การเปรียบเทียบประสิทธิภาพของโครงข่าย MPLS/VPN กับโครงข่าย IP แบบดั้งเดิม

PERFORMANCE COMPARISION OF MPLS/VPN NETWORK VERSUS TRADITIONAL IP NETWORK

บดินทร์ จิ๋วแย้ม

BORDIN JIWYAM

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2552

KMITL 2004-EN-M-010-146

PERFORMANCE COMPARISION OF MPLS/VPN NETWORK VERSUS TRADITIONAL IP NETWORK

BORDIN JIWYAM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN TELECOMMUNICATIONS ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2009

KMITL 2009-EN-M-010-146

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การเปรียบเทียบประสิทธิภาพของโครงข่าย MPLS/VPN กับโครงข่าย IP แบบดั้งเดิม

Thesis Title Performance Comparison of MPLS/VPN Networks Versus Traditional IP Network

นักศึกษา นายบดินทร์ จิ๋วแย้ม

รหัสประจำตัว 48060938

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมโทรคมนาคม

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.กอบชัย เดชหาญ

หมายเลขวิทยานิพนธ์ KMITL-2009-EN-M-010-146

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.สมยศ	จุณณะปิยะ	
ดร.สิรภพ	ตู้ประกาย	
รศ.จิระศักดิ์	ชาญวุฒิธรรม	
รศ.ดร.ฟูศักดิ์	ชีวสุวิทย์	
รศ.ดร.กอบชัย	เดชหาญ	

วัน / เดือน/ ปี ที่สอบ วันศุกร์ที่ 9 ตุลาคม พ.ศ. 2552 เวลา 13.30-15.30 น.

สถานที่สอบ ณ อาคาร A ชั้น 3 ห้องประชุม 1

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(รองศาสตราจารย์ ดร.กอบชัย เดชหาญ)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 9 ตุลาคม พ.ศ. 2552

สำนักทะเบียนและประมวลผล สจล.
วันที่ส่งเล่มวิทยานิพนธ์ฉบับสมบูรณ์
วันที่ 26 **เดือน** ต.ค. **พ.ศ.** 52
ลงชื่อ.....

หัวข้อวิทยานิพนธ์	การเปรียบเทียบประสิทธิภาพของโครงข่าย MPLS/VPN กับโครงข่าย IP แบบดั้งเดิม
นักศึกษา	นายบดินทร์ จิ๋วแย้ม
รหัสนักศึกษา	48060938
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมโทรคมนาคม
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.กอบชัย เดชหาญ

บทคัดย่อ

บทความนี้ได้ทำการศึกษาและวิเคราะห์เปรียบเทียบประสิทธิภาพระหว่างโครงข่าย MPLS/VPN กับโครงข่าย IP แบบดั้งเดิม ในด้านความน่าเชื่อถือของระบบ (Reliability) และความสามารถในด้านคุณภาพการบริการ (Quality of Service) โดยใช้เทคนิคการทำ MPLS Traffic Engineering (TE) Fast Reroute และ MPLS Differentiated Service (DiffServ) แทนการทำงานของ OSPF Routing Protocol และ Best Effort Protocol โดยทำการทดสอบบนโครงข่ายจริงที่ระดับความเร็ว 10 Gbps

Thesis Title	Performance Comparison of MPLS/VPN Network Versus Traditional IP Network
Student	Mr. Bordin Jiwyam
Student ID.	48060938
Degree	Master of Engineering
Program	Telecommunications Engineering
Year	2009
Thesis Advisor	Assoc. Prof. Dr. Kobchai Dejhan

ABSTRACT

This paper studies and analyzes a comparative performance of MPLS/VPN network versus traditional IP network such as reliability and quality of service by means of MPLS traffic engineering (TE). Fast Reroute and MPLS Differentiated Service (MPLS DiffServ) will replace OSPF Routing protocol and best effort protocol by test on real network of 10 Gbps

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก รศ.ดร. กอบชัย เดชหาญ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณ คณะวิศวกรรมศาสตร์ที่ให้ความช่วยเหลือ ในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

[บดินทร์ จิ๋วแย้ม]

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญตาราง(ต่อ)

ตารางที่ หน้า

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเครือข่ายคอมพิวเตอร์ได้มีบทบาทต่อชีวิตประจำวันมากขึ้นทุกขณะ การเจริญเติบโตของเครือข่ายคอมพิวเตอร์เหล่านี้เป็นไปอย่างต่อเนื่อง และยังไม่มีสัญญาณบ่งบอกว่าจะมีการชะลอตัวแต่อย่างใด เครือข่ายแบบท้องถิ่นในองค์กรต่างๆ ตลอดจน บริษัท สถานศึกษาส่วนใหญ่กว่า 80 % จะนิยมใช้เครือข่ายอีเทอร์เน็ท ซึ่งโครงข่ายหลักใช้เทคโนโลยี Internet Protocol แบบดั้งเดิม ด้วยความต้องการการส่งผ่านข้อมูลที่เพิ่มขึ้นอย่างรวดเร็วตามขนาดและจำนวนของเครื่องคอมพิวเตอร์ที่ต่ออยู่บนเครือข่าย ตลอดจนการเติบโตของอินเตอร์เน็ทอย่างรวดเร็วและโปรแกรมประยุกต์ต่างๆที่ต้องการรับส่งข้อมูลแบบ Real Time มากขึ้นเช่น VoIP IPTV , VDO Conference โครงข่ายที่ใช้เทคโนโลยี Internet Protocol แบบดั้งเดิม เริ่มจะไม่สามารถตอบสนองความต้องการของผู้ใช้ได้อย่างมีประสิทธิภาพ จึงจำเป็นต้องสร้างโครงข่ายมารองรับปริมาณข้อมูลที่สูงขึ้นสามารถให้บริการการรับส่งข้อมูลแบบ Real Time มีสเถียรภาพและความน่าเชื่อถือได้ดีมากขึ้น

Multiprotocol Label Switching (MPLS) เป็นเทคโนโลยีที่เริ่มมีการใช้งานกันอย่างกว้างขว้างทั้งใน Internet Service Providers (ISP),Telecommunication Carriers และองค์กรชั้นนำทั่วไปซึ่ง MPLS เป็นเทคโนโลยีที่นำมาแก้ปัญหาที่เกิดขึ้นในปัจจุบันของระบบเครือข่ายเช่น ความเร็ว (speed),ขนาด(scalability),การบริหารคุณภาพการให้บริการ (Quality of Service Management) และการควบคุมการจราจร(Traffic Engineering) ผู้ให้บริการ Internet (ISP),Telecommunication Carriers รวมทั้งองค์กรชั้นนำทั่วไปได้นำเอาเทคโนโลยี MPLS มาประยุกต์ใช้งานในด้านต่างๆเช่น โครงข่ายเสมือนส่วนบุคคล (Virtual Private Network:VPN) Traffic Engineering และการควบคุมคุณภาพการให้บริการ (Quality of Service : QoS) เพื่อรับประกันคุณภาพการให้บริการสำหรับข้อมูลประเภท Voice Video และ Application ที่ต้องการความมีเสถียรภาพของข้อมูล ดังที่ได้กล่าวมาข้างต้นจึงเห็นได้ว่า MPLS เหมาะแก่การนำมาใช้เป็นเครือข่ายหลักสำหรับการสื่อสาร Internet Protocol (IP)

Virtual Private Network (VPN) ได้เริ่มเป็นที่รู้จักและนำมาใช้งานในลักษณะที่เรียกว่าวงจรเช่า (leased line) ให้บริการในลักษณะ point-to-point ระหว่างสำนักงานของผู้ใช้บริการโดยผ่านเครือข่ายของผู้ให้บริการ(Service Provider : SP) โดย Frame Relay และ ATM เป็นเทคโนโลยีแรกที่นำมาใช้เพื่อให้บริการ VPN ซึ่งหลังจากได้มีการนำเทคโนโลยี MPLS มาใช้งานการให้บริการ VPN แบบใหม่จึงเกิดขึ้นโดย MPLS–based VPN สามารถแบ่งได้เป็น 3 ลักษณะคือ

- Layer 3 multipoint VPNs หรือ Internet Protocol (IP) VPNs
- Layer 2 point – to – point VPNs หรือ Virtual Leased Lines (VLL)
- Layer 2 multipoint VPNs หรือ Virtual Private LAN Services (VPLS)

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้มุ่งหวังเพื่อศึกษาและวิเคราะห์เปรียบเทียบประสิทธิภาพระหว่างโครงข่าย MPLS/VPN กับโครงข่าย Internet Protocol แบบดั้งเดิม ดังนั้นในวิทยานิพนธ์นี้จึงเสนอวิธีการทดสอบประสิทธิภาพความความเชื่อถือได้ของระบบและความสามารถการควบคุมคุณภาพการให้บริการ (Quality of Service:QoS) โดยใช้หลักการของ MPLS Traffic Engineering (TE) Fast Reroute , และ MPLS Differentiated Service (MPLS DiffServ) ซึ่งสามารถที่จะช่วยให้ระบบมีประสิทธิภาพที่ดีขึ้น

1.3 สมมติฐานของการศึกษา

ข้อด้อยของโครงข่าย IP แบบดั้งเดิม (Traditional IP Network) คือความเชื่อถือได้ของระบบในกรณีเส้นทางของเคเบิลใยแก้วนำแสงที่เชื่อมโยงระหว่างอุปกรณ์ในโครงข่ายเกิดความบกพร่องและความสามารถในการจัดการคุณภาพการบริการ (Quality of Service) ในกรณีที่โครงข่ายเกิดความคับคั่งของข้อมูล ทั้งสองกรณีนี้โครงข่าย IP แบบดั้งเดิมจะเกิดการสูญเสียข้อมูลและมีค่า Recovery Time ที่ใช้ในการเปลี่ยนเส้นทางที่มากและทำให้การรับส่งข้อมูลช้าจนเป็นเหตุให้ระบบโครงข่ายล่มชั่วขณะ ไม่สามารถให้บริการงานที่ต้องการความเชื่อถือของระบบที่สูง เช่น ระบบ VoIP ระบบ VDO Conference และระบบที่ต้องการรับส่งข้อมูลแบบ real time จากกรณีดังกล่าวทำให้โครงข่าย IP แบบดั้งเดิมด้อยประสิทธิภาพลงมากอย่างเห็นได้ชัด

การแก้ปัญหาข้างต้นนี้ในวิทยานิพนธ์เล่มนี้เราจะใช้โครงข่าย MPLS/VPN แทนโครงข่าย IP แบบดั้งเดิมและใช้เทคนิคการทำ MPLS Traffic Engineering (TE) Fast Reroute และ MPLS Differentiated Service (MPLS DiffServ) ในการปรับปรุงค่า Recovery Time , Delay Time และการสูญเสียข้อมูลให้มีค่าลดลง

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

วิธีการเปรียบเทียบประสิทธิภาพของโครงข่าย MPLS/VPN กับโครงข่าย IP แบบดั้งเดิม ลักษณะเด่นของวิธีการที่นำเสนอในวิทยานิพนธ์นี้คือการใช้เทคโนโลยี MPLS/VPN ร่วมกับการทำ MPLS Traffic Engineering (TE) Fast Reroute เพื่อใช้ในการควบคุมการไหลของข้อมูล (Traffic Flow) ในการให้บริการบนโครงข่ายคอมพิวเตอร์ที่มีปริมาณของ Traffic ที่คับคั่งโดยการ

หาเส้นทางสำรองที่ดีที่สุด ซึ่งระบบจะทำงานโดยอัตโนมัติในกรณีที่สายเคเบิลใยแก้วนำแสงเส้นทางหลักชำรุด ส่วนการทำ MPLS DiffServ จะช่วยควบคุมคุณภาพการส่งข้อมูลในกรณีที่โครงข่ายเกิดความคับคั่ง (Network Congested) ทำให้การส่งข้อมูลประเภท Voice , Video ยังคงสามารถส่งข้อมูลต่อไปได้โดยไม่กระทบต่อคุณภาพการให้บริการ ซึ่งในวิทยานิพนธ์เล่มนี้ได้แสดงผลของการทดสอบด้วยวิธีดังที่ได้กล่าวมา บนโครงข่ายจริงที่ระดับความเร็ว 10 Gbps บนสภาวะแวดล้อมและโทโปโลยีเดียวกัน เพื่อเปรียบเทียบวิธีการที่นำเสนอกับวิธีการแบบดั้งเดิม

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบดั้งเดิม

โครงข่าย IP แบบดั้งเดิม อุปกรณ์ Router จะใช้ Protocol ในการหาเส้นทางแบบ Conventional Routing เช่น OSPF (Open Shortest Path First), BGP (Border Gateway Protocol) เป็นโปรโตคอลที่ใช้ในการเชื่อมต่อและส่งข้อมูล โดยที่ Router แต่ละตัวจะสร้างตารางในการส่งเรียกว่า Forwarding Table เพื่อเป็นตัวบ่งชี้เส้นทางของ Hop ถัดไป เมื่อเส้นทางในการส่งข้อมูลเกิดความบกพร่อง ค่าใน Forwarding Table เกิดการเปลี่ยนแปลง Router จำเป็นต้องทำการคำนวนหาเส้นทางใหม่ที่เหมาะสมในการส่งข้อมูล แต่โครงข่าย MPLS/VPN ใช้วิธีการที่เรียกว่า MPLS Traffic Engineering (TE) Fast Reroute ในการเลือกเส้นทางในการส่งข้อมูลเมื่อเส้นทางเกิดการบกพร่องและการส่งข้อมูลได้อย่างต่อเนื่องโดยไม่หยุดชะงัก

ส่วนการจัดการคุณภาพการบริการ (Quality of Service) โครงข่าย IP แบบดั้งเดิม อุปกรณ์ Router จะใช้วิธีกำหนดคุณภาพการให้บริการแบบ Best Effort Protocol โดยข้อมูลทุกประเภทจะมีลำดับความสำคัญที่เท่าเทียมกันซึ่งโครงข่ายจะให้บริการข้อมูลที่มาถึงก่อนในลำดับแรก ในกรณีเช่นนี้ข้อมูลที่ต้องการให้บริการแบบ Real Time จะมีผลกระทบหากโครงข่ายที่ให้บริการเกิดความคับคั่งของข้อมูล แต่โครงข่าย MPLS ใช้วิธีการที่เรียกว่า MPLS Differentiated Service (MPLS DiffServ) ในการกำหนดคุณภาพการบริการโดยทำการตรวจแยก Packet ที่ต้องการให้บริการเป็นลำดับแรกแล้วทำการ Mark Packet ข้อมูลเพื่อให้โครงข่ายทราบว่าต้องให้บริการ Packet ข้อมูลนี้ในลำดับแรกก่อน Packet ข้อมูลอื่นๆ ถ้าโครงข่ายที่ให้บริการเกิดความคับคั่งของข้อมูล Packet ที่ถูกจัดลำดับความสำคัญสูงสุดจะถูกส่งออกไปยังปลายทางก่อน Packet ข้อมูลปกติทั่วไปดังนั้นวิธีการเช่นนี้สามารถช่วยให้บริการแบบ Real Time สามารถใช้งานได้โดยไม่เกิดผลกระทบ

ดังนั้นในการทดสอบความสามารถด้านความน่าเชื่อถือของระบบ (Reliability) และความสามารถในการจัดการคุณภาพการบริการ (Quality of Service) ของโครงข่าย MPLS/VPN โดยใช้วิธี MPLS Traffic Engineering (TE) Fast Reroute และ MPLS Differentiated Service (MPLS DiffServ) นั้นจะให้ค่า Recovery Time และการสูญเสียข้อมูลที่ต่ำกว่า สามารถควบคุมการให้บริการ Application ที่เป็น Real Time เช่น VoIP , IPTV , VDO Conference ได้ดีมากขึ้นซึ่ง

จะทำให้ระบบมีประสิทธิภาพที่ดีกว่าโครงข่าย IP แบบดั้งเดิมที่ใช้วิธี OSPF Routing protocol , BGP (Border Gateway Protocol) และ Best Effort Protocol

1.6 ขอบเขตการวิจัย

ในวิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการหาเส้นทางการส่งข้อมูลด้วยการทำ MPLS Traffic Engineering (TE) Fast Reroute และ MPLS DiffServ เปรียบเทียบกับวิธีการแบบดั้งเดิมที่ใช้หลักการของ OSPF Routing protocol และ Best Effort protocol โดยทำการทดสอบบนโครงข่ายจริงที่ระดับความเร็ว 10 Gbps ผลที่ได้แสดงประสิทธิภาพของโครงข่ายในด้านของความเชื่อถือและความสามารถในการให้บริการอย่างต่อเนื่องเปรียบเทียบกับวิธีการแบบดั้งเดิม

1.7 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานระบบ Traditional IP Network ที่ใช้ในการวิจัย ซึ่งประกอบด้วยหลักการทำงานของ โพรโตคอลทีซีพี/ไอพี (TCP/IP - Transmission Control Protocol/Internet Protocol) รูปแบบการรับส่งข้อมูลของระบบเครือข่ายอีเทอร์เน็ต (Ethernet) และหลักการทำงานของ OSPF Routing Protocol

บทที่ 3 กล่าวถึงพื้นฐานระบบ MPLS Network ที่ใช้ในการวิจัยซึ่งประกอบด้วยหลักการทำงานของ MPLS Network , Virtual Private Network (VPN) , MPLS Traffic Engineering และ Quality of Service (QoS)

บทที่ 4 กล่าวถึงพารามิเตอร์ที่ใช้ในการจำลองระบบและผลที่ได้จากการทดลอง โดยผลที่ได้จะแสดงให้เห็นว่าประสิทธิภาพของระบบ และวิธีการที่นำเสนอนั้นสามารถช่วยให้ระบบมีประสิทธิภาพและสมรรถนะที่ดีขึ้น

บทที่ 5 เป็นบทสรุปผลการวิจัยและข้อเสนอแนะ

บทที่ 2

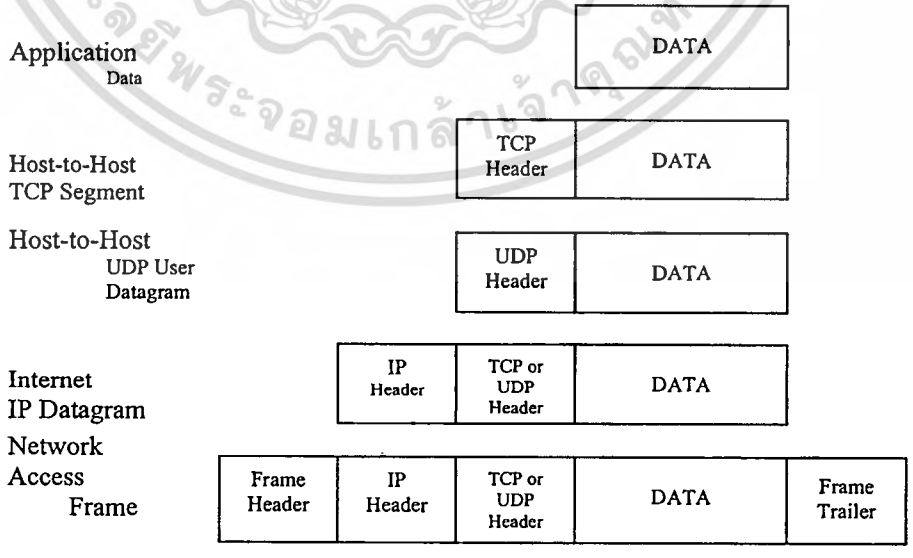
Traditional IP Network

2.1 โพรโตคอลทีซีพี/ไอพี

โพรโตคอล TCP/IP เป็นโพรโตคอลที่ทำงานอยู่ในชั้นโครงข่าย ซึ่งมีการเชื่อมต่อแบบ Connection Oriented โดยมีอุปกรณ์การสื่อสารซึ่งเรียกว่า เราเตอร์ (Router) ทำหน้าที่ในการส่งผ่านข้อมูลของผู้ใช้ในรูปของ IP ดาต้าแกรม กระบวนการในการตัดสินใจเลือกเส้นทางในการส่ง IP ดาต้าแกรมในแต่ละตัวจึงเป็นประเด็นหลักที่ต้องได้รับการพิจารณาและการออกแบบอย่างมีประสิทธิภาพ เพื่อให้การรับส่ง IP ดาต้าแกรมมีความรวดเร็วและมีความผิดพลาดน้อยที่สุด

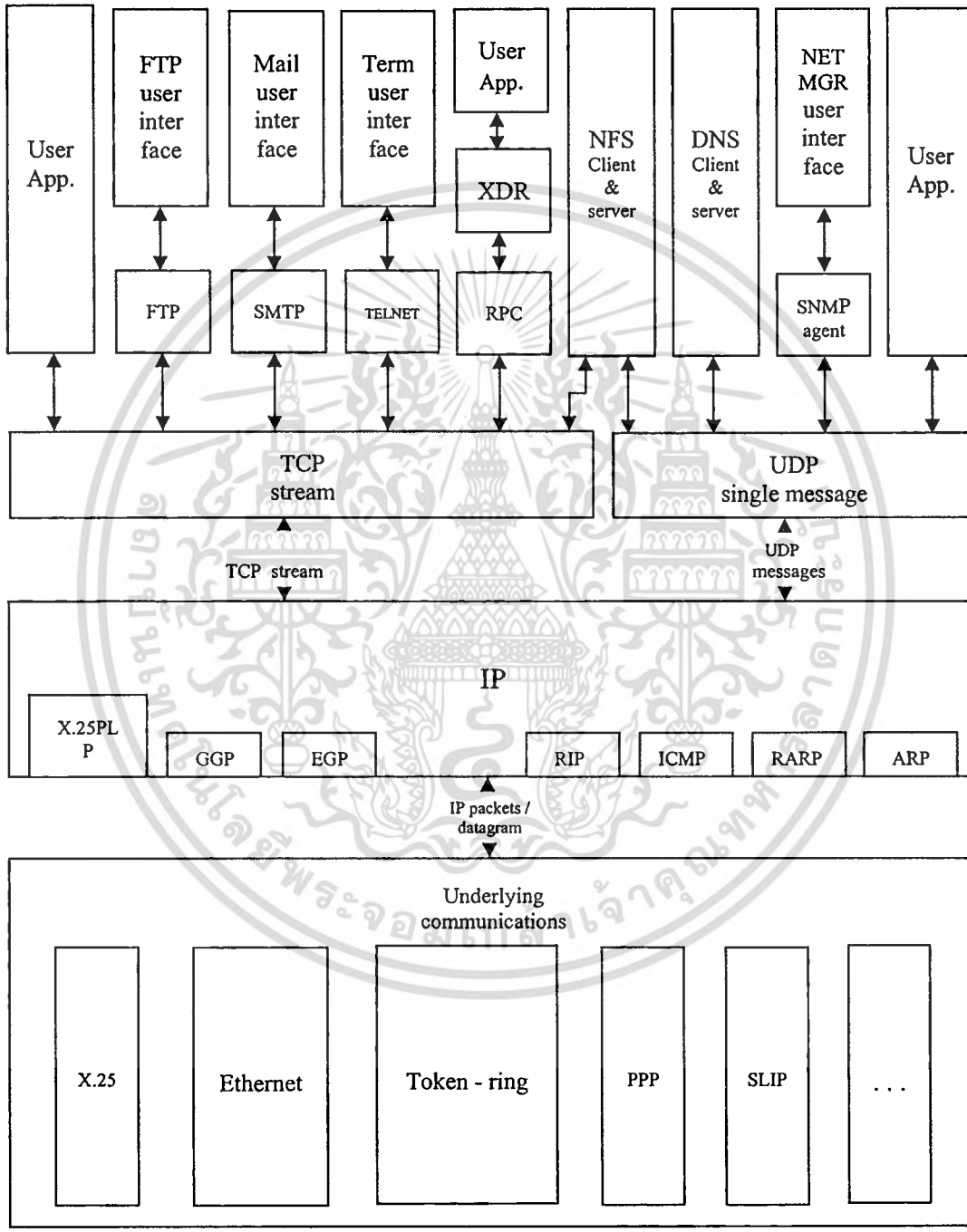
โพรโตคอลทีซีพี/ไอพี (TCP/IP - Transmission Control Protocol/Internet Protocol) เป็นกลุ่มโพรโตคอลที่พัฒนาขึ้นเพื่อให้คอมพิวเตอร์สามารถใช้ทรัพยากรและบริการฟังก์ชั่นพื้นฐานสำหรับการใช้งานบนระบบสื่อสารข้อมูลคอมพิวเตอร์ได้ ในสมัยก่อนนิยมใช้ทีซีพี/ไอพีในการสื่อสารที่ใช้ในเครื่องระดับมินิคอมพิวเตอร์หรือเมนเฟรม ซึ่งจะมีบริการอยู่หลายแบบ เช่น การล็อกอินจากที่อื่น การแลกเปลี่ยนไฟล์ข้อมูล จดหมายอิเลคทรอนิกส์ เป็นต้น ปัจจุบันชุดโพรโตคอลทีซีพี/ไอพีได้รับความนิยมในการใช้งานอย่างแพร่หลาย เนื่องจากใช้เป็นโพรโตคอลหลักในการติดต่อสื่อสารบนระบบเครือข่ายคอมพิวเตอร์ขนาดใหญ่ที่สุดในโลกที่เรียกว่าระบบอินเตอร์เน็ต (Internet)

ข้อมูลที่ใช้ทีซีพี/ไอพีนำส่งจะถูกแบ่งออกเป็นข้อมูลย่อยหลายส่วน ๆ เพื่อทยอยส่งไปตามลำดับเพื่อให้เหมาะสมกับระบบเครือข่ายในชั้นถัดไปที่อาจจะไม่สามารถส่งข้อมูลขนาดใหญ่ได้ทันที และเมื่อส่งไปถึงปลายทางก็จะรวบรวมข้อมูลนั้นกลับเป็นข้อมูลชุดเดิมอีกครั้งหนึ่ง ซึ่งจะมีการจัดรูปแบบแพ็กเก็ตในการสื่อสารดังรูป



รูปที่ 2.1 การจัดเตรียมข้อมูลเป็นแพ็กเก็ตเพื่อทำการส่ง

ชุดโพรโตคอลทีซีพี/ไอพีจะมีทีซีพี/ไอพีเป็นหลักและโพรโตคอลอื่น ๆ ที่ทำงานร่วมกับทีซีพี/ไอพีในชั้นอื่น ๆ ของทีซีพี/ไอพีโมเดล ซึ่งจะมีทั้งที่เป็นโพรโตคอลช่วยเหลือ เช่น ICMP, ARP, RIP และโพรโตคอลที่ใช้ทำงานหลัก เช่น TELNET, FTP, SMTP, HTTP, SNMP เป็นต้น



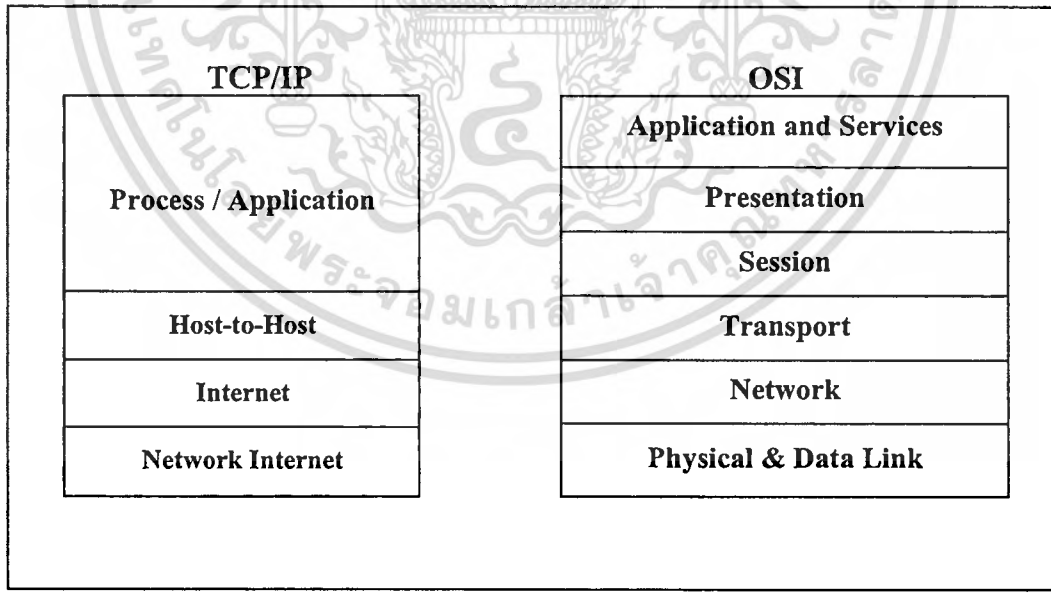
รูปที่ 2.2 การใช้งานชุดโพรโตคอลทีซีพี/ไอพี

2.1.1 ชั้นต่าง ๆ ของทีซีพี/ไอพี (TCP/IP Layer)

การติดต่อสื่อสารของทีซีพี/ไอพีถูกกำหนดให้มีการทำงานเป็นระดับชั้น (Layer) เพื่อให้มีการทำงานเป็นอิสระต่อกันในแต่ละระดับชั้น และเพื่อให้มีขั้นตอนการทำงานในการแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์เป็นไปอย่างถูกต้อง ดังนี้.-

- กำหนดรูปแบบข้อมูล
- จัดเตรียมชุดข้อมูล
- กำหนดเส้นทางการส่งข้อมูล
- กำหนดอัตราความเร็วในการส่งข้อมูล
- ทำการส่งข้อมูลผ่านตัวกลาง
- รวบรวมและจัดลำดับชุดข้อมูลที่ส่งมา
- ตรวจสอบว่ามีชุดข้อมูลซ้ำหรือไม่
- ตอบกลับไปให้ผู้ส่งรู้ว่าได้รับข้อมูลแล้ว
- ส่งผ่านข้อมูลไปให้ชั้นการทำงานถัดไป

เมื่อเปรียบเทียบกับโมเดลอ้างอิงการเชื่อมต่อระบบเปิด (Open System Interconnection Reference Model : OSI-RM) โดย ISO จะได้ดังนี้



รูปที่ 2.3 การแบ่งระดับการทำงานของทีซีพี/ไอพี และ OSI

2.1.2 ชั้นเชื่อมต่อระบบเครือข่าย (Network Interface Layer)

ทำงานในชั้นเดียวกับ OSI Physical Layer และ Data Link Layer ชั้นนี้จะทำหน้าที่ในการสื่อสารข้อมูลทางกายภาพ ในระดับสัญญาณนำส่ง ตัวนำที่ใช้ในการส่ง ระบบสื่อสัญญาณ และรูปแบบสัญญาณที่ใช้ในการแทนข้อมูล ว่าเป็นสัญญาณลอจิก "O" หรือ "I" ตัวอย่างของระบบที่ทำงานในชั้นนี้ เช่น ระบบเครือข่ายแบบอีเทอร์เน็ต หรือระบบเครือข่ายแบบโทเก้นริง และจัดข้อมูลเป็นกลุ่มที่เรียกว่าเฟรม (Frame) เฟรมจะมีส่วนหัวใช้แสดงตำแหน่งของต้นทางและปลายทาง ข่าวสารที่ใช้ในการควบคุม และส่วนท้ายที่ใช้ในการตรวจสอบข้อผิดพลาดการติดต่อสื่อสารระดับล่างสุดเฟรมจะถูกส่งจากอุปกรณ์เชื่อมต่อระบบเครือข่าย (Network Interface Device) ของเครื่องต้นทางผ่านระบบสื่อสัญญาณต่าง ๆ ไปถึงอุปกรณ์เชื่อมต่อระบบเครือข่ายของเครื่องปลายทาง

2.1.3 ชั้นอินเตอร์เนต (Internet Layer)

ชั้นนี้จะมีอินเตอร์เน็ตโพรโตคอล (Internet Protocol) หรือไอพี (IP) เป็นโพรโตคอลหลักคอยทำการหาเส้นทางที่เหมาะสมให้ในการสื่อสารข้อมูลระหว่างระบบข้อมูลที่จะส่งเรียกว่า เดต้าแกรม (Datagram) ซึ่งจะถูกส่งไปในระบบที่อาจจะเชื่อมต่อกันโดยตรง หรือเชื่อมต่อกันผ่านระบบสื่อสารอื่น ๆ อยู่ก็ได้

ไอพีจะทำงานแบบไม่มีการเชื่อมต่อก่อน (Connectionless) เดต้าแกรมแต่ละตัวจะถูกจัดเส้นทางในการส่งเป็นอิสระต่อกัน ไอพีไม่มีการรับประกันความถูกต้อง ความน่าเชื่อถือ หรือแม้แต่การจัดเรียงลำดับเดต้าแกรมให้อยู่ในลำดับที่ถูกต้อง

ชุดข้อมูลจะถูกส่งเข้าไปในระบบเครือข่าย โดยแต่ละเครือข่ายจะมีเครื่องที่ทำหน้าที่จัดเส้นทาง (Router) ซึ่งจะดูหมายเลขปลายทางแล้วตัดสินใจว่าจะส่งข้อมูลไปในเส้นทางไหน ตัวจัดเส้นทางนี้อาจจะเป็นเครื่องคอมพิวเตอร์ธรรมดาซึ่งเพิ่มหน้าที่การหาเส้นทางเข้าไป หรือใช้เครื่องที่ทำหน้าที่จัดเส้นทางโดยเฉพาะ กว่าที่ข้อมูลจะไปถึงปลายทาง อาจจะต้องผ่านตัวจัดเส้นทางของหลายเครือข่าย จึงต้องมีการผนวกหมายเลขของเครื่องต้นทางและเครื่องปลายทางเข้าไปในชุดข้อมูล เพื่อให้เราท์เตอร์รู้ว่าข้อมูลที่ผ่านเข้ามา ต้องการจะไปไหน ถ้าไม่ใช่หมายเลขของเครือข่ายตัวเอง ก็จะส่งต่อไปยังเครือข่ายที่อื่น แต่ถ้าใช่ก็จะส่งไปให้กับสมาชิกทั้งหมดของเครือข่าย เครื่องที่อยู่ในเครือข่ายจะตรวจสอบชุดข้อมูลที่ผ่านมาทั้งหมดว่าเป็นข้อมูลของตัวเองหรือไม่ ถ้าใช่ก็จะดับข้อมูลนั้นไว้ แล้วส่งให้กับส่วนการทำงานในชั้นโฮสต์ทูโฮสต์อีกทีหนึ่ง

2.1.4 ชั้นโฮสต์ทูโฮสต์ (Host – to – Host Layer) - TCP และ UDP

โพรโตคอลที่ทำงานในชั้นโฮสต์ทูโฮสต์นี้มีอยู่ 2 แบบ แบบที่เรียกว่า ทีซีพี (TCP – Transmission Control Protocol) จะทำงานแบบมีการเชื่อมต่อก่อน (Connection Orient) ซึ่งจะเป็นส่วนการทำงานภายในตัวคอมพิวเตอร์แต่ละเครื่อง มีหน้าที่นำส่งข้อมูลโดยรับประกันความน่าเชื่อถือให้

ด้วยว่าข้อมูลที่นำส่งจะไม่มีข้อผิดพลาดและเรียงอยู่ในลำดับที่ถูกต้อง โพรโตคอลทีซีพีจะทำการเพิ่มส่วนหัวของชั้นโพรโตคอลให้กับข้อมูลเพื่อสร้างเป็นเซ็กเมนต์ (Segment) โพรโตคอลอีกแบบหนึ่งได้แก่ ยูดีพี (UDP – User Datagram Protocol) ซึ่งจะทำงานแบบไม่มีการเชื่อมต่อก่อน (Connectionless) และไม่มีการรรับประกันความถูกต้องของข้อมูล เรียกข้อมูลที่ส่งโดยยูดีพีว่า User Datagram ตัวอย่างการทำงานโดยใช้ยูดีพี เช่น การสอบถามข้อมูลชื่อจากฐานข้อมูลในระบบ Domain Name System

2.1.5 ชั้นโปรแกรมประยุกต์ (Application Layer)

ชุดโพรโตคอลทีซีพี/ไอพีจะมีโพรโตคอลในชั้นโปรแกรมประยุกต์ให้ใช้งานอยู่มาก ที่นิยมใช้กันมากและจัดเป็นบริการพื้นฐานของทีซีพี/ไอพี เช่น

การล็อกอินระยะไกล (Remote Login) ทำให้ผู้ใช้เครื่องคอมพิวเตอร์ในระบบเครือข่ายสามารถทำการล็อกอินเข้าไปใช้ทรัพยากรในคอมพิวเตอร์อื่นที่ต่อเชื่อมกันอยู่ในระบบเครือข่ายได้จากเทอร์มินอล (Terminal) ของตัวเองซึ่งมีความแตกต่างกัน โดยใช้เทลเน็ตโพรโตคอล (TELNET – Telecommunication Network Protocol) ทำให้เกิดโปรแกรม Telnet ที่ใช้ระบบเทอร์มินอลจำลอง (NVT – Network Virutal Terminal) ซึ่งสามารถใช้งานกับระบบคอมพิวเตอร์ได้เกือบทุกระบบ

การส่งผ่านแฟ้มข้อมูล (Files Transfer) ทำให้ผู้ใช้เครื่องคอมพิวเตอร์เครื่องใดก็ตามสามารถรับส่งไฟล์จากเครื่องคอมพิวเตอร์อื่นได้ โดยใช้ไฟล์ทรานสเฟอร์โพรโตคอล (FTP – Files Transfer Protocol) ทำหน้าที่ในการคัดลอกแฟ้มข้อมูลระหว่างเครื่อง และทำงานทั่วไปเกี่ยวกับแฟ้มข้อมูล เช่น การเปลี่ยนชื่อแฟ้ม การลบแฟ้ม เป็นต้น

การส่งไปรษณีย์อิเล็กทรอนิกส์ (Electronic Mail) ทำให้ผู้ใช้ส่งข้อความไปหาผู้ใช้คนอื่นในระบบเครือข่ายได้ โดยการกำหนดรูปแบบข้อความที่จะส่งให้เป็นมาตรฐานเดียวกันในกระบวนการในการรับและการส่งระหว่างเครื่องต่าง ๆ

บริการเวิลด์ไวด์เว็บ (World Wide Web) จัดเป็นบริการที่มีความสามารถมากที่สุดในโปรแกรมประยุกต์ที่ทำงานแบบ Client/Server ของทีซีพี/ไอพี และได้รับความนิยมสูงมากในปัจจุบัน ทำให้ผู้ใช้สามารถสืบค้นข้อมูลในลักษณะของ Hypermedia ได้โดยการใช้เอชทีทีพี โพรโตคอล (HTTP – Hypertext Transfer Protocol)

เน็ตเวิร์คไฟล์ซิสเต็ม (NFS – Network File System) เป็นการอนุญาตให้ระบบเข้าถึงข้อมูลจากคอมพิวเตอร์เครื่องอื่นได้โดยผ่านระบบไฟล์ของระบบจัดการนั้น ๆ เอง

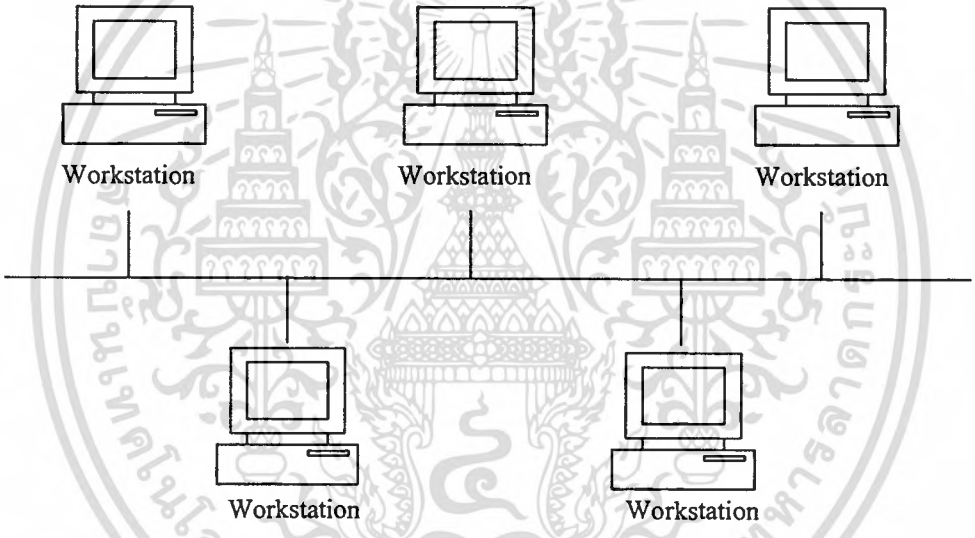
การพิมพ์ระยะไกล (Remote Printing) ทำให้สามารถใช้งานเครื่องพิมพ์ผ่านระบบเครือข่ายได้

นอกจากนี้ ก็มีระบบการให้บริการค้นหาชื่อสมาชิกเครือข่าย (Domain Name Server – DNS) การจัดการระบบเครือข่าย (Simple Network Management) โดยใช้ SNMP (Simple Network Management Protocol)

ในปัจจุบันได้มีการสร้างโปรแกรมที่ทำงานบนเครือข่ายโดยใช้ ทีซีพี/ไอพีจำนวนมาก เช่น ระบบฐานข้อมูลที่ให้บริการระหว่างเครื่องในเครือข่าย หรือการทำงานบนระบบประมวลผลแบบกระจาย (Client / Server Processing) ซึ่งแสดงให้เห็นว่าทีซีพี/ไอพี มีบทบาทเป็นโพรโตคอลพื้นฐานที่สำคัญของการใช้งานระบบเครือข่ายในปัจจุบัน

2.2 อีเทอร์เน็ต

ชั้นล่างสุดของชุดโพรโตคอลทีซีพีไอพีจะเป็นชั้นเชื่อมต่อกับระบบเครือข่าย ซึ่งจะทำงานร่วมกับระบบเครือข่ายได้หลายประเภท ที่นิยมมากที่สุดในระบบเครือข่ายคอมพิวเตอร์แบบท้องถิ่น จะเป็นระบบเครือข่ายที่เรียกกันว่า อีเทอร์เน็ต (Ethernet) ซึ่งเป็นระบบเครือข่ายที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบัน



รูปที่ 2.4 การเชื่อมต่อเครื่องคอมพิวเตอร์ในระบบเครือข่ายอีเทอร์เน็ต

การสื่อสารในระบบเครือข่ายอีเทอร์เน็ตจะต้องมีหมายเลขแอดเดรส (Address) เป็นตัวกำหนดการติดต่อสื่อสารระหว่างสถานีงานต้นทางและสถานีงานปลายทาง ค่าหมายเลขนี้เป็นเลขขนาด 48 บิตที่ถูกกำหนดมาจากโรงงานที่ทำการ์ดเชื่อมต่อระบบเครือข่ายและต้องไม่มีการซ้ำกัน ข้อมูลที่ถูกส่งออกไปให้กับสถานีงานทุก ๆ ตัว เมื่อสถานีงานได้รับแพ็กเก็ตก็จะทำการตรวจสอบว่าเป็นแพ็กเก็ตที่ส่งมาถึงตนเองหรือไม่ ถ้าไม่ก็จะนำข้อมูลในแพ็กเก็ตนั้นมาดำเนินการต่อ ซึ่งจะพิจารณาจากส่วนหัวอีเทอร์เน็ต โดยทุก ๆ อีเทอร์เน็ตแพ็กเก็ตจะมีส่วนหัวขนาด 14 อ็อกเต็ทที่ใช้บอกถึงแอดเดรสต้นทาง (Source Address) แอดเดรสปลายทาง (Destination Address) และชนิด (Type) เมื่อได้รับแพ็กเก็ตที่ถูกต้องแล้ว สถานีงานจะพิจารณาที่ไทพ์โคด (Type Code) เพื่อนำแพ็กเก็ตนั้นมาประมวลผลต่อไป

2.2.1 หลักการทำงานของอีเทอร์เน็ต

อีเทอร์เน็ตใช้หลักการของ CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ซึ่งทุกสถานีงานจะใช้สายสื่อระบบร่วมกัน หากมีการชนกันของข้อมูลเกิดขึ้นจะต้องส่งข้อมูลนั้นใหม่หมด อีเทอร์เน็ตมีข้อดีคือเป็นระบบที่มีการใช้งานกันมานานอย่างแพร่หลายและมีราคาถูก ง่ายในการติดตั้งและใช้งานทำให้มีการใช้งานกันอย่างกว้างขวางในปัจจุบันเมื่อเทียบกับระบบอื่น อย่างไรก็ตามข้อเสียของอีเทอร์เน็ตก็คือการใช้ CSMA/CD เป็นโพรโตคอลในการส่งข้อมูลจะมีประสิทธิภาพลดลงเมื่อมีการใช้งานระบบเครือข่ายเพิ่มมากขึ้นเพราะโอกาสที่จะส่งข้อมูลพร้อมกันในสายสื่อระบบและเกิดการชนกันจะมีมากขึ้น

เด็ค (DEC) อินเทลคอร์ปอร์เรชัน (Intel) และ ซีรอก (Xerox) ได้ร่วมกันกำหนดมาตรฐานอีเทอร์เน็ตเป็นรุ่นที่ 1 (DIX Ethernet) โดยมีลักษณะดังนี้.-

- รูปร่างระบบเครือข่าย ใช้โทโพโลยีแบบบัส (bus)
- ใช้วิธีการสื่อสารแบบ CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
- อัตราการรับส่งข้อมูลสูงสุด 10 เมกกะบิต ต่อวินาที
- ความยาวสูงสุด 2.5 กิโลเมตร
- จำนวนสถานีงานสูงสุดต่อเครือข่าย 1024 เครื่อง
- ใช้หลักการส่งแบบมีแถบกว้างความถี่ (Baseband)
- ขนาดเฟรมเปลี่ยนแปลงได้

64 Bist	48 Bits	48 Bits	16 Bits	368-12000 Bits	32 Bits
Preamble	**Destination Address**	**Source Address**	**Frame Type**	**Frame Data**	**CRC**

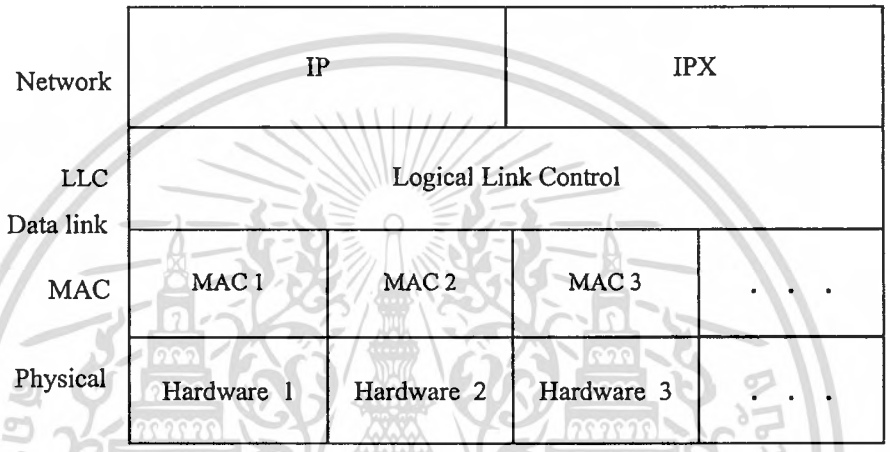
รูปที่ 2.5 โครงสร้างเฟรมของ DIX Ethernet

ซึ่งต่อมา IEEE ก็ได้ปรับปรุงโพรโตคอลนี้เพิ่มเติมและออกเป็นระบบมาตรฐานที่เรียกว่ามาตรฐาน 802.3 ซึ่งเป็นอีเทอร์เน็ตอีกแบบหนึ่งที่มีความเข้ากันได้กับ DIX Ethernet เดิมโดยเปลี่ยนฟิลด์ Frame Type ไปเป็น Data Length แทนค่าของ Frame Type จะเริ่มที่ 0800H ดังนั้น Data Length จึงมีค่าได้ไม่เกิน 0800H ถ้าข้อมูลในฟิลด์นี้มีค่าตั้งแต่ 0800H ขึ้นไปก็จะถือเป็น Frame Type ของ DIX Ethernet แต่ถ้าเป็นค่าที่น้อยกว่าก็จะเป็น Data Length ของ Ethernet 802.3 ซึ่งมีรายละเอียดของเฟรมดังต่อไปนี้

64 Bist	48 Bits	48 Bits	16 Bits	368-12000 Bits	32 Bits
Preamble	**Destination Address**	**Source Address**	**Data Length**	**Frame Data**	**CRC**

รูปที่ 2.6 โครงสร้างเฟรมของ IEEE 802.3 Ethernet

การที่อีเทอร์เน็ตมีเฟรมข้อมูลที่แตกต่างกันทำให้สามารถส่งข้อมูลหลายรูปแบบและหลายปลายทางผ่านไปยังฮาร์ดแวร์ที่ใช้การ์ดเชื่อมต่อระบบเครือข่ายแบบอีเทอร์เน็ตตัวเดียวกันได้ ทำให้มีลักษณะ Multiprotocol Stack เกิดขึ้น โดยจะใช้การบอกตัว Multiplexer ว่าเฟรมข้อมูลนี้จะต้องส่งไปที่โพรโตคอลในชั้นข้างบนอย่างไร ชนิดของเฟรมทั้งหมดได้แก่ ETHERNET_802.2, ETHERNET_802.3, ETHERNET_11 และ ETHERNET_SNAP ในการทำลักษณะนี้จะต้องแบ่งการทำงานในชั้นที่ 2 ออกเป็น 2 ชั้นย่อย คือ LLC – Logical Link Control และ MAC – Media Access Control ดังรูปที่ 2.7



รูปที่ 2.7 การทำมัลติโพรโตคอลแสต็กของอีเทอร์เน็ต

จะเห็นว่าสามารถมีฮาร์ดแวร์ได้หลายชนิดหรือเป็นชนิดเดียวกันแต่มีอุปกรณ์มากกว่า 1 ตัว เช่น มีการ์ดเชื่อมต่อระบบเครือข่ายแบบอีเทอร์เน็ต 2 การ์ด และการ์ดเชื่อมต่อระบบเครือข่ายแบบโทเก้นริง 1 การ์ด เป็นต้น และมีโพรโตคอลในชั้นเหนือขึ้นไปได้มากกว่าหนึ่งแบบเช่นกัน เช่น มี IP และ IPX เป็นต้น ลักษณะการทำงานแบบนี้ เรียกว่า ODI – Open Datalink Interface ซึ่งออกแบบโดยบริษัทโนเวล (Novell) เพื่อใช้กับระบบปฏิบัติการเครือข่ายที่เรียกว่า NetWare

ตัวอย่างเช่น ในสถานีงานมีการ์ดเชื่อมต่อแบบอีเทอร์เน็ตจำนวน 1 การ์ด และมีโพรโตคอลชั้นบน 2 แบบ คือ IP เพื่อติดต่อกับระบบจัดการแบบ UNIX และ IPX เพื่อติดต่อกับระบบจัดการแบบเน็ตแวร์ ส่วน LLC จะต้องมีการกำหนดว่าเฟรมข้อมูลแบบไหนเป็นของโพรโตคอลอะไร เช่นให้ IP เป็น ETHERNET _11 และ IPX ETHERNET_802.3 เป็นต้น เพื่อที่จะได้รับส่งข้อมูลได้อย่างถูกต้อง โดยโปรแกรมที่จะโหลดมีตามลำดับ ดังนี้

- LLC LSL.COM
- LAN Card Driver NE2000.COM

- IPX/SPX IPXODI.COM
- Session, Presentation, Application NETX.EXE
- TCP/IP TCPIP.EXE
- Session, Presentation, Application เช่น TNVT220 หรือ Netscape

2.2.2 ส่วนประกอบของอีเทอร์เน็ตเฟรม

อีเทอร์เน็ตเฟรมมีขนาดประมาณ 46 และ 1500 ไบต์ เฟรมน้อยกว่า 60 ไบต์ ในส่วนข้อมูลจะเรียกว่า Runt Frame

Preamble	Destination Address	Source Address	Type	Data	FCS

รูปที่ 2.8 โครงสร้างของอีเทอร์เน็ตเฟรม

เฟรมนี้ประกอบด้วย 6 ฟิลด์ด้วยกัน ซึ่งมีรายละเอียด ดังนี้

พรีเอมเบิล (Preamble)

เลขลำดับ 64 บิตที่ฟิสิคัลเลเยอร์ใช้เพื่อการสร้างสัญญาณพร้อม (Synchronization Signal) ระหว่างวงจรที่เชื่อมต่อกับสื่อที่ใช้ในการส่ง

ที่อยู่ปลายทาง (Destination Address)

เลขฮาร์ดแวร์แอดเดรส ขนาด 48 บิต ซึ่งเรียกว่า อีเทอร์เน็ตแอดเดรส หรือ แมคแอดเดรส การ์ดเชื่อมต่อระบบเครือข่ายทุกแผ่นจะมีเลข 48 บิตนี้ใช้บอกถึงเลขประจำตัวของการ์ดเชื่อมต่อนั้น ๆ ของสถานีงานที่ต้องการส่งข้อมูลไป

ที่อยู่ต้นทาง (Source Address)

เป็นแมคแอดเดรสของผู้ส่ง

ไทป์ (Type)

เลขขนาด 16 บิต เพื่อบอกชนิดโพรโตคอล ถ้ามีการใช้งานหลายโพรโตคอลในระดับเดียวกันบนสื่อเดียวกัน จะใช้เลขนี้บอกถึงโพรโตคอลในระดับชั้นบนขึ้นไป

ข้อมูล (Data)

เป็นข้อมูลขนาดระหว่าง 46 ถึง 1500 ไบต์

เอฟซีเอส (FCS – Frame Check Sequence)

เลขซีอาร์ซีขนาด 32 บิตที่คำนวณจากทุก ๆ ฟิลด์ยกเว้นฟิลด์ตัวเอง

ในระบบเครือข่ายอีเทอร์เน็ตทั่วไปจะมีโครงสร้างเฟรมของอีเทอร์เน็ตที่ใช้พื้นฐานโครงสร้างเฟรมอีเทอร์เน็ตคล้ายกัน เพียงแต่แตกต่างกันตรงส่วนการใช้งานของแต่ละแบบที่ไม่เหมือนกันซึ่งมีด้วยกันดังนี้

1. อีเทอร์เน็ต 802.3

อีเทอร์เน็ต 802.3 เฟรมนี้มีลักษณะคล้ายกับอีเทอร์เน็ตทู แต่จะมีบางฟิลด์ที่แตกต่างกัน ดังรูปที่ 3.6 ซึ่งมีหน้าที่และขนาดเฉพาะ ดังมีรายละเอียดดังนี้

Preamble	Destination Address	Source Address	Length	Data	FCS

รูปที่ 2.9 โครงสร้างของเฟรมอีเทอร์เน็ต 802.3

โครงสร้างเฟรม

พรีเอมเบิ้ล

เลขลำดับ 56 บิต ที่ฟิสิคัลเลเยอร์ใช้เพื่อการสร้างสัญญาณพร้อม (Synchronization Signal) ระหว่างวงจรที่เชื่อมต่อกับสื่อที่ใช้ในการสื่อสาร

เอสเอฟดี (SFD – Start Frame Delimiter)

เป็นเลขไบนารี่ (Binary) 10101011 ที่ชี้จุดเริ่มต้นของเฟรม

หมายเลขปลายทาง

เลขฮาร์ดแวร์แอดเดรสขนาด 48 บิต เรียกว่า อีเทอร์เน็ตแอดเดรส หรือ แมคแอดเดรส ของสถานีงานที่ต้องการส่งข้อมูลไปแอดเดรสที่มีค่าเป็น OFFFFFFFFFFFFH จะหมายถึงบรอดคาสแอดเดรส

หมายเลขต้นทาง

เลขฮาร์ดแวร์แอดเดรสขนาด 48 บิตของผู้ส่งซึ่งต้องไม่เป็นบรอดคาสท์แอดเดรส

ความยาว (Length)

เป็นเลขขนาด 16 บิต เพื่อบอกขนาดแพ็กเก็ต ค่านี้จะต้องน้อยกว่า 1500

ข้อมูล (Data)

เป็นข้อมูลขนาดระหว่าง 46 ถึง 1500 ไบต์

แพดดิ้ง (Padding)

ฟิลด์นี้มีขนาดเปลี่ยนแปลงได้ เป็นค่าที่เพิ่มเข้าไปเพื่อให้แพ็กเก็ตมีขนาดตรงตามข้อกำหนด เช่น เฟรมอีเทอร์เน็ตต้องมีขนาดอย่างน้อย 64 ไบต์ ซึ่งถ้ามีการส่งข้อมูลที่มีขนาดน้อยกว่านี้จะต้องเพิ่ม

แพดดิ้งเข้าไปเพื่อให้ครบ 64 ไบต์ ถ้าเฟรมถูกต้องและค่าความยาวมากกว่า 1500 ไบท์ จะหมายความว่าเป็นอีเทอร์เน็ตทูเฟรมและเป็นไทป์ฟิลด์

2. อีเทอร์เน็ต 802.2

เฟรมอีเทอร์เน็ต 802.2 มีข้อมูลทั้ง 802.3 ฟิลด์ และ 802.2 ฟิลด์ ซึ่ง 802.2 ฟิลด์จะแสดงถึงชั้น LLC (Logical Control) ภายในเฟรม

Preamble	SFD	DA	SA	Length	DSAP	SSAP	Control	Data	Pad	FCS

รูปที่ 2.10 โครงสร้างเฟรมอีเทอร์เน็ต 802.2

โครงสร้างเฟรม

พรีเอมเบิ้ล

เลขลำดับ 56 บิต ที่ฟิสิคัลเลเยอร์ใช้เพื่อการสร้างสัญญาณพร้อม (Synchronization Signal) ระหว่างวงจรที่เชื่อมต่อกับสื่อที่ใช้ในการสื่อสาร

หมายเลขปลายทาง

เลขฮาร์ดแวร์แอดเดรสขนาด 48 บิต เรียกว่า อีเทอร์เน็ตแอดเดรส หรือ แมคแอดเดรส ของสถานีงานที่ต้องการส่งข้อมูลไป แอดเดรสที่มีค่าเป็น OFFFFFFFFFFFH จะหมายถึงบรอดคาสแอดเดรส

หมายเลขต้นทาง

เลขฮาร์ดแวร์แอดเดรสขนาด 48 บิตของผู้ส่ง

ความยาว (Length)

เป็นเลขขนาด 16 บิต เพื่อบอกขนาดแพ็กเก็ต ค่านี้จะต้องน้อยกว่า 1500

ข้อมูล (Data)

เป็นข้อมูลขนาดระหว่าง 46 ถึง 1500 ไบต์

แพ็ดดิ้ง (Padding)

ขนาดเปลี่ยนแปลงได้

เอฟซีเอส (FCS - Frame Check Sequence)

เลขซีอาร์ซีขนาด 32 บิต ที่คำนวณจากทุก ๆ ฟิลด์ยกเว้นฟิลด์ตัวเอง

ดีเอสเอพี (DSAP - Destination Service Access Point)

เป็นเซอร์วิสแอคเซสพอยต์ปลายทางของสถานีงานปลายทางซึ่งใช้ในเลเยอร์บนหรือเน็ตเวิร์คเลเยอร์

เอสเอสเอพี (SSAP - Source Service Access Point)

เป็นเซอร์วิสแอคเซสพอยต์ต้นทางของสถานีงานต้นทางซึ่งใช้ในเลเยอร์บนหรือเน็ตเวิร์คเลเยอร์

ฟิลด์ควบคุม (Control)

กำหนดการส่งแบบคอนเน็คชันเลสเซอร์วิส

3. อีเทอร์เน็ตทู (Ethernet II)

อีเทอร์เน็ตทูเฟรมแตกต่างจากอีเทอร์เน็ตเฟรม 2 แบบที่กล่าวมาเนื่องจากไทป์ฟิลด์ซึ่งตามหลังที่อยู่ปลายทาง แต่อีเทอร์เน็ต 802.3 อีเทอร์เน็ต 802.2 จะเป็นฟิลด์ความยาวแทน

Preamble	Destination Address	Source Address	Type	Data	FCS

รูปที่ 2.11 โครงสร้างเฟรมอีเทอร์เน็ตทู

โครงสร้างเฟรม

พรีเอมเบิ้ล

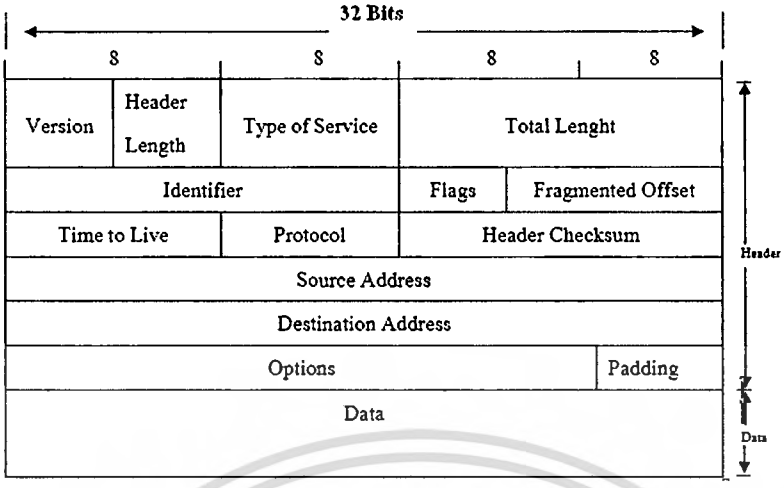
เป็นเลขลำดับ 64 บิตไบต์ที่ฟิสิคัลเลเยอร์ใช้เพื่อการสร้างสัญญาณพร้อม ระหว่างวงจรที่เชื่อมต่อกับสื่อ กำหนดด้วยค่าสลับกันระหว่าง "1" และ "0" ซึ่งจะมีด้วยกันทั้งหมด 7 ไบต์ ส่วนไบต์สุดท้ายเป็นเอสเอฟดี

ไทป์

เป็นส่วนที่บอกถึงชนิดโพรโตคอลที่ใช้ในระดับชั้นที่สูงกว่า ค่าโพรโตคอลที่ใช้มีดังนี้

ไอพี	0800H
เออาร์พี	0806H
อาร์เออาร์พี	8035H
แอบเปิ้ลทอล์ค	809BH
แอบเปิ้ลทอล์ค เออาร์พี	80F3H
เน็ตแวร์ ไอพีเอ็กซ์/เอสพีเอ็กซ์	8137H

2.3 โครงสร้างของ IP Header



รูปที่ 2.12 โครงสร้าง IP ดาต้าแกรม

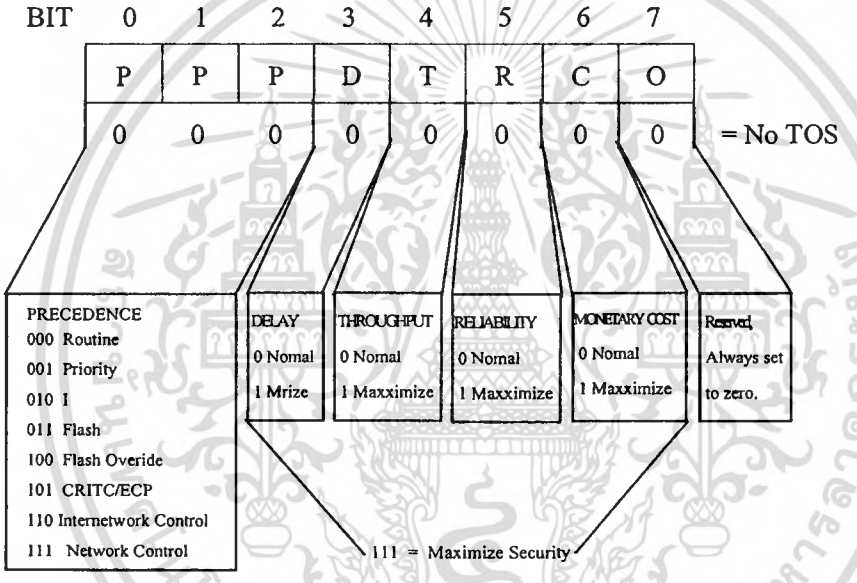
ในรูปที่ 2.12 แสดงถึงรูปแบบโครงสร้างของ IP Header โดยขนาดของ IP Header ปกติจะมีขนาด 20 ไบต์ จากภาพจะแสดงให้เห็นถึงส่วนประกอบของ IP ดาต้าแกรม ซึ่งประกอบด้วยองค์ประกอบหลัก 2 ส่วน คือ ส่วนของ Header และส่วนของข้อมูล (Data)

- ฟิลด์ Version : ระบุเวอร์ชันของ IP ที่ใช้ในการสร้าง IP ดาต้าแกรมในฟิลด์นี้จะมี ขนาด 4 บิตซึ่งปกติจะเซ็ทเป็น 0100 ในระบบของเลขฐานสอง ซึ่งแสดงถึงเวอร์ชัน 4 (IPv4) ซึ่งเป็นเวอร์ชันที่ใช้อยู่ในปัจจุบัน ในตารางที่ 2.1 แสดงให้เห็นถึงเวอร์ชันในแบบต่าง ๆ ที่สัมพันธ์กันกับ RFC

ตารางที่ 2.1 แสดงหมายเลขเวอร์ชันของ IP

Number	Version	RFC
0	Reserved	
1 – 3	Unassigned	
4	Internet Protocol (IP)	791
5	ST Datagram Mode	1190
6	Simple Internet Protocol (SIP)	
6	IPng	1883
7	TP/IX	1475
8	P Internet Protocol (PIP)	1621
9	TCP and UDP over Bigger Address (TUBA)	1347
10 – 14	Unassigned	
15	Reserved	

- ฟิลด์ Header Length : มีขนาด 4 บิตใช้บอกขนาด Header ของ IP โดยจะบอกในรูปของ word ของ 32 บิต ซึ่งปกติจะมีขนาดเท่ากับ 5 หรือเทียบเท่ากับ 20 ไบต์ และถ้าหากมีฟิลด์ Option เพิ่มเข้ามาจะทำให้ขนาดของ Header เท่ากับ 24 ไบต์
- ฟิลด์ Type of Service (TOS) : มีขนาด 8 บิต ใช้สำหรับบ่งบอกถึงคุณลักษณะหรือรูปแบบการให้การบริการที่แพ็กเก็ต IP ต้องการ ในฟิลด์นี้สามารถแบ่งออกได้เป็น 2 ส่วน คือส่วนของ Precedence และส่วนของ TOS ดังในรูปที่ 2.13 ส่วนของ Precedence นั้นมีจำนวน 3 บิตใช้สำหรับจัดลำดับความสำคัญของแพ็กเก็ตซึ่งมีได้ 8 ระดับ ในส่วนของ TOS มีไว้เพื่อใช้ในการเลือกการบริการในการส่งมอบแพ็กเก็ตในรูปของ Delay Throughput Reliability และ Monetary Cost



รูปที่ 2.13 ฟิลด์ Type of Service

- ฟิลด์ Total Length : มีขนาด 16 บิต ใช้สำหรับระบุขนาดของ IP ดาต้าแกรมทั้งหมดซึ่งรวม Header ด้วย ขนาดของ IP ดาต้าแกรมที่ใหญ่ที่สุดมีค่าเท่ากับ 65,535 ไบต์
- ฟิลด์ Identifier : มีขนาด 16 บิต ใช้ในการเชื่อมต่อฟิลด์ Flags และฟิลด์ Fragment Offset สำหรับการแบ่งแพ็กเก็ตออกเป็นแพ็กเก็ตย่อย ๆ ซึ่งแพ็กเก็ตจะถูกทำ Fragment เพื่อทำให้เป็นแพ็กเก็ตย่อย ๆ ก็ต่อเมื่อความยาวของแพ็กเก็ตที่มาจากต้นทางมีความยากมากเกินกว่าค่า Maximun Transmission Unit (MTU) ของดาต้าลิ้งค์ ในแต่ละเส้นทางที่แพ็กเก็ตนี้เดินทางผ่าน เช่น มีแพ็กเก็ตขนาด 5,000 ไบต์ ที่จะต้องเดินทางผ่านเครือข่ายร่วมซึ่งมีค่า MTU เป็น 1,500 ไบต์ เพราะฉะนั้นภายในเฟรมหนึ่ง ๆ จะสามารถบรรจุขนาดของแพ็กเก็ตได้สูงสุด 1,500 ไบต์ ทำให้เราเตอร์ซึ่งบรรจุแพ็กเก็ตไปบนดาต้าแกรม ทำการ Fragment แพ็ก

เก็ตแต่ละแพ็กเก็ตให้มีขนาดไม่เกิน 1,500 ไบต์ จากนั้นเราเตอร์จะทำการ Mark แพ็กเก็ตซึ่งถูก Fragment แล้วด้วยหมายเลขเดียวกันในฟิลด์ Identifier ซึ่งจะทำให้อุปกรณ์ทางด้านรับสามารถระบุได้ว่าแพ็กเก็ตที่ถูก Fragment นั้นเป็นแพ็กเก็ตเดียวกัน

- ฟิลด์ Flags : เป็นฟิลด์ที่มีขนาด 3 บิต โดยที่บิตแรกไม่มีการใช้งานและกำหนดให้เป็น 0 เสมอ บิตที่สองเรียกว่าบิต Don't Fragment (DF) มีไว้เพื่อกำหนดว่า IP ดาต้าแกรมนี้อนุญาตให้ทำ Fragment ได้หรือไม่ ถ้า Host ต้นทางกำหนดให้ DF = O ก็หมายถึงอนุญาตให้เราเตอร์ระหว่างทางทำการ Fragment ได้ถ้ามีความจำเป็น แต่ถ้าหากเซ็ท DF = 1 หมายความว่าห้ามทำการ Fragment ในกรณีนี้ถ้าหากเราท์เตอร์ไม่สามารถส่งดาต้าแกรมต่อไปได้หากไม่มีการทำ Fragment เราเตอร์ก็จะทิ้งดาต้าแกรมนั้นไป และส่งความผิดพลาดที่เกิดขึ้นกลับไปยังโฮสต์ต้นทาง บิตที่สามเรียกว่าบิต More Fragments (MF) เป็นบิตที่ถูกเซ็ทโดยเราเตอร์เมื่อมีการทำ Fragment กับ IP ดาต้าแกรมนั้น โดยจะมีค่า MF = 1 เพื่อแสดงว่ายังมี Fragment อื่นตามมาอีก เพราะฉะนั้นฟิลด์ Flag จึงเป็นตัวระบุให้โฮสต์ปลายทางทราบจุดสิ้นสุดของ IP ดาต้าแกรมมีข้อสังเกตว่า เมื่อ Fragment ในแต่ละส่วนอาจจะถูกส่งผ่านเครือข่ายด้วยเส้นทางที่แตกต่างกัน และ Fragment เหล่านี้อาจเดินทางมาถึงจุดหมายในลำดับที่ผิดไปจากเดิมได้ ดังนั้นหากโฮสต์ปลายทางได้รับ Fragment ของ IP ดาต้าแกรมยังไม่ครบถ้วนสมบูรณ์ ซึ่งจะเห็นได้ว่า การใช้เพียงฟิลด์ Identifier และ Flag จะไม่เพียงพอสำหรับโฮสต์ ปลายทางที่จะนำ Fragment มาประกอบกันได้อย่างถูกต้อง เพราะขาดข้อมูลที่บอกถึงลำดับการเรียงต่อของ Fragment ปัญหานี้สามารถแก้ไขได้โดยอาศัยฟิลด์ Fragment Offset ที่จะได้กล่าวต่อไป
- ฟิลด์ Fragment Offset : ทำหน้าที่ชี้หรือระบุตำแหน่งเริ่มต้นของส่วนย่อยแต่ละส่วนภายใน IP ดาต้าแกรมฟิลด์นี้มีขนาด 13 บิต โดยค่าที่ใช้มีหน่วยเป็นจำนวนเท่าของ 8 ไบท์ เมื่อโฮสต์ปลายทางอ่านค่าฟิลด์นี้ประกอบกับฟิลด์ Total Length ของ Fragment ที่ได้รับแต่ละตัว ก็จะทำให้สามารถตรวจสอบว่าได้รับ Fragment ของ IP ดาต้าแกรมครบถ้วนหรือไม่
- ฟิลด์ Time To Live (TTL) : มีขนาด 8 บิต มีหน้าที่กำหนดจำนวนเราเตอร์สูงสุดที่ IP ดาต้าแกรมสามารถเดินทางผ่านได้ หรือกล่าวในอีกนัยหนึ่งได้ว่าเป็นการกำหนดอายุของ ดาต้าแกรมที่อนุญาตให้อยู่ในเครือข่ายได้ ขั้นตอนในการทำงาน คือ เมื่อโฮสต์ต้นทางทำการส่งดาต้าแกรมออกไปจะตั้งค่าเริ่มต้นให้กับฟิลด์ TTL ค่าหนึ่ง (โดยทั่วไปใช้ 32 หรือ64) ทุกครั้งที่ดาต้าแกรม เดินทางผ่านเราท์เตอร์ตัวหนึ่งค่าของ TTL จะถูกปรับลดลงหนึ่งหน่วย หากเมื่อใดเราเตอร์พบดาต้าแกรมที่ค่า TTL ลดลงจนเป็น 0 เราเตอร์จะตัดดาต้าแกรมนั้นทิ้งไปพร้อมกับแจ้งให้โฮสต์ต้นทางทราบ การทำเช่นนี้จะทำให้สามารถป้องกัน IP ดาต้าแกรมที่รับส่งผิดพลาดได้

- ฟิลด์ Header Checksum : มีขนาด 16 บิตเป็นฟิลด์ที่ทำหน้าที่ตรวจสอบความถูกต้องของ IP Header โดยมีลักษณะการทำงานดังนี้ เมื่อโฮสท์ต้นทางทำการสร้างดาต้าแกรมขึ้นจะคำนวณค่า Header Checksum โดยนำ Header ทีละ 16 บิตมาบวกกันแบบหนึ่งต่อหนึ่งคอมโพแนนซ์ จากนั้นนำผลที่ได้มาทำหนึ่งต่อหนึ่งคอมโพแนนซ์อีกครั้ง จึงจะได้เป็นค่าที่บรรจุลงใน Header Checksum โดยที่ด้านรับจะตรวจสอบความผิดพลาดของ Header โดยนำ Header ทีละ 16 บิตมาบวกกับค่าในฟิลด์ Heaader Checksum แบบหนึ่งต่อหนึ่งคอมโพแนนซ์ หากผลลัพธ์ที่ได้มีค่าเป็นหนึ่งทั้งหมด แสดงว่าไม่มีความผิดพลาดเกิดขึ้นหากไม่ใช่ก็แสดงว่ามีความผิดพลาดเกิดขึ้นกับ Header ในกรณีนี้ IP ดาต้าแกรมจะถูกตัดทิ้งโดยไม่มีการแจ้งความผิดพลาดที่เกิดขึ้น ซึ่งโพรโตคอลในชั้นที่สูงกว่าต้องตรวจสอบปัญหานี้ด้วยตัวเอง
- ฟิลด์ที่อยู่ต้นทางและที่อยู่ปลายทาง : คือที่อยู่ IP ของต้นทางและของปลายทางมีขนาด 32 บิต

ตารางที่ 2.2 ตัวอย่างค่าภายในฟิลด์โพรโตคอล

Protocol Number	Host-to-Host Layer Protocol
1	Internet Control Message Protocol (ICMP)
2	Internet Group Management Protocol (IGMP)
3	Gateway to Gateway Protocol (GGP)
4	IP in IP
6	Transmission Control Protocol (TCP)
8	Exterior Gateway Protocol (EGP)
17	User Datagram Protocol (UDP)
35	Inter-Domain Policy Routing Protocol (IDPR)
45	Inter-Domain Routing Protocol (IDRP)
46	Resource Reservation Protocol (RSVP)
47	Generic Routing Encapsulation (GRE)
54	NBMA Next Hop Resolution Protocol (NHRP)
88	Cisco Internet Gateway Routing Protocol (IGRP)
89	Open Sho test Path First (OSPF)

- ฟิลด์ Option : เป็นส่วนที่เพิ่มเติมเมื่อมีการใช้งานบางอย่าง เช่น การทดสอบเครือข่ายและตรวจหาจุดผิดพลาดของระบบ ฟิลด์นี้จะมีขนาดไม่ตายตัวขึ้นอยู่กับชนิดของ Option ที่

เลือกใช้ เช่น Loose Source Routing Strict Sourec Routing Record Route และ Time Stamp เป็นต้น

2.4 IP Routing

IP Routing เป็นกระบวนการค้นหาเส้นทางในการส่งผ่านข้อมูลจากต้นทางไปยังที่หมายปลายทางโดยผ่านการส่งต่อของอุปกรณ์ IP ที่อยู่ในเน็ตเวิร์กซึ่งจะช่วยกันทำหน้าที่ส่งต่อข้อมูลไปจนกว่าข้อมูลจะถึงปลายทาง กลไกสำคัญที่ทำให้ IP เป็นโปรโทคอลสำหรับขนส่งข้อมูลไปยังทุก ๆ ที่ในโลกบนอินเตอร์เน็ตที่ดีที่สุดในขณะนี้คือการที่ IP มีกระบวนการ IP Routing นี้เอง สิ่งที่น่าสนใจที่สุดของ IP Routing คือการที่ต้นทางและปลายทางของการสื่อสารนั้นในบางโอกาสต่างก็อยู่กันแสนไกล การสื่อสารข้อมูลแต่ละครั้ง ข้อมูลจะต้องเดินทางผ่านโครงข่ายอันสลับซับซ้อนมากมาย แต่ในที่สุดข้อมูลก็สามารถส่งถึงกันได้ในเวลาอันรวดเร็วเป็นที่น่าอัศจรรย์โครงข่ายอินเตอร์เน็ตคงไม่อาจเกิดขึ้นได้หากไม่มีโพรโตคอล IP ที่ช่วยขนส่งข้อมูลไปบนเครือข่ายอย่างมีประสิทธิภาพ การเข้าใจถึงกระบวนการ IP Routing จะช่วยให้เราเข้าใจคุณสมบัติของอินเตอร์เน็ตได้เป็นอย่างดี กระบวนการ IP Routing นี้ได้ถูกออกแบบมาอย่างชาญฉลาดและรัดกุมพอสมควรในอันที่จะให้บรรลุภารกิจในการส่งข้อมูล หลักการพื้นฐานของ IP Routing เริ่มต้นข้อกำหนดที่เรียบง่ายดังนี้.

อุปกรณ์ที่ใช้ในเน็ตเวิร์ค จำแนกได้เป็น 2 ประเภท คือ

- Host โฮสต์เป็นอุปกรณ์ที่ทำหน้าที่ให้กำเนิดข้อมูลในกรณีเป็นผู้ส่ง หรือทำหน้าที่รับข้อมูลไปใช้งานในกรณีเป็นผู้รับ การสื่อสารข้อมูลใด ๆ จะต้องเป็นการสื่อสารจากโฮสต์ไปยังโอสต์เสมอ สำหรับ IP Packet แล้วข้อมูลในเฮดเดอร์ที่ปรากฏในฟิลด์ที่อยู่ต้นทางและที่อยู่ปลายทาง ซึ่งเรียกว่า IP Address จะเป็นหมายเลขระบุตำแหน่งของโฮสต์ต้นทางและโฮสต์ปลายทางเท่านั้น
- Router เราเตอร์เป็นอุปกรณ์สำคัญอย่างยิ่งสำหรับ IP ที่จะทำให้การขนส่งข้อมูลเป็นไปอย่างสมบูรณ์ เราเตอร์ทำการส่งผ่านข้อมูลจากเน็ตเวิร์กหนึ่งไปยังอีกเน็ตเวิร์กหนึ่งตำแหน่งของเราเตอร์จะอยู่ในจุดที่เชื่อมต่อระหว่างสองเน็ตเวิร์กเข้าด้วยกัน ด้วยข้อกำหนดของ IP ข้อมูลจะส่งไปถึงกันโดยตรงข้ามเน็ตเวิร์คไม่ได้ จะต้องอาศัยเราเตอร์เป็นผู้ทำหน้าที่ส่งผ่านข้อมูลไปให้ ดังนั้นเน็ตเวิร์คของ IP ถึงแม้มิได้ต่อถึงกันในทางกายภาพแต่ก็สามารถสื่อสารกันได้โดยอาศัยเราเตอร์เป็นตัวเชื่อมประสานเข้าด้วยกัน

ก่อนที่จะอธิบายกลไกเบื้องต้นของกระบวนการ IP Routing ขออธิบายถึงส่วนสำคัญอีกส่วนหนึ่งที่เกี่ยวข้องกับกระบวนการนี้พอเป็นสังเขปเพื่อทำความเข้าใจในเรื่องของ IP Routing เนื่องจากจะมีการกล่าวถึงในส่วนนี้พอสมควรนั่นคือเน็ตเวิร์ค ในที่นี้จะกล่าวถึงเฉพาะเน็ตเวิร์คใน

ความหมายของ IP (IP Network) เท่านั้น ไม่รวมถึงเน็ตเวิร์คประเภทอื่น ใน IP นั้นจะมีการระบุหมายเลขประจำโฮสต์โดยใช้ที่อยู่ IP เพื่อระบุตำแหน่งของต้นทางและปลายทาง โดยในที่อยู่ IP นั้นนอกจากจะระบุตำแหน่งของโฮสต์แล้ว ยังใช้ระบุตำแหน่งของเน็ตเวิร์คที่โฮสต์นั้นเชื่อมต่ออยู่ด้วย ทั้งนี้โพรโตคอล IP มีกระบวนการที่จะแยกหมายเลขประจำตัวของโฮสต์และของเน็ตเวิร์คออกจากกัน เพื่อทำให้อุปกรณ์ทั้งหลายสามารถพิจารณาในทันทีว่าจะส่งข้อมูลที่ได้รับมานั้นไปในทิศทางใด

เนื่องจาก IP เป็นโพรโตคอลที่อยู่ในระดับชั้นที่สูง จะต้องอาศัยการทำงานที่สอดคล้องกันของโพรโตคอลที่อยู่ในระดับขั้นที่ต่ำกว่าด้วยหมายเลขเน็ตเวิร์คของ IP เป็นค่าที่เป็นลอจิคัล คือ กำหนดขึ้นเองหรืออาจเปลี่ยนแปลงได้โดยมิได้ผูกติดกับอุปกรณ์ทางกายภาพ แต่อย่างไรก็ตามการกำหนดหมายเลขเน็ตเวิร์คของ IP ก็จำเป็นต้องสอดคล้องกับหมายเลขเน็ตเวิร์คของระดับชั้นล่างด้วยเช่นกัน นั่นหมายความว่าถึงแม้ว่าในระดับชั้นที่ต่ำกว่าเช่นระดับชั้นลิ้งค์จะเชื่อมต่อถึงกันอย่างสมบูรณ์ แต่ถ้ามีการกำหนดค่าของ IP เน็ตเวิร์คไม่ถูกต้องก็จะไม่สามารถสื่อสารได้ ในทางกลับกันถึงแม้มีการกำหนดค่า IP เน็ตเวิร์คที่ถูกต้องแต่เน็ตเวิร์คไม่สามารถสื่อถึงกันได้ในระดับชั้นลิ้งค์ ก็ไม่สามารถสื่อสารได้เช่นกัน

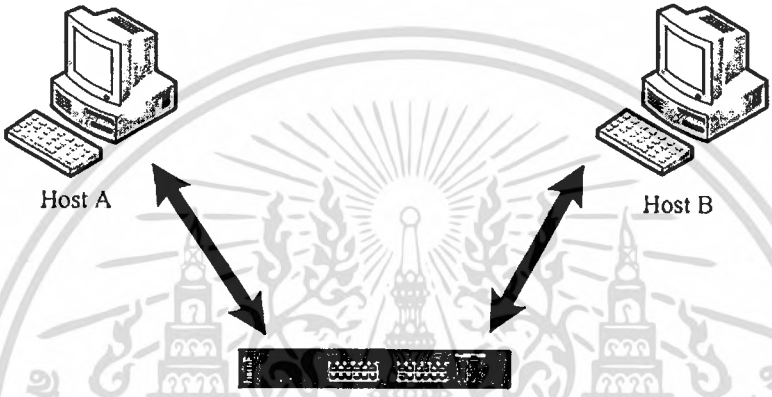
2.4.1 หลักการพื้นฐานของ IP Routing

IP Routing โดยใช้ Default Router กระบวนการ IP Routing เริ่มต้นด้วยหลักพื้นฐานที่ไม่สลับซับซ้อนและเข้าใจได้ไม่ยาก คือ

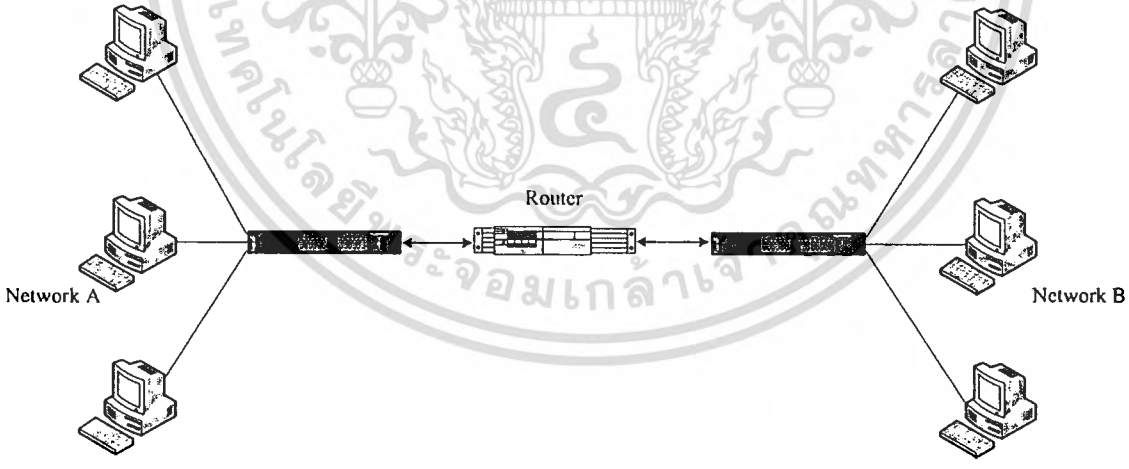
รูปที่ 2.14 การสื่อสารในการเชื่อมต่อแบบจุดต่อจุด

1. ถ้าโฮสต์ต้นทางและปลายทางต่อถึงกันโดยตรงเช่นการเชื่อมต่อแบบจุดต่อจุด ตามปรากฏในรูปที่ 2.14 IP ข้อมูลหรือดาต้าแกรมนั้นจะถูกส่งไปยังโฮสต์ปลายทางโดยตรง
2. ถ้าโฮสต์ต้นทางและปลายทางต่อเชื่อมร่วมอยู่ในเน็ตเวิร์คเดียวกัน เช่น อีเธอร์เน็ตหรือโทเค็นริงดังแสดงในรูปที่ 2.14 IP ดาต้าแกรมก็จะส่งไปยังโฮสต์ปลายทางโดยตรง

3. ถ้าไม่เป็นไปตามข้อที่ 1 และ 2 IP ดาต้าแกรมจะถูกส่งไปยังดีฟอลต์เราเตอร์ เพื่อทำการส่งต่อข้อมูลไปยังปลายทางต่อไป
4. เมื่อเราเตอร์ได้รับ IP ดาต้าแกรมจากข้อ 3 แล้วตรวจสอบดู หากพบว่าโฮสต์ปลายทางต่อร่วมอยู่บนเน็ตเวิร์คเดียวกันกับเราเตอร์ให้ทำการส่งดาต้าแกรมไปที่โฮสต์นั้น หากไม่ได้ต่อร่วมกันก็ส่งดาต้าแกรมไปที่เราเตอร์ตัวต่อไป และกลับไปที่ขั้นตอนในข้อ 2 ใหม่ จนกว่า IP ดาต้าแกรม่จะเดินทางถึงปลายทางหรือหมดเวลาในการส่ง



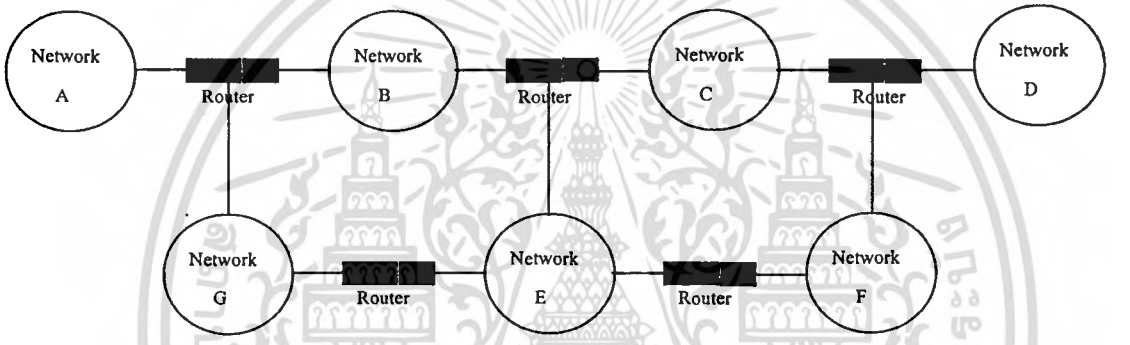
รูปที่ 2.15 การสื่อสารในเน็ตเวิร์คที่ต่อร่วมกัน (Shared Network)



รูปที่ 2.16 การสื่อสารระหว่าง 2 เน็ตเวิร์ค

หากเน็ตเวิร์คมีเพียง 2 เน็ตเวิร์คเหมือนในรูปที่ 2.16 มีเพียงแค่ดีฟอลต์เราเตอร์ก็คงจะเพียงพอและการทำงานในการส่ง IP ดาต้าแกรมข้ามระหว่างเน็ตเวิร์คก็คงจะไม่ยุ่งยากมากนักและคงเป็นไปตามขั้นตอนข้างต้น หากสังเกตจะเห็นว่าตัวเราเตอร์เองนั้น จะมีเน็ตเวิร์คที่ต้องติดต่อด้วย

2 ฝั่ง คือ เน็ตเวิร์ค A และเน็ตเวิร์ค B ซึ่งมีทิศทางการเคลื่อนที่ของข้อมูลเพียงเส้นทางเดียวมีเราเตอร์เพียงตัวเดียว ไม่ว่าปลายทางของข้อมูลจะไปที่ใดหากมิใช่อยู่ในเน็ตเวิร์คเดียวกันแล้ว ดาต้าแกรมทั้งหมดก็ต้องส่งผ่านเราเตอร์อยู่ดีโดยไม่ต้องทำการวิเคราะห์ใด ๆ การที่ดาต้าแกรมถูกส่งข้ามเน็ตเวิร์ค 1 ครั้งที่เรียกว่า 1 ฮอบ (Hop) เปรียบเสมือนระยะในการเดินทางของข้อมูล จากภาพตัวอย่างดาต้าแกรมเดินทางจากโฮสต์ต้นทางเพียง 1 ฮอบก็ถึงโฮสต์ปลายทาง การส่งต่อข้อมูลโดยเราเตอร์ก็มีเพียงส่งไปและส่งกลับระหว่างเน็ตเวิร์ค A และเน็ตเวิร์ค B เท่านั้น แต่หากระยะทางถึงโฮสต์ปลายทางจะต้องเดินทางมากกว่า 1 ฮอบแล้ว เราเตอร์ก็จะทำงานยุ่งยากซับซ้อนเพราะจะมีเน็ตเวิร์คอื่น ๆ ที่ไม่ได้เชื่อมต่อโดยตรงและต้องส่งข้อมูลผ่านเราเตอร์หลายตัวและมีหลายเส้นทางที่ดาต้าแกรมสามารถเดินทางไปได้ ดังนั้นการส่งต่อดาต้าแกรมของเราเตอร์จึงเป็นปัจจัยสำคัญในการกำหนดประสิทธิภาพของ IP Routing



รูปที่ 2.17 การเชื่อมต่อกันของหลายเน็ตเวิร์ค

การเดินทางของดาต้าแกรมโดยกระบวนการ IP Routing นั้น ทำงานอยู่บนพื้นฐานของการส่งข้อมูลทีละฮอบ (Hop-By-Hop) คือเราเตอร์เองจะทำงานโดยรู้จักเฉพาะเน็ตเวิร์คที่ต่ออยู่กับตัวเองเท่านั้น หากโฮสต์ปลายทางมิได้อยู่ในเน็ตเวิร์คที่ต่อเชื่อมอยู่ก็จะทำการส่งข้อมูลต่อไปอีกฮอบให้แก่เราเตอร์ตัวต่อไปส่งต่อและถือว่าหมดหน้าที่ต่อดาต้าแกรมนั้นแล้ว เพราะส่งข้อมูลต่อไปเรียบร้อยแล้ว ส่วนจะถึงปลายทางหรือไม่นั้นเป็นอีกเรื่องหนึ่ง และเราเตอร์ตัวอื่น ๆ ที่อยู่ระหว่างทางก็เช่นกันก็จะส่งต่อดาต้าแกรมไปเรื่อย ๆ เช่นนั้นที่ละฮอบจนกว่าจะถึงปลายทางหรือหมดเวลา

เพื่อให้กระบวนการ IP Routing ดำเนินไปอย่างมีประสิทธิภาพจึงมีการเพิ่มความสามารถของเราเตอร์ให้มากขึ้นกล่าวคือ ในกรณีที่โฮสต์ปลายทางมิได้อยู่ในเน็ตเวิร์คที่ต่ออยู่กับตัวเองนั้นแทนที่จะทำการส่งต่อข้อมูลไปยังดีฟอลต์เราเตอร์ทั้งหมด ก็ให้เราเตอร์ทำการพิจารณาเน็ตเวิร์คปลายทางว่าอยู่ที่ใดแล้วจึงทำการส่งต่อด้าต้าแกรมนั้นไปยังเราเตอร์ที่อยู่ใกล้กับเน็ตเวิร์คนั้นที่สุด (ใช้จำนวนฮอบในการส่งข้อมูลน้อยที่สุด) เพื่อการนี้จึงจำเป็นต้องมีข้อมูลให้แก่ตัวเราเตอร์ว่า

เน็ตเวิร์คใดควรจะส่งข้อมูลไปยังเราเตอร์ใด ข้อมูลเหล่านี้จะเก็บอยู่ในตารางเส้นทาง (Routing Table) ซึ่งจะประกอบด้วยข้อมูลดังนี้.-

- Destintion IP Address : หมายถึง แอดเดรสของโฮสต์หรือเน็ตเวิร์คปลายทาง
- IP Address of a Next-Hop Router : หมายถึง IP Address ของเราเตอร์ตัวอื่นที่ต่อโดยตรงอยู่บนเน็ตเวิร์คเดียวกัน
- Flags : จะเป็นข้อมูลส่วนที่ขยายความเพิ่มเติมของ Destination IP Address และ Next-hop-Router
- Interface : หมายถึง อินเตอร์เฟซของเราเตอร์ที่จะต้องใช้เพื่อการส่งดาต้าแกรมออกไป

เมื่อเราเตอร์มีตารางเส้นทางแล้ว กระบวนการในการที่จะส่งต่อดาต้าแกรมจากเราเตอร์ตัวหนึ่งไปยังตัวต่อไปจะต้องนำข้อมูลในเราต์ติ้งเทเบิลไปร่วมพิจารณาด้วย ดังนี้.-

1. ค้นหาข้อมูลในตารางเส้นทางเพื่อหา IP Address ที่ตรงกันพอดีกับ IP Address ของโฮสต์ปลายทางของดาต้าแกรม หากพบข้อมูลดังกล่าวให้ส่งไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next-Hop Router ทันที หากไม่พบข้อมูลให้ทำต่อในข้อที่ 2
2. ค้นหาในเราต์ติ้งเทเบิ้ลเพื่อหาเน็ตเวิร์คแอดเดรสที่ตรงกับเน็ตเวิร์คแอดเดรสของโฮสต์ปลายทาง หากพบข้อมูลดังกล่าวให้ส่งไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next Hop Router ทันที หากไม่พบข้อมูลให้ทำต่อในข้อที่ 3
3. ค้นหาในตารางเส้นทางเพื่อหาข้อมูลรายการที่ระบุว่า " Default " และให้ส่งต่อไปยังแอดเดรสที่ระบุอยู่ในฟิลด์ของ Next-Hop Router

กระบวนการที่กล่าวมานี้จะทำให้การส่งต่อข้อมูลเป็นไปในทิศทางที่เหมาะสมและมีประสิทธิภาพที่สุด โดยอาศัยข้อมูลที่กำหนดไว้ก่อนแล้วของเราเตอร์ ซึ่งหากไม่พบข้อมูลที่ตรงกับโฮสต์หรือเน็ตเวิร์คเลยในเราต์ติ้งเทเบิ้ลแล้วดาต้าแกรมก็จะถูกส่งไปยังดีฟอลต์เราเตอร์เสมอ ดังนั้นหากเราเตอร์แต่ละตัวต่างก็มีตารางเส้นทางที่ถูกต้องแล้ว ดาต้าแกรมก็จะถูกส่งต่อไปเรื่อย ๆ จนถึงปลายทางในที่สุด อาจจะล่าช้าหรือไม่มีประสิทธิภาพบ้างแต่ก็พอใช้งานได้

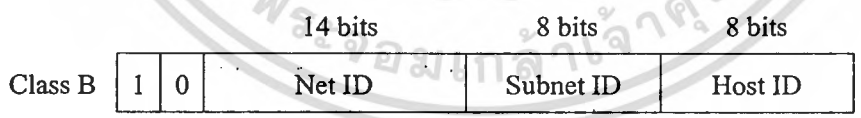
จะเห็นได้ว่าเราเตอร์จำเป็นต้องเก็บตารางเส้นทางของทุก ๆ เน็ตเวิร์คหรือทุก ๆ โฮสต์ไว้ทั้งหมด แต่จะอาศัยการกระจายกันของข้อมูลในเราเตอร์ทุก ๆ ตัวบนเน็ตเวิร์คและแต่ละตัวก็ส่งต่อข้อมูลให้ถูกต้อง ข้อมูลก็จะสามารถเดินทางถึงปลายทางได้ ด้วยเหตุนี้เองทำให้เน็ตเวิร์คสามารถขยายเพิ่มเติมออกไปได้เรื่อย ๆ โดยไม่จำกัดและไม่ต้องทำการแก้ไขโครงสร้างของเน็ตเวิร์คเดิม อินเตอร์เน็ต จึงแผ่ขยายครอบคลุมโลกได้อย่างรวดเร็ว

อย่างไรก็ตาม IP Routing โดยการใช้ตารางเส้นทางเป็นกระบวนการพื้นฐานเท่านั้น เมื่อนำมาซ้อนจนเกินกว่าจะใช้การเราติ้งแบบธรรมดาได้ จึงจำเป็นต้องมีกระบวนการที่มีประสิทธิภาพกว่านี้และมีการพัฒนากระบวนการ IP Routing ที่สลับซับซ้อนออกมาหลายรูปแบบ เช่น RIP, OSPF, BGP เป็นต้น

2.4.2 Subnet Addressing

ในตอนเริ่มต้นใช้โพรโตคอล TCP/IP นั้นการแบ่ง IP Address ออกเป็นแอดเดรสของเน็ตเวิร์ค (Net ID) และแอดเดรสของโฮสต์ (Host ID) เป็นไปตามกติกาที่ระบุของแต่ละคลาด ต่อมามีผู้เสนอให้มีการแบ่งเน็ตเวิร์คย่อยภายในแต่ละ Net ID เพิ่มขึ้นอีกเพื่อจะได้ใช้งาน IP Address ได้อย่างมีประสิทธิภาพที่สุด เนื่องจากในคลาส A และคลาส B นั้นมีการจัดสรรส่วนที่เป็น Host ID ในแต่ละเน็ตเวิร์คเป็นจำนวนมากคือในเน็ตเวิร์คคลาด A แต่ละเน็ตเวิร์คนั้นสามารถมีจำนวนโฮสต์ได้มากถึง = 16,777,214 โฮสต์ และสำหรับในแต่ละเน็ตเวิร์คคลาส C นั้นสามารถมีจำนวนโฮสต์ได้สูงที่สุดถึง = 65,534 โฮสต์ ซึ่งการที่จะนำ IP Address มาใช้อย่างทั่วถึงนั้นมีโอกาสเป็นไปได้ยากมาก ทั้งคลาส A และคลาส B เพราะมีโอกาสน้อยมากที่จะมีเน็ตเวิร์คใดในโลกมีจำนวนโฮสต์มากมายขนาดนั้นอยู่ภายในเน็ตเวิร์คเดียว ดังนั้น IP Address ที่จัดสรรไปให้ในแต่ละเน็ตเวิร์คของคลาสเหล่านี้จึงถูกใช้ไม่หมดและไม่สามารถนำไปใช้ประโยชน์ที่อื่นได้

การทำ Subnet คือ การแบ่งเน็ตเวิร์คย่อยภายในเน็ตเวิร์คหลักเพื่อให้แต่ละเน็ตเวิร์คมีขนาดที่เหมาะสมกับปริมาณโฮสต์ที่มีอยู่โดยใช้หลักการเดียวกับการนำ IP Address มาแยกออกเป็น Host ID และเป็น Network ID คือแทนที่จะให้ค่า Host ID เป็นค่าอิสระตั้งแต่ 1 จนถึงค่าสูงสุด ก็ทำการจัดกลุ่มของ Host ID เหล่านั้นออกเป็นกลุ่มของเน็ตเวิร์คย่อย คือนำค่าในส่วนที่เป็น Host ID เดิมมาแยกออกเป็นสองส่วนคือ Subnet และเป็น Host ID ใหม่ ซึ่งจะทำให้สามารถจัดสรรการใช้งาน IP Address ได้อย่างเหมาะสมกับการมีอยู่จริงของโฮสต์ในแต่ละเน็ตเวิร์ค



รูปที่ 2.18 IP Address ในคลาส B เมื่อทำการ Subnet

รูปที่ 2.18 แสดงการจัดแบ่ง IP Address ของคลาส B ออกเป็นเน็ตเวิร์คย่อยด้วยวิธี Subnetting โดยแบ่งพื้นที่ส่วนที่เป็นของ Host ID เดิมออกเป็น 2 ส่วน โดยเป็นของ Subnet ID ขนาด 8 บิท และ Host ID ใหม่ที่มีขนาดเล็กลงเหลือเพียง 8 บิท

ตารางที่ 2.3 ผลกระทบต่อจำนวนของขนาดของเน็ตเวิร์คเมื่อถูก Subnet

Class B	จำนวนเน็ตเวิร์คที่มีได้	จำนวนโฮสต์สูงสุดในแต่ละเน็ตเวิร์ค
เดิม	16,382	65,532
หลังการ Subnet	4,161,028	254

ผลที่ได้คือขนาดของเน็ตเวิร์คจะเล็กลงและมีจำนวนมากขึ้น จากเดิม 16,382 เน็ตเวิร์คก็เพิ่มเป็น 4,161,028 (16,382 * 254) และในขณะเดียวกันจำนวนโฮสต์ในแต่ละเน็ตเวิร์คจะลดลงจาก 65,532 เหลือเพียง 254 โฮสต์ อย่างไรก็ตามการ Subnet นั้นไม่จำเป็นต้องมีขนาดของ Subnet ID คงที่ตายตัวเสมอไป ผู้บริหารระบบสามารถปรับเปลี่ยนได้ตามความเหมาะสมของการใช้งาน เช่น อาจจะมีจำนวนเน็ตเวิร์คน้อยลง และจำนวนโฮสต์มากขึ้นก็สามารถกระทำได้โดยการแบ่งขนาดของ Subnet ID และ Host ID ใหม่ตามที่ต้องการ

นอกจากการที่สามารถใช้ IP Address ได้อย่างมีประสิทธิภาพแล้ว ข้อดีอีกอย่างหนึ่งของการ Subnet คือช่วยให้ประสิทธิภาพการสื่อสารดีขึ้นด้วย กล่าวคือ กระบวนการของ TCP/IP บางประการมีการใช้การสื่อสารแบบบรอดคาส (Broadcast) เพื่อทำการสื่อสารกระจายไปทุก ๆ โฮสต์ที่อยู่ในเน็ตเวิร์คเดียวกัน ดังนั้นหากเป็นเน็ตเวิร์คคลาส A ซึ่งมีโฮสต์ได้ถึง 16 ล้านโฮสต์แล้วการสื่อสารด้วยวิธีบรอดคาสท์แต่ละครั้งจะเป็นการกระจายข้อมูลไปยังเครื่องอื่น ๆ จำนวนมาก และใช้แบนวิดธ์มากมายมหาศาลทั่วไปทั้งเน็ตเวิร์ค ส่งผลกระทบต่อการสื่อสารตามปกติอย่างยิ่ง และหากเน็ตเวิร์คประเภทนี้ถูกโจมตีโดยเทคนิคที่อาศัยการขยายสัญญาณเนื่องจากการบรอดคาสท์ เช่น Smurf แล้วก็มีโอกาสมากที่การสื่อสารข้อมูลภายในเน็ตเวิร์คส่วนใหญ่จึงมักจะหลีกเลี่ยงการออกแบบให้เน็ตเวิร์คมีขนาดใหญ่เกินไป เนื่องจากควบคุมได้ยุ่งยากและมีประสิทธิภาพต่ำ วิธีการ Subnet จึงเป็นส่วนที่ถูกนำมาใช้ในการออกแบบเสมอ

โดยทั่วไปแล้วเรามักจะพบเห็นการ Subnet สำหรับ IP Address ในคลาส B เสียเป็นส่วนใหญ่ เนื่องจากคลาส B มีผู้ใช้งานกันแพร่หลาย ส่วนคลาส A จะพบได้ไม่บ่อยนักเพราะมีผู้ได้รับจัดสรรไม่มาก ส่วนคลาส C ก็อาจจะพอมีผู้ทำ Subnet อยู่บ้าง แต่เนื่องจากคลาส C มีขนาดของเน็ตเวิร์คไม่ใหญ่อยู่แล้ว จึงสามารถแบ่งย่อยออกไปได้อีก ๆ ไม่มากนัก อย่างไรก็ตาม IP Address ทุกคลาสล้วนแต่สามารถถูกนำมา Subnet ได้ทั้งสิ้น

2.4.3 Subnet Mask

หากกล่าวถึงการ Subnet Mask ด้วย การที่มีเฉพาะ IP Address เพียงอย่างเดียวนั้น กรณีที่เป็นการกำหนด Net ID และ Host ID ตามที่ระบุในคลาสต่าง ๆ นั้น เราก็สามารถทราบว่าค่าทั้งสองได้ไม่ยากนัก โดยพิจารณาว่า IP Address อยู่ในช่วงใด อยู่ในคลาดใด หลังจากนั้นก็สามารถแยก Net ID และ Host ID ได้จากการเปรียบเทียบกับมาตรฐานของคลาสเหล่านั้น

แต่เมื่อมีการแบ่งเน็ตเวิร์คย่อยโดยการ Subnet แล้ว ย่อมไม่สามารถใช้วิธีการข้างต้นเพื่อหา Net ID และ Host ID ได้อีกต่อไป เนื่องจาก Subnet นั้น สามารถกำหนดได้โดยผู้ออกแบบเน็ตเวิร์คเองและมิได้มีข้อบังคับแต่อย่างใด ดังนั้น จึงจำเป็นต้องมีการระบุค่าใดค่าหนึ่งไว้เพื่อให้สามารถนำมาใช้ในการหาค่าจาก IP Address ได้ว่าเป็น Host ID, Net ID และ Subnet ID และค่านั้นก็คือ Subnet Mask นั่นเอง

Subnet Mask เป็นตัวเลขขนาด 32 บิต เท่ากับ IP Address ทำหน้าที่ระบุหมายเลขของ Host ID และ Net ID + Subnet ID ของโฮสต์นั้น การกำหนดค่าของ Subnet Mask จะอยู่ในรูปแบบเดียวกับ IP Address คือทำการแบ่ง Subnet Mask ออกเป็นเลข 16 บิต จำนวน 4 ชุด และแยกแต่ละชุดออกจากกันด้วยจุด (.)

ตัวอย่าง Subnet Mask เช่น FF.FF.FF.OO (Hex) 255.255.255.0 (Dec)

ค่า Subnet Mask นี้จำเป็นต้องกำหนดไว้บนทุกโฮสต์คู่กับค่า IP Address เสมอ เนื่องจากโพรโตคอล IP จำเป็นต้องใช้ค่านี้ไปคำนวณค่า Net id ซึ่งจำเป็นอย่างยิ่งในกระบวนการ IP Routing อย่างไรก็ตามค่า Subnet Mask นี้จะไม่ถูกส่งไปกับ IP ดาต้าแกรมด้วย หมายเลขโฮสต์สามารถนำค่า Subnet Mask มาทำการทางคณิตศาสตร์กับ IP Address ก็จะสามารถหาค่า Host ID, Subnet ID, Net ID ออกมาโดยวิธีดังนี้.-

Net ID + Subnet ID = (IP Address) AND (Subnet Mask)
Host ID = (IP Address) AND (NOT ((Subnet Mask)))

ตัวอย่าง

IP Address	=	192.168.15.20	Subnet Mask		=	255.255.255.0
Net ID	=	192.168.15.20	AND	255.255.255.0	=	192.168.15.0
Host ID	=	192.168.15.20	AND	(0.0.0.255)	=	0.0.0.20

หรือพูดง่าย ๆ ได้ว่า "Net ID" ก็คือ IP Address ส่วนที่ตรงกับบิตของ Subnet Mask ที่มีค่าเป็น 1 ส่วน Host ID คือ IP Address ส่วนที่ตรงกับ Subnet Mask ที่มีค่าเป็น 0 นั่นเอง

ดังนั้นพึงระลึกเสมอว่านอกจากการกำหนด IP Address ที่ถูกต้องแล้ว การกำหนดค่า Subnet Mask ก็มีผลต่อ IP Routing เช่นเดียวกัน การกำหนดค่า Subnet Mask ผิดพลาดย่อมจะส่งผลให้การสื่อสารข้อมูลของ IP ไม่สามารถจะกระทำได้เช่นกัน

2.4.4 IP Address ในกรณีพิเศษ

ถึงแม้ค่าของ IP Address มีขนาด 32 บิต ที่นำมาใช้งานได้จริง แต่มีกรณีพิเศษเป็นข้อยกเว้นที่ไม่อาจนำค่าเหล่านี้มากำหนดเป็นค่า IP Address ได้ ซึ่งค่าส่วนใหญ่จะเป็นค่าที่ IP เอง

นำไปใช้งานเพื่อวัตถุประสงค์อื่นแล้ว ผู้ใช้จึงมิอาจนำค่าเหล่านั้นมาใช้งานอีก ดังรายละเอียดต่อไปนี้

ตารางที่ 2.4 IP Address สำหรับกรณีพิเศษ

IP Address			แอดเดรสของ		รายละเอียด
Net ID	Subnet ID	Host ID	ต้นทาง	ปลายทาง	
0 ทุกบิต	ไม่มี	0 ทุกบิต	ได้	ไม่ได้	เป็นการระบุโฮสต์นี้ภายในเน็ตเวิร์คนี้
0 ทุกบิต	ไม่มี	Host ID	ได้	ไม่ได้	เป็นการระบุโฮสต์หมายเลขตาม Host ID พายในเน็ตเวิร์คนี้
127 บิต	ไม่มี	อะไรก็ได้	ได้	ได้	แอดเดรสที่ภายในโฮสต์ตนเอง (Loopback Address)
ทุกบิต	ไม่มี	1 ทุกบิต	ไม่ได้	ได้	บรอดคาสต์เฉพาะภายในเท่านั้น แต่จะไม่ส่งต่อไปยังเน็ตเวิร์คอื่น
Net ID	ไม่มี	1 ทุกบิต	ไม่ได้	ได้	บรอดคาสต์โดยตรงไปยังเน็ตเวิร์คที่ระบุใน Net ID
Net ID	Subnet ID	1 ทุกบิต	ไม่ได้	ได้	บรอดคาสต์โดยตรงไปยังเน็ตเวิร์คย่อยที่ระบุใน Subnet ID
Net ID	1 ทุกบิต	1 ทุกบิต	ไม่ได้	ได้	บรอดคาสต์โดยตรงไปยังเน็ตเวิร์คย่อยทุกเน็ตเวิร์คภายในเน็ตเวิร์คที่ระบุใน Net ID

ตารางที่ 2.4 แสดง IP Address บางค่าที่ถูกโพรโตคอลนำไปใช้เพื่อวัตถุประสงค์อื่น และผู้ใช้ไม่สามารถนำมากำหนดเป็นแอดเดรสของโฮสต์ได้ โดยส่วนใหญ่ค่าที่มีปัญหาจะมีเพียง 3 ตัว คือ

127 หมายถึง Loopback Address คือส่งกลับเข้าหาตัวเอง
1 ทุกบิต หมายถึง โฮสต์ทุกตัวในเน็ตเวิร์ค (คือการบรอดคาสนั่นเอง)
0 ทุกบิต หมายถึง ตัวเน็ตเวิร์คเอง

ดังนั้น เพื่อป้องกันปัญหาของการกำหนดค่า IP Address จึงควรจะหลีกเลี่ยงการกำหนดค่า Net ID และ Host ID ด้วยเลขดังกล่าวเสีย

2.5 OSPF (Open Shortest Path First)

OSPF เป็นโปรโตคอลเลือกเส้นทางชนิดเปิด คือไม่ขึ้นอยู่กับค่ายผู้ผลิตใด ๆ และเป็นมาตรฐาน RFC 1131 (สำหรับ OSPF Version 1) โดย OSPF ได้รับการ Update มามากมายหลายครั้ง จนกลายมาเป็นมาตรฐาน RFC 2328 ซึ่งเป็น Version 2 ในปัจจุบัน OSPF เป็นโปรโตคอลเลือกเส้นทางที่มีประสิทธิภาพสูง สามารถขยายเครือข่ายได้ดี จัดเป็นโปรโตคอลเลือกเส้นทางแบบ Link State นั่นคือ การเลือกเส้นทางของ OSPF จะอาศัยสถานะการเชื่อมต่อของเครือข่ายเท่านั้น ไม่ต้องการค่า Metric ใด ๆ เพื่อการเลือกเส้นทาง OSPF จะใช้วิธีการตรวจสอบสถานะการเชื่อมต่อทั้งหมดบนเครือข่าย จากนั้นจัดสร้างฐานข้อมูลใน Router ที่ติดตั้ง OSPF ภายในฐานข้อมูลนี้ประกอบด้วย สถานะการเชื่อมต่อเครือข่ายทั้งหมด จากนั้นจะใช้วิธีการที่เรียกว่า Short Path First เพื่อพิสูจน์แสดงเส้นทางที่สั้นที่สุด ในการเดินทางสู่ Router หรือเครือข่ายปลายทาง

OSPF ไม่ต้องการส่งข้อมูลอันประกอบด้วยข่าวสารเกี่ยวกับการ Update เส้นทางในทุกห้วงของเวลา แต่จะมีการ Update ข่าวสารเกี่ยวกับเส้นทางให้แก่กันก็ต่อเมื่อมีการเปลี่ยนแปลงเกิดขึ้นเท่านั้น และข่าวสารที่ Update ให้แก่กันนั้น ก็เป็นเพียงข่าวสารเฉพาะส่วนที่มีการเปลี่ยนแปลงเท่านั้น ไม่ได้ยกให้ทั้งหมด

ข้อดีของ OSPF มีดังนี้

- **มีระบบ Metric ที่ยืดหยุ่นกว่า** แทนที่ OSPF จะใช้ค่า Metric ที่ได้มาจากคำนวณ หรือจากการกำหนดขึ้นโดยโปรโตคอลเลือกเส้นทาง อย่างเช่น IGRP ที่ถูกกำหนดให้ใช้ Bandwidth หรือค่า Delay อย่างใดอย่างหนึ่งเป็น Metric ที่นำมาใช้เพื่อการคำนวณเส้นทาง สำหรับ OSPF จะใช้ค่า Metric ที่ถูกกำหนดโดยชนิดของสื่อที่เชื่อมต่อ เช่น หาก Router OSPF เชื่อมต่อกันทาง Ethernet 10 Mbps ค่า Metric จะเท่ากับ 10 หรือหากเป็น Fast Ethernet ค่า Metric จะเท่ากับ 1 เป็นต้น โดย OSPF Router จะใช้ค่า Metric เหล่านี้ เพื่อเลือกเส้นทางโดยไม่ต้องมีการคำนวณเส้นทางเหมือนกับ EIGRP หรือ IGRP
- **สามารถรับรู้การเปลี่ยนแปลงของเครือข่ายได้เร็วกว่า** OSPF Router สามารถรับรู้การเปลี่ยนแปลงใด ๆ ที่เกิดขึ้นบนเครือข่ายได้เร็วมาก เนื่องจาก OSPF Router หากตรวจพบการเปลี่ยนแปลงใดที่เกี่ยวกับเครือข่ายหรือเส้นทาง มันจะ Update ตารางเส้นทางของมัน จากนั้นจะ Update ให้กับ Router เพื่อบ้านในทันที และทุกทิศทาง นอกจากนี้การ Update จะเกิดขึ้นแบบ เล่าต่อ ๆ กันไป
- **สามารถเชื่อมต่อเครือข่ายในรูปแบบขั้นบันได (Hierarchical Routing)** ความสามารถของ OSPF ประการหนึ่งได้แก่ความสามารถในการเชื่อมต่อแบบขั้นบันได ทำได้โดยการแบ่งกลุ่มของ Router ออกเป็นส่วน ๆ เรียกว่าพื้นที่ (Area) ซึ่งทำให้การบริการ

จัดการเครือข่ายทำได้ง่ายรวมทั้งความสามารถในการขยายเครือข่ายสามารถทำได้โดยง่าย เช่นกัน

- **สามารถลดปัญหา Overhead** หมายความว่า OSPF ไม่กิน Bandwidth ของเครือข่าย เนื่องจากว่าการ Update ข้อมูลเกี่ยวกับ Routing ให้แก่กัน จะเกิดขึ้นในกรณีที่มีการเปลี่ยนแปลงเกิดขึ้นบนเครือข่ายเท่านั้น และการ Update ให้แก่กัน ก็เฉพาะเนื้อหาที่มีการเปลี่ยนแปลงตามความเป็นจริงเท่านั้น
- **ให้การสนับสนุน VLSM และ CIDR** เช่นเดียวกับ EIGRP โปรโตคอลเลือกเส้นทาง OSPF สามารถกำหนดให้มี Subnet Mask รวมอยู่ในกระบวนการ Update เส้นทางให้กับ Router เครือข่ายอื่นด้วย
- **ไม่มีปัญหาเกี่ยวกับ Routing Loop**
- **ไม่จำกัดจำนวน Hop ดังเช่น RIP แต่หากจำกัด HOP ก็เป็น TTL Hop ของ TCP/IP มากกว่า**
- **สามารถทำ Load Balancing ได้ดีกว่า EIGRP รวมทั้ง IGRP** โดย Router OSPF สามารถทำ Load Balancing บนเส้นทางที่มีค่า Path Cost ที่เท่ากันได้มากถึง 6 เส้นทาง
- **สนับสนุน Authentication** OSPF สามารถให้บริการการเข้ารหัสข้อมูลข่าวสารที่ Router Update ให้แก่กันได้ โดยข้อมูลข่าวสารอาจถูกส่งในรูปของอักษรเปล่า (Plain Text) Password หรือเข้ารหัสแบบ MDS ก็ได้

เปรียบเทียบ OSPF กับ RIP

เปรียบระหว่าง OSPF กับ RIP มีข้อแตกต่าง ดังนี้

- OSPF มีขีดความสามารถในการขยายเครือข่ายได้ดีกว่า RIP
- ให้การสนับสนุน VLSM เมื่อเทียบกับ RIP V1 ซึ่งทำไม่ได้
- มีการเลือกเส้นทางที่ดีกว่า RIP
- ใช้ Metric ที่ล้ำหน้ากว่า RIP
- สามารถออกแบบเครือข่ายให้เชื่อมต่อแบบขั้นบันไดได้
- สามารถรับรู้การเปลี่ยนแปลงบนเครือข่ายได้ดีกว่า ไวกว่า

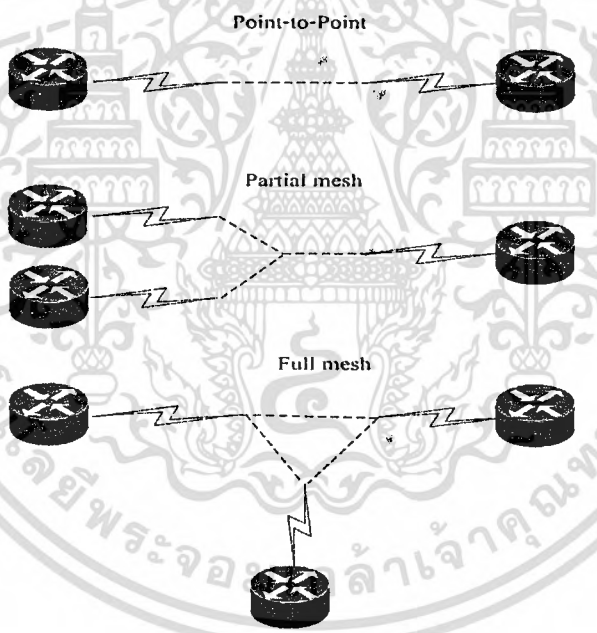
จุดด้อยของ OSPF เมื่อเทียบกับ RIP มีดังนี้

- การออกแบบเครือข่ายแบบขั้นบันไดจะไม่บังเกิดผล หากการออกแบบไม่ดี
- มีความซับซ้อนมากกว่า RIP
- กินกำลังของ CPU และหน่วยความจำใน Router มาก
- ต้องการทักษะการออกแบบ และความเข้าใจค่อนข้างมาก

2.5.1 ลักษณะเชื่อมต่อทาง Topology ที่ OSPF ให้การสนับสนุน

OSPF ให้การสนับสนุนการเชื่อมต่อเชิงกายภาพ ดังต่อไปนี้

- **ให้บริการเชื่อมต่อแบบ Fully Mesh** ในรูปแบบการเชื่อมต่อแบบ Fully Mesh นี้ Router ทุกตัวจะมี Connection ของมันเชื่อมต่อเข้ากับ Router ตัวอื่น ๆ แบบเต็มทุกช่องทาง ผ่านทางวงจรเสมือน ซึ่งอาจจะเป็นแบบ PVC หรือ SVC ก็ได้ โดย Topology การเชื่อมต่อแบบนี้ จะช่วยให้ได้ประโยชน์ในแง่ของ Redundant Link ได้โดยมี Delay ที่เกิดขึ้นน้อยที่สุด จำนวนของ Link ภายใต้ Fully Mesh นี้ สามารถคำนวณได้จากสูตรดังนี้ : (N x (N-1)) /2 โดย N หมายถึงจำนวนของ Router ดังนั้น หากภายใน Fully Mesh มี Router 10 ตัว จำนวนของ Link ทั้งหมด จะเท่ากับ (10 x (10-1) /2 = 45 Link
- **Partial Mesh** ในรูปแบบการเชื่อมต่อแบบ Partial Mesh นี้ ตัว Router ไม่ได้ใช้ Link ที่มีอยู่ทั้งหมดเชื่อมต่อระหว่างกัน ผ่านทางวงจรเสมือน



รูปที่ 2.19 ลักษณะการเชื่อมต่อ WAN ที่ OSPF ให้การสนับสนุน

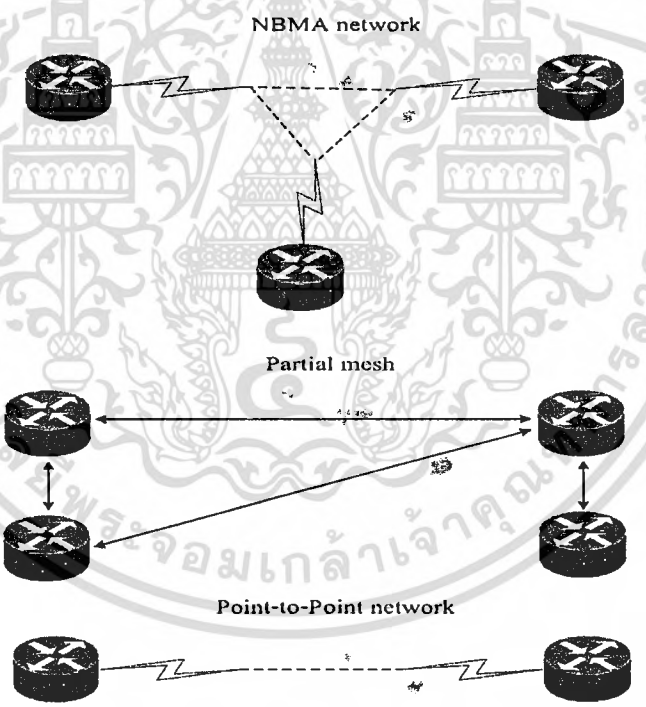
นอกจากการเชื่อมต่อ Topology ทางกายภาพแล้ว OSPF ยังให้การสนับสนุน ประเภทของเครือข่ายดังต่อไปนี้

- **Non-Broadcast Multi-access Network** เป็นเครือข่ายที่มีการเชื่อมต่อ Router มากมายหลายตัวเข้าด้วยกัน แต่เป็นการเชื่อมต่อที่ Router ทุกตัวไม่ได้รับการสื่อสารข้อมูลแบบ Broadcast โดยปริยาย โดยทั่วไปรูปแบบการเชื่อมต่อเครือข่ายแบบ Non-Broadcast Multi-access จะมี Topology การเชื่อมต่อแบบ Fully Mesh ตัวอย่างเครือข่ายที่เชื่อมต่อ

ภายใต้ Non-Broadcast Multi-access นี้ได้แก่ X.25 SMDS และ Frame-Relay Physical Interface รวมทั้ง ATM Physical Interface

- **เครือข่ายแบบ Point To Multipoint** เป็นเครือข่ายที่ประกอบด้วย Router ที่เชื่อต่อกันในลักษณะ Partial Mesh หรือกึ่ง Mesh การเชื่อมต่อแบบนี้ Router ไม่มีความสามารถที่จะสื่อสารแบบ Broadcast ได้โดยปริยาย ตัวอย่างรูปแบบของเครือข่ายได้แก่ เครือข่าย เช่น Frame Relay Sub-interface Point To Multipoint และ ATM Interface Sub-Interface Point To Multipoint

- เครือข่ายแบบ **Point To Point** รูปแบบการเชื่อมต่อแบบนี้ได้แก่การเชื่อมต่อ Router เพียง 2 ตัวเข้าด้วยกัน การเชื่อมต่อแบบนี้ ให้การสนับสนุนการสื่อสารแบบ Broadcast ด้วย ตัวอย่างการเชื่อมต่อเครือข่ายแบบนี้ ได้แก่ PPP HDLC และ T1/E1 เช่น การเชื่อมต่อภาพ Serial Interface โดยทาง Modem V.35 Frame Relay Sub Interface Point To Point และ ATM Sub-interface Point To Point



รูปที่ 2.20 แสดงภาพการเชื่อมต่อ Router เชิงตรรก ที่ OSPF ให้การสนับสนุน

รายละเอียดของเครือข่ายแต่ละแบบที่ OSPF ให้การสนับสนุนเครือข่าย OSPF แบบ Point To Point เป็นเครือข่ายที่ไม่มี DR (Designated Router) หรือ Router ตัวแทนประจำพื้นที่ของเครือข่าย มีการใช้ Address ที่เป็น Multicast (224.0.0.5) เพื่อสื่อสารกันระหว่าง Router เพื่อการสถาปานาการเชื่อมต่อกับ Router เพื่อนบ้าน

เครือข่าย OSPF แบบ Broadcast เป็นเครือข่ายที่มี Router ตัวแทนประจำพื้นที่ (DR) รวมทั้ง Router ตัวแทนชนิดสำรอง (BDR) อีกหนึ่งตัวอยู่ด้วยกันบนเครือข่าย และมีการใช้ Multicast Address 224.0.0.6 เพื่อสื่อสารกันระหว่าง Router ภายในเครือข่ายที่ไม่ใช่ DR กับ DR

เครือข่าย OSPF แบบ Non-Broadcast Multiaccess เป็นเครือข่ายที่มี DR ที่มีการสถาปนาการเชื่อมต่อระหว่าง DR กับ Router ภายในเครือข่ายที่ไม่ใช่ DR

เครือข่าย OSPF แบบ Point To Point เป็นเครือข่ายที่ไม่ต้องมี DR แต่ Router ภายในเครือข่ายแบบนี้ จะส่ง LSA ไปยัง Router ตัวอื่น ๆ ที่เชื่อมต่อแบบประชิดกัน และ OSPF จะใช้ค่า LSA เพื่อการกำหนดเส้นทางที่ดีที่สุด

ฐานข้อมูลภายใน Router ที่ทำงานภายใต้ OSPF

OSPF จะสร้างฐานข้อมูล 3 ประเภทภายใน Router ดังต่อไปนี้

1. ฐานข้อมูลเกี่ยวกับความเป็นอยู่ของ Router เพื่อนบ้าน (Neighbor Database)
2. ฐานข้อมูลเกี่ยวกับลักษณะการเชื่อมต่อภายในเครือข่ายทั้งหมด (Topology Database)
3. ฐานข้อมูลเกี่ยวกับเส้นทางการเชื่อมต่อ (Routing Table)

ฐานข้อมูลเกี่ยวกับ ความเป็นอยู่ของ Router เพื่อนบ้าน (Neighbor Database)

ภายในฐานข้อมูลนี้ ประกอบไปด้วย ข้อมูลข่าวสารเกี่ยวกับ Router เพื่อนบ้านที่ปัจจุบันยังมีตัวตน และมีสถานะการเชื่อมต่อที่ดีอยู่ ข้อมูลข่าวสารภายในฐานข้อมูลนี้ได้มาจากการที่ Router ได้ทำการสถาปนาการเชื่อมต่อระหว่าง Router เพื่อนบ้านทันทีที่เริ่มทำงาน

ฐานข้อมูลเกี่ยวกับลักษณะการเชื่อมต่อภายในเครือข่ายทั้งหมด (Topology Database)

บางครั้งเราจะเรียกฐานข้อมูลนี้ว่า Link State Database เป็นฐานข้อมูลที่เก็บสถานะการเชื่อมต่อในทุกเส้นทางบนเครือข่ายนี้ ที่ Router ได้เรียนรู้มาจาก Router อันเป็นเพื่อนบ้านทั้งหลาย หากจะเปรียบเทียบ Topology Database ได้แก่ แผนผังการเชื่อมต่อที่มีชีวิต เปลี่ยนแปลงได้ตามการเชื่อมต่อ ดังนั้น Router ใด ๆ ที่มี Topology Database อยู่ภายใน ก็เท่ากับว่ามีแผนผังอยู่ในใจที่ทำให้ Router สามารถเลือกเส้นทางได้อย่างสะดวกยิ่งขึ้น องค์ประกอบที่สำคัญภายใน Topology Database ได้แก่

- ชนิดของ LSA ภายใน
- Router ID ของผู้ที่ส่ง LSA ออกมาที่เครือข่าย
- Router ที่กำลังประกาศตัวเองออกมาที่เครือข่าย
- ต้นทุนเส้นทาง

LSA มีอยู่ 7 ชนิดที่ใช้อยู่ใน OSPF LSA แต่ละชนิดมีความแตกต่างกันขึ้นอยู่กับชนิดของข้อมูลที่มากับ LSA นั้น ๆ

ฐานข้อมูลเกี่ยวกับเส้นทางการเชื่อมต่อ (Routing Table)

เป็นข้อมูลข่าวสารเกี่ยวกับเส้นทางที่ดีที่สุด ที่ Router ใช้เพื่อเดินทางไปสู่ปลายทาง ภายใน Routing Database ประกอบด้วยข้อมูลข่าวสาร ดังนี้

- เครือข่ายปลายทางพร้อมด้วย Subnet Mask
- ระยะทางที่ดีที่สุด ยึดหยุ่นที่สุด
- Address ของ Hop ต่อ ๆ ไป
- เวลาที่ผ่านมาหลังจากการ Update เป็นครั้งสุดท้าย
- Interface ที่ใช้เพื่อการเดินทางไปสู่ Hop ต่อไป

2.5.2 การคิดค่า Cost ของ OSPF

OSPF ทำงานบนพื้นฐานของการเลือกเส้นทางที่สั้นที่สุด โดย OSPF จะมีการจ่ายค่า Cost ให้กับ Interface ของ Router โดยอัตโนมัติ โดยมีค่า Default ที่จ่ายให้กับ Interface ของ Router บนพื้นฐานของสูตรดังต่อไปนี้

OSPF Cost = 100,000,000 / Bandwidth ของ Interface

ค่า OSPF Cost ที่จ่ายให้กับ Router โดยมาตรฐานมีดังนี้

ชนิดของการเชื่อมต่อ	ความเร็ว (หารด้วย 10 ยกกำลัง 8)	ค่า Cost ของ OSPF
Serial	56,000	1785
DSO	64,000	1562
T1	1,544,000	65
E1	2,048,000	48
Token Ring 4 Mbpy	4,000,000	25
Ethernet 10 Mbps	10,000,000	10
Token Ring ความเร็ว 16 Mbps	16,000,000	6
T3	44,736,000	3
Fast Ethernet	100,000,000	1
Gigabit Ethernet	1,000,000,000	1
OC-3	155,520,000	1
OC-12	622,080,000	1

ในการจัด Configure Router ภายใต้การทำงานของ OSPF จะต้องกำหนดค่าBandwidth ให้ตรงกับ Interface ให้ตรงตามความเป็นจริง หากไม่ทำเช่นนั้น Router จะเป็นผู้ให้ค่า Cost ของ Interface โดยปริยาย ตามค่าในตารางที่ได้กล่าวมาแล้ว เช่นเดียวกับที่สามารถจัดตั้งค่า Cost ด้วยเอง

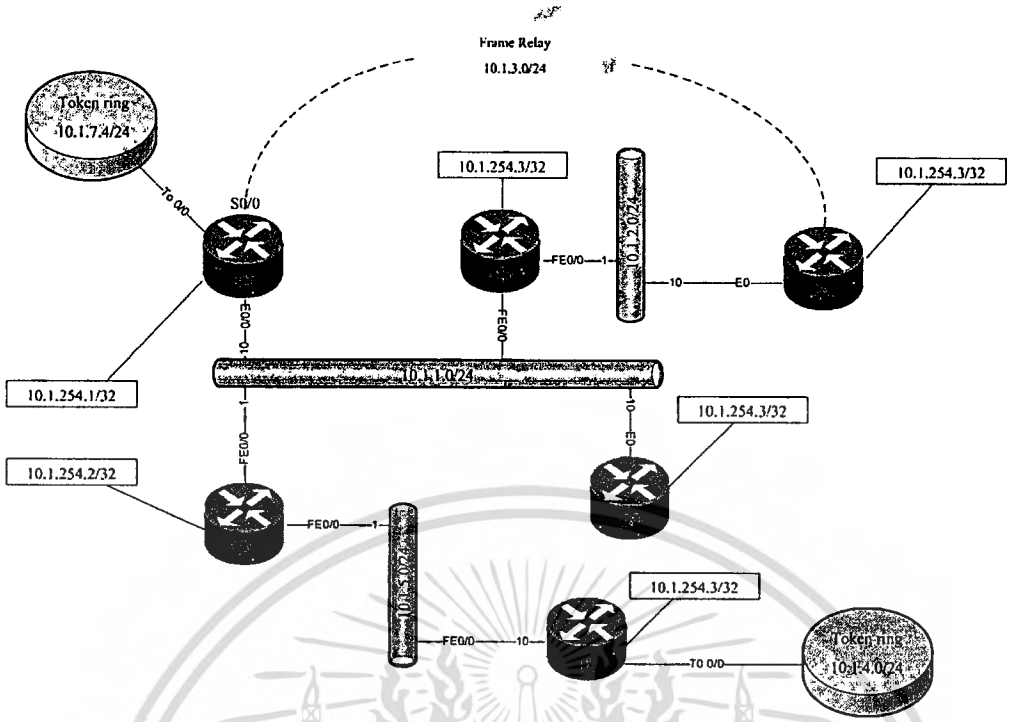
Cost ด้วยเอง และเนื่องจาก Router สามารถสนับสนุน Interface และการเชื่อมต่อที่มีความเร็วสูงกว่า 100 Mbps ดังนั้น Router จึงสามารถถูดจัด Configure ด้วยคำสั่ง auto-cost reference-bandwidth 1000 router configuration โดยคำสั่งนี้จะทำให้ค่า Bandwidth อ้างอิง ซึ่งแต่เดิมมีแค่ 100 Mbps ให้เป็น 1000 Mbps ได้ และด้วย Bandwidth อันใหม่นี้เอง จึงทำให้ Fast Ethernet มีค่า Metric Cost อยู่ที่ 10 ส่วน Ethernet จะมีค่า OSPF Cost อยู่ที่ 100 ซึ่งแน่นอนค่า Metric Cost ยิ่งน้อยยิ่งดี เนื่องจาก Router OSPF จะเลือกค่า Cost น้อย ๆ เป็นหลักอยู่เสมอ

การทำงานแบบ Short Path First

OSPF มีความแตกต่างจาก Distance Vector Protocol ตรงที่ OSPF ไม่ส่งผ่านข่าวสารเกี่ยวกับเส้นทางไปยัง Router เพื่อนบ้าน แต่ OSPF Router จะส่งค่า LAS หรือ Link State Advertisement ไปยัง Router เพื่อนบ้านแทนข้อมูลข่าวสารภายในของ LSA เป็นข้อมูลเกี่ยวกับสถานะการเชื่อมต่อข้อมูลนี้ Router เพื่อนบ้านที่ได้รับ จะนำไปสร้างฐานข้อมูลที่เกี่ยวกับสถานะการเชื่อมต่อระหว่างกัน หรือที่เรียกว่า Link State Database นอกจากนี้ข่าวสารที่เกี่ยวข้องกับ LSA ยังครอบคลุม Router ID และ Link State ID เช่น Address ของ LAN หรือ Serial Link Subnet Address รวมทั้งชนิดของ LSA และค่า Metric Cost ของ OSPF

อย่างไรก็ดี ค่า LSA ไม่ใช่ข้อมูลเกี่ยวกับตารางเส้นทาง แต่เป็นข้อมูลที่ Router นำมาใช้เพื่อประกอบการสร้างตารางเลือกเส้นทาง เพื่อให้ได้ข้อมูลเกี่ยวกับตารางเส้นทาง Router จะต้องนำค่า LAS นี้มาทำการคำนวณด้วยอัลกอริทึม ที่เรียกว่า Dijkstra หรือการคำนวณแบบ Short Path First โดย Router จะมองตัวมันเองเป็นรากของต้นทาง ที่ Router จะเดินทางไปหาปลายทาง จากนั้นมันจะใช้ค่าที่ได้รับจาก Router รอบข้างมาทำการคำนวณช้า ๆ แบบทีละขั้นตอน เพื่อดูว่ามีเส้นทางใดบ้าง ที่มีระยะที่สั้นที่สุด ถัดจากตัวมันลงไป กระบวนการนี้เกิดขึ้นอย่างต่อเนื่องจนกว่าสามารถคำนวณเส้นทางได้ครบทุกเครือข่าย

และเพื่อการคำนวณเพื่อเลือกเส้นทางที่สั้นที่สุด สามารถเกิดขึ้นได้อย่างไม่มีปัญหา ดังนั้นหากมีการเปลี่ยนแปลงใด ๆ เกิดขึ้นบนเครือข่าย การคำนวณแบบ Short Path First นี้ จะไม่กระทำในทันที แต่จะรอกสัก 5 วินาทีก่อน เนื่องจากถือโอกาสรอให้มีการ Update ข่าวสารอื่นพร้อมกัน อย่างไรก็ดีสามารถจัดตั้ง Configure เพื่อกำหนดเวลาเริ่มการคำนวณหลังจากที่เครือข่ายมีการเปลี่ยนแปลงได้ แต่การทำเช่นนี้ จะต้องระวังเนื่องจากอาจก่อให้เกิดผลกระทบต่อประสิทธิภาพการทำงานของเครือข่ายได้

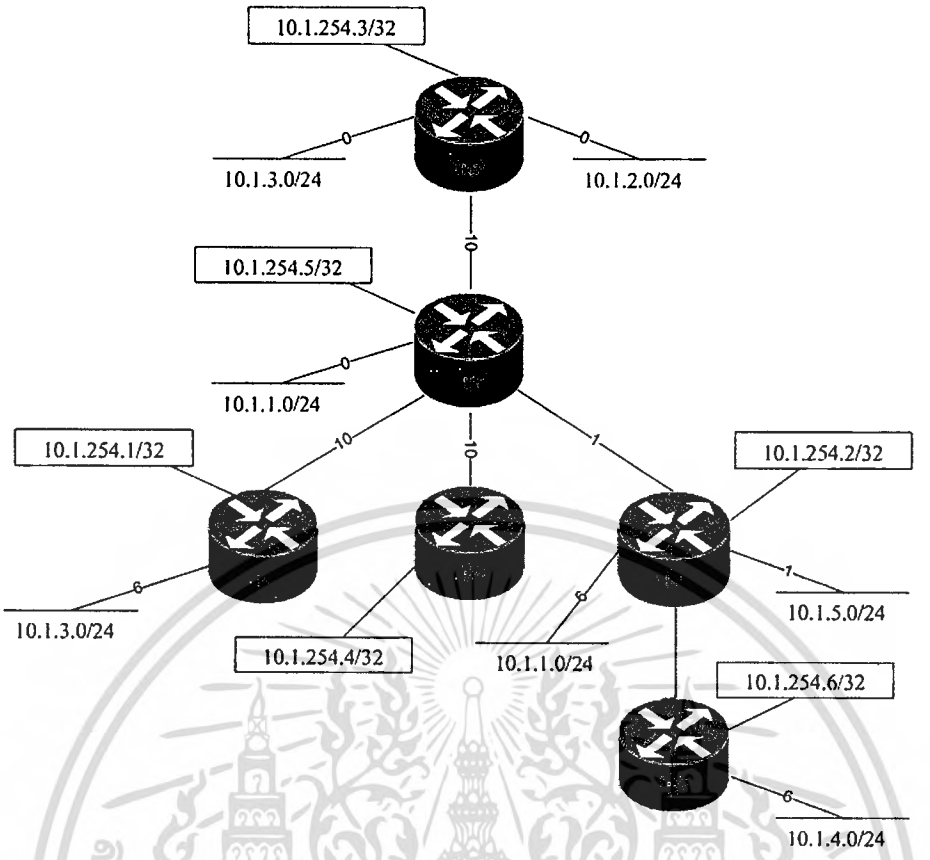


รูปที่ 2.21 แสดงค่า Metric Cost และวิธีการเลือกเส้นทางแบบ Short Path First

จากรูปที่ 2.21 สมมติว่า Router ทุกตัวทำงานภายใต้ OSPF หลังจากที่มีการคำนวณด้วยวิธีการ Short Path First แล้ว รูปแบบของ Routing Tree (ลักษณะที่ Router มองตนเป็นรากของต้นไม้ และเชื่อมต่อจากรากไปสู่กิ่งก้านสาขา โดยจะมอง Router หรือเครือข่ายที่ใกล้มีระยะทางสั้นที่สุด ซึ่งหมายถึงค่า Metric Cost ต่ำไปจนถึงด้านปลายของต้นไม้ที่เป็นเครือข่ายปลายทาง) ของ R3 จะเป็นไปตามรูปที่ 2.22 และแม้ว่า R3 กับ R1 มีการเชื่อมต่อกันโดยตรงด้วย Frame Relay ที่มีความเร็ว1544 K แต่ R3 จะไม่เลือกเส้นทางนี้ เนื่องจากมีค่า Metric Cost เท่ากับ 65 ขณะที่ R5 เชื่อมต่อกับR3 ด้วย Fast Ethernet และมีค่า Metric Cost ที่ 1 ในการนี้ เส้นทางที่สั้นที่สุดในมุมของ R3 ที่จะเดินทางไป R6 ได้แก่ การเลือก R5 จากนั้นผ่านไปยัง R2 และตรงเข้าสู่ R6 (รูปที่ 2.22)

ถ้าหากใช้คำสั่ง sh ip route ที่ Router R3 จะเห็นข่าวสารปรากฏดังนี้

```
      10.0.0.0/8  is variably subnetted,  7 subnets, 2 masks
C       10.1.3.0/24  is directly connected, Serial
O       10.1.4.6/32 [110/18] via  10.1.2.5,  00:00:08,  Ethernet0
C       10.1.2.0/24  is directly connected, Ethernet0
O       10.1.1.0/24 [110/11] via  10.1.2.5,  00:00:08  Ethernet0
O       10.1.7.0/24 [110/17] via  10.1.2.5,  00:00:08,  Ethernet0
O       10.1.5.0/24 [110/12] via  10.1.2.5,  00:00:08,  Ethernet0
C       10.1.254.3/32 is directly connected, Loopback1
```



รูปที่ 2.22 แสดงลักษณะการเลือกเส้นทางของ Router R3 หลังจากที่ใช้ Short Path First คำนวณเส้นทางแล้ว

ชนิดของ Packet ที่ OSPF ใช้งานระหว่าง Router

ชนิดของ Packet ภายใต้ OSPF	คำอธิบาย
Type 1 Hello	ใช้เพื่อการสถาปนา และดูแลรักษาตัวตนของ Router เพื่อนบ้าน
Type 2 Database Description Packet	ประกอบด้วยเนื้อหาเกี่ยวกับ Link State Database ของ OSPF
Type 3 Link State Request	ร้องขอข้อมูลบางส่วนอย่างเฉพาะเจาะจงที่อยู่ใน Link State Database ของ Router
Type 4 Link State Update (LSU)	ใช้ขนถ่าย LSA ไปยัง Router เพื่อนบ้าน
Type 5 Link State Acknowledgement	ตอบรับหลังจากได้รับ LSU แล้ว

2.5.3 Designated Router ภายใต้ OSPF

ภายใต้ OSPF ไม่ว่าจะเป็นเครือข่ายแบบ Broadcast หรือ Non-Broadcast Multi-access (NBMA) แต่ละเครือข่ายจะต้องมี Router ตัวหนึ่งทำหน้าที่เป็น Designated Router (DR) ตัวนี้จะ

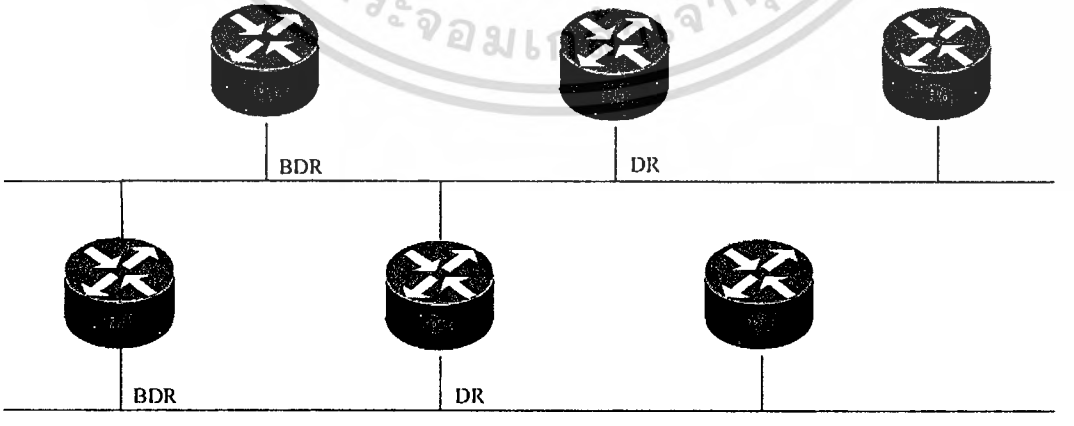
ถูกเลือกโดยอัตโนมัติด้วย Hell Protocol ของ OSPF ลักษณะการเลือกจะอาศัยค่าลำดับความสำคัญสูงสุด ที่มากับ Router หรืออาจจัด Configure ด้วยตัวเองก็ได้ โดยการกำหนดให้ Router ที่ต้องการให้เป็น DR ให้มีค่า Priority สูงกว่า Router ตัวอื่น ๆ เมื่อ Router ถูกจัดตั้งเป็น DR แล้วจะไม่มีการเลือก DR ขึ้นมาอีก แม้ว่าในภายหลังจะมี Router ที่มีค่าลำดับความสำคัญสูงกว่าเข้ามาในเครือข่ายก็ตาม อย่างไรก็ดีสามารถเลือก Router ที่เป็น BDR ก็ได้ ในกรณีที่ Router DR เกิดล่ม ตัว BDR ก็สามารถทำงานแทนได้โดยอัตโนมัติ หากต้องการดูว่าในเครือข่ายมี Router ใดที่เป็น DR หรือ BDR สามารถใช้คำสั่ง show ip ospf neighbor เพื่อตรวจสอบดู จะปรากฏตัวอย่างหน้าจอ ดังนี้

Router R5

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.1.254.1	1	Full/Drother	00:00:35	10.1.1.1	FastEthernet0/0
10.1.254.2	1	Full/Drother	00:00:38	10.1.1.2	FastEthernet0/0
10.1.254.4	1	Full/Drother	00:00:35	10.1.1.4	FastEthernet0/0

Router R2

10.1.254.6	1	Full/DR	00:00:30	10.1.5.6	FastEthernet0/0.7
10.1.254.5	5	Full/DR	00:00:35	10.1.1.5	FastEthernet0/0.8
10.1.254.1	1	2Way/Drother	00:00:34	10.1.1.1	FastEthernet0/0.8
10.1.254.6	1	Full/BDR	00:00:34	10.1.1.4	FastEthernet0/0.8



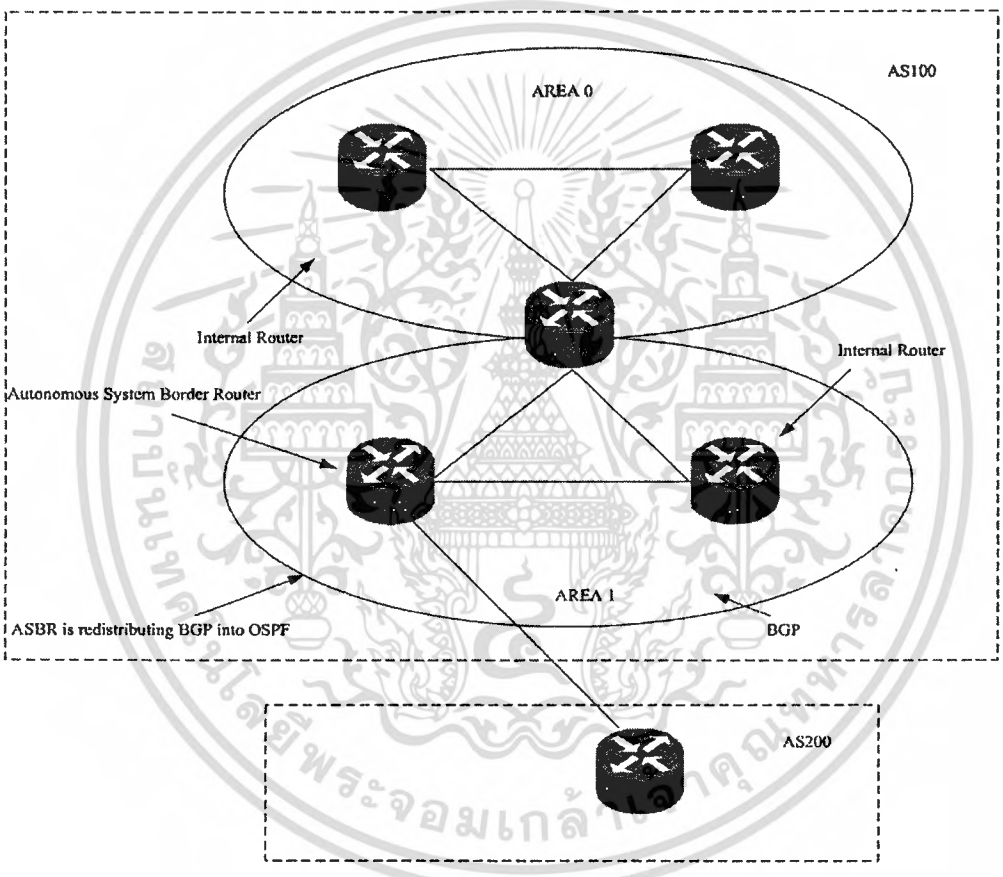
รูปที่ 2.23 แสดงการเชื่อมต่อของ Router ต่าง ๆ ทั้งที่เป็นแบบ DR และ BDR

2.5.4 Router ID ใน OSPF (RID)

OSPF Router ID นับว่าเป็นองค์ประกอบที่สำคัญในโปรโตคอลการทำงานของ OSPF ค่า Router ID ของ OSPF มีขนาด 32 บิต และจะต้องมีค่าที่ไม่ซ้ำกันในเครือข่ายอย่างเด็ดขาด ค่า Router ID นี้หากไม่ได้ถูกจัด Configure ภายใต้กระบวนการทำงานของ OSPF แล้ว กระบวนการ OSPF จะเป็นผู้เลือก IP Address ที่มีค่าสูงที่สุดที่ได้ตั้งให้กับ Router แล้วเลือกเป็น Router ID และเมื่อใดที่ Router ID ถูกจัดตั้งขึ้นให้กับ Router แล้ว จะไม่มีการเปลี่ยนแปลงใด ๆ เกิดขึ้นอีก

2.5.5 ชนิดของ Router ที่ถูกเรียกใช้ใน OSPF

ภายใต้ OSPF ได้มีการกำหนดให้มี Router อยู่ 4 ชนิด ได้แก่

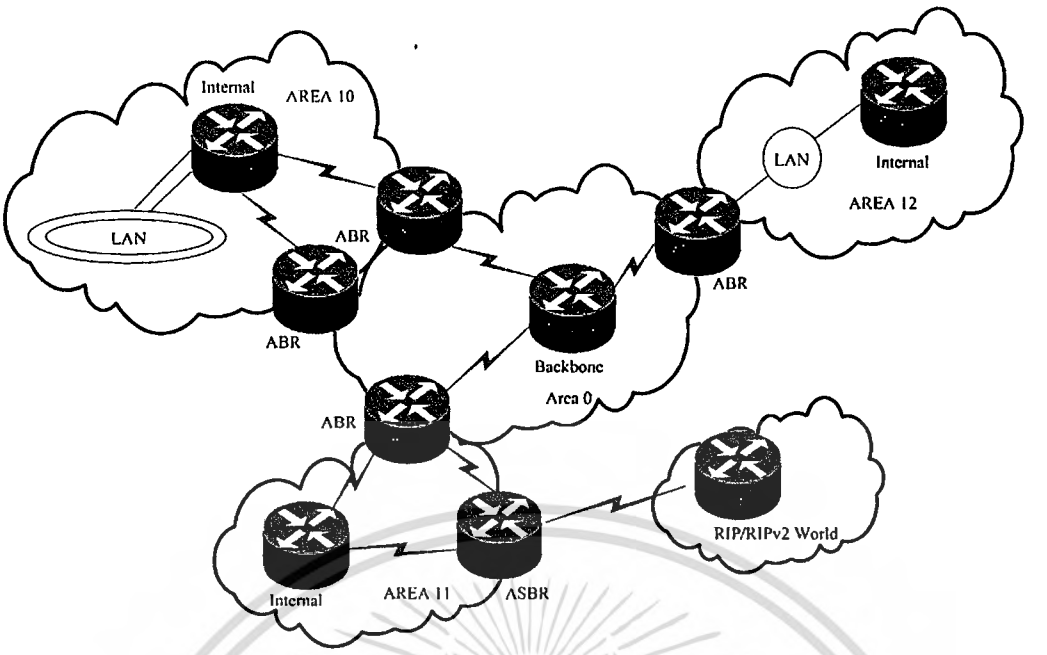


รูปที่ 2.24 แสดงชนิดของ Router ที่ใช้ภายใน OSPF

Internal Route Internal Router เป็น Router ที่มี Interface ทั้งหมด ถูกกำหนดให้อยู่ภายในพื้นที่ (Area) หนึ่ง Internal Router ที่เป็นของ Area 0 จะมี OSPF Interface ทั้งหมดกำหนดให้อยู่ใน Area 0

Area Border Router (ABR) เป็น Router ที่จะต้องมี Interface ใด Interface หนึ่งที่เชื่อมต่อกับส่วนที่เป็น Area 0 หรือ Backbone Area และมี Interface อย่างน้อยอีกหนึ่งที่เชื่อมต่อกับ Area อื่น ๆ หมายความว่า ABR เป็น Router ที่ทำหน้าที่เชื่อมโยงระหว่าง Area ต่าง ๆ ให้เชื่อมต่อกับ Area 0

Backbone router เป็น Router ที่อยู่ภายใน Area 0 ทำหน้าที่เชื่อมต่อ Area ต่าง ๆ เข้าด้วยกัน



รูปที่ 2.25 แสดงการจัดวางตำแหน่งของ Router ชนิดต่าง ๆ ภายใต้ OSPF

2.5.6 ชนิดของ Area

เนื่องจากเครือข่าย OSPF เป็นเครือข่ายที่มีขนาดใหญ่ ไม่จำกัดจำนวน Hop เช่นเดียวกับ Distance Vector ดังนั้น เพื่อให้สะดวกต่อการบริหารจัดการเครือข่าย รวมทั้งรักษาประสิทธิภาพการทำงานด้าน Bandwidth ดังนั้น ไม่เพียงแต่แบ่งชนิดของ Router ที่จะใช้งานบน OSPF แล้ว ยังต้องแบ่งพื้นที่ออกเป็นส่วนต่าง ๆ เรียกว่า Area เพื่อให้ง่ายต่อการจัดการ และเพิ่มประสิทธิภาพในการแลกเปลี่ยนข้อมูลข่าวสารระหว่าง Router ได้ดีมากยิ่งขึ้น โดยมีการแบ่งชนิดของ Area ออกเป็นแบบต่าง ๆ ดังนี้

- **Standard Area** เป็นพื้นที่ ๆ ใช้บ่อยมากที่สุด พื้นที่ใด ๆ ที่ไม่ใช่พื้นที่ Backbone ถือว่าเป็น Standard Area ได้ทั้งสิ้น นอกจากนี้รูปแบบบางประการของ Stub Area ก็จัดว่าเป็น Standard Area ได้เช่นกัน Standard Area ให้การสนับสนุน LSA ตั้งแต่ Type 1-Type 5
- **Backbone Area หรือ Area 0** หน้าที่หลักของ Back Bone Area ได้แก่การส่งผ่าน Traffic ต่าง ๆ ระหว่าง Area ต่าง ๆ ทำหน้าที่เป็นกาวใจเชื่อมสัมพันธ์ระหว่าง Area รวมทั้งให้บริการส่งผ่านข้อมูลข่าวสารระหว่าง Area เครือข่ายใดก็ตามที่ประกอบด้วยหลาย Area จำเป็นต้องมี Backbone Area เพื่อเชื่อมต่อ Area ต่าง ๆ เข้าด้วยกัน
- **Transit Area** เป็นพื้นที่ ๆ บรรดา Traffic ที่มาจากพื้นที่อื่น ๆ ต้องอาศัยพื้นที่นี้เป็นทางผ่านไปสู่พื้นที่อื่น ๆ เรียกว่า Transit Area โดย Backbone Area ก็จัดว่าเป็น Transit Area ได้เช่นกัน
- **Stub Area** เป็นพื้นที่เดียวที่สามารถเดินทางผ่านไปสู่ภายนอก (Autonomous system อื่น ๆ) ด้วยเหตุนี้ Stub Area จึงไม่ต้องการ Type 4 และ Type LAS

- **Totally Stub Area** เป็นพื้นที่เดียวที่สามารถเดินทางออกสู่ Autonomous System (AS) อื่น ๆ รวมทั้งพื้นที่อื่น ๆ ด้วยภายใต้ Totally Stub Area มีเพียงช่องทางเดียวเท่านั้น ที่สามารถเข้าถึงปลายทางซึ่งเชื่อมต่อกับพื้นที่อื่น ๆ ได้ ด้วยเหตุนี้ Totally Stub Area จึงไม่ต้องการใช้ LSA แบบ Type 1 Type 4 หรือ Type 5 Totally Stub Area เป็นมาตรฐานที่นิยามขึ้นโดย Cisco
- **Not So-Stubby Area (NSSA)** เป็นพื้นที่ ๆ ต้องการการแพร่ข่าวสารของ LSA ภายนอกที่มาจาก ASBR เข้ามายังพื้นที่ภายใน และมีเพียงเส้นทางเข้าออกเพียงเส้นทางเดียวเพื่อไปสู่ ASBR ในพื้นที่อื่น ๆ ASBR เป็น Router ที่ทำงานภายใต้ Distance Vector Protocol ที่ใช้ Autonomous System อย่างเช่น EIGRP หรือ IGRP ดังนั้น การที่ Router จากระบบที่ใช้ Autonomous System ต้องการใช้พื้นที่ต่าง ๆ เป็นทางผ่าน จะต้องใช้ Not So Stubby Area เพื่อเป็นช่องทางผ่านระหว่าง AS หนึ่งไปยังอีก AS หนึ่ง ภายใต้พื้นที่ของ OSPF พูดง่าย ๆ คือขออาศัยพื้นที่ในลักษณะ NSSA นี้เพื่อสื่อสารระหว่าง AS ด้วยกัน หรือใช้เพื่อสื่อสารกับเครือข่ายในระบบ OSPF ก็ได้เช่นกัน

2.5.7 ระบบ Link Advertisement (LSA) ของ OSPF

OSPF เป็นโปรโตคอลที่ค่อนข้างเน้นเรื่องโครงสร้างที่ประกอบด้วย LSA หลายชนิด การที่มี LSA หลายชนิดเป็นการทำให้มั่นใจว่า ฐานข้อมูลเกี่ยวกับสถานะกาเชื่อมต่อ ถูกจัดตั้งไว้ในที่ที่เหมาะสม (ในที่ ๆ Router ติดตั้งอยู่ในพื้นที่ (Area) ต่าง ๆ กัน) Router ทุก ๆ ตัว จะต้องมีฐานข้อมูลของสถานะการเชื่อมต่อที่คล้ายคลึงกัน ต่อไปที่คือชนิดของ LSA ภายใต้ OSPF

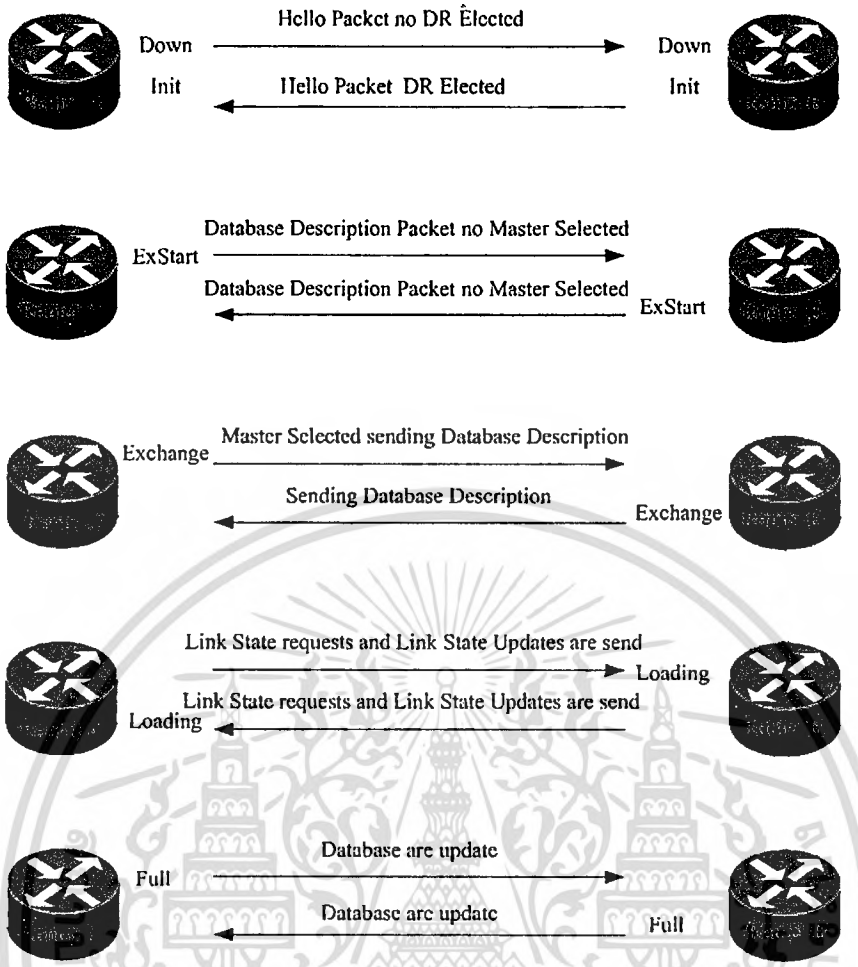
- **LSA Type 1 (Router LSA)** LSA ชนิดนี้ถูกสร้างขึ้นโดย Router และเป็นการแสดงตัว Router และ Interface ของมันเป็นข่าวสารสถานะการเชื่อมต่อที่ส่งกันไปมาระหว่าง Router ภายในพื้นที่เดียวกัน
- **LSA Type 2 (Network LSA)** เป็นข่าวสารเกี่ยวกับสถานะการเชื่อมต่อที่มาจาก DR Router ซึ่งติดตั้งอยู่ในพื้นที่ ๆ มี Router เชื่อมต่อเข้ากับ DR หลายตัว ภายใน LSA นี้ประกอบด้วยรายชื่อของ Router ที่มีอยู่ทั้งหมด (Router ทุกตัวที่เชื่อมต่อกันภายในพื้นที่ ๆ DR ติดตั้งอยู่) ข่าวสารของ LSA ชนิดนี้จะถูกส่งออกไปยังทุกเส้นทางภายในพื้นที่เท่านั้น และจะไม่ถูกนำส่งออกไปนอกพื้นที่
- **LSA Type 3 (Summary LSA)** เป็น LSA ที่กำเนิดขึ้นจาก Area Border Router (ABR) และถูกส่งออกไปที่ Router ทุกตัวที่เชื่อมต่อกันภายใน LSA นี้เป็นข่าวสารที่ระบุเส้นทางที่ Router ภายในพื้นที่ทราบว่าจะออกไปจากพื้นที่ ๆ ตนอยู่ได้อย่างไร LSA นี้จะถูกส่งออกไปที่ Router ทุกตัวภายในพื้นที่

- **LSA Type 4 (ASBR Summary LSA)** กำเนิดโดย Area Border Router เช่นกัน แต่ภายในประกอบด้วยข้อมูลข่าวสารเกี่ยวกับเส้นทางที่สามารถเดินทางไปสู่ Autonomous System Boundary Router (เป็น Router ที่เป็นตัวแทนของระบบเครือข่ายที่ใช้โปรโตคอลเลือกเส้นทางแบบ Distance Vector) LSA นี้จะถูกส่งออกไปโดย ABR ไปยัง Router ทุกตัวที่เชื่อมต่อกับพื้นที่
- **LSA Type 5** ถูกสร้างขึ้นโดย ASBR เป็น LSA ที่จะถูกส่งออกไปยัง Router ทุกตัวที่อยู่ใน OSPF ประกอบด้วยข้อมูลข่าวสารเกี่ยวกับเส้นทางที่จะไปสู่เครือข่ายที่เป็นระบบ Autonomous System
- **LSA Type 7 (NSSA External LSA)** กำเนิดโดย ASBR โดย ASBR นี้เชื่อมต่อกับพื้นที่ ๆ เรียกว่า Not So Stubby Area (NSSA) และให้กำเนิด LSA Type 7 และ ABR ภายในพื้นที่นี้จะแปลง LSA Type 7 และจัดส่ง LSA Type 5 ไปยังส่วนที่เหลือภายใน OSPF

2.5.8 ขบวนการสถาปนาการเชื่อมต่อ Router ที่อยู่ภายใต้ OSPF

เพื่อให้ Router ที่ทำงานภายใต้ OSPF สามารถมองเห็นและรู้จักกันและกันนั้น จำเป็นจะต้องมีการสถาปนาการเชื่อมต่อระหว่างกันเกิดขึ้น การทำงานเช่นนี้ นับว่าเป็นกุญแจสำคัญของ OSPF ขั้นตอนการทำงานที่สำคัญพอที่จะสรุปเป็นข้อ ๆ ดังนี้

1. สถาปนาการเชื่อมต่อกับ Router ที่อยู่ประชิดกัน
2. เลือกตั้ง DR และ DBR (หากจำเป็น)
3. ค้นพบ/ค้นหาเส้นทาง
4. เลือกเส้นทางที่เหมาะสมที่จะเอามาใช้
5. ดูแลรักษาตารางเลือกเส้นทาง (Routing Table)
6. เตือนให้เครือข่ายรู้ถึงการเปลี่ยนแปลง
7. สร้างและประสานความถูกต้องของข้อมูลในฐานข้อมูลระหว่าง Router ด้วยกัน



รูปที่ 2.26 แสดงขั้นตอนการสถาปนาการเชื่อมต่อระหว่าง Router เพื่อบ้าน

จากรูปที่ 2.26 เมื่อ Router A และ B เริ่มทำงานในครั้งแรก เริ่มด้วย Router A จะมีการเตรียมการและส่ง Hello Packet ออกมาที่ Router B ภายในประกอบด้วยข่าวสารที่แสดงความมีตัวตนของ Router A พร้อมด้วยข่าวสารที่บอกว่าไม่ได้รับเลือกตั้งเป็น DR ขณะเดียวกัน Router B ซึ่งถูกกำหนดให้เป็น DR จะส่ง Hello Packet ไปทักทาย Router A เช่นเดียวกัน แต่ได้แจ้งให้ทราบว่าตนได้รับการจัดตั้งเป็น DR ให้ Router A ทราบ จากนั้น Router ทั้งสองจะแลกเปลี่ยนฐานข้อมูลที่เกี่ยวกับการเชื่อมต่อให้แก่กัน และจากนั้นทำการ Update ฐานข้อมูลเกี่ยวกับการเชื่อมต่อระหว่างกันต่อไป

ภายใน Hello Packet ประกอบด้วยข้อมูลข่าวสารเบื้องต้น ดังนี้

Router ID เป็นค่าที่จะต้องไม่ซ้ำกันภายในเครือข่าย

Router Hello Timer เป็นตัวตั้งเวลาเพื่อส่ง Hello ออกไปตรวจสอบความมีตัวตนของ Router มีห้วงเวลา 10s

Router Dead Timer เป็นตัวตั้งเวลาที่ถูกกำหนดขึ้นหลังจากที่ไม่ได้รับตอบสนองจาก Router มีห้วงเวลาคิดเป็น 4 เท่าของค่า Hello Timer

เลขหมายแสดงลำดับความสำคัญของ Router ใช้ในกระบวนการเลือกตั้ง Router ที่จะเป็น DR หากไม่ได้กำหนดค่า Priority ให้กับ Router ตัว DR จะเป็น Router ที่มีค่า Router ID มากที่สุด และ BDR จะเป็น Router ที่มีค่า ID ที่สูงเป็นอันดับสองรองลงมาจาก DR

IP Address ของ DR

IP Address ของ BDR

OSPF Area

Network Mask ของ Interface ที่จัดส่งข้อมูลออกไป

ค่าที่แสดงสถานะของ OSPF Stub

ชนิดของ OSPF Authentication

บทที่ 3

MPLS NETWORK

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการวิจัยเช่นพื้นฐานของ MPLS Network ซึ่งเนื้อหาในบทนี้จะกล่าวถึงหลักการทำงานของ MPLS/VPN , MPLS Traffic Engineering และ Quality of Service (QoS) ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการศึกษา และประเมินประสิทธิภาพของระบบ MPLS

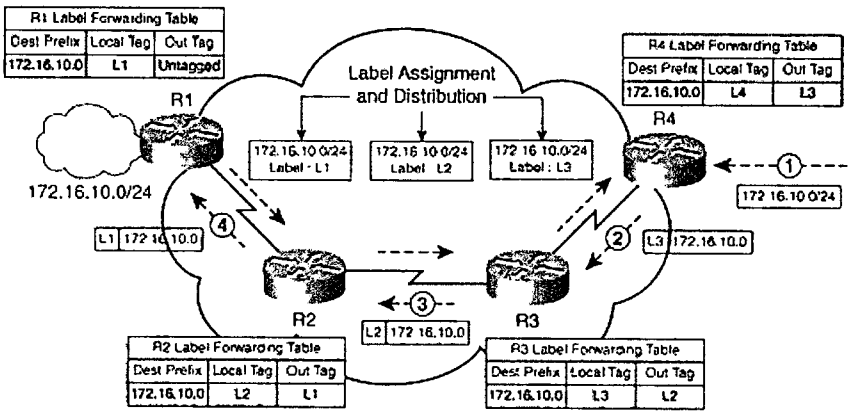
3.1 MPLS Network

Multi Protocol Label Switch (MPLS) ถูกสร้างขึ้นมาจากอุตสาหกรรมครือข่ายคอมพิวเตอร์โดยมีการใช้งานกันอย่างกว้างขว้างในกลุ่มครือข่ายของ Service provider (SP) MPLS สามารถแก้ปัญหาต่างๆมากมายที่พบเจอในครือข่ายปัจจุบัน ทั้งความเร็ว คุณภาพการให้บริการ (Quality of Service) และ Traffic Engineering (TE)

3.1.1 MPLS Forwarding

ในระบบครือข่าย MPLS ข้อมูลที่ส่งจะถูกเพิ่ม Labels โดย Labels นี้เป็นลักษณะเช่นเดียวกันกับ IP Address ปลายทาง ค่าของ Labels จะกำหนดเป็นต่อ Router (และในบางกรณี Labels จะกำหนดเป็นต่อ Interface บน Router) Router กำหนด Labels และกำหนดเส้นทาง ที่เรียกว่า Label Switch Paths (LSP) ระหว่าง ต้นทางไปยังปลายทาง

รูปที่ 1 แสดงการทำงานของ MPLS Forwarding เริ่มจาก PE Router R1 และ R4 Routers ในเครือข่าย MPLS R1, R2, R3 ประกาศ Update เครือข่าย 172.16.10.0/24 ผ่าน IGP Routing Protocol ไปในเส้นทางเดิมของระบบเครือข่าย IP สมมุติว่าไม่มีการกำหนด Filters หรือ Summarization Router ก็จะทำการสร้างตาราง IP Forwarding เส้นทางที่เชื่อมต่อไปยัง Router MPLS และกำหนด Local Labels สำหรับเครือข่ายปลายทาง 172.16.10.0 โดยทำการเผยแพร่ Labels ของเครือข่ายปลายทาง 172.16.10.0 ให้ Router ข้างเคียงทราบไปทาง Upstream ด้วย Labels Distribution Protocols ตัวอย่างเช่น R1 กำหนด Local Labels เป็น L1 และเผยแพร่ไป Upstream ให้ R2 และ R2 ส่งต่อให้ R3 กำหนดให้เผยแพร่เหมือนกันไปทาง Upstream คือ R4 ซึ่งกระบวนการนี้ทำให้ Router สามารถสร้าง Label Forwarding Table เพื่อใช้ในการส่งต่อ Labels Packet



รูปที่ 3.1 Forwarding .ใน MPLS Domain

จากรูปที่ 3.1 แสดงการส่งข้อมูลจากเส้นทาง R4 ไปยัง R1 ซึ่งมีการทำงานดังนี้

- R4 ได้รับข้อมูล Packet จากเครือข่าย 172.16.10.0 และกำหนดเส้นทางต่อไปยังปลายทางของ MPLS ที่สร้างขึ้น ดังนั้น R4 ส่ง Packet ข้อมูลต่อไปยัง Next-Hop Router R3 ด้วย Labels L3 (ไปทาง Downstream Router R3)
- R3 ได้รับ Labels Packet ด้วย Label L3 และเปลี่ยนจาก Label L3 เป็น L2 และส่ง Packet ข้อมูล ต่อให้ R2
- R2 ได้รับ Labels Packet ด้วย Label L2 และเปลี่ยนจาก Label L2 เป็น L1 และส่ง Packet ข้อมูล ต่อให้ R1
- R1 เป็น Router ตัวสุดท้าย (Border Router)ใน MPLS Domain ซึ่งอยู่ระหว่าง IP และ MPLS Domain ดังนั้น R1 จึงเอา Labels ออก เหลือแต่ Packet ที่เป็น IP ปกติและส่งข้อมูลให้ปลายทางเครือข่าย 172.16.10.0

3.1.2 ส่วนประกอบสถาปัตยกรรมของ MPLS

ประกอบด้วย 2 ส่วนประกอบหลักคือ

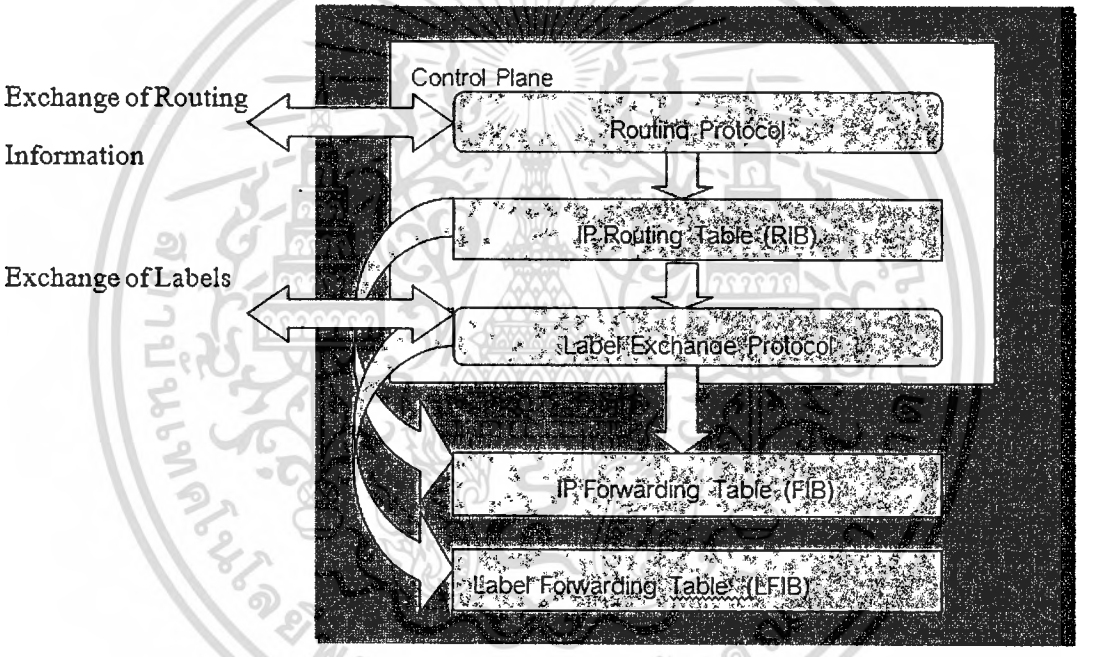
- **Control plane** ทำหน้าที่ในควบคุมกระบวนการในการแลกเปลี่ยน Routing Information และ Label ระหว่างอุปกรณ์ที่อยู่ข้างเคียง Control Plane สร้าง Routing Table จาก Routing Protocol หลาย ๆ Routing Protocol เช่น Open Shortest Path First (OSPF) Interior Gateway Routing Protocol (IGRP) Enhanced Interior Gateway Routing Protocol (EIGRP) Intermediate System-to-Intermediate System (IS-IS) Routing Information Protocol (RIP) และ Border Gateway Protocol (BGP) สามารถใช้ใน Control Plane เพื่อควบคุมจัดการกับ Layer 3 Routing นอกจากนี้ Control Plane ยังใช้ Label Exchange Protocol ในการกำหนดและแลกเปลี่ยน Label ระหว่างอุปกรณ์ที่อยู่ข้างเคียง โดย Label Exchange Protocol กำหนดค่า Label ให้กับ Network โดยการเรียนรู้ผ่าน

Routing Protocol ซึ่ง Label Exchange Protocol ประกอบด้วย MPLS Label Distribution Protocol (LDP) และ BGP (ใช้สำหรับ MPLS VPN) ส่วน Resource Reservation Protocol (RSVP) ใช้สำหรับ MPLS TE ในการแลกเปลี่ยน Label

Control plane ประกอบด้วย Forwarding Table 2 ชนิดคือ

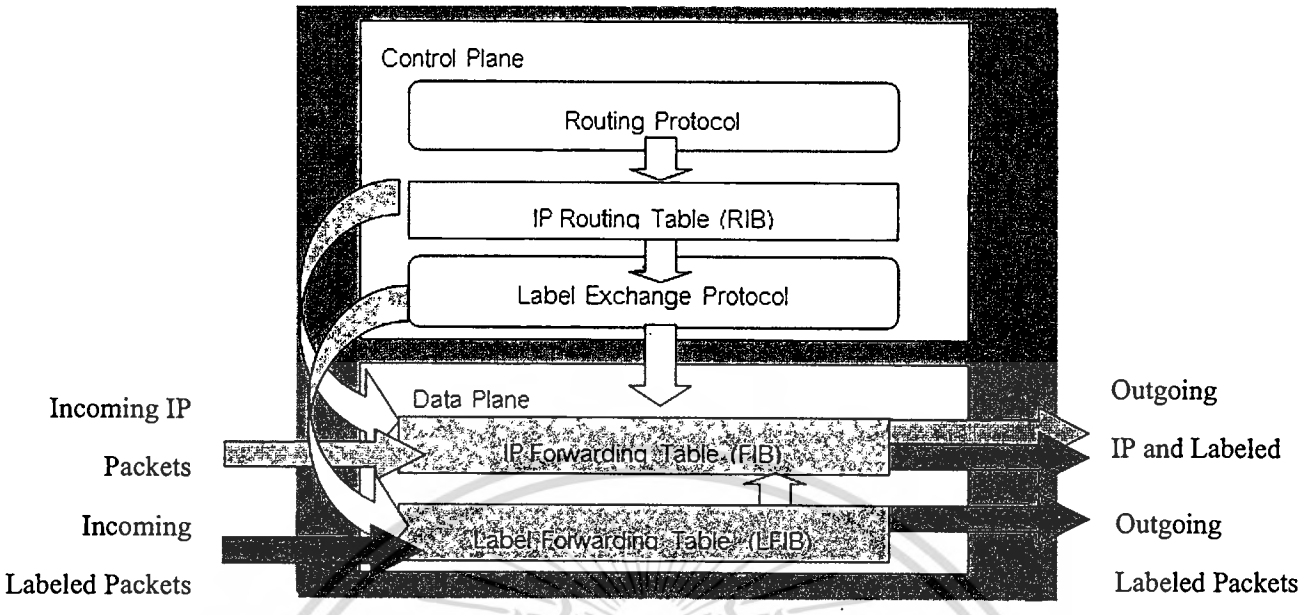
a) FIB ข้อมูลในการสร้าง Table ได้มาจาก RIB

b) Label Forwarding Information base (LFIB) ข้อมูลในการสร้าง Table มาจาก Label Exchange Protocol และ RIB ซึ่งใน LFIB Table ประกอบด้วยค่า Label ที่สอดคล้องกับ Outgoing Interface สำหรับทุกๆ Network Prefix



รูปที่ 3.2 สถาปัตยกรรม MPLS Control Plane

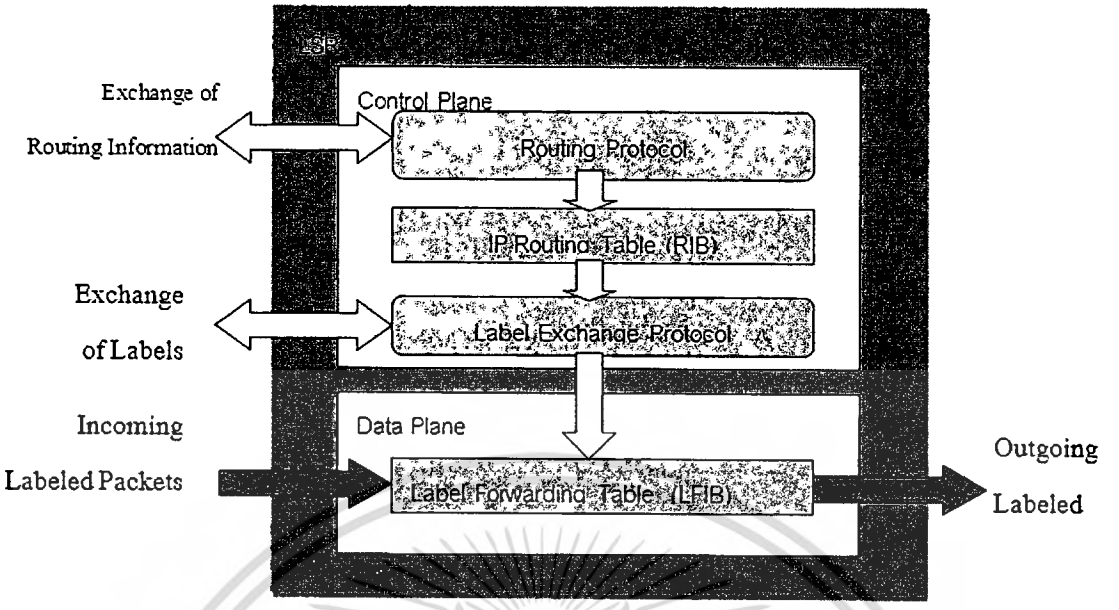
- **Data Plane** - ทำหน้าที่ในการส่ง Packet ข้อมูลโดยสามารถส่ง Packet ด้วย Layer 3 IP Packets หรือ Label IP Packet ข้อมูลที่อยู่ใน Data Plane เช่น Label Values ได้มาจาก Control Plane ข้อมูลที่ได้มาจากการแลกเปลี่ยนกันระหว่าง Router ข้างเคียงซึ่งนำมา Mapping กับข้อมูลของ IP Destination Prefixes และ Label ใน Control Plane ซึ่งนำมาใช้ในการ Forward Data Plane Label Packets



รูปที่ 3.3 สถาปัตยกรรม MPLS Data Plane

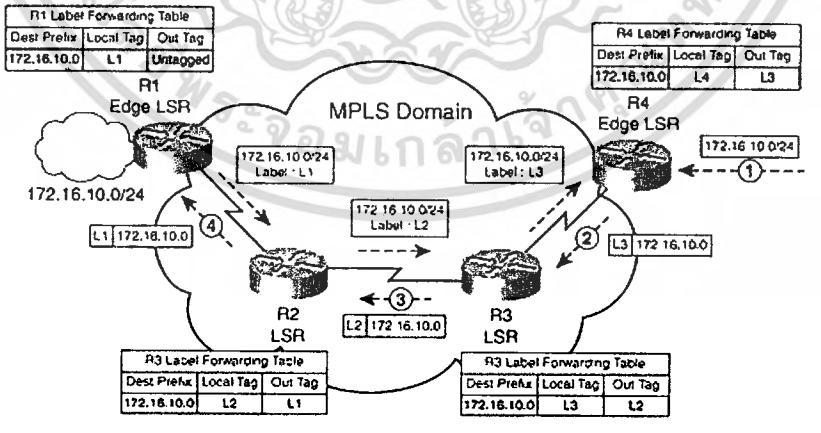
3.1.3 นิยามศัพท์เฉาะของ MPLS

- MPLS Label Switch Router (LSR) ตาม Function ของ Label Switching เมื่อ LSP ได้รับ Labels Packet และมีการแลกเปลี่ยนค่า Labels ด้วย Outgoing Label จาก Labels เดิมและส่งต่อไปด้วย Labels Packet ใหม่ที่ถูกต้องตาม Interface โดยการทำงานของ LSR ขึ้นอยู่กับตำแหน่งที่ติดตั้งใน MPLS Domain โดยสามารถนำ Labels ออก (POP Labels) หรือใส่ Labels เข้ามา(PUSH Labels) หรือแลกเปลี่ยน Label (Swap Label แทนที่ส่วนบนของ Label ใน Label Stack ด้วย Label ใหม่)
- MPLS Edge-Label Switch Router (E-LSR) Edge LSR คือ LSR ที่อยู่ในตำแหน่งขอบของ MPLS Domain Edge LSR ทำหน้าที่ใส่ Labels เข้ามา (PUSH Labels) และ Forwarding Packet ข้อมูลไปยังปลายทางผ่านเครือข่าย MPLS Domain และนำ Labels ออก(POP Labels) Forwarding IP Packet ไปยังเครือข่าย IP Domain ปลายทาง



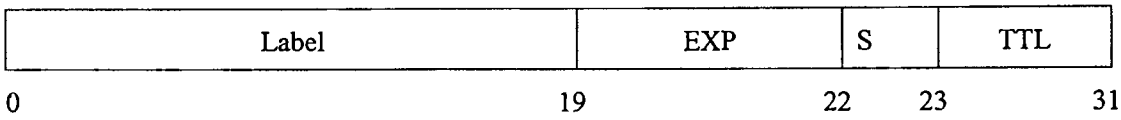
รูปที่ 3.4 สถาปัตยกรรมของ LSRs

- **MPLS Label Switched Path (LSP)** LSP คือเส้นทางในการส่งข้อมูลจากต้นทางไปยังปลายทางผ่านโครงข่าย MPLS ซึ่งโดยปกติโครงข่าย LSP จะเป็นลักษณะทิศทางเดียว (Unidirectional) LSP พัฒนามาจากข้อมูล IGP Routing แต่ก็มีความแตกต่างจาก IGP ตรงที่เป็นการอ้างถึงเส้นทางที่ไปยังปลายทาง ในภาพประกอบรูปที่ 3.5 แสดงเส้นทาง LSP สำหรับเครือข่าย 172.16.10.0/24 จาก R4 คือ R4-R3-R2-R1



รูปที่ 3.5 LSR และ Edge LSR

- **MPLS Labels and Label Stacks** MPLS Labels มีขนาด 20 bit และถูกกำหนดเป็น Prefix แบบปลายทางด้วยการกำหนดบน Router กำหนดคุณสมบัติเพื่อให้มีการนำส่งข้อมูลสำหรับส่งไปยังปลายทางได้ด้วยรูปแบบ MPLS Label ที่แสดงในรูปที่ 3.6



รูปที่ 3.6 MPLS Label

MPLS Label ประกอบไปด้วย

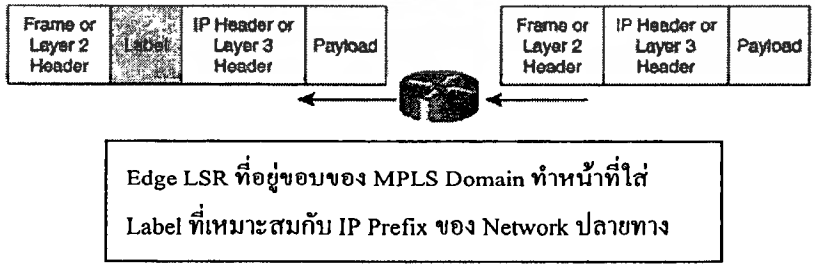
- 20 bit เป็นค่า Label
- 3 bit เป็นค่า Experimental Filed
- 1 bit เป็นตัวแสดงว่าเป็น Bottom-of-Stack
- 8 bit เป็นส่วนตรวจสอบ Time-to-Live Field

20 bit แรกเป็นค่าที่กำหนดโดย Router ด้วยการ Identifies Prefix โดย Labels สามารถกำหนดด้วย Interface หรือกำหนดด้วย Chassis ส่วน Experimental bit จะกำหนด QOS ที่ได้รับจาก IP Packet ซึ่งกำหนดไว้แล้วด้วย Label โดย Experimental bit สามารถ Map กับค่า IP Precedence เพื่อกำหนดค่า IP QoS ของ Packet ที่อยู่ใน MPLS Domain

Label Stack คือ Label ที่มีคุณสมบัติพิเศษที่บอกถึงลักษณะของ Label ถ้า Router (Edge LSR) มีการใส่ Label มากกว่า 1 Label บน IP Packet เดียว ซึ่งเรียกว่า Label Stack ดังนั้นการจะรู้ได้ว่ามีการใส่ Label มากกว่า 1 Label ใน IP Packet เดียวโดยดูจาก Bottom-of-Stack Indicator ซึ่ง Label ตัวสุดท้าย (Bottom label) ค่าของ Bottom-of-Stack Indicator จะถูกตั้งค่าเป็น 1

TTL จะเหมือนค่า Function TTL ของ IP ซึ่ง Packet จะถูกละทิ้งเมื่อค่า TTL เป็น 0 เพื่อเป็นการป้องกันการเกิด Loop เมื่อใดก็ตามที่ Packet ถูกส่งไปตามเส้นทางของ LSR ค่า Label TTL จะลดลงทีละ 1

Label จะถูกใส่เพิ่มเข้าไประหว่าง Frame Header และ Layer 3 Header รูปที่ 7 แสดงให้เห็นการเพิ่ม Label เข้าไปใน Layer 2 และ Layer 3 Header ใน IP Packet

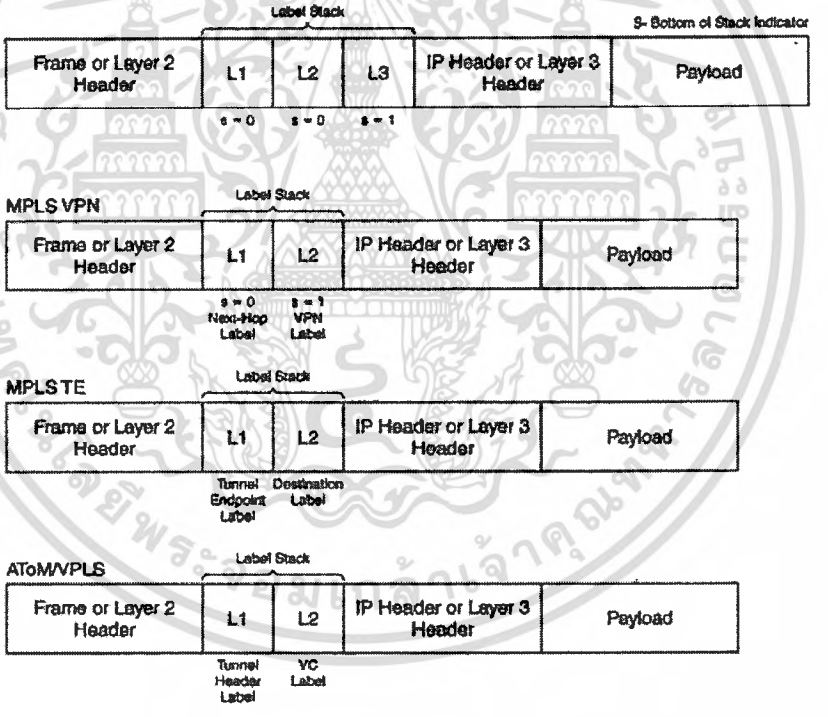


รูปที่ 3.7 MPLS Label Imposition

ถ้าค่าของ S bit (ค่า Label ตัวสุดท้าย) ใน Label มีค่าเป็น 0 Router จะเข้าใจว่ามีการใช้งาน Label Stack โดย LSR จะเปลี่ยนเฉพาะ Label บนสุด ใน Label Stack อย่างไรก็ตาม

Edge LSR จะทำการถอด Label ต่อไปจนกระทั่งพบว่าค่า S bit มีค่าเป็น 1 ซึ่งเป็นเครื่องหมายแสดงว่าเป็น Label สุดท้าย หลังจาก Router ได้รับ Stack ค่าสุดท้าย Router จะพบกับค่า IP Layer 3 Header และปลายทางที่เหมาะสมที่สุดในการส่ง Packet ในกรณี ของ Ingress Edge LSR ซึ่งอาจจะมีการใส่ Label เข้าไปมากกว่า 1 Label Stack ตรง Stack Function

Label Stack เป็นเครื่องมือเพื่อให้บริการ MPLS-base เช่น MPLS VPN หรือ MPLS Traffic Engineering ใน MPLS VPN Label ที่สองใน Label Stack จะกำหนดค่า VPN ส่วน MPLS Traffic Engineering Label บนสุด (Top Label) จะกำหนดจุดสิ้นสุดของ TE Tunnel และ Label ที่สอง(Bottom Label) จะกำหนดปลายทาง สำหรับการทำ MPLS Layer 2 VPN เช่น AToM และ VPLS Label บนสุด(Top Label) เป็น Tunnel Header หรือ Endpoint, และ Label ที่สอง(Bottom Label) จะกำหนด VC ที่กล่าวมาทั้งหมดแสดงในรูปที่ 3.8



รูปที่ 3.8 MPLS Label stack

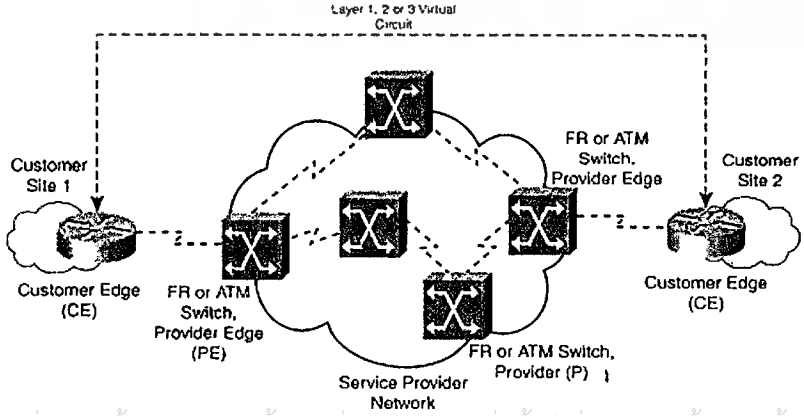
3.2 Virtual Private Network (VPN)

Virtual Private Network (VPN) คือการให้บริการการเชื่อมต่อระหว่างสำนักงานที่อยู่ห่างไกลกันในลักษณะ Point-To-Point หรือ Multipoint โดยใช้ Infrastructure เดียวกันซึ่งเชื่อมต่อกันโดยผ่าน Service Providers เทคโนโลยีแบบดังเดิมที่ใช้ในการให้บริการ VPN คือ Frame Relay และ ATM ซึ่งการเชื่อมต่อระหว่างสำนักงานจะใช้ Dedicate Link ในลักษณะ Point-To-Point แต่หากมีการเชื่อมต่อเป็น Full Mesh ก็จะใช้ในลักษณะ Hub-And-Spokes ส่วนประกอบของการให้บริการ VPN ประกอบด้วย

- Customer Network ประกอบด้วย Router ที่อยู่ที่สำนักงานของผู้ใช้บริการ(Customer site) โดย Router ของลูกค้าจะเชื่อมโยงเข้ากับ Network ของ Service Provider ซึ่งจะเรียก Router นี้ว่า Customer Edge (CE) Router
- Provider Network โครงข่ายของผู้ให้บริการที่เชื่อมโยงกับสำนักงานของผู้ใช้บริการในลักษณะ Dedicate Point-To-Point โดยอุปกรณ์ของ Service Provider ที่เชื่อมโยงเข้ากับ CE Router เรียกว่า Provider Edge (PE) Router และอุปกรณ์ภายในโครงข่ายของ Service Provider ที่ทำหน้าที่ส่งผ่านข้อมูล(Forwarding Data)ภายใน Service Provider Backbone เรียกว่า Provider (P) Router

การ Implement VPN สามารแบ่งได้สองลักษณะคือ

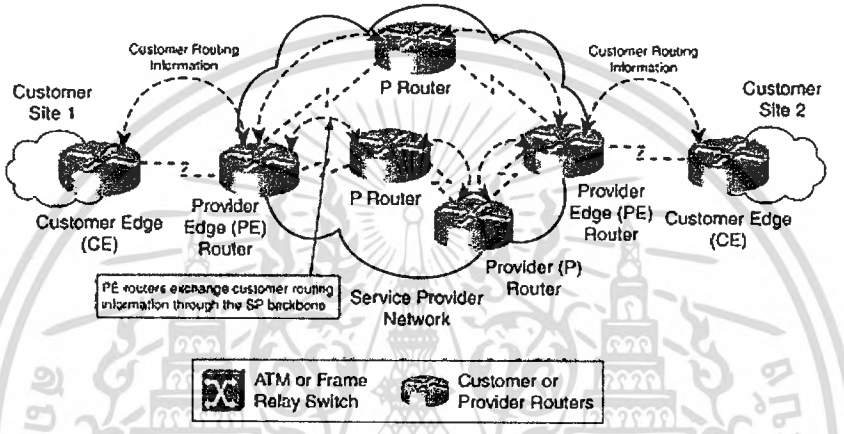
- Overlay Model การเชื่อมต่อระหว่าง Service Provider และผู้ใช้บริการเป็นลักษณะ Virtual Point-To-Point Links ซึ่งการให้บริการ VPN ในลักษณะ Overlay Model ได้แก่
    a) Frame Relay or ATM VPN เชื่อต่อกันแบบ Layer 2 Virtual Circuit Point-To-Point
    b) IP VPN เชื่อต่อกันแบบ Layer 3 โดยการทำ Tunneling Protocol เช่น GRE และIPSec Routing Protocol ของผู้ใช้บริการจะแลกเปลี่ยนกันระหว่างสาขาที่ต่อเชื่อมกันโดยแยกออกจาก Service Provider



รูปที่ 3.9 Overlay Model

ข้อเสียของการ Implement VPN แบบ Overlay Model คือหากต้องทำการเชื่อมโยงสาขาของผู้ใช้บริการโดยต่อ Virtual Circuit ในลักษณะ Full Mesh ซึ่งหากมีจำนวนสาขา N ต้องใช้จำนวน Virtual Circuit จำนวน N(N-1)/2 Circuit

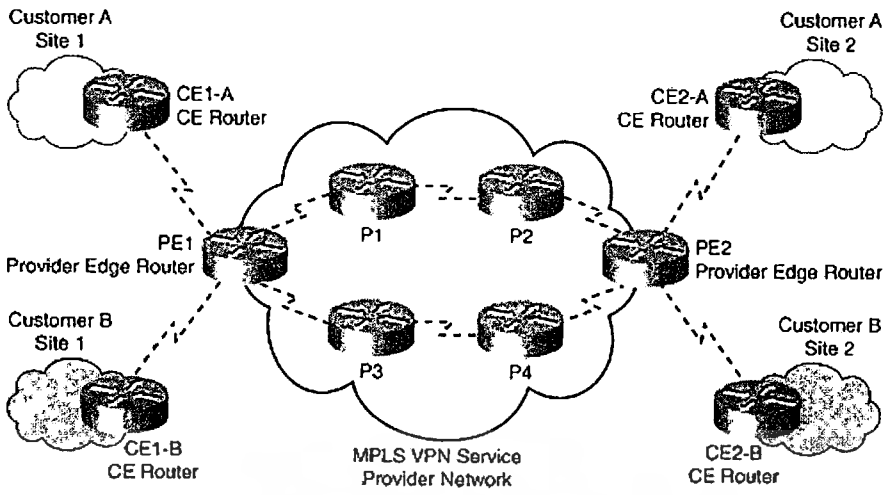
- Peer-To-Peer Model CE Router ของผู้ใช้บริการจะแลกเปลี่ยน Routing Information ระหว่าง Router ในโครงข่ายของ Service Provider (P และ PE Router) เพื่อทำการหาเส้นทางที่ดีที่สุดในการส่งข้อมูลผ่านโครงข่ายของ Service Provider โดยไม่ต้องทำการสร้าง Virtual Circuit หรือใช้ Tunneling Protocol



รูปที่ 3.10 Peer-To-Peer Model

3.2.1 สถาปัตยกรรม MPLS VPN

ในโครงข่าย MPLS VPN Edge Router จะทำการส่งข้อมูล Routing Information เครือข่ายของผู้ใช้บริการโดยทำการหาเส้นทางที่ดีที่สุด (Optimal Routing)ในการส่งข้อมูลไปยังสาขาปลายทางของผู้ใช้บริการโดย MPLS VPN ผู้ใช้บริการแต่ละรายสามารถใช้ IP Address ที่ซ้ำกันได้ (Overlapping Address) ซึ่งจะไม่เหมือนกับ VPN แบบดังเดิมที่เป็นลักษณะ Peer-To-Peer Model ซึ่งการสร้างเส้นทางที่ดีที่ที่สุด (Optimal Routing) ให้แก่ผู้ใช้บริการจำเป็นจะต้องทำการกำหนด IP Address ให้ผู้ใช้บริการแต่ละรายไม่ซ้ำกันเพื่อหลีกเลี่ยงการเกิด Overlapping Address MPLS VPN ประกอบด้วย Customer Network และ Provider Network โดยข้อมูลของผู้ใช้บริการแต่ละรายจะแยกเป็นอิสระจากกันถึงแม้จะต่ออยู่กับ PE ตัวเดียวกันโครงสร้างของ MPLS VPN แสดงดังรูปที่ 3.11



รูปที่ 3.11 สถาปัตยกรรม MPLS VPN Network

สถาปัตยกรรมของ MPLS VPN ประกอบด้วย

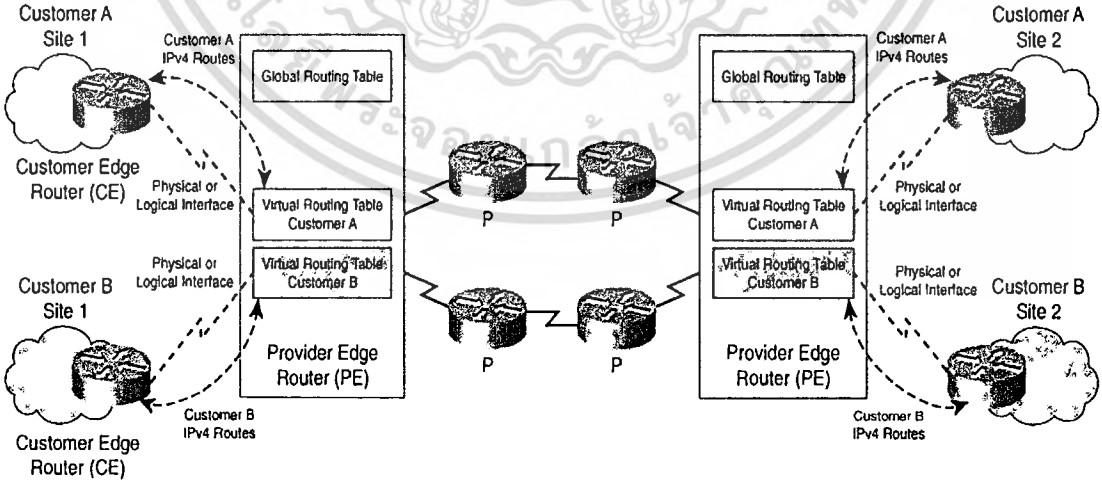
- Customer Network : โครงข่ายภายในของผู้ใช้บริการซึ่งอาจประกอบไปด้วยอุปกรณ์อย่างเช่น Router ในรูปที่ 3.11 Customer Network สำหรับผู้ใช้บริการ A ประกอบด้วย Router CE1-A , CE2-A และ อุปกรณ์เครือข่ายภายในของผู้ใช้บริการที่ต่ออยู่กับ CE1-A และ CE2-A
- CE Router : Router ที่เชื่อมโยงกับโครงข่ายของผู้ให้บริการ ในรูปที่ 3.11 CE Router ของผู้ใช้บริการ A คือ CE1-A และ CE2-A และ CE Router ของผู้ใช้บริการ B คือ CE1-B และ CE2-B
- Provider Network : โครงข่ายภายในของผู้ให้บริการ ซึ่งประกอบด้วย Provider Core และ Provider Edge Router เชื่อมต่อเข้ากับโครงข่ายของผู้ใช้บริการ โดย Provider Network จะทำการควบคุม Routing ระหว่างสาขาของผู้ใช้บริการ ในรูปที่ 3.11 Provider Network ประกอบด้วย Router PE1, PE2, P1, P2, P3 และ P4
- PE Routers : ในโครงข่ายของผู้ให้บริการ (Provider Network) Router ที่เชื่อมต่ออยู่กับ Customer Edge (CE) Router จะเรียกว่า PE Router ในรูปที่ 3.11 PE1 และ PE2 คือ Provider Edge Router ใน MPLS VPN ของ Customer A และ B
- P Router : Core Router ในโครงข่ายของผู้ให้บริการ (Provider Network) ซึ่งเชื่อมต่อระหว่าง Core Router ด้วยกัน หรือ Provider Edge Router โดย Provider Router จะทำหน้าที่ส่งถ่ายข้อมูลข้ามผ่าน Provider Network โดยไม่เกี่ยวข้องกับ Customer Router ในรูปที่ 3.11 Router P1, P2, P3 และ P4 คือ Provider Router

3.2.2 MPLS VPN Routing Model

MPLS VPN มีลักษณะที่เหมือนกับ VPN แบบ Dedicated Router Peer – To – Peer ซึ่ง CE Router จะทำการส่ง Routing Table และข้อมูลไปยัง PE Router โดยที่ CE Router ไม่ต้องทำการ Configuration อะไรเป็นพิเศษเพื่อให้สามารถใช้งานใน MPLS Domain นอกจากการกำหนดให้ CE Router ใช้งาน Routing Protocol ซึ่งอาจจะเป็น Static Route หรือ Default Route เพื่อให้เกิดการแลกเปลี่ยน Routing Information ระหว่าง CE และ PE Router

ในการทำงานของ MPLS VPN PE Router ต้องมีความสามารถในการแยกแยะ Traffic ข้อมูลของลูกค้าแต่ละรายที่ต่ออยู่บน PE Router เดียวกัน ซึ่ง Routing Table ของลูกค้าแต่ละรายต้องเป็นอิสระต่อกันและไม่ปะปน เสมือนว่าไม่ได้ต่ออยู่บน PE เดียวกัน ซึ่งเมื่อ Traffic ผ่านเข้าไปยังโครงข่ายของผู้ให้บริการก็จะใช้ Routing Process ใน Global Routing Table ของ Router ภายในโครงข่าย P Router จะแลกเปลี่ยน Lable ระหว่าง PE Router และ P Router ด้วยกัน โดยไม่รู้ถึง VPN Routes ซึ่ง CE Router ก็ไม่ทราบและไม่เห็นว่ามี P Router อยู่ในโครงข่าย โดยโครงข่ายภายในของ SP Network จะเป็น Transparent เพื่อส่งข้อมูลระหว่างสาขาของลูกค้าโดยที่ลูกค้าจะไม่ทราบถึง Topology ภายในของ SP ว่าเป็นอย่างไร

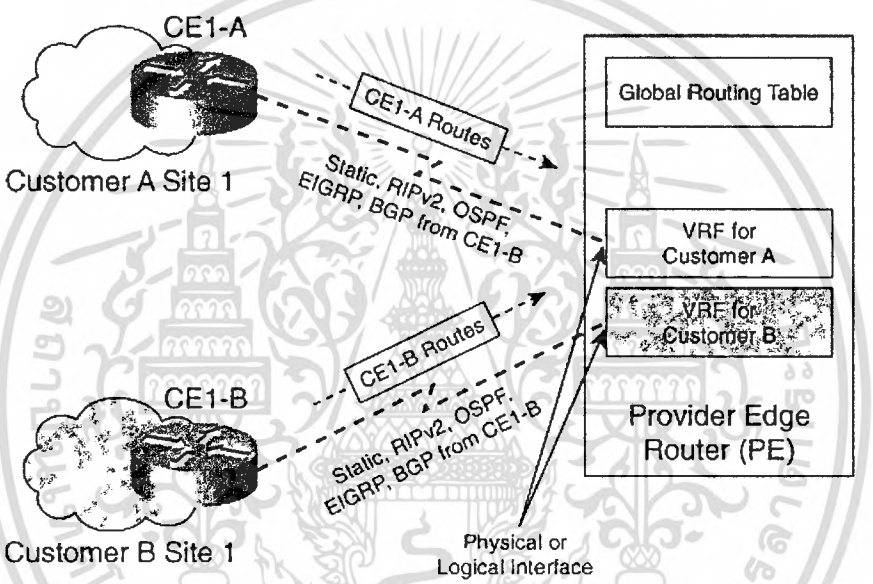
P Router จะทำเพียงการแลกเปลี่ยน Lable โดยที่ไม่ทราบถึง VPN Routes ส่วน PE Router จะแลกเปลี่ยน Routing Table ระหว่าง CE ที่ต่ออยู่โดยใช้ Routing Protocol IGRP หรือ Static Route ในการที่โครงข่ายจะสามารถรองรับจำนวน VPN ของลูกค้าได้เป็นจำนวนมากต้องใช้ Multiprotocol BGP ในการส่ง Routes ของลูกค้า โดย Multiprotocol BGP จะถูกกำหนดค่าระหว่าง PE Router ที่จะต้องส่งค่า Routes ไปยัง CE Router ของลูกค้า



รูปที่ 3.12 สถาปัตยกรรม MPLS VPN

3.2.3 VRF : Virtual Routing และ Forwarding Table

ผู้ใช้บริการแต่ละรายที่ต่ออยู่กับ PE Router เดียวกันสามารถแยก Routing Tables ของตนเองไม่ให้ปะปนกับผู้ใช้บริการรายอื่น ๆ โดยใช้ Virtual Routing Table Instances หรือที่เรียกว่า Virtual Routing and Forwarding Table/Instance (VRFs) การทำงานของ VRF จะเหมือนการทำงานของ Global Routing ยกเว้นข้อมูลเส้นทางใน VRF เป็นข้อมูลของเส้นทางในการเชื่อมต่อ VPN ของผู้ใช้บริการโดยข้อมูลเส้นทางใน VRF ได้มาจาก Routing Protocol ที่ใช้ในแต่ละ VPN โดยส่งผ่านทาง Interface ที่เชื่อมระหว่าง CE และ PE Router ดังนั้นหนึ่ง Interface (Logical or Physical) สามารถต่อร่วมได้เพียงหนึ่ง VRF เท่านั้น รูปที่ 3.13 แสดงการทำงานของ VRF บน PE Router ในการแยกแยะ Routing Table ของผู้ใช้บริการแต่ละราย



รูปที่ 3.13 VRF บน PE Router

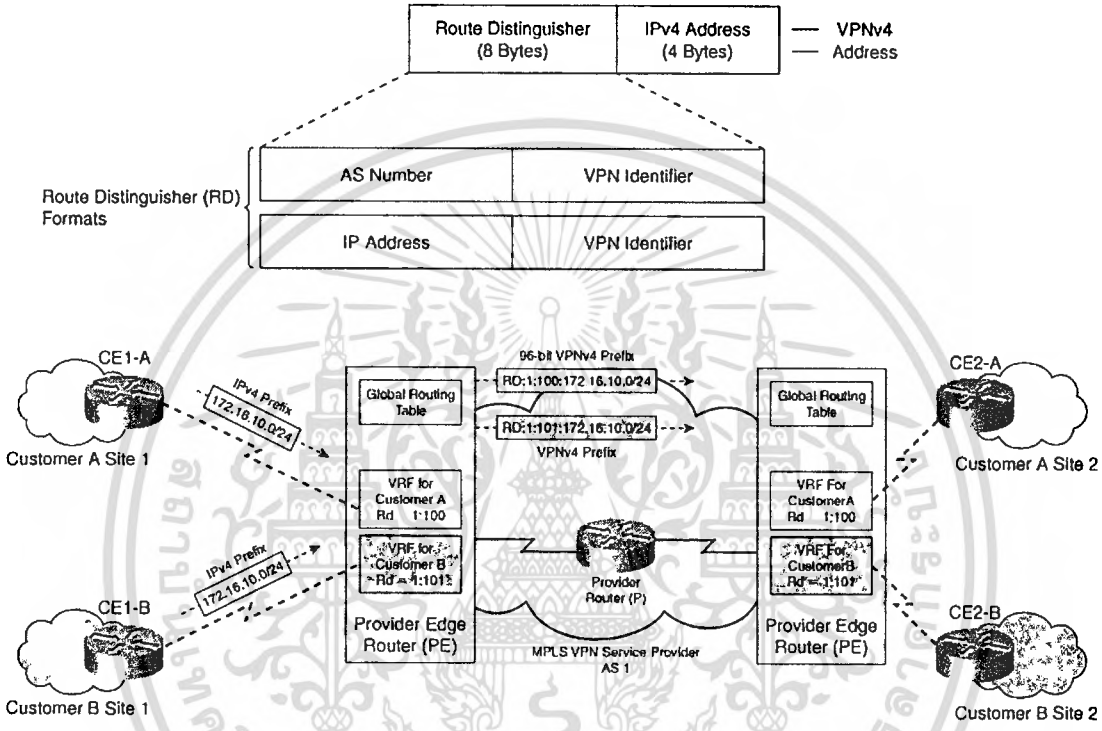
3.2.4 Route Distinguisher

ในการทำงานของ MPLS VPN PE Router จะทำหน้าที่ในการแยกแยะข้อมูลของผู้ใช้บริการแต่ละรายโดยใช้ VRF, ซึ่งข้อมูลระหว่างสาขาของผู้ใช้บริการจะส่งผ่านไปยัง MPLS VPN Backbone PE Router จำเป็นต้องมีความสามารถในการทำ Overlapping Address ในการเชื่อมต่อกับ Customer Network โดย PE Router ต้องทำการเรียนรู้เส้นทางของ Customer Network และทำการประกาศข้อมูลเหล่านั้น โดยผ่าน Provider Backbone ซึ่งขั้นตอนเหล่านี้จะสามารถทำงานได้โดยอาศัยการทำงานร่วมกันระหว่าง Route Distinguisher (RD) และ Virtual Routing Table บน PE Router

RD มีขนาด 64 bit ใช้ร่วมกับ IP Address (IPv4) เพื่อกำหนดเป็น VPN Address ให้ผู้ใช้บริการโดยไม่ต้องกังวลเกี่ยวกับ IP Address ที่จะซ้ำกันของลูกค้าแต่ละรายเพราะลูกค้าแต่ละ

รายจะมี RD ที่ไม่ซ้ำกัน ซึ่งค่า RD เมื่อร่วมกัน IP Address (IPv4 – 32bit) ก็จะมีขนาด 96 bit (32 bit customer prefix + 64 bit RD) โดยเรียกว่า VPN Version 4 (VPN v4) Address

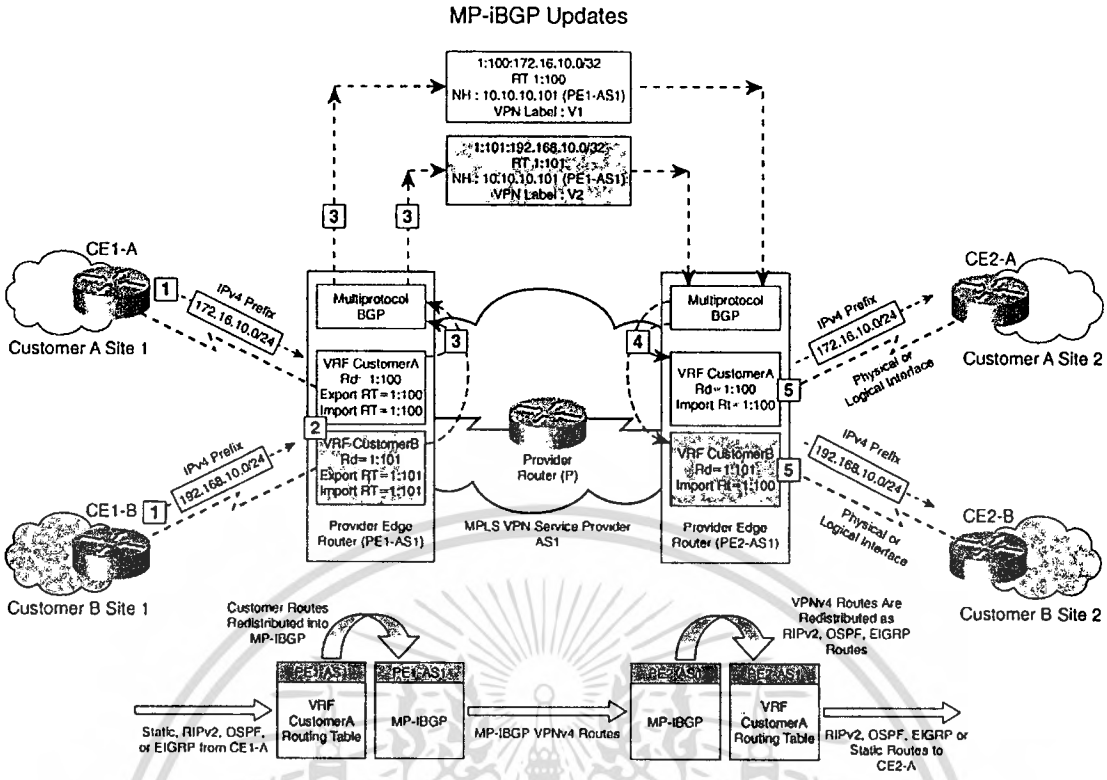
Protocol ที่ใช้ในการแลกเปลี่ยน VPNv4 Routes ระหว่าง PE Router คือ Multiprotocol BGP (MP-BGP) เนื่องจาก BGP มีความสามารถที่จะรองรับ Routing Table ได้เป็นจำนวนมาก รูปที่ 3.14 แสดงให้เห็นเมื่อลูกค้ามี IP Prefix 172.16.10.0/24 ที่ซ้ำกัน แต่จะมี RD ที่แตกต่างกัน คือ 1 : 100 และ 1 : 101



รูปที่ 3.14 RD ใน MPLS VPN

3.2.5 Route Targets (RT)

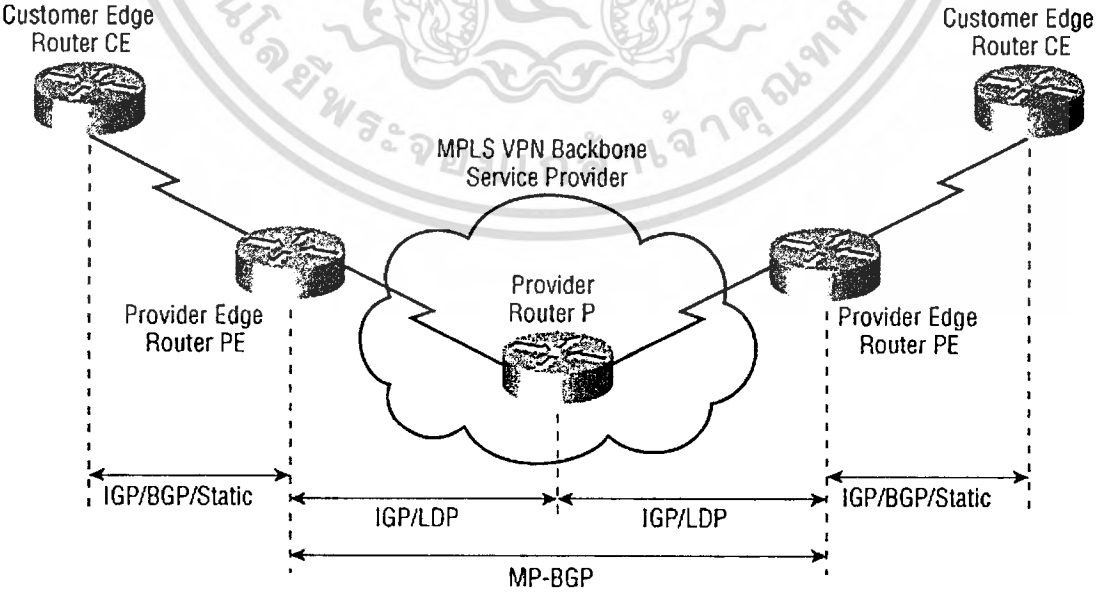
ในการทำ MPLS VPN หากมีความจำเป็นที่ต้องการให้ VPN ที่ต่างกันใน 1 site นั้น สามารถเชื่อมต่อได้มากกว่า 1 VPN ทำได้โดยอาศัย Route Targets (RT) ซึ่ง RT จะถูกระบุไว้กับ VRF คือ หาก VRF ใดมีค่า RT ที่เหมือนกันก็สามารถเชื่อมต่อกันได้ ตัวอย่างลักษณะการใช้งาน VPN เช่นนี้ได้แก่ Extranet VPN, Network Management VPN , Internet Access VPN



รูปที่ 3.15 RT และ RD ใน MPLS VPN

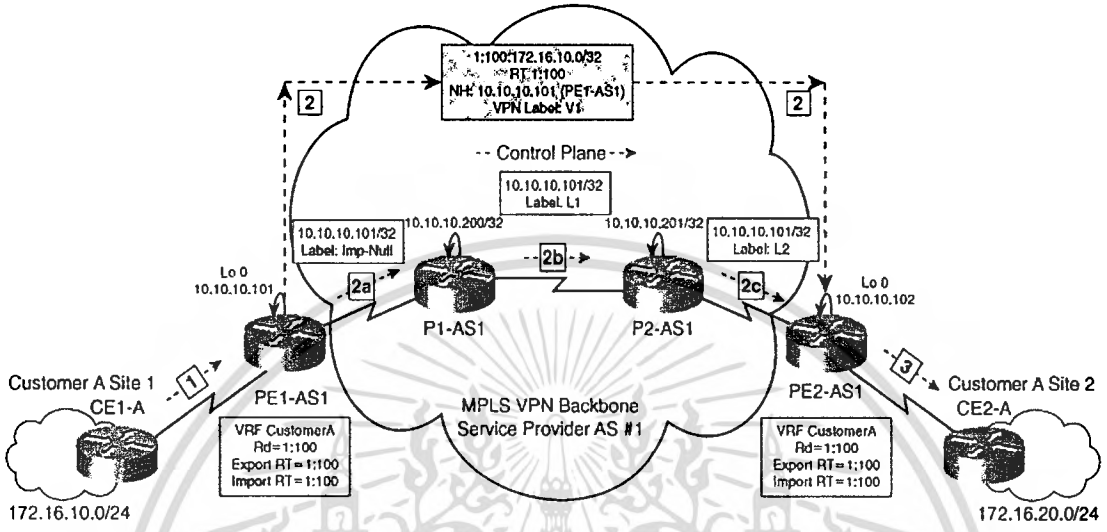
3.2.6 การทำงานของ MPLS VPN Control Plane

Control Plane ใน MPLS VPN ทำงานโดยใช้ Layer 3 Routing Information เพื่อกำหนด IP Prefix ให้กับ Label และทำการกระจาย Label โดยใช้ LDP รูปที่ 3.16 แสดง Protocol ที่ใช้ใน MPLS VPN Control Plane.



รูปที่ 3.16 Control Plane ใน MPLS VPN

CE Router เชื่อมต่อกับ PE Router โดยใช้ IGP, BGP หรือ Static Route MPLS VPN Backbone ประกอบด้วย P และ PE Router โดยใช้ OSPE หรือ ISIS ส่วน LDP ใช้ในการกำหนด Label และทำการกระจาย label ภายใน MPLS Domain การทำงานของ MPLS VPN Control Plane แสดงดังรูปที่ 3.17 มีขั้นตอนดังนี้



รูปที่ 3.17 การทำงานของ Control Plane

Step 1 CE1-A ทำการ Update Network 172.16.10.0 ไปให้ PE Router

Step 2 PE1–AS1 ทำการเก็บข้อมูล Network 172.16.10.0/24 ลงใน VPNv4 Route โดยกำหนดค่า RD และ RT ใน VRF เป็น 1:100 และกำหนดให้ VPNv4 Label มีค่าเป็น v1 สำหรับ Network 172.16.10.0/24 โดยทำการ Update ผ่าน Loopback 0 IP Address 10.10.10.101 ซึ่งจะถูก Map กับ Label โดยใช้ LDP ซึ่งมีขั้นตอนการกำหนด label ดังนี้

(a) 2a : จากรูปที่ 3.17 PE2–AS1 จะส่งคำขอ Label สำหรับ 10.10.10.101/32 ไปยัง P2–AS1 ซึ่ง P2–AS1 ก็ทำการส่งคำขอนี้ต่อไปยัง PE1–AS1 ซึ่ง PE1–AS1 กำหนดให้เป็น Implicit – Null (Label จะถูกถอดออกก่อนมายัง PE1–AS1)

(b) 2b : P1–AS1 ได้รับ Implicit – Null จาก PE1–AS1 ดังนั้นจึงกำหนด Outbound Label สำหรับ 10.10.10.101/32 เป็น L1 และทำการบันทึกค่าลงใน LFIP หลังจากนั้นก็ทำการประกาศค่า Label ให้ P2-AS1 รับทราบ

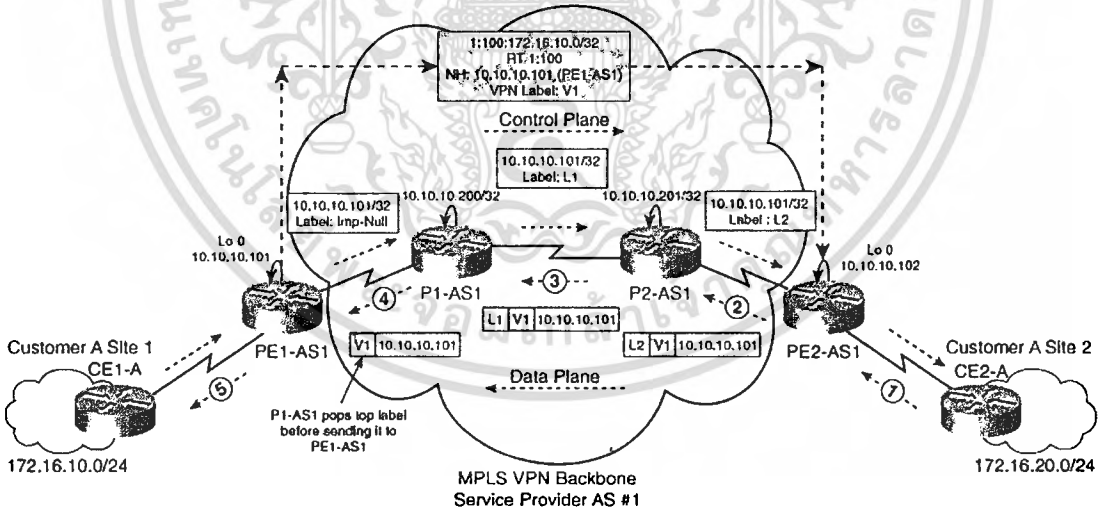
(c) 2c : เมื่อ P2 – AS1 ได้รับทราบว่า P1-AS1 ใช้ Label L1 สำหรับ 10.10.10.101/32 จึงกำหนดให้ Outbound Label เป็น L2 และส่งค่า Label นี้ไปยัง PE2-AS1 และบันทึกค่าลงใน LFIP

Step 3 PE1 – AS1 ตรวจสอบค่า RT ใน VRF ซึ่งมีค่าตรงกันคือ 1 : 100 และทำการแปลง VPNv4 เป็นข้อมูล IPv4 พร้อมกับปรับปรุงค่า Routing โดยเพิ่ม Network 172.12.10.0/24 ใน VRF และส่ง Routing เส้นทางนี้ Updateไปยัง CE2 – A

3.2.7 การทำงานของ MPLS VPN Data Plane

การทำงานของ MPLS VPN Data Plane จะเกี่ยวข้องกับการใช้งาน Label Stack ซึ่ง Label ที่อยู่บนสุดของ Label Stack จะถูกกำหนดเพื่อเป็นตัวบ่งชี้ถึง Egress PE Router ที่ต้องการส่งข้อมูลไป และ Label ลำดับที่สองใน Label Stack เป็น VPN Label ที่ถูกกำหนดโดย Egress PE Router ที่ต่ออยู่กับ CE Router เมื่อมีการใช้งาน MPLS VPN Label ที่กำหนดจะถูกซ้อนกันอยู่ใน Label Stack โดยเมื่อ Label ถูกส่งต่อเข้าไปยังโครงข่าย MPLS P Router จะดู Label บนสุดของ Label Stack ซึ่งค่าของ Label นี้จะสัมพันธ์กับหมายเลข IP Address ของ Egress PE Router ปลายทาง แล้วทำการสลับ Label บนสุดของ Label Stack เมื่อส่งไปยังเส้นทางที่เหมาะสมไปยัง Router ที่ต่ออยู่ถัดไป (Next – Hop) ซึ่งเมื่อถึง Egress PE Router ปลายทาง ค่า Label บนสุดใน Labels Stack จะถูกถอดออกและจะใช้ Label ลำดับที่สองซึ่งจะสัมพันธ์กับ VRF ของ VPN นั้นๆ ซึ่งเมื่อรู้ค่า VPN Label นั้นสัมพันธ์กับ VRF ใดก็จะทำการส่งข้อมูล IP Packet ไปยัง CE Router ที่สัมพันธ์กับ VRF นั้น

การทำงานของ MPLS VPN Data Plane แสดงดังรูปที่ 3.18 มีขั้นตอนดังนี้



รูปที่ 3.18 การทำงานของ Data Plane

Step 1 CE2–A ส่งข้อมูลจาก Source Address 172.16.20.1 ไปยัง Destination Address 172.16.10.1

Step 2 เมื่อ PE2–AS1 รับข้อมูลจาก CE2-A จะทำการกำหนด VPN Label โดยกำหนดให้เป็น V1 (Second Label) และ Label L2 (Top Label) แล้วจึงส่งต่อข้อมูลไปยัง P2–AS1

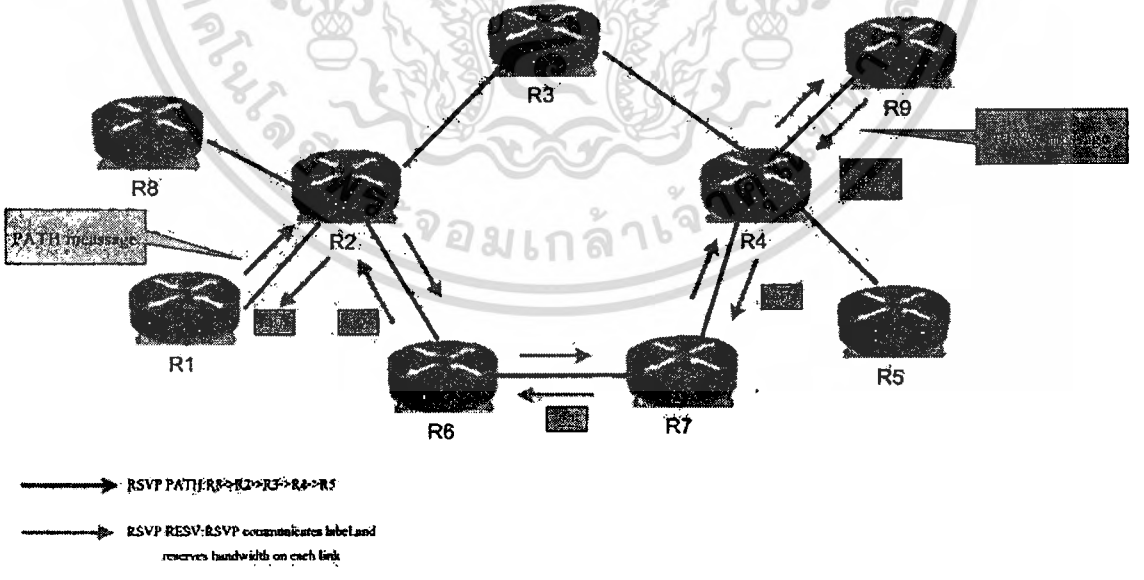
Step 3 เมื่อ P2-AS1 ได้รับข้อมูลที่มาจาก PE2-AS1 ซึ่งมีปลายทางเป็น 172.16.10.1 จึงทำการสลับ label จาก L2 เป็น L1 แล้วจึงส่งต่อให้ P1-AS1

Step 4 เมื่อ P1-AS1 ได้รับข้อมูลซึ่งมีปลายทางเป็น 172.16.10.1 ก็จะทำการถอด Label บนสุดออกแต่เนื่องจาก P1-AS1 ได้รับ Implicit – Null Label ซึ่งแสดงว่า Router ตัวถัดไปเป็นตัวสุดท้ายใน MPLS Domain และสอดคล้องกับ IP Address 10.10.10.101/32 ซึ่งมาจาก PE1-AS1 จึงทำการถอด Label L1 ออก ดังนั้นข้อมูลที่ส่งไปยัง PE1-AS1 จึงเป็น Label Packet ที่มีเพียง VPN Label V1

Step 5 PE1–AS1 จะทำการถอด VPN Label V1 ออกและส่งต่อข้อมูลไปยัง CE1-A ที่มี Network Address ปลายทางเป็น 172.16.10.0

3.3 MPLS Traffic Engineering (TE)

MPLS Network ใช้กลไกการทำ Traffic Engineering (TE) ในการทำให้ความคับคั่งของเครือข่ายข้อมูลเกิดขึ้นน้อยที่สุดและทำให้ประสิทธิภาพของโครงข่ายดีขึ้น โดยสามารถปรับปรุงเส้นทางในการส่งข้อมูลเพื่อให้เหมาะสมกับ Network Resource ที่มีอยู่ซึ่งทำให้สามารถลดการเกิดความคับคั่งของข้อมูลในเครือข่ายและทำให้คุณภาพการให้บริการของเครือข่ายดีขึ้นเช่นทำให้ค่า Latency Time, Jitter , Packet Loss ลดลง นอกจากนี้ MPLS TE ยังสามารถช่วยลดผลกระทบอันเกิดมาจากความบกพร่องของ Network Failures และช่วยเพิ่มความสามารถในการให้บริการ (Service Availability)ได้ดีขึ้น

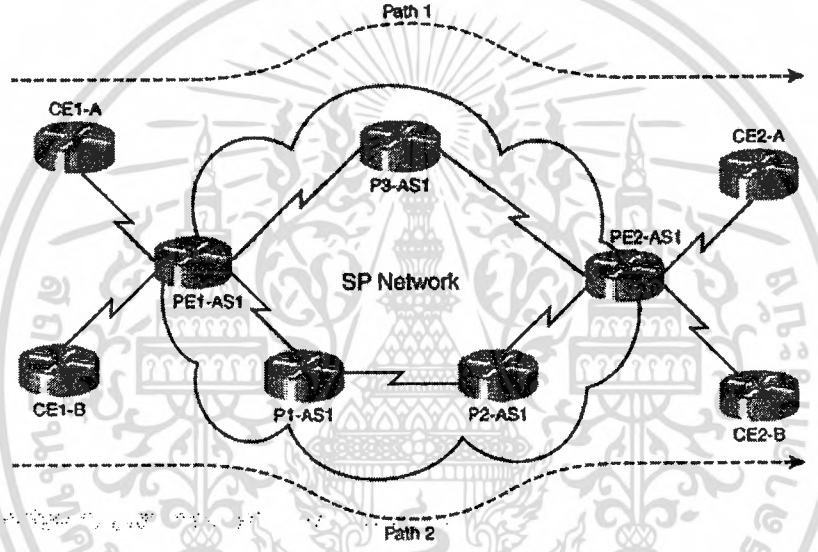


รูปที่ 3.19 การสร้าง TE LSP โดยใช้ RSVP

3.3.1 พื้นฐาน TE

TE เป็นวิธีจัดการจราจรข้อมูลบน Backbone เพื่อให้การสื่อสารระหว่าง Routers สามารถใช้ Bandwidth ได้อย่างมีประสิทธิภาพ ทั้งนี้ ก่อนที่จะใช้ระบบ MPLS TE อย่างในปัจจุบัน วิศวกรจัดการจราจรของข้อมูลด้วย IP หรือ ATM ขึ้นอยู่กับโปรโตคอลระหว่าง Routers ทั้งสองกระทั่งทุกวันนี้ แม้ "Traffic Engineering" จะหันมาใช้ MPLS TE มากขึ้นแล้ว แต่ TE ใน IP Networks ก็ยังทำงานด้วย IP หรือ ATM อยู่

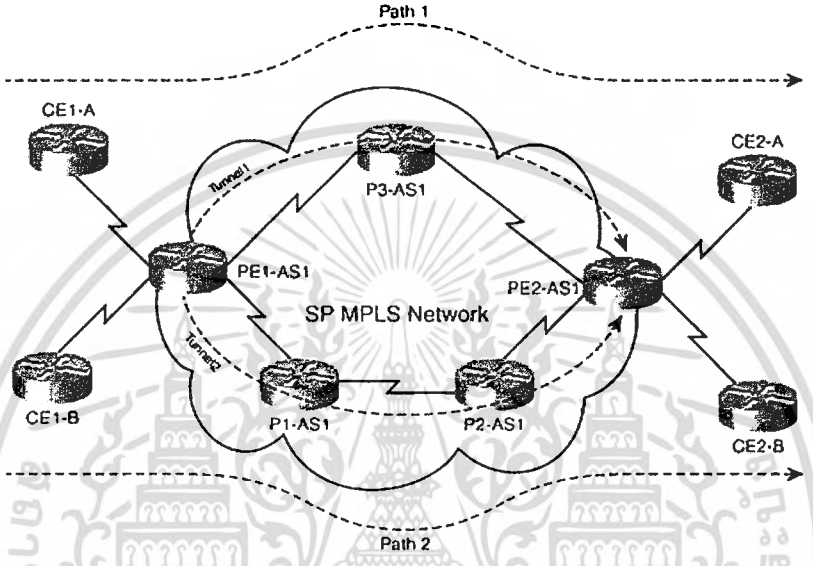
หลักการของTE ที่ใช้ IP คือ จัด Interface cost ระหว่าง 2 จุดใน Network ให้มีค่าต่ำที่สุด โดยระหว่างสองจุดมีเส้นทางเชื่อมต่อเหมือนเดิมเสมอ ดัง IP Network พื้นฐานที่ยกตัวอย่างในรูปที่ 3.20 ซึ่งผู้ใช้ 2 จุด ระหว่าง A และ B เชื่อมต่อกันผ่านผู้ให้บริการเดิม



รูปที่ 3.20 Traffic Engineer ใน Tradition IP Network

จากรูปที่ 3.20 ระหว่าง Routers ผู้ใช้บริการ CE1-A และ CE2-A มีเส้นทางเชื่อมต่อกันผ่านผู้ให้บริการ 2 เส้นทาง ถ้าทุกจุดที่เชื่อมอยู่ระหว่าง Routers มี Cost เท่ากัน เส้นทางเชื่อมต่อระหว่าง Routers CE1-A และ CE2-A ที่มี Cost ต่ำสุดก็คือ PATH1 (ผ่าน Routers PE1-AS1, P3-AS1 และ PE2-AS1) ซึ่งเป็นทางเดียวกับที่ Routers CE1-B และ CE2-B ใช้เหมือนกัน สมมุติให้จุดเชื่อมต่อทั้งหมดเป็น T3 หาก CE1-A ส่งข้อมูลที่ความเร็ว 45 Mbps พร้อมๆ กับ CE1-B ซึ่งส่งข้อมูลที่ความเร็ว 10 Mbps จะทำให้ Packet บางส่วนชะงัก (Drop) อยู่ที่จุด PE1-AS1 เนื่องจากตั้งอยู่ใน PATH1 ที่ทั้งสองใช้เหมือนกัน ขณะที่เส้น PATH2 ไม่ถูกใช้เลย กรณีดังกล่าว TE จะเข้ามาจัดการ Bandwidth ให้ใช้ประโยชน์ได้เต็มที่ยิ่งขึ้น เนื่องจากมีคุณสมบัติ IGP สามารถปรับช่องทางรองหรือ PATH2 ให้มีปริมาณโหลดและ Cost เท่าเทียมกับ PATH1 อย่างไรก็ตาม หากอยู่ในบริบทของ SP วิธีการนี้จะยุ่งยากกว่าเพราะมี Router เป็นจำนวนมาก

สำหรับ ATM Network มีวิธีแก้ปัญหาที่ยืดหยุ่นกว่า กล่าวคือ PVCs สามารถกำหนดช่องทางระหว่าง PE1-AS1 กับ PE2-AS1 ให้มี Cost เท่ากันได้เช่นกัน แต่ต้องมีข่าย PVCs อยู่เต็ม (Full Mesh) ระหว่างกลุ่ม Routers ทั้งสอง อย่างไรก็ตาม TE ที่ใช้ ATM ยังมีปัญหาอื่นอีกมาก โดยเฉพาะเมื่อ Link หรือ Node เสีย โดยในช่วงที่ Link หรือ Node เสียนั้น ข้อมูลจะลอยค้างอยู่ใน Network จำเป็นต้องมีข่าย Layer 3 Topology เสริมการทำงานของ Layer 2 TE โดยเป็นการปรับ Scalability Constraint ของ IGP เนื่องจากมีความเกี่ยวเนื่องอยู่กับ Layer 3



รูปที่ 3.21 MPLS-TE

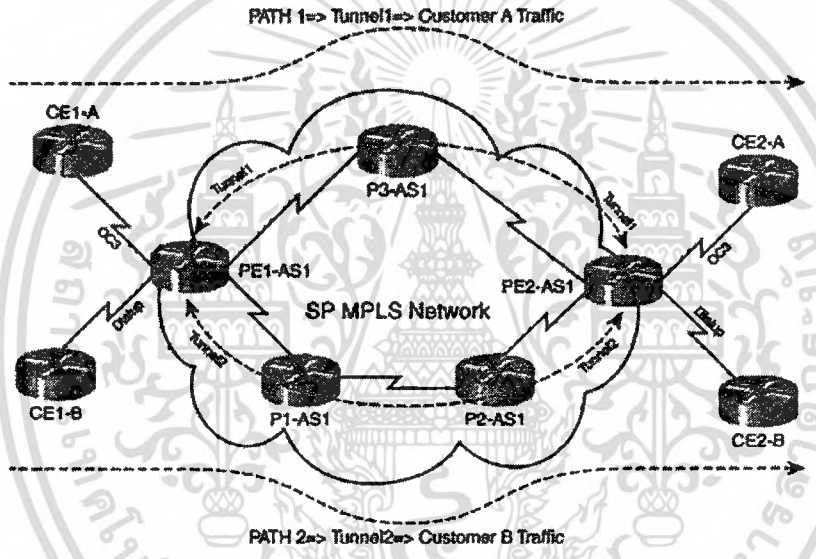
ข้อดีของ MPLS TE คือสามารถรวมความสามารถของ ATM's TE แต่ละชั้นการให้บริการ (COS) ของ IP เข้าด้วยกันได้ ในระบบ MPLS TE มีการจัดเส้นทางจราจรไว้หลายเส้นทางโดยคำนึงถึง Router อื่นๆ นับร้อย จึงไม่จำเป็นต้องมี Full Mesh PVCs เหมือนเครือข่าย ATM ดังนั้น หากรูปที่ 3.20 ใช้ MPLS TE การจัดเส้นทางจราจรก็จะอยู่ในรูป Label Switched Domain ดังรูปที่ 3.21 โดยมี TE Label Switched Paths หรือ TE Tunnels (Tunnel1 และ Tunnel2) เป็นตัวกำหนดเส้นทางระหว่าง PE1-AS1 กับ PE2-AS1

3.3.2 ลักษณะของ MPSL TE

ในระบบ IP เดิมเป็นการส่ง Packets ข้อมูลแบบ Per-Hop Basis โดยมองหาเส้นทาง (Route Lookup) และกำหนด Router แต่ละตัวไว้ก่อนตั้งแต่ต้นจนถึงปลายทาง พูดง่ายๆ คือ หากมีจุดปลายทางเดิมก็ใช้เส้นทางเดิมในการส่ง (Destination-Based) ส่งผลให้ใช้ Bandwidth ระหว่างคู่ Routers ของเครือข่ายผู้ให้บริการได้ไม่เต็มประสิทธิภาพ กล่าวคือยังมีเส้นทางอื่นอีกในเครือข่าย IP ที่ยังไม่ได้ใช้ประโยชน์ เพื่อหลีกเลี่ยงปัญหา Packet Drops จากการใช้ Bandwidth ไม่เต็มประสิทธิภาพดังกล่าว จึงมีการนำระบบ TE เข้ามาใช้เพื่อจัดการจราจรของข้อมูล ส่งผลให้จัดการและใช้ประโยชน์ Bandwidth ระหว่างคู่ Router ดีขึ้น หน้าที่หลักของ TE คือ แก้ไขภาวะแออัดข้อมูลใน Core ของ Network โดยกำหนด TP Maps ขึ้น เพื่อควบคุมการไหลของข้อมูลระหว่าง

สอง Router ให้เหมาะสม และใช้ Bandwidth ใน Core ของ Network ได้อย่างมีประสิทธิภาพ ผลการเพิ่มขนาดช่องทางและประสิทธิภาพใน Core ของ Network นี้ เกิดจากการรวบรวมช่องทาง, รูปแบบการจราจรไว้ขณะที่ผ่านจุดเชื่อมต่างๆ ใน Core ของ Network เพื่อเป็นข้อมูลประกอบการใช้ Bandwidth จึงสามารถกำหนด TE Tunnels หรือ Tunnel1 และ Tunnel2 (รูปที่ 3.21) ได้ตั้งแต่ตัว PE1-AS1 แยกเป็นเส้นทาง (PATH1, PATH2) เพื่อส่งข้อมูลได้อย่างมีประสิทธิภาพที่สุด

ในรูปที่ 3.21 TE Tunnels ที่กำหนดบน Routers มีทิศทางเดียว ดังนั้นหากต้องการให้ข้อมูลไหลได้ทั้ง 2 ทิศทางระหว่าง Routers PE1-AS1 กับ PE2-AS1 จำเป็นต้องกำหนด PE2-AS1 ลงเป็นต้นทางใน Tunnel1 และ Tunnel2 ด้วย ซึ่งระบบ MPLS Network จะมีการกำหนด Tunnel หรือ Provider Edge (PE) Routers เช่นนี้อยู่ตลอดเวลา โดยใช้ TE Tunnels หรือ LSPs ในการเชื่อมต่อระหว่าง Router ใน Core ของผู้ให้บริการ



รูปที่ 3.22 TE Tunnels ที่กำหนดจาก CoS ของผู้ใช้บริการ

นอกจากนี้ ระบบ MPLS TE ยังสามารถกำหนด Certain Classes ของการจราจรได้อีกด้วย กล่าวคือ หาก Customer A CE Routers เชื่อมต่อกับ SP Network โดยใช้ OC3 Link ขณะที่ Customer B เชื่อมต่อกับ SP Network โดยใช้ Dialup Link 64 K ก็สามารถกำหนด TE Tunnel โดยให้ TE Tunnel1 รองรับการจราจรของ Customer A และ Tunnel2 รองรับการจราจรของ Customer B ดังรูปที่ 3.22 ซึ่งเป็นการกำหนด Tunnel ทั้งบน PE1-AS1 และ PE2-AS1

MPLS TE Tunnels สามารถแนบข้อมูลหรือคุณสมบัติบางอย่างติดไปกับชุดข้อมูลที่ไหลอยู่ระหว่างต้นทางกับปลายทางได้ โดยอาจเพิ่มต้นทาง, ปลายทางของ Network เข้าในชุดข้อมูลตามที่ Bandwidth และ COS ที่ต้องการ TE Tunnel จะกำหนดเส้นทางจราจรของข้อมูลด้วย MPLS Label Switching โดยระบุ Label Switched Path (LSPs) ของ TE Tunnels จาก Source จนถึงจุดหมาย (ปกติจะเป็น PE Routers) จึงเป็นการกำหนดเส้นทาง (Mapping) แบบช่วงต่อช่วง กล่าวคือ TE Tunnels ไม่ได้ยึดติดว่าจะต้องผ่าน SP Network สู่ปลายทางด้วยเส้นทางใดเส้นทาง

หนึ่งโดยเฉพาะ เว้นแต่จะได้ระบุไว้เท่านั้น ทั้งนี้ ในระบบ MPLS LSP ตัว TE Tunnels สามารถเปลี่ยน Packet ของเส้นทางใหม่ภายใน Network ได้ โดยอาจกำหนดเส้นทางจาก IGP ที่ใช้ใน Core Backbone

จุดเด่นของระบบ MPLS TE คือ สามารถควบคุมเส้นทางจราจรของข้อมูลใน Network ได้อย่างยืดหยุ่น สามารถใช้เส้นทางรองอื่นๆ เมื่อเส้นทางหลักระหว่างสอง Routers ใดๆ เกิดล่ม หรือเสียหาย โดยมี Label Switching เป็นตัวกำหนดการไหลของข้อมูล กล่าวคือ เมื่อชุด Packet จาก CE Router มาถึง PE ชุด Packet จะถูกแนบด้วย Label แล้วไหลสู่ PE Router จุดหมาย และเมื่อถึงจุดหมายก็จะเปลี่ยน Labels ใหม่ให้ไหลไปตามเส้นทางที่เหมาะสมต่อไปอีกครั้ง โดยอยู่ในรูป IP Packet

OSPE หรือ IS-IS ส่วนขยาย (Extension) ของ TE ใช้สำหรับส่งข่าวสารที่เกี่ยวข้องสำหรับกำหนด Tunnel บน Router ข่าวสารส่วนขยายนี้ (Extension Carry Information) ประกอบด้วย Resource ที่ใช้งานได้ (Available) สำหรับสร้างเป็น Tunnel หรือ Bandwidth ระหว่าง Link ดังนั้น Link ที่ไม่มี Request Resource จึงไม่ถูกเลือกเป็นส่วนหนึ่งของ LSP Tunnel หรือ TE Tunnel ทั้งนี้ ระบบ MPLS TE จะใช้ Resource Reservation Protocol (RSVP) กับส่วนขยายในการส่งสัญญาณ (Signaling) เพื่อสนับสนุนคุณสมบัติของ TE Tunnel

ใน MPLS Domain ข้อมูล (Data Plane) ของ Router ปลายทางจะทำการเรียกหาข้อมูล Resource ที่ใช้งานได้จากทุก Link ใน MPLS TE Tunnel ข้อมูลเหล่านี้ได้มาจาก IGPs ซึ่งก็คือ OSPE และ IS-IS เนื่องจากข้อมูลการ Operation Of Flooding ของ Link ตลอดจน Routers ทั้งหมดบรรจุอยู่ใน IGP Domain ทั้งนี้ IS-IS จะพัฒนา TLV (Type 22) เพื่อส่งข้อมูล Resource ที่ใช้งานได้และสถานะ Link ใน LS-PDUs ขณะที่ OSPE สร้าง LSA Type 10 และข้อมูลสถานะของ Links เมื่อข้อมูลเหล่านี้ปรากฏ (Flood) อยู่ใน IGP Update แล้ว Router ต้นทางจะรวบรวมข้อมูลทั้งหมดจาก Resources ใน Network พร้อมทั้ง Topology เพื่อกำหนดเป็นชุด MPLS Routers ที่จะใช้ใน Tunnel ต่อไป

หลักสำคัญประการหนึ่งของ MPLS LE คือ Constraint Based Routing (CBR) ซึ่งเป็นการรวบรวมเส้นทางทั้งหมดที่เป็นไปได้ระหว่าง Source ถึงจุดหมายใน Network เอาไว้ หลักการ CBR นี้ทำให้ IP Network ทำงานได้ดีขึ้น เนื่องจากมี Cost ต่ำสุด แต่มีเส้นทางจากต้นถึงปลายทางให้เลือกมากมาย CBR ต้องการการสนับสนุนจาก IGP อย่าง OSPG หรือ IS-IS เพื่อกำหนด Routers ปลายทางใน MPLS Domain เช่นกัน โดยจะใช้ Constrained SPF คำนวณค่า Resource Availability และ Link Status ร่วมกับกับปัจจัยอื่นๆ เช่น Bandwidth, Policies และ Topology เพื่อพิจารณาและกำหนดเส้นทางระหว่าง Source ถึงปลายทาง

ผลการคำนวณของ CSPF ร่วมกับ Order ของ IP Address ซึ่งกำหนด Hop IP Addresses ถัดไปของ Routers จะอยู่ในรูป LSP และใช้ในทำการ Mapping ให้กับ TE Tunnel ต่อไป ทั้งนี้ ชุด Order ถูกกำหนดจาก Router ต้นทาง แล้วกระจายสู่ Routers อื่นๆ ที่อยู่ระหว่างเส้นทางใน

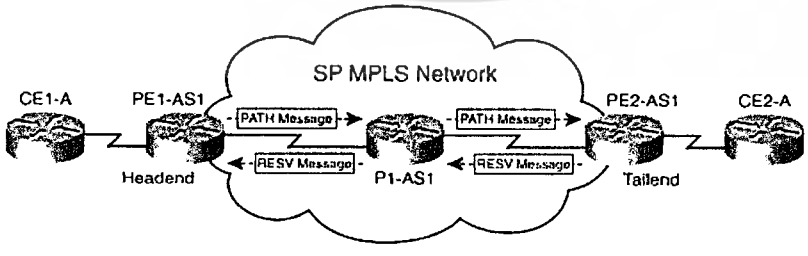
LSP แต่ไม่ได้กำหนดตายตัวว่า Routers นั้นจะต้องถูกเลือกหรือไม่ โดยมี RSVP ใน TE Extension ทำหน้าที่สงวน (Reserve) Resources ใน LSP Path พร้อมกับ Label ที่เกี่ยวข้องไว้สำหรับ TE Tunnel

3.3.3 RSVP ใน TE Extensions : การสร้างสัญญาณ

RSVP ทำหน้าที่สงวนช่อง Bandwidth ตลอดเส้นทางจาก Source ถึงจุดหมาย RSVP สร้างจาก Router ต้นทางใน Network และส่งไปยัง Resource ที่สามารถใช้งานได้ตลอดถึงปลายทางที่กำหนด Router ต้นทางจึงเป็น Source ของ MPLS TE Tunnel เสมอ ขณะที่ Router ปลายสุด (Tailend) จะถูกกำหนดเป็น Router จุดปลายทาง (Endpoint) ของ TE Tunnel หลังจากที่ RSVP ถูกส่งออกไปแล้ว มันจะรวบรวมสถานะของ Router (Source) ในเส้นทางที่ใช้ได้ทั้งหมดไว้ใน Path Massage RSVP จึงทำหน้าที่เสมือนตัวส่งสาร, ร้องขอเส้นทางการจราจร และรวบรวมข้อมูลจากการร้องขอนั้นมาสร้างเป็นเส้นทางใน Network

RSVP หลักๆ ที่ใช้ใน TE ประกอบด้วย 4 อย่าง คือ RSVP PATH Message, RSVP RESERVATION Message, RSVP Error Messages และ RSVP TEAR Messages ระบบ MPLS TE จะใช้ RSVP เป็นเครื่องยืนยันและตรวจสอบ Resource ที่ใช้ได้ รวมถึงใช้ MPLS Labels สร้าง MPLS TE LSP ตลอดเส้นทางที่ผ่าน Router ในระบบ Network

- **RSVP PATH Message** – เป็นข้อมูลที่สร้างจาก Router ต้นทางและส่งผ่านไปใน Network ตลอดเส้นทาง TE LSP ข้างหน้า โดยในแต่ละ Hop ข้อมูล PATH Massage จะทำการตรวจสอบการใช้งานได้ (Availability) ของ Resource และเก็บข้อมูลเหล่านี้เอาไว้ สำหรับ Network ในรูปที่ 3.23 PATH Message ถูกสร้างขึ้นจาก Router PE1-AS1 ซึ่งเป็น Router ต้นทาง และส่งต่อลงมาตาม Resource ที่ตรวจสอบจากแต่ละ Hop แล้วว่าใช้ได้ (P1-AS1 และ PE2-AS1) ทั้งนี้ RSVP PATH Message จะทำงานตามที่ Label ใน MPLS TE Domain กำหนด เนื่องจาก TE Domain ทำงานเรียงลำดับลงมาตาม Mode ที่ Label นั้นกำหนดไว้ ซึ่งตามปกติ Label จะกำหนดหน้าที่ของ Router ต้นทางก่อน และไล่เรียงลงมาเรื่อยๆ ตามเส้นทาง

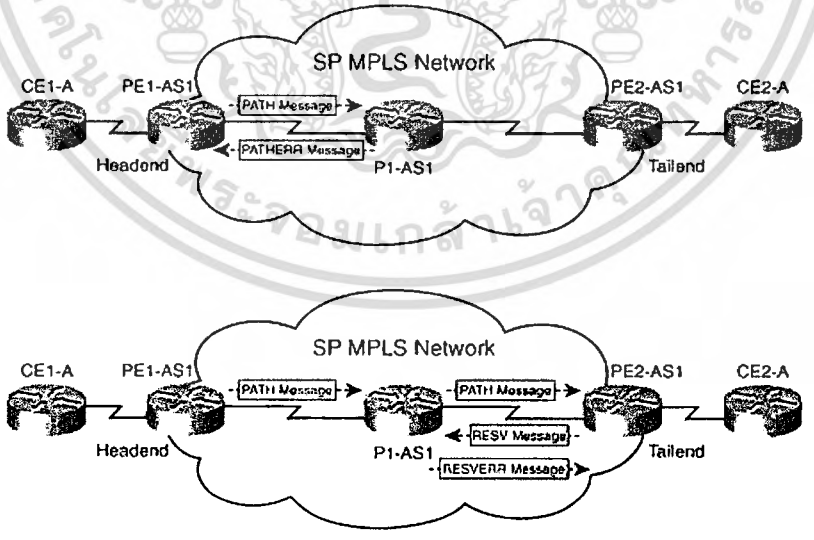


รูปที่ 3.23 RSVP Path และ reservation Messages

- **RSVP RESERVATION Message** – เป็นข้อมูลที่สร้างจาก Router ปลายทางใน MPLS TE Domain ใช้สำหรับสงวนช่องทางส่งข้อมูลตามที่ PATH Message กำหนด ใน

Network รูปที่ 3.23 PE2-AS1 เป็นตัวสร้าง RSVP RESERVATION ซึ่งมีความสอดคล้องกับ PATH Message PATH Message จึงทำหน้าที่สงวนเส้นทาง (Reservation) ขณะที่ RESERVATION Message ทำหน้าที่ยืนยันการร้องขอ (Request) Resource ที่ใช้งานได้ RSVP RESERVATION Message ทำหน้าที่กำหนด Label ให้ LSP ทำการ Mapping เส้นทางแก่ TE Tunnel โดยขณะที่ MPLS Domain Label ถูกกำหนดและกระจายลงไปนั้น ตัว Router ปลายทาง หรือ Edge LSR ที่ช่องทางออกจะสร้าง Label Mapping ไปยัง TE LSP ก่อน แล้วแพร่ไปยังต้นทาง กระบวนการนี้จะเกิดซ้ำๆ ย้อนขึ้นไปบน Hop แต่ละตัวที่ Local Label ทำการ Mapping ไว้จนกระทั่งถึง Router ต้นทาง

- **RSVP Error Messages** – ในกรณีที่ Resource ที่ต้องการนั้นใช้งานไม่ได้ ตัว Router จะสร้าง RSVP Error Message และส่งให้กับ Router ที่ต้องการทราบ ตัวอย่างเช่น หาก Router P1-AS1 ไม่สามารถเรียกใช้เป็น Resource ได้ตาม PATH Message ที่ PE1-AS1 สร้าง (Router ต้นทาง) Router ก็จะสร้าง PATH ERROR (PATHERR) Message และส่งย้อนกลับไปยัง LSR PE1-AS1 ในตัวอย่างรูปที่ 3.24 เมื่อ RSVP PATH Message ส่งถึง Router ปลายทาง Router PE2-AS1 ที่อยู่ปลายทางก็จะสร้าง RESERVATION Message ขึ้น หากช่วงตั้งแต่ P1-AS1 รับ PATH Message จาก PE1-AS1 จนถึงรับ RESERVATION Message จาก PE2-AS1 นั้น P1-AS1 เกิดระบุว่ามี Resource เสียเพื่อขอยืนยันคำร้องใหม่อีกครั้ง P1-AS1 ก็จะส่ง RESERVATION ERROR (RESVERR) Message ลงไปให้ LSR PE2-AS1 เพื่อระงับการ Reservation ดังกล่าวไว้ดังแสดงในรูปที่ 3.24



รูปที่ 3.24 RSVP Path Error และ Reservation Error Messages

- **RSVP Tear Messages** Tear Message ที่ RSVP สร้างมี 2 ชนิด คือ PATH Tear Message และ RESERVATION Tear Message ข้อมูล Tear Message เหล่านี้ทำหน้าที่เคลียร์ PATH หรือ RESERVATION State ของ Router แบบทันทีทันใด กระบวนการเคลียร์ PATH

หรือ RESERVATION State ของ Router ด้วย Tear Message นี้ จะทำให้ Resource กลับมารองรับคำสั่งอื่นๆ ต่อไปได้อีกครั้งหนึ่ง PATH Tear Message มักปรากฏอยู่ใน Inter-Area LSP Creation ซึ่งไม่ได้กำหนดให้ Reroute อย่างรวดเร็ว โดยหาก Link ล่ม LSR ที่เกี่ยวข้องกับ Link นั้นก็จะสร้าง RSVP PATH Error และ RESV Tear Message ไปยังต้นทาง จากนั้นต้นทางจะสร้าง RSVP PATH Tear Message ขึ้น ทำให้ Path Option ที่เกี่ยวข้องในช่วงเวลานั้นถูกระงับ และใช้ Path Option ถัดไปได้ทันที

3.3.4 การทำงานของ RSVP ในระบบ MPLS TE

ผลการคำนวณของ CSPF หรือ CBR ที่ Router ต้นทาง เป็นรายการ ของ IP Address ซึ่งระบุถึง Hops ถัดไปตลอดเส้นทาง TE Tunnel หรือ LSP รายการที่ถูกคำนวณนี้จะถูกแจ้งไปยัง Router นับร้อยที่เป็น Source ของ TE Tunnel เท่านั้น โดย Router ต้นทางจะจัดเตรียมข้อมูลให้กับ Routers อื่นๆ ในเส้นทาง TE Tunnel ผ่านการส่งสัญญาณของ RSVP เพื่อร้องขอหรือยืนยันการใช้งานได้ของ Source ตลอด Tunnel นั้นๆ ขณะที่ RSVP ส่วนขยายจะทำหน้าที่สงวน Source TE ที่เหมาะสมกับ LSR ภายในเส้นทางซึ่ง Router ต้นทางเป็นผู้กำหนด Labels Mapping ให้กับ TE Tunnel LSP

RSVP ส่วนขยายที่ใช้สร้างสัญญาณในระบบ MPLS เพื่อกำหนดการทำงานของ TE มีรายละเอียดดังตาราง 3.1 ดังนี้

ตารางที่ 3.1 RSVP Objects

Object	Message	Function
LABEL_REQUEST	PATH	ใช้ร้องขอ Label Mapping ให้กับ TE Tunnel หรือ LSP; สร้างจาก Router ต้นทาง แนบอยู่ใน PATH Message
LABEL	RESERVATION	ใช้กำหนด (Allocate) Label Mapping ให้กับ TE Tunnel หรือ LSP; สร้างจาก Router ปลายทาง โดยแนบอยู่ใน RESERVATION Message และส่งทวนเส้นทางขึ้นไป
EXPLICIT_ROUTE	PATH	พ่วงติดไปกับ PATH Message ใช้ร้องขอหรือยืนยัน Path/Route อย่างใดอย่างหนึ่งให้กับ Tunnel
ROCORD_ROUTE	PATH, RESERVATION	ทำหน้าที่คล้าย Record Option ของ ICMP Ping ใช้แนบไปกับ PATH หรือ RESERVATION Messages เพื่อแจ้ง Node เดิมให้ทราบถึง Route/Path ของ LSP TE Tunnel

ตารางที่ 3.1 (ต่อ)

Object	Message	Function
SESSION_ATTRIBUTE	PATH	ใช้กำหนด Session Parameters เฉพาะให้แก่ TE LSP Tunnel

ระหว่างกระบวนการเซตเส้นทางให้ LSP TE Tunnels, RSVP Message จะประกอบด้วยส่วนขยายเหล่านี้ 1 ชุดหรือมากกว่า เพื่อระบุความสำคัญของ Message และส่วนประกอบของมัน

Path Message ประกอบด้วยข้อมูล ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 RSVP Object ใน Path Message

Object	Message
SESSION	ระบุ Source และจุดหมายของ LSP Tunnel ปกติจะทำการระบุด้วย IP Address ของ Loopback Interface ที่เกี่ยวข้อง และ Router ปลายทาง
SESSION_ATTRIBUTE	ระบุลักษณะของ LSP Tunnel เฉพาะ เช่น Bandwidth ที่ต้องการ และ Resource ที่กำหนดใช้ใน Tunnel
EXPLICIT_ROUTE	กำหนดให้ทำการ Populate ด้วยรายการของ Hop ถัดไป โดยอาจระบุเองหรือคำนวณด้วย Constraint-Based SPF ก็ได้ มีผลให้ Hop ก่อนหน้า (PHOP) จะถูกเซตเป็น Interface Address ของ Router ที่ผ่านมา ส่วน Record_Route (RRO) เป็นการ Populate ด้วย Address เดิม
RECORD_ROUTE	กำหนดให้ทำการ Populate ด้วย Interface Address ของ Router ที่ผ่านมา ในเส้นทาง LSP Tunnel
SENDER_TEMPLATE	เป็นส่วนเพิ่มเติมจาก Object ข้างต้น กำหนดให้ผู้ที่ส่ง Object ลงใน Path Message แสดงค่า Interface Address ที่จะใช้เป็น LSP-ID ของ Tunnel นั้น, ค่านี้กำหนดจาก Router ต้นทาง

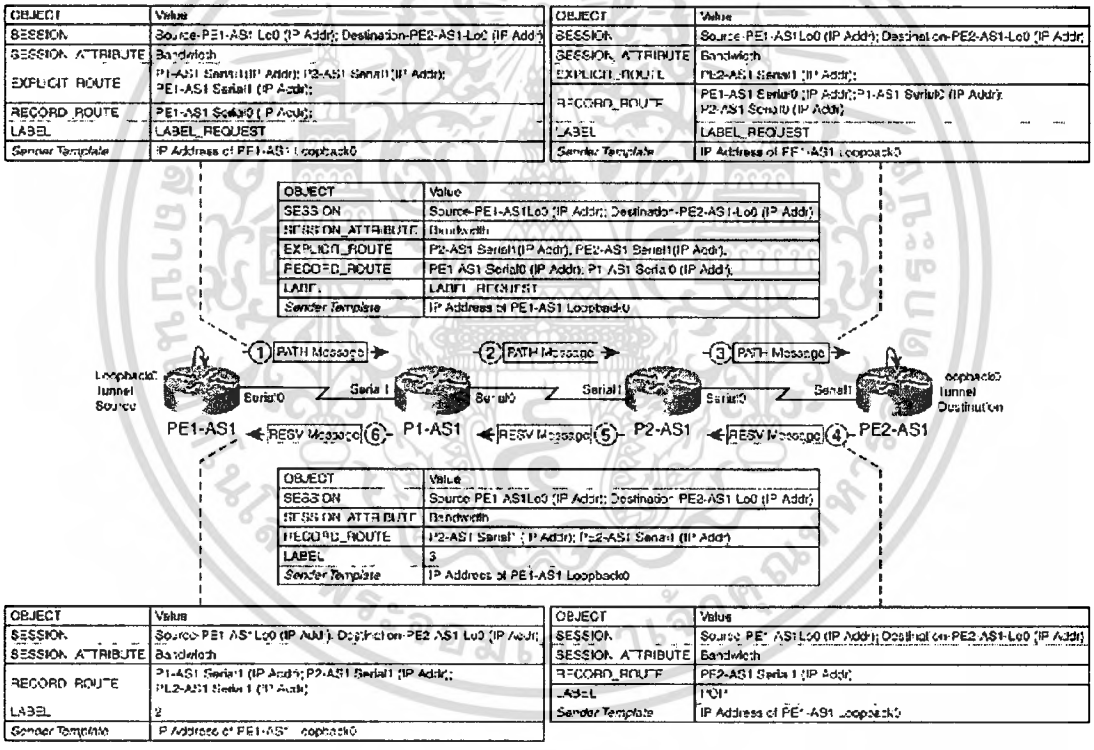
ขั้นตอนการส่ง PATH และ RESV Message ในภาพที่ 3.25 มีรายละเอียดดังนี้

ขั้นที่ 1 ค่าต่างๆ ตามตารางที่ 3.2 ถูก PE1-AS1 ซึ่งเป็น Router ต้นทางนำไปใช้ และส่ง PATH Message ไปยัง Hop Router ถัดไป ใน LSP Tunnel Path

ขั้นที่ 2 เมื่อ P1-AS1 ได้รับ PATH Message แล้ว Router จะทำการตรวจสอบคำสั่ง EXPLICIT_ROUTE และพิจารณา L-bit ของ RSVP Path Message เพื่อดูว่า Hop ถัดไปเชื่อมต่อกับ Network โดยตรงหรือไม่ หาก L-bit ถูกเซต แสดงว่า Local Router ไม่ได้

เชื่อมต่อโดยตรงกับ Hop ถัดไปใน LSP Tunnel Path ดังนั้น ตัว Router จะทำการคำนวณ Constrained-SPF เพื่อกำหนด Hop ถัดไปใน Tunnel Path

แต่หาก L-bit ไม่ถูกเซต แสดงว่า Router P1-AS1 เชื่อมต่อโดยตรงกับ Hop ถัดไปใน LSP Tunnel Path ข้อมูลทั้งหมดใน EXPLICIT_ROUTE Mapping จะถูกส่งไปยัง Local Router (P1-AS1) และส่ง PATH Message ไปยัง Hop ถัดไปตามที่กำหนดไว้ใน EXPLICIT_ROUTE ขณะเดียวกัน P2-AS1 จะทำการ Update ข้อมูล และแนบคำสั่ง RECORD_ROUTE เข้าไป เพื่อแสดง Local Interface ที่ผ่านมาในเส้นทาง LSP Tunnel ค่า PATH Message ที่แสดงในรูปที่ 3.25 เป็น PATH Message ที่ส่งจาก P1-AS1 ไปยัง P2-AS1 หลังจากที่ Update ค่าแล้ว ทั้งนี้ P1-AS1 จะเอาข้อมูลอ้างอิง Local Interface ใน EXPLICIT_ROUTE ออกแล้วเติมรายการ Interface ที่ผ่านมาลงใน RECORD_ROUTE



รูปที่ 3.25 RSVP Path / Reservation Messages และ Object Values

ขั้นที่3 ที่ P2-AS1 เกิดการย้ายเอา Local Interface ใน EXPLICIT_ROUTE ออก และแนบ Interface ที่ผ่านมาใส่ลงใน RECORD_ROUTE อีกหลายครั้ง

ขั้นที่ 4 หลังจาก Router PE2-AS1 ซึ่งอยู่ปลายทางได้รับ RSVP PATH Message แล้ว ก็จะสร้าง RESERVATION Message ขึ้น ภายใต้มีแนวคิดให้กระบวนการ Label Allocation Process เกิดจาก Router ปลายทาง เป็นผู้สร้าง RESERVATION Message ส่งย้อนขึ้นไป ดังนั้น เมื่อ PE2-AS1 สร้าง RESERVATION Message ขึ้น ตัว Router จะกำหนด

POP Label ให้กับ LSP Tunnel (Popping Hop รองสุดท้าย) ระหว่างนี้ RESERVATION Message ที่ Router ปลายทางส่งย้อยกลับไป Router ต้นทางจะมี RECORD_ROUTE ซึ่งชี้ Outgoing Interface แนบอยู่ด้วย ดังนั้น RECORD_ROUTE Object ถูกใส่กลับเข้าไปใน RESERVATION Message ดังค่าที่แสดงในรูปที่ 3.25

ขั้นที่ 5 เมื่อ Reservation Message ส่งถึง P2-AS1 แล้ว RECORD_ROUTE จะถูก Prepended ด้วย Outgoing Interface และสร้าง Local Label Mapping ให้ LSP และทำการ Map ตามที่ LABEL Object กำหนด (ค่า Arbitrary Value 3 ใน LABEL แสดงดังรูปที่ 3.25)

ขั้นที่ 6 เกิดกระบวนการข้างต้นที่ P1-AS1 ซ้ำๆ จากนั้น PE1-AS1 จะได้รับ RESERVATION Message

ขั้นที่ 7 เมื่อ PE1-AS1 ได้รับ RESERVATION Message แล้ว RECORD-ROUTE ซึ่งระบุ TE LSP อยู่จะถูกนำมาพิจารณาร่วมกับ Bandwidth หรือ Resource ที่กำหนดใน SESSION จากนั้น Labels ที่ Map ให้ LSP จะถูกนำมาใช้เป็น Regular MPLS โดย Local Label จะถูก Map ด้วย Hop Label ถัดไป

สำหรับการทำงานของ RSVP ในระบบ MPLS TE นั้น RSVP กับส่วนขยายจะทำหน้าที่ยืนยัน LSP และสงวน Resource ที่ร้องขอบน Router ทุกตัวตลอดเส้นทาง LSP Path และนำ MPLS Label มาสร้างเป็น MPLS LSP ตลอดเส้นทางใน Network ทั้งนี้ Routers จะบันทึกสำเนาของ PATH Request ไว้ขณะที่ Request ถูกส่งไปยัง Hop LSR ถัดไป และขณะที่ LSR เดิมได้รับ Reservation Messages จะมีการกำหนด Interface ส่งไปยัง Interface ทางออกและ Router อีกนับร้อย ในหัวข้อถัดไปเราจะนำเสนอกระบวนการคำนวณ Constraint-Based SPF และ Link-State Protocol ที่จำเป็นต้องใช้ใน MPLS TE แบบไดนามิคใน Core ของผู้ให้บริการ

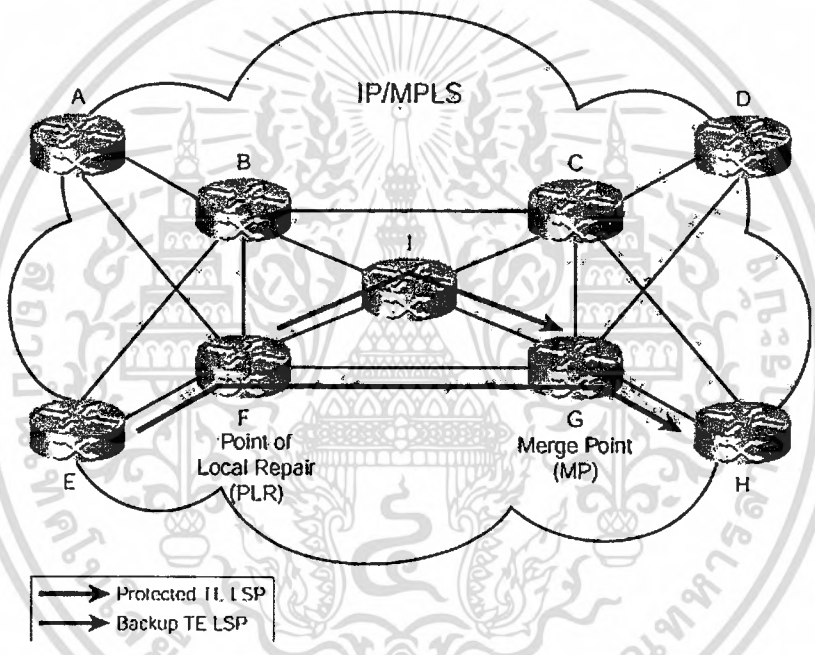
3.3.5 Fast Reroute

MPLS TE มีระบบซ่อมแซม TE LSPs โดยใช้ FRR มีจุดประสงค์เพื่อปกป้องการจราจรข้อมูลในกรณีที่ Network ล่ม ซึ่งสำคัญต่อข้อมูลแบบ Real-Time หรือข้อมูลที่มีค่า Packet สูญหายจำกัดอย่างมาก FRR ปกป้องการจราจรของข้อมูลโดยการสร้างสัญญาณ Backup TE LSP ไว้ล่วงหน้าสำหรับ Reroute ข้อมูลในกรณีที่การส่งนั้นล้มเหลว เนื่องจาก FRR สามารถ Reroute ข้อมูลได้ภายใน 50 ms ประกอบกับ Node ต้นทางบริเวณที่ล่มมีการเตรียม Backup TE LSP ใหม่อยู่เสมอ การประมวลเส้นทางและสร้างสัญญาณ TE LSP ใหม่สำหรับใช้ Reroute จึงไม่เกิดการ Delay

MPLS TE FRR มีเทคโนโลยีปกป้องการจราจรข้อมูลอยู่ 2 แบบ คือ การ Backup อย่างง่าย (Facility Backup)และ Backup แบบหนึ่งต่อหนึ่ง (One-to-One Backup) Backup อย่างง่าย ใช้วิธี Reroute ทุก TE LSPs โดยการทำ Label Stacking และมีการ Backup TE LSP เพียงครั้งเดียว

ขณะที่ Backup แบบหนึ่งต่อหนึ่งไม่ได้ใช้ Label Stacking แต่จะเตรียม Backup TE LSP สำหรับปกป้อง TE LSP ใหม่ทุกชุด

รูปที่ 3.26 แสดงตัวอย่าง MPLS Network ที่ใช้ FRR จากภาพ Node E ส่งสัญญาณ TE LSPs ไปยัง Node H FRR จะปกป้องความเสียหายของ TE LSP อันเกิดจาก Link ระหว่าง Node F กับ G ล่ม โดยกำหนดให้ Node F ทำการ Backup TE LSP ไว้สำหรับ Reroute Node F จึงทำหน้าที่เสมือน Point of Local Repair (PLR) นอกจากนี้ยังจัดเตรียมสัญญาณ Backup TE LSP จาก Node I ถึง Node G ไว้สำหรับ Bypass ได้อีกทางหนึ่งด้วย ทั้งนี้ PLR จะอยู่ที่ต้นทางของการ Backup TE LSP เสมอ ขณะที่ Node G จะมีชื่อว่า Merge Point (MP) เป็น Node ที่การจราจรออกไป หลังจาก Link ล่มและส่ง TE LSP ใหม่แล้ว



รูปที่ 3.26 Fast Reroute ใน MPLS Network

MPLS TE FRR มีชุด RSVP ขยาย สำหรับสร้างสัญญาณปกป้อง TE LSP ดังนี้

- FAST_REROUTE Object ใหม่ ทำหน้าที่กำหนดคุณสมบัติของ Backup TE LSP คุณสมบัติที่กำหนดประกอบด้วย ลำดับความสำคัญ (Setup and Holding), Hop Limit, Bandwidth และ Attributes FAST_REROUTE Object ยังกำหนดสภาวะของ Node ด้วยว่า ควรปกป้อง TE LSP ด้วย Backup อย่างง่าย หรือแบบหนึ่งต่อหนึ่ง
- RECORD_ROUTE Object ขยาย ใช้บ่งชี้ความสามารถในการปกป้องของแต่ละ Hop และชนิดของการปกป้อง (ปกป้อง Link, Node, หรือ Bandwidth)
- SESSION_ATTRIBUTE Object ขยาย ใช้บอกสถานะของ TE LSP ที่ต้องการการปกป้อง และชนิดของการปกป้อง (ปกป้อง Link, Node หรือ Bandwidth)

ตารางที่ 3.3 RSVR Object ที่ใช้สำหรับ MPLS TE FRR

RSVP Object	RSVP Message	FRR Function
FAST_REROUTE	Path	กำหนดเทคนิคการ Backup ให้แก่ FRR (Backup อย่างง่าย หรือ Backup แบบหนึ่งต่อหนึ่ง) และกำหนดคุณสมบัติของการ Backup (ลำดับความสำคัญ, Bandwidth, Attribute ของ Backup TE LSP)
RECORD_ROUTE	Path, Resv	บันทึกรายการ และรายละเอียดของ Hops/Labels ที่จะใช้ปกป้อง TE LSP ประกอบด้วย สถานะการปกป้อง และชนิดการปกป้องของแต่ละ Hop
SESSION_ATTRIBUTE	Path	บ่งชี้สภาพการปกป้องและชนิดของการปกป้อง ที่ TE LSP ต้องการ

หมายเหตุ : สำหรับเทคนิคการ Backup แบบหนึ่งต่อหนึ่ง จะมี RSVP Object เพิ่มเติมอีกชนิดหนึ่ง คือ DETOUR

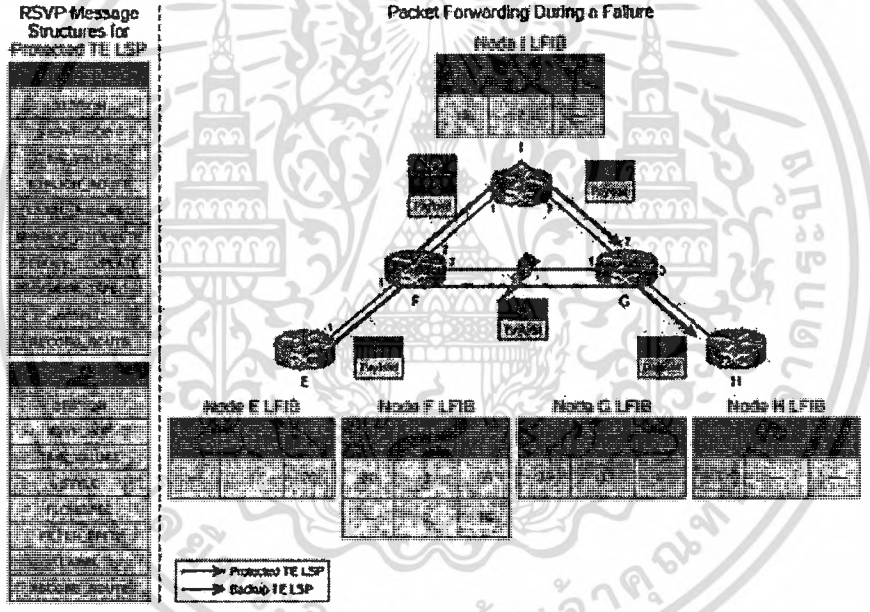
เมื่อการปกป้อง TE LSP ใน Network ไม่สำเร็จ MPLS TE FRR สามารถทำการ Restoration TE LSP ใหม่ได้ทั้งแบบ Global และ Local การทำ Global Restoration อาศัยการ Rerouting ของต้นทาง โดยเมื่อการปกป้องอย่างง่ายล่ม PLR จะส่ง PathErr Message ไปยังต้นทาง ทั้งนี้ Node ต้นทางสามารถรู้สภาพของการล่มได้จากการแจ้งของ RSVP หรือจาก IGP Update ในกรณีที่การล่มนั้นเกิดใน IGP เดียวกัน เมื่อได้รับสัญญาณแจ้งการล่มแล้ว ต้นทางสามารถ Reroute TE LSP รอบๆ บริเวณที่ล่มได้ทั้งหมด ส่วน Local Restoration เป็นการ Reroutes โดยอาศัย Backup เฉพาะจุด กล่าวคือ PLR จะส่งสัญญาณปกป้อง TE LSP ซ้ำไปยังจุดเชื่อมต่อ แล้วกำหนดให้ต้นทางระหว่างจุดเชื่อมต่อส่งสัญญาณซ้ำ การทำ Global Restoration นับว่ามีประโยชน์มากกว่า เนื่องจากปกป้อง TE LSP ได้ครอบคลุมกว่า

การปกป้อง Link

การปกป้อง Link ใช้ Backup TE LSP ในการกำหนดค่าของ PLR Next-Hop (NHOP) โดย Node ตลอดเส้นทางจะเก็บค่า Backup TE LSP ไว้และถูกกำหนดให้เป็น NHOP Downstream ทั้งนี้อาจมี Backup TE LSP อยู่แล้ว หรือให้ Node คำนวณเส้นทางที่เหมาะสมแล้วส่งสัญญาณก็ได้ Node ใดๆ ที่มี Backup LSP อยู่จะมีสถานะเสมือน PLR โดยจะส่งสัญญาณกลับยังไปต้นทางโดยดูรายละเอียดจาก RECORD_ROUTE Object เมื่อ Link เกิดล่ม PLR จะสั่งให้

ต้นทางทำการ Reroute TE LSP ที่ระบุโดยใช้ Backup TE LSP ที่บันทึกไว้ กระบวนการ Rerouting ทั้งหมดประกอบด้วยการทำ Pushing TE LSP Label ที่จะปกป้อง (ทำก่อนการล่ม) จากนั้นจึงทำการ Stacking ค่า Backup TE LSP Label ไปไว้บนสุด

รูปที่ 3.27 แสดงการปกป้อง Link จากภาพ Node E ส่ง TE LSP ไปยัง Node H ระหว่างนี้ SESSION_ATTRIBUTE Object จะทำการระบุว่า TE LSP ต้องการการปกป้อง Link แบบไหน Node F ดำเนินการตาม Object แล้ว Backup ว่าการไปสู่ NHOP (Node G) สามารถอ้อมผ่าน Node I ออกไป เมื่อ Link ระหว่าง Node F และ G ล่ม Node F จะตรวจสอบการล่มในพื้นที่และหาทางเลือกใหม่เพื่อปกป้อง TE LSP โดยทำการ Push Label 35 ตามที่ NHOP ระบุ พร้อมกับ Push Label 16 เพื่อ Reroute การจราจรด้วย Backup TE LSP จากนั้น Backup จะเปลี่ยนมาใช้ Node I ซึ่งกรณีนี้ Node I จะทำหน้าที่ PHP Operation จากนั้น Packet จะถูกส่งถึง MP (Node G) พร้อมกับ Label 35 และส่งถึง Node H ได้ในที่สุด



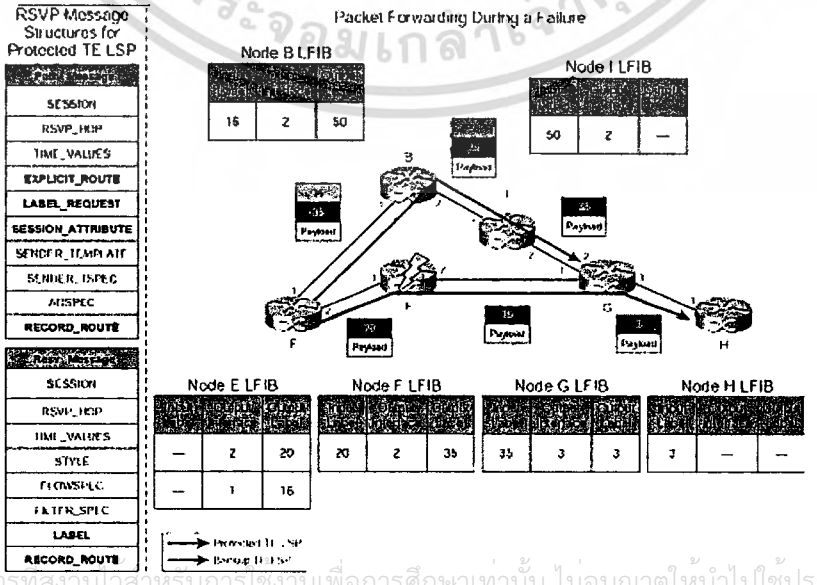
รูปที่ 3.27 MPLS TE FRR Link Protection

การปกป้อง Link สามารถป้องกันการล่มของกลุ่ม Link หรือ Shared-Link Risk Groups (SLRG) ได้ด้วย เนื่องจากบางกรณี Network อาจมี Link หลายตัวมีโอกาสล่มพร้อมกันสูง ซึ่งจะส่งผลต่อ Link ต่างๆ ที่ใช้ Infrastructure เดียวกัน (Layer 2, Layer 1 หรือสิ่งอำนวยความสะดวกทาง Physical) การคำนวณเส้นทางสำหรับ Backup TE LSP จึงควรรวม SLRGs เข้าไปด้วย เพื่อหลีกเลี่ยงการล่มของ Link พร้อมๆ กัน โดย PLRs สามารถรู้ค่า SLRGs แบบไดนามิคได้จาก IGP Extensions หรือผ่านการ Configuration ในพื้นที่ ทั้งนี้ค่า SLRGs อาจมีผลต่อการคำนวณเส้นทางของ PLR แต่ไม่มีผลต่อการปกป้อง Link

การปกป้อง Node

การปกป้อง Node จะใช้ Backup TE LSP กำหนดค่าของ PLR Next-Next Hop (NNHOP) เมื่อ Node ส่งสัญญาณ TE LSP และระบุให้ทำการปกป้อง Node ตลอดเส้นทางจะเก็บ Backup TE LSP ไว้และกำหนดให้เป็น NNHOP Downstream โดย Backup TE LSP ดังกล่าวอาจกำหนดไว้ล่วงหน้า หรือให้ Node คำนวณเส้นทางที่เหมาะสมและส่งสัญญาณก็ได้ Nodes ที่มี TE Backup จะมีหน้าที่เป็น PLR และคอยส่งสัญญาณย้อนกลับไปต้นทางโดยใช้ RECORD_ROUTE Object เมื่อ NHOP ล่ม PLR จะทำการ Reroute TE LSPs ใหม่โดยใช้ค่า Backup TE LSP กระบวนการ Rerouting ดังกล่าวประกอบด้วย การ Pushing TE LSP Label ตามที่ NNHOP ระบุ จากนั้นจะทำการ Stacking TE Backup LSP Label ไว้บนสุด โดย PLR สามารถรู้ค่า NNHOP Label ได้จาก Resv Message ที่ได้จาก RECORD_ROUTE Object ทั้งนี้การปกป้อง Node สามารถป้องกัน SRLG ล่มได้ด้วย เนื่องจากรายละเอียดข้างต้นชี้ว่า SRLGs มีผลต่อการคำนวณเส้นทาง แต่ไม่มีผลต่อการปฏิบัติงานของ FRR จึงปกป้อง Node ได้พร้อมกับปกป้อง SRLG ล่ม

รูปที่ 3.28 แสดงการปกป้อง Node จากภาพ Node E ส่งสัญญาณ TE LSP ไปยัง Node H โดยมี SESSION_ATTRIBUTE บ่งชี้สภาพการปกป้องที่ Node ต้องการ ในกรณีนี้ Node F ทำการ Backup ค่าการไปถึง NNHOP (Node G) ว่าสามารถอ้อมผ่านทาง Node B และ I ได้ เมื่อ Node F ล่ม Node E จะตรวจสอบการล่มในพื้นที่และปรับการส่งใหม่ โดยแทนที่จะ Push Label 20 เหมือนก่อนล่ม Node E จะ Push Label 35 ตามที่ NNHOP (Node G) ระบุ และ Push Label 16 เพื่อ Reroute การจราจรด้วย Backup TE LSP จากนั้น Node B และ I จะเปลี่ยนมารับ Backup TE LSP โดยไม่จำเป็นต้องรู้การปกป้อง TE LSP จากนั้น Packet ก็จะส่งถึง MP (Node G) พร้อมกับ Label 35 และไปถึง Node H ในที่สุด



รูปที่ 3.28 MPLS TE FRR Node Protection

3.4 Quality of Service (QoS)

ในอดีต IP ที่ใช้ Best-Effort Protocol ถือว่าจัดการจราจรข้อมูลถึงปลายทางได้รวดเร็วที่สุด หากแต่ไม่ได้รับประกันว่าข้อมูลนั้นจะถึงปลายทางจริงหรือไม่

ระบบ Network ในปัจจุบันต้องรองรับ Application จำนวนมาก ข้อมูล Application ส่วนใหญ่มักเป็น TCP ซึ่งส่งผลให้เกิดความไม่แน่นอนของ Bandwidth, ค่า Latency, Jitter และการสูญเสีย Packet ข้อมูล จึงจำเป็นต้องรับประกันการส่งเพื่อให้เกิดปัญหาดังกล่าวน้อยที่สุด

Application ใหม่ๆ ในปัจจุบันมีข้อมูลภาพและเสียงแบบ Real-Time มากมาย ประกอบกับองค์กรรัฐบาลและสถาบันการศึกษา ต่างก็มี IP Protocol สำหรับรองรับโครงข่ายส่วนตัวกันมากขึ้น ส่งผลกระทบต่อความไม่แน่นอนของ Bandwidth, ค่า Latency, Jitter และสูญเสีย Packet ซึ่งการบริการแบบ Best-Effort Service ไม่พอจะรองรับได้อีกต่อไป

สถาปัตยกรรม QoS เริ่มกำหนดใช้ตั้งแต่ปี 1990 โดยคณะกรรมการ Internet Engineering Task Force (IETF) ซึ่งกำหนดสถาปัตยกรรม Qos ไว้ 2 แบบ คือ Integrated Services (IntServ) และ Differentiated Services (DiffServ) โดยเริ่มใช้สถาปัตยกรรม IntServ ก่อน จึงเกิด DiffServ ขึ้นภายหลัง ทุกวันนี้คณะกรรมการ IETF กำหนดให้ MPLS ทั้งหมดต้องรองรับสถาปัตยกรรม DiffServ

สถาปัตยกรรมทั้งสองมีแนวคิดและการใช้ใน IP แตกต่างกัน ต่างก็มีจุดแข็งจุดอ่อนแต่เติมเต็มซึ่งกันและกัน และมีจุดหมายเพื่อจัดการจราจรข้อมูลให้ได้มากและเร็วที่สุดเหมือนกัน

3.4.1 Integrated Service

กลุ่มงาน IntServ ได้ออกแบบสถาปัตยกรรมนี้ขึ้นตามข้อกำหนดของ IETE เริ่มใช้ตั้งแต่ปี 1994 การออกแบบให้สัมพันธ์กับ IntServ over Specific Link Layers (ISSLL) และ Resource Reservation Protocol (RSVP) อย่างมาก โดย ISSLL จะทำหน้าที่กำหนดการทำงานของ IntServ ระหว่างจุดเชื่อมต่อที่มี Link-Layer Protocols ต่างกัน (เช่น Ethernet กับ ATM) ส่วน RSVP ทำหน้าที่กำหนด RSVP Protocol ให้ IntServ เลือกใช้สำหรับส่งสัญญาณ กลุ่มงาน IntServ ได้กำหนด RFCs ไว้ทั้งหมด 32 ชุด โดยมี 24 ชุดอยู่ใน IETF Standard Strack แต่ได้ยกเลิกไปในช่วงปี 2000 ถึง 2002

IETF พยายามดัดแปลงสถาปัตยกรรม Internet เดิมให้รองรับ Application แบบ Real-Time มากขึ้น โดยพิจารณาทางเลือกง่ายๆ แต่ไม่สมบูรณ์นัก ดังนี้

- ออกแบบอัลกอริทึมให้จัดคิวข้อมูลอย่างยุติธรรม (Fair-Queuing) เพื่อแก้ปัญหาการส่งข้อมูลทั่วไป และข้อมูล Application แบบ Real-Time ที่ไม่เท่าเทียมกัน แต่ไม่สามารถรับประกันว่าจะเกิดปัญหา Delay หรือ Jitter หรือไม่
- กำหนดให้ Service ที่ต่างกันใช้เครือข่ายแยกกัน แต่พบว่ามีประสิทธิภาพต่ำ เนื่องจากทำให้ระดับ Statistical Multiplexing ลดลง

- การจัดบริการ Bandwidth ยังไม่สามารถแก้ปัญหา Bandwidth ระหว่างผู้ให้บริการได้
- มีกลไกการจัดลำดับความสำคัญข้อมูลแบบง่ายๆ (Simple Priority Mechanism) แต่ไม่สามารถควบคุมจำนวนข้อมูลแบบ Real-Time ใน Network ได้ จึงเกิดความคับคั่งและล่าช้า
- อัตราการรับส่งข้อมูลแบบ Real-Time มีความจำกัด และเกิดการ Delay โดยเฉพาะเมื่อไม่มี Admission Control

นิยามศัพท์เฉพาะของ IntServ

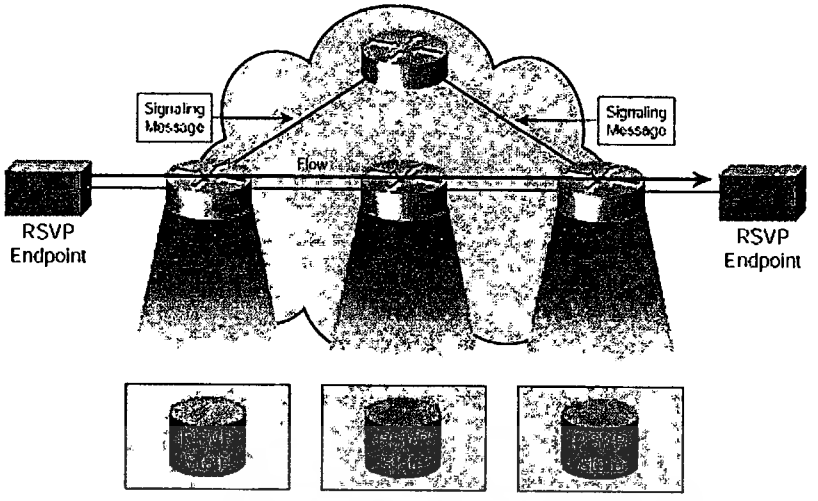
ในส่วนนี้จะได้กล่าวถึง IntServ เบื้องต้นก่อน โดยจะกล่าวถึงรายละเอียดในหัวข้อต่อไป

- Flow หมายถึง การไหลของ Packets ใน Network ผ่าน Node ต่างๆ ที่กำหนด QoS เหมือนกัน โดยอาจส่ง Packet ทีละชุดหรือจำนวนมากๆ พร้อมกันก็ได้
- Traffic Specification (TSpec) การกำหนดรูปแบบจราจรของข้อมูลในช่วงเวลาหนึ่ง
- Service Request Specification (RSpec) การกำหนดคุณสมบัติ QoS ของ Flow
- Flow Specification (Flowspec) เป็นส่วนผสมกันระหว่าง TSpec กับ RSpec โดย Node สามารถใช้ Flowspec เป็น Input สำหรับการตัดสินใจของ Admission-Control ได้

รูปแบบของสถาปัตยกรรม

หลักการสำคัญของสถาปัตยกรรม IntServ คือการร้องขอเพื่อสงวน Resource เพื่อควบคุมและบริหาร Resource ได้แน่นอนยิ่งขึ้น กล่าวคือ IntServ Nodes จะหลีกเลี่ยงการร้องขอที่ไม่ได้รับอนุญาตหรือการร้องขอที่มีผลต่อการสงวนอื่นที่ให้บริการอยู่ ผู้ใช้ต่างชนิดกันมีสิทธิในสงวน Resource แตกต่างกัน นอกจากนี้ยังควบคุมโหลดในเครือข่ายให้มีคุณภาพและปริมาณตามที่กำหนด โดยมี IntServ เป็นผู้กำหนด QoS ที่เหมาะสมให้กับ Application นั้นๆ

สถาปัตยกรรมกำหนด Flow เป็นหน่วยให้บริการพื้นฐาน โดยแบ่งชุด Packet ที่ใช้ QoS เดียวกันออกเป็นส่วนๆ แล้วส่งไปหลายๆ ทิศทางพร้อมกัน สถาปัตยกรรม IntServ กำหนดให้ Node Network ใช้สถานะ Per-Flow State ในการทำงาน สถานะดังกล่าวได้มาจากค่า Flow Granularity และการสงวน Resource ของ Admission Control รูปแบบดังกล่าวแตกต่างจากสถาปัตยกรรม IP เดิมซึ่งกำหนดให้ใช้สถานะ Per-Flow State ที่ Node ปลายทางระบบเท่านั้น นอกจากนี้ IntServ ยังออกแบบให้ใช้ Signaling Protocol ในการกำหนดและปรับปรุง State เพื่อรักษาความคงที่ (Robustness) ของ IP Protocol อีกด้วย รูปที่ 3.29 แสดงตัวอย่าง IntServ Network อย่างง่าย



รูปที่ 3.29 IntServ Network

Service Model

สถาปัตยกรรม IntServ กำหนดให้ Framework ของ Service Model มีลักษณะยืดหยุ่น โดยมีส่วนประกอบของ Service ที่ Receiver ร้องขอตามลักษณะของ Network การร้องขอ Service อาจประกอบด้วย TSpec และ RSpec อย่างใดอย่างหนึ่งหรือสองอย่างรวมกัน เมื่อ Service ได้รับการตอบรับ Network Nodes จะทำการรับประกัน Sevice ตามที่ TSpec กำหนดอย่างต่อเนื่อง ทั้งนี้กรณีที่ร้องขอทั้ง TSpec และ RSpec จะรวมทั้งสองอย่างเข้าด้วยกันและเรียกใหม่ว่า Flowspec

Service Model ใช้ TSpec กำหนดการทำงาน โดยแบ่งพารามิเตอร์การจราจรออกเป็น 4 พารามิเตอร์ ดังนี้

- A Token Bucket (r, b) ประกอบด้วยอัตรา Token (r) และขนาด Token Bucket (b)
- A Peak Rate (p) คืออัตราการไหลสูงสุดของ Packet ข้อมูล
- A Minimum Policed Unit (m) คือขนาดต่ำสุดของ Packet ทั้งนี้ Network Node จะกำหนดขนาด Packet ไม่ต่ำกว่าค่า Minimum Policed Unit ซึ่งกำหนดไว้เท่ากับ m การกำหนดค่า m ส่งผลให้การประเมินขนาด Bandwidth ที่จะใช้ Flow ข้อมูลทำได้สะดวกขึ้น (ประกอบด้วย Layer 2 Header Overhead)
- A Maximum Packet Size (M) Node จะพิจารณา Packet ที่มีขนาดใหญ่กว่า M ว่าไม่ตรงกับที่ Traffic Specification ระบุ และจะไม่รับ Packet นั้น

สถาปัตยกรรม IntServ ประกอบด้วย Service 2 อย่างหลักๆ คือ Guaranteed Service (GS) และ Controlled Load Service (CLS) ทั้งสองอย่างกำหนดขึ้นเพื่อรองรับ Best-Effort Service และถือเป็นส่วนหนึ่งของการกำหนด IP Protocol IntServ ไม่มีผลกระทบต่อการทำงานของ Best-Effort Service นอกจากนี้ IntServ Service Model ยังไม่ดำเนินการใดๆ ที่ส่งผลกระทบต่อการจัดการจราจรที่ใช้ Sevice นั้นอีกด้วย ในสองส่วนต่อไปนี้จะได้อธิบายถึง GS และ CLS Service โดยเฉพาะการกำหนด Flow และการลักษณะทำงานแบบ End-To-End

ตารางที่ 3.4 TSpec parameter

Parameter	**Description**
r	Token Rate
b	Token Bucket Size
p	Peak Rate
m	Minimum Policed Unit
M	Maximum Packet Size

Guaranteed Service (GS)

GS ทำหน้าที่รับประกันการ Delay และขนาด Bandwidth ให้กับ Flows โดยกำหนด Delay สูงสุดของเส้นทางให้สอดคล้องกับ Flowspec แบบจุดต่อจุด และควบคุมไม่ให้ข้อมูลสูญหายขณะที่ Network ล่มหรือการ routing ล้มเหลว ทั้งนี้ service จะไม่กำหนด Flowpath ด้วยค่า Delay ตายตัว (เช่น Propagation Delay หรือ Serialization Delay) โดย Flow จะยอมรับ Guaranteed Service ก็ต่อเมื่อทุก Node ตลอดเส้นทางสนับสนุนต่อ Service เท่านั้น GS รับประกันเฉพาะ Maximum Delay ไม่ได้รับประกัน Average Delay หรือ Minimum Delay ดังนั้น Service นี้จึงไม่ครอบคลุมถึงการรับประกัน Jitter ด้วย

เมื่อมีการร้องขอ GS เกิดขึ้น ตัว Receiver จะกำหนด TSpec และ RSpec ขึ้น โดย RSpec ประกอบด้วย Service Rate (R) และ Time Slack (S) Service Rate เป็นการสั่งให้ Network Node กำหนดค่าโดยประมาณเพื่อจัดเส้นทาง (Line) ในอัตรา (Rate) ที่เหมาะสมสำหรับการ Flow นั้น ค่าประมาณดังกล่าวมีการเผื่อสำหรับ Error ที่อาจเกิดขึ้นไว้แล้ว โดย Application สามารถนำค่าดังกล่าวไปคำนวณหา Delay สูงสุดระหว่างจุดต่อจุดที่ Flow นั้นจะต้องผ่านได้ ส่วน Time Slack ใน RSpec เป็นการระบุ Delay จุดต่อจุดเพิ่มเติม เผื่อไว้ในกรณีที่ Node เปลี่ยนแปลงการกำหนดค่า Flow Resource หรือมีค่า Delay มากเกินกว่าจะยอมรับได้ เพื่อให้ Application สามารถปรับค่า Flowspec ใหม่ได้ทันท่วงที ตาราง 3.5 แสดงค่าพารามิเตอร์ RSpec

Control Load Service (CLS)

CLS เป็นการรับประกันคุณสมบัติ Best-Effort Service ในภาวะ Unload ของข้อมูล โดย Network Node จะยอมรับคุณสมบัตินี้เมื่อเกิดความแออัดของ Packet เพื่อรับประกันข้อมูล Application ว่า Network จะส่ง Packet ส่วนใหญ่ไปถึงปลายทางได้ โดยไม่ล่าช้ากว่า Minimum Delay ของแต่ละ Packet ส่วนข้อมูลที่ล่าช้าเกินไปหรือสูญหายจะถูกปฏิเสธไป Service นี้สนับสนุนการทำงานของ Application ภายใต้ Bet-Effort Service โดยมีความไวต่อสภาพแออัด

ของข้อมูลมาก ทั้งนี้ Application ไม่จำเป็นต้องใช้ RSpec ในการร้องขอ CLS เพราะสามารถใช้ TSpec อย่างเดียวได้

ตารางที่ 3.5 RSpec Parameter

Parameter	**Description**
R	Service Rate
s	Time Slack

การใช้ RSVP ใน IntServ

IntServ ใช้ RSVP เป็น Reservation Setup Protocol โดยมีรูปแบบของสถาปัตยกรรมเริ่มจาก การที่ Application แจ้งการร้องขอ QoS ของแต่ละ Flow ไปยัง Network การร้องขอดังกล่าวจะถูกนำไปใช้ในการจอง Resource และกำหนดการควบคุมของ Admission ที่ RSVP ทำงานอยู่ ทั้งนี้ RSVP ถูกเรียกใช้บ่อยแต่อาจไม่เกี่ยวข้องกับ IntServ เสมอไป กล่าวคือ RSVP กับ IntServ ใช้ Common History ร่วมกันแต่เป็นอิสระต่อกัน IntServ สามารถใช้ RSVP เป็น Protocol สำหรับส่งสัญญาณภายนอก แต่ก็อาจใช้สัญญาณอื่นๆ ได้เช่นกัน

3.4.2 Differentiated Services

กลุ่มงาน DiffServ เป็นผู้กำหนดสถาปัตยกรรมนี้ให้ IETF ในปี 1998 DiffServ ออกแบบด้วยสถาปัตยกรรมง่ายๆ และมี QoS แบบหยายๆ ใช้ได้กับทั้ง IPv4 และ IPv6 โดยไม่จำเป็นต้องกำหนด MicroFlow หรือกลไกการส่งสัญญาณอย่างชัดเจน (แตกต่างจาก IntServ) ทีมผู้ออกแบบได้กำหนด 12 RFCs ขึ้น โดย 5 ส่วนของ 12 RFCs ดังกล่าวเป็น Standards Tract ส่วนที่เหลือเป็น Informational

หมายเหตุ : IP Traffic Stream ใช้ Source Address, Destination Address, Protocol, Source Part และ Destination Part ในการกำหนด Micro Flow

นิยามศัพท์เฉพาะสำหรับ DiffServ

สถาปัตยกรรม DiffServ มีการบัญญัติศัพท์ใหม่หลายคำ ซึ่งส่วนนี้จะอธิบายความหมายเฉพาะสั้นๆ หากต้องการความหมายโดยละเอียดสามารถศึกษาได้จาก RFC 2475 และ 3260

- Domain หมายถึง Network ที่ทำงานด้วย DiffServ (ปรกติจะมี Administrative Control เหมือนกัน)
- Region กลุ่มของ DiffServ Domains ที่อยู่ติดกัน
- Egress Node Node ทางออก หรือ Node สุดท้ายที่ Packet จะต้องผ่านก่อนออกจาก DiffServ Domain ไป
- Ingress Node คือ Node ทางเข้า, Node แรกที่ Packet ต้องผ่านเมื่อเข้าสู่ DiffServ Domain

- Interior Node Node ระหว่างกลาง หรือ Node ใน DiffServ Domain ที่ไม่ใช่ทั้ง Egress Node และ Ingress Node
- DiffServ Field บริเวณส่วนหัวของ Packet ที่บรรจุค่า DiffServ Marking ไว้ ค่าในส่วนนี้จะต้องตรงกับ 6 bit แรกสุดของ byte ที่สองของ IP Header
- Differentiated Services Code Point (DSCP) ค่าเฉพาะที่กำหนดใน DiffServ Field
- Behavior Aggregate (BA) กลุ่ม Packets ที่มี DSCP เหมือนกันและไหลอยู่ระหว่าง DiffServ Node
- Ordered Aggregate (OA) ชุดของ BA ซึ่ง DiffServ Node ต้องรับประกันมิให้เกิดการ Reorder Packets
- BA Classifier การจำแนกประเภท Packets โดยพิจารณาจาก DSCP
- Multifield (MF) Classifier การจำแนก Packets โดยพิจารณาจากหลายๆ Field ใน Packet Header (ได้แก่ Source Address, Destination Address, Protocol และ Protocol Port)
- Per-Hop Behavior (PHB) การทำ Forwarding หรือให้บริการแก่ BA ที่ Node รับไว้
- Per-Hop Behavior Group PHB ตั้งแต่ 1 ชุดขึ้นไปที่ทำงานพร้อมกัน หรือหมายถึงชุดการ Forwarding ที่เกี่ยวข้อง
- PHB Scheduling Class (PSC) ชุดของ PHBs ที่ DiffServ Node จะต้องรับประกันไม่ให้เกิดการ Reorder Packets
- Traffic Profile รายละเอียดของรูปแบบการจราจรในช่วงระยะเวลาหนึ่ง ตามปกติจะเป็นเทอมของ Token Bucket (Rate และ Bufirst)
- Marking การกำหนด DSCP ใน Packet
- Metering การตรวจรายละเอียดของ Traffic Profile ในช่วงระยะเวลาหนึ่ง
- Policing การทิ้งบาง Packet ออกไป เพื่อให้สอดคล้องกับ Traffic Profile
- Shaping การ Buffering Packets เพื่อให้สอดคล้องกับ Traffic Profile
- Service Level Agreement (SLA) พารามิเตอร์แสดงรายละเอียดของสัญญาการให้บริการ (Service Contract) ระหว่าง DiffServ Domain กับ Domain ของผู้ใช้
- Traffic-Conditioning Specification การระบุพารามิเตอร์ที่ใช้กำหนดการให้บริการ
- Traffic Conditioning กระบวนการกำหนดสภาพของ Traffic ผ่านฟังก์ชันควบคุมต่างๆ เช่น การ Marking, Metering , Policing และ Shaping

รูปแบบของสถาปัตยกรรม

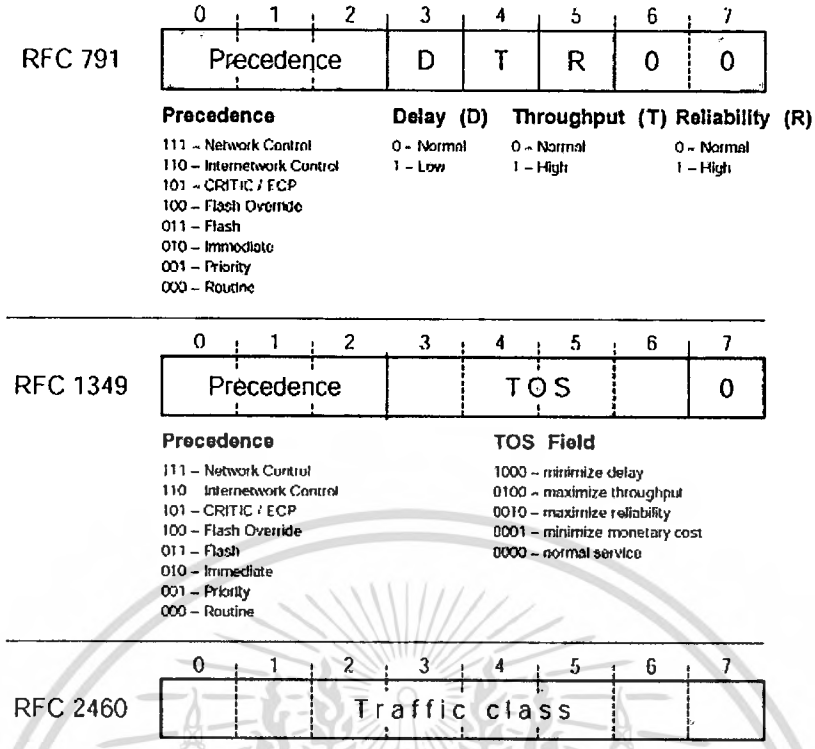
ในสถาปัตยกรรม DiffServ มีการแยก Traffic ออกเป็น Class ย่อยๆ และให้แต่ละ Class ใช้ Service แตกต่างกัน โดยจะระบุ Class ของ Traffic ไว้ที่ส่วนหัวของ Packet เพื่อให้ Network Node สามารถตรวจสอบการ Marking เพื่อระบุ Class ของ Packet และกำหนด Network

Resources ตามที่ Service Policy กำหนดไว้ได้ รายละเอียดของแต่ละ Service จะระบุคุณสมบัติทั้งค่า Latency, Jitter และการสูญหายของ Packet โดยเป็นไปในทิศทางเดียวกัน ทั้งนี้ในสถาปัตยกรรม DiffServ ตัว Packet จะถูกแยกย่อยเป็น Micro Flow แล้ว แบ่งไปตาม Node ตามขีดความสามารถที่กำหนด

การแบ่งระดับ Service ตาม Class และการ Marking ดังกล่าวนี้ไม่ใช่แนวคิดใหม่ ในชีวิตจริง เช่น บริการบินพาณิชย์ก็มีการแบ่ง Service ไว้หลายระดับแบบนี้เหมือนกัน เช่น ระหว่างการ Check-in ผู้โดยสารจะต้องให้ข้อมูลบางอย่างแก่เจ้าหน้าที่ (เทียบได้กับเกณฑ์การแบ่งระดับ) เจ้าหน้าที่จะใช้ข้อมูลดังกล่าวแบ่งระดับของผู้โดยสาร (เช่น ชั้นหนึ่ง, ชั้นธุรกิจ หรือชั้นนักท่องเที่ยว) แล้วออกใบ Boarding Pass ซึ่งระบุ Class ของผู้โดยสารให้ (เทียบได้กับการ Marking) จากนั้นผู้โดยสารก็จะได้รับบริการตามระดับที่ระบุไว้ การแบ่งชั้นผู้โดยสารออกเป็นระดับๆ ส่งผลให้ผู้ให้บริการไม่จำเป็นต้องเตรียมบริการทุกอย่างไว้สำหรับทุกคน การทำงานจึงสะดวกขึ้น

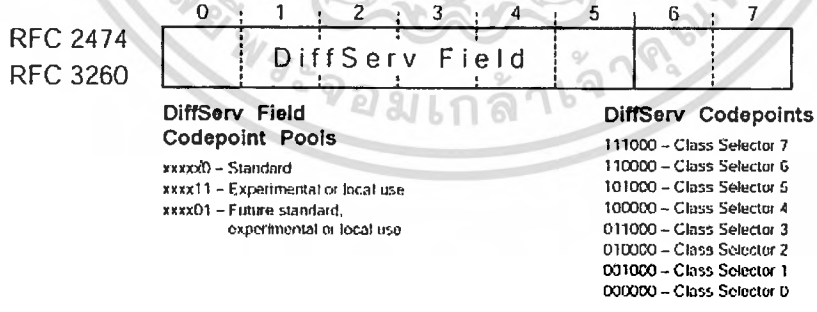
Differentiated Service Code Point

ในการกำหนด IP Protocol จะสงวน bit ส่วนหัวไว้สำหรับ QoS โดยเฉพาะ โดยใน IP เวอร์ชั่น 4 (RFC 791) กำหนดให้ Header Octet ที่สองเป็น TOS Octet ขณะที่ IP เวอร์ชั่น 6 (RFC 2460) กำหนดให้ Header Octet ที่สองเป็น Traffic Class Octet แต่ไม่ได้กำหนดโครงสร้างไว้แน่นอน สำหรับ TOS Octet ใน IP เวอร์ชั่น 4 มี 3 บิทแรกเป็นตัวกำหนดลำดับความสำคัญของ Packet (Precedence) อีก 3 บิทถัดมาใน TOS Octet กำหนดค่า Delay, Throughput และ Reliability ที่ต้องการ ส่วนอีก 2 บิทสุดท้ายไม่ระบุอะไรหรือเซตค่าไว้เท่ากับ 0 ใน RFC 1349 มีการเปลี่ยนแปลง TOS Octet เล็กน้อย โดยกำหนด Field ใหม่ประกอบด้วยบิต Cost bit + Existing Delay, Throughput และ Reliability ดังในรูปที่ 3.30



รูปที่ 3.30 TOS Octet IP v4 และ IPv6

สถาปัตยกรรม DiffServ มีการกำหนด IPv4 TOS Octet และ IPv6 Traffic Class Octet ใหม่ โดยกำหนด DiffServ Field ดัง RFC 2474 และ RFC 3260 (รูปที่ 3.31) ประกอบด้วย 6 บิท สำคัญ ของ IPv4 TOS และ IPv6 Traffic Class Octets รวมกัน รายละเอียดของ DiffServ Field จะแสดงอยู่ใน DSCP โดย DiffServ Node จะให้บริการ Packet ตามรหัสที่กำหนดอยู่ใน DSCP ทั้งนี้กลุ่ม Packet ที่ใช้ DSCP ร่วมกันและไหลอยู่ระหว่าง Link จะเรียกว่า BA โดยแต่ละ Class ของ Traffic อาจมี BA เดียวหรือมากกว่าก็ได้



รูปที่ 3.31 DiffServ Field,Code Point และ Class Selector Code Point

สถาปัตยกรรมกำหนดให้ DiffServ Field ประกอบด้วย 3 Code Point Pool 2 Pool แรก เป็น 32 ใน 64 ที่เป็นไปได้ สงวนไว้สำหรับ Local use อีก 1 กลุ่มที่เหลือเป็น 21 ใน 32 ค่าที่ DiffServ Specification สนับสนุนให้ใช้

สำหรับ "Per-Hop Behaviors" ในบทนี้จะกล่าวถึงเฉพาะการกำหนดค่า และรายละเอียดการให้บริการ โดยเฉพาะ 8 Code Point (Class Selector) ซึ่งแสดงความเข้ากันได้ของ Field หน้ากับหลังใน TOS Octet ทั้งนี้ DiffServ Field ไม่ได้กำหนดการเข้ากันของ TOS Field หน้ากับหลังใน TOS Octet ไว้ล่วงหน้า (Delay, Throughput และ Reliability bits) รูปที่ 3.31 แสดงโครงสร้างของ DiffServ Field ใหม่ รวมทั้ง Code Point Pool และ Class Selector Code Points

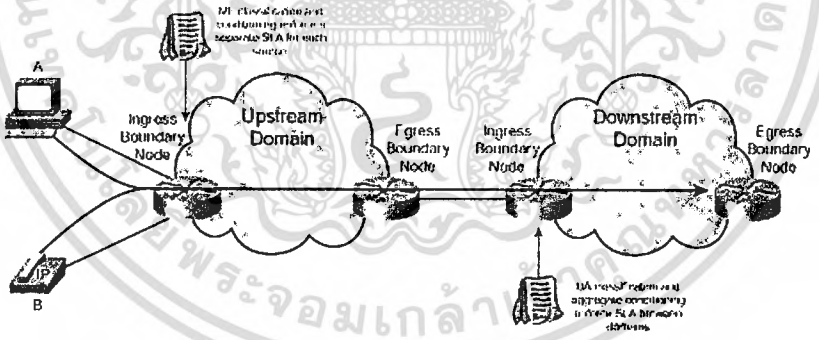
Nodes, Domains และ Regions

สถาปัตยกรรม DiffServ กำหนดลำดับการส่งข้อมูลเรียงจากอุปกรณ์เดี่ยว ไปยัง Network ไปยังกลุ่ม Network กลุ่มของ Node ที่ใช้ DiffServ จะเรียกว่า Domain โดยแต่ละ Node ที่อยู่ใน Domain เดียวกันจะมีรายละเอียดและ Policy การให้บริการแบบเดียวกัน ตามปกติ Domain หนึ่งๆ จะอยู่ภายใต้การควบคุมของ 1 Administrative Control โดยกำหนดชุดของ Domain ที่อยู่ติดกันเป็น DiffServ Region Domain ที่อยู่ภายใต้ Region เดียวกันต้องสามารถจัดการจราจรของข้อมูลภายใต้สถาปัตยกรรม DiffServ ได้ แต่อาจระบุการให้บริการ Policy และทำ Packing Marking แตกต่างกัน ซึ่งกรณีดังกล่าวจะต้องมี Peering Agreement เพื่อกำหนดว่าจะจัดการจราจรข้ามระหว่าง Domain อย่างไร

ใน DiffServ Domain ประกอบด้วย Node 2 ชนิด หลักๆ คือ Boundary Node กับ Interior Node Boundary Node ทำหน้าที่เชื่อมต่อกับ Domain ภายนอก โดยกำหนดคุณสมบัติการจราจรด้วยการจำแนก Packet, Mark ตามข้อตกลง (Agreement) ที่ระบุไว้ล่วงหน้า ซึ่งใน DiffServ จะดำเนินการด้วยการจำแนกและปรับสภาพการจราจร (Traffic Classification and Conditioning) โดยทั้ง Boundary Node และ Interior Node จะระบุ Local Service Policies ตาม Packet Marking และกำหนดระดับบริการแก่ BA ตามที่ Local Policies ระบุ กระบวนการดังกล่าวในสถาปัตยกรรม DiffServ เรียกว่า PHBs ซึ่งจะได้กล่าวโดยละเอียดในหัวข้อต่อไป โดยเฉพาะกรณีที่มีบาง Node ไม่สนับสนุนต่อ DiffServ ซึ่ง Node เหล่านี้จะกระทบถึงการบริการเพียงใด ขึ้นอยู่กับจำนวนและตำแหน่ง Node ใน Domain รูปที่ 3.32 แสดง DiffServ Region ที่ประกอบด้วย 2 Domains

สำหรับการทำ Condition Packet บน Boundary Node นั้นเกิดขึ้นหลังจากทำการ Classification เสร็จแล้ว โดยจะดำเนินการตามที่ระบุไว้ใน SLA เช่นกัน การทำ Traffic Conditioning เป็นการรวมกลไกการ Metering, Marking, Policing หรือ Shaping ตั้งแต่ 1 อย่างขึ้นไปเข้าไว้ด้วยกัน โดยใช้ผลลัพธ์จากการ Classification ก่อนหน้าเป็น Input การทำ Conditioning ในแต่ละ Class อาจไม่เหมือนกัน เพราะปกติ SLA จะกำหนดจำนวน Traffic ที่ Boundary Node สามารถรับได้ตาม Class โดยมี Boundary Node จะเป็นผู้วัดค่า Traffic และทำการ Making, Dropping หรือตัดสินการ Buffering บน Packet การทำ Conditioning จำเป็นต้องมีการเซตค่า DSCP แต่ละ Packet ให้เหมาะสม เพื่อให้ Node ข้างหน้าสามารถใช้ DSCP ดังกล่าวในการทำ Classification Packet ต่อไปได้อย่างรวดเร็ว นอกจากนี้ยังมีการทำ "Traffic Policing" และ "Traffic Shaping" ซึ่งเป็นกลไกการ Conditioning ในระดับรายละเอียดอีกด้วย

การทำ Traffic Classification and Conditioning อาจเกิดขึ้นบนจุดใดของเส้นทาง Packet ก็ได้ แต่ตามปกติมักเกิดที่ Boundary Node ทั้งนี้ตัวสถาปัตยกรรมยังอนุญาตให้ทำการ Classification และ Condition ซ้อนบน Interior Node ได้ด้วยหากจำเป็น (เช่น ใน International Links) รูปที่ 3.33แสดงตัวอย่างซึ่ง Boundary Node ทางเข้าของ Domain ต้นทาง ใช้ MF Classifiers and Conditions แยก Traffic A และ B ออกจากกัน ในตัวอย่างนี้ Boundary Node ที่ทางเข้าของ Domain ปลายทางทำหน้าที่ในการ Condition การจราจรที่ได้รับ โดยใช้ค่าจาก Classification และ Marking ของ Domain ต้นทางดำเนินการ



รูปที่ 3.33 DiffServ Domains

3.4.3 Per-Hop Behaviors (PHB)

สถาปัตยกรรม DiffServ นิยาม PHB ว่า คือการ Forwarding Behavior ที่ Node กระทำต่อ Behavior Aggregate (BA) PHB จะแสดงถึงรายละเอียดค่า Latency, Jitter หรือการสูญหายของ Packet ที่ BA จะต้องได้รับเมื่อผ่านไปยัง DiffServ Node โดย PHB กลุ่มหนึ่งๆ อาจมี PHBs เกี่ยวข้องจำนวนมาก และอาจดำเนินการทำ PHB พร้อมกันตั้งแต่หนึ่งชุดหรือมากกว่าก็ได้

DiffServ Node จะทำการ Map Packet ให้กับ PHB ตามที่ DSCP ระบุ ในตารางที่ 1-3 แสดงการ Mapping ที่กำหนดในสถาปัตยกรรม อย่างไรก็ตามการ Mapping ดังกล่าวไม่จำเป็นต้องทำตาม DSCP เสมอไป แต่สามารถปรับเปลี่ยนได้ยกเว้น Class Selector เนื่องจาก

สถาปัตยกรรมกำหนดให้ Class Selector มีความสามารถในการ Backward โดยใช้ Precedence Filed ใน IPv4 TOS Octet ทั้งนี้ DiffServ Domain อาจไม่จำเป็นใช้ Recommend Mapping ดังตารางที่ 3.6 ก็ได้ หากมีการ Interfacing กับ DiffServ Domain อื่นๆ ที่ซับซ้อนกว่า

จำนวน PHB หรือกลุ่ม PHB จะมีมากน้อยแค่ไหนขึ้นอยู่กับ DiffServ Specification ได้แก่ การ Expedited Forwarding (EF), Assured Forwarding (AF1, AF2, AF3 และ AF4), Class Selector (CS) และ Default Node อาจดำเนินการกับกลุ่ม PHB หลายๆ ตัวพร้อมกัน โดยใช้กลไกการ Packet-Buffering และ Scheduling

ตารางที่ 3.6 Mapping ระหว่าง PHBs and DSCPs

PHB	DSCP (Decimal)	DSCP (Binary)
EF	46	101110
AF43	38	100110
AF42	36	100100
AF41	34	100010
AF33	30	011110
AF32	28	011100
AF31	26	011010
AF23	22	010110
AF22	20	010100
AF21	18	010010
AF13	14	001110
AF12	12	001100
AF11	10	001010
CS7	56	111000
CS6	48	110000
CS5	40	101000
CS4	32	100000
CS3	24	011000
CS2	16	010000
CS1	8	001000
Default	0	000000

Expedited Forwarding (EF)

EF PHB เป็นค่า Latency, Jitter และการสูญเสีย Packet ขั้นต่ำที่ DiffServ Node ต้องทำให้ได้ PHB ดังกล่าวเป็นเสมือนขอบจำกัดขั้นต่ำของการส่งข้อมูลแบบ Real-Time ผ่าน DiffServ Domain ซึ่ง DiffServ Node จะต้องรักษาการให้บริการ EF ให้มีอัตราเร็วกว่าอัตราที่ EF เข้าถึงอยู่เสมอโดยไม่ขึ้นกับจำนวน Non-EF-Traffic ความแตกต่างระหว่างอัตราการเข้าถึงกับอัตราการให้บริการดังกล่าวทำให้ Packet ทั้งสองไม่ปะทะกันหรือเกิดการคั่งค้าง ส่งผลให้ Latency มีค่าต่ำสุด ทั้งนี้ Latency ประเภทนี้เป็นสาเหตุหลักของการ Latency และ Jitter ของ Packet ระหว่าง

การปฏิบัติการของ Node การมีความแออัดของคิวต่ำ นอกจากจะทำให้ Latency และ Jitter ต่ำแล้ว ยังทำให้การสูญเสียของ Packet ต่ำลงอีกด้วยเนื่องจาก Packet Buffers ไม่เกิดการแน่นหรือล้น ทั้งนี้ที่จุด DiffServ Node ไม่ควรมีการ Reorder ของ Micro Flows

Assured Forwarding

AF จัดอยู่ในกลุ่มของ PHB มีหน้าที่กำหนดการรับประกัน Forwarding ที่ DiffServ Node จะต้องรองรับ (มี 4 กลุ่ม) กล่าวง่ายๆ คือ เป็นการกำหนดทางเลือก, วิธีการให้ DiffServ Node รับประกันการสูญเสียของ Packet นั่นเอง AF PHB ทั้ง 4 กลุ่มประกอบด้วย AF1, AF2, AF3 และ AF4 โดยในแต่ของกลุ่ม AF ยังแบ่งระดับการทิ้ง Packet (Drop-Precedence) ออกเป็น 3 ระดับ ทั้งนี้หาก Resource (Bandwidth, Buffers) เกิดการล้นหรือรับ Packet ไม่ทัน DiffServ Node จะทิ้ง Packet ที่มีระดับการ Drop ต่ำก่อน

ระดับการ Drop ทั้ง 12 ระดับของ AF PHB แสดงอยู่ดังตารางที่ 3.7 รายละเอียดการกำหนดเซต PHB เหล่านี้สามารถอ่านเพิ่มเติมได้จาก RFC 2597

AF PHB ทั้ง 4 กลุ่มทำงานเป็นอิสระต่อกัน โดยไม่มีผลต่อการ Forwarding Guarantee ของอีกกลุ่ม และไม่มีผลต่อค่า Latency หรือ Jitter ทั้งนี้การ Guarantee ของแต่ละกลุ่มจะขึ้นอยู่กับ Forwarding Resource ของ Node, จำนวนการจราจรที่เข้าถึง Node และผลสืบเนื่องที่จะเกิดจากการ Drop ของ Packet โดยพิจารณาจาก Bandwidth หรือพื้นที่ของ Buffer เป็นหลัก และกำหนดให้ Node ทำการ Forward ในอัตราสูงที่สุดเท่าที่จะทำได้เพื่อให้ระดับการ Drop Packet ต่ำ และไม่มีการ Reorder Packet ของ Micro Flow ที่ถูก Drop ไปแล้วอีก

ตารางที่ 3.7 ระดับการ Drop ของ AF PHBs

Drop Precedence	AF1	AF2	AF3	AF4
Low	AF11	AF21	AF31	AF41
Medium	AF12	AF22	AF32	AF42
High	AF13	AF23	AF33	AF43

Class Selectors

DiffServ กำหนด CS PHBs ขึ้นเพื่อให้สามารถ Backward โดยใช้ IP Precedence ใน IPv4 TOS Octet ได้ Class Selector ทำหน้าที่รักษา Relative Ordering ของ IP Precedence ให้คงที่ (ค่ามากกว่าหมายถึง Relative Order สูงกว่า) โดย Node จะต้องทำการ Forwarding ไปยัง CSs ด้วยอัตราสูงสุดเท่าที่จะทำได้ เพื่อป้องกันไม่ให้ Latency, Jitter หรือการสูญเสีย Packet มีค่าเกินกว่าที่กำหนด ตารางที่ 3.8 แสดงการ Mapping ระหว่าง CSs กับ IP Precedence

ตารางที่ 3.8 การ Mapping ระหว่าง CSs กับ IP Precedence

PHB	DSCP (Decimal)	DSCP (Binary)	Precedence Name	Precedence (Binary)	Precedence (Decimal)
CS7	56	111000	Network Control	111	7
CS6	48	110000	Internetwork Control	110	6
CS5	40	101000	Critic/ECP	101	5
CS4	32	100000	Flash Override	100	4
CS3	24	011000	Flash	011	3
CS2	16	010000	Immediate	010	2
CS1	8	001000	Priority	001	1
CS0	0	000000	Routine	000	0

Default PHB

DiffServ Domain กำหนด Default PHB ขึ้นเพื่อรองรับบริการ Best-Effort Service กล่าวคือ สถาปัตยกรรมกำหนด Default PHB ขึ้นเพื่อให้เป็นไปตาม Best-Effort Service ที่ RFC 1812 ระบุ ซึ่งหมายความว่า DiffServ Domain จะต้องทำการ Forward จำนวน Packet ให้มากที่สุดเท่าที่จะทำได้ โดยไม่เกิด Latency, Jitter และการสูญเสีย Packet มากเกินกว่าที่กำหนด ทั้งนี้ การทำ PHBs ก็อยู่ภายใต้ Best-Effort Service นี้เช่นกัน

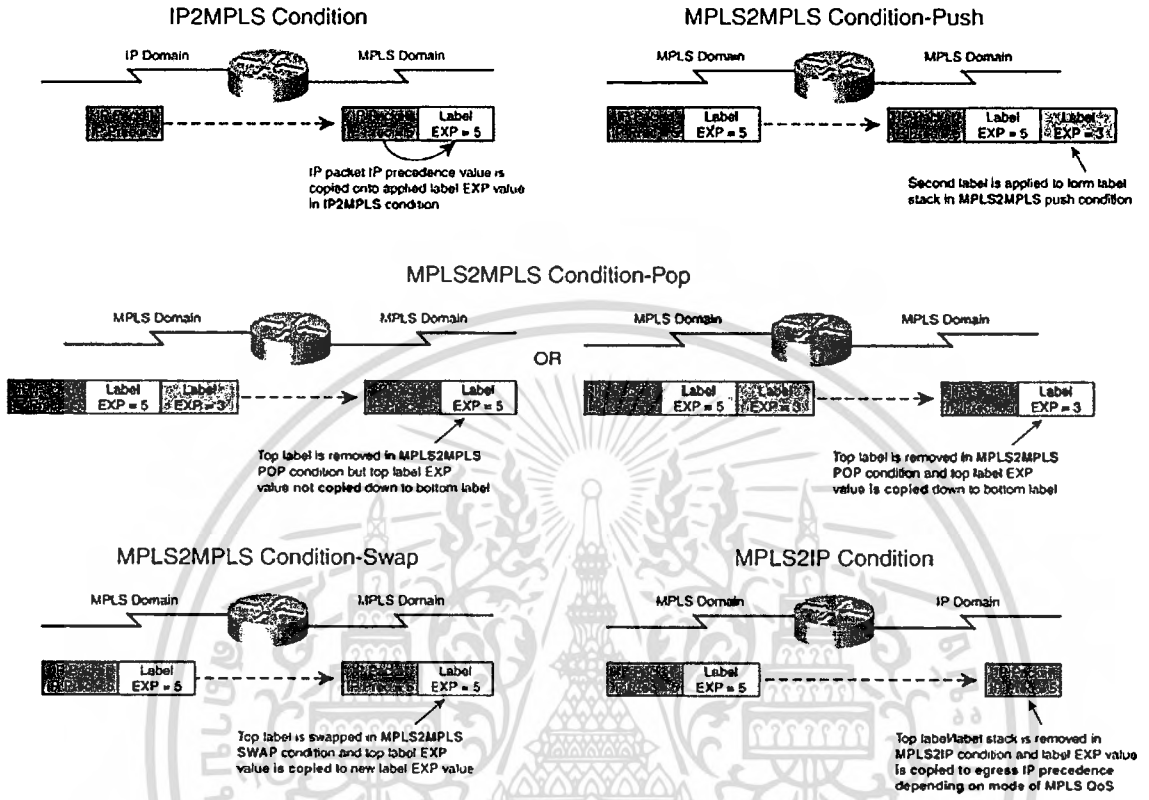
3.4.4 QoS บนระบบ MPLS

เมื่อนำ QoS มาใช้บนระบบ MPLS Edge LSR ระหว่าง IP กับ MPLS Domains จะทำการแปลง IP QoS Domain ให้เป็น MPLS QoS Domain และดำเนินการให้ IP Packet สามารถเข้าสู่ MPLS Domain ได้ (เช่น CE-PE) เรียกว่าการ IP2MPLS Condition ในบางกรณี Packet สามารถใช้ Label Stack โดยแลกเปลี่ยนค่ากับ EXP bit เรียกว่าการ MPLS2MPLS Condition และสุดท้ายอาจเปลี่ยน Label Packet ให้เป็น IP Packet ที่เหมาะสม (เช่น กรณีของ LSR ทางออกใน PE-CE Condition) เรียกว่าการ MPLS2IP Condition

IP2MPLS Condition เป็นการระบุ IP Packet ด้วย MPLS Label โดยหาก IP Packet ที่เข้ามา มี IP Precedence อยู่ด้วย มันจะถูก Copy ค่าไปยัง MPLS EXP Field ดังแสดงในรูปที่ 3.34 สำหรับ Routers เมื่อทำหน้าที่เป็น PE Router ทางเข้า จะทำการ Copy ค่า IP Precedence bit จาก L3 Header ไปใส่ไว้ใน MPLS EXP bit ตรงตำแหน่งของ Label

สำหรับ MPLS2MPLS Condition Packet ที่เข้ามามี Label ระบุอยู่แล้ว MPLS2MPLS Condition มีการดำเนินการ 3 แบบด้วยกัน คือ Push, Pop และ Swap ดังแสดงในรูปที่ 3.34 ในขั้นตอนการทำ MPLS2MPLS Push Condition ตัว Packet ที่เข้ามาจะถูกเปลี่ยน Label ใหม่ ส่วนการทำ MPLS2MPLS Pop Operation ตัว Packet ที่เข้ามาจะถูกถอดเอา Label บนสุดใน Label

Stack ออก และการทำ Swap Condition จะใส่ Label ใหม่ลงในส่วนบนสุดของ Label Stack ดังกล่าว ทั้งนี้ ระหว่างการ MPLS2MPLS Condition จะกำหนดค่า Label ใหม่โดย copy ค่า EXP ของ Label บนสุดไปใส่เป็น Label ใน Label stack



รูปที่ 3.34 QoS บนระบบ MPLS

สำหรับการทำ MPLS2IP Condition Packet ที่เข้ามามี Label ระบุอยู่แล้ว ขณะ Packet ที่ออกไปจะอยู่ในรูป IP Packet MPLS2IP Condition ดำเนินงานตามลักษณะ Tunnel Mode โดย EXP จะถูก Copy กลับไปในค่า IP Precedence ของ IP Packet ซึ่งจะได้อธิบายเกี่ยวกับ Tunnel Mode ในหัวข้อถัดไป การทำ Condition ทั้งสามแบบแสดงดังรูปที่ 3.34

3.4.5 Mode การดำเนินงานของ MPLS QoS

MPLS ที่ใช้ QoS มี Mode ดำเนินการอยู่หลายแบบ เรียก MPLS QoS Tunnel Modes โดยมี Mode หลักๆ ดังนี้

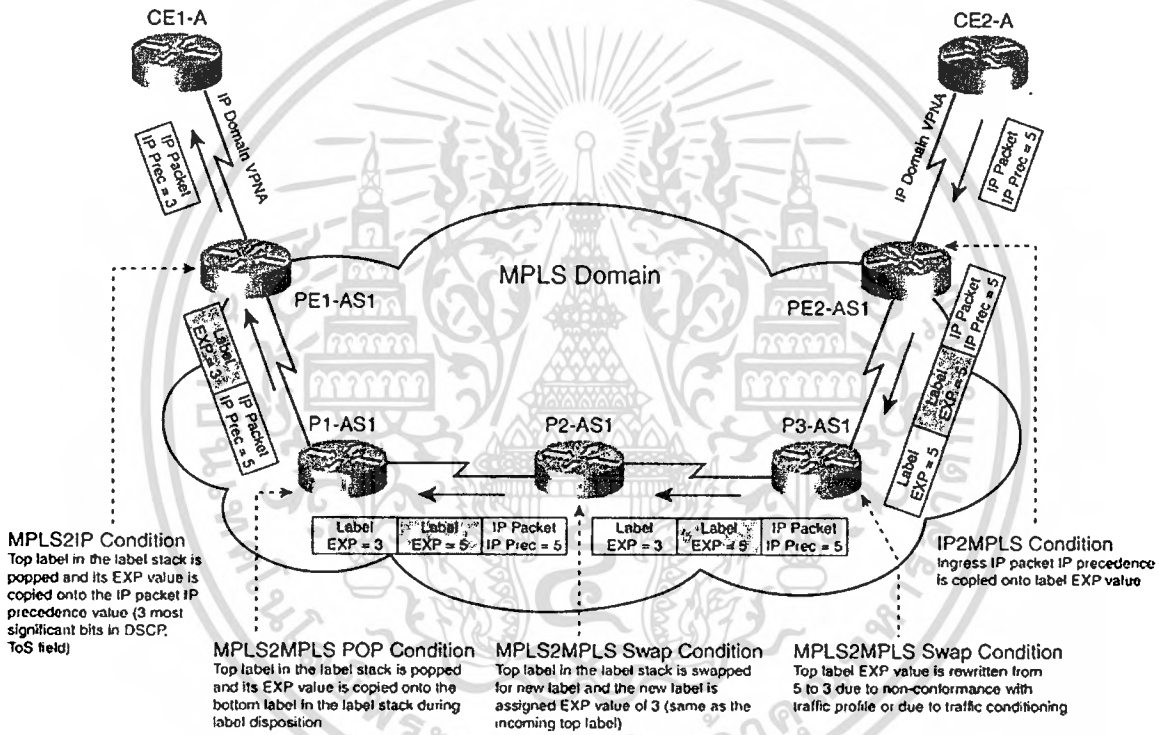
- Uniform Mode
- Pipe Mode
- Short Pipe Mode
- Long Pipe Mode

Uniform Mode

ใน Uniform Mode จะมีการเปลี่ยนแปลงค่าทั้งหมด ทั้ง Class ของ Packet, IP Precedence, DSCP และ MPLS EXP โดยถือว่า Packet ที่ผ่านโครงสร้างของ SP เป็น Packet ที่

ไหลไปสู่ปลายทาง สำหรับการทำ IP2MPLS Condition จะเริ่มด้วยการ Copy ค่า IP Precedence ของ IP Packet ไปใส่เป็นค่า Label EXP จากนั้นเมื่อทำการ MPLS2IP Condition ก็จะ Copy ค่า Label EXP บนสุด (Packet ที่ระบุ Label อาจมีมากกว่า 1 Label โดยเฉพาะใน Label Stack) มาใส่ในค่า IP Precedence ของ IP Packet

ปฏิบัติการที่สำคัญที่สุดของ Uniform Mode คือการทำ MPLS2MPLS Condition โดยเฉพาะการระบุ Label ให้กับ Packet ทั้งนี้การทำ MPLS2MPLS Condition Label จะถูกเปลี่ยนให้มีค่าเหมือนกับ Label EXP ส่วนบนสุด ทั้งนี้หากเป็นการ MPLS2MPLS POP ค่า MPLS EXP จะถูก Copy ลงมาข้างล่างของ Label Stack (ย้ายจาก Label บนสุด เป็น Label ล่างสุด) โดยมีการดำเนินการดังรูปที่ 3.35



รูปที่ 3.35 Uniform Tunnel Mode

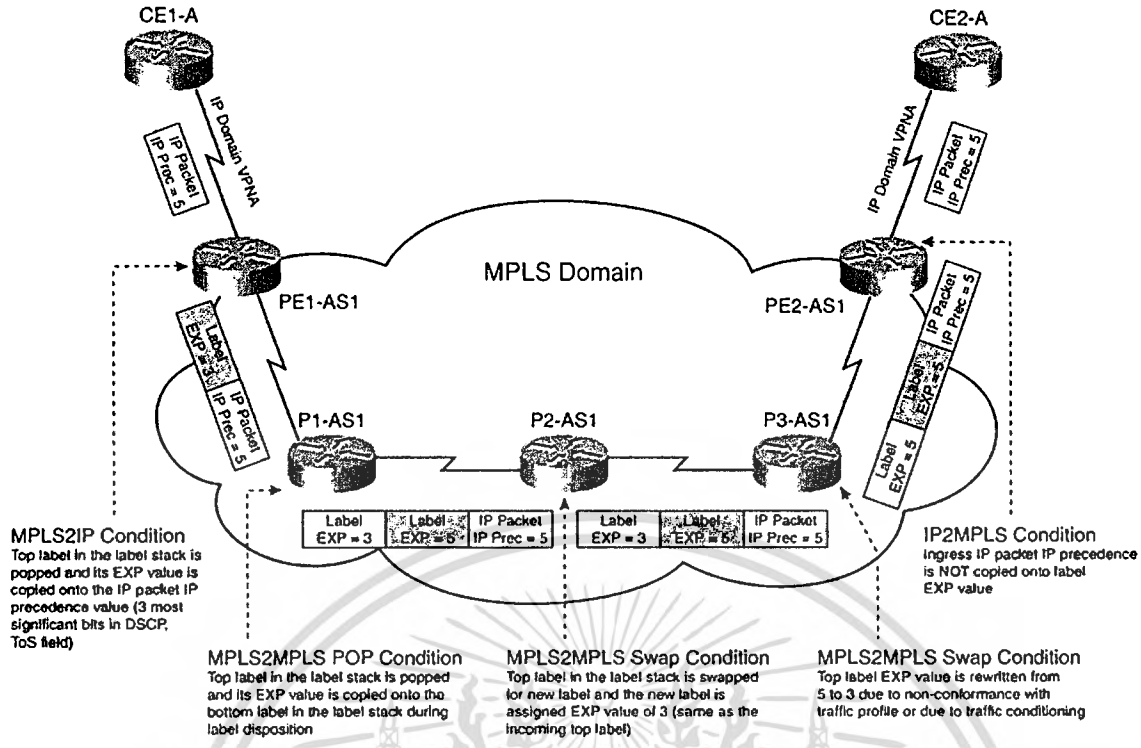
รูปที่ 3.35 แสดง MPLS VPN Network อย่างง่าย ในภาพมี 2 PE Router คือ PE1-AS1 และ PE2-AS1 เชื่อมต่อกับ CE Router โดยมี CE1-A และ CE2-A ให้บริการ MPLS VPN อยู่ โดยทั่วไป Uniform Mode ก็คือการบริหารการทำงานของ CE โดยมี SP ทำหน้าที่ควบคุม QoS จาก CE ถึง CE ผ่าน MPLS Domain เมื่ออยู่ใน Uniform Mode IP Packet ที่ CE2-A กำหนดให้กับ CE1-A จะถูกนำมาระบุ Label Stack (MPLS VPN Label Stack) โดยการ Marking ด้วย EXP เท่ากับ 5 แล้ว Map กับค่า IP Precedence ของ IP Packet ทางเข้าบน PE2-AS1 ในบางกรณี เช่น Traffic Engineering เข้ากันไม่ได้กับ Traffic Profile อาจมีการเขียนค่า EXP ใน LSP Path ขึ้น

ใหม่ก็ได้ ระหว่างการ Swapping P3-AS1 จะกำหนดค่า EXP ของ Label บนสุดใหม่จาก 5 เป็น 3 ส่วน P2-AS1 จะทำการ MPLS2MPLS Swap และ Forward ตัว Packet ที่ระบุ Label แล้วไปยัง P1-AS1 โดยยังคงค่า EXP 3 ไว้เหมือนเดิม จากนั้น P1-AS1 จะเอา Label บนสุดใน Label Stack ออก (การ Penultimate Hop Popping) ระหว่างนี้ค่า EXP ของ Label บนสุดจะถูก Copy ไปใส่เป็น Label ล่างสุด (การทำ MPLS2MPLS POP Condition ในโหมด Uniform) เมื่อ PE1-AS1 ได้รับ Packet ที่ระบุ Label แล้ว จะทำการเขียน IP Precedence ของ IP Packet ใหม่เป็น 3 แล้ว Map กับค่า EXP ของ Packet ทางเข้า ทั้งนี้ ขณะที่ Packet ซึ่งระบุ QoS แล้วเดินทางผ่าน MPLS Domain หรือ IP Domain ของ CE PE กับ CEs จะให้บริการ Service Domain ต่างกัน กระบวนการนี้จึงเหมือนการบริการการทำงานของ CE ดังที่กล่าวมาข้างต้น

Pipe Mode

การทำงานของ Pipe Mode คล้ายคลึงกับของ Uniform Mode ยกเว้นการทำ MPLS2IP Condition กล่าวคือ ค่า EXP ของ Label บนสุดจะไม่ถูก Copy เป็นค่า IP Precedence ของ IP Packet โหมดการทำงานนี้เหมาะกับกรณีที่การทำ QoS ของ SP ไม่จำเป็นต้องขึ้นกับ QoS Policy ของผู้ใช้บริการ ใน Pipe Mode จะไม่มีการเปลี่ยนแปลงค่า IP Precedence ของ IP Packet และไม่มีการ Copy ค่า IP Precedence ของ IP Packet ไปเป็นค่า MPLS EXP ขณะทำ IP2MPLS Condition อีกด้วย ทั้งนี้ หลังจากที่ Packet ผ่านการทำ MPLS2IP Condition แล้ว การดำเนินการกับ IP Packet PHB หรือ QoS บน Router จะขึ้นอยู่กับค่า EXP ของ Label บน LSR ทางออก โดยระหว่างการกำหนด Label LSR ทางออกจะเก็บค่า Copy ของ EXP ไว้ในหน่วยความจำ และกำหนดเป็นตัวแปร QoS-Group ซึ่งจะใช้ในการกำหนด PHB บน LSR ทางออกต่อไป

เมื่อเปรียบเทียบกับ Uniform Mode การดำเนินงานของ Pipe Mode ที่แสดงในรูปที่ 3.36 จะเห็นว่า PE1-AS1 ไม่ได้ Copy ค่า EXP ทางออกเป็นค่า IP Precedence ของ IP Packet ทางออก อย่างไรก็ตาม การเรียงคิวของ Packet ใน PE1-AS1 ยังคงขึ้นกับค่า EXP ทางเข้าซึ่งถูก Copy ในรูปตัวแปร QoS-Group เราจะใช้ Pipe Mode ก็ต่อเมื่อทำการ Forwarding ข้อมูลไปยัง CE Router แล้วพบว่า SP ควรดำเนินการ PHB ตาม QoS Policy ใน SP Core มากกว่าจะดำเนินการตาม QoS Policy ของลูกค้า ด้วยเหตุผลดังกล่าว QoS PHB ของ Packet ใน IP และ MPLS Domain จึงเป็นอิสระต่อกัน



รูปที่ 3.36 Pipe Mode

Short Pipe Mode

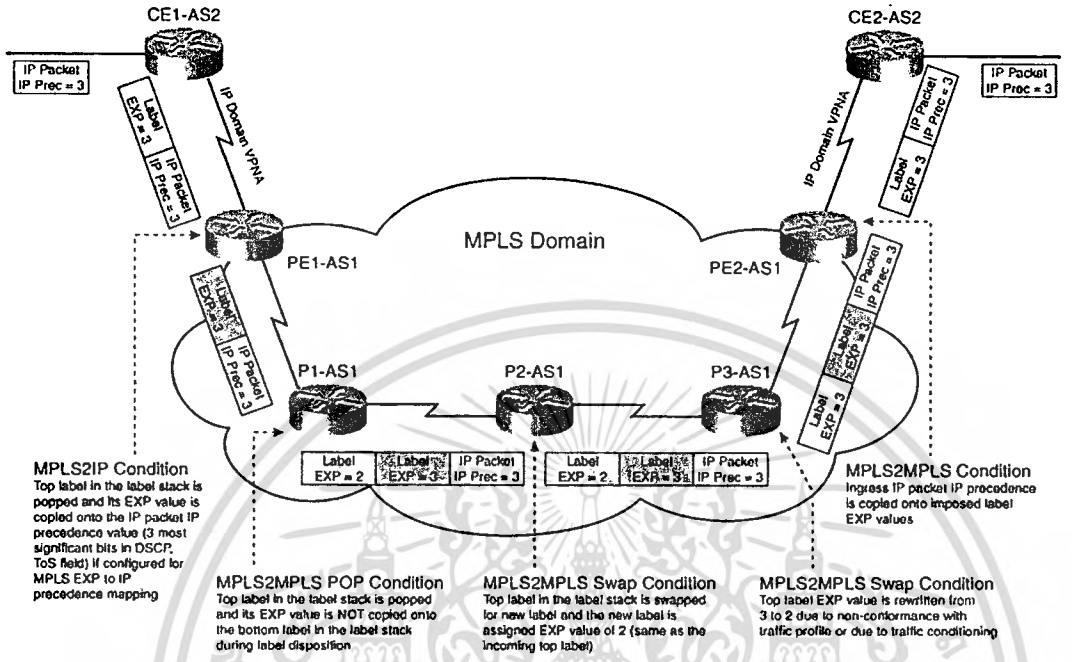
Short Pipe Mode มี MPLS2IP Condition ต่างจากโหมดอื่นๆ กล่าวคือ ไม่มีการทำ PHB กับค่า EXP ของ Packet ทางเข้า มีเฉพาะการทำ PHB กับค่า IP Precedence ของ IP Packet / DSCP เท่านั้น LSR ทางออกไม่มีการเก็บ Copy ค่า EXP ของ Packet ทางเข้าไว้ในรูปตัวแปร Qos-Group วิธีการนี้จะใช้ก็ต่อเมื่อ QoS ที่ใช้กับ Packet จำเป็นต้องเหมือนกับ QoS Policy ของลูกค้าเท่านั้น

Long Pipe Mode

Long Pipe Tunnel Mode ดัดแปลงมาจาก Pipe Tunnel Mode แต่แตกต่างกันที่จุด PE-CE Link กล่าวคือ Packet จะถูก Forward โดยการทำ Label Marking (หรือ Label Stack Marking) และถือว่า Link ดังกล่าวเป็นส่วนหนึ่งของ MPLS QoS Domain ทั้งนี้ CE Router ซึ่งรับ Traffic จาก MPLS Backbone สามารถกำหนด Policy ขาออกให้กับ VPN Site โดยใช้ MPLS Experimental bits หรือ DSCP bits เดิม โดย CE Router อาจทำการ Copy ค่า EXP ไปยัง IP Precedence ด้วยก็ได้หากต้องการ รูปแบบนี้ใช้ได้กับสถาปัตยกรรม Carrier Supporting Carrier (CSC) ดังแสดงในรูปที่ 3.37

จากรูปที่ 3.37 เมื่อ Packet ที่ CE2-AS2 รับถูกกำหนดให้ไปยัง CE1-AS2 Label จะดำเนินการกับค่าปลายทาง จากนั้นค่า EXP จะถูก Copy เป็นค่า IP Precedence ของ IP Packet ทางเข้า เมื่อ PE2-AS1 ได้รับ Packet ทางเข้าซึ่งระบุ Label แล้วเข้ามา ก็จะรวม Label Stack กับค่า EXP กลายเป็นค่า EXP ของ Label ทางเข้า หมายเหตุ : แม้ว่า P3-AS1 จะเขียนค่า EXP ของ Label ใหม่เป็น 2 ตามที่ P1-AS1 กำหนด (เดิมเท่ากับ 3) แต่ค่านี้จะไม่ถูก Copy กลับไปยัง Label

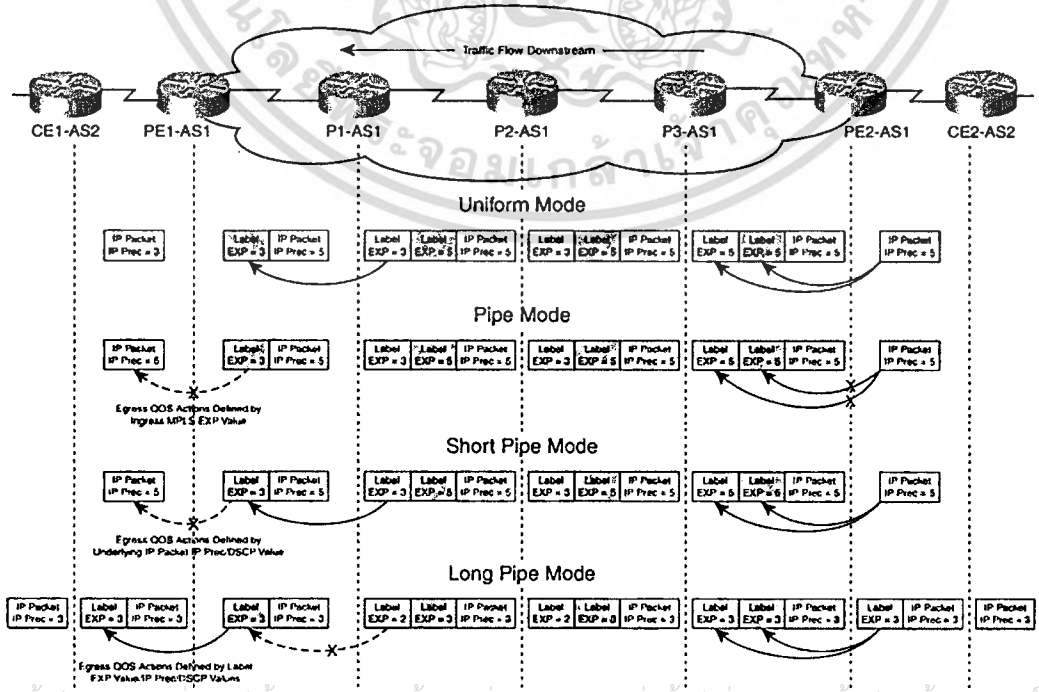
Stack PE1-AS1 จึงทำ MPLS2MPLS Label Swapping โดยใช้การ Mapping ของ EXP bit โดยตรง เมื่อ CE1-AS2 ได้รับ Packet ที่ผ่านการระบุ Label เข้ามา Router ก็จะใช้ค่า EXP ของ Packet ทางเข้า หรือค่า IP Precedence ของ IP Packet ในการทำ PHB ได้ต่อไป



รูปที่ 3.37 Long Pipe Tunnel

สรุป MPLS QoS Modes

สรุปการทำงานของ MPLS QoS Mode ทั้งหมด แสดงอยู่ในรูปที่ 3.38



รูปที่ 3.38 สรุปการทำงานของ Different MPLS QoS Modes

บทที่ 4
แบบจำลองการทดสอบเพื่อศึกษาประสิทธิภาพและการทำงานของระบบเครือข่าย Traditional IP และ MPLS/VPN

ในบทนี้จะกล่าวถึงพารามิเตอร์ที่ใช้ในการจำลองระบบและผลที่ได้จากการทดลองโดยแสดงให้เห็นประสิทธิภาพของระบบเมื่อนำวิธีการที่นำเสนอมาใช้เปรียบเทียบกับวิธีการแบบดั้งเดิมเมื่อส่งข้อมูลผ่านสภาวะแวดล้อมและโครงข่ายที่มี Bandwidth และ Topology แบบเดียวกัน

4.1 แบบจำลองที่ใช้ในการจำลองระบบ

แบบจำลองที่ใช้ในการจำลองเพื่อหาค่าประสิทธิภาพของระบบกำหนดรูปแบบการเชื่อมต่อ (Topology) แสดงดังรูปที่ 4.1 จากรูปเป็นการออกแบบระบบเครือข่ายเพื่อทำการทดสอบ โดยแบ่งการเชื่อมต่อให้เป็นลำดับชั้นเพื่อลดการทำงานของ CPU บนอุปกรณ์เครือข่ายในการจัดการกับแพ็กเกตที่ทำการบรอดคาสท์ ซึ่งสามารถแบ่งได้ 3 ลำดับชั้นดังนี้

1. ลำดับชั้นแกนกลาง (Core Layer)

ในลำดับชั้นนี้ถือได้ว่าเป็นชั้นที่สำคัญที่สุด หรือที่เรียกว่าแบ็กโบนก็ได้ เพราะอุปกรณ์ในชั้นนี้จะทำงานในระดับชั้นเลเยอร์ 3 ของ OSI Model หรือ Network Layer ซึ่งทำหน้าที่หาเส้นทางและส่งข้อมูลระหว่าง VPN ด้วยความเร็วสูงซึ่งในแบบจำลองการทดลองรูปที่ 4.1 Core Layer ประกอบด้วย Router R2 และ R3 ทำหน้าที่เป็น Provider Router เชื่อมต่อกันด้วย 10 Gigabit Ethernet Interface

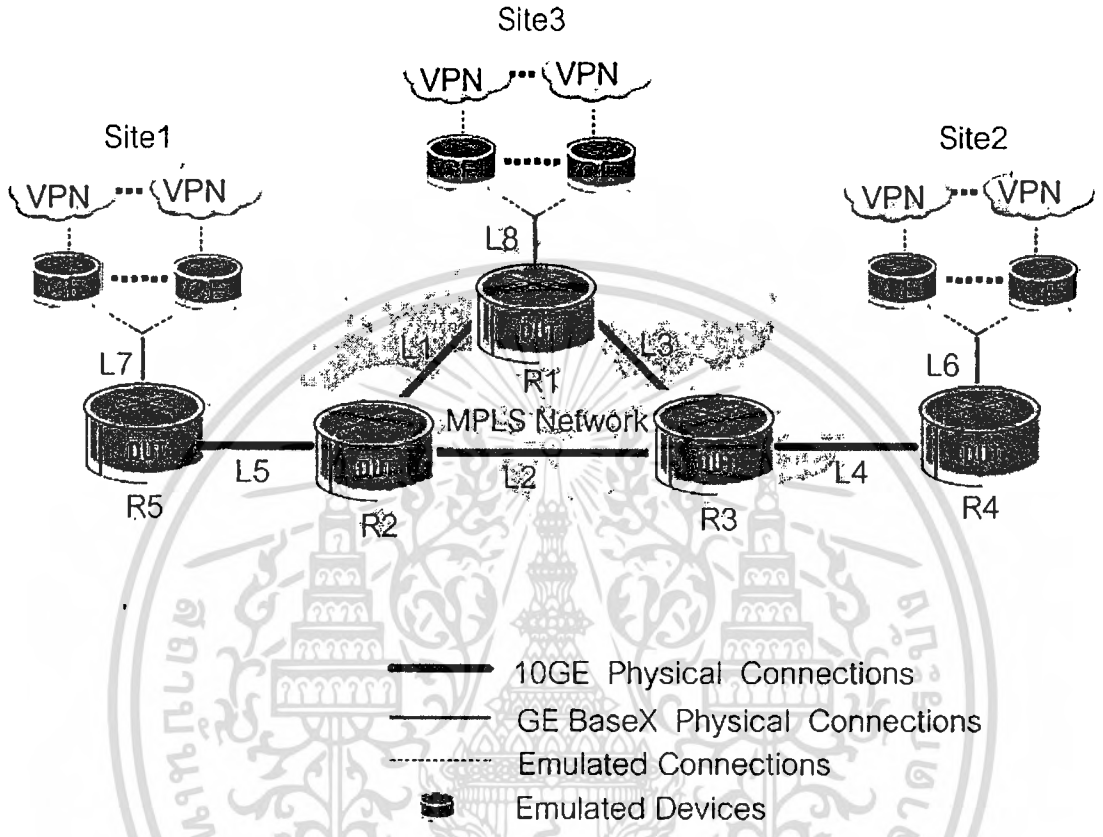
2. ลำดับชั้นการกระจาย (Distribution Layer)

ลำดับชั้นการกระจายเป็นตัวกลางในเครือข่ายเพื่อเชื่อมระหว่างลำดับชั้นแกนกลางและลำดับชั้นการเข้าถึง หน้าที่ของลำดับชั้นนี้คือ ควบคุมการทำงานของอุปกรณ์ในลำดับชั้นการเข้าถึง ซึ่งจะทำให้เกิดความปลอดภัยในเครือข่าย ควบคุมทราฟฟิกในเครือข่ายให้เป็นตามข้อกำหนดในลำดับชั้นแกนกลางและทำการวิเคราะห์บรอดคาสท์โดเมนส์ ในแบบจำลองการทดลองรูปที่ 4.1 Distribution Layer ประกอบด้วย Router R1,R4 และ R5 ทำหน้าที่เป็น Provider Edge Router เชื่อมต่อกับ Core Layer ด้วย 10 Gigabit Ethernet Interface และเชื่อมต่อกับ Access Layer ด้วย Gigabit Ethernet Interface

3. ลำดับชั้นการเข้าถึง (Access Layer)

ลำดับชั้นนี้เป็นการเชื่อมต่อผู้ใช้งานให้สามารถทำการติดต่อกับทรัพยากรที่มีอยู่บนเครือข่ายได้ ซึ่งในแบบจำลองการทดสอบรูปที่ 4.1 Access Layer จะใช้ Traffice Generator จำลอง

เป็นอุปกรณ์ CE (Customer Equipment) ทำการส่งข้อมูล Ethernet ขนาด Frame size 64 byte โดยเชื่อมต่อกับ Distribution Layer ด้วย Gigabit Ethernet Interface ซึ่ง CE แต่ละ Site เชื่อมต่อกันโดย Virtual Private Network (VPN) ผ่าน Core Layer และ Distribution Layer



รูปที่ 4.1 แบบจำลองโครงข่ายการทดสอบ

4.2 พารามิเตอร์ที่ใช้ในการจำลองระบบ

ในส่วนนี้จะทำการแสดงค่าพารามิเตอร์ที่ใช้ในแบบการจำลองการทดสอบระบบซึ่งสามารถแบ่งออกได้เป็น 4 ส่วนหลัก ๆ คือ

1. ค่าพารามิเตอร์ที่ใช้ในการตั้งค่าการใช้งาน MPLS

การออกแบบการโครงข่าย MPLS ประกอบด้วยการเชื่อมโยงอุปกรณ์ในลำดับชั้นแกนกลาง (Core Layer) มี Router R2 และ R3 ทำหน้าที่เป็น Provider Router เชื่อมต่อกันด้วย 10 Gigabit Ethernet Interface โดย R2 และ R3 ทำหน้าที่ในการส่งผ่านข้อมูลที่ได้เพิ่ม Label มาแล้วจาก Provider Edge Router ซึ่งอยู่ในลำดับชั้นการกระจาย (Distribution Layer) มี Router R1 ,R4 และ R5 โดยข้อมูล IP Packet จะถูกเพิ่มและถอด Label ที่ Provider Edge Router การกำหนดค่าพารามิเตอร์สำหรับการใช้งาน MPLS แสดงในตารางที่ 4.1

ตารางที่ 4.1 ค่าพารามิเตอร์ในการตั้งค่า MPLS บน Router R1- R5

!	mpls ldp router-id Loopback0 force
mls flow ip interface-full	!
mls qos	control-plane
mls cef error action reset	!
mls cef maximum-routes ipv6 1	
mls cef maximum-routes ip-multicast 1	
mpls traffic-eng tunnels	
mpls ldp explicit-null	
mpls ldp session protection	
mpls label protocol ldp	

2. ค่าพารามิเตอร์ที่ใช้ในการตั้งค่าการใช้งาน Fast Reroute (FRR)

ในส่วนของค่าพารามิเตอร์ในการตั้งค่าการทำ Fast Reroute (FRR) จะกำหนดที่ Router R1 , R2 และ R3 เพื่อทำให้โครงข่ายมีเสถียรภาพในการส่งข้อมูลมากยิ่งขึ้น โดยค่าพารามิเตอร์แสดงในตารางที่ 4.2 – 4.4

ตารางที่ 4.2 ค่าพารามิเตอร์ในการตั้งค่า Fast Reroute (FRR) บน Router R1

Router R1#	!
interface TenGigabitEthernet2/0/0	interface Loopback0
ip address 10.0.2.1 255.255.255.252	ip address 10.0.0.1 255.255.255.255
logging event link-status	!
mls qos trust dscp	
mpls traffic-eng tunnels	
mpls ip	
ip rsvp bandwidth	
!	
interface TenGigabitEthernet3/0/0	
ip address 10.0.2.9 255.255.255.252	
logging event link-status	
mls qos trust dscp	
mpls traffic-eng tunnels	
mpls ip	
ip rsvp bandwidth	
!	

ตารางที่ 4.3 ค่าพารามิเตอร์ในการตั้งค่า Fast Reroute (FRR) บน Router R2

Router R2#	interface TenGigabitEthernet2/0/0
interface Tunnel1	ip address 10.0.2.5 255.255.255.252
ip unnumbered Loopback0	logging event link-status
mpls ip	carrier-delay msec 0
tunnel destination 10.0.0.3	mls qos trust dscp
tunnel mode mpls traffic-eng	mpls traffic-eng tunnels
tunnel mpls traffic-eng autoroute announce	mpls traffic-eng backup-path Tunnel2
tunnel mpls traffic-eng priority 0 0	mpls ip
tunnel mpls traffic-eng path-option 1 explicit	ip rsvp bandwidth
name R2-to-R3	!
tunnel mpls traffic-eng path-option 2 dynamic	
tunnel mpls traffic-eng fast-reroute	interface TenGigabitEthernet3/0/0
!	ip address 10.0.2.2 255.255.255.252
interface Tunnel2	logging event link-status
ip unnumbered Loopback0	mls qos trust dscp
tunnel destination 10.0.0.3	mpls traffic-eng tunnels
tunnel mode mpls traffic-eng	mpls ip
tunnel mpls traffic-eng priority 0 0	ip rsvp bandwidth
tunnel mpls traffic-eng path-option 1 explicit	!
name R2-to-R1-to-R3	ip explicit-path name R2-to-R3 enable
!	next-address 10.0.2.6
interface Loopback0	!
ip address 10.0.0.2 255.255.255.255	ip explicit-path name R2-to-R1-to-R3 enable
!	next-address 10.0.2.1
	next-address 10.0.2.10

ตารางที่ 4.4 ค่าพารามิเตอร์ในการตั้งค่า Fast Reroute (FRR) บน Router R3

```
Router R3#
interface Tunnel1
 ip unnumbered Loopback0
 mpls ip
 tunnel destination 10.0.0.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng path-option 1 explicit
name R3-to-R2
 tunnel mpls traffic-eng path-option 2 dynamic
 tunnel mpls traffic-eng fast-reroute
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel destination 10.0.0.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng path-option 1 explicit
name R3-to-R1-to-R2
!
interface Loopback0
 ip address 10.0.0.3 255.255.255.255
!
interface TenGigabitEthernet2/0/0
 ip address 10.0.2.10 255.255.255.252
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface TenGigabitEthernet3/0/0
 ip address 10.0.2.6 255.255.255.252
 logging event link-status
 carrier-delay msec 0
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls traffic-eng backup-path Tunnel2
 mpls ip
 ip rsvp bandwidth
!
interface TenGigabitEthernet8/0/0
 ip address 10.0.2.25 255.255.255.252
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
ip explicit-path name R3-to-R2 enable
 next-address 10.0.2.5
!
ip explicit-path name R3-to-R1-to-R2 enable
 next-address 10.0.2.9
 next-address 10.0.2.2
!
```

3. ค่าพารามิเตอร์ในการตั้งค่า MPLS/VPN

ลำดับชั้นการกระจายประกอบด้วย Router R1 ,R4 และ R5 ทำหน้าที่เป็น Provider Edge Router เชื่อมต่อกันด้วย 10 Gigabit Ethernet Interface โดยข้อมูล IP packet จะถูกเพิ่มและถอด Label ที่ Provider Edge Router การทำ MPLS/VPN เพื่อเชื่อมต่อเครือข่ายระหว่าง Site1, Site2 และ Site3 ค่าพารามิเตอร์จะถูกกำหนดบนลำดับชั้นการกระจายเท่านั้น ไม่มีการกำหนดบนลำดับชั้นแกนกลาง ซึ่งค่าพารามิเตอร์สำหรับการใช้งาน MPLS/VPN แสดงในตารางที่ 4.5 – 4.10

ตารางที่ 4.5 ค่าพารามิเตอร์ในการตั้งค่า MPLS/VPN บน Router R1

ip vrf vpn501	router bgp 1
rd 501:1	bgp router-id 10.0.0.1
route-target export 501:1	no bgp default ipv4-unicast
route-target import 501:1	bgp log-neighbor-changes
ip vrf vpn502	bgp graceful-restart restart-time 120
rd 502:1	bgp graceful-restart stalepath-time 360
route-target export 502:1	bgp graceful-restart
route-target import 502:1	neighbor 10.0.0.4 remote-as 1
ip vrf vpn503	neighbor 10.0.0.4 update-source Loopback0
rd 503:1	neighbor 10.0.0.5 remote-as 1
route-target export 503:1	neighbor 10.0.0.5 update-source Loopback0
route-target import 503:1	!
!	address-family ipv4
interface Vlan501	no synchronization
ip vrf forwarding vpn501	no auto-summary
ip address 192.0.0.1 255.255.255.0	exit-address-family
!	!
interface Vlan502	address-family vpnv4
ip vrf forwarding vpn502	neighbor 10.0.0.4 activate
ip address 192.0.1.1 255.255.255.0	neighbor 10.0.0.4 send-community extended
!	neighbor 10.0.0.4 next-hop-self
interface Vlan503	neighbor 10.0.0.5 activate
ip vrf forwarding vpn503	neighbor 10.0.0.5 send-community extended
ip address 192.0.2.1 255.255.255.0	neighbor 10.0.0.5 next-hop-self
	exit-address-family

ตารางที่ 4.5 (ต่อ)

!	neighbor 192.0.1.2 activate
address-family ipv4 vrf vpn501	neighbor 192.0.1.2 next-hop-self
no synchronization	exit-address-family
redistribute connected	!
neighbor 192.0.0.2 remote-as 1000	address-family ipv4 vrf vpn503
neighbor 192.0.0.2 activate	no synchronization
neighbor 192.0.0.2 next-hop-self	redistribute connected
exit-address-family	neighbor 192.0.2.2 remote-as 1000
!	neighbor 192.0.2.2 activate
address-family ipv4 vrf vpn502	neighbor 192.0.2.2 next-hop-self
no synchronization	exit-address-family
redistribute connected	!
neighbor 192.0.1.2 remote-as 1000	

ตารางที่ 4.6 ค่าพารามิเตอร์ในการตั้งค่า MPLS/VPN (VPLS) บน Router R4

l2 vfi vpls2001 manual	!
vpn id 2001	interface Vlan2001
neighbor 10.0.0.4 encapsulation mpls	no ip address
neighbor 10.0.0.5 encapsulation mpls	xconnect vfi vpls2001
!	!
l2 vfi vpls2002 manual	interface Vlan2002
vpn id 2002	no ip address
neighbor 10.0.0.4 encapsulation mpls	xconnect vfi vpls2002
neighbor 10.0.0.5 encapsulation mpls	!
!	interface Vlan2003
l2 vfi vpls2003 manual	no ip address
vpn id 2003	xconnect vfi vpls2003
neighbor 10.0.0.4 encapsulation mpls	!
neighbor 10.0.0.5 encapsulation mpls	

ตารางที่ 4.7 ค่าพารามิเตอร์ในการตั้งค่า MPLS/VPN บน Router R4

```
ip vrf vpn501
 rd 501:1
 route-target export 501:1
 route-target import 501:1
ip vrf vpn502
 rd 502:1
 route-target export 502:1
 route-target import 502:1
ip vrf vpn503
 rd 503:1
 route-target export 503:1
 route-target import 503:1
!
interface Vlan501
 ip vrf forwarding vpn501
 ip address 192.10.0.1 255.255.255.0
!
interface Vlan502
 ip vrf forwarding vpn502
 ip address 192.10.1.1 255.255.255.0
!
interface Vlan503
 ip vrf forwarding vpn503
 ip address 192.10.2.1 255.255.255.0
!
router bgp 1
bgp router-id 10.0.0.4
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 update-source Loopback0
 neighbor 10.0.0.5 remote-as 1
 neighbor 10.0.0.5 update-source Loopback0
 !
 address-family ipv4
  no synchronization
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.1 send-community extended
  neighbor 10.0.0.1 next-hop-self
  neighbor 10.0.0.5 activate
  neighbor 10.0.0.5 send-community extended
  neighbor 10.0.0.5 next-hop-self
 exit-address-family
!
 address-family ipv4 vrf vpn501
  no synchronization
  redistribute connected
  neighbor 192.10.0.2 remote-as 1000
  neighbor 192.10.0.2 activate
  neighbor 192.10.0.2 next-hop-self
 exit-address-family
```

ตารางที่ 4.7 (ต่อ)

!	!
address-family ipv4 vrf vpn502	address-family ipv4 vrf vpn503
no synchronization	no synchronization
redistribute connected	redistribute connected
neighbor 192.10.1.2 remote-as 1000	neighbor 192.10.2.2 remote-as 1000
neighbor 192.10.1.2 activate	neighbor 192.10.2.2 activate
neighbor 192.10.1.2 next-hop-self	neighbor 192.10.2.2 next-hop-self
exit-address-family	exit-address-family
!	!

ตารางที่ 4.8 ค่าพารามิเตอร์ในการตั้งค่า MPLS/VPN (VPLS) บน Router R5

l2 vfi vpls2001 manual	interface Vlan2001
vpn id 2001	no ip address
neighbor 10.0.0.1 encapsulation mpls	xconnect vfi vpls2001
neighbor 10.0.0.5 encapsulation mpls	!
!	interface Vlan2002
l2 vfi vpls2002 manual	no ip address
vpn id 2002	xconnect vfi vpls2002
neighbor 10.0.0.1 encapsulation mpls	!
neighbor 10.0.0.5 encapsulation mpls	interface Vlan2003
!	no ip address
l2 vfi vpls2003 manual	xconnect vfi vpls2003
vpn id 2003	!
neighbor 10.0.0.1 encapsulation mpls	
neighbor 10.0.0.5 encapsulation mpls	
!	

ตารางที่ 4.9 ค่าพารามิเตอร์ในการตั้งค่า MPLS/VPN บน Router R5

```
ip vrf vpn501
 rd 501:1
 route-target export 501:1
 route-target import 501:1
ip vrf vpn502
 rd 502:1
 route-target export 502:1
 route-target import 502:1
ip vrf vpn503
 rd 503:1
 route-target export 503:1
 route-target import 503:1
!
interface Vlan501
 ip vrf forwarding vpn501
 ip address 192.20.0.1 255.255.255.0
!
interface Vlan502
 ip vrf forwarding vpn502
 ip address 192.20.1.1 255.255.255.0
!
interface Vlan503
 ip vrf forwarding vpn503
 ip address 192.20.2.1 255.255.255.0
!
router bgp 1
bgp router-id 10.0.0.5
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 update-source Loopback0
 neighbor 10.0.0.4 remote-as 1
 neighbor 10.0.0.4 update-source Loopback0
 !
address-family ipv4
  no synchronization
  no auto-summary
 exit-address-family
 !
address-family vpnv4
  neighbor 10.0.0.1 activate
  neighbor 10.0.0.1 send-community extended
  neighbor 10.0.0.1 next-hop-self
  neighbor 10.0.0.4 activate
  neighbor 10.0.0.4 send-community extended
  neighbor 10.0.0.4 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf vpn501
  no synchronization
  redistribute connected
  neighbor 192.20.0.2 remote-as 1000
  neighbor 192.20.0.2 activate
  neighbor 192.20.0.2 next-hop-self
 exit-address-family
```

ตารางที่ 4.9 (ต่อ)

!	!
address-family ipv4 vrf vpn502	address-family ipv4 vrf vpn503
no synchronization	no synchronization
redistribute connected	redistribute connected
neighbor 192.20.1.2 remote-as 1000	neighbor 192.20.2.2 remote-as 1000
neighbor 192.20.1.2 activate	neighbor 192.20.2.2 activate
neighbor 192.20.1.2 next-hop-self	neighbor 192.20.2.2 next-hop-self
exit-address-family	exit-address-family
!	!

ตารางที่ 4.10 ค่าพารามิเตอร์ในการตั้งค่า MPLS/VPN (VPLS) บน Router R5

l2 vfi vpls2001 manual	interface Vlan2001
vpn id 2001	no ip address
neighbor 10.0.0.1 encapsulation mpls	xconnect vfi vpls2001
neighbor 10.0.0.4 encapsulation mpls	!
!	interface Vlan2002
l2 vfi vpls2002 manual	no ip address
vpn id 2002	xconnect vfi vpls2002
neighbor 10.0.0.1 encapsulation mpls	!
neighbor 10.0.0.4 encapsulation mpls	interface Vlan2003
!	no ip address
l2 vfi vpls2003 manual	xconnect vfi vpls2003
vpn id 2003	!
neighbor 10.0.0.1 encapsulation mpls	
neighbor 10.0.0.4 encapsulation mpls	
!	

4. ค่าพารามิเตอร์ในการตั้งค่า MPLS DiffServ

ค่าพารามิเตอร์ในการตั้งค่า MPLS DiffServ จะกำหนดที่ลำดับชั้นการกระจายประกอบด้วย Router R1 ,R4 และ R5 ทำหน้าที่เป็น Provider Edge Router และลำดับชั้นแกนกลาง (Core Layer) มี Router R2 และ R3 ทำหน้าที่เป็น Provider Router โดยค่าพารามิเตอร์แสดงในตารางที่ 4.11 – 4.12

ตารางที่ 4.11 ค่าพารามิเตอร์ในการตั้งค่า MPLS DiffServ บน Router R1,R4,R5

```
class-map match-all Data-Traffic
 match dscp default
 !
class-map match-all Voice-Traffic
 match dscp ef
 !
class-map match-any Video-Traffic
 match dscp af31 af41
 !
policy-map Ingress-Policy
 class Data-Traffic
  set qos-group 0
!
 class Voice-Traffic
  set qos-group 5
!
 class Video-Traffic
  set qos-group 4
!
policy-map Egree-Policy
 class Data-Egress
    set mpls experimental topmost 0
    police 500000000
    conform-action transmit
    exceed-action drop
!
class Voice-Egress
    set mpls experimental topmost 5
    police 250000000
    conform-action transmit
    exceed-action drop
 !
 class Video-Egress
    set mpls experimental topmost 4
    police 250000000
    conform-action transmit
    exceed-action drop
!
interface g 1/1
 service-policy input Ingress-Policy
!
interface te 1/0
 service-policy output Egree-Policy
!
```

ตารางที่ 4.12 ค่าพารามิเตอร์ในการตั้งค่า MPLS DiffServ บน Router R2,R3

```
class-map match-all mplsexp0
 match mpls experimental 0
 !
class-map match-all mplsexp5
 match mpls experimental 5
!
class-map match-all mplsexp4
 match mpls experimental 4
!

policy-map MPLS-In
 class-map mplsexp0
 set qos-group 0
!
 class-map mplsexp5
 set qos-group 5
!
 class-map mplsexp4
 set qos-group 4
!

class-map match-all qosgroup0
 match qos-group 0
!
class-map match-all qosgroup5
 match qos-group 5
!
```

```
class-map match-all qosgroup4
 match qos-group 4
policy-map MPLS-Out
!
 class qosgroup0
 set mpls experimental topmost 0
!
 class qosgroup5
 set mpls experimental topmost 5
!
 class qosgroup4
 set mpls experimental topmost 4

int te 1/1
 service-policy input  MPLS-In
 service-policy output  MPLS-Out

int te 1/2
service-policy output  MPLS-Out
 service-policy input  MPLS-In
```

4.3 การทดสอบประสิทธิภาพของโครงข่าย

การทดสอบประสิทธิภาพของโครงข่ายโดยใช้ Traffic Generator จำลองเครือข่ายลำดับชั้นการเข้าถึง(Access Layer) ทำหน้าที่เป็นอุปกรณ์ CE (Customer Equipment) โดยทำการป้อนข้อมูล Ethernet ขนาด Frame size 64 byte ที่ความเร็ว 1 Gbps เชื่อมต่อไปยังลำดับชั้นการกระจาย (Distribution Layer) ทำหน้าที่เป็น Provider Edge Router ได้แก่ Router R1 ,R4 และ R5 โดยมีลำดับชั้นแกนกลางทำหน้าที่เป็น Provider Router ได้แก่ Router R2,R3 โดยแบ่งการทดสอบออกเป็น 2 ส่วนดังนี้

4.3.1 เปรียบเทียบประสิทธิภาพด้านความเชื่อถือได้ของระบบ (Reliability) ระหว่าง MPLS/VPN และ Traditional IP Network

การทดสอบประสิทธิภาพด้านความเชื่อถือได้ของระบบ (Reliability) เป็นการทดสอบความสามารถของระบบในการส่งข้อมูลเมื่อเส้นทางในการส่งข้อมูลเกิดความบกพร่อง เช่น สาย สายเคเบิลใยแก้วนำแสงที่ใช้เชื่อมต่อเกิดชำรุด โดยระบบที่มี Reliability ที่ดีต้องสามารถทำการสลับเพื่อเลือกเส้นทางไปยังเส้นทางสำรองได้โดยไม่กระทบกับการส่งข้อมูล

4.3.1.1 MPLS/VPN

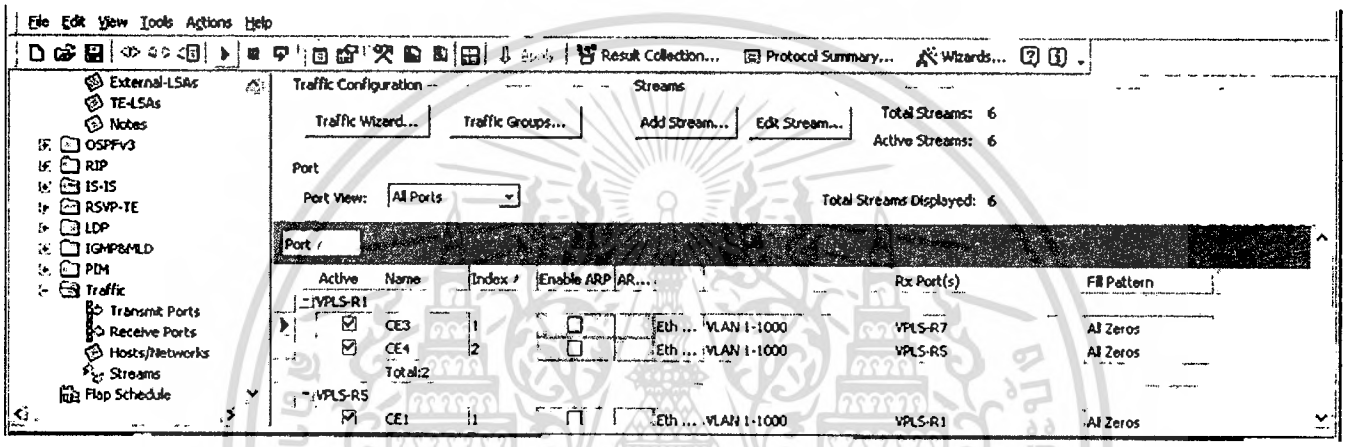
ทำการ Configured Router เพื่อให้บริการ MPLS Traffic Engineering (TE) Fast Reroute ในกรณีที่ Link ระหว่าง Node เกิดการบกพร่อง จากรูปที่ 4.1 ข้อมูลจะถูกส่งจาก Traffic Generator ที่ความเร็ว 1Gbps ลักษณะการส่งข้อมูลจะเป็นแบบ Bidirection ส่งผ่านระหว่าง Site 1 และ Site 2 โดย R2 และ R3 จะทำการ Enable การใช้งาน Fast Reroute (FRR) โดยใช้ Resource Reservation Protocol (RSVP) ในการสถาปนา TE Tunnel เส้นทางหลักในการส่งข้อมูลระหว่าง Site 1 และ Site2 คือ R5->R2->R3->R4 ในกรณีที่เส้นทางหลักเกิดการบกพร่องข้อมูลจะส่งผ่านไปยังเส้นทางสำรอง R5->R2->R1->R3->R4

```
R2#show mpls traffic-eng tunnel tunnel 1

Name: R2_t1                          (Tunnel1) Destination: 10.0.0.3
 Status:
   Admin: up       Oper: up    Path: valid      Signalling: connected
   path option 1, type explicit R2-to-R3 (Basis for Setup, path weight 1)
   path option 2, type dynamic
```

รูปที่ 4.2 แสดงการตั้งค่า FRR เส้นทางหลักที่ R2

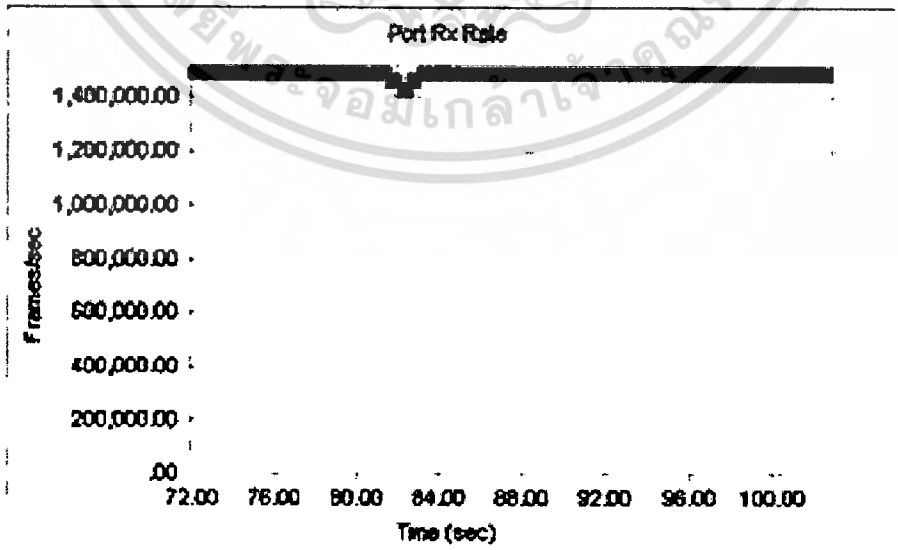
```
R2#show mpls traffic-eng tunnel tunnel 2

Name: R2_t2                          (Tunnel2) Destination: 10.0.0.3
 Status:
   Admin: up       Oper: up    Path: valid      Signalling: connected
   path option 1, type explicit R2-to-R1-to-R3 (Basis for Setup, path weight 2)
```

รูปที่ 4.3 แสดงการตั้งค่า FRR เส้นทางสำรองที่ R2



รูปที่ 4.4 แสดงการตั้งค่า Traffic Generator เพื่อจำลองเป็นอุปกรณ์ CE

รูปที่ 4.5 แสดงผลการทดสอบเมื่อทำการปลดสาย Fiber Optic ที่เชื่อมต่อระหว่าง Router R2 และ R3



รูปที่ 4.5 ค่า Throughput ของโครงข่าย MPLS/VPN ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

ตารางที่ 4.13 ค่า Packet Loss ของโครงข่าย MPLS/VPN ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

Stream	Expected Frames	Tx Frames	Lost Frames	% Loss
Stream1	267857142	267857142	55951	0.02089
Stream2	267857142	267857142	35661	0.01331
Total	535714284	535714284	91612	0.0171

จากตารางที่ 4.9 เกิดการสูญหายของ Packet จำนวน 91612 Packet มีค่า Recovery Time ในการเปลี่ยนเส้นทางจากเส้นทางหลักไปใช้เส้นทางสำรอง 61 msec โดยสามารถหาค่า Ethernet Frame Rate (Frame/Second) ได้จาก

$$= \frac{Interface\ speed\ bps}{(Ethernet\ frame\ size\ byte+preamble\ size\ byte+inter\ frame\ gap\ size\ byte\ )\times 8}$$

เมื่อ $Interface\ speed\ bps$ มีค่า 1Gbps

$Ethernet\ frame\ size\ byte$ มีค่า 64 byte

$preamble\ size\ byte$ มีค่า 8 byte

$inter\ frame\ gap\ size\ byte$ มีค่า 12 byte

Ethernet frame rate (frame/second)

$$= \frac{1Gbps}{(64\ byte+8\ byte+12\ byte)\times 8}$$

$$= 1,488,095\ frame/second \qquad (1)$$

จาก (1) สามารถหาค่า recovery time ที่เกิด lost frames ได้จาก

$$= \frac{lost\ frame}{Ethernet\ frame\ rate\ (frame\ /second)}$$

$$= \frac{91612}{1,488,095\ (frame\ /second)}$$

$$= 61\ msec \qquad (2)$$

4.3.1.2 Traditional IP Network

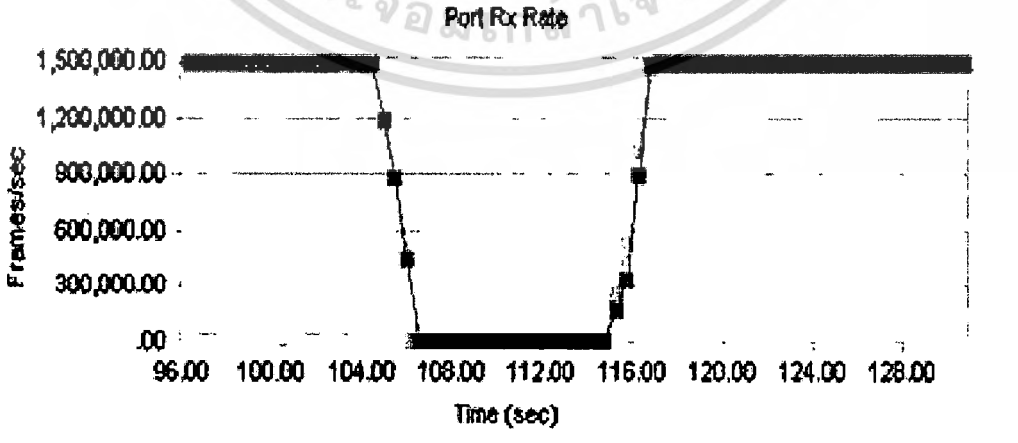
ทำการ Configured Router Traditional IP Network ทำงานโดย Routing Protocol OSPF (Open Shortest Path First) เพื่อหาเส้นทางในการส่งข้อมูล ด้วยการประกาศข้อมูลของเส้นทางเช่น Bandwidth Latency Time เพื่อใช้ประกอบในการคำนวณหาเส้นทางที่ดีที่สุด ถ้าเส้นทางที่ใช้ในการส่งข้อมูลเกิดการบกพร่อง (Fault) Routing Protocol OSPF ต้องทำการคำนวณหาเส้นทางใหม่เพื่อใช้ในการส่งข้อมูลแทนเส้นทางหลัก

```
sh ip route summary
IP routing table name is default (0x0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets    Replicates  Overhead    Memory (bytes)
connected       0           11         0           572         1892
static          0           0          0           0           0
ospf 1          20412       905        0           1108536     3751792
  Intra-area: 317 Inter-area: 1000 External-1: 20000 External-2: 0
  NSSA External-1: 0 NSSA External-2: 0
```

รูปที่ 4.6 Routing Table ของ โครงข่าย Tradition IP ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

ตามรูปที่ 4.1 เส้นทางหลักในการส่งข้อมูลระหว่าง Site 1 และ Site 2 คือ R5->R2->R3->R4 ซึ่งเป็นเส้นทางที่ให้ค่า Cost Path ต่ำสุด รูปที่ 4.5 แสดงผลการทดสอบเมื่อทำการปลดสาย fiber optic ที่เชื่อมต่อระหว่าง Router R2 และ R3 ในขณะที่ทำการปลดสาย Fiber Optic ที่เชื่อมต่อระหว่าง Router R2 และ R3 ค่า Throughput ตกลงเหลือ 0 pps แต่เมื่อ Routing Protocol OSPF ทำการคำนวณหาเส้นทางในการส่งข้อมูลใหม่ได้สำเร็จ ซึ่งก็คือเส้นทาง R5->R2->R1->R3->R4 ระบบก็สามารถส่งข้อมูลได้ปกติ จากรูปที่ 4.7 สามารถหาค่า Recovery Time ได้โดยดูจากกราฟช่วงเวลาที่ค่า Throughput เริ่มลดลงเป็นศูนย์จนเริ่มกลับมาส่งข้อมูลได้อีกครั้งมีค่าประมาณ

117 sec - 104 sec = 13 sec (3)



รูปที่ 4.7 ค่า Throughput ของโครงข่าย Traditional IP ในการ ทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

ตารางที่ 4.14 เปรียบเทียบค่า Recovery Time , Packet Loss ระหว่าง MPLS/VPN และ Traditional IP Network ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

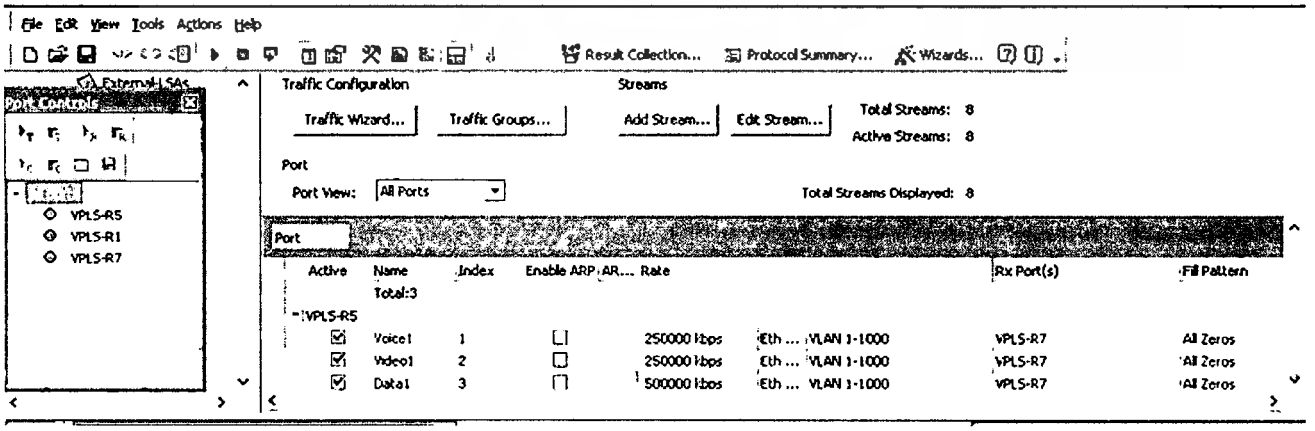
Technology	Recovery Time	% Loss
MPLS/VPN	61 msec	0.0171
Tradition IP	13 sec	100

4.3.2 เปรียบเทียบประสิทธิภาพด้านคุณภาพการบริการ (Quality of Service) ระหว่าง MPLS/VPN และ Traditional IP

การทดสอบประสิทธิภาพด้านคุณภาพการบริการ (Quality of Service) เป็นการทดสอบความสามารถในการให้บริการข้อมูลประเภท Voice Video และ Data โดยโครงข่าย MPLS/VPN สามารถจัดลำดับความสำคัญของข้อมูลประเภท Voice Video ให้สามารถใช้งานได้โดยไม่มีผลกระทบต่อคุณภาพแม้โครงข่ายจะเกิดความคับคั่ง การทดลองจะทำการส่งข้อมูลที่ความเร็ว 1 Gbps จาก Site1และ Site2 ไปยัง Site3 โดยทั้ง 3 Site เชื่อมต่อด้วย Gigabit Ethernet Interface ดังนั้นจึงมีข้อมูลขนาด 2 Gbps วิ่งเข้าไปยัง Site3 ทำให้ Site3 เกิดความคับคั่งของข้อมูล ซึ่งใน Bandwidth 1 Gbps แบ่งการส่งเป็น UDP Traffic Voice 250 Mbps มีค่าลำดับความสำคัญสูงสุด UDP Traffic Video 250 Mbps มีค่าลำดับความสำคัญลำดับสองและ TCP Traffic Data 500 Mbps มีค่าลำดับความสำคัญลำดับสาม ซึ่ง Traffic ทั้งสามแบบตั้งค่าให้อยู่ VPN ที่ต่างกัน

4.3.2.1 MPLS/VPN

จากรูปที่ 4.1 MPLS Network ทำการ Configured Router เพื่อทำฟังก์ชั่น MPLS DiffServ และทำการส่งข้อมูลจาก Site1และ Site2 ไปยัง Site3 หลังจากนั้นทำการวัดค่า Packet Loss ที่ Site3



รูปที่ 4.8 แสดงการตั้งค่า Traffic Generator เพื่อส่งข้อมูล Voice ,Video และ Data

ตารางที่ 4.15 ค่า Frames Loss ของโครงข่าย MPLS/VPN ในการทดสอบประสิทธิภาพด้านคุณภาพการบริการ

Stream	Expected Frames	Tx Frames	Lost Frames	% Loss
voice	12668027	12668027	0	0
video	12668026	12668026	6634	0.05237
data	25339587	25339587	25338574	99.996

```
show mpls l2 vc | i vpls
VFI vpls2001   VFI                    10.0.0.5       2001       UP
VFI vpls2002   VFI                    10.0.0.5       2002       UP
VFI vpls2003   VFI                    10.0.0.5       2003       UP
VFI vpls2004   VFI                    10.0.0.5       2004       UP
VFI vpls2005   VFI                    10.0.0.5       2005       UP
VFI vpls2006   VFI                    10.0.0.5       2006       UP
VFI vpls2007   VFI                    10.0.0.5       2007       UP
VFI vpls2008   VFI                    10.0.0.5       2008       UP
VFI vpls2009   VFI                    10.0.0.5       2009       UP
```

รูปที่ 4.9 แสดงการตั้งค่า MPLS/VPN ในการทดสอบประสิทธิภาพด้านคุณภาพการบริการ

```
sh queueing interface gi1/1
Interface GigabitEthernet1/1 queueing strategy:
Weighted Round-Robin
 Port QoS is enabled
 Trust state: trust COS
 Extend trust state: not trusted [COS = 0]
 Default COS is 0
   Queueing Mode In Tx direction: mode-cos
   Transmit queues [type = 1p3q8t]:
   Queue Id    Scheduling  Num of thresholds
```

รูปที่ 4.10 แสดงการกำหนดค่า policy ในการกำหนด QOS แบบ MPLS DiffServ บน Interface Gigabit Ethernet 1/1

4.3.2.2 Traditional IP Network

ทำการ Configured Router เพื่อทำฟังก์ชั่น การให้บริการแบบ Best Effort และทำการส่งข้อมูลจาก Site1และ Site2 ไปยัง Site3 หลังจากนั้นทำการวัดค่า Packet Loss ที่ Site3 ซึ่งแสดงได้ดังตารางที่ 4.12

ตารางที่ 4.16 ค่า Frames Loss ของโครงข่าย Traditional IP ในการทดสอบประสิทธิภาพด้านคุณภาพการบริการ

Stream	Expected Frames	Tx Frames	Lost Frames	% Loss
voice	12668027	12668027	6589531	52.017
video	12668026	12668026	6589521	52.016
data	25339587	25339587	13187811	52.044

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

ความต้องการใช้เทคโนโลยีการสื่อสารข้อมูล Internet Protocol ในการให้บริการรูปแบบต่าง ๆ เช่น โครงข่ายเสมือนส่วนบุคคล (Virtual Private Network : VPN) Voice Video มีการใช้งานเพิ่มมากขึ้นอย่างรวดเร็ว โครงข่ายที่รองรับจึงต้องมีความสามารถในการให้บริการ มีความยืดหยุ่นและมีความน่าเชื่อถือสูง ซึ่งปัจจุบันโครงข่ายสื่อสารข้อมูล Internet Protocol ที่ใช้เทคนิคการทำ Routing เพื่อหาเส้นทางในการส่งข้อมูลไม่สามารถรองรับการให้บริการข้อมูลที่ต้องการความมีน่าเชื่อถือสูงเช่น Voice และ Video ได้อย่างที่ต้องการ เนื่องจากจะเกิดปัญหาในเรื่องความล่าช้าในการส่งข้อมูลเมื่อโครงข่ายเกิดความบกพร่องหรือมีความคับคั่งของข้อมูล เทคโนโลยี MPLS หรือมีชื่อเรียกเต็ม ๆ ว่า Multiprotocol Label Switching เป็นเทคโนโลยีสำหรับการบริหารจัดการเส้นทางและควบคุมคุณภาพของสัญญาณเชื่อมต่อบนเครือข่าย สื่อสารข้อมูล Internet Protocol ด้วยกระบวนการในการเร่งการจัดส่ง IP-Packet และให้ความยืดหยุ่นสำหรับการจัดการ IP บนเครือข่าย

การสื่อสารข้อมูล Internet Protocol โดยการใช้เทคโนโลยี MPLS/VPN ร่วมกับการทำ MPLS Traffic Engineering (TE) Fast Reroute และ MPLS DiffServ เป็นเทคนิคหนึ่งที่สามารถช่วยให้การส่งข้อมูลประเภท Voice และ Video นั้นสามารถกระทำได้เร็วขึ้นและมีประสิทธิภาพยิ่งขึ้น ซึ่งข้อดีของ Multiprotocol Label Switching (MPLS) มีดังนี้

- มีความเสถียรและปลอดภัยสูงในการรับ-ส่งข้อมูล
- มีปริมาณช่องสัญญาณ (Bandwidth) มากถึง 10 Gbps เพื่อรองรับลูกค้ากลุ่มธุรกิจโดยเฉพาะ
- สามารถเลือกความเร็วได้ตั้งแต่ 64 Kbps-10 Gbps
- พร้อมรองรับ IP Application ต่างๆ ไม่ว่าจะเป็น VoIP, Routing Protocol, QoS, Multicast และ VDO Conference เพื่อตอบสนองชีวิตการทำงาน แบบที่จะเป็นที่นิยมในอนาคต โดยการรวมเทคโนโลยีต่างๆ ไว้เข้าด้วยกัน เพื่ออำนวยความสะดวกในการทำงาน

ในวิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการทดสอบเปรียบเทียบประสิทธิภาพระหว่างโครงข่าย MPLS/VPN กับโครงข่าย IP แบบดั้งเดิม ในด้านความน่าเชื่อถือของระบบ (Reliability) และความสามารถในด้านคุณภาพการบริการ (Quality of Service) โดยใช้เทคนิคการทำ MPLS Traffic Engineering (TE) Fast Reroute และ MPLS DiffServ แทนการทำงานของ OSPF Routing Protocol และ Best Effort Protocol โดยทำการทดสอบบนโครงข่ายจริงที่ระดับความเร็ว 10 Gbps

ซึ่งจากการทดลองวิธีการที่นำเสนอนี้ ได้แสดงให้เห็นว่าโครงข่าย MPLS/VPN ที่ใช้หลักการ MPLS Traffic Engineering (TE) Fast Reroute และ MPLS DiffServ มีประสิทธิภาพตามหัวข้อที่กล่าวมา ดีกว่าโครงข่าย IP แบบดั้งเดิมที่ใช้หลักการ OSPF Routing Protocol และ Best Effort Protocol

วิธีการที่นำเสนอในวิทยานิพนธ์เป็นเทคนิคหนึ่งเท่านั้นที่ช่วยในการปรับปรุงประสิทธิภาพในด้านความน่าเชื่อถือของระบบ (Reliability) และความสามารถด้านคุณภาพการบริการ (Quality of Service) ของระบบการสื่อสารข้อมูล Internet Protocol แต่ก็ยังมีเทคนิควิธีการอื่นที่น่าสนใจ ที่สามารถนำมาใช้ควบคู่ไปกับการทำ MPLS Traffic Engineering Fast Reroute และ MPLS DiffServ เช่นการทำ MPLS Intserv มาใช้ร่วมกันก็จะทำให้ระบบมีประสิทธิภาพที่ดีมากยิ่งขึ้น

บรรณานุกรม

[1] L. Lobo and U. Lakshman, **MPLS Configuration on Cisco IOS Software,** Indianapolis, Indiana Cisco Press,2005.

[2] Student Guide **Implementing Cisco MPLS** Indianapolis, Indiana : Cisco Press,2004

[3] S. Alvarez, **QoS for IP/MPLS Networks** Indianapolis, Indiana Cisco Press,2006

[4] J. Evans and C. Filsfils, **Deploying IP and MPLS QOS for Multiservice Network**, San Francisco, Morgan Kaufmann Publishers,2007

[5] J. T. Moy, **OSPF: Anatomy of an Internet Routing Protocol** , United States of America,Addison-Wesley,2000

[6] W.Y. Lee, R. Bhagavathula, N. Thanthry and R. Pendse, **"MPLS-over-GRE Base VPN Architecture: A Performance Comparison,"** Proc. of the 2002 (45th) IEEE Midwest Symposium on Circuits and Systems (MWSCAS-2002), 4-7 Aug 2002.

[7] F. Fujikawa, K. Kuwabara, Y. Koda, and M. Kiuchi, **"Examination of Electric Power Utility Network Applying IP Router/MPLS Router/Wide-Area Ethernet,"** IEEE Power Engineering Society General Meeting, 6-10 June 2004.

[8] J. Barakovic, H. Bajric, and A. Husic, **"Multimedia Traffic Analysis of MPLS and non-MPLS Network,"** IEEE Multimedia Signal Processing and Communications, 48th International Symposium ELMAR-2006, June 2006.

[9] D.L. Zhang and D. Ionescu, **"QoS Performance Analysis in Deployment of DiffServ-aware MPLS Traffic Engineering,"** Proc. of 2007 IEEE Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007). ACIS International Conference, July 30 2007-Aug. 1 2007

[10] S. Kim, H.- Y. Ryu, Jaehyung Park, and Taell Kim, **"Design and implementation of Martini based Layer 2 VPN"**, Proc. of the 8th IEEE International Conference on Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, 20-22 Feb. 2006

[11] B. Alawieh, and. H.T Mouftah, **"Efficient Delivery of Voice Services over MPLS Internet Infrastructure,"** Proc. of 2007 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2007), 22-26 April 2007.

ภาคผนวก

ภาคผนวก ก.

โปรแกรมที่ใช้ในการจำลองการทำงานของระบบ MPLS/VPN และ Traditional IP Network

```
**********************************************
        Configuration Provider Edge  Router  R1
**********************************************
!
no upgrade fpd auto
version 12.2
service timestamps debug datetime localtime
service timestamps log datetime msec
no service password-encryption
service counters max age 10
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone ICT 7
ip subnet-zero
ip vrf vpn501
 rd 501:1
 route-target export 501:1
 route-target import 501:1
!
ip vrf vpn502

 rd 502:1
 route-target export 502:1
 route-target import 502:1
!
ip vrf vpn503
 rd 503:1
 route-target export 503:1
 route-target import 503:1
!
ip vrf vpn504
 rd 504:1
 route-target export 504:1
 route-target import 504:1
!
ip vrf vpn505
 rd 505:1
 route-target export 505:1
 route-target import 505:1
!
ip vrf vpn506
 rd 506:1
 route-target export 506:1
 route-target import 506:1
!
ip vrf vpn507
 rd 507:1
 route-target export 507:1
 route-target import 507:1
!
ip vrf vpn508
 rd 508:1
 route-target export 508:1
 route-target import 508:1
!
ip vrf vpn509
 rd 509:1
 route-target export 509:1
 route-target import 509:1
!
ip vrf vpn510
 rd 510:1
 route-target export 510:1
 route-target import 510:1
!
no ip domain lookup
!
!
vtp mode transparent
mls ip multicast flow-stat-timer 9
no mls flow ip
no mls flow ipv6
mls qos
no mls acl tcam share-global
mls cef error action reset
mls cef maximum-routes ipv6 1
mls cef maximum-routes ip-multicast 1
multilink bundle-name authenticated
mpls traffic-eng tunnels
mpls ldp graceful-restart timers max-recovery 600
mpls ldp graceful-restart
mpls ldp session protection
```

```
mpls label protocol ldp
!
!
spanning-tree mode pvst
spanning-tree extend system-id
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
username admin privilege 15 secret 5
$1$vLbn$sChE3Ya.B5GpFvZu3UPdN1
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
vlan 99,501-900,2001-3000
!
l2 vfi vpls2001 manual
 vpn id 2001
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2002 manual
 vpn id 2002
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2003 manual
 vpn id 2003
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2004 manual
 vpn id 2004
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2005 manual
 vpn id 2005
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2006 manual
 vpn id 2006
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2007 manual
 vpn id 2007
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2008 manual
 vpn id 2008
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2009 manual
 vpn id 2009
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2010 manual
 vpn id 2010
 neighbor 10.0.0.5 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
!
interface Loopback0
 ip address 10.0.0.1 255.255.255.255
!
interface GigabitEthernet1/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2001-3000
 switchport mode trunk
 mls qos trust cos
!
interface GigabitEthernet1/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 99,501-900
 switchport mode trunk
!
interface GigabitEthernet1/3
```

```
 no ip address
 shutdown
!
interface GigabitEthernet1/4
 no ip address
 shutdown
!
interface GigabitEthernet1/5
 no ip address
 shutdown
!
interface GigabitEthernet1/6
 no ip address
 shutdown
!
interface GigabitEthernet1/7
 no ip address
 shutdown
!
interface GigabitEthernet1/8
 no ip address
 shutdown
!
interface GigabitEthernet1/9
 no ip address
 shutdown
!
interface GigabitEthernet1/10
 no ip address
 shutdown
!
interface GigabitEthernet1/11
 no ip address
 shutdown
!
interface GigabitEthernet1/12
 no ip address
 shutdown
!
interface GigabitEthernet1/13
 no ip address
 shutdown
!
interface GigabitEthernet1/14
 no ip address
 shutdown
!
interface GigabitEthernet1/15
 no ip address
 shutdown
!
interface GigabitEthernet1/16
 no ip address
 shutdown
!
interface GigabitEthernet1/17
 no ip address
 shutdown
!
interface GigabitEthernet1/18
 no ip address
 shutdown
!
interface GigabitEthernet1/19
 no ip address
 shutdown
!
interface GigabitEthernet1/20
 no ip address
 shutdown
!
interface GigabitEthernet1/21
 no ip address
 shutdown
!
interface GigabitEthernet1/22
 no ip address
 shutdown
!
interface GigabitEthernet1/23
 no ip address
 shutdown
!
interface GigabitEthernet1/24
 no ip address
 shutdown
!
interface TenGigabitEthernet2/0/0
```

```
 ip address 10.0.2.1 255.255.255.252
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface TenGigabitEthernet3/0/0
 ip address 10.0.2.9 255.255.255.252
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface GigabitEthernet5/1
 no ip address
 shutdown
!
interface GigabitEthernet5/2
 ip address 192.168.1.1 255.255.255.0
 media-type rj45
 no cdp enable
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan501
 ip vrf forwarding vpn501
 ip address 192.0.0.1 255.255.255.0
!
interface Vlan502
 ip vrf forwarding vpn502
 ip address 192.0.1.1 255.255.255.0
!
interface Vlan503
 ip vrf forwarding vpn503
 ip address 192.0.2.1 255.255.255.0
!
interface Vlan504
 ip vrf forwarding vpn504
 ip address 192.0.3.1 255.255.255.0
!
interface Vlan505
 ip vrf forwarding vpn505
 ip address 192.0.4.1 255.255.255.0
!
interface Vlan506
 ip vrf forwarding vpn506
 ip address 192.0.5.1 255.255.255.0
!
interface Vlan507
 ip vrf forwarding vpn507
 ip address 192.0.6.1 255.255.255.0
!
interface Vlan508
 ip vrf forwarding vpn508
 ip address 192.0.7.1 255.255.255.0
!
interface Vlan509
 ip vrf forwarding vpn509
 ip address 192.0.8.1 255.255.255.0
!
interface Vlan510
 ip vrf forwarding vpn510
 ip address 192.0.9.1 255.255.255.0
!
!
interface Vlan2001
 no ip address
 xconnect vfi vpls2001
!
interface Vlan2002
 no ip address
 xconnect vfi vpls2002
!
interface Vlan2003
 no ip address
 xconnect vfi vpls2003
!
interface Vlan2004
 no ip address
 xconnect vfi vpls2004
!
interface Vlan2005
 no ip address
 xconnect vfi vpls2005
```

```
!
interface Vlan2006
 no ip address
 xconnect vfi vpls2006
!
interface Vlan2007
 no ip address
 xconnect vfi vpls2007
!
interface Vlan2008
 no ip address
 xconnect vfi vpls2008
!
interface Vlan2009
 no ip address
 xconnect vfi vpls2009
!
interface Vlan2010
 no ip address
 xconnect vfi vpls2010
!
class-map match-all Data-Traffic
 match dscp default
 !
class-map match-all Voice-Traffic
 match dscp ef
 !
class-map match-any Video-Traffic
 match dscp af31 af41
 !

policy-map Ingress-Policy
 class Data-Traffic
  set qos-group 0
!
 class Voice-Traffic
  set qos-group 5
!
 class Video-Traffic
  set qos-group 4
!

class-map match-all qosgroup0
  match qos-group 0
!
class-map match-all qosgroup5
  match qos-group 5
!
class-map match-all qosgroup4
  match qos-group 4
!

policy-map Egree-Policy
 class qosgroup0
    set mpls experimental topmost 0
    police 500000000
    conform-action transmit
    exceed-action drop
 !
 class qosgroup5
    set mpls experimental topmost 5
    police 250000000
    conform-action transmit
    exceed-action drop
 !
 class qosgroup4
    set mpls experimental topmost 4
    police 250000000
    conform-action transmit
    exceed-action drop
 !
 interface GigabitEthernet1/1
 des conntect to CE
 service-policy input Ingress-Policy
 !
interface GigabitEthernet2/1
 des conntect to CE
 service-policy input Ingress-Policy
 !
 !
 interface TenGigabitEthernet2/0/0
 des connect to R2
 service-policy output Egree-Policy
 !
interface TenGigabitEthernet3/0/0
 des connect to R3
 service-policy output Egree-Policy
 !
```

```
!
router ospf 1
 log-adjacency-changes
 nsf cisco
 passive-interface GigabitEthernet5/2
 network 0.0.0.0 255.255.255.255 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng multicast-intact
!
router bgp 1
 bgp router-id 10.0.0.1
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.0.0.4 remote-as 1
 neighbor 10.0.0.4 update-source Loopback0
 neighbor 10.0.0.5 remote-as 1
 neighbor 10.0.0.5 update-source Loopback0
 !
 address-family ipv4
  no synchronization
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.0.0.4 activate
  neighbor 10.0.0.4 send-community extended
  neighbor 10.0.0.4 next-hop-self
  neighbor 10.0.0.5 activate
  neighbor 10.0.0.5 send-community extended
  neighbor 10.0.0.5 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf vpn501
  no synchronization
  redistribute connected
  neighbor 192.0.0.2 remote-as 1000
  neighbor 192.0.0.2 activate
  neighbor 192.0.0.2 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf vpn502
  no synchronization
  redistribute connected
  neighbor 192.0.1.2 remote-as 1000
  neighbor 192.0.1.2 activate
  neighbor 192.0.1.2 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf vpn503
  no synchronization
  redistribute connected
  neighbor 192.0.2.2 remote-as 1000
  neighbor 192.0.2.2 activate
  neighbor 192.0.2.2 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf vpn504
  no synchronization
  redistribute connected
  neighbor 192.0.3.2 remote-as 1000
  neighbor 192.0.3.2 activate
  neighbor 192.0.3.2 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf vpn505
  no synchronization
  redistribute connected
  neighbor 192.0.4.2 remote-as 1000
  neighbor 192.0.4.2 activate
  neighbor 192.0.4.2 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf vpn506
  no synchronization
  redistribute connected
  neighbor 192.0.5.2 remote-as 1000
  neighbor 192.0.5.2 activate
  neighbor 192.0.5.2 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf vpn507
  no synchronization
  redistribute connected
  neighbor 192.0.6.2 remote-as 1000
```

```
 neighbor 192.0.6.2 activate
 neighbor 192.0.6.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn508
 no synchronization
 redistribute connected
 neighbor 192.0.7.2 remote-as 1000
 neighbor 192.0.7.2 activate
 neighbor 192.0.7.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn509
 no synchronization
 redistribute connected
 neighbor 192.0.8.2 remote-as 1000
 neighbor 192.0.8.2 activate
 neighbor 192.0.8.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn510
 no synchronization
 redistribute connected
 neighbor 192.0.9.2 remote-as 1000
 neighbor 192.0.9.2 activate
 neighbor 192.0.9.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn511
 no synchronization
 redistribute connected
 neighbor 192.0.10.2 remote-as 1000
 neighbor 192.0.10.2 activate
 neighbor 192.0.10.2 next-hop-self
exit-address-family
!
ip classless
!
!
no ip http server
no ip http secure-server
!
logging trap notifications
logging 192.168.1.10
logging 192.168.1.20
!
snmp-server community c1scoro RO
snmp-server community c1scorw RW
snmp-server system-shutdown
snmp-server enable traps snmp authentication linkdown
linkup coldstart warmstart
snmp-server enable traps ds1
snmp-server enable traps chassis
snmp-server enable traps module
snmp-server enable traps transceiver all
snmp-server enable traps eigrp
snmp-server enable traps casa
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change
nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change
shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change
shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps event-manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps MAC-Notification move
threshold
snmp-server enable traps msdp
snmp-server enable traps pim neighbor-change rp-
mapping-change invalid-pim-message
snmp-server enable traps rf
snmp-server enable traps rtr
```

```
snmp-server enable traps slb real virtual csrp
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-
inconsistency loop-inconsistency
snmp-server enable traps syslog
snmp-server enable traps flash insertion removal
snmp-server enable traps memory bufferpeak
snmp-server enable traps flex-links status
snmp-server enable traps csg agent quota-server database
snmp-server enable traps sonet
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps resource-policy
snmp-server enable traps ethernet cfm cc mep-up mep-
down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-
missing mep-unknown service-up
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps c6kxbar intbus-crcexcd intbus-
crcrcvrd swbus
snmp-server enable traps dot1x
snmp-server enable traps envmon fan shutdown supply
temperature status
snmp-server enable traps port-security
snmp-server enable traps mvpn
snmp-server enable traps mpls traffic-eng
snmp-server enable traps mpls fast-reroute protected
snmp-server enable traps mpls rfc ldp
snmp-server enable traps mpls ldp
snmp-server enable traps pw vc
snmp-server enable traps isakmp policy add
snmp-server enable traps isakmp policy delete
snmp-server enable traps isakmp tunnel start
snmp-server enable traps isakmp tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps alarms informational
snmp-server enable traps vlan-mac-limit
snmp-server enable traps mpls vpn
snmp-server host 192.168.1.10 version 2c c1scoro
snmp-server host 192.168.1.20 version 2c c1scoro
!
mpls ldp router-id Loopback0 force
!
control-plane
!
!
line con 0
 exec-timeout 60 0
 privilege level 15
 password c1sco
 logging synchronous
 login
line vty 0 4
 exec-timeout 60 0
 privilege level 15
 login local
 transport input lat pad udptn telnet rlogin ssh acercon
line vty 5 15
 exec-timeout 60 0
 privilege level 15
 login local
!
exception crashinfo buffersize 80
mac-address-table aging-time 600
!
end
```

```
**********************************************
        Configuration Provider  Router  R2
**********************************************
!
no upgrade fpd auto
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service counters max age 10
!
hostname R2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone ICT 7
ip subnet-zero
!
!
no ip domain lookup
!
!
!
!
vtp mode transparent
mls ip multicast flow-stat-timer 9
mls flow ip interface-full
no mls flow ipv6
mls qos
no mls acl tcam share-global
mls cef error action reset
mls cef maximum-routes ipv6 1
mls cef maximum-routes ip-multicast 1
multilink bundle-name authenticated
mpls traffic-eng tunnels
mpls ldp explicit-null
mpls ldp graceful-restart timers max-recovery 600
mpls ldp graceful-restart
mpls ldp session protection
mpls label protocol ldp
!
!
spanning-tree mode pvst
spanning-tree extend system-id
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
username admin privilege 15 secret 5
$1$vLbn$sChE3Ya.B5GpFvZu3UPdN1
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
interface Tunnel1
 ip unnumbered Loopback0
 mpls ip
 tunnel destination 10.0.0.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng path-option 1 explicit name R2-
to-R3
 tunnel mpls traffic-eng path-option 2 dynamic
 tunnel mpls traffic-eng fast-reroute
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel destination 10.0.0.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng path-option 1 explicit name R2-
to-R1-to-R3
!
interface Loopback0
 ip address 10.0.0.2 255.255.255.255
!
!
class-map match-all mplsexp0
 match mpls experimental 0
!
class-map match-all mplsexp5
```

```
 match mpls experimental 5
!
class-map match-all mplsexp4
 match mpls experimental 4
!

policy-map MPLS-In
 class-map mplsexp0
 set qos-group 0
!
 class-map mplsexp5
 set qos-group 5
!
 class-map mplsexp4
 set qos-group 4
!

class-map match-all qosgroup0
 match qos-group 0
!
class-map match-all qosgroup5
 match qos-group 5
!
class-map match-all qosgroup4
 match qos-group 4

policy-map MPLS-Out
!
 class qosgroup0
 set mpls experimental topmost 0
!
 class qosgroup5
 set mpls experimental topmost 5
!
 class qosgroup4
 set mpls experimental topmost 4

interface TenGigabitEthernet2/0/0
 des connect to R3
 service-policy input  MPLS-In
 service-policy output  MPLS-Out

interface TenGigabitEthernet3/0/0
 des connect to R1
 service-policy output  MPLS-Out
 service-policy input  MPLS-In

interface TenGigabitEthernet9/0/0
 des connect to R5
 service-policy output  MPLS-Out
 service-policy input  MPLS-In
!
!
interface GigabitEthernet1/1
no ip address
 shutdown
!
interface GigabitEthernet1/2
 no ip address
 shutdown
!
interface GigabitEthernet1/3
 no ip address
 shutdown
!
interface GigabitEthernet1/4
 no ip address
 shutdown
!
interface GigabitEthernet1/5
 no ip address
 shutdown
!
interface GigabitEthernet1/6
 no ip address
 shutdown
!
interface GigabitEthernet1/7
 no ip address
 shutdown
!
interface GigabitEthernet1/8
 no ip address
 shutdown
!
interface GigabitEthernet1/9
 no ip address
shutdown
```

```
!
interface GigabitEthernet1/10
 no ip address
 shutdown
!
interface GigabitEthernet1/11
 no ip address
 shutdown
!
interface GigabitEthernet1/12
 no ip address
 shutdown
!
interface GigabitEthernet1/13
 no ip address
 shutdown
!
interface GigabitEthernet1/14
 no ip address
 shutdown
!
interface GigabitEthernet1/15
 no ip address
 shutdown
!
interface GigabitEthernet1/16
 no ip address
 shutdown
!
interface GigabitEthernet1/17
 no ip address
 shutdown
!
interface GigabitEthernet1/18
 no ip address
 shutdown
!
interface GigabitEthernet1/19
 no ip address
 shutdown
!
interface GigabitEthernet1/20
 no ip address
 shutdown
!
interface GigabitEthernet1/21
 no ip address
 shutdown
!
interface GigabitEthernet1/22
 no ip address
 shutdown
!
interface GigabitEthernet1/23
 no ip address
 shutdown
!
interface GigabitEthernet1/24
 no ip address
 shutdown
!
interface TenGigabitEthernet2/0/0
 ip address 10.0.2.5 255.255.255.252
 logging event link-status
 carrier-delay msec 0
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls traffic-eng backup-path Tunnel2
 mpls ip
 ip rsvp bandwidth
!
interface TenGigabitEthernet3/0/0
 ip address 10.0.2.2 255.255.255.252
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface GigabitEthernet5/1
 no ip address
 shutdown
!
interface GigabitEthernet5/2
 ip address 192.168.1.2 255.255.255.0
 media-type rj45
 no cdp enable
!
```

```
interface GigabitEthernet6/1
 no ip address
 shutdown
!
interface GigabitEthernet6/2
 switchport
 switchport access vlan 100
 switchport mode access
 media-type rj45
 spanning-tree portfast
!
interface TenGigabitEthernet9/0/0
 ip address 10.0.2.17 255.255.255.252
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface Vlan1
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 nsf cisco
 passive-interface GigabitEthernet5/2
 network 0.0.0.0 255.255.255.255 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng multicast-intact
!
ip classless
!
!
no ip http server
no ip http secure-server
!
ip explicit-path name R2-to-R3 enable
 next-address 10.0.2.6
!
ip explicit-path name R2-to-R1-to-R3 enable
 next-address 10.0.2.1
 next-address 10.0.2.10
!
logging trap notifications
logging 192.168.1.10
logging 192.168.1.20
!
snmp-server community c1scoro RO
snmp-server community c1scorw RW
snmp-server system-shutdown
snmp-server enable traps snmp authentication linkdown
linkup coldstart warmstart
snmp-server enable traps ds1
snmp-server enable traps chassis
snmp-server enable traps module
snmp-server enable traps transceiver all
snmp-server enable traps eigrp
snmp-server enable traps casa
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change
nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change
shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change
shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps event-manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps MAC-Notification move
threshold
snmp-server enable traps msdp
```

```
snmp-server enable traps pim neighbor-change rp-
mapping-change invalid-pim-message
snmp-server enable traps rf
snmp-server enable traps rtr
snmp-server enable traps slb real virtual csrp
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-
inconsistency loop-inconsistency
snmp-server enable traps syslog
snmp-server enable traps flash insertion removal
snmp-server enable traps memory bufferpeak
snmp-server enable traps flex-links status
snmp-server enable traps csg agent quota-server database
snmp-server enable traps sonet
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps resource-policy
snmp-server enable traps ethernet cfm cc mep-up mep-
down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-
missing mep-unknown service-up
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps c6kxbar intbus-crcexcd intbus-
crcrcvrd swbus
snmp-server enable traps dot1x
snmp-server enable traps envmon fan shutdown supply
temperature status
snmp-server enable traps port-security
snmp-server enable traps mvpn
snmp-server enable traps mpls traffic-eng
snmp-server enable traps mpls fast-reroute protected
snmp-server enable traps mpls rfc ldp
snmp-server enable traps mpls ldp
snmp-server enable traps pw vc
snmp-server enable traps isakmp policy add
snmp-server enable traps isakmp policy delete
snmp-server enable traps isakmp tunnel start
snmp-server enable traps isakmp tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps alarms informational
snmp-server enable traps vlan-mac-limit
snmp-server enable traps mpls vpn
snmp-server host 192.168.1.10 version 2c c1scoro
snmp-server host 192.168.1.20 version 2c c1scoro
!
mpls ldp router-id Loopback0 force
!
control-plane
!
!
line con 0
 exec-timeout 60 0
 privilege level 15
 password c1sco
 logging synchronous
 login
line vty 0 4
 exec-timeout 60 0
 privilege level 15
 login local
 transport input lat pad udptn telnet rlogin ssh acercon
line vty 5 15
 exec-timeout 60 0
 privilege level 15
 login local
!
exception crashinfo buffersize 80
!
end
```

```
************************************************
        Configuration Provider  Router  R3
************************************************
!
no upgrade fpd auto
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service counters max age 10
!
hostname R3
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone ICT 7
ip subnet-zero
!
!
no ip domain lookup
!
!
vtp mode transparent
mls ip multicast flow-stat-timer 9
mls flow ip interface-full
no mls flow ipv6
mls qos
no mls acl tcam share-global
mls cef error action reset
mls cef maximum-routes ipv6 1
mls cef maximum-routes ip-multicast 1
multilink bundle-name authenticated
mpls traffic-eng tunnels
mpls ldp graceful-restart timers max-recovery 600
mpls ldp graceful-restart
mpls ldp session protection
mpls label protocol ldp
!
!
spanning-tree mode pvst
spanning-tree extend system-id
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
username admin privilege 15 secret 5
$1$vLbn$sChE3Ya.B5GpFvZu3UPdN1
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
vlan 100
!
!
interface Tunnel1
 ip unnumbered Loopback0
 mpls ip
 tunnel destination 10.0.0.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng path-option 1 explicit name R3-
to-R2
 tunnel mpls traffic-eng path-option 2 dynamic
 tunnel mpls traffic-eng fast-reroute
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel destination 10.0.0.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 0 0
 tunnel mpls traffic-eng path-option 1 explicit name R3-
to-R1-to-R2
!
interface Loopback0
 ip address 10.0.0.3 255.255.255.255
!
!
class-map match-all mplsexp0
 match mpls experimental 0
!
class-map match-all mplsexp5
```

```
 match mpls experimental 5
!
class-map match-all mplsexp4
 match mpls experimental 4
!

policy-map MPLS-In
 class-map mplsexp0
 set qos-group 0
!
 class-map mplsexp5
 set qos-group 5
!
 class-map mplsexp4
 set qos-group 4
!

class-map match-all qosgroup0
 match qos-group 0
!
class-map match-all qosgroup5
 match qos-group 5
!
class-map match-all qosgroup4
 match qos-group 4

policy-map MPLS-Out
!
 class qosgroup0
 set mpls experimental topmost 0
!
 class qosgroup5
 set mpls experimental topmost 5
!
 class qosgroup4
 set mpls experimental topmost 4

interface TenGigabitEthernet2/0/0
 des connect to R1
 service-policy input  MPLS-In
 service-policy output  MPLS-Out

interface TenGigabitEthernet3/0/0
 des connect to R2
 service-policy output  MPLS-Out
 service-policy input  MPLS-In

interface TenGigabitEthernet8/0/0
 des connect to R4
 service-policy output  MPLS-Out
 service-policy input  MPLS-In
!

interface GigabitEthernet1/1
 no ip address
 shutdown
!
interface GigabitEthernet1/2
 no ip address
 shutdown
!
interface GigabitEthernet1/3
 no ip address
 shutdown
!
interface GigabitEthernet1/4
 no ip address
 shutdown
!
interface GigabitEthernet1/5
 no ip address
 shutdown
!
interface GigabitEthernet1/6
 no ip address
 shutdown
!
interface GigabitEthernet1/7
 no ip address
 shutdown
!
interface GigabitEthernet1/8
 no ip address
 shutdown
!
interface GigabitEthernet1/9
 no ip address
 shutdown
```

```
!
interface GigabitEthernet1/10
 no ip address
 shutdown
!
interface GigabitEthernet1/11
 no ip address
 shutdown
!
interface GigabitEthernet1/12
 no ip address
 shutdown
!
interface GigabitEthernet1/13
 no ip address
 shutdown
!
interface GigabitEthernet1/14
 no ip address
 shutdown
!
interface GigabitEthernet1/15
 no ip address
 shutdown
!
interface GigabitEthernet1/16
 no ip address
 shutdown
!
interface GigabitEthernet1/17
 no ip address
 shutdown
!
interface GigabitEthernet1/18
 no ip address
 shutdown
!
interface GigabitEthernet1/19
 no ip address
 shutdown
!
interface GigabitEthernet1/20
 no ip address
 shutdown
!
interface GigabitEthernet1/21
 no ip address
 shutdown
!
interface GigabitEthernet1/22
 no ip address
 shutdown
!
interface GigabitEthernet1/23
 no ip address
 shutdown
!
interface GigabitEthernet1/24
 no ip address
 shutdown
!
interface TenGigabitEthernet2/0/0
 ip address 10.0.2.10 255.255.255.252
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface TenGigabitEthernet3/0/0
 ip address 10.0.2.6 255.255.255.252
 logging event link-status
 carrier-delay msec 0
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls traffic-eng backup-path Tunnel2
 mpls ip
 ip rsvp bandwidth
!
interface GigabitEthernet5/1
 ip address 192.168.1.3 255.255.255.0
 no cdp enable
!
interface GigabitEthernet5/2
 switchport
 switchport access vlan 100
 switchport mode access
 spanning-tree portfast
```

```
 media-type rj45
!
interface TenGigabitEthernet8/0/0
 ip address 10.0.2.25 255.255.255.252
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface Vlan1
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 nsf cisco
 passive-interface GigabitEthernet5/2
 network 0.0.0.0 255.255.255.255 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng multicast-intact
!
ip classless
!
!
no ip http server
no ip http secure-server
!
ip explicit-path name R3-to-R2 enable
 next-address 10.0.2.5
!
ip explicit-path name R3-to-R1-to-R2 enable
 next-address 10.0.2.9
 next-address 10.0.2.2
!
logging trap notifications
logging 192.168.1.10
logging 192.168.1.20
!
snmp-server community c1scoro RO
snmp-server community c1scorw RW
snmp-server system-shutdown
snmp-server enable traps snmp authentication linkdown
linkup coldstart warmstart
snmp-server enable traps ds1
snmp-server enable traps chassis
snmp-server enable traps module
snmp-server enable traps transceiver all
snmp-server enable traps eigrp
snmp-server enable traps casa
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change
nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change
shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change
shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps event-manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps MAC-Notification move
threshold
snmp-server enable traps msdp
snmp-server enable traps pim neighbor-change rp-
mapping-change invalid-pim-message
snmp-server enable traps rf
snmp-server enable traps rtr
snmp-server enable traps slb real virtual csrp
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-
inconsistency loop-inconsistency
snmp-server enable traps syslog
```

```
snmp-server enable traps flash insertion removal
snmp-server enable traps memory bufferpeak
snmp-server enable traps flex-links status
snmp-server enable traps csg agent quota-server database
snmp-server enable traps sonet
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps resource-policy
snmp-server enable traps ethernet cfm cc mep-up mep-
down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-
missing mep-unknown service-up
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps c6kxbar intbus-crcexcd intbus-
crcrcvrd swbus
snmp-server enable traps dot1x
snmp-server enable traps envmon fan shutdown supply
temperature status
snmp-server enable traps port-security
snmp-server enable traps mvpn
snmp-server enable traps mpls traffic-eng
snmp-server enable traps mpls fast-reroute protected
snmp-server enable traps mpls rfc ldp
snmp-server enable traps mpls ldp
snmp-server enable traps pw vc
snmp-server enable traps isakmp policy add
snmp-server enable traps isakmp policy delete
snmp-server enable traps isakmp tunnel start
snmp-server enable traps isakmp tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps alarms informational
snmp-server enable traps vlan-mac-limit
snmp-server enable traps mpls vpn
snmp-server host 192.168.1.10 version 2c c1scoro
snmp-server host 192.168.1.20 version 2c c1scoro
!
mpls ldp router-id Loopback0 force
!
control-plane
!
!
line con 0
 exec-timeout 60 0
 privilege level 15
 password c1sco
 logging synchronous
 login
line vty 0 4
 exec-timeout 60 0
 privilege level 15
 login local
 transport input lat pad udptn telnet rlogin ssh acercon
line vty 5 15
 exec-timeout 60 0
 privilege level 15
 login local
!
exception crashinfo buffersize 80
!
end
```

```
**********************************************
      Configuration Provider Edge  Router  R4
**********************************************
!
no upgrade fpd auto
version 12.2
service timestamps debug datetime localtime
service timestamps log datetime msec
no service password-encryption
service counters max age 10
!
hostname R4
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone ICT 7
ip subnet-zero
ip vrf vpn501
 rd 501:1
 route-target export 501:1
 route-target import 501:1
!
ip vrf vpn502
 rd 502:1
 route-target export 502:1
 route-target import 502:1
!
ip vrf vpn503
 rd 503:1
 route-target export 503:1
 route-target import 503:1
!
ip vrf vpn504
 rd 504:1
 route-target export 504:1
 route-target import 504:1
!
ip vrf vpn505
 rd 505:1
 route-target export 505:1
 route-target import 505:1
!
ip vrf vpn506
 rd 506:1
 route-target export 506:1
 route-target import 506:1
!
ip vrf vpn507
 rd 507:1
 route-target export 507:1
 route-target import 507:1
!
ip vrf vpn508
 rd 508:1
 route-target export 508:1
 route-target import 508:1
!
ip vrf vpn509
 rd 509:1
 route-target export 509:1
 route-target import 509:1
!
ip vrf vpn510
 rd 510:1
 route-target export 510:1
 route-target import 510:1
!
no ip domain lookup
!
!
vtp mode transparent
mls ip multicast flow-stat-timer 9
no mls flow ip
no mls flow ipv6
mls qos
no mls acl tcam share-global
mls cef error action reset
mls cef maximum-routes ipv6 1
mls cef maximum-routes ip-multicast 1
multilink bundle-name authenticated
mpls traffic-eng tunnels
mpls ldp graceful-restart timers max-recovery 600
mpls ldp graceful-restart
mpls ldp session protection
mpls label protocol ldp
```

```
!
!
spanning-tree mode pvst
spanning-tree extend system-id
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
username admin privilege 15 secret 5
$1$vLbn$sChE3Ya.B5GpFvZu3UPdN1
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
vlan 99,501-900,2001-3000
!
l2 vfi vpls2001 manual
 vpn id 2001
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2002 manual
 vpn id 2002
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2003 manual
 vpn id 2003
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2004 manual
 vpn id 2004
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2005 manual
 vpn id 2005
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2006 manual
 vpn id 2006
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2007 manual
 vpn id 2007
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2008 manual
 vpn id 2008
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2009 manual
 vpn id 2009
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
l2 vfi vpls2010 manual
 vpn id 2010
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.5 encapsulation mpls
!
!
interface Loopback0
 ip address 10.0.0.4 255.255.255.255
!
interface GigabitEthernet1/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2001-3000
 switchport mode trunk
 mls qos trust cos
!
interface GigabitEthernet1/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 99,501-900
 switchport mode trunk
!
interface GigabitEthernet1/3
 no ip address
```

```
 shutdown
!
interface GigabitEthernet1/4
 no ip address
 shutdown
!
interface GigabitEthernet1/5
 no ip address
 shutdown
!
interface GigabitEthernet1/6
 no ip address
 shutdown
!
interface GigabitEthernet1/7
 no ip address
 shutdown
!
interface GigabitEthernet1/8
 no ip address
 shutdown
!
interface GigabitEthernet1/9
 no ip address
 shutdown
!
interface GigabitEthernet1/10
 no ip address
 shutdown
!
interface GigabitEthernet1/11
 no ip address
 shutdown
!
interface GigabitEthernet1/12
 no ip address
 shutdown
!
interface GigabitEthernet1/13
 no ip address
 shutdown
!
interface GigabitEthernet1/14
 no ip address
 shutdown
!
interface GigabitEthernet1/15
 no ip address
 shutdown
!
interface GigabitEthernet1/16
 no ip address
 shutdown
!
interface GigabitEthernet1/17
 no ip address
 shutdown
!
interface GigabitEthernet1/18
 no ip address
 shutdown
!
interface GigabitEthernet1/19
 no ip address
 shutdown
!
interface GigabitEthernet1/20
 no ip address
 shutdown
!
interface GigabitEthernet1/21
 no ip address
 shutdown
!
interface GigabitEthernet1/22
 no ip address
 shutdown
!
interface GigabitEthernet1/23
 no ip address
 shutdown
!
interface GigabitEthernet1/24
 no ip address
 shutdown
!
interface TenGigabitEthernet2/0/0
 ip address 10.0.2.26 255.255.255.252
```

```
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
!
interface GigabitEthernet5/1
 no ip address
 shutdown
!
interface GigabitEthernet5/2
 ip address 192.168.1.4 255.255.255.0
 media-type rj45
 no cdp enable
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan501
 ip vrf forwarding vpn501
 ip address 192.10.0.1 255.255.255.0
!
interface Vlan502
 ip vrf forwarding vpn502
 ip address 192.10.1.1 255.255.255.0
!
interface Vlan503
 ip vrf forwarding vpn503
 ip address 192.10.2.1 255.255.255.0
!
interface Vlan504
 ip vrf forwarding vpn504
 ip address 192.10.3.1 255.255.255.0
!
interface Vlan505
 ip vrf forwarding vpn505
 ip address 192.10.4.1 255.255.255.0
!
interface Vlan506
 ip vrf forwarding vpn506
 ip address 192.10.5.1 255.255.255.0
!
interface Vlan507
 ip vrf forwarding vpn507
 ip address 192.10.6.1 255.255.255.0
!
interface Vlan508
 ip vrf forwarding vpn508
 ip address 192.10.7.1 255.255.255.0
!
interface Vlan509
 ip vrf forwarding vpn509
 ip address 192.10.8.1 255.255.255.0
!
interface Vlan510
 ip vrf forwarding vpn510
 ip address 192.10.9.1 255.255.255.0
!
!
interface Vlan2001
 no ip address
 xconnect vfi vpls2001
!
interface Vlan2002
 no ip address
 xconnect vfi vpls2002
!
interface Vlan2003
 no ip address
 xconnect vfi vpls2003
!
interface Vlan2004
 no ip address
 xconnect vfi vpls2004
!
interface Vlan2005
 no ip address
 xconnect vfi vpls2005
!
interface Vlan2006
 no ip address
 xconnect vfi vpls2006
!
interface Vlan2007
 no ip address
 xconnect vfi vpls2007
!
interface Vlan2008
```

```
 no ip address
 xconnect vfi vpls2008
!
interface Vlan2009
 no ip address
 xconnect vfi vpls2009
!
interface Vlan2010
 no ip address
 xconnect vfi vpls2010
!
class-map match-all Data-Traffic
 match dscp default
 !
class-map match-all Voice-Traffic
 match dscp ef
 !
class-map match-any Video-Traffic
 match dscp af31 af41
 !

policy-map Ingress-Policy
 class Data-Traffic
  set qos-group 0
!
 class Voice-Traffic
  set qos-group 5
!
 class Video-Traffic
  set qos-group 4
!

class-map match-all qosgroup0
  match qos-group 0
!
class-map match-all qosgroup5
  match qos-group 5
!
class-map match-all qosgroup4
  match qos-group 4
!

policy-map Egree-Policy
 class qosgroup0
   set mpls experimental topmost 0
   police 500000000
   conform-action transmit
   exceed-action drop
!
 class qosgroup5
   set mpls experimental topmost 5
   police 250000000
   conform-action transmit
   exceed-action drop
!
 class qosgroup4
   set mpls experimental topmost 4
   police 250000000
   conform-action transmit
   exceed-action drop
!
 interface GigabitEthernet1/1
 des conntect to CE
 service-policy input Ingress-Policy
 !
 interface TenGigabitEthernet2/0/0
 des connect to R3
 service-policy output Egree-Policy
 !
 !
router ospf 1
 log-adjacency-changes
 nsf cisco
 passive-interface GigabitEthernet5/2
 network 0.0.0.0 255.255.255.255 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng multicast-intact
 !
router bgp 1
 bgp router-id 10.0.0.4
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 update-source Loopback0
```

```
neighbor 10.0.0.5 remote-as 1
neighbor 10.0.0.5 update-source Loopback0
!
address-family ipv4
 no synchronization
 no auto-summary
exit-address-family
!
address-family vpnv4
 neighbor 10.0.0.1 activate
 neighbor 10.0.0.1 send-community extended
 neighbor 10.0.0.1 next-hop-self
 neighbor 10.0.0.5 activate
 neighbor 10.0.0.5 send-community extended
 neighbor 10.0.0.5 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn501
 no synchronization
 redistribute connected
 neighbor 192.10.0.2 remote-as 1000
 neighbor 192.10.0.2 activate
 neighbor 192.10.0.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn502
 no synchronization
 redistribute connected
 neighbor 192.10.1.2 remote-as 1000
 neighbor 192.10.1.2 activate
 neighbor 192.10.1.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn503
 no synchronization
 redistribute connected
 neighbor 192.10.2.2 remote-as 1000
 neighbor 192.10.2.2 activate
 neighbor 192.10.2.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn504
 no synchronization
 redistribute connected
 neighbor 192.10.3.2 remote-as 1000
 neighbor 192.10.3.2 activate
 neighbor 192.10.3.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn505
 no synchronization
 redistribute connected
 neighbor 192.10.4.2 remote-as 1000
 neighbor 192.10.4.2 activate
 neighbor 192.10.4.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn506
 no synchronization
 redistribute connected
 neighbor 192.10.5.2 remote-as 1000
 neighbor 192.10.5.2 activate
 neighbor 192.10.5.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn507
 no synchronization
 redistribute connected
 neighbor 192.10.6.2 remote-as 1000
 neighbor 192.10.6.2 activate
 neighbor 192.10.6.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn508
 no synchronization
 redistribute connected
 neighbor 192.10.7.2 remote-as 1000
 neighbor 192.10.7.2 activate
 neighbor 192.10.7.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn509
 no synchronization
 redistribute connected
 neighbor 192.10.8.2 remote-as 1000
 neighbor 192.10.8.2 activate
 neighbor 192.10.8.2 next-hop-self
exit-address-family
```

```
!
address-family ipv4 vrf vpn510
 no synchronization
 redistribute connected
 neighbor 192.10.9.2 remote-as 1000
 neighbor 192.10.9.2 activate
 neighbor 192.10.9.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn511
 no synchronization
 redistribute connected
 neighbor 192.10.10.2 remote-as 1000
 neighbor 192.10.10.2 activate
 neighbor 192.10.10.2 next-hop-self
exit-address-family
!
ip classless
!
!
no ip http server
no ip http secure-server
!
logging trap notifications
logging 192.168.1.10
logging 192.168.1.20
!
snmp-server community c1scoro RO
snmp-server community c1scorw RW
snmp-server system-shutdown
snmp-server enable traps snmp authentication linkdown
linkup coldstart warmstart
snmp-server enable traps ds1
snmp-server enable traps chassis
snmp-server enable traps module
snmp-server enable traps transceiver all
snmp-server enable traps eigrp
snmp-server enable traps casa
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change
nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change
shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change
shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps event-manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps MAC-Notification move
threshold
snmp-server enable traps msdp
snmp-server enable traps pim neighbor-change rp-
mapping-change invalid-pim-message
snmp-server enable traps rf
snmp-server enable traps rtr
snmp-server enable traps slb real virtual csrp
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-
inconsistency loop-inconsistency
snmp-server enable traps syslog
snmp-server enable traps flash insertion removal
snmp-server enable traps memory bufferpeak
snmp-server enable traps flex-links status
snmp-server enable traps csg agent quota-server database
snmp-server enable traps sonet
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps resource-policy
snmp-server enable traps ethernet cfm cc mep-up mep-
down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-
missing mep-unknown service-up
snmp-server enable traps cpu threshold
```

```
snmp-server enable traps rsvp
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps c6kxbar intbus-crcexcd intbus-
crcrcvrd swbus
snmp-server enable traps dot1x
snmp-server enable traps envmon fan shutdown supply
temperature status
snmp-server enable traps port-security
snmp-server enable traps mvpn
snmp-server enable traps mpls traffic-eng
snmp-server enable traps mpls fast-reroute protected
snmp-server enable traps mpls rfc ldp
snmp-server enable traps mpls ldp
snmp-server enable traps pw vc
snmp-server enable traps isakmp policy add
snmp-server enable traps isakmp policy delete
snmp-server enable traps isakmp tunnel start
snmp-server enable traps isakmp tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps alarms informational
snmp-server enable traps vlan-mac-limit
snmp-server enable traps mpls vpn
snmp-server host 192.168.1.10 version 2c c1scoro
snmp-server host 192.168.1.20 version 2c c1scoro
!
mpls ldp router-id Loopback0 force
!
control-plane
!
line con 0
 exec-timeout 60 0
 privilege level 15
 password c1sco
 logging synchronous
 login
line vty 0 4
 exec-timeout 60 0
 privilege level 15
 login local
 transport input lat pad udptn telnet rlogin ssh acercon
line vty 5 15
 exec-timeout 60 0
 privilege level 15
 login local
!
exception crashinfo buffersize 80
mac-address-table aging-time 600
!
end
```

```
**********************************************
        Configuration Provider Edge  Router  R5
**********************************************
!
no upgrade fpd auto
version 12.2
service timestamps debug datetime localtime
service timestamps log datetime msec
no service password-encryption
service counters max age 10
!
hostname R5
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone ICT 7
ip subnet-zero
ip vrf vpn501
 rd 501:1
 route-target export 501:1
 route-target import 501:1
!
ip vrf vpn502
 rd 502:1
 route-target export 502:1
 route-target import 502:1
!
ip vrf vpn503
 rd 503:1
 route-target export 503:1
 route-target import 503:1
!
ip vrf vpn504
 rd 504:1
 route-target export 504:1
 route-target import 504:1
!
ip vrf vpn505
 rd 505:1
 route-target export 505:1
 route-target import 505:1
!
ip vrf vpn506
 rd 506:1
 route-target export 506:1
 route-target import 506:1
!
ip vrf vpn507
 rd 507:1
 route-target export 507:1
 route-target import 507:1
!
ip vrf vpn508
 rd 508:1
 route-target export 508:1
 route-target import 508:1
!
ip vrf vpn509
 rd 509:1
 route-target export 509:1
 route-target import 509:1
!
ip vrf vpn510
 rd 510:1
 route-target export 510:1
 route-target import 510:1
!
no ip domain lookup
!
!
vtp mode transparent
mls ip multicast flow-stat-timer 9
no mls flow ip
no mls flow ipv6
mls qos
no mls acl tcam share-global
mls cef error action reset
mls cef maximum-routes ipv6 1
mls cef maximum-routes ip-multicast 1
multilink bundle-name authenticated
mpls traffic-eng tunnels
mpls ldp graceful-restart timers max-recovery 600
mpls ldp graceful-restart
mpls ldp session protection
mpls label protocol ldp
```

```
!
!
spanning-tree mode pvst
spanning-tree extend system-id
diagnostic cns publish cisco.cns.device.diag_results
diagnostic cns subscribe cisco.cns.device.diag_commands
username admin privilege 15 secret 5
$1$vLbn$sChE3Ya.B5GpFvZu3UPdN1
!
redundancy
 main-cpu
  auto-sync running-config
 mode sso
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
vlan 99,501-900,2001-3000
!
l2 vfi vpls2001 manual
 vpn id 2001
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2002 manual
 vpn id 2002
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2003 manual
 vpn id 2003
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2004 manual
 vpn id 2004
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2005 manual
 vpn id 2005
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2006 manual
 vpn id 2006
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2007 manual
 vpn id 2007
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2008 manual
 vpn id 2008
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2009 manual
 vpn id 2009
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
l2 vfi vpls2010 manual
 vpn id 2010
 neighbor 10.0.0.1 encapsulation mpls
 neighbor 10.0.0.4 encapsulation mpls
!
!
interface Loopback0
 ip address 10.0.0.5 255.255.255.255
!
interface GigabitEthernet1/1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2001-3000
 switchport mode trunk
 mls qos trust cos
!
interface GigabitEthernet1/2
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 99,501-900
 switchport mode trunk
!
interface GigabitEthernet1/3
 no ip address
```

```
 shutdown
!
interface GigabitEthernet1/4
 no ip address
 shutdown
!
interface GigabitEthernet1/5
 no ip address
 shutdown
!
interface GigabitEthernet1/6
 no ip address
 shutdown
!
interface GigabitEthernet1/7
 no ip address
 shutdown
!
interface GigabitEthernet1/8
 no ip address
 shutdown
!
interface GigabitEthernet1/9
 no ip address
 shutdown
!
interface GigabitEthernet1/10
 no ip address
 shutdown
!
interface GigabitEthernet1/11
 no ip address
 shutdown
!
interface GigabitEthernet1/12
 no ip address
 shutdown
!
interface GigabitEthernet1/13
 no ip address
 shutdown
!
interface GigabitEthernet1/14
 no ip address
 shutdown
!
interface GigabitEthernet1/15
 no ip address
 shutdown
!
interface GigabitEthernet1/16
 no ip address
 shutdown
!
interface GigabitEthernet1/17
 no ip address
 shutdown
!
interface GigabitEthernet1/18
 no ip address
 shutdown
!
interface GigabitEthernet1/19
 no ip address
 shutdown
!
interface GigabitEthernet1/20
 no ip address
 shutdown
!
interface GigabitEthernet1/21
 no ip address
 shutdown
!
interface GigabitEthernet1/22
 no ip address
 shutdown
!
interface GigabitEthernet1/23
 no ip address
 shutdown
!
interface GigabitEthernet1/24
 no ip address
 shutdown
!
interface TenGigabitEthernet2/0/0
 ip address 10.0.2.18 255.255.255.252
```

```
 logging event link-status
 mls qos trust dscp
 mpls traffic-eng tunnels
 mpls ip
!
interface GigabitEthernet5/1
 no ip address
 shutdown
!
interface GigabitEthernet5/2
 ip address 192.168.1.5 255.255.255.0
 media-type rj45
 no cdp enable
!
interface Vlan1
 no ip address
 shutdown
!
interface Vlan501
 ip vrf forwarding vpn501
 ip address 192.20.0.1 255.255.255.0
!
interface Vlan502
 ip vrf forwarding vpn502
 ip address 192.20.1.1 255.255.255.0
!
interface Vlan503
 ip vrf forwarding vpn503
 ip address 192.20.2.1 255.255.255.0
!
interface Vlan504
 ip vrf forwarding vpn504
 ip address 192.20.3.1 255.255.255.0
!
interface Vlan505
 ip vrf forwarding vpn505
 ip address 192.20.4.1 255.255.255.0
!
interface Vlan506
 ip vrf forwarding vpn506
 ip address 192.20.5.1 255.255.255.0
!
interface Vlan507
 ip vrf forwarding vpn507
 ip address 192.20.6.1 255.255.255.0
!
interface Vlan508
 ip vrf forwarding vpn508
 ip address 192.20.7.1 255.255.255.0
!
interface Vlan509
 ip vrf forwarding vpn509
 ip address 192.20.8.1 255.255.255.0
!
interface Vlan510
 ip vrf forwarding vpn510
 ip address 192.20.9.1 255.255.255.0
!
!
interface Vlan2001
 no ip address
 xconnect vfi vpls2001
!
interface Vlan2002
 no ip address
 xconnect vfi vpls2002
!
interface Vlan2003
 no ip address
 xconnect vfi vpls2003
!
interface Vlan2004
 no ip address
 xconnect vfi vpls2004
!
interface Vlan2005
 no ip address
 xconnect vfi vpls2005
!
interface Vlan2006
 no ip address
 xconnect vfi vpls2006
!
interface Vlan2007
 no ip address
 xconnect vfi vpls2007
!
interface Vlan2008
```

```
 no ip address
 xconnect vfi vpls2008
!
interface Vlan2009
 no ip address
 xconnect vfi vpls2009
!
interface Vlan2010
 no ip address
 xconnect vfi vpls2010
!
class-map match-all Data-Traffic
 match dscp default
 !
class-map match-all Voice-Traffic
 match dscp ef
 !
class-map match-any Video-Traffic
 match dscp af31 af41
 !

policy-map Ingress-Policy
 class Data-Traffic
  set qos-group 0
!
 class Voice-Traffic
  set qos-group 5
!
 class Video-Traffic
  set qos-group 4
!

class-map match-all qosgroup0
  match qos-group 0
!
class-map match-all qosgroup5
  match qos-group 5
!
class-map match-all qosgroup4
  match qos-group 4
!
policy-map Egree-Policy
 class qosgroup0
    set mpls experimental topmost 0
    police 500000000
    conform-action transmit
    exceed-action drop
!
 class qosgroup5
    set mpls experimental topmost 5
    police 250000000
    conform-action transmit
    exceed-action drop
!
 class qosgroup4
    set mpls experimental topmost 4
    police 250000000
    conform-action transmit
    exceed-action drop
!
 interface g 0/1
 des conntect to CE1
 service-policy input Ingress-Policy
!
 interface TenGigabitEthernet2/0/0
 des connect to R2
 service-policy output Egree-Policy
!
i
router ospf 1
 log-adjacency-changes
 nsf cisco
 passive-interface GigabitEthernet5/2
 network 0.0.0.0 255.255.255.255 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng multicast-intact
!
router bgp 1
 bgp router-id 10.0.0.5
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 bgp graceful-restart restart-time 120
 bgp graceful-restart stalepath-time 360
 bgp graceful-restart
 neighbor 10.0.0.1 remote-as 1
 neighbor 10.0.0.1 update-source Loopback0
 neighbor 10.0.0.4 remote-as 1
```

```
neighbor 10.0.0.4 update-source Loopback0
!
address-family ipv4
 no synchronization
 no auto-summary
exit-address-family
!
address-family vpnv4
 neighbor 10.0.0.1 activate
 neighbor 10.0.0.1 send-community extended
 neighbor 10.0.0.1 next-hop-self
 neighbor 10.0.0.4 activate
 neighbor 10.0.0.4 send-community extended
 neighbor 10.0.0.4 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn501
 no synchronization
 redistribute connected
 neighbor 192.20.0.2 remote-as 1000
 neighbor 192.20.0.2 activate
 neighbor 192.20.0.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn502
 no synchronization
 redistribute connected
 neighbor 192.20.1.2 remote-as 1000
 neighbor 192.20.1.2 activate
 neighbor 192.20.1.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn503
 no synchronization
 redistribute connected
 neighbor 192.20.2.2 remote-as 1000
 neighbor 192.20.2.2 activate
 neighbor 192.20.2.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn504
 no synchronization
 redistribute connected
 neighbor 192.20.3.2 remote-as 1000
 neighbor 192.20.3.2 activate
 neighbor 192.20.3.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn505
 no synchronization
 redistribute connected
 neighbor 192.20.4.2 remote-as 1000
 neighbor 192.20.4.2 activate
 neighbor 192.20.4.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn506
 no synchronization
 redistribute connected
 neighbor 192.20.5.2 remote-as 1000
 neighbor 192.20.5.2 activate
 neighbor 192.20.5.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn507
 no synchronization
 redistribute connected
 neighbor 192.20.6.2 remote-as 1000
 neighbor 192.20.6.2 activate
 neighbor 192.20.6.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn508
 no synchronization
 redistribute connected
 neighbor 192.20.7.2 remote-as 1000
 neighbor 192.20.7.2 activate
 neighbor 192.20.7.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn509
 no synchronization
 redistribute connected
 neighbor 192.20.8.2 remote-as 1000
 neighbor 192.20.8.2 activate
 neighbor 192.20.8.2 next-hop-self
exit-address-family
!
```

```
address-family ipv4 vrf vpn510
 no synchronization
 redistribute connected
 neighbor 192.20.9.2 remote-as 1000
 neighbor 192.20.9.2 activate
 neighbor 192.20.9.2 next-hop-self
exit-address-family
!
address-family ipv4 vrf vpn511
 no synchronization
 redistribute connected
 neighbor 192.20.10.2 remote-as 1000
 neighbor 192.20.10.2 activate
 neighbor 192.20.10.2 next-hop-self
exit-address-family
!
ip classless
!
!
no ip http server
no ip http secure-server
!
logging trap notifications
logging 192.168.1.10
logging 192.168.1.20
!
snmp-server community c1scoro RO
snmp-server community c1scorw RW
snmp-server system-shutdown
snmp-server enable traps snmp authentication linkdown
linkup coldstart warmstart
snmp-server enable traps ds1
snmp-server enable traps chassis
snmp-server enable traps module
snmp-server enable traps transceiver all
snmp-server enable traps eigrp
snmp-server enable traps casa
snmp-server enable traps tty
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change
nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change
shamlink interface-old
snmp-server enable traps ospf cisco-specific state-change
shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps atm subif
snmp-server enable traps bgp
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps event-manager
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps ipmulticast
snmp-server enable traps MAC-Notification move
threshold
snmp-server enable traps msdp
snmp-server enable traps pim neighbor-change rp-
mapping-change invalid-pim-message
snmp-server enable traps rf
snmp-server enable traps rtr
snmp-server enable traps slb real virtual csrp
snmp-server enable traps bridge newroot topologychange
snmp-server enable traps stpx inconsistency root-
inconsistency loop-inconsistency
snmp-server enable traps syslog
snmp-server enable traps flash insertion removal
snmp-server enable traps memory bufferpeak
snmp-server enable traps flex-links status
snmp-server enable traps csg agent quota-server database
snmp-server enable traps sonet
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps resource-policy
snmp-server enable traps ethernet cfm cc mep-up mep-
down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-
missing mep-unknown service-up
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps vtp
```

```
snmp-server enable traps vlancreate
snmp-server enable traps vlandelete
snmp-server enable traps c6kxbar intbus-crcexcd intbus-
crcrcvrd swbus
snmp-server enable traps dot1x
snmp-server enable traps envmon fan shutdown supply
temperature status
snmp-server enable traps port-security
snmp-server enable traps mvpn
snmp-server enable traps mpls traffic-eng
snmp-server enable traps mpls fast-reroute protected
snmp-server enable traps mpls rfc ldp
snmp-server enable traps mpls ldp
snmp-server enable traps pw vc
snmp-server enable traps isakmp policy add
snmp-server enable traps isakmp policy delete
snmp-server enable traps isakmp tunnel start
snmp-server enable traps isakmp tunnel stop
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps alarms informational
snmp-server enable traps vlan-mac-limit
snmp-server enable traps mpls vpn
snmp-server host 192.168.1.10 version 2c c1scoro
snmp-server host 192.168.1.20 version 2c c1scoro
!
mpls ldp router-id Loopback0 force
!
control-plane
!
line con 0
 exec-timeout 60 0
 privilege level 15
 password c1sco
 logging synchronous
 login
line vty 0 4
 exec-timeout 60 0
 privilege level 15
 login local
 transport input lat pad udptn telnet rlogin ssh acercon
line vty 5 15
 exec-timeout 60 0
 privilege level 15
 login local
!
exception crashinfo buffersize 80
mac-address-table aging-time 600
!
end
```

ภาคผนวก ข.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. บดินทร์ จิ๋วแย้ม, กอบชัย เดชหาญ, **"การหาประสิทธิภาพบนโครงข่าย MPLS/VPN เปรียบเทียบกับโครงข่าย IP แบบดั้งเดิม,"** วิศวสารลาดกระบัง, ปีที่ 26 ฉบับที่ 3, กันยายน 2552.

การหาประสิทธิภาพบนโครงข่าย MPLS/VPN เปรียบเทียบกับโครงข่าย IP แบบดั้งเดิม

Performance Evaluation of MPLS/VPN network versus Traditional IP network

บดินทร์ จิ๋วแย้ม กอบชัย เดชหาญ
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

บทความนี้ได้ทำการศึกษาและวิเคราะห์เปรียบเทียบประสิทธิภาพระหว่างโครงข่าย MPLS/VPN กับโครงข่าย IP แบบดั้งเดิม ในด้านความน่าเชื่อถือของระบบ (Reliability) และความสามารถในด้านคุณภาพการบริการ (Quality of Service) โดยใช้เทคนิคการทำ MPLS Traffic Engineering (TE) Fast Reroute และ MPLS Differentiated Service (DiffServ) แทนการทำงานของ OSPF Routing protocol และ Best Effort โดยทำการทดสอบบนโครงข่ายจริงที่ระดับความเร็ว 10 Gbps

คำสำคัญ: MPLS, VPN, ทราฟฟิค เอ็นจิเนียริ่ง

Abstract

This paper studies and analyzes a comparative performance of MPLS/VPN network versus traditional IP network such as Reliability and Quality of Service by means of MPLS Traffic Engineering (TE). Fast Reroute and MPLS Differentiated Service (DiffServ) will replace OSPF Routing protocol and Best Effort by test on real network of 10 Gbps

Key words: MPLS, VPN, Traffic Engineering

1. บทนำ

Multiprotocol Label Switching (MPLS) เป็นเทคโนโลยีที่ขยายความสามารถของสถาปัตยกรรม Internet protocol ซึ่งมีความสามารถในการรองรับการให้บริการข้อมูลแบบ multi traffic voice data และ video โดยเพิ่มความสามารถใหม่ๆ เช่น Virtual Private Network (VPN) Quality of Service (QoS) และ Traffic Engineering (TE) [1]

MPLS เป็นเทคโนโลยีที่เริ่มมีการใช้งานกันอย่างกว้างขวางทั้งใน ผู้ให้บริการอินเตอร์เน็ต (Internet Service Providers : ISP),ผู้ให้บริการคมนาคมขนาดใหญ่ (telecommunication carriers) และองค์กรชั้นนำทั่วไปซึ่ง MPLS เป็นเทคโนโลยีที่นำมาแก้ปัญหาที่เกิดขึ้นในปัจจุบันของระบบเครือข่ายเช่น ความเร็ว (speed) ขนาด (scalability) การบริหารคุณภาพการให้บริการ (quality of service management) และการควบคุมการจราจร (traffic control) ผู้ให้บริการอินเตอร์เน็ต (ISP),telecommunication

carriers รวมทั้งองค์กรชั้นนำทั่วไปได้นำเอาเทคโนโลยี MPLS มาประยุกต์ใช้งานในด้านต่างๆ เช่น โครงข่ายเสมือนส่วนบุคคล (Virtual Private Network :VPN) Traffic Engineering และการควบคุมคุณภาพการให้บริการ (Quality of Service: QoS) เพื่อรับประกันคุณภาพการให้บริการสำหรับข้อมูลประเภท voice video และ application ที่ต้องการความมีเสถียรภาพของข้อมูล ดังที่ได้กล่าวมาข้างต้นจึงเห็นได้ว่า MPLS เหมาะแก่การนำมาใช้เป็นเครือข่ายหลักสำหรับการ สื่อสาร Internet Protocol (IP) [5] [6]

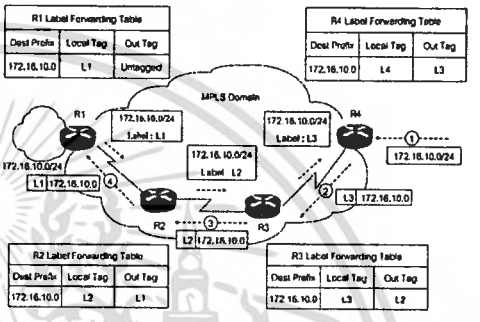
Virtual Private Network (VPN) ได้เริ่มเป็นที่รู้จักและนำมาใช้งานในลักษณะที่เรียกว่าวงจรเช่า (leased line) ให้บริการในลักษณะ point-to-point ระหว่างสำนักงานของผู้ใช้บริการโดยผ่านเครือข่ายของผู้ให้บริการ (service provider:SP) โดย Frame Relay และ ATM เป็นเทคโนโลยีแรกที่นำมาใช้เพื่อให้บริการ VPN [7] ซึ่งหลังจากได้มีการนำเทคโนโลยี MPLS มาใช้งานการให้บริการ VPN แบบใหม่ก็ได้เกิดขึ้นโดย MPLS-based VPN สามารถแบ่งได้เป็น 3 ลักษณะคือ [1]

- Layer 3 multipoint VPNs หรือ Internet Protocol VPNs
- Layer 2 point-to-point VPNs
- Layer 2 multipoint VPNs

2. ทฤษฎีที่ใช้ในการทดสอบ

2.1.MPLS Network

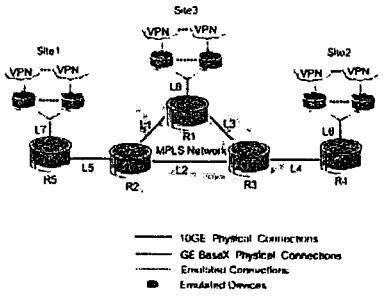
ในระบบเครือข่าย MPLS ข้อมูลที่ส่งจะถูกเพิ่ม Labels โดย Labels นี้เป็นลักษณะเช่นเดียวกันกับ IP address ปลายทาง ค่าของ Labels จะกำหนดบน Router และในบางกรณี Labels จะกำหนด โดยอ้างอิงจาก Interface บน Router ซึ่ง Router จะกำหนด Labels และกำหนดเส้นทางที่เรียกว่า Label Switch Paths (LSP) ระหว่าง ต้นทางไปยังปลายทาง รูปที่ 1 แสดงการทำงานของ MPLS forwarding เริ่มจาก PE Router R1 และ R4 Routers ในเครือข่าย MPLS R1,R2,R3 ประกาศ update เครือข่าย 172.16.10.0/24 ผ่าน IGP Routing protocol ไปในเส้นทางเดิมของระบบเครือข่าย IP สมมุติว่าไม่มีการกำหนด filters หรือ Summarization Router ก็จะทำการสร้างตาราง IP Forwarding เส้นทางที่เชื่อมต่อไปยัง Router MPLS และกำหนด Local Labels สำหรับเครือข่ายปลายทาง 172.16.10.0 โดยทำการเผยแพร่ Labels ของเครือข่ายปลายทาง 172.16.10.0 ให้ Router ข้างเคียงทราบไปทาง Upstream ด้วย Labels distribution protocols ตัวอย่างเช่น R1 กำหนด Local Labels เป็น L1 และเผยแพร่ไป Upstream ให้ R2 และ R2 ส่งต่อให้ R3 กำหนดให้เผยแพร่เหมือนกันไปทาง Upstream คือ R4 ซึ่งกระบวนการนี้ทำให้ Router สามารถสร้าง label forwarding table เพื่อใช้ในการส่งต่อ Labels packet [1] [2]



รูปที่ 1 การส่งข้อมูลในโครงข่าย MPLS

3. การทดสอบการทำงาน

บทความนี้จะทำการทดสอบเปรียบเทียบประสิทธิภาพของโครงข่าย MPLS/VPN กับโครงข่าย Traditional IP รูปที่ 2 ประกอบด้วย Core Router (R2,R3) ทำหน้าที่เป็น P Router, Distributed Router (R1,R4, R5) ทำหน้าที่เป็น PE Router เชื่อมต่อกันด้วย 10 Gigabit Ethernet interface Router ทั้งหมดถูก configure เพื่อให้บริการ MPLS โดยเชื่อมต่อกับ Traffic Generator ด้วย Gigabit Ethernet interface ซึ่งจำลองเป็นอุปกรณ์ CE (Customer Equipment) ส่งข้อมูล Ethernet ขนาด 64 byte โดย CE แต่ละ site เชื่อมต่อกันโดย Virtual Private Network (VPN)



รูปที่ 2 โครงข่าย MPLS ที่ใช้ในการทดลอง

สำหรับโครงข่าย Traditional IP เป็นโครงข่ายเดียวกันกับการทดสอบ MPLS/VPN แต่จะทำการ configured Router (R1-R5) ทำงานโดย routing protocol OSPF (Open Shortest Path First) ทั้งหมดเชื่อมต่อกันด้วย 10 Gigabit Ethernet interface โดยมี Traffic Generator ซึ่งจำลองเป็นอุปกรณ์ CE (Customer Equipment) ส่งข้อมูล Ethernet ขนาด 64 byte เชื่อมต่อกับ Router (R1,R4,R5) ด้วย Gigabit Ethernet interface

3.1 เปรียบเทียบประสิทธิภาพด้านความเชื่อถือได้ของระบบ (Reliability) ระหว่าง MPLS/VPN และ Traditional IP Network

การทดสอบ ประสิทธิภาพด้านความมั่นคง (Reliability) เป็นการทดสอบความสามารถของระบบในการส่งข้อมูลเมื่อเส้นทางในการส่งข้อมูลเกิดความบกพร่อง เช่น สายไยแก้วนำแสงที่ใช้เชื่อมต่อเกิดชำรุดโดยระบบที่มี Reliability ที่ดีต้องสามารถทำการ switch เลือกเส้นทางไปยังเส้นทางสำรองได้โดยไม่กระทบกับการส่งข้อมูล

3.1.1 MPLS/VPN

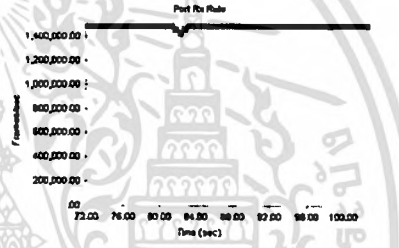
ทำการ Configured Router เพื่อให้บริการ MPLS Traffic Engineering (TE) Fast Reroute ในกรณีที่ link ระหว่าง Node เกิดการบกพร่อง จากรูปที่ 2 ข้อมูลจะถูกส่งจาก Traffic Generator ที่ความเร็ว 1Gbps ลักษณะการส่งข้อมูลจะเป็นแบบ bidirection ส่งผ่านระหว่าง site 1 และ site 2 โดย R2 และ R3 จะทำการ enable การใช้งาน Fast Reroute (FRR) โดยใช้ Resource Reservation Protocol (RSVP) ในการสถาปนา TE tunnel [6] เส้นทางหลักในการส่งข้อมูลระหว่าง site 1 และ site2 คือ R5->R2->R3->R4 ในกรณีที่เส้นทางหลักเกิดการบกพร่องข้อมูลจะส่งผ่านไปยังเส้นทางสำรอง R5->R2->R1->R3->R4

```
R2#show mpls traffic-eng tunnel tunnel 1

Name: R2_t1                    (Tunnel1) Destination: 10.0.0.3
 Status:
   Admin: up      Oper: up    Path: valid     Signalling: connected
   path option 1, type explicit R2-to-R3 (Basis for Setup, path weight 1)
   path option 2, type dynamic
```

รูปที่ 3 แสดงการตั้งค่า FRR เส้นทางหลักที่ R2

```
R2#show mpls traffic-eng tunnel tunnel 2

Name: R2_t2                    (Tunnel2) Destination: 10.0.0.3
 Status:
   Admin: up      Oper: up    Path: valid     Signalling: connected
   path option 1, type explicit R2-to-R1-to-R3 (Basis for Setup, path weight 2)
```

รูปที่ 4 แสดงการตั้งค่า FRR เส้นทางสำรองที่ R2



รูปที่ 5 ค่า Throughput ของโครงข่าย MPLS/VPN ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

รูปที่ 5 แสดงผลการทดสอบเมื่อทำการปลดสาย fiber optic ที่เชื่อมต่อระหว่าง Router R2 และ R3 โดยสามารถหาค่า Ethernet frame rate (frame/second) ได้จาก

$$\frac{Interface\ speed\ bps}{(Ethernet\ frame\ size\ byte + preamble\ size\ byte + inter\ frame\ gap\ size\ byte) \times 8}$$

เมื่อ *Interface speed bps* มีค่า 1Gbps

Ethernet frame size byte มีค่า 64 byte

preamble size byte มีค่า 8 byte

inter frame gap size byte มีค่า 12 byte

Ethernet frame rate (frame/second)

$$= \frac{1Gbps}{(64\ byte + 8\ byte + 12\ byte) \times 8}$$

$$= 1,488,095\ frame/second \quad (1)$$

Stream	Expected Frames	Tx Frames	Lost Frames	% Loss
Stream1	267857142	267857142	55951	0.02089
Stream2	267857142	267857142	35661	0.01331
Total	535714284	535714284	91612	0.0171

ตารางที่ 1 ค่า packet loss ของ โครงข่าย MPLS/VPN ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

จาก (1) สามารถหาค่า recovery time ที่เกิด lost frames ได้จาก

$$= \frac{lost\ frame}{Ethernet\ frame\ rate\ (frame/second)}$$

$$= \frac{91612}{1,488,095\ (frame/second)}$$

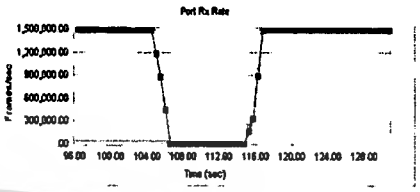
$$= 61\ msec \quad (2)$$

3.1.2 Traditional IP Network

ทำการ Configured Router Traditional IP Network ทำงานโดย routing protocol OSPF (Open Shortest Path First) เพื่อหาเส้นทางในการส่งข้อมูล ด้วยการประกาศข้อมูลของเส้นทางเช่น Bandwidth latency time เพื่อใช้ประกอบในการคำนวณหาเส้นทางที่ดีที่สุด ถ้าเส้นทางที่ใช้ในการส่งข้อมูลเกิดการบกพร่อง (fault) routing protocol OSPF ต้องทำการคำนวณหาเส้นทางใหม่เพื่อใช้ในการส่งข้อมูลแทนเส้นทางหลัก [4]

ตามรูปที่ 2 เส้นทางหลักในการส่งข้อมูลระหว่าง site 1 และ site 2 คือ R5->R2->R3->R4 ซึ่งเป็นเส้นทางที่ให้ค่า cost path ต่ำสุด รูปที่ 6 แสดงผลการทดสอบเมื่อทำการปลดสาย fiber optic ที่เชื่อมต่อระหว่าง Router R2 และ R3 ในขณะที่ทำการปลดสาย fiber optic ที่เชื่อมต่อระหว่าง Router R2 และ R3 ค่า Throughput ตกลงเหลือ 0 Frames/sec แต่เมื่อ routing protocol OSPF ทำการคำนวณหาเส้นทางในการส่งข้อมูลใหม่ได้สำเร็จซึ่งก็คือเส้นทาง R5->R2->R1->R3->R4 ระบบก็สามารถส่งข้อมูลได้ปกติ จากรูปที่ 6 สามารถหาค่า recovery time ได้โดยดูจากกราฟช่วงเวลาที่ค่า throughput เริ่มตกลงเป็นศูนย์จนเริ่มกลับมาส่งข้อมูลได้อีกครั้งมีค่าประมาณ

117 sec - 104 sec = 13 sec (3)



รูปที่ 6 ค่า Throughput ของ โครงข่าย Traditional IP ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

```
sh ip route summary
IP routing table name is default (0x0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Replicates  Overhead    Memory (bytes)
connected       0           11          0           572         1892
static          0           0           0           0           0
ospf 1          20412       905         0           1108536     3751752
    Intra-area: 317 Inter-area: 1000 External-1: 20000 External-2: 0
    NSSA External-1: 0 NSSA External-2: 0
```

รูปที่ 7 Routing Table ของ โครงข่าย Tradition IP ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

ตารางที่ 2 แสดงค่า recovery time และค่า frame Loss ของการทดสอบ reliability ระหว่าง MPLS/VPN และ Traditional IP Network

Technology	Recovery Time	% Loss
MPLS/VPN	61 msec	0.0171
Tradition IP	13 sec	100

ตารางที่ 2 เปรียบเทียบค่า Recovery Time , Packet loss ระหว่าง MPLS/VPN และ Traditional IP Network ในการทดสอบความสามารถด้านความเชื่อถือได้ของระบบ

3.2 เปรียบเทียบประสิทธิภาพด้าน คุณภาพการบริการ (Quality of Service) ระหว่าง MPLS/VPN และ Traditional IP Network

การทดสอบ ประสิทธิภาพด้าน คุณภาพการบริการ (Quality of Service) เป็นการทดสอบความสามารถในการให้บริการข้อมูลประเภท voice video และ data โดยโครงข่าย MPLS/VPN สามารถจัดลำดับความสำคัญของข้อมูลประเภท voice video ให้สามารถใช้งานได้โดยไม่มีผลกระทบต่อคุณภาพแม้โครงข่ายจะเกิดความคับคั่ง การทดลองจะ ทำการส่งข้อมูล ที่ความเร็ว 1 Gbps จาก site1 และ site2 ไปยัง site3 โดยทั้ง 3 site เชื่อมต่อด้วย Gigabit Ethernet interface ดังนั้นจึงมีข้อมูลขนาด 2 Gbps วิ่งเข้าไปยัง site3 ทำให้ site3 เกิดความคับคั่งของข้อมูล ซึ่งใน Bandwidth 1 Gbps แบ่งการส่งเป็น UDP traffic voice 250 Mbps มีค่าลำดับความสำคัญสูงสุด UDP traffic video 250 Mbps มีค่าลำดับความสำคัญลำดับสอง และ TCP traffic data 500 Mbps มีค่าลำดับความสำคัญลำดับสาม ซึ่ง traffic ทั้งสามแบบตั้งค่าให้อยู่ VPN ที่ต่างกัน

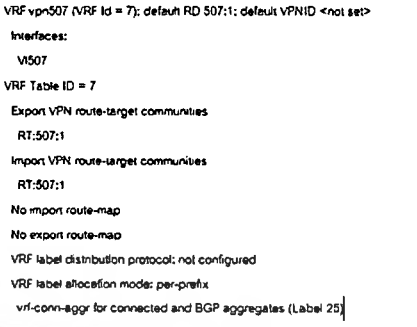
3.2.1 MPLS/VPN

จากรูปที่ 2 MPLS Network ทำการ Configured Router เพื่อทำฟังก์ชั่น MPLS DiffServ และทำการส่งข้อมูลจาก site1และ site2 ไปยัง site3 หลังจากนั้นทำการวัดค่า packet loss ที่ site3 ซึ่งแสดงดังตารางที่ 3

Stream	Expected Frames	Tx Frames	Lost Frames	% Loss
voice	12668027	12668027	0	0
video	12668026	12668026	6634	0.05237
data	25339587	25339587	25338574	99.996

ตารางที่ 3 ค่า frames loss ของโครงข่าย MPLS/VPN ในการทดสอบประสิทธิภาพด้านคุณภาพการบริการ

รูปที่ 8 แสดงการตั้งค่า MPLS DiffServ ในการทดสอบประสิทธิภาพด้านคุณภาพการบริการ



รูปที่ 9 แสดงการตั้งค่า MPLS/VPN ในการทดสอบประสิทธิภาพด้านคุณภาพการบริการ

3.2.2 Traditional IP Network

ทำการ Configured Router เพื่อทำฟังก์ชั่น การให้บริการแบบ Best Effort และทำการส่งข้อมูลจาก site1 และ site2 ไปยัง site3 หลักจากนั้นทำการวัดค่า packet loss ที่ site3 ซึ่งแสดงได้ดังตารางที่ 4

Stream	Expected Frames	Tx Frames	Lost Frames	% Loss
voice	12668027	12668027	6589531	52.017
video	12668026	12668026	6589521	52.016
data	25339587	25339587	13187811	52.044

ตารางที่ 4 ค่า frames loss ของโครงข่าย Traditional IP ในการทดสอบประสิทธิภาพด้านคุณภาพการบริการ

4. สรุปผล

บทความนี้นำเสนอการเปรียบเทียบประสิทธิภาพของโครงข่าย MPLS/VPN กับโครงข่าย IP แบบดั้งเดิม โดยทำการทดลองประสิทธิภาพในด้านความมั่นคงของระบบ (Reliability) และความสามารถ ด้านคุณภาพการบริการ (Quality of Service) ซึ่งผลที่ได้ จากการทดลอง แสดงให้เห็นว่าโครงข่าย MPLS/VPN ที่ใช้หลักการของ Traffic Engineering Fast Reroute ช่วยทำให้โครงข่ายมี Reliability ที่ดีขึ้นระบบมีค่า recovery time ที่ต่ำมากเมื่อเทียบกับการทำงานของ routing protocol ส่วนการให้บริการ QoS จะเห็นว่า โครงข่าย MPLS/VPN ที่มี

การตั้งค่า MPLS DiffServ เมื่อระบบเกิดความคับคั่ง traffic ที่ต้องการ ความมีเสถียรภาพ ที่สูงเช่น voice video ยังคงสามารถส่งข้อมูลต่อไปได้อย่างต่อเนื่องโดยดูได้จากค่า frame loss ที่มีค่าน้อยมาก ตรงกันข้ามกับ Traditional IP ที่ตั้งค่าแบบ Best Effort เมื่อระบบเกิดความคับคั่งจะเกิด frame loss เป็นจำนวนมาก

5. เอกสารอ้างอิง

[1] L. Lobo and U. Lakshman, "MPLS Configuration on Cisco IOS Software", Indianapolis, Indiana Cisco Press,2005.

[2] Student Guide "Implementing Cisco MPLS" Indianapolis, Indiana : Cisco Press,2004

[3] W.Y. Lee, R. Bhagavathula, N. Thanthry and R. Pendse, "MPLS-over-GRE Base VPN Architecture: A Performance Comparison," Proc. of the 2002 (45th) IEEE Midwest Symposium on Circuits and Systems (MWSCAS-2002), 4-7 Aug 2002.

[4] F. Fujikawa, K. Kuwabara, Y. Koda, and M. Kiuchi, "Examination of Electric Power Utility Network Applying IP Router/MPLS Router/Wide-Area Ethernet," IEEE Power Engineering Society General Meeting, 6-10 June 2004.

[5] J. Barakovic, H. Bajric, and A. Husic, "Multimedia Traffic Analysis of MPLS and non-MPLS Network," IEEE Multimedia Signal Processing and Communications, 48th International Symposium ELMAR-2006, June 2006.

[6] D.L. Zhang and D. Ionescu, "QoS Performance Analysis in Deployment of DiffServ-aware MPLS Traffic Engineering," Proc. of 2007 IEEE Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007). ACIS International Conference, July 30 2007-Aug. 1 2007

[7] S. Kim, H.- Y. Ryu, Jaehyung Park, and Taeil Kim, "Design and implementation of Martini based Layer 2 VPN", Proc. of the 8th IEEE International Conference on Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, 20-22 Feb. 2006

[8] B. Alawieh, and. H.T Mouftah, "Efficient Delivery of Voice Services over MPLS Internet Infrastructure," Proc. of 2007 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2007), 22-26 April 2007.

ประวัติผู้เขียน

นายบดินทร์ จิ๋วแย้ม เกิดเมื่อวันที่ 9 กรกฎาคม พ.ศ.2517 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์ จากภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร ในปีการศึกษา 2542 และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมโทรคมนาคม ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2548 โดยในปี พ.ศ. 2544 ได้เข้าทำงานในตำแหน่งวิศวกรสื่อสารระดับ 4 แผนกสื่อสารข้อมูล กองปฏิบัติการระบบสื่อสาร ฝ่ายปฏิบัติการระบบสื่อสารและคอมพิวเตอร์ การไฟฟ้านครหลวง