บทที่ 4
ผลการทดลองและการวิเคราะห์

จากการศึกษาวิธีการควบความผิดพลาดแบบปรับตัว โดยมีการนำเทคนิคการประมาณค่าด้วยวิธีกำลังสองน้อยที่สุด มาใช้ในการวิเคราะห์อนุกรมเวลาที่มีข้อมูลผิดปกติในการควบคุมความผิดพลาดเสียงตามวิธีการที่ได้เสนอขึ้นในบทที่ 3 ในบทนี้จะทำการประเมินผลอัลกอริทึม CNR โดยมีการเปรียบเทียบกับการใช้วิธีการปรับให้เรียบ (Single Exponetial 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 และเนื่องจากว่าแต่ละอัลกอริทึมต่างก็มีพิจารณาค่าเทรสโฮลด์ HIGH และ LOW ในการเลือก Combination ดังนั้นในการเปรียบเทียบสมรรถนะของแต่ละอัลกอริทึมนั้น จำเป็นจะต้องกำหนดค่าเทรสโฮลด์ให้เท่ากัน ในงานวิจัยก่อนหน้าได้ระบุเอาไว้ว่าหากมีข้อมูลเสียงสูญหายมากกว่า 5 % จะทำให้รู้สึกได้ว่าเสียงมีคุณภาพลดลง ดังนั้นการทดลองนี้จึงเลือกใช้ที่จะใช้ค่าเทรสโฮลด์ HIGH เท่ากับ 5% และเพื่อให้ผลการทดลองมีความน่าเชื่อถือมากขึ้น ในการทดลองนี้จึงได้มีการจำลองในกรณีที่ใช้ค่าเทรสโฮลด์ HIGH เท่ากับ 3% ด้วย ซึ่งกรณีนี้ก็อาจจะเป็นไปได้ในการใช้งานจริง เช่น ผู้ใช้ต้องการคุณภาพเสียงที่ดีมากโดยยอมรับการสูญหายของข้อมูลเสียงได้ไม่เกิน 3% เป็นต้น ส่วนค่าเทรสโฮลด์อีกค่าหนึ่งคือค่าเทรสโฮลด์ LOW กำหนดให้มีค่าเท่ากับ 1% ซึ่งค่าของเทรสโฮลด์ LOW นี้ไม่ควรกำหนดให้มีค่าใกล้เคียงกับค่าเทรสโฮลด์ HIGH มากนัก เพื่อป้องกันไม่ให้เกิดการเปลี่ยน Combination บ่อยเกินไป

ในการทดลองนี้ไม่ได้มีการส่งข้อมูลในเครือข่ายจริง แต่เป็นการจำลองบน NS-2 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 นอกจากนี้ยังมีโหนดที่คอยส่งทราฟฟิกชนิดอื่นๆ 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

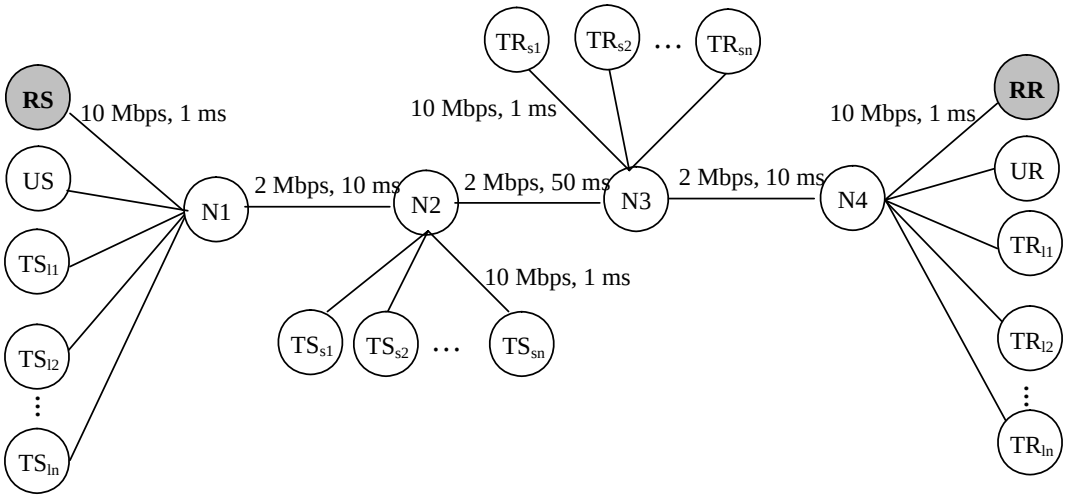
ในการทดลองตอนที่ 1 เป็นการจำลองให้มีการสื่อสารเสียงในเครือข่ายที่มีผู้ที่ส่งทราฟฟิก TCP เป็นช่วงเวลานาน และมีปริมาณการใช้แบนด์วิดท์เต็มความจุของเครือข่าย ซึ่งในสภาพเครือข่ายเช่นนี้สามารถทำให้เกิดการแกว่งของค่าอัตราการสูญหายของแพ็กเก็ตในการสื่อสารเสียงได้ โดยมีจุดประสงค์เพื่อประเมินสมรรถนะของแต่ละอัลกอริทึมว่าจะได้รับผลจากปัญหาการแกว่งของค่าอัตราการสูญหายของแพ็กเก็ตมากน้อยเพียงใด และในกรณีที่สภาพเครือข่ายเป็นเช่นนี้อัลกอริทึมใดสามารถทำงานได้ผลดีที่สุด ซึ่งการส่งข้อมูลของแต่ละโหนดเป็นดังนี้

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

Combination ที่ใช้ ถ้า Combination ใดมีจำนวนข้อมูลซ้ำมาก แพ็กเก็ตเสียงที่ส่งก็จะมีขนาดใหญ่

- โหนด RR ส่งแพ็กเก็ต Receiver Report ให้กับโหนด RS และเนื่องจากในที่นี้เป็นการติดต่อแบบจุดต่อจุด ดังนั้นแพ็กเก็ต Receiver Report นี้จะมีการส่งทุกๆ 5 วินาที
- โหนด US ส่งทราฟฟิก UDP ด้วยอัตราบิต 400 กิโลบิตต่อวินาที โดยใช้แพ็กเก็ตขนาด 1024 ไบต์ ซึ่งโหนด US ทำหน้าที่เป็นตัวแทนของโปรแกรมประยุกต์ในการส่งเสียงหรือวีดิทัศน์ที่ไม่มีการปรับตัวตามสภาพเครือข่าย
- โหนด TS_{l1} – TS_{ln} และ TS_{s1} - TS_{sn} ส่งไฟล์โดยใช้โปรโตคอล FTP ไปให้กับโหนด TR_{l1} – TR_{ln} และ TR_{s1} - TR_{sn} ตาลำดับ โดยเริ่มส่งที่วินาทีที่ 100 และหยุดส่งเมื่อการจำลองยุติ ดังนั้นในช่วง 100 วินาทีแรกจะไม่มีแพ็กเก็ตเสียงสูญหายเลย
- การจำลองยุติที่วินาทีที่ 2400

โดยในการทดลองตอนนี้กำหนดให้การจำลองในแต่ละอัลกอริทึมมีการรันแบบจำลอง 1 ครั้ง ที่โหนด TS_{l1} - TS_{ln} และ TS_{s1} - TS_{sn} เริ่มส่งและหยุดส่งข้อมูลพร้อมกันในวินาทีที่ 100 วินาที และหยุดส่งที่วินาทีที่ 400 เนื่องจากว่าต้องการประเมินสมรรถนะของแต่ละอัลกอริทึมในสภาพแวดล้อมที่ระดับความคับคั่งของเครือข่ายไม่คงที่ และสังเกตผลการทดลองว่าจะเป็นไปในทำนองเดียวกันหรือไม่



รูปที่ 4.1 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง

4.1.2 การทดลองตอนที่ 2

ในการทดลองตอนที่ 2 นี้ ลักษณะการส่งข้อมูลของโหนด RS และ US ยังเหมือนกับในการทดลองตอนที่ 1 แต่มีการเปลี่ยนแปลงที่โหนด TS_{l1} - TS_{ln} และ TS_{s1} - TS_{sn} 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

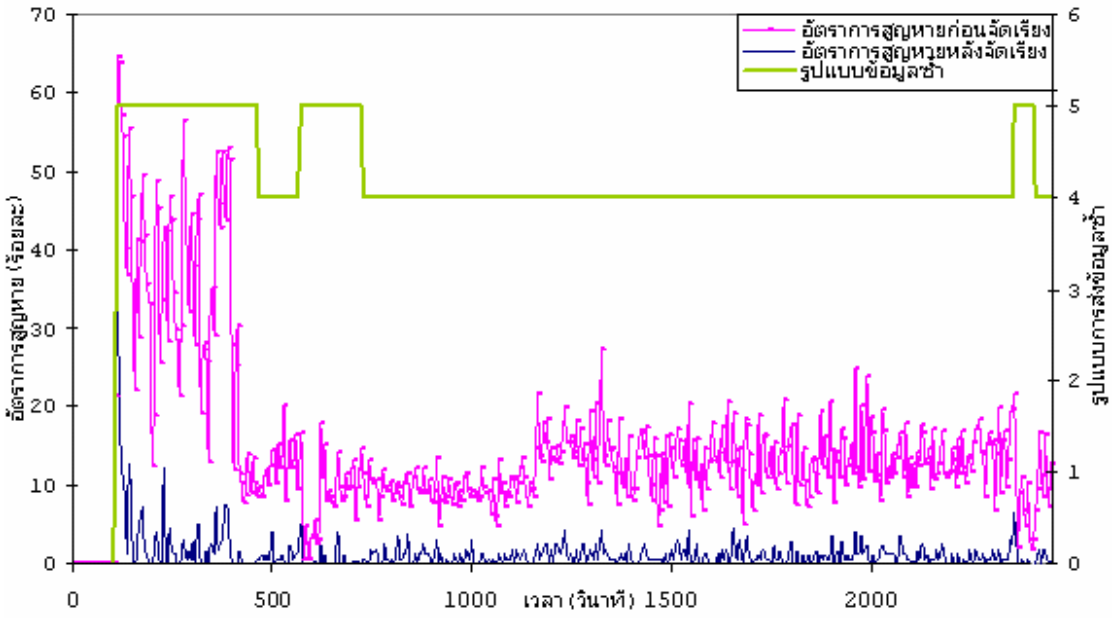
การทดลองนี้ได้ทำการจำลองให้มีทราฟฟิกของข้อมูลในเครือข่ายมีมากที่วินาทีที่ 100 ถึงวินาทีที่ 400 จากนั้นเริ่มสังเกตการณ์ปรับตัวของ combination number ในเครือข่าย ซึ่งจากเครือข่ายที่กำหนด ได้ทำการจำลองอัลกอริทึมทั้งหมด 3 ครั้ง ครั้งแรกเป็นการจำลองการทำงานของอัลกอริทึม Bolot ครั้งที่สองเป็นการจำลองการทำงานของอัลกอริทึม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

เมื่อพิจารณากราฟในรูปที่ 4.2 ด้านล่างที่แสดงร้อยละของข้อมูลที่สูญหายก่อนและหลังการจัดเรียงดังเส้นกราฟ จะเห็นได้ว่า ค่า combination number ที่ทำให้ค่า L_a ไม่เกินค่าเทรสโฮลด์ HIGH จะมีการปรับตัวเมื่อวินาทีที่ 100 จาก combination number เท่ากับ 0 ปรับตัวไปหยุดอยู่ที่ combination number เท่ากับ 5 จนทำให้ค่า L_a ไม่เกินค่าเทรสโฮลด์ HIGH จึงเริ่มมีการเปลี่ยนแปลงค่า Combination number และค่าต่ำสุดอยู่ที่ Combination number เท่ากับ 4

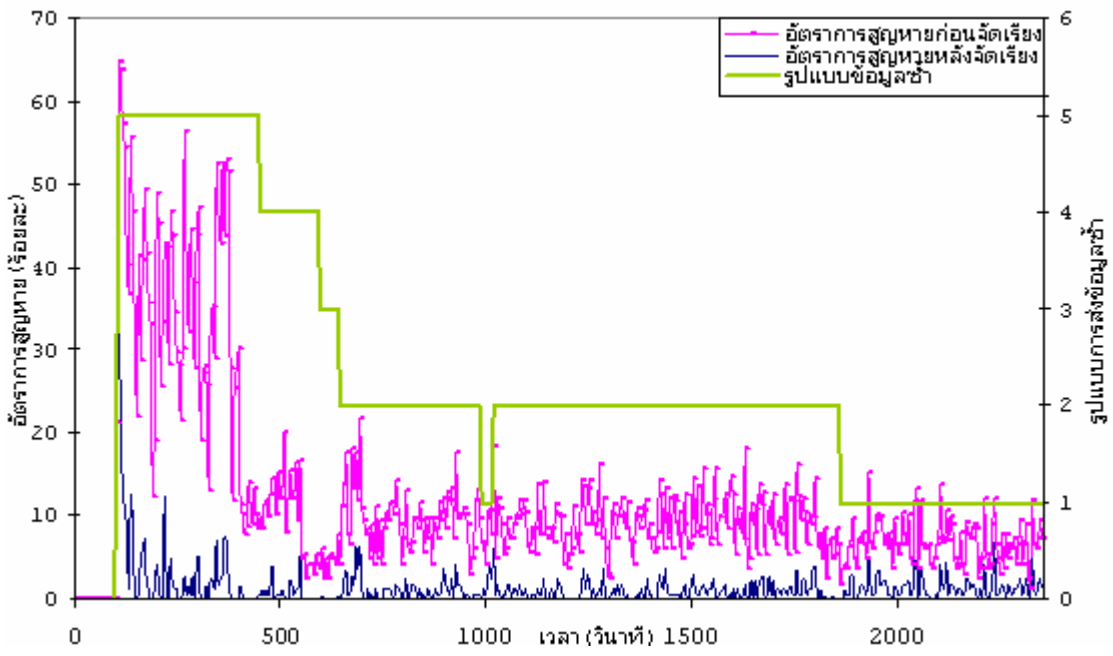
เมื่อพิจารณากราฟในรูปที่ 4.3 แสดงให้เห็นการปรับตัวของค่า Combination number เป็นไปตามความหนาแน่นของปริมาณทราฟฟิกในเครือข่าย อัลกอริทึม CNR มีการปรับค่า Combination number จากค่า 0 ไปเป็นค่า 5 แล้วหลังวินาทีที่ 450 จะเริ่มลดลงจนไปหยุดที่ Combination number เท่ากับ 2 จากนั้นได้มีการปรับตัวลดลงเป็น Combination number เท่ากับ 1 เมื่ออัลกอริทึมคำนวณแล้วว่าค่า L_a มีเทรสโฮลด์ไม่เกินค่าเทรสโฮลด์ HIGH

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

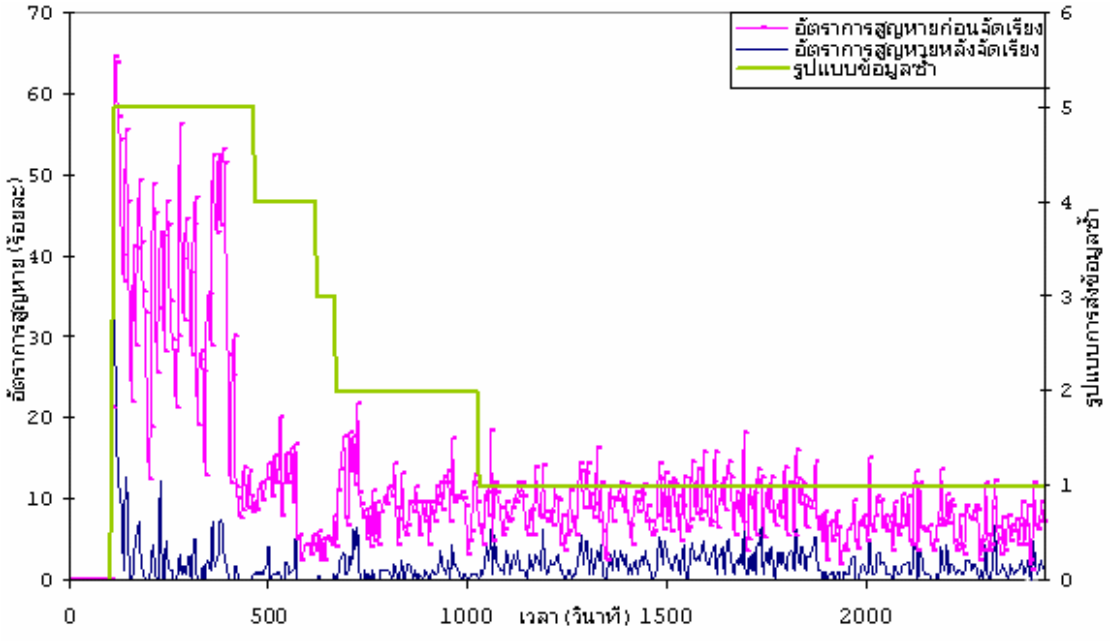
ที่ Combination number เท่ากับ 1 โดยค่าร้อยละของการสูญหายหลังการจัดเรียงแพ็กเก็ตยังอยู่ที่ระดับที่ยอมรับได้ นั่นคือ L_a ไม่เกิน 5%



รูปที่ 4.2 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม Bolot



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

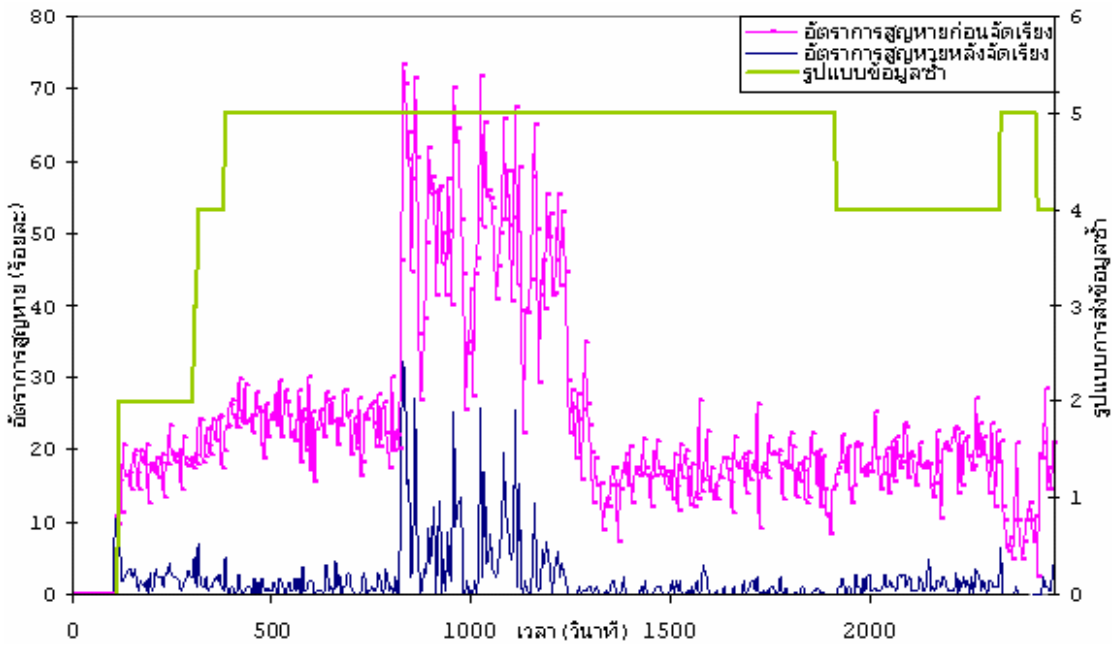


รูปที่ 4.4 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการหาค่า reward

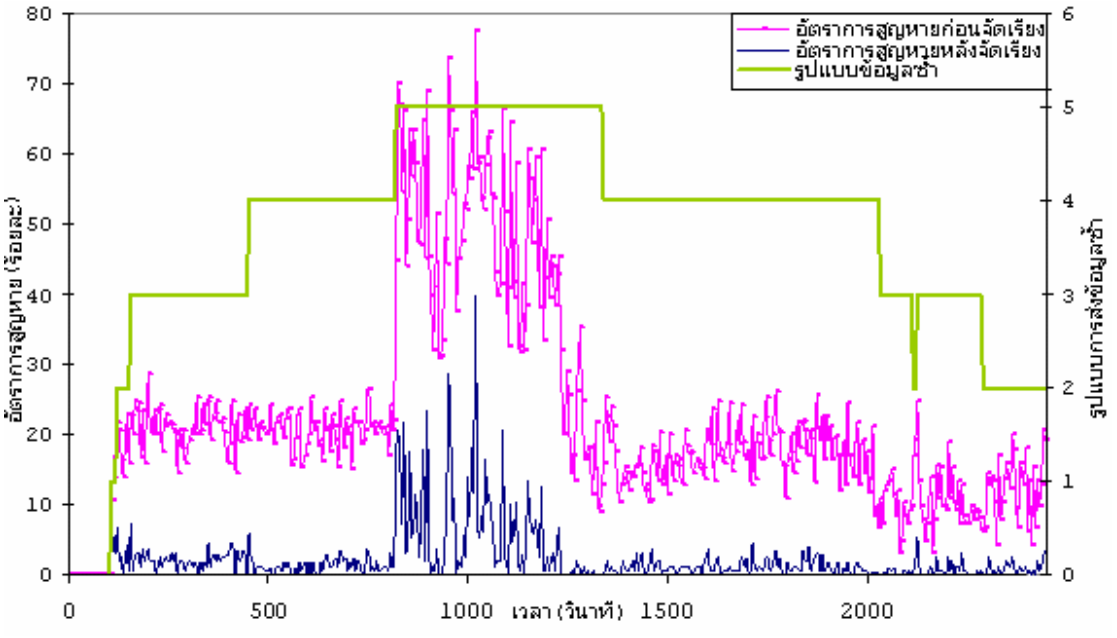
4.2.2 ผลการทดลองตอนที่ 2

การทดลองตอนที่ 2 ได้มีการเปลี่ยนแปลงที่โหนด TS_{l1} - TS_{ln} และ TS_{s1} - TS_{sn} ซึ่งจะเริ่มส่งและหยุดส่งข้อมูลพร้อมกันโดยเริ่มส่งในวินาทีที่ 800 และหยุดส่งที่วินาทีที่ 1200 เพื่อสังเกตการปรับตัวของอัลกอริทึม โดยให้มีความหนาแน่นของทราฟฟิกสูงช่วงกลาง ของเวลาที่ทำการจำลองทั้งหมด และในการทดลองนี้ก็มีการรันแบบจำลอง 2 ครั้งต่อหนึ่งอัลกอริทึมแตกต่างกับในตอนที่ 1 โดยครั้งแรกใช้จำนวนโหนด TCP เท่ากับ 20 โหนด และครั้งที่สองใช้จำนวนโหนด TCP เป็น 40 โหนดเพื่อสังเกตการณ์ปรับตัวเสมือนเพิ่มผู้ส่งและผู้รับขึ้นในเครือข่าย

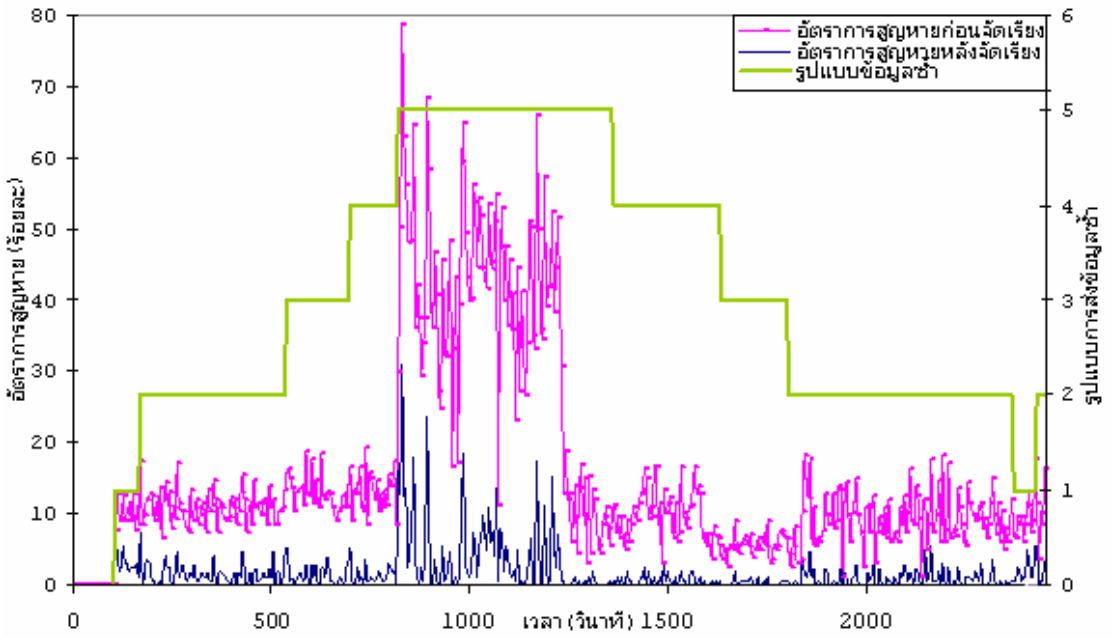
ช่วงแรกทำการจำลองโดยใช้จำนวนโหนด TCP เท่ากับ 20 โหนด ทำการจำลองอัลกอริทึมทั้งหมด 3 ครั้ง ครั้งแรกเป็นการจำลองการทำงานของอัลกอริทึม Bolot ครั้งที่สองเป็นการจำลองการทำงานของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ SES ครั้งที่สามเป็นการจำลองการทำงานของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ Least Square ซึ่งผลการทดลองแสดงดังกราฟในรูปที่ 4.5,4.6 และ 4.7 ตามลำดับ



รูปที่ 4.5 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม Bolot (TCP 20 โหนด)

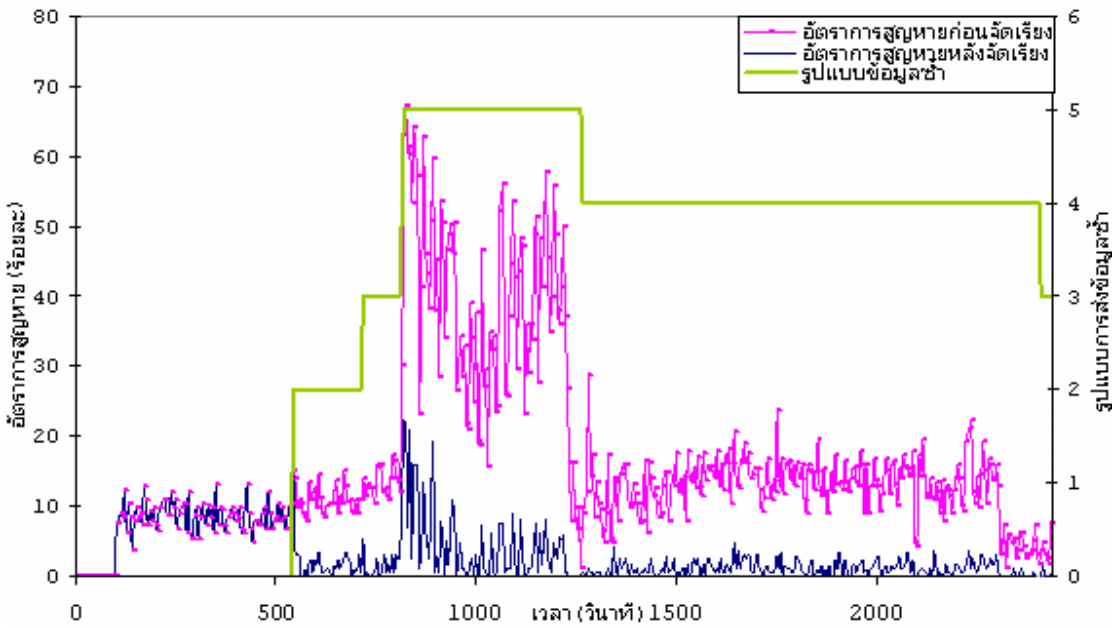


รูปที่ 4.6 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ SES ในการหาค่า reward (TCP 20 โหนด)

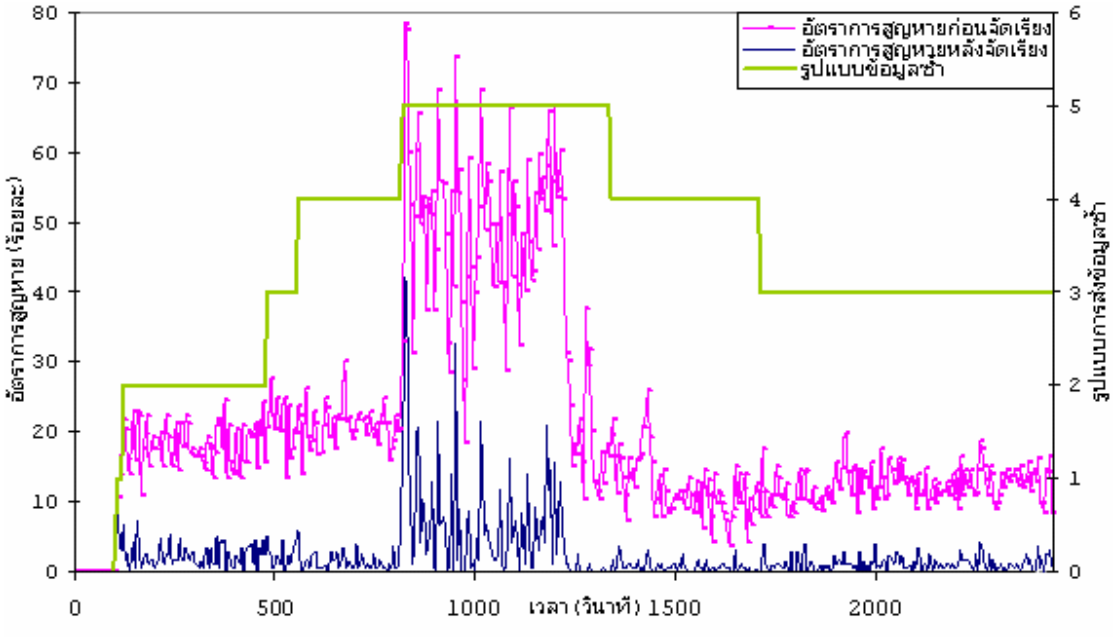


รูปที่ 4.7 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ Least Square ในการหาค่า reward (TCP 20 โหนด)

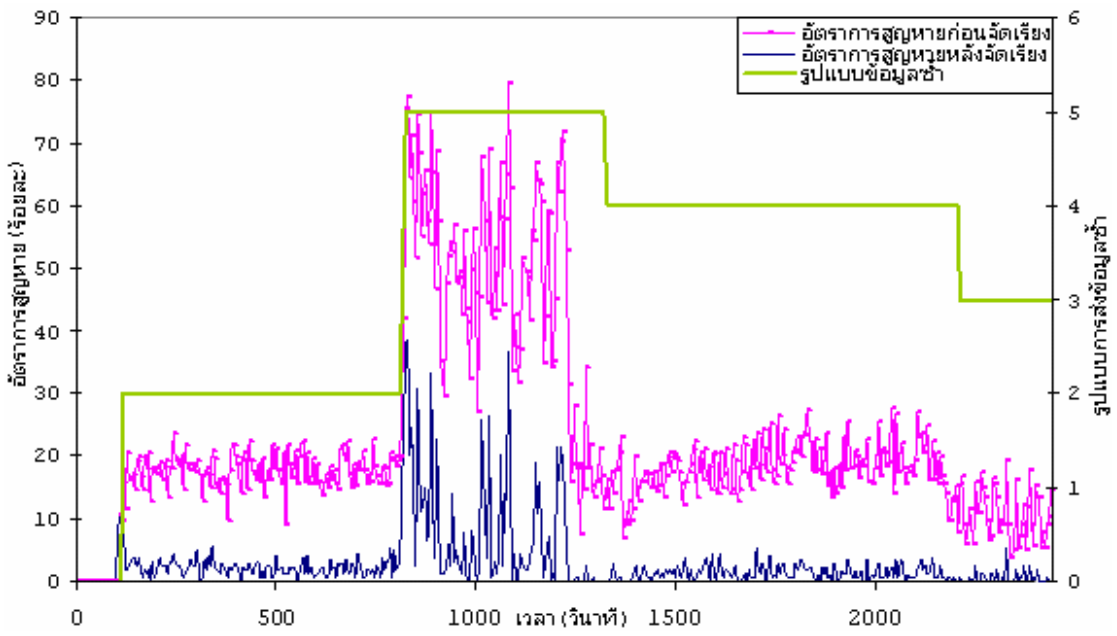
ช่วงที่สองทำการจำลองโดยใช้จำนวนโหนด TCP เท่ากับ 40 โหนด ทำการจำลองอัลกอริทึมทั้งหมด 3 ครั้ง ครั้งแรกเป็นการจำลองการทำงานของอัลกอริทึม Bolot ครั้งที่สองเป็นการจำลองการทำงานของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ SES ครั้งที่สามเป็นการจำลองการทำงานของอัลกอริทึม CNR ซึ่งหาค่า reward โดยวิธีการ Least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



รูปที่ 4.8 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม Bolot

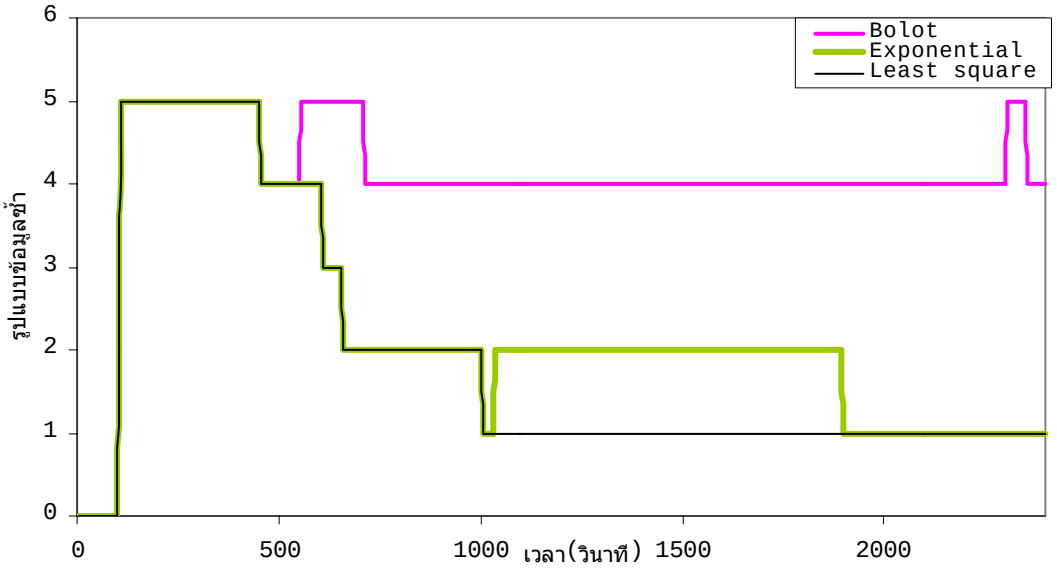


รูปที่ 4.9 แสดงค่า L_a และค่า L_b จากการคำนวณระหว่างการจำลองการทำงานของอัลกอริทึม CNR โดยใช้วิธีการ SES ในการหาค่า reward (TCP 40 โหนด)



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

จะเห็นได้ว่า ในรูปกราฟที่มาการจำลองโดยใช้โหนด 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 ที่มีการเปลี่ยนแปลงตามเวลาและขึ้นอยู่กับการเปลี่ยนแปลงของค่า 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 มีการใช้รูปแบบและปริมาณข้อมูลซ้ำที่น้อยกว่า 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 โดยค่าพารามิเตอร์ที่วัดได้จากการส่งผ่านแพ็กเก็ตเสียงในเส้นทางที่อยู่ระหว่างโหนด N2 กับโหนด N3 ในการวัดจำนวนแพ็กเก็ตที่ส่งไปยังโหนดผู้รับ(Received Packets), ค่าเฉลี่ยทรูพุต(Average Throughput (kbps)) และ ค่าเฉลี่ยเวลาหน่วง(Average Delay (ms)) ต้องระบุค่าขนาดของแพ็กเก็ตในที่นี้คือ 24 ไบต์, ตำแหน่งปลายทางและต้นทางของโหนดที่ทำการวัดค่า รวมทั้งชื่อไฟล์ที่เป็น tracefiles ของระบบที่ทำการจำลองขึ้นสามารถแสดงได้ดังตารางที่ 4.2 ซึ่งเป็นค่าพารามิเตอร์ที่วัดได้จาก Trace file

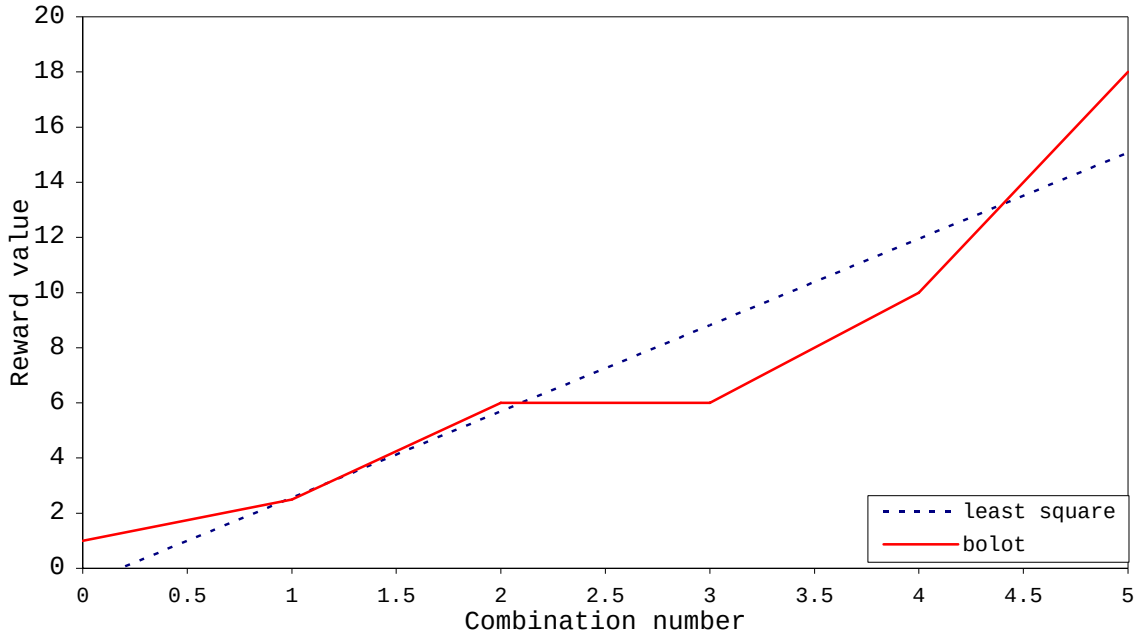
ตารางที่ 4.2 แสดงค่าพารามิเตอร์ที่สำคัญจาก Trace file เปรียบเทียบ 3 วิธีการ

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, ค่าเฉลี่ยทรูพุต (Average Throughput) ที่เพิ่มขึ้น ซึ่งเป็นค่าที่ดีกว่าระบบเดิมที่ใช้อัลกอริทึม Bolot ในการตัดสินเลือกค่า 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 ยังสามารถส่งข้อมูลซ้ำน้อยกว่าเดิม โดยประสิทธิภาพการส่งแพ็กเก็ตเสียงโดยรวมของเครือข่ายดีขึ้น



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