การศึกษาการควบคุมความผิดพลาดโดยใช้การปรับให้เรียบในการสื่อสารเสียง

A STUDY OF ADAPTIVE ERROR CONTROL WITH VARIABLE SMOOTHING FACTOR IN AUDIO COMMUNICATION

กรหทัย กลับพงศ์

KORNHATHAI KLAPPONG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2550

การศึกษาการควบคุมความผิดพลาดโดยใช้การปรับให้เรียบในการสื่อสารเสียง

A STUDY OF ADAPTIVE ERROR CONTROL WITH VARIABLE SMOOTHING FACTOR IN AUDIO COMMUNICATION

กรหทัย กลับพงศ์

KORNHATHAI KLAPPONG

เลขหมู่.....
เลขทะเบียน 74447
วัน,เดือน,ปี - 1 ต.ค. 2550

.b 11824528
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2550

A STUDY OF ADAPTIVE ERROR CONTROL WITH VARIABLE SMOOTHING FACTOR IN AUDIO COMMUNICATION

KORNHATHAI KLAPPONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN TELECOMMUNICATION ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2007

COPYRIGHT 2007

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การศึกษาการควบคุมความผิดพลาดโดยใช้แฟคเตอร์การปรับให้เรียบในการสื่อสารเสียง

A Study of Adaptive Error Control with Variable Smoothing Factor in Audio Communication

นักศึกษา นางสาวกรหทัย กลับพงศ์

รหัสประจำตัว 46064125

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมโทรคมนาคม

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.สมยศ จุณณปิยะ

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.พิเชฐ	ม่วงนวล	
ผศ.ดร.พิพัฒน์	พรหมมี	
ผศ.ดร.เผ่าภัค	ศิริสุข	
ดร.มนตรี	คำเงิน	
รศ.สมยศ	จุณณะปิยะ	

วัน / เดือน/ ปี ที่สอบ 28 พฤษภาคม 2550 เวลา 11.30-13.30 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-402)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.จารุวัตร เจริญสุข)

คณบดีบัณฑิตวิทยาลัย

วันที่.....30.....**เดือน**.....พฤษภาคม.....**พ.ศ.**.....2550.....

หัวข้อวิทยานิพนธ์	การศึกษาการควบคุมความผิดพลาดโดยใช้แฟคเตอร์การปรับให้เรียบในการสื่อสารเสียง
นักศึกษา	น.ส. กรหทัย กลับพงศ์
รหัสนักศึกษา	46064125
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมโทรคมนาคม
พ.ศ.	2550
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.สมยศ จุณณะปิยะ

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นการศึกษาการควบคุมความผิดพลาดแบบปรับตัวในการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต ข้อจำกัดอย่างหนึ่งของเครือข่ายอินเตอร์เน็ตคือ ไม่มีการรับประกันคุณภาพการบริการ ดังนั้นจึงไม่สามารถที่จะรับประกันได้ว่าคุณภาพของเสียงที่ผู้รับได้รับเหมือนกับต้นฉบับทุกประการ ซึ่งอาจจะมีข้อมูลบางส่วนเกิดการสูญหายในระหว่างการเดินทาง ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่ผู้ส่งจะต้องรับทราบถึงคุณภาพของเสียงที่ผู้รับได้รับเพื่อปรับเปลี่ยนพารามิเตอร์ในการสื่อสารเสียงให้เหมาะสมกับสภาพเครือข่ายที่ใช้งาน เพื่อป้องกันไม่ให้คุณภาพของเสียงที่ผู้รับได้รับอยู่ในระดับที่ต่ำจนเกินกว่าจะยอมรับได้ โดยได้ศึกษาวิธีการควบคุมคุณภาพเสียงมีพื้นฐานมาจากวิธีการที่เรียกว่า FEC (Forward Error Correction) แบบ Media Specific และใช้ค่าคงที่เพื่อปรับให้เรียบชนิดปรับค่าเพื่อลดอัตราการสูญหายของแพ็กเก็ต

Thesis Title	A study of adaptive error control with variable smoothing factor in audio communication
Student	Ms. Kornhathai Klappong
Student ID	46064125
Degree	Master of Engineering
Program	Telecommunications Engineering
Year	2007
Thesis Advisor	Assoc. Prof. Somyot Chunnapiya

ABSTRACT

This thesis is the study of adaptive error control in audio transmission. Over Internet Network has sufficient capability to support multimedia communications. The limitation of the Internet service is not guarantee the quality of service (QoS). It cannot guarantee the received media have same quality as the source due to the loss of data. Sender is necessary to know about the quality of media at the receiver and change some parameters of the transmission of each media based on network characteristics in order to control the quality of the received media not to be unacceptable. In this paper, a mechanism for control the quality of audio is proposed. The proposed algorithm for adaptive error control of audio transmission is based on forward error correction (FEC) in media-specific method and varying smoothing factor for decreasing packet loss rate.

กิตติกรรมประกาศ

ขอขอบพระคุณรองศาสตราจารย์สมยศ จุณณะปิยะ ที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้คำปรึกษา และ ผู้ช่วยศาสตราจารย์พิเชฐ ม่วงนวลชี้แนะแนวทาง และให้ความรู้ในด้านต่างๆ ในการทำวิจัย ตลอดจนช่วยตรวจและแก้ไขวิทยานิพนธ์ให้เป็นไปอย่างสมบูรณ์

ขอขอบพระคุณ คุณพ่อ คุณแม่และพี่ๆทุกคนที่ให้โอกาสและสนับสนุนให้ข้าพเจ้าได้มีโอกาสเล่าเรียนจนถึงระดับปริญญามหาบัณฑิต ที่ให้กำลังใจในการเล่าเรียนและทำวิทยานิพนธ์จนสำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณคณาจารย์ และบุคลากรในภาควิชาวิศวกรรมวิศวกรรมโทรคมนาคมทุกท่านที่ให้คำปรึกษาและความช่วยเหลือในระหว่างการทำวิทยานิพนธ์

ขอขอบคุณนักศึกษาปริญญาโทสาขาวิชาวิศวกรรมโทรคมนาคมทุกท่านที่ได้ให้คำแนะนำและเป็นกำลังใจมาโดยตลอด

กรหทัย กลับพงศ์

สารบัญ

หน้า

บทคัดย่อ I

ABSTRACT II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

รายการสัญลักษณ์ X

บทที่ 1 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 1

1.3 สมมติฐานของการศึกษา 1

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย 2

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน 2

1.6 ขอบเขตของการวิจัย 2

1.7 ขั้นตอนของการศึกษา 2

1.8 ภาพรวมของระบบ 3

บทที่ 2 4

2.1 การตรวจเอกสาร (Literatures Review) 4

2.1.1 การปรับอัตราการส่งข้อมูล (Rate Adaptation) 4

2.1.2 การปรับตัวที่ผู้รับเป็นตัวขับ (Receiver-Driven Adaptation) 7

2.1.3 การควบคุมความผิดพลาดแบบปรับตัว (Adaptive Error Control) 8

2.2 การสื่อสารเสียง 10

2.2.1 ขั้นตอนในการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต 10

2.2.2 การบีบอัดเสียง (Audio Compression) 11

2.2.3 การบีบอัดเสียงด้วย G.723.1 12

2.2.4 ปัจจัยที่มีผลกระทบต่อคุณภาพของเสียง 12

2.2.5 โปรโตคอลที่ใช้ในการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต 13

สารบัญ (ต่อ)

หน้า

2.3 เทคนิคการทำให้เรียบแบบเอกซโพเนนเชียล 14

2.3.1 ค่าคงที่ปรับให้เรียบ 16

2.3.2 การปรับค่าคงที่ปรับให้เรียบตามวิธีของ Trigg และ Leach แบบจำกัดขอบเขต 16

2.4 Real-time Transport Protocol (RTP) 17

2.4.1 บทบาทของ RTP 17

2.4.2 โครงสร้างแพ็กเก็ตของ RTP (RTP Packet Structure) 18

2.4.3 RTP Control Protocol (RTCP) 19

2.4.4 การส่งข้อมูลซ้ำของเสียงโดยใช้โปรโตคอล RTP 22

บทที่ 3 25

3.1 การควบคุมความผิดพลาดแบบปรับตัว 25

3.1.1 อัตราการสูญหายของข้อมูลเสียงเมื่อมีการใช้ FEC 27

3.1.2 อัลกอริทึม Bolot 28

3.1.3 อัลกอริทึม RCCS 31

3.1.4 อัลกอริทึม CNR 33

3.1.5 การประยุกต์ใช้การปรับให้เรียบในอัลกอริทึมควบคุมความผิดพลาดแบบปรับตัว 38

3.2 การตรวจสอบความถูกต้องของแบบจำลอง 39

3.2.1 การตรวจสอบการคำนวณค่า L_b และ L_s ของผู้รับ 39

3.2.2 การตรวจสอบการทำงานของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัว 42

บทที่ 4 46

4.1 การจำลองเครือข่ายเพื่อส่งผ่านข้อมูล 46

4.1.1 การจำลองเครือข่ายให้มีความหนาแน่นของทราฟฟิคต่ำ 47

4.1.2 การจำลองเครือข่ายให้มีความหนาแน่นของทราฟฟิคสูง 48

4.2 การประเมินผลการทำงาน 49

4.2.1 ผลการจำลองเครือข่ายให้มีความหนาแน่นของทราฟฟิคต่ำ 49

4.2.2 ผลการจำลองเครือข่ายให้มีความหนาแน่นของทราฟฟิคสูง 51

4.2.3 ผลการปรับค่าคงที่ให้เรียบในอัลกอริทึม CNR เปรียบเทียบกับอัลกอริทึม RCCS 52

สารบัญ (ต่อ)

หน้า

บทที่ 5 55
5.1 สรุปผลการวิจัย 55
5.2 ข้อเสนอแนะ 55

เอกสารอ้างอิง 56
ผลงานที่ตีพิมพ์ 59
ประวัติผู้เขียน 66

สารบัญตาราง

ตาราง หน้า

2.1 เปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดข้อมูลเสียง[2] 12

3.1 ตาราง Combination ที่ใช้อัลกอริทึม Bolot 28

3.3 ค่า Reward และ Penalty ของแต่ละ Combination ในอัลกอริทึม RCCS 31

3.4 คำอธิบายของพารามิเตอร์ที่ใช้ในอัลกอริทึม RCCS 32

3.5 Combination ของข้อมูลซ้ำที่ใช้ในอัลกอริทึม CNR 36

3.6 การตรวจสอบความถูกต้องในการคำนวณค่า L_b ของผู้รับในแบบจำลอง 41

3.7 การตรวจสอบความถูกต้องในการคำนวณค่า L_s ของผู้รับในแบบจำลอง 42

4.1 แสดงค่าพารามิเตอร์ที่สำคัญจาก Trace file เมื่อเครือข่ายมีความหนาแน่นต่ำ 51

4.2 แสดงค่าพารามิเตอร์ที่สำคัญจาก Trace file เมื่อเครือข่ายมีความหนาแน่นสูง 52

สารบัญรูป

รูป หน้า

1.1 ภาพรวมของระบบ 3

2.1 การแบ่งสถานะของผู้รับโดยใช้อัตราการสูญหายของแพ็กเก็ต[6] 5

2.2 Media Independent FEC 8

2.3 Media Specific FEC 9

2.4 การส่งเสียงพูดในรูปแบบดิจิตอลบนเครือข่ายอินเตอร์เน็ต[2] 11

2.5 บทบาทของ RTP ในการสื่อสารเสียงและวีดิทัศน์บนเครือข่ายอินเตอร์เน็ต 18

2.6 โครงสร้างแพ็กเก็ตของ RTP 18

2.7 โครงสร้างเฮดเดอร์ของ RTP[9] 18

2.8 โครงสร้างเฮดเดอร์ของแพ็กเก็ต Receiver Report (RR)[9] 21

2.9 เฮดเดอร์ของบล็อกในแพ็กเก็ต RTP ที่มีข้อมูลซ้ำของเสียง [16] 22

2.10 ตัวอย่างแพ็กเก็ต RTP เมื่อมีการส่งข้อมูลซ้ำของเสียง 1 ชุด 23

3.1 Media Specific FEC 26

3.2 การกู้คืนโดยใช้ข้อมูลซ้ำสามารถทำให้ค่า L_a น้อยกว่า L_b 27

3.3 การเพิ่มจำนวนข้อมูลซ้ำสามารถช่วยลดค่า L_a ได้ 28

3.4 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม Bolot 29

3.5 การเพิ่มและลด Combination ในอัลกอริทึม Bolot 30

3.6 Pseudo Code อัลกอริทึม Bolot 30

3.7 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม RCCS 32

3.8 Pseudo Code ของอัลกอริทึม RCCS 33

3.9 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม CNR 34

3.10 การระบุค่า L_b และ L_a ในแพ็กเก็ต Receiver Report 35

3.11 Pseudo Code ของอัลกอริทึม CNR 37

3.12 การลด Combination โดยใช้ count_L_a_under_low อย่างเดียวอาจจะล่าช้าเกินไป 38

3.13 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง 40

3.14 ตัวอย่างการหาค่า L_a เมื่อกำหนดรูปแบบการตัดทิ้งแพ็กเก็ตแบบ D02 42

3.15 การเปลี่ยนแปลงของ Combination ในระหว่างการจำลองโดยใช้อัลกอริทึม RCCS 44

3.16 การเปลี่ยนแปลงของ Combination ในระหว่างการจำลองโดยใช้อัลกอริทึม CNR 45

4.1 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง 47

สารบัญรูป (ต่อ)

รูป หน้า

4.2 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง 48
4.3 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง 49
4.4 การเปลี่ยนแปลงของ Combination โดยใช้อัลกอริทึม CNR เมื่อไม่ใช้การปรับให้เรียบ 50
4.5 การเปลี่ยนแปลงของ Combination โดยใช้อัลกอริทึม CNR เมื่อใช้การปรับให้เรียบ 50
4.6 การเปลี่ยนแปลงของ Combination โดยใช้อัลกอริทึม CNR เมื่อไม่ใช้การปรับให้เรียบ 51
4.7 การเปลี่ยนแปลงของ Combination โดยใช้อัลกอริทึม CNR เมื่อใช้การปรับให้เรียบ 52
4.8 จำนวนครั้งที่ L_3 เกินเทรสโฮลด์ HIGH (ได้รับ Receiver Report 240 ครั้ง) เมื่อใช้การปรับให้เรียบค่าต่างกันในเครือข่ายความหนาแน่นของทราฟฟิคต่ำ 53
4.9 จำนวนครั้งที่ L_3 เกินเทรสโฮลด์ HIGH (ได้รับ Receiver Report 240 ครั้ง) เมื่อใช้การปรับให้เรียบค่าต่างกันในเครือข่ายความหนาแน่นของทราฟฟิคสูง 53
4.10 การใช้ค่าคงที่ปรับให้เรียบที่แตกต่างในแต่ละสภาพเครือข่าย 54

รายการสัญลักษณ์

ACELP	Algebraic Code-Exited Linear Prediction
AIMD	Additive Increase Multiplicative Decrease
API	Application Programming Interface
FEC	Forward Error Correction
IETF	Internet Engineering Task Force
ITU	International Telecommunication Union
L_a	Loss Rate after Reconstruction
L_b	Loss Rate before Reconstruction
MPEG	Moving Picture Expert Group
MP-MLQ	Multi-Pulse Likelihood Quantization
PCM	Pulse Code Modulation
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RCCS	Redundant Codec Combination Selection
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การใช้บริการสื่อสารเสียงผ่านช่องทางสื่อสาร (Media) มีผู้ใช้งานจำนวนมากพยายามหลีกเลี่ยงการส่งผ่านข้อมูลเสียงบนเครือข่ายแบบเซอร์กิจสวิตช์ (Circuit switched Network) เพราะมีค่าใช้จ่ายสูงและการใช้บริการเสียงผ่านเครือข่ายอินเตอร์เน็ตจึงได้รับความนิยม เนื่องจากมีค่าใช้จ่ายต่ำสำหรับผู้ที่ใช้งานเครื่องคอมพิวเตอร์พีซีอยู่แล้วและสามารถติดตั้งได้เอง เครือข่ายแบบเซอร์กิจสวิตช์มีการติดต่อระหว่างผู้ใช้งานต้นทางและปลายทางโดยการเชื่อมโยงให้สองอุปกรณ์ติดต่อถึงกันทางกายภาพโดยตรงไม่มีการติดขัด ส่วนเครือข่ายอินเตอร์เน็ตเป็นเครือข่ายที่ไม่มีการรับประกันคุณภาพของการบริการ (Quality of Service) ดังนั้นจึงไม่แน่ใจได้ว่าแพ็กเก็ต (Packet) ที่บรรจุข้อมูลของเสียง จะมีการสูญหายในระหว่างการสื่อสารผ่านช่องทางการสื่อสาร ถ้าหากมีแพ็กเก็ตที่สูญหายมากเกินไปก็จะส่งผลกระทบต่อคุณภาพการสื่อสารที่ผู้รับได้รับ ตัวอย่างเช่น การสูญหายของแพ็กเก็ตเสียงทำให้เสียงไม่ต่อเนื่องและฟังไม่ชัดเจน

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อศึกษาวิธีการในการควบคุมความผิดพลาดของการสื่อสารเสียงผ่านอินเตอร์เนตที่สามารถปรับตัวได้ตามสภาพของทราฟฟิกภายในเครือข่ายโดยปรับปรุงให้อัตราการสูญหายของข้อมูลเสียงลดลง จะเป็นการเพิ่มประสิทธิภาพโดยรวมของบริการสื่อสารเสียงผ่านอินเตอร์เนต

1.3 สมมติฐานของการศึกษา

ถ้าผู้ส่งมีการตรวจสอบคุณภาพของเสียงที่ผู้รับได้รับผ่านอินเตอร์เนตแล้วจะสามารถปรับปรุงให้มีคุณภาพใกล้เคียงกับการสื่อสารผ่านเครือข่ายแบบเซอร์กิจสวิตช์ ผู้ส่งจะต้องมีกลไกในการควบคุมความผิดพลาดของแพ็กเก็ต เสียงที่สามารถปรับตัวได้ตามสภาพการใช้งานของเครือข่าย คือถ้าเครือข่ายมีความคับคั่งของทราฟฟิกต่ำก็ใช้วิธีการกำหนดพารามิเตอร์ส่งเสียงในรูปแบบหนึ่ง แต่ถ้าเครือข่ายมีความคับคั่งของทราฟฟิกสูงก็กำหนดพารามิเตอร์ของการส่งในอีกรูปแบบหนึ่งจะสามารถช่วยลดผลกระทบจากการสูญหายของแพ็กเก็ตได้ เพื่อป้องกันไม่ให้คุณภาพของเสียงที่ผู้รับได้รับอยู่ในระดับที่ต่ำจนเกินไปจนไม่สามารถที่จะยอมรับได้

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

วิธีการควบคุมความผิดพลาดแบบปรับตัวที่ศึกษาเป็นวิธีการที่ใช้พื้นฐานของ FEC ซึ่งมีอัลกอริทึมที่ใช้ในการเลือกรูปแบบการการบรรจุข้อมูลซ้ำในแต่ละแพ็กเก็ตหรือที่เรียกว่า Combination ให้เหมาะสมกับสภาพของเครือข่าย เมื่อนำเอาเทคนิคการปรับให้เรียบแบบเอกซ์โพเนนเชียลครั้งเดียวมาใช้ในอัลกอริทึมจะทำให้การปรับตัวในการส่งข้อมูลซ้ำมีประสิทธิภาพดีขึ้นและเหมาะสมกับสภาพเครือข่ายในขณะนั้น

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

วิธีการปรับให้เรียบโดยการปรับค่าคงที่ปรับให้เรียบสามารถทำให้อัลกอริทึมที่ใช้ในการส่งข้อมูลซ้ำเพื่อควบคุมความผิดพลาดส่งข้อมูลได้เหมาะสมกับสภาพเครือข่ายในขณะนั้น ๆ ในอัลกอรึทึมเดิมที่ทำการศึกษายังไม่มีการพิจารณาการใช้ค่าคงที่ปรับให้เรียบในการส่งข้อมูล ดังนั้นเมื่อประยุกต์การปรับให้เรียบในอัลกอริทึมแล้ว ทำให้การควบคุมการส่งข้อมูลซ้ำเพื่อควบคุมความผิดพลาดมีความเหมาะสมกับเครือข่ายและช่วยให้มีการใช้ทรัพยากรของระบบอย่างคุ้มค่า

1.6 ขอบเขตของการวิจัย

1) ในการสื่อสารพหุสื่ออาจมีสื่อได้หลายชนิด แต่วิธีการที่นำเสนอจะใช้ในการปรับคุณภาพของเสียงเท่านั้น
2) ในการสื่อสารเสียงใช้การบีบอัดเสียงแบบ G.723.1
3) การสื่อสารเสียงเป็นแบบจุดต่อจุด (Point-to-Point) เท่านั้น
4) การทดสอบวิธีการที่นำเสนอประกอบด้วยการสร้างแบบจำลองและมีการปรับเปลี่ยนการควบคุมความผิดพลาดของการส่งข้อมูลเสียงได้ตามสภาพของเครือข่าย

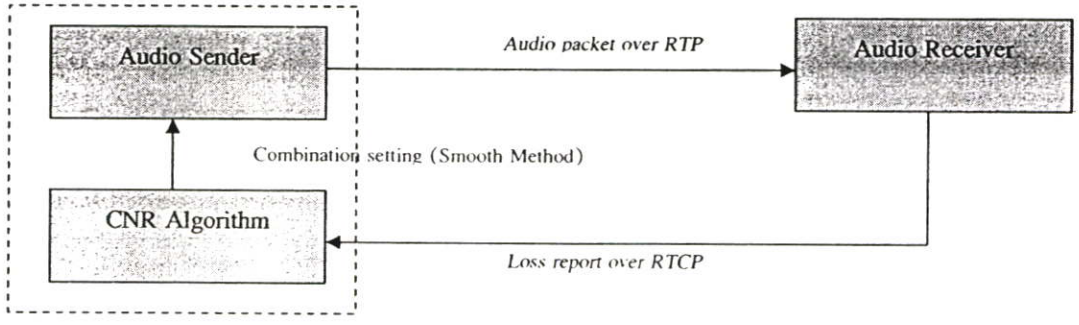
1.7 ขั้นตอนของการศึกษา

1) ศึกษาหลักการของการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต
2) ศึกษาวิธีการในการควบคุมความผิดพลาดของการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ตที่ได้มีการตีพิมพ์ลงในบทความทางวิชาการ
3) ศึกษาอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวสำหรับการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต ในกรณีที่ใช้การบีบอัดเสียงแบบ G.723.1 และทดสอบอัลกอริทึมโดยใช้แบบจำลอง
4) ศึกษาบทความทางวิชาการเกี่ยวกับเทคนิคการปรับให้เรียบแบบเอกซโพเนนเชียลครั้งเดียวในการทำนายอนุกรมเวลาอัตราการสูญหายของข้อมูล

5) นำเทคนิคการปรับให้เรียบแบบเอกซโพเนนเชียลมาพัฒนาในอัลกอริทึมควบคุมความผิดพลาดของการสื่อสารเสียงที่สามารถปรับตัวได้ตามสภาพของเครือข่าย
6) ทดสอบวิธีการควบคุมความผิดพลาดแบบปรับตัวสำหรับการสื่อสารเสียงที่ได้ออกแบบไว้ โดยใช้การทดสอบบนโปรแกรมจำลองเครือข่ายอินเตอร์เน็ต
7) รวบรวมผลการทดลองและจัดทำวิทยานิพนธ์

1.8 ภาพรวมของระบบ

ในวิทยานิพนธ์นี้ ภาพรวมของระบบเป็นดังรูปที่ 1.1 ผู้ส่งจะมีการส่งเสียงโดยใช้โปรโตคอล RTP ส่วนผู้รับจะมีป้อนกลับข้อมูลเกี่ยวกับคุณภาพของเสียงที่ได้รับ โดยใช้โปรโตคอล RTCP การควบคุมความผิดพลาดจะกระทำที่ผู้ส่งซึ่งจะมีการปรับพารามิเตอร์ในการส่งเสียงให้เหมาะสมกับสภาพเครือข่ายที่กำลังใช้งานอยู่ ทั้งนี้เพื่อควบคุมความผิดพลาดของเสียงที่ผู้รับได้รับให้อยู่ในระดับที่ยอมรับได้



รูปที่ 1.1 ภาพรวมของระบบ

บทที่ 2
ทฤษฎีและหลักการ

ในบทนี้เป็นการกล่าวถึงทฤษฎีและหลักการพื้นฐานสำหรับการทำวิทยานิพนธ์ โดยเนื้อหาที่จะกล่าวถึงในบทนี้ประกอบด้วย หัวข้อ 2.1 การตรวจเอกสาร หัวข้อ 2.2 การสื่อสารเสียง ซึ่งอธิบายถึงขั้นตอนในการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต การบีบอัดเสียงโดยเน้นที่การบีบอัดเสียงด้วย G.723.1 หัวข้อ 2.3 เทคนิคการทำให้เรียบแบบเอกซโพเนนเชียลและการปรับค่าคงที่ปรับให้เรียบ และสุดท้ายคือหัวข้อที่ 2.4 กล่าวถึงโปรโตคอลที่ใช้สำหรับการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ตซึ่งก็คือ โปรโตคอลที่ชื่อว่า Real-time Transport Protocol (RTP)

2.1 การตรวจเอกสาร (Literatures Review)

ในการตรวจเอกสารของวิทยานิพนธ์นี้ ได้ศึกษาเทคนิคในการปรับตัว (Adaptation Techniques) ของโปรแกรมประยุกต์พหุสื่อบนเครือข่ายอินเตอร์เน็ต และพบว่ามีผู้เสนอเทคนิคในการปรับตัวขึ้นมาเป็นจำนวนมาก โดยสามารถจำแนกของเทคนิคในการปรับตัวเหล่านี้ได้เป็น 3 ประเภท คือ การปรับอัตราการส่งข้อมูล (Rate Adaptation), การปรับตัวที่ผู้รับเป็นตัวขับ (Receiver-Driven Adaptation) และการควบคุมความผิดพลาดแบบปรับตัว (Adaptive Error Control)

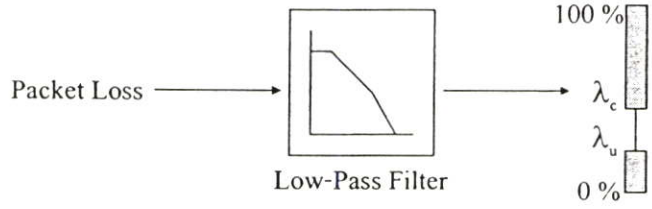
2.1.1 การปรับอัตราการส่งข้อมูล (Rate Adaptation)

ในการใช้เทคนิคการปรับตัวแบบนี้ อัตราการส่งข้อมูลของผู้ส่งจะไม่คงที่ โดยจะมีการเพิ่มหรือลดอัตราการส่งข้อมูลตามค่าที่ป้อนกลับจากผู้รับ ซึ่งเทคนิคนี้ยังสามารถแบ่งย่อยออกได้เป็น 2 ชนิดคือ การปรับตัวตามปริมาณการใช้บัฟเฟอร์ (Buffer-Based Adaptation) และการปรับตัวตามปริมาณการสูญหายของข้อมูล (Loss-Based Adaptation) โดยประเภทแรกซึ่งก็คือการปรับตัวตามปริมาณการใช้บัฟเฟอร์นั้น บัฟเฟอร์ดังกล่าวหมายถึงบัฟเฟอร์ที่ใช้พักข้อมูลจากแหล่งกำเนิดสื่อก่อนที่จะถูกส่งออกไปยังเครือข่าย โดยผู้ส่งจะพิจารณาว่าถ้าบัฟเฟอร์เกือบจะเต็มแล้วผู้ส่งก็จะลดอัตราการส่งข้อมูลลง แต่ถ้าข้อมูลในบัฟเฟอร์กำลังจะหมดผู้ส่งก็จะเพิ่มอัตราการส่งข้อมูล ในบทความ [4] และ [5] Jacobs และ ElefTheriadis ได้เสนอโปรโตคอล (Protocol) ขึ้นมาใหม่โดยใช้ Congestion Window และ Acknowledgement Message ของ Transmission Control Protocol (TCP) ในการตรวจตราความคับคั่งของเครือข่าย โดยจุดประสงค์ของโปรโตคอลนี้คือ ต้องการให้การส่งวีดิทัศน์สามารถปรับตัวเข้ากับสภาพเครือข่ายโดยมีพฤติกรรมในลักษณะเดียวกันกับการใช้โปรโตคอล TCP โปรโตคอลใหม่นี้จะมีการเก็บแพ็กเก็ตวีดิทัศน์ในบัฟเฟอร์ของผู้ส่งก่อนที่จะส่งแพ็กเก็ตเหล่านี้ไปยังเครือข่าย ซึ่ง Congestion Window จะเป็นตัวกำหนดว่าสามารถส่งแพ็กเก็ตได้

เมื่อใด และปริมาณของแพ็กเก็ตที่อยู่บัฟเฟอร์นี้เองที่จะเป็นตัวกำหนดอัตราบิต (Bit Rate) ของตัวเข้ารหัสวีดิทัศน์เพื่อไม่ให้เกิดเหตุการณ์ที่บัฟเฟอร์เต็มหรือบัฟเฟอร์ว่าง ส่วนการควบคุมตัวเข้ารหัสวีดิทัศน์ (Video Encoder) เพื่อให้ได้อัตราบิตของการส่งวีดิทัศน์ตามที่ต้องการนั้น ได้มีการใช้วิธี Dynamic Rate Shaping [6]

เทคนิคการปรับตัวตามปริมาณการใช้บัฟเฟอร์นี้เหมาะสำหรับการถ่ายทอดวีดิทัศน์ซึ่งเป็นการส่งวีดิทัศน์แบบทางเดียว แต่ไม่เหมาะสำหรับการส่งวีดิทัศน์ในลักษณะของการสนทนาเนื่องจากการใช้ TCP Congestion Windows และ Acknowledgement Message นั้นทำให้เกิดเวลาหน่วง (Delay) สูง แต่การส่งวีดิทัศน์ในการสนทนานั้นต้องการความเป็นเวลาจริง (Real-time) ซึ่งต้องการให้เวลาหน่วงในการส่งวีดิทัศน์จากผู้ส่งถึงผู้รับมีค่าต่ำที่สุดเท่าที่จะเป็นไปได้ นอกจากนี้ในการส่งเสียงหรือวีดิทัศน์ที่ต้องการความเป็นเวลาจริง ถ้ามองในระดับชั้นโปรแกรมประยุกต์ (Application Layer) แล้ว บัฟเฟอร์ที่ใช้ในการส่งข้อมูลจะมีขนาดเล็กมาก หรืออาจจะไม่มีการใช้บัฟเฟอร์เลยด้วยซ้ำ

เทคนิคการปรับอัตราการส่งข้อมูลประเภทที่สองคือ การปรับตัวตามตามปริมาณการสูญหายของข้อมูล เทคนิคนี้ผู้รับจำเป็นจะต้องป้อนกลับค่าอัตราการสูญหายของข้อมูลให้ผู้ส่งได้รับทราบ บทความ [6], [7]และ [8] ได้เสนอวิธีการปรับค่าอัตราบิตของการส่งวีดิทัศน์โดยพิจารณาจากการอัตราการสูญหายของแพ็กเก็ต วิธีการที่ใช้ในบทความทั้งสามนี้มีส่วนที่เหมือนกันก็คือ ผู้ส่งได้นิยามสถานะของผู้รับเอาไว้ 3 สถานะคือ CONGESTED, LOADED และ UNLOADED โดยใช้ค่าอัตราการสูญหายของแพ็กเก็ตในการกำหนดสถานะดังกล่าวดังร ถ้าหากผู้รับอยู่ในสถานะ CONGESTED ผู้ส่งจะลดอัตราบิตในการส่งวีดิทัศน์เพื่อลดความคับคั่งของเครือข่าย แต่ถ้าผู้รับอยู่ในสถานะ UNLOADED ผู้ส่งก็จะเพิ่มอัตราบิตเนื่องจากผู้ส่งเครือข่ายยังมีความคับคั่งต่ำ ส่วนในกรณีที่ผู้รับอยู่ในสถานะ LOADED ซึ่งเป็นสถานะที่อัตราการสูญหายของแพ็กเก็ตไม่มากและไม่น้อยเกินไป ผู้ส่งก็จะไม่มีการเปลี่ยนแปลงอัตราบิตของวีดิทัศน์



รูปที่ 2.1 การแบ่งสถานะของผู้รับโดยใช้อัตราการสูญหายของแพ็กเก็ต[6]

วิธีการของ Busse และคณะ [6] ใช้โปรโตคอล RTP (Real-time Transport Protocol) [9] ในการส่งวีดิทัศน์ ซึ่งผู้รับใช้แพ็กเก็ต RTCP ชนิด Receiver Report ในการรายงานอัตราการสูญหาย

ของแพ็กเก็ตให้ผู้ส่งได้รับทราบ แต่วิธีการของ Bolot และ[7] นั้นแตกต่างออกไป โดยผู้ส่งจะทำหน้าที่ในการสอบถามสถานะจากผู้รับเป็นระยะหรือที่เรียกว่า การโพลลิ่ง (Polling) ทั้งนี้เพื่อขจัดปัญหาในกรณีที่มีผู้รับจำนวนมากโดยเฉพาะเมื่อมีการส่งวีดิทัศน์แบบมัลติคาสต์ ซึ่งทำได้โดยการใช้วิธีการโพลลิ่งที่เรียกว่า Probabilistic Polling โดยในการส่งข้อความโพลลิ่งแต่ละครั้งผู้ส่งจะสุ่มค่าคีย์ขึ้นมาหนึ่งค่า ผู้รับที่มีค่าคีย์ตรงกันเท่านั้นจะเป็นผู้ตอบ ซึ่งวิธีการสุ่มที่ว่านี้สามารถใช้ในการประมาณจำนวนผู้รับได้

วิธีการของ Busse [6] และวิธีการของ Bolot [7] ใช้การปรับค่าอัตราบิตของวีดิทัศน์แบบ Additive Increase Multiplicative Decrease (AIMD) นั่นคือมีการเพิ่มอัตราบิตแบบผลบวก แต่ในการลดอัตราบิตนั้นเป็นการลดแบบผลคูณ ซึ่งทั้งอัตราบิตที่เพิ่มและสัดส่วนที่ใช้ในการลดอัตราบิตนั้นล้วนแล้วแต่เป็นค่าคงที่ แต่สำหรับวิธีการของ Sisalem [8] นั้นแตกต่างออกไป เนื่องจากค่าที่ใช้ในการเพิ่มอัตราบิต (Additive Increase Rate, AIR) ในแต่ละครั้งนั้นไม่เท่ากัน โดยค่า AIR นี้จะเปลี่ยนไปตามอัตราการสูญหายและค่าเวลาหน่วง (Delay) ที่ผู้รับป้อนกลับมาให้

จากการทดลองของ Sisalem และ Schulzrinne ใน [10] แสดงให้เห็นว่าในกรณีของการส่งวีดิทัศน์โดยใช้การรายงานสถานะของผู้รับตามโปรโตคอล RTP นั้น หากมีการปรับอัตราบิตแบบ AIMD นั้นจะทำให้การแบ่งแบนด์วิดท์กับโปรแกรมประยุกต์อื่นเป็นไปอย่างไม่เป็นธรรม โดยเฉพาะอย่างยิ่งโปรแกรมประยุกต์ที่ใช้โปรโตคอล TCP เนื่องจากในกรณีที่ใช้โปรโตคอล RTP นั้น ผู้รับจะรายงานสถิติของการรับให้กับผู้ส่งโดยใช้แพ็กเก็ต Receiver Report ซึ่งระยะเวลาระหว่างการส่งแพ็กเก็ต RTCP แต่ละครั้งนั้นใช้เวลาอย่างน้อย 5 วินาที แต่สำหรับโปรโตคอล TCP นั้นผู้รับมีการตอบกลับทุกครั้งที่ได้รับแพ็กเก็ต โปรแกรมประยุกต์ที่ใช้โปรโตคอล TCP จึงถูกบังคับให้ลดอัตราการส่งข้อมูลเร็วกว่าโปรแกรมประยุกต์ที่ใช้โปรโตคอล RTP ด้วยเหตุผลนี้ทำให้การปรับอัตราบิตแบบนี้ไม่เป็นธรรมต่อโปรแกรมประยุกต์ที่ใช้โปรโตคอล TCP ดังนั้น Sisalem และ Schulzrinne จึงเสนอวิธีการปรับอัตราบิตให้มีคุณสมบัติที่สามารถแบ่งปันแบนด์วิดท์กับโปรแกรมประยุกต์ที่ใช้โปรโตคอล TCP ได้อย่างยุติธรรมมากขึ้น โดยได้แก้ไขในส่วนของการเพิ่มอัตราบิตของการส่งวีดิทัศน์ โดยในการเพิ่มแต่ละครั้งจะต้องไม่เกินค่า r_{TCP} ซึ่งค่านี้ได้มาจากการโมเดล TCP Throughput ของ Padhye และคณะ[11] ซึ่งถือเป็นค่าอัตราบิตที่จะทำให้มีการแบ่งแบนด์วิดท์ของเครือข่ายอย่างยุติธรรม

เทคนิคการปรับตัวตามตามปริมาณการสูญหายของข้อมูลนี้ สามารถใช้ในการกำหนดอัตราการส่งข้อมูลให้เหมาะสมกับแบนด์วิดท์เครือข่ายที่เหลืออยู่ (Available Bandwidth) ซึ่งเหมาะสำหรับในกรณีที่เครือข่ายมีแบนด์วิดท์เหลืออยู่มากพอสมควร แต่ถ้าหากแบนด์วิดท์ของเครือข่ายถูกใช้จนเกือบหมดแล้ว ปริมาณแบนด์วิดท์ที่เหลืออยู่อาจจะไม่เพียงพอก็ได้ นอกจากนี้การลดอัตราการส่งข้อมูลของโปรแกรมประยุกต์พหุสื่อเพียงอย่างเดียวอาจไม่สามารถทำให้เครือข่ายลดความคับคั่งลงได้ เนื่องจากว่าผู้ใช้งานเครือข่ายอินเตอร์เน็ตจำนวนมากมีการใช้โปรแกรมประยุกต์ที่ใช้

โปรโตคอล TCP อย่างเช่น WWW, FTP, Telnet เป็นต้น ซึ่งโปรโตคอล TCP จะมีพฤติกรรมที่ไม่อยู่นิ่งเช่นกัน โดยจะพยายามปรับอัตราการส่งข้อมูลให้เหมาะสมกับแบนด์วิดท์ของเครือข่ายที่เหลืออยู่ ดังนั้นหากโปรแกรมประยุกต์พหุสื่อมีการลดอัตราการส่งข้อมูล แบนด์วิดท์ส่วนที่เหลือก็จะถูกโปรแกรมประยุกต์ TCP ดึงไปใช้เช่นกัน จะเห็นว่าความคับคั่งของเครือข่ายก็ไม่ได้ลดน้อยลงไป นอกจากนี้จุดอ่อนของเทคนิคการปรับตัวประเภทนี้ คือ การใช้อัตราการสูญหายของข้อมูลนั้นสามารถบอกระดับความคับคั่งของเครือข่ายได้ แต่บางครั้งก็ไม่อาจบอกถึงคุณภาพของสื่อที่ได้รับได้ โดยเฉพาะในกรณีของการสื่อสารวีดิทัศน์ วีดิทัศน์ 1 เฟรมอาจถูกตัดแบ่งเพื่อบรรจุลงในหลายแพ็กเก็ต เพื่อไม่ให้ขนาดของแพ็กเก็ตใหญ่กว่าค่า Maximum Transfer Unit (MTU) ของลิงค์ ปริมาณการสูญหายของแพ็กเก็ตไม่สามารถบอกได้ว่ามีเฟรมวีดิทัศน์เสียหายกี่เฟรม ตัวอย่างเช่น หากมีแพ็กเก็ตสูญหายจำนวน 4 แพ็กเก็ต ถ้าหากทั้ง 4 แพ็กเก็ตได้มาจากวีดิทัศน์ในเฟรมเดียวกันจะทำให้มีเฟรมวีดิทัศน์สูญหายเพียงหนึ่งเฟรมเท่านั้น แต่ถ้าหากทั้ง 4 แพ็กเก็ตนี้ได้มาจากวีดิทัศน์ในเฟรมที่แตกต่างกันทั้งหมด จะทำให้มีวีดิทัศน์สูญหายทั้งหมด 4 เฟรม

2.1.2 การปรับตัวที่ผู้รับเป็นตัวขับ (Receiver-Driven Adaptation)

ในการปรับตัวของโปรแกรมประยุกต์พหุสื่อวิธีนี้ การปรับตัวส่วนใหญ่จะเกิดขึ้นที่ผู้รับ วิธีการที่ได้รับการเสนอขึ้นมาส่วนใหญ่จะนำเทคนิคการบีบอัดแบบ Layered Encoding เข้ามาช่วย ซึ่งเทคนิคนี้ผู้ส่งจะบีบอัดข้อมูลโดยแยกออกเป็นหลายระดับชั้น โดยระดับชั้นที่อยู่ต่ำสุดหรือระดับชั้นฐาน (Base Layer) เป็นระดับชั้นที่สำคัญที่สุดจะประกอบด้วยข้อมูลที่ทำให้สื่อมีคุณภาพที่ยอมรับได้ในระดับหนึ่ง ส่วนระดับชั้นอื่นที่อยู่ในระดับเหนือขึ้นมาเป็นข้อมูลที่ช่วยในการเพิ่มคุณภาพของสื่อให้ดีขึ้น และเมื่อผู้ส่งนำสื่อมาบีบอัดและแยกเป็นระดับชั้นแล้ว ข้อมูลของแต่ละระดับชั้นก็จะส่งไปยังกลุ่มมัลติคาสต์ (Multicast Group) ที่แตกต่างกันออกไป จากนั้นผู้รับจะเป็นผู้ตัดสินใจเองว่าจะเข้ากลุ่มมัลติคาสต์ (Join Multicast) เพื่อรับข้อมูลของระดับชั้นไหนบ้าง โดยพิจารณาจากระดับความคับคั่งของเครือข่าย เช่น หากเครือข่ายมีความคับคั่งสูงผู้รับก็จะลดจำนวนระดับชั้นลง แต่ถ้าเครือข่ายมีความคับคั่งต่ำผู้รับก็สามารถเพิ่มจำนวนระดับชั้นได้

เทคนิคการปรับตัวที่ผู้รับนี้สามารถแบ่งได้อีกเป็น 2 แบบ คือ แบบที่ผู้ส่งไม่มีปรับอัตราบิตในการบีบอัดแต่ละระดับชั้น ซึ่งได้แก่ เทคนิค Receiver-Driven Layered Multicast (RLM)[13] ซึ่งเสนอโดย McCanne, Jacobson และ Vetterli และเทคนิค Thin Streams[14] ซึ่งเสนอโดย Wu, Sharma และ Smith ส่วนแบบที่สองนั้นผู้ส่งจะมีการปรับอัตราบิตของการบีบอัดแต่ละระดับชั้นโดยพิจารณาจากข้อมูลป้อนกลับจากผู้รับ ตัวอย่างของวิธีการปรับตัวแบบนี้ได้แก่ เทคนิค Adaptive Layered Transmission (ALT)[12] ซึ่งเสนอโดย Sisalem และ Emanuel

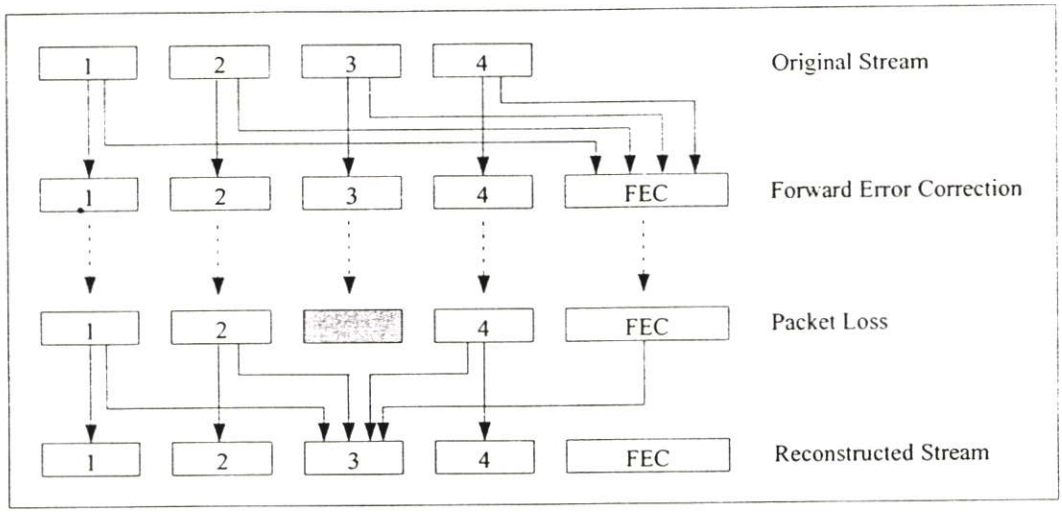
เทคนิคการปรับตัวแบบนี้เหมาะสำหรับกรณีที่มีผู้รับเป็นจำนวนมาก เนื่องจากสามารถรองรับการขยายระบบได้ แต่ในกรณีของการสื่อสารแบบจุดต่อจุด (Point-to-Point) ซึ่งมีผู้รับคน

เดียวก็ไม่มีความจำเป็นที่จะต้องส่งข้อมูลในแต่ละระดับชั้นแยกกัน

2.1.3 การควบคุมความผิดพลาดแบบปรับตัว (Adaptive Error Control)

เนื่องจากว่าการสื่อสารพหุสื่อบนเครือข่ายอินเตอร์เน็ตนั้นไม่สามารถที่หลีกเลี่ยงปัญหาการสูญหายของแพ็กเก็ตได้ ดังนั้นจึงจำเป็นที่จะต้องมีกลไกที่จะช่วยลดผลกระทบจากการสูญหายแพ็กเก็ต Forward Error Correction (FEC) เป็นวิธีการหนึ่งที่สามารถลดผลกระทบจากการสูญหายของแพ็กเก็ตได้โดยไม่เพิ่มค่าเวลาหน่วงมากนัก ซึ่งคุณสมบัตินี้เหมาะสมสำหรับการสื่อสารที่มีลักษณะเป็นเวลาจริง

FEC สามารถแบ่งออกได้เป็น 2 ประเภทคือ FEC ที่ไม่ขึ้นกับสื่อ (Media Independent FEC) และ FEC ที่ขึ้นกับสื่อ (Media Specific FEC) โดยประเภทแรกคือ FEC ที่ไม่ขึ้นกับสื่อนั้นมีวิธีการคือ เมื่อมีการส่งแพ็กเก็ตได้หนึ่งชุด จะมีการส่งแพ็กเก็ตพิเศษเพิ่มเข้าไปดังรูปที่ 2.2 โดยแพ็กเก็ตพิเศษนี้ได้จากการกระทำทางคณิตศาสตร์ของข้อมูลของทุกแพ็กเก็ตที่อยู่ในชุดเดียวกัน ดังนั้นเมื่อมีแพ็กเก็ตใดแพ็กเก็ตหนึ่งสูญหายไป ก็สามารถกู้คืนได้โดยใช้แพ็กเก็ตพิเศษนี้ แต่วิธีการนี้ใช้ไม่ได้ผลถ้ามีการสูญหายของแพ็กเก็ตในชุดเดียวกันมากกว่า 1 แพ็กเก็ต[15]

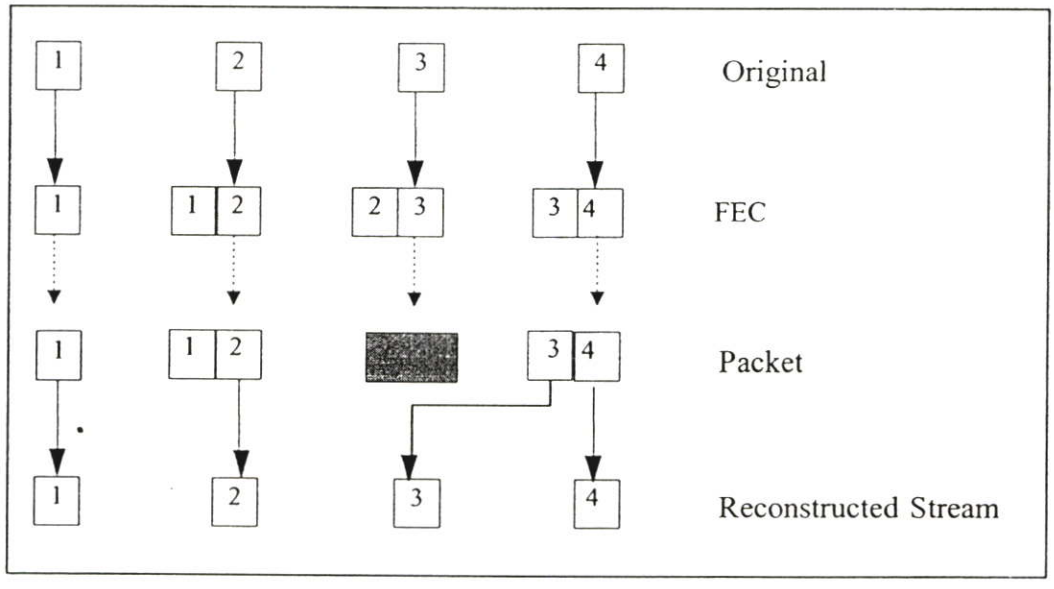


รูปที่ 2.2 Media Independent FEC

FEC ประเภทที่สองคือ FEC ที่ขึ้นกับสื่อซึ่งมีหลักการคือ ในแต่ละแพ็กเก็ตจะบรรจุข้อมูลซ้ำของแพ็กเก็ตที่อยู่ก่อนหน้า แต่ใช้การบีบอัดด้วยอัตราบิตที่ต่ำกว่าเพื่อไม่ให้ปริมาณการใช้แบนด์วิดท์เพิ่มขึ้นมากเกินไป จากรูปที่ 2.3 จะเห็นว่าเมื่อแพ็กเก็ตที่ 3 สูญหาย ข้อมูลเสียงที่อยู่ในแพ็กเก็ตนี้สามารถกู้คืนได้จากข้อมูลซ้ำที่อยู่ในแพ็กเก็ตที่ 4 ซึ่ง FEC ประเภทที่สองนี้เป็นวิธีการที่ได้รับความนิยมมากว่า FEC ประเภทแรก โดยได้มีการพัฒนาอัลกอริทึมในการควบคุมระดับความ

ผิดพลาดโดยใช้พื้นฐานจาก FEC ที่ขึ้นกับสื่อเป็นจำนวนมาก โดยเฉพาะในกรณีของการสื่อสารเสียงได้มีมาตรฐานในการกำหนดรูปแบบของการบรรจุข้อมูลซ้ำลงในแพ็กเก็ต RTP อีกด้วย ซึ่งมาตรฐานดังกล่าวก็คือ RFC 2198 [16]

จะเห็นว่าในการใช้ FEC ที่ขึ้นกับสื่อนั้น หากมีการใช้ปริมาณข้อมูลซ้ำ (Redundancy) มากขึ้นก็จะช่วยลดผลกระทบจากการสูญหายของแพ็กเก็ตได้ดีขึ้น แต่ผลที่ตามมาก็คือปริมาณการใช้แบนด์วิดท์จะสูงขึ้นเช่นกัน ดังนั้นจึงจำเป็นที่จะต้องมีกลไกในการกำหนดปริมาณข้อมูลซ้ำให้เหมาะสมกับสภาพเครือข่ายที่กำลังใช้งาน เพื่อไม่ให้มีการใช้ปริมาณข้อมูลที่ซ้ำที่มากเกินไปโดยเปล่าประโยชน์ หรือน้อยเกินไปจนไม่สามารถลดผลกระทบจากการสูญหายของแพ็กเก็ตได้ ซึ่งได้มีบทความจำนวนมากที่เสนออัลกอริทึมสำหรับการทำงานในส่วนนี้ บทความ[17], [18] และ [19] ได้เสนออัลกอริทึมในการควบคุมระดับความผิดพลาดของการสื่อสารเสียง ส่วน[20]ได้เสนออัลกอริทึมในการควบคุมระดับความผิดพลาดของการสื่อสารวีดิทัศน์ โดยทุกบทความนั้นมีส่วนที่เหมือนกันก็คือ อัลกอริทึมที่เสนอขึ้นใหม่นั้นล้วนมีพื้นฐานจาก FEC และจะมีการกำหนดรูปแบบการบรรจุข้อมูลมูลซ้ำลงในแพ็กเก็ตหรือที่เรียกว่า Combination เอาไว้ก่อน และจัดลำดับของแต่ละ Combination เอาไว้ โดย Combination ในลำดับหลังสามารถลดผลกระทบจากการสูญหายของแพ็กเก็ตได้ดีกว่า Combination ที่อยู่ลำดับแรกๆ แต่ต้องใช้ปริมาณข้อมูลซ้ำที่มากขึ้นด้วย โดยพารามิเตอร์ที่ใช้ในการตัดสินใจว่าควรจะใช้ Combination ใดก็คือปริมาณการสูญหายของข้อมูล สำหรับส่วนที่แตกต่างกันของแต่ละบทความก็คืออัลกอริทึมที่ใช้ในการเลือก Combination เมื่อใดควรที่จะเพิ่ม Combination และเมื่อใดควรที่จะลด



รูปที่ 2.3 Media Specific FEC

จากเทคนิคการปรับตัวทั้ง 3 ประเภทที่ได้กล่าวมาแล้วนั้น แต่ละประเภทก็มีข้อดีและข้อเสียแตกต่างกันไป การที่จะเลือกว่าจะใช้เทคนิคใดนั้นจำเป็นจะต้องดูจากสภาพแวดล้อมในขณะใช้งาน เทคนิคแรกที่สามารถตัดออกได้ก่อนก็คือ เทคนิคการปรับตัวที่ผู้รับเป็นตัวขับ เนื่องจากว่าเทคนิคนี้เหมาะสำหรับกรณีที่มีผู้รับจำนวนมาก แต่ในวิทยานิพนธ์นี้เน้นที่การสื่อสารแบบจุดต่อจุดและมีการส่งข้อมูลแบบยูนิคาสต์ (Unicast) เท่านั้น ส่วนเทคนิคการปรับอัตราการส่งข้อมูลนั้นถึงแม้ว่าจะช่วยควบคุมระดับความคับคั่งของเครือข่ายได้ดี แต่ก็ไม่สามารถควบคุมคุณภาพของสื่อได้เสมอไป เทคนิคที่เหมาะสำหรับการควบคุมคุณภาพของสื่อมากที่สุดก็คือเทคนิคการควบคุมความผิดพลาด เนื่องจากเทคนิคการปรับตัวประเภทนี้จะช่วยลดผลกระทบจากการสูญหายของแพ็กเก็ตได้ แต่เนื่องจากว่าอัลกอริทึมของการปรับตัวที่จัดอยู่ในประเภทนี้มีเป็นจำนวนมาก และแต่ละอัลกอริทึมก็ใช้ได้ผลในเงื่อนไขที่แตกต่างกัน วิทยานิพนธ์นี้จึงได้การศึกษาและปรับปรุงจากอัลกอริทึมที่มีอยู่แล้ว โดยเพิ่มเทคนิคการปรับให้เรียบแบบเอกซโพเนนเชียลครั้งเดียวเพื่อให้กำหนดพารามิเตอร์ที่เหมาะสมกับการสื่อสารเสียงในกรณีที่ใช้การบีบอัดแบบ G.723.1

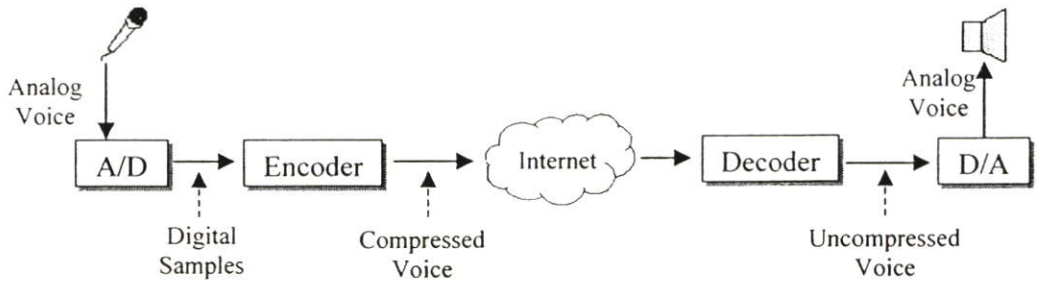
2.2 การสื่อสารเสียง

หลักการพื้นฐานของการสื่อสารเสียงที่จะกล่าวถึงในหัวข้อนี้หมายถึงการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต ซึ่งจำเป็นจะต้องมีการแปลงสัญญาณเสียงจากอนาล็อก (Analog) ให้เป็นดิจิตอล (Digital) รวมทั้งต้องมีการลดขนาดของข้อมูลเสียงก่อนจะบรรจุลงในแพ็กเก็ตและส่งให้ผู้รับ และนอกจากนี้ผู้รับจะต้องเข้าใจข้อมูลที่ได้รับจากผู้ส่ง มิฉะนั้นผู้รับก็จะไม่สามารถนำข้อมูลไปประมวลผลได้ถูกต้อง ดังนั้นทั้งผู้รับและผู้ส่งจึงต้องใช้มาตรฐานในการสื่อสารเดียวกัน

2.2.1 ขั้นตอนในการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต

ในการส่งเสียงพูดบนเครือข่ายอินเตอร์เน็ต ผู้ส่งจะต้องเปลี่ยนสัญญาณเสียงจากไมโครโฟนที่อยู่ในรูปแบบของสัญญาณอนาล็อกให้เป็นดิจิตอลเสียก่อน โดยใช้ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter, A/D) ดังแสดงในรูปที่ 2.4 โดยปกติแล้วเสียงพูดของมนุษย์จะมีความถี่ไม่เกิน 4 กิโลเฮิรตซ์ (kHz) [2] ดังนั้นอัตราการชักตัวอย่าง (Sampling Rate) ตามทฤษฎีของไนควิซต์ก็จะมีค่าเท่ากับ 8 กิโลเฮิรตซ์ (2 เท่าของความถี่สูงสุด) ซึ่งในระบบโทรศัพท์สาธารณะ (Public Switched Telephone Network, PSTN) ก็ใช้อัตราการชักตัวอย่างเท่ากับค่านี้เช่นกัน โดยในระบบโทรศัพท์ PSTN นั้นใช้วิธีการที่เรียกว่า Pulse Code Modulation (PCM) ในการแปลงระดับสัญญาณเสียงแบบอนาล็อกให้กลายเป็นค่าดิจิตอลที่เรียกว่า ค่าตัวอย่าง (Sample) โดยในกรณีที่ใช้คอมพิวเตอร์เป็นเครื่องมือในการสื่อสารเสียงนั้น หน้าที่นี้เป็นของการ์ดเสียง เมื่อได้ข้อมูลเสียงในรูปแบบของดิจิตอลแล้วก็สามารถส่งข้อมูลเสียงผ่านเครือข่ายอินเตอร์เน็ตไปให้กับผู้รับได้ แต่เพื่อไม่ให้ปริมาณข้อมูลเสียงที่ส่งผ่านเครือข่ายมีมากเกินไปจึงต้องมีการบีบอัดข้อมูล

เสียงเสียก่อน (Audio Compression) โดยผู้ส่งจะใช้ตัวเข้ารหัส (Encoder) ที่มีวิธีการเข้ารหัสตามมาตรฐานการบีบอัดที่เลือกใช้ ซึ่งรายละเอียดเกี่ยวกับการบีบอัดเสียงจะได้กล่าวถึงในหัวข้อถัดไป ส่วนทางฝั่งของผู้รับก็จะต้องนำแพ็กเก็ตเสียงที่ได้รับมาถอดรหัส จากนั้นก็แปลงสัญญาณสียงจากดิจิตอลให้กลายเป็นอนาล็อกและส่งสัญญาณเสียงไปเล่น (Play) ออกทางลำโพง



รูปที่ 2.4 การส่งเสียงพูดในรูปแบบดิจิตอลบนเครือข่ายอินเตอร์เน็ต[2]

2.2.2 การบีบอัดเสียง (Audio Compression)

ในการส่งเสียงพูดผ่านเครือข่ายอินเตอร์เน็ตนั้น เพียงแต่การแปลงจากสัญญาณเสียงแบบอนาล็อกให้กลายเป็นข้อมูลเสียงแบบดิจิตอลนั้นยังไม่เพียงพอ สังเกตว่าในการแปลงสัญญาณเสียงจากอนาล็อกเป็นดิจิตอลซึ่งใช้อัตราการชักตัวอย่าง 8000 เฮิรตซ์ (ใน 1 วินาทีมีการชักตัวอย่าง 8000 ครั้ง) ถ้าหากใช้ข้อมูลดิจิตอล 8 บิตต่อหนึ่งค่าตัวอย่าง แบนด์วิดท์ที่ใช้ในการส่งข้อมูลก็จะเป็น 64 กิโลบิตต่อวินาที (kbps) และถ้าใช้ 16 บิตต่อค่าตัวอย่าง แบนด์วิดท์ก็เพิ่มเป็น 128 กิโลบิตต่อวินาที ซึ่งถือว่าเป็นค่าที่สูงทีเดียว ยิ่งไปกว่านั้นถ้าหากมีผู้ที่ต้องการสื่อสารเสียงพร้อมกันหลายคู่สนทนา เครือข่ายที่มีแบนด์วิดท์อย่างจำกัดก็จะไม่สามารถรองรับได้ ผลที่ตามมาก็คือผู้รับก็จะได้ยินเสียงไม่ชัดเจน เสียงไม่ต่อเนื่อง หรืออาจจะฟังไม่รู้เรื่องเลยก็ได้ ซึ่งวิธีแก้ไขก็คือต้องมีการลดขนาดของข้อมูลเสียงก่อนที่จะส่งข้อมูลผ่านเครือข่าย เทคนิคนี้เรียกว่า การบีบอัดเสียง

ในการบีบอัดเสียงนั้น วิธีการถอดรหัสจะต้องสอดคล้องกับวิธีการเข้ารหัส มิฉะนั้นจะไม่สามารถนำมาเสียงมาเล่นออกทางลำโพงได้ถูกต้อง สหภาพโทรคมนาคมนานาชาติ (International Telecommunication Union, ITU) ได้มีการกำหนดมาตรฐานสำหรับการบีบอัดเสียง (Audio Codec) เอาไว้หลายมาตรฐาน ตารางที่ 2.1 เป็นการเปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดเสียงแต่ประเภท โดยในคอลัมน์ของ Quality MOS (Mean Opinion Score)[20] เป็นค่าที่แสดงคุณภาพของเสียง ค่าสูงสุดมีค่าเท่ากับ 5 จะเห็นว่า G.711[25] นั้นมีคุณภาพสูงที่สุดก็จริง แต่ก็มีอัตราบิต (Bit Rate) สูงถึง 64 kbps ทำให้ไม่สามารถนำมาใช้งานได้จริงถ้าเครือข่ายมีแบนด์วิดท์ต่ำกว่านี้ เช่น ในการเชื่อมต่ออินเตอร์เน็ตผ่านทางเครือข่ายโทรศัพท์ซึ่งมีความเร็วอยู่ที่ 28.8 ถึง 56 กิโลบิตต่อวินาทีเท่านั้น ส่วน G.729[26] และ G.723.1 ก็มีคุณภาพใกล้เคียงกับ G.711 แต่มีอัตราบิตน้อยกว่า มาตรฐานทั้งสองจึงน่าจะเป็นทางเลือกที่ดีมาตรฐานอื่น

ตารางที่ 2.1 เปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดข้อมูลเสียง[2]

Codec	Bit Rate (kbps)	Complexity compared with G.726	Algorithmic Delay	Quality MOS
G.711	64	very low	0.125	4.0
G.723.1	5.3*	8	37.5	3.9
G.723.1	6.3*	8	37.5	3.9
G.726	32	1	0.125	3.85
G.728	16	15	0.625	3.61
G.729	8	10	15	3.9
G.729A	8	6	15	3.7

*G.723.1 มี 2 อัลกอริทึม ซึ่งมีอัตราบิตต่างกันคือ 5.3 และ 6.3 กิโลบิตต่อวินาที

2.2.3 การบีบอัดเสียงด้วย G.723.1

เนื่องจากในวิทยานิพนธ์นี้ได้เลือกใช้ G.723.1[1] ในการบีบอัดเสียง ดังนั้นในหัวข้อนี้จึงได้อธิบายถึงรายละเอียดของการบีบอัดเสียงชนิดนี้ G.723.1 มีอัลกอริทึมในการบีบอัดเสียงอยู่ 2 อัลกอริทึม อัลกอริทึมแรกคือ Multi-Pulse Maximum Likelihood Quantization (MP-MLQ) ซึ่งมีการบีบอัดข้อมูลเสียงโดยแบ่งเป็นเฟรมซึ่งเทียบเท่ากับเสียงพูด 30 มิลลิวินาที (ms) หรือ 240 ค่าตัวอย่าง โดยใช้ค่าตัวอย่างขนาด 16 บิต ผลจากการบีบอัดจะได้ข้อมูลที่มีขนาดลดลงเหลือ 24 ไบต์ต่อเฟรมเท่านั้น อัลกอริทึมนี้มีอัตราบิตเท่ากับ 6.4 กิโลบิตต่อวินาที แต่ในมาตรฐาน G.723.1 ได้ระบุว่าการบีบอัดโดยใช้อัลกอริทึมนี้มีอัตราบิต 6.3 กิโลบิตต่อวินาที เนื่องจากไม่ได้คิดในส่วนของบิตที่เป็นเฮดเดอร์ (Header) ส่วนอัลกอริทึมที่สองมีชื่อว่า Algebraic Code-Excited Linear Prediction (ACELP) ซึ่งบีบอัดข้อมูลเสียงจาก 240 ค่าตัวอย่าง (ค่าตัวอย่างมีขนาด 16 บิต) ได้เป็นข้อมูลขนาด 20 ไบต์ ทำให้อัลกอริทึมนี้มีอัตราบิตเป็น 5.3 กิโลบิตต่อวินาที สำหรับอัลกอริทึมที่ใช้ในวิทยานิพนธ์นี้คือ MP-MLQ

2.2.4 ปัจจัยที่มีผลกระทบต่อคุณภาพของเสียง

เนื่องจากเครือข่ายอินเตอร์เน็ตไม่ได้มีการรับประกันคุณภาพการบริการ ดังนั้นการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ตจึงมีโอกาสที่คุณภาพของเสียงที่ผู้รับได้รับจะต่ำกว่าคุณภาพของเสียงต้นฉบับได้ ซึ่งปัจจัยที่มีผลกระทบต่อคุณภาพของเสียงมีดังนี้

2.2.4.1 เวลาหน่วง

เวลาหน่วง คือ เวลาในการเดินทางของเสียงจากไมโครโฟนของผู้ส่งไปถึงลำโพงของผู้รับ ในระบบโทรศัพท์ PSTN จะไม่เกิดปัญหานี้เนื่องจากในการโทรแต่ละครั้ง จะมีการจองทรัพยากรสำหรับแต่ละคู่สายเอาไว้และค่าเวลาหน่วงจะน้อยมาก ประมาณ 50 – 70 มิลลิวินาที [28] ในขณะที่เครือข่ายอินเตอร์เน็ตไม่สามารถที่จะรับประกันได้ว่าค่าเวลาหน่วงเป็นเท่าใด โดยปกติคนเราจะรู้สึกถึงผลกระทบของเวลาหน่วงเมื่อค่าเวลาหน่วงสูงกว่า 250 มิลลิวินาที [27]

2.2.4.2 จิตเตอร์ (Jitter)

จิตเตอร์เกิดจากการที่เวลาหน่วงของแต่ละแพ็กเก็ตไม่เท่ากัน ทั้งนี้เนื่องจากในการเดินทางของแพ็กเก็ตบนเครือข่ายอินเตอร์เน็ตนั้น แต่ละแพ็กเก็ตอาจจะใช้เส้นทางที่ต่างกันได้แม้จะมีปลายทางที่เดียวกัน รวมถึงสภาพความคับคั่งของเครือข่ายที่เปลี่ยนแปลงได้ตลอดเวลาก็สามารถทำให้เกิดจิตเตอร์ได้เช่นกัน ปัญหานี้จึงทำให้สัญญาณเสียงที่ได้รับมีอาการกระตุกและฟังได้ไม่ชัดเจน ดังนั้นจึงต้องมีการใช้บัฟเฟอร์เพื่อรวบรวมข้อมูลก่อนเพื่อให้ข้อมูลมีความต่อเนื่อง แล้วจึงส่งไปประมวลผลและเล่นออกทางลำโพงต่อไป การใช้บัฟเฟอร์ขนาดใหญ่จะสามารถกำจัดจิตเตอร์ได้ดีขึ้น แต่ก็มีผลทำให้มีเวลาหน่วงมากขึ้น แต่ถ้าใช้บัฟเฟอร์ขนาดเล็กเกินไปจะทำให้กำจัดจิตเตอร์ได้ไม่ดีนัก ดังนั้นจึงจำเป็นต้องหาวิธีที่ใช้ในการปรับขนาดของบัฟเฟอร์ ให้เหมาะสม

2.2.4.3 การสูญหายของแพ็กเก็ต (Packet Loss)

การที่แพ็กเก็ตสูญหายนั้นส่วนใหญ่มีสาเหตุมาจากความคับคั่งของเครือข่าย เพราะในขณะที่เราเตอร์ทำการประมวลผลแพ็กเกตหนึ่งอยู่ ถ้ามีอีกแพ็กเกตเข้ามาจะถูกเก็บไว้ในบัฟเฟอร์ของคิว และถ้าบัฟเฟอร์นั้นเต็มจะทำให้แพ็กเกตที่เข้ามาทีหลังถูกตัดทิ้ง ถึงแม้ว่าในการสื่อสารเสียงจะยอมให้มีการสูญหายของแพ็กเก็ตได้บ้าง แต่การที่มีแพ็กเก็ตสูญหายมากเกินไปก็อาจส่งผลกระทบต่อคุณภาพเสียงได้ โดยการสูญหายของแพ็กเก็ตจะมีผลกระทบอย่างชัดเจนเมื่อมีอัตราการสูญหายมากกว่า 5% [27][28]

2.2.4.4 แบนด์วิดท์ที่มีจำกัด (Limited bandwidth)

แบนด์วิดท์เป็นส่วนที่สำคัญมากในการสื่อสารเสียง ซึ่งปริมาณการใช้แบนด์วิดท์นั้นก็จะขึ้นอยู่กับอัลกอริทึมในการบีบอัดข้อมูลเสียง ถ้าแบนด์วิดท์ที่ใช้ในขณะนั้นไม่เพียงพอจะทำให้เกิดปัญหาต่างๆ ตามมาไม่ว่าจะเป็นเวลาหน่วง จิตเตอร์ และสูญหายของแพ็กเก็ต

2.2.5 โปรโตคอลที่ใช้ในการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ต

โปรโตคอลในระดับชั้นทรานสปอร์ต (Transport Layer) ของเครือข่ายอินเตอร์เน็ตที่มีการ

ใช้งานกันอย่างแพร่หลายในปัจจุบันมีอยู่สองโปรโตคอลคือ TCP ซึ่งเป็นโปรโตคอลที่มีการรับประกันว่าสามารถส่งข้อมูลไปถึงปลายทางได้อย่างถูกต้องและครบถ้วน อีกโปรโตคอลหนึ่งก็คือ User Datagram Protocol (UDP) ซึ่งเป็นโปรโตคอลที่ไม่มีการรับประกันความน่าเชื่อถือของข้อมูล จะเห็นว่า TCP เป็นโปรโตคอลที่มีประสิทธิภาพมากกว่าและกลไกการทำงานซับซ้อนมากกว่า UDP แต่ TCP นั้นไม่เหมาะสำหรับการสื่อสารเสียง ทั้งนี้เนื่องจากความล่าช้าของกลไกการทำงานของ TCP นั่นเอง การสื่อสารเสียงนั้นต้องการความเร็ว โดยไม่จำเป็นว่าจะต้องส่งแพ็กเก็ตเสียงให้ไปถึงผู้รับอย่างครบถ้วน เพราะถึงแม้ว่าจะมีข้อมูลบางส่วนสูญหายไปบ้าง ก็ไม่ได้ส่งผลกระทบต่อคุณภาพของเสียงมากนัก ดังนั้น UDP จึงเป็นทางเลือกที่ดีกว่า แต่การที่จะใช้ UDP อย่างเดียวนั้นยังไม่เพียงพอ จึงได้มีการพัฒนาโปรโตคอลตัวหนึ่งที่ทำงานอยู่เหนือโปรโตคอล UDP โปรโตคอลตัวนี้มีชื่อว่า Real-time Transport Protocol (RTP) ซึ่งองค์กรที่เสนอโปรโตคอลตัวนี้ขึ้นมาคือ Internet Engineering Task Force (IETF) โปรโตคอล RTP ได้รับการพัฒนาขึ้นมาเพื่อใช้ในการสื่อสารแบบเวลาจริง ซึ่งรายละเอียดของของโปรโตคอลนี้จะกล่าวถึงอีกครั้งในหัวข้อที่ 2.4

2.3 เทคนิคการทำให้เรียบแบบเอกซโพเนนเชียล

ใช้ในการศึกษาวิธีการพยากรณ์อนุกรมเวลาเมื่อมีข้อมูลผิดปกติเช่นอนุกรมเวลาอัตราการสูญหายของข้อมูลซึ่งมีความแตกต่างกันในลักษณะของระดับค่าเฉลี่ยและความแปรปรวนขึ้นอยู่กับสภาพและสถานะของเครือข่าย เทคนิคการปรับให้เรียบจะสามารถติดตามการเปลี่ยนแปลงและทำการปรับค่าพารามิเตอร์ที่ใช้ในการพยากรณ์ครั้งถัด(ใช้ในอัลกอริทึมฝั่งผู้ส่ง) ไปอย่างสม่ำเสมอโดยอัตโนมัติ ซึ่งในวิทยานิพนธ์นี้นำวิธีการทำให้เรียบแบบเอกซโพเนนเชียลครั้งเดียว (Single Exponential Smoothing) มาใช้ในการวิเคราะห์รูปแบบข้อมูลซ้ำของอัลกอริทึมทางฝั่งผู้ส่งก่อนที่จะส่งผ่านเครือข่าย เนื่องจากมีความสามารถในการปรับตัวตามการเปลี่ยนแปลงของอนุกรมเวลาอัตราการสูญหายของข้อมูลได้ดี อนุกรมเวลาที่มีการเปลี่ยนแปลงค่าเฉลี่ยของอนุกรมเวลาน้อยหรือไม่เปลี่ยนแปลงแสดงดัง สมการ (2.1)

$$Y_t = a + \varepsilon_t \qquad , t = 1,2,...,T \qquad (2.1)$$

โดยที่ a คือ ระดับค่าเฉลี่ยของข้อมูล ซึ่งมีค่าแปรเปลี่ยน ช้า ๆ ตามเวลา

ε_t คือ ความคลาดเคลื่อนสุ่มที่เวลา t

การพยากรณ์โดยใช้เทคนิคการทำให้เรียบแบบเอกซโพเนนเชียลครั้งเดียว จะมีการกำหนดน้ำหนัก (weights) ให้กับข้อมูลหรือค่าสังเกตในอดีตไม่เท่ากัน โดยจะให้น้ำหนักกับค่าสังเกตปัจจุบันมากกว่าค่าสังเกตที่ผ่านมา และวิธีการที่ใช้กำหนดน้ำหนัก คือ จากน้ำหนักทั้งหมด 1 จะกระจายให้กับค่าสังเกตในลักษณะที่ลดลงแบบเรขาคณิตหรือแบบเอกซโพเนนเชียลโดยเริ่มจากค่าสังเกต

ปัจจุบันจะมีมากที่สุดและลดลงเรื่อย ๆ สำหรับค่าสังเกตในอดีต

การหาค่าพยากรณ์สำหรับค่าสังเกตในอนาคตที่เวลา $T+\tau$ จะทำได้โดยประมาณค่า a โดยวิธีค่าต่ำสุดของผลรวมของกำลังสองของความคลาดเคลื่อนแบบลดลง (minimize discounted square error) คือหาค่าต่ำสุดของ $e(a)$ เท่ากับหาค่าต่ำสุดของ $\sum_{t=1}^{T} \beta^{T-t} (Y_t - a)^2$ เมื่อ $0 < \beta < 1$ โดยที่ β^{T-t} คือน้ำหนักที่กำหนดให้กับความคลาดเคลื่อนกำลังสอง ณ เวลาที่ t โดยจะมีค่าลดลงแบบเรขาคณิต เมื่อ t มากขึ้น การประมาณค่า a เมื่อสิ้นเวลา T จะคำนวณได้จากสมการ

$$\left.\frac{\partial e(a)}{\partial a}\right|_{\hat{a}(T)} = -2\sum_{t=1}^{T} \beta^{T-t} (Y_t - \hat{a}(T)) = 0 \qquad (2.2)$$

$$\hat{a}(T)\sum_{t=1}^{T} \beta^{T-t} = \sum_{t=1}^{T} \beta^{T-t} Y_t \qquad (2.3)$$

$$\hat{a}(T) = \frac{(1-\beta)}{(1-\beta^T)} \sum_{t=1}^{T} \beta^{T-t} Y_t \qquad (2.4)$$

หรือ

$$\hat{a}(T) = \frac{(1-\beta)Y_t + \beta(1-\beta^{T-1})\hat{a}(T-1)}{(1-\beta^T)} \qquad (2.5)$$

เมื่อ T มีขนาดใหญ่จะได้ว่า $\beta^T \approx 0$ ดังนั้นสมการ (2.5) เขียนใหม่ได้เป็น

$$\hat{a}(T) = (1-\beta)Y_t + \beta\hat{a}(T-1) \qquad (2.6)$$

ให้ $\alpha = 1-\beta$ และ $S_t = \hat{a}(T)$ จะได้ว่า

$$S_T = \alpha Y_T + (1-\alpha)S_{T-1} \qquad (2.7)$$

S_T คือค่าประมาณของระดับค่าเฉลี่ย a ณ เวลา T

α คือค่าคงที่ปรับให้เรียบ (Smoothing constant) ; $0 < \alpha < 1$

สำหรับการพยากรณ์โดยใช้เทคนิคการทำให้เรียบแบบเอกซโพเนนเชียลครั้งเดียวนี้ใช้สมการ (2.7) โดยค่า S_T จะเป็นค่าพยากรณ์ของข้อมูลในอนาคตทุกค่าดังนี้

$$\hat{Y}_T(\tau) = S_T = \alpha Y_T + (1-\alpha)S_{T-1} \qquad ;\tau = 1,2,... \qquad (2.8)$$

$\hat{Y}_T(\tau)$ คือค่าพยากรณ์ล่วงหน้า τ คาบเวลาจากเวลา $t = T$

จากสมการที่ (2.8) จะพบว่าการคำนวณหาค่า S_T ที่ $T = 1$ จะต้องทราบค่า S_0 หรือค่าเริ่มต้น

ของการทำให้เรียบ (initial smooth value) เพราะว่า $S_1 = \alpha Y_1 + (1-\alpha)S_0$ ดังนั้นจะต้องทำการกำหนดค่าเริ่มต้น S_0 ก่อนจึงทำการพยากรณ์ต่อไปได้ การกำหนดค่าเริ่มต้น S_0 ทำได้หลายวิธีเช่น ใช้ค่าเฉลี่ยเลขคณิตของข้อมูลที่มีอยู่ $Y_1, Y_2, ..., Y_T$ (Brown, 1963, and Montgomory and Johnson, 1976, Quoted in Abraham and Ledolter, 1983) ซึ่งการกำหนดค่าเริ่มต้นด้วยวิธีนี้จะใช้ได้ดีถ้าระดับค่าเฉลี่ย (mean level) ของข้อมูลมีการเปลี่ยนแปลงอย่างช้า ๆ หรือใช้ข้อมูลตัวแรกเป็นตัวเริ่มต้น (Markridakis and Whellwright, 1978) ถ้าระดับค่าเฉลี่ยของข้อมูลมีการเปลี่ยนแปลงอย่างรวดเร็ว สำหรับในการศึกษาครั้งนี้ใช้การกำหนดค่าเริ่มต้นด้วยการใช้ค่าเฉลี่ย นอกจากจะต้องกำหนดค่าเริ่มต้นในการทำให้เรียบแล้ว ยังต้องกำหนดค่าคงที่ปรับให้เรียบ α ซึ่งอาจพิจารณาจากการเปลี่ยนแปลงระดับค่าเฉลี่ยของข้อมูลว่ามีการเปลี่ยนแปลงช้าหรือเร็วอย่างไร ถ้าระดับของข้อมูลมีการเปลี่ยนแปลงอย่างช้า ๆ ค่า α จะมีค่าเข้าใกล้ศูนย์ แต่ถ้าระดับค่าเฉลี่ยของข้อมูลเปลี่ยนแปลงอย่างรวดเร็ว α ควรมีค่าเข้าใกล้ 1 สำหรับการวิจัยครั้งนี้เลือกใช้ α ที่ให้ค่าเฉลี่ยความคลาดเคลื่อนกำลังสองต่ำสุด โดยจะเริ่มค่า α จาก 0.01 ถึง 0.99 ด้วยค่าเพิ่ม 0.01

2.3.1 ค่าคงที่ปรับให้เรียบ

ค่าคงที่ปรับให้เรียบ ที่ปรากฏอยู่ในสมการคำนวณค่าทำนาย(สมการที่2.8) เป็นค่าที่ใช้กำหนดน้ำหนักให้กับค่าสังเกตในปัจจุบัน (ค่าสังเกตเมื่อเวลา T) จากการศึกษาพบว่าค่าคงที่ปรับให้เรียบสามารถกำหนดความเร็วการตอบสนองต่อการเปลี่ยนรูปแบบของอนุกรมเวลา [29] กล่าวคือ ค่าคงที่ปรับให้เรียบที่มีค่าน้อยจะทำให้ค่าทำนายจะปรับตัวตามค่าสังเกตของอนุกรมเวลาที่มีการเปลี่ยนรูปแบบได้ช้าและเกิดความคลาดเคลื่อนสูง เนื่องจากต้องใช้เวลามากในการติดตามรูปแบบของอนุกรมเวลา และเมื่อค่าคงที่ปรับให้เรียบมีค่ามากจะสามารถติดตามการเปลี่ยนรูปแบบของอนุกรมเวลาได้อย่างรวดเร็วแต่จะให้ค่าทำนายที่มีความแปรปรวนสูง ความแปรปรวนที่เกิดขึ้นของค่าทำนายเกิดจากการตอบสนองต่อค่าสังเกตอย่างรวมเร็วซึ่งโดยทั่วไปแล้ว ค่าสังเกตจะมีค่าสุ่มซึ่งเป็นส่วนประกอบหนึ่งของอนุกรมเวลา

ค่าคงที่ปรับให้เรียบสามารถกำหนดความแม่นยำและความผิดพลาดของการทำนายในขณะเดียวกัน ค่าคงที่ปรับให้เรียบที่มีค่าน้อยจะเหมาะสำหรับใช้ทำนายอนุกรมเวลาชนิดคงที่ซึ่งไม่มีการเปลี่ยนรูปแบบของอนุกรมเวลา ส่วนค่าคงที่ปรับให้เรียบที่มีค่ามากเหมาะสมที่จะใช้เพื่อปรับให้ค่าทำนายสามารถติดตามรูปแบบของอนุกรมเวลาที่กำลังเปลี่ยนแปลงได้อย่างรวดเร็ว ในงานวิจัยนี้ใช้ค่าคงที่ปรับให้เรียบตามวิธีของ Trigg กับ Leach แบบจำกับขอบเขตของค่าคงที่ปรับให้เรียบ

2.3.2 การปรับค่าคงที่ปรับให้เรียบตามวิธีของ Trigg และ Leach แบบจำกัดขอบเขต

การปรับค่าคงที่ปรับให้เรียบตามวิธีของ Trigg และ Leach จะให้ค่าคงที่ปรับให้เรียบมีค่า

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อยู่ระหว่าง 0 และ 1 จากการศึกษา [33], [37] พบว่าการกำหนดขอบเขตให้กับค่าคงที่ปรับให้เรียบให้มีช่วงที่แคบลง สามารถลดความคลาดเคลื่อนในการทำนายได้ ในงานวิจัยนี้ได้กำหนดให้ขอบเขตของค่าคงที่ปรับให้เรียบ $\alpha_{max} = 0.98$ และ $\alpha_{min} = 0.2$ และคำนวณค่าทำนายโดยใช้สมการที่ 2.8 เพื่อลดการตอบสนองต่อการเปลี่ยนรูปแบบของอนุกรมเวลาที่เร็วหรือช้าเกินไป

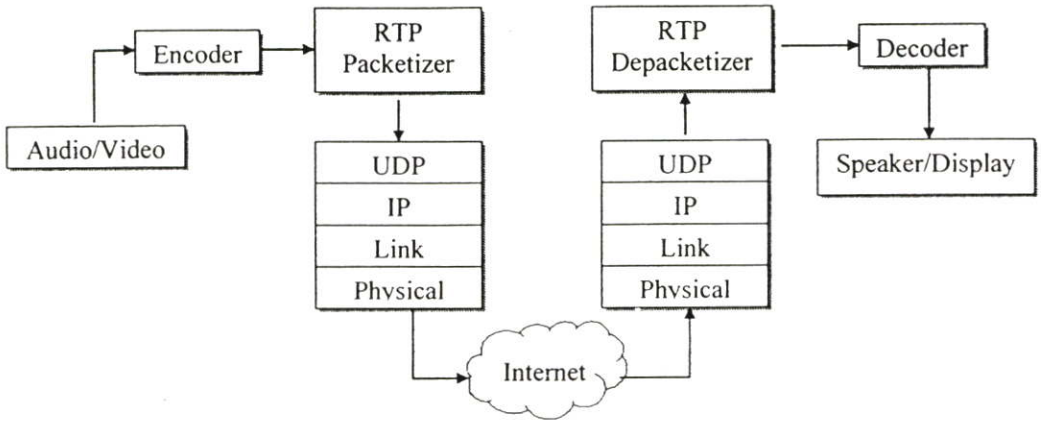
2.4 Real-time Transport Protocol (RTP)

RTP เป็นโปรโตคอลที่ได้รับการพัฒนาโดย IETF และมีข้อกำหนดอยู่ใน RFC 3550 [9] RTP ถูกออกแบบมาเพื่อใช้เป็นโปรโตคอลในการส่งข้อมูลของการสื่อสารที่ทำงานในลักษณะเวลาจริง ในวิทยานิพนธ์นี้จึงเลือกใช้โปรโตคอล RTP ทั้งในการสื่อสารเสียงและการสื่อสารวีดิทัศน์ หน้าที่สำคัญของ RTP คือ ทำให้ทางผู้รับสามารถนำข้อมูลที่ได้จากแพ็กเก็ตไปแสดงผลได้ถูกต้อง กลไกหลักที่ใช้ใน RTP ก็คือ การประทับเวลา (Time Stamping) และการกำหนดหมายเลขลำดับ (Sequence Numbering) ผู้ส่งจะกำหนดค่า Timestamp ซึ่งเป็นฟิลด์หนึ่งในเฮดเดอร์ของ RTP ตามเวลาที่ข้อมูลในไบต์แรกของแพ็กเกตถูกชักตัวอย่าง ทางด้านของผู้รับเมื่อได้รับแพ็กเก็ตจะใช้ค่า Timestamp เพื่อกำหนดจังหวะที่ถูกต้องในการแสดงข้อมูล และยังอาจจะใช้ในการประสานเวลาระหว่างสัญญาณเสียงและวีดิทัศน์ ส่วนกลไกการกำหนดหมายเลขลำดับนั้นใช้ฟิลด์หนึ่งในเฮดเดอร์ RTP ที่ชื่อว่า Sequence Number โดยค่าของหมายเลขนี้จะเพิ่มขึ้นตามจำนวนแพ็กเก็ตที่ส่ง ซึ่งผู้รับจะใช้ข้อมูลนี้ในการเรียงลำดับและตรวจสอบการสูญหายของแพ็กเก็ต นอกจากนี้ RTP ยังให้ข้อมูลเกี่ยวกับวิธีการบีบอัดข้อมูลเสียงหรือวีดิทัศน์ เพื่อให้ผู้รับสามารถแสดงผลข้อมูลได้ถูกต้อง และนอกจากข้อกำหนดเกี่ยวกับการใช้โปรโตคอลในการขนส่งข้อมูลของสื่อแล้ว ใน RFC 3550 ยังมีข้อกำหนดของโปรโตคอลหนึ่งคือ RTP Control Protocol (RTCP) ซึ่งเป็นโปรโตคอลที่ถูกนำมาใช้งานร่วมกับ RTP โดยโปรโตคอล RTCP จะถูกนำไปใช้ในการรายงานเกี่ยวกับสถิติและคุณภาพการบริการ

2.4.1 บทบาทของ RTP

รูปที่ 2.5 แสดงให้เห็นถึงบทบาทของ RTP ในการสื่อสารเสียงและวีดิทัศน์ จะเห็นว่าเมื่อเสียงหรือวีดิทัศน์ได้ผ่านขั้นตอนการเข้ารหัสด้วยวิธีการบีบอัดข้อมูลแล้ว ข้อมูลที่ได้ก็จะมีขนาดลดลง หลังจากนั้นข้อมูลส่วนนี้ก็จะถูกนำมาบรรจุลงในแพ็กเก็ต RTP ซึ่งแต่ละแพ็กเก็ตก็จะมีการระบุด้วยว่าใช้วิธีการบีบอัดข้อมูลแบบใด เป็นแพ็กเก็ตในลำดับที่เท่าไร รวมไปถึงข้อมูลอื่นๆ ที่จะกล่าวถึงในหัวข้อถัดไป ดังนั้นเมื่อข้อมูลนี้เดินทางผ่านเครือข่ายอินเตอร์เน็ตไปถึงฝั่งของผู้รับ ผู้รับก็สามารถใช้ข้อมูลนี้ในการแสดงผลได้ถูกต้อง

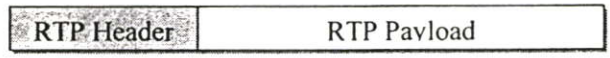
74447



รูปที่ 2.5 บทบาทของ RTP ในการสื่อสารเสียงและวีดิทัศน์บนเครือข่ายอินเตอร์เน็ต

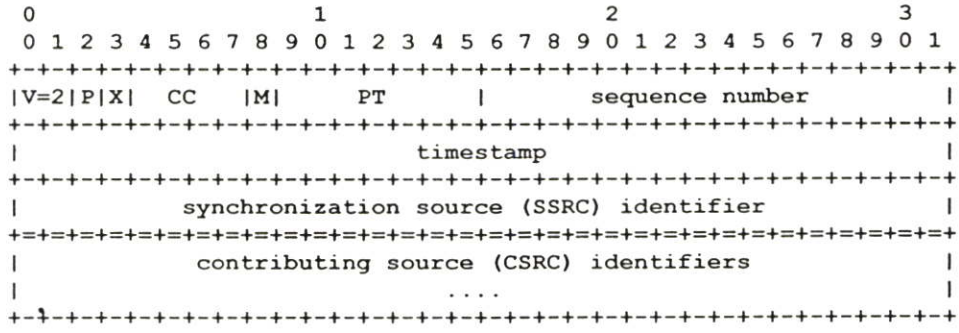
2.4.2 โครงสร้างแพ็กเก็ตของ RTP (RTP Packet Structure)

แพ็กเก็ต RTP แต่ละแพ็กเก็ตนั้นจะมีโครงสร้างดังที่แสดงในรูปที่ 2.6 โดยส่วนแรกของแพ็กเก็ตคือ ส่วนเฮดเดอร์ RTP (RTP Header) เป็นส่วนที่ระบุถึงคุณสมบัติของแต่ละแพ็กเก็ต โดยที่ข้อมูลเสียงหรือวีดิทัศน์ที่ผ่านการเข้ารหัสแล้วจะถูกนำไปวางในตำแหน่งของเพย์โหลด (Payload)



รูปที่ 2.6 โครงสร้างแพ็กเก็ตของ RTP

โดยทั่วไปเฮดเดอร์คือ โครงสร้างข้อมูลของแต่ละแพ็กเกตที่ถูกรวมเข้ากับข้อมูลที่จะส่ง ซึ่งในแต่ละโปรโตคอลจะมีโครงสร้างของเฮดเดอร์ที่แตกต่างกัน เพื่อกำหนดรูปแบบข้อมูลของโปรโตคอลตัวนั้นๆ สำหรับเฮดเดอร์ของ RTP นั้นมีโครงสร้างดังรูปที่ 2.7



รูปที่ 2.7 โครงสร้างเฮดเดอร์ของ RTP [9]

จากโครงสร้างเฮดเดอร์ของ RTP ในรูปที่ 2.7 จะเห็นว่า RTP จะมีเฮดเดอร์อย่างน้อย 12 ไบต์ โดยมีรายละเอียดของฟิลด์ต่างๆ ดังนี้

- V (Version) กำหนดเวอร์ชันของโปรโตคอล RTP ที่ใช้ซึ่งปัจจุบันเป็นเวอร์ชัน 2
- P (Padding) เป็นฟิลด์ที่บอกว่าแพ็กเก็ตนั้นได้ถูกเติมด้วยข้อมูลที่เพิ่มให้พอดีกับขนาดของแพ็กเก็ตหรือไม่ และยังใช้ในกระบวนการการเข้ารหัสข้อมูล (Encryption) อีกด้วย
- X (Extension) เป็นฟิลด์ที่บอกว่าเฮดเดอร์ RTP มีการขยาย (Extension) หรือไม่
- CC (CSRC Count) บอกจำนวนของ Contribution Source Identifier ในแพ็กเกต โดยมี CSRC ได้ตั้งแต่ 0 - 15
- M (Marker) ค่าของบิตนี้จะถูกระบุโดย Profile และ Specification (RFC 3551 [34])
- PT (Payload Type) บอกชนิดของข้อมูลภายในเพย์โหลด ซึ่งก็คือชนิดของการบีบอัดข้อมูลเสียงหรือวีดิทัศน์นั่นเอง ซึ่งใน RFC 3551 ได้มีการกำหนดเอาไว้ว่าใช้หมายเลขใดแทนวิธีการบีบอัดข้อมูลแต่ละแบบ
- Sequence Number บอกลำดับของแพ็กเก็ต และเพื่อให้ผู้รับตรวจสอบการสูญหายของแพ็กเก็ต และสามารถใช้ในการเรียงลำดับแพ็กเกตใหม่ได้
- Timestamp เป็นค่าที่บอกเวลาที่ใช้ในการชักตัวอย่างของข้อมูลในเพย์โหลด ซึ่งนำไปใช้ในกระบวนการคำนวณจิตเตอร์ และค่า Round Trip Time Delay
- SSRC เป็นตัวเลขประจำ Session นั่นคือแพ็กเก็ตที่มาจากผู้ส่งเดียวกัน และใช้ค่า SSRC เท่ากัน แสดงว่าเป็น Session เดียวกัน
- CSRC ฟิลด์นี้จะถูกใช้เมื่อมีการผสมสื่อ (Media Mixing) ซึ่งเป็นกระบวนการที่ใช้เมื่อมีการประชุมแบบหลายจุด (Multipoint Conference)

2.4.3 RTP Control Protocol (RTCP)

RTCP เป็นโปรโตคอลที่ถูกกำหนดอยู่ใน RFC 3550 เช่นเดียวกับ RTP โปรโตคอลตัวนี้ใช้ในการรายงานเกี่ยวกับสถิติและคุณภาพการบริการ RTCP ใช้ช่องสื่อสารแยกกันกับ RTP โดย RTCP นั้นใช้พอร์ตถัดจาก RTP ไปหนึ่งพอร์ต ช่องสื่อสารของ RTCP นั้นใช้สำหรับการรายงานเกี่ยวคุณภาพการบริการ เช่น จำนวนแพ็กเก็ตที่ได้รับ จำนวนแพ็กเก็ตที่สูญหาย จิตเตอร์ เป็นต้น แพ็กเก็ต RTCP นั้นแบ่งออกเป็น 5 ชนิดได้แก่

- SR(Sender Report) สำหรับบอกสถิติเกี่ยวกับการส่งข้อมูลของผู้ส่ง
- RR(Receive Report) สำหรับบอกสถิติเกี่ยวกับการรับข้อมูลของผู้รับ
- SDES (Source description items) เป็นรายละเอียดต่างๆ ของผู้ส่ง
- BYE สำหรับแจ้งถึงการจบการทำงาน
- APP (Application Specific Functions) สำหรับการกำหนดการทำงานบางอย่างในส่วนของโปรแกรมประยุกต์ ซึ่งไม่ได้ระบุเอาไว้เป็นมาตรฐาน

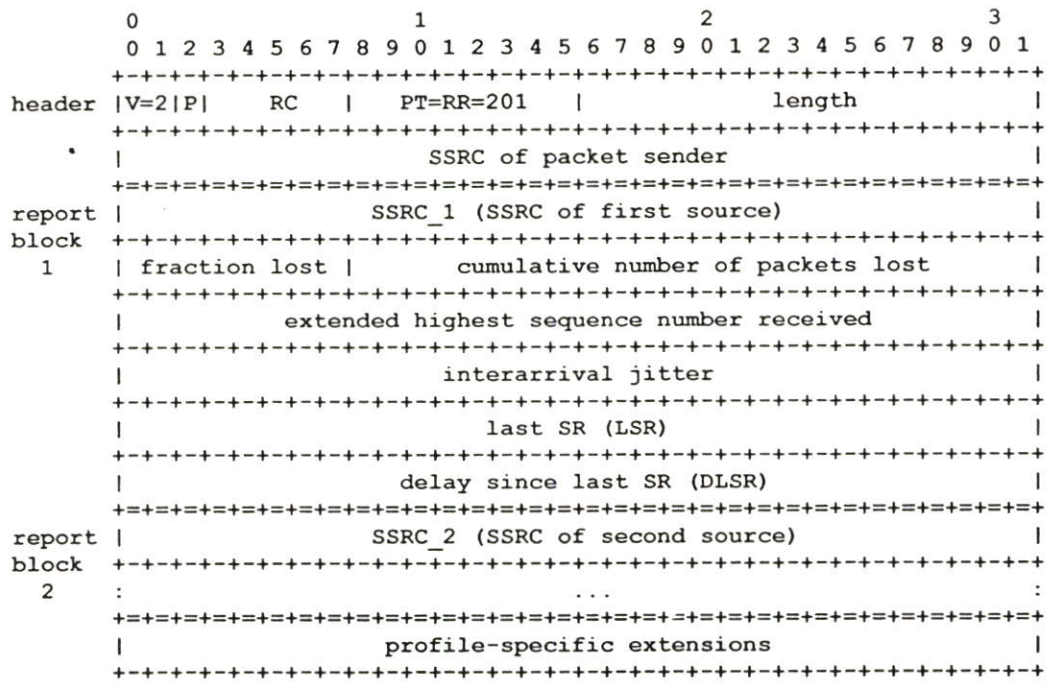
แพ็กเก็ต RTCP ที่น่าสนใจก็คือ RR เพราะว่าผู้รับจะรวบรวมสถิติต่างๆ ของข้อมูลเสียงหรือวีดิทัศน์ที่ได้รับจากแพ็กเก็ต RTP แล้วบรรจุลงในแพ็กเก็ต RR นี้และส่งให้กับผู้ส่ง ผู้ส่งสามารถใช้ข้อมูลในแพ็กเก็ตชนิดนี้ในการประเมินสถานะของเครือข่าย รวมถึงคุณภาพของสื่อที่ผู้รับได้รับ ในการออกแบบวิธีการควบคุมความผิดพลาดแบบปรับตัวของทั้งการสื่อสารเสียงและวีดิทัศน์ของวิทยานิพนธ์นี้ก็จะมีการใช้งานแพ็กเก็ตนี้เช่นกัน ในรูปที่ 2.8 เป็นโครงสร้างเฮดเดอร์ของแพ็กเก็ต RR ซึ่งฟิลด์ต่างๆ ที่กำหนดอยู่ในมาตรฐานแล้ว โครงสร้างเฮดเดอร์ของแพ็กเก็ต RTCP ยังอนุญาตให้มีการใส่ข้อมูลเพิ่มเติมลงในแพ็กเก็ตนี้ได้อีก โดยข้อมูลส่วนที่เพิ่มเติมนี้สามารถใส่ได้ในตำแหน่งที่เรียกว่า Profile-Specific Extension

เฮดเดอร์ของแพ็กเก็ต RR จะประกอบด้วย 3 ส่วน โดยส่วนแรกนั้นจะเป็นเฮดเดอร์ที่มีขนาดคงที่ มีขนาดรวมกันได้ 8 ไบต์ ซึ่งแต่ละฟิลด์มีความหมายดังนี้

- V (Version) กำหนดเวอร์ชันของ RTP ที่ใช้ซึ่งปัจจุบันเป็นเวอร์ชัน 2
- P (Padding) เป็นฟิลด์ที่บอกว่าแพ็กเก็ตนั้นได้ถูกเติมด้วยข้อมูลที่เพิ่มให้พอดีกับขนาดของแพ็กเก็ตหรือไม่ และยังใช้ในกระบวนการการเข้ารหัสข้อมูล (Encryption) อีกด้วย
- P (Payload Type) ชนิดของแพ็กเก็ต ซึ่งในกรณีของแพ็กเก็ต RR ฟิลด์นี้มีค่าเท่ากับ 201
- Length ระบุความยาวของแพ็กเก็ต โดยค่านี้ในฟิลด์นี้มีที่มาดังนี้

  Length = (length of packet in byte / 4) – 1

  โดยค่าในฟิลด์ Length นี้จะมีค่าเป็นลบไม่ได้ ดังนั้นจะเห็นว่าความยาวของทั้งแพ็กเก็ตอย่างน้อยที่สุดต้องเป็น 4 ไบต์
- SSRC of packet sender หมายเลขประจำตัวที่ใช้ในการระบุผู้ที่จะส่งแพ็กเก็ต RR นี้



รูปที่ 2.8 โครงสร้างเฮดเดอร์ของแพ็กเก็ต Receiver Report (RR)[9]

เฮดเดอร์ของแพ็กเก็ต RR ในส่วนที่สองเรียกว่าบล็อกรายงาน (Report Block) ซึ่งในแพ็กเก็ต RR 1 แพ็กเก็ตอาจจะประกอบด้วยบล็อกรายงานหลายบล็อกก็ได้ถ้าหากว่าผู้รับได้รับสื่อจากผู้ส่งหลายคน โดยบล็อกรายงานแต่ละบล็อกจะมีเฮดเดอร์ดังนี้

- SSRC ค่า SSRC ที่ Session ที่ต้องการจะรายงานสถิติการรับ
- Fraction Lost อัตราการสูญหายของแพ็กเก็ตค่าในฟิลด์สามารถคำนวณได้ดังนี้

  Fraction Lost = (Number of lost packets / Number of expected packet) x 256

  และหากต้องการแปลงค่า Fraction Lost ให้กลายเป็นเปอร์เซ็นต์การสูญหายของแพ็กเก็ตก็สามารถทำได้ดังนี้

  Percent Lost = (Fraction Lost / 256) x 100

- Cumulative Number of Packets Lost ผลรวมของจำนวนแพ็กเก็ตที่สูญหายทั้งหมด
- Extended Highest Sequence Number Received หมายเลขลำดับของแพ็กเก็ตสุดท้ายที่ใช้ในการพิจารณาในการคำนวณค่าสถิติในแพ็กเก็ต RR นี้
- Interarrival Jitter ระบุค่าจิตเตอร์ของ Session โดยวิธีคำนวณค่าจิตเตอร์สามารถดูรายละเอียดได้ใน RFC 3550
- LSR (Last SR) เวลาที่ได้รับแพ็กเก็ต SR ครั้งสุดท้าย ซึ่งแพ็กเก็ตนี้ผู้ส่งสื่อจะส่งมาให้

ผู้รับเป็นระยะๆ เพื่อรายงานสถิติการส่ง

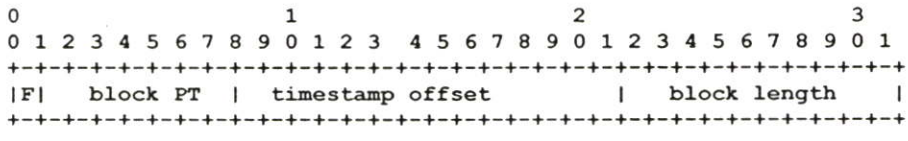
- DLSR (Delay Since Last SR) ระยะห่างระหว่างเวลาที่ได้รับแพ็กเก็ต SR ครั้งสุดท้ายกับเวลาที่เริ่มส่งแพ็กเก็ต RR นี้

เฮดเดอร์ของแพ็กเก็ต RR ในส่วนที่สามซึ่งเป็นส่วนท้ายสุดของแพ็กเก็ตคือ ส่วน Profile-Specific Extension เป็นส่วนที่อนุญาตให้มีการเพิ่มเติมข้อมูลอื่นลงในแพ็กเก็ต RR ได้ ทั้งนี้จำเป็นจะต้องกำหนดค่าฟิลด์ Length ให้สัมพันธ์กับข้อมูลที่เพิ่มลงไปด้วย

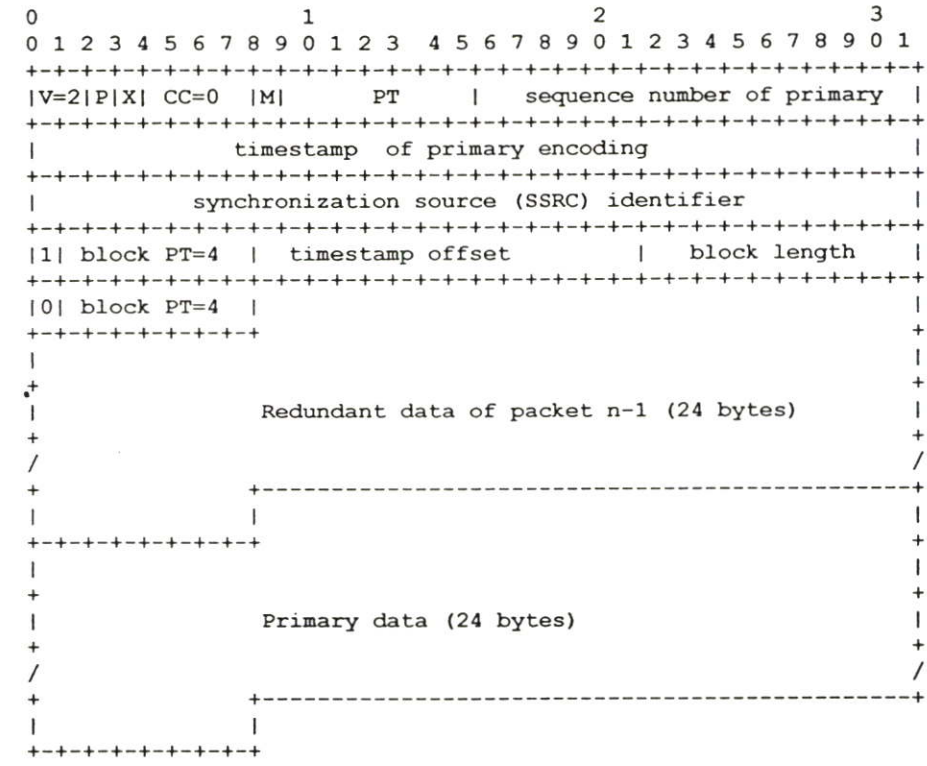
2.4.4 การส่งข้อมูลซ้ำของเสียงโดยใช้โปรโตคอล RTP

เนื่องจากการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ตนั้นไม่สามารถที่จะคาดหมายได้ว่าจะมีปริมาณแพ็กเก็ตที่สูญหายมากน้อยแค่ไหน ดังนั้นจึงได้มีการพัฒนาวิธีการที่จะลดผลกระทบจากการสูญหายของแพ็กเก็ต FEC เป็นวิธีการหนึ่งที่สามารถแก้ปัญหานี้ได้ โดยหลักการของ FEC คือ ในแต่ละแพ็กเก็ตเสียงจะมีการบรรจุข้อมูลซ้ำของแพ็กเก็ตอื่นอยู่ด้วย และในการสื่อสารเสียงโดยใช้โปรโตคอล RTP[9] นั้น ถ้าหากต้องการส่งข้อมูลซ้ำสามารถทำได้โดยการกำหนดรูปแบบของแพ็กเก็ตตามที่กำหนดใน RFC2198[16] จากรูปแบบของแพ็กเก็ตของ RTP นั้น หลังจากส่วนที่เป็นเฮดเดอร์แล้วก็จะตามด้วยส่วนที่เรียกว่า เพย์โหลด (Payload) ซึ่งก็คือข้อมูลเสียงที่ผ่านการบีบอัดแล้วนั่นเอง แต่เมื่อมีการส่งข้อมูลซ้ำ ส่วนของเพย์โหลดจะถูกแบ่งออกเป็นบล็อก ซึ่งมีทั้งบล็อกที่เป็นข้อมูลของแพ็กเก็ตปัจจุบันและบล็อกที่เป็นข้อมูลซ้ำของแพ็กเก็ตก่อนหน้า โดยแต่ละบล็อกอาจเป็นข้อมูลเสียงที่ผ่านการบีบอัดด้วยวิธีที่แตกต่างกันก็ได้ ซึ่งข้อมูลเสียงในแต่ละบล็อกจะมีเฮดเดอร์ดังรูปที่ 2.9 ซึ่งแต่ละฟิลด์มีรายละเอียดดังนี้

- F มีขนาด 1 บิต เป็นฟิลด์ที่ระบุว่ามีเฮดเดอร์ของบล็อกอื่นหลังจากนี้หรือไม่ ถ้า F เป็น 0 แสดงว่าเป็นเฮดเดอร์ของบล็อกสุดท้ายของแพ็กเก็ต
- block PT มีขนาด 7 บิต ใช้ในการระบุชนิดของเพย์โหลด (Payload Type) ซึ่งก็คือวิธีการการบีบอัดของข้อมูลเสียงในแต่ละบล็อก
- timestamp offset มีขนาด 14 บิต ค่านี้จะถูกนำไปลบกับ timestamp ที่อยู่ในเฮดเดอร์ RTP เพื่อใช้ในการระบุว่าเป็นข้อมูลซ้ำของแพ็กเก็ตใด
- block length มีขนาด 10 บิต ใช้ในการระบุความยาวของบล็อกในหน่วยไบต์



รูปที่ 2.9 เฮดเดอร์ของบล็อกในแพ็กเก็ต RTP ที่มีข้อมูลซ้ำของเสียง [16]



รูปที่ 2.10 ตัวอย่างแพ็กเก็ต RTP เมื่อมีการส่งข้อมูลซ้ำของเสียง 1 ชุด

สำหรับเฮดเดอร์ของบล็อกสุดท้ายจะมีแค่ 2 ฟิลด์คือ F และ block PT ทั้งนี้เนื่องจากข้อมูลในบล็อกสุดท้ายนั้นไม่ใช่ข้อมูลซ้ำ แต่เป็นข้อมูลเสียงของแพ็กเก็ตปัจจุบัน รูปที่ 2.10 เป็นตัวอย่างแพ็กเก็ต RTP เมื่อมีการส่งข้อมูลซ้ำของเสียงจำนวน 1 ชุด โดยทั้งข้อมูลหลักและข้อมูลซ้ำใช้การบีบอัดเสียงแบบ G.723.1 โดยทั้งบล็อกที่เป็นข้อมูลหลักและบล็อกที่เป็นข้อมูลซ้ำประกอบด้วยข้อมูลเสียงของ G.723.1 เพียง 1 เฟรม (24 ไบต์)

จากหลักการพื้นฐานของการสื่อสารเสียงที่ได้กล่าวไปแล้วนั้น จะเห็นว่ามีช่องทางสำหรับการควบคุมคุณภาพของเสียงได้ เช่น ในโปรโตคอล RTP ผู้รับสามารถรายงานสถิติการรับให้ผู้ส่งได้รับทราบโดยใช้แพ็กเก็ต RTCP Receiver Report ในการสื่อสารเสียงก็สามารถที่จะควบคุมความผิดพลาดได้โดยใช้วิธีการส่งข้อมูลซ้ำ ซึ่งวิธีการนี้มีมาตรฐานรองรับแล้ว เนื้อหาเกี่ยวกับวิธีการควบคุมความผิดพลาดของการสื่อสารเสียงโดยใช้วิธีการปรับให้เรียบแบบเอกซโพเนนเชียลจะอยู่ในบทที่ 3

บทที่ 3
การควบคุมความผิดพลาดโดยใช้วิธีการปรับให้เรียบ

เนื่องจากการสูญหายของข้อมูลเสียงเป็นสาเหตุสำคัญที่ทำให้คุณภาพเสียงลดลง และการสื่อสารเสียงบนเครือข่ายอินเตอร์เน็ตไม่สามารถที่จะหลีกเลี่ยงปัญหาการสูญหายของแพ็กเก็ตได้ ดังนั้นจึงจำเป็นที่จะต้องมีกลไกในการลดผลกระทบจากการสูญหายของแพ็กเก็ต ซึ่งในที่นี้จะเรียกกลไกที่ทำหน้าที่นี้ว่า การควบคุมความผิดพลาด (Error Control) เนื้อหาของบทนี้เริ่มต้นด้วยการกล่าวถึงอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัว (Adaptive Error Control) ในหัวข้อที่ 3.1 โดยมีการอธิบายถึงอัลกอริทึมที่มีการเสนอขึ้นมาก่อนหน้านี้ แต่อัลกอริทึมเหล่านี้ก็ยังมีจุดบกพร่องอยู่บ้าง วิทยานิพนธ์นี้จึงได้ศึกษาวิธีการควบคุมความผิดพลาดแบบปรับตัวชื่อว่าอัลกอริทึม CNR (Centre Network Research) ซึ่งเป็นอัลกอริทึมที่มีการปรับปรุงจากข้อบกพร่องของอัลกอริทึมที่ได้เสนอก่อนหน้านี้ โดยรายละเอียดของอัลกอริทึมนี้อยู่ในหัวข้อที่ 3.1.4 และวิทยานิพนธ์นี้ได้ประยุกต์ใช้เทคนิคการปรับให้เรียบแบบเอกซโพเนนเชียลครั้งเดียวโดยการปรับค่าคงที่ให้เรียบเพิ่มเติมในอัลกอริทึม CNR เพื่อทำนายอนุกรมอัตราการสูญหายของข้อมูลเสียงอยู่ในหัวข้อที่ 3.1.5 ในการประเมินผลอัลกอริทึมในการควบคุมความผิดพลาด ต้องมีการตรวจสอบความถูกต้องของแบบจำลองที่ได้จัดทำขึ้น ซึ่งเนื้อหาส่วนนี้อยู่ในหัวข้อที่ 3.2

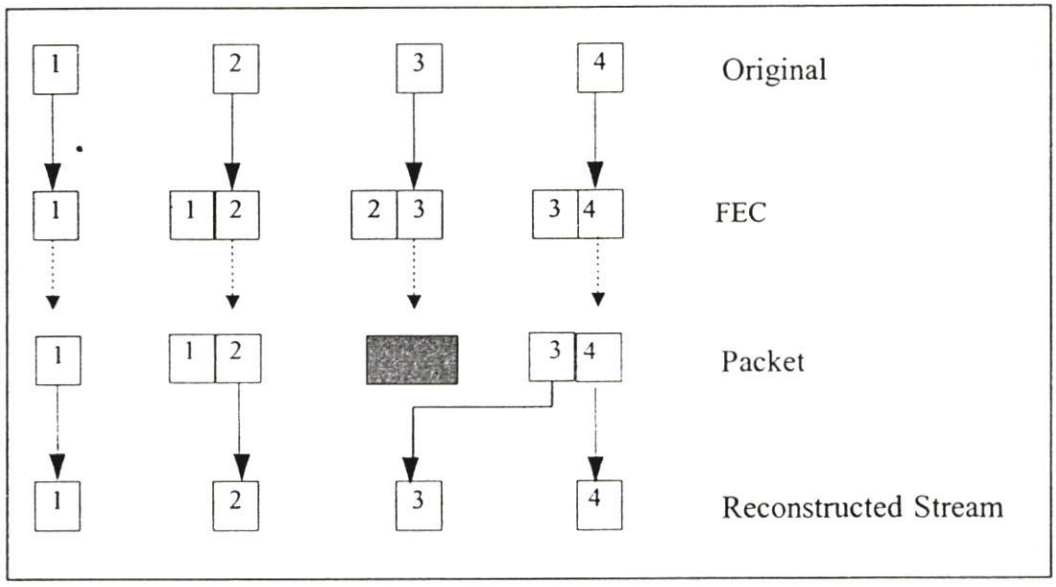
3.1 การควบคุมความผิดพลาดแบบปรับตัว

ในการสื่อสารเสียงบนเครือข่ายที่ไม่มีการรับประกันคุณภาพการบริการอย่างเช่นเครือข่ายอินเตอร์เน็ตนั้น ปัญหาอย่างหนึ่งที่ส่งผลกระทบต่อคุณภาพของเสียงก็คือการสูญหายของแพ็กเก็ต ถึงแม้ว่าในการสื่อสารเสียงนั้นไม่จำเป็นจะต้องได้รับข้อมูลครบทั้งหมดก็ตาม แต่การที่มีแพ็กเก็ตสูญหายมากเกินไปก็ทำให้คุณภาพเสียงต่ำเกินกว่าที่ยอมรับได้ ดังนั้นจึงจำเป็นต้องมีกลไกในการลดผลกระทบจากการสูญหายของแพ็กเก็ต ซึ่งกลไกดังกล่าวก็คือ การควบคุมความผิดพลาด ซึ่ง Forward Error Correction (FEC) ก็เป็นวิธีการควบคุมความผิดพลาดวิธีการหนึ่งที่สามารถลดผลกระทบจากการสูญหายของแพ็กเก็ตได้โดยไม่เพิ่มค่าเวลาหน่วงมากนัก [13] ซึ่งคุณสมบัตินี้เหมาะสมสำหรับการสื่อสารเสียงที่ต้องการความเป็นเวลาจริง

FEC สามารถแบ่งออกได้เป็น 2 ประเภทคือ FEC ที่ไม่ขึ้นกับสื่อ (Media Independent FEC) และ FEC ที่ขึ้นกับสื่อ (Media Specific FEC) โดยประเภทแรกคือ FEC ที่ไม่ขึ้นกับสื่อ มีหลักการคือ เมื่อมีการส่งแพ็กเก็ตได้หนึ่งชุด จะมีการส่งแพ็กเก็ตพิเศษเพิ่มเข้าไป โดยแพ็กเก็ตพิเศษนี้ได้จากการกระทำทางคณิตศาสตร์ของข้อมูลจากทุกแพ็กเก็ตที่อยู่ในชุดเดียวกัน ดังนั้นเมื่อมีแพ็กเก็ตใดแพ็กเก็ตหนึ่งสูญหายไป ข้อมูลในแพ็กเก็ตที่สูญหายสามารถกู้คืนได้โดยใช้แพ็กเก็ตพิเศษนี้

แต่วิธีการนี้ใช้ไม่ได้ผลถ้ามีแพ็กเก็ตในชุดเดียวกันสูญหายมากกว่า 1 แพ็กเก็ต

FEC ประเภทที่สองคือ FEC ที่ขึ้นกับสื่อมีหลักการคือ ในแต่ละแพ็กเก็ตที่ผู้ส่งจะส่งไปให้ผู้รับ จะมีการบรรจุข้อมูลซ้ำ (Redundancy) ของแพ็กเก็ตที่อยู่ก่อนหน้า แต่อาจใช้การบีบอัดด้วยอัตราบิตที่ต่ำกว่าเพื่อไม่ให้ปริมาณการใช้แบนด์วิดท์เพิ่มขึ้นมากเกินไป จากรูปที่ 3.1 จะเห็นว่าเมื่อแพ็กเก็ตที่ 3 สูญหาย ข้อมูลเสียงที่อยู่ในแพ็กเก็ตนี้สามารถกู้คืนได้จากข้อมูลซ้ำที่อยู่ในแพ็กเก็ตที่ 4 ซึ่ง FEC ประเภทที่สองนี้เป็นวิธีการที่ได้รับความนิยมมากกว่า FEC ประเภทแรก และมีมาตรฐานที่กำหนดรูปแบบของการบรรจุข้อมูลซ้ำของเสียงลงในแพ็กเก็ต RTP อีกด้วย ซึ่งมาตรฐานดังกล่าวก็คือ RFC2198 [16] แต่มาตรฐานนี้ก็กำหนดเพียงแค่รูปแบบของแพ็กเก็ตเท่านั้น แต่ไม่ได้กำหนดว่าต้องใช้ปริมาณข้อมูลซ้ำเท่าใด และข้อมูลซ้ำแต่ละชุดใช้วิธีการบีบอัดเสียงแบบใด



รูปที่ 3.1 Media Specific FEC

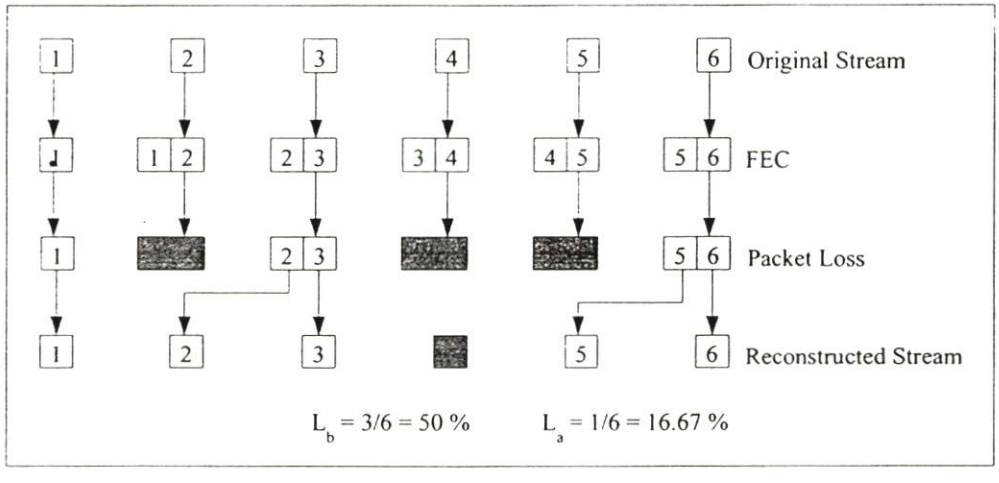
เนื่องจากการควบคุมความผิดพลาดโดยใช้ FEC มีการส่งข้อมูลซ้ำของเสียงซึ่งทำให้ปริมาณการใช้แบนด์วิดท์ของสื่อสารเสียงเพิ่มขึ้น ดังนั้นจึงควรกำหนดปริมาณข้อมูลซ้ำให้เหมาะสมกับสภาพของเครือข่าย ในกรณีที่มีการสูญหายของแพ็กเก็ตเพียงเล็กน้อย หากกำหนดปริมาณข้อมูลซ้ำมากเกินไปจะเป็นการสิ้นเปลืองแบนด์วิดท์โดยไม่จำเป็น ดังนั้นในการสื่อสารเสียงจึงควรมีอัลกอริทึมในการกำหนดรูปแบบและปริมาณของข้อมูลซ้ำที่สามารถปรับตัวได้ตามปริมาณการสูญหายของแพ็กเก็ต ซึ่งอัลกอริทึมที่ได้มีการเสนอก่อนหน้านี้ได้แก่ อัลกอริทึม Bolot[17], อัลกอริทึม RCCS[19] และอัลกอริทึม CNR[35] อัลกอริทึมทั้งสามนี้ล้วนแต่ใช้พื้นฐานของ FEC ประเภทที่ขึ้นกับสื่อ แต่สิ่งที่แตกต่างกันก็คือวิธีการที่จะเลือกว่าเมื่อใดควรที่จะเลือกรูปแบบของข้อมูลซ้ำแบบใดและใช้ปริมาณข้อมูลซ้ำเท่าใด

3.1.1 อัตราการสูญหายของข้อมูลเสียงเมื่อมีการใช้ FEC

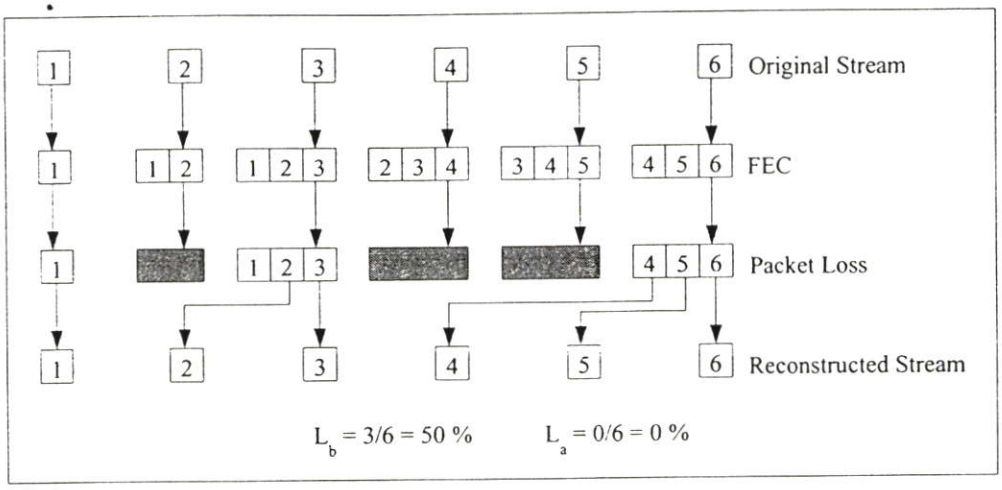
ก่อนจะกล่าวถึงอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวในแต่ละอัลกอริทึม ในหัวข้อนี้จะอธิบายเกี่ยวกับค่าอัตราการสูญหายของข้อมูลเมื่อมีการใช้เทคนิค FEC ทั้งนี้เพื่อให้สามารถเข้าใจถึงรายละเอียดของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวที่จะกล่าวถึงในหัวข้อถัดไปได้ง่ายขึ้น ซึ่งเมื่อมีการใช้เทคนิค FEC แล้ว ค่าอัตราการสูญหายของข้อมูลที่จะต้องพิจารณามีอยู่ 2 ค่าคือ ค่าอัตราการสูญหายก่อนการจัดเรียง (Loss Rate before Reconstruction) ซึ่งในที่นี้จะแทนด้วยสัญลักษณ์ L_b และค่าอัตราการสูญหายหลังการจัดเรียง (Loss Rate after Reconstruction) ซึ่งในที่นี้จะแทนด้วยสัญลักษณ์ L_a

ค่า L_b หมายถึงค่าอัตราการสูญหายของข้อมูลเสียงโดยที่ยังไม่ได้นำข้อมูลซ้ำมาจัดเรียงใหม่ หรืออาจจะกล่าวได้ว่าค่า L_b คืออัตราการสูญหายของแพ็กเก็ตนั่นเอง จากรูปที่ 3.2 จะเห็นว่ามีการส่งแพ็กเก็ตเสียงทั้งหมด 6 แพ็กเก็ต แต่เกิดการสูญหาย 3 แพ็กเก็ต ดังนั้นค่า L_b จึงมีค่าเท่ากับ 3/6 หรือ 50 % นั่นเอง ส่วนค่า L_a จะคำนวณจากจำนวนข้อมูลเสียงที่สูญหายหลังจากที่ได้มีการจัดเรียงข้อมูลแล้ว ค่า L_a จึงมักจะมีค่าน้อยกว่า L_b เนื่องจากอาจมีการกู้ข้อมูลในแพ็กเก็ตที่สูญหายบางส่วนได้จากข้อมูลซ้ำ จากรูปที่ 3.2 จะเห็นว่าเมื่อมีการจัดเรียงข้อมูลแล้วมีข้อมูลเสียงสูญหายไปเพียง 1 เฟรมเท่านั้นคือเฟรมที่ 4 ดังนั้นค่า L จึงมีค่าเท่ากับ 1/6 หรือ 16.67 % ซึ่งค่า L_a นี้เองจะเป็นตัวบอกถึงปริมาณข้อมูลเสียงที่สูญหาย ถ้า L_a มีค่ามากจะทำให้เสียงที่ได้รับมีคุณภาพต่ำ ส่วนค่า L_b เป็นค่าที่บอกถึงปริมาณการสูญหายของแพ็กเก็ตเท่านั้น

การเพิ่มจำนวนชุดของข้อมูลซ้ำสามารถลดค่า L_a ได้ถึงแม้ว่าค่า L_b จะมีค่าเท่าเดิม จากรูปที่ 3.3 จะเห็นว่าเมื่อเพิ่มจำนวนข้อมูลซ้ำเป็น 3 ชุด สามารถกู้คืนข้อมูลเสียงที่สูญหายได้หมด ทำให้ค่า L_a เท่ากับ 0 % ดังนั้นในกรณีที่เครือข่ายมีความมีคับคั่งสูงและมีปริมาณแพ็กเก็ตสูญหายมากจึงควรจะมีเพิ่มจำนวนชุดข้อมูลซ้ำ เพื่อให้สามารถกู้ข้อมูลเสียงในแพ็กเก็ตที่สูญหายได้มากขึ้น และสามารถรักษาระดับของอัตราการสูญหายของข้อมูลเสียงไม่ให้สูงเกินไปได้



รูปที่ 3.2 การกู้คืนโดยใช้ข้อมูลซ้ำสามารถทำให้ค่า L_a น้อยกว่า L_b



รูปที่ 3.3 การเพิ่มจำนวนข้อมูลซ้ำสามารถช่วยลดค่า L_a ได้

3.1.2 อัลกอริทึม Bolot

อัลกอริทึมนี้ได้รับการเสนอโดย Bolot และได้รับการตีพิมพ์ลงใน [17] ซึ่งในบทความนี้ได้มีการกำหนดรูปแบบของการจัดวางข้อมูลซ้ำ (Combination) เอาไว้หลายแบบ และได้ทำการทดลองเพื่อหาค่า Reward ของแต่ละ Combination โดยค่า Reward (R_c) คืออัตราส่วนระหว่างค่า L_b และค่า L_a

$$R_c = L_b / L_a \tag{3.1}$$

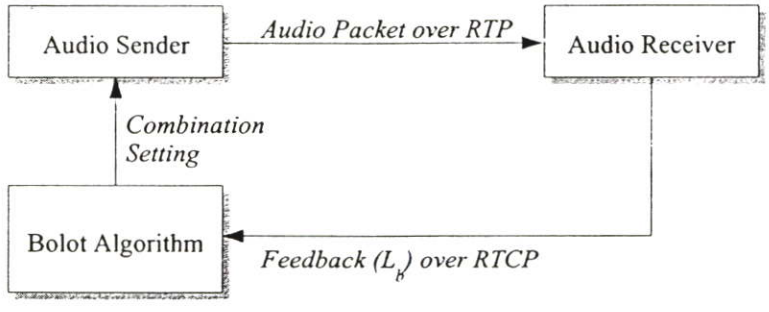
ตารางที่ 3.1 ตาราง Combination ที่ใช้อัลกอริทึม Bolot

No.	Combination	Reward
0	(PCM)	1
1	(PCM, ADM4(1))	2.5
2	(PCM, GSM(1))	2.5
3	(PCM, LPC(1))	2.5
4	(PCM, ADM4(2))	6
5	(PCM, ADM4(1), ADM2(2))	6
6	(PCM, ADM4(1), ADM2(3))	10
7	(PCM, ADM4(1), ADM2(2), ADM2(3))	18

ค่า Reward ของแต่ละ Combination ที่ปรากฏในตารางที่ 3.1 เป็นค่าที่ได้จากการทดลอง

โดยใช้เครือข่ายระหว่างสถาบัน INRIA ในประเทศฝรั่งเศสกับ University College London (UCL) ในประเทศอังกฤษ สำหรับสัญลักษณ์ (PCM) หมายถึง ทุกแพ็กเก็ตใช้การบีบอัดแบบ PCM โดยไม่มีการบรรจุข้อมูลซ้ำ ส่วนสัญลักษณ์ (PCM, ADM4(1)) หมายถึง ในแต่ละแพ็กเก็ตลำดับที่ N ส่วนที่เป็นข้อมูลหลักใช้การบีบอัดแบบ PCM แต่ส่วนที่เป็นข้อมูลซ้ำใช้การบีบอัดแบบ ADM4 และเป็นข้อมูลซ้ำของแพ็กเก็ตที่ N-1 (ตัวเลข 1 ที่อยู่ในวงเล็บ) และสัญลักษณ์ของ Combination แบบอื่นๆ สามารถแปลความหมายได้ในทำนองเดียวกัน ซึ่งค่า Reward นี้เป็นค่าที่สามารถบอกได้ว่าแต่ละ Combination สามารถกู้ข้อมูลเสียงในแพ็กเก็ตที่สูญหายได้ดีเพียงใด และจะเห็นว่า Reward จะเพิ่มขึ้นเมื่อมีการเพิ่มจำนวนชุดของข้อมูลซ้ำ ยกเว้นในกรณีของ Combination ที่ 4 และ 5 ซึ่งมีค่า Reward เท่ากัน และในกรณีของ Combination ที่มีการใช้จำนวนข้อมูลซ้ำเท่ากัน Combination ที่ใช้ข้อมูลซ้ำจากแพ็กเก็ตที่อยู่ห่างจากแพ็กเก็ตปัจจุบันมากกว่าจะมีค่า Reward สูงกว่า ตัวอย่างเช่น Combination ที่ 4 มีค่า Reward สูงกว่า Combination ที่ 3 แต่อย่างไรก็ตามค่า Reward ที่ได้แสดงในตารางที่ 3.1 นี้เป็นค่าที่ได้จากการทดลองเท่านั้น ซึ่งอาจจะไม่เป็นจริงเสมอไปในทุกสภาพแวดล้อม

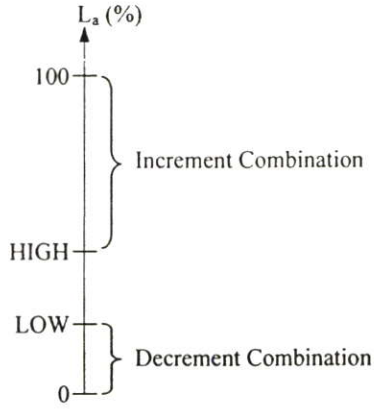
จากแผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม Bolot ในรูปที่ 3.4 จะเห็นว่าผู้รับจะมีการรายงานเฉพาะค่า L_b ให้ผู้ส่งได้ทราบเท่านั้นโดยจะระบุค่าของ L_b ลงฟิลด์ Fraction Lost ของแพ็กเก็ต RTCP ชนิด Receiver Report ซึ่งปกติฟิลด์ Fraction Lost นี้จะใช้ในการระบุค่าอัตราการสูญหายของแพ็กเก็ต และเนื่องจากค่า L_b มีค่าเท่ากับอัตราการสูญหายของแพ็กเก็ต จึงสามารถบรรจุลงในฟิลด์นี้ได้เลย แต่ผู้รับจะไม่รายงานค่า L_a ทั้งนี้เนื่องจากว่าอัลกอริทึม Bolot ไม่ได้ใช้ค่าจริงของ L_a แต่จะคำนวณค่า L_a โดยใช้ค่า L_b หารด้วยค่า Reward ของ Combination ที่กำลังใช้งาน และนำค่า L_a ไปใช้ในการตัดสินใจว่าควรเลือกที่จะเพิ่มหรือลด Combination



รูปที่ 3.4 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม Bolot

จากรูปที่ 3.5 จะเห็นว่าอัลกอริทึม Bolot มีค่าเทรสโฮลด์ที่ใช้ในการพิจารณาค่า L_a อยู่ 2 ค่า คือ HIGH และ LOW ถ้าหากค่า L_a สูงเกินกว่าค่าเทรสโฮลด์ HIGH ให้เพิ่ม Combination โดยการเพิ่ม Combination หมายถึงการเลือกใช้ Combination ในตารางที่ 3.1 ในขั้นที่สูงขึ้น เช่น ถ้าหากเดิมใช้ Combination หมายเลข 1 เมื่อใดที่ค่า L_a สูงกว่าเทรสโฮลด์ HIGH ก็จะเลื่อนขึ้นไปใช้

Combination หมายเลข 2 เป็นต้น แต่ถ้าค่า L_a ต่ำกว่าค่าเทรสโฮลด์ LOW ให้ลด Combination โดยใน [17] ได้กำหนดให้ค่าเทรสโฮลด์ HIGH และ LOW มีค่าเท่ากันคือ 3% เพื่อควบคุมให้ค่า L_a อยู่ที่ 3% แต่การกำหนดค่าเทรสโฮลด์ทั้งสองให้มีค่าเท่ากันจะทำให้มีการเปลี่ยน Combination บ่อยครั้ง เพราะไม่มีช่วงของค่า L_a ช่วงใดเลยที่อัลกอริทึมนี้ไม่ต้องมีการเปลี่ยนแปลง Combination



รูปที่ 3.5 การเพิ่มและลด Combination ในอัลกอริทึม Bolot

หลักการทำงานของอัลกอริทึม Bolot สามารถสรุปเป็น Pseudo Code ได้ดังรูปที่ 3.6 โดยขั้นตอนที่ 1 เป็นคำนวณค่า L_b จากค่าที่อ่านได้จากแพ็กเก็ต Recevier Report โดยค่าของ L_b จะระบุอยู่ในในฟิลด์ Fraction Lost ซึ่งปกติแล้วการระบุค่าอัตราการสูญหายในแพ็กเก็ตในฟิลด์ Fraction Lost จะไม่ระบุเป็นค่าทศนิยมหรือค่าที่เป็นเปอร์เซ็นต์โดยตรง แต่จะระบุเป็นอัตราส่วนเมื่อเทียบกับ 256[9] เนื่องจากฟิลด์นี้มีขนาด 1 ไบต์ ตัวอย่างเช่น ถ้าค่าในฟิลด์ Fraction Loss เท่ากับ 128 หมายถึงอัตราการสูญหายของแพ็กเก็ตเท่ากับ 128/256 = 50%

ในขั้นตอนที่ 2 เป็นการคำนวณหาค่า L_a โดยใช้ค่า L_b หารด้วย Reward ของ Combination ปัจจุบัน จากนั้นในขั้นตอนที่ 3 และขั้นตอนที่ 4 จะนำค่า L_a ที่คำนวณได้ไปใช้ในการตัดสินใจว่าจะเพิ่มหรือลด Combination

```
For each RTCP packet received do
   1. Calculate loss rate before reconstruction, Lb
   2. Calculate loss rate after reconstruction
      La = Lb / Reward associated with current combination number
   3. If (La > HIGH) then
         Increment combination
   4. If (La < LOW) then
         Decrement combination
```

รูปที่ 3.6 Pseudo Code อัลกอริทึม Bolot

3.1.3 อัลกอริทึม RCCS

อัลกอริทึม RCCS[19] ซึ่งย่อมาจาก Redundant Codec Combination Selection เป็นอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวที่มีการเลือก Combination โดยใช้พารามิเตอร์สองค่าคือ อัตราการสูญหายของข้อมูลและค่าเวลาหน่วงระหว่างปลายทาง (End-to-End Delay) ซึ่งแตกต่างจากอัลกอริทึม Bolot ที่ใช้เพียงอัตราการสูญหายของข้อมูลเพียงอย่างเดียว ตารางที่ 3.2 แสดง Combination ที่ใช้ในอัลกอริทึม RCCS ซึ่งจะเห็นว่ารูปแบบของการจัดวางข้อมูลซ้ำ รวมทั้งชนิดของการบีบอัดเสียงในแต่ละ Combination นั้นแตกต่างไปจากอัลกอริทึม Bolot ในคอลัมน์ Reward แสดงค่า Reward ของแต่ละ Combination ซึ่งค่าเหล่านี้ได้มาจากค่า Reward ของ Bolot นั่นเอง โดยค่า Reward ของแต่ละ Combination ในตารางที่ 3.2 ของอัลกอริทึม RCCS นำมาจากค่า Reward ของ Combination ในตารางที่ 3.1 ของอัลกอริทึม Bolot ที่มีจำนวนข้อมูลซ้ำและตำแหน่งของข้อมูลซ้ำตรงกัน โดยไม่สนใจชนิดของการบีบอัดเสียง แต่ค่า Reward ในตารางที่ 3.2 นี้เป็นเพียงค่า Reward เริ่มต้นเท่านั้น เพราะอัลกอริทึม RCCS จะมีการเปลี่ยนแปลงค่า Reward ของ Combination ที่กำลังใช้งานทุกครั้งที่ได้รับแพ็กเก็ต Receiver Report และในคอลัมน์ขวาสุดคือคอลัมน์ Penalty เป็นค่าอัตราส่วนระหว่างค่าเวลาหน่วงระหว่างปลายทางหลังการจัดเรียงข้อมูลและค่าเวลาหน่วงระหว่างปลายทางก่อนการจัดเรียงข้อมูลซ้ำ

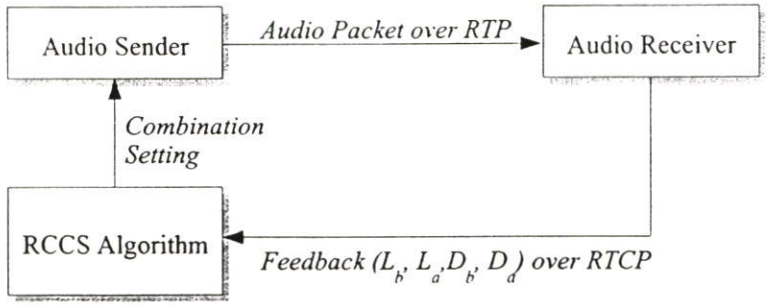
$$Penalty = \frac{End\text{-}to\text{-}End\ Delay\ after\ Reconstruction}{End\text{-}to\text{-}End\ Delay\ before\ Reconstruction} \tag{3.2}$$

ซึ่งค่า Penalty ที่อยู่ในตารางที่ 3.2 เป็นค่าเฉลี่ยที่ได้จากการทดลองหลายๆ ครั้ง แต่ใน [19] ก็ไม่ได้อธิบายเอาไว้อย่างชัดเจนว่าการหาค่าเวลาหน่วงก่อนและหลังการจัดเรียงมีวิธีการอย่างไร ซึ่งการหาค่าเวลาหน่วงระหว่างปลายทางนั้นทำได้ยาก เนื่องจากว่านาฬิกาของเครื่องผู้ส่งและเครื่องผู้รับไม่ประสานกัน จากแผนภาพการควบคุมความผิดพลาดแบบปรับตัวของอัลกอริทึม RCCS ในรูปที่ 3.7 จะเห็นว่าค่าที่ผู้รับจะต้องรายงานให้กับผู้ส่งมีอยู่ 4 ค่า คือ L_b, L_a, D_b (เวลาหน่วงระหว่างปลายทางก่อนการจัดเรียงข้อมูล) และ D_a (เวลาหน่วงระหว่างปลายทางหลังการจัดเรียงข้อมูล)

ตารางที่ 3.2 ค่า Reward และ Penalty ของแต่ละ Combination ในอัลกอริทึม RCCS

No.	Combination	Reward	Penalty
0	(G.711)	1	1
1	(G.711,GSM(1))	2.5	1.5
2	(G.711,G.723(1))	2.5	2
3	(GSM,G.723(1))	2.5	4

No.	Combination	Reward	Penalty
4	(G.711,GSM(1),G.729(2))	6	2.4
5	(G.729,G.723(1),LPC10(2))	6	4.5
6	(G.711,GSM(1),G.729(2),LPC10(3))	18	3.4
7	(G.711,GSM(1),G.723(2),LPC10(3))	18	4.5



รูปที่ 3.7 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม RCCS

รูปที่ 3.8 เป็น Pseudo Code ของอัลกอริทึม RCCS ซึ่งจะเห็นว่ามีการใช้พารามิเตอร์หลายตัว โดยคำอธิบายของพารามิเตอร์แต่ละตัวนั้นอยู่ในตารางที่ 3.3 จาก Pseudo Code จะเห็นเมื่อได้ค่า L_b, L_a, D_b และ D_a ในขั้นตอนที่ 1 - 2 แล้ว ในขั้นตอนที่ 3 จะมีการแก้ไขค่า Reward (R) และ Penalty (P) ของ Combination ปัจจุบัน โดยใช้สมการตัวกรองดังนี้

$$R_i = \alpha(L_b / L_a) + (1-\alpha)R_{i-1} \quad (3.3)$$

$$P_i = \alpha(D_a / D_b) + (1-\alpha)P_{i-1} \quad (3.4)$$

ตารางที่ 3.3 คำอธิบายของพารามิเตอร์ที่ใช้ในอัลกอริทึม RCCS

พารามิเตอร์	คำอธิบาย
R_i	ค่า Reward ของ Combination ในลำดับที่ i
P_i	ค่า Penalty ของ Combination ในลำดับที่ i
L_a (L_a')	อัตราการสูญหายของแพ็กเก็ตหลังการจัดเรียง (ค่า L_a' คือค่าทำนายของ L_a)
L_b (L_b')	อัตราการสูญหายของแพ็กเก็ตก่อนการจัดเรียง (ค่า L_b' คือค่าทำนายของ L_b)
D_a	ค่าเวลาหน่วงระหว่างปลายทางหลังการจัดเรียง
D_b	ค่าเวลาหน่วงระหว่างปลายทางก่อนการจัดเรียง
α	ค่าสำหรับกำหนดความเร็วของการเปลี่ยนแปลงค่า R_i และ P_i

```
For each RTCP packet received do
1.Calculate packet loss and delay before reconstruction Lb and Db
  Lb = Number of packet loss before reconstruction / Number of
       packet expected
  Db = end-to-end delay before reconstruction

2.Calculate packet loss and delay after reconstruction La and Da
  La = Number of packet loss after reconstruction / Number of
       packet expected
  Da = end-to-end delay after reconstruction

3.Update reward and penalty of current combination with Ri and Pi
  in combination table

4.If(La > HIGH or La < LOW)
    For each combination j in the combination table
       Calculate and predict La', La' = Lb/Rj
       Calculate and predict Da', Da' = Db*Pj
          If(La' < HIGH and La' > LOW)
             If(Da' < DMin)
                selected combination = j
                Dmin = Da'

5.combination = selected combination
```

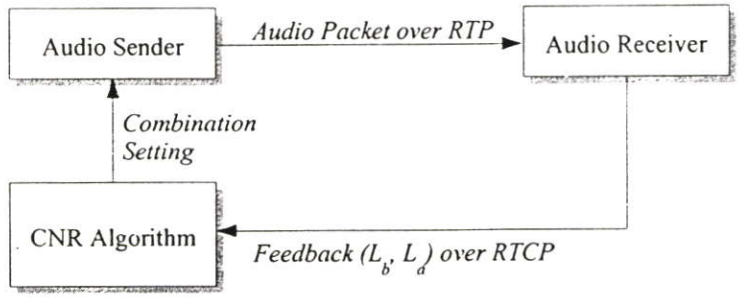
รูปที่ 3.8 Pseudo Code ของอัลกอริทึม RCCS

การแก้ไขค่า Reward นี้เพื่อให้ค่า Reward มีค่าเปลี่ยนแปลงไปตามสภาพเครือข่ายจริง เนื่องจากค่า Reward ที่ได้จากการทดลองของ Bolot อาจจะไม่ได้ถูกต้องเสมอไปในทุกสภาพแวดล้อม จากนั้นในขั้นตอนที่ 4 เป็นการเลือก Combination โดยจะพิจารณาจากตารางที่ 3.4 ตั้งแต่ Combination แรกจนถึง Combination สุดท้าย และจะเลือก Combination แรกที่ให้ค่า L_a' (ค่าทำนายของ L_a ซึ่งได้จากการนำ L_b หารด้วยค่า Reward) อยู่ในช่วงระหว่างเทรสโฮลด์ LOW และ HIGH และ Combination ดังกล่าวจะต้องให้ค่า D_a' (ค่าทำนายของเวลาหน่วงระหว่างปลายทางหลังการจัดเรียงข้อมูล) ไม่เกินค่าเทรสโฮลด์ DMin ด้วย

3.1.4 อัลกอริทึม CNR

อัลกอริทึม CNR (Centre for Network Research) ได้แก้ไขจุดบกพร่องของอัลกอริทึม Bolot โดยการใช้ค่าจริงของ L_a ในการตัดสินเปลี่ยน Combination และได้แก้ไขปัญหาการแกว่งของค่า L_a โดยการนับจำนวนครั้งที่ค่าของ L_a ต่ำกว่าเทรสโฮลด์ LOW โดยจะลด Combination ก็ต่อเมื่อ L_a มีค่าต่ำกว่าเทรสโฮลด์ LOW ติดต่อนานเพียงพอแล้ว เพื่อให้แน่ใจความคับคั่งของเครือข่ายลดลงจริง และอัลกอริทึม CNR ได้แก้ไขจุดบกพร่องของอัลกอริทึม RCCS โดยการแยกเงื่อนไขในการตัดสินใจลดและเพิ่ม Combination ออกจากกัน

จากแผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม CNR ใน [9] จะเห็นว่าผู้รับจะมีการรายงานค่า L_b และ L_a ให้กับผู้ส่งโดยใช้โปรโตคอล RTCP ซึ่งผู้ส่งจะนำค่าทั้งสองนี้ไปใช้ในการตัดสินใจว่าควรจะเลือก Combination ใด โดยแพ็กเก็ต RTCP ที่ใช้ในการรายงานค่าทั้งสองนี้ก็คือแพ็กเก็ต Receiver Report ผู้รับจะนำค่า L_b วางในฟิลด์ที่ชื่อ Fraction Lost ส่วนค่า L_a นั้นจะถูกนำไปอาไว้ในส่วนขยายเฮดเดอร์ RTCP



รูปที่ 3.9 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม CNR

รูปที่ 3.9 เป็นรูปแบบของแพ็กเก็ต Receiver Report ที่มีการระบุทั้งค่า L_b และ L_a โดยระบุที่ฟิลด์ fraction lost และ fraction L_a ตามลำดับ ฟิลด์ fraction L_a อยู่ในส่วนขยายเฮดเดอร์ และมีขนาด 8 บิตเช่นเดียวกับฟิลด์ fraction lost แต่เนื่องจากใน[9] ได้ระบุได้ว่าความยาวของแพ็กเก็ต RTCP จะต้องมีขนาดที่หาร 4 ลงตัว ดังนั้นจึงมีส่วนที่ไม่ได้ใช้งานเหลืออยู่ 3 ไบต์ และเนื่องจากว่าใน [9] ได้กำหนดเอาไว้ว่าค่าในฟิลด์ fraction lost คือค่าอัตราการสูญหายของแพ็กเก็ตคูณด้วย 256 ดังนั้นการระบุค่า L_b และ L_a จึงใช้วิธีการเดียวกันคือ

$$fraction\ loss = L_b \times 256 \tag{3.5}$$

$$fraction\ L_a = L_a \times 256 \tag{3.6}$$

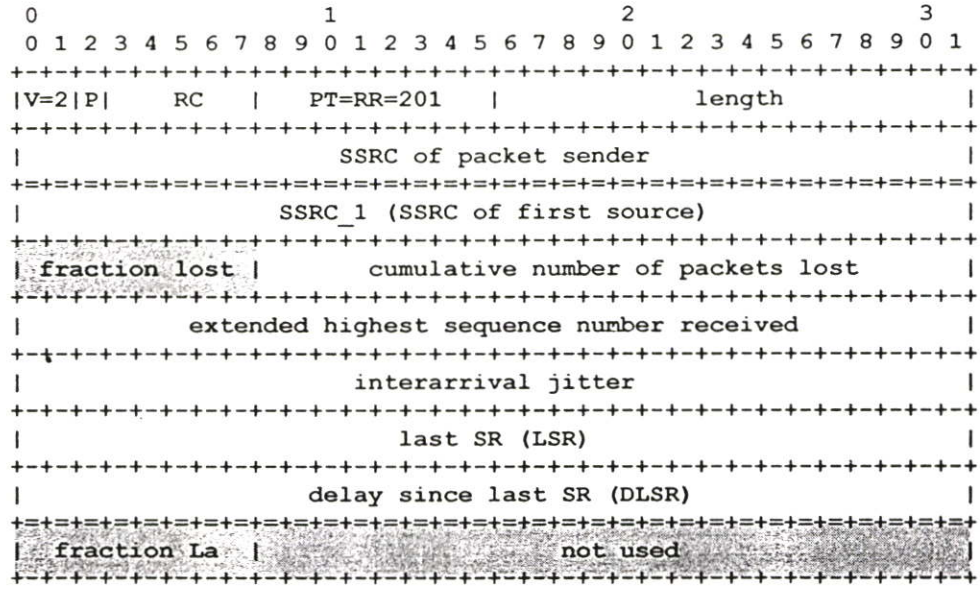
และในการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม CNR เมื่อผู้ส่งได้รับแพ็กเก็ต Receiver Report ผู้ส่งก็จะอ่านค่า L_b และ L_a ที่อยู่ในแพ็กเก็ตดังนี้

$$L_b = fraction\ loss / 256 \tag{3.7}$$

$$L_a = fraction\ L_a / 256 \tag{3.8}$$

ซึ่งค่า L_b และ L_a จะถูกนำไปใช้ในการตัดสินเลือกว่าจะใช้ Combination ใด แต่อัลกอริทึม CNR นั้นไม่ได้มีการเพิ่ม Combination ครั้งละหนึ่งขั้นเช่นเดียวกับอัลกอริทึม Bolot แต่อัลกอริทึม CNR จะนำค่า Reward ของแต่ละ Combination มาใช้ในการเลือกว่า Combination ที่เหมาะสมกับสภาพเครือข่ายคือ Combination ใด และสิ่งที่อัลกอริทึม CNR แตกต่างจากอัลกอริทึมอื่นอีกประการหนึ่งคือ ในอัลกอริทึมอื่นๆ ข้อมูลซ้ำมักจะใช้การบีบอัดเสียงที่มีอัตราบิตต่ำกว่าข้อมูลหลักเพื่อเป็นการประหยัดแบนด์วิดท์ แต่อัลกอริทึม CNR ใช้การบีบอัดเสียง G.723.1 ทั้งข้อมูลหลักและ

ข้อมูลซ้ำ เนื่องจากการบีบอัดเสียงแบบนี้มีอัตราบิตที่ต่ำอยู่แล้ว (6.3 kbps) และการที่ข้อมูลหลักและข้อมูลเสียงใช้การบีบอัดเสียงที่แตกต่างกันก็ทำให้ผู้ส่งต้องเสียเวลามากขึ้น เนื่องจากต้องมีการเข้ารหัสเสียงในแต่ละเฟรมมากกว่า 1 ครั้ง แต่ถ้าใช้การบีบอัดเสียงชนิดเดียวกันก็สามารถเข้ารหัสเพียงครั้งเดียวและเก็บข้อมูลหลังการบีบอัดเอาไว้เป็นข้อมูลซ้ำในภายหลังได้เลย



รูปที่ 3.10 การระบุค่า L_b และ L_s ในแพ็กเก็ต Receiver Report

สำหรับ Combination ของข้อมูลซ้ำที่ใช้ในอัลกอริทึม CNR มี 6 Combination ดังแสดงในตารางที่ 3.4 โดยในคอลัมน์ Combination Format ได้กำหนดสัญลักษณ์เป็นจำนวนเต็มลบเพื่อใช้แทนรูปแบบของข้อมูลซ้ำ ตัวอย่างเช่น สัญลักษณ์ -1 หมายความว่า ในแต่ละแพ็กเก็ตลำดับที่ N จะมีข้อมูลซ้ำ ของแพ็กเก็ตลำดับที่ N-1 อยู่ และสัญลักษณ์ -1-2 หมายความว่า ในแต่ละแพ็กเก็ต N จะมีข้อมูลซ้ำของแพ็กเก็ต N-1 และ N-2 อยู่ สำหรับสัญลักษณ์ของ Combination อื่นๆ ก็สามารถแปลความหมายได้ในทำนองเดียวกันนี้ ส่วนในคอลัมน์ Initial Reward เป็นค่าเริ่มต้นของ Reward ในแต่ละ Combination ซึ่งก็คือค่า Reward ที่ได้จากการทดลองของ Bolot นั่นเอง แต่ในอัลกอริทึม CNR นี้ค่า Reward จะไม่คงที่ โดยจะถูกแก้ไขให้เปลี่ยนแปลงไปตามสภาพของเครือข่าย

ตารางที่ 3.4 Combination ของข้อมูลซ้ำที่ใช้ในอัลกอริทึม CNR

No.	Combination Format	Initial Reward (Bolot Reward)	Bit Rate include Overhead (kbps)
0	-	1	17.1
1	-1	2.5	24.8
2	-2	6	24.8
3	-1-2	6	32.3
4	-1-3	10	32.3
5	-1-2-3	18	39.7

รูปที่ 3.11 เป็น Pseudo Code ของอัลกอริทึม CNR เมื่อผู้ส่งได้รับแพ็กเก็ต Receiver Report ก็จะอ่านค่า L_b และ L_a ในขั้นตอนแรกจะแก้ไขค่า Reward ของ Combination ปัจจุบัน ส่วนในขั้นตอนที่สองเป็นการนับจำนวนครั้งที่ค่า L_b มีค่าต่ำกว่าเทรสโฮลด์ LOW (เก็บไว้ในตัวแปร count_L_b_under_low) ซึ่งตัวแปรตัวนี้จะมีการตรวจสอบอีกครั้งเมื่อมีการตัดสินใจลด Combination ในขั้นตอนที่สามเป็นการตรวจสอบว่าถ้า L_a ที่อ่านจากแพ็กเก็ต Receiver Report มีค่าสูงกว่าเทรสโฮลด์ HIGH ก็จะต้องมีการเพิ่ม Combination ซึ่งทำได้โดยนำค่า L_b และค่า Reward ของ Combination ที่อยู่ในลำดับถัดจาก Combination ปัจจุบันเป็นต้นไปมาประมาณว่าถ้าเลือกใช้แต่ละ Combination แล้วจะได้ค่า L_a เท่าไร จากนั้นก็จะเลือก Combination แรกที่ให้ค่าประมาณของ L_a (ตัวแปร L_{ai} ใน Pseudo Code) ไม่เกินเทรสโฮลด์ HIGH ซึ่งการเพิ่ม Combination แบบนี้สามารถประหยัดเวลาได้มากกว่าการเพิ่ม Combination ครั้งละหนึ่งขั้น

ส่วนในกรณีที่ค่า L_a ที่อ่านมาจากแพ็กเก็ต Receiver Report มีค่าต่ำกว่าเทรสโฮลด์ LOW อัลกอริทึม CNR จะไม่ลด Combination ทันที แต่จะนับจำนวนครั้งที่ L_a มีค่าต่ำกว่าเทรสโฮลด์ LOW ในช่วงเวลาติดกัน (เก็บไว้ในตัวแปร count_L_a_under_low) การลด Combination ในอัลกอริทึมจะเกิดขึ้นได้ก็ต่อเมื่อค่า count_L_a_under_low หรือค่า count_L_b_under_low มีค่ามากกว่าหรือเท่ากับ MIN_UNDER_LOW โดยการที่ค่าของ count_L_a_under_low มีค่ามากกว่าหรือเท่ากับ MIN_UNDER_LOW หมายความว่า Combination ปัจจุบันที่ใช้อยู่สามารถทำให้ค่า L_a ต่ำกว่าเทรสโฮลด์ LOW เป็นระยะเวลานานพอแล้ว และปลอดภัยพอที่จะลองลด Combination ซึ่งเทคนิคนี้จะช่วยลดผลกระทบที่เกิดจากการแกว่งของค่า L_a ได้

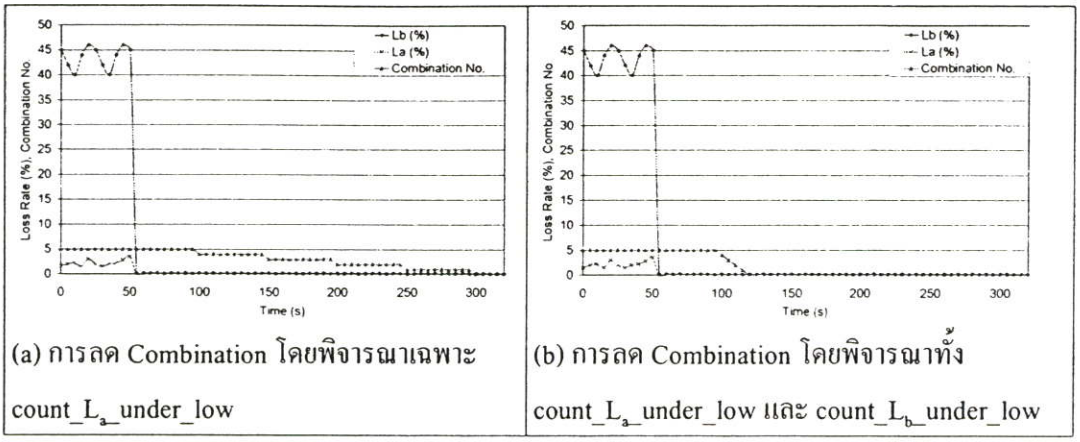
แต่การพิจารณาเฉพาะ count_L_a_under_low เพียงอย่างเดียวอาจจะทำให้การลด Combination ทำได้ช้าเกินไปในบางสถานการณ์ ตัวอย่างเช่น ในรูปที่ 3.12(a) หลังจากวินาทีที่ 50 เครือข่ายมีความคับคั่งต่ำมาก (L_b เกือบจะเป็น 0%) ซึ่งการใช้ Combination หมายเลข 0 (ไม่มีข้อมูล

ซ้ำเลย) ก็เพียงพอแล้ว แต่การลด Combination จากCombination หมายเลข 5 ไปเป็นหมายเลข 0 กลับต้องใช้เวลาถึง 250 วินาที เนื่องจากในที่นี้กำหนดค่า MIN_UNDER_LOW เท่ากับ 10 ซึ่งหมายความว่าการลด Combination จะเกิดขึ้นเมื่อ L_a หรือ L_b มีค่าต่ำกว่าเทรสโฮลด์ LOW อย่างน้อย 10 ครั้ง และแพ็กเก็ต Receiver Report จะมาถึงทุก 5 วินาที รวมแล้วต้องใช้เวลา 250 วินาที

```
For each receiver report packet received do
1.Update reward of the current combination
    //Let Rc stands for reward of the current combination
    Rc = Lb/La

2.if(Lb < LOW)
    increment count_Lb_under_low
  else
    count_Lb_under_low = 0

3.if(La > HIGH)
    selected_combination = MAX_COMBINATION;
    for(i = current_combination+1 to MAX_COMBINATION)
        Ri = reward of combination No.i
        Lai = Lb/Ri
        if(Lai <= HIGH) {
            selected_combination = i
            break;
    current_combination = selected_combination;
    count_La_under_low = 0

  else if(La < LOW)
    increment count_La_under_low

  else
    count_La_under_low = 0

4.if((count_La_under_low >= MIN_UNDER_LOW)OR
     (count_Lb_under_low >= MIN_UNDER_LOW))
    decrement current_combination
    count_La_under_low = 0
```

รูปที่ 3.11 Pseudo Code ของอัลกอริทึม CNR

ดังนั้นในอัลกอริทึม CNR จึงเพิ่มเงื่อนไขในการลด Combination อีกหนึ่งเงื่อนคือ ถ้าหากค่า count_L_b_under_low มีค่ามากกว่าหรือเท่ากับ MIN_UNDER_LOW แสดงว่าเครือข่ายมีความคับคั่งต่ำเป็นระยะเวลานานพอแล้ว และปลอดภัยพอที่จะลองลด Combination ได้ ซึ่งวิธีการนี้ช่วยให้การลด Combination ทำได้เร็วขึ้นในสภาพที่เครือข่ายมีความคับคั่งต่ำ จากรูปที่ 3.12(b) จะเห็นว่าการลด Combination จาก Combination หมายเลข 5 ไปเป็น Combination หมายเลข 0 ใช้เวลาเพียง 70 วินาที



(a) การลด Combination โดยพิจารณาเฉพาะ count_L_a_under_low

(b) การลด Combination โดยพิจารณาทั้ง count_L_a_under_low และ count_L_b_under_low

รูปที่ 3.12 การลด Combination โดยใช้ count_L_a_under_low อย่างเดียวอาจจะล่าช้าเกินไป

สำหรับค่าของ MIN_UNDER_LOW นั้น ถ้าหากมีค่ามากจะทำให้ป้องกันการแกว่งของค่า L_a ได้ดี แต่การลด Combination ทำได้ช้า และส่งผลให้มีปริมาณการใช้แบนด์วิดท์สูงเนื่องจากใช้ปริมาณข้อมูลซ้ำมาก ในที่นี้เลือกใช้ค่า MIN_UNDER_LOW เท่ากับ 10 ซึ่งมีค่าเท่ากับระยะเวลา 50 วินาทีเนื่องจากเป็นค่าที่มากพอที่จะลดปัญหาการแกว่งของค่า L_a ได้ดี

3.1.5 การประยุกต์ใช้การปรับให้เรียบในอัลกอริทึมควบคุมความผิดพลาดแบบปรับตัว

การปรับให้เรียบมีผลต่อค่า Reward value ของอัลกอริทึมแบบปรับตัว ในงานวิจัยได้ทำการปรับให้เรียบของอัตราการสูญหายข้อมูล เมื่อมีการปรับค่าอัลฟาที่เหมาะสม ทำให้อัตราการสูญหายของข้อมูลมีอัตราที่น้อยลง อันส่งผลให้คุณภาพของข้อมูลเสียงที่ได้รับอยู่ในเกณฑ์ที่ยอมรับได้

ในอัลกอริทึม Bolot และ RCCS ค่า reward value ถูกกำหนดไว้แล้ว ทั้งอัลกอริทึม Bolot และอัลกอริทึม RCCS ต่างก็มีวิธีการที่คล้ายคลึงกันก็คือ มีการสร้างตาราง Combination เอาไว้ก่อน สิ่งที่แตกต่างกันก็คือวิธีการที่จะเป็นตัวกำหนดว่าเมื่อใดควรจะเพิ่มหรือลด Combination ซึ่งอัลกอริทึม Bolot พิจารณาจากค่า L_a ส่วนอัลกอริทึม RCCS นั้นพิจารณาจากทั้ง L_a และค่าเวลาหน่วงระหว่างปลายทาง ส่วนอัลกอริทึม CNR มีการปรับค่าของ reward value ที่เป็นอัตราส่วนของ L_b ต่อ L_a ซึ่งสามารถนำการปรับค่าคงที่ให้เรียบมาพัฒนาต่อได้ เพื่อให้มีการปรับค่า reward value ให้เหมาะสมกับเครือข่ายในขณะนั้น ๆ ในวิทยานิพนธ์ฉบับนี้จึงนำการปรับให้เรียบมาใช้ในการปรับค่า reward ของอัลกอริทึม CNR แล้วเปรียบเทียบกับการใช้ค่า reward value แบบกำหนดค่าในอัลกอริทึม RCCS ซึ่งอัลกอริทึม RCCS มีจุดบกพร่องในส่วนของการเลือก Combination ซึ่งไม่ได้มีการแยกเงื่อนไขในการลดและเพิ่ม Combination จาก Pseudo Code ของอัลกอริทึม RCCS ในรูปที่ 3.8 จะเห็นว่าอัลกอริทึมนี้จะใช้ค่า Reward เพื่อค่า L_a' (ค่าทำนายของ L_a)ในแต่ละ Combination และจะเลือก Combination ที่มีค่า L_a' ที่มีค่าต่ำกว่าเทรสโฮลด์ HIGH และสูงกว่าเทรสโฮลด์ LOW

ซึ่งการใช้เงื่อนไขร่วมกันอย่างนี้อาจทำให้อัลกอริทึมนี้ไม่สามารถเพิ่ม Combination ได้ในบางครั้ง เช่น ถ้า Combination ในขั้นที่สูงกว่า Combination ปัจจุบันทุก Combination ต่างก็มีค่า L_a' ที่ต่ำกว่าเทรสโฮลด์ LOW ก็ถือว่าไม่ตรงตามเงื่อนไขแล้ว ซึ่งทำให้ไม่สามารถเพิ่ม Combination ได้ถึงแม้ว่าในขณะนั้นค่า L_a จะมีค่าสูงเกินกว่าเทรสโฮลด์ HIGH ก็ตาม การที่จะวัดว่าอัลกอริทึมใดดีกว่ากันต้องดูว่าอัลกอริทึมเหล่านี้สามารถเลือกใช้ Combination ที่สามารถรักษาระดับของ L_a ไม่ให้สูงเกินไปได้หรือไม่ รวมทั้งต้องไม่ใช้ปริมาณข้อมูลซ้ำมากเกินความจำเป็นอีกด้วย เพราะถ้าไม่คำนึงถึงปริมาณแบนด์วิดท์ที่เพิ่มขึ้นจากการใช้ข้อมูลซ้ำแล้ว ก็ไม่จำเป็นจะต้องใช้อัลกอริทึมใดเลยก็ได้ เพราะยิ่งใช้ข้อมูลซ้ำมากก็ยิ่งสามารถกู้ข้อมูลที่สูญหายได้มาก แต่การใช้อัลกอริทึมในการควบคุมความผิดพลาดเหล่านี้ก็เพื่อที่จะกำหนดให้ใช้ปริมาณข้อมูลซ้ำเท่าที่จำเป็นเท่านั้นและเพื่อให้ใช้แบนด์วิดท์ได้อย่างคุ้มค่า

อัลกอริทึม C NR ทำได้โดยเปลี่ยนค่า reward value จากสมการที่ 3.9 มาใช้สมการของการปรับให้เรียบดังสมการที่ 3.10 จะได้ค่า reward value ที่นำไปปรับค่า combination number ของแต่ละรอบการทำงานถัด ๆ ไปดังสมการ 3.11

$$R_c = L_b / L_a \tag{3.9}$$

จาก Smooth method ใช้สมการดังนี้

$$S_T = \alpha Y_T + (1-\alpha) S_{T-1} \tag{3.10}$$

ดังนั้นค่า Reward value ที่ใช้ดังสมการ 3.11

$$R_c = \alpha \left(L_b / L_a \right) + (1-\alpha) R_{c-1} \tag{3.11}$$

3.2 การตรวจสอบความถูกต้องของแบบจำลอง

เนื่องจากในการทดลองเพื่อประเมินผลอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวแต่ละอัลกอริทึมทำโดยการใช้แบบจำลอง ดังนั้นก่อนที่จะทำการทดลองจำเป็นที่ต้องแน่ใจเสียก่อนว่าแบบจำลองที่จะนำมาใช้ในการทดลองนั้นสามารถทำงานได้อย่างถูกต้อง ซึ่งการที่จะยืนยันว่าแบบจำลองที่ใช้ในการทดลองนั้นสามารถทำงานได้อย่างถูกต้องจำเป็นที่จะต้องมีการตรวจสอบสองขั้นตอน โดยขั้นตอนแรกคือตรวจสอบการคำนวณค่า L_b และ L_a ของผู้รับ และขั้นตอนที่สองคือตรวจสอบการทำงานของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวแต่ละอัลกอริทึม

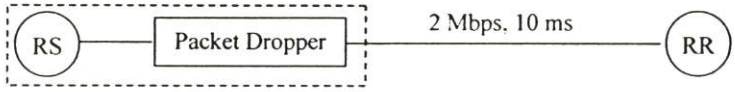
3.2.1 การตรวจสอบการคำนวณค่า L_b และ L_a ของผู้รับ

เนื่องจากว่าค่า L_b และค่า L_a เป็นค่าที่จำเป็นสำหรับอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวที่จะต้องนำไปใช้ในการตัดสินใจเลือก Combination ของข้อมูลซ้ำ ดังนั้นหากเกิดข้อผิดพลาดในการคำนวณค่าทั้งสอง จะส่งผลให้การทำงานในขั้นตอนต่อไปเกิดข้อผิดพลาดตาม

ไปด้วย

ในการทดลองนี้ได้กำหนดรูปแบบการเชื่อมต่อภายในเครือข่ายดังรูปที่ 3.13 ซึ่งจะเห็นว่ามีโหนดเพียง 2 โหนดเท่านั้น โดยโหนด RS จะส่งแพ็กเก็ตเสียงไปให้โหนด RR แต่ที่โหนด RS จะมีตัวตัดทิ้งแพ็กเก็ต (Packet Dropper) ที่สามารถตัดทิ้งแพ็กเก็ตได้ตามรูปแบบที่กำหนด ทั้งนี้เพื่อใช้ในการทดสอบว่าการทำงานของแบบจำลองในส่วนของการคำนวณค่า L_a และ L_b ทำได้ถูกต้องหรือไม่ โดยได้กำหนดรูปแบบการตัดทิ้ง (Drop) แพ็กเก็ตทั้งหมด 5 รูปแบบดังตารางที่ 3.5 ซึ่งจากรูปแบบการตัดทิ้งแพ็กเก็ตดังกล่าวสามารถที่จะคำนวณค่า L_b และ L_a ที่ถูกต้องได้เลย ซึ่งหากแบบจำลองสามารถทำงานได้อย่างถูกต้อง ค่า L_b และ L_a ที่ได้จากการทดลองจะต้องมีค่าตรงกัน โดยในการตัดทิ้งแพ็กเก็ตแต่ละรูปแบบจะมีการจำลอง 6 ครั้ง (สำหรับ 6 Combination) และในการจำลองแต่ละครั้ง พฤติกรรมในการส่งแพ็กเก็ตของโหนด RS และโหนด RR เป็นดังนี้

- โหนด RS ส่งแพ็กเก็ตเสียงโดยใช้โปรโตคอล RTP ไปให้โหนด RR โดยส่งแพ็กเก็ตทุกๆ 30 ms ซึ่งเป็นระยะเวลาของเสียงพูด 1 เฟรมสำหรับการบีบอัดเสียงด้วย G.723.1 ส่วนขนาดแพ็กเก็ตขึ้นอยู่กับ Combination ที่ใช้
- โหนด RR จะมีการรายงานค่า L_b และ L_a โดยส่งมาในแพ็กเก็ต Receiver Report ให้กับโหนด RS ซึ่งแพ็กเก็ตนี้มีการส่งทุก 5 วินาที
- โหนด RS เริ่มส่งแพ็กเก็ตเสียงตั้งแต่วินาทีที่ 0 และหยุดที่วินาทีที่ 500
- ค่า L_b และ L_a ที่ได้จากการทดลองเป็นค่าเฉลี่ยของค่า L_b และ L_a ที่ RR รายงานให้กับ RS ทุกครั้ง (100 ครั้ง)



รูปที่ 3.13 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง

ตารางที่ 3.5 เป็นผลการตรวจสอบความถูกต้องในการคำนวณค่าของ L_b ของผู้รับ เนื่องจากว่าค่า L_b คืออัตราการสูญหายของข้อมูลเสียงก่อนการจัดเรียง ซึ่งก็คืออัตราการสูญหายของแพ็กเก็ตนั่นเอง ดังนั้นในกรณีนี้ค่า L_b ที่ถูกต้องจะต้องมีค่าเท่ากับอัตราการตัดทิ้งแพ็กเก็ต เนื่องจากในการทดลองการสูญหายของแพ็กเก็ตเกิดจากการตัดทิ้งเพียงอย่างเดียวเท่านั้น และไม่ว่าจะใช้จำนวนข้อมูลซ้ำเท่าใดก็จะไม่มีผลต่อค่า L_b เพราะว่าเป็นการพิจารณาในขณะที่ยังไม่มีการนำข้อมูลซ้ำมากู้ข้อมูลในแพ็กเก็ตที่สูญหาย จากผลการทดลองในตารางที่ 3.5 จะเห็นว่าการคำนวณค่า L_b ของผู้รับสามารถทำได้ถูกต้องในทุกกรณี นั่นคือค่า L_b มีค่าเท่ากับอัตราการตัดทิ้งแพ็กเก็ตเสมอไม่ว่าจะใช้รูปแบบการตัดทิ้งแพ็กเก็ตแบบใด

ตารางที่ 3.5 การตรวจสอบความถูกต้องในการคำนวณค่า L_b ของผู้รับในแบบจำลอง

รูปแบบในการตัดทิ้ง	หมายเลขแพ็กเก็ตที่ถูกตัดทิ้ง (ในทุก 100 แพ็กเก็ต)	อัตราการตัดทิ้งแพ็กเก็ต (%)	ค่า L_b (%) จากการทดลองของแต่ละ Combination					
			0	1	2	3	4	5
D01	10N; N = 1, 2, …,10	10	10	10	10	10	10	10
D02	10N-2, 10N-1, 10N; N = 1, 2,…,10	30	30	30	30	30	30	30
D03	10N-3, 10N-2, 10N; N = 1, 2,…,10	30	30	30	30	30	30	30
D04	10N-2, 10N-1, 10N; N = 1, 2,…,10 10N-3; N=1,2,3	33	33	33	33	33	33	33
D05	10N-1, 10N; N = 1, 2,…,10 10N-2; N=1,2,3,4	24	24	24	24	24	24	24

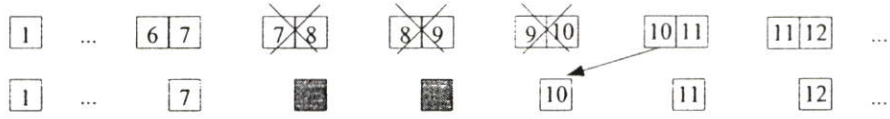
ตารางที่ 3.6 แสดงผลการตรวจสอบความถูกต้องในการคำนวณค่า L_a ของผู้รับในแบบจำลอง ซึ่งมีการตัดทิ้งแพ็กเก็ตตามรูปแบบเดียวกันกับที่ใช้ในตารางที่ 3.5 ซึ่งการตัดทิ้งแพ็กเก็ตทั้ง 5 รูปแบบสามารถคำนวณได้ล่วงหน้าแล้วว่าค่า L_a ที่ถูกต้องของแต่ละ Combination มีค่าเท่าใด ดังนั้นหากแบบจำลองสามารถทำงานได้ถูกต้อง ค่า L_a ที่จากการทดลองจะต้องมีค่าตรงกันกับค่าที่ถูกต้อง

รูปที่ 3.14 เป็นตัวอย่างของการหาค่า L_a เมื่อกำหนดให้มีการตัดทิ้งแพ็กเก็ตแบบ D02 ซึ่งจากตารางที่ 3.6 จะเห็นว่าการตัดทิ้งแพ็กเก็ตแบบนี้ ในทุก 100 แพ็กเก็ตจะมีการตัดทิ้งแพ็กเก็ตเป็นจำนวน 30 แพ็กเก็ต (แพ็กเก็ตในหมายเลขที่ลงท้ายด้วย 8, 9 และ 0) หรืออาจจะกล่าวได้ว่าในทุก 10 แพ็กเก็ตจะมีการตัดทิ้ง 3 แพ็กเก็ต ดังนั้นในการหาค่า L_a ที่ถูกต้องก็สามารถพิจารณาเฉพาะ 10 แพ็กเก็ต และหาว่าหลังจากมีการกู้ข้อมูลในแพ็กเก็ตที่สูญหายแล้วมีข้อมูลเสียงสูญหายไปกี่เฟรมก็สามารถหาค่า L_a ได้แล้ว ซึ่งจะเห็นได้ว่าค่า L_a จากการทดลองของ Combination 0, 1 และ 2 ที่ได้จากการทดลองมีค่าถูกต้อง สำหรับในกรณีอื่นๆ หลังจากที่ได้มีการตรวจสอบแล้วพบว่าการคำนวณค่า L_a ของแบบจำลองสามารถทำงานได้ถูกต้องในทุกกรณี

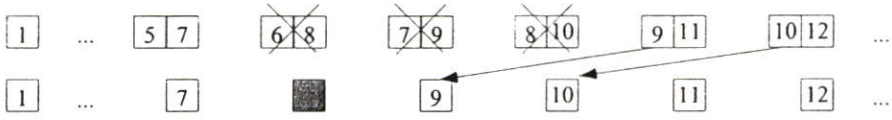
ตารางที่ 3.6 การตรวจสอบความถูกต้องในการคำนวณค่า L_a ของผู้รับในแบบจำลอง

รูปแบบในการตัดทิ้ง	หมายเลขแพ็กเก็ตที่ถูกตัดทิ้ง (ในทุก 100 แพ็กเก็ต)	อัตราการตัดทิ้งแพ็กเก็ต (%)	ค่า L_a (%) จากการทดลองของแต่ละ Combination					
			0	1	2	3	4	5
D01	10N; N = 1, 2, …,10	10	10	0	0	0	0	0
D02	10N-2, 10N-1, 10N; N = 1, 2,…,10	30	30	20	10	10	0	0
D03	10N-3, 10N-2, 10N; N = 1, 2,…,10	30	30	10	10	0	10	0
D04	10N-2, 10N-1, 10N; N = 1, 2,…,10 10N-3; N=1,2,3	33	33	23	13	13	3	3
D05	10N-1, 10N; N = 1, 2,…,10 10N-2; N=1,2,3,4	24	24	14	4	4	0	0

(a) Combination 0, ทุก 10 แพ็กเก็ตจะมีข้อมูลเสียงสูญหาย 3 เฟรม, L_a = 30%



(b) Combination 1, ทุก 10 แพ็กเก็ตจะมีข้อมูลเสียงสูญหาย 2 เฟรม, L_a = 20%



(c) Combination 2, ทุก 10 แพ็กเก็ตจะมีข้อมูลเสียงสูญหาย 1 เฟรม, L_a = 10%

- **รูปที่ 3.14** ตัวอย่างการหาค่า L_a เมื่อกำหนดรูปแบบการตัดทิ้งแพ็กเก็ตแบบ D02

3.2.2 การตรวจสอบการทำงานของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัว

เมื่อแน่ใจแล้วว่าการคำนวณค่า L_b และ L_a ในแบบจำลองสามารถทำได้อย่างถูกต้อง ขั้นตอนต่อไปคือการตรวจสอบการทำงานของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวของ 2 อัลกอริทึมคือ อัลกอริทึม RCCS ซึ่งมีการใช้การปรับให้เรียบอยู่แล้วและอัลกอริทึม CNR ซึ่งการทดลองในหัวข้อนี้ได้กำหนดให้ใช้เครือข่ายในการจำลองดังรูปที่ 3.13 โดยมีการจำลอง 4

ครั้ง (สำหรับ 4 Combination) และเพื่อที่จะตรวจสอบว่าการทำงานของแบบจำลองในส่วนของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวว่าสามารถทำงานได้อย่างถูกต้องตามวิธีการที่ได้อธิบายในหัวข้อ 3.1 หรือไม่ จึงได้กำหนดรูปแบบการตัดทิ้งแพ็กเก็ตในแต่ละช่วงเวลาให้แตกต่างกัน เพื่อจะได้เห็นว่าการเปลี่ยนแปลง Combination ของข้อมูลซ้ำเป็นไปอย่างถูกต้องหรือไม่ และในการจำลองแต่ละครั้งพฤติกรรมการส่งข้อมูลของโหนด RS และ RR เป็นดังนี้

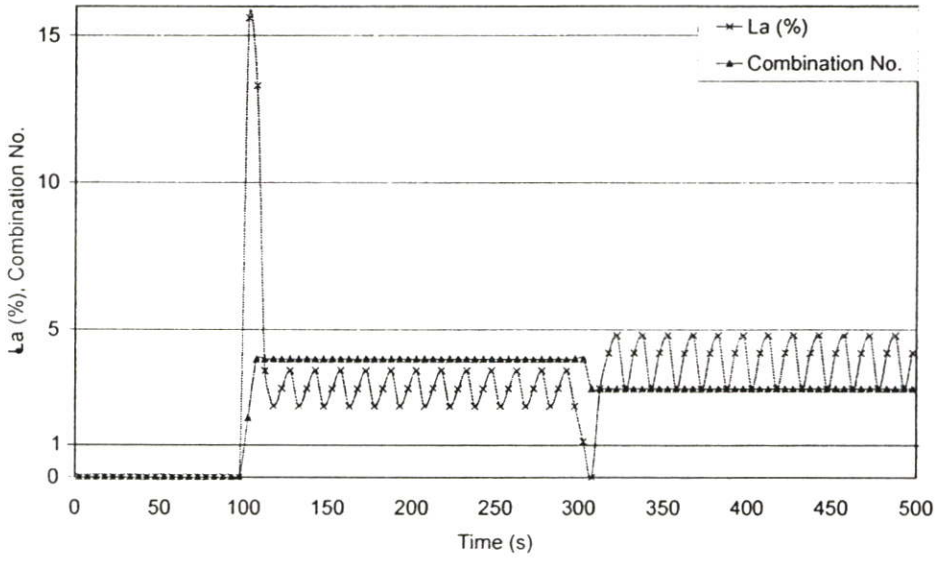
- โหนด RS ส่งแพ็กเก็ตเสียงโดยใช้โปรโตคอล RTP ไปให้โหนด RR โดยส่งแพ็กเก็ตทุกๆ 30 ms ซึ่งเป็นระยะเวลาของเสียงพูด 1 เฟรมสำหรับการบีบอัดเสียงด้วย G.723.1 ส่วนขนาดแพ็กเก็ตขึ้นอยู่กับ Combination ที่ใช้
- โหนด RR จะมีการรายงานค่า L_b และ L_s โดยส่งมาในแพ็กเก็ต Receiver Report ให้กับโหนด RS ซึ่งแพ็กเก็ตนี้มีการส่งทุก 5 วินาที
- โหนด RS เริ่มส่งแพ็กเก็ตเสียงตั้งแต่วินาทีที่ 0 และหยุดที่วินาทีที่ 500
- ในช่วงวินาทีที่ 0 – 100 ไม่มีการตัดทิ้งแพ็กเก็ต
- ในช่วงวินาทีที่ 101 – 300 มีการตัดทิ้งแพ็กเก็ตแบบ D04 (ดูรายละเอียดในตารางที่ 3.5)
- ในช่วงวินาทีที่ 301 – 500 มีการตัดทิ้งแพ็กเก็ตแบบ D05 (ดูรายละเอียดในตารางที่ 3.5)

การทดลองนี้กำหนดให้ทุก Combination ใช้การบีบอัดเสียง G.723.1 ทั้งข้อมูลหลักและข้อมูลซ้ำ เนื่องจากในวิทยานิพนธ์นี้เลือกใช้การบีบอัดเสียง G.723.1 ในการสื่อสารเสียง ประกอบกับการบีบอัดเสียงแบบนี้มีอัตราบิตที่ต่ำอยู่แล้ว (6.3 kbps) และการที่ข้อมูลหลักและข้อมูลเสียงใช้การบีบอัดเสียงที่แตกต่างกันก็ทำให้ผู้ส่งต้องเสียเวลามากขึ้น เนื่องจากต้องมีการเข้ารหัสเสียงในแต่ละเฟรมมากกว่า 1 ครั้ง แต่ถ้าใช้การบีบอัดเสียงชนิดเดียวกันก็สามารถเข้ารหัสเพียงครั้งเดียวและเก็บข้อมูลหลังการบีบอัดเอาไว้เป็นข้อมูลซ้ำในภายหลังได้เลย สำหรับ Combination ของข้อมูลซ้ำที่ใช้ในการทดลองมี 6 Combination ดังแสดงในตารางที่ 3.4 ส่วนค่าเทรสโฮลด์ที่ใช้ในทุกอัลกอริทึมนั้นกำหนดให้เท่ากัน โดยค่าเทรสโฮลด์ HIGH เท่ากับ 5% เนื่องจากว่าหากมีข้อมูลเสียงสูญหายมากกว่า 5% จะทำให้รู้สึกได้ว่าเสียงมีคุณภาพลดลง[28] และกำหนดค่าเทรสโฮลด์ LOW เท่ากับ 1% ซึ่งถือว่าเป็นค่าที่ต่ำพอที่จะลองเพิ่ม Combination ได้แล้ว

3.2.2.1 การตรวจสอบการทำงานของอัลกอริทึม RCCS

อัลกอริทึม RCCS มีการใช้ทั้งค่า L_s และค่าเวลาหน่วงระหว่างปลายทางในการเลือก Combination จาก Pseudo Code ของอัลกอริทึม RCCS ในรูปที่ 3.8 จะเห็นว่าอัลกอริทึมนี้จะพิจารณาตาราง Combination ตั้งแต่ Combination แรกจนถึง Combination สุดท้าย และจะเลือก Combination แรกที่มีค่า L_s' (ค่าทำนายของ L_s ซึ่งได้จากการนำ L_b หารด้วยค่า Reward) และค่า D_s' (ค่าทำนายของเวลาหน่วงระหว่างปลายทางหลังการจัดเรียงข้อมูลซ้ำ) ไม่เกินค่าที่กำหนด แต่เนื่องจากว่าการหาค่าเวลาหน่วงระหว่างปลายทางนั้นทำได้ยากเพราะนาฬิกาของผู้ส่งกับผู้รับนั้นไม่

ประสานกัน อีกทั้งใน [19] ยังอธิบายวิธีการหาค่าเวลาระหว่างปลายทางได้ไม่ชัดเจนพอที่จะนำมาสร้างแบบจำลองได้ ดังนั้นในการเลือก Combination ของอัลกอริทึม RCCS ในแบบจำลองที่สร้างขึ้น จึงตัดเงื่อนไขของการตรวจสอบค่าเวลาหน่วงระหว่างปลายทางออกไป โดยถือว่าเวลาหน่วงที่เกิดขึ้นจากการใช้ Combination ในตารางที่ 3.4 ถือว่าเป็นค่าที่ยอมรับได้ทั้งหมด อย่างไรก็ตามการตัดเงื่อนไขของการตรวจสอบค่าเวลาหน่วงระหว่างปลายทางก็ไม่ได้ทำให้อัลกอริทึมนี้เสียเปรียบอัลกอริทึมอื่น เนื่องจากว่าหากยังมีการตรวจสอบค่าเวลาหน่วงระหว่างปลายทางอยู่ ในบางครั้งอัลกอริทึมนี้อาจจะไม่สามารถเพิ่ม Combination ได้เนื่องจากค่าเวลาหน่วงสูงเกินค่าเทรสโฮลด์ ทำให้ไม่สามารถกู้ข้อมูลที่สูญหายได้เพียงพอ แต่หากไม่มีการตรวจสอบค่าเวลาหน่วงระหว่างปลายทาง สามารถเพิ่ม Combination ได้จนถึง Combination สูงสุด และสำหรับค่า α ซึ่งเป็นพารามิเตอร์ของสมการตัวกรองในการแก้ไขค่า Reward ของอัลกอริทึมนี้ใช้ค่าตามที่กำหนดเอาไว้ใน [19] คือ 0.98

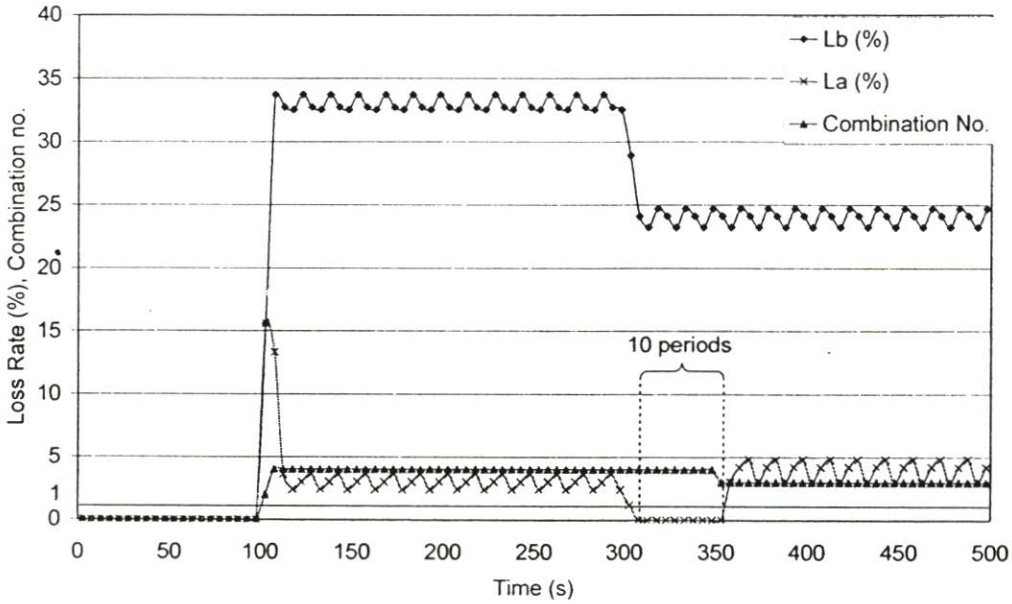


รูปที่ 3.15 การเปลี่ยนแปลงของ Combination ในระหว่างการจำลองโดยใช้อัลกอริทึม RCCS

จากการตรวจสอบการทำงานของแบบจำลองเมื่อใช้อัลกอริทึม RCCS ปรากฏว่าแบบจำลองสามารถทำงานได้ถูกต้อง ซึ่งจะเห็นได้จากกราฟในรูปที่ 3.15 โดยหลังจากวินาทีที่ 100 ซึ่งเริ่มมีการตัดทิ้งแพ็กเก็ตแบบ D04 (ดูรายละเอียดได้ในตารางที่ 3.5) ซึ่งมีการตัดทิ้งแพ็กเก็ต 33% ค่า L_a ก็เพิ่มขึ้นมากกว่า 5 % อัลกอริทึม RCCS จึงได้มีการเปลี่ยน Combination ไปเป็น Combination หมายเลข 4 และหลังจากนั้นก็ไม่มีการเปลี่ยน Combination เนื่องจากค่า L_a มีค่าต่ำกว่า 5% และหลังจากวินาทีที่ 300 ซึ่งได้มีการลดการตัดทิ้งแพ็กเก็ตลงโดยใช้การตัดทิ้งแบบ D05 ซึ่งมีอัตราการตัดทิ้งแพ็กเก็ต 24% ค่า L_a จึงมีค่าลดลง และเมื่อค่า L_a ต่ำกว่า 1% อัลกอริทึม RCCS ก็ได้มีการเปลี่ยน Combination อีกครั้ง โดยลด Combination ลงมาอยู่ที่ Combination หมายเลข 3

3.2.2.2 การตรวจสอบการทำงานของอัลกอริทึม CNR

จากการตรวจสอบการทำงานของแบบจำลองเมื่อมีการควบคุมความผิดพลาดแบบปรับตัว โดยใช้อัลกอริทึม CNR การเปลี่ยนแปลง Combination เป็นไปตามกราฟในรูปที่ 3.16 ซึ่งจะเห็นว่าแบบจำลองสามารถทำงานได้ถูกต้องตาม Pseudo Code ของอัลกอริทึม CNR ในรูปที่ 3.11



รูปที่ 3.16 การเปลี่ยนแปลงของ Combination ในระหว่างการจำลองโดยใช้อัลกอริทึม CNR

โดยหลังจากวินาทีที่ 100 ซึ่งเริ่มมีการตัดทิ้งแพ็กเก็ต ได้มีการเพิ่ม Combination ไปหยุดที่ Combination หมายเลข 4 เนื่องจากค่า L_a มีค่าต่ำกว่า 5% แล้ว และหลังจากวินาทีที่ 300 ซึ่งได้มีการลดการตัดทิ้งแพ็กเก็ตลง ค่า L_a จึงมีค่าลดลง และเมื่อค่า L_a ต่ำกว่า 1% เป็นจำนวน 10 ครั้ง ก็ได้มีการลด Combination ลงมาอยู่ที่ Combination หมายเลข 3 ทั้งนี้เนื่องจากในที่นี้ได้กำหนดค่า MIN_DURATION เท่ากับ 10

บทที่ 4
ผลการจำลองการทำงานของระบบ

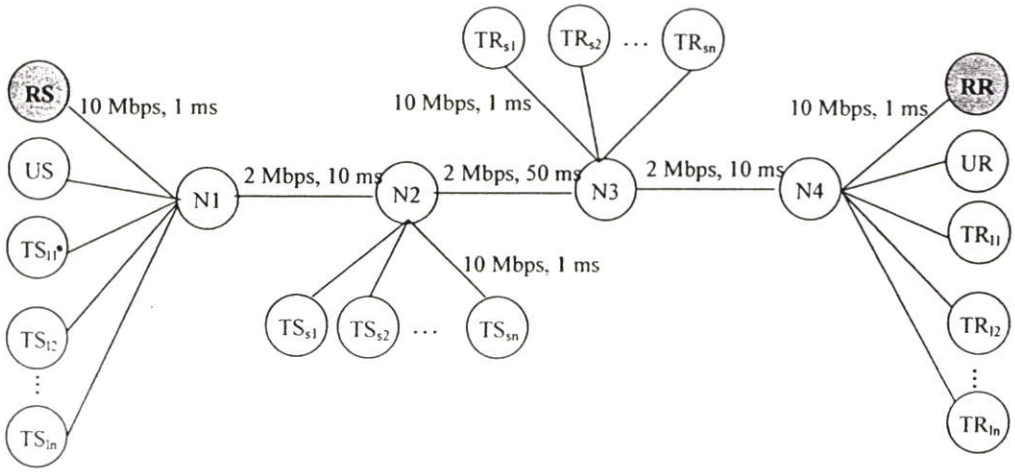
จากการศึกษาวิธีการควบความผิดพลาดแบบปรับตัว โดยมีการนำเทคนิคการทำให้เรียบแบบเอกซโพเนนเชียลมาใช้ในการวิเคราะห์อนุกรมเวลาที่มีข้อมูลผิดปกติในการควบคุมความผิดพลาดเสียงตามวิธีการที่ได้เสนอขึ้นในบทที่ 3 ในบทนี้จะทำการประเมินผลอัลกอริทึม CNR โดยมีการเปรียบเทียบกับอัลกอริทึม CNR เดิม และ เปรียบเทียบกับอัลกอริทึม RCCS ซึ่งมีการใช้สมการของการปรับให้เรียบอยู่แล้ว

4.1 การจำลองเครือข่ายเพื่อส่งผ่านข้อมูล

การจำลองเครือข่ายเพื่อส่งผ่านข้อมูลเสียงในกรณีที่ใช้การบีบอัดเสียง G.723.1 ทั้งข้อมูลหลักและข้อมูลซ้ำ สำหรับ Combination ของข้อมูลซ้ำที่ใช้ในการทดลองมี 6 Combination ดังแสดงในตารางที่ 3.4 และเนื่องจากว่าแต่ละอัลกอริทึมต่างก็มีพิจารณาค่าเทรสโฮลด์ HIGH และ LOW ในการเลือก Combination ดังนั้นในการเปรียบเทียบสมรรถนะของแต่ละอัลกอริทึมนั้น จำเป็นจะต้องกำหนดค่าเทรสโฮลด์ให้เท่ากัน ใน [28] ได้ระบุเอาไว้ว่าหากมีข้อมูลเสียงสูญหายมากกว่า 5 % จะทำให้รู้สึกได้ว่าเสียงมีคุณภาพลดลง ดังนั้นการทดลองนี้จึงเลือกที่จะใช้ค่าเทรสโฮลด์ HIGH เท่ากับ 5% และเพื่อให้ผลการทดลองมีความน่าเชื่อถือมากขึ้น ในการทดลองนี้[28]จึงได้มีการจำลองในกรณีที่ใช้ค่าเทรสโฮลด์ HIGH เท่ากับ 3% ด้วย ซึ่งกรณีนี้ก็อาจจะเป็นไปได้ในการใช้งานจริง เช่น ผู้ใช้ต้องการคุณภาพเสียงที่ดีมากโดยยอมรับการสูญหายของข้อมูลเสียงได้ไม่เกิน 3% เป็นต้น ส่วนค่าเทรสโฮลด์อีกค่าหนึ่งคือค่าเทรสโฮลด์ LOW กำหนดให้มีค่าเท่ากับ 1% ซึ่งค่าของเทรสโฮลด์ LOW นี้ไม่ควรกำหนดให้มีค่าใกล้เคียงกับค่าเทรสโฮลด์ HIGH มากนัก เพื่อป้องกันไม่ให้เกิดการเปลี่ยน Combination บ่อยเกินไป

ในการทดลองนี้ไม่ได้มีการส่งข้อมูลในเครือข่ายจริง แต่เป็นการจำลองบน NS-2[36] ในรูปที่ 4.1 แสดงการเชื่อมต่อของแต่ละโหนดที่อยู่ภายในเครือข่ายที่ใช้ในการจำลอง ลิงค์ระหว่างโหนด N2 และ N3 มีแบนด์วิดท์ 2 เมกะบิตต่อวินาที (Mbps) เวลาหน่วง 50 มิลลิวินาที (ms) ซึ่งลิงค์นี้ถือเป็นลิงค์คอขวดของระบบ ส่วนลิงค์ระหว่างโหนด N1 กับ N2 และลิงค์ระหว่างโหนด N3 กับ N4 มีแบนด์วิดท์ 2 เมกะบิตต่อวินาที เวลาหน่วง 10 มิลลิวินาที ส่วนลิงค์อื่นที่เหลือมีความจุ 10 เมกะบิตต่อวินาที เวลาหน่วง 1 มิลลิวินาที และลิงค์ทุกลิงค์เป็นลิงค์แบบ Drop Tail โดยโหนด RS เป็นโหนดที่ส่งแพ็กเก็ตของเสียง และมีการกำหนด Combination ของข้อมูลซ้ำตามอัลกอริทึมที่เลือกใช้ นอกจากนี้ยังมีโหนดที่คอยส่งทราฟฟิคชนิดอื่นๆ เพื่อให้เครือข่ายมีสภาพแวดล้อมเหมือนมีผู้ใช้งานคนอื่นกำลังใช้เครือข่ายอยู่ด้วย โดยมีทั้งผู้ใช้ที่ส่งทราฟฟิค UDP และผู้ใช้ที่ส่งทราฟฟิค TCP

ในที่นี้ได้แบ่งการทดลองออกเป็น 2 ตอน โดยสิ่งที่เหมือนกันคือก็คือ พฤติกรรมในการส่งข้อมูลของโหนดที่ส่งทราฟฟิค TCP จะมีการส่งข้อมูลติดต่อกันเป็นเวลานานโดยไม่หยุด แต่ตำแหน่งที่ตั้งของโหนดที่ส่งทราฟฟิค TCP จะแตกต่างกัน โดยกำหนดให้การจำลองในแต่ละอัลกอริทึมมีการรันแบบจำลอง 2 ครั้ง โดยครั้งแรกใช้จำนวนโหนด TCP เท่ากับ 20 โหนด (TS_{l1}-TS_{l10} และ TS_{s1}-TS_{s10}) และครั้งที่สองใช้จำนวนโหนด TCP เป็น 40 โหนด (TS_{l1}-TS_{l20} และ TS_{s1}-TS_{s20}) ทั้งนี้เพื่อให้จะได้พิจารณาผลการทดลองว่าในกรณีที่มีจำนวนโหนด TCP ต่างกัน ผลการทดลองจะเป็นไปในทำนองเดียวกันหรือไม่ และทำการวัดค่า average through, average delay time ในค่าคงที่ปรับให้เรียบที่แตกต่างกัน ทั้งนี้เพื่อที่จะประเมินสมรรถนะของอัลกอริทึม CNR ในสภาพเครือข่ายที่แตกต่างกัน เพื่อเพิ่มความน่าเชื่อถือของผลการทดลอง



รูปที่ 4.1 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง

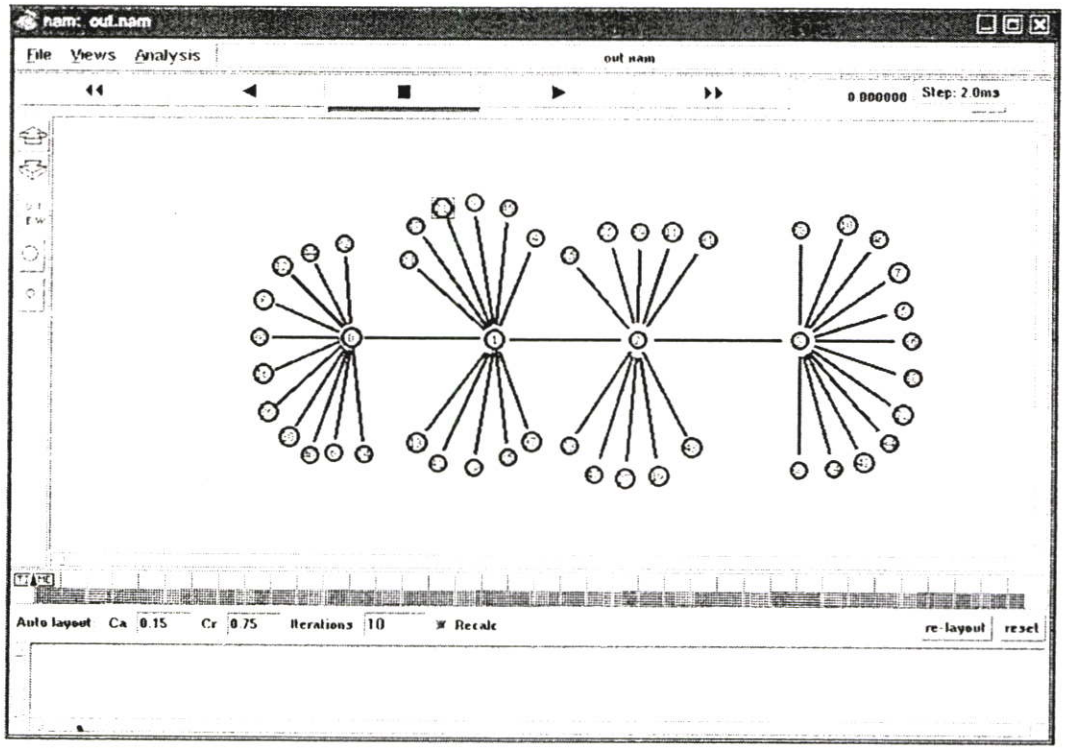
4.1.1 การจำลองเครือข่ายให้มีความหนาแน่นของทราฟฟิคต่ำ

ในการทดลองตอนที่ 1 เป็นการจำลองให้มีการสื่อสารเสียงในเครือข่ายที่มีผู้ที่ส่งทราฟฟิค TCP เป็นช่วงเวลานาน และมีปริมาณการใช้แบนด์วิดท์เต็มความจุของเครือข่าย ซึ่งในสภาพเครือข่ายเช่นนี้สามารถทำให้เกิดการแกว่งของค่าอัตราการสูญหายของแพ็กเก็ตในการสื่อสารเสียงได้ โดยมีจุดประสงค์เพื่อประเมินสมรรถนะของแต่ละอัลกอริทึมว่าจะได้รับผลจากปัญหาการแกว่งของค่าอัตราการสูญหายของแพ็กเก็ตมากน้อยเพียงใด และในกรณีที่สภาพเครือข่ายเป็นเช่นนี้อัลกอริทึมใดสามารถทำงานได้ผลดีที่สุด ซึ่งการส่งข้อมูลของแต่ละโหนดเป็นดังนี้

- โหนด RS ส่งเสียงโดยใช้โปรโตคอล RTP ไปให้โหนด RR โดยส่งแพ็กเก็ตทุกๆ 30 มิลลิวินาที ซึ่งระยะเวลาเท่ากับขนาดเฟรมของ G.723.1 ส่วนขนาดแพ็กเก็ตขึ้นอยู่กับ Combination ที่ใช้ ถ้า Combination ใดมีจำนวนข้อมูลซ้ำมาก แพ็กเก็ตเสียงที่ส่งก็จะมีขนาดใหญ่

- โหนด RR ส่งแพ็กเก็ต Receiver Report ให้กับโหนด RS และเนื่องจากในที่นี้เป็นการติดต่อแบบจุดต่อจุด ดังนั้นแพ็กเก็ต Receiver Report นี้จะมีการส่งทุกๆ 5 วินาที
- โหนด US ส่งทราฟฟิค UDP ด้วยอัตราบิต 400 กิโลบิตต่อวินาที โดยใช้แพ็กเก็ตขนาด 1024 ไบต์ ซึ่งโหนด US ทำหน้าที่เป็นตัวแทนของโปรแกรมประยุกต์ในการส่งเสียงหรือวีดิทัศน์ที่ไม่มีการปรับตัวตามสภาพเครือข่าย
- โหนด TS_{11} – TS_{1n} และ TS_{s1} - TS_{sn} ส่งไฟล์โดยใช้โปรโตคอล FTP ไปให้กับโหนด TR_{11} – TR_{1n} และ TR_{s1} - TR_{sn} ตาลำดับ โดยเริ่มส่งที่วินาทีที่ 100 และหยุดส่งเมื่อการจำลองยุติ ดังนั้นในช่วง 100 วินาทีแรกจะไม่มีแพ็กเก็ตเสียงสูญหายเลย
- การจำลองยุติที่วินาทีที่ 1200

โดยในการทดลองตอนนี้กำหนดให้การจำลองในแต่ละอัลกอริทึมมีการรันแบบจำลอง 2 ครั้ง โดยครั้งแรกใช้จำนวนโหนด TCP เท่ากับ 20 โหนด (TS_{11}-TS_{110} และ TS_{s1}-TS_{s10}) และครั้งที่สองใช้จำนวนโหนด TCP เป็น 40 โหนด (TS_{11}-TS_{120} และ TS_{s1}-TS_{s20}) ทั้งนี้เพื่อให้จะได้พิจารณาผลการทดลองว่าในกรณีที่มีจำนวนโหนด TCP ต่างกัน ผลการทดลองจะเป็นไปในทำนองเดียวกันหรือไม่

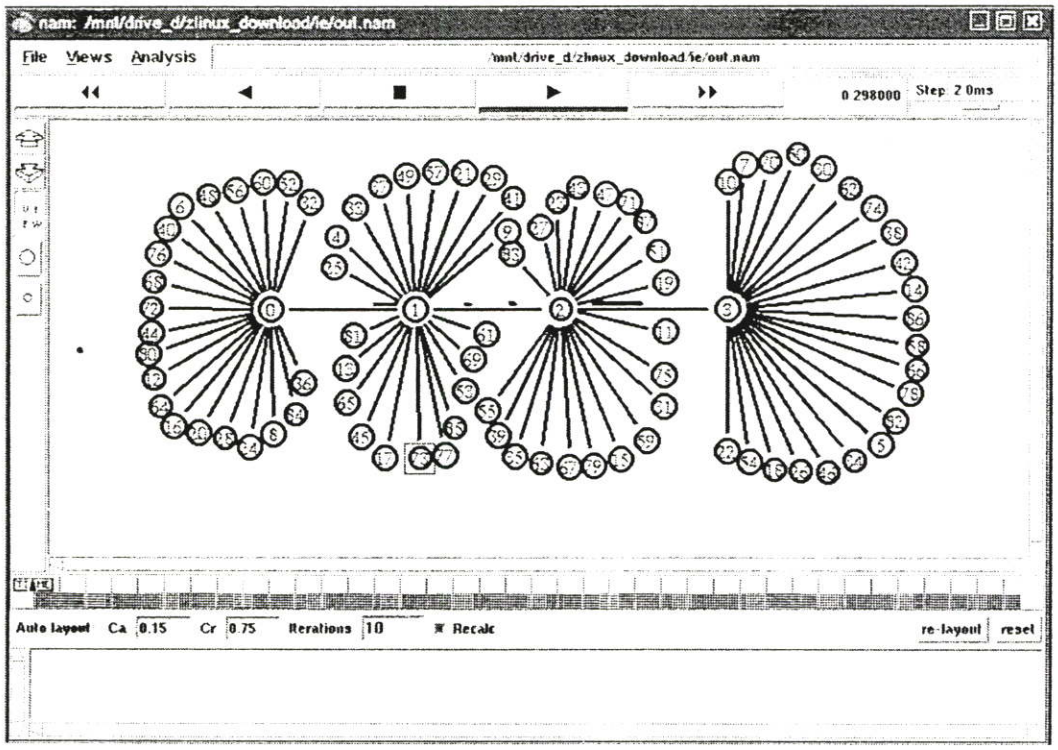


รูปที่ 4.2 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง

4.1.2 การจำลองเครือข่ายให้มีความหนาแน่นของทราฟฟิคสูง

ในการทดลองตอนที่ 2 นี้ ลักษณะการส่งข้อมูลของโหนด RS และ US ยังเหมือนกับในการ

ทดลองตอนที่ 1 โดยใช้จำนวนโหนด TCP เป็น 40 โหนด



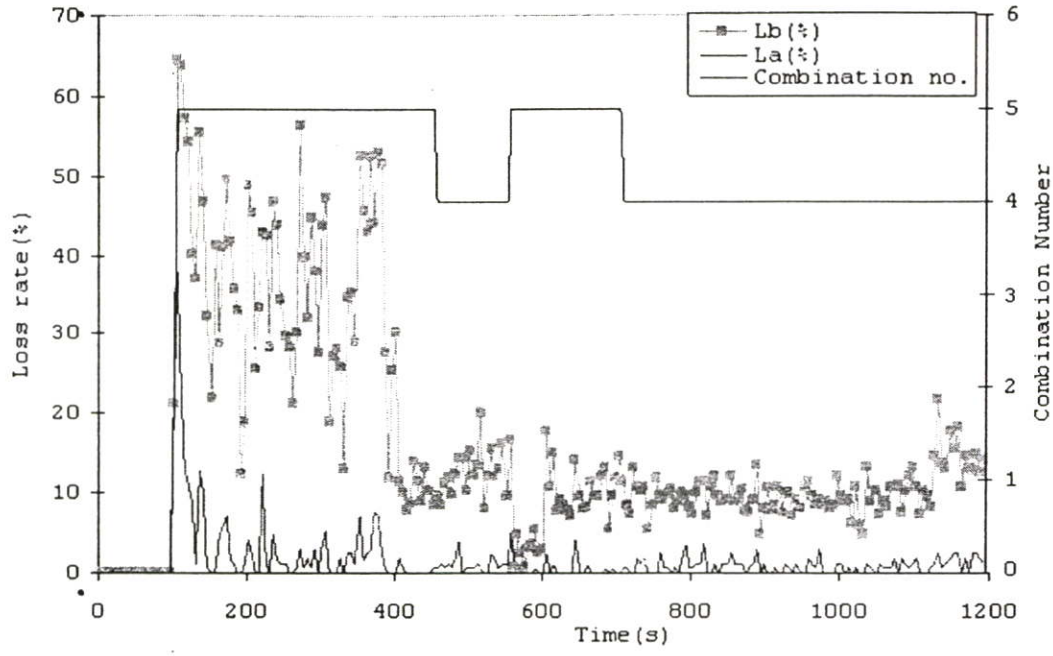
รูปที่ 4.3 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง

4.2 การประเมินผลการทำงาน

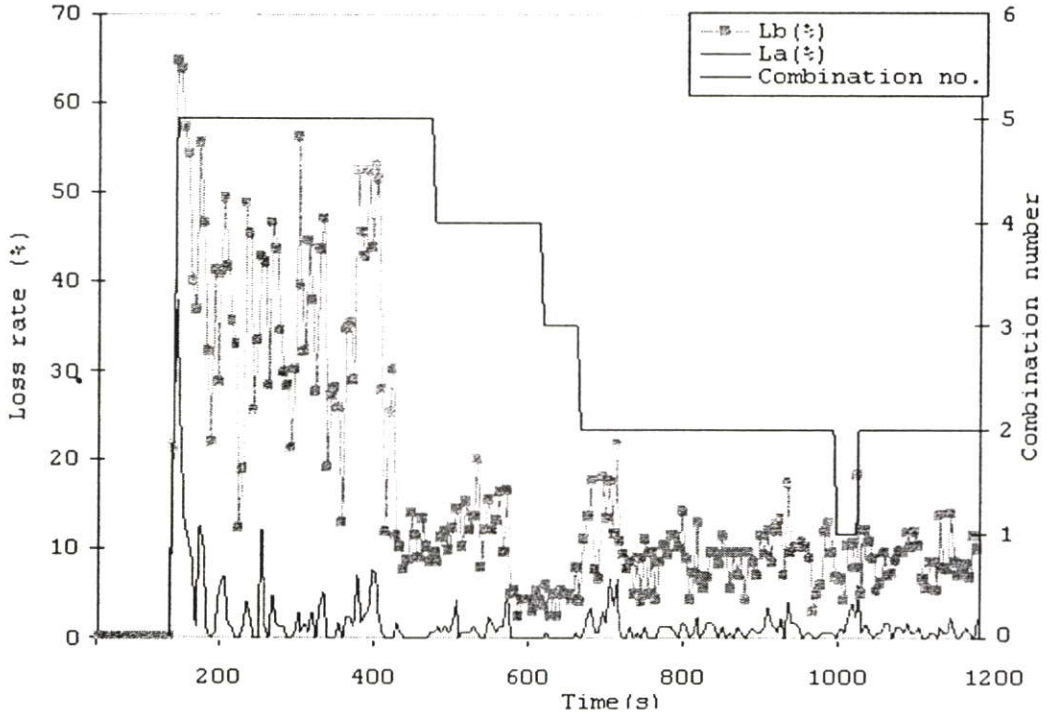
ในการประเมินสมรรถนะของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวแต่ละอัลกอริทึม ใช้วิธีการนับจำนวนครั้งที่ L_a มีค่าเกินค่าเทรสโฮลด์ HIGH ซึ่งเป็นการนับว่ามีกี่ครั้งที่มีการสูญหายของข้อมูลเสียงมากจนส่งผลกระทบต่อคุณภาพเสียงได้ รวมทั้งพิจารณาค่าเฉลี่ยของ L_a จากการจำลองของแต่ละอัลกอริทึม และพิจารณาถึงอัตราบิตที่ใช้ในการที่ส่งเสียงซึ่งเป็นผลจากการเลือก Combination ของแต่ละอัลกอริทึม ซึ่งผลการทดลองในแต่ละตอนมีดังนี้

4.2.1 ผลการจำลองเครือข่ายให้มีความหนาแน่นของทราฟฟิกต่ำ

เมื่อพิจารณาค่า Combination number ซึ่งระบุจำนวนซ้ำของข้อมูลเสียงที่มีการปรับตัวตามสภาพของเครือข่าย จากกราฟในรูปที่ 4.4 และ 4.5 จะเห็นว่าเมื่ออัลกอริทึม CNR ที่มีการใช้ค่า reward value ที่ใช้การปรับค่าคงที่ปรับให้เรียบ จะทำให้ค่า combination number มีค่าน้อยกว่าการคำนวณแบบวิธีก่อนหน้า และจำนวนครั้งที่ L_a มีค่าไม่เกินค่าเทรสโฮลด์ HIGH ของอัลกอริทึมมีค่าเพิ่มขึ้น



รูปที่ 4.4 การเปลี่ยนแปลงของ Combination โดยใช้อัลกอริทึม CNR เมื่อไม่ใช้การปรับให้เรียบ



รูปที่ 4.5 การเปลี่ยนแปลงของ Combination โดยใช้อัลกอริทึม CNR เมื่อใช้การปรับให้เรียบ

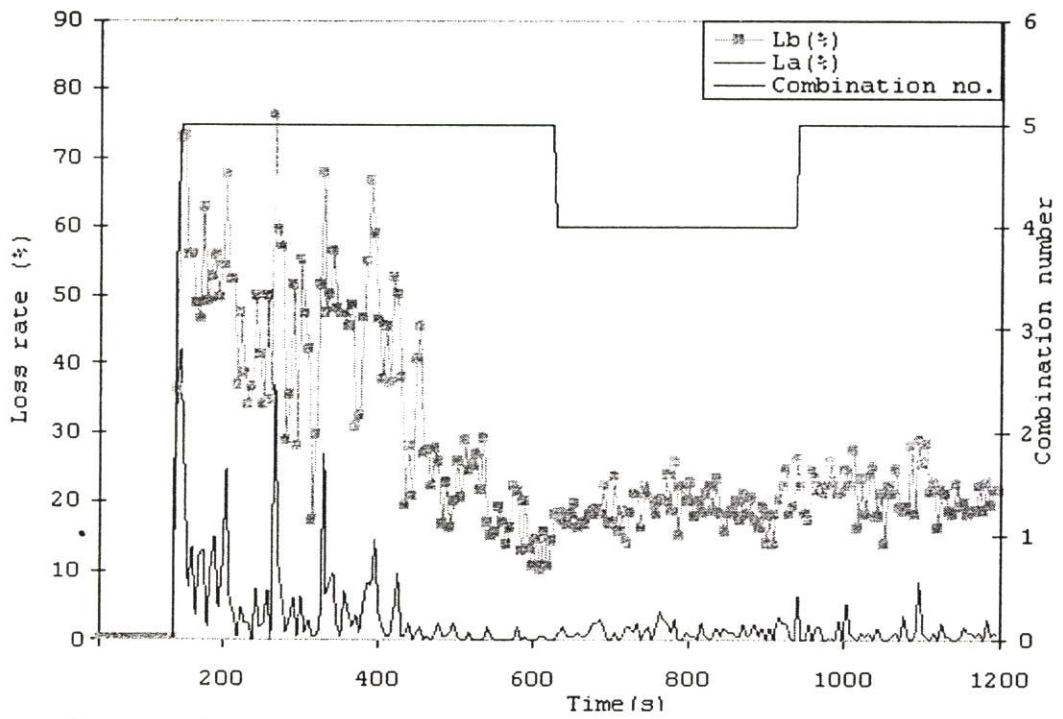
ค่าพารามิเตอร์สำคัญที่สามารถอ่านได้จาก Trace file แสดงดังตารางที่ 4.1 ได้แก่จำนวนครั้งที่ L_a มีค่าเกินค่าเทรสโฮลด์ HIGH, จำนวนแพ็กเก็ตที่ฝั่งผู้รับได้รับ, ค่าเฉลี่ยจำนวนบิตที่ได้รับต่อวินาที, ค่าเฉลี่ยเวลาหน่วงแล้วค่าเฉลี่ยจิตเตอร์

ตารางที่ 4.1 แสดงค่าพารามิเตอร์ที่สำคัญจาก Trace file เมื่อเครือข่ายมีความหนาแน่นทราฟฟิคต่ำ

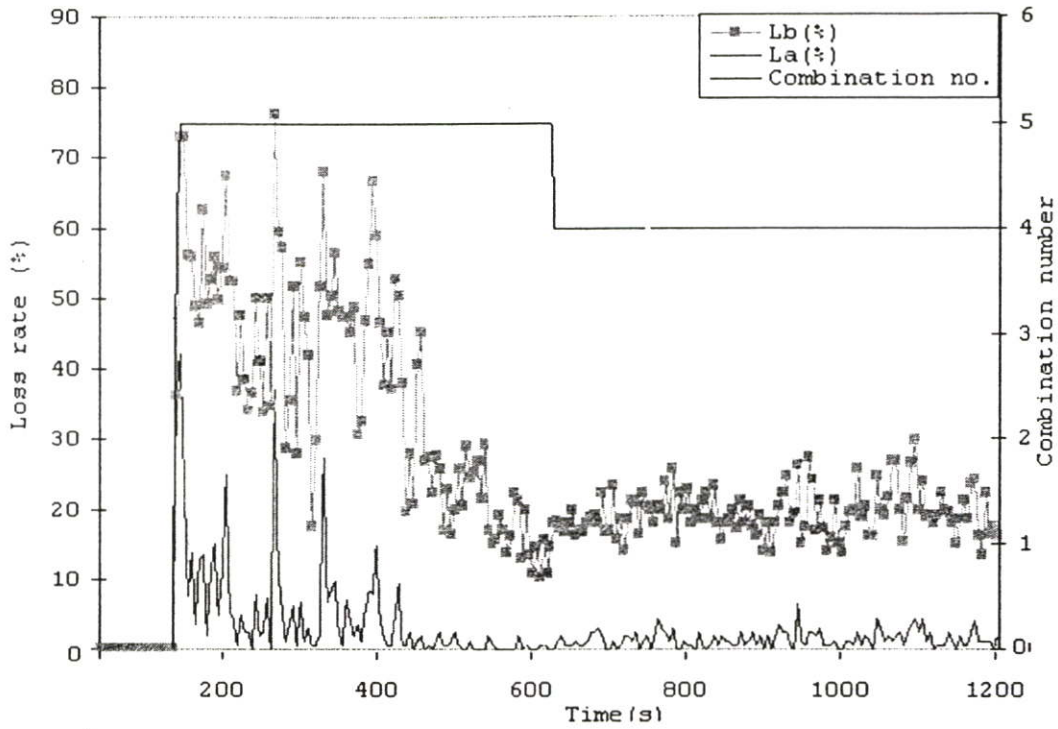
Parameter	ใช้ค่าคงที่ปรับให้เรียบ	ไม่ใช้ค่าคงที่ปรับให้เรียบ
No. period that L_a>high threshold	13	17
Received Packets	505,940	505,242
Average Throughput (kbps)	1,882.79	1,882.86
Average Delay (ms)	82.6599	79.2426
Average Jitter1(ms)	154.238	151.888

4.2.2 ผลการจำลองเครือข่ายให้มีความหนาแน่นของทราฟฟิกสูง

ในการทดลองตอนที่ 2 สภาพความคับคั่งของเครือข่ายจะแตกต่างจากตอนที่ 1 โดยเพิ่มปริมาณทราฟฟิคให้หนาแน่นขึ้น และปรับค่าคงที่ปรับให้เรียบมาใช้ค่าคงที่เป็น 0.98 ตามอัลกอริทึม RCCS จะได้ผลการทดลองดังรูปที่ 4.7 และ 4.8



รูปที่ 4.6 การเปลี่ยนแปลงของ Combination โดยใช้อัลกอริทึม CNR เมื่อไม่ใช้การปรับให้เรียบ



รูปที่ 4.7 การเปลี่ยนแปลงของ Combination โดยใช้อัลกอริทึม CNR เมื่อใช้การปรับให้เรียบ

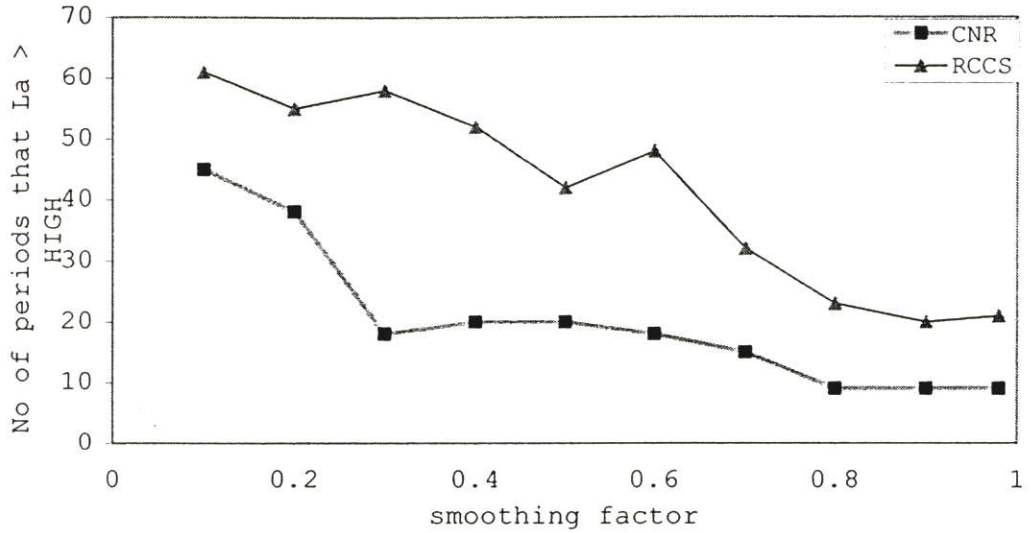
ค่าพารามิเตอร์ที่สำคัญเมื่อใช้อัลกอริทึมแบบมีการปรับค่าคงที่ปรับให้เรียบในเครือข่ายที่มีทราฟฟิคสูงแสดงดังตารางที่ 4.2

ตารางที่ 4.2 แสดงค่าพารามิเตอร์ที่สำคัญจาก Trace file เมื่อเครือข่ายมีความหนาแน่นสูง

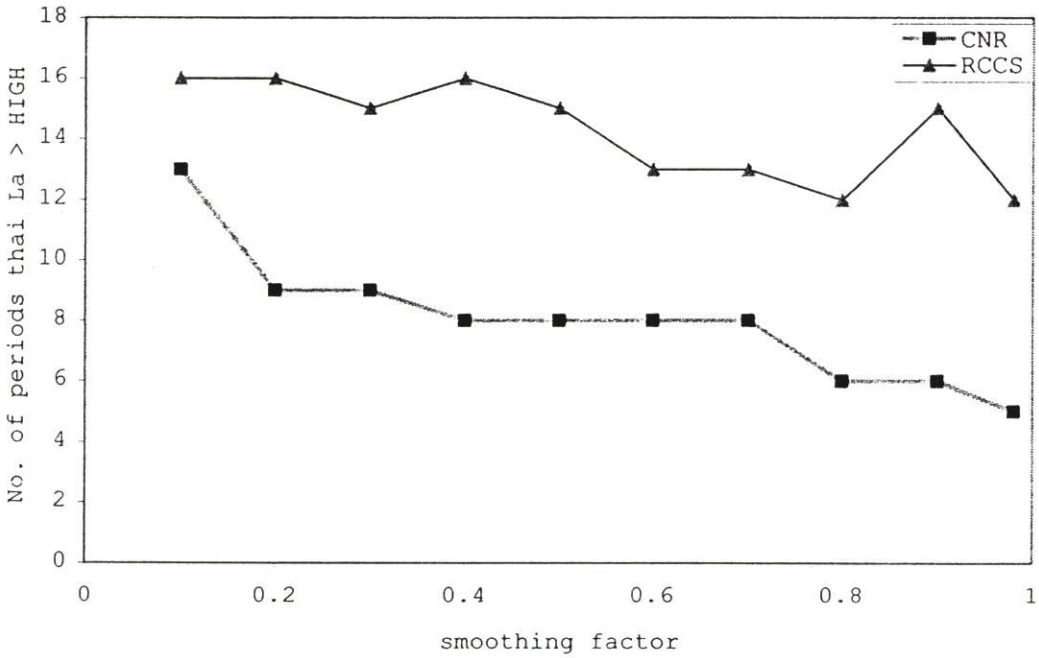
Parameter	ใช้ค่าคงที่ปรับให้เรียบ	ไม่ใช้ค่าคงที่ปรับให้เรียบ
No. period that L_a>high threshold	30	35
Received Packets	506,720	506,418
Average Throughput (kbps)	1,884.16	1,884.11
Average Delay (ms)	111.273	113.569
Average Jitter1(ms)	194.376	196.926

4.2.3 ผลการปรับค่าคงที่ให้เรียบในอัลกอริทึม CNR เปรียบเทียบกับอัลกอริทึม RCCS

เมื่อพิจารณาโดยละเอียดถึงแนวโน้มอัตราการสูญหายของข้อมูลเสียงที่จำนวนครั้งของ L_a สูงกว่าเทรสโฮลด์ HIGH พบว่าเมื่อใช้ค่าคงที่ปรับให้เรียบที่ค่าต่าง ๆ กัน อัลกอริทึม CNR กับ RCCS จะให้ค่าเปลี่ยนแปลงไปในทิศทางเดียวกันดังรูปที่ 4.8 และ รูปที่ 4.9 นั่นคือในเครือข่ายที่ทราฟฟิคมีความหนาแน่นสูงจะต้องใช้ค่าคงที่ปรับให้เรียบที่มีค่ามาก เพื่อให้จำนวนครั้งที่ของ L_a สูงกว่าเทรสโฮลด์ HIGH ลดลง

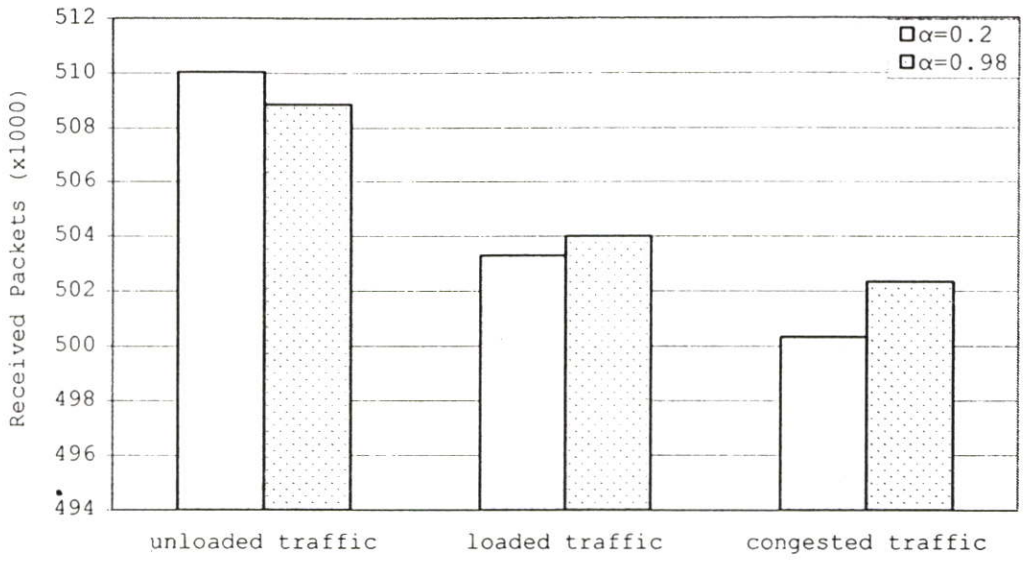


รูปที่ 4.8 จำนวนครั้งที่ L_a เกินเทรสโฮลด์ HIGH (ได้รับ Receiver Report 240 ครั้ง) เมื่อใช้การปรับให้เรียบค่าต่างกันในเครือข่ายความหนาแน่นของทราฟฟิคต่ำ



รูปที่ 4.9 จำนวนครั้งที่ L_a เกินเทรสโฮลด์ HIGH (ได้รับ Receiver Report 240 ครั้ง) เมื่อใช้การปรับให้เรียบค่าต่างกันในเครือข่ายความหนาแน่นของทราฟฟิคสูง

ดังแผนภูมิรูปที่ 4.10 นั่นคือการใช้ค่าคงที่ปรับให้เรียบที่มีค่าน้อย จะเหมาะกับเครือข่ายทีมีความหนาแน่นของทราฟฟิคต่ำ และการใช้ค่าคงที่ปรับเรียบค่าสูงจะเหมาะกับเครือข่ายที่มีความหนาแน่นของทราฟฟิคสูง



รูปที่ 4.10 การใช้ค่าคงที่ปรับให้เรียบที่แตกต่างในแต่ละสภาพเครือข่าย

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ศึกษาวิธีการควบคุมความผิดพลาดแบบปรับตัวซึ่งเป็นวิธีการที่ใช้พื้นฐานของ FEC และประยุกต์การปรับให้เรียบ เพื่อให้อัลกอริทึมสามารถทำงานได้ดีขึ้นในสภาพเครือข่ายที่แตกต่างกัน จากการศึกษาอัลกอริทึมของการควบคุมความผิดพลาดแบบปรับตัวที่ได้เสนอก่อนหน้าคือ อัลกอริทึม CNR และอัลกอริทึม RCCS พบว่าค่าคงที่ปรับให้เรียบมีผลต่อรูปแบบการส่งข้อมูลซ้ำที่ดีที่สุดให้แก่ระบบ ในวิทยานิพนธ์นี้จึงได้ประยุกต์ เทคนิคการปรับให้เรียบแบบเอกซโพเนนเชียลโดยปรับค่าคงที่ปรับให้เรียบในอัลกอริทึม CNR จากการประเมินผลการทำงานโดยการจำลองบน NS-2 พบว่า ค่าคงที่ปรับให้เรียบมีผลต่อประสิทธิภาพการทำงานของอัลกอริทึม ในสภาพเครือข่ายที่แตกต่างกัน นั่นคือในเครือข่ายที่มีความหนาแน่นของทราฟฟิคต่ำเมื่อใช้ค่าคงที่ปรับให้เรียบค่าน้อย (0.2) อัลกอริทึม CNR สามารถควบคุมความผิดพลาดได้ดีกว่าเดิม โดยมีจำนวนครั้งที่ L_q เกินค่าเทรสโฮลด์ HIGH ต่ำลง รวมทั้งมีค่าพารามิเตอร์ที่สำคัญอื่นๆ เช่นค่าหน่วงเวลา,จิตเตอร์ และจำนวนแพ็กเก็ตที่ผู้รับได้รับมีค่าที่ดีขึ้น ส่วนในเครือข่ายที่มีความหนาแน่นของทราฟฟิคสูง เมื่อใช้ค่าคงที่ปรับให้เรียบค่าสูง (0.98) ในเครือข่ายที่มีความหนาแน่นของทราฟฟิคมากมีการควบคุมอัตราการสูญหายของข้อมูลเสียงได้ดีกว่าอัลกอริทึมเดิมเช่นกัน

เมื่อเปรียบเทียบ CNR กับอัลกอริทึม RCCS ที่ค่าคงที่ปรับให้เรียบค่าต่าง ๆ จะเห็นได้ว่าอัลกอริทึม CNR มีอัตราบิตเฉลี่ยในการส่งข้อมูลเสียงดีกว่าอัลกอริทึม RCCS และให้แนวโน้มผลอัตราการสูญหายของข้อมูลเสียง ไปในทิศทางเดียวกัน ดังนั้นการใช้ค่าคงที่ปรับให้เรียบที่ปรับตัวตามสภาพเครือข่ายทำให้การควบคุมความผิดพลาดมีการใช้ข้อมูลซ้ำที่น้อยลงแต่อัตราความถูกต้องยังอยู่ในเกณฑ์ที่ยอมรับได้ (อัตราการสูญหายของข้อมูลเสียงไม่เกิน 5%) ส่งผลให้ค่าเวลาหน่วงน้อยลง, ลดการสิ้นเปลืองแบนด์วิท และช่วยให้มีการใช้ทรัพยากรของระบบอย่างคุ้มค่า

5.2 ข้อเสนอแนะ

ในวิทยานิพนธ์นี้ไม่ได้จำลองการทำงานบนเครือข่ายจริง จึงมีความเป็นไปได้ว่าอาจมีปัจจัยอื่นทำให้อัตราการสูญหายของข้อมูลเพิ่มขึ้นหรืออาจมีการติดขัดสูงเกินปกติ ดังนั้นจึงต้องมีการทดลองหาความสัมพันธ์ของอัตราการสูญหายของข้อมูลกับตัวค่าพารามิเตอร์อื่น ๆ นอกเหนือไปจากค่าคงที่ปรับให้เรียบ และติดตามอัตราการสูญหายที่อาจเกิดขึ้นในเครือข่ายการใช้งานจริงว่าเมื่อใช้ การควบคุมความผิดพลาดโดยอัลกอริทึม CNR แล้ว ยังทำให้ประสิทธิภาพในการส่งผ่านข้อมูลที่อยู่ในเกณฑ์ที่ยอมรับได้หรือไม่

เอกสารอ้างอิง

[1] ITU-T Recommendation G.723.1, "Speech coders: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," 1996.

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

[3] ISO/IEC 14496-2, "Coding of audio-visual objects – Part 2: Visual," 2001.

[4] S. Jacobs and A. Eleftheriadis, "Streaming video using dynamic rate shaping and TCP congestion control," *Journal of Visual Communication and Image Representation*, vol. 9, no. 3, pp. 211-222, Sept. 1998.

[5] S. Jacobs and A. Eleftheriadis, "Real-time dynamic rae shaping and control for Internet video applications," in *Proceedings of Workshop on Multimedia Signal Processing*, Princeton, USA, Jun. 1997, pp. 23–25.

[6] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," *Computer Communications*, vol.9, no.1, pp. 49-58, Jan. 1996.

[7] J.-C. Bolot, T. Terletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet," in *Proceeding of ACM SIGCOMM '94*, London, England, Aug. 1994, pp. 58-67.

[8] D. Sisalem and H. Schulzrinne, "The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme," in *Proceedings of International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Cambridge, England, Jul. 1998, pp. 215-226.

[9] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson "RTP: A transport protocol for real-time applications," RFC 3550, Jul. 2003.

[10]D. Sisalem and H. Schulzrinne, "The direct Adjustment Algorithm: a TCP-friendly adaptation scheme", in *Proceedings of International Workshop on Quality of Future Internet Services*, Berlin, Germany, Sept. 2000, pp. 68–79.

[11]J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of ACM SIGCOMM '98*, Vancouver, Canada, Oct. 1998, pp. 303–314.

[12]D. Sisalem and F. Emanuel, "QoS control using adaptive layered data transmission," in

Proceedings of IEEE International Conference on Multimedia Computing and System, Austin, USA, Jul. 1998, pp. 4–12.

[13]S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-Driven Layered Multicast," in *Proceedings of ACM SIGCOMM '96*, Stanford, USA, Aug. 1996, pp. 117–130.

[14]L. Wu, R. Sharma, and B. Smith, "Thin streams: An architecture for multicasting layered video," in *Proceedings of International Workshop on Network and Operating Systems Support for Digital Audio and Video*, St. Loius, USA, May. 1997.

[15]C. Perkins, 0. Hodson, and V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," *IEEE Network*, vol. 12, no. 5, pp. 40–48, Sept.- Oct. 1998.

[16] C. S. Perkins et al., "RTP Payload for redundant audio data," RFC2198, Sept. 1997.

[17]J-C. Bolot, A.V Garcia, "Control mechanisms for packet audio in the Internet," in *Proceedings of IEEE INFOCOM '96*, vol. 1, San Francisco, USA, Mar. 1996, pp. 232-239.

[18]C. Padhye, K.J. Christensen and W. Moreno, "A new adaptive FEC loss control algorithm for voice over IP applications", in *Proceedings of IEEE IPCCC 'OO*, Phoenix, USA, Feb. 2000, pp. 307-313.

[19]M.-K. Ji, S.-H. Lee, T.-U. Choi, S.-H. Park, J.-G. Kang, K.-D. Chung, "Selecting an audio redundancy codec combination for error control in Internet telephony," in *Proceedings of IEEE ICPADS 2001*, Kyongju, South Korea, Jun. 2001, pp. 645–649.

[20]J-C. Bolot, T. Turletti, "Adaptive error control for packet video in the Internet", in *Proceeding of IEEE ICIP '96*, vol. 1, Lausanne, Switzerland, Sept. 1996, pp. 25–28.

[21]ITU-T Recommendation H.261, "Video codec for audiovisual services at px64 kbits/s," 1996.

[22]ITU-T Recommendation H.263, "Video coding for low bit rate communications," version 1, 1996, version 2, 1998.

[23]T. Turletti, C. Huitema, "Videoconferencing in the Internet," *IEEE/ACM Transactions on Networking Journal*, vol. 4, no. 3, pp.340-351, Jun. 1996.

[24]ITU-T Recommendation P.800: "Methods for subjective determination of transmission quality," Aug. 1996.

[25]CCITT Recommendation G.711, "Pulse Code Modulation (PCM) of voice frequencies," 1988.

[26]ITU-T Recommendation G.729, "Coding of speech at 8 kbit/s using Conjugate Structure

Algebraic-Code-Excited Linear-Prediction (CS-ACELP)," 1996.

[27]RADCOM Equipment, Inc., "Fine-tuning Voice over Packet services," http://www.radcom.com/radcom/technlgy/pdf/FineTuningVoiceOverPackeServices.pdf, Apr. 2, 2005.

[28]N. Jayant and S. W. Christensen, "Effect of Packet Losses in Waveform Coded Speech and Improvements due to an Odd-Even Sample-Interpolation Procedure," *IEEE Transactions on Communications*, vol.COM-29, no.2, pp. 101-109, Feb. 1981.

[29]L. C. Ramac, P. K. Varshney, "A wavelet domain diversity method for transmission of images overwireless channels," *IEEE Journal on Selected Areas in Communications*, vol.18, no.6, pp. 891-898, Jun. 2000.

[30]Bruce L. Bowerman, Richard T. O'Connell, *"Forecasting and time series: an applied approach Edition 3rd "*, Belmont, CA : Duxbury Press, 1993.

[31]Robert G. Brown, *"Smoothing, Forecasting and Prediction of Discrete Time Series. "*, Printice-Hall, Inc., 1963.

[32]Douglas C. Montgomery, Lynwood A. Johnson, John S. Gardiner, "Forecasting & Time Series Analysis 2nd Edition" , McGraw-Hill, Inc., 1990.

[33]B. Piyatamrong P. Piyakhun., "Smooth loss with variability smoothing factor to increase bandwidth utilization", IEEE Pacific Rim Conference on Communications, Computers and signal Processing. (PACRIM 2001), vol.2, pages. 493-494.

[34]H. Schulzrine, S. Casner, "RTP profile for audio and video conferences with minimal control," RFC 3551, Jul. 2003.

[35]S. Chuaywong, S. Kamolphiwong, T. Kamolphiwong, "Performance Enhancement of FEC Mechanism for IP Telephony," in *Proceeding of the 2005 Electrical/Electronics, Computer, Telecommunications, and Information Technologiey (ECTI) International Conference*, vol.2, Cholburi, Thailand, May. 2005, pp. 697-700.

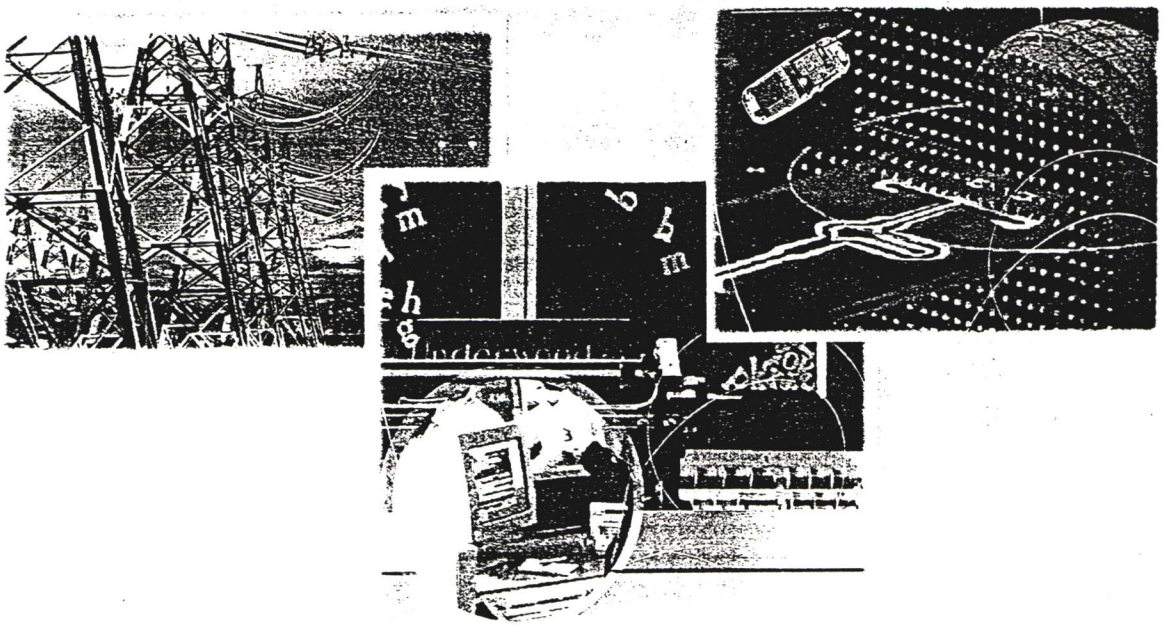
[36] The Network Simulator, NS-2, http://www.isi.edu /nsnam/ns/, Oct. 23, 2005.

[37]รศ.ดร. วิชิต หล่อจีระชุณห์กุล, "การวิเคราะห์ความคลาดเคลื่อนในการพยากรณ์", การประชุมวิชาการสถิติประยุกต์ ครั้งที่ 10, คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์.

ผลงานทางวิชาการที่ได้รับการตีพิมพ์

1. P.MOUNGNOUL, K. KLAPPONG, W. INKIEW, S. JUNNAPIYA, "A Study of Adaptive Forward Error Control in Audio Communication using Smooth Method," in *Proceeding of the 2007 Electrical/Electronics, Computer, Telecommunications, and Information Technologiey (ECTI) International Conference*, vol.I, Chiang Rai, Thailand, May. 2007, pp. 121-124.

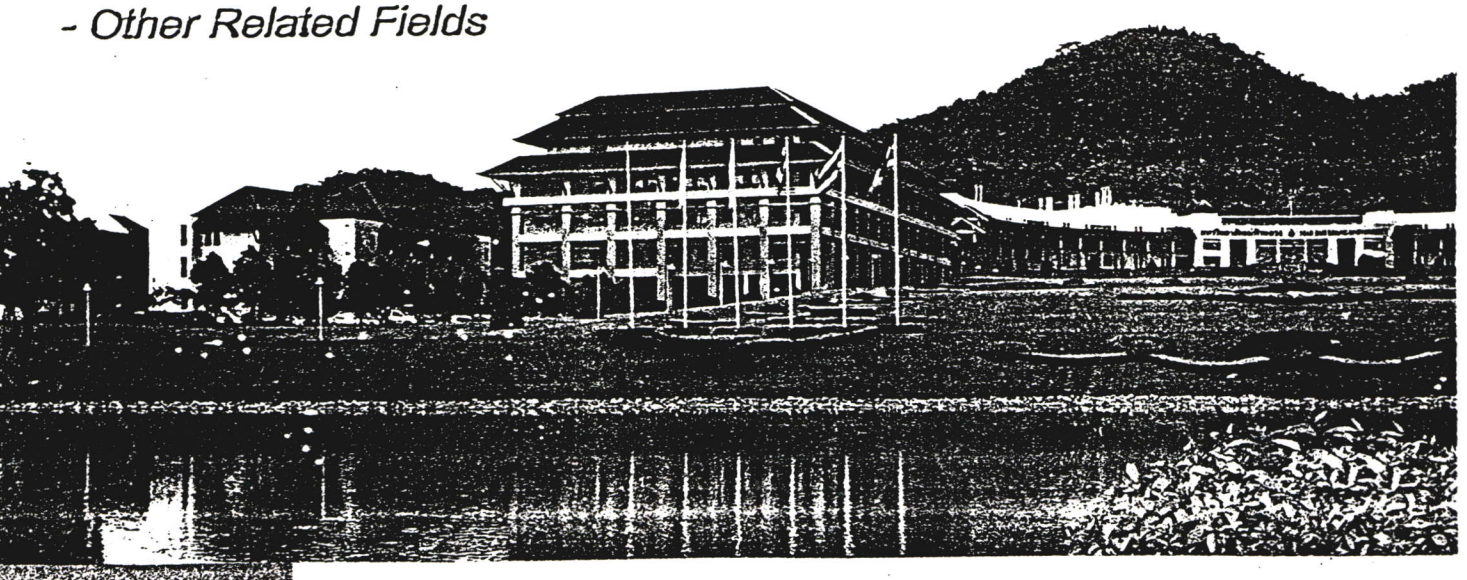
VOLUME 1



ECTI-CON 2007

Mae Fah Luang University, Chiang Rai, Thailand
May 9-12, 2007

VOLUME 1
- *Circuits and Systems*
- *Control Engineering*
- *Electrical Power Engineering*
- *Other Related Fields*



ECTI IEEE NECTEC Western Digital

10:30 AM - 12:10 AM | **Session FAM2-2** | **Friday, 11 May 2007**

Analog Circuits 2

Chairperson : Banlue Srisuchinwong , Sirindhorn International Institute of Technology

FAM2-2-1 10:30 AM — **Composite Four-Terminal Floating Nullor with Variable Voltage and Current Gains** — 113

Jirawat Hirunporm[1], Sumalee Unhavanich[2], Worapong Tangsrirat[1], Wanlop Surakampontorn[1]

[1]Research Center for Communication and Information Technology (ReCCIT), King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Thailand

[2]Department of Industrial Electrical Technology (IET), King Mongkut's Institute of Technology North Bangkok, Bangkok, Thailand

FAM2-2-2 10:50 AM — **Current-mode Universal Biquad Including Minimum Component Count Circuits** — 117

Pratya Mongkolwai, Tattaya Pukkalanun, Worapng Tangsrirat

Research Center for Communication and Information Technology (ReCCIT), King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

FAM2-2-3 11:10 AM — **A Study of Adaptive Forward Error Control in Audio Communication using Smooth Method** — 121

Phichet Moungnoul, Kornhathai Klappong, Wiphada Inkiew, Somyot Junnapiya

Research Center for Communication and Information Technology (ReCCIT), King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

FAM2-2-4 11:30 AM — **Rectangular Wave Delta Modulation Buck Regulator for Voltage Regulator Modules** — 125

J. Ngarmnil, K. Sooksood

Department of Electronic Engineering, Faculty of Engineering, Mahanakorn University of Technology, Bangkok, Thailand

FAM2-2-5 11:50 AM — **Dual Input All-Pass Networks Using MO-OTA and its Application** — 129

Pipat Prommee[1], Krit Angkeaw[2], Jirasak Chanwutitum[2], Kobchai Dejhan[1]

[1]Research Center for Communication and Information Technology (ReCCIT), King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

[2]Industrial Electrical Technology Department, Faculty of Engineering, King Mongkut's Institute of Technology North Bangkok, Bangkok, Thailand

10:30 AM - 12:10 AM | **Session FAM2-3** | **Friday, 11 May 2007**

Electronic Devices

Chairperson : Boonying Charoen, Khon Kean University

FAM2-3-1 10:30 AM — **A Physics-Based InP/InGaAs SHBT Model for SPICE simulation** — 133

Suchat Osaklang[1], Chiranut Sangiamsak[2]

[1]Graduate School, Department of Electrical Engineering, Khon Kaen University, Thailand

[2]Department of Electrical Engineering, Khon Kaen University, Thailand

A Study of Adaptive Forward Error Control in Audio Communication using Smooth Method

Phichet Moungnoul, Kornhathai Klappong, Wiphada Inkiew and Somyot Junnapiya
Faculty of Engineering and Research Center for Communication and Information Technology
King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Road, Bangkok, 10520, Thailand
E-mail: {phichet,somyot}@telecom.kmitl.ac.th,s46064125@kmitl.ac.th

***Abstract*-This paper studies the adaptive FEC control algorithms to improve control the amount of redundancy. The recently adaptive FEC "CNR algorithm" considers the history of packet loss in the network before changing the amount of redundancy and does not waste time. The performance of the CNR algorithm is used for simulation model. The smoothing parameter is varied in order to adapt and changed the characteristics of time series from the packet loss. Result shows the adaptive error control algorithm can decrease the loss rate by using a variability smoothing factor to appropriate network condition.**

I. INTRODUCTION

The motivation for transporting audio stream over IP networks potential is cost savings to achievable by eliminating or bypass the circuit-switched telephony infrastructure. The conversation quality between the sender and the receiver in the Internet telephony system is depends on specific impairments such as packet loss, delay, jitter, etc., which related to the characteristic of packet network. A number of techniques dealing with packet loss problems exist for application that exchange real-time voice [2].

Packet loss causes degradation in the quality of audio transmission applications. Forward Error Correction (FEC) methods which add redundant information to audio packets can be used to minimize the effects of packet loss. While these methods can reduce the effects of packet loss, they increase the amount of bandwidth used by an audio stream [9].

Most of audio transmission applications use UDP as the transport layer protocol. The Real-time Transport Protocol (RTP) [4] is used to provide additional functionality including sequence numbers and time stamps. The Real Time Control Protocol (RTCP) is used to return receiver statistics (e.g., the number of packets detected as lost) to the sender. RTCP packets are sent every 5 seconds by a receiver to a sender and consume very little bandwidth. IP networks are inherently best effort networks with variable packet delays and loss. While audio traffic can tolerate some amount of packet loss, a packet loss rate greater than 5% is considered harmful to the audio quality [8]. The amount of packet loss rate that can be tolerated depends on the nature of the encoding algorithm and on the sampling rate of the audio stream. Changing the IP infrastructure to support guaranteed bandwidth sessions would allow for effective transport of audio streams. Changing the Internet infrastructure is a difficult proposition, hence the interest in application level techniques for compensating for packet delay jitter and loss. Algorithms using Forward Error Correction (FEC) methods have been developed to compensate for packet loss.

However, existing methods which used media specific FEC technique can not forecast a network performance in advance. The drawback of this algorithm is in part of increasing combination number (number which specific the pattern of a redundant copy of packet) [7].

Using a currently loss report to make a decision for a transmission rate that suited for the next loss situation has not always accurate. Because of the loss rate in the next interval may not be rated to the previous intervals.

The single exponential smoothing (SES) technique is a simple and pragmatic approach to forecasting, whereby the forecast is constructed from the exponentially weighted average of pass observation [5]. If the packet loss prediction use bounded adaptive smoothing constant from the forecast loss rate in the next interval. The loss rate will be a predictable variation sequence of the number as time series (time series of packet loss), more correctly and less error in case of significant time series average changing [1].

In this paper, to study the media specific FEC-based algorithm based on increasing the combination number. The adaptive FEC control algorithm developed in [11] (the "CNR algorithm") is studied and significant improvements are evaluated. Also, Redundant Codec Combination Selection (RCCS) algorithm uses smoothing factor to calculate the reward and penalty at selecting combination [8]. Both RCCS and CNR are FEC-based error control mechanisms that the amount of redundancy is controlled by the network loss rate. This paper is to develop mechanism that attempts to eliminate or at least minimize the impact of packet loss and end-to-end delay on the quality of the audio delivered to the destinations. Analyze the effect of the variability parameters that use in process of each algorithm for improve audio transmission in the future.

This paper is organized as follows. Section II reviews the adaptive error recovery algorithms which use media specific FEC-based technique and smoothed loss scheme. Section III and IV are the experiment setup and experimental result respectively. Finally, section V presents the conclusions.

II. Recovery Algorithm with Smoothed Loss

A. RCCS Algorithm

Myoung-Kyoung Ji[8] proposed new adaptive FEC-based algorithm, called RCCS algorithm using loss and delay in order to improve the internet telephony. They used smoothing factor for update the reward and the penalty of current combination, as in equation (1) and equation (2).

$$R_i = \alpha(L_b / L_a) + (1-\alpha)R_{i-1} \quad (1)$$

$$P_i = \alpha(D_a / D_b) + (1-\alpha)P_{i-1} \quad (2)$$

This algorithm has 3 steps: the first step updates the reward and the penalty of current combination. The second step is the procedure that selects the combination whose estimate of loss rate within the maximum and minimum threshold. The last is selects the combination that exceeds the maximum of end-to-end delay.

B. CNR Algorithm

Centre for Network Research (CNR) algorithm [11] is the media specific FEC-based algorithm. Improves the Bolot (1996) [6] algorithm by increasing the combination number of audio redundancy by one (1) in each period. This will reduce waste time from CNR algorithm chooses G.723.1 codec, because it is the most widely-used audio codec in the IP Telephony system and it has low bit rate enough (6.3 kbps) [11]. Besides, it is better than encoding each audio unit two times with high and low bit rate codec. The CNR algorithm requires the Real-time Transport Control Protocol (RTCP) packet including packet loss rate before and after the reconstruction at the receiver. Whenever the RTCP packet is received, the reward value (the L_a and L_b ratio) of current combination R_c using the loss information in the packet is calculated such as equation (3) in Figure 1.

```
For each receiver report packet received do
1.Update reward of the current combination
  //Let Rc stands for reward of the current combination
                    Rc = La/Lb                              (3)
2. Calculate the percentage of packets lost
  before reconstruction, Lb
  Lb = Number of packets lost before
     reconstruction . Number of
     packets expected
3. Calculate the percentage of packets lost
  after reconstruction, La
  La = Lb - Reward associated with current
     combination number
4. If (La > HIGH) then
    new_combination = MAX_COMBINATION
    For i =1 to MAX_COMBINATION
      La = Lb / reward of combination #i
      If (La <= HIGH) then
        new_combination = i
        exit for loop
    combination = new_combination
5. If (La < LOW)for N times then
    Decrement combination
```

Figure 1 The pseudo code of the CNR algorithm

C. Proposed Method

Base on our research, the CNR algorithm can uses the exponential smoothing method to calculate the reward value in the next interval. The reward value in a previous interval will be used to decide which combination number should be adjusted.

We added the adjustment smoothing constant which was developed by Trigg and Leach [3] in the CNR algorithm to calculate reward value for unpredictable variation sequences of the number via time series method, as in Figure 2.

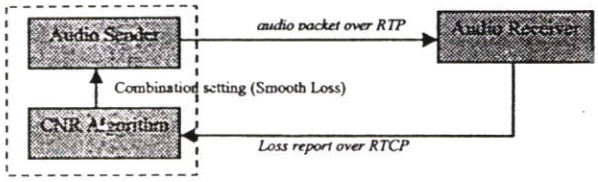


Figure 2. Mechanism to smoothing loss report

The smoothing value can obtain by the following equation. [10]

$$S_t = \alpha X_t + (1-\alpha)S_{t-1}(X) \quad (4)$$

Where S_t represents for smoothed value at time t with $0<\alpha<1$ is a smoothing factor.

Change the reward calculation (R_c) by including smoothing factor in reward formula following equation (4) by equation (5), like reward calculation in RCCS algorithm.

$$R_c = \alpha(L_b / L_a) + (1-\alpha)R_{i-1} \quad (5)$$

III. Experiment Setup

This experiment simulated the CNR algorithm which included the adaptation logic of smoothing factor using NS-2 program [13] used a network simulator.

The topology of the simulation model is shown in Figure 3. All links are drop tail links. Most of them have 10 Mbps capacity and 1ms delay except the bottleneck link between N2 and N3 which has 2 Mbps capacity and 50 ms delay.

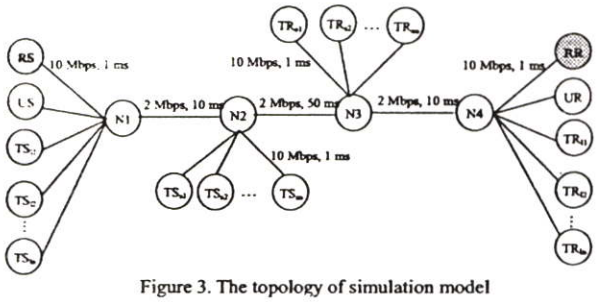


Figure 3. The topology of simulation model

The audio packets was sent from the sender (node RS) to the receiver (node RR) via the Real-time Transport Protocol (RTP) in every 30 milliseconds. RR sends the receiver report every five (5) seconds to RS. There were other nodes that sent the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffics. Each session was repeated by adding the smoothing factor by 0.1 until it is 0.98. The whole experiment would be repeated in each level of the loss model.

IV. Experimental Result

The performance of the recovery algorithm can be evaluated from many important parameters. Table I shows some statistics from NS-2 trace files in network. All average parameters were calculated from N1 source and N2 destination node. The packet size was 24 bytes. The audio traffic started to send a packet at 0.1 second and stopped at 1200 seconds.

The results shows that is used 0.2 as smoothing constant in the unloaded traffic situation, the receiver (RR) would receives amount of packets greater than smoothing constant at 0.98.

TABLE I
SIMULATION RESULTS FROM NS-2 TRACE FILE

Parameters	$\alpha = 0.2$	$\alpha = 0.98$
No. period that L_a>high threshold	19	16
Received Packets	528,500	516,505
Average Throughput (kbps)	1,882.86	1,871.72
Average Delay (ms)	7.82829	7.68629
Average Jitter1(ms)	71.018	78.1758

Figure 4 shows the comparison of different traffic congestion. It also shows that smoothing constant of 0.2 can be used in the unloaded traffic to have better amount of the received packets. And for the better result in the loaded and congested traffics, recommended adjusting the smoothing constant is 0.98.

Using the CNR to compare with RCCS mechanism for simulated the topology in the loaded traffic network, the results shown in Figure the effect of higher smoothing factor is better for adaptive error control mechanism than the lower value, as Figure 5.

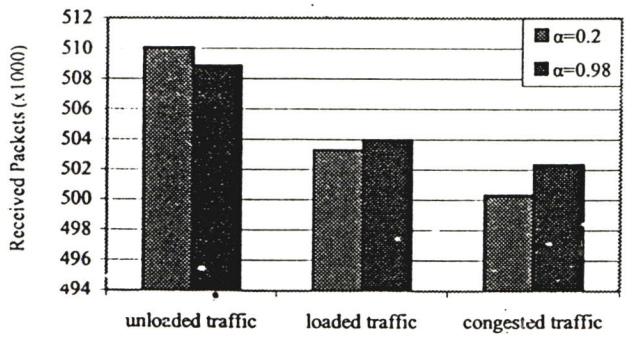


Figure 4. The received packets in different traffic congestion

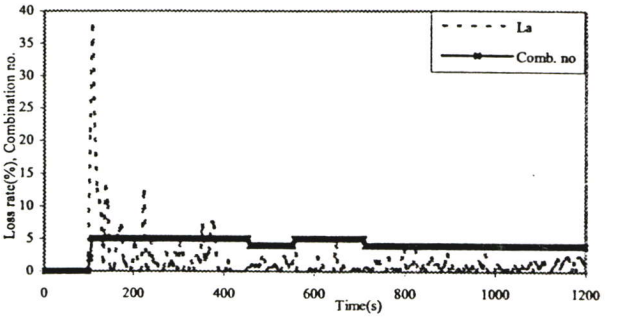


Figure 5. Behavior of loss rate when reward value follows equation (3)

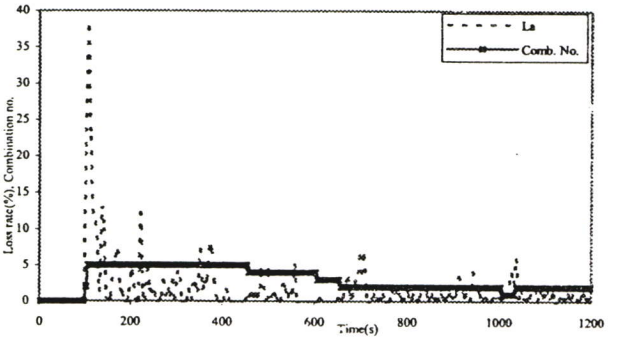


Figure 6. Behavior of loss rate when reward value follows equation (5)

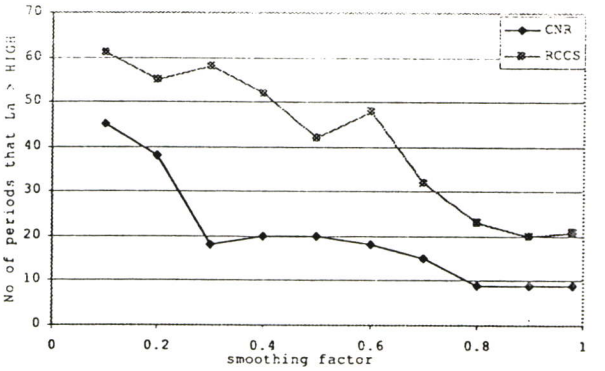


Figure 7. L_a when smoothing factor is 0.1 until 0.98 in unload traffic

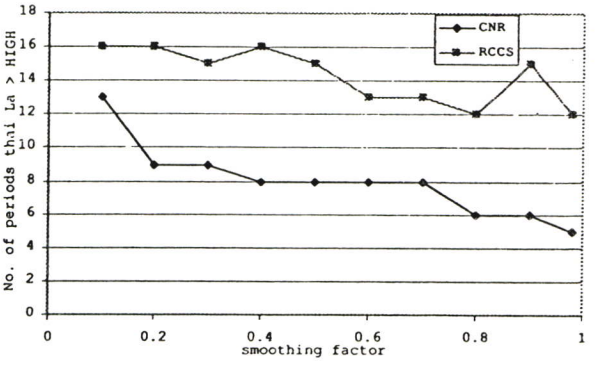


Figure 8. L_a when smoothing factor is 0.1 until 0.98 in loaded traffic

We observed from running the CNR algorithm the behavior perform follow Figure 5-8 that after changing the combination number of loaded and unloaded traffic, these two behaviors use the same value of smoothing factor, which was 0.98. However, when we adjusted a situation of network by using a smoothing factor of 0.2 in the unloaded traffic behavior and calculated the reward value (R), the results showed that the loss rate was not greater than the high threshold (5%) and not lower than the low threshold (1%). Moreover, the combination number of communication transmission was decreased. Consequently, the redundant packets were decrease. The receiver and the sender would have a better quality of communication in exchanging the real time voice.

V. CONCLUSION

In this paper, introduce an adaptive parameter exponential smoothing method to solve the drawback of the media specific FEC algorithm in part of increasing combination number. The CNR algorithm performs well when varying smooth parameter at low value (0.2) in unloaded traffic and better performance when set this parameter at high value (0.98) in loaded or congested traffic. The suited redundant information of audio packets can be adjusted to appropriate network in advance. Therefore, the adaptive recovery algorithm can apply the smooth method to recovery mechanism in order to obtain more available utilized bandwidth.

REFERENCES

[1] B. Piyatamrong P. Piyakhun., "Smooth loss with variability smoothing factor to increase bandwidth utilization," *PACRIM 2001*, Vol.2, 26 -28 August, Victoria, B.C., Canada, 2001, pp. 493 – 494.

[2] C. Perkins, O. Hodson, and V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," *IEEE Network*, Vol. 12, No. 5, September-October 1998, pp. 40 – 48.

[3] Douglas C. Montgomery, Lynwood A. Johnson, John S. Gardiner, *Forecasting & Time series Analysis 2nd Edition*, McGraw-Hill, Inc., 1990.

[4] H. Schulzrinne, S. Casner, R. Fredrick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," Internet Engineering Task Force, RFC 1889, January 1996

[5] James W. Taylor, "Smooth Transition Exponential Smoothing," Journal of Forecasting, 2004, Vol. 23, pp. 385-394

[6] J-C. Bolot, A.V Garcia, "Control mechanisms for packet audio in the Internet," *Proc. IEEE INFOCOM '96*, April 1996, pp.232-239.

[7] J-C Bolot, S. Fosse-Parisis and D. Towsely, "Adaptive FEC-Based Error Control for Internet Telephony," *Proceedings of IEEE INFORCOM*, Vol. 3, 1999, pp.1453-1460.

[8] M. Ji, S. Lee, T. Choi, S. Park, J. Hang, K. Chung, "Selecting an audio redundancy codec combination for error control in Internet telephony," *ICPADS 2001*. Proceeding. Eighth International Conference on 26-29 June 2001, pp. 645-649.

[9] N. Jayant and S. W. Christensen, "Effects of Packet Losses in Waveform Coded Speech and Improvements due to an Odd-Even Sample-Interpolation Procedure," *IEEE Transactions on Communications*, Vol. COM-29, No. 2, February 1981,pp. 101 - 109,

[10] Robert G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*. Prentice-Hall, Inc., 1963.

[11] S.Chuaywong, S. Kamolphiwong,T. Kamolphiwong, "Performance Enhancement of FEC Mechanism for IP Telephony," *Proceeding of the 2005 ECTI*, vol.2, Cholburi, Thailand, May. 2005, pp. 697-700.

[12] V. Kumar, M. Korpi and S. Sengoodan, *IP Telephony with H.323*, Wiley, United State of America, 2001.

[13] The Network Simulator, NS-2, "http://www.isi.edu /nsnam/ns/"

ประวัติผู้เขียน

นางสาวกรหทัย กลับพงศ์ เกิดเมื่อวันที่ 27 พฤศจิกายน 2521 ที่จังหวัดนครศรีธรรมราช ปีการศึกษา 2545 สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ จากมหาวิทยาลัยสงขลานครินทร์ จังหวัดสงขลา