อัลกอริธึมการสร้างเส้นทางแบบคำนึงถึงคุณภาพการให้บริการที่มีความสามารถในการแก้ไขข้อบกพร่องในการเลือกเส้นทางได้อย่างรวดเร็ว

EARLY CRANKBACK QOS-BASED ROUTING ALGORITHM

สมพร กันทะวงศ์
SOMPHONE KANTHAVONG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2546

ISBN 974-324-858-7

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อัลกอริธึมการสร้างเส้นทางแบบคำนึงถึงคุณภาพการให้บริการที่มีความสามารถในการแก้ไขข้อบกพร่องในการเลือกเส้นทางได้อย่างรวดเร็ว

EARLY CRANKBACK QOS-BASED ROUTING ALGORITHM

สมพร กันทะวงศ์

SOMPHONE KANTHAVONG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2546

ISBN 974-324-858-7

เลขหมู่.....
เลขทะเบียน.....48920
วัน, เดือน, ปี 12 ส.ค. 2547

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

EARLY CRANKBACK QOS-BASED ROUTING ALGORITHM

SOMPHONE KANTHAVONG

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

2003

ISBN 974-324-858-7

COPYRIGHT 2003

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ อัลกอริธึมการสร้างเส้นทางแบบคำนึงถึงคุณภาพของการให้บริการที่มีความสามารถในการแก้ไขข้อบกพร่องในการเลือกเส้นทางได้อย่างรวดเร็ว
EARLY CRANKBACK QoS-BASED ROUTING ALGORITHM

ชื่อนักศึกษา นายสมพร กันทะวงศ์

รหัสประจำตัว 44612904

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ประกิจ ตังติสานนท์

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.นภพินทุ์	อนันตรศิริชัย	
รศ.ดร.รัตติกร	วรากูลศิริพันธุ์	
อาจารย์มยุรี	เลิศเวชกุล	
รศ.ดร.ปีติเขต	สู้รักษา	
รศ.ประกิจ	ตังติสานนท์	

วัน/เดือน/ปี ที่สอบ 15 ตุลาคม 2546 เวลา 14.00-16.00 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-402)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่.....15.....**เดือน**.....ตุลาคม.....**พ.ศ**.....2546.....

หัวข้อวิทยานิพนธ์	อัลกอริธึมการสร้างเส้นทางแบบคำนึงถึงคุณภาพการให้บริการที่มีความสามารถในการแก้ไขข้อบกพร่องในการเลือกเส้นทางได้อย่างรวดเร็ว
ชื่อนักศึกษา	นายสมพร กันทะวงศ์
รหัสนักศึกษา	44612904
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรม	วิศวกรรมสารสนเทศ
พ.ศ.	2546
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ประกิจ ตังติสานนท์

บทคัดย่อ

เทคโนโลยีในระบบสื่อสารแพ็คเก็จที่ใช้กันอยู่ในระบบเครือข่ายอินเตอร์เน็ตในปัจจุบันนี้นั้น ยังไม่สามารถที่จะรองรับแอพพลิเคชั่นต่าง ๆ ที่มีความต้องการที่หลากหลายทั้งในแง่ของแบนด์วิดท์, เวลาหน่วง และการรับประกันอัตราการสูญหายของแพ็คเก็จข้อมูลได้ จึงมีความพยายามที่จะพัฒนาเทคโนโลยีในการจัดการและควบคุมช่องสัญญาณสื่อสารแพ็คเก็จให้สามารถที่จะให้บริการตอบสนองต่อความต้องการอันหลากหลายของผู้ใช้ และสามารถที่จะรับประกันคุณภาพในการให้บริการ (QoS Services) แก่ผู้ใช้ได้อีกด้วย การจัดตั้งเส้นทางที่เหมาะสมกับความต้องการของผู้ใช้นั้นเป็นกลไกหลักที่สำคัญในการให้บริการแบบรับประกันคุณภาพของการให้บริการได้

หัวข้อวิจัยนี้จึงขอนำเสนออัลกอริธึมสำหรับจัดตั้งเส้นทางแบบสามารถตอบสนองต่อความต้องการระดับคุณภาพของบริการ สำหรับใช้ในระบบเครือข่ายมัลติโปรโตคอลเลเบิลสวิทช์ (MPLS) ที่สามารถแก้ไขการตัดสินใจเลือกเส้นทางที่ผิดพลาด (crankback ability) ได้ภายในโหนดเราท์เตอร์ระหว่างเส้นทาง (intermediate router) อัลกอริธึมดังกล่าวนอกจากจะสามารถตอบสนองต่อความต้องการของผู้ใช้ตามที่ปรากฏในคำร้องขอเชื่อมต่อช่องสัญญาณ (call set-up request) แล้วยังสามารถที่จะประหยัดเวลาที่ใช้ในการจัดตั้งเส้นทางกรณีที่ระบบเครือข่ายมีภาระโหลดสูงได้อย่างมีประสิทธิภาพ

Thesis Title	Early Crankback QoS Based Routing Algorithm
Student	Mr. Somphone Kanthavong
Student ID.	44612904
Degree	Master of Engineering
Programme	Information Engineering
Year	2003
Thesis Advisor	Assoc. Prof. Prakit Tangtisanon

ABSTRACT

Nowadays, Packet switching technology used in the Internet still can not provide differentiated services for today applications whose requirement are different both for bandwidth and transit delay. Many new traffic engineering and controlling techniques were proposed as to solve the mentioned problem and to guarantee the quality of services (QoS) in the same time. QoS-based routing is an important mechanism that helpfully provide QoS-based services onto a communication network.

This research proposes a new QoS-based routing algorithm that could be used in MPLS (Multi-Protocol Label Switching) network. The algorithm provide early crank-back mechanism that improves the QoS call-setup time in medium to high load situation significantly.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ประสบความสำเร็จลุลวงได้เป็นอย่างดีโดยได้รับการช่วยเหลือและสนับสนุนจากบุคคลหลาย ๆ ฝ่าย ซึ่งผู้เขียนขอขอบพระคุณทุกท่าน ดังต่อไปนี้

คุณพ่อ คุณแม่และครอบครัว ผู้ซึ่งคอยเป็นกำลังใจให้อย่างเสมอมา

รศ. ประกิจ ตังติสานนท์ อาจารย์ผู้ควบคุมวิทยานิพนธ์ ที่กรุณาให้คำปรึกษา ชี้แนะแนวทางและข้อคิดเห็นที่เป็นประโยชน์ต่อการทำวิจัย นี้จนสำเร็จลุล่วงไปด้วยดี

อาจารย์ มยุรี เลิศเวชกุล อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม ที่กรุณาให้คำปรึกษา ชี้แนะแนวทางและข้อคิดเห็นที่เป็นประโยชน์ต่อการทำวิจัย การทำวิทยานิพนธ์ รวมทั้งตรวจทานแก้ไขวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงไปด้วยดี

อาจารย์ทุกท่าน ที่กรุณาให้การอบรมสั่งสอนและให้ความรู้เสมอมา

เพื่อนร่วมงานในห้องปฏิบัติการเครือข่ายคอมพิวเตอร์ (Computer Network Lab) ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบังที่ได้ให้ความช่วยเหลือแนะนำและคำปรึกษาต่าง ๆ ในการทำวิจัยจนสำเร็จลุล่วงด้วยดี

AUN/SEED-Net และ JICA ที่ให้การสนับสนุนทุนการศึกษาเสมอมา ซึ่งต้องขอขอบคุณเป็นอย่างมาก

และทุกท่านที่ไม่ได้กล่าวถึงในที่นี้ ที่ให้ความช่วยเหลือ และกำลังใจในการทำวิทยานิพนธ์ฉบับนี้จนสำเร็จ

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

สมพร กันทะวงศ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของวิทยานิพนธ์ 2

1.3 ทฤษฎีที่เกี่ยวข้อง 2

1.4 วิธีการที่นำเสนอ 3

1.5 ขอบเขตของวิทยานิพนธ์ 3

1.6 ขั้นตอนการทำวิทยานิพนธ์ 3

1.7 โครงสร้างของวิทยานิพนธ์ 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 5

2.1 การออกแบบโปรแกรมสื่อสารชั้นควบคุมเครือข่าย (Network Layer Design Issue) 5

2.1.1 บริการสำหรับชั้นนำส่งข้อมูล (Service Provided to the Transport Layer) 5

2.1.2 การจัดโครงสร้างภายในชั้นควบคุมเครือข่าย 6

2.2 อัลกอริธึมสำหรับเลือกทางเดินข้อมูล (Routing Algorithm) 7

2.2.1 หลักการจำแนกประเภทของการเลือกเส้นทาง 9

2.2.1.1 อัลกอริธึมเลือกเส้นทางจำแนกตามจำนวนของโหนดปลายทาง 9

2.2.1.2 อัลกอริธึมเลือกเส้นทางจำแนกตามการควบคุมอำนาจการ ตัดสินใจเลือกเส้นทาง 10

2.2.1.2.1 การหาเส้นทางโดยใช้ข้อมูลรวม 10

2.2.1.2.2 การหาเส้นทางแบบโครงสร้างลำดับชั้น(hierarchical routing) . 10

2.2.1.2.3 การเลือกเส้นทางแบบฟลัดดิ้ง 12

2.2.1.3 การกำหนดเส้นทางแบบสถิตย์ (Static Routing) 13

สารบัญ(ต่อ)

หน้า

2.2.1.4 อัลกอริธึมเลือกเส้นทางจำแนกตามการคำนึงถึงความต้องการของผู้ใช้ 13
2.2.1.4.1 การเลือกเส้นทางทางที่สั้นที่สุด 13
2.2.1.4.2 การเลือกเส้นทางแบบตารางระยะทาง 14
2.2.1.4.3 โพรโตคอลเลือกทางเดินของเกตเวย์ภายในเครือข่าย
แบบ OSPF 17
2.3 เทคโนโลยีในการสลับแพ็คเก็ต (MPLS) 20
2.3.1 การหาเส้นทางและการสวิทชิ่งแบบดั่งเดิม 20
2.3.2 เทคโนโลยีในการสลับแพ็คเก็ต (Multi-Protocol Label Switch) 21
2.3.2.1 หน้าที่ของเทคโนโลยีในการสลับแพ็คเก็ต 22
2.3.2.2 ประเภทของเราท์เตอร์ภายในเครือข่าย MPLS..... 22
2.3.3 การกำหนดกลุ่มของแพ็คเก็ต FEC (Forward Equivalence Class) 22
2.3.4 เลเบิล และการกำหนดเลเบิล 23
2.3.5 การสร้างเลเบิล 25
2.3.6 การประกาศเลเบิล (Label Distribution) 25
2.3.7 การสลับเลเบิลเส้นทาง (Label-Switched Path) 26
2.3.8 ขอบเขตของหมายเลขเลเบิล (LSP Space) 27
2.3.9 การรวมเลเบิล (Label Merging) 27
2.3.10 การรักษาเลเบิล (Label Retention) 27
2.3.11 การควบคุมการประกาศเลเบิล (Label Control) 28
2.3.12 กลไกของสัญญาณควบคุม(Signaling Mechanism) 28
2.3.13 โปรโตคอลประกาศค่าเลเบิลในระบบเครือข่าย (Label Distribution Protocol) ... 29
2.3.14 เลเบิลแบบสแตค(Label Stack) 29
2.3.15 การบริหารการจราจรของทราฟฟิก (Traffic Engineering) 29
2.4 การหาเส้นทางแบบคำนึงถึงความต้องการแอพพลิเคชั่น (QoS Based routing) 30
2.4.1 ประสิทธิภาพในการจัดตั้งเส้นทางที่มีคุณภาพในการให้บริการ 31

สารบัญ(ต่อ)

หน้า

2.4.2 อัลกอริธึมสำหรับจัดตั้งเส้นทางที่มีคุณภาพในการให้บริการ 31

2.5 การเชื่อมต่อของแอพพลิเคชั่น (MPLS operation) 32

2.6 โครงสร้างการทำงานของโปรโตคอลสื่อสาร (MPLS Protocol Stack Architecture) 34

2.7 แอพลิเคชั่นของเครือข่ายเทคโนโลยีในการสลับแพ็คเก็ต (MPLS Applications) 35

บทที่ 3 อัลกอริธึมการสร้างเส้นทางแบบคำนึงถึงคุณภาพของการให้บริการสำหรับระบบ เครือข่าย MPLS ที่มีความสามารถในการแก้ไขข้อบกพร่องในการเลือกเส้นทางได้อย่างรวดเร็ว (Early Crankback QoS-Based Routing Algorithm : ECQR) 37

3.1 บทนำ 37

3.2 การแก้ไขปัญหาในการเลือกเส้นทางในเครือข่ายย่อย 38

3.3 ประสิทธิภาพของอัลกอริธึม ECQR 41

3.4 อัลกอริธึมการทำงานของเราเตอร์ LER และ LSR 43

3.4.1 อัลกอริธึมการทำงานของเราเตอร์ LER 43

3.4.2 อัลกอริธึมการทำงานของเราเตอร์ LSR 44

บทที่ 4 การทดลองและผลการทดลอง 45

4.1 การทดลอง 45

4.2 ผลการทดลอง 54

บทที่ 5 บทสรุปและแนวทางการพัฒนา 59

5.1 สรุปผลการทดลอง 59

5.2 แนวทางการพัฒนาต่อไป 59

เอกสารอ้างอิง 61

ภาคผนวก 63

บทความวิจัยที่ได้รับการตีพิมพ์ 64

ประวัติผู้เขียน 65

สารบัญรูป

รูปที่ หน้า

2.1 ความขัดแย้งระหว่างความเหมาะสมและความเป็นธรรม 9

2.2 การเลือกเส้นทางแบบโครงสร้างลำดับชั้น 11

2.3 ขั้นตอนในการคำนวณเส้นทางเดินที่สั้นที่สุด 14

2.4 (a) เครือข่ายย่อย 15

(b) แสดงข้อมูลใหม่จาก A,I,H,K และตารางทางเดินข้อมูลของ J 15

2.5 กรณีศึกษาของปัญหา Count-to-infinity 16

2.6 ระบบเครือข่ายอัตโนมัติ 18

2.7 กราฟที่ใช้แทนระบบเครือข่ายอัตโนมัติ 18

2.8 ความสัมพันธ์ระหว่างระบบเครือข่ายอัตโนมัติ,ระบบสื่อสารหลัก และเขตย่อยใน OSPF 19

2.9 รูปแบบเลเบิลของ MPLS 24

2.10 ATM กับชั้นเชื่อมต่อข้อมูล 24

2.11 Frame Relay กับชั้นเชื่อมต่อข้อมูล 25

2.12 กลไกสัญญาณ 28

2.13 การทำงานของระบบเครือข่าย MPLS ตั้งแต่เริ่มต้นร้องขอเส้นทางไปจนกระทั่งส่งแพ็คเก็ตข้อมูล 32

2.14 ลำดับชั้นและการทำงานร่วมกันระหว่างโปรโตคอลต่าง ๆ 35

3.1 flow chart การทำงานของโหนด LER 43

3.2 flow chart การทำงานของโหนด LSR 44

4.1 เครือข่ายโทโปโลยีตัวอย่าง 46

4.2 ตัวอย่างเส้นทางที่เป็นได้สำหรับการเชื่อมต่อที่กำหนดขึ้น 46

4.3 การร้องขอเส้นทางเลือกที่ 1 ในกรณีที่เกิดการบล๊อก (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป) 49

4.4 การส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางย้อนกลับมาถึงโหนดปากทาง (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป) 49

4.5 การร้องขอเส้นทางเลือกที่ 2 ในกรณีที่เกิดการบล๊อก (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป) 50

4.6 การส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางที่ 2 ย้อนกลับมาถึงโหนดปากทาง (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป) 50

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.7 การร้องขอการจัดตั้งเส้นทางบนเส้นทางที่ 3 ในกรณีที่สำเร็จ (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป) 51

4.8 การร้องขอเส้นทางเลือกที่ 1 ในกรณีที่เกิดการบล๊อก (ด้วยการใช้อัลกอริธึม ECQR) 51

4.9 การส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางย้อนกลับไปยังโหนด LSR6 ซึ่งเป็นโหนดที่เชื่อมระหว่างเส้นทางที่ 1 และเส้นทางที่ 2 (ด้วยอัลกอริธึม ECQR) 52

4.10 การร้องขอเส้นทางเลือกที่ 2 ในกรณีที่เกิดการบล๊อก (ด้วยการใช้อัลกอริธึม ECQR) 52

4.11 การส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางย้อนกลับไปยังโหนด LSR7 ซึ่งเป็นโหนดที่เชื่อมระหว่างเส้นทางที่ 2 และเส้นทางที่ 3 (ด้วยอัลกอริธึม ECQR) 53

4.12 การร้องขอเส้นทางเลือกที่ 3 ในกรณีที่สำเร็จ (ด้วยอัลกอริธึม ECQR) 53

4.13 แสดงการเปรียบเทียบค่า D_total ระหว่างวิธี ECQR (D_total (New)) และวิธีที่มีอยู่เดิม (D_total (Old)) 54

4.14 แสดงการเปรียบเทียบเวลาหน่วง ในการสร้างเส้นทางในระบบเครือข่ายที่มีค่า $\beta_{k_r}(i) = 0.05$ 55

4.15 แสดงการเปรียบเทียบเวลาหน่วง ในการสร้างเส้นทางในระบบเครือข่ายที่มีค่า $\beta_{k_r}(i) = 0.18$ 56

4.16 แสดงการเปรียบเทียบเวลาหน่วง ในการสร้างเส้นทางในระบบเครือข่ายที่มีค่า $\beta_{k_r}(i) = 0.25$ 57

บทที่1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบสื่อสารในปัจจุบันถูกพัฒนาเพื่อรองรับการใช้งานร่วมกันระหว่างระบบสื่อสารที่แตกต่างกันไป ทั้งแบบซิงโครนัส (Synchronous) เช่น TDM (Time Division multiplex) และแบบกึ่งอะซิงโครนัส ซึ่งสามารถรับประกันคุณภาพของการบริการ (Quality of service guarantee) ได้สำหรับทราฟฟิคหลายประเภท ไม่ว่าจะเป็นแบบอัตราเร็วคงที่ เช่น CBR (Constant bit rate) ซึ่งมีลักษณะการส่งผ่านคล้ายคลึงกับกับระบบ TDM หรือการส่งผ่านแบบที่มีอัตราเร็วไม่คงที่เช่น VBR (Variable bit rate) ซึ่งแบ่งออกเป็นแบบต่อเนื่อง (Realtime VBR : RT-VBR) และแบบไม่ต่อเนื่อง (Non-realtime VBR : NRT-VBR) ซึ่งแตกต่างกันตรงที่การกำหนดเวลาในการตอบโต้ ในกรณีที่เป็นแบบ RT-VBR จะมีข้อกำหนดของช่วงเวลารับส่ง เช่น การส่งข้อมูลภาพเคลื่อนไหวที่มีการบีบอัดข้อมูล เช่น MPEG (Moving Picture Expert Group) ซึ่งใช้เทคนิคการสร้างเฟรมและส่งเฟรมออกมาอย่างต่อเนื่อง ดังนั้นอัตราการส่งผ่านจะไม่คงที่จึงต้องมีการจัดการกับเวลาหน่วง (delay) ที่ดี ส่วนแบบ NRT-VBR นั้นจะไม่กำหนดเวลา เช่นการส่งภาพเคลื่อนไหวไปพร้อมกับจดหมายอิเล็กทรอนิกส์ (e-mail) และ web browser เป็นต้น ส่วนระบบสื่อสารแบบ ABR (Available bit rate) เป็นระบบที่มีการส่งข้อมูลกลับเพื่อแจ้งให้โหนดต้นทางปรับเปลี่ยนอัตราเร็วในการส่งตามความคับคั่งของทราฟฟิค และระบบสื่อสารแบบ UBR (Unspecified bit rate) ซึ่งไม่มีการรับประกันและไม่บอกสถานะการรับส่งข้อมูล เช่นการให้บริการสำหรับ IP แพ็กเก็ตเป็นต้น

วิธีการหาเส้นทางและสำรองทรัพยากร [7] ในสวิตชิ่งโหนดของการสื่อสารแต่ละระบบนั้นจะต่างกันไปตามความต้องการในการรับประกันคุณภาพของการให้บริการ การพิจารณาเลือกเส้นทางจะพิจารณาจากทราฟฟิคพารามิเตอร์ (traffic parameter) ที่ถูกระบุมากับแพ็กเก็ตของการร้องขอเส้นทาง (Call request packet) โดยอาจจะพิจารณาเลือกเส้นทางที่เป็นไปได้โดยพิจารณาค่าใช้จ่ายหรือภาระการใช้ลิงค์ (cost) ที่ต่ำที่สุด หรือมีแบนด์วิดท์ที่มากที่สุด หรือมีการควบคุมเวลาหน่วงและการแกว่งของเวลาหน่วงที่ดี เป็นต้น การพิจารณาเลือกเส้นทางโดยทั่วไปจะพิจารณาทราฟฟิค พารามิเตอร์ตัวใดตัวหนึ่งเป็นหลักเท่านั้น การที่จะพิจารณาทราฟฟิคพารามิเตอร์ทุกตัวจะทำก็ต่อเมื่อเป็นการสื่อสารที่รองรับการบริการหลายประเภท ซึ่งการหาเส้นทางสามารถจะรองรับบริการสองประเภทเหล่านั้นถูกเรียกรวมกันว่า QoS based Routing [5] โดยที่พารามิเตอร์ส่วนใหญ่จะเป็นเพียงตัวควบคุมและจะมีพารามิเตอร์หลักเพียงตัวเดียวในการตัดสินใจ การพิจารณาเส้นทางที่เป็นไปได้จะทำโดยตัดเส้นทางที่ไม่ต้องการออกไปเหลือเพียงเส้นทางที่อยู่ในความควบคุมแล้วใช้พารามิเตอร์หลักเป็นตัวเลือกเส้นทางที่ต้องการ และระบบสื่อสารแบบแพ็กเก็ตข้อมูลได้มีการพัฒนากัน

อย่างต่อเนื่องทั้งนี้เพื่อตอบสนองต่อความต้องการของผู้ใช้ซึ่งมีความแตกต่างกันอย่างหลากหลายตามการพัฒนาไปของรูปแบบแอพลิเคชั่น ซึ่งการที่จะรับประกันคุณภาพในการให้บริการแก่ผู้ใช้ในระดับที่ผู้ใช้พึงพอใจนั้นจำเป็นต้องอาศัยกลไกหลายอย่างภายในระบบเครือข่าย ทั้งส่วนของการหาเส้นทาง (routing) การร้องขอและอนุมัติ เส้นทาง (call-setup และ call admission control) การควบคุมปริมาณทราฟฟิก (traffic shaping) การตรวจวัดและกำจัดแพ็กเก็ตเป็นต้น ซึ่งการค้นหาและสร้างเส้นทางนั้นเป็นกลไกแรกที่จะต้องใช้เพื่อสร้างเส้นทางที่มีทรัพยากรและคุณภาพที่เหมาะสมแก่ความต้องการของผู้ใช้

ในวิทยานิพนธ์นี้จึงได้นำเสนออัลกอริธึมสำหรับสร้างเส้นทางแบบกำหนดเส้นทางครั้งเดียว [15], [19], [20] (source routing call-setup process) โดยการคำนึงถึงคุณภาพในการให้บริการของเส้นทาง และสามารถแก้ไขปัญหาในกรณีที่พบว่าเราท์เตอร์ในเส้นทางย่อยบางตัวมีทรัพยากรไม่เพียงพอสำหรับให้บริการในเส้นทางแรกที่กำหนดให้ด้วยการเลือกเส้นทางเผื่อเลือกที่แนบมาพร้อมกับเส้นทางหลักในแพ็กเก็ตร้องขอการเชื่อมต่อ ทั้งนี้เป็นการปรับปรุงวิธีการของบทความ [4], [5] ที่จำเป็นต้องส่งแพ็กเก็ตร้องขอการเชื่อมต่อย้อนกลับไปถึงเราท์เตอร์ปากทางเพื่อให้กำหนดเส้นทางใหม่

1.2 วัตถุประสงค์ของวิทยานิพนธ์

เพื่อพัฒนาวิธีการหาเส้นทางที่เหมาะสมและเลือกใช้เส้นทางที่สามารถให้บริการที่สอดคล้องกับความต้องการของผู้ใช้ โดยจำกัดปริมาณของทราฟฟิกที่เกิดจากการควบคุมการร้องขอเส้นทางและเพื่อลดเวลาในกระบวนการจัดตั้งเส้นทางและแก้ไขความผิดพลาดของการเลือกเส้นทางได้เมื่อระบบเครือข่ายยังคงมีเส้นทางที่ให้บริการได้

1.3 ทฤษฎีที่เกี่ยวข้อง

- การออกแบบโปรแกรมสื่อสารชั้นควบคุมเครือข่าย
- อัลกอริทึมสำหรับเลือกทางเดินข้อมูล (Routing)
- เทคโนโลยีในการสลับแพ็คเก็ต (MPLS: Multi-Protocol Label Switch)
- การหาเส้นทางแบบคำนึงถึงความต้องการแอพพลิเคชั่น (QoS Based routing)
- การเชื่อมต่อของแอพพลิเคชั่น (MPLS operation)
- โครงสร้างการทำงานของโปรโตคอลสื่อสาร (MPLS Protocol Stack Architecture)
- แอพลิเคชั่นของเครือข่ายเทคโนโลยีในการสลับแพ็คเก็ต (MPLS Applications)

1.4 วิธีการที่นำเสนอ

เป็นการนำเสนออัลกอริธึมที่ใช้ในการสร้างเส้นทางในระบบเครือข่ายแบบคำนึงถึงคุณภาพของการให้บริการให้สอดคล้องกับความต้องการของผู้ใช้ โดยปรับปรุงวิธีการสร้างเส้นทางแบบกำหนดเส้นทางครั้งเดียวแบบหลายลำดับชั้น ซึ่งเพิ่มเติมความสามารถในการแก้ไขและเลือกเส้นทางย่อยใหม่ในกรณีที่พบว่าเส้นทางที่ถูกกำหนดโดยเราท์เตอร์ปากทางเข้าของระบบเครือข่ายย่อยไม่สามารถให้บริการที่ตอบสนองต่อความต้องการที่ถูกระบุไว้ในแพ็จเก็ตร้องขอการเชื่อมต่อได้ การแก้ปัญหาดังกล่าวใช้การบรรจุเส้นทางเผื่อเลือกหลาย ๆ เส้นทางลงในแพ็กเก็ตร้องขอการเชื่อมต่อซึ่งทำให้ไม่จำเป็นต้องย้อนกระบวนการสร้างเส้นทางกลับไปถึงโหนดเราท์เตอร์ปากทางเข้าในกรณีที่มีปัญหาเกิดขึ้น ซึ่งวิธีการที่เสนอสามารถช่วยลดเวลาที่อาจจะต้องสูญเสียไปในการสร้างเส้นทางได้อย่างมีนัยสำคัญ

1.5 ขอบเขตของวิทยานิพนธ์

- พัฒนาวิธีการหาเส้นทางที่เหมาะสมสำหรับระบบการสื่อสารแพ็กเก็ตข้อมูล
- จำลองวิธีการหาเส้นทางแบบคำนึงถึงคุณภาพการให้บริการที่มีความสามารถในการตอบสนองต่อความผิดพลาดของการกำหนดเส้นทาง
- ทดสอบประสิทธิภาพของวิธีการที่นำเสนอ ด้วยการจำลองการทำงานและวัดผลที่ได้เทียบกับวิธีการหาเส้นทางแบบอื่น ๆ

1.6 ขั้นตอนการทำวิทยานิพนธ์

1. กำหนดจุดประสงค์ หัวข้อ และขอบเขตการทำวิทยานิพนธ์
2. ศึกษาทฤษฎี หลักการพื้นฐาน ข้อมูล และแนวคิดที่จะนำมาใช้ในการทำวิทยานิพนธ์
3. ศึกษาพัฒนาวิธีการหาเส้นทาง
4. ออกแบบและ สร้างโมเดลจำลองเพื่อใช้ในการทดลองวัดประสิทธิภาพของวิธีการที่นำเสนอ
5. สรุป วิเคราะห์ผลการทดลอง และนำเสนอแนวทางในการทำวิจัยต่อไป
6. จัดทำเอกสารประกอบวิทยานิพนธ์

1.7 โครงสร้างของวิทยานิพนธ์

เนื้อหาในวิทยานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ ในบทนี้กล่าวถึงความเป็นมา ความสำคัญของปัญหา วัตถุประสงค์ ทฤษฎีและหลักการที่นำมาใช้ในงานวิจัย ขอบเขตของงานวิจัย ขั้นตอนการทำงาน และโครงสร้างของวิทยานิพนธ์

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง

- การออกแบบโปรแกรมสื่อสารชั้นควบคุมเครือข่าย
- อัลกอริธึมสำหรับเลือกทางเดินข้อมูล
- เทคโนโลยีในการสลับแพ็คเก็ต
- โครงสร้างการทำงานของโปรโตคอลสื่อสาร
- แอพลิเคชั่นของเครือข่ายเทคโนโลยีในการสลับแพ็คเก็ต
- อธิบายถึงปัญหาการเลือกเส้นทางในระบบเครือข่ายคอมพิวเตอร์ หลักการจำแนกการเลือกเส้นทาง และตัวอย่างการเลือกเส้นทาง

บทที่ 3 วิธีการที่นำเสนอ

- การแก้ไขปัญหาในการเลือกเส้นทางในเครือข่ายย่อย
- ประสิทธิภาพของอัลกอริธึม ECQR อธิบายถึงอัลกอริธึม ที่ใช้ในการสร้างเส้นทางในระบบเครือข่ายแบบคำนึงถึงคุณภาพของการให้บริการให้สอดคล้องกับความต้องการของผู้ใช้ เพื่อให้มีความสามารถในการแก้ไขข้อบกพร่องในการเลือกเส้นทางได้อย่างรวดเร็ว

บทที่ 4 การทดลองและผลการทดลอง ในบทนี้จะอธิบายถึงวิธีการทดลองและผลการทดลอง

บทที่ 5 บทสรุปและแนวทางในการพัฒนา อธิบายถึงบทสรุปและแนวทางการพัฒนาในอนาคต

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

การเลือกเส้นทางจากโหนดต้นทางไปยังโหนดปลายทางเป็นกิจกรรมอันหนึ่งที่มีขึ้นภายใต้กติกาการสื่อสารมาตราฐาน ISO (Open System Interconnection) ในเลเยอร์ที่ 3 หรือชั้นควบคุมเครือข่าย (Network Layer) ซึ่งภายในชั้นนี้จะมีโครงสร้างการทำงานดังต่อไปนี้

ในโปรแกรมการสื่อสารชั้นควบคุมเครือข่ายมีหน้าที่ในการควบคุมดูแล และจัดส่งข้อมูลแต่ละชิ้นจากผู้ส่งไปถึงจุดหมาย หรือโหนดผู้รับที่กำหนด ถ้าการจัดส่งดังกล่าวเป็นเพียงการจัดส่งข้อมูลภายในกลุ่มเดียวกัน โหนดผู้ส่งสามารถติดต่อกับโหนดผู้รับได้โดยตรง ส่วนการส่งข้อมูลระหว่างผู้รับและผู้ส่งที่อยู่ห่างไกลกันโดยไม่มีการเชื่อมต่อตรงด้วยวิธีการใด ๆ ในกรณีนี้โหนดของกลุ่มผู้ส่งจำเป็นจะต้อง "ฝาก" ข้อมูลไปยังโหนดกลุ่มที่อยู่ติดกัน ซึ่งก็จะ "ฝาก" ข้อมูลนั้นต่อไปเรื่อย ๆ จนถึงโหนดของกลุ่มผู้รับ ซึ่งจะส่งข้อมูลไปยังผู้รับตัวจริงได้

เพื่อให้บรรลุวัตถุประสงค์ดังกล่าว "การรับ-ส่งข้อมูล" จึงได้รับการพัฒนาขึ้นมาใช้งานในชั้นควบคุมเครือข่าย เมื่อรับข้อมูลที่ถูกส่งมาจากผู้ใดก็ตาม โหนดต่าง ๆ ที่เข้าใจกฎเกณฑ์การรับ-ส่งข้อมูลก็จะตรวจสอบข้อมูลที่ได้รับ และพิจารณาเลือกเส้นทางที่จะส่งต่อไปให้เหมาะสมที่สุด คือจะไม่ส่งข้อมูลไปยังระบบเครือข่ายย่อยที่มีงานล้นอยู่โดยปล่อยให้อีกระบบเครือข่ายย่อยหนึ่งไม่มีงานทำ นอกจากนี้ยังสามารถแก้ไขปัญหาในการส่งข้อมูลผ่านระบบที่แตกต่างกันได้ด้วย ซึ่งการจัดการในส่วนต่าง ๆ นั้นจะมีรายละเอียดดังต่อไปนี้

2.1 การออกแบบชั้นควบคุมเครือข่าย (Network Layer Design Issue)

ในส่วนนี้จะกล่าวถึงข้อพิจารณาเบื้องต้นสำหรับการออกแบบชั้นควบคุมเครือข่าย [17] รวมถึงบริการต่าง ๆ ที่จะต้องจัดเตรียมไว้สำหรับให้ชั้นนำส่งข้อมูลเรียกใช้ได้โดยสะดวก

2.1.1 บริการสำหรับชั้นนำส่งข้อมูล (Service Provided to the transport Layer)

โปรแกรมสื่อสารควบคุมเครือข่ายให้บริการแก่โปรแกรมในชั้นนำส่งข้อมูลผ่านทางส่วนติดต่อ (Interface) ระหว่างโปรแกรมสื่อสารทั้งสอง ส่วนติดต่อนี้ยังมีความสำคัญอีกประการหนึ่งคือ เป็นการเชื่อมต่อระหว่างโหนดผู้นำส่งข้อมูล ซึ่งอาจจะเป็นโหนดผู้รับข้อมูลจริงหรือเป็นเพียงตัวกลางที่จะต้องนำข้อมูลนั้นส่งต่อไปอีกก็ได้ ในกรณีส่วนติดต่อจะทำงานที่เขตติดต่อระหว่างเครือข่ายย่อย คือโหนดผู้นำส่งข้อมูลจะอยู่ในระบบเครือข่ายย่อยหนึ่ง ส่วนโหนดผู้รับช่วงข้อมูลอยู่ในระบบเครือข่ายย่อยอื่นที่อยู่ติดกันโดยปกติโหนดผู้นำส่งข้อมูลมักจะเป็นโหนดผู้ควบคุมโพรโตคอลตั้งแต่

โพรโตคอลตั้งแต่ระดับล่างขึ้นไปจนถึงส่วนติดต่อในโปรแกรมสื่อสารชั้นควบคุมเครือข่าย ซึ่งมีความรับผิดชอบในการนำแพ็กเก็ตไปส่งให้ได้ ส่วนติดต่อจึงต้องได้รับการออกแบบเป็นอย่างดี

บริการในโปรแกรมสื่อสารชั้นควบคุมเครือข่ายมีวัตถุประสงค์ในการออกแบบ คือ

1 บริการที่มีให้แก่โปรแกรมสื่อสารชั้นนำส่งข้อมูลต้องเป็นอิสระจากเทคโนโลยีที่ใช้ในระบบเครือข่ายย่อย

2 โปรแกรมสื่อสารชั้นนำส่งข้อมูลจะต้องไม่เข้ามาเกี่ยวข้องกับโครงสร้างของระบบเครือข่าย

3 วิธีการกำหนดที่อยู่ในเครือข่ายที่ส่งให้โปรแกรมสื่อสารชั้นนำส่งข้อมูลจะต้องเป็นที่อยู่มาตรฐานที่เป็นที่เข้าใจได้แม้โปรแกรมสื่อสารในระดับต่ำกว่าจะแตกต่างกัน

2.1.2 การจัดโครงสร้างภายในชั้นควบคุมเครือข่าย

การจัดโครงสร้างระบบย่อยแบ่งออกเป็นสองแบบคือ โครงสร้างที่ใช้การสื่อสารแบบต่อเนื่องและโครงสร้างที่ใช้การสื่อสารแบบไม่ต่อเนื่อง ช่องสื่อสารสำหรับการเชื่อมต่อภายในระบบเครือข่ายย่อยแบบต่อเนื่องเรียกว่าวงจรชั่วคราวหรือวงจรเสมือน (Virtual Circuit) ซึ่งเปรียบเทียบได้กับการเชื่อมต่อสัญญาณที่ใช้ในระบบโทรศัพท์ ส่วนแพ็กเก็ตที่ใช้ในโครงสร้างการสื่อสารแบบไม่ต่อเนื่องเรียกว่าดาต้าแกรม (Datagram) เปรียบเทียบได้กับข้อความที่ส่งไปในระบบโทรเลข

แนวความคิดหลักของการใช้วงจรเสมือนก็เพื่อต้องการหลีกเลี่ยงการค้นหาเส้นทางเดินข้อมูลในทุก ๆ ครั้งที่ส่งแพ็กเก็ต ดังนั้นตั้งแต่เริ่มต้นการติดต่อ โปรแกรมในชั้นควบคุมเครือข่ายจะต้องจัดการกำหนดเส้นทางเดินสำหรับการรับ-ส่งข้อมูลจากผู้ส่งไปยังผู้รับขึ้นมาใช้งาน จากนั้นข้อมูลทั้งหมดจะถูกส่งไปยังผู้รับโดยใช้เส้นทางนี้ตลอดช่วงของการสื่อสาร เมื่อการติดต่อสิ้นสุดลงเส้นทางนี้จะถูกยกเลิก การกำหนดเส้นทางเดินข้อมูลจึงเป็นการกำหนดขึ้นมาใช้เป็นการชั่วคราวซึ่งมีลักษณะการทำงานแบบเดียวกับวิธีที่ใช้ในระบบโทรศัพท์ ส่วนที่เรียกว่าเป็นเส้นทางเสมือนนั้นก็เนื่องจากว่าเส้นทางสื่อสารชั่วคราวนี้จะไม่มีการกำหนดเป็นที่แน่นอน การกำหนดเส้นทางเดินข้อมูลแต่ละครั้งจึงไม่มีการรับประกันว่าจะเป็นเส้นทางเดิมที่เคยใช้ในการติดต่อครั้งที่แล้วมา

ในทางตรงข้ามดาต้าแกรมแต่ละตัวจะไม่มีการกำหนดเส้นทางเดินของข้อมูลเอาไว้ล่วงหน้าแม้ว่าจะใช้การสื่อสารแบบต่อเนื่องก็ตาม แพ็กเก็ตแต่ละตัวจะถูกส่งไปยังผู้รับโดยใช้เส้นทางเดินข้อมูลที่เป็นอิสระต่อแพ็กเก็ตอื่น ๆ ดังนั้นแม้ว่าแพ็กเก็ตจะถูกส่งออกมาอย่างต่อเนื่องกันไปก็ไม่มีการรับประกันใด ๆ แพ็กเก็ตทั้งหมดจะเดินทางไปถึงผู้รับด้วยเส้นทางเดียวกัน อย่างไรก็ตามการส่งข้อมูลด้วยวิธีนี้ทำให้เกิดความอ่อนตัวในการส่งข้อมูล โดยเฉพาะในกรณีที่เกิดความคับคั่งของเส้นทางเดินข้อมูลบางส่วนหรือเส้นทางบางส่วนเกิดการเสียหายจนใช้การไม่ได้ ดาต้าแกรมจะสามารถเลือกตารางเส้นทางอื่นเพื่อไปยังผู้รับได้โดยอัตโนมัติ

เมื่อกำหนดเส้นทางวงจรเสมือนขึ้นมาใช้งาน โหนดแต่ละตัวที่อยู่ในเส้นทางจะต้องบันทึกข้อมูลไว้ในตารางภายในของโหนดว่าจะต้องส่งแพ็กเก็ตของแต่ละวงจรเสมือนไปยังโหนดตัวใด ตารางนี้จะต้องทำการบันทึกข้อมูลสำหรับทุกวงจรเสมือนที่ถูกกำหนดขึ้นใช้งานดังกล่าวไว้ข้างต้น

เมื่อมีการกำหนดวงจรเสมือนก็จะต้องกำหนดหมายเลขพิเศษกำกับไว้เสมอ ในแต่ละแพ็กเก็ตจึงต้องบันทึกหมายเลขพิเศษไว้แทนที่อยู่ของผู้รับและส่ง โหนดจะทำการตรวจสอบหมายเลขเหล่านี้กับตารางที่เก็บไว้และจะทราบว่าจะต้องส่งแพ็กเก็ตไปยังเส้นทางใด

วงจรเสมือนแบ่งออกตามขอบเขตของระบบเครือข่ายเป็น 2 ประเภท คือวงจรเสมือนภายในและวงจรเสมือนภายนอก วงจรเสมือนภายในนั้นเป็นการกำหนดเส้นทางของข้อมูลภายในระบบเครือข่ายย่อยระบบหนึ่งอย่างเป็นอิสระต่อวิธีการที่ใช้ในระบบเครือข่ายย่อยอื่น ตั้งแต่โหนดตัวแรกที่รับแพ็กเก็ตจากระบบเครือข่ายย่อยอื่นเข้ามา (หรือเป็นผู้ส่งข้อมูลเอง) ไปจนถึงโหนดตัวสุดท้ายก่อนที่จะส่งแพ็กเก็ตนั้นออกไปภายนอกระบบเครือข่ายย่อยนั้น คือจากเครือข่ายย่อยของผู้ส่งจะต้องผ่านเครือข่ายย่อยใดบ้างไปจนถึงเครือข่ายย่อยของผู้รับ

2.2 อัลกอริธึมสำหรับเลือกทางเดินข้อมูล (Routing Algorithm)

หน้าที่หลักของชั้นควบคุมเครือข่าย คือการจัดส่งแพ็กเก็ตจากเครื่องผู้ส่งไปยังเครื่องผู้รับในระบบเครือข่ายย่อย ส่วนมากแพ็กเก็ตจะต้องถูก "รับแล้วส่งต่อ" (Hop) หลายครั้งกว่าจะถึงจุดหมายปลายทาง ถ้าผู้ส่งและผู้รับไม่ได้อยู่ในระบบเครือข่ายเดียวกัน ดังนั้นอัลกอริธึมและโครงสร้างข้อมูลที่ทำหน้าที่เลือกเส้นทางของข้อมูล จึงเป็นส่วนประกอบหลักในการออกแบบชั้นควบคุมเครือข่าย

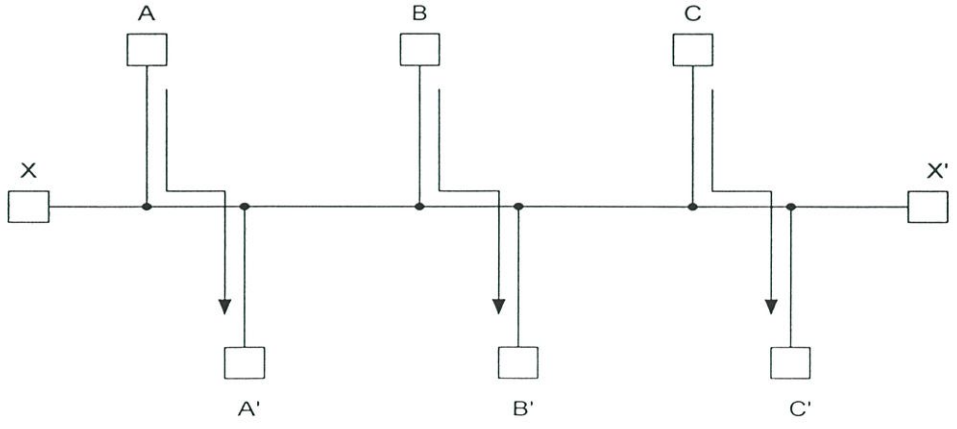
อัลกอริธึมสำหรับเลือกเส้นทางของข้อมูล [1] ทำหน้าที่ตัดสินใจให้โหนดว่าแพ็กเก็ตที่รับเข้ามาควรจะส่งต่อไปให้ที่โหนดตัวใดเป็นลำดับต่อไป ถ้าเครือข่ายย่อยใช้ดาต้าแกรมในการส่งข้อมูลภายในแล้ว การตกลงใจของแต่ละโหนดจะเกิดขึ้นทุกครั้งที่รับดาต้าแกรมตัวใหม่เข้ามา เพราะทางเลือกที่ดีที่สุด (สำหรับปลายทางเดิม) อาจเปลี่ยนแปลงไปแล้ว ถ้าเครือข่ายย่อยใช้วงจรเสมือนในการส่งข้อมูล การเลือกเส้นทางเดินข้อมูลตลอดทั้งเครือข่ายจะเกิดขึ้นเพียงครั้งเดียวในช่วงการจัดตั้งช่องสื่อสาร แพ็กเก็ตที่ถูกส่งมาในลำดับหลังจะถูกส่งไปยังโหนดในเส้นทางที่จัดตั้งไว้แล้วที่แพ็กเก็ตแรก ๆ ถูกส่งไปอยู่เสมอ วิธีการเลือกเส้นทางเดินแบบนี้บางครั้งเรียกว่า "เซสชั่นเราติ้ง (Session Routing)" เนื่องจากเส้นทางจะคงเดิมตลอดช่วงเวลาสื่อสารที่กำหนดขึ้น

ไม่ว่าเส้นทางเดินข้อมูลจะได้รับการกำหนดขึ้นมาด้วยอัลกอริธึมใดก็ตาม อัลกอริธึมที่เลือกใช้ควรจะต้องมีคุณลักษณะ 6 ประการคือความถูกต้อง (Correctness) ความง่าย (Simplicity) ความคงทน (Robustness) ความแน่นอน (Stability) ความเป็นธรรม (Fairness) และความเหมาะสม (Optimality) ความถูกต้องและความง่ายเป็นคุณสมบัติพื้นฐานที่สำคัญสองข้อแรกที่ทุกอัลกอริธึมจะต้องมี ส่วนความคงทนของอัลกอริธึมสามารถพิจารณาได้จากเครือข่ายใหม่ที่เพิ่งได้รับการจัดตั้งขึ้นมาใช้งาน ซึ่งจะมีความต้องการที่ให้บริการคงอยู่โดยไม่ประสบปัญหาความล้มเหลวในระดับเครือข่าย แต่ความล้มเหลวของอุปกรณ์หรือโปรแกรมบางส่วนเป็นเหตุการณ์ที่เกิดขึ้นได้ตลอดเวลาสำหรับทุกระบบและไม่สามารถหลีกเลี่ยงได้ เช่น โฮสต์ เราท์เตอร์ และสายสื่อสารอาจเกิดการเสีย

หายเมื่อใดก็ได้ซึ่งจะต้องได้รับการซ่อมแซมอยู่เสมอ หรือการจัดโครงสร้างของเครือข่าย (Network Topology) อาจมีการเปลี่ยนแปลงได้เช่นกัน สภาพปัญหาและการเปลี่ยนแปลงเหล่านี้ได้ เมื่อเกิดปัญหาขึ้นโฮสต์ส่วนใหญ่จะต้องสามารถทำงานต่อไปได้ โดยไม่ต้องเข้ามามีส่วนร่วมในการแก้ไขหรือจัดการปัญหาที่เกิดขึ้น

ความแน่นอนก็เป็นส่วนสำคัญมากเช่นกัน อัลกอริธึมสำหรับเลือกเส้นทางข้อมูลจำนวนไม่น้อยที่ไม่สามารถจัดการให้ระบบเครือข่ายเข้าสู่สถานะความสมดุลในการกระจายปริมาณทราฟฟิก ทำให้ระบบอยู่ในสภาวะที่เกิดความไม่แน่นอนในการสื่อสารข้อมูลในระดับสูงซึ่งเป็นสภาวะที่ควรหลีกเลี่ยง ส่วนความเป็นธรรม และความเหมาะสมเป็นสิ่งที่ผู้ใช้ทุกคนในระบบต้องการอย่างไม่ต้องสงสัยแต่ไม่ว่าจะทำอย่างไรก็ตามในทางปฏิบัติจะต้องมีผู้ที่ไม่ได้รับความเป็นธรรมอยู่บ้าง ในรูปที่ 2.1 แสดงระบบเครือข่ายหนึ่งที่สมมติให้ปริมาณข้อมูลไหลจาก A ไป A' จาก B ไป B' และจาก C ไป C' มีปริมาณมากพอที่จะทำให้การส่งข้อมูลแนวราบอยู่ในสภาวะอิ่มตัว ถ้าต้องการให้ระบบนี้มีประ สิทธิภาพโดยรวมสูงสุดแล้วการสื่อสารระหว่าง X และ X' ควรจะต้องถูกระงับไว้ก่อนมิฉะนั้นจะทำให้การสื่อสารทั้งหมดที่เหลือหยุดชะงัก ผลคือทำให้ประสิทธิภาพโดยรวมของระบบต่ำลง การตัดสินใจเช่นนี้เป็นการขัดต่อความเป็นธรรมที่พึงจะให้แก่สมาชิกทุกคนของระบบอย่างเท่าเทียมกัน ดังนั้นจึงต้องมีการประนีประนอมระหว่างความเหมาะสมกับความเป็นธรรม

สิ่งหนึ่งที่ทุกระบบต้องการ คือ การลดเวลารอคอยโดยเฉลี่ยของแพ็กเก็ตและการเพิ่มอัตราการสัมฤทธิ์ผล (Throughput) ความต้องการทั้งสองข้อนี้ขัดแย้งซึ่งกันและกัน เนื่องจากการเพิ่มอัตราสัมฤทธิ์ผลของระบบต้องการให้แพ็กเก็ตข้อมูลมารอคอยอยู่ในโหนดให้มากที่สุด เพื่อจะได้รักษาความต่อเนื่องในการส่งข้อมูลให้อยู่ในระดับสูงสุดตลอดเวลา แต่การทำงานนี้กลับทำให้การรอคอยโดยเฉลี่ยของแต่ละแพ็กเก็ตสูงขึ้น เพื่อลดความขัดแย้งดังกล่าวเครือข่ายหลายระบบพยายามลดจำนวนโหนดที่ใช้เป็นตัวกลางลงไปให้มากที่สุด ทั้งนี้จะทำให้การรอคอยของแพ็กเก็ตลดลงในขณะที่ช่วยเพิ่มอัตราการสัมฤทธิ์ผลของระบบได้ในเวลาเดียวกัน



รูปที่ 2.1 ความขัดแย้งระหว่างความเหมาะสมและความเป็นธรรม

2.2.1 หลักการจำแนกประเภทของการเลือกเส้นทาง

อัลกอริธึมการเลือกเส้นทางที่ใช้ในการค้นหาเส้นทางที่เชื่อมต่อ จากโหนดต้นทางถึงกลุ่มผู้รับของเซสชั่นนั้น โดยปกติแล้วอัลกอริธึมเส้นทางที่สั้นที่สุดจะเลือกเส้นทางที่มีดีเลย์ต่ำสุดหรือเส้นทางที่มีจำนวนลิงค์น้อยที่สุด ในเครือข่ายคอมพิวเตอร์มีหลาย ๆ อัลกอริธึมการเลือกเส้นทาง ซึ่งเราสามารถจัดประเภทของอัลกอริธึมการเลือกเส้นทางโดยการใช้หลักการ [21], [23] ดังต่อไปนี้

2.2.1.1 อัลกอริธึมเลือกเส้นทางจำแนกตามจำนวนของโหนดปลายทาง

พิจารณาตามจำนวนของโหนดปลายทาง ประเภทของอัลกอริธึมการเลือกเส้นทางมี 3 ประเภท ดังต่อไปนี้

1. ยูนิคาสต์ (Unicast) เป็นการสื่อสารจากโหนดต้นทางหนึ่งโหนดถึงโหนดปลายทาง
2. มัลติคาสต์ (Multicast) เป็นการสื่อสารจากโหนดต้นทางหนึ่งถึงโหนดปลายทางจำนวนหนึ่ง
3. บรอดคาสต์(Broadcast) เป็นการสื่อสารจากโหนดต้นทางหนึ่งโหนดถึงโหนดปลายทางทุกโหนด

แอพพลิเคชั่นการสื่อสารหลายจุดมักเป็นแอพพลิเคชั่นแบบสื่อผสม เช่นการประชุมผ่านระบบเครือข่ายโดยปกติแล้วต้องการการเลือกเส้นทางที่แบบมัลติคาสต์ เพื่อรองรับการเลือกเส้นทางจากผู้ส่งไปยังผู้รับทุกโหนด อัลกอริธึมการเลือกเส้นทางมัลติคาสต์ที่มีประสิทธิภาพจะทำให้การจัดการทรัพยากรมีประสิทธิภาพมากกว่าบรอดคาสต์

2.2.1.2 อัลกอริธึมที่อัลกอริธึมเลือกเส้นทางจำแนกตามการควบคุมอำนาจการตัดสินใจเลือกเส้นทาง

2.2.1.2.1 การหาเส้นทางโดยใช้ข้อมูลรวม

เนื่องจากรูปร่างและลักษณะการสัญจรของทราฟฟิกภายในเครือข่ายมีการเปลี่ยนแปลงอยู่เสมอ จึงต้องมีการคำนวณตารางหาเส้นทางบ่อยครั้ง โดยอัลกอริธึมการคำนวณนี้อาจจะทำที่ศูนย์กลางภายในระบบ วิธีนี้เรียกว่า "การหาเส้นทางโดยใช้ข้อมูลรวม" ซึ่งจะมี RCC (Routing Control Center) ทำหน้าที่คำนวณหาเส้นทาง สำหรับวิธีนี้ทุกช่วงเวลาหนึ่ง แต่ละโหนดในเครือข่ายจะส่งข้อมูลที่แสดงสภาวะของเครือข่ายให้แก่ RCC ข้อมูลเหล่านั้นอาจจะเป็นโหนดข้างเคียงที่ทำงานอยู่ จำนวนข้อมูลที่รอการส่งในบัฟเฟอร์ จำนวนของแพ็กเก็ตที่ถูกส่งผ่านแต่ละสายของโหนด เป็นต้น เมื่อ RCC ได้ข้อมูลเหล่านี้ก็จะทราบสภาวะของทั้งเครือข่าย และสามารถคำนวณหาเส้นทางที่ดีที่สุดระหว่างแต่ละคู่โหนดได้ โดยอาจใช้การคำนวณดังที่ได้อธิบายมาข้างต้น แล้วจึงส่งตารางหาเส้นทางที่คำนวณได้กลับคืนไปยังโหนดอื่น ๆ

ข้อดีคือ การหาเส้นทางได้ใช้ข้อมูลทั้งหมดของเครือข่ายทำให้ได้เส้นทางที่เหมาะสม นอกจากนี้ยังลดภาระของโหนดอื่น ๆ ในการคำนวณหาเส้นทาง

ข้อเสียก็คือหากเครือข่ายมีการเปลี่ยนแปลงบ่อย ทำให้ต้องมีการคำนวณบ่อยครั้งและสำหรับเครือข่ายขนาดใหญ่ต้องใช้เวลาในการคำนวณนาน ถึงแม้จะเป็นคอมพิวเตอร์ระดับใหญ่ก็ตาม อีกทั้งหากคอมพิวเตอร์ที่เป็น RCC เกิดเสียหรือสายสื่อสารเชื่อมต่อเข้าสู่ RCC เสียหมดก็จะไม่สามารถหาเส้นทางที่เหมาะสมสำหรับเครือข่ายได้ ซึ่งทางแก้อาจทำได้โดยการใช้คอมพิวเตอร์สำรอง (backup computer) ซึ่งทำให้สิ้นเปลืองค่าใช้จ่ายเพิ่มขึ้น นอกจากนั้นในการส่งตารางหาเส้นทางกลับคืนให้แก่โหนดต่างอัลกอริธึมนั้น โหนดที่อยู่ใกล้ RCC ก็จะได้รับตารางก่อนและใช้เส้นทางที่คำนวณใหม่ได้ทันที แต่โหนดที่อยู่ห่างไกลออกไปยังใช้ข้อมูลเดิม ดังนั้นช่วงก่อนที่ข้อมูลใหม่จะถูกส่งไปถึงทุกอัลกอริธึมโหนดนั้น การส่งข้อมูลก็จะไม่สอดคล้องเหมาะสม สำหรับข้อเสียได้แสดงในรูปที่ 10.25 ซึ่งจะเห็นว่าในการส่งข้อมูลมาสู่ RCC จะทำให้เกิดการแน่นขนัดของข้อมูลในสายสื่อสารที่อยู่ใกล้ RCC ทำให้เกิดความล่าช้ามากในบริเวณนี้

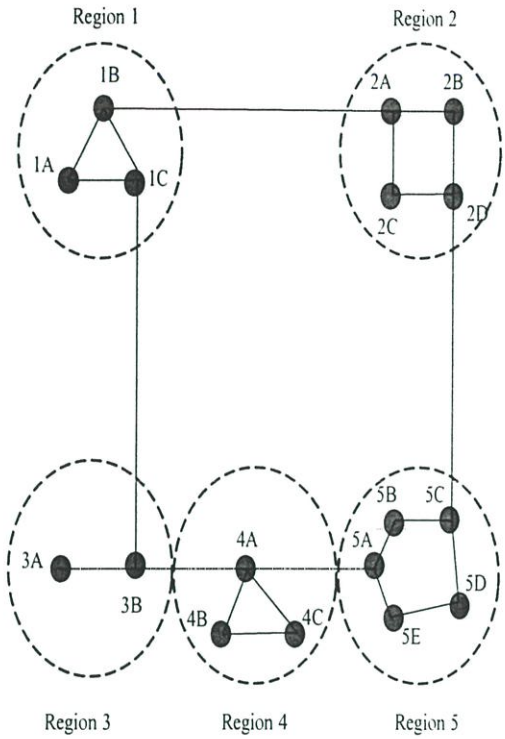
2.2.1.2.2 การหาเส้นทางแบบโครงสร้างลำดับชั้น (hierarchical routing)

ในการแก้ไขปัญหาของระบบเครือข่ายที่มีจำนวนเราท์เตอร์เพิ่มขึ้นซึ่งทำให้มีผลตามมาหลาย ๆ อย่างโดยอัลกอริธึมเลือกเส้นทางเดินข้อมูลแบบโครงสร้างลำดับชั้น [5] จะแบ่งเราท์เตอร์ในระบบทั้งหมดออกเป็นกลุ่มเล็ก ๆ เรียกว่า "ตำบล (regions)" แต่ละเราท์เตอร์จึงเปรียบเหมือนหมู่บ้านที่อยู่ภายในตำบลเดียวกัน โดยมีคอมพิวเตอร์แต่ละโหนดเป็นเสมือนกับชาวบ้านที่อาศัยอยู่ในหมู่บ้าน เราท์เตอร์จึงทราบเส้นทางการส่งข้อมูลไปยังเราท์เตอร์ทุกตัวภายในตำบลเดียวกัน แต่จะไม่มีข้อมูลเกี่ยวกับเราท์เตอร์ที่อยู่ในตำบลอื่น ในกรณีที่ระบบประกอบด้วยเครือข่ายต่างชนิดกันหลายระบบเครือข่ายแต่ละชนิดสมควรได้รับการกำหนดให้เป็นตำบล เพื่อที่เราท์เตอร์ในระบบต่างชนิดกันจะไม่ต้องเข้าไปศึกษาโครงสร้างการสื่อสารของเราท์เตอร์ที่อยู่ในเครือข่ายชนิดอื่น

ถ้าระบบเครือข่ายขนาดใหญ่มาก การกำหนดระบบเครือข่ายให้มีโครงสร้างแบบสองลำดับชั้นอาจไม่เพียงพอ ดังนั้นจึงควรรวมกลุ่มตำบลให้เป็นกลุ่มที่มีขนาดใหญ่ขึ้นตามลำดับ ซึ่งสามารถเปรียบเทียบได้คือ รวมกลุ่มตำบลเป็นอำเภอ รวมกลุ่มอำเภอเป็นจังหวัด รวมกลุ่มจังหวัดเป็นภาค และรวมกลุ่มภาคเป็นประเทศ เป็นต้น สมมุติว่าต้องการส่งข้อมูลจากโหนดหนึ่งที่อยู่ในหมู่บ้านหนึ่ง ในตำบลหนึ่ง ไปยังโหนดผู้รับที่อยู่อีกหมู่บ้านที่สอง ในตำบลที่สอง แพ็กเก็ตข้อมูลจะถูกส่งจากโหนดผู้ส่งไปยัง (ที่ทำการ) หมู่บ้านหนึ่ง ซึ่งจะส่งข้อมูลไปตามลำดับดังนี้ (ที่ว่าการ) ตำบลไปยัง (ที่ว่าการ) อำเภอ ต่อไปยัง ที่ว่าการจังหวัด ต่อไปยัง (ศาลาว่าการ) กรุงเทพมหานคร ต่อไปยัง (ที่ว่าการ) อำเภอต่อไปยัง (ที่ว่าการ) ตำบล ซึ่งจะส่งข้อมูลไปยังโหนดผู้รับที่อยู่ในหมู่บ้านสองในตำบลนั้น

ตัวอย่างของการจัดแบ่งกลุ่มข้อมูลออกเป็นสองระดับที่ประกอบด้วย ห้าตำบล หมู่บ้าน ถ้าจะเขียนตารางนำส่งข้อมูลในรูปแบบเดิมคือ แต่ละเราท์เตอร์จะต้องทราบข้อมูลของเราท์เตอร์อื่นทุกตัว แต่ละเราท์เตอร์ เช่น เราท์เตอร์ 1A จะต้องมีข้อมูลถึง 17 รายการดังแสดงในรูป 2.2 (b) เมื่อนำหลักโครงสร้างลำดับชั้นแบบสองระดับมาใช้ เราท์เตอร์ 1A จะมีรายการข้อมูลเหลือเพียง 7 รายการ

ดังแสดงในรูป 2.2 (c) เห็นได้ว่าเมื่อจำนวนเราท์เตอร์เพิ่มมากขึ้นเท่าใด วิธีนี้จะช่วยลดจำนวนรายการในตารางข้อมูลมากเท่านั้น



(a)

Full table for 1A

Dest.	line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	line	Hops
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

รูปที่ 2.2 การเลือกเส้นทางแบบโครงสร้างลำดับชั้น

2.2.1.2.3 การเลือกเส้นทางแบบฟลัดดิ้ง

การเลือกเส้นทางข้อมูลแบบฟลัดดิ้ง (Flooding) [6] เป็นวิธีการที่ไม่ต้องปรับตัวให้เข้ากับสภาพแวดล้อมของระบบ โหนดจะส่งทุกแพ็กเก็ตที่รับเข้ามาออกไปทุกทิศทุกทางที่มีการเชื่อมต่อกับโหนดตัวอื่นยกเว้นโหนดที่เป็นผู้ส่งเข้ามา วิธีนี้จะเพิ่มปริมาณข้อมูลในเครือข่ายขึ้นมาอย่างมากซึ่งส่วนมากเป็นข้อมูลที่ซ้ำกัน ถ้าไม่ใช้กรรมวิธีอื่นเข้ามาช่วยแล้วจะทำให้เกิดปริมาณทราฟฟิกขึ้นอย่างหาศาลเลยทีเดียว หนทางหนึ่งที่นำมาใช้คือการใส่ตัวเลขเพื่อกำหนดจำนวนโหนดสูงสุดที่แพ็กเก็ตสามารถเข้าไปในข้อมูลส่วนหัวของแต่ละแพ็กเก็ต ทุกครั้งที่แพ็กเก็ตเข้ามาก็จะตรวจสอบค่าตัวเลขนี้ ถ้าหากมีค่าเป็นศูนย์ก็จะลบแพ็กเก็ตนี้ทิ้ง มิฉะนั้นก็จะลดค่าจำนวนนับลงไปหนึ่งหน่วยก่อนที่จะส่งต่อไป เลขจำนวนนับโหนดนี้ควรจะมีค่าเท่ากับจำนวนโหนดที่จะต้องเดินทางผ่านจากผู้ส่งไปยังผู้รับ ถ้าผู้ส่งไม่ทราบจำนวนที่แน่นอนก็จะใช้ค่าสมมุติ ซึ่งเป็นค่าสูงสุดของเครือข่ายนั้น ๆ แทน

วิธีการฟลัดดิ้งเหมาะกับงานบางประเภท เช่น นำไปใช้ในกิจการทางทหาร ซึ่งในยามสงครามโหนดแต่ละโหนดอาจถูกข้าศึกทำลายได้ทุกเมื่อ ดังนั้นการที่มีสำเนาข้อมูลค้างอยู่ในระบบจำนวนมากจะทำให้มีความมั่นใจได้ว่าข่าวสารจะเดินทางไปถึงผู้รับ และในระบบฐานข้อมูลแบบกระจายนั้นสามารถใช้วิธีฟลัดดิ้งเข้ามาช่วยในการปรับปรุงข้อมูลในทุกฐานข้อมูลพร้อมกัน นอกจากนั้นคือการใช้ฟลัดดิ้งเพื่อการเปรียบเทียบอัลกอริธึมแบบต่าง ๆ เนื่องจากวิธีฟลัดดิ้งจะส่งข้อมูลออกไปทุกทิศทาง ทำให้หาเส้นทางที่สั้นที่สุดได้เสมอ ยิ่งกว่านั้นไม่มีอัลกอริธึมใดที่สามารถส่งข้อมูลโดยใช้เวลาการรอคอยต่ำกว่านี้ได้

2.2.1.3 การกำหนดเส้นทางแบบสถิตย์ (Static Routing)

การเลือกเส้นทางแบบนี้ ผู้ดูแลระบบเป็นผู้พิจารณาและคำนวณหาเส้นทางทั้งหมดที่ต้องกำหนดให้เราท์เตอร์ทุกตัวในเครือข่ายที่ดูแลอยู่และป้อนตารางเส้นทางให้กับเราท์เตอร์ ตารางเลือกเส้นทางจะมีค่าตายตัวไปตลอดจนกว่าผู้ดูแลระบบจะเปลี่ยนแปลงใหม่ บางครั้งเรียกว่าเลือกเส้นทางแบบคำนวณไว้ล่วงหน้า ข้อดีของการเลือกเส้นทางแบบคำนวณไว้ล่วงหน้า คือ

1. สะดวกต่อการใช้งานกับเครือข่ายขนาดเล็ก
2. ไม่ต้องใช้ซอฟท์แวร์เลือกเส้นทางเราท์เตอร์ ไม่จำเป็นต้องมีซีพียูสมรรถนะสูง
3. ประหยัดแบนด์วิดธ์เครือข่าย เนื่องจากไม่ต้องแลกเปลี่ยนข้อมูลระหว่างเราท์เตอร์

การเลือกเส้นทางแบบคำนวณไว้ล่วงหน้านิยมใช้กับการเชื่อมโยงแบบจุดต่อจุดระหว่างเราท์เตอร์สองตัว ตัวอย่างเครือข่ายที่มีทางออกไปสู่ภายนอก หรืออินเตอร์เน็ตเพียงช่องทางเดียว การกำหนดเส้นทางจะเป็นแบบคำนวณไว้ล่วงหน้า ถึงแม้ว่าการเลือกเส้นทางแบบคำนวณไว้ล่วงหน้าจะมีข้อดีหลายประการ แต่ก็มีข้อเสีย คือ

1. ไม่เหมาะกับเครือข่ายขนาดใหญ่ ผู้ดูแลสามารถคำนวณหาเส้นทางในเครือข่ายที่มีขนาดเล็กได้ไม่ยากนัก แต่ในเครือข่ายขนาดใหญ่มีเราท์เตอร์เป็นจำนวนมากการคำนวณและป้อนค่าเข้าสู่เราท์เตอร์โดยตรงเป็นสิ่งที่เกินวิสัย
2. ไม่สะดวกต่อการเปลี่ยนโทโปโลยี เพราะต้องคำนวณและป้อนเส้นทางใหม่
3. ตารางเส้นทางคงตัวและไม่สามารถเปลี่ยนแปลงเองได้ หากเส้นทางใดถูกตัดขาด
4. ผู้ดูแลต้องคอยตรวจสอบและแก้ไขปัญหาเอง

2.2.1.4 อัลกอริธึมการเลือกเส้นทางแบบพลวัตร (Dynamic Routing Algorithms)

การเลือกเส้นทางแบบใช้ซอฟท์แวร์ทำหน้าที่แลกเปลี่ยนข้อมูลการเลือกเส้นทางระหว่างเราท์เตอร์ด้วยกันโดยใช้ โปรโตคอลเลือกเส้นทาง (Routing Protocol) เราท์เตอร์จะสร้างตารางเลือกเส้นทางจากสภาพเครือข่ายขณะนั้น หากเครือข่ายเปลี่ยนแปลงไป ตารางเลือกเส้นทางจะเปลี่ยนไปด้วย บางครั้งเรียกว่าการเลือกเส้นทางแบบขึ้นอยู่กับความต้องการ ข้อดีของการเลือกเส้นทางแบบขึ้นอยู่กับความต้องการคือ

1. รองรับเครือข่ายที่ขยายขึ้นเป็นลำดับได้

2. ตารางเส้นทางเปลี่ยนแปลงค่าเองตามการทำงานของซอฟท์แวร์เส้นทางใดที่ถูกตัดขาดจะมีการหาเส้นทางใหม่ทดแทน

การเลือกเส้นทางแบบนี้ขึ้นอยู่กับความต้องการนี้ ต้องอาศัยการแลกเปลี่ยนเส้นทางระหว่างเราท์เตอร์ และใช้ซีพียูในเราท์เตอร์เพื่อสร้างตารางเส้นทาง เราท์เตอร์ประเภทนี้จึงมักมีราคาสูงกว่าเราเตอร์มีตารางเส้นทางแบบสถิตย์เพียงอย่างเดียว เพราะต้องออกแบบซอฟท์แวร์ทำงานตามโปรโตคอลเลือกเส้นทาง และต้องมีซีพียูสมรรถนะสูงเพียงพอในการคำนวณ ตารางเส้นทางสำหรับอัลกอริธึมแบบพลวัตรเป็นการเลือกเส้นทางที่ค้นหาเส้นทางทุกครั้งที่มีการร้องขอการเชื่อมต่อเซสชั่นมาถึง สำหรับอัลกอริธึมการคำนวณ เป็นการเลือกเส้นทางที่มีการค้นหา (การคำนวณไว้ล่วงหน้า) และข้อมูลในตารางเส้นทางจะมีการปรับปรุง เมื่อมีการร้องขอเส้นทางหรือมีการแลกเปลี่ยนข้อมูลเส้นทาง โดยเส้นทางที่ใช้การได้หนึ่งเส้นทางจะถูกเลือกและติดตั้งการเชื่อมต่อเส้นทางเนื่องจากแต่ละการร้องขอมีข้อกำหนดเกี่ยวกับทราฟฟิกพารามิเตอร์และความต้องการรับประกันคุณภาพของการบริการที่แตกต่างกัน อัลกอริธึมขึ้นอยู่กับความต้องการสามารถหาเส้นทางที่เหมาะสมได้อย่างถูกต้องกับความต้องการ ได้ดีกว่าการใช้ตารางเส้นทางแบบสถิตย์

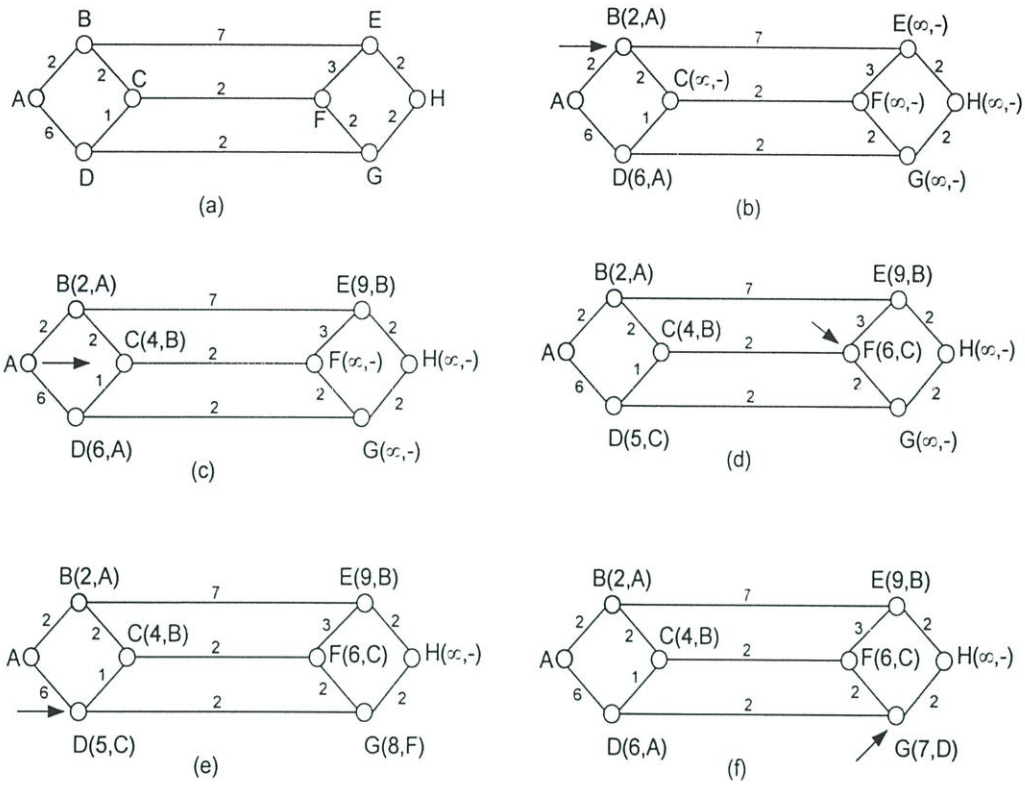
2.2.1.4.1 การเลือกเส้นทางที่สั้นที่สุด

การเลือกเส้นทางที่สั้นที่สุด เป็นวิธีการที่ถูกนำไปใช้มากที่สุดแบบหนึ่งหลักการทำงานเริ่มต้นด้วยการสร้างรูปกราฟของระบบเครือข่ายย่อย โดยให้แต่ละโหนดในรูปกราฟแทนเราท์เตอร์แต่ละตัวในเครือข่าย และให้เส้นเชื่อมโยงโหนดแทนสายสื่อสารที่เชื่อมต่อระหว่างเราท์เตอร์ การเลือกเส้นทางเดินระหว่างเราท์เตอร์คู่หนึ่งทำได้โดยการค้นหาเส้นทางที่สั้นที่สุดในรูปกราฟ

นิยามของคำว่าเส้นทางที่สั้นที่สุดอาจมีได้หลายความหมาย เช่น การใช้จำนวนครั้งของการรับส่งข้อมูลหรือจำนวนเส้นเชื่อมโยงในรูปกราฟเป็นหลักพิจารณา เส้นทาง ABCF และ ABEF ในรูปที่ 2.3 จะถือว่ายาวเท่ากัน แต่ถ้าใช้ระยะทาง (เช่น เป็นกิโลเมตร) มาพิจารณาแล้วเส้นทาง ABEF ย่อมสั้นกว่า

อีกวิธีการหนึ่งที่สามารถนำมาใช้แทนการนับจำนวนเส้นทางเชื่อมหรือการวัดระยะทาง คือการกำหนดให้ตัวเลขบนเส้นเชื่อมแต่ละเส้นเป็นตัวเลขที่บอกจำนวนเพ็กเกตที่รอการจัดส่ง และเวลารอคอยโดยเฉลี่ยที่คำนวณมาจากการรับ-ส่งแพ็กเก็ตมาตรฐาน ด้วยวิธีการนี้เส้นทางที่สั้นที่สุดจึงหมายถึงเส้นทางที่ส่งข้อมูลได้เร็วที่สุด

วิธีการที่นำมาใช้กับกรณีทั่วไปนั้นจะกำหนดให้ตัวเลขบนเส้นเชื่อมโยงคือ ผลลัพธ์ที่ได้จากการคำนวณโดยมีระยะทาง ความเร็วในการส่งข้อมูล ปริมาณข้อมูลโดยเฉลี่ย ค่าใช้จ่าย ค่าเฉลี่ยมาตราฐานของจำนวนแพ็กเก็ตที่รอการจัดส่ง ระยะเวลารอคอย และอื่น ๆ เป็นตัวประกอบการเปลี่ยนค่าความสำคัญของตัวประกอบเหล่านี้ ทำให้แต่ละเราท์เตอร์สามารถใช้อัลกอริธึมเดียวกันในการคำนวณ แต่ให้ความสำคัญกับตัวประกอบไม่เหมือนกันได้



รูปที่ 2.3 ขั้นตอนในการคำนวณเส้นทางเดินที่สั้นที่สุด

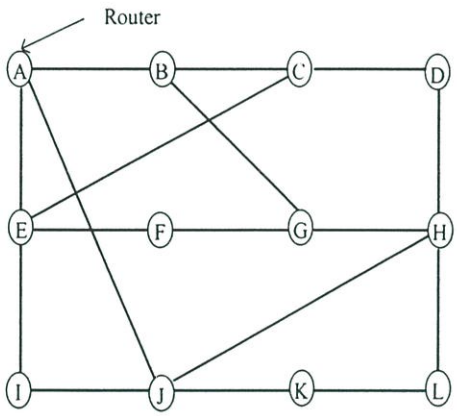
2.2.1.4.2 การเลือกเส้นทางแบบตารางระยะทาง

อัลกอริธึมเลือกเส้นทางระยะทางและเส้นทางแบบตารางระยะทาง (Distance vector routing) เป็นอัลกอริธึมพลวัตรแบบหนึ่งที่ได้รับความนิยมในการนำไปใช้งานเป็นอย่างมาก อัลกอริธึมนี้ยังเป็นที่รู้จักในนาม อัลกอริธึม Bellman-Ford

เราท์เตอร์ใช้อัลกอริธึมเลือกเส้นทางแบบตารางระยะทางจะต้องสร้างตารางเก็บข้อมูล ซึ่งจะบอกระยะทางและเส้นทางที่ดีที่สุดในการส่งข้อมูลไปยังเราท์เตอร์ถัดไป ข้อมูลในตารางนี้จะมีการปรับปรุงอยู่เสมอโดยการแลกเปลี่ยนข่าวสารกับเราท์เตอร์ที่อยู่ติดกัน จำนวนข้อมูลในตารางจะเท่ากับจำนวนเราท์เตอร์ทั้งหมดในเครือข่าย (ไม่นับตัวเอง) ข้อมูลแต่ละตัวประกอบด้วยเส้นทางสำหรับส่งข้อมูลไปยังจุดหมายและระยะทางหรือเวลาโดยประมาณที่จะต้องใช้เส้นทางนั้น มาตรที่ใช้วัดได้แก่ จำนวนเราท์เตอร์ที่อยู่ในเส้นทาง เวลารอคอยตลอดเส้นทาง จำนวนแพ็กเก็ตที่รอการนำส่ง เป็นต้น

สมมุติว่าใช้เวลารอคอยเป็นมาตรในการวัดระยะทาง ในทุกอัลกอริธึม T มิลลิวินาที แต่ละเราท์เตอร์จะต้องส่งตารางข้อมูลที่ตนเองมีอยู่ไปให้เราท์เตอร์ข้างเคียงทุกตัว ในทางกลับกันเราท์เตอร์แต่ละตัวจะได้รับตารางข้อมูลจากเราท์เตอร์ข้างเคียงข้อมูลได้รับจากเราท์เตอร์ X จะบอกให้ทราบว่าจะ

ต้องใช้เวลา Xi เพื่อส่งข้อมูลไปยังเราท์เตอร์ I ถ้าเวลาที่ใช้ส่งข้อมูลไปยังเราท์เตอร์ X เป็น m มิลลิวินาที ดังนั้นเวลาที่จะต้องใช้ในการส่งข้อมูลไปยังเราท์เตอร์ I จะเป็น Xi + m มิลลิวินาที โดยการส่งข้อมูลผ่านเราท์เตอร์ X เมื่อใช้วิธีการนี้สำหรับทุกตารางที่ได้รับ เราท์เตอร์แต่ละตัวก็จะทราบเวลาที่จะต้องใช้ในการส่งข้อมูลไปยังเราท์เตอร์ทุกตัวในระบบ ดังนั้นจึงสามารถเลือกเส้นทางที่ใช้ในระยะเวลาน้อยที่สุดในการส่งข้อมูลไปยังจุดหมายที่ต้องการได้ ซึ่งตัวเลขเหล่านี้ก็จะถูกนำไปเก็บไว้ในตารางของตนเอง สังเกตว่าข้อมูลเก่าในตารางไม่ได้นำมาใช้ในการคำนวณเลย



(a)

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	127	10	0	—
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6		

Vectors received from J's four neighbores

New routing table for J

(b)

รูปที่ 2.4 (a) เครือข่ายย่อย

(b) ข้อมูลใหม่จาก A,I,H,K และตารางทางเดินข้อมูลของ J

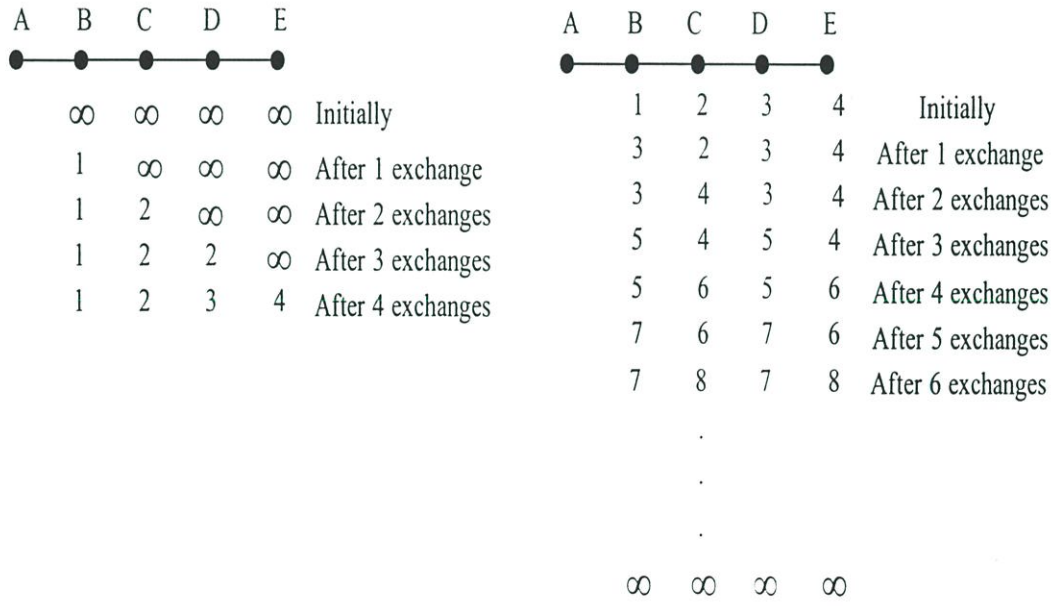
กระบวนการคำนวณระยะทางนี้แสดงให้เห็นในรูปที่ 2.4 (a) รูปโครงสร้างเครือข่ายย่อยตารางสี่อันดับแรกในรูปที่ 2.4 (b) เป็นตารางแสดงระยะเวลาที่ใช้ของเราท์เตอร์ A, I, H และ K ที่ส่งมาให้เราท์เตอร์ J ตารางแรกระบุว่าเราท์เตอร์ A ใช้เวลา 12 มิลลิวินาทีในการส่งข้อมูลไปยังเราท์เตอร์ B

และใช้เวลา 25 มิลลิวินาทีในการส่งข้อมูลไปยังเราท์เตอร์ C ฯลฯ และให้เวลาในการส่งข้อมูลจาก J ไปยังเราท์เตอร์ A, I, H และ K เป็น 8, 10, 12 และ 6 มิลลิวินาที ตามลำดับ

ถ้าเราท์เตอร์ J ต้องการคำนวณเวลาส่งข้อมูลไปเราท์เตอร์ G เส้นทางแรกจะต้องส่งข้อมูลผ่านเราท์เตอร์ A การส่งข้อมูลจาก J ไปยัง A ใช้เวลา 8 มิลลิวินาที และจาก A ไปยัง G ใช้เวลา 18 มิลลิวินาที เวลารวมเป็น 26 มิลลิวินาที ในทำนองเดียวกันเมื่อส่งข้อมูลผ่าน I, H, และ K จะใช้เวลารวม 41 (31+10), 18 (6+12), และ 37(31+6) มิลลิวินาที ตามลำดับ เวลาที่ดีที่สุดคือ 18 มิลลิวินาที ดังนั้นเราท์เตอร์ J จะบันทึกข้อมูลในตารางของตนเองว่า การส่งข้อมูลไปยังเราท์เตอร์ G ใช้เวลา 18 มิลลิวินาที โดยการส่งข้อมูลผ่านเราท์เตอร์ H ผลจากการคำนวณดังที่แสดงไว้ในตารางสุดท้าย

ปัญหาการปรับปรุงตารางข้อมูล

การเลือกเส้นทางแบบตารางระยะทางมีความเป็นไปได้ในทางทฤษฎีแต่ในทางปฏิบัติแล้วไม่สามารถนำไปใช้ได้ เพราะแม้ว่าวิธีการนี้จะช่วยในการหาคำตอบได้ในที่สุดแต่ก็ใช้เวลานานมาก อัลกอริธึมนี้สามารถตอบสนองต่อการเปลี่ยนแปลงทางด้านดีได้อย่างรวดเร็ว เช่น สมมุติให้มีเส้นทางหนึ่งส่งข้อมูลไปยังเราท์เตอร์ K มีระยะเวลานานมาก เมื่อได้รับข้อมูลใหม่โดยอาจพบเส้นทางที่มีระยะทางสั้นกว่าเดิม เส้นทางใหม่ก็จะถูกนำไปใช้แทนเส้นทางเดิมในทันที แต่เหตุการณ์เช่นนี้ไม่เกิดขึ้นถ้าเป็นการเปลี่ยนแปลงในทางตรงกันข้าม ซึ่งเรียกว่า ปัญหา "Count-to-infinity"



รูปที่ 2.5 กรณีศึกษาของปัญหา Count-to-infinity

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.1.4.3 โพรโตคอลเลือกทางเดินของเกตเวย์ภายในเครือข่ายแบบ OSPF

ในระบบเครือข่ายอินเทอร์เน็ตประกอบด้วยระบบเครือข่ายอัตโนมัติ (AS: autonomous systems) จำนวนมาก แต่ละองค์กรจะบริหารเครือข่าย AS ด้วยนโยบายของตนเอง และใช้อัลกอริธึมในการค้นหาเส้นทางเดินข้อมูลภายในที่แตกต่างกัน ตัวอย่างเช่น ถ้าเครือข่ายภายในของบริษัท 3 แห่ง (X , Y และ Z) ทำการเชื่อมต่อกันระบบอินเทอร์เน็ต ระบบทั้ง 3 นี้จะถูกมองว่าเป็นระบบเครือข่ายอัตโนมัติ ซึ่งอาจใช้อัลกอริธึมในการหาเส้นทางเดินข้อมูลภายในระบบที่แตกต่างกันได้ อย่างไรก็ตามในการกำหนดมาตรฐานสำหรับการหาเส้นทางภายในทำให้การเชื่อมต่อระหว่างเขตติดต่อของระบบเครือข่ายอัตโนมัติเป็นไปได้ง่ายขึ้น

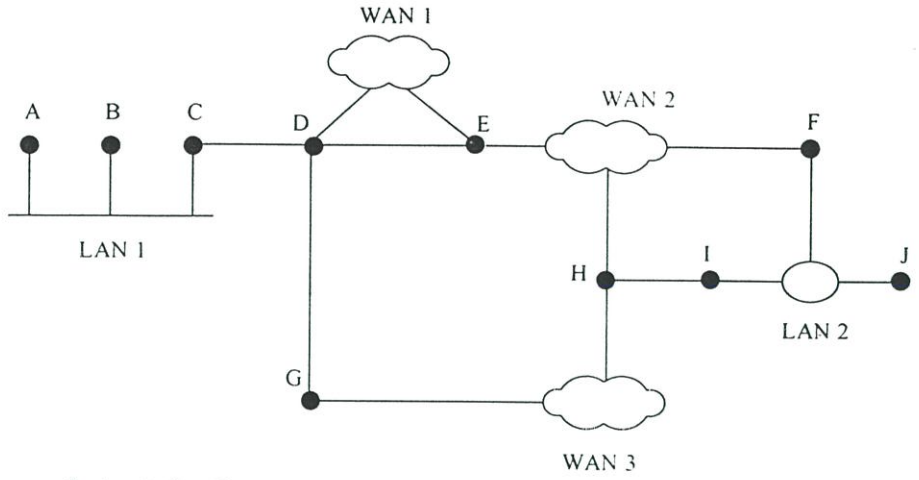
ในระยะเริ่มแรกนั้น โพรโตคอลสำหรับเกตเวย์ในระบบเครือข่ายก็คือโพรโตคอลเลือกเส้นทางแบบตารางระยะทาง (RIP) ที่มีพื้นฐานมาจากอัลกอริธึมของ Bellman-Ford วิธีการนี้สามารถทำงานได้ดีในระบบขนาดเล็ก แต่ประสิทธิภาพจะลดลงเมื่อระบบมีขนาดใหญ่ขึ้น รวมทั้งมีปัญหาที่เรียกว่า "Count-to-infinity" คณะทำงาน "The Internet Engineering Task Force" ได้ประสบความสำเร็จในการพัฒนาอัลกอริธึมที่ดีกว่ามาทดแทน เรียกว่า "OSPF (Open Shortest Path First [12]")

ระบบ OSPF สนับสนุนการติดต่อและเครือข่ายได้ 3 แบบ

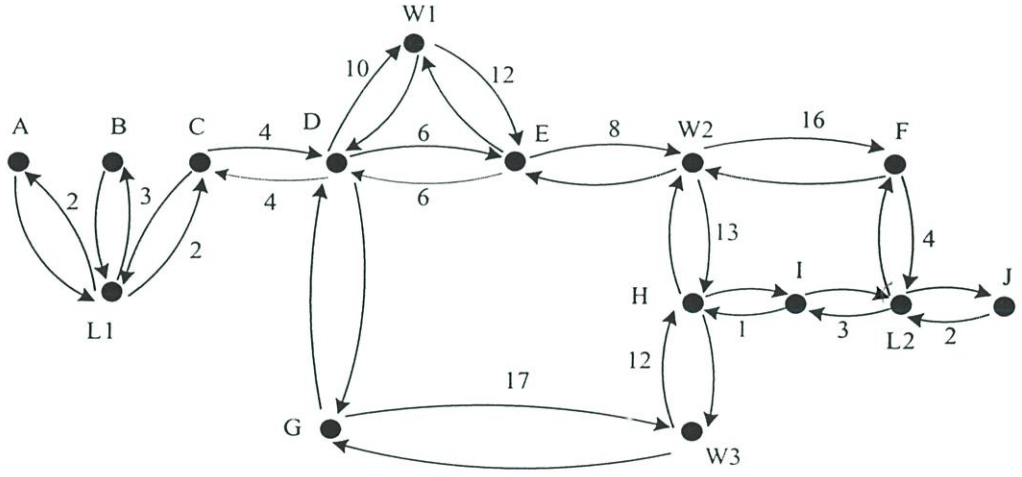
1. แบบจุด-ต่อ-จุด ซึ่งเป็นการติดต่อกันโดยตรงระหว่างเราท์เตอร์ 2 ตัว
2. แบบการสื่อสารข้อมูลหลายจุดที่สนับสนุนการกระจายข่าวส่วนใหญ่เป็นระบบเครือข่ายเฉพาะบริเวณ
3. แบบการสื่อสารข้อมูลหลายจุดที่ไม่สนับสนุนการกระจายข่าว เช่น ระบบแพ็กเก็ตสวิตช์ที่ใช้ในเครือข่ายวงกว้าง

OSPF ทำงานโดยรวบรวม เครือข่ายที่มีอยู่ เราท์เตอร์ และสายสื่อสารเข้าไว้ด้วยกันในลักษณะของรูปกราฟแบบมีทิศทาง (directed graph) โดยกำหนดให้เส้นเชื่อมโหนด (arc) ของกราฟแต่ละเส้นมีมูลค่า หรือน้ำหนัก (weight) ที่คำนวณมาจาก ระยะทาง เวลารอคอย และองค์ประกอบอื่นที่ต้องการ จากนั้นจึงทำการคำนวณหาเส้นทางที่สั้นที่สุด (shortest path) โดยพิจารณาจากมูลค่าที่กำหนดไว้ให้แต่ละเส้นกราฟนั้น การเชื่อมต่อแบบอนุกรมระหว่างเราท์เตอร์สองตัว จะกำหนดโดยใช้เส้นเชื่อมโหนดคู่หนึ่ง แต่ละเส้นใช้แทนทิศทางการไหลของข้อมูลในแต่ละทิศทางซึ่งอาจจะมีน้ำหนักต่างกันดังนั้น เครือข่ายแบบสื่อสารข้อมูลหลายจุดจะแทนด้วยโหนดหนึ่งโหนดที่ใช้แทนเครือข่าย และโหนดอีกจำนวนหนึ่งที่ใช้แทนเราท์เตอร์แต่ละตัว เส้นเชื่อมระหว่างโหนดเครือข่ายไปยังโหนดเราท์เตอร์จะมีน้ำหนักเป็นศูนย์

48920



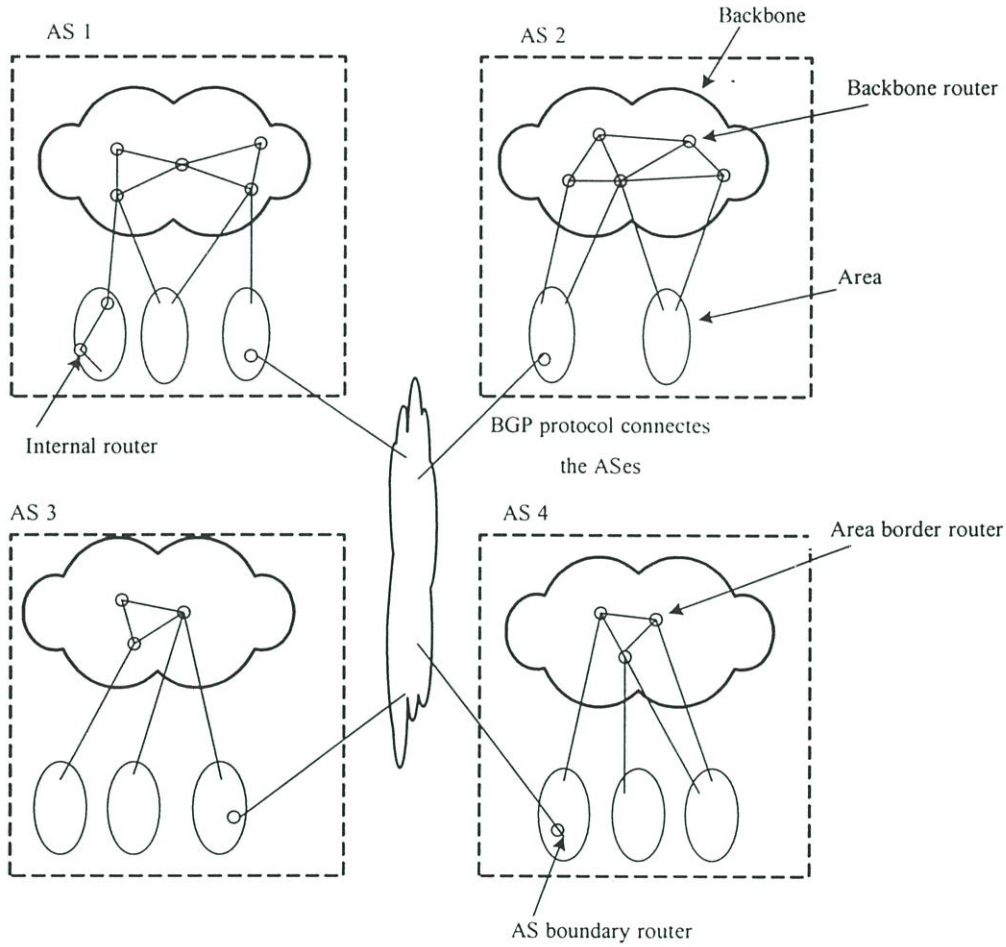
รูปที่ 2.6 ระบบเครือข่ายอัตโนมัติ



รูปที่ 2.7 กราฟที่ใช้แทนระบบเครือข่ายอัตโนมัติ

อัลกอริธึม OSPF มีวิธีบริหารประเภทการให้บริการโดยสร้างกราฟขึ้นมาหลายรูป รูปที่หนึ่งใช้น้ำหนักหรือมูลค่ากำกับสำหรับมาตรวัดเวลารอคอย รูปที่สองใช้กับมาตรวัดปริมาณผลลัพธ์ รูปที่สามใช้กับมาตรวัดระดับความน่าเชื่อถือของระบบเป็นต้น แม้ว่าการแยกประเภทนี้จะเพิ่มปริมาณงานในการคำนวณขึ้นหลายเท่า แต่ก็ให้ผลดีในการใช้งานระบบให้ตรงกับวัตถุประสงค์ที่ต้องการอย่างแท้จริง ในการทำงานตามปกติระบบเครือข่ายต้องการใช้เส้นทางเดินข้อมูลสามประเภท คือ เส้นทางภายในเขตย่อย เส้นทางเดินระหว่างเขตย่อย และเส้นทางเดินระหว่างเครือข่ายอัตโนมัติ เส้นทางภายในเขตย่อยนั้นคำนวณหาได้ง่ายที่สุด เนื่องจากเราท์เตอร์ผู้ส่งข้อมูลจะทราบเส้นทางที่สั้นที่สุดเพื่อไปยังจุดหมายที่ต้องการอยู่แล้วเส้นทางเดินระหว่างเขตย่อยจะมีการทำงานอยู่สามขั้นตอนเสมอ คือ หาเส้นทางจากผู้ส่งข้อมูลไปยังระบบสื่อสารหลัก ส่งข้อมูลผ่านระบบสื่อสารหลักไป

ยังเขตย่อยของผู้รับข้อมูล และส่งข้อมูลไปยังผู้รับข้อมูล กระบวนการนี้บังคับให้เกิดโครงสร้างการทำงานแบบรูปดาว โดยมีระบบสื่อสารหลักอยู่ที่ศูนย์กลางและเขตข้อมูลทั้งหมดเป็นโหนดอยู่ตามแฉกต่าง ๆ แพ็กเก็ตจะถูกส่งจากผู้ส่งไปยังผู้รับตามสภาพที่แท้จริงของแพ็กเก็ตนั้น ๆ โดยไม่มีการสร้างแพ็กเก็ตห่อหุ้มหรือใช้อุโมงค์สื่อสาร ยกเว้นเส้นทางส่งข้อมูลไปยังผู้รับนั้นเป็นแบบอุโมงค์สื่อสารอยู่แล้ว รูปดังต่อไปนี้แสดงโครงสร้างบางส่วนของระบบอินเทอร์เน็ตที่ประกอบด้วยระบบเครือข่ายอัตโนมัติและเขตเครือข่ายเขตย่อย



รูปที่ 2.8 ความสัมพันธ์ระหว่างระบบเครือข่ายอัตโนมัติ,ระบบสื่อสารหลัก และเขตย่อยใน OSPF

OSPF แบ่งประเภทของเราท์เตอร์ออกเป็นสี่ประเภท คือ

1. เราท์เตอร์ภายในเขตย่อยที่อยู่ในเขตย่อยเดียว
2. เราท์เตอร์ชายแดนที่เชื่อมต่อกับเขตย่อยตั้งแต่สองเขตขึ้นไป
3. เราท์เตอร์ในเขตระบบสื่อสารหลัก
4. เราท์เตอร์ที่เชื่อมต่อระหว่างระบบเครือข่ายอัตโนมัติ

เราท์เตอร์ทั้งสี่ประเภทนี้สามารถเป็นซับเซ็ทของกันและกันได้ เช่น เราท์เตอร์ชายแดนที่เชื่อมต่อกับเขตย่อยตั้งแต่สองเขตขึ้นไปนั้น จะอยู่ในประเภทเราท์เตอร์ในเขตระบบสื่อสารหลักด้วยเสมอ หรือเราท์เตอร์ในระบบสื่อสารหลักที่ไม่ได้ติดต่อกับเราท์เตอร์นอกเขต ก็จะเป็นเราท์เตอร์ภายในเขตย่อยเช่นกัน

OSPF ทำงานโดยการแลกเปลี่ยนข่าวสารระหว่างเราท์เตอร์ที่อยู่ติดกัน (adjacent routers) เพื่อหลีกเลี่ยงปัญหาการสื่อสารกับเราท์เตอร์ข้างเคียง (neighboring routers) ที่มีจำนวนค่อนข้างมาก เราท์เตอร์ตัวหนึ่งจะได้รับเลือกให้เป็น "ตัวแทนกลุ่มย่อย" ซึ่งจะอยู่ติดกับเราท์เตอร์ตัวอื่นทุกตัวในกลุ่ม และสามารถแลกเปลี่ยนข่าวสารระหว่างกันได้ เราท์เตอร์ข้างเคียงที่ไม่ได้อยู่ติดกันก็จะไม่มีการสื่อสารถึงกัน ตัวสำรองตัวแทนกลุ่มย่อยจะถูกกำหนดไว้เช่นกันเพื่อให้สามารถทำงานแทนได้ในกรณีที่ตัวแทนหลักชำรุด

2.3 เทคโนโลยีในการสลับแพ็คเก็ต (MPLS)

ในหลายปีผ่านมานี้ความตื่นตัวในการใช้งานระบบอินเทอร์เน็ตได้กระตุ้นให้มีการพัฒนาแอพพลิเคชั่นแบบใหม่สำหรับวงการธุรกิจและผู้ใช้หลากหลายรูปแบบ และแอพพลิเคชั่นใหม่ ๆ เหล่านี้ก็ได้ผลักดันให้ระบบเครือข่ายหลักของระบบอินเทอร์เน็ตต้องมีการปรับปรุง เพื่อสามารถรองรับความต้องการเฉพาะด้านของแต่ละแอพพลิเคชั่นได้ ทั้งในแง่ของความรวดเร็วและขนาดของแบนวิดท์ เพราะโครงสร้างหลักของระบบอินเทอร์เน็ตในปัจจุบันยังไม่สามารถที่จะให้บริการแบบแบ่งแยกได้ นอกเหนือจากความต้องการในการร้องขอบริการในแง่ของความรวดเร็วและขนาดของแบนวิดท์แล้ว โครงข่ายหลักก็ยังถูกคาดหวังให้สามารถรองรับความต้องการที่หลากหลายได้ (differentiated services) สำหรับผู้ใช้และปริมาณทราฟฟิกที่มีจำนวนเพิ่มมากขึ้นอย่างรวดเร็วจึงทำให้ CoS (Class of Services) และ QoS (Quality of Services) กลายเป็นประเด็นสำคัญในการพัฒนาเพื่อให้ระบบเครือข่ายสามารถรองรับความต้องการของผู้ใช้จำนวนมากที่มีความต้องการที่แตกต่างกันได้

โดยสรุปแล้ว MPLS [16] เป็นเทคโนโลยีที่คาดหวังว่าจะเป็นทางออกสำหรับการให้บริการต่อผู้ใช้ในวงกว้างโดยอาศัยเทคนิคของการหาเส้นทางและการสวิทชิ่ง เพื่อให้สามารถจัดตั้งเส้นทางและให้บริการได้อย่างสอดคล้องกับความต้องการของผู้ใช้ และมีความสามารถในการส่งถ่ายแพ็คเก็ตระหว่างระบบเครือข่ายที่มีอยู่ในปัจจุบัน ไปยังระบบเครือข่ายที่อาจจะถูกพัฒนาขึ้นใหม่ในอนาคตได้

2.3.1 การหาเส้นทางและการสวิทชิ่งแบบดั่งเดิม

ความต้องการในการสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ตในยุคแรกเริ่มนั้น เป็นเพียงความต้องการในการโอนย้ายไฟล์ข้อมูล (file transfer) และการเข้าถึงระยะไกล (telnet) ผ่านระบบเครือข่ายเท่านั้น ซึ่งการหาเส้นทางด้วยโปรแกรมอย่างง่ายผ่านเส้นทางบนการเชื่อมต่อแบบ T1/E1 หรือ T3/E3 ก็เพียงพอสามารถที่จะให้บริการสำหรับแอพพลิเคชั่นประเภทนี้ได้ แต่ต่อมาเมื่อความ

ต้องการในการสื่อสารสูงขึ้นทั้งในแง่ของความเร็วและแบนด์วิดท์ ทำให้เกิดการพัฒนาให้ฮาร์ดแวร์สามารถที่จะสวิทชิ่งข้อมูลในระดับของเฟรม (layer 2) หรือแพ็กเก็ตได้ (layer 3)

ซึ่งการพัฒนาให้เกิดการสวิทช์แพ็กเก็ตในระดับฮาร์ดแวร์นั้น ฮาร์ดแวร์จำเป็นที่จะต้องสามารถสวิทช์แพ็กเก็ตหรือเฟรมได้ในความเร็วระดับเดียวกันกับความเร็วการส่งแพ็กเก็ตบนสายสื่อสาร แต่พัฒนาการในด้านของฮาร์ดแวร์นี้สามารถเพิ่มขีดความสามารถในแง่ของความเร็วเพียงอย่างเดียว โดยยังไม่ได้มีการคำนึงถึงความต้องการของแอพพลิเคชั่นที่เป็นปลายทางของแพ็กเก็ตข้อมูล นอกจากนี้อัลกอริธึมในการหาเส้นทางปัจจุบันก็ให้ความสำคัญในแง่ของเส้นทางที่สั้นที่สุดเพียงประการเดียว ซึ่งโดยมากมักจะหมายถึงจำนวน hop ของเส้นทางนั้น ๆ แต่ไม่ได้คำนึงถึงปัจจัยอื่น ๆ ร่วมด้วย เช่น ระยะเวลารอคอยในแถวคิว ความแปรปรวนของระยะเวลารอคอย และความหนาแน่นของทราฟฟิกในระบบเครือข่าย ซึ่งก็เป็นผลให้ประสิทธิภาพโดยรวมของระบบเครือข่ายมักจะถูกใช้ประโยชน์ไปอย่างไม่ค่อยจะเต็มที่นัก และภาระในการปรับปรุงประสิทธิภาพโดยรวมก็จะขึ้นอยู่กับประสบการณ์และการตัดสินใจในการควบคุมทราฟฟิก (traffic engineering) ของผู้ดูแลระบบเครือข่ายเป็นหลัก

2.3.2 เทคโนโลยีในการสลับแพ็กเก็ต (Multi-Protocol Label Switch)

MPLS คือเทคโนโลยีในการสลับแพ็กเก็ตที่ถูกพัฒนาขึ้นโดย IETF (Internet Engineering Task Force) เพื่อกำหนดใช้เป็นกรอบสำหรับการพัฒนาระบบเครือข่ายหลักสำหรับรองรับการใช้งานระบบอินเทอร์เน็ต

2.3.2.1 เทคโนโลยีในการสลับแพ็กเก็ต MPLS

1. กำหนดกลไกในการจัดการกับทราฟฟิกโฟล์วในระดับต่าง ๆ ทั้งระหว่างฮาร์ด-แวร์ที่ต่างแพลทฟอร์มกัน และแม้กระทั่งการสื่อสารระหว่างแอพพลิเคชั่นที่แตกต่างกัน

2. ไม่ขึ้นอยู่กับการทำงานของโปรโตคอลในเลเยอร์ 2 หรือ เลเยอร์ 3

3. สามารถที่จะแม็ปหมายเลข IP ไปอยู่ในรูปแบบของเลเบิลที่มีขนาดคงที่ เพื่อระบบเครือข่ายที่เป็นทางผ่านสามารถที่จะส่งต่อแพ็กเก็ตนั้นต่อไปยังระบบเครือข่ายปลายทางได้โดยการใช้เลเบิลที่กำหนดให้

4. สามารถทำงานร่วมกับโปรโตคอลในระดับควบคุมเครือข่ายที่มีอยู่แล้วได้ เช่น RSVP (Resource Reservation Protocol) และ OSPF (Open Shortest Path First)

5. สามารถรองรับการสื่อสารจากระบบเครือข่าย IP, ATM และ Frame Relay ได้

ในระบบเครือข่าย MPLS การส่งผ่านแพ็กเก็ตข้อมูลนั้นจะส่งไปตามเส้นทางที่กำหนดด้วยเลเบิลหรือ LSP (Label-Switched Path) ซึ่งอาจจะเป็นเส้นทางที่กำหนดไว้ก่อนล่วงหน้า (control driven) หรือเป็นเส้นทางที่ถูกเลือกใช้เมื่อมีแพ็กเก็ตข้อมูลส่งมาถึงก็ได้ (data driven) เส้นทางนี้จะถูกกำหนดขึ้นด้วยการกำหนดค่าของเลเบิลและพอร์ตสื่อสารของสวิทชิ่งโหนดใด ๆ ที่เป็นส่วนหนึ่งของลิงค์ภายในเส้นทาง และค่าของเลเบิลที่กำหนดให้กับทราฟฟิกโฟล์วใด ๆ นี้ก็จะถูก

ประกาศให้กับโหนดก่อนหน้าทราบโดยการใช้โปรโตคอล LDP (Label Distribution Protocol) หรือโปรโตคอล RSVP หรือใช้การแทรกข่าวสารการกำหนดเลเบิลไว้ในเฟรม acknowledge (piggyback) ของโปรโตคอลหาเส้นทางอย่าง OSPF หรือ BGP ก็ได้

แพ็กเก็ตข้อมูลที่ถูกส่งผ่านเครือข่าย MPLS จะถูกห่อหุ้ม (encapsulate) และบรรจุค่าของเลเบิลไว้ในส่วนหัวใหม่ และค่าของเลเบิลนี้ก็จะถูกใช้ในการค้นหาเส้นทางภายในโหนดใด ๆ และค่าเลเบิลใหม่ที่จะถูกใช้สำหรับโหนดถัด ๆ ไป การดำเนินการในการส่งแพ็กเก็ตข้อมูลก็จะเป็นดังที่กล่าวมาไปจนตลอดเส้นทาง การสื่อสารข้อมูลภายในระบบเครือข่าย MPLS นั้นสามารถที่จะสวิทช์แพ็กเก็ตได้อย่างรวดเร็วเนื่องจากว่าค่าของเลเบิลนี้จะถูกพิจารณาโดยฮาร์ดแวร์และทำการสวิทช์แพ็กเก็ตได้ในระดับของฮาร์แวร์โดยไม่ต้องส่งผ่านไปให้แก่โปรแกรมสื่อสารในระดับของซอฟแวร์แต่อย่างใด

2.3.2.2 ประเภทของเราท์เตอร์อยู่ภายในเครือข่าย MPLS

เราท์เตอร์ในระบบเครือข่าย MPLS นั้นสามารถแบ่งออกได้เป็น 2 กลุ่มตามลักษณะการทำงาน คือ

1. LSR เป็นเราท์เตอร์ความเร็วสูงที่ทำงานอยู่ภายในเครือข่าย MPLS มีหน้าที่ในการสร้างเส้นทาง LSP และมีความสามารถในการใช้โปรโตคอลในการประกาศข้อมูลเลเบิล (label signaling protocol) และสามารถสวิทช์แพ็กเก็ตไปตามเส้นทางที่กำหนดไว้ได้อย่างรวดเร็ว

2. LER คือ LSR ที่มีตำแหน่งการทำงานอยู่ที่ขอบของระบบเครือข่าย MPLS ทั้งที่ปากทางขาเข้าและปากทางขาออกของระบบเครือข่าย มีหน้าที่ในการเชื่อมต่อการสื่อสารระหว่างระบบเครือข่ายอื่น ๆ เช่น frame relay, ATM, Ethernet และระบบเครือข่าย MPLS โดยการใช้โปรโตคอลในการประกาศข้อมูลเลเบิลที่เราท์เตอร์ปากทางขาเข้า (ingress router) และทำหน้าที่ส่งผ่านทราฟฟิกระหว่างกันเมื่อเส้นทาง LSP ได้ถูกสร้างขึ้นเรียบร้อยแล้ว หน้าที่สำคัญของ LER ก็คือการห่อหุ้มแพ็กเก็ตและกำหนดเลเบิลที่ปากทางขาเข้า และแกะห่อแพ็กเก็ตที่ปากทางขาออก (egress router)

2.3.3 การกำหนดกลุ่มของแพ็กเก็ต FEC (Forward Equivalence Class)

คือการกำหนดกลุ่มของแพ็กเก็ตที่มีความต้องการในการใช้บริการระบบเครือข่ายที่เหมือน ๆ กัน แพ็กเก็ตต่าง ๆ ที่อยู่ในกลุ่ม FEC เดียวกันนี้ก็จะได้รับการให้บริการที่เท่าเทียมกันไปตลอดเส้นทาง การกำหนดว่าแพ็กเก็ตไหนจะอยู่ในกลุ่ม FEC ใดนั้นจะเกิดขึ้นเพียงครั้งเดียวเมื่อแพ็กเก็ตนั้น ๆ เดินทางมาถึงเราท์เตอร์ปากทาง (LER) ของระบบเครือข่าย MPLS ซึ่งอาจจะเป็นเพียงการพิจารณาจากที่อยู่ต้นทางและที่อยู่ปลายทางของแพ็กเก็ตหรืออาจจะใช้ข้อมูลอื่น ๆ ร่วมด้วยก็ได้ และภายในระบบเครือข่าย MPLS แพ็กเก็ตจะถูกส่งต่อโดยพิจารณาเส้นทางจากตารางเส้นทางที่เรียกว่า LIB (Label Information Base) ซึ่งเป็นตารางที่บรรจุข้อมูลในการแม็ปปิ้งค่าของ FEC ไปเป็น Label และบ่งบอกหมายเลขของพอร์ตสื่อสารขาออกของโหนดนั้น ๆ สำหรับแพ็กเก็ตด้วย

2.3.4 เลเบิล และการกำหนดเลเบิล

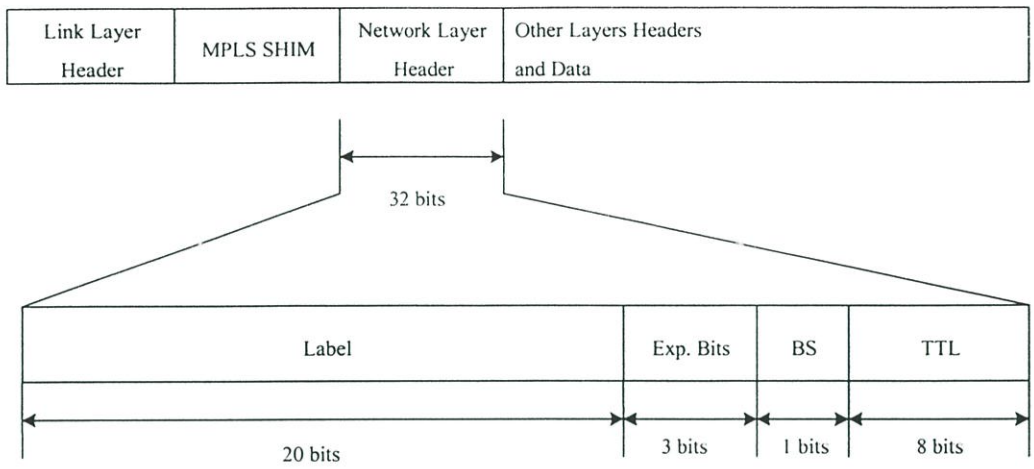
เลเบิลคือค่าใด ๆ (จำนวนเต็ม) ที่ถูกใช้ในการบ่งชี้เส้นทางของแพ็กเก็ต ค่าของเลเบิลนี้อาจจะบรรจุอยู่ในส่วนหัวของโปรโตคอลเลเยอร์2 หรือถูกห่อหุ้มไว้ภายในโปรโตคอลเลเยอร์2 เราท์เตอร์เมื่อได้รับแพ็กเก็ตมาก็จะพิจารณาค่าของเลเบิลนี้เพื่อนำไปค้นหาเส้นทางใน LIB เพื่อจะได้ทราบเส้นทางในการส่งไปยังโหนดถัดไป โดยค่าของเลเบิลนี้จะถูกกำหนดอย่างอิสระภายใต้โหนดใด ๆ (local significant) โดยไม่ขึ้นกับโหนดอื่น ๆ และมีความหมายต่อการสื่อสารระหว่าง 2 โหนดใด ๆ ที่อยู่ติดกันเท่านั้น (ลิงค์เดียว)

เมื่อแพ็กเก็ตใด ๆ ถูกจำแนกว่าเป็นกลุ่ม FEC เดิมที่มีเส้นทางอยู่แล้วค่าของเลเบิลที่ถูกกำหนดไว้ให้ใช้สำหรับ FEC กลุ่มนั้นก็จะถูกนำมาใช้กับแพ็กเก็ตนั้นทันที แต่ถ้าหากพบว่าเป็นแพ็กเก็ตของ FEC กลุ่มใหม่ก็จะมีการกำหนดค่าเลเบิลที่ว่างอยู่ให้แก่แพ็กเก็ตของ FEC กลุ่มใหม่นี้ โดยค่าของเลเบิลนี้ MPLS สามารถเลือกใช้ค่าบ่งชี้เส้นทาง (path identifier) ของโปรโตคอลเลเยอร์ 2 ที่ให้บริการการสื่อสารแบบต่อเนื่อง (connection oriented) ได้โดยตรง เช่น กรณีที่สื่อสารกับระบบเครือข่าย frame relay (DLCI : Data Link Connection Identifier) หรือ ระบบเครือข่าย ATM (VPI / VCI : Virtual Path Identifier / Virtual Channel Identifier)

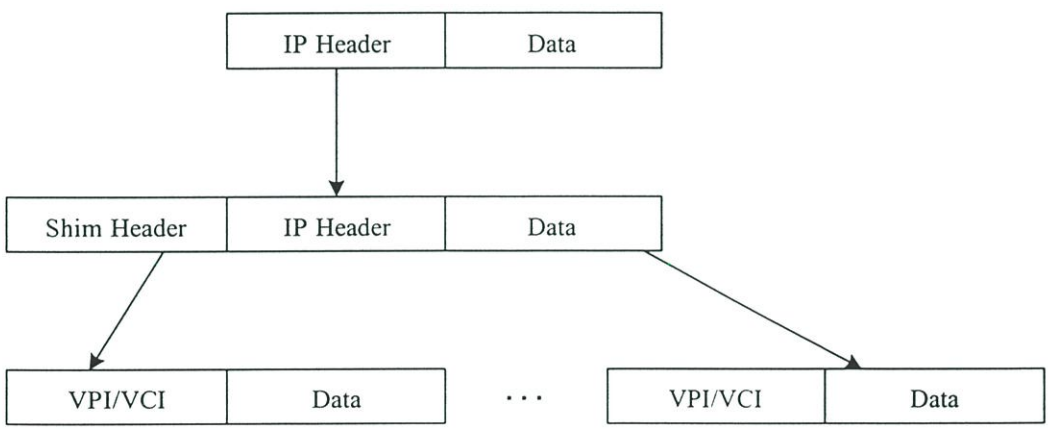
การกำหนดค่าของเลเบิลให้กับ FEC ใด ๆ หรือกลุ่มของ FEC ใด ๆ นั้น ขึ้นอยู่กับนโยบายในการให้บริการของระบบเครือข่าย MPLS ที่ถูกกำหนดไว้ ซึ่งอาจจะมีการกำหนดค่าของเลเบิลให้กับ FEC แบบ data driven หรือ แบบ control driven ก็ได้ ซึ่งการกำหนดค่าของเลเบิลในรูปแบบหลังน่าจะเหมาะสมแก่การนำมาใช้งานจริงมากกว่าเนื่องจากมีข้อดีอยู่หลายประการ นอกจากนี้การกำหนดค่าของเลเบิลยังขึ้นอยู่กับปัจจัยอื่น ๆ อีกดังต่อไปนี้

1. การสื่อสารแบบจุดต่อจุด (unicast) หรือแบบจุดไปยังหลายจุด (multicast)
2. การควบคุมการจราจรของทราฟฟิก
3. ความต้องการในการสร้างการเชื่อมต่อแบบส่วนตัว (VPN : Virtual Private Network)
4. ระดับของความต้องการในการใช้บริการ QoS

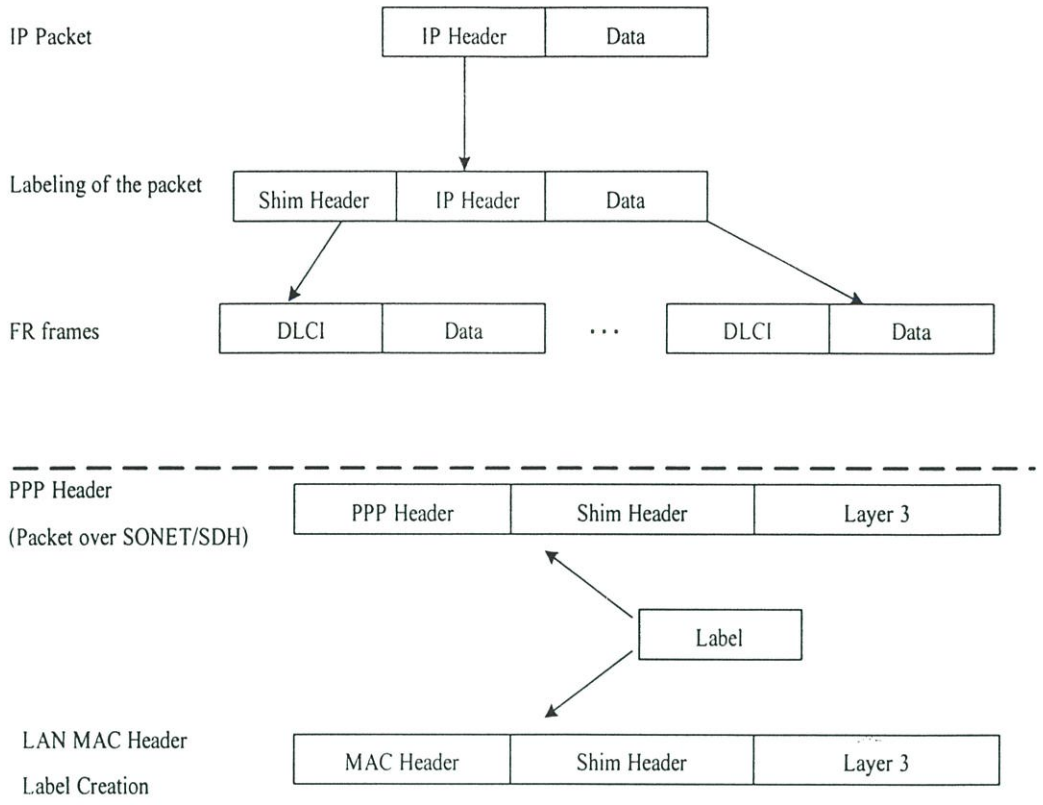
รูปแบบทั่ว ๆ ไปของเลเบิลนั้นแสดงไว้ดังรูปที่ 2.9 โดยค่าของเลเบิลอาจจะถูกบรรจุอยู่ภายในส่วนหัวของโปรโตคอลเลเยอร์ 2 (embeded) ดังรูปที่ 2.10 และ 2.11 หรือถูกแทรกอยู่ระหว่างส่วนหัวของโปรโตคอลเลเยอร์ 2 และส่วนหัวของโปรโตคอลเลเยอร์ 3 (shim) ดังรูปที่ 2.11



รูปที่ 2.9 รูปแบบเลเบิลของ MPLS



รูปที่ 2.10 ATM กับชั้นเชื่อมต่อข้อมูล



รูปที่ 2.11 Frame Relay กับชั้นเชื่อมต่อข้อมูล

2.3.5 การสร้างเลเบิล

การสร้างเลเบิลนั้นมีอยู่หลายวิธีเช่น

1. Topology-based method เป็นวิธีกระทำร่วมกับการหาเส้นทางโดยอาศัยโปรโตคอลหาเส้นทาง เช่น OSPF และ BGP

2. Request-based method ใช้การสร้างเส้นทางแบบจับจองทรัพยากรเช่นเดียวกับโปรโตคอล RSVP

3. Traffic-based method เป็นการกำหนดเส้นทางและเลเบิลพร้อมทั้งประกาศเลเบิลไปยังโหนดก่อนหน้าโดยทันทีเมื่อได้รับแพ็กเก็ตข้อมูล

สองวิธีแรกนั้นเป็นตัวอย่างของวิธีแบบ control driven และวิธีสุดท้ายนั้นเป็นรูปแบบของการกำ หนดเลเบิลแบบ data driven

2.3.6 การประกาศเลเบิล (Label Distribution)

มาตรฐานที่ถูกกำหนดโดย IETF นั้นไม่ได้กำหนดตายตัวว่าการประกาศเลเบิลสำหรับทราฟฟิกโฟล์วใด ๆ จะต้องใช้โปรโตคอลใด หรือวิธีการใด ปัจจุบัน BGP และ RSVP ต่างก็ได้ถูกแก้ไขปรับปรุงให้สามารถรองรับการประกาศค่าเลเบิลในทิศทางที่ตรงกันข้ามกับทิศทางที่สร้างข้อมูล โดย

การประกาศเลเบิลย้อนกลับมายังโหนดก่อนหน้าที่อยู่ติดกันแทรกไว้ในแพ็กเก็ต acknowledge นอกเหนือจากนี้แล้ว IETF เองก็ได้พัฒนาโปรโตคอลสำหรับการนี้โดยเฉพาะที่ถูกเรียกว่า LDP ไว้เป็นทางเลือกอีกทางหนึ่ง และ CR-LDP ก็เป็นโปรโตคอล LDP ที่ถูกพัฒนาขึ้นไปอีกเพื่อให้สามารถรองรับการกำหนดเส้นทางเฉพาะ (explicit routing) สำหรับรองรับความต้องการใช้บริการที่แตกต่างกันในลักษณะของการควบคุมแบบ QoS หรือแบบ CoS ได้อีกด้วย

ดังนั้นสามารถสรุปรูปแบบต่าง ๆ ที่ใช้ในการประกาศเลเบิลในระบบเครือข่าย MPLS ที่ถูกนำเสนอไว้ในปัจจุบันได้ดังนี้คือ

1. LDP ใช้ในการแม็ปหมายเลข unicast IP เป็นเลเบิล

2. RSVP, CR-LDP ใช้ในการสำรองทรัพยากรในระหว่างการสร้างเส้นทางเพื่อให้ได้เส้นทางที่สามารถให้บริการได้ตรงกับความต้องการของผู้ใช้

3. Protocol-independent multicast (PIM) ใช้สำหรับการแม็ปที่อยู่แบบหลายจุด (multicast address) ไปเป็นหนึ่งเลเบิลหรือหลาย ๆ เลเบิล

4. BGP เป็นการกำหนดเลเบิลที่ใช้ระหว่างระบบเครือข่าย 2 ระบบเครือข่าย (VPN)

2.3.7 การสลับเส้นทางด้วยเลเบิล Label-Switched Path (LSP)

ภายในระบบเครือข่าย MPLS ใด ๆ เส้นทาง LSP เส้นทางจะถูกสร้างขึ้นสำหรับ FEC ใด ๆ เส้นทาง LSP นี้จะต้องถูกสร้างขึ้นก่อนที่การสื่อสารข้อมูลจะเกิดขึ้นได้ โดยการสร้างเส้นทาง LSP นั้นสามารถสร้างขึ้นได้ตาม 2 แนวทางที่จะอธิบายต่อไปนี้

1. Hop-by-hop routing [11] เป็นวิธีการกำหนดเส้นทางแบบกระจายที่เราท์เตอร์ LSR แต่ละตัวจะเลือกเส้นทางสำหรับ FEC ใด ๆ อย่างอิสระโดยอาศัยข้อมูลภายในตารางเส้นทางที่มีอยู่ ณ ขณะนั้น ซึ่งถูกสร้างขึ้นโดยโปรโตคอลหาเส้นทางอย่าง OSPF, PNNI (Private Network-to-Network Interface ของ ATM) ซึ่งเป็นการทำงานที่คล้ายคลึงกับการหาเส้นทางที่ใช้กันอยู่ในระบบเครือข่ายอินเทอร์เน็ตในปัจจุบัน

2. Explicit Routing เป็นวิธีการกำหนดเส้นทางแบบมีศูนย์กลางลักษณะเดียวกันกับการหาเส้นทางแบบกำหนดเส้นทางครั้งเดียว (source routing) โดยกำหนดให้เราท์เตอร์ปากทางขาเข้า (LER) ทำหน้าที่กำหนดเส้นทางภายในระบบเครือข่าย MPLS ที่แพ็กเก็ตของ FEC นั้น ๆ จะต้องเดินทางผ่านไปโดยเส้นทางนี้ไม่จำเป็นต้องเป็นเส้นทางที่สั้นที่สุดหรือดีที่สุด เพียงแต่ว่าเส้นทางที่สร้างขึ้นนี้จะต้องสามารถตอบสนองต่อความต้องการใช้งานของผู้ใช้ได้อย่างเหมาะสมเท่านั้นและทรัพยากรตลอดเส้นทางดังกล่าวก็จะต้องถูกสำรองไว้สำหรับทุก FEC ที่ใช้บริการเพื่อที่รับประกันคุณภาพของการให้บริการได้อย่างถูกต้อง การหาเส้นทางแบบนี้เนื่องจากไม่จำเป็นที่จะต้องเลือกใช้เส้นทางที่ดีที่สุด ดังนั้นจึงมีข้อดีที่สามารถจะช่วยกระจายทราฟฟิกให้เกิดการใช้งานอย่างสมดุลในระบบเครือข่ายได้ และยังสามารถตอบสนองความต้องการของผู้ใช้ และสามารถบังคับใช้นโยบายในการบริหารระบบเครือข่ายได้อีกด้วย

เส้นทาง LSP ใด ๆ นั้นเป็นเส้นทางที่ใช้สำหรับการสื่อสารทางเดียวเท่านั้น ดังนั้นการสื่อสารสองทางจำเป็นต้องสร้างเส้นทาง LSP อย่างน้อย 2 เส้นทางในทิศทางที่ตรงกันข้ามกันจึงจะสามารถรองรับการสื่อสารสองทางได้

2.3.8 ขอบเขตของหมายเลขเลเบิล (LSP Space)

ขอบเขตของหมายเลขเลเบิลที่ใช้ในการระบุเส้นทางภายใน MPLS นั้น นอกจากจะขึ้นอยู่กับวิธีการกำหนดค่าเลเบิลดังที่กล่าวไปแล้วคือ ถ้ากำหนดให้เลเบิลถูกแทรกอยู่ระหว่างส่วนหัวของโปรโตคอลเลเยอร์ 2 และส่วนหัวของโปรโตคอลเลเยอร์ 3 นั้น ค่าของเลเบิลจะมีค่าอยู่ระหว่าง 0 ถึง 2^{20}-1 แต่ถ้าเป็น MPLS ที่ทำงานอยู่บนเครือข่าย ATM หรือ frame relay แล้วเลือกใช้ค่าบ่งชี้เส้นทางของโปรโตคอลเหล่านั้นเป็นเลเบิล ขอบเขตของค่าเลเบิลก็จะถูกกำหนดด้วยขนาดของค่าบ่งชี้เส้นทางภายในโปรโตคอลเหล่านั้นแทน

นอกจากนี้แล้วการกำหนดเลเบิลยังขึ้นอยู่กับว่า การควบคุมให้ค่าของเลเบิลเป็นค่าที่ซ้ำกันนั้นเป็นการควบคุมแบบแยกกันระหว่างพอร์ตสื่อสาร (unique within an interface) หรือเป็นการควบคุมแบบรวมกันภายในโหนดใด ๆ ซึ่งการควบคุมแบบแรกนั้นทำให้มีจำนวนของเลเบิลที่นำมากำหนดให้แก่ FEC ใด ๆ ได้มากกว่าวิธีการที่สองเป็นจำนวนเท่าของจำนวนพอร์ตสื่อสารที่มีอยู่

2.3.9 การรวมเลเบิล (Label Merging)

ทราฟฟิกโฟล์วจากหลาย ๆ การเชื่อมต่อ (connection) ใด ๆ ที่มีต้นทางที่ตำแหน่งเดียวกัน (ingress LER) และมีปลายทางเดียวกัน (egress LER) อาจจะถูกจับมารวมอยู่ใน FEC เดียวกัน และใช้เลเบิลเดียวกันไปตลอดเส้นทางภายในระบบเครือข่าย MPLS ก็ได้ ถ้าหากการเชื่อมต่อเหล่านั้นมีความต้องการในการสื่อสารที่เหมือนกันหรือใกล้เคียงกัน การนำเอาทราฟฟิกหลาย ๆ โฟล์วนี้มารวมกันและนำส่งไปภายใต้การควบคุมแบบเดียวกันนั้นถูกเรียกว่า stream merging หรือ aggregation of flows

ถ้าหากระบบเครือข่าย MPLS นั้นวิ่งอยู่บนระบบเครือข่าย ATM แล้ว LSR จะสามารถใช้ค่าของ VPI หรือ VCI ในการรวมทราฟฟิกโฟลว์ได้เช่นเดียวกัน

2.3.10 การรักษาเลเบิล (Label Retention)

การกำหนดและประกาศเลเบิลภายในระบบเครือข่าย MPLS โดยปกติแล้วโหนด LSR ใด ๆ จะกำหนดเลเบิลให้แก่ FEC โดยอิสระ แล้วประกาศเลเบิลนั้นกลับไปให้ LSR ก่อนหน้า ซึ่งเป็นโหนดที่ส่งแพ็กเก็ตร้องขอการเชื่อมต่อมาให้ ดังนั้นโหนดที่ได้รับการประกาศเลเบิลย่อมจะต้องรู้จักทั้ง FEC นั้น ๆ และทราบว่าโหนดที่ประกาศเลเบิลมานั้นเป็นโหนดในเส้นทางที่กำหนดให้แก่ FEC นั้นหรือไม่ ในกรณีที่ LSR ใด ๆ ได้รับการประกาศเลเบิลจากโหนดที่ไม่ใช่โหนดภายในเส้นทางเดิมที่กำหนดไว้สำหรับ FEC นั้น ๆ LSR ก็มีแนวทางในการตอบสนองต่อการประกาศเลเบิลนั้นได้ 2 แนวทางดังต่อไปนี้

1. Conservative mode ถ้า LSR ทำงานในโหมดนี้ LSR จะเพิกเฉยต่อการได้รับการประกาศเลเบิลที่ได้รับ โหมดการทำงานนี้เป็นโหมดที่ถูกแนะนำให้เลือกใช้โดย IETF

2. Liberal mode ในโหมดนี้ LSR จะเพิ่มข้อมูลที่ใช้ในการแม็ประหว่าง FEC และ เลเบิลลงไปใน LIB ถึงแม้ว่าจะมีข้อมูลการแม็ประหว่าง FEC นั้น ๆ และเลเบิลจาก LSR โหนดอื่นอยู่แล้วก็ตาม ซึ่งการที่ LSR ทำงานในลักษณะนี้จะเป็นการช่วยให้ FEC ใด ๆ มีเส้นทาง LSP สำรองที่สามารถนำมาเลือกใช้ได้ทันทีที่เส้นทาง LSP หลัก ที่กำหนดไว้เกิดความเสียหาย

2.3.11 การควบคุมการประกาศเลเบิล (Label Control)

IETF ได้กำหนดโหมดการควบคุมการประกาศเลเบิลไปยังโหนดข้างเคียงดังต่อไปนี้

1. Independent LSR จะกำหนดเลเบิลและประกาศเลเบิลออกไปให้แก่ FEC ใดๆ โดยอิสระจากการตัดสินใจของ LSR โหนดอื่น ๆ เมื่อพบว่ามีเส้นทางที่ให้บริการได้สำหรับ FEC นั้น ๆ

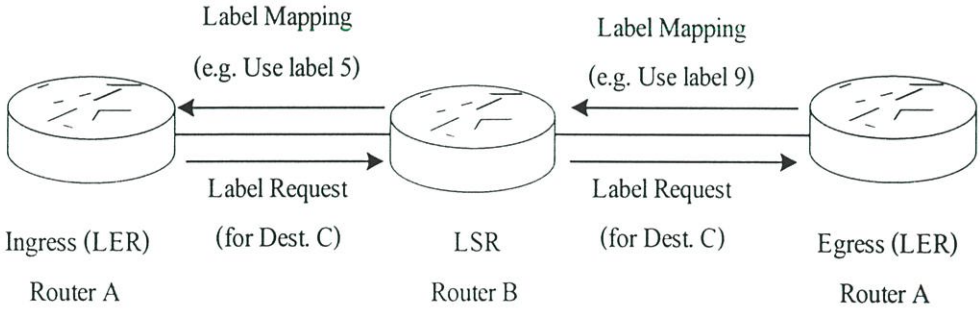
2. Ordered การทำงานในโหมดนี้นั้น LSR ใด ๆ จะกำหนดค่าเลเบิลให้แก่ FEC ใด ๆ และประกาศเลเบิลออกไปก็ต่อเมื่อ LSR โหนดนั้น เป็นเราท์เตอร์ปากทางขาออกของระบบเครือข่ายย่อย MPLS หรือเมื่อได้รับการประกาศเลเบิลจากโหนดถัดไปในเส้นทางที่กำหนดสำหรับ FEC ใด ๆ และโหมดการทำงานนี้เป็นโหมดการทำงานที่ถูกแนะนำให้เลือกใช้

2.3.12 กลไกของสัญญาณควบคุม (Signaling Mechanism)

1. Label Request เมื่อ LSR ใด ๆ ได้รับการร้องขอเส้นทางของ FEC ใหม่ก็จะกำหนดค่าเลเบิลให้แก่ FEC นั้น ๆ เมื่อทราบเส้นทางที่เป็นไปได้ แล้วจะร้องขอให้ LSR โหนดถัดไปในเส้นทางช่วยกำหนดเลเบิลให้แก่ FEC นั้นเป็นทอด ๆ ไปจนถึงเราท์เตอร์ปากทางขาออกของเครือข่าย MPLS

2. Label Mapping LSR ใด ๆ จะตอบสนองต่อการร้องขอเลเบิลเพื่อกำหนดเส้นทางด้วยการกำหนดค่าเลเบิลให้แก่ FEC และประกาศเลเบิลย้อนกลับไปให้แก่ LSR โหนดที่ร้องขอมา

ขั้นตอนการควบคุมของการสร้างเส้นทางและการสื่อสารจะเป็นไปดังอธิบายไว้ในรูปที่ 2.12



รูปที่ 2.12 กลไกการควบคุมสัญญาณ

2.3.13 โปรโตคอลประกาศค่าเลเบิลในระบบเครือข่าย (Label Distribution Protocol)

LDP เป็นโปรโตคอลใหม่ ที่ถูกนำเสนอขึ้นมาเพื่อเป็นอีกหนึ่งทางเลือกสำหรับใช้เป็นโปรโตคอลประกาศค่าเลเบิลในระบบเครือข่าย MPLS ผลลัพธ์ของการประกาศเลเบิลนี้ในภาพรวมก็คือได้ข้อมูลที่บ่งบอกถึงโครงสร้างของเส้นทาง LSP สำหรับ FEC ใด ๆ การแลกเปลี่ยนข้อมูลเลเบิลระหว่างกันนี้จะดำเนินการภายใต้เซสชั่น (session) ที่ถูกจัดตั้งขึ้นมาระหว่าง LSR คู่ใด ๆ (LDP peer) ซึ่งไม่จำเป็นที่จะต้องเป็น LSR โหนดที่อยู่ติดกันก็ได้ ข่าวสารที่จะมีการแลกเปลี่ยนกันระหว่าง LDP peer ถูกกำหนดไว้ดังต่อไปนี้

1. Discovery messages ใช้ในการประกาศตัวของ LSR ว่า ณ ขณะนั้นมีสถานะการทำงานเป็นอย่างไร

2. Session messages ใช้ในการจัดตั้งเซสชั่น ตรวจสอบสถานะ บำรุงรักษาเซสชั่น และยกเลิกเซสชั่น

3. Advertisement messages ใช้ในการสร้าง แก้ไข และลบข้อมูลเลเบิลสำหรับ FEC ใด ๆ

4. Notification messages ช่วยในการรายงานความผิดพลาดและการกำหนดพารามิเตอร์ที่ใช้ในการควบคุมต่าง ๆ

2.3.14 โครงสร้างเลเบิลแบบสแตค (Label Stack)

โครงสร้างเลเบิลแบบสแตคช่วยให้การทำงานของระบบเครือข่าย MPLS สามารถดำเนินการในลักษณะที่เป็นลำดับชั้น (hierarchical) ได้ โดยโครงสร้างการทำงานแบบพื้นฐานนั้น LSR ทุกตัวควรจะสามารถหาเส้นทางได้ทั้งในลักษณะที่เป็นเส้นทางโดยละเอียด (fine grain level) คือรู้ว่าเส้นทางนั้นผ่าน LSR โหนดใดบ้างภายใต้ระบบเครือข่ายย่อยเดียวกัน และสามารถหาเส้นทางในระดับเครือข่ายที่สูงขึ้นมาได้ (domain-by-domain level) กล่าวคือรู้จักเส้นทางว่าผ่านระบบเครือข่ายย่อย ๆ ใดบ้างไปจนถึงปลายทาง ซึ่งการหาเส้นทางในระดับเครือข่ายนั้นจะช่วยสนับสนุนการทำงานในลักษณะอุโมงค์สื่อสารของ MPLS ได้

2.3.1.5 การบริหารการจราจรของทราฟฟิก (Traffic Engineering)

การบริหารการจราจรของทราฟฟิก [10] เป็นกระบวนการที่มีจุดมุ่งหมายในการเพิ่มอัตราการใช้ประโยชน์ (utilization) ภายในระบบเครือข่ายด้วยความพยายามที่จะกระจายโหลดของทราฟฟิกไปยังส่วนต่าง ๆ ของระบบเครือข่ายให้เกิดความสมดุลและเหมาะสมมากที่สุดภายใต้นโยบายที่กำหนด ทั้งนี้ก็เพื่อลดโอกาสเสี่ยงในการที่จะเกิดความคับคั่งของการสื่อสารข้อมูลในเส้นทางใด ๆ ซึ่งจะส่งผลกระทบไปยังโหนดต่าง ๆ ในเส้นทางอย่างรวดเร็ว และทำให้ประสิทธิภาพของระบบลดลงและขาดความน่าเชื่อถือได้ หลักการที่สำคัญของการบริหารการจราจรภายในระบบเครือข่ายนั้นก็คือ การเลือกใช้เส้นทางสำหรับการเชื่อมต่อใด ๆ นั้นไม่จำเป็นจะต้องเป็นเส้นทางที่ดีที่สุดหรือสั้นที่สุดเท่านั้น แต่จะต้องคำนึงถึงความเหมาะสมต่อความต้องการและข้อตกลงกับผู้ใช้เป็นสำคัญ การบริหารการจราจรในบางโครงข่ายนั้น ถึงกับยอมให้แพ็กเก็ตของทราฟฟิกโฟล์วเดียวกันเดินทางไป

บนเส้นทางที่ต่างกันได้ ซึ่งการให้บริการเช่นนี้ก็เปิดช่องให้ระบบเครือข่ายสามารถให้บริการที่หลากหลายสำหรับความต้องการที่แตกต่างของผู้ใช้ได้อีกด้วย (differentiated services)

การบริหารการจราจรระบบเครือข่าย MPLS นั้น ได้เลือกนำเอาการกำหนดเส้นทางแบบแน่ชัด (explicit routing) มาใช้ โดยเราท์เตอร์ปากทางขาเข้าอาจจะกำหนดเส้นทาง LSP ขึ้นสำหรับ FEC ใด ๆ มากกว่า 1 เส้นทาง แต่ทุกเส้นทางจะต้องสามารถให้บริการที่สอดคล้องกับความต้องการของผู้ใช้ได้ ซึ่งโปรโตคอล RSCP และ CR-LDP นั้นก็สามารถนำมาใช้ในการบริการการจราจรภายใน MPLS ได้ นอกจากนี้การบริหารการจราจรใน MPLS ก็ยังคงจำเป็นต้องใช้ผู้ดูแลระบบเครือข่ายในการปรับปรุงค่าพารามิเตอร์ควบคุมต่าง ๆ ให้เหมาะสมกับสภาวะการใช้งานของผู้ใช้ร่วมด้วย

2.4 การหาเส้นทางแบบคำนึงถึงความต้องการของแอพพลิเคชั่น (QoS-based routing)

ความต้องการในการใช้บริการระบบเครือข่ายของผู้ใช้ มักจะถูกเสนอให้แสดงด้วยค่าพารามิเตอร์ต่างๆ ดังต่อไปนี้ เช่น คุณสมบัติของลิงค์ (ความกว้างของแบนด์วิดท์ ระยะทาง เวลาหน่วง และอื่น ๆ) จำนวนโหนดในเส้นทาง (hop count) และระดับความต้องการในการรับประกันคุณภาพในการให้บริการของเส้นทาง (QoS parameter) การเลือกใช้และจัดตั้งเส้นทางการส่งแพ็กเก็ตข้อมูลเพื่อให้สอดคล้องกับความต้องการของผู้ใช้ (CR-LSP) อาจจะใช้การกำหนดเส้นทางอย่างแน่ชัด (explicit routing) หรือใช้การควบคุมการจัดลำดับแถวคอยและการจัดลำดับส่งข้อมูล (queuing and scheduling) ที่จะต้องคำนึงถึงค่าพารามิเตอร์ความต้องการของผู้ใช้ควบคู่ไปกับข้อมูลของสภาวะระบบเครือข่ายประกอบกัน

การเลือกใช้เส้นทางให้เหมาะสมกับความต้องการของ FEC ใด ๆ นั้น เส้นทางที่เลือกใช้อาจจะไม่ใช่เส้นทางที่ดีที่สุดหรือสั้นที่สุด แต่ว่าอาจจะเป็นเส้นทางที่มีภาระโหลดในขณะนั้นน้อยกว่าเส้นทางอื่น ๆ ซึ่งการตัดสินใจเช่นนี้นอกจากจะเป็นการเลือกเส้นทางที่อาจจะมีระยะเวลาหน่วงของการสื่อสารน้อยกว่าเส้นทางที่มีความคับคั่งมากกว่าแล้ว ยังเป็นการเพิ่มอัตราการใช้สอยของระบบเครือข่ายโดยรวมให้เพิ่มมากขึ้นอีกด้วย แต่ประโยชน์ที่อาจจะเพิ่มขึ้นเหล่านี้ก็ต้องอาศัยการพัฒนาอัลกอริธึมการหาเส้นทางที่มีความยุ่งยากซับซ้อนมากกว่าเดิมอีกไม่น้อย โดยโปรโตคอลที่ IETF ได้นำเสนอเพื่อใช้ในกระบวนการนี้ก็คือ CR-LDP

นอกจากกลไกที่ใช้ในการเลือกเส้นทางจะมีความสำคัญต่อการให้บริการแบบรับประกันคุณภาพแล้ว ระบบเครือข่ายยังต้องอาศัยกลไกอีกหลายกลไกอย่างเช่น วิศวกรรมทราฟฟิกและการควบคุมการไหลของข้อมูล ที่จะช่วยควบคุมให้เกิดการให้บริการที่มีคุณภาพได้ในขณะที่ยังคงรักษาประสิทธิภาพของระบบโดยรวมไว้ได้

การจัดตั้งเส้นทางแบบคำนึงถึงคุณภาพในการให้บริการนั้น ต้องอาศัยการทำงานร่วมกันระหว่างอัลกอริธึมการหาเส้นทางแบบคำนึงถึงคุณภาพของเส้นทาง การควบคุมการอนุมัติใช้ทรัพยากร (call admission control) และวิศวกรรมจราจร

2.4.1 ประสิทธิภาพในการจัดตั้งเส้นทางที่มีคุณภาพในการให้บริการ

การจัดตั้งเส้นทางแบบคำนึงถึงคุณภาพในการให้บริการประกอบขึ้นจากกลไกการจัดตั้งเส้นทาง 2 รูปแบบด้วยกันคือ Inter-domain path-setup และ intradomain path-setup ซึ่งเป็นโครงสร้างที่สนับสนุนสถาปัตยกรรมการจัดตั้งเส้นทางแบบลำดับชั้น โดยกลไกการจัดตั้งเส้นทางแบบ intradomain นั้น เน้นที่การตอบสนองให้สอดคล้องกับความต้องการมากที่สุด โดยที่กลไกการทำงานและกลไกการแลกเปลี่ยนข้อมูลข่าวสารภายในโดเมนนั้นคล้ายคลึงกับสถาปัตยกรรม IGP/EGP ที่กระบวนการคำนวณเส้นทางภายในโดเมนค่อนข้างซับซ้อน

การจัดตั้งเส้นทางที่มีคุณภาพในการให้บริการแบบ Interdomain นั้นคุณภาพของการให้บริการควรที่จะสามารถตรวจวัดได้ การแลกเปลี่ยนข้อมูลของเส้นทางระหว่างโดเมนนี้มักจะไม่ให้ประโยชน์ในแง่ของการใช้สอยทรัพยากรมากนัก เพราะต้องอาศัยการแลกเปลี่ยนข้อมูลกันอย่างบ่อยครั้งเพื่อให้สอดคล้องกับการเปลี่ยนแปลงของปริมาณทรัพยากรที่เกิดขึ้นอย่างไดนามิก แต่อย่างน้อยที่สุด การแลกเปลี่ยนข้อมูลเส้นทางระหว่างโดเมนนี้ควรจะเกิดขึ้นอย่างเพียงพอที่จะสามารถใช้ตรวจสอบความสามารถในการเข้าถึง (reachability) ระดับคุณภาพของบริการที่ได้รับและการคำนวณหาเส้นทางได้อย่างถูกต้อง ซึ่งโดยมากแล้วข้อมูลข่าวสาร QoS ค่อนข้างจะคงที่เพราะอาศัยข้อมูลการใช้บริการในลักษณะที่เป็นแบบค่าโดยรวมระหว่างโดเมน ซึ่งสามารถใช้อธิบายรูปแบบการจราจรระหว่างโดเมนที่อยู่ข้างเคียงกันได้เป็นอย่างดี ทำให้สามารถควบคุมคุณภาพของการให้บริการได้ดีกว่าการให้บริการแบบแบ่งแยกกับทราฟฟิกที่มักจะมีความแปรปรวนสูงกว่าทราฟฟิกแบบรวมมาก

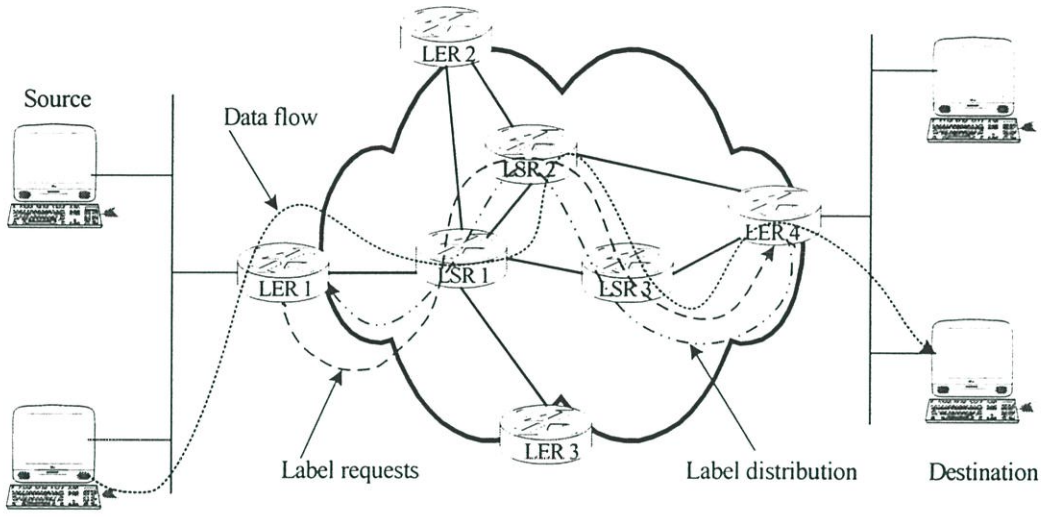
2.4.2 อัลกอริธึมสำหรับจัดตั้งเส้นทางที่มีคุณภาพในการให้บริการ

ข้างล่างนี้จะแสดงรายละเอียดของอัลกอริธึมการจัดตั้งเส้นทางแบบ unicast ซึ่งเป็นการนำเสนอแนวคิดการจัดตั้งเส้นทางแบบหลากหลายวิธี รวมถึงแบบช่วงต่อช่วงและแบบกำหนดเส้นทางครั้งเดียวโดยตรง รวมทั้งสถานะการเชื่อมต่อ และเวกเตอร์ระยะทาง, แบนด์วิดท์ และเวลาหน่วงของการจัดตั้งเส้นทางที่มีคุณภาพการให้บริการ Wang และ Crowcroft [13] ได้เสนอข้อคิดเห็นในการจัดตั้งเส้นทางที่มีคุณภาพของการให้บริการทั้งแบบรวมศูนย์และแบบกระจาย การจัดตั้งเส้นทางแบบรวมศูนย์เป็นวิธีการจัดตั้งเส้นทางที่ศูนย์รวบรวมข้อมูลข่าวสารเส้นทางภายในโดเมนใด ๆ เมทริกซ์ของระยะทางที่มักจะถูกนำมาใช้คือ จำนวนแบนด์วิดท์ และเวลารอคอย โทโปโลยีของระบบเครือข่ายจะถูกจำลองด้วยกราฟ โดยพารามิเตอร์ที่จะต้องใช้ในการกำกับลิงค์ที่เชื่อมระหว่างแต่ละโหนด (หรือโดเมน) จะประกอบไปด้วย แบนด์วิดท์สำรอง แบนด์วิดท์ที่คงเหลือ (bottle neck) ค่าเวลาหน่วง และระยะทางของสายสื่อสาร

2.5 การเชื่อมต่อของแอพพลิเคชั่น (MPLS operation)

การสื่อสารข้อมูลผ่านระบบเครือข่าย MPLS นั้นการเชื่อมต่อของแอพพลิเคชั่นหลาย ๆ แอพพลิเคชั่นระหว่าง 2 โฮสต์ใด ๆ อาจจะไม่จำเป็นที่ต้องใช้เส้นทางเดียวกันสำหรับทุก ๆ การเชื่อมต่อก็ได้ นอกเหนือจากการพิจารณาสภาวะของระบบเครือข่ายแล้ว การจัดตั้งเส้นทางใน MPLS นั้นจะคำนึงถึงรูปแบบของทราฟฟิกและความต้องการของทราฟฟิกโฟล์วนั้น ๆ ประกอบร่วมด้วยในขั้นตอนการเลือกเส้นทาง

ขั้นตอนการทำงานของระบบเครือข่าย MPLS ตั้งแต่เริ่มต้นร้องขอเส้นทางไปจนกระทั่งส่งแพ็กเก็ตข้อมูลสามารถแสดงได้ดังตัวอย่างในรูปที่ 2.13



รูปที่ 2.13 การทำงานของระบบเครือข่าย MPLS ตั้งแต่เริ่มต้นร้องขอเส้นทางไปจนกระทั่งส่งแพ็กเก็ตข้อมูล

ขั้นตอนที่ 1 การกำหนดใช้เลเบิลและการประกาศเลเบิล LSR ใด ๆ จะกำหนดค่าเลเบิลที่จะใช้ในการค้นหาเส้นทางภายในตาราง LIB ให้แก่ FEC ใด ๆ แล้วจะประกาศค่าเลเบิลนั้นย้อนกลับไปให้กับ LSR ที่ร้องขอมา การร้องขอเลเบิลนี้ เป็นส่วนหนึ่งของการร้องขอเส้นทางซึ่งอาจจะมีการระบุลักษณะของทราฟฟิกและความต้องการของการเชื่อมต่อนั้นเพื่อใช้ในการร้องขอทรัพยากรและถูกใช้ในการต่อรองและสำรองทรัพยากรของระบบเครือข่ายด้วย (negotiate) ขั้นตอนแรกของการร้องขอเส้นทางนี้เป็นหน้าที่ของโปรโตคอล LDP ที่เลือกใช้ TCP เป็นโปรโตคอลชั้นนำส่ง

ขั้นตอนที่ 2 การสร้างข้อมูลในตารางเส้นทาง LIB เมื่อ LSR ใด ๆ ได้รับการประกาศเลเบิล ก็จะนำข้อมูลของ FEC และข้อมูลเส้นทางที่เกี่ยวข้อง อาทิเช่น FEC หมายเลขพอร์ตขาเข้า (incoming port no.) เลเบิลขาเข้า (incoming label) (ที่โหนด LSR นั้นเป็นผู้กำหนดขึ้นเอง) หมายเลขพอร์ตขาออก (outgoing port no.) เลเบิลขาออก (outgoing label) (ที่ได้รับมาจากการประกาศเลเบิลจาก

โหนดถัดไป) และข้อมูลพารามิเตอร์อื่น ๆ บรรจุลงในตาราง LIB เพื่อจะได้นำข้อมูลนั้นมาใช้ในช่วงของการแม็ปปิ้งต่อไป ข้อมูลภายใน LIB นี้สามารถที่จะปรับปรุงแก้ไขได้ เมื่อการสื่อสารของผู้ใช้จบสิ้นลง หรือเมื่อความต้องการของผู้ใช้เปลี่ยนแปลงไป และได้มีการตกลงรูปแบบของการสื่อสารและการให้บริการกับระบบเครือข่ายใหม่ (renegotiate)

ขั้นตอนที่ 3 การสร้าง LSP ทิศทางของการสร้างเส้นทาง LSP ก็คือลำดับของโหนด LSR ที่เพิ่มข้อมูลเส้นทางที่ได้จัดตั้งขึ้นในตาราง LIB นั่นเอง ซึ่งทำให้ทิศทางของการสร้างเส้นทางนั้นสวนทิศกันกับทิศทางของการร้องขอเส้นทางซึ่งเป็นทิศทางของการส่งข้อมูลอีกด้วย

ขั้นตอนที่ 4 การส่งผ่านแพ็กเก็ตข้อมูล

1. เมื่อเราท์เตอร์ปากทางขาเข้า (LER_1) ได้รับแพ็กเก็ตข้อมูลของ FEC ใด ๆ LER_1 ก็จะใช้ค่า FEC ในส่วนหัวของโปรโตคอลในการหาข้อมูลเส้นทางภายใน LIB ของ FEC นั้น ๆ และจะใช้ค่าเลเบิลและหมายเลขพอร์ตขาออกที่พบในการแทรกหรือบรรจุค่าเลเบิลและควบคุมการสวิทช์ให้แพ็กเก็ตนั้นถูกส่งออกไปยังเส้นทางที่ถูกต้อง

2. เมื่อแพ็กเก็ตข้อมูลเดินทางมาถึงยัง LSR ซึ่งเป็นเราท์เตอร์ภายใน (intermediate router) ระบบเครือข่าย MPLS LSR เหล่านั้นก็จะพิจารณาเลเบิลของแพ็กเก็ตเพื่อใช้ในการค้นหาข้อมูลในตาราง LIB และแม็ปปิ้งค่าของเลเบิลขาเข้าให้เป็นค่าของเลเบิลขาออกและ สวิทช์ต่อไปยัง LSR ตัวถัดไป (LSR_1, LSR_2, LSR_3)

3. เมื่อแพ็กเก็ตเดินทางมาถึงเราท์เตอร์ปากทางขาออก LER_4 ของระบบเครือข่ายย่อย MPLS LER_4 ก็จะลบส่วนหัวที่ใช้ควบคุมการสื่อสาร (ที่บรรจุเลเบิลอยู่ด้วย) ภายใน MPLS ออกและส่งออกไปยังระบบเครือข่ายปลายทางที่เชื่อมโยงอยู่กับพอร์ตสื่อสารขาออกที่พบในตาราง LIB ต่อไป

ตารางที่ 2.1 แสดงตัวอย่างข้อมูลของตาราง LIB

Input Port	Incoming Port Label	Output Port	Outgoing Port Label
1	3	2	6
1	9	4	7

ตารางดังกล่าวแสดงตัวอย่างข้อมูลของตาราง LIB ซึ่งเป็นตัวอย่างข้อมูลเส้นทางของ 2 ทราฟฟิกโฟล์วที่มีความต้องการในการสื่อสารที่แตกต่างกันแต่เป็นการสื่อสารระหว่างโฮสต์คู่เดียวกัน

1. ทราฟฟิกโฟล์วที่หนึ่งเป็นทราฟฟิกที่เกิดจากการแลกเปลี่ยนข้อมูลกันระหว่าง 2 เซิร์ฟเวอร์ ซึ่งอาจจะใช้โปรแกรม FTP (File Transfer Protocol)

2. ทราฟฟิกโฟล์วที่สองเป็นทราฟฟิกของการสื่อสารภาพและเสียงแบบต่อเนื่อง เช่น video-conference ซึ่งต้องการการควบคุมการรับส่งข้อมูลที่เหมาะสมกับความต้องการทั้งทางด้านเวลาและแบนด์วิดท์

3. ทราฟฟิกโฟล์วทั้งสองนี้ถูก LSR พิจารณาพบว่ามีความต้องการที่แตกต่างกัน ถึงแม้จะเป็นการสื่อสารระหว่างโฮสต์คู่เดียวกันก็ตาม ดังนั้นทราฟฟิกโฟล์วทั้งสองจึงถูกจำแนกออกเป็น FEC ต่างกลุ่มกัน

4. ค่าของเลเบิลที่กำหนดให้แก่ทราฟฟิกโฟล์วที่หนึ่งคือ 3 และเลเบิลของทราฟฟิกโฟล์วที่สองคือ 9

5. หมายเลขพอร์ตสื่อสารขาเข้าของทั้งสองทราฟฟิกโฟล์วคือ พอร์ตหมายเลข 1

6. หมายเลขพอร์ตสื่อสารขาออกที่ถูกกำหนดให้เป็นเส้นทางสำหรับทราฟฟิกโฟล์วทั้งสองคือพอร์ตหมายเลข 2 และ 4 ตามลำดับ

7. เมื่อ LSR นี้ได้รับการประกาศเลเบิลของ FEC ที่ร้องขอไป ก็จะนำค่าเลเบิลที่ได้จากพอร์ตหมายเลข 2 ซึ่งก็คือเลเบิล 6 สำหรับทราฟฟิกโฟล์วที่หนึ่ง และค่าเลเบิลที่ได้จากพอร์ตหมายเลข 4 ซึ่งก็คือเลเบิล 7 สำหรับทราฟฟิกโฟล์วที่สอง มาบรรจุลงในตาราง LIB เพื่อประโยชน์ในการแม็ปปิ้งต่อไป

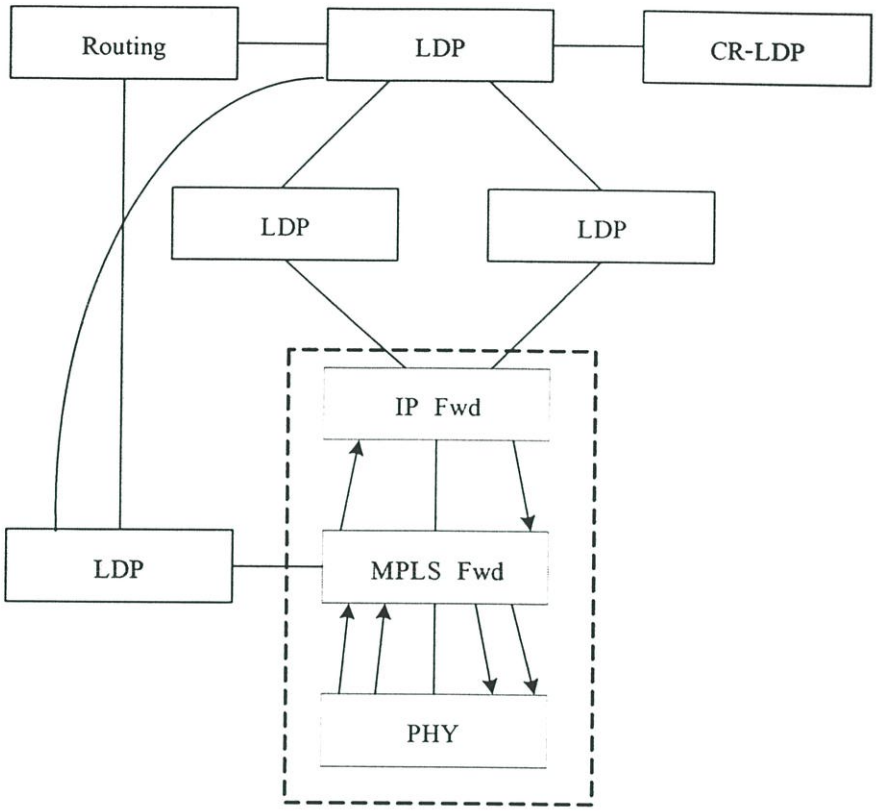
2.6 โครงสร้างการทำงานของโปรโตคอลสื่อสาร (MPLS Protocol Stack Architecture)

โครงสร้างการทำงานของโปรโตคอลสื่อสารที่ใช้เทคโนโลยีของ MPLS สามารถแบ่งย่อยได้เป็นดังนี้

1. โปรโตคอลชั้นควบคุมเครือข่าย (Network Layer เช่น IP)
2. การส่งแพ็คเก็ตข้อมูลระหว่างเครือข่าย (Edgae of network layer forwarding)
3. การสวิทชิ่งโดยการใช้เลเบิลในโครงข่ายหลัก
4. รูปแบบของเลเบิลและวิธีการกำหนดค่าเลเบิล
5. โปรโตคอลที่ใช้ในการควบคุมการสื่อสาร (Signaling protocol) ในที่นี้หมายถึงโปรโตคอลที่ใช้ในการร้องขอและประกาศเลเบิล
6. เทคโนโลยีการบริหารการจราจรภายในระบบเครือข่าย (traffic engineering)
7. เทคนิคที่ใช้ในการเชื่อมโยงการทำงานระหว่าง MPLS และโปรโตคอลในระดับชั้นเชื่อมต่อข้อมูล (layer 2) ที่มีอยู่หลากหลายชนิด เช่น ATM, Ethernet, Frame relay และ PPP เป็นต้น

รูปที่ 2.14 แสดงลำดับชั้นและการทำงานร่วมกันระหว่างโปรโตคอลต่าง ๆ เพื่อสนับสนุนการทำงานของระบบเครือข่ายแบบ MPLS โดยโปรโตคอลที่จะนำมาประยุกต์ใช้จริงอาจจะไม่จำเป็นจะต้องใช้โปรโตคอลเดียวกันกับที่แสดงไว้ในรูปก็ได้ เช่น โปรโตคอลที่ใช้ในการหาเส้นทางอาจจะเป็น OSPF, BGP หรือ PNNI ก็ได้ ในขณะที่การร้องขอเส้นทางและการประกาศเลเบิลนั้น อาจจะใช้ประโยชน์จาก BGP, RSVP หรือ LDP เลยก็ได้ โดยโปรโตคอลเหล่านี้ต่างก็อาศัยโปรโตคอลอย่าง TCP ในการนำส่งข้อมูลเช่นเดียวกัน โปรโตคอล LDP เองได้มีการกำหนดให้ใช้โปรโตคอล UDP ในช่วงของการเริ่มต้นการทำงาน (discovery phase) โดยการใช้แพ็คเก็ตทักทาย (hello packet)

ประกาศตัวตนและสถานะการทำงาน รวมทั้งร้องขอข้อมูลเบื้องต้นที่ใช้ในการควบคุมการทำงานจากโหนดข้างเคียง เช่น หมายเลขอ้างอิงของโหนดข้างคียง และโหมดการทำงาน เป็นต้น



รูปที่ 2.14 ลำดับชั้นและการทำงานร่วมกันระหว่างโปรโตคอลต่าง ๆ

ชั้นโปรแกรมสื่อสาร IP_Fwd ในรูปคือการทำงานของโปรโตคอล IP ปัจจุบันที่ใช้การหาเส้นทางแบบ classless โดยพยายามหาเส้นทางที่มีหมายเลขของเครือข่ายย่อยปลายทาง (subnet_id) ที่สอดคล้องกับที่อยู่ IP ปลายทางที่กำหนดในส่วนหัวของแพ็กเก็ตมากที่สุด (ยาวที่สุด) แต่ในระบบเครือข่าย MPLS นี้ การใช้งานโปรแกรม IP_Fwd จะพบในเราท์เตอร์ที่เป็น LER เท่านั้น ส่วน MPLS_Fwd เป็นโปรแกรมที่มีหน้าที่ในการแม็ปปิ้งระหว่างเลเบิลขาเข้าและเลเบิลขาออกอีกทั้งกำหนดพอร์ตขาออกให้แก่แพ็กเก็ตด้วย โปรแกรมสื่อสารที่ถูกล้อมกรอบด้วยเส้นประดังในรูปสามารถจะนำไปพัฒนาให้ทำงานอยู่ในส่วนของฮาร์ดแวร์ เพื่อเพิ่มประสิทธิภาพการทำงานของระบบเครือข่าย MPLS ให้สูงยิ่งขึ้นได้

2.7 แอพลิเคชั่นของเครือข่ายเทคโนโลยีในการสลับแพ็กเก็ต (MPLS Applications)

เทคโนโลยีของ MPLS ถูกคาดหวังว่าจะสามารถตอบสนองความต้องการของระบบเครือข่ายหลักได้อย่างลงตัว และสามารถจะนำมาใช้เป็นมาตรฐานในอนาคตได้

1. เพิ่มประสิทธิภาพในการส่งผ่านแพ็กเก็ต

- โดยการสวิทช์แพ็กเก็ตในเลเยอร์ 2 ที่สามารถทำงานด้วยฮาร์ดแวร์ในระดับความเร็วของสายสื่อสารได้

- เนื่องจากการทำงานของ MPLS ไม่มีความซับซ้อน ดังนั้นการนำไปพัฒนาให้เป็นรูปธรรมสามารถที่จะทำได้ง่าย

2. สนับสนุนการให้บริการที่หลากหลาย

- MPLS สนับสนุนการใช้โปรโตคอลที่สามารถจัดตั้งเส้นทางและการส่งผ่านข้อมูลเพื่อให้สอดคล้องกับความต้องการของผู้ใช้

3. สนับสนุนการขยายตัวของระบบเครือข่าย

- การหาเส้นทางภายใน MPLS สามารถหลีกเลี่ยงการใช้เส้นทางที่ทับซ้อนกัน ซึ่งเป็นปัญหาที่พบในการส่งแพ็กเก็ตข้อมูล IP ผ่านเครือข่าย ATM ได้

4. สามารถรองรับการสื่อสารระหว่างระบบเครือข่ายอินเตอร์เน็ต (IP) กับระบบเครือข่าย ATM เข้าด้วยกันได้

- MPLS ได้กำหนดการแปลงวิธีควบคุมการสื่อสารระหว่างระบบเครือข่าย IP และระบบเครือข่าย ATM ไว้แล้ว

- MPLS เป็นเทคโนโลยีที่สามารถนำเอาอุปกรณ์สื่อสารของระบบเครือข่ายเดิมที่มีอยู่ เช่น เราท์เตอร์ IP สวิทช์ ATM มาใช้ได้โดยไม่ต้องปรับปรุงอะไรมากนัก

บทที่ 3

อัลกอริธึมการสร้างเส้นทางแบบคำนึงถึงคุณภาพของการให้บริการ สำหรับระบบเครือข่าย MPLS ที่มีความสามารถในการแก้ไขข้อบกพร่องในการเลือกเส้นทางได้อย่างรวดเร็ว (Early Crankback QoS-Based Routing Algorithm : ECQR)

3.1 บทนำ

วิทยานิพนธ์ฉบับนี้เป็นการนำเสนออัลกอริธึมการหาเส้นทางแบบใหม่เพื่อนำมาใช้ในระบบเครือข่าย MPLS โดยได้ใช้การกำหนดบทบาทของเราท์เตอร์หรือสวิทช์ออกเป็น 2 ประเภทเช่นเดียวกันกับข้อเสนออื่น ๆ คือ เราท์เตอร์ที่อยู่บริเวณขอบของระบบเครือข่ายย่อย (LER : Label Edge Router หรือ ABR : Area Border Router) และเราท์เตอร์ที่อยู่ภายในระบบเครือข่ายย่อย (LSR : Label Switch Router)

หน้าที่แตกต่างกันของเราท์เตอร์สองประเภทนี้คือ หน้าที่ในช่วงของการสร้างเส้นทางเมื่อเกิดการร้องขอการเชื่อมต่อในระบบเครือข่ายขึ้น โดย LER ที่ปากทางขาเข้าของแพ็คเก็ตร้องขอการเชื่อมต่อจะทำ การพิจารณาที่อยู่ของโหนดปลายทาง และทำการค้นหาว่าควรจะสร้างเส้นทางผ่านเครือข่ายย่อยใดที่อยู่ติดกันเพื่อให้ได้เส้นทางที่สั้นที่สุด แล้วจึงมาพิจารณาว่า มี LER ใดบ้างที่เชื่อมโยงระบบเครือข่ายย่อยของตนไปยังระบบเครือข่ายย่อยนั้น แล้วเลือกเส้นทางสั้นที่สุดที่ไปถึง LER เหล่านั้นเพียง 1 เส้นทาง และนำเอาข้อมูลของเส้นทางนั้น ๆ ซึ่งจะเป็นการระบุหมายเลขของ LSR ไปจนถึง LER ขาออกของระบบเครือข่ายย่อยบรรจุลงในแพ็คเก็ตร้องขอเส้นทาง และทำการส่งแพ็คเก็ตดังกล่าวไปยังเราท์เตอร์ถัดไปภายในเส้นทางที่กำหนด โดยทั้งนี้การเลือกเส้นทางที่กล่าวมาเป็นการพิจารณาจากเส้นทางที่ได้คำนวณเอาไว้ล่วงหน้าด้วยอัลกอริธึม QOSPF

เมื่อ LSR ตัวถัดไปสามารถที่จะรองรับความต้องการของการเชื่อมต่อนั้น ๆ ได้ก็จะส่งแพ็คเก็ตร้องขอเส้นทางดังกล่าวไปเป็นทอด ๆ ตามเส้นทางที่ระบุไว้จนกว่าจะผ่าน LER ขาออกของระบบเครือข่ายย่อยไปยังระบบเครือข่ายย่อยถัดไป แต่ถ้าหากปรากฏว่ามี LSR ตัวใดตัวหนึ่งในเส้นทางไม่มีทรัพยากรเหลือเพียงพอที่จะให้บริการตามที่ร้องขอมาได้นั้น ก็จะต้องส่งแพ็คเก็ตปฏิเสธการร้องขอนั้น ๆ ย้อนกลับไปจนถึง LER ปากทางเข้าเพื่อกำหนดหาเส้นทางใหม่ ซึ่ง LER ปากทางเข้าก็อาจจะเลือกใช้เส้นทางอื่น ๆ จากเส้นทางที่เหลือจากการคำนวณล่วงหน้าหรือไม่ ก็จะทำการคำนวณหาเส้นทางใหม่ (on-demand calculation) [9] โดยตัดทอนเอาสายสัญญาณ (link) ที่มีปัญหาออกจากข้อมูลสายสัญญาณเสียก่อนแล้วใช้กระบวนการเช่นเดิมในการสร้างเส้นทาง

3.2 การแก้ไขปัญหาในการเลือกเส้นทางในเครือข่ายย่อย

ในวิทยานิพนธ์นี้ได้นำเสนอแนวทางในการปรับปรุงอัลกอริธึมการแก้ไขปัญหาในการเลือกเส้นทางในเครือข่ายย่อยให้มีความสามารถในการแก้ไขข้อ ผิดพลาดของการเลือกเส้นทาง ด้วยการแนบเส้นทางเผื่อเลือกไปกับแพ็คเก็ตร้องขอการเชื่อมต่อด้วย โดยเส้นทางเผื่อเลือกนี้สามารถจะแนบไปได้มากกว่า 1 เส้นทาง ถ้าหากสายสัญญาณขาออกที่ถูกระบุไว้ในเส้นทางหลักนั้นรองรับความต้องการได้ก็จะทำการส่งต่อแพ็คเก็ตร้องขอเส้นทางนั้นไปยังโหนดถัดไป หรือเลือกใช้เส้นทางเผื่อเลือกอื่น ๆ ที่ผ่านตัวมันในกรณีที่สายสัญญาณในเส้นทางหลักไม่สามารถรองรับได้ หรือใช้การส่งแพ็คเก็ตปฏิเสธการร้องขอเส้นทางย้อนกลับไปยังโหนดก่อนหน้าเพื่อให้พิจารณาหาเส้นทางที่เป็นไปได้เส้นทางอื่น ๆ จากข้อมูลเส้นทางที่เหลือจากการตัดทอนเอาเส้นทางที่เป็นไปไม่ได้ออก แล้วการส่งแพ็คเก็ตปฏิเสธการร้องขอนี้จะถูกส่งย้อนกลับไปเป็นทอด ๆ จนกว่าจะถึงเราท์เตอร์ตัวที่อยู่ในเส้นทางเผื่อเลือกนั้น และเราท์เตอร์ตัวนั้นก็จะทำการส่งแพ็คเก็ตร้องขอเส้นทางไปยังเราท์เตอร์ถัดไปในเส้นทางใหม่ การสร้างเส้นทางย่อยนี้จะสำเร็จได้เมื่อมีการยอมรับแพ็คเก็ตร้องขอการเชื่อมต่อนี้ไปจนถึง LER ปากทางของเครือข่ายย่อยถัดไป

แบบจำลองทางคณิตศาสตร์ของวิธีการจัดตั้งเส้นทางแบบ ECQR

ถ้ากำหนดให้ N^{k_r} คือจำนวนเราท์เตอร์ในเส้นทาง r ที่ถูกกำหนดให้สำหรับการเชื่อมต่อ k ระยะเวลาขั้นต่ำที่ต้องใช้ในการสร้างเส้นทาง (ไม่รวม queuing delay) จะมีค่าเป็น

$$D_{setup}\left(N_{k_r}\right) = \sum_{i=1}^{N_k-1} \tau_{ij}\Big|_{j-i+1} + \sum_{i=1}^{N_k-1} \delta_i + \sum_{i=2}^{N_k} \rho_i + \eta_{k_r} \qquad (1)$$

τ_{ij} คือ propagation delay ระหว่างการส่งจากโหนด i ไปยังโหนด j

δ_i คือ transmission delay ของแพ็คเก็ตร้องขอการเชื่อมต่อบนสายสัญญาณของโหนด i

ρ_i คือ processing delay ในโหนด i ที่ใช้ในการพิจารณาและเปลี่ยนแปลงข้อมูลทรัพยากรรวมถึงการปรับเปลี่ยนข้อมูลในแพ็คเก็ตที่จะนำส่งออกไปด้วย

η_{k_r} คือเวลาที่ใช้ในการหาเส้นทางในโหนดปากทางรวมกับเวลาที่ใช้ในการปรับปรุงแพ็คเก็ตที่จะนำส่ง

ซึ่งกำหนดให้ค่าของ τ_{ij}, δ_i และ ρ_i เป็นค่าคงที่สำหรับโหนด i และ j ใด ๆ และเมื่อกำหนดให้ $\overline{N}$ คือจำนวนเราท์เตอร์เฉลี่ยของทุกเส้นทางที่เป็นไปได้ในระบบเครือข่าย ดังนั้นระยะเวลาขั้นต่ำที่ต้องใช้ในการสร้างเส้นทางจะมีค่าเป็น

$$\overline{D}_{setup} = \overline{N}(\overline{\tau} + \overline{\delta} + \overline{\rho} + \overline{\eta}) \qquad (2)$$

เมื่อเครื่องหมาย " − " (บาร์) หมายถึงค่าเฉลี่ยของค่าเหล่านั้นในเครือข่ายย่อยใด ๆ และถ้าหากปรากฏว่า สายสัญญาณขาออกของโหนดในลำดับที่ $M_{k_r}\left(2 \le M_{k_r} \le N_{k_r}\right)$ ในเส้นทางที่กำหนดไม่สามารถรองรับความต้องการได้ ก็จะเกิดการส่งแพ็คเก็ตปฏิเสธการร้องขอเส้นทางย้อนกลับมาจนถึงโหนดปากทาง ซึ่งเวลารวมที่ต้องใช้ในการร้องขอเส้นทางจากโหนดปากทางไปยังโหนดที่เกิดการบล๊อกและส่งแพ็คเก็ตปฏิเสธการร้องขอเส้นทางย้อนกลับมาจน ถึงโหนดปากทางจะใช้เวลาอย่างต่ำเท่ากับ

$$D_{CB}(M_{kr}) = \sum_{i=1}^{M_{mr}-1} \tau_{ij}\Big|_{j=i+1} + \sum_{i=2}^{M_{kr}} \rho_i + \eta_{kr} + \sum_{j=2}^{M_{kr}} \tau_{ij}\big|_{j=i+1} + \sum_{j=2}^{M_{kr}} \delta_i + \sum_{j=2}^{M_{kr}} \psi_i \qquad (3)$$

โดย ψ_j คือเวลาที่ใช้ในการประมวลผลแพ็คเก็ตปฏิเสธการร้องขอเส้นทางเพื่อส่งย้อนกลับ และโอกาสที่จะเกิดการยอมรับได้ตั้งแต่โหนดที่ 1 จนถึงโหนดที่ *j-1* และ ถูกปฏิเสธที่โหนด *j* ของเส้นทาง *r* คือ

$$B_{kr}(j) = \beta_{kr}(j)\prod_{i=1}^{j-1}(1-\beta_{kr}(i))\Big|_{j \le N_{kr}} \qquad (4)$$

เมื่อ $\beta_{k_r}(j)$ คือความน่าจะเป็นของการเกิดการบล๊อก (call blocking) ที่โหนด *j* ซึ่งพบว่าทรัพยากรของโหนด *j* ไม่เพียงพอสำหรับการเชื่อมต่อ *k* ในเส้นทาง *r* ใด ๆ ดังนั้นค่าที่คาดหวังสำหรับเลขลำดับของโหนดที่เกิดการบล๊อกจะมีค่าเป็น

$$E(M_{kr}) = \sum_{i=1}^{N_{kr}} i \cdot B_{kr}(j) + \sum_{i=N_{kr}+1}^{\infty} i \cdot B_{kr}(i) \qquad (5)$$

โดยค่าของ $B_{k_r}(i)$ ที่ใช้ในผลรวมที่สองซึ่งคำนวณได้จาก $B_{k_r}(i)$ ของโหนดลำดับที่ $i > N_{k_r}$ ซึ่งเกินจำนวนของโหนดที่มีอยู่จริงนั้น จำเป็นต้องใช้ค่าเฉลี่ย $B_{k_r}(i)$ ของทุกโหนดในเส้นทางแทนค่า $\beta_{k_r}(i)$ ของโหนดที่ไม่มีอยู่จริง และจากค่า $E(M_{k_r})$ ที

คำนวณได้ทำให้เราสามารถประมาณค่าของเวลาหน่วงที่อาจจะเกิดจากการบล็อกในเส้นทาง r ของการเชื่อมต่อ k ใด ๆ ได้เท่ากับ

$$\hat{D}_{CB}(M_{kr}) = \frac{E(M_{kr}) \cdot D_{CB}(N_{kr})}{N_{kr}} \tag{6}$$

และระยะเวลาเฉลี่ยที่อาจจะเสียเพิ่มขึ้นจากระยะเวลาขั้นต่ำที่ต้องใช้ในการสร้างเส้นทางสำหรับการเชื่อมต่อ k ใด ๆ ที่มีจำนวนเส้นทางที่เป็นไปได้จำนวน R เส้นทาง คือ

$$D_{CB} = \frac{\sum_{i=1}^{R} \hat{D}_{CB}(M_{kr})}{R} \tag{7}$$

เมื่อกำหนดให้โอกาสที่จะเกิดการบล๊อกขึ้นอย่างน้อย 1 โหนดในเส้นทาง r คือ

$$\beta_{kr} = 1 - \prod_{i=1}^{N_{kr}} (1 - \beta_{kr}(i)) \tag{8}$$

และถ้ากำหนดให้ $\overline{\beta}_k$ คือโอกาสเฉลี่ยของ β_{k_r} ในทุกเส้น ทางย่อยที่เป็นไปได้ในระบบเครือข่ายย่อยนั้น ๆ โดยกำหนดให้โอกาสที่จะเกิดการบล๊อกที่เวลาใกล้เคียงกันมีค่าเกือบเท่ากัน ดังนั้นโอกาสที่จะเกิดการบล๊อกขึ้น f ครั้งจากความพยายามในการสร้างเส้นทาง $f+1$ ครั้งของการเชื่อมต่อ k จะสามารถแสดงได้ด้วยสมการ **Bernoulli**

$$\zeta_k(f) = (\beta_k)^f (1 - \beta_k) \tag{9}$$

ก็จะได้ค่าเฉลี่ยของจำนวนครั้งที่อาจจะเกิดความผิดพลาดในการสร้างเส้นทางเป็น

$$E_k(f) = \sum_{f=1}^{\infty} f \cdot \zeta_k(f) \tag{10}$$

ดังนั้นค่าเฉลี่ยของเวลาที่จะต้องใช้ทั้งหมดในการสร้างเส้นทางในระบบเครือข่ายย่อยใด ๆ สำหรับการเชื่อมต่อ k ได้สำเร็จคือ

$$\overline{D}_{Total} = \overline{D}_{setup} + \overline{E}_k(f) \cdot \overline{D}_{CB} \tag{11}$$

3.3 ประสิทธิภาพของอัลกอริธึม ECQR

เพื่อที่จะลดปัญหาที่เกิดขึ้นจากการสร้างเส้นทางโดยใช้การกำหนดเส้นทางทั้งหมดในครั้งเดียวนั้น บทความนี้จึงได้พิจารณาเลือกใช้การกำหนดเส้นทางหลายเส้นทางไว้ตั้งแต่ครั้งแรก ส่งผลให้กระบวนการสร้างเส้นทางไม่จำเป็นต้องย้อนกลับไปจน ถึงเราท์เตอร์ปากทางทุกครั้ง เมื่อเกิดการบล็อกขึ้นที่โหนดลำดับที่ m_{k_r} ใด ๆ ในเส้นทางย่อย r เพราะโหนดต่าง ๆ ตั้งแต่โหนดลำดับที่ 1 ถึง m_{k_r} ตามลำดับในเส้นทางย่อย r นั้นอาจจะเป็นโหนดในเส้นทางเผื่อเลือก $\tilde{r}$ อื่น ๆ ที่มีโอกาสในการให้บริการได้ในเส้นทางเหล่านั้นเท่ากับ

$$P_{kr}(i) = \max\{(1-\beta_{kr}(i)\}\big|_{j\neq r} \tag{12}$$

และถ้ากำหนดให้จำนวนครั้งที่พบความผิดพลาดในการเลือกเส้นทางเป็น f ครั้ง (f เส้นทาง) แล้วระยะเวลาที่ต้องใช้ในการสร้างเส้นทางรวมในกรณีนี้จะเท่ากับ

$$D_{Total}(f) = \sum_{r=1}^{f+1} D_{progress} + \sum_{r=1}^{f} D_{back} \tag{13}$$

โดยที่

$$D_{back} = \sum_{j=y_{kr}+1}^{m_{kr}} \tau_{ij}\big|_{i=j+1} + \sum_{j=y_{kr}+1}^{m_{kr}} \delta_j + \sum_{j=y_{kr}+1}^{m_{kr}} \psi_j \tag{14}$$

$$D_{progress} = \sum_{i=1}^{m_{kr}} \tau_{ij}\big|_{j=i+1} + \sum_{i=1}^{m_{kr}} \delta_i + \sum_{i=2}^{m_{kr}} \rho_i + \eta_{kr} \tag{15}$$

ซึ่ง D'_{Total} นั้นเป็นตัวแปรสำคัญที่จะแสดงให้เห็นถึงประ สิทธิภาพของวิธีการที่นำเสนอเพื่อเปรียบเทียบกับวิธีการเดิมเพราะเป็นระยะเวลาที่คาดว่าจะต้องเสียไปในการสร้างเส้นทางในระบบเครือข่ายย่อยหนึ่ง ๆ ของวิธีที่นำเสนอซึ่งจะคำนวณได้ก็ต่อเมื่อเราทราบค่าของจำนวนโหนดหรือจำนวนก้าวที่ก้าวไปข้างหน้า m_{k_r} ในเส้นทางใหม่จากสมการที่ (4) และจำนวนก้าวที่ก้าวถอยหลัง S_{k_r} จากสมการต่อไปนี้

$$E(S_{kr}) = \sum_{i=1}^{m_{kr}} G_{kr}(i, m_{kr}) \cdot (m_{kr} - i + 1) \tag{16}$$

เมื่อ $G_{k_r}(y_{k_r}, m_{k_r})$ซึ่งมีโอกาสที่เกิดการบล็อกที่โหนด m_{k_r} และย้อนกลับมายังโหนดที่ให้บริการในเส้นทางอื่น ๆได้ซึ่งเป็นโหนดลำดับที่ y_{k_r} ของเส้นทางย่อยเดิมและมีค่าเป็น

$$G_{kr}(y_{kr}, m_{kr}) = P_{kr}(y_{kr}) \cdot \prod_{i=y_{kr}+1}^{m_{kr}} (1 - P_{kr}(i)) \tag{17}$$

ซึ่ง $E(S_{k_r})$ เป็นค่าโดยประมาณที่สามารถหาค่าได้เมื่อทราบค่าของ $\beta_{k_r}(i)$ และ $P_{k_r}(i)$ โหนด ต่าง ๆ และทำให้ทราบเลขลำดับโดยประมาณของโหนด $\hat{y}_{k_r}$ ที่สามารถให้บริการบนเส้นทางอื่น ๆ ได้จาก

$$\hat{y}_{kr} = m_{kr} - E(S_{kr}) \tag{18}$$

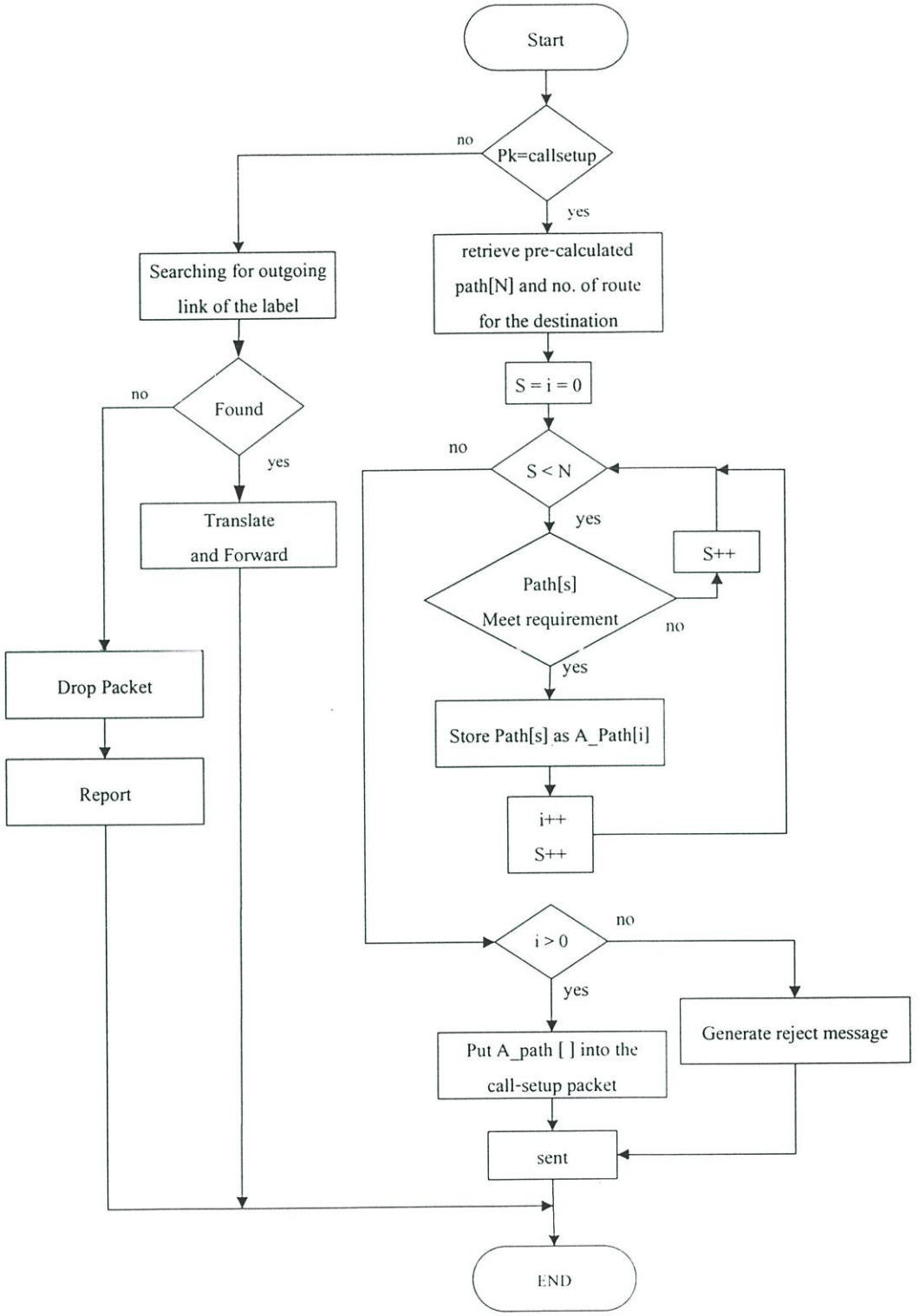
$$n_{kr+1} = N_{kr+1} - \sum_{i=1}^{r} y_{ki} - r + 1 \tag{19}$$

ซึ่งทำให้สามารถย้อนกลับไปหาค่า m_{kr} , S_{kr+1}, y_{kr+1} และ n_{kr+1} ในรอบถัดๆไปได้ ซึ่งโดยปกติแล้วก็มักจะพบว่า $n_{kr+1} \le n_{kr} \le n_{kr-1} \le \cdots \le N_{k1}$ (ยกเว้นในกรณีที่ $N_{kr+1} >> N_{kr} >> N_{kr-1}$ และค่า $P_{kr}(i)$ น้อยมาก ๆ)

ดังนั้นจึงทำให้ β_{kr} มีค่าน้อยลง เพราะถ้าจำนวนโหนดในเส้นทางน้อยลงย่อมทำให้โอกาสการบล็อกของทั้งเส้นทางลดลงด้วย

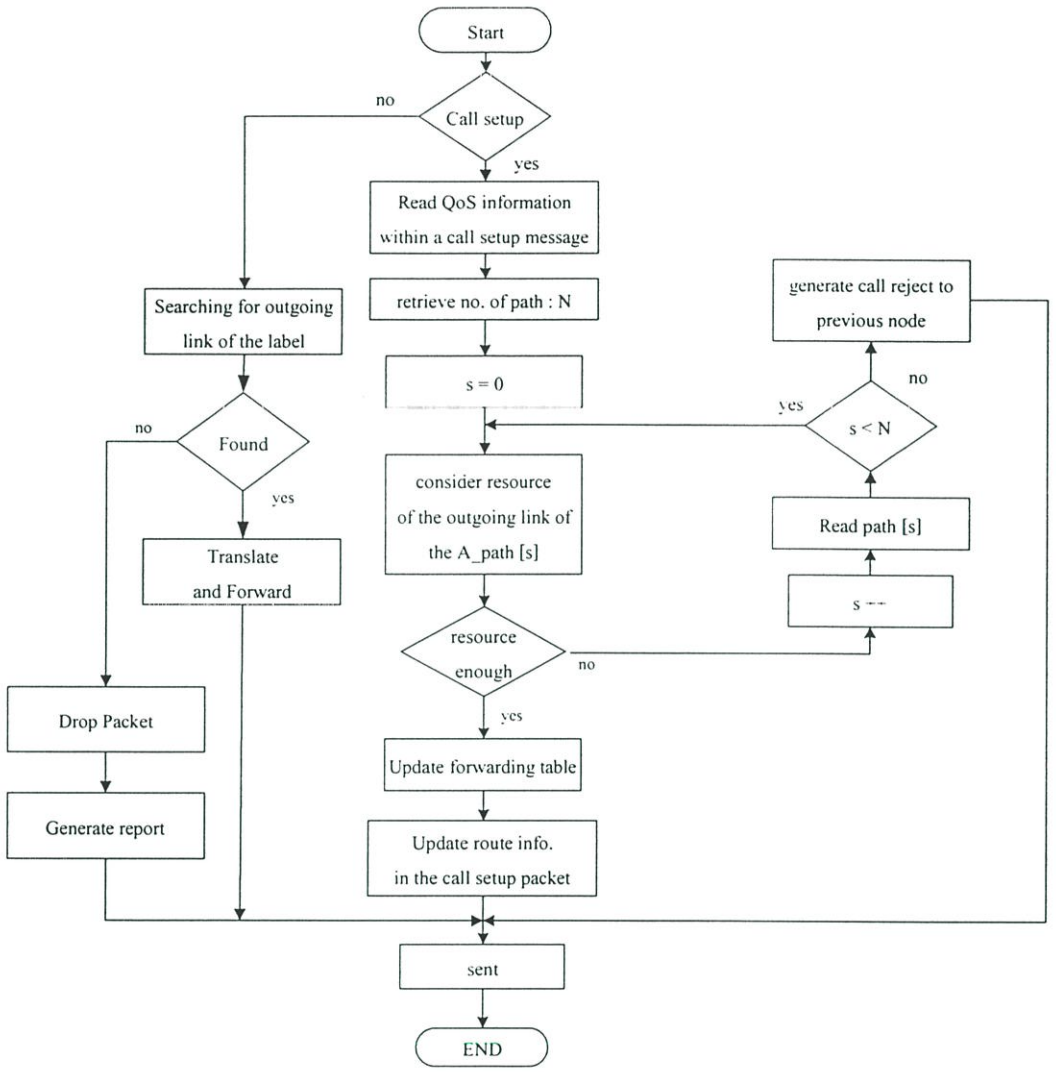
3.4 อัลกอริธึมการทำงานของเราเตอร์ LER และ LSR

3.4.1 อัลกอริธึมการทำงานของเราเตอร์ LER



รูปที่ 3.1 แสดง flow chart การทำงานของโหนด LER

3.4.2 อัลกอริธึมการทำงานของเราเตอร์ LSR



รูปที่ 3.2 แสดง flow chart การทำงานของโหนด LSR

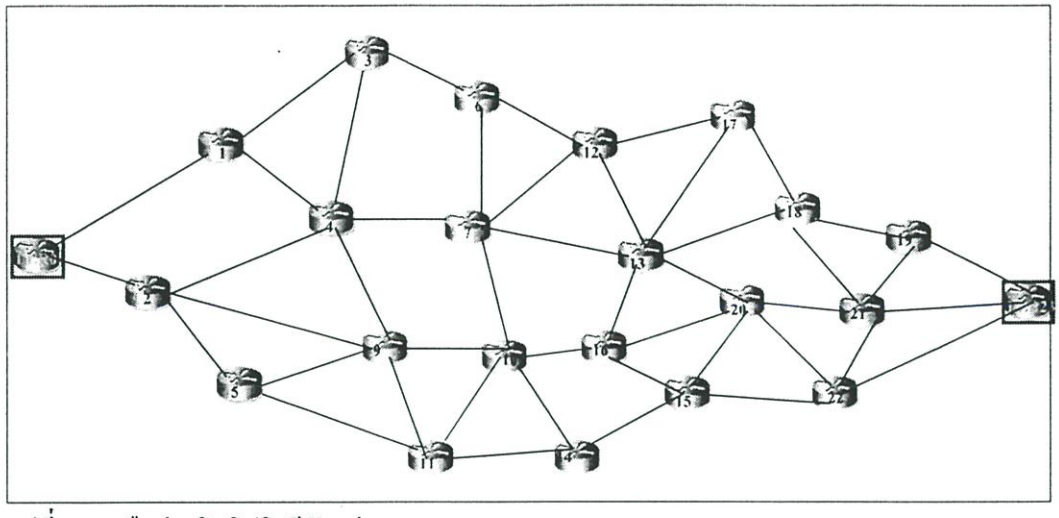
บทที่4

การทดลองและผลการทดลอง

4.1 วิธีการทดลอง

เนื่องจากหัวข้อการวิจัยนี้เป็นการเสนอวิธีการจัดตั้งเส้นทางแบบที่มีการแก้ไขเพิ่มเติมแนวความคิดให้สามารถแก้ไขปัญหาของการกำหนดเส้นทางผิดพลาดเมื่อการกำหนดเส้นทางนั้นเกิดขึ้นครั้งเดียวที่โหนดเราท์เตอร์ปากทางได้อย่างรวดเร็ว (ECQR) ซึ่งแต่เดิมนั้นความสามารถในการแก้ไขข้อผิดพลาดในการเลือกเส้นทาง (crankback) นั้นได้มีการนำเสนอและนำมาประยุกต์ใช้ในหลาย ๆ อัลกอริธึม แต่วิธีที่นำมาใช้ร่วมกับการหาเส้นทางแบบกำหนดเส้นทางครั้งเดียว (source routing) ในระบบเครือข่ายย่อยใด ๆ นั้น ต่างก็กำหนดให้กระบวนการแก้ไขข้อผิดพลาดนั้นย้อนกลับไปแก้ที่โหนดเราท์เตอร์ปากทาง ดังนั้นเพื่อให้สามารถที่จะเปรียบเทียบประสิทธิภาพของวิธีการของ ECQR กับวิธีการจัดตั้งเส้นทางแบบที่สามารถแก้ไขข้อผิดพลาดของการเลือกเส้นทางได้แบบที่มีการนำเสนอไว้แล้วนั้น จำเป็นที่จะต้องกำหนดแบบจำลองเครือข่ายขึ้นมาสำหรับศึกษาความแตกต่างของประสิทธิภาพของแนวทางทั้งสองในด้านของเวลาที่ต้องใช้ในการสร้างเส้นทาง โดยการทดลองในแต่ละครั้งจะมีการปรับค่าของโอกาสที่จะเกิดการบล๊อกของการสร้างเส้นทางจากน้อยไปมาก ซึ่งตามผลการทดลองที่แสดงในบทนี้นั้นเป็นการแสดงผลการทดลองเมื่อกำหนดให้โอกาสที่จะเกิดการบล๊อกโดยเฉลี่ยของทุกเส้นทางในระบบเครือข่ายในแต่ละครั้งของการจำลองผลการทดลองมีค่าเท่ากับ 0.72, 0.05, 0.18 และ 0.25 ตามลำดับ

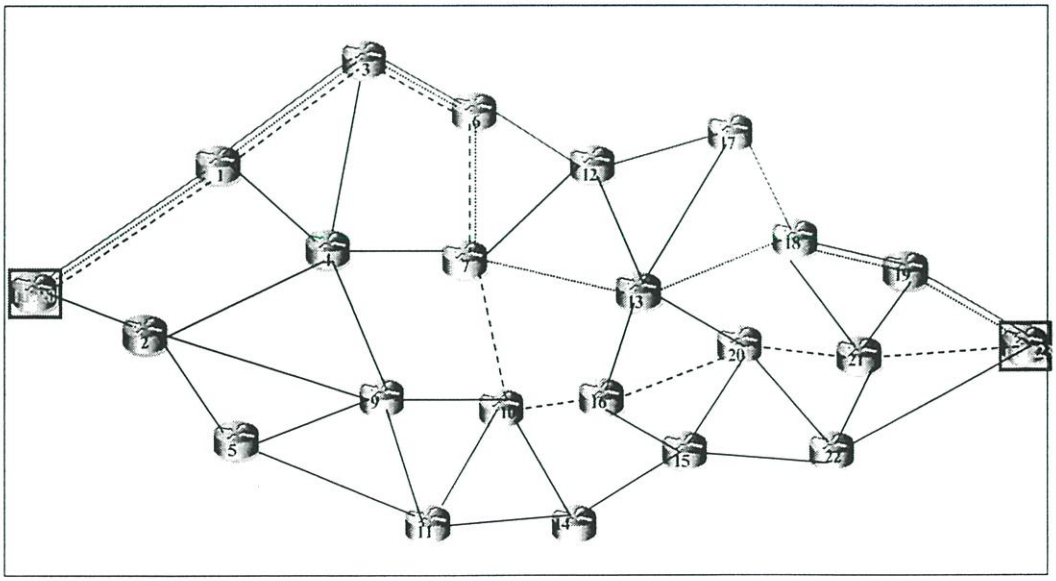
โดยในแบบจำลองระบบเครือข่ายที่กำหนดขึ้นเพื่อการทดลองครั้งนี้เป็นแบบจำลองของระบบเครือข่ายย่อยขนาดกลางที่ประกอบด้วย LSR หมายเลขที่ 1 ไปจนถึง LSR หมายเลข 22 มีทั้งสิ้น 22 ตัว และมี LER0 และ LER1 เป็นเราท์เตอร์ปากทางเข้าออก และเส้นทางที่เป็นไปได้จากโหนดต้นทางของระบบเครือข่ายจำลอง ซึ่งก็คือ LER0 ไปยังโหนดปลายทางของระบบเครือข่ายจำลองคือ LER1 นั้นจะต้องมีเส้นทางทั้งหมดที่เป็นไปได้ที่เหมาะสมกับ QoS ของการร้องขอการเชื่อมต่อใด ๆ อย่างน้อยที่สุด 3 เส้นทางด้วยกัน แบบจำลองของระบบเครือข่ายดังกล่าวได้ถูกแสดงไว้ดังรูปที่ 4.1 และตัวอย่างข้อมูลตารางเส้นทางได้แสดงไว้ดังตารางที่ 4.1 โดยโอกาสที่จะเกิดการบล็อกการร้องขอการเชื่อมต่อใด ๆ นั้นจะกำหนดไว้ตายตัวดังตารางที่ 4.2 สำหรับ scenario ที่ 1 และกำหนดไว้ดังตารางที่ 4.3 สำหรับ scenario ที่ 2, 3 และ 4 ของการจำลองผลการทดลอง



รูปที่ 4.1 เครือข่ายโทโปโลยีตัวอย่าง

ตารางที่ 4.1 แสดงตัวอย่างเส้นทางที่เป็นไปได้จาก LER0 ไปยัง LER1

ลำดับเส้นทางเผื่อเลือก	โหนดในเส้นทาง
1	LER0 →1 →3 →6 →12 →17 →18 →19 →LER1
2	LER0 →1 →3 →6 → 7 →13 → 18 →19 →LER1
3	LER0 →1 →3 →6 →7 →10 →16 →20 →21 →LER1



รูปที่ 4.2 ตัวอย่างเส้นทางที่เป็นได้สำหรับการเชื่อมต่อที่กำหนดขึ้น

ตารางที่ 4.2 แสดงค่าโอกาสที่จะเกิดการบล๊อก scenario1

ลำดับโหนดในเส้นทาง	ค่าโอกาสที่จะเกิดการบล๊อก	ลำดับโหนดในเส้นทาง	ค่าโอกาสที่จะเกิดการบล๊อก
LER0	0.05	16	0.05
1	0.24	17	0.18
3	0.18	18	0.18
6	0.18	19	0.24
7	0.05	20	0.18
10	0.05	21	0.05
12	0.05	LER1	-
13	0.05		

ตารางที่ 4.3 แสดงค่าโอกาสที่จะเกิดการบล๊อกของโหนดต่าง ๆ ใน scenario2 scenario3 และ scenario4

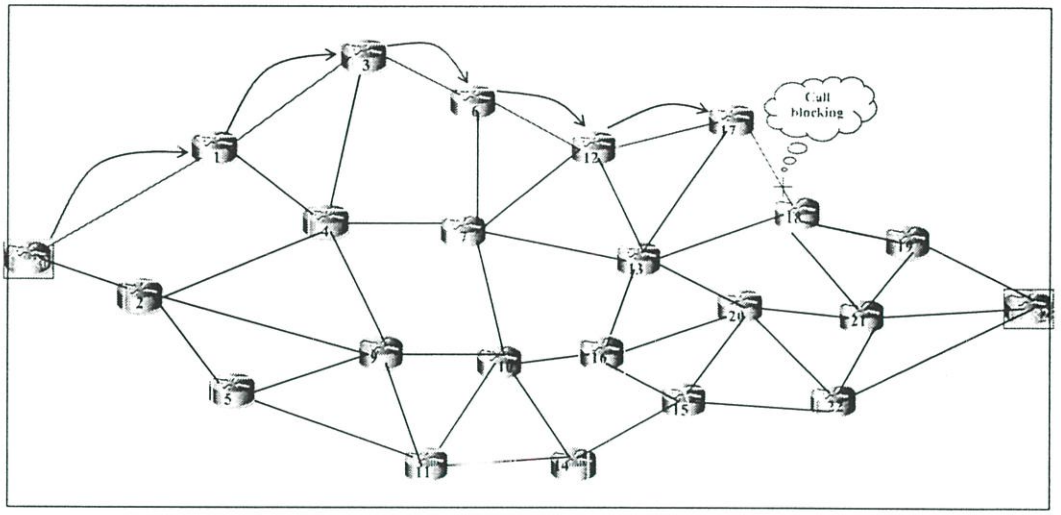
ลำดับโหนด	ค่าโอกาสที่จะเกิดการบล๊อก			ลำดับโหนด	ค่าโอกาสที่จะเกิดการบล๊อก		
	scenario2	scenario3	scenario4		Scenario2	scenario3	Scenario4
LER0	0.05	0.18	0.24	16	0.05	0.18	0.24
1	0.05	0.18	0.24	17	0.05	0.18	0.24
3	0.05	0.18	0.24	18	0.05	0.18	0.24
6	0.05	0.18	0.24	19	0.05	0.18	0.24
7	0.05	0.18	0.24	20	0.05	0.18	0.24
10	0.05	0.18	0.24	21	0.05	0.18	0.24
12	0.05	0.18	0.24	LER1	-	-	-
13	0.05	0.18	0.24				

ตารางที่ 4.4 แสดงค่าพารามิเตอร์ต่าง ๆ ที่ใช้ในการทดลอง

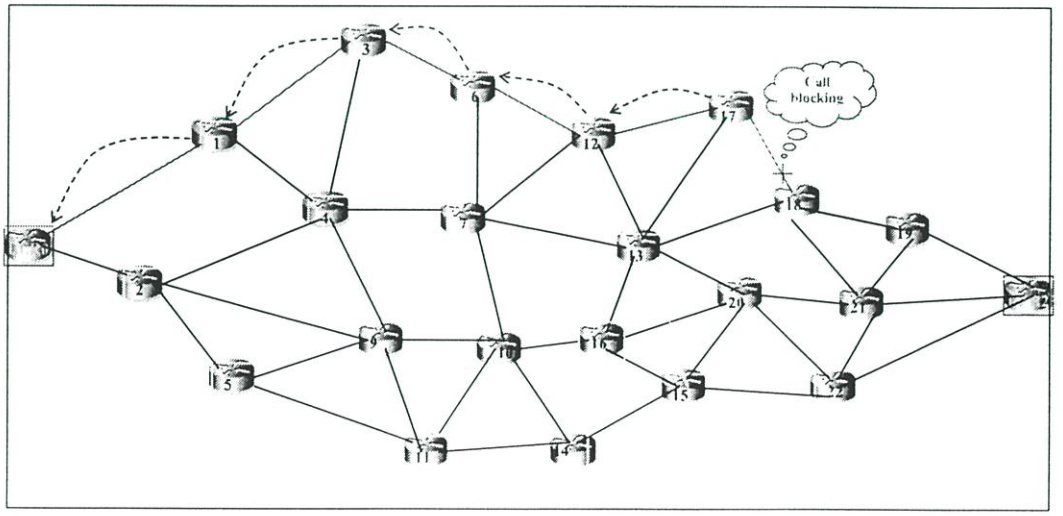
พารามิเตอร์	ค่าที่ใช้ในการทดลอง	รายละเอียด
τ_{ij}	constant	propagation delay ระหว่างการส่งจากโหนด i ไปยังโหนด j
δ_i	constant	transmission delay ของแพ็คเก็ตร้องขอการเชื่อมต่อบนสายสัญญาณของโหนด i
ρ_i	constant	คือ processing delay ในโหนด i
η_{k_r}	constant	คือเวลาที่ใช้ในการหาเส้นทางในโหนดปากทางรวมกับเวลาที่ใช้ในการปรับปรุงแพ็คเก็จ
ψ_j	constant	คือเวลาที่ใช้ในการประมวลผลแพ็คเก็จปฏิเสธการร้องขอเส้นทางเพื่อส่งย้อนกลับ
N^{k_r}		คือจำนวนโหนดตลอดเส้นทาง
n_{k_r}		คือจำนวนโหนดในเส้นทางใหม่
M_{k_r}		คือจำนวนโหนดที่เกิดการบล๊อก
m_{k_r}		คือจำนวนก้าวไปข้างหน้า
β_{k_r}	0.05, 0.18, 0.24	ค่าโอกาสที่จะเกิดการบล๊อก
$\overline{\beta}_k$		ค่าเฉลี่ยโอกาสที่จะเกิดการบล๊อก
S_{k_r}		คือจำนวนก้าวที่ก้าวถอยหลัง
y_{k_r}		หมายเลกลำดับของโหนด
F		จำนวนครั้งของการเกิดการบล๊อก
K		การเชื่อมต่อใดๆ
R		เส้นทางใดๆ
P		โอกาสในการให้บริการได้ในเส้นทาง

รูปที่ 4.3, 4.4, 4.5, 4.6 และ 4.7 แสดงขั้นตอนการทำงานของอัลกอริธึมการจัดตั้งเส้นทางแบบที่สามารถแก้ไขความผิดพลาดของการเลือกเส้นทางแบบกำหนดเส้นทางครั้งเดียวและการเลือกเส้นทางใหม่จะต้องกระทำที่โหนดปากทางเท่านั้น โดยในรูปที่ 4.3 นั้นเป็นการแสดงความพยายามในการจัดตั้งเส้นทางสำหรับการเชื่อมต่อที่กำหนดบนเส้นทางที่ 1 ในตารางที่ 4.1 ซึ่งการจัดตั้งเส้นทางในครั้งแรกนี้ได้รับการยอมรับและสามารถดำเนินการไปได้จนถึงโหนด LSR17 ซึ่งพบว่าไม่

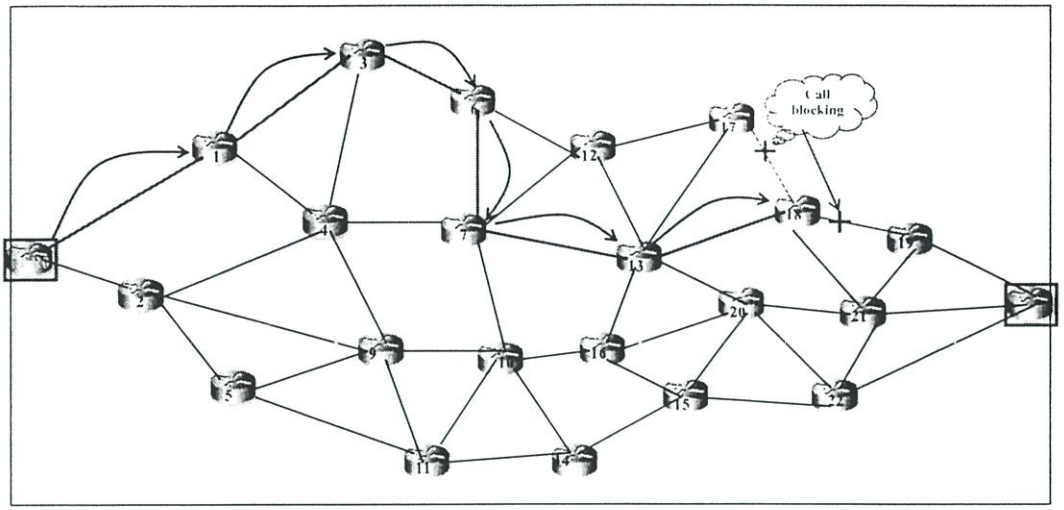
สามารถให้บริการได้ตามที่การร้องขอการเชื่อมต่อนี้ต้องการบนลิงค์ที่เชื่อมต่อระหว่างโหนด LSR17 และ LSR18 ซึ่งเป็นลิงค์หนึ่งบนเส้นทางที่กำหนด ดังนั้นกระบวนการสร้างเส้นทางนี้จึงต้องย้อนกระบวนการกลับไปยังโหนดปากทางโดยการส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางย้อนกลับมาจนถึงโหนด LER0 เพื่อให้กำหนดเส้นทางให้ใหม่ ดังแสดงไว้ในรูปที่ 4.4 ซึ่งเส้นทางใหม่นี้จะเป็นเส้นทางที่สองในตารางที่ 4.1 ที่ถูกนำมาใช้ในการจัดตั้งเส้นทางครั้งที่สองดังรูปที่ 4.5 ซึ่งก็ยังพบว่าเกิดการบล๊อกขึ้นที่โหนด LSR18 ที่พบว่าลิงค์ที่เชื่อมอยู่ระหว่างตัวมันไปยังโหนด LSR19 นั้นไม่สามารถให้บริการได้ กระบวนการแก้ไขข้อผิดพลาดของการเลือกเส้นทางก็จะเกิดขึ้นอีกครั้งดังรูปที่ 4.6 และการสร้างเส้นทางจะสำเร็จได้ดังกระบวนการสร้างเส้นทางที่แสดงด้วยรูปที่ 4.7



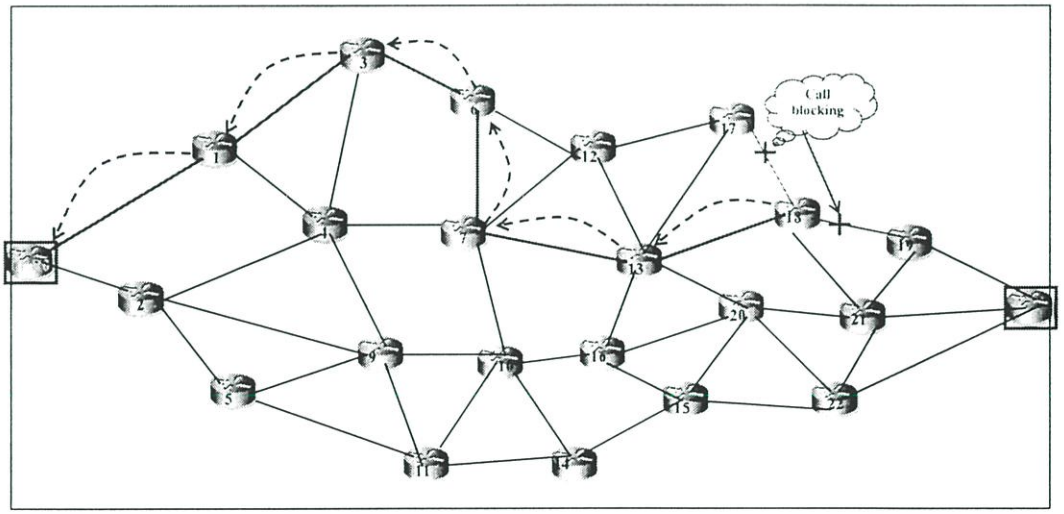
รูปที่ 4.3 การร้องขอเส้นทางเลือกที่ 1 ในกรณีที่เกิดการบล๊อก (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป)



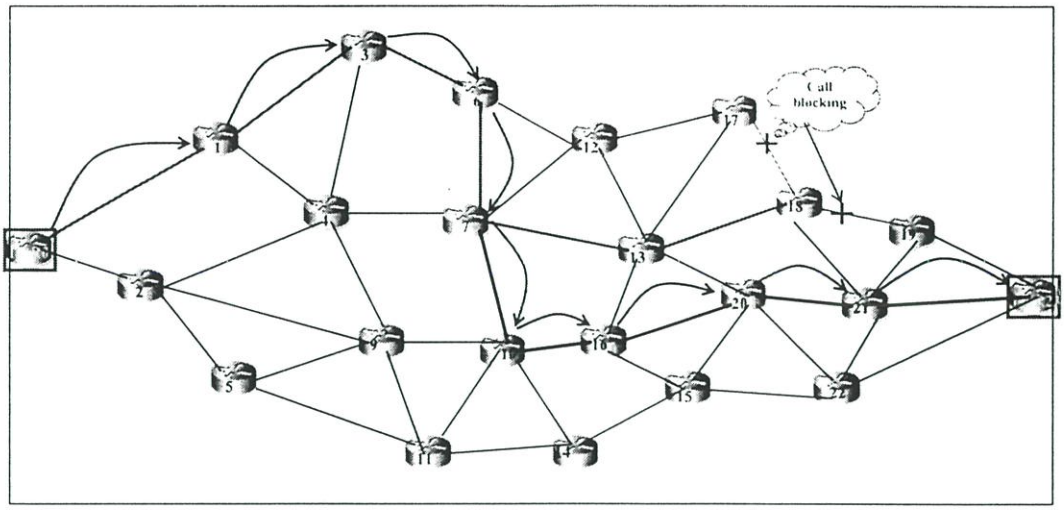
รูปที่ 4.4 การส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางย้อนกลับมาถึงโหนดปากทาง (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป)



รูปที่ 4.5 การร้องขอเส้นทางเลือกที่ 2 ในกรณีที่เกิดการบล๊อก (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป)

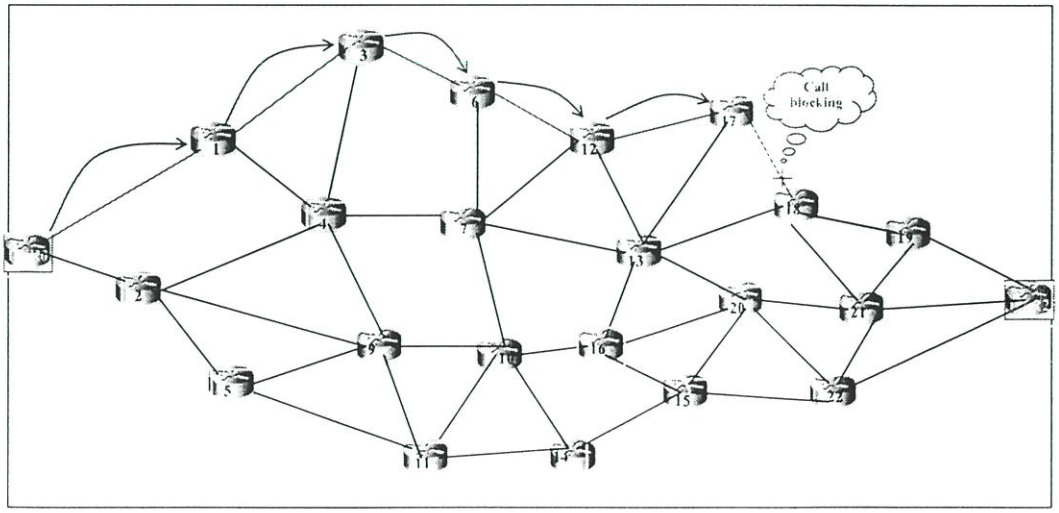


รูปที่ 4.6 การส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางที่ 2 ย้อนกลับมาถึงโหนดปากทาง (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป)

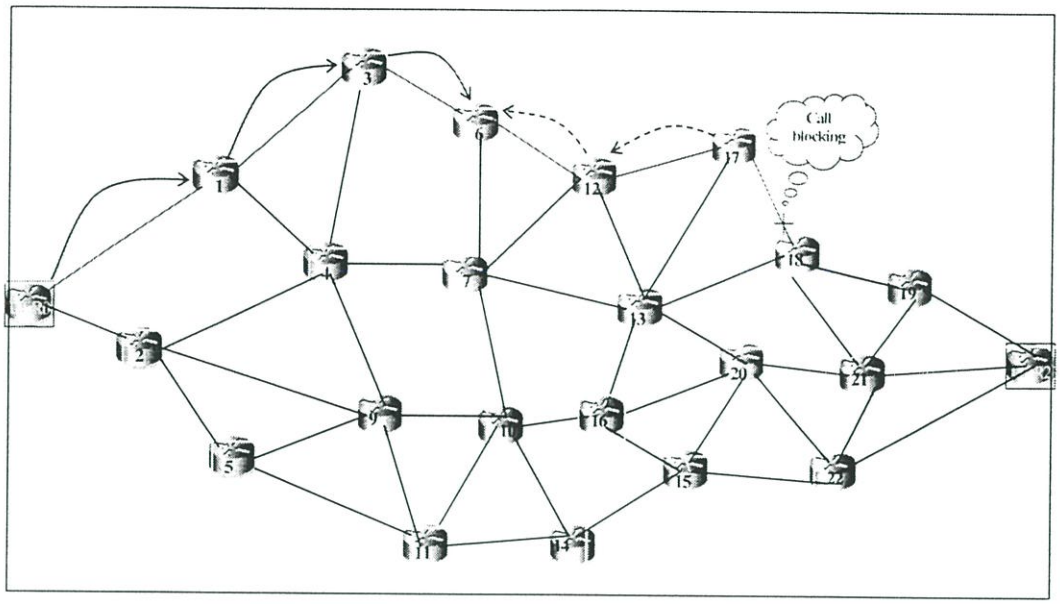


รูปที่ 4.7 การร้องขอการจัดตั้งเส้นทางบนเส้นทางที่ 3 ในกรณีที่สำเร็จ (ด้วยการใช้อัลกอริธึมการจัดตั้งเส้นทางแบบใช้ crank-back ทั่วไป)

ส่วนรูปที่ 4.8, 4.9, 4.10, 4.11 และ 4.12 นั้น ก็แสดงกระบวนการสร้างเส้นทางที่เกิดเหตุการณ์เช่นเดียวกันกับกรณีที่ได้อธิบายไปแล้วกล่าวคือจะเกิดความพยายามจัดตั้งเส้นทางในระบบเครือข่ายย่อยนี้ทั้งหมด 3 ครั้งบนเส้นทางที่ 1 เส้นทางที่ 2 และเส้นทางที่ 3 ตามลำดับ แต่ว่าการย้อนกระบวนการต่อไปนี้นั้นเป็นการดำเนินงานตามหลักการของอัลกอริธึม ECQR ดังนั้นจำนวน hop ที่เกิดขึ้นในกระบวนการแก้ไขข้อผิดพลาดของการสร้างเส้นทางจึงน้อยกว่าวิธีเดิมซึ่งได้นำเสนอไปแล้ว ทั้งนี้เนื่องจากโหนด LSR ทุกโหนดต่างก็สามารถที่จะใช้ประโยชน์จากข้อมูลเส้นทางเผื่อเลือกที่แนบมากับแพ็คเก็ตร้องขอเส้นทางได้

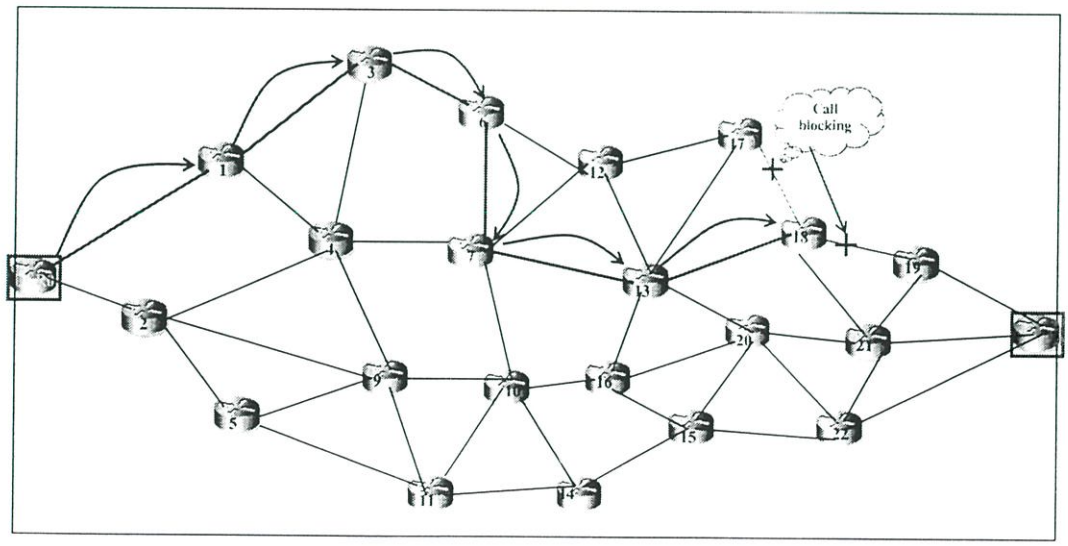


รูปที่ 4.8 การร้องขอเส้นทางเลือกที่ 1 ในกรณีที่เกิดการบล๊อก (ด้วยการใช้อัลกอริธึม ECQR)



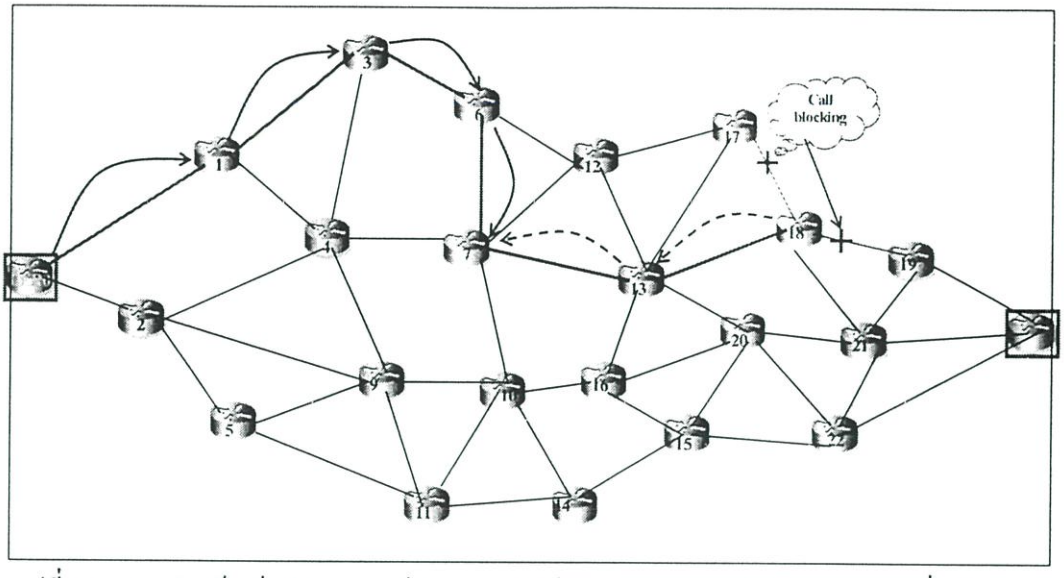
รูปที่ 4.9 การส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางย้อนกลับไปยังโหนด LSR6 ซึ่งเป็นโหนดที่เชื่อมระหว่างเส้นทางที่ 1 และเส้นทางที่ 2 (ด้วยอัลกอริธึม ECQR)

รูปที่ 4.9 คือการส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางย้อนกลับมาจนถึงโหนดพบว่ามีเส้นทางเผื่อเลือกอื่น ๆ ผ่านตัวมัน หากพบว่าสายสัญญาณขาออกของโหนดในเส้นทางที่กำหนดไม่สามารถรองรับความต้องการได้

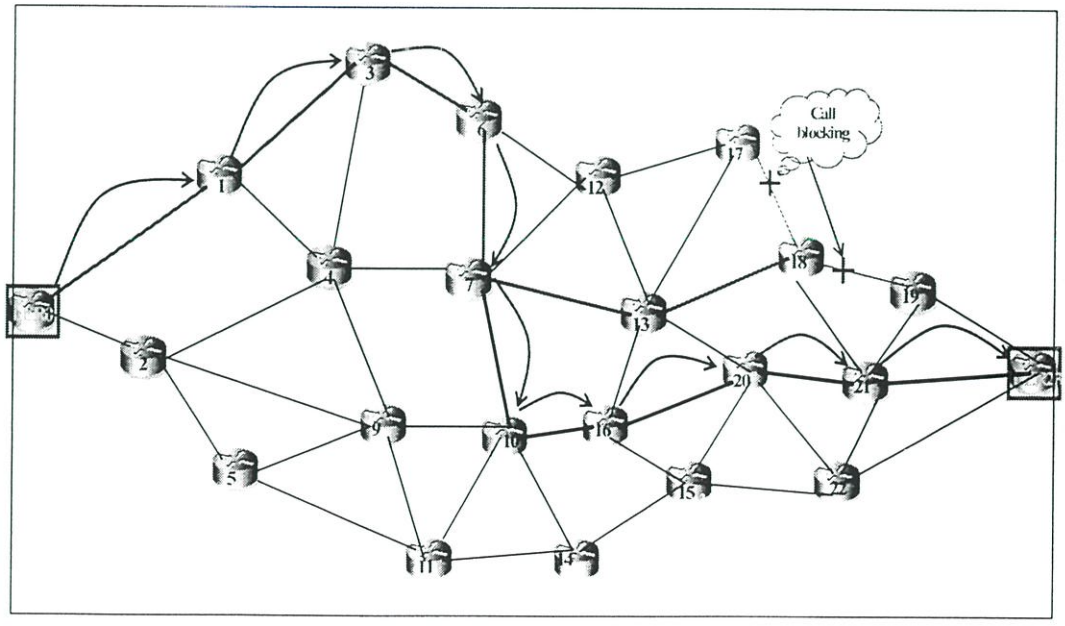


รูปที่ 4.10 การร้องขอเส้นทางเลือกที่ 2 ในกรณีที่เกิดการบล๊อก (ด้วยการใช้อัลกอริธึม ECQR)

รูปที่ 4.10 คือการร้องขออนุมัติเส้นทางครั้งที่ 2 โดยการใช้เส้นทางเผื่อเลือกที่ 2 ในกรณีที่ไม่สำเร็จ และเวลาที่ต้องใช้ในการร้องขออนุมัติเส้นทางจากโหนดที่พบว่ามีเส้นทางเผื่อเลือกผ่านตัวมันไปยังโหนดที่เกิดการบล๊อก



รูปที่ 4.11 การส่งแพ็กเก็จปฏิเสธการร้องขออนุมัติเส้นทางย้อนกลับไปยังโหนด LSR7 ซึ่งเป็นโหนดที่เชื่อมระหว่างเส้นทางที่ 2 และเส้นทางที่ 3 (ด้วยอัลกอริธึม ECQR)



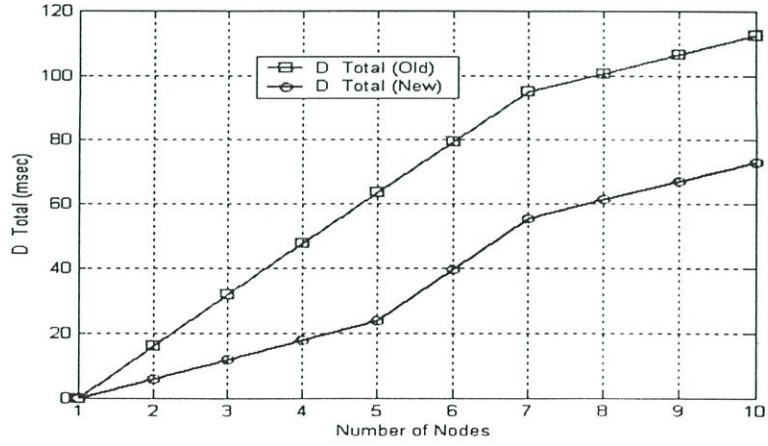
รูปที่ 4.12 การร้องขอเส้นทางเลือกที่ 3 ในกรณีที่สำเร็จ (ด้วยอัลกอริธึม ECQR)

4.2 ผลการทดลอง

ผลการทดลองที่เกิดจากการจำลองผลดังข้อกำหนดของโทโปโลยีตามรูปที่ 4.1 และค่าโอกาสของการเกิดการบล๊อกดังตารางที่ 4.2 ถูกแสดงไว้ในตารางที่ 4.6, 4.7, 4.8, และ 4.9 ตามลำดับโดยแสดงให้เห็นการเปรียบเทียบเวลาทั้งหมดที่อาจจะต้องสูญเสียไปในการจัดตั้งเส้นทางด้วยอัลกอริธึมแบบเดิมและอัลกอริธึม ECQR ส่วนการแสดงผลแบบกราฟสำหรับผลการจำลองแต่ละตารางนั้นแสดงไว้ดังรูปที่ 4.9, 4.10 , 4.11 และ 4.12

ตารางที่ 4.5 แสดงค่าเวลาการเปรียบเทียบจากวิธีเดิมและวิธีที่นำเสนอ (ECQR) โดยกำหนดให้ค่าโอกาสที่จะเกิดการบล๊อกเท่ากับ 0.05, 0.18 และ 0.25

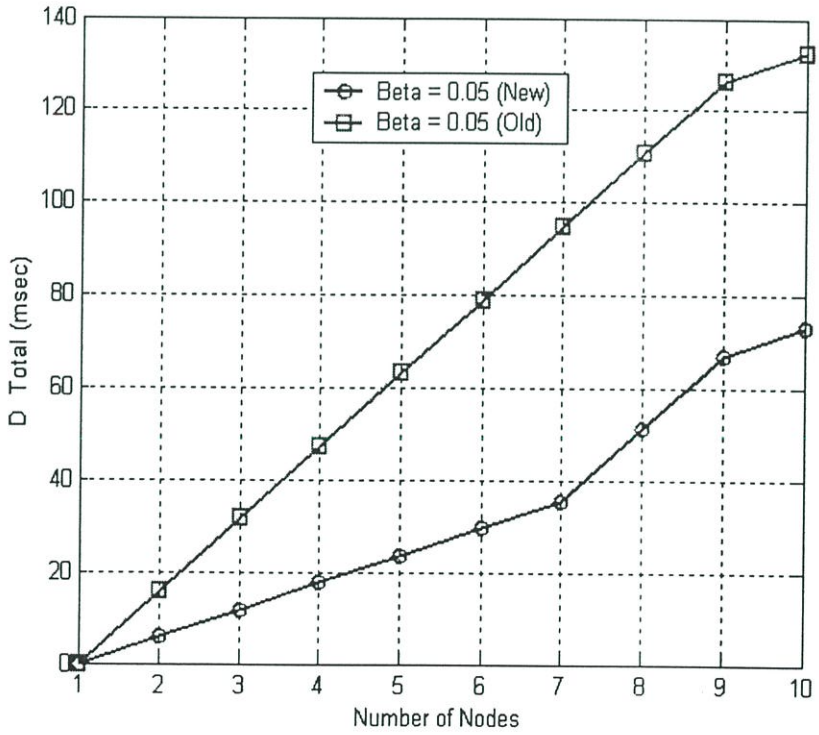
ลำดับโหนด	ค่าโอกาสที่จะเกิดการบล๊อก	D_TotalOld	D_TotalNew
1	0.05	0	0
2	0.24	25.70	5.90
3	0.18	51.40	11.80
4	0.18	77.10	17.70
5	0.05	102.80	23.60
6	0.05	128.50	39.40
7	0.05	144.30	45.30
8	0.18	160.10	61.10
9	0.05	166.00	79.90
10	-	171.90	82.80



รูปที่ 4.13 แสดงการเปรียบเทียบค่า D_total ระหว่างวิธี ECQR (D_total (New)) และวิธีที่มีอยู่เดิม (D_total (Old))

ตารางที่ 4.6 แสดงค่าเวลาการเปรียบเทียบจากวิธีเดิมและวิธีที่นำเสนอ (ECQR) โดยกำหนดให้ค่าโอกาสที่จะเกิดการบล๊อกเท่ากับ 0.05 ตลอดทั้งเส้นทาง

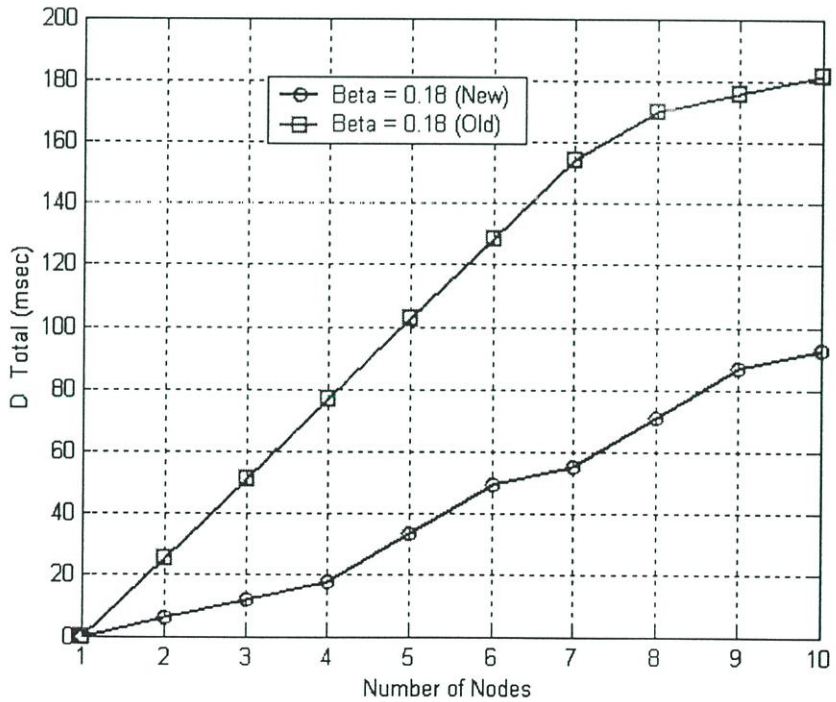
ลำดับโหนด	ค่าโอกาสที่จะเกิดการบล๊อก	D_Total(Old)	D_Total(New)
1	0.05	0	0
2	0.05	15.80	5.90
3	0.05	31.60	11.80
4	0.05	47.40	17.70
5	0.05	63.20	23.60
6	0.05	79.00	29.50
7	0.05	94.80	35.40
8	0.05	110.70	51.20
9	0.05	126.40	67.00
10	-	132.30	72.90



รูปที่ 4.14 แสดงการเปรียบเทียบเวลาหน่วง ในการสร้างเส้นทางในระบบเครือข่ายที่มีค่า $\beta_{k_r}(i) = 0.05$

ตารางที่ 4.7 แสดงค่าเวลาการเปรียบเทียบจากวิธีเดิมและวิธีที่นำเสนอ (ECQR) โดยกำหนดให้ค่าโอกาสที่จะเกิดการบล๊อกเท่ากับ 0.18 ตลอดทั้งเส้นทาง

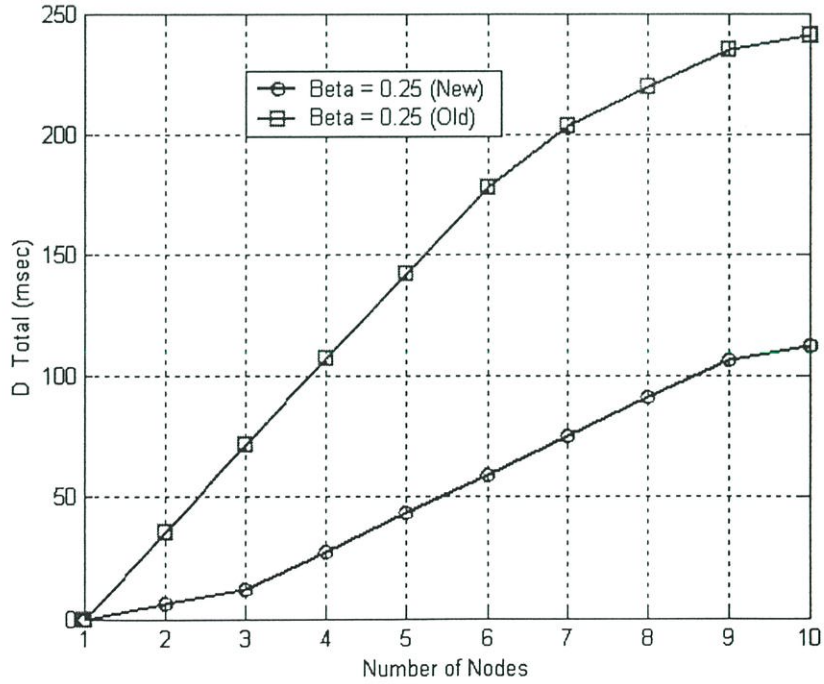
ลำดับโหนด	ค่าโอกาสที่จะเกิดการบล๊อก	D_Total(Old)	D_Total(New)
1	0.18	0	0
2	0.18	25.70	5.90
3	0.18	51.40	11.80
4	0.18	77.10	17.70
5	0.18	102.80	33.50
6	0.18	128.50	49.30
7	0.18	1ถ4.20	55.20
8	0.18	170.00	71.20
9	0.18	175.90	86.80
10	-	181.80	92.70



รูปที่ 4.15 แสดงการเปรียบเทียบเวลาหน่วง ในการสร้างเส้นทางในระบบเครือข่ายที่มีค่า $\beta_{k_r}(i) = 0.18$

ตารางที่ 4.8 แสดงค่าเวลาการเปรียบเทียบจากวิธีเดิมและวิธีที่นำเสนอ (ECQR) โดยกำหนดให้ค่าโอกาสที่จะเกิดการบล๊อกเท่ากับ 0.24 ตลอดทั้งเส้นทาง

ลำดับโหนด	ค่าโอกาสที่จะเกิดการบล๊อก	D_Total(Old)	D_Total(New)
1	0.24	0	0
2	0.24	35.60	5.90
3	0.24	71.20	11.80
4	0.24	106.80	27.60
5	0.24	142.40	43.40
6	0.24	178.00	59.20
7	0.24	203.70	75.00
8	0.24	219.50	90.80
9	0.24	235.30	106.60
10	-	241.20	112.50



รูปที่ 4.16 แสดงการเปรียบเทียบเวลาหน่วง ในการสร้างเส้นทางในระบบเครือข่ายที่มีค่า $\beta_{k_r}(i) = 0.25$

ตารางที่ 4.9 แสดงถึงผลรับของการเปรียบเทียบเป็นเปอร์เซนต์ของทั้งสองวิธี

ลำดับ	ค่าโอกาสเฉลี่ยที่จะเกิดการบล๊อก	ค่าเฉลี่ย D_Total(Old)	ค่าเฉลี่ย D_Total(New)	เปอร์เซ็นต์การเปรียบเทียบ
1	0.05	77.90	35	45%
2	0.18	118.6	47	40%
3	0.24	159.3	59	37%

บทที่5
บทสรุปและแนวทางการพัฒนา

5.1 สรุปผลการทดลอง

ในงานวิจัยนี้ได้นำเสนอแนวทางในการปรับปรุงอัลกอริธึมเลือกเส้นทางในเครือข่ายย่อยให้มีความสามารถในการแก้ไขข้อผิดพลาดของการเลือกเส้นทาง ด้วยการแนบเส้นทางเผื่อเลือกไปกับแพ็คเก็จร้องขอการเชื่อมต่อ โดยเส้นทางเผื่อเลือกนี้สามารถจะแนบไปได้มากกว่าหนึ่งเส้นทาง ถ้าโหนดที่กำลังพิจารณาแพ็คเก็จร้องขอการเชื่อมต่ออยู่นั้นพิจารณาเห็นว่าสายสัญญาณขาออกที่ถูกระบุไว้ในเส้นทางหลักนั้นรองรับความต้องการได้ ก็จะทำการส่งต่อแพ็คเก็จร้องขอเส้นทางนั้นไปยังโหนดถัดไปหรือเลือกใช้เส้นทางเผื่อเลือกอื่น ๆ ที่ผ่านตัวมัน ในกรณีที่สายสัญญาณในเส้นทางหลักไม่สามารถรองรับได้ หรือใช้การส่งแพ็คเก็จปฏิเสธการร้องขอเส้นทางย้อนกลับไปยังโหนดก่อนหน้าเพื่อให้พิจารณาหาเส้นทางที่เป็นไปได้เส้นทางอื่น ๆ จากข้อมูลเส้นทางที่เหลือจากการตัดทอนเอาเส้นทางที่เป็นไปไม่ได้ออกแล้ว

จากแนวคิดดังกล่าวถึงแม้จะเป็นการเพิ่มจำนวนข้อมูลเส้นทางลงในแพ็คเก็จร้องขอการเชื่อมต่อและเป็นการเพิ่มภาระในการประมวลผลใน LSR แต่จากการศึกษาด้วยการจำลองผล ซึ่งกำหนดค่าพารามิเตอร์ให้สอดคล้องใกล้เคียงกับความเป็นจริงแล้ว ปรากฏว่าผลจำลองการทำงานแสดงให้เห็นอย่างเด่นชัดว่าแนวคิดใหม่ในการจัดตั้งเส้นทางแบบ ECBQR สามารถลดเวลาในการสร้างเส้นทางได้มากกว่า 30 ถึง 70 เปอร์เซ็นต์ เมื่อเทียบกับกระบวนการสร้างเส้นทางแบบกำหนดเส้นทางครั้งเดียวทั่ว ๆ ไปสำหรับระบบเครือข่ายที่มีขนาดกลางจนถึงระบบเครือข่ายที่มีขนาดใหญ่ในขณะที่มีสภาวะการใช้งานอย่างหนาแน่นปานกลางไปจนสภาวะการใช้งานมีความหนาแน่นสูง

5.2 แนวทางการพัฒนาต่อไป

อัลกอริทึมที่นำเสนอในงานวิจัยนี้ เป็นอัลกอริธึมการสร้างเส้นทางแบบคำนึงถึงคุณภาพของการให้บริการสำหรับระบบเครือข่าย MPLS ที่มีความสามารถในการแก้ไขข้อบกพร่องในการเลือกเส้นทางได้อย่างรวดเร็ว เนื่องจากวิธี ECBQR นี้เป็นวิธีที่ใช้การสร้างเส้นทางแบบกำหนดเส้นทางครั้งเดียว ซึ่งในการวิจัยครั้งนี้ได้เน้นไปที่การประเมินค่าโดยประมาณของเวลาที่ต้องใช้ในการสร้างเส้นทางเมื่อระบบเครือข่ายมีค่าความน่าจะเป็นของการเกิดการบล๊อก (call blocking) ที่เปลี่ยนแปลงไปเท่านั้น แต่การตรวจวัดประสิทธิภาพของวิธี ECBQR นี้อาจจะจำเป็นต้องการการศึกษาผลกระทบอื่น ๆ ที่อาจจะเกิดขึ้นได้ถ้าหากนำวิธีการนี้ไปใช้งาน เช่น อัตราการเพิ่มขึ้นของทราฟฟิกที่เกิดจากขนาดของแพ็คเก็จร้องขอเส้นทางที่เพิ่มมากขึ้น และน่าจะมีการพัฒนาแบบจำลองเพื่อวัดผล

การทำงานในสภาวะการทำงานของระบบเครือข่ายที่มีการเปลี่ยนแปลงอยู่เสมอด้วย และควรจะมีการเปรียบเทียบกับวิธีการหาเส้นทางแบบกระจายเพื่อใช้ในการพิจารณาข้อดีและข้อเสียที่สามารถเปรียบเทียบกันได้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายสมพร กันทะวงศ์
วันเดือนปีเกิด	02 กุมภาพันธ์ 2507.
สถานที่เกิด	จังหวัดไชยะบูรี ประเทศลาว
วุฒิการศึกษาระดับปริญญาตรี	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
สถานที่สำเร็จการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีที่สำเร็จการศึกษา	2544.
ผลงานวิจัยที่ได้รับการตีพิมพ์	

1. สมพร กันทะวงศ์, มยุรี เลิศเวชกุล, ประกิจ ตังติสานนท์. " อัลกอริธึมสร้างเส้นทางแบบคำนึงถึงคุณภาพของการให้บริการที่มีความสามารถในการตอบสนองต่อความผิดพลาดของการกำหนดเส้นทาง."วิศวสาร ลาดกระบัง คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. ปีที่ 20, ฉบับที่ 2, มิถุนายน 2546. หน้า 94 - 99

2. S. Kanthavong, M. Lertwatechakul and P. Tangtisanon " QoS Based Routing Algorithm with Crank-Back Ability." International Conference on Control, Automation and System (ICCAS' 20 03) , pp.2413 –2417, 23-25 Oct 2003