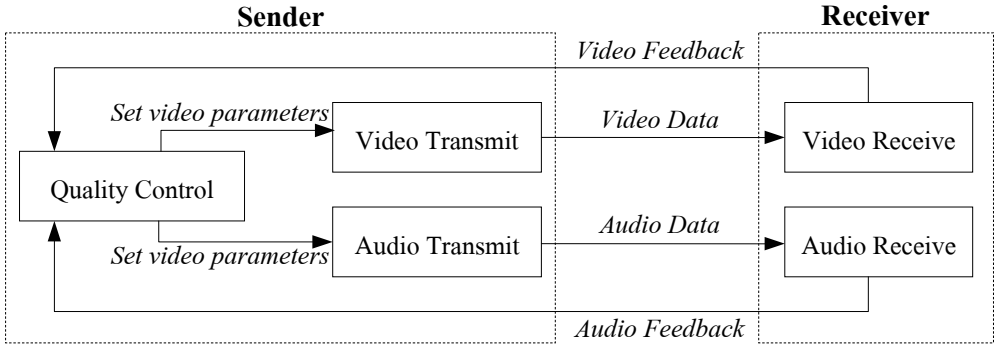
บทที่ 5

การพัฒนา API สำหรับการสื่อสารแบบพหุสื่อ

จากวิธีการควบคุมคุณภาพแบบปรับตัวสำหรับการสื่อสารเสียงและวีดิทัศน์ที่ได้กล่าวถึงในบทที่ 3 และบทที่ 4 ตามลำดับ เพื่อให้วิธีการดังกล่าวเป็นประโยชน์ต่อการพัฒนาโปรแกรมทางด้านการสื่อสารแบบพหุสื่อ ในวิทยานิพนธ์นี้จึงได้มีการพัฒนา API สำหรับการสื่อสารแบบพหุสื่อที่สามารถปรับตัวได้ตามสภาพของเครือข่าย โดยมีการควบคุมคุณภาพเสียงและวีดิทัศน์ตามวิธีการที่ได้เสนอขึ้นในบทที่ 3 และบทที่ 4 ซึ่งในบทนี้จะกล่าวถึงสถาปัตยกรรมของ API สำหรับการสื่อสารพหุสื่อที่ได้พัฒนาขึ้น รวมถึงเครื่องมือที่นำมาใช้ร่วมกัน และเพื่อที่จะแสดงให้เห็นว่า API ดังกล่าวสามารถทำงานได้ถูกต้อง จึงมีการพัฒนาโปรแกรมตัวอย่างขึ้นมาและทดสอบบนเครือข่ายในห้องทดลอง

5.1 การออกแบบ API สำหรับการสื่อสารแบบพหุสื่อ

API สำหรับการสื่อสารแบบพหุสื่อของวิทยานิพนธ์นี้ได้รับการออกแบบมาเพื่อใช้ในการพัฒนาโปรแกรมประยุกต์ที่มีความสามารถในการสื่อสารเสียงและวีดิทัศน์แบบจุดต่อจุด โดยมีสถาปัตยกรรมดังรูปที่ 5.1 ซึ่งประกอบด้วยการทำงานในฝั่งของผู้ส่ง 3 โมดูล (Module) คือ โมดูล Video Transmit, โมดูล Audio Transmit และโมดูล Quality Control ส่วนในฝั่งของผู้รับมี 2 โมดูลคือ โมดูล Video Receiver และโมดูล Audio Receive ซึ่งรายละเอียดของแต่ละโมดูลรวมไปถึงความสัมพันธ์ของแต่ละโมดูลมีดังนี้



รูปที่ 5.1 สถาปัตยกรรมของการออกแบบ API สำหรับการสื่อสารแบบพหุสื่อ

5.1.1 โมดูล Video Transmit

โมดูลนี้ทำหน้าที่ในการส่งวีดิทัศน์ไปให้ผู้รับโดยใช้โปรโตคอล RTP การทำงานในโมดูลนี้ประกอบด้วยการจับวีดิทัศน์ การบีบอัดวีดิทัศน์ด้วย MPEG-4 และบรรจุข้อมูลของเฟรมวีดิทัศน์หลังการบีบอัดลงแพ็กเก็ต RTP เพื่อส่งไปให้ผู้รับ ในโมดูลนี้สามารถปรับพารามิเตอร์ในการส่งวีดิทัศน์ได้หลายค่า เช่น อัตราเฟรม อัตราบิตเป้าหมาย ค่าระยะห่างระหว่างเฟรมหลัก ค่าความยาวสูงสุดของเพย์โหลดในแพ็กเก็ต RTP เป็นต้น ดังนั้นในการพัฒนาโปรแกรมประยุกต์สามารถให้ผู้ใช้กำหนดค่าเหล่านี้ได้เอง หรืออาจใช้วิธีการกำหนดโดยใช้อัลกอริทึมในการควบคุมคุณภาพวีดิทัศน์ก็ได้ ซึ่งหน้าที่นี้เป็นของโมดูล Quality Control

5.1.2 โมดูล Audio Transmit

โมดูลนี้ทำหน้าที่ในการส่งเสียงไปให้ผู้รับโดยใช้โปรโตคอล RTP โดยเริ่มจากการจับเสียงและบีบอัดด้วย G.723.1 จากนั้นนำข้อมูลเสียงที่บีบอัดแล้วบรรจุลงในแพ็กเก็ต RTP เพื่อส่งไปให้ผู้รับ โดยสามารถที่จะกำหนด Combination ของข้อมูลซ้ำได้ ซึ่ง Combination ของข้อมูลซ้ำในการสื่อสารเสียงในโมดูลนี้มีอยู่ 6 Combination สำหรับจำนวนข้อมูลซ้ำและรูปแบบการจัดวางข้อมูลซ้ำของแต่ละ Combination มีรายละเอียดอยู่ในตารางที่ 3.5 ซึ่งการบรรจุข้อมูลซ้ำลงในแพ็กเก็ต RTP ของโมดูลนี้ ใช้รูปแบบแพ็กเก็ตตามที่กำหนดใน RFC 2198

5.1.3 โมดูล Quality Control

โมดูลนี้มีหน้าที่ในการควบคุมคุณภาพของการสื่อสารเสียงและวีดิทัศน์ โดยการปรับพารามิเตอร์ในการส่งเสียงและวีดิทัศน์ตามค่าป้อนกลับจากผู้รับ โดยในการควบคุมคุณภาพเสียงใช้อัลกอริทึม CNR ตามที่ได้อธิบายไว้ในบทที่ 3 ส่วนการควบคุมคุณภาพวีดิทัศน์ใช้อัลกอริทึมในการปรับระยะห่างระหว่างเฟรมหลักตามที่ได้อธิบายไว้ในบทที่ 4 โดยสามารถกำหนดค่าพารามิเตอร์ต่าง ๆ ของอัลกอริทึมนี้ได้ทั้งหมด สำหรับสาเหตุที่ใช้โมดูลนี้ควบคุมคุณภาพของทั้งการสื่อสารเสียงและวีดิทัศน์ก็เพื่อให้สามารถปรับค่าอัตราบิตเป้าหมายของการส่งวีดิทัศน์ให้สัมพันธ์กับปริมาณข้อมูลซ้ำของเสียง เช่น เมื่อมีการเพิ่มจำนวนข้อมูลซ้ำของเสียง โมดูลนี้จะลดอัตราบิตเป้าหมายของวีดิทัศน์ลงเพื่อเป็นการชดเชย

5.1.4 โมดูล Video Receive

โมดูลนี้ทำหน้าที่รับแพ็กเก็ตวีดิทัศน์มาถอดรหัสเพื่อให้ได้เฟรมวีดิทัศน์ที่พร้อมจะนำไปแสดงผล และต้องมีการคำนวณค่าอัตราเฟรมฝั่งรับโดยนับเฉพาะเฟรมที่ถอดรหัสถูกต้องเท่านั้นเพื่อใช้ในการป้อนกลับไปให้ผู้ส่งโดยใช้แพ็กเก็ต RTCP Receiver Report โดยบรรจุลงในส่วนขยายเฮดเดอร์ดังที่ได้อธิบายไว้ในหัวข้อ 4.6.2

5.1.5 โมดูล Audio Receive

โมดูลทำหน้าที่ในการรับแพ็กเก็ตเสียงที่ถูกส่งมาจากผู้ส่ง เพื่อนำมาถอดรหัสและเล่นเสียงออกทางลำโพง นอกจากนี้ยังต้องมีการคำนวณค่าอัตราการสูญหายของแพ็กเก็ตก่อนและหลังการจัดเรียงข้อมูลซ้ำ เพื่อบรรจุลงในแพ็กเก็ต RTCP Receiver Report และป้อนกลับไปให้ผู้ส่ง

5.2 เครื่องมือที่ใช้ประกอบการพัฒนา API

ในการพัฒนา API สำหรับการสื่อสารแบบพหุสื่อของวิทยานิพนธ์นี้ ในการเขียนโปรแกรมบางส่วนได้มีการใช้ API อื่นเข้าร่วมด้วยเพื่อช่วยเพิ่มความรวดเร็วในการพัฒนา สำหรับเครื่องมือที่ใช้ประกอบในการพัฒนา API สำหรับการสื่อสารแบบพหุสื่อมีดังนี้

5.2.1 Java 2 Software Development Kit (J2SDK)

ในวิทยานิพนธ์นี้ได้ใช้ภาษาจาวา (Java) ในการพัฒนา API สำหรับการสื่อสารแบบพหุสื่อ และใช้ J2SDK เวอร์ชัน 1.4.2 ในการคอมไพล์และรันโปรแกรม

5.2.2 Java Media Framework (JMF)

JMF เป็น API สำหรับการเขียนโปรแกรมทั้งด้านพหุสื่อ ไม่ว่าจะเป็นการจับเสียงและวีดิทัศน์ การเล่นเสียง การแสดงผลวีดิทัศน์ การบีบอัดเสียงและวีดิทัศน์ นอกจากนี้ JMF ยังรองรับการสื่อสารโดยใช้โปรโตคอล RTP อีกด้วย แต่ในวิทยานิพนธ์นี้ได้ใช้ API ของ JMF เพียงบางส่วนเท่านั้น ซึ่งได้แก่ การจับวีดิทัศน์ การแสดงผลวีดิทัศน์ และการบีบอัดเสียงด้วย G.723.1 แต่ในส่วนของโปรโตคอล RTP นั้น ในวิทยานิพนธ์นี้ได้เขียนโปรแกรมในส่วนนี้ขึ้นมาใหม่ เนื่องจากใน API ของ JMF ไม่ได้รองรับการส่งข้อมูลซ้ำของเสียงตาม RFC 2198 และไม่สามารถเพิ่มส่วนขยายเฮดเดอร์ได้ สำหรับเวอร์ชันของ JMF ที่ใช้ในวิทยานิพนธ์นี้คือเวอร์ชัน 2.1.1e

5.2.3 XviD MPEG-4 Codec

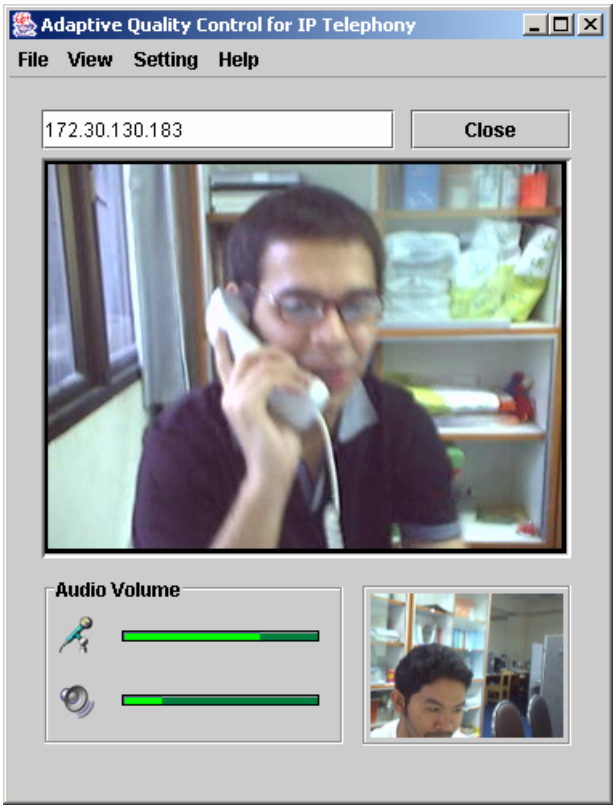
XviD[43] เป็น API สำหรับการบีบอัดวีดิทัศน์ด้วย MPEG-4 ซึ่งมีการเผยแพร่ Source Code ให้ใช้งานได้ฟรี แต่เนื่องจาก XviD ได้รับการพัฒนาโดยใช้ภาษาซี ดังนั้นจึงจำเป็นต้องใช้เทคโนโลยี Java Native Interface (JNI) เพื่อให้โปรแกรมภาษาจาวาสามารถเรียกใช้ API ที่เขียนด้วยภาษาซีได้ สำหรับเวอร์ชันของ XviD ที่ใช้ในวิทยานิพนธ์นี้คือเวอร์ชัน 1.0.3 และได้ใช้โปรแกรม MicroSoft Visaul C++ เวอร์ชัน 6 ในการคอมไพล์ Source Code ของ XviD

5.3 การพัฒนาโปรแกรมตัวอย่างและการทดสอบ

เพื่อเป็นการทดสอบว่า API สำหรับการสื่อสารแบบพหุสื่อที่ได้พัฒนาขึ้นสามารถทำงานได้ถูกต้อง และเพื่อเป็นการพิสูจน์ว่าวิธีการควบคุมคุณภาพเสียงและวีดิทัศน์ที่ได้เสนอขึ้นในวิทยานิพนธ์นี้สามารถใช้งานได้จริง จึงได้มีการพัฒนาโปรแกรมตัวอย่างขึ้น เพื่อใช้โปรแกรมนี้ในการทดสอบบนเครือข่ายในห้องทดลอง

5.3.1 โปรแกรมตัวอย่าง

รูปที่ 5.2 เป็นภาพของโปรแกรมตัวอย่างที่จะใช้ในการทดสอบ โดยโปรแกรมนี้สามารถกำหนดแอดเดรสของเครื่องผู้รับได้ รวมทั้งยังสามารถกำหนดค่าพารามิเตอร์ในการส่งเสียงและวีดิทัศน์ได้ทั้งในแบบที่กำหนดโดยผู้ใช้ หรือควบคุมแบบอัตโนมัติโดยใช้วิธีการควบคุมคุณภาพแบบปรับตัว นั่นคือใช้อัลกอริทึม CNR สำหรับการควบคุมคุณภาพเสียง และใช้อัลกอริทึมการปรับค่าระยะห่างระหว่างเฟรมหลักสำหรับการควบคุมคุณภาพวีดิทัศน์ นอกจากนี้โปรแกรมนี้ยังมีการบันทึกค่าสถิติต่าง ๆ ลงในไฟล์ซึ่งจะช่วยอำนวยความสะดวกในการบันทึกผลการทดลอง



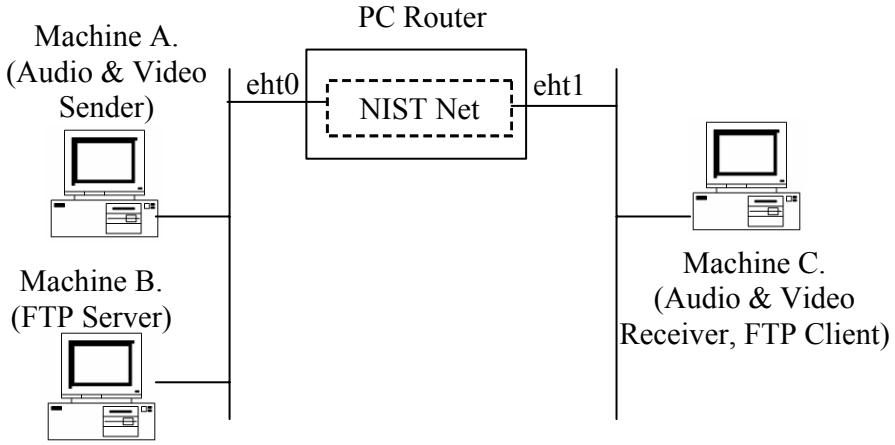
รูปที่ 5.2 โปรแกรมตัวอย่างที่ใช้ในการทดสอบ API

ส่วนในการทดสอบว่าอัลกอริทึมในการควบคุมคุณภาพของสื่อทั้งสองสามารถใช้งานได้จริงหรือไม่นั้น ได้มีการจัดเครือข่ายสำหรับการทดสอบขึ้นภายในห้องทดลอง แต่เนื่องจากว่าเครือข่ายในห้องทดลองนั้นเป็นระบบเครือข่ายท้องถิ่น (Local Area Network) และมีความจุของเครือข่ายถึง 100 เมกะบิตต่อวินาที ซึ่งเป็นการยากที่จะทำให้มีแพ็กเก็ตสูญหาย ดังนั้นในการทดลองนี้จึงได้มีการใช้โปรแกรมประเภทตัวจำลองเครือข่าย (Network Emulator) เพื่อช่วยในกำหนดความจุและเวลาหน่วงของลิงค์ภายในเครือข่าย โดยในที่นี้ได้เลือกใช้โปรแกรม NIST Net[44] เวอร์ชัน 2.0.11

5.3.2 การทดลอง

ในการทดลองได้จัดการเชื่อมต่อของเครื่องคอมพิวเตอร์ภายในเครือข่ายรูปที่ 5.3 โดยมีการรันโปรแกรม NIST Net ที่เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเราเตอร์ และกำหนดแบนด์วิดท์ระหว่างอินเตอร์เฟส (Interface) eth0 และ eth1 เท่ากับ 2 เมกะบิตต่อวินาที ค่าเวลาหน่วงเท่ากับ 10 มิลลิวินาที และคิวที่ใช้ในการพักแพ็กเก็ตมีขนาดเท่ากับ 75 แพ็กเก็ต เครื่อง A และเครื่อง C จะรันโปรแกรมตัวอย่าง โดยที่เครื่อง A จะส่งแพ็กเก็ตของเสียงและวีดิทัศน์ไปยังเครื่อง C นอกจากนี้เครื่อง C ยังมีการดาวน์โหลดไฟล์จากเครื่อง B โดยใช้โปรโตคอล FTP ซึ่งในที่นี้ใช้โปรแกรม CuteFTP เวอร์ชัน 4.0.19 ในการดาวน์โหลดไฟล์ โดยการดาวน์โหลดไฟล์นี้เปรียบเสมือนกับการใช้งานของผู้ใช้คนอื่น ๆ ที่ใช้ลิงค์ร่วมกันกัน และเนื่องจากว่าโปรโตคอล FTP นั้นทำงานอยู่บนโปรโตคอล TCP ซึ่งจะพยายามใช้แบนด์วิดท์ที่มีอยู่ให้มากที่สุดเท่าที่จะเป็นไปได้ ซึ่งจะทำให้แพ็กเก็ตของเสียงและวีดิทัศน์ที่ส่งจากเครื่อง A ไปยังเครื่อง B อาจมีการสูญหายได้ สำหรับขั้นตอนของการส่งข้อมูลของแต่ละเครื่องในการทดลองนี้เป็นดังนี้

1) ที่วินาทีที่ 0 เครื่อง A เริ่มส่งแพ็กเก็ตเสียงและวีดิทัศน์ โดยในการส่งเสียงนั้นเริ่มต้นด้วยใช้ Combination หมายเลข 0 ซึ่งไม่มีการส่งข้อมูลซ้ำ ส่วนในการส่งวีดิทัศน์นั้นซึ่งเป็นวีดิทัศน์ที่ถ่ายด้วยกล้องวีดิทัศน์รุ่น Logitech QuickCam Pro 4000 และกำหนดขนาดภาพของวีดิทัศน์เป็น 320x240 จุดภาพ อัตราเฟรม 24 เฟรมต่อวินาที ค่าเริ่มต้นของระยะห่างระหว่างเฟรมหลักเท่ากับ 24 เฟรม
2) ที่วินาทีที่ 100 เครื่อง C เริ่มดาวน์โหลดไฟล์ โดยมีการดาวน์โหลดทั้งหมด 5 การติดต่อ แต่ละการติดต่อดาวน์โหลดไฟล์เดียวกันและเริ่มดาวน์โหลดห่างกันประมาณ 2–3 วินาที
3) ที่วินาทีที่ 300 เครื่อง C หยุดดาวน์โหลดไฟล์ทุกการติดต่อ แต่เนื่องจากว่าการสั่งหยุดดาวน์โหลดไฟล์เป็นการสั่งโดยผู้ใช้ ดังนั้นการติดต่อแรกเท่านั้นที่หยุดดาวน์โหลดไฟล์ที่วินาทีที่ 300 ส่วนการติดต่ออื่นนั้นใช้เวลาห่างกันประมาณ 2 – 3 วินาที
4) ที่วินาทีที่ 500 เครื่อง A หยุดส่งแพ็กเก็ตเสียงและวีดิทัศน์



รูปที่ 5.3 เครือข่ายที่ใช้ในการทดลอง

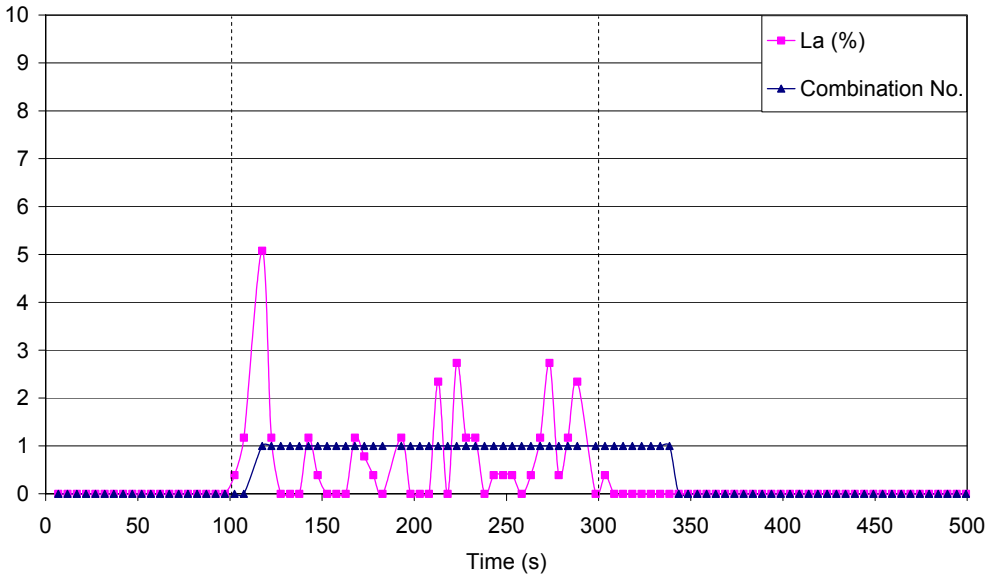
สำหรับการควบคุมคุณภาพเสียงด้วยอัลกอริทึม CNR นั้น ในการทดลองนี้ได้มีการกำหนดค่าเทรสโฮลด์ HIGH เท่ากับ 5% และกำหนดค่าเทรสโฮลด์ LOW เท่ากับ 1% ส่วนการควบคุมคุณภาพวีดิทัศน์ด้วยอัลกอริทึมในการปรับค่าระยะห่างระหว่างเฟรมหลักนั้น ในการทดลองนี้กำหนดให้ใช้ค่าพารามิเตอร์ต่าง ๆ ตามค่าแนะนำที่ได้อธิบายไว้ในหัวข้อ 4.6 โดยกำหนดค่าของ I_c เท่ากับ 5 เฟรม, F_{low} เท่ากับ 15 เฟรมต่อวินาที, F_{high} เท่ากับ 20 เฟรมต่อวินาที และ MIN_DURATION เท่ากับ 4 รอบ และได้กำหนดค่าเป้าหมายของอัตราบิตรวมในการส่งเสียงและวีดิทัศน์ไว้เท่ากับ 900 กิโลบิตต่อวินาที

5.3.3 ผลการทดลอง

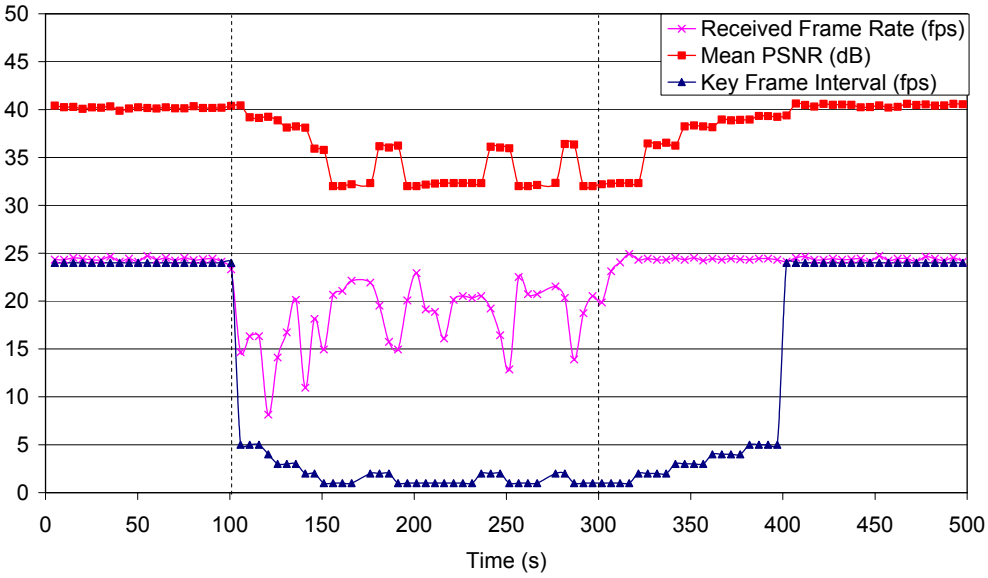
รูปที่ 5.4 เป็นกราฟแสดงค่าอัตราการสูญหายของแพ็กเก็ตเสียงหลังการจัดเรียง (L_a) และการเปลี่ยนแปลง Combination ของข้อมูลซ้ำในการสื่อสารเสียง จะเห็นว่าช่วงเวลาในวินาทีที่ 0 – 100 และช่วงวินาทีที่ 300 – 500 ซึ่งเป็นช่วงเวลาที่เครือข่ายไม่คับคั่ง อัลกอริทึม CNR จะกำหนดให้ใช้ Combination หมายเลข 0 ซึ่งไม่มีการส่งข้อมูลซ้ำ แต่ในช่วงวินาทีที่ 100 – 300 ซึ่งเป็นช่วงเวลาที่เครือข่ายมีความคับคั่งสูง อัลกอริทึม CNR จะมีการเปลี่ยนแปลง Combination เพื่อควบคุมไม่ให้มีข้อมูลเสียงสูญหายมากเกินไป ซึ่งจากกราฟในรูปที่ 5.4 จะเห็นว่าในช่วงเวลาดังกล่าวอัลกอริทึม CNR จะใช้ Combination หมายเลข 1 เพื่อรักษาระดับของ L_a ไม่ให้เกิน 5%

รูปที่ 5.5 เป็นกราฟแสดงการเปลี่ยนแปลงของค่าอัตราเฟรมฝั่งรับและค่าระยะห่างระหว่างเฟรมหลักของการส่งวีดิทัศน์ของโปรแกรมตัวอย่างในการทดลอง จะเห็นว่าเมื่อเครือข่ายเริ่มคับคั่งขึ้นในวินาทีที่ 100 ค่าระยะห่างระหว่างเฟรมหลักจะถูกลดลงเพื่อควบคุมค่าอัตราเฟรม

ฝั่งรับไม่ให้ต่ำกว่าค่า F_{low} (15 เฟรมต่อวินาที) แต่เมื่อถึงวินาทีที่ 300 เครือข่ายมีความคับคั่งลดลง ค่าระยะห่างระหว่างเฟรมหลักก็จะถูกปรับเพิ่มขึ้นเพื่อเพิ่มความชัดเจนของภาพในแต่ละเฟรมวีดิทัศน์ ซึ่งสามารถสังเกตได้จากค่า PSNR เมื่อเพิ่มค่าระยะห่างระหว่างเฟรมหลักค่า PSNR ก็จะเพิ่มขึ้นตามไปด้วย



รูปที่ 5.4 อัตราการสูญหายของแพ็กเก็ตเสียงและการเปลี่ยนแปลง Combination



รูปที่ 5.5 การเปลี่ยนแปลงของค่าอัตราเฟรมฝั่งรับและค่าระยะห่างระหว่างเฟรมหลัก

จากผลการทดลองสามารถสรุปได้ว่าทั้งอัลกอริทึม CNR และอัลกอริทึมในการปรับระยะห่างระหว่างเฟรมหลัก สามารถนำมาใช้ในการควบคุมคุณภาพเสียงและวีดิทัศน์ได้ในเครือข่ายจริง มิใช่จะทำงานได้ดีเฉพาะในแบบจำลองเท่านั้น

5.4 สรุป

ในบทนี้ได้นำเสนอการพัฒนา API สำหรับการสื่อสารแบบพหุสื่อที่สามารถปรับตัวได้ตามสภาพเครือข่าย โดยได้นำอัลกอริทึม CNR มาใช้ในการควบคุมคุณภาพเสียง และใช้อัลกอริทึมในการปรับระยะห่างระหว่างเฟรมหลักในการควบคุมคุณภาพวีดิทัศน์ และเพื่อเป็นการยืนยันว่าอัลกอริทึมทั้งสองสามารถนำมาใช้งานได้จริง มิใช่จะทำงานได้เฉพาะในแบบจำลองเท่านั้น จึงได้มีการนำ API ดังกล่าวมาพัฒนาโปรแกรมตัวอย่างและทำการทดสอบบนเครือข่ายในห้องทดลอง จากผลการทดลองได้แสดงให้เห็นแล้วว่าอัลกอริทึมทั้งสองสามารถใช้งานได้จริง โดยสามารถปรับพารามิเตอร์ของการส่งเสียงและวีดิทัศน์ให้เหมาะสมกับสภาพเครือข่ายได้ ซึ่ง API ที่พัฒนาขึ้นนี้ถือได้ว่าเป็นประโยชน์ต่อการพัฒนาโปรแกรมทางด้านการสื่อสารแบบพหุสื่อ เนื่องจากจะช่วยอำนวยความสะดวกในการเขียนโปรแกรมและทำให้สามารถพัฒนาโปรแกรมได้รวดเร็วขึ้น