การศึกษาพฤติกรรมของทราฟฟิกแบบต่อเนื่อง

STUDY ON STREAMING TRAFFIC BEHAVIOR

สราวุธ อมราสิงห์

SARAWUT AMARASING

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

ISBN 974-15-2707-1

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การศึกษาพฤติกรรมของทราฟฟิกแบบต่อเนื่อง

STUDY ON STREAMING TRAFFIC BEHAVIOR

สราวุธ อมราสิงห์

SARAWUT AMARASING

เลขหมู่.....
เลขทะเบียน.....63624
วัน,เดือน,ปี.....30 ส.ค. 2549

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พศ. 2549

ISBN 974-15-2707-1

STUDY ON STREAMING TRAFFIC BEHAVIOR

SARAWUT AMARASING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN INFORMATION ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

ISBN 974-15-2707-1

COPYRIGHT 2006

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การศึกษาพฤติกรรมของทราฟฟิกแบบต่อเนื่อง
นักศึกษา	นายสราวุธ อมราสิงห์
รหัสประจำตัว	44061804
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ. มยุรี เลิศเวชกุล

บทคัดย่อ

ทุกวันนี้การใช้งานระบบเครือข่ายไม่ได้ถูกออกแบบเพียงเพื่อรองรับแต่การใช้งานรับ-ส่งไฟล์ เว็บ และอีเมล์อีกต่อไปแล้ว การใช้เครือข่ายในปัจจุบันมีหลากหลายรูปแบบมากยิ่งขึ้น ตามความต้องการของผู้ใช้งาน เพื่ออำนวยความสะดวกและแทนที่การใช้งาน หรือการสื่อสารหลาย ๆ อย่าง ระบบการประชุมทางไกล และ ความบันเทิงหลาย ๆ รูปแบบ ถูกนำมาปรับใช้งานบนเครือข่ายมากขึ้น เช่น การประชุมทางไกล การดูหนัง- ฟังเพลงผ่านระบบอินเทอร์เน็ต ซึ่งรูปแบบการใช้งานดังกล่าวส่วนใหญ่นั้นเป็นการสื่อสารแบบต่อเนื่องและการใช้งานเหล่านี้จะเป็นการใช้งานหลักและเข้ามามีบทบาทมากยิ่งขึ้นต่อระบบเครือข่ายในอนาคต ดังนั้นความเข้าใจการทำงานของทราฟฟิกแบบต่อเนื่องจึงมีประโยชน์อย่างยิ่งในการใช้งานและออกแบบระบบเครือข่ายให้มีความเหมาะสมมากขึ้น แนวทางในการศึกษานี้เป็นการการนำเสนอพฤติกรรมของทราฟฟิกแบบต่อเนื่องที่ถูกต้องและชัดเจนทั้งแบบเรียลไทม์และออนดีมานด์ ผลจากการวิจัยนี้จะแสดงถึงพฤติกรรมและความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูล (อัตราการเข้ารหัสข้อมูลที่ส่ง) กับพฤติกรรมของการส่งข้อมูลแบบต่อเนื่องและนอกจากนี้ยังแสดงให้เห็นพฤติกรรมที่แตกต่างกันระหว่างทราฟฟิกแบบต่อเนื่องทั้ง 2 ประเภทกับทราฟฟิกประเภทอื่น ๆ พร้อมทั้งยังพัฒนาแบบจำลองของทราฟฟิกแบบต่อเนื่องขึ้นเพื่อให้เข้าใจการทำงานของทราฟฟิกแบบต่อเนื่องมากยิ่งขึ้นและนำไปใช้ในการศึกษาหรือออกแบบระบบเครือข่ายต่อไป

Thesis Title	Study on Streaming Traffic behavior
Student	Mr. Sarawut Amarasing
Student ID.	44061804
Degree	Master of Engineering
Programme	Information Engineering
Year	2006
Thesis Adviser	Asst.Prof. Mayuree Lertwatechakul

ABSTRACT

Recently designing of computer network has no longer been limited only for sending and/or receiving files, webpages, or e-mails; it has also been designed and developed continuously to serve the increasing users' requirements. Nowadays, teleconference, tele-lecturing and Internet entertainment such as VoD, live concert via the Internet have been widely popular (for example, the use of real-time teleconference or net meeting, online multimedia, etc.). Most of those applications generate streaming traffic which would play an important role for computer network in the future. Therefore, the understanding of the streamline traffic's behavior is very crucial. Studying of streaming traffic behavior will be useful for the improvement of the computer network system designing. As to provide an useful information about streaming traffic behavior, this research studied on 2 types of streaming traffic: stored media and real-time live broadcast with various sample data. The captured traffics were investigated in terms of packet size distribution and packet interval time distribution. The results of the analysis, we could make a conclusion on the different of the stored media traffic and the real-time live broadcast traffic and the relationship between the encoding rate and the transmission rate. Moreover, this research also develope the streaming traffic model for both stored media traffic and real-time live broadcast traffic. The developed traffic model could generate traffic similar to the real traffic as shown in the experiment results.

กิตติกรรมประกาศ

การทำวิทยานิพนธ์ครั้งนี้ สำเร็จลงได้ด้วยปัจจัยหลาย ๆ ด้าน ผู้เขียนขอกล่าวขอบพระคุณทุกท่านไว้ ณ ที่นี้อันได้แก่ เพื่อน ๆ ที่ช่วยให้การสนับสนุนทั้งในเรื่องของอุปกรณ์ ความรู้และวิธีการต่าง ๆ เพื่อให้วิทยานิพนธ์เล่มนี้ออกมาสมบูรณ์ได้ในทุกรายละเอียด

และที่ลืมไม่ได้ ผศ. มยุรี เลิศเวชกุล อาจารย์ผู้ควบคุมวิทยานิพนธ์นี้ ผู้ซึ่งให้คำแนะนำต่าง ๆ พร้อมทั้งช่วยตรวจทานแก้ไข รวมถึงให้การสนับสนุนในทุก ๆ ด้านเพื่อให้วิทยานิพนธ์เล่มนี้เสร็จสมบูรณ์ รวมถึงอาจารย์ภาควิชาวิศวกรรมสารสนเทศทุกท่านที่ให้ข้อเสนอแนะในการแก้ปัญหา และสุดท้าย ขอขอบพระคุณ บิดามารดา ที่ทำให้ผู้เขียนได้มีโอกาสตรงนี้ จึงขอขอบคุณทุกท่านมา ณ โอกาสนี้ด้วย

สราวุธ อมราสิงห์

สารบัญ

หน้า

บทคัดย่อ I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย 1

1.3 สมมติฐานของการวิจัย 2

1.4 ขอบเขตงานวิจัย 2

1.5 ขั้นตอนของการวิจัย 3

บทที่ 2 ประเภทของอินเทอร์เน็ตทราฟฟิก 4

2.1 ประเภทของอินเทอร์เน็ตทราฟฟิกตามลักษณะการใช้งาน 4

2.2 การทำงานของระบบทราฟฟิกแบบต่อเนื่อง 5

2.3 เทคโนโลยีของระบบทราฟฟิกแบบต่อเนื่องในปัจจุบัน 6

2.4 ประเภทการให้บริการของระบบทราฟฟิกแบบต่อเนื่อง 7

2.4.1 ทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ 7

2.4.2 ทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ 7

2.5 การเข้ารหัสและการบีบอัดข้อมูลมัลติมีเดีย 7

2.6 โครงสร้างการทำงานของระบบทราฟฟิกแบบต่อเนื่องโดยใช้เรียลมีเดีย 9

2.7 คุณลักษณะเฉพาะตัวของทราฟฟิกแบบต่อเนื่อง 10

2.8 สรุปท้ายบท 11

บทที่ 3 การทดลองเพื่อระบุคุณลักษณะของทราฟฟิกแบบต่อเนื่อง 12

3.1 การทดลองทราฟฟิกแบบต่อเนื่อง 12

3.1.1 สมมติฐานการทดลอง 12

สารบัญ (ต่อ)

หน้า

3.1.2 แนวทางการทดลอง 12
3.2 ผลและการวิเคราะห์ผลการทดลอง ..... .15
3.2.1ผลการทดลองและการวิเคราะห์ทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์15
3.2.2 ผลการทดลองและการวิเคราะห์ทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ .. 19
3.2.3 ความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลกับพฤติกรรมการส่งข้อมูล 21
3.2.4 ความแตกต่างของการส่งข้อมูลทราฟฟิกแบบออนดีมานด์กับเรียลไทม์..... 23
3.3 สรุปท้ายบท 41

บทที่ 4 การจำลองทราฟฟิกแบบต่อเนื่อง..... 42
4.1 การจำลองระบบเครือข่ายเพื่อการวิเคราะห์ 42
4.2 การออกแบบจำลองทราฟฟิกแบบต่อเนื่อง 43
4.2.1 แบบจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ 43
4.2.2 แบบจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์..... 44
4.3 การสร้างแบบจำลองทราฟฟิกแบบต่อเนื่องบน NS-2 45
4.4 ผลการทดลองการจำลองทราฟฟิกแบบต่อเนื่อง 49
4.4.1 รูปแบบเครือข่ายจำลองบน NS-2..... 49
4.4.2 ผลการจำลองทราฟฟิกแบบต่อเนื่อง 50
4.4.2.1 ผลการจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ 50
4.4.2.2 ผลการจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์..... 52
4.4.2.3 เปรียบเทียบผลการจำลองทราฟฟิกกับทราฟฟิกจริง 54
4.4.2.4 ความแตกต่างของทราฟฟิกจริงกับทราฟฟิกจำลองแบบต่อเนื่อง.. 58
4.5 สรุปท้ายบท..... 58

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 59

เอกสารอ้างอิง 61

สารบัญ (ต่อ)

หน้า

ภาคผนวก ก โปรแกรมฟังก์ชันทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ในช่วงเริ่มต้น..... 62
โปรแกรมฟังก์ชันทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ในช่วงปกติ..... 65
โปรแกรมฟังก์ชันทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์..... 68
ภาคผนวก ข โปรแกรมจำลองเครือข่ายทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์..... 73
โปรแกรมจำลองเครือข่ายทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์..... 77
ภาคผนวก ค การคอมไพล์อ็อบเจกต์บน NS-2..... 82

ประวัติผู้เขียน..... 83

สารบัญตาราง

ตารางที่ หน้า

2.1 ประเภทของอินเทอร์เน็ตทราฟฟิก 4

2.2 ระบบทราฟฟิกแบบต่อเนื่องที่นิยมใช้ในปัจจุบัน 6

2.3 เทคโนโลยีการบีบอัดข้อมูลของไฟล์มัลติมีเดียแต่ละประเภท 8

3.1 คุณสมบัติของเครื่องที่ใช้ในการทดลอง 13

3.2 กลุ่มตัวอย่างการทดลองและขนาดของไฟล์ตัวอย่างที่ผ่านการเข้ารหัสข้อมูล 14

3.3 ความสัมพันธ์ของอัตราการนำส่งข้อมูลกับค่าเฉลี่ยของขนาดแพ็กเกตระหว่างวีดีโอ ประเภทแอคชั่นและทอล์คโชว์ในทราฟฟิกประเภทออนดีมานด์ 17

3.4 ความสัมพันธ์ของอัตราการนำส่งข้อมูลกับค่าสูงสุดของขนาดแพ็กเกตระหว่างวีดีโอ ประเภทแอคชั่นและทอล์คโชว์ในทราฟฟิกประเภทออนดีมานด์ 18

3.5 ความสัมพันธ์ของอัตราการนำส่งข้อมูลกับค่าเฉลี่ยของช่วงเวลาการส่งแต่ละแพ็กเกต ระหว่างวีดีโอประเภทแอคชั่นและทอล์คโชว์ในทราฟฟิกประเภทออนดีมานด์ 18

3.6 ความสัมพันธ์ของอัตราการนำส่งข้อมูลกับค่าเฉลี่ยของขนาดแพ็กเกต ระหว่างวีดีโอ ประเภทแอคชั่นและทอล์คโชว์ของทราฟฟิกประเภทเรียลไทม์ 20

3.7 ความสัมพันธ์ของอัตราการนำส่งข้อมูลกับค่าสูงสุดของขนาดแพ็กเกต ระหว่างวีดีโอ ประเภทแอคชั่นและทอล์คโชว์ของทราฟฟิกประเภทเรียลไทม์ 21

3.8 ความสัมพันธ์ของอัตราการนำส่งข้อมูลกับค่าเฉลี่ยของช่วงเวลาการส่งแต่ละแพ็กเกต ระหว่างวีดีโอประเภทแอคชั่นและทอล์คโชว์ 21

3.9 ค่าเฉลี่ยและมีเดียนของช่วงเวลาการส่งแต่ละแพ็กเกตของการส่งประเภทต่าง ๆ 27

3.10 ความสัมพันธ์ของอัตราการนำส่งข้อมูลกับค่าเฉลี่ยของช่วงส่ง ON-OFF ในการส่งข้อมูล ระหว่างวีดีโอประเภทแอคชั่นและทอล์คโชว์ของเรียลไทม์ทราฟฟิก 36

4.1 ค่าอินพุตที่ใช้สำหรับจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ 43

4.2 ค่าอินพุตที่ใช้สำหรับจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ 45

สารบัญรูป

รูปที่ หน้า

2.1 องค์ประกอบของระบบทราฟฟิกแบบต่อเนื่อง 5

2.2 ขั้นตอนการสื่อสารของระบบทราฟฟิกแบบต่อเนื่อง 10

3.1 การเชื่อมต่อที่ใช้ในการทดลอง 13

3.2 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบออนดีมานด์ด้วยอัตราการนำส่งต่าง ๆ กัน . 16

3.3 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบออนดีมานด์ด้วยประเภทของตัวอย่างไฟล์มัลติมีเดียที่ต่างประเภทกัน 16

3.4 พฤติกรรมการส่งข้อมูลของ เรียลเซิร์ฟเวอร์แบบออนดีมานด์โดยเวลาแสดงผลจริงของไฟล์มีความยาว 138 วินาที จาก http://news.bbc.co.uk 16

3.5 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบเรียลไทม์ด้วยอัตราการนำส่งต่าง ๆ กัน 19

3.6 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบเรียลไทม์ด้วยประเภทของตัวอย่างมัลติมีเดียที่ต่างกัน 20

3.7 ความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลกับค่าเฉลี่ยของขนาดแพ็กเกต 22

3.8 ความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลกับช่วงเวลาเฉลี่ยการส่งข้อมูลแต่ละแพ็กเกต ... 22

3.9 (a) - (d) ช่วงเวลาระยะห่างแต่ละแพ็กเกตของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps 24-25

3.10 (a) - (d) ช่วงเวลาระยะห่างแต่ละแพ็กเกตของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps 25-26

3.11 (a) - (d) กราฟการกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของทราฟฟิกประเภทออนดีมานด์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps 27-29

3.12 (a) - (d) กราฟการกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของทราฟฟิกประเภทเรียลไทม์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps 29-31

3.13 (a) - (d) กราฟช่วงเวลาและขนาดของแพ็กเกตที่ส่งด้วยทราฟฟิกแบบเรียลไทม์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps 32-33

3.14 (a) - (d) กราฟช่วงเวลาและขนาดของแพ็กเกตที่ส่งด้วยทราฟฟิกแบบออนดีมานด์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps 34-35

3.15 (a) - (d) กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกประเภทออนดีมานด์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps 37-38

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.16 (a) - (d) กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกประเภทเรียลไทม์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps 39-40

4.1 ฟังก์ชันแบบจำลองของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ 44

4.2 ฟังก์ชันแบบจำลองของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ 45

4.3 ขั้นตอนการทำงานของแบบจำลองของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ 47

4.4 การจำลองรูปแบบของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ 48

4.5 ขั้นตอนการทำงานของแบบจำลองของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ 48

4.6 รูปแบบการเชื่อมต่อในการจำลองทราฟฟิกแบบต่อเนื่อง 49

4.7 ผลการทดลองจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ 50

4.8 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตราการนำส่ง 56 Kbps 51

4.9 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตราการนำส่ง 128 Kbps 51

4.10 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตราการนำส่ง 512 Kbps 51

4.11 ผลการทดลองจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ 52

4.12 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตราการนำส่ง 56 Kbps 53

4.13 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตราการนำส่ง 128 Kbps 53

4.14 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตราการนำส่ง 512 Kbps 53

4.15 (a) - (b) เปรียบเทียบพฤติกรรมของทราฟฟิกจริงและทราฟฟิกที่จำลองขึ้นจากทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ 54

4.16 (a) - (b) เปรียบเทียบพฤติกรรมของทราฟฟิกจริงและทราฟฟิกที่จำลองขึ้นจากทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตราการนำส่ง 128 Kbps ในช่วงเวลาการส่งข้อมูลที่เล็กลง 55

4.17 (a) - (b) เปรียบเทียบพฤติกรรมของทราฟฟิกจริงและทราฟฟิกที่จำลองขึ้นจากทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ 56

4.18 (a) - (b) เปรียบเทียบพฤติกรรมของทราฟฟิกจริงและทราฟฟิกที่จำลองขึ้นจากทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตราการนำส่ง 128 Kbps ในช่วงเวลาการส่งข้อมูลที่เล็กลง 57

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันทราฟฟิกแบบต่อเนื่อง มีการใช้งานเพิ่มขึ้นอย่างแพร่หลายในระบบอินเทอร์เน็ตโดยมีความก้าวหน้าและรูปแบบหลากหลายมากมายขึ้นอยู่กับวัตถุประสงค์การใช้งาน เช่น การดูหนังฟังเพลงออนไลน์บนอินเทอร์เน็ต การสนทนาหรือประชุมผ่านระบบเครือข่าย เป็นต้น ซึ่งทั้งหมดที่กล่าวมานี้ล้วนเป็นลักษณะการส่งข้อมูลด้วยทราฟฟิกแบบต่อเนื่องทั้งสิ้น โดยเบื้องต้นแล้วระบบการสื่อสารทราฟฟิกแบบต่อเนื่องเป็นลักษณะการส่งข้อมูลประเภทที่มีความอ่อนไหวต่อปัจจัยทางด้านขนาดช่องสัญญาณและความหน่วงของข้อมูลเป็นอย่างมาก โดยการแสดงผลของแอพพลิเคชันและการใช้งานของผู้ใช้จะมีผลกระทบทันทีถ้าช่องสัญญาณมีขนาดไม่เพียงพอหรือเกิดความหน่วงของข้อมูลขึ้นในระบบในระดับที่ยอมรับไม่ได้

ระบบเครือข่ายและอุปกรณ์ในปัจจุบันหลาย ๆ ส่วนเริ่มให้ความสำคัญต่อระบบทราฟฟิกแบบต่อเนื่องมากขึ้นในการออกแบบและพัฒนาการทำงานให้เหมาะสมกับการใช้งาน เช่น เรื่องของการกำหนดขนาดช่องสัญญาณและการกำหนดคุณสมบัติช่องสัญญาณของทราฟฟิก (QOS) เป็นต้น แต่ว่าในปัจจุบันนี้จากที่ได้ศึกษามาพบว่ายังไม่มีการจำแนกรูปแบบที่ชัดเจนของทราฟฟิกแบบต่อเนื่อง โดยเฉพาะอย่างยิ่งสำหรับความแตกต่างระหว่างประเภทของทราฟฟิกแบบต่อเนื่องระหว่างทราฟฟิกออนดีมานด์กับเรียลไทม์ และความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลที่ต่างกันซึ่งปัจจัยเหล่านี้น่าจะมีผลกับพฤติกรรมของการส่งข้อมูล ดังนั้นการเข้าใจพฤติกรรมของทราฟฟิกแบบต่อเนื่องจะมีประโยชน์อย่างยิ่งกับการออกแบบและพัฒนาระบบเครือข่ายให้เหมาะสมต่อไป

1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย

ในปัจจุบันนี้การที่จะศึกษาและออกแบบระบบเครือข่ายให้มีประสิทธิภาพสูงสุดนั้นจำเป็นต้องเข้าใจถึงการทำงานและคุณลักษณะโดยละเอียดของการสื่อสารต่าง ๆ ที่เกิดขึ้น เพื่อที่จะออกแบบระบบให้ถูกต้องและเหมาะสมกับการทำงาน รวมถึงสามารถจำลองรูปแบบการสื่อสารที่จะเกิดขึ้นในระบบเพื่อพิจารณาผลการทำงานก่อนที่จะติดตั้งใช้งานจริง จากการใช้งานระบบเครือข่ายในปัจจุบันพบว่าการสื่อสารระบบทราฟฟิกแบบต่อเนื่องมีการใช้งานแพร่หลายมากขึ้น แต่จากการศึกษานั้น พบว่ายังไม่มีการกำหนดคุณลักษณะที่ชัดเจนของทราฟฟิกแบบต่อเนื่อง ทำให้ผลของ

การศึกษาและออกแบบระบบเครือข่ายที่อ้างอิงการสื่อสารของทราฟฟิกแบบต่อเนื่องได้ผลลัพธ์ที่ยังไม่ถูกต้องนัก

งานวิจัยนี้มีจุดมุ่งหมายหลักเพื่อศึกษาพฤติกรรมและปัจจัยต่าง ๆ ที่มีผลต่อพฤติกรรมการส่งข้อมูลของทราฟฟิกแบบต่อเนื่องโดยละเอียด ซึ่งศึกษาทราฟฟิกแบบต่อเนื่องประเภทต่าง ๆ และสรุปหารูปแบบของพฤติกรรมการส่งข้อมูลของทราฟฟิกแบบต่อเนื่องแต่ละประเภท ทั้งยังพัฒนาแบบจำลองของทราฟฟิกแบบต่อเนื่องบน NS-2 ที่สามารถจำลองทราฟฟิกแบบต่อเนื่องได้ใกล้เคียงกับทราฟฟิกที่เกิดขึ้นในระบบจริง ทั้งยังสามารถกำหนดค่าตัวแปรและอัตราการนำส่งในการจำลองทราฟฟิกได้ง่าย เพื่อนำไปใช้ในการศึกษา วิจัย หรือออกแบบระบบเครือข่ายให้เหมาะสมกับทราฟฟิกแบบต่อเนื่องต่อไป

1.3 สมมติฐานของการวิจัย

ทราฟฟิกแบบต่อเนื่องนั้นมีรูปแบบเฉพาะตัวที่แตกต่างจากทราฟฟิกประเภทอื่น ๆ โดยในทราฟฟิกแบบต่อเนื่องด้วยกันนั้น ยังแบ่งประเภทของทราฟฟิกออกด้วยความแตกต่างกันของวัตถุประสงค์การใช้งานได้แก่ทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์และออนดีมานด์ ซึ่งทั้ง 2 ประเภทนี้น่าจะมีความแตกต่างกันทางด้านพฤติกรรมการส่งข้อมูลอีกด้วย นอกจากนั้นการเผยแพร่ไฟล์ประเภทมัลติมีเดียด้วยทราฟฟิกแบบต่อเนื่องมักจะมีการกำหนดอัตราการนำส่งข้อมูลสำหรับผู้ใช้บริการ โดยการเข้ารหัสไฟล์ด้วยอัตราการนำส่งต่าง ๆ กันเป็นอีกปัจจัยหนึ่งที่มีผลต่อพฤติกรรมการส่งข้อมูล

1.4 ขอบเขตการวิจัย

ทำการทดลองเพื่อหาคุณลักษณะพฤติกรรมทราฟฟิกแบบต่อเนื่องและความแตกต่างของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์และออนดีมานด์ รวมถึงความสัมพันธ์ของอัตราการนำส่งข้อมูลที่มีผลต่อพฤติกรรมการส่งข้อมูล และนำข้อสรุปของพฤติกรรมการส่งข้อมูลที่ได้จากวิเคราะห์ผลการทดลองมาสร้างแบบจำลองของทราฟฟิกแบบต่อเนื่องทั้ง 2 ประเภทเพื่อนำไปจำลองระบบเครือข่ายให้ได้ผลใกล้เคียงกับทราฟฟิกแบบต่อเนื่องที่ใช้ในระบบจริงมากยิ่งขึ้น

1.5 ขั้นตอนของการวิจัย

1. ศึกษาคุณลักษณะต่าง ๆ และพฤติกรรมพื้นฐานของทราฟฟิกแบบต่อเนื่อง
2. กำหนดทดลองทราฟฟิกแบบต่อเนื่องทั้งประเภทออนดีมานด์และเรียลไทม์จากการให้บริการของโปรแกรมเรียลมีเดีย (RealMedia)
3. วิเคราะห์รูปแบบของทราฟฟิกแบบต่อเนื่องและความแตกต่างของทราฟฟิกแบบต่อเนื่องทั้ง 2 ประเภทรวมถึงปัจจัยต่าง ๆ ที่มีผลต่อลักษณะการส่งข้อมูล
4. กำหนดลักษณะของแบบจำลองทราฟฟิกแบบต่อเนื่อง จากผลการวิเคราะห์ที่ได้จากการทดลอง
5. สร้างแบบจำลองและพัฒนาฟังก์ชันของทราฟฟิกบนโปรแกรมจำลองระบบเครือข่าย NS-2
6. เปรียบเทียบผลการทดลองของทราฟฟิกที่ได้จากแบบจำลองกับทราฟฟิกจริง
7. สรุปผลและวิจารณ์

บทที่ 2
ประเภทของอินเทอร์เน็ตทราฟฟิก

ปัจจุบันอินเทอร์เน็ตทราฟฟิกมีการใช้งานอย่างแพร่หลายและสามารถแบ่งออกมาได้มากมายหลายประเภท โดยแบ่งได้ด้วยวิธีการต่าง ๆ กัน

2.1 ประเภทของอินเทอร์เน็ตทราฟฟิกตามลักษณะการใช้งาน

มีวิธีแบ่งประเภทของอินเทอร์เน็ตทราฟฟิกมากมายในปัจจุบัน การแบ่งประเภทของอินเทอร์เน็ตทราฟฟิกออกตามลักษณะการใช้งานนั้นถือเป็นอีกวิธีหนึ่งที่นิยมนำมาใช้ในการแบ่งประเภท โดยสามารถแบ่งเป็นประเภทใหญ่ ๆ ได้ดังตารางที่ 2.1 ดังเช่น ตัวอย่างของแอพพลิเคชั่นของทราฟฟิกประเภท อีเมล์ได้แก่ โปรโตคอลเอสเอ็มทีพี (SMTP) สำหรับส่งอีเมล์ โปรโตคอลไอแมพ (IMAP) และพ็อพ (POP) ที่ใช้สำหรับรับอีเมล์ เป็นต้น

ตารางที่ 2. 1 ประเภทของอินเทอร์เน็ตทราฟฟิก

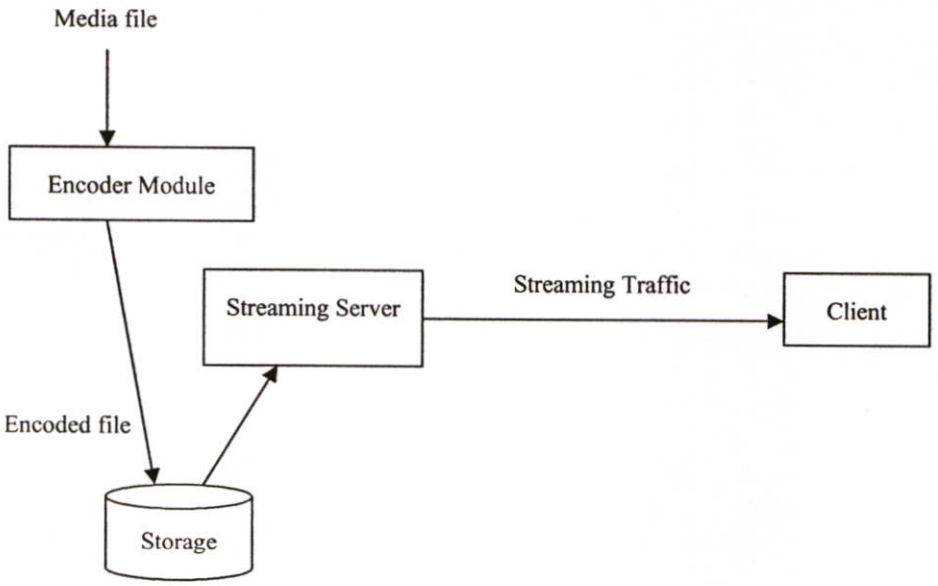
Classification	**Example Application**
BULK	ftp
DATABASE	postgres, sqlnet, oracle, ingres
INTERACTIVE	ssh, klogin, rlogin, telnet
MAIL	imap, pop2/3, smtp
SERVICES	X11, dns, ident, ldap, ntp
WWW	www
P2P	KaZaa, BitTorrent, GnuTella
ATTACK	Internet worm and virus attacks
GAMES	Half-Life
MULTIMEDIA	Windows Media Player, Real Player

จากการแบ่งประเภทดังกล่าว มีประเภทของอินเทอร์เน็ตทราฟฟิกประเภทหนึ่งซึ่งน่าสนใจ คือประเภทมัลติมีเดีย ทราฟฟิกประเภทนี้กำลังมีการใช้งานเพิ่มขึ้นเป็นอย่างมาก วัตถุประสงค์ของการใช้งานทราฟฟิกประเภทนี้มีหลากหลายประเภท เพื่อการสื่อสาร ความบันเทิงและการศึกษา เช่น การดูหนัง ฟังเพลงผ่านระบบอินเทอร์เน็ต การเชื่อมต่อสื่อสารแบบไร้พรมแดน การศึกษาผ่านระบบอินเทอร์เน็ต เป็นต้น เนื่องจากการพัฒนาของระบบอินเทอร์เน็ต การขยายแบนด์วิดท์ให้มากขึ้น ทำให้ผู้ให้บริการและผู้ใช้งานอินเทอร์เน็ตเริ่มหันมาให้บริการอินเทอร์เน็ตทราฟฟิกประเภทนี้มากยิ่งขึ้น และยังมีผู้ให้บริการทราฟฟิกแบบต่อเนื่องด้วยเนื้อหาและวัตถุประสงค์ต่าง ๆ มากมาย ซึ่งทำให้ทราฟฟิกแบบต่อเนื่องมีปริมาณความต้องการใช้งานเติบโตขึ้นอย่างรวดเร็วและมีอิทธิพลต่อระบบเครือข่ายต่อไปในอนาคต [1]

2.2 การทำงานของระบบทราฟฟิกแบบต่อเนื่อง

ภาพรวมการทำงานของระบบทราฟฟิกแบบต่อเนื่องเป็นดังรูปที่ 2.1 โดยจะประกอบด้วยองค์ประกอบหลัก 3 ส่วนดังนี้

1. ส่วนให้บริการทราฟฟิกแบบต่อเนื่อง
2. ส่วนเข้ารหัสไฟล์มีเดีย
3. ส่วนผู้รับบริการ



รูปที่ 2.1 องค์ประกอบของระบบทราฟฟิกแบบต่อเนื่อง

ขั้นตอนการทำงานเริ่มด้วยไฟล์ข้อมูลที่จะให้บริการจะถูกเข้ารหัสข้อมูลด้วยโปรแกรมเข้ารหัสซึ่งสามารถเลือกอัตราการนำส่งข้อมูลที่จะให้บริการได้ตามต้องการ การเข้ารหัสไฟล์ข้อมูลประกอบด้วยการสร้างเฟรมข้อมูลที่แตกต่างกัน 2 ประเภทคือ อินเทอร์เฟรมและอินทราเฟรม อินทราเฟรมจะเรียกว่าไอเฟรม สำหรับอินเทอร์เฟรมจะประกอบด้วยพีเฟรมและบีเฟรม วิธีการแสดงผลนั้นจะใช้ไอเฟรมซึ่งมีขนาดใหญ่เป็นเฟรมอ้างอิงและใช้พีเฟรมหรือบีเฟรมที่มีขนาดเล็กเพื่อแสดงผลของเฟรมถัดไปหรือเฟรมก่อนหน้านี้ [2]

หลังจากเข้ารหัสไฟล์เรียบร้อยแล้วข้อมูลดังกล่าวจะถูกจัดเก็บลงบนดิสก์เพื่อนำไปให้บริการต่อไป เครื่องผู้ให้บริการจะกำหนดวิธีการให้บริการรูปแบบต่าง ๆ ตามต้องการ หลังจากนั้นเมื่อฝ่ายผู้ใช้บริการต้องการใช้งานจะร้องขอไปยังผู้ให้บริการทราฟฟิกแบบต่อเนื่อง และเมื่อผู้ให้บริการได้รับการร้องขอผู้ให้บริการจะส่งข้อมูลออกไปยังฝ่ายผู้ใช้บริการโดยควบคุมลักษณะการส่งข้อมูลตามอัตราการนำส่งที่ตกลงกันไว้ หลังจากฝ่ายรับได้รับข้อมูลแล้วจึงนำข้อมูลดังกล่าวมาแสดงผลต่อไป

2.3 เทคโนโลยีของระบบทราฟฟิกแบบต่อเนื่องในปัจจุบัน

เทคโนโลยีของทราฟฟิกแบบต่อเนื่องในปัจจุบันนี้ได้รับการพัฒนาขึ้นอย่างมากและมีการพัฒนาแยกออกไปตามผู้ผลิตแต่ละราย ซึ่งต่างออกแบบระบบทราฟฟิกแบบต่อเนื่องและการเข้ารหัสโดยใช้เทคโนโลยีที่ต่างกันออกไป แต่ทั้งหมดนี้ล้วนพัฒนาขึ้นด้วยมาตรฐานการทำงานของระบบทราฟฟิกแบบต่อเนื่องเดียวกัน ระบบที่นิยมใช้งานในปัจจุบันมี 3 รูปแบบด้วยกัน ดังตารางที่ 2.2

ตารางที่ 2.2 ระบบทราฟฟิกแบบต่อเนื่องที่นิยมใช้ในปัจจุบัน

Vendor	**Streaming Server**	**Streaming Format**	**Media player**
Microsoft	Window Media Server, WMS	.wmv or .asf	Window Media Player
Real Network	Real Server, Helix	.rm	RealOne, RealPlayer
Apple	QuickTime Streaming Server, QTSS	.mov or .qt	QuickTime Player

ทราฟฟิกจากแอพพลิเคชันทั้ง 3 รูปแบบนี้มีพฤติกรรมที่แตกต่างกันเพราะลักษณะการเข้ารหัสไฟล์ของแอพพลิเคชันเองและโปรโตคอลชั้นนำส่งที่แต่ละแอพลลิเคชั่นเลือกใช้ เช่น ความแตกต่างระหว่างวินโดว์มีเดีย (Windows Media Service) และเรียลมีเดียนั้น เรียลมีเดียมีลักษณะการส่งข้อมูลด้วยโปรโตคอลยูดีพี ในขณะที่วินโดว์มีเดียใช้โปรโตคอลทีซีพี ทางด้านพฤติกรรมการส่งข้อมูลนั้น พบว่าเรียลมีเดียมีการส่งข้อมูลที่หลากหลายมากกว่าทั้งทางด้านของขนาดแพ็กเกตและช่วง

ระยะห่างระหว่างแพ็กเกต ซึ่งต่างจากวินโดว์มีเดียที่ลักษณะการส่งค่อนข้างคงที่ตลอดช่วงการส่งข้อมูล [3]

2.4 ประเภทการให้บริการของระบบทราฟฟิกแบบต่อเนื่อง

ระบบทราฟฟิกแบบต่อเนื่องถูกออกแบบมาเพื่อรองรับการใช้งานตามวัตถุประสงค์ของผู้ใช้งานและรูปแบบของการให้บริการ ซึ่งทั้ง 2 รูปแบบนี้ประกอบไปด้วย

2.4.1 ทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

เมื่อผู้ใช้งานเรียกใช้ทราฟฟิกประเภทนี้ การแสดงผลที่ผู้ใช้จะสามารถแสดงผลได้ตั้งแต่ต้น นอกจากนี้ยังสามารถเลือกช่วงของการแสดงผลไปข้างหน้า ย้อนกลับ หรือหยุดได้ตามต้องการ เหมือนกับว่ามีข้อมูลนั้น ๆ อยู่ที่เครื่อง โดยผู้ใช้งานส่งสัญญาณร้องขอข้อมูลในส่วนที่ต้องการ เมื่อผู้ให้บริการได้รับสัญญาณดังกล่าวจะทำการประมวลผลและส่งข้อมูลส่วนนั้น ๆ กลับไปแสดงผลยังผู้ใช้งานต่อไป การให้บริการประเภทนี้นิยมใช้กับการฟังเพลง ดูหนังบนอินเทอร์เน็ต

2.4.2 ทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

ทราฟฟิกประเภทนี้มีการแสดงผลที่ต่างไปจากประเภทออนดีมานด์ การแสดงผลจะเป็นไปอย่างต่อเนื่องไปตลอด โดยผู้ใช้งานไม่สามารถเลือกช่วงของการแสดงผลได้ การให้บริการประเภทนี้นิยมใช้กับการถ่ายทอดสดวิทยุหรือโทรทัศน์ผ่านอินเทอร์เน็ต

นอกจากความแตกต่างทางด้านวัตถุประสงค์การใช้งานแล้ว ทราฟฟิกแบบต่อเนื่องทั้ง 2 ประเภทนี้มีลักษณะของการส่งข้อมูลที่แตกต่างกัน ดังนั้นจึงควรแยกประเภทของทราฟฟิกแบบต่อเนื่องให้ชัดเจนเพื่อจะได้เข้าใจถึงพฤติกรรมของทราฟฟิกแบบต่อเนื่องประเภทนั้น ๆ ได้อย่างถูกต้อง

2.5 การเข้ารหัสและการบีบอัดข้อมูลมัลติมีเดีย

สำหรับขั้นตอนในการจัดเตรียมไฟล์ให้อยู่ในรูปแบบที่เหมาะสมกับระบบทราฟฟิกแบบต่อเนื่องที่ใช้นั้น จำเป็นต้องใช้การเข้ารหัสและบีบอัดข้อมูลให้ได้รูปแบบดังกล่าวโดยประโยชน์ของการเข้ารหัสนั้น นอกจากจะเพื่อให้เหมาะสมกับลักษณะการส่งข้อมูลของระบบทราฟฟิกแบบต่อเนื่องนั้น ๆ แล้ว ยังช่วยให้สามารถลดปริมาณแบนด์วิทด์ที่ใช้งานได้อีกด้วย ซึ่งปัจจุบันโปรแกรมหรืออุปกรณ์การเข้ารหัสข้อมูลมัลติมีเดียต่าง ๆ จะรวมการบีบอัดข้อมูลเข้าไปด้วยเสมอ โดยวิธีการบีบอัดข้อมูลนั้นมีอยู่ด้วยกันหลัก ๆ 2 ประเภท [11]

- การบีบอัดข้อมูลแบบ Spatial หรือ Intra-Frame : จะทำการขจัดความซ้ำซ้อนของข้อมูลพิคเซลในแต่ละเฟรม ด้วยการใช้การระบุความเหมือนของพิคเซลในแต่ละพื้นที่ของเฟรมนั้น ๆ ซึ่ง

ช่วยให้ขนาดของไฟล์ข้อมูลเล็กลงได้ แต่การบีบอัดข้อมูลลักษณะนี้จะทำให้ภาพมีคุณภาพลดลงตามอัตราการบีบอัดที่สูงขึ้น

- การบีบอัดข้อมูลแบบ Temporal หรือ Inter-Frame : ลักษณะการบีบอัดข้อมูลนี้จะใช้ความแตกต่างระหว่างเฟรมข้อมูลมาช่วย โดยใช้ข้อมูลของเฟรมแรกมาเป็นเฟรมข้อมูลอ้างอิง ซึ่งจะเรียกว่า Key-Frame และข้อมูลในเฟรมต่อ ๆ ไปจะใช้การบันทึกข้อมูลของเฟรมที่มีการเปลี่ยนแปลงไปจาก Key-Frame เท่านั้น และต่อจากนั้นจะมีการแทรก Key-Frame เป็นช่วง ๆ เพื่อใช้อ้างอิงต่อ ๆ ไป โดยคุณภาพของไฟล์จะขึ้นอยู่กับจำนวนของ Key-Frame ที่แทรกเข้าไป ซึ่งหากไฟล์มีจำนวน Key-Frame มากจะทำให้คุณภาพของข้อมูลสูงขึ้นตามด้วย

ซึ่งการเข้ารหัสข้อมูลของไฟล์มัลติมีเดียแต่ละรูปแบบนั้น อ้างอิงใช้เทคโนโลยีการบีบอัดข้อมูลจากทั้ง 2 วิธีดังกล่าวนี้เป็นพื้นฐานและพัฒนาเพิ่มเติมให้แตกต่างกันไปตามแต่ผู้ผลิตแต่ละราย จากข้อมูลดังตารางที่ 2.3 [12] พบว่าไฟล์มัลติมีเดียในรูปแบบต่าง ๆ ส่วนใหญ่จะใช้การบีบอัดข้อมูลแบบ Temporal รวมทั้งไฟล์รูปแบบเรียลมีเดียด้วย เมื่อข้อมูลที่ถูกเข้ารหัสและบีบอัดถูกส่งไปยังฝ่ายรับแล้ว ฝ่ายรับต้องมีโปรแกรมที่ใช้ถอดรหัสข้อมูลในรูปแบบนั้น ๆ เพื่อนำมัลติมีเดียกลับมาแสดงผลต่อไปได้

ตารางที่ 2.3 เทคโนโลยีการบีบอัดข้อมูลของไฟล์มัลติมีเดียแต่ละประเภท

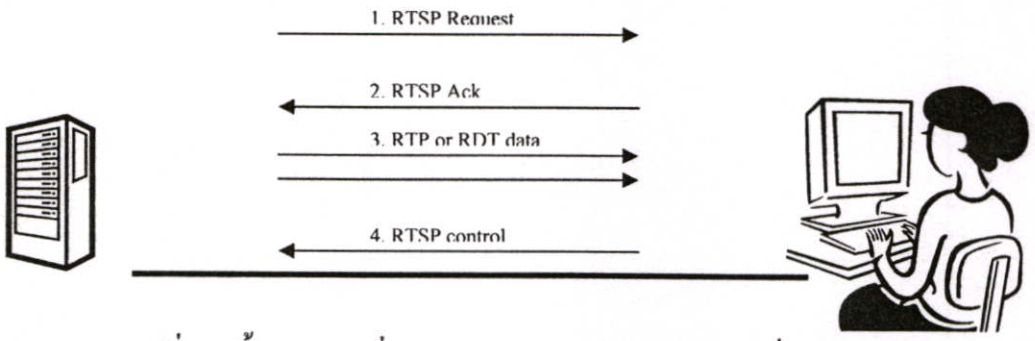
File Format	**Compression**
Advanced Streaming Format (ASF)	Temporal
Audio Video Interleave (AVI)	Temporal
MPEG-1	Temporal
MPEG-2	Temporal
QuickTime (QT)	Temporal
QuickTime Pro	Temporal
RealMedia (RM)	Temporal
Windows Media Video (WMV)	Temporal
DivX	Temporal
MJPEG	Spatial

2.6 โครงสร้างการทำงานของระบบทราฟฟิกแบบต่อเนื่องโดยใช้เรียลมีเดีย

จากเทคโนโลยีทราฟฟิกแบบต่อเนื่องในปัจจุบันที่มีหลายประเภทด้วยกัน การทดลองในวิทยานิพนธ์นี้ได้อ้างอิงการทำงานของโปรแกรมเรียลมีเดีย [4] ซึ่งเป็นระบบให้บริการทราฟฟิกแบบต่อเนื่องที่ถูกออกแบบมาเพื่อรองรับการเผยแพร่มัลติมีเดียซึ่งได้รับความนิยมมากในปัจจุบัน นอกจากนี้โปรแกรมเรียลมีเดียยังมีลักษณะการส่งข้อมูลโดยใช้โปรโตคอลยูดีพี ซึ่งทำให้สามารถวิเคราะห์พฤติกรรมของทราฟฟิกได้โดยปราศจากปัจจัยที่เกี่ยวเนื่องจากการจัดการทราฟฟิกของโปรโตคอลทีซีพี

โปรแกรมเรียลมีเดียเป็นระบบที่ถูกออกแบบมาเพื่อรองรับการใช้งานทราฟฟิกมีโครงสร้างและองค์ประกอบในการใช้งานดังนี้ เรียลเซิร์ฟเวอร์ (RealServer) เป็นเซิร์ฟเวอร์ที่สนับสนุนการเผยแพร่ข้อมูลวีดีโอหรือออดิโอทั้งรูปแบบทราฟฟิกแบบต่อเนื่องทั้งประเภทออนดีมานด์และเรียลไทม์ ซึ่งเรียลมีเดียสนับสนุนไฟล์มัลติมีเดียหลายรูปแบบ เช่น MPEG, RM, MP3, WMV เป็นต้น โดยทั่วไปไฟล์เรียลมีเดียสามารถจัดเตรียมได้ด้วยโปรแกรมเรียลโปรดิวเซอร์ (RealProducer) ซึ่งจะช่วยแปลงข้อมูลจากไฟล์ประเภทอื่น ๆ มาเป็นเรียลมีเดียไฟล์และสามารถเลือกอัตราการนำส่งข้อมูล (อัตราการเข้ารหัส) หรือการแสดงผลข้อมูลแบบต่อเนื่องที่ได้ตามต้องการ เมื่อได้ไฟล์เรียลมีเดียแล้ว ขั้นตอนต่อไปคือนำไฟล์ดังกล่าวไปจัดเก็บบนเรียลเซิร์ฟเวอร์และเผยแพร่ออกมาในรูปแบบทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์หรือเรียลไทม์ได้ตามต้องการ

สำหรับการแสดงผลนั้นผู้ใช้งานต้องติดตั้งโปรแกรมเรียลเพลเยอร์ (RealPlayer) เพื่อเรียกใช้ข้อมูล โดยทั่วไปแล้วการเรียกใช้ข้อมูลสามารถเรียกใช้ผ่านเว็บโดยโปรโตคอลเอชทีทีพี (HTTP) หรือโปรโตคอลอาร์ทีเอสพี (Real-Time Streaming Protocol) [5] ได้ ซึ่งโปรโตคอลอาร์ทีเอสพีเป็นโปรโตคอลประเภททีซีพีที่ถูกออกแบบมาเพื่อใช้ควบคุมการสื่อสารระหว่างผู้ใช้งานและผู้ให้บริการทราฟฟิกแบบต่อเนื่อง การทำงานของโปรโตคอลอาร์ทีเอสพีจะส่งข้อมูลควบคุมการเรียกใช้ เริ่ม หยุดและอื่น ๆ เพื่อสื่อสารระหว่างเรียลเซิร์ฟเวอร์และเรียลเพลเยอร์ (RealPlayer) หลังจากเรียกใช้ข้อมูลแล้วข้อมูลในเรียลเซิร์ฟเวอร์จะถูกจัดส่งข้อมูลโดยโปรโตคอลอาร์ทีพี (RTP) [6] หรืออาร์ดีที (RDT) อย่างใดอย่างหนึ่ง ซึ่งใช้บริการโปรโตคอลยูดีพีเพื่อส่งข้อมูลไปให้เรียลเพลเยอร์แสดงผลต่อไป ทั้งโปรโตคอลอาร์ทีพีและอาร์ดีทีถูกออกแบบมาเพื่อรองรับการส่งข้อมูลทราฟฟิกแบบต่อเนื่องโดย เฉพาะ รวมทั้งยังสนับสนุนการส่งข้อมูลแบบมัลติคาสท์ ซึ่งเป็นรูปแบบการใช้งานที่มีขึ้นเพื่อช่วยลดปริมาณของการส่งทราฟฟิกให้กับผู้ใช้บริการน้อยลง ในขณะที่ส่งข้อมูลอยู่นั้นเรียลเซิร์ฟเวอร์และเรียลเพลเยอร์จะสื่อสารกันด้วยโปรโตคอลอาร์เอสทีพีเพื่อตรวจสอบสถานะระหว่างกัน ขั้นตอนการทำงานเป็นดังรูปที่ 2.2



รูปที่ 2.2 ขั้นตอนการสื่อสารของระบบทราฟฟิกแบบต่อเนื่อง

2.7 คุณลักษณะเฉพาะตัวของทราฟฟิกแบบต่อเนื่อง

ทราฟฟิกแบบต่อเนื่องนั้นมีลักษณะและพฤติกรรมการทำงานเฉพาะตัวที่แตกต่างกับอินเทอร์เน็ตทราฟฟิกประเภทอื่น ๆ เช่น เว็บ อีเมล์ หรือเอฟทีพี โดยทั่วไปแล้วทราฟฟิกแบบต่อเนื่องสนับสนุนการส่งข้อมูลด้วยโปรโตคอลเอชทีทีพี ทีซีพีและยูดีพี ขึ้นอยู่กับช่วงของการทำงาน ในส่วนของการส่งข้อมูลนั้นจะส่งด้วยยูดีพีแพ็กเกตเป็นหลักมากถึง 70 – 80% ของทราฟฟิกทั้งหมดในการสื่อสารแต่ละครั้งและใช้ทีซีพีแพ็กเกตในการส่งสถานะหรือคำสั่งต่าง ๆ เช่น การเรียกใช้งาน หรือหยุดการแสดงผล เป็นต้น [7] การส่งข้อมูลระหว่างผู้ให้บริการและผู้ใช้บริการนั้นข้อมูลส่วนใหญ่จะส่งออกมาจากผู้ให้บริการมากกว่าโดยมีอัตราสูงถึง 50:1 ซึ่งปกติแล้วการสื่อสารประเภททราฟฟิกแบบต่อเนื่องจะใช้เวลาค่อนข้างยาวนานกว่าทราฟฟิกประเภทอื่น ๆ เนื่องจากมีการแสดงผลของข้อมูลที่ยาวกว่าทราฟฟิกประเภทเว็บหรือเอฟทีพี [7]

จากผลการวิจัยเชิงการสังเกตพฤติกรรมการส่งข้อมูลของทราฟฟิกแบบต่อเนื่อง [2] [7] ได้อธิบายถึงคุณสมบัติของทราฟฟิกแบบต่อเนื่องด้วยการศึกษาพฤติกรรมจากทราฟฟิกประเภทเรียลออดิโอ (RealAudio) จากการใช้งานในระบบเครือข่ายจริงพบว่า หากพิจารณารูปแบบของทราฟฟิกแบบต่อเนื่องโดยภาพรวมนั้นทราฟฟิกแบบต่อเนื่องจะส่งข้อมูลด้วยโปรโตคอลยูดีพีมากกว่าทีซีพีและเป็นรูปแบบอัตราการส่งคงที่ (Constant Bit Rate) แต่เมื่อพิจารณาข้อมูลเดียวกันนั้นโดยละเอียดจะเห็นว่ารูปแบบการส่งข้อมูลจะมีลักษณะเป็นช่วง ๆ ซึ่งเป็นรูปแบบ ON-OFF [7]

ในการวิจัยของ Kuang, Tianbo, Williamson, Carey L [2] ซึ่งแสดงให้เห็นถึงการเข้ารหัสข้อมูลวีดีโอในรูปแบบเรียลมีเดียนั้นจะแยกข้อมูลออกมาเป็น 2 ส่วนคือส่วนที่เป็นข้อมูลภาพกับส่วนที่เป็นข้อมูลเสียง โดยปกติข้อมูลส่วนที่เป็นภาพนั้นจะมีขนาดสูงกว่าและมีอัตราการนำส่งข้อมูลต่อเฟรมค่อนข้างผันผวนทั้งทางด้านจำนวนแพ็กเกตต่อเฟรมและขนาดแพ็กเกตในแต่ละเฟรม ต่างจากข้อมูลเสียงซึ่งมีขนาดค่อนข้างคงที่มากกว่า ผลจากการวิจัยดังกล่าวนี้ได้อธิบายถึงพฤติกรรมการส่งข้อมูลรูปแบบ ON-OFF ของงานวิจัยข้างต้นไว้ว่า การเกิดลักษณะ ON-OFF ดังกล่าวเกิดจากการที่เซิร์ฟเวอร์ส่งแพ็กเกตที่เป็นข้อมูลเสียงด้วยอัตราการส่งที่คงที่ในช่วงเวลาหนึ่งแล้ว จะหยุดส่งข้อมูลเสียงและเริ่มส่งแพ็กเกตข้อมูลภาพซึ่งมีขนาดที่ใหญ่ขึ้น โดยลักษณะการส่งข้อมูลดังกล่าวนี้เกิดขึ้นซ้ำ ๆ เป็นช่วงจึงทำให้เห็นว่าการส่งข้อมูลเป็นรูปแบบ ON-OFF

ซึ่งในวิทยานิพนธ์เล่มนี้ได้ศึกษาต่อเนื่องใน 2 ส่วนหลัก ๆ ดังนี้คือ ความแตกต่างกันระหว่างพฤติกรรมการส่งข้อมูลของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์กับเรียลไทม์ทราฟฟิกและความสัมพันธ์ระหว่างพฤติกรรมการส่งข้อมูลของทราฟฟิกแบบต่อเนื่องกับอัตราการนำส่งข้อมูล

2.8 สรุปท้ายบท

บทนี้กล่าวถึงทราฟฟิกอินเทอร์เน็ตที่มีการใช้งานและประเภทของทราฟฟิกต่าง ๆ ซึ่งประกอบไปด้วยทราฟฟิกหลาย ๆ ประเภทรวมถึงทราฟฟิกแบบต่อเนื่องที่นำมาศึกษาในวิทยานิพนธ์เล่มนี้ โดยทราฟฟิกแบบต่อเนื่องนั้นเป็นส่วนหนึ่งของทราฟฟิกประเภทมัลติมีเดีย

ทราฟฟิกแบบต่อเนื่องนั้น สามารถแบ่งประเภทได้ตามวัตถุประสงค์การใช้งานเป็น 2 ประเภทใหญ่ ๆ ด้วยกันประกอบด้วยทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์และออนดีมานด์ ซึ่งทั้ง 2 ประเภทนี้ยังมีความแตกต่างกันในพฤติกรรมการส่งข้อมูลอีกด้วย ดังผลการทดลองในบทต่อไป

บทที่ 3
การทดลองเพื่อระบุคุณลักษณะของทราฟฟิกแบบต่อเนื่อง

จากความสนใจในการศึกษารูปแบบของทราฟฟิกแบบต่อเนื่อง ดังนั้นในบทนี้ได้อธิบายถึงขั้นตอนการทดลอง และวิเคราะห์ผลการทดลองเพื่อให้ได้รูปแบบของทราฟฟิกแบบต่อเนื่องที่ถูกต้องสำหรับนำไปสร้างแบบจำลองของทราฟฟิกแบบต่อเนื่องต่อไป โดยในการวิเคราะห์ผลการทดลองนี้ใช้การวิเคราะห์ทั้งการกระจายของขนาดแพ็กเกตและช่วงเวลาระยะห่างการส่งแต่ละแพ็กเกต นอกจากนั้นยังวิเคราะห์ความแตกต่างของพฤติกรรมทราฟฟิกแบบต่อเนื่องทั้งประเภทเรียลไทม์และออนดีมานด์ รวมถึงความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลกับพฤติกรรมการส่งข้อมูลด้วย

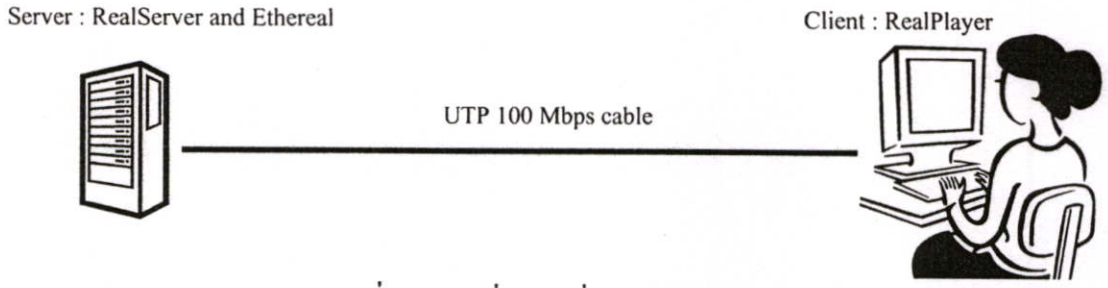
3.1 การทดลองทราฟฟิกแบบต่อเนื่อง

3.1.1 สมมติฐานการทดลอง

เนื่องจากสมมติฐานที่มีต่อระบบเครือข่ายในอนาคตที่ถูกพัฒนาขึ้นเพื่อรองรับความต้องการใช้งานอินเทอร์เน็ตแบนด์วิดท์อย่างเพียงพอ โปรแกรมต่าง ๆ ที่ใช้งานในระบบอินเทอร์เน็ตสามารถสื่อสารกับระบบเครือข่ายเพื่อจองพื้นที่แบนด์วิดท์ได้ตามต้องการ ดังนั้นการจัดการหรือควบคุมความคับคั่งของการรับส่งข้อมูลด้วยโปรโตคอลชั้นนำส่งข้อมูลไม่มีความจำเป็นอีกต่อไปในอนาคต ดังนั้นการทดลองนี้จึงอ้างอิงการทดลองกับระบบเรียลมีเดียที่สามารถส่งข้อมูลด้วยโปรโตคอลยูดีพี ทำให้สามารถวิเคราะห์ถึงพฤติกรรมการส่งข้อมูลที่เกิดขึ้นจริงจากระบบทราฟฟิกแบบต่อเนื่องนั้น ๆ โดยไม่มีผลกระทบจากการควบคุมความคับคั่งจากโปรโตคอลทีซีพี

3.1.2 แนวทางการทดลอง

ในการทดลองนี้ทดสอบขึ้นในห้องวิจัย โดยใช้โปรแกรมเรียลเซิร์ฟเวอร์เพื่อทำการแพร่ภาพข้อมูลประเภทวิดีโอเนื่องจากที่เป็นโปรแกรมที่ติดตั้งใช้งานได้ง่าย จึงเป็นที่นิยมอย่างมากสำหรับผู้ที่จะติดตั้งเซิร์ฟเวอร์ที่ให้บริการวีดีโอและออดิโอในปัจจุบัน สำหรับการเก็บข้อมูลที่ส่งเพื่อนำมาวิเคราะห์นั้นได้ใช้โปรแกรมดักจับข้อมูลอีเทอร์เรียล (Ethereal) ที่เป็นโปรแกรมจับข้อมูลแพ็กเกตที่ถูกส่งหรือรับผ่านระบบเครือข่าย โดยสามารถดูรายละเอียดต่าง ๆ ของแพ็กเกตและแสดงผลออกมาในรูปแบบสถิติได้



รูปที่ 3.1 การเชื่อมต่อที่ใช้ในการทดลอง

สำหรับลักษณะการเชื่อมต่อในการทดลองนี้จะเชื่อมต่อเรียลเซิร์ฟเวอร์ตรงไปยังเครื่องฝ่ายรับข้อมูลที่ติดตั้งโปรแกรมเรียลเพลเยอร์สำหรับแสดงผลไฟล์เรียลมีเดียไว้ดังรูปที่ 3.1 เพื่อหลีกเลี่ยงการเกิดความผิดเพี้ยนของข้อมูลจากปัจจัยรอบข้างต่าง ๆ จากระบบเครือข่าย เช่น ความหนาแน่นของการใช้งานระบบเครือข่าย โดยในฝั่งเครื่องเรียลเซิร์ฟเวอร์นั้นได้ติดตั้งโปรแกรมจับข้อมูลเพื่อเก็บผลการส่งข้อมูลและนำมาวิเคราะห์ต่อไป คุณสมบัติของอุปกรณ์ที่ใช้ในการทดลองนี้เป็นดังตารางที่ 3.1

ตารางที่ 3.1 คุณสมบัติของเครื่องที่ใช้ในการทดลอง

Device	Specification	Software	Connection
Server	PC : Pentium 4 2.0 GHz, RAM 384 MByte	RealServer, Ethereal	UTP 100 Mbps
Client	LAPTOP : Pentium 4 2.0 GHz, RAM 384 MByte	RealPlayer	UTP 100 Mbps

แนวทางการทดลองนั้นแบ่งการทดลองออกเป็น 2 ส่วนคือในส่วนของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์และเรียลไทม์ทราฟฟิก ผลของการทดลองจะแสดงรายละเอียดของการกระจายของทราฟฟิกและนำมาวิเคราะห์ข้อมูล การวิเคราะห์ข้อมูลทราฟฟิกได้วิเคราะห์ทั้งการกระจายของขนาดแพ็กเกตที่ถูกส่งและการกระจายของช่วงระยะห่างในการส่งข้อมูลแต่ละแพ็กเกตเพื่อนำไปสร้างแบบจำลองของทราฟฟิกแบบต่อเนื่องต่อไป

กลุ่มตัวอย่างวีดีโอที่ใช้ทดลองนั้น เป็นไฟล์วีดีโอต้นฉบับความยาว 5 นาทีและ 10 นาที โดยเลือกทดลองใช้ตัวอย่างวีดีโอที่มีเนื้อหาและลักษณะการนำเสนอที่แตกต่างกันดังต่อไปนี้

วีดีโอแอคชั่น	เรื่องสเตลท์และแบดบอย	ความยาว 5 นาที
วีดีโอทอล์คโชว์	เรื่องเดี่ยวไมโครโฟน 3 และ 4	ความยาว 5 นาที
วีดีโอดราม่า	เรื่องชิคาโกและซีโมน	ความยาว 10 นาที
วีดีโอสารคดี	เรื่องเชื่อหรือไม่และตามรอยพระพุทธเจ้า	ความยาว 10 นาที

เนื่องจากกลุ่มตัวอย่าง 4 ประเภทนี้น่าจะมีความแตกต่างกันในเรื่องของการเปลี่ยนแปลงข้อมูลในแต่ละเฟรม โดยสำหรับประเภทแอคชั่นนั้นมีการเคลื่อนไหวมากกว่าประเภทอื่น ๆ ซึ่งมีผลให้การเปลี่ยนแปลงข้อมูลในพีเฟรมและบีเฟรมของแต่ละเฟรมมีค่าสูงกว่าเมื่อเทียบกับประเภทอื่น ๆ ที่มีการเคลื่อนไหวน้อยกว่า [8] สาเหตุที่นำกลุ่มตัวอย่าง 4 ประเภทที่แตกต่างกันนี้มาทดลองก็เพื่อนำข้อมูลที่มีความแตกต่างกันของเนื้อหาข้อมูลมาทดลองให้ได้ผลการทดลองที่ครอบคลุมในเงื่อนไขต่าง ๆ หลังจากนั้นนำกลุ่มตัวอย่างดังกล่าวมาเข้ารหัสข้อมูลด้วยอัตราการนำส่งข้อมูลต่าง ๆ กันด้วย 56 Kbps, 128 Kbps และ 512 Kbps ออกมาเป็นไฟล์เรียลมีเดียด้วยโปรแกรมเรียลโปรดิวเซอร์ ที่อัตรานำส่งข้อมูลทั้ง 3 อัตราแทนความเร็วการใช้งานอินเทอร์เน็ตที่นิยมใช้ในปัจจุบันคือ 56 Kbps แทนการใช้งานอินเทอร์เน็ตผ่านไดอัลอัพโมเด็ม (Dial-up modem) 128 Kbps และ 512 Kbps แทนการใช้งานผ่านระบบเอดีเอสแอล (ADSL) ด้วยอัตราที่นิยมใช้กันทั่วไปตามบ้าน โดยเผยแพร่ข้อมูลออกไปด้วยรูปแบบออนดีมานด์หรือเรียลไทม์ จากนั้นนำผลการทดลองที่ได้จากโปรแกรมจับข้อมูลมาเปรียบเทียบและวิเคราะห์ผลการทดลอง กลุ่มตัวอย่างที่เลือกใช้ในการทดลองนี้แสดงดังตารางที่ 3.2

ตารางที่ 3.2 กลุ่มตัวอย่างการทดลองและขนาดของไฟล์ตัวอย่างที่ผ่านการเข้ารหัสข้อมูล (MByte)

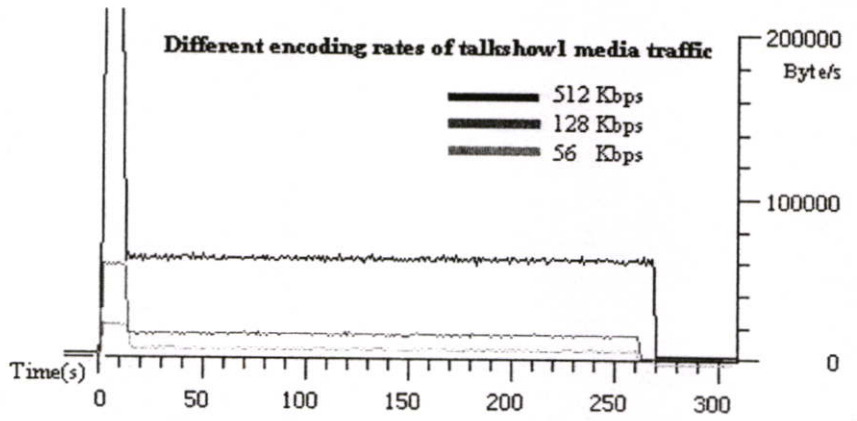
Movie Name	Type	Duration(Minute)	File size (MBbyte)		
			56 Kbps	128 Kbps	512 Kbps
Stealth – Action1	Action	5	1.28	3.72	16.5
Badboy – Action2	Action	5	1.29	3.79	16.7
Note3 – Talkshow1	Talkshow	5	1.26	3.67	16.2
Note4 – Talkshow2	Talkshow	5	1.26	3.64	16.1
Chicago – Drama1	Drama	10	2.52	7.37	32.5
Simone – Drama2	Drama	10	2.58	7.43	32.9
Unbelieve – Documentary1	Documentary	10	2.54	7.35	32.6
Buddism – Documentary2	Documentary	10	2.54	7.37	32.7

3.2 ผลและการวิเคราะห์ผลการทดลอง

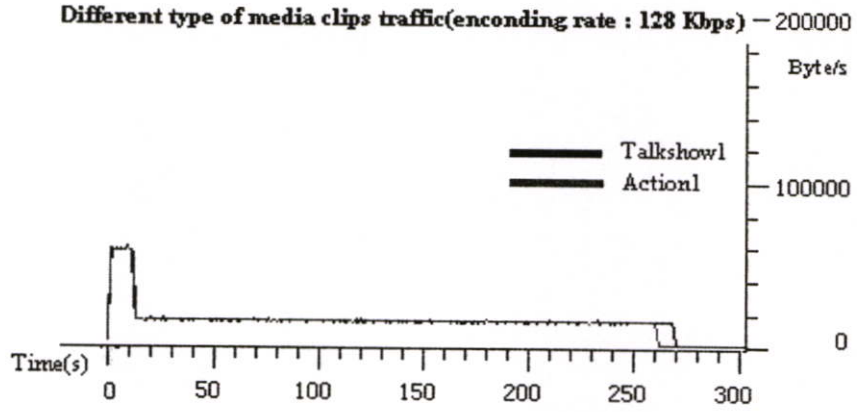
3.2.1 ผลการทดลองและการวิเคราะห์ทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

จากผลของการเก็บข้อมูลที่ได้จากโปรแกรมดักจับข้อมูล การวิเคราะห์ผลของข้อมูลดังกล่าว พบว่า ลักษณะพฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์นั้นเป็นดังรูปที่ 3.2 การส่งข้อมูลของเรียลเซิร์ฟเวอร์ในแบบออนดีมานด์นั้น จะเห็นได้ว่าช่วงการส่งข้อมูลนั้นมีความหนาแน่นมากในช่วงต้น ๆ ของการแสดงผลและหลังจากนั้นจะส่งข้อมูลอย่างต่อเนื่องที่อัตราการส่งต่ำลง แต่การส่งข้อมูลอย่างต่อเนื่องดังกล่าวไม่ได้เกิดขึ้นตลอดช่วงทั้งหมดของการแสดงผล ช่วงท้ายของการแสดงผลจะมีการส่งข้อมูลในอัตราการส่งที่ลดลง

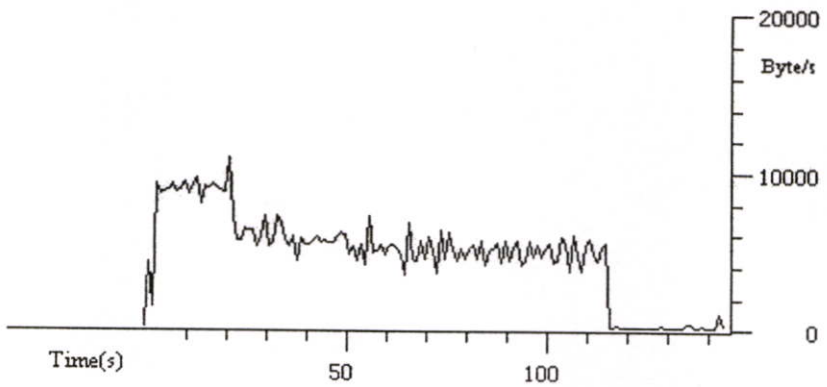
จากผลการทดลองในรูปที่ 3.2 จะเห็นได้ว่าการแสดงผลจริง ๆ ของไฟล์นั้นมีความยาวถึง 300 วินาที แต่สำหรับการส่งชุดข้อมูลนั้นมีการส่งอยู่ในช่วง 270 วินาทีเท่านั้น และเมื่อเทียบกับข้อมูลที่ได้จากการทดลองกับทราฟฟิกแบบต่อเนื่องที่ให้บริการจริงบนอินเทอร์เน็ต http://news.bbc.co.uk [8] ซึ่งเป็นวีดีโอคลิปข่าว ลักษณะของทราฟฟิกที่ได้เป็นดังรูปที่ 3.4 จะเห็นว่ามีลักษณะคล้ายกันกับผลการทดลองที่เกิดขึ้นในห้องวิจัยคือช่วงเวลาการแสดงผลจริงนั้นมีความยาว 138 วินาที แต่ช่วงการส่งข้อมูลนั้นมีเพียง 110 วินาทีเท่านั้น ซึ่งอาจสรุปได้ว่าการส่งข้อมูลของออนดีมานด์นั้นในช่วงแรกจะส่งข้อมูลจำนวนมาก ๆ เข้าสู่บัฟเฟอร์ก่อนที่จะเริ่มแสดงผล เพื่อให้มีข้อมูลเพียงพอในฝ่ายรับข้อมูลและการแสดงผลของข้อมูลดังกล่าวจะถูกควบคุมโดยฝ่ายรับเอง หลังจากนั้นจึงจะส่งข้อมูลอย่างต่อเนื่องเข้าสู่บัฟเฟอร์ต่อไป เมื่อการนำส่งไฟล์ได้เสร็จเรียบร้อยแล้ว การนำส่งข้อมูลหลังจากนั้นระหว่างทั้ง 2 ฝ่ายเป็นเพียงการตรวจสอบสถานะระหว่างกันเท่านั้น ซึ่งเป็นแพ็กเกตเล็ก ๆ ประมาณ 50 ไบต์เป็นโปรโตคอลอาร์ทีเอสพีเพื่อตรวจสอบสถานะซึ่งกันและกันต่อไป จนกว่าฝ่ายรับจะหยุดการแสดงผลหรือการให้บริการข้อมูลสิ้นสุดลง พฤติกรรมการส่ง-รับข้อมูลลักษณะนี้จะขึ้นอยู่กับประสิทธิภาพและขนาดบัฟเฟอร์ของเครื่องฝ่ายรับด้วย สำหรับขั้นตอนการสื่อสารในส่วนของการเลื่อนการแสดงผลไปข้างหน้าหรือย้อนหลังนั้น ฝ่ายรับจะส่งสัญญาณด้วยโปรโตคอลอาร์ทีเอสพีไปยังเรียลเซิร์ฟเวอร์เพื่อขอเปลี่ยนช่วงข้อมูลในการแสดงผล เมื่อเรียลเซิร์ฟเวอร์ได้รับการร้องขอดังกล่าว เรียลเซิร์ฟเวอร์จึงจะเริ่มส่งข้อมูลใหม่ ณ ตำแหน่งที่ต้องการนั้น ๆ ไปให้ฝ่ายรับต่อไป



รูปที่ 3.2 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบออนดีมานด์ด้วยอัตราการนำส่งต่าง ๆ กัน



รูปที่ 3.3 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบออนดีมานด์ด้วยประเภทของตัวอย่างไฟล์มัลติมีเดียที่ต่างประเภทกัน



รูปที่ 3.4 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบออนดีมานด์โดยเวลาแสดงผลจริงของไฟล์มีความยาว 138 วินาที จาก http://news.bbc.co.uk

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หากพิจารณาข้อมูลโดยละเอียดของความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลและลักษณะการส่งข้อมูลทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ จะเห็นได้ว่าอัตราการนำส่งข้อมูลจะมีผลโดยตรงกับขนาดแพ็กเกตและช่วงเวลาระยะห่างในการส่งแพ็กเกต จากตารางที่ 3.3 แสดงให้เห็นว่าขนาดเฉลี่ยของแพ็กเกตที่ถูกส่งออกไปจะมีค่าสูงขึ้นเมื่อไฟล์ที่เผยแพร่มีอัตราการนำส่งข้อมูลที่สูงขึ้น ในทางกลับกันช่วงเวลาระยะห่างเฉลี่ยของการส่งข้อมูลแต่ละแพ็กเกตจะมีค่าลดลงดังตารางที่ 3.5 ซึ่งทำให้ส่งข้อมูลได้สูงขึ้นตามอัตราการนำส่งที่สูงขึ้น สำหรับขนาดแพ็กเกตสูงสุดของแต่ละอัตราการนำส่งจะมีค่าเท่ากันทั้งหมดและมีค่าสูงขึ้นในอัตราการนำส่งที่สูงขึ้น ซึ่งสามารถสรุปได้ว่าค่าสูงสุดของขนาดแพ็กเกตจะถูกกำหนดด้วยอัตราการนำส่ง

ตารางที่ 3.3 ความสัมพันธ์ของอัตราการนำส่งข้อมูลกับค่าเฉลี่ยของขนาดแพ็กเกต(Byte/packet) ระหว่างวีดีโอประเภทต่าง ๆ กันในทราฟฟิกประเภทออนดีมานด์

Movie Name	Average packet Size (Byte/packet)		
	56Kbps	128Kbps	512Kbps
Action1	356.2439	421.1875	841.5653
Action2	395.5455	467.2739	844.3776
Talkshow1	382.2839	425.6566	785.7644
Talkshow2	307.778	439.5985	834.7268
Drama1	347.9745	429.4882	817.9952
Drama2	295.5692	438.9251	810.8021
Documentary1	322.0526	446.2059	834.6358
Documentary2	323.2544	453.5573	823.4596

63624

ตารางที่ 3.4 ความสัมพันธ์ของอัตราการนำส่งข้อมูล (Kbps) กับค่าสูงสุดของขนาดแพ็กเกต (Byte/packet) ระหว่างวีดีโอประเภทต่าง ๆ กันในทราฟฟิกประเภทออนดีมานด์

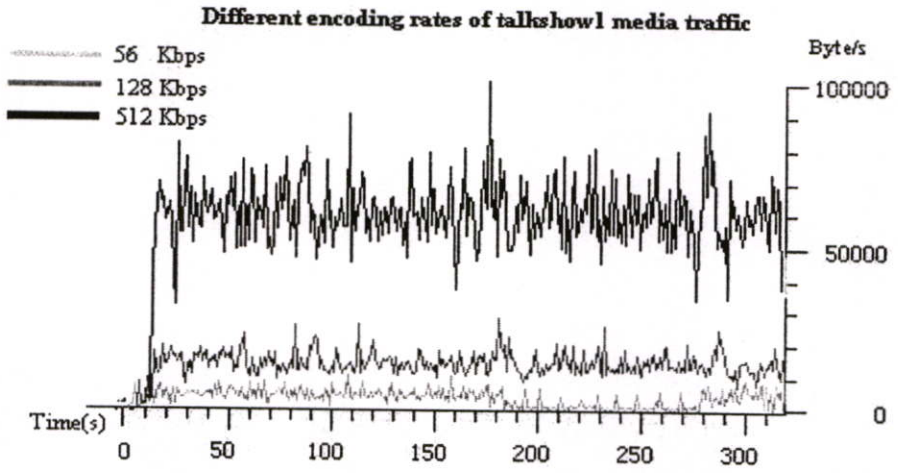
Movie Name	Maximum packet Size (Byte/packet)		
	56Kbps	128Kbps	512Kbps
Action1	659	661	1461
Action2	659	661	1461
Talkshow1	659	661	1461
Talkshow2	659	661	1461
Drama1	659	661	1461
Drama2	659	661	1461
Documentary1	659	661	1461
Documentary2	659	661	1461

ตารางที่ 3.5 ความสัมพันธ์ของอัตราการนำส่งข้อมูล (Kbps) กับค่าเฉลี่ยของช่วงเวลาการส่งแต่ละแพ็กเกต (s/packet) ระหว่างวีดีโอประเภทต่าง ๆ กันในทราฟฟิกประเภทออนดีมานด์

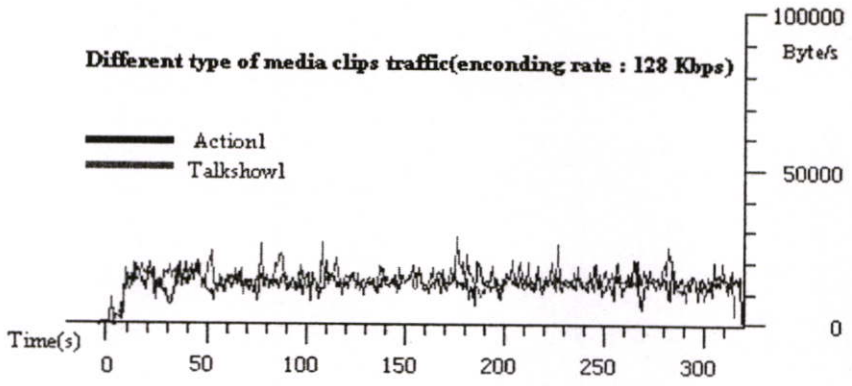
Movie Name	Average packet interval time (s/packet)		
	56Kbps	128Kbps	512Kbps
Action1	0.063156	0.026187	0.012644
Action2	0.070224	0.029238	0.012726
Talkshow1	0.066325	0.026007	0.011683
Talkshow2	0.052341	0.027594	0.012346
Drama1	0.070381	0.03081	0.013891
Drama2	0.057303	0.031476	0.013727
Documentary1	0.063759	0.032123	0.014292
Documentary2	0.063479	0.032766	0.01404

3.2.2 ผลการทดลองและการวิเคราะห์ทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

การส่งข้อมูลของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์จะแบ่งขั้นตอนการส่งข้อมูล โดยเริ่มจากการสื่อสารด้วยข้อมูลควบคุมเพื่อตกลงลักษณะการนำส่ง และหลังจากนั้นจึงส่งข้อมูลมัลติมีเดียที่จะแสดงผลเช่นเดียวกับการส่งแบบออนดีมานด์ แต่ในส่วนของการส่งข้อมูลนั้นจะเห็นว่ามีการส่งอย่างต่อเนื่องตลอดช่วงของการแสดงผล จนกว่าจะหยุดรับข้อมูลดังรูปที่ 3.5 ซึ่งแตกต่างกันกับการส่งทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ที่จะส่งข้อมูลเข้าสู่บัฟเฟอร์ของฝ่ายรับด้วยอัตราการส่งที่สูงมากในช่วงแรก นอกจากนี้เมื่อวิเคราะห์ข้อมูลทางด้านการกระจายของขนาดแพ็กเกตและช่วงเวลาระยะห่างการส่งแพ็กเกตเฉลี่ย พบว่าความสัมพันธ์ระหว่างการกระจายของขนาดแพ็กเกตจะเพิ่มขึ้นเมื่ออัตราการนำส่งข้อมูลเพิ่มขึ้นดังตารางที่ 3.6 และข้อมูลที่ถูกเข้ารหัสด้วยอัตราการนำส่งที่เท่ากันจะมีขอบเขตของขนาดแพ็กเกตสูงสุดเท่ากันดังตารางที่ 3.7 และช่วงเวลาการส่งข้อมูลของแพ็กเกตจะลดลงเมื่ออัตราการนำส่งข้อมูลเพิ่มขึ้นดังตารางที่ 3.8 ซึ่งหมายความว่าปริมาณการส่งข้อมูลจะสูงขึ้น โดยการเพิ่มขึ้นของขนาดแพ็กเกตและความถี่ของการส่งข้อมูลเช่นเดียวกันกับลักษณะของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์



รูปที่ 3.5 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบเรียลไทม์ด้วยอัตราการนำส่งต่าง ๆ กัน



รูปที่ 3.6 พฤติกรรมการส่งข้อมูลของเรียลเซิร์ฟเวอร์แบบเรียลไทม์ด้วยประเภทของตัวอย่างมัลติมีเดียที่ต่างกัน

ตารางที่ 3.6 ความสัมพันธ์ของอัตราการนำส่งข้อมูล (Kbps) กับค่าเฉลี่ยของขนาดแพ็กเกต (Byte/packet) ระหว่างวีดีโอประเภทต่าง ๆ กันของทราฟฟิกประเภทเรียลไทม์

Movie Name	Average packet Size (Byte/packet)		
	56Kbps	128Kbps	512Kbps
Action1	356.2565	419.4343	839.903
Action2	389.4876	465.1517	841.1983
Talkshow1	369.6748	408	834.6242
Talkshow2	306.69	420	789.991
Drama1	347.6989	430.5023	816.6093
Drama2	297.9385	430.8446	809.6411
Documentary1	325.584	446.5323	834.4902
Documentary2	323.2495	454.4883	822.3516

ตารางที่ 3.7 ความสัมพันธ์ของอัตราการนำส่งข้อมูล (Kbps) กับค่าสูงสุดของขนาดแพ็กเกต (Byte/packet) ระหว่างวีดีโอประเภทต่าง ๆ กันของทราฟฟิกประเภทเรียลไทม์

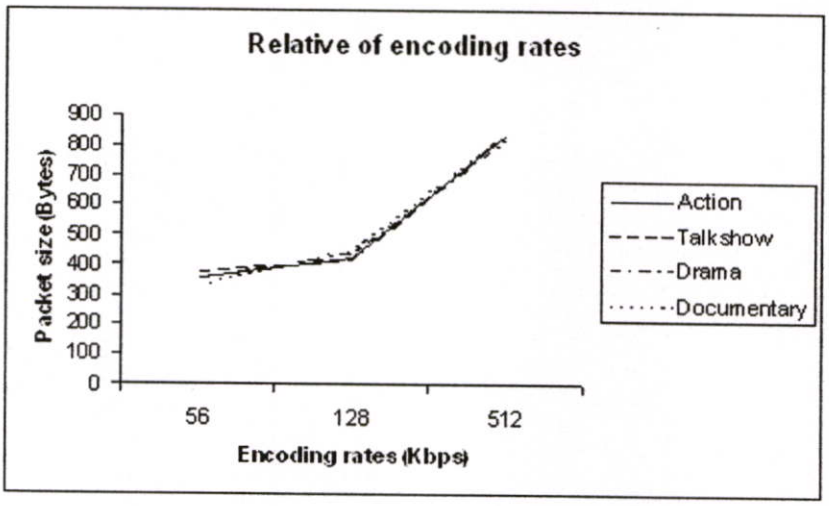
Movie Name	Maximum packet Size (Byte/packet)		
	56Kbps	128Kbps	512Kbps
Action1	659	661	1461
Action2	659	661	1461
Talkshow1	659	661	1452
Talkshow2	658	661	1461
Drama1	659	661	1461
Drama2	659	661	1461
Documentary1	659	661	1461
Documentary2	659	661	1461

ตารางที่ 3.8 ความสัมพันธ์ของอัตราการนำส่งข้อมูล (Kbps) กับค่าเฉลี่ยของช่วงเวลาการส่งแต่ละแพ็กเกต(s/packet) ระหว่างวีดีโอประเภทแอคชั่นและทอล์คโชว์

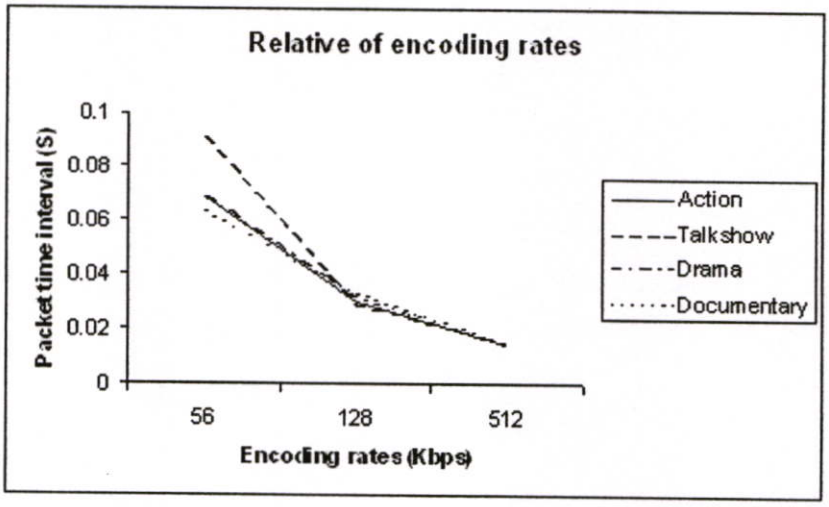
Movie Name	Average packet interval time (s/packet)		
	56Kbps	128Kbps	512Kbps
Action1	0.069	0.0297	0.0142
Action2	0.078	0.0329	0.0142
Talkshow1	0.0905	0.0285	0.01419
Talkshow2	0.0586	0.033	0.013
Drama1	0.068827	0.031249	0.013826
Drama2	0.057279	0.03206	0.013669
Documentary1	0.063117	0.032513	0.0141
Documentary2	0.062852	0.033258	0.013988

3.2.3 ความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลกับพฤติกรรมการส่งข้อมูล

จากการวิเคราะห์ผลข้อมูลที่ได้ดังในตารางที่ 3.6 และ 3.8 พบว่าความสัมพันธ์ของอัตราการนำส่งข้อมูลนั้นมีผลโดยตรงกับพฤติกรรมการส่งข้อมูล โดยเมื่ออัตราการนำส่งข้อมูลของไฟล์วีดีโอที่จะเผยแพร่มีค่าสูงขึ้น ค่าเฉลี่ยของขนาดแพ็กเกตที่จะส่งจะมีขนาดสูงขึ้นตามและในขณะเดียวกันช่วงเวลาการส่งข้อมูลแต่ละแพ็กเกตจะสั้นลงด้วย ดังกราฟในรูปที่ 3.7 และ 3.8



รูปที่ 3.7 ความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลกับค่าเฉลี่ยของขนาดแพ็กเกต



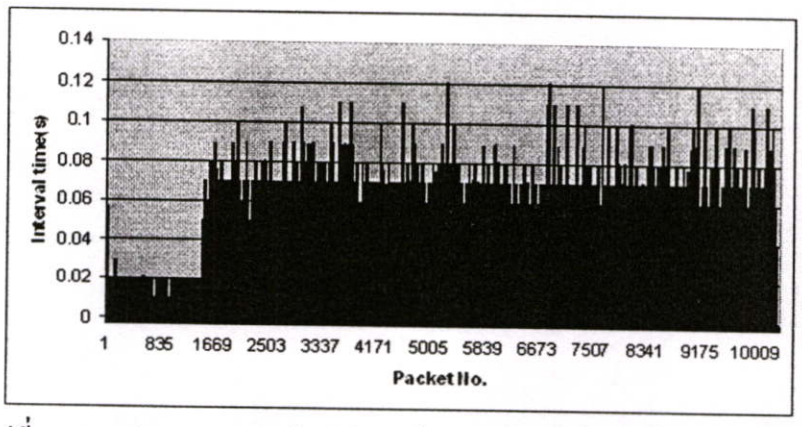
รูปที่ 3.8 ความสัมพันธ์ระหว่างอัตราการนำส่งข้อมูลกับช่วงเวลาเฉลี่ยของการส่งข้อมูลแต่ละแพ็กเกต

ซึ่งลักษณะดังกล่าวสามารถอธิบายได้โดย เมื่อต้องการคุณภาพของการรับชมที่ดีขึ้นก็จะต้องกำหนดให้อัตราการนำส่งข้อมูลสูงขึ้นซึ่งทำให้ปริมาณข้อมูลที่จะส่งสูงขึ้นตามไปด้วย ดังนั้นเซิร์ฟเวอร์จึงจัดการข้อมูลที่จะส่งด้วยขนาดแพ็กเกตและช่วงระยะห่างการส่งที่เหมาะสมกับข้อมูลแต่ละอัตราการส่ง ซึ่งวิธีการดังกล่าวเป็นกลไกที่เรียลเซิร์ฟเวอร์ใช้จัดการส่งข้อมูลให้ได้ตามความเหมาะสมที่อัตราการนำส่งต่าง ๆ กัน แต่ทั้งนี้ขนาดสูงสุดของแพ็กเกตที่จะถูกส่งออกมา จะถูกกำหนดด้วยอัตราการนำส่งข้อมูลนั้น ๆ เสมอดังตารางที่ 3.7

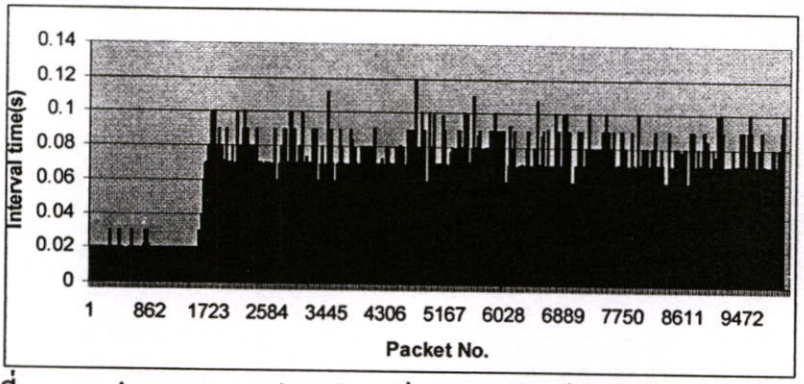
3.2.4 ความแตกต่างระหว่างการส่งข้อมูลของทราฟฟิกแบบต่อเนื่องออนดีมานด์กับเรียลไทม์

จากผลที่ได้จากการทดลอง สามารถวิเคราะห์ถึงความแตกต่างระหว่างทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์กับเรียลไทม์ได้ดังนี้ เมื่อพิจารณาภาพโดยรวมของลักษณะการส่งทราฟฟิกทั้ง 2 ประเภทนั้น จะเห็นลักษณะความแตกต่างคือการส่งข้อมูลของออนดีมานด์นั้นจะมีการส่งข้อมูลจำนวนมากในช่วงเริ่มต้น และจะลดปริมาณการนำส่งข้อมูลลงแต่ยังคงส่งข้อมูลแบบต่อเนื่องเข้าสู่บัฟเฟอร์ของฝ่ายรับเพื่อแสดงผลต่อไป ซึ่งการส่งข้อมูลนั้นจะไม่ส่งไปตลอดช่วงของการแสดงผล ดังรูปที่ 3.2 กราฟของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ จากการแสดงผลจริง 300 วินาทีนั้น จะมีช่วงการส่งข้อมูลอยู่เพียง 270 -280 วินาที หลังจากนั้นเรียลเซิร์ฟเวอร์และฝ่ายรับจะส่งแพ็กเกตสื่อสารสถานะกันเท่านั้น ซึ่งถ้าพิจารณาภาพเดียวกันของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์จะเห็นว่าความสม่ำเสมอของข้อมูลในการส่งเป็นไปอย่างต่อเนื่องตลอดช่วงเวลาการแสดงผลดังรูปที่ 3.5

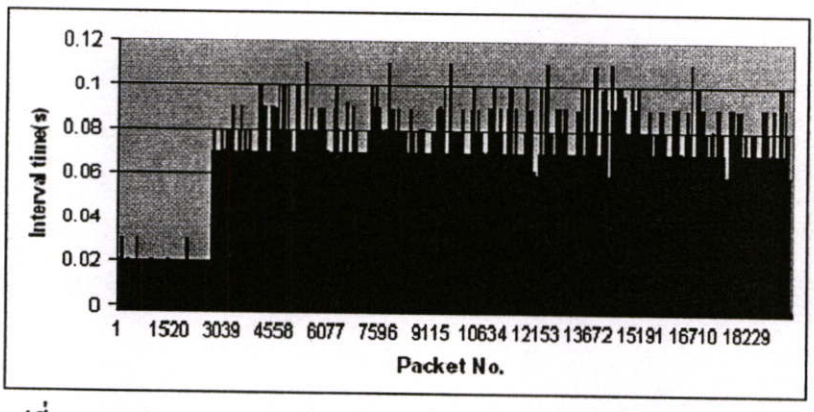
ต่อมาเมื่อนำข้อมูลมาวิเคราะห์ถึงพฤติกรรมการส่งแพ็กเกตทั้งในเรื่องของการกระจายของขนาดแพ็กเกตที่ส่งและการกระจายของช่วงเวลาห่างในการส่งแต่ละแพ็กเกต พบว่าความแตกต่างที่สำคัญนั้นอยู่ที่การกระจายของช่วงระยะห่างในการส่งแพ็กเกตดังรูปที่ 3.9 และ 3.10 แสดงค่าช่วงเวลาระยะห่างระหว่างการส่งแต่ละแพ็กเกต เมื่อพิจารณาลักษณะการส่งข้อมูลของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ จะเห็นว่าช่วงการส่งข้อมูลช่วงเริ่มต้นมีช่วงเวลาห่างระหว่างแพ็กเกตที่สั้นมาก ทำให้ความถี่ของการส่งข้อมูลสูง จึงเป็นผลให้สามารถส่งข้อมูลเข้าสู่บัฟเฟอร์ของฝ่ายรับได้ปริมาณที่มากขึ้นให้ได้เพียงพอสำหรับแสดงผล แล้วหลังจากนั้นจึงเพิ่มช่วงระยะห่างการส่งให้ยาวขึ้นโดยส่งด้วยความถี่ที่ต่ำลง หากนำผลนี้ไปเปรียบเทียบกับรูปการส่งแพ็กเกตข้างต้นนี้ จะสามารถอธิบายถึงปัจจัยที่มีผลต่อพฤติกรรมของออนดีมานด์ทราฟฟิกได้ว่า เรียลเซิร์ฟเวอร์มีวิธีการจัดรูปแบบการส่งข้อมูลด้วยการกำหนดค่าช่วงเวลาการส่งแต่ละแพ็กเกต ไม่ใช่ทางด้านของขนาดแพ็กเกต เพราะถ้าหากพิจารณารูปที่ 3.15 และ 3.16 แล้วจะเห็นว่าความผกผันของขนาดแพ็กเกตนั้นมีสูงมากแต่แพ็กเกตที่มีขนาดใหญ่จะมีปริมาณสูงสุด



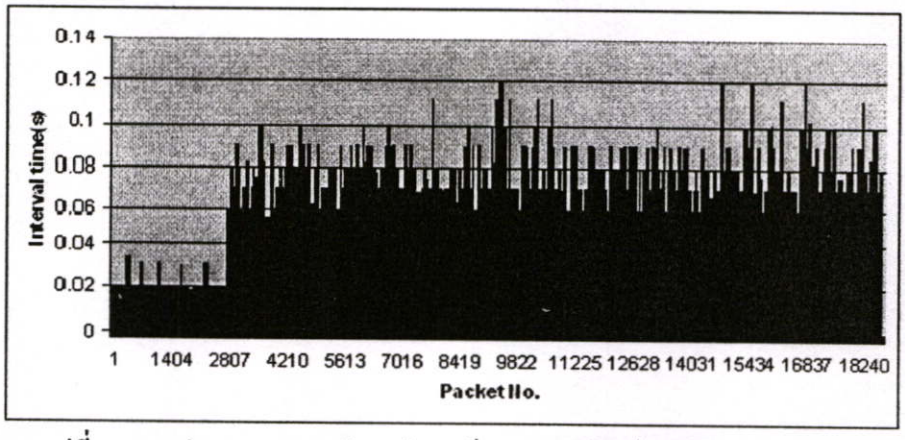
รูปที่ 3.9a ช่วงเวลาระยะห่างแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.9b ช่วงเวลาระยะห่างแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภททอล์คโชว์

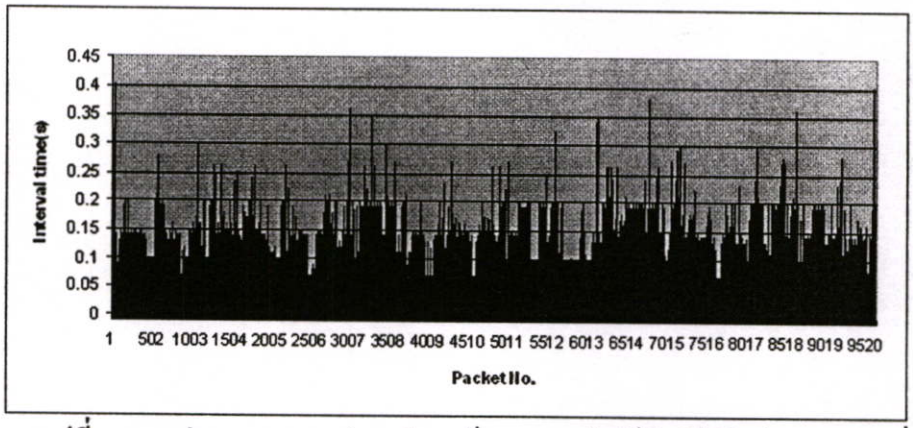


รูปที่ 3.9c ช่วงเวลาระยะห่างแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทดราม่า

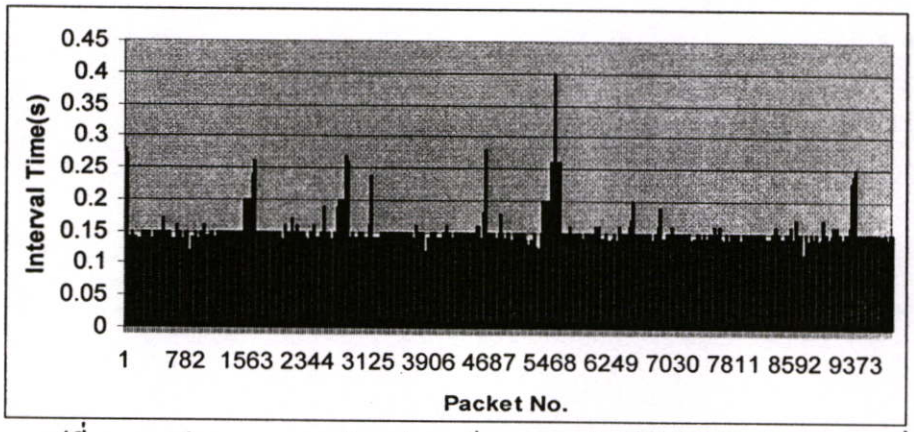


รูปที่ 3.9d ช่วงเวลาระยะห่างแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทสารคดี

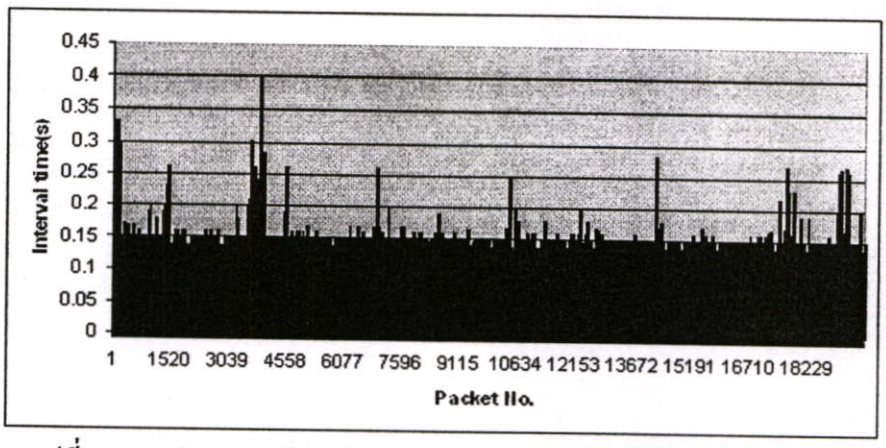
รูปที่ 3.9 ช่วงเวลาระยะห่างแต่ละแพ็กเกตของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps



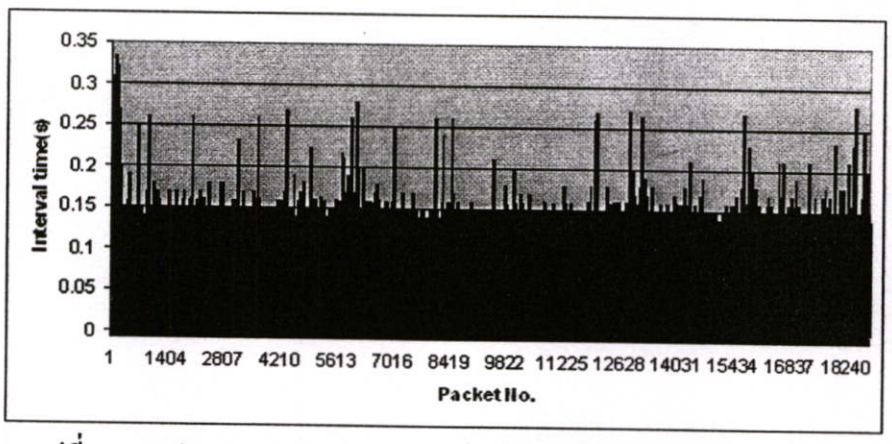
รูปที่ 3.10a ช่วงเวลาระยะห่างแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.10b ช่วงเวลาระยะห่างแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.10c ช่วงเวลาระยะห่างแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทดราม่า



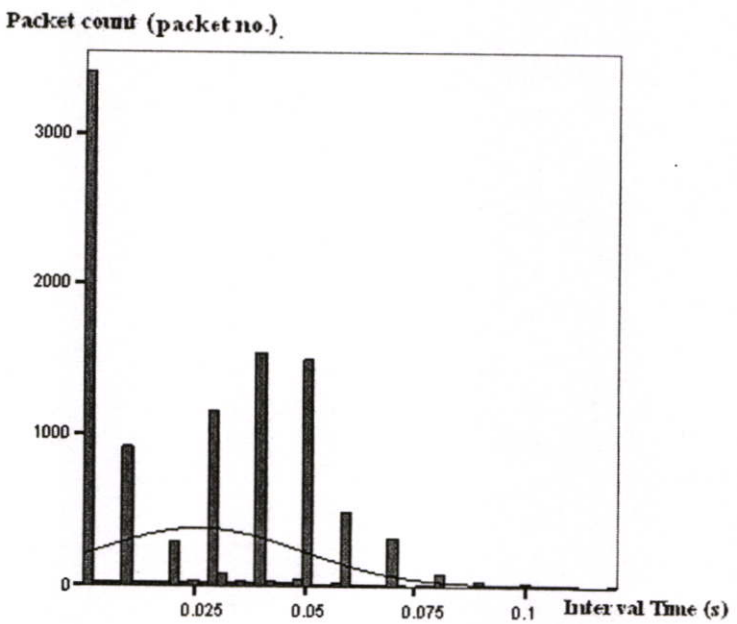
รูปที่ 3.10d ช่วงเวลาระยะห่างแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทสารคดี

รูปที่ 3.10 ช่วงเวลาระยะห่างแต่ละแพ็กเกตของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps

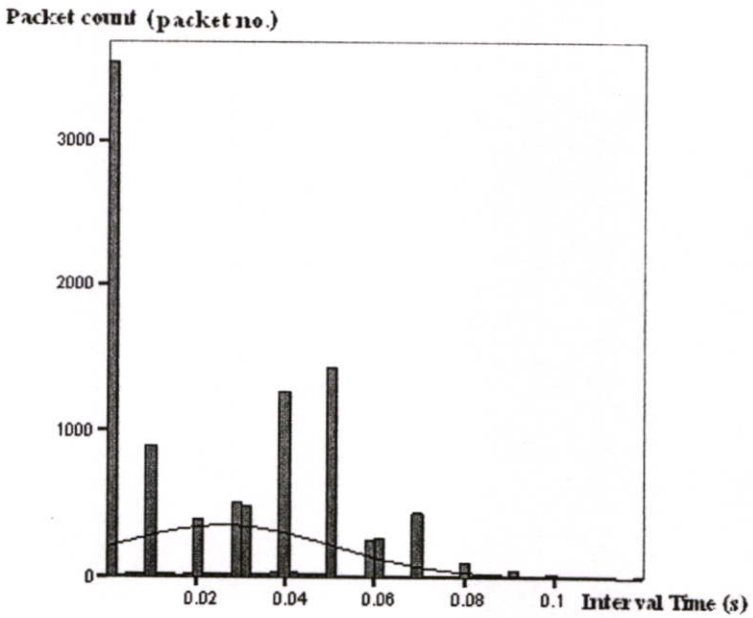
ดังนั้นเมื่อนำผลของความแตกต่างระหว่างช่วงเวลาระยะห่างการส่งแต่ละแพ็กเกต มาวิเคราะห์โดยละเอียดในกรอบของช่วงเวลาการส่งข้อมูลเพียง 1-2 วินาทีจะเห็นถึงความแตกต่างได้จากตาราง ที่ 3.9 ซึ่งแสดงค่าเฉลี่ยและค่ามีเดียนของการส่งข้อมูลระหว่างแพ็กเกตนั้น ค่าเฉลี่ยดังกล่าวค่อนข้างใกล้เคียงกันแต่ค่าของมีเดียนนั้นมีความแตกต่างกัน และจากรูปที่ 3.11 และ 3.12 แสดงให้เห็นว่าการกระจายของช่วงเวลาการส่งแพ็กเกตนั้นมีความแตกต่างกันอยู่ เส้นโค้งที่แสดงในรูป 3.11 และ 3.12 เป็นเส้นกราฟอ้างอิงแสดงถึงลักษณะความสัมพันธ์ของการกระจายตามปกติ จะเห็นได้ว่าการกระจายของช่วงเวลาระหว่างแพ็กเกตในทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์มีการกระจายที่กว้างกว่าทราฟฟิกแบบต่อเนื่องแบบเรียลไทม์ที่จะมีการส่งเป็นช่วง ๆ ซึ่งมีลักษณะเป็นรูปแบบ ON-OFF โดยจะมีช่วงการส่งข้อมูลในช่วง ON ด้วยความถี่ที่สูง (ช่วงเวลาระยะห่างของการส่งแต่ละแพ็กเกตต่ำ ๆ) ต่อเนื่องกันหลาย ๆ แพ็กเกตและคั่นด้วยช่วง OFF ซึ่งเป็นช่วงเวลาที่สูงบ้างต่ำบ้างเป็นช่วง ๆ สลับกัน ดังรูปที่ 3.12

ตารางที่ 3.9 ค่าเฉลี่ยและมีเดียนของช่วงเวลาการส่งแต่ละแพ็กเกตของการส่งประเภทต่าง ๆ (s)

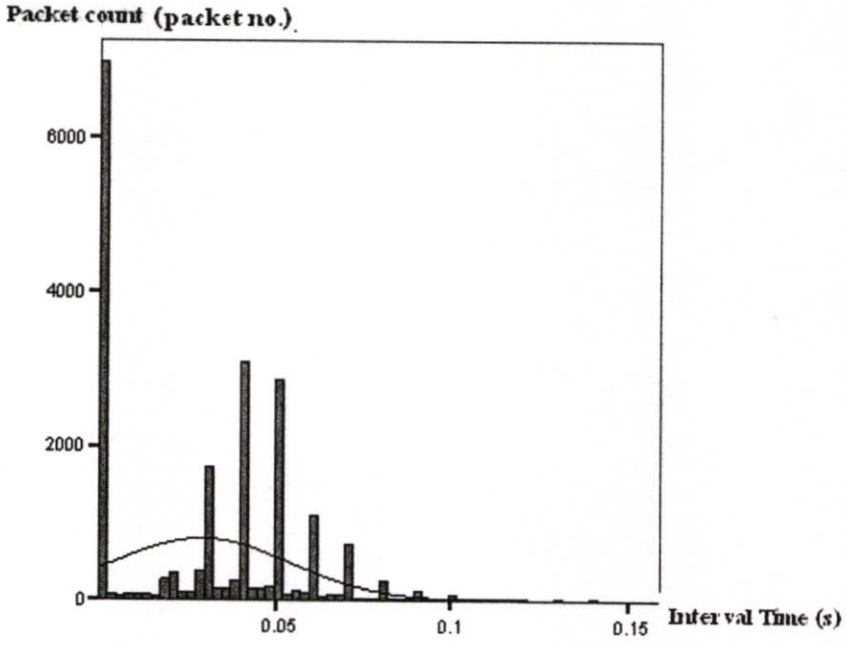
Movie Name	Rate 56 Kbps		Rate 128 Kbps		Rate 512 Kbps	
	Avg Pkt Time Interval	Median Time Interval	Avg Pkt Time Interval	Median Time Interval	Avg Pkt Time Interval	Median Time Interval
Action1-Realtime	0.069	0.000155	0.0297	0.000154	0.0142	0.000111
Action1-On demand	0.063156	0.060238	0.026187	0.029895	0.012644	0.0096
Talkshow1-Realtime	0.0905	0.000153	0.0285	0.000123	0.01419	0.00004
Talkshow1-On demand	0.066325	0.067438	0.026007	0.029403	0.011683	0.009679
Drama1- Realtime	0.068827	0.000101	0.031249	0.000054	0.013826	0.00004
Drama1-On demand	0.070381	0.062043	0.03081	0.030108	0.013891	0.009806
Doc1- Realtime	0.063117	0.0001	0.032513	0.000097	0.0141	0.000039
Doc1- On demand	0.063759	0.060065	0.032123	0.039826	0.014292	0.009822



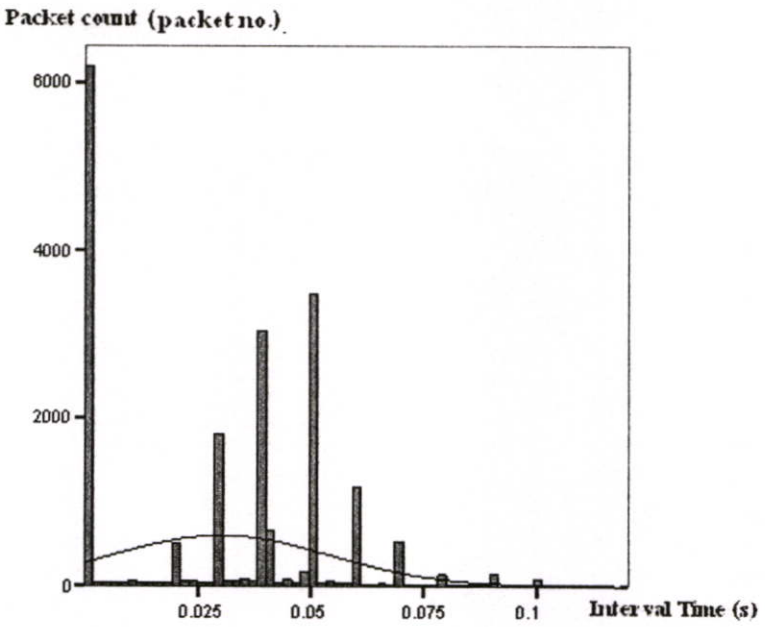
รูปที่ 3.11a การกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.11b การกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภททอล์คโชว์

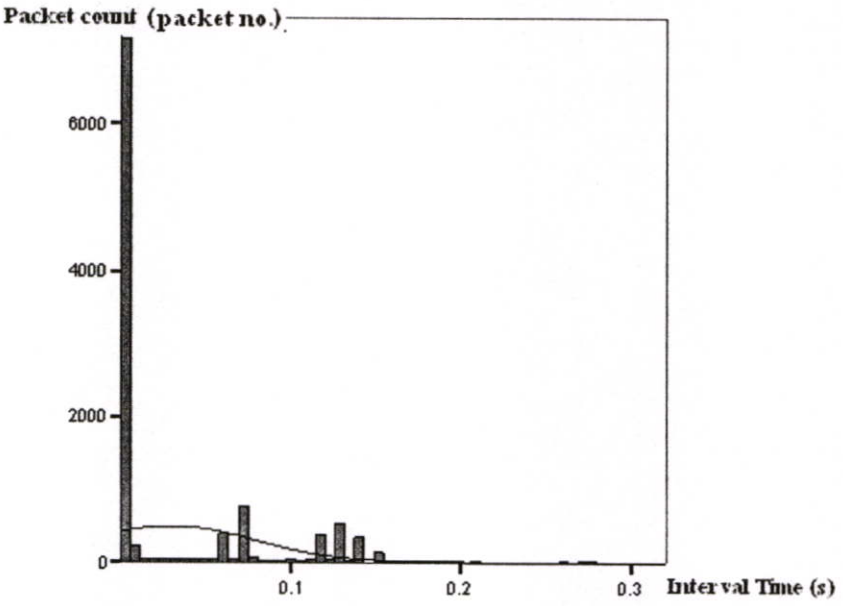


รูปที่ 3.11c การกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทดราม่า

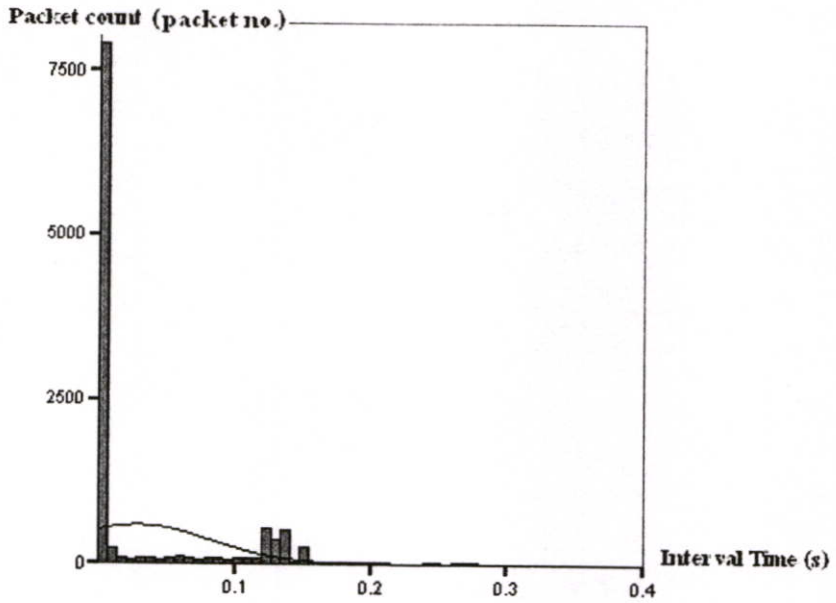


รูปที่ 3.11d การกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทสารคดี

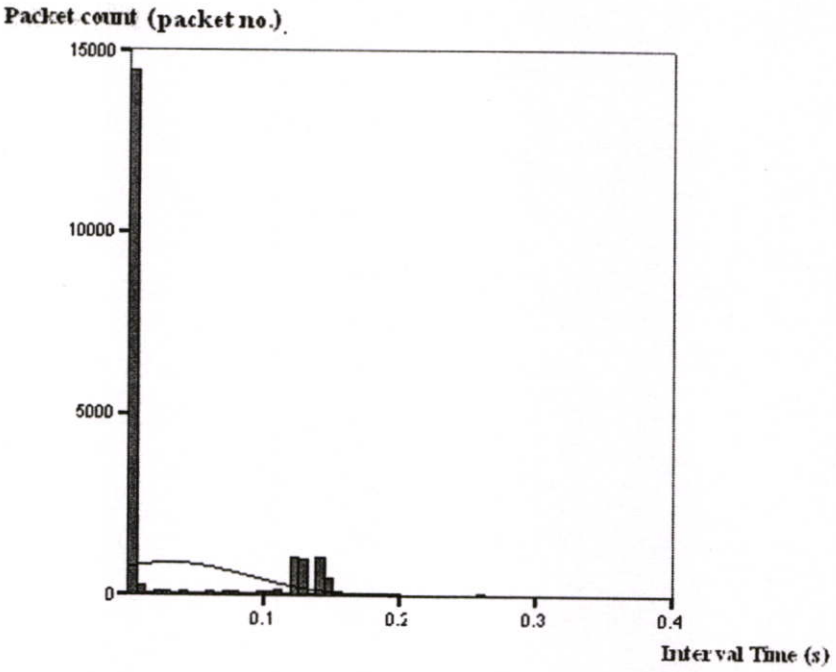
รูปที่ 3.11 กราฟการกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของทราฟฟิกประเภทออนดีมานด์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps



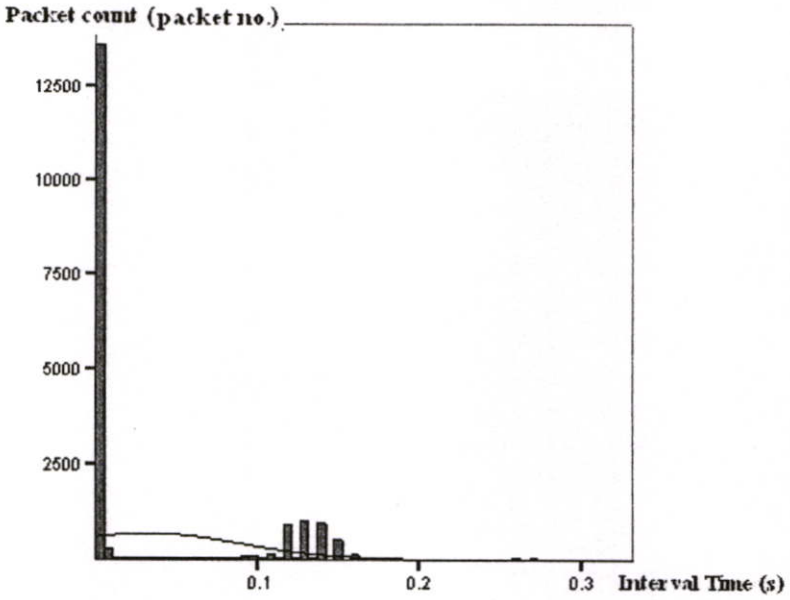
รูปที่ 3.12a การกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.12b การกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภททอล์คโชว์



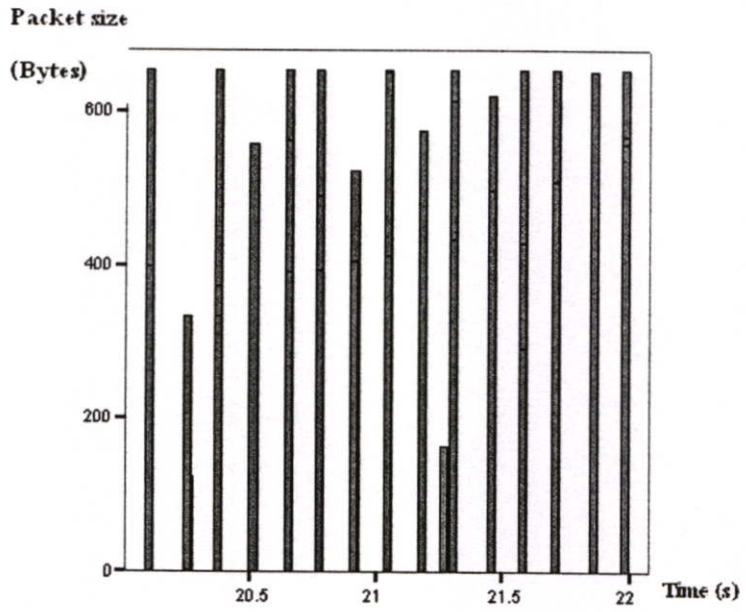
รูปที่ 3.12c การกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทดราม่า



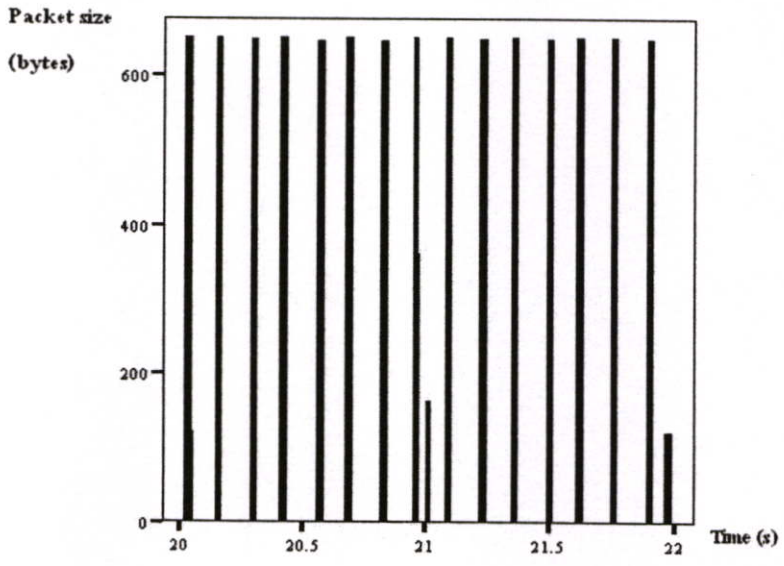
รูปที่ 3.12d การกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของไฟล์มัลติมีเดียประเภทสารคดี

รูปที่ 3.12 กราฟการกระจายของช่วงเวลาการส่งแต่ละแพ็กเกตของทราฟฟิกประเภทเรียลไทม์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps

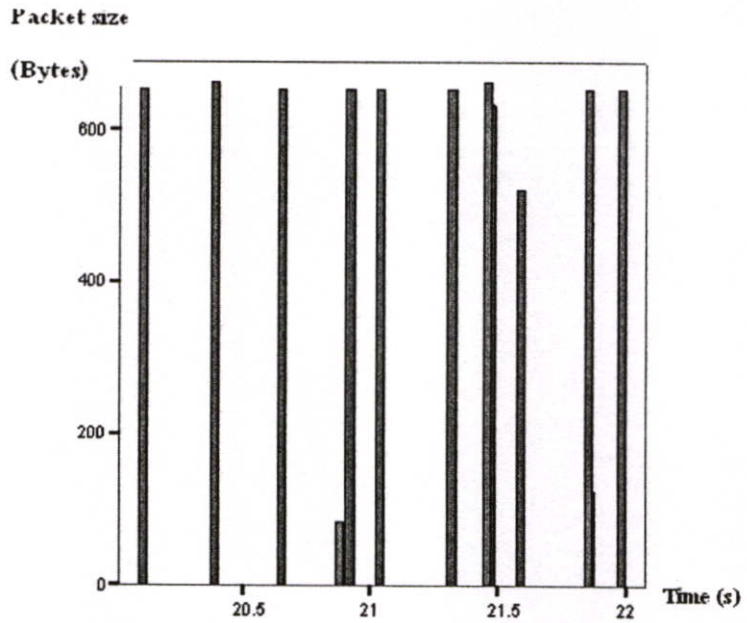
นอกจากนี้ดังรูปที่ 3.13 และ 3.14 แสดงให้เห็นว่าช่วงเวลาระยะห่างการส่งข้อมูลของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์นั้นค่อนข้างสม่ำเสมอมากกว่าและมีช่วงเวลาที่ใกล้เคียงกัน ทำให้การส่งข้อมูลเป็นไปอย่างต่อเนื่องด้วยขนาดแพ็กเกตและช่วงเวลาการส่งแพ็กเกตที่ใกล้เคียงกันดังรูปที่ 3.14 ซึ่งต่างกับทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ดังรูปที่ 3.13 โดยความสัมพันธ์ระหว่างช่วง ON-OFF ดังกล่าวมีผลสืบเนื่องมาจากอัตราการนำส่งข้อมูลด้วย ดังตารางที่ 3.10 เมื่อมีอัตราการนำส่งข้อมูลเพิ่มมากขึ้น ช่วงเวลา OFF จะลดลง ในทางกลับกันช่วงเวลา ON ของการส่งแพ็กเกตจะเพิ่มขึ้น ขณะเดียวกันขนาดของแพ็กเกตก็เพิ่มสูงขึ้นด้วย



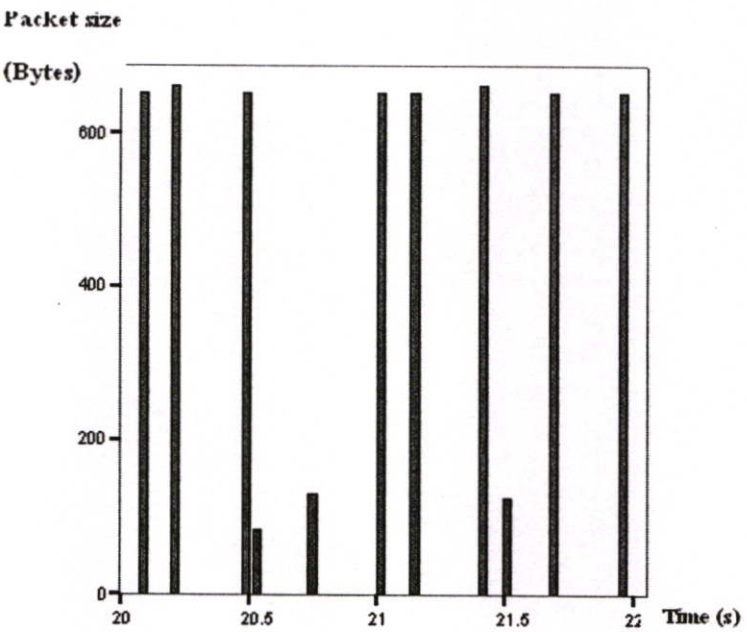
รูปที่ 3.13a กราฟช่วงเวลาและขนาดของการส่งแพ็กเกตไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.13b กราฟช่วงเวลาและขนาดของการส่งแพ็กเกตไฟล์มัลติมีเดียประเภททอล์คโชว์

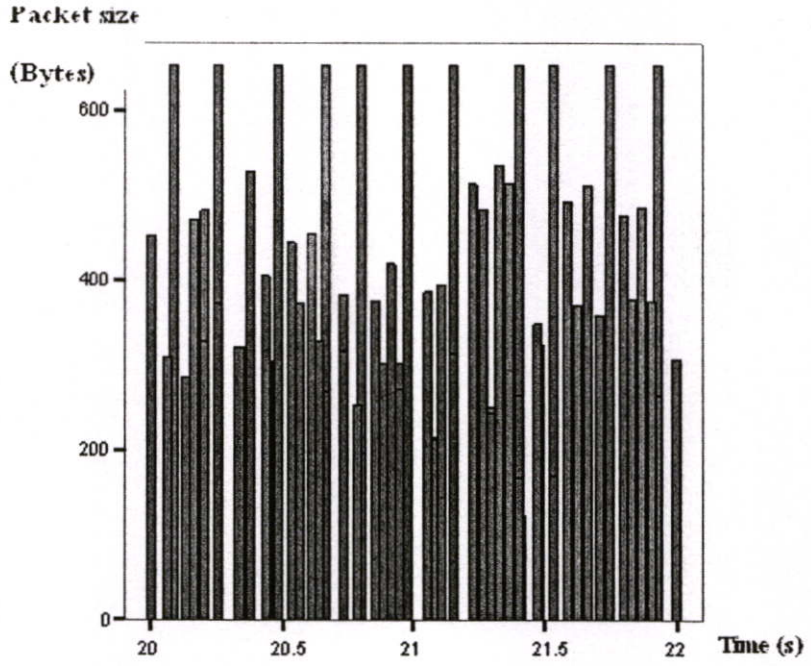


รูปที่ 3.13c กราฟช่วงเวลาและขนาดของการส่งแพ็กเกตไฟล์มัลติมีเดียประเภทดราม่า

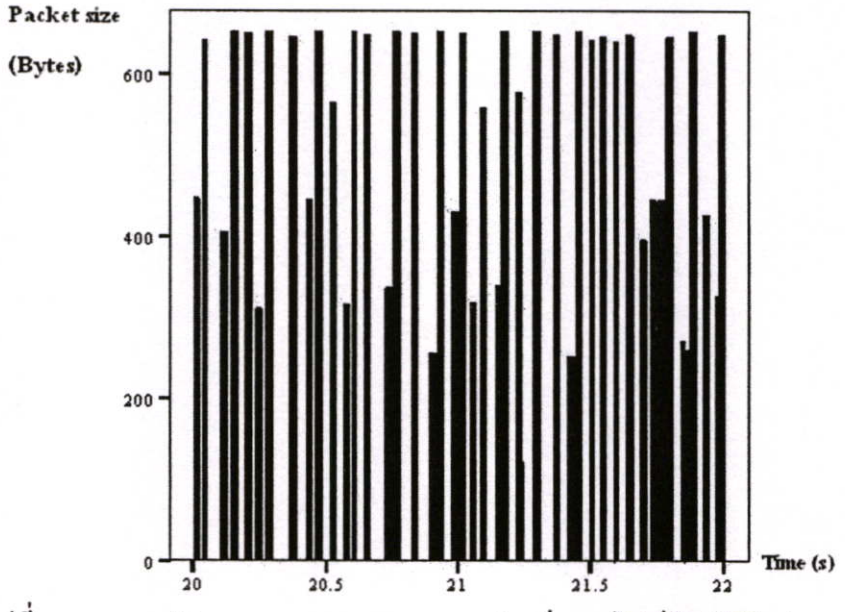


รูปที่ 3.13d กราฟช่วงเวลาและขนาดของการส่งแพ็กเกตไฟล์มัลติมีเดียประเภทสารคดี

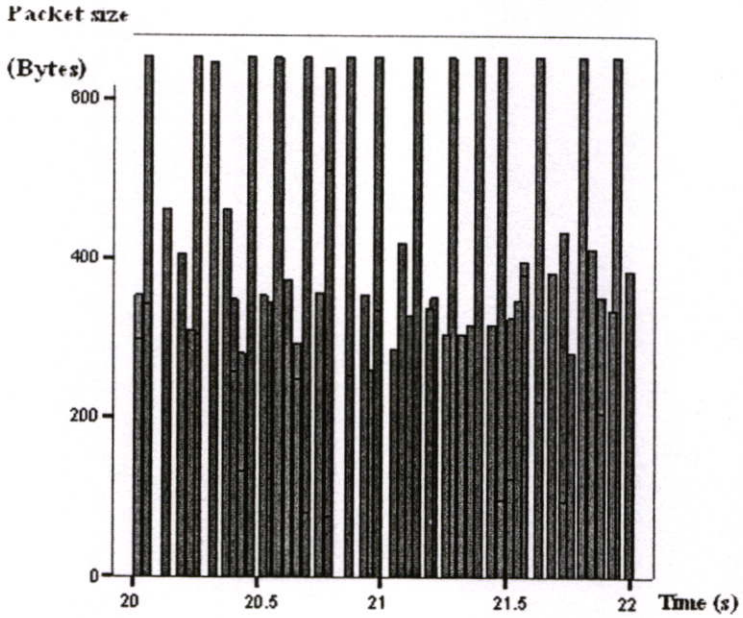
รูปที่ 3.13 กราฟช่วงเวลาและขนาดของแพ็กเกตที่ส่งด้วยทราฟฟิกแบบเรียลไทม์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps



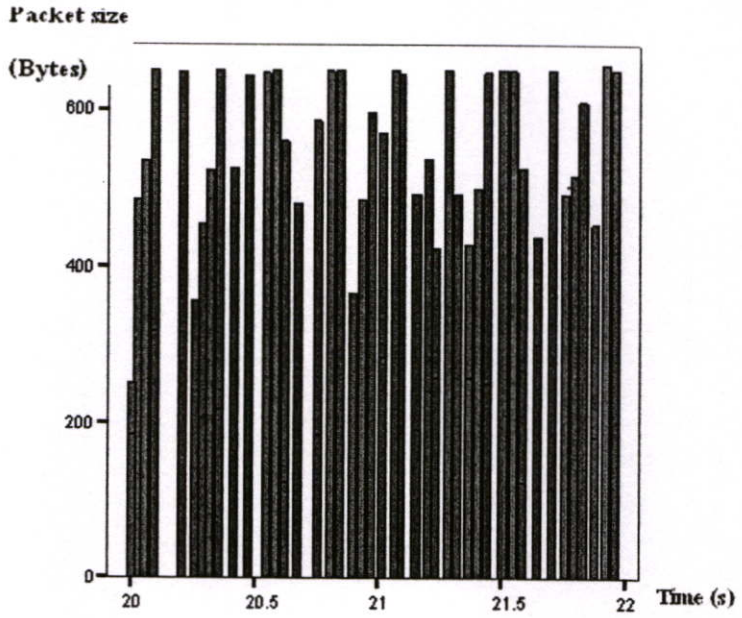
รูปที่ 3.14a กราฟช่วงเวลาและขนาดของการส่งแพ็กเกตไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.14b กราฟช่วงเวลาและขนาดของการส่งแพ็กเกตไฟล์มัลติมีเดียประเภททอล์คโชว์



รูปที่ 3.14c กราฟช่วงเวลาและขนาดของการส่งแพ็กเกตไฟล์มัลติมีเดียประเภทดราม่า



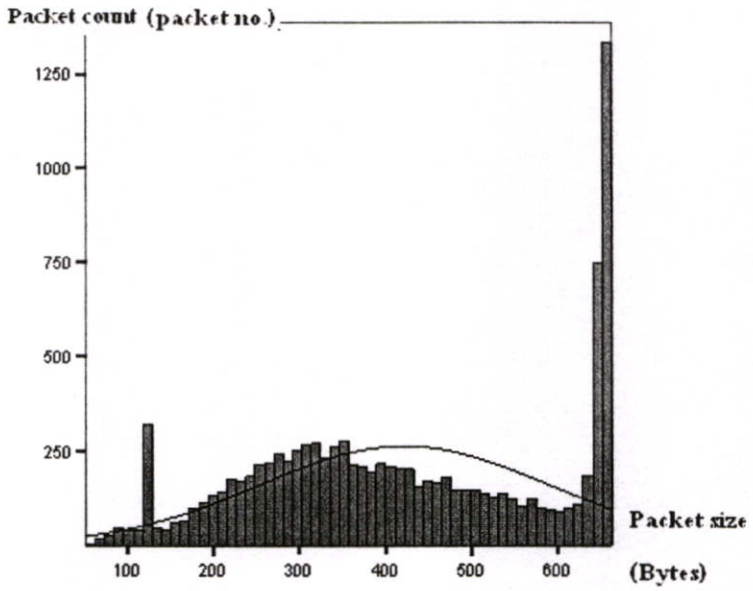
รูปที่ 3.14d กราฟช่วงเวลาและขนาดของการส่งแพ็กเกตไฟล์มัลติมีเดียประเภทสารคดี

รูปที่ 3.14 กราฟช่วงเวลาและขนาดของแพ็กเกตที่ส่งด้วยทราฟฟิกแบบออนดีมานด์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps

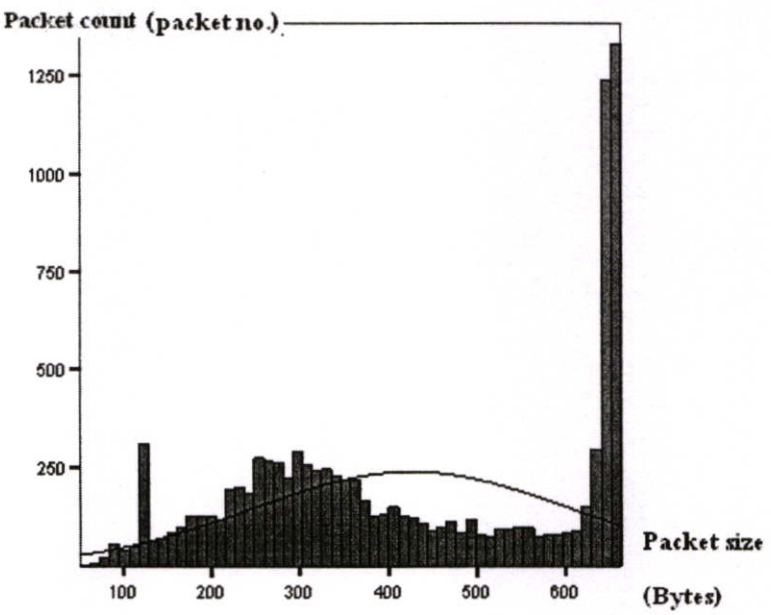
ตารางที่ 3.10 ความสัมพันธ์ของอัตราการนำส่งข้อมูล (Kbps) กับค่าเฉลี่ยของช่วง ON-OFF (s) ในการส่งข้อมูลระหว่างวีดีโอประเภทแอคชั่นและทอล์คโชว์ของเรียลไทม์ทราฟฟิก

Movie Name	Rate 56 Kbps			Rate 128 Kbps			Rate 512 Kbps		
	Avg T*off*(s)	Avg T*on*/Pkt	ON/OFF Ratio	Avg T*off*(s)	Avg T*on*/Pkt	ON/OFF Ratio	Avg T*off*(s)	Avg T*on*/Pkt	ON/OFF Ratio
Action1	0.17298	0.000121	1.48	0.08936	0.000121	2.056	0.04524	0.000084	2.2268
Action2	0.17437	0.000124	1.256	0.105474	0.000119	2.24	0.050114	0.000082	2.58
Talkshow1	0.22382	0.000117	1.4749	0.099329	0.000111	2.51904	0.07991	0.000058	4.44626
Talkshow2	0.16084	0.000113	1.7931	0.11126	0.000109	2.43689	0.07025	0.000065	4.02928
Drama1	0.22548	0.000744	2.301	0.120965	0.000191	2.8894	0.064824	0.000048	3.6518
Drama2	0.15985	0.000564	1.8085	0.12996	0.000239	2.796	0.079614	0.000042	4.7581
Doc1	0.19442	0.000494	2.096	0.121515	0.00023	2.757	0.069961	0.000045	4.1876
Doc2	0.18779	0.00078	2.196	0.121613	0.000258	2.677	0.075227	0.000043	4.242

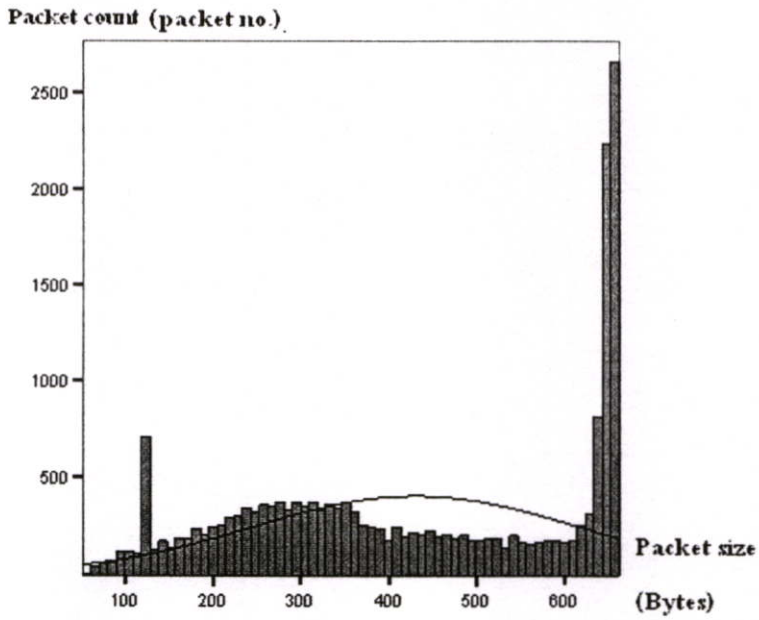
สำหรับความแตกต่างระหว่างการกระจายของขนาดแพ็กเกตนั้น ทั้งรูปแบบทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์และออนดีมานด์ มีค่าเฉลี่ยและค่าการกระจายที่ใกล้เคียงกันดังรูปที่ 3.15 และ 3.16 ซึ่งสามารถสรุปได้ว่า ความแตกต่างในส่วนนี้ไม่ใช่ปัจจัยหลักสำหรับพฤติกรรมการส่งข้อมูลรูปแบบที่ต่างกันของเรียลเซิร์ฟเวอร์ จากผลการทดลองสามารถสรุปได้ว่าขนาดของแพ็กเกตนั้นจะถูกกำหนดไว้ตั้งแต่ขั้นตอนของการเข้ารหัสข้อมูล แต่ช่วงเวลาของการส่งข้อมูลแต่ละแพ็กเกตนั้นจะถูกควบคุมโดยเรียลเซิร์ฟเวอร์ในขั้นตอนของการส่ง ซึ่งขึ้นอยู่กับรูปแบบของประเภทของทราฟฟิกแบบต่อเนื่องที่ให้บริการ เมื่อวิเคราะห์ผลการทดลองข้างต้นของทราฟฟิกแบบต่อเนื่องจากประเภทของไฟล์วีดีโอที่ต่างชนิดกันคือประเภทแอคชั่นและประเภททอล์คโชว์จากกลุ่มตัวอย่าง พบว่าไฟล์มัลติมีเดียทั้ง 4 ประเภทมีลักษณะพฤติกรรมการส่งข้อมูลและค่าเฉลี่ยหรือค่าการกระจายต่าง ๆ ที่ใกล้เคียงกัน จึงสรุปได้ว่าความแตกต่างกันระหว่างประเภทของไฟล์วีดีโอที่มีเนื้อหาต่างประเภทกันที่เผยแพร่ไม่มีผลต่อลักษณะการส่งข้อมูล



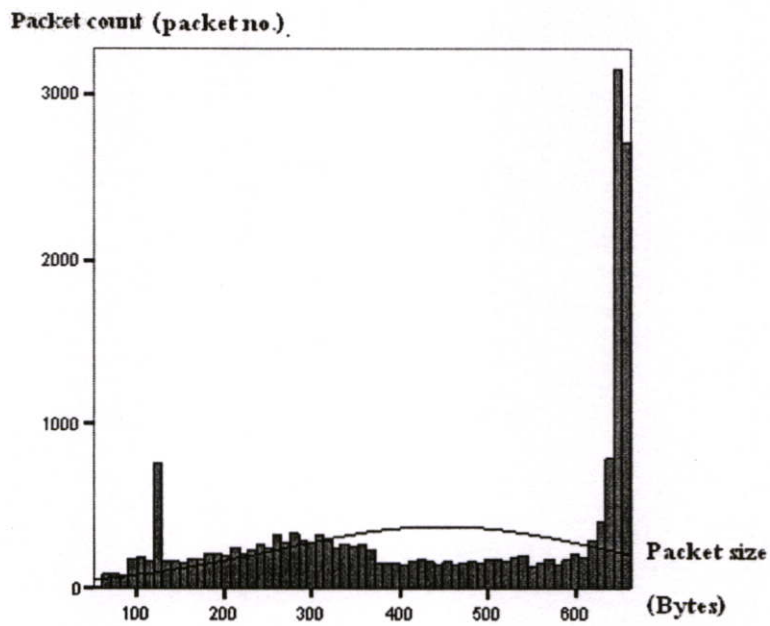
รูปที่ 3.15a กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.15b กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกไฟล์มัลติมีเดียประเภททอล์คโชว์

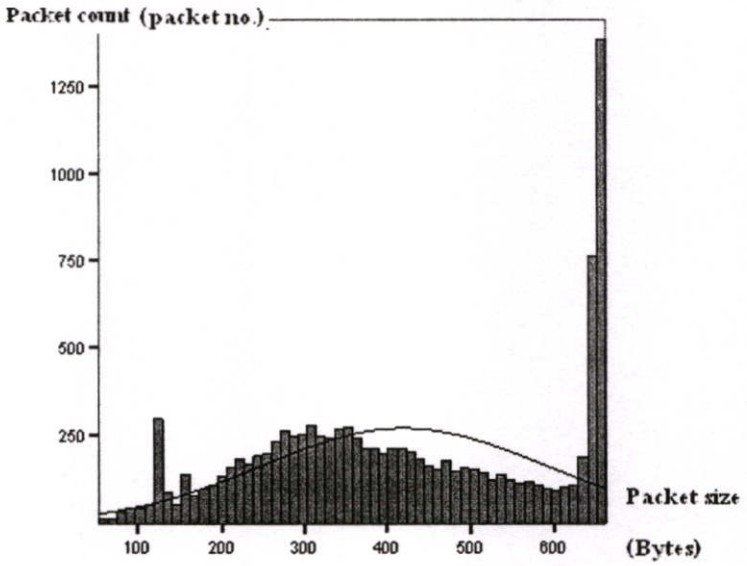


รูปที่ 3.15c กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกไฟล์มัลติมีเดียประเภทดราม่า

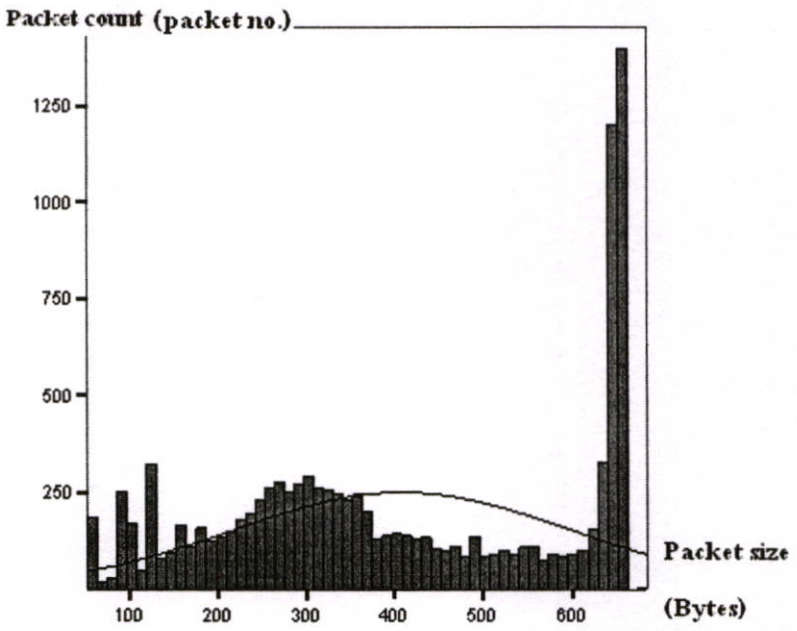


รูปที่ 3.15d กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกไฟล์มัลติมีเดียประเภทสารคดี

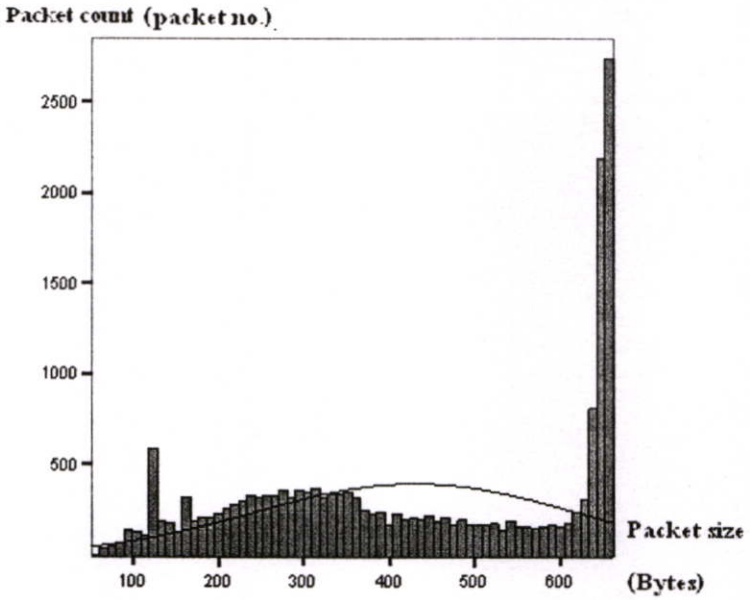
รูปที่ 3.15 กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกประเภทออนดีมานด์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps



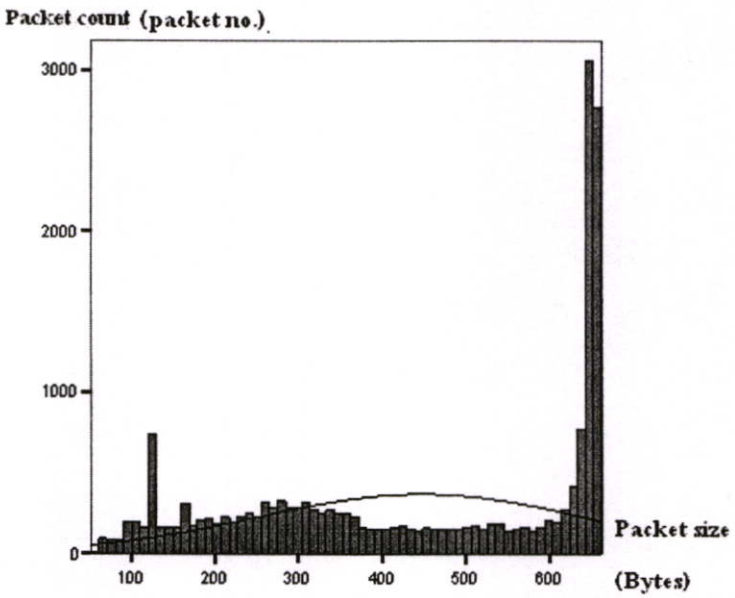
รูปที่ 3.16a กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกไฟล์มัลติมีเดียประเภทแอคชั่น



รูปที่ 3.16b กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกไฟล์มัลติมีเดียประเภททอล์คโชว์



รูปที่ 3.16c กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกไฟล์มัลติมีเดียประเภทคราม่า



รูปที่ 3.16d กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกไฟล์มัลติมีเดียประเภทสารคดี

รูปที่ 3.16 กราฟการกระจายขนาดแพ็กเกตของทราฟฟิกประเภทเรียลไทม์ของไฟล์มัลติมีเดียประเภทต่าง ๆ ในอัตราการนำส่ง 128 Kbps

3.3 สรุปท้ายบท

ในบทนี้ได้นำเสนอการศึกษาความแตกต่างของพฤติกรรมการส่งข้อมูลระหว่างทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์และเรียลไทม์ทราฟฟิก ซึ่งเมื่อนำข้อมูลที่ได้มาวิเคราะห์จะพบว่าหากพิจารณาข้อมูลในภาพรวมทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์นั้นจะส่งข้อมูลจำนวนมาก ๆ ในช่วงเริ่มต้นเข้าสู่บัฟเฟอร์ของฝ่ายรับ หลังจากนั้นจึงส่งข้อมูลอย่างต่อเนื่องด้วยอัตราการส่งที่ลดลงจนสิ้นสุดการส่งข้อมูล แต่สำหรับทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์นั้นจะส่งข้อมูลออกไปด้วยอัตราการส่งสม่ำเสมอตลอดช่วงของการแสดงผล และเมื่อพิจารณาให้ละเอียดลงไปจะพบว่าลักษณะการส่งทราฟฟิกแบบต่อเนื่องทั้ง 2 ประเภทใช้ช่วงเวลาระยะห่างระหว่างการส่งแต่ละแพ็กเกตเป็นกลไกและปัจจัยของพฤติกรรมที่ต่างกันในส่งข้อมูล ซึ่งลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์นั้นจะส่งข้อมูลอย่างต่อเนื่องด้วยอัตราที่ค่อนข้างคงที่เข้าสู่บัฟเฟอร์ของฝ่ายรับโดยที่ฝ่ายรับจะเป็นฝ่ายควบคุมการแสดงผล แต่สำหรับการส่งทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์นั้นจะมีลักษณะการส่งข้อมูลเป็นแบบ ON-OFF โดยฝ่ายส่งจะควบคุมการส่งข้อมูล ซึ่งสามารถสรุปได้ว่าวิธีการส่งข้อมูลของเรียลเซิร์ฟเวอร์จะควบคุมช่วงเวลาระยะห่างระหว่างแพ็กเกตเท่านั้น สำหรับความแตกต่างของช่วงเวลาการส่งแพ็กเกตนั้นขึ้นอยู่กับรูปแบบของประเภทของทราฟฟิกแบบต่อเนื่องที่ให้บริการ แต่ขนาดของแพ็กเกตนั้นจะถูกกำหนดไว้ตั้งแต่ขั้นตอนของการเข้ารหัสข้อมูลตั้งแต่ต้น

นอกจากนั้นความสัมพันธ์ของอัตราการนำส่งข้อมูลกับพฤติกรรมการส่งนั้น มีผลโดยตรงกับการกระจายตัวของขนาดแพ็กเกตและกระจายตัวของช่วงเวลาห่างในการส่งข้อมูลแต่ละแพ็กเกต รวมถึงช่วงเวลา ON-OFF อีกด้วย โดยข้อมูลและผลการวิเคราะห์เหล่านี้สามารถนำไปสร้างแบบจำลองทราฟฟิกแบบต่อเนื่องได้อย่างถูกต้องต่อไปในบทที่ 4

บทที่ 4
การจำลองทราฟฟิกแบบต่อเนื่อง

4.1 การจำลองระบบเครือข่ายเพื่อการวิเคราะห์

ในการศึกษาการทำงานทางด้านระบบเครือข่ายไม่ว่าจะเป็นการสร้างการสื่อสารใหม่ ๆ การออกแบบระบบเครือข่ายและการวิเคราะห์ผลกระทบที่จะเกิดขึ้นกับระบบเครือข่าย ในปัจจุบันนี้ใช้วิธีการสร้างระบบเครือข่ายจำลองขึ้นจากโปรแกรมคอมพิวเตอร์ เพื่อที่จะวิเคราะห์ผลมากกว่าที่จะกระทำกับระบบเครือข่ายจริง ทั้งนี้เพื่อหลีกเลี่ยงผลกระทบต่าง ๆ ที่จะเกิดขึ้นกับระบบจริงและเพื่อการศึกษา ที่ได้ผลเร็วขึ้น

เนื่องจากมีผู้ใช้งานการจำลองระบบเครือข่ายเพื่อการศึกษามากขึ้น จึงมีนักวิจัยที่พัฒนาองค์ประกอบต่าง ๆ ขึ้นมาบนโปรแกรมจำลองนี้ให้เหมือนกับการใช้งานบนระบบเครือข่ายจริงมากขึ้น เช่น การสร้างการทำงานของอีเมลหรือเว็บทราฟฟิกขึ้นเพื่อให้ผู้ศึกษาต่อไปนำไปใช้จำลองระบบเครือข่าย เป็นต้น โดยโปรแกรมจำลองระบบเครือข่ายที่นิยมใช้กันอยู่ในปัจจุบันมี 2 ประเภทด้วยกัน

- NS-2 เป็นโปรแกรมจำลองระบบเครือข่ายที่มีผู้นิยมใช้งานมากในปัจจุบัน เนื่องจากเป็นโปรแกรมฟรีแวร์และเปิดเผยโค้ดของโปรแกรม ซึ่งทำให้มีผู้ใช้สามารถพัฒนาต่อได้ง่ายและเผยแพร่ให้ผู้อื่นนำไปศึกษาต่อไปได้ ดังนั้นจึงมีนักวิจัยใช้งาน NS-2 และพัฒนาออบเจ็กต์ต่าง ๆ ขึ้นมากมาย การศึกษาวิจัยทางด้านระบบเครือข่ายมากมายอ้างอิงผลการทดลองจากโปรแกรม NS-2 ลักษณะการใช้งาน NS-2 จะเป็นการเขียนโปรแกรมด้วยภาษาทีซีแอล (TCL) เพื่อสร้างแบบจำลองระบบเครือข่ายด้วยอ็อบเจกต์ต่าง ๆ ที่มีให้เรียกใช้มากมาย อ็อบเจกต์ดังกล่าวสามารถแก้ไขการทำงานภายในหรือสร้างอ็อบเจกต์ใหม่ ๆ เพิ่มเติมได้ตามต้องการโดยใช้ภาษาซี [9]
- OPNET เป็นอีกหนึ่งโปรแกรมที่มีผู้ใช้งานอย่างแพร่หลาย เนื่องจากเป็นโปรแกรมที่ใช้งานง่ายและเป็นกราฟฟิกยูสเซอร์อินเทอร์เฟส นอกจากนั้นยังมีอ็อบเจ็กต์ให้ใช้งานมากมายพร้อมกับการแสดงผลค่าต่าง ๆ จากการจำลองระบบเครือข่ายได้หลากหลายและ OPNET ยังสามารถแก้ไขกระบวนการทำงานภายในอ็อบเจกต์หรือสร้างอ็อบเจกต์ใหม่ ๆ ได้เช่นเดียวกันกับ NS-2 [10]

การออกแบบจำลองของทราฟฟิกแบบต่อเนื่องในวิทยานิพนธ์เล่มนี้ เลือกใช้การออกแบบบนระบบ NS-2 เนื่องจากสามารถพัฒนาอ็อบเจกต์เพิ่มเติมได้และเป็นโปรแกรมฟรีแวร์เหมาะที่จะใช้ในการศึกษาวิจัย

4.2 การออกแบบจำลองทราฟฟิกแบบต่อเนื่อง

จากผลการทดลองเพื่อวิเคราะห์หาพฤติกรรมของทราฟฟิกแบบต่อเนื่องจากบทที่ 3 ทำให้ได้ข้อมูลต่าง ๆ ซึ่งสรุปได้ดังนี้

4.2.1 แบบจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

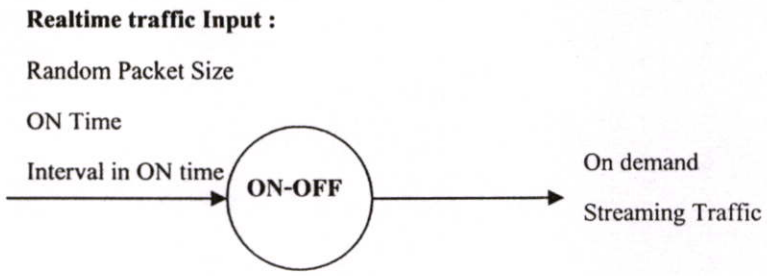
ทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์มีคุณสมบัติต่าง ๆ ดังต่อไปนี้

- มีรูปแบบการส่งข้อมูลแบบ ON-OFF ซ้ำ ๆ กันตลอดช่วงการแสดงผล โดยมีความถี่ของคาบเวลาช่วง ON-OFF และช่วงเวลาการส่งแต่ละแพ็กเกตในช่วง ON ที่ค่อนข้างคงที่ซึ่งค่าดังกล่าวแปรผันตามอัตราการนำส่งที่ต่าง ๆ กัน
- การส่งข้อมูลจะส่งข้อมูลไปเรื่อย ๆ ตลอดช่วงการแสดงผลของมัลติมีเดียตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดการแสดงผล
- อัตราการนำส่งข้อมูลจะมีผลต่อขนาดเฉลี่ยของแพ็กเกตและช่วงเวลาระยะห่างของการส่งแต่ละแพ็กเกต

เมื่อวิเคราะห์หาค่าต่าง ๆ ที่จำเป็นในการออกแบบจำลองของทราฟฟิกแบบต่อเนื่องจากบทที่ 4 ทำให้ได้ข้อมูลต่าง ๆ สำหรับเป็นอินพุตให้กับฟังก์ชันรูปแบบทราฟฟิกประเภท ON-OFF ดังตารางที่ 4.1 ซึ่งขนาดของแต่ละแพ็กเกตในแบบจำลองทราฟฟิกนี้ใช้ฟังก์ชันการสุ่มด้วยการกระจายแบบปกติ (Normal distribution) โดยมีช่วงขอบเขตของขนาดแพ็กเกตดังในตารางที่ 4.1 และสำหรับช่วงเวลาการส่งแต่ละแพ็กเกตที่ได้จากผลการทดลองในบทที่ 3 แสดงให้เห็นว่าคาบเวลาในช่วง ON และ OFF มีค่าใกล้เคียงกับค่าเฉลี่ยในทุก ๆ คาบหรือมีการกระจายแบบคงที่ (Uniform distribution) ดังนั้นเพื่อให้สอดคล้องกับรูปแบบทราฟฟิกต่อเนื่องที่ได้ศึกษามาในฟังก์ชันจำลองนี้จึงใช้วิธีการกำหนดค่าคงตัวให้กับฟังก์ชัน ON-OFF จะทำให้ได้แบบจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตรานำส่งที่ต้องการ ดังรูปที่ 4.1

ตารางที่ 4.1 ค่าอินพุตที่ใช้สำหรับจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

Encoding Rate (Kbps)	Average Packet size (Byte)	Packet size range (Byte)	ON Time period (s)	Packet Interval in ON time period (s)	OFF Time period (s)
56	389.3044	66 - 659	0.0002655	0.000177	0.02058
128	443.5446	66 - 661	0.0005877	0.00025443	0.00833
512	829.2451	66 - 1459	0.0008244	0.0002407	0.003628



รูปที่ 4.1 ฟังก์ชันแบบจำลองของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

4.2.2 แบบจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

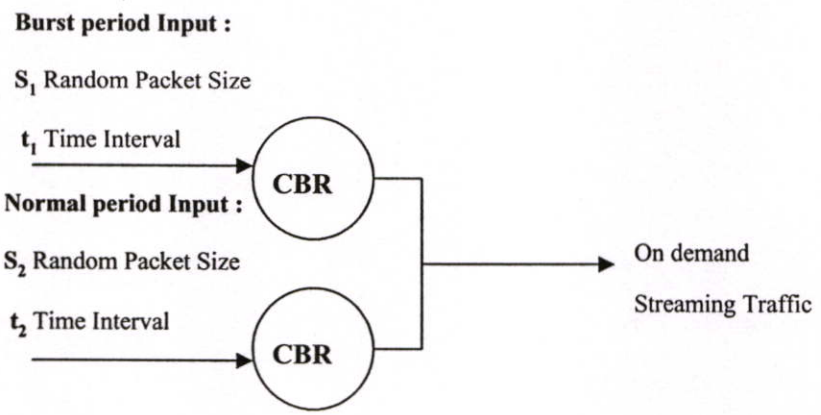
ทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์มีคุณสมบัติต่าง ๆ ดังต่อไปนี้

- ประกอบด้วยรูปแบบการส่งหลัก 2 ส่วนด้วยกัน โดยการส่งข้อมูลช่วงแรกมีการส่งข้อมูลด้วยความถี่สูง หลังจากนั้นจะส่งข้อมูลด้วยความถี่ที่ต่ำลงอย่างต่อเนื่องไปจนกระทั่งสิ้นสุดการส่งข้อมูล ซึ่งทั้ง 2 ช่วงดังกล่าวมีลักษณะการส่งข้อมูลที่มีช่วงระยะห่างของการส่งแต่ละแพ็กเกตค่อนข้างคงที่และมีการกระจายขนาดแพ็กเกตที่ใกล้เคียงกัน
- การส่งข้อมูลจะส่งข้อมูลเข้าสู่บัฟเฟอร์ของฝ่ายรับจนได้รับข้อมูลที่จะแสดงผลทั้งหมด โดยการส่งข้อมูลไม่ได้ถูกส่งตลอดช่วงของการแสดงผลดังเช่นในประเภทเรียลไทม์
- อัตราการนำส่งข้อมูลจะมีผลต่อขนาดเฉลี่ยของแพ็กเกตและช่วงเวลาการส่งระหว่างแพ็กเกต

เมื่อวิเคราะห์หาค่าต่าง ๆ ที่จำเป็นในการออกแบบจำลองของทราฟฟิกแบบต่อเนื่องจากบทที่ 3 ทำให้ได้ข้อมูลต่าง ๆ ดังตารางที่ 4.2 ซึ่งเมื่อนำค่าอินพุตต่าง ๆ ดังกล่าวผ่านฟังก์ชันรูปแบบทราฟฟิกประเภทคงที่โดยแยกอินพุตทราฟฟิกออกเป็น 2 ส่วนหลัก ๆ จะทำให้ได้แบบจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตรานำส่งที่ต้องการ ดังรูปที่ 4.2 ซึ่งทั้ง 2 ส่วนเมื่อเปรียบเทียบในอัตราเดียวกัน จะเห็นได้ว่ามีค่าอินพุตของขนาดแพ็กเกตเฉลี่ยใกล้เคียงกันและช่วงการสุ่มของแพ็กเกตเท่ากัน แต่ช่วงระยะห่างของการส่งแต่ละแพ็กเกตนั้นมีความแตกต่างกัน

ตารางที่ 4.2 ค่าอินพุตที่ใช้สำหรับจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

Encoding Rates (Kbps)	Burst period			Normal period		
	Average Packet size (Byte)	Packet Size Range (Byte)	Packet Interval (s)	Average Packet size (Byte)	Packet Size Range (Byte)	Packet Interval (s)
56	389.3044	66-659	0.02058	375.9529	66-659	0.075997
128	443.5446	66-661	0.00833	448.965	66-661	0.031572
512	829.2451	66-1461	0.003628	837.5425	66-1461	0.013933



รูปที่ 4.2 ฟังก์ชันแบบจำลองของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

4.3 การสร้างแบบจำลองทราฟฟิกแบบต่อเนื่อง NS-2

จากเดิมที่ฟังก์ชันทราฟฟิกบน NS-2 ต้องระบุค่าต่าง ๆ ตามตัวแปรที่จำกัดและข้อจำกัดของตัวแปรที่ต้องระบุขนาดแพ็กเกตที่คงที่เท่านั้นดังฟังก์ชันที่ 4.1 และ 4.2 ดังนั้นรูปแบบที่สร้างขึ้นมาจึงมีความผิดเพี้ยนจากรูปแบบของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ ในวิทยานิพนธ์เล่มนี้ได้นำเสนอรูปแบบฟังก์ชันใหม่ ที่พัฒนาขึ้นให้ได้ผลใกล้เคียงกับทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์โดยเฉพาะ ดังฟังก์ชันที่ 4.3 ซึ่งมีค่าตัวแปรรองรับอัตราการนำส่งของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ที่ต้องการในอัตรา 56 Kbps 128 Kbps และ 256 Kbps ตามลำดับ ซึ่งทำให้ง่ายสำหรับผู้ใช้ในการกำหนดค่า นอกจากนั้นยังได้พัฒนาลักษณะการส่งข้อมูลด้วยการสุ่มขนาดแพ็กเกตแต่ละแพ็กเกต ทำให้มีรูปแบบใกล้เคียงกับทราฟฟิกแบบต่อเนื่องในระบบจริงมากขึ้น โดยค่าในอัตราการนำส่งต่าง ๆ ใช้ค่าอินพุตจากการศึกษาในบทที่ 4 ดังตารางที่ 4.1

Application/Traffic/Exponential object

ค่าอินพุตของฟังก์ชัน :	PacketSize_	แทนขนาดแพ็กเกต	
	burst_time_	แทนช่วงเวลาการส่งทราฟฟิก	
	idle_time_	แทนช่วงเวลาการหยุดส่งทราฟฟิก	
	rate_	แทนอัตราการนำส่งข้อมูล	(4.1)

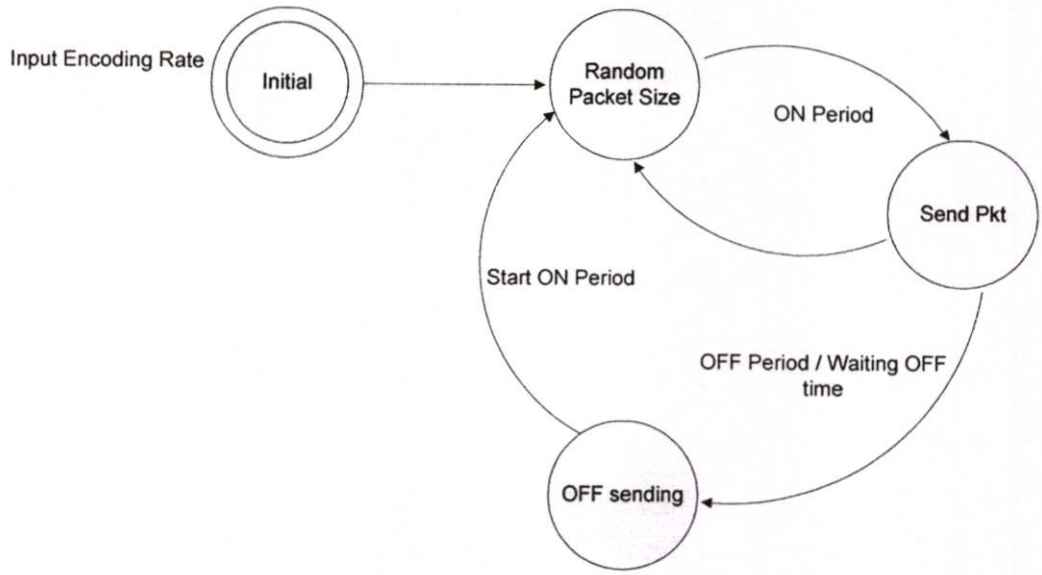
Application/Traffic/CBR object

ค่าอินพุตของฟังก์ชัน :	PacketSize_	แทนขนาดแพ็กเกต	
	interval_	แทนช่วงเวลาห่างของการส่งแต่ละแพ็กเกต	
	rate_	แทนอัตราการนำส่งข้อมูล	(4.2)

รูปแบบอ็อบเจกต์ ON-OFF ของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ที่พัฒนาดังฟังก์ชันที่ 4.3 มีหลักการทำงานดังรูปที่ 4.3 คือรับค่าอินพุตของอัตราการนำส่งที่ต้องการเพื่อนำไปกำหนดค่าการสุ่มขนาดแพ็กเกต เวลาช่วง ON เวลาช่วง OFF และช่วงเวลาห่างระหว่างแต่ละแพ็กเกตในช่วง ON หลังจากนั้นจะทำการส่งแพ็กเกตในช่วง ON จนกระทั่งสิ้นสุดช่วง ON ฟังก์ชันจะหยุดส่งข้อมูลและรอด้วยระยะเวลาของช่วง OFF หลังจากนั้นจะกลับสู่ขั้นตอนการส่งข้อมูลในช่วง ON อีกครั้ง ซึ่งลักษณะการทำงานซ้ำ ๆ ดังกล่าวของฟังก์ชันทำให้เกิดทราฟฟิกที่มีรูปแบบ ON-OFF ด้วยขนาดของแต่ละแพ็กเกตที่เกิดจากฟังก์ชันการสุ่มขนาด เช่นเดียวกับทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์บนระบบจริง

Application/Traffic/Realtime object

ค่าอินพุตของฟังก์ชัน :	rate_	แทนอัตราการนำส่งข้อมูล	(4.3)



รูปที่ 4.3 ขั้นตอนการทำงานของแบบจำลองของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

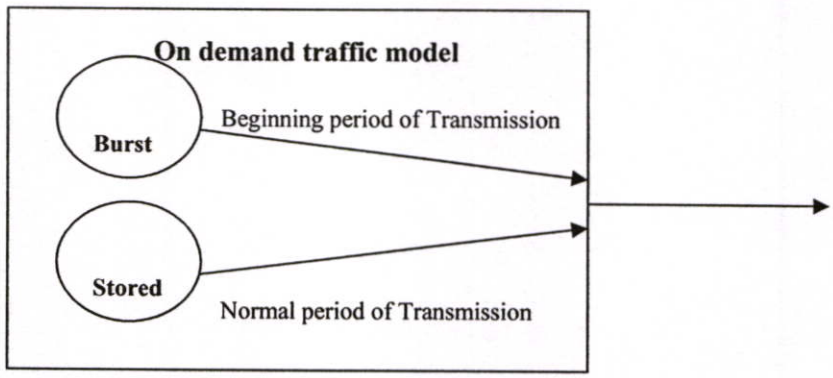
สำหรับทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ ในการทดลองนี้ได้พัฒนาฟังก์ชันขึ้นมารองรับรูปแบบทราฟฟิกดังกล่าวเช่นกัน ดังฟังก์ชันที่ 4.4 และ 4.5 โดยรองรับค่าตัวแปรอัตราการนำส่ง และพัฒนาการส่งโดยสุ่มขนาดของแต่ละแพ็กเกตเช่นเดียวกับในอ็อบเจกต์เรียลไทม์ที่พัฒนาขึ้น โดยใช้ค่าในอัตราการนำส่งต่าง ๆ กันดังตารางที่ 4.2 เป็นค่าอินพุต และฟังก์ชันที่ 4.4 แทนช่วงการส่งข้อมูลช่วงต้นของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ที่มีการส่งข้อมูลความถี่สูง ส่วนฟังก์ชันที่ 4.5 แทนช่วงการส่งข้อมูลในช่วงปกติที่ส่งข้อมูลด้วยความถี่ที่ต่ำลง ดังนั้นการจำลองออนดีมานด์ทราฟฟิกต้องประกอบด้วยอ็อบเจกต์ที่สร้างทราฟฟิก 2 ส่วนด้วยกัน ดังรูปที่ 4.4 การจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วย 2 ฟังก์ชันทำให้สามารถปรับช่วงการส่งข้อมูลเริ่มต้นได้ตามต้องการ โดยขั้นตอนการทำงานของอ็อบเจกต์ทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์เป็นดังรูปที่ 4.5 ประกอบด้วยการกำหนดอัตราการนำส่งข้อมูลที่ต้องการ หลังจากนั้นฟังก์ชันจะทำการกำหนดค่าเริ่มต้นให้กับอ็อบเจกต์ด้วยช่วงการสุ่มของขนาดแพ็กเกตและช่วงเวลาห่างระหว่างแต่ละแพ็กเกต ซึ่งค่าดังกล่าวจะถูกกำหนดใน 2 อ็อบเจกต์คือ Burst และ Stored อ็อบเจกต์ด้วยค่าเริ่มต้นที่ต่างกันดังตารางที่ 4.2 หลังจากนั้นฟังก์ชันจะจำลองทราฟฟิกด้วยอ็อบเจกต์ Burst ก่อนจนกระทั่งถึงช่วงเวลาที่กำหนด ซึ่งช่วงเวลานี้สามารถปรับให้เหมาะสมได้ตามเงื่อนไขคุณสมบัติของเครื่องฝ่ายรับได้ตามต้องการ หลังจากนั้นอ็อบเจกต์ Stored จะทำหน้าที่สร้างทราฟฟิกแทนจนกระทั่งสิ้นสุดช่วงเวลาการส่งข้อมูลช่วงต้น ด้วยขั้นตอนดังกล่าวข้างต้นทำให้ทราฟฟิกที่ได้จากแบบจำลองมีลักษณะการส่งข้อมูลแบบคงที่ เช่นเดียวกับทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ที่ใช้ในระบบจริง

Application/Traffic/Burst object

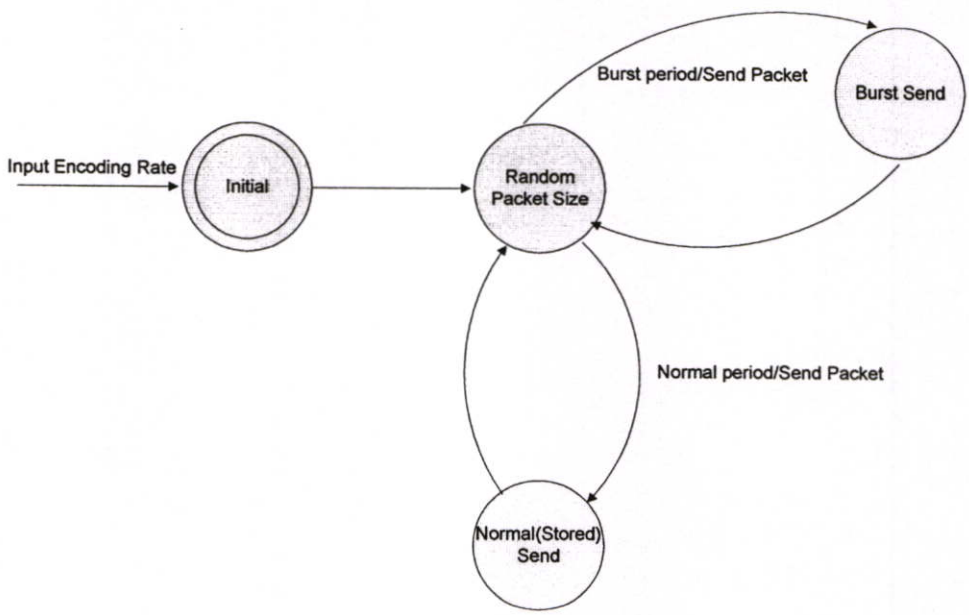
ค่าอินพุตของฟังก์ชัน : rate_ แทนอัตราการนำส่งข้อมูล (4.4)

Application/Traffic/Stored object

ค่าอินพุตของฟังก์ชัน : rate_ แทนอัตราการนำส่งข้อมูล (4.5)



รูปที่ 4.4 การจำลองรูปแบบของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์



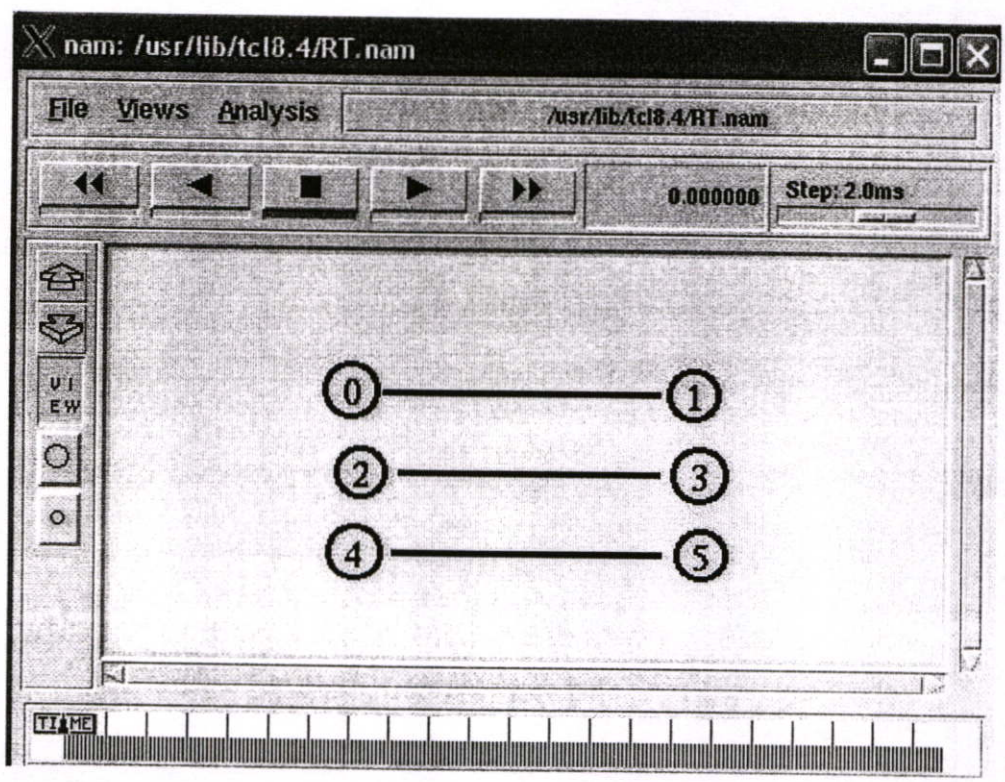
รูปที่ 4.5 ขั้นตอนการทำงานของแบบจำลองของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

จากฟังก์ชันที่พัฒนาขึ้นดังกล่าว ทำให้สามารถกำหนดรูปแบบของทราฟฟิกแบบต่อเนื่องทั้ง 2 ประเภทได้ใกล้เคียงมากขึ้น โดยในการจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ขึ้นนั้นสามารถใช้อ็อบเจกต์เรียลไทม์ได้โดยตรง แต่หากจะจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์จำเป็นต้องประกอบด้วยอ็อบเจกต์ 2 ส่วนเพื่อให้ได้รูปแบบของทราฟฟิกดังรูปที่ 4.4

4.4 ผลการทดลองการจำลองทราฟฟิกแบบต่อเนื่อง

4.4.1 รูปแบบเครือข่ายจำลองบน NS-2

การจำลองทราฟฟิกขึ้นบน NS-2 ใช้การเขียนโปรแกรมเพื่อทดลองสร้างทราฟฟิกแบบต่อเนื่องขึ้น โดยแบ่งการทดลองออกเป็น 2 ส่วนคือการทดลองจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์และประเภทออนดีมานด์ด้วยอัตราการนำส่ง 56 Kbps 128 Kbps และ 512 Kbps ซึ่งลักษณะการเชื่อมต่อในการจำลองระบบเครือข่ายเป็นไปดังรูปที่ 4.6 เช่นเดียวกับการทดลองกับทราฟฟิกแบบต่อเนื่องจริง เพื่อให้ได้ผลการทดลองที่สามารถเปรียบเทียบกันได้โดยตรงลักษณะการเชื่อมต่อจึงใช้การเชื่อมต่อเซิร์ฟเวอร์เข้ากับฝ่ายรับโดยตรงด้วยสายสัญญาณขนาดแบนด์วิดท์ 100 Mbps ประกอบด้วย 3 คู่การสื่อสารที่เชื่อมต่อกันด้วยสื่อสัญญาณ 100 Mbps โดยโหนด 0 2 และ 4 จำลองเป็นเซิร์ฟเวอร์ที่ให้บริการทราฟฟิกแบบต่อเนื่องด้วยอัตรา 56 Kbps 128 Kbps และ 512 Kbps ตามลำดับและกำหนดให้ทั้ง 3 โหนดส่งข้อมูลเป็นเวลา 5 นาที

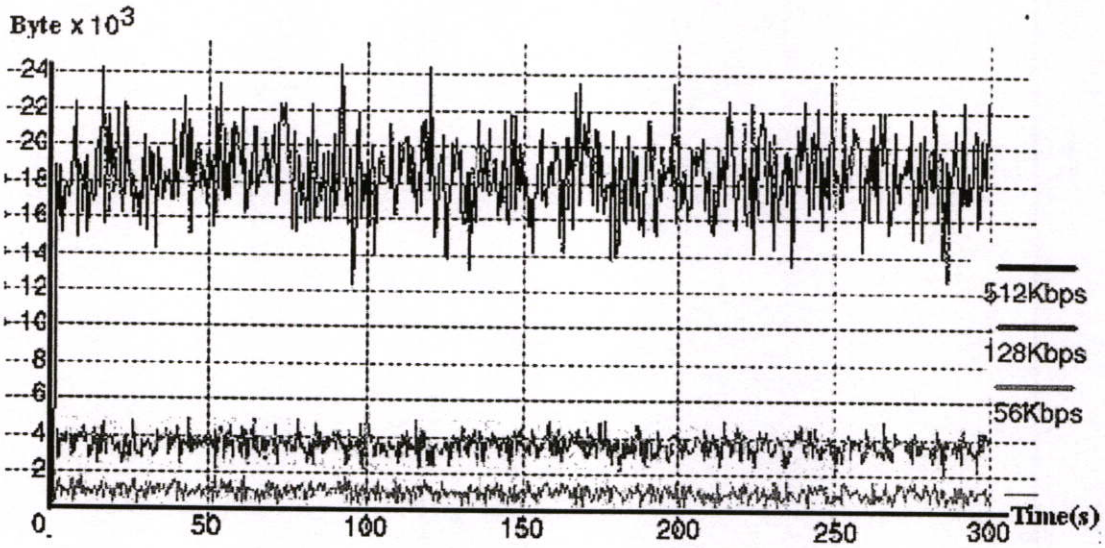


รูปที่ 4.6 รูปแบบการเชื่อมต่อในการจำลองทราฟฟิกแบบต่อเนื่อง

4.4.2 ผลการจำลองทราฟฟิกแบบต่อเนื่อง

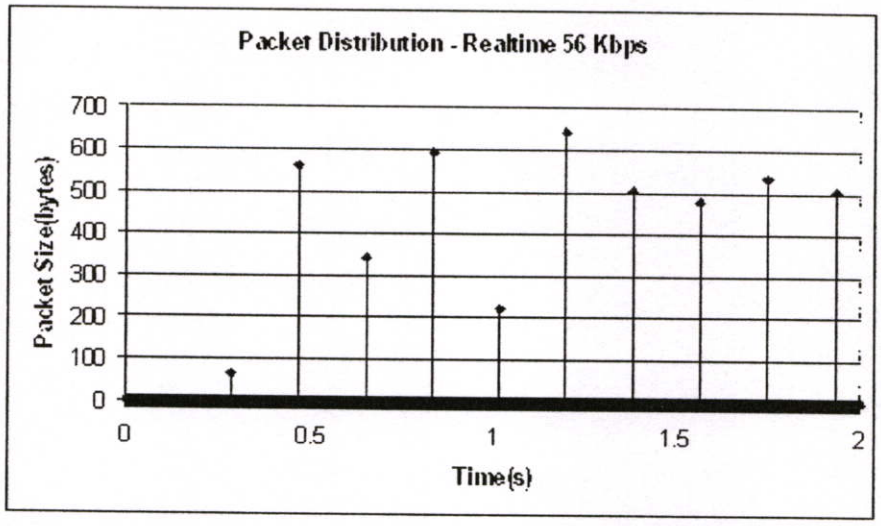
4.4.2.1 ผลการจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

จากการจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ ภาพรวมของพฤติกรรมทราฟฟิกเป็นดังรูปที่ 4.7 ซึ่งเมื่อพิจารณาลักษณะของทราฟฟิกที่ได้จะพบว่ามีการส่งข้อมูลต่อเนื่องตลอดช่วงการแสดงผลและลักษณะการส่งมีความแปรผันไปมาของทราฟฟิกค่อนข้างสูงเนื่องจากรูปแบบ ON-OFF ของทราฟฟิก

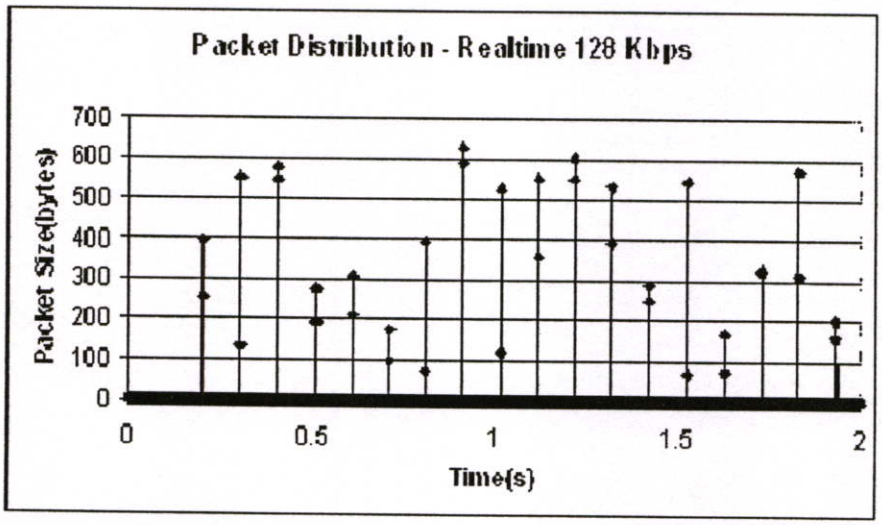


รูปที่ 4.7 ผลการทดลองจำลองทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

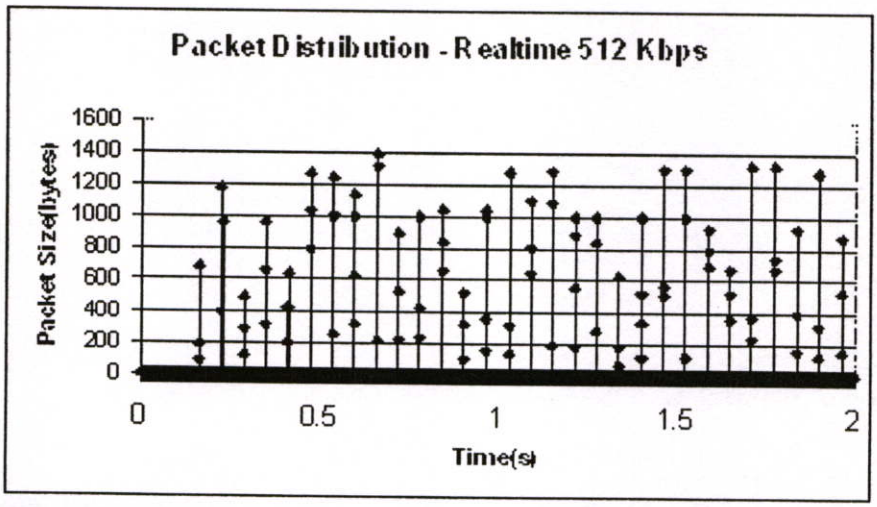
เมื่อนำผลของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ที่จำลองมาวิเคราะห์ในภาพที่เล็กลงด้วยช่วงการส่งข้อมูล 1-2 วินาที พบว่าลักษณะของทราฟฟิกเป็นลักษณะ ON-OFF ดังรูปที่ 4.8-4.10 โดยเมื่อเปรียบเทียบจากอัตราการนำส่งที่ต่างกันพบว่าอัตราการนำส่งที่สูงขึ้นจะทำให้ความถี่ในการส่งข้อมูลและขนาดของแพ็กเกตในรูปแบบ ON-OFF มีสูงขึ้น ซึ่งทำให้ส่งข้อมูลได้มากขึ้น จากกราฟรูปที่ 4.9 และ 4.10 แสดงให้เห็นว่ามีการส่งข้อมูลในช่วงเวลาที่ใกล้เคียงกันมากในช่วง ON จนทำให้เห็นเหมือนมีการส่งข้อมูลซ้อนกันในคาบเวลาเดียวกัน 2-3 ครั้ง ซึ่งแต่ละจุดบนกราฟแทนการส่งข้อมูล 1 ครั้ง นอกจากนั้นการส่งแต่ละอัตราการนำส่งถูกควบคุมค่าสูงสุดของขนาดแพ็กเกตด้วยค่าที่ต่าง ๆ กันอีกด้วยจากฟังก์ชันการสุ่มขนาดแพ็กเกต



รูปที่ 4.8 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตราการนำส่ง 56 Kbps



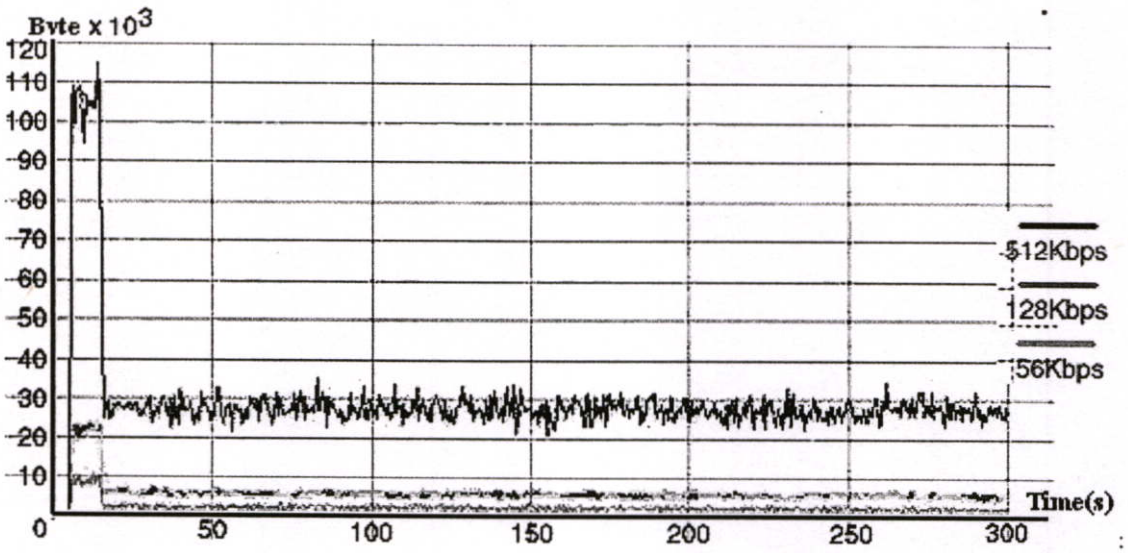
รูปที่ 4.9 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตราการนำส่ง 128 Kbps



รูปที่ 4.10 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตราการนำส่ง 512 Kbps

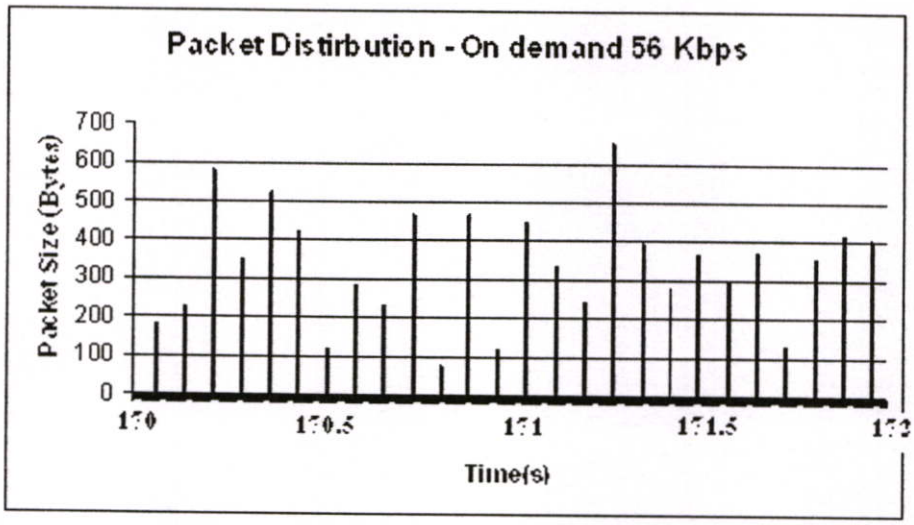
4.4.2.2 ผลการจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

จากการจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ ลักษณะภาพรวมของพฤติกรรมทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ที่จำลองขึ้นเป็นดังรูปที่ 4.11 ซึ่งต่างกับพฤติกรรมโดยรวมของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ เมื่อพิจารณาลักษณะของทราฟฟิกที่ได้จะพบว่ามีรูปแบบการส่งข้อมูลอย่างต่อเนื่องหลัก 2 ส่วนด้วยกัน ลักษณะการส่งมีความสม่ำเสมอตลอดช่วงการส่งข้อมูลเนื่องจากรูปแบบการส่งแบบคงที่ ซึ่งต่างจากประเภทเรียลไทม์ที่มีความแปรผันของพฤติกรรมการส่งทราฟฟิกสูงกว่า

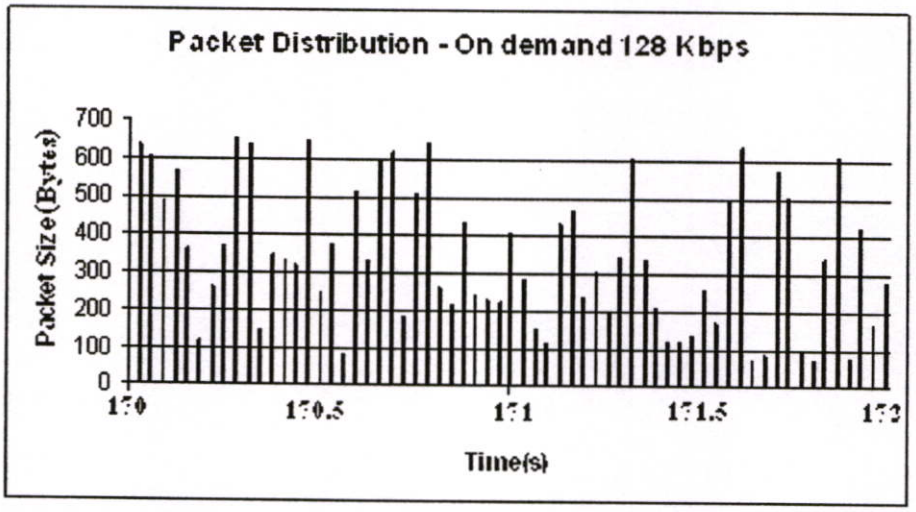


รูปที่ 4.11 ผลการทดลองจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

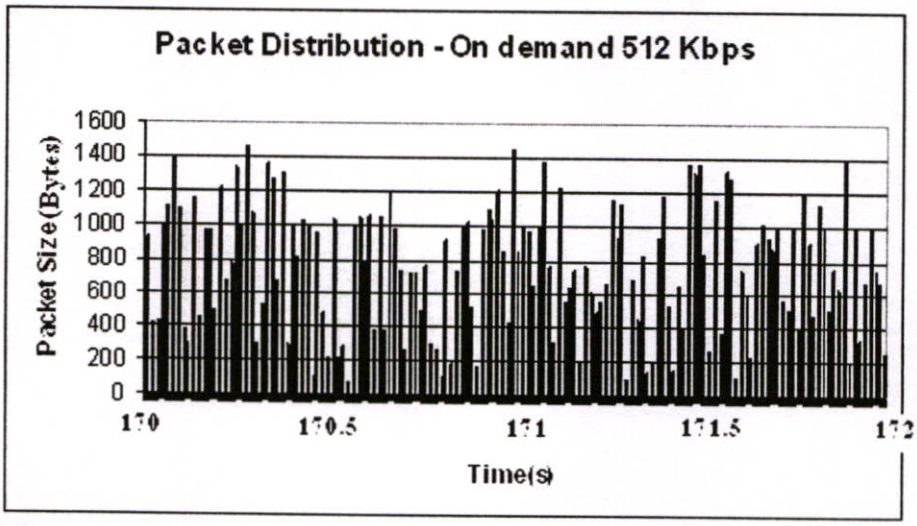
เมื่อพิจารณาด้วยภาพที่เล็กลง พบว่าลักษณะทราฟฟิกเป็นการส่งที่สม่ำเสมอไปตลอดช่วงการส่งข้อมูล โดยมีความสัมพันธ์กับอัตราการนำส่งข้อมูลเช่นเดียวกัน ซึ่งเมื่อข้อมูลถูกส่งด้วยอัตราการนำส่งที่สูงขึ้น ความถี่ในการส่งข้อมูลและขนาดของแพ็กเกตจะสูงขึ้นตาม ดังรูปที่ 4.12-4.14 เช่นเดียวกับทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ ขนาดสูงสุดของแพ็กเกตจะถูกควบคุมด้วยฟังก์ชันการสุ่มขนาดแพ็กเกตที่กำหนดอัตราการนำส่งข้อมูลที่ต่างกัน



รูปที่ 4.12 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตราการนำส่ง 56 Kbps



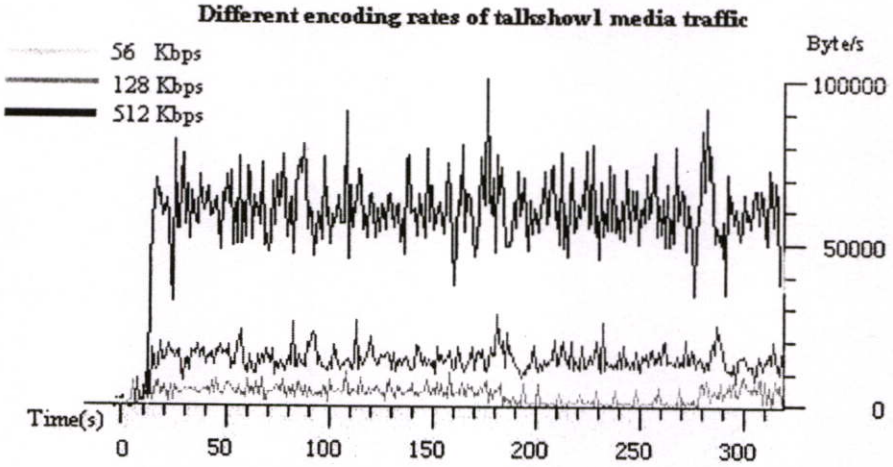
รูปที่ 4.13 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตราการนำส่ง 128 Kbps



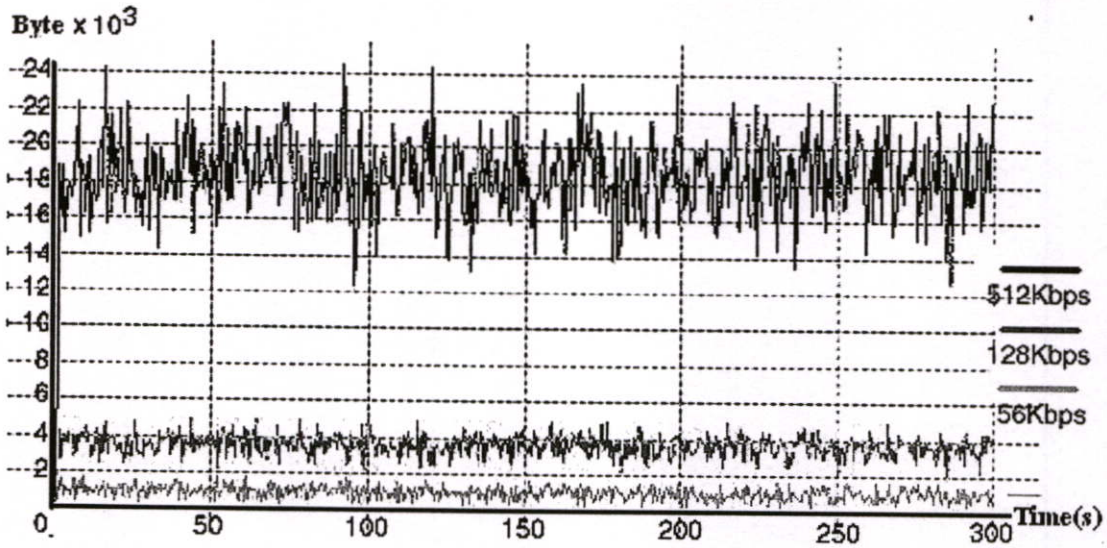
รูปที่ 4.14 ลักษณะการส่งทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตราการนำส่ง 512 Kbps

4.4.2.3 เปรียบเทียบผลการจำลองทราฟฟิกกับทราฟฟิกจริง

เมื่อเปรียบเทียบรูปแบบทราฟฟิกที่ได้จากการจำลองทราฟฟิกแบบต่อเนื่องบน NS-2 กับทราฟฟิกแบบต่อเนื่องจริงพบว่า เมื่อนำข้อมูลจากการวิเคราะห์ผ่านฟังก์ชันจำลองที่พัฒนาขึ้นจากรูปแบบ ON-OFF ทำให้ได้ทราฟฟิกจำลองออกมาในรูปแบบของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ ซึ่งเมื่อนำมาเปรียบเทียบด้วยภาพรวมแล้วพบว่ามีรูปแบบพฤติกรรมการส่งข้อมูลที่ใกล้เคียงกันดังรูปที่ 4.15



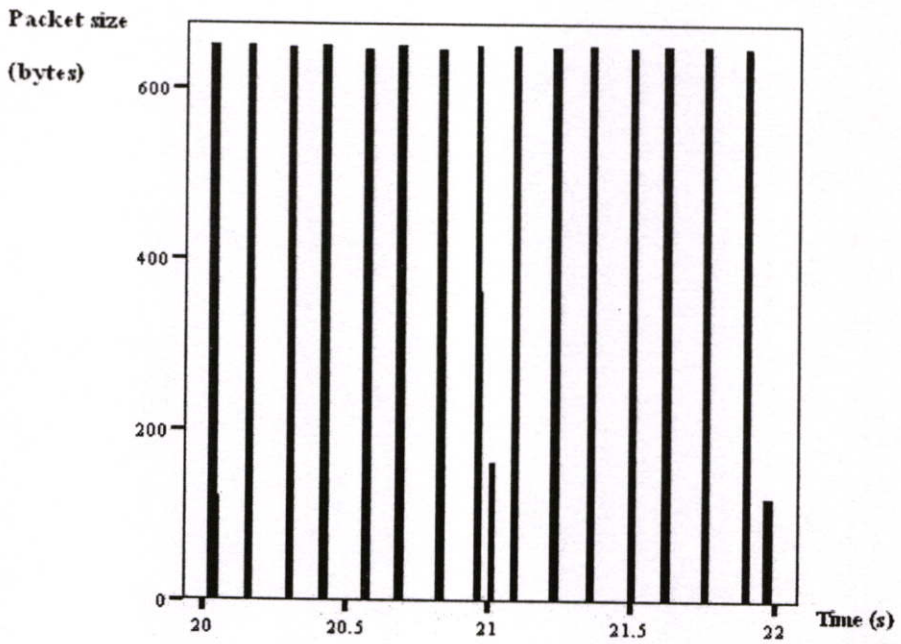
รูปที่ 4.15a รูปแบบทราฟฟิกจริงของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์



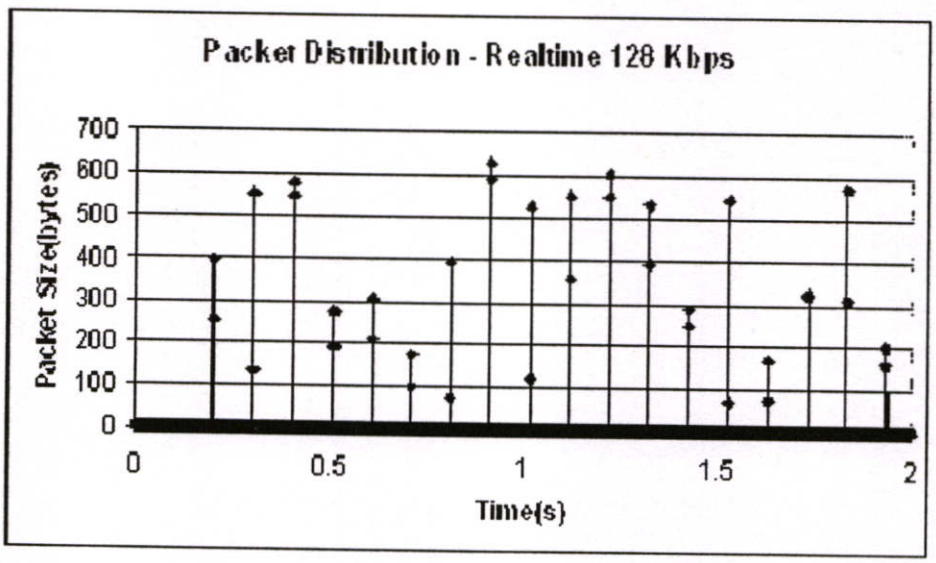
รูปที่ 4.15b รูปแบบทราฟฟิกที่จำลองขึ้นของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

รูปที่ 4.15 เปรียบเทียบพฤติกรรมของทราฟฟิกจริงและทราฟฟิกที่จำลองขึ้นจากทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

นอกจากนั้นหากนำข้อมูลภาพที่เล็กลงของรูปแบบทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ที่ใช้ในระบบจริงและที่ได้จากการจำลองมาเปรียบเทียบถึงรูปแบบการส่งข้อมูล รูปที่ 4.16 แสดงให้เห็นว่ารูปแบบการส่งข้อมูลในอัตราการนำส่งที่เท่ากันนั้นสร้างรูปแบบทราฟฟิกที่ใกล้เคียงกันในลักษณะการแบบ ON-OFF



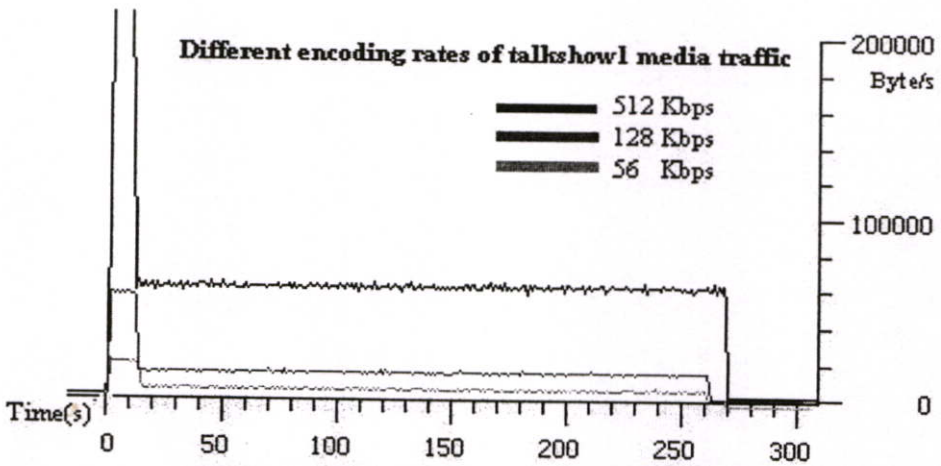
รูปที่ 4.16a รูปแบบทราฟฟิกจริงของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์



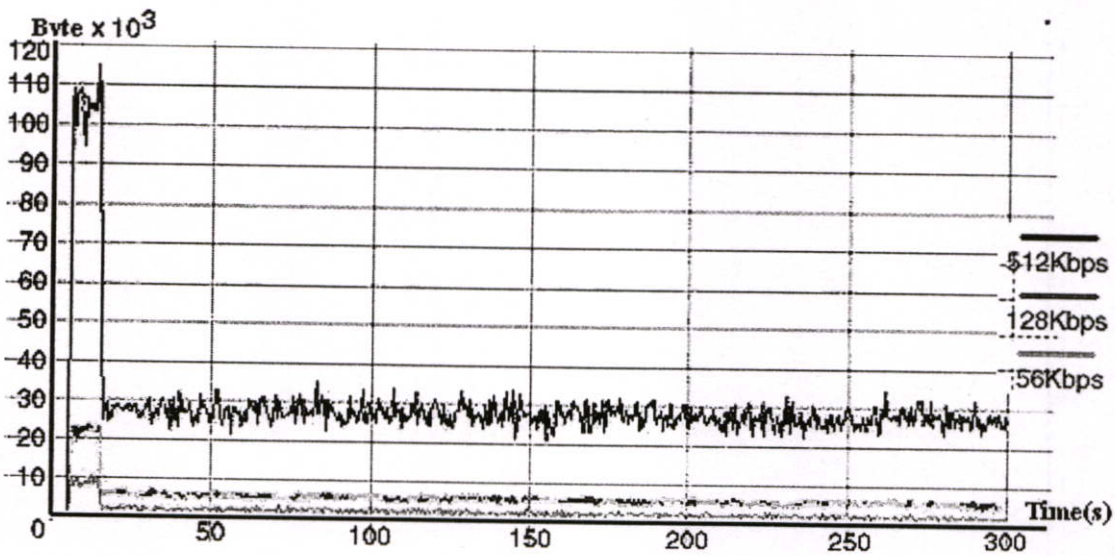
รูปที่ 4.16b รูปแบบทราฟฟิกจำลองของทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

รูปที่ 4.16 เปรียบเทียบพฤติกรรมของทราฟฟิกจริงและที่จำลองขึ้นจากทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ด้วยอัตราการนำส่ง 128 Kbps ในช่วงเวลาการส่งข้อมูลที่เล็กลง

สำหรับในรูปแบบของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ เมื่อเปรียบเทียบรูปแบบทราฟฟิกที่ได้จากการจำลองบน NS-2 กับทราฟฟิกแบบต่อเนื่องจริงด้วยวิธีเช่นเดียวกัน พบว่า เมื่อนำข้อมูลจากการวิเคราะห์ผ่านฟังก์ชันจำลองที่พัฒนาขึ้นจากรูปแบบการส่งคงที่ทำให้ได้ทราฟฟิกจำลองออกมาในรูปแบบของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ ซึ่งประกอบด้วยรูปแบบหลัก 2 ส่วนคือช่วงเริ่มต้นของการส่งข้อมูลและช่วงปกติของการส่งข้อมูลซึ่งแตกต่างกัน และเมื่อนำมาเปรียบเทียบด้วยภาพรวมแล้วพบว่ามีรูปแบบพฤติกรรมการส่งข้อมูลที่ใกล้เคียงกันดังรูปที่ 4.17



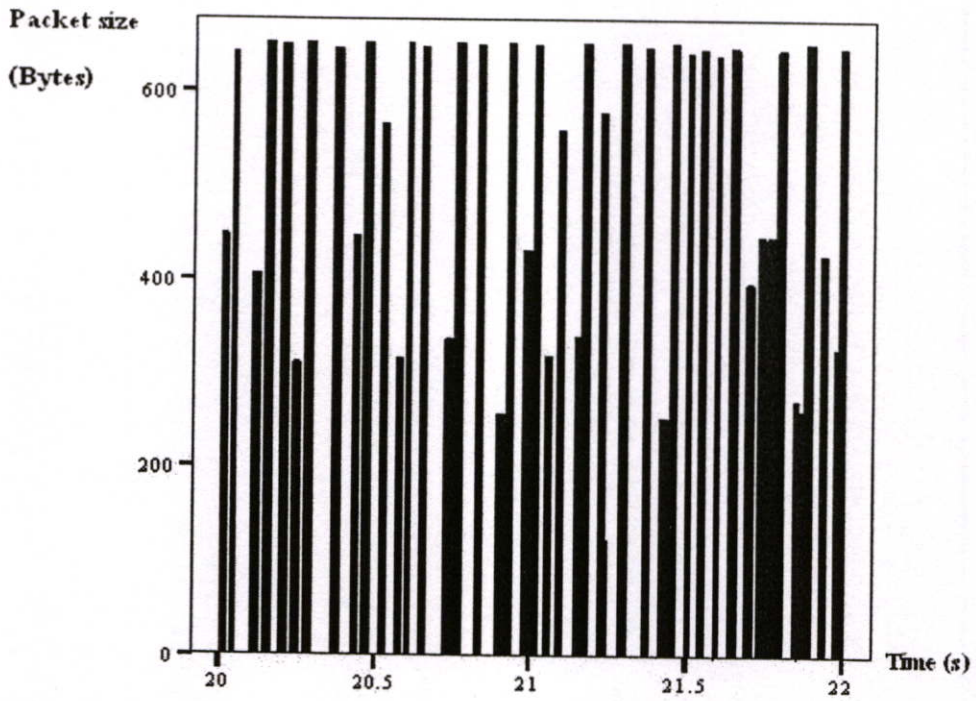
รูปที่ 4.17a รูปแบบทราฟฟิกจริงของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์



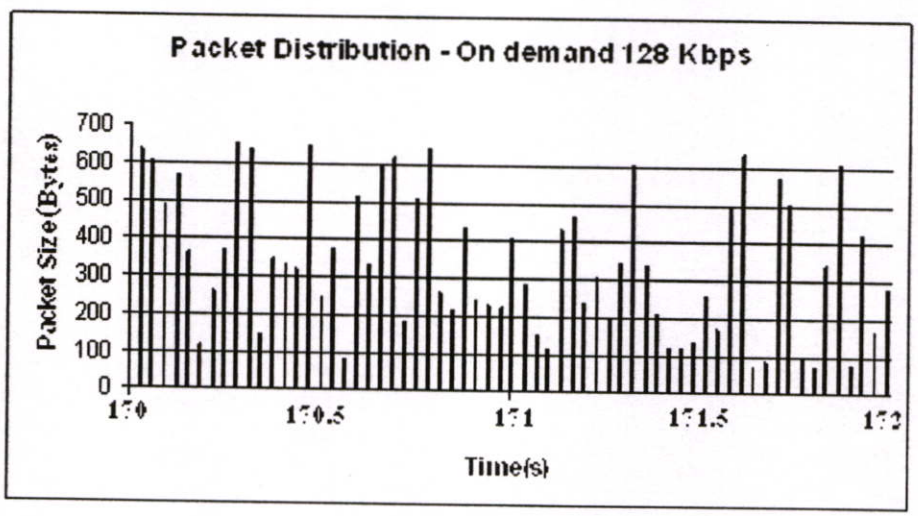
รูปที่ 4.17b รูปแบบทราฟฟิกที่จำลองขึ้นของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

รูปที่ 4.17 เปรียบเทียบพฤติกรรมของทราฟฟิกจริงและทราฟฟิกที่จำลองขึ้นจากทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

หากพิจารณาข้อมูลโดยละเอียดเปรียบเทียบระหว่างพฤติกรรมของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ที่ใช้ในระบบจริงและที่ได้จากการจำลอง รูปที่ 4.18 แสดงให้เห็นว่ารูปแบบการส่งข้อมูลในอัตราการนำส่งที่เท่ากันนั้นสร้างรูปแบบทราฟฟิกที่ใกล้เคียงกันในลักษณะการส่งคงที่



รูปที่ 4.18a รูปแบบทราฟฟิกจริงของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์



รูปที่ 4.18b รูปแบบทราฟฟิกจำลองของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

รูปที่ 4.18 เปรียบเทียบพฤติกรรมของทราฟฟิกจริงและที่จำลองขึ้นจากทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ด้วยอัตราการนำส่ง 128 Kbps ในช่วงเวลาการส่งข้อมูลที่เล็กลง

4.4.2.4 ความแตกต่างระหว่างทราฟฟิกแบบต่อเนื่องจริงกับทราฟฟิกจำลอง

เนื่องจากข้อจำกัดของการจำลองทราฟฟิก ซึ่งทำให้เกิดความแตกต่างบางส่วนระหว่างลักษณะทราฟฟิกแบบต่อเนื่องที่จำลองขึ้นกับทราฟฟิกแบบต่อเนื่องจริง โดยการกระจายขนาดแพ็กเกตของทราฟฟิกจริงนั้นมีความสัมพันธ์ขึ้นกับขนาดข้อมูลของไฟล์มัลติมีเดียที่เผยแพร่ออกมาด้วย ซึ่งต่างจากทราฟฟิกที่ได้การจำลองขึ้นในวิทยานิพนธ์นี้ที่มีการกระจายของขนาดแพ็กเกตเป็นรูปแบบชัดเจนด้วยฟังก์ชันที่ใช้เพื่อสุ่มขนาดของแพ็กเกตในอ็อบเจกต์ที่พัฒนาขึ้น

4.5 สรุปท้ายบท

จากผลการวิเคราะห์ทราฟฟิกแบบต่อเนื่องจากระบบจริงในบทที่ 3 ซึ่งทำให้สามารถระบุรูปแบบของทราฟฟิกและค่าคุณสมบัติต่าง ๆ ของทราฟฟิกแบบต่อเนื่องทั้ง 2 ประเภทนั้น ในบทนี้ได้นำผลดังกล่าวมาสร้างแบบจำลองของทราฟฟิกแบบต่อเนื่องขึ้นบน NS-2 ซึ่งเป็นโปรแกรมสำหรับจำลองระบบเครือข่าย โดยพัฒนาฟังก์ชันรูปแบบทราฟฟิกแบบต่อเนื่องให้สามารถกำหนดขนาดแพ็กเกตได้แบบสุ่ม นอกจากนั้นยังพัฒนาฟังก์ชันขึ้นให้ผู้ใช้สามารถนำไปใช้งานได้สะดวกขึ้น เพียงใส่อัตราการนำส่งที่ต้องการของทราฟฟิกแบบต่อเนื่องเท่านั้น ทำให้การศึกษา วิจัย และจำลองระบบเครือข่ายด้วยทราฟฟิกแบบต่อเนื่อง เป็นไปได้ง่ายและได้ผลการจำลองที่ถูกต้องใกล้เคียงกับระบบจริงมากยิ่งขึ้น

จากผลการทดลองจำลองทราฟฟิกแบบต่อเนื่องทั้ง 2 ประเภทและนำมาเปรียบเทียบกับทราฟฟิกแบบต่อเนื่องในระบบจริงพบว่ามีรูปแบบการส่งข้อมูลใกล้เคียงกัน แต่ทั้งนี้ยังมีข้อแตกต่างของแบบจำลองที่สร้างขึ้นกับทราฟฟิกจริงคือ ข้อมูลและขนาดของไฟล์มัลติมีเดียที่ทดลองจริงนั้นมีผลต่อขนาดแพ็กเกตจริงแต่ละแพ็กเกตที่ถูกส่งออกไป ซึ่งในแบบจำลองที่สร้างขึ้นนี้ใช้รูปแบบการสุ่มขนาดของแพ็กเกตด้วยฟังก์ชันการสุ่ม ทำให้ค่าที่ได้มีความแตกต่างกันอยู่บ้าง

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

วิทยานิพนธ์เล่มนี้นำเสนอรูปแบบของทราฟฟิกแบบต่อเนื่อง ซึ่งจากที่ได้ศึกษามานั้น ทราฟฟิกแบบต่อเนื่องยังไม่ได้แบ่งแยกประเภทไว้อย่างชัดเจน ในวิทยานิพนธ์เล่มนี้ได้เสนอถึงรูปแบบที่แตกต่างกันของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์และเรียลไทม์ โดยทั้ง 2 ประเภทนี้นอกจากจะแตกต่างกันด้านวัตถุประสงค์การใช้แล้ว ยังแตกต่างกันทางด้านพฤติกรรมการส่งข้อมูลอีกด้วย

จากผลการทดลอง เพื่อหาคุณลักษณะเฉพาะของทราฟฟิกแบบต่อเนื่องแต่ละประเภทพบว่า ทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์นั้นมีลักษณะการส่งในภาพรวมที่ต่อเนื่องไปตลอดช่วงการแสดงผลของข้อมูลที่เผยแพร่ออกไป โดยฝ่ายส่งเป็นฝ่ายควบคุมข้อมูลในการแสดงผลด้วยซึ่งต่างจากทราฟฟิกประเภทออนดีมานด์ที่มีลักษณะการส่งเหมือนการรับส่งไฟล์ทั่วไปมากกว่า ดังนั้นข้อมูลจะถูกส่งเข้าบัฟเฟอร์ของเครื่องผู้ใช้บริการและแสดงผล โดยควบคุมจากโปรแกรมในฝ่ายรับด้วยวิธีการดังกล่าวทำให้ผู้ใช้บริการสามารถควบคุมช่วงเวลาการแสดงผลได้ แต่การส่งลักษณะนี้ทำให้ปัจจัยจากคุณสมบัติของเครื่องผู้ใช้บริการมีผลต่อรูปแบบของทราฟฟิกด้วย

หากนำข้อมูลของทราฟฟิกทั้ง 2 ประเภทมาพิจารณาให้ละเอียดลงไปจะพบว่าสาเหตุหลักที่ทำให้ลักษณะการส่งทราฟฟิกทั้ง 2 ประเภทต่างกันคือ การควบคุมช่วงเวลาระยะห่างการส่งข้อมูลแต่ละแพ็กเกตของเซิร์ฟเวอร์ โดยเมื่อพิจารณารูปแบบของทราฟฟิกนั้น ทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์จะมีลักษณะการส่งแบบ ON-OFF ซึ่งถูกควบคุมโดยตรงจากเซิร์ฟเวอร์เพื่อควบคุมการแสดงผลของข้อมูลด้วย ต่างกับในลักษณะของทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ที่จะเป็นการส่งที่คงที่ตลอดช่วงของการส่งข้อมูลไปยังบัฟเฟอร์ของผู้ใช้บริการ สำหรับการกระจายของขนาดแพ็กเกตที่จะส่งนั้นจะถูกกำหนดด้วยขั้นตอนของการเข้ารหัสไฟล์ด้วยอัตราการนำส่งที่กำหนดตั้งแต่ต้น ดังนั้นขนาดของแพ็กเกตจึงไม่ได้ถูกกำหนดด้วยประเภทของทราฟฟิกที่ให้บริการแต่อย่างใด นอกจากนั้นผลการทดลองยังแสดงให้เห็นว่าอัตราการนำส่งข้อมูลมีผลต่อพฤติกรรมการส่งข้อมูลอีกด้วย เนื่องจากเมื่อนำข้อมูลมาเข้ารหัสด้วยอัตราการนำส่งที่สูงขึ้นทำให้ข้อมูลมีขนาดใหญ่ขึ้น เซิร์ฟเวอร์จึงปรับพฤติกรรมการส่งให้เหมาะสมกับข้อมูล โดยเมื่ออัตราการนำส่งสูงขึ้นขนาดแพ็กเกตที่ถูกส่งออกไปจะมีขนาดใหญ่ขึ้นและช่วงเวลาในการส่งข้อมูลแต่ละแพ็กเกตจะสั้นลงหรือถี่ขึ้นนั่นเอง ทำให้สามารถส่งข้อมูลได้มากขึ้น

จากผลการวิเคราะห์ข้างต้น ทำให้ได้คุณสมบัติและค่าอินพุตที่สามารถนำไปสร้างรูปแบบจำลองของทราฟฟิกแบบต่อเนื่องได้ ซึ่งในวิทยานิพนธ์นี้ได้สร้างรูปแบบจำลองขึ้นโดยใช้โปรแกรม NS-2 พัฒนาฟังก์ชันรูปแบบของทราฟฟิกให้ใกล้เคียงกับพฤติกรรมของทราฟฟิกแบบต่อเนื่อง โดยได้พัฒนาให้การส่งข้อมูลสามารถสุ่มขนาดของแพ็กเกตจากฟังก์ชันการสุ่มด้วยรูปแบบการกระจายแบบปกติทั้งทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์และออนดีมานด์ สำหรับการกำหนดช่วงเวลาการส่งแต่ละแพ็กเกตนั้นใช้การกระจายแบบคงที่ สำหรับแบบจำลองทราฟฟิกประเภทเรียลไทม์นั้นได้ใช้ฟังก์ชัน ON-OFF และใช้ฟังก์ชันคงที่สำหรับจำลองทราฟฟิกประเภทออนดีมานด์ โดยฟังก์ชันกำหนดค่าช่วงเวลาการส่งแพ็กเกตของทราฟฟิกประเภทออนดีมานด์นั้นจะประกอบด้วยฟังก์ชันคงที่ 2 ส่วนสำหรับสร้างทราฟฟิกในช่วงต้นที่มีความถี่ในการส่งข้อมูลสูงกับช่วงปกติ นอกจากนั้นแบบจำลองทราฟฟิกแบบต่อเนื่องนี้ยังได้พัฒนาให้กำหนดค่าเริ่มต้นได้ง่ายขึ้นเพียงกำหนดอัตราการนำส่งที่ต้องการและประเภทของทราฟฟิกเท่านั้น โดยจากผลการทดลองที่ได้เมื่อนำมาเปรียบเทียบกับรูปแบบทราฟฟิกจริง พบว่ามีพฤติกรรมการส่งข้อมูลที่ใกล้เคียงกัน โดยทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์ที่จำลองขึ้นมีรูปแบบการส่งเป็นแบบ ON-OFF ดังรูปที่ 4.16 และสำหรับทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์มีลักษณะการส่งคงที่ด้วยความถี่ที่แตกต่างกันในช่วงต้นกับช่วงปกติดังรูปที่ 4.18 แต่ขนาดของข้อมูลหรือขนาดของแพ็กเกตที่ได้นั้นยังคงแตกต่างกันอยู่บ้าง เนื่องจากปัจจัยของขนาดไฟล์และข้อมูลจากทราฟฟิกจริงนั้นจะมีผลต่อขนาดของแพ็กเกตที่ถูกส่งออกไป ซึ่งต่างจากลักษณะการกำหนดขนาดของแพ็กเกตด้วยฟังก์ชันการสุ่มในแบบจำลองที่สร้างขึ้น

จากคุณสมบัติและรูปแบบจำลองของทราฟฟิกแบบต่อเนื่องที่ได้จากงานวิจัยในวิทยานิพนธ์เล่มนี้ ทำให้สามารถเข้าใจคุณลักษณะของทราฟฟิกแบบต่อเนื่อง รวมถึงปัจจัยต่าง ๆ ที่มีผลต่อพฤติกรรมการส่งทราฟฟิกมากยิ่งขึ้น นอกจากนั้นยังสามารถใช้รูปแบบจำลองที่พัฒนาขึ้นเพื่อศึกษาวิจัยและออกแบบเครือข่ายจำลองที่ให้รูปแบบของทราฟฟิกแบบต่อเนื่องที่ใกล้เคียงกับที่เกิดขึ้นในระบบจริง ซึ่งแบบจำลองดังกล่าวสามารถช่วยให้กำหนดค่าตัวแปรต่าง ๆ และเลือกอัตราการนำส่งของข้อมูลได้สะดวกยิ่งขึ้นอีกด้วย

เอกสารอ้างอิง

[1] Mingzhe Li, Mark Claypool, Robert Kinicki, and James Nichols. 2003. "Characteristics of Streaming Media Stored on the Internet". **Tech. Rep. WPI-CS-TR-03-18.** CS Department, Worcester Polytechnic Institute,

[2] Kuang, Tianbo, Williamson, Carey L. 2002. "Measurement study of RealMedia streaming traffic." 68-79. in **Proc. SPIE Vol. 4865 Internet Performance and Control of Network Systems III.**

[3] M. Li, M. Claypool, and B. Kinicki. 2002. "MediaPlayer versus RealPlayer --- a comparison of network turbulence" **In Proceedings of ACM Internet Measurement Workshop.**

[4] Real Networks Inc. **"RealNetwork Tutorials".** http://www.realnetworks.com/resources/tutorials/

[5] The Internet Engineering Task Force. **"Realtime Streaming Protocol (RTSP)".** http://www.ietf.org/rfc/rfc2326.txt

[6] The Internet Engineering Task Force. **"RTP: A Transport Protocol for Real-Time ApplicationsRTP".** http://www.faqs.org/rfcs/rfc1889.html

[7] Art Mena and John Heidemann. 2000. "An Empirical Study of Real Audio Traffic." 101-110. **In Proceedings of the IEEE INFOCOM.** Tel-Aviv, Israel.

[8] BBC News. **"BBC News : Video and audio"** http://news.bbc.co.uk/

[9] Information Sciences Institute. **"Network Simulator-2."** http://www.isi.edu/nsnam/ns/

[10] OPNET Technologies. **"Opnet Modeler."** http://www.opnet.com

[11] ADOBE. **"Video codec Compression methods"** ADOBE PREMIER TECHNICAL GUIDE. http://teched.vt.edu/gcc/html/VirtualTextbook/PDFs/ AdobeTutorialsPDFs/ Premiere/ PremiereVideoCodecMethods.pdf

[12] Marieke Guy. **"Choosing A Suitable Digital Video Format"** http://www.ukoln.ac.uk/qa-focus/documents/briefings/briefing-25/html/. UKOLN.

ภาคผนวก ก

โปรแกรมฟังก์ชันทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ในช่วงเริ่มต้น

```
/* -*-    Mode:C++; c-basic-offset:8; tab-width:8; indent-tabs-mode:t -*- */
/* Module of Streaming on-demand Traffic in the beginning of transfer process*/

#include <stdlib.h>
#include "random.h"
#include "trafgen.h"
#include "ranvar.h"

// random noise in the interval, and packet size
class burst_traffic : public TrafficGenerator {
 public:
        burst_traffic();
        virtual double next_interval(int&);
        inline double interval() { return (interval_); }
 protected:
        virtual void start();
        void init();
        double rate_;    /* send rate during on time (bps) */
        double interval_; /* packet inter-arrival time during burst (sec) */
        double random_;
        int seqno_;
        int maxpkts_;
};
static class BurstTrafficClass : public TclClass {
 public:
```

```
        BurstTrafficClass() : TclClass("Application/Traffic/Burst") {}
        TclObject* create(int, const char*const*) {
                return (new burst_traffic());
        }
} class_burst_traffic;
burst_traffic::burst_traffic() : seqno_(0)
{
        bind_bw("rate_", &rate_);
        bind("random_", &random_);
        //bind("packetSize_", &size_);
        bind("maxpkts_", &maxpkts_);
}
void burst_traffic::init()
{

    // compute inter-packet interval
        if (rate_ == 56000)
                interval_ = 0.02058;
        else if (rate_ == 128000)
                interval_ = 0.00833;
        else
                interval_ = 0.003628;
        if (agent_)
                if (agent_->get_pkttype() != PT_TCP &&
                  agent_->get_pkttype() != PT_TFRC)
                        agent_->set_pkttype(PT_CBR);
}
void burst_traffic::start()
{
    init();
    running_ = 1;
    timeout();
```

```
}
double burst_traffic::next_interval(int& size)   // Compute interval time and random packet size
{
        if (rate_ == 56000) {
                interval_ = 0.02058;
                size_ = Random::uniform(66, 659); }
        else if (rate_ == 128000) {
                interval_ = 0.00833;
                size_ = Random::uniform(66, 661); }
        else {
                interval_ = 0.003628;
                size_ = Random::uniform(66, 1461); }
        double t = interval_;
        if (random_)
                t += interval_ * Random::uniform(-0.01, 0.01);
        size = size_;
        if (++seqno_ < maxpkts_)
                return(t);          else
                return(-1); }
```

โปรแกรมฟังก์ชันทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์ในช่วงปกติ

```
/* -*-    Mode:C++; c-basic-offset:8; tab-width:8; indent-tabs-mode:t -*- */
/* Module of Streaming on-demand Traffic in the normal transfer process*/

#include <stdlib.h>
#include "random.h"
#include "trafgen.h"
#include "ranvar.h"

// random noise in the interval, and packet size.
class stored_media : public TrafficGenerator {
 public:
        stored_media();
        virtual double next_interval(int&);
        //HACK so that udp agent knows interpacket arrival time within a burst
        inline double interval() { return (interval_); }
 protected:
        virtual void start();
        void init();
        double rate_;    /* send rate during on time (bps) */
        double interval_; /* packet inter-arrival time during burst (sec) */
        double random_;
        int seqno_;
        int maxpkts_;
};

static class StoredTrafficClass : public TclClass {
```

```
 public:
        StoredTrafficClass() : TclClass("Application/Traffic/Stored") {}
        TclObject* create(int, const char*const*) {
                return (new stored_media());
        }
} class_stored_media;

stored_media::stored_media() : seqno_(0)
{
        bind_bw("rate_", &rate_);
        bind("random_", &random_);
        bind("maxpkts_", &maxpkts_);
}

void stored_media::init()
{
     // compute inter-packet interval
        if (rate_ == 56000)
                interval_ = 0.075997;
        else if (rate_ == 128000)
                interval_ = 0.031572;
        else
                interval_ = 0.013933;
        if (agent_)
                if (agent_->get_pkttype() != PT_TCP &&
                  agent_->get_pkttype() != PT_TFRC)
                        agent_->set_pkttype(PT_CBR);
}

void stored_media::start()
{
     init();
```

```
        running_ = 1;
        timeout();
}

double stored_media::next_interval(int& size)
{
        // compute inter-packet interval

        if (rate_ == 56000) {
                interval_ = 0.075997;
                size_ = Random::uniform(66, 659); }
        else if (rate_ == 128000) {
                interval_ = 0.031572;
                size_ = Random::uniform(66, 661); }
        else {
                interval_ = 0.013933;
                size_ = Random::uniform(66, 1461); }
        double t = interval_;
        if (random_)
                t += interval_ * Random::uniform(-0.01, 0.01);
        size = size_;
        if (++seqno_ < maxpkts_)
                return(t);
        else
                return(-1);
}
```

โปรแกรมฟังก์ชันทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

```
/* -*-    Mode:C++; c-basic-offset:8; tab-width:8; indent-tabs-mode:t -*- */

#ifndef lint
static const char rcsid[] =
   "@(#) $Header: /nfs/jade/vint/CVSROOT/ns-2/tools/expoo.cc,v 1.15 2005/08/26 05:05:30 tomh
Exp $ (Xerox)";
#endif
#include <stdlib.h>
#include "random.h"
#include "trafgen.h"
#include "ranvar.h"

/* implement an on/off source with exponentially distributed on and
 * off times.  parameterized by average burst time, average idle time,
 * burst rate and packet size.
 */

class Realtime_Traffic : public TrafficGenerator {
public:
        Realtime_Traffic();
        virtual double next_interval(int&);
     virtual void timeout();
        // Added by Debojyoti Dutta October 12th 2000
        int command(int argc, const char*const* argv);
protected:
        void init();
        double ontime_;  /* average length of burst (sec) */
        double offtime_; /* average length of idle time (sec) */
        double rate_;    /* send rate during on time (bps) */
```

```
        double interval_; /* packet inter-arrival time during burst (sec) */
        unsigned int rem_; /* number of packets left in current burst */

        /* new stuff using RandomVariable */
        ExponentialRandomVariable burstlen_;
        ExponentialRandomVariable Offtime_;
};

static class RealtimeTrafficClass : public TclClass {
 public:
        RealtimeTrafficClass() : TclClass("Application/Traffic/Realtime") {}
        TclObject* create(int, const char*const*) {
                return (new Realtime_Traffic());
        }
} class_Realtime_traffic;

// Added by Debojyoti Dutta October 12th 2000
// This is a new command that allows us to use
// our own RNG object for random number generation
// when generating application traffic

int Realtime_Traffic::command(int argc, const char*const* argv){

    if(argc==3){
        if (strcmp(argv[1], "use-rng") == 0) {
            burstlen_.seed((char *)argv[2]);
            Offtime_.seed((char *)argv[2]);
            return (TCL_OK);
        }
    }
    return Application::command(argc,argv);
}
```

```
Realtime_Traffic::Realtime_Traffic() : burstlen_(0.0), Offtime_(0.0)
{
        //bind_time("burst_time_", &ontime_);
        //bind_time("idle_time_", Offtime_.avgp());
        bind_bw("rate_", &rate_);
        //bind("packetSize_", &size_);
}

void Realtime_Traffic::init()
{
    /* compute inter-packet interval during bursts based on
        * packet size and burst rate.  then compute average number
        * of packets in a burst.
        */
        //interval_ = (double)(size_ << 3)/(double)rate_;
        if (rate_ == 56000) {
                interval_ = 0.000177;
                ontime_ = 0.0002655;
                rem_ = 3; }
        else if (rate_ == 128000) {
                interval_ = 0.00025443;
                ontime_ = 0.0005877;
                rem_ = 4; }
        else {
                interval_ = 0.0002407;
                ontime_ = 0.0008204;
                rem_ = 5; }
        burstlen_.setavg(ontime_/interval_);
        rem_ = 0;
        //if (agent_)
                agent_->set_pkttype(PT_CBR);
```

```
                //PT_EXP
}

double Realtime_Traffic::next_interval(int& size)
{
        double t = interval_;

        if (rate_ == 56000) {
                offtime_ = 0.183;
                size_ = Random::uniform(66, 659);}
        else if (rate_ == 128000) {
                offtime_ = 0.10135;
                size_ = Random::uniform(66, 661);}
        else {
                offtime_ = 0.06137;
                size_ = Random::uniform(66, 1461);}

        if (rem_ == 0) {
                /* compute number of packets in next burst */
                //rem_ = int(burstlen_.value() + .5);
                rem_ = int(ontime_/interval_);
                /* make sure we got at least 1 */
                if (rem_ == 0)
                        rem_ = 1;
                /* start of an idle period, compute idle time */
        //t = Offtime_.value();
        t += offtime_;
        }
        rem_--;

        size = size_;
        return(t);
```

```
}

void Realtime_Traffic::timeout()
{
        if (! running_)
                return;

        /* send a packet */
        // The test tcl/ex/test-rcvr.tcl relies on the "NEW_BURST" flag being
        // set at the start of any exponential burst ("talkspurt").
        if (nextPkttime_ != interval_ || nextPkttime_ == -1)
                agent_->sendmsg(size_, "NEW_BURST");
        else
                agent_->sendmsg(size_);
        /* figure out when to send the next one */
        nextPkttime_ = next_interval(size_);
        /* schedule it */
        if (nextPkttime_ > 0)
                timer_.resched(nextPkttime_);
}
```

ภาคผนวก ข

โปรแกรมจำลองเครือข่ายทราฟฟิกแบบต่อเนื่องประเภทเรียลไทม์

```
#Create a simulator object
set ns [new Simulator]
# Create a nam trace datafile.
set namfile [open /usr/lib/tcl8.4/RT.nam w]
$ns namtrace-all $namfile

#Open the output files
set f0 [open RT56.tr w]
set f1 [open RT128.tr w]
set f2 [open RT512.tr w]

#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]

set n2 [$ns node]
set n3 [$ns node]

set n4 [$ns node]
set n5 [$ns node]

#Connect the nodes
$ns duplex-link $n0 $n1 100Mb 100ms DropTail
$ns duplex-link $n2 $n3 100Mb 100ms DropTail
$ns duplex-link $n4 $n5 100Mb 100ms DropTail
```

```
#Define a 'finish' procedure
proc finish {} {
        global f0 f1 f2 ns namfile
        #Close the output files
        close $f0
        close $f1
        close $f2

        #Call xgraph to display the results
        exec xgraph RT56.tr RT128.tr RT512.tr -geometry 800x400 &
        $ns flush-trace
        close $namfile
        exec nam -r 2000.000000us /usr/lib/tcl8.4/RT.nam &
    exit 0
}


#Define a procedure that attaches a UDP agent to a previously created node
#'node' and attaches an Realtime traffic generator to the agent with the
#characteristic values 'rate'

proc attach-expoo-traffic-UDP { node sink rate} {
        #Get an instance of the simulator
        set ns [Simulator instance]

        #Create a UDP agent and attach it to the node
        set source [new Agent/UDP]
        $ns attach-agent $node $source

        #Create an Expoo traffic agent and set its configuration parameters
        set traffic [new Application/Traffic/Realtime]
        $traffic set rate_ $rate
```

```
    # Attach traffic source to the traffic generator
    $traffic attach-agent $source
        #Connect the source and the sink
        $ns connect $source $sink
        return $traffic
}

#Define a procedure which periodically records the bandwidth received by the
#three traffic sinks sink0/1/2 and writes it to the three files f0/1/2.
proc record {} {
    global sink0 sink1 sink2 f0 f1 f2
        #Get an instance of the simulator
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
    set time 0.5
        #How many bytes have been received by the traffic sinks?
    set bw0 [$sink0 set bytes_]
                set bw1 [$sink1 set bytes_]
                set bw2 [$sink2 set bytes_]

        #Get the current time
    set now [$ns now]
        #Calculate the bandwidth (in MBit/s) and write it to the files
    puts $f0 "$now [expr $bw0]"
    puts $f1 "$now [expr $bw1]"
    puts $f2 "$now [expr $bw2]"

        #Reset the bytes_ values on the traffic sinks
    $sink0 set bytes_ 0
    $sink1 set bytes_ 0
    $sink2 set bytes_ 0
```

```
        #Re-schedule the procedure
    $ns at [expr $now+$time] "record"
}

#Create three traffic sinks and attach them to the node n4
set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]
set sink2 [new Agent/LossMonitor]

$ns attach-agent $n1 $sink0
$ns attach-agent $n3 $sink1
$ns attach-agent $n5 $sink2
#Create three traffic sources
set source0 [attach-expoo-traffic-UDP $n0 $sink0 56Kb]
set source1 [attach-expoo-traffic-UDP $n2 $sink1 128Kb]
set source2 [attach-expoo-traffic-UDP $n4 $sink2 512Kb]
#Start logging the received bandwidth
$ns at 0.0 "record"
#Start the traffic sources
$ns at 5.0 "$source0 start"
$ns at 5.0 "$source1 start"
$ns at 5.0 "$source2 start"
#Stop the traffic sources
$ns at 300.0 "$source0 stop"
$ns at 300.0 "$source1 stop"
$ns at 300.0 "$source2 stop"
#Call the finish procedure after 60 seconds simulation time
$ns at 300.0 "finish"
#Run the simulation
$ns run
```

โปรแกรมจำลองทราฟฟิกแบบต่อเนื่องประเภทออนดีมานด์

```
#Create a simulator object
set ns [new Simulator]

# Create a nam trace datafile.
set namfile [open /usr/lib/tcl8.4/Store.nam w]
$ns namtrace-all $namfile

#Open the output files
set f0 [open ST56.tr w]
set f1 [open ST128.tr w]
set f2 [open ST512.tr w]

#Create 6 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

#Connect the nodes
$ns duplex-link $n0 $n1 100Mb 100ms DropTail
$ns duplex-link $n2 $n3 100Mb 100ms DropTail
$ns duplex-link $n4 $n5 100Mb 100ms DropTail

#Define a 'finish' procedure
proc finish {} {
        global f0 f1 f2 ns namfile
        #Close the output files
```

```
        close $f0
        close $f1
        close $f2

        #Call xgraph to display the results
        exec xgraph ST56.tr ST128.tr ST512.tr -geometry 800x400 &
                $ns flush-trace
        close $namfile
        exec nam -r 2000.000000us /usr/lib/tcl8.4/Store.nam &
    exit 0
}

#Define a procedure that attaches a UDP agent to a previously created node
#'node' and attaches an normal on-demand traffic generator to the agent with the
#characteristic values 'rate'

proc attach-expoo-traffic-UDP { node sink rate } {
        #Get an instance of the simulator
        set ns [Simulator instance]

        #Create a UDP agent and attach it to the node
        set source [new Agent/UDP]
        $ns attach-agent $node $source

        #Create an Expoo traffic agent and set its configuration parameters
        set traffic [new Application/Traffic/Stored]
        $traffic set rate_ $rate

    # Attach traffic source to the traffic generator
    $traffic attach-agent $source
        #Connect the source and the sink
        $ns connect $source $sink
```

```
        return $traffic
}

#Define a procedure that attaches a UDP agent to a previously created node
#'node' and attaches an Burst on-demand traffic generator to the agent with the
#characteristic values 'rate'
proc attach-expoo-traffic-burst { node sink rate } {
        #Get an instance of the simulator
        set ns [Simulator instance]

        #Create a UDP agent and attach it to the node
        set source [new Agent/UDP]
        $ns attach-agent $node $source

        #Create an Expoo traffic agent and set its configuration parameters
        set traffic [new Application/Traffic/Burst]
        $traffic set rate_ $rate

    # Attach traffic source to the traffic generator
    $traffic attach-agent $source
        #Connect the source and the sink
        $ns connect $source $sink
        return $traffic
}

#Define a procedure which periodically records the bandwidth received by the
#three traffic sinks sink0/1/2 and writes it to the three files f0/1/2.
proc record {} {
    global sink0 sink1 sink2 f0 f1 f2
        #Get an instance of the simulator
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
```

```
    set time 0.5
        #How many bytes have been received by the traffic sinks?
    set bw0 [$sink0 set bytes_]
                set bw1 [$sink1 set bytes_]
                set bw2 [$sink2 set bytes_]

        #Get the current time
    set now [$ns now]
        #Calculate the bandwidth (in MBit/s) and write it to the files
    puts $f0 "$now [expr $bw0]"
    puts $f1 "$now [expr $bw1]"
    puts $f2 "$now [expr $bw2]"

        #Reset the bytes_ values on the traffic sinks
    $sink0 set bytes_ 0
    $sink1 set bytes_ 0
    $sink2 set bytes_ 0

        #Re-schedule the procedure
    $ns at [expr $now+$time] "record"
}

#Create three traffic sinks and attach them to the node n4
set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]
set sink2 [new Agent/LossMonitor]

$ns attach-agent $n1 $sink0
$ns attach-agent $n3 $sink1
$ns attach-agent $n5 $sink2

#Create three traffic sources
```

```
set source0 [attach-expoo-traffic-burst $n0 $sink0 56Kb]
set source1 [attach-expoo-traffic-burst $n2 $sink1 128Kb]
set source2 [attach-expoo-traffic-burst $n4 $sink2 512Kb]

set source0_1 [attach-expoo-traffic-UDP $n0 $sink0 56Kb]
set source1_1 [attach-expoo-traffic-UDP $n2 $sink1 128Kb]
set source2_1 [attach-expoo-traffic-UDP $n4 $sink2 512Kb]

#Start logging the received bandwidth
$ns at 0.0 "record"
#Start the traffic sources
$ns at 5.0 "$source0 start"
$ns at 5.0 "$source1 start"
$ns at 5.0 "$source2 start"

#Stop the traffic sources
$ns at 15.0 "$source0 stop"
$ns at 15.0 "$source1 stop"
$ns at 15.0 "$source2 stop"

$ns at 15.0 "$source0_1 start"
$ns at 15.0 "$source1_1 start"
$ns at 15.0 "$source2_1 start"

$ns at 300.0 "$source0_1 stop"
$ns at 300.0 "$source1_1 stop"
$ns at 300.0 "$source2_1 stop"

#Call the finish procedure after 300 seconds simulation time
$ns at 300.0 "finish"
#Run the simulation
$ns run
```

ภาคผนวก ค

การคอมไพล์อ็อบเจกต์บน NS-2

ขั้นตอนการคอมไพล์อ็อบเจกต์ที่สร้างหรือพัฒนาขึ้นบน NS-2 เพื่อนำมาใช้มีดังนี้

1. สร้างหรือแก้ไขไฟล์อ็อบเจกต์ที่ต้องการซึ่งเป็นไฟล์นามสกุล cc
2. แก้ไขหรือเพิ่มค่าชื่ออ็อบเจกต์ในไฟล์ Packet.h โดยสร้างชื่อที่ต้องการเรียกใช้ เช่น

```
PT_STORED,
```

และ

```
name_[PT_REALTIME]= "Realtime";
```

3. กำหนดค่าเริ่มต้นของอ็อบเจกต์ที่สร้างขึ้นที่ไฟล์ tcl/lib/ns-default.tcl ตัวอย่างเช่น

```
Application/Traffic/Realtime set burst_time_ .5
Application/Traffic/Realtime set idle_time_ .5
Application/Traffic/Realtime set rate_ 128Kb
Application/Traffic/Realtime set packetSize_ 2
```

4. แก้ไข Makefile สคริปต์เพื่อสั่งให้คอมไพล์ไฟล์ที่สร้างขึ้น ตัวอย่างเช่น

```
tools/stored_media.o tools/realtime.o tools/burst.o \
```

5. คอมไพล์ไฟล์ที่ต้องการด้วยคำสั่ง “make depend; make”
6. เรียกใช้อ็อบเจกต์ที่สร้างขึ้นเพื่อจำลองระบบเครือข่ายได้ตามชื่อที่กำหนด

ประวัติผู้เขียน

นายสราวุธ อมราสิงห์ เกิดเมื่อวันที่ 11 สิงหาคม 2522 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2544 และเข้ารับการศึกษาในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อปีการศึกษา 2544