

การตรวจสอบการกักกันแพ็กเก็ตสูญหายในการส่งผ่านข้อมูลเสียง
โดยใช้ FEC ที่ชนกับสื่อ

INVESTIGATING PACKET LOSS RECOVERY IN MEDIA SPECIFIC FEC
AUDIO TRANSMISSION

อรุณี กาลวิบูลย์
ARUNEE KANWIBOON

วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2550

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

**การตรวจสอบการกู้คืนแพ็กเก็ตที่สูญหายในการส่งผ่านข้อมูลเสียง
โดยใช้ FEC ที่ขึ้นกับสื่อ**

**INVESTIGATING PACKET LOSS RECOVERY IN MEDIA SPECIFIC FEC
AUDIO TRANSMISSION**

อรุณี กาลวิบูลย์

ARUNEE KANWIBOON

เลขหมู่.....
เลขทะเบียน.....**74886**.....
วัน,เดือน,ปี...**15 ต.ค. 2550**.....

**วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2550**

**INVESTIGATING PACKET LOSS RECOVERY IN MEDIA SPECIFIC FEC
AUDIO TRANSMISSION**

ARUNEE KANWIBOON

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING TELECOMMUNICATIONS ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKARANANG**

2007

COPYRIGHT 2007

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

| | |
|-----------------------------|--|
| หัวข้อวิทยานิพนธ์ | การตรวจสอบการก่อกวนเพื่ก่เกิดที่สูญหายในการส่งผ่านข้อมูลเสียง โดยใช้ FEC ที่ขึ้นกับสื่อ |
| นักศึกษา | น.ส. อรุณี กาลวิบูลย์ |
| รหัสประจำตัว | 48060929 |
| ปริญญา | วิศวกรรมศาสตรมหาบัณฑิต |
| สาขาวิชา | วิศวกรรมโทรคมนาคม |
| พ.ศ. | 2550 |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์ | รศ.ดร. กอบชัย เดชหาญ |

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นการตรวจการควบคุมความผิดพลาดแบบปรับตัวในการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต ข้อจำกัดอย่างหนึ่งของเครือข่ายอินเทอร์เน็ตคือ ไม่มีการรับประกันคุณภาพการบริการ ดังนั้นจึงไม่สามารถที่จะรับประกันได้ว่าคุณภาพของเสียงที่ผู้รับได้รับเหมือนกับต้นฉบับทุกประการ ซึ่งอาจจะมีข้อมูลบางส่วนเกิดการสูญหายในระหว่างการเดินทาง โดยได้ศึกษาวิธีการควบคุมคุณภาพเสียงมีพื้นฐานมาจากวิธีการที่เรียกว่า FEC (Forward Error Correction) แบบ Media Specific และใช้วิธี Least Square Method กับ อัลกอริทึม (Algorithm) CNR ในการประมาณค่าพารามิเตอร์ (parameter) สำคัญ ในการกำหนดรูปแบบและปริมาณของข้อมูลซ้ำ (Combination)

| | |
|-----------------------|--|
| Thesis Title | Investigating Packet Loss Recovery in Media Specific FEC Audio Transmission |
| Student | Ms. Arunee Kanwiboon |
| Student ID | 48060929 |
| Degree | Master of Engineering |
| Program | Telecommunications Engineering |
| Year | 2007 |
| Thesis Advisor | Assoc. Prof. Dr.Kobchai Dejhan |

ABSTRACT

This thesis proposes the packet loss recovery technique in the audio stream at the sender side. The adaptive error control in Internet telephony system which uses this packet recovery scheme based on the media specific Forward Error Correction (FEC). By adapting the method with least square in recovery algorithm for the best fitting curve of reward value. This technique can estimate the trend of outcomes and improve the adjustment of redundant packets of the recovery algorithm. The results of experiment show that the adaptive error control algorithm can decrease the packet loss rate.

กิตติกรรมประกาศ

ขอขอบพระคุณ รศ.ดร. กอบชัย เดชหาญ ที่ปรึกษาวิทยานิพนธ์ ที่ได้ให้คำปรึกษาและ
ผู้ช่วยศาสตราจารย์พิเชฐ ม่วงนวล ที่ชี้แนะแนวทาง และให้ความรู้ในด้านต่างๆ ในการทำวิจัย
ตลอดจนช่วยตรวจและแก้ไขวิทยานิพนธ์ให้เป็นไปอย่างสมบูรณ์

ขอขอบพระคุณ คุณพ่อ คุณแม่และพี่ๆทุกคนที่ให้โอกาสและสนับสนุนให้ข้าพเจ้าได้มี
โอกาสเล่าเรียนจนถึงระดับปริญญาโทบัณฑิต ที่ให้กำลังใจในการเล่าเรียนและทำวิทยานิพนธ์จน
สำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณคณาจารย์ และบุคลากรในภาควิชาวิศวกรรมวิศวกรรมโทรคมนาคมทุก
ท่านที่ให้คำปรึกษาและความช่วยเหลือในระหว่างการทำวิทยานิพนธ์

ขอขอบคุณนักศึกษาปริญญาโทสาขาวิชาวิศวกรรมโทรคมนาคมทุกท่านที่ได้ให้คำแนะนำ
และเป็นกำลังใจมาโดยตลอด

อรุณี กาลวิบูลย์

สารบัญ

| | หน้า |
|--|------|
| บทคัดย่อ ภาษาไทย..... | I |
| บทคัดย่อ ภาษาอังกฤษ..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ..... | IV |
| สารบัญตาราง..... | VII |
| สารบัญรูป..... | VIII |
| รายการสัญลักษณ์..... | X |
| บทที่ 1 บทนำ..... | 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 |
| บทที่ 2 ทฤษฎีและหลักการ..... | 4 |
| 2.1 การสื่อสารเสียง..... | 4 |
| 2.1.1 ขั้นตอนในการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต..... | 4 |
| 2.1.2 การบีบอัดเสียง (Audio Compression)..... | 5 |
| 2.1.3 การบีบอัดเสียงด้วย G.723.1..... | 6 |
| 2.1.4 ปัจจัยที่มีผลกระทบต่อคุณภาพของเสียง..... | 6 |
| 2.1.4.1 เวลาหน่วง..... | 7 |
| 2.1.4.2 จิตเตอร์ (Jitter)..... | 7 |
| 2.1.4.3 การสูญหายของแพ็กเก็ต (Packet Loss)..... | 7 |
| 2.1.4.4 แบนด์วิดท์ที่มีจำกัด (Limited bandwidth)..... | 8 |
| 2.1.5 โพรโทคอลที่ใช้ในการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต..... | 8 |
| 2.1.5.1 สถาปัตยกรรมเครือข่ายรูปแบบ OSI | 8 |
| 2.1.5.2 ซิกแนลลิง โพรโทคอล(Signaling Protocol)..... | 10 |

สารบัญ (ต่อ)

| | หน้า |
|---|------|
| 2.2 Real-time Transport Protocol (RTP)..... | 12 |
| 2.2.1 บทบาทของ RTP..... | 12 |
| 2.2.2 โครงสร้างแพ็คเกจของ RTP (RTP Packet Structure)..... | 13 |
| 2.2.3 RTP Control Protocol (RTCP)..... | 14 |
| 2.2.4 การส่งข้อมูลซ้ำของเสียงโดยใช้โปรโตคอล RTP..... | 17 |
| 2.3 เทคนิคการพยากรณ์..... | 18 |
| 2.3.1 ประเภทของเทคนิคการพยากรณ์..... | 19 |
| 2.3.1.1 เทคนิคพยากรณ์เชิงคุณภาพ..... | 19 |
| 2.3.1.2 เทคนิคพยากรณ์เชิงปริมาณ..... | 19 |
| 2.3.1.2.1 เทคนิคอนุกรมเวลา..... | 19 |
| 2.3.1.2.2 เทคนิคความสัมพันธ์ของข้อมูล..... | 19 |
| 2.4 เทคนิควิธีกำลังสองน้อยที่สุด (Least Square Method)..... | 20 |
| 2.4.1 การประมาณค่าโดยวิธีกำลังสองน้อยที่สุด..... | 20 |
| 2.4.2 เทคนิคความสัมพันธ์ของข้อมูล..... | 22 |
| 2.4.2.1 ข้อมูลมีความสัมพันธ์แบบเชิงเส้น (Linear Regression)..... | 22 |
| 2.4.2.2 ข้อมูลมีความสัมพันธ์แบบพหุนาม (Polynomial Regression)..... | 24 |
| 2.4.2.3 ข้อมูลมีความสัมพันธ์แบบหลายเชิง (Multiple Regression)..... | 24 |
| 2.4.3 การตรวจสอบความเหมาะสมของฟังก์ชัน..... | 26 |
| บทที่ 3 การปรับปรุงและการทำแบบจำลองของอัลกอริทึมแบบปรับตัว..... | 28 |
| 3.1 การควบคุมความผิดพลาดแบบปรับตัว..... | 28 |
| 3.1.1 ประเภทคือ FEC..... | 28 |
| 3.1.2 อัตราการสูญหายของข้อมูลเสียงเมื่อมีการใช้ FEC..... | 30 |
| 3.2 อัลกอริทึมแบบปรับตัวที่ใช้ในการศึกษา..... | 32 |
| 3.2.1 อัลกอริทึม Bolot..... | 32 |
| 3.2.2 อัลกอริทึม RCCS..... | 34 |
| 3.2.3 อัลกอริทึม CNR..... | 38 |
| 3.3 การประยุกต์ใช้วิธีกำลังสองน้อยที่สุดในอัลกอริทึมแบบปรับตัว CNR..... | 42 |

สารบัญ (ต่อ)

| | หน้า |
|--|------|
| 3.4 การตรวจสอบความถูกต้องของแบบจำลอง..... | 43 |
| 3.4.1 การตรวจสอบการคำนวณค่า L_p และ L_q ของผู้รับ..... | 44 |
| 3.4.2 การตรวจสอบการทำงานของอัลกอริทึม..... | 46 |
| 3.4.2.1 การตรวจสอบการทำงานของอัลกอริทึม Bolot..... | 47 |
| 3.4.2.2 การตรวจสอบการทำงานของอัลกอริทึม CNR..... | 48 |
| บทที่ 4 ผลการทดลองและการวิเคราะห์..... | 50 |
| 4.1 การทดลองเพื่อประเมินผลอัลกอริทึม CNR..... | 50 |
| 4.1.1 การทดลองตอนที่ 1..... | 51 |
| 4.1.2 การทดลองตอนที่ 2..... | 53 |
| 4.2 ผลการทดลอง..... | 53 |
| 4.2.1 ผลการทดลองตอนที่ 1..... | 53 |
| 4.2.2 ผลการทดลองตอนที่ 2..... | 55 |
| 4.3 การวิเคราะห์..... | 60 |
| บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ..... | 62 |
| 5.1 สรุปผลการวิจัย..... | 62 |
| 5.2 ข้อเสนอแนะ..... | 62 |
| เอกสารอ้างอิง..... | 64 |
| ผลงานที่ตีพิมพ์..... | 67 |
| ประวัติผู้เขียน..... | 72 |

สารบัญตาราง

| ตาราง | หน้า |
|---|------|
| 2.1 เปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดข้อมูลเสียง..... | 6 |
| 3.1 Combination ที่ใช้อัลกอริทึม Bolot..... | 32 |
| 3.2 ค่า Reward และ Penalty ของแต่ละ Combination ในอัลกอริทึม RCCS..... | 36 |
| 3.3 คำอธิบายของพารามิเตอร์ที่ใช้ในอัลกอริทึม RCCS..... | 36 |
| 3.4 Combination ของข้อมูลซ้ำที่ใช้ในอัลกอริทึม CNR..... | 40 |
| 3.5 การตรวจสอบความถูกต้องในการคำนวณค่า L_b ของผู้รับในแบบจำลอง..... | 45 |
| 3.6 การตรวจสอบความถูกต้องในการคำนวณค่า L_a ของผู้รับในแบบจำลอง..... | 46 |
| 4.1 Combination ของข้อมูลซ้ำที่ใช้ในอัลกอริทึม CNR | 51 |
| 4.2 แสดงค่าพารามิเตอร์ที่สำคัญจาก Trace file เปรียบเทียบ 3 วิธีการ..... | 60 |

สารบัญรูป

| รูป | หน้า |
|---|------|
| 2.1 การส่งเสียงพูดในรูปแบบดิจิทัลบนเครือข่ายอินเทอร์เน็ต..... | 5 |
| 2.2 การแก้ไข Jitter ที่เกิดกับแพ็กเก็ตข้อมูลเสียงโดยการใช้บัฟเฟอร์ข้อมูลเสียงที่ภาครับ..... | 7 |
| 2.3 สถาปัตยกรรมเครือข่ายรูปแบบ OSI..... | 9 |
| 2.4 โครงสร้างโปรโตคอลสำหรับระบบโทรศัพท์บนโครงข่ายอินเทอร์เน็ต..... | 10 |
| 2.5 การส่ง Signaling และ Media ผ่านโครงข่ายอินเทอร์เน็ต..... | 11 |
| 2.6 บทบาทของ RTP ในการสื่อสารเสียงและวิดีโอบนเครือข่ายอินเทอร์เน็ต..... | 13 |
| 2.7 โครงสร้างแพ็กเก็ตของ RTP..... | 13 |
| 2.8 โครงสร้างเฮดเดอร์ของ RTP..... | 13 |
| 2.9 โครงสร้างเฮดเดอร์ของแพ็กเก็ต Receiver Report (RR)..... | 16 |
| 2.10 เฮดเดอร์ของบล็อกในแพ็กเก็ต RTP ที่มีข้อมูลซ้ำของเสียง..... | 17 |
| 2.11 ตัวอย่างแพ็กเก็ต RTP เมื่อมีการส่งข้อมูลซ้ำของเสียง 1 ชุด..... | 18 |
| 2.12 แสดงการหาค่าเบี่ยงเบนของวิธีกำลังสองน้อยที่สุด..... | 21 |
| 2.13 ข้อมูลที่มีลักษณะไม่เป็นเชิงเส้นแบบต่างๆ..... | 23 |
| 3.1 Media Independent FEC..... | 29 |
| 3.2 Media Specific FEC..... | 29 |
| 3.3 การกู้คืนโดยใช้ข้อมูลซ้ำสามารถทำให้ค่า L_u น้อยกว่า L_b | 31 |
| 3.4 การเพิ่มจำนวนข้อมูลซ้ำสามารถช่วยลดค่า L_u ได้..... | 31 |
| 3.5 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม Bolot..... | 33 |
| 3.6 การเพิ่มและลด Combination ในอัลกอริทึม Bolot..... | 34 |
| 3.7 Pseudo Code อัลกอริทึม Bolot..... | 34 |
| 3.8 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม RCCS..... | 35 |
| 3.9 Pseudo Code ของอัลกอริทึม RCCS..... | 37 |
| 3.10 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม CNR..... | 38 |
| 3.11 การระบุค่า L_b และ L_u ในแพ็กเก็ต Receiver Report..... | 39 |
| 3.12 Pseudo Code ของอัลกอริทึม CNR..... | 41 |
| 3.13 การเปลี่ยนแปลงการลด Combination..... | 42 |

สารบัญรูป (ต่อ)

| รูป | หน้า |
|--|------|
| 3.14 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง..... | 44 |
| 3.15 ตัวอย่างการหาค่า L_a เมื่อกำหนดรูปแบบการตัดทิ้งแพ็กเก็ตแบบ D02..... | 46 |
| 3.16 ค่า L_a และค่า L_b จากการคำนวณในระหว่างการจำลองโดยใช้อัลกอริทึม Bolot..... | 48 |
| 3.17 การเปลี่ยนแปลงของ Combination ในระหว่างการจำลองโดยใช้อัลกอริทึม CNR..... | 49 |
| 4.1 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง..... | 52 |
| 4.2 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม Bolot..... | 54 |
| 4.3 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ SES ในการหาค่า reward..... | 54 |
| 4.4 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการหาค่า reward..... | 55 |
| 4.5 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม Bolot (TCP 20 โหนด)..... | 56 |
| 4.6 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ SES ในการหาค่า reward (TCP 20 โหนด)..... | 56 |
| 4.7 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการหาค่า reward (TCP 20 โหนด)..... | 57 |
| 4.8 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม Bolot..... | 57 |
| 4.9 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ SES ในการหาค่า reward (TCP 40 โหนด)..... | 58 |
| 4.10 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการหาค่า reward (TCP 40 โหนด)..... | 58 |
| 4.11 เปรียบเทียบการเปลี่ยนแปลง combination number ของอัลกอริทึม Bolot และอัลกอริทึม CNR ซึ่งใช้เทคนิค SES และ Least square method ในการหาค่า reward..... | 59 |
| 4.12 แสดงความสัมพันธ์ระหว่างค่า Reward กับค่า Combination number ของวิธีการ Least Square กับวิธีการ Bolot..... | 61 |

รายการสัญลักษณ์

| | |
|--------|---|
| ACELP | Algebraic Code-Excited Linear Prediction |
| AIMD | Additive Increase Multiplicative Decrease |
| API | Application Programming Interface |
| CNR | Centre Network Research |
| 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 |
| LSM | Least Square Method |
| 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 |
| SES | Single Exponential Smoothing |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การใช้บริการสื่อสารเสียงผ่านช่องทางสื่อสาร (Media) มีผู้ใช้งานจำนวนมากพยายามหลีกเลี่ยงการส่งผ่านข้อมูลเสียงบนเครือข่ายแบบเซอร์กิตสวิทช์ (Circuit switched Network) เพราะมีค่าใช้จ่ายสูงและการใช้บริการเสียงผ่านเครือข่ายอินเทอร์เน็ตจึงได้รับความนิยมและแพร่หลายเนื่องจากมีค่าใช้จ่ายต่ำ ซึ่งการส่งข้อมูลทางอินเทอร์เน็ตนั้น จะส่งไปแบบแพ็กเก็ต (packet) เนื่องจากการส่งในลักษณะนี้ข้อมูลจะถูกทำการแยกออกเป็นบล็อกข้อมูล และส่งไปในเส้นทางที่แตกต่างกัน ซึ่งแต่ละเส้นทางนั้นจะเป็นเส้นทางที่เร็วที่สุด ที่จะไปถึงปลายทาง และจะถูกนำมารวมกันอีกครั้งที่ปลายทาง เพราะว่าเครือข่ายอินเทอร์เน็ตเป็นเครือข่ายที่ไม่มีการรับประกันคุณภาพของการบริการ (Quality of Service) ดังนั้นจึงไม่แน่ใจได้ว่าแพ็กเก็ต (Packet) ที่บรรจุข้อมูลของเสียง จะมีการสูญหายในระหว่างการสื่อสารผ่านช่องทางสื่อสาร ถ้าหากมีแพ็กเก็ตที่สูญหายมากเกินไปก็จะส่งผลกระทบต่อคุณภาพการสื่อสารที่ผู้รับได้รับ ตัวอย่างเช่น การสูญหายของแพ็กเก็ตเสียงทำให้เสียงไม่ต่อเนื่องและฟังไม่ชัดเจน

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อศึกษาวิธีการในการควบคุมความผิดพลาดของการสื่อสารเสียงผ่านอินเทอร์เน็ตที่สามารถปรับตัวได้ตามสภาพของกราฟฟิคภายในเครือข่ายโดยปรับปรุงอัตราการกู้คืนของแพ็กเก็ตข้อมูลเสียงที่สูญหายให้ดีขึ้น ซึ่งจะเป็นการเพิ่มประสิทธิภาพโดยรวมของบริการสื่อสารเสียงผ่านอินเทอร์เน็ต

1.3 สมมติฐานของการศึกษา

เมื่อผู้ส่งมีการตรวจสอบคุณภาพของเสียงที่ผู้รับได้รับผ่านอินเทอร์เน็ตแล้วจะสามารถปรับปรุงให้มีคุณภาพใกล้เคียงกับการสื่อสารผ่านเครือข่ายแบบเซอร์กิตสวิทช์ ผู้ส่งจะต้องมีกลไกในการควบคุมความผิดพลาดของแพ็กเก็ตเสียงที่สามารถปรับตัวได้ตามสภาพการใช้งานของเครือข่าย คือถ้าเครือข่ายมีความคับคั่งของกราฟฟิคต่ำก็ใช้วิธีการกำหนดพารามิเตอร์ในการส่งข้อมูลเสียงในรูปแบบหนึ่ง แต่ถ้าเครือข่ายมีความคับคั่งของกราฟฟิคสูงก็กำหนดพารามิเตอร์ของการส่งในอีกรูปแบบหนึ่ง เพื่อให้คุณภาพของเสียงที่ผู้รับได้รับอยู่ในระดับที่สามารถจะยอมรับได้ และมีการใช้ทรัพยากรของระบบอย่างมีประสิทธิภาพ

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

วิธีการควบคุมความผิดพลาดแบบปรับตัวที่ศึกษานั้นเป็นวิธีการที่ใช้พื้นฐานของ Forward Error Correction (FEC) แบบ Media Specific ซึ่งมีอัลกอริทึมที่ใช้ในการเลือกรูปแบบการการบรรจุข้อมูลซ้ำในแต่ละแพ็กเก็ตหรือที่เรียกว่า Combination ให้เหมาะสมกับสภาพของเครือข่าย โดยการนำวิธี Least Square Method (LSM) มาใช้ในการประมาณค่า พารามิเตอร์ (parameter) สำคัญ ที่ใช้ในการกำหนดรูปแบบ และปริมาณข้อมูลซ้ำ (Combination) ที่ใช้ในอัลกอริทึมแบบปรับตัว (adaptive error control algorithm) เพื่อเพิ่มประสิทธิภาพและให้สอดคล้องกับสภาพของเครือข่ายในขณะนั้น

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

เนื่องจากอัลกอริทึมแบบปรับตัว ที่ใช้ในการส่งข้อมูลซ้ำเพื่อควบคุมความผิดพลาดในการส่งข้อมูลแบบเดิมใช้ค่าคงที่ปรับให้เรียบและการประมาณค่าแบบเอ็กซ์โพเนนเชียล ในการกำหนดรูปแบบการส่งข้อมูลซ้ำ ดังนั้นจึงได้ประยุกต์ ใช้วิธีการ Least Square Method (LSM) ในการประมาณค่าพารามิเตอร์สำคัญ ในการกำหนดรูปแบบ และปริมาณข้อมูลซ้ำ เพื่อให้การควบคุมการส่งข้อมูลซ้ำมีความเหมาะสม กับสภาพเครือข่ายและช่วยให้มีการใช้ทรัพยากรของระบบอย่างคุ้มค่า

1.6 ขอบเขตของการวิจัย

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

1.7 ขั้นตอนของการศึกษา

- 1) ศึกษาหลักการของการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต
- 2) ศึกษาวิธีการในการควบคุมความผิดพลาดของการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต ที่ได้มีการตีพิมพ์ลงในบทความทางวิชาการ
- 3) ศึกษาอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวสำหรับการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต ในกรณีที่ใช้การบีบอัดเสียงแบบ G.723.1 และทดสอบอัลกอริทึมโดยใช้แบบจำลอง

- 4) ศึกษาบทความทางวิชาการเกี่ยวกับเทคนิคการปรับให้เรียบแบบเอกซ์โพเนนเชียลในอัลกอริทึมควบคุมความผิดพลาดของการสื่อสารเสียงที่สามารถปรับตัวได้ตามสภาพของเครือข่าย
- 5) นำเทคนิค Least Square Method มาพัฒนาในอัลกอริทึมควบคุมความผิดพลาดของการสื่อสารเสียงที่สามารถปรับตัวได้ตามสภาพของเครือข่าย
- 6) ทดสอบวิธีการควบคุมความผิดพลาดแบบปรับตัวสำหรับการสื่อสารเสียงที่ได้ออกแบบไว้ โดยใช้การทดสอบบนโปรแกรมจำลองเครือข่ายอินเทอร์เน็ต
- 7) รวบรวมผลการทดลองและจัดทำวิทยานิพนธ์

บทที่ 2

ทฤษฎีและหลักการ

ในบทนี้เป็นการกล่าวถึงทฤษฎีและหลักการพื้นฐานสำหรับการทำวิทยานิพนธ์ โดยเนื้อหาที่จะกล่าวถึงในบทนี้ประกอบด้วย หัวข้อ 2.1 การสื่อสารเสียง ซึ่งอธิบายถึงขั้นตอนในการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต การบีบอัดเสียงโดยเน้นที่การบีบอัดเสียงด้วย G.723.1 หัวข้อ 2.2 กล่าวถึงโปรโตคอลที่ใช้สำหรับการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ตซึ่งก็คือ โปรโตคอลที่ชื่อว่า Real-time Transport Protocol (RTP) หัวข้อ 2.3 เทคนิคการพยากรณ์ และสุดท้ายคือ หัวข้อ 2.4 เทคนิควิธีกำลังสองน้อยที่สุด (Least Square Method)

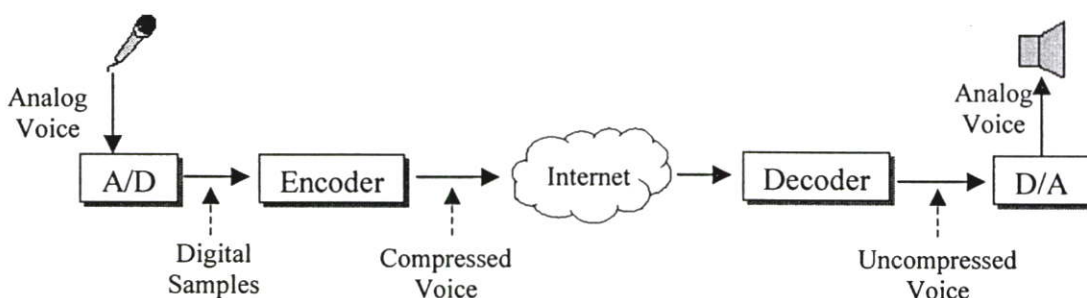
2.1 การสื่อสารเสียง

หลักการพื้นฐานของการสื่อสารเสียงที่จะกล่าวถึงในหัวข้อนี้หมายถึงการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต ซึ่งจำเป็นจะต้องมีการแปลงสัญญาณเสียงจากอนาล็อก (Analog) ให้เป็นดิจิทัล (Digital) รวมทั้งต้องมีการลดขนาดของข้อมูลเสียงก่อนจะบรรจุลงในแพ็กเก็ตและส่งให้ผู้รับ และนอกจากนี้ผู้รับจะต้องเข้าใจข้อมูลที่ได้รับจากผู้ส่ง มิฉะนั้นผู้รับก็จะไม่สามารถนำข้อมูลไปประมวลผลได้ถูกต้อง ดังนั้นทั้งผู้รับและผู้ส่งจึงต้องใช้มาตรฐานในการสื่อสารเดียวกัน

2.1.1 ขั้นตอนในการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต

ในการส่งเสียงพูดบนเครือข่ายอินเทอร์เน็ต ผู้ส่งจะต้องเปลี่ยนสัญญาณเสียงจากไมโครโฟนที่อยู่ในรูปแบบของสัญญาณอนาล็อกให้เป็นดิจิทัลเสียงก่อน โดยใช้ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล (Analog to Digital Converter, A/D) ดังแสดงในรูปที่ 2.1 โดยปกติแล้วเสียงพูดของมนุษย์จะมีความถี่ไม่เกิน 4 กิโลเฮิร์ตซ์ (kHz) [1] ดังนั้นอัตราการสุ่มตัวอย่าง (Sampling Rate) ตามทฤษฎีของไนควิสต์ก็จะมีค่าเท่ากับ 8 กิโลเฮิร์ตซ์ (2 เท่าของความถี่สูงสุด) ซึ่งในระบบโทรศัพท์สาธารณะ (Public Switched Telephone Network, PSTN) ก็ใช้อัตราการสุ่มตัวอย่างเท่ากับค่านี้เช่นกัน โดยในระบบโทรศัพท์ PSTN นั้นใช้วิธีการที่เรียกว่า Pulse Code Modulation (PCM) ในการแปลงระดับสัญญาณเสียงแบบอนาล็อกให้กลายเป็นค่าดิจิทัลที่เรียกว่า ค่าตัวอย่าง (Sample) โดยในกรณีที่ใช้อุปกรณ์เป็นเครื่องมือในการสื่อสารเสียงนั้น หน้าที่นี้เป็นของการ์ดเสียง เมื่อได้ข้อมูลเสียงในรูปแบบของดิจิทัลแล้วก็สามารถส่งข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ตไปให้กับผู้รับได้ แต่เพื่อไม่ให้ปริมาณข้อมูลเสียงที่ส่งผ่านเครือข่ายมีมากเกินไปจึงต้องมีการบีบอัดข้อมูลเสียงเสียงก่อน (Audio Compression) โดยผู้ส่งจะใช้ตัวเข้ารหัส (Encoder) ที่มีวิธีการเข้ารหัสตามมาตรฐานการบีบอัดที่เลือกใช้ ซึ่งรายละเอียดเกี่ยวกับการบีบอัดเสียงจะได้กล่าวถึงในหัวข้อถัดไป

ส่วนทางฝั่งของผู้รับก็จะต้องนำแฟ้มเกิดเสียงที่ได้รับมาถอดรหัส จากนั้นก็แปลงสัญญาณเสียงจากดิจิทัลให้กลายเป็นอนาล็อกและส่งสัญญาณเสียง ไปเล่น (Play) ออกทางลำโพง



รูปที่ 2.1 การส่งเสียงพูดในรูปแบบดิจิทัลบนเครือข่ายอินเทอร์เน็ต [1]

2.1.2 การบีบอัดเสียง (Audio Compression)

ในการส่งเสียงพูดผ่านเครือข่ายอินเทอร์เน็ตนั้น เพียงแต่การแปลงจากสัญญาณเสียงแบบอนาล็อกให้กลายเป็นข้อมูลเสียงแบบดิจิทัลนั้นยังไม่เพียงพอ สังเกตว่าในการแปลงสัญญาณเสียงจากอนาล็อกเป็นดิจิทัลซึ่งใช้อัตราการชักตัวอย่าง 8000 เฮิรตซ์ (ใน 1 วินาทีมีการชักตัวอย่าง 8000 ครั้ง) ถ้าหากใช้ข้อมูลดิจิทัล 8 บิตต่อหนึ่งค่าตัวอย่าง แบนด์วิดท์ที่ใช้ในการส่งข้อมูลก็จะเป็น 64 กิโลบิตต่อวินาที (kbps) และถ้าใช้ 16 บิตต่อค่าตัวอย่าง แบนด์วิดท์ก็เพิ่มเป็น 128 กิโลบิตต่อวินาที ซึ่งถือว่าเป็นค่าที่สูงทีเดียว ยิ่งไปกว่านั้นถ้าหากมีผู้ที่ต้องการสื่อสารเสียงพร้อมกันหลายคู่สนทนา เครือข่ายที่มีแบนด์วิดท์อย่างจำกัดก็จะไม่สามารถรองรับได้ ผลที่ตามมาคือผู้รับก็จะได้ยินเสียงไม่ชัดเจน เสียงไม่ต่อเนื่อง หรืออาจจะฟังไม่รู้เรื่องเลยก็ได้ ซึ่งวิธีแก้ไขก็คือต้องมีการลดขนาดของข้อมูลเสียงก่อนที่จะส่งข้อมูลผ่านเครือข่าย เทคนิคนี้เรียกว่า การบีบอัดเสียง

ในการบีบอัดเสียงนั้น วิธีการถอดรหัสจะต้องสอดคล้องกับวิธีการเข้ารหัส มิฉะนั้นจะไม่สามารถนำมาเสียงมาเล่นออกทางลำโพงได้ถูกต้อง สหภาพโทรคมนาคมนานาชาติ (International Telecommunication Union, ITU) ได้มีการกำหนดมาตรฐานสำหรับการบีบอัดเสียง (Audio Codec) เอาไว้หลายมาตรฐาน ตารางที่ เป็นการเปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดเสียงแต่ละประเภท โดยในคอลัมน์ของ Quality MOS (Mean Opinion Score) [2] เป็นค่าที่แสดงคุณภาพของเสียง ค่าสูงสุดมีค่าเท่ากับ 5 จะเห็นว่า G.711[3] นั้นมีคุณภาพสูงที่สุดก็จริง แต่ก็มีอัตราบิต (Bit Rate) สูงถึง 64 kbps ทำให้ไม่สามารถนำมาใช้งานได้จริงถ้าเครือข่ายมีแบนด์วิดท์ต่ำกว่านี้ เช่น ในการเชื่อมต่ออินเทอร์เน็ตผ่านทางเครือข่ายโทรศัพท์ซึ่งมีความเร็วอยู่ที่ 28.8 ถึง 56 กิโลบิตต่อวินาที เท่านั้น ส่วน G.729[4] และ G.723.1 ก็มีคุณภาพใกล้เคียงกับ G.711 แต่มีอัตราบิตน้อยกว่ามาตรฐานทั้งสองจึงน่าจะเป็นทางเลือกที่ดีมาตรฐานอื่น

ตารางที่ 2.1 เปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดข้อมูลเสียง[1]

| 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

2.1.3 การบีบอัดเสียงด้วย G.723.1

เนื่องจากในวิทยานิพนธ์นี้ได้เลือกใช้ G.723.1[5] 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

2.1.4 ปัจจัยที่มีผลกระทบต่อคุณภาพของเสียง

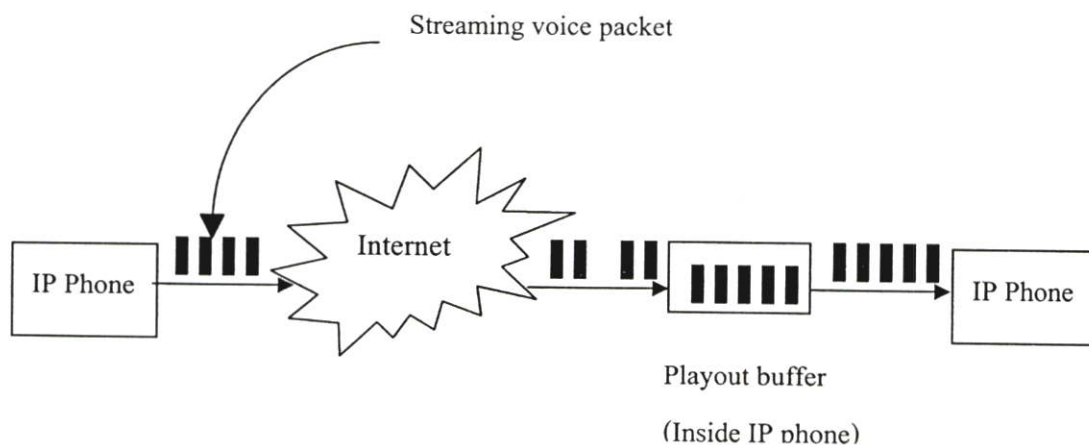
เนื่องจากเครือข่ายอินเทอร์เน็ตไม่ได้มีการรับประกันคุณภาพการบริการ ดังนั้นการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ตจึงมีโอกาสที่คุณภาพของเสียงที่ผู้รับได้รับจะต่ำกว่าคุณภาพของเสียงต้นฉบับได้ ซึ่งปัจจัยที่มีผลกระทบต่อคุณภาพของเสียงมีดังนี้

2.1.4.1 เวลาหน่วง

เวลาหน่วง คือ เวลาในการเดินทางของเสียงจากไมโครโฟนของผู้ส่งไปถึงลำโพงของผู้รับ ในระบบโทรศัพท์ PSTN จะไม่เกิดปัญหานี้เนื่องจากการโทรแต่ละครั้ง จะมีการจองทรัพยากรสำหรับแต่ละคู่สายเอาไว้และค่าเวลาหน่วงจะน้อยมาก ประมาณ 50 – 70 มิลลิวินาที [6] ในขณะที่เครือข่ายอินเทอร์เน็ตไม่สามารถที่จะรับประกันได้ว่าค่าเวลาหน่วงเป็นเท่าใด โดยปกติคนเราจะรู้สึกถึงผลกระทบของเวลาหน่วงเมื่อค่าเวลาหน่วงสูงกว่า 250 มิลลิวินาที [7]

2.1.4.2 จิตเตอร์ (Jitter)

จิตเตอร์เกิดจากการที่เวลาหน่วงของแต่ละแพ็กเก็ตไม่เท่ากัน ทั้งนี้เนื่องจากการเดินทางของแพ็กเก็ตบนเครือข่ายอินเทอร์เน็ตนั้น แต่ละแพ็กเก็ตอาจจะใช้เส้นทางที่ต่างกันได้แม้จะมีปลายทางที่เดียวกัน รวมถึงสภาพความคับคั่งของเครือข่ายที่เปลี่ยนแปลงได้ตลอดเวลาที่สามารถทำให้เกิดจิตเตอร์ได้เช่นกัน ปัญหานี้จึงทำให้สัญญาณเสียงที่ได้รับมีอาการกระตุกและฟังได้ไม่ชัดเจน ดังนั้นจึงต้องมีการใช้บัฟเฟอร์เพื่อรวบรวมข้อมูลก่อนเพื่อให้ข้อมูลมีความต่อเนื่อง แล้วจึงส่งไปประมวลผลและเล่นออกทางลำโพงต่อไป การใช้บัฟเฟอร์ขนาดใหญ่จะสามารถกำจัดจิตเตอร์ได้ดีขึ้น แต่ก็มีผลทำให้มีเวลาหน่วงมากขึ้น แต่ถ้าใช้บัฟเฟอร์ขนาดเล็กเกินไปจะทำให้กำจัดจิตเตอร์ได้ไม่ดีนัก ดังนั้นจึงจำเป็นต้องหาวิธีที่ใช้ในการปรับขนาดของบัฟเฟอร์ ให้เหมาะสม



รูปที่ 2.2 การแก้ไข Jitter ที่เกิดกับแพ็กเก็ตข้อมูลเสียงโดยการใช้บัฟเฟอร์ข้อมูลเสียงที่ภาครับ

2.1.4.3 การสูญหายของแพ็กเก็ต (Packet Loss)

การที่แพ็กเก็ตสูญหายนั้นส่วนใหญ่มีสาเหตุมาจากความคับคั่งของเครือข่าย เพราะในขณะที่เราเตอร์ทำการประมวลผลแพ็กเก็ตหนึ่งอยู่ ถ้ามีอีกแพ็กเก็ตเข้ามาจะถูกเก็บไว้ในบัฟเฟอร์ของคิว และถ้าบัฟเฟอร์นั้นเต็มจะทำให้แพ็กเก็ตที่เข้ามาที่หลังถูกตัดทิ้ง ถึงแม้ว่าในการสื่อสารเสียงจะยอมให้มีการสูญหายของแพ็กเก็ตได้บ้าง แต่การที่มีแพ็กเก็ตสูญหายมากเกินไปก็อาจส่งผลกระทบต่อ

คุณภาพเสียงได้ โดยการสูญหายของแพ็กเก็ตจะมีผลกระทบอย่างชัดเจนเมื่อมีอัตราการสูญหายมากกว่า 5% [6][7]

2.1.4.4 แบนด์วิดท์ที่มีจำกัด (Limited bandwidth)

แบนด์วิดท์เป็นส่วนที่สำคัญมากในการสื่อสารเสียง ซึ่งปริมาณการใช้แบนด์วิดท์นั้นก็ขึ้นอยู่กับอัลกอริทึมในการบีบอัดข้อมูลเสียง ถ้าแบนด์วิดท์ที่ใช้ในขณะนั้นไม่เพียงพอจะทำให้เกิดปัญหาต่างๆ ตามมา ไม่ว่าจะเป็นเวลาหน่วง จิตเตอร์ และสูญหายของแพ็กเก็ต

2.1.5 โพรโทคอลที่ใช้ในการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต

2.1.5.1 สถาปัตยกรรมเครือข่ายรูปแบบ OSI

จุดมุ่งหมายของการกำหนดมาตรฐานรูปแบบ OSI ก็เพื่อเป็นการกำหนดการแบ่งโครงสร้างของสถาปัตยกรรม เครือข่ายออกเป็นเลเยอร์ๆ (Layer) และกำหนดหน้าที่การทำงานในแต่ละเลเยอร์ รวมถึงกำหนดรูปแบบการอินการอินเทอร์เน็ตเฟสระหว่างเลเยอร์ด้วย โดยมีหลักเกณฑ์ในการกำหนดดังต่อไปนี้

1. ไม่แบ่งโครงสร้างออกเป็นเลเยอร์ๆ มากจนเกินไป
2. แต่ละเลเยอร์จะต้องมีหน้าที่การทำงานแตกต่างกันทั้งขบวนการและเทคโนโลยี
3. จัดกลุ่มหน้าที่การทำงานที่คล้ายกันให้อยู่เลเยอร์เดียวกัน
4. เลือกเฉพาะการทำงานที่เคยใช้ได้ผลประสบความสำเร็จมาแล้ว
5. กำหนดหน้าที่การทำงานเฉพาะง่ายๆ แก่เลเยอร์ เพื่อว่าต่อไปถ้ามีการออกแบบเลเยอร์ใหม่ หรือมีการเปลี่ยนแปลงโปรโตคอลใหม่ในอันที่จะทำให้สถาปัตยกรรมมีประสิทธิภาพดียิ่งขึ้น จะไม่มีผลทำให้อุปกรณ์ฮาร์ดแวร์ และซอฟต์แวร์ที่เคยใช้ได้ผลอยู่เดิมจะต้องเปลี่ยนแปลงตาม
6. กำหนดอินเทอร์เน็ตเฟสมาตรฐาน
7. ให้มีความยืดหยุ่นในการเปลี่ยนแปลงโปรโตคอลในแต่ละเลเยอร์
8. สำหรับเลเยอร์ย่อยของแต่ละเลเยอร์ให้ใช้หลักเกณฑ์เดียวกันกับที่กล่าวใน 7 ข้อแรก

รูปแบบ OSI แบ่งโครงสร้างออกเป็น 7 เลเยอร์ และในแต่ละเลเยอร์ได้มีการกำหนดหน้าที่การทำงานไว้ดังต่อไปนี้

1. Physical เป็นชั้นล่างสุดของการติดต่อสื่อสาร ทำหน้าที่ส่ง-รับข้อมูลจริงๆ จากช่องทางการสื่อสาร ระหว่างคอมพิวเตอร์เครื่องหนึ่งกับคอมพิวเตอร์เครื่องอื่นๆ มาตรฐานสำหรับเลเยอร์ชั้นนี้จะกำหนดว่าแต่ละคอนเนคเตอร์ เช่น RS-232-C มีกี่พิน แต่ละพินทำหน้าที่อะไรบ้าง ใช้สัญญาณไฟกี่โวลต์ เทคนิคการมัลติเพล็กซ์แบบต่างๆ ก็จะถูกกำหนดอยู่ในเลเยอร์ชั้นนี้

| | | |
|----------------|-----------------------|----------------|
| 7 Application | โพรโทคอล Application | 7 Application |
| 6 Presentation | โพรโทคอล Presentation | 6 Presentation |
| 5 Session | โพรโทคอล Session | 5 Session |
| 4 Transport | โพรโทคอล Transport | 4 Transport |
| 3 Network | โพรโทคอล Network | 3 Network |
| 2 Data Link | โพรโทคอล Data Link | 2 Data Link |
| 1 Physical | สายสื่อสาร | 1 Physical |

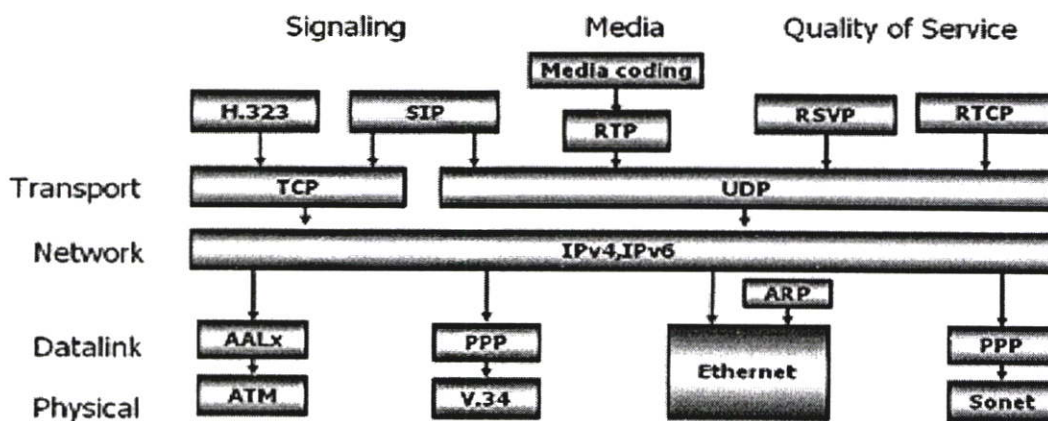
รูปที่ 2.3 สถาปัตยกรรมเครือข่ายรูปแบบ OSI

2. Data Link จะเป็นเสมือนผู้ตรวจสอบ หรือควบคุมความผิดพลาดในข้อมูล โดยจะแบ่งข้อมูลที่จะส่งออกเป็นเฟรมหรือแพคเกจถ้าผู้รับได้รับข้อมูลถูกต้องก็จะส่งสัญญาณยืนยันกลับมาว่าได้รับข้อมูลแล้วเรียกว่า สัญญาณ ACK (Acknowledge) ให้กับผู้ส่ง
3. Network มีหน้าที่เลือกเส้นทางที่ใช้เวลาในการสื่อสารน้อยที่สุด และระยะทางสั้นที่สุดด้วยข่าวสารที่รับมาจากเลเยอร์ชั้นที่4 จะถูกแบ่งออกเป็นแพคเกจๆ ในชั้นที่ 3 นี้
4. Transport บางครั้งเรียกว่า Host-to-Host ซึ่งจะทำหน้าที่ตรวจสอบว่าข้อมูลที่ส่งมาจากเลเยอร์ชั้น Session นั้นไปถึงปลายทางจริงๆหรือไม่ ดังนั้นการกำหนดตำแหน่งของข้อมูล (Address)จึงเป็นเรื่องสำคัญในชั้นนี้ เนื่องจากจะต้องรู้ว่าใครคือผู้ส่ง และใครคือผู้รับข้อมูลนั้น
5. Session ทำหน้าที่เชื่อมโยงระหว่างผู้ใช้งานกับคอมพิวเตอร์เครื่องอื่นๆ โดยผู้ใช้จะใช้คำสั่งหรือข้อความที่กำหนดไว้ป้อนเข้าไปในระบบ เลเยอร์ชั้นนี้ จะส่งข้อมูลทั้งหมดให้กับเลเยอร์ชั้น Transport เป็นผู้จัดการต่อไป ในบางเครือข่ายทั้งเลเยอร์ Session และเลเยอร์ Transport อาจจะเป็นเลเยอร์ชั้นเดียวกัน
6. Presentation ทำหน้าที่เหมือนบรรณารักษ์ กล่าวคือคอยรวบรวมข้อความ (Text) และแปลงรหัส หรือแปลงรูปแบบของข้อมูลให้เป็นรูปแบบการสื่อสารเดียวกัน เพื่อช่วยลดปัญหาต่างๆ ที่อาจจะเกิดขึ้นกับผู้ใช้งานในระบบ
7. Application เป็นเลเยอร์ชั้นบนสุดของรูปแบบ OSI ซึ่งเป็นชั้นที่ใช้ติดต่อกันระหว่างผู้ใช้โดยตรง ซึ่งได้แก่ โฮสต์คอมพิวเตอร์ เทอร์มินัลหรือคอมพิวเตอร์ PC เป็นต้น แอปพลิเคชันในเลเยอร์ชั้นนี้สามารถนำเข้า หรือออกจากระบบเครือข่ายได้โดยไม่จำเป็นต้องสนใจว่าจะมีขั้นตอนการทำงานอย่างไร

2.1.5.2 ซิกแนลิ่งโปรโตคอล (Signaling Protocol)

การสื่อสารข้อมูลเสียงระหว่างต้นทางกับปลายทางในระบบโทรศัพท์บนโครงข่ายอินเทอร์เน็ตนั้น จำเป็นต้องใช้ซิกแนลิ่งเพื่อควบคุมการสร้าง สิ้นสุด และเปลี่ยนแปลงเซสชันการสนทนาระหว่างต้นทาง กับปลายทาง ซึ่งในปัจจุบันมีซิกแนลิ่งสำหรับระบบโทรศัพท์อินเทอร์เน็ตที่นิยมใช้อยู่ 2 ชนิดด้วยกันคือ H.323 และ SIP (Session Initiation Protocol) แต่เนื่องจาก SIP ถูกพัฒนาขึ้นภายหลัง และมีแนวทางในการพัฒนาขึ้นเพื่อรองรับการทำงานบนโครงข่ายอินเทอร์เน็ตโดยตรง SIP จึงเป็นโปรโตคอล ที่มีความซับซ้อนน้อย สามารถเพิ่มเติมคุณลักษณะของโปรโตคอลได้ง่าย และสามารถทำงานบนโครงข่ายขนาดใหญ่ได้ดีกว่าเมื่อเปรียบเทียบกับโปรโตคอล H.323

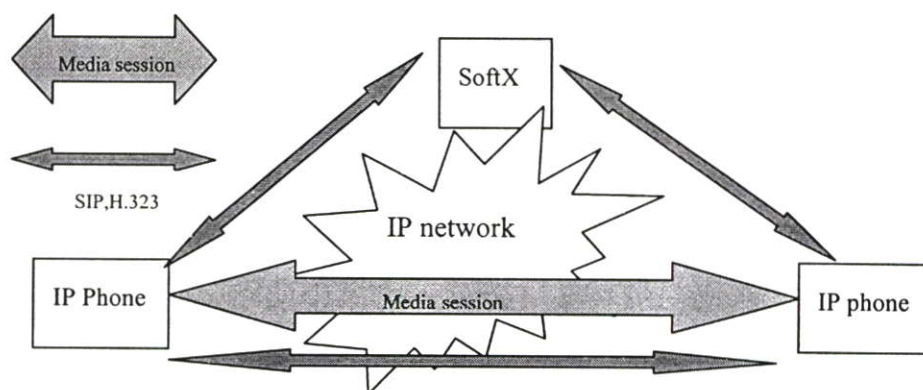
โครงสร้างโปรโตคอลของระบบโทรศัพท์อินเทอร์เน็ตดังแสดงในรูปที่ 2.4 โปรโตคอล H.323 และ SIP อยู่ในระดับชั้นประยุกต์ใช้งาน (Application layer) ทำงานบนโปรโตคอลในชั้นเคลื่อนย้ายที่ต่ำกว่า



รูปที่ 2.4 โครงสร้างโปรโตคอลสำหรับระบบโทรศัพท์บนโครงข่ายอินเทอร์เน็ต

การอ้างอิงที่อยู่ของโครงข่ายอินเทอร์เน็ตนั้นใช้ไอพีแอดเดรส แต่สำหรับระบบโทรศัพท์ การใช้ไอพีแอดเดรส เป็นตัวระบุเครื่องลูกข่ายของผู้ใช้โทรศัพท์นั้นไม่เหมาะสม เนื่องจากในการเชื่อมต่อโครงข่ายอินเทอร์เน็ตแต่ละครั้ง เครื่องลูกข่ายอาจได้ไอพีแอดเดรสที่ไม่เหมือนเดิม ดังนั้นจึงต้องมีแอดเดรสที่ใช้ระบุเครื่องลูกข่ายสำหรับการใช้งานโทรศัพท์อินเทอร์เน็ตโดยตรง ซึ่งในการใช้งานจริงภายในโครงข่ายต้องมี ซอฟต์แวร์ (SoftX) ทำงานเป็นเซิร์ฟเวอร์จัดการแมปไอพีแอดเดรส กับ แอดเดรสสำหรับระบบโทรศัพท์ของเครื่องลูกข่ายแต่ละเครื่อง และช่วยควบคุมเซสชันระหว่างต้นทางกับปลายทาง ดังแสดงในรูปที่ 2.5 เมื่อผู้ใช้งานเข้าใช้ระบบ เครื่องลูกข่ายของระบบโทรศัพท์อินเทอร์เน็ตจะลงทะเบียนกับ ซอฟต์แวร์ ก่อนโดยใช้ซิกแนลิ่งโปรโตคอล SIP

หรือ H.323 เมื่อ ซอฟต์แวร์สวิตช์ ได้รับการร้องขอลงทะเบียนจากเครื่องลูกข่าย ซอฟต์แวร์สวิตช์ จะเก็บข้อมูลไอพีแอดเดรส และแอดเดรสสำหรับระบบโทรศัพท์ของเครื่องลูกข่ายไว้ ดังนั้น ซอฟต์แวร์สวิตช์ จะมีข้อมูลของเครื่องลูกข่ายทุกเครื่องที่ได้เข้ามาลงทะเบียนไว้ เมื่อผู้ใช้งานต้องการสร้างเซสชันกับผู้ใช้ปลายทาง ผู้ใช้งานไม่จำเป็นต้องทราบไอพีแอดเดรสของเครื่องปลายทาง เพียงแต่ทราบแอดเดรสสำหรับโทรศัพท์อินเทอร์เน็ตปลายทาง เครื่องลูกข่ายต้นทางจะส่งข้อความร้องขอไปยังซอฟต์แวร์สวิตช์ โดยใช้ซิกเนลลิ่งโปรโตคอล SIP หรือ H.323 เมื่อซอฟต์แวร์สวิตช์ ได้รับข้อความร้องขอจากเครื่องลูกข่ายต้นทางซอฟต์แวร์สวิตช์ จะตรวจสอบแอดเดรสปลายทาง พร้อมกับค้นหาไอพีแอดเดรสของเครื่องลูกข่ายปลายทางจากฐานข้อมูล และส่งข้อความร้องขอนั้นต่อไปยังเครื่องลูกข่ายปลายทางตามไอพีแอดเดรสที่ได้ลงทะเบียนไว้ หรืออาจส่งไอพีแอดเดรสของเครื่องลูกข่ายปลายทางกลับไปให้เครื่องลูกข่ายต้นทางทราบเพื่อใช้ในการติดต่อต่อไป (ทั้งนี้ขึ้นอยู่กับโหมดการทำงาน of ซอฟต์แวร์สวิตช์ และซิกเนลลิ่งโปรโตคอล ที่ใช้) เมื่อเครื่องลูกข่ายทั้ง 2 ทราบไอพีแอดเดรสของกันและกันแล้ว ก็จะสามารถส่งซิกเนลลิ่ง รวมไปถึงข้อมูลเสียงหากันได้โดยตรง ทั้งนี้จะเห็นได้ว่าซอฟต์แวร์สวิตช์ เป็นอุปกรณ์ที่ประมวลผลซิกเนลลิ่ง เพื่อช่วยสร้างเซสชันในระบบโทรศัพท์อินเทอร์เน็ตเท่านั้น โดยข้อมูลเสียงไม่จำเป็นต้องวิ่งเข้าไปที่ซอฟต์แวร์สวิตช์ เลย



รูปที่ 2.5 การส่ง Signaling และ Media ผ่าน โครงข่ายอินเทอร์เน็ต

โปรโตคอลในระดับชั้น ทรานสปอร์ต (Transport Layer) ของเครือข่ายอินเทอร์เน็ตที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบันมีอยู่สองโปรโตคอลคือ TCP ซึ่งเป็นโปรโตคอลที่มีการรับประกันว่าสามารถส่งข้อมูลไปถึงปลายทางได้อย่างถูกต้องและครบถ้วน อีกโปรโตคอลหนึ่งก็คือ User Datagram Protocol (UDP) ซึ่งเป็นโปรโตคอลที่ไม่มีการรับประกันความน่าเชื่อถือของข้อมูล จะเห็นว่า TCP เป็นโปรโตคอลที่มีประสิทธิภาพมากกว่าและกลไกการทำงานซับซ้อนมากกว่า UDP แต่ TCP นั้นไม่เหมาะสำหรับการสื่อสารเสียง ทั้งนี้เนื่องจากความล่าช้าของกลไกการทำงาน of TCP นั้นเอง การสื่อสารเสียงนั้นต้องการความเร็ว โดยไม่จำเป็นว่าจะต้องส่งแพ็กเก็ตเสียงให้ไปถึงผู้รับอย่างครบถ้วน เพราะถึงแม้ว่าจะมีข้อมูลบางส่วนสูญหายไปบ้าง ก็ไม่ได้ส่งผลกระทบต่อ

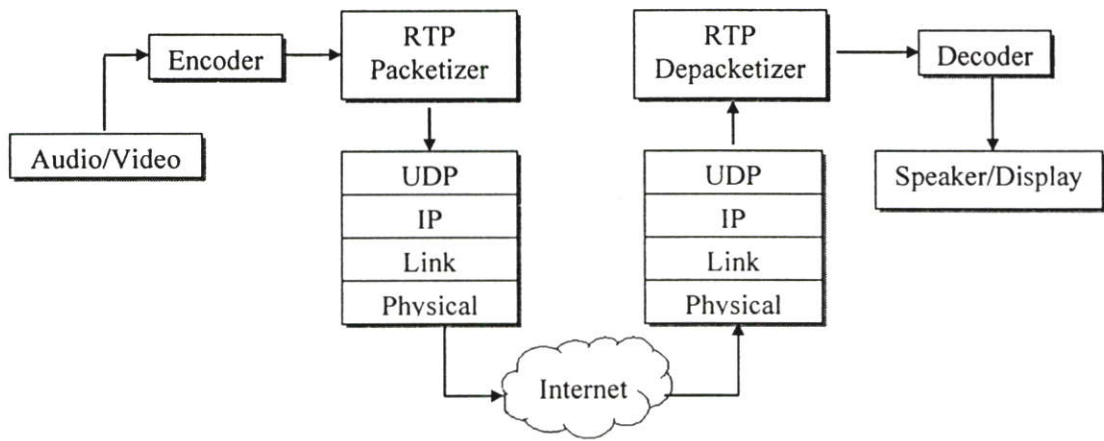
คุณภาพของเสียงมากนัก ดังนั้น UDP จึงเป็นทางเลือกที่ดีกว่า แต่การที่จะใช้ UDP อย่างเดียวนั้นยังไม่เพียงพอ จึงได้มีการพัฒนาโปรโตคอลตัวหนึ่งที่ทำงานอยู่เหนือโปรโตคอล UDP โปรโตคอลตัวนี้มีชื่อว่า Real-time Transport Protocol (RTP) ซึ่งองค์กรที่เสนอโปรโตคอลตัวนี้ขึ้นมาคือ Internet Engineering Task Force (IETF) โปรโตคอล RTP ได้รับการพัฒนาขึ้นมาเพื่อใช้ในการสื่อสารแบบเวลาจริง ซึ่งรายละเอียดของของโปรโตคอล RTP นี้จะกล่าวถึงอีกครั้งในหัวข้อที่ 2.2

2.2 Real-time Transport Protocol (RTP)

RTP เป็นโปรโตคอลที่ได้รับการพัฒนาโดย IETF และมีข้อกำหนดอยู่ใน RFC 3550 [8] RTP ถูกออกแบบมาเพื่อใช้เป็นโปรโตคอลในการส่งข้อมูลของการสื่อสารที่ทำงานในลักษณะเวลาจริง ในวิทยานิพนธ์นี้จึงเลือกใช้โปรโตคอล RTP ทั้งในการสื่อสารเสียงและการสื่อสารวิดีโอ หน้าที่สำคัญของ RTP คือ ทำให้ทางผู้รับสามารถนำข้อมูลที่ได้จากแพ็กเก็ตไปแสดงผลได้ถูกต้อง กลไกหลักที่ใช้ใน RTP ก็คือ การประทับเวลา (Time Stamping) และการกำหนดหมายเลขลำดับ (Sequence Numbering) ผู้ส่งจะกำหนดค่า Timestamp ซึ่งเป็นฟิลด์หนึ่งในเฮดเดอร์ของ RTP ตามเวลาที่ข้อมูลในไบต์แรกของแพ็กเก็ตถูกซัดตัวอย่าง ทางด้านของผู้รับเมื่อได้รับแพ็กเก็ตจะใช้ค่า Timestamp เพื่อกำหนดจังหวะที่ถูกต้องในการแสดงข้อมูล และยังสามารถใช้ในการประสานเวลาระหว่างสัญญาณเสียงและวิดีโอ ส่วนกลไกการกำหนดหมายเลขลำดับนั้นใช้ฟิลด์หนึ่งในเฮดเดอร์ RTP ที่ชื่อว่า Sequence Number โดยค่าของหมายเลขนี้จะเพิ่มขึ้นตามจำนวนแพ็กเก็ตที่ส่ง ซึ่งผู้รับจะใช้ข้อมูลนี้ในการเรียงลำดับและตรวจสอบการสูญหายของแพ็กเก็ต นอกจากนี้ RTP ยังให้ข้อมูลเกี่ยวกับวิธีการบีบอัดข้อมูลเสียงหรือวิดีโอ เพื่อให้ผู้รับสามารถแสดงผลข้อมูลได้ถูกต้อง และนอกจากข้อกำหนดเกี่ยวกับการใช้โปรโตคอลในการขนส่งข้อมูลของสื่อแล้ว ใน RFC 3550 ยังมีข้อกำหนดของโปรโตคอลหนึ่งคือ RTP Control Protocol (RTCP) ซึ่งเป็นโปรโตคอลที่ถูกนำมาใช้งานร่วมกับ RTP โดยโปรโตคอล RTCP จะถูกนำไปใช้ในการรายงานเกี่ยวกับสถิติและคุณภาพการบริการ

2.2.1 บทบาทของ RTP

รูปที่ 2.6 แสดงให้เห็นถึงบทบาทของ RTP ในการสื่อสารเสียงและวิดีโอ จะเห็นว่าเมื่อเสียงหรือวิดีโอได้ผ่านขั้นตอนการเข้ารหัสด้วยวิธีการบีบอัดข้อมูลแล้ว ข้อมูลที่ได้ก็จะมีขนาดลดลง หลังจากนั้นข้อมูลส่วนนี้ก็จะถูกนำมาบรรจุลงในแพ็กเก็ต RTP ซึ่งแต่ละแพ็กเก็ตก็จะมีภาระบดด้วยว่าใช้วิธีการบีบอัดข้อมูลแบบใด เป็นแพ็กเก็ตในลำดับที่เท่าไร รวมไปถึงข้อมูลอื่นๆ ที่จะกล่าวถึงในหัวข้อถัดไป ดังนั้นเมื่อข้อมูลนี้เดินทางผ่านเครือข่ายอินเทอร์เน็ตไปถึงฝั่งของผู้รับ ผู้รับก็สามารถใช้ข้อมูลนี้ในการแสดงผลได้ถูกต้อง



รูปที่ 2.6 บทบาทของ RTP ในการสื่อสารเสียงและวิดีโอผ่านเครือข่ายอินเทอร์เน็ต

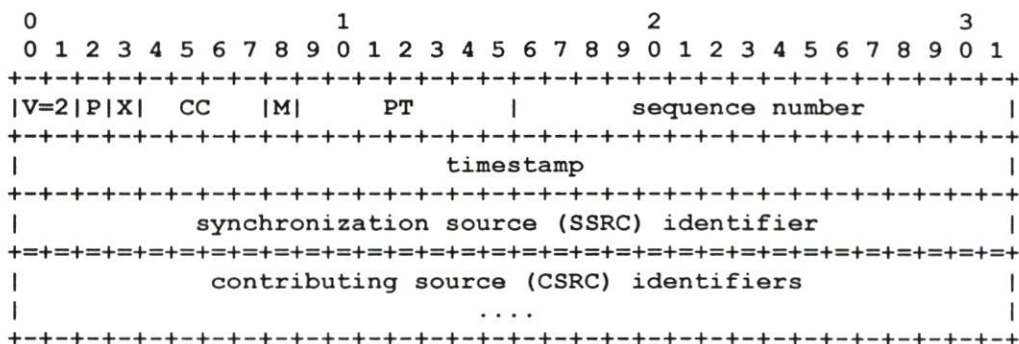
2.2.2 โครงสร้างแพ็กเก็ตของ RTP (RTP Packet Structure)

แพ็กเก็ต RTP แต่ละแพ็กเก็ตนั้นจะมีโครงสร้างดังที่แสดงในรูปที่ 2.7 โดยส่วนแรกของแพ็กเก็ตคือ ส่วนเฮดเดอร์ RTP (RTP Header) เป็นส่วนที่ระบุถึงคุณสมบัติของแต่ละแพ็กเก็ต โดยที่ข้อมูลเสียงหรือวิดีโอที่ผ่านการเข้ารหัสแล้วจะถูกนำไปวางในตำแหน่งของเพย์โหลด (Payload)



รูปที่ 2.7 โครงสร้างแพ็กเก็ตของ RTP

โดยทั่วไปเฮดเดอร์คือ โครงสร้างข้อมูลของแต่ละแพ็กเก็ตที่ถูกรวมเข้ากับข้อมูลที่จะส่ง ซึ่งในแต่ละโปรโตคอลจะมีโครงสร้างของเฮดเดอร์ที่แตกต่างกัน เพื่อกำหนดรูปแบบข้อมูลของโปรโตคอลตัวนั้นๆ สำหรับเฮดเดอร์ของ RTP นั้นมีโครงสร้างดัง รูปที่ 2.8



รูปที่ 2.8 โครงสร้างเฮดเดอร์ของ RTP [8]

จากโครงสร้างเฮดเดอร์ของ RTP ในรูปที่ 2.8 จะเห็นว่า RTP จะมีเฮดเดอร์อย่างน้อย 12 ไบต์ โดยมีรายละเอียดของฟิลด์ต่างๆ ดังนี้

- V (Version) กำหนดเวอร์ชันของโปรโตคอล RTP ที่ใช้ซึ่งปัจจุบันเป็นเวอร์ชัน 2
- P (Padding) เป็นฟิลด์ที่บอกว่าแพ็กเก็ตนั้นได้ถูกเติมด้วยข้อมูลที่เพิ่มให้พอดีกับขนาดของแพ็กเก็ตหรือไม่ และยังใช้ในกระบวนการเข้ารหัสข้อมูล (Encryption) อีกด้วย
- X (Extension) เป็นฟิลด์ที่บอกว่าเฮดเดอร์ RTP มีการขยาย (Extension) หรือไม่
- CC (CSRC Count) บอกจำนวนของ Contribution Source Identifier ในแพ็กเก็ต โดยมี CSRC ได้ตั้งแต่ 0 - 15
- M (Marker) ค่าของบิตนี้จะถูกระบุโดย Profile และ Specification ซึ่งวิธีการใช้งานบิตนี้ได้กำหนดเอาไว้ใน RFC 3551 [9]
- PT (Payload Type) บอกชนิดของข้อมูลภายในแพย์โหลด ซึ่งก็คือชนิดของการบีบอัดข้อมูลเสียงหรือวีดิทัศน์นั่นเอง ซึ่งใน RFC 3551 ได้มีการกำหนดเอาไว้ว่าใช้หมายเลขใดแทนวิธีการบีบอัดข้อมูลแต่ละแบบ
- Sequence Number บอกลำดับของแพ็กเก็ต และเพื่อให้ผู้รับตรวจสอบการสูญหายของแพ็กเก็ต และสามารถใช้ในการเรียงลำดับแพ็กเก็ตใหม่ได้
- Timestamp เป็นค่าที่บอกเวลาที่ใช้ในการซักรหัสของข้อมูลในแพย์โหลด ซึ่งนำไปใช้ในกระบวนการคำนวณจิตเตอร์ และค่า Round Trip Time Delay
- SSRC เป็นตัวเลขประจำ Session นั่นคือแพ็กเก็ตที่มาจากผู้ส่งเดียวกัน และใช้ค่า SSRC เท่ากัน แสดงว่าเป็น Session เดียวกัน
- CSRC ฟิลด์นี้จะถูกใช้เมื่อมีการผสมสื่อ (Media Mixing) ซึ่งเป็นกระบวนการที่ใช้เมื่อมีการประชุมแบบหลายจุด (Multipoint Conference)

2.2.3 RTP Control Protocol (RTCP)

RTCP เป็นโปรโตคอลที่ถูกกำหนดอยู่ใน RFC 3550 เช่นเดียวกับ RTP โปรโตคอลตัวนี้ใช้ในการรายงานเกี่ยวกับสถิติและคุณภาพการบริการ RTCP ใช้ช่องสื่อสารแยกกันกับ RTP โดย RTCP นั้นใช้พอร์ตถัดจาก RTP ไปหนึ่งพอร์ต ช่องสื่อสารของ RTCP นั้นใช้สำหรับการรายงานเกี่ยวกับคุณภาพการบริการ 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 ของข้อมูลเสียงหรือวิดีโอที่ตนได้รับจากแพ็กเก็ต RTP แล้วบรรจุลงในแพ็กเก็ต RR นี้และส่งให้กับผู้ส่ง ผู้ส่งสามารถใช้ข้อมูลในแพ็กเก็ตชนิดนี้ในการประเมินสถานะของเครือข่าย รวมถึงคุณภาพของสื่อที่ผู้รับได้รับ ในการออกแบบวิธีการควบคุมความผิดพลาดแบบปรับตัวของทั้งการสื่อสารเสียงและวิดีโอของวิทยานิพนธ์นี้ก็จะมีการใช้งานแพ็กเก็ตนี้เช่นกัน ในรูปที่ 2.9 เป็นโครงสร้างเฮดเดอร์ของแพ็กเก็ต RR ซึ่งฟิลด์ต่างๆ ที่กำหนดอยู่ในมาตรฐานแล้ว โครงสร้างเฮดเดอร์ของแพ็กเก็ต RTCP ยังอนุญาตให้มีการใส่ข้อมูลเพิ่มเติมลงในแพ็กเก็ตนี้ได้อีก โดยข้อมูลส่วนที่เพิ่มเติมนี้สามารถใส่ได้ในตำแหน่งที่เรียกว่า Profile-Specific Extension

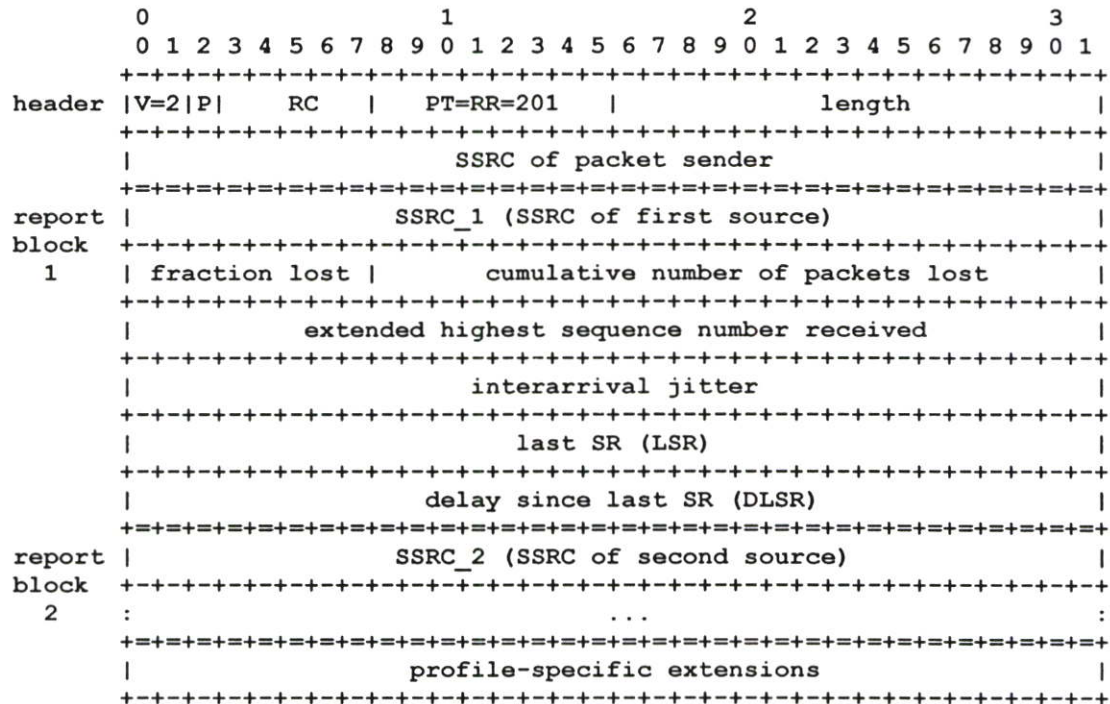
เฮดเดอร์ของแพ็กเก็ต RR จะประกอบด้วย 3 ส่วน โดยส่วนแรกนั้นจะเป็นเฮดเดอร์ที่มีขนาดคงที่ มีขนาดรวมกันได้ 8 ไบต์ ซึ่งแต่ละฟิลด์มีความหมายดังนี้

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

$$\text{Length} = (\text{length of packet in byte} / 4) - 1$$

โดยค่าในฟิลด์ Length นี้จะมีค่าเป็นลบไม่ได้ ดังนั้นจะเห็นว่าความยาวของทั้งแพ็กเก็ตอย่างน้อยที่สุดต้องเป็น 4 ไบต์

- SSRC of packet sender หมายเลขประจำตัวที่ใช้ในการระบุผู้ที่ส่งแพ็กเก็ต RR นี้



รูปที่ 2.9 โครงสร้างเฮดเดอร์ของแพ็กเก็ต Receiver Report (RR)[8]

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

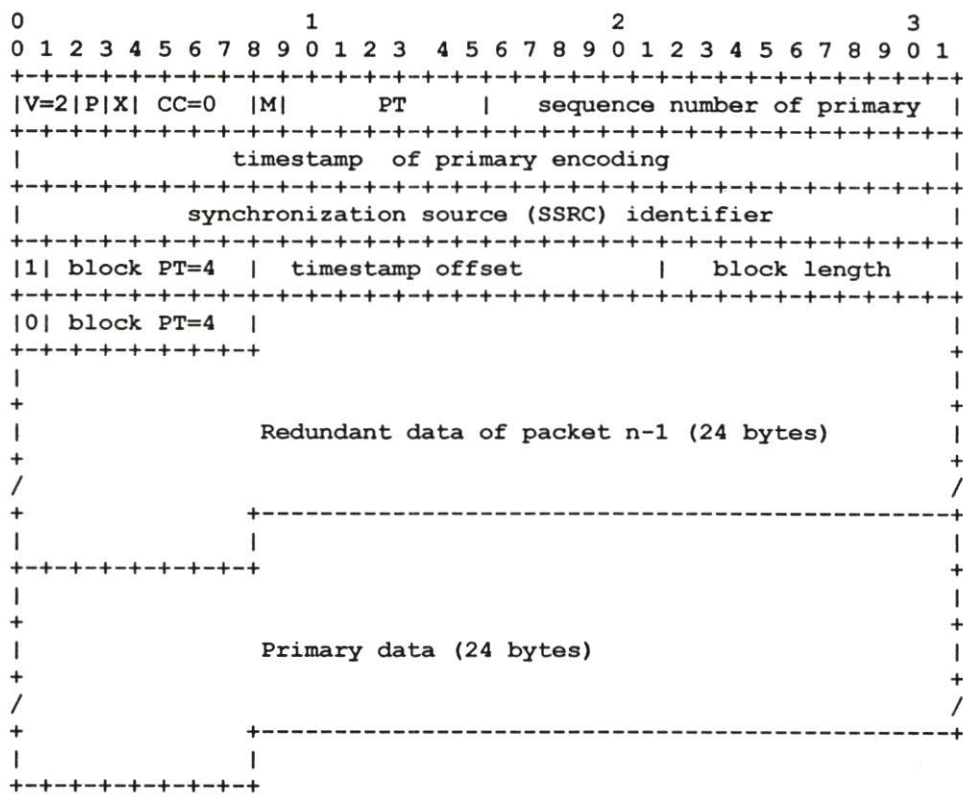
- SSRC ค่า SSRC ที่ Session ที่ต้องการจะรายงานสถิติการรับ
- Fraction Lost อัตราการสูญหายของแพ็กเก็ตค่าในฟิลด์สามารถคำนวณได้ดังนี้

$$\text{Fraction Lost} = (\text{Number of lost packets} / \text{Number of expected packet}) \times 256$$

และหากต้องการแปลงค่า Fraction Lost ให้กลายเป็นเปอร์เซ็นต์การสูญหายของแพ็กเก็ตก็สามารถทำได้ดังนี้

$$\text{Percent Lost} = (\text{Fraction Lost} / 256) \times 100$$

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



รูปที่ 2.11 ตัวอย่างแพ็กเก็ต RTP เมื่อมีการส่งข้อมูลซ้ำของเสียง 1 ชุด

สำหรับเฮดเดอร์ของบล็อกสุดท้ายจะมีแค่ 2 ฟิลด์คือ F และ block PT ทั้งนี้เนื่องจากข้อมูลในบล็อกสุดท้ายนั้นไม่ใช่ข้อมูลซ้ำ แต่เป็นข้อมูลเสียงของแพ็กเก็ตปัจจุบัน รูปที่ 2.11 เป็นตัวอย่างแพ็กเก็ต RTP เมื่อมีการส่งข้อมูลซ้ำของเสียงจำนวน 1 ชุด โดยทั้งข้อมูลหลักและข้อมูลซ้ำใช้การบีบอัดเสียงแบบ G.723.1 โดยทั้งบล็อกที่เป็นข้อมูลหลักและบล็อกที่เป็นข้อมูลซ้ำประกอบด้วยข้อมูลเสียงของ G.723.1 เพียง 1 เฟรม (24 ไบต์)

จากหลักการพื้นฐานของการสื่อสารเสียงที่ได้กล่าวไปแล้วนั้น จะเห็นว่ามิช่องทางสำหรับการควบคุมคุณภาพของเสียงได้ เช่น ในการโปรโตคอล RTP ผู้รับสามารถรายงานสถิติการรับให้ผู้ส่งได้รับทราบโดยใช้แพ็กเก็ต RTCP Receiver Report ในการสื่อสารเสียงก็สามารถที่จะควบคุมความผิดพลาดได้โดยใช้วิธีการส่งข้อมูลซ้ำ ซึ่งวิธีการนี้มีมาตรฐานรองรับแล้ว เนื้อหาเกี่ยวกับวิธีการควบคุมความผิดพลาดของการสื่อสารเสียงจะกล่าวถึงในบทที่ 3 ต่อไป

2.3 เทคนิคการพยากรณ์

การพยากรณ์ คือ การคาดการณ์ว่าอะไรเกิดขึ้นในอนาคต เช่น การพยากรณ์ด้านเศรษฐกิจ การพยากรณ์การเปลี่ยนแปลงเทคโนโลยี การพยากรณ์เป็นสิ่งที่ไม่สามารถควบคุมได้ เป็นการคาดการณ์สิ่งที่จะเกิดโดยนำผลการพยากรณ์มาวางแผนเพื่อจัดการอนาคตให้ดีที่สุด

2.3.1 ประเภทของเทคนิคการพยากรณ์

2.3.1.1 เทคนิคพยากรณ์เชิงคุณภาพ

เป็นเทคนิคการพยากรณ์ที่อาศัยวิจารณญาณ ผลของการพยากรณ์ใช้เทคนิคเดียวกัน อาจได้ค่าไม่เท่ากันขึ้นอยู่กับวิจารณญาณของผู้พยากรณ์เป็นหลัก วิจารณญาณมาจากประสบการณ์พื้นฐานและพื้นฐานของผู้พยากรณ์ ดังนั้นเทคนิคเหมาะสมในกรณี

- เมื่อไม่มีข้อมูล
- ใช้สำหรับการพยากรณ์ในอนาคตที่ไกลมาก เพราะในอนาคตอันไกลความไม่แน่นอนของปัจจัยต่างๆ ที่กระทบต่อสิ่งที่พยากรณ์มาก

2.3.1.2 เทคนิคพยากรณ์เชิงปริมาณ

2.3.1.2.1 เทคนิคอนุกรมเวลา

เป็นเทคนิคที่ใช้ข้อมูลในอดีตเพื่อพยากรณ์ในอนาคต ในการศึกษาเกี่ยวกับอนุกรมเวลาเป็นการศึกษาความเปลี่ยนแปลงของเหตุการณ์หรือพฤติกรรมที่เกิดขึ้นในช่วงเวลาที่ผ่านมามาเพื่อใช้สำหรับคาดการณ์เหตุการณ์ที่น่าจะเป็นไปได้ในอนาคต การทำนายอนุกรมเวลาเป็นการใช้วิธีการทางคณิตศาสตร์และสถิติเพื่อกำหนดรูปแบบ หรือโมเดล ของเหตุการณ์ที่ผ่านมามาโดยตั้งอยู่บนสมมติฐานว่าพฤติกรรมหรือเหตุการณ์ที่จะเกิดขึ้นในอนาคตย่อมจะมีรูปแบบเช่นเดียวกับพฤติกรรมหรือเหตุการณ์ที่เกิดขึ้นมาแล้วในอดีต ส่วนวิธีการสร้างเส้นแนวโน้มจะใช้ผลรวมของกำลังสองของผลต่างระหว่างค่าแนวโน้มกับค่าข้อมูลที่เก็บรวบรวมมาได้มีค่าน้อยที่สุด (least square error) การสร้างเส้นแนวโน้มด้วยวิธีกำลังสองน้อยที่สุด

การพยากรณ์โดยสร้างเส้นแนวโน้มที่ให้ผลรวมกำลังสองของผลต่างระหว่างค่าแนวโน้มกับค่าของข้อมูลที่รวบรวมได้ให้มีค่าน้อยที่สุดจาก

$$\sum_{i=1}^n (Y_i - Y_c)^2 = \text{มีค่าน้อยที่สุด} \quad (2.1)$$

โดยที่ $Y_i =$ ค่าของข้อมูลที่เก็บรวบรวมไว้

$$i = 1, \dots, n$$

$$Y_c = \text{ค่าแนวโน้ม}$$

2.3.1.2.2 เทคนิคความสัมพันธ์ของข้อมูล

เทคนิคการพยากรณ์โดยเทคนิคอนุกรมเวลา ซึ่งเป็นเทคนิคที่ไม่ได้คำนึงถึงตัวแปรที่มีผลกระทบต่อตัวแปรที่ต้องการพยากรณ์ ซึ่งแตกต่างจากแนวคิดของเทคนิคความสัมพันธ์ของข้อมูล (Causal Model) ซึ่งเน้นความสัมพันธ์ของตัวแปรในการพยากรณ์ ถ้ากรณีที่เราวิเคราะห์ความสัมพันธ์ของตัวแปรเพียง 2 ตัว จะเรียกว่าการวิเคราะห์การถดถอย

แบบง่าย (Simple Regression) โดยตัวแปรที่ใช้ในการพยากรณ์มี 2 ประเภทคือ

- ตัวแปรตาม (Dependent variable) คือตัวแปรที่ต้องการพยากรณ์ใช้สัญลักษณ์แทนด้วยอักษร Y
- ตัวแปรอิสระ (Independent variable) คือ ตัวแปรที่มีผลกระทบต่อตัวแปรที่ต้องการพยากรณ์ใช้สัญลักษณ์แทนด้วยอักษร X

2.4 เทคนิควิธีกำลังสองน้อยที่สุด (Least Square Method)

ใช้ในการศึกษาวิธีการพยากรณ์อนุกรมเวลาเมื่อมีข้อมูลผิดปกติเช่นอนุกรมเวลาอัตราการสูญหายของข้อมูลซึ่งมีความแตกต่างกันในลักษณะของระดับค่าเฉลี่ยและความแปรปรวนขึ้นอยู่กับสภาพและสถานะของเครือข่าย ซึ่งในวิทยานิพนธ์นี้ นำ วิธีกำลังสองน้อยที่สุด มาใช้ในการวิเคราะห์รูปแบบข้อมูลซ้ำของอัลกอริทึมทางฝั่งผู้ส่งก่อนที่จะส่งผ่านเครือข่าย เนื่องจากมีความสามารถในการปรับตัวตามการเปลี่ยนแปลง ของอนุกรมเวลาอัตราการสูญหายของข้อมูลได้ดี โดยหลักการ วิธีกำลังสองน้อยที่สุดมีดังนี้

2.4.1 การประมาณค่าโดยวิธีกำลังสองน้อยที่สุด

การสร้างแบบจำลองทางคณิตศาสตร์เพื่ออธิบายปรากฏการณ์ทางวิทยาศาสตร์วิธีหนึ่งก็คือ นำข้อมูลจากการทดลองมาหาความสัมพันธ์ระหว่างตัวแปรที่ปรากฏในการทดลอง แล้วเขียนเป็นฟังก์ชันหรือสมการทางคณิตศาสตร์ เพื่อเป็นตัวแทนใช้ในการอธิบายปรากฏการณ์นั้น การประมาณค่าฟังก์ชันโดยวิธีกำลังสองน้อยที่สุดจะได้ฟังก์ชันที่เป็นตัวแทนที่ดีที่สุดของข้อมูล เพราะได้จากการแก้ค่าความคลาดเคลื่อนของข้อมูลให้เหลือน้อยที่สุด ดังนั้น เมื่อเขียนกราฟของฟังก์ชันเส้นกราฟจะผ่านไปบริเวณจุดต่างๆ ของข้อมูล โดยจะตัดผ่านจุดของข้อมูลบางจุด การประมาณค่าโดยวิธีต่างจากการประมาณค่าในช่วง (interpolation) ดังนี้

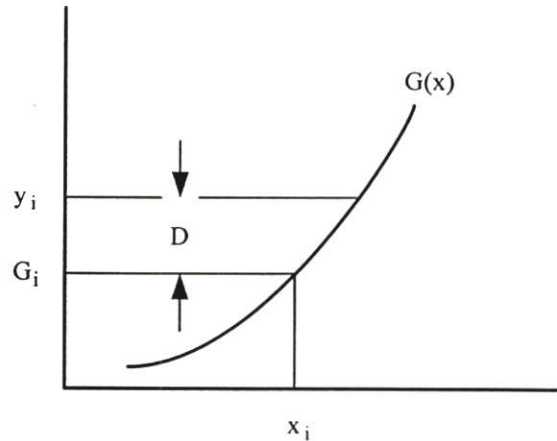
1. ฟังก์ชันที่ได้จากการประมาณค่าในช่วงซึ่งเป็นฟังก์ชันพหุนาม จะต้องผ่านจุดทุกจุดของข้อมูล เมื่อเขียนเส้นกราฟจะเห็นเป็นรอยหยักของเส้น เส้นกราฟจริงๆ โดยธรรมชาติมีลักษณะราบเรียบหรือโค้งเป็นแนวต่อเนื่อง การประมาณค่าโดยวิธีกำลังสองน้อยที่สุดจะได้กราฟผ่านไปบริเวณจุดต่างๆ ของข้อมูล อาจผ่านข้อมูลบางจุดและไม่ผ่านบางจุด เส้นกราฟจะมีลักษณะราบเรียบ (smooth) ใกล้เคียงความเป็นจริงมากกว่า

2. ถ้ามีข้อมูลอยู่ n จุด พหุนามที่ได้จากวิธีประมาณค่าในช่วงจะมีองศาเท่ากับ $n-1$ ถ้า n มีค่ามากๆ พหุนามจะมีองศาตามไปด้วย พหุนามจะมีลักษณะซับซ้อน ส่วนการประมาณค่าด้วยวิธีกำลังสองน้อยที่สุด จะได้สมการเส้นกระแทกรัด ไม่ขึ้นอยู่กับจำนวนของข้อมูล สามารถนำมาหาอนุพันธ์หรืออินทิเกรต เพื่อให้ได้ค่าที่ต้องการได้ง่าย

หลักการของวิธีกำลังสองน้อยที่สุด มีดังนี้ ถ้ามีข้อมูล x, y ทั้งสิ้น n จุด ให้ฟังก์ชันที่ประมาณค่าข้อมูลนี้เป็น $G(x)$ โดยที่ $G(x)$ อยู่ในรูป

$$G(x) = a_1g_1(x) + a_2g_2(x) + \dots + a_mg_m(x) \quad (2.2)$$

โดยที่ $m \leq n$ และ $g_1(x), \dots, g_m(x)$ เป็นฟังก์ชันซึ่งขึ้นอยู่กับค่า x อาจอยู่ในรูปพหุนาม (polynomial) รูปสี่เหลี่ยม หรือเอ็กโพเนนเชียล สมการ (2.2) จะสมบูรณ์ได้ก็ต่อเมื่อทราบค่า a_1, a_2, \dots, a_m โดยหาค่าสัมประสิทธิ์เหล่านี้จากการทำให้ค่าเบี่ยงเบนของข้อมูลกับค่าประมาณที่ได้จากฟังก์ชัน $G(x)$ มีค่าน้อยที่สุด



รูปที่ 2.12 แสดงการหาค่าเบี่ยงเบนของวิธีกำลังสองน้อยที่สุด

จากรูปที่ 2.12 ค่าแตกต่างของข้อมูลจุดที่ i คือ

$y_i - G(x_i)$ เมื่อหาค่าแตกต่างของข้อมูลทุกจุดแล้ว นำค่าแตกต่างเหล่านี้มารวมกัน แล้วยกกำลังสอง เพื่อจัดเครื่องหมายลบ จะได้

$$D = \sum_{i=1}^n [y_i - G(x_i)]^2 \quad (2.3)$$

ค่าสัมประสิทธิ์ a_1, a_2, \dots, a_m จะเป็นตัวแปร เพราะเมื่อค่าเหล่านี้มีค่าต่าง ๆ กัน ฟังก์ชัน $G(x)$ จะเป็นฟังก์ชันที่แตกต่างกันออกไป แต่ต้องการหาค่า a_1, a_2, \dots, a_m ที่มีเงื่อนไขทำให้เกิดค่า D มีค่าน้อยที่สุด จากแคลคูลัสเราจึงได้

$$\left. \begin{aligned} \frac{\partial D}{\partial a_1} &= 0 \\ \frac{\partial D}{\partial a_2} &= 0 \\ \vdots \\ \frac{\partial D}{\partial a_m} &= 0 \end{aligned} \right\} \quad (2.4)$$

จะได้สมการออกมา m ชุด สามารถหา a_1, a_2, \dots, a_m ได้โดยใช้สมการระบบเชิงเส้น

2.4.2 เทคนิคความสัมพันธ์ของข้อมูล

การหาฟังก์ชันเพื่อประมาณค่าชุดข้อมูลโดยวิธีกำลังสองน้อยที่สุดนี้เรียกชื่ออีกอย่างหนึ่งว่า การถดถอย (Regression)

2.4.2.1 ข้อมูลมีความสัมพันธ์แบบเชิงเส้น (Linear Regression)

ถ้าข้อมูลทั้ง n ชุดมีความสัมพันธ์กันในลักษณะเชิงเส้นตรง สมการเส้นตรงที่หาได้จากวิธีนี้เรียกว่า การถดถอยแบบเชิงเส้น ให้ฟังก์ชันที่จะใช้เป็นตัวแทนข้อมูลชุดนี้มีรูปสมการเป็น

$$G(y) = A + Bx \quad (2.5)$$

เมื่อเปรียบเทียบกับสมการ (2.2) นั่นคือ $g_1(x) = 1$, $g_2(x) = x$ เทอมต่อไปเป็นศูนย์ทั้งหมด ผลรวมกำลังสองของค่าเบี่ยงเบนจากสมการ (2.3) คือ

$$D = \sum_{i=1}^n (y_i - A - Bx_i)^2$$

A, B คือค่าคงที่ที่ต้องการหา โดยทำให้ค่า D มีค่าน้อยที่สุด นั่นคือ

$$\frac{\partial D}{\partial A} = -2 \sum_{i=1}^n (y_i - A - Bx_i) = 0$$

$$\frac{\partial D}{\partial B} = -2 \sum_{i=1}^n (y_i - A - Bx_i)(x_i) = 0$$

จัดรูปใหม่จะได้

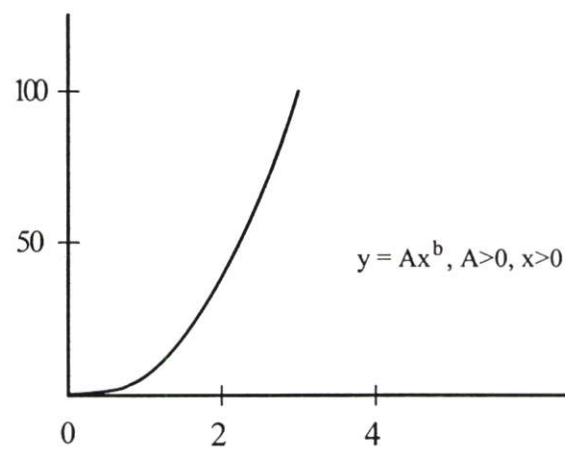
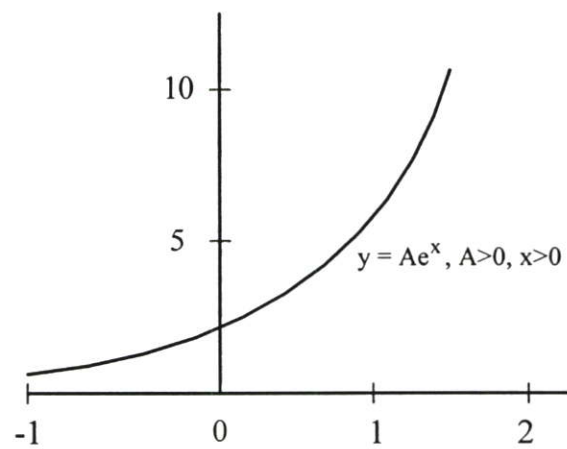
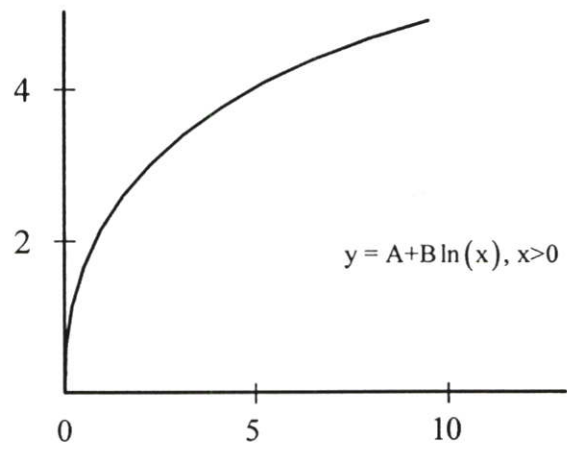
$$nA + B \sum x_i = \sum y_i$$

$$A \sum x_i + B \sum x_i^2 = \sum x_i y_i$$

แก้สมการหาค่า A, B จะได้

$$\left. \begin{aligned} A &= \frac{\sum y_i \sum x_i^2 - \sum x_i \sum x_i y_i}{n \sum x_i^2 - (\sum x_i)^2} \\ B &= \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \end{aligned} \right\} \quad (2.6)$$

เราสามารถประยุกต์วิธีถดถอยแบบเชิงเส้น ไปใช้กับข้อมูลที่มีความสัมพันธ์แบบไม่เป็นเชิงเส้นได้ ข้อมูลที่มีความสัมพันธ์แบบไม่เป็นเชิงเส้นอาจอยู่ในรูปล็อกการริซึม เอ็กโพเนนเชียล หรือในรูปยกกำลังสองขึ้นไป ดังรูป 2.13



รูปที่ 2.13 ข้อมูลที่มีลักษณะไม่เป็นเชิงเส้นแบบต่างๆ

2.4.2.2 ข้อมูลมีความสัมพันธ์แบบพหุนาม (Polynomial Regression)

ข้อมูล n ชุด มีความสัมพันธ์กันแบบพหุนามอันดับ m เขียนเป็นสมการได้ดังนี้

$$G(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m$$

ผลรวมกำลังสองของความเบี่ยงเบนของข้อมูลกับ $G(x)$ คือ

$$D = \sum (y_i - G(x))^2$$

หาค่า $a_0, a_1, a_2, \dots, a_m$ โดยใช้เงื่อนไข D ต้องมีค่าน้อยที่สุด

$$\frac{\partial D}{\partial a_i} = 0, \quad i = 0, 1, 2, \dots, m$$

จะได้เป็นระบบสมการเชิงเส้น $m+1$ สมการ ได้คำตอบของสมการ

ให้ $\sum_{i=1}^n x_i = \sum x$ ละเว้นพิสัยของการหาผลรวมเพื่อให้ดูง่าย

$$\begin{vmatrix} n & \sum x & \sum x^2 & \dots & \sum x^m \\ \sum x & \sum x^2 & \sum x^3 & \dots & \sum x^{m+1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \sum x^m & \sum x^{m+1} & \sum x^{m+2} & \dots & \sum x^{2m} \end{vmatrix} \begin{vmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{vmatrix} = \begin{vmatrix} \sum y \\ \sum xy \\ \vdots \\ \sum x^m y \end{vmatrix} \quad (2.7)$$

จะเห็นว่าสมการ (2.6) เป็นส่วนหนึ่งของสมการ (2.7) เมื่อ $m=1$ นั่นเอง

ถ้าฟังก์ชันไม่เป็นพหุนาม มีเทอมอื่นๆ เข้ามาปนอยู่ด้วย เช่น $y = A + Bx + C/x^2$ เป็นต้น ฟังก์ชันเหล่านี้สามารถเขียนให้อยู่ในรูปทั่วไปได้เป็น

$$G(x) = a_0 + a_1f(x) + a_2g(x) \quad (2.8)$$

เมื่อ $f(x) = x$

$$g(x) = \frac{1}{x^2}$$

เราสามารถใช้หัวข้อ 2.4.2.3 ข้อมูลมีความสัมพันธ์แบบหลายเชิงช่วยแก้ปัญหาได้

2.4.2.3 ข้อมูลมีความสัมพันธ์แบบหลายเชิง (Multiple Regression)

ข้อมูลที่ได้จากการทดลองทางวิทยาศาสตร์อาจขึ้นอยู่กับตัวแปรหลายๆตัวแปร เช่น กระแสไฟฟ้าในวงจร RC ถ้าตัวต้านทานมีการเปลี่ยนค่าได้ กระแสไฟฟ้าที่วัดได้ขึ้นอยู่กับค่าความต้านทานและเวลาที่เปลี่ยนไป ความดันของอากาศในกระบอกสูบขึ้นอยู่กับอุณหภูมิของอากาศและปริมาตรของกระบอกสูบที่เปลี่ยนไปเมื่อลูกสูบเคลื่อนที่ อันที่จริงแล้วความดันของอากาศยังคง

ขึ้นอยู่กับความสูงจากระดับน้ำทะเลและไอน้ำในอากาศ จะเห็นว่าความดันอากาศขึ้นอยู่กับตัวแปรหลายๆตัวแปร

เราสามารถเขียนตัวแปรตามที่สูงขึ้นกับตัวแปรอิสระหลายๆตัว ให้อยู่ในรูปทั่วไปได้เป็น

$$y = f(x_1, x_2, x_3, \dots, x_j) \quad (2.9)$$

เมื่อ j คือจำนวนตัวแปรอิสระทั้งหมด

ถ้าตัวแปรตามมีการเปลี่ยนแปลงตามตัวแปร x แต่ละตัวในลักษณะเชิงเส้น เราสามารถเขียนฟังก์ชันเชิงเส้นตรงที่ใช้แทนข้อมูลชุดนี้ ดังนี้

$$G(x_1, \dots, x_j) = a_0 + a_1x + a_2x + \dots + a_jx_j \quad (2.10)$$

ผลรวมกำลังสองของความเบี่ยงเบนของข้อมูล n ชุด กับ $G(x_j)$ คือ

$$D = \sum_{i=1}^n (y_i - G(x_1, x_2, \dots, x_j))^2 \quad (2.11)$$

หาค่า a_0, a_1, \dots, a_j โดยที่ความคลาดเคลื่อนที่เกิดขึ้นต้องมีค่าต่ำสุด นั่นคือ

$$\left. \begin{aligned} \frac{\partial D}{\partial a_0} &= 0 \\ \frac{\partial D}{\partial a_1} &= 0 \\ \vdots \\ \frac{\partial D}{\partial a_j} &= 0 \end{aligned} \right\} \quad (2.12)$$

สมการ (2.12) จะประกอบด้วยสมการย่อย $(j+1)$ สมการ เขียนให้อยู่ในรูปเมทริกซ์ดังนี้

$$\begin{vmatrix} n & \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{2i} & \dots & \sum_{i=1}^n x_{ji} \\ \sum_{i=1}^n x_{1i} & \sum_{i=1}^n x_{1i}^2 & \sum_{i=1}^n x_{1i}x_{2i} & \dots & \sum_{i=1}^n x_{1i}x_{ji} \\ \sum_{i=1}^n x_{2i} & \sum_{i=1}^n x_{1i}x_{2i} & \sum_{i=1}^n x_{2i}^2 & \dots & \sum_{i=1}^n x_{2i}x_{ji} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \sum_{i=1}^n x_{ji} & \sum_{i=1}^n x_{1i}x_{ji} & \sum_{i=1}^n x_{2i}x_{ji} & \dots & \sum_{i=1}^n x_{ji}^2 \end{vmatrix} \begin{vmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_j \end{vmatrix} = \begin{vmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_{1i}y_i \\ \sum_{i=1}^n x_{2i}y_i \\ \vdots \\ \sum_{i=1}^n x_{ji}y_i \end{vmatrix} \quad (2.13)$$

2.4.3 การตรวจสอบความเหมาะสมของฟังก์ชัน

ปัญหาที่สำคัญประการหนึ่งคือจะรู้ได้อย่างไรว่าข้อมูลที่ได้มานั้นสอดคล้องกับสมการเส้นตรงหรือสมการพหุนาม หรือสมการเอ็กโพเนนเชียล วิธีตรวจสอบที่ง่ายคือการพล็อตกราฟสมการนั้นเปรียบเทียบกับกราฟที่ได้จากข้อมูล แต่วิธีนี้ก็จะทำได้ลำบากถ้ามีตัวแปรในสมการมากกว่า 2 ตัวขึ้นไป ด้วยเหตุนี้จึงมีการคำนวณทางสถิติ เช่น สัมประสิทธิ์ของการกำหนด (Coefficient of determination) จะใช้วัดความสัมพันธ์กันหรือสหสัมพันธ์ระหว่างตัวแปรอิสระทั้งหลายกับตัวแปรตาม สามารถบอกได้ว่าสอดคล้องกับสมการที่ได้หรือไม่

หลักการเบื้องต้นของสัมประสิทธิ์การกำหนดสามารถอธิบายได้ดังนี้ ข้อมูลที่ได้จากการทดลอง (x_i, y_i) และค่าที่ได้จากการคำนวณของ $G(x_i)$ ความแตกต่างที่ได้คือ $y_i - G(x_i)$ เมื่อหาความแตกต่างของทุกๆจุด ยกกำลังสองทุกค่าเพื่อจัดค่าที่เป็นลบ ผลรวมกำลังสองของความแตกต่างของ y_i (จากข้อมูล) กับ $G(x_i)$ (ที่ได้จากการคำนวณ) เรียกว่า ผลรวมกำลังสองของเศษเหลือ (The sum square of the residuals)

$$SSE = \sum_{i=1}^n (y_i - G(x_i))^2 \quad (2.14)$$

ถ้านำค่า y_i จากข้อมูลที่ได้ลบออกจากค่าเฉลี่ยของ $y(\bar{y})$ ผลรวมกำลังสองของความแตกต่าง y_i และ (\bar{y}) ทุกค่าเรียกว่า ผลรวมกำลังสองของความถดถอย (The sum of the square of the regression)

$$SSR = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (2.15)$$

สัมประสิทธิ์การกำหนดจะบอกว่าข้อมูลกับฟังก์ชันมีความเหมาะสมสอดคล้องกันดีหรือไม่ หาได้จาก

$$R^2 = 1 - \frac{SSE}{SSR} \quad (2.16)$$

ในกรณีที่ระบบสมการเชิงเส้น สมการ R^2 สามารถเขียนได้เป็น

$$R^2 = \frac{\left(n \sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n y_i \right)^2 \right)^2}{\left(\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 \right) \left(n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2 \right)} \quad (2.17)$$

ค่า R^2 จะมีค่าอยู่ระหว่าง 0 ถึง 1 ถ้าค่า y_i และ $y(x_i)$ มีความแตกต่างกันน้อยหรือไม่มี ความแตกต่าง จะได้ค่า R^2 ใกล้เคียง 1 เส้นกราฟที่ได้จะสอดคล้องเหมาะสมกับข้อมูล ควรจะได้ค่า R^2 มากกว่า 0.9

ค่าทางสถิติอีกค่าหนึ่งคือค่าความคลาดเคลื่อนมาตรฐานของการประมาณค่า y (Standard error of the estimate) จะเป็นการหาค่าความคลาดเคลื่อนของค่า y ที่ได้จากการคำนวณ และค่า y ของข้อมูล ซึ่งหาได้จากสูตรต่อไปนี้

$$S_x = \sqrt{\frac{\sum (y_i - G(x_i))^2}{(n-m)}} \quad (2.18)$$

เมื่อ n คือจำนวนข้อมูล และ m คือจำนวนสัมประสิทธิ์ในสมการถดถอย เรียก $(n-m)$ ว่าเป็น องศาของความเป็นอิสระ (number of degree of freedom) ถ้าฟังก์ชันที่ใช้ในการประมาณเป็น สมการเส้นตรง องศาแห่งความเป็นอิสระคือ $n-2$ และ ค่า S_x ยิ่งน้อย และ R^2 ใกล้เคียง 1 มาเท่าใด แสดงว่าฟังก์ชันที่ได้สอดคล้องและเหมาะสมกับข้อมูลที่มีอยู่มากขึ้นเพียงนั้น

ซึ่งในวิทยานิพนธ์นี้ได้ใช้เทคนิควิธีกำลังสองน้อยที่สุด มาใช้ในการประมาณค่า พารามิเตอร์ (parameter) สำคัญ ที่ใช้ในการกำหนดรูปแบบ และปริมาณข้อมูลซ้ำ (Combination) เพื่อควบคุมความผิดพลาดของการสื่อสารเสียง ที่ใช้ในอัลกอริทึมแบบปรับตัว (adaptive error control algorithm) ให้สอดคล้องกับสถานะของเครือข่ายในขณะนั้น และจะได้กล่าวถึงในบทที่ 3

บทที่ 3

การปรับปรุงและการทำแบบจำลองของอัลกอริทึมแบบปรับตัว

เนื่องจากการสื่อสารพหุสื่อบนเครือข่ายอินเทอร์เน็ตนั้นไม่สามารถที่หลีกเลี่ยงปัญหาการสูญหายของแพ็กเก็ตได้ ซึ่งการสูญหายของแพ็กเก็ตเกิดข้อมูลเสียงเป็นสาเหตุสำคัญที่ทำให้คุณภาพเสียงลดลง ดังนั้นจึงจำเป็นที่จะต้องมียกไกที่จะช่วยลดผลกระทบจากการสูญหายแพ็กเก็ต ซึ่งในที่นี้จะเรียกยกไกที่ทำหน้าที่นี้ว่า การควบคุมความผิดพลาด (Error Control) โดยเนื้อหาของบทนี้เริ่มต้นด้วยการกล่าวถึง การควบคุมความผิดพลาดแบบปรับตัว (Adaptive Error Control) ในหัวข้อที่ 3.1 โดยจะกล่าวถึงเทคนิคการควบคุมความผิดพลาด Forward Error Correction (FEC) และประเภทของ FEC หัวข้อที่ 3.2 จะกล่าวถึง อัลกอริทึมที่ใช้ในการศึกษา ส่วนหัวข้อที่ 3.3 และ 3.4 จะเป็นเนื้อหาเกี่ยวกับ การใช้เทคนิควิธีกำลังสองน้อยที่สุด (Least Square Method) มาประยุกต์ใช้ในอัลกอริทึมแบบปรับตัว CNR (Centre Network Research) และ การตรวจสอบความถูกต้องของแบบจำลองที่ใช้ในการศึกษาตามลำดับ

3.1 การควบคุมความผิดพลาดแบบปรับตัว

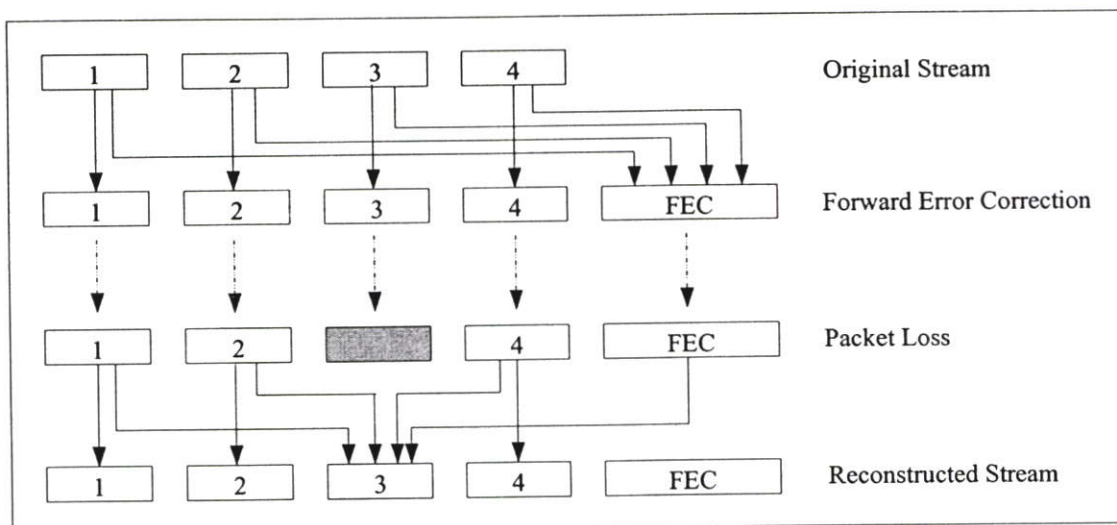
ในการสื่อสารเสียงบนเครือข่ายที่ไม่มีการรับประกันคุณภาพการบริการอย่างเช่นเครือข่ายอินเทอร์เน็ตนั้น ปัญหาอย่างหนึ่งที่ส่งผลกระทบต่อคุณภาพของเสียงก็คือการสูญหายของแพ็กเก็ตถึงแม้ว่าในการสื่อสารเสียงนั้นไม่จำเป็นจะต้องได้รับข้อมูลครบทั้งหมดก็ตาม แต่การที่มีแพ็กเก็ตสูญหายมากเกินไปก็ทำให้คุณภาพเสียงต่ำเกินกว่าที่ยอมรับได้ ดังนั้นจึงจำเป็นต้องมียกไกในการลดผลกระทบจากการสูญหายของแพ็กเก็ต ซึ่งยกไกดังกล่าวก็คือ การควบคุมความผิดพลาด ซึ่ง Forward Error Correction (FEC) ก็เป็นวิธีการควบคุมความผิดพลาดวิธีการหนึ่งที่สามารถลดผลกระทบจากการสูญหายของแพ็กเก็ตได้โดยไม่เพิ่มค่าเวลาหน่วงมากนัก[11] ซึ่งคุณสมบัตินี้เหมาะสมสำหรับการสื่อสารเสียงที่ต้องการความเป็นเวลาจริง

3.1.1 ประเภทคือ FEC

FEC สามารถแบ่งออกได้เป็น 2 ประเภท คือ

1. ประเภทที่ไม่ขึ้นกับสื่อ (Media Independent FEC)

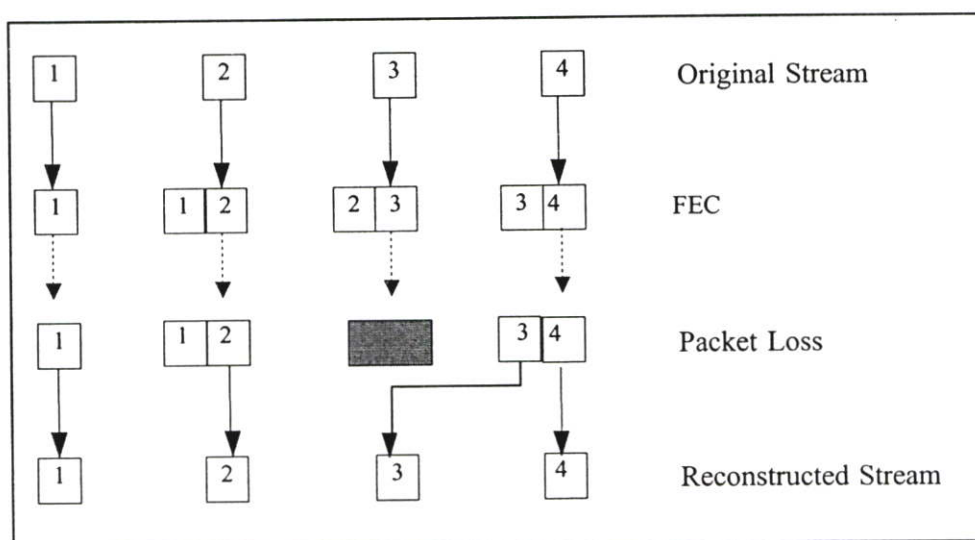
มีหลักการคือ เมื่อมีการส่งแพ็กเก็ตได้หนึ่งชุด จะมีการส่งแพ็กเก็ตพิเศษเพิ่มเข้าไป ดังรูปที่ 3.1 โดยแพ็กเก็ตพิเศษนี้ได้จากการกระทำทางคณิตศาสตร์ของข้อมูลจากทุกแพ็กเก็ตที่อยู่ในชุดเดียวกัน ดังนั้นเมื่อมีแพ็กเก็ตใดสูญหายไป ข้อมูลในแพ็กเก็ตที่สูญหายสามารถกู้คืนได้โดยใช้แพ็กเก็ตพิเศษนี้ แต่วิธีการนี้ใช้ไม่ได้ผลถ้ามีแพ็กเก็ตในชุดเดียวกันสูญหายมากกว่า 1 แพ็กเก็ต [12]



รูปที่ 3.1 Media Independent FEC

2. ประเภทที่ขึ้นกับสื่อ (Media Specific FEC)

มีหลักการคือ ในแต่ละแพ็กเก็ตที่ผู้ส่งจะส่งไปให้ผู้รับ จะมีการบรรจุข้อมูลซ้ำ (Redundancy) ของแพ็กเก็ตที่อยู่ก่อนหน้า แต่อาจใช้การบีบอัดด้วยอัตราบิตที่ต่ำกว่าเพื่อไม่ให้ปริมาณการใช้แบนด์วิดท์เพิ่มขึ้นมากเกินไป จากรูปที่ 3.2 จะเห็นว่าเมื่อแพ็กเก็ตที่ 3 สูญหาย ข้อมูลเสียงที่อยู่ในแพ็กเก็ตนี้สามารถกู้คืนได้จากข้อมูลซ้ำที่อยู่ในแพ็กเก็ตที่ 4 ซึ่ง FEC ประเภทที่สองนี้เป็นวิธีการที่ได้รับความนิยมมากกว่า FEC ประเภทแรก และมีมาตรฐานที่กำหนดรูปแบบของการบรรจุข้อมูลซ้ำของเสียงลงในแพ็กเก็ต RTP อีกด้วย ซึ่งมาตรฐานดังกล่าวก็คือ RFC2198[10] แต่มาตรฐานนี้ก็กำหนดเพียงแค่รูปแบบของแพ็กเก็ตเท่านั้น แต่ไม่ได้กำหนดว่าต้องใช้ปริมาณข้อมูลซ้ำเท่าใด และข้อมูลซ้ำแต่ละชุดใช้วิธีการบีบอัดเสียงแบบใด



รูปที่ 3.2 Media Specific FEC

เนื่องจากการควบคุมความผิดพลาดโดยใช้ FEC มีการส่งข้อมูลซ้ำของเสียงซึ่งทำให้ปริมาณการใช้แบนด์วิดท์ของสื่อสารเสียงเพิ่มขึ้น ดังนั้นจึงควรกำหนดปริมาณข้อมูลซ้ำให้เหมาะสมกับสภาพของเครือข่าย ในกรณีที่มีการสูญหายของแพ็กเก็ตเพียงเล็กน้อย หากกำหนดปริมาณข้อมูลซ้ำมากเกินไปจะเป็นการสิ้นเปลืองแบนด์วิดท์โดยไม่จำเป็น ดังนั้นในการสื่อสารเสียงจึงควรมีอัลกอริทึมในการกำหนดรูปแบบและปริมาณของข้อมูลซ้ำที่สามารถปรับตัวได้ตามปริมาณการสูญหายของแพ็กเก็ต ซึ่งอัลกอริทึมที่ได้มีการเสนอได้แก่ อัลกอริทึม BoLot[13], อัลกอริทึม RCCS[14] และอัลกอริทึม CNR[15] โดยทุกบทความนั้นมีส่วนที่เหมือนกันก็คือ อัลกอริทึมที่เสนอนั้นล้วนมีพื้นฐานจาก FEC และจะมีการกำหนดรูปแบบการบรรจุข้อมูลซ้ำลงในแพ็กเก็ตหรือที่เรียกว่า Combination เอาไว้ก่อน และจัดลำดับของแต่ละ Combination เอาไว้ โดย Combination ในลำดับหลังสามารถลดผลกระทบจากการสูญหายของแพ็กเก็ตได้ดีกว่า Combination ที่อยู่ลำดับแรกๆ แต่ต้องใช้ปริมาณข้อมูลซ้ำที่มากขึ้นด้วย โดยพารามิเตอร์ที่ใช้ในการตัดสินใจว่าควรจะใช้ Combination ใดก็คือปริมาณการสูญหายของข้อมูล สำหรับส่วนที่แตกต่างกันของแต่ละบทความก็คืออัลกอริทึมที่ใช้ในการเลือก Combination เมื่อใดควรที่จะเพิ่ม Combination และเมื่อใดควรที่จะลด เนื่องจากว่าอัลกอริทึมของการปรับตัวที่จัดอยู่ในประเภนี้มีเป็นจำนวนมาก และแต่ละอัลกอริทึมก็ใช้ได้ผลในเงื่อนไขที่แตกต่างกัน วิทยานิพนธ์นี้จึงได้การศึกษาและปรับปรุงจากอัลกอริทึมที่มีอยู่แล้ว โดยเพิ่มเทคนิควิธีกำลังสองน้อยที่สุด (Least Square Method) เพื่อให้กำหนดพารามิเตอร์ที่เหมาะสมกับการสื่อสารเสียงในกรณีที่ใช้การบีบอัดแบบ G.723.1

3.1.2 อัตราการสูญหายของข้อมูลเสียงเมื่อมีการใช้ FEC

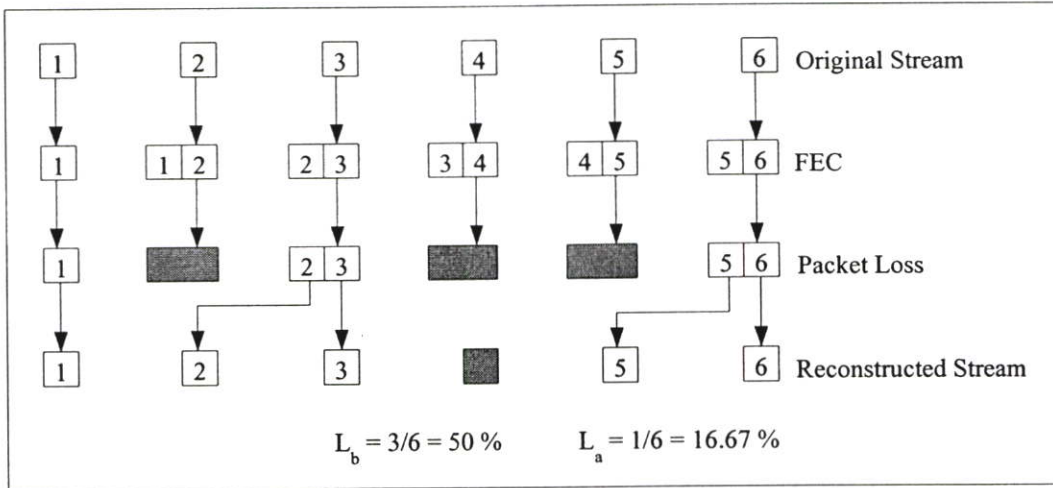
ก่อนจะกล่าวถึงอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวที่ใช้ในการศึกษา ในหัวข้อนี้จะอธิบายเกี่ยวกับค่าอัตราการสูญหายของข้อมูลเมื่อมีการใช้เทคนิค FEC ทั้งนี้เพื่อให้สามารถเข้าใจถึงรายละเอียดของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวที่จะกล่าวถึงในหัวข้อถัดไปได้ง่ายขึ้น ซึ่งเมื่อมีการใช้เทคนิค FEC แล้ว ค่าอัตราการสูญหายของข้อมูลที่จะต้องพิจารณามีอยู่ 2 ค่าคือ

L_b คือ ค่าอัตราการสูญหายก่อนการจัดเรียง (Loss Rate before Reconstruction)

L_a คือ ค่าอัตราการสูญหายหลังการจัดเรียง (Loss Rate after Reconstruction)

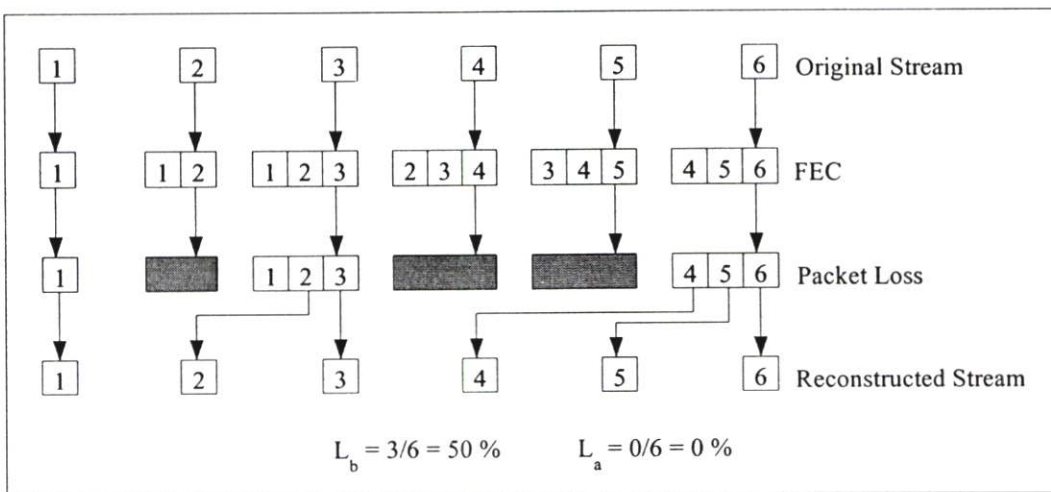
ซึ่งค่า L_b นั้นหมายถึงค่าอัตราการสูญหายของข้อมูลเสียงโดยที่ยังไม่ได้นำข้อมูลซ้ำมาจัดเรียงใหม่ หรืออาจจะกล่าวได้ว่าค่า L_b คืออัตราการสูญหายของแพ็กเก็ตนั่นเอง จากรูปที่ จะเห็นว่ามีการส่งแพ็กเก็ตเสียงทั้งหมด 6 แพ็กเก็ต แต่เกิดการสูญหาย 3 แพ็กเก็ต ดังนั้นค่า L_b จึงมีค่าเท่ากับ $3/6$ หรือ 50 % นั่นเอง ส่วนค่า L_a จะคำนวณจากจำนวนข้อมูลเสียงที่สูญหายหลังจากที่ได้มีการจัดเรียงข้อมูลแล้ว ค่า L_a จึงมักจะมีค่าน้อยกว่า L_b เนื่องจากอาจมีการกู้คืนข้อมูลในแพ็กเก็ตที่

สูญหายบางส่วนได้จากข้อมูลซ้ำ จากรูปที่ 3.3 จะเห็นว่าเมื่อมีการจัดเรียงข้อมูลแล้วมีข้อมูลเสียงสูญหายไปเพียง 1 เฟรมเท่านั้นคือเฟรมที่ 4 ดังนั้นค่า L_b จึงมีค่าเท่ากับ $1/6$ หรือ 16.67 % ซึ่งค่า L_a นี้เองจะเป็นตัวบอกถึงปริมาณข้อมูลเสียงที่สูญหาย ถ้า L_a มีค่ามากจะทำให้เสียงที่ได้รับมีคุณภาพต่ำ ส่วนค่า L_b เป็นค่าที่บอกถึงปริมาณการสูญหายของแพ็คเกจที่แน่นอน



รูปที่ 3.3 การกู้คืนโดยใช้ข้อมูลซ้ำสามารถทำให้ค่า L_a น้อยกว่า L_b

การเพิ่มจำนวนชุดของข้อมูลซ้ำสามารถลดค่า L_a ได้ถึงแม้ว่าค่า L_b จะมีค่าเท่าเดิม จากรูปที่ 3.4 จะเห็นว่าเมื่อเพิ่มจำนวนข้อมูลซ้ำเป็น 3 ชุด สามารถกู้คืนข้อมูลเสียงที่สูญหายได้หมด ทำให้ค่า L_a เท่ากับ 0 % ดังนั้นในกรณีที่เครือข่ายมีความมึนคั่งสูงและมีปริมาณแพ็คเกจที่สูญหายมากจึงควรจะมีเพิ่มจำนวนชุดข้อมูลซ้ำ เพื่อให้สามารถกู้ข้อมูลเสียงในแพ็คเกจที่สูญหายได้มากขึ้น และสามารถรักษาระดับของอัตราการสูญหายของข้อมูลเสียงไม่ให้สูงเกินไปได้



รูปที่ 3.4 การเพิ่มจำนวนข้อมูลซ้ำสามารถช่วยลดค่า L_a ได้

3.2 อัลกอริทึมแบบปรับตัวที่ใช้ในการศึกษา

3.2.1 อัลกอริทึม Bolot

อัลกอริทึมนี้ได้รับการเสนอโดย Bolot และได้รับการตีพิมพ์ลงใน [13] ซึ่งในบทความนี้ได้มีการกำหนดรูปแบบของการจัดวางข้อมูลซ้ำ (Combination) เอาไว้หลายแบบ และได้ทำการทดลองเพื่อหาค่า Reward ของแต่ละ Combination โดยค่า Reward (R_c) คืออัตราส่วนระหว่างค่า L_b และค่า L_a

$$R_c = L_b/L_a \quad (3.1)$$

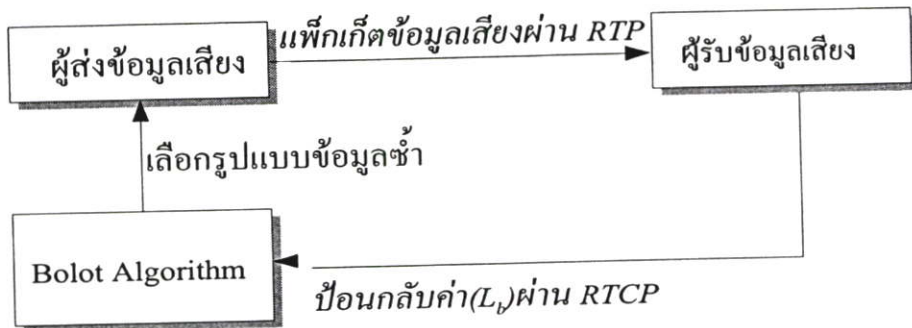
ตารางที่ 3.1 Combination ที่ใช้อัลกอริทึม Bolot

| 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 ที่ปรากฏใน ตารางที่ 3.1 เป็นค่าที่ได้จากการทดลอง โดยใช้เครือข่ายระหว่างสถาบัน INRIA ในประเทศฝรั่งเศสกับ University College London (UCL) ในประเทศอังกฤษ สำหรับสัญญาณ (PCM) หมายถึง ทุกแพ็กเก็ตใช้การบีบอัดแบบ PCM โดยไม่มีการบรรจุข้อมูลซ้ำ ส่วนสัญญาณ (PCM, ADM4(1)) หมายถึง ในแต่ละแพ็กเก็ตลำดับที่ N ส่วนที่เป็นข้อมูลหลักใช้การบีบอัดแบบ PCM แต่ส่วนที่เป็นข้อมูลซ้ำใช้การบีบอัดแบบ ADM4 และเป็นข้อมูลซ้ำของแพ็กเก็ตที่ N-1 (ตัวเลข 1 ที่อยู่ในวงเล็บ) และสัญญาณของ Combination แบบอื่นๆ สามารถแปลความหมายได้ในทำนองเดียวกัน ซึ่งค่า Reward นี้เป็นค่าที่สามารถบอกได้ว่าแต่ละ

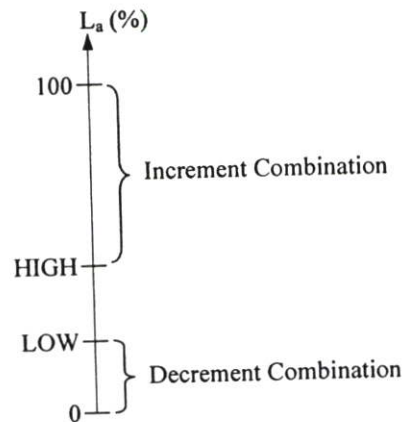
Combination สามารถกู้ข้อมูลเสียงในแพ็กเก็ตที่สูญหายได้ดีเพียงใด และจะเห็นว่า Reward จะเพิ่มขึ้นเมื่อมีการเพิ่มจำนวนชุดของข้อมูลซ้ำ ยกเว้นในกรณีของ Combination ที่ 4 และ 5 ซึ่งมีค่า Reward เท่ากัน และในกรณีของ Combination ที่มีการใช้จำนวนข้อมูลซ้ำเท่ากับ Combination ที่ใช้ข้อมูลซ้ำจากแพ็กเก็ตที่อยู่ห่างจากแพ็กเก็ตปัจจุบันมากกว่าจะมีค่า Reward สูงกว่า ตัวอย่างเช่น Combination ที่ 4 มีค่า Reward สูงกว่า Combination ที่ 3 แต่อย่างไรก็ตามค่า Reward ที่ได้แสดงในตารางที่ 3.1 นี้เป็นค่าที่ได้จากการทดลองเท่านั้น ซึ่งอาจจะไม่เป็นจริงเสมอไปในทุกสภาพแวดล้อม

จากแผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม Bolot ในรูปที่ 3.5 จะเห็นว่าผู้รับจะมีการรายงานเฉพาะค่า L_r ให้ผู้ส่งได้ทราบเท่านั้นโดยจะระบุค่าของ L_r ลงฟิลด์ Fraction Lost ของแพ็กเก็ต RTCP ชนิด Receiver Report ซึ่งปกติฟิลด์ Fraction Lost นี้จะใช้ในการระบุค่าอัตราการสูญหายของแพ็กเก็ต และเนื่องจากค่า L_r มีค่าเท่ากับอัตราการสูญหายของแพ็กเก็ต จึงสามารถบรรจุลงในฟิลด์นี้ได้เลย แต่ผู้รับจะไม่รายงานค่า L_s ทั้งนี้เนื่องจากว่าอัลกอริทึม Bolot ไม่ได้ใช้ค่าจริงของ L_s แต่จะคำนวณค่า L_s โดยใช้ค่า L_r หาค่าด้วยค่า Reward ของ Combination ที่กำลังใช้งาน และนำค่า L_s ไปใช้ในการตัดสินใจว่าควรเลือกที่จะเพิ่มหรือลด Combination



รูปที่ 3.5 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม Bolot

จากรูปที่ 3.6 จะเห็นว่าอัลกอริทึม Bolot มีค่าเทรสโฮลด์ที่ใช้ในการพิจารณาค่า L_s อยู่ 2 ค่า คือ HIGH และ LOW ถ้าหากค่า L_s สูงเกินกว่าค่าเทรสโฮลด์ HIGH ให้เพิ่ม Combination โดยการเพิ่ม Combination หมายถึงการเลือกใช้ Combination ในตารางที่ 3.1 ในขั้นที่สูงขึ้น เช่น ถ้าหากเดิมใช้ Combination หมายเลข 1 เมื่อใดที่ค่า L_s สูงกว่าเทรสโฮลด์ HIGH ก็จะได้เลื่อนขึ้นไปใช้ Combination หมายเลข 2 เป็นต้น แต่ถ้าค่า L_s ต่ำกว่าค่าเทรสโฮลด์ LOW ให้ลด Combination โดยใน [13] ได้กำหนดให้ค่าเทรสโฮลด์ HIGH และ LOW มีค่าเท่ากันคือ 3% เพื่อควบคุมให้ค่า L_s อยู่ที่ 3% แต่การกำหนดค่าเทรสโฮลด์ทั้งสองให้มีค่าเท่ากันจะทำให้มีการเปลี่ยน Combination บ่อยครั้ง เพราะไม่มีช่วงของค่า L_s ช่วงใดเลยที่อัลกอริทึมนี้ไม่ต้องมีการเปลี่ยนแปลง Combination



รูปที่ 3.6 การเพิ่มและลด Combination ในอัลกอริทึม Bolot

หลักการการทำงานของอัลกอริทึม Bolot สามารถสรุปเป็น Pseudo Code ได้ดังรูปที่ 3.7 โดยขั้นตอนที่ 1 เป็นคำนวณค่า L_b จากค่าที่อ่านได้จากแพ็กเก็ต Receiver Report โดยค่าของ L_b จะระบุอยู่ในฟิลด์ Fraction Lost ซึ่งปกติแล้วการระบุค่าอัตราการสูญหายในแพ็กเก็ตในฟิลด์ Fraction Lost จะไม่ระบุเป็นค่าทศนิยมหรือค่าที่เป็นเปอร์เซ็นต์โดยตรง แต่จะระบุเป็นอัตราส่วนเมื่อเทียบกับ 256[8] เนื่องจากฟิลด์นี้มีขนาด 1 ไบต์ ตัวอย่างเช่น ถ้าค่าในฟิลด์ Fraction Loss เท่ากับ 128 หมายถึงอัตราการสูญหายของแพ็กเก็ตเท่ากับ $128/256 = 50\%$

ในขั้นตอนที่ 2 เป็นการคำนวณหาค่า L_a โดยใช้ค่า L_b หาดด้วย 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, L_b
2. Calculate loss rate after reconstruction

$L_a = L_b /$ Reward associated with current combination number

3. If ($L_a > \text{HIGH}$) then
Increment combination
4. If ($L_a < \text{LOW}$) then
Decrement combination

รูปที่ 3.7 Pseudo Code อัลกอริทึม Bolot

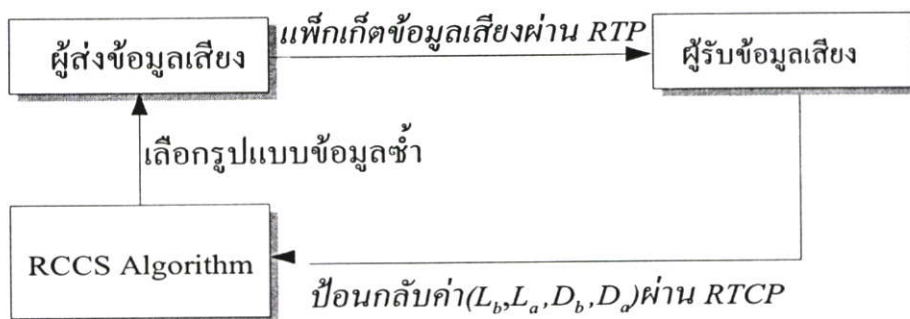
3.2.2 อัลกอริทึม RCCS

อัลกอริทึม RCCS[14] ซึ่งย่อมาจาก Redundant Codec Combination Selection เป็นอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวที่มีการเลือก Combination โดยใช้พารามิเตอร์

สองค่าคือ 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 ที่กำลังใช้งานทุกครั้งที่ได้รับแพ็กเก็ตเกิด Receiver Report และในคอลัมน์ขวาสุดคือ คอลัมน์ Penalty เป็นค่าอัตราส่วนระหว่างค่าเวลาหน่วงระหว่างปลายทางหลังการจัดเรียงข้อมูลและค่าเวลาหน่วงระหว่างปลายทางก่อนการจัดเรียงข้อมูลซ้ำ

$$\text{Penalty} = \frac{\text{ค่าเวลาหน่วงระหว่างปลายทางหลังการจัดเรียงข้อมูลซ้ำ}}{\text{ค่าเวลาหน่วงระหว่างปลายทางก่อนการจัดเรียงข้อมูลซ้ำ}} \quad (3.2)$$

ซึ่งค่า Penalty ที่อยู่ในตารางที่ 3.2 เป็นค่าเฉลี่ยที่ได้จากการทดลองหลายๆ ครั้ง แต่ใน[14] ก็ไม่ได้อธิบายเอาไว้อย่างชัดเจนว่าการหาค่าเวลาหน่วงก่อนและหลังการจัดเรียงมีวิธีการอย่างไร ซึ่งการหาค่าเวลาหน่วงระหว่างปลายทางนั้นทำได้ยาก เนื่องจากว่านาฬิกาของเครื่องผู้ส่งและเครื่องผู้รับไม่ประสานกัน จากแผนภาพการควบคุมความผิดพลาดแบบปรับตัวของอัลกอริทึม RCCS ในรูปที่ 3.8 จะเห็นว่าค่าที่ผู้รับจะต้องรายงานให้กับผู้ส่งมีอยู่ 4 ค่า คือ L_b , L_a , D_b (เวลาหน่วงระหว่างปลายทางก่อนการจัดเรียงข้อมูล) และ D_a (เวลาหน่วงระหว่างปลายทางหลังการจัดเรียงข้อมูล)



รูปที่ 3.8 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม RCCS

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 |
| 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 3.9 เป็น Pseudo Code ของอัลกอริทึม RCCS ซึ่งจะเห็นว่ามีการใช้พารามิเตอร์หลายตัว โดยคำอธิบายของพารามิเตอร์แต่ละตัวนั้นอยู่ในตารางที่ 3.3 จาก Pseudo Code จะเห็นเมื่อได้ค่า L_a , L_b , D_a และ D_b 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 L_b and D_b

L_b = Number of packet loss before reconstruction / Number of packet expected

D_b = end-to-end delay before reconstruction

2. Calculate packet loss and delay after reconstruction L_a and D_a

L_a = Number of packet loss after reconstruction / Number of packet expected

D_a = end-to-end delay after reconstruction

3. Update reward and penalty of current combination with R_i and P_i

in combination table

4. If ($L_a > \text{HIGH}$ or $L_a < \text{LOW}$)

For each combination j in the combination table

Calculate and predict L_a' , $L_a' = L_b/R_j$

Calculate and predict D_a' , $D_a' = D_b * P_j$

If ($L_a' < \text{HIGH}$ and $L_a' > \text{LOW}$)

If ($D_a' < \text{DMin}$)

selected combination = j

$\text{Dmin} = D_a'$

5. combination = selected combination

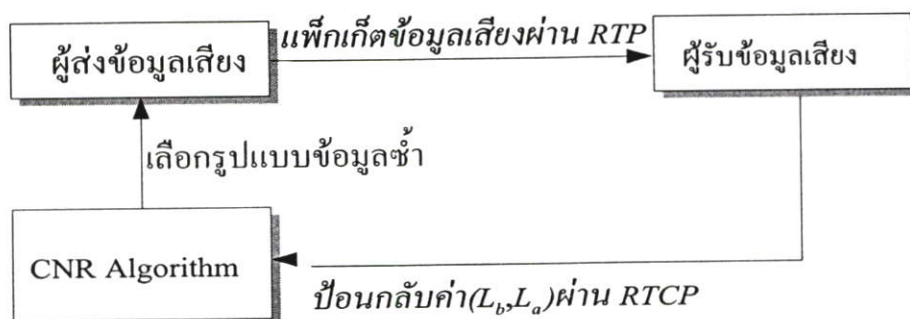
รูปที่ 3.9 Pseudo Code ของอัลกอริทึม RCCS

การแก้ไขค่า Reward นี้เพื่อให้ค่า Reward มีค่าเปลี่ยนแปลงไปตามสภาพเครือข่ายจริง เนื่องจากว่าค่า Reward ที่ได้จากการทดลองของ Bolot อาจจะไม่ได้ออกต้องเสมอไปในทุกสภาพแวดล้อม จากนั้นในขั้นตอนที่ 4 เป็นการเลือก Combination โดยจะพิจารณาจากตารางที่ 3.2 ตั้งแต่ Combination แรกจนถึง Combination สุดท้าย และจะเลือก Combination แรกที่ให้ค่า L_a' (ค่าทำนายของ L_a ซึ่งได้จากการนำ L_b หารด้วยค่า Reward) อยู่ในช่วงระหว่างเทรฮ็อลด์ LOW และ HIGH และ Combination ดังกล่าวจะต้องให้ค่า D_a' (ค่าทำนายของเวลาหน่วงระหว่างปลายทางหลังการจัดเรียงข้อมูล) ไม่เกินค่าเทรฮ็อลด์ DMin ด้วย

3.2.3 อัลกอริทึม CNR

อัลกอริทึม CNR (Centre for Network Research) ได้แก้ไขจุดบกพร่องของอัลกอริทึม Bolot โดยการใช้ค่าจริงของ L_b ในการตัดสินใจเปลี่ยน Combination และได้แก้ไขปัญหาการแกว่งของค่า L_a โดยการนับจำนวนครั้งที่ค่าของ L_a ต่ำกว่าเทรชโฮลด์ LOW โดยจะลด Combination ก็ต่อเมื่อ L_a มีค่าต่ำกว่าเทรชโฮลด์ LOW ติดต่อกันเพียงพอแล้ว เพื่อให้แน่ใจความคับคั่งของเครือข่ายลดลงจริง และอัลกอริทึม CNR ได้แก้ไขจุดบกพร่องของอัลกอริทึม RCCS โดยการแยกเงื่อนไขในการตัดสินใจลดและเพิ่ม Combination ออกจากกัน

จากแผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม CNR ใน[8] จะเห็นว่าผู้รับจะมีการรายงานค่า L_b และ L_a ให้กับผู้ส่งโดยใช้โปรโตคอล RTCP ซึ่งผู้ส่งจะนำค่าทั้งสองนี้ไปใช้ในการตัดสินใจว่าจะเลือก Combination ใด โดยแพ็กเก็ต RTCP ที่ใช้ในการรายงานค่าทั้งสองนี้ก็คือแพ็กเก็ต Receiver Report ผู้รับจะนำค่า L_b วางในฟิลด์ที่ชื่อ Fraction Lost ส่วนค่า L_a นั้นจะถูกนำไปเอาไว้ในส่วนขยายเฮดเดอร์ RTCP



รูปที่ 3.10 แผนภาพการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม CNR

รูปที่ 3.10 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[8] ได้ระบุได้ว่าความยาวของแพ็กเก็ต RTCP จะต้องมีความยาวที่หาร 4 ลงตัว ดังนั้นจึงมีส่วนที่ไม่ได้ใช้งานเหลืออยู่ 3 ไบต์ และเนื่องจากว่าใน [8] ได้กำหนดเอาไว้ว่าค่าในฟิลด์ fraction lost คือค่าอัตราการสูญหายของแพ็กเก็ตคูณด้วย 256 ดังนั้นการระบุค่า L_b และ L_a จึงใช้วิธีการเดียวกันคือ

$$\text{fraction lost} = L_b \times 256 \quad (3.5)$$

$$\text{fraction } L_a = L_a \times 256 \quad (3.6)$$

และในการควบคุมความผิดพลาดแบบปรับตัวโดยใช้อัลกอริทึม CNR เมื่อผู้ส่งได้รับแพ็กเก็ต Receiver Report ผู้ส่งก็จะอ่านค่า L_b และ L_a ที่อยู่ในแพ็กเก็ตเกิดดังนี้

$$L_b = \text{fraction loss} / 256 \quad (3.7)$$

$$L_a = \text{fraction } L_a / 256 \quad (3.8)$$

ซึ่งค่า L_b และ L_a จะถูกนำไปใช้ในการตัดสินใจว่าจะใช้ Combination ใด แต่อัลกอริทึม CNR นั้นไม่ได้มีการเพิ่ม Combination ครั้งละหนึ่งขั้นเช่นเดียวกับอัลกอริทึม Bolot แต่อัลกอริทึม CNR จะนำค่า Reward ของแต่ละ Combination มาใช้ในการเลือกว่า Combination ที่เหมาะสมกับสภาพเครือข่ายคือ Combination ใด และสิ่งที่อัลกอริทึม CNR แตกต่างจากอัลกอริทึมอื่นอีกประการหนึ่งคือ ในอัลกอริทึมอื่นๆ ข้อมูลซ้ำมักจะใช้การบีบอัดเสียงที่มีอัตราบิตต่ำกว่าข้อมูลหลัก เพื่อเป็นการประหยัดแบนด์วิดท์ แต่อัลกอริทึม CNR ใช้การบีบอัดเสียง G.723.1 ทั้งข้อมูลหลักและข้อมูลซ้ำ เนื่องจากการบีบอัดเสียงแบบนี้มีอัตราบิตที่ต่ำอยู่แล้ว (6.3 kbps) และการที่ข้อมูลหลักและข้อมูลเสียงใช้การบีบอัดเสียงที่แตกต่างกันก็ทำให้ผู้ส่งต้องเสียเวลามากขึ้น เนื่องจากต้องมีการเข้ารหัสเสียงในแต่ละเฟรมมากกว่า 1 ครั้ง แต่ถ้าใช้การบีบอัดเสียงชนิดเดียวกันก็สามารถเข้ารหัสเพียงครั้งเดียวและเก็บข้อมูลหลังการบีบอัดเอาไว้เป็นข้อมูลซ้ำในภายหลังได้เลย

| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|-----------------------------------|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| V=2 P | | | | | | | | | | RC | | | | | | | | | | PT=RR=201 | | | | | | | | | | length | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SSRC of packet sender | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SSRC_1 (SSRC of first source) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| fraction lost | | | | | | | | | | cumulative number of packets lost | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| extended highest sequence number received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| interarrival jitter | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| last SR (LSR) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| delay since last SR (DLSR) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| fraction La | | | | | | | | | | not used | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

รูปที่ 3.11 การระบุค่า L_b และ L_a ในแพ็กเก็ตเกิด Receiver Report

สำหรับ Combination ของข้อมูลซ้ำที่ใช้ในอัลกอริทึม CNR มี 6 Combination ดังแสดงในตารางที่ 3.4 โดยในคอลัมน์ Combination Format ได้กำหนดสัญลักษณ์เป็นจำนวนเต็มลบเพื่อใช้แทนรูปแบบของข้อมูลซ้ำ ตัวอย่างเช่น สัญลักษณ์ -1 หมายความว่า ในแต่ละแพ็กเก็ตลำดับที่ N จะมีข้อมูลซ้ำของแพ็กเก็ตลำดับที่ 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 3.12 เป็น Pseudo Code ของอัลกอริทึม CNR เมื่อผู้ส่งได้รับแพ็กเก็ต Receiver Report ก็จะอ่านค่า L_b และ L_u ในขั้นตอนแรกจะแก้ไขค่า Reward ของ Combination ปัจจุบัน ส่วนในขั้นตอนที่สองเป็นการนับจำนวนครั้งที่ค่า L_b มีค่าต่ำกว่าเทรชโฮลด์ LOW (เก็บไว้ในตัวแปร `count_Lb_under_low`) ซึ่งตัวแปรตัวนี้จะมีการตรวจสอบอีกครั้งเมื่อมีการตัดสินใจลด Combination ในขั้นตอนที่สามเป็นการตรวจสอบว่าถ้า L_u ที่อ่านจากแพ็กเก็ต Receiver Report มีค่าสูงกว่าเทรชโฮลด์ HIGH ก็จะต้องมีการเพิ่ม Combination ซึ่งทำได้โดยนำค่า L_b และค่า Reward ของ Combination ที่อยู่ในลำดับถัดจาก Combination ปัจจุบันเป็นต้นไปมาประมาณว่าถ้าเลือกใช้แต่ละ Combination แล้วจะได้ค่า L_u เท่าไร จากนั้นก็จะเลือก Combination แรกที่ให้ค่าประมาณของ L_u (ตัวแปร L_{ai} ใน Pseudo Code) ไม่เกินเทรชโฮลด์ 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 $\text{count_}L_a\text{_under_low}$) การลด Combination ในอัลกอริทึมจะเกิดขึ้นได้ก็ต่อเมื่อค่า $\text{count_}L_a\text{_under_low}$ หรือค่า $\text{count_}L_b\text{_under_low}$ มีค่ามากกว่าหรือเท่ากับ MIN_UNDER_LOW โดยการที่ค่าของ $\text{count_}L_a\text{_under_low}$ มีค่ามากกว่าหรือเท่ากับ MIN_UNDER_LOW หมายความว่า Combination ปัจจุบันที่ใช้อยู่สามารถทำให้ค่า L_a ต่ำกว่าเทรสโฮลด์ LOW เป็นระยะเวลาสั้นพอแล้ว และปลอดภัยพอที่จะลด Combination ซึ่งเทคนิคนี้จะช่วยลดผลกระทบที่เกิดจากการแกว่งของค่า L_a ได้

```

For each receiver report packet received do
1.Update reward of the current combination
  //Let  $R_c$  stands for reward of the current combination
   $R_c = L_b/L_a$ 

2.if( $L_b < \text{LOW}$ )
  increment  $\text{count\_}L_b\text{\_under\_low}$ 
else
   $\text{count\_}L_b\text{\_under\_low} = 0$ 

3.if( $L_a > \text{HIGH}$ )
  selected_combination = MAX_COMBINATION;
  for( $i = \text{current\_combination}+1$  to MAX_COMBINATION)
     $R_i = \text{reward of combination No.}i$ 
     $L_{ai} = L_b/R_i$ 
    if( $L_{ai} \leq \text{HIGH}$ ) {
      selected_combination =  $i$ 
      break;
    }
  current_combination = selected_combination;
   $\text{count\_}L_a\text{\_under\_low} = 0$ 

else if( $L_a < \text{LOW}$ )
  increment  $\text{count\_}L_a\text{\_under\_low}$ 

else
   $\text{count\_}L_a\text{\_under\_low} = 0$ 

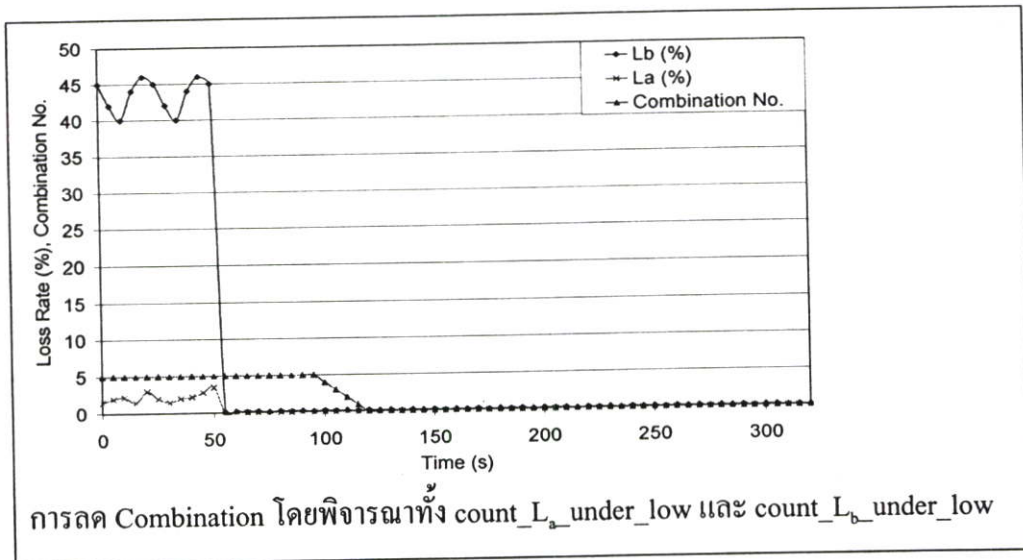
4.if(( $\text{count\_}L_a\text{\_under\_low} \geq \text{MIN\_UNDER\_LOW}$ ) OR
  ( $\text{count\_}L_b\text{\_under\_low} \geq \text{MIN\_UNDER\_LOW}$ ))
  decrement current_combination
   $\text{count\_}L_a\text{\_under\_low} = 0$ 

```

รูปที่ 3.12 Pseudo Code ของอัลกอริทึม CNR

การลด Combination เมื่อค่า $\text{count_}L_a\text{_under_low}$ หรือค่า $\text{count_}L_b\text{_under_low}$ มีค่าต่ำกว่าค่า MIN_UNDER_LOW ดังแสดงในรูปที่ 3.13 ซึ่งค่าของ MIN_UNDER_LOW นั้น ถ้าหากมีค่ามากจะทำให้ป้องกันการแกว่งของค่า L_a ได้ดี แต่การลด Combination ทำได้ช้า และส่งผลให้มี

ปริมาณการใช้แบนด์วิดท์สูงเนื่องจากใช้ปริมาณข้อมูลซ้ำมาก ในที่นี้เลือกใช้ค่า MIN_UNDER_LOW เท่ากับ 10 ซึ่งมีค่าเท่ากับระยะเวลา 50 วินาทีเนื่องจากเป็นค่าที่มากพอที่จะลดปัญหาการแกว่งของค่า L_a ได้ดี



รูปที่ 3.13 การเปลี่ยนแปลงการลด Combination (โดยพิจารณาทั้ง $count_{L_a_under_low}$ และ $count_{L_b_under_low}$)

3.3 การประยุกต์ใช้วิธีกำลังสองน้อยที่สุดในอัลกอริทึมแบบปรับตัว CNR

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 ซึ่งอัลกอริทึม RCCS มีจุดบกพร่องในส่วนของทางเลือก Combination ซึ่งไม่ได้มีการแยกเงื่อนไขในการลดและเพิ่ม Combination จาก Pseudo Code ของอัลกอริทึม RCCS ในรูปที่ 3.9 จะเห็นว่าอัลกอริทึมนี้จะใช้ค่า Reward เพื่อค่า L_a' (ค่าทำนายของ L_a) ในแต่ละ Combination และจะเลือก Combination ที่มีค่า L_a' ที่มีค่าต่ำกว่าเทรชโฮลด์ HIGH และสูงกว่าเทรชโฮลด์ LOW ซึ่งการใช้เงื่อนไขร่วมกันอย่างนี้อาจทำให้อัลกอริทึมนี้ไม่สามารถเพิ่ม Combination ได้ในบางครั้ง เช่น ถ้า Combination ในขั้นที่สูงกว่า Combination ปัจจุบันทุก Combination ต่างก็มีค่า L_a' ที่ต่ำกว่าเทรชโฮลด์ LOW ก็ถือว่าไม่ตรงตามเงื่อนไขแล้ว ซึ่งทำให้ไม่สามารถเพิ่ม Combination ได้ถึงแม้ว่าในขณะนั้นค่า L_a จะมีค่าสูงเกินกว่าเทรชโฮลด์ HIGH ก็ตาม การที่จะวัดว่าอัลกอริทึมใดดีกว่ากัน ต้องดูว่าอัลกอริทึมเหล่านี้สามารถเลือกใช้ Combination ที่สามารถรักษา

ระดับของ L_u ไม่ให้สูงเกินไปได้หรือไม่ รวมทั้งต้องไม่ใช่ปริมาณข้อมูลซ้ำมากเกินไปจนเกิดความจำเป็นอีกด้วย เพราะถ้าไม่คำนึงถึงปริมาณแบนด์วิธที่เพิ่มขึ้นจากการใช้ข้อมูลซ้ำแล้ว ก็ไม่จำเป็นจะต้องใช้อัลกอริทึมใดเลยก็ได้ เพราะยิ่งใช้ข้อมูลซ้ำมากก็ยังสามารถกู้ข้อมูลที่สูญหายได้มาก แต่การใช้ อัลกอริทึมในการควบคุมความผิดพลาดเหล่านี้ก็เพื่อที่จะกำหนดให้ใช้ปริมาณข้อมูลซ้ำเท่าที่จำเป็นเท่านั้นและเพื่อให้ใช้แบนด์วิธได้อย่างคุ้มค่า ส่วนอัลกอริทึม CNR มีการปรับค่าของ reward value ที่เป็นอัตราส่วนของ L_b ต่อ L_u เพื่อให้มีการปรับค่า reward value ให้เหมาะสมกับเครือข่ายในขณะนั้น ๆ ในวิทยานิพนธ์ฉบับนี้จึงนำเทคนิคการประมาณค่าด้วยวิธีกำลังสองน้อยที่สุดมาใช้ในการปรับค่า reward ของอัลกอริทึม CNR แล้วเปรียบเทียบกับการใช้ค่า reward value แบบกำหนดค่าในอัลกอริทึม Bolot และจากรูปที่ 3.7 ซึ่งแสดง Pseudo Code อัลกอริทึม Bolot จะเห็นว่า อัลกอริทึม Bolot มีจุดบกพร่องในส่วนของ การตัดสินใจเพิ่มหรือลด Combination โดยพิจารณาจากค่า L_u แต่ค่า L_u ที่ใช้ไม่ใช่ค่าจริง แต่เป็นค่า L_u ที่ได้จากการคำนวณโดยการนำค่า L_b มาร่วมด้วยค่า Reward ของ Combination ปัจจุบัน ซึ่งค่า L_u จากการคำนวณดังกล่าวอาจจะไม่ถูกต้องเสมอไปในทุกสภาพแวดล้อม และอาจส่งผลให้อัลกอริทึมนี้ทำงานผิดพลาดได้ และเงื่อนไขในการปรับเพิ่มหรือลด Combination นั้นจะปรับเพิ่มหรือลดทีละ 1 Combination ซึ่งถ้าค่า reward value ในขณะนั้นยังไม่เหมาะสมกับสภาพเครือข่ายในขณะนั้น ก็จะใช้เวลานานกว่าจะทำการปรับเลือกค่า Combination เพื่อควบคุมความผิดพลาด โดยการกู้คืนแพ็กเก็ตที่สูญเสียให้เหมาะสมกับสภาพเครือข่ายในขณะนั้น

โดยในอัลกอริทึม CNR ได้นำเทคนิคการประมาณค่าด้วยวิธีกำลังสองน้อยที่สุดมาใช้ โดยเปลี่ยนค่า reward value จากสมการที่ 3.1 ซึ่งเป็นค่าเดียวกันกับการหาค่า reward value (R_c) ในรูปที่ 3.12 ซึ่งแสดง Pseudo Code ของอัลกอริทึม CNR ดังนั้นจะได้ค่า reward value ที่นำไปปรับค่า combination number ของแต่ละรอบการทำงานถัด ๆ ไปดังสมการ 3.9

$$R_c = A + B(L_b/L_u) \quad (3.9)$$

โดยค่า A และ B เป็นค่าคงที่ ที่หาได้จากสมการที่ 2.6

3.4 การตรวจสอบความถูกต้องของแบบจำลอง

เนื่องจากในการทดลองเพื่อประเมินผลอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวแต่ละอัลกอริทึมทำโดยการใช้แบบจำลอง ดังนั้นก่อนที่จะทำการทดลองจำเป็นต้องแน่ใจเสียก่อนว่าแบบจำลองที่จะนำมาใช้ในการทดลองนั้นสามารถทำงานได้อย่างถูกต้อง ซึ่งการที่จะยืนยันว่าแบบจำลองที่ใช้ในการทดลองนั้นสามารถทำงานได้อย่างถูกต้องจำเป็นต้องมีการตรวจสอบสองขั้นตอน โดยขั้นตอนแรกคือตรวจสอบการคำนวณค่า L_b และ L_u ของผู้รับ และ

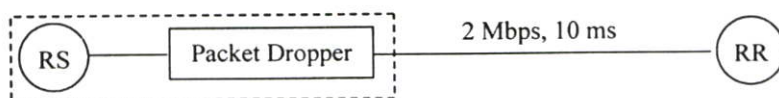
ขั้นตอนที่สองคือตรวจสอบการทำงานของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวแต่ละอัลกอริทึม

3.4.1 การตรวจสอบการคำนวณค่า L_b และ L_r ของผู้รับ

เนื่องจากค่า L_b และค่า L_r เป็นค่าที่จำเป็นสำหรับอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวที่จะต้องนำไปใช้ในการตัดสินใจเลือก Combination ของข้อมูลซ้ำ ดังนั้นหากเกิดข้อผิดพลาดในการคำนวณค่าทั้งสอง จะส่งผลให้การทำงานในขั้นตอนต่อไปเกิดข้อผิดพลาดตามไปด้วย

ในการทดลองนี้ได้กำหนดรูปแบบการเชื่อมต่อภายในเครือข่าย ดังรูปที่ 3.14 ซึ่งจะเห็นว่า มีโหนดเพียง 2 โหนดเท่านั้น โดยโหนด RS จะส่งแพ็กเก็ตเสียงไปให้โหนด RR แต่ที่โหนด RS จะมีตัวตัดทิ้งแพ็กเก็ต (Packet Dropper) ที่สามารถตัดทิ้งแพ็กเก็ตได้ตามรูปแบบที่กำหนด ทั้งนี้เพื่อใช้ในการทดสอบว่าการทำงานของแบบจำลองในส่วนของการคำนวณค่า L_b และ L_r ทำได้ถูกต้องหรือไม่ โดยได้กำหนดรูปแบบการตัดทิ้ง (Drop) แพ็กเก็ตทั้งหมด 5 รูปแบบดังตารางที่ 3.5 ซึ่งจากรูปแบบการตัดทิ้งแพ็กเก็ตดังกล่าวสามารถที่จะคำนวณค่า L_b และ L_r ที่ถูกต้องได้เลย ซึ่งหากแบบจำลองสามารถทำงานได้อย่างถูกต้อง ค่า L_b และ L_r 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_r โดยส่งมาในแพ็กเก็ต Receiver Report ให้กับโหนด RS ซึ่งแพ็กเก็ตนี้มีการส่งทุก 5 วินาที
- โหนด RS เริ่มส่งแพ็กเก็ตเสียงตั้งแต่วันที่ 0 และหยุดที่วินาทีที่ 500
- ค่า L_b และ L_r ที่ได้จากการทดลองเป็นค่าเฉลี่ยของค่า L_b และ L_r ที่ RR รายงานให้กับ RS ทุกครั้ง (100 ครั้ง)



รูปที่ 3.14 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง

ตารางที่ 3.5 เป็นผลการตรวจสอบความถูกต้องในการคำนวณค่าของ L_b ของผู้รับ เนื่องจากค่า L_b คืออัตราการสูญหายของข้อมูลเสียงก่อนการจัดเรียง ซึ่งก็คืออัตราการสูญหายของ

แพ็กเกจนั้นเอง ดังนั้นในกรณีนี้ค่า L_u ที่ถูกต้องจะต้องมีค่าเท่ากับอัตราการตัดทิ้งแพ็กเกจ เนื่องจากในการทดลองการสูญหายของแพ็กเกจเกิดจากการตัดทิ้งเพียงอย่างเดียวเท่านั้น และไม่ว่าจะใช้จำนวนข้อมูลเข้าเท่าใดก็จะมีผลต่อค่า L_u เพราะว่าเป็นการพิจารณาในขณะที่ยังไม่มีการนำข้อมูลเข้ามาดูข้อมูลในแพ็กเกจที่สูญหาย จากผลการทดลองในตารางที่ 3.5 จะเห็นว่าการคำนวณค่า L_u ของผู้รับสามารถทำได้ถูกต้องในทุกกรณี นั่นคือค่า L_u มีค่าเท่ากับอัตราการตัดทิ้งแพ็กเกจเสมอไม่ว่าจะใช้รูปแบบการตัดทิ้งแพ็กเกจแบบใด

ตารางที่ 3.5 การตรวจสอบความถูกต้องในการคำนวณค่า L_u ของผู้รับในแบบจำลอง

| รูปแบบในการตัดทิ้ง | หมายเลขแพ็กเกจที่ถูกตัดทิ้ง (ในทุก 100 แพ็กเกจ) | อัตราการตัดทิ้งแพ็กเกจ (%) | ค่า L_u (%) 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_u ของผู้รับในแบบจำลอง ซึ่งมีการตัดทิ้งแพ็กเกจตามรูปแบบเดียวกันกับที่ใช้ในตารางที่ 3.5 ซึ่งการตัดทิ้งแพ็กเกจทั้ง 5 รูปแบบสามารถคำนวณได้ล่วงหน้าแล้วว่าค่า L_u ที่ถูกต้องของแต่ละ Combination มีค่าเท่าใด ดังนั้นหากแบบจำลองสามารถทำงานได้ถูกต้อง ค่า L_u ที่จากการทดลองจะต้องมีค่าตรงกับค่าที่ถูกต้อง

รูปที่ 3.15 เป็นตัวอย่างของการหาค่า L_u เมื่อกำหนดให้มีการตัดทิ้งแพ็กเกจแบบ D02 ซึ่งจากตารางที่ 3.6 จะเห็นว่า การตัดทิ้งแพ็กเกจแบบนี้ ในทุก 100 แพ็กเกจจะมีการตัดทิ้งแพ็กเกจเป็นจำนวน 30 แพ็กเกจ (แพ็กเกจในหมายเลขที่ลงท้ายด้วย 8, 9 และ 0) หรืออาจจะกล่าวได้ว่าในทุก 10 แพ็กเกจจะมีการตัดทิ้ง 3 แพ็กเกจ ดังนั้นในการหาค่า L_u ที่ถูกต้องก็สามารถพิจารณาเฉพาะ 10 แพ็กเกจ และหาว่าหลังจากมีการดูข้อมูลในแพ็กเกจที่สูญหายแล้วมีข้อมูลเสียหายไปที่เฟรมก็สามารถหาค่า L_u ได้แล้ว ซึ่งจะเห็นได้ว่าค่า L_u จากการทดลองของ Combination 0, 1 และ 2 ที่ได้จากการทดลองมีค่าถูกต้อง สำหรับในกรณีอื่นๆ หลังจากที่ได้มีการตรวจสอบแล้วพบว่า

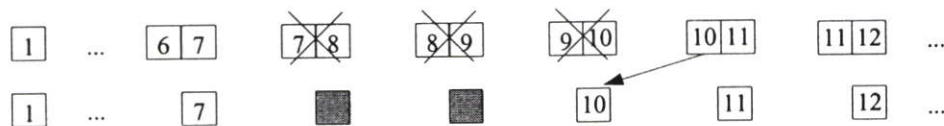
ค่านวนค่า L_u ของแบบจำลองสามารถทำงานได้ถูกต้องในทุกกรณี

ตารางที่ 3.6 การตรวจสอบความถูกต้องในการคำนวณค่า L_u ของผู้รับในแบบจำลอง

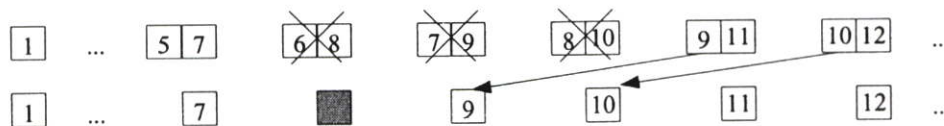
| รูปแบบในการตัดทิ้ง | หมายเลขแพ็กเก็ตที่ถูกตัดทิ้ง (ในทุก 100 แพ็กเก็ต) | อัตราการตัดทิ้งแพ็กเก็ต (%) | ค่า L_u (%) จากการทดลองของแต่ละ Combination | | | | | |
|--------------------|---|-----------------------------|---|----|----|----|----|---|
| | | | 0 | 1 | 2 | 3 | 4 | 5 |
| D01 | 10N; N = 1, 2, ...,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, $L_u = 30\%$



(b) Combination 1, ทุก 10 แพ็กเก็ตจะมีข้อมูลเสียงสูญหาย 2 เฟรม, $L_u = 20\%$



(c) Combination 2, ทุก 10 แพ็กเก็ตจะมีข้อมูลเสียงสูญหาย 1 เฟรม, $L_u = 10\%$

รูปที่ 3.15 ตัวอย่างการหาค่า L_u เมื่อกำหนดรูปแบบการตัดทิ้งแพ็กเก็ตแบบ D02

3.4.2 การตรวจสอบการทำงานของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัว

เมื่อแน่ใจแล้วว่าการคำนวณค่า L_u และ L_d ในแบบจำลองสามารถทำได้ถูกต้อง ขั้นตอนต่อไปคือการตรวจสอบการทำงานของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัว

ของ 2 อัลกอริทึมคือ อัลกอริทึม Bolot และอัลกอริทึม CNR ซึ่งการทดลองในหัวข้อนี้ได้กำหนดให้ใช้เครือข่ายในการจำลอง ดังรูปที่ 3.14 โดยมีการจำลอง 4 ครั้ง (สำหรับ 4 Combination) และเพื่อที่จะตรวจสอบว่าการทำงานของแบบจำลองในส่วนของอัลกอริทึมในการควบคุมความผิดพลาดแบบปรับตัวว่าสามารถทำงานได้อย่างถูกต้องตามวิธีการที่ได้อธิบายในหัวข้อ 3.1 หรือไม่ จึงได้กำหนดรูปแบบการตัดทิ้งแพ็กเก็ตในแต่ละช่วงเวลาให้แตกต่างกัน เพื่อจะให้เห็นว่าการเปลี่ยนแปลง Combination ของข้อมูลซ้ำเป็นไปอย่างถูกต้องหรือไม่ และในการจำลองแต่ละครั้ง พฤติกรรมการส่งข้อมูลของโหนด RS และ RR เป็นดังนี้

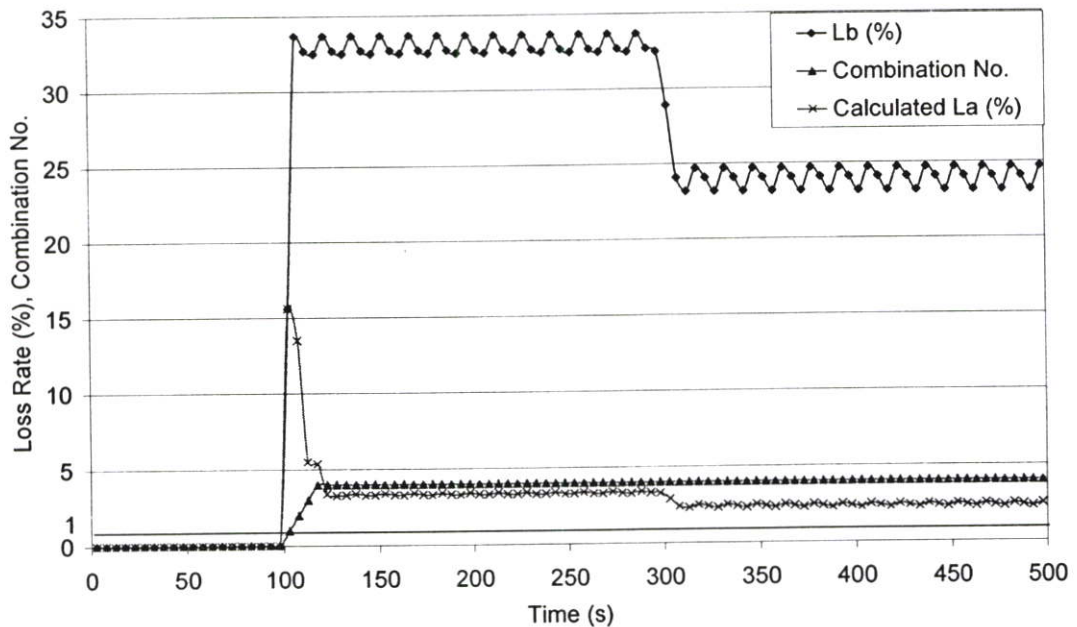
- โหนด RS ส่งแพ็กเก็ตเสียงโดยใช้โปรโตคอล RTP ไปให้โหนด RR โดยส่งแพ็กเก็ตทุกๆ 30 ms ซึ่งเป็นระยะเวลาของเสียงพูด 1 เฟรมสำหรับการบีบอัดเสียงด้วย G.723.1 ส่วนขนาดแพ็กเก็ตขึ้นอยู่กับ Combination ที่ใช้
- โหนด RR จะมีการรายงานค่า L_r และ L_s โดยส่งมาในแพ็กเก็ต Receiver Report ให้กับโหนด RS ซึ่งแพ็กเก็ตนี้มีการส่งทุก 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[6] และกำหนดค่าเทรสโฮลด์ LOW เท่ากับ 1% ซึ่งถือว่าเป็นค่าที่ต่ำพอที่จะลองเพิ่ม Combination ได้แล้ว

3.4.2.1 การตรวจสอบการทำงานของอัลกอริทึม Bolot

รูปที่ 3.16 เป็นกราฟแสดงผลการจำลองโดยใช้อัลกอริทึม Bolot ซึ่งจะเห็นว่าค่า L_r ในแต่ละช่วงเวลาตรงกับอัตราการตัดทิ้งแพ็กเก็ตตามที่ได้กำหนดไว้ข้างต้น นั่นคือช่วง 100 วินาทีแรกค่า L_r เป็น 0 % ส่วนช่วงวินาทีที่ 101 – 300 ค่า L_r เฉลี่ยมีค่าประมาณ 33 % ส่วนวินาทีที่ 301 – 500 ค่า L_r เฉลี่ยมีค่าประมาณ 24 % อัลกอริทึม Bolot ไม่ได้ค่าจริงของ L_s แต่คำนวณค่า L_s โดยใช้ค่า L_r

หารด้วยค่า Reward ของ Combination ที่ใช้อยู่ จากกราฟในรูปที่ 3.16 สามารถยืนยันได้ว่าค่า L_u จากการคำนวณของแบบจำลองมีค่าถูกต้อง ตัวอย่างเช่น หลังจากวินาทีที่ 300 เป็นต้นไปมีการใช้ Combination ที่ 4 ซึ่งจากตารางที่ 3.4 ค่า Reward ของ Combination นี้เท่ากับ 10 และในช่วงเวลานี้ค่า L_u เฉลี่ยอยู่ที่ 24% ดังนั้นค่า L_u จากการคำนวณจะต้องอยู่ที่ 2.4 % ซึ่งตรงกับที่ได้แสดงอยู่ในกราฟรูปที่ 3.16



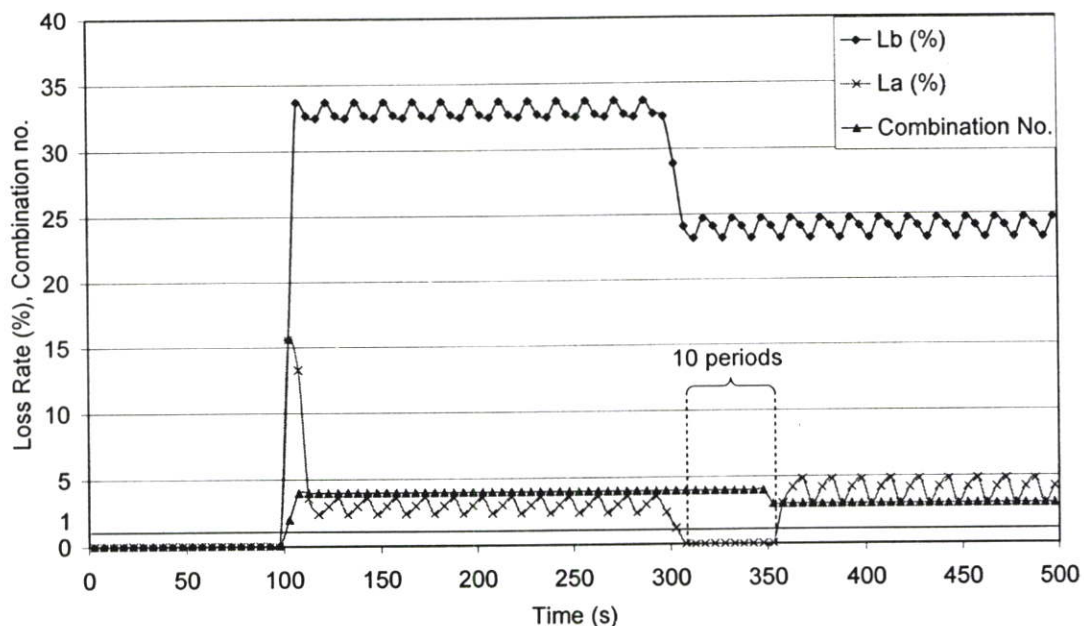
รูปที่ 3.16 ค่า L_u และค่า L_b จากการคำนวณในระหว่างการจำลองโดยใช้อัลกอริทึม Bolot

หลังจากที่ได้ตรวจสอบแล้วค่า L_u จากการคำนวณนั้นได้มาอย่างถูกต้อง ขั้นตอนต่อไปก็คือ การตรวจสอบการเปลี่ยน Combination ว่าถูกต้องตาม Pseudo Code ของอัลกอริทึม Bolot ในรูปที่ 3.7 หรือไม่ ซึ่งกราฟในรูปที่ 3.16 ก็สามารถยืนยันได้ว่าแบบจำลองสามารถทำงานได้ถูกต้อง โดยในการทดลองนี้ได้กำหนดให้เทรสโฮลด์ LOW และ HIGH เป็น 1% และ 5% ตามลำดับ จะเห็นว่าหลังจากวินาทีที่ 100 เมื่อค่า L_u (จากการคำนวณ) สูงเกินกว่า 5% จะมีการเพิ่ม Combination ขึ้นที่ ละหนึ่งขั้นและหยุดที่ Combination ที่ 4 เนื่องจากค่า L_u ลดลงมาต่ำกว่า 5% และหลังจากวินาทีที่ 300 ถึงแม้ว่าค่า L_u จะลดลงแต่ไม่ถึง 1% ดังนั้นจึงไม่มีการลด Combination

3.4.2.2 การตรวจสอบการทำงานของอัลกอริทึม CNR

จากการตรวจสอบการทำงานของแบบจำลองเมื่อมีการควบคุมความผิดพลาดแบบปรับตัว โดยใช้อัลกอริทึม CNR การเปลี่ยนแปลง Combination เป็นไปตามกราฟในรูปที่ 3.17 ซึ่งจะเห็นว่า

แบบจำลองสามารถทำงานได้ถูกต้องตาม Pseudo Code ของอัลกอริทึม CNR ในรูปที่ 3.12



รูปที่ 3.17 การเปลี่ยนแปลงของ Combination ในระหว่างการจำลองโดยใช้อัลกอริทึม CNR

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

บทที่ 4

ผลการทดลองและการวิเคราะห์

จากการศึกษาวิธีการควบคุมความผิดพลาดแบบปรับตัว โดยมีการนำเทคนิคการประมาณค่า ด้วยวิธีกำลังสองน้อยที่สุด มาใช้ในการวิเคราะห์หอนุกรมเวลาที่มีข้อมูลผิดปกติในการควบคุมความผิดพลาดเสียงตามวิธีการที่ได้เสนอขึ้นในบทที่ 3 ในบทนี้จะทำการประเมินผลอัลกอริทึม CNR โดยมีการเปรียบเทียบกับการใช้วิธีการปรับให้เรียบ (Single Exponential Smoothing) ในอัลกอริทึม CNR[16] และ เปรียบเทียบกับอัลกอริทึม Bolot

4.1 การทดลองเพื่อประเมินผลอัลกอริทึม CNR

การทดลองในหัวข้อนี้เป็นการประเมินผลอัลกอริทึม CNR โดยมีการเปรียบเทียบกับอัลกอริทึม Bolot ในกรณีที่ใช้การบีบอัดเสียง G.723.1 ทั้งข้อมูลหลักและข้อมูลซ้ำ สำหรับ Combination ของข้อมูลซ้ำที่ใช้ในการทดลองมี 6 Combination ดังแสดงในตารางที่ 4.1 และเนื่องจากว่าแต่ละอัลกอริทึมต่างก็มีพิกัดค่าเทรสโพลด์ HIGH และ LOW ในการเลือก Combination ดังนั้นในการเปรียบเทียบสมรรถนะของแต่ละอัลกอริทึม นั้น จำเป็นจะต้องกำหนดค่าเทรสโพลด์ให้เท่ากัน ในงานวิจัยก่อนหน้านี้ได้ระบุเอาไว้ว่าหากมีข้อมูลเสียงสูญหายมากกว่า 5% จะทำให้รู้สึกได้ว่าเสียงมีคุณภาพลดลง ดังนั้นการทดลองนี้จึงเลือกใช้ค่าเทรสโพลด์ HIGH เท่ากับ 5% และเพื่อให้ผลการทดลองมีความน่าเชื่อถือมากขึ้น ในการทดลองนี้จึงได้มีการจำลองในกรณีที่ใช้ค่าเทรสโพลด์ HIGH เท่ากับ 3% ด้วย ซึ่งกรณีนี้ก็อาจจะเป็นไปได้ในการใช้งานจริง เช่น ผู้ใช้ต้องการคุณภาพเสียงที่ดีมากโดยยอมรับการสูญหายของข้อมูลเสียงได้ไม่เกิน 3% เป็นต้น ส่วนค่าเทรสโพลด์อีกค่าหนึ่งคือค่าเทรสโพลด์ LOW กำหนดให้มีค่าเท่ากับ 1% ซึ่งค่าของเทรสโพลด์ LOW นี้ไม่ควรกำหนดให้มีค่าใกล้เคียงกับค่าเทรสโพลด์ HIGH มากนัก เพื่อป้องกันไม่ให้เกิดการเปลี่ยน Combination บ่อยเกินไป

ในการทดลองนี้ไม่ได้มีการส่งข้อมูลในเครือข่ายจริง แต่เป็นการจำลองบน NS-2 ดังแสดงในรูปที่ 4.1 ได้แสดงการเชื่อมต่อของแต่ละโหนดที่อยู่ภายในเครือข่ายที่ใช้ในการจำลอง ลิงค์(Link) ระหว่างโหนด 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 นอกจากนี้ยังมีโหนดที่คอยส่งกราฟฟิกชนิดอื่นๆ เพื่อให้เครือข่ายมีสภาพแวดล้อมเหมือน

มีผู้ใช้งานคนอื่นกำลังใช้เครือข่ายอยู่ด้วย โดยมีทั้งผู้ใช้ที่ส่งทราฟฟิก UDP และผู้ใช้ที่ส่งทราฟฟิก TCP

ตารางที่ 4.1 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 |

ในที่นี้ได้แบ่งการทดลองออกเป็น 2 ตอน โดยสิ่งที่แตกต่างกันก็คือ พฤติกรรมในการส่งข้อมูลของโหนดที่ส่งทราฟฟิก TCP โดยในตอนที่ 1 โหนดที่ส่งทราฟฟิก TCP จะมีการส่งข้อมูลติดต่อกันเป็นเวลานานโดยไม่หยุด ส่วนในตอนที่ 2 โหนดที่ส่งทราฟฟิก TCP จะมีการส่งและหยุดส่งข้อมูลสลับกันไปเป็นระยะ ซึ่งในการทดลองนี้ก็มีการรันแบบจำลอง 2 ครั้งต่อหนึ่งอัลกอริทึม แตกต่างกับในตอนที่ 1 โดยครั้งแรกใช้จำนวนโหนด TCP เท่ากับ 20 โหนด และครั้งที่สองใช้จำนวนโหนด TCP เป็น 40 โหนด เพื่อสังเกตการณ์การปรับตัว เสมือนเพิ่มผู้ส่งและผู้รับขึ้นในเครือข่าย ทั้งนี้เพื่อที่จะประเมินสมรรถนะของอัลกอริทึม CNR ในสภาพเครือข่ายที่แตกต่างกัน เพื่อเพิ่มความน่าเชื่อถือของผลการทดลอง

4.1.1 การทดลองตอนที่ 1

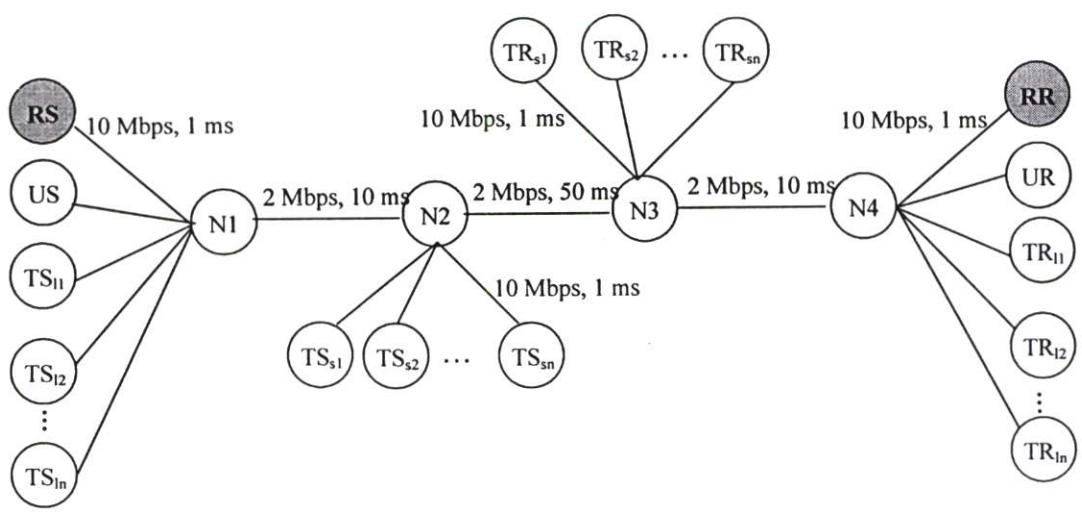
ในการทดลองตอนที่ 1 เป็นการจำลองให้มีการสื่อสารเสียงในเครือข่ายที่มีผู้ใช้ที่ส่งทราฟฟิก TCP เป็นช่วงเวลานาน และมีปริมาณการใช้แบนด์วิดท์เต็มความจุของเครือข่าย ซึ่งในสภาพเครือข่ายเช่นนี้สามารถทำให้เกิดการแกว่งของค่าอัตราการสูญหายของแพ็กเก็ตในการสื่อสารเสียงได้ โดยมีจุดประสงค์เพื่อประเมินสมรรถนะของแต่ละอัลกอริทึมว่าจะได้รับผลจากปัญหาการแกว่งของค่าอัตราการสูญหายของแพ็กเก็ตมากน้อยเพียงใด และในกรณีที่สภาพเครือข่ายเป็นเช่นนี้ อัลกอริทึมใดสามารถทำงานได้ผลดีที่สุด ซึ่งการส่งข้อมูลของแต่ละโหนดเป็นดังนี้

- โหนด RS ส่งเสียงโดยใช้โปรโตคอล RTP ไปให้โหนด RR โดยส่งแพ็กเก็ตทุกๆ 30 มิลลิวินาที ซึ่งระยะเวลาเท่ากับขนาดเฟรมของ G.723.1 ส่วนขนาดแพ็กเก็ตขึ้นอยู่กับ

Combination ที่ใช้ ถ้า Combination ใดมีจำนวนข้อมูลซ้ำมาก แเพ็กเก็ตเสียงที่ส่งก็จะมีขนาดใหญ่

- โหนด RR ส่งแพ็กเก็ต Receiver Report ให้กับโหนด RS และเนื่องจากในที่นี่เป็นการติดต่อแบบจุดต่อจุด ดังนั้นแพ็กเก็ต Receiver Report นี้จะมีการส่งทุกๆ 5 วินาที
- โหนด US ส่งทราฟฟิก UDP ด้วยอัตราบิต 400 กิโลบิตต่อวินาที โดยใช้แพ็กเก็ตขนาด 1024 ไบต์ ซึ่งโหนด US ทำหน้าที่เป็นตัวแทนของโปรแกรมประยุกต์ในการส่งเสียงหรือวีดิทัศน์ที่ไม่มีการปรับตัวตามสภาพเครือข่าย
- โหนด $TS_{i1} - TS_{in}$ และ $TS_{s1} - TS_{sn}$ ส่งไฟล์โดยใช้โปรโตคอล FTP ไปให้กับโหนด $TR_{i1} - TR_{in}$ และ $TR_{s1} - TR_{sn}$ ตามลำดับ โดยเริ่มส่งที่วินาทีที่ 100 และหยุดส่งเมื่อการจำลองยุติ ดังนั้นในช่วง 100 วินาทีแรกจะไม่มีแพ็กเก็ตเสียงสูญหายเลย
- การจำลองยุติที่วินาทีที่ 2400

โดยในการทดลองตอนนี้กำหนดให้การจำลองในแต่ละอัลกอริทึมมีการรันแบบจำลอง 1 ครั้ง ที่โหนด $TS_{i1} - TS_{in}$ และ $TS_{s1} - TS_{sn}$ เริ่มส่งและหยุดส่งข้อมูลพร้อมกันในวินาทีที่ 100 วินาที และหยุดส่งที่วินาทีที่ 400 เนื่องจากว่าต้องการประเมินสมรรถนะของแต่ละอัลกอริทึมในสภาพแวดล้อมที่ระดับความคับคั่งของเครือข่ายไม่คงที่ และสังเกตผลการทดลองว่าจะเป็นไปได้ในทำนองเดียวกันหรือไม่



รูปที่ 4.1 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง

4.1.2 การทดลองตอนที่ 2

ในการทดลองตอนที่ 2 นี้ ลักษณะการส่งข้อมูลของโหนด RS และ US ยังเหมือนกับในการทดลองตอนที่ 1 แต่มีการเปลี่ยนแปลงที่โหนด TS_{11} - TS_{1n} และ TS_{21} - TS_{2n} ซึ่งจะเริ่มส่งและหยุดส่งข้อมูลพร้อมกันในวินาทีที่ 800 วินาที และหยุดส่งที่วินาทีที่ 1200 ทั้งนี้เนื่องจากว่าต้องการประเมินสมรรถนะของแต่ละอัลกอริทึมในสภาพแวดล้อมที่ระดับความคับคั่งของเครือข่ายไม่คงที่ ซึ่งในการทดลองนี้ก็มีการรันแบบจำลอง 2 ครั้งต่อหนึ่งอัลกอริทึมแตกต่างกับในตอนที่ 1 โดยครั้งแรกใช้จำนวนโหนด TCP เท่ากับ 20 โหนด และครั้งที่สองใช้จำนวนโหนด TCP เป็น 40 โหนด

4.2 ผลการทดลอง

4.2.1 ผลการทดลองตอนที่ 1

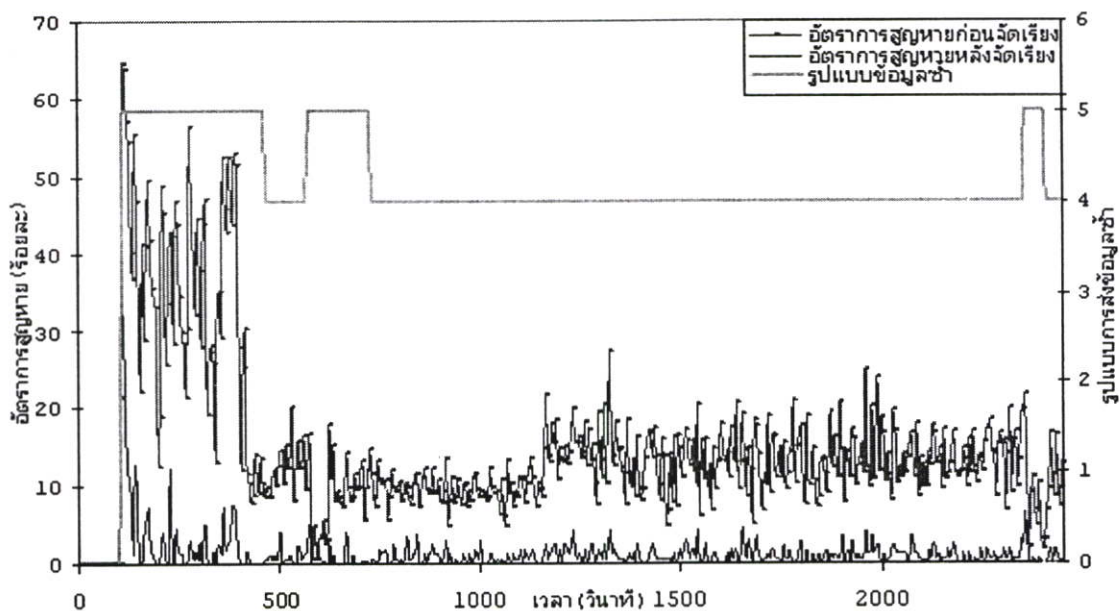
การทดลองนี้ได้ทำการจำลองให้มีกราฟฟิกของข้อมูลในเครือข่ายมีมากที่วินาทีที่ 100 ถึงวินาทีที่ 400 จากนั้นเริ่มสังเกตการณ์ปรับตัวของ combination number ในเครือข่าย ซึ่งจากเครือข่ายที่กำหนด ได้ทำการจำลองอัลกอริทึมทั้งหมด 3 ครั้ง ครั้งแรกเป็นการจำลองการทำงานของอัลกอริทึม Bolot ครั้งที่สองเป็นการจำลองการทำงานของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ Single Exponential Smoothing (SES) ครั้งที่สามเป็นการจำลองการทำงานของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ Least Square ผลการทดลองแสดงดังกราฟรูปที่ 4.2, 4.3 และ 4.4 ตามลำดับ

เมื่อพิจารณารูปที่ 4.2 ด้านล่างที่แสดงร้อยละของข้อมูลที่สูญหายก่อนและหลังการจัดเรียงคั่งเส้นกราฟ จะเห็นได้ว่า ค่า combination number ที่ทำให้ค่า L_u ไม่เกินค่าเทรสโฮลด์ HIGH จะมีการปรับตัวเมื่อวินาทีที่ 100 จาก combination number เท่ากับ 0 ปรับตัวไปหยุดอยู่ที่ combination number เท่ากับ 5 จนทำให้ค่า L_u ไม่เกินค่าเทรสโฮลด์ HIGH จึงเริ่มมีการเปลี่ยนแปลงค่า Combination number และค่าต่ำสุดอยู่ที่ Combination number เท่ากับ 4

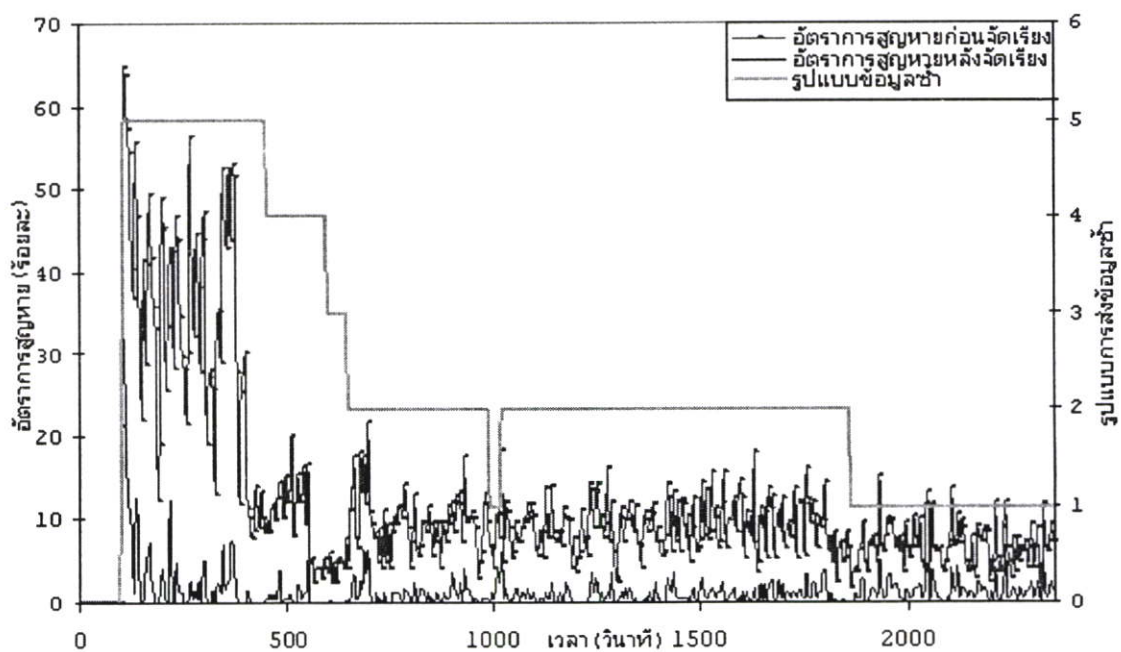
เมื่อพิจารณารูปที่ 4.3 แสดงให้เห็นการปรับตัวของค่า Combination number เป็นไปตามความหนาแน่นของปริมาณกราฟฟิกในเครือข่าย อัลกอริทึม CNR มีการปรับค่า Combination number จากค่า 0 ไปเป็นค่า 5 แล้วหลังวินาทีที่ 450 จะเริ่มลดลงจนไปหยุดที่ Combination number เท่ากับ 2 จากนั้นได้มีการปรับตัวลดลงเป็น Combination number เท่ากับ 1 เมื่ออัลกอริทึมคำนวณแล้วว่าค่า L_u มีเทรสโฮลด์ไม่เกินค่าเทรสโฮลด์ HIGH

และเมื่อพิจารณารูปที่ 4.4 ซึ่งแสดงการปรับตัวของ Combination number ที่มีการเปลี่ยนแปลงตามค่า L_u , L_b และค่า reward ที่มีการหาแนวโน้มของค่า reward โดยใช้หลักการของ Least Square Method จะเห็นว่า Combination number เริ่มปรับค่าจากค่า 0 เป็น ไปหยุดที่ Combination number เท่ากับ 5 เมื่อมีปริมาณกราฟฟิกในเครือข่ายหนาแน่นในช่วงวินาทีที่ 100 ถึงวินาทีที่ 400 จากนั้นอัลกอริทึมจะเริ่มคำนวณค่า combination number ให้มีการปรับตัวลดลงมาหยุด

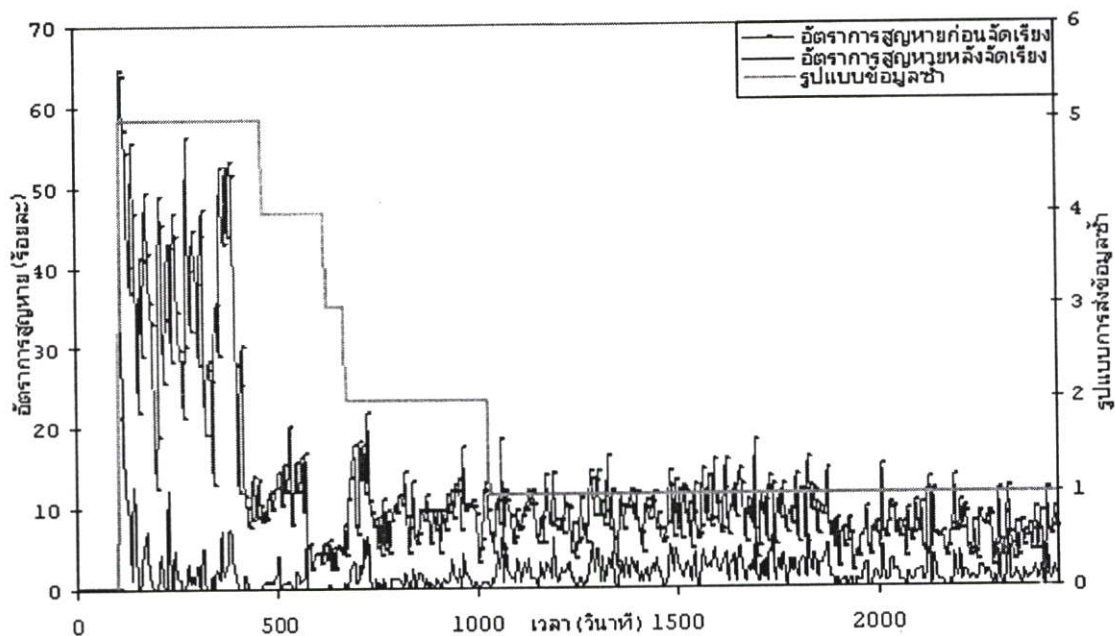
ที่ Combination number เท่ากับ 1 โดยค่าร้อยละของการสูญหายหลังการจัดเรียงแพ็กเก็ตเกิดขึ้นอยู่ที่ระดับที่ยอมรับได้ นั่นคือ L_u ไม่เกิน 5%



รูปที่ 4.2 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม Bolot



รูปที่ 4.3 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ SES ในการหาค่า reward

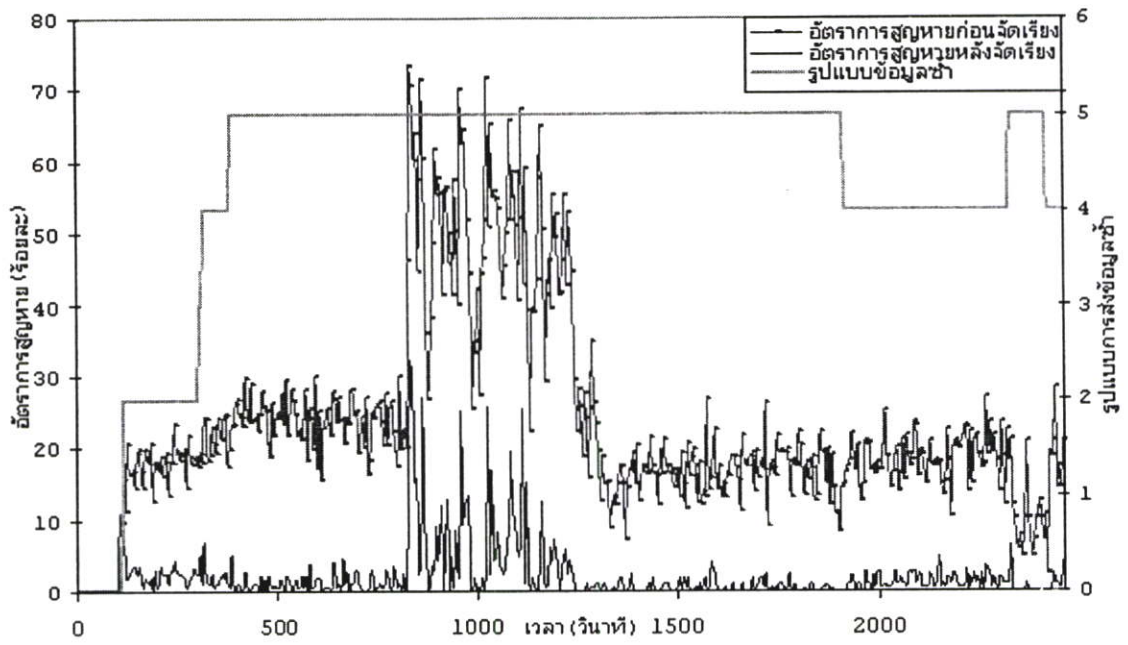


รูปที่ 4.4 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการทำงานของการจัดเรียงของอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการหาค่า reward

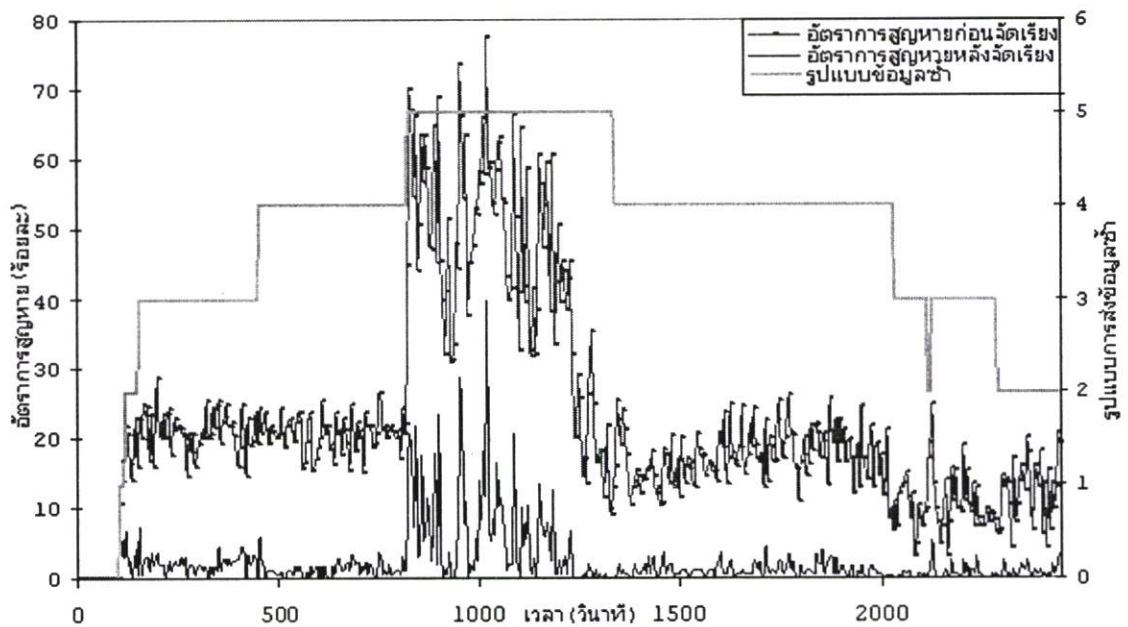
4.2.2 ผลการทดลองตอนที่ 2

การทดลองตอนที่ 2 ได้มีการเปลี่ยนแปลงที่โหนด TS_{11} - TS_{1n} และ TS_{s1} - TS_{sn} ซึ่งจะเริ่มส่งและหยุดส่งข้อมูลพร้อมกันโดยเริ่มส่งในวินาทีที่ 800 และหยุดส่งที่วินาทีที่ 1200 เพื่อสังเกตการปรับตัวของอัลกอริทึม โดยให้มีความหนาแน่นของกราฟฟิกสูงช่วงกลาง ของเวลาที่ทำการจำลองทั้งหมด และในการทดลองนี้ก็มีการรันแบบจำลอง 2 ครั้งต่อหนึ่งอัลกอริทึมแตกต่างกันในตอนที่ 1 โดยครั้งแรกใช้จำนวนโหนด TCP เท่ากับ 20 โหนด และครั้งที่สองใช้จำนวนโหนด TCP เป็น 40 โหนดเพื่อสังเกตการณ์ปรับตัวเสมือนเพิ่มผู้ส่งและผู้รับขึ้นในเครือข่าย

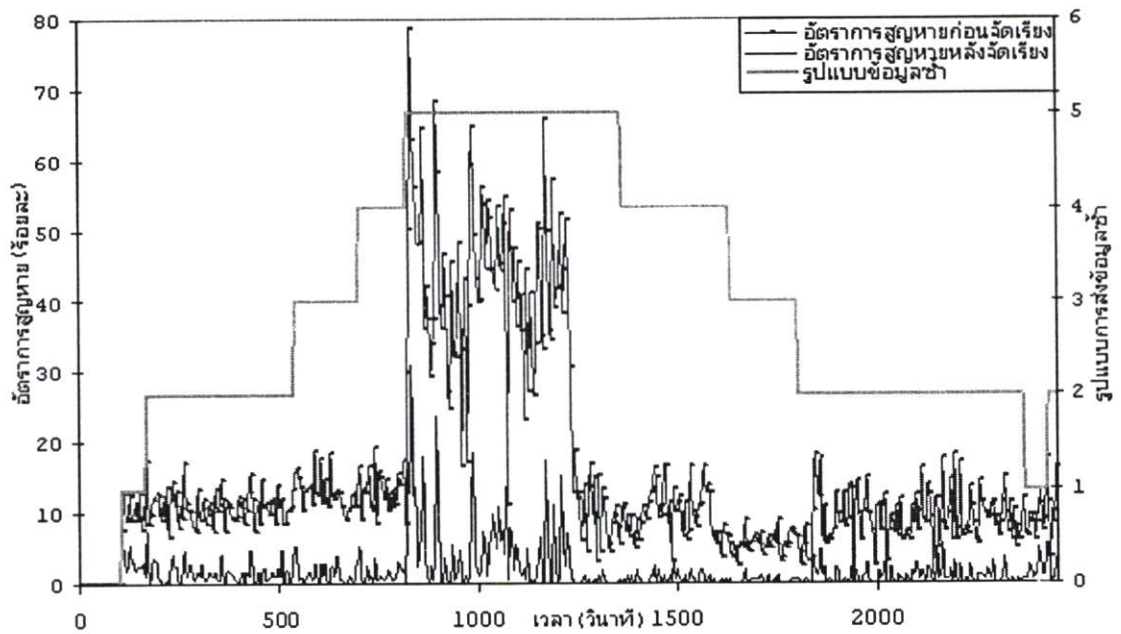
ช่วงแรกทำการจำลองโดยใช้จำนวนโหนด TCP เท่ากับ 20 โหนด ทำการจำลองอัลกอริทึมทั้งหมด 3 ครั้ง ครั้งแรกเป็นการจำลองการทำงานของการจัดเรียงของอัลกอริทึม Bolot ครั้งที่สองเป็นการจำลองการทำงานของการจัดเรียงของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ SES ครั้งที่สามเป็นการจำลองการทำงานของการจัดเรียงของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ Least Square ซึ่งผลการทดลองแสดงดังกราฟในรูปที่ 4.5, 4.6 และ 4.7 ตามลำดับ



รูปที่ 4.5 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการทำงานของอัลกอริทึม Bolot (TCP 20 โหนด)

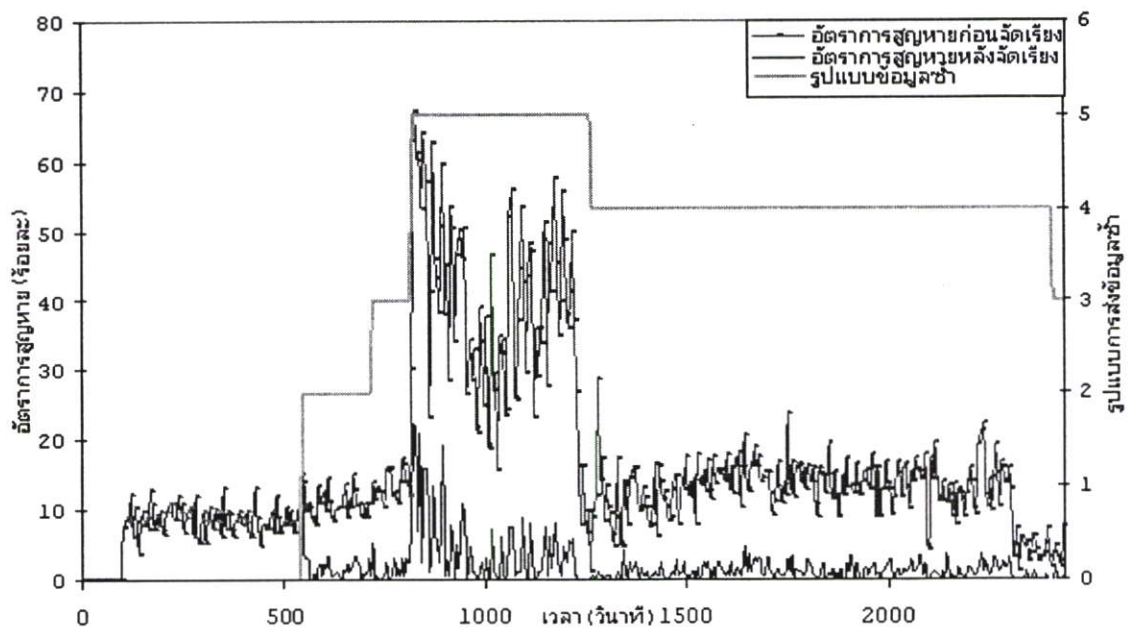


รูปที่ 4.6 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ SES ในการหาค่า reward (TCP 20 โหนด)

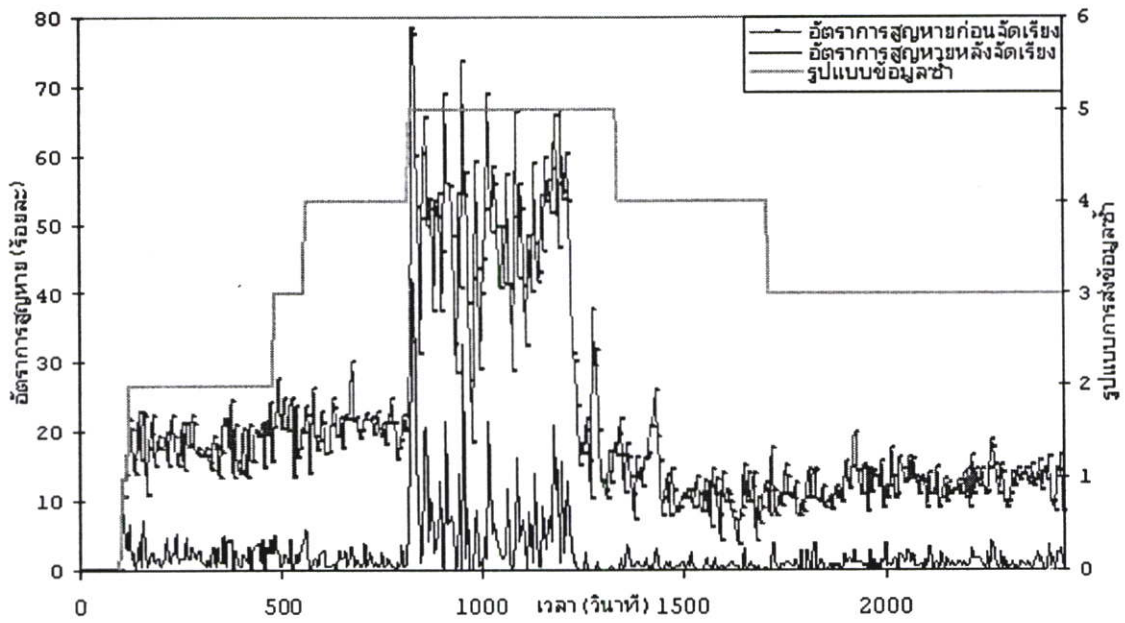


รูปที่ 4.7 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการหาค่า reward (TCP 20 โหนด)

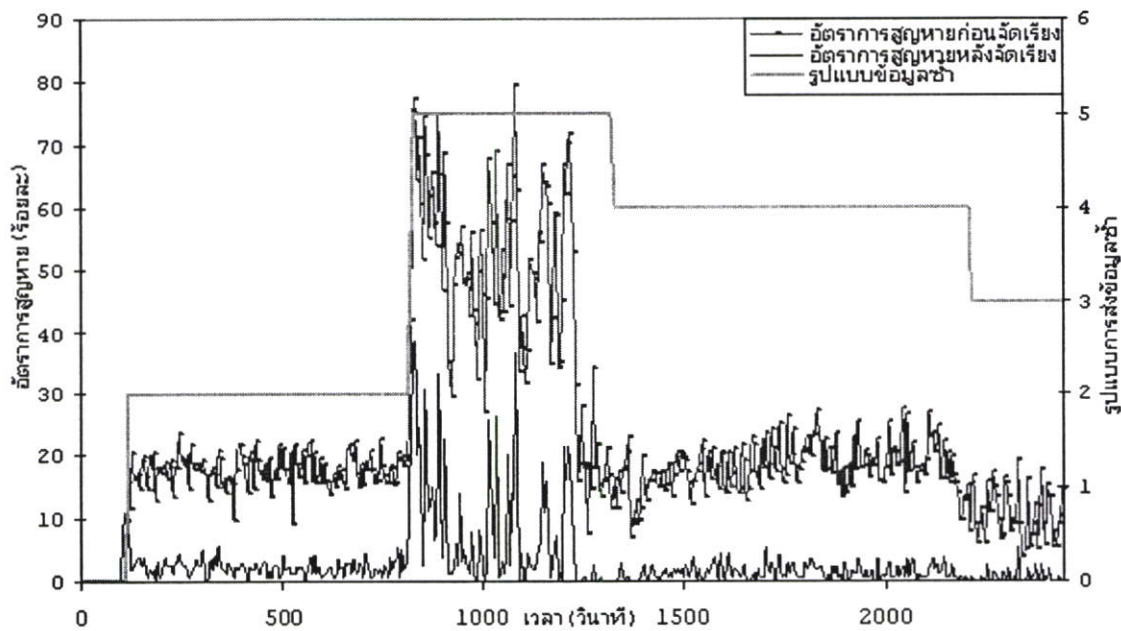
ช่วงที่สองทำการจำลองโดยใช้จำนวนโหนด TCP เท่ากับ 40 โหนด ทำการจำลองอัลกอริทึมทั้งหมด 3 ครั้ง ครั้งแรกเป็นการจำลองการทำงานของอัลกอริทึม Bolot ครั้งที่สองเป็นการจำลองการทำงานของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ SES ครั้งที่สามเป็นการจำลองการทำงานของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ Least Square ผลการทดลองแสดงดังกราฟในรูปที่ 4.8, 4.9 และ 4.10 ตามลำดับ



รูปที่ 4.8 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม Bolot

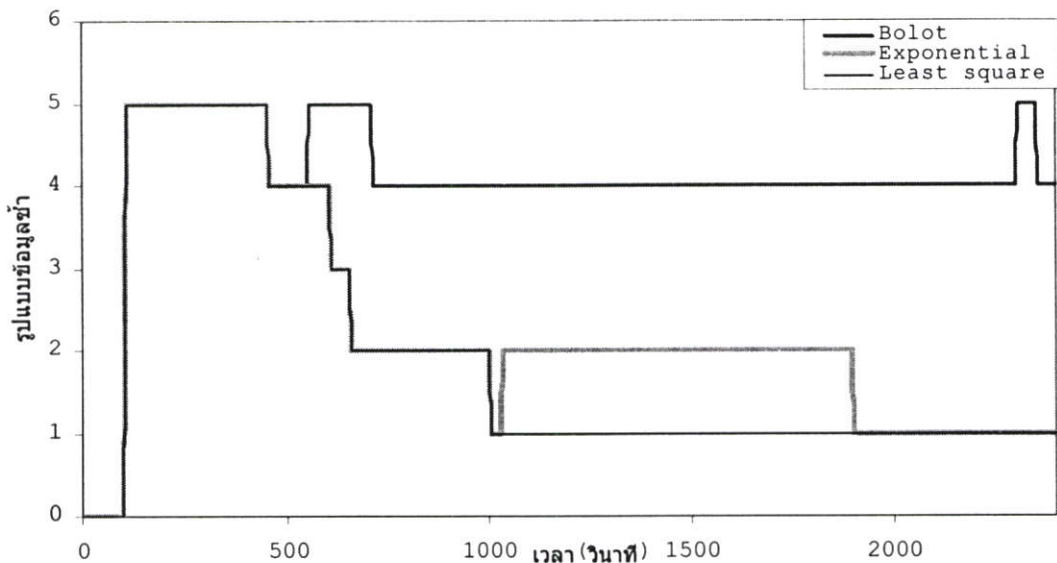


รูปที่ 4.9 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ SES ในการหาค่า reward (TCP 40 โหนด)



รูปที่ 4.10 แสดงค่า L_u และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการหาค่า reward (TCP 40 โหนด)

จะเห็นได้ว่า ในรูปกราฟที่ทำการจำลองโดยใช้โหนด TCP ส่งข้อมูลในเครือข่ายที่มีจำนวนโหนด 20 และ 40 โหนดในสองช่วงเวลา จะให้แนวโน้มการปรับตัวของ Combination number ที่เหมือนกันทั้งสองอัลกอริทึมคือ เมื่อปริมาณทราฟฟิกในเครือข่ายมีจำนวนมาก ค่า combination number จะปรับค่าเป็นค่าสูง และเมื่อปริมาณทราฟฟิกในเครือข่ายเริ่มน้อยลงซึ่งจะมีปริมาณการสูญหายของข้อมูลน้อยลงนั้น ค่า Combination number ก็จะปรับตัวลดลงด้วย



รูปที่ 4.11 เปรียบเทียบการเปลี่ยนแปลง combination number ของอัลกอริทึม Bolot และอัลกอริทึม CNR ซึ่งใช้เทคนิค SES และ Least square method ในการหาค่า reward

จากรูปที่ 4.11 แสดงความสัมพันธ์ของค่า Combination number ที่มีการเปลี่ยนแปลงตามเวลาและขึ้นอยู่กับ การเปลี่ยนแปลงของค่า reward จากรูปได้เปรียบเทียบกันระหว่างอัลกอริทึม Bolot และอัลกอริทึม CNR ในอัลกอริทึม Bolot การเปลี่ยนแปลงของค่า Reward เปลี่ยนแปลงโดยใช้ Lb/La เป็นตัวแปรหลักทำให้ได้การปรับตัวของ Combination number ตามเส้นกราฟ ส่วนในอัลกอริทึม CNR ได้ทดลองหาค่า reward โดยใช้วิธีการของ Single Exponential Smoothing (SES) ได้การเปลี่ยนแปลงของ Combination number ดังเส้นกราฟ และทดลองหาค่า reward โดยใช้วิธีการของ Least square จะได้การเปลี่ยนแปลงของ Combination number ดังเส้นกราฟตามลำดับ

ซึ่งจะเห็นว่าในการเปลี่ยนแปลงของค่า combination number เมื่อใช้วิธีการของ Least Square ทำให้การปรับตัวของ Combination number มีค่าต่ำซึ่งหมายความว่า มีการใช้รูปแบบและปริมาณข้อมูลที่น้อยกว่า วิธีการของ Bolot และวิธีการ SES ที่ใช้ในอัลกอริทึม CNR เหมือนกัน

จึงสรุปได้ว่า ในอัลกอริทึมที่มีการปรับตัว CNR เมื่อใช้เทคนิค Least square method มาประยุกต์ใช้ เพื่อประมาณหาค่า reward และนำไปใช้ในการกำหนดรูปแบบของปริมาณข้อมูลซ้ำ (Combination) นั้น วิธีนี้จะลดปริมาณของข้อมูลซ้ำ โดยยังคงสามารถกู้คืนแพ็กเก็ตข้อมูลที่สูญหายได้ และคุณภาพของเสียงอยู่ในระดับที่ยอมรับได้ ซึ่งการใช้ปริมาณข้อมูลซ้ำต่ำนั้นจะทำให้สามารถประหยัดการใช้แบนด์วิดท์และเป็นการใช้ทรัพยากรของระบบอย่างมีประสิทธิภาพ

4.3 การวิเคราะห์

เมื่อจำลองการทำงานตามการทดลองตอนที่ 1 โดยค่าพารามิเตอร์ที่วัดได้จากการส่งผ่านแพ็กเก็ตเสียงในเส้นทางที่อยู่ระหว่าง โหนด N2 กับ โหนด N3 ในการวัดจำนวนแพ็กเก็ตที่ส่งไปยัง โหนดผู้รับ(Received Packets), ค่าเฉลี่ยทราฟฟิค(Average Throughput (kbps)) และ ค่าเฉลี่ยเวลาหน่วง(Average Delay (ms)) ต้องระบุค่าขนาดของแพ็กเก็ตในที่นี่คือ 24 ไบต์, ตำแหน่งปลายทางและต้นทางของโหนดที่ทำการวัดค่า รวมทั้งชื่อไฟล์ที่เป็น tracefiles ของระบบที่ทำการจำลองขึ้นสามารถแสดงได้ดังตารางที่ 4.2 ซึ่งเป็นค่าพารามิเตอร์ที่วัดได้จาก Trace file

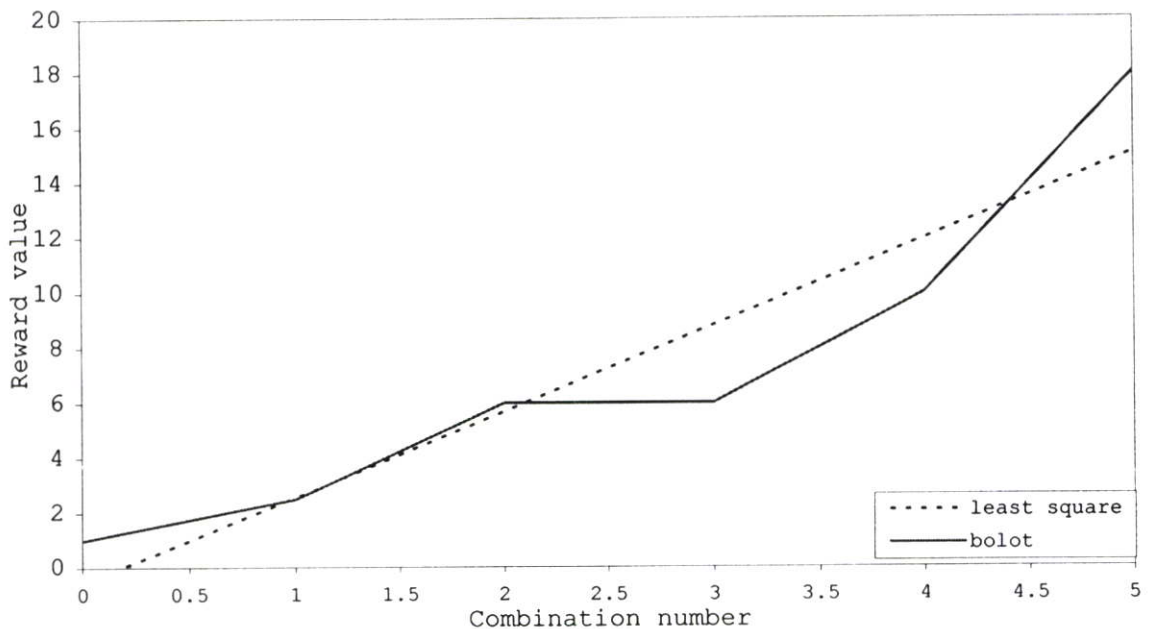
ตารางที่ 4.2 แสดงค่าพารามิเตอร์ที่สำคัญจาก Trace file เปรียบเทียบ 3 วิธีการ

| พารามิเตอร์ | Least Square | SES | Bolot |
|---------------------------|--------------|---------|---------|
| Received Packets | 545,780 | 545,695 | 544,445 |
| Average Throughput (kbps) | 1814.84 | 1809.93 | 1804.27 |
| Average Delay (ms) | 140.112 | 147.289 | 159.341 |

เมื่อพิจารณาจากตารางที่ 4.2 จะเห็นว่า การจำลองอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการคำนวณค่า reward ที่ใช้ในการตัดสินใจว่าจะใช้ combination number ค่าไหนในช่วงการส่งครั้งถัดไป สามารถทำให้ระบบการส่งข้อมูลเสียงในขณะนั้น มีประสิทธิภาพในการส่งดีกว่าการใช้วิธีการ SES รวมถึงเมื่อเปรียบเทียบกับอัลกอริทึมของ Bolot โดยจะเห็นได้ว่ามีจำนวนแพ็กเก็ตที่ส่งไปยังโหนดผู้รับ, ค่าเฉลี่ยทราฟฟิค (Average Throughput) ที่เพิ่มขึ้น ซึ่งเป็นค่าที่ดีกว่าระบบเดิมที่ใช้อัลกอริทึม Bolot ในการตัดสินใจเลือกค่า combination number เนื่องจากทำให้ผู้รับข้อมูลเสียง สามารถได้รับเสียงที่มีความชัดเจนและต่อเนื่อง ส่วนค่าเฉลี่ยเวลาหน่วงมีแนวโน้มที่น้อยกว่าการใช้วิธีการ SES และอัลกอริทึม Bolot แสดงให้เห็นว่า แพ็กเก็ตเสียงมีความล่าช้าในการส่งน้อยลง จะทำให้ประสิทธิภาพการส่งโดยรวมของระบบดีขึ้น เนื่องจากสามารถสื่อสารได้เข้าใจไม่ติดขัด

จากกราฟในรูปที่ 4.12 ด้านล่างเป็นการเปรียบเทียบความสัมพันธ์ของค่า Reward กับค่า Combination number ที่เปลี่ยนแปลงตามค่า Reward เส้นกราฟที่เป็นเส้นประ เป็นค่าที่ได้จากห้องวิจัยของ Bolot ซึ่งเป็นค่าที่กำหนดให้อัลกอริทึมในการปรับตัวใช้งานตามค่านี้โดยไม่ได้คำนึงถึงการเปลี่ยนแปลงของการส่งข้อมูลในเครือข่ายขณะนั้นๆ ในงานวิจัยนี้จึงได้นำวิธีการของ Least Square และใช้เทคนิคความสัมพันธ์ของข้อมูลในลักษณะ Linear regression มาใช้ในอัลกอริทึมแบบปรับตัวซึ่งคืออัลกอริทึม CNR เพื่อหาแนวโน้มของการเปลี่ยนของค่า Reward ให้เหมาะสมกับเครือข่ายในขณะนั้น

เมื่อพิจารณาการเปลี่ยนแปลงของค่า reward จะเห็นได้ว่ายังคงมีแนวโน้มทิศทางการเปลี่ยนแปลงไปในทิศทางเดียวกับอัลกอริทึม Bolot และเมื่อใช้วิธีการ Least Square ยังสามารถส่งข้อมูลช้ากว่าเดิม โดยประสิทธิภาพการส่งแพ็กเก็ตเสี่ยงโดยรวมของเครือข่ายดีขึ้น



รูปที่ 4.12 แสดงความสัมพันธ์ระหว่างค่า Reward กับค่า Combination number ของวิธีการ Least Square กับวิธีการ Bolot

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ศึกษาวิธีการควบคุมความผิดพลาดแบบปรับตัวซึ่งเป็นวิธีการที่ใช้พื้นฐานของ FEC และประยุกต์นำเทคนิคการประมาณค่าด้วยวิธีกำลังสองน้อยที่สุด (Least Square Method) มาใช้เพื่อให้อัลกอริทึมสามารถทำงานได้ดีขึ้นในสภาพเครือข่ายที่แตกต่างกัน จากการศึกษาอัลกอริทึมของการควบคุมความผิดพลาดแบบปรับตัวที่ได้เสนอก่อนหน้าคือ อัลกอริทึม Bolot และอัลกอริทึม CNR ซึ่งใช้เทคนิคการปรับให้เรียบแบบเอกซ์โพเนนเชียล (Single Exponential Smoothing) พบว่าอัลกอริทึม CNR มีผลต่อรูปแบบการส่งข้อมูลซ้ำดีกว่าอัลกอริทึม Bolot ดังนั้นในวิทยานิพนธ์นี้จึงได้ประยุกต์นำเทคนิคการประมาณค่าด้วยวิธีกำลังสองน้อยที่สุดในอัลกอริทึม CNR จากการศึกษาผลการทดลองระบบบน NS-2 พบว่าเทคนิคการประมาณค่าด้วย Least square method มีผลต่อประสิทธิภาพการทำงานของอัลกอริทึมในสภาพเครือข่ายที่แตกต่างกัน นั่นคือเมื่อใช้เทคนิค Least square method มาประยุกต์ใช้เพื่อประมาณค่า reward และนำไปใช้ในการกำหนดรูปแบบของปริมาณข้อมูลซ้ำ (Combination) นั้น ซึ่งวิธีนี้จะลดปริมาณของข้อมูลซ้ำน้อยกว่าอัลกอริทึม Bolot และอัลกอริทึม CNR ซึ่งใช้เทคนิคการปรับให้เรียบแบบเอกซ์โพเนนเชียล (SES) โดยยังคงสามารถกู้คืนแพ็กเก็ตข้อมูลที่สูญหายได้ และคุณภาพของเสียงอยู่ในระดับที่ยอมรับได้ นั่นคืออัตราความถูกต้องยังอยู่ในเกณฑ์ที่ยอมรับได้ (อัตราการสูญหายของข้อมูลเสียงไม่เกิน 5%) ซึ่งการใช้ปริมาณข้อมูลซ้ำค่านั้นจะทำให้สามารถประหยัดการใช้แบนด์วิธและเป็นการใช้ทรัพยากรของระบบอย่างมีประสิทธิภาพ

รวมทั้งมีค่าพารามิเตอร์ที่สำคัญอื่นๆ โดยจะเห็นได้ว่ามีจำนวนแพ็กเก็ตที่ส่งไปยังโหนดผู้รับ, ค่าเฉลี่ยทรูพุด ที่เพิ่มขึ้น ซึ่งเป็นค่าที่ดีกว่าระบบเดิมที่ใช้อัลกอริทึม Bolot ในการตัดสินใจเลือกค่า combination number เนื่องจากทำให้ผู้รับข้อมูลเสียง สามารถได้รับเสียงที่มีความชัดเจนและต่อเนื่อง ส่วนค่าเฉลี่ยเวลาหน่วงมีแนวโน้มที่น้อยกว่าการใช้วิธีการ SES และอัลกอริทึม Bolot แสดงให้เห็นว่า แพ็กเก็ตเสียงมีความล่าช้าในการส่งน้อยลง จะทำให้ประสิทธิภาพการส่งโดยรวมของระบบดีขึ้น

5.2 ข้อเสนอแนะ

ในวิทยานิพนธ์นี้ไม่ได้จำลองการทำงานบนเครือข่ายจริง จึงมีความเป็นไปได้ว่าอาจมีปัจจัยอื่นทำให้อัตราการสูญหายของข้อมูลเพิ่มขึ้นหรืออาจมีการติดขัดสูงเกินปกติ ดังนั้นจึงต้องมีการทดลองหาความสัมพันธ์ของอัตราการสูญหายของข้อมูลกับตัวค่าพารามิเตอร์อื่น ๆ หรือใช้เทคนิค

วิธีอื่นๆมาใช้ในการประมาณค่าพารามิเตอร์สำคัญ ที่ใช้ในการกำหนดรูปแบบ และปริมาณข้อมูลซ้ำ (Combination) และติดตามอัตราการสูญหายที่อาจเกิดขึ้นในเครือข่ายการใช้งานจริงว่าเมื่อใช้ การควบคุมความผิดพลาดโดยอัลกอริทึม CNR แล้ว ยังทำให้ประสิทธิภาพในการส่งผ่านข้อมูลที่อยู่ใน เกณฑ์ที่ยอมรับได้หรือไม่

เอกสารอ้างอิง

- [1] V. Kumar, M. Korpi and S. Sengoodan, *IP Telephony with H.323: Architectures for Unified Networks and Integrated Services*, USA: John Wiley & Sons, 2001.
- [2] J-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.
- [3] CCITT Recommendation G.711, "Pulse Code Modulation (PCM) of voice frequencies," 1988.
- [4] ITU-T Recommendation G.729, "Coding of speech at 8 kbit/s using Conjugate Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)," 1996.
- [5] 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.
- [6] 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.
- [7] RADCOM Equipment, Inc., "Fine-tuning Voice over Packet services," <http://www.radcom.com/radcom/technlgy/pdf/FineTuningVoiceOverPackeServices.pdf>, Apr. 2, 2005.
- [8] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson "RTP: A transport protocol for real-time applications," RFC 3550, Jul. 2003.
- [9] H. Schulzrine, S. Casner, "RTP profile for audio and video conferences with minimal control," RFC 3551, Jul. 2003.
- [10] C. S. Perkins et al., "RTP Payload for redundant audio data," RFC2198, Sept. 1997.
- [11] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-Driven Layered Multicast," in *Proceedings of ACM SIGCOMM '96*, Stanford, USA, Aug. 1996, pp. 117–130.
- [12] C. Perkins, O. 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.
- [13] 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.

- [14] 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.
- [15] 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 Technology (ECTI) International Conference*, vol.2, Cholburi, Thailand, May. 2005, pp. 697-700.
- [16] P. Moungnoul, K. Klappong S. Junnapiya, "Effect of Loss Control in Adaptive FEC Audio for Transmission," in *ObCom-2006 International Conference on Mobile, Ubiquitous and Pervasive Computing School of Computing Sciences*, vol.1, Vellore, India, 18th -19th December. 2006.
- [17] ISO/IEC 14496-2, "Coding of audio-visual objects – Part 2: Visual," 2001.
- [18] 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.
- [19] S. Jacobs and A. Eleftheriadis, "Real-time dynamic rate shaping and control for Internet video applications," in *Proceedings of Workshop on Multimedia Signal Processing*, Princeton, USA, Jun. 1997, pp. 23–25.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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 '00*, Phoenix, USA, Feb. 2000, pp. 307-313.
- [24] ITU-T Recommendation H.261, "Video codec for audiovisual services at px64 kbits/s," 1996.

- [25] ITU-T Recommendation H.263, "Video coding for low bit rate communications," version 1, 1996, version 2, 1998.
- [26] Bruce L. Bowerman, Richard T. O'Connell, "*Forecasting and time series: an applied approach Edition 3rd*", Belmont, CA : Duxbury Press, 1993.
- [27] Robert G. Brown, "*Smoothing, Forecasting and Prediction of Discrete Time Series.*", Printice-Hall, Inc., 1963.
- [28] 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.
- [29] The Network Simulator, NS-2, <http://www.isi.edu/nsnam/ns/>, Oct. 23, 2005.
- [30] A. A. Giordano, Frank M. Hsu, *Least Square Estimation with Applications to Digital Signal Pprocessing*, John Wiley & Sons, USA, 1989.
- [31] J-C. Bolot, S. Fosse-Parisis and D. Towsely, "Adaptive FEC-Based Error Control for Internet Telephony," Proc. of IEEE INFORCOM, Vol. 3, pp.1453-1460 , 1999.
- [32] P. Moungnoul, "Adaptive FEC in Audio Communication using Smooth Method," Proc. 35th AIC Conference, Saitama, Japan, Doc.144, pp.1-7, 5-9 March 2007.
- [33] รศ.ดร.วิชิต หล่อจีระชุนท์กุล, "การวิเคราะห์ความคลาดเคลื่อนในการพยากรณ์", การประชุมวิชาการสถิติประยุกต์ ครั้งที่ 10, คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์.

ผลงานทางวิชาการที่ได้รับการตีพิมพ์

1. P.MOUNGNOUL,A.KANWIBOON,K.DEJHAN, “Packet Loss Recovery in Media Specific FEC Audio Transmission by Least Square Method,” in *CITA'07 Fifth International Conference on Information Technology in Asia 2007, Social Computing : Engaging Communities*, vol.5, Kuching, Sarawak, Malaysia, July. 2007, pp. 300-302.

PROCEEDINGS OF CITA'07



CITA'07

Fifth International Conference on Information Technology in Asia 2007

Social Computing: Engaging Communities

9th - 12th July 2007
Kuching, Sarawak, Malaysia
www.cita07.org



Organised by:



In Collaboration with:



Information & Communications Technology Unit (ICTU)
Chief Minister's Department of Sarawak



Global Information & Telecommunication Institute

Packet Loss Recovery in Media Specific FEC Audio Transmission by Least Square Method

PHICHET MOUNGNOUL, ARUNEE KANWIBOON and KOBCHAI DEJHAN
 Faculty of Engineering and ReCCIT, King Mongkut's Institute of Technology Ladkrabang,
 Chalongkrung Road, Bangkok 10520
 THAILAND
 phichet@telecom.kmitl.ac.th, kobchai@telecom.kmitl.ac.th

Abstract - This paper proposes a packet loss recovery technique in audio stream at a sender side. The adaptive error control in the Internet telephony system uses this packet recovery scheme based on the media specific Forward Error Correction (FEC). By adapting with the least square method used recovery algorithm with the best fitting curve of a reward value. This technique can estimate the trend of outcomes, and improve the adjustment of redundant packets of the recovery algorithm. The results of experiments show that the adaptive error control algorithm can decrease the packet loss rate.

Keywords: Internet telephony, packet loss recovery, least square method

1 Introduction

The current internet normally uses the best-effort service model. This cannot guarantee the quality of service (QoS). As a result the transmission of audio traffic over the internet is always affected by the packet loss problem. The FEC (Forward Error Correction) is one of the error recovery mechanisms that can reduce the effect of packet loss with the low latency [2]. FEC technique adds some redundancy that increases the usage bandwidth. The adaptive FEC-based algorithm which can set the quantity of audio redundancy based on the number of packet lost, therefore, is required. [4]

There are several proposed algorithms that can do this task. Some algorithms have to use a loss report forecast or estimate the reward parameters for the next interval time [2]. The least square method is very popular technique employed in computing estimation parameters and data curve fitting. This method can be used in part of increase or decrease combination number (pattern of redundant data in packet) for the adaptive FEC-based algorithm. In addition, the better forecasting can be performed because the result of this forecasting will be used in the optimization of the recovery mechanism.

The SES (single exponential smoothing) technique is a simple and pragmatic approach to forecasting, whereby the forecast is constructed from the exponentially weighted average of pass observation [6]. If the packet loss prediction use bounded adaptive smoothing constant from the forecast loss rate in the next interval.

This paper studies the performance of the media specific FEC-based algorithm by increasing the number of combination data and comparing the performances. The adaptive FEC control algorithm developed in [5] CNR (Centre for Network Research) algorithm is studied, and the reward value of this algorithm is forecasted using the least square and exponential method. Process of quantitatively is estimated and the trend of the outcomes known as *regression* or curve fitting becomes necessary [1]. The develop mechanism attempts to eliminate or to minimize the impact of packet loss and end-to-end delay on the quality of the delivered audio to the destinations. The effect of the variability parameters that use in process of each algorithm for improve audio transmission in the future is employed.

2 Packet Recovery Algorithm by Least Square Method

2.1 CNR Algorithm

CNR algorithm [6] is a media specific FEC-based algorithm. This algorithm improved from Bolot [4] algorithm by increase the combination number of audio redundancy in each period. G.723.1 codec is the most widely-use in IP telephony system because it has low bit rate enough (6.3 kbps), this can reduce waste time from Bolot algorithm.

CNR algorithm requires RTP (Real-time Transport Protocol) for send audio data from the sender side and RTCP (Real-time Transport Control Protocol) use for report the loss rate (packet loss rate before and after the reconstruction) of the received data packet to the sender [3]. The redundant packet used for recovery the lost packets and the better combination of codec for higher the recovery rate. By used Bolot formulation to calculate a reward value of each combination data as equation.

$$\text{reward} = \frac{\text{packet recovery rate before reconstruction}}{\text{packet recovery rate after reconstruction}} \quad (1)$$

RTCP packet is received in sender side, the reward value (L_b/L_a) of current combination R_c to calculate the loss information as follow.

$$R_c = c_0 + c_1(L_b / L_a) \quad (2)$$

Where R_c is the reward value of current combination, c_0 and c_1 are coefficients to be determined the least squares error solution.

2.2 The Least Square Method

The least square concept presented here will continue to develop for estimation, prediction or numerical analysis the reward value in CNR algorithm. Least square method with Hilbert space [1] can predict the optimized of reward value for system in Figure 1. From Figure 1, w represented the input data and z represented an output, the relationship between w and z as the transformation $z = f(w)$. Discrete input data is represented by w_i , the output result is $z_i = f(w_i)$, $i = 1, \dots, m$. Linear regression method is sloved this problem by a linear function $f_a(w)$, as follow.

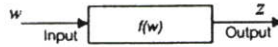


Figure 1. System model of linear regression method.

$$f_a(w) = c_0 + c_1 w \quad (3)$$

Let the individual deviations δ_i , defined as

$$\delta_i = f(w_i) - f_a(w_i) \quad i = 1, \dots, m \quad (4)$$

The solution problem by least square method is obtained to choose the coefficients c_0 and c_1 with minimize S .

$$S = \sum_{i=1}^m \delta_i^2 \quad (4)$$

$$= \sum_{i=1}^m [f(w_i) - f_a(w_i)]^2 \quad (5)$$

$$= \sum_{i=1}^m [z_i - (c_0 + c_1 w_i)]^2 \quad (6)$$

S is the sum of the individual deviation square, $f(w_i)$ is collected data and $f_a(w_i)$ is trend of reward data. The 2nd derivatives of S , as follow.

$$\frac{\partial S}{\partial c_0} = \sum_{i=1}^m 2[z_i - (c_0 + c_1 w_i)](-1) = 0 \quad (7)$$

$$\frac{\partial S}{\partial c_1} = \sum_{i=1}^m 2[z_i - (c_0 + c_1 w_i)](-w_i) = 0 \quad (8)$$

Rewrite (6) as follow.

$$\sum_{i=1}^m z_i = m c_0 + \left(\sum_{i=1}^m w_i \right) c_1 \quad (9)$$

$$\sum_{i=1}^m w_i z_i = \left(\sum_{i=1}^m w_i \right) c_0 + \left(\sum_{i=1}^m w_i^2 \right) c_1 \quad (10)$$

and

$$c_0 = \frac{\left(\sum_{i=1}^m z_i \right) \left(\sum_{i=1}^m w_i^2 \right) - \left(\sum_{i=1}^m w_i \right) \left(\sum_{i=1}^m w_i z_i \right)}{\Delta} \quad (11)$$

$$c_1 = \frac{m \sum_{i=1}^m w_i z_i - \left(\sum_{i=1}^m w_i \right) \left(\sum_{i=1}^m z_i \right)}{\Delta} \quad (12)$$

$$\Delta = m \sum_{i=1}^m w_i^2 - \left(\sum_{i=1}^m w_i \right)^2 \quad (13)$$

2.3 The Proposed Method

Base on the CNR algorithm used the least square method for calculates the reward value in the next time interval. The reward value in the previous time interval used to decision the optimize combination number.

The reward value and combination number can predict by linear regression formula. Table 1 show the reward value in difference combination of Bolot.

Table 1. Reward value of difference combinations type

| No. | Combination | Bolot |
|-----|-------------|-------|
| 0 | - | 1 |
| 1 | 1- | 2.5 |
| 2 | -2 | 6 |
| 3 | -1-2 | 6 |
| 4 | -1-3 | 10 |
| 5 | -1-2-3 | 18 |

3 Experiment Setup and result

This simulation setup with CNR algorithm uses NS-2 program for network simulator.

The network topology used for simulation model as shown in figure 2, all of links are drop tail. Most of them have 10 Mbps capacity and 1ms delay except the bottleneck link between N1 and N2 has 2 Mbps capacity and 50 ms delay.

The audio packets was sent from the sender (node RS) to the receiver (node RR) via the RTP in every 30 milliseconds. Node RR sends the receiver report packet in every 5 seconds to RS and another node sent the TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) traffics.

From Figure 2, all average value calculated from source node N1 and destination node N2. The packet size was 24 bytes and audio traffic started for send a packet data at 0.1 second and stopped at 2400 seconds. Table 2 shows result data from NS-2 trace files in network. This data determined the performance of the recovery algorithm. The number period that $L_a >$ high threshold of least square method has lower than Bolot. It can increase number of data recovery and received packet data.

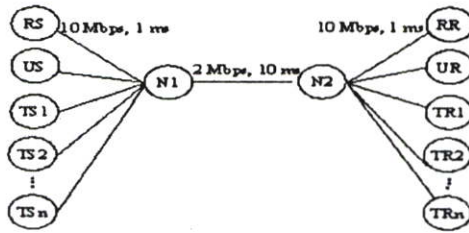


Figure 2. Topology model of the simulation

Then the performance is increased by the number of average throughput and average of delay is decreased.

| Parameters | Least Square | Bolot |
|---|--------------|-----------|
| No. period that $L_a > \text{high threshold}$ | 13 | 17 |
| Received Packets | 1,056,000 | 1,052,190 |
| Average Throughput (kbps) | 1,950.67 | 1,950.46 |
| Average Delay (ms) | 62.56 | 70.14 |

Table 2 Simulation results from NS-2 trace file.

From (2) can plot graph of difference combination scheme and compare with the other the results [4][5][6] shows in Figure 3. The least square method is better than Bolot and exponential smoothing method.

Figure 4 and 5 shows loss rate and combination adjustment by Bolot and least square method, respectively. The adaptive algorithm of CNR can control the loss rate in the next time interval and reduce the number of loss rate after reconstruction as L_a more than the HIGH threshold.

4 Conclusions

In this paper, proposes a reward value forecasting by the least square method to perform the better forecasting and solve the drawback of the media specific FEC algorithm in part of increasing combination number. The experimental result shows the better performance of loss rate and important parameters in network. The recovery algorithm at sender side can send the appropriate redundant packets to audio receiver.

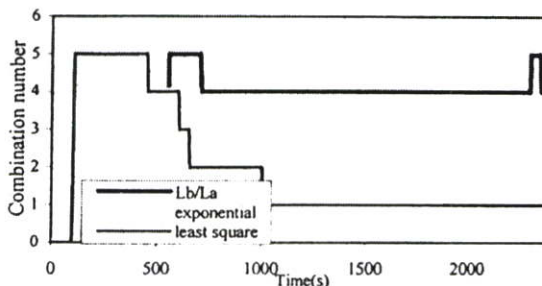
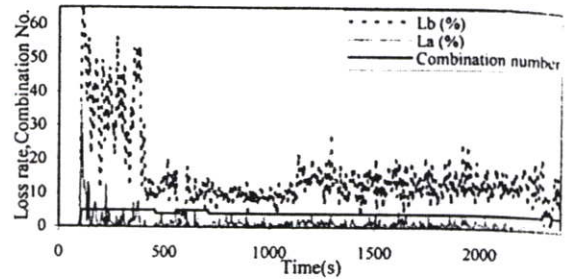
Figure 3. Comparative combination adjustment with different formula of R_c 

Figure 4. Loss rate of Bolot method

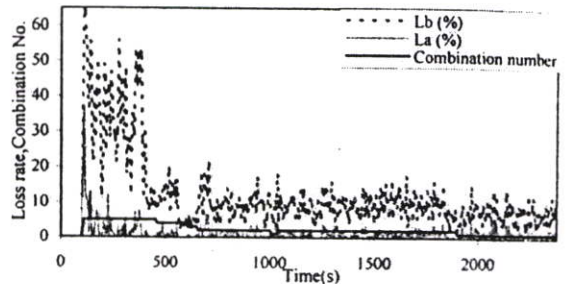


Figure 5. Loss rate of least square method

References

- [1] A. A. Giordano, Frank M. Hsu, *Least Square Estimation with Applications to Digital Signal Processing*, John Wiley & Sons, USA, 1989.
- [2] C. Perkins, O. Hodson, and V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," *IEEE Network*, Vol. 12, No. 5, pp. 40 - 48, September-October 1998.
- [3] 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.
- [4] J-C. Bolot, S. Fosse-Parisis and D. Towsely, "Adaptive FEC-Based Error Control for Internet Telephony," Proc. of IEEE INFORCOM, Vol. 3, pp.1453-1460, 1999.
- [5] S.Chuaywong,S. Kamolphiwong,T. Kamolphiwong, "Performance Enhancement of FEC Mechanism for IP Telephony," Proceeding of the 2005ECTI, vol.2, Cholburi, Thailand, pp. 697-700, May-2005.
- [6] P. Moungnoul, "Adaptive FEC in Audio Communication using Smooth Method," Proc. 35th AIC Conference, Saitama, Japan, Doc.144, pp.1-7, 5-9 March 2007.

ประวัติผู้เขียน

นางสาวอรุณี กาลวิบูลย์ เกิดเมื่อวันที่ 25 มกราคม 2519 ที่จังหวัดอุดรธานี ปีการศึกษา 2542 สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยขอนแก่น จังหวัดขอนแก่น