การประเมินประสิทธิภาพโพรโทคอลควบคุมความคับคั่งแบบหลายอัตรา
ของมัลติคาสท์ในการให้บริการมัลติมีเดียทราฟฟิก
ภายใต้กฎความเป็นมิตรต่อทีซีพี

วิทยานิพนธ์

ของ

กฤติน พันธุ์เสนา

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

พฤษภาคม 2555

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

Mahasarakham University

การประเมินประสิทธิภาพโพรโทคอลควบคุมความคับคั่งแบบหลายอัตรา
ของมัลติคาสท์ในการให้บริการมัลติมีเดียทราฟฟิก
ภายใต้กฎความเป็นมิตรต่อทีซีพี

วิทยานิพนธ์

ของ

กฤติน พันธุ์เสนา

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

พฤษภาคม 2555

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

Mahasarakham University

คณะกรรมการสอบวิทยานิพนธ์ ได้พิจารณาวิทยานิพนธ์ของนายกฤติน พันธุ์เสนา แล้วเห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ
(อาจารย์ ดร.ฉัตรเกล้า เจริญผล) (อาจารย์บัณฑิตศึกษาประจำคณะ)

.... กรรมการ
(อาจารย์ ดร.สมนึก พ่วงพรพิทักษ์) (ประธานกรรมการควบคุมวิทยานิพนธ์)

.... กรรมการ
(ผศ.ดร.จิรัฏฐา ภูบุญอบ) (กรรมการควบคุมวิทยานิพนธ์)

.... กรรมการ
(อาจารย์ ดร.คำรณ สุนัติ) (ผู้ทรงคุณวุฒิ)

มหาวิทยาลัยอนุมัติให้รับวิทยานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

....
(รศ.วิรัตน์ พงษ์ศิริ)
คณบดีคณะวิทยาการสารสนเทศ

....
(ศ.ดร.ประดิษฐ์ เทอดทูล)
ผู้รักษาการคณบดีบัณฑิตวิทยาลัย

วันที่ 15 เดือน พ.ค. พ.ศ. 2555

Mahasarakham University

กิตติกรรมประกาศ

งานวิจัยครั้งนี้สำเร็จลุล่วงไปด้วยดีเพราะได้รับการดูแลด้วยเมตตาความกรุณาเป็นอย่างยิ่ง จาก อาจารย์ ดร.สมนึก พ่วงพรพิทักษ์ และผู้ช่วยศาสตราจารย์ ดร.จิรัฏฐา ภูบุญอบ ที่ได้กรุณาให้คำปรึกษาแนะนำ ชี้แนะเกี่ยวกับการศึกษาวิจัย การทำงาน การนำเสนอผลงานวิจัย และคอยดูแลเอาใจใส่ในการทำงานตลอดมา

ขอกราบขอบพระคุณคณาจารย์ของภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคามทุกท่านที่ได้อบรม สั่งสอน และให้ความรู้ทั้งวิชาการทางคอมพิวเตอร์และการทำงานวิจัยให้สามารถใช้ในงานวิจัยครั้งนี้ให้ประสบผลสำเร็จออกมา

ขอกราบขอบพระคุณคุณพ่อและคุณแม่ที่อบรมสั่งสอน ให้เป็นคนดี และตั้งใจทำงานโดยไม่หวังสิ่งใดตอบแทน และเป็นผู้สนับสนุนค่าใช้จ่ายในการเล่าเรียน การทำวิทยานิพนธ์ และอื่นๆ ตลอดมา ทำให้การทำงานเป็นไปด้วยความราบรื่น

ขอขอบคุณ นางสารดา พันธุ์เสนา ที่คอยช่วยเหลือในบางเวลาที่ยากลำบากและเป็นกำลังใจให้ตั้งใจทำงานจนประสบผลสำเร็จ

กฤติน พันธุ์เสนา

Mahasarakham University

ชื่อเรื่อง	การประเมินประสิทธิภาพโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราของมัลติคาสท์ในการให้บริการมัลติมีเดียทราฟฟิกภายใต้กฎความเป็นมิตรต่อทีซีพี
ผู้วิจัย	นายกฤติน พันธุ์เสนา
กรรมการควบคุม	ดร.สมนึก พ่วงพรพิทักษ์ และ ผู้ช่วยศาสตราจารย์ ดร.จิรัฏฐา ภูบุญอบ
ปริญญา	วท.ม. **สาขาวิชา** เทคโนโลยีสารสนเทศ
มหาวิทยาลัย	มหาวิทยาลัยมหาสารคาม **ปีที่พิมพ์** 2555

บทคัดย่อ

โพรโทคอลควบคุมความคับคั่งแบบหลายอัตรา (MR-MCC) ได้ถูกเสนอให้เป็นเครื่องมือสำหรับควบคุมความคับคั่งของมัลติคาสท์บนอินเทอร์เน็ต เนื่องจากทีซีพีเป็นทราฟฟิกหลักบนอินเทอร์เน็ตจึงทำให้โพรโทคอล MR-MCC ที่ดีควรใช้แบนด์วิดท์น้อยกว่าหรือเท่ากับทีซีพีในสถานการณ์เดียวกัน ซึ่งกฎนี้ถูกเรียกว่ากฎความเป็นมิตรต่อทีซีพี (TCP-friendliness) โพรโทคอล MR-MCCจำนวนมากที่ถูกเสนอขึ้นมาต่างอ้างว่าเป็นโพรโทคอลที่เป็นมิตรต่อทีซีพี อย่างไรก็ตามยังไม่มีการประเมินประสิทธิภาพของโพรโทคอล MR-MCC เหล่านั้นในการให้บริการมัลติมีเดียทราฟฟิก เนื่องจากมัลติมีเดียทราฟฟิกมีคุณสมบัติที่แตกต่างจากทีซีพีทราฟฟิกที่อาจส่งผลให้ประสิทธิภาพของโพรโทคอล MR-MCC ลดลงเมื่อต้องปฏิบัติตามกฎความเป็นมิตรต่อทีซีพี ดังนั้นงานวิจัยนี้จึงนำเสนอการประเมินประสิทธิภาพของโพรโทคอล MR-MCC ที่ให้บริการมัลติมีเดียทราฟฟิกภายใต้กฎความเป็นมิตรต่อทีซีพีโดยการจำลองระบบเครือข่ายด้วย ns-2 ซึ่งผลการทดลองได้เผยให้เห็นหลายแง่มุมและกฎความเป็นมิตรต่อทีซีพีสำหรับมัลติมีเดียทราฟฟิกอาจจะต้องมีการทบทวนใหม่

คำสำคัญ : มัลติคาสท์; ความคับคั่ง; ความเป็นมิตรต่อทีซีพี

TITLE An Evaluation of TCP-friendliness of Multi-rate Multicast Congestion Control in serving Multimedia Traffic

AUTHOR Mr. Kittin Phansena

ADVISORS Somnuk Puangpronpitag, Ph.D.

Asst.Prof. Jiratta Phuboon-ob, Ph.D.

DEGREE M.Sc. **MAJOR** Information Technology

UNIVERSITY Mahasarakham University **DATE** 2012

ABSTRACT

Several Multi-rate Multicast Congestion Control (MR-MCC) protocols have been proposed as engines for Internet multicast transport congest control mechanism. A good MR-MCC protocol (according to IETF) should use bandwidth less than or equal to Transmission Control Protocol (TCP) in the same situation since TCP traffic is a major traffic on the Internet. This rule is called TCP-friendliness. A lot of proposed MR-MCC protocols have claimed themselves as the TCP-friendly protocols. However, there has no performance evaluation of those TCP-friendly MR-MCC protocols in serving multimedia traffic so far. Since multimedia traffic has several different features from TCP-based traffic, there may be some performance penalty when being TCP-friendly. So, this thesis has proposed to evaluate the TCP-friendliness of MR-MCC in serving multimedia traffic using network simulation via ns-2. The experimental results have revealed several aspects, and the TCP-friendliness definition for multimedia traffic may have to be rethought.

Key Words : Multicast; Congestion; TCP-Friendliness

สารบัญ

หน้า

กิตตกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญภาพประกอบ	ซ
บทที่ 1 บทนำ	1
หลักการและเหตุผล	1
วัตถุประสงค์ของการวิจัย	2
ความสำคัญของการวิจัย	2
ขอบเขตของการวิจัย	3
นิยามศัพท์เฉพาะ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
รูปแบบการสื่อสารข้อมูล	4
รูปแบบการแพร่สัญญาณเฉพาะกลุ่มบนอินเทอร์เน็ต	7
Internet Application	8
การส่งผ่านข้อมูลแบบมัลติมีเดีย	10
Multicast Multimedia Streaming	14
การควบคุมการส่งผ่านข้อมูลในระบบเครือข่าย	15
ความคับคั่งของระบบเครือข่าย	17
การควบคุมความคับคั่งของข้อมูลในมัลติคาสท์	18
เกณฑ์การประเมินผลโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราส่ง	23
เทคนิคที่ใช้ในการทดสอบประสิทธิภาพของโพรโทคอล	25
Network Simulation	25

Mahasarakham University

หน้า

งานวิจัยที่เกี่ยวข้อง 25
ประเด็นปัญหาการวิจัย 33

บทที่ 3 วิธีดำเนินการวิจัย 36
เครื่องมือการจำลองระบบเครือข่าย 36
ขั้นตอนการจำลองระบบเครือข่าย 37
Network Simulation 2 (ns-2) 39
การเตรียมการทดลอง 40
การดำเนินการทดลอง 45
การบันทึกผลการทดลอง 47

บทที่ 4 ผลการดำเนินงาน 50
การทดลองที่ 1 การทดสอบเพื่อยืนยันความสามารถของโพรโทคอล MeERA ในการตอบสนองต่อสภาวะเครือข่าย 50
การทดลองที่ 2 การทดสอบเพื่อยืนยันความสามารถของโพรโทคอล MeERA ในการเป็นมิตรต่อทีซีพี 52
การทดลองที่ 3 การทดสอบส่งผ่านข้อมูลเสียง 53
การทดลองที่ 4 การทดสอบการส่งผ่านข้อมูลมัลติมีเดียชนิด Video (MPEG2) 54
การทดลองที่ 5 การทดสอบเพื่อวิเคราะห์ประเด็น Responsiveness vs. Smoothness 55
การทดลองที่ 6 การทดสอบเพื่อวิเคราะห์ประเด็น Bandwidth Consumption 57
การทดลองที่ 7 การทดสอบเพื่อวิเคราะห์ประเด็น Layer Granularity 58
การทดลองที่ 8 การทดสอบเพื่อวิเคราะห์ประเด็น Layer Granularity ที่มีการเข้ารหัสไฟล์ที่แตกต่างกัน 60

บทที่ 5 สรุปผล และข้อเสนอแนะ 62
ข้อสรุปของงานวิจัย 62
ข้อเสนอแนะ 63

Mahasarakham University

หน้า

เอกสารอ้างอิง 64

ภาคผนวก 69
ภาคผนวก ก 70

ประวัติย่อผู้วิจัย 90

Mahasarakham University

สารบัญตาราง

ตาราง **หน้า**

1 ขอบเขตของความต้องการของซอฟต์แวร์ต่างๆ 9
2 แสดงความต้องการแบนด์วิดท์ของข้อมูลมัลติมีเดียและคุณภาพที่ได้รับ 11
3 แสดงการเปรียบเทียบข้อมูลบนอินเทอร์เน็ตที่มีความไวต่อพารามิเตอร์ต่าง ๆ 14
4 การเปรียบเทียบเทคนิคการทดสอบประสิทธิภาพ 25
5 ERA Packet Headers 27
6 แสดงสัดส่วนการใช้แบนด์วิดท์ ระหว่าง MeERA กับ TCP 30
7 ตารางรูปแบบของ Trace ไฟล์ 47
8 ตารางความหมายของ Trace ไฟล์ แต่ละคอลัมน์ 48
9 เปรียบเทียบอัตราการใช้แบนด์วิดท์เฉลี่ยระหว่าง TCP และ MeERA 53
10 เปรียบเทียบอัตราการใช้แบนด์วิดท์เฉลี่ยระหว่าง TCP และ MeERA 54
11 แสดงการใช้แบนด์วิดท์ของ MeERA ในช่วงเวลาต่างๆ 56
12 แสดงการใช้แบนด์วิดท์ของ MeERA ทีมีผลต่อคุณภาพของมัลติมีเดีย 56
13 แสดงการใช้แบนด์วิดท์ของ MeERA ในช่วงเวลาต่างๆ 58
14 แสดงขนาดชั้นข้อมูลสะสมที่เพิ่มขึ้นตามระดับคุณภาพของสัญญาณ 59
15 แสดงค่า Throughput ของโพรโทคอล ERA และ TCP 60
16 เปรียบเทียบคุณภาพมัลติมีเดียที่มีการเข้ารหัสไฟล์ต่างกันที่อัตราบิตเท่ากัน 61

Mahasarakham University

สารบัญภาพประกอบ

ภาพประกอบ หน้า

1	การส่งข้อมูลแบบยูนิคาสท์	4
2	Multiple Unicast	5
3	รูปแบบการส่งข้อมูลของมัลติคาสท์	6
4	แสดงการเปรียบเทียบ Traffic ระหว่าง Unicast กับ Multicast	6
5	Sample of the IP Multicasting	7
6	TCP/IP Model vs OSI Model	15
7	เปรียบเทียบโพรโทคอลมาตรฐานในยูนิคาสท์และมัลติคาสท์	16
8	Network Congestion	17
9	แสดง SR-MCC	21
10	Multi-rate multicast congestion controls	20
11	ขั้นตอนการจำลอง Network ด้วย Network Simulation	38
12	แสดงผลการรันโปรแกรม ns-2	39
13	ขั้นตอนการทำงานของ Network Simulator	40
14	โครงสร้างเครือข่ายกรณีที่ 1	42
15	โครงสร้างเครือข่ายกรณีที่ 2	43
16	โครงสร้างเครือข่ายในการทดลอง 2	46
17	เมื่อ Network Animator Tool เริ่มมีการทำงาน	49
18	กราฟผลการทดลอง	49
19	โทพอโลยีที่ใช้ในการทดลองที่ 1	50
20	กราฟค่า Throughput ของโพโทคอล MeERA ที่ตอบสนองต่อการเปลี่ยนแปลงของสภาวะในระบบเครือข่าย	51
21	โทพอโลยีที่ใช้ในการทดลองที่ 2	51
22	กราฟแสดงการแบ่งปันแบนด์วิดท์ระหว่าง MeERA และ TCP ขณะเกิดสถานการณ์ความคับคั่งในระบบเครือข่าย	52
23	กราฟ Throughput ของโพรโทคอล MeERA และ TCP	53
24	กราฟ Throughput ของโพรโทคอล MeERA และ TCP	54

ภาพประกอบ **หน้า**

25 กราฟ Throughput ของโพรโทคอล MeERA และ TCP 55

26 กราฟ Throughput ของโพรโทคอล MeERA และ TCP 56

27 โทพอโลยีที่ใช้ในการทดลองที่ 8 59

28 แสดงค่า Throughput ของ MeERA และ TCP ระหว่างทำการส่งข้อมูล 59

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

เนื่องจากอินเทอร์เน็ตในปัจจุบันได้มีการเติบโตขึ้นอย่างรวดเร็ว ส่งผลให้การสื่อสารแบบกลุ่ม (Multiparty Communication) เช่น Video Streaming, Media-on-demand Broadcast และ Distributed Computing นั้นมีบทบาทมากขึ้น การสื่อสารแบบมัลติคาสท์ (Multicast) [1] (Deering. 1989 : 1-15) จึงถูกเสนอขึ้นมาเพื่อช่วยลดจำนวนข้อมูลจากจุดหนึ่งไปยังปลายทางหลายๆจุด โดยอาศัยเราท์เตอร์ในการทำซ้ำข้อมูลเพื่อส่งไปยังผู้รับปลายทางหลายจุด โดยจะทำซ้ำและส่งต่อเพียงหนึ่งแพ็กเกตต่อหนึ่งเส้นทางเท่านั้น วิธีการนี้ช่วยลดปริมาณการส่งข้อมูลของระบบเครือข่ายลงอย่างมากเมื่อเทียบกับการส่งข้อมูลแบบดั้งเดิม (ยูนิคาสท์) ที่เป็นการส่งแบบจุดต่อจุด

จากการศึกษางานวิจัยพบว่าการที่มัลติคาสท์ไม่ถูกใช้งานในอินเทอร์เน็ตแต่กลับมีการใช้งานในระบบเครือข่ายแบบที่สามารถจัดการได้ (Manage Network) เนื่องมาจากกลไกควบคุมความคับคั่งสำหรับมัลติคาสท์ในประเด็นหลายประเด็นยังไม่มีข้อสรุป โดยเฉพาะในประเด็นของกลไกการควบคุมความคับคั่ง เพราะปัญหาความคับคั่งในมัลติคาสท์นั้นสามารถสร้างความเสียหายต่อระบบเครือข่ายที่รุนแรงและกว้างขวางกว่ายูนิคาสท์

โดยจากการศึกษางานวิจัยค้นพบว่า กลไกการควบคุมความคับคั่งของมัลติคาสท์สำหรับอินเทอร์เน็ตที่ได้รับการยอมรับมากที่สุดได้แก่เทคนิคที่เรียกว่า กลไกการควบคุมความคับคั่งแบบหลายอัตรา (Multi-rate Multicast Congestion Control , MR-MCC) [2] โดย MR-MCC เป็นวิธีที่อนุญาตให้ผู้รับข้อมูลสามารถรับข้อมูลในอัตราที่ต่างกันได้ขึ้นอยู่กับสถานการณ์ความคับคั่งของระบบเครือข่ายที่เชื่อมต่อกับผู้รับ ซึ่งกลไกนี้จะอาศัยความสามารถของผู้ส่งข้อมูลในการเข้ารหัสข้อมูลให้เป็นชั้น (Layer) หากผู้รับบอกรับ (Subscribe) จำนวนชั้นมากขึ้นจะได้รับข้อมูลที่รวดเร็วและมีคุณภาพดีขึ้น แต่ก็จะใช้แบนด์วิดท์ของระบบเครือข่ายมากขึ้นเช่นกัน

แนวทางของ MR-MCC ที่ดูเหมือนจะมีความเป็นไปได้ในการใช้งานจริงบนอินเทอร์เน็ต ดังที่ได้มีการนำเสนอโพรโทคอล MR-MCC ขึ้นมา เช่น PLM, ERA, WEBRC [2] โดยโพรโทคอลเหล่านี้ถูกสร้างตามแนวคิดของ MR-MCC ซึ่งผู้เสนอโพรโทคอลต่างบอกว่าโพรโทคอลของตนนั้นมีความเป็นมิตรต่อทีซีพีซึ่งเป็นหนึ่งในปัจจัยสำคัญในการดำเนินการควบคุมความคับคั่งบนอินเทอร์เน็ต โดยยึดหลักว่ามัลติคาสท์ต้องไม่ใช้แบนด์วิดท์มากกว่าทีซีพี ตามนิยามของ IETF [3]

Mahasarakham University

ต่อมา Puangpronpitag [4] ได้เสนอให้มัลติคาสท์โฟล์ว (Multicast flow) ที่ให้บริการผู้รับที่มากกว่าทีซีพีโฟล์ว (TCP flow) มีโอกาสได้รับแบนด์วิดท์มากกว่าทีซีพี หากมีการให้บริการที่มากกว่าหนึ่งผู้รับ ซึ่งเป็นแนวคิดความเป็นมิตรต่อทีซีพีแบบใหม่ที่ทำให้มีความคุ้มค่าที่จะนำมัลติคาสท์ไปใช้และมีแนวโน้มว่ามัลติคาสท์จะได้รับการยอมรับมากขึ้น

แต่ยังมีข้อสังเกตว่าบรรดาโปรแกรมประยุกต์ (Application) ที่จะนำมาใช้ในมัลติคาสท์นั้นยังแบ่งออกเป็น 2 ชนิด คือ ข้อมูลแบบสื่อผสมมัลติมีเดีย (Multimedia) และข้อมูลเนื้อหา (Content) ทั่วไป เช่น HTTP, FTP, Mail ซึ่งข้อมูลประเภทมัลติมีเดียที่จะถูกมัลติคาสท์นั้นอาจไม่สามารถเป็นไปตามกรอบนิยามความเป็นมิตรต่อทีซีพีของ IETF ได้โดยสมบูรณ์อันเนื่องมาจากลักษณะการสื่อสารของมัลติมีเดียนั้นมีคุณลักษณะที่แตกต่างไปจากทีซีพีโดยสิ้นเชิง โดยทีซีพีต้องการความถูกต้องครบถ้วนของข้อมูลมากกว่าความรวดเร็วในการรับ-ส่ง แต่มัลติมีเดียให้ความสำคัญกับความเร็วของการรับ-ส่ง มากกว่าความถูกต้องของข้อมูล ถ้าหากต้องเป็นไปตามกฎความเป็นมิตรต่อทีซีพีของ IETF แล้วอาจจะทำให้ประสิทธิภาพของการสื่อสารของมัลติมีเดียบนมัลติคาสท์นั้นล้มเหลวได้

จากการศึกษาผลงานวิจัยที่มีอยู่ในปัจจุบันที่มีการนำเอาโพรโทคอลมัลติคาสท์ไปใช้ในการส่งผ่านข้อมูลแบบมัลติมีเดีย ทำให้เกิดคำถามตามมาเกี่ยวกับประสิทธิภาพที่ได้ เพราะการที่มัลติคาสท์ต้องพยายามทำตัวให้มีพฤติกรรมแบบเดียวกันกับทีซีพี ในขณะที่ทำการส่งผ่านข้อมูลมัลติมีเดียที่มีคุณสมบัติในการสื่อสารที่แตกต่างกันอย่างสิ้นเชิง ดังนั้นงานวิจัยนี้จึงได้ทำการศึกษาประเด็นปัญหาที่เกิดขึ้นจากการที่มัลติคาสท์ต้องปฏิบัติตามกฎความเป็นมิตรต่อทีซีพีเมื่อให้บริการมัลติมีเดีย เพื่อเป็นแนวทางในการพัฒนาโพรโทคอลใหม่ให้มีประสิทธิภาพยิ่งขึ้นต่อไป

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาและวิเคราะห์ปัญหาจากการปฏิบัติตามกฎความเป็นมิตรต่อทีซีพีของโพรโทคอล MR-MCC
2. เพื่อประเมินประสิทธิภาพของโพรโทคอล MR-MCC เมื่อให้บริการมัลติมีเดียภายใต้กฎความเป็นมิตรต่อทีซีพี

1.3 ความสำคัญของการวิจัย

1. ได้ทราบถึงปัญหาที่เกิดขึ้นเมื่อโพรโทคอล MR-MCC ให้บริการมัลติมีเดียภายใต้กฎความเป็นมิตรต่อทีซีพี
2. ได้องค์ความรู้ใหม่ในการสะท้อนปัญหาของกฎความเป็นมิตรต่อทีซีพีที่ควรมีการทบทวนใหม่

3. ได้ผลการทดสอบการทำงานของโพโทคอล MR-MCC ในประเด็นความเป็นมิตรต่อทีซีพี

4. ได้ทราบถึงปัญหาที่โพโทคอล MR-MCC ต้องคำนึงถึงเมื่อให้บริการมัลติมีเดียในระบบเครือข่ายเดียวกันกับทีซีพี

5. ได้ทราบถึงปัญหาที่โพโทคอล MR-MCC ต้องคำนึงถึงเมื่อให้บริการมัลติมีเดีย เพื่อไม่ให้ส่งผลกระทบต่อคุณภาพของมัลติมีเดีย

1.4 ขอบเขตของการวิจัย

ในการวิจัยนี้จะมุ่งเน้นศึกษาวิเคราะห์โพรโทคอลในระดับชั้นการขนส่ง (Transport Layer) จุดหมายหลักจะอยู่ที่การเผยให้เห็นถึงข้อปัญหาในการให้บริการมัลติมีเดียของโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราส่ง (Multi-Rate) ภายใต้กฎความเป็นมิตรต่อทีซีพี โดยทำการศึกษาวิจัยผ่านทางตัวจำลองการทำงานของเครือข่าย (ns-2) [5]

1.5 นิยามศัพท์เฉพาะ

1. ชั้นข้อมูล หมายถึง ข้อมูลที่ถูกเข้ารหัสเป็นชั้นก่อนทำการส่ง

2. มัลติคาสท์ หมายถึง การสื่อสารแบบจุดต่อหลายจุด (1 ถึง N; N=ทั้งหมด)

3. ความคับคั่ง หมายถึง สภาพความแออัดของระบบเครือข่ายที่เกิดจากจำนวนข้อมูลที่ต้องการส่งมีมากกว่าช่องทางในการส่ง

4. กลไกการควบคุมความคับคั่งแบบหลายอัตราส่ง (MR-MCC) หมายถึง การควบคุมความคับคั่งในมัลติคาสท์แบบหลายอัตราส่งที่ผู้รับสามารถเลือกรับข้อมูลได้ตามสภาวะระบบเครือข่ายของตนเอง

5. มัลติมีเดียทราฟฟิก หมายถึง ข้อมูลสื่อประสมทั้งภาพและเสียงที่มีการส่งผ่านระบบเครือข่าย

6. กฎความเป็นมิตรต่อทีซีพี หมายถึง นิยามของ IETF ว่าด้วยการพัฒนาโพรโทคอลขึ้นมาใช้งานบนอินเทอร์เน็ตร่วมกับทีซีพีซึ่งเป็นโพรโทคอลหลัก โดยโพรโทคอลที่พัฒนาขึ้นมาใหม่ต้องไม่ใช้แบนด์วิดท์ของระบบมากกว่าทีซีพี [3]

บทที่ 2

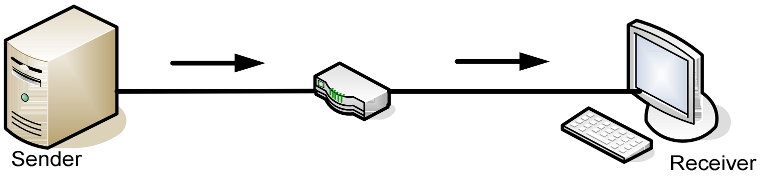
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้จะอธิบายถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องอันเป็นความรู้พื้นฐานของงานวิจัย ซึ่งจะกล่าวถึงลักษณะของการสื่อสารข้อมูล มาตรฐาน หน้าที่และความสามารถของโพรโทคอลที่สำคัญ ที่นำมาใช้ในการทดลอง สถานการณ์อันเป็นต้นกำเนิดของปัญหา ข้อมูลงานวิจัยที่เกี่ยวข้องในการออกแบบและแก้ไขปัญหาที่เกิดขึ้น ฐานความรู้อันจะนำไปสู่การทดลอง ในการประเมินประสิทธิภาพโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราส่งของมัลติคาสท์ที่ให้บริการมัลติมีเดียภายใต้กฎความเป็นมิตรต่อทีซีพี โดยมีรายละเอียดดังนี้

2.1 รูปแบบการสื่อสารข้อมูล

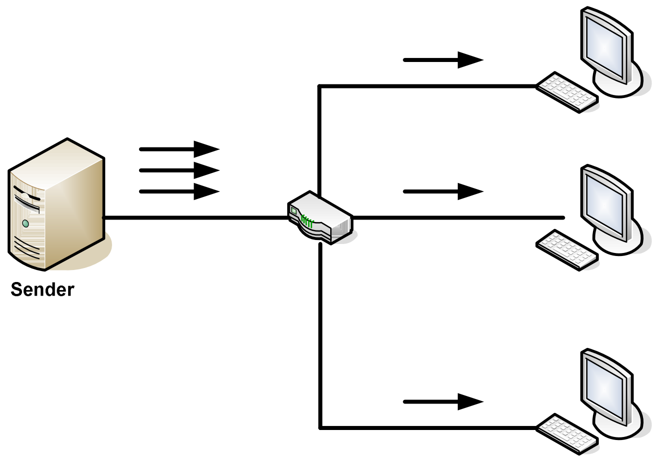
รูปแบบของการสื่อสารข้อมูล (Mode of Communication) [2] แบ่งออกได้ 6 ประเภทใหญ่ๆ คือ ยูนิคาสท์ (Unicast) บรอดคาสท์ (Broadcast) แนโรว์คาสท์ (Narrowcast) มัลติคาสท์ (Multicast) เอนี่คาสท์ (Anycast) และ คอนคาสท์ (Concast) ในงานวิจัยนี้จะมุ่งเน้นการสื่อสารในรูปแบบของยูนิคาสท์และมัลติคาสท์เท่านั้น เพราะเป็นรูปแบบที่ใช้สื่อสารกันมากที่สุดบนเครือข่ายอินเทอร์เน็ตในปัจจุบัน

2.1.1 ยูนิคาสท์ (Unicast) เป็นรูปแบบวิธีการสื่อสารที่มีการรับส่งข้อมูลจากโหนดส่งหนึ่งโหนดไปยังโหนดปลายทางหนึ่งโหนด หรืออยู่ในลักษณะจุดต่อจุด (One-to-One) การทำงานจะอยู่ในระดับชั้น Transport ของตัวแบบโอเอสไอ (OSI Model) หรือในระดับชั้นทีซีพี (TCP) ของตัวแบบทีซีพี/ไอ(TCP/IP Model) หน้าที่การทำงานคือจะทำการส่งข้อมูลระหว่างโหนดต่อโหนดเท่านั้น ไม่พิจารณาถึงเส้นทางว่าแพ็คเก็ต (Packet) ข้อมูลจะถูกไปตามเส้นทางใด การส่งข้อมูลแบบยูนิคาสท์ผ่านเครือข่ายไปยังปลายทางนั้นต้องส่งผ่านเราเตอร์ (Router) ต่างๆมากมาย โดยเราเตอร์จะทำการส่งต่อแพ็คเก็ตข้อมูล ไปยังปลายทางที่กำหนดในลักษณะดังภาพประกอบ 1



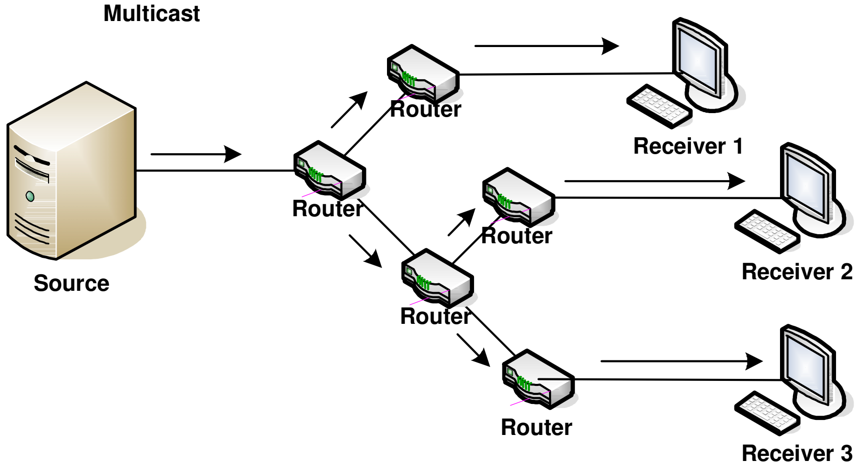
ภาพประกอบ 1 การส่งข้อมูลแบบยูนิคาสท์

การส่งข้อมูลแบบยูนิคาสท์ จะทำการส่งข้อมูลแบบเจาะจงผู้รับปลายทางที่ใดที่หนึ่ง ถ้าหากปลายทางมีหลายๆจุด ต้นทางจำเป็นต้องมีการส่งข้อมูลเดิมซ้ำๆ โดยกำหนดที่อยู่ (Address) ปลายทางของแต่ละแพ็คเก็ตต่างๆกันไปในลักษณะดังภาพประกอบ 2 เรียกว่าการสื่อสารแบบยูนิคาสท์หลายๆจุด (Multiple Unicast) การสื่อสารแบบนี้จะทำให้ใช้แบนด์วิดท์ (Bandwidth) มากขึ้นตามจำนวนของโหนดปลายทางที่ต้องการรับข้อมูลดังนั้นอาจทำให้เกิดปัญหาการจราจรที่หนาแน่นบนเครือข่าย (Network load Problem) ขึ้นได้

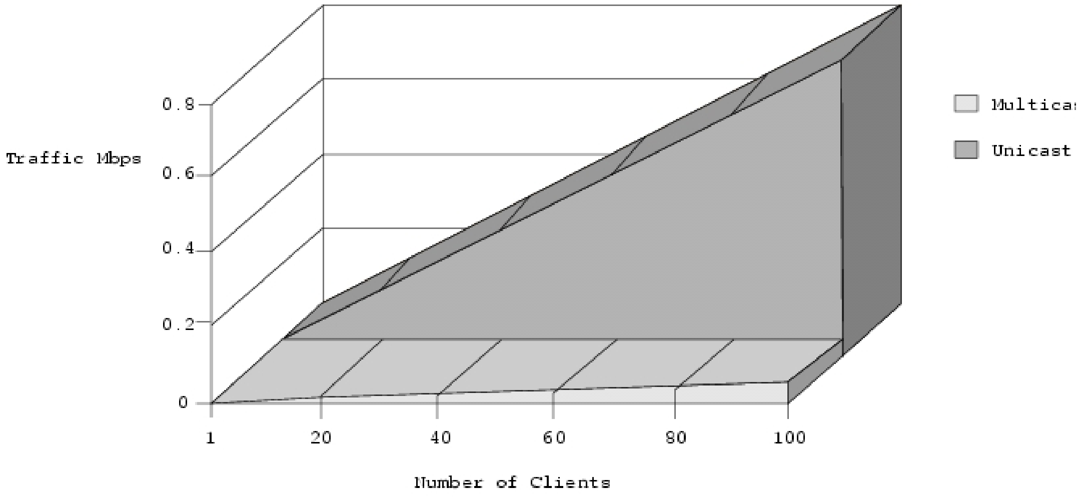


ภาพประกอบ 2 Multiple Unicast

2.1.2 มัลติคาสท์ (Multicast) เป็นรูปแบบการส่งข้อมูลแบบกระจาย ที่ทำการส่งจากจุดหนึ่งไปยังปลายทางหลายๆจุดโดยการส่งเพียงครั้งเดียว ทั้งนี้ในโหมดการแพร่สัญญาณเฉพาะกลุ่ม (Multicast Mode) ผู้ส่งจะทำการส่งชุดข้อมูล (Data Packet) เข้าไปบนเส้นทางระบบเครือข่าย (Network Path) ใดๆ เพียงชุดเดียวและครั้งเดียวจากนั้นจะอาศัยความสามารถพิเศษของ Router ในการทำซ้ำข้อมูลเพื่อส่งไปยังผู้รับปลายทางหลายๆจุด โดยการทำซ้ำและส่งต่อข้อมูลจะส่งเพียงหนึ่งชุดข้อมูลต่อหนึ่ง Network Link เท่านั้นดังภาพประกอบ 3



ภาพประกอบ 3 รูปแบบการส่งข้อมูลของมัลติคาสท์



ภาพประกอบ 4 แสดงการเปรียบเทียบ Traffic ระหว่าง Unicast กับ Multicast [6]

จากภาพประกอบ 4 แสดงการเปรียบเทียบภาระงานในเครือข่าย (Network Load) ระหว่างการส่งผ่านข้อมูลแบบยูนิคาสท์กับมัลติคาสท์ โดยจากภาพเป็นผลจากการทดลองของบริษัท Cisco [6] เป็นการส่งไฟล์เสียงแบบต่อเนื่อง (Stream) ที่อัตรา 8 kbps โดยเป็นการส่งจากผู้ส่ง 1 คน ไปยังผู้รับหลายคน (เป็นการส่งหลาย ๆ แบบตั้งแต่ 1 ถึง 100 คน) โดยใช้วิธีการส่งผ่านข้อมูลทั้งแบบมัล

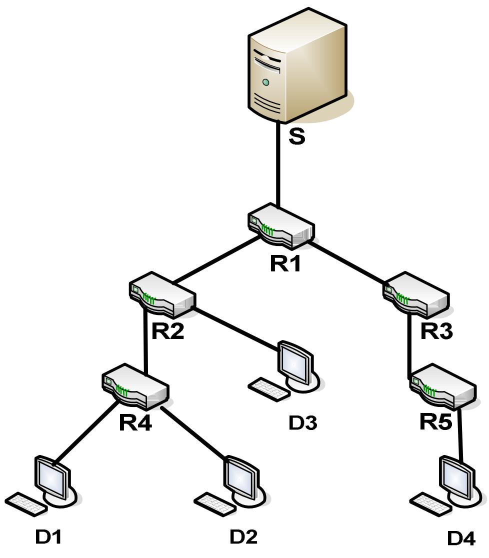
ติคาสท์และยูนิคาสท์ จากผลการทดลองพบว่ามัลติคาสท์ใช้แบนด์วิดท์น้อยกว่าการส่งผ่านข้อมูลโดยใช้แบบ Multiple Unicast มาก โดยเฉพาะเมื่อจำนวนของผู้รับเพิ่มมากขึ้น

2.2 รูปแบบการแพร่สัญญาณเฉพาะกลุ่มบนอินเทอร์เน็ต (Internet Multicast Model)

คือรูปแบบการรับส่งข้อมูลของ Internet Protocols (IP) ไปยังกลุ่มของ End host และมีหมายเลข IP เป็นที่ใช้ชี้ไปยังกลุ่มของ end host ซึ่งเรียกว่า Multicast Address ทั้งนี้ IP Multicast ใช้วิธีในการรับ-ส่งข้อมูลเหมือนกับ Unicast IP

สำหรับ IPv4 ใช้ Internet Group Management Protocols (IGMP) [7] สำหรับควบคุมการรับส่งข้อมูล ส่วน IPv6 นั้นใช้ Internet Control Message Protocols (ICMP) ซึ่งเป็นการรวมเอาความสามารถของ IGMP เข้ามาไว้แล้ว

IP Multicast นั้นถูกควบคุมโดย Multicast Routing Algorithms ซึ่งออกแบบมาสำหรับ Multicast Routers ซึ่งปัจจุบันอัลกอริทึมของ Multicast Routing ได้ถูกนำเสนอมาแล้วหลายวิธี เช่น Distance-Vector Multicast Routing Protocol (DVMRP) [8], Protocol Independent Multicast dense Mode (PIM-DM) [7], Protocol Independent Multicast Source Specific Mode (PIM-SSM) [9] และ Protocol Independent Multicast Sparse Mode (PIM-SM) [10] เป็นต้น



ภาพประกอบ 5 Sample of the IP Multicasting [2]

จากภาพประกอบ 5 Sender (S) ทำการส่งข้อมูลแบบ Multicast ไปกลุ่มยังผู้รับ (M) โดย sender ใช้ Multicast address ของ M เป็นจุดหมายในการส่งข้อมูลและถ้าหากว่าไม่มี Receiver เข้าร่วมกับ Multicast Group การส่งข้อมูลจาก Sender ไปยัง M นั้นก็จะถูกส่งไปยัง Router (R1) เพียงอันเดียว ซึ่งก็จะไม่มีการส่งข้อมูลต่อถ้าหากว่า D1 ต้องการจะรับข้อมูล D1 ก็จะทำการ Join กับ Multicast Group (M) โดยการส่งสัญญาณ IGMP JOIN สำหรับ Multicast Session (S, M) ไปยัง Router ที่ใกล้ที่สุด (R4) Router (R4) ก็จะทำการส่ง IGMP Join Message ไปยัง Router (R2) เพื่อเข้าร่วมกับ Multicast Group (M) แล้วจึงทำการเชื่อมโยงระหว่าง R4 to R2 หลังจากนั้น R2 ก็จะทำการส่งสัญญาณ IGMP Join Message ไปยัง R1 แล้วจึงทำการเชื่อมโยงระหว่าง R2 to R1 ซึ่งถือได้ว่าเป็นการเชื่อมต่อกันอย่างสมบูรณ์ระหว่าง S กับ D1 ซึ่งทำให้ D1 สามารถที่รับข้อมูลจาก S ถ้าหาก D2 ต้องการที่จะเข้าร่วมกับ Session (S, M) ก็จะทำการส่ง IGMP Join ไปยัง Router ที่อยู่ใกล้ที่สุด (R4) หลังจากนั้น R4 ก็จะทำการส่งข้อมูลไปยัง D2

หลังจากนั้นเมื่อ D1 ไม่ต้องการที่จะรับข้อมูลจาก Multicast Session (S, M) ก็จะทำการส่ง IGMP Leave ไปยัง Router ที่ใกล้ที่สุด (R4) ในขณะเดียวกัน D2 ยังต้องการที่จะรับข้อมูลนั้นๆต่อ Router (R4) ก็จะทำการหยุดส่งข้อมูลระหว่าง R4 กับ D1 ส่วน R4 กับ D2 ก็จะทำการส่งข้อมูลอยู่

ถ้าหาก D2 ต้องการที่จะออกจาก Multicast Session (S, M) Router (R4) จะต้องทำการตรวจสอบไปยังผู้รับว่า ไม่มีผู้รับที่ต้องการข้อมูลนั้นอีกแล้วโดยการส่ง IGMP Query ไปยังผู้รับที่อยู่ใน Subnet ซึ่งถ้าหากว่า ไม่มีการส่งสัญญาณตอบรับกลับมา Router (R4) ก็จะส่งสัญญาณไปยัง Router (R2) เพื่อให้หยุดการส่งข้อมูลมายัง Router (R4) ส่วน R2 นั้นยังคงอยู่ใน Multicast Session (S, M) เนื่องจาก D3 นั้นยังคงอยู่ใน Multicast Session (S, M)

2.3 Internet Application

สามารถแบ่งซอฟต์แวร์ที่ทำงานอยู่บนอินเทอร์เน็ตออกเป็นสองกลุ่มที่มีชื่อว่าโปรแกรมประยุกต์แบบน่าเชื่อถือ (Reliable Application) และ โปรแกรมประยุกต์แบบสื่อประสม (Multimedia Application) ตัวอย่างของ Reliable Application ก็ได้แก่ ฐานข้อมูลแบบกระจาย การกระจายข้อมูลข่าวสารของตลาดหุ้นและโปรแกรมส่งเนื้อหาอื่นๆ โปรแกรมเหล่านี้ต้องการความน่าเชื่อถือของข้อมูล เพราะจะอ่อนไหวต่อการสูญเสียข้อมูล ทำให้ต้องการกลไกการควบคุมความเสียหายของข้อมูลในระหว่างการส่งเพื่อทำให้แน่ใจว่าข้อมูลที่สูญหายได้รับการแก้ไข ส่วนข้อมูลมัลติมีเดียนั้นจะมีขนาดใหญ่และต้องการความต่อเนื่องในการส่งข้อมูลทำให้การส่งข้อมูลแบบยูนิคาสท์ไม่สามารถรองรับได้ดี เกิดการนำเอารูปแบบการส่งข้อมูลแบบมัลติคาสท์มาใช้งานมากขึ้นในปัจจุบัน

ตาราง 1 ขอบเขตของความต้องการของซอฟต์แวร์ต่างๆ [2]

Requirement	Application
In-sequence+Reliable+Non Real-time	FTP and other TCP-style application
In-sequence+Reliable+Delay sensitive	Financial and stock exchange application
Out of sequence + practically reliable+Delay sensitive	Video player application
Out of sequence + reliable+Non real time	Image server
Out of sequence + non reliable+Non real time	UDP style application
In sequence+Partically reliable+delay sensitive+jitter sensitive	Audio and Video conference

จากตาราง 1 เป็นตัวอย่างซอฟต์แวร์ที่ทำงานบนเครือข่ายอินเทอร์เน็ต ซึ่งซอฟต์แวร์ที่ทำงานต่างๆกันนั้นก็จะต้องการคุณสมบัติในการเชื่อมต่อที่ต่างกันไปด้วย ซึ่งจากเงื่อนไขดังกล่าว โพรโทคอลที่ทำงานอยู่ในระดับ Transport Layer (เช่น TCP และ UDP) ในปัจจุบันยังไม่สามารถสนับสนุนได้ดีพอ ตัวอย่างเช่น TCP มีคุณสมบัติในการส่งข้อมูลที่มีการรับประกันการส่ง (reliable) และมีการจัดเรียง แพ็คเก็ตข้อมูล (Packet Ordering) ส่วน UDP นั้นมีคุณสมบัติในการส่งข้อมูลแบบไม่จัดเรียงแพ็คเก็ตข้อมูล (Non-Ordered) และไม่รับประกันความสมบูรณ์ของข้อมูล (Non-Reliable) ทำให้ไม่สามารถรองรับกับความต้องการของซอฟต์แวร์ที่ต้องการคุณสมบัติต่างๆ อย่างหลากหลายได้ ซึ่งปัญหานี้เรียกได้ว่าปัญหา “One size does not fit all” [11] ที่โพรโทคอลหนึ่งไม่สามารถที่จะรองรับการส่งข้อมูลของทุกๆซอฟต์แวร์ได้ ดังนั้น โพรโทคอลแบบมัลติคาสท์จึงต้องมีคุณสมบัติที่จะสนับสนุนกับความต้องการที่หลากหลายของแต่ละซอฟต์แวร์ให้ได้

2.3.1 Multimedia Application

เป็นซอฟต์แวร์ที่ใช้บริการทางด้านมัลติมีเดียผ่านทางอินเทอร์เน็ต โดยที่ซอฟต์แวร์ประเภทนี้ ยอมให้เกิดความสูญหายหรือผิดพลาดของข้อมูลได้บ้าง สามารถแบ่งซอฟต์แวร์ที่ใช้บริการทางด้านมัลติมีเดียออกได้เป็น 3 กลุ่ม [11] ได้แก่

2.3.1.1 Streaming Stored Multimedia เป็นซอฟต์แวร์ที่มีลักษณะการทำงานในรูปแบบที่ไฟล์มัลติมีเดีย จะต้องถูกบีบอัด (Compress) และถูกจัดเก็บไว้ที่เซิร์ฟเวอร์ แล้วให้ผู้ใช้เรียกใช้งานไฟล์เหล่านั้น โดยที่ซอฟต์แวร์ประเภทนี้มีคุณลักษณะการทำงานที่สำคัญ 3 ประการคือ

(1) Stored Media ไฟล์มัลติมีเดียจะต้องถูกบันทึกล่วงหน้าและจัดเก็บไว้ที่เซิร์ฟเวอร์ เมื่อผู้ใช้เรียกใช้งานไฟล์มัลติมีเดียที่อยู่บนเซิร์ฟเวอร์ ผู้ใช้สามารถที่จะเล่นไฟล์เหล่านั้นโดยสามารถที่จะหยุดชั่วคราว (Pause) เล่นย้อนกลับ (Rewind) ไปข้างหน้า (Fast-Forward)

(2) Streaming คือ การที่ผู้ใช้ดาวน์โหลดไฟล์มัลติมีเดียบางส่วนมาแล้ว สามารถเล่นไฟล์นั้นๆ ได้เลยโดยไม่ต้องรอดาวน์โหลดไฟล์ทั้งหมดมาให้ครบก่อน โดยที่ในไฟล์มัลติมีเดียนั้นจะต้องมีการกำหนดว่าในแต่ละส่วนที่รับมา คือส่วนไหนจากทั้งหมดของไฟล์ โดยในปัจจุบันมีโปรแกรมที่ทำงานในลักษณะนี้ หลายโปรแกรม เช่น Real Player, Apple’s Quick Time และ Microsoft –Windows Media Player เป็นต้น

(3) Continuous Play out คือ ลักษณะการที่เมื่อมีการเล่นไฟล์มัลติมีเดียขึ้นมาแล้ว ไฟล์ที่ผู้ใช้เรียกขึ้นมาใช้งานนั้นจะต้องถูกเล่นไปตามเวลาของไฟล์นั้นจริงๆ ซึ่งอาจทำให้เกิดการดีเลย์ขึ้นได้ ดังนั้นโปรแกรมที่ต้องการความเร็ว เช่น Internet Telephony และ Video Conferencing จึงไม่เหมาะที่จะใช้งานในลักษณะนี้

2.3.2 Streaming Live Multimedia เป็นซอฟต์แวร์ ที่มีลักษณะการทำงานคล้ายกับการแพร่ภาพของสถานีวิทยุ หรือสถานีโทรทัศน์ โดยที่ซอฟต์แวร์ประเภทนี้อนุญาตให้ผู้ใช้สามารถรับไฟล์ได้ทันทีในขณะนั้น กล่าวคือเมื่อมีการสร้างไฟล์หรือผลิตไฟล์นั้นๆ ขึ้นมา ผู้ใช้ก็สามารถรับไฟล์นั้นๆไปได้เลย โดยตัวอย่างการทำงานของโปรแกรมในลักษณะนี้ เช่น สถานีวิทยุบนอินเทอร์เน็ตหรือการถ่ายทอดสดเสียงผ่านทางอินเทอร์เน็ตเป็นต้น

2.3.3 Real-Time Interactive Media

เป็นซอฟต์แวร์ที่ใช้ในการสื่อสารโต้ตอบกันระหว่างผู้ใช้งาน 2 ฝั่ง ผ่านทางอินเทอร์เน็ต ตัวอย่างของซอฟต์แวร์ประเภทนี้ เช่น IP Telephony (Video Conference) โดยที่ซอฟต์แวร์ที่ทำงานในลักษณะนี้จะต้องคำนึงถึงเรื่องของเวลาเป็นสำคัญ คือจะต้องให้มีการดีเลย์ในการส่งข้อมูล (Delay-Sensitive) ระหว่าง 2 ฝั่งให้น้อยที่สุด

2.4 การส่งผ่านข้อมูลแบบมัลติมีเดีย (Multimedia Streaming)

คุณสมบัติสำคัญที่ต้องการในการส่งผ่านข้อมูลมัลติมีเดียนั้น มีคุณสมบัติสำคัญที่ต้องการในการส่งผ่านข้อมูล [11] ดังนี้

2.4.1 Delay-Sensitive (การตอบสนองที่ไวต่อความล่าช้า) หมายถึง คุณสมบัติในการส่งข้อมูลที่จะต้องให้เกิดความล่าช้าในการส่งข้อมูลระหว่างผู้ส่งและผู้รับให้น้อยที่สุด ซึ่งชนิดของการเกิดความล่าช้านั้น มีอยู่ 2 ชนิดคือ

2.4.1.1 End -to- end delay คือความล่าช้าที่เกิดขึ้นยังโหนดของการส่งข้อมูลทั้งฝั่งผู้ส่ง และผู้รับ ซึ่งสาเหตุอาจเกิดได้หลายประการ เช่น ความเร็วในการประมวลผล เป็นต้น

2.4.1.2 Delay jitter คือ ความล่าช้าที่หลากหลายของแพ็คเก็ตข้อมูลที่ส่งมาในสตรีมแพ็คเก็ตเดียวกัน

2.4.2 Loss-Tolerant (ยอมให้ข้อมูลสูญหายได้) หมายถึง คุณสมบัติที่การสูญหายของข้อมูลบางส่วนไม่มีผลกระทบมากจนทำให้ข้อมูลทั้งหมดของชุดนั้นไม่สามารถทำงานได้ เช่น การสูญหายของแพ็คเก็ตที่ส่งข้อมูลที่เป็นไฟล์เสียงมาได้สูญเสียไปบางส่วน แต่ไฟล์เสียงนั้นก็ยังสามารถฟังได้

ในการส่งผ่านข้อมูลแบบมัลติมีเดียบนอินเทอร์เน็ตนั้น นอกจากต้องคำนึงถึงคุณสมบัติสำคัญข้างต้นดังที่ได้กล่าวมาแล้ว ยังต้องคำนึงถึงความต้องการแบนด์วิดท์(Bandwidth require) และคุณภาพของการบริการด้วย (Quality of Service: QoS)

1. ความต้องการแบนด์วิดท์

จากที่กล่าวมาในข้างต้นแสดงให้เห็นว่า ข้อมูลมัลติมีเดียนั้นต้องการคุณภาพการให้บริการ ซึ่งข้อมูลมัลติมีเดียแต่ละชนิดก็มีความต้องการขนาดของแบนด์วิดท์ที่แตกต่างกันออกไป ซึ่งสามารถสรุปแยกออกตามประเภทข้อมูลได้ดังตาราง 2 ดังนี้

ตาราง 2 แสดงความต้องการแบนด์วิดท์ของข้อมูลมัลติมีเดียและคุณภาพที่ได้รับ [12]

ชนิดข้อมูลมัลติมีเดีย	แบนด์วิดท์ที่ต้องการ	คุณภาพที่ได้รับ
Audio (MP3)	32 kbit/s	MW (AM) quality
	96 kbit/s	FM quality
	128–160 kbit/s	Standard Bit rate quality; difference can sometimes be obvious (e.g. bass quality)
	192 kbit/s	DAB (Digital Audio Broadcasting)
	224–320 kbit/s	Near CD Quality
Other audio	800 bit/s	minimum necessary for recognizable speech (using special-purpose FS-1015 speech codec)
	8 kbit/s	telephone quality (using speech codec)

ตาราง 2 (ต่อ)

	500-1024 kbit/s	lossless audio as used in formats such as FLAC, WavPack or Monkey's Audio
	1411 kbit/s	PCM sound format of Compact Disc Digital Audio
Video (MPEG2)	16 kbit/s	videophone quality (minimum necessary for a consumer-acceptable "talking head" picture)
	128 -320 kbit/s	business-oriented videoconferencing system quality
	1 Mbit/s	VHS quality
	5 Mbit/s	DVD quality
	15 Mbit/s	HDTV quality
	36 Mbit/s	HD DVD quality
	54 Mbit/s	Blue-ray Disc quality

2. คุณภาพการให้บริการ Quality of Service (QoS)

จากที่ได้กล่าวในบทนำ ได้แสดงให้เห็นว่าปัจจุบันการใช้งานข้อมูลมัลติมีเดียบนอินเทอร์เน็ตได้รับความนิยมอย่างมาก แต่จากการที่อินเทอร์เน็ตในยุคแรกเน้นการรับส่งข้อมูลระหว่างกันเป็นหลัก โดยไม่คำนึงถึงคุณภาพการให้บริการบางประการ โดยเฉพาะการกำหนดให้ไอพีแพ็กเก็ตมีลักษณะเป็นดาต้าแกรม คือ การจ่าหน้าแอดเดรสปลายทางให้กับแพ็กเก็ตแล้วส่งไปโดยไม่คำนึงว่าจะถึงปลายทางเมื่อไร การให้บริการในลักษณะนี้จึงเหมาะสมกับการบริการบางประเภท เช่น อีเมล์ ซึ่งจะถึงล่าช้าไปบ้างไม่เป็นไร แต่ในปัจจุบันการให้บริการบนอินเทอร์เน็ตมีหลายอย่างที่ต้องการคุณภาพ เช่น การส่งข้อมูลเสียงการใช้งานแบบ Voice over IP การใช้เป็น IP-TV การใช้งานเป็น IP-Phone การใช้งานมัลติมีเดียอื่นๆเช่น ระบบวิดีโอคอนเฟอเรนซ์ เป็นต้น งานเหล่านี้ต้องได้คุณภาพจึงจะใช้งานได้ตามวัตถุประสงค์

การกำหนดคำว่า QoS - Quality of Service จึงเป็นคำที่มีการกล่าวถึงกันมากในยุคนี้เป็นมาตรฐานที่ใช้ในการวัดการรับส่งข้อมูล การยอมรับในเรื่องคุณภาพ ซึ่งเป็นระดับที่บางครั้งก็ยากแก่ การกำหนดว่ามาตรฐานคุณภาพบนอินเทอร์เน็ตเป็นอย่างไร แต่หากพิจารณา QoS ในเชิงเทคนิคพอจะนิยามได้ด้วยพารามิเตอร์หลาย ๆ ตัวดังนี้

1. การมีให้ใช้งานได้ (Availability) ในทางอุดมคติต้องได้ 100 เปอร์เซ็นต์ของเวลาการใช้งานหรือกล่าวได้ว่า เวลาของดาวน์ไทม์ (Downtime) มีค่าเป็นศูนย์ แต่แน่นอนที่สุดว่าไม่มีเครือข่ายใดจะให้บริการได้ 100 เปอร์เซ็นต์ หากกำหนดค่าควรมีให้ใช้งานได้เป็น 99.9999 เปอร์เซ็นต์ ก็แปลความได้ว่าในหนึ่งเดือนจะต้องมีเวลาดาวน์ไทม์ หรือเครือข่ายขัดข้องได้ไม่เกิน 2.6 วินาที

2. ช่องสัญญาณที่ส่งได้ (Throughput) ค่านี้มีความหมายถึง การรับส่งข้อมูลจากปลายหนึ่งไปยังอีกปลายหนึ่งได้ด้วยอัตราเท่าไรในจำนวนบิตต่อวินาที ค่านี้มิได้หมายถึงค่าสูงสุดของช่องสัญญาณที่จะรับส่งได้ ทั้งนี้เพราะช่องสัญญาณที่ใช้รับส่งได้มีแพ็กเก็ตและข้อมูลของผู้อื่นร่วมอยู่ด้วยช่องสัญญาณของผู้ส่งต่อกับผู้รับมีลักษณะการส่งรวมกับผู้อื่น ค่า Throughput นี้อาจใช้ค่าที่ ISP รับประกันช่องสัญญาณน้อยที่สุดที่จะต้องส่งได้ หรือที่เรียกว่า Minimum throughput guarantee เช่น เช่าสายวงจรเช่าขนาด 64 กิโลบิตต่อวินาที แต่มีการรับประกันว่าจะใช้ได้ไม่ต่ำกว่า 32 กิโลบิตต่อวินาที ค่า 32 จึงเป็นค่า Throughput แต่เมื่อใช้กับงานบริการเฉพาะด้านจำเป็นต้องมีการประกันช่องสัญญาณ เช่น การส่งสัญญาณเสียง สัญญาณวิดีโอ เป็นต้น

3. การหายของแพ็กเก็ต (Packet Loss) เมื่อพิจารณาที่สวิตช์หรือเราท์เตอร์ที่ต้องรับแพ็กเก็ตไว้เป็นจำนวนมากแต่ไม่สามารถให้บริการได้ทัน จำเป็นต้องนำแพ็กเก็ตบางส่วนทิ้งไป แพ็กเก็ตที่หายไปโดยไม่สามารถส่งจากผู้ส่งไปยังผู้รับได้เรียกว่า ค่าการหายของแพ็กเก็ต เมื่อแพ็กเก็ตหายไปก็ต้องส่งใหม่ ซึ่งจะทำให้ปริมาณของข้อมูลยิ่งเพิ่มมากขึ้น การเพิ่มมากขึ้นก็มีโอกาสที่จะสูญหายมากขึ้นด้วยเช่นกัน

4. เวลาลาเทนซี่ (Latency) ค่าเวลาลาเทนซี่เป็นค่าเวลาที่เกิดจากการเดินทางของแพ็กเก็ตข้อมูลจากต้นทางไปยังปลายทาง เช่น ข้อมูลอินเทอร์เน็ตเดินทางจากต้นทางไปยังเซิร์ฟเวอร์ที่อยู่ห่างออกไป 5,000 กิโลเมตร ต้องใช้เวลาไม่น้อยกว่า 150 มิลลิวินาที ค่า 150 มิลลิวินาทีที่วัดได้เป็นค่าลาเทนซี่ ค่าลาเทนซี่จะขึ้นกับวิธีการและเทคโนโลยี รวมถึงสูญเสียเวลาจากการหน่วงในอุปกรณ์เราเตอร์และสวิตชิ่งด้วย เช่น สัญญาณเดินทางผ่านดาวเทียมซึ่งต้องเสียเวลาเพิ่มขึ้นเพราะดาวเทียมอยู่ห่างไกล

5. เวลาจิตเตอร์ (Jitter) อาจกล่าวได้ง่าย ๆ ว่าเป็นค่าการปรวนแปรของค่าเวลาลาเทนซี่ กล่าวคือ แพ็กเก็ตที่เคลื่อนที่จากต้นทางไปยังปลายทางหลายๆแพ็กเก็ต ปรากฏว่าการไปถึงปลายทางใช้ระยะเวลาต่างกันทำให้ข้อมูลบางส่วนที่ไปก่อนอาจถึงทีหลังหรือมีเวลาเหลื่อมกัน ทำให้การตรวจสอบลำดับของแพ็กเก็ตในผู้รับต้องกระทำด้วยข้อมูลที่ใช้ในการรับส่งบนเครือข่ายอินเทอร์เน็ตจึงมีลักษณะที่ต้องการในคุณสมบัติที่แตกต่างกัน ดังตาราง 3

ตาราง 3 แสดงการเปรียบเทียบข้อมูลบนอินเทอร์เน็ตที่มีความไวต่อพารามิเตอร์ต่าง ๆ

ชนิดของข้อมูล	Throughput	Packet loss	Latency	Jitter
อีเมล์	ต่ำ	สูง	ต่ำ	ต่ำ
telnet	ต่ำ	สูง	ปานกลาง	ต่ำ
วิดีโอคอนเฟอเรนซ์	สูง	สูง	ปานกลาง	สูง
มัลติคาสท์	สูง	สูง	สูง	สูง

2.5 Multicast Multimedia Streaming

ในการส่งข้อมูลของมัลติคาสท์นั้น ผู้ส่งมีหน้าที่ในการเข้ารหัสข้อมูลเป็นชั้นๆ (Layer) ก่อนทำการส่งโดยอัตราเร็วในการรับนั้น ผู้รับจะเป็นคนกำหนดเองว่าจะรับเอาข้อมูลกี่ชั้น ถ้ารับได้หลายชั้นก็จะทำให้ส่งข้อมูลได้เร็วและมีคุณภาพที่ดี แต่ถ้ารับได้เพียงแค่ชั้นพื้นฐานก็จะทำให้รับข้อมูลได้ช้าและคุณภาพต่ำ โดยได้มีการจำแนกเป็น 3 รูปแบบ [13] คือ

2.5.1 Single stream with source rate adaptation เป็นวิธีการการส่งข้อมูลมัลติมีเดียที่เข้ารหัสแล้วไปในการส่งครั้งเดียว โดยที่การส่งข้อมูลฝั่งผู้ส่งจะทำการคำนวณอัตราการส่งที่จะใช้ในการส่งจาก Feedback ที่ทางผู้รับเป็นผู้ตอบกลับ ซึ่งทำให้เกิดปัญหาใหญ่ที่สำคัญคือปัญหาการเพิ่มขึ้นมากของ Feedback (Feedback Implosion) และจากการที่สภาพของการส่งข้อมูลในปัจจุบันที่มีความแตกต่างกันอย่างมากของจำนวนผู้รับ ทำให้ Feedback ที่จะเกิดขึ้นย่อมมีมาก ทำให้แบนด์วิดท์ที่จะใช้งานลดลงเนื่องจากต้องมาเสียไปให้กับ Feedback ดังนั้นวิธีการนี้จึงไม่นิยมนำมาใช้ในการส่งข้อมูลประเภทมัลติมีเดีย

2.5.2 Multiple streams with receiver switching เป็นวิธีการส่งที่หลีกเลี่ยงปัญหา Feedback Implosion โดยผู้ส่งทำการส่งข้อมูลมัลติมีเดียไปหลายๆ สตรีมโดยที่มีเนื้อหาเดียวกันแต่อัตราการส่งข้อมูลต่างกัน โดยแต่ละสตรีมทำการส่งในกลุ่มมัลติคาสท์ของตนเอง ผู้รับสามารถตอบรับสตรีมได้ตามขนาดของแบนด์วิดท์ที่ตนเองมีอยู่ และสามารถสลับการรับสตรีมได้ถ้าขนาดของแบนด์วิดท์เพิ่มขึ้นตามกำหนด แต่ละสตรีมมีความแตกต่างกันในการเข้ารหัสไฟล์มัลติมีเดียโดยใช้ค่าพารามิเตอร์ที่ต่างกัน

2.5.3 Layered Multicast Streaming เป็นการแบ่งข้อมูลออกเป็นชั้นๆ (Layer) โดยที่ข้อมูลจะมีเนื้อหาเดียวกันแต่แตกต่างกันที่คุณภาพ โดย Layered Multicast Streaming นี้สามารถแบ่ง

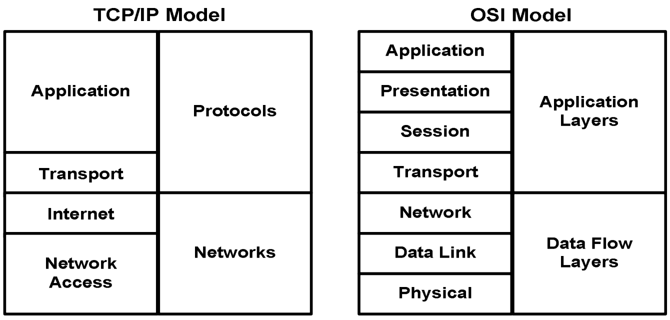
ออกได้เป็น 2 แบบคือ

2.5.3.1 Cumulative layering approach คือการเข้ารหัสชั้นข้อมูลแบบสะสม เป็นวิธีการส่งที่เข้าไฟล์มัลติมีเดียถูกรหัสเป็น 2-3 ชั้นที่ไม่ซ้ำกัน ซึ่งประกอบด้วยชั้นข้อมูลพื้นฐาน (Base-Layer) และชั้นข้อมูลเสริม (Enhancement Layer) โดยชั้นข้อมูลพื้นฐานสามารถถอดรหัสได้อย่างอิสระ แต่ชั้นข้อมูลเสริมไม่สามารถถอดรหัสได้ถ้าผู้รับไม่รับชั้นข้อมูลพื้นฐานก่อน ทำให้การเข้ารหัสของชั้นข้อมูลที่สูงกว่าจะขึ้นอยู่กับชั้นข้อมูลที่ต่ำกว่า (Base-Layer) โดยชั้นข้อมูลพื้นฐานคือชั้นข้อมูลที่มีความสำคัญที่สุดที่ประกอบด้วยคุณภาพพื้นฐานมัลติมีเดียนั้น ด้วยเหตุที่ชั้นข้อมูลเสริมถูกจัดเตรียมเพื่อเสริมคุณภาพให้กับชั้นข้อมูลพื้นฐาน ทำให้ผู้รับต้องทำการเข้าร่วมรับชั้นข้อมูลแบบชั้นสะสมโดยต้องเข้าร่วมรับชั้นข้อมูลพื้นฐานก่อนแล้วเข้ารับชั้นข้อมูลเสริมในชั้นสูงขึ้นไปเช่นในกรณีที่มีการเข้ารหัสสัญญาณเป็น 4 ชั้นข้อมูล ผู้รับต้องเข้าร่วมรับตามลำดับดังนี้ 1, 1-2,1-2-3,1-2-3-4

2.5.3.2 Non-Cumulative approach layering คือการเข้ารหัสชั้นข้อมูลแบบไม่สะสม เป็นวิธีการในการที่ไฟล์มัลติมีเดียถูกเข้ารหัสไว้ ตั้งแต่ 2 ชั้นข้อมูลขึ้นไป โดยที่ในแต่ละชั้นข้อมูลไม่มีความเกี่ยวเนื่องกันเลย กล่าวคือแต่ละชั้นข้อมูลจะเข้ารหัสโดยอิสระ โดย Multiple Description Coding (MDC) เป็นวิธีการในการใช้เข้ารหัส แต่ปัญหาใหญ่คือในการเข้ารหัสไฟล์ที่อยู่คนละชั้นข้อมูลกันจะต้องทำการสร้างไฟล์ต้นฉบับขึ้นมาใหม่ทุกครั้งทำให้ไม่เป็นที่นิยมในการใช้งาน

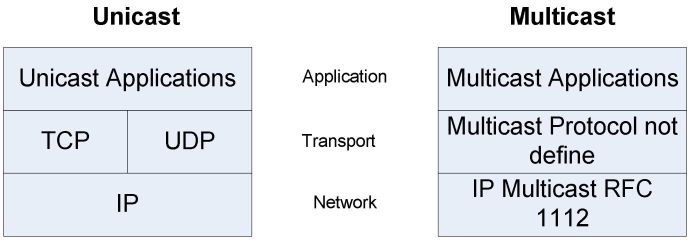
2.6 การควบคุมการส่งผ่านข้อมูลในระบบเครือข่าย

ในตัวแบบของ TCP/IP (TCP/IP Model) เมื่อเปรียบเทียบกับ OSI แล้วสามารถแบ่งชั้นของเลเยอร์ออกไปได้เป็น 4 ชั้นดังภาพประกอบ 6



ภาพประกอบ 6 TCP/IP Model vs. OSI Model [11]

จาก TCP/IP Model จะเห็นว่าสามารถแบ่งออกเป็น 2 ส่วนใหญ่ๆคือส่วนที่เป็น Protocols และส่วนที่เป็น Networks ส่วนของ Protocols จะทำหน้าที่ควบคุมการทำงานที่เกี่ยวข้องกับโปรแกรมประยุกต์ที่ต้องทำการรับส่งข้อมูลกับเครือข่าย ส่วนของ Networks จะทำหน้าที่ควบคุมทิศทางและความผิดพลาดจากการสื่อสารข้อมูล ในการวิจัยนี้จะมุ่งเน้นไปยังเลเยอร์ที่ทำหน้าที่ควบคุมการรับส่งข้อมูลระหว่างเครื่องต่อเครื่องให้ถูกต้องและมีประสิทธิภาพ (host-to-host) นั่นคือชั้น Transport Layer จากภาพประกอบ 7 นั้นหากพิจารณาถึงการทำงานในระดับชั้นต่างๆของทั้งยูนิคาสท์และมัลติคาสท์ จะเห็นได้ว่ามีมาตรฐานรองรับในชั้นต่างๆแล้ว แต่ในส่วนของมัลติคาสท์นั้นยังไม่มีโพรโทคอลมาตรฐานมารองรับ แต่อาศัยวิธีการที่ Multicast Application ใช้งาน IP Multicast โดยไม่ผ่าน Multicast Protocol ในระดับ Transport ที่เรียกว่า Overlaying แทน



ภาพประกอบ 7 เปรียบเทียบโพรโทคอลมาตรฐานในยูนิคาสท์และมัลติคาสท์

Transport Layer นั้นมีหน้าที่สำคัญคือการควบคุมการทำงานรับส่งข้อมูลระหว่างเครื่องต่อเครื่องโดยไม่ได้สนใจในเรื่องของสื่อที่ใช้ในการส่งผ่านข้อมูลว่าเป็นสื่อชนิดใด เพียงแต่ควบคุมการทำงานการรับส่งข้อมูลให้ถูกต้องและมีประสิทธิภาพ ดังนั้นหน้าที่ความรับผิดของของเลเยอร์นี้จะมีอยู่ 3 หน้าที่หลักๆ คือ

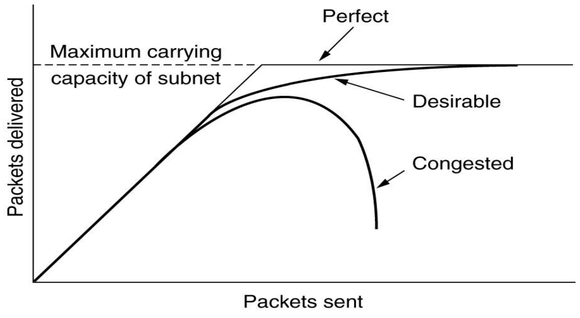
1. การควบคุมการไหลของข้อมูล (Flow Control) การควบคุมให้ข้อมูลมีการส่งได้อย่างสม่ำเสมอไม่ขาดช่วง โดยที่ฝั่งส่งจะต้องทำการส่งข้อมูลไปยังฝั่งรับไม่เกินกว่าความสามารถที่ฝั่งรับจะรับได้ เนื่องจากหากฝั่งรับไม่สามารถที่จะรับข้อมูลได้จะทำให้เกิดการสูญหายของข้อมูล โดยเทคนิคที่นำมาใช้ควบคุมเช่น Sliding Windows เป็นต้น

2. การควบคุมความผิดพลาดของข้อมูล (Error Control) เป็นการควบคุมความผิดพลาดของข้อมูลในการรับส่งระหว่างต้นทางและปลายทาง โดยใช้เทคนิคที่สำคัญได้แต่ Auto Repeat Request (ARQ) [11] คือการร้องขอให้ส่งข้อมูลใหม่หากมีการสูญหายของข้อมูลหรือข้อมูลมีความผิดพลาด และ Forward Error Correction (FEC) [11] คือการแก้ไขข้อมูลใหม่ หากข้อมูลที่ได้รับมีความผิดพลาด

3. การควบคุมความคับคั่งของข้อมูล (Congestion Control) เป็นการควบคุมความคับคั่งของข้อมูลที่อยู่ในเครือข่ายไม่ให้มีความคับคั่งมากจนไม่สามารถที่จะส่งข้อมูลระหว่างต้นทางและปลายทางได้ ทำให้เกิดการหยุดชะงักของการรับส่งข้อมูลจนอาจสามารถทำให้เครือข่ายล่มไปได้

2.7 ความคับคั่งของระบบเครือข่าย (Network Congestion)

“ความคับคั่งของข้อมูลในระบบเครือข่ายจะถูกเรียกว่า คับคั่ง (Congested) เมื่อผู้ใช้ลงความเห็นว่าข้อมูลสามารถส่งได้น้อยลงในการเชื่อมต่อเครือข่ายของผู้ใช้คนนั้น” [14] ดังนั้นจะเห็นได้ว่าความคับคั่งที่เกิดในเครือข่ายขึ้นอยู่กับความเห็นของผู้ใช้งานระบบ อาจไม่สามารถจำกัดความได้ว่าอัตราการส่งข้อมูลระดับใดจึงจะถูกเรียกว่าเกิดความคับคั่ง นั่นคือในบางกรณีอัตราการส่งข้อมูลอาจลดลงได้อันเนื่องจากปัจจัยต่างๆในเครือข่าย แต่ข้อมูลยังสามารถที่จะส่งต่อไปได้ถึงแม้จะช้าก็ตาม หากผู้ใช้ยังพึงพอใจต่อการส่งข้อมูลนั้นอาจถือว่ายังไม่เกิดความคับคั่งของข้อมูลขึ้น แต่หากผู้ใช้ไม่สามารถที่จะส่งข้อมูลได้และอัตราการส่งข้อมูลตกลงดังแสดงในภาพประกอบ 8 นั่นหมายถึงเกิดความคับคั่งของข้อมูลขึ้นแล้ว



ภาพประกอบ 8 Network Congestion [15]

2.7.1 การควบคุมความคับคั่ง (Congestion Control)

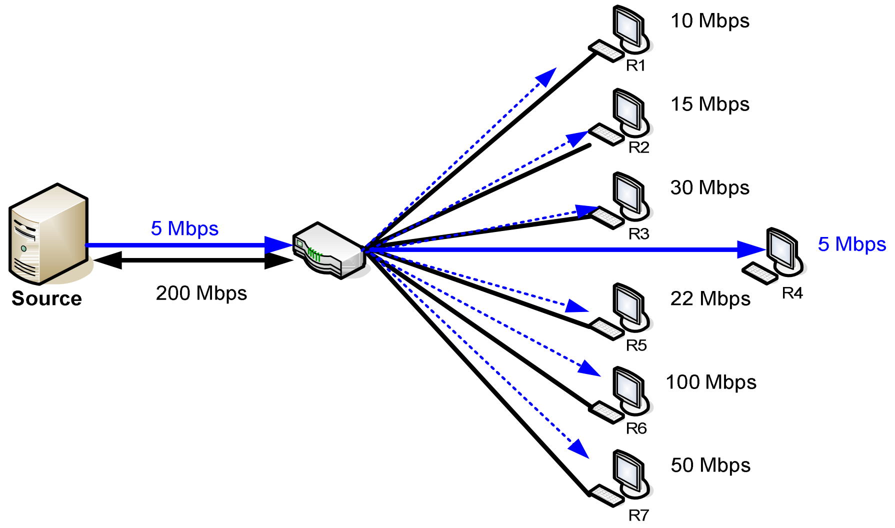
จากสถานการณ์ความคับคั่งที่ได้กล่าวมาแล้วข้างต้นจึงจำเป็นที่จะต้องมีกลไกที่ใช้ในการจัดสรรแบนด์วิดท์ของระบบเครือข่ายที่มีอยู่อย่างจำกัด แต่มีความต้องการที่ไม่จำกัดและเนื่องจากความต้องการข้อมูลประเภทมัลติมีเดียที่รวดเร็วโดยใช้เทคโนโลยีบอร์ดแบรนด์ ในการเข้าถึงข้อมูลก็ยิ่งทำให้มีโอกาสเกิดความคับคั่งที่รุนแรงขึ้นได้อีกซึ่งมีการเสนอแนวทางในการแก้ไขปัญหานี้ไว้ว่า

ความคับคั่งในระบบเครือข่าย นั้นไม่สามารถแก้ไขได้โดยการเพิ่มขนาดของแบนด์วิดท์ เพราะความคับคั่งในระบบนั้นไม่ใช่ปัญหาที่เกิดจากการขาดแคลนทรัพยากร (static resource shortage problem) แต่เป็นปัญหาของการจัดสรรทรัพยากรที่มีการเปลี่ยนแปลงอยู่ตลอดเวลา (dynamic resource allocation problem) ดังนั้นการเพิ่มแบนด์วิดท์ จึงไม่ใช่แนวทางในการแก้ไขปัญหาที่ถูกต้องแต่เป็นการหลีกเลี่ยงปัญหาที่เกิดขึ้นและยิ่งไปกว่านั้นยังทำให้ต้องเสียค่าใช้จ่ายที่สูงขึ้นในการเพิ่มประสิทธิภาพอุปกรณ์ต่างๆเพื่อทำให้เครือข่าย มีแบนด์วิดท์ที่มากขึ้น เพื่อรองรับกับความต้องการที่เกิดขึ้นเพียงแค่ 2-3 ชั่วโมงต่อวัน ดังนั้นกลไกการควบคุมความคับคั่งจึงเป็นวิธีที่ใช้ในการแก้ไขปัญหาความคับคั่งในระบบเครือข่ายที่น่าเชื่อถือมากที่สุด

2.8 การควบคุมความคับคั่งของข้อมูลในมัลติคาสท์ (Multicast Congestion Control)

ความคับคั่งของข้อมูลที่เกิดขึ้น เป็นอันตรายต่อการทำงานของระบบเครือข่ายอย่างมาก ไม่ว่าจะเป็นความคับคั่งของการส่งข้อมูลหรือการรับข้อมูลหรือที่เรียกว่า Feedback Implosion [2] คือการเกิดคอขวด (Bottleneck) ของการตอบกลับข้อมูล การควบคุมความคับคั่งของข้อมูลในการสื่อสารแบบมัลติคาสท์นั้นอาจแบ่งออกตามอัตราการส่งข้อมูลได้เป็น 2 กลุ่ม [2] คือ Single-Rate Multicast Congestion Control (SR-MCC) และ Multi-Rate Multicast Congestion control (MR-MCC)

2.8.1 Single-rate Multicast Congestion Control (SR-MCC) SR-MCC นั้นเป็นวิธีที่ผู้รับทั้งหมดจะรับข้อมูลเดียวกัน โดยวิธีการนี้จะเลือกผู้รับที่มีแบนด์วิดท์ น้อยที่สุดมาเป็นตัวแทน (Representative) จากนั้นผู้ส่งข้อมูลจะปรับอัตราการส่งข้อมูลตามแบนด์วิดท์ของตัวแทนเพื่อไม่ให้เกิดปัญหาความคับคั่ง ตัวอย่างโพรโทคอล SR-MCC ที่ได้เสนอไว้ได้แก่ Pseudofed [16], Multicast TCP (MTCP) [17], Pragmatic Multicast Congestion Control (PGMCC) [18] and TCP-friendly Multicast Congestion Control (TFMCC) [19] แต่วิธีการเหล่านี้นั้นมีจุดอ่อนที่ผู้รับทั้งหมดจะต้องรับข้อมูลที่อัตราเร็วต่ำที่สุดของตัวแทน หากมีผู้รับที่ช้าเป็นพิเศษในกลุ่มผู้รับข้อมูลก็จะทำให้อัตราการรับส่งข้อมูลของผู้รับอื่นๆที่มีแบนด์วิดท์มากกว่านั้นช้าไปด้วยโดยเฉพาะอย่างยิ่งอินเทอร์เน็ตเป็นระบบเครือข่ายขนาดใหญ่ที่มีความหลากหลาย ซึ่งกลุ่มผู้รับข้อมูลจากการมัลติคาสท์หนึ่งๆอาจจะกระจายอยู่ในสถานการณ์ทางระบบเครือข่ายที่แตกต่างกันมาก จึงเป็นการไม่ถูกต้องที่จะให้ผู้รับที่ช้าขัดขวางการรับส่งข้อมูลของผู้อื่นที่เร็วกว่า



ภาพประกอบ 9 แสดง SR-MCC

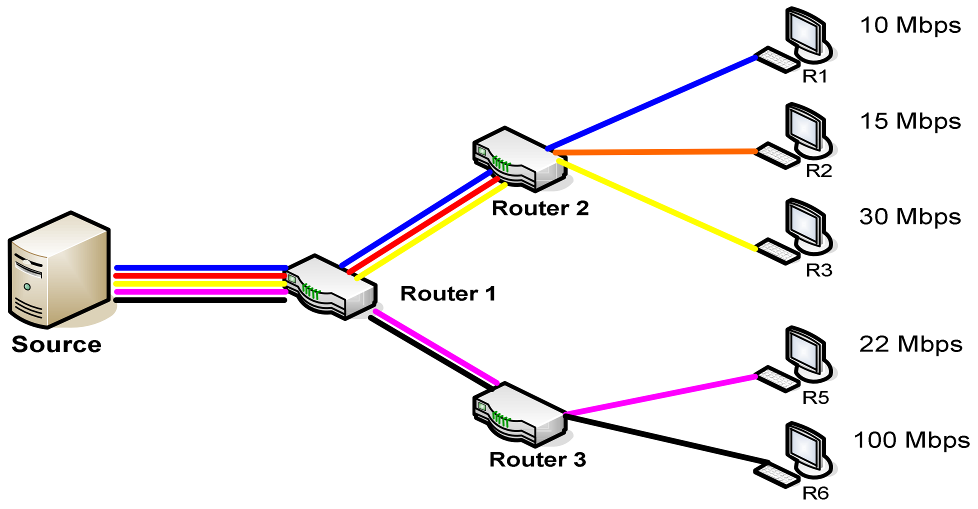
จากภาพประกอบ 9 แสดงให้เห็นว่า ถึงแม้เครื่องฝั่งส่งหรือฝั่งรับจะมีความสามารถในการรับข้อมูลที่ไม่เท่ากันก็ได้ แต่เมื่อมีการควบคุมความคับคั่งของข้อมูล โดยใช้เทคนิคแบบอัตราเดียว (Single-Rate) เครื่องทุกเครื่องใน Multicast Group จะส่งข้อมูลด้วยอัตราเดียวกันทั้งหมด ดังนั้นจากภาพประกอบ 9 จะใช้อัตราการส่งข้อมูลที่ 5 Mbps ทำให้เครื่องทั้งฝั่งส่งและฝั่งรับใช้ประโยชน์จากเครือข่ายได้อย่างไม่มีประสิทธิภาพ (Low utilization) ในกรณีเลวร้ายที่สุดอัตราการส่งข้อมูลอาจลดลงตามความสามารถของเครื่องที่มีอัตราต่ำสุดจนถึง 0 bps หรือไม่สามารถส่งข้อมูลได้เลย ปรากฏการณ์ที่อัตราการส่งข้อมูลลดลงจนเหลือศูนย์นี้เรียกว่า Drop-to-Zero Problem [2] ดังนั้น วิธีการควบคุมความคับคั่งของข้อมูลโดยอัตราเดียวนั้นจึงไม่เป็นที่นิยมในการนำมาใช้งาน

2.8.2 Multi-rate Multicast Congestion control (MR-MCC) ดังที่ได้กล่าวมาแล้วว่าการควบคุมความคับคั่งของข้อมูลโดยอาศัยหลักการลดอัตราการส่งให้น้อยลงจนเท่ากับอัตราการรับข้อมูลที่ต่ำที่สุด จึงเกิดปัญหาเรื่องการใช้เครือข่ายอย่างไม่มีประสิทธิภาพ ต่อมาได้มีการคิดวิธีการส่งข้อมูล โดยอาศัยหลักของการส่งข้อมูลด้วยอัตราส่งที่ไม่เท่ากันเรียกว่า Multi-Rate โดยอาศัยหลักการทำงานที่สำคัญ 2 อย่างคือ

2.8.2.1 Layer-Coding Transport (LCT) การเข้ารหัสข้อมูลเป็นชั้นก่อนส่ง เป็นการแบ่งข้อมูลออกเป็นชั้น (Layer) ก่อนส่งโดยใช้เทคนิคที่เรียกว่า Layer-Coding Transport (LCT) [14] โดยการสร้างข้อมูลของฝั่งส่งนั้นจะสร้างชุดข้อมูลที่มีลักษณะเหมือนกันแต่ทำการเข้ารหัสข้อมูลที่แตกต่างกัน ตัวอย่างการส่งข้อมูลสื่อประสม (Multimedia) เช่น Video Streaming ฝั่งส่งจะทำการสร้าง

ชั้นข้อมูลที่มีเนื้อหาเดียวกันแต่มีคุณภาพ (Quality) แตกต่างกัน เพื่อส่งไปยัง Multicast group หากฝั่งรับมีแบนด์วิดท์ในการรับข้อมูลเพียงพอตามคุณภาพของชุดข้อมูลที่ส่งมา ก็จะรับข้อมูลได้ในคุณภาพที่แตกต่างกันแต่เป็นข้อมูลที่เหมือนกันนั่นเอง

2.8.2.2 Receiver–driven การควบคุมโดยผู้รับ เป็นการกำหนดอัตราการส่งโดยพิจารณาจากฝั่งรับเป็นสำคัญโดยมีวิธีการคือ การที่จะกำหนดอัตราการส่งข้อมูลจะพิจารณาจากความสามารถในการรับข้อมูลของฝั่งรับเป็นสำคัญ หากฝั่งรับมีความสามารถในการรับข้อมูลมากฝั่งส่งจะส่งข้อมูลให้ในชั้นที่มีขนาดใหญ่กว่าฝั่งรับที่มีความสามารถในการรับต่ำ ผู้รับจะทำหน้าที่ตัดสินใจว่าจะรับข้อมูลกี่ชั้นเพื่อให้เกิดการรับข้อมูลที่เร็วที่สุดโดยไม่ก่อให้เกิดปัญหาความคับคั่งของระบบเครือข่ายโดยผู้รับที่มีความสามารถในการรับต่างกันอาจจะบอกรับชั้นข้อมูลในจำนวนที่ต่างกันซึ่งทำให้มีอัตราการรับที่แตกต่างไป



ภาพประกอบ 10 Multi-rate multicast congestion controls

โดยปกติแล้วขนาดของช่องสัญญาณจะมีการเปลี่ยนแปลงตลอดเวลาบางครั้งอาจมีขนาดกว้างมากหรือบางครั้งอาจไม่มีความสามารถในการรับข้อมูลได้เลย ดังนั้น การปรับเปลี่ยนขนาดของชั้นข้อมูลที่ส่งให้ฝั่งรับนั้นจะต้องมีการปรับเปลี่ยนตลอดเวลาตามความสามารถในการรับข้อมูลเป็นสำคัญ ซึ่งจากการที่ต้องมีการปรับเปลี่ยนอัตราอย่างต่อเนื่อง หากปรับเปลี่ยนได้อย่างเหมาะสมจะทำ

ให้ใช้งานระบบเครือข่ายได้อย่างมีประสิทธิภาพ แต่ต้องคำนึงถึง ความเร็วในการเปลี่ยนแปลงอัตรา (Responsive) ด้วยเนื่องจากหากการปรับเปลี่ยนไม่เร็วพอจะทำให้ข้อมูลเกิดการสูญหายได้ กลไกการตรวจจับหาความคับคั่งในเครือข่ายได้ 2 กลุ่มดังนี้

1. Join experiment [2] (การทดลองเข้าร่วม) คือเทคนิค การตรวจจับความคับคั่งและการบอกรับชั้นข้อมูล (Subscribe Layer) โดยฝั่งส่งข้อมูลจะทำการสร้าง Synchronous Point เพื่อตรวจสอบหาความกว้างของช่องสัญญาณที่เหลือพอที่จะรับส่งข้อมูลได้ หากฝั่งรับหากมีแบนด์วิดท์เหลือพอที่จะ Subscribe ชั้นที่สูงขึ้น ฝั่งส่งจะเพิ่มชั้นข้อมูลขึ้นไปเรื่อยๆ ทีละชั้นจนกระทั่งถึงจุดที่ฝั่งรับไม่มีแบนด์วิดท์พอที่จะรับข้อมูล ชั้นข้อมูลนั้นจะสูญหาย (เกิด Packet loss) ซึ่งฝั่งส่งจะควบคุมความคับคั่งโดยการลดจำนวนชั้นที่ Subscribe ลง ทำให้มีการใช้แบนด์วิดท์ของเครือข่ายได้อย่างมีประสิทธิภาพมากขึ้นโดยพยายามที่จะใช้แบนด์วิดท์ที่มีอยู่ให้มากที่สุด และยังสามารถควบคุมและปรับเปลี่ยนอัตราในการสื่อสารข้อมูลให้เพิ่มขึ้นหรือลดลงได้ตามสภาวะของความคับคั่งของเครือข่ายนั้นๆ ด้วย

2. Packet-pair Probe (PP) [20] (การประมาณค่าของแบนด์วิดท์ด้วยแพ็คเก็ตคู่) เป็นเทคนิคที่ได้รับถูกนำเสนอขึ้นหลัง Join- experiment มีแนวคิดต่างกัน ซึ่งจะทำการตรวจจับความคับคั่งในเครือข่ายโดยอาศัยการส่งแพ็คเก็ตข้อมูลทดสอบขึ้นหนึ่งคู่ เพื่อที่จะเอาไว้ใช้ในการหาช่องสัญญาณที่ว่าง โดยที่จะมีหนึ่งบิตใน Header ของแพ็คเก็ตเป็นตัวบอกว่าแพ็คเก็ตใดเป็นแพ็คเก็ตแรกของคู่แพ็คเก็ตนั้น ที่ฝั่งรับค่าประมาณการของแบนด์วิดท์ที่เหลืออยู่นั้นจะคำนวณได้จากขนาดของแพ็คเก็ตหารด้วยช่องว่างของเวลา (gap) ของการมาถึงของแพ็คเก็ต จากนั้นเมื่อแพ็คเก็ตแรกออกจากคิวนั่นหมายถึงเป็นสัญญาณบ่งบอกว่าได้เกินความคับคั่งของข้อมูลขึ้นแล้ว จากนั้นอัตราการส่งข้อมูลก็จะถูกปรับให้ลดลงโดยใช้ค่าจากการประมาณค่าข้างต้น ซึ่งการทำเช่นนี้จะทำครั้งเดียวของแต่ละช่วง เรียกว่า Check Period เพื่อป้องกันไม่ให้เกิดการแกว่งของสัญญาณที่มากเกินไปในส่วนของการปรับอัตราการส่งข้อมูลอาจแสดงได้โดยขั้นตอนวิธีดังนี้

กำหนดให้

PP_t คือ ขนาดของช่องสัญญาณที่ได้จากการประมาณค่าของ Packet-Pair

t_i คือ เวลา

B_n คือ ขนาดช่องสัญญาณปัจจุบันที่รองรับ Layer ได้ n Layer

B_e คือ ขนาดช่องสัญญาณโดยการประมาณ

L_i คือ Layer ข้อมูล

C คือ ค่าในการตรวจสอบช่องสัญญาณ

T_c คือ เวลาที่ได้จากการตรวจสอบ

if $PP_t < B_n$ then

T_c := t_i + C

until $B_n < PP_t$ do

drop layer n

n := n – 1

else if $PP_t >= B_n$ and $T_c < t_i$ then

$B_e := \min_{T_c - C < t_i <= PP_t}$

$T_c := t_i + C$

if $B_e >= B_n$ then while $B_{n+1} < B_e$ do

add layer $n+1$

$n := n+1$

หรืออธิบายอย่างง่ายคือ

If (estimated bandwidth < subscribed bandwidth) then

Drop layers until (subscribed bandwidth < estimated bandwidth)

If layer dropped is the lowest then

EXIT the session

End If

Else If (estimated bandwidth >= subscribed bandwidth)

Subscribe more layers

While (subscribed bandwidth < estimated bandwidth)

End If

แนวทาง MR-MCC นี้เป็นแนวทางที่เปิดโอกาสให้การควบคุมความคับคั่งของการสื่อสารแบบมัลติคาสท์บนอินเทอร์เน็ตนั้นมีความเป็นไปได้ และในช่วงไม่กี่ปีมานี้ได้มีการทำการศึกษาพัฒนาโพรโทคอลตามแนวทางนี้เป็นอย่างมากเพื่อพยายามที่จะตอบคำถามหลายๆ อย่างที่ยังรอคำตอบที่เหมาะสม เช่น จะทำการตรวจสอบรู้ว่าเกิดความคับคั่งขึ้นมาแล้วได้อย่างไร และจะทำการปรับอัตราการรับข้อมูลโดยการ Unsubscribe จากชั้นข้อมูลอย่างไรให้เหมาะสมและมีประสิทธิภาพ

จนถึงบัดนี้ ได้มีการนำเสนอโพรโทคอลที่พัฒนาตามแนวทางนี้อยู่หลายตัวเช่น Receiver-driven Layered Multicast (RLM) [21], Receiver-driven Layered congestion Control (RLC) [22], Fair Layer Increase Decrease with Dynamic Layering (FLID-DL) [23], Packet-pair receiver-driven cumulative Layered Multicast (PLM) [20], Explicit Rate Adjustment (ERA) [2] and MeERA [4]

แต่ยังไม่มีโพรโทคอลใดที่ออกแบบมาเพื่อรองรับกับการส่งผ่านข้อมูลของซอฟต์แวร์แบบเจาะจง เป็นเพียงการออกแบบเพื่อรองรับการส่งผ่านข้อมูลแบบกลางๆไม่ได้เจาะจงซอฟต์แวร์ อีกทั้งแต่ละโพรโทคอลยังไม่ได้กล่าวถึงประเด็นในการให้บริการมัลติมีเดียภายใต้กฎความเป็นมิตรต่อทีซีพีทำให้งานวิจัยนี้มีความสนใจในประเด็นปัญหาดังกล่าว

2.9 เกณฑ์การประเมินผลโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราส่ง

ในการประเมินผลถึงประสิทธิภาพของโพรโทคอล MR-MCC ที่ถูกพัฒนาและนำเสนอมานั้น มีเกณฑ์การพิจารณา ดังนี้ [2]

2.9.1 Responsiveness เป็นเกณฑ์ในการชี้วัดว่าโพรโทคอลมีความสามารถในการตอบสนองกับการเกิดความคับคั่ง (Congestion) ภายในเครือข่ายได้เร็วเพียงใด ซึ่งโพรโทคอลที่ดีจะต้องสามารถตรวจสอบหาความคับคั่งของเครือข่ายได้อย่างรวดเร็วเพื่อป้องกันการเกิดความล่มสลายของเครือข่าย (Congestion Collapse)

2.9.2 High Network Utilizations เป็นเกณฑ์ในการพิจารณาว่าโพรโทคอลตัวนั้นๆ สามารถตรวจสอบแบนด์วิดท์ของเครือข่ายเพื่อใช้งานได้อย่างคุ้มค่าเพียงไร โดยโพรโทคอลที่ดีจะต้องตรวจสอบแบนด์วิดท์และสามารถเข้าไปใช้ประโยชน์จากแบนด์วิดท์นั้นให้ได้มากและรวดเร็วที่สุด

2.9.3 Packet Loss เกณฑ์นี้เป็นเกณฑ์พิจารณาว่าโพรโทคอลตัวนั้นๆ มีความสามารถในการจัดการกับเรื่องของการสูญเสียของข้อมูล ซึ่งเกิดจากความคับคั่งขงเครือข่ายได้ดีเพียงใด โดยที่โพรโทคอลที่ดีจะต้องทำให้การสูญเสียข้อมูล (Packet loss) เกิดขึ้นให้น้อยที่สุดเท่าที่จะเป็นไปได้

2.9.4 Fairness เป็นเกณฑ์ในการพิจารณาถึงความสามารถของโพรโทคอลในการจัดสรรทรัพยากรที่ยุติธรรมต่อโพรโทคอลตัวอื่นๆได้ดีเพียงใด ทั้งนี้เนื่องจากโพรโทคอล MR-MCC เป็นโพรโทคอลที่สามารถใช้ประโยชน์จากแบนด์วิดท์ได้ดีกว่าตัวอื่น ดังนั้นการให้โพรโทคอลตัวอื่นสามารถทำงานร่วมกันได้จึงเป็นเรื่องที่ต้องให้ความสำคัญเป็นอย่างยิ่ง

2.9.5 Smoothness หมายถึง ความราบเรียบของสัญญาณข้อมูลขณะทำการส่งซึ่งการที่โพรโทคอล มีการตอบสนองที่รวดเร็วเกินไป (Responsiveness) ต่อสภาวะของระบบเครือข่ายในขณะนั้น ก็จะเป็นสาเหตุทำให้สัญญาณในการส่งเกิดการแกว่ง (สั่น) ทำให้ข้อมูลที่ผู้รับได้รับไม่มีความต่อเนื่อง ทำให้เกิดกระทบผลอย่างมากต่อการส่งข้อมูลที่เป็นสื่อผสม (Multimedia) ซึ่งเป็นการส่งทั้งภาพและเสียง เพราะภาพและเสียงที่ได้รับนั้นจะไม่ชัดเจนเนื่องจากสัญญาณที่ได้รับนั้นไม่ต่อเนื่อง ดังนั้น โพรโทคอล MR-MCC ที่ดีต้องไม่มีการแกว่ง (สั่น) ของสัญญาณในขณะที่ทำการรับ-ส่งมากเกินไป

2.9.6 Feasibility เป็นเกณฑ์ที่ใช้พิจารณาถึงความเป็นไปได้ในการใช้งานจริงของโพรโทคอลที่นำเสนอมา ทั้งนี้เนื่องจากโพรโทคอลที่นำเสนอมาในปัจจุบันยังไม่มีการใช้งานจริง ดังนั้นความเป็นไปได้ของการนำไปใช้งานได้จริงจึงเป็นเกณฑ์ที่สำคัญอย่างหนึ่งในการพิจารณาถึงประสิทธิภาพที่ดี

2.10 เทคนิคที่ใช้ในการทดสอบประสิทธิภาพของโพรโทคอล

ในกระบวนการที่ต้องการศึกษานั้นยังคงเป็นเพียงการสันนิษฐานโดยใช้โครงสร้างของตัวแบบเพื่อทำความเข้าใจในการศึกษาวิจัยที่นั้นเกี่ยวข้องกับ MR-MCC Protocol ซึ่งเป็น Network Protocol และในการประเมินค่าประสิทธิภาพของระบบก็มีวิธีที่เป็นไปได้อยู่ 3วิธี [24] คือ การวิเคราะห์จากตัวแบบทางคณิตศาสตร์ (Analytical Modeling), การทดลองในสถานการณ์จริง (Measurement) และ การจำลอง (Simulation)

Analytical Modeling เป็นกระบวนการที่เกี่ยวข้องกับ Mathematical Model เช่น Queuing Network, Petri-Nets เทคนิคนี้นั้นเป็นวิธีที่ใช้ค่าใช้จ่ายและเวลาน้อยมากเมื่อเทียบกับอีกสองเทคนิค ซึ่งผลลัพธ์จาก Analytical Modeling สามารถที่จะให้ค่าผลลัพธ์ที่เป็นตัวเลขได้ดีกว่า Measurement หรือ Simulation [25] ซึ่งแสดงให้เห็นว่าวิธีนี้เป็นเครื่องมือที่ดีที่จะนำมาใช้ในการศึกษาแบบเฉพาะทางได้ดี แต่ก็เป็นวิธีที่ไม่ดีถ้าหากจะนำไปศึกษาในด้านของกระบวนการทำงาน Keshav [26] แสดงให้เห็นข้อเสียของ Analytical Model ดังนี้ ต้องมีการแก้ไขเพื่อให้เข้าใจง่ายและการสันนิษฐานที่เกี่ยวข้องกับระบบเครือข่ายจริงไม่มีปฏิกิริยาตอบสนองในขณะทำการทดลอง Measurement เป็นวิธีที่ต้องอาศัยอุปกรณ์ การเขียนโปรแกรม และเวลาในการจำลองสถานการณ์ค่อนข้างมาก หัวใจสำคัญของวิธีนี้คือ การ Monitoring ถึงแม้ว่าเทคนิคนี้จะเป็นวิธีที่ให้ผลลัพธ์ที่ใกล้เคียงกับสิ่งที่เกิดขึ้นจริงแต่ก็ไม่ถูกนำมาใช้เนื่องจากมีค่าใช้จ่ายสูงยากในการที่จะปรับเปลี่ยน [27] อาจจะอันตรายเกินไปที่จะนำไปทดลองในระบบจริง Network Protocol ส่วนใหญ่นั้นจะมีความซับซ้อนเกินกว่าที่จะใช้ Analytical Modeling และจากข้อเสียของ Analytical Modeling และ Measurement ดังนั้นการศึกษาโดยใช้ Simulation จึงเป็นวิธีที่เหมาะสมมากที่สุดในการศึกษา MR-MCC Protocol นอกจากนั้น Network Simulator ยังทำให้การทำงานวิจัยร่วมกันนั้นง่ายขึ้นอีกด้วย Jain [24] ได้ทำการเปรียบเทียบความถูกต้องแม่นยำของเทคนิคทั้ง 3 ดังนี้ Analytical Modeling มีความแม่นยำน้อยที่สุดSimulation นั้นสามารถที่จะให้รายละเอียดได้มาก และ ต้องใช้การสันนิษฐานน้อยกว่า Analytical Modeling ดังนั้น Simulation จึงให้ค่าที่ใกล้ความจริงมาก Measurement นั้นมีการแปรผันค่อนข้างมากเนื่องจากสภาวะของระบบนั้นมีความแตกต่างกัน ทำให้ผลการทดลองที่ได้นั้นอาจจะผิดพลาดได้ โดยสามารถเปรียบเทียบ 3 เทคนิควิธีการข้างต้นได้ ดังตาราง 4

ตาราง 4 การเปรียบเทียบเทคนิคการทดสอบประสิทธิภาพ

Criteria	Analytical Modeling	Simulation	Measurement
Time Consumption	Low	Medium	High
Cost	Low	Medium	High
Accuracy	Low	Medium	Various

2.11 Network Simulation

ในการศึกษา Telecommunication networks นั้นได้ยอมรับการใช้งาน Computer Simulation [28] และ Network Simulation นั้นเป็นวิธีหลักที่ใช้ในการวิจัยเกี่ยวกับอินเทอร์เน็ตอย่างแพร่หลายวิธีหนึ่ง Floyd [29] กล่าวว่าบางงานวิจัยที่ได้ทำการเสนอมาแล้วนั้นอาจจะสามารถทำได้โดย Analytical Modeling หรือ test-beds ซึ่ง Network Simulation นั้นเป็นสิ่งที่ใช้ในการทำวิจัยภายใต้สภาวะของ Network ที่ยังไม่เกิดขึ้นจริงในอินเทอร์เน็ตแต่อาจจะเกิดขึ้นในอนาคต นอกจากนั้นการใช้ Network Simulation ยังแสดงข้อมูลที่ชัดเจนเพื่อใช้ในการวิจัยเกี่ยวกับ Network ได้อีกด้วย เช่น การพัฒนาโพรโทคอลที่กำลังใช้งานอยู่ในระบบปัจจุบัน การศึกษาผลกระทบที่เกิดขึ้นระหว่างโพรโทคอล และช่วยให้การเปรียบเทียบผลการทดลองของแต่ละงานวิจัยนั้นง่ายขึ้น [30] ซึ่ง Simulation นั้นใช้ลักษณะการทำงานของระบบอินเทอร์เน็ตในปัจจุบัน เพื่อหาผลกระทบที่อาจจะเกิดขึ้น ได้ในการทำงาน การวิจัยนี้ได้เลือกที่จะใช้ Network Simulation ในการทดสอบประสิทธิภาพของโพรโทคอล MR-MCC เมื่อให้บริการมัลติมีเดียภายใต้กฎความเป็นมิตรต่อทีซีพีของ IETF

2.12 งานวิจัยที่เกี่ยวข้อง

การควบคุมความคับคั่งแบบหลายอัตราส่งเป็นวิธีที่มีประสิทธิภาพและน่าจะเป็นวิธีที่สามารถนำไปพัฒนาให้สามารถทำงานได้จริงในอนาคต โดยเริ่มขึ้นตั้งแต่ปี คศ.1996 และพัฒนาต่อมาเรื่อยๆ ซึ่งมีโพรโทคอลที่น่าสนใจดังต่อนี้

2.12.1 Receiver-Driven Layered Multicast (RLM) ผลงานวิจัยของ Steven McCanne [21] เป็นโพรโทคอลควบคุมความคับคั่งของข้อมูลในการสื่อสารแบบมัลติคาสท์โพรโทคอลที่นำเสนอโดยใช้เทคนิคของ Receiver-Driven และ Layer-Coding Transport (LCT) ในการควบคุมการ

สื่อสารข้อมูล โดยการออกแบบและพัฒนาโพรโทคอล RLM ได้มุ่งเน้นพัฒนาขึ้นเพื่อการสื่อสารข้อมูลมัลติมีเดียเป็นสำคัญ

2.12.2 Receiver-Driven Layered Congestion Control (RLC) โพรโทคอล RLC ถูกนำเสนอโดยLorenzo Vicisano [22] เป็นโพรโทคอลที่มีการพัฒนาต่อจาก RLM เพื่อแก้ปัญหาต่างๆ ของRLM แต่ยังคงใช้หลักการพื้นฐานเดิมของ RLM ในการทำงานเพียงแต่ปรับปรุงในส่วนที่ใช้งานเกี่ยวกับการสื่อสารข้อมูลมัลติมีเดียให้มีมีประสิทธิภาพมากยิ่งขึ้น

2.12.3 Packet pair receiver-driven Layered Multicast (PLM) [20] ถูกนำเสนอโดย A. Legout ในปี ค.ศ.1999 เป็นการพัฒนาโดยใช้พื้นฐานจาก RLM ในส่วนของการแบ่งข้อมูลเป็นชั้น (Layer) โดยพยายามแก้ไขปัญหาของ RLM ในเรื่องของอัตราการสูญเสียข้อมูลในขณะปรับเปลี่ยนความสามารถในการส่งข้อมูล (Packet Loss Ratio) ซึ่ง PLM ไม่ได้ใช้เทคนิคการตรวจสอบหาขนาดช่องสัญญาณโดยการใช้ Join experiment เหมือนอย่างที่ RLM ใช้ แต่ PLM ใช้เทคนิคของ Receiver-Side Packet-Pair Probe (PP) และกำหนดว่าเราเตอร์ ใช้ระบบคิวแบบ Fair Queuing (FQ)

2.12.4 Explicit Rate Adjustment (ERA) ERA [2] ถูกนำเสนอโดย Somnuk Puangpronpitag เมื่อปี ค.ศ. 2003 ERA เป็นโพรโทคอลที่ยังคงใช้หลักการของ Packet-Pair Probe แต่เพิ่มจำนวนของแพ็คเก็ตที่ใช้สำหรับการตรวจสอบความคับคั่งให้แม่นยำมากขึ้นเรียกว่า Packet-Pair Bunch Probe และปรับปรุงการคำนวณค่าความเป็นมิตรของทีซีพีใหม่โดยใช้สมการจาก Jorge Widmer และ Mark Handley [19] เพื่อให้สามารถรองรับคุณสมบัติต่างๆ เหล่านี้ได้คือ ความเร็วในการตอบสนองที่จะทำการเปลี่ยนแปลงอัตราส่งข้อมูล (Responsive) ประสิทธิภาพของการใช้งานเครือข่าย (Utilization) การรวมข้อมูลที่รวดเร็ว (Fast Convergence) ขนาดที่สามารถขยายได้ (Scalability) อัตราการสูญหายของข้อมูล (Packet Loss Ratio) และความเป็นไปได้ที่จะนำไปใช้จริง

2.12.4.1 หลักการทำงานของ ERA

(1) โหนดฝั่งส่ง (Sender) โหนดฝั่งส่งจะทำหน้าที่แบ่งข้อมูลออกเป็นหลายๆชั้นข้อมูล โดยแต่ละชั้นข้อมูลจะถูกส่งเป็นคู่ (Packet Pair: PP) PP จะใช้ในการหาความกว้างของช่องสัญญาณในขณะปัจจุบันแต่ละแพ็คเก็ตข้อมูลจะประกอบด้วย Header ที่มีโครงสร้างดังนี้

ตาราง 5 ERA Packet Headers

ชื่อ	คำอธิบาย
OID	Object Identifier
LID	Layer Identifier
PSN	Packet Sequence Number
SCT	Sender Current Time
FPF	First PP Flag

Session Announcement Message (SAM) เป็นตัวอธิบายเซสชั่น (Session) ที่ประกอบไปด้วยอัตราการส่งของชั้นข้อมูล จำนวนของชั้นข้อมูล IP address ของผู้ส่ง หมายเลขที่อยู่และพอร์ตของแต่ละชั้นข้อมูล ขนาดของ Packet ความยาวของ Object และ Rate Adaptation Interval (RAI)

(2) โหนดฝั่งรับ (Receiver) ฝั่งรับจะรับ SAM มาแล้วทำการแปลข้อมูลก่อนที่จะ join session หลังจากนั้นฝั่งรับจะทำงานโดยกฎการควบคุมซึ่งจะสร้างแพ็คเก็ตเท่าที่จำเป็นเท่านั้น

2.12.4.2 ขั้นตอนวิธีการทำงานของ ERA

(1) ทุกๆกลุ่มของพ็คเก็ตข้อมูลที่มาถึง โหนดฝั่งรับจะประมาณค่าของความกว้างของช่องสัญญาณ (R'_{pp}) โดยใช้เทคนิคการประมาณค่าของ receiver-side Packet-pair bunches Probe ของ Paxson [31] โดยคำนวณได้จากสูตร

$$R'_{pp} = \frac{8M}{t_{gap}} \quad (1)$$

ซึ่ง M คือขนาดของแพ็คเก็ต (Byte) และ t_{gap} คือช่วงหากของการมาถึง (Internal arrival- time) เป็นวินาที ถ้าอัตราการส่งปัจจุบันมีค่ามากกว่า R'_{pp} หมายถึงช่องสัญญาณไม่พอ ให้ทำการลดชั้นข้อมูลลงทันทีเพื่อปรับอัตราส่ง

(2) ในทุกๆ RAI ฝั่งรับจะคำนวณค่าประมาณของช่องความสัญญาณที่คงเหลือ R'_{pp} ที่ซึ่งอย่างน้อยต้องเท่ากับ R'_{pp} ของ RAI สุดท้ายที่มี เมื่อกลุ่มของแพ็คเก็ตสูญหายเนื่องจากความคับคั่งของข้อมูลนั่นหมายถึง ช่องสัญญาณไม่พอ ดังนั้น จะต้องทำการกำหนดค่า R'_{pp} ใหม่ โดยเปลี่ยนจากค่าของ R'_{pp} ไปเป็น -1 เพื่อให้แสดงว่าเกิดความคับคั่งของข้อมูลขึ้นแล้ว

(3) จากนั้นฝั่งรับจะทำการคำนวณหาค่า PLR, RTT และ TCP-friendly Rate ใหม่

(4) ฝั่งรับจะคำนวณหาอัตราการรับในปัจจุบัน

(5) ฝั่งรับจะประมาณค่าของช่องความสัญญาณปลายทาง R_{Target} ได้จาก

If (l > 0) Then

If (R'_{pp} >= 0) Then

Set R_{Target} = Min(R_{TCP} , R'_{pp})

Else If (R'_{pp} = -1) Then

Set R_{Target} = R_{TCP}

End If

Else

Set R_{Target} = R_{TCP}

End If

ฝั่งรับจะขอเพิ่มหรือลดอัตราจากขั้นตอนวิธีนี้

If ($R_i > R_{Target}$) Then

Repeat Until ($R_i <= R_{Target}$)

If I > 0 Then

Unsubscribe from layer

$i = i - 1;$

Else

Exit the session

End If

Loop

Else If ($R_i < R_{Target}$) Then

Do While ($R_{i+1} < R_{Target}$)

Subscribe to an additional layer

$i = i + 1;$

Loop

Else If ($R_i = R_{Target}$)

Maintain the current Subscription level

End If

จากดัชนีวัดความสามารถของโพรโทคอลต่างๆ ที่จะพัฒนาให้เกิดขึ้นมาอยู่ ร่วมกับทีซีพีในอินเทอร์เน็ตปัจจุบันความยุติธรรมต่อทีซีพี เป็นเหตุผลหลักที่ต้องคำนึงถึงในโพรโทคอลของ ERA นั้นก็คำนึงถึงจุดนี้ โดยจะต้องมีการทำงานที่ไม่ทำให้ทีซีพีได้รับผลกระทบไปด้วย และจากสมการหาอัตราส่งที่มีความเป็นมิตรต่อทีซีพี (TCP Friendly Rate Estimation) ในยูนิคาสท์โดย Jamshid Mahdavi และ Sally Floyd [32] คำนวณได้จาก

$$R_{TCP} = \frac{8cM}{t_{RTT}\sqrt{l}} \qquad (2)$$

เมื่อ c คือค่าคงที่มีช่วงระหว่าง 0.87 – 1.31 ขึ้นอยู่ตามช่วงเวลา, M คือขนาดของ Packet (Byte), t_{RTT} ช่วงเวลา ไปกลับของข้อมูล (Round Trip Time), l คืออัตราสูญหายของข้อมูลหรือ PLR (ระหว่าง 0.0 – 1.0) ซึ่งค่าที่ได้จากการคำนวณนั้นเป็นค่าที่ สนับสนุนการทำงานในแบบยูนิคาสท์เท่านั้นและยังมีค่า c ที่เป็นค่าไม่แน่นอน ดังนั้นจึงได้มีการพัฒนาสมการใหม่โดย Jorge Widmer และ Mark Handley [19] เพื่อให้สนับสนุนการทำงานแบบมัลติคาสท์ได้เป็นสมการดังนี้

$$R_{TCP} = \frac{8M}{T_{RTT}\sqrt{\frac{2l}{3}} + 12\sqrt{\frac{3l}{8}}\,l(1+32l^2)} \qquad (3)$$

จะเห็นได้ว่าจากสมการจะมีตัวแปรของ T_{RTT} ซึ่งเป็นค่าตัวแปรที่หาค่าได้ลำบาก เนื่องการหา RTT นั้น ยังไม่มีเทคนิคที่เป็นมาตรฐานอาจทำได้ 3 วิธีคือ

1. การคำนวณ RTT จาก $RTT - request\ \ packet$ โดยทำการส่งแพ็คเก็ต ลงไปยังเครือข่ายก่อนส่งจะทำการเก็บเวลาส่งไว้ที่ฝั่งส่ง จากนั้นให้ส่งแพ็คเก็ตไปจนถึงฝั่งรับ และฝั่งรับทำการตอบกลับแพ็คเก็ตทันที เมื่อแพ็คเก็ตกลับมาถึงฝั่งส่งจะได้เวลากลับถึงมาลบออกด้วยเวลาส่งจะทำให้ได้ RTT ในวิธีนี้ทำให้ได้ค่าของ RTT ที่ใกล้เคียงความจริงแต่เมื่อการส่งแพ็คเก็ตลงไปเพื่อหา RTT เท่านั้นเป็นการใช้งานเครือข่ายอย่างไม่มีประสิทธิภาพนัก อีกทั้งยังทำให้เกิดเหตุการณ์ที่เรียกว่า Feedback Implosion หรือการตอบกลับข้อมูลอย่างมหาศาลทำให้ เกิดผลเสียกับระบบได้

2. การคำนวณ RTT โดยหลักการส่งทางเดียว จากข้างต้นปัญหา Feedback Implosion อาจทำให้เกิดปัญหากับเครือข่ายดังนั้นหากทำการส่งเพียงครั้งเดียวไม่ต้องรอการตอบกลับของข้อมูล คือ ฝั่งส่ง ส่งแพ็คเก็ตข้อมูลไปยังฝั่งรับแล้วฝั่งรับจะประมาณค่า RTT ได้จากระยะเวลาตั้งแต่ส่งจนไปถึง จะถือเป็นครึ่งหนึ่งของ RTT ก็จะคำนวณหา RTT ได้ แต่ในความเป็นจริงแล้วนั้น

การส่งข้อมูลในเครือข่ายนั้น เส้นทางไปและกลับอาจมีปัญหาหรือแบนด์วิดท์ที่ต่างๆกันไป ดังนั้นการอุปมาว่าเวลาเป็นครึ่งหนึ่งนั้น จึงได้ค่าที่คลาดเคลื่อนจากความเป็นจริง

3. การคำนวณ RTT โดยใช้ Layered Multicast เมื่อการคำนวณ 2 วิธี ข้างต้นให้ผลที่ไม่มีประสิทธิภาพเท่าที่ควรดังนั้น Michael Luby [33] จึงได้เสนอแนวคิดที่ว่าการส่งข้อมูลแบบมัลติคาสท์นั้นต้องมีการเสียเวลาในการ Join เข้าสู่ session ดังนั้น การหาค่า RTT จึงต้องมีการนับรวมค่าความเสียเวลาต่างๆเข้าไปด้วย ดังได้เป็นสมการ

$$t_{RTT} = (2 * onewaylatency) + \varepsilon \quad (3)$$

จากสมการ ค่า ε เป็นค่าที่ตั้งขึ้นโดยการประมาณตามความเหมาะสม

2.12.5 Multicast-encouraging ERA (MeERA) [4] จากปัญหาความเป็นมิตรต่อทีซีพีของมัลติคาสท์ที่ถูกบังคับให้มัลติคาสท์นั้นจะต้องไม่ใช้แบนด์วิดท์มากกว่าทีซีพีไม่ว่ากรณีใดๆทำให้เกิดข้อโต้แย้งว่าถ้าในกรณีที่มีผู้รับเท่าๆกันนั้น การที่มัลติคาสท์จะไม่ใช้แบนด์วิดท์มากกว่าทีซีพีนั้นดูจะสมเหตุ
สมผล แต่ถ้าเมื่อพิจารณาจากจำนวนผู้รับของมัลติคาสท์ที่มากกว่าทีซีพีแล้ว ถ้ามัลติคาสท์จะต้องใช้แบนด์วิดท์น้อยกว่าหรือเท่ากับทีซีพีนั้นดูจะไม่สมเหตุสมผลนักเพราะมัลติคาสท์ต้องให้บริการผู้รับในจำนวนที่มากกว่าทีซีพีนั่นเอง

ตาราง 6 แสดงสัดส่วนการใช้แบนด์วิดท์ ระหว่าง MeERA กับ TCP [4]

Receiver	**Multicast**		**TCP**	
	Kbps	**%**	**Kbps**	**%**
1	200.00	50.00%	200.00	50.00%
2	251.47	62.87%	148.53	37.13%
3	270.91	67.73%	129.09	32.27%
4	281.88	70.47%	118.12	29.53%
5	289.18	72.29%	110.82	27.71%
10	307.03	76.76%	92.97	23.24%
100	339.44	84.86%	60.56	15.14%
1000	355.10	88.77%	44.90	11.23%

จากกฎเรื่องความเป็นมิตรต่อทีซีพี (TCP-friendliness) ที่เคร่งครัดนี้ถือประเด็นปัญหาที่เป็นอุปสรรคสำคัญในการขายตัวของมัลติคาสท์บนอินเทอร์เน็ต ทำให้ได้มีการนำเสนอการออกแบบ MR-MCC ใหม่โดยการปรับปรุง MR-MCC เดิมได้ถูกนำเสนอมาก่อนหน้านี้ที่มีชื่อว่า ERA โดย S. Puangpronpitag ในการออกแบบใหม่ของ MeERA นี้ได้มีการเพิ่มประสิทธิภาพในการเป็นมิตรกับทีซีพีอย่างสมเหตุสมผล [34] โดยได้มีการเพิ่มเทคนิคใหม่เข้าไปคือ Logarithmic receiver dependent bandwidth allocation scheme ซึ่งถูกใช้ที่เราท์เตอร์โดยจะให้แบนด์วิดท์แก่ MeERA มากกว่าทีซีพีในกรณีที่ MeERA มีผู้รับมากกว่าทีซีพีโดยกลไกนี้ไม่ได้ทำให้ทีซีพีขาดแคลนแบนด์วิดท์มากนัก และผู้รับของ MeERA จะปรับอัตราการรับของตัวเองตามสถานการณ์ของเครือข่าย โดยการใช้ Packet-pair Probe (PP)

จากการแก้ปัญหาการจัดสรรแบนด์วิดท์ที่ไม่ยุติธรรมของ MeERA ทำให้ในบางสถานการณ์ที่ MeERA ให้บริการกลุ่มผู้ใช้มากกว่าทีซีพีจะได้รับการจัดสรรแบนด์วิดท์ที่มากกว่าทีซีพีซึ่งนับได้ว่ามีความสมเหตุสมผล แต่ในงานวิจัยนี้จะมุ่งเน้นไปที่การเผยให้เห็นถึงปัญหาของกฎความเป็นมิตรต่อทีซีพีของ IETF ว่าอาจจะไม่ยุติธรรมกับโพรโทคอล MR-MCC เมื่อต้องให้บริการมัลติมีเดีย

2.12.6 MR-MCC กับความเป็นมิตรต่อทีซีพี (TCP-friendliness) จากการศึกษา [35] ทำให้พบว่าในปัจจุบันโพรโทคอลที่ทำงานอยู่บนอินเทอร์เน็ตกว่าร้อยละ 90 คือทีซีพี ดังนั้นการพัฒนา MR-MCC โพรโทคอลขึ้นมาใช้งานบนอินเทอร์เน็ตจึงมีความจำเป็นที่จะต้องคำนึงถึงเรื่องการแบ่งปันการใช้งานแบนด์วิดท์ของเครือข่ายกับทีซีพีหรือที่เรียกว่าความเป็นมิตรต่อทีซีพี (TCP-friendliness หรือ TCP-compatibility) เป็นปัจจัยสำคัญปัจจัยหนึ่งเพื่อไม่ให้โพรโทคอล MR-MCC ที่พัฒนาขึ้นมานั้นแย่งแบนด์วิดท์ของทีซีพีไปมากจนทีซีพีทำงานต่อไม่ได้ ซึ่งจะทำให้เกิดปัญหาขึ้นในเครือข่าย ทำให้กลายเป็นข้อเสนอแนะของ IETF ที่กำหนดขึ้นมาสำหรับโพรโทคอลที่จะพัฒนาขึ้นมาเพื่อใช้งานบนอินเทอร์เน็ตว่าจะต้องมีคุณสมบัติที่มีความเป็นมิตรต่อทีซีพี พชระ พฤกษะศรี [36] ได้ทำการศึกษาในเรื่องของความเป็นมิตรต่อทีซีพีของโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราส่งโดยได้ทำการทดสอบโพรโทคอล PLM กับ TCP และERA กับ TCP ซึ่งจากผลการทดสอบได้ดังนี้

2.12.6.1 PLM กับ TCP ภายใต้สถานการณ์ที่ทำงานบนเราเตอร์แบบ FQ และ FIFO Drop-tail ผลทดลองที่ได้นั้นปรากฏว่าโพรโทคอล PLM จะมีความเป็นมิตรในการทำงานกับทีซีพีก็ต่อเมื่อเราเตอรนั้น สนับสนุนการทำงานแบบ FQ เท่านั้น ซึ่งเมื่อทำการทดสอบโดยกำหนดสถานการณ์การทำงานที่PLM ทำงานบนเราเตอร์ที่ทำงานแบบ FQ ปรากฏว่า PLM มีความเป็นมิตรต่อทีซีพีแต่เมื่อเปลี่ยนกากำหนดสถานการณ์ให้ PLM ทำงานบนเราเตอร์ที่ทำงานแบบ FIFO Drop-tail ผลที่ได้ปรากฏว่า PLM ไมสามารถรักษาความเป็นมิตรกับทีซีพีได้เลย ทำให้สรุปได้ว่า PLM นั้นมี

ความเป็นไปได้น้อยมากที่จะนขึ้นมาใช้งานบนอินเทอร์เน็ตเนื่องจากเราเตอร์ในปัจจุบันทำงานแบบ FIFO Drop-tail เกือบทั้งหมด ดังนั้นจึงเป็นการยากที่จะเปลี่ยนให้เราเตอร์มาทำงานแบบ FQ

2.12.6.2 ERA กับ TCP ภายใต้สถานการณ์ที่ทำงานบนเราเตอร์แบบ FIFO Drop-tail ผลทดลองที่ได้นั้นในสถานการณ์ที่ ERA และ TCP เริ่มทำงานพร้อมๆ กันนั้น ERA สามารถที่จะรักษาสภาพความเป็นมิตรต่อทีซีพีได้เป็นอย่างดีภายใต้เราเตอร์ที่ทำงานแบบ FIFO Drop-tail แต่เมื่อเปลี่ยนเงื่อนไขการจำลองสถานการณ์เป็น ให้ทีซีพีเริ่มทำงานอยู่ในเครือข่ายก่อน ERA ผลที่ได้ปรากฏว่า ERA นั้นไม่สามารถเริ่มทำงานได้เลย เนื่องจากทีซีพีมีการประมาณค่าแบนด์วิดท์ในระยะเวลา 1 round trip time แต่ ERA ใช้เวลาประมาณค่าเพียงครึ่งหนึ่งโดยเทคนิค PP ดังนั้น ในขณะที่ทีซีพีใช้แบนด์วิดท์เกือบทั้งหมดในเครือข่ายแล้ว ERA ที่เริ่มทำงานหลังทีซีพีที่มีอยู่จำนวนมากจะพยายามแทรกเข้าไปใช้แบนด์วิดท์ด้วยความเร็วที่ ERA ประมาณค่าได้เร็วกว่าและรู้ว่าแบนด์วิดท์เหลือน้อยมาก จึงปล่อยให้ทีซีพีนำไปใช้ก่อน แต่ทีซีพีเองซึ่งประมาณค่าช้ากว่าไม่รู้ว่า ERA พยายามจะขอแบ่งแบนด์วิดท์ จึงทำงานต่อไปเรื่อยๆเสมือนว่าไม่มีโพรโทคอลอื่นมาขอแบ่งแบนด์วิดท์นั่นเอง

จากงานวิจัยดังกล่าวข้างต้นทำให้เห็นว่า MR-MCC โพรโทคอลที่พัฒนาขึ้นมาในปัจจุบันยังมีปัญหาในเรื่องของความเป็นมิตรต่อทีซีพีอยู่ ไม่ว่าจะเป็น PLM หรือ ERA ซึ่งในส่วนของ PLM นั้นมีจุดอ่อนที่สำคัญอย่างยิ่งคือต้องทำงานกับเราเตอร์ที่ทำงานแบบ FQ เท่านั้น ซึ่งเป็นปัญหาใหญ่เนื่องจาก เราเตอร์ที่ทำงานในปัจจุบันนั้นเป็นการทำงานแบบ FIFO Drop-tail เกือบทั้งหมด ดังนั้นการที่จะเปลี่ยนให้เราเตอร์ที่เชื่อมต่ออินเทอร์เน็ตทั้งหมดให้ทำงานแบบ FQ จึงแทบจะเป็นไปไม่ได้ ส่วน ERA นั้นมีจุดอ่อนในส่วนของการทำงานถ้าทำงานหลังจากทีซีพี โพรโทคอล ERA จะไม่สามารถทำงานหรือรักษาสภาพความเป็นมิตรต่อทีซีพีไว้ได้เลย ซึ่งแนวทางในการแก้ปัญหาของ ERA คือการปรับปรุงอัลกอริธึมในการทำงานของ ERA ให้สามารถทำงานได้ในสถานการณ์ที่มีทีซีพีทำงานอยู่ก่อนหน้าในเครือข่ายให้ได้นั่นเอง

2.12.7 MR-MCC กับ Multimedia over Best-Effort Service จากการที่ทีซีพีทำงานอยู่บนอินเทอร์เน็ตมากกว่าร้อยละ 90 แต่ในทางตรงกันข้ามอินเทอร์เน็ตกลับถูกใช้งานเพื่อส่งผ่านข้อมูลแบบกลุ่มมากขึ้น โดยเฉพาะอย่างยิ่งการใช้งานซอฟต์แวร์มัลติมีเดีย ซึ่งตัวทีซีพีเองไม่สนับสนุนการทำงานดังกล่าว โดยที่ในปัจจุบันวิธีการที่ใช้แก้ปัญหาคือ การพัฒนาซอฟต์แวร์ให้รองรับการทำงานมัลติมีเดีย (เช่น Streaming , Buffer) แต่ทั้งนี้ทั้งนั้นปัญหาสำคัญคือเรื่องของความแตกต่างในเรื่องของการควบคุมความคับคั่ง โดยที่ซอฟต์แวร์มัลติมีเดียนั้นมีกลวิธีการควบคุมความคับคั่งที่ตอบสนองต่อสถานการณ์ช้ากว่า แต่แกว่ง (Oscillatory) ได้มากกว่าทีซีพี ทั้งนี้เพื่อทำให้การส่งข้อมูลมีความราบเรียบ (Smoother) เพื่อคุณภาพในการใช้งานมากกว่าทีซีพี ดังนั้นโพรโทคอลที่ทำงานรองรับกับมัลติมีเดียน่าจะสามารถได้รับการแบ่งปันแบนด์วิดท์ของเครือข่ายเท่าๆกับทีซีพี นอกจากนี้ซอฟต์แวร์มัลติมีเดียบางประเภทยังต้องการสิ่งที่เรียกว่า QoS (Quality of Service) คือ การรับประกันการบริการ

(เช่น การที่ต้องมีการการันตีว่าจะต้องมีแบนด์วิดท์ขั้นต่ำเท่าใดจึงจะสามารถส่งข้อมูลได้) เพื่อให้ผู้ใช้งาน หรือผู้รับไฟล์ได้มาซึ่งคุณภาพ และความพึงพอใจมากที่สุด ซึ่งอาจจะไม่จำเป็นต้องมีความเป็นมิตรกับทีซีพีมากจนเกินไปก็ได้

จากการศึกษาของ Jon Crowcroft [37] พบว่า ความเป็นมิตรต่อทีซีพีไม่ได้สนับสนุนการทำงานของซอฟต์แวร์บางประเภทเท่าใดนักโดยเฉพาะอย่างยิ่งซอฟต์แวร์มัลติมีเดีย ทั้งนี้ปัญหาที่พบคือ ในเรื่องของ TCP throughput equation กล่าวคือ TCP throughput มีขนาดลดลงมากเมื่อเกิดเงื่อนไขของการสูญเสียแพ็คเก็ตข้อมูล ดังนั้น TCP จะไม่สนับสนุนลักษณะการทำงานของซอฟต์แวร์มัลติมีเดีย และจากการทดลองในงานวิจัยดังกล่าวพบว่าการใช้งานการควบคุมความคับคั่งที่เป็นมิตรกับทีซีพีนั้นให้ประสิทธิภาพการใช้งานที่แย่กับซอฟต์แวร์ที่มีคุณสมบัติแบบ loss-tolerant ซึ่งจากการที่ทีซีพีทำงานอยู่บนอินเทอร์เน็ตทำให้การพัฒนาโพรโทคอล MR-MCC ที่มีขึ้นเพื่อใช้งานบนอินเทอร์เน็ตอาจต้องเปลี่ยนแนวความคิดในเรื่องของความเป็นมิตรต่อทีซีพีใหม่ก็เป็นไปได้ ดังที่ได้มีการนำเสนอแนวคิดไว้ใน [38] ที่มองว่า1 ช่องสัญญาณ (1 flow) มัลติคาสท์ต้องรับภาระโหนดปลายทางถึง n โหนด ในขณะที่ยูนิคาสท์อย่างทีซีพีรับภาระเพียง 1 โหนดเท่านั้น “เปรียบเสมือนคนสองคน ที่คนหนึ่งมีร่างกายใหญ่กว่าทำงานมากกว่าส่วนอีกคนร่างกายเล็กกว่าทำงานน้อยกว่าแต่ได้รับอาหารเท่ากัน” ซึ่งไม่เกิดความยุติธรรมขึ้นอย่างแท้จริงส่งผลให้ผู้ใช้งานกลับไปใช้ยูนิคาสท์แทนเพราะได้แบนด์วิดท์ที่สูงกว่าทำให้คุณภาพของข้อมูลที่ได้ดีตามไปด้วย จากเหตุผลนี้การส่งผ่านข้อมูลให้ผู้รับจำนวนเท่าๆแบบยูนิคาสท์ เช่น ทีซีพี ย่อมจะทำให้เกิดการใช้งานแบนด์วิดท์มากกว่าการส่งผ่านข้อมูลโดยใช้ MR-MCC ใช่หรือไม่? ดังนั้นความเป็นมิตรต่อทีซีพีจะหมายถึงเพียงแค่การใช้งานแบนด์วิดท์ของ MR-MCC ที่เท่ากับหรือน้อยกว่า ทีซีพีเท่านั้นเองหรือ? ถ้าหากทีซีพีใช้งานแบนด์วิดท์ไม่หมดและเหลือแบนด์วิดท์ที่ไม่ใช้งานอยู่แล้วเหตุใด MR-MCC จึงไม่สามารถนำแบนด์วิดท์ที่เหลืออยู่ตรงนั้นมาใช้งานได้ โดยเฉพาะอย่างยิ่งซอฟต์แวร์มัลติมีเดียที่ต้องการการันตีคุณภาพในการให้บริการ (QoS) ดังนั้น งานวิจัยนี้จึงสนใจในเรื่องของการทำงานของซอฟต์แวร์มัลติมีเดียกับโพรโทคอล MR-MCCภายใต้กฎความเป็นมิตรต่อทีซีพี

2.13 ประเด็นปัญหาการวิจัย

จากที่กล่าวมาข้างต้นนั้นทำให้สามารถสรุปได้ว่าโพรโทคอลที่ทำงานอยู่บนอินเทอร์เน็ตในปัจจุบันไม่มีโพรโทคอลใดเลยที่มีคุณสมบัติที่สนับสนุนการส่งผ่านข้อมูลแบบมัลติมีเดียอย่างเจาะจง และในการส่งผ่านข้อมูลแบบมัลติมีเดียโดยทั่วไปจะเป็นการส่งผ่านข้อมูลแบบกลุ่ม (เช่น การกระจายเสียงวิทยุผ่านอินเทอร์เน็ต, การเผยแพร่วีดีโอผ่านอินเทอร์เน็ต) ซึ่งโพรโทคอลที่ทำงานอยู่ในอินเทอร์เน็ตในปัจจุบันเป็นโพรโทคอลที่มีวิธีการส่งผ่านข้อมูลแบบยูนิคาสท์ จึงเกิดปัญหาในเรื่อง

ของความคุ้มค่าในการใช้งานเครือข่าย เนื่องจากต้องมีการส่งข้อมูลเดิมซ้ำๆกันอยู่ตลอดเวลา โพรโทคอล MR-MCC ซึ่งมีลักษณะการทำงานในการส่งผ่านข้อมูลแบบหนึ่งต่อกลุ่ม จึงน่าจะเป็นวิธีการที่เหมาะสมกว่าในการส่งผ่านหรือกระจายข้อมูลมัลติมีเดีย แต่ปัญหาของโพรโทคอล MR-MCC ในปัจจุบันที่ยังต้องทำการศึกษาอยู่ คือ

2.13.1 Loss tolerant / Delay sensitive vs. Delay tolerant / Loss sensitive คุณสมบัติที่สำคัญในการส่งข้อมูลมัลติมีเดียนั้นก็คือ ในการส่งจะต้องเกิดความล่าช้า (Delay) ระหว่างผู้ส่งกับผู้รับน้อยที่สุด และยอมให้เกิดความสูญหายของข้อมูลบางส่วนเพื่อให้ข้อมูลส่วนใหญ่ยังสามารถทางานได้ คุณสมบัติที่สำคัญนี้เรียกว่า Loss tolerant/Delay sensitive ซึ่งเป็นคุณสมบัติที่ตรงกันข้ามกับการส่งข้อมูลของทีซีพีที่ยอมให้เกิดความล่าช้า แต่ไม่ยอมให้เกิดการสูญเสียข้อมูล (Delay tolerant/ Loss sensitive) ดังนั้น การที่มัลติคาสท์พยายามทำตัวให้เหมือนกับทีซีพี เมื่อให้บริการมัลติมีเดียอาจทำให้เกิดปัญหาความล่าช้าจนล้มเหลวในการส่งข้อมูลที่มีคุณภาพ ซึ่งงานวิจัยนี้จะทำการทดลองเพื่อทดสอบประเด็นปัญหานี้

2.13.2 Responsiveness vs. Smoothness ความรวดเร็วในการตอบสนองต่อความคับคั่งเป็นคุณสมบัติสำคัญของโพรโทคอล MR-MCC แต่โพรโทคอล MR-MCC ที่ให้บริการมัลติมีเดียนั้นไม่สามารถที่จะตอบสนองต่อความคับคั่งอย่างรวดเร็วได้ เพราะต้องคำนึงถึงความราบเรียบ (Smoothness) ของการส่ง ซึ่งจะส่งผลต่อคุณภาพที่ผู้รับได้รับด้วย จากประเด็นนี้จึงสันนิษฐานได้ว่าโพรโทคอล MR-MCC แบบเดิมที่ได้มีการเสนอมานั้น อาจมีปัญหาในประเด็นของความราบเรียบในการส่ง เพราะโพรโทคอลเหล่านั้นต่างถูกออกแบบให้ตอบสนองต่อความคับคั่งอย่างรวดเร็ว ซึ่งแน่นอนว่าส่งผลกระทบต่อความราบเรียบด้วย ซึ่งความราบเรียบนี้นับว่ามีความสำคัญต่อการส่งข้อมูลมัลติมีเดียเป็นอย่างมาก งานวิจัยนี้จะทำการทดลองให้เห็นว่าความราบเรียบระหว่างการส่งข้อมูลนั้นส่งผลกระทบอย่างไรกับมัลติมีเดีย

2.13.3 Bandwidth Consumption ในสถานการณ์บางสถานการณ์ของมัลติเดียนั้น การเข้าใช้แบนด์วิดท์ที่มีอยู่อย่างรวดเร็วอาจไม่เกิดประโยชน์ต่อคุณภาพของข้อมูลที่ผู้รับได้รับ แต่อาจก่อให้เกิดปัญหาเรื่องความราบเรียบของข้อมูลได้ งานวิจัยนี้จะทำการทดลองให้เห็นว่า การเพิกเฉยต่อแบนด์วิดท์ที่มีอยู่อาจจะส่งผลดีมากกว่าการเข้าใช้แบนด์วิดท์อย่างรวดเร็ว

2.13.4 Layer Granularity ในโพรโทคอล MR-MCC ทั่วๆไปนั้น การกำหนดขนาดของชั้นข้อมูล (Layer) และจำนวนชั้นที่ใช้นั้น ไม่ได้ถูกกำหนดอย่างชัดเจน จากการศึกษางานวิจัยที่เกี่ยวข้องพบว่าล้วนเป็นการกำหนดคร่าวๆ โดยผู้เสนอโพรโทคอลทั้งสิ้น แต่ประเด็นของชั้นของการสื่อสารนั้นมีผลอย่างยิ่งต่อมัลติมีเดีย เพราะถ้าการกำหนดค่าที่เกี่ยวกับชั้นของการสื่อสารไม่เหมาะสมกับมาตรฐานการเข้ารหัส (CODEC) ข้อมูลของมัลติมีเดียแล้ว ก็อาจส่งผลต่อการทำงานของมัลติมีเดีย

นั้นล้มเหลวได้ งานวิจัยนี้จะทำการทดลองให้เห็นว่าการจัดการชั้นข้อมูลของมัลติคาสท์นั้นมีความสัมพันธ์กับมาตรฐานการเข้ารหัสข้อมูลของมัลติมีเดียอย่างไร

บทที่ 3

วิธีดำเนินการวิจัย

ในการดำเนินการวิจัยของงานวิจัยในบทนี้ เป็นการนำเสนอเกี่ยวกับการทดสอบประสิทธิภาพของโพรโทคอล MR-MCC เพื่อใช้เป็นแนวทางสำหรับปรับปรุงโพรโทคอลควบคุมความคับคั่ง ได้แก่ วิธีการ ขั้นตอน เครื่องมือที่ใช้ทดลอง การกำหนดองค์ประกอบของการทดลอง รวมถึงผลการทดลองในมุมมองต่างๆ เพื่อแสดงให้เห็นผลการทดลองซึ่งจะไปสู่การเพิ่มประสิทธิภาพโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราส่งในเครือข่ายมัลติคาสท์ในอนาคต ซึ่งการดำเนินการวิจัยที่ถูกต้องนั้นจะต้องทำในลักษณะต่อยอดของความรู้ คือในเบื้องต้นต้องทำการศึกษาระบบหรืองานวิจัยที่ได้มีการพัฒนามาก่อน เพื่อเป็นแนวทางและเป็นข้อมูลให้สามารถวิเคราะห์หาข้อดีและข้อเสียของงานวิจัยนั้นๆ ขั้นตอนการดำเนินการวิจัยได้มีขั้นตอน ดังนี้

3.1 เครื่องมือการจำลองระบบเครือข่าย

ในงานวิจัยที่เกี่ยวข้องกับการสื่อสารโทรคมนาคม ให้การยอมรับการใช้ตัวจำลองสำหรับจำลองการทำงานของระบบเครือข่าย (Network Simulator) และใช้เป็นเครื่องมือในการทดสอบหาผลลัพธ์ของโพรโทคอลที่ออกแบบและพัฒนา ซึ่งในปัจจุบันอาจถือเป็นวิธีการหลักที่ใช้ในงานวิจัยที่เกี่ยวข้องกับอินเทอร์เน็ต ดังจะเห็นได้จากงานวิจัยทางวิชาการส่วนมากเลือกใช้วิธีการนี้เพื่อทดสอบผลของงานวิจัย เช่น IEEE, ACM หรือ SIGCOMM เป็นต้น การวิจัยบางงานอาจใช้การวิเคราะห์ทางคณิตศาสตร์เพื่อนำไปสู่การพัฒนาระบบงานไปใช้จริงบนอินเทอร์เน็ต แต่บางงานวิจัยต้องอาศัยสถานการณ์จริงในการวิเคราะห์ แต่ด้วยข้อจำกัดในเรื่องงบประมาณและตัวแปรต่างๆในสถานการณ์จริงที่ไม่อาจจะควบคุมได้ ทำให้การใช้ตัวจำลองได้รับความนิยมมากขึ้น เพราะสามารถควบคุมค่าตัวแปรต่างๆของระบบที่คาดว่าจะเกิดขึ้นได้ ตั้งแต่อดีตจนถึงปัจจุบันมีการสร้างเครื่องมือจำลองการทำงานของเครือข่ายออกมามากมาย โดยมีจุดประสงค์ของแต่ละตัวแตกต่างกันไปซึ่งมีเครื่องมือจำลองการทำงานที่สำคัญได้แก่

3.1.2 REAL (REalistic And Large) [39] พัฒนาโดย S. Keshav จาก Cornell University ทำงานบนระบบปฏิบัติการ Unix โครงสร้างทางกายภาพของเครือข่าย (Network Topologies) และการกำหนดสถานการณ์ของเครือข่าย สามารถเขียนได้โดยการใช้ Script

3.1.2 YATS (Yet Another Tiny Simulator) [40] พัฒนาโดย Dresden University of

Technology ซึ่งจะมีโมดูลที่สำคัญคือการจำลองการทำงานเครือข่ายแบบ ATM (Asynchronous Transfer Mode) ตัวจำลอง YATS นี้ไม่สนับสนุนการทำงานแบบมัลติคาสท์

3.1.3 GloMoSim (Global Mobile Simulator) [41] พัฒนาโดย University of California Los Angeles ซึ่งเป็นตัวจำลองที่เน้นจำลองการทำงานของระบบเครือข่ายไร้สาย (Wireless Network) และระบบเครือข่ายแบบสาย (Wired Network)

3.1.4 ns-2 (Network Simulator version 2) [5] พัฒนาโดย Information Sciences Institute ในโครงการ VINT Project ที่ Lawrence Berkeley National Laboratory (LGNL) ซึ่งพัฒนาโดยใช้ภาษาเชิงวัตถุ (Object Oriented Programming) ด้วยภาษา C++ NS ได้เริ่มมีการพัฒนารุ่นที่ 1 เมื่อปี ค.ศ. 1995 ในขณะที่ REAL Simulator ยังได้รับความนิยมอย่างสูง เมื่อทาง ns-1 เริ่มมีการทดลองใช้งาน ปรากฏว่าค่อนข้างยืดหยุ่นทำให้เริ่มได้รับความนิยมมากขึ้นเรื่อยๆ จนกระทั่งถึงปัจจุบัน ไดออกเป็นรุ่นที่ 2.31 การทดสอบโพรโทคอลโดยใช้ ns-2 สามารถติดตั้งโพรโทคอลที่พัฒนาหรือออกแบบ เข้าไปในตัวจำลองได้ โดยการเขียนคำสั่งเพื่อสร้างโพรโทคอล กำหนดสถานการณ์หรือสภาพแวดล้อมของระบบเครือข่าย คำสั่งของ ns-2 ที่ใช้เขียนควบคุมการทำงาน เรียกว่า object-oriented Tool Command Language (oTCL) script ซึ่งเป็นคำสั่งที่มีการพัฒนาโดยใช้ภาษา C++ และที่สำคัญ ns-2 มีโมดูลของการสื่อสารข้อมูลระหว่างเครือข่ายแบบมัลติคาสท์ ดังนั้น ns-2 จึงเหมาะที่จะนำมาใช้ในงานวิจัยที่เกี่ยวข้องกับการสื่อสารข้อมูลแบบมัลติคาสท์

3.2 ขั้นตอนในการจำลองระบบเครือข่าย

ขั้นตอนการจำลองระบบเครือข่าย ด้วย Network Simulator ในการนำตัวจำลองการทำงานระบบเครือข่ายมาใช้ในการทดลองหาเพื่อผลลัพธ์ของงานวิจัยจะมีขั้นตอนในการดำเนินการทดลองทั้งหมด 6 ขั้นตอน ดังรายละเอียดภาพประกอบ 11

1. กำหนดจุดประสงค์ของการทดลอง คือการกำหนดเป้าหมายของงานวิจัยว่าต้องการทราบอะไรจากการทดลองด้วยตัวจำลองระบบเครือข่าย

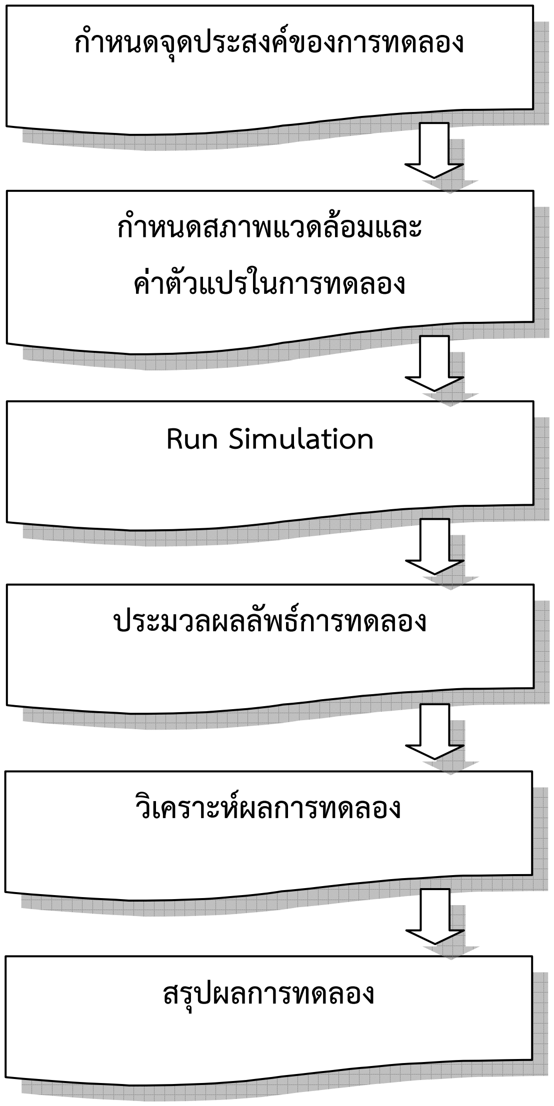
2. กำหนดสภาพแวดล้อมและค่าตัวแปรในการทดลอง เนื่องการสภาพแวดล้อมของระบบเครือข่ายนั้นมีผลต่อการทดลองต่างกัน ดังนั้น เราจึงจำเป็นจะต้องกำหนดสภาวะแวดล้อมของระบบเครือข่ายตามที่เราต้องการก่อนที่จะทดลอง

3. Run Simulation คือ การเริ่มดำเนินการทดลองด้วยตัวจำลองระบบเครือข่าย โดยนำสภาพแวดล้อมและค่าตัวแปรในการทดลองที่ได้กำหนดไว้แล้วมาทำการเขียน Script เพื่อทำการจำลองใน Network Simulator เพื่อที่จะทำให้ได้ผลลัพธ์จากการทดลอง

4. ประมวลผลลัพธ์การทดลอง คือ การประมวลผลลัพธ์ให้อยู่ในรูปแบบที่ต้องการโดยการเขียนโปรแกรมแยกเอาส่วนที่ต้องการมาใช้ เนื่องจากลัพธ์จากการทดลองที่ได้นั้นเป็น text ไฟล์ที่มีทั้งส่วนที่ต้องการและไม่ต้องการ

5. วิเคราะห์ผลการทดลอง คือการวิเคราะห์ผลการทดลองของแต่ละสถานการณ์ที่ได้ทำการทดลองไป โดยมุ่งไปที่การตอบคำถามของประเด็นปัญหาการวิจัยที่ได้กำหนดไว้

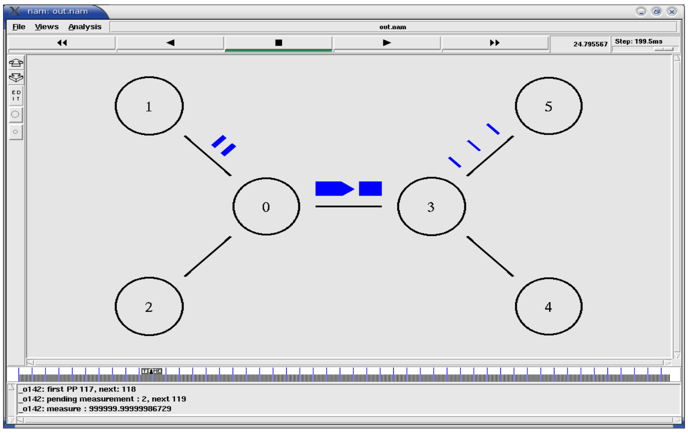
6. สรุปผลการทดลอง คือ การสรุปผลที่ได้จากการวิเคราะห์ เพื่อสร้างองค์ความรู้ใหม่ให้เป็นแนวทางในการพัฒนาโพรโทคอล MR-MCC ต่อไปในอนาคต



ภาพประกอบ 11 ขั้นตอนการจำลอง Network ด้วย Network Simulation

3.3 Network Simulation 2 (ns-2)

Ns (Network Simulator) [5] นั้นเป็นส่วนหนึ่งของโครงการ Virtual InterNet Test-bed (VINT) ที่ถูกพัฒนาขึ้นใน Lawrence Berkeley National Laboratory (LBNL) ซึ่งเป็น Simulator ที่ใช้ในการแบ่งแยกเหตุการณ์ต่างๆภายในสถานการณ์ที่ได้จำลองระบบเครือข่ายขึ้น ซึ่ง ns-2 นั้นสนับสนุนการจำลองระบบเครือข่ายต่าง ๆ เช่น TCP, routing และ Multicast protocol ซึ่งทำงานอยู่บนอุปกรณ์แบบสายและไร้สาย จากที่กล่าวมาข้างต้นทำให้งานวิจัยนี้เลือกใช้ ns-2 ในการศึกษาวิจัย

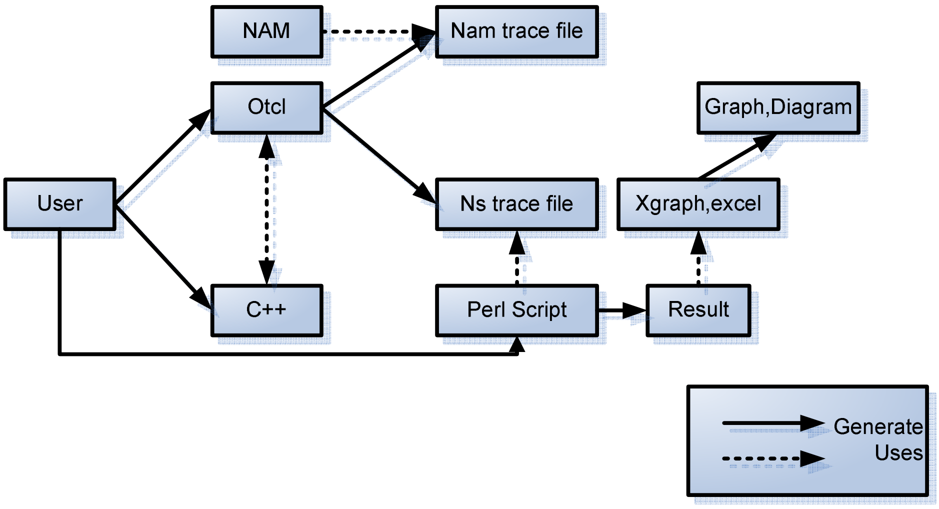


ภาพประกอบ 12 แสดงผลการรันโปรแกรม NS-2

3.3.1 ขั้นตอนการทำงานของ Network Simulator

จากภาพประกอบ 13 จะเป็นขั้นตอนการทำงานของ Network Simulator โดย User จะสามารถสร้าง oTCL Script, C++ Network elements และ Perl Script เพื่อที่จะนำมาใช้ในการออกแบบเพื่อจำลองการทำงานของ Network ซึ่ง User สร้าง C++ network ขึ้นมาเพื่อให้เป็น object ที่อยู่ใน library ที่สามารถจะเรียกใช้ได้โดยเรียกใช้ผ่าน Command ของ oTCL Script ซึ่งจะมี OTCL linkage เป็นตัวที่จะ เรียก Object ใน library ขึ้นมาใช้งาน เมื่อ oTCL script ถูกประมวลผล oTCL script จะสร้าง nam trace file และ ns trace file ซึ่งตัว nam trace file ที่ได้นั้นจะถูกเรียกใช้โดย nam เพื่อที่จะนำไปประมวลผลแล้วแสดงผลออกมาในรูปกราฟฟิก ในส่วนของ perl script นั้นจะเรียกใช้ ns trace file เพื่อนำ ns trace file มา filter ให้ได้ ผลลัพธ์ออกมาเป็น Result ที่จะถูกเรียกใช้โดย Visualization tool เช่น Xgraph, gupolt, excel เพื่อที่จะนำมาใช้สร้างกราฟที่แสดงผลการทำงานของ

การจำลอง Network



ภาพประกอบ 13 ขั้นตอนการทำงานของ Network Simulator

3.3.2 ส่วนประกอบของ ns-2

ns-2 Simulator มีส่วนประกอบต่างๆที่รวมเข้าไว้ด้วยกันหลายส่วน เรียกว่า แพคเกจ สามารถเลือกที่จะติดตั้งเพียงบางส่วนได้ แต่จำเป็นต้องติดตั้งส่วนหลักก่อนเสมอ ซึ่งจะประกอบด้วย แพคเกจหลัก(จำเป็นต้องติดตั้ง)

3.3.2.1 Tcl/Tk (Tool command language/Tool kit) เป็นแพคเกจที่ใช้ในการสร้าง command script ที่จะนำมาเขียนเพื่อควบคุมการทำงานบน ns-2 โดย จะมีการสร้าง script โดยภาษา C++

3.3.2.2 oTCL (object Tool command language) เป็นแพคเกจที่สนับสนุนการทำงานของ Tcl ให้มีลักษณะเป็นโปรแกรมเชิงวัตถุ เพื่อให้ง่ายต่อการนำไปใช้

3.3.2.3 TclCL เป็นแพคเกจที่รวบรวม library ที่ได้มีการสร้างไว้แล้วของ Tcl

3.3.2.4 ns-2 core package ของโปรแกรมการทำงานของ ns-2 ร่วมกับ Tcl และ C++ ที่มีการคาบเกี่ยวกันและใช้ library บางส่วนและส่งข้อมูลระหว่างกันโดยมี interface ติดต่อกันเรียกว่า oTCL Linkage

3.4 การเตรียมการทดลอง

3.4.1 เกณฑ์ที่นำมาใช้ในการทดลอง

เกณฑ์ที่นำมาใช้ในการทดสอบประสิทธิภาพของโพรโทคอล MR-MCC ที่ให้บริการมัลติมีเดียนั้น เกณฑ์ที่จะนำมาใช้วัดนั้นมีดังต่อไปนี้

3.4.1.1 Fairness คือ ความเป็นมิตรต่อโพรโทคอลอื่น เพราะในระบบเครือข่ายหนึ่งๆ นั้น มีโพรโทคอลทำงานอยู่ร่วมกันมากมายหลายโพรโทคอล ซึ่งการที่โพรโทคอลใดๆสามารถที่จะทำงานร่วมกับโพรโทคอลอื่นๆได้โดยที่ตัวมันเองนั้นไม่ไปสร้างความเสียหายให้กับโพรโทคอลอื่น ในขณะที่ตัวมันเองก็สามารถทำงานได้อย่างราบรื่นนั้นก็แสดงว่า โพรโทคอลนั้นมีความเป็นมิตร (Fairness) กับโพรโทคอลอื่น

3.4.1.2 Smoothness คือความราบเรียบของสัญญาณข้อมูลขณะทำการส่ง ซึ่งการที่โพรโทคอลมีความอ่อนไหวหรือ มีการตอบสนองต่อสภาวะของระบบเครือข่ายที่รวดเร็วเกินไป (Responsiveness) อาจทาให้ข้อมูลที่ผู้รับได้รับไม่มีความต่อเนื่อง เพราะมีการเปลี่ยนแปลงรวดเร็วเกินไป ซึ่งมีผลกระทบอย่างมากต่อการส่งข้อมูลประเภทมัลติมีเดีย ที่เป็น การส่งทั้งภาพและเสียง เพราะภาพและเสียงที่ได้รับนั้นจะไม่ ชัดเจน ซึ่งมีสาเหตุมาจากสัญญาณที่ได้รับนั้นเกิดการกระตุก เช่น สัญญาณภาพและเสียงมีความชัดเจนและไม่ชัดเจนสลับกัน ไป ดังนั้น MR-MCC ที่ดีต้องไม่ปรับเปลี่ยนอัตราการใช้งาน แบนด์วิดท์ขณะที่ทาการรับ-ส่งรวดเร็วมากเกินไป

3.4.2 ตัววัดสมรรถนะ (Performance Metrics)

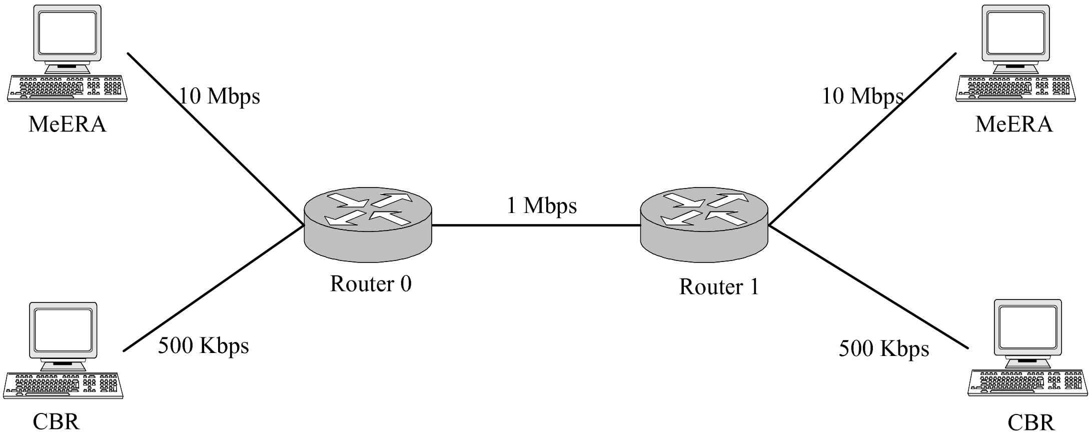
3.4.2.1 ปริมาณงาน (Throughput) คือ ค่าของจำนวนแพ็กเก็ต (หน่วยเป็น บิต) ที่ผู้รับได้รับในแต่ละช่วงเวลา ในการทดลองของงานวิจัยนี้ได้ใช้ค่า Throughput ในหน่วยของ Kbps (กิโลบิตต่อวินาที) โดยทั่วไปแล้ว ค่า Throughput นั้นจะแสดงออกถึงความคับคั่งในระบบเครือข่ายด้วย กล่าวคือ ถ้าค่า Throughput มีค่าที่ลดลงอย่างรุนแรง แสดงว่าในระบบเครือข่ายนั้นได้เกิดความคับคั่งขึ้นแล้ว แต่ถ้าค่า Throughput มีค่าเพิ่มขึ้น ก็แสดงว่าความคับคั่งในระบบเครือข่ายได้ลดลงแล้ว

3.4.2.2 อัตราการสูญเสียข้อมูล (Packet Loss Ratio) คือ อัตราการสูญเสียของข้อมูลทั้งหมดขณะที่ทำการส่งตลอดการทดลอง ซึ่งค่า PLR ที่สูงสามารถแสดงถึงค่า Throughput ที่ต่ำด้วย อีกทั้งยังแสดงถึงประสิทธิภาพในการส่งข้อมูลด้วย จากการทดลองของ Boyce และ Galianello [42] แสดงให้เห็นว่าถ้ามีค่า PLR เกินกว่า 3% แล้วจะทำให้คุณภาพของภาพและเสียงที่ทำการรับส่งนั้นลดลง 30%

3.4.3 โครงสร้างเครือข่ายที่ใช้ในการทดลอง (Network Topology)

การทดสอบโพรโทคอลของเรานั้นจะใช้ โทพอโลยี (Topology) แบบตุ้มน้ำหนัก (Dumbbell) เป็นมาตรฐานในการทดสอบเพราะเป็นโทพอโลยีที่สามารถเข้าใจได้ง่าย และมีการใช้อย่างแพร่หลายในการวิจัยเกี่ยวกับระบบเครือข่าย ซึ่งสามารถที่จะวิเคราะห์ผลได้ถูกต้องกว่าการที่จะกำหนดสถานการณ์ที่ซับซ้อนแล้วทำให้เกิด Parameter ขึ้นมากมายโดยที่ไม่อาจจะสามารถควบคุมได้ อีกทั้งการออกแบบให้มีจำนวนผู้รับที่หลากหลายซับซ้อนสำหรับมัลติคาสท์แล้วไม่ได้ส่งผลถึงทราฟฟิกที่เกิดขึ้น เพราะมัลติคาสท์มีการส่งข้อมูลเพียงหนึ่งโฟล์วเท่านั้น ถึงแม้ว่าจะมีผู้รับจำนวนเท่าใดก็ตาม ดังนั้นการเลือกใช้โทพอโลยีที่เข้าใจได้ง่ายจึงเป็นวิธีที่ดีที่สุดในการจำลองระบบเครือข่าย [24] โดยมีการกำหนดโทพอโลยีในงานวิจัยนี้ออกเป็น 2 กรณี ดังนี้

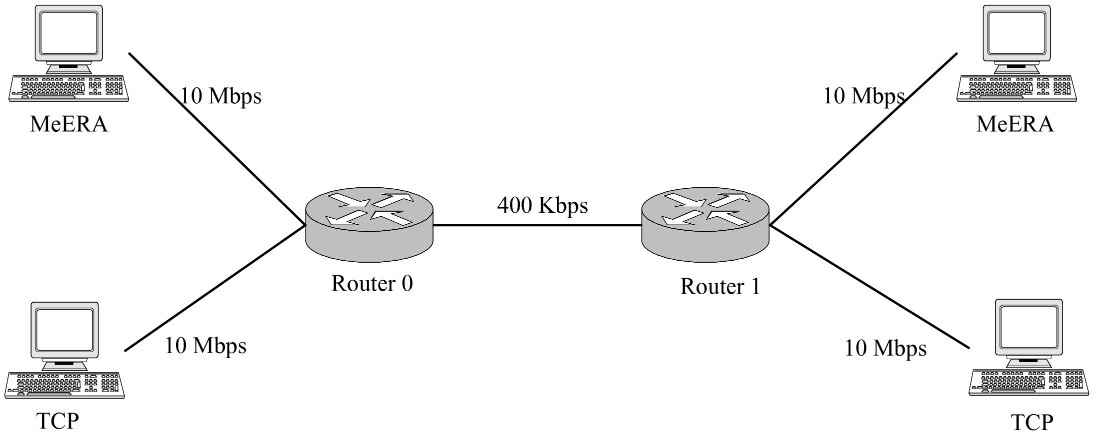
กรณีที่ 1 โทพอโลยีแบบตุ้มน้ำหนักที่มีการทำงานร่วมกับอัตราบิตคงที่ (Constant bitrate)



ภาพประกอบ 14 โครงสร้างเครือข่ายกรณีที่ 1

จากภาพประกอบ 14 เป็นการออกแบบโครงสร้างของเครือข่ายเพื่อจุดประสงค์ในการทดสอบความสามารถในการตอบสนองต่อสถานการณ์ของระบบเครือข่าย โดยมีอัตราบิตคงที่ (CBR) เป็นตัวสร้างสถานการณ์การแบ่งปันแบนด์วิดท์ในระบบ ซึ่งจำลองการสื่อสารข้อมูลออกเป็นสองฝั่งโดยกำหนดให้มีฝั่งส่งและฝั่งรับ ที่รับส่งข้อมูลผ่านลิงค์คอขวดเดียวกัน โดยมีการใช้โทพอโลยีนี้ในการทดลองที่ 1

กรณีที่ 2 โทพอโลยีแบบตุ้มนำหนักที่มีการทำงานร่วมกับทีซีพี



ภาพประกอบ 15 โครงสร้างเครือข่ายกรณีที่ 2

จากภาพประกอบ 15 เป็นการออกแบบโครงสร้างของเครือข่ายเพื่อจุดประสงค์ในการทดสอบประสิทธิภาพของโพรโทคอล MeERA ว่าจะสามารถทำงานร่วมกับโพรโทคอลทีซีพีภายใต้กฎความเป็นมิตรต่อทีซีพีได้หรือไม่ โดยกำหนดให้โพรโทคอล MeERA และทีซีพี ทำงานร่วมกันในการรับส่งข้อมูลผ่านลิงค์คอขวดเดียวกัน โดยมีการใช้โทพอโลยีนี้ในการทดลองที่ 2 ถึงการทดลองที่ 8

3.4.4 การกำหนดค่าพารามิเตอร์ในการทดลอง

การทดลองการโดยใช้ตัวจำลองการทำงานของเครือข่ายจะสามารถกำหนดค่าพารามิเตอร์ต่างๆได้ โดยใช้วิธีการเขียนโปรแกรมเพื่อควบคุมและกำหนดค่าของพารามิเตอร์ต่างๆที่เกี่ยวข้อง การทดลองเพื่อประเมินประสิทธิภาพของโพรโทคอล MR-MCC ในการให้บริการมัลติมีเดียภายใต้กฎความเป็นมิตรต่อทีซีพีจะต้องมีการกำหนดค่าพารามิเตอร์เพื่อสร้างสภาพแวดล้อมของเครือข่ายที่ใช้ในการทดลอง ดังนี้

3.4.4.1 ขนาดของแพ็คเก็ตข้อมูล (Packet size)

ในการทดลองของงานวิจัยนี้มีการกำหนดขนาดของแพ็คเก็ตข้อมูลที่มีขนาดเท่ากับขนาดของแพ็คเก็ตข้อมูลของทีซีพี ซึ่งมีค่าเท่ากับ 512 kilobyte

3.4.4.2 ขนาดชั้นข้อมูล (Layer size)

ในการสื่อสารข้อมูลโดยใช้มัลติคาสท์โพรโทคอลแบบหลายอัตราส่งที่มีการเข้ารหัสข้อมูลเป็นชั้นเพื่อให้รับส่งข้อมูลได้ในอัตราส่งที่แตกต่างกันไปนั้น การกำหนดขนาดของชั้นข้อมูลมีความจำเป็นที่ต้องคำนึงถึงอย่างมากเนื่องจากมัลติคาสท์โพรโทคอลแบบหลายอัตราส่งจะทำการบอกรับชั้นข้อมูล (Subscribe Layer) ทีละชั้นไปเรื่อยๆจนกว่าจะได้ชั้นข้อมูลที่เหมาะสมกับแบนด์วิดท์ที่เหลืออยู่ แต่เนื่องจากขนาดของชั้นข้อมูลนั้นมีผลโดยตรงกับอัตราการบอกรับชั้นขอมูลและ

คุณภาพของมัลติมีเดียที่ผู้รับได้รับ ทำให้การกำหนดขนาดของชั้นข้อมูลในงานวิจัยนี้ได้กำหนดไว้ 2 กรณี ดังนี้

กรณีที่ 1 การกำหนดชั้นข้อมูลแบบเท่ากันหมด (equal) โดยมีการกำหนดขนาดชั้นข้อมูลไว้ที่ขนาด 20 kbps เพื่อให้เกิดการบอกรับข้อมูลในอัตราคงที่ แต่วิธีการกำหนดขนาดชั้นข้อมูลแบบนี้มีข้อด้อยในการก่อให้เกิดช่องว่างของแบนด์วิดท์ (gap) ที่ไม่ถูกใช้งานในกรณีที่มีแบนด์วิดท์ที่ว่าอยู่ไม่ถึง 20 kbps โพรโทคอลจะไม่บอกรับข้อมูลทำให้เกิดช่องว่างดังกล่าว

กรณีที่ 2 เป็นการกำหนดขนาดชั้นข้อมูลตามระดับคุณภาพของสัญญาณมัลติมีเดีย เช่น 8 kbps, 24 kbps, 64 kbps, 32 kbps, 64 kbps, 96 kbps ทั้งนี้เพื่อสร้างสถานการณ์เครือข่ายให้โพรโทคอล MR-MCC ทำการบอกรับแบนด์วิดท์สะสมในอัตราที่ดีที่สุดตามระดับคุณภาพของมัลติมีเดียที่อ้างอิงใน ตารางที่ 2 [12]

3.4.4.3 แบนด์วิดท์ของลิงค์คอขวด (Bottleneck link bandwidth)

การศึกษาความเป็นมิตรของโพรโทคอลมัลติคาสท์ที่มีต่อทีซีพีนั้น ในการทดลองจำเป็นต้องมีการออกแบบให้มีการรับส่งข้อมูลผ่านลิงค์คอขวดเดียวกันเพื่อให้ทราบถึงการใช้งานแบนด์วิดท์ของโพรโทคอลทั้งสอง โดยมีหลักการกำหนดแบนด์วิดท์ลิงค์คอขวดว่าจะต้องมีขนาดกว้างพอที่จะรองรับปริมาณข้อมูลทั้งหมดที่จะถูกส่งผ่านไป และต้องมีขนาดที่ไม่กว้างเกินกว่าขนาดของแบนด์วิดท์หลังเราท์เตอร์ เพื่อให้การส่งข้อมูลจากต้นทางทำได้อย่างต่อเนื่องตัวอย่างเช่นแบนด์วิดท์หลังเราท์เตอร์กำหนดให้มีค่าเท่ากับ 10 Mbps ดังนั้นแบนด์วิดท์ของลิงค์คอขวดต้องมีความน้อยกว่า 10 Mbps ทั้งนี้เพื่อสร้างสถานการณ์ความคับคั่งในระหว่างการทดลองด้วย และถ้าในการทดสอบโพรโทคอลแล้วมีการกำหนดให้มีการสื่อสารข้อมูลผ่านลิงค์คอขวดเท่ากับ 2 การเชื่อมต่อในเวลาเดียวกัน และกำหนดให้ลิงค์คอขวดมีแบนด์วิดท์เท่ากับ1 Mbps นั่นหมายความว่า แต่ละการเชื่อมต่อจะใช้แบนด์วิดท์ได้ประมาณ 500 kbps นั่นเอง ในงานวิจัยนี้จึงกำหนดขนาดของลิงค์คอขวดให้มีขนาดที่แตกต่างกันไปตามสถานการณ์ เพื่อให้ได้ผลการทดลองที่หลากหลายเช่น 1 หรือ 10 Mbps

3.4.4.4 แบนด์วิดท์หลังเราท์เตอร์ทั้งฝั่งส่งและฝั่งรับ

การรับส่งข้อมูลจากฝั่งส่งไปยังฝั่งรับโดยการส่งผ่านเราท์เตอร์ออกไป หากต้องการสร้างสถานการณ์ความคับคั่งให้เกิดขึ้นในระบบ ปริมาณของข้อมูลที่ส่งต้องมีจำนวนมากกว่าลิงค์คอขวดที่มีอยู่เพื่อให้เห็นความสามารถในการตอบสนองต่อสถานการณ์ความคับคั่งที่เกิดขึ้นของโพรโทคอล จึงกำหนดให้มีแบนด์วิดท์หลังเราท์เตอร์มีขนาดที่กว้างกว่าลิงค์คอขวด โดยอาจกำหนดให้มีค่าเท่ากับ 10 Mbps เป็นต้น

3.4.4.5 เวลาที่ใช้ในการทดลอง

ในงานวิจัยทางด้านระบบเครือข่าย การทดลองโดยการใช้ตัวจำลองการทำงานของเครือข่ายนั้นเพื่อให้ได้ผลลัพธ์ที่ใกล้เคียงความเป็นจริง ความสามารถของเครื่องมือจำลองมีส่วนสำคัญอย่างมาก ในงานวิจัยนี้เลือกใช้ ns-2 ในการจำลองการทำงานของโพรโทคอล จากคุณสมบัติของ ns-2 ได้ระบุว่าในการจำลองการทำงานนั้นเมื่อ ns-2 เริ่มทำงานจะอยู่ในสถานะเริ่มต้น ซึ่งโปรแกรมจะมีการประมวลผลเพื่อเรียนรู้โพรโทคอลและสร้างสภาพแวดล้อมที่จำเป็นในการทดลอง ซึ่งจะใช้เวลาในระยะหนึ่ง แล้ว ns-2 จึงจะเริ่มเข้าสู่สถานะที่มั่นคงสม่ำเสมอ (Steady State) อาจใช้เวลาประมาณ 10 ถึง 20 วินาที ดังนั้นในการทดลองของงานวิจัยนี้จะเริ่มทดลองและวัดผลการทดลองตั้งแต่ประมาณ 20 วินาทีเป็นต้นไป เพื่อให้ได้ผลการทดลองที่เชื่อถือได้ ตรงตามข้อกำหนดของเครื่องมือที่ใช้ในการทดลอง

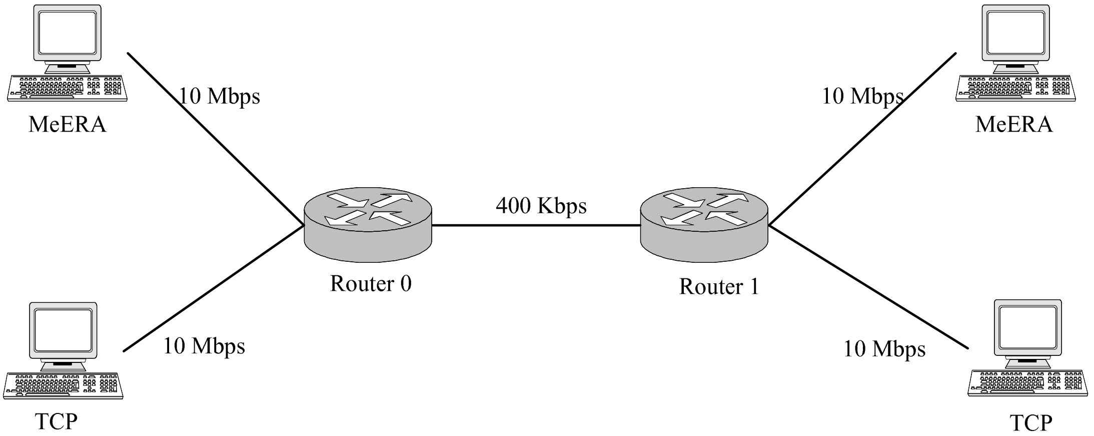
3.5 การดำเนินการทดลอง

3.5.1 แนวทางในการทดลอง

ในการทดลองเพื่อศึกษาเกี่ยวกับประสิทธิภาพของโพรโทคอลมัลติคาสท์ที่ให้บริการมัลติมีเดียภายใต้กฎความเป็นมิตรต่อทีซีพีนั้น จะทำการทดลองโดยอาศัยการกำหนดสถานการณ์เพื่อแสดงให้เห็นถึงปัญหาที่เกิดขึ้น เช่น การกำหนดขนาดของชั้นข้อมูลที่ไม่สัมพันธ์กันกับระดับคุณภาพของมัลติมีเดีย การก่อให้เกิดการตอบสนองต่อสภาวะเครือข่ายอย่างรวดเร็ว เหล่านี้ จะแสดงให้เห็นถึงผลการทดลองที่ชัดเจนว่า เมื่อโพรโทคอลมัลติคาสท์ต้องปฏิบัติตามกฎความเป็นมิตรต่อทีซีพีแล้วเกิดข้อด้อยอย่างไรในการให้บริการมัลติมีเดีย โดยมีการดำเนินการทดลอง ดังนี้

3.5.1.1 การทดสอบความสามารถของมัลติคาสท์โพรโทคอลเพื่อยืนยันความสามารถของ โพรโทคอลที่ผู้พัฒนาได้นำเสนอ

การทดลองเพื่อทำการทดสอบการทำงานของโพรโทคอลทดสอบอีกครั้ง ภายใต้สภาพแวดล้อมและสถานการณ์เดียวกับผู้พัฒนาโพรโทคอลนั้นๆได้นำเสนอนั้นมีความจำเป็นอย่างยิ่ง เพราะจะเป็นสิ่งช่วยยืนยันความถูกต้องของโพรโทคอลที่จะนำมาทดสอบว่าเป็นไปตามที่ผู้พัฒนาได้นำเสนอหรือไม่ โพรโทคอลที่จะนำมาทดสอบเพื่อประเมินประสิทธิภาพในการวิจัยนี้ คือ MeERA ที่ผู้พัฒนาได้ยืนยันว่าสามารถปฏิบัติตามกฎความเป็นมิตรต่อทีซีพีได้เป็นอย่างดี ดังนั้นในงานวิจัยนี้จึงทำการทดลองอีกครั้ง เพื่อยืนยันความสามารถดังกล่าว ภายใต้เงื่อนไขและข้อกำหนดที่ให้มีการทำงานร่วมกันระหว่างทีซีพีและโพรโทคอล MeERA บนตัวจำลองการทำงานของเครือข่าย ns-2 และโครงสร้างเครือข่ายทดสอบเป็นดังภาพประกอบ 16



ภาพประกอบ 16 โครงสร้างเครือข่ายในการทดลอง 2

การกำหนดค่าพารามิเตอร์ต่างๆเพื่อสร้างสภาพแวดล้อมในการทดลองที่ 2 เพื่อทำการทดลองยืนยันความสามารถของโพรโทคอลที่นำมาใช้ในงานวิจัยนี้ มีการกำหนดให้มีแบนด์วิดท์ลิงค์คอขวดเท่ากับ 400 kbps และมีแบนด์วิดท์หลังเราท์เตอร์ของทั้งสองโพรโทคอลเท่ากับ 10 Mbps เวลาที่ใช้ในการทดลองเท่ากับ 200 วินาที และมีการทดลองซ้ำ 30 รอบ เพื่อนำไปใช้ในการประมาณค่าเพื่อให้ได้ผลการทดลองที่น่าเชื่อถือ [43] ผลการทดลองจากตัวจำลองการทำงานของเครือข่ายจะเป็นลักษณะของไฟล์ข้อมูลที่สามารถนำมาแสดงเป็นกราฟ เพื่อให้ทำความเข้าใจได้ง่ายยิ่งขึ้น ซึ่งผลการทดลองของการทดลองที่ 2 จะแสดงในบทต่อไป

3.5.1.2 การทดลองเพื่อวิเคราะห์ประเด็น Loss tolerant / Delay sensitive vs. Delay tolerant / Loss sensitive

ในประเด็นคุณสมบัติการส่งข้อมูลที่แตกต่างกันของทีซีพีและมัลติมีเดีย ที่มัลติมีเดียจะต้องเกิดความล่าช้า (Delay) ระหว่างผู้ส่งกับผู้รับน้อยที่สุด และยอมให้เกิดความสูญหายของข้อมูลบางส่วนเพื่อให้ข้อมูลส่วนใหญ่ยังสามารถทำงานได้ คุณสมบัติที่สำคัญนี้เรียกว่า Loss Tolerant /Delay sensitive ซึ่งเป็นคุณสมบัติที่ตรงกันข้ามกับการส่งข้อมูลของทีซีพีที่ยอมให้เกิดความล่าช้า แต่ไม่ยอมให้เกิดการสูญเสียข้อมูล (Delay tolerant / Loss sensitive) การที่มัลติคาสท์พยายามทำตัวให้เหมือนกับทีซีพี เมื่อให้บริการมัลติมีเดียอาจทำให้เกิดปัญหาความล่าช้าจนล้มเหลวในการส่งข้อมูลที่มีคุณภาพ ซึ่งในผลการทดลองที่ 3-5 แสดงให้เห็นถึงประเด็นปัญหาดังกล่าว

3.5.1.3 การทดลองเพื่อวิเคราะห์ประเด็น Responsiveness vs. Smoothness

โพรโทคอล MR-MCC ที่ให้บริการมัลติมีเดียนั้น ไม่สามารถที่จะตอบสนองต่อความคับคั่งอย่างรวดเร็วได้ เพราะต้องคำนึงถึงความราบเรียบ (Smoothness) ของการส่ง ซึ่งจะส่งผลต่อคุณภาพที่ผู้รับได้รับด้วย จากประเด็นนี้จึงสันนิษฐานได้ว่า โพรโทคอล MR-MCC แบบเดิมที่ได้มีการเสนอมานั้น อาจมีปัญหาในประเด็นของความราบเรียบในการส่ง เพราะ โพรโทคอลเหล่านั้น

ต่างถูกออกแบบให้ตอบสนองต่อความคับคั่งอย่างรวดเร็ว ซึ่งแน่นอนว่าส่งผลกระทบต่อความราบเรียบด้วย ซึ่งในผลการทดลองที่ 5 แสดงให้เห็นถึงประเด็นปัญหาดังกล่าว

3.5.1.4 การทดลองเพื่อวิเคราะห์ประเด็น Bandwidth Consumption

การเข้าใช้แบนด์วิดท์ที่มีอยู่อย่างรวดเร็วอาจไม่เกิดประโยชน์ต่อคุณภาพของข้อมูลที่ผู้รับได้รับ แต่อาจก่อให้เกิดปัญหาเรื่องความราบเรียบของข้อมูลได้ การทดลองที่ 6 แสดงให้เห็นว่า การเพิกเฉยต่อแบนด์วิดท์ที่มีอยู่ส่งผลดีมากกว่าการเข้าใช้แบนด์วิดท์อย่างรวดเร็ว

3.5.1.5 การทดลองเพื่อวิเคราะห์ประเด็น Layer Granularity

การกำหนดค่าที่เกี่ยวกับชั้นของการสื่อสารที่ไม่เหมาะสมกับมาตรฐานการเข้ารหัส (CODEC) ข้อมูลของมัลติมีเดียนั้นส่งผลต่อการทำงานของมัลติมีเดียอย่างเห็นได้ชัด โดยในการทดลองที่7-8 แสดงให้เห็นถึงข้อแตกต่างระหว่างการเข้ารหัสไฟล์มัลติมีเดียแบบ H.264 และ MPEG2

3.6 การบันทึกผลการทดลอง

3.6.1 ข้อมูลที่ได้จากการทดลอง

ผลการทดลองที่ได้จาก Network Simulator 2 นั้นซึ่งมีอยู่ 2 แบบ คือ

3.6.1.1 Trace

Trace ไฟล์นั้นเป็นไฟล์ที่เก็บรวบรวมเหตุการณ์ต่างๆ ที่เกิดขึ้นในระหว่างการทดลอง ซึ่งมีการเก็บค่าและบันทึกผลลงในไฟล์ติดตาม (Trace) ขณะตัวจำลองทำงาน โดยรายละเอียดของ trace file นั้นจะมีรูปแบบดัง ตาราง 7 และ ตาราง 8

ตาราง 7 ตารางรูปแบบของ Trace ไฟล์

event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	Pkt id

ตาราง 8 ตารางความหมายของ Trace ไฟล์ แต่ละคอลัมน์

คอลัมน์		ความหมาย
Event	=	เหตุการณ์ที่เกิดขึ้น ณ เวลานั้น ที่สำคัญประกอบด้วย r (receive at to node), + (enqueue at queue), - (dequeue at queue) และ d (drop at queue)
Time	=	เวลา
from node	=	node ที่ ส่ง packet
to node	=	node ที่รับ packet
pkt type	=	ชนิดของ packet เช่น tcp, cbr, ack
pkt size	=	ขนาดของ packet
flags	=	เป็นคอลัมน์ที่ไม่ถูกใช้งาน
fid	=	flow id
src addr	=	source address
dst addr	=	destination address
seq num	=	ลำดับของ packet
pkt id	=	packet id

ตัวอย่าง Trace ไฟล์ ที่ได้จากการทดลอง

r 49.684574 1 0 cbr 512 ------- 1 1.2 -2147483646.0 600 6398

+ 49.684574 0 3 cbr 512 ------- 1 1.2 -2147483646.0 600 6398

r 50.285862 3 4 tcp 552 ------- 1 2.0 4.0 298 6307

ซึ่งจาก Trace ไฟล์ที่ได้เราจะต้องนำไปทำการคำนวณเพื่อหาค่า Throughput ของแต่ละโหนด

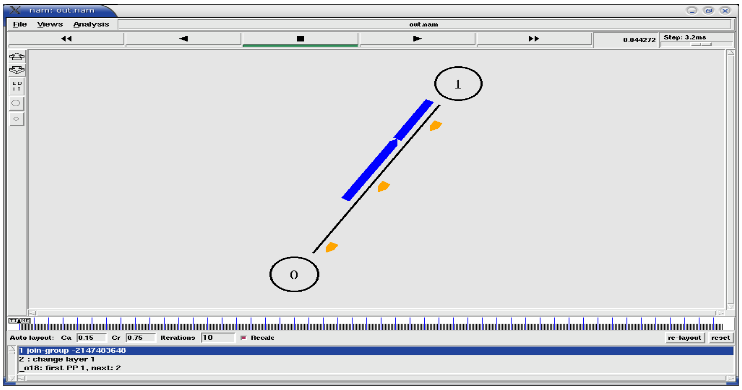
3.6.2 การแสดงผลการทดลอง

การแสดงผลของข้อมูลในการทดสอบโพรโทคอลนั้นจะสามารถแสดงผลออกมาได้เป็น 2 ลักษณะคือการแสดงผลออกเป็น ภาพเคลื่อนไหว (Animation) และ กราฟ ดังนี้

3.6.2.1 NAM (Network animator tool)

ถูกพัฒนาด้วย TCL/TK ที่ LBL [44] โดยมีจุดประสงค์ในการพัฒนาเพื่อให้ผู้ใช้สามารถที่จะสร้าง ns scripts (ไม่สามารถเรียกใช้งาน module ที่มีอยู่ใน ns-2 ได้ครบ) โดยไม่

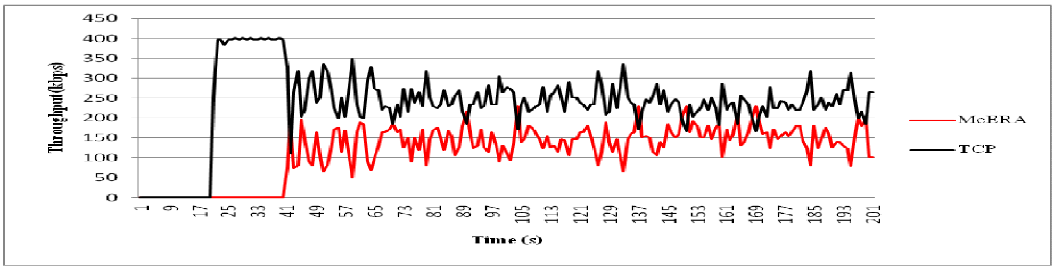
จำเป็นต้องมีความรู้เกี่ยวกับภาษา TCL เพื่อใช้งานในห้องเรียนเพื่อให้นักศึกษาสามารถเข้าใจทฤษฎีของ network ได้ดียิ่งขึ้น แสดงผลที่มีลักษณะเป็น topology layout, การส่งข้อมูลที่เป็นแพ็กเก็ต โดยการนำ name trace file ที่ได้จาก ns-2 มาทำการแปลงออกมาเป็นรูป 2 มิติดัง ภาพประกอบ 17



ภาพประกอบ 17 เมื่อ Network Animator Tool เริ่มมีการทำงาน

3.6.2.2 การวาดกราฟ

เป็นการนำเสนอผลการทดลองที่สามารถเข้าใจง่าย โดยสามารถใช้ Package ที่มีอยู่ในns-2 วาดออกมา แต่ในงานวิจัยนี้ทำการวาดกราฟโดยโปรแกรม MS Excel ซึ่งจะได้ผลลัพธ์ดังตัวอย่างในภาพประกอบ 18



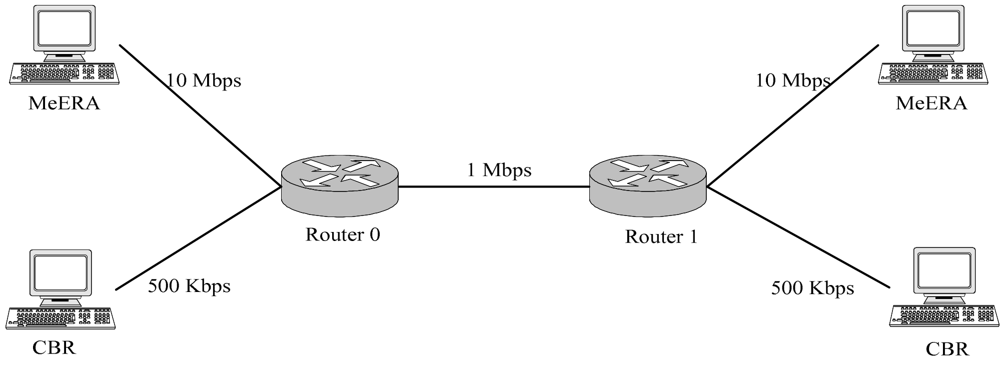
ภาพประกอบ 18 กราฟผลการทดลอง

บทที่ 4

ผลการดำเนินงาน

จากที่ได้ทำการนำเสนอแนวทางการดำเนินการทดลองของงานวิจัยมาแล้วในบทที่ 3 ในบทนี้จะเป็นการนำเสนอผลการทดลองในเชิงของการวิเคราะห์โดยสรุปถึงจุดที่สำคัญอันก่อให้เกิดความรู้จากงานวิจัยนี้ ซึ่งมุ่งเน้นในการศึกษาความสามารถในการส่งผ่านข้อมูลมัลติมีเดียของมัลติคาสท์โพรโทคอลที่ได้นำเสนอคือโพรทคอล MeERA ในมุมมองที่ผู้พัฒนาไม่ได้ทำการทดลองหรือคาดการณ์ไว้ ทำให้เกิดข้อด้อยหรือปัญหาต่อการส่งผ่านข้อมูลมัลติมีเดียของมัลติคาสท์โพรโทคอลที่ออกแบบตามหลักการรักษาความเป็นมิตรต่อทีซีพี และในบทนี้ยังเป็นการประมวลผลการทดลองแต่ละการทดลองเพื่อนำเสนอสาเหตุของปัญหาที่เกิดขึ้น

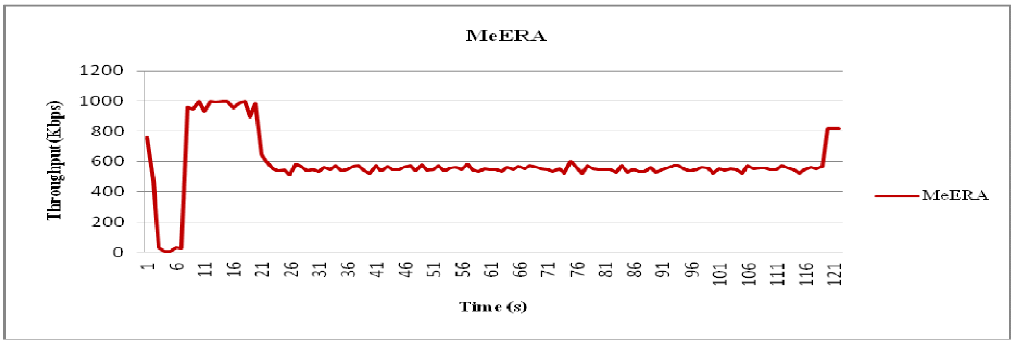
4.1 การทดลองที่ 1 การทดสอบเพื่อยืนยันความสามารถของโพรโทคอล MeERA ในการตอบสนองต่อสภาวะเครือข่าย



ภาพประกอบ 19 โทพอโลยีที่ใช้ในการทดลองที่ 1

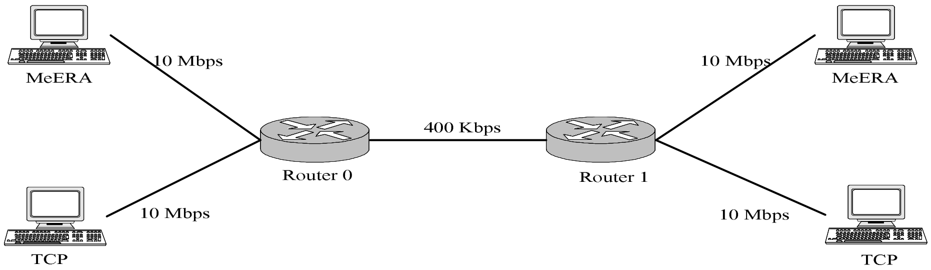
ความสามารถในการตอบสนองต่อสภาวะของระบบเครือข่ายนั้นนับได้ว่ามีความสำคัญมาก เพราะโพรโทคอลควบคุมความคับคั่งที่ดีนั้น นอกจากจะช่วยแก้ความคับคั่งของระบบเครือข่ายแล้ว ตัวโพรโทคอลเองต้องไม่กลายเป็นสาเหตุความคับคั่งของเครือข่ายเสียเองด้วย นอกจากนี้ การทดลองนี้ยังมุ่งประเด็นไปที่ความสามารถในการใช้ประโยชน์แบนด์วิดท์ที่มีอยู่และความราบเรียบในการส่งข้อมูลเมื่อมีการเปลี่ยนแปลงขนาดของแบนด์วิดท์ในระหว่างการสข้อมูล โดยงานวิจัยนี้ใช้โทโพโลยีในการทดลองตามภาพประกอบ 19 ที่มีการส่งข้อมูลของมัลติคาสท์ 1 ช่วงเวลา (session) และมี

อัตราบิตคงตัว (CBR) เป็นตัวสร้างความเปลี่ยนแปลงของแบนด์วิดท์ในระหว่างที่ทำการทดลอง มีแบนด์วิดท์ในการส่งของ MeERA ที่ 10 Mbps มีขนาดแพ็คเก็ตข้อมูลเท่ากับ 512 kb ขนาดชั้นข้อมูลเท่ากับ 20 kbps จำนวน 50 ชั้น แบนด์วิดท์ที่ลิงค์คอขวด 1 Mbps ทำการเริ่มส่งข้อมูลของ MeERA ในวินาทีที่ 5 และเริ่มการส่งข้อมูลของ CBR ในวินาทีที่ 20 โดยมีขนาดแบนด์วิดท์การส่งที่ 500 Kbps ใช้เวลาในการทดลอง 120 วินาที เพื่อแสดงให้เห็นถึงการแบ่งปันแบนด์วิดท์ระหว่าง MeERA และ CBR อย่างชัดเจน ซึ่งจะแสดงให้เห็นว่า MeERA จะทำอย่างไรกับสภาวะที่ขนาดของแบนด์วิดท์ที่ใช้ได้เปลี่ยนแปลงไปผลการทดลองเป็นไปตามที่ผู้พัฒนาได้นำเสนอไว้แล้วคือสามารถตอบสนองต่อสภาวะเครือข่ายได้อย่างรวดเร็ว ดังภาพประกอบ 20



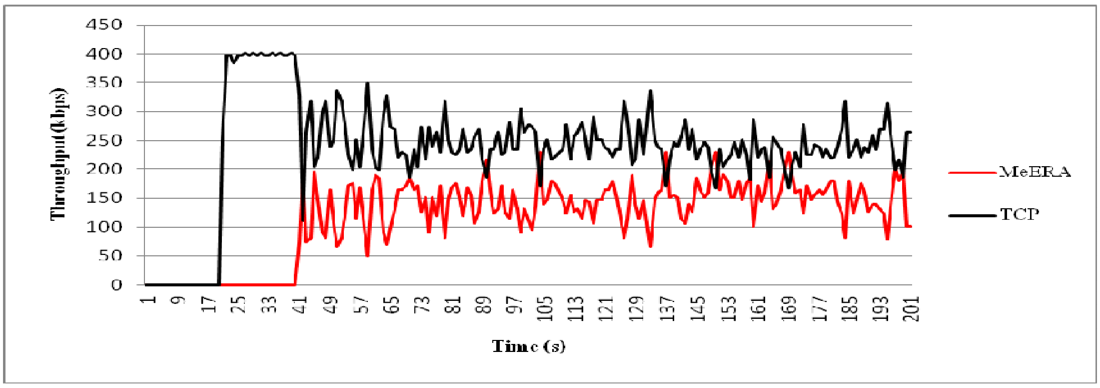
ภาพประกอบ 20 กราฟค่า Throughput ของโพโทคอล MeERA ที่ตอบสนองต่อการเปลี่ยนแปลงของสภาวะในระบบเครือข่าย

4.2 การทดลองที่ 2 การทดสอบเพื่อยืนยันความสามารถของโพรโทคอล MeERA ในการเป็นมิตรต่อทีซีพี



ภาพประกอบ 21 โทพอโลยีที่ใช้ในการทดลองที่ 2

การทดลองนี้มีวัตถุประสงค์เพื่อทำการทดสอบประสิทธิภาพของโพรโทคอล MeERA ว่าจะสามารถทำงานร่วมกับโพรโทคอลทีซีพี ภายใต้กฎความเป็นมิตรต่อทีซีพีได้หรือไม่ โดยกำหนดให้โพรโทคอล MeERA และทีซีพีทำงานร่วมกันบนสภาพแวดล้อมเดียวกับชุดทดสอบ (Test suite) ที่ผู้พัฒนา MeERA ได้กำหนดขึ้น

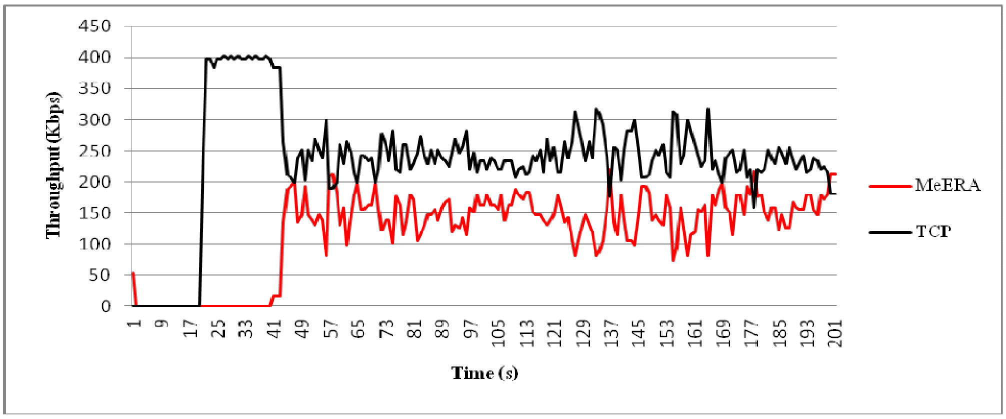


ภาพประกอบ 22 กราฟแสดงการแบ่งปันแบนด์วิดท์ระหว่าง MeERA และ TCP ขณะเกิดสถานการณ์ความคับคั่งในระบบเครือข่าย

จากภาพประกอบ 22 เมื่อพิจารณาถึงผลการทดลองที่เกิดขึ้น พบว่าโพรโทคอล MeERA สามารถรักษาสภาพความเป็นมิตรต่อทีซีพีได้เป็นอย่างดี เมื่ออยู่ภายใต้สภาพแวดล้อมที่กำหนดขึ้น

4.3 การทดลองที่ 3 การทดสอบส่งผ่านข้อมูลเสียง

จากการทดลองที่ 2 เป็นการทดสอบเพื่อแสดงให้เห็นถึงคุณสมบัติของความเป็นมิตรกับโพรโทคอลทีซีพีแต่ในการทดลองนี้ มีวัตถุประสงค์เพื่อทดสอบการทำงานของมัลติคาสท์โพรโทคอลภายใต้กฎความเป็นมิตรต่อทีซีพีจะส่งผลต่อคุณภาพของข้อมูลมัลติมีเดียที่ได้รับหรือไม่ จึงทำการทดลองโดยกำหนดค่าพารามิเตอร์มาตรฐานที่ใช้ในการทดลองนั้นให้คือ ขนาดของแพ็คเก็ตข้อมูลเท่ากับ 512 kb ขนาดชั้นข้อมูลเท่ากับ 20 kbps จำนวน 20 ชั้น มีแบนด์วิดท์ลิงค์คอขวดเท่ากับ 400 kbps แบนด์วิดท์หลังเราเตอร์ของโพรโทคอล MeERA เท่ากับ 1 Mbps ของทีซีพีเท่ากับ 1 Mbps และกำหนดเวลาในการทดลองกับ 200 โดยกำหนดให้ชั้นข้อมูล (Layer) ที่คุณภาพดีที่สุดของข้อมูลมัลติมีเดียอยู่ที่ 192 kbps



ภาพประกอบ 23 กราฟ Throughput ของโพรโทคอล MeERA และ TCP

จากภาพประกอบ 23 ผลการทดลองที่ได้คือ MeERA สามารถบอกรับต่อชั้นข้อมูลที่มีคุณภาพดีที่สุดของข้อมูลมัลติมีเดียที่ 192 kbps ได้เพียงบางช่วง ทั้งนี้เนื่องจากโพรโทคอลทีซีพีมีการใช้งานแบนด์วิดท์ที่ประมาณ 240 kbps อยู่ ทำให้ MeERA ซึ่งต้องตอบรับชั้นข้อมูลครั้งละ 20 kbps ไม่สามารถใช้งานแบนด์วิดท์เกินกว่าทีซีพีได้ ซึ่งเป็นไปตามหลักการของการรักษาความเป็นมิตรต่อทีซีพี

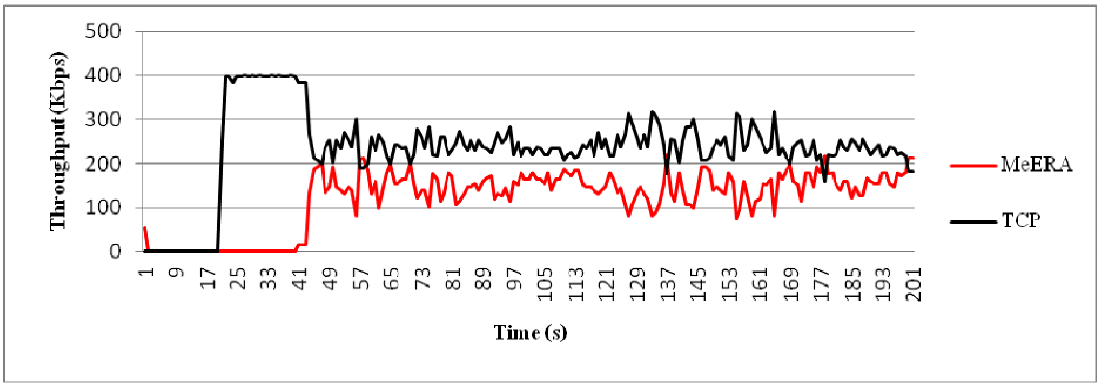
ตาราง 9 เปรียบเทียบอัตราการใช้แบนด์วิดท์เฉลี่ยระหว่าง TCP และ MeERA

โพรโทคอล	ค่าเฉลี่ย Throughput (Kbps)
MeERA	151.39
TCP	239.56

จากผลการทดลองเมื่อพิจารณาจากตารางที่ 2 ในบทที่ 2 จะพบว่าจากอัตราการใช้แบนด์วิดท์เฉลี่ยที่โพรโทคอล MeERA ทำได้อยู่ในระดับที่ต่ำกว่าระดับ Standard Bitrates ของมาตรฐาน DAB อยู่ 2 ชั้นข้อมูลเลยทีเดียว (DAB = 192 Kbps)

4.4 การทดลองที่ 4 การทดสอบการส่งผ่านข้อมูลมัลติมีเดียชนิด Video (MPEG2)

จากการทดลองที่ 3 แสดงให้เห็นว่าภายใต้กฎความเป็นมิตรต่อทีซีพีก่อให้เกิดอุปสรรคต่อการส่งผ่านข้อมูลมัลติมีเดียชนิด Audio (mp3) เนื่องจากไม่สามารถตอบรับแบนด์วิดท์เพื่อให้ได้มาตรฐานตามต้องการ ดังนั้นในการทดลองนี้มีวัตถุประสงค์เพื่อทดสอบการทำงานของมัลติคาสท์โพรโทคอลภายใต้กฎความเป็นมิตรต่อทีซีพีจะส่งผลต่อคุณภาพของข้อมูลมัลติมีเดียชนิด Video (MPEG2) ที่ได้รับหรือไม่ จึงทำการทดลอง โดยกำหนดค่าพารามิเตอร์มาตรฐานที่ใช้ในการทดลองเหมือนกับการทดลองที่ 3 แต่กำหนดให้ชั้นข้อมูลที่คุณภาพดีที่สุดของข้อมูลมัลติมีเดียอยู่ที่ 320 kbps



ภาพประกอบ 24 กราฟ Throughput ของโพรโทคอล MeERA และ TCP

ผลการทดลองที่ได้ แสดงให้เห็นว่าโพรโทคอล MeERA ไม่สามารถตอบสนองต่อความต้องการในการใช้งานแบนด์วิดท์เพื่อส่งผ่านข้อมูลมัลติมีเดียให้ได้ตามมาตรฐานที่ต้องการได้

ตาราง 10 เปรียบเทียบอัตราการใช้แบนด์วิดท์เฉลี่ยระหว่าง TCP และ MeERA

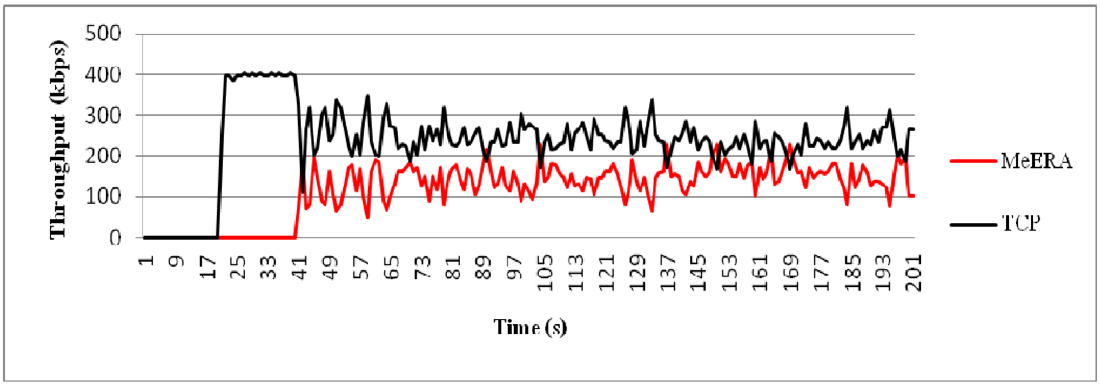
โพรโทคอล	**ค่าเฉลี่ย Throughput (Kbps)**
MeERA	153.39
TCP	239.56

จากผลการทดลองที่ได้หากพิจารณาในแง่ของคุณภาพมัลติมีเดียที่ได้ ย่อมแสดงให้เห็นว่าการปฏิบัติตามกฎความเป็นมิตรต่อทีซีพีเป็นอุปสรรคที่ทำให้มัลติคาสท์โพรโทคอลไม่สามารถตอบสนองต่อแบนด์วิดท์ที่ต้องการได้ หรือได้ในคุณภาพที่ต่ำกว่ามาตรฐานที่ต้องการ ซึ่งเมื่อพิจารณาจากตาราง 10 จะพบว่า แบนด์วิดท์ที่ข้อมูลมัลติมีเดียต้องการอยู่ที่ 320 kbps แต่โพรโทคอล MeERA

สามารถใช้ได้เพียง 153.39 kbps เนื่องจากการปฏิบัติตามกฎความเป็นมิตรต่อทีซีพีนั่นเอง

4.5 การทดลองที่ 5 การทดสอบเพื่อวิเคราะห์ประเด็น Responsiveness vs. Smoothness

ในการทดลองนี้จะทำการทดสอบมัลติคาสท์โพรโทคอล เพื่อพิจารณาถึงความราบเรียบ (Smoothness) ในการส่งผ่านข้อมูล โดยกำหนดค่าพารามิเตอร์มาตรฐานที่ใช้ในการทดลองนั้นให้คือ ขนาดของแพ็คเก็ตข้อมูลเท่ากับ 512 kb ขนาดชั้นข้อมูลเท่ากับ 20 kbps จำนวน 20 ชั้น มีแบนด์วิดท์ลิงค์คอขวดเท่ากับ 400 kbps แบนด์วิดท์หลังเราเตอร์ของโพรโทคอล MeERA เท่ากับ 10 Mbps ของทีซีพีเท่ากับ 10 Mbps และกำหนดเวลาในการทดลองกับ 200 วินาที โดยกำหนดให้ทีซีพีเริ่มทำงานที่วินาทีที่ 20 และ ให้โพรโทคอล MeERA เริ่มทำงานที่วินาทีที่ 40 เพื่อพิจารณาถึงผลกระทบที่เกิดขึ้นระหว่างโพรโทคอล MeERA และทีซีพี



ภาพประกอบ 25 กราฟ Throughput ของโพรโทคอล MeERA และ TCP

จากผลการทดลองที่ได้หากพิจารณาในแง่ของความราบเรียบในการส่งข้อมูล คือ MeERA มีการส่งผ่านข้อมูลที่ไม่ราบเรียบมากนัก เนื่องจากระหว่างการส่งข้อมูล MeERA พยายามประพฤติตัวให้อยู่ภายใต้กฎความเป็นมิตรต่อทีซีพี ดังนั้นอัตราการส่งผ่านข้อมูลของ MeERA ก็จะแปรผันตามอัตราการส่งผ่านข้อมูลของโพรโทคอลทีซีพีที่มีอยู่ในเครือข่ายด้วย ซึ่งจะสังเกตได้จากกราฟ Throughput ที่แสดงให้เห็นถึงการขึ้นลงตลอดเวลาของแบนด์วิดท์ที่ MeERA ใช้

ตาราง 11 แสดงการใช้แบนด์วิดท์ของ MeERA ในช่วงเวลาต่างๆ

เวลา (วินาทีที่) / ดัชนีการใช้งาน	58	60	63	68	72	75
แบนด์วิดท์ที่ MeERA ได้รับ (kbps)	98	188	90	173	102	180.20
แบนด์วิดท์ที่ TCP ได้รับ (kbps)	348.80	251.71	264.96	251.71	167.80	219.80
แบนด์วิดท์ที่ไม่ถูกใช้	1.90	0.80	3.90	0.80	2.80	0

เมื่อพิจารณาจากตาราง 11 จะเห็นถึงการขึ้นลงตลอดเวลาของแบนด์วิดท์ที่ MeERA ได้รับ โดยวินาทีที่ 58 นั้น MeERA ได้รับแบนด์วิดท์ที่ 98 kbps แล้ว วินาทีที่ 60 MeERA ได้เพิ่มอัตราการได้รับแบนด์วิดท์ขึ้นเป็น 188 kbps หลังจากนั้นก็ลดอัตราการได้รับแบนด์วิดท์ลงเหลือ 90 kbps ในวินาทีที่ 63 เป็นเหตุให้คุณภาพของมัลติมีเดียที่ผู้รับ ได้รับมีปัญหาด้วย ดังที่ตาราง 12

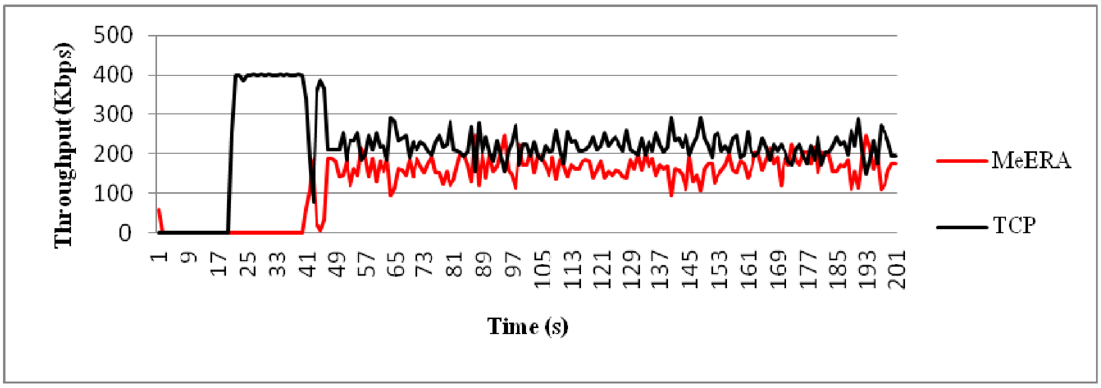
ตาราง 12 แสดงการใช้แบนด์วิดท์ของ MeERA ทีมีผลต่อคุณภาพของมัลติมีเดีย

วินาทีที่ 58	วินาทีที่ 60	วินาทีที่ 63	วินาทีที่ 68	วินาทีที่ 72

จากตาราง 12 แสดงให้เห็นว่าในอัตราบิต (แบนด์วิดท์) ที่ต่างกันคุณภาพของมัลติมีเดียที่ผู้รับได้รับก็จะต่างกันด้วย ข้อมูลมัลติมีเดียนั้นต้องการการส่งผ่านข้อมูลที่ราบเรียบ หากเกิดการแกว่งของข้อมูลขึ้นแล้ว คุณภาพที่ได้ก็จะไม่สามารถตอบสนองต่อผู้ใช้งานได้ตามมาตรฐานที่ต้องการ

4.6 การทดลองที่ 6 การทดสอบเพื่อวิเคราะห์ประเด็น Bandwidth Consumption

การทดลองนี้มีวัตถุประสงค์เพื่อทดสอบว่า เมื่อ MeERA สามารถเพิ่มอัตราการใช้แบนด์วิดท์ได้มากขึ้นแล้วจะส่งผลกระทบต่อคุณภาพของมัลติมีเดียหรือไม่ โดยกำหนดค่าพารามิเตอร์มาตรฐานที่ใช้ในการทดลองนั้นให้คือ ขนาดของแพ็คเก็ตข้อมูลเท่ากับ 512 kb ขนาดชั้นข้อมูลเท่ากับ 10 kbps จำนวน 40 ชั้น มีแบนด์วิดท์ลิงค์คอขวดเท่ากับ 400 kbps แบนด์วิดท์หลังเราเตอร์ของโพรโทคอล MeERA เท่ากับ 10 Mbps ของทีซีพีเท่ากับ 10 Mbps และกำหนดเวลาในการทดลองเท่ากับ 200 วินาที



ภาพประกอบ 26 แสดงค่า Throughput ของ MeERA และ TCP

จากภาพประกอบ 26 แสดงให้เห็นว่า เมื่อทำการลดขนาดชั้นข้อมูลลงมาที่ 10 kbps ปรากฏว่าโพรโทคอล MeERA สามารถตอบสนองต่อการตอบรับแบนด์วิดท์ได้ดีขึ้น ใกล้เคียงกับแบนด์วิดท์ที่ ทีซีพีใช้อยู่ ทำให้ลดช่องว่างของแบนด์วิดท์ที่ไม่ได้ใช้งานลง ซึ่งหากเปรียบเทียบกับคุณภาพของข้อมูลมัลติมีเดียที่ทำการส่งไป จะเห็นได้ว่าการปรับลดขนาดชั้นข้อมูลลงจะทำให้โพรโทคอล MeERA สามารถนำแบนด์วิดท์ที่เหลืออยู่มาใช้ได้มากขึ้น อันส่งผลให้คุณภาพของข้อมูลมัลติมีเดียที่ผู้รับได้มีคุณภาพมากยิ่งขึ้น ซึ่งสามารถพิจารณาจากข้อมูลผลการทดลองที่ได้ตามตารางที่ 13

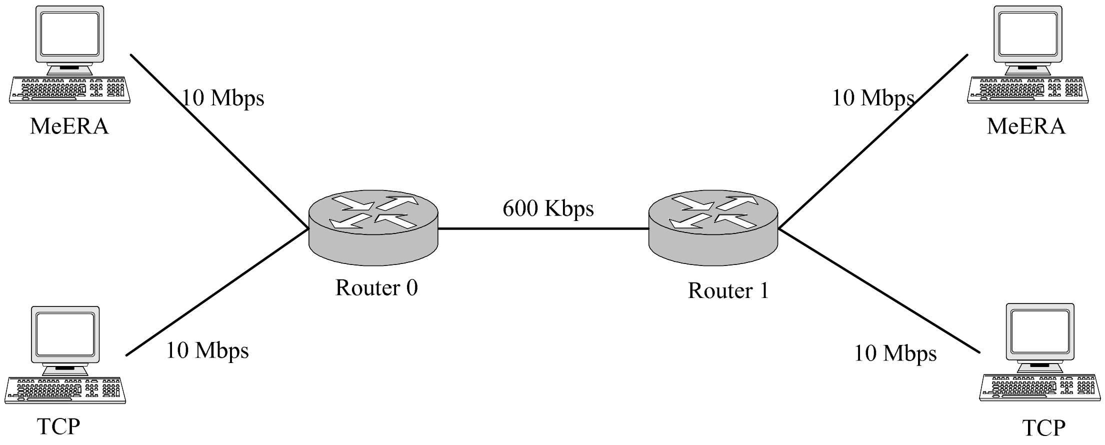
ตาราง 13 แสดงการใช้แบนด์วิดท์ของ MeERA ในช่วงเวลาต่างๆ

เวลา (วินาทีที่) / ดัชนีการใช้งาน	59	80	100	120	169	180	195
แบนด์วิดท์ที่ MeERA ได้รับ (kbps)	188.41	118.78	172.03	159.74	192.51	151.55	163.84
แบนด์วิดท์ที่ TCP ได้รับ (kbps)	210.96	281.22	225.21	211.96	207.45	242.88	234.04
แบนด์วิดท์ที่ไม่ถูกใช้	0.62	0	2.49	28.28	0.04	18.56	2.11

จากตาราง 13 เมื่อพิจารณาอัตราการได้รับแบนด์วิดท์ที่เพิ่มขึ้นของ MeERA เทียบกับมาตรฐาน Audio (mp3) จะพบว่า แม้โพรโทคอล MeERA จะสามารถปรับอัตราการรับแบนด์วิดท์ได้สูงขึ้นแต่ก็ไม่ได้ทำให้คุณภาพของมัลติมีเดียเพิ่มขึ้นตามไปด้วย เพราะอัตราบิตของมาตรฐาน Audio (mp3) จะอยู่ในช่วง 128-160 kbps และมาตรฐานของ DAB อยู่ที่ >192 kbps ดังนั้น นอกจากจะไม่ส่งผลดีต่อมัลติมีเดียแล้ว การปรับอัตราของโพรโทคอล MeERA กลับส่งผลเสียต่อมัลติมีเดียในแง่ของความราบเรียบด้วย ดังนั้นในสถานการณ์นี้การที่โพรโทคอล MeERA เพิกเฉยต่อแบนด์วิดท์ที่ว่างอยู่อาจส่งผลดีมากกว่า

4.7 การทดลองที่ 7 การทดสอบเพื่อวิเคราะห์ประเด็น Layer Granularity

จากความสนใจในเรื่องปัญหาของขนาดของชั้นข้อมูลที่มีผลต่อการส่งผ่านข้อมูลมัลติมีเดีย ทำให้งานวิจัยนี้มีความสนใจถึงข้อแตกต่างของการส่งผ่านข้อมูลมัลติมีเดียที่มีการแบ่งขนาดชั้นข้อมูลแบบเท่ากันหมด (Equal) และแบบที่มีการแบ่งชั้นข้อมูลตามระดับคุณภาพของสัญญาณ โดยในการทดลองนี้ได้แบ่งขนาดชั้นข้อมูลตามระดับคุณภาพของเสียง ดังตาราง 2 ในบทที่ 2 มีระดับคุณภาพดังนี้ AM quality ที่ 32 kbps, FM quality ที่ 96 kbps, Bass quality ที่128–160 kbps, DAB (Digital Audio Broadcasting) ที่ 192 kbps, CD Quality ที่ 224–320 kbps

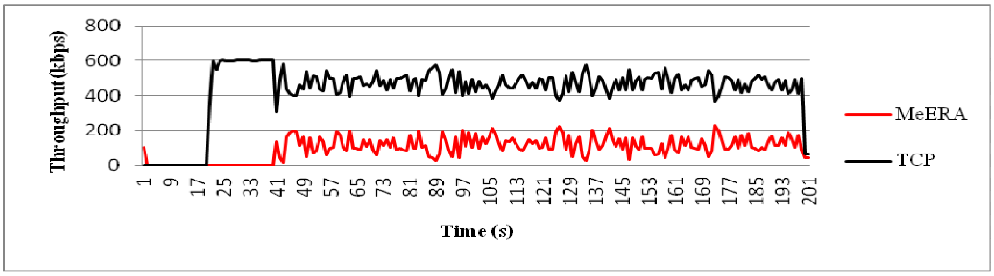


ภาพประกอบ 27 โทพอโลยีที่ใช้ในการทดลองที่ 8

การทดลองนี้มีการกำหนดค่าพารามิเตอร์ต่างๆดังนี้ ขนาดแพ็คเก็ตข้อมูลเท่ากับ 512 kb ขนาดของแต่ละชั้นข้อมูลเท่ากับ 8, 24, 64, 32, 64, 32 ,96, 64, 32, 64 Kbps ตามลำดับ แบนด์วิดท์การส่งข้อมูลหลังเราเตอร์ของทีซีพีและ MeERA เท่ากับ 10 Mbps มีแบนด์วิดท์ที่ลิงค์คอขวดเท่ากับ 600 kbps ใช้เวลาในการทดลองเท่ากับ 200 วินาที กำหนดให้ชั้นข้อมูลที่คุณภาพดีที่สุดของข้อมูลเสียงอยู่ที่ 320 kbps ใช้วิธีการบอกรับชั้นข้อมูลแบบสะสม (cumulative)

ตาราง 14 แสดงขนาดชั้นข้อมูลสะสมที่เพิ่มขึ้นตามระดับคุณภาพของสัญญาณ

เลเยอร์สะสม (kbps)	8	32	96	128	192	224	320
ขนาดเลเยอร์ (kbps)	8	24	64	32	64	32	96



จากภาพประกอบ 28 แสดงให้เห็นว่า เมื่อทำการกำหนดขนาดของชั้นข้อมูลตามระดับ

คุณภาพของสัญญาณเสียงแล้วปรากฏว่าโพรโทคอล MeERA ไม่สามารถเข้าใช้งานแบนด์วิดท์ได้อย่างมีประสิทธิภาพได้เหมือนกับการกำหนดขนาดชั้นข้อมูลแบบเท่ากันหมด ทำให้เกิดช่องว่างของแบนด์วิดท์ที่ไม่ถูกใช้งานจำนวนมาก อันเนื่องมาจากขนาดของชั้นข้อมูลที่มีขนาดเล็กใหญ่ต่างกันนั่นเอง

ตาราง 15 แสดงค่า Throughput ของโพรโทคอล ERA และ TCP

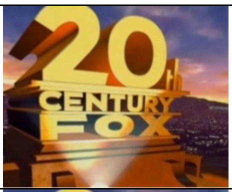
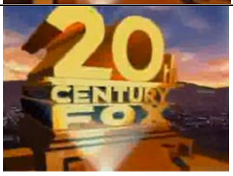
โพรโทคอล	**ค่า Throughput เฉลี่ย (Kbps)**
MeERA	121.48
TCP	467.84

จากตาราง 15 จะเห็นแบนด์วิดท์ที่โพรโทคอล MeERA ได้รับนั้น เมื่อเทียบกับระดับคุณภาพที่อ้างอิงนั้น ผู้รับจะได้รับข้อมูลเสียงในระดับ FM Quality เท่านั้น ทั้งๆที่ควรจะได้รับแบนด์วิดท์ในระดับ CD Quality ที่ 224–320 kbps ตามลิงค์คอขวดที่แบ่งกันกับโพรโทคอลทีซีพีโพรโทคอลละ 300 kbps ตามกฎความเป็นมิตรต่อทีซีพี

4.8 การทดลองที่ 8 การทดสอบเพื่อวิเคราะห์ประเด็น Layer Granularity ที่มีการเข้ารหัสไฟล์มัลติมีเดียที่แตกต่างกัน

นอกเหนือจากความสนใจในเรื่องปัญหาของขนาดของชั้นข้อมูลที่มีผลต่อการส่งผ่านข้อมูลมัลติมีเดียแล้ว งานวิจัยนี้ยังมีความสนใจถึงข้อแตกต่างของการส่งผ่านข้อมูลมัลติมีเดียที่มีวิธีการเข้ารหัสไฟล์ (Codec) ที่แตกต่างกัน จึงได้ทำการทดสอบการส่งผ่านข้อมูลมัลติมีเดีย ที่มีวิธีการเข้ารหัสไฟล์ที่แตกต่างกัน โดยได้ทำการเลือกไฟล์มัลติมีเดีย ที่มีขนาด (Dimension) 720 x 480 มาทำการทดสอบโดยเลือกวิธีการเข้ารหัสไฟล์มา 2 วิธี คือ วิธีการแบบ MPEG2 และ H.264 มาทำการเปรียบเทียบถึงความแตกต่าง โดยทำการวัดความแตกต่างทางด้านคุณภาพในอัตราการบิต (bit rate) ตามขนาดของแบนด์วิดท์ที่โพรโทคอล MeERA ได้รับจากการทดลองที่ 5 ตั้งแต่อัตราบิตสูงสุดคือ 229.37 kbps และอัตราบิตต่ำสุดคือ 49.15 kbps

ตาราง 16 เปรียบเทียบคุณภาพมัลติมีเดียที่มีการเข้ารหัสไฟล์ต่างกันที่อัตราบิตเท่ากัน

แบนด์วิดท์(kbps) / **Encoder**	**320 (Kbps)**	**229.37(Kbps)**	**49.15(Kbps)**
MPEG II			
H.264			

จากภาพในตาราง 16 แสดงให้เห็นว่า นอกจากปัญหาที่ได้กล่าวในเรื่องของการหาขนาดชั้นข้อมูลที่เหมาะสมสำหรับการส่งมัลติมีเดียแล้ว ยังมีปัญหาในเรื่องความเหมาะสมในการเลือกวิธีการเข้ารหัสไฟล์มาใช้อีกด้วย ซึ่งจากการทดสอบแม้ว่าไฟล์ทั้งสองจะถูกส่งผ่านในอัตราบิตที่มีขนาดเท่ากันแต่คุณภาพที่ได้กลับมีความแตกต่างกัน การส่งมัลติมีเดียของโพรโทคอล MR-MCC ที่ได้รับการพัฒนามาตลอดนั้น ควรคำนึงถึงวิธีการเข้ารหัสไฟล์ข้อมูลด้วย

Mahasarakham University

บทที่ 5

สรุปผล และข้อเสนอแนะ

5.1 ข้อสรุปของงานวิจัย

จากการศึกษาและทำการทดลองเกี่ยวกับประเด็นปัญหาของการวิจัยของโพรโทคอล MR-MCC ซึ่งเป็นการเผยให้เห็นถึงข้อด้อยของการปฏิบัติตามกฎของ IETF [3] เมื่อโพรโทคอล MR-MCC ต้องให้บริการมัลติมีเดีย โดยสามารถสรุปผลลัพธ์จากการศึกษาได้ดังนี้

5.1.1 Loss tolerant / Delay sensitive vs. Delay tolerant / Loss sensitive

จากการศึกษาพบว่า เนื่องมาจากมัลติคาสท์มีคุณสมบัติการส่งข้อมูลที่แตกต่างจากทีซีพีโดยสิ้นเชิง การที่โพโทคอล MR-MCC พยายามทำตัวให้เหมือนกับทีซีพีเมื่อให้บริการมัลติมีเดีย ทำให้เกิดปัญหาความล่าช้าจนล้มเหลวในการส่งข้อมูลที่มีคุณภาพได้

5.1.2 Responsiveness vs. Smoothness

ในประเด็นนี้จากศึกษาพบว่า การพยายามตอบสนองต่อความคับคั่งอย่างรวดเร็วของโพโทคอล MR-MCC นั่นได้ส่งผลอย่างยิ่งต่อคุณภาพของมัลติมีเดียที่ผู้รับได้รับ เพราะการตอบสนองต่อความคับคั่งอย่างรวดเร็วหมายถึง การปรับอัตราการรับส่งข้อมูลอย่างรวดเร็วซึ่งส่งผลโดยตรงต่อความราบเรียบในการส่งข้อมูล เป็นเหตุให้คุณภาพของมัลติมีเดียฝั่งผู้รับมีปัญหาได้

5.1.3 Bandwidth Consumption

จากศึกษาพบว่าโพรโทคอล MR-MCC มีข้อด้อยในการที่ไม่สามารถเข้าใช้แบนด์วิดท์ที่มีอยู่ได้อย่างเต็มที่ เพราะต้องพยายามปฏิบัติตามกฎความเป็นมิตรต่อทีซีพีของ IETF และในบางสถานการณ์การเพิกเฉยต่อแบนด์วิดท์ที่ว่างอยู่อาจส่งผลดีมากกว่าการเข้าใช้แบนด์วิดท์อย่างรวดเร็วโดยไม่ทำให้ได้คุณภาพของมัลติมีเดียที่ดีขึ้น

5.1.4 Layer Granularity

ในประเด็นปัญหานี้ จากการศึกษาได้แสดงให้เห็นว่า ขนาดของชั้นข้อมูลนั้นส่งผลโดยตรงคุณภาพของมัลติมีเดีย และจากผลการทดลอง ยังแสดงให้เห็นว่าปัญหาในเรื่องความเหมาะสมในการเลือกวิธีการเข้ารหัสไฟล์ (CODEC) มาใช้ด้วย

สรุปผลการวิจัย พบว่าแม้โพรโทคอล MR-MCC จะพยายามแก้ปัญหาเรื่องความเป็นมิตรต่อทีซีพีทำให้มีความเป็นมิตรต่อทีซีพีตามกฎของ IETF ก็ตาม แต่ประเด็นปัญหาที่งานวิจัยนี้ค้นพบก็คือ การใช้งานแบนด์วิดท์ที่ไม่เหมาะสมในการส่งผ่านข้อมูลมัลติมีเดีย ซึ่งจากผลการทดลองเผยให้เห็นว่าโพรโทคอล MR-MCC ไม่สามารถ Subscribe Layer เพื่อรักษาระดับความราบเรียบของการส่งข้อมูล เป็นเหตุให้ข้อมูลมัลติมีเดียที่ผู้รับได้รับนั้นไม่มีคุณภาพเพียงพอต่อการรับชมหรือที่ร้ายแรงกว่านั้นอาจถึงขั้นการสื่อสารล้มเหลวได้ และแม้ว่าโพรโทคอล MR-MCC จะสามารถเพิ่มอัตราการใช้แบนด์วิดท์ของตนเองให้มากขึ้น แต่ก็ไม่ได้ส่งผลให้คุณภาพของมัลติมีเดียดีขึ้น นอกจากนี้ปัญหาในเรื่องวิธีการเข้ารหัสไฟล์ข้อมูลมัลติมีเดียที่เหมาะสม ว่าควรใช้วิธีการใดยังเป็นปัญหาที่ควรจะศึกษาเพิ่มเติมอีกด้วย

5.2 ข้อเสนอแนะ

งานวิจัยในอนาคตที่สามารถทำต่อไปได้ คือการใช้วิธีการที่ยอมให้โพรโทคอล MR-MCC สามารถได้รับแบนด์วิดท์มากกว่าทีซีพี ในบางช่วงเวลาเพื่อรักษาระดับความราบเรียบในการส่งข้อมูลแบบมัลติมีเดีย ซึ่งจะเป็นการแก้ปัญหาในเรื่องคุณภาพของมัลติมีเดียที่ผู้รับได้รับแต่วิธีการนี้อาจจะก่อให้เกิดปัญหาในเรื่องความเป็นมิตรต่อทีซีพี แต่ถ้าเมื่อคำนวณแบนด์วิดท์ในช่วงเวลาในการส่งข้อมูลโดยรวมแล้วสามารถทาให้ทีซีพีได้รับแบนด์วิดท์มากกว่าโพรโทคอล MR-MCC ก็อาจจะยอมรับได้ว่าแนวทางใหม่นี้ไม่ก่อปัญหาความเป็นมิตรกับทีซีพีตามกฎของ IETF ก็เป็นได้

เอกสารอ้างอิง

เอกสารอ้างอิง

[1] Deering S. Host Extensions for IP Multicasting. IETF, RFC1112, August 1989.

[2] Puangpronpitag S. Design and Performance Evaluation of Multicast Congestion Control for the Internet. Ph.D. thesis. Leeds: University of Leeds; 2003.

[3] Handley M, Sally F. TCP Friendly Rate Control (TFRC): Protocol Specification. IETF, RFC 5348, January 2003.

[4] Puangpronpitag S. Multi-Rate Multicast Congestion Control by Explicit Rate Adjustment and Multicast-encouraging TCP-friendliness. Proceedings of the 15th IEEE International Conference on Networks (ICON2007); 19-21 November 2007; Adelaide, Australia. pp. 101-106.

[5] Network Simulator -- ns version 2. [computer program]. Version 2.30: Virtual InterNetwork Testbed; 2003.

[6] Multicast Deployment Made Easy. [online]. 1999; [cited 2 December 2008]; http://www.cisco.com.

[7] Adams A, Nicholas J, Siadak W. Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised). IETF, RFC 2362, January 2005.

[8] Waitzman D, Partridge C, Deering S. Distance Vector Multicast Routing Protocol. IETF, RFC 1075, November 1988.

[9] Bhattacharyya S. An Overview of Source-Specific Multicast (SSM). IETF, RFC 3569, July 2003.

[10] Sharma, Wei L. Protocol Independent Multicast-Sparse Mode (PIM-SM) : Protocol Specification. IETF, RFC 4609, June 1998.

[11] Kurose, Roos. Computer Networking : A Top-Down Approach Featuring the Internet. 4th ed. Massachusetts University.; 2007.

[12] Brandenburg, Karlheinz, Bosi M. Overview of MPEG Audio: Current and Future Standards for Low Bit-Rate Audio Coding. J Audio Eng Soc 1997; 45[1/2]: 4-21.

[13] Bettaharh. Tutorial on Multicast Video Streaming Techniques. Proceeding of 3rd International Conference: Sciences of Electronic, Technologies of Information and elecommunications (SETIT2005): 27-31 March 2005; TUNISIA.

[14] Luby M, Gemmell J, other. Layered Coding Transport (LCT) Building Block. IETF, RFC 3451, December 2002.

[15] Tanenbaum. Computer Network. 2nd ed. Netherlands: Vrije Universiteit.; 1989.

[16] DeLucia D, Obraczka K. Congestion Control Performance of a Reliable Multicast Protocol. Proceeding of 6th IEEE International Conference on Network Protocols. 13 - 16 October 1998; Austin, Texas, USA.

[17] Rhee, Balaguru N, Rouskas G. MTCP: Scalable TCP-like Congestion Control for Reliable Multicast. Proceedings of the Conference on Computer Communications (IEEE Infocom); Mar 1999; New York. pp. 1265-1273.

[18] Rizzo. PGMCC: A TCP-friendly Single-rate Multicast Congestion Control Scheme. Proceeding of ACM SIGCOMM; 28 August – 1 September 2000; Stockholm Sweden. pp. 17-28.

[19] Joerg W, Handley M. TCP-friendly Multicast Congestion Control (TFMCC) : Protocol Specification. IETF, RFC 4654, August 2006.

[20] Legout, Biersack. PLM: Fast Convergence for Cumulative Layered Multicast Transmission. Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems. 18-21 June 2000; Santa Clara California USA.pp. 13-22.

[21] McCanne S, Jacobson V, Vetterli M. Receiver-driven Layered Multicast. Proceedings of the ACM SIGCOMM '96; 26-30 August 1996; Stanford, CA, USA. pp. 117-130.

[22] Lorenzo V, Rizzo L, Crowcroft J. TCP-like Congestion Control for Layered Multicast Data Transfer. Proceedings of Proceedings IEEE INFOCOM '98; March 29 - April 2 1998; San Francisco CA USA. pp. 996-1003.

[23] Byers J, Frumin M, Horn G. FLID-DL:Congestion Control for Layer Multicast. Proceedings of ACM NGC; 8-10 November 2000; Palo Alto USA. pp. 71-82.

[24] Jain R. The Art of Computer Systems Performance Analysis. John Wiley & Sons Inc.; 1991.

[25] Law, Kelton. Simulation Modelling and Analysis. 2 nd ed. S.l: McGraw-Hill.; 1991.

[26] Keshav, Srinivasan. Congestion Control in Computer Networks. EECS Department UC Berkeley, CSD-91-649, 27 September 1991.

[27] Bajaj S, Breslau L, Estrin D, et al. Improving Simulation for Network Research. University of Southern California, 99-702, 4 March 1999.

[28] Pawlikowski, Jeong H-DJ, Lee J-SR. On Credibility of Simulation Studies of Telecommunication networks. IEEE Communications Magazine. April 2002. pp. 132-139.

[29] Floyd S. Simulation is Crucial. Appear as a sidebar in IEEE Spectrum. January 2001.

[30] Breslau L, Deborah, Estrin D, et al. Advances in Network Simulation. IEEE Computer Journal [online] May 2000; 33: 59-67. [cited 16 April 2008]; http://ieeexplore.ieee.org

[31] Mankin A, Paxson V, other. IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols. IETF, RFC 2357, June 1998.

[32] Mahdavi J, Floyd S. TCP-Friendly Unicast Rate-Based Flow Control. [online] January 1997; [cited 21 April 2008]; http://www.psc.edu/networking/papers/tcp_friendly.html.

[33] Luby M, other. "Wave and Equation Based Rate Control Using Multicast Round Trip Time". Proceedings of the ACM SIGCOMM 2002; 19-23 August 2002; Pittsburgh PA USA. pp. 191-204.

[34] Puangpronpitag S, Boyle RD, Djemame K. Explicit Rate Adjustment: an Efficient Congestion Control Protocol for Layered Multicast. Proceedings of the 11th IEEE International Conference on Networks (ICON 03); 28 September – 1 October 2003; Sydney Australia.pp. 179-184.

[35] Huston G. TCP Performance. The Internet Protocol Journal [online] June 2000; 3[2]: 2-24. [cited 25 May 2008]; http://www.cisco.com/web/about/ac123/ac147/ac174/ac196 /about_cisco_ipj_archive_arti cle09186a00800c8901.pdf

[36] พชร พฤกษะศรี. การศึกษาความเป็นมิตรต่อทีซีพีของโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราส่งในเครือข่ายการสื่อสารมัลติคาสท์. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต. ขอนแก่น : มหาวิทยาลัยขอนแก่น; 2548.

[37] Crowcroft J. "TCP-friendliness Considered Unfriendly". [online]. [cited 10 December 2008]; http://www.acu.rl.ac.uk/msn2001.

[38] สุชาติ คุ้มมณี. ปรับปรุงความเร็วและความเป็นมิตรต่อทีซีพีของโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราส่งในเครือข่ายการสื่อสารมัลติคาสท์. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต. ขอนแก่น: มหาวิทยาลัยขอนแก่น; 2550.

[39] Keshav. REAL: a Network Simulator. University of California Berkeley; 1998.

[40] Baumann M. YATS User's and Programmer's Manual for version 0.3. Dresden University of Technology; September 1997.

[41] Global Mobile Information System Simulator. [computer program]. Version 2.0: University of California, Los Angeles; 2004.

[42] Boyce JM, Gaglianello RD. Packet loss effects on MPEG video sent over the public Internet. Proceedings of the sixth ACM international conference on Multimedia; 13 - 16 September 1998; Bristol, UK. pp. 189-190.

[43] กัลยา วานิชย์บัญชา. การวิเคราะห์สถิติ: เพื่อการตัดสินใจ. กรุงเทพฯ: จุฬาลงกรณ์มหาวิทยาลัย; 2539.

[44] Nam: Network Animator. [computer program]. Version 1.11: VINT project; 2001.

ภาคผนวก

ภาคผนวก ก

เครื่องมือจำลองการทำงานเครือข่าย (Network Simulator)

1. การติดตั้งโปรแกรม Network Simulator รุ่น 2.30 บนระบบปฏิบัติการลินุกซ์

2. การติดตั้งโพรโทคอล MeERA

3. MeERA Script

Mahasarakham University

1. การติดตั้งโปรแกรม Network Simulator รุ่น 2.30 บนระบบปฏิบัติการลินุกซ์

1. ดาวน์โหลดตัวติดตั้งโปรแกรมชื่อ ns-allinone-2.30.tar.gz (รุ่น All in One) จากเว็บไซต์ http://www.isi.edu/nsnam/dist/ns-allinone-2.30.tar.gz

2. ทำการ extract ไฟล์ ns-allinone-2.30.tar.gz ด้วยคำสั่ง *gunzip ns-allinone-2.30.tar.gz | tar xvf –*

3. ทำการเปลี่ยนโหมด ของไฟล์ เชลล์สคริปท์ เป็น 755 ทุกไฟล์

4. ทำการติดตั้งโปรแกรม โดยใช้คำสั่ง *./install*

5. หลังจากทำการติดตั้งโปรแกรมแล้ว ทำการรันคำสั่ง *make* เพื่อสร้างโปรแกรม

6. เมื่อสิ้นสุดการติดตั้งโปรแกรมจะมีการแสดง path ที่ต้องทำการกำหนดเอง ให้ทำการกำหนด path ตามที่โปรแกรมระบุ

7. เมื่อกำหนด path เรียบร้อยแล้ว ให้ไปที่ directory tcl/ex

8. ทดสอบการทำงานของตัวจำลอง โดยการเรียกสคริปท์ไฟล์ โดยคำสั่ง *ns simple-plm.tcl*

9. หากโปรแกรมทำงานได้ เป็นอันเสร็จสิ้นการติดตั้งโปรแกรม

2. การติดตั้งโพรโทคอล MeERA

1. คัดลอกไฟล์ plm.tcl, plm-ns.tcl, plm-topo.tcl ที่ได้จากผู้พัฒนาโพรโทคอล MeERA ไปไว้ใน ..\ns-allinone-2.30\ns-2.30\tcl\plm

2. ทำการ recompile NS2 ใหม่ ด้วยคำสั่ง ./install

3. ทำการ validate NS2 ใหม่ ด้วยคำสั่ง ./validate

4. ทดสอบการทำงานของตัวจำลอง โดยการเรียกสคริปท์ไฟล์ โดยคำสั่ง *ns simple-plm.tcl*

5. หากโปรแกรมทำงานได้ เป็นอันเสร็จสิ้นการติดตั้งโปรแกรม

3. MeERA Script

3.1 plm.tcl

*************************plm.tcl***

```
Class PLM

PLM instproc init {levels rai n_id} {
    $self next
    $self instvar PP_estimate R_pp R_tcp
    $self instvar time_estimate check_estimate node_id
    $self instvar hh_npkts hh_nloss
    set PP_estimate {}
    set R_pp -1
    set R_tcp -1
    set hh_npkts 0
    set hh_nloss 0
    set time_estimate 0
    set check_estimate $rai
    set node_id $n_id
            $self instvar debug_ env_ maxlevel_
            set debug_ 2
            set env_ [lindex [split [$self info class] /] 1]
            set maxlevel_ $levels
            global plm_debug_flag
            if [info exists plm_debug_flag] {
                        set debug_ $plm_debug_flag }
            $self instvar subscription_
            # we number the subscription level starting at 1.
            # level 0 means no groups are subscribed to.
            $self instvar layer_ layers_
```

```
            set i 1
            while { $i <= $maxlevel_ } {
                        set layer_($i) [$self create-layer [expr $i - 1]]
                        lappend layers_ $layer_($i)
                        incr i }
            set subscription_ 0
            $self add-layer }
PLM instproc make_estimate {PP_value} {
   $self instvar PP_estimate R_pp ns_ time_estimate
   $self instvar check_estimate debug_
   global PP_estimation_length mRTT packetSize nukDebug
   #----- LogRD paramiter -------
   global k m
   #-----------------------------
   $self instvar R_tcp hh_npkts hh_nloss
   #Add PP_value to the list of single PP estimates PP_estimate
   lappend PP_estimate $PP_value
   #Drop layer(s) if the PP_value is lower than the current subscription level
   $self stability-drop $PP_value
   #time_estimate is the minimum period of time during which we collect PP_value
   #to make the global estimate PP_estimate_value
   set ns_time [$ns_ now]
   # if it is the first time PP_arrived, wait till RAI
   if {$time_estimate==0} {set time_estimate [expr $ns_time + $check_estimate]  }
   if {$debug_>=3} {trace_annotate "[$self node]: check: $check_estimate $PP_estimate , nb: 
[llength $PP_estimate]" }
    # For every RAI, do Rate Adaptation Process
   if {$time_estimate <= $ns_time} {
     if {$nukDebug >= 1} {
```

```
    puts stderr "* IN make_estimate: every RAI, rate adaptation process:"
    puts stderr "    Time Estimate=$time_estimate Now=$ns_time " }
# ========================== Find Rpp ================================
# By PP bunch concept, need PP at least PP_estimation_length to find Rpp
  if {([llength $PP_estimate] >= $PP_estimation_length)} {
    set R_pp [lindex [lsort -real $PP_estimate] 0]
    if {$debug_>=3} {trace_annotate "[$self node]: check: $check_estimate PP estim:
$PP_estimate, value: $R_pp" }
    if {$nukDebug >= 1} {  puts stderr "* IN make_estimate: Rpp is min of R'pp:"
      puts stderr "    List of R'pp = [set PP_estimate]"
      # ?? [set properties] vs. $properties
      puts stderr "    Rpp = $R_pp" }
    if {$debug_>=2} {trace_annotate [expr round($R_pp)] }
    set PP_estimate {}
  } else {  # if not enough PP to calculate Rpp
    if {$nukDebug >= 1} {
      puts stderr "* IN make_estimate: Not enough PP for calculate Rpp"
      puts stderr "    PP_estimation_length=$PP_estimation_length"}
    set R_pp -1
    set PP_estimate {} }
  # ================== Calculate Rtcp ========================
  ###?? need more than 10 packets received to calculate R_tcp
  ###?? puts stderr "*mPkt* nPkts = $npkts, Rtcp need 10 pkt to calculate"
  puts stderr "    hh_npkts=$hh_npkts"
  puts stderr "    hh_nloss=$hh_nloss"
  set npkts [expr [$self plm_pkts] - $hh_npkts]
  set nloss [expr [$self plm_loss] - $hh_nloss]
  puts stderr "    npkts=$npkts"
  puts stderr "    nloss=$nloss"
```

```
set loss [expr double($nloss) / ($nloss + $npkts)]
# keep the history
set hh_npkts $npkts
set hh_nloss $nloss
# ?? keep hhloss  and use weight.
if {$nukDebug >= 1} {  puts stderr "* IN make_estimate: Calculate Rtcp"
  puts stderr "      PLR=$loss" }
if { $loss > 0 } {
   # Eq. 1 draft-ietf-rmt-bb-tfmcc-01
   set tfm [expr ($packetSize*8) / ($mRTT * ((sqrt(2*$loss/3)) + (12*sqrt(3*$loss/8) * $loss *
(1+32*$loss*$loss))))]
   if {$nukDebug >= 1} {
      # Mahadavi Equation, mentioned in Eq. 1 [TBP97]
      set tbp [expr ($packetSize*8*1.22) / ($mRTT * sqrt($loss))]
      puts stderr "     Rtcp of Mahadavi=$tbp"
      puts stderr "     Rtcp of TFRMCC = $tfm"}
   set R_tcp $tfm
 } else { set R_tcp -1 }
 # ============== Rate adjustment using Rpp and Rtcp ==============
        #$self choose_layer [LogRD $k $m $R_pp "multicast"]
        #puts stderr "testing"
 if {($R_pp == -1) && ($R_tcp == -1)} {
   if {$nukDebug >= 1} { puts stderr "No Rate Adaptation as both Rtcp and Rpp = -1" }
 } elseif {($R_pp != -1) && ($R_tcp == -1)} { $self choose_layer $R_pp
                        #$self choose_layer [LogRD $k $m $R_pp "multicast"]
 } elseif {($R_pp == -1) && ($R_tcp != -1)} {$self choose_layer $R_tcp
                } elseif {($R_pp != -1) && ($R_tcp != -1)} {  $self choose_layer [Min $R_pp
$R_tcp]
                        #$self choose_layer [LogRD $k $m $R_pp "multicast"]}
```

```
            set time_estimate [expr $ns_time + $check_estimate]}}
#stability_drop drops layer(s) if a PP_value is lower than the current subscription level.
PLM instproc stability-drop {PP_value} {
    $self instvar subscription_ time_estimate PP_estimate
    $self instvar check_estimate ns_
    global rates_cum nukDebug
    set ns_time [$ns_ now]
    for {set i 0} {[lindex $rates_cum $i] < [expr round($PP_value)]} {incr i} {
            if {$i > [llength $rates_cum]} {
             puts stderr "ERROR: the no. of layers is not enough to cover available bandwidth!"
             break } }
     # To be conservative, subscription rate <= R'pp only
    if {[lindex $rates_cum $i] > [expr round($PP_value)]} {   incr i -1 }
    if {$nukDebug >= 2 } {
      puts stderr "**IN stability-drop() at $ns_time"
      puts stderr "   R'pp: $PP_value"
      puts stderr "   The Target Layer is layer no. $i"
      puts stderr "   R$i = [lindex $rates_cum $i]"  }
      # subscription level is higher than R'pp (over-subscribed)
    if {$subscription_ > $i} {
            for {set j $subscription_} {$i < $j} {incr j -1} {
              $self drop-layer}
            set PP_estimate {}
            set time_estimate [expr $ns_time + $check_estimate]
             if {$nukDebug >= 1 } {
         puts stderr "   next time_estimate=$time_estimate" } }}
#calculate the cumulated rates. (usefull for choose_layer)
proc calc_cum {rates} {
    set temp 0
```

```
    set rates_cum {}
    for {set i 0} {$i<[llength $rates]} {incr i} {
            set temp [expr $temp + [lindex $rates $i]]
            lappend rates_cum $temp }
    return $rates_cum }
#choose_layer chooses a layer according to the PP_estimate_value
#and the current subscription level.
PLM instproc choose_layer {R_target} {
    $self instvar subscription_
    global rates_cum nukDebug

     for {set i 0} {[lindex $rates_cum $i] < [expr round($R_target)]} {incr i} {
       if {$i > [llength $rates_cum]} {
         # NUK
         puts stderr "ERROR: the no. of layers is not enough to cover available bandwidth!"
         break } }
    # subscription rate <= Rtarget only
    if {[lindex $rates_cum $i] > [expr round($R_target)]} {
      incr i -1 }
    if {$nukDebug >= 1 } {
      puts stderr "*IN choose_layer()"
      puts stderr "   Rtarget: [expr round($R_target)]"
      puts stderr "   The Target Layer is layer no. $i"
      puts stderr "   R$i = [lindex $rates_cum $i]"  }
    if {$subscription_ < $i} {
            for {set j $subscription_} {$j < $i} {incr j} {
              $self add-layer}
    } elseif {$subscription_ > $i} {
            for {set j $subscription_} {$i < $j} {incr j -1} {
```

```
                  $self drop-layer}
    } elseif {$subscription_ == $i} {
              return }}
#                         log-loss {}
# Function: In case of loss, log-loss is called.
#          Do nothing as disable in ERA
# --------------------------------------------------------------------------
PLM instproc log-loss {} {
   global nukDebug
    if {$nukDebug >= 2} {
      puts stderr "NUK: ** I HAVE DISABLED LOG-LOSS! **"
     }  }
#                  ERA:drop-layer{}
PLM instproc drop-layer {} {
    $self instvar subscription_ layer_ node_id debug_
    set n $subscription_
    # if we have an active layer, drop it
    if { $n > 0 } {
              $self debug "DRP-LAYER $n"
              $layer_($n) leave-group
              incr n -1
              set subscription_ $n
              if {$debug_>=2} {
                  trace_annotate " [$self set node_id] : change layer $subscription_ "}}
    #rejoin the session after 30 seconds if drop all the layers
    if { $subscription_ == 0 } {
              set ns [Simulator instance]
              set rejoin_timer 30
              $ns at [expr [$ns now] + $rejoin_timer] "$self add-layer"
```

```
            if {$debug_>=2} {
               trace_annotate " Try to re-join the session after dropping all the layers "} }}
#              ERA:add-layer{}
PLM instproc add-layer {} {
   $self instvar maxlevel_ subscription_ layer_ node_id debug_
   set n $subscription_
   if { $n < $maxlevel_ } {
            $self debug "ADD-LAYER"
            incr n
            set subscription_ $n
            $layer_($n) join-group
            if {$debug_>=2} {
               trace_annotate " [$self set node_id] : change layer $subscription_ "} }}


#              ERA:era_loss{}
# return the no. of loss packets across all the groups of the given era
PLM instproc plm_loss {} {
            $self instvar layers_
            set loss 0
            foreach l $layers_ {
            incr loss [$l nlost]}
            return $loss }
#              ERA:era_pkts{}
# return the no. of packets received across all the groups of the given era
PLM instproc plm_pkts {} {
            $self instvar layers_
            set npkts 0
            foreach l $layers_ {incr npkts [$l npkts] }
```

```
        return $npkts}
#           ERA:debug{msg}
PLM instproc debug { msg } {$self instvar debug_ subscription_ ns_
        if {$debug_ <1} { return }
        set time [format %.05f [$ns_ now]]
        puts stderr "PLM: $time  layer $subscription_ $msg"}
#------------ LogRD ---------------------
proc LogRD {k m Rpp type} {
if {$type == "unicast"} {
     return [expr $k * [expr (1/($k+(1+log($m)))*$Rpp)]]
        } elseif {$type == "multicast"} { return [expr (1+log($m))/($k+(1+log($m)))*$Rpp]
        } else { puts stderr "can't cassify traffic type !!!!"
     return -1}}
########################### Class ERALayer ###################################
Class PLMLayer
PLMLayer instproc init { plm } {
        $self next
        $self instvar plm_ npkts_
        set plm_ $plm
        set npkts_ 0
        # loss trace created in constructor of derived class }
PLMLayer instproc join-group {} {
        $self instvar npkts_ add_time_ plm_
        set npkts_ [$self npkts]
        set add_time_ [$plm_ now]
        # derived class actually joins group}
PLMLayer instproc leave-group {} { }

PLMLayer instproc getting-pkts {} {
```

```
        $self instvar npkts_
        return [expr [$self npkts] != $npkts_]
}

# Usage: source utils.tcl (or, recompile it in ns2 as shown in README)
# Description:
#    This is a part of ERA codes.
# Author:
#    Somnuk Puangpronpitag
#    a Ph.d. student in University of Leeds, UK.
#    http://www.comp.leeds.ac.uk/nuk
#    a Lecturer of Mahasarakham University, Thailand
#    http://www.msu.ac.th
proc Min {x y} {
   return [expr $x < $y ? $x : $y] }
proc Max {x y} {
   return [expr $x > $y ? $x : $y] }



proc Square {x} {
   return [expr $x * $x] }
proc CalCumulativeRates {rate_of_each_layer} {
   set tt 0
   set cumulativeRates {}
   for {set i 0} {$i<[llength $rate_of_each_layer]} {incr i} {
      set tt [expr $tt + [lindex $rate_of_each_layer $i]]
      lappend cumulativeRates $tt
   }
   return $cumulativeRates
```

```
}
************************plm.tcl*********************************************

**********************************plm-ns.tcl*********************************
Application/Traffic/CBR_PP instproc set args {
        $self instvar packet_size_ rate_
        if { [lindex $args 0] == "interval_" } {
                puts "Cannot use CBR_PP with interval_, specify rate_ instead"}
        eval $self next $args}
Agent/LossMonitor/PLM instproc log-PP {} {}

Class PLMLossTrace -superclass Agent/LossMonitor/PLM
PLMLossTrace set expected_ -1
PLMLossTrace instproc init {} {
  $self next
  $self instvar lastTime measure debug_
  set lastTime 0
  set measure -1
  global plm_debug_flag
  if [info exists plm_debug_flag] {
        set debug_ $plm_debug_flag}}
PLMLossTrace instproc log-loss {} {
  $self instvar plm_
  $plm_ log-loss}
PP_estimate.
PLMLossTrace instproc log-PP {} {
  $self instvar plm_ PP_first measure next_pkt debug_
  global PP_burst_length packetSize
  if {[$self set flag_PP_] == 128} {
```

```
            set measure 1
            set next_pkt [expr [$self set seqno_] + 1]
            set PP_first [$self set packet_time_PP_]
            if {$debug_>=2} {trace_annotate "[$plm_ node]:  first PP [$self set seqno_], next:
$next_pkt"}
   } elseif {$measure>-1} {
            if {[$self set seqno_]==$next_pkt} {set measure [expr $measure + 1]
               set next_pkt [expr [$self set seqno_] + 1]
               if {$debug_>=2} {
            trace_annotate "[$plm_ node]:   pending measurement : $measure, next $next_pkt"}
               if {$measure==$PP_burst_length} {
                        set PP_value [expr $packetSize*8.*($PP_burst_length - 1)/([$self set
packet_time_PP_] - $PP_first)]
                        set measure -1
                        if {$debug_>=2} { trace_annotate "[$plm_ node]:  measure : $PP_value"}
                        $plm_ make_estimate $PP_value}
            } else {    if {$debug_>=2} {
                        trace_annotate "[$plm_ node]:  out of sequence : [$self set seqno_], next:
$next_pkt"}
               set measure -1}   }}
Class PLMLayer/ns -superclass PLMLayer
PLMLayer/ns instproc init {ns plm addr layerNo} {
   $self next $plm
   $self instvar ns_ addr_ mon_
   set ns_ $ns
   set addr_ $addr
   set mon_ [$ns_ PLMcreate-agent [$plm node] PLMLossTrace 0]
   $mon_ set layerNo $layerNo
   $mon_ set plm_ $plm
```

```
   $mon_ set dst_addr_ $addr
   $mon_ set dst_port_ 0}
#add a layer
PLMLayer/ns instproc join-group {} {
        $self instvar mon_ plm_ addr_
        $mon_ clear
        [$plm_ node] join-group $mon_ $addr_
        $self next}
#drop a layer
PLMLayer/ns instproc leave-group {} {
        $self instvar mon_ plm_ addr_
        [$plm_ node] leave-group $mon_ $addr_
        $self next}
#number of packets received for that layer
PLMLayer/ns instproc npkts {} {
        $self instvar mon_
        return [$mon_ set npkts_]}
#number of packets lost for that layer
PLMLayer/ns instproc nlost {} {
        $self instvar mon_
        return [$mon_ set nlost_]}
#allow to get statistics (number of packets received and lost) for a layer.
PLMLayer/ns instproc mon {} {
        $self instvar mon_
        return $mon_}
# This class serves as an interface between the PLM class which
# implements the PLM protocol machinery, and the objects in ns
# that are involved in the PLM protocol (i.e., Node objects
# join/leave multicast groups, LossMonitor objects report packet
```

```
# loss, etc...).<p>
# See tcl/ex/test-plm.tcl for an example of how to create a
# simulation script that uses PLM

Class PLM/ns -superclass PLM
PLM/ns instproc init {ns localNode addrs check_estimate nn} {
        $self instvar ns_ node_ addrs_
        set ns_ $ns
        set node_ $localNode
        set addrs_ $addrs
        $self next [llength $addrs] $check_estimate $nn}
PLM/ns instproc create-layer {layerNo} {
        $self instvar ns_ addrs_
        return [new PLMLayer/ns $ns_ $self [lindex $addrs_ $layerNo] $layerNo]}
PLM/ns instproc now {} {
        $self instvar ns_
        return [$ns_ now]}
PLM/ns instproc node {} {
        $self instvar node_
        return $node_}
PLM/ns instproc debug { msg } {
        $self instvar debug_ ns_
        if {$debug_ <1} { return }
        $self instvar subscription_ node_
        set time [format %.05f [$ns_ now]]
#       puts stderr "PLM/ns: $time node [$node_ id] layer $subscription_ $msg"}

PLM/ns instproc trace { trace } {
    $self instvar layers_
```

```
    foreach s $layers_ {
                [$s mon] trace $trace}}
PLM/ns instproc total_bytes_delivered {} {
        $self instvar layers_
    set v 0
    foreach s $layers_ {
        incr v [[$s mon] set bytes]  }
    return $v}

************************************plm-ns.tcl**********************************
****************************plm-topo.tcl***************************************
Simulator instproc PLMcreate-agent { node type pktClass } {
        $self instvar Agents PortID
        set agent [new $type]
        $agent set fid_ $pktClass
        $self attach-agent $node $agent
        $agent proc get var {
                return [$self set $var]}
        return $agent}
Simulator instproc PLMcbr_flow_PP { node fid addr bw } {
        global packetSize PP_burst_length
        set agent [$self PLMcreate-agent $node Agent/UDP $fid]
        set cbr [new Application/Traffic/CBR_PP]
        $cbr attach-agent $agent
    $agent set dst_addr_ $addr
    $agent set dst_port_ 0
        $cbr set packet_size_ $packetSize
        $cbr set rate_ $bw
        $cbr set random_ 1
```

```
        $cbr set PBM_ $PP_burst_length
                return $cbr}
#create a set fo layers for a given PLM source.
Simulator instproc PLMbuild_source_set { plmName rates addrs baseClass node when } {
    global src_plm src_rate
    set n [llength $rates]
    #mise en place de la layer de base avec PP
    set r [lindex $rates 0]
    set addr [expr [lindex $addrs 0]]
    set src_rate($addr) $r
    set k $plmName:0
    set src_plm($k) [$self PLMcbr_flow_PP $node $baseClass $addr $r]
    $self at 0 "$src_plm($k) set maxpkts_ 1; $src_plm($k) start"
    $self at $when "$src_plm($k) set maxpkts_ 268435456; $src_plm($k) start"
    #incr baseClass
   for {set i 1} {$i<$n} {incr i} {
                set r [lindex $rates $i]
                set addr [expr [lindex $addrs $i]]
                set src_rate($addr) $r
                set k $plmName:$i
                set src_plm($k) [$self PLMcbr_flow_PP $node $baseClass $addr $r]
                $self at 0 "$src_plm($k) set maxpkts_ 1; $src_plm($k) start"
                $self at $when "$src_plm($k) set maxpkts_ 268435456; $src_plm($k) start"
                 }}
Class PLMTopology
PLMTopology instproc init { simulator } {
                $self instvar ns id
                set ns $simulator
    #id determine de fid de chaque session PLM, faire attention aux
```

```
		set id 0}

PLMTopology instproc mknode nn {
		$self instvar node ns
		if ![info exists node($nn)] {set node($nn) [$ns node]}}
# build a link between nodes $a and $b
# if either node doesn't exist, create it as a side effect.
# (we don't build any sources)
PLMTopology instproc build_link { a b delay bw } {
		global buffers packetSize Queue_sched_
		if { $a == $b } {
				puts stderr "link from $a to $b?"
				exit 1}
		$self instvar node ns
		$self mknode $a
		$self mknode $b
		$ns duplex-link $node($a) $node($b) $bw $delay $Queue_sched_}
PLMTopology instproc build_link-simple { a b delay bw f} {
		global buffers packetSize Queue_sched_
		if { $a == $b } {
				puts stderr "link from $a to $b?"
				exit 1}
		$self instvar node ns
		$self mknode $a
		$self mknode $b
		$ns duplex-link-trace $node($a) $node($b) $bw $delay $Queue_sched_ $f
}
# build a new source (by allocating a new address) and
# place it at node $nn.  start it up at $when.
```

```
PLMTopology instproc place_source { nn when } {
        global rates
        $self instvar node ns id addrs
        incr id
        set addrs($id) {}
        foreach r $rates {lappend addrs($id) [Node allocaddr]}
        $ns PLMbuild_source_set s$id $rates $addrs($id) $id $node($nn) $when
        return $id}
#by default nb=1 does not give any functionality, the instance of the last
#PLM receiver is in PLMrcvr(1)
#If one specifies nb when one create the receiver, PLMrcvr($nb) contains
#the PLM instance of each PLM receiver. One must specifies a different nb per receiver.
PLMTopology instproc place_receiver { nn id when check_estimate {nb 1}} {
        $self instvar ns
        $ns at $when "$self build_receiver $nn $id $check_estimate $nb"}
# build a new receiver for source $id and
# place it at node $nn.
PLMTopology instproc build_receiver { nn id check_estimate nb} {
  $self instvar node ns addrs
  global PLMrcvr
  set PLMrcvr($nb) [new PLM/ns $ns $node($nn) $addrs($id) $check_estimate $nn]
  global plm_debug_flag
  $PLMrcvr($nb) set debug_ $plm_debug_flag}
****************************plm-topo.tcl****************************************
```

ประวัติย่อผู้วิจัย

ประวัติย่อผู้วิจัย

ชื่อ	นายกฤติน พันธุ์เสนา
วัน เดือน ปีเกิด	วันที่ 16 เมษายน พ.ศ. 2525
จังหวัด และประเทศที่เกิด	อำเภอห้วยเม็ก จังหวัดกาฬสินธุ์

ประวัติการศึกษา

พ.ศ. 2540	มัธยมศึกษาตอนต้น โรงเรียนห้วยเม็กวิทยาคม อำเภอห้วยเม็ก จังหวัดกาฬสินธุ์
พ.ศ. 2543	มัธยมศึกษาตอนปลาย โรงเรียนห้วยเม็กวิทยาคม อำเภอห้วยเม็ก จังหวัดกาฬสินธุ์
พ.ศ. 2548	ปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม
พ.ศ. 2550	ประกาศนียบัตรบัณฑิตวิชาชีพครู มหาวิทยาลัยราชภัฏมหาสารคาม
พ.ศ. 2555	ปริญญาวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยมหาสารคาม

ตำแหน่ง สถานที่ทำงาน	ข้าราชการครู โรงเรียนบ้านจานโนนสูง สพป.มค 3
ที่อยู่ที่สามารถติดต่อได้	โรงเรียนบ้านจานโนนสูง ตำบลเสือเฒ่า อำเภอเชียงยืน จังหวัดมหาสารคาม

รางวัลเรียนดี ทุนวิจัย และทุนการศึกษา

ทุนอุดหนุนการวิจัยเพื่อการพัฒนานิสิตระดับบัณฑิตศึกษา ปีการศึกษา 2552

ผลงานวิจัย

กฤติน พันธุ์เสนา, สมนึกพ่วงพรพิทักษ์ และ จิรัฏฐา ภูบุญอบ. การประเมินประสิทธิภาพโพรโทคอลควบคุมความคับคั่งแบบหลายอัตราของมัลติคาสท์ในการให้บริการมัลติมีเดียทราฟฟิกภายใต้กฎความเป็นมิตรต่อทีซีพี. ใน : **การประชุมวิชาการทางคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 12**. (หน้า 36-42). เชียงใหม่ : ดิอิมเพรสคอนเวนชั่นเซ็นเตอร์เชียงใหม่.