ซอฟต์แวร์จำลองการทำงานของไอพีสวิตช์

IP SWITCH SOFTWARE

โดย

นายธาริน กิ่งใบสมบูรณ์

นายธีรนันท์ หุตางกูร

นายภาคย์ อยู่รอต

อาจารย์ที่ปรึกษา

ผศ. บรรจง ปิยธำรงค์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ซอฟต์แวร์จำลองการทำงานของไอพีสวิตช์

IP SWITCH SOFTWARE

ผู้จัดทำ

1. นาย ธาริน กิ่งใบสมบูรณ์ รหัสประจำตัว 40010319
2. นาย ธีรนันท์ หุตางกูร รหัสประจำตัว 40010329
3. นาย ภาคย์ อยู่รอต รหัสประจำตัว 40010554

_____ อาจารย์ที่ปรึกษา

(ผศ. บรรจง ปิยธำรงค์)

ซอฟต์แวร์จำลองการทำงานของไอพีสวิตช์

นายธาริน กิ่งใบสมบูรณ์ 40010319
นายธีรนันท์ หุตางกูร 40010329
นายภาคย์ อยู่รอต 40010554
ผศ. บรรจง ปิยธำรงค์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ในปัจจุบันพัฒนาการทางด้านเทคโนโลยีคอมพิวเตอร์และสารสนเทศได้เจริญก้าวไกลไปมาก ความต้องใช้งานข้อมูลข่าวสารต่างๆ รวมทั้งการติดต่อสื่อสารระหว่างกันผ่านระบบเครือข่ายคอมพิวเตอร์ ได้กลายมาเป็นสิ่งสำคัญที่ขาดมิได้ในทุกองค์กร และยังคงมีการขยายตัวมากขึ้นทุกขณะ นอกจากนี้ เทคโนโลยีทางด้านซอฟต์แวร์ต่างๆ ซึ่งกำลังเป็นที่ตื่นตัวอย่างมากในปัจจุบัน อย่างเช่นการกระจายส่วนประกอบต่างๆ (Distributed Components) ของซอฟต์แวร์ไปไว้บนเครื่องเซิร์ฟเวอร์หลายๆ เครื่อง แล้วค่อยเรียกใช้ผ่านเครือข่ายนั้น ก็เป็นอีกปัจจัยหนึ่งที่ทำให้เทคโนโลยีการจัดการระบบเครือข่ายจำเป็นจะต้องมีประสิทธิภาพสูงขึ้น และเป็นการกระตุ้นให้เกิดพัฒนาการต่างๆ ตามมานั่นเอง

ไอพีสวิตช์ เป็นแนวทางหนึ่งของการพัฒนาเทคโนโลยีการจัดการระบบเครือข่ายรูปแบบใหม่ โดยไอพีสวิตช์แบบดั้งเดิมนั้นมีวิวัฒนาการมาจากแนวคิดที่ต้องการนำเอาข้อดีของการค้นหาเส้นทางบนระบบเครือข่ายที่ทรงประสิทธิภาพตามแบบไอพีเราติง (IP Routing) รวมทั้งความสามารถในการแบ่งแยกการสื่อสารข้อมูลในระบบเครือข่าย (LAN Traffic) ของอุปกรณ์เราเตอร์ มาใช้ควบคู่กันไปกับอุปกรณ์เอทีเอ็มสวิตช์ที่มีความสามารถในการส่งผ่านแพ็คเก็ตที่ความเร็มเร็วสูงมาก ซึ่งทำให้การสื่อสารข้อมูลบนระบบเครือข่ายเป็นไปได้อย่างเต็มประสิทธิภาพมากยิ่งขึ้น นอกจากนี้แล้วการรวมเอา เราติงอัลกอริทึมกับสวิตชิงฮาร์ดแวร์เข้ามาทำงานร่วมกันนั้น ยังสามารถลดความซับซ้อนของอุปกรณ์ลงไปได้ ไอพีสวิตช์จึงมีราคาถูกกว่าเราเตอร์ และยังสามารถลดปัญหาคอขวดของการดำเนินงานในส่วนของอัลกอริทึมได้ โดยการแปลงการทำงานของเราติงซอฟต์แวร์ ให้กลายเป็นวงจรฮาร์ดแวร์เสีย ไอพีสวิตช์ในปัจจุบันได้มีการพัฒนาขึ้นมาเป็นลำดับ มีความหลากหลายเพิ่มขึ้น จึงไม่จำเป็นต้องทำงานบนเอทีเอ็ม-สวิตช์แต่เพียงอย่างเดียวอีกต่อไป

ปริญญานิพนธ์ฉบับนี้ แสดงถึงการพัฒนาซอฟต์แวร์เพื่อจำลองการทำงานของอุปกรณ์ไอพีสวิตช์ซึ่งโดยปกติแล้วอุปกรณ์ไอพีสวิตช์จะเป็นฮาร์ดแวร์ทั้งสิ้น เพื่อให้สามารถอธิบายการทำงานของอุปกรณ์ไอพีสวิตช์ได้อย่างชัดเจน รวมทั้งสามารถนำมาใช้เป็นแบบอย่างที่ดีสำหรับการศึกษาแนวคิดและกระบวนการปรับปรุงรูปแบบการจัดการการสื่อสารข้อมูลในระบบเครือข่ายที่มีอยู่เดิมนั้น ให้มีประสิทธิภาพสูงขึ้นต่อไปในอนาคตได้อีกด้วย

IP Switch Software

Tarin Kingbaisomboon

Theeranan Hutangkul

Parck Yoorot

ASST. PROF. Banjong Piyatamrong Advisor

ABSTRACT

Nowadays, the need of information and communication over computer networks has become critical part in information technology age. The development of software technologies such as distributed computing (DCOM), which requires better network management, is also pushing the development of computer network furthermore.

IP Switch is one new way of network management. Originally, IP Switch evolved from the concept of merging the advantages of IP Routing including the LAN traffic separation of Router together with the use of ATM switch, which is capable of transmitting packets at very high speed, in order to achieve better network communication performance. The merging the routing algorithm and switching hardware not only reduce the complication but also makes IP Switch cheaper than router and alleviate the bottleneck of the algorithm by changing routing software to hardware. With current technologies, IP Switch no longer need to work only with ATM switch.

This thesis concerns the developing of IP Switch Simulator Software in order to study the concepts, explain how IP Switch, which normally are hardware, works and improve the traffic management methods of existing networks.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายเข้าด้วยกัน บุคคลแรกที่จะต้องกล่าวถึง เนื่องจากเป็นผู้ที่มีส่วนสำคัญอย่างมากที่ทำให้วิทยานิพนธ์ฉบับนี้เสร็จสิ้นลงได้ด้วยดีก็คือ ผู้ช่วยศาสตราจารย์ บรรจง ปิยธำรงค์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งให้ความดูแลเอาใจใส่ ให้คำแนะนำ และคอยให้ความช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก ขอขอบคุณ อาจารย์ ธนา หงส์สุวรรณ และ พี่แพง (รุ่นที่ 34) สำหรับคำแนะนำเกี่ยวกับเรื่อง Packet Driver ขอขอบคุณ อาจารย์ อัครเดช วัชรภูพงษ์ และ พี่นเรศ (ป.โทฯ) สำหรับเครื่องพิมพ์ ขอขอบคุณ พี่มานต (ESL LAB), พี่เสือ (ศูนย์วิจัยฯ), พี่โอ๊ตกับพี่อรรถ (TelecomASIA), พี่สโรชิน (กองทัพเรือ), พี่เช (รุ่น 33), โก้ 4D (ห้อง Hardware Lab), เชษฐ์ 4D (ห้อง Olala Lab), บริษัท ไมโครซอฟต์ประเทศไทย จำกัด, บริษัท ซิสโก้ ซีสเต็มส์ (ประเทศไทย), และบริษัทไอโซเน็ต สำหรับข้อมูลต่างๆ เกี่ยวกับการพัฒนาซอฟต์แวร์ระบบเครือข่าย ขอขอบคุณ ยุวดี (มือพิมพ์ดีด ภาคคอนโทรลฯ) ขอขอบคุณ พี่ๆ เพื่อนๆ ห้อง Network, ห้อง Hardware, ห้อง Olala, ห้อง AI, และห้อง ISAG สำหรับกำลังใจ และความช่วยเหลือต่างๆ ที่มีให้กันตลอดมา ทั้งทางด้านเงินทุนหมุนเวียน อาหารมื้อดึก ที่พักชั่วคราวอันอบอุ่น รวมทั้งความบันเทิงในยามราตรีที่มีอยู่อย่างเหลือเฟือ ณ ชั้น 3 ตึก B ภาควิชาวิศวกรรมคอมพิวเตอร์แห่งนี้

สุดท้ายนี้ก็ต้องขอกราบขอบพระคุณบุคคลที่มีความสำคัญที่สุดที่ทำให้ข้าพเจ้าทั้งสามมีวันนี้ได้ ซึ่งก็คือ บิดา-มารดา อันเป็นที่เคารพรักยิ่งของพวกเรา เป็นผู้ซึ่งได้เลี้ยงดูผู้เขียนแต่ละคนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ อีกทั้งยังให้กำลังใจ และคอยดูแลเอาใจใส่เสมอมา ในทุกๆ ด้าน อันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณบิดามารดาของคณะผู้จัดทำทั้งสามผ่านมา ณ ที่นี้ด้วย

ธาริน กิ่งใบสมบูรณ์
ธีรนันท์ หูตางกูร
ภาคย์ อยู่รอต

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	**I**
บทคัดย่อภาษาอังกฤษ	**II**
กิตติกรรมประกาศ	**III**
สารบัญ	**IV**
สารบัญตาราง	**VII**
สารบัญรูปภาพ	**VIII**
บทที่ 1 บทนำ	**1**
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	2
1.4 วิธีดำเนินงาน	2
บทที่ 2 ระดับชั้นของโอเอสไอกับอุปกรณ์เครือข่าย	**3**
2.1 Hub หรือ Repeater	3
2.2 Switch หรือ Bridge	3
2.3 Router	4
2.4 Gateway	4
2.5 Layer 3 Switch	5
บทที่ 3 สถาปัตยกรรมเครือข่ายและระดับชั้นของโพรโตคอล	**7**
3.1 แบบอ้างอิงโอเอสไอ (OSI Reference Model)	7
3.2 ชุดโพรโตคอลทีซีพี/ไอพี (TCP/IP Protocol suite)	9
3.2.1 เลเยอร์เน็ตเวิร์คของทีซีพี/ไอพี	10
3.2.2 เลเยอร์ทรานสปอร์ตของทีซีพี/ไอพี	24
3.2.3 เลเยอร์แอพพลิเคชั่นของทีซีพี/ไอพี	29
3.3 เครือข่ายท้องถิ่นแบบอีเธอร์เนต (Ethernet LAN)	32
3.3.1 เฟรมอีเธอร์เนตแบบต่าง ๆ	37
3.3.1.1 อีเธอร์เนต 802.3	37
3.3.1.2 อีเธอร์เนต 802.2	38
3.3.1.3 อีเธอร์เนตสแนป	39
3.3.1.4 อีเธอร์เนตทู	41

3.3.2 ความแตกต่างระหว่างเฟรมอีเธอร์เนตแบบต่าง ๆ 42
3.3.3 โพรโตคอลไอพีเอ็กซ์/เอสพีเอ็กซ์ของเน็ตแวร์ 42
3.3.3.1 เฮดเดอร์ไอพีเอ็กซ์ 42
3.3.3.2 เฮดเดอร์เอสพีเอ็กซ์ (Sequence Packet Exchange : SPX) 44
บทที่ 4 แนวคิดพื้นฐานของอุปกรณ์สวิตชิ่ง (Switching Concept) **46**
4.1 แนวคิดพื้นฐานของอุปกรณ์สวิตชิ่ง 46
4.1.1 คุณสมบัติพื้นฐาน 2 ประการ 46
4.1.2 มุมมอง 2 ประการ 46
4.2 กลวิธีในการส่งต่อชุดข้อมูล 47
4.2.1 ในกรณีของการใช้ค่าตำแหน่งเครื่องปลายทาง (Destination Address) 47
4.2.2 ในกรณีของการใช้ค่าทิศทางจากเครื่องต้นทาง (Source-Route Vector) 48
4.2.3 ในกรณีของการใช้ค่าระบุช่องทางการเชื่อมต่อ (Connection Identifier) 49
4.3 กลวิธีควบคุมเส้นทาง 50
4.3.1 การเรียนรู้ตำแหน่ง (Address Learning) 50
4.3.2 แนวคิดแบบรากไม้เชิงกว้าง (Spanning Tree Algorithm) 51
4.3.3 การสำรวจเส้นทาง Broadcast and Discover 52
4.3.4 การจัดการเส้นทางแบบสถานะการต่อเชื่อม (Link State Routing) 53
4.3.5 การใช้สัญญาณระบุเส้นทางที่ชัดเจน (Explicit Signaling) 53
4.4 การสับเปลี่ยนช่องทางการสื่อสารบนระบบเครือข่ายท้องถิ่น (LAN Switching) 54
4.4.1 กลวิธีแบบส่งต่อทันที (Cut-Through Forwarding) 56
4.4.2 กลวิธีแบบเก็บพักข้อมูล (Store and Forward) 57
บทที่ 5 หลักการพื้นฐานของเทคโนโลยีไอพีสวิตช์ **58**
5.1 แนวคิดพื้นฐานของไอพีสวิตช์ 58
5.2 นิยามและศัพท์บัญญัติ 58
5.2.1 ไอพีสวิตช์ 58
5.2.2 อินเกรสและอีเกรส (Ingress and Egress) 60
5.2.3 เส้นทางลัด (Shortcut Path) 61
5.3 ปัจจัยที่กระตุ้นให้เกิดการพัฒนา 62
5.4 รูปแบบค่าตำแหน่ง (IP Switching Addressing Models) 63
5.4.1 Separated Addressing Model 63
5.4.2 IP-to-VC 64
5.5 โครงแบบของไอพีสวิตช์ (IP Switching Model) 64
5.5.1 โครงแบบซ้อนทับ (Overlay Model) 64
5.5.2 โครงแบบเพียร์ (Peer Model) 66

5.6 ประเภทของไอพีสวิตช์ (IP Switching Types) 67
5.6.1 ทำงานโดยกระแสข้อมูล (Flow-Driven Solution) 67
5.6.2 ทำงานโดยโครงสร้างเครือข่าย (Topology-Driven Solution) 67
5.7 การจัดจำแนกประเภทของเทคโนโลยีไอพีสวิตชิ่ง (IP Switching Taxonomy) 68
บทที่ 6 การติดต่อใช้งานแพ็กเก็ตไดรเวอร์ (Packet driver usage) **69**
6.1 แพ็กเก็ตไดรเวอร์ (Packet Driver Description) 69
6.2 WinPcap 69
6.3 Packet Driver API 70
6.4 ข้อกำหนดสำหรับเอกสารฉบับนี้ (Document Conventions) 71
6.5 ความรู้เบื้องต้นและแรงจูงใจในการพัฒนา (Introduction and Motivation) 71
บทที่ 7 ขั้นตอนการออกแบบและพัฒนาซอฟต์แวร์ **72**
7.1 กำหนดสภาพแวดล้อมของระบบ 72
7.1.1 ระบบไอพีสวิตช์ (IP Switch System) 72
7.1.2 รูปแบบการจัดการค่าตำแหน่งที่อยู่ของเครื่อง (Addressing Model) 72
7.1.3 โครงแบบไอพีสวิตช์ (IP Switching Model) 72
7.2 การดำเนินการภายในของไอพีสวิตช์ 73
7.2.1 โครงสร้างข้อมูลหลักของโปรแกรม 72
7.2.2 ฟังก์ชั่นการทำงานหลักของโปรแกรม 74
บทที่ 8 การทดลอง และผลการทดสอบ **76**
8.1 กำหนดรูปแบบการทดลอง 76
8.2 ขั้นตอนการทดลอง 76
8.3 การตรวจสอบความถูกต้อง 79
8.4 การตรวจสอบประสิทธิภาพ 82
บทที่ 9 บทวิจารณ์และสรุป
9.1 วิจารณ์โครงงาน 83
9.2 บทสรุปโครงงาน 84
9.3 ข้อควรปรับปรุง 84
ภาคผนวก ก **85**
ภาคผนวก ข **90**
บรรณานุกรม **93**

สารบัญตาราง

หน้าที่

ตารางที่ 5-1 ตารางเปรียบเทียบคุณสมบัติต่างๆ ของโครงแบบไอพีสวิตช์ทั้ง 2 ชนิด 66

สารบัญรูปภาพ

หน้าที่

รูปที่ 3-1 แบบอ้างอิงโอเอสไอ 7
รูปที่ 3-2 สถาปัตยกรรมทีซีพี/ไอพี 9
รูปที่ 3-3 การส่งผ่านข้อมูลระหว่างเลเยอร์ 10
รูปที่ 3-4 สถาปัตยกรรมไอพี 10
รูปที่ 3-5 แสดงไอพีในเฟรมอีเธอร์เน็ตทู (ก) และในเฟรม SNAP (ข) 11
รูปที่ 3-6 รูปแบบของดาต้าแกรมไอพีเวอร์ชั่น 4 12
รูปที่ 3-7 ออปชั่นในการวัดและความปลอดภัย 14
รูปที่ 3-8 รูปแบบดาต้าแกรมเออาร์พี 16
รูปที่ 3-9 แสดงไอซีเอ็มพีชนิดต่าง ๆ ถูกเอ็นแคปซูเลทในไอพี 17
รูปที่ 3-10 เฮดเดอร์พื้นฐานของโพรโตคอลไอซีเอ็มพี 18
รูปที่ 3-11 รูปแบบดาต้าแกรมไอซีเอ็มพีชนิด Echo 18
รูปที่ 3-12 ค่าเดสติเนชั่นอันริชเอเบิล ในฟิลด์โค้ด 19
รูปที่ 3-13 รูปแบบของแมสเสจ route change request (ก)
และค่าในฟิลด์โค้ดที่แมสเสจประเภทนี้ใช้ (ข) 21
รูปที่ 3-14 รูปแบบแมสเสจไอซีเอ็มพี Router advertisement 21
รูปที่ 3-15 รูปแบบแมสเสจไอซีเอ็มพี Router solicitation 22
รูปที่ 3-16 รูปแบบแมสเสจไอซีเอ็มพี parameter problem 22
รูปที่ 3-17 รูปแบบแมสเสจไอซีเอ็มพี time stamp request/reply 23
รูปที่ 3-18 รูปแบบแมสเสจไอซีเอ็มพี information request 23
รูปที่ 3-19 รูปแบบแมสเสจไอซีเอ็มพี address mask request 23
รูปที่ 3-20 หมายเลขพอร์ตที่ใช้ในยูดีพีและทีซีพี 24
รูปที่ 3-21 ฟิลด์ในเฮดเดอร์ของยูพีดี 25
รูปที่ 3-22 ค่าเช็คซัมประกอบด้วยเฮดเดอร์และซูโดเฮดเดอร์ 26
รูปที่ 3-23 เฮดเดอร์ทีซีพี 27
รูปที่ 3-24 โฟล์วชาร์ตแสดงการส่ง 34
รูปที่ 3-25 โฟล์วชาร์ตแสดงการรับ 36
รูปที่ 3-26 เฟรมอีเธอร์เนต 802.3 37
รูปที่ 3-27 โครงสร้างเฟรม 802.2 39

รูปที่ 3-28 โครงสร้างเฟรมอีเธอร์เน็ตสแนป 40
รูปที่ 3-29 โครงสร้างของอีเธอร์เนตทู 41
รูปที่ 3-30 โฟล์วชาร์ตแสดงแบบของเฟรมต่าง ๆ 42
รูปที่ 3-31 แสดงรูปแบบของเฮดเดอร์ไอพีเอ็กซ์ 42
รูปที่ 3-32 แสดงรูปแบบเฮดเดอร์ของเอสพีเอ็กซ์ 44
รูปที่ 4-1 ระดับการเชื่อมต่อในลำดับชั้นของการสื่อสารบนระบบเครือข่ายสวิตชิ่ง (Protocol Stack) 46
รูปที่ 4-2 กลไกการทำงานของกรณีใช้ค่าตำแหน่งเครื่องปลายทาง (Destination Address) 48
รูปที่ 4-3 กลไกการทำงานของกรณีที่ใช้ค่าทิศทางจากเครื่องต้นทาง (Source-Route Vector) 48
รูปที่ 4-4 กลไกการทำงานของกรณีที่ใช้ค่าระบุช่องทางการเชื่อมต่อ (Connection Identifier) 49
รูปที่ 4-5 กลไกการทำงานของกรรมวิธีการเรียนรู้ตำแหน่ง (Address Learning) 50
รูปที่ 4-6 กลไกการทำงานของระบบเครือข่าย ตามแนวคิดแบบรากไม้เชิงกว้าง (Spanning Tree) 51
รูปที่ 4-7 กลไกการทำงานของกรรมวิธีการสำรวจเส้นทาง (Broadcast and Discover) 52
รูปที่ 4-8 ขั้นตอนการทำงานของระบบสัญญาณระบุแสดงเส้นทางที่ชัดเจน (Explicit Signaling) 53
รูปที่ 4-9 ระบบเครือข่ายท้องถิ่นแบบร่วมกัน (Shared LAN) 54
รูปที่ 4-10 ระบบเครือข่ายท้องถิ่นย่อย 55
รูปที่ 4-11 การส่งต่อชุดข้อมูลแบบส่งต่อทันที (Cut-Through Forwarding) 56
รูปที่ 4-12 การส่งต่อชุดข้อมูลแบบเก็บพักข้อมูล (Store and Forward) 57
รูปที่ 5-1 อุปกรณ์ไอพีสวิตช์ และไอพีสวิตช์แบบเสมือน 59
รูปที่ 5-2 ฟังก์ชั่น Ingress และ Egress ของไอพีสวิตช์ 60
รูปที่ 5-3 โครงแบบไอพีสวิตช์ชนิดซ้อนทับ (Overlay Model) โดยใช้ระบบไอพีสวิตช์แบบเสมือน (Virtual IP Switch) 65
รูปที่ 5-4 โครงแบบไอพีสวิตช์ชนิดเพียร์ (Peer Model) 66
รูปที่ 5-5 แผนผังการทำงานของระบบเครือข่ายไอพีสวิตช์ประเภทที่ทำงานด้วยกระแสข้อมูล 67
รูปที่ 5-6 แผนผังการทำงานของระบบเครือข่ายไอพีสวิตช์ประเภทที่ทำงานโดยโครงสร้างเครือข่าย 67
รูปที่ 5-7 แผนภูมิรากไม้แสดงการจัดจำแนกประเภทของเทคโนโลยต่างๆ ที่ใช้หลักการไอพีสวิตช์ 68
รูปที่ 7-1 ตารางเก็บข้อมูลแบบ Separated Addressing Model ที่ใช้วิธีการเรียนรู้ตำแหน่งที่อยู่ 73
รูปที่ 7-2 Flowchart แสดงการจัดเก็บค่าตำแหน่ง IP และ MAC ลงในตาราง 75
รูปที่ 7-3 Flowchart แสดงการทำงานของ Algorithm ของ โปรแกรม 75
รูปที่ 8-1 แสดงรูปแบบโครงสร้างการต่อเชื่อมทางกายภาพของระบบที่ใช้ทดสอบ 76
รูปที่ 8-2 แสดงลำดับการทำงานของโปรแกรมในระยะเริ่มต้น 77
รูปที่ 8-3 แสดงลำดับการทำงานของโปรแกรมในระยะที่ 2 ของการทดสอบ 78
รูปที่ 8-4 แสดงลำดับการทำงานของโปรแกรมในระยะที่ 3 ของการทดสอบ 78
รูปที่ 8-5 แสดงลำดับการทำงานของโปรแกรมในระยะที่ 4 ของการทดสอบ 79

รูปที่ 8-6 แสดงผลลัพธ์การ Capture ณ เครื่อง C (161.246.5.184) 80
รูปที่ 8-7 แสดงผลลัพธ์การ Capture ณ เครื่อง B (161.246.5.202) 81
รูปที่ 8-8 แสดงผลลัพธ์การ Capture ณ เครื่อง A (161.246.5.169) 81

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากในปัจจุบัน ระบบเครือข่ายคอมพิวเตอร์ได้กลายมาเป็นปัจจัยหนึ่ง ซึ่งมีส่วนสำคัญต่อการดำเนินงานในชีวิตประจำวันของผู้คนมากขึ้นทุกขณะ อีกทั้งข้อมูลข่าวสารที่มีให้บริการอยู่นั้นก็มีแนวโน้มที่จะกลายไปเป็นรูปแบบสื่อผสม (Multimedia) มากขึ้นเรื่อยๆ เช่นกัน ทำให้มีความต้องการทางด้านการสื่อสารข้อมูลเกิดขึ้นอย่างมากมาย และเป็นผลให้เกิดปัญหาในเรื่องความคับคั่งของข้อมูลในระบบเครือข่ายตามมา เนื่องจากอุปกรณ์ต่างๆ ที่ทำหน้าที่จัดการเส้นทางของข้อมูลบนระบบเครือข่ายนั้น ไม่สามารถทำงานได้ทันกับปริมาณการสื่อสารข้อมูลที่เกิดขึ้นอย่างมากมายจนเกินขีดความสามารถของระบบนั่นเอง และจุดนี้ก็คือความจำเป็นและที่มาของความพยายามต่างๆ ในการพัฒนาเทคโนโลยี และข้อตกลงการสื่อสารข้อมูลบนระบบเครือข่ายในรูปแบบใหม่ เพื่อให้สามารถรองรับการขยายตัวของความต้องการการสื่อสารข้อมูลที่เกิดขึ้นได้ ซึ่งเทคโนโลยีไอพีสวิตช์ (IP Switch) ของบริษัท Ipsilon Inc. ก็เป็นหนึ่งในหลายๆ แนวทางที่ได้รับการพัฒนาขึ้นมาตามจุดประสงค์นี้นั่นเอง

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อให้เกิดความเข้าใจเกี่ยวกับปัญหา และที่มาของการพัฒนาเทคโนโลยีการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์

1.2.2 เพื่อศึกษาแนวคิดและทฤษฎีที่นำมาใช้กับเทคโนโลยีไอพีสวิตช์ อันเป็นจุดเริ่มต้นที่ดีจุดหนึ่งในการนำไปพัฒนาเป็นเทคโนโลยีการสื่อสารข้อมูลขั้นสูงอย่างอื่นต่อไป

1.2.3 เพื่อเปรียบเทียบความแตกต่างระหว่างเทคโนโลยีการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์ที่มีอยู่เดิม อย่างเช่นเทคโนโลยีเราเตอร์พื้นฐาน กับเทคโนโลยีการสื่อสารข้อมูลบนระบบเครือข่ายสมัยใหม่อย่างเช่นไอพีสวิตช์

1.2.4 เพื่อศึกษากลไกการทำงานของระบบควบคุมการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์แบบไอพีสวิตช์ และเพื่อเรียนรู้การสร้างซอฟต์แวร์เพื่อควบคุมการทำงานของอุปกรณ์สื่อสารเครือข่ายด้วย Visual C++ บนระบบปฏิบัติการณ์ Microsoft Windows 32 Bits

1.2.5 เพื่อเผยแพร่ตัวอย่างแนวคิดในการแก้ไขปัญหาการสื่อสารข้อมูลบนระบบเครือข่าย รวมทั้งเทคโนโลยีการจัดการเครือข่ายแบบใหม่ๆ

1.3 ขอบเขตของโครงงาน

งานวิจัยนี้เป็นการศึกษาความก้าวหน้าของเทคโนโลยีต่างๆ ที่เกี่ยวข้องกับการจัดการการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์สมัยใหม่ และพัฒนาซอฟต์แวร์จำลองการทำงานขึ้นมาภายใต้ขอบเขตดังต่อไปนี้

1.3.1 รูปแบบของเทคโนโลยีที่ศึกษาจะมุ่งเน้นไปที่เทคโนโลยีไอพีสวิตช์ และเทคโนโลยีอื่นๆ ที่พัฒนาขึ้นมาตามแนวทางของไอพีสวิตช์ เช่นแท็กสวิตช์ (Tag Switch) ของบริษัทซิสโก้ซิสเต็มส์, CSR ของบริษัทโตชิบา, ARIS ของบริษัทไอบีเอ็ม, และ MPLS เป็นต้น

1.3.2 แนวทางในการพัฒนาซอฟต์แวร์ขึ้นมานั้น จะยึดตามมาตรฐานของการสื่อสารข้อมูลในรูปแบบไอพีสวิตช์ ร่วมกับรูปแบบแท็กสวิตช์เท่านั้น ซึ่งเป็นรูปแบบพื้นฐานที่สุดนั่นเอง

1.3.3 การพัฒนาซอฟต์แวร์จะใช้ ภาษา Visual C++

1.3.4 ซอฟต์แวร์ที่พัฒนาขึ้นมานี้ จะทำงานในลักษณะของ Text Mode และสามารถรองรับการทำงานบนระบบปฏิบัติการ Windows 98/ME และ Windows NT/2000 ได้ทุกแพลตฟอร์ม

1.3.5 ซอฟต์แวร์ที่พัฒนาขึ้นมานี้ ถูกออกแบบมาเพื่อให้ใช้งานบนระบบเครือข่ายท้องถิ่นแบบอีเธอร์เน็ต (Ethernet LAN) เท่านั้น โดยใช้รูปแบบการต่อเชื่อมแบบดาว (Star Topology) ซึ่งสามารถเชื่อมต่อได้ทั้งแบบจุดต่อจุด (Point-to-point) และแบบหลายจุด (Multipoint) ผ่านรูปแบบบัส

1.3.6 ซอฟต์แวร์ที่พัฒนาขึ้นมานี้ จะใช้เพื่อจำลองการทำงานของอุปกรณ์ไอพีสวิตช์อย่างง่าย โดยทำงานบนระบบคอมพิวเตอร์ส่วนบุคคลธรรมดาที่ได้รับการติดตั้งแผงวงจรสื่อสารเครือข่ายท้องถิ่น (NICs: Network Interface Card) จำนวน 2 – 3 ใบ เป็นที่เรียบร้อยแล้ว และต่อเชื่อมแผงวงจรสื่อสารแต่ละใบนั้นเข้ากับเครือข่ายท้องถิ่น หรือเครื่องพีซีที่แตกต่างกัน

1.4 วิธีดำเนินงาน

1.4.1 เริ่มต้นศึกษาข้อกำหนดและมาตรฐานต่างๆ ที่ใช้กันอยู่โดยทั่วไปในการจัดการการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์

1.4.2 ศึกษาแนวคิดต่างๆ และรูปแบบการทำงานของอุปกรณ์เราเตอร์ (Router) โดยทั่วไป

1.4.3 ศึกษาข้อกำหนดและมาตรฐานโดยรวมเกี่ยวกับเทคโนโลยีไอพีสวิตช์ และเทคโนโลยีอื่นๆ ที่มีความคล้ายคลึงกัน

1.4.4 ศึกษาเทคโนโลยีไอพีสวิตช์อย่างละเอียด ทั้งทางด้านแนวคิดหลัก กลไกการทำงาน และประสิทธิภาพในด้านต่างๆ เทียบกับระบบที่ใช้งานเราเตอร์ทั่วไป

1.4.5 ออกแบบ และกำหนดขอบเขตโครงสร้างการทำงานของซอฟต์แวร์ที่จะพัฒนาขึ้นมา ว่าจะจำลองการทำงานในส่วนใดของไอพีสวิตช์ซึ่งเป็นอุปกรณ์ทางฮาร์ดแวร์ขึ้นมาบ้าง

1.4.6 ศึกษากรรมวิธีในการติดต่อควบคุมการทำงานของแผงวงจรสื่อสารเครือข่ายผ่านระบบปฏิบัติการวินโดวส์ 32 บิต โดยใช้ Visual C++ 6.0

1.4.7 พัฒนาและทดสอบการทำงานของซอฟต์แวร์ที่สร้างขึ้น

บทที่ 2

ระดับชั้นของโอเอสไอกับอุปกรณ์เครือข่าย

การใช้งานรูปแบบระบบเครือข่ายตามมาตรฐานของ OSI 7-Layer Model เพื่อรับส่งข้อมูลระหว่างคอมพิวเตอร์สองระบบนั้น ปลายทางทั้งด้านผู้รับ และผู้ส่งจะต้องมีขบวนการรับส่งข้อมูลตาม OSI ครบทั้ง 7 ชั้นจึงจะสามารถรับส่งข้อมูลถึงกันได้ แต่ระหว่างเส้นทางนั้นอาจจะผ่านอุปกรณ์เครือข่ายบางอย่างที่ทำงานโดยมีลำดับชั้นไม่ครบ 7 ชั้นก็ได้ เช่น ถ้าระหว่างทางมีการส่งผ่านเครือข่ายอื่นๆ อาจจะมีอุปกรณ์เครือข่ายและลำดับชั้นที่เกี่ยวข้องเพียง 3 ชั้นล่าง คือ ชั้นที่ 1, 2 และ 3 เท่านั้นก็ได้ เมื่อถึงปลายทางผู้รับจึงจะมีการเชื่อมต่อครบทั้ง 7 ชั้น

เมื่อเรามองว่าการรับส่งข้อมูลระหว่างคอมพิวเตอร์สองระบบนั้นจะผ่านชั้นต่างๆของ OSI 7-Layer Model ดังนั้นอุปกรณ์เน็ตเวิร์กต่างๆที่ใช้ก็จะมีการทำงานตรงตามชั้นต่างๆที่กำหนดไว้ใน OSI ด้วย ซึ่งจะแยกอธิบายถึงอุปกรณ์แต่ละอย่างได้ดังนี้

2.1 Hub หรือ Repeater

เป็นอุปกรณ์ที่ทำหน้าที่ขยายสัญญาณที่ได้รับมาส่งต่อให้กับอุปกรณ์อื่นที่ต่อเข้ากับมัน จัดเป็นอุปกรณ์ที่มีการทำงานอยู่ในชั้นที่ 1 หรือ Physical Layer ของ OSI เพราะว่าตัว Hub หรือ Repeater นี้จะขยายสัญญาณที่ได้รับโดยไม่มีซอร์ฟแวร์มาเกี่ยวข้องในการจัดรูปแบบของข้อมูล หรือเปลี่ยนแปลงข้อมูลที่ได้รับมาแต่อย่างใด การติดตั้ง Hub จึงทำได้ง่าย เพราะไม่มีอะไรต้องปรับแต่งในแง่ของซอฟต์แวร์ จุดประสงค์ในการติดตั้ง Hub ก็เพื่อขยายสัญญาณให้ต่อกับอุปกรณ์ต่างๆได้เพิ่มมากขึ้น หรือในบางกรณีก็ต่อเพื่อเพิ่มระยะทางของสายให้ไกลเพิ่มขึ้นเท่านั้น

2.2 Switch หรือ Bridge

เป็นอุปกรณ์สำหรับเชื่อมต่อ LAN สองเครือข่ายเข้าด้วยกัน โดยจะต้องเป็น LAN ประเภทเดียวกัน และใช้ข้อกำหนดในการรับส่งข้อมูล (Protocol) เหมือนกัน เช่น ใช้ในการเชื่อมต่อระบบเครือข่ายท้องถิ่นแบบอีเธอร์เน็ต (Ethernet LAN) สองเครือข่ายเข้าด้วยกัน หรือต่อ Token Ring LAN สองเครือข่ายเข้าด้วยกัน ซึ่งเราจะใช้ Hub หรือ Repeater มาต่อในกรณีนี้ไม่ได้ เพราะ Hub หรือ Repeater จะเป็นการขยายเครือข่ายระบบเดียวให้มีจุดเชื่อมต่อมากขึ้น ไม่ใช่การนำเครือข่ายสองเครือข่ายมาต่อเข้าด้วยกันเหมือนการใช้ Switch ทั้งนี้ Switch หรือ Bridge จะมีการทำงานในระดับชั้นที่ 2 คือ Data Link Layer ของ OSI คือมีความสามารถในการเชื่อมต่อฮาร์ดแวร์ที่ใช้รับส่งข้อมูลเข้าด้วยกัน และตรวจสอบข้อผิดพลาดของการรับส่งข้อมูลในระดับของฮาร์ดแวร์ การติดตั้งใช้งานจะคล้ายกับการติดตั้ง Hub คือไม่ยุ่งยากอะไรนัก โดยทั่วไปมักจะไม่จำเป็นต้องปรับแต่งค่าต่างๆที่มีอยู่ สามารถต่อใช้งานได้ทันที แต่ก็อาจจะกำหนดตัวแปรของค่าที่ใช้ควบคุม Switch หรือ Bridge ได้ถ้าต้องการ ซึ่งไม่ยากมากนัก ผู้ดูแลเครือข่ายขนาดเล็กๆก็สามารถทำได้เอง

2.3 Router

เป็นอุปกรณ์ที่ใช้เชื่อมต่อ LAN หลายๆเครือข่ายเข้าด้วยกันคล้ายกับ Switch แต่จะมีส่วนเพิ่มเติมขึ้นมา คือ Router สามารถเชื่อมต่อ LAN ที่ใช้โพรโตคอลในการรับส่งข้อมูลเหมือนกัน แต่ใช้ media หรือสายส่งต่างชนิดกันได้ เช่น เชื่อมต่อ Ethernet LAN ที่ใช้รับส่งข้อมูลแบบ Unshielded Twisted Pair (UTP) เข้ากับ Ethernet อีกเครือข่ายหนึ่งที่ใช้สายส่งข้อมูลแบบ coaxial cable ได้ Router มีการทำงานในระดับชั้นที่ 3 ของ OSI คือ Network Layer และสามารถรับส่งข้อมูลที่เป็นกลุ่มข้อมูลหรือ Frame จากต้นทางไปยังปลายทางได้ โดยเลือกหรือกำหนดเส้นทางที่ข้อมูลจะถูกส่งไป และแปลงข้อมูลให้เหมาะสมกับอุปกรณ์ฮาร์ดแวร์ที่ใช้รับส่งใน 2 ชั้นล่างถัดไปที่เชื่อมต่ออยู่ การติดตั้งใช้งาน Router จึงยุ่งยากกว่าการติดตั้ง Hub และ Switch โดยจะต้องมีความเข้าใจเกี่ยวกับระบบการกำหนด Network Address ของเครือข่ายแต่ละชนิดที่จะใช้ Router เชื่อมเครือข่ายเหล่านั้นเข้าด้วยกัน รวมทั้งชนิดของฮาร์ดแวร์ที่ใช้เชื่อมต่ออย่างละเอียด Router จึงมีราคาแพงกว่า Switch และ Hub ตามลำดับ

2.4 Gateway

เป็นอุปกรณ์ที่มีความสามารถสูงสุดในการเชื่อมต่อเครือข่ายต่างๆเข้าด้วยกัน โดยสามารถเชื่อมต่อ LAN หลายๆเครือข่ายที่ใช้โพรโตคอลต่างกัน และใช้ media หรือสายส่งข้อมูลต่างชนิดกันได้อย่างไม่มีขีดจำกัด ตัวอย่างเช่น เชื่อมต่อ Ethernet LAN ที่ใช้สายส่งข้อมูลแบบ Unshielded Twisted Pair เข้ากับ Token Ring LAN ได้ หากโพรโตคอลที่ใช้รับส่งข้อมูลของเครือข่ายของทั้งสองไม่เหมือนกัน Gateway ก็จะทำหน้าที่แปลงโพรโตคอลให้ตรงกับปลายทางและเหมาะสมกับอุปกรณ์ของฮาร์ดแวร์ที่แต่ละเครือข่ายใช้งานอยู่นั้นได้ด้วย อุปกรณ์ Gateway จึงมีราคาแพงและขั้นตอนการติดตั้งใช้งานจะซับซ้อนที่สุดในบรรดาอุปกรณ์เครือข่ายทั้งหมด และอุปกรณ์ที่มีจำหน่ายในท้องตลาดส่วนมากจะรวมฟังก์ชันการทำงานของ Router และ Gateway เข้าด้วยกันและเรียกชื่ออุปกรณ์นั้นว่า Router ซึ่งทำให้ผู้ใช้สับสนได้

ในการที่ Gateway จะสามารถส่งข้อมูลจากเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่งได้อย่างถูกต้องนั้น ตัวของ Gateway เองจะต้องสร้างตารางการส่งข้อมูล หรือที่เรียกว่า Routing Table ขึ้นมาในตัวของมัน ซึ่งตารางนี้จะบอกว่า Server ไหนอยู่เครือข่ายใดและอยู่ภายใต้ Gateway อะไร ตารางนี้จะต้องมีการปรับปรุงข้อมูลเองทุกระยะ เช่น ทุกๆ 30 วินาที เป็นต้น เมื่อเราให้ Server เริ่มทำงาน มันก็จะส่งข้อมูลเพื่อแจ้งให้ Gateway ที่คุม Server นั้นอยู่ทราบว่า Server นั้นอยู่ทราบว่า Server นี้เชื่อมต่อเข้ากับเครือข่ายแล้ว Server อื่นๆก็จะสามารถรับส่งข้อมูลกับมันผ่าน Gateway ได้อย่างถูกต้อง

สำหรับเครือข่ายขนาดใหญ่ อุปกรณ์ที่ทำหน้าที่เป็น Gateway อาจจะรวมเอาฟังก์ชันการทำงานที่เรียกว่า Firewall ไว้ในตัวด้วย เพื่อทำหน้าที่ป้องกันไม่ให้คอมพิวเตอร์ที่อยู่นอกเครือข่ายของบริษัทเข้ามาเชื่อมต่อลักลอบนำข้อมูลภายในออกไปได้โดยการกรองข้อมูลที่มาจาก IP address อื่นๆที่ไม่รู้จักทิ้งไป และมีการตรวจสอบผู้ใช้ด้วยการเข้ารหัสนอกเหนือจากการตรวจสอบรหัสผ่านตามปกติอีกชั้นหนึ่ง สำหรับการใส่ค่า Default Gateway ของเครื่องคอมพิวเตอร์ที่อยู่ในเครือข่ายเดียวกันนั้น ค่า IP address ที่เราใส่ลงไปจะเป็นค่า IP address ของอุปกรณ์ Gateway นี้นั่นเอง

2.5 Layer-3 Switch

ในการเชื่อมต่อเครือข่ายขนาดใหญ่เข้าด้วยกันนั้น การจัดแยกกลุ่มของเครือข่ายเป็นกลุ่มย่อย ๆจะทำโดยอาศัยอุปกรณ์ที่เรียกว่า Switch หรือ Bridge เป็นตัวแบ่งแยกปริมาณการรับส่งข้อมูลหรือ LAN traffic ของแต่ละกลุ่มออกจากกัน โดย Switch จะมีตารางการรับส่งข้อมูลเล็กๆภายในตัวเองที่จะคอยตรวจสอบว่าข้อมูลจากคอมพิวเตอร์ IP address ไหนส่งไปยังปลายทางที่ใดและอยู่ที่พอร์ตใดของ Switch เมื่อมีการส่งข้อมูลจาก IP address นั้นไปยังปลายทางที่เดิมอีก Switch จะส่งข้อมูลนั้นไปให้คอมพิวเตอร์ปลายทางโดยตรง ไม่ต้องส่งข้อมูลนี้กระจายไปให้พอร์ตอื่นๆ ทำให้ข้อมูลไม่ถูกส่งกระจายไปทั่วทั้งเครือข่าย และเป็นการแบ่งแยกข้อมูลของแต่ละกลุ่มออกจากกันไปในตัวด้วย ทั้งนี้ Switch เองซึ่งจะทำงานได้อย่างรวดเร็ว แต่การใช้ Switch เพียงอย่างเดียวเพื่อแบ่งแยก LAN traffic ที่ต้องมีการรับส่งข้อมูลข้ามกลุ่มย่อยๆเป็นปริมาณมากๆจะทำได้ไม่ดี จึงจำเป็นต้องอาศัยอุปกรณ์ที่เรียกว่า Router เข้าช่วย

อุปกรณ์ Router มีการทำงานในระดับ Layer 3 ของ OSI ซึ่งนอกจากจะมีราคาแพงแล้ว การทำงานยังช้ากว่า Switch อีกด้วย เนื่องจาก Router โดยทั่วๆไปมักจะใช้ไมโครโปรเซสเซอร์ควบคุมการทำงาน การแบ่งแยก LAN traffic จะทำด้วยซอฟต์แวร์ของ Router ซึ่งอาจทำให้เกิดความคับคั่งหรือเรียกว่าเกิดปัญหาคอขวด (bottleneck) ขึ้นได้ ทำให้ผู้ผลิตอุปกรณ์เครือข่ายต้องพัฒนา Layer-3 Switch ขึ้นมาแก้ปัญหานี้

Layer-3 Switch ได้รวมเอาข้อดีของ Switch และ Router เข้าไว้ด้วยกันในอุปกรณ์ตัวเดียว คือมีความสามารถในการแบ่งแยก LAN traffic ของเครือข่ายที่มีจำนวนเครื่องลูกข่ายมากๆได้เหมือนกับ Router และมีความเร็วในการแบ่งแยกข้อมูลเหมือน Switch โดยอาศัยฮาร์ดแวร์เป็นตัวทำงาน ซึ่งทำให้การแบ่งแยกข้อมูล LAN traffic ระหว่างกลุ่มย่อยๆในเครือข่ายมีความเร็วสูงในทั้ง Layer 2 และ Layer 3 ของ OSI นอกจากนี้ Layer-3 Switch ยังติดตั้งง่ายกว่าและมีราคาถูกกว่า Router อีกด้วย

ฟังก์ชันการทำงานหลักๆของ Layer-3 Switch มีสามด้านคือ

- การรับส่งข้อมูลระหว่างพอร์ตต่างๆเพื่อแบ่งแยก LAN traffic ของกลุ่มย่อยๆออกจากกัน หรือที่เรียกว่า Packet Switch function ซึ่งจะเทียบเท่ากับการใช้ Switch ใน Layer 2 ของ OSI
- อีกด้านหนึ่งก็คือฟังก์ชันการกำหนดเส้นทางของการรับส่งข้อมูลข้ามเครือข่ายที่เรียกว่า Route Processing function เหมือนกับการใช้ Router ใน Layer-3 ของ OSI แต่การทำงานในส่วนนี้จะใช้วงจรฮาร์ดแวร์กึ่งสำเร็จรูป ที่เรียกว่า Application-Specific Integrated Circuit (ASIC) ซึ่งจะมีความเร็วในการทำงานสูงกว่าการใช้ Router ทั่วๆไปที่เป็นซอฟต์แวร์อย่างเดียว
- และส่วนสุดท้ายคือฟังก์ชันในการจัดการระบบเครือข่าย เช่น การกำหนดระดับความสำเร็จของการรับส่งข้อมูลในเครือข่ายได้หลายระดับ (Traffic Prioritization) และการควบคุมความปลอดภัยในเครือข่ายโดยใช้ “รายชื่อผู้มีสิทธิเข้าใช้งาน” หรือ Access Control List (ACL) เพื่อกำหนดให้โพรโตคอลบางชนิดสามารถรับส่งผ่านหรือไม่ผ่านตัว Layer-3 Switch ก็ได้

ทั้งหมดนี้เป็นจุดเริ่มต้นของการรับส่งข้อมูลในเครือข่ายและการกำหนดมาตรฐานที่ใช้ ไม่ว่าจะเป็นมาตรฐานของ OSI 7 Layer Model และมาตรฐานการรับส่งข้อมูลของ TCP/IP รวมทั้งคุณสมบัติในแต่ละส่วนของมาตรฐานต่างๆ พร้อมกับอุปกรณ์เครือข่ายที่เกี่ยวข้อง ว่าจัดอยู่ในประเภทใดและทำงานอยู่ในชั้นไหนของมาตรฐานการรับส่งข้อมูล

บทที่ 3

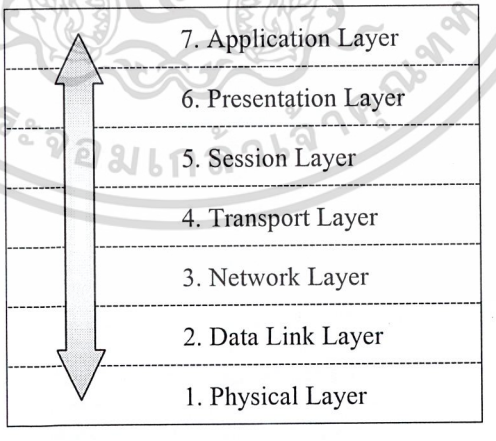
สถาปัตยกรรมเครือข่ายและระดับชั้นของโพรโตคอล

3.1 แบบอ้างอิงโอเอสไอ (OSI Reference Model)

ในการกล่าวถึงระบบเครือข่ายคอมพิวเตอร์ เราจะต้องใช้คำที่มีความหมายเฉพาะเจาะจงเกี่ยวกับระบบสื่อสารข้อมูล ซึ่งแต่ละคนอาจใช้ไม่เหมือนกัน การมีสิ่งไว้อ้างอิงจึงเป็นสิ่งที่จำเป็นในการทำความเข้าใจความหมายของคำที่ใช้ในการสื่อสารข้อมูล โมเดลแบบสถาปัตยกรรม (Architectural model) ที่ถูกพัฒนาโดยองค์การมาตรฐาน (International Standard Organization: ISO) ได้ถูกใช้อย่างแพร่หลายในการอธิบายถึงโครงสร้างและหน้าที่ของโพรโตคอลในการสื่อสารข้อมูล โมเดลดังกล่าวนี้มีชื่อว่าแบบอ้างอิงการเชื่อมต่อของระบบเปิด (Open System Interconnection: OSI Reference Model)

โมเดลนี้ประกอบไปด้วยเลเยอร์จำนวน 7 ชั้น ที่กำหนดหน้าที่ของโพรโตคอลการสื่อสารข้อมูล แต่ละเลเยอร์แสดงถึงหน้าที่เมื่อข้อมูลถูกส่งระหว่างแอพพลิเคชั่นที่ทำงานร่วมกันข้ามเครือข่าย ดังรูปที่ 3-1 ที่แสดงถึงแต่ละเลเยอร์พร้อมทั้งคำอธิบายเกี่ยวกับหน้าที่ของมันสั้นๆ มักเรียกโครสร้างนี้ว่า “สแต็ก” หรือ “โพรโตคอลสแต็ก”

หนึ่งเลเยอร์ไม่ได้กำหนดโพรโตคอลเดียว แต่กำหนดหน้าที่ในการสื่อสารข้อมูลที่สามารถทำได้ โดยโพรโตคอลจำนวนเท่าไรก็ได้ ดังนั้นแต่ละเลเยอร์จึงมีได้หลายโพรโตคอลซึ่งให้บริการที่เหมาะสมกับหน้าที่ของเลเยอร์นั้น ตัวอย่างเช่น โพรโตคอลไฟล์ทรานสเฟอร์และโพรโตคอลอิเลคทรอนิคส์เมล์ ซึ่งทั้งสองให้บริการแก่ผู้ใช้ จึงเป็นส่วนหนึ่งของเลเยอร์แอพพลิเคชั่น



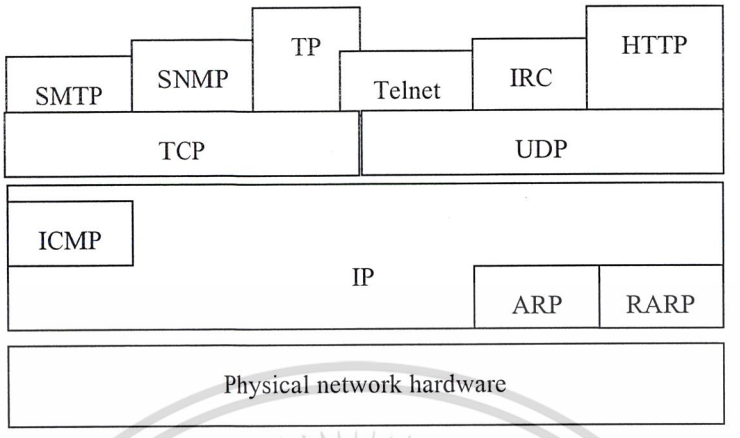
รูปที่ 3-1 แบบอ้างอิงโอเอสไอ

ทุกโพรโตคอลติดต่อสื่อสารกับเพียร์(Peer) ของมัน เพียร์คือการนำโพรโตคอลเดียวกันในเลเยอร์ที่เทียบเท่ากันบนระบบไกล (Remote system) มาใช้งานจริง เช่น โพรโตคอลไฟล์ทรานสเฟอร์ของเรา คือ เพียร์ของโพรโตคอลไฟล์ทรานสเฟอร์ที่ใช้อยู่อีกระบบหนึ่ง โดยนามธรรมแล้วแต่ละโพรโตคอลจะเกี่ยวข้องกับการสื่อสารกับเพียร์ของมัน โดยที่ไม่ต้องสนใจเลเยอร์ที่อยู่เหนือหรือต่ำกว่ามัน แต่การส่ง

ข้อมูลจริง ๆ จะเกี่ยวข้องกับทุกเลเยอร์ที่จะต้องส่งข้อมูลจากแอพพลิเคชั่นที่เราใช้ไปยังแอพพลิเคชั่นที่เทียบเท่ากันในระบบไกล เลเยอร์ที่สูงกว่าจะเกี่ยวข้องกับเลเยอร์ที่ต่ำกว่าในการส่งข้อมูลไปบนเครือข่ายที่มีอยู่ ข้อมูลถูกส่งมาจากสแต็กหนึ่งจากเลเยอร์หนึ่งสู่เลเยอร์ถัดไป จนกระทั่งข้อมูลถูกส่งไปบนเครือข่ายโดยโพรโตคอลในเลเยอร์ฟิสิคัล (Physical layer) ที่ปลายอีกข้างหนึ่งก็จะทำในลักษณะตรงกันข้ามคือ ข้อมูลจะถูกส่งไปบนสแต็กให้แก่แอพพลิเคชั่นที่ต้องการ แต่ละเลเยอร์ไม่จำเป็นต้องรู้ว่าเลเยอร์บนและล่างมันทำหน้าที่อะไร มันจำเป็นต้องรู้เฉพาะว่าทำอย่างไรจึงจะส่งข้อมูลไปให้ได้ การแบ่งหน้าที่ของการสื่อสารข้อมูลออกเป็นหลาย ๆ เลเยอร์จะช่วยลดผลกระทบที่จะต้องเปลี่ยนแปลงทั้งชุดโพรโตคอล เมื่อเทคโนโลยีเปลี่ยนไป ตัวอย่างเช่น เราสามารถเพิ่มแอพพลิเคชั่นเข้าไปใหม่ โดยไม่ต้องเปลี่ยนเครือข่ายในระดับฟิสิคัล หรือฮาร์ดแวร์ใหม่ สามารถติดตั้งได้โดยไม่ต้องเขียนซอฟแวร์ใหม่

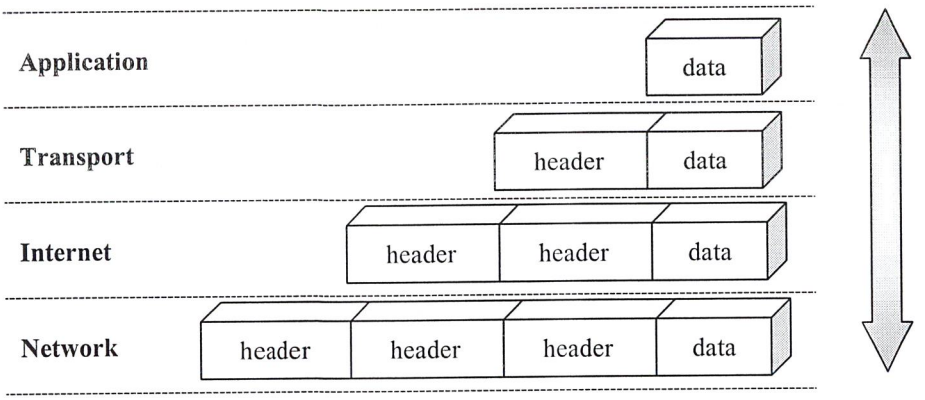
- แบบอ้างอิงโอเอสไอ ประกอบด้วยเลเยอร์ต่าง ๆ 7 ชั้น ดังต่อไปนี้คือ
  - Application Layer เป็นเลเยอร์ที่ติดต่อกับผู้ใช้ ประกอบด้วยแอพพลิเคชั่นโปรแกรมที่ใช้เครือข่าย
  - Presentation Layer การที่แอพพลิเคชั่นจะทำงานร่วมกันได้ จะต้องทำการตกลงว่าจะแทนข้อมูลกันอย่างไร เลเยอร์นี้จะกำหนดมาตรฐาน การแทนข้อมูลให้
  - Session Layer จัดการการติดต่อระหว่างแอพพลิเคชั่นที่ทำงานร่วมกัน
  - Transport Layer รับประกันว่าผู้รับจะต้องได้รับข้อมูลอย่างถูกต้องครบถ้วน
  - Network Layer จัดการการติดต่อข้ามเครือข่าย และแยกโพรโตคอลในเลเยอร์ที่สูงกว่าออกจากรายละเอียดของเครือข่าย
  - Data Link Layer ดูแลความน่าเชื่อถือ ของการส่งข้อมูลข้ามเครือข่ายฟิสิคัล
  - Physical Layer กำหนดคุณลักษณะของฮาร์ดแวร์ที่จำเป็นต้องใช้ในการพาสัญญาณการสื่อสารข้อมูล เช่น ระดับโวลเตจ จำนวนและตำแหน่งของขา (pins) ที่ใช้ในการอินเตอร์เฟส (Interface) ตัวอย่างมาตรฐานในเลเยอร์นี้คือ คอนเน็คเตอร์ในการอินเตอร์เฟส เช่น RS232C และ V3.5 และมาตรฐานในการเชื่อมสายของเครือข่าย เช่น ไอทริปเปิ้ลอี 802.3 (IEEE 802.3)

3.2 ชุดโพรโตคอลทีซีพี/ไอพี (TCP/IP Protocol suite)



รูปที่ 3-2 สถาปัตยกรรมทีซีพี/ไอพี

ประกอบด้วยเลเยอร์หลายชั้นเช่นเดียวกับแบบอ้างอิงโอเอสไอ ดังรูปที่ 3-2 ถ้าพิจารณาจากหน้าที่ของแต่ละเลเยอร์แล้ว 4 เลเยอร์ล่างของทีซีพี/ไอพี สามารถนำมาเปรียบเทียบกับ 4 เลเยอร์ล่างของแบบอ้างอิงโอเอสไอได้ โดยเลเยอร์ 1 และ 2 เป็นเลเยอร์ที่ใช้ร่วมกันได้ (Compatible) เพราะโอเอสไอกำหนดระบบตัวกลางหลายระบบในเลเยอร์ดังกล่าว และทีซีพี/ไอพี ก็ถูกออกแบบให้ใช้ตัวกลางใดก็ได้ (Medium independent) เมื่อพิจารณาเลเยอร์ 3 และ 4 คือ อินเทอร์เน็ตและทรานสปอร์ตของทีซีพี/ไอพี และเน็ตเวิร์คและทรานสปอร์ตของของโอเอสไอ จะเห็นได้ว่าโอเอสไอมีทางเลือกมากมายที่สามารถใช้ในเลเยอร์ทั้งสองนี้ได้ และบางตัวก็ทำหน้าที่ในลักษณะคล้ายคลึงกับทีซีพี/ไอพี ข้อแตกต่างสำคัญระหว่างทีซีพี/ไอพี กับโอเอสไอ คือ เลเยอร์แอพพลิเคชั่นของทีซีพี/ไอพี ซึ่งในโอเอสไอแล้ว จะเท่ากับ 3 เลเยอร์บน ถึงแม้จะมีความแตกต่างกันในการแบ่งเป็นเลเยอร์อยู่บ้าง แต่ลักษณะในการส่งข้อมูลของทีซีพี/ไอพีจะเหมือนกับโอเอสไอ คือข้อมูลจะถูกส่งลงมาจากสแต็กหรือส่งขึ้นไปบนสแต็ก ขณะที่ข้อมูลถูกส่งลงมาแต่ละเลเยอร์ในสแต็กก็จะเพิ่มข้อมูลควบคุม (Control information) เข้าไปเพื่อจะแน่ใจได้ว่าการส่งเกิดขึ้นอย่างเหมาะสม ข้อมูลควบคุมนี้เรียกว่า "เฮดเดอร์ (Header)" เพราะมันถูกใส่ไว้หน้าข้อมูลที่ถูกส่ง แต่ละเลเยอร์ปฏิบัติกับข้อมูลทั้งหมดที่มันได้รับมาจากเลเยอร์ที่สูงกว่าเหมือนกับเป็นข้อมูลจริง ๆ และเพิ่มเฮดเดอร์ของมันเองไว้ข้างหน้าข้อมูลทั้งหมดนั้น การทำดังกล่าวนี้เรียกว่า "เอ็นแคปซูเลชั่น (Encapsulation)" ดังรูปที่ 3-3 เมื่อผู้รับได้รับข้อมูลก็จะทำตรงกันข้ามกับที่กล่าวมา คือ แต่ละเลเยอร์จะนำเฮดเดอร์ออกมาก่อนที่จะส่งข้อมูลขึ้นไปยังเลเยอร์ที่สูงกว่า ข้อมูลก็จะไหลลงสแต็ก ข้อมูลที่ได้รับก็จะถูกแปลความหมายทั้งเฮดเดอร์และข้อมูล

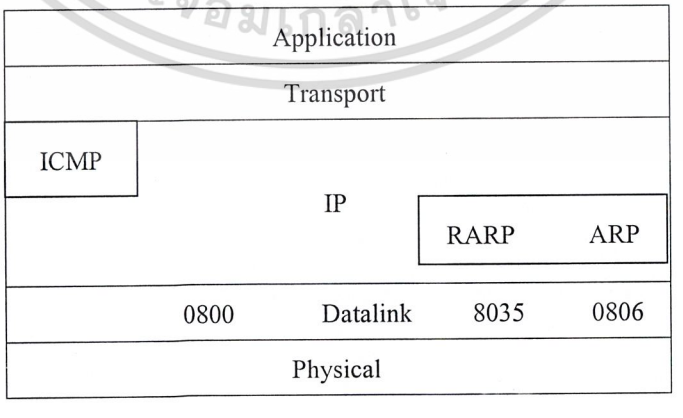


รูปที่ 3-3 การส่งผ่านข้อมูลระหว่างเลเยอร์

เลเยอร์ต่าง ๆ และโพรโตคอลที่เกี่ยวข้องในโพรโตคอลชุดทีซีพี/ไอพี มีดังนี้

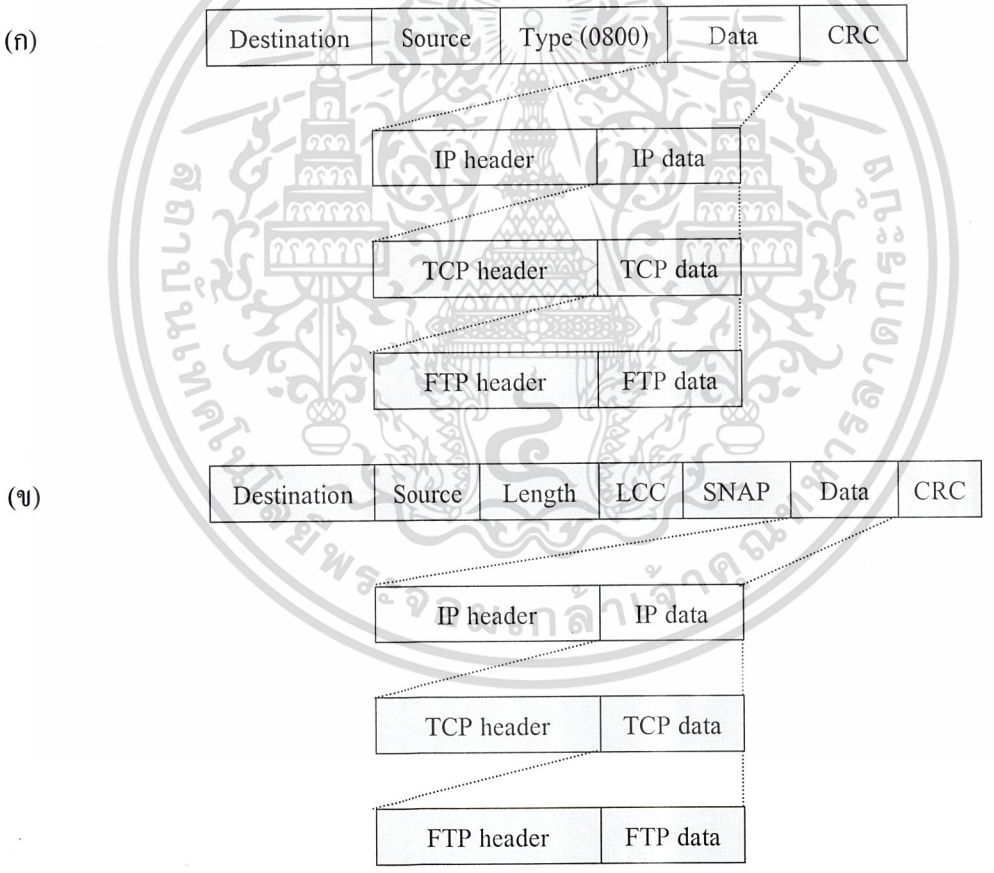
3.2.1 เลเยอร์เน็ตเวิร์คของทีซีพี/ไอพี

มีโพรโตคอลไอพี (Internet Protocol: IP) ทำหน้าที่พื้นฐานในการส่งโพรโตคอลชั้นสูงกว่าของทีซีพี/ไอพี ไปบนเครือข่ายฟิสิคัลทั้งหมด ซึ่งทำให้โพรโตคอลในเลเยอร์ที่สูงกว่าไม่จำเป็นต้องรู้อะไรเกี่ยวกับความสามารถของตัวกลางเลย และยังทำให้ผู้พัฒนาแอพพลิเคชั่นที่เขียนโปรแกรมเหนือเลเยอร์ทรานสปอร์ตทำงานได้ง่ายขึ้นด้วยเหตุผลเดียวกัน นอกจากนี้โพรโตคอลไอพียังเกี่ยวข้องกับการส่งข้อมูลไปยังเครื่องและเครือข่ายที่ต้องการอย่างถูกต้องหรือการเราท์หาเส้นทางนั่นเอง ไอพีเป็นบริการแบบไม่ต้องการก่อตั้งการเชื่อมต่อก่อน (Connectionless หรือ Connectionless datagram service) เพราะไม่มีการเรียก (call) หรือก่อตั้งวงจรเสมือนก่อนที่จะเริ่มส่งข้อมูล เนื่องจากแต่ละดาต้าแกรมมีข้อมูลทั้งหมดที่จำเป็นต้องใช้ในการเราท์เส้นทางอยู่แล้ว และระหว่างโหนด 2 โหนดก็ไม่มีเส้นทางที่เฉพาะเจาะจง ซึ่งทำให้ง่ายในการเราท์ใหม่ แม้จะเสียเวลาในการสวิตชิ่งเล็กน้อย เมื่อเครือข่ายเกิดข้อผิดพลาด ส่วนแอดเดรสปลายทางที่ใช้ มีทั้งคนเดียว (Unique) เป็นกลุ่ม (Multicast) หรือทุกคน (Broadcast)



รูปที่ 3-4 สถาปัตยกรรมไอพี

นอกจากไอพีแล้ว ยังมีโพรโตคอลอื่นในเลเยอร์นี้อีก คือ โพรโตคอลเออาร์พี (Address Resolution Protocol : ARP) , โพรโตคอลอาร์เออาร์พี (Reverse Address Resolution Protocol : RARP) , โพรโตคอลไอซีเอ็มพี (Internet Control Message Protocol : ICMP) ดังรูปที่ 3-4 โพรโตคอลเออาร์พีและอาร์เออาร์พีแสดงที่ตำแหน่งล่างของชั้นไอพี เพราะโพรโตคอล 2 ตัวนี้ ไม่ได้ใช้ไอพีและเป็นที่รู้จักโดยชั้นดาต้าลิงค์ที่สนับสนุนเหมือนเป็นโพรโตคอลที่แยกออกมาต่างหาก ไอซีเอ็มพีแสดงไว้ตำแหน่งบนของชั้นไอพี เพราะมันถูกส่งข้ามเครือข่ายโดยอยู่ในดาต้าแกรมไอพี ชั้นไอพีรู้ว่าเป็นดาต้าแกรมไอซีเอ็มพี โดยค่าโพรโตคอลที่เท่ากับ 1 ทั้งฟิลด์เฮดเดอร์และฟิลด์ข้อมูลของดาต้าแกรมไอพี จะกลายเป็น ฟิลด์ข้อมูลของเฟรมในขั้นดาต้าลิงค์ ลักษณะเช่นนี้เรียกว่าการเอ็นแคปซูเลชั่น ดังได้กล่าวไปแล้ว หรือบางครั้งก็เรียกการเอ็นเวลล็อปปิ้ง (Enveloping) ซึ่งฟิลด์ข้อมูลของดาต้าแกรมไอพีเองก็บรรจุเฮดเดอร์ของโพรโตคอลในชั้นที่สูงกว่าเช่นเดียวกัน ดังรูปที่ 3-5 ซึ่งแสดงการเอ็นแคปซูเลชั่น ในเฟรมอีเธอร์เน็ตทูและเฟรม SNAP



***รูปที่ 3-5** แสดงไอพี (ก) ในเฟรมอีเธอร์เน็ตทู และ (ข) ในเฟรม SNAP*

ลักษณะดาต้าแกรมไอพีเวอร์ชั่น 4 เป็นดังรูปที่ 3-6 ซึ่งแสดงในลักษณะกว้าง 32 บิต เมื่อดาต้าแกรมนี้ถูกส่งไปบนเครือข่าย ลำดับการส่งจะเป็นจากซ้ายบนไปขวาล่าง ซึ่งเรียกว่า ลำดับไบต์ของเครือข่าย

(Network byte order) และตัวเลขในทีซีพี/ไอพี จะถูกส่งโดยให้บิตที่สำคัญสูงสุด (Most significant octet) ไปก่อน แต่ละฟิลด์ในรูปที่ 3-6 สามารถอธิบายได้ดังต่อไปนี้

V.	IHL	TOS	Total length	
Identification			Flags	Fragment offset
Time to live		Protocol	Header checksum	
Source IP address				
Destination IP address				
Options				Padding
Data				

รูปที่ 3-6 รูปแบบของดาต้าแกรมไอพีเวอร์ชั่น 4

1. Version มีขนาด 4 บิต แสดงถึงเวอร์ชั่นของโพรโตคอลไอพี ขณะนี้คือเวอร์ชั่น 4

2. Internet Header Length มีขนาด 4 บิต แสดงถึงความยาวของเฮดเดอร์ หน่วยเป็น 32 บิตเวิร์ด ทำให้พบจุดเริ่มต้นของข้อมูลได้ง่าย ถ้ามีการใช้ฟิลด์ออพชั่น (Option Field) แต่ปกติจะมีค่าเป็น 5 คือไม่มีการใช้ฟิลด์ออพชัน

3. Type of Service มีขนาด 8 บิต ประกอบไปด้วยแฟล็กที่ใช้สำหรับทีโอเอส (TOS) และลำดับความสำคัญ โดย 3 บิตแรกใช้บ่งถึงลำดับความสำคัญ 8 ระดับ ซึ่งทำให้โหนดของไอพีรู้ว่าดาต้าแกรมใดมีความสำคัญมากกว่าดาต้าแกรมอื่น แต่เราท์เตอร์บางตัวก็ไม่สนใจแฟล็กนี้

   แฟล็กดี (D-flag) เป็นการร้องขอการเชื่อมต่อที่มีการเสียเวลา (delay) ต่ำ

   แฟล็กที (T-flag) บอกถึงความต้องการทรูพุธ (Throughput) สูง

   แฟล็กอาร์ (R-flag) บอกถึงความต้องการความเชื่อถือ (Reliability) สูง หมายถึงความน่าจะเป็นในการละทิ้ง (discard) ดาตาแกรมมมีต่ำกว่า

   แฟล็กซี (C-flag) บอกถึงความต้องการเสียค่าใช้จ่ายที่ต่ำกว่า

   บิตสุดท้ายไม่ใช้

4. Total Length มีขนาด 16 บิต เป็นการวัดทั้งเฮดเดอร์และข้อมูลในหน่วยออคเต็ด (Octets) ซึ่งทำให้คำนวณขนาดข้อมูลโดยคิดจากฟิลด์ Total Length และฟิลด์ IHL ได้ จะเห็นได้ว่าฟิลด์นี้มีขนาด 16 บิต ซึ่งหมายความว่าขนาดดาต้าแกรมที่ใหญ่ที่สุดมีขนาดเป็น 65,535 ออคเต็ด ซึ่งใหญ่กว่าที่เครือข่ายฟิสิคัลสนับสนุนมาก ถ้าดาต้าแกรมถูกแบ่งย่อย (Fragment) ค่าในฟิลด์สี่คือค่าใหม่ ไม่ใช่ค่าเก่าของขนาดดาต้าแกรม

5. Identification — มีขนาด 16 บิต บ่งถึงการแบ่งย่อยทั้งหมดของดาต้าแกรม มีลักษณะเฉพาะของใครของมัน (Unique) สำหรับแต่ละดาต้าแกรมใหม่ที่ถูกส่งโดยโฮส ฟิลด์นี้ไม่ใช่หมายถึงหมายเลขลำดับ (Sequence number) เพราะไอพีเป็นบริการแบบไม่ต้องก่อตั้งการติดต่อการติดต่อก่อนส่งข้อมูล แต่เป็นเพราะไอพีสนับสนุนบริการการเชื่อมต่อของเลเยอร์ทรานสปอร์ตได้หลายแบบ

6. Flags — มีขนาด 3 บิต ใช้ในการควบคุมการแบ่งย่อย ถ้าบิตลำดับต่ำมีค่าเป็น 0 หมายถึงเป็นส่วนสุดท้ายของดาต้าแกรมที่ถูกแบ่งย่อย บางครั้งจึงเรียกบิตนี้ว่า More flag หรือ บิต MF บิตกลางใช้บ่งถึงว่าดาต้าแกรมนี้ห้ามแบ่งย่อย จึงเรียกว่า Do not fragment หรือบิต DF บิตลำดับสูงไม่ถูกใช้

7. Fragment offset — มีขนาด 13 บิต ฟิลด์นี้ใช้ร่วมกับดาต้าแกรมที่ถูกแบ่งย่อย เพื่อบอกถึงตำแหน่งของข้อมูลในดาต้าแกรมเดิม วัดในหน่วย 8 ออคเต็ด ดังนั้นการแบ่งย่อยดาต้าแกรมจึงต้องทำในหน่วยนี้

8. Time to live — มีขนาด 8 บิต ฟิลด์นี้ถูกเซตโดยผู้ส่งดาต้าแกรมและจะถูกลดค่าโดยเราท์เตอร์เมื่อดาต้าแกรมผ่านมัน ถ้าฟิลด์ TTL ถูกลดค่าจนเป็น 0 ดาต้าแกรมนั้นจะถูกละทิ้งเพื่อป้องกันไม่ให้ดาต้าแกรมถูกเราท์เป็นลูป (loop) ตลอดไป

9. Protocol — มีขนาด 8 บิต บ่งถึงว่าดาต้าแกรมนั้นบรรจุโพรโตคอลใดของเลเยอร์ทรานสปอร์ต ค่าปกติคือ

17	ยูดีพี (UDP)
6	ทีซีพี (TCP)
1	ไอซีเอ็มพี (ICMP)
7	อีจีพี (EGP)
89	โอเอสพีเอฟ (OSPF)

10 Header checksum — มีขนาด 16 บิต ป้องกันเฉพาะเฮดเดอร์ไม่รวมข้อมูล เพราะจะต้องคำนวณใหม่ทุกครั้งที่ผ่านเราท์เตอร์ เพราะค่าฟิลด์ TTL Flags และ Fragment offset เปลี่ยนไป ถ้าคำนวณข้อมูลด้วยจะทำให้เสียเวลามากขึ้น

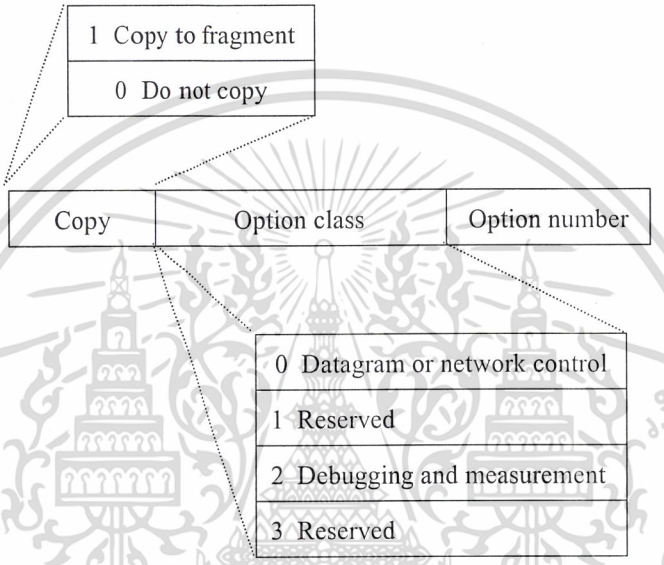
11. Source IP address — มีขนาด 32 บิต

12. Destination IP address — มีขนาด 32 บิต

13. Data — มีขนาดไม่แน่นอน ซึ่งฟิลด์นี้จะรวมเฮดเดอร์ของโพรโตคอลในเลเยอร์ที่สูงกว่าไว้กับข้อมูลจริง ๆ ด้วย

14. Padding มีขนาดไม่แน่นอน ค่าของฟิลด์นี้จะแทนด้วย 0 ใช้เพื่อต่อเฮดเดอร์ให้ครบ 32 บิตเวิร์ด ซึ่งทำให้ IHL บอกถึงจุดเริ่มต้นของข้อมูลได้ถูกต้อง เมื่อมีการใช้ฟิลด์ Options ซึ่งความยาวไม่คงที่

15.Options สนับสนุนการดีบัก (Debugging) การวัด (Measurement) และความปลอดภัย(Security) ซึ่งสามารถมีหลายออปชั่นได้ในดาต้าแกรมเดียว ดังรูปที่ 3-7 ซึ่งฟิลด์นี้ประกอบด้วย



รูปที่ 3-7 ออปชั่นในการวัดและความปลอดภัย

15.1 Copy มีขนาด 1 บิต ใช้ตัดสินในว่าออปชั่นจะอยู่ในทุกส่วนดาต้าแกรมที่ถูกแบ่งหรือไม่ ถ้ามีค่าเป็น 0 หมายถึงออปชั่นจะปรากฏในส่วนย่อยแรก (fragment) เท่านั้น ตัวอย่างของออปชั่นที่จำเป็นต้องคัดลอก (Copy) ให้แก่ทุกส่วนย่อยคือออปชั่นเกี่ยวกับความปลอดภัย

15.2 Option Class มีขนาด 2 บิต บอกถึงคลาสของออปชั่น ซึ่งออปชั่นไทม์แสมป์ (Time stamp option) มีคลาสเป็น 2 นอกจากนั้นคลาสปกติจะเป็น 0

15.3 Option numbers มีขนาด 5 บิต

15.3.1 Security: คือ ออปชั่น 2 มีการกำหนดระดับของดาต้าแกรมจากธรรมดาไปจนถึงดาต้าแกรมที่เป็นความลับมาก ซึ่งช่วยให้เราท์เตอร์รู้ว่าดาต้าแกรมใดบรรจุข้อมูลที่สำคัญและป้องกันข้อมูลเหล่านั้นไม่ให้ออกจากสิ่งแวดล้อมที่ปลอดภัย

15.3.2 Time stamp: คือ ออปชั่น 4 ทำให้ดาต้าแกรมที่ถกส่งไปในเครือข่ายสามารถรวบรวมไทม์แสตมป์จากแต่ละเราท์เตอร์ที่มันผ่าน ซึ่งเราสามารถนำสิ่งที่ได้นี้มาใช้ประเมินการเสียเวลาและความเปลี่ยนแปลงในเครือข่ายของเราท์เตอร์ได้

15.3.3 Loose source route: คือ ออปชั่น 3 ในการกำหนดค่าของเราท์เตอร์เพื่อให้ดาต้าแกรมผ่านตามไอพีแอดเดรสของเราท์เตอร์ ออปชั่นนี้จะอนุญาตให้ใช้เราท์เตอร์อื่นได้ในระหว่างลิสต์ของเราท์เตอร์ที่ถูกกำหนด

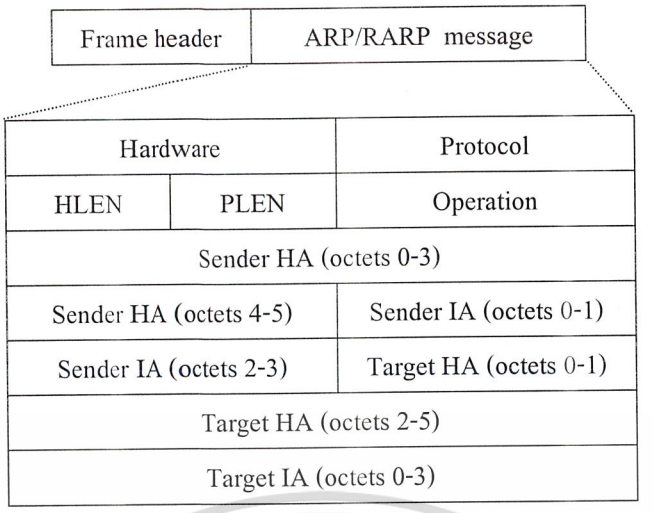
15.3.4 Record route: คือ ออปชั่น 7 ทำให้แต่ละเราท์เตอร์ใส่ไอพีแอดเดรสของมันในฟิลด์ออปชั่นของดาต้าแกรมเมื่อดาต้าแกรมเดินทางผ่านเครือข่าย ซึ่งทำให้ค้นหาทางที่ดาต้าแกรมใช้ในการไปถึงโฮสหรือเราท์เตอร์ใด ๆ ได้

15.3.5 Strict source route: คือ ออปชั่น 9 คล้ายกับ Loose source route ยกเว้นว่าเฉพาะเราท์เตอร์ที่กำหนดในลิสต์เท่านั้น ที่สามารถใช้ได้

โพรโตคอลเออาร์พี (ARP: Address Resolution Protocol)

การ์ดแลนส่งและรับเฟรมโดยใช้แมคแอดเดรส (MAC address) แต่ทีซีพี/ไอพีใช้ไอพีแอดเดรสที่กำหนดโดยผู้ดูแลระบบเครือข่าย ณ เวลาติดตั้ง ซึ่งไม่มีความสัมพันธ์โดยตรงกับแมคแอดเดรส การสื่อสารแบบปลายถึงปลาย (end-to-end) ใช้ไอพีแอดเดรส แต่แบบฮ็อพถึงฮ็อพ (hop-to-hop) ใช้แมคแอดเดรส ดังนั้นเลเยอร์แมค (MAC) จึงต้องการแมคแอดเดรสของฮ็อพถัดไประหว่างไอพีแอดเดรสต้นทาง และปลายทาง เราสามารถรู้แมคแอดเดรสของไอพีแอดเดรสที่กำหนดโดยใช้โพรโตคอลเออาร์พี แต่จะใช้ได้เฉพาะบนตัวกลางที่สนับสนุนการบรอดคาสท์เท่านั้น และแต่ละโหนดจะมีแคช (cache) ที่เรียกว่าแคชเออาร์พี ซึ่งเก็บไอพีแอดเดรส และแมคแอดเดรสที่สัมพันธ์กัน เมื่อไอพีจะส่งดาต้าแกรมไปยังไอพีแอดเดรสอื่น มันจะไปหาแมคแอดเดรสของไอพีแอดเดรสที่เลเยอร์ดาต้าลิงค์จำเป็นต้องใช้ในการส่งจากแคชเออาร์พีก่อน ถ้าไม่พบมันจะพยายามหาแมคแอดเดรสจากไอพีแอดเดรสโดยใช้โพรโตคอลเออาร์พี ซึ่งการทำดังกล่าวนี้ โพรโตคอลเออาร์พีจะส่งดาต้าแกรมร้องขอ (ARP request datagram) ไปยังทุกการ์ดแลน โดยใช้แมคแอดเดรสสำหรับการบรอดคาสท์ (0xFFFF_FFFF_FFFF) พร้อมทั้งไอพีแอดเดรสของแมคแอดเดรสที่ต้องการ การ์ดในเครือข่ายจะอ่านคำขอนี้ และทุกการ์ดที่รู้คำตอบจะตอบกลับ (ARP response) ซึ่งเมื่อได้รับคำตอบ คำตอบนี้ก็จะถูกเก็บไว้ในแคชเพื่อใช้ต่อไปในอนาคต แต่ถ้าไม่ได้รับคำตอบภายในเวลาไม่กี่วินาทีเออาร์พีรีเควสก็จะถูกส่งซ้ำ เพราะเออาร์พีอาจถูกละทิ้งได้เนื่องจากความผิดพลาดในการส่งหรือความคับคั่งของบริดจ์ (bridge) เพื่อลดความจำเป็นในการบรอดคาสท์เออาร์พี โหนดที่ตอบกลับจะคัดลอกไอพีแอดเดรสและแมคแอดเดรสของผู้ร้องขอเก็บไว้ในแคชเออาร์พี โหนดที่ตอบกลับจะคัดลอกไอพีแอดเดรส ซึ่งทำให้ป้องกันการเกิดข้อมูลจำนวนมาก (flooding) วิ่งไปทั่วทั้งระบบได้

รูปแบบของดาต้าแกรมเออาร์พีแสดงดังรูปที่ 3-8 สามารถใช้กับเครือข่ายแบบใดก็ได้ ไม่เฉพาะทีซีพี/ไอพีเท่านั้น แต่ต้องมีตัวกลางที่สามารถส่งเฟรมบรอดคาสท์ได้ เออาร์พีทำงานโดยตรงบนเลเยอร์ดาต้าลิงค์ ดังนั้นจึงถูกเอ็นแคปซูเลชั่น โดยเฟรมดาต้าลิงค์เท่านั้น ทำให้มันต้องการฟิลด์ Ethernet type ของมันเอง คือ 0x0806



รูปที่ 3-8 รูปแบบดาต้าแกรมเออาร์พี

ฟิลด์ในดาต้าแกรมเออาร์พีประกอบด้วย

1. Hardware บอกถึงชนิดของฮาร์ดแวร์ ที่ใช้ในเครือข่ายซึ่งสร้างดาต้าแกรมนี้ขึ้นมา **ชนิดที่ใช้ได้ คือ**

Type	Description
1	Ethernet (10 Mbps)
2	Experimental Ethernet (3 Mbps)
3	Amateur radio AX.25
4	Proteon ProNET Token Ring
5	Chaos
6	IEEE 802 networks
7	ARCNET
8	Hyperchannel
9	Lanstar
10	Autonet Short address
11	Local Talk
12	LocalNet (IBM PCNet or Sytek Inc. LocalNet)

2. Protocol แสดงถึงโพรโตคอลที่ร้องขอ ค่าที่ใช้ในฟิลด์นี้จะเหมือนกับฟิลด์Ethernet type ในเฟรมอีเธอร์เน็ต ซึ่งก็คือ 0x0800 สำหรับ IP

3. HLEN บอกถึงความยาวของฮาร์ดแวร์แอดเดรสในหน่วยออคเต็ด ปกติจะมีค่าเป็น 6 สำหรับแมคแอดเดรสของแลยไอทริปเปิ้ลอี

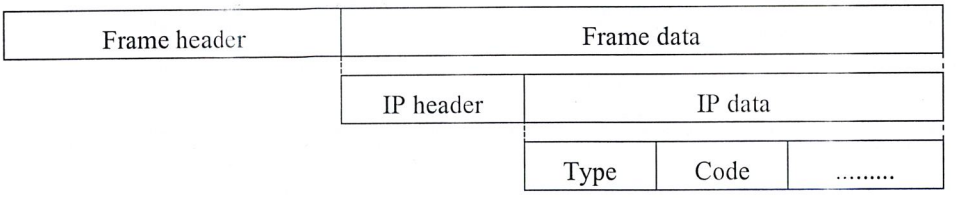
4.PLEN	บอกถึงความยาวของแอดเดรสในเลเยอร์เน็ตเวิร์คในหน่วยออกเต็ด ปกติมีค่าเป็น 4 สำหรับ IP
5. Operation	มีค่าเป็น 1 สำหรับเออาร์พีรีเควส และ 2 สำหรับเออาร์พีเรสปอนด์ (ARP response) และยังใช้กับอาร์เออาร์พีด้วย โดยมีค่าเป็น 3 สำหรับอาร์เออาร์พีรีเควส และ 4 สำหรับอาร์เออาร์พีเรสปอนด์
6. Addresses	ประกอบด้วยฮาร์ดแวร์แอดเดรสของผู้ส่ง (แมคแอดเดรสต้นทาง) , ไอพีแอดเดรสต้นทาง , ฮาร์ดแวร์แอดเดรสเป้าหมาย (แมคแอดเดรสปลายทาง) , และไอพีแอดเดรสปลายทาง

โพรโตคอลอาร์เออาร์พี (RARP: Reverse Address Resolution Protocol)

ใช้สำหรับอุปกรณ์ที่ไม่สามารถเก็บไอพีแอดเดรสของตัวเองได้ เช่น เวิร์คสเตชั่นที่ไม่มีฮาร์ดดิสค์อาร์เออาร์พีทำงานในลักษณะตรงกันข้ามกับเออาร์พี คือ หาไอพีแอดเดรสจากแมคแอดเดรสที่กำหนด อาร์เออาร์พีทำงานโดยตรงกับเลเยอร์ดาต้าลิงค์โดยมีหมายเลขชนิดอีเธอร์เน็ตเท่ากับ 0x8035 โหนดที่ทำหน้าที่เป็นเซิร์ฟเวอร์อาร์เออาร์พี (RARP server) ที่พบแมคแอดเดรสที่กำหนดจะตอบกลับโดยอาร์เออาร์พีเรสปอนด์ พร้อมทั้งไอพีแอดเดรสที่ต้องการ รูปแบบของดาต้าแกรมจะเหมือนเออาร์พีแต่ฟิลด์โอเปอร์เรชั่น จะใช้ค่าเป็น 3 สำหรับรีเควส และ 4 สำหรับเรสปอนด์ ถึงแม้ว่าอาร์เออาร์พีจะทำงานได้ดี แต่ก็มีข้อจำกัดมาก ในทางปฏิบัติจึงถูกแทนที่โดยโพรโตคอลบูท (Boot Protocol : BOOTP) ซึ่งสามารถทำงานผ่านเราท์เตอร์และหาข้อมูลที่เป็นประโยชน์ได้มากกว่าอาร์เออาร์พี เมื่อเวิร์คสเตชั่นที่ไม่มีฮาร์ดดิสค์ทำการบูท

โพรโตคอลไอซีเอ็มพี (ICMP: Internet Control Message Protocol)

ถึงแม้ว่าไอพีจะไม่รับรองในการส่งข้อมูล แต่ไอซีเอ็มพีซึ่งใช้ได้ในไอพีสามารถสร้างแมสเสจ(Message) เกี่ยวกับความผิดพลาดเพื่อช่วยเลเยอร์ไอพีในการให้บริการส่งข้อมูลให้ดีที่สุด และยังช่วยผู้ดูและระบบในการวิเคราะห์หาสาเหตุเกี่ยวกับการทำงานของเครือข่าย ไอซีเอ็มพีใช้ดาต้าแกรมไอพีในการส่งแมสเสจระหว่างโหนด แมสเสจแสดงข้อผิดพลาดของไอซีเอ็มพีจะถูกสร้างโดยโหนดที่พบว่ามีปัญหาในการส่งเกิดขึ้น และจะส่งแมสเสจนี้กลับไปยังแอดเดรสที่เป็นต้นทางของดาต้าแกรมที่ทำให้เกิดปัญหา รูปที่ 3-9 แสดงถึงแมสเสจไอซีเอ็มพีที่ถูกเอ็นแคปซูเลทในดาต้าแกรมไอพีและแมสเสจแบบต่าง ๆ ที่เป็นไปได้ ไอซีเอ็มพีมีหมายเลขโพรโตคอล (Protocol number) ของตัวเอง ซึ่งเท่ากับ 1 ทำให้ไอพีรู้ว่าได้รับไอซีเอ็มพี ถึงแม้ว่าไอซีเอ็มพีจะใช้เลเยอร์ไอพี แต่มันถูกมองว่าอยู่ภายในไอพีทั้งหมด เพราะไม่ได้ให้บริการแก่เลเยอร์ที่อยู่เหนือมัน



รูปที่ 3-9 แสดงไอซีเอ็มพีชนิดต่าง ๆ ถูกเอ็นแคปซูเลทในไอพี

รูปแบบพื้นฐานของดาต้าแกรมไอซีเอ็มพีแสดงได้ดังรูปที่ 3-10 แต่ฟิลด์ต่าง ๆ จะต่างกัน ซึ่งขึ้นอยู่กับชนิดที่ใช้อยู่ ฟิลด์ไทป์ (Type) บ่งถึงชนิดของแมสเสจไอซีเอ็มพี ฟิลด์โค้ด (Code) ใช้แสดงถึงข่าวสารที่ละเอียดมากขึ้น ฟิลด์เช็คซัม (Checksum) ใช้เพราะไอพีไม่ได้ป้องกันข้อมูลของมันด้วยเช็คซัม แต่เมื่อทำงานบนเครือข่ายฟิสิคัล ซึ่งมีเฟรมเช็คซีเควน (Frame Check Sequence : FCS) เช็คซัมของไอซีเอ็มพีอาจเท่ากับ 0 หมายถึงไม่ถูกคำนวณ

Type	Code	Checksum
Context specific		
Context specific		
Context specific		

รูปที่ 3-10 เฮดเดอร์พื้นฐานของโพรโตคอลไอซีเอ็มพี

1. ไอซีเอ็มพีชนิด 0 และ 8 - echo

ใช้เพื่อจุดประสงค์ในการหาสาเหตุ ถูกสร้างจากโปรแกรมอรรถประโยชน์ (Utility program) ที่รู้จักกันดีคือ ping ซึ่งจะส่งไอซีเอ็มพีชนิด 8 ไปยังโหนดและคาดว่าจะได้รับไอซีเอ็มพีชนิด 0 ตอบกลับมา รูปแบบของไอซีเอ็มพีสองชนิดนี้เป็นดังรูปที่ 3-11

Type	Code	Checksum
Identification		Sequence No.
Optional data		
.....		

รูปที่ 3-11 รูปแบบดาต้าแกรมไอซีเอ็มพีชนิด Echo

ฟิลด์ไอเดนติไฟเออร์ (Identfier) และ ซีเควนนัมเบอร์ (Sequence number) ใช้ในการทำให้ดาต้าแกรมนี้แตกต่างจากดาต้าแกรมอื่น ถ้ามีข้อมูลส่งในฟิลด์ออปชั่นแนลดาต้า (Optional data) มันจะต้องถูกส่งกลับในการตอบกลับ

2. ไอซีเอ็มพีชนิด 3 – destination unreachable

ถ้าเราท์เตอร์ไม่สามารถส่งดาต้าแกรมได้ มันจะส่งแมสเสจไอซีเอ็มพีชนิดเดสติเนชั่นอันริชเอเบิล (destination unreachable) เพื่อบอกถึงสาเหตุ ฟิลด์โค้ด (Code) จะถูกใช้บอกถึงสาเหตุ ส่วนเฮดเดอร์อินเตอร์เน็ตรวมทั้งดาต้าแกรมพรีฟิกซ์ (Datagram prefix) 64 บิต จะใช้ในการบอกถึงดาต้าแกรมที่เป็นสาเหตุของปัญหาดังรูปที่ 3-12

Type	Code	Checksum
Unused (must be 0)		
Internet header + 64 bits of datagram prefix		
.....		

Code value	Meaning
0	Network unreachable
1	Host unreachable
2	Protocol unreachable
3	Port unreachable
4	Fragmentation needed and the do not fragment bit set
5	Source route failed
7	Destination host unknown
11	Network unreachable for type of service
12	Host unreachable for type of service
13	Communication administratively prohibited e.g. firewall blocked
14	Host precedence violation
15	Precedence cut-off in effect

รูปที่ 3-12 ค่าเดสติเนชั่นอันริชเอเบิล ในฟิลด์โค้ด

ซึ่งความหมายของแต่ละเหตุผลเป็นดังนี้

2.1 Network unreachable : หมายถึงเครือข่ายที่ระบุในไอพีแอดเดรสไม่สามารถพบได้ ควรจะตรวจที่ไอพีแอดเดรส หรืออาจเกิดความผิดพลาดในตารางเราท์เส้นทาง (routing table) ของเราท์เตอร์ ระหว่างทาง แมสเสจแสดงข้อผิดพลาดนี้ถูกสร้างโดยเราท์เตอร์เท่านั้น จุดที่เกิดข้อผิดพลาดจะทราบได้จากแอดเดรสต้นทางในเฮดเดอร์ของไอพีที่บรรจุแมสเสจของไอซีเอ็มพี ซึ่งก็คือ เราท์เตอร์ที่เจอข้อผิดพลาดนั่นเอง

2.2 Host unreachable : ดาต้าแกรมที่เป็นสาเหตุของความผิดพลาดได้ไปถึงเราท์เตอร์ที่ต่อตรงกับเครือข่ายปลายทางแล้ว แต่เมื่อเราท์เตอร์พยายามที่จะส่งดาต้าแกรมมันกลับไม่สามารถสื่อสารกับโฮสนั้นได้ ซึ่งอาจเกิดจากเออาร์พีล้มเหลวในดาต้าแกรมแรก โฮสดาวน์ หรือเหตุอื่นใดก็ตาม ซึ่งอาจเพราะไม่มีไอพีแอดเดรสนั้น เช่นเดียวกับ Network unreachable คือ แมสเสจนี้จะถูกสร้างจากเราท์เตอร์เท่านั้น จุดที่เกิดข้อผิดพลาดก็คือ จุดที่เป็นแอดเดรสต้นทางในเฮดเดอร์ไอพีที่บรรจุแมสเสจไอซีเอ็มพี ซึ่งก็คือเราท์เตอร์ที่พบข้อผิดพลาดนั่นเอง

2.3 Protocol unreachable : ในกรณีนี้ดาต้าแกรมได้ไปถึงโฮสปลายทางแล้ว แต่ไม่สามารถใช้โพรโตคอลที่ถูกพามาในดาต้าแกรมไอพีได้ ซึ่งพบได้ไม่บ่อยนัก แต่ก็เป็นไปได้ถ้าเครื่องระยะไกล (Remote machine) นั้นถูกกำหนดติดตั้งผิด

2.4 Port unreachable : ส่งจากโฮสเพื่อบอกว่าบริการในเลเยอร์แอพพลิเคชั่นที่โฮสระยะไกลที่การติดต่อกำลังถูกก่อตั้งนั้น ไม่พร้อมที่จะใช้งานได้ (not available) ในแต่ละบริการของแอพพลิเคชั่น (application service) ที่โฮสนั้นจะถูกอีเนเบิล (enable) และดิสเอเบิล (disable) ขณะโฮสเริ่มทำงานโดยไฟล์คอนฟิคจูเรชั่น (Configuration file) ดังนั้นเมื่อเกิดข้อผิดพลาดขึ้นก็ควรจะตรวจสอบที่ไฟล์นี้

2.5 Fragmentation needed and the do not fragment bit set : ปกติจะมาจากเราท์เตอร์บ่งถึงความจำเป็นที่จะต้องย่อยดาต้าแกรม แต่บิต Do not fragment หรือบิต DF ในฟิลด์แฟลก (Flags) ของเฮดเดอร์ไอพีไม่ยอมให้ทำ

2.6 Source route failed : ออปชั่นสำหรับการจัดการในไอพีอนุญาตให้ดาต้าแกรมไอพีไปตามเส้นทางที่กำหนดไว้ก่อนได้ แมสเสจนี้จะบ่งถึงข้อผิดพลาดที่ดาต้าแกรมไปตามเส้นทางที่กำหนดนี้ไม่สำเร็จ

2.7 Destination host unknown : ถูกสร้างขึ้นมาจากเราท์เตอร์ เมื่อรู้จากซอฟท์แวร์สำหรับเชื่อมเลเยอร์ (link layer software) ว่าไม่มีโฮสปลายทาง

2.8 Network unreachable for type of service : ถูกสร้างเมื่อเราท์เตอร์ไม่สามารถส่งแพ็คเก็ตต่อไปได้ เนื่องมาจากการกรองแพ็คเก็ต ตัวอย่างเช่น เหตุผลในความปลอดภัย หรือการคิดค่าบริการจากระบบ (Local charging)

2.9 Communication administratively prohibited e.g. firewall blocked : ถูกสร้างเมื่อเราท์เตอร์ไม่สามารถส่งแพ็คเก็ตต่อไปได้ เนื่องมาจากการกรองแพ็คเก็ต ตัวอย่างเช่น เหตุผลในความปลอดภัย หรือการคิดค่าบริการจากระบบ (Local charging)

2.10 Host precedence violation : ถูกส่งจากเราท์เตอร์ที่เป็นฮ็อพแรกไปยังโฮส เพื่อบอกว่าไม่สามารถกำหนดลำดับความสำคัญดังที่ร้องขอได้ สำหรับโฮสต้นทางหรือปลายทาง เครือข่ายต้นทางหรือปลายทาง โพรโตคอลในเลเยอร์ที่สูงกว่า และพอร์ตต้นทางหรือปลายทาง

2.11 Precedence cut-off in effect : บ่งถึงว่าผู้ดูแลระบบเครือข่ายได้กำหนดค่าระดับความสำคัญที่เส้นทางนี้ต้องการไว้ต่ำสุด คือ ดาต้าแกรมถูกส่งด้วยลำดับความสำคัญที่ต่ำกว่าที่ต้องการ

3. ไอซีเอ็มพีชนิด 4 และ โค้ด 0 – source quence

รูปแบบของไอซีเอ็มพีชนิดนี้ จะเหมือนกับไอซีเอ็มพีชนิดเดสติเนชั่นอันริชเอเบิล (Destination unreachable) แต่จะมีไทป์เป็น 4 และโค้ดเป็น 0 เท่านั้น ไอซีเอ็มพีแบบนี้ใช้ในการทำโฟล์วคอนโทรล (Flow Control) เราท์เตอร์ที่พบว่าเครือข่ายหรือโปรเซสเซอร์ถูกใช้งานหนักเกินไป จะส่งเมสเสจไอซีเอ็มพีนี้ไปยังโฮสที่เป็นสาเหตุหลักของการใช้งานมาก ซึ่งโฮสดังกล่าวเมื่อได้รับแมสเสจก็จะลดอัตรการสร้างแพ็คเก็ตไปสู่ปลายทางที่ระบุมา

4. ไอซีเอ็มพีชนิด 5 – route change request

มีรูปแบบดังรูปที่ 3-13 ถูกใช้โดยเราท์เตอร์เท่านั้น สำหรับเราท์เตอร์ที่รู้ว่ามันไม่ใช่เราท์เตอร์ที่เหมาะสมที่สุดสำหรับการไปถึงปลายทางที่กำหนดก็จะใช้แมสเสจนี้แนะนำเราท์เตอรีที่เหมาะสมกว่าแก่ผู้ส่ง คือไอพีแอดเดรสต้นทางของดาต้าแกรม เพื่อความต่อเนื่องในการส่งข้อมูลเราท์เตอร์จะส่งดาต้าแกรมที่เป็นสาเหตุให้เกิดแมสแสจนี้ไปยังเราท์เตอร์ที่เชื่อว่าเข้าถึงเส้นทางที่ดีกว่าด้วย

(ก)

Type	Code	Checksum
Internet address of a more suitable router		
Internet header +64 bits of datagram prefix		
.....		

(ข)

Code value	Meaning
0	Redirect datagrams to go to that network
1	Redirect datagrams to reach that host
2	Redirect datagrams for that network with that TOS
3	Redirect datagrams for that host with that TOS

รูปที่ *3-13 (ก) รูปแบบของแมสเสจ route change request และ (ข) ค่าในฟิลด์โค้ดที่แมสเสจประเภทนี้ใช้*

5. ไอซีเอ็มพีชนิด 9 – router advertisement

ทำให้เราท์เตอร์ประกาศตัวกับโฮสบนระบบเครือข่ายได้ดังรูปที่ 3-14 ซึ่งจะส่งทุก ๆ 7–10 นาที หรือเพื่อตอบสนองโฮสจากแมสเสจไอซีเอ็มพีชนิด 10 แมสเสจนี้ไม่มีข้อมูลว่าจะติดต่เราท์เตอร์นี้ผ่านเส้นทางไหน เพราะฉะนั้น ถ้าโฮสเลือกเราท์เตอร์แรกไม่เหมาะสม ก็จะได้รับแมสเสจไอซีเอ็มพีชนิด 5

Type	Code	Checksum
Num addrs	Addr entry size	Life time
Router address [1]		
Preference level [1]		
Router address [2]		
Preference level [2]		
.....		

รูปที่ *3-14 รูปแบบแมสเสจไอซีเอ็มพี Router advertisement*

6. ไอซีเอ็มพีชนิด 10 – router solicitation

สามารถถูกส่งได้โดยโฮสเวลาใดก็ได้ แต่มักจะเป็นตอนเปิดเครื่อง (Strat-up) เพื่อหาเราท์เตอร์ที่สามารถใช้ได้ในเครือข่ายที่โฮสอยู่ เราท์เตอร์จะตอบสนองแมสเสจนี้ด้วยแมสเสจไอซีเอ็มพีชนิด 9 (router advertisement response) หลังจากส่งแมสเสจนี้ไปแล้ว โฮสจะรอ unsolicited advertisement จากเราท์เตอร์ทุก ๆ 7-10 นาที รูปแบบของแมสเสจชนิดนี้เป็นดังรูปที่ 3-15

Type	Code	Checksum
Reserved		

รูปที่ 3-15 *รูปแบบแมสเสจไอซีเอ็มพี Router solicitation*

7. ไอซีเอ็มพีชนิด 11 – time exceeded for datagram

รูปแบบของไอซีเอ็มพีชนิดนี้ จะเหมือนกับแมสเสจไอซีเอ็มพี destination unreachable ซึ่งแมสเสจชนิดนี้จะถูกส่งในสถานการณ์ดังต่อไปนี้

7.1 จากเราท์เตอร์ : ใช้บอกว่าค่าฟิลด์ TTL ในเฮดเดอร์ไอพีถูกลดจนมีค่าเป็น 0 ในกรณีนี้ค่าโค้ด จะเป็น 0 ทำให้ดาต้าแกรมถูกละทิ้งก่อนถึงปลายทาง ซึ่งส่วนใหญ่จะแสดงถึงว่าฟิลด์ TTL ที่ตั้งค่าเอาไว้ตอนเริ่มต้นไม่เหมาะสม หรือเกิดจากความเสียหายที่เกิดขึ้นกับเครือข่าย ซึ่งทำให้ความยาวของเส้นทางไม่ปกติ

7.2 จากโหนดปลาย : ค่าในฟิลด์โค้ดจะเป็น 1 บอกถึงการพยายามรวมส่วนย่อยเพื่อให้เป็นดาต้าแกรมเดิมไม่สำเร็จ การรวมดาต้าแกรมอีกครั้งโดยใช้เวลามากจนเกินไป ถ้าเกิดขึ้นไม่บ่อยนัก ก็ถือว่าเป็นปัญหาร้ายแรงแต่อย่างใด

8. ไอซีเอ็มพีชนิด 12 – parameter problem

ใช้อาร์กิวเมนท์ผิดในฟิลด์ออปชั่น ของเฮดเดอร์ไอพี แต่ถ้าร้ายแรงกว่านั้นก็คือเกิดข้อผิดพลาดในการทำงานของไอพี มันแสดงถึงว่ามีค่าในเฮดเดอร์ที่ไม่สามารถเข้าใจได้ แมสเสจนี้จะไม่ถูกส่งถ้าดาต้าแกรมไม่ถูกละทิ้ง ฟิลด์พอยท์เตอร์ (Pointer) บ่งถึงตำแหน่งของอ็อคเต็ดที่สงสัย เช่น ถ้ามีค่าเป็น 1 ก็หมายถึงฟิลด์ TOS ถ้ามีค่าเป็น 20 ก็จะหมายถึงอ็อคเต็ดแรกของฟิลด์ออปชั่น เป็นต้น รูปแบบของเมสเสจชนิดนี้เป็นดังรูปที่ 3-16

Type	Code	Checksum
Pointer	Unused (must be 0)	
Internet header + 64 bits of datagram prefix		
.....		

รูปที่ 3-16 *รูปแบบแมสเสจไอซีเอ็มพี parameter problem*

9. ไอซีเอ็มพีชนิด 13 และ 14 - time stamp request and reply

ใช้เก็บเวลาจากนาฬิกา (clock) ของเครื่องระยะไกล ผู้ร้องขอจะส่งแมสเสจไอซีเอ็มพีชนิด 13 และ 14 ปลายทางจะตอบด้วยแมสเสจชนิด 14 ฟิลด์ Original time stamp จะถูกเติมก่อนดาต้าแกรมถูกส่ง ฟิลด์ Receive time stamp จะเติมทันทีที่ได้รับการร้องขอ และฟิลด์ Transmit time stamp จะถูกเติมทันทีก่อนที่จะตอบกลับไปยังต้นทาง รูปแบบของแมสเสจไอซีเอ็มพีชนิดนี้เป็นดังรูปที่ 3-17 โดยแมสเสจชนิด

นี้จะถูกใช้สำหรับเก็บสถิติเกี่ยวกับสมรรถนะของการเชื่อมต่อไปยังโฮส หรือเพื่อการเข้าจังหวะกันของนาฬิกาในโฮส

Type	Code	Checksum
Identifier		Sequence
Originate time stamp		
Receive time stamp		
Transmit time stamp		

รูปที่ 3-17 รูปแบบแมสเสจไอซีเอ็มพี time stamp request/reply

10. ไอซีเอ็มพีชนิด 15 และ 16 – information request

โฮสใช้เพื่อหาหมายเลขของเครือข่าย (network number) ถ้าโฮสนั้นไม่รู้แอดเดรสที่ใช้ในเฮดเดอร์ไอพีจะมีค่าเป็น 0 หมายถึงเครือข่ายนี้ ซึ่งจะถูกเติมอย่างถูกต้องโดยปรายทางและถูกส่งกลับมา รูปแบบของแมสเสจไอซีเอ็มพีชนิดนี้เป็นดังรูปที่ 3-18 กลไกนี้ใช้กับระบบไดอัลอิน (dial-in) ที่ใช้สลิป (SLIP) เป็นวิธีในการกำหนดเน็ตเวิร์คแอดเดรสที่เหมาะสม ให้กับแต่ละปลายของการเชื่อมต่อ

Type	Code	Checksum
Identifier		Sequence

รูปที่ 3-18 รูปแบบแมสเสจไอซีเอ็มพี information request

11. ไอซีเอ็มพีชนิด 17 และ 18 – address mask request

ใช้ร่วมกับการกำหนดแอดเดรสเป็นสับเน็ต (Subnet addressing) ได้เพื่อให้โหนดรู้ถึงสับเน็ตมาสค์ (Subnet mask) ของเครือข่ายที่มันต่ออยู่ด้วย โหนดสามารถส่งคำร้องขอไปยังแอดเดรสที่มันรู้จักซึ่งอาจเป็นเราท์เตอร์ หรือทำการบรอดคาสท์ไปยังเครือข่ายก็ได้ คำตอบกลับ (reply) จะส่งตรงถ้าโหนดรู้แอดเดรสของมัน หรือทำการบรอดคาสท์ก็ได้ สับเน็ตมาสค์จะถูกใส่มาในฟิลด์แอดเดรสมาสค์ (Address mask) ของการตอบกลับ ดังรูปที่ 3-19

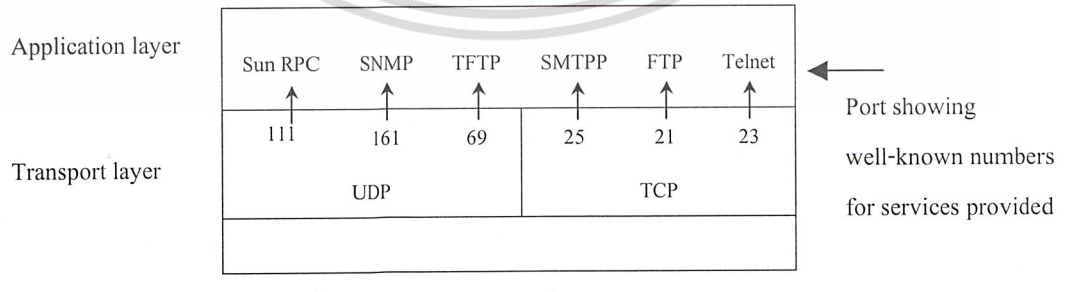
Type	Code	Checksum
Identifier		Sequence
Address mask		

รูปที่ 3-19 รูปแบบแมสเสจไอซีเอ็มพี address mask request

3.2.2 เลเยอร์ทรานสปอร์ตของทีซีพี/ไอพี

บริการที่ไอพีมีให้ ยังต้องการให้ดาต้าแกรมส่งไปยังบริการในเลเยอร์แอพพลิเคชั่นที่เหมาะสมโดยตรง และต้องการบริการที่เชื่อถือได้ สำหรับแอพพลิเคชั่นที่จำเป็นต้องใช้มัน ซึ่งหน้าที่เหล่านี้เป็นของเลเยอร์นี้ ทำโดยโพรโตคอลทรานสปอร์ต 2 ตัว คือ โพรโตคอลยูดีพี (User Datagram Protocol : UDP) และโพรโตคอลทีซีพี (Transmission Control Protocol : TCP) ซึ่งการเลือกใช้นั้น ขึ้นอยู่กับประเภทของบริการที่แอพพลิเคชั่นของผู้ใช้ต้องการ

เมื่อข้อมูลถูกส่งไปยังเครื่องที่ต้องการโดยไอพีมันจะถูกส่งไปยังบริการแอพพลิเคชั่นที่เกี่ยวข้องบนเครื่องนั้น ๆ การมัลติเพล็กซ์ข้อมูลไปยัง หรือจากเลเยอร์ไอพีและส่งข้อมูลไปยังแอพพลิเคชั่นที่ถูกต้องเป็นหน้าที่อย่างหนึ่งของเลเยอร์ทรานสปอร์ตนี้ รวมทั้งการทำให้ไม่มีข้อผิดพลาด (error-free) และบริการส่งข้อมูลไปยังแอพพลิเคชั่นที่ถูกต้องแบบต้องก่อตั้งหรือไม่ก่อตั้งการเชื่อมต่อก่อน ก็เป็นหน้าที่ของเลเยอร์นี้เช่นกัน โพรโตคอลยูดีพีจะให้บริการแบบไม่ต้องก่อตั้งการเชื่อมต่อก่อน (Connectionless) ซึ่งเป็นบริการที่ไม่มีความน่าเชื่อถือ เพราะว่ามันจะอนุญาตให้ส่งข้อมูลไปยังเครื่องหรือกลุ่มของเครื่องโดยไม่จำเป็นต้องก่อตั้งการเชื่อมต่อก่น ดังนั้นดาต้าแกรมหนึ่งจะถูกส่งไปยังโหนดอื่นได้โดยไม่ต้องการการตอบสนองว่าดาต้าแกรมดังกล่าวได้ไปถึงแล้วหรือยัง ในสิ่งแวดล้อมบางอย่าง บริการแบบนี้ก็เป็นวิธีที่มีประสิทธิภาพในการทำงานมาก บริการแอพพลิเคชันที่ใช้โพรโตคอลนี้ได้แก่ ทีเอฟทีพี (TFTP) เอ็นเอฟเอส (NFS) และการบรอดคาสท์ เป็นต้น โพรโตคอลทีซีพีจะให้บริการแบบจำเป็นต้องก่อตั้งการเชื่อมต่อก่อน (connection-oriented) การเชื่อมต่อมีลักษณะคล้ายท่อของข้อมูลที่อยู่ระหว่างจุด 2 จุด ไม่มีการทำบรอดคาสท์หรือมัลติคาสท์ในโพรโตคอลนี้ โพรโตคอลทีซีพีมีฟีเจอร์ (feature) ในการให้บริการที่น่าเชื่อถือระหว่างคอมพิวเตอร์ 2 เครื่อง เพื่อให้มีความน่าเชื่อถือโพรโตคอลทีซีพียังได้เพิ่มโอเวอร์เฮดจำนวนมาก เพื่อใช้ในการทำแอคโนวเลดเมนท์ (Acknowledgement) , โฟล์วคอนโทรล (Flow control) , ไทม์เมอร์ (Timers) และความสะดวกสบายในการจัดการการเชื่อมต่อ (Connection management faciliteis) ทีซีพีโอเวอร์เฮดมากกว่ายูดีพีในแง่ของการประมวลผลที่ต้องการและขนาดของเฮดเดอร์ที่ต้องใช้ ตัวอย่างของแอพพลิเคชั่นที่ต้องการบริการแบบนี้ ได้แก่ เทลเน็ต (Telnet) และเอฟทีพี (FTP) เป็นต้น



***รูปที่ 3-20** หมายเลขพอร์ตที่ใช้ในยูดีพีและทีซีพี*

ทั้งทีซีพีและยูดีพีต่างก็ใช้การอ้างแอดเดรสเป็นพอร์ต (Port addressing) ในการส่งข้อมูลไปยังบริการในชั้นแอพพลิเคชั่นที่สัมพันธ์กัน ซึ่งพอร์ตก็คือแอดเดรสขนาด 16 บิต ซึ่งหมายเลขของพอร์ตที่เป็นที่รู้จักกันดี (Well-know port) ถูกกำหนดเป็น 0-255 ดังรูปที่ 3-20 นอกจากนี้ยังมีการใช้ซ็อคเก็ต (Socket) ในทีซีพี/ไอพีอีกด้วย ซึ่งซ็อคเก็ตก็คือการนำไอพีแดเดรสมาต่อกับหมายเลขพอร์ต เมื่อไอพีแอดเดรสนั้นไม่มีใครซ้ำในแต่ละโหนด และหมายเลขพอร์ตก็ไม่ซ้ำบนโหนดนั้น ดังนั้นซ็อคเก็ตก็จะเป็นการระบุถึงบริการในชั้นแอพพลิเคชั่นที่ไม่มีการใช้ซ้ำกัน เพราะซ็อคเก็ตไม่ซ้ำดังนั้นทั้งโพรโตคอลทีซีพและยูดีพีจึงได้รวมไอพีแอดเดรสกับหมายเลขพอร์ตเข้าไปคำนวณเช็คซัมด้วย เพื่อให้แน่ใจได้ว่าดาต้าแกรมที่ส่งไปถึงโฮสผิดจะไม่เป็นที่ยอมรับโดยเลเยอร์ทรานสปอร์ตของโฮสนั้น แม้หมายเลขพอร์ตนั้นจะเป็นที่รู้จักกันดีก็ตาม บริการในชั้นแอพพลิเคชั่นส่วนใหญ่จะอนุญาตให้มีหลายเซสชั่น (session) ได้ ซึ่งทำให้จำเป็นต้องแยกเซสชั่นเหล่านี้ให้ได้เพื่อให้แน่ใจได้ว่าข้อมูลจะถูกส่งกลับคอมพิวเตอร์ที่เหมาะสม ตัวอย่างเช่น ผู้ใช้ทุกคนที่ใช้เทลเน็ตจะติดต่ออยู่กับโฮสเดียวกันด้วยหมายเลขพอร์ตเดียวกัน คือ 23 ทางหนึ่งที่จะแยกแยะได้คือดูว่าดาต้าแกรมมาจากไหน แต่ก็เป็นไปได้ที่ผู้ใช้ 2 คนจะติดต่อมาจากโฮสเดียวกัน การแก้ปัญหานี้ทำได้โดยใช้พอร์ตที่ยังไม่มีใครใช้ในเครื่องนั้นมาใช้ ด้วยวิธีนี้ถึงแม้ว่า 2 เซสชั่นจะใช้เซิร์ฟเวอร์เดียวกันและยังมาจากโฮสเดียวกันก็ง่ายในการแยกแยะ 2 เซสชั่นนี้

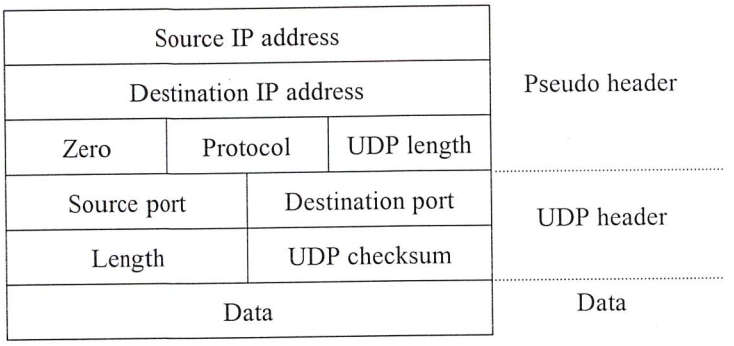
โพรโตคอลยูดีพี (User Datagram Protocol : UDP)

ยูดีพีเพิ่มความสามารถเข้าไปในการให้บริการของไอพีเพียงเล็กน้อย และใช้ฟิลด์ Source Port และ Destination Port ในเฮดเดอร์ของมันเพื่อส่งข้อมูลไปยังบริการในเลเยอร์แอพพลิเคชั่น ฟิลด์ต่าง ๆ ในโพรโตคอลยูดีพีเป็นดังรูปที่ 3-21 ประกอบด้วย

Source port	Destination port
Length	UDP checksum
Data	

รูปที่ 3-21 ฟิลด์ในเฮดเดอร์ของยูพีดี

1. Source Port — บอกถึงว่าดาต้าแกรมมาจากพอร์ตหมายเลขอะไร ซึ่งก็คือบริการใดในชั้นแอพพลิเคชั่น มีขนาด 16 บิต
2. Destination Port — บอกว่าดาต้าแกรมนั้นต้องการส่งไปยังพอร์ตหมายเลขอะไร บริการใดในชั้นแอพพลิเคชั่น
3. Length — ความยาวของด้านดาต้าแกรมยูพีดี มีขนาด 16 บิต
4. Chechsum — เป็นการป้องกันข้อมูลที่ยูพีดีบรรทุกมา ซึ่งคิดทั้งฟิลด์ยูพีดีและซูโดเฮดเดอร์ (pseudo header) ดังรูปที่ 3-22 มีขนาด 16 บิต



รูปที่ 3-22 ค่าเช็คซัมประกอบด้วยเฮดเดอร์และซูโดเฮดเดอร์

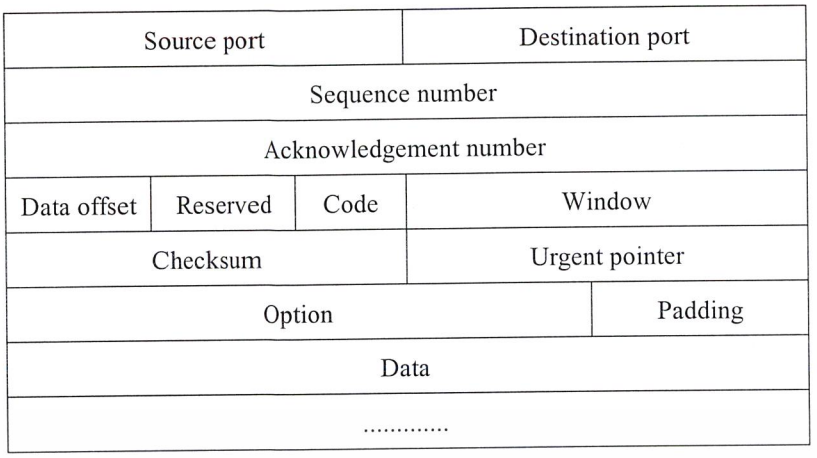
โพรโตคอลทีซีพี (TCP: Transmission Control Protocol)

ช่วยเพิ่มความน่าเชื่อถือให้ไอพีและใช้พอร์ตในการกำหนดแอดเดรสของเลเยอร์แอพพลิเคชั่น เช่นเดียวกันกับยูพีดี โพรโตคอลทีซีพีเป็นโพรโตคอลที่ต้องการการเชื่อมต่อก่อน(connection-oriented) ดังกล่าวมาแล้ว คือจะต้องเปิดการติดต่อก่อนส่งและเมื่อส่งเสร็จก็จะต้องปิดการติดต่อนั้นด้วย ข้อมูลที่ทีซีพีส่งให้ไอพีนั้นจะประกอบด้วยเฮดเดอร์ของทีซีพีรวมกับข้อมูลจากเลเยอร์แอพพลิเคชั่น ซึ่งรวมกันแล้วเรียกว่าเซกเมนท์ (segment) เพื่อความน่าเชื่อถือที่มากขึ้นเราต้องการสิ่งดังต่อไปนี้

- การตรวจและแก้ไขข้อผิดพลาด: ซึ่งเกี่ยวข้องกับความเป็นไปได้ที่เซกเมนท์จะเสียหายจากสายสื่อสารหรือซอฟต์แวร์ในเลเยอร์ที่สูงกว่า
- โฟล์วคอลโทรล: ใช้ในการป้องกันไม่ให้ผู้ส่งทำให้ผู้รับประสบปัญหาเนื่องมาจากข้อจำกัดในทรัพยากร
- การจัดลำดับการส่ง: จำเป็นต้องมีเพราะเลเยอร์ไอพีสามารถส่งดาต้าแกรมซึ่งมีเซกเมนท์ทีซีพีในลำดับใดก็ได้ ซึ่งเกิดขึ้นเมื่อดาต้าแกรมถูกส่งคนละเส้นทาง
- การกำจัดเซกเมนท์ที่ซ้ำ: เกิดเพราะกลไกกู้คืน(error-recovery) ที่ทีซีพีใช้

ทีซีพีได้ทำการเพิ่มความสามารถที่กล่าวข้างต้นได้โดย

- ใช้หมายเลขแสดงลำดับในการแยกแยะข้อมูล
- ต้องได้รับเอคโนวเลดเมนท์ของการส่งข้อมูลในลำดับที่ถูกต้อง
- มีการส่งเซกเมนท์ซ้ำเมื่อไม่ได้รับการตอบกลับในเวลาที่กำหนด



รูปที่ 3-23 เฮดเดอร์ทีซีพี

และเพื่อให้เกิดหน้าที่ที่ได้กล่าวข้างต้น เฮดเดอร์ทีซีพีจึงซับซ้อน มีฟิลด์มากกว่าเฮดเดอร์ยูดีพี ดังรูปที่ 3-23 ทีซีพีมีฟิลด์ Source port และ Destination port ด้วยเหตุผลเดียวกันกับยูดีพี คือ ใช้แยกแยะแอพพลิเคชั่น ฟิลด์ที่เหลือส่วนมากมีเพื่อความน่าเชื่อถือ และเกี่ยวข้องกับการควบคุมการติดต่อ ดังนี้

1. **Sequence number** มีขนาด 32 บิต ในทีซีพีจะนับออคเต็ดในการส่ง แต่โพรโตคอลอื่นที่ใช้เลขลำดับเพื่อควบคุมความผิดพลาดจะนับเซกเมนท์ เลขลำดับในเฮดเดอร์นี้จะกำหนดตำแหน่งในข้อมูลทั้งหมดของออคเต็ดแรกในเซกเมนท์ ซึ่งช่วยให้ทีซีพีใส่เซกเมนท์ในตำแหน่งที่ถูกต้องในข้อมูลได้ แม้ไอพีจะส่งข้อมูลไม่เป็นลำดับก็ตาม การที่มันมี 32 บิต ทำให้ไม่เกิดการซ้ำของค่าแม้เวลาในการส่งจะเร็วมากก็ตาม คือ เซกเมนท์ที่ได้รับอาจมีเลขซ้ำกับเซกเมนท์ซึ่งแอคโนเลจไปเรียบร้อยแล้วได้ แต่ถ้าเกิดข้อผิดพลาดเนื่องจากการซ้ำก็ไม่เป็นปัญหา เพราะแค่ไม่สนใจ โดยไม่ต้องทำอะไร

2. **Acknowledgement number** มีขนาด 32 บิต บอกให้รู้ว่าได้รับออคเต็ดทั้งหมดอย่างถูกต้องจนถึงเลขแอคโนวเลจลบด้วย 1 เมื่อผู้ส่งได้รับค่านี้ก็ไม่จำเป็นต้องเก็บข้อมูลไว้เพื่อส่งใหม่อีกต่อไป ซึ่งเลขแอคโนวเลจนี้จะใช้ได้เมื่อเซตแฟลก ACK

3. **Data Offset** วัดออฟเซตที่เป็นจุดเริ่มต้นของฟิลด์ข้อมูลใหม่ในหน่วย 32 บิตเวิร์ด ค่าปกติคือ 5 ซึ่งก็คือเฮดเดอร์ 20 ออคเต็ด เมื่อไม่ใช้ฟิลด์ออปชั่น ฟิลด์นี้มีขนาด 4 บิต

4. **Flags** มีขนาด 6 บิต ต่อจากฟิลด์ Reserve ซึ่งมีขนาด 6 บิตเช่นกัน ใช้บอกว่าฟิลด์อื่นใช้ได้หรือไม่ และสำหรับการควบคุมการติดต่อ ประกอบด้วย 6 แฟลกดังนี้

- URG บอกว่าใช้ฟิลด์ Urgent pointer ได้ ซึ่งฟิลด์นี้ชี้ไปยังออคเต็ดในฟิลด์ข้อมูลซึ่งเป็นปลายทางของข้อมูล urgent ซึ่งไม่ถูกมองเป็นข้อมูลปกติ และควรจะถูกประมวลผลก่อนข้อมูลอื่น ๆ
- ACK บอกว่าฟิลด์ Acknowledge ใช้ได้ ซึ่งฟิลด์นี้จะใช้ไม่ได้เมื่อก่อตั้งการเชื่อมต่อ คือก่อนที่แต่ละโหนดจะสามารถตัดสินใจได้ว่าจะใช้ค่า sequence และ acknowledge ใด
- PSH คือ แฟลก push ซึ่งทำให้เลเยอร์ทีซีพีที่ระยะไกลส่งเซกเมนท์นี้ไปให้เลเยอร์แอพพลิเคชั่นอย่างทันทีทันใด ปกติทีซีพีจะหันมาสนใจข้อมูลจากเซกเมนท์ที่เข้ามา และส่งข้อมูลนี้ไปยังเลเยอร์แอพพลิเคชั่นในบัฟเฟอร์ที่ใหญ่กว่าเพื่อลดโอเวอร์เฮดในการประมวลผล
- RST คือ แฟลก reset ใช้เมื่อเกิดข้อผิดพลาดอื่น ๆ บอกถึงว่ามีข้อผิดพลาดเกิดขึ้นและควรจะหยุดการติดต่อ
- SYN คือ แฟลก synchronize ใช้ขณะเริ่มต้นการก่อตั้งการเชื่อมต่อระหว่าง 2 โหนด ซึ่ง ณ เวลานั้นทั้ง 2 โหนดไม่รู้ว่าควรจะใช้เลขแอคโนวเลจใด การก่อตั้งการเชื่อมต่อจะประกอบด้วยการแลกเปลี่ยนเซกเมนท์แบบ 2 ทาง(2-way exchange of segment) พร้อมทั้งเซตแฟลก SYN ซึ่งแต่ละอันจะถูกแอคโนวเลจในเซกเมนท์โดยเซตแฟลก ACK
- FIN ใช้ในการเลิกการติดต่อเมื่อข้างใดข้างหนึ่งไม่มีข้อมูลที่จะส่ง ก็จะส่งเซกเมนท์ที่มีการเซตแฟลก FIN เมื่อทั้ง 2 ข้างส่งแฟลก FIN การติดต่อก็จะปิดลง

5. **Window** มีขนาด 16 บิต บอกถึงขนาดเนื้อที่ในบัฟเฟอร์ของโหนดนี้ที่ใช้ได้ในการเชื่อมต่อนี้ โหนดอื่นจะต้องไม่ส่งข้อมูลที่ยังไม่แอคโนวเลจมาเกินเนื้อที่ในบัฟเฟอร์ที่ระบุนี้

6. **Checksum** มีขนาด 16 บิต ใช้ตรวจสอบเฮดเดอร์และข้อมูล

7. **Urgent pointer** มีขนาด16 บิต ค่าในฟิลด์นี้ชี้ไปยังปลายของข้อมูลในฟิลด์ข้อมูลที่เร่งด่วนและต้องการความสนใจทันทีจะใช้ได้เมื่อมีการเซตแฟลก URG

8. **Options** ขนาดไม่คงที่ มีออปชั่นเดียวที่ใช้เป็นปกติในทีซีพีคือขนาดเซกเมนท์มากที่สุด (Maximum Segment Size : MSS) เพื่อบอกเลเยอร์ทีซีพีปลายทางถึงขนาดเซกเมนท์ที่มากที่สุดที่ควรจะส่ง ซึ่งรวมเฮดเดอร์ทีซีพีแล้ว

9. **Padding** ถ้าใช้ฟิลด์ออปชั่นแพดดิ้งจะทำให้มั่นใจได้ว่าข้อมูลเริ่มที่ขอบ 32 บิตอย่างที่ออฟเซ็ตข้อมูลชี้อย่างถูกต้อง

3.2.3 เลเยอร์แอพพลิเคชั่นของทีซีพี/ไอพี

บริการของเลเยอร์แอพพลิเคชั่นจะรับผิดชอบในการเชื่อมต่อ (interface) ระหว่างแอพพลิเคชั่น ของผู้ใช้ และบริการในชั้นทรานสปอร์ต บริการของแอพพลิเคชั่นไม่ใช่แอพพลิเคชั่นของผู้ใช้ แต่เป็นการเชื่อมต่อกับแอพพลิเคชั่นนั้นกับเครือข่ายสื่อสาร มีบริการของแอพพลิเคชั่นหลายตัวที่เหมาะสมกับแอพพลิเคชั่นหลายชนิดและยังมียูทิลิตี้ในการจัดการอีกจำนวนหนึ่ง ตัวอย่างของบริการในชั้นนี้คือ

1.**FTP** ย่อมาจาก File Transfer Protocol ใช้พอร์ตหมายเลข 20 เป็นโพรโตคอลมาตรฐาน และเป็นวิธีที่ง่ายที่สุดในการแลกเปลี่ยนไฟล์กันในอินเทอร์เน็ต FTP เป็นโพรโตคอลแอพพลิเคชั่นที่ใช้โพรโตคอลชุดทีซีพี/ไอพี และมักจะถูกเรียกใช้เสมอในการส่งไฟล์เว็บเพจจากผู้สร้างเวบเพจไปยังคอมพิวเตอร์ที่ทำตัวเป็นเซิร์ฟเวอร์ เพื่อให้ใครก็ตามในอินเทอร์เน็ตสามารถใช้ได้ นอกจากนี้ FTP ยังใช้ในการดาวน์โหลดโปรแกรมและไฟล์อื่นๆ จากเซิร์ฟเวอร์มายังคอมพิวเตอร์ของเราได้ด้วย ในฐานะผู้ใช้ เราสามารถใช้ FTP ด้วยอินเทอร์เฟสแบบคอมมานไลน์ง่าย ๆ เช่น จากหน้าต่างดอสพรอมท์ หรือด้วยโปรแกรมที่มีขาย ซึ่งจะเสนออินเทอร์เฟสแบบกราฟฟิกให้ นอกจากนี้เว็บบราวเซอร์ของเราก็ยังสามารถใช้สำหรับดาวน์โหลดโปรแกรมที่เราเลือกจากเว็บเพจโดยส่ง FTP request ได้ด้วย ในการใช้ FTP เรายังสามารถอัพเดท หมายถึงการลบ การเปลี่ยนชื่อ การย้ายตำแหน่ง และการคัดลอกไฟล์ที่เซิร์ฟเวอร์ได้อีกด้วย แต่เราจำเป็นต้องล็อคออนเข้าไปในเซิร์ฟเวอร์ก่อน อย่างไรก็ตามไฟล์ที่ให้ใครก็ได้ใช้ สามารถเข้าถึงได้โดย anonymous FTP

2. **Telnet** ใช้พอร์ตหมายเลข 23 Telnet คือทางที่จะช่วยให้เราสามารถเข้าถึงคอมพิวเตอร์ของใครก็ตามถ้าเขาอนุญาต ซึ่งมักจะเรียกคอมพิวเตอร์ลักษณะนี้ว่าโฮสคอมพิวเตอร์ หรือถ้าจะกล่าวให้ลึกกว่านั้นก็สามารถจะพูดได้ว่า Telnet คือคำสั่งของผู้ใช้บนโพรโตคอลทีซีพี/ไอพีสำหรับการเข้าถึงคอมพิวเตอร์ระยะไกล โพรโตคอลเว็บหรือ HTTP และ FTP นั้นอนุญาตให้เราร้องขอไฟล์ที่เจาะจงจากคอมพิวเตอร์ระยะไกล แต่ไม่ได้ให้เราล็อกออนจริง ๆ ในฐานะผู้ใช้ของคอมพิวเตอร์เครื่องนั้น แต่ด้วย Telnet เราสามารถล็อกออนเหมือนเป็นผู้ใช้ปกติ ด้วยสิทธิอะไรก็ตามที่เราได้รับอนุญาตให้ทำบนเครื่องคอมพิวเตอร์เครื่องนั้น

3. **SMTP** ย่อมาจาก Simple Mail Transfer Protocol ใช้พอร์ตหมายเลขที่ 25 SMTP คือโพรโตคอลทีซีพี/ไอพีที่ถูกใช้ในการส่งหรือรับอีเมล์ (E-mail) อย่างไรก็ตามเพราะข้อจำกัดในความสามารถของมันในการจัดคิวข่าวสารที่ฝั่งผู้รับ เราจึงมักจะใช้โพรโตคอลอื่นแทน เช่น POP3 หรือ IMAP ซึ่งจะให้ผู้ใช้เก็บข่าวสารในกล่องจดหมาย (mail box) ของเซิร์ฟเวอร์และดาวน์โหลดข่าวสารเหล่านั้นจากเซิร์ฟเวอร์เป็นระยะ ๆ หรือกล่าวได้อีกอย่างว่าปกติแล้วผู้ใช้โปรแกรมที่ใช้ SMTP สำหรับการส่งอีเมล์ และใช้ POP3 หรือ IMAP สำหรับการรับอีเมล์ โปรแกรมเกี่ยวกับการส่งเมล์ส่วนใหญ่ เช่น Eudora จะให้เราระบุทั้งเซิร์ฟเวอร์ SMTP และ เซิร์ฟเวอร์ POP

4. **Gopher** ใช้พอร์ตหมายเลข 70 Gopher เป็นโพรโตคอลแอพพลิเคชั่นในเซิร์ฟเวอร์ซึ่งโครงสร้างไฟล์ถูกจัดการเรียงเป็นลำดับชั้น Gopherได้จัดหาทางที่จะนำเท็กซ์ไฟล์จากทั่วโลกมายัง viewer บนคอมพิวเตอร์ของเรา Gopherได้รับความนิยมเป็นเวลาหลายปีโดยเฉพาะอย่างยิ่งในมหาวิทยาลัย และยังเป็นก้าวหนึ่งที่นำไปสู่ HTTP แต่ด้วยไฮเปอร์เท็กซ์ลิงค์ ภาษา HTML และการปรากฏตัวของบราวเซอร์แบบกราฟฟิคทำให้ Gopher เสื่อมความนิยมลงอย่างรวดเร็ว โครงสร้างไฟล์แบบดั้งเดิมจำนวนหนึ่ง โดยเฉพาะในมหาวิทยาลัย ยังคงใช้อยู่และสามารถเข้าถึงเว็บบราวเซอร์ส่วนใหญ่เพราะ มันยังคงสนับสนุนโพรโตคอลGopher Gopher ถูกพัฒนาที่มหาวิทยาลัยมินเนโซต้า(The University of Minnesota) ถึงแม้ว่าบราวเซอร์Gopher และไฟล์จะเป็นเท็กซ์แต่บราวเซอร์Gopher ก็ได้ถูกพัฒนาให้แสดงรูปกราฟฟิคได้ คือ ไฟล์ GIF และ JPEG ซึ่งถูกรวมไว้ในไฟล์ไดเรคทอรี่ Gopher

5. **HTTP** ย่อมาจาก Hypertext Transfer Protocol (HTTP) ใช้พอร์ตหมายเลข 80 HTTP เป็นชุดของกฎสำหรับการแลกเปลี่ยนไฟล์ ซึ่งมีทั้งเท็กซ์ กราฟฟิก ภาพ เสียง วีดีโอ และไฟล์มัลติมีเดียอื่นๆ บนเวิลด์ไวด์เว็บ (World Wide Web) เมื่อเปรียบเทียบกับชุดโพรโตคอลทีซีพี/ไอพีซึ่งเป็นพื้นฐานสำหรับการแลกเปลี่ยนข้อมูลข่าวสารบนอินเทอร์เน็ต HTTP ก็คือแอพพลิเคชั่น โพรโตคอล คอนเซ็ปท์ (concept) สำคัญที่เป็นส่วนหนึ่งของ HTTP ประกอบไปด้วยความคิดที่ว่าไฟล์สามารถอ้างอิงไปยังไฟล์อื่นได้ โดยเว็บเซิร์ฟเวอร์ใดๆ ก็ตาม นอกจากจะเก็บไฟล์ HTML และไฟล์อื่น ๆ แล้ว ยังมี HTTP daemon ซึ่งเป็นโปรแกรมที่ถูกออกแบบมาให้รอ HTTP requests และจัดการเมื่อคำร้องขอมาถึง เว็บบราวเซอร์ ของเราก็คือไคลเอ็นท์ HTTP ซึ่งจะส่งคำร้องขอไปยังเซิร์ฟเวอร์ เมื่อผู้ใช้บราวเซอร์ใส่การร้องขอไฟล์ โดยการเปิดเว็บไฟล์ (โดยการพิมพ์ URL : Uniform Resource Locator) หรือคลิ๊ก (Click) บน Hypertext link บราวเซอร์ก็จะสร้าง HTTP request และส่งไปยังไอพีแอดเดรสที่ระบุโดย URL หลังจากนั้น HTTP daemon ที่เซิร์ฟเวอร์ปลายทางจะได้รับคำร้องขอและเมื่อทำการประมวลผลสิ่งที่จำเป็นแล้วไฟล์ที่ถูกร้องขอก็จะถูกส่งกลับ

6. **POP3** ย่อมาจาก Post Office Protocol 3 ใช้พอร์ตหมายเลข 110 POP3 เป็นเวอร์ชั่นล่าสุดของโพรโตคอลมาตรฐานสำหรับการรับอีเมล์ POP3 เป็นโพรโตคอลแบบไคลเอ็นท์/เซิร์ฟเวอร์ ซึ่งจะรับอีเมล์ที่ถูกส่งมาและเก็บเอามาไว้ให้ในเซิร์ฟเวอร์ของเรา เราสามารถดูเมล์ได้ที่เซิร์ฟเวอร์และดาวน์โหลดได้ นอกจาก POP3 แล้วยังมีโพรโตคอลที่ทำงานคล้ายคลึงกัน คือ โพรโตคอล IMAP (Interactive Mail Access Protocol) ด้วย IMAP เราสามารถดูอีเมล์ที่เซิร์ฟเวอร์เหมือนกับว่าอีเมล์เหล่านั้นอยู่บนเครื่องคอมพิวเตอร์ของเราเอง อีเมล์ที่ถูกลบที่เครื่องเรา จะยังอยู่บนเซิร์ฟเวอร์เช่นเดิม นอกจากนี้อีเมล์ยังสามารถเก็บและค้นหาได้ที่เซิร์ฟเวอร์ เราสามารถคิดได้ว่า POP คือบริการแบบเก็บและส่งต่อไป (Store-and-forward) ส่วน IMAP ก็คือไฟล์เซิร์ฟเวอร์ระยะไกล (remote file server)

POP และ IMAP เกี่ยวข้องกับการรับอีเมล์ และไม่ยุ่งเกี่ยวกับ SMTP ซึ่งเป็นโพรโตคอลสำหรับส่งอีเมล์ข้ามอินเทอร์เน็ต

7. **NNTP** ย่อมาจาก Network News Transfer Protocol ใช้พอร์ตหมายเลข 119 NNTP คือโพรโตคอลที่ใช้โดยคอมพิวเตอร์ทั้งเซิร์ฟเวอร์ และไคลเอ็นท์สำหรับจัดการข้อความ (notes) ที่ทิ้งไว้บนกลุ่มข่าวUsenet (Usenet newsgroup) NNTP ได้มาแทนที่โพรโตคอล Usenet ดั้งเดิมคือ UUCP (UNIX-to-UNIX Copy Protocol) เซิร์ฟเวอร์ NNTP จะจัดการเครือข่ายของกลุ่มข่าว Usenet ที่ถูกรวบรวมและรวมเซิร์ฟเวอร์เข้าไว้ที่ผู้ให้บริการอินเทอร์เน็ตของเรา ไคลเอ็นท์ NNTP อาจจะถูกรวบรวมเป็นส่วนหนึ่งของ Netscape , Internet Explorer , Opera หรือเว็บบราวเซอร์อื่น ๆ หรือเราอาจจะใช้โปรแกรมแยกต่างหากที่เรียกว่า Newsreader ก็ได้

8. **SNMP** ย่อมาจาก Simple Network Management Protocol ใช้พอร์ตหมายเลข 161 SNMP คือโพรโตคอลที่ใช้บริการการจัดการเครือข่ายและการมอนิเตอร์อุปกรณ์ในเครือข่ายและฟังก์ชั่นของอุปกรณ์เหล่านั้น ซึ่งไม่ได้จำกัดอยู่เฉพาะเครือข่ายที่ใช้ทีซีพี/ไอพี

9. **IRC** ย่อมาจาก Internet Relay Chat ใช้พอร์ตหมายเลข 194 IRC คือระบบสำหรับ chat ที่เกี่ยวข้องกับชุดของกฎข้อตกลงและซอฟต์แวร์ประเภทไคลเอ็นท์/เซิร์ฟเวอร์ในเว็บไซต์เฉพาะ เช่น เมืองแห่งการคุย (Talk City) หรือเครือข่าย IRC และช่วยให้เราดาวน์โหลดไคลเอ็นท์ IRC มายังเครื่องคอมพิวเตอร์ของเรา เราสามารถเริ่มการคุยในกลุ่ม (เรียกว่าแชนเนล) ใดก็ได้ที่มีอยู่ ซึ่งโพรโตคอลสำหรับค้นหากลุ่มการคุยที่มีอยู่และสมาชิกของกลุ่มนั้น ๆ ด้วย ผู้ที่เข้าไปร่วมคุยในกลุ่มการคุยใดก็ตามจะใช้ชื่อเล่นซึ่งใช้ได้เฉพาะครั้งนั้นๆ (เราไม่สามารถเป็นเจ้าของชื่อเล่นนั้นได้ คืออาจมีคนใช้ซ้ำกับเราได้)

3.3 เครือข่ายท้องถิ่นแบบอีเธอร์เนต (Ethernet LAN)

ก่อนที่จะอธิบายถึงรายละเอียด เราควรจะรู้ก่อนว่าเครือข่ายท้องถิ่นแบบนี้มีข้อดีข้อเสียอย่างไร

ข้อดี :-

1. ง่ายต่อการติดตั้ง เพราะทุกสเตชั่นสามารถต่อเข้ากับเซกเมนท์ (segment) ได้โดยใช้คอนเน็คเตอร์แบบที (T-connector) หรือทรานซีฟเวอร์ (Transceiver) ง่าย ๆ และเครือข่ายท้องถิ่นแบบอีเธอร์เนตยังไม่ต้องการฮับ (Hub) ในการเชื่อมต่ออีกด้วย
2. เป็นเทคโนโลยีที่รู้จักกันดี เนื่องจากในภาคธุรกิจได้ติดตั้งอีเธอร์เนตเป็นเวลาหลายปีแล้ว
3. ราคาการ์ดได้ลดลง ทำให้การ์ดมีราคาไม่แพง
4. มีการเชื่อมโยงสายได้หลายแบบ

ข้อเสีย :-

1. ทรูพุท (Throughput) จะลดลงเมื่อมีโหลดในแลนมาก เนื่องจากใช้วิธีในการเข้าถึงสายสื่อสารแบบซีเอสเอ็มเอ/ซีดี (CSMA/CD)
2. เนื่องจากการที่ระบบมีการใช้สารสื่อสารร่วมกัน ทำให้ยากในการค้นหาสาเหตุเมื่อเกิดข้อผิดพลาดขึ้น (Troubleshooting) เช่น ถ้าเคเบิ้ลเสียหาย เซกเมนท์ของแลนทั้งหมดจะใช้งานไม่ได้ (down) และยากมากในการที่จะแยกโหนดที่เป็นต้นเหตุของปัญหาได้ ถึงแม้ว่าเท็นเบสที (10BaseT) จะแก้ปัญหาเหล่านี้ได้บางปัญหาก็ตาม

วิธีที่แลนแบบนี้ใช้ในการเข้าถึงสายสื่อสาร (access method) คือแบบซีเอสเอ็มเอ/ซีดี (Carrier Sense Multiple Access with Collision Detection : CSMA/CD) ซึ่งวิธีเข้าถึงสายสื่อสารนี้จะเป็นตัวกำหนดชนิดของการ์ดที่ใช้ในการอินเทอร์เฟส (Interface card) ที่จะต้องติดตั้งไว้ในเวิร์คสเตชั่น และบอกถึงวิธีที่เวิร์คสเตชั่นจะเข้าถึงระบบเคเบิ้ล หรือสายสื่อสาร รวมทั้งวิธีที่ข้อมูลถูกเตรียมเพื่อส่งและถูกส่งออกไปอย่างไร ก่อนที่เราจะทำงานกับอีเธอร์เนตได เราควรจะรู้วิธีในการเข้าถึงสายสื่อสารที่แลนแบบนี้ใช้เป็นอย่างไร ซึ่งจะช่วยให้เข้าใจถึงสถิติ ข้อผิดพลาด และระดับการใช้งานของแลนได้ การทำงานของซีเอสเอ็มเอ/ซีดี อธิบายได้ดังต่อไปนี้

การส่ง (Transmitting)

เนื่องจากแลนแบบนี้ใช้ระบบเคเบิ้ลร่วมกัน ดังนั้นจึงต้องมีกฎในการหลีกเลี่ยงไม่ให้มีการส่งพร้อมกัน อย่างไรก็ตามถ้าสเตชั่นหลายสเตชั่นส่งข้อมูลพร้อมกันก็จะต้องมีวิธีที่จะรู้ว่าแพ็คเก็ตของมันถูกชนหรือไม่ และเมื่อไรจะเริ่มส่งใหม่ ซึ่งสเตชั่นจะทำตามวิธีดังนี้ เมื่อจะส่งข้อมูล

ขั้นตอนที่1 : ฟังก่อนส่ง (Listen before transmitting)

สเตชั่นจะคอยตรวจดูเคเบิ้ลว่ามีสัญญาณของสัญญาณพาหะหรือไม่ โดยสัญญาณในเคเบิ้ลจะตรวจพบได้โดยการวัดโวลเตจ (Voltage) ที่บ่งถึงการใช้งานเคเบิ้ล ถ้าสเตชั่นไม่พบว่ามีการใช้งานอยู่มันก็จะสรุปว่าเคเบิ้ลว่าง และจะเริ่มทำการส่ง แต่ถ้าไม่ว่าง เมื่อสเตชั่นทำการส่งข้อมูล ข้อมูลก็จะถูกชนกับสัญญาณที่มีในสายในขณะนั้น

ขั้นตอนที่ 2 : รอถ้าพบว่าเคเบิ้ลไม่ว่าง (Defer if the cable is busy)

เพื่อหลีกเลี่ยงการชน สเตชั่นจะต้องรอ ถ้าพบว่าเคเบิ้ลถูกใช้งานอยู่ ซึ่งจะรอเป็นระยะเวลาหนึ่งจนกระทั่งสายส่งว่าง ก่อนที่จะพยายามส่งใหม่อีกครั้ง

ขั้นตอนที่ 3 : ส่งและรอฟังการชน (Transmit and listen for collistions)

เมื่อสายส่งว่างเป็นเวลาอย่างน้อย 9.6 มิลลิวินาที สเตชั่นอาจจะส่งข้อมูล ถ้าสเตชั่นอื่น ๆ ในเซกเมนท์ก็ทำการส่งแพ็คเก็ตในเวลาเดียวกัน ก็จะเกิดการชนกันในสายสื่อสาร แพ็คเก็ตที่เกี่ยวข้องในการชนคือแพ็คเก็ตส่วนที่อยู่บนสายสื่อสารเท่านั้น ดังนั้นช่วงที่ทำการส่ง สเตชั่นจะคอยดูว่าเกิดการชนหรือไม่ โดยจับสัญญาณในเคเบิ้ลว่าเท่ากับ หรือมากกว่าสัญญาณที่เกิดจากทรานซีฟเวอร์ (Transceiver) 2 ตัวหรือมากกว่าทำการส่งในเวลาเดียวกัน โดยทรานซีฟเวอร์คืออุปกรณ์ไฟฟ้าที่ส่งและรับข้อมูลบนสายสื่อสาร ถ้าเกิดการชนแต่สเตชั่นอื่นไม่รู้ว่ามีเหตุการณ์นี้เกิดขึ้น ก็อาจจะพยายามส่งด้วย ซึ่งสเตชั่นดังกล่าวก็จะทำให้เกิดการชนอีก เพื่อหลีกเลี่ยงเหตุการณ์นี้สเตชั่นที่เกี่ยวข้องกับการชนก็จะส่งสัญญาณแจม (jam) เพื่อบอกให้สเตชั่นอื่นรู้ว่าขณะนี้เคเบิ้ลไม่ว่าง สัญญาณแจมคือการส่งข้อมูลออกไปอย่างน้อย 32 บิตที่ไม่เท่ากับค่าซีอาร์ซีของการส่งก่อนหน้านี้ หลังจากมีการชน สเตชั่นที่เกี่ยวข้องก็จะเพิ่มค่าพยายามส่ง (transmit attempt) ขึ้นอีกหนึ่ง

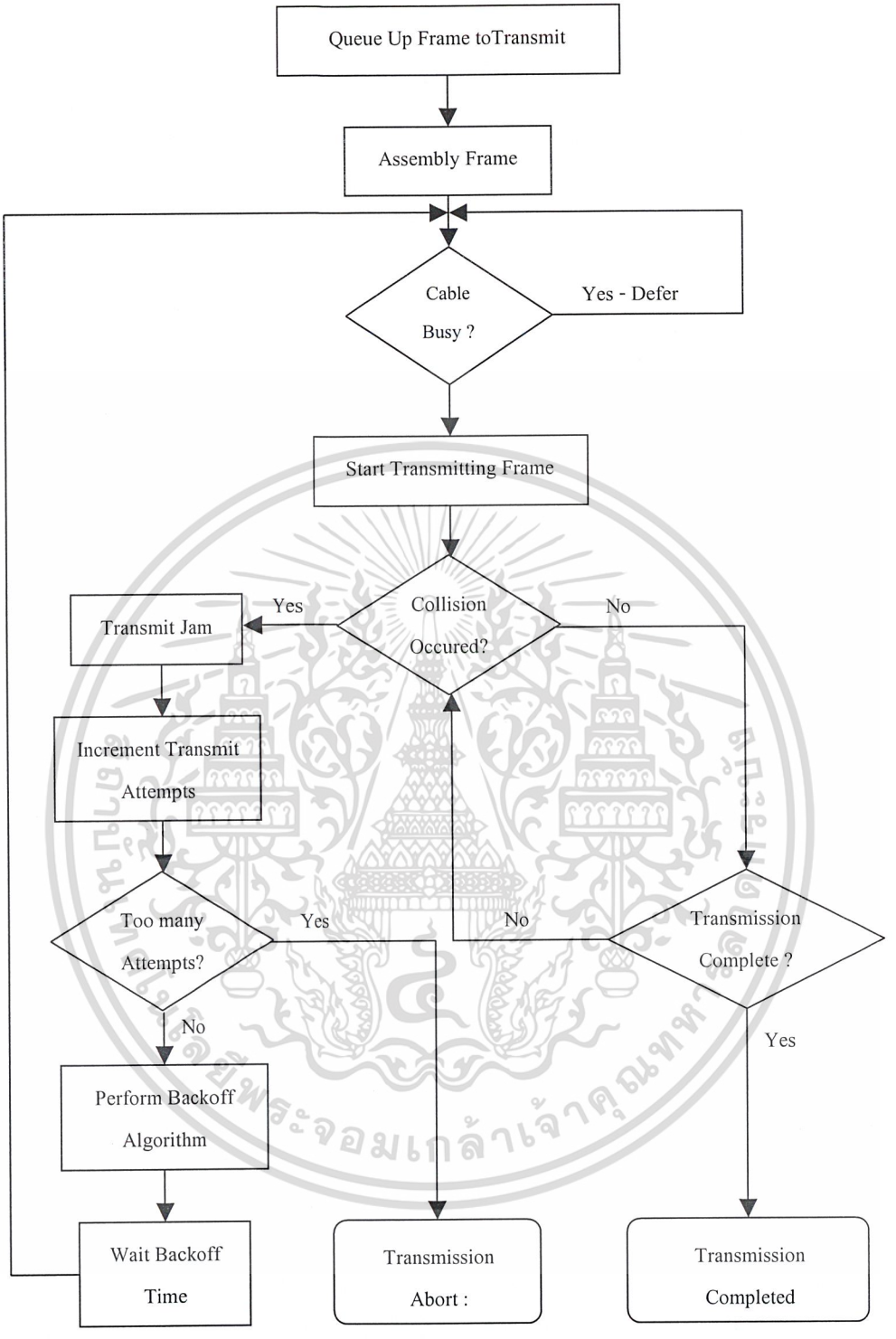
ขั้นตอนที่ 4 : รอก่อนจะส่งอีกครั้ง (Wait before retransmitting)

ถ้าสเตชั่นทำการส่งใหม่ทันทีหลังจากเกิดการชน ก็อาจจะเกิดการชนครั้งที่ 2 ได้อีก จำเป็นต้องสุ่ม (random) เวลาที่สเตชั่นจะต้องรอก่อนพยายามส่งอีกครั้งหนึ่ง สเตชั่นจะใช้อัลกอริธึมแบคออฟ (Backoff algorithm) เพื่อที่จะรู้ว่าเมื่อไหร่จะสามารถส่งใหม่ได้ ซึ่งจากอัลกอริธึมดังกล่าว จะได้เวลาที่สามารถจะใช้ได้จำนวนมาก เมื่อสเตชั่นสุ่มเลือกเวลาที่จะใช้ได้แล้ว จะทำให้เกดโอกาสน้อยลงที่จะเกิดการชนของข้อมูลจากสองสเตชั่นขึ้นไป

ขั้นตอนที่ 5 : ส่งใหม่หรือเลิกส่ง (Retransmit or Abort)

ถ้าสเตชั่นอยู่บนเซกเมนท์ที่มีการใช้งานมาก (busy) อาจทำให้ไม่สามารถส่งโดยไม่มีการชนเกิดขึ้นได้ ซึ่งสเตชั่นจะพยายามส่งจนถึง 16 ครั้ง จึงจะหยุดส่ง ถ้าสเตชั่นส่งใหม่แล้วพบว่าไม่มีการชนเกิดขั้นอีก จะถือว่าส่งสำเร็จ ถ้าส่งไม่สำเร็จเป็นจำนวน 16 ครั้ง ก็จะเลิกส่ง

เมื่อเข้าใจวิธีที่แต่ละสเตชั่นใช้ในการเข้าถึงสายสื่อสารแล้ว ก็ทำให้เรารู้ได้ว่าสเตชั่นส่งข้อมูลมีประสิทธิภาพหรือทำตามกฎหรือไม่ โฟล์วชาร์ตแสดงดังรูปที่ 3-24 ได้กำหนดแต่ละขั้นตอนที่ทุกสเตชั่นต้องทำ เพื่อส่งเฟรมไปบนเครือข่ายแลนซึ่งถ้ามีสเตชั่นใดไม่ทำตามขั้นตอนเหล่านี้ ก็จะทำให้เกิดปัญหาขึ้นในเซกเมนท์ที่มันใช้งานอยู่ได้



รูปที่ 3-24 โฟล์วชาร์ตแสดงการส่ง

การรับ (Receiveing)

แต่ละสเตชั่นที่ใช้งานอยู่ (active) จะต้องทำงานตามขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 : ดูแพ็คเก็ตที่เข้ามาและตรวจว่าถูกแบ่งส่วนหรือไม่ (View incoming packets and check for fragments)

ทุกสเตชั่นบนเซกเมนท์จะคอยดูแต่ละแพ็คเก็ตที่อยู่บนสายสื่อสารโดยไม่สนใจว่าแพ็คเก็ตนั้นส่งถึงมันหรือไม่ และสเตชั่นจะตรวจขนาดแพ็คเก็ตด้วยว่ามีขนาดปกติและไม่ถูกแบ่งส่วนย่อยเนื่องมาจากการชนหรือไม่

ขั้นตอนที่ 2 : ตรวจแอดเดรสปลายทาง (Check the destination address)

หลังจากพบว่าแพ็คเก็ตไม่ถูกแบ่งย่อย สเตชั่นก็จะตรวจสอบแอดเดรสปลายทางของแอดเดรสนั้น เพื่อดูว่าแพ็คเก็ตนั้นควรถูกประมวลผลต่อไปหรือไม่ ถ้าแพ็คเก็ตมีแอดเดรสปลายทางเป็นของสเตชั่นนั้น บรอดคาสท์ หรือมัลติคาสท์ สเตชั่นก็จะตรวจสอบความถูกต้องของแพ็คเก็ตต่อไป

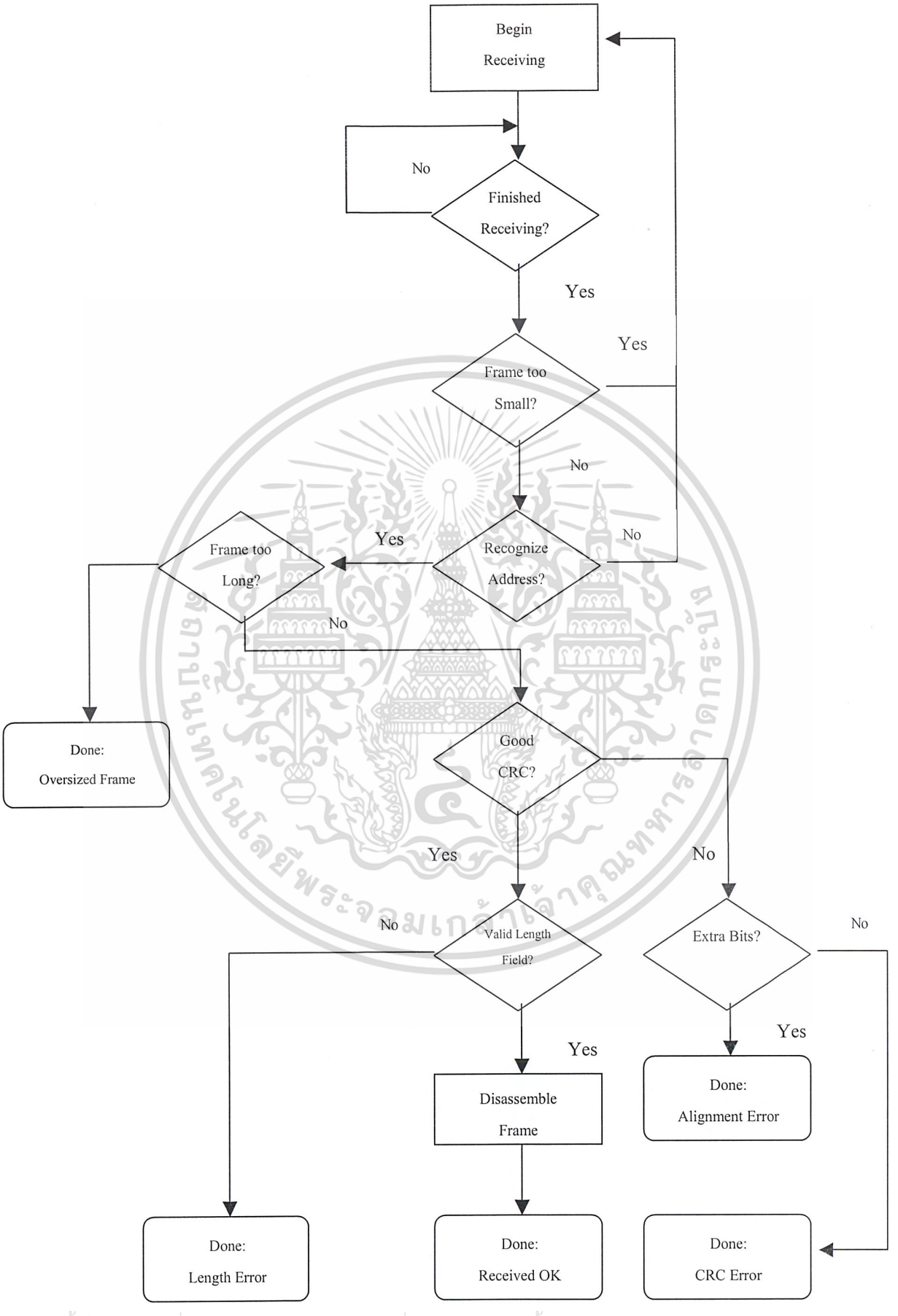
ขั้นตอนที่ 3 : ตรวจความถูกต้องสมบูรณ์ของแพ็คเก็ต (Check the integrity of the packet)

ณ จุดนี้สเตชั่นรู้แล้วว่าแพ็คเก็ตไม่ถูกแบ่งย่อยและมีแอดเดรสปลายทางของมันหรือเป็นแอดเดรสที่มันเป็นส่วนหนึ่งของกลุ่มด้วย แต่สเตชั่นไม่รู้ว่าแพ็คเก็ตมีรูปแบบที่เหมาะสมหรือไม่ เพราะสเตชั่นอาจจะกำลังอ่านแพ็คเก็ตที่ถูกทำให้เสียหายบนเซกเมนท์หรือถูกจัดรูปแบบไม่เหมาะสมโดยสเตชั่นที่สร้างแพ็คเก็ตนั้นขึ้นมา(เป็นผู้ส่ง) อยู่ก็ได้ ดังนั้นเพื่อหลีกเลี่ยงที่จะประมวลผลแพ็คเก็ตเหล่านี้ สเตชั่นที่ได้รับแพ็คเก็ตจะต้องทำการตรวจสอบคุณลักษณะต่าง ๆ ของแพ็คเก็ตก่อน โดยอันดับแรกจะต้องทำการตรวจว่าแพ็คเก็ตนั้นมีความยาวปกติหรือไม่ ถ้ายาวมากกว่า 1,518 ไบต์ จะถือว่าเป็นเฟรมที่ยาวมากเกินไป (Oversized frame) ซึ่งอาจเกิดความผิดพลาดของไดรเวอร์แวน (WAN driver) ถ้าแพ็คเก็ตมีขนาดไม่ยาวไปกว่าปกติแล้ว ก็จะถูกตรวจสอบต่อไปว่าข้อมูลที่มันนำมาด้วยนั้น เหมือนกับตอนที่ถูกส่งหรือไม่ เพราะอาจมีการสลับบิต 0 เป็น 1 หรือ บิต 1 เป็น 0 ขณะที่ถูกส่งมาในสายสื่อสารก็ได้ ซึ่งการตรวจสอบนั้นสามารถตรวจสอบได้โดยตรวจซีอาร์ซี (Cyclic Redundancy Check : CRC) ถ้าพบว่าซีอาร์ซีผิดพลาด แพ็คเก็ตนั้นก็จะถูกตรวจต่อไปว่าจบที่ขอบ 8 บิตหรือไม่ ถ้าไม่ (Misaligned packet) จะถือว่าเป็นข้อผิดพลาดที่เกิดจากมีบางไบต์ไม่ครบ 8 บิต (Alignment error) แต่ถ้าเป็นแพ็คเก็ตที่ทุกไบต์ครบ 8 บิต ก็จะถือว่าเป็นแพ็คเก็ตที่มีข้อผิดพลาดจากซีอาร์ซี (CRC error) ถ้าแพ็คเก็ตสามารถผ่านการตรวจที่กล่าวมาทั้งหมดได้ สเตชั่นก็จะทำการตรวจสอบความยาวอีกครั้งหนึ่ง เพื่อดูว่าแพ็คเก็ตมีขนาดสั้นเกินไปหรือไม่ ถ้าสั้นกว่า 64 ไบต์แต่มีรูปแบบปกติจะถือว่ามีขนาดสั้นกว่าปกติ (Undersizes) ซึ่งอาจเกิดจากไดรเวอร์แลน(LAN driver) การตรวจสอบที่กล่าวมาทั้งหมดนี้เพื่อให้แน่ใจได้ว่าแพ็คเก็ตที่ได้รับมีขนาดและข้อมูลปกติก่อนที่มันจะถูกประมวลผลต่อไป ถ้าเฟรมไม่ผ่านการตรวจจุดใดก็ตาม มันจะไม่ถูกส่งขึ้นไปให้โพรโตคอลในเลเยอร์ที่สูงกว่าเพื่อทำงานต่อไป

โฟล์วชาร์ตดังรูปที่ 3-25 กำหนดขั้นตอนที่สเตชั่นจะต้องทำก่อนการประมวลผลแพ็คเก็ตที่ได้รับมาในเครือข่ายที่ใช้วิธีเข้าถึงสายสื่อสารแบบซีเอสเอ็มเอ/ซีดี

ขั้นตอนที่ 4 : ประมวลผลแพ็คเก็ต (process the Packet)

ถ้าแพ็คเก็ตผ่านการตรวจสอบทุกจุดก็จะถือว่าเป็นแพ็คเก็ตปกติ สามารถนำมาใช้งานได้ ถ้ายังพบปัญหาอีก จะต้องดูเข้าไปในแพ็คเก็ตอีกเพื่อหาปัญหา ซึ่งอาจเนื่องมาจากสเตชั่นใช้ชนิดเฟรม (Frame Type) ผิด หรือมีความผิดพลาดที่เฮดเดอร์



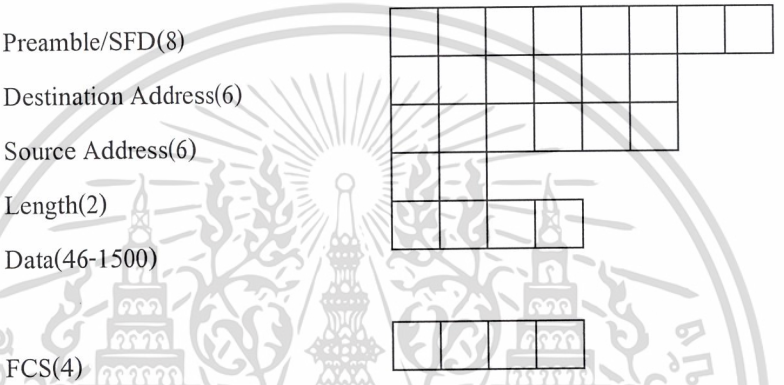
รูปที่ 3-25 โฟล์วชาร์ตแสดงการรับ

3.3.1 เฟรมอีเธอร์เนตแบบต่าง ๆ

ข้อมูลจำเป็นจะต้องถูกเอ็นแคปซูเลทในเฟรม เมื่อสเตชั่นเข้าถึงสายสื่อสาร ซึ่งเฟรมจะมีวิธีในการเข้าจังหวะ (Synchronizing) กับสเตชั่นที่รับข้อมูล รวมทั้งกำหนดหรือเก็บข่าวสารว่าใครคือต้นทาง หลายทาง และโพรโตคอลในเลเยอร์ที่สูงกว่าโพรโตคอลใดที่ใช้เฟรมนี้ โครงสร้างของเฟรมที่ใช้ในอีเธอร์เนตมีอยู่ 4 ลักษณะ ดังนี้

2.3.1.1 อีเธอร์เนต 802.3

มีโครงสร้าง ดังรูปที่ 3-26 ซึ่งประกอบด้วยฟิลด์ต่าง ๆ ดังต่อไปนี้



รูปที่ 3-26 เฟรมอีเธอร์เนต 802.3

1. Preamble และ ฟิลด์ Preamble Start Frame Delimiter (SFD) มีขนาด 7 ไบต์ ใช้เพื่อเข้าจังหวะกับสเตชั่นที่รับข้อมูล ซึ่งจะมีค่าเป็น 1 และ 0 สลับกัน (10101010...) ส่วนฟิลด์ SFD จะมีขนาด 1 ไบต์ อยู่ต่อท้ายฟิลด์ Preamble และมีค่าเป็น 1 และ 0 สลับกันเช่นเดียวกับฟิลด์ Preamble แต่จะลงท้ายด้วย 1 สองตัวติดกัน (10101011) ซึ่งจะช่วยให้รู้จุดเริ่มต้นของเฟรมได้
2. Destination Address มีขนาด 6 ไบต์ บรรจุฮาร์ดแวร์แอดเดรสหรือโหนดแอดเดรสของสเตชั่นบนเซกเมนท์เดียวกัน ซึ่งแพ็คเก็ตถูกกำหนดให้ส่งไป ถ้าแอดเดรสมีค่าเป็น 0xFF-FF-FF-FF-FF-FF จะหมายถึงบรอดคาสท์แอดเดรส
3. Source Address มีขนาด 6 ไบต์ ใช้บรรจุโหนดแอดเดรสของสเตชั่นบนเซกเมนท์เดียวกันที่เป็นผู้ส่งแพ็คเก็ต
4. Length มีขนาด 2 ไบต์ บอกถึงความยาวของข้อมูลจากเลเยอร์ที่สูงกว่าที่บรรจุอยู่ในฟิลด์ข้อมูลของเฟรม ซึ่งค่าจะต้องเป็น 1500 หรือน้อยกว่าสำหรับเฟรมอีเธอร์เนต 802.3 ที่ถูกต้อง
5. ฟิลด์ Data จะเป็นจุดที่เฮดเดอร์ของไอพีเอ็กซ์เริ่มต้น ความยาวของฟิลด์นี้จะอยู่ระหว่าง 46-1518 ไบต์ เพราะว่าโพรโตคอลไอพีเอ็กซ์/เอสพีเอ็กซ์ (IPX/SPX) ของโนเวล (Novell) เท่านั้นที่ใช้เฟรมอีเธอร์เนต 802.3

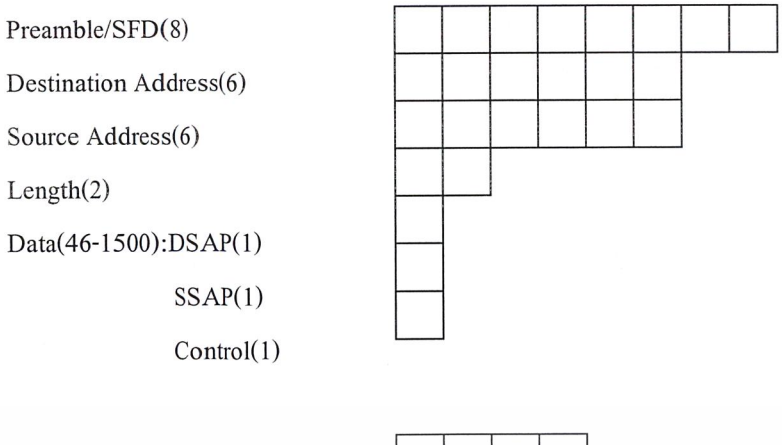
ดังนั้นฟิลด์ข้อมูลจะเริ่มด้วยเฮดเดอร์ของไอพีเอ็กซ์ ซึ่งเฮดเดอร์ไอพีเอ็กซ์จะเริ่มต้นด้วยค่า 0xFF-FF

6. Padding เพื่อที่จะให้เฟรมมีขนาดอย่างน้อย 64 ไบต์ เพราะฟิลด์ข้อมูลจะต้องมีขนาดอย่างน้อย 46 ไบต์ ซึ่งเมื่อรวมกับเฮดเดอร์อีก 18 ไบต์โดยไม่นับฟิลด์ Preamble และ SFD แล้วจะได้ 64 ไบต์ แต่ถ้าข้อมูลที่จะส่งมีไม่ถึง 46 ไบต์ ก็จะทำการเติมให้ครบ (padded) อย่างไรก็ตามอาจพบว่ามีการแพด (pad) 1 ไบต์ ถึงแม้ว่าตัวข้อมูลจะมีขนาดมากกว่า 46 ไบต์แล้วก็ตาม ทั้งนี้เพราะโนเวลร้องขอให้ผู้ผลิตไดรเวอร์แลนทุกแห่งพัฒนาไดรเวอร์แลนที่สร้างแพ็คเก็ตที่มีขนาดเป็นจำนวนคู่(Evenize) ซึ่งก็คือการเพิ่ม 1 ไบต์เข้าไป เหตุที่แพ็คเก็ตต้องเป็นเลขคู่เพราะว่าเราท์เตอร์บางตัวสามารถจัดการและประมวลผลได้เฉพาะแพ็คเก็ตที่มีความยาวเป็นจำนวนคู่เท่านั้น

7. Frame Check Sequence (FCS) การตรวจสอบข้อผิดพลาด(error checking) ถูกสร้างให้กับแต่ละเฟรมอีเธอร์เนต เพื่อให้มั่นใจได้ว่าเฟรมที่ถูกต้องเท่านั้นที่จะถูกประมวลผลโดยสเตชั่นที่ได้รับแพ็คเก็ต ฟิลด์ FCS มีขนาด 4 ไบต์บรรจุค่าซีอาร์ซี (CRC) โดยสเตชั่นที่เป็นผู้ส่งแพ็คเก็ตจะทำการคำนวณค่าซีอาร์ซีจากทุกฟิลด์ยกเว้นฟิลด์ Preamble และ SFD ก่อนส่งและใส่ค่านี้ลงในฟิลด์ FCS เมื่อสเตชั่นปลายทางได้รับข้อมูล ก็จะทำการคำนวณค่าซีอาร์ซีนี้อีกครั้ง เช่นเดียวกัน แล้วจะนำค่าที่คำนวณได้นี้มาเปรียบเทียบกับค่าในฟิลด์ FCS ซึ่งถ้าสอดคล้องกัน (เหมือนกัน) ก็แสดงว่าเฟรมที่รับมานั้นถูกต้อง

2.3.1.2 อีเธอร์เนต 802.2

ถูกพิจารณาว่าเป็นสิ่งที่สอดคล้องกับไอทริปเปิ้ลอี (IEEE-compliant) เพราะบรรจุทั้งฟิลด์ 802.3 และ 802.2 ฟิลด์ 802.2 ยังถูกเรียกว่าเลเยอร์แอลแอลซี (Logical Link Control : LLC) ภายในเฟรมอีเธอร์เนต 802.2 มีโครงสร้างดังรูปที่ 3-27 ซึ่งเริ่มต้นด้วยเฮดเดอร์ 802.3 ส่วนฟิลด์ 802.2 นั้นจะเริ่มต่อจากฟิลด์ Length ของเฮดเดอร์ 802.3 ซึ่งรายละเอียดของฟิลด์ 802.2 เป็นดังนี้



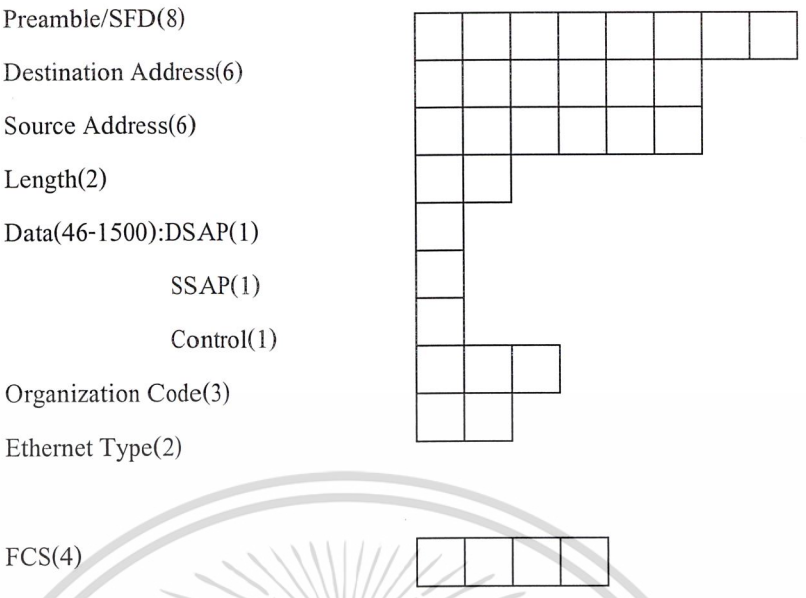
รูปที่ 3-27 โครงสร้างเฟรม 802.2

1. Destination Service Access Point (DSAP) มีขนาด 1 ไบต์ บอกถึงชนิดของโพรโตคอลในเลเยอร์ที่สูงกว่า (เลเยอร์เน็ตเวิร์ค) ณ ปลายทางของแพ็คเก็ต ถ้าเป็นแพ็คเก็ตที่ใช้โพรโตคอลไอพีเอ็กซ์/เอสพีเอ็กซ์ ฟิลด์นี้จะมีค่าเป็น 0xE0
2. Source Service Access Point (SSAP) มีขนาด 1 ไบต์ บอกถึงชนิดโพรโตคอลในเลเยอร์ที่สูงกว่า (หรือเลเยอร์เน็ตเวิร์ค) เช่นเดียวกับฟิลด์ DSAP ถ้าเป็นแพ็คเก็ตที่ใช้โพรโตคอลไอพีเอ็กซ์/เอสพีเอ็กซ์ ฟิลด์นี้จะมีค่าเป็น 0xE0
3. Control มีขนาด 1 ไบต์ ถ้าเฟรมนั้นใช้โพรโตคอลไอพีเอ็กซ์/เอสพีเอ็กซ์ ฟิลด์นี้จะค่าเป็น 0x03 ซึ่งหมายถึงเป็นเฟรม 802.2 แบบอันนัมเบอร์ (802.2 unnumbered format) หรือเลเยอร์แอลแอลซีให้บริการแบบไม่ต้องก่อตั้งการเชื่อมต่อก่อน (connectionless) นั่นเอง

ขนาดเฟรมน้อยที่สุดและมากที่สุดยังคงเป็น 64-1,518 ไบต์ ซึ่งไม่นับฟิลด์ Preamble และ SFD เช่นเดิม

2.3.1.3 อีเธอร์เนตสแนบ

สแนบ (SNAP) ย่อมาจาก Sub-Network Access Protocol ซึ่งเฟรมชนิดนี้พัฒนามาจากเฟรม 802.2 ซึ่งแสดงได้ดังรูปที่ 3-28 ซึ่งฟิลด์ต่าง ๆ จะเหมือนกับเฟรม 802.2 แต่มีส่วนที่แตกต่างกันดังนี้



รูปที่ 3-28 โครงสร้างเฟรมอีเธอร์เน็ตสแนป

1. DSAP , SSAP และ Control ในอีเธอร์เนตสแนป ค่าของฟิลด์ DSAP และ SSAP จะเป็น 0xAA เสมอ เพื่อบอกว่าเป็นเฟรมแบบอีเธอร์เนตสแนป ส่วนฟิลด์ Control นั้นจะมีขนาด 1 ไบต์ และมีค่า 0x03 (unnumbered) เสมอ และจะตามด้วยฟิลด์ Organization code และ Ethernet type

2. Organization code มีขนาด 3 ไบต์ บอกถึงว่าองค์การใดที่ใช้ฟิลด์ Ethernet type นั้น ๆ ซึ่งโพรโตคอลไอพีเอ็กซ์/เอสพีเอ็กซ์ของเน็ตแวร์ (NetWare) จะมีค่าในฟิลด์นี้เป็น 0x00-00-00

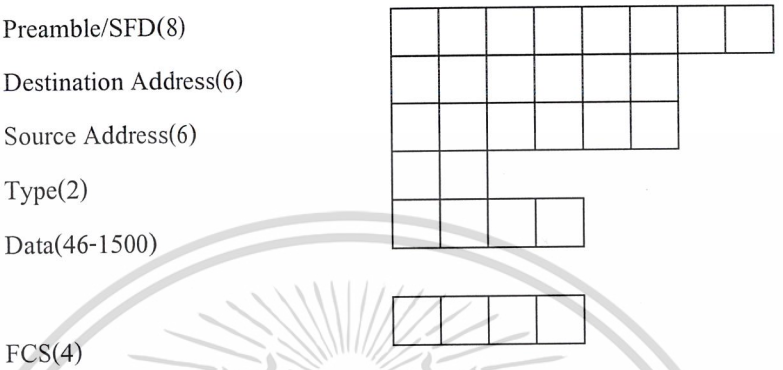
3. Ethernet type มีขนาด 2 ไบต์ ใช้เพื่อกำหนดโพรโตคอลของเลเยอร์ที่สูงกว่า ซึ่งฟิลด์ Ethernet type ของเน็ตแวร์จะมีค่าเป็น 0x8137 (และโนเวลยังได้จองหมายเลข 0x8138 ไว้ด้วย ถึงแม้ว่าจะยังไม่ใช้ในขณะนี้) และตามด้วยค่า 0xFF-FF ในฟิลด์ข้อมูลเช่นเดิม ตัวอย่างของค่าในฟิลด์นี้ของโพรโตคอลหลายชนิดเป็นดังนี้

IP (Internet Protocol)	0x0800
ARP	0x0806
RARP	0x8035
AppleTalk	0x809B
AppleTalk ARP	0x80F3
NetWare IPX/SPX	0x8137

ขนาดของเฟรมน้อยที่สุดและมากที่สุด ยังคงเป็น 64-1,518 ไบต์ ซึ่งไม่นับฟิลด์ Preamble และ SFD เช่นเดิม

2.3.1.4 อีเธอร์เนตทู

เฟรมอีเธอร์เนตทูแตกต่างจากเฟรมอื่น ๆ ตรงฟิลด์ Type ซึ่งตามหลังแอดเดรสต้นทาง ในขณะที่อีเธอร์เนต 802.3 อีเธอร์เนต 802.2 และอีเธอร์เนตสแนบที่มีฟิลด์ Length ตามหลังแอดเดรสต้นทาง โครงสร้างเฟรมอีเธอร์เนตทูแสดงได้ดังรูปที่ 3-29



รูปที่ 3-29 โครงสร้างของอีเธอร์เนตทู

โครงสร้างเฟรมของอีเธอร์เนตทูจะมี 2 อย่างที่แตกต่างจากโครงสร้างเฟรมแบบอื่น ๆ คือ ฟิลด์ Preamble/SFD และฟิลด์ Type

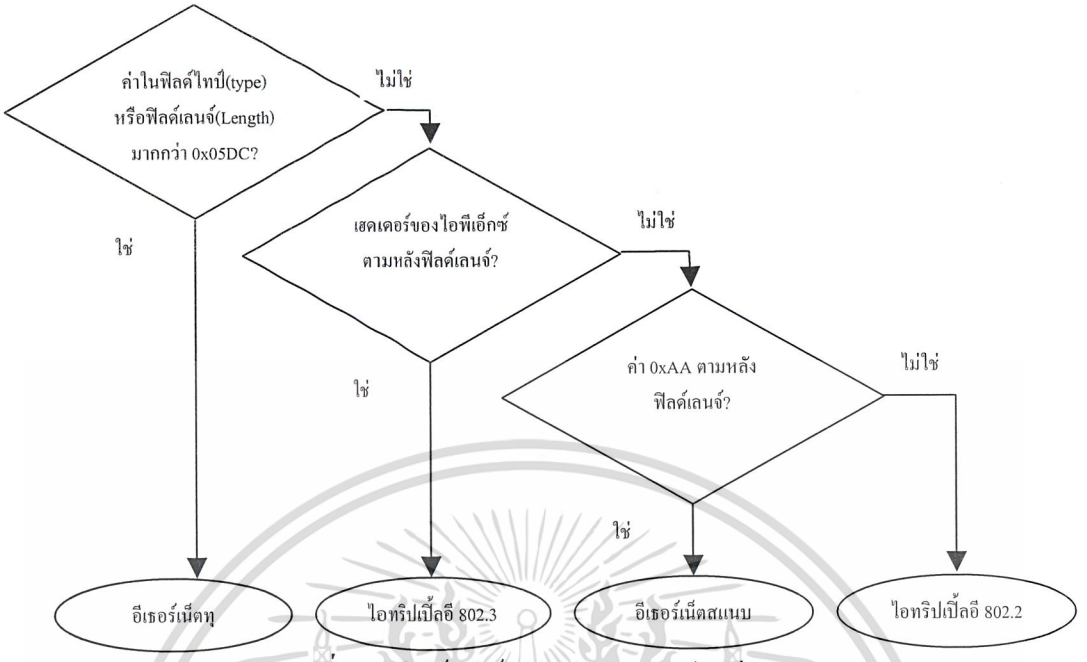
1. Preamble มีขนาด 8 ไบต์ บรรจุเลข 1 และ 0 สลับกัน เช่นเดียวกับฟิลด์นี้ของเฟรมแบบอื่น ๆ ที่มีขนาด 7 ไบต์ แต่ในเฟรมอีเธอร์เนตทู ฟิลด์ SFD ที่มีขนาด 1 ไบต์ (10101011) จะถูกพิจารณาว่าเป็นส่วนหนึ่งของฟิลด์ Preamble ด้วย

2. Type ไม่เหมือนเฟรมอื่น ๆ อีเธอร์เนตทูบรรจุฟิลด์นี้แทนฟิลด์ Length ซึ่งฟิลด์นี้จะระบุโพรโตคอลในชั้นที่สูงกว่าซึ่งใช้แพ็คเก็ตนี้ ตัวอย่างของค่าโพรโตคอลที่บรรจุในฟิลด์ Type ได้ซึ่งจะคล้ายกับค่าที่ใช้ในฟิลด์ Type ของอีเธอร์เนตสแนป คือ

IP (Internet Protocol	0x0800
ARP	0x0806
RARP	0x8035
AppleTalk	0x809B
AppleTalk ARP	0x80F3
NetWare IPX/SPX	0x8137

ขนาดของเฟรมที่น้อยที่สุดและมากที่สุด ยังคงเป็น 64-1,518 ไบต์ ซึ่งไม่นับฟิลด์ Preamble และ SFD เช่นเดิม

3.3.2 ความแตกต่างระหว่างเฟรมอีเธอร์เนตแบบต่าง ๆ



รูปที่ 3-30 โฟล์วชาร์ตแสดงแบบของเฟรมต่าง ๆ

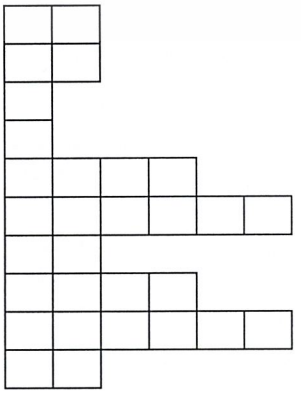
3.3.3 โพรโตคอลไอพีเอ็กซ์/เอสพีเอ็กซ์ของเน็ตแวร์

เมื่อไคลเอ็นท์และเซิร์ฟเวอร์ของเน็ตแวร์จะติดต่อสื่อสารกัน จะใช้บริการที่ต้องก่อตั้งการเชื่อมต่อหรือไม่ต้องก่อตั้งการเชื่อมต่ออย่างใดอย่างหนึ่ง ในการใช้บริการแบบต้องก่อตั้งการเชื่อมต่อก่อนนั้นจะต้องใช้โพรโตคอลเอสพีเอ็กซ์ แต่ถ้าจะใช้บริการแบบไม่ต้องการการก่อตั้งเชื่อมต่อก่อน ก็จะใช้โพรโตคอลไอพีเอ็กซ์ ซึ่งโพรโตคอลทั้งสองมีข้อแตกต่างกันคือ โพรโตคอลไอพีเอ็กซ์จะดีในแง่ที่จำเป็นต้องใช้เฮดเดอร์เป็นจำนวนที่น้อยกว่า แต่จะไม่รับรองว่าข้อมูลที่ส่งไปจะไปถึงหรือไม่ และเป็นลำดับอย่างไร ในขณะที่ไอพีเอ็กซ์จะมีลักษณะตรงข้าม

3.3.3.1 เฮดเดอร์ไอพีเอ็กซ์

จะมีความยาว 30 ไบต์และจะเริ่มต้นด้วยค่า 0xFFFF ซึ่งถูกส่งหลังจากเฟรมในเลเยอร์ย่อยแมค (MAC) ของเลเยอร์ดาต้าลิงค์ แต่จะอยู่ก่อนส่วนที่เป็นข้อมูล

Checksum(2)
Length(2)
Transport Control(1)
Packet Type(1)
Destination Network Address(4)
Destination Node Address(6)
Destination Socket(2)
Source Network Address(4)
Source Node Address(6)
Source Socket(2)



รูปที่ 3-31 แสดงรูปแบบของเฮดเดอร์ไอพีเอ็กซ์

1. Checksum

โดยปกติแล้วฟิลด์เช็คซัมในเฮดเดอร์ไอพีเอ็กซ์จะไม่ถูกใช้ และจะใส่ค่า 0xFFFF ลงไปในฟิลด์นี้ ซึ่งมีขนาด 2 ไบต์แทน โนเวลตั้งใจจะไม่ใช้ฟิลด์นี้เพราะการตรวจซีอาร์ซีก็ครอบคลุมทั้งเฟรมอยู่แล้ว แต่เราก็ยังสามารถใช้งานฟิลด์นี้ได้ถ้าต้องการ แต่จะใช้ไม่ได้เมื่อเฟรมเป็นอีเธอร์เนตแบบ 802.3 เพราะฟิลด์นี้เท่านั้นที่ทำหน้าที่ชี้ให้รู้ว่าเป็นเฟรมของเน็ตแวร์ ในขณะที่เฟรมอีเธอร์เนตอีก 3 แบบมีฟิลด์ที่ระบุโพรโตคอลในเลเยอร์ที่สูงกว่าอยู่แล้ว เช็คซัมในไอพีเอ็กซ์เฮดเดอร์นี้จะตรวจสอบเฉพาะเฮดเดอร์เท่านั้น ไม่รวมไปถึงข้อมูล

2. Length

มีขนาด 2 ไบต์ บรรจุความยาวของแพ็คเก็ตซึ่งรวมไอพีเอ็กซ์เฮดเดอร์และส่วนข้อมูลที่ตามมาเท่านั้นไม่ได้รวมฟิลด์ในส่วนเฟรมอีเธอร์เนต

3. Transport Control

มีขนาด 1 ไบต์ ฟิลด์นี้ถูกใช้โดยเราท์เตอร์ไอพีเอ็กซ์ และแสดงถึงจำนวนของเราท์เตอร์ที่แพ็คเก็ตไอพีเอ็กซ์ได้ผ่านมา สเตชั่นที่เป็นต้นกำเนิดจะเซตฟิลด์นี้ให้มีค่าเป็น 0 แพ็คเก็ตจะผ่านไปได้ถึง 15 เราท์เตอร์และจะถูกละทิ้งโดยเราท์เตอร์ตัวที่ 16

4. Packet Type

มีขนาด 1 ไบต์ ฟิลด์นี้แสดงให้ทราบว่าบริการประเภทใดที่แพ็คเก็ตนี้ใช้อยู่ สำหรับการสื่อสารที่ใช้ไอพีเอ็กซ์ ค่าจะเป็น 0 , 4 , 5 หรือ 17

IPX-base communications	0 or 4
SPX-base communications	5
NCP communications	17

5. Destination Network

ฟิลด์นี้บรรจุแอดเดรสของระบบเครือข่ายจำนวน 4 ไบต์ ซึ่งโหนดปลายทางตั้งอยู่ ถ้าฟิลด์นี้มีค่าเป็น 0x00-00-00-00 จะหมายถึงว่าปลายทางของแพ็คเก็ตนี้อยู่ในระบบเครือข่ายเดียวกันกับสเตชั่นต้นทาง

6. Destination Node

เป็นฟิลด์ที่มีขนาด 6 ไบต์ บรรจุหมายเลขซ็อคเก็ตของโพรเซสภายในโหนดที่แพ็คเก็ตถูกส่งไป หมายเลขซ็อคเก็ตที่รู้จักกันดีคือ

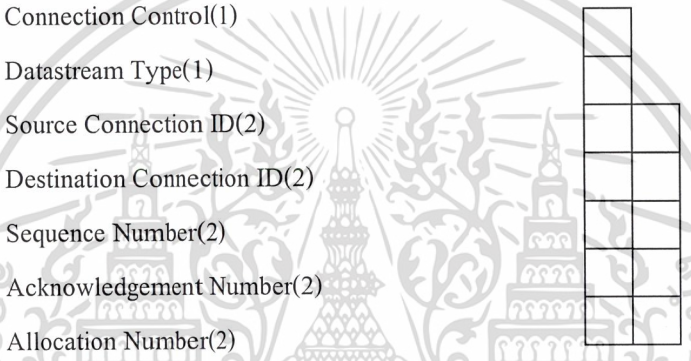
0x451	NetWare Core Protocol
0x452	Service Advertising Protocol packet
0x453	Routing Information Protocol packet
0x455	NetBIOS packet
0x456	Diagnostic packet
0x457	Serialization packet

ที่เวิร์คสเตชั่นหมายเลขซ็อคเก็ตจะถูกกำหนดแบบไดนามิค ซึ่งจะอยู่ในช่วง 0x4000 – 0x8000

8.	Source Network	มีขนาด 4 ไบต์ บรรจุแอดเดรสของระบบเครือข่ายที่โหนดต้นทางอยู่
9.	Source Node	มีขนาด 6 ไบต์ บรรจุแอดเดรสของสเตชั่นที่ส่งข้อมูล แต่ไม่สามารถบรรจุแอดเดรสสำหรับการบรอดคาสท์ในฟิลด์นี้ได้
10	Source Socket	มีขนาด 2 ไบต์ บรรจุหมายเลขซ็อคเก็ตของโพรเซสที่ส่งแพ็คเก็ตนี้

3.3.3.2 เฮดเดอร์เอสพีเอ็กซ์ (Sequence Packet Exchange : SPX)

มีความยาว 42 ไบต์ และจะเริ่มต้นด้วยฟิลด์ของเฮดเดอร์ไอพีเอ็กซ์ ประกอบด้วยฟิลด์ดังรูปที่2.32



รูปที่ 3-32 แสดงรูปแบบเฮดเดอร์ของเอสพีเอ็กซ์

1. Connection Control มี 1 ไบต์ ช่วยควบคุมการไหลของข้อมูลแบบสองทิศทางซึ่งตัวอย่างค่าของฟิลด์นี้คือ

Value	Name	Description
0x10	End-of-Message	บ่งบอกถึงไคลเอ็นท์ต้องการหยุดการติดต่อ
0x20	Attention	ฟิลด์นี้ยังไม่มีการนำมาใช้
0x40	Acknowledgement Required	ข้อมูลได้ถูกส่งไปแล้ว และต้องการการรับรองว่าข้อมูลได้ส่งไปถึงแล้วจริงๆ
0x80	System Packet	เป็นแพ็คเก็จแอคโนวเลจเมนท์ ซึ่งไอพีเอ็กซ์ใช้ฟิลด์นี้เป็นการภายใน เพราะมันไม่ได้ถูกส่งไปสู่เชลล์ของสเตชั่นปลายทาง

2. Datastream Type — ฟิลด์นี้มีขนาด 1 ไบต์ ใช้บ่งถึงชนิดของข้อมูลที่บรรจุอยู่ภายในแพ็คเก็จ ฟิลด์นี้ยังสามารถเก็บค่าที่ไคลเอ็นท์เป็นตัวกำหนด หรือค่าดังต่อไปนี้

Value	Name	Description
0xFE	End-of-Connection	สร้างขึ้นมาเพื่อให้ทราบว่าไคลเอ็นท์ต้องการจะหยุดการสื่อสาร
0xFF	End-of-Connection Acknowledgement	ถูกส่งเมื่อได้รับคำร้องขอในการหยุดการเชื่อมต่อ

3. Source Connection ID — เก็บตัวเลขจำนวน 2 ไบต์ ที่ถูกกำหนดโดยสเตชั่นต้นทางที่ใช้เอสพีเอ็กซ์ ซึ่งใช้สำหรับการดิมัลติเพล็กซ์ (Demultiplexing) การสื่อสารของเอสพีเอ็กซ์ เนื่องจากการเชื่อมต่อบนเครื่องสามารถใช้หมายเลขซ็อคเก็ตเดียวกันได้ ฟิลด์นี้จึงมีความสำคัญในการแยกแยะแต่ละการเชื่อมต่อเสมือน และหมายเลขที่บรรจุในฟิลด์นี้ก็ยังสามารถใช้กับฟิลด์ Destination Connection ID ได้โดยเอสพีเอ็กซ์อีกฝั่งเพื่อตอบสนอง

4. Destination Connection ID — มีขนาด 2 ไบต์ ใช้บรรจุหมายเลขการเชื่อมต่อของสเตชั่นปลายทางในระหว่างการก่อตั้งการเชื่อมต่อ ฟิลด์นี้จะถูกเซตค่าให้เป็น 0xFFFF เพราะว่าผู้ส่งยังไม่รู้ว่าหมายเลขการเชื่อมต่อปลายทางที่ผู้รับใช้เป็นเท่าใด

5. Sequence Number — เป็นฟิลด์ที่มีขนาด 2 ไบต์ ซึ่งบรรจุตัวเลขที่นับจำนวนข้อมูลที่ถูกส่งจากสเตชั่น ตัวเลขนี้จะถูกเพิ่มค่าเมื่อได้รับแอคโนวเลจเมนท์แล้วเท่านั้น สเตชั่นจะไม่เพิ่มค่าตัวเลขดังกล่าวนี้เมื่อทำการส่งแพ็คเก็ตที่ใช้ทำแอคโนวเลจเมนท์

6. Acknowledge Number — ระหว่างการส่งข้อมูล แพ็คเก็ตอาจเกิดการสูญหายได้ ฟิลด์แอคโนวเลจเมนท์จะบรรจุค่าของหมายเลขลำดับ (Sequence Number) ถัดไปที่คิดว่าจะได้รับ ถ้าหมายเลขลำดับไม่ถูกต้อง สเตชั่นที่เป็นผู้รับจะถือว่ามีข้อผิดพลาดเกิดขึ้นในการสื่อสาร ฟิลด์นี้มีขนาด 2 ไบต์

7. Allocation Number — มีขนาด 2 ไบต์ บ่งถึงจำนวนของบัฟเฟอร์ของการรับที่ว่างในสเตชั่นซึ่งจะมีค่าเริ่มที่ 0 สำหรับบัฟเฟอร์จะสามารถรับแพ็คเก็ตได้ 7 แพ็คเก็ต เมื่อมีการใช้ข้อมูลที่ได้รับมาโดยแอพพลิเคชั่น บัฟเฟอร์ก็จะว่างเพิ่มขึ้น ถ้าสเตชั่นกำลังยุ่งและไม่สามารถเคลียร์แพ็คเก็ตจากบัฟเฟอร์ได้ จำนวนบัฟเฟอร์ที่ว่างก็จะลดลงทุกครั้งที่สเตชั่นได้รับแพ็คเก็ต

บทที่ 4

หลักการทำงานของอุปกรณ์สวิตชิ่ง

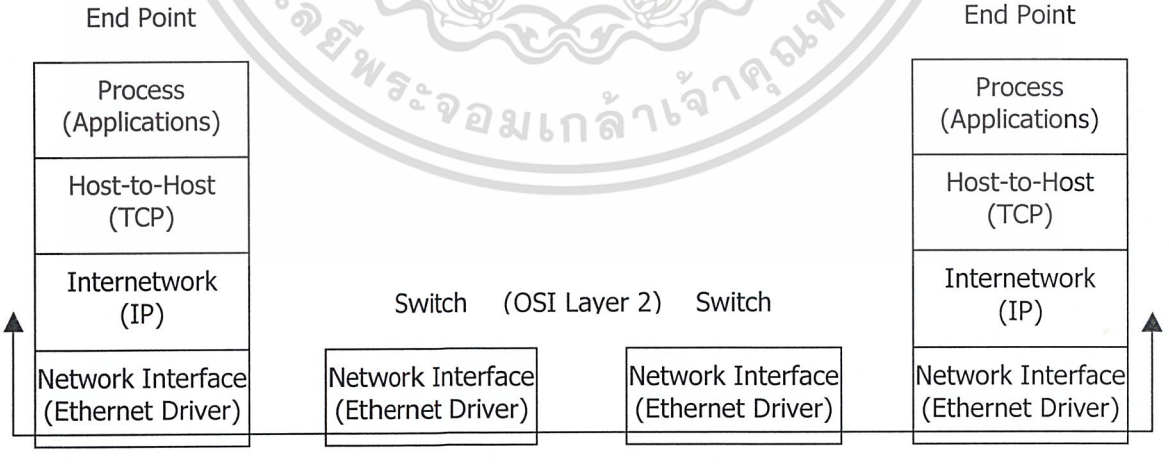
4.1 แนวคิดพื้นฐานของอุปกรณ์สวิตชิ่ง (Switching Concept)

4.1.1 คุณสมบัติพื้นฐาน 2 ประการ

- การดำเนินงานต่างๆ จะเกิดขึ้นภายใต้ระดับดาตาลิงค์ ของทุกๆ ชุดการสื่อสารข้อมูล (Protocol Stack) ซึ่งทำให้เกิดความเป็นเป็นอิสระ (Transparent) ไม่ขึ้นอยู่กับรูปแบบของการสื่อสารข้อมูลใดๆ ในระดับเน็ตเวิร์ค และระดับอื่นๆ ที่อยู่เหนือขึ้นไปด้วย
- การดำเนินงานต่างๆ จะกระทำโดยตัวอุปกรณ์ทางกายภาพ (Hardware) ทำให้สามารถส่งผ่านข้อมูลจากช่องทางการสื่อสารขาเข้าไปยังช่องทางการสื่อสารขาออกได้อย่างรวดเร็ว เนื่องจากไม่ต้องพึ่งพาการคำนวณของส่วนประมวลผลกลางแต่อย่างใด

4.1.2 มุมมอง 2 ประการ

- กลวิธีในการส่งต่อชุดข้อมูล (Switch Forwarding Technique) จะเป็นการพิจารณาว่าควรมีค่าใดบ้างที่จะต้องเก็บเอาไว้ในชุดข้อมูล (Packet) แต่ละชุดที่ถูกส่งเข้าไปในระบบเครือข่ายสวิตชิ่ง และตัวสวิตช์เองจะต้องจัดเก็บข้อมูลเหล่านั้นอย่างไร เพื่อให้สามารถส่งผ่านชุดข้อมูลเหล่านี้ได้รวดเร็ว
- กลวิธีในการควบคุมเส้นทาง (Switch Path Control) จะพิจารณาที่กระบวนการสร้างและรักษาเส้นทางลัดระหว่างจุดปลายใดๆ ในระบบเครือข่ายแบบสวิตช์



รูปที่ 4-1 *ระดับการเชื่อมต่อในลำดับชั้นของการสื่อสารบนระบบเครือข่ายสวิตชิ่ง (Protocol Stack)*

4.2 กลวิธีในการส่งต่อชุดข้อมูล

สวิตช์ (Switch) หรืออุปกรณ์สับเปลี่ยนช่องทางการสื่อสารในระบบเครือข่ายคอมพิวเตอร์ คือ อุปกรณ์ที่มีช่องทางการเชื่อมต่อ (Port) หลายช่องทาง ซึ่งแต่ละช่องทางก็จะมีอุปกรณ์ต่างๆ ต่อเชื่อมอยู่ เช่น เครื่องสถานีงาน (Workstation), อุปกรณ์จัดการเส้นทาง (Routers), หรืออุปกรณ์สลับช่องทางการสื่อสารด้วยกันเองก็ได้

งานของสวิตช์ ก็คือการเลือกที่จะละทิ้งหรือเคลื่อนย้ายชุดข้อมูลที่ได้รับเข้ามาจากช่องทางการสื่อสารหนึ่ง ไปยังช่องทางการสื่อสารอื่นที่เหมาะสมต่อไป โดยอาศัยข้อมูลที่มีอยู่ในแต่ละชุดข้อมูลเป็นหลัก แต่ในบางกรณีก็อาจจะต้องใช้ข้อมูลที่เก็บเอาไว้ภายในตัวสวิตช์เอง เข้ามาช่วยในการตัดสินใจด้วย

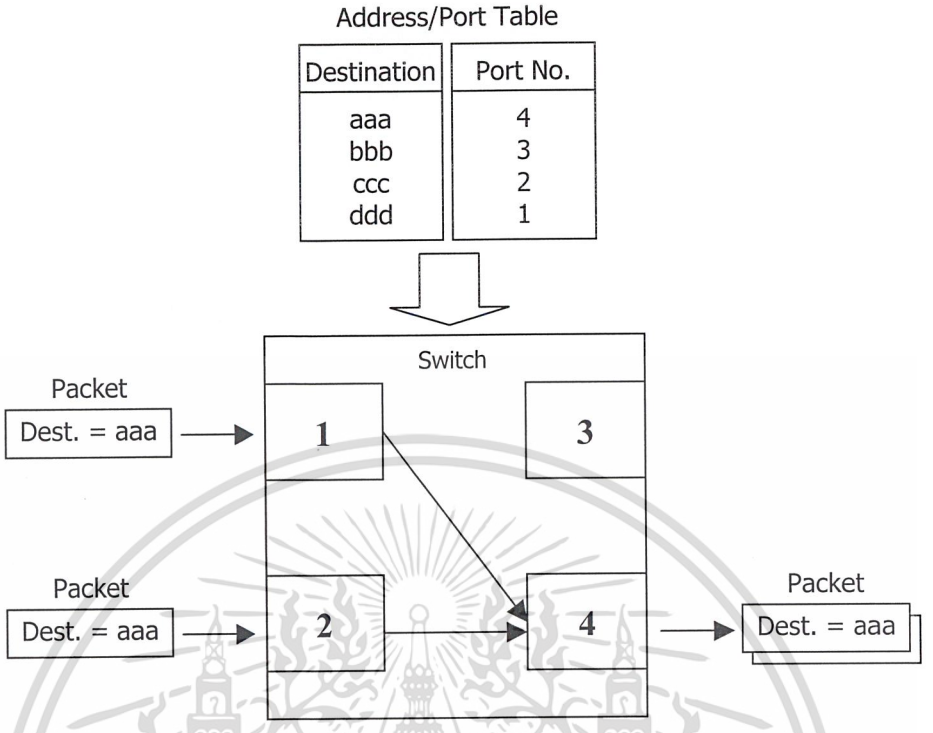
➢ ข้อมูลสำคัญที่จะต้องมีอยู่ภายในชุดข้อมูลแต่ละชุดที่ส่งผ่านระบบเครือข่ายสวิตช์นั้น จะต้องเป็นรูปแบบใดรูปแบบหนึ่ง ใน 3 รูปแบบดังต่อไปนี้เท่านั้น

- ค่าตำแหน่งเครื่องปลายทาง (Destination Address)
- ค่าทิศทางจากเครื่องต้นทาง (Source-Route Vector)
- ค่าระบุการช่องทางเชื่อมต่อ (Connection Identifier)

4.2.1 ในกรณีของการใช้ค่าตำแหน่งเครื่องปลายทาง (Destination Address)

เครื่องต้นทางจะทำหน้าที่เป็นเพียงผู้ป้อนชุดข้อมูลที่ระบุตำแหน่งเครื่องปลายทางเอาไว้แล้วเข้าสู่ระบบเครือข่าย ส่วนขั้นตอนการส่งต่อชุดข้อมูลจากต้นทางไปยังปลายทางนั้น จะเป็นหน้าที่ของระบบเครือข่ายนั้นๆ เอง โดยกลุ่มเครือข่ายของสวิตช์ที่เกี่ยวข้องในเส้นทางจะประสานการทำงานระหว่างกันเพื่อส่งต่อข้อมูลเป็นทอดๆ จนระทั่งชุดข้อมูลเหล่านั้นเดินทางไปถึงยังที่หมายตามที่ได้ระบุเอาไว้ในส่วนต้นของชุดข้อมูลแต่ละชุด (Packet Header) ซึ่งรูปแบบนี้จะนิยมใช้กันโดยทั่วไปในระบบสวิตช์เครือข่ายท้องถิ่นประเภทอีเธอร์เน็ต (Ethernet LAN Switching)

- ชุดข้อมูลทุกชุดจะต้องมีเขตข้อมูลตำแหน่งปลายทาง (Destination Address Field) เก็บเอาไว้ในส่วนต้นของชุดข้อมูล ซึ่งจะระบุค่าตำแหน่งที่อยู่ของเครื่องปลายทางเอาไว้
- สวิตช์ทุกตัวจะต้องมีตารางบันทึกค่าตำแหน่งที่อยู่ของเครื่องต้นทางกับช่องทางที่ชุดข้อมูลนั้นๆ ผ่านเข้ามา (Address/Port Table) เพื่อใช้สำหรับจับคู่ (Map) ค่าตำแหน่งปลายทางของชุดข้อมูลเข้ากับช่องทางการสื่อสารที่เหมาะสมที่จะส่งต่อชุดข้อมูลนั้นออกไป
- ถ้าหาช่องทางที่เหมาะสมไม่ได้ สวิตช์จะต้องปฏิบัติอย่างใดอย่างหนึ่งดังต่อไปนี้คือ
  - ⊕ กำจัดชุดข้อมูลนั้นทิ้งไปเลย (Drop Packet)
  - ⊕ ส่งต่อชุดข้อมูลนั้นๆ ออกไปยังทุกๆ ช่องทางการสื่อสารที่ต่อเชื่อมอยู่ด้วย ยกเว้นช่องทางที่ชุดข้อมูลผ่านเข้ามา (Flood Packet)
- หากพบว่าช่องทางการสื่อสารที่เหมาะสมเป็นช่องทางเดียวกับที่ชุดข้อมูลนั้นผ่านเข้ามา สวิตช์จะต้องกำจัดชุดข้อมูลเหล่านั้นทิ้งทันที เนื่องจากโดยหลักการแล้ว จะไม่มีการ ส่งต่อชุดข้อมูลย้อนกลับไปออกในทางที่ชุดข้อมูลนั้นผ่านเข้ามา เพราะจะทำให้ชุดข้อมูลถูกตีกลับไปในเครือข่ายต้นทางเรื่อยๆ อย่างไม่รู้จบนั่นเอง

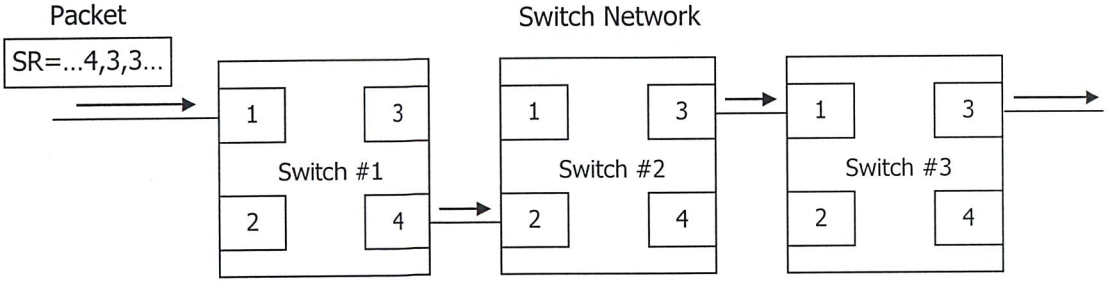


รูปที่ 4-2 กลไกการทำงานของกรณีใช้ค่าตำแหน่งเครื่องปลายทาง (Destination Address)

4.2.2 ในกรณีของการใช้ค่าทิศทางจากเครื่องต้นทาง (Source-Route Vector)

ค่าทิศทางจากเครื่องต้นทางคือค่ากำหนดลำดับของอุปกรณ์ในระบบเครือข่าย (Topological Elements) ที่ชุดข้อมูลจะต้องเดินทางผ่านไป ซึ่งลำดับของอุปกรณ์ในระบบเครือข่าย นี้ก็คือ ลำดับของหมายเลขช่องทางการสื่อสารภายในอุปกรณ์สวิตชิ่งแต่ละตัวที่อยู่ในเส้นทางนั่นเอง ดังนั้นการใช้ค่าทิศทางจากเครื่องต้นทาง จึงเป็นการระบุเส้นทางของชุดข้อมูลแต่ละชุดให้แก่สวิตช์ได้ทราบ ซึ่งสวิตช์จำเป็นต้องดำเนินการตามอย่างเลี่ยงไม่ได้นั่นเอง

รูปแบบนี้จะนิยมใช้กันโดยทั่วไปในระบบเครือข่ายแบบวงแหวนข้อมูล (Token Ring), และแบบเอทีเอ็ม (ATM) ในขั้นตอนการร้องขอการจัดสร้างช่องทางการสื่อสารเสมือนแบบส่งสัญญาณในระบบเอทีเอ็ม (ATM PNNI Phase I: Routing SVC Setup Request)



รูปที่ 4-3 กลไกการทำงานของกรณีที่ใช้ค่าทิศทางจากเครื่องต้นทาง (Source-Route Vector)

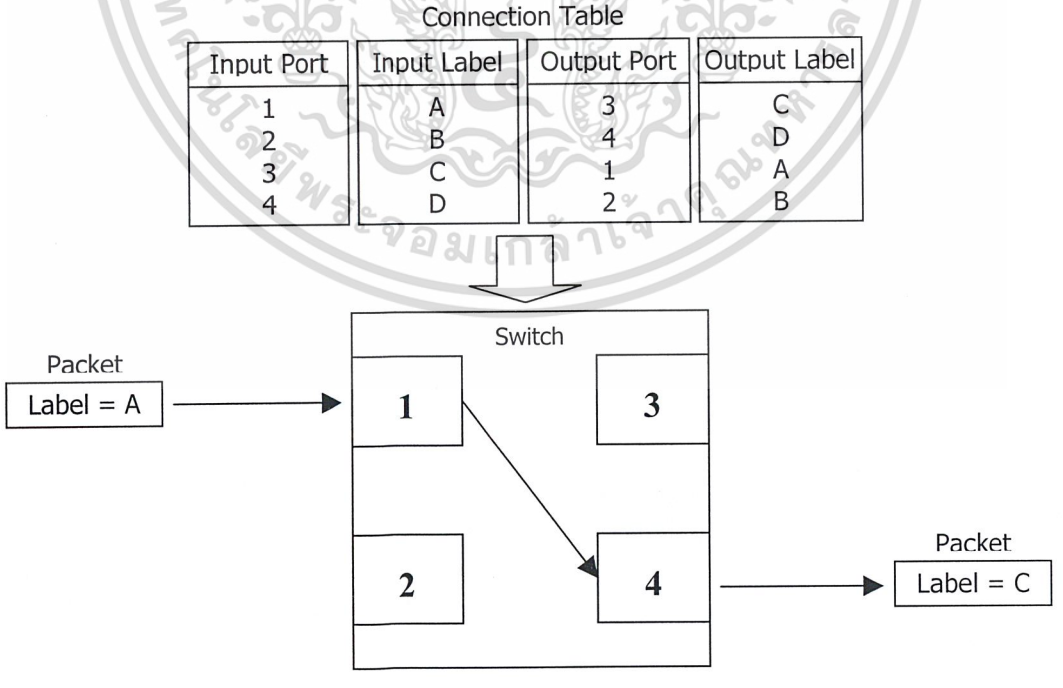
4.2.3 ในกรณีของการใช้ค่าระบุช่องทางการเชื่อมต่อ (Connection Identifier)

ค่าระบุช่องทางการเชื่อมต่อ เป็นค่าที่เก็บอยู่ในทุกๆ ชุดข้อมูล เพื่อบอกให้สวิตช์แต่ละตัวที่มันเดินทางผ่านทราบว่า สวิตช์ควรจะส่งต่อชุดข้อมูลนี้ออกไปยังช่องทางการสื่อสารใด เพื่อให้เกิดเป็นเส้นทางตามที่ต้องการนั่นเอง ซึ่งค่าระบุช่องทางการเชื่อมต่อนี้ ในบางครั้งอาจเรียกว่าฉลาก (Label)

ในกรณีของการใช้ค่าระบุช่องทางการเชื่อมต่อนั้น จะมีความแตกต่างกับ 2 วิธีที่ผ่านมาตรงที่ค่าที่เก็บเอาไว้ในชุดข้อมูลหรือฉลากนี้ จะไม่ได้เก็บไว้ที่ส่วนต้นของชุดข้อมูลแต่จะถูกนำเอาไปต่อไว้ที่ส่วนท้ายสุดของชุดข้อมูลแทน และฉลากนี้จะถูกสับเปลี่ยนไปเรื่อยๆ ทุกครั้งที่ชุดข้อมูลเดินทางผ่านอุปกรณ์สวิตช์ไป โดยสวิตช์จะมีโครงสร้างข้อมูลสำหรับจัดการฉลากและช่องทางการสื่อสารที่เรียกว่าตารางช่องทางการเชื่อมต่อ ซึ่งจะประกอบไปด้วยข้อมูล 4 ค่าคือ ค่าช่องทางการสื่อสารขาเข้า (Input Port), ค่าฉลากเดิมของชุดข้อมูล (Input Label), ค่าช่องทางการสื่อสารขาออก (Output Port), และค่าฉลากใหม่ของชุดข้อมูล (Output Label)

เมื่อชุดข้อมูลเดินทางมาถึงตัวอุปกรณ์สวิตช์ สวิตช์จะใช้ค่าช่องทางการสื่อสารที่ชุดข้อมูลนั้นผ่านเข้ามาร่วมกับค่าฉลากที่แนบมาด้วยในตอนท้ายของชุดข้อมูล เพื่อชี้ไปยังค่าช่องทางการสื่อสารขาออกและค่าฉลากใหม่ของชุดข้อมูลที่เหมาะสมจากตารางการเชื่อมต่อนี้ จากนั้นจะสับเปลี่ยนค่าฉลากให้กับชุดข้อมูล แล้วส่งชุดข้อมูลนั้นๆ ออกไปทางช่องทางการสื่อสารขาออกที่ระบุไว้ในตารางต่อไป ซึ่งการทำงานเหล่านี้มักจะเกิดขึ้นอย่างรวดเร็วมาก เนื่องจากสวิตช์ส่วนใหญ่จะดำเนินการด้วยอุปกรณ์ฮาร์ดแวร์ทั้งหมด

การทำการสับเปลี่ยนฉลาก (Label Switching) นี้ เป็นแบบอย่างการส่งต่อชุดข้อมูลที่ถูกใช้อยู่ในระบบเอทีเอ็ม, เฟรมรีเลย์ (Frame Relay), รวมทั้งผลิตภัณฑ์ไอพีสวิตช์ของหลายบริษัทอีกด้วย



รูปที่ 4-4 *กลไกการทำงานของกรณีที่ใช้ค่าระบุช่องทางการเชื่อมต่อ (Connection Identifier)*

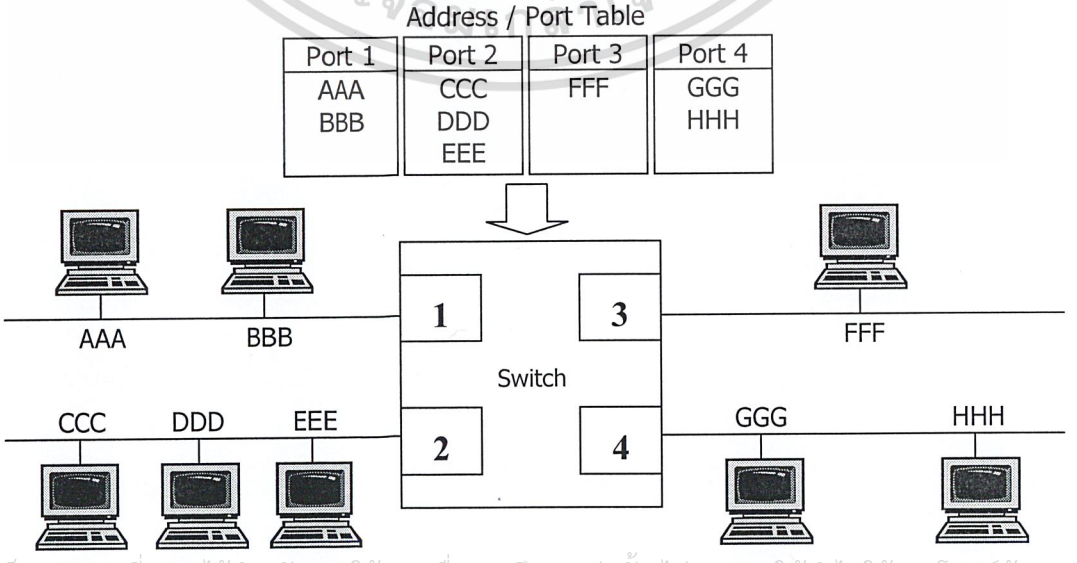
4.3 กลวิธีในการควบคุมเส้นทาง

การควบคุมเส้นทาง คือกระบวนการสร้างและเก็บรักษาข้อมูลที่ใช้ในการส่งต่อชุดข้อมูลของอุปกรณ์สวิตชิ่ง ซึ่งจะเกี่ยวข้องกับการแลกเปลี่ยน Control Message ระหว่าง Switch ด้วยกันเองบนระบบเครือข่าย เช่น Path setup message

- การควบคุมเส้นทางมีอยู่ด้วยกันหลายวิธีดังนี้
  - การเรียนรู้ตำแหน่ง (Address Learning)
  - แนวคิดแบบรากไม้เชิงกว้าง (Spanning Tree)
  - การสำรวจเส้นทาง (Broadcast & Discover)
  - การจัดการเส้นทางแบบสถานะการต่อเชื่อม (Link State Routing)
  - การใช้สัญญาณระบุเส้นทางที่ชัดเจน (Explicit Signaling)

4.3.1 การเรียนรู้ตำแหน่ง (Address Learning)

ในกรณีนี้ สวิตช์จะมีหน่วยความจำชั่วคราวชุดหนึ่ง (Address/Port Cache) ทำหน้าที่คอยดักทุกๆ ชุดข้อมูลที่ผ่านเข้ามาแล้วบันทึกเอาไว้ว่า ที่แต่ละช่องทางการสื่อสารนั้นมีชุดข้อมูลที่มีต้นกำเนิดมาจากที่ใดผ่านเข้ามาบ้าง แล้วสร้างเป็นตารางช่องทางการสื่อสารขึ้นมาโดยอาศัยค่าตำแหน่งที่อยู่ของเครื่องต้นทาง (Source Address) ของชุดข้อมูลแต่ละชุดที่ผ่านเข้ามานั่นเอง ส่วนการส่งต่อชุดข้อมูลนั้น หากสามารถจับคู่ค่าตำแหน่งที่อยู่ของเครื่องปลายทาง (Destination Address) เข้ากับหมายเลขช่องทางการสื่อสารใดในตารางได้ ก็จะส่งต่อชุดข้อมูลออกไปตามช่องทางนั้น เว้นเสียแต่ว่าหมายเลขช่องทางนั้นเป็นช่องทางเดียวกันกับที่ชุดข้อมูลผ่านเข้ามา สวิตช์ก็จะลบชุดข้อมูลนั้นทิ้งไป (Packet Dropping) ถ้าหากค่าตำแหน่งที่อยู่ของเครื่องปลายทางนั้นไม่มีอยู่ในตาราง สวิตช์ก็จะส่งต่อชุดข้อมูลเหล่านั้นออกไปทางทุกๆ ช่องทางการสื่อสารที่มีอยู่ (Packet Flooding) ยกเว้นช่องทางการสื่อสารที่ชุดข้อมูลใช้ผ่านนั้นเข้ามาเช่นกัน และเมื่อผ่านไประยะเวลาหนึ่งแล้วยังไม่สามารถระบุเส้นทางให้แก่ค่าตำแหน่งปลายทางใดๆ ได้ สวิตช์จะถือว่าค่าที่บันทึกไว้นั้นล้าสมัย (Cache – Time out) และจะลบค่าเหล่านั้นทิ้งไป



รูปที่ 4-5 กลไกการทำงานของกรรมวิธีการเรียนรู้ตำแหน่ง (Address Learning)

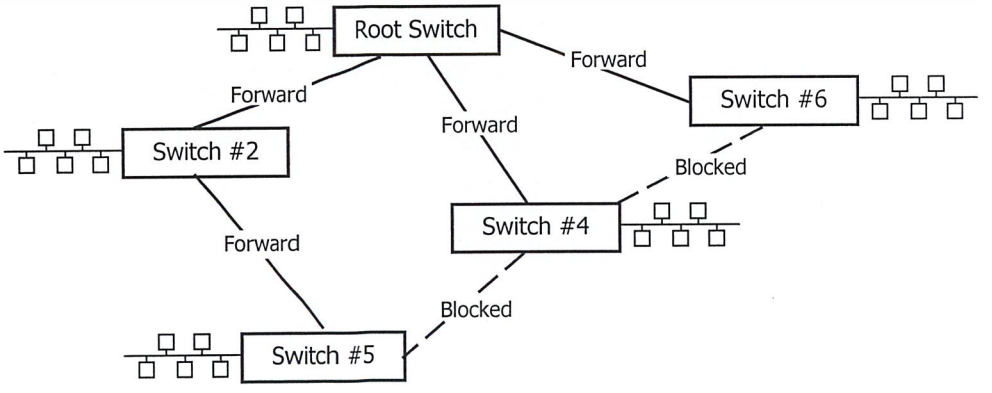
การเรียนรู้ตำแหน่งเป็นวิธีการที่ง่ายและนิยมใช้กันโดยทั่วไปในระบบ Ethernet Transparent Bridge แต่ข้อเสียที่สำคัญของวิธีการนี้คือ ปัญหาการวนกลับ (Loop) ของเส้นทางนั่นเอง ซึ่งจะเกิดขึ้นในกรณีที่มี เส้นทางจากเครื่องต้นทางไปยังเครื่องปลายทางมากกว่า 1 เส้นทาง โดยปัญหาการวนกลับจะเกิดขึ้นถ้าชุดข้อมูลจากระบบเครือข่ายต้นทางถูกส่งผ่านบริดจ์ (Bridge) หรือกลุ่มของบริดจ์ไปยังเครื่องปลายทางผ่านเส้นทางเส้นทางหนึ่ง แต่ได้รับการตอบกลับมาจากอีกเส้นทางหนึ่งที่ไม่เกี่ยวข้องกัน บริดจ์ต่างๆ ที่อยู่บนเส้นทางแรกก็จะเข้าใจว่าชุดข้อมูลเหล่านั้นมีต้นกำเนิดมาจากระบบเครือข่ายแรก จึงส่งต่อชุดข้อมูลเหล่านั้นกลับไปยังระบบเครือข่ายที่สองอีกครั้ง บริดจ์บนระบบเครือข่ายที่สองก็จะเข้าใจว่าชุดข้อมูลนั้นมีต้นกำเนิดมาจากระบบเครือข่ายที่สองเองและก็ส่งต่อชุดข้อมูลเหล่านั้นกลับไปยังระบบเครือข่ายแรก เช่นนี้เรื่อยไปไม่จบสิ้นนั่นเอง

4.3.2 แนวคิดแบบรากไม้เชิงกว้าง (Spanning Tree Algorithm)

แนวคิดรากไม้ในเชิงกว้างได้รับการพัฒนาขึ้นมาเพื่อแก้ไขปัญหาเรื่องการวนกลับของเส้นทางการสื่อสารข้อมูลในระบบเครือข่าย และได้ถูกระบุเอาไว้ในมาตรฐานการสื่อสารข้อมูล IEEE 802.1d LAN Bridging Standard

Spanning Tree คือส่วนย่อยส่วนหนึ่งของโครงสร้างการเชื่อมต่อทั้งหมดในระบบเครือข่าย (Network Topology) ที่ใช้ในการส่งผ่านชุดข้อมูลของการสื่อสารข้อมูลใดๆ ที่เกิดขึ้น ซึ่งก็หมายถึงตัวอุปกรณ์สวิตช์และเส้นทางการเชื่อมต่อที่ต่อเข้ากับสวิตช์ตั้งต้น (Root Switch) ของการสื่อสารนั้นๆ ที่ไม่ก่อให้เกิดการวนกลับของเส้นทางด้วยนั่นเอง การสื่อสารของข้อมูล (Data Traffic) ต่างๆ ที่เกิดขึ้นนั้นสามารถนำมารวมเข้าด้วยกัน (Concentrate) บนกลุ่มของสวิตช์และเส้นทางการเชื่อมต่อกลุ่มเล็กๆ ในระบบเครือข่ายได้ โดยขึ้นอยู่กับรูปแบบของการเชื่อมต่อ (Topology)

สวิตช์จะสร้างและรักษาโครงสร้างข้อมูลแบบรากไม้นี้เอาไว้ โดยแลกเปลี่ยนข้อความควบคุม (Control Message) ระหว่างกัน ซึ่งข้อความควบคุมเหล่านี้จะประกอบไปด้วยหมายเลขประจำตัวของสวิตช์ตั้งต้น (Root Switch ID), ระยะทางจากสวิตช์ตั้งต้นไปยังสวิตช์ตัวอื่นๆ และช่องทางการเชื่อมต่อที่ใกล้กับสวิตช์ตั้งต้นที่สุด ทั้งหมดนี้จะถูกรวมกันเข้าเป็นโครงสร้างรากไม้โครงสร้างหนึ่ง ช่องทางการเชื่อมต่ออื่นๆ ที่เหลือจะถูกตัดทิ้งไป (Blocked) ทำให้เหลืออยู่เฉพาะกลุ่มของสวิตช์และช่องทางการเชื่อมต่อที่ทำให้ไม่เกิดการวนกลับของเส้นทางเท่านั้น ที่อยู่ในโครงสร้างรากไม้นี้



รูปที่ 4-6 กลไกการทำงานของระบบเครือข่าย ตามแนวคิดแบบรากไม้ (Spanning Tree Algorithm)

แนวคิดนี้ถูกนำมาใช้อย่างแพร่หลายในระบบเครือข่ายที่เป็น Ethernet Transparent Bridge และระบบ Token-Ring Source Routing Bridge และค่อนข้างมีประโยชน์อย่างมากกับระบบเครือข่ายท้องถิ่นแบบขยาย (Extended LAN Environments) ที่ต่อเชื่อมกันด้วยบริดจ์หรือสวิตช์ อย่างไรก็ตามแนวคิดนี้ก็ยังมีขีดจำกัดในเรื่องความสามารถในการขยายระบบ (Scalability) และความแข็งแกร่งของระบบ (Robustness) อยู่บ้างคือ

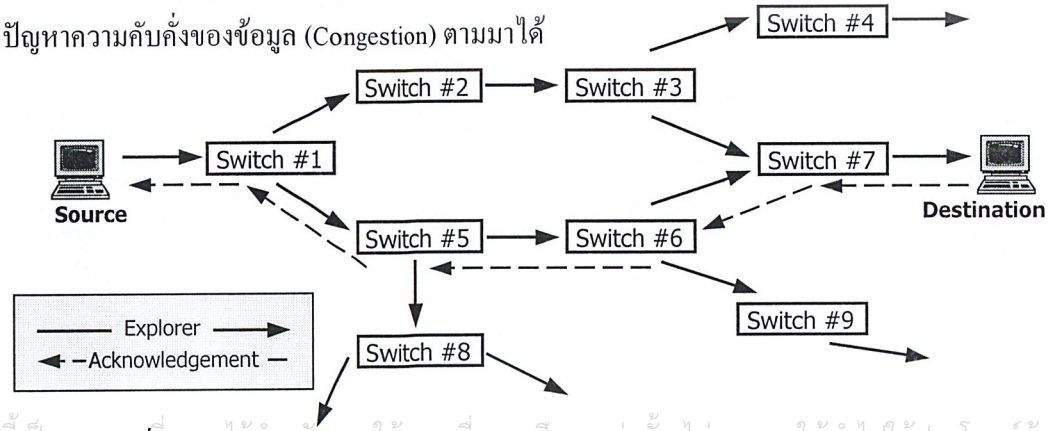
- เนื่องจาก Spanning Tree จะเป็นเพียงส่วนย่อยส่วนหนึ่งของระบบเครือข่ายทั้งหมดที่มีอยู่ ทำให้ในบางครั้งเส้นทางที่ถูก Block ไป อาจเป็นเส้นทางที่ดีกว่าเส้นทางที่ถูกเลือกขึ้นมาก็เป็นได้
- การใช้งานระบบเครือข่ายตามรูปแบบนี้จะมี Overhead สูง เนื่องมาจากต้อง

4.3.3 การสำรวจเส้นทาง Broadcast and Discover

การสำรวจเส้นทางเป็นอีกวิธีการหนึ่งที่นิยมนำมาใช้กับระบบ LAN Switching และ Bridging เพื่อจัดตั้งเส้นทางการสวิตช์ในระบบเครือข่ายขึ้นมา การสำรวจเส้นทางทั้งหมดที่เป็นไปได้ จะทำโดยการส่งชุดข้อมูลพิเศษที่เรียกว่า Broadcast Explorer Packet กระจายออกไปในทุกๆ เส้นทางที่สามารถไปถึงยังเครื่องปลายทางที่ต้องการได้ แล้วเมื่อเครื่องปลายทางได้รับชุดข้อมูลพิเศษนี้เอาไว้ ก็จะตอบกลับมาทางเส้นทางใดเส้นทางหนึ่ง ซึ่งชุดข้อมูลที่ตอบกลับมานี้จะมีค่าระบุเส้นทางการเชื่อมต่อ (Source Route Vector) ที่ดีที่สุดที่เครื่องต้นทางสามารถนำมาใช้ส่งต่อชุดข้อมูลส่วนที่เหลือกลับไปยังเครื่องปลายทางตามที่ต้องการได้ด้วยนั่นเอง

ชุดข้อมูลพิเศษเพื่อใช้สำรวจเส้นทาง (Explorer Packet) จะถูกแพร่กระจาย (Flooding) ออกไปทั่วทั้งระบบเครือข่าย ซึ่งในที่นี้จะเรียกว่าชุดข้อมูลพิเศษเหล่านี้ว่า ARE หรือ All Route Explorer และทำ Load Balancing ได้ แต่ถ้าหากระบบเครือข่ายมีรูปแบบเป็นระบบรากไม้เชิงกว้างอยู่แล้วด้วย ชุดข้อมูลพิเศษเหล่านี้ก็จะถูกเลือกให้เดินทางไปเฉพาะในเส้นทางตามที่ได้กำหนดไว้ในโครงสร้างข้อมูลแบบรากไม้ (Spanning Tree) เท่านั้น และจะถูกเรียกว่าชุดข้อมูล STE หรือ Spanning Tree Explorer นั่นเอง

เนื่องจากกลวิธีการควบคุมเส้นทางตามแบบนี้จะต้องใช้งานร่วมกับกลวิธีการส่งต่อข้อมูลแบบ Source Route จึงทำให้ในส่วนของตัวสวิตช์เองนั้นเกิด Overhead น้อยลง แต่ก็มีข้อเสียตรงที่ การใช้ ARE ในตอนเริ่มต้นนั้น จะสิ้นเปลืองขีดความสามารถ (Bandwidth) ของเส้นทางไปอย่างมาก และอาจนำไปสู่ปัญหาความคับคั่งของข้อมูล (Congestion) ตามมาได้



รูปที่ 4-7 กลไกการทำงานของกรรมวิธีการสำรวจเส้นทาง (Broadcast and Discover)

4.3.4 การจัดการเส้นทางแบบสถานะการต่อเชื่อม (Link State Routing)

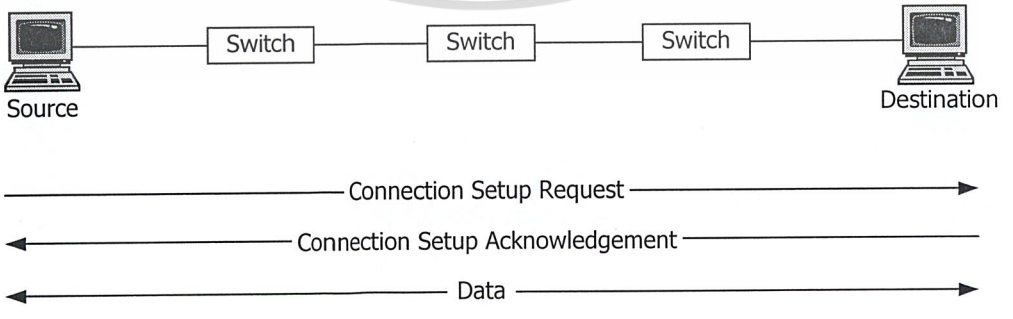
กลวิธีนี้จะสมมติให้สวิตช์ดำเนินการตามรูปแบบข้อตกลงการจัดการเส้นทางแบบสถานะการต่อเชื่อม (Link State Routing Protocol) ซึ่งสวิตช์แต่ละตัวจะต้องแลกเปลี่ยนข้อมูลสถานะการต่อเชื่อมของตนเองให้กับสวิตช์ตัวอื่นๆ ที่อยู่รอบข้างทราบ และนำมาข้อมูลเหล่านี้มาทำเป็นฐานข้อมูลการเชื่อมต่อรวมของทั้งเครือข่ายขึ้นมา ซึ่งจะประกอบไปด้วย Active Link และ Node ทั้งหมดที่มีอยู่

เนื่องจากสวิตช์แต่ละตัวจะรู้จักกับ Link และ Node ต่างๆ ที่ทำงานอยู่ภายในเครือข่ายเดียวกันได้ดี ดังนั้นชุดข้อมูลจึงสามารถถูกส่งผ่านไปยังเป้าหมายปลายทางในลักษณะเป็นทอดๆ (Hop-by-Hop หรือ Destination-Based Switching) ได้ และเมื่อสวิตช์มีฐานข้อมูลสถานะการต่อเชื่อมในบริเวณพื้นที่ของตนเองแล้ว สวิตช์ตัวแรกที่ได้รับชุดข้อมูลเข้ามาจะทำหน้าที่ตรวจสอบหาเส้นทางการต่อเชื่อมจากฐานข้อมูลของมันและรวบรวมขึ้นเป็นค่าทิศทาง (Source Route Vector) แนบไปกับชุดข้อมูลแล้วส่งออกไปตามเส้นทางที่ระบุไว้

การจัดการเส้นทางแบบสถานะการต่อเชื่อมนี้จะถูกใช้กับข้อตกลงการจัดการเส้นทางชนิดไอพี อันได้แก่ OSPF และ IS-IS เพื่อสร้างตารางการส่งผ่านข้อมูลในอุปกรณ์ค้นหาเส้นทาง

4.3.5 การใช้สัญญาณระบุเส้นทางที่ชัดเจน (Explicit Signaling)

การควบคุมเส้นทางตามรูปแบบนี้ เครื่องต้นทางจะต้องส่งข้อความร้องขอการจัดสร้างการเชื่อมต่อกับเครื่องปลายทาง (Setup Request Message) ออกไปในระบบเครือข่ายเสียก่อน เพื่อตรวจสอบให้มั่นใจว่า ณ เวลานั้นๆ มีเส้นทาง (Active Path) ที่สามารถส่งผ่านการสื่อสารข้อมูลไปยังเครื่องปลายทางทำงานอยู่จริงในระบบเครือข่ายหรือไม่ และหากเป็นไปได้อาจตรวจสอบด้วยว่าเส้นทางที่มีอยู่นั้น มีความสามารถเพียงพอที่จะรองรับการสื่อสารข้อมูลที่จะเกิดขึ้นมาได้หรือไม่อีกด้วย จากนั้นเมื่อเครื่องต้นทางได้รับข้อความตอบรับ (Acknowledgment) ยืนยันการเชื่อมต่อกลับมาจากเครื่องปลายทางแล้ว เครื่องต้นทางจึงจะส่งผ่านชุดข้อมูลออกไปตามเส้นทางที่ได้รับการจัดสรรขึ้นมาเฉพาะ (Dedicated Switched Path) เป็นที่เรียบร้อยแล้วเท่านั้น



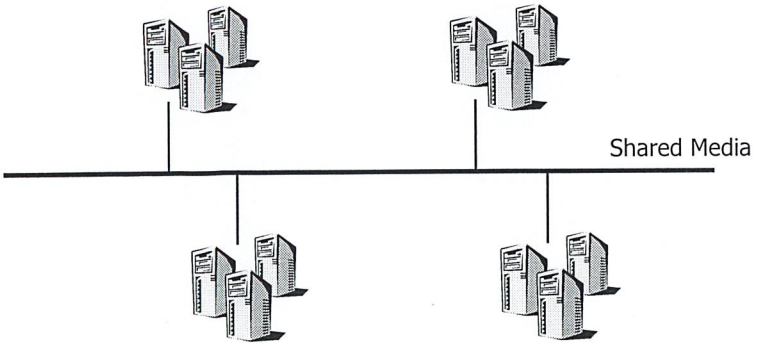
รูปที่ 4-8 *ขั้นตอนการทำงานของระบบสัญญาณระบุแสดงเส้นทางที่ชัดเจน (Explicit Signaling)*

- ข้อดีของการควบคุมเส้นทางตามรูปแบบนี้คือ
  1. สามารถยืนยันได้ว่ามีเส้นทางที่แท้จริงไปยังเครื่องปลายทางหรือไม่ ก่อนที่จะส่งข้อมูลออกนั้นๆ ออกไป ทำให้มั่นใจได้ว่าข้อมูลจะไปถึงปลายทางอย่างแน่นอน
  2. การคำนวณเส้นทางจะทำเพียงครั้งเดียว ในช่วงเริ่มต้นจัดสร้างเส้นทางเท่านั้น
  3. เมื่อทราบว่ามีเส้นทางอยู่แน่นอนแล้ว ก็สามารถลด Overhead ในส่วนของการคำนวณความถูกต้องของเส้นทางที่ชุดข้อมูลถูกส่งออกไปได้
  4. การสื่อสารข้อมูลพิเศษที่มีการระบุค่าขีดความสามารถต่ำสุดของระบบเครือข่าย (Minimum Bandwidth) หรือการหน่วงเวลาสูงสุด (Maximum Delay) ที่สามารถยอมรับได้เอาไว้ จะทำงานกับระบบเครือข่ายแบบนี้ได้เป็นอย่างดี
  5. เป็นรูปแบบใช้ได้ดีกับระบบคิดบัญชีค่าใช้บริการระบบเครือข่ายที่ยึดเอาการเชื่อมต่อเป็นหลัก (Per-Connection Basis Accounting and Billing)
- ข้อเสียของการควบคุมเส้นทางตามรูปแบบนี้คือ
  1. ระบบของตัวสวิตช์เอง จะต้องมีความซับซ้อนเพิ่มขึ้นมาก
  2. การหน่วงเวลาในช่วงเริ่มต้นของการสื่อสารข้อมูลจะเกิดขึ้นมาก

4.4 การสับเปลี่ยนช่องทางการสื่อสารบนระบบเครือข่ายท้องถิ่น (LAN Switching)

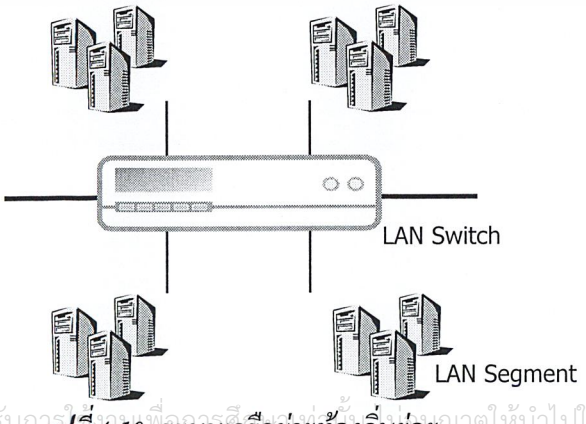
LAN Switching เป็นวิวัฒนาการมาจากทฤษฎีระบบเครือข่ายท้องถิ่นแบบดั้งเดิม เริ่มตั้งแต่การจัดกลุ่มวางเครื่องสถานีงาน (Workstations) ที่ทำให้ผู้ใช้งานทั่วๆ ไป (End Users) สามารถติดต่อถึงกันและใช้งานข้อมูลรวมทั้งทรัพยากรเครือข่ายต่างๆ เช่น Printer ร่วมกันได้ อย่างสะดวกสบายมากยิ่งขึ้น โดยในกรณีของระบบเครือข่ายแบบอีเธอร์เน็ตนั้น เพียงแค่ผู้ใช้แต่ละคนนำเครื่องของตนเข้ามาต่อ (Tap) เข้ากับสายส่งข้อมูล (Cable) หลัก ที่ให้บริการเป็นสื่อกลางสำหรับส่งผ่านข้อมูลร่วมกัน ก็สามารถแบ่งกันใช้งานระบบเครือข่ายได้แล้ว

แต่เนื่องจากเป็นการแบ่งกันใช้สายส่งข้อมูลร่วมกัน ดังนั้น เพื่อไม่ให้มีการชนกันของข้อมูล (Collision) เกิดขึ้น ข้อตกลงสำหรับการสื่อสารข้อมูลเช่น CSMA/CD (Carrier Sense Multiple Access with Collision Detection) และ Token Passing จึงได้รับการพัฒนาขึ้นมา เพื่อให้มั่นใจได้ว่า ในช่วงเวลาหนึ่งๆ จะมีเพียงเครื่องเดียวเท่านั้นที่ได้รับอนุญาต หรือมีโอกาสได้ใช้งานสายส่งข้อมูลเส้นนั้นๆ



รูปที่ 4-9 ระบบเครือข่ายท้องถิ่นแบบร่วมกัน (Shared LAN)

แม้ว่าในเวลาต่อมาข้อตกลงสำหรับการสื่อสารข้อมูลจะไม่ถูกเปลี่ยนแปลง แต่ลักษณะทางกายภาพของระบบเครือข่ายท้องถิ่นก็ได้เปลี่ยนแปลงไป จากที่แต่เดิมเครื่องๆ หนึ่งจะต่อเชื่อมเข้ากับสายส่งข้อมูลเส้นเพียง 1 เส้น (Single Media) ที่มีความยาวคงที่ (Fixed Length Cable) ก็ค่อยๆ พัฒนาขึ้นมาเป็นการเชื่อมต่อแบบกระจายผ่านอุปกรณ์รวมสื่อ (Star Wired Hubs) มากขึ้น นอกจากนี้การใช้งานอุปกรณ์รวมสื่อในระบบเครือข่ายร่วมแบบท้องถิ่น (Shared LAN Hub) ก็มีจุดเด่นในเรื่องของค่าใช้จ่าย, ความยืดหยุ่นของสื่อ (Media) ที่จะนำมาใช้ ซึ่งมีให้เลือกมากขึ้น รวมไปถึงความสะดวกในการบริหารระบบเครือข่ายอีกด้วย เพราะแต่ละเครื่องสามารถเชื่อมต่อเข้าสู่เข้าสู่ระบบเครือข่ายได้อย่างง่ายดายผ่านทางช่องทางการเชื่อมต่อ (Port) ต่างๆ ที่มีอยู่ภายในตัวอุปกรณ์รวมสื่อ (Hub) เอง การจัดการที่ง่ายดายขึ้นนี้ ได้รับการพัฒนาขึ้นมาจากหลายๆ องค์ประกอบควบคู่กันไป อันได้แก่สมรรถนะของเครื่องที่สูงขึ้น, เครื่องมือการจัดการเครือข่ายที่ดีขึ้น, การเติบโตของเทคโนโลยีทางด้านโปรแกรมประยุกต์ที่เกี่ยวข้องกับการสื่อสารระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย (Client/Server Applications) จนกระทั่งได้ก้าวเข้าสู่การเป็นระบบเครือข่ายแบบหลายข้อกำหนด (Multiprotocol Networking Environments) ที่สามารถรองรับการทำงานร่วมกับข้อกำหนดของการสื่อสารข้อมูลที่มีอยู่หลากหลายชนิดไปพร้อมๆ กันได้ บนระบบเครือข่ายเดียวดังเช่นที่เป็นอยู่ในปัจจุบัน ซึ่งผลักดันให้เทคโนลีระบบเครือข่ายท้องถิ่นเติบโตขึ้นอย่างรวดเร็ว แต่อย่างไรก็ตาม ในบางกรณี การแย่งกันใช้ทรัพยากรบนระบบเครือข่ายท้องถิ่น ก็กลายเป็นปัญหาขึ้นมาได้ ถ้าหากมีเครื่องลูกข่าย (Client) เครื่องใดพยายามโหมเข้าใช้เครื่องแม่ข่าย (Server) อย่างหนักเป็นระยะเวลาหนึ่ง จะทำให้ขีดความสามารถของระบบเครือข่าย (Bandwidth) ถูกใช้ไปจนหมด เครื่องลูกข่ายตัวอื่นๆ จะต้องหยุดรอ เป็นผลให้ปริมาณการใช้งานระบบเครือข่าย (Network Utilization) และปริมาณงานที่สามารถทำได้ต่อหน่วยเวลา (Application Throughput) ของระบบโดยรวม ลดต่ำลงไปมาก ยิ่งไปกว่านั้น หากระบบเครือข่ายมีขนาดใหญ่ ปริมาณการสื่อสารข้อมูลและจำนวนผู้ใช้งานก็จะมีมากขึ้น ปัญหาก็จะยิ่งทวีความรุนแรงยิ่งขึ้นไปอีก จึงเป็นการดีกว่าถ้าจะแบ่งระบบเครือข่ายท้องถิ่นขนาดใหญ่ออกเป็นระบบเครือข่ายย่อยที่มีปริมาณเครื่องลูกข่ายน้อยลง เพราะจะทำให้เครื่องลูกข่ายแต่ละเครื่องสามารถทำงานได้ที่ขีดความสามารถสูงสุดของระบบเครือข่ายจะให้ได้ เช่น 100 Mbps ในระบบ Fast Ethernet และสามารถส่งข้อมูลโดยไม่ต้องรอให้ Client เครื่องอื่นทำงานเสร็จก่อน นอกจากนี้การขยายประสิทธิภาพของการสื่อสารจากแบบสองทางครึ่งอัตรา (Half-Duplex) ไปเป็นการสื่อสารแบบสองทางเต็มอัตรา (Full-Duplex) ยังทำให้การรับและส่งข้อมูลสามารถทำพร้อมกันได้อย่างเต็มขีดความสามรถของระบบเครือข่ายอีกด้วย



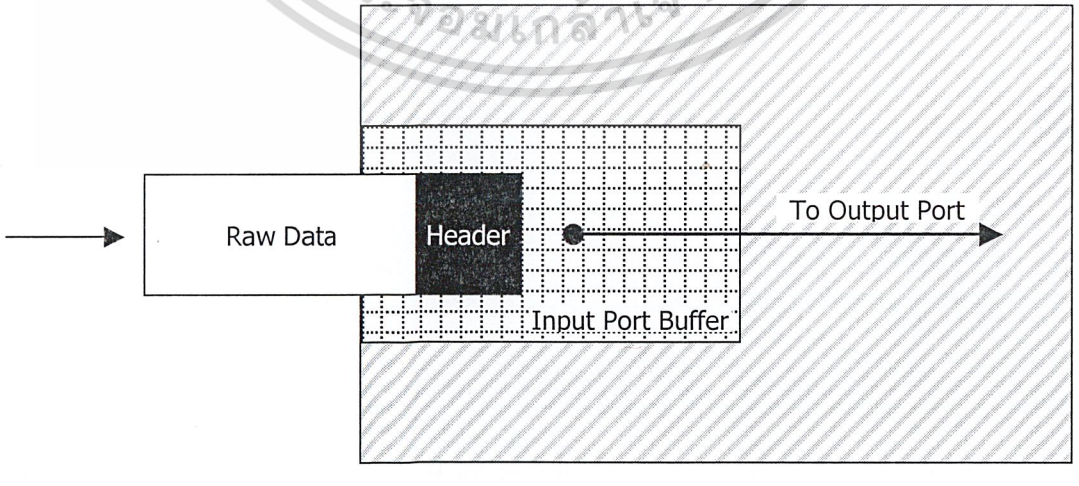
รูปที่ 4-10 ระบบเครือข่ายท้องถิ่นย่อย

แนวคิดของการสื่อสารแบบสองทางเต็มอัตรานี้ จะเป็นประโยชน์อย่างยิ่งต่อเครื่องแม่ข่ายที่ให้บริการกับเครื่องลูกข่ายหลายๆ เครื่องพร้อมกัน ส่วนแนวคิดของการแบ่งระบบเครือข่ายท้องถิ่นออกเป็นกลุ่มย่อยๆ หรือที่เรียกว่า Microsegmentation นั้น แต่ละ Microsegment จะต่อเข้ากับช่องทางการเชื่อมต่อต่างๆ ที่มีอยู่ในตัวอุปกรณ์สวิตช์ของระบบเครือข่ายท้องถิ่น (LAN Switch) ทำให้แต่ละช่องทางสามรถรองรับการสื่อสารแบบสองทางเต็มอัตราได้ ด้วยเหตุนี้จึงสามารถมองตัวอุปกรณ์สวิตช์ของระบบเครือข่ายท้องถิ่น เป็นเหมือนอุปกรณ์บริดจ์ที่มีหลายช่องทางการเชื่อมต่อ (Multiport Bridge) ได้ และหากนำเอา LAN Switch ไปเชื่อมต่อเข้ากับสายส่งสัญญาณความเร็วสูง (High-Speed Trunks) ก็จะสามารถทำเป็นแกนหลักของการสื่อสารหลักความเร็วสูง (High-Speed Backbone) ระหว่างกลุ่มเครือข่ายได้

กระบวนการส่งต่อชุดข้อมูลจากช่องทางการสื่อสารขาเข้าไปยังช่องทางการสื่อสารขาออกนั้น จำเป็นจะต้องใช้ใช้ข้อมูลที่ระบุไว้ที่ตัวชุดข้อมูลเองไม่ว่าจะเป็น ค่าตำแหน่งเครื่องปลายทาง, ค่าทิศทางจากเครื่องต้นทาง, หรือค่าระบุการช่องทางเชื่อมต่อ แต่อย่างไรก็ตามอุปกรณ์สวิตช์บนระบบเครือข่ายท้องถิ่นโดยทั่วไป ก็ยังจะส่งผ่านชุดข้อมูลโดยอ้างอิงรูปแบบของค่าตำแหน่งเครื่องปลายทางเป็นหลัก นอกจากนี้อุปกรณ์สวิตช์บนระบบเครือข่ายท้องถิ่นยังดำเนินการส่งผ่านชุดข้อมูลภายในตัวเอง โดยใช้กลวิธีที่เรียกว่า Internal-Forwarding ซึ่งมีอยู่ 2 รูปแบบคือ Cut Through Forwarding และ Store and Forward

4.4.1 กลวิธีแบบส่งต่อทันที (Cut-Through Forwarding)

ในแบบส่งต่อทันทีนั้น สวิตช์จะเริ่มต้นส่งต่อชุดข้อมูลทันทีที่สามารถตรวจสอบและยืนยันค่าตำแหน่งเครื่องปลายทางได้ การส่งต่อชุดข้อมูลในส่วนต้นๆ นั้น สามารถทำได้เลย แม้ว่าส่วนที่เหลือของชุดข้อมูล จะยังเข้ามาในหน่วยความจำชั่วคราว (Buffer) ของช่องทางการสื่อสารขาเข้าไม่ครบก็ตาม ซึ่งทำให้เกิดข้อได้เปรียบในเรื่องการหน่วงเวลาของการส่งผ่านข้อมูล (Latency) เพราะแต่ละชุดข้อมูลที่ผ่านเข้ามาจะไม่ถูกหน่วงเอาไว้ก่อนที่จะส่งต่อออกไป ส่วน ข้อเสียของกลวิธีนี้ก็คือ ชุดข้อมูลทั้งหมดจะถูกส่งผ่านออกไปยังช่องทางการสื่อสารขาออกโดยไม่ว่าจะเสียหรือไม่ก็ตาม

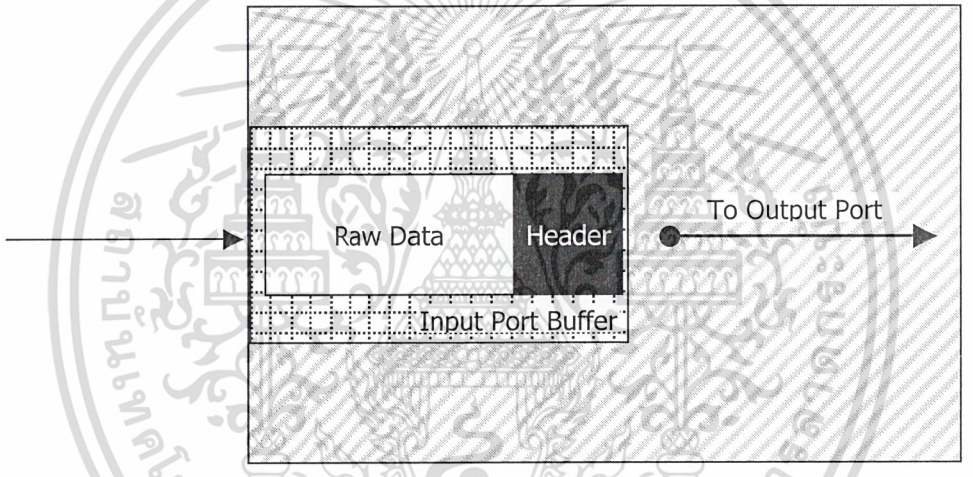


รูปที่ 4-11 การส่งต่อชุดข้อมูลแบบส่งต่อทันที (Cut-Through Forwarding)

4.4.2 กลวิธีแบบเก็บพักข้อมูล (Store and Forward)

กลวิธีแบบเก็บพักข้อมูลเอาไว้ก่อนนั้น อุปกรณ์สวิตช์บนระบบเครือข่ายจะต้องรอให้ชุดข้อมูลเข้าสู่หน่วยความจำชั่วคราวของช่องทางการสื่อสารขาเข้าให้ครบเสียก่อน แล้วจึงจะส่งต่อออกไปยังช่องทางการสื่อสารขาออกได้ ซึ่งจะทำให้สวิตช์สามารถตรวจสอบความยาวของเฟรมข้อมูล และตรวจสอบความถูกต้องของชุดข้อมูลโดย CRC (Cyclic Redundancy Check) ได้ ถ้าหากเฟรมข้อมูลมีขนาดใหญ่หรือเล็กเกินไป หรือตรวจสอบ CRC แล้วพบข้อผิดพลาด สวิตช์ก็จะกำจัดชุดข้อมูลนั้นๆ ทิ้งไป โดยการไม่ส่งต่อชุดข้อมูลนั้นๆ ออกไป แต่อย่างไรก็ตาม ทุกจุดที่มีการตรวจสอบชุดข้อมูลนั้น จะต้องมีการหน่วงเวลาของการสื่อสารข้อมูลเกิดขึ้นตามมา ซึ่งเวลาที่เสียไปนั้นขึ้นอยู่กับความยาวของเฟรมข้อมูลแต่ละเฟรมนั่นเอง

กลวิธีนี้นิยมใช้ในอุปกรณ์สวิตช์บนระบบเครือข่าย ที่จะต้องทำหน้าที่ส่งผ่านชุดข้อมูลจากเครือข่ายที่ใช้สื่อชนิดหนึ่งไปยังเครือข่ายที่ใช้สื่ออีกชนิดหนึ่ง เช่นอีเธอร์เน็ทกับเอทีเอ็ม



รูปที่ 4-12 การส่งต่อชุดข้อมูลแบบเก็บพักข้อมูล Store and Forward

บทที่ 5
หลักการพื้นฐานของเทคโนโลยีไอพีสวิตช์

5.1 แนวคิดพื้นฐานของไอพีสวิตช์

ไอพีสวิตช์ (IP Switch) คือรูปแบบหนึ่งของกลุ่มข้อกำหนดทางการสื่อสารข้อมูล (Protocols) และกรรมวิธี (Mechanism) ต่างๆ ที่พยายามนำเอากลไกการสวิตช์ในระดับดาตาลิงค์ (Layer-2 Switching) เข้ามาใช้เพื่อปรับปรุงหรือเร่งความเร็วในการส่งผ่านชุดข้อมูลในรูปแบบไอพี (IP Packet Forwarding) ผ่านระบบเครือข่ายคอมพิวเตอร์ที่มีอยู่เดิม ให้มีประสิทธิภาพสูงขึ้น โดยทำให้เกิดผลกระทบต่อโครงสร้างการเชื่อมต่อทางกายภาพ (Topology) เดิมของระบบเครือข่ายนั้นๆ ให้น้อยที่สุด เท่าที่จะเป็นไปได้

ไอพีสวิตช์นั้น ได้รับการคิดค้นพัฒนาต่อเนื่องมาจากแนวคิดของ IPOA หรือ IP Over ATM ซึ่งนำเอาการสื่อสารข้อมูลในรูปแบบของไอพีไปใช้งานบนเครือข่ายที่มีโครงสร้างการเชื่อมต่อทางกายภาพแบบเอทีเอ็ม แต่ไอพีสวิตช์ในปัจจุบันสามารถใช้ร่วมกับเทคโนโลยีการสวิตช์ใดๆ ในระดับดาตาลิงค์ก็ได้ เช่น Frame Relay แต่อย่างไรก็ตามกรรมวิธีที่ได้รับความนิยมสูงสุดก็ยังคงเป็นระบบ ATM ตามแบบไอพีสวิตช์ดั้งเดิมที่ได้รับการพัฒนาขึ้นมาโดยบริษัท Ipsilon นั่นเอง

5.2 นิยามและศัพท์บัญญัติ

5.2.1 ไอพีสวิตช์

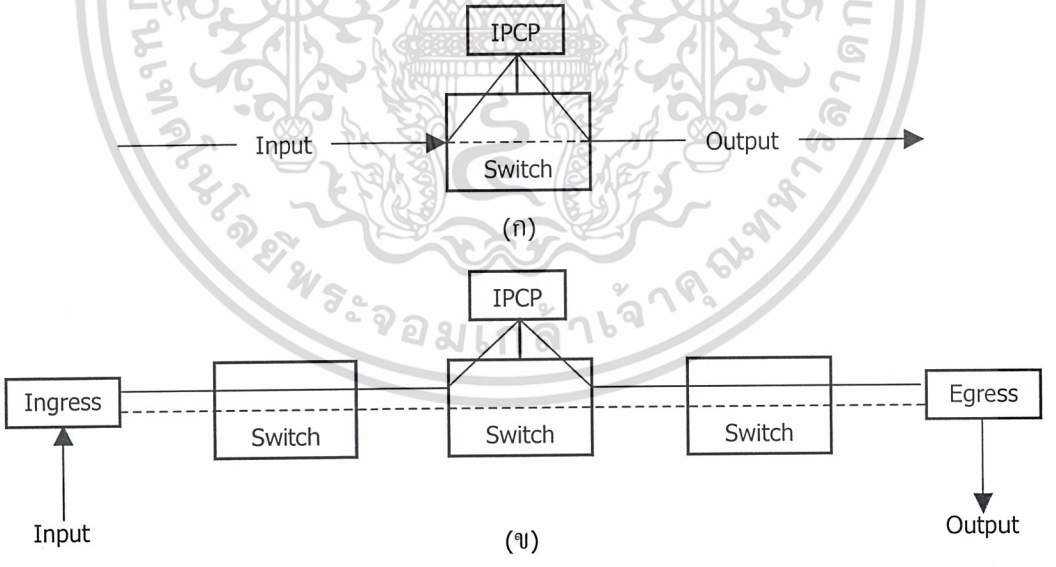
ไอพีสวิตช์ คือกลุ่มข้อกำหนดของการสื่อสารข้อมูล และกรรมวิธีที่นำเอากลไกการสวิตช์ความเร็วสูงในระดับดาต้าลิงค์เข้ามาช่วยในการส่งผ่านข้อมูลในรูปแบบไอพีผ่านระบบเครือข่ายคอมพิวเตอร์ ไอพีสวิตช์อาศัยข้อได้เปรียบในเรื่องความกว้างของช่องสัญญาณ (Bandwidth) ที่สูงประกอบกับระดับการหน่วงเวลา (Latency Time) ที่ต่ำของเทคโนโลยีสวิตชิ่งมาใช้ เพื่อให้สามารถส่งชุดข้อมูลไปในระบบเครือข่ายได้เร็วที่สุดเท่าที่จะเป็นไปได้ โดยพยายามควบคุมทิศทางการไหล (Redirect) ของชุดข้อมูลบางชุดหรือทั้งหมด ให้ผ่านไปทางส่วนประกอบของระบบสวิตชิ่ง (Switch Components) ต่างๆ โดยตรง ซึ่งเป็นการพยายาม ขจัดความต้องการการประมวลผลในระดับเน็ตเวิร์คออกไป เพื่อลดความสิ้นเปลือง (Overhead) ในการสื่อสารข้อมูลลงให้ได้มากที่สุดนั่นเอง

ไอพีสวิตช์ ในทางปฏิบัติจึงหมายถึง อุปกรณ์หรือระบบใดๆ ที่สามารถส่งผ่านชุดข้อมูลในรูปแบบของการสื่อสารระบบไอพีไปในเครื่อข่ายระดับเน็ตเวิร์ค ร่วมกับการครอบครองและสามารถควบคุมการทำงานของส่วนประกอบสวิตชิ่งในระดับดาต้าลิงค์ได้ เพื่อให้ชุดข้อมูลในรูปแบบไอพี สามารถถูกสวิตช์ออกไปในระดับดาตาลิงค์ได้โดยตรง ซึ่งทั้งนี้ ไอพีสวิตช์จะต้องมีกลไกบางอย่างที่สามารถใช้ระบุได้ว่าชุดข้อมูลชุดใดควรถูกส่งออกไปในระดับเน็ตเวิร์ค (Layer-3 Forwarding) และชุดข้อมูลชุดใดที่ควรจะถูกส่งออกไปในระดับดาตาลิงค์ (Layer-2 Switching) พร้อมกันนี้ไอพีสวิตช์ยังจะต้องมีกรรมวิธีบางอย่างที่สามารถเปลี่ยนแปลงทิศทางการไหลของชุดข้อมูลบางส่วนหรือทั้งหมด จากที่เคยผ่านเส้นทางที่ค้นหาได้ (Routed Path) ในระดับเน็ตเวิร์ค ให้ไปใช้เส้นทางของระบบสวิตชิ่ง (Switched Path) ในระดับดาตาลิงค์แทนอีกด้วย

ระบบไอพีสวิตช์สามารถมองได้เป็น 2 ลักษณะ คือ IP Switch Device กับ Virtual IP Switch โดยอาศัยขอบเขตของเส้นทางในระดับดาตาลิงค์ในการพิจารณา

- อุปกรณ์ไอพีสวิตช์ (IP Switch Device)

เส้นทางของระบบสวิตช์แบบ IP Switch Device จะถูกพิจารณาให้อยู่ภายในอุปกรณ์เพียงตัวเดียว ส่วนประกอบต่างๆ ของสวิตช์จะอยู่ภายใต้การควบคุมโดยตรงจากส่วนควบคุม (IPCP) เดียวเท่านั้น ซึ่งส่วนควบคุม IPCP หรือ IP Control Point นั้นก็คือส่วนการทำงานที่ดำเนินการด้วยรูปแบบของการจัดการเส้นทาง (Routing Protocol) แบบทั่วๆไป เช่น RIP, OSPF, BGP เป็นต้น โดย IPCP จะจัดการเกี่ยวกับเส้นทางตั้งต้นแต่ละจุดในระดับเน็ตเวิร์ค (Default Layer-3 Hop-by-Hop Routed Path) รวมทั้งการควบคุมทิศทางการไหลของชุดข้อมูลให้เดินทางผ่านทางส่วนประกอบต่างๆ ของสวิตช์ ทั้งโดยทางตรงและทางอ้อม ซึ่งส่วนประกอบ ของสวิตช์แต่ละตัวนั้นจะจัดสร้างตารางการเชื่อมต่อ (Connection Table) ของตนเองขึ้นมาโดยจะประกอบไปด้วยข้อมูลเกี่ยวกับ ช่องทางการสื่อสารขาเข้า (Input Ports), หมายเลขช่องทางขาเข้า (Input Labels หรือ VPI/VCI), ช่องทางการสื่อสารขาออก (Output Ports), และ หมายเลขช่องทางขาออก (Output Labels) และเนื่องจากช่องทางการสื่อสารขาเข้าและขาออกของ IP Switch Device จะอยู่บนอุปกรณ์ตัวเดียวกัน ส่วนควบคุมของ IP Switch Device แต่ละตัวจึงเป็นอิสระต่อกัน IP Switch หลายๆ ตัวที่อยู่ติดกันจะจัดสร้างเส้นทางการสวิตช์แบบจุดต่อจุด (End-to-End Switch Path) ระหว่างกันขึ้นมาผ่านกลุ่มของอุปกรณ์ IP Switch Device ด้วยกันเอง



รูปที่ 5-1 อุปกรณ์ไอพีสวิตช์ และไอพีสวิตช์แบบเสมือน

- ไอพีสวิตช์เสมือน (Virtual IP Switch)

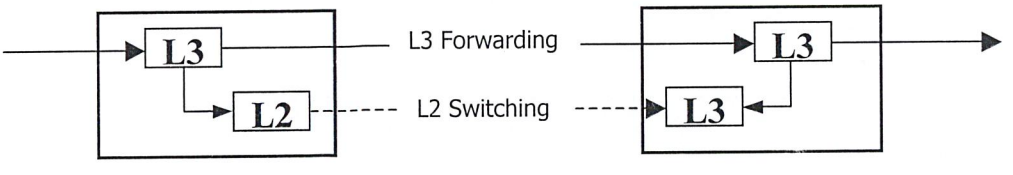
เส้นทางการสวิตช์แบบจุดต่อจุดของ Virtual IP Switch นั้นอาจครอบคลุมอุปกรณ์ไอพีสวิตช์หลายตัวด้วยกัน ซึ่งแต่ละตัวจะทำงานอยู่ภายใต้การควบคุมโดยอ้อมจาก IPCP ของอุปกรณ์ไอพีสวิตช์ตัวใดตัวหนึ่งเพียงตัวเดียวเท่านั้น ช่องทางการสื่อสารขาเข้าและขาออกของ Virtual IP Switch นั้นอาจอยู่บนอุปกรณ์เดียวกันหรือคนละตัวกันก็ได้

5.2.2 อินเกรสและอีเกรส (Ingress and Egress)

คือส่วนประกอบของสวิตช์ตัวที่อยู่ปลายสุดของการสื่อสาร (Edge Device) ในระบบเครือข่ายไอพีสวิตช์ โดยอินเกรสจะหมายถึงส่วนประกอบของสวิตช์ที่ทำหน้าที่รับปรับเปลี่ยนรูปแบบการไหลของชุดข้อมูล จากที่เดิมใช้รูปแบบเส้นทางในระดับเน็ตเวิร์คให้กลายไปเป็นรูปแบบเส้นทางในระดับดาต้าลิงค์แทน เมื่อใดก็ตามที่สามารถกระทำได้ เพื่อเพิ่มความเร็วให้กับการบติดต่อรับส่งข้อมูลระหว่างไอพีสวิตช์ด้วยกันเอง ในระหว่างเส้นทางที่ชุดข้อมูลเหล่านั้นจะต้องผ่านไป ส่วนอีเกรสจะทำงานในลักษณะตรงกันข้าม คือจะปรับเปลี่ยนรูปแบบของชุดข้อมูลที่กลายมาเป็นรูปแบบการสื่อสารในระดับดาต้าลิงค์ให้กลับไปเป็นรูปแบบการสื่อสารในระดับเน็ตเวิร์คเหมือนเดิม เพื่อให้ชุดข้อมูลสามารถถูกส่งต่อไปยังเครื่องที่เป็นจุดหมายปลายทางได้อย่างถูกต้อง

ทั้งอินเกรสและอีเกรสจะต้องประกอบไปด้วยชุดคำสั่งพิเศษสำหรับดำเนินงานบนสถานีงาน (Workstation) ต่างๆ เพื่อเพิ่มความสามารถให้แก่สวิตช์แต่ละตัวที่อยู่ปลายสุดของการสื่อสารนั่นเอง

- รูปแบบการทำงานของอินเกรสกับอีเกรสจะประกอบไปด้วยการทำงานดังต่อไปนี้
  - จัดหาช่องทางการสื่อสารเริ่มต้น (Normal Default IP Forwarding) สำหรับแต่ละการเชื่อมต่อที่เกิดขึ้น ไม่ว่าจะเป็นการส่งข้อมูลเข้าสู่เครือข่ายไอพีสวิตช์หรือออกจากเครือข่ายไอพีสวิตช์
  - จัดการเกี่ยวกับการแปลงรูปแบบชุดข้อมูล (Media Translation) ให้เหมาะสมกับสื่อที่ใช้ เช่น การแปลงชุดข้อมูลระหว่างรูปแบบอีเธอร์เนต (Ethernet) กับ เอทีเอ็ม (ATM) เป็นต้น
  - ดำเนินการสร้าง / รักษา / และถอดถอนเส้นทางการสื่อสารข้อมูลในระดับดาตาลิงค์ ร่วมกับอุปกรณ์ไอพีสวิตช์อื่นๆ ที่อยู่ระหว่างอินเกรสกับอีเกรสที่เหมาะสมของแต่ละการเชื่อมต่อ
  - ในส่วนของอินเกรส จะตรวจสอบและคัดแยกชุดข้อมูล แล้วส่งผ่านชุดข้อมูลที่เหมาะสมไปบนเส้นทางการสื่อสารข้อมูลในระดับดาต้าลิงค์ ซึ่งทำได้โดยการตรวจสอบเขตข้อมูล (Field) บางจุดที่อยู่ในส่วนต้นของชุดข้อมูล (Packet Header) เพื่อพิจารณาว่าควรจะส่งชุดข้อมูลชุดนั้นไปบนเส้นทางการสื่อสารระดับดาต้าลิงค์หรือไม่, ส่งไปในเส้นทางใด, ค้นหาตารางบันทึกเส้นทางมาตรฐาน (Standard Routing Table), แทรกหมายเลขช่องทางการสื่อสาร (Label) ที่จะใช้ เข้าไปในชุดข้อมูลแต่ละชุด
  - ในระบบ ATM จะต้องทำการแบ่งย่อยชุดข้อมูล (Segmentation) ให้มีขนาดเล็กลง แล้วส่งผ่านช่องทางเสมือน (Virtual Connection หรือ VC) ออกไป
  - ในส่วนของอีเกรส จะรับชุดข้อมูลเข้ามาทางช่องทางการสื่อสารระดับดาตาลิงค์ แล้วทำการส่งผ่านชุดข้อมูลในรูปแบบไอพี (IP Forwarding) ตามแบบมาตรฐาน ไปยังเครื่องปลายทาง



รูปที่ 5-2 ฟังก์ชั่น Ingress และ Egress ของไอพีสวิตช์

แนวคิดหลักของอินเกรส-อีเกรส ก็คือการนำส่วนการทำงานเสริม (Complementary Function) ไปเพิ่มไว้ในส่วนของอินเกรส เพื่อรับชุดข้อมูลที่เหมาะสมเข้าสู่ระบบไอพีสวิตช์ แล้วเลือกว่าจะส่งต่อชุดข้อมูลนั้นๆ ไปบนเส้นทางการเชื่อมต่อที่ระดับใด โดยอ้างอิงตามบรรทัดฐานของไอพี (IP-Layer Criteria) บางประการ เช่นตำแหน่งไอพีของเครื่องต้นทาง / ปลายทาง (Source / Destination IP Address) หรือส่วนต้นของค่าหมายเลขเครือข่ายปลายทาง (Destination Network Prefix) เป็นต้น จากนั้นชุดข้อมูลชุดใดที่สามารถจำแนกหรือระบุตำแหน่งในระดับดาต้าลิงค์ (MAC Address) ได้ ก็จะถูกส่งผ่านเข้าไปในระบบเครือข่ายไอพีสวิตชิ่ง และใช้การจัดการเส้นทางในระดับดาตาลิงค์ ส่วนชุดข้อมูลที่เหลือก็จะถูกส่งผ่านระบบเครือข่ายไอพีสวิตช์ในระดับเน็ตเวิร์ค โดยใช้การจัดการเส้นทางด้วยค่าตำแหน่งไอพีตามปกติ เช่นเดียวกันกับที่ใช้ในอุปกรณ์เราเตอร์ทั่วๆ ไป นั่นเอง

5.2.3 เส้นทางลัด (Shortcut Path)

ในระบบที่เป็นไอพีเราเตอร์ทั่วไปนั้น ชุดข้อมูลจะถูกส่งผ่านจากต้นทางไปยังปลายทางทีละจุดๆ (Hop-by-Hop) ซึ่งในแต่ละ Hop นั้นจะต้องมีขั้นตอนการดำเนินงานต่างๆ เกิดขึ้นหลายอย่างด้วยกัน ได้แก่การเลือกหาเส้นทางในตาราง (Routing Table Lookup), การตรวจสอบความถูกต้องในส่วนต้นของชุดข้อมูล (Header Checksum), การลดค่าเวลาที่เหลืออยู่ของชุดข้อมูล (TTL Decrement), การแปลงรูปแบบชุดข้อมูล (Media Translation), และอื่นๆ ซึ่งกระบวนการต่างๆเหล่านี้ทำให้เกิดการสูญเสียเวลาในการส่งผ่านข้อมูล (Delay) และเกิดการหน่วงของกระแสข้อมูล (Latency) ขึ้นตามมาอีกมากมาย วิธีการหนึ่งซึ่งสามารถลดปัญหาเหล่านี้ลงไปได้ คือการสร้างเส้นทางลัดผ่านกระบวนการ Routing Hop ในรายทางลงให้มากที่สุดเท่าที่จะเป็นไปได้ ซึ่งก็คือการทำ Shortcut Path หรือ Layer-2 Switched Path นั่นเอง

โดยนิยามแล้วเส้นทางลัด หรือ Shortcut Path นี้ หมายถึงการสร้างช่องทางการสื่อสารจำลองในระดับดาตาลิงค์ (Virtual Connection) ระหว่างเครื่องต้นทางกับเครื่องปลายทางขึ้นมา โดยลัดผ่านการทำงานของ Layer-3 Routing Hop ระหว่างทางไป ซึ่ง Shortcut Path สามารถสร้างขึ้นระหว่าง เครื่องต้นทางกับเครื่องปลายทาง (Host-to-Host), ระหว่างอุปกรณ์สวิตชิ่งที่ปลายสุดของการสื่อสาร (Edge Device) ด้วยกันเอง, หรือผสมกันก็ได้ ขึ้นอยู่กับการออกแบบและการจัดวางอินเกรสกับอีเกรสในระบบ

- คุณสมบัติพื้นฐานต่างๆของ Shortcut Path
  - ลัดขั้นตอนการดำเนินงานต่างๆ ของกระบวนการจัดการเส้นทางในระดับเน็ตเวิร์ค
  - เส้นทางลัดจะถูกสร้างขึ้นมาได้จาก 2 กรณีด้วยกันคือการสื่อสารข้อมูลทั่วไป (Data Traffic) กับการสื่อสารการควบคุม (Control Traffic) ในกรณีของเส้นทางลัดที่ถูกสร้างขึ้นเพื่อการสื่อสารข้อมูลทั่วไป (Data Driven Shortcut Path) จะเกิดขึ้นได้ก็ต่อเมื่อชุดข้อมูลนั้นถูกกำหนดเส้นทาง (Routed) ในระดับเน็ตเวิร์คเป็นที่เรียบร้อยแล้ว อุปกรณ์ไอพีสวิตช์ก็จะจัดสร้างเส้นทางลัดในระดับดาต้าลิงค์ขึ้นตามมา ส่วนในกรณีของการสื่อสารข้อมูลเพื่อการควบคุม (Control Driven Shortcut Path) จะถูกสร้างขึ้นมาก็ต่อเมื่อมีความต้องการเกิดขึ้นเท่านั้น
  - เมื่อใดก็ตามที่ไม่มีเส้นทางลัดระหว่างอินเกรสกับอีเกรส หรือเส้นทางลัดถูกยกเลิกไป การสื่อสารข้อมูลก็จะยังคงมีอยู่ต่อไปได้ โดยอาศัยเส้นทางตามการเชื่อมต่อระดับเน็ตเวิร์คแทน

- ในระบบไอพีสวิตช์ (IP Switch Device) นั้น เส้นทางลัดที่ถูกสร้างขึ้น จะสอดคล้องกับโครงสร้างการเชื่อมต่อทางกายภาพที่ได้จากการจัดการเส้นทางในระดับเน็ตเวิร์ค แต่สำหรับระบบไอพีสวิตช์จำลอง (Virtual IP Switch) นั้น รูปแบบโครงสร้างการสื่อสารของเส้นทางลัดที่ถูกสร้างขึ้น จะเป็นอิสระจากโครงสร้างการสื่อสารที่ได้จากการจัดการเส้นทางในระดับเน็ตเวิร์ค
- เส้นทางลัดระหว่างอินเกรสกับอีเกรส (Ingress-to-Egress Shortcut Path) สามารถสร้างขึ้นระหว่างไอพีสวิตช์แต่ละ คู่ที่ชุดข้อมูลจะต้องเดินทางผ่านไป หรือจะสร้างโดยการรวมเอาเส้นทางลัดจากหลายๆ ส่วนที่มีอยู่แล้ว เข้าด้วยกันก็ได้
- เส้นทางลัดสามารถมีต้นกำเนิดมาจากเครือข่ายที่เป็นอินเกรส, อีเกรส หรือจุดใดๆ ก็ได้ ขึ้นกับความพร้อมและการดำเนินงานร่วมกันระหว่างกลุ่มไอพีสวิตช์ในแต่ละส่วนของเส้นทาง
- เส้นทางลัดสามารถสร้างเป็นแบบจุดต่อจุด (Point-to-Point), จุดเดียวต่อหลายจุด (Point-to-Multipoint), หรือ หลายจุดต่อจุดเดียว (Multi Point-to-Point) ก็ได้
- การใช้เส้นทางลัดสามารถช่วยเพิ่มพูนประสิทธิภาพทางด้าน QoS และ CoS ให้ดีขึ้นกว่าที่การเส้นทางที่ค้นหาได้ในระดับเน็ตเวิร์คแต่เพียงอย่างเดียว

5.3 ปัจจัยที่กระตุ้นให้เกิดการพัฒนา

➢ เหตุผลสำคัญ ที่ทำให้เกิดการพัฒนาเทคโนโลยีไอพีสวิตช์ขึ้นมา มีดังต่อไปนี้

- ในทางปฏิบัติกับระบบที่ใช้งานอยู่จริงนั้น การใช้งานเราเตอร์จะทำให้เกิดปัญหาคอขวดในการส่งถ่ายข้อมูลเกิดขึ้นในแต่ละจุด ทำให้ไม่สามารถรองรับการสื่อสารข้อมูลขนาดใหญ่ที่มีอัตราการส่งผ่านข้อมูลเกินระดับ OC3 (155.52 Mbps) ได้ ในขณะที่ระบบเอทีเอ็มสามารถส่งผ่านข้อมูลได้ถึงระดับ OC12 หรือ 622.08 Mbps เลยทีเดียว
- ตารางการจัดการเส้นทางในระดับเน็ตเวิร์คจะมีขนาดใหญ่มากเนื่องจากจะต้องเก็บค่าเป็นตำแหน่งไอพีซึ่งมีความยาวหลายหลัก ทำให้การมองหาข้อมูลในตารางต้องใช้เวลานาน การปรับปรุงนั้นสามารถทำได้โดยการรวมเอากลุ่มของเราเตอร์หลายๆ ตัว เข้าด้วยกันเป็นกลุ่มของเลขหมายย่อย (Label) ที่มีจำนวนหลักน้อยค่าหนึ่ง หลายๆ กลุ่ม (เช่น การทำ VPI/VCI) และการมองหาข้อมูลในตารางเส้นทางจะทำโดยการใช้ดรรชนี (Indexing) ในตารางการเชื่อมต่อ (VPI/VCI Table) ที่ทำงานอยู่บนอุปกรณ์ทางกายภาพ (Hardware) ล้วนๆ ซึ่งจะทำให้เกิดความคล่องตัว และความเร็วในการทำงานที่สูงกว่าเดิมมาก
- โปรแกรมประยุกต์ต่างๆ ที่ทำงานอยู่บนระบบเครือข่ายที่ใช้ที่ใช้รูปแบบการสื่อสารข้อมูลแบบไอพี มีแนวโน้มที่จะต้องการประสิทธิภาพและการจัดการช่องทางการสื่อสารที่ดีขึ้น

5.4 รูปแบบค่าตำแหน่ง (IP Switching Addressing Models)

ระบบเครือข่ายที่สามารถรองรับการทำไอพีสวิตชิ่งร่วมกับเทคโนโลยีเอทีเอ็มได้นั้น อย่างน้อยที่สุดจะต้องสามารถรองรับการทำงานในรูปแบบของค่าตำแหน่งที่อยู่ชนิดไอพี (IP Address Space) ได้ ซึ่งค่าตำแหน่งที่อยู่ชนิดไอพีนี้อาจผูกติดอยู่กับค่าตำแหน่งที่อยู่ชนิดเอทีเอ็มก็ได้ ในกรณีที่ชุดข้อมูลนั้นระบุตำแหน่งต้นทางหรือปลายทางเอาไว้อย่างชัดเจนแล้ว หรืออาจใช้การเชื่อมโยง (Map) ค่าตำแหน่งที่อยู่ชนิดไอพีที่พ่วงมาด้วยไปเป็นช่องทางการเชื่อมต่อเสมือน โดยอาศัยคุณสมบัติของการทำเลขหมาย (Label) ให้กับแต่ละการสื่อสารข้อมูลที่เกิดขึ้นด้วย VPI/VCI

อย่างไรก็ตาม ในระบบไอพีสวิตช์นั้น มีรูปแบบพื้นฐานสำหรับจัดการกับค่าตำแหน่งที่อยู่ของเครื่องบนระบบเครือข่ายอยู่ 2 ลักษณะด้วยกัน ดังต่อไปนี้

5.4.1 Separated Addressing Model

รูปแบบระบบเครือข่ายที่ใช้ค่าตำแหน่งที่อยู่ในลักษณะนี้ จะต้องสามารถใช้ค่าตำแหน่งที่อยู่ได้ทั้งชนิดไอพีและเอทีเอ็มได้ ดังนั้นไม่ว่าจะเป็นเราเตอร์หรือเครื่องคอมพิวเตอร์ใดๆ ก็ตาม ที่เชื่อมต่อเข้ากับเครือข่ายรูปแบบนี้ จะต้องได้รับการกำหนดให้รู้จักทั้งตำแหน่งที่อยู่ชนิดไอพีและชนิดเอทีเอ็มควบคู่กันไป ซึ่งไม่ต่างจากระบบเครือข่ายท้องถิ่น (LAN) ที่ได้รับการกำหนดให้ใช้ตำแหน่งที่อยู่ทั้งแบบไอพี (IP Address) และแบบแมค (MAC Address) ไปด้วยกันนั่นเอง

ในกรณีที่ยังไม่มีการจัดสร้างช่องทางการเชื่อมต่อเสมือนแบบถาวร (Permanent Virtual Connection หรือ PVC) ขึ้นมา เครื่องต้นทางก็จำเป็นจะต้องทราบตำแหน่งที่อยู่ชนิดเอทีเอ็มของเครื่องปลายทางที่ต้องการจะติดต่อด้วยก่อน ซึ่งในที่นี้จำเป็นจะต้องใช้กรรมวิธีของการแปลงค่าตำแหน่งที่อยู่จากชนิดไอพีไปเป็นเอทีเอ็ม (IP-to-ATM Address Resolution) ซึ่งกระทำได้โดย เครื่องต้นทางจะติดต่อร้องขอการสร้างช่องทางการเชื่อมต่อเสมือนแบบส่งสัญญาณ (Signaling Virtual Channel หรือ SVC) ไปยังเครื่องปลายทางที่ต้องการ โดยใช้ค่าตำแหน่งที่อยู่แบบไอพีในการติดต่อ

ในทางปฏิบัติแล้วจะมีการนำรูปแบบข้อตกลงการจัดการเส้นทางแบบแยกจากกัน (Separate Routing Protocols) มาใช้ เพื่อประกาศค่า (Advertise) เครือข่ายของเครื่องปลายทาง (Destination Prefixes) ทั้งในระบบไอพีและเอทีเอ็ม อย่างไรก็ตาม มีข้อตกลงสำหรับการจัดการเส้นทางบนระบบเครือข่าย (Routing Protocol) บางรูปแบบ เช่น I - PNNI (Integrated Private Network-to-Network Interface) ที่สามารถใช้งานได้กับทั้งค่าตำแหน่งในระบบไอพีและระบบเอทีเอ็ม

- ระบบเครือข่ายที่ใช้รูปแบบค่าตำแหน่งที่อยู่ของเครื่องในลักษณะแยกต่างหากจากกัน จะมีลักษณะการทำงานดังต่อไปนี้
  - จำเป็นจะต้องกำหนดค่าทั้งแบบไอพีและเอทีเอ็มให้กับทุกตำแหน่งบนระบบเครือข่าย ที่สามารถใช้รูปแบบไอพีได้ และต่ออยู่กับอุปกรณ์เอทีเอ็ม (IP-Addressable ATM- Attached)
  - มักรองรับการทำงานตามรูปแบบข้อตกลงการจัดการเส้นทางแบบแยกจากกัน ซึ่งก็คือ ทั้งในระบบไอพีและระบบเอทีเอ็มต่างก็มีรูปแบบการเชื่อมต่อเป็นของตนเอง และไม่เกี่ยวข้องกัน

- ในทางปฏิบัติแล้ว การแปลงค่าตำแหน่งที่อยู่จากชนิดไอพีไปเป็นเอทีเอ็มในแบบไม่ตายตัว (Dynamic Resolution) จะต้องมีกลไกการในการสอบถามไปยังเครื่องแม่ข่าย (Query-to-Server Mechanism) เพิ่มเติมเข้ามา ซึ่งจะทำให้สูญเสียเวลาไปบ้าง ในระยะเริ่มต้นของการติดต่อทั้งแบบปกติและแบบเส้นทางลัด

รูปแบบระบบเครือข่ายที่ใช้กันนี้ ถูกใช้ในระบบไอพีสวิตช์แบบซ้อนทับ (Overlay Model) ตัวอย่างระบบเครือข่ายไอพีสวิตช์ที่ใช้เทคนิคนี้ได้แก่ ระบบ MPOA (Multi-Protocol Over ATM)

5.4.2 IP-to-VC

เป็นลักษณะของระบบเครือข่ายสวิตชิ่ง ที่ใช้ค่าตำแหน่งที่อยู่ชนิดไอพีแต่เพียงอย่างเดียวเท่านั้น ชุดข้อมูลที่ถูกส่งผ่านเข้าสู่ระบบเครือข่ายจะอาศัยเส้นทางลัดที่แยกต่างหากออกจากกันเป็นช่องทางการเชื่อมต่อเสมือน ซึ่งขึ้นอยู่กับข้อมูลระดับเน็ตเวิร์คภายในส่วนต้นของชุดข้อมูลแต่ละชุด ระบบนี้จึงไม่จำเป็นต้องใช้ค่าตำแหน่งที่อยู่ของเครื่องในรูปแบบเอทีเอ็มแต่อย่างใด และใช้ข้อกำหนดการสื่อสารข้อมูลแบบแยกการควบคุม (Separate Control Protocols) แทนการจัดการกับค่าตำแหน่งแบบแยกจากกัน ในการเชื่อมโยงค่าตำแหน่งชนิดไอพีไปยังเส้นทางลัดที่เหมาะสม แม้ว่าระบบเครือข่ายในลักษณะนี้จะไม่ต้องอาศัยค่าตำแหน่งในรูปแบบเอทีเอ็ม หรือข้อกำหนดใดๆ ของเอทีเอ็ม (ATM Forum Protocols) เลยก็ตาม แต่กระบวนการเชื่อมโยงค่าตำแหน่งชนิดไอพีไปยังเส้นทางลัดนั้น ก็ยังคงใช้งาน ATM VPI/VCI ในด้านความสัมพันธ์ระหว่างข้อมูลในส่วนของระดับเน็ตเวิร์คของชุดข้อมูลกับช่องทางเสมือน

- ระบบเครือข่ายที่ใช้รูปแบบค่าตำแหน่งที่อยู่ของเครื่องในลักษณะแปลงจากระบบไอพีไปเป็นช่องทางเสมือน จะมีลักษณะการทำงานดังต่อไปนี้
  - ใช้ค่าตำแหน่งที่อยู่ของเครื่องชนิดไอพีเท่านั้น และไม่จำเป็นต้องมีการจัดการกับค่าตำแหน่งที่อยู่ชนิดเอทีเอ็มใดๆ อยู่เลย
  - ใช้ข้อกำหนดการจัดการเส้นทางแบบไอพีเท่านั้น ในการประกาศค่าเครือข่ายที่เครื่องปลายทางตั้งอยู่
  - ใช้ข้อกำหนดพิเศษ ในการเชื่อมโยงค่าที่อยู่ชนิดไอพีไปยังช่องทางเสมือนที่เป็นเส้นทางลัดที่ถูกต้อง

5.5 โครงแบบของไอพีสวิตช์ (IP Switching Model)

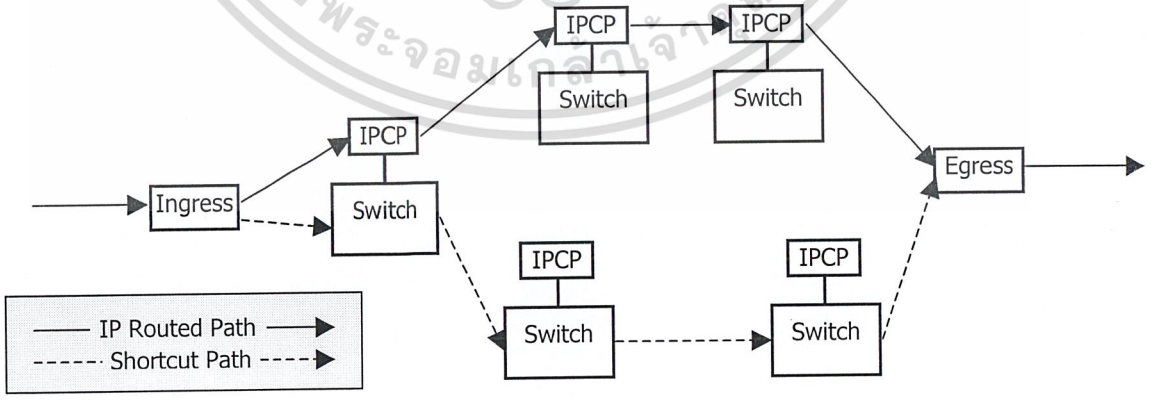
ระบบไอพีสวิตช์มีที่ใช้งานอยู่ทั่วไปในปัจจุบันมีโครงแบบพื้นฐานอยู่ 2 แบบด้วยกัน โดยพิจารณาจากการใช้งาน หรือไม่ใช้งานข้อกำหนดต่างๆ ของเอทีเอ็ม (ATM Forum Protocols) รวมทั้งพิจารณาองค์ประกอบของเครือข่ายด้วยว่าเป็นไอพีสวิตช์จำลอง หรือเป็นอุปกรณ์ไอพีสวิตช์

5.5.1 โครงแบบซ้อนทับ (Overlay Model)

เครือข่ายไอพีสวิตช์ในรูปแบบนี้ จะมีส่วนดำเนินงานแบบไอพีในระดับเน็ตเวิร์ค (IP-Layer) ทำงานอยู่เหนือส่วนการทำงานแบบเอทีเอ็มในระดับดาตาลิงค์ที่แยกการทำงานออกมาต่างหากอีกที ระบบ

จะต้องประกอบไปด้วยอุปกรณ์เครือข่ายชนิดไอพี (IP Device) ที่มีค่าตำแหน่งที่อยู่แบบไอพี (IP Address) เป็นของตัวเอง และใช้การจัดการเส้นทางแบบไอพี (IP Routing Protocols) กับอุปกรณ์สื่อสารเครือข่ายแบบเอทีเอ็ม (ATM Device) ใช้ค่าตำแหน่งแบบเอทีเอ็ม (ATM Address) และใช้การค้นหาเส้นทางแบบ เอทีเอ็ม (ATM Signaling and Routing Protocols) เครือข่ายไอพีสวิตช์แบบซ้อนทับนี้เป็นรูปแบบที่นำมาใช้งานได้ง่ายที่สุด เนื่องจากทั้งระบบไอพีและเอทีเอ็ม ต่างก็เป็นระบบมาตรฐานที่ถูกนำมาใช้งานกันอย่างแพร่หลาย แต่การทำงานร่วมกันของทั้งสองระบบนี้มีความซ้ำซ้อนกันอยู่มาก และเครือข่ายไอพีสวิตช์ตามรูปแบบนี้ยังจะต้องสามารถรองรับข้อกำหนดของการสื่อสารในทั้งสองรูปแบบไปพร้อมๆ กันได้อีกด้วย

- เครือข่ายไอพีสวิตช์ในรูปแบบซ้อนทับ จะมีลักษณะการทำงานดังต่อไปนี้
  - ใช้รูปแบบค่าตำแหน่งที่อยู่ของเครื่องในลักษณะแยกต่างหากจากกัน (Separated Addressing)
  - ใช้ข้อกำหนดการจัดการเส้นทางแบบแยกจากกัน (Separate Routing Protocols) เช่น OSPF ในระบบไอพี และ PNNI ในระบบเอทีเอ็ม ซึ่งเท่ากับว่ามี 2 โครงสร้างการสื่อสารที่ไม่เกี่ยวข้องกันอยู่บนระบบเครือข่าย และต่างก็ไม่รู้ว่ามีอีกโครงสร้างหนึ่งทำงานอยู่ด้วยกัน
  - ต้องการการแปลงค่าตำแหน่งระหว่างระบบไอพีกับเอทีเอ็ม (Address Resolution) รวมทั้งรูปแบบการสื่อสัญญาณและจัดการเส้นทางการแบบ UNI/PNNI (User Network Interface / Private Network-to-Network Interface) ในกรณีที่มีการใช้งาน (SVCs) ไม่ว่าจะเป็นการเชื่อมต่อปกติและการเชื่อมต่อแบบเส้นทางลัด
  - โดยทั่วไปแล้วจะใช้ระบบไอพีสวิตช์แบบเสมือนในการทำงาน แต่ก็มีอีกตัวอย่างหนึ่งซึ่งสามารถทำได้เช่นกันคือการใช้รูปแบบที่เรียกว่าบันช์ (Bunch) ซึ่งจะเป็นการนำเอาเครื่องโฮสต์และเราเตอร์ในแบบไอพี ไปต่อเข้ากับแกนหลักของการสื่อสารที่เป็นระบบเอทีเอ็ม (ATM Backbone)

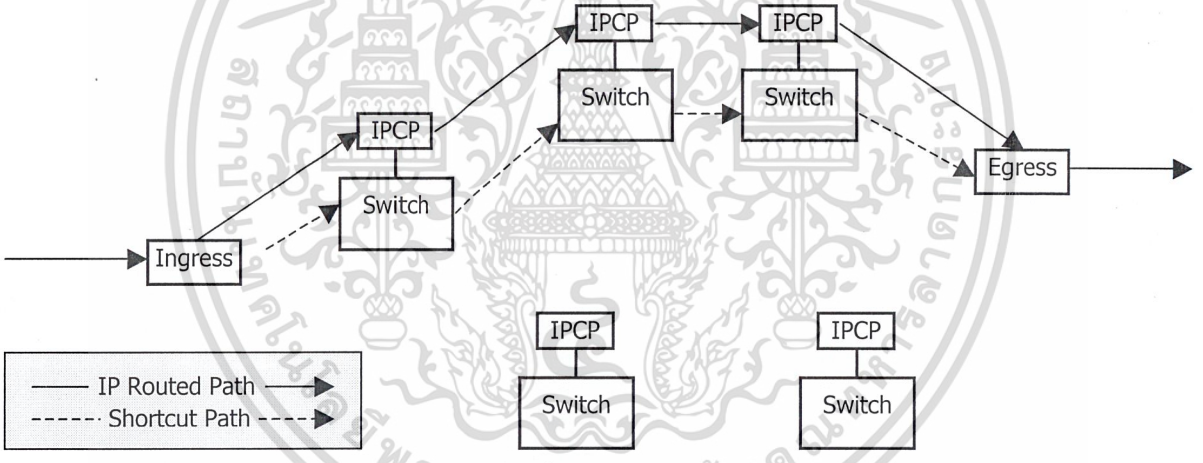


รูปที่ 5-3 โครงแบบไอพีสวิตช์ชนิดซ้อนทับ (Overlay Model) โดยใช้ระบบไอพีสวิตช์แบบเสมือน (Virtual IP Switch)

5.5.2 โครงแบบเพียร์ (Peer Model)

ในโครงแบบชนิดเพียร์นี้ ไอพีสวิตช์จะถูกกำหนดขึ้นมาจากองค์ประกอบไอพีสวิตช์ที่มีระบบค่าตำแหน่งที่อยู่ชนิดไอพี (IP Address Space) และรองรับการทำงานภายใต้ข้อตกลงการค้นหาเส้นทางในแบบไอพี (IP Routing Protocol) แต่เพียงอย่างเดียวเท่านั้น ในโครงแบบนี้จะใช้ข้อตกลงการควบคุมการสื่อสารแบบแยกจากกัน (Separate Control Protocol) ในการเชื่อมโยงการสื่อสารข้อมูลในระบบไอพีให้ไปเป็นการใช้เส้นทางลัด ตัวอย่างของระบบเครือข่ายที่ใช้โครงแบบนี้ก็คือระบบเครือข่ายที่ทำงานด้วยโพรโตคอล IFMP กับ GSMP ของบริษัทอิปสิลอนนั่นเอง

- เครือข่ายไอพีสวิตช์ในรูปแบบเพียร์ จะมีลักษณะการทำงานดังต่อไปนี้
  - ระบบไอพีสวิตช์จะเก็บรักษาเฉพาะค่าตำแหน่งที่อยู่ในระบบไอพีอย่างเดียวเท่านั้น
  - มีรูปแบบข้อตกลงการค้นหาเส้นทางแบบไอพีเท่านั้น
  - ไอพีสวิตช์ใช้ข้อตกลงการควบคุมการสื่อสารพิเศษในการเชื่อมโยงการสื่อสารข้อมูลในระบบไอพีให้ไปเป็นการใช้เส้นทางลัด
  - สามารถรองรับการทำงานได้ทั้งแบบเรา



รูปที่ 5-4 โครงแบบไอพีสวิตช์ชนิดเพียร์ (Peer Model)

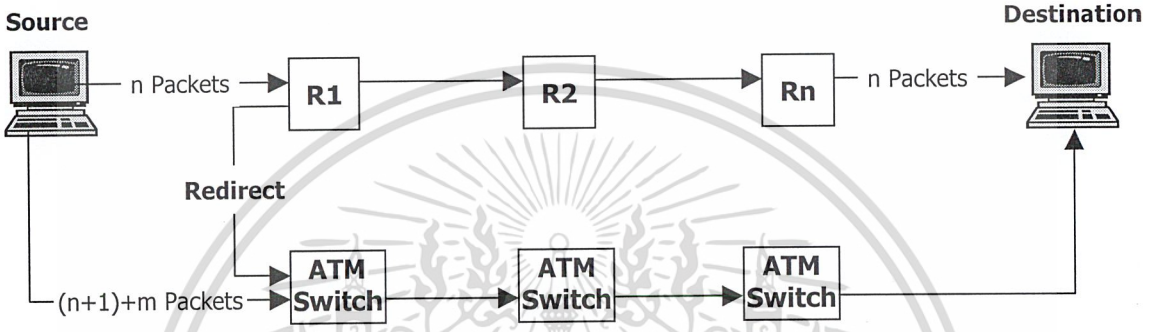
Attributes	Overlay Model	Peer Model
Address Space	Separate (IP และ ATM)	Single (IP)
Routing Protocols	IP และ ATM	IP เท่านั้น
Control Protocols	ATM Forum Protocols	Special IP-to-Shortcut Protocols
Address Conversion	Address Resolution (IP-to-ATM)	Direct IP-to-VC Mapping
IP Switch Components	Virtual IP Switches, Routers, Hosts	IP Switch Devices, Routers, Hosts
ตัวอย่าง	MPOA, Router attached to ATM Network	ARIS, IP Switches runnung IFMP/GSMP

ตารางที่ 5-1 ตารางเปรียบเทียบคุณสมบัติต่างๆ ของโครงแบบไอพีสวิตช์ทั้ง 2 ชนิด

5.6 ประเภทของไอพีสวิตช์ (IP Switching Types)

5.6.1 ทำงานโดยกระแสข้อมูล (Flow-Driven Solution)

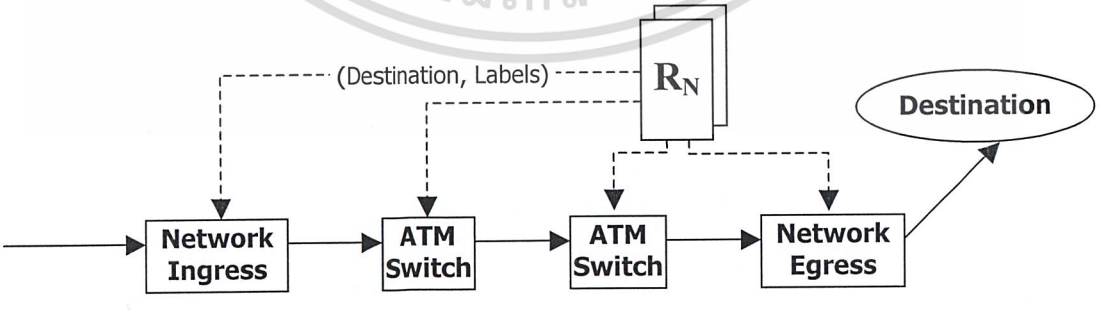
ระบบเครือข่ายไอพีสวิตช์ที่ทำงานด้วยกระแสข้อมูลนั้น จะทำงานโดยอาศัยกระแสข้อมูลชนิดไอพีที่ไหลผ่านเข้ามา กระแสข้อมูลเหล่านี้จะถูกเรียกว่าโฟลว์ (Flow) ซึ่งถูกกำหนดขึ้นมาจากลำดับของชุดข้อมูลที่มีต้นกำเนิดมาจากที่เดียวกัน มีจุดหมายปลายทางไปยังที่เดียวกัน และถูกระบุให้ส่งไปยังช่องทางการสื่อสารเดียวกันด้วยนั่นเอง การทำงานพื้นฐานของระบบเครือข่ายไอพีสวิตช์ประเภทนี้จะเป็นไปตามแผนภาพที่ 5-5



รูปที่ 5-5 แผนผังการทำงานของระบบเครือข่ายไอพีสวิตช์ประเภทที่ทำงานโดยกระแสข้อมูล

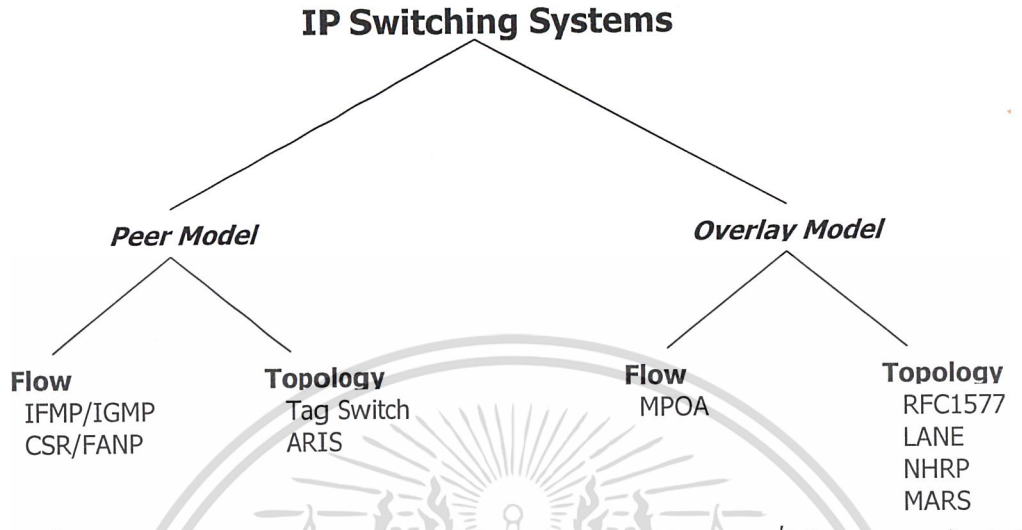
5.6.2 ทำงานโดยโครงสร้างเครือข่าย (Topology-Driven Solution)

เครือข่ายไอพีสวิตช์ที่ทำงานด้วยโครงสร้างเครือข่ายนั้น การทำงานจะขึ้นอยู่กับรูปแบบการเชื่อมต่อทางกายภาพของระบบเครือข่ายไอพีสวิตช์เอง โดยอาศัยข้อตกลงการค้นหาเส้นทางที่กำหนดไว้ในเครือข่าย เช่น OSPF, BGP เป็นต้น และค่าฉลาก (Label) ใหม่ที่สัมพันธ์กับค่า IP Prefix(s) ปลายทาง ซึ่งแสดงถึงตำแหน่งเครือข่ายปลายทางที่เครื่องเป้าหมายตั้งอยู่นั้น จะถูกสร้างขึ้นและถูกส่งกระจายออกไปยังไอพีสวิตช์ตัวอื่นๆ ที่อยู่ภายในขอบเขตการค้นหาเส้นทางเดียวกัน จากนั้นทุกๆ การสื่อสารข้อมูลที่เกิดจะเดินทางไปตามเส้นทางที่ระบุโดยค่าฉลาก VPI/VCI Label นั่นเอง



รูปที่ 5-6 แผนผังการทำงานของระบบเครือข่ายไอพีสวิตช์ประเภทที่ทำงานโดยโครงสร้างเครือข่าย

5.7 การจัดจำแนกประเภทของเทคโนโลยีไอพีสวิตชิ่ง (IP Switching Taxonomy)



รูปที่ 5-7 แผนภูมิรากไม้แสดงการจัดจำแนกประเภทของเทคโนโลยต่างๆ ที่ใช้หลักการของไอพีสวิตช์

บทที่ 6

การติดต่อใช้งานแพ็กเก็ตไดรเวอร์ (Packet driver usage)

6.1 แพ็กเก็ตไดรเวอร์ (Packet Driver Description)

แพ็กเก็ตไดรเวอร์ คือ ซอฟต์แวร์ที่ช่วยประสานการทำงาน (Interface) กับแผงวงจรสื่อสารในระบบเครือข่าย (Network Interface Card) ซึ่งเป็นเครื่องมือที่ช่วยให้นักพัฒนาซอฟต์แวร์เครือข่ายในระดับลึกๆ สามารถเขียนโปรแกรมจัดการการสื่อสารข้อมูล และควบคุมการทำงานของอุปกรณ์สื่อสารในระบบเครือข่ายผ่านระบบปฏิบัติต่างๆ ได้ โดยไม่จำเป็นต้องศึกษาฟังก์ชั่นคำสั่งของระบบปฏิบัติการที่มีอยู่อย่างมากมายให้ครบเสียก่อน เนื่องจากเราสามารถใช้ชุดคำสั่งที่แพ็กเก็ตไดรเวอร์สร้างเอาไว้ให้แล้วได้โดยตรง ซึ่งแต่ละชุดคำสั่งของแพ็กเก็ตไดรเวอร์นั้น ก็คือรูทีนที่จะเข้าไปเรียกใช้กลุ่มฟังก์ชั่นคำสั่งต่างๆ ที่ซับซ้อนของระบบปฏิบัติการให้แทนอีกทีหนึ่ง ทำให้การเขียนโค้ดโปรแกรมง่ายขึ้นนั่นเอง

6.2 WinPcap

แพ็กเก็ตไดรเวอร์ “WinPcap” คือสถาปัตยกรรมโครงสร้างการทำงานของระบบคอมพิวเตอร์ ซึ่งได้รับการพัฒนาขึ้นมาโดยห้องวิจัย *Lawrence Berkeley Laboratory* ของมหาวิทยาลัยแห่งแคลิฟอร์เนีย เพื่อใช้สำหรับตรวจจับชุดข้อมูล (Packet Capture) และวิเคราะห์ระบบเครือข่าย (Network Analysis) ภายใต้สภาวะแวดล้อมของระบบปฏิบัติการวินโดวส์ 32 บิต (Win32) โดยโครงงานนี้จะอ้างอิงกับ WinPcap รุ่นที่ 2.1 ซึ่งออกเผยแพร่ในวันที่ 26 มีนาคม พ.ศ.2544 เป็นต้นมา เนื่องจากเป็นรุ่นล่าสุดที่มีในขณะที่กำลังทำโครงงานนี้อยู่ และสนับสนุนการทำงานร่วมกับระบบปฏิบัติการณ์วินโดวส์ได้ตั้งแต่รุ่น Windows95, Windows98, WindowsME, จนกระทั่งถึง Windows 2000 ทุกแพลตฟอร์ม แต่ยังไม่สามารถสนับสนุนการทำงานร่วมกับ Windows XP (Win64) ได้

WinPcap ประกอบไปด้วย การกรองชุดข้อมูลในระดับเคอร์เนล (Kernel-Level Packet Filter), คลังคำสั่งที่ใช้งานร่วมกันในระดับล่าง (Low-Level Dynamic Link Library) ซึ่งถูกรวบรวมเข้าไว้เป็นแฟ้มข้อมูลที่ชื่อ "Packet.DLL", และคลังคำสั่งอิสระในระดับสูง (High-Level and System Independent Library) ในแฟ้มข้อมูลที่ชื่อ "WPcap.DLL" ซึ่งยึดตามรูปแบการทำงานของ “LibPcap” รุ่นที่ 0.5 อันเป็นคลังคำสั่งสำหรับดักจับชุดข้อมูลที่ได้รับการยอมรับอย่างกว้างขวางในระบบยูนิกซ์ (UNIX)

ตัวกรองชุดข้อมูล (Packet Filter) คือซอฟต์แวร์ขับอุปกรณ์ (Device Driver) ตัวหนึ่งซึ่งช่วยเพิ่มความสามารถในการตรวจจับและส่งต่อข้อมูลดิบ (Raw Data) ที่อยู่ในชุดข้อมูลจากแผงวงจรสื่อสารเครือข่ายได้ และมีความเป็นไปได้ที่จะตรวจสอบและจัดเก็บชุดข้อมูลที่ผ่านการกรองมาแล้วเอาไว้ในหน่วยความจำชั่วคราว (Buffer) ได้อีกด้วย

คลังคำสั่งที่ใช้งานร่วมกันได้ในระดับล่างที่อยู่ในแฟ้มข้อมูล "Packet.DLL" นั้น เป็นข้อกำหนดกรรมวิธีการติดต่อสื่อสารระหว่างซอฟต์แวร์ที่พัฒนาขึ้นมากับระบบปฏิบัติการ หรือที่เรียกว่าเอพีไอ (API: Application Programming Interface) ซึ่งสามารถนำมาใช้เพื่อเข้าถึงฟังก์ชั่นการทำงานต่างๆ ของแพ็กเก็ตไดรเวอร์ได้โดยตรง และเป็นอิสระจากระบบปฏิบัติการอีกด้วย

คลังคำสั่งอิสระในระดับสูงที่เก็บไว้ในแฟ้มข้อมูล WPcap.DLL จะทำหน้าที่ถ่ายทอด (Export) กลุ่มของ Capture Primitives ในระดับสูงที่เข้ากันได้กับ LibPcap ของระบบยูนิกซ์ ออกมา ซึ่งฟังก์ชั่นเหล่านี้จะยอมให้ทำการดักจับชุดข้อมูลได้โดยไม่ยึดติดกับแพลตฟอร์มของระบบปฏิบัติการ หรือฮาร์ดแวร์เครือข่ายแต่อย่างใด

WinPcap ได้ถ่ายทอดการเรียกใช้ฟังก์ชั่น (Call) ของระบบปฏิบัติการออกมาเป็น 2 กลุ่มด้วยกัน คือกลุ่มของฟังก์ชั่นขับชุดข้อมูลในระดับล่าง กับกลุ่มฟังก์ชั่นดักจับชุดข้อมูลในระดับสูง ซึ่งกลุ่มแรกจะทำหน้าที่รับและส่งชุดข้อมูลในแบบ "Raw Mode" ส่วนกลุ่มที่สองนั้นจะเป็นกลุ่มที่รวมเอารูปแบบคลังคำสั่งของ LibPcap ในระบบยูนิกซ์ เอาไว้ด้วย

ข้อมูลเพิ่มเติมเกี่ยวกับ WinPcap สามารถหาได้จากโฮมเพจของเน็ตกรุ๊ป ตาม URL ต่อไปนี้

http://netgroup-serv.polito.it/winpcap/default.htm

6.3 Packet Driver API

Packet Driver API (Packet.DLL) คือคลังคำสั่งที่ใช้งานร่วมกันได้ในระดับล่างชนิดหนึ่ง ที่ทำหน้าที่ประสานการทำงานระหว่างโปรแกรมประยุกต์ของผู้ใช้กับโปรแกรมขับการดักจับชุดข้อมูล โดย Packet Driver API จะใช้งานกลุ่มของฟังก์ชั่นที่ทำให้การติดต่อกับโปรแกรมขับระบบการสื่อสารข้อมูลทำได้สะดวกง่ายดายยิ่งขึ้น เนื่องจากหลีกเลี่ยงการใช้งาน System Calls หรือ IOCTLs ในโปรแกรมประยุกต์ของผู้ใช้ ยิ่งไปกว่านั้น Packet Driver API ยังได้จัดเตรียมฟังก์ชั่นสำหรับควบคุมการดำเนินงานของแผงวงจรสื่อสารเครือข่าย (Handle), การอ่านและเขียนชุดข้อมูลไปยังระบบเครือข่าย, การจัดหน่วยความจำชั่วคราวและการกรองชุดข้อมูลในตัวขับเร้า, และอื่นๆ เอาไว้ให้เรียบร้อยแล้วอีกด้วย

Packet Driver API มีอยู่ด้วยกัน 2 รุ่นให้เลือกใช้ คือรุ่นที่สนับสนุนการทำงานกับระบบปฏิบัติการ Windows 95/98 กับรุ่นที่สนับสนุนการทำงานกับระบบปฏิบัติการ Windows NT/2000 ซึ่งทั้ง 2 รุ่นทำหน้าที่ถ่ายทอดรูปแบบการเชื่อมต่อกับระบบปฏิบัติการออกมาในลักษณะเดียวกัน ทำให้การพัฒนาซอฟต์แวร์ดักจับชุดข้อมูลที่ไม่อิงกับระบบปฏิบัติการนั้นสามารถทำได้อย่างง่ายดาย โดยการเรียกผ่าน Packet Driver API จะทำให้ซอฟต์แวร์ที่ถูกเขียนขึ้นมาในสภาวะแวดล้อมแบบใดแบบหนึ่ง สามารถนำไปใช้งานได้กับทั้งระบบปฏิบัติการ Windows 95/98 และ Windows NT/2000 ในทันที

6.3.1 Packet.DLL กับ WPcap.DLL

ในกรณีที่ต้องการจะเขียนซอฟต์แวร์ที่ต้องมีการดักจับชุดข้อมูลขึ้นมานั้น ถ้าหากไม่มีความจำเป็นที่จะต้องเข้าถึงการทำงานในระดับลึกแล้ว การเขียนโปรแกรมเรียกใช้ฟังก์ชั่นผ่าน WPcap.DLL จะเหมาะสมกว่าเขียนผ่าน API

6.4 ข้อกำหนดสำหรับเอกสารฉบับนี้ (Document Conventions)

ตัวเลขที่ปรากฏในเอกสารฉบับนี้จะแสดงในรูปแบบของภาษาซี (C-Style Representation) เช่น เลข "11" จะถูกเขียนอยู่ในรูปของเลขฐานสิบหกได้เป็น "0x0B" และเลขฐานแปดจะแทนด้วย "013" ซึ่งจะใช้ในการอ้างอิงหมายเลขฮาร์ดแวร์เครือข่าย (Network Hardware Address), หมายเลขเครื่องปลายทาง (Destination Address), หมายเลขเครื่องต้นทาง (Source Address), และข้อมูลการแยกสัญญาณ (Demultiplexing) เป็นต้น

6.5 ความรู้เบื้องต้นและแรงจูงใจในการพัฒนา (Introduction and Motivation)

ในหัวข้อต่อไปนี้ จะอธิบายถึงหลักการการเขียนซอฟต์แวร์ ควบคุมการทำงานของแผงวงจรสื่อสารในระบบเครือข่าย, การจัดการ - การดำเนินการกับชุดข้อมูล รวมทั้งการบริหารการสื่อสารข้อมูลในระดับดาตาลิงค์ บนระบบปฏิบัติการ Windows 2000 Server ผ่านแพ็กเก็ตไดรเวอร์ ซึ่งทำให้การพัฒนาซอฟต์แวร์ทำได้ง่าย และสะดวกกว่าการเขียนโค้ดโดยเรียกใช้ฟังก์ชั่นพื้นฐานที่ติดต่อกับระบบปฏิบัติการโดยตรง เนื่องจากแพ็กเก็ตไดรเวอร์ได้รวบรวมเอาลำดับการเรียกใช้ฟังก์ชั่นต่างๆ ของวินโดวส์เข้าด้วยกันแล้วสร้างเป็นรูปแบบการทำงานเฉพาะอย่างขึ้นมาใหม่ สำหรับให้ผู้พัฒนาซอฟต์แวร์นำไปใช้ในโปรแกรมของตนเองอีกที

แพ็กเก็ตไดรเวอร์จะทำการคัดแยกชุดข้อมูลที่มาจากโปรแกรมประยุกต์ต่างๆ โดยอาศัยประเภทของชุดข้อมูลมาตรฐานของสื่อในระบบเครือข่าย (Network Media's Standard Packet Type) หรืออาศัยการเข้าถึงบริการของแพ็กเก็ตไดรเวอร์ เพื่อให้การทำงานในระดับชั้นต่างๆ ของข้อตกลงการสื่อสารข้อมูลบนระบบเครือข่าย (Protocol Stack) สามารถทำงานได้อย่างเป็นอิสระ โดยไม่ขึ้นกับชนิดของแผงวงจรสื่อสารระบบเครือข่ายที่ใช้งานอยู่

แพ็กเก็ตไดร์เวอร์ได้จัดเตรียมฟังก์ชั่นต่างๆ สำหรับเข้าถึงชนิดของแพ็กเก็ตไดรเวอร์, การยุติการเข้าถึงแพ็กเก็ตไดรเวอร์, การส่งแพ็กเก็ต, การเก็บสถิติของการสื่อสารข้อมูลบนแผงวงจร และ การเก็บข้อมูลเกี่ยวกับแผงวงจรสื่อสารเครือข่าย

การจัดการเกี่ยวกับข้อตกลงการสื่อสารข้อมูลผ่านแพ็กเก็ตไดรเวอร์สามารถใช้งานร่วมกับเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) ได้อย่างสมบูรณ์ และสามารถใช้ประโยชน์จากบริการอื่นๆได้อีก แต่ซอฟต์แวร์บางตัว ที่ไม่ได้ใช้แพ็กเก็ตไดรเวอร์นั้น จะไม่สามารถทำงานร่วมกับซอฟต์แวร์อื่นๆ ในเครื่องเดียวกันได้อย่างสมบูรณ์

บทที่ 7
ขั้นตอนการออกแบบและพัฒนาซอฟต์แวร์

7.1 กำหนดสภาพแวดล้อมของระบบ

7.1.1 ระบบไอพีสวิตช์ (IP Switch System)

จากการที่ระบบไอพีสวิตช์มีอยู่ด้วยกัน 2 รูปแบบ คือแบบที่เรียกว่า “อุปกรณ์ไอพีสวิตช์” (IP Switch Device) กับ “ไอพีสวิตช์แบบเสมือน” (Virtual IP Switch) แต่เนื่องจากระบบไอพีสวิตช์แบบเสมือนนั้นจะเป็นการทำงานของข้อตกลงการสื่อสารแบบไอพีบนโครงสร้างการส่งผ่านข้อมูลแบบเอทีเอ็ม ซึ่งแม้ว่าจะให้การทำงานที่มีประสิทธิภาพสูงมาก แต่ก็มีความซับซ้อนสูง และไม่ยืดหยุ่นเท่าที่ควร ส่วนในรูปแบบของอุปกรณ์ไอพีสวิตช์นั้นไม่ได้ระบุไว้ว่าจะต้องทำงานบนโครงสร้างแบบเอทีเอ็ม ผู้ออกแบบจึงสามารถเลือกได้ว่าจะทำบนโครงสร้างแบบใด ซึ่งในโครงงานนี้ก็ได้เลือกที่จะพัฒนาเป็นซอฟต์แวร์จำลองการทำงานของอุปกรณ์ไอพีสวิตช์บนเครือข่ายแบบอีเธอร์เน็ต (Ethernet LAN) และเนื่องจากอุปกรณ์ไอพีสวิตช์นั้นตั้งอยู่บนพื้นฐานของการทำงานในของระบบสวิตชิ่งเป็นหลัก ซึ่งมีข้อกำหนดอยู่ว่าระบบเครือข่ายใดๆ ก็ตามที่สวิตช์จะสามารถเชื่อมต่อเข้าด้วยกันได้ ระบบเครือข่ายเหล่านั้นจะต้องใช้สื่อในการส่งผ่านข้อมูลเป็นชนิดเดียวกัน และใช้ข้อตกลงการสื่อสารข้อมูลแบบเดียวกันเท่านั้น ดังนั้นในโครงงานนี้จึงได้เลือกที่จะใช้ข้อตกลงการสื่อสารข้อมูลเป็น “IPv4” บนเครือข่ายอีเธอร์เน็ตนั่นเอง

7.1.2 รูปแบบการจัดการค่าตำแหน่งที่อยู่ของเครื่อง (Addressing Model)

สืบเนื่องมาจากการเลือกระบบแบบอุปกรณ์ไอพีสวิตช์ ดังนั้นจึงควรจะต้องเลือกใช้ค่าตำแหน่งแบบ “IP-to-VC” แต่เนื่องจากการจัดสร้างช่องทางการเชื่อมต่อเสมือน (Virtual Connection) นั้นค่อนข้างซับซ้อนและยุ่งยาก ในโครงงานนี้จึงเลือกที่จะใช้ค่าตำแหน่งแบบ “Separated Addressing Model” โดยแยกการทำงานของ “MAC” กับ “IP” ออกจากกัน แทนที่จะเป็น “ATM” ซึ่งในความเป็นจริงแล้ว การใช้ค่าตำแหน่งแบบแยกจากกันนั้น จะใช้แยกการทำงานของ “IP” กับ “ATM” ออกจากกันในระบบที่เป็นไอพีสวิตช์แบบเสมือน

7.1.3 โครงแบบไอพีสวิตช์ (IP Switching Model)

และสืบเนื่องมาจากการการเลือกใช้ค่าตำแหน่งที่อยู่ของเครื่องแบบแยกจากกันนั้น จะเป็นการระบุโครงแบบของไอพีสวิตช์ให้เป็นชนิดซ้อนทับ (Overlay Model) ไปในตัว คือจะแยกการทำงานในระดับดาตาลิงค์กับเน็ตเวิร์คให้มีการทำงานที่เป็นอิสระจากกันโดยสิ้นเชิงนั่นเอง แล้วอาศัยกรรมวิธีของการควบรวมแผนที่เส้นทาง (Address Mapping) ในการเชื่อมโยงตำแหน่งปลายทางที่หาพบในระดับเน็ตเวิร์คลงสู่ระดับดาตาลิงค์ แล้วลัดการคำนวณ (Bypass) การค้นหาเส้นทาง และส่งผ่านข้อมูลในระดับบนทิ้งไป ให้เหลือแต่ทำการสวิตช์ในระดับล่างที่เร็วกว่าเท่านั้น

7.2 การดำเนินการภายในของไอพีสวิตช์

ไอพีสวิตช์มีการทำงานอยู่ 2 ส่วนด้วยกัน คือส่วนที่ทำหน้าที่สับเปลี่ยนทิศทางของชุดข้อมูลในระดับดาตาลิงค์ (Layer-2 Switching) กับส่วนการค้นหาเส้นทางและส่งต่อชุดข้อมูลในระดับเน็ตเวิร์ค (Layer-3 Forwarding) โดยในส่วนที่ทำหน้าที่จัดการทิศทางของชุดข้อมูลในระดับดาตาลิงค์นั้น จะอาศัยหลักการการทำงานตามรูปแบบของระบบสวิตชิ่งตามปกติเป็นหลัก ซึ่งมีกลวิธีต่างๆ ที่เกี่ยวข้องอยู่ 2 ประเภทด้วยกัน คือ กลวิธีการส่งต่อชุดข้อมูล (Switch Forwarding Technique) และกลวิธีการควบคุมเส้นทาง (Switch Path Control) ซึ่งในแต่ละประเภทนั้นก็มีวิธีการต่างๆ ให้เลือกใช้หลายแบบตามที่ได้กล่าวไว้แล้วในบทที่ 3 ดังนั้นในขั้นตอนการออกแบบและทดลอง เราจึงได้เลือกที่จะวิธีใดวิธีหนึ่งของทั้ง 2 กลวิธีดังต่อไปนี้

ในส่วนของการส่งต่อชุดข้อมูลนั้น เราได้เลือกเอากลวิธีตามแบบของการใช้ค่าตำแหน่งเครื่องปลายทาง (Destination Address) ส่วนกลวิธีกลวิธีในการควบคุมเส้นทางนั้น เราจะเลือกใช้วิธีการเรียนรู้เส้นทาง (Address Learning) เนื่องจากทั้ง 2 วิธีนี้ มีความซับซ้อนน้อย ง่ายต่อการศึกษาทดลอง อีกทั้งยังเป็นวิธีที่นิยมใช้กันทั่วไปในระบบเครือข่ายท้องถิ่นอีกด้วย

ในส่วนของการควบคุมการทำงานของสวิตช์ (IPCP: IP Control Point) ซึ่งเป็นส่วนประกอบในระดับบน (Layer3) ของไอพีสวิตช์ และทำให้สวิตช์ทั่วไปกลายมาเป็นไอพีสวิตช์ได้นั้น เราได้ทำการออกแบบดังต่อไปนี้

7.2.1 โครงสร้างข้อมูลหลักของโปรแกรม

ตารางเก็บค่าตำแหน่งที่อยู่

PORT#1	PORT#2	.....	PORT#n
00-02-AB-EA-12-FF 00-80-3C-85-3D-E6 00-40-05-2F-F9-91	00-10-B5-54-26-C4		XX-XX-XX-XX-XX-XX

(ก)

PORT#1	PORT#2	.....	PORT#n
161.246.5.169 161.246.5.173 161.246.5.177	161.246.6.180 161.246.6.195		XXX.XXX.XXX.XXX

(ข)

รูปที่ 7-1 ตารางเก็บข้อค่าตำแหน่งที่อยู่แบบ Separated Addressing Model ที่ใช้วิธีการเรียนรู้ค่าตำแหน่ง
(ก) ตารางในระบบสวิตชิ่งทั่วไป ซึ่งถูกนำมาใช้ในไอพีสวิตช์ระดับดาตาลิงค์ (Layer2)
ข) ตารางในระบบไอพีสวิตช์ระดับเน็ตเวิร์ค (Layer3)

7.2.2 ฟังก์ชั่นการทำงานหลักของโปรแกรม

ซอฟต์แวร์ตัวนี้ได้ถูกออกแบบมาให้ถูกติดตั้งและใช้งานบนระบบคอมพิวเตอร์ส่วนบุคคล (PC: Personal Computer) ในตระกูล IBM PC Compatible ที่ได้รับการติดตั้งและสามารถใช้งานแผงวงจรสื่อสารเครือข่ายท้องถิ่น (LAN Card) ชนิดอีเธอร์เน็ต (Ethernet) ตั้งแต่ 2 ใบขึ้นไปเป็นที่เรียบร้อยแล้ว ซอฟต์แวร์ตัวนี้ทำงานบนระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 32 บิต (Win32) ซึ่งได้ทำการทดสอบแล้วว่าสามารถทำงานได้ทั้งบน Windows 95/98, Windows NT และ Windows 2000 รุ่น Professional, Server, และ Advanced Server โดยไม่ต้องปรับแต่งค่าใดๆ เพิ่มเติม ทั้งก่อนและหลังการติดตั้งและใช้งานซอฟต์แวร์ตัวนี้

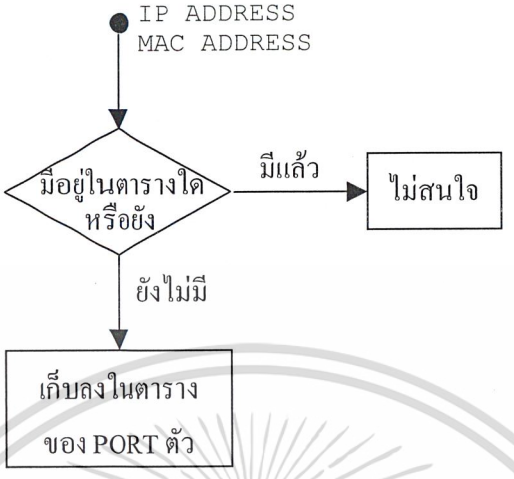
เมื่อติดตั้งซอฟต์แวร์เป็นที่เรียบร้อยแล้ว หลังจากเรียกใช้ ตัวโปรแกรมจะทำหน้าที่เลือกว่าจะส่งต่อชุดข้อมูล (Forward Packets) ที่แผงวงจรสื่อสารเครือข่าย (NIC: Network Interface Card) บนเครื่องได้รับเข้ามาหรือไม่ และถ้าเห็นควรว่าจะส่งต่อชุดข้อมูลเหล่านั้นออกไป ก็จะพิจารณาว่าจะส่งออกไปยังแผงวงจรสื่อสารใดในเครื่องคอมพิวเตอร์ต่อไป

ซึ่งในการที่จะทำเช่นนี้ได้จะต้องมีกระบวนการจัดเก็บข้อมูลบางอย่างเอาไว้ ซึ่งตามวิธีการของการเรียนรู้ค่าตำแหน่งที่อยู่นั้น สวิตช์โดยทั่วไปแล้วจะจัดเก็บเฉพาะค่าตำแหน่งแบบ MAC ของเครื่องต้นทางจากชุดข้อมูลที่ผ่านเข้ามาในแต่ละช่องทางการเชื่อมต่อเท่านั้น แต่สำหรับไอพีสวิตช์ในโครงงานนี้จะต้องจัดเก็บค่าตำแหน่งแบบ IP ของแต่ละชุดข้อมูลเพิ่มเข้ามาด้วย เพื่อใช้ในการควบคุมทิศทางการส่งต่อชุดข้อมูลนั่นเอง

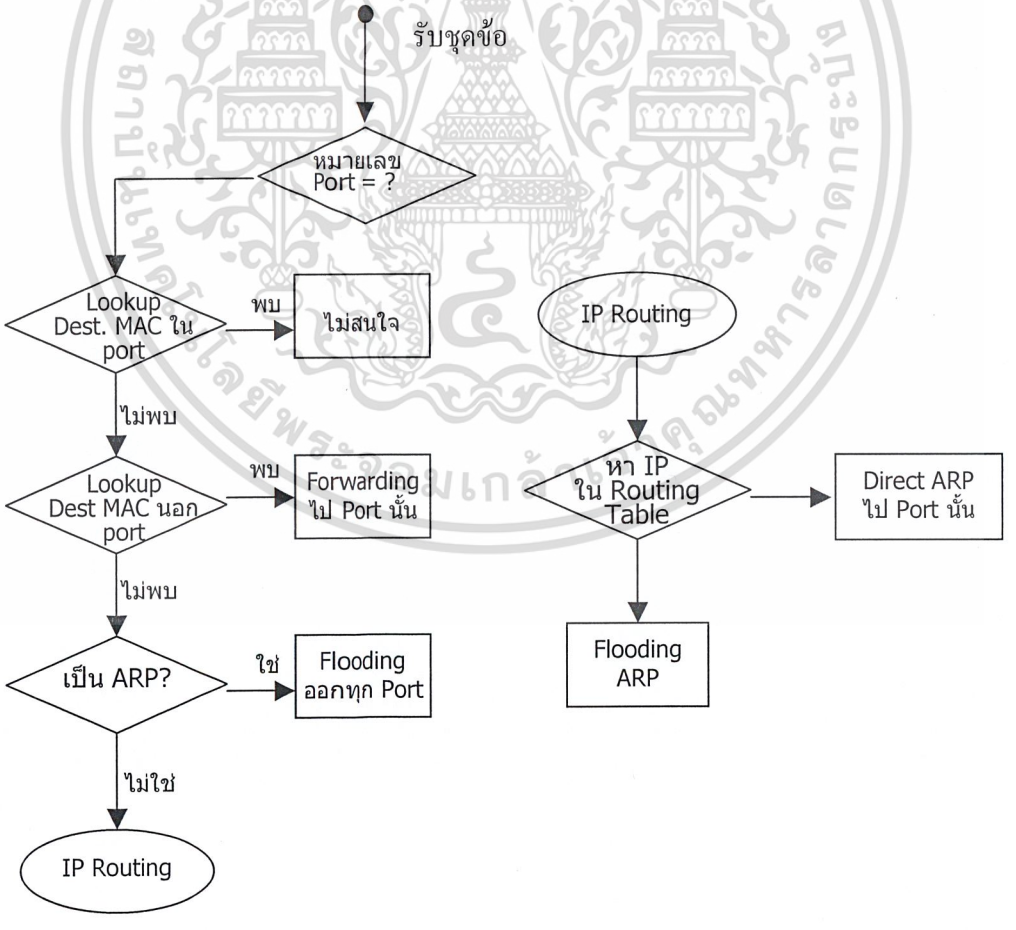
เมื่อไอพีสวิตช์เริ่มต้นทำงานขึ้นมาในครั้งแรกนั้น มันจะสร้างโครงสร้างข้อมูลว่างๆ ตามในรูปที่ XXX สำหรับแต่ละ Port ขึ้นมาก่อน เพื่อใช้รอรับค่า จากนั้นเมื่อมีชุดข้อมูลผ่านเข้ามายังตัวไอพีสวิตช์ทาง Port ใด ก็จะบันทึกค่า SM,SIP ของชุดข้อมูลนั้นๆ ลงไปในโครงสร้างข้อมูลของ Port นั้นๆ

เมื่อไอพีสวิตช์ได้รับชุดข้อมูลเข้ามา มันจะต้องเลือกที่จะส่งต่อชุดข้อมูลเหล่านั้นหรือไม่ และถ้าจะต้องส่งต่อชุดข้อมูลออกไป ก็จะต้องพิจารณาว่าควรจะส่งต่อชุดข้อมูลนั้นในระดับใด ถ้าหากเราทราบ MAC ปลายทางนั้นต่อเชื่อมอยู่กับ Port ใด ก็ FW ไปเลย แต่ถ้าไม่รู้จัก DM นั้น ก็ให้ทำการเราต์หาเส้นทางในแบบเราเตอร์ต่อไป

เมื่อ IPCP ได้รับชุดข้อมูลมาจากส่วนการทำงานในระดับล่าง แสดงว่าชุดข้อมูลนั้นไม่สามารถถูกสวิตช์ออกไปในระดับดาตาลิงค์ได้ โปรแกรมจะไปดึงเอาข้อมูลที่เป็นค่า Destination IP Address มาจากส่วนที่เป็น Raw Data ใน Ethernet Frame ของชุดข้อมูลนั้นๆ แล้วนำค่าที่ได้ไปเทียบกับตาราง เร้าต์ของของไอพีสวิตช์



รูปที่ 7-2 Flowchart แสดงการจัดเก็บค่าตำแหน่ง IP และ MAC ลงในตาราง



รูปที่ 7-3 Flowchart แสดงการทำงานของ Algorithm ของ โปรแกรม

บทที่ 8

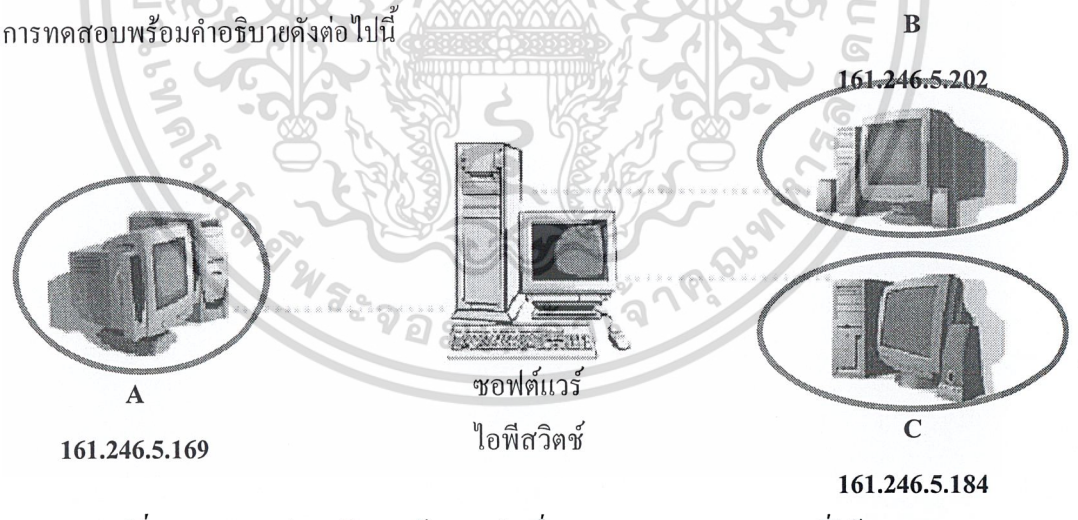
การทดลอง และผลการทดสอบ

8.1 กำหนดรูปแบบการทดลอง

เนื่องจากซอฟต์แวร์นี้มีวัตถุประสงค์เพื่อจำลองการทำงานของอุปกรณ์ไอพีสวิตช์ที่ทำงานด้วยฮาร์ดแวร์ ดังนั้นซอฟต์แวร์ที่พัฒนาขึ้นมานี้จึงถูกออกแบบให้ไม่มีส่วนการทำงานที่เป็น Graphical User Interface (GUI) เพื่อลดการใช้งานทรัพยากรระบบลงให้น้อยที่สุด การทำงานทั้งหมดจึงทำผ่าน Command Line ใน DOS Mode เท่านั้น ซอฟต์แวร์ที่พัฒนาขึ้นมาจะมีอยู่หลายๆ รุ่นด้วยกัน โดยแต่ละรุ่นก็จะมีคุณสมบัติที่แตกต่างกันไป การทดสอบซอฟต์แวร์จะทำโดยเรียกใช้ซอฟต์แวร์รุ่นต่างๆ ผ่านทาง Command Line โดยตรง บนเครื่องที่ทำหน้าที่เป็นอุปกรณ์ไอพีสวิตช์จำลอง แล้วใช้โปรแกรมดักจับแพ็กเก็ตที่ทำงานอยู่บนเครื่องปลายทางแต่ละเครื่องในระบบเครือข่ายท้องถิ่นจำลอง ตรวจสอบความถูกต้องและประสิทธิภาพการทำงานซอฟต์แวร์ไอพีสวิตช์ในรุ่นนั้นต่อไป

8.2 ขั้นตอนการทดลอง

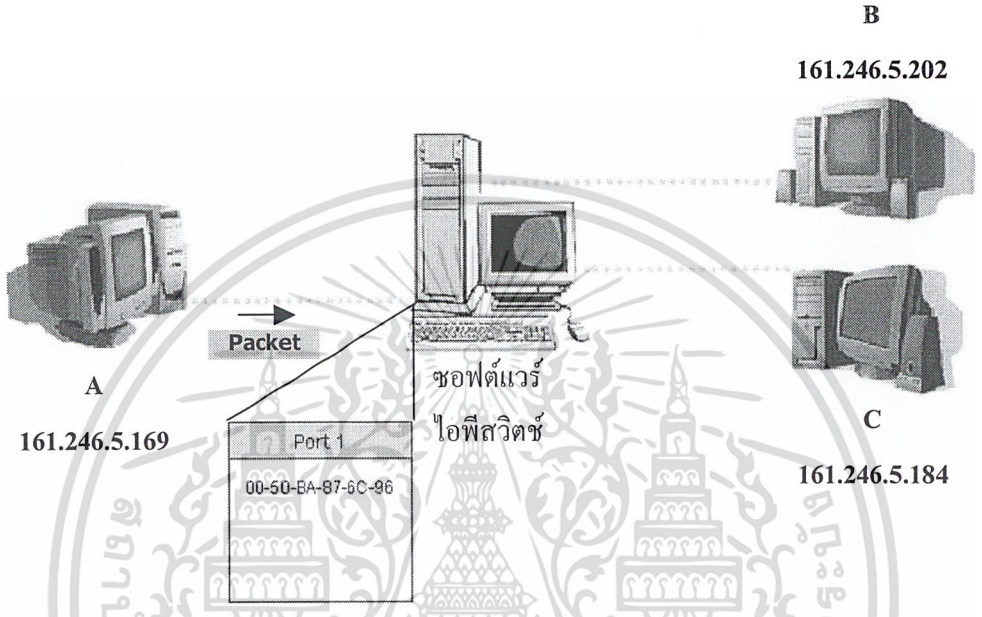
ด้วยหลักการการทำงานของไอพีสวิตช์ดังที่ได้กล่าวไปแล้วนั้น จึงได้จัดสร้างระบบเครือข่ายท้องถิ่นจำลองขึ้นมา เพื่อใช้ทำการทดสอบและแสดงการทำงานอย่างง่ายโดยมีการกำหนดเงื่อนไขการทดสอบพร้อมคำอธิบายดังต่อไปนี้



รูปที่ 8-1 *แสดงรูปแบบโครงสร้างการต่อเชื่อมทางกายภาพของระบบที่ใช้ทดสอบ*

จากแผนภาพในรูปที่ 8-1 เราใช้เครื่องคอมพิวเตอร์ส่วนบุคคลแบบธรรมดาๆ จำนวน 4 เครื่องต่อเชื่อมเข้าด้วยกันในลักษณะ Star Topology โดยให้เครื่องที่เป็นศูนย์กลางรันซอฟต์แวร์ไอพีสวิตช์ที่พัฒนาขึ้นมา ส่วนเครื่องที่เหลือจะถูกต่อเชื่อมเข้ากับเครื่องศูนย์กลางผ่านแผงวงจรสื่อสารเครือข่ายแบบอีเธอร์เน็ต โดยใช้สาย UTP ในการต่อเชื่อม เครื่องลูกข่ายทั้งสามจะมีหน้าที่ส่งข้อมูลไปมาระหว่างกันผ่านเครื่องศูนย์กลาง เพื่อให้มีการสื่อสารข้อมูลเกิดขึ้นในระบบ เสมือนกับว่าเครื่องลูกข่ายแต่ละเครื่องนั้นเป็นเครือข่ายย่อยแต่ละวงที่เชื่อมต่อถึงกันได้โดยผ่านไอพีสวิตช์นั่นเอง

เมื่อเริ่มต้นการทำงาน ซอฟต์แวร์ไอพีสวิตซ์ที่สร้างขึ้นจะสามารถตรวจสอบได้เองว่ามีเครื่องที่ใช้หมายเลขไอพีอะไรเชื่อมต่ออยู่กับตัวมันบ้าง และแต่ละเครื่องนั้นต่อเข้ากับช่องทางการสื่อสารใด ซึ่งแต่ละช่องทางการสื่อสารหรือพอร์ตนั้น โดยนิยามแล้วก็หมายถึงแผงวงจรสื่อสารเครือข่ายแต่ละใบที่มีอยู่ในเครื่องนั่นเอง การทำงานดังกล่าวนี้เกิดขึ้นได้โดยอาศัยการตรวจจับเฟรมและแพ็กเก็ตข้อมูลที่วิ่งผ่านเครื่องศูนย์กลาง ตามอัลกอริทึมของซอฟต์แวร์ที่พัฒนาขึ้นมานั่นเอง

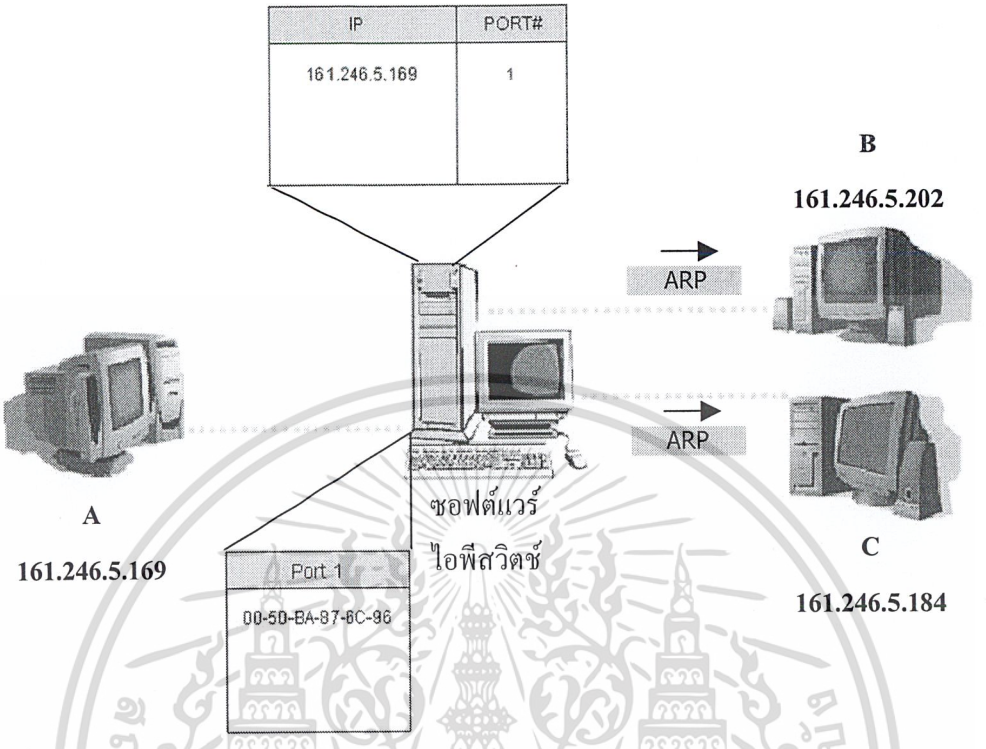


รูปที่ 8-2 แสดงลำดับการทำงานของโปรแกรมในระยะเริ่มต้น

เมื่อเครื่องต้นทาง (A) ชุดข้อมูลออกมาในระบบเครือข่ายโดยระบุไว้ว่าต้องการจะส่งชุดข้อมูลชุดนี้ไปให้แก่เครื่องปลายทาง (C) การทำงานเริ่มแรกของซอฟต์แวร์ไอพีสวิตช์ที่อยู่บนเครื่องศูนย์กลางก็คือ จะเก็บค่าตำแหน่งที่อยู่แบบแม็คของเครื่องต้นทางจากเฟรมข้อมูลดังกล่าวมาและเก็บบันทึกลงในตารางค่าตำแหน่งของช่องทางการสื่อสารที่ 1 ดังรูปที่ 8-2 ซึ่งการดำเนินการดังที่กล่าวไปนี้ จะเป็นไปตามหลักของกลวิธีการควบคุมเส้นทางของอุปกรณ์สวิตชิ่งที่เรียกว่า การเรียนรู้ค่าตำแหน่งนั่นเอง

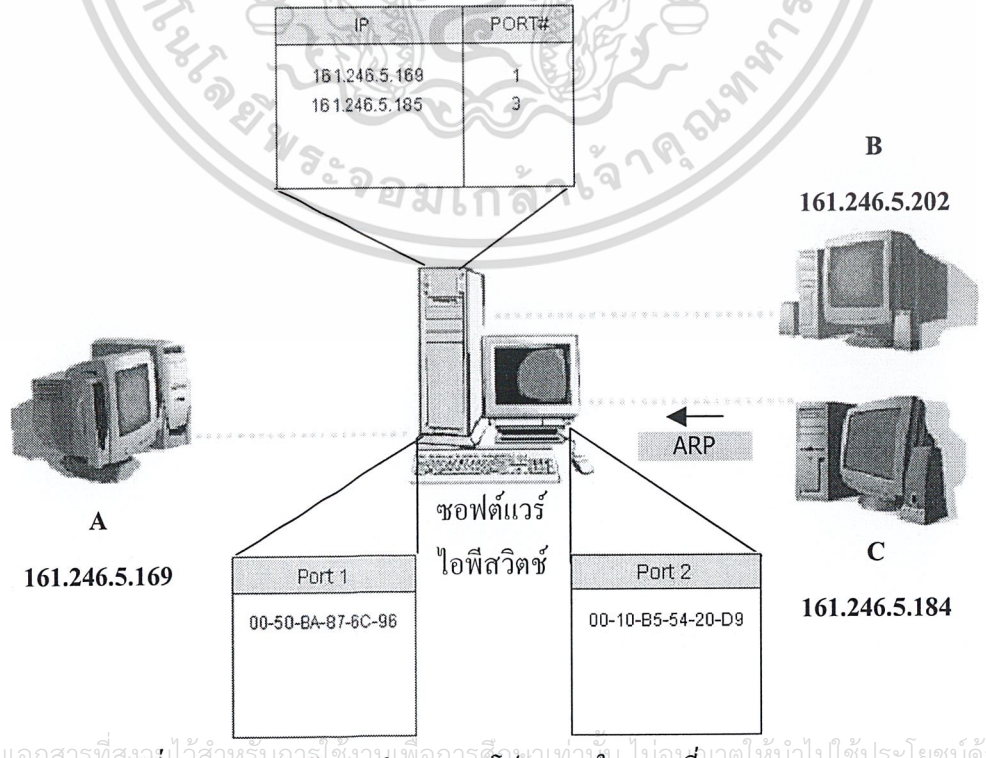
จากนั้นเครื่องที่เป็นศูนย์กลางของการสื่อสารนี้จะต้องทำหน้าที่ส่งต่อชุดข้อมูลนี้ให้ไปถึงยังเครื่องปลายทางที่ถูกต้องโดยใช้กลวิธีการส่งต่อชุดข้อมูลแบบใช้ค่าตำแหน่งปลายทาง โดยซอฟต์แวร์ไอพีสวิตช์จะดึงเอาค่าตำแหน่งที่อยู่แบบแม็คของเครื่องปลายทางจากเฟรมชุดข้อมูลนี้ออกมาแล้วค้นหาเทียบกับตารางเก็บค่าตำแหน่งของช่องทางการสื่อสารต่างๆ ที่มีอยู่ รวมทั้งจากตารางของ ช่องทางการสื่อสารที่ชุดข้อมูลนี้ผ่านเข้ามาเองด้วยโดยหากพบว่าค่าตำแหน่งปลายทางนั้นมีอยู่ในตารางของช่องทางการสื่อสารที่มันใช้ผ่านเข้ามาเองแล้ว ซอฟ์ตแวร์ไอพีสวิตช์ก็จะไม่ส่งต่อชุดข้อมูลนี้ออกไปที่ใดแต่หากไม่พบในตารางใดเลย ดังเช่นในตัวอย่างนี้ ซอฟต์แวร์ไอพีสวิตช์ก็จะจัดเก็บค่าตำแหน่งที่อยู่ในแบบไอพีของเครื่องต้นทางที่ส่งแพ็กเก็ตข้อมูลนี้มาไว้ในตารางไอพีรวม เพื่อใช้ทำการค้นหาเส้นทาง หรือในที่นี้จะใช้ค้นหาเครื่องปลายทางที่ถูกต้องว่าต่อเชื่อมอยู่กับช่องทางการสื่อสารใดโดยซอฟต์แวร์ไอพีสวิตช์จะนำค่าตำแหน่งที่อยู่แบบไอพีของเครื่องปลายทางไปใช้ในเฟรม ARP Request แล้วส่งออกไปยัง

ทุกๆ ช่องทางการเชื่อมต่อของไอพีสวิตช์ ยกเว้นช่องทางที่ชุดข้อมูลผ่านเข้ามา ดังในรูปที่ 8-3 ซึ่งส่วนนี้ถือเป็นหนึ่งในส่วนการทำงานของ IPCP (IP Control Point) ของซอฟต์แวร์ไอพีสวิตช์นั่นเอง



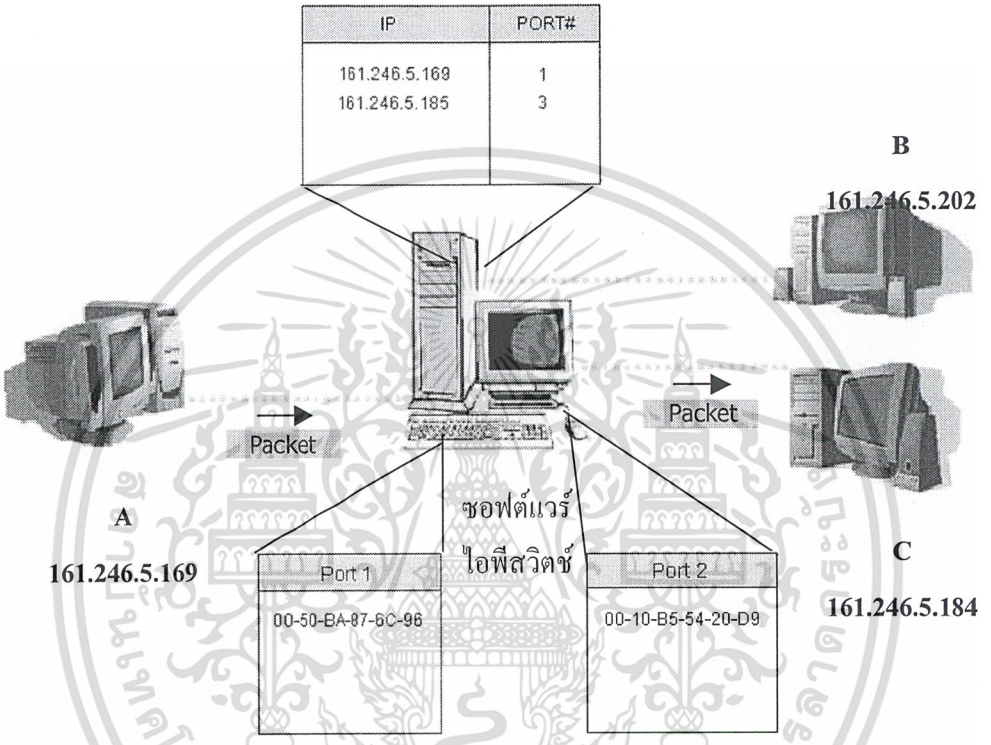
รูปที่ 8-3 แสดงลำดับการทำงานของโปรแกรมในระยะที่ 2 ของการทดสอบ

เมื่อซอฟต์แวร์ไอพีสวิตช์ส่งเฟรม ARP Request ออกไปแล้ว เครื่องที่มีค่าตำแหน่งแบบไอพีตรงกับค่าที่อยู่ในเฟรม ARP เท่านั้นที่ส่ง ARP Reply ตอบกลับมา ดังในรูป ที่ 8-4



รูปที่ 8-4 แสดงลำดับการทำงานของโปรแกรมในระยะที่ 3 ของการทดสอบ

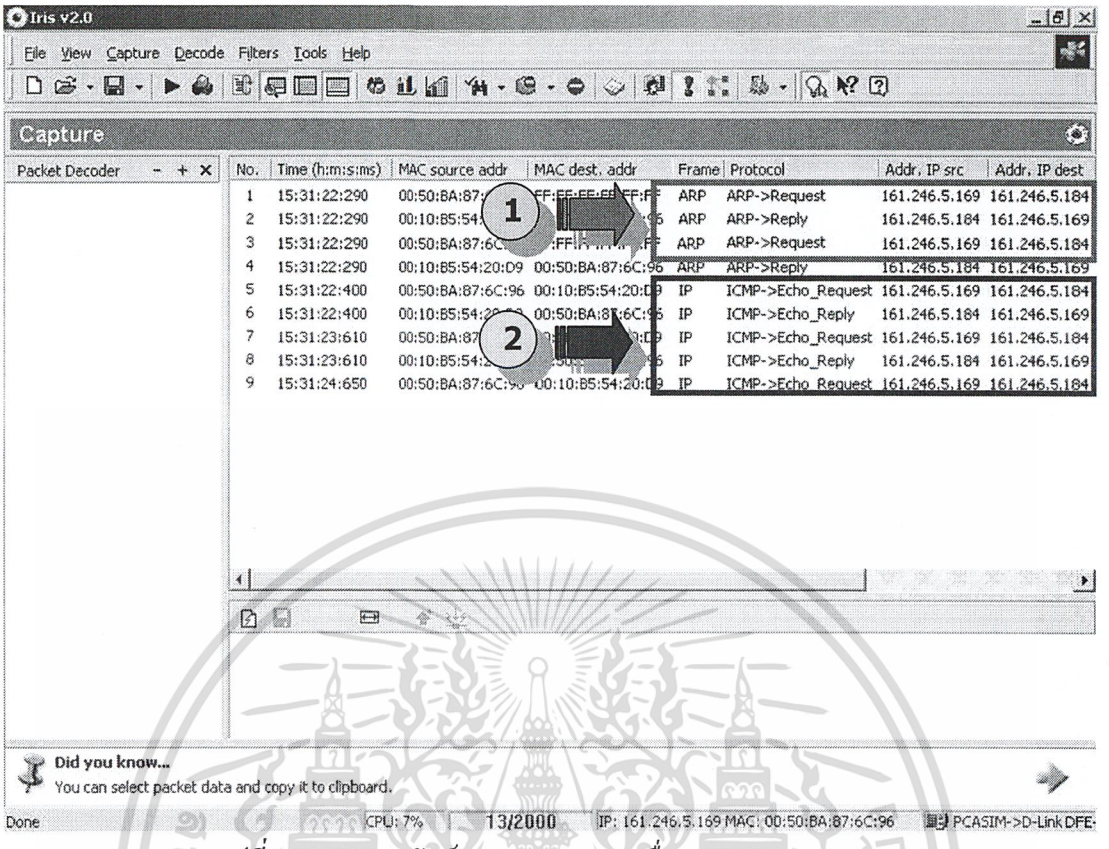
จากนั้น ซอฟต์แวร์ไอพีสวิตช์ก็จะเริ่มทำงานเหมือนเช่นเดิมอีก ก็คือเก็บค่าตำแหน่งที่อยู่แบบแม็คของเครื่องต้นทางจากเฟรมข้อมูล ARP Reply ที่ตอบกลับมายังช่องทางการสื่อสารที่ 2 บันทึกลงในตารางค่าตำแหน่งของช่องทางที่ 2 พร้อมทำการบันทึกค่าตำแหน่งไอพีของเครื่องต้นทางนี้เก็บเอาไว้ในตารางไอพีรวมที่เป็นของส่วน IPCP ด้วย จากนั้นเมื่อมีชุดข้อมูลที่ส่งมาจากเครื่อง A ไปยังเครื่อง C อีก ซอฟต์แวร์ไอพีสวิตช์ก็จะส่งต่อชุดข้อมูลนั้นออกไปได้ทันที อย่างถูกต้อง และไม่ต้องทำการค้นหาเส้นทางในส่วน IPCP อีก ดังในรูปที่ 8-5



รูปที่ 8-5 แสดงลำดับการทำงานของโปรแกรมในระยะที่ 4 ของการทดสอบ

8.3 การตรวจสอบความถูกต้อง

ในการตรวจสอบผลลัพธ์การทำงานของซอฟต์แวร์นั้น จะใช้โปรแกรมดักจับชุดข้อมูลบนระบบเครือข่ายที่มีชื่อว่า IRIS v2.0 มาใช้ Capture แพ็กเก็ตที่วิ่งอยู่จริงบนระบบเครือข่ายจำลองที่สร้างขึ้นมานี้ในช่วงระยะเวลาใดเวลาหนึ่ง เพื่อตรวจสอบว่าการสื่อสารข้อมูลที่เกิดขึ้นนั้น เครื่องศูนย์กลางได้เข้ามาทำหน้าที่ควบคุมทิศทางการเคลื่อนที่ของชุดข้อมูลได้ถูกต้องตามที่ต้องการ และเป็นไปตามหลักการของไอพีสวิตช์ได้อย่างถูกต้องครบถ้วนหรือไม่อย่างไร โดยผลลัพธ์การตรวจสอบที่เกิดขึ้นจะเป็นดังนี้

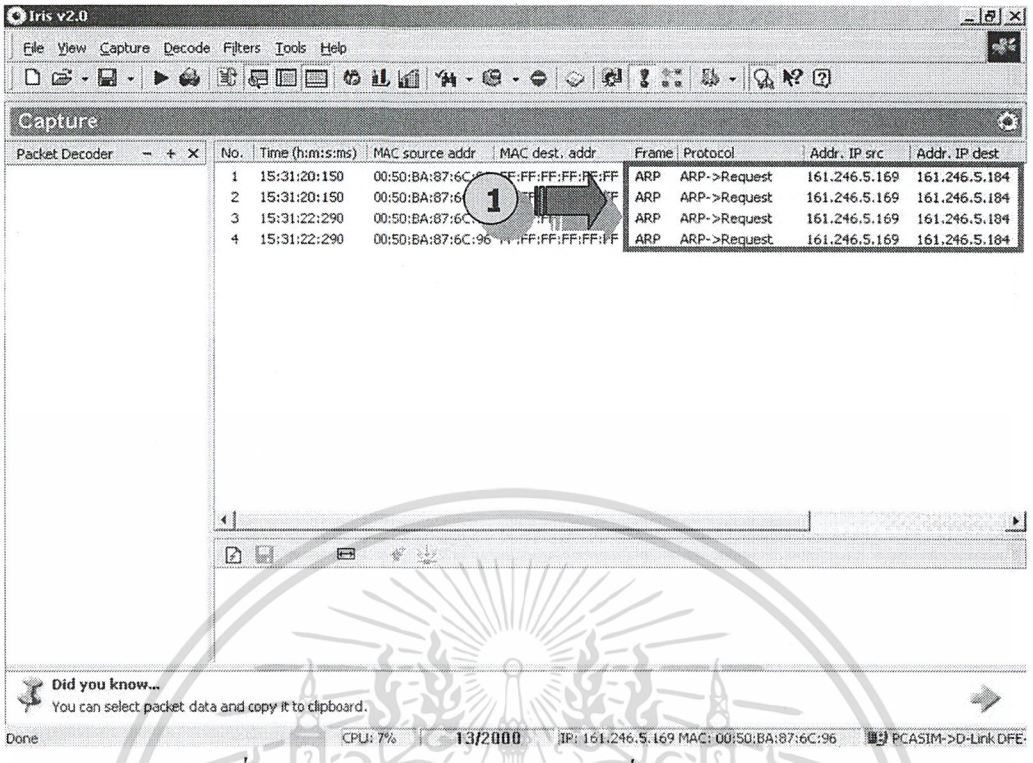


รูปที่ 8-6 แสดงผลลัพธ์การ Capture ณ เครื่อง C (161.246.5.184)

หลังจากที่ซอฟต์แวร์ไอพีสวิตช์ได้ทำการค้นหาค่าตำแหน่งที่อยู่แบบแม็คของเครื่องปลายทางจากทุกช่องทางการสื่อสารเพื่อจะส่งต่อเฟรมข้อมูลไปในระดับดาตาลิงค์ (Layer 2) แต่ก็ไม่พบ เนื่องจากตารางเก็บค่าตำแหน่งในระยะเริ่มต้นยังคงว่างเปล่าอยู่ ดังนั้นจึงต้องค้นหาช่องทางการสื่อสารปลายทางที่ถูกต้องโดยทำการค้นหาในระดับเน็ตเวิร์ก (Layer 3) คือการส่งเฟรม ARP Request ออกไปยังทุกช่องทางการสื่อสาร (ยกเว้นช่องทางที่ชุดข้อมูลผ่านเข้ามา) โดยระบุค่าตำแหน่งเครื่องปลายทางแบบไอพีไว้ในเฟรม ARP ดังกล่าว เมื่อเฟรม ARP Request นี้เดินทางไปถึงยังเครื่องปลายทางที่มีค่าตำแหน่งแบบไอพี ตรงกับที่ระบุไว้ในเฟรม ARP เครื่องปลายทางนั้นก็จะส่ง ARP Reply ตอบกลับมาให้

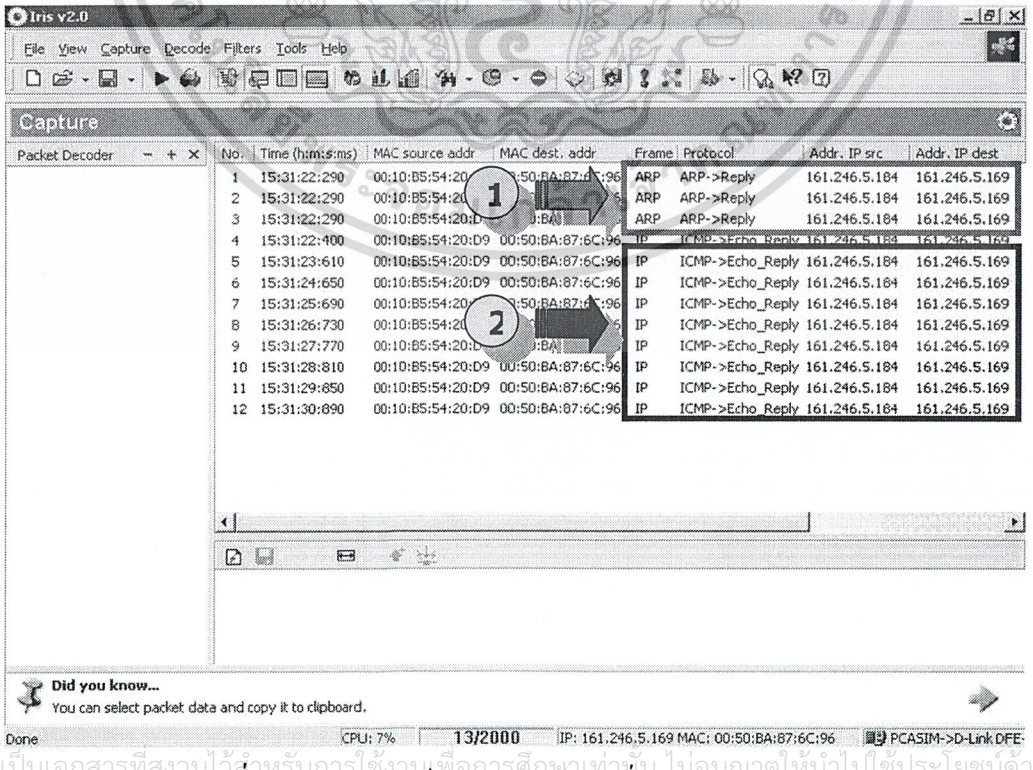
ซึ่งจากการ Capture ที่เครื่อง C (161.246.5.184) ซึ่งเป็นเครื่องปลายทางที่มีค่าตำแหน่งแบบไอพีตรงกับค่าที่ระบุไว้ในเฟรม ARP ที่ส่งมา ภายหลังจากที่ได้รับ ARP REQUEST ที่ส่งมา (หมายเลข 1) เครื่อง C ก็จะตอบ ARP Reply กลับไปด้วย

จากนั้น เมื่อซอฟต์แวร์ไอพีสวิตช์ได้รับ ARP Reply ตอบกลับมาแล้ว ก็จะบันทึกค่าตำแหน่งแบบแม็คของเครื่อง C นี้เก็บลงในตารางของช่องทางการสื่อสารที่ 2 ทันที และเมื่อใดก็ตาม ที่มีการส่งชุดข้อมูลจากเครื่องต้นทาง A ไปยังเครื่องปลายทาง B อีก ซอฟต์แวร์ไอพีสวิตช์ก็จะส่งต่อเฟรมข้อมูลที่ระดับดาตาลิงค์ได้โดยทันที ไม่ต้องทำการค้นหาช่องทางการสารสื่อสารปลายทางในระดับเน็ตเวิร์คอีกเพราะทราบแล้วว่าจะต้องส่งต่อชุดข้อมูลออกไปยังช่องทางใด ทำให้มีการสื่อสารข้อมูลเกิดขึ้นตามมาได้ในที่สุด (หมายเลข 2)



รูปที่ 8-7 แสดงผลลัพธ์การ Capture ณ เครื่อง B (161.246.5.202)

ภาพแสดงการ Capture ที่เครื่อง B (161.246.5.202) ซึ่งเป็นเครื่องปลายทางที่มีค่าตำแหน่งแบบไอพีไม่ตรงกับค่าใน ARP Request ที่ส่งมาจึงไม่ตอบกลับ ARP ดังกล่าว ดังนั้นเมื่อ Capture ดูที่เครื่อง B ดูจึงพบเพียงแต่ ARP Request ที่ส่งออกมายังทุกๆ ช่องทางการสื่อสารในช่วงแรกโดยซอฟต์แวร์ไอพีสวิตช์นั่นเอง (หมายเลข 1)



รูปที่ 8-8 แสดงผลลัพธ์การ Capture ณ เครื่อง A (161.246.5.169)

ภาพแสดงการ Capture ที่เครื่อง A ซึ่งเป็นเครื่องต้นทางที่ส่งชุดข้อมูล แสดงการได้รับ ARP Reply ที่ตอบกลับมาจากเครื่อง C (หมายเลข 1) และ (หมายเลข 2) แสดงการสื่อสารข้อมูลกันได้ภายหลังจากที่ซอฟต์แวร์ไอพีสวิตช์ทราบที่อยู่ของเครื่องปลายทางแล้ว ว่าอยู่ที่ช่องทางการสื่อสารที่ 2

8.4 การตรวจสอบประสิทธิภาพ

ซอฟต์แวร์ไอพีสวิตช์ที่พัฒนาขึ้นมานั้นมีอยู่ด้วยกันหลายรุ่น แต่เมื่อนำมาทำการทดสอบความถูกต้องตามกระบวนการข้างต้นแล้ว มีอยู่เพียง 3 รุ่นที่ทำงานได้ถูกต้องตามที่ได้แสดงให้เห็นในตัวอย่าง คือรุ่นที่สามารถรองรับการทำงานได้ 2 ช่องทางการสื่อสารเท่านั้น กับรุ่นที่สามารถรองรับการทำงานได้ 3 ช่องทางการสื่อสาร และรุ่นที่สามารถรองรับการทำงานกับระบบที่มีมากกว่า 3 ช่องทางการสื่อสารขึ้นไป โดยความแตกต่างระหว่างซอฟต์แวร์ทั้ง 2 รุ่นนี้ ก็ทำให้ประสิทธิภาพของการทำงานของซอฟต์แวร์แต่ละรุ่นแตกต่างกันอย่างเห็นได้ชัดคือ ในรุ่นที่สามารถรองรับได้ 2 ช่องทางนั้น เครื่องปลายทางต่างๆ สามารถใช้คำสั่ง ping ติดต่อถึงกันได้ไม่ติดขัดแต่อย่างใด ส่วนรุ่นที่รองรับได้ตั้งแต่ 3 ช่องทางขึ้นไปนั้น เมื่อใช้คำสั่ง ping แล้ว จะปรากฏ Error Message “Request timed out” บ้างในบางครั้ง ส่วนรุ่นสุดท้ายนั้นจะปรากฏ Error Message “Request timed out” เกือบตลอดเวลา เนื่องจากตัวซอฟต์แวร์ที่ทำงานอยู่บนเครื่องศูนย์กลางนั้นไม่สามารถทำงานกับระบบที่มีช่องทางการสื่อสารมากๆ ได้ทันนั่นเอง

แต่อย่างไรก็ตาม ซอฟต์แวร์ทั้งสามรุ่นนี้ก็ยังมีความสามารถเพียงพอที่จะให้เครื่องลูกข่ายสามารถใช้งานอินเตอร์เน็ตได้ตามปกติ ไม่ว่าจะเป็นการใช้งานโปรแกรมเว็บบราวเซอร์, FTP, Telnet, และอื่นๆ เชื่อมต่อไปยังเครื่องที่อยู่ภายนอกโดยผ่านเครื่องศูนย์กลางได้อย่างไม่มีปัญหา

บทที่ 9

บทวิจารณ์และสรุป

9.1 วิจารณ์โครงงาน

จากบันทึกขั้นตอนการดำเนินงานในภาคผนวก ข นั้น แสดงให้เห็นได้อย่างชัดเจนว่าเทคโนโลยีตัวนี้ เป็นเทคโนโลยีที่ค่อนข้างใหม่ ยังคงมีการพัฒนาเกิดขึ้นอยู่อย่างต่อเนื่องและรวดเร็ว อีกทั้งยังเป็นนวัตกรรมที่มีอิทธิพลต่อองค์กรทางธุรกิจต่างๆ ที่มีความเกี่ยวข้องอีกมากมาย มีผลต่อการแข่งขันทางการค้ามากกว่าการวิจัย ทำให้แหล่งข้อมูลที่ทันต่อเหตุการณ์นั้นหาได้ยาก ส่วนมากจะเป็นแหล่งข้อมูลที่อยู่ในช่วงแรกๆ ของการเปิดตัวเทคโนโลยีเท่านั้น ซึ่ง ณ วันนี้ก็ได้มีการเปลี่ยนแปลงไปมากมาย ทำให้ข้อมูลที่ได้มานั้นมักไม่ค่อยทันสมัย

โครงงานนี้จึงมีความยากอยู่ที่การศึกษาค้นคว้า และรวบรวมข้อมูลเกี่ยวข้องกับตัวทฤษฎีของระบบสวิตช์ และไอพีสวิตช์ ในช่วงต้นของการดำเนินงาน อีกทั้งยังเสียเวลาไปค่อนข้างมากในขั้นตอนนี้อีกด้วย ทำให้ในช่วงที่ลงมือปฏิบัติจริงนั้น เหลือเวลาอยู่เพียง 1-2 เดือน เท่านั้น ซึ่งก็แทบจะไม่สามารถทำอะไรได้เลย ส่วนที่ง่ายที่สุดของโครงงานนี้ก็คือขั้นตอนการออกแบบระบบซอฟต์แวร์ไอพีสวิตช์ เนื่องจากระบบไอพีสวิตช์นั้น เน้นความรวดเร็วในการทำงาน ดังนั้นโครงสร้างของอัลกอริทึมต่างๆ ที่เกี่ยวข้องนั้น จึงมักจะมีรูปแบบที่ง่าย ตรงไปตรงมา และไม่มีความซับซ้อนมากนัก แต่นอกจากนี้แล้วในขั้นตอนการพัฒนาซอฟต์แวร์ขึ้นมาจริงๆ นั้น ก็มีความยากอย่างมากอยู่ตรงที่ ซอฟต์แวร์ที่พัฒนาขึ้นมาจะต้องมีความสามารถในการเข้าถึง และควบคุมการดำเนินงานของแผงวงจรสื่อสารเครือข่าย (NICs) ในระดับลึก ซึ่งไม่สามารถใช้คำสั่งทั่วไปได้ เราจำเป็นจะต้องเข้าถึงการทำงานในระดับ Kernel ของระบบปฏิบัติการ เพื่อส่งผ่านการควบคุมไปยังฮาร์ดแวร์ของระบบโดยตรงให้ได้ ซึ่งในการพัฒนาซอฟต์แวร์บนระบบปฏิบัติการลีนุกซ์นั้น แม้ว่าจะยอมให้มีการเข้าถึงส่วนที่เป็น Kernel ได้อย่างง่ายดาย และมีข้อมูลให้ศึกษาได้มากกว่า แต่ในส่วนของการเขียนโปรแกรมนั้น ณ ช่วงเวลาที่กำลังดำเนินโครงงานอยู่นี้ การพัฒนาซอฟต์แวร์บนระบบปฏิบัติการลีนุกซ์ ก็ยังคงมีความคล่องตัวต่ำกว่าการพัฒนาซอฟต์แวร์บนระบบปฏิบัติการวิโดวส์อยู่มาก ซึ่งอาจมีสาเหตุมาจากการที่คณะผู้จัดทำยังไม่มีความคุ้นเคยกับการพัฒนาซอฟต์แวร์บนระบบปฏิบัติการลีนุกซ์ก็เป็นได้ ทำให้ไม่สามารถทำอะไรได้ตามต้องการ จึงต้องหันมาเลือกที่จะพัฒนาซอฟต์แวร์บนระบบปฏิบัติการวินโดวส์ที่คุ้นเคยแทน แต่อย่างไรก็ตาม เนื่องจากระบบปฏิบัติการวินโดวส์เป็นระบบปฏิบัติการในเชิงธุรกิจ จึงไม่ได้มีความเป็นระบบเปิด (Open System) เหมือนอย่างที่ระบบปฏิบัติการลีนุกซ์เป็น ข้อมูลที่เกี่ยวข้องกับการทำงานยากๆ ในระดับลึกจริงๆ จึงไม่ค่อยมีออกมาเผยแพร่ให้ศึกษา เพราะมักกลายเป็นเชิงพาณิชย์ไปเสียหมด แต่ก็ยังพอมีให้พบได้บ้างอย่าง เช่นตัว WinPcap ซึ่งเข้ามามีส่วนช่วยให้การเขียนโปรแกรมในระดับล่างเพื่อติดต่อกับแผงวงจรสื่อสารเครือข่ายนั้นทำได้ง่ายขึ้น โดยไม่จำเป็นต้องเรียนรู้ทำความเข้าใจกับ Function พื้นฐานทั้งหมดที่มีอยู่ในระบบแต่อย่างใด อาศัยเพียงความเข้าใจที่ชัดเจนในลำดับการทำงานของข้อตกลงการสื่อสารข้อมูลก็พอ

9.2 บทสรุปโครงงาน

สำหรับโครงงานนี้ แม้ว่าจะสามารถพัฒนาซอฟต์แวร์จำลองการทำงานของระบบไอพีสวิตช์ขึ้นมาได้อย่างถูกต้องตามหลักการที่เกี่ยวข้อง และสามารถนำมาใช้งานได้จริงแล้วก็ตาม แต่ก็ยังติดปัญหาในเรื่องประสิทธิภาพของซอฟต์แวร์ในระดับการ Coding อยู่อีกมากนั่นเอง อันเนื่องมาจากสาเหตุต่างๆ ดังต่อไปนี้

1. ในขั้นตอนของการตรวจสอบหาจำนวนและตำแหน่งของแผงวงจรสื่อสารเครือข่ายที่ติดตั้งอยู่ในระบบนั้น ในรุ่นนี้ยังคงเป็นการตรวจสอบผ่าน Registry ของวินโดวส์อยู่ ทำให้ไม่สามารถตรวจสอบสภาวะของระบบที่เป็นอยู่จริงในขณะนั้นๆ ได้ ทำให้ในบางครั้งการเริ่มต้นทำงานของระบบไอพีสวิตช์ซอฟต์แวร์ เกิดความผิดพลาดขึ้นมา เช่นตำแหน่งแผงวงจรที่ตรวจพบ และพยายามที่จะเข้าไปจัดการนั้นไม่ได้มีอยู่จริงระบบขณะนั้นนั่นเอง

2. ซอฟต์แวร์ที่เขียนขึ้นมานี้ ใช้การตรวจสอบสถานะของแผงวงจรสื่อสารเครือข่าย และตอบสนองในแบบ “Polling” ไม่ได้มีการใช้งาน “Thread” แต่อย่างใด ซึ่งในความเป็นจริงแล้ว ซอฟต์แวร์จัดการเครือข่ายทั้งหลาย ควรจะต้องถูกเขียนขึ้นมาให้เป็นแนว Thread และ Object Oriented จึงจะสามารถทำงานได้อย่างเป็นอิสระต่อกัน และทำให้สามารถตอบสนองได้ทัน ต่อเหตุการณ์ต่างๆ ในกรณีที่มีแผงวงจรสื่อสารเครือข่ายอยู่ในระบบมากกว่า 3 ใบอีกด้วย ดังที่ได้กล่าวไว้ในบทที่ 8 เกี่ยวกับการทดสอบประสิทธิภาพนั่นเอง

3. ส่วนการทำงานในระดับของ IPCP ที่ใช้หลักการค้นหาเส้นทางแบบ IP Routing นั้น แม้ว่าจะสามารถทำงานในระบบเครือข่ายท้องถิ่นจำลองที่มีขนาดเล็กได้แล้ว แต่ก็ยังไม่ได้นำเอาหลักการของ IP Routing เข้ามาใช้อย่างเต็มรูปแบบ ทำให้การทำงานในส่วนนี้ไม่ค่อยมีบทบาทที่ชัดเจนต่อระบบ ตามแนวคิดของไอพีสวิตช์ รวมทั้งรูปแบบโครงสร้างตารางจัดเก็บค่าตำแหน่งที่อยู่ก็ยังไม่สามารถรองรับการทำงานกับเครือข่ายที่มีขนาดใหญ่ได้อีกด้วย

9.3 ข้อควรปรับปรุง

จากข้อบกพร่องที่กล่าวไปทั้งหมดนี้ หากผู้ที่มีความสนใจที่จะนำไปพัฒนาต่อ สามารถแก้ไขข้อบกพร่องเหล่านี้ได้ จะช่วยให้ประสิทธิภาพของซอฟต์แวร์สูงขึ้นจนคาดว่าน่าจะสามารถในไปใช้งานในระบบเครือข่ายที่มีขนาดใหญ่ขึ้นได้ แต่อย่างไรก็ตาม ระบบนี้เป็นเพียงการจำลองการทำงานของอุปกรณ์ไอพีสวิตช์ขึ้นมาเท่านั้น การที่จะนำไปใช้จริงนั้น หากสามารถสร้างเป็นระบบบนตัวอุปกรณ์ฮาร์ดแวร์ได้ จึงจะได้ประสิทธิภาพเต็มที่ และสามารถเห็นผลความแตกต่างจากเราเตอร์ได้อย่างชัดเจนตามที่ได้กล่าวไว้ในส่วนของทฤษฎีไอพีสวิตช์ ในบทที่ 5 นั่นเอง ซึ่งก็ได้มีผลการพิสูจน์จากหลายๆ สถาบันในต่างประเทศออกมายืนยันให้เห็นอยู่อีกมากมายอีกด้วย

ภาคผนวก ก

รูปแบบเฮดเดอร์ของโพรโตคอลต่างๆ (Protocol Header Formats)

Ethernet II Frame

Destination Address	Source Address	Type	Data
6 bytes	6 bytes	2 bytes	46-1500 bytes

Ethernet 802.3 Frame

Destination Address	Source Address	Length	Data (เริ่มต้นด้วย OxFFFF)
6 bytes	6 bytes	2 bytes	46-1500 bytes

Ethernet 802.2 Frame

Destination Address	Source Address	Length	DSAP	SSAP	Control	Data
6 bytes	6 bytes	2 bytes	1 byte	1 byte	1 byte	43-1497 bytes

Ethernet SNAP Frame

Destination Address	Source Address	Length	DSAP	SSAP	Control	Organization Code	Ethernet Type	Data
6 bytes	6 bytes	2 bytes	1 byte	1 byte	1 byte	3 bytes	2 bytes	38-1492 bytes

IPX Packet

Check Sum	Length	Transport	Packet Type	Destination Network	Destination Host	Destination Socket	Source Network	Source Host	Source Socket	Data
2 bytes	2 bytes	1 byte	1 byte	4 bytes	6 bytes	2 bytes	4 bytes	6 bytes	2 bytes	

IP Packet

Version+Header Length	Type of Service	Length	Identifier	Flag	Fragment Offset	Time to live	Protocol	Header Check Sum	Source Address	Destination Address	Option	Data
1 byte (1/2+1/2)	1 byte	2 bytes	2 bytes	3 bits	13 bits	1 byte	1 byte	2 bytes	4 bytes	4 bytes	variable	

ARP Packet

Hardware Type	Protocol Type	Hardware Address Length	Protocol Address Length	Operation Code	Send Hardware Address	Send Protocol Address	Target Hardware Address	Target Protocol Address
2 bytes	2 bytes	1 byte	1 byte	2 bytes	6 bytes	4 bytes	6 bytes	4 bytes

TCP Packet

Source Port	Destination Port	Sequence Number	Acknowledge-Ment Number	Data offset	Rese-rved	Code	Windows	Check Sum	Urgent Pointer	Option	Data
2 bytes	2 bytes	4 bytes	4 bytes	4 bytes	6 bits	6 bits	2 bytes	2 bytes	2 bytes	variable	

UDP Packet

Source Port	Destination Port	Length	Check Sum	Data
2 bytes	2 bytes	2 bytes	2 bytes	

SPX Packet

Control	Data Type	Source ID	Destination ID	Sequence Number	Acknowledge-Ment Number	Allocation Number	Data
1 byte	1 byte	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	

ICMP Packet

Type	Code	Check Sum	Data
1 byte	1 byte	2 bytes	

Echo and Echo Reply Message

Type	Code	Check Sum	Identifier	Sequence Number	Data
1 byte (0 or 8)	1 byte	2 bytes	2 bytes	2 bytes	

Information and Information Reply Message

Type	Code	Check Sum	Identifier	Sequence Number
1 byte (15 or 16)	1 byte	2 bytes	2 bytes	2 bytes

Destination Unreachable Message

Source Quench Message

Time Exceeded Message

Type	Code	Check Sum	Unused	Data
1 byte	1 byte	2 bytes	4 bytes	

ICMP Message Type

ชนิด	ความหมาย
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
	Echo
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

Redirect Message

Type	Code	Check Sum	Gateway Internet Address	Data
1 byte	1 byte	2 bytes	4 bytes	

(5)

Parameter Problem Message

Type	Code	Check Sum	Pointer	Unused	Data
1 byte	1 byte	2 bytes	1 byte	3 bytes	

(12)

Timestamp and Timestamp Reply Message

Type	Code	Check Sum	Original Timestamp	Receive Timestamp	Transmit Timestamp
1 byte	1 byte	2 bytes	4 bytes	4 bytes	4 bytes

(13 or 14)

ภาคผนวก ข

บันทึกขั้นตอนการดำเนินงาน

ในช่วงภาคเรียนแรกของการทำโครงงานนี้(ส.ค.) คณะผู้จัดทำได้เริ่มต้นค้นคว้าเกี่ยวกับเทคโนโลยี IP Switching ก่อน โดยเริ่มจากการค้นคว้าต้นกำเนิดและที่มาของเทคโนโลยีนี้ ทั้งจากห้องสมุดภายในคณะวิศวกรรมศาสตร์และสำนักหอสมุดกลาง แต่ไม่พบจึงต้องเปลี่ยนไปหาข้อมูลจาก Internet แทน ทำให้พบว่าเทคโนโลยี IP Switching นี้ จริงๆแล้วเป็นแนวคิดทางด้านการจัดการการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์ที่ค่อนข้างใหม่ ซึ่งถือกำเนิดขึ้นในปี ค.ศ.1996 โดยการคิดค้นพัฒนาของบริษัท Ipsilon Networks, Inc. (www.ipsilon.com) ที่ตั้งอยู่ที่ Sunnyvale, California ประเทศสหรัฐอเมริกา ซึ่งในปัจจุบันได้ถูกรวมเข้ามาเป็นส่วนหนึ่งของบริษัท NOKIA Telecommunications ประเทศฟินแลนด์ ตั้งแต่วันที่ 9 ธันวาคม ปี ค.ศ. 1997 ทำให้ข้อมูลต่างๆ ทั้งหมด รวมทั้ง Website ของบริษัท Ipsilon ที่เคยเผยแพร่เกี่ยวกับแนวคิดและทฤษฎีพื้นฐานของ IP Switching ถูกปิดและกันเป็นความลับทางธุรกิจของบริษัท NOKIA (http://www.nokia.com/securitysolutions) ไปในที่สุด จึงไม่สามารถหาข้อมูลเพิ่มเติมจาก Website นี้ได้ และต้องไปค้นหาข้อมูลตามมหาวิทยาลัยในต่างประเทศทดแทน จนกระทั่งได้ข้อมูลเกี่ยวกับการทำงานของเทคโนโลยี IP Switch ชุดแรกมาจาก RFCXXXX ของมหาวิทยาลัยที่มีชื่อว่า “Computer and Information Science” (www.cis.ohio-state.edu) ในเดือนกันยายน พ.ศ. 2543 ซึ่งเป็นข้อมูลในเรื่องแนวคิดพื้นฐาน ที่มาของการพัฒนาแนวคิดนี้ขึ้นมา รวมทั้ง Protocol ต่างๆ ที่ใช้ร่วมกับเทคโนโลยีนี้ด้วยคือ IFMP (RFC1953) และ GSMP (RFC1987)

จากนั้น ก็ได้ติดต่อไปยังบริษัท TelecomASAI ซึ่งผู้จัดทำได้เคยไปฝึกงานมาเมื่อครั้งเรียนอยู่ชั้นปีที่ 3 จึงได้ทราบว่า ที่บริษัทนี้เคยมีการติดต่อกับบริษัทคู่ค้ารายอื่นที่เกี่ยวกับเทคโนโลยีตัวนี้มาก่อนแล้ว เพื่อขอรายละเอียดเกี่ยวกับบริษัทเหล่านั้น ซึ่งก็ได้รับความช่วยเหลือด้วยดี และได้รับคำแนะนำให้ติดต่อไปยังบริษัท CISCO Systems (Thailand) ทำให้ได้ข้อมูลเกี่ยวกับผลิตภัณฑ์ รวมทั้งคุณสมบัติที่ทันสมัยเกี่ยวกับเทคโนโลยีนี้มาอย่างมากมาย แต่อย่างไรก็ตาม ข้อมูลที่ได้นั้นก็เป็นเพียงรายละเอียดของสินค้าเท่านั้น ไม่มีรายละเอียดทางเทคนิคแต่อย่างใด เนื่องจากสาขาในประเทศไทยของบริษัท CISCO นั้นเป็นเพียงตัวแทนจำหน่าย จึงไม่สามารถจัดหาข้อมูลทางเทคนิคให้ได้

เนื่องจากเทคโนโลยีนี้ยังค่อนข้างใหม่อยู่ ทำให้ข้อมูลมีอยู่น้อยและหาได้ยากมาก การรายงานสรุปและนำเสนอโครงงานในภาคเรียนแรก จึงยึดตามหลักการไอพีสวิตช์แบบดั้งเดิมของบริษัท Ipsilon คือทำ IP Routing บน ATM Switch เป็นหลัก

ต่อมาในวันที่ 18 สิงหาคม 2543 สำนักหอสมุดกลาง สจ.ล.ได้สั่งหนังสือชื่อ IP Switching Protocols and Architectures ของ McGraw-Hill เข้ามา และอณุญาตให้ใช้หนังสือได้ในวันที่ 27 พฤษจิกายน 2543 ทำให้ได้ข้อมูลเพิ่มเติมขึ้นมาอีกมาก และทำให้ทราบว่าตามความเป็นจริงแล้วเทคโนโลยีไอพีสวิตช์ไม่จำเป็นต้องทำงานบน ATM Fabric เท่านั้น ยังสามารถทำงานบนระบบเครือข่าย

แบบอื่นๆได้อีกด้วย ระบบ ATM เป็นเพียงระบบที่ทำให้ไอพีสวิตช์ทำงานได้อย่างมีประสิทธิภาพและได้รับความนิยมสูงสุดในการนำมาใช้งานร่วมกับเทคโนโลยีไอพีสวิตช์เท่านั้น

ระยะต่อมา ในช่วงธันวาคม 2543 ได้ศึกษาเพิ่มเติมเกี่ยวกับการทำงานของ Protocol "IFMP" และ "GSMP" โดยอ้างอิงข้อมูลจาก Website www.protocols.com ร่วมกับทฤษฎีตามหนังสือ IP Switch ของสำนักหอสมุดกลาง เพื่อให้เกิดความเข้าใจเกี่ยวกับการทำงานของ IP Switch ในระดับ Algorithm อย่างชัดเจน

ในช่วงต้นเดือนมกราคม 2544 ได้เริ่มออกแบบซอฟต์แวร์ต้นแบบเพื่อใช้ทดลองศึกษาและวัดผลประสิทธิภาพในระดับอัลกอริทึม โดยเปรียบเทียบกับการทำงานตามอัลกอริทึมของเราเตอร์ทั่วๆ ไป หลังจากออกแบบตัวซอฟต์แวร์เสร็จในช่วงกลางเดือนมกราคม จึงเริ่มต้นเขียนซอฟต์แวร์ขึ้นมา แต่เนื่องจากผู้จัดทำโครงงานบางคนมีความคุ้นเคยกับการพัฒนาซอฟต์แวร์บนวินโดวส์ ส่วนบางคนมีความถนัดทางด้านลีนุกซ์มากกว่าจึงได้แยกกันไปศึกษาแนวทางการ Coding ในทั้ง 2 แพลตฟอร์มควบคู่กันไป แต่เมื่อผ่านไประยะหนึ่งก็ได้ข้อสรุปว่าการพัฒนาซอฟต์แวร์บนวินโดวส์มีความสะดวกมากกว่าจึงกลับมาร่วมกันศึกษาการ Coding บนวินโดวส์ทั้งหมด

ทางด้านการพัฒนาซอฟต์แวร์บนวินโดวส์นั้น การศึกษาได้เริ่มต้นจากการศึกษาวิธีการเขียนซอฟต์แวร์ติดต่อควบคุมการทำงานของแผงวงจรสื่อสารเครือข่ายผ่านระบบปฏิบัติการวินโดวส์ด้วย โดยเริ่มหาข้อมูลจาก MSDN ก่อน แต่ปรากฏว่าข้อมูลที่ MSDN มีให้นั้นไม่สามารถช่วยอะไรได้มากนัก เนื่องจากไม่ค่อยมีข้อมูลที่ลึกเพียงพอ ส่วนข้อมูลในระดับลึกจริงๆ ที่หาได้นั้นก็ไม่ละเอียดพอที่จะนำมาใช้งานจริงได้

จึงติดต่อไปยังบริษัท เพื่อขอทราบแหล่งข้อมูลเพิ่มเติม ซึ่งทางบริษัท Microsoft ก็ได้แนะนำให้ติดต่อไปที่บริษัท Isonet อีกที่

ติดต่อขอคำแนะนำเกี่ยวกับแพลตฟอร์มและการ Coding ไปยังบริษัทไมโครซอฟต์ประเทศไทยทั้งทางโทรศัพท์และ E-Mail แต่ไม่ได้รับการติดต่อกลับมาในวันที่ 20 มกราคม 2544 จึงได้เดินทางไปยังบริษัทโดยตรง จึงได้รับคำแนะนำมาว่าควรใช้ Visual C++ บน Windows 2000 Server จากนั้นก็ได้รับการแนะนำให้ติดต่อขอรายละเอียดเพิ่มเติมไปที่บริษัท ISONET อีกที่หนึ่ง ซึ่งเมื่อติดต่อไปที่บริษัท Isonet แล้ว จึงได้ทราบว่าซอฟต์แวร์ที่ต้องการจะพัฒนาขึ้นมานี้ ควรเริ่มศึกษาในระดับ "Sock_Raw" ก่อน จึงสามารถเริ่มทำการศึกษาได้อย่างชัดเจนเรื่อยมาจากนั้น และได้ข้อมูลต่างๆ เพิ่มขึ้นอีกมากมายเป็นลำดับ โดยสรุปได้ว่า ส่วนการทำงานของซอฟต์แวร์นั้นจะมีอยู่ด้วยกันหลายระดับ ได้แก่ การใช้ WSAxxx Functions ของ Winsock2 ทำการเชื่อมต่อผ่าน Sock_Raw เพื่อทำการรับส่งข้อมูลระหว่าง End-Point ต่างๆ การใช้ WSPxxx Functions ในการติดต่อกับ Protocols ระดับล่าง และใช้ LLCxxx Functions สำหรับควบคุมการรับ/ส่งเฟรมข้อมูลของ NICs ส่วนในระดับของการจัดการเส้นทางของเครื่องที่ทำงานเป็น Server จำลองการทำงานของ IP Switch นั้น ส่วนของการ Route ก็จะใช้ RTMxxx Functions ในการจัดการตาราง Routing Table และสำหรับการควบคุมการทำงานของ NICs โดยตรงนั้นจะต้องใช้ NDISxxx Function เป็นต้น ซึ่ง Function Sets ต่างๆ เหล่านี้ แต่ละกลุ่มก็มี Function ให้เลือกใช้มากมาย และในการเรียกใช้ Function แต่ละตัวก็มีเงื่อนไขมากมายจนคาดว่าไม่น่าจะศึกษาได้ทัน จึงได้นำ

แพ็กเก็ตดรเวอร์และซอฟต์แวร์ช่วยเขียนโปรแกรมเครือข่ายที่มีชื่อว่า WinPcap เข้ามาช่วย ซึ่งช่วยให้การเขียนโปรแกรมเครือข่ายทำได้ง่ายขึ้นเนื่องจากตัว WinPcap จะรวมกลุ่มการทำงานของ Function ต่างๆ เข้าไว้ด้วยกันเป็นส่วนการทำงานย่อยต่างๆ ให้เลือกใช้ ซึ่งเมื่อเราเรียกใช้ Function ใดใน WinPcap มันก็จะไปเรียกใช้ Function พื้นฐานต่างๆ ของระบบเช่น NDISxxx ในลำดับที่ถูกต้องให้เองอีกที ดังที่กล่าวไว้แล้วในบทที่ 6 นั่นเอง เราจึงสามารถเขียนซอฟต์แวร์ขึ้นมาได้ทันในที่สุด โดยตัว WinPcap นี้สามารถหาได้จาก URL นี้

http://netgroup-serv.polito.it/winpcap/install/default.htm

ส่วนข้อมูลเพิ่มเติมสามารถหาได้จาก URL นี้

http://netgroup-serv.polito.it/winpcap/docs/default.htm

บรรณานุกรม

หนังสืออ้างอิง

[1] Pat Bonner (Bonner,Pat) : "*Network Programming with windows*" Prentice Hall PTR

[2] Authony Jones and Jim Ohlund : " *Network Programming for Microsoft Windows*" QA 76.625 J65 1999,QA 76.76 063 N47 1997,TK S105.35.M38 1998

[3] Daniel Minoli ,Andrew Schmidt : "*Network Layer Switched Services*" New York : Wiley C1998,pp. 81-148, 269-300

[4] Alok K. Sinha Reading : "*Network Programming in Windows NT*" Addison-Wesley Pub C1996

[5] Anthony Jones,Jim Ohland Redmond : "*Network Programming for Microsoft Windows*" Wash Microsoft Press C1999,pp. 59-69, 117-558

[6] John D. Ruley : "*Networking Windows NT 4.0*" New York Wiley C1997,pp.509-599

โฮมเพจอ้างอิง

[1] **www.microsoft.com** : *"Basic Networking & Protocols"* http://msdn.microsoft.com/workshop/networking/default.asp , http://msdn.microsoft.com/downloads/sdks/platform/tpipv6.asp

[2] **www.protocols.com** : *"TCP/IP v4 & v6"* http://www.protocols.com/pbook/tcpip.htm

[3] **www.protocols.com** : *"ARP"* http://www.protocols.com/pbook/tcpip1.htm#ARP

[4] **www.protocols.com** : *"ICMP"* http://www.cis.ohio-state.edu/htbin/rfc/rfc826.html

[5] **www.protocols.com** : *"ICMP"* http://www.protocols.com/pbook/tcpip1.htm#ICMP , http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc792.html , http://codeguru.earthweb.com/network/cicmp.shtml#Overview

[6] **www.protocols.com** : *"RIP2"* http://www.protocols.com/pbook/tcpip2.htm#RIP2 , http://www.cis.ohio-state.edu/htbin/rfc/rfc1058.html

[7] **www.protocols.com** : *"RIPng(IPv6)"* http://www.protocols.com/pbook/tcpip2.htm#RIPng , http://www.cis.ohio-state.edu/htbin/rfc/rfc2080.html

[8] **www.protocols.com** : *"BGP-4"* http://www.protocols.com/pbook/tcpip3.htm#BGP-4 , http://www.cis.ohio-state.edu/htbin/rfc/rfc1654.html

[9] **www.protocols.com** : *"NHRP"* http://www.protocols.com/pbook/tcpip4.htm#NHRP , http://www.cis.ohio-state.edu/htbin/rfc/rfc2332.html

[10] **www.protocols.com** : *"OSPF"* http://www.protocols.com/pbook/tcpip4.htm#OSPF , http://www.cis.ohio-state.edu/htbin/rfc/rfc1583.html

[11] **www.protocols.com** : *"ISO (OSI)"* http://www.protocols.com/pbook/iso.htm

[12] **www.protocols.com** : *"IP Switching (Ipsilon)"* http://www.protocols.com/pbook/ip.htm

[13] **www.cis.ohio-state.edu** : *"IFMP version 1.0 for IPv4"* http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1953.html

[14] **www.cis.ohio-state.edu** : *"GSMP version 1.1"* http://www.cis.ohio-state.edu/htbin/rfc/rfc1987.html

[15] **www.ietf.org** : *"Tag Switching (CISCO) TDP MPLS"* http://www.ietf.org/ids.by.wg/mpls.html

[16] **msdn.microsoft.com** : *"Network Programing Guide for Windows 2000 Microsoft Library"* http://msdn.microsoft.com/library/default.asp

[17] **msdn.microsoft.com** : *"Microsoft Platform SDK"* http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/sdkstrt/aboutsdk.htm , http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/sdkintro/mainport.htm , http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/rras/rrasport_6r6t.htm , http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/winsock/apistart_9g1e.htm

[18] **msdn.microsoft.com** : *"Microsoft Device Driver Kit"* http://msdn.microsoft.com/library/default.asp?URL=/library/ddkdoc/ddkstrt/aboutddk.htm

[19] **communities.microsoft.com** : *"Microsoft Newsgroup"* http://communities.microsoft.com/newsgroups/

[20] *"Windows Socket 2.0 Programming"* http://www.ece.wpi.edu/courses/ee535/hwk97/hwk4cd97/murti/node10.html , http://www.inficad.com/~evel/jgaa/winsock.html , http://www.sockets.com/winsock2.htm , http://www.stardust.com/winsock/ws_src.htm#WS2SPs

[21] **msdn.microsoft.com** : *"Visual Studio & Visual C++ User Guide"* http://msdn.microsoft.com/code/ , http://msdn.microsoft.com/vstudio/ , http://msdn.microsoft.com/library/default.asp?URL=/library/devprods/vs6/visualc/vcedit/vcstartpage.htm

[22] **www.codeproject.com** : *"Visual C++ Programming Guide"* http://www.codeproject.com/cpp/ , http://www.codeproject.com/shell/ , http://www.codeproject.com/threads/ , http://www.codeproject.com/internet/ ,

[23] **www.cwinapp.com** : *"Visual C++ Programming Guide"* http://www.cwinapp.com/tutorials/

[24] **www.pcusa.com** : *"Network Driver Interface Specification (NDIS) Frequently Asked Questions"*
http://www.pcausa.com/resources/ndisfaq.htm

[25] **www.codeguru.com** : *"Network Programming"*
http://www.codeguru.com/network/index.shtml

[26] **code.jacobbieling.de** : *"Networking v1.0a Documentation"*
http://code.jakobbieling.de/Networking-Help.html

[27] **http://netgroup-serv.polito.it** : *"WinPcap: the Free Packet Capture Architecture for Windows"*
http://netgroup-serv.polito.it/winpcap/,
http://netgroup-serv.polito.it/winpcap/install/default.htm,
http://netgroup-serv.polito.it/winpcap/docs/default.htm