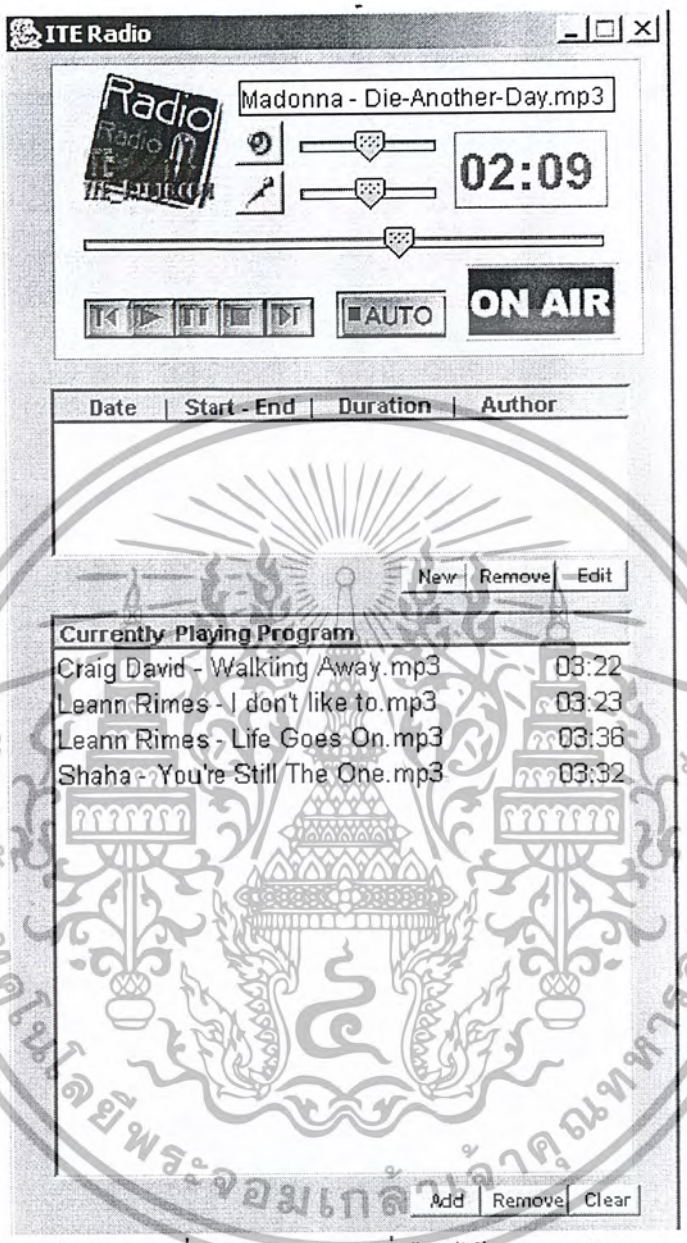
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.15 หน้าต่างหลักก่อนลบมีเดียที่เลือกไว้ในผังรายการปัจจุบัน

4.8.2 ผลการทดลอง

รายการเสียงเพลงหรือเสียงโฆษณาที่เลือกไว้ จะถูกลบออกจากลิสท์แสดงรายการมีเดียที่เลือกไว้ ดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 หน้าต่างหลักเมื่อทำการลบมีเดียที่เลือกไว้ออกจากผังรายการปัจจุบัน

4.9 ทดลองล้างผังรายการปัจจุบัน

4.9.1 วิธีทดลอง

กดปุ่มเคลียร์ผังรายการปัจจุบัน (Clear Button) จากหน้าต่างหลักดังรูปที่ 4.17

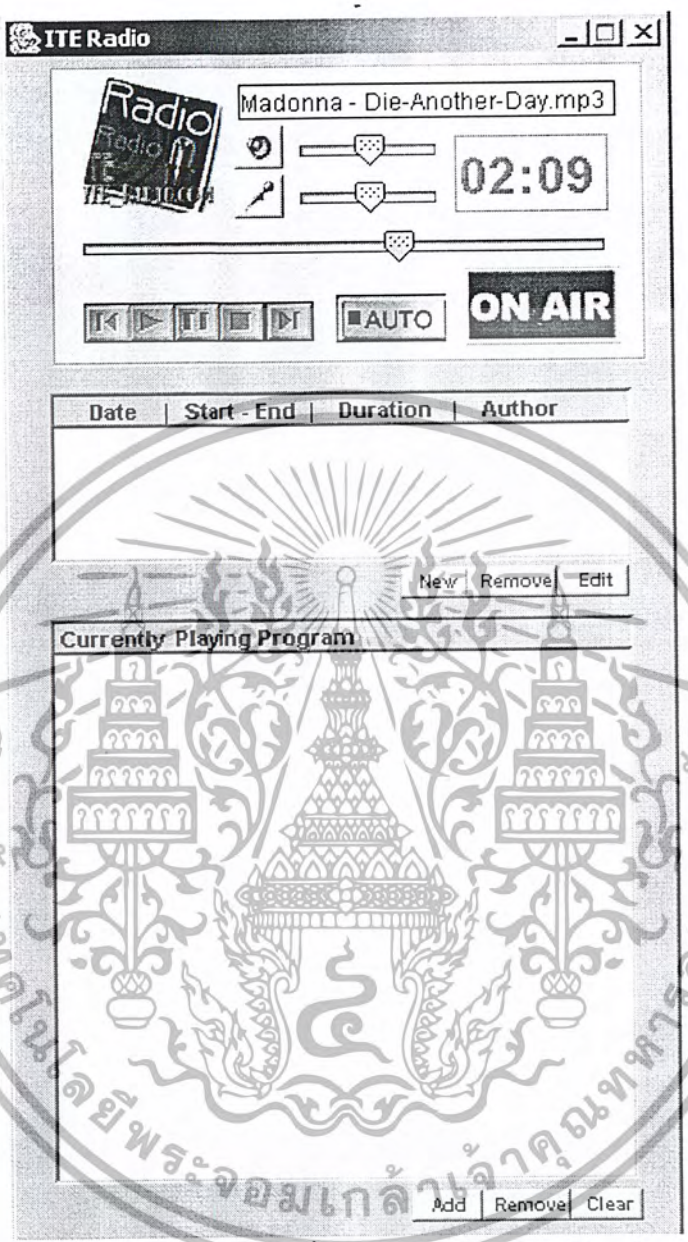
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.17 หน้าต่างหลักก่อนทำการล้างผังรายการปัจจุบัน

4.9.2 ผลการทดลอง

รายการเสียงเพลง หรือเสียงโฆษณาทั้งหมดในลิสท์แสดงรายการมีเดียที่เลือกไว้จะถูกลบออกทั้งหมด ดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 หน้าต่างหลักเมื่อทำการล้างผังรายการปัจจุบัน

4.10 ทดลองสร้างผังรายการล่วงหน้า

4.10.1 วิธีทดลอง

กดปุ่มนิวลิสท์ (New List Button) ในส่วนคอนโทรลของหน้าต่างหลัก ดังรูปที่ 4.19

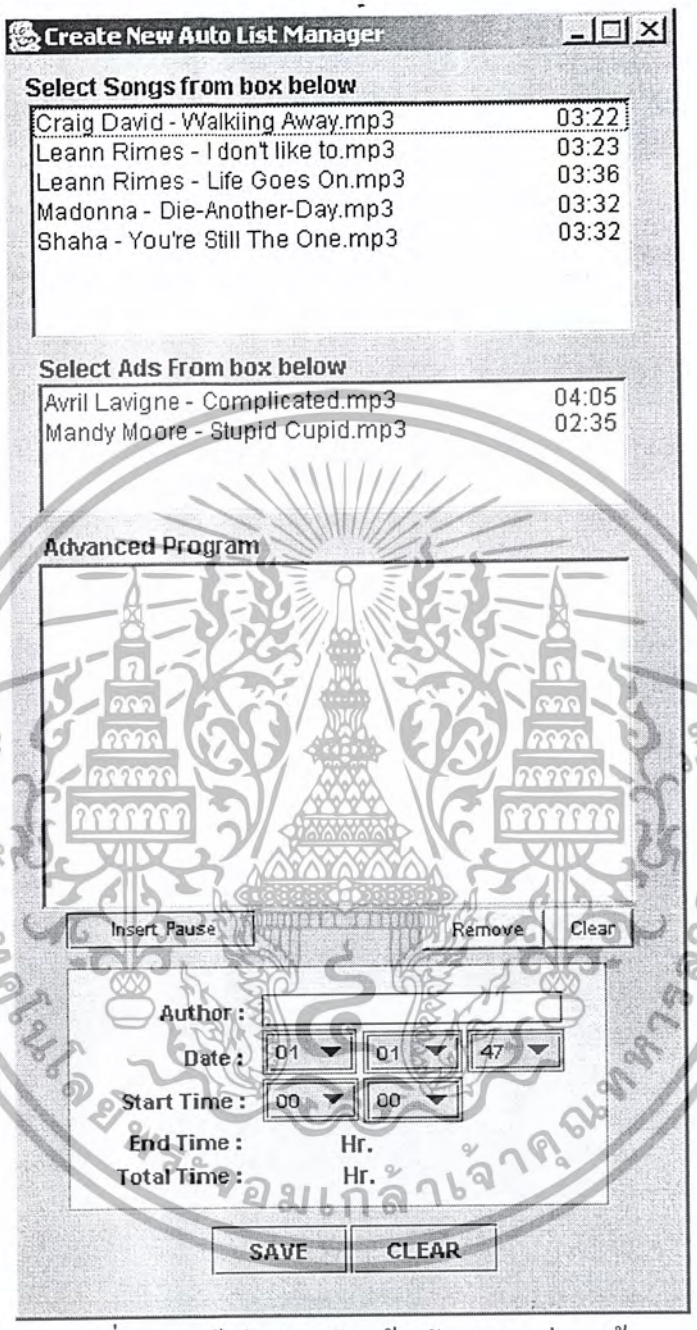
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.19 หน้าต่างหลักก่อนสร้างผังรายการล่วงหน้า

4.10.2 ผลการทดลอง

เมื่อกดปุ่มนิวลิสท์เพื่อสร้างผังรายการล่วงหน้า จะปรากฏหน้าต่างสำหรับสร้างผังรายการล่วงหน้า (Create New Auto List Manager) ขึ้น มีลักษณะดังรูปที่ 4.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



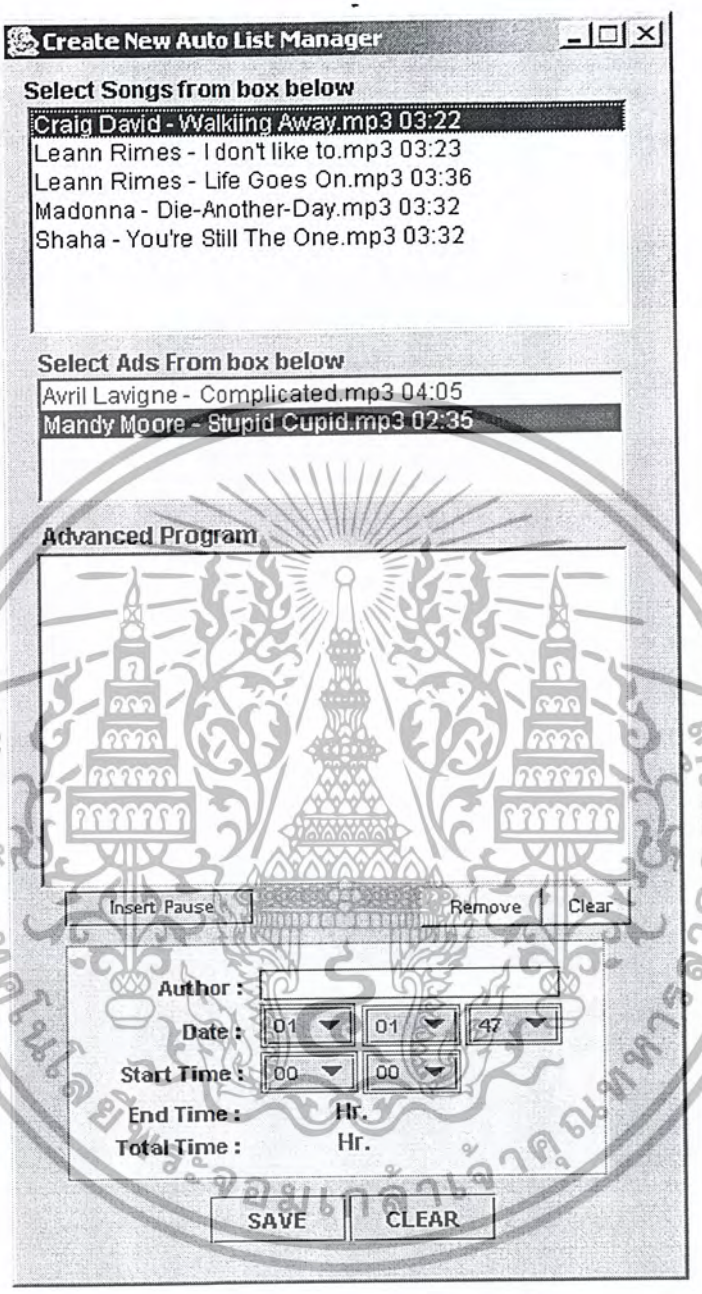
รูปที่ 4.20 หน้าต่างสำหรับสร้างผังรายการล่วงหน้า

4.11 ทดลองเพิ่มเสียงเพลงหรือเสียงโฆษณาลงในผังรายการล่วงหน้า

4.11.1 วิธีทดลอง

เลือกมีเดียจากลิสท์แสดงรายการแหล่งข้อมูลเสียง โดยการดับเบิ้ลคลิ๊กที่รายการมีเดียเสียงเพลง หรือดับเบิ้ลคลิ๊กที่รายการมีเดียเสียงโฆษณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

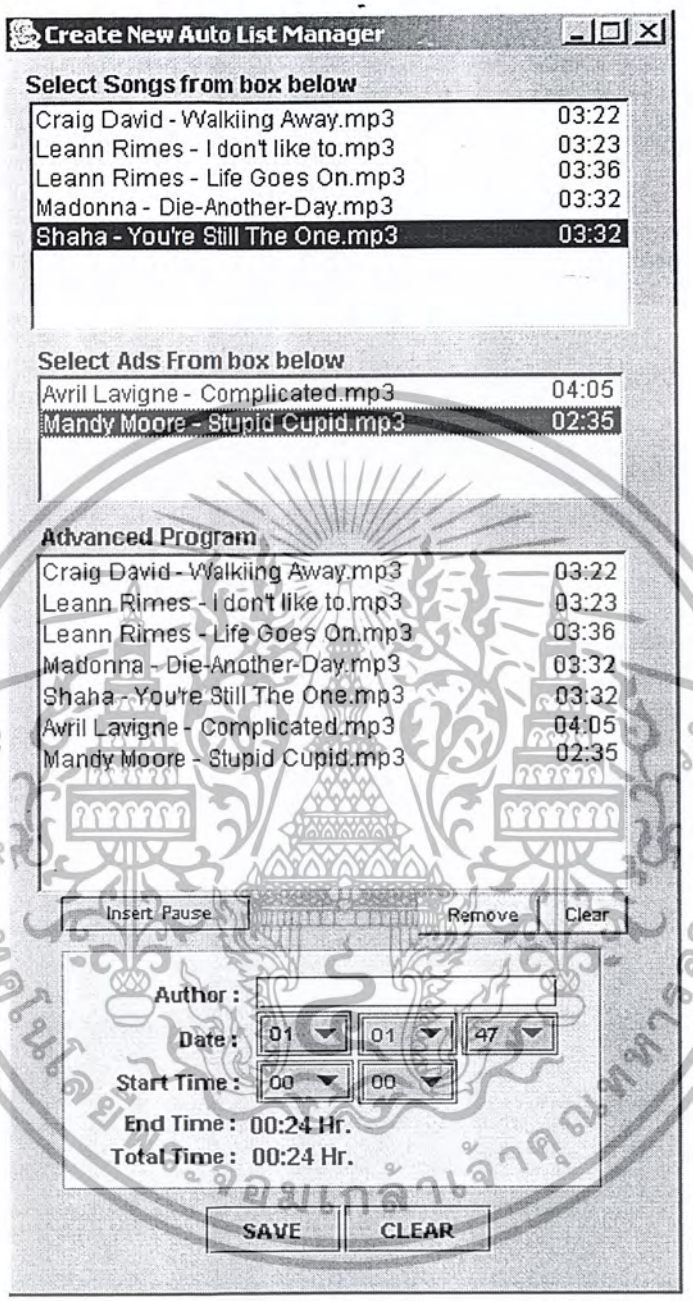


รูปที่ 4.21 วิธีเลือกแหล่งเสียงสำหรับสร้างผังรายการล่วงหน้า

4.11.2 ผลการทดลอง

หน้าต่างควบคุมแบบอัตโนมัติ ดังรูปที่ 4.22 รายการมีเดียที่ถูกเลือกจะเพิ่มเข้ามาในลิสท์รายการมีเดียที่เลือกไว้ ในส่วนแสดงเวลาสิ้นสุด (End Time) และในส่วนแสดงเวลาทั้งหมดที่ต้องใช้สำหรับผังรายการล่วงหน้านี้ (Total Time) จะมีค่าเพิ่มขึ้น ตามเวลามีเดีย (Media Time) ของรายการมีเดียที่เลือกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าหลังจากทำการเลือกมีเดียที่ต้องการ

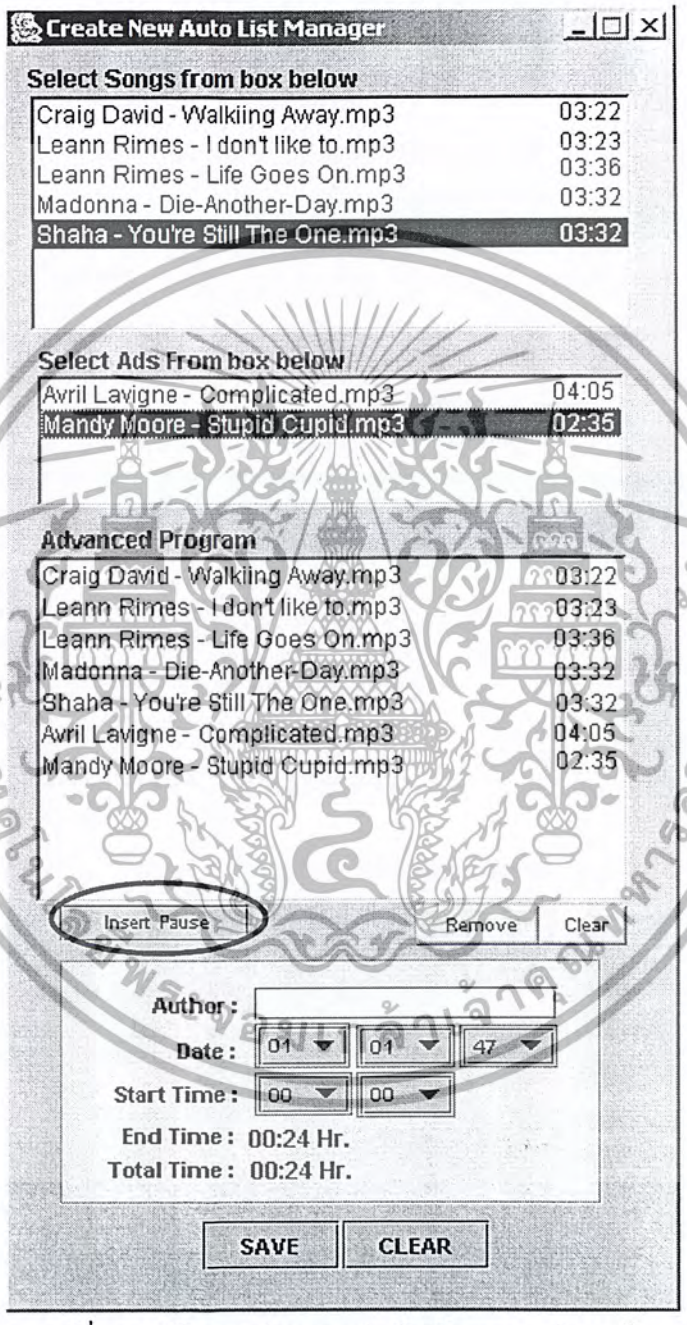
4.12 ทดลองแทรกช่วงหยุดเล่นมีเดียชั่วคราวในผังรายการล่วงหน้า

4.12.1 วิธีทดลอง

1. กดปุ่มแทรกการหยุดเล่นมีเดียชั่วคราว (Insert Pause) ในหน้าต่างควบคุมแบบอัตโนมัติจะปรากฏไดอะล้อคบ๊อกซ์อินพุต (Input Dialog Box) ดังรูปที่ 4.24

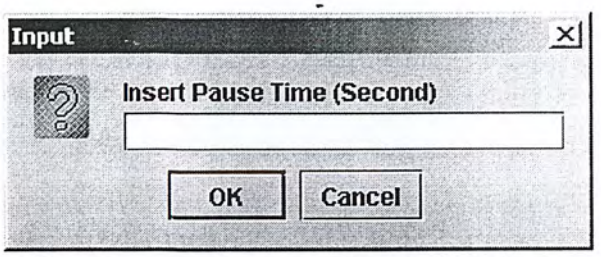
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ป้อนช่วงเวลาที่ต้องการแทรกการหยุดเล่นมีเดียชั่วคราว หน่วยเป็นวินาที ลงในไดอะล้อคบ๊อกซ์อินพุต แล้วกดปุ่มตกลง (OK Button)

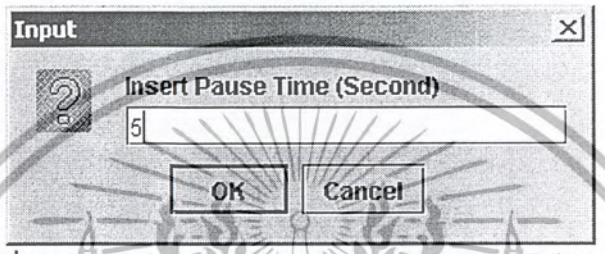


รูปที่ 4.23 วิธีแทรกช่วงเวลาว่างในผังรายการล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 ไดอะล้อกซ์อินพุต (Input Dialog Box) สำหรับรับค่าช่วงเวลาว่างหน่วยเป็นวินาที

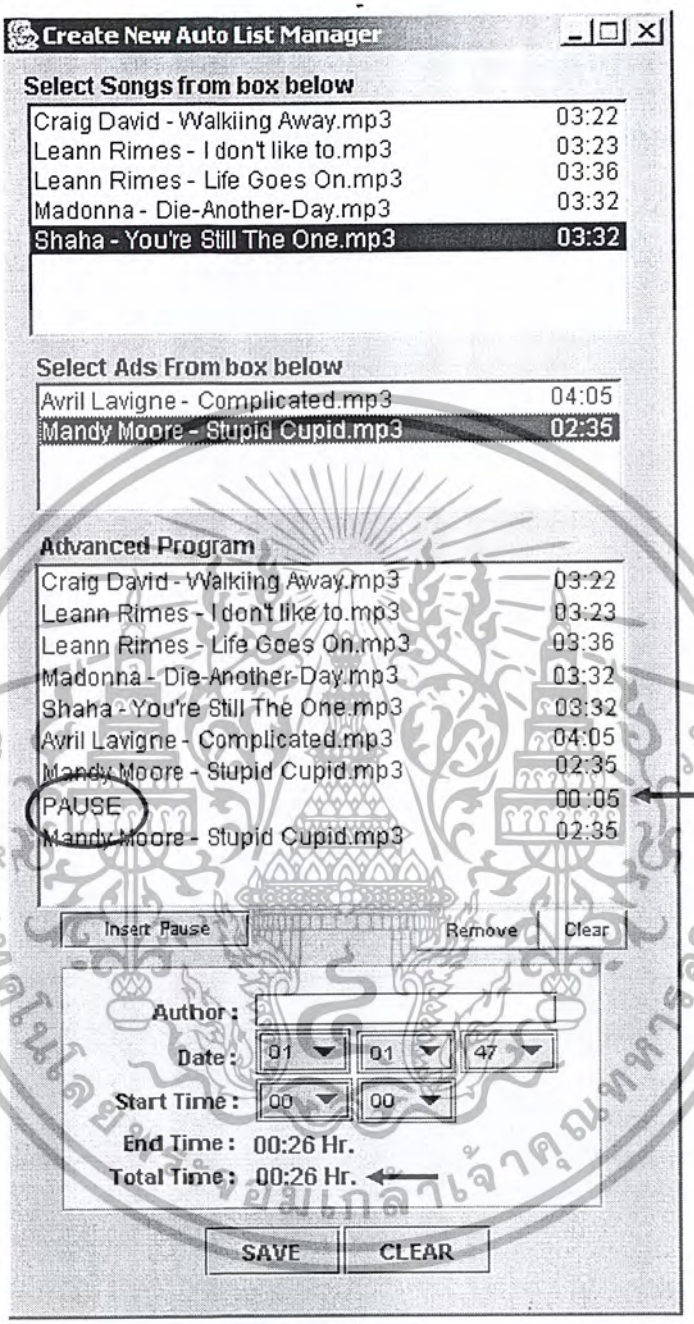


รูปที่ 4.25 ทดลองป้อนค่าช่วงเวลาว่างในไดอะล้อกซ์อินพุต

4.12.2 ผลการทดลอง

เมื่อกดปุ่มตกลง หลังจากใส่เวลาที่ต้องการแทรกการหยุดเล่นมีเดียชั่วคราวลงในไดอะล้อกบ๊อกซ์อินพุต หน้าต่างควบคุมแบบอัตโนมัติจะแทรกเวลาสำหรับการหยุดเล่นมีเดียชั่วคราวลงในลิสท์แสดงรายการมีเดียที่เลือกไว้ ถ้ากดปุ่มยกเลิกของไดอะล้อกบ๊อกซ์อินพุต ที่หน้าต่างควบคุมแบบอัตโนมัติจะไม่เกิดการเปลี่ยนแปลงใด ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



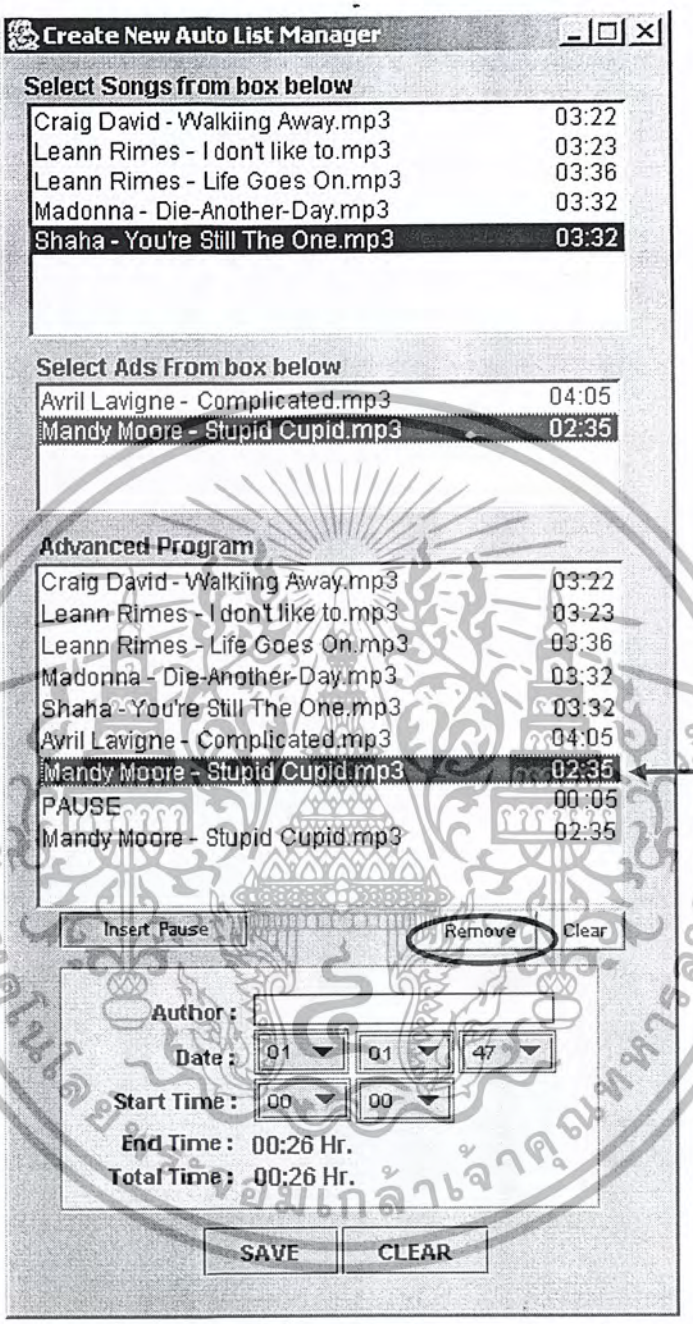
รูปที่ 4.26 หน้าต่างควบคุมแบบอัตโนมัติ เมื่อทำการแทรกการหยุดเล่นมีเดียชั่วคราว

4.13 ทดลองลบมีเดียออกจากผังรายการ

4.13.1 วิธีทดลอง

เลือกรายการมีเดียที่ต้องการลบออกจากผังรายการล่วงหน้า จากนั้นกดปุ่มรีมูฟ ดังรูปที่ 4.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

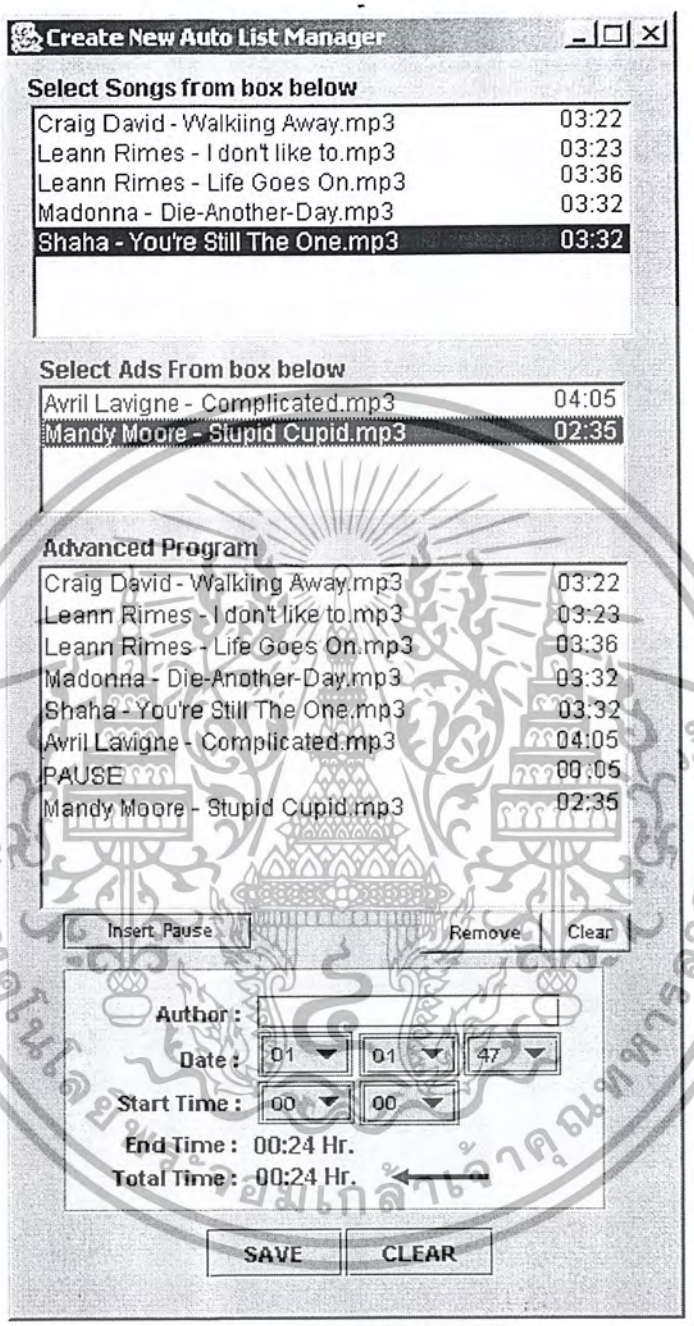


รูปที่ 4.27 วิธีลบมีเดียออกจากผังรายการล่วงหน้า

4.13.2 ผลการทดลอง

รายการมีเดียที่ถูกเลือก จะถูกลบออกจากผังรายการล่วงหน้าที่กำลังสร้างอยู่ ในส่วนแสดงเวลาสิ้นสุดของผังรายการ และส่วนแสดงเวลารวมของผังรายการจะลดลง ตามเวลาของมีเดียที่ถูกลบไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



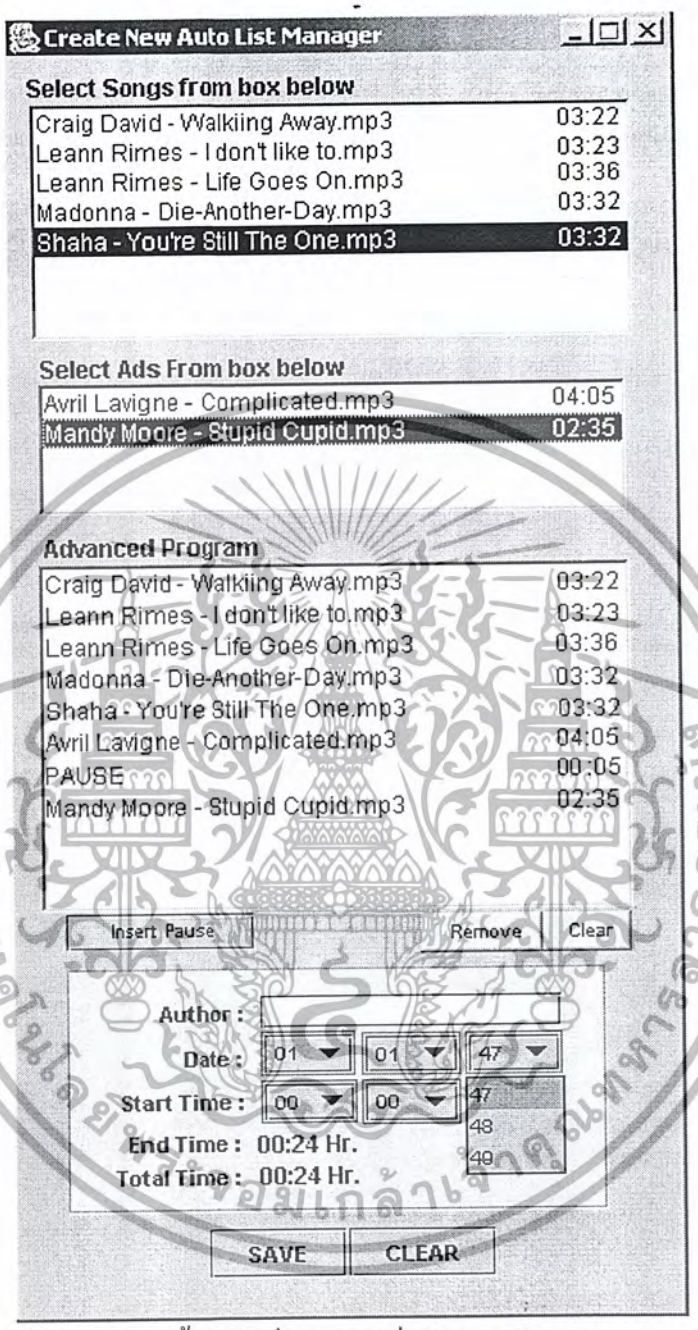
รูปที่ 4.28 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าหลังจากทำการลบมีเดียออกจากผังรายการ

4.14 ทดลองการตั้งค่าวันที่ และเวลาของผังรายการล่วงหน้า

4.14.1 วิธีทดลอง

เลือกค่าวันที่ และเวลาเริ่มต้นของผังรายการล่วงหน้า จากดร็อปดาวน์เมนู (Drop down Menu) ในส่วนกำหนดเวลาออกอากาศ ดังรูปที่ 4.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

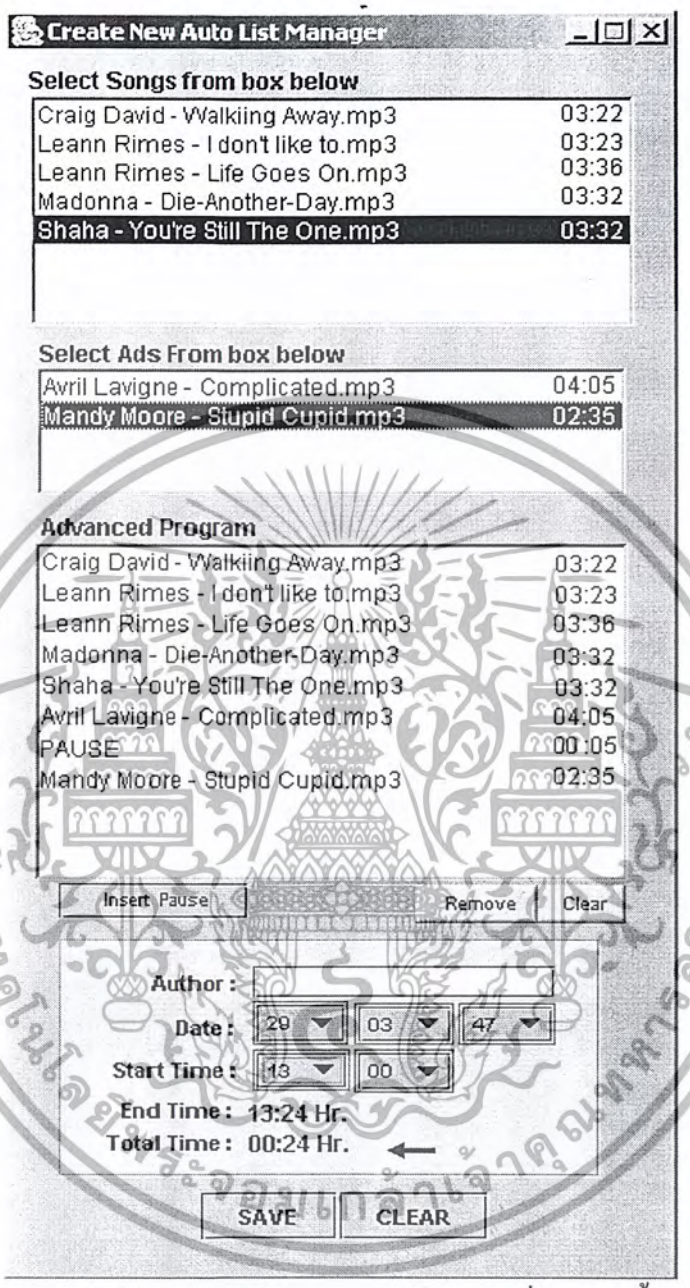


รูปที่ 4.29 วิธีการตั้งค่าวันที่และเวลาเริ่มต้นของผังรายการล่วงหน้า

4.14.2 ผลการทดลอง

ในส่วนของเวลาสิ้นสุดของผังรายการ (End Time) จะเปลี่ยนไปเป็นค่าเวลาที่ตั้งไว้รวมกับเวลาทั้งหมดที่ใช้เล่นเสียงเพลงหรือเสียงโฆษณา ที่เลือกไว้ในลิสท์แสดงรายการมีเดียที่เลือกไว้ ดังรูปที่ 4.30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



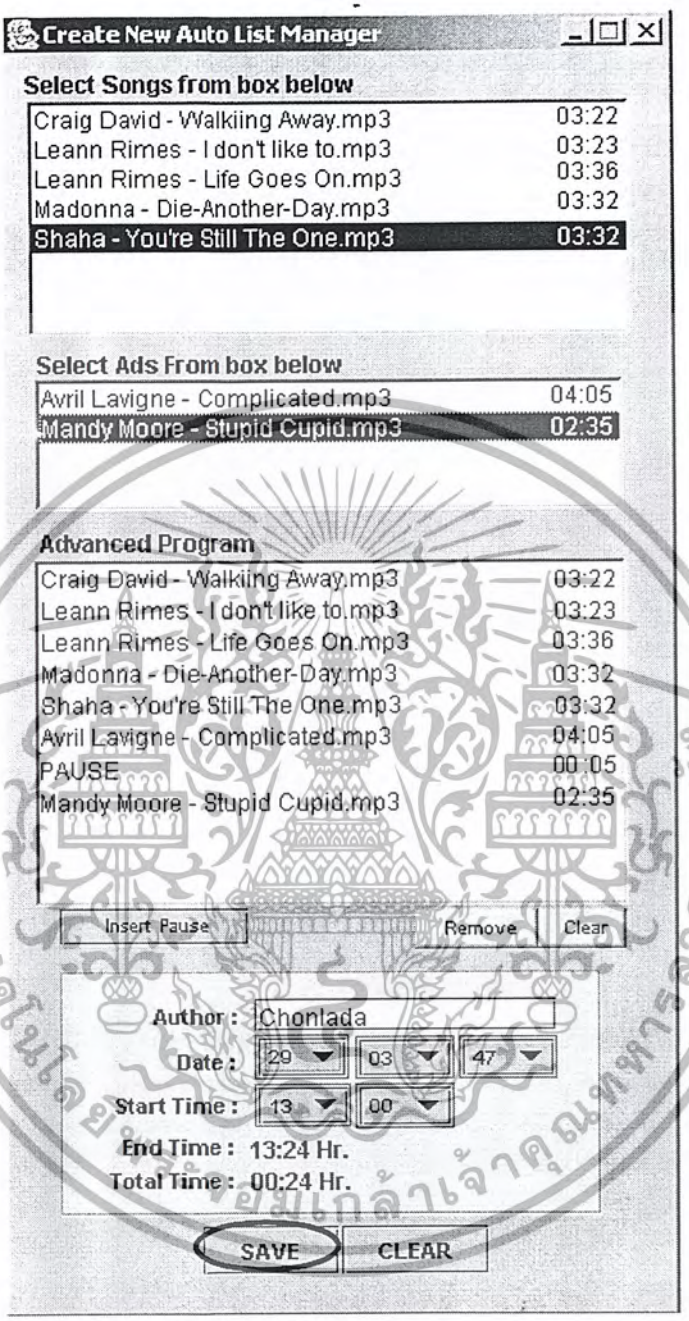
รูปที่ 4.30 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าเมื่อทำการตั้งค่าวันที่และเวลาสำหรับผังรายการล่วงหน้า

4.15 ทดลองบันทึกการสร้างผังรายการล่วงหน้า

4.15.1 วิธีทดลอง

ทดลองสร้างผังรายการล่วงหน้าโดยกำหนดค่าต่าง ๆ ให้ครบถ้วน จากนั้นกดปุ่มเซฟ (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 หรือวันที่หลังจากวันที่สร้างผังรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การตั้งค่าวันที่และเวลาผิดพลาด คือ วันที่เป็นวันก่อนวันจริงขณะสร้างผังรายการ หรือ วันที่เป็นวันจริงขณะสร้างผังรายการแต่เวลาเป็นเวลาก่อนเวลาจริงขณะสร้างผังรายการ

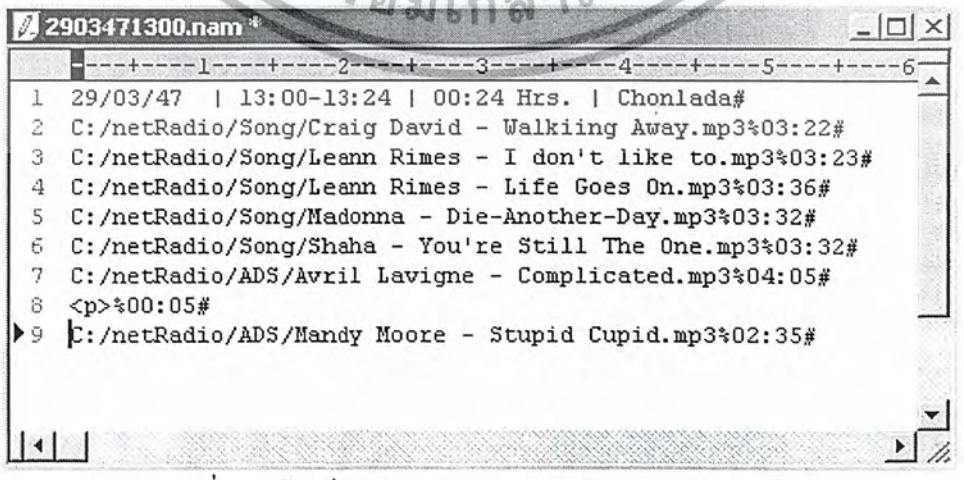
4.15.2.1 ผลการทดลอง กรณีแรก ไม่มีรายการของมีเดียในผังรายการที่จะบันทึก

ระบบจะแสดงไดอะล้อคบ๊อกซ์เตือน (ITE Radio Alert Dialog Box) ว่าในผังรายการล่วงหน้าที่สร้างไว้ ไม่มีรายการเสียงเพลง หรือเสียงโฆษณา ที่เลือกไว้ ดังรูปที่ 4.32

รูปที่ 4.32 ไดอะล้อคบ๊อกซ์เตือนกรณีไม่มีรายการเสียงในผังรายการที่สร้าง

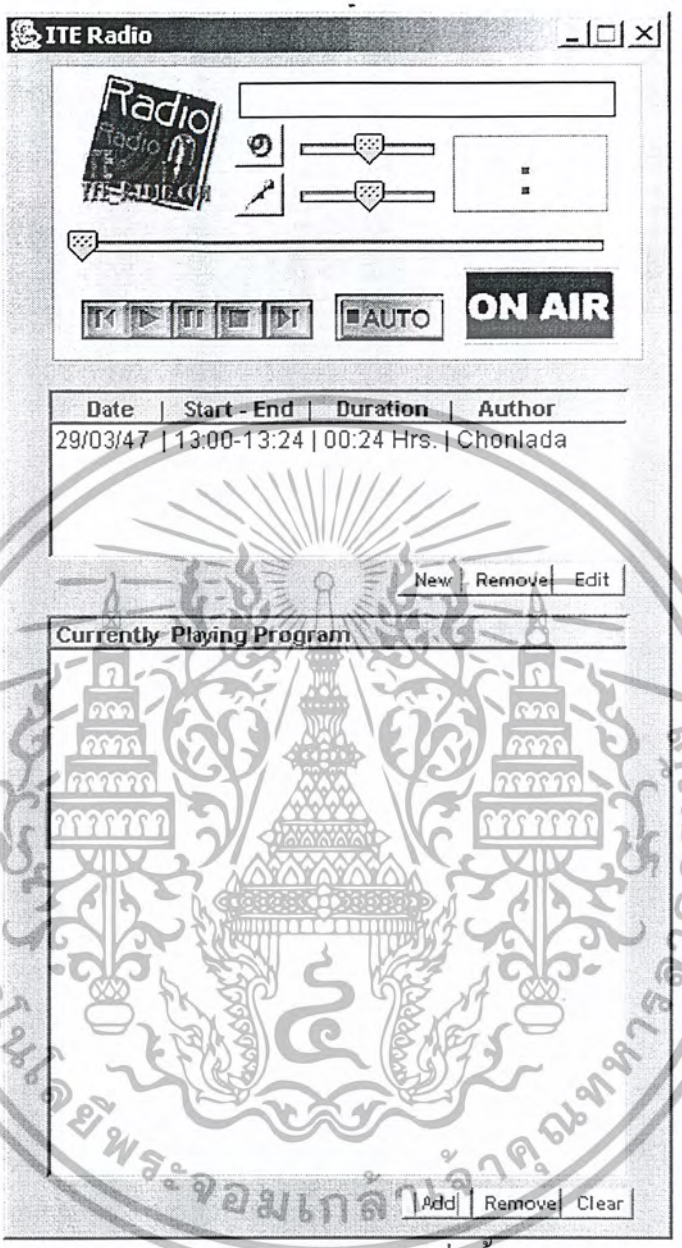
4.15.2.2 ผลการทดลอง กรณีที่สอง การตั้งค่าวันที่ และเวลาถูกต้อง

1. รายละเอียดของการสร้างผังรายการล่วงหน้าจะถูกบันทึกอยู่ในรูปแบบไฟล์ นามสกุล เอ็นเอเอ็ม (.nam) มีรายละเอียด ดังรูปที่ 4.33
2. หน้าต่างหลัก รายละเอียดของผังรายล่วงหน้าที่เพิ่งสร้างเสร็จจะถูกเพิ่มลงในลิสท์แสดงรายการผังรายการล่วงหน้า (Auto Playlist)



รูปที่ 4.33 ไฟล์แสดงรายละเอียดของผังรายการล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

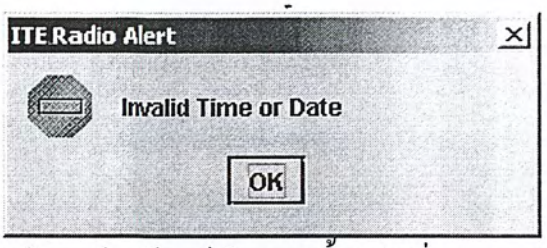


รูปที่ 4.34 ข้อมูลของผังรายการล่วงหน้าในหน้าต่างหลักเพิ่มขึ้นหลังจากทำการบันทึกผังรายการ

4.15.2.3 ผลการทดลอง กรณีที่สาม การตั้งค่าวันที่ และเวลาผิดพลาด

ระบบจะแสดงไดอะล้อคบ๊อกซ์เตือน (ITE Radio Alert Dialog Box) ว่า วันที่ หรือเวลาที่ตั้งไว้ผิดพลาด ให้ทำการตั้งค่าวันที่ และเวลาใหม่อีกครั้ง ดังรูปที่ 4.35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



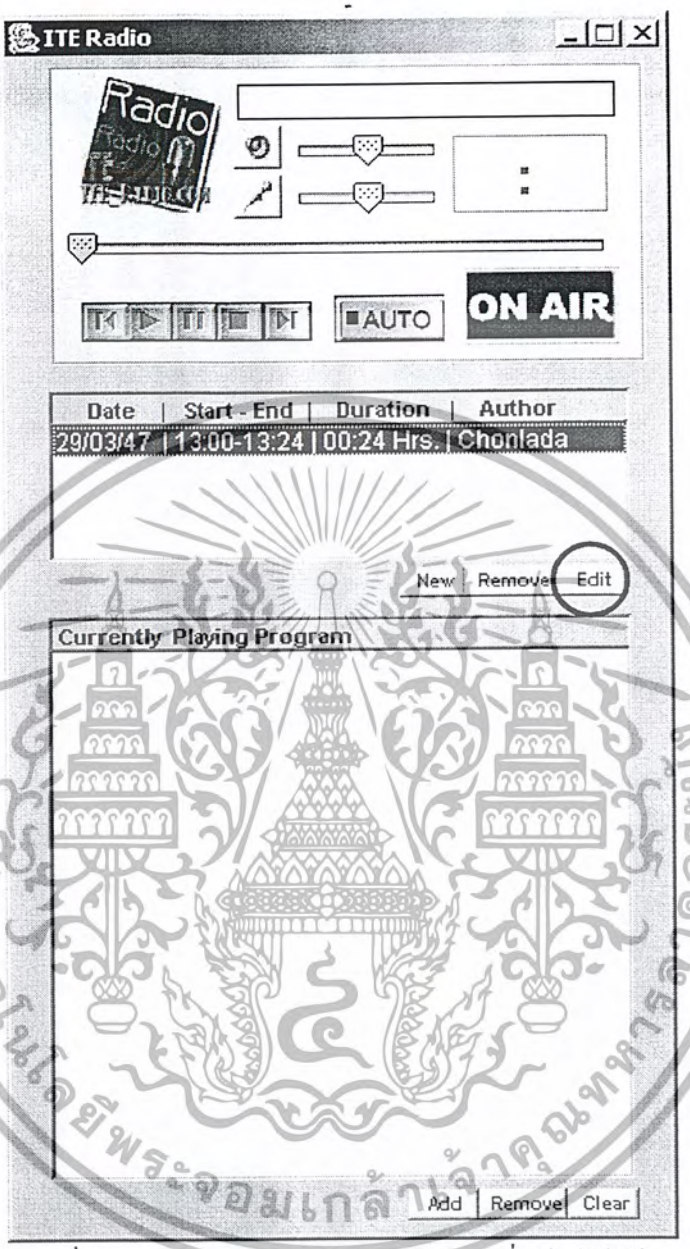
รูปที่ 4.35 ไดอะล้อคบ๊อกซ์เตือนการตั้งค่าวันที่และเวลาผิดพลาด

4.16 ทดลองแก้ไขผังรายการล่วงหน้า

4.16.1 วิธีทดลอง

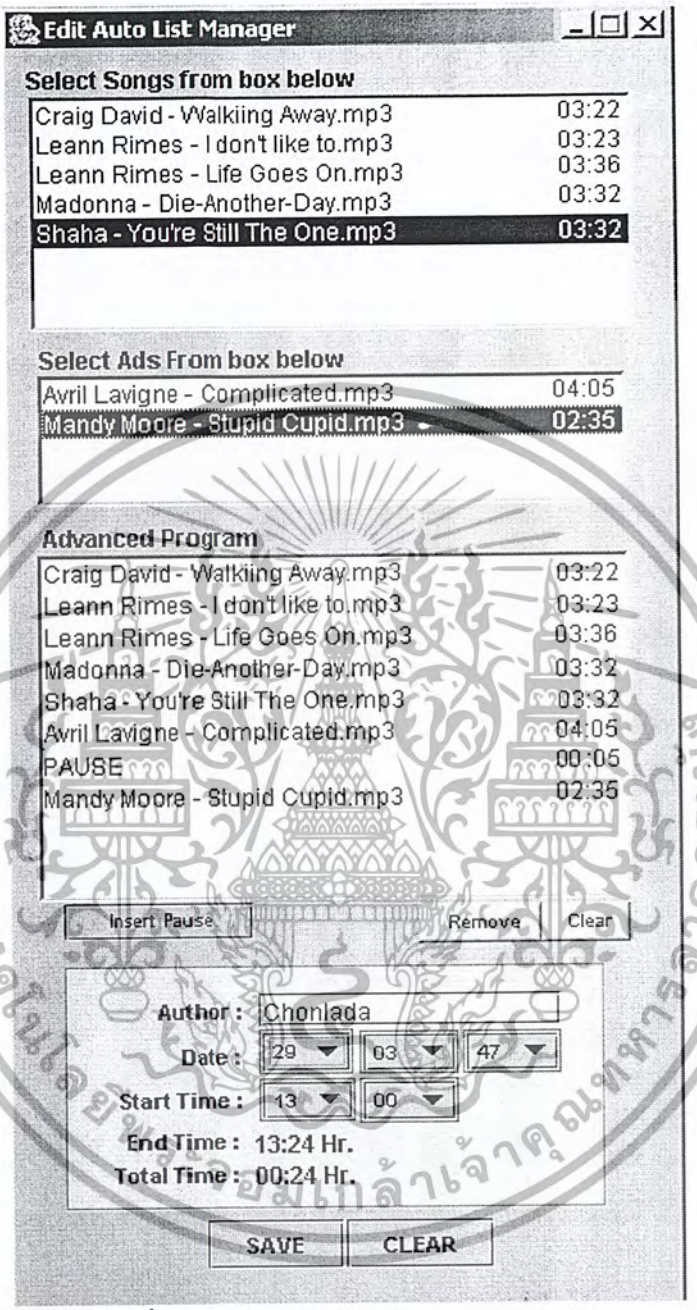
1. เลือกผังรายการล่วงหน้าที่สร้างไว้แล้วจากลิสท์แสดงผังรายการล่วงหน้า และกดปุ่มอีดิท (Edit Button) ดังรูปที่ 4.36
2. ปรากฏหน้าต่างแก้ไขผังรายการล่วงหน้า (Edit Auto List Manager) ดังรูปที่ 4.37
3. ทดลองแก้ไขรายละเอียดของผังรายการล่วงหน้า โดยเปลี่ยนวันออกอากาศของผังรายการล่วงหน้าจากวันที่ 29 เป็น 21 เดือน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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



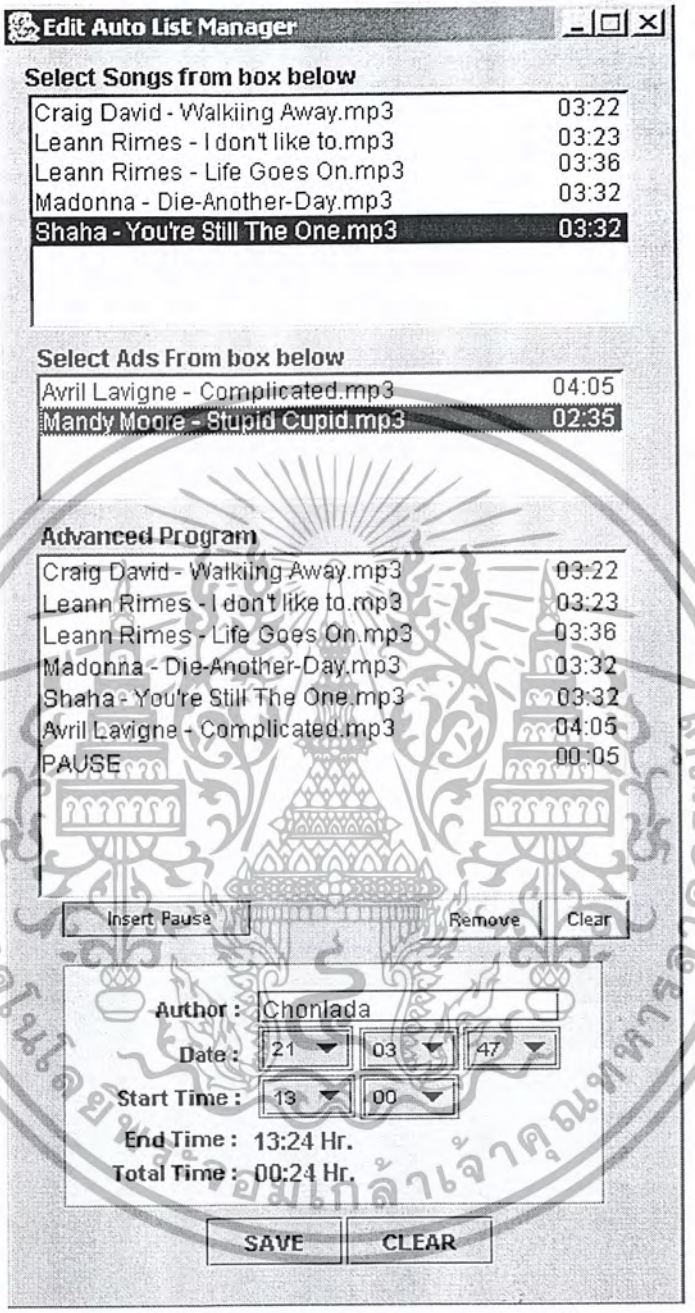
รูปที่ 4.36 วิธีการแก้ไขผังรายการล่วงหน้าที่สร้างไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.37 หน้าต่างแก้ไขผังรายการล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

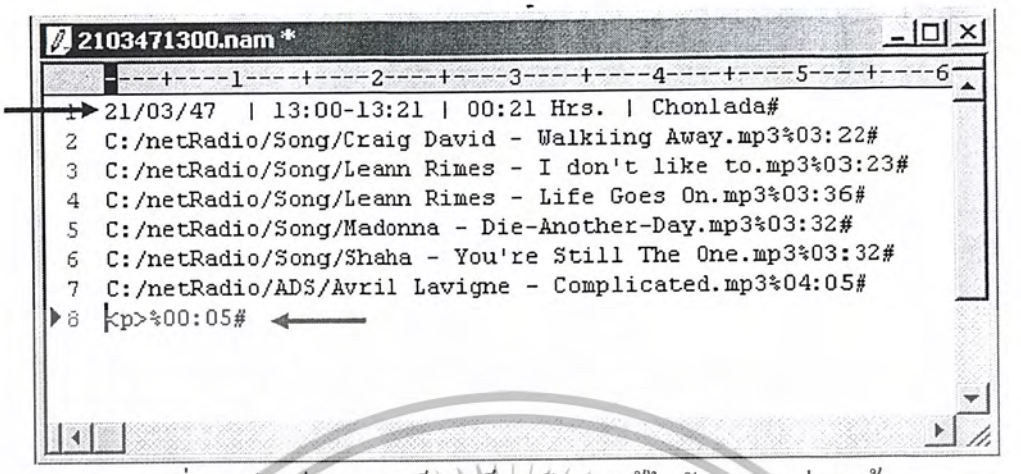


รูปที่ 4.38 ทดลองแก้ไขผังรายการล่วงหน้า

4.16.2 ผลการทดลอง

1. รายละเอียดของไฟล์นามสกุลเอ็นเอเอ็ม จะถูกเปลี่ยนแปลงไปจากเดิมจากรูปที่ 4.33 เป็นดังรูปที่ 4.39

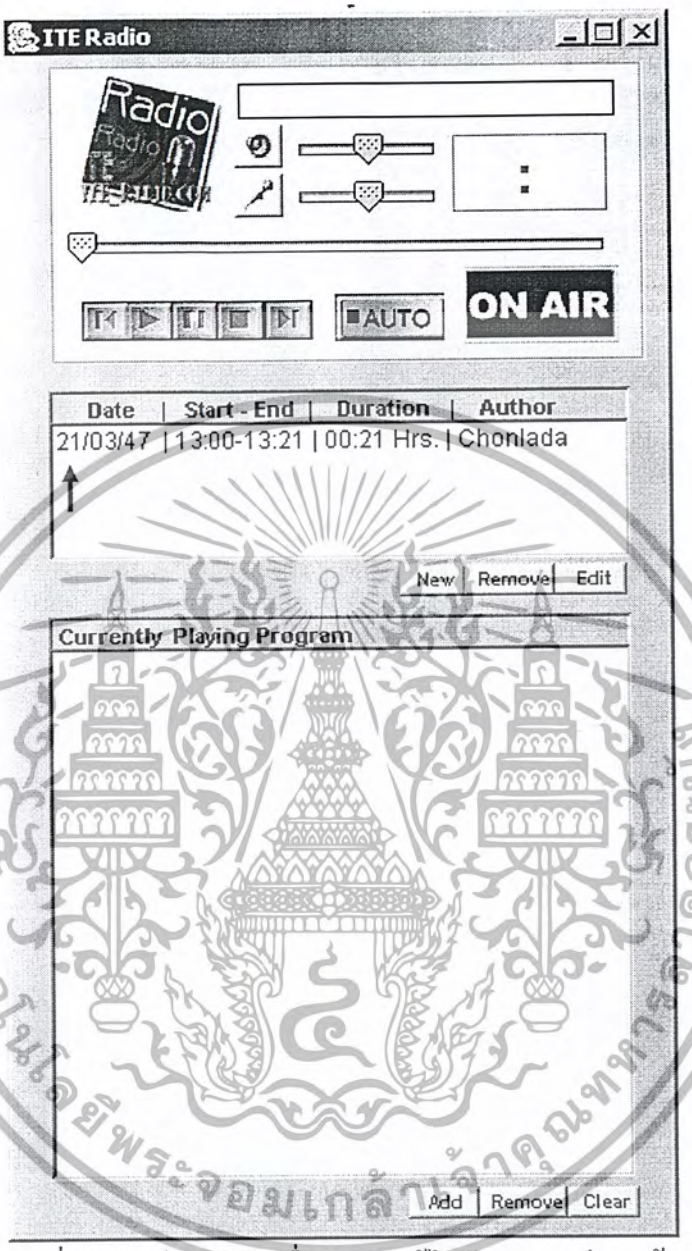
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.39 ไฟล์นามสกุลเอ็นเอเอ็มหลังการแก้ไขผังรายการล่วงหน้า

2. หน้าต่างหลัก ในส่วนแสดงรายการผังรายการล่วงหน้า จะทำการอัพเดทข้อมูลผังรายการที่แก้ไขแล้ว นำมาแสดงผล ดังรูป 4.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



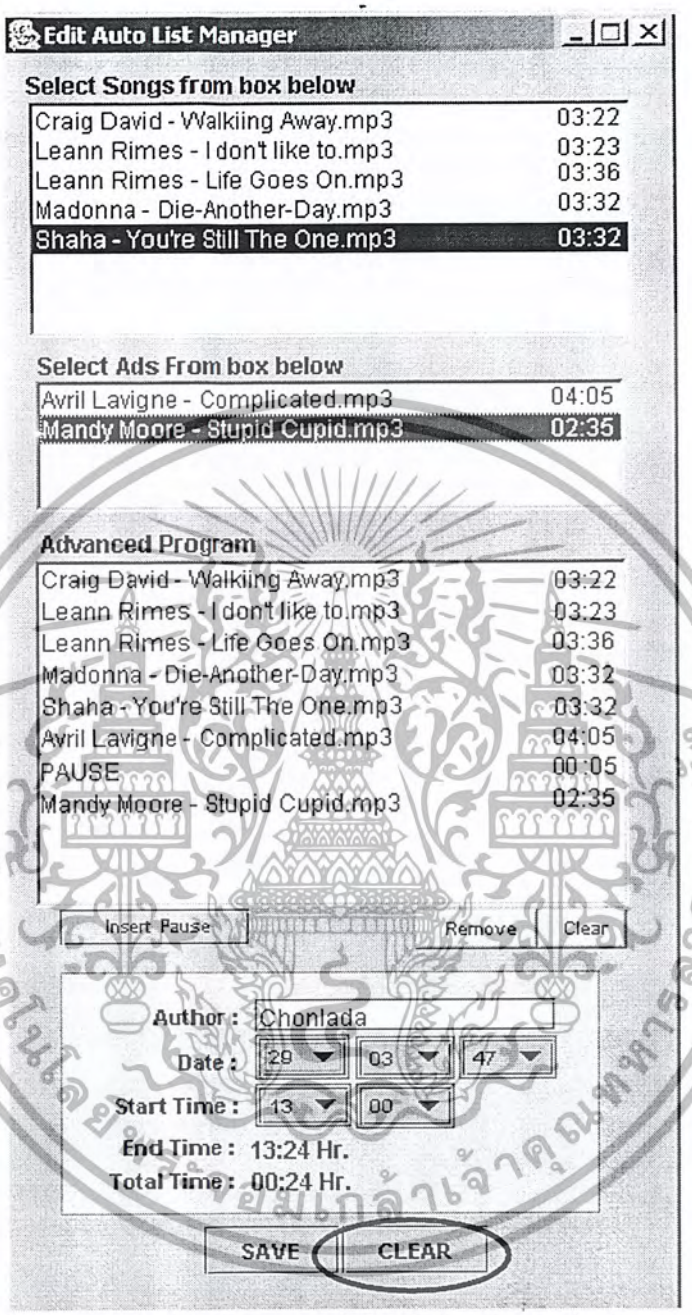
รูปที่ 4.40 หน้าต่างหลักเมื่อทำการแก้ไขผังรายการล่วงหน้า

4.17 ทดลองการล้างผังรายการล่วงหน้าที่สร้างไว้

4.17.1 วิธีทดลอง

1. หน้าต่างควบคุมแบบอัตโนมัติ กดปุ่มเคลียร์ที่อยู่ล่างสุดของหน้าต่าง

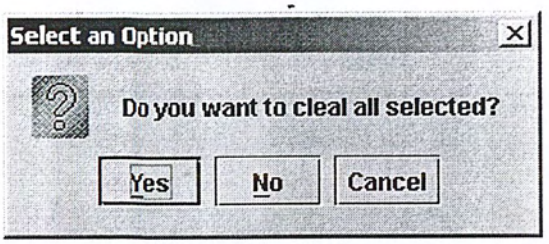
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าก่อนทำการล้างผังรายการที่สร้างไว้

2. ปรากฏไดอะล้อคบ๊อกซ์ยืนยัน (ITE Radio Alert Confirm Dialog Box) ถามว่าต้องการยืนยันการเคลียร์รายละเอียดผังรายการล่วงหน้า หรือไม่ ดังรูปที่ 4.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

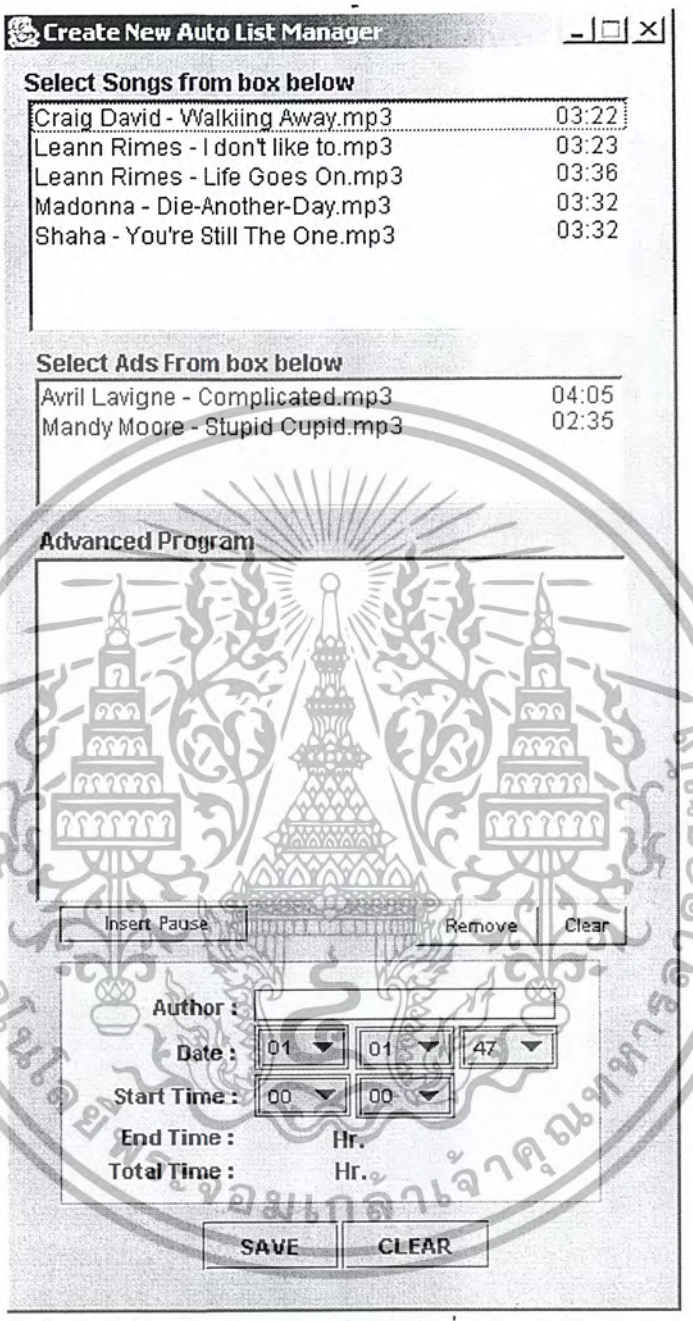


รูปที่ 4.42 ไดอะล้อคบ๊อกซ์ยืนยันการเคลียร์รายละเอียดผังรายการล่วงหน้า

4.17.2 ผลการทดลอง

1. เมื่อกดปุ่มเยส (Yes Button) จากไดอะล้อคบ๊อกซ์เพื่อยืนยันการล้างผังรายการที่สร้างไว้ รายการของมีเดียที่เลือกไว้ในผังรายการล่วงหน้า จะถูกลบออกทั้งหมด รวมถึงชื่อผู้สร้างผังรายการล่วงหน้า วันที่และเวลาที่ถูกเลือกไว้จะกลับเป็นค่าเริ่มต้น ดังรูปที่ 4.43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



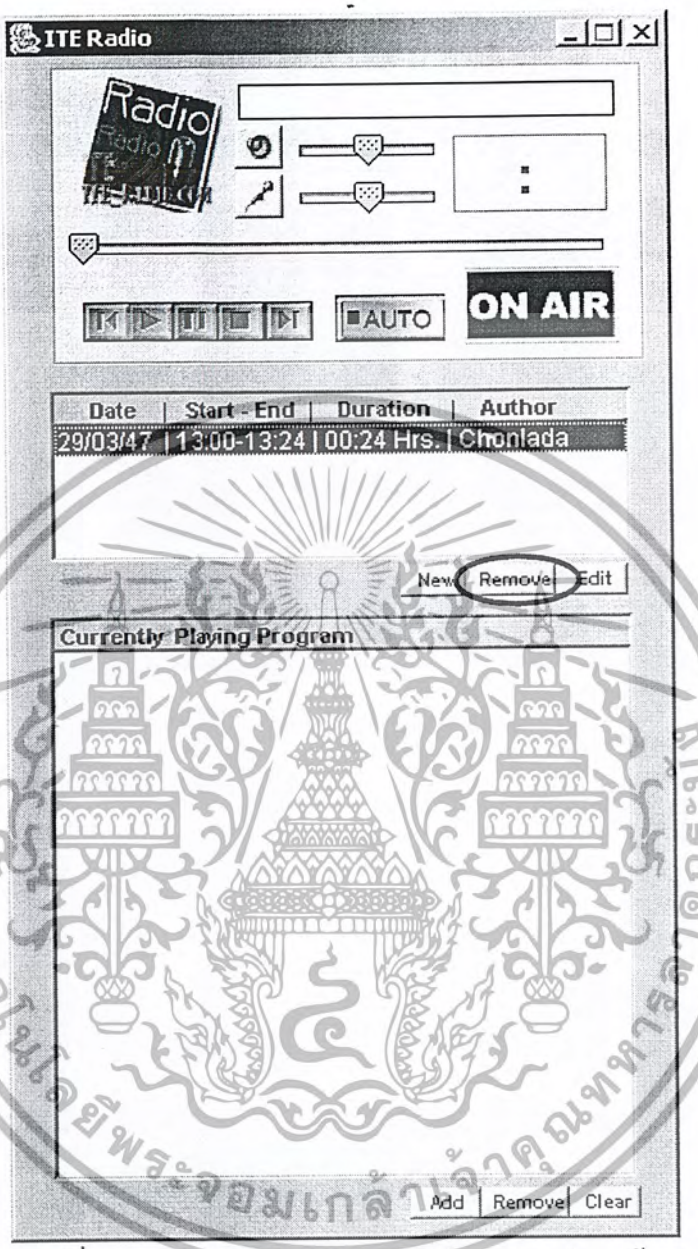
รูปที่ 4.43 หน้าต่างสำหรับสร้างผังรายการล่วงหน้าเมื่อทำการล้างผังรายการที่สร้างไว้

4.18 ทดลองลบผังรายการล่วงหน้า

4.18.1 วิธีทดลอง

1. เลือกผังรายการล่วงหน้าจากลิสท์แสดงรายการผังรายการล่วงหน้า (Auto Playlist) กดปุ่มดีลีท (Delete Button)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

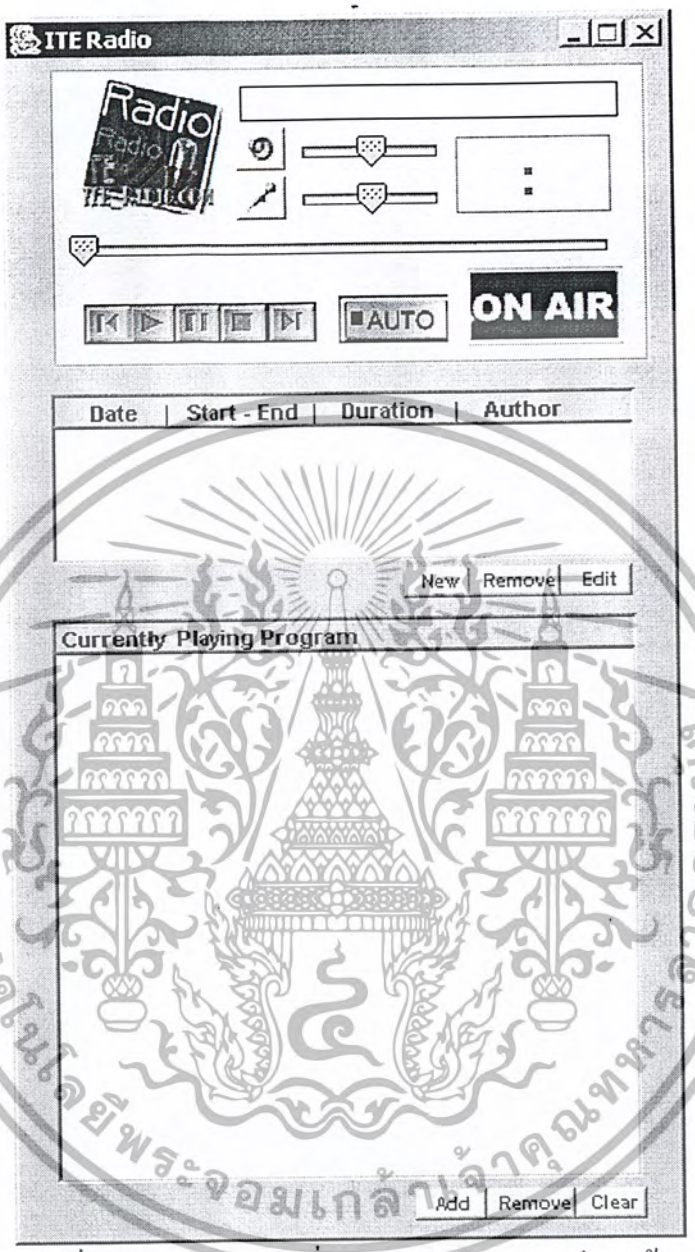


รูปที่ 4.44 หน้าต่างหลัก ก่อนการลบผังรายการล่วงหน้า

4.18.2 ผลการทดลอง

1. หน้าต่างหลัก รายการผังรายการล่วงหน้าที่ต้องการลบ จะถูกลบออกจากลิสท์แสดงรายการผังรายการล่วงหน้า ดังรูปที่ 4.45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



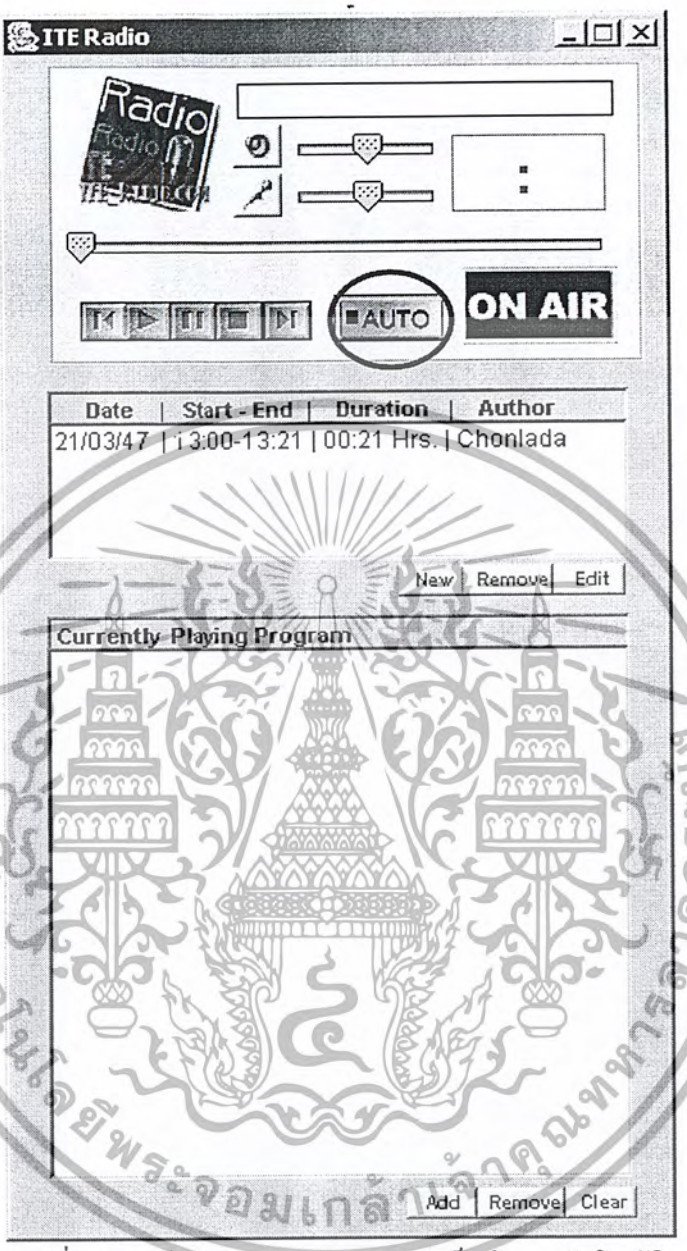
รูปที่ 4.45 หน้าต่างหลักเมื่อทำการลบผังรายการล่วงหน้า

4.19 ทดลองสลับการทำงานเป็นโหมดอัตโนมัติ

4.19.1 วิธีทดลอง

กดปุ่มออโต้ (Auto Button) ที่ส่วนควบคุมของหน้าต่างหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

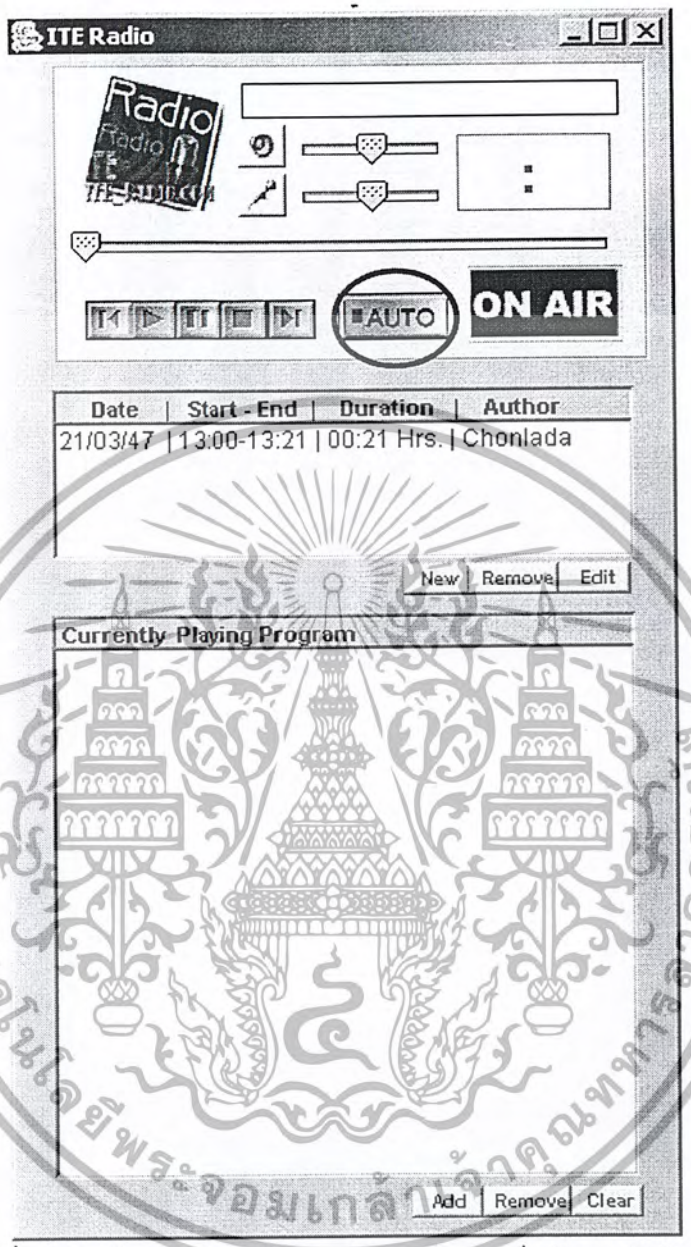


รูปที่ 4.46 หน้าต่างหลัก ก่อนการสลับเป็นโหมดอัตโนมัติ

4.19.2 ผลการทดลอง

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

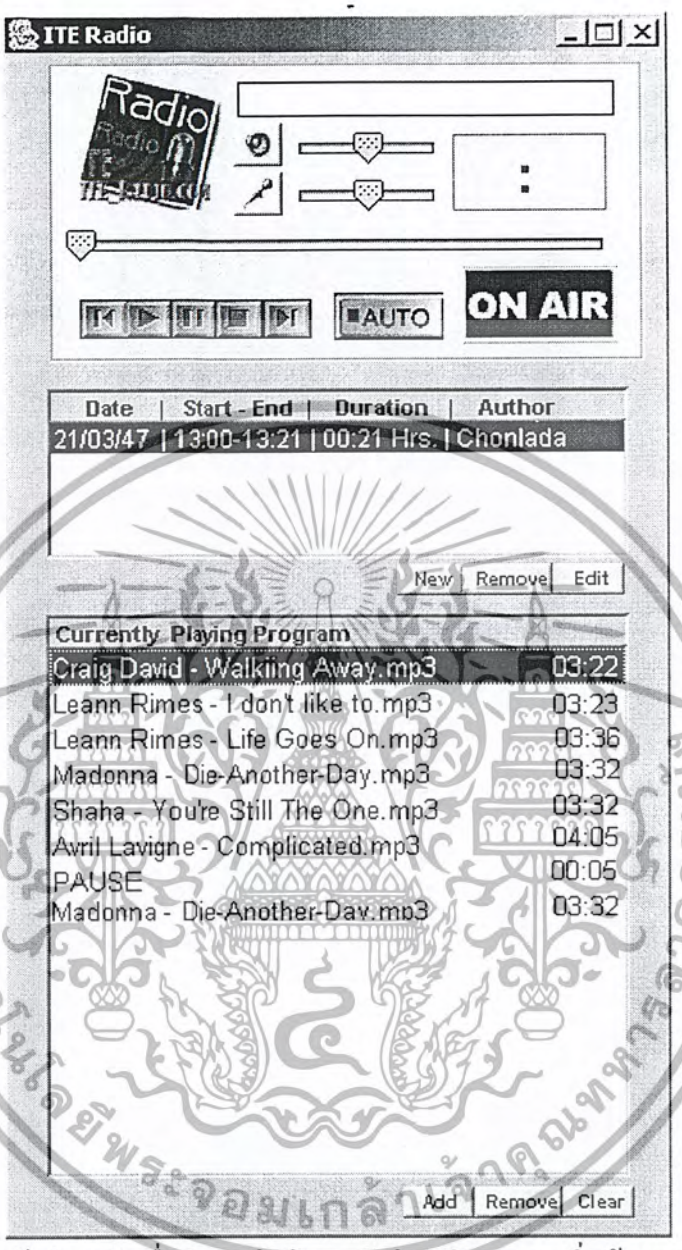
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.47 หน้าต่างหลักหลังจากสลับการทำงานเป็นโหมดอัตโนมัติ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



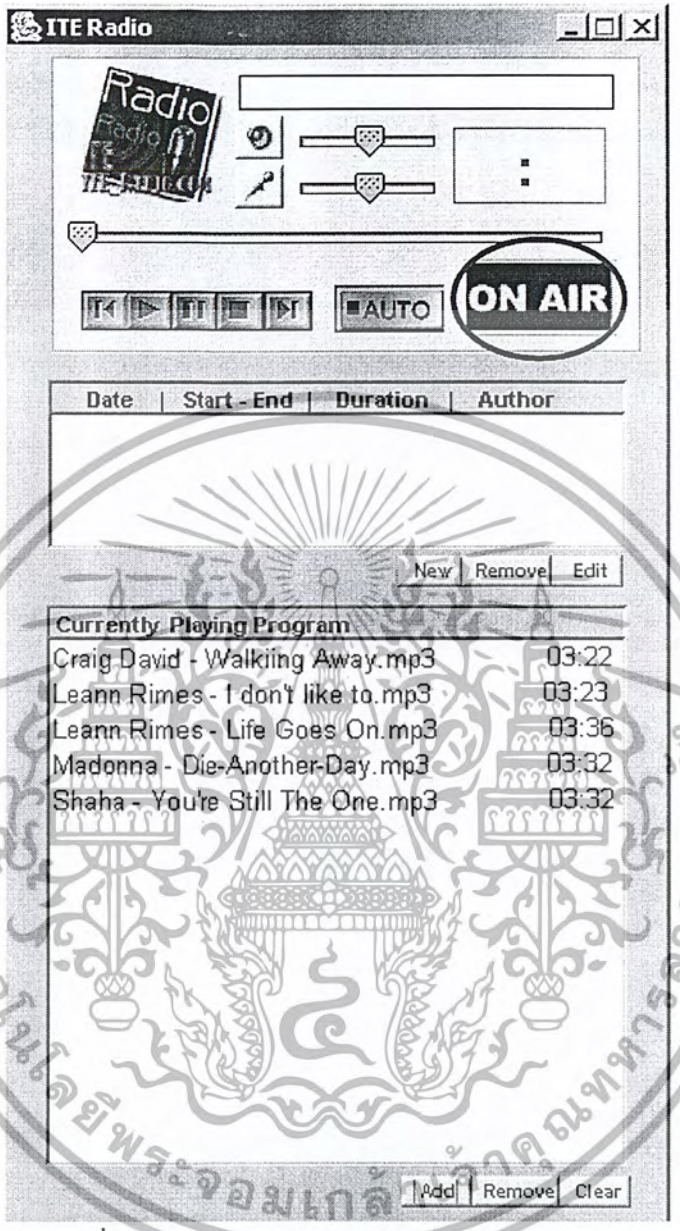
รูปที่ 4.48 หน้าต่างหลักเมื่อทำงานในโหมดอัตโนมัติตามเวลาเริ่มต้นของผังรายการ

4.20 ทดลองส่งข้อมูลเสียงไปยังไคลเอนท์

4.20.1 วิธีทดลอง

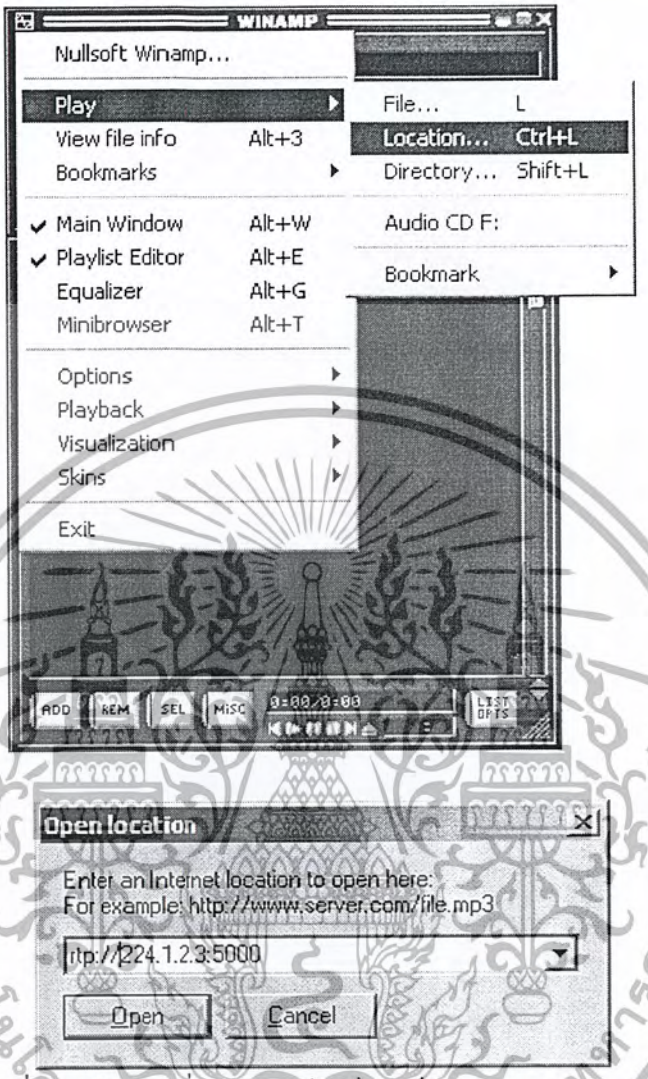
1. กดปุ่มออนแอร์ที่หน้าต่างควบคุมหลัก
2. เปิดโปรแกรมวินแอมป์ (Winamp) ติดต่อไปยังมัลติคาสท์ไอพีแอดเดรส ผ่านโปรโตคอลอาร์ทีพี มีรูปแบบคอนเนคชั่นสตริงคือ “rtp://224.1.2.3:5000”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.49 วิธีทดลองส่งข้อมูลเสียงไปยังไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.50 วิธีการเชื่อมต่อของไคล์เอนท์โปรแกรมมายังระบบ

4.20.2 ผลการทดลอง

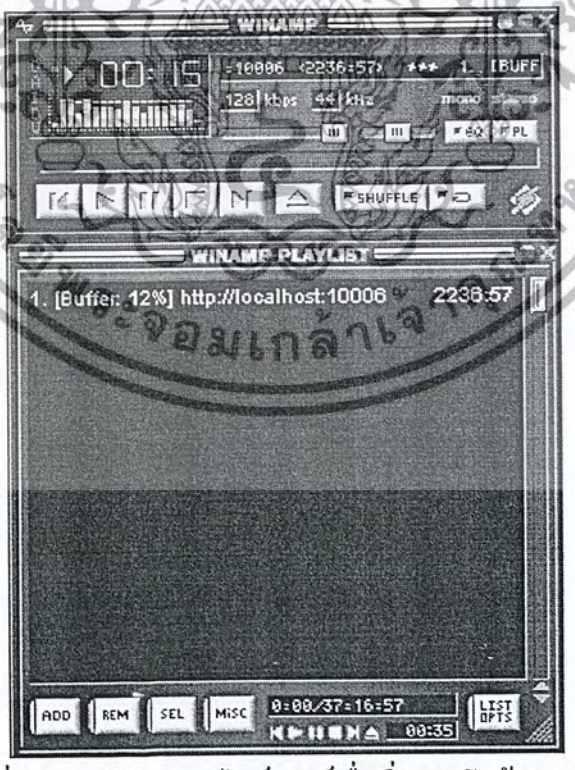
จากรูปที่ 4.51 รูปออนแอร์ (ON AIR) ของปุ่มออนแอร์ที่หน้าต่างหลักเปลี่ยนเป็นสีสว่างขึ้น แสดงถึงสถานะออนแอร์ จากรูปที่ 4.52 ก. ไคล์เอนท์โปรแกรมแสดงสถานะกำลังเชื่อมต่อไปยังเซิร์ฟเวอร์ และรูปที่ 4.52 ข. ไคล์เอนท์โปรแกรมเมื่อเริ่มการรับสตรีมเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.51 หน้าต่างหลักเมื่อทำการส่งข้อมูลเสียงไปยังไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.52 ก. โปรแกรมไคล์เอนท์เมื่อเริ่มทำการเชื่อมต่อไปยังเซิร์ฟเวอร์



รูปที่ 4.52 ข. โปรแกรมไคล์เอนท์เมื่อเริ่มการรับข้อมูลเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.21 ทดลองปรับระดับเสียงของระบบ

4.21.1 วิธีทดลอง

เลื่อนสไลด์บาร์ของระบบ (Internet Radio Slide Bar) ไปข้างหน้า หากต้องการเพิ่มเสียงของระบบ เลื่อนย้อนกลับเมื่อต้องการลดความดังเสียงของระบบ

รูปที่ 4.53 วิธีปรับเสียงระดับเสียงของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.21.2 ผลการทดลอง

โปรแกรมไคลเอนท์ยังคงรับสตรีมเสียงได้อยู่ และมีระดับเสียงที่เปลี่ยนไปตามเซิร์ฟเวอร์กำหนด โดยสถานะภายนอกไม่มีการเปลี่ยนแปลง

รูปที่ 4.54 โปรแกรมไคล์เอนท์เมื่อทำการปรับระดับเสียงของระบบ

4.22 ทดลองปรับระดับเสียงไมโครโฟน

4.22.1 วิธีทดลอง

เลื่อนสไลด์บาร์ของไมโครโฟน (Microphone Slide Bar) ไปข้างหน้า หากต้องการเพิ่มเสียงไมโครโฟน เลื่อนย้อนกลับเมื่อต้องการลดความดังเสียงไมโครโฟน

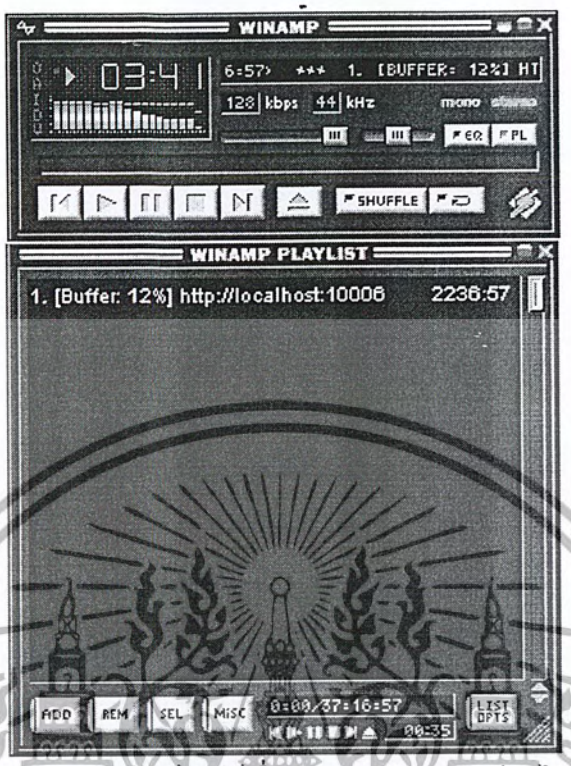
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.55 วิธีปรับเสียงไมโครโฟน

4.22.2 ผลการทดลอง

โปรแกรมไคลเอนท์ยังคงรับสตรีมเสียงได้อยู่ และมีระดับเสียงไมโครโฟนที่เปลี่ยนไปตามเซิร์ฟเวอร์กำหนด โดยสถานะภายนอกไม่มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.56 โปรแกรมไคล์เอนท์เมื่อทำการปรับระดับเสียงไมโครโฟน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

5.1.1 ลักษณะของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เนตโดยรวม

5.1.1.1 ส่วนจัดการผังรายการ (Program Management Module)

ทำหน้าที่จัดทำผังรายการปัจจุบันที่กำลังออกอากาศ และสร้างผังรายการล่วงหน้าสำหรับออกอากาศอัตโนมัติตามเวลาที่กำหนด สามารถเลือกโหมดการทำงานได้ว่าต้องการเลือกใช้ผังรายการล่วงหน้าหรือไม่

5.1.1.2 ส่วนมิกซ์เซอร์ (Broadcast Server)

ทำหน้าที่ผสมเสียงและควบคุมระดับความดังของเสียงจากสองแหล่ง คือ เสียงจากไมโครโฟนและเสียงเพลงจากซาวด์การ์ด การผสมเสียงคือการที่สามารถเลือกเปิดหรือปิดเสียงใดเสียงหนึ่ง และการเปิดหรือปิดทั้งสองเสียงได้

5.1.1.3 ส่วนบรอดคาสซ์เซิร์ฟเวอร์ (Broadcast Server)

ทำหน้าที่รับข้อมูลเสียงจากแหล่งเสียงที่เป็นไฟล์เช่นไฟล์เสียงเพลงหรือไฟล์เสียงโฆษณา และแหล่งเสียงที่เป็นไมโครโฟน จากนั้นทำการจัดรูปแบบของข้อมูลให้สามารถส่งแบบเรียลไทม์ผ่านโปรโตคอลอาร์ทีพีได้ โดยการนำข้อมูลเสียงที่จับ (Capture) ได้ มาสร้างเป็นแพ็คเกตอาร์ทีพี (RTP Packets) และหน้าที่สุดท้ายนำเสียงที่จับได้ส่งไปยังไคล์เอนท์ที่อยู่ในมัลติคาสท์กรุ๊ปแอดเดรสผ่านเครือข่ายที่สนับสนุน
มัลติคาสท์

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 ส่วนมิกซ์เซอร์ (Broadcast Server)

สามารถทำหน้าที่ผสมเสียง และควบคุมระดับความดังของเสียงจากสองแหล่งคือ เสียงจากไมโครโฟนและเสียงเพลงจากซาวด์การ์ด ได้ตามที่ออกแบบไว้

5.1.2.3 ส่วนบรอดคาสซ์เซิร์ฟเวอร์ (Broadcast Server)

สามาถรับข้อมูลเสียง และทำการจัดรูปแบบของข้อมูลให้สามารถส่งแบบเรียลไทม์ผ่านโปรโตคอลอาร์ทีพีได้ โดยสามารถสร้างเป็นแพ็คเกตอาร์ทีพี (RTP-Packets) และกระจายแพคเกตเหล่านั้นไปยังไคล์เอนท์ที่อยู่ในเครือข่ายแลน (LAN) เดียวกันได้ เนื่องจากไม่มีอุปกรณ์เครือข่ายที่สนับสนุนมัลติคาสท์ทำให้ไม่สามารถทำการทดลองข้ามเครือข่ายได้ แต่จากการทดลองในเครือข่ายเดียวกันโดยใช้หลักการเดียวกันปรากฏว่าสามารถทำได้ จึงสามารถสรุปว่าส่วนบรอดคาสท์เซิร์ฟเวอร์สามารถใช้งานได้ตามที่ออกแบบ

5.2 ปัญหาที่เกิดขึ้นระหว่างการทดลอง

5.2.1 ปัญหาที่เกิดระหว่างขั้นตอนการสร้างระบบ

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

การใช้ชุดเครื่องมือภาษาจาวาที่ต่างรุ่นกันในการพัฒนาโปรแกรมเดียวกันเป็นปัญหาสำคัญ เนื่องจากในรุ่นที่ใหม่กว่า จะมีบางคลาสที่ถูกออกแบบให้เปลี่ยนแปลงไป ทำให้โปรแกรมที่เขียนกับรุ่นหนึ่งซึ่งสามารถทำงานได้ปกติกลายเป็นโปรแกรมที่มีความผิดพลาดในบางส่วนหากสลับชุดเครื่องมือต่างรุ่นกันทำงาน บางกรณีถึงขั้นคอมไพล์ไม่ผ่าน และไม่สามารถทำงานได้เลยทีเดียว

5.2.1.3 การใช้เครื่องมือสำหรับเขียนโปรแกรมต่างยี่ห้อกัน

การใช้ชุดเครื่องมือสำหรับเขียนโปรแกรมที่ต่างยี่ห้อกันในการพัฒนาตัวโปรแกรมเดียวกันเป็นปัญหาสำคัญ ปัญหาเกิดจากสองประเด็นหลักคือ ข้อหนึ่งชุดเครื่องมือแต่ละยี่ห้อจะมีมาตรฐานต่างกันเล็กน้อยถึงแม้ว่าจะอยู่บนพื้นฐานของโครงสร้างภาษาจาวาก็ตาม เช่น ในช่วงแรกมีผู้พัฒนาสองคน คนหนึ่งใช้โปรแกรมอิดิทพลัสในการพัฒนาโปรแกรม ส่วนอีกคนใช้โปรแกรมบอร์แลน เจบิลด์เดอร์ ส่วนที่ต่างกันคือ อิดิทพลัสจะไม่มีการใช้แพคเก็จ (Package) เมื่อเรานำซอร์สโค้ด (Source Code) ของคนแรกมาสั่งให้มันทำงานบนสิ่งแวดล้อม (Environment) ของคนที่สอง ซอร์สโค้ดนั้นจะมีปัญหาเกิดขึ้น ทำให้โปรแกรมไม่สามารถทำงานได้ตามปกติ ต้องทำการแก้ไขโค้ดให้ตรงกับมาตรฐานของอีกฝ่ายก่อน และข้อที่สอง สำหรับชุดเครื่องมือพัฒนาบางยี่ห้อก็อาจมีแพคเก็จพิเศษที่สร้างมาเฉพาะของยี่ห้อนั้น ซึ่งถ้ามีการใช้งานแพคเก็จพิเศษนี้ ผู้พัฒนาร่วมที่ใช้ชุดเครื่องมือพัฒนาต่างกัน จะต้องทำการเพิ่มแพคเก็จพิเศษนี้ในสิ่งแวดล้อมของตอนเองจึงจะทำให้โปรแกรมสามารถทำงานได้ตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ปัญหาที่เกิดระหว่างขั้นตอนการทดลอง

5.2.2.1เครือข่ายและอุปกรณ์เครือข่ายไม่สนับสนุนการทดลองส่งข้อมูลแบบมัลติคาสท์

เป็นปัญหาที่ใหญ่ที่สุดในการทดลองครั้งนี้ เนื่องจากไม่มีอุปกรณ์เครือข่ายที่สนับสนุนมัลติคาสท์ ทำให้ไม่สามารถทำการทดลองข้ามเครือข่ายได้ หรือทำการทดลองผ่านโมเด็มได้ แม้ว่าจากการทดลองในเครือข่ายเดียวกันโดยใช้หลักการเดียวกันสามารถสรุปได้ว่า ส่วนบรอดคาสท์เซิร์ฟเวอร์สามารถใช้งานได้ตามที่ออกแบบ แต่การที่ไม่ได้ทำการทดลองจริง ทำให้ไม่ทราบรายละเอียดสำคัญที่แท้จริงได้ เช่นค่าดีเลย์ ทราฟฟิค ของการเชื่อมต่อผ่านเนตเวิร์กขนาดใหญ่

5.2.2.2 การขาดแคลนอุปกรณ์มิกซ์เซอร์ (ชนิดฮาร์ดแวร์) ซึ่งมีราคาแพง

มิกซ์เซอร์เป็นอุปกรณ์ที่มีราคาแพง และมีขนาดใหญ่ไม่สะดวกต่อการนำมาทดลองบ่อยครั้ง ทำให้การทดลองในส่วนมิกซ์เซอร์ที่เป็นฮาร์ดแวร์ต้องรอโอกาสที่เหมาะสม ซึ่งเป็นเวลาไม่มากนัก เป็นอีกปัญหาหนึ่งในการทำการทดลอง

5.2.3 ปัญหาอื่น ๆ

5.2.3.1 ลิขสิทธิ์ของเอ็มพีทรีเอนโค้ดเดอร์

ปัจจุบันทางซันไมโครซิสเต็มส์ (Sun Microsystems) ได้ตัดส่วนเอ็มพีทรีเอนโค้ดเดอร์ออกจากจาวามีเดียเฟรมเวิร์กเรียบร้อยแล้ว เนื่องจากไม่ต้องการให้เกิดปัญหาละเมิดลิขสิทธิ์ของผู้คิดค้นเอ็มพีทรีเอนโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนาระบบต่อ

5.3.1 การพัฒนาส่วนจัดการการกระจายเสียง

ส่วนจัดการการกระจายเสียง (Program Management Module) ของระบบจัดการวิทยุกระจายเสียงผ่านอินเทอร์เนตนั้น นับเป็นจุดสำคัญจุดหนึ่งที่ทำให้ระบบนี้ต่างจากวิทยุกระจายเสียงผ่านอินเทอร์เนตโดยทั่วไป ระบบสามารถนำมาพัฒนาเพิ่มเติมให้สามารถอำนวยความสะดวกให้แก่ผู้ใช้ได้มากยิ่งขึ้น จากการวิเคราะห์สามารถพัฒนาเพิ่มเติมในส่วนต่าง ๆ ของระบบดังนี้

5.3.1.1 พัฒนาส่วนการจัดเก็บไฟล์เสียงและแสดงรายละเอียดของไฟล์เสียง

พัฒนาให้สามารถเก็บรายละเอียดของไฟล์เสียงเพลง และเสียงโฆษณาได้มากขึ้นเช่น ชื่อวงหรือชื่อศิลปิน ชื่ออัลบั้ม ปีที่ออกอัลบั้ม แนวเพลง ชื่อเพลง ผู้แต่ง เนื้อ-ร้อง และสามารถแบ่งแสดงไฟล์ตามหมวดหมู่ได้

5.3.1.2 พัฒนาส่วนแสดงผังรายการล่วงหน้า

พัฒนาให้สามารถแสดงรายละเอียดได้มากขึ้นเช่น แฟลก (Flag) สำหรับระบุว่าผังรายการนี้มีการใช้ไปแล้วหรือยัง และอาจจะพัฒนาให้ระบบนั้นมีการเก็บล็อกไฟล์ (Log Files) ของการใช้ผังรายการล่วงหน้าเพื่อให้ผู้ควบคุมสามารถทราบรายละเอียดหากเกิดปัญหาระหว่างการออกอากาศ

5.3.2 การพัฒนาโมดูลทางไคล์เอนท์เพื่อแก้ไขปัญหาลิขสิทธิ์เอ็มพีทรีเอนโค้ดเดอร์

พัฒนาโมดูลหรือปลั๊กอินสำหรับโปรแกรมทางฝั่งไคล์เอนท์ขึ้นมาเอง สำหรับการถอดรหัสสัญญาณสตรีมเสียงแบบอ็อก-วอร์บิส (Ogg-Vorbis) ซึ่งเป็นมาตรฐานการเข้ารหัสไฟล์เสียงชนิดใหม่ที่แจกฟรี ไม่เสียค่าลิขสิทธิ์ และมีแนวโน้มว่าจะเป็นที่นิยมทดแทนมาตรฐานแบบเอ็มพีทรีในอนาคตอันใกล้นี้ ในส่วนฝั่งเซิร์ฟเวอร์ซึ่งต้องใช้การเข้ารหัสแบบอ็อก-วอร์บิสนั้น ทางซันไมโครซิสเต็มส์ผู้ผลิตจาวาได้พัฒนาโมดูลไว้เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ 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 ปุณณชัยยะ, ต้น ตัณฑ์สุทธิวงศ์, สุพจน์ ปุณณชัยยะ **เปิดโลก TCP/IP และโปรโตคอลของอินเตอร์เน็ต** (กรุงเทพมหานคร : บริษัท ด่านสุทธาการพิมพ์, 2545)
8. ดร.วีระศักดิ์ ซึงถาวร, **JAVA Programming Volume I**
   (กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2545)
9. ดร.วีระศักดิ์ ซึงถาวร, **JAVA Programming Volume II**
   (กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), 2545)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้