บทที่ 4

การควบคุมคุณภาพแบบปรับตัวสำหรับการสื่อสารวีดิทัศน์

เนื่องจากอินเตอร์เน็ตเป็นเครือข่ายที่ไม่มีการรับประกันคุณภาพการบริการ ดังนั้นจึงไม่สามารถที่จะรับประกันได้ว่าในการสื่อสารวีดิทัศน์บนเครือข่ายอินเตอร์เน็ต วีดิทัศน์ที่ผู้รับได้รับจะมีคุณภาพอยู่ในระดับที่ยอมรับได้หรือไม่ เนื่องจากว่าอาจมีแพ็กเก็ตสูญหายไปในระหว่างการเดินทางผ่านเครือข่าย และถ้ามีปริมาณแพ็กเก็ตสูญหายมากเกินไปก็อาจจะทำให้คุณภาพของวีดิทัศน์ต่ำเกินกว่าที่จะยอมรับได้ ดังนั้นโปรแกรมประยุกต์สำหรับการสื่อสารวีดิทัศน์ที่ดีจึงควรมีความสามารถในการปรับตัวได้ตามสภาพเครือข่าย โดยอาจจะมีการปรับพารามิเตอร์บางตัวในการส่งวีดิทัศน์ เพื่อลดโอกาสในการสูญหายของแพ็กเก็ต หรือมีวิธีการที่ทำให้การสูญหายของแพ็กเก็ตมีผลกระทบต่อสายตาผู้รับให้น้อยที่สุด เป็นต้น แต่เนื่องจากว่างานวิจัยก่อนหน้านี้เกี่ยวกับการปรับตัวของการสื่อสารวีดิทัศน์บนเครือข่ายอินเตอร์เน็ต[7][9][11][21] ยังมีจุดบกพร่องอยู่และไม่ได้มีลักษณะที่เป็นการควบคุมคุณภาพของวีดิทัศน์อย่างชัดเจน ดังนั้นในวิทยานิพนธ์นี้จึงได้เสนอวิธีการการควบคุมคุณภาพแบบปรับตัวสำหรับการสื่อสารวีดิทัศน์ขึ้นมาใหม่ เพื่อให้การสื่อสารวีดิทัศน์สามารถปรับพารามิเตอร์ของการส่งวีดิทัศน์ได้เหมาะสมกับสภาพเครือข่าย และสามารถควบคุณภาพของวีดิทัศน์ไม่ให้อยู่ในระดับที่ต่ำเกินกว่าที่ผู้ใช้จะยอมรับได้ โดยวิธีการดังกล่าวได้รับการออกแบบเพื่อใช้กับการสื่อสารวีดิทัศน์ที่ใช้การบีบอัดวีดิทัศน์ด้วย MPEG-4 ซึ่งหลักการพื้นฐานเกี่ยวกับ MPEG-4 นั้นได้มีการอธิบายไว้ในบทที่ 2

เนื้อหาในบทนี้เริ่มต้นด้วยการกล่าวถึงวิธีการวัดคุณภาพวีดิทัศน์ในหัวข้อที่ 4.1 โดยได้มีการระบุว่ามีพารามิเตอร์ตัวใดบ้างที่ใช้ในการบ่งบอกถึงคุณภาพของวีดิทัศน์ จากนั้นในหัวข้อที่ 4.2 ได้มีการกล่าวถึงงานวิจัยที่เกี่ยวข้องกับการปรับตัวของโปรแกรมประยุกต์สำหรับการสื่อสารวีดิทัศน์บนเครือข่ายอินเตอร์เน็ตที่ได้รับการเสนอขึ้นก่อนหน้านี้ หัวข้อ 4.3 มีเนื้อหาเกี่ยวกับการทดลองเกี่ยวกับการปรับอัตราบิตในการส่งวีดิทัศน์ โดยมีจุดประสงค์เพื่อพิสูจน์ให้เห็นว่าการปรับอัตราบิตของการส่งวีดิทัศน์ไม่สามารถช่วยให้คุณภาพของวีดิทัศน์ดีขึ้นได้เสมอไป และในหัวข้อที่ 4.4 มีเนื้อหาเกี่ยวกับอัลกอริทึม Adaptive Increase Multiplicative Decrease (AIMD) ซึ่งเป็นตัวอย่างหนึ่งของการควบคุมคุณภาพวีดิทัศน์ด้วยการการปรับอัตราบิต และได้มีการทดลองที่ชี้ให้เห็นถึงจุดบกพร่องของอัลกอริทึมนี้ หัวข้อ 4.5 อธิบายถึงวิธีการลดผลกระทบจากการสูญหายของแพ็กเก็ตในการสื่อสารวีดิทัศน์ หัวข้อ 4.6 อธิบายวิธีการควบคุมคุณภาพแบบปรับตัวสำหรับการสื่อสารวีดิทัศน์โดยใช้การปรับค่าระยะห่างระหว่างเฟรมหลัก ซึ่งค่านี้เป็นพารามิเตอร์ตัวหนึ่งในการบีบอัดวีดิทัศน์ด้วย MPEG-4 โดยมีการประเมินผลอัลกอริทึมดังกล่าวในหัวข้อที่ 4.7 และสุดท้ายคือหัวข้อที่ 4.8 เป็นการสรุปเนื้อหาของบทนี้

4.1 การวัดคุณภาพวีดิทัศน์

การที่จะบอกได้ว่าคุณภาพของการสื่อสารวีดิทัศน์ดีหรือไม่ สามารถดูได้จากวีดิทัศน์ที่ผู้รับได้รับ เนื่องจากวีดิทัศน์ต้นฉบับที่ส่งจากผู้ส่งกับวีดิทัศน์ที่ผู้รับได้รับอาจจะมีคุณภาพไม่เท่ากันก็ได้ การวัดคุณภาพของวีดิทัศน์สามารถแบ่งออกได้เป็น 2 ประเภทคือ การวัดแบบ Subjective และการวัด Objective เช่นเดียวกับการวัดคุณภาพเสียง การวัดแบบ Subjective ทำได้โดยการใช้มนุษย์เป็นผู้ตัดสินว่าวีดิทัศน์ที่ทดสอบมีคุณภาพระดับใด ส่วนการวัดแบบ Objective เป็นการวัดที่ระบุเป็นตัวเลขได้ เช่น การวัดค่า Peak Signal to Noise Ratio (PSNR) ซึ่งเป็นค่าที่บอกความแตกต่างระหว่างวีดิทัศน์ที่ทดสอบกับวีดิทัศน์ต้นฉบับ และการวัดค่าอัตราเฟรม เป็นต้น การควบคุมคุณภาพของการสื่อสารวีดิทัศน์ในบทนี้ ใช้วิธีการวัดคุณภาพวีดิทัศน์แบบ Objective เนื่องจากการวัดแบบ Subjective นั้นต้องใช้จำนวนคนและเวลามาก ไม่เหมาะสมกับการสื่อสารวีดิทัศน์ที่มีลักษณะเป็นเวลาจริง

พารามิเตอร์ที่ใช้ในการระบุคุณภาพของวีดิทัศน์ในวิทยานิพนธ์นี้มี 2 ค่าคือ ค่าอัตราเฟรม และค่า PSNR โดยอัตราเฟรมเป็นค่าที่บอกถึงจำนวนเฟรมวีดิทัศน์ในหนึ่งหน่วยเวลา มีหน่วยเป็นเฟรมต่อวินาที วีดิทัศน์ที่มีอัตราเฟรมสูงแสดงว่ามีความราบรื่นสูงและเป็นที่พอใจของผู้ใช้ วีดิทัศน์ที่มีอัตราเฟรมต่ำเกินไปอาจทำให้ผู้ใช้รู้สึกได้ถึงความไม่ต่อเนื่องของวีดิทัศน์ ส่วนค่า PSNR เป็นค่าที่บอกถึงความแตกต่างของเฟรมวีดิทัศน์ที่ทดสอบกับเฟรมวีดิทัศน์ต้นฉบับ ค่า PSNR สามารถคำนวณได้ดังนี้[37]

$$PSNR = 10\log(\frac{MAX_I^2}{MSE})$$

ค่า MAX_I คือ ค่าสูงสุดของจุดภาพ ซึ่งในกรณีที่จุดภาพมีขนาด 8 บิต ค่าสูงสุดจะมีค่าเท่ากับ 255 ส่วนค่า MSE ซึ่งย่อมาจาก Mean Square Error เป็นค่าเฉลี่ยของความแตกต่างของแต่ละจุดภาพของเฟรมวีดิทัศน์ต้นฉบับกับเฟรมวีดิทัศน์ที่ต้องการจะทดสอบ ซึ่งสามารถคำนวณค่าของ MSE ได้โดยใช้สูตรต่อไปนี้

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j) - K(i,j)]^2$$

เมื่อ I คือเฟรมวีดิทัศน์ต้นฉบับ และ K คือ เฟรมวีดิทัศน์ที่ต้องการจะตรวจสอบ ซึ่งทั้ง I และ K มีขนาด m x n จุดภาพ ถ้าค่า PSNR มีค่ามากแสดงว่าข้อมูลของเฟรมวีดิทัศน์ที่ตรวจสอบกับเฟรมวีดิทัศน์ต้นฉบับมีค่าใกล้เคียงกันมาก นั่นคือมีการตัดข้อมูลออกน้อยนั่นเองและมีความชัดเจนสูงซึ่งเป็นที่พอใจของผู้ใช้ สรุปได้ว่าวีดิทัศน์ที่มีคุณภาพสูงหมายถึงวีดิทัศน์มีค่าอัตราเฟรมสูงและค่า PSNR สูง ซึ่งก็คือวีดิทัศน์ที่มีความราบรื่นสูงและมีความชัดเจนของภาพใกล้เคียงกับต้นฉบับนั่นเอง

4.2 งานวิจัยที่เกี่ยวข้อง (Related Works)

งานวิจัยเกี่ยวกับเทคนิคในการปรับตัวของโปรแกรมประยุกต์สำหรับการสื่อสารวีดิทัศน์บนเครือข่ายอินเตอร์เน็ตสามารถจำแนกได้เป็น 3 ประเภทคือ การปรับอัตราการส่งข้อมูล การควบคุมความผิดพลาด และการปรับตัวที่ผู้รับเป็นตัวขับ แต่ในที่นี้จะพิจารณาเฉพาะสองประเภทแรกเท่านั้น เนื่องจากการปรับตัวที่ผู้รับเป็นตัวขับเป็นเทคนิคที่ใช้ในกรณีที่มีผู้รับจำนวนมาก แต่ในวิทยานิพนธ์นี้เน้นที่การติดต่อแบบจุดต่อจุดซึ่งมีผู้ส่งหนึ่งคนและผู้รับหนึ่งคน และเมื่อพิจารณาจากงานวิจัยเกี่ยวกับเทคนิคการปรับตัวอีกสองประเภทคือ การปรับอัตราการส่งข้อมูลและการควบคุมความผิดพลาด พบว่ายังไม่มีงานวิจัยใดที่มีลักษณะเป็นการควบคุมคุณภาพของวีดิทัศน์อย่างเด่นชัด

สำหรับการปรับตัวโดยการปรับอัตราการส่งข้อมูลนั้น ส่วนใหญ่แล้วจะมีการเพิ่มหรือลดอัตราบิตของการส่งวีดิทัศน์ตามปริมาณการสูญหายของแพ็กเก็ต ใน [7] ได้เสนอวิธีการปรับค่าอัตราบิตของการส่งวีดิทัศน์ โดยวิธีการนี้จะลดอัตราบิตเมื่อเห็นมีแพ็กเก็ตสูญหายมากเกินไป และจะเพิ่มอัตราบิตเมื่อเครือข่ายเห็นว่ามีแพ็กเก็ตสูญหายไม่มากนัก โดยใช้การปรับอัตราบิตแบบ Additive Increase Multiplicative Decrease (AIMD) และมีการประเมินผลวิธีการที่นำเสนอโดยการทดลองบนเครือข่ายจริง โดยใช้โปรแกรม vic[38] ในการส่งวีดิทัศน์ ซึ่งโปรแกรม vic ผู้ใช้สามารถกำหนดอัตราบิตเป้าหมายได้ตามต้องการ แต่ใน [7] ก็ไม่ได้มีการอธิบายว่า vic ปรับพารามิเตอร์ใดบ้างเพื่อให้อัตราบิตของการส่งวีดิทัศน์เป็นไปตามอัตราบิตเป้าหมาย

ใน [9] และ [11] ได้เสนอวิธีการปรับอัตราบิตของการส่งวีดิทัศน์เพื่อให้มีการแบ่งแบนด์วิดท์กับโปรแกรมประยุกต์ที่ใช้โปรโตคอล TCP อย่างเป็นธรรม ซึ่งสมการที่ใช้ในการคำนวณค่าอัตราบิตจะมีการพิจารณาทั้งอัตราการสูญหายของแพ็กเก็ตและค่า Round Trip Time (RTT) ระหว่างผู้ส่งกับผู้รับ แต่ในบทความทั้งสองนี้ไม่ได้อธิบายว่าเมื่อคำนวณอัตราบิตได้แล้วจะมีการปรับพารามิเตอร์ของการส่งวีดิทัศน์อย่างไรเพื่อให้ได้อัตราบิตตามที่ต้องการ

ในโปรแกรม IVS[24] ซึ่งเป็นโปรแกรมประยุกต์สำหรับการประชุมวีดิทัศน์ที่ใช้การบีบอัดวีดิทัศน์ด้วย H.261 โปรแกรมนี้มีการปรับอัตราบิตของการส่งวีดิทัศน์ตามอัตราการสูญหายของแพ็กเก็ตเช่นกัน โดยจะลดอัตราบิตเมื่อมีแพ็กเก็ตสูญหายมากเกินไป และจะเพิ่มอัตราบิตเมื่อเห็นว่ามีแพ็กเก็ตสูญหายไม่มากนัก เมื่อทราบว่าควรจะใช้อัตราบิตเท่าไรแล้ว โปรแกรม IVS มีโหมดในการปรับพารามิเตอร์ของการส่งวีดิทัศน์อยู่ 2 โหมดเพื่อให้ได้อัตราบิตของการส่งวีดิทัศน์เป็นไปตามอัตราบิตเป้าหมายที่คำนวณเอาไว้ โดยโหมดแรกมีชื่อว่า โหมด Priviledge Quality (PQ) ซึ่งโหมดนี้จะปรับค่าอัตราเฟรมเพื่อให้ได้อัตราบิตตามที่ต้องการ โดยค่าระดับการควอนไทซ์ในแต่ละเฟรมมีค่าคงที่ นั่นคือในแต่ละปริมาณการตัดข้อมูลออกในระดับคงที่ ความชัดเจนของภาพในแต่ละเฟรมจึงไม่มีการเปลี่ยนแปลงตามอัตราบิตเป้าหมาย หรืออาจจะกล่าวได้ว่าในโหมด PQ นี้มีการปรับอัตราบิตโดยที่คุณภาพของภาพคงที่ แต่เปลี่ยนอัตราเฟรม ส่วนโหมดที่

สองคือ โหมด Privilede Frame Rate (PFR) ซึ่งในโหมดนี้จะไม่มีการเปลี่ยนค่าอัตราเฟรม แต่จะมีการปรับค่าระดับการควอนไทซ์ของการบีบอัดวีดิทัศน์แต่ละเฟรมเพื่อให้ได้อัตราบิตของการส่งวีดิทัศน์ตามที่ต้องการ เช่น หากต้องการลดอัตราบิตสามารถทำได้โดยการเพิ่มค่าระดับการควอนไทซ์ซึ่งจะทำให้ข้อมูลของวีดิทัศน์แต่ละเฟรมถูกตัดทิ้งมากขึ้น โดยสรุปแล้วโหมด PFR นี้ปรับอัตราบิตโดยที่อัตราเฟรมยังคงที่แต่มีการเปลี่ยนคุณภาพของภาพในแต่ละเฟรม

จากเทคนิคการปรับตัวโดยการปรับอัตราบิตของการส่งวีดิทัศน์ที่กล่าวมาทั้งหมดนี้ ต่างมีสมมุติฐานว่าการลดอัตราบิตของการส่งวีดิทัศน์สามารถช่วยลดความคับคั่งของเครือข่ายและทำให้อัตราการสูญหายของแพ็กเก็ตวีดิทัศน์ลดลงตามไปด้วย ผลที่ตามคือทำให้วีดิทัศน์ที่ผู้รับได้รับมีคุณภาพดีขึ้น แต่สมมุติฐานนี้ไม่ได้เป็นจริงเสมอไปในทุกสภาพแวดล้อม โดยเฉพาะในกรณีที่มีปริมาณการใช้เครือข่ายเต็มแบนด์วิดท์ของเครือข่ายซึ่งจะเห็นได้จากผลการทดลองในหัวข้อที่ 4.3

นอกจากนี้เทคนิคการปรับตัวดังกล่าวจะมีการปรับอัตราบิตตามค่าอัตราการสูญหายของแพ็กเก็ต ซึ่งไม่สามารถระบุถึงคุณภาพของวีดิทัศน์ได้ชัดเจน เนื่องจากว่าข้อมูลหลังการบีบอัดของวีดิทัศน์ 1 เฟรมอาจจะมีขนาดใหญ่และต้องบรรจุลงในแพ็กเก็ตมากกว่า 1 แพ็กเก็ต จำนวนแพ็กเก็ตที่สูญหายไม่สามารถบอกได้ว่ามีผลทำให้เฟรมวีดิทัศน์เสียหายไปกี่เฟรม ตัวอย่างเช่น หากมีแพ็กเก็ตสูญหายจำนวน 4 แพ็กเก็ต ถ้าทั้ง 4 แพ็กเก็ตได้มาจากวีดิทัศน์ในเฟรมเดียวกันจะทำให้มีเฟรมวีดิทัศน์เสียหายเพียงหนึ่งเฟรม แต่ถ้าทั้ง 4 แพ็กเก็ตนี้ได้มาจากวีดิทัศน์ในเฟรมที่แตกต่างกันทั้งหมด จะทำให้มีเฟรมวีดิทัศน์เสียหาย 4 เฟรม ดังนั้นหากต้องการควบคุมคุณภาพของวีดิทัศน์ แทนที่จะใช้อัตราการสูญหายของแพ็กเก็ต ควรใช้อัตราการสูญหายของวีดิทัศน์ทั้งเฟรมหรือใช้ค่าอัตราเฟรมของวีดิทัศน์ที่ผู้รับได้รับจะเหมาะสมกว่า

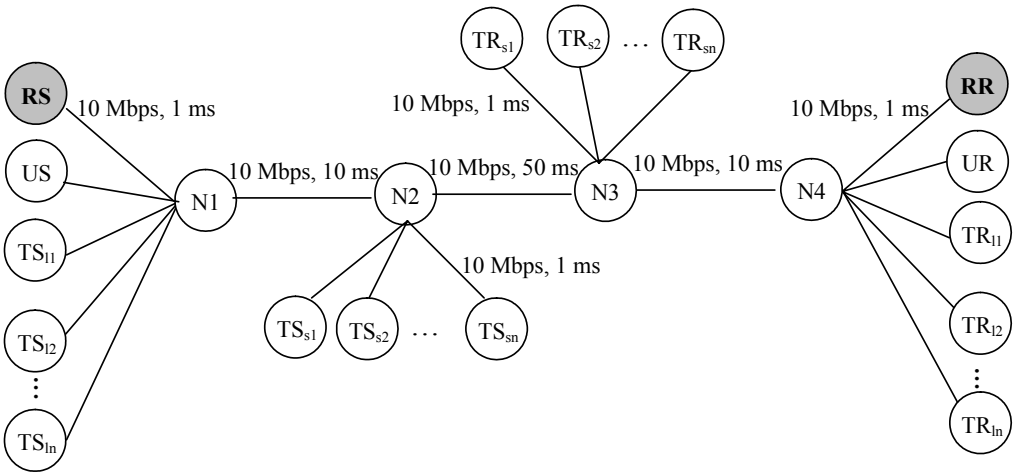
ส่วนงานวิจัยเกี่ยวกับเทคนิคการควบคุมความผิดพลาดสามารถแบ่งย่อยได้อีกเป็น 2 ประเภท ประเภทแรกคือมีการใช้ Forward Error Correction (FEC) ในการลดผลกระทบจากการสูญหายของแพ็กเก็ต[21] แต่การใช้ FEC กับการสื่อสารวีดิทัศน์ดูจะไม่เหมาะสมเท่าไรนัก เนื่องจากว่าอัตราบิตของการส่งวีดิทัศน์มีค่าสูงกว่าอัตราบิตของการสื่อสารเสียงมาก การใช้ FEC ซึ่งมีการส่งข้อมูลซ้ำจึงทำให้ปริมาณการใช้แบนด์วิดท์เพิ่มขึ้นด้วยค่าที่สูงมากเช่นกัน ส่วนการควบคุมความผิดพลาดของวีดิทัศน์ประเภทที่สองทำได้โดยการปรับพารามิเตอร์ของการบีบอัด เช่น การลดจำนวนเฟรม P ซึ่งสามารถลดผลกระทบจากการสูญหายของแพ็กเก็ตได้ ในโปรแกรม IVS[24] ได้มีการปรับตัวโดยการปรับระยะห่างระหว่างเฟรมหลัก (เฟรม I) แต่ใน [24] ไม่ได้มีการอธิบายเอาไว้ว่าจะเพิ่มหรือลดระยะห่างระหว่างเฟรมเมื่อใด เพียงแต่อธิบายว่ามีการปรับระยะห่างระหว่างเฟรมหลักตามอัตราการสูญหายของแพ็กเก็ตเท่านั้น ซึ่งได้กล่าวไปแล้วข้างต้นค่าอัตราการสูญหายของแพ็กเก็ตนั้นไม่สามารถระบุได้ว่ามีเฟรมวีดิทัศน์เสียหายจำนวนกี่เฟรม การปรับตัวควรที่จะปรับตามอัตราการสูญหายของวีดิทัศน์ทั้งเฟรม หรือใช้ค่าอัตราเฟรมของวีดิทัศน์ที่ผู้รับได้รับจะเหมาะสมกว่า

เนื่องจากเทคนิคการปรับตัวสำหรับการสื่อสารวีดิทัศน์ที่มีอยู่ก่อนหน้านี้ยังไม่ดีพอสำหรับการควบคุมคุณภาพของวีดิทัศน์ ดังนั้นในวิทยานิพนธ์นี้จึงได้ออกแบบวิธีการควบคุมคุณภาพสำหรับการสื่อสารวีดิทัศน์ที่สามารถปรับตัวได้ตามสภาพความคับคั่งของเครือข่าย โดยใช้ค่าอัตราเฟรมและค่า PSNR เป็นตัวชี้วัดคุณภาพของวีดิทัศน์ แทนการปรับตัวโดยใช้ค่าอัตราการสูญหายของแพ็กเก็ตตามที่ใช้กันในวิธีการก่อนหน้านี้

4.3 การปรับอัตราบิตของการส่งวีดิทัศน์

เทคนิคการปรับตัวโดยการปรับอัตราการส่งข้อมูลส่วนใหญ่จะลดอัตราการส่งข้อมูลเพื่อลดความคับคั่งของเครือข่าย และจะช่วยให้ปริมาณการสูญหายของแพ็กเก็ตลดลงด้วย แต่ในหัวข้อนี้จะพิสูจน์ให้เห็นว่าเทคนิคนี้อาจจะใช้ไม่ได้ผลในบางสถานการณ์ และหากพิจารณาถึงวิธีการในการปรับค่าอัตราบิตของการส่งวีดิทัศน์ให้ได้ตามอัตราบิตเป้าหมายแล้ว สามารถแบ่งการปรับค่าอัตราบิตได้เป็น 2 วิธี โดยวิธีแรกคือการปรับอัตราเฟรมของการส่งวีดิทัศน์ ถ้าวีดิทัศน์มีอัตราเฟรมมากอัตราบิตของการส่งวีดิทัศน์ก็จะมีค่ามากตามไปด้วย วิธีที่สองคือการเปลี่ยนแปลงปริมาณข้อมูลของวีดิทัศน์หลังการบีบอัดในแต่ละเฟรมหรือที่เรียกว่า ความยาวเฟรม (Frame Length) ถ้าหากวีดิทัศน์ในแต่ละเฟรมมีความยาวลดลงอัตราบิตก็จะลดลงถึงแม้จะมีอัตราเฟรมเท่าเดิม ในกรณีของการบีบอัดวีดิทัศน์ด้วย MPEG-4 การลดความยาวเฟรมทำได้โดยการเพิ่มค่าระดับการควอนไทซ์ซึ่งจะทำให้มีการตัดข้อมูลในแต่ละเฟรมออกมากขึ้น (ความชัดเจนของภาพลดลง)

และเพื่อเป็นการพิสูจน์สมมุติฐานที่ว่า การลดอัตราบิตของการส่งวีดิทัศน์จะสามารถลดปริมาณแพ็กเก็ตที่สูญหาย และช่วยให้คุณภาพของวีดิทัศน์ดีขึ้นได้ ในหัวข้อนี้จึงได้ทำการทดลองโดยใช้การจำลองบน NS-2 และกำหนดรูปแบบของเครือข่ายที่ใช้ในการจำลองดังแสดงในรูปที่ 4.1 โหนด RS เป็นโหนดที่ส่งแพ็กเก็ตวีดิทัศน์ นอกจากนี้ยังมีโหนดที่ส่งทราฟฟิก TCP (TS_{l1} – TS_{ln} และ TS_{s1} – TS_{sn}) และโหนดที่ส่งทราฟฟิก UDP (โหนด US) เพื่อให้เครือข่ายมีสภาพแวดล้อมเหมือนมีผู้ใช้งานคนอื่นกำลังใช้เครือข่ายอยู่ด้วย ลิงค์ระหว่างโหนด N2 และ N3 มีแบนด์วิดท์ 10 เมกะบิตต่อวินาที (Mbps) เวลาหน่วง 50 มิลลิวินาที ซึ่งลิงค์นี้ถือเป็นลิงค์คอขวดของระบบ ส่วนลิงค์ระหว่างโหนด N1 กับ N2 และลิงค์ระหว่างโหนด N3 กับ N4 มีแบนด์วิดท์ 10 เมกะบิตต่อวินาที เวลาหน่วง 10 มิลลิวินาที ส่วนลิงค์อื่นที่เหลือมีแบนด์วิดท์ 10 เมกะบิตต่อวินาที เวลาหน่วง 1 มิลลิวินาที และลิงค์ทุกลิงค์เป็นลิงค์แบบ Drop Tail และเนื่องจากการปรับค่าอัตราบิตนั้นสามารถทำได้ 2 วิธี ดังนั้นจึงต้องทดลองโดยใช้การปรับค่าอัตราบิตทั้งสองวิธี และแบ่งการทดลองออกเป็น 2 ตอนคือ ตอนที่1 การปรับอัตราบิตโดยการเปลี่ยนค่าอัตราเฟรม และตอนที่ 2 การปรับอัตราบิตโดยการเปลี่ยนความยาวเฟรม



รูปที่ 4.1 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลอง

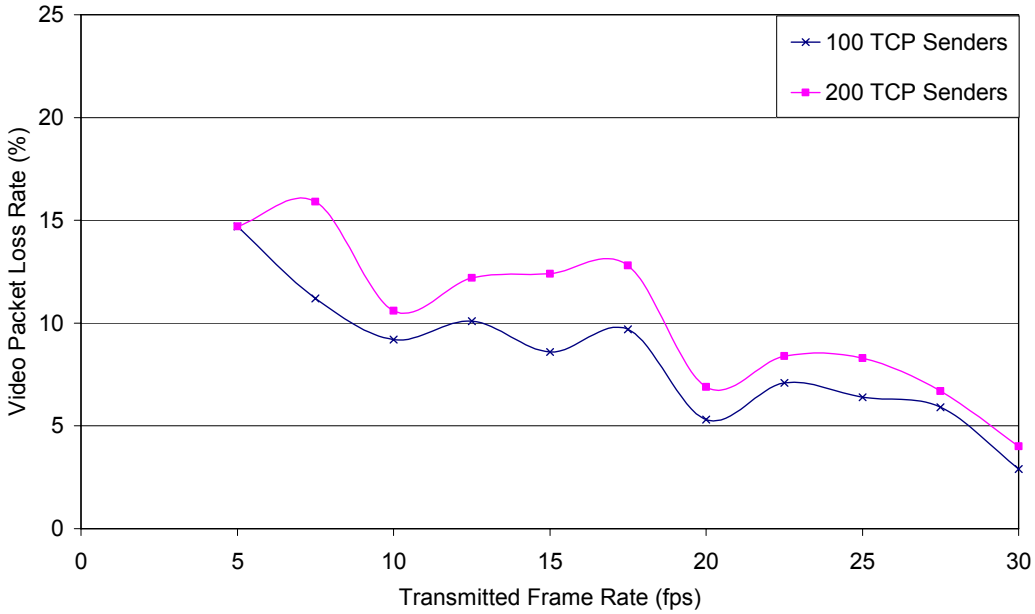
4.3.1 การทดลองตอนที่1: การเปลี่ยนอัตราเฟรม

การทดลองในตอนนี้มีจุดประสงค์เพื่อพิสูจน์ว่าการลดอัตราบิตของการส่งวีดิทัศน์โดยการลดอัตราเฟรม สามารถลดปริมาณการสูญหายของแพ็กเก็ตและทำให้คุณภาพของวีดิทัศน์ดีขึ้นได้หรือไม่ โดยการทดลองตอนนี้ต้องทำการทดลอง 2 รอบซึ่งแต่ละรอบใช้จำนวนโหนดที่ส่งทราฟฟิก TCP ไม่เท่ากัน เพื่อจะได้พิจารณาผลการทดลองในสภาพเครือข่ายที่แตกต่างกัน โดยจำนวนโหนดที่ส่ง TCP ในแต่ละรอบเป็นดังนี้

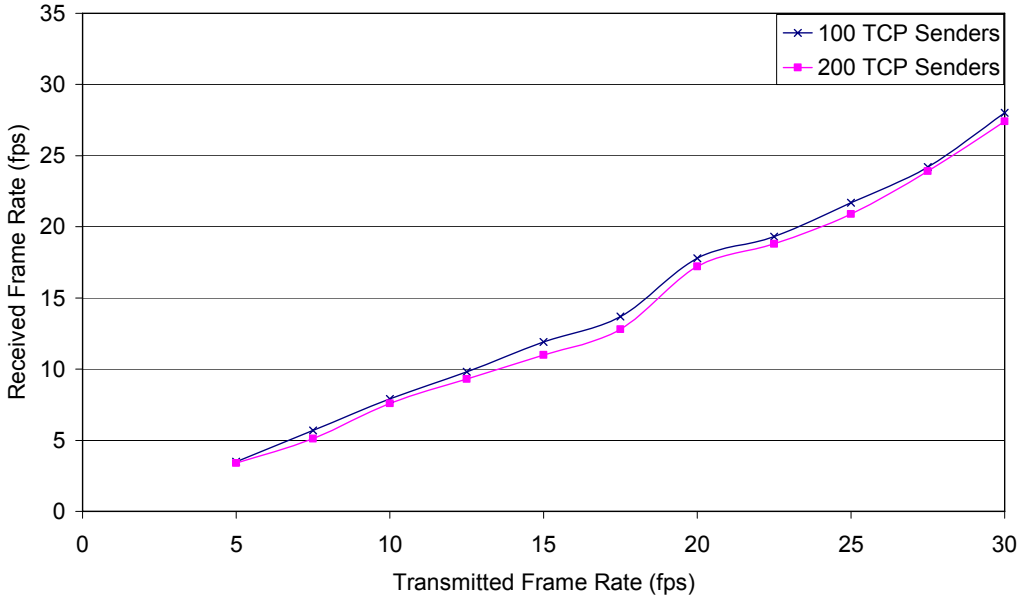
- รอบที่ 1 มีโหนดที่ส่ง TCP 100 โหนด (TS_{l1} – TS_{l50} และ TS_{sl1} – TS_{s50})
- รอบที่ 2 มีโหนดที่ส่ง TCP 200 โหนด (TS_{l1} – TS_{l100} และ TS_{s1} – TS_{s100})

โดยแต่ละรอบมีการจำลอง 6 ครั้งโดยเปลี่ยนค่าอัตราเฟรมตั้งแต่ 5 จนถึง 30 เฟรมต่อวินาที และบันทึกค่าเฉลี่ยของอัตราการสูญหายของแพ็กเก็ตวีดิทัศน์ และค่าเฉลี่ยของอัตราเฟรมฝั่งรับ (Received Frame Rate) ส่วนข้อกำหนดอื่น ๆ มีดังนี้ ความยาวของเฟรมวีดิทัศน์ทุกเฟรมเท่ากับ 4000 ไบต์ การจำลองแต่ละครั้งกำหนดให้ทุกโหนดเริ่มส่งข้อมูลพร้อมกันวินาทีที่ 0 และใช้เวลาการจำลอง 600 วินาที

จากกราฟแสดงความสัมพันธ์ระหว่างค่าอัตราการสูญหายของแพ็กเก็ตและอัตราเฟรมฝั่งส่ง (Transmitted Frame Rate) ในรูปที่ 4.2 จะเห็นว่าการลดอัตราเฟรมฝั่งส่งไม่ได้ทำให้อัตราการสูญหายของแพ็กเก็ตลดลงเสมอไป แต่กลับมีแนวโน้มที่จะทำให้อัตราการสูญหายของแพ็กเก็ตเพิ่มขึ้นด้วยซ้ำ ทั้งนี้เนื่องจากว่าเมื่อโหนด RS ลดอัตราเฟรมในการส่งวีดิทัศน์ อัตราบิตที่ใช้ก็ลดลงตามไปด้วย โหนดที่ส่งทราฟฟิก TCP จะเพิ่มอัตราการส่งข้อมูลขึ้นเนื่องจากว่ามีแบนด์วิดท์ที่สามารถใช้งานได้เพิ่มขึ้น ความคับคั่งของเครือข่ายจึงไม่ได้ลดลงแต่อย่างใด และอาจจะทำให้อัตราการสูญหายของแพ็กเก็ตวีดิทัศน์เพิ่มขึ้นได้



รูปที่ 4.2 ค่าอัตราการสูญหายของแพ็กเก็ตวีดิทัศน์เมื่อมีการเปลี่ยนอัตราเฟรมฝั่งส่ง



รูปที่ 4.3 อัตราเฟรมของวีดิทัศน์ที่ได้รับเมื่อมีการเปลี่ยนอัตราเฟรมฝั่งส่ง

เนื่องจากการพิจารณาเฉพาะอัตราการสูญหายของแพ็กเก็ตเพียงอย่างเดียวไม่สามารถบอกได้ว่ามีวีดิทัศน์เสียหายกี่เฟรม และวีดิทัศน์ที่ผู้รับได้รับมีอัตราเฟรมเท่าไร ดังนั้นในการทดลองนี้จึงได้มีการบันทึกค่าอัตราเฟรมของวีดิทัศน์ที่ฝั่งรับด้วย และเมื่อพิจารณาการได้รับเฟรม

วีดิทัศน์ของผู้รับ จากกราฟแสดงความสัมพันธ์อัตราเฟรมฝั่งรับและอัตราเฟรมฝั่งส่งในรูปที่ 4.3 จะเห็นว่าเมื่อผู้ส่งลดอัตราเฟรมในการส่งวีดิทัศน์ ผู้รับก็จะได้รับวีดิทัศน์ในอัตราเฟรมที่ต่ำลงไปด้วย ดังนั้นการลดอัตราเฟรมในการส่งวีดิทัศน์จึงไม่ได้ช่วยให้วีดิทัศน์มีคุณภาพดีขึ้น

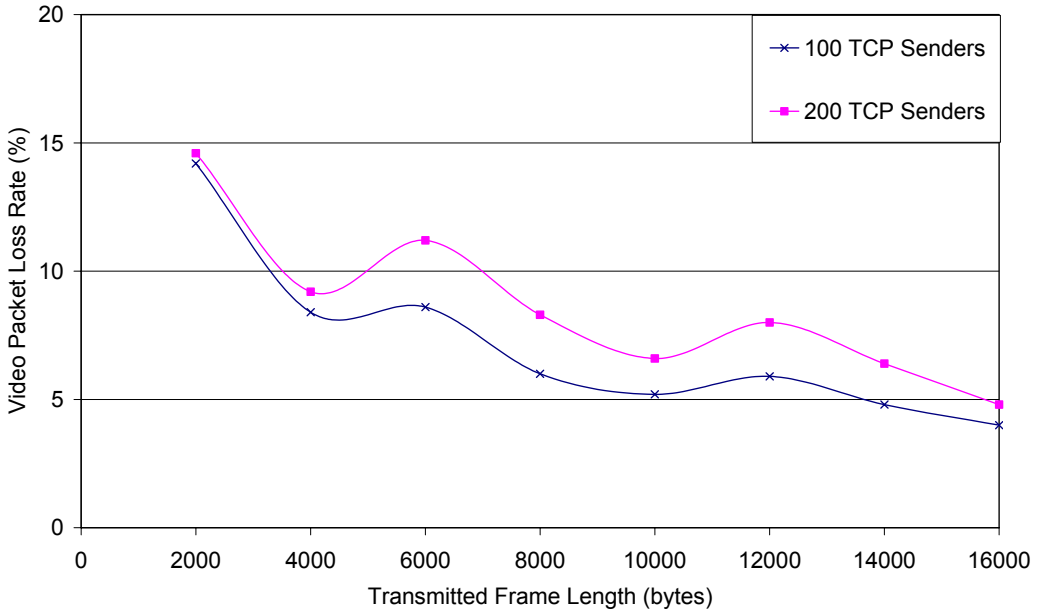
4.3.2 การทดลองตอนที่ 2: การเปลี่ยนความยาวของเฟรม

การทดลองตอนนี้ใช้วิธีการปรับอัตราบิตโดยการเปลี่ยนความยาวเฟรม แต่ใช้อัตราเฟรมคงที่ ซึ่งในทางปฏิบัตินั้นการเปลี่ยนแปลงความยาวเฟรมทำได้โดยการปรับความชัดเจนของวีดิทัศน์ในแต่ละเฟรม ตัวอย่างเช่น ในกรณี MPEG-4 การเพิ่มค่าระดับการควอนไทซ์จะทำให้ทีการตัดข้อมูลในแต่ละเฟรมออกมากขึ้น (ความชัดเจนของภาพลดลง) ส่งผลให้ความยาวเฟรมลดลง โดยรูปแบบของการทดลองในตอนที่ 2 นี้เหมือนกันในตอนที่ 1 แต่แทนที่จะเปลี่ยนอัตราเฟรมในการจำลองแต่ละครั้ง ในตอนนี้จะเปลี่ยนความยาวเฟรมแทน ขั้นตอนในการทดลองนี้แบ่งออกเป็น 2 รอบ โดยมีจำนวนโหนดที่ส่งทราฟฟิก TCP ในแต่ละรอบเท่ากับจำนวนที่ใช้ในการทดลองตอนที่ 1 โดยแต่ละรอบมีการจำลอง 6 ครั้ง ซึ่งมีการใช้ความยาวเฟรมของวีดิทัศน์ตั้งแต่ 2000 – 16000 ไบต์ และบันทึกค่าเฉลี่ยของอัตราการสูญหายของแพ็กเก็ต และค่าเฉลี่ยของอัตราเฟรมฝั่งรับ ส่วนข้อกำหนดอื่น ๆ มีดังนี้ ขนาดของเพย์โหลดสูงสุดของแต่ละแพ็กเก็ตเท่ากับ 1460 ไบต์ ดังนั้นการบรรจุข้อมูลของเฟรมที่มีขนาดใหญ่ก็จำเป็นที่จะต้องใช้จำนวนแพ็กเก็ตมากกว่าเฟรมที่มีขนาดเล็ก การจำลองแต่ละครั้งกำหนดให้ทุกโหนดเริ่มส่งข้อมูลพร้อมกันวินาทีที่ 0 และใช้เวลาการจำลอง 600 วินาที

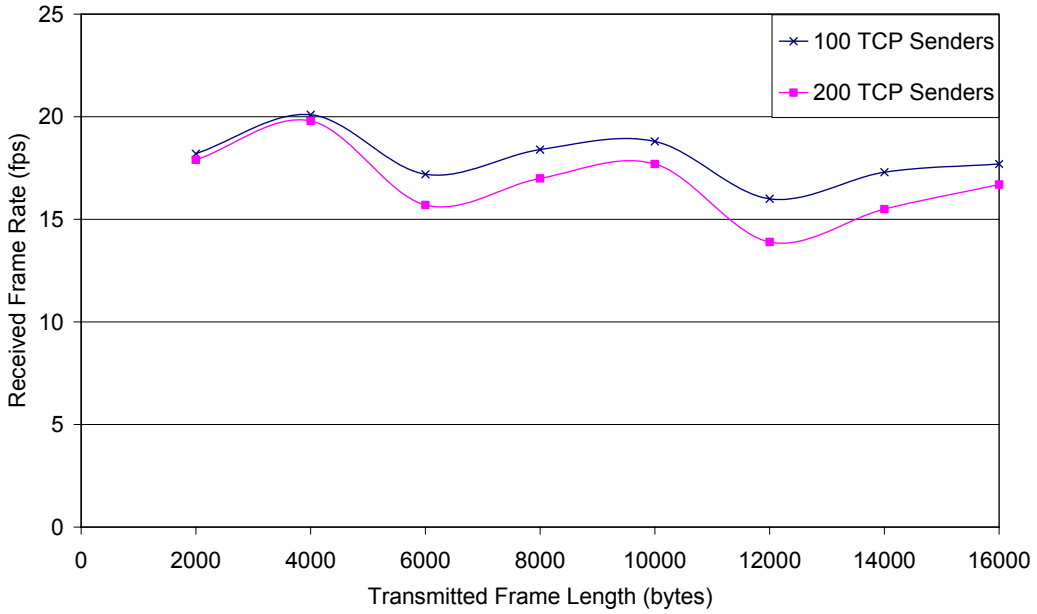
จากกราฟแสดงอัตราการสูญหายของแพ็กเก็ตวีดิทัศน์เมื่อมีการเปลี่ยนความยาวเฟรมในรูปที่ 4.4 จะเห็นว่าการลดอัตราบิตโดยการลดความยาวของเฟรมวีดิทัศน์ไม่ได้ทำให้อัตราการสูญหายของแพ็กเก็ตวีดิทัศน์ลดลงเสมอไป เช่น เมื่อใช้ความยาวเฟรมเท่ากับ 2000 ไบต์ทำให้มีแพ็กเก็ตสูญหายมากกว่าเมื่อใช้ความยาวเฟรมเท่ากับ 16000 ไบต์ และจากกราฟจะเห็นว่าการลดความยาวเฟรมมีแนวโน้มที่จะทำให้อัตราการสูญหายของแพ็กเก็ตเพิ่มขึ้น ซึ่งไม่เป็นไปตามสมมุติฐานที่ว่าการลดอัตราบิตในการส่งวีดิทัศน์ทำให้อัตราการสูญหายของแพ็กเก็ตลดลง ทั้งนี้เนื่องจากว่าเมื่อโหนด RS ลดความยาวเฟรมลง อัตราบิตที่ใช้ก็ลดลงตามไปด้วย โหนดที่ส่งทราฟฟิก TCP จะเพิ่มอัตราการส่งข้อมูลขึ้นเนื่องจากว่ามีแบนด์วิดท์ที่สามารถใช้งานได้เพิ่มขึ้น ความคับคั่งของเครือข่ายจึงไม่ได้ลดลงแต่อย่างใด และอาจจะทำให้อัตราการสูญหายของแพ็กเก็ตวีดิทัศน์เพิ่มขึ้นได้

เมื่อพิจารณากราฟแสดงอัตราเฟรมฝั่งรับในรูปที่ 4.5 จะเห็นว่าอัตราเฟรมฝั่งรับมีค่าค่อนข้างใกล้เคียงกันไม่ว่าจะใช้ความยาวเฟรมเท่าใด โดยทั่วไปแล้วในกรณีที่อัตราการสูญหายของแพ็กเก็ตคงที่ เฟรมที่มีขนาดใหญ่จะมีโอกาสเสียหายได้ง่ายกว่าเฟรมที่มีขนาดเล็ก เนื่องจากเฟรมที่มีขนาดใหญ่จะต้องมีการตัดแบ่งและต้องใช้จำนวนแพ็กเก็ตมากกว่าเฟรมที่มีขนาดเล็ก แต่

ในการทดลองนี้ค่าอัตราการสูญหายของแพ็กเก็ตไม่คงที่ ซึ่งจากกราฟในรูปที่ 4.4 จะเห็นว่าเฟรมที่มีขนาดเล็กจะมีแพ็กเก็ตสูญหายมากกว่าเฟรมที่มีขนาดใหญ่ จึงทำให้ค่าอัตราเฟรมฝั่งรับมีค่าค่อนข้างใกล้เคียงกันไม่ว่าจะใช้ความยาวเฟรมเท่าใด



รูปที่ 4.4 อัตราการสูญหายของแพ็กเก็ตวีดิทัศน์เมื่อมีการเปลี่ยนความยาวของเฟรมวีดิทัศน์



รูปที่ 4.5 อัตราเฟรมของวีดิทัศน์ที่ได้รับเมื่อมีการเปลี่ยนความยาวของเฟรมวีดิทัศน์

จากการทดลองเกี่ยวกับการปรับอัตราบิตทั้งสองตอน สามารถสรุปได้ว่าการลดอัตราบิตในการส่งวีดิทัศน์ไม่ได้ทำให้คุณภาพของวีดิทัศน์ดีขึ้นได้เสมอไป และจากการทดลองพบว่าการลดอัตราบิตกลับทำให้ให้อัตราการสูญหายของแพ็กเก็ตเพิ่มขึ้นด้วยซ้ำ และไม่ได้ทำให้คุณภาพของวีดิทัศน์ดีขึ้นได้ ดังนั้นในการควบคุมคุณภาพแบบปรับตัวสำหรับการสื่อสารวีดิทัศน์จึงควรจะใช้แนวทางอื่น เช่น การพิจาณาว่าการปรับพารามิเตอร์ค่าใดบ้างที่จะสามารถช่วยลดผลกระทบจากการสูญหายของแพ็กเก็ตและช่วยให้คุณภาพของวีดิทัศน์ดีขึ้นได้ แทนที่จะพิจารณาเฉพาะการปรับค่าอัตราบิตเพียงอย่างเดียว

4.4 การควบคุมคุณภาพวีดิทัศน์ด้วยวิธีการปรับอัตราบิต

มีงานวิจัยเกี่ยวกับเทคนิคการปรับตัวของการสื่อสารวีดิทัศน์ที่ได้รับการเสนอก่อนหน้านี้หลายงานที่ใช้วิธีการปรับอัตราบิตของการส่งวีดิทัศน์ โดยมีการลดอัตราบิตเมื่อเครือข่ายคับคั่ง แต่ในการทดลองในหัวข้อที่ 4.3 ได้ชี้ให้เห็นแล้วว่าการลดอัตราบิตของวีดิทัศน์ไม่ได้ช่วยให้คุณภาพของวีดิทัศน์ดีขึ้น และเพื่อที่จะชี้ให้เห็นถึงจุดบกพร่องของการควบคุมคุณภาพวีดิทัศน์ด้วยการปรับอัตราบิต ในหัวข้อนี้จึงได้มีการทดลองเพื่อประเมินผลวิธีการดังกล่าว โดยเลือกอัลกอริทึมการปรับอัตราบิตแบบ Adaptive Increase Multiplicative Decrease (AIMD)[7] เป็นกรณีศึกษา ซึ่งอัลกอริทึมนี้ใช้ค่าอัตราการสูญหายของแพ็กเก็ตในการตัดสินใจเพิ่มหรือลดอัตราบิตวีดิทัศน์ โดยมีหลักการคือ ถ้ามีแพ็กเก็ตสูญหายมากเกินไปก็ให้ลดค่าอัตราบิตลงโดยใช้การลดแบบผลคูณ และถ้ามีแพ็กเก็ตสูญหายน้อยมาก แสดงว่าความคับคั่งของเครือข่ายน้อยก็ให้เพิ่มอัตราบิตแบบผลบวก ค่าอัตราการสูญหายของแพ็กเก็ตนี้ได้มาจากผู้รับซึ่งจะรายงานด้วยแพ็กเก็ต RTCP ชนิด Receiver Report เป็นระยะ ๆ เพื่อรายงานสถิติของการรับ ซึ่ง Pseudo Code ของอัลกอริทึมนี้แสดงอยู่ในรูปที่ 4.6 จะเห็นว่าใน Pseudo Code มีการใช้สัญลักษณ์แทนพารามิเตอร์อยู่หลายค่า ซึ่งคำอธิบายของพารามิเตอร์เหล่านี้ได้แสดงไว้ในตารางที่ 4.1

```
For each receiver report packet received do

//low-pass filter equation for loss rate
λ = (1-α)*λ + α*b

if (λ > λc)  //congested state, decrease bit rate
    br = max(br*μ, bmin)
else if (λ > λu)  //underloaded state, increase bit rate
    br = min(br+ν, bmax)
```

รูปที่ 4.6 Pseudo Code ของอัลกอริทึม AIMD

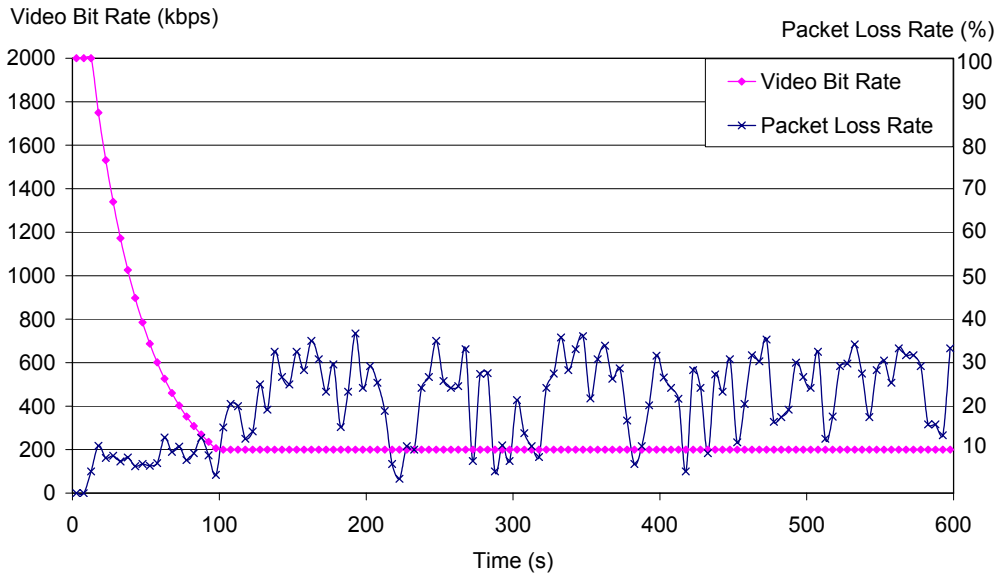
ตารางที่ 4.1 คำอธิบายพารามิเตอร์ในอัลกอริทึม AIMD

พารามิเตอร์	คำอธิบาย
λ	ค่าอัตราการสูญหายของแพ็กเก็ตที่ผ่านตัวกรองแล้ว
b	ค่าอัตราการสูญหายของแพ็กเก็ตที่ได้จากแพ็กเก็ต RTCP แพ็กเก็ตล่าสุด
α	เป็นค่าที่ใช้ในการกำหนดว่าจะให้ค่า b มีผลมากหรือน้อย โดยที่ $0 < \alpha < 1$
λ_c	ค่าเทรสโฮลด์ของ λ ที่ใช้ในการลดอัตราบิตวีดิทัศน์
λ_u	ค่าเทรสโฮลด์ของ λ ที่ใช้ในการเพิ่มอัตราบิตวีดิทัศน์
b_r	อัตราบิตของวีดิทัศน์
μ	ตัวคูณที่ใช้ในการลดอัตราบิตของวีดิทัศน์
ν	ค่าที่ใช้ในการเพิ่มอัตราบิตของวีดิทัศน์
b_{min}	อัตราบิตต่ำสุด
b_{max}	อัตราบิตสูงสุด

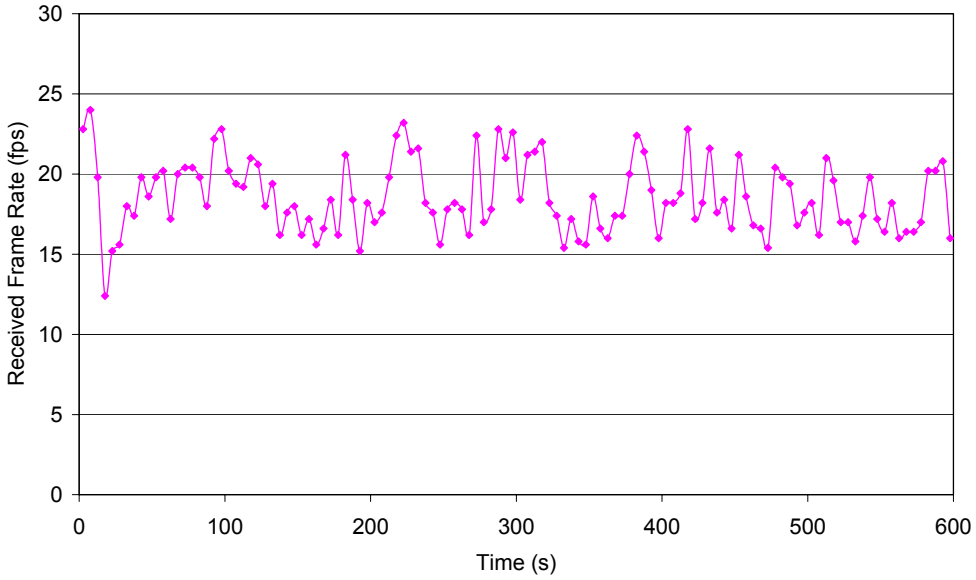
เพื่อเป็นการประเมินผลอัลกอริทึม AIMD ในหัวข้อนี้จึงได้ทำการทดลองโดยการจำลองบน NS-2 โดยใช้เครือข่ายในการจำลองดังรูปที่ 4.1 โดยข้อกำหนดต่าง ๆ ในการจำลองมีดังนี้

- จำนวนโหนดที่ส่งทราฟฟิก TCP เท่ากับ 200 โหนด (T_{Sl1} – TS_{l100} และ TS_{s1} – TS_{s100}) และเริ่มส่งที่วินาทีที่ 10
- โหนด RS และโหนด US เริ่มส่งทราฟฟิกเมื่อเริ่มการจำลอง
- ทุกโหนดหยุดส่งทราฟฟิกที่วินาทีที่ 600
- ค่าของพารามิเตอร์ในอัลกอริทึม AIMD มีดังนี้ α = 0.3, μ = 0.875, ν = 50 กิโลบิตต่อวินาที, λ_c = 4% และ λ_u = 2% ซึ่งค่าของพารามิเตอร์ทั้ง 5 ค่านี้เป็นไปตามค่าแนะนำใน [7], ส่วนค่า b_{min} = 200 กิโลบิตต่อวินาที และค่า b_{max} = 2000 กิโลบิตต่อวินาที
- อัลกอริทึม AIMD นั้นกล่าวถึงเฉพาะการปรับอัตราบิต แต่สำหรับการส่งวีดิทัศน์ที่ใช้การบีบอัดด้วย MPEG-4 จำเป็นจะต้องกำหนดค่าระยะห่างระหว่างเฟรมหลักด้วย ซึ่งในที่นี้กำหนดให้มีค่าเท่ากับ 1 เฟรม นั่นคือวีดิทัศน์ทุกเฟรมเป็นเฟรม I

จากกราฟในรูปที่ 4.7 ซึ่งแสดงให้เห็นถึงการปรับอัตราบิตของการส่งวีดิทัศน์และอัตราการสูญหายของแพ็กเก็ตวีดิทัศน์ โดยการปรับอัตราบิตนั้นเริ่มขึ้นหลังจากวินาทีที่ 10 เนื่องจากเป็นเวลาที่เริ่มมีการส่งทราฟฟิก TCP เข้าในเครือข่าย แต่ไม่ว่าจะลดอัตราบิตลงเท่าใดก็ไม่สามารถที่จะทำให้อัตราการสูญหายของแพ็กเก็ตลดต่ำลงกว่า 4% (λ_c) ได้ และอัลกอริทึม AIMD ก็ได้ลดอัตราบิตลงเรื่อยจนกระทั่งถึงค่าอัตราบิตต่ำสุด (200 กิโลบิตต่อวินาที)



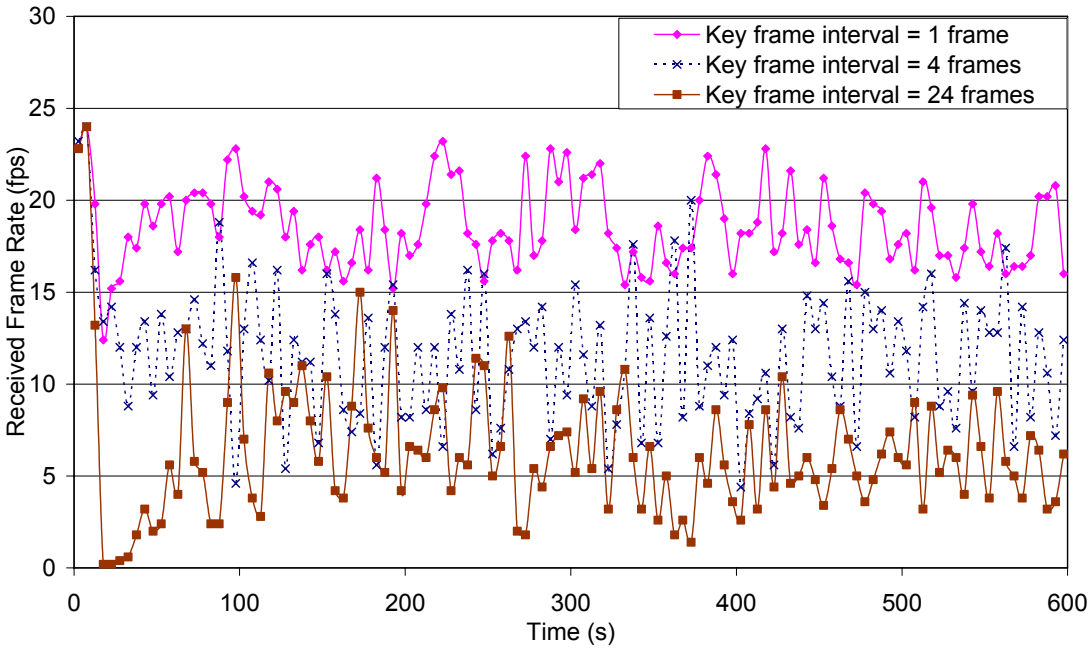
รูปที่ 4.7 อัตราบิตของวีดิทัศน์และอัตราการสูญหายของแพ็กเก็ตในระหว่างการจำลอง



รูปที่ 4.8 อัตราเฟรมฝั่งรับในระหว่างการจำลอง (ระยะห่างระหว่างเฟรมหลัก = 1 เฟรม)

เมื่อพิจารณากราฟในรูปที่ 4.8 จะเห็นว่าอัตราเฟรมฝั่งรับอยู่ในระดับที่ยอมรับได้ นั่นคือส่วนใหญ่แล้วอัตราเฟรมฝั่งรับจะสูงกว่า 15 เฟรมต่อวินาที แต่การที่ผลมาออกมาเช่นนี้ก็เนื่องมาจากว่าในการทดลองนี้ได้กำหนดค่าระยะห่างระหว่างเฟรมหลักให้เท่ากับ 1 เฟรม ซึ่งเป็นค่าที่ทำให้การส่งวีดิทัศน์ทนต่อการสูญหายของแพ็กเก็ตได้ดีที่สุดเพราะทุกเฟรมเป็นเฟรม I ถ้าหากใช้ค่าระยะห่างระห่างเฟรมหลักค่าอื่นอาจจะได้ผลที่แตกต่างออกไป

จากกราฟในรูปที่ 4.9 ซึ่งแสดงค่าอัตราเฟรมฝั่งรับในกรณีที่กำหนดค่าระยะห่างระหว่างเฟรมหลักเท่ากับ 1, 4 และ 24 เฟรม จะเห็นว่าในกรณีที่กำหนดค่าระยะห่างระหว่างเฟรมหลักเท่ากับ 4 เฟรม ค่าอัตราเฟรมฝั่งรับส่วนใหญ่จะต่ำกว่า 15 เฟรมต่อวินาที และจะยิ่งต่ำลงไปอีกในกรณีที่ใช้ค่าระยะห่างระหว่างเฟรมหลักเท่ากับ 24 เฟรม ดังนั้นการควบคุมคุณภาพวีดิทัศน์โดยใช้วิธีการปรับอัตราบิตเพียงอย่างเดียวจึงไม่เพียงพอ จำเป็นต้องมีการพิจารณาด้วยว่าควรจะใช้ค่าระยะห่างระหว่างเฟรมหลักเท่าใดด้วย



รูปที่ 4.9 อัตราเฟรมฝั่งรับเมื่อกำหนดระยะห่างระหว่างเฟรมหลักต่างกัน

ถึงแม้ว่ากราฟในรูปที่ 4.9 จะแสดงให้เห็นว่าถ้าหากกำหนดค่าของระยะห่างระหว่างเฟรมหลักให้คงที่เท่ากับ 1 เฟรม วิธีการปรับอัตรบิตแบบ AIMD สามารถควบคุมอัตราเฟรมฝั่งรับให้สูงกว่า 15 เฟรมต่อวินาทีได้ แต่การที่จะกำหนดค่าของระยะห่างระหว่างเฟรมหลักเท่ากับ 1 เฟรมตลอดนั้นดูจะไม่เหมาะสมเท่าใดนัก เพราะในกรณีที่เครือข่ายมีความคับคั่งต่ำก็ควรที่จะเพิ่มค่าระยะห่างระหว่างเฟรมหลักเพื่อจะได้ทำให้วีดิทัศน์มีความชัดเจนมากขึ้น เนื่องจากว่าในกรณีที่กำหนดอัตราบิตเป้าหมายเท่ากัน หากเพิ่มค่าระยะห่างระหว่างเฟรมหลักจะทำให้มีการตัดข้อมูลออกน้อยลงส่งผลให้วีดิทัศน์มีความชัดเจนเพิ่มขึ้น นอกจากนี้การที่กำหนดให้ทุกเฟรมเป็นเฟรม I นั้น ถือเป็นการลดความสามารถของการบีบอัดวีดิทัศน์ด้วย MPEG-4 ซึ่งเป็นการบีบอัดที่สามารถลดความซ้ำซ้อนระหว่างเฟรมได้ แต่การกำหนดให้ทุกเฟรมเป็นเฟรม I จะทำให้ตัวเข้ารหัสไม่สามารถลดความซ้ำซ้อนระหว่างเฟรมได้เลย

จากผลการทดลองในหัวข้อนี้ สามารถสรุปได้ว่าการปรับอัตราบิตด้วยอัลกอริทึม AIMD ยังไม่สามารถที่จะใช้ในการควบคุมคุณภาพของวีดิทัศน์ได้ ในขณะที่เครือข่ายคับคั่งการลดอัตราบิตของส่งวีดิทัศน์ไม่ได้ทำให้อัตราการสูญหายของแพ็กเก็ตลดลง และไม่ได้ทำให้อัตราเฟรมฝั่งรับสูงขึ้นแต่อย่างใด และสุดท้ายก็ต้องลดอัตราบิตลงเรื่อย ๆ จนถึงค่าอัตราบิตต่ำสุด ทั้งนี้เนื่องจากว่าเมื่อมีการลดอัตราบิตของวีดิทัศน์ โหนดอื่นที่ส่งทราฟฟิก TCP ก็พยายามที่จะใช้แบนด์วิดท์ส่วนที่เหลือเช่นกัน ความคับคั่งของเครือข่ายจึงไม่ได้น้อยลง นอกจากนี้อัลกอริทึม AIMD มีการปรับเฉพาะอัตราบิตเท่านั้น โดยไม่ได้คำนึงถึงพารามิเตอร์ที่สำคัญอีกค่าหนึ่งคือระยะห่างระห่างเฟรมหลัก จากการทดลองจะเห็นว่าการปรับค่าระยะห่างระหว่างเฟรมหลักมีผลต่ออัตราเฟรมฝั่งรับมากกว่าการปรับอัตราบิต ดังนั้นในวิทยานิพนธ์นี้จึงเลือกที่ควบคุมคุณภาพวีดิทัศน์ด้วยการปรับระยะห่างระหว่างเฟรมหลักแทนการปรับอัตราบิตของการส่งวีดิทัศน์

4.5 การลดผลกระทบจากการสูญหายของแพ็กเก็ตในการสื่อสารวีดิทัศน์

การสูญหายของแพ็กเก็ตเป็นปัญหาที่แทบจะหลีกเลี่ยงไม่ได้สำหรับการสื่อสารวีดิทัศน์บนเครือข่ายอินเตอร์เน็ต ดังนั้นจึงจำเป็นที่จะต้องมีกลไกที่จะช่วยลดผลกระทบจากการสูญหายของแพ็กเก็ต และเนื่องจากว่าในการส่งวีดิทัศน์นั้น วีดิทัศน์ 1 เฟรมอาจถูกตัดแบ่งเพื่อบรรจุลงในหลายแพ็กเก็ต หากลดจำนวนแพ็กเก็ตที่ใช้ต่อวีดิทัศน์ 1 เฟรมลง จะช่วยลดโอกาสที่เฟรมวีดิทัศน์จะเสียหายได้ เนื่องจากการการสูญหายของแพ็กเก็ตเพียง 1 แพ็กเก็ตอาจทำให้วีดิทัศน์เสียหายได้ทั้งเฟรม ซึ่งการลดจำนวนแพ็กเก็ตที่ใช้ต่อวีดิทัศน์ 1 เฟรมสามารถทำได้ 3 วิธีคือ การเพิ่มขนาดเพย์โหลด (Payload) ของแต่ละแพ็กเก็ต การลดความยาวของเฟรมวีดิทัศน์หลังการบีบอัด และการลดจำนวนเฟรมที่ถอดรหัสผิดพลาดซึ่งจะเกิดได้กับเฟรม P ในกรณีที่เฟรมก่อนหน้ามีการสูญหาย

4.5.1 การเพิ่มขนาดเพย์โหลดของแพ็กเก็ต

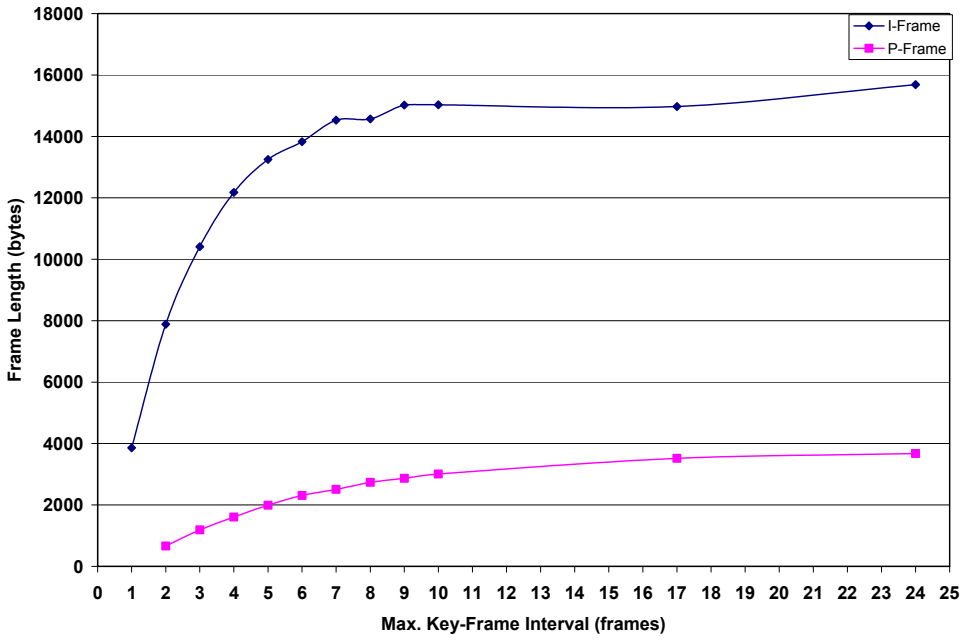
การเพิ่มขนาดเพย์โหลดของแพ็กเก็ตสามารถลดจำนวนแพ็กเก็ตที่ใช้ต่อวีดิทัศน์ 1 เฟรมได้ ตัวอย่างเช่น หากความยาวเฟรมวีดิทัศน์หลังการบีบอัดเท่ากับ 2000 ไบต์ ถ้าใช้เพย์โหลดขนาด 1000 ไบต์จะต้องใช้ 2 แพ็กเก็ต แต่ถ้าใช้เพย์โหลดขนาด 500 ไบต์จะใช้ต้องใช้แพ็กเก็ตถึง 4 แพ็กเก็ต ดังนั้นจึงควรกำหนดขนาดเพย์โหลดของแพ็กเก็ตให้มีขนาดใหญ่ที่สุดเท่าที่จะเป็นไปได้ ตัวอย่างเช่น ถ้าหาก MTU ของลิงค์มีค่าเท่ากับ 1500 ไบต์ ขนาดของแพ็กเก็ตที่ใหญ่ที่สุดที่เป็นไปได้สามารถคำนวณได้โดยนำค่า MTU ลบด้วยขนาดเฮดเดอร์ของ RTP (12 ไบต์), UDP (8 ไบต์) และ IP (20 ไบต์) ได้เป็นขนาดเพย์โหลดเท่ากับ 1460 ไบต์

4.5.2 การลดความยาวของเฟรมวีดิทัศน์หลังการบีบอัด

การลดความยาวเฟรมวีดิทัศน์หลังการบีบอัดทำได้โดยการตัดข้อมูลออกไปบางส่วนซึ่งทำให้ความชัดเจนของภาพมีน้อยลง แต่ผลดีก็คือทำให้จำนวนแพ็กเก็ตที่ใช้ต่อวีดิทัศน์ 1 เฟรมมีจำนวนน้อยลงตามไปด้วย ซึ่งการลดความยาวเฟรมวีดิทัศน์ในกรณีที่ใช้การบีบอัดด้วย MPEG-4 สามารถทำได้โดยการเพิ่มค่าระดับการควอนไทซ์ โดยหากใช้การบีบอัดที่ไม่มีการควบคุมอัตราบิตก็สามารถปรับค่าระดับการควอนไทซ์ได้โดยตรง แต่ในกรณีที่ใช้การบีบอัดแบบควบคุมอัตราบิตที่มีการปรับค่าระดับการควอนไทซ์โดยอัตโนมัติ สามารถลดค่าระดับการควอนไทซ์ได้โดยการปรับที่พารามิเตอร์ตัวอื่น เช่น เพิ่มอัตราเฟรม หรือลดระยะห่างระหว่างเฟรมหลัก เป็นต้น

4.5.3 การลดจำนวนเฟรมที่ถอดรหัสผิดพลาด

ในการสื่อสารวีดิทัศน์ที่ใช้การบีบอัดแบบ MPEG-4 สำหรับวีดิทัศน์บางเฟรมแล้ว ถึงแม้จะได้รับแพ็กเก็ตของเฟรมนั้นครบทุกแพ็กเก็ตก็อาจจะไม่สามารถถอดรหัสเฟรมดังกล่าวได้อย่างถูกต้อง ตัวอย่างเช่น เฟรม P ซึ่งต้องใช้ข้อมูลของเฟรมที่อยู่ก่อนหน้า การสูญหายของเฟรมวีดิทัศน์ 1 เฟรมทำให้เฟรม P ทั้งหมดที่อยู่หลังจากเฟรมที่สูญหาย เกิดข้อผิดพลาดในการถอดรหัส และการถอดรหัสของผู้รับจะกลับมาทำงานได้อย่างถูกต้องอีกครั้งเมื่อได้รับเฟรม I ดังนั้นการลดจำนวนเฟรมที่ถอดรหัสผิดพลาดทำได้โดยการลดระยะห่างระหว่างเฟรม I



รูปที่ 4.10 ความยาวของเฟรมหลังการบีบอัดเมื่อมีการเปลี่ยนระยะห่างระหว่างเฟรมหลัก

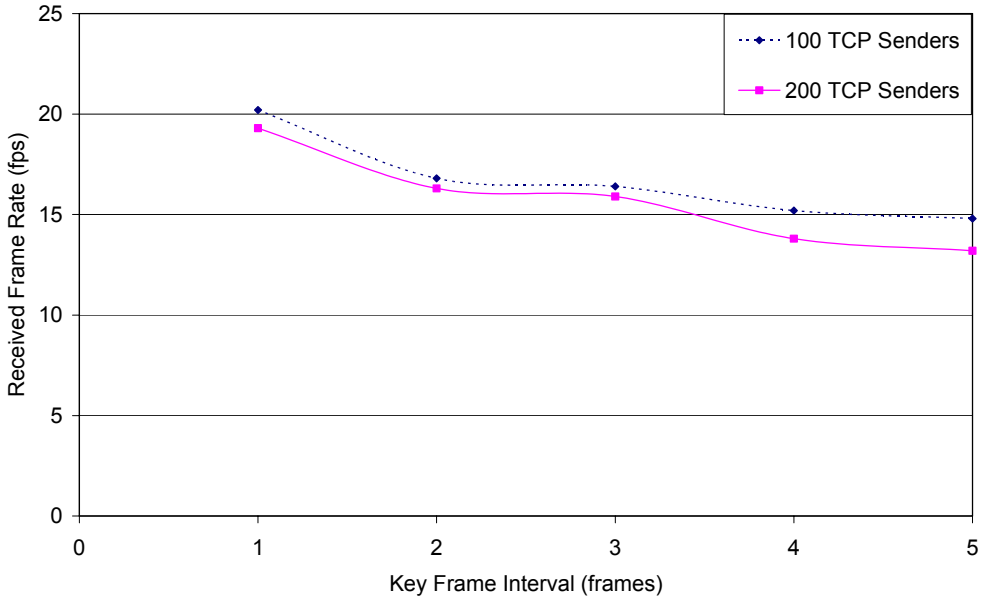
ในการบีบอัดวีดิทัศน์ด้วย MPEG-4 ที่มีการกำหนดอัตราบิตเป้าหมายคงที่ การลดค่าระยะห่างระหว่างเฟรมหลักสามารถลดผลกระทบจากการสูญหายของแพ็กเก็ตได้ เพราะนอกจากจะช่วยลดจำนวนเฟรมที่ถอดรหัสผิดพลาดได้แล้ว การลดค่าระยะห่างระหว่างเฟรมหลักยังทำให้ค่าระดับการควอนไทซ์เพิ่มขึ้น ส่งผลให้เฟรมวีดิทัศน์หลังการบีบอัดมีขนาดเล็กลง (ดูรายละเอียดเพิ่มเติมได้ในหัวข้อที่ 2.2.4) รูปที่ 4.10 เป็นกราฟแสดงความสัมพันธ์ระหว่างค่าระยะห่างระหว่างเฟรมหลักและความยาวของเฟรมวีดิทัศน์หลังการบีบอัด ซึ่งวัดจากการบีบอัดวีดิทัศน์ตัวอย่างขนาด 320x240 จุดภาพ อัตราเฟรม 24 เฟรมต่อวินาที และกำหนดอัตราบิตเป้าหมายเท่ากับ 800 กิโลบิตต่อวินาที จะเห็นว่าเมื่อลดค่าระยะห่างระหว่างเฟรมหลัก ความยาวของเฟรมวีดิทัศน์หลังการบีบอัดจะลดลงตามไปด้วยไม่ว่าจะเป็นเฟรม I หรือเฟรม P

และเพื่อเป็นการทดสอบว่าการลดระยะห่างระหว่างเฟรมหลักสามารถใช้ในการควบคุมคุณภาพของวีดิทัศน์ได้จริง จึงได้มีการทดลองโดยการจำลองบน NS-2 เช่นเดียวกับในหัวข้อที่ 4.3 การเชื่อมต่อภายในเครือข่ายที่ใช้ในการจำลองมีลักษณะเดียวกันกับเครือข่ายในรูปที่ 4.1 ขั้นตอนในการทดลองนี้แบ่งออกเป็น 2 รอบ โดยจำนวนโหนดที่ส่งทราฟฟิก TCP ในแต่ละรอบเป็นดังนี้

- รอบที่ 1 มีโหนดที่ส่ง TCP 100 โหนด (TS_{l1} – TS_{l50} และ TS_{sl1} – TS_{s50})
- รอบที่ 2 มีโหนดที่ส่ง TCP 200 โหนด (TS_{l1} – TS_{l100} และ TS_{s1} – TS_{s100})

โดยแต่ละรอบมีการจำลอง 5 ครั้งโดยแต่ละครั้งมีการเปลี่ยนค่าระยะห่างระหว่างเฟรมหลักจาก 1 – 5 เฟรม และบันทึกค่าเฉลี่ยของอัตราเฟรมฝั่งรับ ส่วนข้อกำหนดอื่น ๆ มีดังนี้ ขนาดของเพย์โหลดของทุกแพ็กเก็ตเท่ากับ 1460 ไบต์ การจำลองแต่ละครั้งกำหนดให้ทุกโหนดเริ่มส่งข้อมูลพร้อมกันวินาทีที่ 0 และใช้เวลาการจำลอง 600 วินาที สำหรับความยาวของเฟรมวีดิทัศน์แต่ละเฟรมจะไม่เท่ากัน เนื่องจากเป็นขนาดจริงที่ได้จากการบีบอัดวีดิทัศน์ตัวอย่างซึ่งมีขนาด 320x240 จุดภาพ อัตราเฟรมเท่ากับ 24 เฟรมต่อวินาที และบีบอัดที่อัตราบิตเป้าหมาย 800 กิโลบิตต่อวินาที

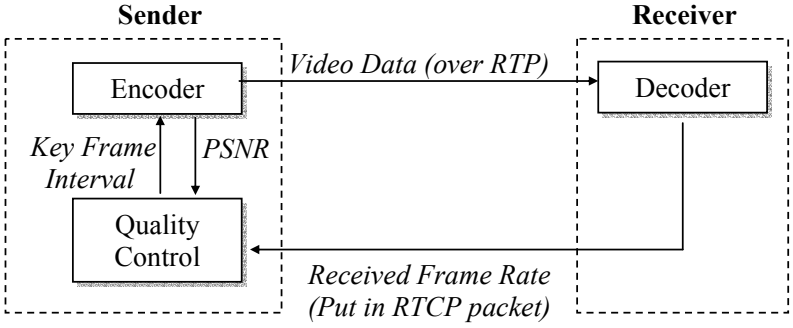
เมื่อพิจารณากราฟแสดงค่าอัตราเฟรมฝั่งรับในรูปที่ 4.11 ซึ่งค่าอัตราเฟรมในกราฟนี้ได้จากการคำนวณโดยนับเฉพาะเฟรมที่มีการถอดรหัสถูกต้องนั้น โดยไม่นับเฟรม P ที่ถอดรหัสผิดพลาด จะเห็นว่าเมื่อลดค่าระยะห่างระหว่างเฟรมหลักทำให้อัตราเฟรมฝั่งรับสูงขึ้น จากผลการทดลองสามารถสรุปได้ว่าในกรณีที่กำหนดอัตราเฟรม และอัตราบิตเป้าหมายคงที่ การลดค่าระยะห่างระหว่างเฟรมหลักสามารถช่วยให้อัตราเฟรมฝั่งรับเพิ่มขึ้นได้ แต่การลดค่าระยะห่างระหว่างเฟรมหลักในกรณีนี้ทำให้ความชัดเจนของวีดิทัศน์น้อยลง ดังนั้นจึงต้องมีอัลกอริทึมในการตัดสินใจว่าควรจะลดค่าระยะห่างระหว่างเฟรมหลักเท่าใดจึงจะเหมาะสม ซึ่งรายละเอียดของอัลกอริทึมดังกล่าวนี้อยู่ในหัวข้อถัดไป



รูปที่ 4.11 ค่าอัตราอัตราเฟรมฝั่งรับเมื่อมีการเปลี่ยนระยะห่างระว่างเฟรมหลัก

4.6 อัลกอริทึมในการปรับค่าระยะห่างระหว่างเฟรมหลัก

จากการทดลองในหัวข้อ 4.5 จะเห็นว่าการลดค่าระยะห่างระหว่างเฟรมหลักสามารถลดจำนวนเฟรมที่ถอดรหัสผิดพลาด และลดโอกาสที่เฟรมวีดิทัศน์จะสูญหายได้ ดังนั้นในวิทยานิพนธ์นี้จึงเลือกใช้วิธีการปรับค่าระยะห่างระหว่างเฟรมหลักในการควบคุมคุณภาพวีดิทัศน์ แต่ผลกระทบอย่างหนึ่งของการลดค่าระยะห่างระหว่างเฟรมหลักก็คือวีดิทัศน์มีความชัดเจนน้อยลงเนื่องจากมีการตัดข้อมูลออกมากขึ้น ดังนั้นจึงจำเป็นที่จะต้องมีอัลกอริทึมในพิจารณาว่าเมื่อใดควรที่จะลดค่าระยะห่างระหว่างเฟรมหลักและเมื่อใดควรจะเพิ่ม ในวิทยานิพนธ์นี้จึงได้เสนออัลกอริทึมในการปรับระยะห่างระหว่างเฟรมหลักขึ้น โดยพารามิเตอร์มิเตอร์ที่ใช้ในการตัดสินในว่าควรจะเพิ่มหรือลดค่าระยะห่างระหว่างเฟรมหลักคือ ค่าอัตราเฟรมฝั่งรับซึ่งคำนวณจากจำนวนเฟรมที่ถอดรหัสถูกต้องเท่านั้น



รูปที่ 4.12 แผนภาพการควบคุมคุณภาพวีดิทัศน์ด้วยการปรับค่าระยะห่างระหว่างเฟรมหลัก

จากแผนภาพในรูปที่ 4.12 จะเห็นว่าการควบคุมคุณภาพวีดิทัศน์ของผู้ส่งจะมีการรับค่าอัตราเฟรมฝั่งรับที่ผู้รับป้อนกลับมาให้ทางแพ็กเก็ต RTCP ชนิด Receiver Report รวมทั้งอ่านค่า PSNR จากตัวเข้ารหัสเพื่อใช้ในการปรับค่าระยะห่างระหว่างเฟรมหลักซึ่งเป็นพารามิเตอร์หนึ่งที่จะต้องกำหนดให้กับตัวเข้ารหัสวีดิทัศน์ในกรณีที่ใช้การบีบอัดแบบ MPEG-4

4.6.1 Pseudo Code

รูปที่ 4.13 เป็น Pseudo Code ของอัลกอริทึมที่ใช้ในการปรับค่าระยะห่างระหว่างเฟรมหลัก จะเห็นว่าในอัลกอริทึมนี้มีตัวแปรอยู่หลายตัว โดยตัวแปรแต่ละตัวมีความหมายดังนี้

I	ระยะห่างระหว่างเฟรมหลักที่ผู้ส่งใช้ในการเข้ารหัสวีดิทัศน์
I_c	ค่า I ที่ทำให้มีการเปลี่ยนแปลงความชัดเจนของภาพอย่างรวดเร็ว
F_{tran}	อัตราเฟรมที่ผู้ส่งใช้ในการส่งวีดิทัศน์ ในที่นี้กำหนดให้เป็น 24 เฟรมต่อวินาที ซึ่งเป็นอัตราเฟรมที่ใช้ในการฉายภาพยนตร์
F_{rcv}	อัตราเฟรมของวีดิทัศน์ที่ฝั่งผู้รับ โดยนับเฉพาะเฟรมที่ถอดรหัสถูกต้องเท่านั้น
F_{low}, F_{high}	ค่าเทรสโฮลด์ของอัตราเฟรมที่ใช้ในการตัดสินใจเพื่อปรับค่าระยะห่างระหว่างเฟรมหลัก
$PSNR_{avg}$	ค่า PSNR เฉลี่ยระหว่างเวลาที่ได้รับแพ็กเก็ต Receiver Report ในรอบก่อนหน้าและรอบปัจจุบัน โดยในที่นี้เลือกใช้ค่า PSNR ขององค์ประกอบ Y เนื่องจากเป็นองค์ประกอบที่มีความละเอียดมากที่สุด และเมื่อมีการเปลี่ยนค่าระยะห่างระหว่างเฟรมหลัก ค่า PSNR ขององค์ประกอบ Y จะมีการเปลี่ยนแปลงมากที่สุด
MIN_PSNR	ค่าเทรสโฮลด์ของ $PSNR_{avg}$ ซึ่งเป็นค่าที่ระบุว่าความชัดเจนของวีดิทัศน์มีน้อยเกินไป ในที่นี้กำหนดให้เป็น 30 dB เนื่องจากว่าโดยทั่วไปแล้วภาพที่มีค่า PSNR ตั้งแต่ 30 dB ถือว่ามีความชัดเจนอยู่ในระดับที่ยอมรับได้[39][40][41][42]
N_h	จำนวนครั้งที่ F_{rcv} มากกว่า F_{high}
MIN_DURATION	ค่าเทรสโฮลด์ของ N_h ที่ใช้ในการตัดสินใจเพิ่มค่า I เพื่อลดการแกว่งของค่า I

หลักการของอัลกอริทึมนี้ก็คือ ถ้าอัตราเฟรมฝั่งรับมีค่าต่ำเกินไป ($F_{rcv} < F_{low}$) ผู้ส่งก็จะลดค่า I ลงเพื่อช่วยให้อัตราเฟรมฝั่งรับสูงขึ้น ยกเว้นในกรณีที่ค่า PSNR เฉลี่ยมีค่าต่ำเกินไป ผู้ส่งจะไม่ลดค่า I เนื่องจากว่าการที่ค่า PSNR ต่ำกว่า MIN_PSNR แสดงวีดิทัศน์มีความชัดเจนน้อย

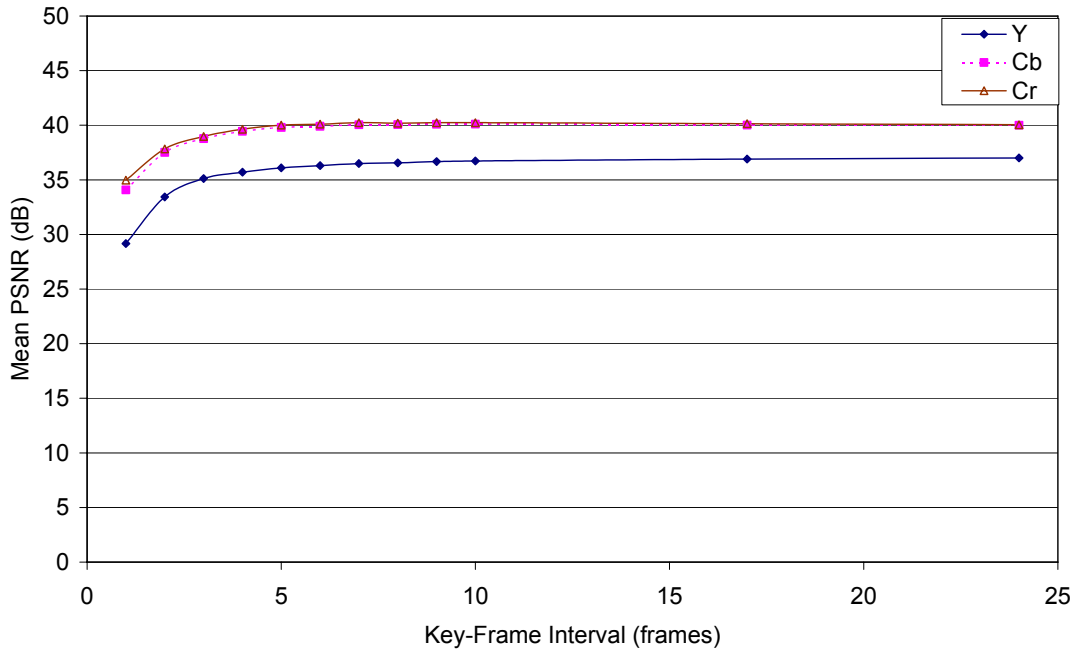
เกินไปแล้ว การได้รับวีดิทัศน์ที่มีอัตราเฟรมสูงก็ไม่มีประโยชน์ถ้ามองไม่ออกว่าภาพในแต่ละเฟรมเป็นภาพอะไร ส่วนในกรณีที่อัตราเฟรมฝั่งรับสูงเพียงพอแล้ว ($F_{rcv} > F_{high}$) ผู้ส่งก็จะเพิ่มค่า I เพื่อให้วีดิทัศน์มีความชัดเจนมากขึ้น แต่เพื่อลดการแกว่งของค่า I จึงมีการนับจำนวนครั้งที่ F_{rcv} มีค่ามากกว่าเทรสโฮลด์ F_{high} ซึ่งก็คือค่า N_h โดยอัลกอริทึมนี้จะเพิ่มค่า I ก็ต่อเมื่อค่า N_h เท่ากับ MIN_DURATION เท่านั้น ในที่นี้เลือกที่จะใช้ค่า MIN_DURATION เท่ากับ 4 ซึ่งหมายถึงระยะเวลาเท่ากับการรับแพ็กเก็ต Receiver Report จำนวน 4 ครั้ง และเนื่องจากว่าผู้ส่งจะได้รับแพ็กเก็ตนี้ทุก 5 วินาที ดังนั้นระยะเวลาของ MIN_DURATION จึงเท่ากับ 20 วินาที โดยค่านี้มีที่มาจากการทดลองโดยการเปลี่ยนค่าระยะห่างระหว่างเฟรมหลักสลับไปมาระหว่าง 1 และ 2 เฟรม พบว่าค่าช่วงเวลาที่น้อยที่สุดที่ไม่ทำให้เกิดความรำคาญต่อสายตาคือ 20 วินาที

```
When receive each RTCP receiver report:

Frcv = The value received frame rate in the RTCP extension
       Header.
if((Frcv < Flow) AND (PSNRavg > MIN_PSNR))
       //Decrement key-frame interval
       if(I > Ic)
              I = Ic
       else if(I > 1)
              I = I - 1
       Nh = 0

else if(Frcv > Fhigh)
       Nh = Nh+1
else
       Nh = 0
if(Nh == MIN_DURATION)
       //Increment key-frame interval
       if(I < Ic)
              I = I + 1
       else if(I < Ftran)
              I = Ftran  //key frame interval = 1 seconds
       Nh = 0
```

รูปที่ 4.13 อัลกอริทึมในการปรับค่าระยะห่างระหว่างเฟรมหลัก

ค่า F_{low} คือค่าต่ำสุดของอัตราเฟรมที่ฝั่งผู้รับ (F_{rcv}) ที่ยังสามารถยอมรับได้ ถ้าค่า F_{rcv} ต่ำกว่านี้แสดงว่าค่าระยะห่างระหว่างเฟรมหลัก (I) ที่ใช้อยู่ปัจจุบันนั้นมีค่ามากเกินไป จำเป็นจะต้องลดค่า I ลง ซึ่งในที่นี้ได้เลือกใช้ F_{low} เท่ากับ 15 เฟรมต่อวินาที ซึ่งเป็นค่าอัตราเฟรมที่ใช้ในกันมากในถ่ายทอดวีดิทัศน์ (Video Streaming) บนเครือข่ายอินเตอร์เน็ต [30][31][32] วีดิทัศน์ที่มีอัตราเฟรมต่ำกว่า 15 เฟรมต่อวินาทีจะทำให้ผู้ใช้เริ่มรู้สึกถึงความไม่ต่อเนื่องของวีดิทัศน์ ส่วนค่าเทรสโฮลด์ค่าที่สองคือ F_{high} หมายถึงค่าอัตราเฟรมที่ใช้ในการระบุว่าเครือข่ายมีความคับคั่งไม่

มากนัก และมีความปลอดภัยเพียงพอที่จะลองเพิ่มค่า I เพื่อให้ได้วีดิทัศน์ในแต่ละเฟรมมีคุณภาพดีขึ้น ซึ่งหากต้องการความแน่ใจว่าเครือข่ายไม่มีความคับคั่งแล้ว ผู้รับจะต้องได้รับวีดิทัศน์ด้วยอัตราเฟรมเท่ากับอัตราเฟรมที่ส่ง (F_{tran}) ซึ่งเท่ากับ 24 เฟรมต่อวินาที แต่หากกำหนด F_{high} เท่ากับ F_{tran} โอกาสที่จะมีการลองเพิ่มค่า I ก็จะมีน้อย ดังนั้นในที่นี้จึงเลือกกำหนดค่า F_{high} เท่ากับ 20 เฟรมต่อวินาที เนื่องจากวีดิทัศน์ที่มีอัตราเฟรม 20 เฟรมต่อวินาที มีความราบรื่นแตกต่างจาก 24 เฟรมต่อวินาทีไม่มากนัก

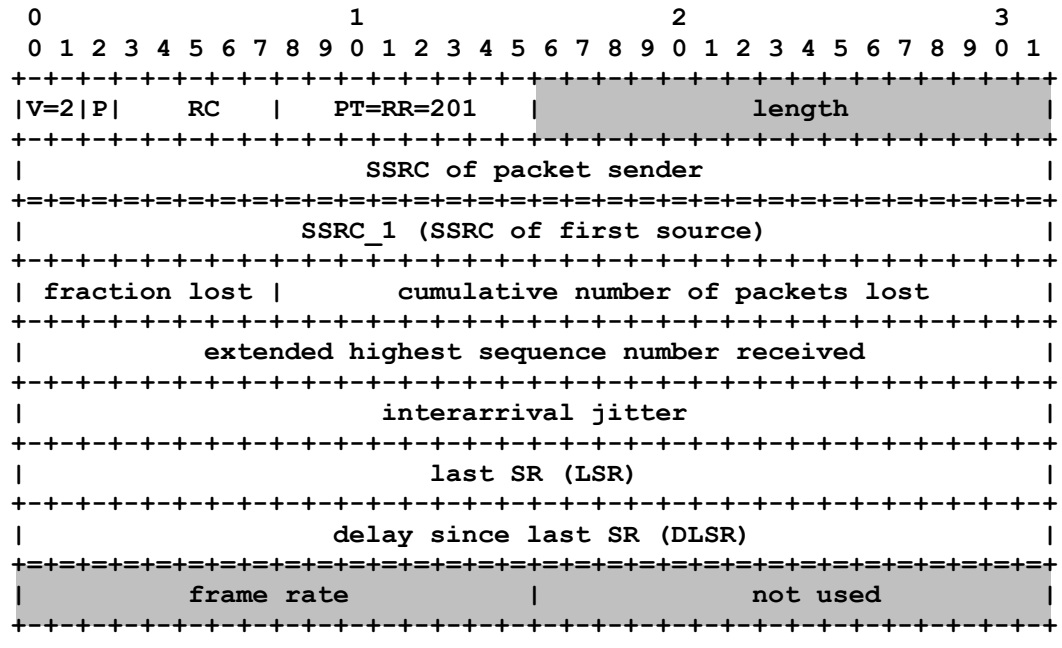


รูปที่ 4.14 ความสัมพันธ์ระหว่างระยะห่างระหว่างเฟรมหลักและค่า PSNR เฉลี่ย

การเพิ่มหรือลดค่า I ของอัลกอริทึมนี้ไม่ได้มีการเปลี่ยนแปลงทีละ 1 เฟรมเสมอไป ทั้งนี้เนื่องจากว่าการเปลี่ยนแปลงค่า I ในบางช่วงจะมีผลต่อความชัดเจนของภาพน้อยมาก ดังนั้นจึงจำเป็นต้องมีค่า I_c เพื่อกำหนดว่าในช่วงที่ I มากกว่า I_c ก็ไม่จำเป็นต้องเปลี่ยนแปลงค่า I ทีละ 1 เฟรม รูปที่ 4.14 แสดงให้เห็นถึงความสัมพันธ์ระหว่างเฟรมหลักและค่าเฉลี่ยของ PSNR ในองค์ประกอบ Y, Cb และ Cr ซึ่งค่า PSNR นี้เป็นค่าที่บอกถึงความชัดเจนของภาพในแต่ละเฟรมวีดิทัศน์ ถ้าค่า PSNR มากแสดงว่าเฟรมวีดิทัศน์ที่ส่งให้กับผู้รับนั้นมีคุณภาพใกล้เคียงกับวีดิทัศน์ต้นฉบับมาก นั่นคือมีการตัดข้อมูลออกไปเพียงเล็กน้อยนั่นเอง จะเห็นว่าในช่วงที่ใช้ค่าระยะห่างระหว่างเฟรมหลักมากกว่า 5 ค่า PSNR จะมีความแตกต่างกันน้อยมาก แต่ถ้าใช้ค่าระยะห่างระหว่างเฟรมหลักตั้งแต่ 5 เฟรมลงมา ค่า PSNR มีความแตกต่างกันอย่างชัดเจน ดังนั้นในที่นี้จึงเลือกที่จะใช้ค่า I_cเท่ากับ 5 เฟรม

4.6.2 การขยายเฮดเดอร์ RTCP

จากอัลกอริทึมในการปรับค่าระยะห่างระหว่างเฟรมหลัก จะเห็นว่าผู้รับจะต้องมีการป้อนกลับค่าอัตราเฟรมของวีดิทัศน์ที่ได้รับ โดยการบรรจุลงในแพ็กเก็ต Receiver Report ด้วย ซึ่งค่านี้เป็นข้อมูลที่ไม่ได้มีอยู่ในฟิลด์มาตรฐานของแพ็กเก็ต Receiver Report ที่กำหนดไว้ใน RFC 3550 [10] ดังนั้นจึงจำเป็นจะต้องเพิ่มข้อมูลนี้ลงในส่วนขยายเฮดเดอร์ RTCP ดังรูปที่ 4.15 โดยส่วนแรกเงา 4 ไบต์ล่างสุดคือส่วนขยายเฮดเดอร์ โดยมีการเพิ่มฟิลด์ที่ชื่อ frame rate ซึ่งมีขนาด 2 ไบต์ ฟิลด์นี้ใช้ในการใส่ค่าของอัตราของอัตราเฟรมวีดิทัศน์ที่ฝั่งผู้รับ (F_{rcv}) สำหรับสาเหตุที่ใช้ข้อมูลขนาด 2 ไบต์ก็เนื่องจากว่า ค่าอัตราเฟรมในการส่งวีดิทัศน์ที่ใช้กันอยู่ในปัจจุบันจะไม่เกิน 30 เฟรมต่อวินาที และเพื่อที่จะสามารถระบุความละเอียดได้โดยใช้ทศนิยม 2 ตำแหน่ง ดังนั้นค่าที่จะใส่ในฟิลด์ frame rate จึงต้องคูณ F_{rcv} ด้วย 100 เสียก่อน เมื่อต้องการอ่านค่า F_{rcv} ก็ให้นำค่าในฟิลด์ frame rate หารด้วย 100 ดังนั้นค่าสูงสุดของฟิลด์ frame rate คือ 3000 ซึ่งต้องใช้เนื้อที่ขนาด 2 ไบต์



รูปที่ 4.15 การเพิ่มค่าอัตราเฟรมในส่วนขยายเฮดเดอร์ของแพ็กเก็ต Receiver Report

และเมื่อมีการเพิ่มส่วนขยายเฮดเดอร์แล้ว จะต้องมีการแก้ไขค่าที่ฟิลด์ length ให้ถูกต้องด้วย โดยค่าในฟิลด์นี้สามารถคำนวณได้ดังนี้

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

และเนื่องจากว่าค่าของฟิลด์ length จะต้องมากกว่า 0 และต้องเป็นจำนวนเต็ม ดังนั้นจึงเป็นการบังคับไปในตัวว่าแพ็กเก็ต Receiver Report จะต้องมีขนาดอย่างน้อย 4 ไบต์ รวมทั้งขนาดของแพ็กเก็ตจะต้องหารด้วย 4 ลงตัว

4.7 การประเมินผลอัลกอริทึมในการปรับค่าระยะห่างระหว่างเฟรมหลัก

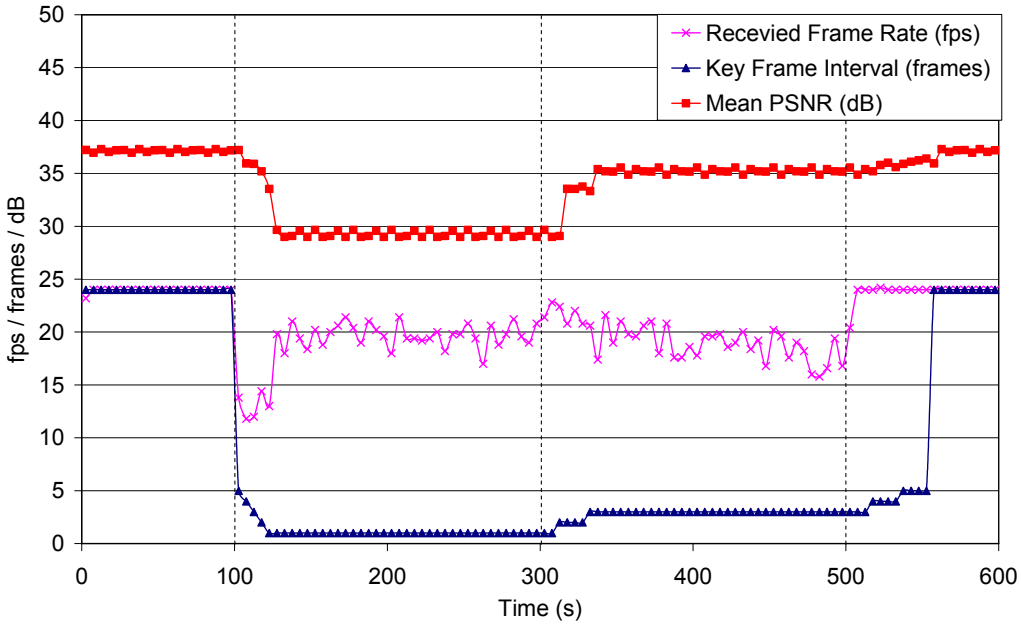
เพื่อเป็นการทดสอบว่าอัลกอริทึมในการปรับระยะห่างระหว่างเฟรมหลักที่ได้นำเสนอในหัวข้อ 4.6 สามารถกำหนดค่าระยะห่างระหว่างเฟรมหลักได้เหมาะสมกับสภาพเครือข่ายจริงหรือไม่ ในหัวข้อนี้จึงได้มีการทดลองโดยใช้การจำลองบน NS-2 และใช้รูปแบบของเครือข่ายในการจำลองในลักษณะเดียวกับเครือข่ายในรูปที่ 4.1 ในการทดลองนี้โหนด RS ส่งวีดิทัศน์โดยมีการใช้อัลกอริทึมในการปรับระยะห่างระหว่างเฟรมหลัก กำหนดให้มีจำนวนโหนดที่ส่ง TCP เท่ากับ 200 โหนด (TS_{l1} – TS_{l100} และ TS_{s1} – TS_{s100}) โดยแต่ละโหนดเริ่มส่งทราฟฟิกพร้อมกันแต่หยุดส่งไม่พร้อมกันดังนี้

- โหนด RS และ US เริ่มส่งทราฟฟิกที่วินาทีที่ 0 และหยุดส่งที่วินาทีที่ 600
- โหนดที่ส่ง TCP ทุกโหนด (TS_{l1} – TS_{l100} และ TS_{s1} – TS_{s100}) เริ่มส่งทราฟฟิกพร้อมกันที่วินาทีที่ 100
- โหนด TS_{l1} – TS_{l50} และ TS_{s1} – TS_{s50} หยุดส่งทราฟฟิกที่วินาทีที่ 300
- โหนด TS_{l51} – TS_{l100} และ TS_{s51} – TS_{s100} หยุดส่งทราฟฟิกที่วินาทีที่ 500

ทำให้ในแต่ละช่วงเวลามีจำนวนโหนดที่ส่งทราฟฟิก TCP ไม่เท่ากัน โดยในช่วงเวลาตั้งแต่วินาทีที่ 0 – 100 และวินาทีที่ 500 – 600 ไม่มีโหนดใดที่ส่งทราฟฟิก TCP เลย ส่วนในช่วงวินาทีที่ 100 – 300 มีโหนดที่ส่งทราฟฟิก TCP 200 โหนด และในช่วงวินาทีที่ 300 – 500 มีโหนดที่ส่งทราฟฟิก TCP จำนวน 100 โหนด ทั้งนี้เพื่อให้ระดับความคับคั่งของเครือข่ายไม่เท่ากัน และได้ตรวจสอบได้ว่าอัลกอริทึมในการปรับค่าระยะห่างระหว่างเฟรมหลักสามารถกำหนดค่าระยะห่างระหว่างเฟรมหลักได้เหมาะสมกับความคับคั่งของเครือข่ายในแต่ละระดับหรือไม่

และจากกราฟแสดงผลการทดลองในรูปที่ 4.16 จะเห็นว่าในช่วงวินาทีที่ 0 – 100 ซึ่งไม่มีโหนดที่ส่งทราฟฟิก TCP เลย ค่าระยะห่างระหว่างเฟรมหลักที่ใช้จึงเป็นค่าสูงสุดคือ 24 เฟรม แต่ในช่วงวินาทีที่ 100 – 300 เครือข่ายมีความคับคั่งสูงขึ้นเนื่องจากมีโหนดที่ส่งทราฟฟิก TCP จำนวน 200 โหนด ค่าระยะห่างระหว่างเฟรมหลักจึงถูกลดมาอยู่ที่ 1 เฟรม เพื่อรักษาระดับของค่าอัตราเฟรมฝั่งรับไม่ให้ต่ำกว่า 15 เฟรมต่อวินาที ส่วนในช่วงวินาทีที่ 300 – 500 ระดับความคับคั่งของเครือข่ายลดลงเล็กน้อยเนื่องจากจำนวนโหนดที่ส่งทราฟฟิก TCP ลดลงเหลือ 100 โหนด ค่าระยะห่างระหว่างเฟรมหลักจึงถูกปรับเพิ่มมาอยู่ที่ 3 เฟรม ส่วนช่วงเวลาตั้งแต่วินาทีที่

500 เป็นต้นไป ไม่มีโหนดใดส่งทราฟฟิก TCP เลย ค่าระยะห่างระหว่างเฟรมหลักจึงถูกปรับเพิ่มไปจนถึงค่าสูงสุดคือ 24 เฟรม ซึ่งโดยรวมแล้วอัลกอริทึมในการปรับค่าระยะห่างระหว่างเฟรมหลักสามารถกำหนดค่าระยะห่างระหว่างเฟรมหลักได้เหมาะสมกับสภาพของเครือข่าย และสามารถรักษาระดับของอัตราเฟรมฝั่งรับไม่ให้ต่ำกว่า 15 เฟรมต่อวินาทีได้



รูปที่ 4.16 การเปลี่ยนแปลงของระยะห่างระหว่างเฟรมหลัก อัตราเฟรมฝั่งรับ และ PSNR

และเมื่อพิจารณาถึงค่า PSNR เฉลี่ยในแต่ละช่วงเวลาซึ่งเป็นค่าที่บอกถึงความชัดเจนวีดิทัศน์ในแต่ละเฟรม พบว่าส่วนใหญ่แล้วค่า PSNR เฉลี่ยจะมีค่าสูงกว่าค่า MIN_PSNR (30 dB) ยกเว้นในช่วงเวลาประมาณวินาทีที่ 170 – 320 ซึ่งมีค่า PSNR เฉลี่ยอยู่ที่ 29 dB แต่ก็ไม่ได้มีผลต่อการตัดสินใจลดระยะห่างระหว่างเฟรมหลัก เนื่องจากช่วงเวลาดังกล่าวค่าระยะห่างระหว่างเฟรมหลักมีค่าเป็น 1 เฟรม ซึ่งเป็นค่าต่ำสุดอยู่แล้ว ไม่สามารถลดค่าต่ำลงกว่านี้ได้อีก

4.8 การกำหนดอัตราบิตเป้าหมายในการส่งวีดิทัศน์

จากการทดลองในหัวข้อ 4.6 จะเห็นว่าการควบคุมคุณภาพวีดิทัศน์โดยใช้อัลกอริทึมการปรับค่าระยะห่างระหว่างเฟรมหลักสามารถทำงานได้อย่างมีประสิทธิภาพ แต่ข้อจำกัดอย่างหนึ่งของวิธีการนี้ก็คือการใช้อัตราบิตเป้าหมายคงที่ ทำให้การส่งวีดิทัศน์ไม่สามารถปรับค่าอัตราบิตให้สอดคล้องกับแบนด์วิดท์ของลิงค์ได้ จึงเป็นหน้าที่ของผู้ใช้ที่จะต้องกำหนดอัตราบิตเป้าหมายในขั้นตอนของการบีบอัดวีดิทัศน์ให้เหมาะสม ซึ่งการกำหนดค่าอัตราบิตเป้าหมายอย่างง่ายที่สุดก็คือ

ใช้ค่าอัตราบิตที่ไม่เกินแบนด์วิดท์ของเครือข่าย แต่เนื่องจากว่าการสื่อสารแบบพหุสื่อในวิทยานิพนธ์นี้ประกอบด้วยการสื่อสารเสียงและวีดิทัศน์ และจากบทที่ 3 อัลกอริทึมที่ใช้ในการควบคุมคุณภาพเสียงคือ อัลกอริทึม CNR ซึ่งเป็นอัลกอริทึมที่มีพื้นฐานมาจาก FEC จึงมีโอกาสที่การส่งเสียงจะมีปริมาณการใช้แบนด์วิดท์มากขึ้น ดังนั้นในกรณีที่ผู้ส่งมีการส่งทั้งเสียงและวีดิทัศน์หากต้องการรักษาปริมาณการใช้แบนด์วิดท์ให้คงที่ เมื่อการส่งเสียงใช้แบนด์วิดเพิ่มขึ้นเนื่องจากมีการส่งข้อมูลซ้ำ การส่งวีดิทัศน์จะต้องลดอัตราบิตเป้าหมายลงเพื่อเป็นการชดเชย แต่อัตราบิตส่วนที่ลดนั้นมีค่าน้อยมากเมื่อเปรียบเทียบกับอัตราบิตของวีดิทัศน์ทำให้ไม่มีผลต่อความชัดเจนของภาพมากนัก

หากต้องการรักษาอัตราบิตรวมของการส่งเสียงและการส่งวีดิทัศน์ให้คงที่อยู่ที่ค่าหนึ่งซึ่งในที่นี้จะใช้สัญลักษณ์ R_{total} (ค่าอัตราบิตรวมนี้เป็นค่าอัตราบิตที่ได้รวมเฮดเดอร์ RTP, UDP และ IP แล้ว) สามารถคำนวณค่าอัตราบิตเป้าหมายที่จะต้องกำหนดให้กับตัวเข้ารหัสวีดิทัศน์ได้ตามขั้นตอนต่อไปนี้

1) เมื่ออัตราบิตการส่งเสียงมีการเปลี่ยนแปลงเนื่องจากการลดหรือเพิ่มจำนวนข้อมูลซ้ำ สามารถคำนวณอัตราบิตสำหรับการส่งวีดิทัศน์ได้ดังนี้

$$R_{vh} = R_{total} - R_{ah}$$

เมื่อ R_{vh} คืออัตราบิตของการส่งวีดิทัศน์ที่รวมเฮดเดอร์แล้ว ส่วน R_{ah} คืออัตราบิตของการส่งเสียงที่รวมเฮดเดอร์แล้ว

2) จากนั้นก็สามารถหาค่าอัตราบิตเป้าหมาย ที่ใช้สำหรับกำหนดให้กับตัวเข้ารหัสวีดิทัศน์วีดิทัศน์ได้ดังนี้

$$R_{target} = \frac{R_{vh} P}{(P + H)}$$

เมื่อ R_{target} คือ ค่าอัตราบิตเป้าหมาย, P คือค่าความยาวเพย์โหลดสูงสุดของแพ็กเก็ตวีดิทัศน์ และ H ผลรวมของความยาวของเฮดเดอร์ RTP, UDP และ IP

4.9 สรุป

เนื่องจากงานวิจัยเกี่ยวกับการปรับตัวของโปรแกรมประยุกต์เกี่ยวกับการสื่อสารวีดิทัศน์ก่อนหน้านี้ยังไม่สมบูรณ์พอที่จะใช้ในการควบคุมคุณภาพวีดิทัศน์ ดังนั้นในวิทยานิพนธ์นี้จึงได้เสนออัลกอริทึมในการควบคุมคุณภาพแบบปรับตัวสำหรับการสื่อสารวีดิทัศน์ขึ้นมาใหม่ โดยใช้วิธีการปรับระยะห่างระหว่างเฟรมหลัก และกำหนดให้ใช้อัตราเฟรมฝั่งส่งและอัตราบิตเป้าหมายคงที่ เนื่องจากว่าการลดอัตราบิตหรืออัตราเฟรมไม่ได้ช่วยให้คุณภาพของวีดิทัศน์ดีขึ้นเสมอไป

โดยเฉพาะในกรณีที่ปริมาณการใช้แบนด์วิดท์จนเต็มความจุเครือข่าย และจากผลการทดลองได้ชี้ให้เห็นว่าวิธีการดังกล่าวสามารถใช้ในการควบคุมคุณภาพวีดิทัศน์ได้ โดยสามารถควบคุมอัตราเฟรมฝั่งรับไม่ให้ต่ำกว่าค่าที่กำหนดได้ นอกจากนี้อัลกอริทึมในการควบคุมคุณภาพวีดิทัศน์ที่ได้เสนอขึ้นนี้ยังมีการพิจารณาถึงค่า PSNR อีกด้วย เพื่อไม่ให้มีการตัดข้อมูลของวีดิทัศน์ในแต่ละเฟรมมากเกินไป เพราะถึงแม้ผู้รับจะได้รับวีดิทัศน์ที่มีอัตราเฟรมสูงก็ไม่มีประโยชน์ถ้าข้อมูลของวีดิทัศน์ในแต่ละเฟรมถูกตัดออกจนผู้รับมองไม่ออกว่าเป็นภาพอะไร