กลไกการจองทรัพยากรที่มีประสิทธิภาพในขั้นตอนการโพรบสำหรับเครือข่ายมัลติมีเดียแบบยูนิแคสต์

AN EFFICIENT RESOURCE RESERVATION MECHANISM IN PROBING PHASE FOR UNICAST MULTIMEDIA NETWORKS



ไกรศักดิ์ เกษร

KRAISAK KESORN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

ISBN 974-324-351-8

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

กลไกการจองทรัพยากรที่มีประสิทธิภาพในขั้นตอนการโพรบสำหรับเครือข่ายมัลติมีเดียแบบยูนิแคสต์

AN EFFICIENT RESOURCE RESERVATION MECHANISM IN PROBING PHASE FOR UNICAST MULTIMEDIA NETWORKS

T047712

ไกรศักดิ์ เกษร

KRAISAK KESORN

เลขหมู่.....
เลขทะเบียน 47712
วัน, เดือน, ปี 22 ส.ค. 2546

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2546

ISBN 974-324-351-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AN EFFICIENT RESOURCE RESERVATION MECHANISM IN PROBING PHASE FOR UNICAST MULTIMEDIA NETWORKS

KRAISAK KESORN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

ISBN 974-324-351-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2003

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	กลไกการจองทรัพยากรที่มีประสิทธิภาพในขั้นตอนการโพรบสำหรับเครือข่ายมัลติมีเดียแบบยูนิแคสต์
นักศึกษา	นายไกรศักดิ์ เกษร
รหัสประจำตัว	43067026
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ. ศ.	2546
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

บทคัดย่อ

ในเครือข่ายอินเทอร์เน็ตแบบเดิมนั้น โพรโทคอลในการหาเส้นทาง (Routing Protocol) จะใช้เมตริกซ์ (Metrics) ในการพิจารณาเพียงอย่างเดียว เช่น จำนวนเราเตอร์บนเส้นทาง (Hop-Count) หรือความหน่วงเวลา (Delay) และจะใช้อัลกอริทึมในการเลือกเส้นทางที่สั้นที่สุด แต่อัลกอริทึมดังกล่าวไม่สามารถตอบสนองความต้องการการประกันการส่งข้อมูล (QoS) ของแอปพลิเคชั่นใหม่ๆ ได้ เนื่องจากเส้นทางที่สั้นที่สุดอาจจะมีทรัพยากรไม่เพียงพอกับความต้องการโพรโทคอลในการหาเส้นทาง QoS จำเป็นต้องพิจารณาหลายๆ เงื่อนไขพร้อมกัน เช่น แบนด์วิดท์ และดีเลย์ งานวิจัยนี้เสนออัลกอริทึมในการหาเส้นทางสำหรับแอปพลิเคชั่นที่ต้องการการประกันการส่งข้อมูล (QoS) แบบยูนิแคสต์ (Unicast) ภายใต้เครือข่ายที่ใช้เทคโนโลยีเดียวกัน (Homogeneous Network) ซึ่งอยู่บนพื้นฐานของอัลกอริทึมการหาเส้นทางแบบกระจาย (Distributed Routing Algorithm) จากผลการทดลองที่ได้แสดงให้เห็นว่าอัลกอริทึมที่นำเสนอในงานวิจัยนี้สามารถที่จะช่วยเพิ่มอัตราการสร้างการเชื่อมต่อได้สำเร็จมากขึ้น เวลาที่ใช้ในการสร้างการเชื่อมต่อลดลง และจำนวนแพ็กเก็ตที่ใช้สร้างการเชื่อมต่อลดลงอย่างชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	An Efficient Resource Reservation Mechanism in Probing Phase for Unicast Multimedia Networks
Student	Mr.Kraisak Kesorn
Student ID.	43067026
Degree	Master of Science
Programmed	Information Technology
Year	2003
Thesis Advisor	Dr.Chanboon Sathitwiriyawong

ABSTRACT

In traditional networks, most routing protocols employ the shortest-path algorithm for path computation. It is usually characterized by a single performance metric such as hop count or delay. However, those protocols do not sustain the QoS requirement for multimedia applications. In QoS-based networks, additional routing performance metrics such as available bandwidth, delay, and loss rate need to be considered. Although several methods were proposed in recent years, these means still have some problems such as low call acceptance rate and high overhead. This research presents a new resource reservation mechanism of distributed routing algorithm for multimedia traffic. The unicast flow and homogeneous networks, QoS awareness, are proposed to the scope of this algorithm. The simulation results show that the new algorithm can improve the call acceptance rate, the average route distance, the average call setup time, and also the number of call setup packet simultaneously.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี บุคคลที่สำคัญอย่างยิ่งที่คอยให้คำปรึกษาและคำแนะนำในเรื่องต่างๆ มาโดยตลอดคือ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยต้องกราบขอบพระคุณเป็นอย่างสูง ในเรื่องของการศึกษาขอขอบพระคุณอาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ประสิทธิประสาทวิชาความรู้และนำมาใช้ในงานวิจัยนี้จนสำเร็จลุล่วงได้ด้วยดี นอกจากนี้ขอขอบพระคุณ มหาวิทยาลัยนเรศวรและมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร (Computer & Communication Education Foundation) ที่ให้ทุนสนับสนุนในการศึกษาเป็นระยะเวลา 2 ปี

ขอขอบคุณ Research Center for Communication and Information Technology (ReCCIT) สำหรับสถานที่ที่ใช้ในการเก็บข้อมูลและทำการทดลองต่างๆ และขอบคุณเพื่อนๆ IS9 ภาคปกติทุกท่าน ที่ให้ความช่วยเหลือในการทำวิจัยนี้ในเรื่องต่าง ๆ และกำลังใจมาโดยตลอด

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

ไกรศักดิ์ เกษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง..... VIII

สารบัญรูป.....IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 จุดมุ่งหมายและวัตถุประสงค์ของงานวิจัย.....2

1.3 แนวคิดและหลักการที่ใช้ในงานวิจัย.....2

1.4 ขอบเขตของงานวิจัย.....3

1.5 ขั้นตอนในการดำเนินงานวิจัย.....3

1.6 ระยะเวลาที่ใช้ในแต่ละขั้นตอน.....4

1.7 ประโยชน์ที่คาดว่าจะได้รับ.....4

1.8 เนื้อหาในวิทยานิพนธ์.....5

บทที่ 2 หลักการทำงานของ QoS.....6

2.1 หลักการทำงานและองค์ประกอบที่เกี่ยวข้อง.....6

2.2 หลักการทำงานของ Integrated Service และ RSVP.....8

2.3 หลักการทำงานของ Differentiated Service (DiffServ).....10

2.3.1สถาปัตยกรรมการให้บริการระหว่างผู้ส่งถึงผู้รับ.....12

2.3.1.1 บริการแบบเบสเอฟฟอร์ด (Best Effort Service).....12

2.3.1.2 บริการแบบแอสชัวร์ (Assure Service).....12

2.3.1.3 บริการแบบพรีเมียม (Premium Service).....13

2.3.2 การให้บริการในโดเมนของผู้ใช้.....14

2.3.3 การให้ทรัพยากรในโดเมนของผู้ให้บริการ.....15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.4 ตัวอย่างการทำงาน....15

2.3.5 ความต้องการของเราเตอร์ 18

2.4 การทำงานของ MPLS....18

2.5 วิศวกรรมทราฟฟิกและการค้นหาเส้นทางบนพื้นฐานของข้อจำกัดต่าง ๆ

(Traffic Engineer and Constraint Based Routing)....20

2.5.1 วิศวกรรมทราฟฟิก (Traffic Engineering)....20

2.5.2 การค้นหาเส้นทางบนพื้นฐานของข้อจำกัดต่างๆ.... 20

2.5.3 การกระจายข้อมูลสถานะเส้นทางเชื่อมโยง (Link State Information)....21

2.5.4 การคำนวณหาเส้นทาง (Route Computation)....21

2.6 ข้อดีและข้อเสียของ Constraint Based Routing.... 22

2.6.1 โครงสร้างและขนาดของตารางเส้นทาง....22

2.6.2 การเลือกระหว่างการใช้ทรัพยากรให้น้อยที่สุดและการกระจายภาระงาน
ในเครือข่าย (Resource Conservation and Load Balancing)....23

2.6.3 ความมีเสถียรภาพ (Stability)....23

2.7 ความสำคัญของการเลือกเส้นทางเพื่อรองรับ QoS.... 24

บทที่ 3 หลักการเพิ่มประสิทธิภาพสำหรับกลไกการจองทรัพยากรในเครือข่าย....25

3.1 ความรู้พื้นฐาน....25

3.2 วิธีการเดิมในการหาเส้นทาง QoS ในเครือข่าย IP....27

3.2.1 โครงสร้างของตารางเส้นทาง (Routing Table Structure)....27

3.2.2 กลไกการส่งต่อแพ็กเก็ต (Packet Forwarding Mechanism)....28

3.2.3 วิเคราะห์วิธีการเดิม....29

3.3 การเพิ่มประสิทธิภาพการหาเส้นทาง QoS....30

3.4 แผนภาพแสดงการเปลี่ยนสถานะของส่วนประกอบในเครือข่าย....35

3.4.1 แผนภาพการเปลี่ยนสถานะของแพ็กเก็ต....35

3.4.2 แผนภาพการเปลี่ยนสถานะของเส้นทางเชื่อมโยง....35

3.4.3 แผนภาพการเปลี่ยนสถานะของเวิร์คสเตชั่น....36

3.4.4 แผนภาพการเปลี่ยนสถานะของเราเตอร์....36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 การประเมินประสิทธิภาพของการค้นหาเส้นทาง.....39

4.1 การกำหนดเป้าหมายและขอบเขตของระบบ.....39

4.2 กำหนดรายละเอียดการให้บริการของระบบและความคาดหวังที่จะเกิดขึ้น.....39

4.3 กำหนดมาตรวัดประสิทธิภาพ.....40

4.4 กำหนดรายละเอียดของพารามิเตอร์ของระบบและภาระงาน.....40

4.5 กำหนดแฟกเตอร์และค่าที่เป็นไปได้.....40

4.6 เทคนิคการประเมินประสิทธิภาพ.....41

บทที่ 5 การออกแบบการทดลอง.....42

5.1 ส่วนประกอบแบบจำลองเครือข่าย (Network Components).....42

5.1.1 โหนด (Node).....42

5.1.2 เครือข่าย (Network).....43

5.1.3 เส้นทางเชื่อมโยง (Link).....44

5.1.4 อัลกอริทึมค้นหาเส้นทาง (Routing Algorithm).....44

5.1.5 แพ็กเก็ต (Packet).....45

5.2 ส่วนประกอบของความหน่วงเวลา (Delay Components).....45

5.2.1 ความหน่วงเวลาจากการตรวจสอบส่วนหัวของแพ็กเก็ต (Nodal Processing Delay).....45

5.2.2 ความหน่วงเวลาที่เกิดจากคิว (Queuing Delay).....45

5.2.3 ความหน่วงเวลาที่เกิดจากการส่งข้อมูล (Transmission Delay).....45

5.2.4 ความหน่วงเวลาที่เกิดการแพร่กระจายข้อมูล (Propagation Delay).....45

5.3 การสร้างแบบจำลอง (Simulation Modeling).....46

5.3.1 ออบเจ็กต์เราเตอร์ (Router Object).....46

5.3.2 ออบเจ็กต์เวิร์คสเตชั่น (Work Station Object).....47

5.3.3 ออบเจ็กต์เส้นทางเชื่อมโยง (Link Object).....47

5.3.4 ออบเจ็กต์แพ็กเก็ต (Packet Object).....47

5.4 ส่วนประกอบของแบบจำลอง (Simulation Components).....48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 6 การทดลองและผลการทดลอง.....51

6.1 การกำหนดภาระงานของระบบ.....51

6.2 การทดลอง.....52

6.3 ผลการทดลอง.....53

6.3.1 ผลกระทบจากภาระงาน.....53

6.3.2 ผลกระทบจากแบนด์วิดท์ที่ผู้ใช้แต่ละคนต้องการ.....55

6.3.3 ผลกระทบจากเวลาที่ใช้ในการสร้างการเชื่อมต่อ.....57

6.3.4 ผลกระทบต่อทรูพุทเฉลี่ย (Average Throughput).....59

6.3.5 ผลกระทบต่อความหน่วงเวลา (Delay).....60

6.3.6 ผลกระทบต่อค่าความแปรปรวนของการใช้เส้นทางเชื่อมโยง..... 60

6.3.7 เปรียบเทียบทรูพุทเฉลี่ยอัลกอริทึมที่มีและไม่มีฟังก์ชั่น SetStatus().....61

6.3.8 เปรียบเทียบข้อดีของการมีฟังก์ชั่น Bracktract().....62

6.4 สรุปผลการทดลอง.....63

บทที่ 7 บทสรุปและข้อเสนอแนะ.....65

เอกสารอ้างอิง.....67

ภาคผนวก ผลงานที่รับการตีพิมพ์.....69

ประวัติผู้เขียน.....77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ความหมายคำศัพท์ที่เกี่ยวข้อง.....7

6.1 พารามิเตอร์ของแบบจำลองเครือข่าย.....51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 การส่งสัญญาณของโพรโทคอล RSVP (RSVP Signaling).....9

2.2 รูปแบบของ ToS byte.....10

2.3 รูปแบบของ DS byte.....10

2.4 ตัวอย่างปัญหาของ Premium Traffic.....14

2.5 กระบวนการส่งข้อมูลแบบ Assured Service ด้วย Static SLA.....15

2.6 กระบวนการส่งข้อมูลแบบ Premium Service ด้วย Dynamic SLA.....16

2.7 กระบวนการส่งข้อมูลแบบ Premium Service ในสถาปัตยกรรมแบบ MPLS.....19

3.1 โครงสร้างตารางเส้นทางของเราเตอร์ v.....28

3.2 การทำงานของอัลกอริทึมการเลือกเส้นทางในเครือข่ายอินเทอร์เน็ตเพื่อทำให้เกิด QoS.....33

3.3 อัลกอริทึมการรับและส่งแพ็กเก็ตของเราเตอร์.....34

3.4 แผนภาพการเปลี่ยนสถานะของแพ็กเก็ต.....35

3.5 แผนภาพการเปลี่ยนสถานะของเส้นทางเชื่อมโยง.....35

3.6 แผนภาพการเปลี่ยนสถานะของเครื่องเวิร์คสเตชั่น.....36

3.7 แสดงแผนภาพการเปลี่ยนสถานะของเราเตอร์.....37

5.1 ตัวอย่างส่วนประกอบต่าง ๆ ในเราเตอร์.....48

5.2 การติดต่อระหว่างส่วนประกอบต่าง ๆ.....49

5.3 การจัดตารางเวลาให้กับแพ็กเก็ตแต่ละประเภทที่เข้ามายังเราเตอร์.....50

6.1 โทโปโลยีมาตราฐานของ NSFNET-T1.....52

6.2 ผลกระทบของภาระงานต่อค่า ACAR.....54

6.3 ผลกระทบของภาระงานต่อค่า ACST.....55

6.4 ผลกระทบของภาระงานต่อค่า ARD.....55

6.5 สัดส่วนของแพ็กเก็ตโอเวอร์เฮดต่อแพ็กเก็ตข้อมูล.....55

6.6 ผลกระทบของแบนด์วิดท์ที่ร้องขอต่อค่า ACAR.....57

6.7 ผลกระทบของแบนด์วิดท์ที่ร้องขอต่อค่า ACST.....57

6.8 ผลกระทบของแบนด์วิดท์ที่ร้องขอต่อค่า ARD.....57

6.9 ผลกระทบของแบนด์วิดท์ที่ร้องขอต่อจำนวนแพ็กเก็ตในการสร้างการเชื่อมต่อ.....58

6.10 ผลกระทบต่อเวลาที่ใช้ในการสร้างการเชื่อมต่อ ต่อค่า ACAR.....59

6.11 ผลกระทบต่อเวลาที่ใช้ในการสร้างการเชื่อมต่อ ต่อค่า ACST.....59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

6.12 ผลกระทบต่อเวลาที่ใช้ในการสร้างการเชื่อมต่อ ต่อค่า ARD.....59

6.13 ผลกระทบต่อเวลาที่ใช้ในการสร้างการเชื่อมต่อต่อจำนวนแพ็กเก็ตในการสร้างการเชื่อมต่อ...60

6.14 ทรูพุทเฉลี่ยเมื่อมีการเปลี่ยนแปลงภาระงาน.....60

6.15 เปรียบเทียบความหน่วงเวลาที่เกิดจากอัลกอริทึมในการหาเส้นทางของสามอัลกอริทึม.....61

6.16 เปรียบเทียบความแปรปรวนของการใช้เส้นทางเชื่อมโยง.....62

6.17 เปรียบเทียบทรูพุทเฉลี่ยอัลกอริทึมที่มีและไม่มีฟังก์ชั่น SetStatus().....62

6.18 เปรียบเทียบความหน่วงเวลาเฉลี่ยของแพ็กเก็ตของอัลกอริทึมที่มีและ
ไม่มีฟังก์ชั่น SetStatus().....63

6.19 เปรียบเทียบค่า ACAR ของอัลกอริทึมที่มีและไม่มีฟังก์ชั่น Bracktract().....64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

แอปพลิเคชั่น (Application) ใหม่ ๆ ที่ส่งผ่านอินเทอร์เน็ตในปัจจุบันต้องการการประกันคุณภาพของการให้บริการในด้านต่าง ๆ เช่น แบนด์วิดท์ (Bandwidth) ความหน่วงเวลาระหว่างสถานีปลายทางหรือโหนด (End-to-End Delay) อัตราการสูญหายของข้อมูล (Loss Rate) เพื่อให้การส่งข้อมูลมีความราบรื่นมากยิ่งขึ้น จึงมีการนำเสนอวิธีการต่าง ๆ เพื่อประกันคุณภาพในการส่งข้อมูลหรือเรียกว่า "Quality of Service (QoS)" ตัวอย่างเช่นโพรโทคอล (Protocol) RSVP[1], Integrated Service Architecture[2], Differentiated Service Architecture[3] เป็นต้น จะเห็นได้ว่าแอปพลิเคชั่นบนอินเทอร์เน็ตในปัจจุบันนี้แบ่ง 2 กลุ่มใหญ่ ๆ [4] คือ แอปพลิเคชั่นที่ต้องการแบนด์วิดท์ที่คงที่แน่นอน เรียกว่า "Bandwidth-Guaranteed Traffic" และแอปพลิเคชั่นแบบเดิมคือ "Best-Effort Traffic" ซึ่งเครือข่ายอินเทอร์เน็ตในอนาคตนั้นจะต้องสามารถส่งข้อมูลทั้ง 2 ประเภทได้อย่างมีประสิทธิภาพตามความต้องการของแอปพลิเคชั่นต่าง ๆ ที่จะเกิดขึ้นในอนาคตได้

ในการส่งข้อมูลประเภท Bandwidth-Guaranteed Traffic จะต้องมีกระบวนการเพื่อจองทรัพยากรของเครือข่ายซึ่งประกอบด้วย 2 ขั้นตอน [5] คือ 1). หาเส้นทางที่เหมาะสม 2). ทำการจองทรัพยากรบนเส้นทางเหล่านั้น ขั้นตอนแรกนั้นจะสามารถทำได้เมื่อพบเส้นทางที่มีทรัพยากรเพียงพอ กับความต้องการของแอปพลิเคชั่นนั้น ๆ ดังนั้นกลไกในการหาเส้นทางจึงเป็นกลไกที่สำคัญอันหนึ่งที่จะทำให้เกิด QoS ขึ้นมาได้

ในเครือข่ายอินเทอร์เน็ตแบบเดิมนั้น โพรโทคอลในการหาเส้นทาง (Routing Protocol) จะใช้เมตริกซ์ (Metrics) ในการพิจารณาเพียงอย่างเดียว เช่น จำนวนเราเตอร์(Hop-Count) หรือความหน่วงเวลา (Delay) และจะใช้อัลกอริทึมในการเลือกเส้นทางที่สั้นที่สุด แต่อัลกอริทึมดังกล่าวไม่สามารถตอบสนองความต้องการ QoS ของแอปพลิเคชั่นใหม่ ๆ ได้ โพรโทคอลในการหาเส้นทางจำเป็นต้องพิจารณาหลาย ๆ เงื่อนไขพร้อมกัน เช่น แบนด์วิดท์ ความหน่วงเวลา และอัตราการสูญหายของข้อมูล เป็นต้น ปัญหาในการเลือกเส้นทางที่ทำให้เกิด QoS (QoS Routing) คือการเลือกเส้นทางที่ตรงกับเมตริกซ์ต่าง ๆ เหล่านั้นให้ได้ครบถ้วน แต่ในบางครั้งไม่สามารถหาเส้นทางที่สอดคล้องกับเมตริกซ์ทั้งหมดในเส้นทางใดเส้นทางหนึ่งได้ เช่น เส้นทางที่มีแบนด์วิดท์มากที่สุดอาจจะไม่ใช่เส้นทางที่มีความหน่วงเวลาน้อยที่สุดเสมอไป เช่น ดาวเทียม ดังนั้นจึงจำเป็นต้องเลือกว่าจะยึดเมตริกซ์ใดเป็นอันดับแรก อันดับที่สอง และไล่ระดับความสำคัญลงไปเรื่อย ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตของงานวิจัยนี้คือการนำเสนอกลไกการจองทรัพยากรสำหรับอัลกอริทึมในการเลือกเส้นทางในชั้นเครือข่าย (Network Layer) [6] ในการเลือกเส้นทางในการส่งข้อมูลประเภท Bandwidth-Guaranteed Traffic โดยเมตริกซ์ที่พิจารณาคือ แบนด์วิดท์และความหน่วงเวลาตามลำดับ และมีการจองทรัพยากรของเครือข่ายเพื่อให้ได้ QoS ตามที่แอปพลิเคชั่นต้องการ โดยอัลกอริทึมที่นำเสนอนี้จะมุ่งเน้นไปที่ อัตราการความสำเร็จในการร้องขอเพื่อจองทรัพยากร (Call Acceptance Rate) ที่สูงกว่าอัลกอริทึมแบบเดิม

1.2 จุดมุ่งหมายและวัตถุประสงค์ของงานวิจัย

จุดประสงค์ของวิทยานิพนธ์ คือ เพื่อนำเสนอกลไกการจองทรัพยากรสำหรับอัลกอริทึมสำหรับการหาเส้นทาง QoS (QoS Routing) ซึ่งอยู่บนพื้นฐานของการหาเส้นทางแบบกระจาย (Distributed Routing Algorithm) โดยมุ่งเน้นในการเพิ่มอัตราการสร้างการเชื่อมต่อได้สำเร็จมากขึ้น (Call Acceptance Rate) ลดระยะทางของเส้นทางที่ใช้ส่งข้อมูล (Route Distance) ลดเวลาที่ใช้ในการสร้างการเชื่อมต่อ (Call Setup Time) และลดจำนวนแพ็กเก็ตโอเวอร์เฮด (Overhead Packet or Call Setup Packet) ที่ใช้ในการสร้างการเชื่อมต่อ

วัตถุประสงค์ของวิทยานิพนธ์มีดังนี้

- เพื่อศึกษาการค้นหาเส้นทางของอัลกอริทึมเดิมที่นำเสนอก่อนหน้าและปรับปรุงประสิทธิภาพการทำงานให้ดีขึ้น
- เพื่อพัฒนาแบบจำลองของการค้นหาเส้นทางในเครือข่าย
- เพื่อศึกษาพฤติกรรมการค้นหาเส้นทางในเครือข่ายโดยใช้แบบจำลองที่สร้างขึ้น
- เพื่อศึกษามาตรวัดสำหรับการนำมาใช้เปรียบเทียบประสิทธิภาพการค้นหาเส้นทางในเครือข่าย

1.3 แนวคิดและหลักการที่ใช้ในงานวิจัย

เนื่องจากเครือข่ายอินเทอร์เน็ตในปัจจุบันมีแนวโน้มที่จะรองรับแอปพลิเคชั่นประเภทมัลติมีเดีย เช่น เสียง ภาพ การประชุมทางไกล การศึกษาทางไกล มากขึ้น ดังนั้นจึงต้องคำนึงถึงการประกันคุณภาพในการส่งข้อมูลเหล่านี้ เพื่อให้การส่งข้อมูลประเภทมัลติมีเดียบนเครือข่าย IP มีประสิทธภาพมากที่สุด การเลือกเส้นทางสำหรับข้อมูลประเภทมัลติมีเดียกับข้อมูลแบบเดิมจึงอยู่บนพื้นฐานที่แตกต่างกัน งานวิจัยนี้เสนออัลกอริทึมที่ใช้หลักการของการหาเส้นทางแบบกระจาย (Distributed Routing) ในการหาเส้นทางของข้อมูลโดยเส้นทางดังกล่าวจะต้องมีทรัพยากรตัวอย่างเช่น แบนด์วิดท์ ที่เพียงพอในการส่งข้อมูลมัลติมีเดียเหล่านี้ และมุ่งเน้นให้โอกาสที่จะถูกปฏิเสธการจองทรัพยากรของเครือข่ายมีน้อยลงกว่าวิธีการที่มีผู้นำเสนอก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของงานวิจัย

จากการศึกษาหลักการทำงานของ QoS นั้นจะแบ่งการทำงานออกเป็นส่วน ๆ ตาม OSI layer ดังนี้ [6]

Layer	QoS Technique
Transport Layer	Integrated Service, RSVP, Differentiated Service
Network Layer	QoS Routing, QoS Queuing
Data Link Layer	MPLS [1]

ซึ่งขอบเขตในงานวิจัยนี้จะอยู่ในชั้นเครือข่าย (Network Layer) ส่วนของ QoS Routing คือการเลือกเส้นทางในการส่งข้อมูลเพื่อทำให้เกิด QoS และศึกษากลไกลการทำงานและการจองทรัพยากรให้ตรงกับความต้องการของแอปพลิเคชั่นต่าง ๆ ได้อย่างมีประสิทธิภาพ และประสิทธิภาพของอัลกอริทึมที่นำเสนอจะวัดจาก Call Acceptance Rate เป็นหลัก นอกจากนั้นในการศึกษางานวิจัยนี้จะอยู่บนสมมติฐานที่ว่าทุกๆ เราเตอร์บนเครือข่ายรองรับการทำงานของ QoS เหมือนกัน หรือมีลักษณะเครือข่ายเป็นแบบ Homogeneous Networks

1.5 ขั้นตอนในการดำเนินงานวิจัย

ขั้นตอนในการดำเนินงานประกอบด้วย ขั้นตอนหลัก ๆ ดังนี้

1. กำหนดหัวข้อ เป้าหมาย จุดประสงค์และขอบเขตการทำวิทยานิพนธ์
2. ศึกษาทฤษฎีและหลักการพื้นฐานที่เกี่ยวข้องกับ QoS
3. ศึกษาเครื่องมือและวิธีการที่เหมาะสมที่จะใช้ในงานวิจัย
4. ออกแบบการทดลอง
5. ทำการทดลองและวัดผล
6. ปรับปรุงแก้ไข
7. จัดทำเอกสารประกอบวิทยานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ระยะเวลาที่ใช้ในแต่ละขั้นตอน

ขั้นตอนการทำงาน	ระยะเวลาที่ใช้ (เดือนที่)											
	1	2	3	4	5	6	7	8	9	10	11	12
1.ศึกษาทฤษฎีและหลักการพื้นฐานที่เกี่ยวข้องกับ QoS												
2. กำหนดหัวข้อ เป้าหมาย จุดประสงค์และขอบเขตการทำวิทยานิพนธ์												
3. ศึกษาเครื่องมือและวิธีการที่เหมาะสมที่จะใช้ในงานวิจัย												
4. ออกแบบการทดลอง												
5. ทำการทดลองและวัดผล												
6. ปรับปรุงแก้ไข												
7. จัดทำเอกสารประกอบวิทยานิพนธ์												

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการทำงานของ QoS และเข้าใจอัลกอริทึมการเลือกเส้นทางเพื่อทำให้เกิด QoS ที่มีใช้อยู่ในปัจจุบันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. สามารถปรับปรุงอัลกอริทึมที่มีใช้อยู่ในปัจจุบันให้มีการทำงานที่มีประสิทธิภาพมากขึ้นโดยเน้นในส่วนของอัตราความสำเร็จในการร้องขอเพื่อจองทรัพยากร (Call Acceptance Rate)
3. ลดระยะทางของเส้นทางที่ใช้ส่งข้อมูล (Route Distance) ลดเวลาที่ใช้ในการสร้างการเชื่อมต่อ (Call Setup Time) และลดจำนวนโอเวอร์เฮดแพ็กเก็ต (Overhead Packet) ที่ใช้ในการสร้างการเชื่อมต่อ
4. เครือข่ายถูกใช้อย่างมีประสิทธิภาพ

1.8 เนื้อหาในวิทยานิพนธ์

เนื้อหาในวิทยานิพนธ์นี้สามารถแบ่งออกเป็น 4 ส่วนหลักๆ คือ ส่วนแรกกล่าวถึงทฤษฎีที่เกี่ยวข้อง ส่วนที่สองนำเสนอวิธีการใหม่ของงานวิจัยนี้ ส่วนที่สามออกแบบการทดลอง ส่วนที่สี่เป็นส่วนของการประเมินประสิทธิภาพของวิธีการที่นำเสนอ เปรียบเทียบวิธีการที่นำเสนอกับวิธีการเดิม และส่วนสุดท้ายเป็นส่วนของการสรุปผลและข้อเสนอแนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการทำงานของ QoS

ในบทนี้จะกล่าถึงหลักการทำงาน QoS ในภาพกว้าง ๆ ถึงโมเดล (Model) ต่าง ๆ ที่มีผู้นำเสนอขึ้นมา จากนั้นจะกล่าวถึงลักษณะการทำงานของ QoS Routing ที่มีอยู่ในปัจจุบัน ว่ามีวิธีการทำงานอย่างไร

2.1 หลักการทำงานและองค์ประกอบที่เกี่ยวข้อง

ปัจจุบันนี้อินเทอร์เน็ตมีการทำงานแบบ Best-Effort Service ซึ่งความต้องการทางด้านธุรกรรมบนอินเทอร์เน็ตมีความต้องการคุณภาพการบริการเพิ่มขึ้นอย่างรวดเร็ว บริษัทต่าง ๆ ยินดีจ่ายเงินเพื่อทำให้บริการของบริษัทตนมีความน่าเชื่อถือ และผู้ใช้รู้สึกว่ามีการให้บริการที่รวดเร็ว

บริการเหล่านี้อาจจะประกอบด้วยบริการระดับเดียวหรือแบ่งเป็นหลายระดับเช่น Gold Service, Silver Service, Bronze Service ซึ่งระดับของคุณภาพก็ต่างกันไป บริการบางระดับจะให้ความหน่วงเวลา (Delay) และ จิตเตอร์ (Jitter) ที่ต่ำกับแอปพลิเคชั่นบางแอปพลิเคชั่น เช่น Internet Telephony และ Video Conferencing

กลไกที่ทำให้เกิด QoS ยังเป็นการถกเถียงกันอยู่ บางคนให้เหตุผลว่าสายสัญญานแบบไฟเบอร์ (fibers) จะให้แบนด์วิดท์มากกว่าและทำให้เกิด QoS ขึ้นได้ บางคนพูดว่าไม่สำคัญว่าเครือข่ายจะมีแบนด์วิดท์มากน้อยแค่ไหน เพราะแอปพลิเคชั่นใหม่ ๆ ที่ถูกสร้างขึ้นมานั้นจะใช้แบนด์วิดท์เหล่านั้นหมดอยู่ดี ดังนั้นกลไกเพื่อทำให้เกิด QoS จึงยังมีความจำเป็นอยู่

องค์กร IETF (Internet Engineering Task Force) ได้นำเสนอโมเดลและกลไกต่าง ๆ เพื่อให้สอดคล้องความต้องการ QoS ได้แก่ Integrated Service/RSVP, Differentiated Service, MPLS, Traffic Engineer, Constrain Based Routing ขึ้นมา

- Integrated Service มีลักษณะของการจองทรัพยากรไว้ก่อน สำหรับแอปพลิเคชั่นแบบ Real-Time ก่อนที่จะส่งข้อมูลจะต้องมีการสร้างเส้นทาง (Path) ในการส่งข้อมูลก่อน
- RSVP เป็นโพรโทคอลเพื่อส่งสัญญาณในการสร้างเส้นทางเพื่อจะใช้ในการส่งข้อมูล
- Differentiated Service แพ็กเก็ตข้อมูลจะถูกทำสัญลักษณ์ให้ต่างกันเพื่อให้ระดับบริการที่ต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- MPLS ใช้เทคนิคการส่งต่อแพ็กเก็ตโดยที่แพ็กเก็ตจะถูกทำสัญลักษณ์ในทางเข้าและมีการแบ่งแยกแพ็กเก็ตและให้บริการในแต่ละแพ็กเก็ตจะต่างกัน
- Traffic Engineer จะทำกระบวนการในการจัดเรียงข้อมูลให้ไหลอย่างไรในเครือข่าย
- Constraint Based Routing ใช้ในการหาเส้นทางซึ่งขึ้นกับเงื่อนไข ข้อจำกัด เช่น ความหน่วงเวลา แบนด์วิดท์ ที่ต้องการเป็นต้น

ก่อนจะอธิบายรายละเอียดถึงสถาปัตยกรรมต่าง ๆ จะขออธิบายความหมายของคำศัพท์ต่าง ๆ ที่จะกล่าวถึงเพื่อความเข้าใจที่ดีขึ้นดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 ความหมายคำศัพท์ที่เกี่ยวข้อง

คำศัพท์	ความหมาย
Guaranteed Service	มีการประกันแบนด์วิดท์ และจำกัดขอบเขตของความหน่วงเวลา บริการนี้เหมาะสำหรับแอปพลิเคชั่นที่ต้องการประกันอย่างสูงสุด (Highest assurance)
Control Load Service	เนื่องจาก Guaranteed Service มีราคาแพงและทำให้เครือข่ายถูกใช้เป็นประโยชน์น้อยลง Control Load Service จะเหมาะกับแอปพลิเคชั่นที่ไม่ต้องการการประกันที่สูงมากและมีต้นทุนต่ำ บริการนี้ไม่มีการประกันแบนด์วิดท์และความหน่วงเวลา
Multi-Field (MF) Classification	กระบวนการแบ่งแยกแพ็กเก็ตตามข้อมูลของหลาย ๆ ฟิลด์ เช่น Source Address, Destination Address, ToS byte ฯลฯ
Service Level Agreement (SLA)	ข้อตกลงกันระหว่างผู้ใช้บริการและผู้ให้บริการเพื่อระบุถึงระดับของการบริการที่ผู้ใช้ต้องการ
TSpec	การกำหนดลักษณะของทราฟฟิกที่ผู้ส่งต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 (ต่อ)

Policing	จะทำที่เราเตอร์ขอบของเครือข่าย โดยการเปรียบเทียบทราฟฟิกกับพารามิเตอร์ของ TSpec แพ็กเก็ตที่ไม่ตรงกับ TSpec จะถูกปฏิบัติเสมือนกับแพ็กเก็ตแบบ Best-Effort
Shaping	คือการหน่วงเวลาของแพ็กเก็ตเพื่อทำให้เข้ากับโพรไฟล์ (Profile) ที่กำหนดไว้ โดยการเก็บไว้ในบัฟเฟอร์จนกระทั่งตรงกับพารามิเตอร์ของ TSpec
Marking	การทำเครื่องหมายแพ็กเก็ตที่ไม่สอดคล้องกับ TSpec แพ็กเก็ตเหล่านั้นจะถูกทิ้งเป็นอันดับแรกหากเกิดความคับคั่งในเครือข่าย
Behavior Aggregate (BA) Classification	การแบ่งแยกแพ็กเก็ตโดยดูจากข้อมูลใน DS Field ในส่วนหัวของแพ็กเก็ต ซึ่งจะถูกทำสัญลักษณ์โดยเราเตอร์ของผู้ส่ง
Per-Hop-Behavior	ลักษณะพฤติกรรมการส่งต่อแพ็กเก็ต ณ เราเตอร์ต่าง ๆ
DS-Field	ฟิลด์ที่บอกถึงบริการที่ต้องการ แตกต่างกันไปตามแต่ละแพ็กเก็ต ใน IPv4 คือ ToS Field, IPv6 คือ Traffic Class Field

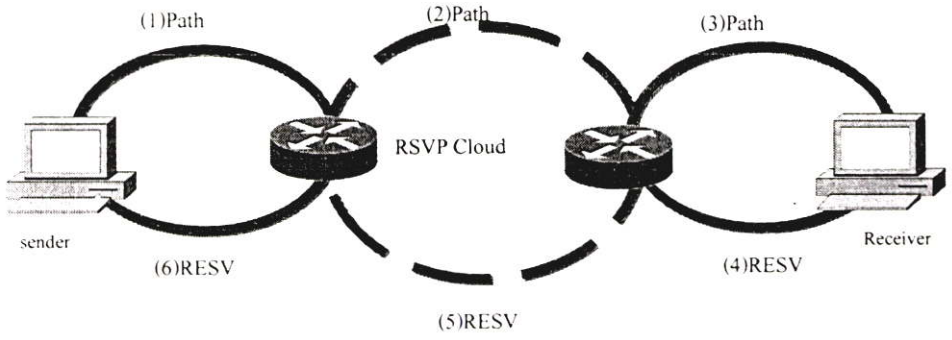
2.2 หลักการทำงานของ Integrated Service และ RSVP

ในโมเดล Integrated Service จะมีการแบ่งออกเป็น 2 ระดับ [6]

5. Guaranteed Service สำหรับแอปพลิเคชั่นที่ต้องการความหน่วงเวลาที่แน่นอน
6. Predictive Service สำหรับแอปพลิเคชั่นที่ต้องการความหน่วงเวลาที่สามารถคาดคะเนได้

วิธีการเหล่านี้เราเตอร์จะต้องมีความสามารถในการจองทรัพยากรได้เพื่อจะให้การประกันคุณภาพกับแพ็กเก็ตข้อมูลที่ต้องการ QoS ดังนั้นที่เราเตอร์จะต้องการมีการกำหนดการไหลของข้อมูล (Flow-Specific State) ไว้ด้วย RSVP เป็นโพรโทคอลในการส่งสัญญาณเพื่อจองทรัพยากร ซึ่งการทำงานแสดงดังรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การส่งสัญญาณของโพรโทคอล RSVP (RSVP Signaling)

ผู้ส่งจะทำการส่ง Path Message ไปยังผู้รับเพื่อระบุถึงลักษณะของทราฟฟิก (Traffic) ที่ต้องการ ทุก ๆ เราเตอร์ที่อยู่ตรงกลาง (Intermediate Router) จะส่งต่อแพ็กเก็ตไปยังเราเตอร์ที่อยู่ถัดไปตาม Routing Protocol เมื่อผู้รับได้รับ Path Message ผู้รับจะตอบกลับด้วย RESV Message เพื่อร้องขอทรัพยากรที่ต้องการ เราเตอร์ที่อยู่ระหว่างทางสามารถที่จะรับหรือปฏิเสธคำร้องขอดังกล่าวได้ ถ้า RSVP Message ถูกปฏิเสธเราเตอร์จะส่ง Error Message กลับไปยังผู้รับ และสัญญาณดังกล่าวจะถูกตัดทิ้ง ถ้า RESV Message ได้รับการตอบรับจะมีการจองแบนด์วิดท์ บัฟเฟอร์ (Buffer) และมีการเก็บข้อมูลบางอย่างไว้

Integrated Service ถูกอิมพลีเมนต์ด้วยเครื่องมือ 4 อย่างคือ Signaling Protocol (RSVP), Admission Control, Classifier และ Packet Scheduler

- แอปพลิเคชั่นที่ต้องการบริการ Guaranteed Service หรือ Control Load Service จะต้องมีการสร้างเส้นทางและจองทรัพยากรก่อนที่จะทำการส่งข้อมูล
- Admission Control จะเป็นตัวตัดสินใจว่าจะอนุญาตให้มีการจองทรัพยากรหรือไม่
- เมื่อเราเตอร์ได้รับแพ็กเก็ตตัว Classifier จะทำการแบ่งแยกแพ็กเก็ตโดยใช้กระบวนการ Multi-Field (MF) Classification และใส่แพ็กเก็ตไปในคิวที่กำหนด
- หลังจากนั้น Packet Scheduler จะจัดตารางเพื่อส่งแพ็กเก็ตให้ได้ QoS ตามต้องการ

ปัญหาของ Integrated Service คือ

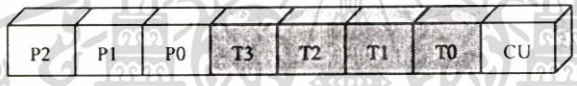
1. จำนวนของ State Information เพิ่มขึ้นตามจำนวนข้อมูลที่ส่งผ่านเครือข่าย ซึ่งเป็นโอเวอร์เฮด ที่เราเตอร์และเปลืองทรัพยากรบนเราเตอร์ด้วย ดังนั้นสถาปัตยกรรม Integrated Service จึงไม่เหมาะกับหลักการของอินเทอร์เน็ต
2. เราเตอร์จะต้องมีประสิทธิภาพสูง เราเตอร์จะต้องอิมพลีเมนต์ RSVP, Admission Control, MF Classification, Packet Scheduler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ถ้าทุก ๆ เส้นทางต้องการ Guaranteed Service จึงเป็นไปได้ว่าจะต้องใช้ Controlled-Load Service มากขึ้นด้วย ดังนั้นจะทำให้เกิดปัญหาคอขวด (Bottle-Neck) ที่เราเตอร์

2.3 หลักการทำงานของ Differentiated Service (DiffServ)

เนื่องจากปัญหาและความยุ่งยากของการใช้ Integrated Service และ RSVP จึงมีผู้นำเสนอหลักการของ Differentiated Service ขึ้น ในส่วนหัวของแพ็กเก็ต IPv4 และ IPv6 นั้น จะมีฟิลด์ ToS (Type-of-Service) ซึ่งบอกประเภทของบริการ ประกอบด้วย 3 บิต ซึ่งจะบอกถึงความต้องการบางอย่างเช่น ต้องการความหน่วงเวลาต่ำ ๆ หรือต้องการแบนด์วิดท์สูง ๆ หรืออัตราการสูญหายของแพ็กเก็ตต่ำ ๆ เป็นต้น อย่างไรก็ตามยังมีข้อจำกัดอยู่ DiffServ ได้กำหนดรูปแบบของฟิลด์ ToS เรียกว่า DS Field (Differentiated Services Field) และการปฏิบัติต่อแพ็กเก็ตข้อมูลที่จะส่งต่อไป โดยการทำสัญลักษณ์ที่ DS Field ตามระดับของบริการที่กำหนดไว้ ดังนั้นวิธีการนี้จึงเป็นวิธีการที่มีลำดับความสำคัญเข้ามาเกี่ยวข้อง รูปที่ 2.2 และ รูปที่ 2.3 แสดง ToS byte และ DS byte ตามลำดับ [14]

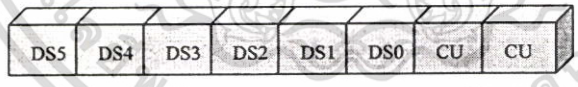


IP precedence : 3 bits (P2-P0)

Type of service (ToS) :4 bits(T3-T0)

Currently unused (CU):1 bit

รูปที่ 2.2 รูปแบบของ ToS byte



Differentiated services code point (DSCP) : 6 bits (DS5-DS0)

Currently unused (CU) : 2 bits

รูปที่ 2.3 รูปแบบของ DS byte

สำหรับผู้ใช้จะได้รับ DiffServ จากผู้ให้บริการอินเทอเน็ต (Internet Service Provider:ISP) ก็ต่อเมื่อจะมีการตกลงกันโดยมี Service Level Agreement (SLA) กับ ISP ก่อน ซึ่ง SLA จะเป็นพื้นฐานที่ใช้กำหนดระดับของการให้บริการ และลักษณะของทราฟฟิกที่จะยอมให้แต่ระดับของการบริการ SLA นั้นจะเป็นแบบ Static หรือ Dynamic ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Static SLA จะมีการเจรจาพูดคุยกันระหว่างผู้ใช้กับผู้ให้บริการ อาจจะเป็นเดือนละครั้ง หรือปีละครั้ง
- Dynamic SLA จะต้องมีการใช้ Signaling Protocol (เช่น RSVP) เพื่อร้องขอบริการที่ต้องการจะมีการร้องขอทุกครั้งที่จะมีการรับ-ส่งข้อมูล ผู้ใช้สามารถทำสัญลักษณ์ที่ DS Field ได้เองเพื่อระบุถึงบริการที่ต้องการหรือให้เราเตอร์ต้นทางเป็นคนทำก็ได้

ณ จุดต้นทางของ ISP แพ็กเก็ตจะถูกแบ่งแยกโดยใช้นโยบาย (Policies) และ Shaping ซึ่งจะใช้เราเตอร์ต้นทางของเครือข่าย ISP ซึ่งอยู่บนพื้นฐานของ SLA กระบวนการนี้ต้องใช้พื้นที่ว่างบนบัฟเฟอร์ ซึ่งจะอยู่บนพื้นฐานของ SLA เช่นกัน เมื่อแพ็กเก็ตเข้าโดเมนหนึ่งไปยังอีกโดเมนหนึ่งอาจจะมีการทำสัญลักษณ์ใหม่ (Remark) โดยดูจาก SLA ระหว่างโดเมนนั้น

การใช้ Classification, Policing, Shaping, Scheduling ทำให้เกิดบริการได้หลายอย่างเช่น

- Premium Service สำหรับแอปพลิเคชั่นที่ต้องการความหน่วงเวลาและจิตเตอร์ต่ำ ๆ
- Assured Service สำหรับแอปพลิเคชั่นที่ต้องการความเชื่อถือมากกว่า Best-effort
- Olympic Service ให้บริการแบบ 3 แบบคือ Gold, Silver, Bronze คุณภาพของการให้บริการจะลดลงตามลำดับ

ข้อดีของ DiffServ คือจะมีการกำหนดเพียง DS Field และ Per-Hop-Behavior (PHB) เท่านั้น ส่วน ISP มีหน้าที่ตัดสินใจว่าจะให้บริการอะไรบ้าง

DiffServ มีความแตกต่างอย่างมากจาก Integrated Service ดังนี้

1. มีข้อจำกัดในส่วนของระดับของการบริการที่ระบุใน DS Field เท่านั้น จำนวนของ State Information จะเป็นสัดส่วนที่พอเหมาะกับจำนวนของระดับการให้บริการ ดังนั้น DiffServ จึงยืดหยุ่นมากกว่า
2. การทำ Packet Classifier, Marking, Policing, Shaping จะทำที่เราเตอร์จุดปลายหรือขอบของเครือข่ายเท่านั้นเราเตอร์ของ ISP เพียงอิมพลีเมนต์ฟังก์ชั่นในการจัดลำดับแพ็กเก็ตเรียกว่า "Behavior Aggregate (BA) Classification" เท่านั้น ดังนั้นจึงง่ายที่จะนำ DiffServ มาใช้ เหตุผลอื่น ๆ ที่เป็นที่พิจารณาสำหรับ ISP คือเครือข่ายของ ISP ปกติจะประกอบด้วย เราเตอร์ปลายทางซึ่งใช้เชื่อมต่อไปยังผู้ใช้ และเราเตอร์ตรงกลางใช้เชื่อมต่อกับเราเตอร์ปลายทาง เราเตอร์ส่วนกลางเหล่านี้จะต้องส่งต่อแพ็กเก็ตอย่างรวดเร็วและไม่ยุ่งยาก ส่วนเราเตอร์ปลายทางไม่มีความจำเป็นจะต้องส่งแพ็กเก็ตอย่างรวดเร็ว เนื่องจากการเชื่อมต่อของผู้ใช้มีความช้ากว่า ดังนั้นจึงจะใช้เวลาในการแยกแพ็กเก็ตนานกว่า จึงเป็นจุดยกเว้นที่ไม่ต้องส่งอย่างรวดเร็ว ยกเว้นเราเตอร์บริเวณที่เรียกว่า Network Access Point (NAPs) จะต้องทำการส่งข้อมูลอย่างรวดเร็ว นอกจากนั้นยังต้องทำการ classification, policing, shaping อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 สถาปัตยกรรมการให้บริการระหว่างผู้ส่งถึงผู้รับ (End-to-End Service Architecture)

แบ่งระดับการบริการออกเป็น 3 ระดับ[6]

- บริการแบบแอสชัวร์ (Assured Service)
- บริการแบบพรีเมียม (Premium Service)
- บริการแบบเบสเอฟฟอร์ด (Best Effort Service)

2.3.1.1 บริการแบบเบสเอฟฟอร์ด (Best Effort Service)

บริการแบบเบสเอฟฟอร์ด [19] เป็นบริการที่ใช้ในสถาปัตยกรรมอินเทอร์เน็ตในปัจจุบัน ซึ่งเป็นบริการที่ไม่มีการรับประกันเวลาในการส่งแพ็กเก็ตข้อมูล ไม่รับประกันลำดับการมาถึงของแพ็กเก็ตข้อมูล และรวมถึงไม่รับประกันการสูญหายของแพ็กเก็ตข้อมูล ดังนั้นเกิดปัญหาการสูญหายของข้อมูล (Packet Loss) และความหน่วงเวลาระหว่างสถานีปลายทาง (End-to-End Delay) เป็นต้น จึงมีเทคนิคการตรวจสอบความผิดพลาด และจะทำการแก้ไขความผิดพลาดดังกล่าว

2.3.1.2 บริการแบบแอสชัวร์ (Assured Service)

บริการระดับนี้ใช้สำหรับผู้ใช้ที่ต้องการความเชื่อถือมากจากผู้ให้บริการ แม้ว่าเครือข่ายมีทราฟฟิกที่คับคั่งก็ตาม ผู้ใช้ต้องมี SLA กับ ISP ของตน ใน SLA จะบอกถึงแบนด์วิดท์ที่ผู้ใช้ต้องการ ปกติแล้ว Assured Service จะมีลักษณะ Static หมายถึงผู้ใช้จะส่งข้อมูลเมื่อใดก็ได้ โดยไม่ต้องบอก ISP บริการแบบ Assured Serviced สามารถอิมพลีเมนต์ได้ดังนี้

1. ฟังก์ชั่น Classification และ Policing จะต้องมีอยู่ที่เราเตอร์ที่จุดเข้าของ ISP ถ้าทราฟฟิกไม่หนาแน่นมากเกินตามที่มีใน SLA จะเรียกว่า "In Profile" แต่ถ้าทราฟฟิกหนาแน่นเกิน SLA จะเรียกว่า "Out Profile"

2. แพ็กเก็ตทั้งหมด (ทั้ง In, Out) จะถูกใส่ใน Assured Queue (AQ) เพื่อให้การส่งแพ็กเก็ตเป็นไปตามลำดับที่ถูกต้อง

3. คิวจะถูกจัดการโดยวิธีที่เรียกว่า RED with In and Out หรือ RIO

RED เป็นวิธีการจัดการคิวซึ่งจะทิ้งแพ็กเก็ตโดยการสุ่ม ซึ่งจะกระตุ้นกลไกควบคุมการไหลของข้อมูล (Flow Control) ของ TCP เพื่อให้ผู้ส่งลดอัตราการส่งข้อมูลลง ณ เวลาต่าง ๆ กัน ดังนั้น RED สามารถป้องกันคิวที่เราเตอร์ไม่ให้เกิดการล้นของข้อมูล (Overflow) ได้

Tail-Drop Behavior คือ การทิ้งแพ็กเก็ตทิ้งเมื่อเกิดการล้นของข้อมูล ซึ่งจะเป็นตัวกระตุ้นการไหลของข้อมูลของ TCP ทั้งลดและเพิ่มอัตราการส่งทำให้มีประสิทธิภาพลดลง วิธีการ RED พิสูจน์แล้วว่านำมาช่วยแก้ข้อเสียตรงนี้ได้

RIO เป็นวิธีการประยุกต์จาก RED เป็นการใช้อัลกอริทึม RED 2 ตัว 1) สำหรับแพ็กเก็ตที่เข้ามา และ 2) สำหรับแพ็กเก็ตที่ออกไป และมีค่าเทสโฮลด์ (Threshold) สำหรับแต่ละคิวเมื่อขนาดของคิวมีขนาดต่ำกว่าค่าเทสโฮลด์ค่าแรกจะไม่มีการทิ้งแพ็กเก็ต และเมื่อคิวมีขนาดอยู่ระหว่างค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทสโชลด์ทั้ง 2 แพ็กเก็ตขาออกเท่านั้นที่จะถูกทิ้งแบบสุ่ม เมื่อคิวมีขนาดเกินค่าเทสโชลด์ที่สองจะเป็นสัญลักษณ์บอกว่าเครือข่ายจะเกิดความคับคั่งเกิดขึ้น ทั้งแพ็กเก็ตที่เข้าและออกจะถูกสุ่มเพื่อตัดแพ็กเก็ตทิ้งแต่ แพ็กเก็ตขาออกจะถูกตัดทิ้งมากกว่า

2.3.1.3 บริการแบบพรีเมียม (Premium Service)

บริการแบบ Premium Service จะให้บริการที่ ความหน่วงเวลาต่ำและค่าจิตเตอร์ที่ต่ำ ผู้ใช้แต่ละคนจะต้องมี SLA กับ ISP ของตน ใน SLA จะระบุถึง Bit-Rate สูงสุดของการไหลของข้อมูล ผู้ใช้จะต้องรับผิดชอบไม่ส่งข้อมูลที่มีอัตราเร็วไม่เกินที่กำหนด ข้อมูลส่วนที่เกินจะถูกตัดทิ้งไป ส่วน ISP จะประกันว่ามีแบนด์วิดท์ให้ตามที่ผู้ใช้ต้องการ เมื่อมีการส่งข้อมูล Premium Service นั้นจะเหมาะกับบริการประเภท Internet Telephony, Video Conferencing หรือพูดได้ว่าเป็นการสร้าง Virtual Lease Line สำหรับ VPN ก็ได้ บริการแบบ Premium Service นั้นมีราคาแพงกว่า Assured Service และ ISP ต้องพิจารณาว่าควรจะ Support Static SLA หรือ Dynamic SLA

Dynamic SLA จะอนุญาตให้ผู้ใช้ร้องขอเพื่อขอใช้บริการ Premium Service ได้ตามความต้องการโดยไม่ต้องลงทะเบียนก่อนหน้า ซึ่งวิธีการนี้ต้องใช้ Admission Control ในการตัดสินใจว่าจะให้บริการที่ร้องขอหรือไม่ Premium Service แสดงดังรูปที่ 2.4 มีหลักการทำงานได้ดังนี้ [6]

- ฝั่งผู้ใช้ (Customer)

เราเตอร์ปลายทางจะติดต่อไปยังผู้ส่งโดยทำการ MF classification และ traffic shaping เราเตอร์จะพิจารณา P-bit ใน DS field ถ้ามีการกำหนด P-bit ไว้ แพ็กเก็ตนี้จะอยู่ใน Premium Class ถ้าไม่มีการกำหนด P-bit แพ็กเก็ตนี้จะอยู่ใน Assured Service หรือ Best effort หลังจากทำการ shaping แล้ว แพ็กเก็ตทุกแพ็กเก็ตจะมีการกำหนดให้เป็น P-bit ที่เราเตอร์ทางออกของโดเมนอาจจะมีการ reshape ใหม่เพื่อให้แน่ใจว่าทราฟฟิกจะไม่เกินอัตราสูงสุดที่กำหนดไว้ใน SLA

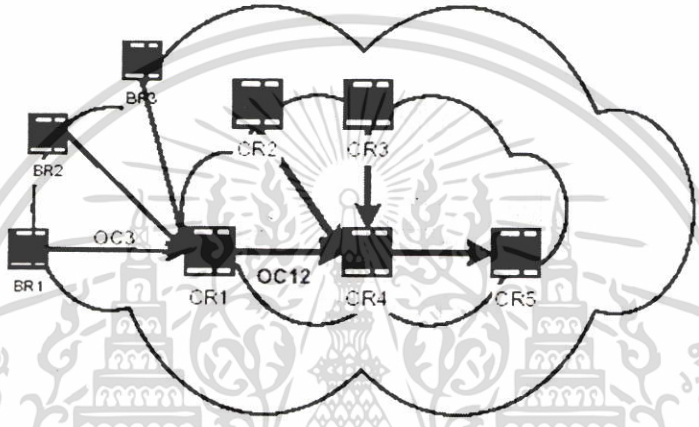
- ฝั่งผู้ให้บริการอินเทอร์เน็ต (ISP)

เราเตอร์ทางเข้าโดเมนของ ISP จะทำการตรวจสอบทราฟฟิก ถ้าเกินจะถูกทิ้ง (Drop) หลังจากนั้นแพ็กเก็ตที่กำหนดให้เป็น P-bit จะเข้าไปอยู่ในคิว Premium Queue (PQ) แพ็กเก็ตใน PQ จะถูกส่งออกไปก่อนแพ็กเก็ตใน Assured Queue (AQ) เสมอ หลังจากนั้น Admission control จะจัดการจำนวนทราฟฟิกแบบ Premium เช่นให้ 10% ของแบนด์วิดท์ทั้งหมด แพ็กเก็ตที่เกินทราฟฟิกที่กำหนดจะถูกทิ้งไป แพ็กเก็ตใน PQ จะถูกส่งออกไปก่อนแพ็กเก็ตในระดับอื่น ๆ ซึ่งทำให้สามารถใช้แบนด์วิดท์ได้ 100% อย่างไรก็ตามมีปัจจัย 3 ประการที่ต้องทำการประกันคือ

1. อัตราการให้บริการ (Service Rate) ใน PQ ต้องมากกว่าอัตราการมาถึงของแพ็กเก็ต (Arrival Rate)
2. แพ็กเก็ตที่มาถึงใหม่จะต้องหา PQ ที่ว่างหรือสั้นที่สุด
3. ความหน่วงเวลาหรือจิตเตอร์จะต้องต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตาม การกระจายของ Premium traffic ที่ไม่แน่นอนอาจก่อปัญหาให้กับ Premium Service ได้ ในเครือข่าย ISP การรวมกันของทราฟฟิกจากเราเตอร์ต้นทางถึงเราเตอร์ส่วนกลาง (Core Router) ไม่สามารถหลีกเลี่ยงได้ เช่น CR1 ในรูป 2.4 แต่กรณีนี้ไม่เกิดปัญหาเนื่องจากเส้นทางขาออก (Output Link) จะมีความเร็วกว่าเส้นทางขาเข้า (Input Link) แต่ถ้าเป็น CR4 จะมีอัตราการมาถึงของแพ็กเก็ตกับอัตราการส่งแพ็กเก็ตออกไป มีค่าใกล้เคียงกันหรืออัตราการมาถึงของแพ็กเก็ตอาจจะมาเร็วกว่า ลักษณะแบบนี้อาจจะเกิดปัญหาขึ้นได้ วิธีแก้คือใช้ Constraint Based Routing มาช่วยแก้ปัญหาความหนาแน่นของข้อมูล และ Admission Control ต้องจำกัดจำนวนแบนด์วิดท์ที่ Premium traffic ร้องขอ หรือใช้ Weight Fair Queuing (WFQ) ระหว่าง PQ และ AQ จะช่วยแก้ปัญหานี้



รูปที่ 2.4 ตัวอย่างปัญหาของ Premium Traffic

2.3.2 การให้บริการในโดเมนของผู้ใช้ (Service Allocation in Customer Domains)

ในโดเมนของผู้ใช้จะต้องมีการตัดสินใจว่าจะมีการใช้บริการที่ระบุใน SLA ร่วมกันอย่างไร กระบวนการนี้เราเรียกว่า "Service Allocation" ซึ่งจะมีทางเลือกอยู่ 2 ทาง คือ 1).แต่ละโฮสต์ (host) จะทำการตัดสินใจเองว่าจะใช้บริการใด 2).มีตัวควบคุมทรัพยากร หรือเรียกว่า "Bandwidth Broker (BB)" เป็นตัวตัดสินใจให้กับทุก ๆ โฮสต์ BB จะเป็นจัดการทรัพยากรในโดเมนและจัดสรรให้กับแต่ละโฮสต์ตามที่ระบุไว้ใน SLA ซึ่งวิธีการที่ 2 นี้จะมีประสิทธิภาพมากกว่า หลักการทำงานคือ เมื่อโฮสต์ต้องการส่งข้อมูลจะไปปรึกษา BB เพื่อจะขอใช้บริการ BB จะเป็นตัวตัดสินใจว่าจะให้บริการระดับไหนและจะตอบกลับไปยังโฮสต์ ถ้าเป็นบริการแบบ Premium Service ตัว BB ใช้ โพรโทคอลเช่น RSVP เพื่อทำการกำหนดระดับ และกฎการทำสัญลักษณ์ของแพ็กเก็ต ที่เราเตอร์ต้นทาง (Leaf Router) ที่ติดต่อโดยตรงกับผู้ส่ง เราเตอร์เหล่านี้จะเป็นตัวทำสัญลักษณ์แพ็กเก็ต

ในกรณีที่ SLA ระหว่างผู้ใช้และ ISP เป็นแบบ Dynamic SLA ตัว BB ของผู้ใช้จะใช้โพรโทคอลในการส่งสัญญาณ (Signal Protocol) ได้แก่ RSVP เพื่อร้องขอทรัพยากรกับ ISP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 การให้ทรัพยากรในโดเมนของผู้ให้บริการ (Resource Allocations in ISP Domains)

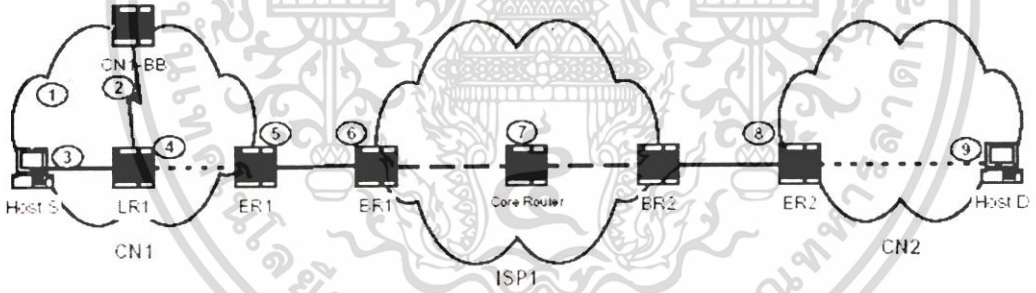
ISP จะต้องตัดสินใจว่าจะปรับแต่งค่าของเราเตอร์ที่เป็นเราเตอร์ทางออก และเราเตอร์ทางเข้าโดเมนของ ISP อย่างไรเพื่อจะจัดการกับทราฟฟิกที่จะเข้ามา กระบวนการนี้เรียกว่า "Resource Allocation"

สำหรับ Static SLA นั้นเราเตอร์ที่ขอบของโดเมน (Boundary Router) สามารถปรับแต่งได้ง่ายเนื่องจาก ทรัพยากรถูกจัดให้กับผู้ใช้แต่ละคนอย่างตายตัว และทรัพยากรที่ไม่มีใครจองไว้สามารถให้ผู้อื่นใช้ร่วมกันได้

สำหรับ Dynamic SLA การจัดสรรทรัพยากรจะมีลักษณะคล้ายกับกระบวนการส่งสัญญาณมาเพื่อขอทรัพยากร BB ในโดเมนของผู้ใช้จะใช้ RSVP ส่งสัญญาณมาเพื่อขอทรัพยากรจาก ISP และ Admission Control จะเป็นตัวตัดสินใจและส่งผลการตัดสินใจนี้ไปยัง BB ในโดเมนของผู้ใช้และเราเตอร์ที่ขอบของโดเมน ISP เพื่อทำการปรับแต่งค่า (Configure) Classification, Policing, Shaping เพื่อให้สอดคล้องกับความต้องการของผู้ใช้ ในส่วนของ BB จะทำการส่งผลการตัดสินใจของ ISP นี้ไปยังเราเตอร์ของตนเพื่อทำการปรับแต่งค่าในเราเตอร์ในโดเมนของผู้ใช้ด้วยเช่นกัน

2.3.4 ตัวอย่างการทำงาน

ตัวอย่างที่ 1 : ส่งข้อมูลแบบ Assured Service ด้วย Static SLA [6]



รูปที่ 2.5 กระบวนการส่งข้อมูลแบบ Assured Service ด้วย Static SLA

ก่อนที่จะอธิบายถึงขั้นตอนการทำงานในรูปที่ 2.5จะขออธิบายถึงอักษรย่อในรูป ดังนี้

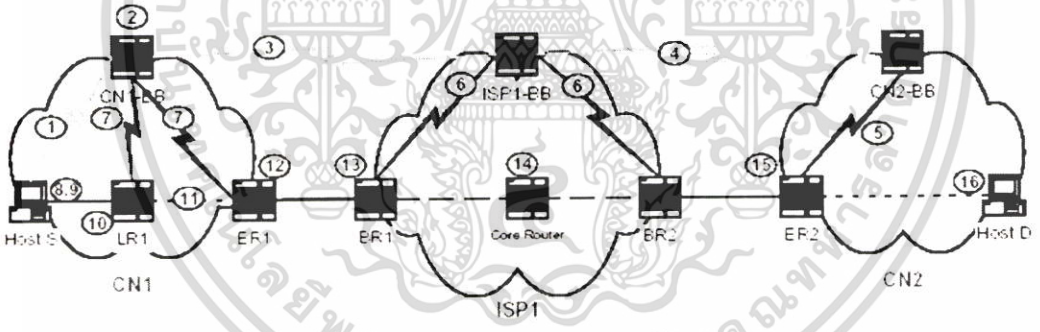
1. LR = Leaf Router
2. ER = Edge Router
3. BR = Boundary Router
4. BB = Bandwidth Broker
5. CN = Corporate Network
6. ISP = Internet Service Provider

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมติว่าโฮสต์ S ใน CN1 ต้องการใช้ Assured Service เพื่อส่งข้อมูลให้โฮสต์ D ใน CN2 และ CN1 มี Static SLA กับ ISP1 ขั้นตอนการทำงานในรูปที่ 2.3 มีดังต่อไปนี้

1) โฮสต์ S ส่ง RSVP ไปยัง CN1-BB เพื่อร้องขอ Assured Service
2) ถ้า CN1-BB ให้สิทธิแก่โฮสต์ S แล้ว CN1-BB จะส่งสัญญาณเพื่อทำการปรับแต่งค่าที่ LR1 โดยกำหนด A-bit ของ DS-Field ของแพ็กเก็ตและ CN1-BB จะตอบกลับไปยังโฮสต์ S ถ้าไม่ให้สิทธิจะส่ง Error Message กลับไปยังโฮสต์ S
3) เมื่อโฮสต์ S ได้สิทธิจะส่งแพ็กเก็ตไปยัง LR1
4) LR1 จะทำสัญลักษณ์แพ็กเก็ตโดยการกำหนด A-Bit ของแพ็กเก็ต
5) เราเตอร์ทุกเราเตอร์ที่อยู่ระหว่าง LR1 ถึง ER1 จะทำ Behavior Aggregate (BA) Classification แพ็กเก็ตที่กำหนดเป็น A-bit จะถูกพิจารณาเป็น In ส่วนแพ็กเก็ตอื่นๆ จะพิจารณาเป็น Out และนำแพ็กเก็ตทั้งหมดใส่ใน AQ และมีการใช้ RIO
6) BR1 มีการกำหนดนโยบายว่าถ้าแพ็กเก็ตที่เป็น In เกิดอัตราที่กำหนดไว้ แพ็กเก็ตนั้น ๆ จะถูกกำหนด A-bit ให้เป็นแพ็กเก็ตธรรมดาและถูกใส่ใน AQ อีกครั้ง และใช้ RIO
7) เราเตอร์หว่าง BR1 ถึง BR2 ทำการ BA Classification และใช้ RIO กับ AQ ของตน
8) ER2 ทำเหมือน BR1
9) แพ็กเก็ตจะถูกส่งไปยังโฮสต์ D

ตัวอย่างที่ 2 : ส่งข้อมูลแบบ Premium Service ด้วย Dynamic SLA [6]



รูปที่ 2.6 กระบวนการส่งข้อมูลแบบ Premium Service ด้วย Dynamic SLA

จากรูป 2.6 สมมติว่าโฮสต์ S ใน CN1 ต้องการใช้ Premium Service ส่งข้อมูลให้โฮสต์ D ใน CN2 และ CN1 มี SLA แบบ Dynamic กับ ISP1

ช่วงที่ 1 ส่งสัญญาณ (Signaling)

1) โฮสต์ S ส่ง RSVP Path Message ไปยัง CN1-BB
2) CN1-BB จะทำการตัดสินใจ
    - ถ้าไม่ให้สิทธิจะส่ง Error Message กลับไปยังโฮสต์ S และจบการส่งสัญญาณ
3) คำขอถูกรับโดย CN1-BB และ CN1-BB จะส่ง Path Message ไปยัง ISP1-BB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

4) ISP1-BB ใช้ Admission Control ช่วยตัดสินใจ
   - ถ้าไม่ให้สิทธิจะส่ง Error Message กลับไปยัง CN1-BB และแจ้งให้กับโฮสต์ S ทราบ
   - ถ้าให้สิทธิ ISP1-BB ส่ง Path Message ไปให้ CN2-BB
5) CN2-BB ใช้ Admission Control ในการตัดสินใจ
   - ถ้าไม่ให้สิทธิ ดังนั้น Error Message ถูกส่งกลับไปยัง ISP1-BB และแจ้งให้กับโฮสต์ S ทราบ
   - ถ้าให้สิทธิ ดังนั้น CN2-BB จะใช้ RSVP กำหนดกฎสำหรับ Classification, Policing ที่เราเตอร์ ER2 ต่อจากนั้น CN2-BB จะให้ RSVP message ไปยัง ISP1-BB
6) เมื่อ ISP1-BB ได้รับ RESV Message จะทำการกำหนดกฎในการ Classification, Policing บนเราเตอร์ BR1 และ BR2 และทำการส่ง RESV Message ไปยัง CN1-BB
7) เมื่อ CN1-BB ได้รับ RESV Message จะทำการส่งกฎ Classification, Shaping สำหรับเราเตอร์ LR1, ER1 และส่ง RESV Message ไปยังโฮสต์ S
8) เมื่อโฮสต์ S ได้รับ RESV Message จึงเริ่มส่งข้อมูลได้

ข้อสังเกตุ การส่งสัญญาณนี้ต่างจาก Signal Process ใน Integrated Service
- ผู้ส่งจะทำการร้องขอทรัพยากร ไม่ใช่ผู้รับ
- คำขอถูกปฏิเสธได้ เมื่อ BB ได้รับ Path Message จากผู้ส่ง ใน Integrated Service/RSVP คำขอจะถูกปฏิเสธเมื่อได้รับ RESV Message จากผู้รับ
- BB สามารถรวมคำขอไว้ด้วยกันและส่งไปยัง BB ต่อไปในคราวเดียว ซึ่งช่วยลดทราฟฟิก
- แต่ละโดเมนเป็นเหมือนโหนดเดียวซึ่งแทนด้วย BB และ State Information ถูกเก็บไว้ในเราเตอร์ที่ขอบของเครือข่าย ทำให้ปรับปรุงได้ง่าย
- ถ้ามี ISP หลาย ๆ ISP ให้ทำ 4-6 ซ้ำ ในแต่ละ ISP
- ถ้า SLA ระหว่าง CN1 กับ ISP1 เป็นแบบ Static ให้ข้ามขั้น 3-6 ไป

ช่วงที่ 2 ส่งข้อมูล (Data Transmission)

9) โฮสต์ S ส่งแพ็กเก็ตไปยัง LR1
10) LR1 ทำการ MF Classification ถ้าทราฟฟิกไม่เป็นรูปแบบเดียวกัน LR1 จะทำการ Shape มัน LR1 จะกำหนด P-bit ในทุกแพ็กเก็ต และแพ็กเก็ตจะส่งไปอยู่ใน PQ
11) เราเตอร์ระหว่าง LR1 และ ER1 จะทำ BA Classification และใส่แพ็กเก็ตใน PQ และส่งแพ็กเก็ตออกไป

47712
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12) ER1 ทำการ BA Classification และ Reshapes

13) BR1 ทำการแยก Premium Traffic ส่วนแพ็กเก็ตที่เกิน Bit-Rate จะถูกตัดทิ้ง

14) เราเตอร์ที่อยู่ระหว่าง BR1 และ BR2 ทำการ BA Classification ส่วน BR2 ทำการ Reshape Traffic

15) ER2 ทำการแยกแพ็กเก็ตและทิ้งแพ็กเก็ตที่เกิน Bit-Rate ที่กำหนดใน SLA

16) แพ็กเก็ตถูกส่งให้โฮสต์ D

2.3.5 ความต้องการของเราเตอร์ (Router Requirement)

เราเตอร์จะรองรับ Premium Service และ Assured Service ต้องมีลักษณะดังนี้

1. เราเตอร์ในโดเมนของผู้ใช้จะต้องมี MF Classification, Marking, Shaping
2. เราเตอร์ทางเข้าของ ISP ต้องมีฟังก์ชั่น Policing และ Remarking
3. เราเตอร์ทางออกของ ISP ต้องมีฟังก์ชั่น Reshape
4. ทุกเราเตอร์ต้องมี BA Classification และมี 2 คิว
5. ถ้าเป็น Dynamic SLA โดเมนของผู้ใช้ต้องใช้ Bandwidth Broker (BB) เพื่อขอใช้บริการ กลไก Signaling และ Admission Control ต้องมีทั้งในโดเมนของผู้ใช้และโดเมนของ ISP

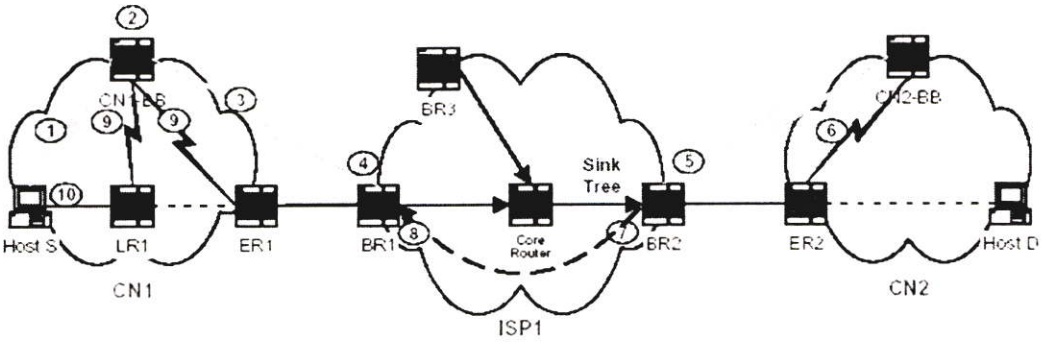
2.4 การทำงานของ MPLS

ในงานวิจัยนี้จะขอพูดเกี่ยวกับ MPLS พอสังเขปเท่านั้น MPLS เป็นวิธีการส่งแพ็กเก็ต ใน OSI Layer จะอยู่ระหว่าง Link Layer และ Network Layer แพ็กเก็ตของ MPLS จะส่วนหัวของแพ็กเก็ตประกอบด้วยลาเบล (Label) 20 บิต, Class of Service(CoS) 3 บิต, Stack Indicator 1 บิต, TTL 8 บิต เราเตอร์ที่ทำงานด้วยวิธี MPLS จะเรียกว่า "Label Switched Router (LSR)" เพื่อทำการตรวจสอบลาเบลและส่งแพ็กเก็ตนั้นต่อไป โพรโทคอลในเครือข่ายอาจจะเป็น IP หรือโพรโทคอลอื่น ๆ ก็ได้ จึงเป็นสาเหตุให้เรียกว่า "Multi-Protocol Label Switching"

MPLS ต้องการโพรโทคอลในการกระจายข้อมูลเพื่อทำการสร้างเส้นทางเรียกว่า "Label Switched Paths (LSPs)" โพรโทคอลนั้นเรียกว่า "Label Distribution Protocol (LDP)" อาจจะเป็นการสร้างโพรโทคอลขึ้นมาใหม่หรือว่าขยายการทำงานของ RSVP ก็ได้ LSP นั้นจะมีลักษณะคล้ายกับ Virtual Circuit ในเครือข่าย ATM โดยมีการใช้โพรโทคอลเพื่อดูความหมายของลาเบล เช่นจะมีการจัดการกับแพ็กเก็ตเหล่านั้นอย่างไร MPLS นั้นสามารถรวมทราฟฟิกระดับเดียวกันไว้และส่งไปยัง LSP เดียวกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 3 : ขั้นตอนการทำงานของ MPLS [6]



รูปที่ 2.7 กระบวนการส่งข้อมูลแบบ Premium Service ในสถาปัตยกรรมแบบ MPLS

ขั้นตอนการทำงานของสถาปัตยกรรม MPLS มีดังต่อไปนี้

1) โฮสต์ S ส่ง RSVP PATH Message ไปยัง Bandwidth Broker ของตนนั่นคือ CN1-BB
2) CN1-BB จะทำการตัดสินใจ
3) ถ้าปฏิเสธการร้องขอ จะส่ง Error Message กลับไปยังโฮสต์ S และหยุดกระบวนการส่งสัญญาณนี้ (Signal process)
    - เมื่อ CN1-BB ได้รับคำร้องขอ CN1-BB จะส่ง PATH Message ไปยัง BR1
    - BR1 จะทำการตรวจสอบว่ามีทรัพยากรพอหรือไม่ที่จะส่งทราฟฟิกไปยัง BR2
4) ถ้ามีไม่พอจะปฏิเสธคำร้องขอนั้น และส่ง Error Message กลับไปยัง CN1-BB และโฮสต์ S
5) ถ้ามีพอ ISP1-BB จะส่ง PATH Message ผ่าน LSP ไปยัง BR2
    - BR2 ส่ง PATH Message ไปยัง CN2-BB
    - CN2-BB จะทำการตัดสินใจว่าจะสามารถรองรับทราฟฟิกได้หรือไม่
6) ถ้าไม่ได้ จะปฏิเสธคำขอนั้น และส่ง Error Message กลับไปยัง BR2 และแจ้งให้โฮสต์ S ทราบ
7) ถ้าได้ จะรับคำขอนั้นมา จากนั้น CN2-BB จะใช้ LDAP หรือ RSVP เพื่อกำหนดกฎในการ Classification และ Policing สำหรับเราเตอร์ ER2 และ CN2-BB จะส่ง RSVP PATH Message ไปยัง BR2
    - BR2 จะทำการปรับแต่งกฎสำหรับทราฟฟิกดังกล่าว และส่ง RESV Message ผ่าน LSP ไปยัง BR1
    - BR1 จะทำการปรับแต่งกฎสำหรับ Classification และ Policing เพื่อรองรับทราฟฟิก และส่ง RESV Message ไปยัง CN1-BB
    - เมื่อ CN1-BB ได้รับ RESV Message มันจะปรับแต่งกฎในการ Classification และ Shaping ที่เราเตอร์ LR1 ดังนั้นหากทราฟฟิกไม่เป็นไปตามที่กำหนดไว้ LR1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถที่จะทำการ Reshape ทราฟฟิกได้ CN1-BB จะส่งกฎในการ Reshape ไปยัง ER1 ด้วยเช่นกัน หลังจากนั้น CN-BB จะส่ง RESV Message ไปยังโฮสต์ S

- โฮสต์ S เริ่มส่งข้อมูล

8) ถ้ามี ISP หลาย ISP ระหว่าง CN1 และ CN2 จะทำขั้นตอน 4-5 และ 7-8 ซ้ำในแต่ละ ISP

2.5 วิศวกรรมทราฟฟิกและการค้นหาเส้นทางบนพื้นฐานของข้อจำกัดต่างๆ (Traffic Engineer and Constraint Based Routing)

วิธี Integrated Service, Differentiated Service จะทำให้ประสิทธิภาพของเครือข่ายลดลง เนื่องจากมีแพ็กเก็ตส่งระหว่างการทำงานเป็นจำนวนมาก ทำให้ทราฟฟิกสูงในเครือข่าย ดังนั้นเพื่อเป็นการหลีกเลี่ยงปัญหาดังกล่าวจึงเกิด Traffic Engineer ขึ้น

2.5.1 วิศวกรรมทราฟฟิก (Traffic Engineering)

ถ้าเกิดความหนาแน่นในเครือข่ายเป็นสาเหตุทำให้มีทรัพยากรน้อย เนื่องจากการกระจายของแพ็กเก็ตที่ไม่ปกติในเครือข่าย กรณีแรกเราเตอร์ทุกเราเตอร์และสายเชื่อมโยงจะเกิดโอเวอร์โหลด (Overload) ทางแก้ไขคือการเพิ่มทรัพยากรโดยการอัพเกรดปัจจัยพื้นฐานของเครือข่ายเช่น เพิ่มแบนด์วิดท์ บัฟเฟอร์ เป็นต้น กรณีที่ 2 บางส่วนของเครือข่ายเท่านั้นที่เกิดโอเวอร์โหลดในขณะที่ส่วนอื่น ๆ ที่โหลดน้อยสาเหตุที่เกิดการกระจายที่ไม่สม่ำเสมอในเครือข่ายมีสาเหตุมาจาก Dynamic Routing Protocol ที่ใช้เช่น RIP, OSPF, IS-IS เพราะโพรโทคอลเหล่านี้จะเลือกเส้นทางสั้นที่สุดในการส่งแพ็กเก็ตเสมอ ทำให้เส้นทางที่สั้นที่สุดนั้นเกิดความหนาแน่นเกินขึ้น ขณะที่เส้นทางที่ยาวกว่าอื่น ๆ นั้นว่างอยู่ Equal-Cost Multi-Path (ECMP), IS-IS ได้ถูกนำเสนอมาช่วยแก้ปัญหานี้ แต่ถ้าในเครือข่ายมีเส้นทางที่สั้นที่สุดเพียงเส้นทางเดียว ECMP ก็ไม่ช่วยอะไร Traffic Engineering เป็นกระบวนการจะหลีกเลี่ยงความหนาแน่นในเครือข่ายและมีการกระจายอย่างสม่ำเสมอ Constraint Based Routing เป็นเครื่องมือที่สำคัญใน Traffic Engineering

2.5.2 การค้นหาเส้นทางบนพื้นฐานของข้อจำกัดต่างๆ (Constraint Based Routing)

Constraint Based Routing (CBR) ถูกใช้คำนวณหาเส้นทางจากเงื่อนไขหลาย ๆ อย่าง CBR เกี่ยวกับ QoS Routing การจะให้ QoS นั้น QoS Routing ต้องให้เส้นทางที่ดีที่จะทำให้เกิด QoS ได้ CBR เป็นส่วนขยายจาก QoS Routing โดยการพิจารณาเงื่อนไขอื่น ๆ ของเครือข่ายเช่น Policy

เป้าหมายของ CBR คือ

1. หาเส้นทางที่จะทำให้เกิด QoS ได้
2. เครือข่ายถูกใช้อย่างมีประสิทธิภาพมากขึ้น (Utilization)

CBR ไม่สนใจเฉพาะโทโพโลยี (Topology) ของเครือข่าย แต่ยังคำนึงถึงความต้องการของส่งข้อมูลด้วย ทรัพยากรที่เหลืออยู่ของสายเชื่อม นโยบายอื่น ๆ ที่กำหนดโดยผู้ดูแลเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Admin) อย่างไรก็ตาม CBR อาจจะหาเส้นทางที่ยาวกว่าและมีโหลดที่น้อยกว่าเส้นทางที่สั้นกว่า ดังนั้นจะเกิดการกระจายที่สม่ำเสมอในเครือข่าย

2.5.3 การกระจายข้อมูลสถานะเส้นทางเชื่อมโยง (Link State Information)

เราเตอร์ต้องการข้อมูลเกี่ยวกับโทโพโลยีและทรัพยากรที่มีอยู่ เพื่อคำนวณหาเส้นทาง QoS ในที่นี้คือแบนด์วิดท์ที่มี วิธีการที่กระจายข้อมูลของแบนด์วิดท์ คือใช้วิธีโฆษณาแบนด์วิดท์ เช่น OSPF เนื่องจากแบนด์วิดท์ที่เหลือมีการเปลี่ยนแปลงบ่อย จึงจะต้องเลือกระหว่างความถูกต้องและหลีกเลี่ยงข้อมูล Link State Advertisement ที่มากเกินไปในเครือข่าย วิธีหนึ่งที่จะหลีกเลี่ยงไม่ให้ Link State Advertisement มากเกินไปคือการกระจายข้อมูลเหล่านั้นเมื่อมีการเปลี่ยนโทโพโลยีเท่านั้น หรือการเปลี่ยนแปลงแบนด์วิดท์มากกว่าค่าที่กำหนดไว้ เช่น ถ้าเปลี่ยนแปลงมากกว่า 50% หรือมากกว่า 10 Mbps อีกวิธีหนึ่งคือใช้นาฬิกาจับเวลา เช่นกระจายข้อมูลทุก 30 วินาที เป็นต้น

2.5.4 การคำนวณหาเส้นทาง (Route Computation)

อัลกอริทึมในการคำนวนหาตารางเส้นทาง (Routing Table) ใน CBR ความซับซ้อนของอัลกอริทึมขึ้นอยู่กัลเมตริกซ์ที่เลือกใช้ เมตริกซ์โดยทั่วไปที่ใช้ใน CBR ได้แก่ ค่าใช้จ่าย (Cost) จำนวนเราเตอร์ที่ผ่าน (Hop-Count) แบนด์วิดท์ ความน่าเชื่อถือ (Reliability) ความหน่วงเวลา และจิตเตอร์ อัลกอริทึมในการหาเส้นทางจะเส้นทางที่ดีที่สุดและตรงกับเงื่อนไขของเมตริกซ์เหล่านี้

เมตริกซ์เหล่านี้แบ่งเป็น 3 ประเภท [6] ให้ d(i,j) เป็นเมตริกซ์สำหรับเส้นทางเชื่อมจาก i ไป j (i,j) สำหรับเส้นทาง P ใด ๆ $P = (i, j, k, \ldots, l, m)$ เมตริกซ์ d คือ

- *Additive if* $d(P)=d(i,j)+d(j,k)+\ldots+d(l,m)$
- *Multiplicative if* $d(P)=d(i,j)*d(j,k)*\ldots*d(l,m)$
- *Concave if* $d(P)=\min\{d(i,j), d(j,k), \ldots, d(l,m)\}$

จากนิยามข้างบนเมตริกซ์ ความหน่วงเวลา จิตเตอร์ ค่าใช้จ่าย และจำนวนเราเตอร์ที่ผ่านอยู่ในประเภท Additive และความน่าเชื่อถือ อยู่ในประเภท Multiplicative แบนด์วิดท์อยู่ในประเภท Concave [6]

เป็นที่ทราบกันดีว่าทฤษฎีของ CBR คือการคำนวณหาเส้นทางโดยมีเงื่อนไขหรือเมตริกซ์ที่พิจารณามากกว่า 2 เมตริกซ์ ส่วนใหญ่แล้วแบนด์วิดท์และจำนวนเราเตอร์ที่ผ่านในเส้นทางถูกใช้เป็นเมตริกซ์ที่ดีกว่าความหน่วงเวลาและจิตเตอร์ เนื่องจาก

1. ถึงแม้ว่าแอปพลิเคชั่นจะให้ความสำคัญกับความหน่วงเวลาและจิตเตอร์แต่แอปพลิเคชั่นส่วนใหญ่ก็สามารถทนต่อความหน่วงเวลาและจิตเตอร์ได้
2. แอปพลิเคชั่นแบบเรียลไทม์ (Real-Time) ส่วนมากต้องการจำนวนแบนด์วิดท์ที่แน่นอน ดังนั้นใช้แบนด์วิดท์เป็นเมตริกซ์จะมีประโยชน์ต่อแอปพลิเคชั่นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรียลไทม์มากกว่า จำนวนเราเตอร์ที่ผ่านก็สำคัญเช่นกัน เนื่องจากยิ่งผ่านเราเตอร์มากเท่าใดหมายถึงการใช้ทรัพยากรของเครือข่ายมากขึ้นไปด้วย

ใน CBR เส้นทางสามารถคำนวณได้เมื่อต้องการ (On-Demand) หรือสามารถคำนวณไว้ก่อน (Pre-Compute) สำหรับแต่ละระดับของทราฟฟิก การคำนวณเมื่อต้องการจะถูกกระตุ้นเมื่อมีร้องขอมาดังนั้นจึงทำให้มีการคำนวณบ่อยมาก และเป็นโหลดงานที่สูงสำหรับเราเตอร์ วิธีการที่จะลดการคำนวณของ CBR ได้แก่

1. ใช้การจับเวลามาช่วย เช่นทุก ๆ 30 วินาทีจึงจะคำนวณ
2. เลือกแบนด์วิดท์และจำนวนเราเตอร์ที่ผ่านเป็นเมตริกซ์
3. กำหนดนโยบายในการตัดเส้นทางที่ไม่เหมาะสมออกก่อนที่จะทำการคำนวณตารางเส้นทาง เช่น ถ้าต้องเส้นทางที่ความหน่วงเวลาต่ำ ๆ ดังนั้นเส้นทางที่มีความหน่วงเวลาสูง ๆ เช่น ดาวเทียมความจะถูกตัดทิ้งก่อนที่จะคำนวณ

2.6 ข้อดีและข้อเสียของ Constraint Based Routing

ข้อดีของ Constraint Based Routing คือ

1. ตรงกับความต้องการของ QoS มากกว่า
2. เพิ่มประสิทธิภาพการใช้เครือข่ายมากขึ้น (Network Utilization)

ข้อเสียของ Constraint Based Routing คือ

1. เพิ่มภาระในการคำนวณและการติดต่อสื่อสารมากขึ้น
2. ขนาดของตารางเส้นทาง (Routing Table)
3. เส้นทางที่ยาวมากกว่าจะถึงแม้จะได้ QoS แต่ก็ใช้ทรัพยากรมากขึ้นด้วยเช่นกัน
4. มีความไม่เสถียรสูง (Instability)

2.6.1 โครงสร้างและขนาดของตารางเส้นทาง

โครงสร้างและขนาดของตารางเส้นทางมีผลโดยตรงกับเมตริกซ์ เราสามารถมองโครงสร้างของตารางเส้นทางเป็นอาเรย์ 2 มิติ จำนวนแถวหมายถึง เงื่อนไขในการเลือกเส้นทาง (Routing Granularity) และจำนวนคอลัมน์หมายถึงจำนวนเมตริกซ์ ตัวอย่างเช่นการหาเส้นทางโดยเลือกแบนด์วิดท์และจำนวนเราเตอร์ที่ผ่านเป็นเมตริกซ์และยึดปลายทางเป็นหลัก (Base on Destination) ตารางเส้นทางจะมีขนาดอาร์เรย์ $K \times H$ ซึ่ง K คือจำนวนของปลายทาง (Destination) และ H คือจำนวนสูงสุดของเราเตอร์ที่ผ่านและแต่ละเส้นทางจะต้องมีแบนด์วิดท์ที่สอดคล้องกับความต้องการด้วย

จะเห็นว่าตารางเส้นทางของ CBR จะมีขนาดใหญ่กว่าตารางเส้นทางแบบปกติจึงทำให้เกิดข้อเสียคือเปลืองพื้นที่ในการจัดเก็บ (Storage Overhead) ทำให้การหาเส้นทางนั้นใช้เวลามากขึ้น วิธีการที่จะลดขนาดของตารางเส้นทางของ CBR คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ใช้ เงื่อนไขในการเลือกเส้นทาง (Routing Granularity) ที่ไม่มีคุณภาพ
2. ใช้วิธี Hop Quantization เช่น แบ่งจำนวนเราเตอร์เป็นช่วง ๆ เพื่อลดจำนวนคอลัมน์ของตารางเส้นทาง
3. เก็บข้อมูลในตารางเส้นทางเป็นแบบ Best-Effort traffic และทำการคำนวณหาเส้นทางเพื่อให้ได้ QoS เมื่อต้องการส่งข้อมูล แต่วิธีการนี้ต้องเสียเวลาในคำนวณ

2.6.2 การเลือกระหว่างการใช้ทรัพยากรให้น้อยที่สุด (Resource Conservation) และการกระจายภาระงานในเครือข่าย (Load Balancing)

วิธี Constraint Based Routing จะต้องเลือกเส้นทางใดเส้นทางหนึ่งที่มีลักษณะดังต่อไปนี้

1. *Widest-Shortest Path* คือเส้นทางที่มีจำนวนเราเตอร์ที่ผ่านน้อยสุด และถ้ามีเส้นทางที่มีจำนวนเราเตอร์ที่ผ่านน้อยสุดมีมากกว่า 1 เส้นทางจะเลือกเส้นทางที่มีแบนด์วิดท์มากที่สุด
2. *Shortest-Widest Path* คือเส้นทางที่มีแบนด์วิดท์มากที่สุด และถ้ามีมากกว่า 1 เส้นทางจะเลือเส้นทางที่มีจำนวนเราเตอร์ที่ต้องผ่านน้อยสุด
3. *Shortest-Distance Path* คือเส้นทางที่เลือกผ่านจำนวนเราเตอร์น้อยสุด นิยามคือ

$$dist(P) = \sum_{i=1}^{k} \frac{1}{r_i} \qquad (2.1)$$

เมื่อ r_i คือแบนด์วิดท์ของเส้นทาง i

การเลือกใช้เส้นทางอื่นที่ไม่ใช่เส้นทางที่สั้นที่สุดต้องใช้ทรัพยากรของเครือข่ายมากขึ้น ทำให้ประสิทธิภาพของเครือข่ายน้อยลงเมื่อมีทราฟฟิกสูง ๆ การเลือกระหว่างจะใช้ทรัพยากรของเครือข่ายน้อย ๆ หรือการกระจายทราฟฟิกไปใช้เส้นทางอื่นที่ไม่ใช่เส้นทางที่สั้นสุดเป็นสิ่งสำคัญ วิธีการเลือกเส้นทางที่สั้นที่สุดนั้นเป็นวิธีการเลือกเส้นทางที่ใช้อยู่ในปัจจุบัน วิธีการที่สองเลือกเส้นที่กระจายโหลดทราฟฟิกโดยการเลือกใช้เส้นทางที่มีแบนด์วิดท์สูง ๆ ในเครือข่ายแทนที่จะเลือกเส้นทางที่สั้นที่สุด วิธีการที่สามคือการเลือกใช้ทั้ง 2 วิธี เมื่อเครือข่ายมีทราฟฟิกสูง ๆ จะใช้วิธีเลือกเส้นทางที่สั้นที่สุดและเมื่อทราฟฟิกปานกลางถึงน้อย จะเลือกใช้วิธีเลือกแบนด์วิดท์ที่สูง ๆ เพื่อกระจายทราฟฟิกในเครือข่าย

2.6.3 ความมีเสถียรภาพ (Stability)

เนื่องจากอัลกอริทึมของ Constraint Based Routing จะทำการคำนวณที่ซับซ้อนมากกว่าเนื่องจากมีเงื่อนที่ใช้ในการคำนวณมากกว่า Dynamic Routing Algorithm จึงทำให้มีความมีเสถียรภาพต่ำ

ความมีเสถียรภาพของเครือข่ายกับ Constraint Based Routing ขึ้นอยู่กับ เงื่อนไขในการเลือกเส้นทาง (Routing Granularity) ถ้าการเลือกเส้นทางใช้เงื่อนไขที่ไม่ดี เช่นการเลือกเส้นทางที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่บนพื้นฐานของที่อยู่ของปลายทาง เมื่อเส้นทางระหว่างต้นทางและปลายทางเริ่มมีความหนาแน่นของข้อมูล ทราฟฟิกจะเปลี่ยนเส้นทางไปใช้เส้นทางอื่น ทำให้เส้นทางอื่นอาจจะเกิดความหนาแน่นได้เช่นกัน และทราฟฟิกจะเปลี่ยนเส้นทางอีกครั้ง

การที่ Constraint Based Routing มีการคำนวณที่สูงทำให้เกิดความไม่มีเสถียรภาพบนเครือข่ายได้ เมื่อเราเตอร์ต้องยุ่งอยู่กับการคำนวณตารางเส้นทาง มันจะทำงานช้าลงเมื่อมีการเปลี่ยนโทโพโลยี ถ้าต้องการให้เครือข่ายมีความเสถียรอาจจะใช้วิธีการการจับเวลาเข้ามาช่วยเช่น จะให้คำนวณทุกๆ กี่นาที หรือพยายามลดความซับซ้อนของการคำนวณหาเส้นทางจะช่วยให้มีเสถียรภาพมากขึ้น สรุปคือ Constraint Based Routing ต้องใช้อย่างระมัดระวัง

2.7 ความสำคัญของการเลือกเส้นทางเพื่อรองรับ QoS

ปัจจุบันมีผู้คิดค้นวิธีการและเทคนิคเพื่อรองรับแอปพลิเคชั่นใหม่ๆ ที่จะเกิดทั้งในปัจจุบันและอนาคตมากมาย ส่วนของการหาเส้นทาง (Routing) เพื่อทำให้เกิด QoS เป็นส่วนสำคัญส่วนหนึ่งที่จะให้เกิด QoS ได้ จากการเลือกเส้นทางด้วยวิธีแบบเดิมคือเลือกเส้นทางที่สั้นที่สุดไม่ใช่วิธีที่จะทำให้เกิด QoS ได้ จึงต้องเลือกใช้เส้นทางอื่นที่ตรงกับความต้องการเพื่อให้เกิด QoS วิธีการในการเลือกเส้นทางเหล่านี้เป็นสิ่งที่น่าศึกษา นอกจากนั้นการเลือกเส้นทางนี้จะต้องไม่ส่งผลกระทบต่อทราฟฟิกแบบเดิมคือ Best-Effort จะต้องไม่ทำให้เส้นทางที่ Best-Effort ใช้เกิดความหนาแน่นของข้อมูลเกิดขึ้น นอกจากนั้นต้องไม่ทำให้เครือข่ายมีเสถียรภาพต่ำลง ซึ่งเป็นสิ่งที่น่าศึกษาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการเพิ่มประสิทธิภาพสำหรับกลไกการจองทรัพยากรในเครือข่าย

3.1 ความรู้พื้นฐาน

การทำงานของอินเทอร์เน็ตในปัจจุบันนี้ไม่รองรับการทำงานของ Quality-of-Service นั่นคือการจองทรัพยากรก่อนทำการส่งข้อมูล แพ็กเก็ตข้อมูลถูกส่งไปในหลาย ๆ เส้นทางและถึงปลายทางอย่างไม่เป็นลำดับ เครือข่ายในอนาคตนั้นจะต้องรองรับข้อมูลประเภทมัลติมีเดียหรือแอปพลิเคชั่นแบบเรียลไทม์มากขึ้น ซึ่งแอปพลิเคชั่นเหล่านี้ต้องการการประกันบริการในการส่งข้อมูลเช่น แบนด์วิดท์ ความหน่วงเวลา เป็นต้น โพรโทคอลที่ใช้ในเครือข่าย IP ในปัจจุบันไม่ได้มีการประกันในการส่งข้อมูลแต่อย่างใด ซึ่งมีลักษณะที่เรียกว่า Best-Effort ซึ่งการเลือกเส้นทางนั้นไม่คำนึงถึงตัวแปรต่าง ๆ ของ QoS ทำให้เส้นทางเหล่านั้นไม่สามารถรองรับความต้องการของแอปพลิเคชั่นใหม่ ๆ ที่เกิดขึ้นได้ การเลือกเส้นทางที่สั้นแต่มีทรัพยากรไม่เพียงพอในขณะที่ยังมีเส้นทางอื่น ๆ ที่มีทรัพยากรเพียงพอแต่อาจจะไม่ใช่เส้นทางที่สั้นที่สุด ทำให้เพิ่มอัตราการเกิดการปฏิเสธการร้องขอ (Call Blocking) ได้ การเลือกเส้นทางที่ถูกต้องนั้นได้มาจากการมีตารางเส้นทางที่ดี ที่บอกถึงเส้นทางที่มีทรัพยากรเพียงพอกับความต้องการได้ วิธีการเลือกเส้นทางในปัจจุบันนี้แบ่งเป็น 3 ประเภท [7] คือ

1. Source Routing คือการที่ให้เราเตอร์ต้นทาง (Source Router) ทำการคำนวณหาเส้นทางทั้งหมด ซึ่งมีข้อเสียคือต้องมีการเก็บและปรับปรุงข้อมูลของเครือข่ายทั้งหมด (Global Network State Information) อยู่เสมอซึ่งทำให้เกิดโอเวอร์เฮด (Overhead) สูงเมื่อต้องมีการปรับปรุงข้อมูลตลอดเวลา แต่ถ้าไม่ทำตลอดเวลาจะทำให้ข้อมูลที่มีนั้นไม่ใช่ข้อมูลที่เป็นจริง ณ ปัจจุบัน (Imprecise) ดังนั้นจึงต้องเลือกระหว่างการปรับปรุงข้อมูลตลอดเวลาเวลา ซึ่งทำให้มีแพ็กเก็ต ในเครือข่ายสูงเพราะต้องมีการแลกเปลี่ยนข้อมูลกันตลอดเวลาหรือมีการอัพเดตข้อมูลน้อยครั้งลงแต่ข้อมูลนั้นไม่เป็นปัจจุบัน

2. Distributed Routing เป็นการคำนวณเส้นทางโดยการแบ่งไปยัง เราเตอร์อื่น ๆ ในเครือข่าย แต่ละเราเตอร์จะมีตารางเส้นทางของเราเตอร์ที่ติดกัน แพ็กเก็ตจะมีส่วนหัวที่เล็กกว่าเมื่อเทียบกับแพ็กเก็ตของ Source Routing นอกจากนั้นวิธีการนี้เป็นอัลกอริทึมแบบกระจายการคำนวณ (Distributed Computation Algorithm) ซึ่งจะใช้พื้นที่ในหน่วยความจำในเราเตอร์น้อยกว่า สรุปคือวิธีการนี้จะมีโอเวอร์เฮดน้อยกว่า Source Routing [5] ในงานวิจัยนี้จะอยู่ในลักษณะของ Distributed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Routing และแพ็กเก็ตจะถูกส่งไปโดยพิจารณาถึงความต้องการการประกันคุณภาพ (QoS Requirement) ด้วย แต่การทำงานของงานวิจัยนี้เป็นแบบมีผู้รับคนเดียว (Unicast) และสมมุติว่าทำงานในเครือข่ายประเภทเดียวกัน (Homogenous Network)

3. Hierarchical Routing จะมีความเหมาะสมกับเครือข่ายขนาดใหญ่มาก ๆ เราเตอร์จะมีหลายระดับ ระดับล่างสุดจะเป็นเราเตอร์จริง ๆ เราเตอร์เหล่านี้จะรวมแบบลอจิกคอลเป็นกลุ่ม (Logical Group) และมีตัวแทนกลุ่มเรียกว่าลอจิกคอลโหนด (Logical Node) ซึ่งจะอยู่ในระดับที่สูงกว่าและในระดับเดียวกันนั้นก็สามารถรวมกันเป็นกลุ่มและมีตัวแทนกลุ่มในระดับที่สูงกว่าได้เช่นกัน และข้อมูลของโหนดต่าง ๆ จะรวมกันอยู่ที่ลอจิกคอลโหนด และมีข้อมูลของกลุ่มอื่น ๆ ด้วย ตัวอย่างเช่นโพรโทคอลที่ใช้อัลกอริทึมนี้ได้แก่ PNNI

โพรโทคอล RSVP [1] ได้ถูกเสนอขึ้นเพื่อใช้ในการจองทรัพยากรบนเครือข่ายอินเทอร์เน็ต อย่างไรก็ตาม RSVP ไม่ใช่โพรโทคอลในการหาเส้นทางและใช้ตารางเส้นทางในเครือข่าย IP เป็นหลัก ดังนั้นจึงไม่มีประโยชน์เนื่องจากตารางเส้นทางดังกล่าวไม่ได้บอกถึงทรัพยากรที่เส้นทางนั้นมีด้วย ดังนั้นเครือข่าย IP จึงต้องมีการปรับปรุงอัลกอริทึมในการหาเส้นทางที่ใช้อยู่ในปัจจุบัน แต่อย่างไรก็ตามควรมีการเปลี่ยนแปลงอัลกอริทึมที่ใช้ในปัจจุบันให้น้อยที่สุดเพื่อความสะดวกในการนำไปใช้และเข้ากับแอปพลิเคชั่นหรือส่วนอื่น ๆ ในเครือข่ายในปัจจุบันได้

การหาเส้นทางจะมีโอเวอร์เฮดอยู่ 3 ประการ [8] คือ 1) จากการคำนวณ 2) จากการใช้ทรัพยากรในการเก็บข้อมูล 3) จากการติดต่อสื่อสารกันระหว่างเราเตอร์ จะเห็นว่าอัลกอริทึมแบบ Source Routing ที่มีโอเวอร์เฮดสูงจากการคำนวณและมีขนาดตารางเส้นทางที่ใหญ่และต้องคอยปรับปรุงข้อมูลของเครือข่ายทั้งหมด และความไม่ถูกต้องของข้อมูลทำให้ประสิทธิภาพของอัลกอริทึมต่ำลง การทำให้ข้อมูลในตารางเส้นทางมีความถูกต้องมากที่สุดนั้นเป็นเรื่องสำคัญและทำได้ยาก เพราะการปรับปรุงข้อมูลบ่อย ๆ ทำให้เกิดโอเวอร์เฮดสูงจึงมีผู้นำเสนอใน [11] ถ้าหากไม่มีการเก็บข้อมูลเหล่านี้ไว้ เทคนิคการ Flooding [9],[10] ได้ถูกนำเสนอขึ้นเพื่อใช้ในการสร้างเส้นทางโดยจะถูกส่งออกไปยังเส้นทางออกทุก ๆ เส้นทางจากเราเตอร์เพื่อหาเส้นทางที่มีทรัพยากรเพียงพอกับความต้องการ แต่วิธีการนี้ยังมีโอเวอร์เฮดสูงในการส่งแพ็กเก็ตออกไปเพื่อสร้างการเชื่อมต่อ วิธีแก้คือมีการกำหนดจำนวนเส้นทางที่จะส่งแพ็กเก็ตออกไปเพื่อสร้างการเชื่อมต่อแทนที่จะส่งไปทุกเส้นทางออกได้ถูกนำเสนอขึ้นใน [12] แต่วิธีการดังกล่าวใช้ได้ดีเฉพาะในช่วงเวลาที่มี ทราฟฟิกน้อย ๆ ในกรณีที่มีทราฟฟิกสูง ๆ ประสิทธิภาพยังไม่ดีกว่าเทคนิคการ Flooding

สิ่งที่น่าสนใจอีกอย่างหนึ่งคือ โอเวอร์เฮดในการคำนวณหาเส้นทางซึ่งวิธีการหาเส้นทางแบบ Distributed Routing นั้นจะมีโอเวอร์เฮดสูงเนื่องจากจะต้องเสียเวลาในการคำนวณหาเส้นทางในทุก ๆ เราเตอร์ วิธีหนึ่งที่น่าสนใจและจะนำมาใช้ในงานวิจัยนี้คือการเก็บข้อมูลของเราเตอร์ที่อยู่ติดกัน (Immediate Neighbor) และเราเตอร์ที่อยู่ถัดจากเราเตอร์ที่อยู่ติดกันนั้นไปอีกหนึ่งเราเตอร์ (Second-Degree Neighbor) ซึ่งถูกเสนอใน [12] ข้อดีของวิธีนี้คือ มีจำนวนของแพ็กเก็ตในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แลกเปลี่ยนข้อมูลมีไม่มากเนื่องจากจะมีการแลกเปลี่ยนข้อมูลเฉพาะกับเราเตอร์ที่ติดกันเท่านั้นและความถูกต้องของข้อมูล (Impreciseness) ไม่ยากแก่การปรับปรุงค่าจนเกินไปเนื่องจากเก็บข้อมูลของ Second-Degree Neighbor เท่านั้นจะไม่ใหญ่เท่ากับการเก็บข้อมูลทั้งเครือข่ายเหมือนใน Source Routing ข้อดีอีกประการหนึ่งคือ เนื่องจากมีการเก็บข้อมูลของ Second-Degree Neighbor ทำให้ส่งข้อมูลได้ถูกต้องมากขึ้นเนื่องจากเราเตอร์สามารถเห็นเป็น 2 ระดับ ดังนั้น โอเวอร์เฮดในการสร้างการเชื่อมต่อจึงลดลงนั่นคือการสร้างการเชื่อต่อจะทำสำเร็จในอัตราที่เพิ่มขึ้นและการคำนวณจะทำน้อยลงหากปลายทางอยู่ภายใน Second-Degree Neighbor รวมถึงแพ็กเก็ตในการ Flooding จะมีจำนวนน้อยลงด้วย แต่วิธีใน [12] นี้ไม่ได้ทำการเปลี่ยนแปลงอัลกอริทึมการจองทรัพยากรแต่อย่างใด

3.2 วิธีการเดิมในการหาเส้นทาง QoS ในเครือข่าย IP QoS (Quality-of-Service in IP Networks)

เราสมมุติว่าทุกเราเตอร์จะมีข้อมูลของทรัพยากรของตนและทุก ๆ เราเตอร์มีนโยบายว่าเมื่อใดจะทำการปรับปรุงข้อมูลในตารางเส้นทางของตน แต่ละเราเตอร์จะทำการปรับปรุงตารางเส้นทางของตนเอง ซึ่งตารางเส้นทางเหล่านี้จะให้ข้อมูลเกี่ยวกับทรัพยากรในแต่ละเส้นทางเพื่อให้ได้ QoS ตามที่แต่ละแพ็กเก็ตต้องการ

3.2.1 โครงสร้างของตารางเส้นทาง (Routing Table Structure)

ตารางเส้นทางนี้จะใช้ในการส่งต่อแพ็กเก็ตออกไปซึ่งวิธีการนี้ถูกนำเสนอขึ้นใน [12] เพื่อลดโอเวอร์เฮดในการคำนวณหาเส้นทาง ก่อนที่จะอธิบายต่อไปจะกำหนดสัญลักษณ์ที่จะใช้อธิบายดังต่อไปนี้

- $N^1(v)$ หมายถึงเราเตอร์ที่อยู่ติดกันกับเราเตอร์ v
- $E^1(v)$ หมายถึงเส้นทางที่เชื่อมต่อระหว่างเราเตอร์ v และ $N^1(v)$
- $N^2(v)$ หมายถึงเราเตอร์ที่อยู่ติดกันกับเราเตอร์ $N^1(v)$
- $E^1(v)$ หมายถึงเส้นทางที่เชื่อมต่อระหว่างเราเตอร์ $N^1(v)$ และ $N^2(v)$

โดยโครงสร้างของตารางเส้นทางแสดงดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Next Hop	Link	Available Metric
v_1	l_1^1	$x_1 B$
:	:	:
v_n	l_n^1	$x_n B$
u_1	$< l_1^1, l_1^2 >$	$x_k B$
:	:	:
u_m	$< l_m^1, l_m^2 >$	$x_r B$

รูปที่ 3.1 โครงสร้างตารางเส้นทางของเราเตอร์ v

โครงสร้างตารางเส้นทางของเราเตอร์ v ประกอบด้วยข้อมูลเกี่ยวกับเมตริกซ์ของทุก ๆ เส้นทางเชื่อมโยง (links) ใน $E^1(v)$ เรียกว่า Fist-Level Entries และ ข้อมูลของทุก ๆ เส้นทางใน $E^2(v)$ เรียกว่า Second-Level Entries ซึ่งจะแสดงในรูปแบบ $< l_i^1, l_j^2 >$ ซึ่ง $l_i^1 \in E^1(v)$ และ $l_j^2 \in E^2(v)$ เราเตอร์ v จะได้รับข้อมูลเกี่ยวกับ QoS Metrics ได้จาก $N^1(v)$ โดยการแลกเปลี่ยนแพ็กเก็ตข้อมูล และ $N^2(v)$ ที่เรียกว่า Hello2 แต่จะบ่อยครั้งแค่ไหนนั้นขึ้นกับนโยบายที่กำหนดไว้ ส่วน $x_n B$ คือข้อมูลที่จะใช้เป็นเงื่อนไข เช่น แบนด์วิดท์ หรือ ความหน่วงเวลา เป็นต้น

3.2.2 กลไกการส่งต่อแพ็กเก็ต (Packet Forwarding Mechanism)

อัลกอริทึมในการเลือกเส้นทางในงานวิจัยนี้จะเป็นอัลกอริทึมที่จะเลือกเส้นทางที่ให้ได้ความต้องการ QoS โดยเมตริกซ์ที่สำคัญคือแบนด์วิดท์ การสร้างการเชื่อมต่อมี 3 ขั้นตอนคือ 1). Probing 2). Ack 3). Failure Handling

ขั้นตอนการทำงานของวิธิการเดิม [12]

ขั้นที่ 1 Probing

1. ผู้ส่งทำการส่งคำร้องขอออกไป แต่ละคำร้องขอจะมีตัวเลขประจำคำร้องนั้น ๆ(cid) แต่ละคำร้องขอจะเรียกว่า "*Probe*" ซึ่งมีรูปแบบคือ

   [*k, QoS(Bandwidth = B), s, t, cid, {l}*]

   s = source request

   cid = connection identifier

   QoS(Bandwidth = B) = Metric requires

   t = destination

   k = router that has forwarded the probe

   {l} = list of neighbors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เราเตอร์ที่ติดกัน (v) ตรวจสอบว่า Probe นี้เคยมาแล้วหรือยัง ถ้าเคยมาแล้วจะทิ้งไป แต่ถ้ามาเป็นครั้งแรกจะทำการเก็บข้อมูล ไว้ว่ามาจาก cid อะไรและใครเป็นผู้ส่งมา
3. ทำการเก็บข้อมูลของเราเตอร์ที่ส่ง Probe มาให้ ($p_v(cid)$) และทำการส่ง Probe ต่อไป
4. เราเตอร์ (v) ทำการตรวจสอบว่าปลายทางอยู่ในตารางเส้นทางเป็น Fist-Level หรือ Second-Level Entry หรือไม่ ถ้ามีและเป็นเส้นทางที่ตรงกับ QoS Requirement จะทำการส่ง Probe ไปให้ตามเส้นทางที่ระบุไว้ทันที ถ้าไม่อยู่ในตารางเส้นทาง แสดงว่าปลายทางอยู่ไกลออกไปมากกว่า 2 เราเตอร์จะทำการส่งต่อไปตาม list {I}
5. เราเตอร์ใน list {I} จะทำซ้ำขั้นตอนที่ 1 -4 ไปเรื่อย ๆ จนกว่าจะถึง ปลายทาง

หมายเหตุ:ใน phase I นี้ไม่มีการจองทรัพยากรแต่อย่างใด

ขั้นที่ 2 Ack

6. Ack Message จะคล้ายกับ Probe แต่ว่าไม่มี {I} เราเตอร์จะทำการตรวจสอบทรัพยากรในเส้นทางที่ได้รับ Ack มาว่ามีแบนด์วิดท์เพียงพอกับที่ร้องขอหรือไม่ ถ้ามีจะทำการจองทรัพยากรไว้ สำหรับ cid ดังกล่าวและเก็บข้อมูลไว้เป็น Downstream Router's ID แทนโดยสัญลักษณ์ $n_v(cid)$
7. เราเตอร์จะทำการส่ง Ack ต่อไปโดยใช้ค่าของ ($p_v(cid)$)
8. ทำซ้ำ 6-7 ไปเรื่อย ๆ จนกว่า source จะได้รับ Ack
9. เมื่อ Source ได้รับ Ack แล้วถือว่าการสร้างการเชื่อมต่อสมบูรณ์

ขั้นที่ 3 Failure Handling

10. ถ้าไม่มีทรัพยากรพอจะส่ง Error Message ไปยัง Downstream Router เมื่อ Router ได้รับ Error Message จะทำการปล่อยทรัพยากรทันที
11. ทำซ้ำ 10 ไปเรื่อย ๆ จนกว่าทุกเราเตอร์จะปล่อยทรัพยากรหมด

3.2.3 วิเคราะห์วิธีการเดิม

จะสังเกตเห็นว่าวิธีการดังกล่าวนั้นยังมีข้อเสียบางประการเช่น

- กว่าจะรู้ว่าเราเตอร์ในเส้นทางดังกล่าวนั้นมีทรัพยากรไม่เพียงพอ ต้องรอจนกว่า Ack Message จะตรวจสอบเจอในขากลับ ทำให้เสียเวลานาน
- อัตราการเกิด Call Blocking มีสูงเนื่องจากไม่มีการจองทรัพยากรไว้ตั้งแต่แรก ในช่วงเวลา Probing อาจจะโดนผู้ส่งคนอื่นจองทรัพยากรไปก่อน ทำให้ถูกปฏิเสธการจองแบนด์วิดท์ โอกาสที่เกิดในการล้มเหลว (Fail) จึงสูงตามไปด้วย
- จำนวนแพ็กเก็ตที่เกิดจากการสร้างการเชื่อมต่อในกรณีที่เกิดการล้มเหลวจะมีจำนวนสูงซึ่งอาจเป็นสาเหตุทำให้เกิดความคับคั่งของข้อมูลบนเส้นทางดังกล่าวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเพิ่มประสิทธิภาพการหาเส้นทาง QoS

ในส่วนของโครงสร้างของตารางเส้นทางนั้นจะใช้วิธีการเดิมใน [12] ซึ่งอธิบายแล้วในหัวข้อ 3.2.1 แต่งานวิจัยนี้จะปรับปรุงอัลกอริทึมในส่วนของหัวข้อ 3.2.2 ทั้งนี้เนื่องจากข้อเสียของวิธีการดังกล่าวที่แสดงในหัวข้อ 3.2.3 คือการลดอัตราการเกิด call blocking และลดเวลาที่ใช้ในการสร้างการเชื่อมต่อและเวลาที่ใช้ในการหาเส้นทาง โดยวิธีการใหม่นี้จะประกอบด้วย 2 ขั้นตอน ดังนี้

- Probing Phase ขั้นตอนนี้จะเริ่มจากผู้ส่งไปยังผู้รับ ในเราเตอร์ระหว่างทางจะมีการเรียกใช้ Admission Test เพื่อตรวจสอบว่ามีทรัพยากรเพียงพอหรือไม่ และมีการจองทรัพยากรไว้หากมีทรัพยากรเพียงพอกับความต้องการ
- Ack Phase เป็นขั้นตอนที่ส่งจาก ผู้รับทำการส่งข้อมูลเพื่อยืนยันการร้องขอ

ในขั้นตอนแรกเราเตอร์ผู้ส่งจะส่งแพ็กเก็ต Probe ออกไปซึ่งอยู่บนพื้นฐานของวิธี Flooding นั่นคือส่งแพ็กเก็ต Probe ออกไปทุก ๆ เส้นทาง เมื่อเราเตอร์ระหว่างทางได้รับแพ็กเก็ต Probe นั้นจะตรวจสอบว่าแพ็กเก็ต Probe นั้นเคยได้รับแล้วหรือยัง ถ้าตรวจสอบว่าเคยได้รับแล้ว จะไม่สนใจแพ็กเก็ต Probe นั้น แต่ถ้ายังไม่ได้รับ จะบันทึกข้อมูลไว้และตรวจสอบว่ามีทรัพยากรตามที่ร้องขอมาหรือไม่ถ้ามีทรัพยากรเพียงพอจะทำการจองทรัพยากรนั้นไว้ให้ ถ้ามีไม่พอจะทำการส่ง Error Message กลับไป หลังจากนั้นจะทำการหาข้อมูลในตารางเส้นทาง หากปลายทางอยู่ในตารางเส้นทางจะส่งไปยังปลายทางโดยตรงตามเส้นทางที่ระบุไว้ในตารางเส้นทางนั้น หากปลายทางไม่อยู่ในตารางเส้นทางแสดงว่าปลายทางอยู่ไกลเกิน 2 เราเตอร์ที่อยู่ติดกันกับเราเตอร์ปัจจุบัน ดังนั้นจะทำการหาเส้นทางที่จะส่งต่อแพ็กเก็ต Probe นั้นต่อไปเพื่อให้ถึงปลายทางโดยเริ่มใช้วิธีการ Flooding อีกครั้ง ทำซ้ำอย่างนี้ไปเรื่อย ๆ จนกว่าแพ็กเก็ต Probe จะถึงปลายทาง และเริ่มทำ Ack phase ต่อไป ในขั้นตอนนี้จะมีฟังก์ชั่นในการทำงานดังนี้คือ ProbRsv(), SetStatus(), Forward(), Release(), Backtrack(), และ Reject(). ซึ่งจะอธิบายการทำงานดังต่อไปนี้

- ProbeRsv(Probe_pkt) ฟังก์ชันนี้จะทำการตรวจสอบแพ็กเก็ต Probe ซึ่งจะมีรูปแบบที่มีการปรับปรุงเล็กน้อยคือเพิ่มข้อมูลของผู้รับแพ็กเก็ต Probe เข้าไปดังนั้นแพ็กเก็ต Probe จะมีรูปแบบดังนี้ [*k ,r, QoS(Bandwidth = B), s, t ,pid, {l}*]

  k = router that has forwarded the probe

  r= receiver

  QoS(Bandwidth = B,D=Delay) Metric requires

  s = source requesting

  t = destination

  pid = probe identifier

  {l} = list of visited node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และทำการเรียกใช้ Admission Test หากมีทรัพยากรเพียงพอตามที่ระบุในแพ็กเก็ต Probe จะทำการจองทรัพยากรนั้นไว้ หากมีทรัพยากรไม่เพียงพอจะส่ง Error Message กลับไปยังผู้ที่ส่งแพ็กเก็ต Probe นี้มา

```
ProbeRsv (Probe_pkt) { // Check Probe in list
 If  found probe in list //this probe not first pkt
    Reject(Probe_pkt)// probe_pkt is rejected
Else {
    if (call admission is successful) { //check available bandwidth
        Reserve bandwidth on preferred link.
        SetStatus(wait)
        return (success)
     }
    else return (failure).
  }// end if
} //end function
```

- SetStatus(status) ฟังก์ชั่นนี้จะกำหนดสถานะของการจองทรัพยากร หากจองทรัพยากรได้และกำลังรอการ Ack จะกำหนดสถานะให้เป็น Wait เพื่อให้ผู้ส่งคนอื่นสามารถใช้ทรัพยากรได้ในขณะที่รอ Ack อยู่นั้น และเมื่อได้รับ Ack จะกำหนดให้เป็น Reserve เพื่อเตรียมทรัพยากรไว้สำหรับการส่งข้อมูลเมื่อ Ack message เดินทางไปถึงผู้ส่ง สถานะของทรัพยากรจะมี 2 สถานะ คือ Used และ Unused
- Forward(Probe_pket) ฟังก์ชั่นนี้จะทำการส่งแพ็กเก็ต Probe ไปยังเส้นทางต่าง ๆ ที่หาได้จากตารางเส้นทาง หากพบว่าปลายทางอยู่ในตารางเส้นทางของตนแล้วจะส่งแพ็กเก็ต Probe ไปยังเส้นทางที่ระบุไว้ทันที และจะมีการเรียกใช้ฟังก์ชั่น Reserve() เพื่อทำการจองทรัพยากร หากจองทรัพยากรไม่ได้จะเรียกใช้ฟังก์ชั่น Release() เพื่อทำการปล่อยทรัพยากร

```
Forward(Probe_pkt) {
        If (receiver is in routing table )
                Sent probe packet direct to destination t
        Else {
                For (i=1,i<=k, i++){ // send probe packet in k path (Flooding)
                        send Probe_pkt[i] to R[i]   //R[i] =  Next Router[i]
                        status[i]=Reserve(Probe_pkt)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    if(status[i] is unsuccessful){
                        Release(Probe_pkt)
                    } // end if
                } // end for
            } // else
        }// end function forward
```

- Release(Probe_pkt) ฟังก์ชั่นนี้ใช้ในการปล่อยทรัพยากรที่จองไว้ในเส้นทางระหว่าง k และ r ที่ระบุไว้ในแพ็กเก็ต Probe
- Backtrack(Probe_pkt) ฟังก์ชั่นนี้จะใช้เมื่อเกิด error ขึ้นที่โหนด r นั่นคือแพ็กเก็ต Probe ที่ไปถึงไม่สามารถจองทรัพยากรได้ ดังนั้นจะต้องกลับไปที่โหนดก่อนหน้า r คือ k นั่นเอง
- Reject(Probe_pkt) ฟังก์ชั่นนี้ใช้เมื่อเราเตอร์ได้รับแพ็กเก็ต Probe ซ้ำกันที่มาทีหลัง

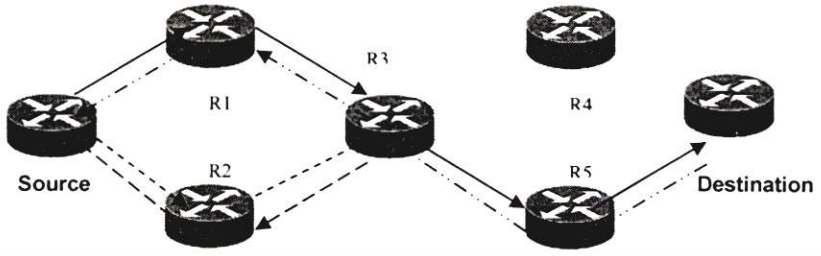
```
Reject(Probe_pkt){
 for(i=1,i<=count_of_probe_list,i++) {  //check incoming probe with probe_list
  if cid == probe packet in probe list
     {probe_pkt is rejected
      send error back to previous router for release resource
     }
  } //end for
 }// end function
```

ในส่วนของ Ack phase นั้นจะมีการใช้ฟังก์ชั่น Ack(Ack_pkt) ในการส่ง Ack message กลับไปยังผู้ส่งโดยใช้เส้นทางที่เก็บไว้ในแพ็กเก็ต Probe เพื่อเป็นการยืนยันกับผู้ส่งว่าสามารถจองทรัพยากรได้แล้ว และให้เริ่มส่งข้อมูลจริงๆ ได้

```
Ack(Ack_pkt){
  For (i=1,i<=count_probe_in_list,i++){
    If aid==probe_id_in_list {
       Delete probe from list
       Setstatus(reserve)
    }//end if
  }//end for
}// end function
```

ตัวอย่างการทำงานของอัลกอริทึมการเลือกเส้นทางในเครือข่ายอินเทอร์เน็ตเพื่อทำให้เกิด QoS แสดงดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



R1, R2, …, R5 คือเราเตอร์ตัวที่ 1 ถึงเราเตอร์ตัวที่ 5

——→ เส้นทางที่แพ็กเก็ต Probe ถึงก่อน - - - - - - - - → เส้นทางที่แพ็กเก็ต Probe ถึงทีหลัง

- - - - - - → เส้นทางที่จะปล่อยทรัพยากร - - - - - - - - → เส้นทางที่ใช้ในการส่งข้อมูลจริง

รูปที่ 3.2 การทำงานของอัลกอริทึมการเลือกเส้นทางในเครือข่ายอินเทอร์เน็ตเพื่อทำให้เกิด QoS

จากรูปที่ 3.2 มีขั้นตอนการทำงานดังต่อไปนี้

ขั้นที่ 1 Probing Phase

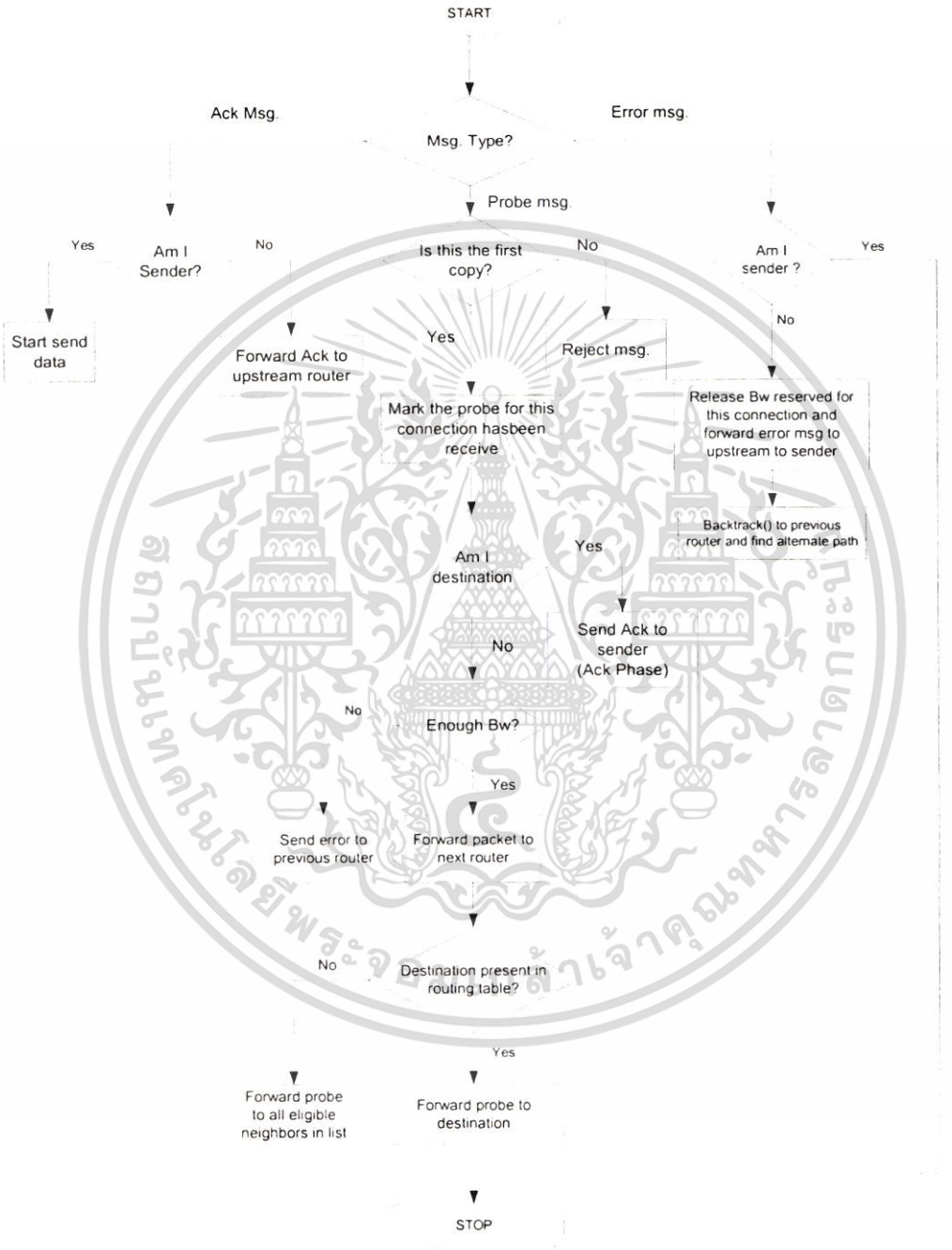
1. ผู้ส่ง (source) จะใช้วิธี Flooding ส่งแพ็กเก็ต Probe ไปยัง R1 และ R2
2. R1 และ R2 ตรวจสอบว่าเคยได้รับแพ็กเก็ต Probe นี้หรือไม่ ถ้าไม่เคยได้รับจะทำการบันทึกข้อมูลไว้ และทำการตรวจสอบปลายทางว่าอยู่ในตารางเส้นทางของตนหรือไม่ ถ้าอยู่จะส่งไปตามเส้นทางที่ระบุไว้ทันที ถ้าไม่อยู่จะส่งแพ็กเก็ต Probe ต่อไป
3. R3 ได้รับแพ็กเก็ต Probe มาจาก R1 และ R2 ในกรณีนี้ให้แพ็กเก็ต Probe จาก R1 ถึงก่อน R2 ดังนั้นที่ R3 นี้จะทิ้งแพ็กเก็ต Probe ของ R2 ไป (ถ้า R2 ถึงก่อนจะทิ้ง Probe ของ R1)
4. หลังจากที่ R3 ทิ้งแพ็กเก็ต Probe ของ R2 ไปจะมีการส่ง Error Message ไปบอก R2 เพื่อให้ R2 ปล่อยทรัพยากรที่จองไว้สำหรับผู้ส่ง (Source) โดยการเรียกใช้ฟังก์ชั่น Release(Probe_pkt) ดังนั้นทรัพยากรจะถูกปล่อยทันทีเพื่อให้ผู้อื่นสามารถใช้ต่อไปได้
5. เมื่อ R3 ได้รับแพ็กเก็ต Probe จาก R1 ซึ่งถือเป็นแพ็กเก็ตที่ชนะจะทำการตรวจสอบว่าปลายทางที่จะส่งไปอยู่ในตารางเส้นทางของตนหรือไม่ ในภาพตัวอย่างจะเห็นว่าปลายทางนั้นเป็นเราเตอร์ในระดับที่ 2 (second-degree neighbor) ซึ่งจะมีข้อมูลอยู่ในตารางเส้นทาง จึงสามารถส่งแพ็กเก็ต Probe ไปยังปลายทางได้ทันที โดยไม่ต้องใช้วิธีการ Flooding ในที่นี้เส้นทางด้านล่างมีทรัพยากรเพียงพอกับความต้องการ จึงส่ง Probe ไปตามเส้นทาง R5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2 Ack Phase

6. เมื่อปลายทางได้รับแพ็กเก็ต Probe จะทำการตอบกลับเพื่อเป็นการยืนยันไปยังผู้ส่งตามเส้นทางที่ระบุในแพ็กเก็ต Probe

จากขั้นตอนทั้ง 6 ขั้นตอนดังที่อธิบายไว้ด้านบน สามารถนำมาเขียนเป็นแผนผังการทำงาน (Flow Chart) ได้ดังรูปที่ 3.3



รูปที่ 3.3 อัลกอริทึมการรับและส่งแพ็กเก็ตของเราเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 แผนภาพแสดงการเปลี่ยนสถานะของส่วนประกอบในเครือข่าย (State Transition Diagram)

แผนภาพแสดงการเปลี่ยนสถานะของส่วนประกอบในเครือข่าย แสดงถึงสถานะของส่วนประกอบนั้นว่าประกอบไปด้วยสถานะใดบ้าง และเป็นสถานะที่เกิดขึ้นในแบบจำลอง

3.4.1 แผนภาพการเปลี่ยนสถานะของแพ็กเก็ต

รูปที่ 3.4 แสดงแผนภาพการเปลี่ยนสถานะของแพ็กเก็ตทุกประเภท

เมื่อแพ็กเก็ตถูกสร้างขึ้นมาจะอยู่ในสถานะ Arrival

- เมื่อถูกส่งออกไปจะอยู่ในสถานะ (Departure)

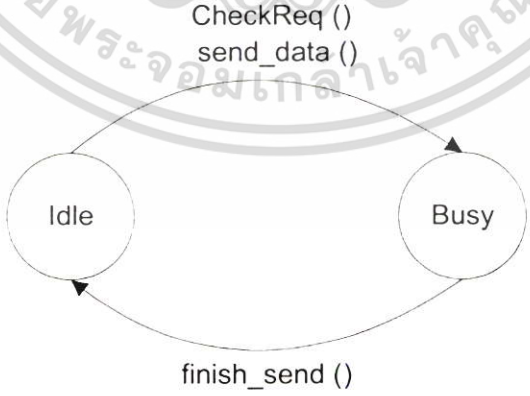


รูปที่ 3.4 แผนภาพการเปลี่ยนสถานะของแพ็กเก็ต

3.4.2 แผนภาพการเปลี่ยนสถานะของเส้นทางเชื่อมโยง

รูปที่ 3.5 แสดงแผนภาพการเปลี่ยนสถานะของเส้นทางเชื่อมโยง โดยแบ่งออกเป็น 2 สถานะ คือ

- Idle หมายถึงเส้นทางเชื่อมโยงนั้นว่าง
- เมื่อเส้นทางเชื่อมโยงใช้ในการส่งข้อมูลจะเปลี่ยนเป็นสถานะ Busy
- เมื่อส่งข้อมูลเสร็จจะเปลี่ยนกลับไปสู่สถานะ Idle เช่นเดิม



รูปที่ 3.5 แผนภาพการเปลี่ยนสถานะของเส้นทางเชื่อมโยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 แผนภาพการเปลี่ยนสถานะของเครื่องเวิร์คสเตชั่น

เครื่องเวิร์คสเตชั่น (Work Station) หมายถึงเครื่องคอมพิวเตอร์ปลายทางที่ทำการส่ง หรือรับข้อมูล แผนภาพการเปลี่ยนสถานะของเครื่องเวิร์คสเตชั่นแสดงดังรูปที่ 3.6

- เริ่มแรกเครื่องเวิร์คสเตชั่นจะอยู่ในสถานะว่าง (Idle)
- เมื่อต้องการส่งข้อมูลจะทำการส่งการร้องขอออกไป และเปลี่ยนไปอยู่ในสถานะการรอคอยการตอบกลับมาจากผู้รับ (Wait for Ack)
- เมื่อได้รับการตอบกลับมาแสดงว่ามีการจองทรัพยากรเรียบร้อยแล้ว จึงเปลี่ยนไปอยู่ในสถานะที่จะส่งข้อมูล (Data Transmission)
- เมื่อทำการส่งข้อมูลเรียบร้อยแล้วจะกลับไปอยู่ในสถานะว่างเช่นเดิม
- หากได้รับแพ็กเก็ตความผิดพลาด (Error Packet) จะกลับไปอยู่ในสถานะว่างต่อไป
- กรณีต่อไปเมื่อมีการร้องขอที่จะส่งข้อมูลมาจากเครื่องเวิร์คสเตชั่นอื่น ๆ เมื่อได้รับการร้องขอ เครื่องผู้รับจะตรวจสอบข้อมูลการร้องขอและทำการตอบกลับไป (Ack) และเข้าสู่สถานะการรอรับข้อมูล



รูปที่ 3.6 แผนภาพการเปลี่ยนสถานะของเครื่องเวิร์คสเตชั่น (Station)

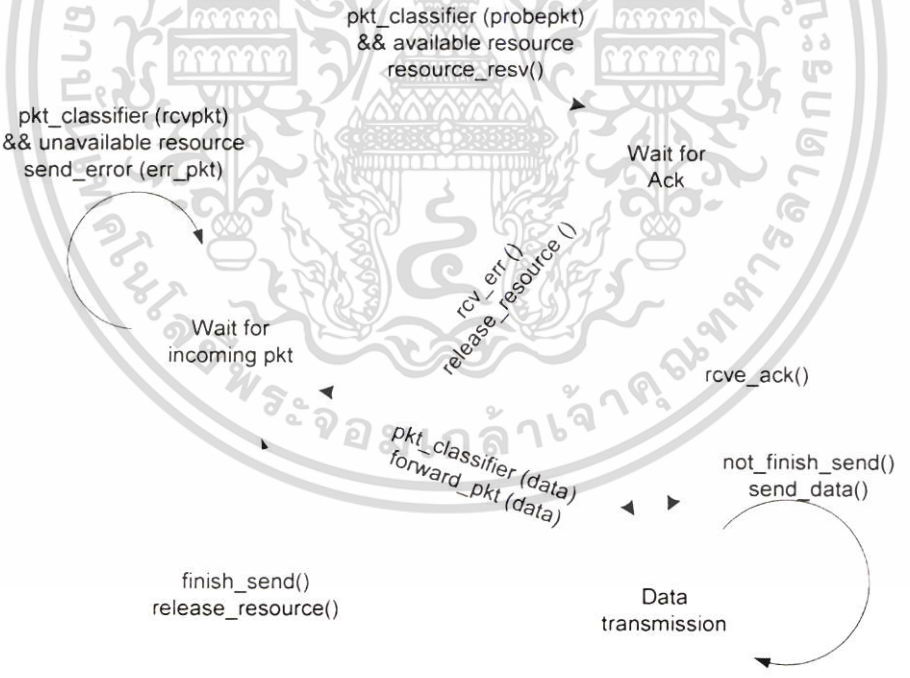
3.4.5 แสดงแผนภาพการเปลี่ยนสถานะของเราเตอร์

การเปลี่ยนสถาะของเราเตอร์สามารถอธิบายได้ดังรูปที่ 3.7

- โดยเริ่มแรกเราเตอร์จะรอรับแพกเก็ตข้อมูลที่เข้ามา (Wait for incoming packet)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อมีแพ็กเก็ตเข้ามาและทำการตรวจสอบว่าเป็นแพ็กเก็ต Probe จะทำการตรวจสอบทรัพยากร ถ้าพบว่ามีทรัพยากรเพียงพอ จะทำการจองทรัพยากรโดยใช้ฟังก์ชั่น resv_sesource(bw_requirement)
- เราเตอร์จะเข้าสู่สถานการณ์รอคอยการตอบกลับจากผู้รับ (Ack) หากได้รับแพ็กเก็ตความผิดพลาดจะทำการปล่อยทรัพยากรที่จองไว้นั้น หรือถ้าได้รับการตอบกลับ เราเตอร์จะเปลี่ยนไปอยู่ในสถานะการส่งข้อมูล (Data Transmission)
- และจะตรวจสอบว่าส่งข้อมูลหมดหรือยังโดยใช้ฟังก์ชั่น not_finish_data(data) หากยังส่งข้อมูลไม่เสร็จเราเตอร์จะยังอยู่ในสถานะนี้เหมือนเดิม และทำการส่งข้อมูลจนกว่าจะหมดโดยใช้ฟังก์ชั่น send_data(data)
- เมื่อตรวจสอบพบว่าส่งข้อมูลหมดเรียบร้อยแล้ว จะทำการปล่อยทรัพยากรที่จองไว้ หรือถ้าได้รับแพ็กเก็ตความผิดพลาดจะทำการปล่อยทรัพยากรที่จองไว้นั้นเช่นกัน และกลับไปอยู่สถานการณ์รอคอยรับแพ็กเก็ตเหมือนเดิม
- ในกรณีที่ได้รับแพ็กเก็ต Probe แต่ทำการจองทรัพยากรไม่ได้เนื่องมาจากทรัพยากรไม่เพียงพอ จะทำการส่งแพ็กเก็ตความผิดพลาดกลับไปให้กับเราเตอร์ก่อนหน้านี้ และกลับไปอยู่ในสถานะการรอรับแพ็กเก็ตเช่นเดิม



รูปที่ 3.7 แสดงแผนภาพการเปลี่ยนสถานะของเราเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากอัลกอริทึมดังกล่าวได้มีการปรับปรุงข้อเสียของอัลกอริทึมเดิมคือ

- สามารถรู้ได้ทันทีว่าเราเตอร์ในเส้นทางดังกล่าวนั้นมีทรัพยากรไม่เพียงพอเนื่องจากมีการตรวจสอบทรัพยากรทันที ไม่ต้องรอในขากลับเพราะฉะนั้นจะใช้เวลาน้อยกว่าในการตรวจสอบข้อผิดพลาด
- อัตราการเกิด Call Blocking มีน้อยกว่าเนื่องจากมีการจองทรัพยากรไว้ตั้งแต่แรกในช่วงเวลา Ack Phase โอกาสจะถูกปฏิเสธการจองแบนด์วิดท์ทีหลังจะไม่มี โอกาสที่เกิดในการล้มเหลวในการเชื่อมต่อจึงต่ำ
- จำนวนแพ็กเก็ตที่เกิดจากการสร้างการเชื่อมต่อในกรณีที่เกิดการล้มเหลวจะมีจำนวนต่ำกว่าวิธีการเดิมทำให้ไม่เปลืองทรัพยากรของเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การประเมินประสิทธิภาพการค้นหาเส้นทาง

ขั้นตอนการประเมินประสิทธิภาพการค้นหาเส้นทางของเครือข่ายสามารถแบ่งได้เป็น 10 ขั้นตอน [17] ได้แก่

1. กำหนดเป้าหมายและขอบเขตของระบบ
2. กำหนดรายละเอียดการให้บริการของระบบและความคาดหวังที่จะให้เกิดขึ้น
3. กำหนดมาตรหรือเกณฑ์สำหรับการวัดประสิทธิภาพของระบบ
4. กำหนดรายละเอียดพารามิเตอร์ของระบบและภาระงาน (Workload) ที่เข้ามา
5. กำหนดพารามิเตอร์ที่สนใจศึกษาหรือแฟกเตอร์ (Factor) และค่าที่เป็นไปได้
6. กำหนดเทคนิคการประเมินประสิทธิภาพ
7. กำหนดภาระงานของระบบที่จะเกิดขึ้น
8. ออกแบบการทดลอง
9. วิเคราะห์และแปลผลการทดลอง
10. นำเสนอ

ใน 4 ขั้นตอนสุดท้ายซึ่งเป็นส่วนการทดลองอธิบายไว้ในบทต่อไป

4.1 การกำหนดเป้าหมายและขอบเขตของระบบ

เป้าหมายของงานวิจัยนี้คือ การประเมินประสิทธิภาพการค้นหาเส้นทางของอัลกอริทึมที่นำเสนอและผลกระทบที่เกิดขึ้นจากการใช้อัลกอริทึมดังกล่าวค้นหาเส้นทางในเครือข่าย โดยขอบเขตของระบบที่สนใจคือ เครือข่ายที่มีลักษณะเดียวกัน (Homogeneous Networks)

4.2 กำหนดรายละเอียดการให้บริการของระบบและความคาดหวังที่จะเกิดขึ้น

การให้บริการหรือหน้าที่หลักของเครือข่ายเป็นการส่งต่อแพ็กเก็ตข้อมูลจากเครือข่ายย่อยต้นทางไปยังเครือข่ายย่อยปลายทางที่ถูกต้อง แต่หน้าที่รองซึ่งเป็นหน้าที่จำเป็นที่ต้องมี คือ การค้นหาเส้นทางสำหรับการส่งต่อแพ็กเก็ตข้อมูลเหล่านั้น

ความต้องการหรือความคาดหวังที่เกิดขึ้นต่อหน้าที่ของเครือข่ายจึงแบ่งออกเป็นสองส่วนเช่นเดียวกับหน้าที่ของเครือข่าย คือ ส่วนแรกเป็นความคาดหวังของการส่งต่อแพ็กเก็ตข้อมูลไปยังเครือข่ายปลายทางซึ่งต้องสามารถส่งต่อแพ็กเก็ตไปยังเครือข่ายปลายทางได้อย่างรวดเร็วและรองรับปริมาณแพ็กเก็ตข้อมูลได้เป็นจำนวนมาก ส่วนที่สองเป็นความคาดหวังของการค้นหาเส้นทาง คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การค้นหาเส้นทางที่ดีสำหรับการส่งต่อแพ็กเก็ตข้อมูลและใช้ทรัพยากรเครือข่ายได้อย่างมีประสิทธิภาพ

4.3 กำหนดมาตรวัดประสิทธิภาพ

กำหนดมาตรวัดประสิทธิภาพมาจากความคาดหวังหรือความต้องการที่อยากได้จากเครือข่าย นั่นคือ ความรวดเร็วของการส่งต่อแพ็กเก็ตข้อมูลซึ่งใช้มาตรวัดเป็น ความหน่วงเวลาของแพ็กเก็ต (Data Packet Transfer Delay) ทรูพุทของเครือข่าย (Network Throughput) การใช้ทรัพยากรเครือข่ายของการค้นหาเส้นทางใช้มาตรวัดเป็นเปอร์เซ็นต์ของโอเวอร์เฮดของการค้นหาเส้นทางที่เกิดขึ้น (Percent of Routing Overhead) การค้นหาเส้นทางที่ดีจะส่งผลต่อเครือข่ายทำให้มีประสิทธิภาพที่ดี มาตรวัดสำหรับการค้นหาเส้นทางเพื่อทำให้เกิดการประกันการส่งข้อมูล (QoS) คือการหาเส้นทางที่รองรับกับความต้องการของผู้ส่งได้ และใช้เวลาในการหาเส้นทางไม่มาก ซึ่งรายละเอียดจะอธิบายไว้ในบทที่ 6

4.4 กำหนดรายละเอียดของพารามิเตอร์ของระบบและภาระงาน

พารามิเตอร์ในแบบจำลองแบ่งได้เป็น 2 ประเภท

พารามิเตอร์ที่มีผลต่อประสิทธิภาพของเครือข่าย (Network Parameters) ได้แก่

- โทโปโลยีเครือข่าย
- คุณสมบัติของโหนด ได้แก่ ขนาดของบัฟเฟอร์แพ็กเก็ตข้อมูล
- คุณสมบัติของเส้นทางเชื่อมโยง (Link) ได้แก่ อัตราความเร็วของผู้ใช้ (User Data Rate) และระยะทางของเส้นทางเชื่อม

พารามิเตอร์ที่มีผลต่อประสิทธิภาพการค้นหาเส้นทาง (Workload Parameters)

- อัลกอริทึมค้นหาเส้นทาง
- ค่าใช้จ่ายของเส้นทางที่เชื่อมโยงไปยังโหนดข้างเคียง
- จำนวนของผู้ขอใช้บริการ (Call Request)
- จำนวนแบนด์วิดท์ที่ผู้ขอใช้แต่ละคนต้องการ (Bandwidth Required by a Call)
- ภาระงานของระบบเครือข่ายเป็นอัตราการมาถึงของแพ็กเก็ตข้อมูล

4.5 กำหนดแฟกเตอร์และค่าที่เป็นไปได้

พารามิเตอร์ที่นำมาใช้เป็นแฟกเตอร์และค่าที่เป็นไปได้คือ

- อัลกอริทึมค้นหาเส้นทาง ได้แก่ อัลกอริทึมที่นำเสนอ (Proposed Algorithm) อัลกอริทึม [10] (Existing Algorithm) และอัลกอริทึม [9] (Flooding Algorithm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อัตราการมาถึงของแพ็กเก็ตข้อมูล (Workload) มีค่าเท่ากับ 10% 20% 30% 40% 50% 60% 70% 80% 90% และ 100% ของเส้นทางเชื่อมโยง
- จำนวนแบนด์วิดท์ที่ผู้ใช้แต่ละคนต้องการ มีค่าเท่ากับ 10% 20% 30% 40% 50% 60% 70% 80% 90% และ 100% ของเส้นทางเชื่อมโยง
- เวลาจำกัดที่ใช้ในการค้นหาเส้นทาง คือ 0.2 0.3 0.4 0.5 0.6 0.7 และ 0.8 วินาที

4.6 เทคนิคการประเมินประสิทธิภาพ

เทคนิคที่ใช้ประเมินประสิทธิภาพใช้การจำลองแบบทางคอมพิวเตอร์ซึ่งรายละเอียดอธิบายไว้ในบทที่ 6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบการทดลอง

การจำลอง (Simulation) เป็นกระบวนการออกแบบแบบจำลอง (Model) โดยเลียนแบบเหตุการณ์หรือระบบที่สนใจซึ่งอาจจะมีอยู่จริงหรือไม่มีอยู่จริงก็ได้ แล้วนำแบบจำลองที่สร้างขึ้นตามที่ได้ออกแบบไปใช้ตามเป้าหมายที่วางไว้ การจำลองแบบเครือข่ายเป็นการจำลองแบบทางคอมพิวเตอร์ (Computer Simulation) โดยการออกแบบใช้แบบจำลองแบบเหตุการณ์ที่ไม่ต่อเนื่อง (Discrete Event Simulation Model) [16] และวิธีการเชิงโพรเซส (Process Oriented) [16] วิเคราะห์ลำดับเหตุการณ์และกระบวนการทำงานที่เกิดขึ้นภายในเครือข่าย ตัวแบบจำลองในงานวิจัยนี้พัฒนาด้วยภาษา MODSIM III [17] ซึ่งเป็นภาษาสำหรับการจำลองทั่วไป (General Simulation Language) ของบริษัท CACI ภาษา MODSIM III สนับสนุนวิธีการเชิงวัตถุ (Object Oriented) ทำให้โครงสร้างของตัวแบบจำลองแบ่งออกเป็นวัตถุซึ่งมีความใกล้เคียงกับโลกความจริง (Real World) ผลที่ตามมาคือ การตรวจสอบความถูกต้องต่างๆ สามารถกระทำได้ง่าย ดังนั้น การทำงานของตัวแบบจำลองที่สร้างขึ้นจึงมีความถูกต้องและความน่าเชื่อถือของผลการจำลองแบบจากตัวแบบจำลอง

5.1 ส่วนประกอบแบบจำลองเครือข่าย (Network Components)

การออกแบบจำลองเครือข่าย (Network Model) ในส่วนที่เกี่ยวข้องกับการค้นหาเส้นทางสามารถแบ่งเครือข่ายออกเป็น 4 องค์ประกอบ ได้แก่ โหนด (Node) เส้นทางเชื่อมโยง (Link) อัลกอริทึมค้นหาเส้นทาง (Routing Algorithm) และแพ็กเก็ต (Packet) การแบ่งออกเป็นองค์ประกอบทำให้ง่ายต่อการวิเคราะห์ ออกแบบ ตรวจสอบความถูกต้องและการพัฒนาตัวแบบจำลองโดยแต่ละองค์ประกอบจะมีคุณลักษณะ สถานะ และหน้าที่ที่แตกต่างกันและอิสระจากกัน แต่มีการทำงานที่สัมพันธ์กัน

5.1.1 โหนด (Node)

โหนดใช้แทนอุปกรณ์ค้นหาเส้นทางในเครือข่ายโดยเชื่อมต่อกับอัลกอริทึมค้นหาเส้นทางและเส้นทางเชื่อมโยง แต่ละโหนดจะต้องมีเส้นทางเชื่อมโยงที่เชื่อมโยงไปยังโหนดอื่นๆ ในเครือข่ายอย่างน้อยสองโหนดและมีเส้นทางเดินที่สามารถเดินทางไปยังทุกโหนดในเครือข่ายอย่างน้อยโหนดละหนึ่งเส้นทาง โหนดจะทำการตรวจสอบทุกแพ็กเก็ตที่มาถึง ถ้าเป็นแพ็กเก็ตข้อมูลจะส่งไปพักไว้ในบัฟเฟอร์ข้อมูล แต่ถ้าเป็นแพ็กเก็ตเส้นทางจะนำข้อมูลเส้นทางภายในส่งต่อให้อัลกอริทึมค้นหาเส้นทาง และอัลกอริทึมค้นหาเส้นทางจะส่งข้อมูลเส้นทางไปยังโหนดข้างเคียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึมจะนำข้อมูลเส้นทางนั้นส่งมาให้โหนดเพื่อบรรจุลงในแพ็กเก็ตเส้นทางแล้วส่งไปพักไว้ในบัฟเฟอร์เส้นทาง เพื่อรอคอยการส่งต่อไปยังโหนดข้างเคียง

เมื่อแพ็กเก็ตมาถึงบัฟเฟอร์ ตัวจัดการบัฟเฟอร์จะตรวจสอบเส้นทางเชื่อมโยงที่จะส่งต่อแพ็กเก็ตนั้นออกไป ยกเว้นในกรณีแพ็กเก็ตข้อมูลที่มาถึงระหว่างการปรับปรุงเส้นทาง จากนั้นจะทำการตรวจสอบไปยังเส้นทางเชื่อมโยงนั้นว่าอยู่ระหว่างการโอนย้ายข้อมูลหรือไม่ ถ้าไม่จะต้องส่งคำสั่งให้เส้นทางเชื่อมโยงเริ่มต้นการโอนย้ายเฟรมข้อมูล เมื่อเส้นทางเชื่อมโยงเริ่มต้นการโอนย้ายเฟรมข้อมูลหรือโอนย้ายเฟรมข้อมูลก่อนหน้าเสร็จเรียบร้อยจะส่งคำสั่งมายังตัวจัดการบัฟเฟอร์เพื่อให้ส่งแพ็กเก็ตที่ต้องการจะโอนย้ายต่อไปมาให้ เมื่อตัวจัดการบัฟเฟอร์ได้รับคำสั่งจะทำการค้นหาแพ็กเก็ตที่ต้องการส่งต่อไปยังเส้นทางเชื่อมโยงนั้น ในกรณีที่ค้นเจอแพ็กเก็ตข้อมูลจะต้องไม่อยู่ระหว่างการปรับปรุงเส้นทาง จากนั้นจะส่งแพ็กเก็ตไปให้เส้นทางเชื่อมโยงในรูปของเฟรมข้อมูล แต่ถ้าไม่มีหรือมีแพ็กเก็ตข้อมูลแต่อยู่ระหว่างการปรับปรุงตารางเส้นทาง ตัวจัดการบัฟเฟอร์จะส่งค่าว่างไปให้เส้นทางเชื่อมโยงเป็นการบอกว่าไม่มีแพ็กเก็ตที่จะส่งออกไปในขณะนั้น และเมื่อเส้นทางเชื่อมโยงได้รับค่าว่างจะหยุดการโอนย้ายเฟรมข้อมูล

การส่งต่อแพ็กเก็ตของโหนดจะให้ความสำคัญกับแพ็กเก็ตเส้นทางก่อนเสมอ คือ ถ้าเส้นทางเชื่อมโยงว่างและมีทั้งแพ็กเก็ตข้อมูลรอคอยอยู่ ถึงแม้ว่าแพ็กเก็ตข้อมูลจะมาถึงก่อน แต่แพ็กเก็ตเส้นทางจะต้องถูกส่งออกไปก่อนเสมอ ทุกโหนดในแบบจำลองเครือข่ายมีคุณสมบัติและการทำงานเหมือนกัน มีอิสระจากกันในการทำงาน และมีความสำคัญเท่ากัน แต่ละโหนดไม่สามารถตรวจสอบการมีอยู่ (Existing) ของโหนดข้างเคียงได้ด้วยฮาร์ดแวร์ ในกรณีที่บัฟเฟอร์ของแพ็กเก็ตข้อมูลของโหนดมีขนาดจำกัด โหนดจะจัดการบัฟเฟอร์นั้นด้วยอัลกอริทึมถังน้ำรั่ว (Leaky Bucket Algorithm) ซึ่งในขณะที่บัฟเฟอร์เต็มแพ็กเก็ตที่มาถึงจะถูกทิ้งไป พารามิเตอร์ที่สำคัญของโหนดได้แก่ ขนาดสูงสุดของบัฟเฟอร์ข้อมูลซึ่งมีค่าได้ตั้งแต่หนึ่งจนถึงอนันต์ (Infinity)

5.1.2 เครือข่าย (Network)

เครือข่ายย่อยในแบบจำลองเครือข่ายนี้แบ่งออกเป็นเครือข่ายย่อยปลายสุด (Stub Subnet) และเครือข่ายย่อยทางผ่าน (Transit Subnet) เครือข่ายย่อยปลายสุดเป็นจุดเริ่มต้นและจุดสิ้นสุดของแพ็กเก็ตข้อมูลโดยเชื่อมต่อกับเส้นทางเชื่อมโยงเพียงเส้นทางเดียวไปยังโหนดใดโหนดหนึ่งเท่านั้น เครือข่ายย่อยปลายสุดจึงถูกกำหนดให้เป็นเครือข่ายปลายทางสำหรับการค้นหาเส้นทางภายในเครือข่าย นอกจากนี้ เครือข่ายย่อยมีหน้าที่สร้างแพ็กเก็ตข้อมูลโดยแต่ละแพ็กเก็ตมีขนาดเท่ากับ 512 ไบต์ โดยโอกาสการเกิดขึ้นมีการกระจายแบบปัวซอง (Poisson Distribution) ส่วนการกำหนดปลายทางของแพ็กเก็ตข้อมูลใช้การกระจายแบบนอร์มอล (Normal Distribution) จำนวนแพ็กเก็ตข้อมูลที่สร้างขึ้นเฉลี่ยในแต่ละวินาทีสัมพันธ์กับอัตราความเร็วข้อมูลของผู้ใช้ (User Data Rate)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราความเร็วของผู้ใช้เป็นแบนด์วิดท์ (Bandwidth) ที่ผู้ใช้ได้ซึ่งจะน้อยกว่าแบนด์วิดท์ของเส้นทางเชื่อมโยง

ตัวแบบจำลองของการจำลองแบบเครือข่ายในครั้งนี้จะประยุกต์ใช้พารามิเตอร์การควบคุมการเชื่อมโยงทางตรรกะเพียงการเชื่อมโยงแบบจุดต่อจุดและใช้การสื่อสารแบบสองทาง (Full Duplex) เท่านั้น

5.1.3 เส้นทางเชื่อมโยง (Link)

เส้นทางเชื่อมโยงเป็นเส้นทางที่เชื่อมระหว่างโหนดกับเครือข่ายย่อย โดยถ้าเชื่อมโยงกับเครือข่ายย่อยปลายทางจะทำหน้าที่โอนย้ายข้อมูลระหว่างโหนดกับเครือข่ายย่อยปลายทางส่วนถ้าเชื่อมโยงกับเครือข่ายย่อยทางผ่านจะทำหน้าที่เป็นคิวส่ง (Transmission Queue) และคิวรับแต่เนื่องจากสมมุติฐานที่ว่าการโอนย้ายมีความถูกต้องเสมอจึงไม่ทำให้เกิดการส่งซ้ำของเฟรมที่เกิดข้อผิดพลาด ดังนั้นคิวส่งและคิวรับมีขนาดเท่ากับหนึ่งเสมอ

หน้าที่อีกอย่างหนึ่งของเส้นทางเชื่อมโยง คือถ้าโพรโทคอลดาต้าลิงค์กำหนดให้มีเฟรมตอบรับ เส้นทางเชื่อมโยงต้องสร้างภาระงานที่เกิดจากเฟรมตอบรับด้วย การส่งเฟรมตอบรับในบางโพรโทคอลด้าต้าลิงค์มีทั้งรวมการส่งเฟรมตอบรับรวมอยู่ในเฟรมข้อมูลและการส่งเฟรมตอบรับเฉพาะ ในกรณีดีที่สุด คือการส่งเฟรมตอบรับรวมอยู่ในเฟรมข้อมูล แต่ในกรณีที่แย่ที่สุดคือ การส่งเฉพาะเฟรมตอบรับ ในแบบจำลองเครือข่ายจะใช้การส่งเฟรมตอบรับในกรณีที่แย่ที่สุดหรือมีภาระงานที่เกิดเฟรมตอบรับมากที่สุด เฟรมตอบรับมีขนาดเล็กมากเมื่อเปรียบเทียบกับเฟรมข้อมูลทั่วไป จึงใช้เวลาการโอนย้ายน้อยกว่ามากและมีผลต่อความความหน่วงเวลาของแพ็กเก็ตข้อมูลน้อยมาก การส่งเฟรมตอบรับที่รวดเร็วจะทำให้จำนวนเฟรมที่รอคอยในหน้าต่าง (Window) ของอีกฝั่งหนึ่งลดลง ดังนั้น เฟรมตอบรับจึงถูกส่งออกไปทันทีเมื่อเส้นทางเชื่อมโยงว่างหรือถ้าอยู่ระหว่างโอนย้ายเฟรมข้อมูลจะทำการส่งหลังจากโอนย้ายเฟรมนั้นเสร็จ

5.1.4 อัลกอริทึมค้นหาเส้นทาง (Routing Algorithm)

อัลกอริทึมค้นหาเส้นทางเป็นตัวจักรสำคัญของการค้นหาเส้นทางโดยมีหน้าที่ค้นหาคำนวณเส้นทางเดินที่สั้นที่สุด และกำหนดเส้นทางสำหรับการส่งต่อแพ็กเก็ตข้อมูลของโหนดไปยังเครือข่ายย่อยปลายทางโดยข้อมูลที่สำคัญ ได้แก่ ที่อยู่ของเครือข่ายปลายทาง เส้นทางเชื่อมโยงและโหนดถัดไป (Next Node) ที่อยู่ของเครือข่ายปลายทางใช้สำหรับจับคู่กับเครือข่ายปลายทางภายในแพ็กเก็ตแล้วจะได้เส้นทางเชื่อมโยงและโหนดถัดไปซึ่งอยู่ในเส้นทางเดินที่สั้นที่สุด ที่สามารถเดินทางไปยังเครือข่ายปลายทางนั้นเป็นข้อมูลสำหรับการส่งต่อแพ็กเก็ตออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.5 แพ็กเก็ต (Packet)

แพ็กเก็ตที่เกิดขึ้นในแบบจำลองเครือข่ายแบ่งออกเป็นแพ็กเก็ตเส้นทางและแพ็กเก็ตข้อมูล แพ็กเก็ตเส้นทางใช้สำหรับบรรจุข้อมูลเส้นทางเพื่อแลกเปลี่ยนกันระหว่างโหนด การเกิดขึ้นของแพ็กเก็ตเส้นทางขึ้นอยู่กับการค้นหาเส้นทาง ส่วนแพ็กเก็ตข้อมูลเป็นการจำลองการแลกเปลี่ยนข้อมูลระหว่างกันของเครือข่ายย่อยปลายสุดและทำให้เกิดภาระงานในเส้นทางต่าง ๆ ที่แพ็กเก็ตข้อมูลเดินทางผ่าน แต่ละแพ็กเก็ตข้อมูลมีขนาดเท่ากับ 512 ไบต์และการเกิดขึ้นอยู่กับพารามิเตอร์ของเครือข่ายย่อยปลายสุด

5.2 ส่วนประกอบของความหน่วงเวลา (Delay Components)

ความหน่วงเวลาแบ่งเป็น 4 ประเภท [19]

- ความหน่วงเวลาจากการตรวจสอบส่วนหัวของแพ็กเก็ต (Nodal Processing Delay)
- ความหน่วงเวลาที่เกิดจากคิว (Queuing Delay)
- ความหน่วงเวลาที่เกิดจากเราเตอร์ (Transmission Delay)
- ความหน่วงเวลาที่เกิดจากการส่งข้อมูล (Propagation Delay)

5.2.1 ความหน่วงเวลาจากการตรวจสอบส่วนหัวของแพ็กเก็ต (Nodal Processing Delay)

ความหน่วงเวลาประเภทนี้ เกิดจากเวลาที่ใช้ในการตรวจสอบ ส่วนหัว (Header) และหาเส้นทางของแพ็กเก็ตว่าจะไปเส้นทางไหน และรวมถึงเวลาที่ใช้ในการตรวจสอบ Bit-Level Error ในแพ็กเก็ตด้วย

5.2.2 ความหน่วงเวลาที่เกิดจากคิว (Queuing Delay)

ความหน่วงเวลาประเภทนี้ เกิดจากเวลาที่แพ็กเก็ตอยู่ในคิว เพื่อรอจะถูกส่งไปในเครือข่าย ถ้าคิวว่าง นั่นคือไม่มีแพ็กเก็ตรออยู่ในคิว เพราะฉะนั้นความหน่วงประเภทนี้จะเท่ากับ 0

5.2.3 ความหน่วงเวลาที่เกิดจากการส่งข้อมูล (Transmission Delay)

ความหน่วงเวลาประเภทนี้ เกิดจากเวลาที่บิตแรกมาถึงเราเตอร์จึงถึงเวลาที่บิตสุดท้ายมาถึงเราเตอร์นั้น

$$Transmission\ Delay = \frac{Length\ of\ Packet}{Transmission\ Rate\ of\ Link} \qquad (5.1)$$

5.2.4 ความหน่วงเวลาที่เกิดจากการแพร่กระจายข้อมูล (Propagation Delay)

ความหน่วงเวลาประเภทนี้ เกิดจากเวลาที่ใช้ส่งหรือถ่ายทอดข้อมูลไปยัง ปลายทาง

$$Propagation\ Delay = \frac{Distance}{Speed} \qquad (5.2)$$

ปกติ Speed = $2 * 10^8$ m/s [19]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในงานวิจัยนี้จะคิดความหน่วงเวลาที่เกิดจากความหน่วงเวลาที่เกิดจากคิว ความหน่วงเวลาที่เกิดจากการส่งข้อมูล ความหน่วงเวลาที่เกิดจากการแพร่กระจายข้อมูล และไม่คิดความหน่วงเวลาที่เกิดจากการตรวจสอบส่วนหัวของแพ็กเก็ต เนื่องจากมีค่าน้อยมากเมื่อเทียบกับความหน่วงเวลาประเภทอื่น สามารถตัดทิ้งได้

5.3 การสร้างแบบจำลอง (Simulation Modeling)

ตามที่ได้กล่าวไปแล้วว่าการสร้างแบบจำลองนี้พัฒนาขึ้นด้วยภาษา MODSIM III ซึ่งเป็นภาษาสำหรับสร้างแบบจำลองโดยตรง และเขียนในเชิงวัตถุหรือออบเจ็กต์ โดยจะมีการสร้างออบเจ็กต์ต่างๆ ดังนี้คือ

ออบเจ็กต์เราเตอร์ (Router Object)

ออบเจ็กต์เวิร์คสเตชั่น (Work Station Object)

ออบเจ็กต์เส้นทางเชื่อมโยง (Link Object)

ออบเจ็กต์แพ็กเก็ต (Packet Object)

ซึ่งแต่ละออบเจ็กต์จะมีการเปลี่ยนสถานะเป็นอะไรบ้าง ด้วยเงื่อนไขอะไร ได้อธิบายรายละเอียดแล้วในหัวข้อ 3.4 โดยหัวข้อนี้จะกล่าวถึงรายละเอียดของแต่ละออบเจ็กต์ว่ามีส่วนประกอบอะไรบ้าง

5.3.1 ออบเจ็กต์เราเตอร์ (Router Object)

ออบเจ็กต์เราเตอร์ประกอบด้วยฟังก์ชั่นต่างๆ ดังต่อไปนี้

- ResourceMnge() ฟังก์ชั่นนี้ใช้สำหรับตรวจสอบแบนด์วิดท์ของเส้นทางที่มีการร้องขอทรัพยากรว่ามีเพียงพอหรือไม่ ถ้ามีเพียงพอจะทำการจองทรัพยากรโดยใช้ฟังก์ชั่น Rsvf() พร้อมทั้งคำนวณว่ามีทรัยากรเหลือจากการจอง หรือการปล่อยทรัพยากรที่ใช้เสร็จแล้วว่าเหลืออยู่เท่าไหร่
- Rsvf() ฟังก์ชั่นนี้ใช้สำหรับจองทรัพยากร
- ReleaseRscre() ฟังก์ชั่นสำหรับปล่อยทรัพยากรที่ใช้เสร็จเรียบร้อยแล้ว
- Ack() ฟังก์ชั่นสำหรับส่งแพ็กเก็ตตอบรับ หรือแพ็กเก็ตความผิดพลาด
- FindNextHop() ฟังก์ชั่นสำหรับหาว่าจะส่งข้อมูลไปเราเตอร์ที่อยู่ถัดไปจะส่งออกไปเส้นทางใด
- ServePacket() ฟังก์ชั่นสำหรับคอยตรวจสอบว่ามีแพ็กเก็ตข้อมูลมาถึงเราเตอร์เป็นแพ็กเก็ตประเภทใด และทำการแบ่งแยกแพ็กเก็ต พร้อมทั้งทิ้งแพ็กเก็ต Probe ที่ซ้ำหรือมาจากผู้ส่งคนเดียวกันนั่นเอง
- ObjInit() ฟังก์ชั่นสำหรับการกำหนดคุณสมบัติเริ่มต้นให้กับเราเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Advertisement() ฟังก์ชั่นสำหรับการแลกเปลี่ยนข้อมูลระหว่างเราเตอร์และทำการอัพเดตข้อมูลของตารางเส้นทาง (Routing Table)

เมื่อเริ่มรันโปรแกรมนั้นออบเจ็กต์เราเตอร์จะถูกกำหนดว่าจะมีเส้นทางเชื่อมโยงติดต่อกับโหนดไหนบ้าง โดยทุกเราเตอร์จะเชื่อมต่อกันเป็นลักษณะของโทโปโลยี NSFNET-T1

5.3.2 ออบเจ็กต์เวิร์คสเตชั่น (Work Station Object)

ออบเจ็กต์สเตชั่นนี้เปรียบเสมือนผู้ใช้ที่มีความต้องการรับและส่งข้อมูลประเภทมัลติมีเดียนั่นเอง โดยประกอบด้วยฟังก์ชั่นต่างๆ ดังต่อไปนี้

- GenReq() ฟังก์ชั่นสำหรับสร้างและส่งคำร้องขอหรือแพ็กเก็ต Probe ออกไป
- RcvePkt() ฟังก์ชั่นสำหรับรับแพ็กเก็ตที่เข้ามาว่าเป็นแพ็กเก็ต Probe เพื่อร้องขอสร้างการเชื่อมต่อหรือเป็นแพ็กเก็ตบอกความผิดพลาด (Error Packet)
- SendData() ฟังก์ชั่นสำหรับส่งข้อมูลเมื่อได้รับการยืนยันว่าจองทรัพยากรได้แ
- ObjInit() ฟังก์ชั่นสำหรับการกำหนดคุณสมบัติเริ่มต้นให้กับเวิร์คสเตชั่น เช่น หมายเลขประจำของแต่ละเวิร์คสเตชั่น (Work Station ID)

5.3.3 ออบเจ็กต์เส้นทางเชื่อมโยง (Link Object)

ออบเจ็กต์เส้นทางเชื่อมโยงประกอบด้วยฟังก์ชั่นต่างๆ ดังต่อไปนี้

- CheckReq() ฟังก์ชั่นสำหรับตรวจสอบว่ามีการร้องขอเพื่อใช้เป็นเส้นทางในการส่งข้อมูลหรือไม่
- ObjInit() ฟังก์ชั่นสำหรับการกำหนดคุณสมบัติเริ่มต้นให้กับเส้นทางเชื่อโยง เช่น เป็นเส้นทางที่เชื่อมต่อระหว่างเราเตอร์ตัวใด และมีแบนด์วิดท์เริ่มต้น เป็นต้น

5.3.4 ออบเจ็กต์แพ็กเก็ต (Packet Object)

ออบเจ็กต์แพ็กเก็ตประกอบด้วยฟังก์ชั่นต่างๆ ดังต่อไปนี้

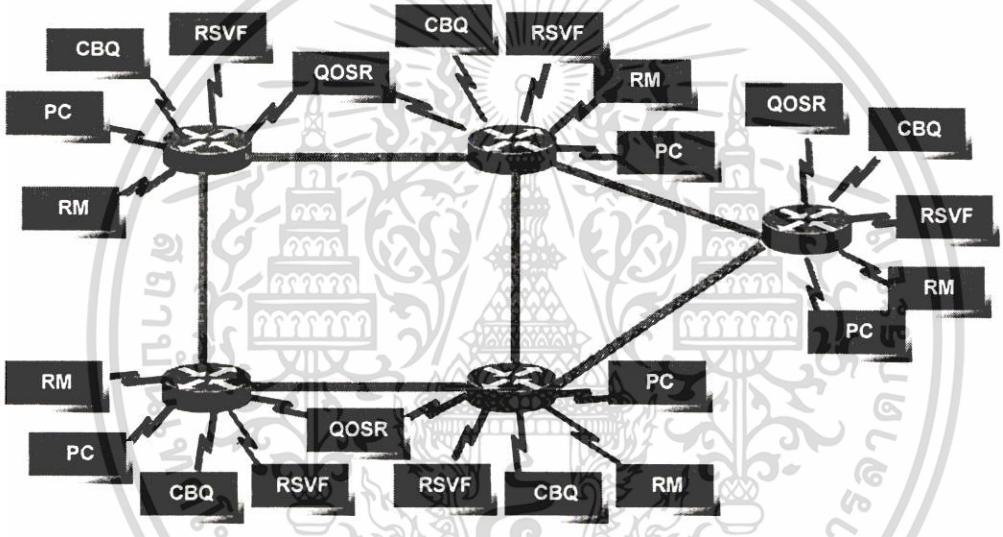
GenPacket() ฟังก์ชั่นในการหาจำนวนแพ็กเก็ตที่เกิดจากไฟล์ ซึ่งขนาดของไฟล์ได้มาจากการสุ่มแบบยูนิฟอร์ม หลังจากนั้นฟังก์ชั่นนี้จะคำนวนว่าขนาดไฟล์ดังกล่าวจะมีจำนวนแพ็กเก็ตเท่าไหร่ ซึ่งแต่ละแพ็กเก็ตจะมีขนาด 512 ไบต์ และเศษที่เหลือจากการคำนวณจะคิดเป็นหนึ่งแพ็กเก็ตเช่นกัน

การทำงานคือเมื่อเริ่มรันโปรแกรมออบเจ็กต์ทุกออบเจ็กต์จะถูกกำหนดคุณสมบัติเริ่มต้นต่างๆ โดยฟังก์ชั่น ObjInit() ของแต่ละออบเจ็กต์ ออบเจ็กต์เราเตอร์เริ่มมีการแลกเปลี่ยนข้อมูลกันและออบเจ็กต์แพ็กเก็ตถูกสร้างขึ้นมาและถูกส่งออกไปโดยออบเจ็กต์เวิร์คสเตชั่น เมื่อเราเตอร์ได้รับแพ็กเก็ตจะทำการตรวจสอบว่ามีการร้องขอทรัพยากรหรือไม่ หากมีจะทำการตรวจสอบทรัพยากรและจองทรัพยากรไว้ และส่งต่อแพ็กเก็ตไปยังเส้นทางที่เหมาะสมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ส่วนประกอบของแบบจำลอง (Simulation Components)

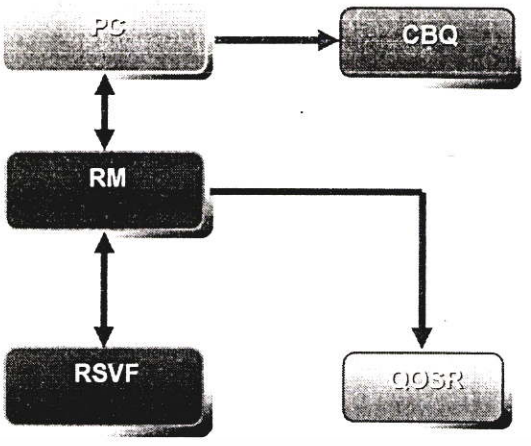
การออกแบบการทดลองในงานวิจัยนี้ ได้ทำการสร้างแบบจำลองเพื่อเลียนแบบการทำงานของเครือข่ายอินเทอร์เน็ต (Simulation) และมีการออกแบบส่วนประกอบ (Component) ที่เพิ่มเติมสำหรับเราเตอร์ตัวอย่างเช่น อัลกอริทึมการหาเส้นทาง QoS เรียกว่า "QOSR" เป็นฟังก์ชั่นในการจองทรัพยากร (Resource Reservation Function) เรียกว่า "RSVF"เป็นฟังก์ชั่นในการจัดการทรัพยากร (Resource Management) เรียกว่า "RM" เป็นฟังก์ชั่นสำหรับตรวจสอบทราฟฟิกที่เข้ามาว่าเป็นทราฟฟิกแบบใด (Packet Classifier) เรียกว่า "PC" เป็นฟังก์ชั่นสำหรับจัดตารางเวลาให้กับทราฟฟิกต่างๆ (Traffic Schedule) เรียกว่า "CBQ" ซึ่งใช้หลักการของ Class Based Queuing เนื่องจากเป็นอัลกอริทึมที่ใช้ในการจัดตารางเวลาให้กับทราฟฟิกได้ดีที่สุดอัลกอริทึมหนึ่ง [15] รูปที่ 5.1 แสดงตัวอย่างของเราเตอร์และส่วนประกอบต่าง ๆ



รูปที่ 5.1 ตัวอย่างส่วนประกอบต่าง ๆ ในเราเตอร์

สำหรับการติดต่อกันระหว่างแต่ละส่วนนั้นแสดงดังรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 การติดต่อระหว่างส่วนประกอบต่าง ๆ

ในแบบจำลองนี้เราได้กำหนดให้ลักษณะของทราฟฟิกในเครือข่ายมีทั้งหมด 3 ลักษณะ ซึ่งมีลำดับความสำคัญเรียงจากมากไปน้อยตามลำดับดังนี้

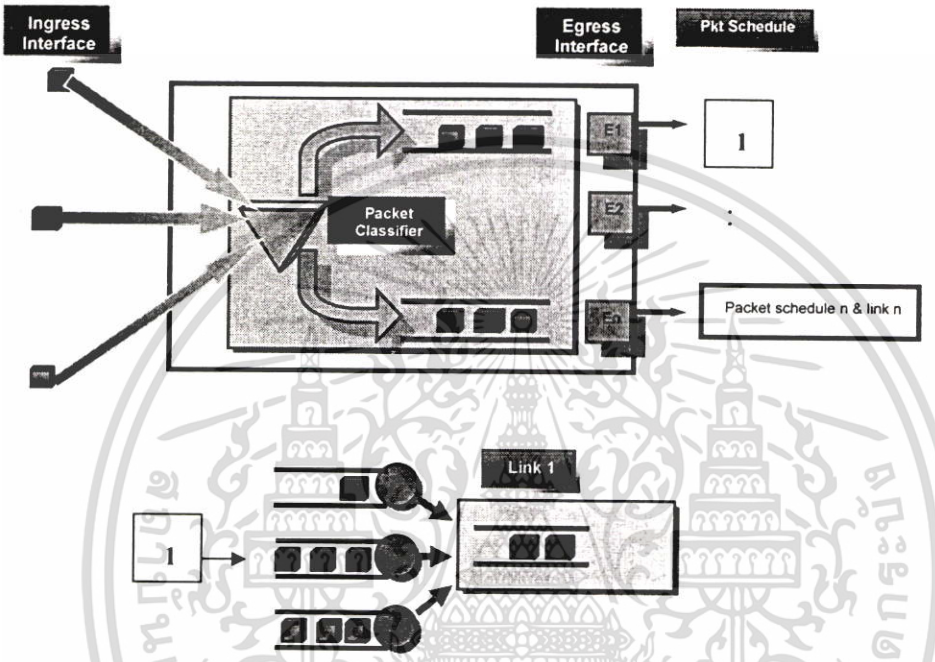
- Class A เป็นทราฟฟิกที่มีลำดับความสำคัญมากที่สุด ซึ่งได้แก่แพ็กเก็ต Probe ซึ่งใช้ในการร้องขอจองทรัพยากร เพื่อขอใช้บริการ QoS
- Class B เป็นทราฟฟิกที่มีความสำคัญเป็นอันดับสอง ได้แก่ทราฟฟิกที่ต้องการ QoS เช่น มัลติมีเดีย, การประชุมทางไกล เป็นต้น
- Class C เป็นทราฟฟิกที่มีความสำคัญน้อยสุดในเครือข่ายได้แก่ทราฟฟิกแบบ Best-effort นั่นคือข้อมูลประเภท FTP, E-Mail, HTTP เป็นต้น

การติดต่อกันระหว่างส่วนประกอบต่าง ๆ อธิบายได้พอสังเขปดังนี้

- Packet Classifier (PC): เมื่อทราฟฟิกแบ่งออกเป็น 3 ลักษณะดังนั้นการจัดตารางเวลาให้กับทราฟฟิกต้องรองรับลักษณะของทราฟฟิกแต่ละประเภทด้วย ดังนั้นต้องมีเครื่องมือในการแยกแพ็กเก็ตว่าเป็นของทราฟฟิกประเภทใดโดยจะรับแพ็กเก็ตที่เข้ามาแล้วหากมีการร้องขอทรัพยากรจะส่งไปถาม RM และเมื่อได้คำตอบแล้วจะส่งแพ็กเก็ตออกไปตามเส้นทางที่เหมาะสม
- Class Based Queuing (CBQ): เป็นส่วนที่จัดการคิวของแพ็กเก็ตของทราฟฟิกแต่ละประเภทตามกลไกของ CBQ และทำการส่งแพ็กเก็ตออกไปตามลิงค์ที่กำหนดไว้เพื่อส่งไปยังเราเตอร์ถัดไป
- Resource Management (RM): จะทำการตรวจสอบการร้องขอทรัพยากรที่เข้ามาเมื่อตรวจสอบว่ามีทรัพยากรเพียงพอหรือไม่ โดยใช้ฟังก์ชั่น RSVF ในการจองทรัพยากรและส่งผลให้กับ QOSR ทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Resource Reservation Function (RSVF): เป็นฟังก์ชั่นที่ใช้ในการจองทรัพยากร โดยรับคำสั่งมาจาก RM และส่งผลการจองให้กับ RM ทราบ
- Quality of Service Routing algorithm (QOSR): เป็นอัลกอริทึมในการหาเส้นทาง โดยหาการจองสำเร็จจะทำการหาเส้นทางเพื่อไปยังปลายทางต่อไป หากไม่สำเร็จจะส่งแพ็กเก็ตความผิดพลาด (Error Packet) กลับไปยังเราเตอร์ก่อนหน้า รายละเอียดได้อธิบายไว้ในบทที่ 3 ส่วนที่ 3.3



รูปที่ 5.3 การจัดตารางเวลาให้กับแพ็กเก็ตแต่ละประเภทที่เข้ามายังเราเตอร์

จากรูปที่ 5.3 แสดงกระบวนการเมื่อแพ็กเก็ตถูกแยกไปตามเส้นทางเชื่อมโยงต่างๆ (E_1-E_n) โดยตัวแยกแพ็กเก็ต (Packet Classifier) และในแต่ละเส้นทางเชื่อมโยงจะมีการจัดการคิวโดยอัลกอริทึมของ CBQ [20]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองและผลการทดลอง

ในบทนี้เป็นการกำหนดภาระงานของระบบที่เกิดขึ้น การออกแบบการทดลอง แล้วนำผลจากตัวแบบจำลองมาวิเคราะห์และแปลผล โดยวัตถุประสงค์ของการทดลองเพื่อศึกษาผลที่เกิดการเปลี่ยนแปลงกลไกในการจองทรัพยากรของเครือข่าย โดยค่าที่ต้องการศึกษาคือ อัตราเฉลี่ยความสำเร็จในการสร้างการเชื่อมต่อ (ACAR) ระยะทางของเส้นทางที่ใช้ในการส่งข้อมูล (ARD) ระยะเวลาที่ใช้ในการสร้างการเชื่อมต่อ (ACST) ซึ่งรายละเอียดของทั้ง 3 ค่าจะอธิบายรายละเอียดในส่วนต่อไป โดยตัวแปรที่มีผลต่อค่าทั้ง 3 ค่าคือ 1.ค่าภาระงาน 2.ค่าความต้องการแบนด์วิดท์ของผู้ใช้แต่ละคน 3.ค่าเวลาที่จำกัดในการสร้างการเชื่อมต่อ ทั้งนี้เพื่อศึกษาถึงความไวในการสร้างการเชื่อมต่อ นอกจากนั้นยังศึกษาถึงผลกระทบที่เกิดขึ้นจากการเปลี่ยนกลไกการจองทรัพยากรดังกล่าว ว่ามีผลเป็นอย่างไร เช่น ค่าทรูพุท (Throughput) ค่าความหน่วงเวลาของแพ็กเก็ต (Packet Delay) ทั้งนี้เนื่องจากหากสามารถเพิ่มความสำเร็จในการสร้างการเชื่อมต่อได้สูงขึ้น แต่มีค่าทรูพุทและค่าความหน่วงเวลามากเกินไป แสดงว่ากลไกที่เสนอดังกล่าวไม่เป็นประโยชน์มากนัก

6.1 การกำหนดภาระงานของระบบ

ภาระงานของระบบมาจากแพ็กเก็ตข้อมูลโดยแพ็กเก็ตข้อมูลเกิดขึ้นจากเครือข่ายย่อยปลายทางหนึ่งเดินทางไปยังอีกปลายทางหนึ่ง โดยขนาดของภาระงานจะส่งผลต่อค่า ACAR ACST ARD ทรูพุท และค่าความหน่วงเวลาของแพ็กเก็ต

พารามิเตอร์ในแบบจำลองแบ่งได้เป็น 2 ประเภท ซึ่งกล่าวรายละเอียดไปแล้วในบทที่ 4 ในส่วนนี้จะแสดงถึงรายละเอียดของค่าพารามิเตอร์ทั้ง 2 ประเภทที่นำมาใช้ในการจำลอง

ตารางที่ 6.1 พารามิเตอร์ของแบบจำลองเครือข่าย

ชื่อพารามิเตอร์	ค่าที่เป็นไปได้	หมายเหตุ
โทโปโลยีเครือข่าย	- โทโปโลยี NSFNET-T1	ประกอบด้วย 13 โหนด และ 19 เส้นทางเชื่อม [13]
ขนาดของบัฟเฟอร์แพ็กเก็ตข้อมูลภายในโหนด	- อนันต์(infinity)	

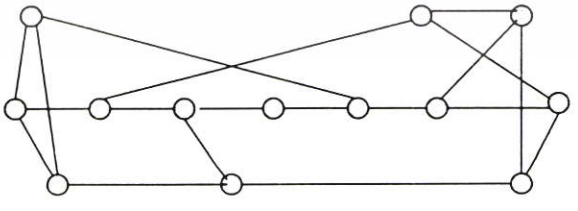
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.1 (ต่อ)

ชื่อพารามิเตอร์	ค่าที่เป็นไปได้	หมายเหตุ
อัลกอริทึมในการค้นหาเส้นทาง	- อัลกอริทึมเดิมที่มีอยู่ (Existing Algorithm) - อัลกอริทึมที่นำเสนอ (Proposed Algorithm) - อัลกอริทึม Flooding	Existing Algorithm คืออัลกอริทึมที่นำเสนอใน [10]
แบนด์วิดท์ของเส้นทางเชื่อมโยง	T1	1.544 Mbps
ภาระงาน	10%, 20%, 30%, 40%, 50% 60%, 70%, 80%, 90%, 100%	กำหนดให้เป็นเปอร์เซ็นต์ของอัตราความเร็วข้อมูลของผู้ใช้ของ T1 (1.344 Mbps) ซึ่งเป็นแบนด์วิดท์สูงสุดของเส้นทางเชื่อมโยงของโหนดเครือข่ายย่อยปลายสุดเชื่อมโยงอยู่
จำนวนแบนด์วิดท์ที่ผู้ส่งต้องการ	10%, 20%, 30%, 40%, 50% 60%, 70%, 80%, 90%, 100%	กำหนดแบนด์วิดท์สูงสุดของผู้ส่งแต่ละคนที่ทำการร้องขอ
จำกัดเวลาในการสร้างการเชื่อมต่อ	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	วินาที
ขนาดของแพ็กเก็ตข้อมูล	512 ไบต์	ขนาดไอพีแพ็กเก็ตโดยเฉลี่ย

6.2 การทดลอง

ในการทดลองนั้นใช้รูปแบบของเครือข่ายที่นำมาใช้คือ NSFNET-T1 [13] แสดงดังรูปที่ 6.1 ในแต่ละเส้นทางการเชื่อมต่อมีการส่งข้อมูลแบบสองทาง (Full Duplex) การเดินทางของแพ็กเก็ตการร้องขอไปยังเราเตอร์จะเป็นการกระจายแบบเอ็กซ์โปเน็นเชียล (Exponential Distribution) และแบนด์วิดท์ที่ขอจะเป็นการสุ่มโดยใช้การกระจายแบบยูนิฟอร์ม (Uniform Distribution) ส่วนผู้ส่งและผู้รับจะทำการสุ่มขึ้นมาโดยใช้การกระจายแบบยูนิฟอร์มเช่นกัน การทดลองจะทำทดลองทั้งหมด 5 ครั้ง แต่ละค่าของผลการทดลองที่แสดงบนกราฟจะเป็นค่าเฉลี่ยของการทดลองทั้ง 5 ครั้งดังกล่าว



รูปที่ 6.1 โทโปโลยีมาตราฐานของ NSFNET-T1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดประสิทธิภาพของอัลกอริทึมในงานวิจัยนี้ เมตริกซ์ที่น่าในใจได้แก่

1. อัตราเฉลี่ยความสำเร็จในการสร้างการเชื่อมต่อ (Average Call Acceptance Rate : ACAR)

$$ACAR = \frac{\sum_{i=1}^{N} accepted(R_i)}{N} \qquad (6.1)$$

เมื่อ N คือ จำนวนของคำร้องขอที่เข้ามา

accepted(R_i) = 1 หมายถึง คำร้องขอที่ i ได้รับการยอมรับ

accepted(R_i) = 0 หมายถึง คำร้องขอที่ i ได้รับการปฏิเสธ

ACAR ไม่มีหน่วย เนื่องจากหน่วยทั้งเศษและส่วนตัดกันหมดพอดี

2. เวลาเฉลี่ยที่ใช้ในการสร้างการเชื่อมต่อ (Average Call Setup Time : ACST)ในเทอมของจำนวนโหนดที่ได้รับการสร้างการเชื่อมต่อได้สำเร็จ

$$ACST = \frac{\sum_{i=1}^{N} setup(R_i)}{\sum_{i=1}^{N} accepted(R_i)} \qquad (6.2)$$

เมื่อ setup(R_i) คือ จำนวนของโหนดที่ได้รับคำร้องขอที่ i

ACST มีหน่วยเป็นโหนดต่อครั้ง

3. ค่าเฉลี่ยของจำนวนเราเตอร์ที่ใช้ในการสร้างเส้นทาง (Average Routing Distance : ARD)

$$ARD = \frac{\sum_{i=1}^{N} dist(R_i)}{\sum_{i=1}^{N} accepted(R_i)} \qquad (6.3)$$

เมื่อ dist(R_i) คือ ความยาวของเส้นทางที่ใช้ส่งข้อมูลของคำร้องขอที่ i (ในเทอมของจำนวนของเราเตอร์ในเส้นทาง)

ARD มีหน่วยเป็นโหนดต่อครั้ง

เมตริกซ์ตัวแรกมีความสำคัญมากเนื่องจากเป็นตัววัดประสิทธิภาพการทำงานของการเชื่อมต่อทั้งหมด และเมตริกซ์ตัวที่สองสำคัญต่อแอปพลิเคชั่นประเภทมัลติมีเดียซึ่งต้องการสร้างการเชื่อมต่อให้เร็วที่สุด และเมตริกซ์ที่สามก็มีความจำเป็นเนื่องจากการมีเส้นทางที่สั้นจะหมายถึงจะมีค่าใช้จ่าย (Cost) ที่น้อยลงด้วยเช่นกัน

6.3 ผลการทดลอง

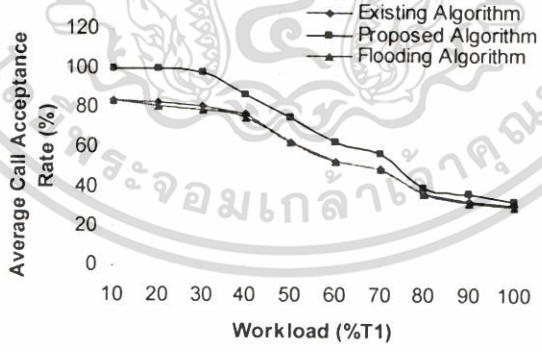
ในส่วนนี้มีจุดประสงค์เพื่อตรวจสอบประสิทธิภาพของการทำงานของอัลกอริทึมที่นำเสนอในงานวิจัยนี้ ค่าสถิติรวบรวมในช่วงเวลาสถานะคงที่ (Steady State) ของการจำลองแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำเสนอด้วยค่าเฉลี่ยรวมและวิเคราะห์ช่วงค่าความเชื่อมั่น (Confidence Interval) ที่ 95% การวัดประสิทธิภาพของอัลกอริทึมดังกล่าวแบ่งเป็นหลายๆ กรณี ดังนี้

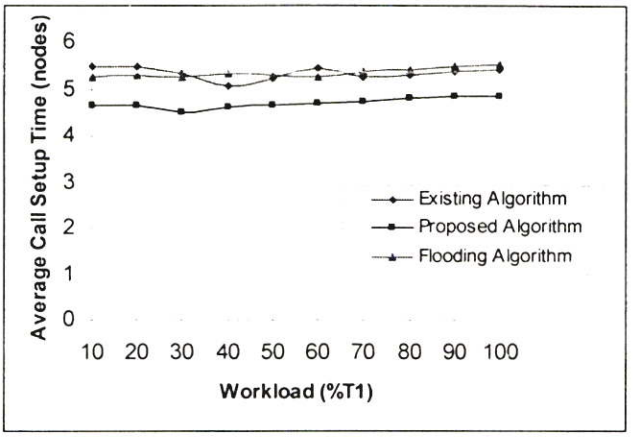
6.3.1 ผลกระทบจากภาระงาน

เมื่อเปลี่ยนแปลงภาระงาน (Workload) จากกราฟรูปที่ 6.2-6.5 แสดงผลกระทบของค่าเมตริกซ์ ACAR, ACST และ ARD เมื่อมีการเปลี่ยนแปลงค่าของภาระงานในรูปที่ 6.2 ค่าของ ACAR จะมีค่าลดลงเมื่อภาระงานมากขึ้นเนื่องมาจากภาระงานเพิ่มขึ้นทำให้มีการใช้ทรัพยากรของเครือข่ายมากขึ้น ดังนั้นการร้องขอที่สำเร็จนั่นมีน้อยลง (accepted(R)) เพราะไม่สามารถจองทรัพยากรได้นั่นเอง ทำให้ค่า ACAR มีค่าต่ำลง จะเห็นว่าอัลกอริทึมใหม่ที่นำเสนอมีค่า ACAR สูงกว่าอัลกอริทึมเดิม เนื่องจากการจองทรัพยากรตั้งแต่แรกในขั้น Probing phase ทำให้มีความแน่นอนมากกว่า ที่ระดับความเชื่อมั่น 95% เมื่อภาระงานน้อย ๆ (10%-30%) และภาระงานปานกลาง (40%-70%) ทั้งสามอัลกอริทึมมีค่า ACAR แตกต่างกัน เมื่อภาระงานสูง (80%-100%) ค่า ACAR ของทั้งสามอัลกอริทึมไม่แตกต่างกัน รูปที่ 6.3 จะเห็นว่าเมื่อภาระงานสูงขึ้นค่า ACST มีค่าสูงขึ้นด้วยเนื่องจากต้องใช้เวลาในการหาเส้นทางในเครือข่ายที่ตรงกับความต้องการมากขึ้นตามไปด้วย รูปที่ 6.3 แสดงให้เห็นอัลกอริทึมที่นำเสนอใหม่มีค่า ACST ต่ำว่าเนื่องจากมีการตรวจสอบทรัพยาก่อนที่จะส่งแพ็กเก็ต Probe ต่อไปยัง เราเตอร์ถัดไป ถ้ามีทรัพยากรไม่เพียงพอ แพ็กเก็ต Probe จะถูกทิ้งไป ทำให้จำนวนเราเตอร์ที่ได้รับแพ็กเก็ต Probe มีน้อยกว่าวิธีการเดิม ที่ระดับความเชื่อมั่น 95% ค่า ACST ของทั้งสามอัลกอริทึมมีความแตกต่างกัน ที่ภาระงานทั้ง 3 ขนาด ส่วนรูปที่ 6.4 จะเห็นว่าทั้งสามอัลกอริทึมมีค่า ARD ไม่ต่างกันมากนักทั้งนี้เนื่องจากผลลัพธ์ของการหาเส้นทางของทั้งสามวิธีดังกล่าวได้ผลลัพธ์เป็นเส้นทางเดียวกัน หรือมีจำนวนโหนดใกล้เคียงกัน

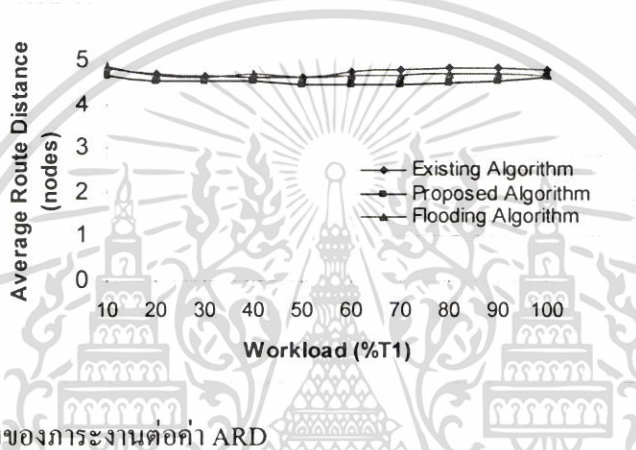


รูปที่ 6.2 ผลกระทบของภาระงานต่อค่า ACAR

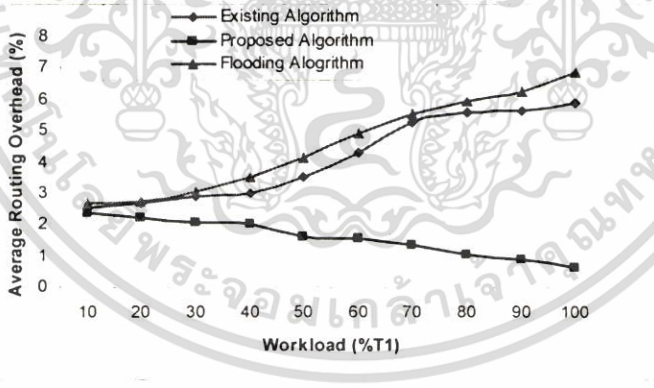
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 ผลกระทบของภาระงานต่อค่า ACST



รูปที่ 6.4 ผลกระทบของภาระงานต่อค่า ARD



รูปที่ 6.5 แสดงสัดส่วนของแพ็กเก็ตโอเวอร์เฮดต่อแพ็กเก็ตข้อมูล

จำนวนของแพ็กเก็ตที่ใช้ในการหาเส้นทาง (Routing Packet) มีความสำคัญเช่นกัน เนื่องจากจำนวนแพ็กเก็ตที่ใช้ในการหาเส้นทางมีสูงจะทำให้ใช้ทรัพยากรของเครือข่ายมากขึ้น และทำให้เกิดความคับคั่ง (Congestion) บนเครือข่ายได้ ในรูปที่ 6.5 จะเห็นว่าเมื่อจำนวนภาระงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

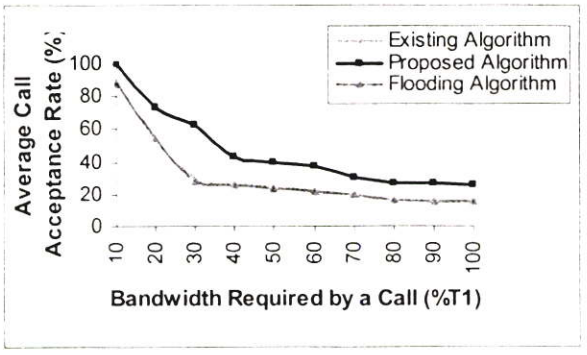
สูงขึ้น ทำให้มีโอเวอร์เฮดสูงขึ้น เนื่องจากทำให้ทรัพยากรของเครือข่ายถูกใช้มากขึ้นจึงเกิดการปฏิเสธการร้องขอมากขึ้น ดังนั้นโอเวอร์เฮดจึงมากขึ้นตามไปด้วย แต่จะเห็นว่าจำนวนแพ็กเก็ตของอัลกอริทึมที่นำเสนอในงานวิจัยนี้ค่าน้อยลงเรื่อย ๆ เนื่องจากจะไม่ส่งแพ็กเก็ต Probe ต่อหากตรวจสอบพบว่ามีทรัพยากรไม่พอ ตามขั้นตอนที่อธิบายในส่วนที่ 3.3 ที่ระดับความเชื่อมั่น 95% อัลกอริทึมทั้ง 3 มีความแตกต่างกันในทุกขนาดของภาระงาน

6.3.2 ผลกระทบจากแบนด์วิดท์ที่ผู้ใช้แต่ละคนต้องการ

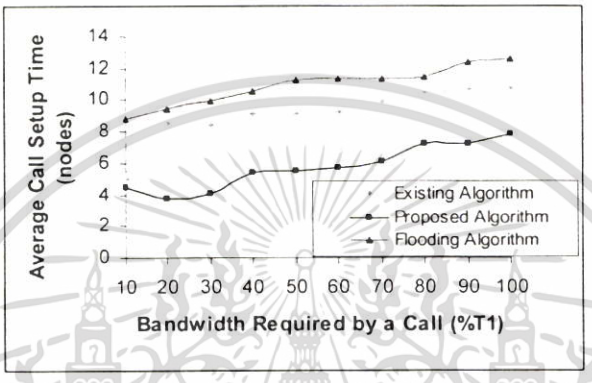
เมื่อมีการกำหนดขนาดของแบนด์วิดท์สูงสุดที่ผู้ส่งต้องการ (Maximum Bandwidth Required by a Call) รูปที่ 6.6 จะเห็นว่าค่า ACAR มีต่ำลงเมื่อมีการร้องขอแบนด์วิดท์มากขึ้น ทั้งนี้เนื่องจากการร้องขอแบนด์วิดท์ที่มากขึ้นนั้นทำให้โอกาสที่จะการร้องขอจะสำเร็จนั้นมีน้อยลงนั่นคือค่า accepted(R) น้อยลงจากสมการที่ (6.1) จึงทำให้ค่าของ ACAR น้อยลง ที่ระดับความเชื่อมั่น 95% เมื่อมีการร้องขอแบนด์วิดท์น้อยๆ ทั้งสามอัลกอริทึมจะมีค่า ACAR ต่างกันและเมื่อมีการร้องขอ แบนด์วิดท์ปานกลาง และการร้องขอแบนด์วิดท์สูงๆ ค่า ACAR ของทั้งสามอัลกอริทึมแตกต่างกัน เช่นเดียวกับค่า ACST รูปที่ 6.7 เมื่อมีการขอแบนด์วิดท์ที่มากขึ้นทำให้โอกาสการร้องขอที่จะสำเร็จมีน้อยลงนั่นคือค่า accepted(R) น้อยลงทำให้ต้องหาเส้นทางอื่นๆ ที่จะยอมรับการร้องขอนี้ทำให้เสียเวลาในการหาเส้นทางดังกล่าว ดังนั้นเส้นทางที่ได้อาจจะเป็นเส้นทางที่ยาวขึ้นซึ่งตรงกับสมการที่ (6.2) ค่าของ ACST มากขึ้นเมื่อ accepted(R) น้อยลง รูปที่ 6.8 มีเหตุผลเช่นเดียวกับรูปที่ 6.4 ซึ่งจะเป็นตามสมการที่ (6.3) นั่นคือตัวหารมาก ค่า ACST จะน้อยลง รูปที่ 6.9 แสดงให้เห็นว่าอัลกอริทึมที่นำเสนอในงานวิจัยนี้มีจำนวนแพ็กเก็ตน้อยกว่า เนื่องจากเมื่อต้องการแบนด์วิดท์มากขึ้นทำให้เครือข่ายไม่สามารถให้แบนด์วิดท์ตามต้องการได้จึงทำให้เกิดความผิดพลาดในการร้องขอ

อัลกอริทึมที่นำเสนอในงานวิจัยนี้จะทิ้งแพ็กเก็ต Probe ไปแทนที่จะถูกส่งต่อไปยังเราเตอร์ตัวต่อไปเมื่อทราบว่ามีแบนด์วิดท์ไม่พอ ในขณะที่อัลกอริทึมเดิมทั้งสองอัลกอริทึมไม่มีการตรวจสอบแบนด์วิดท์ว่ามีเพียงพอหรือไม่ จึงส่งแพ็กเก็ต Probe ต่อไปยังเราเตอร์ถัดไป เหตุผลดังกล่าวทำให้อัลกอริทึมที่นำเสนอนี้มีจำนวนแพ็กเก็ตน้อยกว่า และลดลงเรื่อย ๆ ที่ระดับความเชื่อ 95% เมื่อมีการร้องขอแบนด์วิดท์น้อยๆ อัลกอริทึมที่นำเสนอมีโอเวอร์เฮดแพ็กเก็ตไม่แตกต่างจากอัลกอริทึมเดิม และเมื่อมีการร้องขอแบนด์วิดท์ปานกลางถึงร้องขอแบนด์วิดท์สูง ๆ อัลกอริทึมที่นำเสนอมีโอเวอร์เฮดแพ็กเก็ตแตกต่างจากอัลกอริทึมเดิม

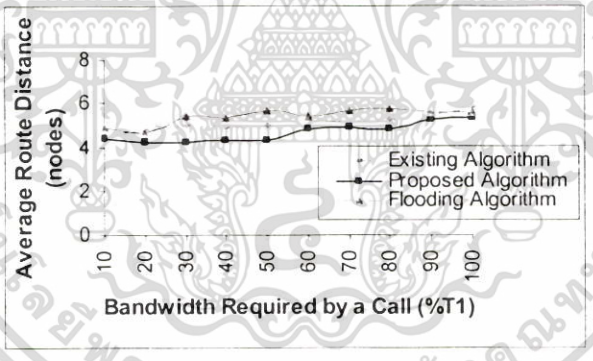
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 ผลกระทบของแบนด์วิดท์ที่ร้องขอต่อค่า ACAR

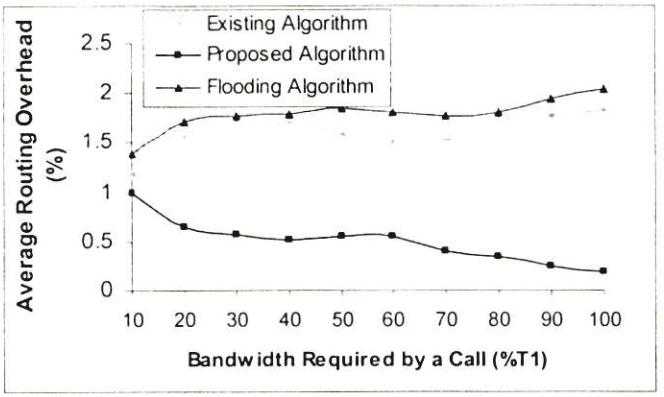


รูปที่ 6.7 ผลกระทบของแบนด์วิดท์ที่ร้องขอต่อค่า ACST



รูปที่ 6.8 ผลกระทบของแบนด์วิดท์ที่ร้องขอต่อค่า ARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

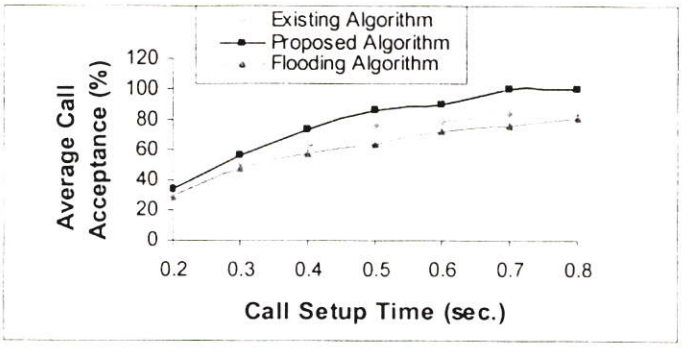


รูปที่ 6.9 ผลกระทบของแบนด์วิดท์ที่ร้องขอต่อจำนวนแพ็กเก็ตในการสร้างการเชื่อมต่อ

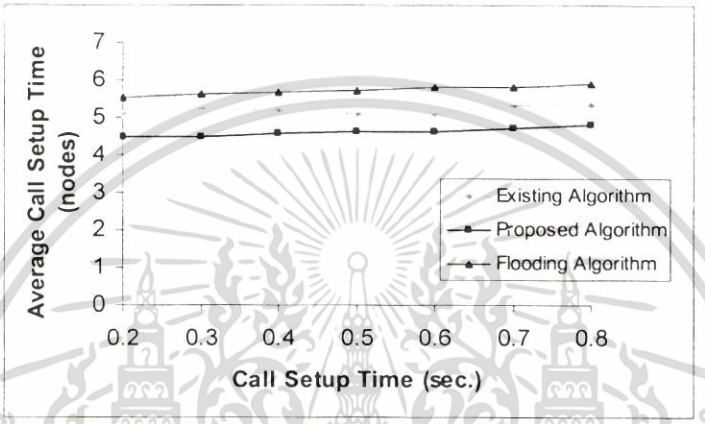
6.3.3 ผลกระทบของเวลาที่ใช้ในการสร้างการเชื่อมต่อ

การศึกษาถึงผลกระทบของเวลาที่ใช้ในการสร้างการเชื่อมต่อ โดยมีการจำกัดเวลาที่ใช้ในการสร้างการเชื่อมต่อเพื่อเปรียบเทียบความเร็วในการสร้างการเชื่อมต่อในงานวิจัยนี้กับวิธีการเดิม ข้อดีคือหากอัลกอริทึมใดใช้เวลาในการสร้างเชื่อมต่อได้น้อย แสดงว่าจะมีค่าใช้จ่าย (Cost) ที่ต่ำเช่นกัน จากรูป 6.10 จะเห็นว่าค่า ACAR สูงขึ้นเมื่อมีการเพิ่มเวลาที่ใช้ในการสร้างการเชื่อมต่อ เมื่อพิจารณา ณ เวลาที่เท่ากันจะเห็นว่าอัลกอริทึมที่นำเสนอมีค่า ACAR สูงกว่าวิธีการเดิม เนื่องจากวิธีการที่นำเสนอใหม่นี้มีความสำเร็จในการจองทรัพยากรได้มากกว่าจึงทำให้ ACAR สูงกว่า ที่ระดับความเชื่อมั่น 95% เมื่อกำหนดเวลาในการสร้างการเชื่อมต่อน้อย ๆ (0.1-0.4 Sec.) ทั้งสามอัลกอริทึมจะมีค่า ACAR ไม่ต่างกันและเมื่อกำหนดเวลาในการสร้างการเชื่อมต่อปานกลาง (0.5-0.6 Sec.) และกำหนดเวลาในการสร้างการเชื่อมต่อสูงๆ (0.7-0.8 Sec.) ค่า ACAR ของทั้งสามอัลกอริทึมแตกต่างกัน ในรูป 6.11 แสดงให้เห็นว่าอัลกอริทึมที่นำเสนอมีค่า ACST ต่ำกว่าเหตุผลคือ อัลกอริทึมใหม่มีการตรวจสอบทรัพยากรก่อน ก่อนที่จะส่งแพ็กเก็ต Probe ต่อไป เมื่อพบว่าทรัพยากรที่ร้องขอไม่พอจะไม่ส่งต่อ ทำให้จำนวนเราเตอร์ที่ได้รับแพ็กเก็ต Probe มีน้อยกว่าอัลกอริทึมเดิม รูป 6.13 เมื่อเวลาในการสร้างการเชื่อมต่อมากขึ้น จำนวนแพ็กเก็ตที่ใช้ในการสร้างการเชื่อมต่อจะเพิ่มขึ้นตามไปด้วย เนื่องจากแพ็กเก็ต Probe จะสามารถเดินทางได้นานขึ้นนั่นเอง จากรูปที่ 6.13 อัลกอริทึมเดิมมีจำนวนแพ็กเก็ตสูงกว่าเนื่องจากเมื่อเกิดความผิดพลาดขึ้นจะทำให้มีการส่งแพ็กเก็ตเพื่อบอกความผิดพลาดสูงกว่าอัลกอริทึมใหม่ที่นำเสนอ ตามที่อธิบายในส่วน 3.2 และ 3.3 ส่วนรูปที่ 6.12 มีเหตุผลเช่นเดียวกับรูปที่ 6.4 ซึ่งจะเป็นตามสมการที่ (6.3)

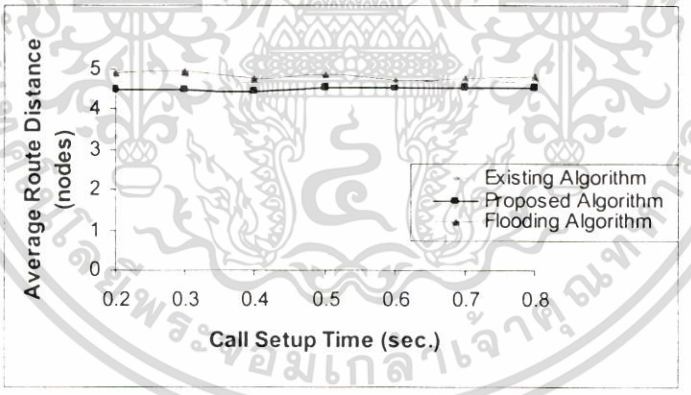
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.10 ผลกระทบต่อเวลาที่ใช้ในการสร้างการเชื่อมต่อ ต่อค่า ACAR

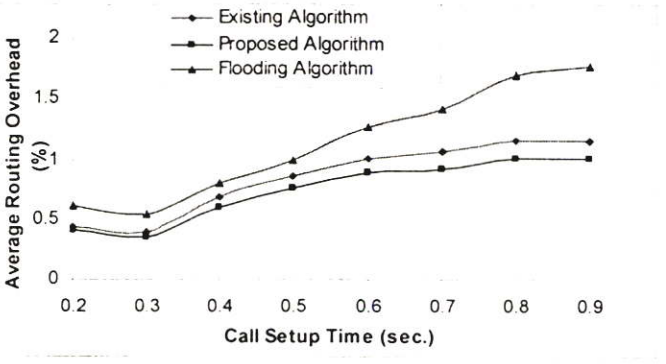


รูปที่ 6.11 ผลกระทบต่อเวลาที่ใช้ในการสร้างการเชื่อมต่อ ต่อค่า ACST



รูปที่ 6.12 ผลกระทบต่อเวลาที่ใช้ในการสร้างการเชื่อมต่อ ต่อค่า ARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



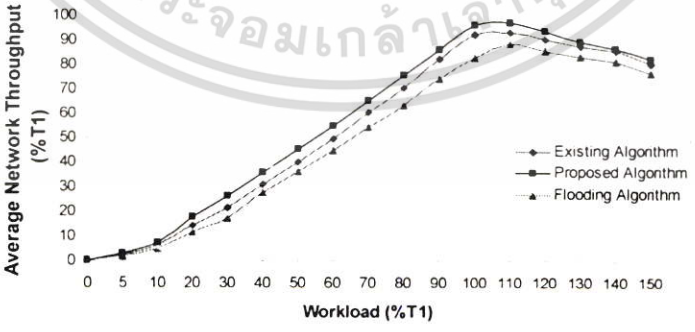
รูปที่ 6.13 ผลกระทบต่อเวลาที่ใช้ในการสร้างการเชื่อมต่อจำนวนแพ็กเก็ตในการสร้างการเชื่อมต่อ

6.3.4 ผลกระทบต่อทรูพุทเฉลี่ย (Average Throughput)

การศึกษาทรูพุท (Throughput) ที่ได้ จะได้จากสูตร

$$Throughput = \frac{datapaket}{datapacket + overhead} \; X \; 100 \qquad (6.4)$$

จะเห็นว่าทรูพุทที่ได้จากอัลกอริทึมใหม่สูงกว่าอัลกอริทึมเดิม เนื่องจากอัลกอริทึมที่นำเสนอนี้สามารถมีอัตราการสร้างการเชื่อมต่อสำเร็จที่สูงกว่าและมีจำนวนของแพ็กเก็ตที่ใช้ในการสร้างการเชื่อมต่อมีน้อยกว่าอัลกอริทึมเดิมจึงทำให้มีทรูพุทที่สูงกว่าเล็กน้อย เมื่อเพิ่มภาระงานมากขึ้นดังนั้นสัดส่วนของแพ็กเก็ตโอเวอร์เฮดต่อแพ็กเก็ตของมูลจึงเป็นสัดส่วนที่น้อยลงจึงทำให้กราฟออกมาในลักษณะกราฟเส้นตรงแสดงดังรูป 6.14 ที่ระดับความเชื่อมั่น 95% เมื่อมีการร้องขอแบนด์วิดท์น้อยๆ (0%-20%) ทั้งสามอัลกอริทึมจะมีค่าทรูพุทเฉลี่ยไม่ต่างกันและที่ภาระงานปานกลาง (30%-70%) และภาระงานสูงค่า (80%-100%) ทรูพุทเฉลี่ยของทั้งสามอัลกอริทึมแตกต่างกันที่ภาระงานที่เกินกว่าระบบจะรองรับได้ (>100%) ค่าทรูพุทเฉลี่ยของทั้งสามอัลกอริทึมแตกต่างกัน

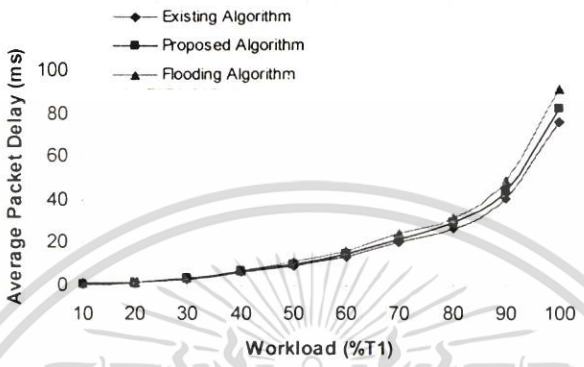


รูปที่ 6.14 ทรูพุทเฉลี่ยเมื่อมีการเปลี่ยนแปลงภาระงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.5 ผลกระทบต่อความหน่วงเวลา (Delay)

ในหัวข้อนี้จะทำการศึกษาผลกระทบของอัลกอริทึมค้นหาเส้นทางต่อความหน่วงเวลา จากรูปที่ 6.15 จะเห็นว่าอัลกอริทึมที่นำสเนอมีความหน่วงเวลาที่สูงกว่าอัลกอริทึมเดิมเล็กน้อย (ประมาณ 6%) ทั้งนี้เนื่องจากอัลกอริทึมที่นำเสนอนั้นจะต้องเสียเวลาในการทำการตรวจสอบทรัพยากร แต่จากคำนวณทางสถิติ ที่ระดับความเชื่อมั่น 95% ทั้งสามอัลกอริทึมมีค่าความหน่วงเวลาไม่แตกต่างกัน

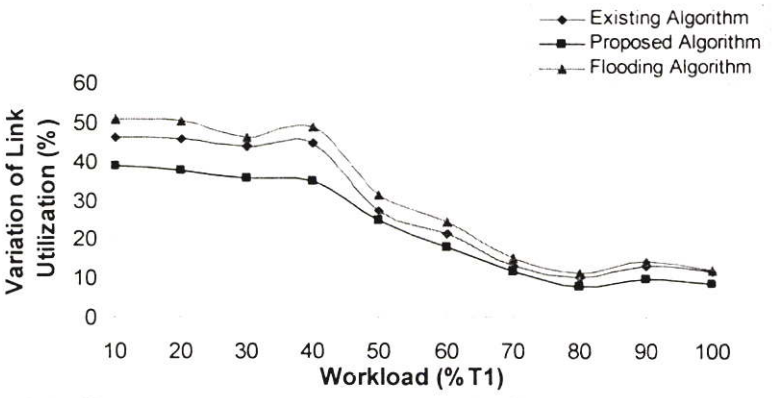


รูปที่ 6.15 เปรียบเทียบความหน่วงเวลาที่เกิดจากอัลกอริทึมในการหาเส้นทางของสามอัลกอริทึม

6.3.6 ผลกระทบต่อค่าความแปรปรวนของการใช้เส้นทางเชื่อมโยง (Variation of Link Utilization)

อัลกอริทึมในการค้นหาเส้นทางสำหรับ QoS นั้นมีข้อดีกว่าอัลกอริทึมการค้นหาเส้นทางแบบ Best-Effort คือว่ามีการกระจายของทราฟฟิกที่ดีกว่า เนื่องจากอัลกอริทึมในการหาเส้นทางแบบ Best-Effort นั้นจะใช้เส้นทางที่สั้นที่สุดเพียงอย่างเดียว ความแปรปรวนของการใช้เส้นทางเชื่อมโยง แสดงให้เห็นถึงการกระจายทราฟฟิกว่าดีเพียงใด จากรูปที่ 6.16 แสดงเปรียบเทียบความแปรปรวนของการใช้เส้นทางเชื่อมโยงที่เกิดจากการใช้อัลกอริทึมทั้งสาม จะเห็นว่าเมื่อภาระงานเพิ่มขึ้นความแปรปรวนของการใช้เครือข่ายจะน้อยลง นั่นหมายถึงมีการใช้เครือข่ายให้เป็นประโยชน์มากขึ้นมีการกระจายทราฟฟิกที่ดี อัลกอริทึมที่นำเสนอมีความแปรปรวนน้อยกว่าอัลกอริทึมเดิม หมายถึงมีการกระจายทราฟิกที่ดีกว่า เนื่องจากมีฟังก์ชั่น Bracktract() มีการใช้เส้นทางอื่นๆ ที่ว่างอยู่เป็นไปอย่างมีประสิทธิภาพ การใช้ประโยชน์จากเครือข่าย (Utilization) จึงสูงกว่าอัลกอริทึมเดิม ที่ระดับความเชื่อมั่น 95% ที่ภาระงานทุกระดับ ความแปรปรวนของอัลกอริทึมที่นำเสนอแตกต่างจากอัลกอริทึมเดิมอย่างชัดเจน

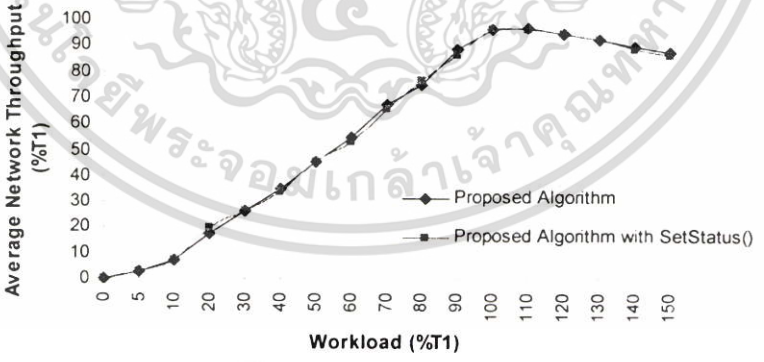
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.16 เปรียบเทียบความแปรปรวนของการของการใช้เส้นทางเชื่อมโยง

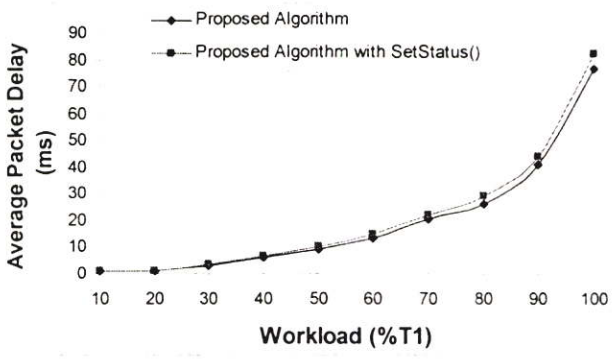
6.3.7 เปรียบเทียบทรูพุทเฉลี่ยอัลกอริทึมที่มีและไม่มีฟังก์ชั่น SetStatus()

งานวิจัยนี้ได้นำเสนออัลกอริทึมในการค้นหาเส้นทาง โดยมีการปรับเปลี่ยนกลไกการจองทรัพยากร และนำเสนอฟังก์ชั่น SetStatus() เพื่อใช้ในการกำหนดสถานะของทรัพยากร รายละเอียดกล่าวไว้ในหัวข้อ 3.3 ทั้งนี้เนื่องจากต้องการที่จะเพิ่มการใช้ทรัพยากรของเครือข่ายให้มีประสิทธิภาพมากขึ้น แต่จากการทดลองพบว่าทรูพุทเฉลี่ยของเครือข่ายที่ได้จากอัลกอริทึมที่มีฟังก์ชั่น SetStatus() และอัลกอริทึมที่ไม่มีฟังก์ชั่น SetStatus() ไม่แตกต่างกันมากนัก ดังแสดงในรูปที่ 6.17 ทั้งนี้สันนิษฐานว่าเกิดจากการที่ทราฟฟิกที่เกิดขึ้นในเครือข่าย มีการกระจายที่สม่ำเสมอ จึงไม่ค่อยมีการใช้เส้นทางเดิมในการส่งข้อมูลมากนัก ดังนั้นทำให้การจองทรัพยากรไม่ส่งผลต่อทรูพุทของเครือข่ายมากนัก และถ้าหากเส้นทางใดที่มีการจองทรัพยากรไว้ และมีผู้อื่นต้องการใช้เส้นทางนั้น และหากเส้นทางนั้นมีทรัพยากรไม่เพียงพอ ก็จะไปใช้เส้นทางอื่น ๆ ที่มีทรัพยากรเพียงพอกับความต้องการ



รูปที่ 6.17 เปรียบเทียบทรูพุทเฉลี่ยอัลกอริทึมที่มีและไม่มีฟังก์ชั่น SetStatus()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



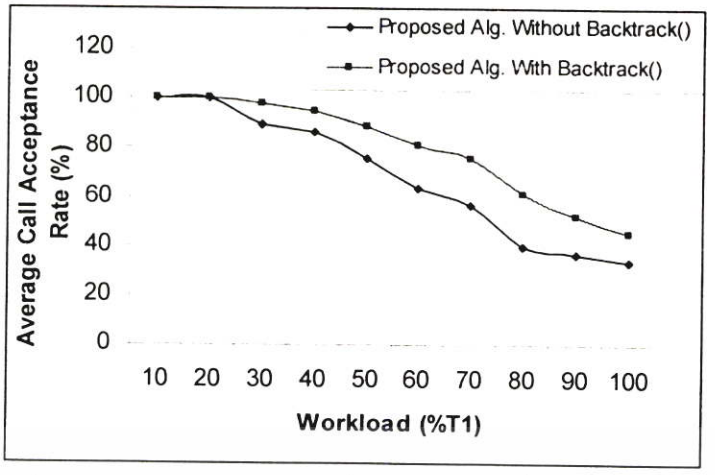
รูปที่ 6.18 เปรียบเทียบความหน่วงเวลาเฉลี่ยของแพ็กเก็ตของอัลกอริทึมที่มีและไม่มีฟังก์ชั่น SetStatus()

เมื่อทำการเปรียบเทียบความหน่วงเวลาเฉลี่ยของแพ็กเก็ตของอัลกอริทึมที่มีและไม่มีฟังก์ชั่น SetStatus() จะเห็นว่าอัลกอริทึมที่มีฟังก์ชั่น SetStatus() มีความหน่วงเวลามากกว่าอัลกอริทึมที่ไม่มีฟังก์ชั่น SetStatus() ประมาณ 8% ดังแสดงในรูปที่ 6.18 ทั้งนี้เกิดจากที่เราเตอร์เสียเวลาในการตรวจสอบสถานะของทรัพยากร ทำให้แพ็กเก็ตเสียเวลาในการรอผลจากการตรวจสอบดังกล่าว แต่ที่ระดับความเชื่อมั่น 95% อัลกอริทึมทั้งสองมีทรูพุทเฉลี่ยและความหน่วงเวลาเฉลี่ยไม่แตกต่างกัน

6.3.8 เปรียบเทียบข้อดีของการมีฟังก์ชั่น Bracktract()

ตามที่ได้อธิบายไว้ในหัวข้อ 3.3 ว่าวิธีการใหม่ที่นำเสนอนี้มีฟังก์ชั่น Bracktract() ที่ใช้ช่วยสำหรับหาเส้นทางอื่น ๆ ที่อยู่ใน Second-Level Router หากเส้นทางที่ส่งแพ็กเก็ต Probe ไปนั้นมีทรัพยากรไม่เพียงพอ และมีเส้นทางอื่นๆ ที่ยังไม่ได้ส่งแพ็กเก็ต Probe ไป ทั้งนี้เพื่อช่วยการเพิ่มความสำเร็จในการสร้างการเชื่อมต่อให้สูงขึ้น เนื่องจากเมื่อเกิดการล้มเหลวในการจองทรัพยากรอัลกอริทึมเดิมนั้นจะต้องส่งแพ็กเก็ตความผิดพลาดไปบอกกับผู้ส่ง และผู้ส่งจะทำการส่งแพ็กเก็ต Probe ออกไปใหม่ ซึ่งทำให้เสียเวลามาก และจำนวนแพ็กเก็ตในเครือข่ายก็สูงขึ้น แต่ถ้ามีฟังก์ชั่น Bracktract() แล้ว จะพยายามหาเส้นทางอื่นๆ ก่อน หากไม่สามารถหาเส้นทางได้จริงๆ จึงจะส่งแพ็กเก็ตความผิดพลาดไปบอกกับผู้ส่ง ในหัวข้อนี้จะมาพิจารณาถึงประสิทธิภาพของฟังก์ชั่น Bracktract() ว่าช่วยเพิ่มความสำเร็จในการเชื่อมต่อเพียงใด ซึ่งแสดงดังรูปที่ 6.19 จะเป็นว่าการมีฟังก์ชั่น Bracktract() ช่วยเพิ่มความสำเร็จในการเชื่อมต่อให้สูงขึ้น เฉลี่ยแล้วประมาณ 12% เลยทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.19 เปรียบเทียบค่า ACAR ของอัลกอริทึมที่มีและไม่มีฟังก์ชั่น Bracktract()

6.4 สรุปผลการทดลอง

การปรับเปลี่ยนกลไกการจองทรัพยากรสำหรับอัลกอริทึมค้นหาเส้นทางเพื่อทำให้เกิดการประกันการส่งข้อมูล (QoS) สามารถเพิ่มอัตราความสำเร็จในการสร้างเชื่อมต่อ (Average Call Acceptance Rate: ACAR) ได้มากกว่ากลไกการจองทรัพยากรแบบเดิมทั้งวิธีการที่นำเสนอใน [12] และ [9] นอกจากนี้จำนวนโหนดที่ได้รับแพ็กเก็ตขอสร้างการเชื่อมต่อยัง (ACST) มีจำนวนน้อยกว่าวิธีการเดิม นั่นหมายถึงวิธีการที่นำเสนอใช้ทรัพยากรของเครือข่ายน้อยลงไปด้วยเช่นกัน ในขณะที่เส้นทางที่ใช้ในการส่งข้อมูลของทั้งสามวิธีไม่เพิ่มขึ้น (ARD) ในส่วนของโอเวอร์เฮดในการค้นหาเส้นทางนั้น อัลกอริทึมที่นำเสนอมีโอเวอร์เฮดน้อยกว่าวิธีการเดิมอย่างชัดเจน สาเหตุเนื่องมาจากการตรวจสอบทรัพยากรก่อนที่จะส่งแพ็กเก็ต Probe ต่อไปนั่นเอง

ในส่วนของทรูพุทเฉลี่ยนั่น ที่ภาระงานสูง ๆ อัลกอริทึมที่นำเสนอจะมีค่าทรูพุทเฉลี่ยสูงกว่าอัลกอริทึมเดิมอย่างชัดเจน เนื่องจากอัลกอริทึมที่นำเสนอนี้สามารถมีอัตราการสร้างการเชื่อมต่อสำเร็จที่สูงกว่าและมีจำนวนของแพ็กเก็ตที่ใช้ในการสร้างการเชื่อมต่อมีน้อยกว่าอัลกอริทึมเดิมจึงทำให้มีทรูพุทที่สูงกว่า แต่อัลกอริทึมที่นำเสนอมีความหน่วงเวลาที่สูงกว่าอัลกอริทึมเดิมเล็กน้อย (ประมาณ 6%) ทั้งนี้เนื่องจากอัลกอริทึมที่นำเสนอนั้นจะต้องเสียเวลาในการทำการตรวจสอบทรัพยากร วิธีการที่นำเสนอในงานวิจัยนี้เหมาะสำหรับแอปพลิเคชั่นที่ต้องการการประกันแบนด์วิดท์เป็นหลัก เช่นแอปพลิเคชั่นประเภทมัลติมีเดีย เป็น ต้น และหากพบว่ามีเส้นทางมากกว่าหนึ่งเส้นทางที่ตรงกับความต้องการจะพิจารณาเส้นทางที่มีความหน่วงเวลาน้อยสุดเป็นลำดับต่อไป แต่สำหรับแอปพลิเคชั่นที่อาจจะต้องการการประกันความหน่วงเวลาหรือจิตเตอร์ เป็นหลัก อาจจะยังไม่เหมาะสมมากนักเนื่องจากวิธีการที่นำเสนอในงานวิจัยนี้ยังมีความหน่วงเวลาที่สูงกว่า Existing algorithm [12] และวิธีการนี้จะเหมาะสำหรับเครือข่ายที่มีความซับซ้อนหรือมีจำนวนโหนดสูงๆ หรือไม่ ผู้วิจัยจะทำการศึกษาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทสรุปและข้อเสนอแนะ

วิทยานิพนธ์ฉบับนี้มุ่งเน้นในการเพิ่มอัตราความสำเร็จในการสร้างการเชื่อมต่อ (Average Call Acceptance Rate: ACAR) ตามที่แอปพลิเคชั่นต้องการเป็นสำคัญ และศึกษาถึงผลกระทบที่เกิดจากการปรับเปลี่ยนกลไกในการจองทรัพยากรบนเครือข่ายของอัลกอริทึมค้นหาเส้นทาง (Routing Algorithm) ดังกล่าว โดยเปรียบเทียบกับอัลกอริทึมอัลกอริทึมเดิม 2 อัลกอริทึมคือ อัลกอริทึมของ Ghosh [12] ซึ่งในวิทยานิพนธ์ฉบับนี้เรียกว่า "Existing Algorithm" และ "Flooding Algorithm" [9]

วิธีการที่นำเสนอในวิทยานิพนธ์ฉบับนี้เป็นการเสนอกลไกในการจองทรัพยากรสำหรับอัลกอริทึมค้นหาเส้นทางสำหรับส่งข้อมูลมัลติมีเดียเพื่อทำให้เกิดการประกันการส่งข้อมูล (QoS) บนเครือข่ายอินเทอร์เน็ตแบบยูนิแคสต์ ซึ่งจะเน้นที่การหาเส้นทางที่มีแบนด์วิดท์เพียงพอเป็นอันดับแรกซึ่งเป็นความต้องการของการส่งข้อมูลมัลติมีเดีย ส่วนความหน่วงเวลาหรือจิตเตอร์นั้นจะมีความสำคัญเป็นอันดับรองลงไป การประเมินประสิทธิภาพเครือข่ายและผลกระทบที่เกิดจากการปรับเปลี่ยนกลไกการจองทรัพยากรดังกล่าวจากแบบจำลองที่สร้างขึ้น โดยการค้นหาเส้นทางในแบบจำลองเครือข่าย มาตรวัดประสิทธิภาพที่นำมาใช้ได้แก่ อัตราความสำเร็จในการสร้างการเชื่อมต่อ (ACAR) เวลาที่ใช้ในการสร้างการเชื่อมต่อในเทอมของโหนดที่ไดรับแพ็กเก็ตร้องขอ (ACST) และระยะทางของเส้นทางที่ใช้ส่งข้อมูล (ARD) นอกจากนี้ยังศึกษาผลกระทบอื่นๆ ที่เกิดขึ้นเช่น ทรูพุทเฉลี่ยของเครือข่าย โอเวอร์เฮดของการค้นหาเส้นทางที่เกิดขึ้น และความหน่วงเวลาเฉลี่ยของแพ็กเก็ต ผลลัพธ์จากการทดลองแสดงให้เห็นว่าประสิทธิภาพของกลไกการจองทรัพยากรดังกล่าวดีขึ้นเมื่อเปรียบเทียบกับวิธีการเดิม

จากการประเมินประสิทธิภาพทำให้ทราบได้ว่าการปรับเปลี่ยนกลไกการจองทรัพยากรสำหรับอัลกอริทึมค้นหาเส้นทางเพื่อทำให้เกิด QoS สามารถเพิ่มอัตราความสำเร็จในการสร้างเชื่อมต่อ (ACAR) ได้มากกว่ากลไกการจองทรัพยากรแบบเดิมทั้งวิธีการที่นำเสนอใน [12] และ [9] นอกจากนี้จำนวนโหนดที่ได้รับแพ็กเก็ตขอสร้างการเชื่อมต่อ (ACST) ยังมีจำนวนน้อยกว่าวิธีการเดิม นั่นหมายถึงวิธีการที่นำเสนอในวิทยานิพนธ์ฉบับนี้ใช้ทรัพยากรของเครือข่ายน้อยลงไปด้วยเช่นกัน เส้นทางที่ได้จากการค้นหาเส้นทางและใช้ในการส่งข้อมูลทั้งสาม อัลกอริทึมมีค่าใกล้เคียงกันแสดงให้เห็นว่าผลลัพธ์ที่ได้จากการค้นหาเส้นทางอาจจะได้เส้นทางเดียวกัน สรุปได้ว่าการเปลี่ยนแปลงกลไกในการจองทรัพยากรดังกล่าวสามารถเพิ่มอัตราความสำเร็จในการสร้างการเชื่อมต่อได้อย่างชัดเจนและยังได้ผลลัพธ์ของการหาเส้นทางที่มีประสิทธิภาพเช่นเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ เปอร์เซ็นต์ของแพ็กเก็ตที่ใช้ในการสร้างการเชื่อมต่อถือเป็นสิ่งสำคัญอันหนึ่ง ทั้งนี้เนื่องจากหากการสร้างการเชื่อมต่อมีการใช้จำนวนแพ็กเก็ตสูงทำให้มีการใช้ทรัพยาการสูงขึ้นตามไปด้วย ผลกระทบต่อประสิทธิภาพของเครือข่ายในส่วนของโอเวอร์เฮดในการค้นหาเส้นทางนั้น อัลกอริทึมที่นำเสนอมีโอเวอร์เฮดน้อยกว่าวิธีการเดิมอย่างชัดเจน เนื่องจากมีการตรวจสอบทรัพยากรก่อนที่จะส่งแพ็กเก็ต Probe ต่อไปยังเราเตอร์อื่นๆ นั่นเอง ส่วนค่าทรูพุทเฉลี่ยของเครือข่ายจากผลการทดลองพบว่าทรูพุทที่ได้จากอัลกอริทึมใหม่สูงกว่าอัลกอริทึมเดิม เนื่องจากอัลกอริทึมที่นำเสนอนี้มีอัตราการสร้างการเชื่อมต่อสำเร็จที่สูงกว่าและมีเปอร์เซ็นต์ของแพ็กเก็ตที่ใช้ในการสร้างการเชื่อมต่อน้อยกว่าอัลกอริทึมเดิมจึงทำให้มีทรูพุทที่สูงกว่า ในส่วนของความหน่วงเวลาที่เกิดขึ้นพบว่า ที่ภาระงานปานกลางทั้งสามอัลกอริทึมมีค่าความหน่วงเวลาไม่แตกต่างกัน

งานวิจัยนี้ได้นำเสนอฟังก์ชั่น SetStatus() เพื่อใช้ในการกำหนดสถานะของทรัพยากร ทั้งนี้เนื่องจากต้องการที่จะเพิ่มการใช้ทรัพยากรของเครือข่ายให้มีประสิทธิภาพมากขึ้น แต่จากการทดลองพบว่าทรูพุทเฉลี่ยของเครือข่ายที่ได้จากอัลกอริทึมที่มีฟังก์ชั่น SetStatus() และไม่มีฟังก์ชั่น SetStatus() ไม่แตกต่างกันมากนัก เนื่องจากถ้าหากเส้นทางใดที่มีการจองทรัพยากรไว้ และมีผู้อื่นต้องการใช้เส้นทางนั้น และหากเส้นทางนั้นมีทรัพยากรไม่เพียงพอ ก็จะไปใช้เส้นทางอื่น ๆ ที่มีทรัพยากรเพียงพอกับความต้องการแทน นั่นคือมีการกระจายทราฟฟิกที่ดีแทนที่จะแย่งกันใช้เส้นทางเดียวกัน ยิ่งไปกว่านั้นพบว่าเมื่อทำการเปรียบเทียบความหน่วงเวลาเฉลี่ยของแพ็กเก็ตของอัลกอริทึมที่มีและไม่มีฟังก์ชั่น SetStatus() จะเห็นว่าอัลกอริทึมที่มีฟังก์ชั่น SetStatus() มีความหน่วงเวลามากกว่าอัลกอริทึมที่ไม่มีฟังก์ชั่น SetStatus() เล็กน้อย ทั้งนี้เกิดจากที่เราเตอร์เสียเวลาในการตรวจสอบสถานะของทรัพยากร ทำให้แพ็กเก็ตต้องรอผลคอยการตรวจสอบดังกล่าว ทำให้มีความหน่วงเวลามากขึ้น สรุปได้ว่าฟังก์ชั่น SetStatus() ไม่ทำให้เครือข่ายมีทรูพุทเพิ่มขึ้นแต่อย่างใด

อย่างไรก็ตามการวิธีการที่นำเสนอในงานวิจัยนี้อาจจะมีโอเวอร์เฮดในการตรวจสอบทรัพยากรและทำการจองทรัพยากร ซึ่งเป็นสาเหตุให้วิธีการที่นำเสนอมีความหน่วงเวลาสูงกว่าวิธีการเดิม ถึงแม้จะไม่สูงกว่าอย่างชัดเจนก็ตาม นอกจากนี้การศึกษาถึงสถานะ (State) ของเราเตอร์ว่าสถานะอะไรบ้างเมื่อเปรียบเทียบกับอัลกอริทึมเดิม การที่เราเตอร์ต้องเปลี่ยนสถานะมากกว่า เป็นสาเหตุหนึ่งที่ทำให้เราเตอร์จัดการกับแพ็กเก็ตได้ช้าลง ทำให้แพ็กเก็ตมีความหน่วงเวลามากขึ้น ดังนั้น ในอนาคตจะพยายามลดสถานะที่ไม่จำเป็นออกไป เพื่อให้แพ็กเก็ตมีความหน่วงเวลาน้อยลง

ในการทำวิจัยนี้ได้ใช้เครือข่าย NFSNET-T1 ซึ่งเป็นเครือข่ายที่มีมานาน และเป็นที่รู้จักกันดี แต่เนื่องจากแนวโน้มในอนาคตเครือข่ายอินเทอร์เน็ตจะมีความซับซ้อนมากยิ่งขึ้น ดังนั้นในอนาคตจะทำการทดลองกับเครือข่ายที่มีความซับซ้อนมากยิ่งขึ้นเพื่อศึกษาประสิทธิภาพของอัลกอริทึมที่นำเสนอนี้และปรับปรุงเพื่อใช้ในการส่งข้อมูลมัลติมีเดียผ่านเครือข่ายไร้สายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] R.Braden, L. Zhang, S. Berson, S.Herzog and S.Jamin, "Resource Reservation Protocol (RSVP) Version 1 Function Specification", RFC 2205, Sep, 1997.

[2] R.Braden, D. Clark and S.Shenker, "Integrated Services in The Internet Architecture: An Overview", Internet RFC 1633, Jun. 1994.

[3] S. Blake, D. Black, M. Carson, E.Davies, Z.Wang and W.Weiss, "An Architecture for Differentiated Services", RFC 2475, Dec, 1998.

[4] Y.Chen and R.Hwang, "QoS Routing Algorithms for Multiple Traffic Classes", Communications, 2002. ICC 2002. IEEE International Conference on, Volume: 4, 2002. pp. 2217 -2221

[5] Z.Wang and J.Crowcoft, "Quality of Service Routing for Support Multimedia Applications", IEEE JSAC, Sep, 1996.

[6] X.Xipeng and Lionel M.Ni, "Internet QoS : the Big Picture" [Online]. Available: http://www.cse.msu.edu/~xiaoxipe/papers/msu.inet.qos.bigpicture.pdf. 2002.

[7] W. Sun, "QoS/Policy/Constriant Based Routing" [Online]. Available: http://www.cis.ohio-state.edu/~jain/cis788-99/qos_routing/index.html. 2002.

[8] E.Crawly,R.Nair,B.Jajagopalan and H.Sandick, "A Framework for QoS-based Routing in The Internet", RFC 2386, Aug, 1998.

[9] K. G. Shin and C. Chou, "A Distributed Route-Selection Scheme for Establishing Real-Time Channels", in Proc. High Performance Networking, 1995, pp. 319-330.

[10] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in high-speed networks based on selective probing", in Proc. IEEE local Computer Network (LCN), 1998, pp.80-89.

[11] Shigang Chen, Nahrstedt, K., "Distributed QoS Routing with Imprecise State Information", in Proc. IEEE Int. Conf. Computer Communications and Networks, 1998. pp.614-621.

[12] D.Ghosh, V.Sarangan, and R. Acharya, "Quality-of-Service Routing in IP Networks", IEEE Transactions on Multimedia, Vol. 3, No.2, Jun, 2001. pp.200-208.

[13] D. E. Comer, "Internetworking with TCP/IP", Englewood Cliffs, NJ: Prentice-Hall, vol. 1, Princples, and Architecture, 1995.

[14] V. Srinivas, "IP Quality of Service", Indianapolis: Cisco Press, Dec, 2000.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[15] Floy, S.; Jacobson, "Link-Sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, 3, no. 4, Aug, 1995, pp. 365-386.

[16] Law, A.M. and Kelton, W.David, "Simuation Modeling And Analysis" 2nd New York: McGraw-Hill, 1991.

[17] B. Wood and K. Tumay, "MODSIM III and CACI's Applications" Proceedings of the 1999 Winter Simulation Conference, vol.1, Dec, 1999, pp. 234–238.

[18] Jain, R., "The Art of Computer Systems Performance Analysis" New York: John Wiley & Sons Inc., 1991.

[19] James F. Kurose and Keith W. Ross, "Computer Networking : A Top-Down Approach Featuring the Internet" New York: Addison Wesley Longman., 2001.

[20] P. Ferguson and G. Huston, "Quality of Service Delivering QoS on the Internet and in Corporate Networks" New York: John Wiley & Sons Inc., 1998.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ผลงานที่ได้รับการตีพิมพ์

K. Kesorn and C. Sathiwiriyawong, "An Efficient Call Establishment and Resorce Reservation Mechanisms for Unicast QOS Routing," Proceedings 3rd Information and Computer Engineering Postgraduate Workshop 2003, Jan 30-31, 2003, pp. 33-39.

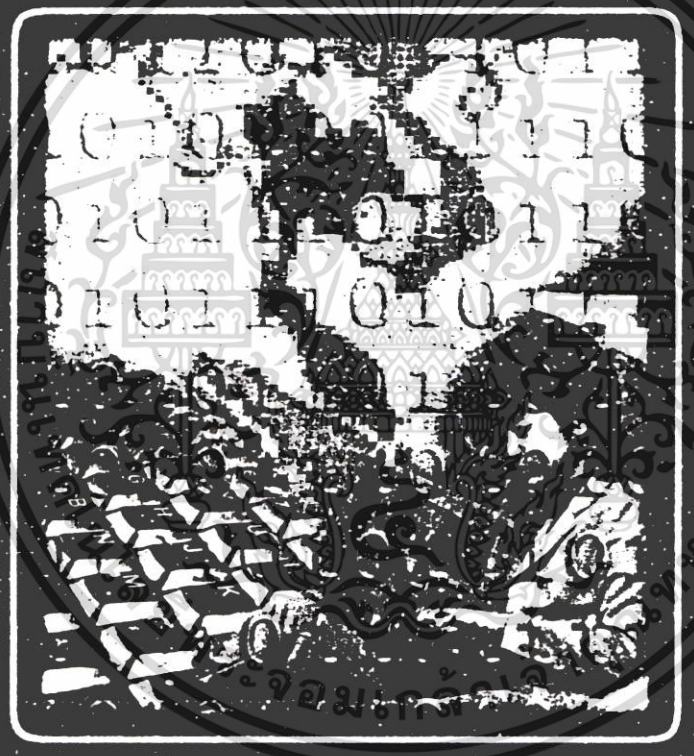


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCEEDINGS

3rd Information and Computer Engineering Postgraduate Workshop 2003

ICEP 2003



30-31 January 2003

Organised and sponsored by
Department of Computer Engineering,
Faculty of Engineering,
Prince of Songkla University, Thailand
http://www.coe.psu.ac.th

In cooperation with
IEEE ComSoc, Thailand Chapter
Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology Association of Thailand (ECTI)

Department of Computer Engineering, Faculty of Engineering,

ISBN: 974-644-418-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AN EFFICIENT CALL ESTABLISHMENT AND RESOURCE RESERVATION MECHANISMS FOR UNICAST QOS ROUTING

K. Kesorn and C. Sathitwiriyawong

Faculty of Information Technology and
Research Center for Communication and Information Technology (ReCCIT),
King Mongkut's Institute of Technology Ladkrabang,
Ladkrabang, Bangkok 10520, Thailand
Email: kraisak@msn.com, chanboon@it.kmitl.ac.th

ABSTRACT

Applications on the Internet have a variety of service requirements. Multimedia applications require high bandwidth and low delay while E-Mail and File-Transfer require only accuracy. Therefore, multimedia applications need the guarantee of service that is called "Quality of Service (QoS)". In recent years, several architectures were proposed to support QoS such as RSVP, Integrated Service, and Differentiated Service. Unfortunately, these architectures do not provide mechanism to find the path that has sufficient resource for QoS requirement. Therefore, QoS routing technique was proposed and it is a key component to build up QoS-based networks. Several QoS routing algorithms were proposed but we found they have some problems such as low call acceptance rate, high call setup time, and high routing overhead. This paper proposes a new QoS routing algorithm that is able to find a suitable path for multimedia traffic. The simulation result shows that the proposed algorithm has higher call acceptance rate, lower call setup time, and lower routing overhead than the existing algorithm.

KEYWORDS

Quality of Service, QoS, QoS routing, multimedia technologies, multimedia networks, routing algorithm, resource reservation

1. INTRODUCTION

Internet applications can be classified into two categories [4]. They are "Bandwidth-guaranteed traffic" that need certain bandwidth and "Best-effort traffic" that is currently used in the Internet. The former has two steps [5] to reserve network resource. The first step is to find a suitable path when the path has enough resource for the application. The second step is to reserve resource on that path. RSVP [1], Integrated Service [2], and Differentiated Service [3] were proposed to reserve network resource. However, they are not a protocol that finds the suitable path in IP network. Furthermore, the current protocols in IP network cannot support QoS requirement efficiently. Therefore, QoS routing was proposed. Basically, QoS-based routing algorithms can be classified into three categories [11]. They are hop-by-hop routing (also called distributed routing), source-based routing, and hierarchical routing.

This research proposes a new QoS routing algorithm based on distributed routing algorithm. In comparison to source routing, this scheme has some advantages of having smaller packet header and lesser memory usage for routing table. Routing overhead includes computation time, storage of routing information, and communication overhead between routers. The source routing algorithm has very high overhead since it has to calculate a very large routing table and frequently updates routing information for the entire network. Therefore, this scheme has low performance and routing information is also imprecise. The distributed routing algorithm can solve the problem from calculate a large routing table by only calculate from local information. One technique was proposed in [8] to solve the imprecise of information. To solve the storage problem, an existing approach was proposed in [9] and was adopted in this paper. It stores routing information of the immediate neighbor router and second-degree neighbor router. The good point of this idea has twofold. First, each router exchanges routing information only between immediate neighbor router and its second-degree neighbor. Therefore, the routing overhead is significantly reduced. The imprecise of routing information is easier to maintain than source routing. Therefore, the routing information is more precise since each router exchanges routing information only with its neighbors. Although this algorithm use "Flooding technique" that was proposed in [6, 7]. We found these techniques have low overhead for call establishment but it has also low call acceptance rate that can reduce the total call throughput. In the next section we describe the new mechanism that can increase the call acceptance rate and reduce the call setup overhead.

Most researches trends to propose multicast algorithms. However, unicast algorithms are still necessary for multimedia systems such a video on demand that sends video streaming to a single receiver who pays for viewing. It is also suitable to

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

send data through routers that are not configured to enable multicast traffic. A unicast flow and homogeneous network, which routers are QoS awareness, are proposed to the scope of this paper.

2. A CALL ESTABLISHMENT MECHANISM

This paper focuses on improving major QoS routing metrics. Three metrics are used to measure routing algorithm performance for multimedia applications [6]. They are the average call acceptance rate (ACAR), average call setup time (ASCT), and average route distance. In this paper uses only two metrics that very important for multimedia applications. They are ACAR and ASCT.

- Average call acceptance rate (ACAR): the probability of accepting a call request, defined as the ratio of number of call accepted to the number of calls arrived in the system.

$$ACAR = \frac{\sum_{i=1}^{N} accepted(R_i)}{N} \quad (1)$$

N = total number of call requests generated.

$accepted(R_i) = 1$ (request ith is accepted)

$accepted(R_i) = 0$ (request ith is rejected)

- Average Call Setup Time (ACST): the average time required to setup connection measured in terms of number of nodes visited by the probe packet.

$$ACST = \frac{\sum_{i=1}^{N} setup(R_i)}{\sum_{i=1}^{N} accepted(R_i)} \quad (2)$$

$setup(R_i)$ = number of nodes visited by the probe packet ith

The first metric is very important as it is a measure of call throughput. The second metric is crucial in the context of multimedia applications that require fast channel setup. The new algorithm comprises two phases: Probing phase and Ack phase.

2.1 Probing phase

This phase consists of the following functions: ProbRsv(), SetStatus(), Forward(), Release(), Backtrack(), and Reject(). They can be explained as follows.

- ProbeRsv(Probe_pkt): This function is used by a source to send a probe packet to its destination. The format of the probe packet is

[k, r, QoS(Bandwidth=B, Delay=D), s ,t, cid, {l}].

where

k = probe packet forwarder
r = probe packet receiver
QoS (Bandwidth=B, Delay=D) = required metrics
s = probe packet sender
t = destination
cid = identification of the probe packet
{l}= list of routers that are visited by the probe packet

- SetStatus(status): This function is used to set the status of network resource. A router reserves resource and waits for an Ack message. The reserved resource is then set to be "wait" status. The advantage of this function is to share the resource with other sources that want to use the resource during the waiting time. After the router receives the Ack message, the status of the resource is set to be "reserve" and is prepared for data transmission.
- Forward(Probe_pkt): This function is used to send probe packet to the receiver along the path that corresponds to routing table. The probe packet is sent directly to the destination if it is found in the routing table. Otherwise, the probe packet is sent by using flooding technique.
- Release(Probe_pkt): This function is used to release the resource that is reserved between k and r .
- Backtrack(Probe_pkt): This function is used to backtrack to router k and sends the probe packet to another path when the resource is unavailable.
- Reject(Probe_pkt): This function is used to discard the duplicated probe packet.

2.2 Ack phase

After the probing phase completes, the destination calls the Ack(Ackl_pkt) function to send an Ack message to the source along the path specified in list {l}. The Ack message is used to confirm that the resource has been successfully reserved. After the source receives the Ack message, it starts to transmit data. Fig. 2.1 shows our flowchart and Fig 2.2 depicts a connection establishment of the new QoS routing mechanism. The sequence of events taking place in Fig. 2.2 is given as follows.

Phase I: Probing Phase

1) To start with source using flooding technique for sending probe packet to R1 and R2.
2) When a router (R1, R2) receives the probe packet, it checks the probe packet with its own probe list. If a duplicated probe packet is received, it is discarded by calling the Reject() function. The identification of the probe packet is stored in the probe list and the admission is called to check available resource for reservation. If there is enough resource then it is reserved and set to "wait" status. The error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

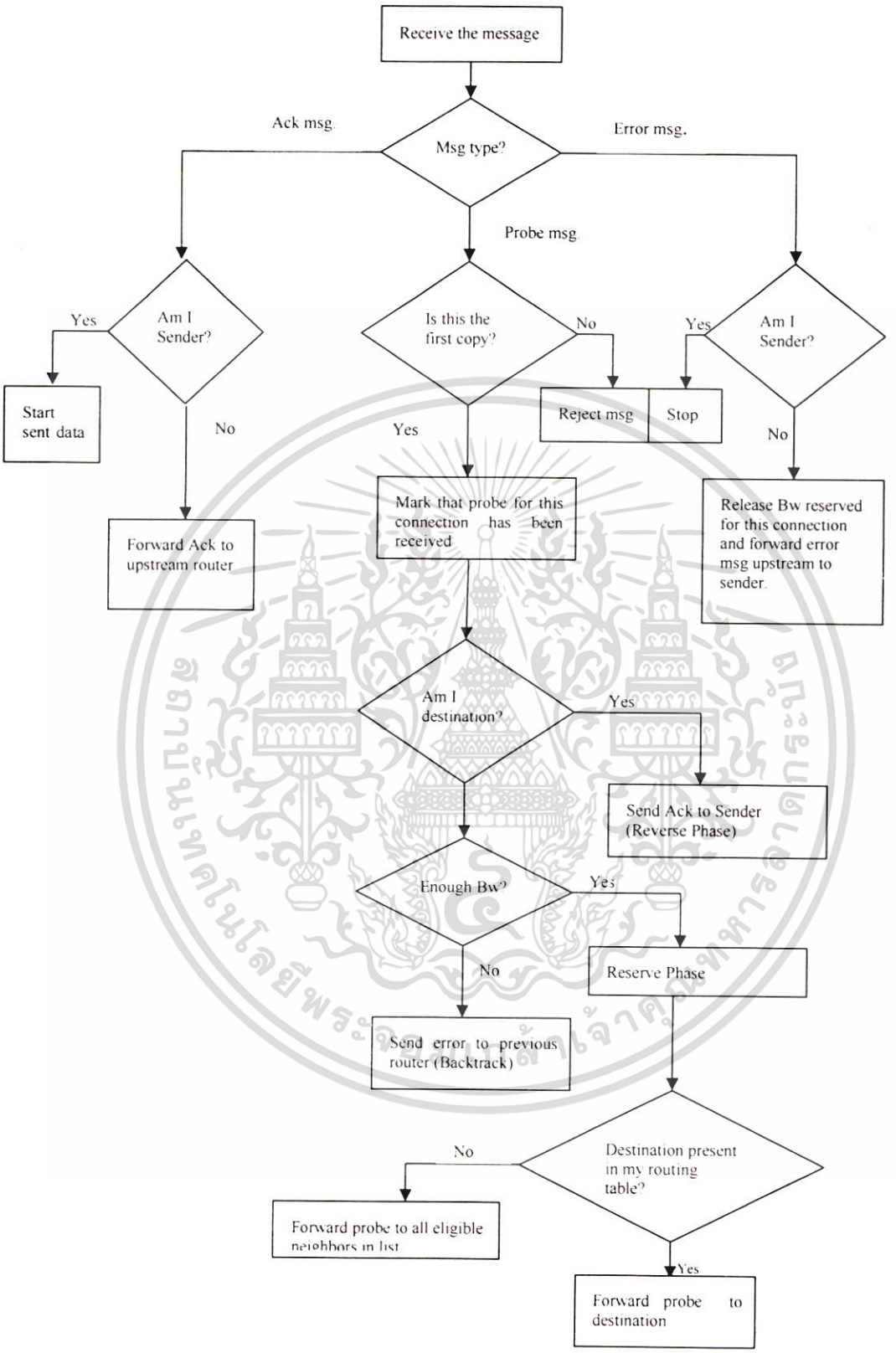


Fig. 2.1 Packet forwarding and packet receiving algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

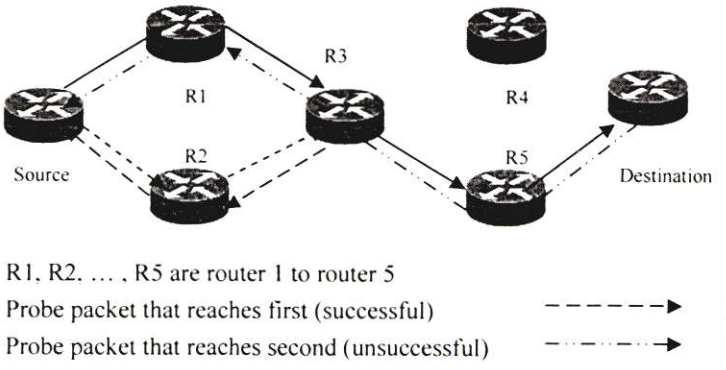


Fig. 2.2 Typical connection establishment of proposed QoS routing algorithm

message is then sent back to the previous router if resource is unavailable. If destination is in routing table, the router will forward the probe packet directly to destination.

3) The probe packet from R1 and R2 reaches R3. In this case probe packet from R1 reaches R3 first. Therefore, R3 rejects probe packet from R2 and sends the error message back to R2 to release the resource.
4) R3 repeats step 2. In this case, the destination is the second-degree neighbor. Therefore, R3 can forward probe packet directly to destination without using flooding technique.

Phase II: Ack phase

When destination receives probe packet. It confirms the source by sending the Ack message along the path that probe packet was visited.

From the above algorithm, its advantages are as follows.

- Each probe packet is identified by its cid. All duplicate probe packets are rejected to ensure that no loops are created.
- Available resource can be checked early at the probing phase and is reserved immediately. Therefore, there is no need to wait for the Ack phase. The error checking time is also less than the existing algorithm.
- The call acceptance rate is also higher than the existing one since the resource is reserved in the probing phase. The chance of being ignored from reserving resource is therefore very low and the chance of failure connection establishment is also low.
- SetStatus() function can prevent bandwidth starvation because it allows other sources to use that link for data transmission while the router is waiting for the reply of the Ack message. Therefore, it increases link utilization.
- Since the number of packets from failure connection establishment is low, less system resource is required.

3. SIMULATION DESIGN

3.1 Simulation Components

In the simulation model, several new components for a router are added i.e., QOSR, RSVF, RM, PC, and CBQ. We select CBQ because it is one of the most promising traffic scheduling algorithms used in future network [10]. The interaction of each component is shown in Fig. 3.1. There are three classes of network traffic: Class A, Class B, and Class C. Class A is for probe packets that have the highest priority. Class B is for multimedia or real-time traffic. Class C is for best-effort traffic. Traffic with the higher priority is served before a lower one in accordance with the CBQ scheduling algorithm.

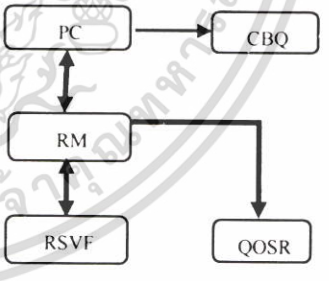


Fig. 3.1 Interaction between components

Interactions between components are briefly explained as follows.

- Packet Classifier (PC): Since traffic can be prioritized to different classes, the corresponding traffic scheduling should also support this classification as well. PC is used to classify incoming packet. If there is a requirement for resource reservation, PC will call resource management (RM). When it

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

receives the result from RM. PC puts the packet in the suitable queue (CBQ).

- Class Based Queuing (CBQ): This component is used to manage queue of packets in each class and forward packets through a suitable link to the next router.
- Resource Management (RM): If an incoming packet asks for resource reservation, RM will check for available resource. RSVF is used to reserve resource and it returns result of reservation back to QOSR.
- Resource Reservation Function (RSVF): This component is used to reserve resource on receiving request from RM. Since resource is reserved, it sends the result back to RM.
- Quality of Service Routing algorithm (QOSR): This algorithm is used for finding a suitable path. If the resource reservation is succeeded, QOSR will find a suitable path to the next router or destination. On the other hand, if the resource reservation fails, QOSR sends error message to the previous router.

3.2 Simulation Model

To investigate the effect of routing algorithms on multimedia network performance, a simulator for the study of all aspects of routing algorithms was developed. The simulator was written in an object-oriented simulation language [12] using a process-oriented approach to discrete-event simulation modeling. The simulation language is able to model large and complex systems. The simulation uses NSFNET-T1 topology as shown in Fig. 3.2. Each link has a capacity of 1.544 Mb/s with full duplex transmission. The call requests are exponential distributed. Bandwidth requirement of a call is uniform distributed between its respective minimum and maximum values. The source and destination are randomly picked up as a uniformly distribution as well.



Fig. 3.2 NSFNET-T1 Topology

4. SIMULATION RESULTS

In this section, some simulation results of the proposed algorithm are presented to examine routing performance. The purpose is to verify the correctness and the efficiency of the proposed algorithm. The simulation for performance measurement can be classified to many cases. We study the effect of workload (call arrival rate). The simulation result is shown in Fig. 4.1.

ACAR is decreased while increasing values of workload. This is because more workload consumes more resources, thus resulting in more call getting dropped. Therefore, the percent of call acceptance rate is low. The new algorithm has higher call acceptance rate than the existing algorithm.

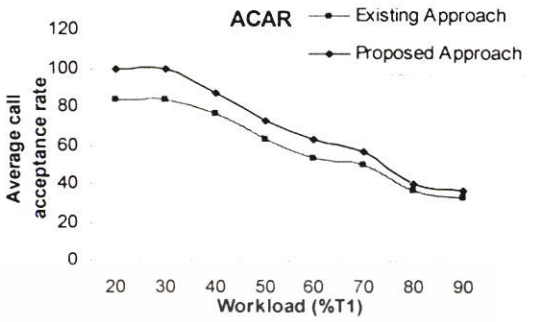


Fig. 4.1 Effect of workload on ACAR

Since resource reservation in probing phase can increase call acceptance rate, ACAR is improved. At higher loads, more and more calls attempt to use the fixed available resources. Also, it can be observed that the ACST increase with workload. This is because at higher workload, finding a qualified path involves more searches in the network. In Fig 4.2, the existing approach shows that the numbers of routers that receive probe packets are still the same when workload is increased. When the acceptance rate is lesser, ACST is higher. The new algorithm checks for required resource before forwarding the probe packet. If there is not enough resource along the path, the probe packet is discarded. Therefore, the number of routers that receive probe packet is smaller than the existing algorithm.

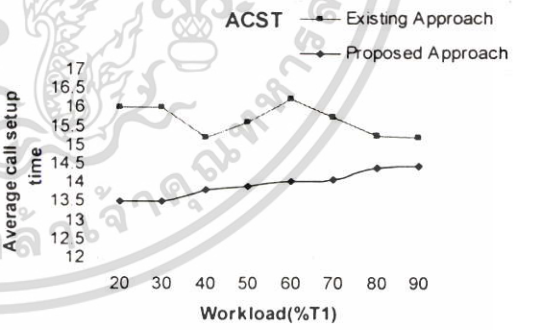


Fig. 4.2 Effect of workload on ACST

In Fig. 4.3, the effect of varying call setup time is studied. The simulation shows that ACAR is improved when the call setup time is increased. With equal call setup time, the proposed algorithm gives higher ACAR than the existing algorithm since the time used to establish connection is lower. The proposed algorithm has longer time to find path that has sufficient resource. Therefore, the acceptance rate is higher. Fig. 4.4 shows that

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

the proposed algorithm has smaller ACST when the call setup time is increased as explained in Fig 4.2.

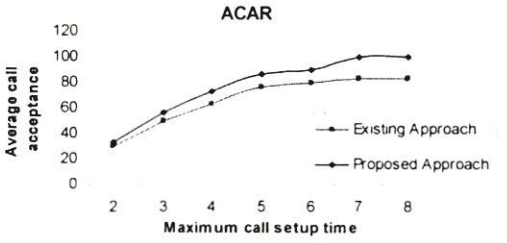


Fig. 4.3 Effect of call setup time on ACAR

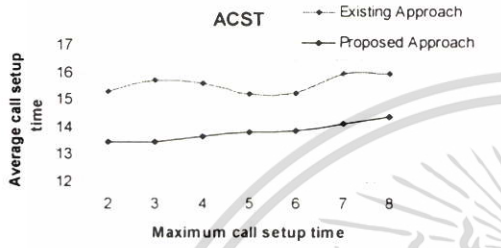


Fig. 4.4 Effect of call setup time on ACST

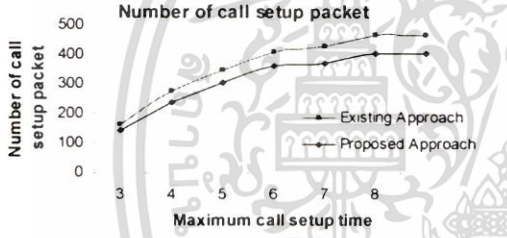


Fig. 4.5 Effect of call setup time on average number of packet in network

When the call setup time is increased, the number of call setup packets is also increased since the probe packet will traverse a longer path as shown in Fig. 4.5. For the existing algorithm, the number of routing packets is higher because it generates more error messages when the resource cannot be reserved.

5. CONCLUSIONS

In this paper, we propose a new QoS routing algorithm for unicast multimedia networks. The proposed algorithm attempts to improve both performance metrics (call acceptance rate, call setup time). Our simulation result shows that the resource reservation in the probing phase offers higher call acceptance rate than in the Ack phase. Furthermore, the number of packets from unsuccessful connection establishment is reduced. Therefore, less system resource is required. However, the proposed algorithm has some overhead from the resource checking and reservation processes. Nowadays, hardware is developed for higher performance and faster speed. Therefore, the effect of this problem is negligible. In the future, we consider a larger system that has about 20 to 30 hops. Furthermore, we will desire to develop an algorithm to support multicast routing and QoS routing for wireless networks.

REFERENCES

[1] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, *Resource Reservation Protocol (RSVP) Version 1: Function Specification, RFC 2205*, Sept. 1997.

[2] R. Braden, D. Clark and S. Shenker, *Integrated Services in the Internet Architecture: An Overview, RFC 1633*, Jun. 1994.

[3] S. Blake, D. Black, M. Carson, E. Davies, Z. Wang and W. Weiss, *An Architecture for Differentiated Services, RFC 2475*, Dec.1998.

[4] Y. Chen and R. Hwang, "QoS Routing Algorithms for Multiple Traffic Classes," *IEEE International Conference on Communications (ICC), Volume 4*, pp. 2217-2221, 2002.

[5] Z. Wang and J. Crowcoft, "Quality of Service Routing for Support Multimedia Applications," *IEEE JSAC*, Sept.1996.

[6] K. G. Shin and C. Chou, "A Distributed Route-Selection Scheme for Establishing Real-Time Channels," *Proceedings High Performance Networking*, pp. 319-330, 1995.

[7] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in High-speed Networks based on Selective Probing," *Proceedings IEEE Local Computer Network (LCN)*, pp. 80-89, 1998.

[8] S. Chen and K. Nahrstedt, "Distributed QoS Routing with Imprecise State Information," *Proceedings IEEE International Conference on Computer Communications and Networks*, pp 614-621, 1998.

[9] D. Ghosh, V. Sarangan, and R. Acharya, "Quality-of-Service Routing in IP Networks," *IEEE Transactions on Multimedia, Vol. 3, No.2*, pp.200-208, Jun. 2001.

[10] S. Floyd and V. Jacobson, "Link-Sharing and Resource Management Models for Packet Networks," *IEEE/ACM Transactions on Networking, vol. 3, no. 4*, pp.365-386, Aug 1995.

[11] Shigang Chen, "Routing Support for Providing Guaranteed End-to-End Quality-of-Service," *Ph.D. thesis*, UIUC, May 1999.

[12] B. Wood and K. Tumay, "MODSIM III and CACI's Applications," *Proceedings of the 1999 Winter Simulation Conference, vol.1*, pp. 234–238, December 1999.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTHOR BIOGRAPHY

KRAISAK KESORN was born in Pitsanulok and received the B.Sc. in computer science from Chiangmai University, Chiangmai, Thailand, in 1998. Now, he studies in Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok, Thailand. kraisak@msn.com

CHANBOON SATHIWIRIYAWONG received an M.Sc. in Data Telecommunications and Networks and a Ph.D. in Electronic and Electrical Engineering (Computer Networks) from the University of Salford, UK in 1993 and 1996, respectively. Since then, he has been a lecturer at the Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang. He joined Multimedia and Virtual Research Laboratory, Research Center for Communication and Information Technology (ReCCIT), as a researcher in 1997, and as an assistant to laboratory leader in 1999. His research interest includes multimedia network performance, and multimedia security. chanboon@it.kmitl.ac.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

นายไกรศักดิ์ เกษร เกิดเมื่อวันที่ 29 กรกฎาคม 2519 จังหวัดพิษณุโลก จบจากโรงเรียนพิษณุโลกพิทยาคม ปี พ.ศ. 2537 และเข้ารับการศึกษา ณ มหาวิทยาลัยเชียงใหม่ ในปีเดียวกัน และสำเร็จการศึกษาจากมหาวิทยาลัยเชียงใหม่ คณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ วุฒิวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ปี พ.ศ.2540 หลังจากนั้นได้เข้าทำงานในตำแหน่งอาจารย์พิเศษ ประจำโปรแกรมวิชาคอมพิวเตอร์ สถาบันราชภัฏพิบูลสงคราม จังหวัดพิษณุโลก เป็นเวลา 2 ปี จากนั้นได้เข้าศึกษาต่อ ระดับปริญญาโท คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปี พ.ศ. 2543 และในปี 2544 ได้รับทุนพัฒนาอาจารย์สาขาขาดแคลนของมหาวิทยาลัยนเรศวรและทุนการศึกษาของมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร (Computer & Communication Education Foundation) ให้ทุนสนับสนุนในการศึกษาเป็นระยะเวลา 2 ปี และเข้าบรรจุเป็นพนักงานของมหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก ตำแหน่งอาจารย์ สังกัดคณะวิทยาศาสตร์ ภาควิชาคณิตศาสตร์ ในปี 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้