วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธในระบบเครือข่ายแบบคำนึงถึงคุณภาพในการให้บริการ

CALL REQUEST DELAYING METHOD FOR QoS AWARE NETWORK

กิตติศักดิ์ องค์ไพบูลย์
KITTISAK ONGPAIBOOL

พ.ศ. 2543
ISBN 974-15-2531-1

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธในระบบเครือข่ายแบบคำนึงถึงคุณภาพในการให้บริการ

CALL REQUEST DELAYING METHOD FOR QoS AWARE NETWORK

กิตติศักดิ์ องค์ไพบูลย์

KITTISAK ONGPAIBOOL

วน.
ก675อ
2549

เลขหมู่.....
เลขทะเบียน.....61666
วัน,เดือน,ปี.....19 ก.ค. 2549

.b.....11601127.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2549

ISBN 974-15-2531-1

CALL REQUEST DELAYING METHOD FOR QoS AWARE NETWORK

KITTISAK ONGPAIBOOL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN INFORMATION ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

ISBN 974-15-2531-1

COPYRIGHT 2006

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธในระบบเครือข่ายแบบคำนึงถึงคุณภาพในการให้บริการ
นักศึกษา	นายกิตติศักดิ์ องค์ไพบูลย์
รหัสนักศึกษา	46061318
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.มยุรี เลิศเวชกุล

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ได้นำเสนอกลไกเพิ่มเติมสำหรับโปรแกรมควบคุมการอนุมัติการเชื่อมต่อและเรียกการทำงานแบบนี้ว่า วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ (Call Request Delaying Method: CRDM) ซึ่งวิธีนี้สามารถที่จะลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ เวลาเฉลี่ยในการรอคอยที่จะทำการเชื่อมต่อกับเครือข่าย และทราฟฟิกที่เกิดจากความพยายามจัดตั้งการเชื่อมต่อในเครือข่ายแบบรับประกันคุณภาพในการให้บริการ โดยเฉพาะในสภาวะที่มีทรัพยากรไม่เพียงพอแก่การให้ผู้ใช้งานทำการเชื่อมต่อได้ วิธี CRDM ได้ออกแบบมาเพื่อควบคุมการร้องขอจัดตั้งการเชื่อมต่อที่เข้ามาใหม่แทนที่จะตอบปฏิเสธไปในทันที โดยจะทำการหน่วงเวลาการร้องขอไว้ในคิวรอการเชื่อมต่อเพื่อรอการปล่อยแบนด์วิดธ์คืนกลับมายังเครือข่ายจากผู้ใช้งานที่ใช้เสร็จแล้ว ซึ่งจะสามารถทำการประเมินประสิทธิภาพได้โดยใช้ทราฟฟิกที่มีกำหนดเวลาในการเชื่อมต่อที่แน่นอน โดยในวิทยานิพนธ์นี้จะทำการเพิ่มประสิทธิภาพของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธโดยใช้วิธีการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลาซึ่งเมื่อได้ทำการทดลองหาค่าประสิทธิภาพปรากฏว่ามีผลอัตราปฏิเสธการร้องขอที่ดีขึ้นกว่าวิธีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่างประมาณ 56.232 % ซึ่งวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธนี้สามารถที่จะลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ ลดทราฟฟิกที่เกิดจากความพยายามจัดตั้งการเชื่อมต่อ และลดเวลาเฉลี่ยในการรอคอยที่จะทำการเชื่อมต่อกับเครือข่ายเมื่อได้จำลองการทำงานและเปรียบเทียบบนเครือข่ายแบบ IntServ over DiffServ ที่ยังไม่มีการปรับปรุง

Thesis Title	Call Request Delaying Method for QoS aware Network
Student	Mr. Kittisak Ongpaibool
Student ID	46061318
Degree	Master of Engineering
Programme	Information Engineering
Year	2006
Thesis Advisor	Asst. Prof. Mayuree Lertwetchakul

ABSTRACT

This thesis proposes a new theme of call admission control mode called "Call Request Delaying Method" (CRDM) that could improve call blocking rate, call attempt waiting time and call admission control signaling traffic in QoS network, especially in saturated situation. The CRDM was designed to handle new incoming call requests instead of reject them when the request bandwidth is still not available. The call request holding time for a new released bandwidth could be estimated by traffic descriptor of the present active connections. We also improved the original CRDM by add delay technique on resource allocation process. The modified CRDM has shown its performance on call blocking rate reduction approximate 56.232 % from the CRDM while induce a little bit more on waiting time delay. By applying this delaying method, we can show the improvement of call blocking rate, call attempt waiting time and call admission control signaling traffic through simulation results compared to the traditional call admission control (CAC) model on IntServ over DiffServ framework.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ.มยุรี เลิศเวชกุล ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณ ดร.พิทักษ์ ธรรมวาริน และ รศ.ดร.ชวลิต เบญจางคประเสริฐ กรรมการสอบหัวข้อและโครงร่างวิทยานิพนธ์ ที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะจนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบพระคุณอาจารย์ทุกท่านที่กรุณาให้การอบรมสั่งสอนและให้ความรู้เสมอมา

สุดท้ายต้องขอขอบคุณเพื่อนร่วมงาน พี่ ๆ น้อง ๆ ในห้องปฏิบัติการเครือข่ายคอมพิวเตอร์ (Computer Network Lab) ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้ความช่วยเหลือแนะนำในการทำวิจัยให้สำเร็จลุล่วงด้วยดีและทุกท่านที่ไม่ได้กล่าวถึงในที่นี้ที่ให้ความช่วยเหลือและกำลังใจในการทำวิทยานิพนธ์ฉบับนี้จนสำเร็จ

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดาซึ่งเป็นที่รักและเคารพยิ่งตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ให้วิชาความรู้และถ่ายทอดประสบการณ์ที่ดีแก่ข้าพเจ้า

กิตติศักดิ์ องค์ไพบูลย์

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VIII

สารบัญรูป IX

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของวิทยานิพนธ์ 3

1.3 สมมุติฐานของการศึกษา 3

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย 4

1.5 ขอบเขตของวิทยานิพนธ์ 4

1.6 ขั้นตอนการทำวิทยานิพนธ์ 5

1.7 โครงสร้างของวิทยานิพนธ์ 5

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 6

2.1 ส่วนควบคุมการอนุมัติการเชื่อมต่อ (Call Admission Control: CAC) 6

2.1.1 องค์ประกอบพื้นฐานของส่วนควบคุมการอนุมัติการเชื่อมต่อ 6

2.1.1.1 ลักษณะทราฟฟิก (Traffic Descriptor) 7

2.1.1.2 กระบวนการวัดค่า (Measurement Process) 7

2.1.1.3 เงื่อนไขการอนุมัติ (Admission Criteria) 7

2.1.2 การพัฒนาส่วนควบคุมการอนุมัติการเชื่อมต่อ 7

2.1.2.1 ส่วนควบคุมการอนุมัติการเชื่อมต่อบนทุก ๆ เราเตอร์ตามเส้นทาง 7

2.1.2.2 ส่วนควบคุมการอนุมัติการเชื่อมต่อจากส่วนกลาง 8

2.1.2.3 ส่วนควบคุมการอนุมัติการเชื่อมต่อผ่านตัวแทนของระบบเครือข่าย 9

สารบัญ(ต่อ)

หน้า

2.1.2.4 ส่วนควบคุมการอนุมัติการเชื่อมต่อที่มีการตัดสินใจ
โดยการวัดค่าพารามิเตอร์ 10

2.2 สถาปัตยกรรมการรับประกันคุณภาพในการให้บริการ
(Quality of Service Architectures) 11

2.2.1 นิยามของคุณภาพในการให้บริการ 11

2.2.2 The Integrated Services Architecture 11

2.2.2.1 โมเดล Integrated Services 12

2.2.2.1.1 ส่วนควบคุมการอนุมัติและนโยบาย 12

2.2.2.1.2 ส่วนแยกประเภทของแพ็กเกต 13

2.2.2.1.3 ส่วนจัดลำดับแพ็กเกต 13

2.2.2.2 โพรโตคอลสำรองทรัพยากร (Resource Reservation Protocol) 13

2.2.2.3 คลาสของบริการ (Classes of Service) 15

2.2.3 Differentiated Services 15

2.2.3.1 DiffServ (DS) field 16

2.2.3.2 Per-Hop Behavior (PHB) 17

2.2.3.2.1 Class Selector PHB 18

2.2.3.2.2 Expedited Forwarding PHB 18

2.2.3.2.3 Assured Forwarding PHB 18

2.2.4 Integrated Services over Differentiated Services 19

2.2.4.1 สถาปัตยกรรมพื้นฐาน 19

2.2.4.2 กระบวนการควบคุมการจัดตั้งการเชื่อมต่อ 20

2.2.4.3 กระบวนการทำงานของ Trunk 21

บทที่ 3 วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ 24

3.1 การทำงานของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ
(Call Request Delaying Method: CRDM) 25

3.1.1 ส่วนควบคุมการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธใน
สภาวะปกติ 25

สารบัญ(ต่อ)

หน้า

3.1.2 ส่วนควบคุมการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธใน
สภาวะอิ่มตัว 25
3.1.3 การจัดสรรแบนด์วิดธ์ 28
3.1.3.1 การจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง 28
3.1.3.2 การจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา 29
3.2 การสร้างระบบเครือข่ายบนโปรแกรมจำลองระบบเครือข่าย 30
3.3 การสร้างโมเดลจำลองของอุปกรณ์ต่าง ๆ ในระบบเครือข่าย 31
3.3.1 กลุ่มผู้ใช้งาน (LANs) 31
3.3.1.1 การทำงานของโมดูล usr_gen 32
3.3.1.2 การทำงานของโมดูล queue 35
3.3.1.3 การทำงานของโมดูล proc 37
3.3.2 เราเตอร์ขอบ (Edge Router) 38
3.3.2.1 การทำงานของโมดูล proc 39
3.3.3 เราเตอร์หลัก (Core Router) 41
3.3.3.1 การทำงานของโมดูล proc 42
3.3.4 เซิร์ฟเวอร์ (Server) 43
3.3.4.1 การทำงานของโมดูล proc 44

บทที่ 4 การทดลองและผลการทดลอง 46
4.1 การทดลอง 46
4.1.1 ค่าพารามิเตอร์ที่ใช้ในการวัดประสิทธิภาพ 48
4.1.2 ลำดับในการทดลอง 48
4.2 ผลการทดลอง 49
4.2.1 ผลการทดลองที่ 1 49
4.2.2 ผลการทดลองที่ 2 54

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 61

สารบัญ(ต่อ)

หน้า

เอกสารอ้างอิง 63

ภาคผนวก 65

ภาคผนวก ก. สมการการสุ่มที่ใช้ในโปรแกรมจำลองระบบเครือข่าย 66

ภาคผนวก ข. ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่ 68

ประวัติผู้เขียน 74

สารบัญตาราง

ตารางที่ หน้า

2.1 DiffServ (DS) codepoint 17

2.2 Assured Forwarding (AF) Classes 18

2.3 VIP table 20

2.4 Trunk table 21

2.5 Provisioned Link table 22

3.1 ผลการทดลองเมื่อใช้ช่วงคาบเวลาในการจัดสรรแบนด์วิดธ์ 29

4.1 พารามิเตอร์ของทราฟฟิกแต่ละประเภทใช้ในการทดลอง 46

4.2 การเปรียบเทียบผลการทดลองหาช่วงคาบเวลาที่เหมาะสมในการจัดสรรแบนด์วิดธ์ 55

สารบัญรูป

รูปที่ หน้า

2.1 ความสัมพันธ์ขององค์ประกอบพื้นฐานกับส่วนควบคุมการอนุมัติจัดตั้งการเชื่อมต่อ 6

2.2 Implement Model 12

2.3 IntServ/RSVP Model 14

2.4 Host-Router model in RSVP 14

2.5 DiffServ Model 16

2.6 DiffServ (DS) codepoint 17

2.7 IntServ over DiffServ Architecture 20

2.8 TRRequest / TRRelease / TRConfirm Message Format 22

2.9 TRRequest Message Process Algorithm 23

2.10 TRRelease Message Process Algorithm 23

3.1 การทำงานของ CRDM ในสภาวะปกติ 25

3.2 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีผู้ร้องขอปฏิเสธ 26

3.3 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่ 1 27

3.4 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่ 2 27

3.5 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่รอนานเกินกว่าเวลาในการรอคอย 28

3.6 เครือข่าย IntServ over DiffServ บนโปรแกรมจำลองระบบเครือข่าย 30

3.7 เราเตอร์หลักแต่ละตัวที่รองรับการเชื่อมต่อจากเราเตอร์ขอบและกลุ่มผู้ใช้งาน 31

3.8 กลุ่มผู้ใช้งานในโปรแกรมจำลองระบบเครือข่าย 31

3.9 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในกลุ่มผู้ใช้งาน 32

3.10 โครงสร้างการทำงานของ state ภายในโมดูล usr_gen 33

3.11 โครงสร้างการทำงานของ QoS_usr Child Process ที่ถูกเรียกขึ้นมาใช้งาน 33

3.12 ลักษณะการทำงานของโมดูล usr_gen 34

3.13 โครงสร้างการทำงานของ state ภายในโมดูล queue 36

3.14 ลักษณะการทำงานของโมดูล queue 37

3.15 โครงสร้างการทำงานของ state ภายในโมดูล proc 37

3.16 ลักษณะการทำงานของโมดูล proc 38

3.17 เราเตอร์ขอบในโปรแกรมจำลองระบบเครือข่าย 38

3.18 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเราเตอร์ขอบ 39

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.19 โครงสร้างการทำงานของ state ภายในโมดูล proc 39

3.20 ลักษณะการทำงานของโมดูล proc 40

3.21 เราเตอร์หลักในโปรแกรมจำลองระบบเครือข่าย 41

3.22 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเราเตอร์หลัก 42

3.23 โครงสร้างการทำงานของ state ภายในโมดูล proc 43

3.24 ลักษณะการทำงานของโมดูล proc 43

3.25 เซิร์ฟเวอร์ในโปรแกรมจำลองระบบเครือข่าย 44

3.26 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเซิร์ฟเวอร์ 44

3.27 โครงสร้างการทำงานของ state ภายในโมดูล proc 44

3.28 ลักษณะการทำงานของโมดูล proc 45

4.1 โครงสร้างของเครือข่ายที่ใช้ในการทดลอง 47

4.2 โครงสร้างการเชื่อมต่อระหว่างผู้ใช้งาน เราเตอร์ขอบ และเราเตอร์หลัก 47

4.3 กราฟอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ 49

4.4 กราฟเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อ 50

4.5 กราฟปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อ 52

4.6 กราฟประสิทธิภาพการใช้งานแบนด์วิดธ์ 53

4.7 กราฟเวลาในการตอบสนองเฉลี่ย 54

4.8 กราฟแนวโน้มของการหาช่วงคาบเวลาที่เหมาะสมในการจัดสรรแบนด์วิดธ์ 56

4.9 กราฟอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ 57

4.10 กราฟเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อ 58

4.11 กราฟปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อ 58

4.12 กราฟประสิทธิภาพการใช้งานแบนด์วิดธ์ 59

4.13 กราฟเวลาในการตอบสนองเฉลี่ย 60

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบเครือข่ายคอมพิวเตอร์ได้เริ่มพัฒนาขึ้นมาตั้งแต่ในทศวรรษที่ 90 และได้เริ่มเข้ามามีบทบาทแก่บริษัทหรือองค์กรขนาดใหญ่ซึ่งมีวัตถุประสงค์หลักในการใช้ข้อมูล โปรแกรม และอุปกรณ์ต่าง ๆ ร่วมกัน จนในปัจจุบันระบบเครือข่ายคอมพิวเตอร์ได้มีบทบาทอย่างมากในการให้บริการแก่ผู้ใช้คอมพิวเตอร์ส่วนบุคคลที่บ้าน โดยวัตถุประสงค์ในการเชื่อมต่อเข้ากับเครือข่ายจะแตกต่างกันไป เช่น การสื่อสารระหว่างบุคคลหรือกลุ่มบุคคล การรับบริการข้อมูลข่าวสารจากระยะไกล ความบันเทิงส่วนบุคคล เป็นต้น เมื่อมีจำนวนผู้ใช้งานที่มากขึ้นอย่างรวดเร็ว ผลที่ตามมาก็คือ ปัญหาในการให้บริการในช่วงที่มีความต้องการในการเชื่อมต่อกันเครือข่ายที่พร้อม ๆ กัน จำนวนแบนด์วิดธ์ (Bandwidth) ที่ให้บริการ และคุณภาพในการให้บริการ (Quality of Service: QoS) เป็นต้น จึงเกิดความคิดที่จะให้มีการรับประกันคุณภาพในการให้บริการแก่เครือข่ายเกิดขึ้น

เทคโนโลยีของสื่อบันเทิงได้รับการพัฒนาไปมากทั้งกระบวนการประมวลผลและการแสดงผลแบบต่อเนื่องของสื่อวิดิทัศน์ (Video) และเสียง (Audio) ซึ่งสามารถนำไปใช้เป็นส่วนหนึ่งของโปรแกรมประยุกต์ (Application) เช่น การเรียนทางไกล (Distance Learning) การประชุมผ่านวิดิทัศน์ (Video-Conference) และวิดิทัศน์ตามอุปสงค์ (Video On Demand: VOD) เป็นต้น ซึ่งโปรแกรมประยุกต์เหล่านี้มีลักษณะโดยธรรมชาติที่ต้องการการตอบสนองสูงในการที่จะใช้สื่อบันเทิงเหล่านี้ในการสื่อสาร แต่โปรแกรมประยุกต์เหล่านี้มีความต้องการให้การสื่อสารที่แตกต่างกัน จึงต้องมีการปรับระดับตัวแปรของการให้บริการ เช่น เวลาแฝง (latency) แบนด์วิดธ์ และการชนกันของแพ็กเกต เป็นต้น นอกจากนั้นยังมีความต้องการที่จะรับประกันระดับในการให้บริการของสื่อวิดิทัศน์ดิจิตอลและเสียงด้วย

การสื่อสารผ่านระบบเครือข่ายต้องการมาตรฐานที่เป็นรูปแบบเพื่อใช้กับการสื่อสารที่มีความหลากหลายต่าง ๆ กัน เช่น วิดิทัศน์ เสียง เว็บบราวเซอร์ และจดหมายอิเล็กทรอนิกส์ (E-mail) เป็นต้น ซึ่งได้มีหลายสถาปัตยกรรมได้เสนอที่จะรวมมาตรฐานของการสื่อสารที่จำเป็นต่าง ๆ ไว้ด้วยกันในเครือข่ายเดียว เช่น เครือข่ายเอทีเอ็ม (Asynchronous Transfer Mode: ATM) เครือข่ายการสื่อสารร่วมระบบดิจิตอล (Integrated Services Digital Network: ISDN) และเครือข่ายระบบเคเบิล เป็นต้น โดยมีความพยายามที่จะรวมมาตรฐานของการสื่อสารที่จำเป็นทั้งหมดเข้าด้วยกันในเครือข่ายเดียว ต่อมาคณะทำงานจึงได้มีความคิดเห็นเป็นข้อตกลงร่วมกันคือ “ถ้ามีการรวมมาตรฐานของการสื่อสารทั้งหลายไว้ด้วยกัน รูปแบบนั้นก็จะเป็นไปเพื่ออินเทอร์เน็ต” ระบบเครือข่ายอินเทอร์เน็ต

โพรโตคอลจึงได้รับการพัฒนาจากหลายงานวิจัย ในเริ่มแรกระบบเครือข่ายจะสามารถรองรับเฉพาะบริการแบบ best–effort ต่อมาได้มีการพัฒนาให้ระบบเครือข่ายอินเทอร์เน็ตโพรโตคอลมีการรับประกันคุณภาพในการให้บริการด้วย ซึ่งมีการพัฒนาไปในหลายรูปแบบ โดยคณะทำงาน IETF (Internet Engineering Task Force) ได้กำหนดสถาปัตยกรรมในการรับประกันคุณภาพในการให้บริการแรกขึ้นมาคือ The Integrated Services Architecture (IntServ) [1] ซึ่งเป็นการรับประกันคุณภาพในการให้บริการตั้งแต่ต้นทางไปจนถึงปลายทางของแต่ละการเชื่อมต่อ แต่สถาปัตยกรรมนี้มีข้อด้อยคือ ต้องใช้สัญญาณในการควบคุมคุณภาพในการให้บริการที่ทุก ๆ เราเตอร์ทำให้ไม่เหมาะที่จะใช้กับเครือข่ายที่มีขนาดใหญ่ได้ จึงได้มีการพัฒนาสถาปัตยกรรม Differentiated Services Architecture (DiffServ) [6] เพื่อแก้ไขปัญหาจากสถาปัตยกรรมที่แล้ว โดยในสถาปัตยกรรมนี้ได้เสนอที่จะรวมทุกการเชื่อมต่อไว้ด้วยกันแล้วให้บริการในระดับที่แตกต่างกันไปตามประเภทของบริการนั้น ๆ แต่ปัญหาของสถาปัตยกรรมนี้คือ ไม่มีกลไกส่วนควบคุมการอนุมัติการเชื่อมต่อ (Call Admission Control) จึงทำให้ไม่สามารถควบคุมจำนวนผู้ใช้งานและนโยบายของทราฟฟิกได้ ต่อมาจึงได้มีความต้องการที่จะพัฒนาสถาปัตยกรรมเครือข่ายโดยได้มีการรวมเอาข้อดีของสถาปัตยกรรม The Integrated Services Architecture และ Differentiated Services Architecture เอาไว้ด้วยกันคือ สถาปัตยกรรม The Integrated Services over Differentiated Services Architecture (IntServ over DiffServ) ซึ่งเป็นสถาปัตยกรรมที่กำลังได้รับการพัฒนาขึ้นอยู่ในขณะนี้โดย IntServ จะทำงานอยู่ที่เราเตอร์ขอบของเครือข่าย และ DiffServ จะทำงานอยู่ที่เราเตอร์หลักของเครือข่าย ซึ่งในวิทยานิพนธ์นี้ได้เลือกเครือข่าย IntServ over DiffServ ที่พัฒนาโดย Mei Yang [15] มาเป็นต้นแบบในการพัฒนา

จากที่ได้กล่าวมา ส่วนสำคัญที่จะสามารถทำให้ระบบเครือข่ายสามารถรับประกันคุณภาพในการให้บริการได้ก็คือ ส่วนควบคุมการอนุมัติจัดตั้งการเชื่อมต่อ ซึ่งจะคอยทำหน้าที่ควบคุมจำนวนผู้ใช้งานที่จะมาเชื่อมต่อกับเครือข่าย จำนวนการใช้แบนด์วิดธ์ และตัวแปรที่ใช้ในการรับประกันคุณภาพในการให้บริการ

เมื่อมีการร้องขอจัดตั้งการเชื่อมต่อใหม่จากผู้ใช้งานเข้ามาสู่เครือข่าย ส่วนควบคุมการอนุมัติการเชื่อมต่อจะทำการตรวจสอบทรัพยากรทั้งหมดที่ยังคงเหลืออยู่ในเครือข่าย ถ้าพบว่ายังมีทรัพยากรที่สามารถจัดสรรให้แก่ผู้ใช้งานได้ก็จะส่งแพ็กเกต (Packet) ตอบยอมรับให้ผู้ใช้งานทำการเชื่อมต่อและส่งข้อมูลได้ แต่ถ้าตรวจสอบแล้วพบว่าไม่มีทรัพยากรเพียงพอที่จะสามารถจัดสรรให้แก่ผู้ใช้งานก็จะส่งแพ็กเกตตอบปฏิเสธกลับไปยังผู้ใช้งานในทันที ซึ่งสิ่งนี้จะมีผลต่อค่าอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ (Blocking Rate) โดยทางด้านผู้ใช้งานก็จะต้องทำการส่งแพ็ก-เกตร้องขอเข้ามายังเครือข่ายเรื่อย ๆ จนกว่าจะสามารถเชื่อมต่อกับเครือข่ายได้ ซึ่งมีบางผู้ใช้งานต้องทำการร้องขอจัดตั้งการเชื่อมต่อหลายครั้งจึงจะสามารถทำการเชื่อมต่อได้ แต่ในขณะที่บางผู้ใช้งาน

ทำการร้องขอเพียงไม่กี่ครั้งก็สามารถทำการเชื่อมต่อได้ ทำให้เกิดความไม่สะดวกในการส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อและเกิดความไม่เสมอภาคขึ้นมา

โดยปกติแล้วหลาย ๆ งานวิจัยที่ผ่านมา [13 - 22] ได้มีการรับประกันคุณภาพในการให้บริการแก่ผู้ใช้งานเมื่อได้ทำการเชื่อมต่อกับเครือข่ายอย่างสมบูรณ์แล้ว ซึ่งในวิทยานิพนธ์ฉบับนี้มีความต้องการที่จะรับประกันคุณภาพในการให้บริการแก่ผู้ใช้งานตั้งแต่เริ่มส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อมายังเครือข่าย โดยเฉพาะในสภาวะที่เครือข่ายไม่มีทรัพยากรเพียงพอที่จะอนุมัติให้ผู้ใช้ทำการเชื่อมต่อได้ ซึ่งความท้าทายก็คือ ยังไม่มีงานวิจัยใดได้ทำการรับประกันคุณภาพในการให้บริการในจุดนี้

1.2 วัตถุประสงค์ของวิทยานิพนธ์

วิจัยและศึกษาการรับประกันคุณภาพในการให้บริการและส่วนควบคุมการอนุมัติการเชื่อมต่อที่ทำงานบนระบบเครือข่ายอินเทอร์เน็ตโพรโตคอล เพื่อสร้างกลไกใหม่ของส่วนควบคุมการอนุมัติการเชื่อมต่อเพื่อใช้ในการรับประกันคุณภาพในการให้บริการให้แก่ผู้ใช้งานตั้งแต่เริ่มส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อมายังเครือข่าย โดยกลไกใหม่ที่สร้างขึ้นมาจะหน่วงเวลาปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อเมื่อเครือข่ายอยู่ในสภาวะที่มีทรัพยากรไม่เพียงพอที่จะรับการเชื่อมต่อจากผู้ใช้งานเพิ่มได้ แล้วเมื่อเครือข่ายมีทรัพยากรเพียงพอให้สามารถทำการเชื่อมต่อได้ ส่วนควบคุมอนุมัติการเชื่อมต่อจะทำการส่งแพ็กเกตอนุญาตให้ผู้ใช้งานทำการเชื่อมต่อได้เลย ซึ่งจะทำการทดลองหาวิธีที่เหมาะสมในการลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ ลดเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อ และลดปริมาณทราฟฟิกจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อ นอกเหนือจากนี้แล้วจะทำการวัดค่าตัวแปรอื่น ๆ ที่มีผลกระทบต่อเครือข่ายเมื่อได้ใช้กลไกใหม่กับเครือข่ายด้วย

1.3 สมมุติฐานของการศึกษา

เมื่อได้สร้างกลไกใหม่ให้แก่ส่วนควบคุมอนุมัติการเชื่อมต่อเพื่อใช้กับระบบเครือข่ายอินเทอร์เน็ตโพรโตคอล โดยการหน่วงเวลาปฏิเสธการจัดตั้งการเชื่อมต่อ ซึ่งจะเป็นการเพิ่มโอกาสในการได้รับการจัดสรรทรัพยากรและสามารถเชื่อมต่อกับเครือข่ายได้ ทำให้สามารถลดอัตราปฏิเสธการจัดตั้งการเชื่อมต่อได้ และผู้ใช้งานไม่ต้องทำการสร้างแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อแล้วส่งมาซ้ำ ๆ ผลที่ตามมาก็คือ จะสามารถลดจำนวนทราฟฟิก (Traffic) ที่เกิดจากการส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อซ้ำ ๆ ได้

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

โดยปกติแล้วเมื่อผู้ใช้งานต้องการจะสื่อสารผ่านเครือข่ายหนึ่ง ๆ ก็จะต้องทำการสร้างแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อแล้วส่งมาร้องขอที่เครือข่าย ซึ่งในเครือข่ายอินเทอร์เน็ตโพรโตคอลที่มีการรับประกันคุณภาพในการให้บริการจะมีส่วนควบคุมการอนุมัติการเชื่อมต่อ ซึ่งจะทำการตรวจสอบทรัพยากรทั้งหมดของเครือข่าย ถ้าเครือข่ายมีทรัพยากรเพียงพอที่จะสามารถรองรับการเชื่อมต่อจากผู้ใช้งานเพิ่มได้ ก็จะทำการสร้างแพ็กเกตตอบยอมรับแล้วส่งกลับไปยังผู้ที่ร้องขอมาทำให้สามารถสร้างการเชื่อมต่อและส่งข้อมูลได้ต่อไป แต่ถ้าส่วนควบคุมการอนุมัติการเชื่อมต่อตรวจสอบแล้วพบว่ามีทรัพยากรไม่เพียงพอที่จะให้ผู้ร้องขอทำการเชื่อมต่อได้ ก็จะทำการสร้างแพ็กเกตตอบปฏิเสธแล้วส่งกลับไปยังผู้ร้องขอในทันที และถ้าผู้ที่ร้องขอต้องการที่จะติดต่อสื่อสารผ่านเครือข่ายให้ได้ ผู้ร้องขอจะต้องทำการสร้างแพ็กเกตร้องขอขึ้นมาใหม่แล้วส่งไปยังเครือข่ายเพื่อทำการสร้างการเชื่อมต่อซ้ำ ๆ จนกว่าจะได้รับการตอบยอมรับจากเครือข่าย

สภาวะที่เครือข่ายมีทรัพยากรเพียงพอที่จะสามารถรองรับผู้ใช้งานให้ทำการเชื่อมต่อเพิ่มได้จะเรียกว่า “สภาวะปกติ” ส่วนสภาวะที่เครือข่ายมีทรัพยากรไม่เพียงพอที่จะรองรับการเชื่อมต่อจากผู้ใช้งานเพิ่มได้จะเรียกว่า “สภาวะอิ่มตัว” จากปัญหาที่กล่าวมาข้างต้นนี้ ในสภาวะที่เครือข่ายอิ่มตัวซึ่งผู้ใช้งานต้องการที่จะสื่อสารส่งข้อมูลผ่านเครือข่ายแล้วต้องทำการสร้างและส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อซ้ำ ๆ จนกว่าจะได้รับแพ็กเกตตอบยอมรับ ในวิทยานิพนธ์ฉบับนี้ได้มีแนวคิดที่ต้องการลดความยุ่งยากในการร้องขอจัดตั้งการเชื่อมต่อ โดยทำการหน่วงเวลาแพ็กเกตร้องขอไว้ในคิวรอการเชื่อมต่อ แทนที่จะปฏิเสธแพ็กเกตร้องขอนั้นไปในทันที ซึ่งสามารถลดทราฟฟิกที่เกิดจากการร้องขอซ้ำ ๆ และลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อได้ จึงได้เสนอ “วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ” เพื่อเป็นกลไกในการทำงานของส่วนควบคุมการอนุมัติการเชื่อมต่อในสภาวะที่เครือข่ายอิ่มตัว

1.5 ขอบเขตของวิทยานิพนธ์

1. ศึกษาวิจัยการทำงานระบบเครือข่ายแบบ IntServ over DiffServ บนโปรแกรมจำลองการทำงานระบบเครือข่าย OPNET
2. เสนอกลไกใหม่ของส่วนควบคุมอนุมัติการร้องขอจัดตั้งการเชื่อมต่อบนระบบเครือข่ายด้วยวิธีการหน่วงเวลาปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ
3. เสนอวิธีการจัดสรรแบนด์วิดธ์ของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่างและการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา

4. วัดผลการจำลองโดยใช้ตัวแปรที่มีผลกระทบต่อเครือข่ายเมื่อได้ใช้วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ เช่น อัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ เวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อ ทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อ ประสิทธิภาพการใช้งานแบนด์วิดธ์ และเวลาเฉลี่ยในการตอบสนอง โดยจะเปรียบเทียบกับผลการจำลองที่วัดได้จากระบบเครือข่ายที่ยังไม่ได้ปรับปรุง

1.6 ขั้นตอนการทำวิทยานิพนธ์

1. กำหนดจุดประสงค์ หัวข้อ และขอบเขตการทำวิทยานิพนธ์
2. ศึกษาทฤษฎี หลักการพื้นฐาน ข้อมูล และแนวคิดที่จะนำมาใช้ในการทำวิทยานิพนธ์
3. ศึกษาและพัฒนาวิธีการสร้างกลไกของส่วนควบคุมการอนุมัติจัดตั้งการเชื่อมต่อ
4. ออกแบบและสร้างโมเดลจำลองการทำงานของระบบเครือข่ายอินเทอร์เน็ตโพรโตคอลแบบ IntServ over DiffServ บนโปรแกรมจำลองการทำงานระบบเครือข่าย OPNET
5. ออกแบบและสร้างโมเดลของกลไกส่วนควบคุมการอนุมัติการเชื่อมต่อแบบใหม่โดยผสานเข้ากับส่วนควบคุมการอนุมัติจัดตั้งการเชื่อมต่อเดิมของเครือข่ายอินเทอร์เน็ตโพรโตคอลแบบ IntServ over DiffServ บนโปรแกรมจำลองการทำงานระบบเครือข่าย OPNET แล้วทำการทดลองเพื่อหาค่าประสิทธิภาพของกลไกใหม่ที่ได้พัฒนาขึ้น
6. วิเคราะห์ผลการทดลอง สรุปและนำเสนอแนวทางในการทำวิทยานิพนธ์ต่อไป
7. จัดทำเอกสารประกอบวิทยานิพนธ์

1.7 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของวิทยานิพนธ์ ขั้นตอนการทำวิทยานิพนธ์ และโครงสร้างของวิทยานิพนธ์

บทที่ 2 กล่าวถึงสถาปัตยกรรมระบบเครือข่ายอินเทอร์เน็ตโพรโตคอลที่ใช้ในการรับประกันคุณภาพในการให้บริการ และการพัฒนาส่วนควบคุมการอนุมัติการเชื่อมต่อ

บทที่ 3 กล่าวถึงวิธีการทำงานของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ และการพัฒนาบนโปรแกรมจำลองระบบเครือข่าย OPNET

บทที่ 4 กล่าวถึงการทดลอง ผลการทดลอง และการวิเคราะห์ผลการทดลองที่ได้มา

บทที่ 5 บทสรุปผลการวิจัยและข้อเสนอแนะ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

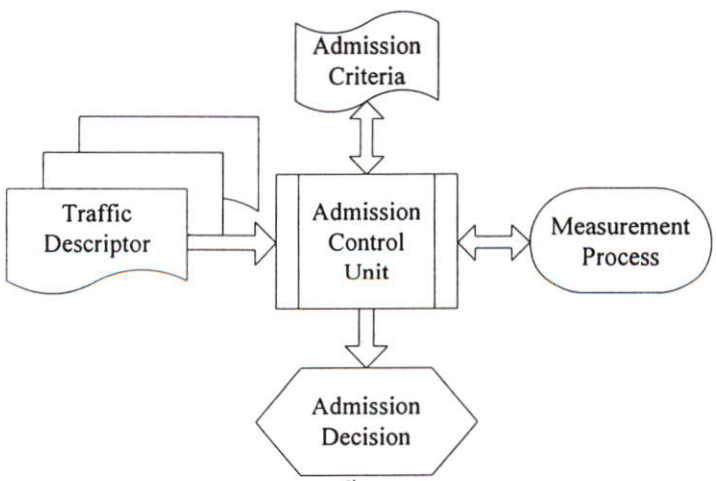
2.1 ส่วนควบคุมการอนุมัติการเชื่อมต่อ (Call Admission Control: CAC)

บางแอพพลิเคชันต้องการการรับประกันคุณภาพในการให้บริการจากระบบเครือข่ายซึ่งความต้องการคุณภาพในการให้บริการนี้อาจจะอยู่ในเทอมของแบนด์วิดธ์ขั้นต่ำหรือปริมาณการสูญหายของแพ็กเกตสูงที่สุดที่ยอมรับได้ ซึ่งเครือข่ายจะต้องมีอุปกรณ์ที่จะรองรับการเชื่อมต่อและสามารถจัดสรรหรือดูแลแบนด์วิดธ์ที่มีอยู่อย่างจำกัดของเครือข่ายให้สามารถบริการแก่การเชื่อมต่อที่ดำเนินการอยู่ตามระดับคุณภาพการให้บริการที่ตกลงกันเอาไว้ดังนั้นอุปกรณ์นี้อาจจะทำการปฏิเสธการเชื่อมต่อใหม่ในกรณีที่ทรัพยากรที่มีอยู่อาจจะไม่เพียงพอและการให้บริการแก่การเชื่อมต่อใหม่นั้นอาจเป็นสาเหตุให้เกิดผลกระทบต่อคุณภาพในการให้บริการแก่การเชื่อมต่อที่มีอยู่แล้วโดยกระบวนการตัดสินใจที่จะยอมรับหรือปฏิเสธการเชื่อมต่อที่มีการร้องขอต่อระบบเครือข่ายนี้ถูกเรียกว่า ส่วนควบคุมการอนุมัติการเชื่อมต่อ

โดยมากแล้วขั้นตอนการตัดสินใจยอมรับหรือปฏิเสธการร้องขอทรัพยากรของส่วนควบคุมการอนุมัติการเชื่อมต่อมักจะทำการตัดสินใจอย่างง่าย ๆ จากผลการคำนวณผลรวมของจำนวนแบนด์-วิดธ์ที่ใช้อยู่กับจำนวนแบนด์วิดธ์ที่การเชื่อมต่อใหม่ต้องการและถ้าผลรวมที่ได้มีค่ามากกว่าจำนวนแบนด์วิดธ์ทั้งหมดของเครือข่าย ก็จะทำการปฏิเสธการเชื่อมต่อนั้น

2.1.1 องค์ประกอบพื้นฐานของส่วนควบคุมการอนุมัติการเชื่อมต่อ

ส่วนควบคุมการอนุมัติการเชื่อมต่อจะมีองค์ประกอบที่สำคัญสามส่วนคือ ลักษณะทราฟฟิกเงื่อนไขในการอนุมัติ และกระบวนการวัดค่าพารามิเตอร์ ซึ่งทั้ง 3 องค์ประกอบจะมีความสัมพันธ์ตามรูปที่ 2.1



รูปที่ 2.1 ความสัมพันธ์ขององค์ประกอบพื้นฐานกับส่วนควบคุมการอนุมัติการเชื่อมต่อ

2.1.1.1 ลักษณะทราฟฟิก (Traffic Descriptor)

ลักษณะทราฟฟิกจะเป็นกลุ่มของค่าพารามิเตอร์ซึ่งเป็นคุณลักษณะของทราฟฟิกต้นกำเนิด เช่น ลักษณะทราฟฟิกของโทเก้นบัคเก็ต (token bucket) จะมีอัตราการใส่โทเก้นเท่ากับ r และมีขนาดของบัคเก็ตเท่ากับ b ซึ่งถ้าใช้โทเก้นบัคเก็ตเป็นแหล่งกำเนิดทราฟฟิก จะส่งจำนวนทราฟฟิกเท่ากับ r*t+b ตลอดช่วงระยะเวลา t ถ้าโทเก้นบัคเก็ตจะมีอัตราการส่งสูงสุดเท่ากับ p จะทำให้ระยะเวลาระหว่างแต่ละแพ็กเกตจะเท่ากับ 1/p

2.1.1.2 กระบวนการวัดค่าพารามิเตอร์ (Measurement Process)

อัลกอริทึมของส่วนควบคุมการอนุมัติการเชื่อมต่อต้องการค่าพารามิเตอร์เป็นอินพุท ซึ่งจะใช้ค่าพารามิเตอร์ที่ได้จากกระบวนการวัดเพื่อไปใช้ในการกำหนดเงื่อนไขในการอนุมัติการเชื่อมต่อ โดยส่วนใหญ่แล้วจะใช้จำนวนแบนด์วิดธ์ของเครือข่าย อัตราการมาถึงของแพ็กเกตเฉลี่ยของแพ็กเกต และความคับคั่งของข้อมูลในระบบเครือข่าย เป็นต้น

2.1.1.3 เงื่อนไขการอนุมัติ (Admission Criteria)

เงื่อนไขการอนุมัติคือ กฎซึ่งส่วนควบคุมการอนุมัติจัดตั้งการเชื่อมต่อที่ใช้ในการตอบรับหรือตอบปฏิเสธการร้องขอ เพื่อจัดสรรแบนด์วิดธ์ให้แก่ผู้ใช้งาน โดยการตัดสินใจของส่วนควบคุมการอนุมัติการเชื่อมต่อจะกระทำบนพื้นฐานของการประเมินผลกระทบที่มีต่อการเชื่อมต่ออื่น ๆ รวมถึงประสิทธิภาพการใช้งานแบนด์วิดธ์ของเครือข่ายด้วย

2.1.2 การพัฒนาส่วนควบคุมการอนุมัติการเชื่อมต่อ

โดยการพัฒนาส่วนควบคุมการอนุมัติการเชื่อมต่อจากนักวิจัยหลายท่านที่ผ่านมา จะมีแนวทางไปในด้านการลดปัญหาเรื่องขนาดของสถาปัตยกรรมแบบ IntServ และการแก้ปัญหาความเชื่อถือได้จากการรับประกันคุณภาพในการให้บริการของสถาปัตยกรรมแบบ DiffServ ซึ่งสามารถจำแนกได้เป็น 4 ประเภทตามลักษณะการควบคุมได้แก่ (1) ส่วนควบคุมการอนุมัติการเชื่อมต่อบนทุก ๆ เราเตอร์ตามเส้นทาง (2) ส่วนควบคุมการอนุมัติการเชื่อมต่อจากส่วนกลาง (3) ส่วนควบคุมการอนุมัติการเชื่อมต่อผ่านตัวแทนของระบบเครือข่าย และ (4) ส่วนควบคุมการอนุมัติการเชื่อมต่อที่มีการตัดสินใจโดยการวัดค่าพารามิเตอร์

2.1.2.1 ส่วนควบคุมการอนุมัติการเชื่อมต่อบนทุก ๆ เราเตอร์ตามเส้นทาง

ในประเภทนี้จะมีส่วนควบคุมการอนุมัติการเชื่อมต่อที่ทุก ๆ เราเตอร์ตามเส้นทางที่ต้องการจะสื่อสาร โดยในแต่ละเราเตอร์จะมีสิทธิ์ขาดในการตัดสินใจที่จะยอมรับหรือไม่ในการให้บริการแก่ผู้ที่ร้องขอ ซึ่งวิธีนี้จะสามารถรับประกันคุณภาพในการให้บริการได้อย่างดี

ก. สัญญาณโพรโตคอลสำรองทรัพยากร (RSVP Signaling)

โพรโตคอลสำรองทรัพยากร [2] [3] คือ สัญญาณที่ใช้ในการสร้างและบำรุงรักษาการสำรองทรัพยากรในเครือข่าย IntServ โดยมีวัตถุประสงค์เพื่อต้องการทรัพยากรและสำรองไว้ที่แต่ละเราเตอร์ตามเส้นทางที่จะไปสู่ปลายทาง โดยจะมีการทำงานดังนี้ (1) ผู้ส่งจะส่ง PATH message ไปยังผู้รับเพื่อบอกความต้องการคุณภาพในการให้บริการของการเชื่อมต่อและแต่ละเราเตอร์ตามเส้นทางที่จะต้องส่งต่อไปจนถึงปลายทาง (2) เมื่อได้รับ PATH message ผู้รับจะส่ง RESV message กลับไปยังผู้ส่งเพื่อทำการสำรองทรัพยากรของเครือข่ายไว้สำหรับการเชื่อมต่อ โดยแต่ละเราเตอร์ตามเส้นทางสามารถยอมรับหรือปฏิเสธการร้องขอนี้ถ้าไม่มีทรัพยากรเพียงพอ ถ้าเราเตอร์ปฏิเสธการร้องขอ ก็จะส่ง error message ไปยังผู้รับ และการติดต่อโดยใช้สัญญาณการสำรองทรัพยากรนี้จะสิ้นสุดลง แต่ถ้าตอบยอมรับเราเตอร์ก็จะมีการสำรองทรัพยากรและแบนด์วิดธ์ไว้ให้สำหรับการเชื่อมต่อนั้น

2.1.2.2 ส่วนควบคุมการอนุมัติการเชื่อมต่อจากส่วนกลาง

ในประเภทนี้จะมีการใช้ส่วนควบคุมการอนุมัติการเชื่อมต่อจากส่วนกลาง ซึ่งจะได้รับการติดต่อจากเราเตอร์ขอบอีกทีนึง โดยในประเภทนี้จะเป็นการพัฒนาเพื่อลดการใช้สัญญาณในการดูแลรักษาสถานะการรับประกันคุณภาพในการให้บริการ

ก. ส่วนควบคุมการอนุมัติการเชื่อมต่อที่มีส่วนต่อรองแบนด์วิดธ์

(Bandwidth Broker based Admission Control)

ส่วนต่อรองแบนด์วิดธ์ (Bandwidth Broker: BB) [17] จะเป็นการเอาสัญญาณการสำรองทรัพยากรที่เราเตอร์หลักจากสถาปัตยกรรมแบบ IntServ ออก โดยจะมีการเก็บและจัดการข้อมูลที่ศูนย์กลาง ซึ่งส่วนต่อรองแบนด์วิดธ์จะต้องเป็นเราเตอร์หรือซอฟต์แวร์แพ็กเกจที่ได้ติดตั้งไว้ในเราเตอร์/สวิตช์ของเครือข่าย

โมดูลหลักของส่วนต่อรองแบนด์วิดธ์คือ ส่วนควบคุมการอนุมัติการเชื่อมต่อและการหาเส้นทาง โดยการทำงานคือจะดูแลรักษาสถานะของคุณภาพในการให้บริการของเครือข่ายและรับผิดชอบเกี่ยวกับส่วนควบคุมการอนุมัติการเชื่อมต่อและการสำรองทรัพยากร โดยภายหลังจากการตัดสินใจที่จะยอมรับการเชื่อมต่อโดยพิจารณาตลอดเส้นทางไปจนถึงผู้รับแล้ว ส่วนต่อรองแบนด์วิดธ์จะใส่ข้อมูลลงในฐานข้อมูลเกี่ยวกับโครงสร้างของเครือข่าย การเชื่อมต่อ และสถานะการควบคุมคุณภาพในการให้บริการตามเส้นทางในทุก ๆ โหนด ซึ่งโดยปกติแล้วในหนึ่งเครือข่ายจะมีส่วนต่อรองแบนด์วิดธ์หนึ่งตัว

2.1.2.3 ส่วนควบคุมการอนุมัติการเชื่อมต่อผ่านตัวแทนของระบบเครือข่าย

ในประเภทนี้จะมีส่วนควบคุมการอนุมัติการเชื่อมต่อโดยใช้ตัวแทนของระบบเครือข่ายซึ่งส่วนใหญ่แล้วจะเป็นเราเตอร์ตัวแรกที่ผู้ร้องขอได้ส่งมายังเครือข่ายหรือที่เรียกว่า เราเตอร์ขอบขาเข้า โดยในเราเตอร์นี้จะมีอำนาจเต็มในการตัดสินใจที่จะรับหรือไม่รับการเชื่อมต่อจากผู้ร้องขอ และในประเภทนี้จะมีการพัฒนาเพื่อแก้ปัญหาจากสถาปัตยกรรม DiffServ โดยไม่มีการใช้สัญญาณในการรักษาคุณภาพในการให้บริการที่เราเตอร์หลักของเครือข่าย

ก. สถานะการเปลี่ยนแปลงของแพ็กเกต (Dynamic Packet State: DPS)

เทคนิคสถานะการเปลี่ยนแปลงของแพ็กเกต [18] คือ ข้อมูลสถานะของการเชื่อมต่อที่ถูกใส่เข้าไปที่ส่วนหัวของแพ็กเกต (เช่นอัตราการสำรองแบนด์วิดธ์) ซึ่งจำเป็นสำหรับสัญญาณการเชื่อมต่อและการจัดการสถานะของการควบคุมคุณภาพ โดยที่เราเตอร์ขอบขาเข้าจะเริ่มต้นจัดเตรียมข้อมูลสถานะ และที่เราเตอร์หลักจะทำการประมวลผลแพ็กเกตที่เข้ามาบนพื้นฐานของสถานะที่มากับแพ็กเกต และจะทำการอับเดทแล้วใส่ไว้ที่ส่วนหัวของแพ็กเกตแล้วส่งไปยังเราเตอร์ตัวถัด ไป

ในเทอมของส่วนควบคุมการอนุมัติการเชื่อมต่อจะใช้สัญญาณโพรโตคอลสำรองทรัพยากรเพื่อสื่อสารกันระหว่างผู้ส่งและผู้รับ แต่สัญญาณโพรโตคอลสำรองทรัพยากรนี้จะทำงานกับเราเตอร์ขอบเท่านั้น เริ่มแรกที่เราเตอร์ขอบขาเข้าเมื่อได้รับ PATH message ก็จะส่งตรงต่อไปยังเราเตอร์ขอบขาออกเลย ซึ่งเมื่อเราเตอร์ขอบขาออกได้รับ RESV message แล้วก็จะส่งตอบกลับมายังเราเตอร์ขอบขาเข้าเลย และเมื่อแต่ละโหนดระหว่างเส้นทางของการสื่อสารได้รับสัญญาณโพรโตคอลสำรองทรัพยากรนี้ก็จะทำการตรวจสอบอัตราการสำรองทรัพยากรที่โหนดนั้น ๆ ด้วย โดยจะใช้วิธีการคำนวณอย่างง่าย ๆ [18] และเมื่อการเชื่อต่อสิ้นสุดลงก็จะทำการปล่อยแบนด์วิดธ์กลับสู่เครือข่าย

ข. การรวมสัญญาณ IntServ (Aggregation in IntServ)

การรวมสัญญาณ [7] เป็นกลไกที่ใช้ในการลดจำนวนสัญญาณที่ส่งโดยสถาปัตยกรรมแบบ IntServ โดยส่วนควบคุมการอนุมัติการเชื่อมต่อในเทคนิคนี้จะกระทำการรวมกลุ่มของสัญญาณของแต่ละการเชื่อมต่อและรวมสัญญาณที่จำเป็นในการดูแลรักษาสถานะการสำรองทรัพยากรที่แต่ละเราเตอร์หลักด้วย

นอกเหนือจากนั้น จะมีส่วนควบคุมการอนุมัติการเชื่อมต่อที่เราเตอร์ขอบขาเข้าซึ่งจะทำการตัดสินใจบนพื้นฐานของจำนวนแบนด์วิดธ์ โดยจะอนุญาตให้ทำการปรับเปลี่ยนการสำรองทรัพยากรได้ แต่เราเตอร์ขอบขาเข้าจะใช้เวลาในการปรับการสำรองทรัพยากรที่เราเตอร์หลักมากกว่าวิธีการสำรองทรัพยากรของสถาปัตยกรรมแบบ IntServ

2.1.2.4 ส่วนควบคุมการอนุมัติการเชื่อมต่อที่มีการตัดสินใจโดยการวัดค่าพารามิเตอร์

ในประเภทนี้ส่วนใหญ่จะมีส่วนควบคุมการอนุมัติการเชื่อมต่อที่เราเตอร์ขาออกโดยจะมีการวัดค่าพารามิเตอร์ตามคุณภาพในการให้บริการที่ร้องขอมาจากแพ็กเกตร้องขอ ซึ่งจะนำผลที่ได้จากการวัดค่าพารามิเตอร์ไปใช้ในการตัดสินใจที่จะยอมรับหรือไม่ยอมรับการเชื่อมต่อนั้น

ก. ส่วนควบคุมการอนุมัติการเชื่อมต่อที่มีการวัดค่าที่เราเตอร์ขาออก (Measurement-Based Admission Control in Egress Router)

ส่วนควบคุมการอนุมัติการเชื่อมต่อในวิธีนี้ [19] [20] จะทำการตัดสินใจที่เราเตอร์ขาออกเท่านั้นโดยไม่มีการดูแลรักษาสถานะตลอดการเชื่อมต่อที่เราเตอร์หลักหรือที่เราเตอร์ขาออก โดยในการตัดสินใจจะทำบนพื้นฐานของการวัดค่าที่เราเตอร์ขาออกซึ่งเทคนิคหลักคือ การวัดบริการที่สามารถใช้ได้ตลอดเส้นทางตั้งแต่ต้นทางไปจนถึงปลายทางโดนใช้ระบบโมเดลกล่องดำ ซึ่งจะทำการวัดหรือควบคุมค่าตัวแปรที่มีผลกระทบต่อทราฟฟิกอื่น ที่เราเตอร์ขาออกจะทำการคำนวณอัตราการมาถึงและบริการที่ให้ได้ต่ำสุดและจะทำการกำหนดอัลกอริทึมของส่วนควบคุมการอนุมัติการเชื่อมต่อออกมาเพื่อทำการรับหรือปฏิเสธการจัดตั้งการเชื่อมต่อ ซึ่งถ้าบริการขั้นต่ำที่สามารถให้ได้มีเพียงพอและการรับประกันคุณภาพในการให้บริการมีครบตามความต้องการของการเชื่อมต่อนั้นก็จะทำการยอมรับและให้ทำการเชื่อมต่อได้และจะต้องไม่มีผลกระทบต่อการเชื่อมต่ออื่น ๆ ที่ได้ยอมรับไปแล้วในก่อนหน้านี้

ข. การตรวจผลของส่วนควบคุมการอนุมัติการเชื่อมต่อที่จุดปลายทาง (End-Point Admission Control through Probing)

ในกลไกนี้ [21] [22] จะทำการอนุมัติการเชื่อมต่อใหม่ที่โฮสต์ปลายทางหรือเราเตอร์ขาออกหรือเราเตอร์ขาเข้าโดยผ่านการวิเคราะห์ความคับคั่งของเครือข่ายในแต่ละเส้นทางของการเชื่อมต่อ โดยก่อนที่จะอนุญาตให้มีการเชื่อมต่อใหม่ได้ ผู้ส่งที่ส่งแพ็กเกตมาตามลักษณะทราฟฟิกที่ได้ร้องขอไว้จะถูกตรวจสอบค่าตัวแปรอัตราการสูญหายของแพ็กเกตและค่าเวลาหน่วงเป็นต้นที่ผู้รับ เพื่อทำการตรวจสอบระดับความคับคั่งของเครือข่าย ถ้ามีประสิทธิภาพตามคุณภาพในการให้บริการที่ร้องขอมาก็จะสามารถรับการเชื่อมต่อใหม่นี้ได้ แต่ถ้าประสิทธิภาพลดต่ำลงก็จะทำการปฏิเสธการเชื่อมต่อใหม่นี้ซึ่งกลไกนี้จะนำไปไว้ที่จุดปลายทางทำให้ไม่ต้องใช้สัญญาณโพรโตคอลหรือฟังก์ชันพิเศษที่จะส่งไปยังเราเตอร์หลักหรือเราเตอร์ขอบ

2.2 สถาปัตยกรรมการรับประกันคุณภาพในการให้บริการ (Quality of Service Architectures)

สถาปัตยกรรมที่สำคัญซึ่งกำหนดโดยคณะทำงานอินเทอร์เน็ต (Internet Engineering Task Force: IETF) เพื่อให้บริการในระดับที่แตกต่างกันของการเชื่อมต่ออินเทอร์เน็ตโพรโตคอล ซึ่งสามารถแบ่งตามการทำงานออกได้เป็น 3 ประเภท

- *The Integrated Services Architecture* เป็นการสำรองทรัพยากรตั้งแต่ตั้งทางไปจนถึงปลายทาง
- *The Differentiated Services Architecture* เป็นการรับประกันให้แต่ละแพ็กเกตที่แตกต่างกันไปตามประเภทของแพ็กเกตนั้น
- *Integrated Services over Differentiated Services* เป็นการรวมข้อดีของทั้งสองสถาปัตยกรรมไว้ด้วยกัน

2.2.1 นิยามของคุณภาพในการให้บริการ

คุณภาพในการให้บริการจะเป็นข้อตกลงระหว่างผู้ใช้บริการและผู้ให้บริการ ซึ่งข้อตกลงนั้นจะอยู่ในรูปของตัวแปรที่สามารถวัดค่าประสิทธิภาพได้ เช่น การสูญหายของข้อมูล (data loss) เวลาหน่วงของเครือข่ายสูงสุด (maximum network delay) ความแปรปรวนต่ำสุด (minimal jitter) แบนด์วิดธ์ต่ำสุด/สูงสุด (minimal/maximum bandwidth) เป็นต้น

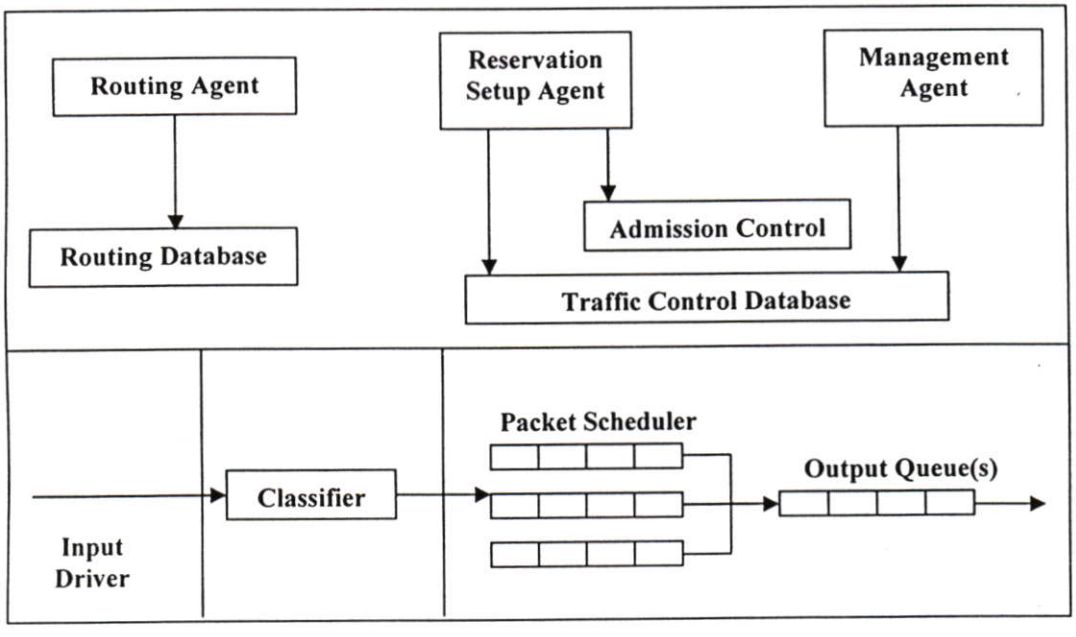
2.2.2 The Integrated Services Architecture

สถาปัตยกรรม The Integrated Services (IntServ: IS) [1] ได้ถูกกำหนดโดยคณะทำงาน IETF โดยเสนอให้มีการขยายการรองรับการสื่อสารนอกเหนือจากทราฟฟิกแบบ best-effort นั่นก็คือแอพพลิเคชั่นแบบ real-time ที่ใช้กันแพร่หลายบนอินเทอร์เน็ต ซึ่งอินเทอร์เน็ตแบบดั้งเดิมได้มีการรับประกันคุณภาพในการให้บริการแบบง่าย ๆ คือ การส่งข้อมูล best-effort แบบจุดต่อจุด

เนื่องจากมีเวลาที่สูญเสียไปในการหาเส้นทางและการสูญหายของแพ็กเกตจากความคับคั่งของข้อมูล ซึ่งทำให้แอพพลิเคชั่นแบบ real-time ไม่สามารถทำงานได้ดีบน best-effort อินเทอร์เน็ต ซึ่งการประชุมผ่านวิดิทัศน์ การแพร่ภาพวิดิทัศน์ และการประชุมผ่านเสียง เป็นต้น ต่างก็ต้องการการรับประกันแบนด์วิดธ์ที่ให้กับภาพและเสียงในคุณภาพที่ยอมรับได้ เพื่อความต้องการที่รองรับบริการต่าง ๆ เหล่านี้จึงจำเป็นที่จะต้องมีการปรับเปลี่ยนโครงสร้างหลักของอินเทอร์เน็ต เพื่อที่จะให้มีการควบคุมการเสียเวลาของแพ็กเกตและการอนุญาตใช้แบนด์วิดธ์ตั้งแต่ต้นทางไปจนถึงปลายทาง (end-to-end)

2.2.2.1 โมเดล Integrated Services

เพื่อที่จะรองรับโมเดล IntServ อินเทอร์เน็ตเราเตอร์จึงจำเป็นที่จะต้องให้คุณภาพในการให้บริการที่เหมาะสำหรับแต่ละการเชื่อมต่อ ฟังก์ชันของเราเตอร์ต้องให้คุณภาพในการให้บริการที่แตกต่างกันเรียกว่า ส่วนควบคุมทราฟฟิก (Traffic Control) ซึ่งจะประกอบไปด้วย ส่วนควบคุมการอนุมัติ (Admission Control) ส่วนแยกประเภทของแพ็กเกต (Packet Classifier) ส่วนจัดลำดับของแพ็กเกต (Packet Scheduler) และโพรโตคอลสำรองทรัพยากร (Reservation Setup Protocol) ดังรูปที่ 2.2



รูปที่ 2.2 Implement Model [1]

2.2.2.1.1 ส่วนควบคุมการอนุมัติและนโยบาย

ส่วนควบคุมการอนุมัติของ IntServ จะประกอบไปด้วยอัลกอริทึม (Algorithm) ที่ใช้ในการตัดสินใจ ซึ่งเราเตอร์จะใช้ในการหาค่า ถ้าในเส้นทางที่จะไปสู่ปลายทางมีทรัพยากรเพียงพอกับคุณภาพในการให้บริการก็จะยอมรับการเชื่อมต่อใหม่ แต่ถ้าในเส้นทางนั้นไม่มีทรัพยากรเพียงพอที่จะรับการเชื่อมต่อเพิ่มขึ้นก็จะทำการตอบปฏิเสธการร้องขอนั้น ในกรณีที่ตอบรับการร้องขอใหม่ก็จะมีการสำรองแบนด์วิดธ์ไว้ที่เราเตอร์นั้น ๆ ซึ่งส่วนควบคุมการอนุมัติจะต้องมีในเราเตอร์ทุกตัวตลอดเส้นทางที่จะไปสู่ปลายทาง โดยจะทำหน้าที่ตัดสินใจตอบรับ/ตอบปฏิเสธเมื่อมีการร้องขอและสำรองแบนด์วิดธ์ให้แต่ละการร้องขอ

2.2.2.1.2 ส่วนแยกประเภทของแพ็กเกต

การจำแนกประเภทของแพ็กเกตเพื่อระบุแพ็กเกตของแต่ละโฮสต์ และเราเตอร์จะให้บริการในระดับที่ต่างกันเพื่อที่จะให้มีการควบคุมทราฟฟิกที่มีประสิทธิภาพโดยการจำแนกแต่ละแพ็กเกตที่เข้ามาไปสู่ในแต่ละคลาส (class) ทุกแพ็กเกตที่อยู่ในคลาสเดียวกันจะได้รับการบริการที่เหมือนกันจากส่วนจัดลำดับแพ็กเกต การเลือกคลาสจะขึ้นอยู่กับไอพีแอดเดรส (IP Address) ต้นทาง ไอพีแอดแดรสปลายทาง และหมายเลขช่องทาง (port number) ที่อยู่ในส่วนหัวของแพ็กเกต หรือหมายเลขการจำแนกประเภทซึ่งจะเพิ่มเข้าไปในแต่ละแพ็กเกต คลาสสามารถจำแนกไปตามการเชื่อมต่อได้ เช่น การเชื่อมต่อแบบวิดิทัศน์ทั้งหมดจากการประชุมผ่านวิดิทัศน์ ซึ่งมีหลายสมาชิกสามารถรวมได้เป็นหนึ่งประเภทของบริการแต่อาจจะเป็นไปได้ที่มีหนึ่งการเชื่อมต่อแยกไปเป็นหนึ่งประเภทของบริการได้

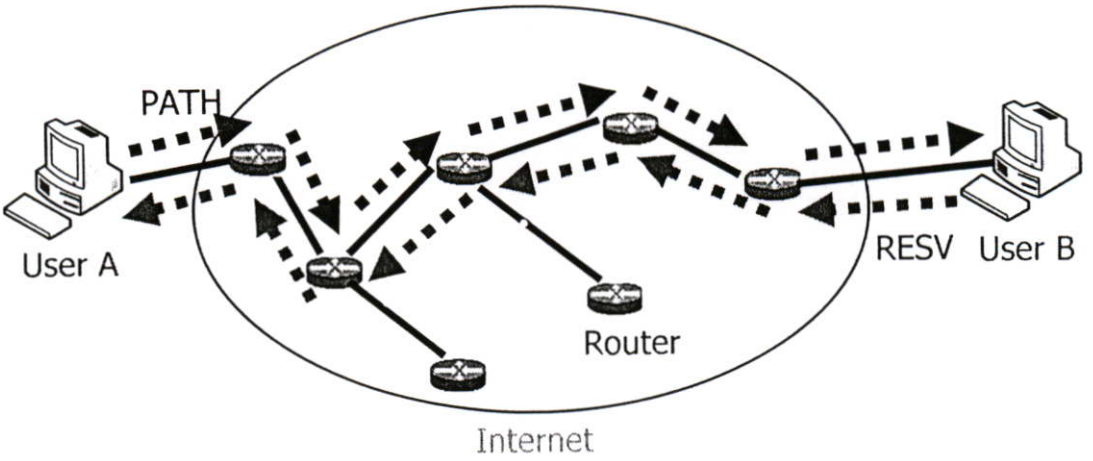
2.2.2.1.3 ส่วนจัดลำดับแพ็กเกต

ส่วนจัดลำดับแพ็กเกตจะมีหน้าที่จัดการส่งแพ็กเกตทั้งหลายของโฮสต์ต่อไปยังเราเตอร์บนพื้นฐานของระดับของบริการ การจัดการคิว (queue) และอัลกอริทึมการจัดลำดับที่แตกต่างกัน ส่วนจัดลำดับของแพ็กเกตจะทำให้แน่ใจว่าแพ็กเกตที่ส่งไปนั้นจะได้รับคุณภาพในการให้บริการตามที่กำหนดไว้ ซึ่งส่วนการจัดลำดับสามารถบริหารทราฟฟิกให้เป็นไปตามระดับของบริการโดยการจัดลำดับคิวในการรับส่งแพ็กเกต

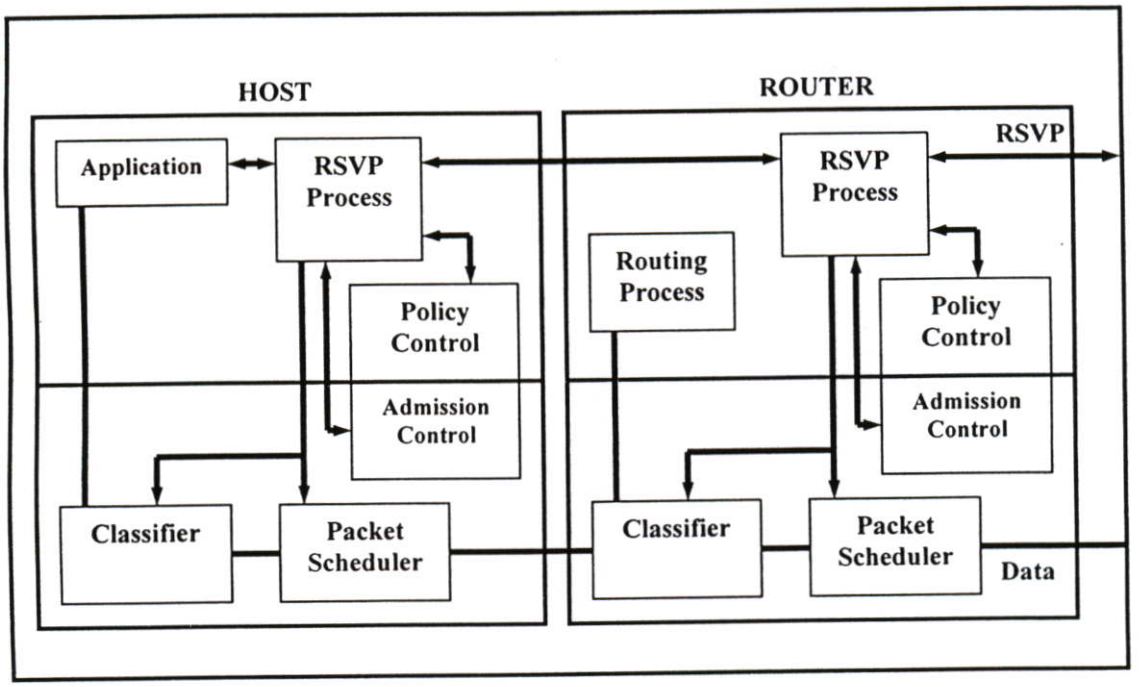
2.2.2.2 โพรโตคอลสำรองทรัพยากร (Resource Reservation Protocol)

โมเดล IntServ โดยการใช้โพรโตคอลสำรองทรัพยากร (Resource Reservation Protocol: RSVP) [2] [3] เพื่อเป็นการจองไว้ตามคุณภาพของบริการในเครือข่ายอินเทอร์เน็ตโพรโตคอล โดยโพรโตคอลสำรองทรัพยากรจะมีการสำรองทรัพยากรไปตามเส้นทางเดียวกันกับเส้นทางที่เชื่อมต่อ

จากรูปที่ 2.3 เมื่อผู้ใช้ A ต้องการส่งข้อมูลผ่านเครือข่าย IntServ ไปยังผู้ใช้ B ในขั้นแรกผู้ใช้ A จะต้องสร้างแพ็กเกตร้องขอที่ประกอบไปด้วยข้อมูลของผู้ส่ง ลักษณะทราฟฟิก และแบนด์วิดธ์ที่จะใช้ซึ่งจะเรียกว่า PATH message ซึ่ งจะมีตัวแปรของทราฟฟิกอยู่ด้วย โดยจะส่งแพ็กเกตนี้ผ่านเครือข่ายไปตามเราเตอร์ต่าง ๆ ตามเส้นทางที่จะไปถึงผู้รับ B เมื่อเราเตอร์ได้รับ PATH message นี้ก็จะเก็บข้อมูลไว้แล้วส่งไปให้เราเตอร์ถัดไป เมื่อ PATH message มาถึง ผู้ใช้ B ก็จะสร้างแพ็กเกตตอบรับขึ้นมา ซึ่งจะเรียกว่า RESV message แล้วส่งกลับไปยังผู้ใช้ A โดยจะส่งไปในเส้นทางเดียวกันกับที่ส่ง PATH message มา เมื่อผู้ใช้ A ได้รับ RESV message ก็จะเริ่มทำการส่งข้อมูลไปยังผู้ใช้ B ได้ โดยในระหว่างที่กำลังส่งข้อมูลอยู่นั้น แต่ละเราเตอร์จะยังคงเก็บรักษาสถานะการรับประกันคุณภาพในการให้บริการตลอดเส้นทางของการสื่อสารระหว่างผู้ใช้ A และผู้ใช้ B ไว้จนสิ้นสุดการเชื่อมต่อ



รูปที่ 2.3 IntServ/RSVP Model



รูปที่ 2.4 Host-Router model in RSVP [2]

รูปที่ 2.4 เมื่อแอพพลิเคชั่นต้องการที่จะส่งแพ็กเกตโดยการสำรองทรัพยากร โพรโตคอล RSVP จะพยายามจัดตั้งการเชื่อมต่อตามคุณภาพในการให้บริการ RSVP จะรายงานแก่ส่วนจำแนกแพ็กเกตและส่วนจัดลำดับแพ็กเกตในแต่ละโหนดในการประมวลผลแพ็กเกตสำหรับการเชื่อมต่อ เมื่อแอพพลิเคชันได้ส่งแพ็กเกตข้อมูลไปยังส่วนจำแนกข้อมูลในโหนดแรก ก็จะทำการจำแนกการเชื่อมต่อนี้ไปสู่คลาสที่ให้บริการเฉพาะตามคุณภาพในการให้บริการ ซึ่งการเชื่อมต่อนี้จะจดจำไอพีแอดเดรสของผู้รับ ไอพีแอดเดรสผู้ส่ง และพอร์ต แล้วจะส่งต่อไปยังส่วนจัดลำดับแพ็กเกต โดยใน

ส่วนจัดลำดับแพ็กเกตจะทำการส่งแพ็กเกตต่อไปตามคลาสที่ให้บริการไปที่เราเตอร์ถัดไปจนถึงจุดหมายปลายทาง

2.2.2.3 คลาสของบริการ (Classes of Service)

โมเดล Integrated Services ได้ให้บริการในคลาสที่ต่าง ๆ กันโดยรวมเอาบริการที่กำหนดโดยคณะทำงาน IETF ไว้ในโมเดลนี้ด้วย ซึ่งก็คือ บริการแบบรับประกัน (Guaranteed Service) [4] และบริการแบบไม่รับประกัน (Controlled Service) [5]

IntServ ได้กำหนดบริการแบบไม่รับประกันจากพฤติกรรมของทราฟฟิกตั้งแต่ต้นทางไปจนถึงปลายทาง โดยมีการทดสอบประสิทธิภาพของบริการแบบ best-effort จากค่าพารามิเตอร์เมื่อเครือข่ายไม่มีภาระ ซึ่งระดับในการบริการที่สูงกว่า best-effort จะสามารถทำให้เครือข่ายเกิดความคับคั่งได้แต่แอพพลิชันที่ใช้บริการแบบนี้มีโอกาสสูงที่จะส่งแพ็กเกตได้สำเร็จ ซึ่งจำนวนแพ็กเกตที่ถูกละทิ้งควรจะเท่ากับอัตราความผิดพลาดในการส่งข้อมูล เวลาแฝงของเครือข่ายจะต้องมีค่าไม่มากกว่าค่าเวลาหน่วงที่ยอมรับได้ในการส่งแพ็กเกต

บริการแบบไม่รับประกันจะไม่มีการกำหนดตัวแปรที่ใช้ในการควบคุมเวลาหน่วงหรือการสูญหายของแพ็กเกต แต่บริการนี้จะใช้กับแอพพลิเคชันแบบ real-time เช่น การกระจายภาพและเสียงแบบทิศทางเดียว ซึ่งไม่จำเป็นที่จะต้องควบคุมเวลาหน่วงโดยเฉพาะเจาะจง แต่ต้องมีการสำรองไว้ที่เครื่องผู้รับในการรับข้อมูลมาก่อนที่จะแสดงผล

บริการแบบรับประกันจะเป็นการสื่อสารสำหรับแอพพลิเคชันที่ต้องการการรับประกันทั้งแบนด์วิดธ์และค่าเวลาหน่วงสูงสุดที่ยอมรับได้จากเครือข่าย ในการรับประกันแบนด์วิดธ์ นั่นหมายความว่าจะต้องไม่มีการสูญหายของแพ็กเกตในระหว่างอยู่ในแถวคอยของส่วนสำรองข้อมูล ส่วนการรับประกันค่าเวลาหน่วงหมายถึงเวลาหน่วงของแพ็กเกตสูงสุดที่ยอมรับได้ในระหว่างส่งแพ็กเกตผ่านเครือข่ายตั้งแต่ต้นทางไปจนถึงปลายทาง

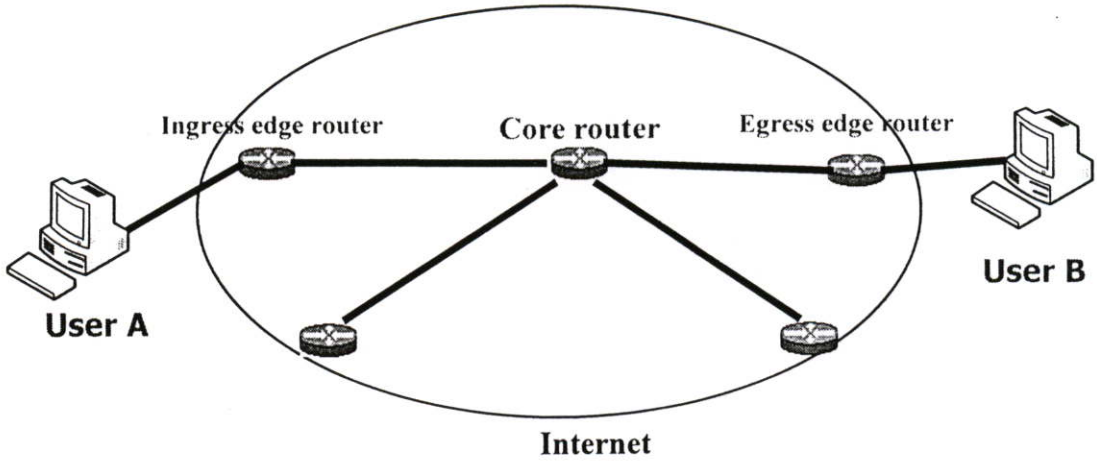
ปัญหาหลักของสถาปัตยกรรมแบบ IntServ คือ จะมีการเก็บรักษาสถานะการสำรองทรัพยากรไว้ที่ทุกเราเตอร์ตามเส้นทางของการเชื่อมต่อ ซึ่งเมื่อมีผู้ใช้งานจำนวนมากขึ้นจะทำให้เราเตอร์มีภาระอย่างหนักในการประมวลผล (เช่น การคำนวณ หน่วยความจำ) และดูแลรักษาสถานะการสำรองทรัพยากรของแต่ละการเชื่อมต่อ

2.2.3 Differentiated Services

สถาปัตยกรรม Differentiated Service (DiffServ) [6] ได้ขยายงานที่ทำจาก IntServ และ RSVP ซึ่งมีเป้าหมายพื้นฐานที่เหมือนกันคือ คุณภาพในการให้บริการและการบริการที่แตกต่างกันในเครือข่ายอินเทอร์เน็ตโพรโตคอล ซึ่งสถาปัตยกรรมแบบ IntServ มีการรับประกันสำหรับแต่ละการเชื่อมต่อ ส่วนสถาปัตยกรรม DiffServ ได้เปลี่ยนจากการรับประกันสำหรับแต่ละการเชื่อมต่อไป

เป็นการรับประกันตามระดับของบริการ ซึ่งจะเรียกว่า Class of Service (CoS) โดยการให้บริการแบบนี้จะทำให้เครือข่ายสามารถรองรับผู้ใช้งานจำนวนมากได้และไม่ใช้สัญญาณในการดูแลรักษาสถานะการสำรองทรัพยากรที่เราเตอร์ตามเส้นทาง

ส่วนประกอบหลักที่สำคัญของสถาปัตยกรรม DiffServ คือ Service Level Agreement (SLA) ซึ่ง SLA คือ ข้อตกลงของบริการระหว่างลูกค้าและผู้ให้บริการโดยจะมีการเจาะจงรายละเอียดของทราฟฟิก และการตอบสนองของบริการในการส่งต่อแพ็กเกตที่ลูกค้าควรจะได้รับ ซึ่งลูกค้าอาจจะเป็นกลุ่มผู้ใช้งานหรือเป็นกลุ่ม DiffServ อื่น ๆ โดยผู้ให้บริการมั่นใจว่าทราฟฟิกของลูกค้าจะได้รับคุณภาพในการให้บริการตามสัญญา SLA ที่ได้ตกลงกันไว้ ฉะนั้นผู้ให้บริการของเครือข่ายจะต้องมีการกำหนดนโยบายในการให้บริการที่เหมาะสมและสามารถวัดประสิทธิภาพในการรับประกันคุณภาพในการให้บริการได้ ซึ่งเครือข่าย DiffServ จะต้องมีการจำแนกประเภทของแพ็กเกตข้อมูลจากลูกค้า จึงมีการปรับเปลี่ยนแพ็กเกตให้มีขอบเขตพิเศษขึ้นมาเพื่อใช้ในการเก็บรหัสจำนวนหนึ่งซึ่งเรียกว่า DS field (DiffServ field) โดยกำหนดให้ DS field จะต้องมีอยู่ในทุก ๆ แพ็กเกตซึ่งจะใช้เป็นตัวบ่งบอกถึงระดับในการให้บริการที่ได้กำหนดไว้ตามข้อตกลง



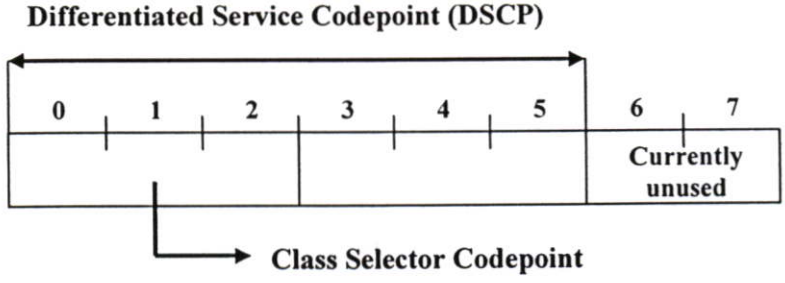
รูปที่ 2.5 DiffServ Model

2.2.3.1 DiffServ (DS) field

ส่วนสำคัญของสถาปัตยกรรมแบบ DiffServ คือ DS field ซึ่งจะใช้เป็นตัวบ่งบอกถึงระดับที่แพ็กเกตนั้น ๆ จะได้รับการบริการจากเราเตอร์ในเครือข่าย โดย DS field จะมีขนาด 8 บิต โดยจะใช้พื้นที่ของ TOS (Type of Services) ในส่วนหัวของแพ็กเกต โดยจะมีการจำแนกระดับในการให้บริการให้แก่ทุก ๆ แพ็กเกตที่เข้ามายังเครือข่ายที่เราเตอร์ตัวขอบขาเข้า (ingress edge router)

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

จะใช้พื้นที่ขนาด 6 บิต ของ DS field ในการกำหนดรหัสของระดับในการให้บริการ ซึ่งค่าของขอบเขตนี้จะเรียกว่า DiffServ Codepoint (DSCP) ตามรูปที่ 2.6 โดยแต่ละรหัสของ DSCP จะมีความหมายของ PHB (อธิบายในหัวข้อถัดไป) ที่แตกต่างกันไป ซึ่ง 2 บิต ที่เหลือยังไม่ใช้ในขณะนี้ โดยจะกำหนดเป็น Currently Unused (CU) field



รูปที่ 2.6 DiffServ (DS) codepoint

DSCP field จะมีขนาด 6 บิตโดยจะแบ่งเป็น 3 กลุ่ม ไปใช้ในการกำหนดรหัสและการจัดการร่วมกัน กลุ่มที่ 1 จะมีทั้งหมด 32 รหัส ซึ่งจะกำหนดให้เป็นมาตรฐาน กลุ่มที่ 2 จะมีทั้งหมด 16 รหัส ซึ่งใช้สำรองในการทดลอง และกลุ่มที่ 3 จะมี 16 รหัส ซึ่งจะเป็นค่าเริ่มต้นในการทดลอง หรือใช้เป็นมาตรฐาน ตามตารางที่ 2.1

ตารางที่ 2.1 DiffServ (DS) codepoint [8]

Pool	Codepoint Space	Assignment Policy
1	XXXXX0	Standard Action
2	XXXX11	EXP/LU
3	XXXX01	EXP/LU or Standard Action

2.2.3.2 Per-Hop Behavior (PHB)

Per-Hop Behavior ถูกกำหนดเป็นคุณลักษณะที่แสดงถึงพฤติกรรมการให้บริการแพ็กเกตที่เข้ามายังเราเตอร์ภายในเครือข่ายแบบ DiffServ ซึ่งความสำคัญของ PHB คือการให้บริการที่สร้างขึ้นเพื่อให้มีลักษณะในการปฏิบัติต่อแพ็กเกตเหมือนกันสำหรับแพ็กเกตที่อยู่ในระดับเดียวกันในที่ทุก ๆ เราเตอร์

ในแต่ละโหนดจะมีการเลือกใช้ PHB ซึ่งจะได้มาจากรหัส DSCP ที่กำหนดมาในแต่ละแพ็กเกต โดยเราเตอร์ในเครือข่ายแบบ DiffServ จะต้องมีฟังก์ชันที่จะใช้ในการควบคุมแพ็กเกตในแต่ละระดับ โดยการกำหนดลักษณะของ PHB จะกำหนดในรูปของตัวแปรที่สามารถวัด

61666

ประสิทธิภาพได้ เช่น การกำหนดความสำคัญของแพ็กเกตที่อยู่ในแถวคอย ความยาวของแถวคอย หรืออัลกอริทึมในการละทิ้งแพ็กเกต ในปัจจุบันได้มีการกำหนดลักษณะของ PHB เป็น 3 ประเภท คือ Class Selector (CS) PHB [8] Expedited Forwarding (EF) PHB [9] และ Assured Forwarding (AF) PHB [10]

2.2.3.2.1 Class Selector PHB

Class Selector PHB จะเป็นระดับในการให้บริการที่ไม่มีการรับประกัน ซึ่งจะกำหนดค่า DSCP เท่ากับ "XXX000" โดยปกติบริการประเภทนี้จะเหมาะกับแอพพลิเคชันประเภทเว็บบราวเซอร์

2.2.3.2.2 Expedited Forwarding PHB

Expedited Forwarding PHB จะเป็นลักษณะที่กำหนดให้ค่าสูญหายของแพ็กเกตต่ำ ค่าเวลาแฝงต่ำ และมีบริการรับรองแบนด์วิดธ์ตั้งแต่ต้นทางไปจนถึงปลายทาง ซึ่งการสูญหายของแพ็กเกต ค่าเวลาแฝง และการชนกันของแพ็กเกตทั้งหมดนี้จะเกิดขึ้นในช่วงของแถวคอยในขณะที่กำลังเดินทางในเครือข่าย โดยการที่จะทำให้มีค่าเวลาแฝงและการชนกันของแพ็กเกตที่ต่ำคือ การทำให้ทราฟฟิกที่ไหลผ่านเราเตอร์โดยไม่ต้องมีการรอคอยในคิวหรือมีเพียงเล็กน้อย ซึ่งการที่จะทำให้ไม่มีการอคอยจะต้องให้มีอัตราการมาถึงสูงสุดของแพ็กเกตมีค่าน้อยกว่าอัตราการไหลออกของแพ็กเกต โดยบริการประเภทนี้จะเหมาะสำหรับแอพพลิเคชันประเภท real-time โทรศัพท์ผ่านอินเทอร์เน็ตโพรโตคอล และการประชุมผ่านวิดิทัศน์ เป็นต้น และใช้รหัส DSCP เท่ากับ "101110"

2.2.3.2.3 Assured Forwarding PHB

จะเป็นบริการที่มีการบริการในระดับต่าง ๆ ไว้ด้วยกันโดยจะใช้ข้อตกลงเงื่อนไขของทราฟฟิก (Traffic Conditioning Agreement: TCA) เพื่อบ่งบอกระดับของการรับประกันในการให้บริการต่อแพ็กเกตที่แตกต่างกันในประเภทนี้ ซึ่งในปัจจุบันมีการแบ่งประเภทไว้ 4 ประเภท โดยแต่ละประเภทจะแบ่งระดับความสำคัญได้อีก 3 ระดับ ตามตารางที่ 2.2 แพ็กเกตในระดับที่หนึ่งจะได้รับบริการโดยทันทีและไม่ขึ้นกับคลาส AF ในระดับอื่น ๆ ซึ่งเราเตอร์ DiffServ จะต้องจัดสรรทรัพยากรให้แต่ละคลาส AF ตามระดับที่ได้ตกลงกันไว้ในการให้บริการต่อแพ็กเกต

ตารางที่ 2.2 Assured Forwarding (AF) Classes [10]

Drop Precedence	AF Class 1	AF Class 2	AF Class 3	AF Class4
Low	001010	010010	011010	100010
Medium	001100	010100	011100	100100
High	001110	010110	011110	100110

2.2.4 Integrated Services over Differentiated Services

ในสังคมสื่อสารมีความเห็นตรงกันที่จะต้องการดัดแปลงโมเดล IntServ หรือโมเดล DiffServ เพื่อให้สามารถรองรับบริการของเครือข่ายได้หลากหลายรูปแบบ ซึ่ง RFC 2998 [11] ได้แนะนำให้รวมสถาปัตยกรรมทั้งสองไว้ด้วยกันซึ่งก็คือ IntServ over DiffServ Framework โดยจะให้ IntServ ทำงานควบคุมกลไกที่เราเตอร์ขอบ (Edge router) ของเครือข่าย และ DiffServ จะทำงานควบคุมกลไกที่เราเตอร์หลัก (Core router)

IntServ over DiffServ Framework [12] [13] [14] สามารถจัดสรรการสำรองทรัพยากรได้ทั้งแบบคงที่ (statically) และแบบเปลี่ยนแปลงได้ (dynamically) วิธีการสำรองทรัพยากรแบบคงที่จะมีการกำหนดระดับในการให้บริการแก่ลูกค้าที่เราเตอร์ขอบขาเข้า และทรัพยากรของเครือข่ายจะถูกสำรองแบบคงที่ตามระดับในการให้บริการที่ได้ตกลงกันไว้ ที่แต่ละเราเตอร์ขอบจะประกอบไปด้วยตารางที่ใช้ในการสำรองแบนด์วิดธ์ให้แก่ประเภทของบริการต่าง ๆ โดยเราเตอร์ขอบจะสามารถทำการตัดสินใจอนุมัติการเชื่อมต่อได้ซึ่งเมื่อเปรียบเทียบกับสถาปัตยกรรมแบบ DiffServ แล้ว วิธีนี้จะรักษาคุณภาพในการให้บริการแก่ผู้ใช้งานโดยการใช้ส่วนควบคุมการอนุมัติการเชื่อมต่อที่ขอบของเครือข่าย ส่วนวิธีการสำรองทรัพยากรแบบเปลี่ยนแปลงได้จะมีการใช้สัญญาณ RSVP ในเริ่มแรกจะใช้ RSVP เพื่อสำรองแบนด์วิดธ์ไว้ก่อนแล้วจะมีการนำมาใช้กับส่วนควบคุมอนุมัติการเชื่อมต่อและการสำรองแบนด์วิดธ์ตั้งแต่ต้นทางไปสู่ปลายทาง โดยในเราเตอร์หลักจะยังคงมีการทำงานแบบ DiffServ เช่นกัน

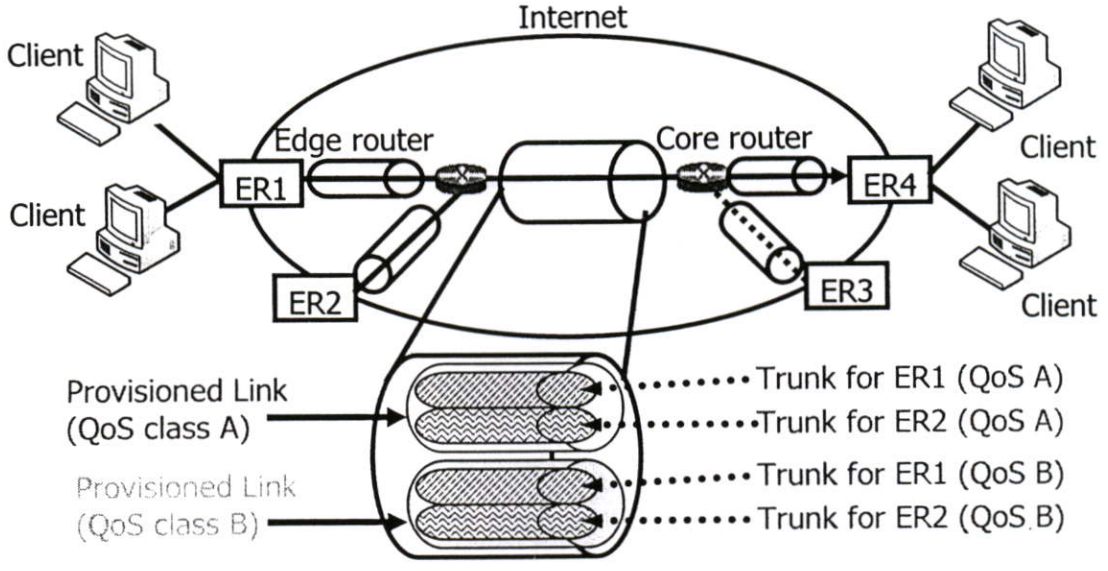
IntServ over DiffServ Framework ได้มีการพัฒนาอย่างแพร่หลายในขณะนี้ซึ่งได้พัฒนาไปในหลายทิศทาง โดยในวิทยานิพนธ์ฉบับนี้ได้เลือกเอาสถาปัตยกรรมแบบ IntServ over DiffServ ของ Mei Yang [15] มาเป็นเครือข่ายมาตรฐานที่ใช้ในการทดลองและนำมาเปรียบเทียบกับวิธีการที่นำเสนอ (อธิบายในบทที่ 3) ซึ่งเครือข่ายนี้จะมีหลักการทำงานดังต่อไปนี้

2.2.4.1 สถาปัตยกรรมพื้นฐาน

ในสถาปัตยกรรมนี้จะกำหนดให้มีเราเตอร์ 2 ประเภทในเครือข่ายคือ เราเตอร์ขอบ (Edge Router) และเราเตอร์หลัก (Core Router) โดยเราเตอร์ขอบจะทำหน้าที่เป็นส่วนควบคุมการตัดสินใจและทำการเชื่อมโยงแต่ละการเชื่อมต่อไปเป็นบริการตามประเภทที่แตกต่างกันและส่งแพ็กเกตไปยังเครือข่าย เราเตอร์หลักจะทำหน้าที่เป็นเราเตอร์แบบ DiffServ ซึ่งให้บริการตามประเภททราฟฟิกในระดับบริการที่แตกต่างกัน

สถาปัตยกรรมนี้ได้ให้บริการตามประเภทของบริการต่าง ๆ และมีส่วนควบคุมการอนุมัติที่ขอบของเครือข่ายโดยปราศจากสัญญาณที่ส่งไปเป็นช่วงที่เราเตอร์ มีการจัดการแบ่งแบนด์วิดธ์เป็นลำดับชั้น (Hierachical) โดยในแต่ละสายสื่อสาร physical link จะแบ่งได้เป็นหลาย Provision Link (PL) ซึ่งหนึ่ง PL จะใช้สำหรับทราฟฟิกประเภทเดียวเท่านั้น ในแต่ละ PL จะสามารถแบ่งได้เป็น

หลาย trunk ซึ่งหนึ่ง trunk จะใช้สำหรับหนึ่งเราเตอร์ขอบ โดย trunk จะรองรับการเชื่อมต่อจากเราเตอร์ขอบที่เป็นแหล่งกำเนิดเดียวกันโดยไม่สนใจจุดหมายปลายทาง การเชื่อมต่อผ่าน PL เดียวกันในเครือข่ายแต่มีแหล่งกำเนิดของเราเตอร์ขอบที่แตกต่างกันจะใช้ trunk ต่างกัน โดยจะมีลักษณะการทำงานตามรูปที่ 2.7



รูปที่ 2.7 IntServ over DiffServ Architecture

2.2.4.2 กระบวนการควบคุมการจัดตั้งการเชื่อมต่อ

เราเตอร์ขอบจะมีส่วนควบคุมการอนุมัติการเชื่อมต่อโดยไม่ต้องใช้สัญญาณในดูแลรักษาสถานะการเชื่อมต่อไปที่เราเตอร์หลัก โดยจะมีการสร้างตารางในการเก็บข้อมูลขึ้นมา 2 ตารางคือ ตาราง Virtual IP path (VIP) และตาราง trunk ซึ่ง VIP คือ เส้นทางของแต่ละการเชื่อมต่อจากเราเตอร์ขอบที่เป็นต้นทางไปจนถึงเราเตอร์ขอบที่เป็นจุดปลายทาง ในตาราง VIP จะบันทึกเส้นทางทั้งหมดที่มาจากเราเตอร์ขอบขาเข้าไปจนถึงเราเตอร์ขอบขาออกอื่น ๆ ทั้งหมด โดยในตารางจะสามารถกำหนดหรือสร้างกลไกการหาเส้นทางไว้ [16] ในตาราง VIP จะประกอบไปด้วย VIP ID ID ของเราเตอร์ปลายทาง ประเภทของทราฟฟิก รายการ ID ของ trunk ที่ประกอบเป็นเส้นทางของ VIP ตามตารางที่ 2.3

ตารางที่ 2.3 VIP table

VIP ID	Destination Edge Router	Traffic Class	List of Trunks

ในตาราง trunk จะบันทึก trunk ทั้งหมดที่เป็นของเราเตอร์ขอบ และข้อมูลการใช้งานแบนด์วิดธ์ ซึ่งภายในตาราง trunk จะประกอบไปด้วย Trunk ID, Provision Link ID จำนวนแบนด์-วิดธ์ที่ได้สำรองไว้ให้ trunk และจำนวนแบนด์วิดธ์ที่ใช้อยู่ในขณะนั้นของ trunk ตามตารางที่ 2.4

ตารางที่ 2.4 Trunk table

Trunk ID	Provisioned Link ID	Reserved Bandwidth	Bandwidth Being Used

เมื่อมีการร้องขอใหม่มาถึงที่เราเตอร์ขอบตัวแรกของเครือข่าย จะทำการตรวจสอบ VIP ที่เราเตอร์ขอบปลายทางว่ามีอยู่แล้วในตารางหรือไม่ (ได้มีการแยกประเภทของทราฟฟิกไว้เรียบร้อยแล้วก่อนทำการตรวจสอบเราเตอร์ขอบจากตาราง VIP) ต่อมาเราเตอร์ขอบขาเข้าจะทำการตรวจสอบแบนด์วิดธ์ที่เหลืออยู่ของแต่ละ trunk จากตาราง trunk ถ้ามีแบนด์วิดธ์เพียงพอที่จะรองรับการร้องขอก็จะทำการตอบยอมรับการร้องขอนั้นและทำการสำรองแบนด์วิดธ์ตั้งแต่ต้นทางไปจนถึงปลายทางสำหรับการร้องขอนั้น และจะอับเดทค่าในช่องแบนด์วิดธ์ที่กำลังใช้สำหรับทุก ๆ trunk แต่ในกรณีที่มีแบนด์วิดธ์ไม่เพียงพอที่จะรองรับการเชื่อมต่อใหม่เพิ่มได้ ก็จะทำการตอบปฏิเสธกลับไปยังผู้ใช้งาน และถ้าผู้ใช้งานยังคงต้องการที่จะติดต่อผ่านเครือข่ายก็จะต้องทำการสร้างแพ็กเกตร้องขอการเชื่อมต่อขึ้นมาใหม่และส่งมาร้องขอจัดตั้งการเชื่อมต่อที่เครือข่ายซ้ำ ๆ จนกว่าจะได้รับการตอบยอมรับ เมื่อการเชื่อมต่อได้สิ้นสุดลง แบนด์วิดธ์ที่เคยใช้อยู่ก็จะถูกปลดปล่อยกลับสู่เครือข่ายและก็จะทำการจัดสรรให้การเชื่อมต่ออื่นแล้วก็จะมีการปรับปรุงตาราง trunk ที่มีผลเปลี่ยนแปลง

2.2.4.3 กระบวนการทำงานของ Trunk

ในงานนี้ได้มีการทำการปรับเปลี่ยนทราฟฟิกที่กำหนดแบนด์วิดธ์ให้กับ trunk ในช่วงเวลาสั้น ๆ โดยเราเตอร์ขอบที่เป็นจุดกำเนิดจะสามารถขอแบนด์วิดธ์เพิ่มให้แก่ trunk หรือทำการปล่อยแบนด์วิดธ์ที่ไม่ได้ใช้ของ trunk ซึ่งแบนด์วิดธ์นี้จะทำการปรับเปลี่ยนภายใต้ provision link ซึ่งได้กำหนดเป็นคงที่ กระบวนการทำงานของ trunk จะเป็นดังต่อไปนี้

ขั้นที่ 1 การจัดการตาราง Provisioned Link

ตาราง Provisioned Link ได้มีการบำรุงประสิทธิภาพการใช้งานแบนด์วิดธ์ของ provision link ในเครือข่าย ซึ่งจะแสดงตามตารางที่ 2.5 ในตารางจะประกอบไปด้วย Provisioned Link ID, Physical Link ID จำนวนแบนด์วิดธ์ที่สำรองไว้ จำนวนแบนด์วิดธ์ที่ใช้ในขณะนั้น และประเภทของทราฟฟิกที่ provision link นั้นรองรับ

ตารางที่ 2.5 Provisioned Link table

Provisioned Link ID	Physical Link ID	Reserved Bandwidth	Bandwidth Being Used	Traffic Class

ขั้นที่ 2 การปล่อยแบนด์วิดธ์ที่ไม่ได้ใช้งานของ Trunk

เราเตอร์ขอบที่เป็นจุดกำเนิดจะมีการตรวจสอบ trunk เป็นช่วง ๆ ตามตารางที่ 2.4 และมีการหาค่าประสิทธิภาพการใช้งานของ trunk ถ้าประสิทธิภาพการใช้แบนด์วิดธ์ของ trunk ต่ำกว่าค่าที่กำหนดเอาไว้ (lower threshold) เราเตอร์ขอบจะทำการปรับปรุงตาราง trunk และส่ง control message เพื่อทำการปรับปรุงตาราง provisioned link

ขั้นที่ 3 ความต้องการแบนด์วิดธ์เพิ่มของ Trunk

เมื่อมีการร้องขอมาถึงที่เราเตอร์ขอบ จะมีการกำหนด VIP ให้แก่การเชื่อมต่อจากตามตารางที่ 2.3 และจะมีการตรวจสอบตาราง trunk ตามตารางที่ 2.4 ถ้าพบว่ามีการใช้งานแบนด์วิดธ์ของแต่ละ trunk เกินกว่าค่าสูงสุดที่กำหนดไว้ (upper threshold) เราเตอร์ขอบจะส่ง control message เพื่อขอแบนด์วิดธ์เพิ่มจาก provision link

ตามที่ได้อธิบายในขั้นที่ 2 การปล่อยแบนด์วิดธ์ของ trunk ที่ไม่ใช้แล้วและในขั้นที่ 3 การขอแบนด์วิดธ์เพิ่ม จะมีการใช้ control message 3 ประเภทคือ Trunk Reconfiguration Request Message (TRRequest), Trunk Reconfiguration Release Message (TRRelease) และ Trunk Reconfiguration Confirm Message (TRConfirm) ซึ่งทั้ง 3 ประเภทจะมีรูปแบบของแพ็กเกตที่เหมือนกันตามรูปที่ 2.8 ซึ่งรูปแบบของแพ็กเกตจะประกอบไปด้วย ประเภทของแพ็กเกต (เช่น TRRequest TRRelease และ TRConfirm) ID ของเราเตอร์ขอบต้นกำเนิด ID ของเราเตอร์ขอบปลายทาง VIP ID ความยาวของ control message และจะมีชุดข้อมูลของ trunk ID จำนวนแบนด์-วิดธ์ที่จะร้องขอเพิ่มหรือจำนวนแบนด์วิดธ์ที่จะปล่อย

Message Type	Source Edge Router ID	Destination Edge Router ID	VIP ID	Length	(Trunk i, Bdw_requested / Bdw_released)	(Trunk j, Bdw_requested / Bdw_released)

รูปที่ 2.8 TRRequest / TRRelease / TRConfirm Message Format

ซึ่ง TRRequest Message จะมีอัลกอริทึมในการทำงานตามรูปที่ 2.9 และ TRRelease จะมีอัลกอริทึมการทำงานตามรูปที่ 2.10

1. Genarate a TRConfirm message.
2. Copy each (trunk I, Bdw_requested or Bdw_released) pair tuple in received TRRequest to the TRConfirm.
3. For trunk I specified in the TRRequest,
3.1 Examine if provisioned link PL_i that trunk I belongs to is in X's provisioned link table;
If yes, then do 3.2
Else set the Bdw_requested field of trunk I in TRConfirm message to 0 (0 indicates that X cannot make a decision for this request.)
3.2 Examine if Provisioned link PL_i has sufficient bandwidth available for reconfiguration;
If (the reserved bandwidth on PL_i – bandwidth being used on PL_i) ≥ Bdw_requested for trunk i
Then increase the bandwidth for trunk I, update the provisioned link table by
Bandwidth being used on PL_i = bandwidth being used on PL_i + Bdw_requested for trunk i;
And set the bandwidth requested field for trunk i in TRConfirm message to Bdw_requested (i.e. requested bandwidth has been assign).
Else set the bandwidth requested field for trunk i in TRConfirm message to – 1 (i.e. the bandwidth reconfiguration request for the trunk has failed, and no bandwidth has been assign.)
4. Send the TRConfirm message to the edge router which initiated the TRRequest message.

รูปที่ 2.9 TRRequest Message Process Algorithm

For trunk i specified in the TRRelease message,
1. Exanmin if provisioned link PL_i that trunk I belongs to is in X's provisioned link table.
2. If yes, update the provisioned link table by
bandwidth being used on PL_i = bandwidth being used on PL_i – Bdw_released for trunk i.
3. If no, ignore the TRRelease message.

รูปที่ 2.10 TRRelease Message Process Algorithm

บทที่ 3
วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ

โดยปกติแล้วเมื่อผู้ใช้งานต้องการจะสื่อสารผ่านเครือข่ายหนึ่ง ๆ ก็จะต้องทำการสร้างแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อแล้วส่งมาร้องขอที่เครือข่าย ซึ่งในเครือข่ายอินเทอร์เน็ตโพรโตคอลที่มีการรับประกันคุณภาพในการให้บริการจะมีส่วนควบคุมการอนุมัติการเชื่อมต่อ ซึ่งจะทำการตรวจสอบทรัพยากรทั้งหมดของเครือข่าย ถ้าเครือข่ายมีทรัพยากรเพียงพอที่จะสามารถรองรับการเชื่อมต่อจากผู้ใช้งานเพิ่มได้ ก็จะทำการสร้างแพ็กเกตตอบยอมรับแล้วส่งกลับไปยังผู้ที่ร้องขอมาทำให้สามารถสร้างการเชื่อมต่อและส่งข้อมูลได้ต่อไป แต่ถ้าส่วนควบคุมการอนุมัติการเชื่อมต่อตรวจสอบแล้วพบว่ามีทรัพยากรไม่เพียงพอที่จะให้ผู้ร้องขอทำการเชื่อมต่อได้ ก็จะทำการสร้างแพ็กเกตตอบปฏิเสธแล้วส่งกลับไปยังผู้ร้องขอในทันที และถ้าผู้ที่ร้องขอต้องการที่จะติดต่อสื่อสารผ่านเครือข่ายให้ได้ ผู้ร้องขอจะต้องทำการสร้างแพ็กเกตร้องขอขึ้นมาใหม่แล้วส่งไปยังเครือข่ายเพื่อทำการสร้างการเชื่อมต่อซ้ำ ๆ จนกว่าจะได้รับการตอบยอมรับจากเครือข่าย

สภาวะที่เครือข่ายมีทรัพยากรเพียงพอที่จะสามารถรองรับผู้ใช้งานให้ทำการเชื่อมต่อเพิ่มได้จะเรียกว่า “สภาวะปกติ” ส่วนสภาวะที่เครือข่ายมีทรัพยากรไม่เพียงพอที่จะรองรับการเชื่อมต่อจากผู้ใช้งานเพิ่มได้จะเรียกว่า “สภาวะอิ่มตัว” จากปัญหาที่กล่าวมาข้างต้นนี้ ในสภาวะที่เครือข่ายอิ่มตัวซึ่งผู้ใช้งานต้องการที่จะสื่อสารส่งข้อมูลผ่านเครือข่ายแล้วต้องทำการสร้างและส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อซ้ำ ๆ จนกว่าจะได้รับแพ็กเกตตอบยอมรับ ในวิทยานิพนธ์ฉบับนี้ได้มีแนวคิดที่ต้องการลดความยุ่งยากในการร้องขอจัดตั้งการเชื่อมต่อ โดยทำการหน่วงเวลาแพ็กเกตร้องขอไว้ในคิวรอการเชื่อมต่อ แทนที่จะปฏิเสธแพ็กเกตร้องขอนั้นไปในทันที ซึ่งสามารถลดทราฟฟิกที่เกิดจากการร้องขอซ้ำ ๆ และลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อได้ จึงได้เสนอ “วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ” เพื่อเป็นกลไกในการทำงานของส่วนควบคุมการอนุมัติการเชื่อมต่อในสภาวะที่เครือข่ายอิ่มตัว

วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ (Call Request Delaying Method: CRDM) เมื่อเปรียบเทียบหลักการของส่วนควบคุมอนุมัติการเชื่อมต่อในแบบพื้นฐานแล้ว จะสามารถลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ และทำให้เครือข่ายลดทราฟฟิกที่เกิดจากการส่งแพ็กเกตร้องขอซ้ำ ๆ ได้ นอกเหนือจากนั้นยังได้คำนึงถึงความเสมอภาคของผู้ร้องขอในการได้รับการจัดสรรทรัพยากรจากเครือข่ายด้วย

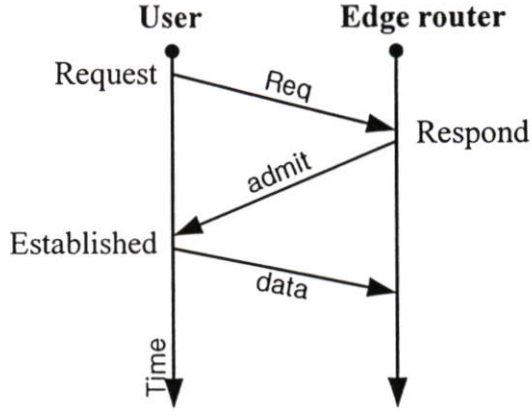
ในบทที่ 3 นี้จะกล่าวถึงการทำงานของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธและการพัฒนาบนโปรแกรมจำลองระบบเครือข่าย

3.1 การทำงานของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ (Call Request Delaying Method: CRDM)

ในวิทยานิพนธ์ฉบับนี้ได้เสนอกลไกการทำงานของส่วนควบคุมการอนุมัติการเชื่อมต่อแบบใหม่ โดยจะแบ่งการทำงานของส่วนควบคุมการอนุมัติการเชื่อมต่อเป็น 2 สภาวะคือ สภาวะปกติและสภาวะอิ่มตัว

3.1.1 ส่วนควบคุมการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธในสภาวะปกติ

ในสภาวะปกติคือ สภาวะที่เครือข่ายมีแบนด์วิดธ์อยู่เพียงพอที่จะสามารถรองรับการเชื่อมต่อจากผู้ใช้งานที่ร้องขอได้ซึ่งส่วนควบคุมการจัดตั้งการเชื่อมต่อในสภาวะนี้จะมีการทำงานเหมือนกับส่วนควบคุมการจัดตั้งการเชื่อมต่อแบบทั่ว ๆ ไปในเครือข่ายที่ยังมีแบนด์วิดธ์



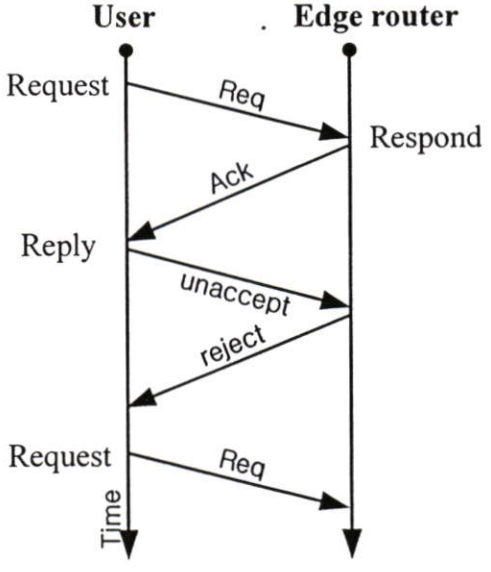
รูปที่ 3.1 การทำงานของ CRDM ในสภาวะปกติ

จากรูปที่ 3.1 เมื่อผู้ใช้งานได้ส่งแพ็กเกตร้องขอ (Req) มายังเครือข่าย ส่วนควบคุมการจัดตั้งการเชื่อมต่อก็จะสามารถอนุญาตให้ผู้ใช้งานทำการเชื่อมต่อได้เลยโดยการส่งแพ็กเกตตอบอนุมัติ (admit) ไปยังผู้ใช้งาน เนื่องจากในสภาวะปกติมีแบนด์วิดธ์อยู่เพียงพอที่จะสามารถทำการรับการเชื่อมต่อจากผู้ใช้งานที่ร้องขอเข้ามาได้

3.1.2 ส่วนควบคุมการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธในสภาวะอิ่มตัว

ในสภาวะอิ่มตัวคือ สภาวะที่เครือข่ายมีแบนด์วิดธ์ไม่เพียงพอที่จะสามารถรองรับการเชื่อมต่อจากผู้ใช้งานที่ร้องขอเข้ามาได้ ซึ่งส่วนควบคุมการจัดตั้งการเชื่อมต่อในสภาวะนี้จะทำงานโดยการหน่วงเวลาการปฏิเสธเมื่อมีการร้องขอเข้ามาและเก็บแพ็กเกตร้องขอไว้ในคิวรอการเชื่อมต่อ โดยจะ

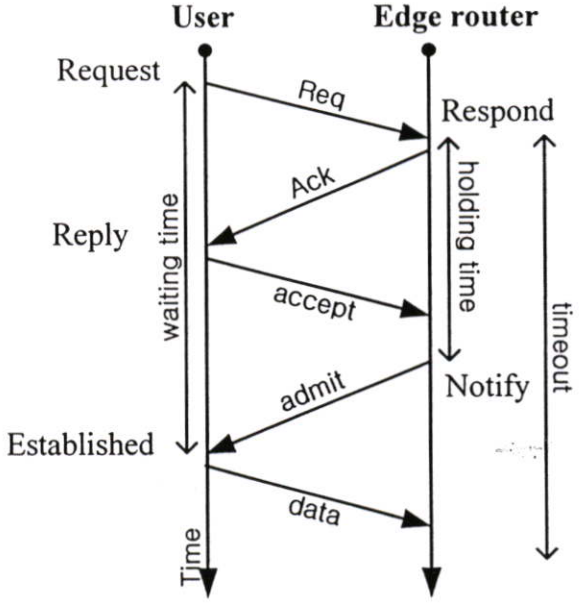
จัดลำดับตามการมาถึงก่อนหลังเพื่อให้เกิดความเสมอภาคในการได้รับการจัดสรรแบนด์วิดธ์ทำให้มีโอกาสในการเชื่อมต่อได้เท่า ๆ กัน โดยแต่ละแพ็กเกตจะถูกตั้งค่าเวลาในการรอคอยของแพ็กเกตไว้เพื่อป้องกันการถูกหน่วงเวลาไปโดยไม่มีที่สิ้นสุด ต่อมาส่วนควบคุมการจัดตั้งการเชื่อมต่อจะทำการส่งแพ็กเกตตอบสนอง (Ack) กลับไปยังผู้ที่ร้องขอเพื่อให้ทราบถึงสถานะการหน่วงเวลาแพ็กเกตไว้ ซึ่งผู้ร้องขอมีสิทธิ์ที่จะยอมรับการหน่วงเวลาจากเครือข่ายหรือไม่ก็ได้ ซึ่งในกรณีที่ไม่ยอมรับการหน่วงเวลา ผู้ใช้งานจะส่งแพ็กเกตตอบปฏิเสธ (unaccept) กลับไปยังเครือข่าย แล้วส่วนควบคุมการจัดตั้งการเชื่อมต่อจะทำการลบข้อมูลแพ็กเกตร้องขอที่อยู่ในคิวรอการเชื่อมต่อและจะส่งแพ็กเกตละทิ้ง (reject) กลับไปยังผู้ร้องขอและเพื่อให้ผู้ใช้งานได้ทราบว่าส่งแพ็กเกตตอบปฏิเสธมาถึงเครือข่ายแล้ว ซึ่งถ้าผู้ร้องขอยังคงต้องการที่จะติดต่อสื่อสารส่งข้อมูลผ่านเครือข่ายก็จะต้องสร้างแพ็กเกตร้องขอและส่งเข้ามาร้องขออีกครั้ง ตามรูปที่ 3.2



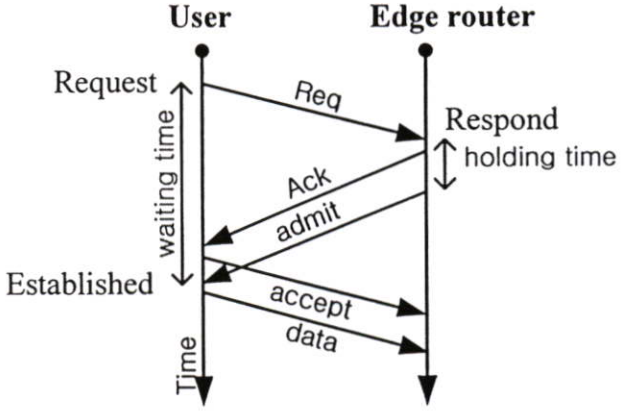
รูปที่ 3.2 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีผู้ร้องขอปฏิเสธ

ในกรณีที่ผู้ใช้งานยอมรับการหน่วงเวลาก็จะส่งแพ็กเกตตอบยอมรับ (accept) กลับไปยังส่วนควบคุมการจัดตั้งการเชื่อมต่อแล้วก็รอการจัดสรรแบนด์วิดธ์จากเครือข่าย เมื่อส่วนควบคุมการจัดตั้งการเชื่อม ต่อได้รับแพ็กเกตตอบยอมรับการหน่วงเวลาแล้ว ก็จะทำการให้แพ็กเกตร้องขอที่อยู่ในคิวรอการเชื่อมต่อรอการจัดสรรแบนด์วิดธ์จากส่วนควบคุมการจัดตั้งการเชื่อมต่อ ในการจัดสรรแบนด์วิดธ์ ในวิทยานิพนธ์นี้ได้คิดและทำการทดลอง 2 รูปแบบคือ การจัดสรรแบนด์วิดธ์ทันทีที่มีแบนด์วิดธ์ว่าง และการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา (อธิบายในหัวข้อที่ 3.1.3) เมื่อถึงเวลาที่จะจัดสรรแบนด์วิดธ์ให้แก่แพ็กเกตที่อยู่ในคิวรอการเชื่อมต่อแล้ว ส่วนควบคุมการจัดตั้งการ

เชื่อมต่อจะทำการส่งแพ็กเกตเตือนไปให้แก่ผู้ร้องขอที่ได้รับการจัดสรร และอนุญาตให้ผู้ร้องขอสามารถทำการเชื่อมต่อกับเครือข่ายและส่งข้อมูลได้ต่อไป ตามรูปที่ 3.3

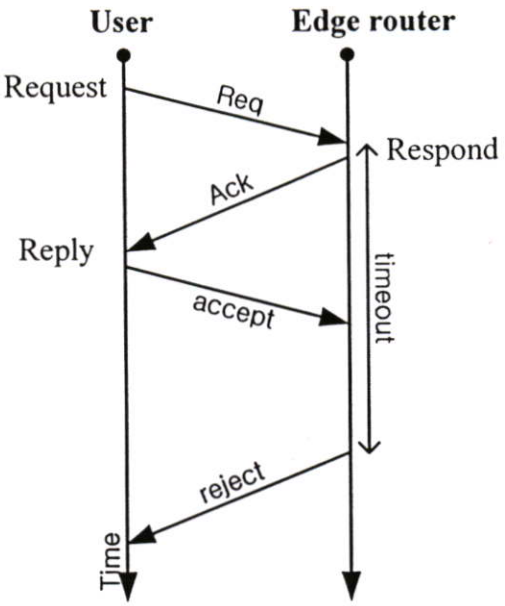


รูปที่ 3.3 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่ 1



รูปที่ 3.4 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่ 2

จากรูปที่ 3.4 จะเป็นการทำงานของส่วนควบคุมการจัดตั้งการเชื่อมต่อในกรณีที่มีการจัดสรรแบนด์วิดธ์ให้แก่ผู้ร้องขอเมื่อได้หน่วงแพ็กเกตร้องขอไว้ไม่นานและยังไม่ได้รับแพ็กเกตตอบรับจากผู้ใช้งาน โดยจะทำการส่งแพ็กเกตตอบยอมรับไปให้แก่ผู้ร้องขอตามไปทันทีและผู้ใช้งานสามารถทำการเชื่อมต่อได้ทันที ซึ่งในกรณีนี้จะใช้เวลาในการหน่วงแพ็กเกตในระยะเวลาที่สั้นกว่าในรูปที่ 3.3



รูปที่ 3.5 การทำงานของ CRDM ในสภาวะอิ่มตัวกรณีที่รอนานเกินกว่าเวลาในการรอคอย

ในรูปที่ 3.5 จะเป็นการทำงานของส่วนควบคุมการจัดตั้งการเชื่อมต่อเมื่อแพ็กเกตที่ถูกหน่วงเวลาไว้ในคิวรอการเชื่อมต่อนานเกินกว่าเวลาในการรอคอยที่ได้ตั้งไว้ ซึ่งเมื่อถึงเวลารอคอยที่ได้ตั้งไว้ส่วนควบคุมการจัดตั้งการเชื่อมต่อจะทำการลบแพ็กเกตที่ถูกหน่วงไว้ในคิวรอการเชื่อมต่อและส่งแพ็กเกตละทิ้งกลับไปยังผู้ร้องขอ ถ้าผู้ใช้งานยังคงต้องการที่จะติดต่อสื่อสารผ่านเครือข่ายก็จะต้องทำการสร้างแพ็กเกตร้องขอขึ้นมาใหม่แล้วส่งไปร้องขอยังเครือข่ายอีกครั้ง

3.1.3 การจัดสรรแบนด์วิดธ์

ในวิทยานิพนธ์นี้ได้ใช้การจัดสรรแบนด์วิดธ์ให้แก่แพ็กเกตร้องขอที่ถูกหน่วงเวลาตามที่ได้อธิบายในหัวข้อที่ผ่านมา ซึ่งมีอยู่ 2 แบบคือ การจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง และการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา

3.1.3.1 การจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง

หลังจากที่แพ็กเกตร้องขอได้ถูกหน่วงเวลาและตั้งเวลาในการรอคอยเอาไว้ในคิวรอการเชื่อมต่อแล้วก็จะต้องรอการจัดสรรแบนด์วิดธ์จากส่วนควบคุมการจัดตั้งการเชื่อมต่อ ซึ่งในวิธีนี้จะเป็นการรอคอยจนกว่าจะมีการเชื่อมต่อที่ใช้งานเสร็จแล้วและได้ปล่อยแบนด์วิดธ์กลับสู่เครือข่าย ต่อจากนั้นส่วนควบคุมการจัดตั้งการเชื่อมต่อก็จะทำการจัดสรรให้แก่แพ็กเกตร้องขอที่อยู่ในคิวรอ

การเชื่อมต่อตามลำดับการมาถึงก่อนหลัง (First Come First Served: FCFS) ในทันทีโดยจะส่งแพ็กเกตเตือนไปยังผู้ร้องขอให้สามารถทำการเชื่อมต่อกับเครือข่ายได้ทันที

3.1.3.2 การจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา

ในการจัดสรรแบนด์วิดธ์แบบโดยทันทีที่มีการปลดปล่อยกลับมาสู้เครือข่ายยังมีความไม่เสมอภาคเกิดขึ้น เนื่องจากบางแพ็กเกตมีความต้องการแบนด์วิดธ์จำนวนมากกว่าแพ็กเกตอื่นในการเชื่อมต่อ จึงถูกปฏิเสธการร้องขอจากเครือข่ายอยู่เรื่อย ๆ ในวิทยานิพนธ์นี้จึงได้คิดวิธีที่จะให้มีการรวบรวมแบนด์วิดธ์ในช่วงเวลาหนึ่งก่อนที่จะทำการจัดสรรให้แก่แพ็กเกตร้องขอที่ถูกหน่วงเวลาไว้ในคิวรอการเชื่อมต่อ ในวิธีนี้จะเป็นการนำอัลกอริทึมของการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่างมาทำการปรับปรุงเพื่อเพิ่มประสิทธิภาพ โดยให้ส่วนควบคุมอนุมัติการเชื่อมต่อทำการจัดสรรแบนด์วิดธ์ให้แก่แพ็กเกตร้องขอตามลำดับการมาถึงก่อนหลังเป็นช่วงคาบเวลาต่ำสุดเพื่อสามารถรวบรวมแบนด์วิดธ์ได้จำนวนหนึ่งและมากพอที่จะสามารถจัดสรรให้แก่การร้องขอที่ต้องการใช้แบนด์วิดธ์จำนวนมาก

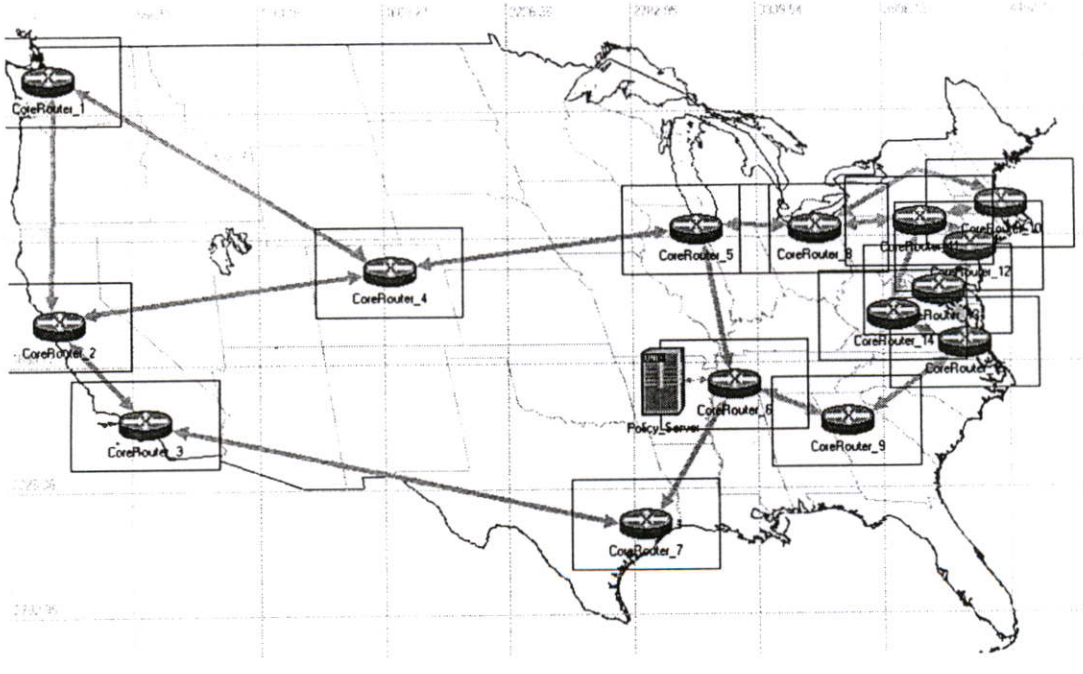
ตารางที่ 3.1 ผลการทดลองเมื่อใช้ช่วงคาบเวลาในการจัดสรรแบนด์วิดธ์

Period Time (sec) (Lifetime 240 sec)	Blocking Rate	Average Setup Time (sec)	CAC Signaling Traffic (Bytes)	Average Response Time (sec)
240	0.011524716	4.860347515	53127168	1.85401E-05
200	0.010054293	4.57568353	52984832	1.85401E-05
100	0.006195438	3.753727242	52917248	1.85401E-05
50	0.002935854	2.867760786	52942848	1.85401E-05
25	0.00229709	2.523446641	53152768	1.85401E-05
20	0.002573589	2.149222968	53075968	1.85401E-05
15	0.000884671	2.130281092	53174272	1.85401E-05
10	**0.00060367**	**1.793002075**	**53112832**	**1.85401E-05**
5	0.001005308	1.710762779	53335040	1.85401E-05
1	0.003058227	3.318515097	55015424	1.85414E-05
0.5	0.003740047	3.740515684	55499776	1.85403E-05

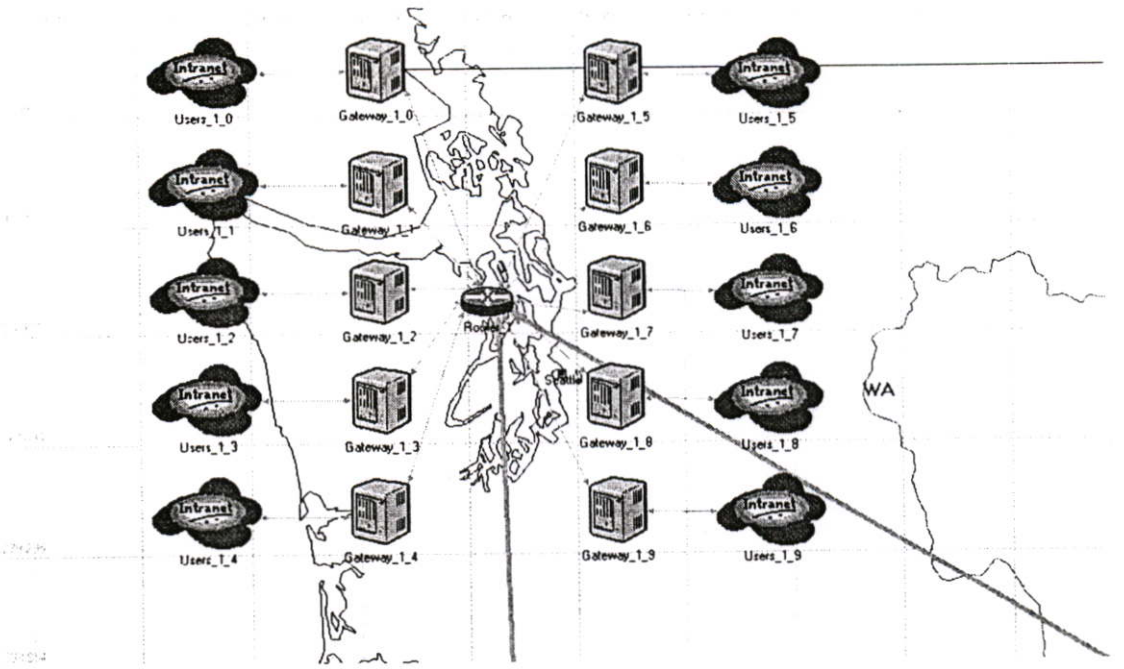
ซึ่งจากผลการทดลองในตารางที่ 3.1 แสดงให้เห็นว่าเมื่อได้ทำการเปลี่ยนช่วงคาบเวลาในการจัดสรรแบนด์วิดธ์ ทำให้อัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อและเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อมีการเปลี่ยนแปลง โดยในบางช่วงจะมีค่าที่เพิ่มขึ้นบางช่วงจะมีค่าที่ลดลง ในขณะที่ค่าพารามิเตอร์ทราฟฟิกที่เกิดจากสัญญาณการจัดตั้งการเชื่อมต่อมีค่าใกล้เคียงกัน ซึ่งค่าคาบเวลาที่มีอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อที่น้อยที่สุดและมีเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อที่ต่ำคือ ค่าคาบเวลาเท่ากับ 10 วินาที เนื่องจากวิทยานิพนธ์นี้ต้องการที่จะทดสอบประสิทธิภาพสูงสุดที่ได้จากการใช้วิธีการจัดตั้งการร้องขอแบบหน่วงเวลาการปฏิเสธ จึงได้เลือกใช้เวลาในการรอคอยสำหรับแพ็กเกตร้องขอที่ถูกหน่วงเวลาเท่ากับ 240 วินาที ซึ่งเป็นเวลาสูงสุดที่สามารถใช้ได้ในการจำลองการทำงานในระบบเครือข่าย

3.2 การสร้างระบบเครือข่ายบนโปรแกรมจำลองระบบเครือข่าย

ตามที่ได้อธิบายในบทที่ 2 ในวิทยานิพนธ์นี้ให้เลือกที่จะใช้ระบบเครือข่ายที่รับประกันคุณภาพในการให้บริการแบบ IntServ over DiffServ ของ Mei Yang [15] ขั้นแรกได้สร้างระบบเครือข่ายที่มีการทำงานแบบ IntServ over DiffServ ตามวิธีของ Mei Yang บนโปรแกรมจำลองระบบเครือข่าย OPNET ในระบบเครือข่ายตามรูปที่ 3.6 จะกำหนดให้มีเราเตอร์หลักทั้งหมด 15 ตัววางไว้ตามตำแหน่งและมีเซิร์ฟเวอร์ 1 ตัวซึ่งต่ออยู่กับเราเตอร์หลักตัวที่ 6



รูปที่ 3.6 เครือข่าย IntServ over DiffServ บนโปรแกรมจำลองระบบเครือข่าย [15]



รูปที่ 3.7 เราเตอร์หลักแต่ละตัวที่รองรับการเชื่อมต่อจากเราเตอร์ขอบและกลุ่มผู้ใช้งาน

ในรูปที่ 3.7 จะเป็นการเชื่อมต่อของเราเตอร์หลักแต่ละตัวที่จะรองรับการเชื่อมต่อจากเราเตอร์ขอบอีก 10 ตัว และที่เราเตอร์ขอบแต่ละตัวจะเชื่อมต่ออยู่กับกลุ่มผู้ใช้งาน ซึ่งกลุ่มผู้ใช้งานนี้จะเป็นส่วนที่ใช้ในการสร้างแพ็กเกตร้องขอและรับส่งข้อมูล

3.3 การสร้างโมเดลจำลองของอุปกรณ์ต่าง ๆ ในระบบเครือข่าย

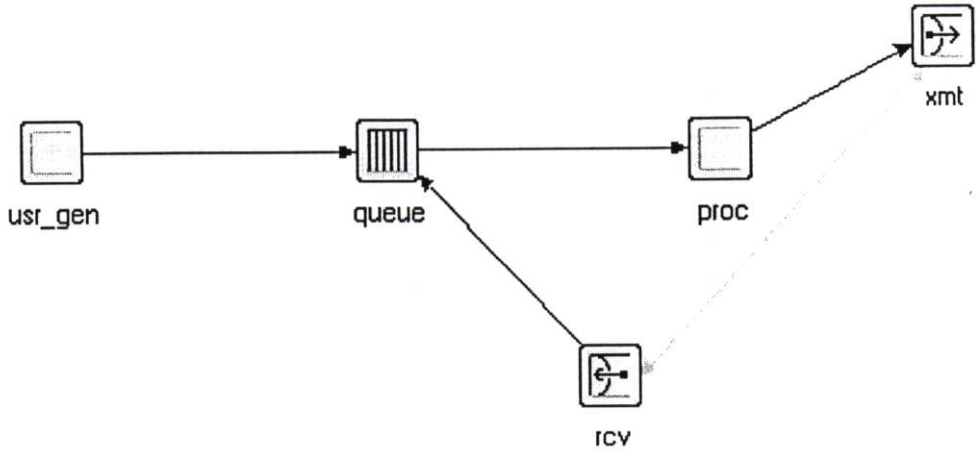
ในระบบเครือข่ายมีโมเดลจำลองของอุปกรณ์ต่าง ๆ คือ กลุ่มผู้ใช้งาน (LANs) เราเตอร์ขอบ (Edge Router) เราเตอร์หลัก (Core Router) ซึ่งในหัวข้อนี้จะอธิบายการทำงานของอุปกรณ์เหล่านี้

3.3.1 กลุ่มผู้ใช้งาน (LANs)

โมเดลของกลุ่มผู้ใช้งานจะเป็นส่วนที่สร้างแพ็กเกตซึ่งเปรียบเสมือนผู้ใช้งานต้นทางและปลายทาง โดยจะมีการทำงานหลัก ๆ คือ สร้างการร้องขอไปตามเราเตอร์ขอบต่าง ๆ โดยใช้วิธีการสุ่ม ซึ่งจะมีลักษณะตามภาพที่ 3.8 ในโปรแกรมจำลองระบบเครือข่าย

รูปที่ 3.8 กลุ่มผู้ใช้งานในโปรแกรมจำลองระบบเครือข่าย

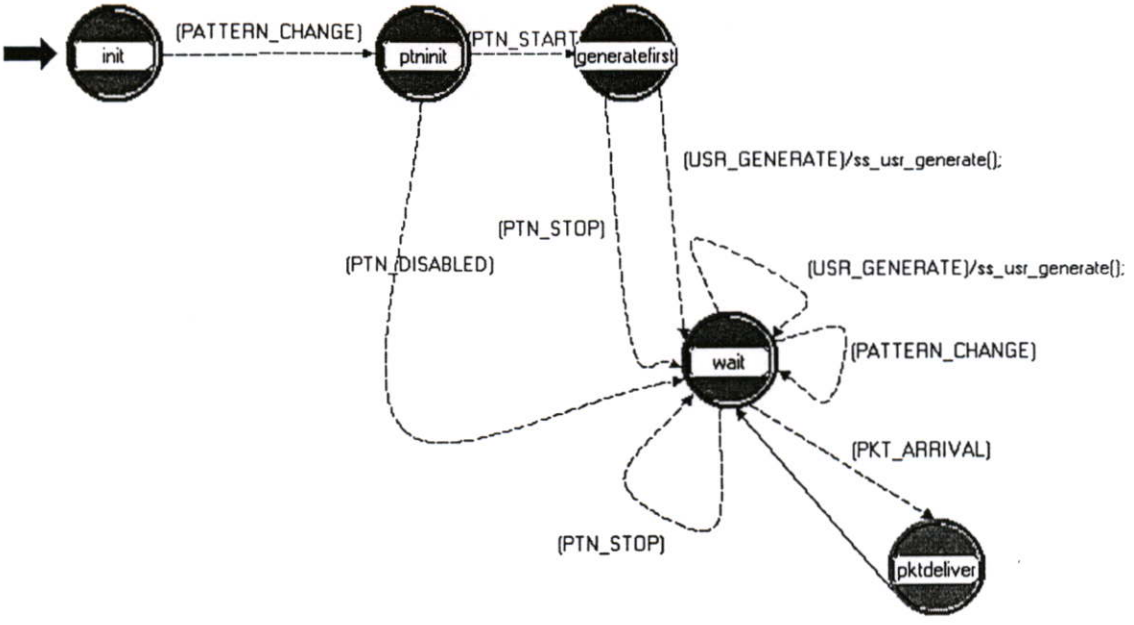
โดยภายในกลุ่มผู้ใช้งานนี้ จะมีโมดูลต่าง ๆ อยู่ภายในโหนดนี้ ซึ่ง usr_gen จะทำงานเป็นตัวสร้างแพ็กเกตร้องขอขึ้นมาตามลักษณะทราฟฟิกที่จะใช้ในการทดลอง แล้วจะส่งต่อให้ queue เพื่อเป็นตัวในการจัดลำดับตามการมาถึงก่อนหลังก่อนที่จะนำไปประมวลผลที่ proc แล้วจะทำการบันทึกค่าต่าง ๆ ไว้ก่อนจะนำส่งแพ็กเกตออกไปจากโหนดกลุ่มผู้ใช้งานทางพอร์ต xmt ซึ่งในโหนดกลุ่มผู้ใช้งานนี้จะสามารถรับแพ็กเกตข้อมูลได้ผ่านทางพอร์ต rcv แสดงดังรูปที่ 3.9



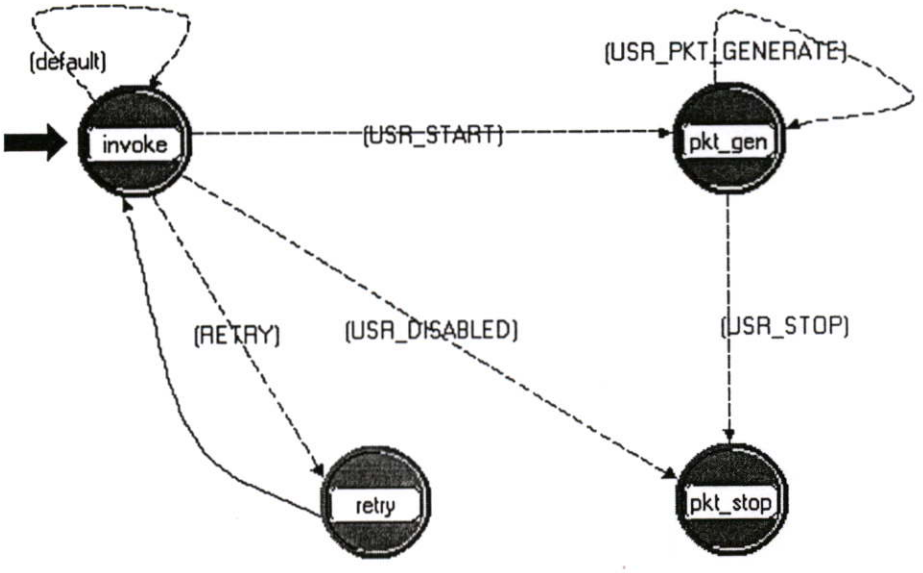
รูปที่ 3.9 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในกลุ่มผู้ใช้งาน

3.3.1.1 การทำงานของโมดูล usr_gen

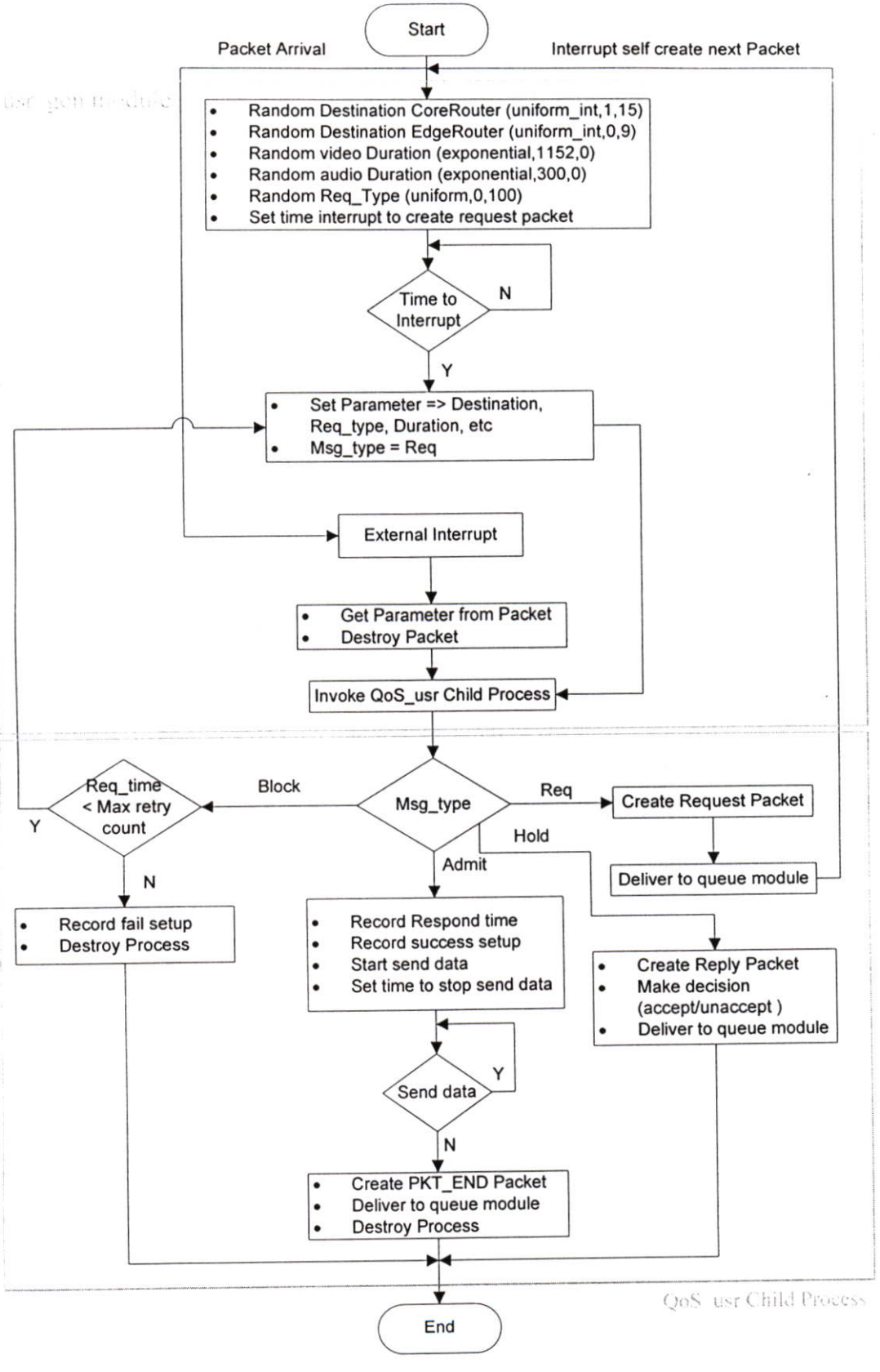
ในส่วนนี้จะทำงานเปรียบเสมือนเป็นผู้ใช้งานหลาย ๆ คนที่มีความต้องการที่จะใช้งานผ่านเครือข่าย โดยจะสร้างแพ็กเกตร้องขอขึ้นมาแล้วส่งไปยังเครือข่ายซึ่งความต้องการในการใช้ประเภทของทราฟฟิกจะมีการสุ่มขึ้นมา โดยภายในโมดูล usr_gen จะมีโครงสร้างการทำงานของ state ภายในดังรูปที่ 3.10 3.11 และลักษณะการทำงานดังรูปที่ 3.12



รูปที่ 3.10 โครงสร้างการทำงานของ state ภายในโมดูล usr_gen



รูปที่ 3.11 โครงสร้างการทำงานของ QoS_usr Child Process ที่ถูกเรียกขึ้นมาใช้งาน

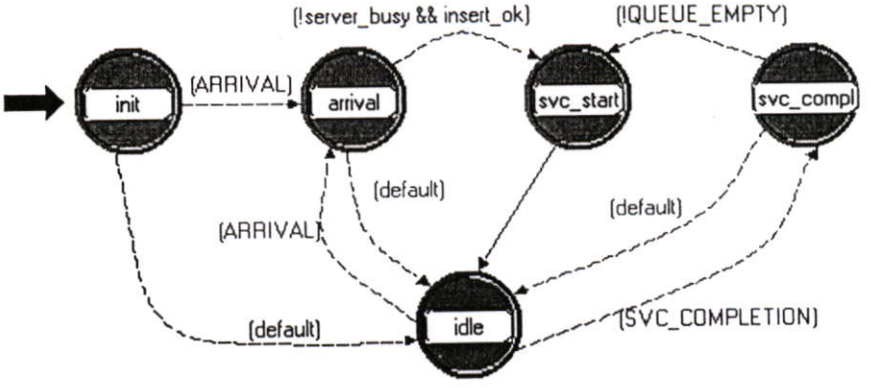


รูปที่ 3.12 ลักษณะการทำงานของโมดูล usr_gen

ส่วนโมดูล usr_gen จะเป็นกระบวนการภายในโหนดของกลุ่มผู้ใช้งานต้นทางและปลายทางมีโครงสร้างดังรูปที่ 3.10 ในขั้นแรกจะทำการสุ่มเลือกเราเตอร์ขอบและเราเตอร์หลักปลายทาง ต่อมาก็จะทำการสุ่มค่าระยะเวลาในการใช้งานของทราฟฟิกประเภทวิดิทัศน์จะใช้วิธีการสุ่มที่มีการกระจายตัวแบบเอกซ์โปเนนเชียลมีค่าเฉลี่ยเท่ากับ 1,152 วินาที (exponential, 1152, 0) และทราฟฟิกประเภทเสียงจะใช้วิธีการสุ่มที่มีการกระจายตัวแบบมีค่าเฉลี่ยเท่ากับ 300 วินาที (exponential, 300, 0) แล้วเก็บในค่าตัวแปรรอการเรียกใช้งาน ต่อมาก็จะสุ่มเลือกประเภทของทราฟฟิกและทำการตั้งค่าเวลาอินเทอร์รัปต์ (interrupt) เพื่อสร้างแพ็กเกตที่ใช้ในการร้องขอจัดตั้งเส้นทาง เมื่อถึงเวลาอินเทอร์รัปต์แล้วก็จะตั้งค่าตัวแปรต่าง ๆ ที่จำเป็นจากการสุ่มแล้วเรียกกระบวนการ (invoke process) QoS_usr เพื่อใช้ในการสร้างแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อขึ้นมา แล้วจะส่งต่อไปยังโมดูล queue แล้วจะไปรออยู่ในสถานะรอคอยการเข้ามาของแพ็กเกต ซึ่งใน QoS_usr Child Process จะถูกเรียกขึ้นมาทำงานอีกครั้งเมื่อมีแพ็กเกตถูกส่งเข้ามาที่โมดูล usr_gen และจะแยกไปทำงานตามประเภทของแพ็กเกตซึ่งมีแพ็กเกตร้องขอ (Req) แพ็กเกตอนุมัติ (Admit) แพ็กเกตปฏิเสธ (Block) และแพ็กเกตหน่วงเวลาการปฏิเสธ (Hold) ในกระบวนการสร้างแพ็กเกตร้องขอจะทำการสร้างแพ็กเกตร้องขอใหม่ขึ้น แล้วใส่ข้อมูลที่จำเป็นลงในแพ็กเกตตามที่ได้สุ่มขึ้นมาและส่งไปยังเราเตอร์ขอบเพื่อทำการร้องขอจัดตั้งการเชื่อมต่อ ถ้าได้รับแพ็กเกตอนุมัติมาจะทำการสร้างแพ็กเกตตอบสนอง (Respond) และแพ็กเกตเวลาในการจัดตั้งการเชื่อมต่อ (Setup time) แล้วส่งไปยังโมดูล proc เพื่อทำการบันทึกค่าสถิติ ต่อมาจะเริ่มทำการส่งข้อมูลไปยังปลายทางโดยได้ตั้งอินเทอร์รัปต์ไว้ตามเวลาที่ได้ขอ เมื่อส่งข้อมูลเสร็จสมบูรณ์แล้วจะทำการสร้างแพ็กเกตหยุดการเชื่อมต่อ (PKT_END) เพื่อส่งไปบอกถึงการหยุดส่งข้อมูลแล้วจะทำการปล่อยแบนด์วิดธ์กลับสู่เครือข่ายและจะทำการทำลายโพรเซส (process) นั้นทิ้ง ถ้าได้รับแพ็กเกตปฏิเสธมาจะทำการเปรียบเทียบจำนวนครั้งที่ได้ร้องขอไปแล้วกับจำนวนครั้งสูงสุดในการร้องขอที่กำหนดเอาไว้ ถ้ามีค่าน้อยกว่าก็จะทำตั้งค่าอินเทอร์รัปต์เพื่อส่งแพ็กเกตร้องขอครั้งต่อไป ถ้ามีค่ามากกว่าจะทำการสร้างแพ็กเกตเวลาในการจัดตั้งการเชื่อมต่อ แล้วส่งออกไปยังโมดูล proc ต่อมาก็จะทำการทำลายโพรเซสนั้นทิ้ง ถ้าได้รับแพ็กเกตหน่วงเวลาการปฏิเสธมาจะทำการสร้างแพ็กเกตตอบรับการหน่วงเวลาแล้วทำการตัดสินใจที่จะยอมรับการหน่วงเวลานั้นหรือไม่และจะส่งกลับไปยังเราเตอร์ขอบเพื่อให้ส่วนควบคุมการจัดตั้งการเชื่อมต่อทราบผลการตัดสินใจของผู้ใช้งาน

3.3.1.2 การทำงานของโมดูล queue

ในส่วนนี้จะเป็นจุดแรกที่แพ็กเกตที่เข้ามายังโหนดต่าง ๆ จะต้องผ่านแถวคอยเพื่อทำการจัดระเบียบก่อนที่จะไปทำการประมวลผลในส่วนถัดไป ซึ่งภายในโมดูล queue จะมีโครงสร้างการทำงานของ state ภายในดังรูปที่ 3.13



รูปที่ 3.13 โครงสร้างการทำงานของ state ภายในโมดูล queue

โมดูล queue จะทำหน้าที่เป็นแถวคอย (queueing) โดยจะมีเวลารอในแถวคอยเท่ากับสัดส่วนขนาดของแพ็กเกต (Packet Length) ต่ออัตราเร็วในการให้บริการข้อมูล (Service rate)

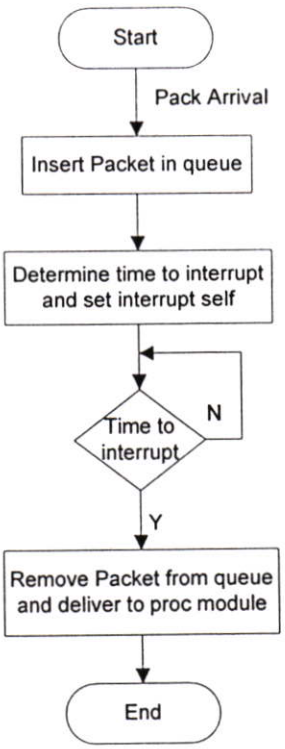
$$t_w = \frac{L_p}{t_x} \tag{3.1}$$

t_w เวลาที่รอในแถวคอย

L_p ขนาดของแพ็กเกต

t_x อัตราเร็วในการให้บริการข้อมูล

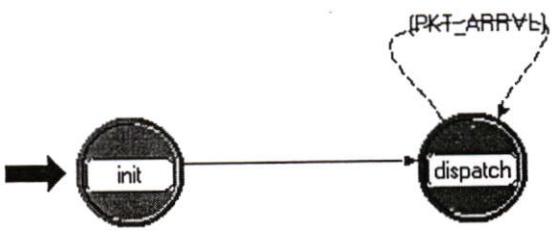
โมดูล queue จะอยู่ในโหนดต่าง ๆ ทุกโหนดในระบบเครือข่าย ซึ่งเมื่อแพ็กเกตถูกส่งมายังโหนดแล้ว แพ็กเกตทุกประเภทจะต้องรออยู่ในแถวคอยซึ่งเป็นเหมือนบัฟเฟอร์ขาเข้าที่เก็บแพ็กเกตไว้รอการบริการแพ็กเกตที่อยู่ในโมดูล proc อยู่ในขณะนี้ให้ได้รับการบริการที่เรียบร้อยก่อนจึงจะสามารถส่งแพ็กเกตที่อยู่ในคิวต่อไปไปยังโมดูล proc ได้ ซึ่งลักษณะการทำงานของโมดูล queue จะเป็นดังรูปที่ 3.14



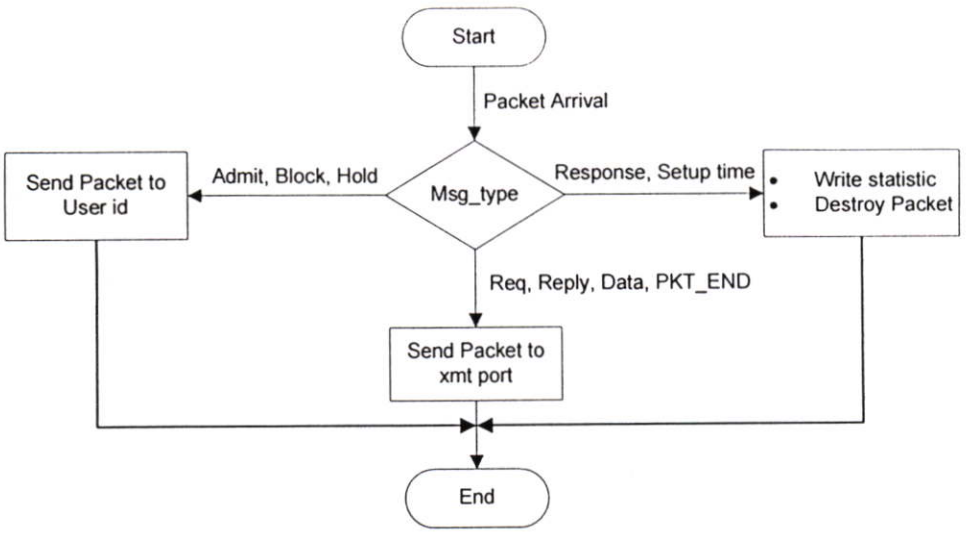
รูปที่ 3.14 ลักษณะการทำงานของโมดูล queue

3.3.1.3 การทำงานของโมดูล proc

ในโมดูล proc ที่อยู่ในกลุ่มผู้ใช้งานนี้จะทำหน้าที่เป็นตัวรับแพ็กเกตจากส่วนที่สร้างแพ็กเกตขึ้นมา แล้วส่งแพ็กเกตเข้าไปยังเครือข่ายที่เราเตอร์ขอบขาเข้า และหน้าที่อีกประการหนึ่งของส่วนนี้คือ ทำหน้าที่บันทึกสถิติต่าง ๆ ในการร้องขอจัดตั้งเส้นทางด้วย โดยจะมีโครงสร้างการทำงานของ state ภายในโมดูล proc ดังรูปที่ 3.15



รูปที่ 3.15 โครงสร้างการทำงานของ state ภายในโมดูล proc



รูปที่ 3.16 ลักษณะการทำงานของโมดูล proc

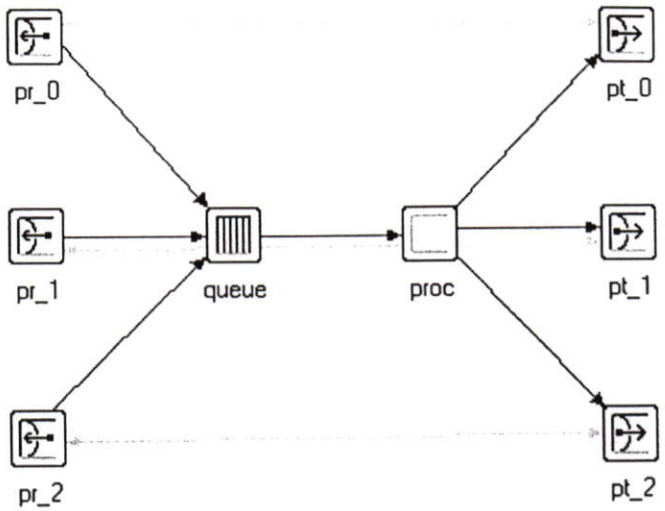
จากรูปที่ 3.16 ในเริ่มแรกกระบวนการทำงานจะไปรอการเข้ามาของแพ็กเกต เมื่อมีแพ็กเกตเข้ามาจะมีการตรวจสอบประเภทของแพ็กเกตโดยจะแบ่งการทำงานเป็น 3 กลุ่ม ในกลุ่มที่หนึ่งจะประกอบไปด้วยแพ็กเกต Req, Reply, data, PKT_END ซึ่งเมื่อได้รับแพ็กเกตมาจะทำงานโดยการส่งแพ็กเกตต่อไปยังปลายทางโดยผ่านพอร์ต xmt ในกลุ่มที่สองจะประกอบไปด้วยแพ็กเกต Response, Set_up ซึ่งเมื่อได้รับแพ็กเกตมาจะทำการบันทึกสถิติที่ส่งมาด้วยกับข้อมูลในแพ็กเกต และในกลุ่มที่สามจะประกอบไปด้วยแพ็กเกต Admit, Block, Hold ซึ่งเมื่อได้รับแพ็กเกตมาก็จะทำการส่งแพ็กเกตต่อไปยังผู้ใช้งาน

3.3.2 เราเตอร์ขอบ (Edge Router)

โมเดลจำลองของเราเตอร์ขอบนี้จะเป็นส่วนที่จะตัดสินใจที่จะรับหรือไม่รับการร้องขอที่มาจากผู้ใช้งานและทำการจัดการแบนด์วิดธ์ที่ใช้ในการส่งข้อมูล ซึ่งจะมีลักษณะตามรูปที่ 3.17 ในโปรแกรมจำลองระบบเครือข่าย

รูปที่ 3.17 เราเตอร์ขอบในโปรแกรมจำลองระบบเครือข่าย

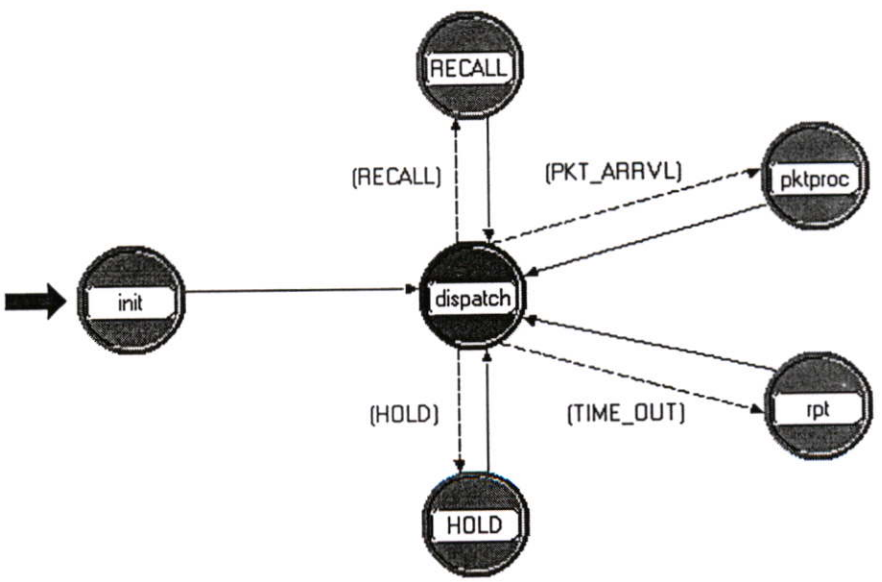
โดยภายในโหนดเราเตอร์ขอบนี้ จะมีโมดูลต่าง ๆ อยู่ภายในโหนดนี้ ซึ่งจะมีพอร์ตรับและพอร์ตส่งจำนวน 3 ตัว เท่ากันซึ่งพอร์ตรับจะเชื่อมอยู่กับโมดูล queue จะมีหน้าที่รับแพ็กเกตเข้ามาในโหนด ตามรูปที่ 3.18 โดยการทำงานของโมดูล queue จะมีการทำงานเหมือนกับโมดูล queue ในโหนดกลุ่มผู้ใช้งาน และได้อธิบายไว้แล้วในหัวข้อที่ 3.3.1.2



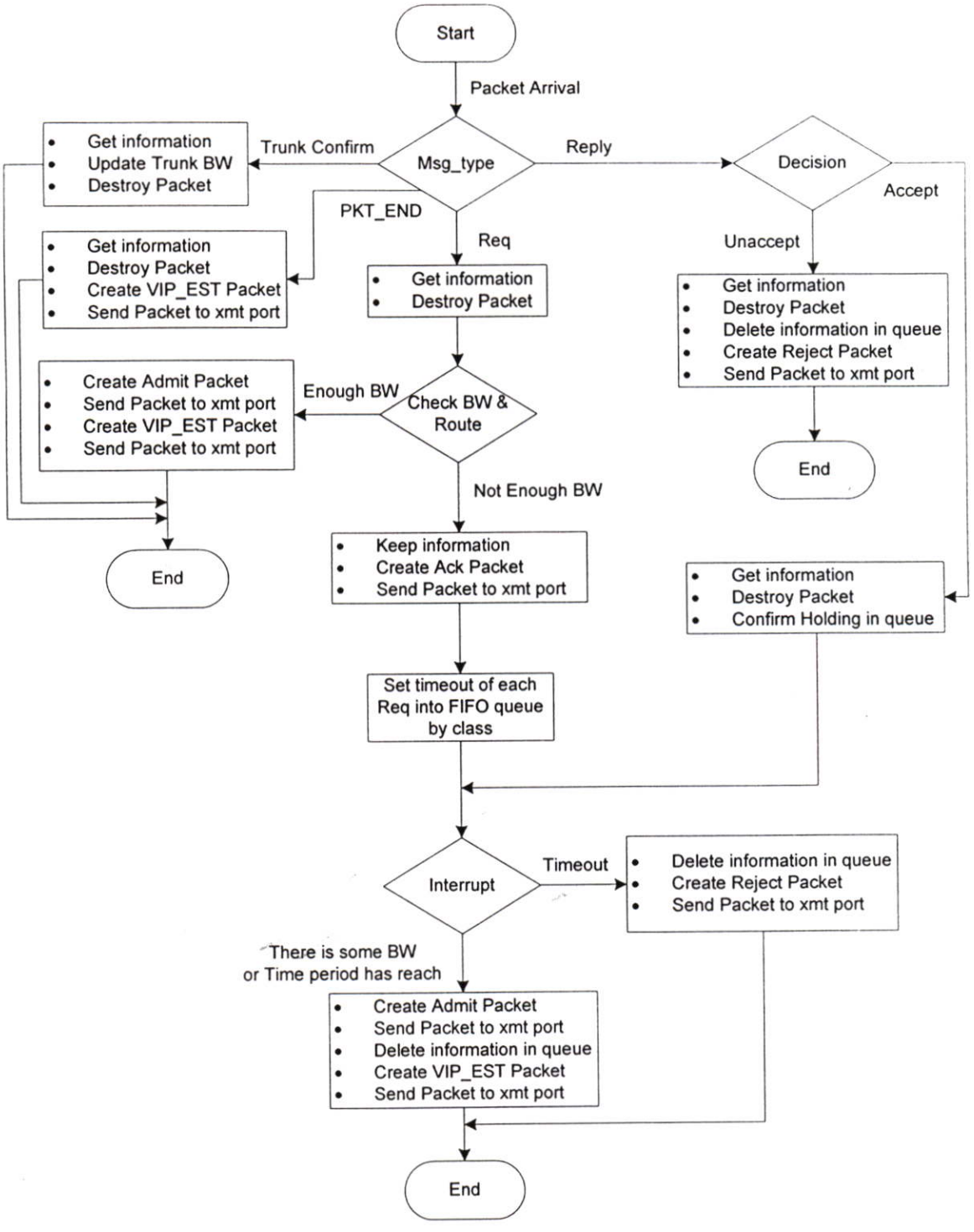
รูปที่ 3.18 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเราเตอร์ขอบ

3.3.2.1 การทำงานของโมดูล proc

ส่วนโมดูล proc ที่อยู่ในเราเตอร์ขอบนี้จะเป็นส่วนที่ควบคุมการร้องขอทั้งหมดที่เข้ามา หรือที่เรียกว่า ส่วนควบคุมการจัดตั้งการเชื่อมต่อ (Call Admission Control: CAC) ซึ่งจะควบคุมจำนวนผู้ที่จะเข้ามาใช้งานเครือข่าย จำนวนแบนด์วิดธ์ การรับประกันคุณภาพในการให้บริการ การรับหรือไม่รับการร้องขอ โดยจะมีโครงสร้างการทำงานของ state ต่าง ๆ ภายในโมดูล proc ตามรูปที่ 3.19



รูปที่ 3.19 โครงสร้างการทำงานของ state ภายในโมดูล proc



รูปที่ 3.20 ลักษณะการทำงานของโมดูล proc

เริ่มแรกกระบวนการทำงานจะไปรอการเข้ามาของแพ็กเกต เมื่อมีแพ็กเกตเข้ามาจะมีการตรวจสอบประเภทของแพ็กเกต ถ้าเป็นแพ็กเกต Trunk Confirm จะทำการรับข้อมูลจากแพ็กเกตแล้วนำข้อมูลนั้นไปใช้ในการปรับปรุงแบนด์วิดธ์ของ Trunk เมื่อเรียบร้อยแล้วจะทำลายแพ็กเกตทิ้ง ถ้า

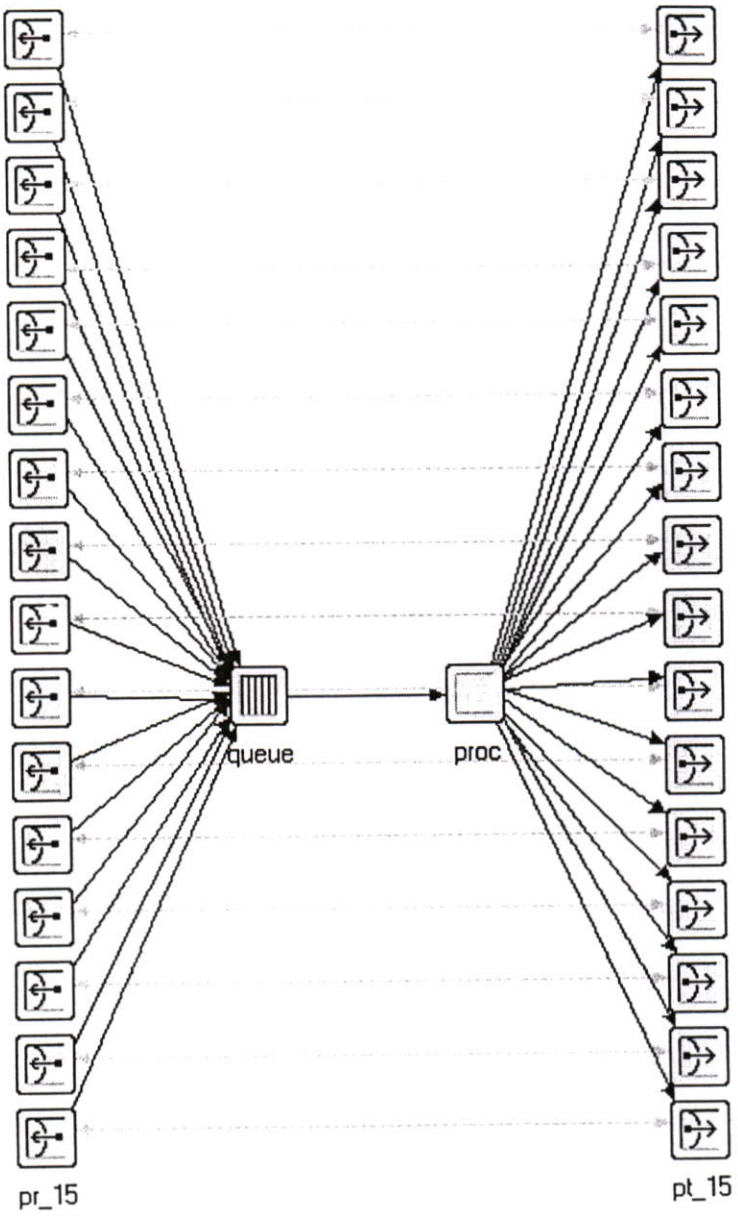
เป็นแพ็กเกต PKT_END จะทำกระบวนการรับข้อมูลจากแพ็กเกต แล้วจะทำการปล่อยแบนด์วิดธ์ที่เคยให้แก่ผู้ใช้งานนั้นกลับสู่เครือข่าย เมื่อเรียบร้อยแล้วจะทำลายแพ็กเกตนั้นทิ้ง ถ้าเป็นแพ็กเกต Reply จะมีการตรวจสอบการตัดสินใจที่ส่งมากับแพ็กเกต ถ้าตัดสินใจยอมรับการหน่วงเวลาจะทำการยืนยันการหน่วงเวลานั้นที่คิวรอการเชื่อมต่อตามข้อมูลที่ส่งมากับแพ็กเกตแต่ถ้าตัดสินใจไม่ยอมรับการหน่วงเวลาจะทำการลบแพ็กเกตที่ถูกหน่วงเวลาไว้ในคิวรอการเชื่อมต่อ และจะทำการสร้างแพ็กเกต Reject ส่งกลับไปยังผู้ใช้งาน ถ้าเป็นแพ็กเกตประเภท Req จะทำการรับข้อมูลจากแพ็กเกตไว้ แล้วทำการตรวจสอบแบนด์วิดธ์ตามที่ได้ร้องขอมากับแพ็กเกต ถ้ามีแบนด์วิดธ์เพียงพอจะทำการสร้างแพ็กเกต Admit ส่งกลับไปยังผู้ใช้งาน และแพ็กเกต VIP_EST ส่งไปยังเราเตอร์ต่าง ๆ ตามเส้นทางที่จะไปสู่ปลายทางเพื่อทำการสำรองทรัพยากร แต่ถ้ามีแบนด์วิดธ์ไม่เพียงพอที่จะให้ทำการเชื่อมต่อได้ก็จะเก็บข้อมูลของแพ็กเกตนั้นไว้แล้วสร้างแพ็กเกต Ack ตอบกลับไปยังผู้ใช้งาน ต่อมาก็จะตั้งเวลาในการรอคอยไว้ให้แก่แพ็กเกตแล้วเก็บแพ็กเกตไว้ในคิวรอการเชื่อม ต่อแบบมาถึงก่อนได้รับการบริการก่อน (First Come First Serve: FCFS) โดยแยกตามประเภทของบริการ แล้วก็จะมารอคอยการจัดสรรแบนด์วิดธ์ ในระหว่างการรอคอยถ้าเวลาในการรอคอยมาถึงแล้วก็จะทำการลบข้อมูลแพ็กเกตนั้นทิ้งไปจากคิวรอการเชื่อมต่อแล้วจะสร้างแพ็กเกต Reject ส่ง กลับไปยังผู้ใช้งาน แต่ถ้ามีแบนด์วิดธ์ว่างหรือเวลาในการจัดสรรมาถึงแล้ว (จัดสรรแบนด์วิดธ์เป็นช่วงคาบเวลา) ก็จะทำการสร้างแพ็กเกต Admit ส่งกลับไปยังผู้ใช้งานที่ได้รับการจัดสรร และสร้างแพ็กเกต VIP_EST ส่งไปสำรองแบนด์วิดธ์ที่เราเตอร์ต่าง ๆ จนถึงปลายทางและทำการลบข้อมูลออกจากคิวรอการเชื่อมต่อ

3.3.3 เราเตอร์หลัก (Core Router)

โมเดลจำลองของเราเตอร์หลักนี้จะทำหน้าที่เป็นส่วนในการสำรองแบนด์วิดธ์และส่งต่อแพ็กเกตข้อมูลไปยังเราเตอร์หลักปลายทางให้แก่ผู้ใช้งาน จะมีลักษณะตามภาพที่ 3.21 ในโปรแกรมจำลองระบบเครือข่าย

รูปที่ 3.21 เราเตอร์หลักในโปรแกรมจำลองระบบเครือข่าย

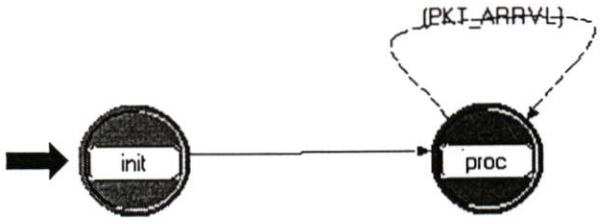
โดยภายในโหนดเราเตอร์หลักนี้จะมีโมดูลต่าง ๆ อยู่ภายในโหนดนี้ ซึ่งจะมีพอร์ตรับและพอร์ตส่งจำนวน 16 ตัว เท่ากัน ซึ่งพอร์ตรับจะเชื่อมอยู่กับโมดูล queue ซึ่งจะรับแพ็กเกตเข้ามาในโหนดตามรูปที่ 3.22 โดยการทำงานของโมดูล queue จะมีการทำงานเหมือนกับโมดูล queue ในโหนดกลุ่มผู้ใช้งาน และได้อธิบายไว้แล้วในหัวข้อที่ 3.3.1.2



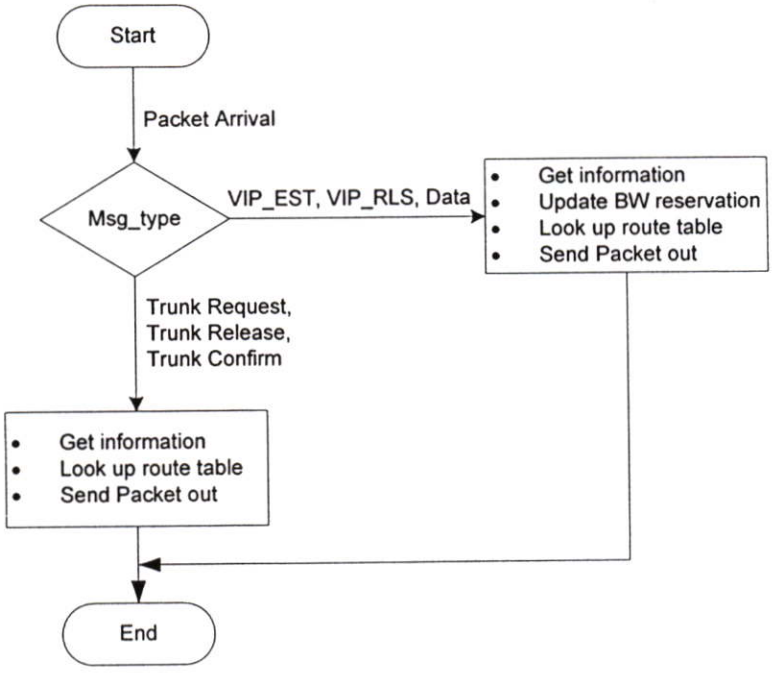
รูปที่ 3.22 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเราเตอร์หลัก

3.3.3.1 การทำงานของโมดูล proc

ในโมดูล proc ในเราเตอร์หลักนี้จะทำหน้าที่ในการส่งต่อแพ็กเกตข้อมูลไปยังปลายทาง และมีการจัดการสำรองแบนด์วิดธ์ตามลำดับชั้น (เช่น การสำรองแบนด์วิดธ์ให้แก่ provisioned link และ trunk) โดยจะมีโครงสร้างการทำงานของ state ต่าง ๆ ภายในโมดูล proc ตามรูปที่ 3.23



รูปที่ 3.23 โครงสร้างการทำงานของ state ภายในโมดูล proc



รูปที่ 3.24 ลักษณะการทำงานของโมดูล proc

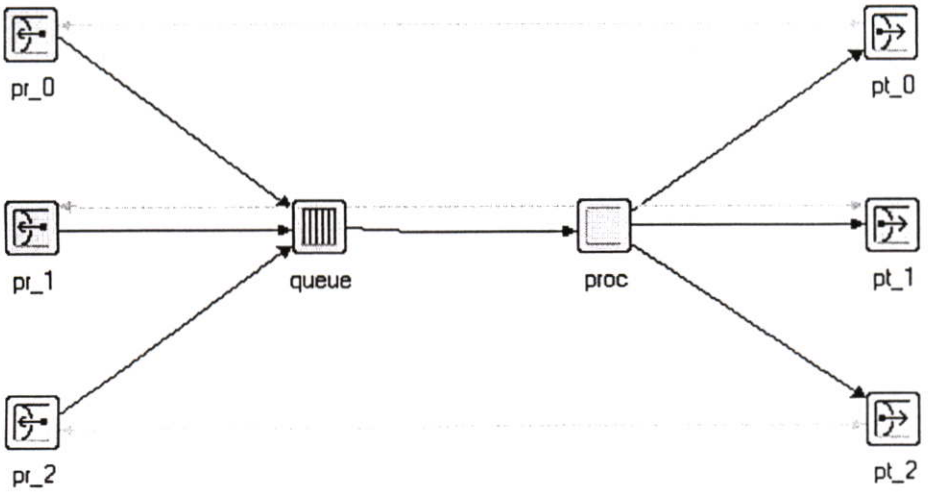
จากรูปที่ 3.24 เริ่มแรกกระบวนการทำงานจะไปรอการเข้ามาของแพ็กเกต เมื่อมีแพ็กเกตเข้ามาจะทำการตรวจสอบประเภทของแพ็กเกตซึ่งแบ่งออกเป็นสองกลุ่ม กลุ่มแรกจะมีแพ็กเกต VIP_EST, VIP_RLS ซึ่งจะทำการปรับปรุงการสำรองแบนด์วิดธ์ที่เปลี่ยนแปลง ถ้าเป็นแพ็กเกต data จะทำการส่งต่อไปจนถึงปลายทาง ในกลุ่มที่สองจะเป็นแพ็กเกต Trunk Request, Trunk Release, Trunk Confirm จะเป็นการส่งแพ็กเกตต่อไปจนถึงปลายทาง

3.3.4 เซิร์ฟเวอร์ (Server)

โมเดลจำลองของเซิร์ฟเวอร์นี้จะทำหน้าที่เป็นส่วนในการบันทึกการใช้งานแบนด์วิดธ์ทั้งหมดของเครือข่ายและเป็นส่วนกลางในการของใช้แบนด์วิดธ์จากส่วน provisioned link จะมีลักษณะตามภาพที่ 3.25 ในโปรแกรมจำลองระบบเครือข่าย

รูปที่ 3.25 เซิร์ฟเวอร์ในโปรแกรมจำลองระบบเครือข่าย

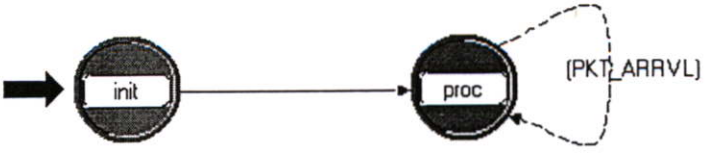
โดยภายในโหนดเซิร์ฟเวอร์นี้ จะมีโมดูลต่าง ๆ อยู่ภายในโหนดนี้ ซึ่งจะมีพอร์ตรับและพอร์ตส่งจำนวน 3 ตัว เท่ากัน ซึ่งพอร์ตรับจะเชื่อมอยู่กับโมดูล queue ซึ่งจะรับแพ็กเกตเข้ามาในโหนดตามรูปที่ 3.26 โดยการทำงานของโมดูล queue จะมีการทำงานเหมือนกับโมดูล queue ในโหนดกลุ่มผู้ใช้งาน และได้อธิบายไว้แล้วในหัวข้อที่ 3.3.1.2



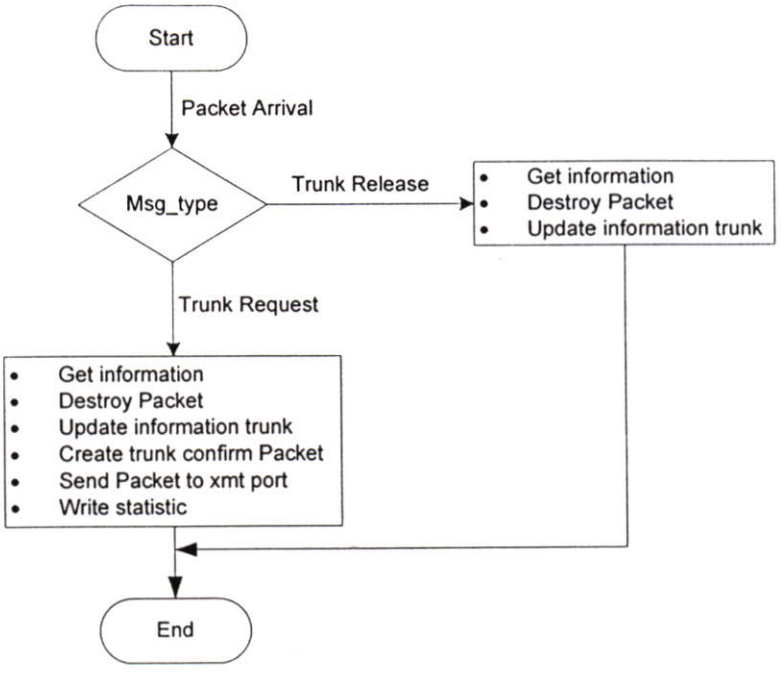
รูปที่ 3.26 โมเดลของโมดูลต่าง ๆ ที่อยู่ภายในเซิร์ฟเวอร์

3.3.4.1 การทำงานของโมดูล proc

ในโมดูล proc ในเซิร์ฟเวอร์นี้จะทำหน้าที่ โดยจะมีโครงสร้างการทำงานของ state ต่าง ๆ ภายในโมดูล proc ตามรูปที่ 3.27



รูปที่ 3.27 โครงสร้างการทำงานของ state ภายในโมดูล proc



รูปที่ 3.28 ลักษณะการทำงานของโมดูล proc

จากรูปที่ 3.28 เริ่มแรกมาจะทำกระบวนการอ่านข้อมูลจากไฟล์ตาราง Provision มาเก็บไว้ในตัวแปรของโปรแกรม แล้วจะรอการเข้ามาของแพ็กเกต เมื่อมีแพ็กเกตเข้ามาจะทำการตรวจสอบประเภทของแพ็กเกต ถ้าเป็นแพ็กเกต Trunk Request เข้ามา จะทำการรับข้อมูลจากแพ็กเกตเพื่อดูว่ามีการขอขยายแบนด์วิดธ์ trunk เท่าไรและจะสามารถให้ได้หรือไม่ แล้วจะทำการสร้างแพ็กเกต Confirm เพื่อตอบกลับไปยังผู้ที่ส่งมา แล้วทำการบันทึกสถิติการใช้งานแบนด์วิดธ์ที่เปลี่ยนแปลง ถ้าเป็นแพ็กเกต Trunk Release เข้ามา จะทำการรับข้อมูลจากแพ็กเกตเพื่อที่จะดูว่ามีการปล่อยแบนด์วิดธ์กลับคืนสู่เครือข่ายเท่าไร แล้วจะทำการปรับปรุงจำนวนแบนด์วิดธ์ที่เหลืออยู่ทั้งหมด แล้วทำการบันทึกสถิติการใช้งานแบนด์วิดธ์ที่เปลี่ยนแปลง

บทที่ 4

การทดลองและผลการทดลอง

เนื่องจากการวิจัยนี้เสนอวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธเป็นวิธีแบบใหม่ที่คิดขึ้นมาเพื่อที่จะลดปัญหาต่าง ๆ จึงจำเป็นที่จะต้องมีการจำลองการทำงานของวิธีในงานวิจัยนี้ลงบนโปรแกรมจำลองระบบเครือข่าย OPNET ในบทนี้จะกล่าวถึงค่าตัวแปรต่าง ๆ ที่ใช้ในการทำการทดลองรวมถึงค่าพารามิเตอร์ที่ใช้ในการวัดค่าประสิทธิภาพของวิธีที่นำเสนอ และผลการทดลองที่ได้จากการจำลองระบบเครือข่ายเปรียบเทียบวิธีการที่นำเสนอกับวิธีพื้นฐาน

4.1 การทดลอง

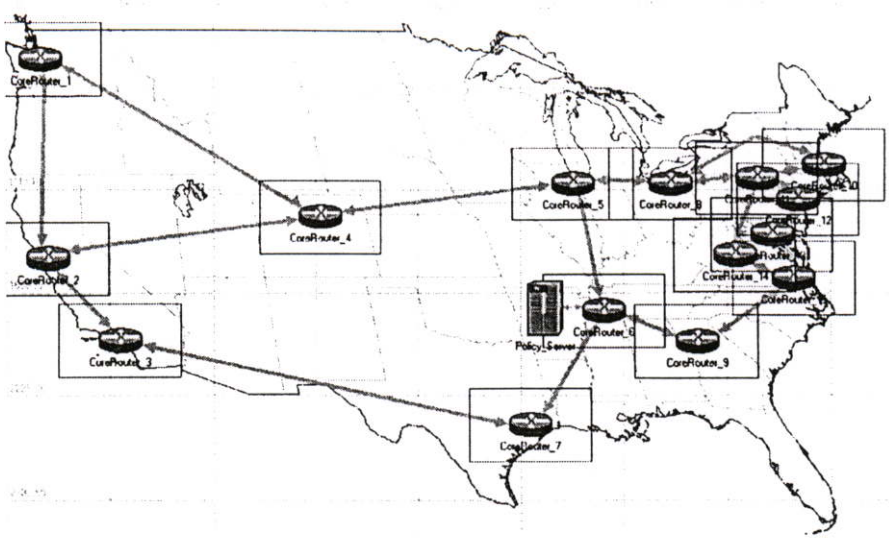
ตามที่ได้กล่าวมาแล้วในบทที่ 3 ถึงการสร้างระบบเครือข่ายในโปรแกรมจำลองตามรูปที่ 4.1 ซึ่งจะใช้ระบบสายส่งข้อมูลตามมาตรฐานที่เชื่อมระหว่างเราเตอร์หลักในโปรแกรมจำลองการทำงาน 2 แบบ คือ OC–48 และ OC-24 โดยได้ใช้ทราฟฟิกในการทดลอง 2 ประเภทคือ วิดิทัศน์และเสียง ซึ่งมีค่าพารามิเตอร์ที่ใช้ในการทดลองตามตารางที่ 4.1

- สายส่งข้อมูล OC-48 สามารถรองรับแบนด์วิดธ์ได้เท่ากับ 2.488 กิกะบิตต่อวินาที (Gigabits per second: Gbps) ซึ่งได้กำหนด Provisioned Link สำหรับทราฟฟิกประเภทวิดิทัศน์เท่ากับ 2.42 กิกะบิตต่อวินาที และกำหนด Provisioned Link สำหรับทราฟฟิกประเภทเสียงเท่ากับ 64 เมกะบิตต่อวินาที (Megabits per second: Mbps)
- สายส่งข้อมูล OC-24 สามารถรองรับแบนด์วิดธ์ได้เท่ากับ 1.244 กิกะบิตต่อวินาที ซึ่งได้กำหนด Provisioned Link สำหรับทราฟฟิกประเภทวิดิทัศน์เท่ากับ 1.21 กิกะบิตต่อวินาที และกำหนด Provisioned Link สำหรับทราฟฟิกประเภทเสียงเท่ากับ 34 เมกะบิตต่อวินาที

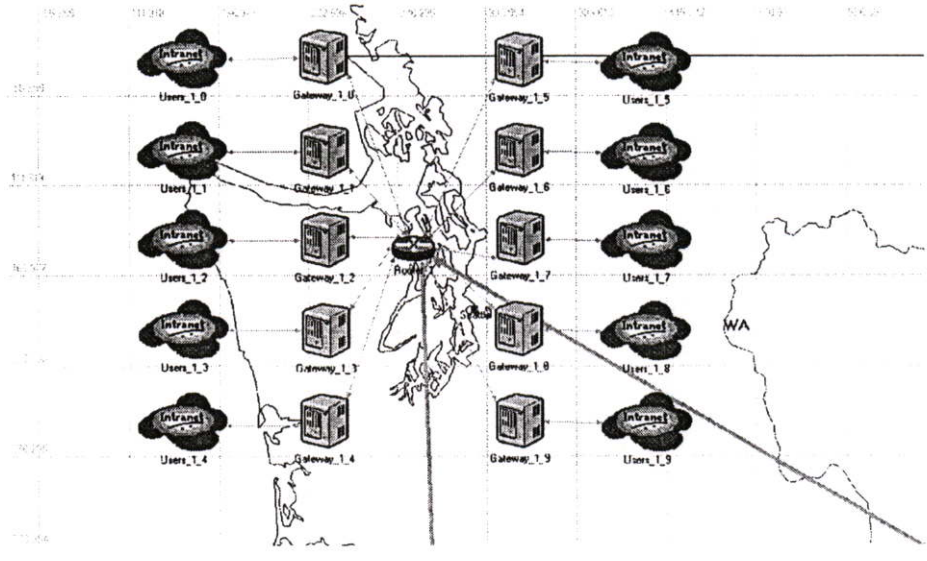
และสายส่งข้อมูลที่เชื่อมระหว่างเราเตอร์ขอบกับเราเตอร์หลัก และระหว่างเราเตอร์ขอบกับกลุ่มผู้ใช้งานคือ OC–12 ซึ่งสามารถรองรับแบนด์วิดธ์ได้เท่ากับ 622.08 เมกะบิตต่อวินาที ตามรูปที่ 4.2

ตารางที่ 4.1 พารามิเตอร์ของทราฟฟิกแต่ละประเภทใช้ในการทดลอง

Class	Bandwidth Requirement	Average Bandwidth Requirement	Connection Duration	Average Connection Duration	Example
A	64 Kbps (CBR)	-	1 – 10 นาที	300 วินาที	VoIP
B	1 – 6 Mbps	3 Mbps	5 นาที – 5 ชั่วโมง	1,152 วินาที	Video MPEG-4



รูปที่ 4.1 โครงสร้างของเครือข่ายที่ใช้ในการทดลอง



รูปที่ 4.2 โครงสร้างการเชื่อมต่อระหว่างผู้ใช้งาน เราเตอร์ขอบ และเราเตอร์หลัก

และมีค่าพารามิเตอร์อื่น ๆ ที่เหมือนกันในโปรแกรมจำลองระบบเครือข่าย มีดังต่อไปนี้

- ความยาวของแพ็กเกต 1,024 ไบต์
- ระยะเวลาที่ใช้สร้างการร้องขอ (inter arrival times) จะใช้กระบวนการสุ่มที่มีการกระจายตัวแบบเอกซ์โปเนนเชียลที่มีค่าเฉลี่ยเท่ากับ 50 วินาที (exponential distribution with mean 50 seconds)
- ค่าเวลาในการรอคอยเท่ากับ 240 วินาที
- จำนวนครั้งในการพยายามสร้างการร้องขอเท่ากับ 10 ครั้ง
- ระยะเวลาระหว่างการพยายามสร้างการร้องขอเท่ากับ 15 วินาที

4.1.1 ค่าพารามิเตอร์ที่ใช้ในการวัดประสิทธิภาพ

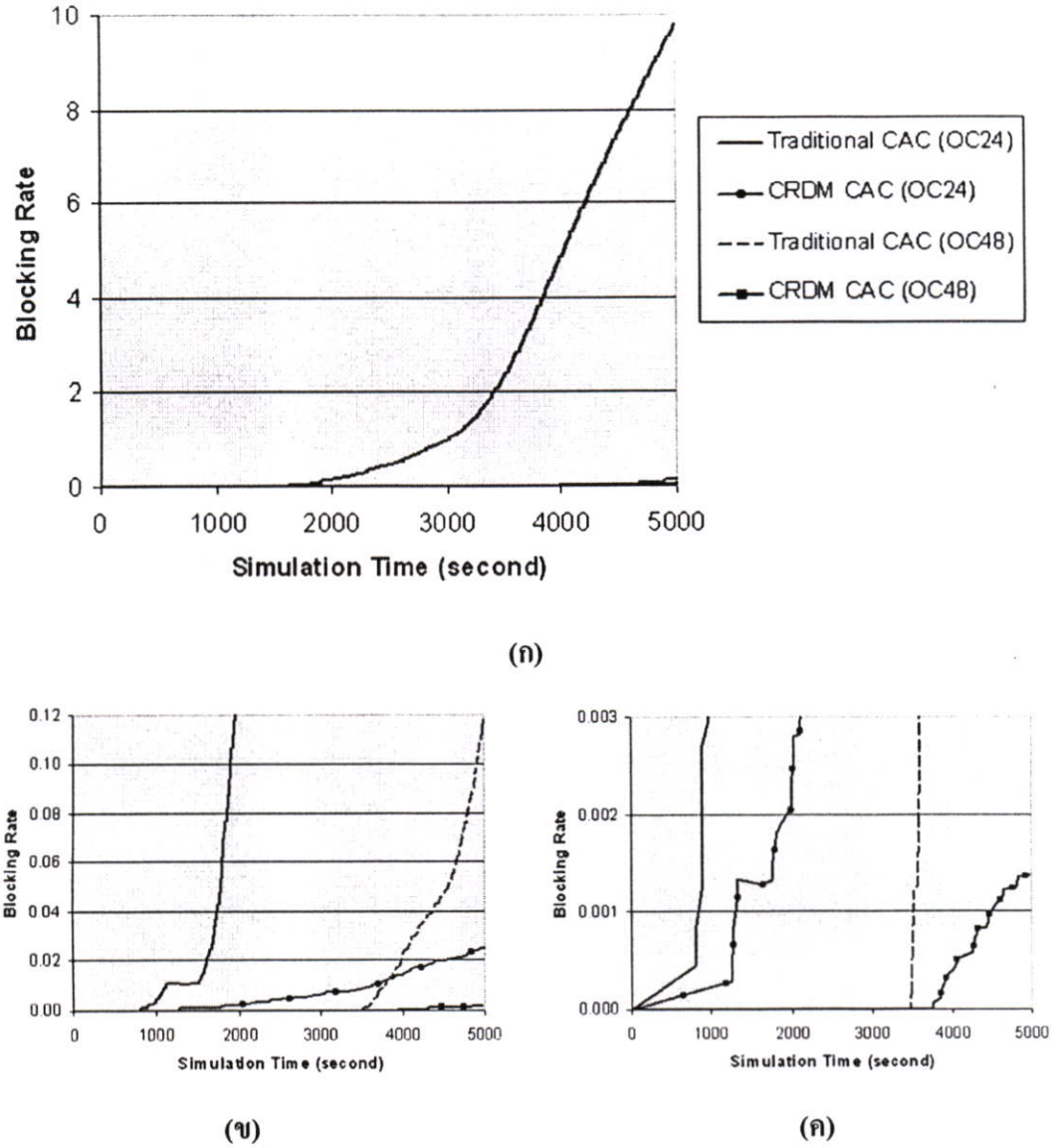
- *อัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ (Blocking Rate)* คือ อัตราส่วนระหว่างจำนวนการร้องขอจัดตั้งการเชื่อมต่อที่ถูกปฏิเสธต่อจำนวนการร้องขอจัดตั้งการเชื่อมต่อทั้งหมด
- *เวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อ (Average Connection Setup Times)* คือ เวลาเฉลี่ยตั้งแต่ผู้ร้องขอได้ส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อไปยังเครือข่ายจนถึงเวลาที่ได้รับการอนุมัติให้ทำการเชื่อมได้
- *ปริมาณทราฟฟิกจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อ (Call Admission Control Signaling Traffic)* คือ ปริมาณทราฟฟิกที่เกิดขึ้นจากการร้องขอจัดตั้งการเชื่อมต่อตั้งแต่ผู้ร้องขอได้เริ่มส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อทั้งหมดจนถึงแพ็กเกต อนุมัติให้ทำการเชื่อมต่อได้ เช่น แพ็กเกตร้องขอ แพ็กเกตตอบรับ และแพ็กเกตอนุมัติ เป็นต้น
- *ประสิทธิภาพการใช้งานแบนด์วิดธ์ (Bandwidth Utilization)* คือ อัตราส่วนระหว่างจำนวนการใช้งานแบนด์วิดธ์ในขณะนั้นต่อจำนวนแบนด์วิดธ์ที่มีอยู่ทั้งหมดโดยจะคิดเป็นร้อยละ
- *เวลาในการตอบสนองเฉลี่ย (Average Response Times)* คือ เวลาตั้งแต่ส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อไปยังเครือข่ายและได้รับการตอบกลับมาจากเครือข่ายเป็นครั้งแรก (acknowledge)

4.1.2 ลำดับในการทดลอง

- ***การทดลองที่ 1*** จะเป็นการทดลองเพื่อทดสอบประสิทธิภาพวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่างโดยเปรียบเทียบกับวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐาน ซึ่งได้ทำการทดลองโดยใช้สายส่งข้อมูลที่เชื่อมระหว่างเราเตอร์หลักคือ OC-48 และ OC-24
- ***การทดลองที่ 2*** จะเป็นการทดลองหาค่าเวลาที่เหมาะสมที่จะใช้สำหรับของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา โดยผลการทดลองที่ได้จากค่าคาบเวลาที่เหมาะสมที่สุดมาเปรียบเทียบกับวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานและวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง ซึ่งใช้สายส่งข้อมูล OC-48 เชื่อมระหว่างเราเตอร์หลัก

4.2 ผลการทดลอง

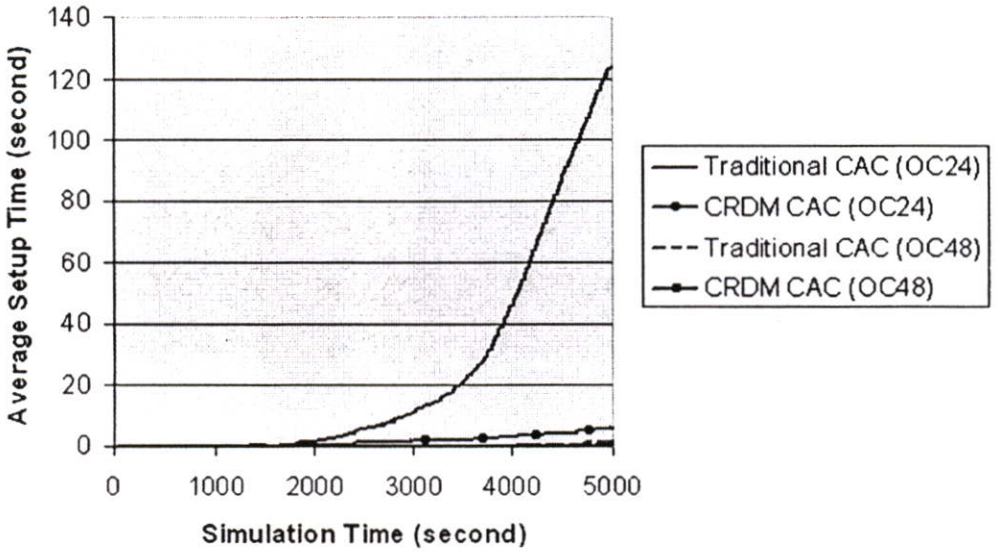
4.2.1 ผลการทดลองที่ 1



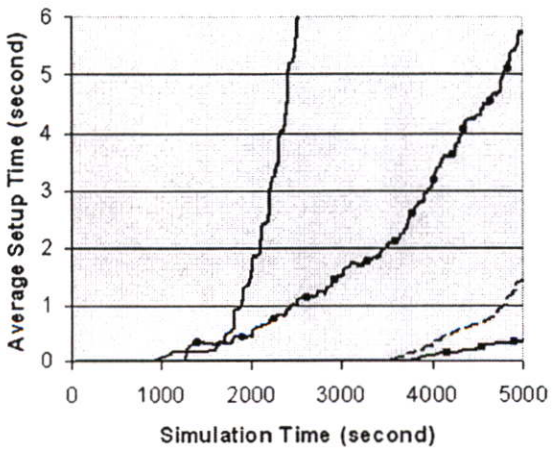
รูปที่ 4.3 กราฟอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ

ผลการทดลองในรูปที่ 4.3 เป็นการเปรียบเทียบอัตราการปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานกับอัตราการปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง จากผลการทดลองเมื่อใช้สาบส่งข้อมูล OC-48 พบว่าจะเริ่มเกิดการปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อตั้งแต่ในวินาทีที่ 3500 ของการทดลอง เพราะเครือข่ายเริ่มเข้าสู่ในสภาวะอิ่มตัวและยังมีผู้ใช้งานได้ส่งแพ็กเกตร้องขอเข้ามายังเครือข่ายเรื่อย ๆ ซึ่งวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธสามารถลดอัตราการปฏิเสธการร้องขอจัดตั้ง

การเชื่อมต่อจากวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานได้ประมาณ 98.834 % เนื่องมาจากวิธีที่นำเสนอจะเป็นการเพิ่มโอกาสให้แก่แพ็กเกตร้องขอจัดตั้งการเชื่อมต่อที่จะได้รับการจัดสรรแบนด์-วิดธ์จึงทำให้การปฏิเสธแพ็กเกตการร้องขอจัดตั้งการเชื่อมต่อลดลงตามไปด้วย เมื่อได้ทดลองเปลี่ยนสายส่งข้อมูลจาก OC-48 มาเป็น OC–24 แล้ว จะพบว่าวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานจะมีอัตราปฏิเสธการร้องขอที่สูงขึ้นกว่าการทดลองที่ใช้สายส่งข้อมูล OC–48 เนื่องจากเมื่อเครือข่ายมีแบนวิดธ์ที่ลดลงทำให้เกิดการอิ่มตัวของเครือข่ายที่เร็วขึ้นและยังคงมีผู้ใช้งานทำการร้องขอการเชื่อมต่อมาเรื่อย ๆ จึงทำให้มีอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อที่สูงขึ้น แต่อย่างไรก็ตามผลการทดลองของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธยังคงสามารถลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อลงจากวิธีพื้นฐานได้ประมาณ 99.734 %



(ก)

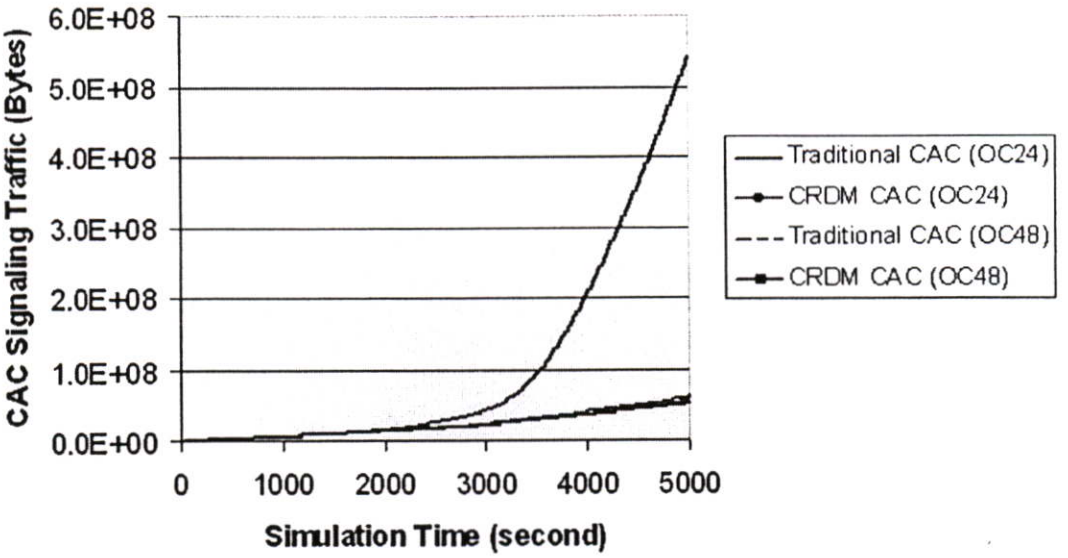


(ข)

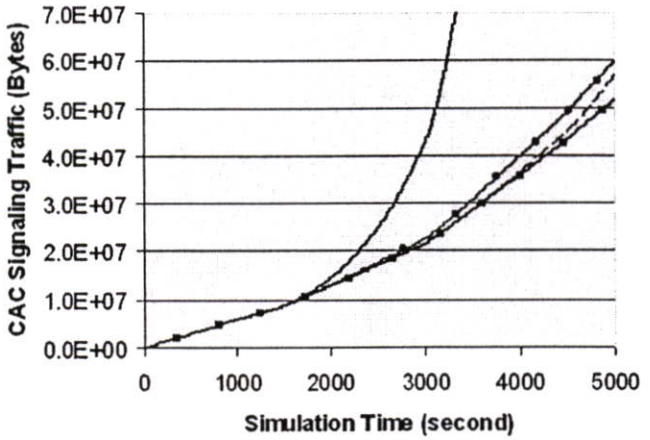
รูปที่ 4.4 กราฟเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อ

ผลการทดลองในรูปที่ 4.4 เป็นการเปรียบเทียบเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อของวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานกับแบบวิธีหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง จากผลการทดลองเมื่อได้ใช้สายส่งข้อมูล OC-48 พบว่าวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธจะใช้เวลาเฉลี่ยในการรอเพื่อจัดตั้งการเชื่อมต่อน้อยกว่าวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานประมาณ 75.593 % เนื่องจากวิธีที่นำเสนอจะมีการจัดคิวรอการเชื่อมต่อให้แก่แพ็กเกตร้องขอจัดตั้งการเชื่อมต่อ เมื่อเครือข่ายมีแบนด์วิดธ์ว่างขึ้นมาก็จะสามารถทำการจัดสรรแบนด์วิดธ์ให้แก่แพ็กเกตที่รออยู่ได้เลย แต่ในวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานผู้ใช้งานแต่ละคนจะต้องพยายามส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อมายังเครือข่ายซึ่งต้องอาศัยโอกาสจังหวะเวลาที่เครือข่ายมีแบนด์วิดธ์ว่างพอดีถึงจะสามารถทำการเชื่อมต่อได้ เมื่อได้ทดลองเปลี่ยนสายส่งข้อมูลจาก OC-48 มาเป็น OC–24 แล้ว จะพบว่าวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานจะมีเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อที่สูงขึ้นมากกว่าการทดลองที่ใช้สายส่งข้อมูล OC–48 เนื่องมาจากเครือข่ายมีแบนวิดธ์ที่ลดลงและเมื่อเกิดการอิ่มตัวของเครือข่ายทำให้ผู้ใช้งานเกิดการรอคอยที่จะเชื่อมต่อกับเครือข่ายนานขึ้น ต่อมาเมื่อได้นำวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาปฏิเสธมาใช้กับเครือข่าย ทำให้สามารถลดเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อลงได้จากวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานประมาณ 95.386 %

ผลการทดลองในรูปที่ 4.5 เป็นการเปรียบเทียบปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อของวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานกับวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ จากผลการทดลองเมื่อได้ใช้สายส่งข้อมูล OC-48 พบว่าวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธสามารถลดทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อได้ประมาณ 9.147 % เมื่อเทียบกับวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐาน เนื่องจากวิธีที่นำเสนอเป็นการลดจำนวนแพ็กเกตการร้องขอจัดตั้งการเชื่อมต่อที่ผู้ใช้งานจะต้องพยายามส่งซ้ำ ๆ ได้ เมื่อได้ทดลองเปลี่ยนสายส่งข้อมูลจาก OC-48 มาเป็น OC–24 แล้ว จะพบว่าวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานจะมีอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อที่สูงขึ้นเรื่อย ๆ ทำให้เกิดการส่งแพ็กเกตร้องขอซ้ำ ๆ มายังเครือข่าย เป็นผลให้ปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อมากตามไปด้วยดังผลการทดลองในรูปที่ 4.5 (ก) และเมื่อได้นำวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาปฏิเสธมาใช้กับเครือข่ายทำให้สามารถลดปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อได้ประมาณ 88.966 % เมื่อเปรียบเทียบกับวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐาน



(ก)

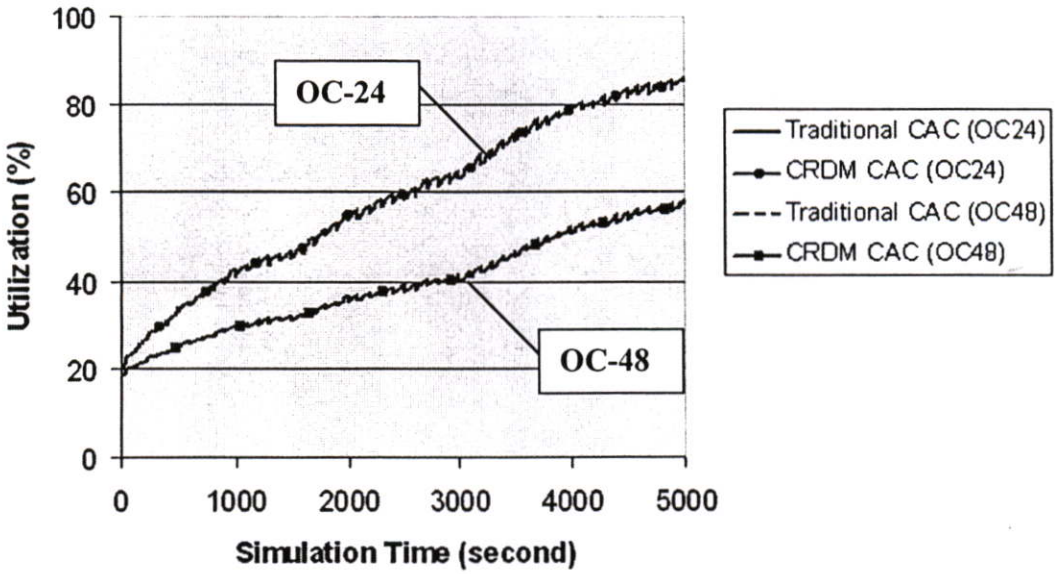


(ข)

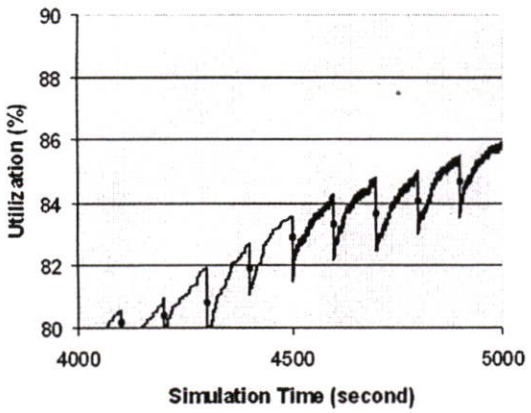
รูปที่ 4.5 กราฟปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อ

ผลการทดลองในรูปที่ 4.6 เป็นการเปรียบเทียบประสิทธิภาพการใช้งานแบนด์วิดธ์ของวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานกับวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ จากผลการทดลองเมื่อใช้สายส่งข้อมูล OC-48 พบว่าทั้งสองวิธีมีประสิทธิภาพการใช้งานแบนด์วิดธ์ไม่แตกต่างกัน เนื่องจากในวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานเมื่อผู้ใช้งานถูกปฏิเสธการร้องขอไปแล้วก็จะพยายามสร้างและส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อมาเรื่อย ๆ พอเครือข่ายมีแบนด์วิดธ์ว่างก็จะถูกใช้งานในทันที ส่วนวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธจะให้แพ็กเกตร้องขอรออยู่ในคิวรอการเชื่อมต่ออยู่แล้ว เมื่อเครือข่ายมีแบนด์วิดธ์ว่างก็จะสามารถให้ผู้ใช้งานทำการเชื่อมต่อได้เลย และเมื่อได้ทดลองเปลี่ยนสายส่งข้อมูลจาก OC-48 มาเป็นสายส่งข้อมูล OC-24

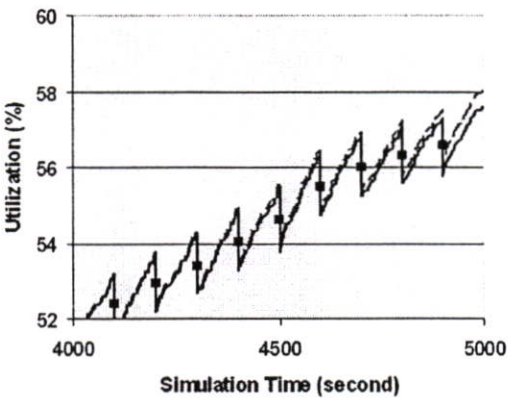
พบว่าทั้งสองวิธีมีประสิทธิภาพการใช้งานแบนด์วิดธ์ของเครือข่ายไม่แตกต่างกัน แต่มีค่ามากกว่าเครือข่ายที่ใช้สายส่งข้อมูล OC-48 เนื่องมาจากเมื่อเปลี่ยนสายส่งข้อมูลมาเป็น OC-24 จะทำให้เครือข่ายมีแบนด์วิดธ์ที่ลดลง ในขณะที่ใช้ค่าพารามิเตอร์ของทราฟฟิกเดียวกันกับเครือข่ายที่ใช้สายส่งข้อมูล OC-48 จึงทำให้ประสิทธิภาพการใช้งานแบนด์วิดธ์ที่สูงขึ้น



(ก)



(ข)

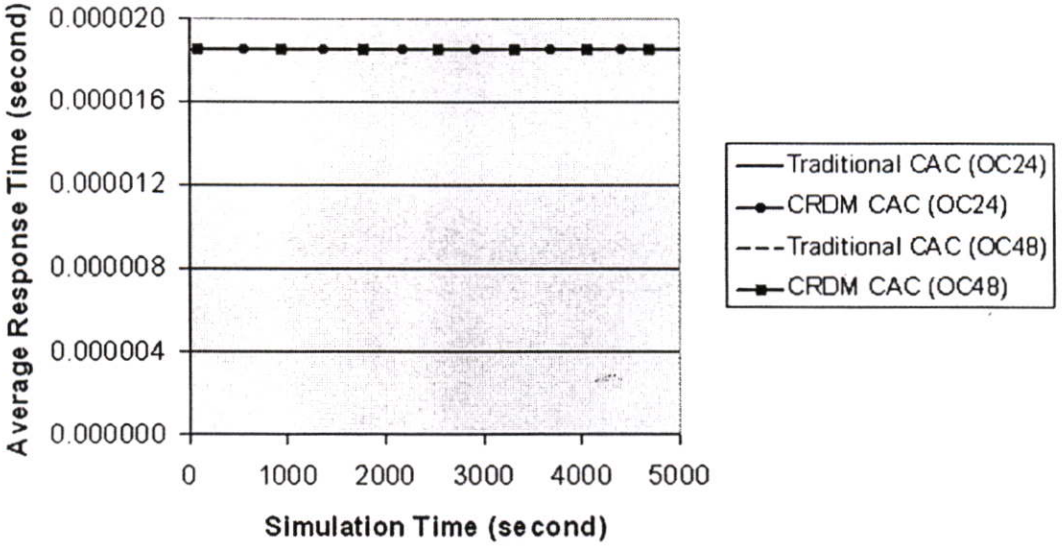


(ค)

รูปที่ 4.6 กราฟประสิทธิภาพการใช้งานแบนด์วิดธ์

ผลการทดลองในรูปที่ 4.7 เป็นการเปรียบเทียบเวลาเฉลี่ยในการตอบสนองจากเครือข่ายของวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานกับวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธของเครือข่ายเมื่อใช้สายส่งข้อมูล OC-24 และสายส่งข้อมูล OC-48 จากผลการทดลองแสดงให้เห็นว่า

วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธใช้เวลาในการตอบกลับแพ็กเกตเท่ากับวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐาน เนื่องจากวิธีที่นำเสนอเมื่อนำมาใช้งานร่วมกับเครือข่ายจำลองแล้วจะไม่ทำให้มีการประมวลผลช้าลง



รูปที่ 4.7 กราฟเวลาในการตอบสนองเฉลี่ย

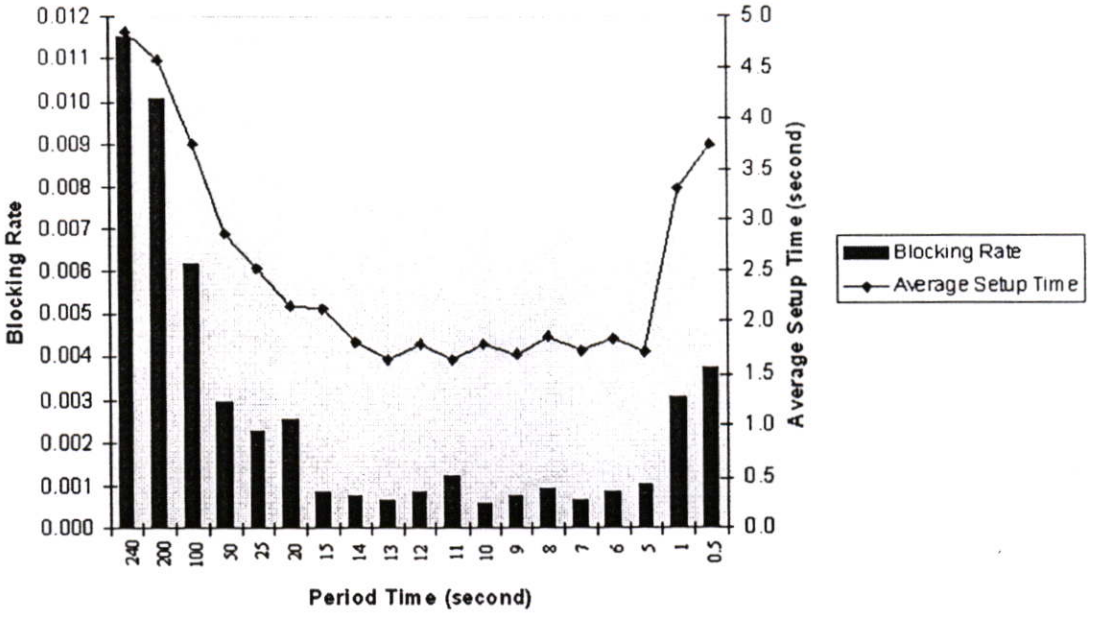
จากผลการทดลองที่ผ่านมาโดยการใช้สายส่งข้อมูล OC-48 ในการทดลองนี้ ได้แสดงให้เห็นแล้วว่าวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธสามารถที่จะลดอัตราการปฏิเสธการร้องขอ เวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อ และปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อได้ดังที่ได้แสดงประสิทธิภาพแล้วใน [27]

4.2.1 ผลการทดลองที่ 2

ในส่วนนี้เป็นการแสดงผลการทดลองจากการทดลองหาค่าช่วงคาบเวลาที่เหมาะสมในการจัดสรรแบนด์วิดธ์ของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ

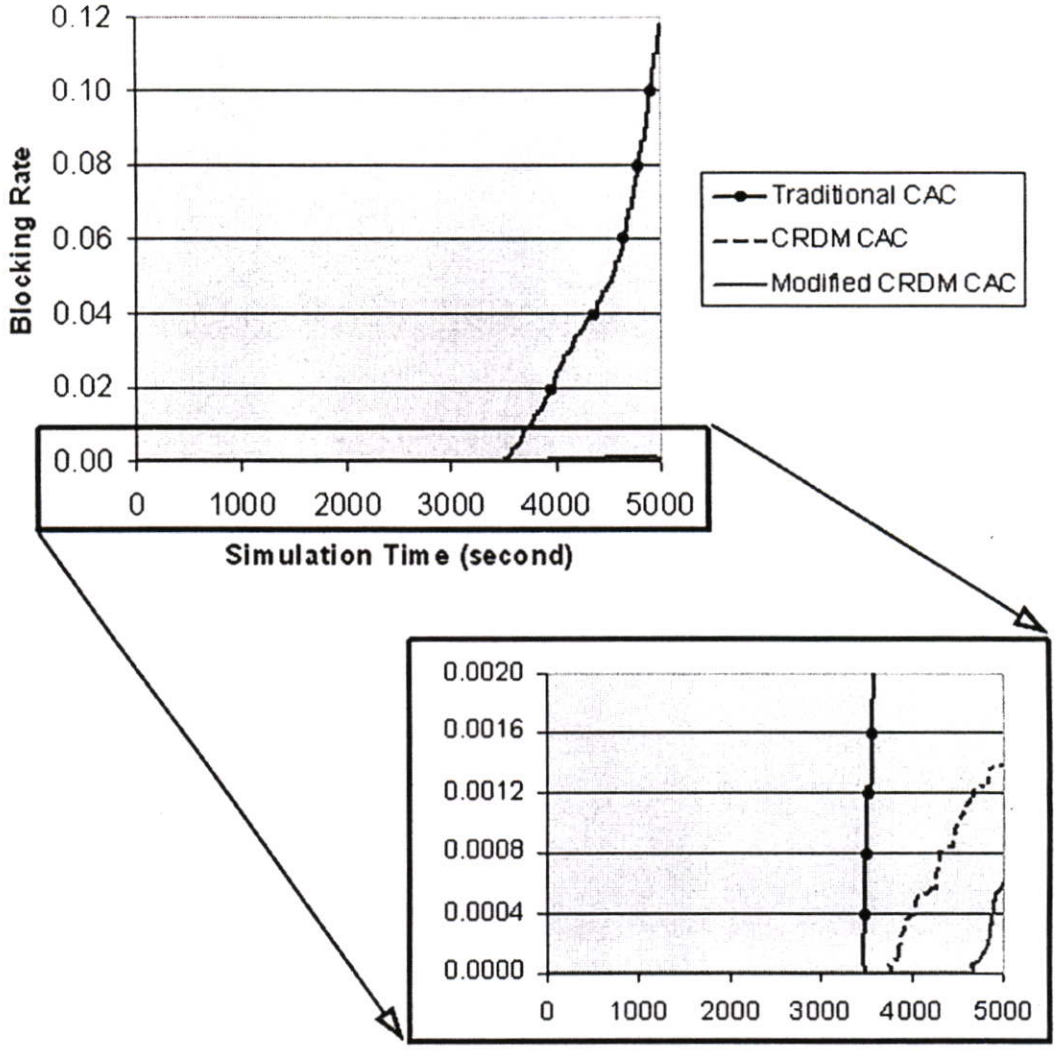
ตารางที่ 4.2 การเปรียบเทียบผลการทดลองหาช่วงคาบเวลาที่เหมาะสมในการจัดสรรแบนด์วิดธ์

Period Time (sec) (Lifetime 240 sec)	Blocking Rate	Average Setup Time (sec)	CAC Signaling Traffic (Bytes)	Average Response Time (sec)
240	0.011524716	4.860347515	53127168	1.85401E-05
200	0.010054293	4.57568353	52984832	1.85401E-05
100	0.006195438	3.753727242	52917248	1.85401E-05
50	0.002935854	2.867760786	52942848	1.85401E-05
25	0.00229709	2.523446641	53152768	1.85401E-05
20	0.002573589	2.149222968	53075968	1.85401E-05
15	0.000884671	2.130281092	53174272	1.85401E-05
14	0.000764034	1.802639341	53045248	1.85401E-05
13	0.000684518	1.645708266	52957184	1.85401E-05
12	0.000884671	1.788370801	53063680	1.85401E-05
11	0.00120637	1.634048628	53014528	1.85401E-05
10	**0.00060367**	**1.793002075**	**53112832**	**1.85401E-05**
9	0.000764034	1.689909775	53064704	1.85401E-05
8	0.000926112	1.844576134	53262336	1.85401E-05
7	0.000684518	1.723973146	53189632	1.85401E-05
6	0.000884671	1.840571756	53414912	1.85401E-05
5	0.001005308	1.710762779	53335040	1.85401E-05
1	0.003058227	3.318515097	55015424	1.85414E-05
0.5	0.003740047	3.740515684	55499776	1.85403E-05



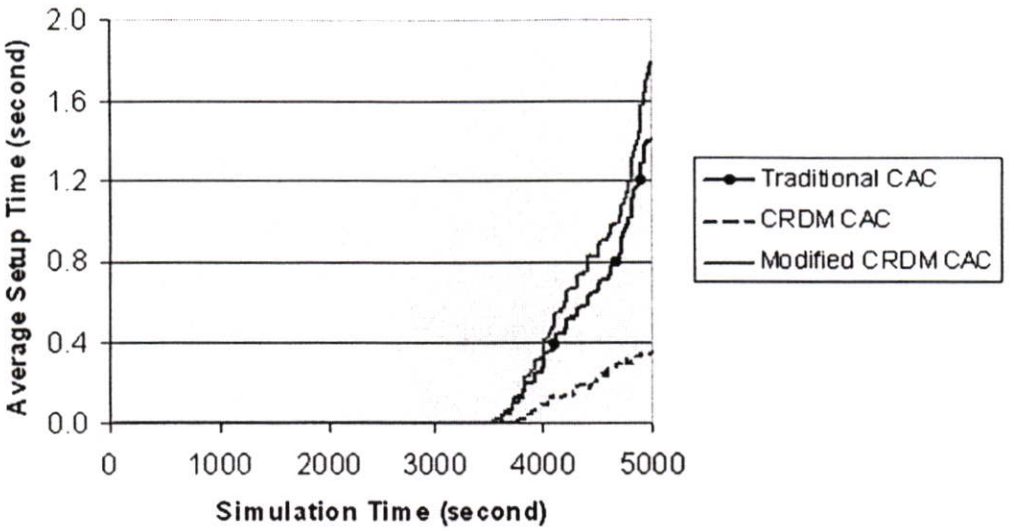
รูปที่ 4.8 กราฟแนวโน้มของการหาช่วงคาบเวลาที่เหมาะสมในการจัดสรรแบนด์วิดธ์

จากตารางที่ 4.2 และรูปที่ 4.8 ได้ทำการทดลองโดยทำการเปลี่ยนค่าคาบเวลาไปเรื่อย ๆ เพื่อหาค่าที่เหมาะสมที่สุด โดยเริ่มจากค่าคาบเวลาเท่ากับ 240 วินาที ซึ่งมีค่าเท่ากับเวลาในการรอคอยของแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อ แล้วจะทำการวัดค่าตามพารามิเตอร์ในหัวข้อที่ 4.1.1 และทำการเปลี่ยนค่าคาบเวลาไปเรื่อย ๆ ตั้งแต่ 200 วินาที 100 วินาที 50 วินาที 25 วินาที 20 วินาที 15 วินาที 10 วินาที 5 วินาที 1 วินาที และ 0.5 วินาที ต่อจากนั้นทำการพิจารณาผลจากการวัดค่าพารามิเตอร์จะพบว่าค่าคาบเวลาที่เหมาะสมจะอยู่ในช่วงตั้งแต่ 5 – 15 วินาที ที่จะทำให้ค่าอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อและค่าเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อที่ต่ำที่สุด ต่อมาได้ทำการปรับหาค่าคาบเวลาแบบละเอียดตั้งแต่ 5 – 15 วินาที เมื่อพิจารณาผลที่ได้จากการทดลอง ค่าคาบเวลาเท่ากับ 10 วินาที จะให้ค่าอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อที่ต่ำที่สุดเท่ากับ 0.00060367 และค่าเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อเท่ากับ 1.79300207 วินาที ซึ่งใกล้เคียงกับค่าที่ต่ำที่สุด จึงเลือกค่าคาบเวลาเท่ากับ 10 วินาที เป็นค่าคาบเวลาที่เหมาะสมที่สุด ซึ่งจะนำผลการทดลองจากการหาค่าคาบเวลาที่ได้ไปเปรียบเทียบกับวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานและวิธีการจัดตั้งการเชื่อมต่อที่มีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง

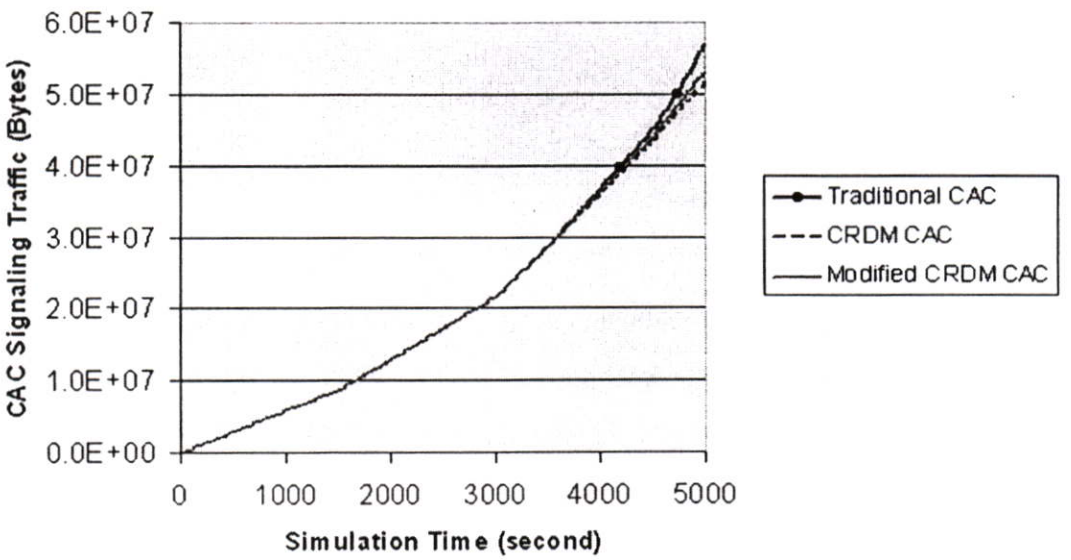


รูปที่ 4.9 กราฟอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อ

ผลการทดลองในรูปที่ 4.9 เป็นการเปรียบเทียบอัตราการปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานกับอัตราการปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์ทั้งสองแบบ จากผลการทดลองพบว่าวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธสามารถลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อลงจากวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานได้ เนื่องมาจากวิธีที่นำเสนอจะเป็นการเพิ่มโอกาสให้แก่แพ็กเกตร้องขอจัดตั้งการเชื่อมต่อที่จะได้รับการจัดสรรแบนด์วิดธ์และจากการจัดสรรแบนด์วิดธ์เป็นช่วงคาบเวลาทำให้สามารถรวบรวมแบนด์วิดธ์ได้มากจึงทำให้อัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อลดลงได้จากวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานประมาณ 99.49 %



รูปที่ 4.10 กราฟเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อ

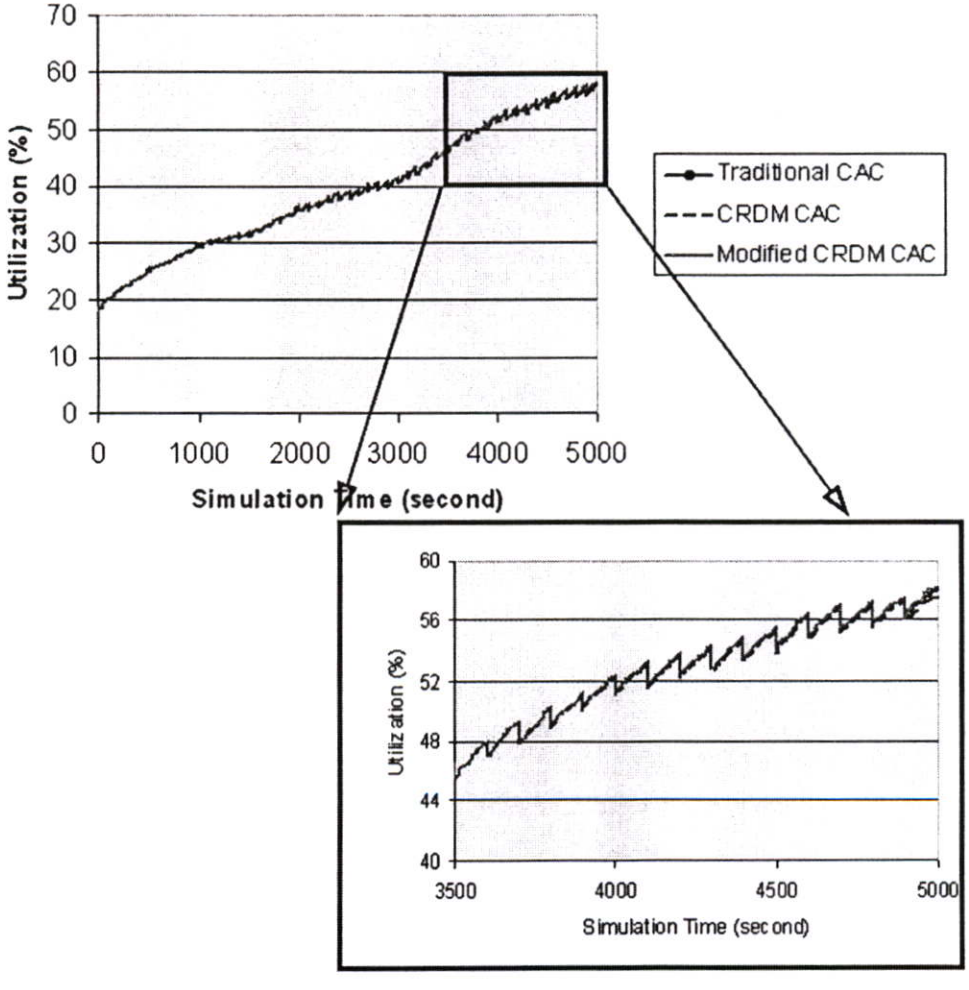


รูปที่ 4.11 กราฟปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อ

ผลการทดลองในรูปที่ 4.10 เป็นการเปรียบเทียบเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อของวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานกับแบบวิธีหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์ทั้งสองแบบ จากผลการทดลองวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์-วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่างจะใช้เวลาเฉลี่ยในการรอเพื่อจัดตั้งการเชื่อมต่อน้อยกว่าวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐาน และวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลามีเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อมากที่สุด ซึ่งเมื่อเปรียบเทียบกับวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานจะมีค่าเพิ่มขึ้นประมาณ 27.357 % เนื่องจาก

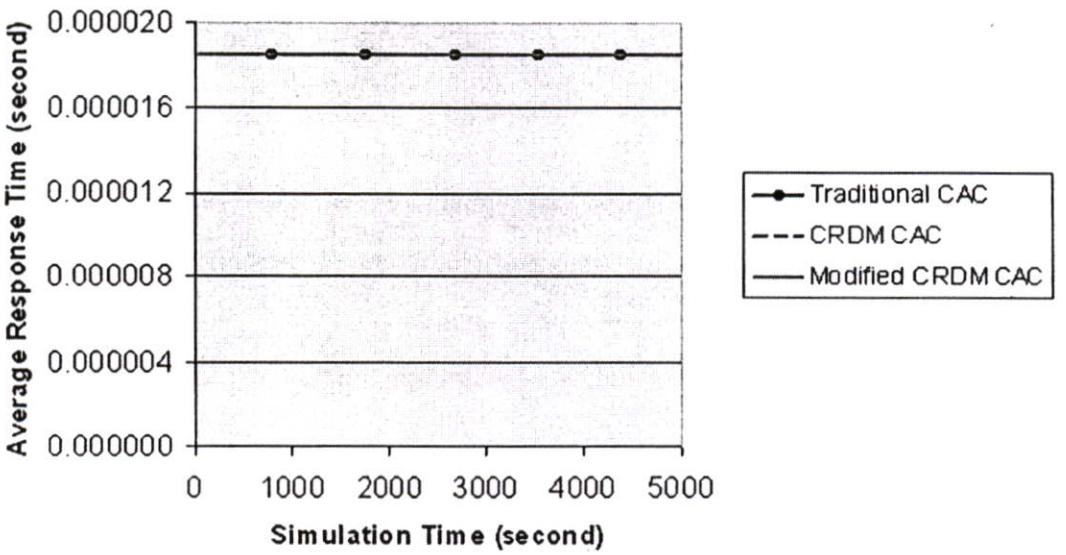
วิธีนี้ต้องให้แพ็กเกตร้องขอรอจนถึงเวลาที่จะทำการจัดสรรแบนด์วิดธ์ จึงทำให้แพ็กเกตร้องขอเกิดเวลาสะสมขึ้น เพราะฉะนั้นเวลาเฉลี่ยให้การจัดตั้งการเชื่อมต่อจึงมากกว่าวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐาน

ผลการทดลองในรูปที่ 4.11 เป็นการเปรียบเทียบปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อของวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานกับวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์ทั้งสองแบบ จากผลการทดลองพบว่าวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธสามารถลดทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อได้ แต่วิธีการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลาจะมีปริมาณทราฟฟิกที่เกิดจากสัญญาณการร้องขอจัดตั้งการเชื่อมต่อมากกว่าวิธีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง เนื่องจากการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลาสามารถลดอัตราปฏิเสธการร้องขอได้มากกว่า จึงทำให้มีการใช้ทราฟฟิกเพิ่มขึ้นจากวิธีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่างเล็กน้อย ซึ่งวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์เป็นช่วงคาบเวลาจะใช้ทราฟฟิกลดลงประมาณ 6.554 % เมื่อเปรียบเทียบกับวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐาน



รูปที่ 4.12 กราฟประสิทธิภาพการใช้งานแบนด์วิดธ์

ผลการทดลองในรูปที่ 4.12 เป็นการเปรียบเทียบประสิทธิภาพการใช้งานแบนด์วิดธ์ของวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานกับวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์ทั้งสองแบบ จากผลการทดลองพบว่าทั้งสามวิธีมีประสิทธิภาพการใช้งานแบนด์วิดธ์ไม่แตกต่างกัน เนื่องจากในวิธีการจัดตั้งการเชื่อมต่อวิธีพื้นฐานเมื่อผู้ใช้งานถูกปฏิเสธการร้องขอไปแล้วก็จะพยายามสร้างและส่งแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อมาเรื่อย ๆ เมื่อเครือข่ายมีแบนด์วิดธ์ว่าง ก็จะถูกใช้งานในทันที ส่วนวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธ แพ็กเกตร้องขอจัดตั้งการเชื่อมต่อจะรออยู่ในคิวรอการเชื่อมต่ออยู่แล้วเมื่อถึงเวลาจัดสรรแบนด์วิดธ์ก็จะสามารถทำการจัดสรรแบนด์วิดธ์ให้ผู้ใช้งานทำการเชื่อมต่อได้ทันที



รูปที่ 4.13 กราฟเวลาในการตอบสนองเฉลี่ย

ผลการทดลองในรูปที่ 4.13 เป็นการเปรียบเทียบเวลาเฉลี่ยในการตอบสนองจากเครือข่ายของวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐานกับวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์ทั้งสองแบบ จากผลการทดลองแสดงให้เห็นว่าวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธใช้เวลาในการตอบกลับแพ็กเกตเท่ากับวิธีการจัดตั้งการเชื่อมต่อแบบวิธีพื้นฐาน เนื่องจากวิธีที่นำเสนอเมื่อนำมาใช้งานร่วมกับเครือข่ายจำลองแล้วจะไม่ทำให้มีการประมวลผลช้าลง

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

• สรุปผลการวิจัย

วิทยานิพนธ์นี้ได้นำเสนอวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธบนเครือข่ายที่มีการรับประกันคุณภาพในการให้บริการ IntServ over DiffServ Architecture โดยวิทยานิพนธ์นี้ได้แยกสภาวะการทำงานของวิธีการจัดตั้งการเชื่อมต่อเป็นสองสภาวะคือสภาวะปกติที่สามารถรับการเชื่อมต่อเพิ่มได้และสภาวะอิ่มตัวที่ไม่สามารถรับการเชื่อมต่อเพิ่มได้ และได้คิดวิธีใหม่ขึ้นมาคือวิธีการจัดตั้งการเชื่อมต่อแบบการหน่วงเวลาการปฏิเสธเพื่อมาใช้กับเครือข่ายที่อยู่ในสภาวะอิ่มตัว โดยในการหน่วงเวลาแพ็กเกตร้องขอจัดตั้งการเชื่อมต่อจะทำการหน่วงเวลาไว้ในคิวรอการเชื่อมต่อที่มีการทำงานแบบมาถึงก่อนได้รับการบริการก่อนเพื่อให้เกิดความเสมอภาคแก่ผู้ร้องขอ นอกเหนือจากนั้น ได้คิดวิธีการจัดสรรแบนด์วิดธ์ให้แก่แพ็กเกตที่ถูกหน่วงเวลาไว้สองแบบคือ การจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง และการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา ต่อมาได้ทำการพัฒนาวิธีการที่นำเสนอลงบนโปรแกรมจำลองระบบเครือข่าย OPNET เพื่อทำการทดลองหาค่าประสิทธิภาพที่มีเราเตอร์หลัก 15 โหนด เราเตอร์ขอบ 150 โหนด และกลุ่มผู้ใช้งาน 150 โหนด

จากผลการทดลองแสดงให้เห็นว่าวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่นำเสนอนี้สามารถลดอัตราปฏิเสธการร้องขอจัดตั้งการเชื่อมต่อจากผู้ใช้งานได้ 98.8 % จากวิธีพื้นฐาน ผลที่ตามมาก็คือ เมื่อไม่ต้องมีการสร้างและส่งแพ็กเกตร้องขอมายังเครือข่ายซ้ำ ๆ ทำให้สามารถลดจำนวนทราฟฟิกที่เกิดสัญญาณการร้องขอจัดตั้งการเชื่อมต่อได้ 9.15 % โดยที่ไม่มีผลกระทบต่อประสิทธิภาพการใช้งานแบนด์วิดธ์และเวลาในการตอบสนองของเครือข่ายหรือการประมวลผล นอกจากนี้วิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่างยังสามารถลดเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อได้ 75.6 % เพราะเมื่อเครือข่ายมีแบนด์วิดธ์ว่างก็จะสามารถส่งแพ็กเกตไปเตือนให้ผู้ใช้งานทำการส่งข้อมูลได้ทันที และยังได้ปรับปรุงส่วนวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธให้มีการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลา ซึ่งจะมีเวลาเฉลี่ยในการจัดตั้งการเชื่อมต่อที่มากกว่าวิธีพื้นฐานเนื่องมาจากการรอคอยที่จะจัดสรรแบนด์วิดธ์เป็นช่วงคาบเวลาทำให้เกิดเวลาสะสมในการรอคอย แต่ในวิธีนี้สามารถรวบรวมแบนด์วิดธ์ให้เป็นกลุ่มก้อนได้จึงทำให้วิธีนี้มีอัตราปฏิเสธการร้องขอที่ต่ำกว่าวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบทันทีที่มีแบนด์วิดธ์ว่าง

56.232 % ซึ่งในวิธีนี้จะเป็นการปรับปรุงความเสมอภาคในการให้บริการแก่การร้องขอที่ต้องการแบนด์วิดธ์มากด้วย

- **แนวทางในการวิจัยและพัฒนา**

แนวทางในการพัฒนาต่อจากวิทยานิพนธ์นี้คือ การปรับค่าคาบเวลาในการจัดสรรแบนด์วิดธ์ให้มีการเปลี่ยนแปลงได้ตามสภาพแวดล้อม เช่น ปริมาณของผู้ใช้งาน จำนวนแบนด์วิดธ์ที่สายส่งข้อมูลสามารถรองรับได้ เป็นต้น ซึ่งผลที่ได้คาดว่าจะทำให้สามารถลดเวลาสะสมเฉลี่ยในการจัดตั้งการเชื่อมต่อของวิธีการจัดตั้งการเชื่อมต่อแบบหน่วงเวลาการปฏิเสธที่มีการจัดสรรแบนด์วิดธ์แบบเป็นช่วงคาบเวลาได้

เอกสารอ้างอิง

[1] Braden R., Clark D., Shenker S., "Integrated Services in the Internet Architecture: An Overview," IETF RFC 1633, June 1994.

[2] Braden R., Zhang L., Berson S., Herzog S., Jamin S., "Resource Reservation Protocol (RSVP) – Version 1, Functional Specification," IETF RFC 2205, September 1997.

[3] Wroclawski J., "The use of RSVP with IETF integrated Services," IETF RFC 2210, 1997.

[4] Shenker S., Partridge C., Guerin R., "Specification of Guaranteed Quality of Service," IETF RFC 2212, September 1997.

[5] Wroclawski J., "Specification of the Controlled-Load Network Element Service," IETF RFC 2211, September 1997.

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

[7] Baker F., Iturralde C., Faucher Le F., Davie B., "Aggregation of RSVP for IPv4 and IPv6 Reservation," Internet Draft, draft-ietf-issll-rsvp-aggr-02.txt, March 2000.

[8] Nichols K., Blake S., Baker F., Black D., "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," IETF RFC 2474, December 1998.

[9] Jacobson V., Nichols K., Poduri K., "An Expedited Forwarding PHB," IETF RFC 2598, 1999.

[10] Heinanen J., Baker F., Weiss W., Wroclawski J., "Assured Forwarding PHB group," IETF RFC 2597, 1999.

[11] Bernet Y., Ford P., Yavatkar R., Baker F., Zhang L., Speer M., etc., "A Framework for Integrated Services Operation over DiffServ Networks," IETF RFC 2998, Nov. 2000.

[12] Reichmeyer F., Ong L., Terzis A., Zhang L., Yavatkar R., "A Two-tier Resource Management Model for Differentiated Services Networks," IETF Internet Draft, Nov. 1998.

[13] Borgonovo F., Capone A., Fratta L., Marchese M., Petrioli C., "End-to-End QoS Provisioning Mechanism for Differentiated Services," Internet Draft, July 1998.

[14] Borgonovo F., Capone A., Fratta L., Petrioli C., "VBR Bandwidth Guaranteed Services over DiffServ Networks," IEEE RTAS Workshop, June 1999.

[15] Mei Yang, Yan Huang, Jaime Kim, Meejeong Lee, Tatsuya Suda, Mastubara Daisuke., "An End-to-End QoS Framework with On-Demand Bandwidth Reconfiguration," InfoCom 2004, Hongkong, China.

[16] Apostolopoulos G., Kamat S., Guerin R., Tripathi SK., "Quality of Service Based Routing: A Performance Perspective," SIGCOMM, 1998.

[17] Zhang Z., et al., "Decoupling QoS Control from Core Routers: a Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services," In Proceedings of ACM SIGCOMM'00, Stockholm, Sweden, August 2000.

[18] Stica I., Zhang H., "Providing Guaranteed Services without per Flow Management," In Proceedings of ACM SIGCOMM'99, Cambridge, MA, August 1999.

[19] Centinkaya C., Knightly E., "Scalable Services via Egress Admission Control," In Proceedings of IEEE INFOCOM'00, Tel Aviv, Israel, March 2000.

[20] Qiu J., Knightly E., "Inter-class Resource Sharing using Statistical Service Envelopes," In Proceedings of IEEE INFOCOM'99, New York, NY, March 1999.

[21] Elek V., et al., "Admission Control Based on End-to-End Measurements," In Proceedings of IEEE INFOCOM'00, Tel Aviv, Israel, March 2000.

[22] Breslau L., et al., "End-point Admission Control: Architectural Issues and Performance," In Proceedings of ACM SIGCOMM'00, Stockholm, Sweden, August 2000.

[23] Lin H., et al., "VoIP Signaling Performance Requirements and Expectations," IETF Internet Draft, June 1999.

[24] Pual T. Brady, "A Model for Generating On-Off Speech Patterns in Two-Way Conversion," The Bell System Technical Journal, September 1969, pp. 2445-2471.

[25] OPNET Technologies Inc., "OPNET Modeler Modeling Manual," Bethesda, MD, 2001.

[26] OPNET Technologies Inc., "OPNET Protocol Model Documentation," Bethesda, MD, 2001.

[27] K. Ongpaibool, M. Lertwatechakul, and P. Thumwarin, "A Call Admission Control with Call Request Delaying Method in IP Networks," The 5th Asia Pacific International Symposium on Information Technology (APIS5), pp. 227-230, Hangzhou China, Jan 9-10, 2006.

ภาคผนวก

ภาคผนวก ก.

สมการของกระบวนการสุ่มที่ใช้ในโปรแกรมจำลองระบบเครือข่าย

1. **Uniform Integer PMF**

   (uniform_int, a, b)

$$p_x(x_0) = 1/(1+b-a) \qquad x_0 = 0,\pm1,\pm2,\ldots \quad a \le x_0 \le b$$

$$a = 0,\pm1,\pm2,\ldots$$
$$b = 0,\pm1,\pm2,\ldots \qquad a \le b$$
$$E(x) = (a+b)/2$$
$$\sigma_x^2 = \frac{1}{12}[(b-a)(2+b-a)]$$

Arg #0: min_outcome = a
Arg #1: max_outcome = b

2. **Exponential PDF**

   (exponential, a, b)

$$f_x(x_0) = \begin{cases} ae^{-ax_0} \\ 0 \end{cases} \qquad x_0 > 0 \quad otherwise$$

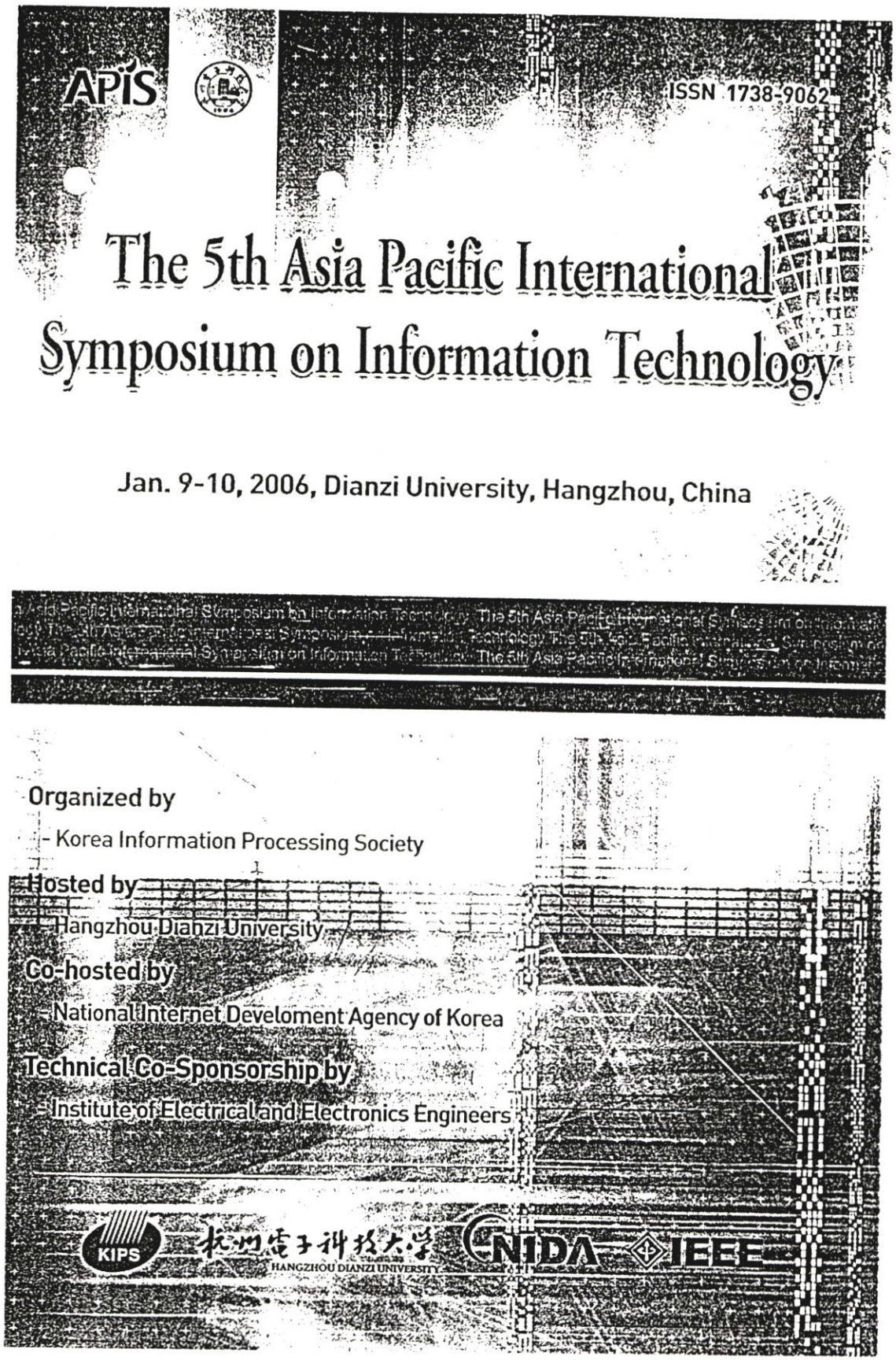
$$a > 0$$
$$E(x) = a^{-1}$$
$$\sigma_x^2 = a^{-2}$$

Arg #0: mean_outcome = 1/a
Arg #1: <ignored>

ภาคผนวก ข.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. **K. Ongpaibool**, M. Lertwatechakul and P. Thumwarin, "**A Call Admission Control with Call Request Delaying Method in IP Networks**", The 5th Asia Pacific International Symposium on Information Technology (APIS5), pp. 227-230, Hangzhou, China, January 9-10, 2006.


APIS
ISSN 1738-9062
The 5th Asia Pacific International Symposium on Information Technology
Jan. 9-10, 2006, Dianzi University, Hangzhou, China
Organized by
- Korea Information Processing Society
Hosted by
- Hangzhou Dianzi University
Co-hosted by
- National Internet Develoment Agency of Korea
Technical Co-Sponsorship by
- Institute of Electrical and Electronics Engineers
KIPS
HANGZHOU DIANZI UNIVERSITY
NIDA
IEEE

The 5th Asia Pacific International Symposium on Information Technology, Hangzhou, China

A Call Admission Control with Call Request Delaying Method in IP Networks

K. Ongpaibool, M. Lertwatechakul, P. Thumwarin

Department of Information Engineering, Faculty of Engineering,

King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand.

ultraplus78@hotmail.com klmayure@kmitl.ac.th ktpitak@kmitl.ac.th

ABSTRACT

In this paper, we propose a new theme of call admission control mode called "Call Request Delaying Method" (CRDM) that could improve call blocking rate and utilization in QoS network, especially in saturated situation. The CRDM was designed to handle new incoming call requests instead of reject them when the request bandwidth is still not available. The call request holding time for a new released bandwidth could be estimated by traffic descriptor of present active connections. By applying this delaying and prediction technique, we can show the improvement of call blocking rate and call attempt waiting time through simulation results compared to the traditional call admission control (CAC) model on IntServ over DiffServ framework.

Keywords : Admission Control, QoS, DiffServ, IntServ

1. Introduction

The future trend of communication network is to integrate many services onto the single IP network. As we know, the traditional IP networks support only best-effort services, it is not possible to differentiate among them nor to assure specific QoS targets for a given service. But the need of QoS support in IP network has driven in many research works and also standard body to provide the QoS onto IP network.

One of the main elements required in the network to provide QoS is the CAC mechanism. If the network has no control on the number of active flows and their consuming bandwidth, the overall traffic demand may exceed the total provided and the QoS of every flow (e.g. the transmission delay, lost packets ratio) could be degraded With CAC support, once a new call request attempt to establish a connection, CAC would accept the new call request only if the available resources are sufficient or else just reject it.

Call request rejection is ordinarily raised immediately after CAC module find out that the requested bandwidth is not available. This induce more requests and tedious work for user. Moreover, in the saturate situation, the traditional CAC process may not be fair because a new incoming call request may be granted just after a previous call request was rejected. Consequently, besides of the QoS concerns, the awareness of network services access opportunity should be considered too.

If we consider on today applications and the future trend, the on-demand stored media applications and defined time period applications such as video on-demand, e-lecturing, Internet channel program is going to quickly consume the Internet bandwidth. By utilizing the traffic descriptor on allocating bandwidth and duration these applications, we could propose a new scheme of CAC process that could reduce call blocking ratio, amount of call request signaling message and enhance more fairness of network access in saturate network: the CRDM. The CAC scheme handles incoming call requests in FIFO queue and be waiting until the requested bandwidth is available or their timeout is reached. This scheme could reduce call blocking rate and help enhance the fairness of CAC process.

As to evaluate the proposed idea we applied the CRDM to Mei Yang's framework [11]. We have developed the CAC scheme onto OPNET model and the simulation results show the improvement in average call request retrying times significantly compared to the unmodified scheme. In this

paper, the content was organized as follows, section 2 mention on a brief of existing admission control methods in [11] framework. The basic principles of our approach in detail was described in section 3. In Section 4, simulated results and analyses are provided, and finally the conclusion.

2. Call Admission Control Methods in IntServ over DiffServ

IETF has attempted to provide QoS on top of the current Internet and two approaches are introduced: IntServ and DiffServ. IntServ promotes end-to-end QoS guarantees by setting up an end-to-end connection for each flow and maintaining the states of all connections, whereas DiffServ promotes scalability by pushing flow classification to network boundary and offering QoS to service classes. Targeting at both end-to-end QoS guarantees and scalability, RFC 2998 [8] suggests a combination of the two architectures, an IntServ over DiffServ framework as to combine a good point of two architectures.

The CAC mechanism for Integrated Services (IntServ) [1] architectures was specified by IETF in the means of a signaling protocol called RSVP (Resource Reservation Protocol). [2] Using RSVP, IntServ faces to scalability problem since all routers need to maintain state information of every flow. As to solve this scalability problem DiffServ architecture was proposed [3]. In this architecture, flows are aggregated in classes according to their specific characteristics and to be treated differently according to their classes. Since DiffServ has no admission control and traffic policing mechanisms, therefore there is no certain QoS guarantee in DiffServ network.

As trying to take advantage of both two IETF architectures and reduce the scalability problem, several novel architectures and algorithms have been proposed: [4], [5], [6], [7]. The proposed framework of [11] adopts overall best ideas of previous works, it assume that two types of routers, edge router and core routers are exist in a network. Function of edge router is making admission decision, mapping individual flows to different service classes and transmit packets to the network for the network objects (i.e., clients). Core routers are DiffServ routers that provide class-based service differentiation. In this framework, to provide class-based service differentiation and local admission control at network edge without hop-by-hop signaling, link bandwidth is organized in hierarchy. First, the physical link is statically divided into multiple Provisioned Links (PLs); and a PL is dedicated to only one traffic class. Each PL is further divided into multiple trunks; one trunk is assigned to a given traffic class of one incoming edge router irrespective to their destination. As to renegotiate trunks' bandwidth, source edge routers have to keep track of available bandwidth of their assigned trunks and performs admission control locally without hop-by-hop signaling. Figure 1 illustrates this hierarchical bandwidth organization.

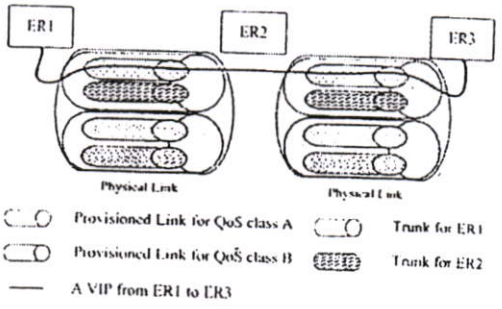


Figure 1 : Hierarchical bandwidth organizations.

3. Proposed Method

The proposed method is implemented by dividing the call admission control operation into two modes: normal mode and saturated mode. A network will be considered in saturated situation when there is not enough bandwidth for a new call request. Saturate network can not admit a new call request until some bandwidth would be released from the terminated connections. The traditional CAC responds to this situation by immediately performing call request rejection to deny new call request when the available resources is not sufficient. In order to access the network, user may try to send call requests iteratively and wait until the request would be granted. The call request iteration induces more signaling message and increase the connection establishing time. As to reduce the call request waiting time and call rejection, we proposed CRDM to delay call requests in FIFO queue and notifies the user later when the delayed request is admitted.

In order to make an admission decision, the CRDM determines the exist connections' traffic descriptors and their state information to examine an amount of bandwidth will being released in the near future. And to prevent excessive waiting time, we set a threshold on the wait time. If a request's lifetime is expire, the request will be dropped from the queuing buffer and CRDM will send a rejection packet to the user.

Figure 2.a shows the operation of the traditional CAC at edge router on the testbed framework. The CAC respond to a call request by sending an admit message or a reject message at once the resources determination is finished. In normal mode, the CRDM send admit message as same as the traditional CAC operation, but in saturate mode, CRDM operation is different as shown in figure 2.b – 2.d. CRDM immediately responds for every incoming call request by sending an acknowledge, this is to let the user know and

make their decision on the admission postponement. User may cancel the delayed request or just hold on a few seconds for the admission response. In case of accepting the admission postponement, CRDM would keep the call request for future resource reservation. In contrast, if user deny to postpone the request by sending cancel message, CRDM will delete the request information and user have to initiate a new call request again if he or she still need to access the network.

Figure 2.b and 2.c demonstrates success admissions for two delayed requests. Figure 2.b shows the shorter delay compare to figure 2.c, this is because the network got sufficient resources back from the terminated connection for just after the request in figure 2.b is received. In case that the network is still busy and some delayed requests lifetime are expire, the CRDM will send a reject message and clear the queued requests, as described in figure 2.d.

Other than, expecting CRDM to resolve call request iteration problems, we also utilized queue to improve the fairness of CAC process too. Even if the CRDM request queue was mentioned as FIFO but the CRDM operation detail in saturate mode is not truly FIFO. CRDM always admit the first queued request iff the current available resources are sufficient to provide the requested QoS level. In contrast, if the current resources is not sufficient to the first queued request, the second and the following requests will be determined. The first admissible request found by in sequentially scanning the delayed request queue, would be granted. This means, in some case the following request might be granted before of the head of queued request. In order to prevent call requests of moderate requirement application to be frequently rejected, CRDM was programmed to make an admission decision periodically. The admission period give more chance for the network to collect the released resources before determine the next admissible request.

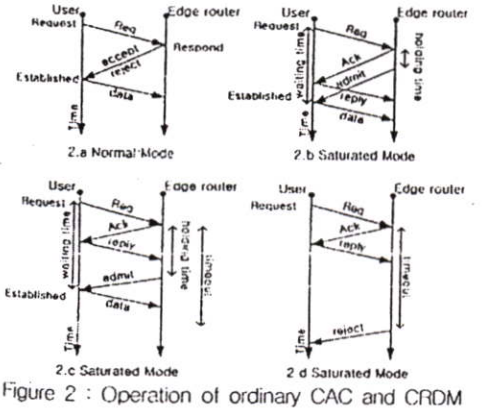


Figure 2 : Operation of ordinary CAC and CRDM

4. Simulation Results

This section show the simulation results of the traditional CAC over the reference framework [11] compared to the CAC modified by applying the CRDM. Since we concern on the services of predictable traffic then two traffic classes; video and audio are used as source traffics. Table 1 shows traffic characteristics of these two types of traffic. Descriptors of video and audio traffic are obtained from MPEG-4, VoIP [9][10] ITUT G.723.1 standard header correspondingly.

Table 1 : Traffic parameter values

Traffic Class	Bandwidth Requirement	Session Duration	Example
Audio	64 Kbps (CBR)	Exponential dist. with 300 seconds	VoIP ITUT G723.1
Video	1-6 Mbps	Exponential dist. with 1152 seconds	Video MPEG-4

The network topology in our simulation is shown in Figure 3. As in the network, there are 15 core routers, and 10 edge routers are connected to each core router that support traffic from LANs. OC-48 fiber links are used as physical links between core routers and OC-12 fibers links were used to connect among edge routers and core routers. Therefore, there are 150 edge routers and 150 LANs in our simulation model. In the simulation, we defined the parameters as follows : packet length was fixed to 1024 bytes, the average inter-arrival time of call attempts was 50 seconds with exponential distribution. The simulation run time was set to 5000 seconds.

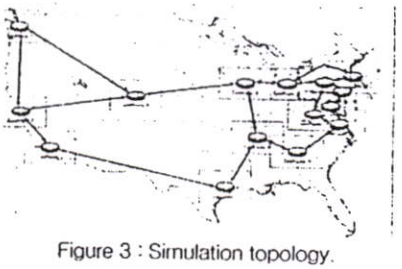


Figure 3 : Simulation topology.

Using the simulation model described above, the performance of the proposed method is evaluated and compared with the traditional admission control. The following three performance metrics are used in comparison:

- *Blocking Rate:* This is the ratio of the number of blocked user flows to the number of all user flows.
- *Average Waiting Time:* The average amount of time that an user attempt to access the network was requested until it was granted.

- *CAC Signaling Traffic:* A total number of CAC signaling messages counted in the simulation including all messages for call request, acknowledge, accept, cancel, reject, admit.

Figure 4 shows the blocking rate of traditional admission control and CRDM. It was shown that the proposed method could reduce call blocking rate significantly because we this method in saturated mode that was reduce rejection rate of the new incoming call request.

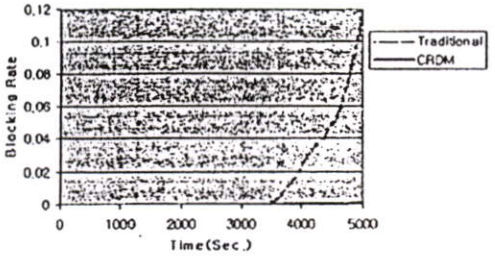


Figure 4 : Comparison on Call Blocking Rate

Figure 5 shows average waiting time of each call request. It was shown in this figure that the proposed method has extremely lower average waiting time for a call attempt. This is because waiting for the latest released bandwidth is quite shorter than making a new (or several) call request. As the result of decreasing call blocking rate, the total CAC signaling traffic are also reduced as shown in Figure 6.

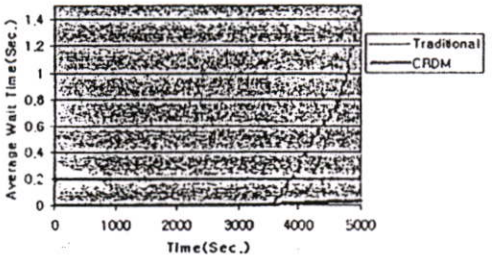


Figure 5 : Comparison on Average Waiting Time

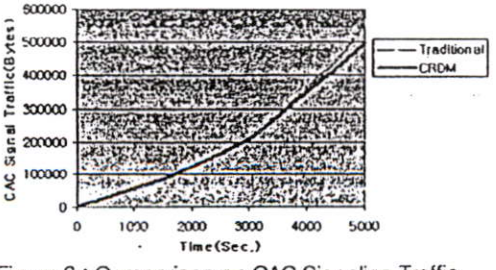


Figure 6 : Comparison on CAC Signaling Traffic

5. Conclusions

In this paper, we propose CRDM as a new CAC operation mode to solve the CAC problem in saturate network. The objective of CRDM is to reduce call attempt waiting time, call blocking rate and signaling traffic caused by call rejection and improve the CAC fairness in saturated situation. By extra delaying call requests and utilizing traffic descriptors, the CRDM could help improve the traditional admission control as show in simulation results. As result, this method offers user more ease and convenient access scheme to take place of the call request iteration and reduce signaling traffic when the network is busy.

References

[1] R. Braden, et al., "Integrated Services in the Internet Architecture", Internet RFC 1633, 1944.

[2] L. Zhang, et al., "RSVP: A New Resource Reservation Protocol", *IEEE Network*, Vol. 7, pp. 8-18, September 1993.

[3] S. Blake, et al., "An Architecture for Differentiated Services", Internet RFC 1475, 1998.

[4] I. Stoica and H. Zhang, "Providing Guaranteed Services without per Flow Management", *In Proceedings of ACM SIGCOMM'99*, Cambridge, MA, August 1999.

[5] F. Baker, C. Iturralde, F. Le Faucher and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", Internet Draft, draft-ietf-issll-rsvp-aggr-02.txt, March 2000.

[6] C. Centinkaya and E. Knightly, "Scalable Services via Egress Admission Control", *In Proceedings of IEEE INFOCOM'00*, Tel Aviv, Israel, March 2000.

[7] J. Qiu and E. Knightly, "Inter-class Resource Sharing using Statistical Service Envelopes", *In Proceedings of IEEE INFOCOM'99*, New York, NY, March 1999.

[8] Y. Bamet, et al., "A Framework for Integrated Services Operation over DiffServ Networks", RFC 2998, November 2000.

[9] H. Lin, et al., "VoIP Signaling Performance Requirements and Expectations", IETF Internet Draft, June 1999.

[10] Paul T. Brady, "A Model for Generating On-Off Speech Patterns in Two-Way conversation", The Bell System Technical Journal, pp. 2445-2471, September 1969.

[11] Mei Yang, et al., "An End-to-End QoS Framework with On-Demand Bandwidth Reconfiguration", *InfoCom 2004*, HongKong, China.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายกิตติศักดิ์ องค์ไพบูลย์
วัน เดือน ปีเกิด	26 กุมภาพันธ์ 2525
สถานที่เกิด	สุราษฎร์ธานี
ที่อยู่ปัจจุบัน	254/10 หมู่บ้านประชาชื่นแมนชัน ซอยเจิมบุญ ถนนริมคลองประปา แขวงบางซื่อ เขตบางซื่อ กรุงเทพฯ 10800 โทร. 0-2959-3788
ประวัติการศึกษา	2546 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี 2549 วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง