การตรวจจับแพ็กเก็ตข้อมูลและการประยุกต์ใช้งาน บนระบบ ทีซีพี/ไอพี

DATA PACKET DETECTION ON TCP/IP AND ITS APPLICATIONS

นภัทร สระเอี่ยม

NAPAT SRA-IUM

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2541

ISBN 974-622-122-1

การตรวจจับแพ็กเก็ตข้อมูลและการประยุกต์ใช้งานบนระบบทีซีพี/ไอพี

DATA PACKET DETECTION ON TCP/IP AND ITS APPLICATIONS

นภัทร สระเอี่ยม

NAPAT SRA-IUM

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2541

ISBN 974-622-122-1

DATA PACKET DETECTION ON TCP/IP AND ITS APPLICATIONS

NAPAT SRA-IUM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1998

ISBN 974-622-122-1

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การตรวจจับแพ็กเก็ตข้อมูลและการประยุกต์ใช้งานบนระบบทีซีพี/ไอพี

DATA PACKET DETECTION ON TCP/IP AND ITS APPLICATIONS

ชื่อนักศึกษา นายนภัทร สระเอี่ยม **รหัสประจำตัว** 38061247

หลักสูตร วิศวกรรมศาสตรมหาบัณฑิต **สาขาวิชา** วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.กอบชัย เดชหาญ

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.กอบชัย	เดชหาญ	
รศ.ดร.ฟูศักดิ์	ชีวสุวิทย์	
รศ.ดร.ถวิล	พึ่งมา	
ผศ.ดร.ไกรสิน	ส่งวัฒนา	
ผศ.บรรจง	ปิยธำรง	

ค่าระดับคะแนนที่ผ่านเป็นเอกฉันท์จากคณะกรรมการสอบ **GOOD**

วัน/เดือน/ปี ที่สอบ 7 เมษายน 2541 **เวลา** 13.30-15.30 น.

สถานที่สอบ ห้องสอบวิทยานิพนธ์ คณะวิศวกรรมศาสตร์ ตึก 12 ชั้น ชั้น 4 ห้อง (E12-404)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.มนัส สังวรศิลป์)

คณบดีบัณฑิตวิทยาลัย

วันที่ 29 **เดือน** เมษายน **พ.ศ.** 2541

หมายเหตุ การวัดผลวิทยานิพนธ์ให้ใช้ค่าระดับคะแนนดังนี้

ค่าระดับคะแนน	ผลการศึกษา
O	Outstanding (ดีเยี่ยม)
G	Good (ดี)
P	Pass (ผ่าน)
F	Fail (ไม่ผ่าน)

หัวข้อวิทยานิพนธ์	การตรวจจับแพ็กเก็ตข้อมูลและการประยุกต์ใช้งานบนระบบทีซีพี/ไอพี
นักศึกษา	นายนภัทร สระเอี่ยม
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. กอบชัย เดชหาญ
หลักสูตร	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2541

บทคัดย่อ

บทความนี้กล่าวถึงระบบเฝ้าดูและตรวจสอบแพ็กเก็ต ที่ใช้โพรโตคอลแบบทีซีพี/ไอพีบนระบบเครือข่ายคอมพิวเตอร์แบบอีเทอร์เน็ต ในการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์จะมีแพ็กเก็ตไหลอยู่ และจะใช้โพรโตคอลตามที่กำหนดไว้ในการติดต่อสื่อสารกัน การเฝ้าดูแพ็กเก็ตที่วิ่งอยู่ในระบบเครือข่ายจะสามารถทราบรายละเอียดเกี่ยวกับกระบวนการต่างๆที่เกิดขึ้นภายในระบบ และสามารถนำข้อมูลต่างๆที่ได้มาวิเคราะห์เพื่อใช้ในการเพิ่มประสิทธิภาพในการสื่อสารข้อมูล การวางแผนด้านความปลอดภัย และการแก้ไขปัญหาอันเนื่องมาจากข้อมูลในระบบ

Thesis Title	Data packet detection on TCP/IP and its applications
Student	Mr. Napat Sra-ium
Thesis Advisor	Assoc.Prof. Dr. Kobchai Dejhan
Degree	Master of Engineering in Electrical Engineering
Year	1998

ABSTRACT

This paper presents the packet detection and monitoring system under the TCP/IP protocol environment, implementing on the Ethernet local area network. In data communication network, the data, as a datagram flow through the net using the predetermined protocol. By monitoring traffic in the network, it is able to know the exact operation taking place in the system. Furthermore, the collected data is then analyzed and the outcome can be employed to enhance the system performance, to make network security planning and to resolve the traffic problem.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษาเกี่ยวกับการพัฒนาโปรแกรมที่ใช้ในการทดสอบ ซึ่งผู้จัดทำขอขอบพระคุณแก่บุคคลดังต่อไปนี้

ขอขอบพระคุณ รองศาสตราจารย์ ดร.กอบชัย เดชหาญ ที่ช่วยให้คำแนะนำและช่วยเหลือทุกๆอย่างในการจัดทำวิทยานิพนธ์

ขอขอบพระคุณ ท่านอาจารย์วัชระ ฉัตรวิริยะ, อาจารย์สมศักดิ์ วลัยรัตน์ และอาจารย์บรรจง ปิยธำรงที่ช่วยเหลือแก้ไขและให้คำแนะนำในบางจุดที่ผู้จัดทำติดปัญหาบางอย่าง

ขอขอบพระคุณ ท่านอาจารย์บุญชัย เรืองสุขนุกูล ที่ได้ให้คำแนะนำและแลกเปลี่ยนความรู้เกี่ยวกับระบบเครือข่ายและชุดโพรโตคอลทีซีพี/ไอพี

ขอขอบคุณ คุณอนันตศักดิ์ ทิพย์พญาชัยที่ได้ให้คำแนะนำเกี่ยวกับการตรวจจับข้อมูลบนระบบเครือข่าย

ขอขอบพระคุณ คุณแม่ พี่ๆ น้องและเพื่อนๆ ที่ให้กำลังใจในการทำวิจัยเสมอมา

สุดท้ายขอขอบคุณบัณฑิตวิทยาลัย ที่ได้ให้ทุนสนับสนุนการทำวิทยานิพนธ์ครั้งนี้

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

นภัทร สระเอี่ยม

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญภาพ

ภาพที่ หน้า

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

บทที่ 1

บทนำ

1.1 แนวความคิดของวิทยานิพนธ์

ทีซีพี/ไอพีเป็นโพรโตคอลในการสื่อสารข้อมูลบนระบบเครือข่ายคอมพิวเตอร์ที่นิยมใช้กันมากในปัจจุบัน เหตุผลอย่างหนึ่งคือมันเป็นโพรโตคอลหลักที่ใช้ในระบบเครือข่ายคอมพิวเตอร์ที่เรียกกันว่า อินเทอร์เน็ต ซึ่งเป็นระบบเครือข่ายที่ใหญ่ที่สุดในโลก นอกจากนี้การใช้งานระบบเครือข่ายคอมพิวเตอร์ที่แต่เดิมมักใช้กันแบบระบบประมวลผลที่ศูนย์กลาง (Centralized system) ก็เริ่มเปลี่ยนมาใช้ระบบประมวลผลแบบกระจาย (Distributed system) หรือ Client/Server แทน ซึ่งก็มักจะใช้ทีซีพี/ไอพีเป็นโพรโตคอลหลักในการให้บริการแบบต่างๆ เช่น การล็อกอินระยะไกล และการแลกเปลี่ยนไฟล์ เป็นต้น การจัดสร้างระบบตรวจสอบและเฝ้าดูแพ็กเก็ตที่ไหลอยู่บนระบบเครือข่ายคอมพิวเตอร์ที่ใช้ทีซีพี/ไอพีเป็นโพรโตคอลหลักจะทำให้สามารถเข้าใจการสื่อสารข้อมูลและกระบวนการต่างๆที่เกิดขึ้นในระบบได้เป็นอย่างดี

1.2 วัตถุประสงค์ในการทำวิทยานิพนธ์

ในการทำวิทยานิพนธ์เรื่องการตรวจจับแพ็กเก็ตข้อมูลและการประยุกต์ใช้งานบนระบบทีซีพี/ไอพี ได้ศึกษาถึงรูปแบบการรับส่งข้อมูลของชุดโพรโตคอลทีซีพี/ไอพี ซึ่งเป็นโพรโตคอลที่ได้รับความนิยมในการใช้งานอย่างสูง และได้กำหนดจุดประสงค์ไว้ดังนี้

- เพื่อศึกษาลักษณะการรับส่งข้อมูลบนระบบเครือข่ายชั้นล่างสุดของทีซีพี/ไอพี ซึ่งในวิทยานิพนธ์นี้ใช้ระบบเครือข่ายแบบอีเทอร์เน็ต
- เพื่อศึกษารูปแบบแพ็กเก็ตข้อมูลของระบบทีซีพี/ไอพี โดยเฉพาะโพรโตคอลในระดับโปรแกรมประยุกต์ที่นิยมใช้งานกันทั่วไป เช่น TELNET, FTP และ HTTP

- เพื่อสามารถพัฒนาโปรแกรมที่ใช้ในการแสดงข้อมูลภายในแพ็กเก็ตข้อมูลของระบบทีซีพี/ไอพี เพื่อนำข้อมูลที่ได้มาใช้ในการวิเคราะห์หรือแก้ไขปัญหาที่อาจเกิดขึ้นในระบบเครือข่ายได้
- เพื่อศึกษาการเขียนชุดคำสั่งในการติดต่อกับโปรแกรมแพ็กเก็ตไดร์เวอร์ ที่ใช้ในการตรวจสอบแพ็กเก็ตข้อมูลที่ส่งมายังการ์ดเชื่อมต่อระบบเครือข่ายที่ใช้งานกันอยู่

1.3 รายละเอียดในวิทยานิพนธ์

ในวิทยานิพนธ์นี้ ได้แบ่งเนื้อหาออกเป็นบทได้ทั้งหมด 8 บท โดยในบทที่ 1 จะเป็นการกล่าวนำถึง แนวความคิดและวัตถุประสงค์ในการทำวิทยานิพนธ์ และได้กล่าวถึงเนื้อหาโดยย่อของแต่ละบท ซึ่งในบทอื่นๆ จะมีเนื้อหาดังนี้

บทที่ 2 กล่าวถึงชุดโพรโตคอลทีซีพี/ไอพี ที่เป็นโพรโตคอลมาตรฐานตัวหนึ่งและเป็นโพรโตคอลหลักในระบบเครือข่ายอินเตอร์เน็ต

บทที่ 3 กล่าวถึงทฤษฎี หลักการ และรูปแบบการรับส่งข้อมูลของระบบเครือข่ายอีเทอร์เน็ต

บทที่ 4 กล่าวถึงโพรโตคอลในชั้นอินเตอร์เน็ต ลักษณะ โครงสร้างและตัวอย่างโพรโตคอลที่นิยมใช้กัน

บทที่ 5 กล่าวถึงโพรโตคอลในชั้นโฮสต์ทูโฮสต์ซึ่งได้แก่ ทีซีพี และ ยูดีพี ลักษณะและโครงสร้าง

บทที่ 6 กล่าวถึงหลักการและรูปแบบในการเขียนโปรแกรมติดต่อกับแพ็กเก็ตไดร์เวอร์ เพื่อให้สามารถทำการครวจจับแพ็กเก็ตที่เกิดขึ้นในระบบเครือข่ายได้ หลักการของระบบตรวจสอบและโปรแกรมที่พัฒนาขึ้น

บทที่ 7 กล่าวถึงวิธีการทดลองโปรแกรมที่พัฒนาขึ้นกับระบบเครือข่ายขนาดเล็กที่ใช้งานอยู่จริง

บทที่ 8 กล่าวสรุปปัญหาที่เกิดขึ้น แนวทางการนำโปรแกรมไปประยุกต์ใช้และพัฒนาต่อ

ภาคผนวก ได้แสดงรูปแบบ ชนิด ของข้อมูลในระดับชั้นอีเทอร์เน็ตและอินเตอร์เน็ตโพรโตคอล

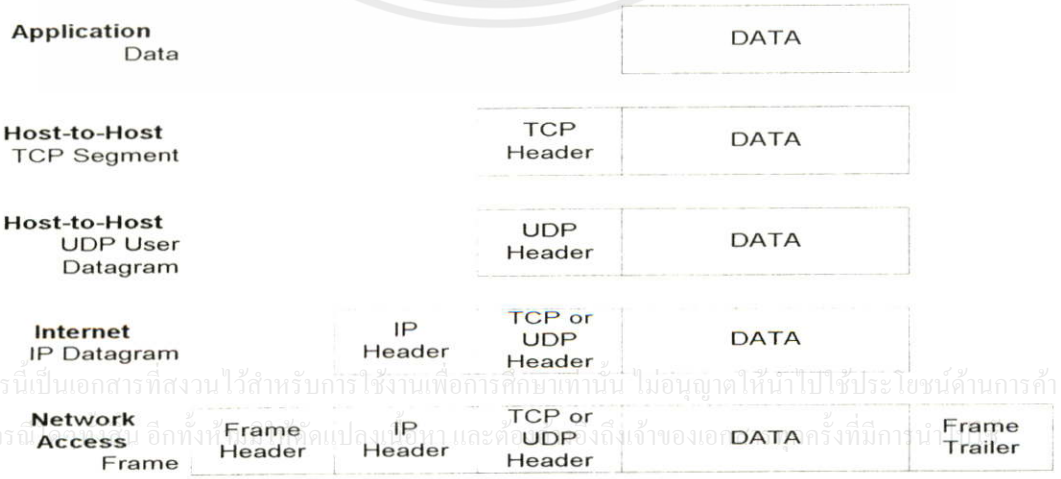
บทที่ 2

ชุดโพรโตคอลทีซีพี/ไอพี

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

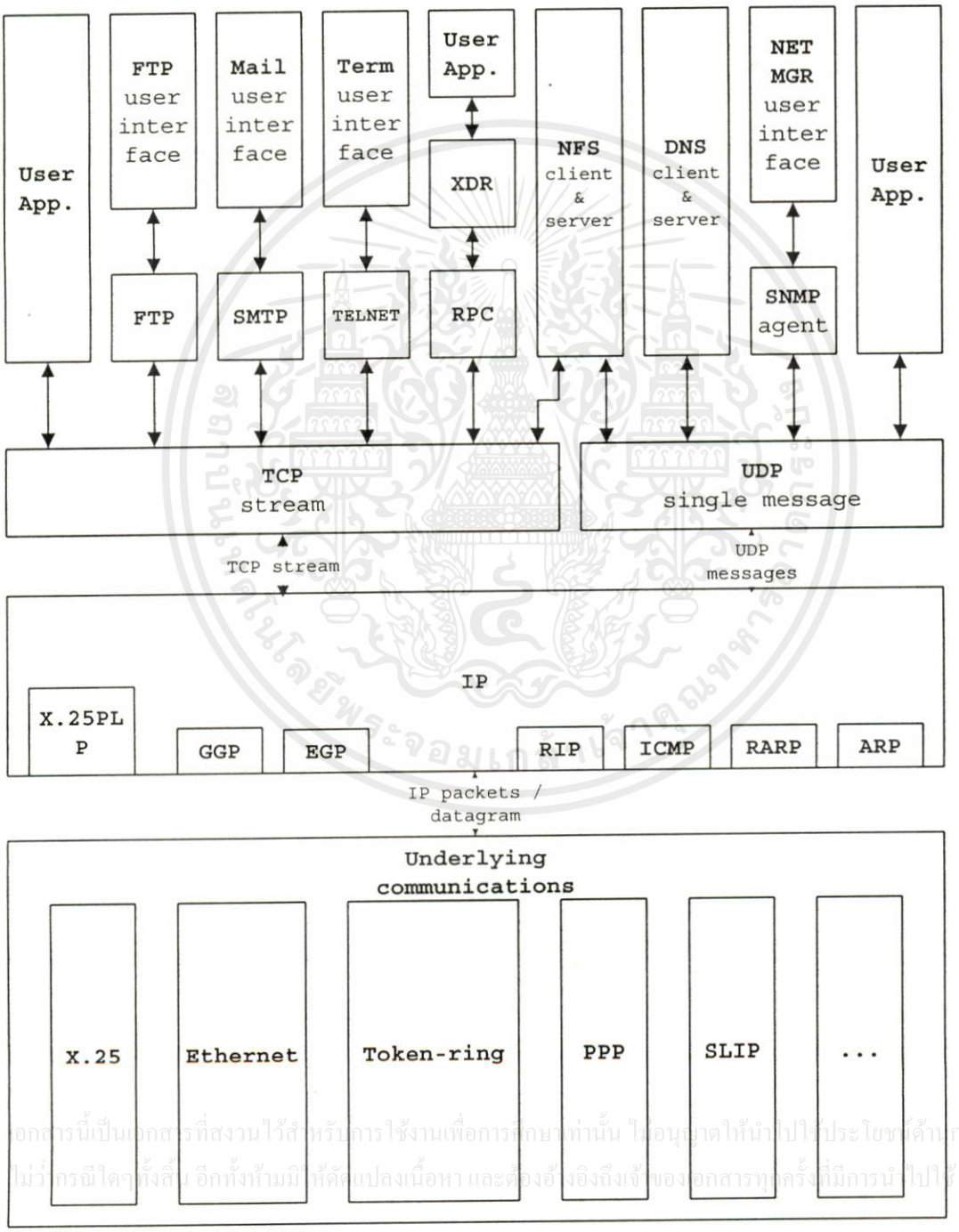
ข้อมูลที่ใช้ทีซีพี/ไอพีนำส่งจะถูกแบ่งออกเป็นข้อมูลย่อยหลายส่วนๆเพื่อทยอยส่งไปตามลำดับเพื่อให้เหมาะสมกับระบบเครือข่ายในชั้นถัดไปที่อาจจะไม่สามารถส่งข้อมูลขนาดใหญ่ได้ทันที และเมื่อส่งไปถึงปลายทางก็จะรวบรวมข้อมูลนั้นกลับเป็นข้อมูลชุดเดิมอีกครั้งหนึ่ง ซึ่งจะมีการจัดรูปแบบแพ็กเก็ตในการสื่อสารดังรูป

รูปที่ 2.1 การจัดเตรียมข้อมูลเป็นแพ็กเก็ตเพื่อทำการส่ง



ชุดโพรโตคอลทีซีพี/ไอพีจะมีทีซีพี/ไอพีเป็นหลักและโพรโตคอลอื่นๆที่ทำงานร่วมกับทีซีพี/ไอพีในชั้นอื่นๆของทีซีพี/ไอพีโมเดล ซึ่งจะมีทั้งที่เป็นโพรโตคอลช่วยเหลือ เช่น ICMP, ARP, RIP และโพรโตคอลที่ใช้ทำงานหลัก เช่น TELNET, FTP, SMTP, HTTP, SNMP เป็นต้น

รูปที่ 2.2 การใช้งานชุดโพรโตคอลทีซีพี/ไอพี



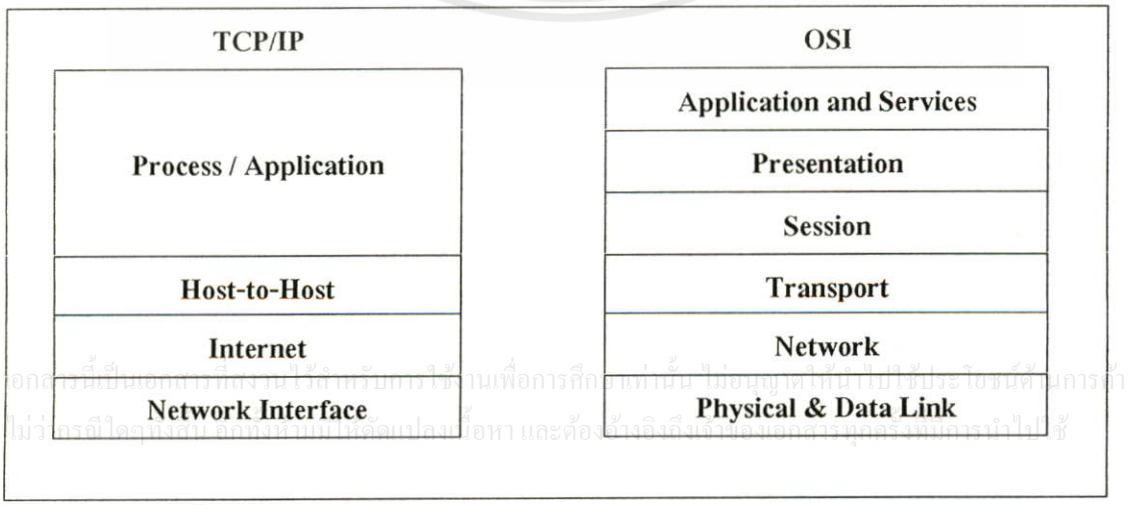
2.1 ชั้นต่างๆของทีซีพี/ไอพี (TCP/IP Layer)

การติดต่อสื่อสารของทีซีพี/ไอพีถูกกำหนดให้มีการทำงานเป็นระดับชั้น (layer) เพื่อให้มีการทำงานเป็นอิสระต่อกันในแต่ละระดับชั้น และเพื่อให้มีขั้นตอนการทำงานในการแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์เป็นไปอย่างถูกต้องดังนี้

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

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

รูปที่ 2.3 การแบ่งระดับการทำงานของทีซีพี/ไอพี และ OSI



2.2 ชั้นเชื่อมต่อระบบเครือข่าย (Network Interface Layer)

ทำงานในชั้นเดียวกับ OSI Physical Layer และ Data Link Layer ชั้นนี้จะทำหน้าที่ในการสื่อสารข้อมูลทางกายภาพ ในระดับสัญญาณนำส่ง ตัวนำที่ใช้ในการส่ง ระบบสื่อสัญญาณ และรูปแบบสัญญาณที่ใช้ในการแทนข้อมูล ว่าเป็นสัญญาณลอจิก "0" หรือ "1" ตัวอย่างของระบบที่ทำงานในชั้นนี้ เช่น ระบบเครือข่ายแบบอีเทอร์เน็ต หรือระบบเครือข่ายแบบโทเก้นริง และจัดข้อมูลเป็นกลุ่มที่เรียกว่าเฟรม (Frame) เฟรมจะมีส่วนหัวใช้แสดงตำแหน่งของต้นทางและปลายทาง ข่าวสารที่ใช้ในการควบคุม และส่วนท้ายที่ใช้ในการตรวจสอบข้อผิดพลาดการติดต่อสื่อสารระดับล่างสุด เฟรมจะถูกส่งจากอุปกรณ์เชื่อมต่อระบบเครือข่าย (Network Interface Device) ของเครื่องต้นทางผ่านระบบสื่อสัญญาณต่าง ๆ ไปถึงอุปกรณ์เชื่อมต่อระบบเครือข่ายของเครื่องปลายทาง

2.3 ชั้นอินเตอร์เนต (Internet Layer)

ชั้นนี้จะมีอินเตอร์เน็ตโพรโตคอล (Internet Protocol) หรือไอพี (IP) เป็นโพรโตคอลหลัก คอยทำการหาเส้นทางที่เหมาะสมให้ในการสื่อสารข้อมูลระหว่างระบบ ข้อมูลที่จะส่งเรียกว่าเดต้าแกรม (Datagram) ซึ่งจะถูกส่งไปในระบบที่อาจจะเชื่อมต่อกันโดยตรงหรือเชื่อมต่อกันผ่านระบบสื่อสารอื่นๆอยู่ก็ได้

ไอพีจะทำงานแบบไม่มีการเชื่อมต่อก่อน (connectionless) เดต้าแกรมแต่ละตัวจะถูกจัดเส้นทางในการส่งเป็นอิสระต่อกัน ไอพีไม่มีการรับประกันความถูกต้อง ความน่าเชื่อถือ หรือแม้แต่การจัดเรียงลำดับเดต้าแกรมให้อยู่ในลำดับที่ถูกต้อง

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

2.4 ชั้นโฮสต์ทูโฮสต์ (Host - to - Host Layer) - TCP และ UDP

โพรโตคอลที่ทำงานในชั้นโฮสต์ทูโฮสต์นี้มีอยู่ 2 แบบ แบบที่เรียกว่า ทีซีพี (TCP - Transmission Control Protocol) จะทำงานแบบมีการเชื่อมต่อก่อน (connection orient) ซึ่งจะเป็นส่วนการทำงานภายในตัวคอมพิวเตอร์แต่ละเครื่อง มีหน้าที่นำส่งข้อมูลโดยรับประกันความน่าเชื่อถือให้ด้วยว่าข้อมูลที่นำส่งจะไม่มีข้อผิดพลาดและเรียงอยู่ในลำดับที่ถูกต้อง โพรโตคอลทีซีพีจะทำการเพิ่มส่วนหัวของชั้นโพรโตคอลให้กับข้อมูลเพื่อสร้างเป็นเซ็กเมนต์ (Segment) โพรโตคอลอีกแบบหนึ่งได้แก่ ยูดีพี (UDP - User Datagram Protocol) ซึ่งจะทำงานแบบไม่มีการเชื่อมต่อก่อน (connectionless) และไม่มีการรับประกันความถูกต้องของข้อมูล เรียกข้อมูลที่ส่งโดยยูดีพีว่า User Datagram ตัวอย่างการทำงานโดยใช้ยูดีพี เช่น การสอบถามข้อมูลชื่อจากฐานข้อมูลในระบบ Domain Name System

2.5 ชั้นโปรแกรมประยุกต์ (Application Layer)

ชุดโพรโตคอลทีซีพี/ไอพีจะมีโพรโตคอลในชั้นโปรแกรมประยุกต์ให้ใช้งานอยู่มาก ที่นิยมใช้กันมากและจัดเป็นบริการพื้นฐานของทีซีพี/ไอพี เช่น

การล็อกอินระยะไกล (**remote login**) ทำให้ผู้ใช้เครื่องคอมพิวเตอร์ในระบบเครือข่ายสามารถทำการล็อกอินเข้าไปใช้ทรัพยากรในคอมพิวเตอร์อื่นที่ต่อเชื่อมกันอยู่ในระบบเครือข่ายได้จากเทอร์มินอล (Terminal) ของตัวเองซึ่งมีความแตกต่างกัน โดยใช้เทลเน็ตโพรโตคอล (TELNET - Telecommunication Network Protocol) ทำให้เกิดโปรแกรม telnet ที่ใช้ระบบเทอร์มินอลจำลอง (NVT - Netwrok Virutal Terminal) ซึ่งสามารถใช้งานกับระบบคอมพิวเตอร์ได้เกือบทุกระบบ

การส่งผ่านแฟ้มข้อมูล (files transfer) ทำให้ผู้ใช้เครื่องคอมพิวเตอร์เครื่องใดก็ตามสามารถรับส่งไฟล์จากเครื่องคอมพิวเตอร์อื่นได้ โดยใช้ไฟล์ทรานสเฟอร์โพรโตคอล (FTP – Files Transfer Protocol) ทำหน้าที่ในการคัดลอกแฟ้มข้อมูลระหว่างเครื่อง และทำงานทั่วไปเกี่ยวกับแฟ้มข้อมูลเช่น การเปลี่ยนชื่อแฟ้ม การลบแฟ้ม เป็นต้น

การส่งไปรษณีย์อิเล็กทรอนิกส์ (Electronic Mail) ทำให้ผู้ใช้ส่งข้อความไปหาผู้ใช้คนอื่นในระบบเครือข่ายได้ โดยการกำหนดรูปแบบข้อความที่จะส่งให้เป็นมาตรฐานเดียวกันในกระบวนการในการรับและการส่งระหว่างเครื่องต่าง ๆ

บริการเวิลด์ไวด์เว็บ (World Wide Web) จัดเป็นบริการที่มีความสามารถมากที่สุดในโปรแกรมประยุกต์ที่ทำงานแบบ Client/Server ของทีซีพี/ไอพี และได้รับความนิยมสูงมากในปัจจุบัน ทำให้ผู้ใช้สามารถสืบค้นข้อมูลในลักษณะของ Hypermedia ได้โดยการใช้เอชทีทีพีโพรโตคอล (HTTP - Hypertext Transfer Protocol)

เน็ตเวิร์คไฟล์ซิสเต็ม (NFS - Network File System) เป็นการอนุญาตให้ระบบเข้าถึงข้อมูลจากคอมพิวเตอร์เครื่องอื่นได้โดยผ่านระบบไฟล์ของระบบจัดการนั้นๆเอง

การพิมพ์ระยะไกล (remote printing) ทำให้สามารถใช้งานเครื่องพิมพ์ผ่านระบบเครือข่ายได้

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

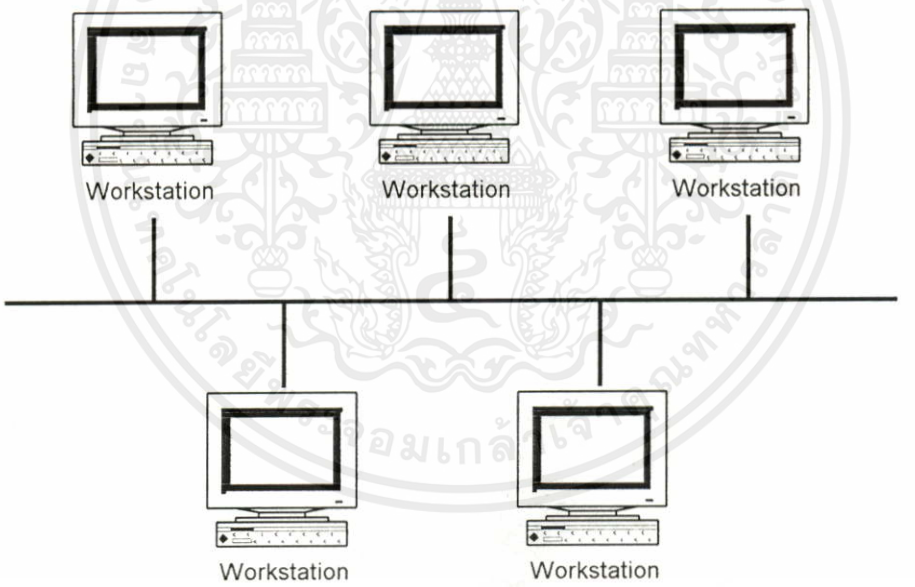
ในปัจจุบันได้มีการสร้างโปรแกรมที่ทำงานบนเครือข่ายโดยใช้ ทีซีพี/ไอพีจำนวนมาก เช่น ระบบฐานข้อมูลที่ให้บริการระหว่างเครื่องในเครื่องข่าย หรือการทำงานบนระบบประมวลผลแบบกระจาย (Client / Server Processing) ซึ่งแสดงให้เห็นว่าทีซีพี/ไอพี มีบทบาทเป็นโพรโตคอลพื้นฐานที่สำคัญของการใช้งานระบบเครือข่ายในปัจจุบัน

บทที่ 3

อีเทอร์เน็ต

ชั้นล่างสุดของชุดโพรโตคอลทีซีพีไอพีจะเป็นชั้นเชื่อมต่อกับระบบเครือข่าย ซึ่งจะทำงานร่วมกับระบบเครือข่ายได้หลายประเภท ที่นิยมมากที่สุดในระบบเครือข่ายคอมพิวเตอร์แบบท้องถิ่นจะเป็นระบบเครือข่ายที่เรียกกันว่า **อีเทอร์เน็ต (Ethernet)** ซึ่งเป็นระบบเครือข่ายที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบัน

รูปที่ 3.1 การเชื่อมต่อเครื่องคอมพิวเตอร์ในระบบเครือข่ายอีเทอร์เน็ต



การสื่อสารในระบบเครือข่ายอีเทอร์เนตจะต้องมีหมายเลขแอดเดรส (address) เป็นตัวกำหนดการติดต่อสื่อสารระหว่างสถานีงานต้นทางและสถานีงานปลายทาง ค่าหมายเลขนี้เป็นเลขขนาด 48 บิตที่ถูกกำหนดมาจากโรงงานที่ทำการ์ดเชื่อมต่อระบบเครือข่ายและต้องไม่มีการซ้ำกัน ข้อมูลที่ถูกส่งออกไปจะเป็นการส่งแบบกระจายโดยใช้ตัวกลาง (broadcast medium) คือส่งกระจายออกไปให้กับสถานีงานทุกๆตัว เมื่อสถานีงานได้รับแพ็กเก็ตก็จะทำการตรวจสอบว่าเป็นแพ็กเก็ตที่ส่งมาถึงตนเองหรือไม่ ถ้าใช่ก็จะนำข้อมูลในแพ็กเก็ตนั้นมาดำเนินการต่อ ซึ่งจะพิจารณาจากส่วนหัวอีเทอร์เนต โดยทุกๆ อีเทอร์เนตแพ็กเก็ตจะมีส่วนหัวขนาด 14 อ็อกเต็ทที่ใช้บอกถึงแอด

เดรสต้นทาง (source address) แอดเดรสปลายทาง (destination address) และ ชนิด (type) เมื่อได้รับแพ็กเก็ตที่ถูกต้องแล้วแล้วสถานีงานจะพิจารณาที่ไทพ์โคด (type code) เพื่อนำแพ็กเก็ตนั้นมาประมวลผลต่อไป

3.1 หลักการทำงานของอีเทอร์เน็ต

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

เด็ค (DEC) อินเทลคอร์ปปอร์เรชัน (Intel) และ ซีรอก (Xerox) ได้ร่วมกันกำหนดมาตรฐานอีเทอร์เนตเป็นรุ่นที่ 1 (DIX Ethernet) โดยมีลักษณะดังนี้

- รูปร่างระบบเครือข่าย ใช้โทโพโลจีแบบบัส (bus)
- ใช้วิธีการสื่อสารแบบ CSMA/CD (Carrier Sense Multiple Accesss with Collision Detecion)
- อัตราการรับส่งข้อมูลสูงสุด10 เมกกะบิตต่อวินาที
- ความยาวสูงสุด 2.5 กิโลเมตร
- จำนวนสถานีงานสูงสุดต่อเครือข่าย 1024 เครื่อง
- ใช้หลักการส่งแบบมีแถบกว้างความถี่ (baseband)
- ขนาดเฟรมเปลี่ยนแปลงได้

รูปที่ 3.2 โครงสร้างเฟรมของ DIX Ethernet

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

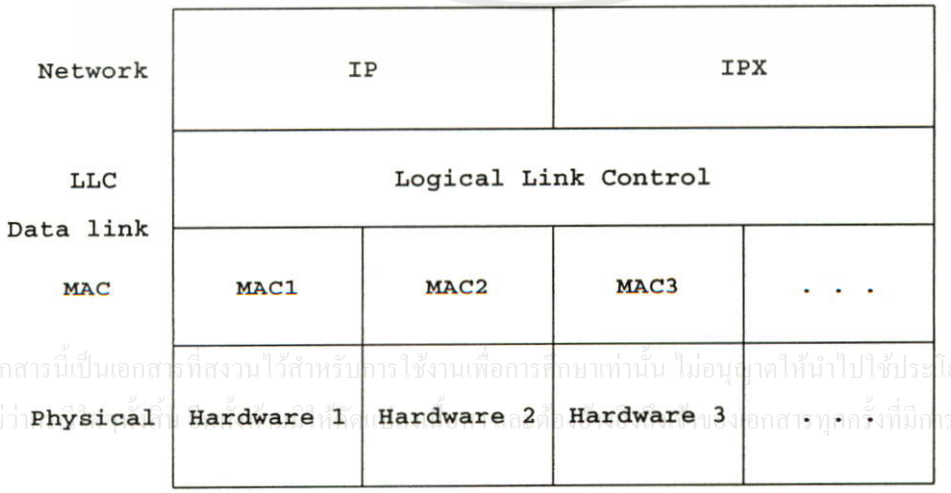
ซึ่งต่อมา IEEE ก็ได้ปรับปรุงโพรโตคอลนี้เพิ่มเติมและออกเป็นระบบมาตรฐานที่เรียกว่ามาตรฐาน 802.3 ซึ่งเป็นอีเทอร์เน็ตอีกแบบหนึ่งที่มีความเข้ากันได้กับ DIX Ethernet เดิมโดยเปลี่ยนฟิลด์ Frame type ไปเป็น Data Length แทน ค่าของ Frame type จะเริ่มที่ 0800H ดังนั้น Data Length จึงมีค่าได้ไม่เกิน 0800H ถ้าข้อมูลในฟิลด์นี้มีค่าตั้งแต่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

รูปที่ 3.3 โครงสร้างเฟรมของ IEEE 802.3 Ethernet

64 Bits	48 Bits	48 Bits	16 Bits	368-12000 Bits	32 Bits
Preamble	Destination Address	Source Address	DataLength	Frame Data	CRC

การที่อีเทอร์เน็ตมีเฟรมข้อมูลที่แตกต่างกันทำให้สามารถส่งข้อมูลหลายรูปแบบและหลายปลายทางผ่านไปยังฮาร์ดแวร์ที่ใช้การ์ดเชื่อมต่อระบบเครือข่ายแบบอีเทอร์เน็ตตัวเดียวกันได้ ทำให้มีลักษณะ Multiprotocol stack เกิดขึ้น โดยจะใช้การบอกตัว Multiplexer ว่าเฟรมข้อมูลนี้จะต้องส่งไปที่โพรโตคอลในชั้นข้างบนอย่างไร ชนิดของเฟรมทั้งหมดได้แก่ ETHERNET_802.2, ETHERNET_802.3, ETHERNET_II และ ETHERNET_SNAP ในการทำลักษณะนี้จะต้องแบ่งการทำงานในชั้นที่ 2 ออกเป็น 2 ชั้นย่อย คือ LLC - Logical Link Control และ MAC - Media Access Control ดังรูปที่ 3.4

รูปที่ 3.4 การทำมัลติโพรโตคอลแสต็กของอีเทอร์เน็ต



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

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

- LLC LSL.COM
- LAN Card Driver NE2000.COM
- IPX/SPX IPXODI.COM
- Session, Presentation, Application NETX.EXE
- TCP/IP TCPIP.EXE
- Session, Presentation, Application เช่น TNVT220 หรือ Netscape

3.2 ส่วนประกอบของอีเทอร์เนตเฟรม

อีเทอร์เนตเฟรมมีขนาดประมาณ 46 และ 1500 ไบต์ เฟรมน้อยกว่า 60 ไบต์ในส่วนข้อมูลจะเรียกว่า Runt Frame

รูปที่ 3.5 โครงสร้างของอีเทอร์เน็ตเฟรม

Preamble	Destination Address	Source Address	Type	Data	FCS

เฟรมนี้ประกอบด้วย 6 ฟิลด์ด้วยกัน ซึ่งมีรายละเอียดดังนี้

พรีเอมเบิล (Preamble)

เลขลำดับ 64 บิตที่ฟิสิคัลเลเยอร์ใช้เพื่อการสร้างสัญญาณพร้อม (synchronization signal) ระหว่างวงจรที่เชื่อมต่อกับสื่อที่ใช้ในการส่ง

ที่อยู่ปลายทาง (Destination Address)

เลขฮาร์ดแวร์แอดเดรส ขนาด 48 บิต ซึ่งเรียกว่า อีเทอร์เน็ตแอดเดรส หรือ แมคแอดเดรส การ์ดเชื่อมต่อระบบเครือข่ายทุกแผ่นจะมีเลข 48 บิตนี้ใช้บอกถึงเลขประจำตัวของการ์ดเชื่อมต่อนั้นๆของสถานีงานที่ต้องการส่งข้อมูลไป

ที่อยู่ต้นทาง (Source Address)

เป็นแมคแอดเดรสของผู้ส่ง

ไทป์ (Type)

เลขขนาด 16 บิต เพื่อบอกชนิดโพรโตคอล ถ้ามีการใช้งานหลายโพรโตคอลในระดับเดียวกันบนสื่อเดียวกัน จะใช้เลขนี้บอกถึงโพรโตคอลในระดับชั้นบนขึ้นไป

ข้อมูล (Data)

เป็นข้อมูลขนาดระหว่าง 46 ถึง 1500 ไบต์

เอฟซีเอส (FCS – Frame Check Sequence)

เลขซีอาร์ซีขนาด 32 บิตที่คำนวณจากทุกๆฟิลด์ยกเว้นฟิลด์ตัวเอง

ในระบบเครือข่ายอีเทอร์เน็ตทั่วไปจะมีโครงสร้างเฟรมของอีเทอร์เนตที่ใช้พื้นฐานโครงสร้างเฟรมอีเทอร์เนตคล้ายกัน เพียงแต่แตกต่างกันตรงส่วนการใช้งานของแต่ละแบบที่ไม่เหมือนกันซึ่งมีด้วยกันดังนี้

3.3 อีเทอร์เนต 802.3

อีเทอร์เนต 802.3 เฟรมนั้นมีลักษณะคล้ายกับอีเทอร์เน็ตทู แต่จะมีบางฟิลด์ที่แตกต่างกันดังรูปที่ 3.6 ซึ่งมีหน้าที่และขนาดเฉพาะ ดังมีรายละเอียดดังนี้

รูปที่ 3.6 โครงสร้างของเฟรมอีเทอร์เนต 802.3

Preamble	Destination Address	Source Address	Length	Data	FCS

โครงสร้างเฟรม

พรีเอมเบิ้ล

เลขลำดับ 56 บิตที่ฟิสิคัลเลเยอร์ใช้เพื่อการสร้างสัญญาณพร้อม (synchronization signal) ระหว่างวงจรที่เชื่อมต่อกับสื่อที่ใช้ในการสื่อสาร

เอสเอฟดี (SFD – Start Frame Delimiter)

เป็นเลขไบนารี่ (binary) 10101011 ที่ชี้จุดเริ่มต้นของเฟรม

หมายเลขปลายทาง

เลขฮาร์ดแวร์แอดเดรสขนาด 48 บิต เรียกว่า อีเทอร์เน็ตแอดเดรส หรือ แมคแอดเดรส ของสถานีงานที่ต้องการส่งข้อมูลไป แอดเดรสที่มีค่าเป็น 0FFFFFFFFFFFH จะหมายถึงบรอดคาสแอดเดรส

หมายเลขต้นทาง

เลขฮาร์ดแวร์แอดเดรสขนาด 48 บิตของผู้ส่งซึ่งต้องไม่เป็นบรอดคาสท์แอดเดรส

ความยาว (Length)

เป็นเลขขนาด 16 บิต เพื่อบอกขนาดแพ็กเก็ต ค่านี้จะต้องน้อยกว่า 1500

ข้อมูล (Data)

เป็นข้อมูลขนาดระหว่าง 46 ถึง 1500 ไบต์

แพดดิ้ง (Padding)

ฟิลด์นี้มีขนาดเปลี่ยนแปลงได้ เป็นค่าที่เพิ่มเข้าไปเพื่อให้แพ็กเก็ตมีขนาดตรงตามข้อกำหนด เช่น เฟรมอีเทอร์เนตต้องมีขนาดอย่างน้อย 64 ไบต์ ซึ่งถ้ามีการส่งข้อมูลที่มีขนาดน้อยกว่านี้

จะต้องเพิ่มแพดดิ้งเข้าไปเพื่อให้ครบ 64 ไบต์ ถ้าเฟรมถูกต้องและค่าความยาวมากกว่า 1500 ไบท์จะหมายความว่าเป็นอีเทอร์เนตทูเฟรมและเป็นไทบ์ฟิลด์

3.4 อีเทอร์เนต 802.2

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

รูปที่ 3.7 โครงสร้างเฟรมอีเทอร์เนต 802.2

Preamble	SFD	DA	SA	Length	DSAP	SSAP	Control	Data	Pad	FCS

โครงสร้างเฟรม

พรีเอมเบิ้ล

เลขลำดับ 56 บิตที่ฟิสิคัลเลเยอร์ใช้เพื่อการสร้างสัญญาณพร้อม (synchronization signal) ระหว่างวงจรที่เชื่อมต่อกับสื่อที่ใช้ในการสื่อสาร

หมายเลขปลายทาง

เลขฮาร์ดแวร์แอดเดรสขนาด 48 บิต เรียกว่า อีเทอร์เน็ตแอดเดรส หรือ แมคแอดเดรส ของสถานีงานที่ต้องการส่งข้อมูลไป แอดเดรสที่มีค่าเป็น 0FFFFFFFFFFFH จะหมายถึงบรอดคาสแอดเดรส

หมายเลขต้นทาง

เลขฮาร์ดแวร์แอดเดรสขนาด 48 บิตของผู้ส่ง

ความยาว (Length)

เป็นเลขขนาด 16 บิต เพื่อบอกขนาดแพ็กเก็ต ค่านี้จะต้องน้อยกว่า 1500

ข้อมูล (Data)

เป็นข้อมูลขนาดระหว่าง 46 ถึง 1500 ไบต์

แพ็ดดิ้ง (Padding)

ขนาดเปลี่ยนแปลงได้

เอฟซีเอส (FCS – Frame Check Sequence)

เลขซีอาร์ซีขนาด 32 บิตที่คำนวณจากทุกๆฟิลด์ยกเว้นฟิลด์ตัวเอง

ดีเอสเอพี (DSAP - Destination Service Access Point)

เป็นเซอร์วิสแอคเซสพอยต์ปลายทางของสถานีงานปลายทางซึ่งใช้ในเลเยอร์บนหรือเน็ตเวิร์คเลเยอร์

เอสเอสเอพี (SSAP - Source Service Access Point)

เป็นเซอร์วิสแอคเซสพอยต์ต้นทางของสถานีงานต้นทางซึ่งใช้ในเลเยอร์บนหรือเน็ตเวิร์คเลเยอร์

ฟิลด์ควบคุม (Control)

กำหนดการส่งแบบคอนเน็คชันเลสเซอร์วิส (Connectionless Service)

3.5 อีเทอร์เนตทู (Ethernet II)

อีเทอร์เนตทูเฟรมแตกต่างจากอีเทอร์เนตเฟรม 2 แบบที่กล่าวมาเนื่องจากไทป์ฟิลด์ซึ่งตามหลังที่อยู่ปลายทาง แต่ อีเทอร์เนต 802.3 อีเทอร์เนต 802.2 จะเป็นฟิลด์ความยาวแทน

รูปที่ 3.8 โครงสร้างเฟรมอีเทอร์เนตทู

Preamble	Destination Address	Source Address	Type	Data	FCS

โครงสร้างเฟรม

พรีเอมเบิ้ล

เป็นเลขลำดับ 64 บิต ไบต์ที่ฟิสิคัลเลเยอร์ใช้เพื่อการสร้างสัญญาณพร้อม (synchronization signal) ระหว่างวงจรที่เชื่อมต่อกับสื่อ กำหนดด้วยค่าสลับกันระหว่าง “1” และ “0” ซึ่งจะมีด้วยกันทั้งหมด 7 ไบต์ ส่วนไบต์สุดท้ายเป็นเอสเอฟดี

ไทบ์

เป็นส่วนที่บอกถึงชนิดโพรโตคอลที่ใช้ในระดับชั้นที่สูงกว่า ค่าโพรโตคอลที่ใช้มีดังนี้

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

บทที่ 4

โพรโตคอลในชั้นอินเตอร์เน็ต

ทีซีพี/ไอพีเป็นโพรโตคอลที่ใช้สื่อสารข้อมูลกันในระบบเครือข่ายคอมพิวเตอร์ โดยที่คอมพิวเตอร์แต่ละตัวจะส่งข้อมูลเข้าไปในระบบเครือข่าย จากนั้นจะใช้อุปกรณ์เชื่อมต่อระบบเครือข่ายที่เรียกว่าเราท์เตอร์ (router) ในการสื่อสารข้อมูลเข้าหากัน ซึ่งจะอาศัยการทำงานในชั้นอินเตอร์เน็ตของชุดโพรโตคอลทีซีพี/ไอพี

4.1 อินเตอร์เน็ตแอดเดรส (Internet address)

โฮสต์ที่เชื่อมต่อเข้ากับทีซีพี/ไอพีนั้นจะต้องมีเบอร์ประจำตัวของมันที่ต้องไม่ซ้ำกับเครื่องอื่นๆในระบบเครือข่ายที่มันเชื่อมต่ออยู่ เรียกหมายเลขนี้ว่าอินเตอร์เน็ตแอดเดรส (Internet Address) หรือ ไอพีแอดเดรส (IP address)

อินเตอร์เน็ตแอดเดรสเป็นตัวเลขขนาด 32 บิต ใช้บ่งบอกเลขประจำตัวของโฮสต์นั้นๆ เพื่อความสะดวกในการอ่าน จึงมักจะเขียนอินเตอร์เน็ตแอดเดรสเป็นเลขฐาน 10 จำนวน 4 ชุด แต่ละชุดจะถูกแบ่งด้วยเครื่องหมายจุด (dot) เรียกกันว่า dotted decimal notation ค่าขนาด 32 บิตนี้จะถูกแบ่งออกเป็นค่าขนาด 8 บิตต่อหนึ่งฟิลด์ เรียกกันว่า อ็อกเตท (octet) ซึ่งจะมีทั้งหมด 4 ฟิลด์ และจะกำหนดค่าในแต่ละฟิลด์ด้วยค่าเลขฐานสิบ เช่น อินเตอร์เน็ตแอดเดรสของระบบหนึ่งๆเป็น 10100001 1111 0110 0000 1010 0001 0101 จะได้ค่าในแต่ละอ็อกเตทคือ 161 246 10 21 ซึ่งเมื่อเขียนอยู่ในรูป dotted decimal notation จะได้เป็น 161.246.10.21

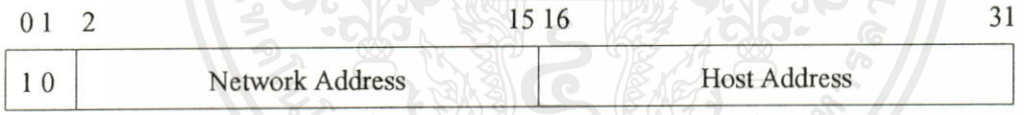
อินเตอร์เน็ตแอดเดรสขนาด 32 บิตนี้จะถูกมองเป็น 2 ส่วน ส่วนแรกเป็นหมายเลขเน็ตเวิร์ค (Network Number) ซึ่งเป็นเครือข่ายที่โฮสต์นั้นเชื่อมต่ออยู่ และส่วนที่เหลือเป็นหมายเลขท้องถิ่น (Local Host Address) หรือหมายเลขโฮสต์ของตัวเอง (Host Number)

ซึ่งจากอินเตอร์เน็ตแอดเดรสขนาด 32 บิตนี้ จะแบ่งระบบเครือข่ายออกเป็น 5 คลาสใหญ่ ดังนี้

ระบบเครือข่ายในคลาส A

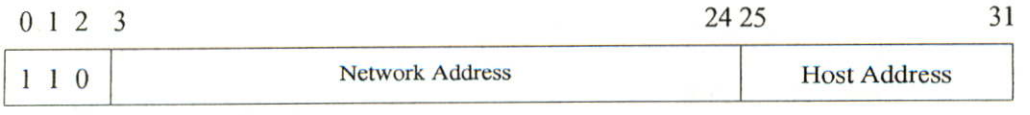
คลาส A จะใช้ 8 บิตแรกในการกำหนดหมายเลขเน็ตเวิร์ค และอีก 24 บิตที่เหลือในการกำหนดหมายเลขโฮสต์ โดยที่บิตแรกของหมายเลขเน็ตเวิร์คต้องเป็น 0 ทำให้สามารถกำหนดระบบเครือข่ายในคลาส A ได้ 2^7 ระบบเครือข่าย ซึ่งในการกำหนดอินเตอร์เน็ตแอดเดรสจะมีข้อกำหนดบางอย่างว่า ห้ามมีระบบเครือข่ายที่หมายเลขเครือข่ายมีค่าเป็น "0" ตลอด หรือเป็น "1" ตลอด ทำให้สามารถกำหนดระบบเครือข่ายในคลาส A ได้ 126 เครือข่าย (ระบบเครือข่ายหมายเลข 0 และ 127 จะถูกสงวนไว้) การกำหนดหมายเลขโฮสต์ให้มีขนาด 24 บิต ทำให้ระบบเครือข่ายในคลาส A แต่ละระบบเครือข่ายนี้มีโฮสต์ได้สูงสุด 2^{24} เครื่องด้วย ซึ่งก็จะมีข้อกำหนดเช่นเดียวกับการกำหนดหมายเลขเครือข่าย ทำให้มีเครื่องได้สูงสุด 16,777,214 เครื่อง

ระบบเครือข่ายในคลาส B



Class B จะใช้ 16 บิตแรกในการกำหนดหมายเลขเน็ตเวิร์ค และอีก 16 บิตที่เหลือในการกำหนดหมายเลขโฮส โดยที่ 2 บิตแรกของหมายเลขเน็ตเวิร์คต้องเป็น 10 ทำให้สามารถกำหนดระบบเครือข่ายใน class B ได้ 16,384 ระบบเครือข่าย (ระบบเครือข่ายหมายเลข 0 และ 16,383 จะถูกสงวนไว้) และระบบเครือข่ายในคลาส B แต่ละเน็ตเวิร์คนี้มีโฮสต์ได้สูงสุด 65534 เครื่อง

ระบบเครือข่ายในคลาส C



Class C จะใช้ 24 บิตแรกในการกำหนดหมายเลขเน็ตเวิร์ค และอีก 8 บิตที่เหลือในการกำหนดหมายเลขโฮสต์ โดยที่ 3 บิตแรกของหมายเลขเน็ตเวิร์คต้องเป็น 110 ทำให้สามารถกำหนด

ระบบเครือข่ายใน class C ได้ 2,097,152 ระบบเครือข่าย และระบบเครือข่ายในคลาส C แต่ละเน็ตเวิร์คนี้มีโฮสต์ได้สูงสุด 254 เครื่อง

ระบบเครือข่ายในคลาส D

1110	Network Address	Host Address

Class D เป็นอินเตอร์เน็ตแอดเดรสที่ใช้สำหรับการกระจายข้อมูลเฉพาะกลุ่มของระบบเครือข่ายที่เรียกว่า Multicast จะมี 4 บิตแรกเป็น 1110 ซึ่งจะอยู่ระหว่างหมายเลข 224 ถึง 239

ระบบเครือข่ายในคลาส E

1111	Network Address	Host Address

Class E เป็นกลุ่มไอพีแอดเดรส ที่ถูกสงวนไว้สำหรับการทำงานในอนาคต จะมี 4 บิตแรกเป็น 1111

4.2 อินเตอร์เน็ตแอดเดรสสงวน

นอกจากนี้ยังมีข้อกำหนดบางอย่างของการกำหนดไอพีแอดเดรสดังนี้

บรอดคาสท์แอดเดรส (Broadcast address) บิตที่ถูกกำหนดเป็นหมายเลขท้องถิ่นนั้นห้ามกำหนดทุกบิตในแอดเดรสนี้เป็น "1" หมด เนื่องจากในระบบการจัดตั้งอินเตอร์เน็ตแอดเดรสนั้น ถ้าทุกบิตของหมายเลขท้องถิ่นเป็น "1" หมด จะหมายความถึงสถานีงานทุกเครื่องที่อยู่บนระบบเครือข่ายนี้ เช่น 161.246.255.255 จะเป็นบรอดคาสท์แอดเดรสของระบบเครือข่ายคลาส B หมายเลข 161.246.0.0

ลูพแบ็คแอดเดรส (Loopback address) อินเตอร์เน็ตแอดเดรสในเครือข่ายคลาส A เบอร์ 127.0.0.0 จะถูกกำหนดไว้สำหรับทำเป็น "loopback" เพื่อใช้ในการทดสอบและการติดต่อสื่อสารระหว่างโปรเซส บนเครื่องนั้นๆเอง หมายความว่าเมื่อมีโปรแกรมต้องการส่งข้อมูลไปยัง loopback address ส่วนโพรโตคอลจะส่งข้อมูลนั้นกลับโดยไม่มีการส่งข้อมูลไปยังระบบเครือข่ายใดๆ เนื่องจากเป็นการส่งข้อมูลไปยังแอดเดรสที่หมายถึงตัวมันเอง

4.3 อินเตอร์เน็ตโพรโตคอล (Internet Protocol)

อินเตอร์เน็ตโพรโตคอลหรือไอพี เป็นโพรโตคอลหลักที่ทำหน้าที่นำส่งข้อมูลในชั้นอินเตอร์เน็ตนี้ ข้อมูลในส่วนของการทำงานในชั้นนี้จะเรียกว่า เดต้าแกรม (datagram) ชั้นนี้ไม่มีการรับประกันว่าข้อมูลที่ผ่านจากไอพีขึ้นไปนั้นจะถูกต้อง อาจจะมีการสูญหาย การซ้ำซ้อนของข้อมูล การไม่เรียงลำดับของข้อมูล ซึ่งจะเป็นหน้าที่ของโพรโตคอลชั้นบนขึ้นไปจะเป็นผู้จัดการ ไอพีจะไม่มีการตรวจสอบข้อผิดพลาดของข้อมูล จะมีเฉพาะผลรวมตรวจสอบส่วนหัวเท่านั้น ไม่มีการส่งซ้ำ (re-transmission) ไม่มีการควบคุมการรับส่ง (flow control)

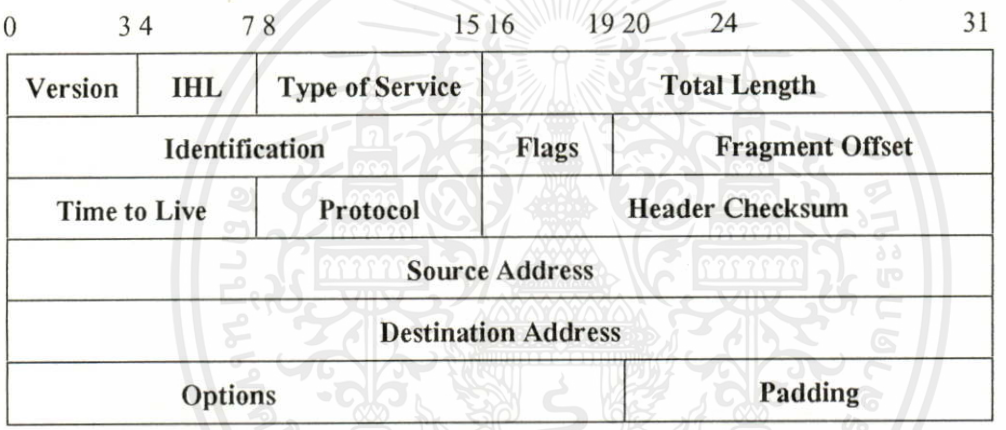
ไอพีมีหน้าที่หลักคือรับข้อมูลมาจากทีซีพีหรือยูดีพี แล้วสร้างเป็นเดต้าแกรม จากนั้นจึงทำการค้นหาเส้นทางที่จะใช้นำส่งเดต้าแกรมตัวนี้ไปยังแอดเดรสปลายทางต่อไป ซึ่งเดต้าแกรมแต่ละตัวจะถูกจัดเส้นทางเป็นอิสระต่อกัน ในการหาเส้นทางนั้น ไอพีจำเป็นจะต้องมีความสามารถในการแลกเปลี่ยนข่าวสารที่ใช้ กับโพรโตคอลตัวอื่นด้วย เช่น อาร์ไอพี (RIP : Routing Information Protocol) จีจีพี (GGP : Gateway-Gateway Protocol) อีจีพี (EGP : External Gateway Protocol) เป็นต้น

การส่งข้อมูลของไอพีจะต้องบอกอินเทอร์เนตแอดเดรสของเครื่องปลายทาง งานหลักของไอพีคือ ค้นหาเส้นทางเพื่อที่จะส่งเดต้าแกรมไปยังปลายทาง ในการที่จะให้อุปกรณ์ระหว่างทาง (Intermediate System) ส่งเดต้าแกรมจะต้องมีข้อมูลในส่วนหัวของเดต้าแกรมเป็นตัวบอก ในส่วนนี้จะบอกถึงอินเทอร์เนตแอดเดรสต้นทางและปลายทางขนาด 32 บิต โพรโตคอลนัมเบอร์ (protocol number) และ ผลรวมตรวจสอบ ตำแหน่งต้นทางและปลายทางใช้เพื่อบอกให้ทั้ง 2 ฝ่ายทราบว่าข้อมูลมาจากที่ใดและจะไปที่ใด โพรโตคอลนัมเบอร์ใช้บอกไอพีว่าจะส่งเดต้าแกรมไปยังชั้นทีซีพีหรืออื่นๆ ผลรวมตรวจสอบใช้เพื่อตรวจสอบว่าข้อมูลของส่วนหัวไม่มีข้อผิดพลาดระหว่างการส่ง แฟลก (flag) และ แฟรกเมนต์ออฟเซ็ต (fragment offset) ใช้เพื่อแบ่งเดต้าแกรม

เป็นขนาดเล็กๆ เนื่องจากอาจจะส่งข้อมูลขนาดใหญ่เกินกว่าที่จะให้ระบบเครือข่ายนั้นส่งได้ทั้งหมดภายในการส่งครั้งเดียว เวลาคงอยู่เป็นตัวเลขที่ใช้บอกถึงช่วงเวลาที่เดต้าแกรมตัวนี้จะคงอยู่ในระบบเครือข่ายได้ ค่านี้จะลดลงเรื่อยๆ เมื่อมีการประมวลผลเดต้าแกรมในแต่ละระบบ เมื่อค่านี้เป็น "0" เดต้าแกรมนี้ก็จะถูกกำจัดทิ้งไป เพื่อป้องกันการส่งข้อมูลวนในเน็ตเวิร์ค

รูปแบบส่วนหัวอินเตอร์เน็ตโพรโตคอล (Internet Protocol Header Format)

รูปที่ 4.1 ส่วนหัวของอินเตอร์เน็ตโพรโตคอล



ประกอบด้วยส่วนต่างๆดังนี้

เวอร์ชั่น (Version) ขนาด 4 บิต

เป็นเลขรุ่นของโพรโตคอลไอพี ที่ใช้กันอยู่ในปัจจุบันเป็นรุ่นที่ 4

ความยาวส่วนหัว (IHL - Internet Header Length) ขนาด 4 บิต

ใช้เก็บความยาวส่วนหัวในรูปของ 32 บิตเวิร์ด ค่าต่ำสุดที่มีได้เท่ากับ 5 นั่นคือส่วนหัวของเดต้าแกรมหนึ่งๆต้องมีความยาวอย่างน้อย 160 บิต ถ้าเดต้าแกรมนี้มีข้อมูลส่วน option เพิ่มเติมด้วยจะต้องทำการเติม "0" ไปเพื่อให้ได้ความยาวส่วนหัวเป็นจำนวนเท่าของ 32 บิตเวิร์ด

Precedence และ ชนิดของบริการ (Type of Service) ขนาด 8 บิต

ใช้บ่งบอกคุณภาพของข่าวสารบริการว่าจะประมวลผลเดต้าแกรมตัวนี้อย่างไร

ตารางที่ 4.1 รูปแบบชนิดของบริการ (ToS format)

ตำแหน่งบิต	ความหมาย
บิต 0-2	Precedence Level 0-7, 0 is normal and level 7 is highest priority
บิต 3	0=Normal Delay, 1=Low Delay
บิต 4	0=Normal Throughput, 1=High Throughput
บิต 5	0=Normal Reliability, 1=High Reliability
บิต 6-7	สำรองไว้ใช้ในอนาคต จะมีค่าเป็น 0

ความยาวทั้งหมด (Total Length) ขนาด 16 บิต

เป็นความยาวทั้งหมดของเดต้าแกรม ซึ่งนับรวมทั้งส่วนหัวและส่วนของข้อมูลในหน่วยอ็อกเต็ท ซึ่งจะมีค่าได้สูงสุด 65535 อ็อกเต็ท ในข้อกำหนดของไอพี โฮสต์ในระบบเครือข่ายต้องสามารถรองรับเดต้าแกรมที่มีความยาวได้ต่ำสุด 576 อ็อกเต็ท

ตัวบ่งชี้ (Identification) ขนาด 16 บิต

กำหนดโดยผู้ส่ง ใช้สำหรับการแยกเดต้าแกรมเป็นเดต้าแกรมย่อยๆ

แฟลก (Flags) ขนาด 3 บิต

ตารางที่ 4.2 รูปแบบแฟลกของอินเตอร์เน็ตโพรโตคอล

ตำแหน่งบิต	ความหมาย
บิต 0	Reserved, must be zero
บิต 1	(DF) 0=May Fragment, 1=Don't Fragment
บิต 2	(MF) 0=Last Fragment, 1=More Fragments

แฟรกเมนต์ออฟเซ็ต (Fragment Offset) ขนาด 13 บิต

ใช้ชี้ตำแหน่งของแฟรกเมนต์ (fragment) ในเดต้าแกรมเดิมก่อนทำการแฟรกเมนเตชั่น ค่าของแฟรกเมนต์ออฟเซ็ตจะอยู่ในหน่วยของ 8 อ็อกเต็ท

เวลาคงอยู่ (TTL - Time to Live) ขนาด 8 บิต

ใช้แสดงถึงระยะเวลาสูงสุดที่เดต้าแกรมสามารถอยู่ในระบบอินเทอร์เนตได้เพื่อป้องกันไม่ให้มีเดต้าแกรมไหลวนอยู่ในระบบเครือข่ายที่เกิดผิดพลาดโดยไม่มีการสิ้นสุด โฮสต์ที่ส่งจะกำหนดค่านี้ และเราท์เตอร์ที่นำเดต้าแกรมไปทำการประมวลผลจะลดค่า TTL ลง 1 ค่า ทุกครั้งที่เดต้าแกรมถูกประมวลผล ซึ่งถ้าฟิลด์นี้มีค่าเป็น "0" เดต้าแกรมตัวนี้จะถูกทำลายทิ้งโดยอัตโนมัติ ทั้งนี้แล้วแต่ด้วยว่าใช้โพรโตคอลในการหาเส้นทางแบบใด

โพรโตคอล (Protocol) ขนาด 8 บิต

บอกถึงโพรโตคอลในเลเยอร์ถัดไปที่จะนำข้อมูลในเดต้าแกรมตัวนี้ไปใช้งาน เช่น "1" คือ โพรโตคอลไอซีเอ็มพี "6" คือ โพรโตคอลทีซีพี เป็นต้น

ผลรวมตรวจสอบส่วนหัว (Header Checksum) ขนาด 16 บิต

จะมีการคำนวณใหม่ทุกครั้งถ้ามีบางฟิลด์ของส่วนหัวเปลี่ยนค่าไป เช่น ค่า TTL

ที่อยู่ต้นทาง (Source Address) ขนาด 32 บิต

เป็นหมายเลขอินเตอร์เน็ตแอดเดรสของเครื่องที่ส่งเดต้าแกรมนี้มา

ที่อยู่ปลายทาง (Destination Address) ขนาด 32 บิต

เป็นหมายเลขอินเตอร์เน็ตแอดเดรสของเครื่องปลายทางที่จะส่งเดต้าแกรมตัวนี้ไป

ออปชั่น (Options) ขนาดเปลี่ยนแปลงได้

ออปชั่นจะใช้เป็นตัวกำหนดถึงข้อมูลที่มาในเดต้าแกรม ซึ่งอาจจะไม่มีหรือมีได้หลาย ๆ ออปชั่นก็ได้ สามารถยาวได้ถึง 40 อ็อกเต็ท แต่ปกติจะไม่มีการใช้งานฟิลด์นี้

แพดดิ้ง (Padding) ขนาดเปลี่ยนแปลงได้

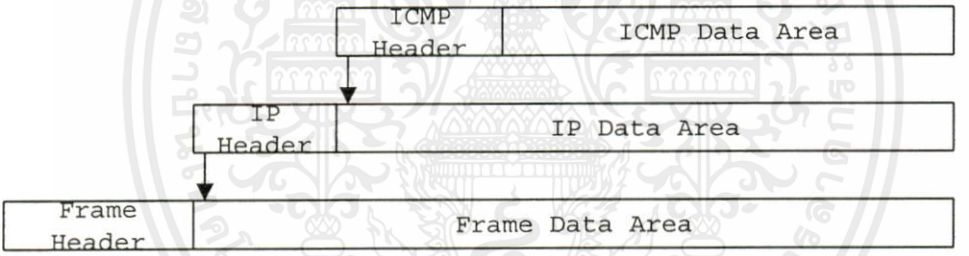
ใช้เพิ่ม "0" เข้าไปเพื่อทำให้ส่วนหัวของอินเทอร์เนตมีความยาวเป็นจำนวนเท่าของ 32 บิต

4.4 โพรโตคอลไอซีเอ็มพี (ICMP - Internet Control Message Protocol)

ไอซีเอ็มพีโพรโตคอลเป็นโพรโตคอลที่ใช้รายงานข้อผิดพลาดที่เกิดกับการประมวลผลเดต้าแกรมเท่านั้น จะไม่มีการส่งไอซีเอ็มพีแมสเซสสำหรับข้อผิดพลาดที่เกิดกับไอซีเอ็มพีแมสเซสเอง เพื่อแก้ปัญหาการรายงานข้อผิดพลาดกันไม่มีที่สิ้นสุด และไอซีเอ็มพีแมสเซสจะรายงานเฉพาะข้อผิดพลาดที่เกิดกับแฟรกเมนต์ซีโร่ (fragment zero) ของเดต้าแกรมที่ทำการแฟรกเมนต์เท่านั้น

ไอซีเอ็มพีแมสเซส จะใช้รูปแบบของส่วนหัวไอพีในการส่ง ซึ่งข้อมูลนี้จะอยู่ในส่วนข้อมูลของเดต้าแกรมของไอพีดังรูปที่ 4.2 โดยอ็อกเต็ทแรกในส่วนฟิลด์ข้อมูลของเดต้าแกรมจะเป็นฟิลด์ชนิดของไอซีเอ็มพี ซึ่งใช้บ่งบอกถึงรูปแบบของข้อมูลที่มีอยู่

รูปที่ 4.2 การจัดโครงสร้างเฟรมของไอซีเอ็มพีแมสเซส



ไอซีเอ็มพีจะมีการใช้งานอยู่มากมาย เช่น ในกรณีที่เดต้าแกรมเดินทางไปไม่ถึงปลายทาง ซึ่งอาจจะเกิดมาจากข้อมูลมีข้อผิดพลาด หรือค่า Time to Live ของเดต้าแกรมนั้นหมด หรือเส้นทางที่ใช้ในการส่งมีปัญหา โมดูลไอพีที่ตรวจพบปัญหานั้นก็จะสร้างข่าวสารไอซีเอ็มพีส่งกลับไปผู้ส่ง

ชนิดของข่าวสารข้อผิดพลาด (ICMP Error Messages)

ตารางที่ 4.3 ชนิดของข่าวสารข้อผิดพลาด

Message	Description
Destination Unreachable	A Datagram cannot reach its destination hosts, utility, or application.
Time Exceeded	The Time-to-Live has expired at a router, or the Fragment Reassembly Time has expired at a destination host.
Parameter problem	There is a bad parameter in the IP header.
Source Quench	A router or destination is congested. It is recommended that systems should not send Quench messages.
Redirect	A host has routed a datagram to the wrong local router.

ชนิดของประเภทของข่าวสารไอซีเอ็มพี

ตารางที่ 4.4 ชนิดของประเภทของข่าวสารไอซีเอ็มพี

Type Code	ICMP Messages
0	Echo reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
17	Address Mask Request
18	Address Mask Reply

4.5 โพรโตคอลเออาร์พี (ARP - Address Resolution Protocol)

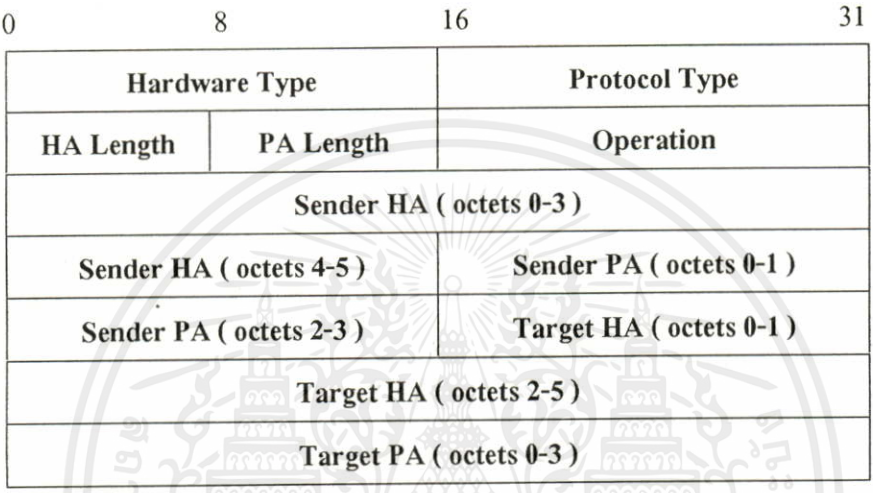
อินเตอร์เน็ตใช้ไอพีแอดเดรสขนาด 32 บิตในการกำหนดเบอร์ประจำตัวของโฮสต์ในระบบเครือข่ายเสมือนของมัน ส่วนในการส่งข้อมูลเข้าหากันในระบบเครือข่ายจริงๆบนระบบเครือข่ายแบบต่างๆ เช่น อีเทอร์เน็ต โทเก้นริง หรือ FDDI จะสามารถทำงานได้ก็ต่อเมื่อทราบแอดเดรสระบุปลายทางที่จะสามารถส่งข้อมูลได้อย่างถูกต้อง แอดเดรสดังกล่าวนี้ไม่ใช่อินเตอร์เน็ตแอดเดรสและแยกออกจากระบบของอินเตอร์เน็ตแอดเดรส เรียกว่า MAC (Medium Access Control) Number ซึ่งบางครั้งอาจจะเรียกว่า ฟิสิคัลแอดเดรส, ฮาร์ดแวร์แอดเดรส หรือ ลิงค์แอดเดรส อินเตอร์เน็ตแอดเดรสนั้นเป็นเบอร์ที่ถูกกำหนดขึ้นภายหลัง แต่ MAC นั้นผู้ผลิตฮาร์ดแวร์นั้นจะเป็นผู้ที่กำหนดมาตั้งแต่ตอนสร้างฮาร์ดแวร์นั้น ซึ่งส่วนใหญ่มักจะเป็นการ์ดเชื่อมโยงระบบเครือข่าย (Network interface card - NIC)

ในระบบเครือข่ายแบบอีเทอร์เน็ต อุปกรณ์ทุกตัวในระบบเครือข่ายจะต้องมีอีเทอร์เน็ตแอดเดรสประจำตัวที่ถูกกำหนดมาโดยผู้ผลิตอยู่ เบอร์นี้จะเป็นตัวเลขขนาด 48 บิตและมักจะเขียนโดยใช้ตัวเลขฐาน 16 แยกจากกันด้วยเครื่องหมาย ":" เช่น 00:40:05:29:F0:F4 อีเทอร์เน็ตแอดเดรสถูกกำหนดโดย IEEE (Insititute of Electrical and Electronics Engineers) และต้องเป็นเบอร์เฉพาะของการ์ดแต่ละแผ่น ตัวเลขของอีเทอร์เน็ตแอดเดรสจะแบ่งออกเป็น 2 ส่วน 3 ไบท์แรกจะเป็นเบอร์ประจำตัวของผู้ผลิต ซึ่งผู้ผลิตแต่ละที่จะถูกกำหนดเบอร์ประจำตัวโดย IEEE อีก 5 ไบท์ที่เหลือจะเป็นเบอร์ที่ผู้ผลิตกำหนดให้กับการ์ดนั้น เมื่อฮาร์ดแวร์เกิดความเสียหายและต้องการเปลี่ยนอุปกรณ์เชื่อมต่อใหม่ก็จะทำให้ฟิสิคัลแอดเดรสของระบบนั้นๆเปลี่ยนไปด้วย และเนื่องจากอีเทอร์เน็ตใช้ฟิสิคัลแอดเดรสขนาด 48 บิตจึงไม่สามารถจะเข้ารหัสในรูปของไอพีแอดเดรสขนาด 32 บิตได้

สมมติว่ามีโฮสต์ 2 ตัวต่ออยู่บนระบบเครือข่ายเดียวกันอยู่ แต่ละตัวถูกกำหนดให้มีไอพีแอดเดรสเป็นเบอร์ IA และ IB มีฟิสิคัลแอดเดรสเป็น Pa และ Pb การทำเช่นนี้เพื่อต้องการให้โปรแกรมที่ทางานในเลเยอร์สูงๆสามารถทำงานได้ดีโดยไม่ต้องทราบถึงรายละเอียดของการเชื่อมต่อกันจริงทางกายภาพ เพียงแต่ต้องทราบแค่ไอพีแอดเดรสเท่านั้น แต่ในการสื่อสารข้อมูลจริงๆบนระบบเครือข่าย จะต้องทราบถึงรายละเอียดของฟิสิคัลแอดเดรสที่ขึ้นกับว่าใช้ระบบเครือข่ายในการส่งข้อมูลเป็นแบบใด เช่น โฮสต์ A ต้องการส่งแพ็กเก็ตข้อมูลไปหาโฮสต์ B บนระบบเครือข่ายเดียวกัน แต่ A จะทราบเฉพาะอินเตอร์เน็ตแอดเดรสของ B เท่านั้น A จะทำการแปลงอินเตอร์เน็ตแอดเดรสของ B ไปเป็นฟิสิคัลแอดเดรสจริงๆได้โดยการใช้โพรโตคอล **ARP** หรือ Address Resolution Protocol เป็นโพรโตคอลในการหาฟิสิคัลแอดเดรสของไอพีแอดเดรสที่รู้แล้ว เช่น A จะติดต่อกับ B โดยที่ A รู้เฉพาะไอพีแอดเดรสของ B ก่อนที่จะส่งข้อมูลกันจริงๆได้ A จะต้องรู้ฟิสิคัลแอดเดรส

ของ B ก่อนด้วยการใช้โพรโตคอลเออาร์พี ส่วน **RARP** (Reverse Address Resolution Protocol) จะเป็นโพรโตคอลที่ทำหน้าที่ตรงข้ามกับ ARP คือ ใช้หาว่าสถานีงานที่มีแมคแอดเดรสเบอร์หนึ่งๆ จะมีอินเตอร์เน็ตแอดเดรสเป็นเบอร์อะไร

รูปที่ 4.3 โครงสร้างของเออาร์พี



ซึ่งประกอบด้วยส่วนต่างๆดังนี้

Hardware Type

เป็นส่วนที่แสดงชนิดของฮาร์ดแวร์ ที่ใช้ในระดับชั้นเครือข่าย เช่น ถ้าเป็นอีเทอร์เน็ตจะมีค่าเป็น "1" หรือ ถ้าเป็นระบบเครือข่าย IEEE 802 จะมีค่าเป็น "6"

Protocol Type

แสดงถึงค่าโพรโตคอลแอดเดรสที่ใช้ในระดับเครือข่าย เช่น ถ้าเป็น 0800H หมายถึงไอพีแอดเดรส

Hardware Address Length

แสดงขนาดของฮาร์ดแวร์แอดเดรสเป็นไบท์ ซึ่งจะทำให้สามารถใช้เออาร์พีกับโครงสร้างแอดเดรสแบบใดก็ได้ เช่น ไอพี มีขนาด 32 บิท ส่วนอีเทอร์เน็ต จะมีขนาดเป็น 48 บิท

Protocol Address Length

แสดงขนาดของโพรโตคอลแอดเดรสเป็นไบท์ สำหรับทีซีพี/ไอพีจะมีค่าเป็น "4"

Operation

สำหรับแสดงรายละเอียดหน้าที่ของแพ็กเก็ต

ARP Request มีค่าเป็น "1"

ARP Reply มีค่าเป็น "2"

RARP Request มีค่าเป็น "3"

RARP Reply มีค่าเป็น "4"

Sender Hardware Address

สำหรับฮาร์ดแวร์แอดเดรสของโฮสต์ที่ส่งแพ็กเก็ต

Sender Protocol Address

เป็นโพรโตคอลแอดเดรสของโฮสต์ที่ส่งแพ็กเก็ต

Target Hardware Address

เมื่อมีการใช้ ARP request ส่วนนี้จะถูกส่งโดยกำหนดค่าในฟิลด์เป็น "0" ทั้งหมด ซึ่งโฮสต์ปลายทางจะทำการใส่ค่าแอดเดรสลงไปในฟิลด์นี้เมื่อส่งแพ็กเก็ตตอบกลับมาเป็น ARP Reply

Target Protocol Address

เมื่อมีการใช้ ARP request ส่วนนี้จะถูกส่งโดยกำหนดค่าในฟิลด์เป็น "0" ทั้งหมด ซึ่งโฮสต์ปลายทางจะทำการใส่ค่าแอดเดรสลงไปในฟิลด์นี้เมื่อส่งแพ็กเก็ตตอบกลับมาเป็น ARP Reply

4.6 การหาเส้นทาง (Routing)

ในการที่โฮสต์จะส่งเดต้าแกรมลงไปในระบบเครือข่ายนั้นจะต้องระบุอินเตอร์เน็ตแอดเดรสต้นทางและปลายทางเสียก่อน จากนั้นเราท์เตอร์จะนำข้อมูลส่วนนี้ในเดต้าแกรมมาวิเคราะห์หาเส้นทางที่เหมาะสมในการส่งเดต้าแกรมตัวนั้นโดยดูจากตารางข่าวสารเส้นทาง (routing information table) ของมัน เราท์เตอร์จะมีตารางเส้นทางที่อาจจะได้ข้อมูลแบบสเตติกหรือไดนามิค ในการจัดเส้นทางแบบสเตติก เราท์เตอร์จะมีตารางเส้นทางตายตัวคงที่ ซึ่งไม่เหมาะในระบบเครือข่ายขนาดใหญ่ที่อาจจะมีการเปลี่ยนแปลงเส้นทางได้ตลอดเวลา ส่วนการจัดเส้นทางแบบไดนามิค

นั้น เราท์เตอร์จะสามารถปรับเปลี่ยนตารางเส้นทางให้เหมาะสมกับระบบเครือข่ายขณะนั้นได้ตลอด

เราท์เตอร์จะใช้เมตริกซ์ (metric) เป็นหน่วยวัดระยะทาง เพื่อหาระยะทางที่สั้นที่สุด โดยทั่วไปแล้ว เมตริกซ์จะวัดจากว่ามีเราท์เตอร์กี่ตัวที่อยู่ในเส้นทางถัดไป (hop) เราท์เตอร์จะส่งเดต้าแกรมไปยังเส้นทางที่มี hop count หรือเมตริกซ์น้อยสุด โดยอาจจะพิจารณาองค์ประกอบอื่นๆด้วย เช่น เส้นทางนี้มีการจราจรมากน้อยเพียงใด หรือ มีความน่าเชื่อถือเพียงใด การหาเส้นทางนั้นจะใช้อัลกอริธึมในการหาเส้นทางอยู่ 2 แบบ คือ

1. Distance Vector algorithm
2. Link State algorithm

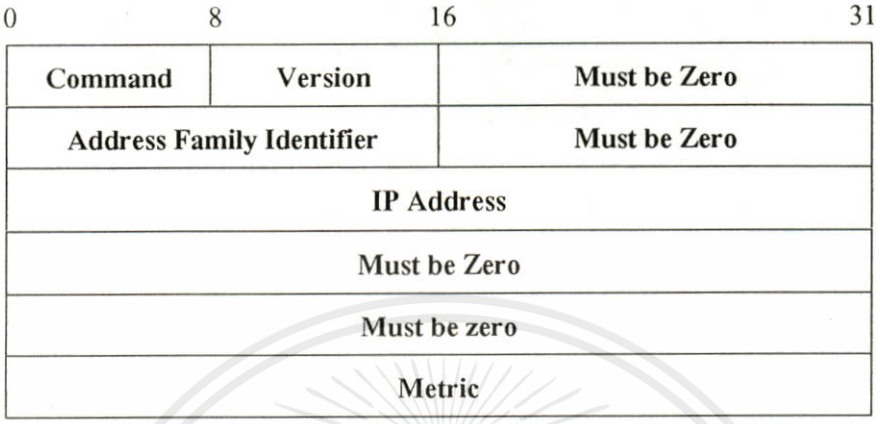
ซึ่งเราท์เตอร์จะใช้โพรโตคอลในการหาเส้นทางภายในระบบเครือข่าย (intra-network) ที่เรียกว่า **Interior Gateway Protocol (IGP)**

4.7 โพรโตคอลอาร์ไอพี (RIP - Routing Information Protocol)

โพรโตคอลอาร์ไอพีเป็น Interior Gateway Protocol ที่ใช้อัลกอริธึมแบบ Distance Vector algorithm ทุกๆฮอบในระบบเครือข่ายจะถูกกำหนดค่าไว้เป็น 1 เราท์เตอร์จะเลือกส่งเดต้าแกรมไปในเส้นทางที่สั้นที่สุดโดยดูจากผลรวมของฮอบในเส้นทางทั้งหมดว่าเส้นทางใดมีค่าฮอบน้อยที่สุด อาร์ไอพีมีการทำงานแบบง่ายๆไม่ซับซ้อนทำให้เหมาะกับการทำงานภายในระบบเครือข่ายขนาดเล็ก แต่จะไม่มีประสิทธิภาพที่ดีนักในระบบเครือข่ายขนาดใหญ่ เนื่องจากการใช้เมตริกซ์ของอาร์ไอพีกำหนดว่าค่าเมตริกซ์สูงสุดในเส้นทางคือ 15 ซึ่งในระบบเครือข่ายขนาดใหญ่อาจจะมีเราท์เตอร์มากกว่านี้ทำให้ไม่สามารถใช้อาร์ไอพีส่งเดต้าแกรมไปยังเส้นทางที่มีฮอบเป็น 16 ได้ นอกจากนี้อาร์ไอพีจะกำหนดค่าฮอบเป็น 1 เสมอโดยไม่สนใจว่าฮอบนั้นๆจะมีเส้นทางการเชื่อมต่อที่เร็วหรือช้าเพียงใด

ข้อมูลการจัดเส้นทางของอาร์ไอพีจะใช้บริการของยูดีพีพอร์ทหมายเลข 520 แพ็กเก็ตของอาร์ไอพีจะมีส่วนหัวขนาด 32 บิต และอาจจะมีข่าวสารสำหรับระบบเครือข่ายแต่ละเครือข่ายได้มากถึง 25 เครือข่ายต่อหนึ่งแพ็กเก็ต ข่าวสารทั้งหมดจะมีความยาวสูงสุดได้ 512 อ็อกเต็ด ซึ่งถ้าต้องการส่งข้อมูลของระบบเครือข่ายมากกว่านี้จะต้องส่งข่าวสารตัวใหม่ไปแทน

รูปที่ 4.4 โครงสร้างของอาร์ไอพี



คำสั่ง (Command) ขนาด 8 บิต

เป็นคำสั่งของอาร์ไอพี ซึ่งมีรายละเอียดดังนี้

ตารางที่ 4.5 ชุดคำสั่งของไออาร์พี

Command	Meaning
1	Request for routing table information
2	Response containing routing table information
5	Reserved for SUN Microsystems

เวอร์ชั่น (Version) ขนาด 4 บิต

เป็นเลขรุ่นของโพรโตคอลอาร์ไอพี ที่ใช้กันอยู่ในปัจจุบันเป็นรุ่นที่ 1

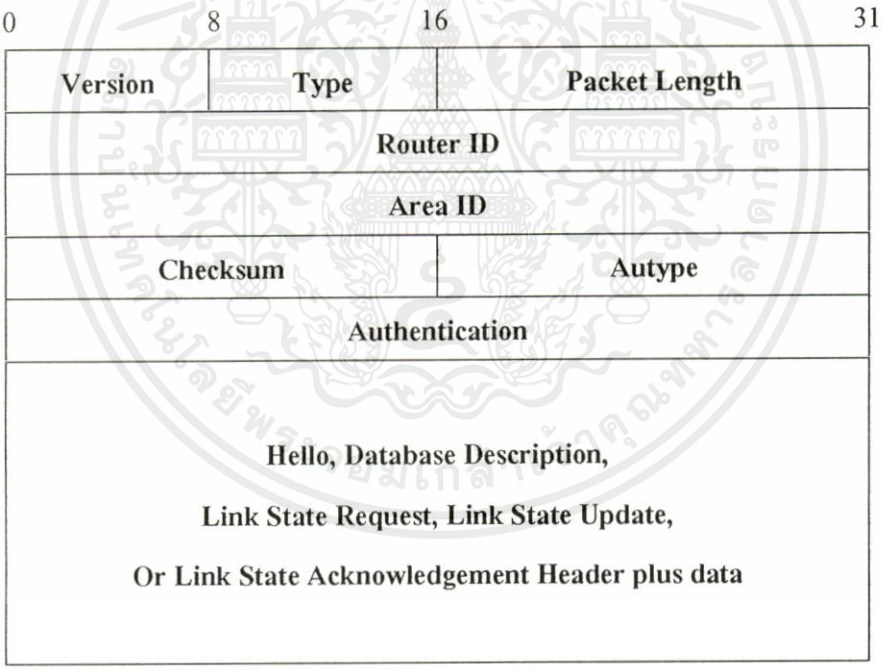
ตัวบ่งกลุ่มแอดเดรส (Address Family Identifier) ขนาด 16 บิต

กลุ่มแอดเดรสที่จะใช้ส่งแพ็กเก็ตอาร์ไอพี ที่ใช้กันขณะนี้คือ ไอพี มีค่าเป็น 2

4.8 โพรโตคอลโอเอสพีเอฟ (OSPF – Open Shortest Path First Protocol)

โพรโตคอลโอเอสพีเอฟเป็นโพรโตคอลในการจัดเส้นทางที่ทำงานโดยใช้อัลกอริธึมแบบ Link state ซึ่งมีข้อดีมากกว่าอาร์ไอพีหลายด้าน เช่น โครงสร้างในการจัดการข่าวสารเส้นทางจะเป็นแบบลำดับชั้น (hierachical topology) สามารถปรับเปลี่ยนเส้นทางได้อย่างมีประสิทธิภาพและรวดเร็ว เหมาะสมในระบบเครือข่ายขนาดใหญ่ และสามารถคำนวณหาเส้นทางที่มีประสิทธิภาพดีที่สุดได้หลายทาง ทำให้สามารถแบ่งภาระการจราจร (traffic load) ไปในเส้นทางหลายๆทางได้ โพรโตคอลโอเอสพีเอฟจะถูกส่งโดยไอพีและจะมีฟิลด์ไอพีโพรโตคอลเป็น 89

รูปที่ 4.5 รูปแบบส่วนหัวของแพ็กเก็ตโอเอสพีเอฟ



เวอร์ชั่น (Version) ขนาด 8 บิต

เป็นเลขรุ่นของโพรโตคอลโอเอสพีเอฟ

ประเภท (Type) ขนาด 8 บิต

แพ็กเก็ตของโพรโตคอลโอเอสพีเอฟจะมีได้ 5 แบบดังนี้

ตารางที่ 4.6 ประเภทของโอเอสพีเอฟแพ็กเก็ต

Type	Meaning
1	Hello
2	Database Description
3	Link State Request
4	Link State Update
5	Link State Ack

ความยาวแพ็กเก็ต (Packet Length) ขนาด 16 บิต

เป็นความยาวของแพ็กเก็ตโอเอสพีเอฟซึ่งนับรวมความยาวส่วนหัวด้วย

บทที่ 5

โพรโตคอลในชั้นโฮสต์ทูโฮสต์

5.1 ทรานสมิตชั่นคอนโทรลโพรโตคอล (Transmission Control Protocol)

ทีซีพีเป็นโพรโตคอลหลักที่ทำงานในชั้นโฮสต์ทูโฮสต์ ใช้เพื่อตรวจสอบความถูกต้องของข้อมูลที่รับส่งโดยไอพีในชั้นอินเตอร์เน็ต เช่น การรวบรวมข้อมูล (Re-assembling) ในกรณีที่ข้อมูลเข้ามาไม่มีการเรียงลำดับกัน การขอให้ส่งข้อมูลใหม่ในกรณีที่ไม่ได้รับข้อมูล (Retransmit) การตัดข้อมูลทิ้ง (Discard) ในกรณีที่ได้รับข้อมูลซ้ำ เป็นต้น ข้อมูลที่ผ่านชั้นนี้ไปจะถือเป็นข้อมูลที่ปราศจากข้อผิดพลาด (Error free) และข้อมูลที่ส่งขึ้นไปบนชั้นบนจะเป็นลักษณะไบท์ต่อเนื่อง (Byte Stream) การติดต่อสื่อสารกันจะเป็นลักษณะ Connection oriented คือ ต้องมีการสร้างการเชื่อมต่อขึ้นมาก่อน เหมาะกับการสื่อสารที่ใช้เวลานานและมีข้อมูลมาก สามารถทำงานแบบมัลติเปิลคอนเนคชั่น (multiple connection) คือการที่สามารถใช้งานได้หลายอย่างในการเชื่อมต่อแต่ละครั้ง

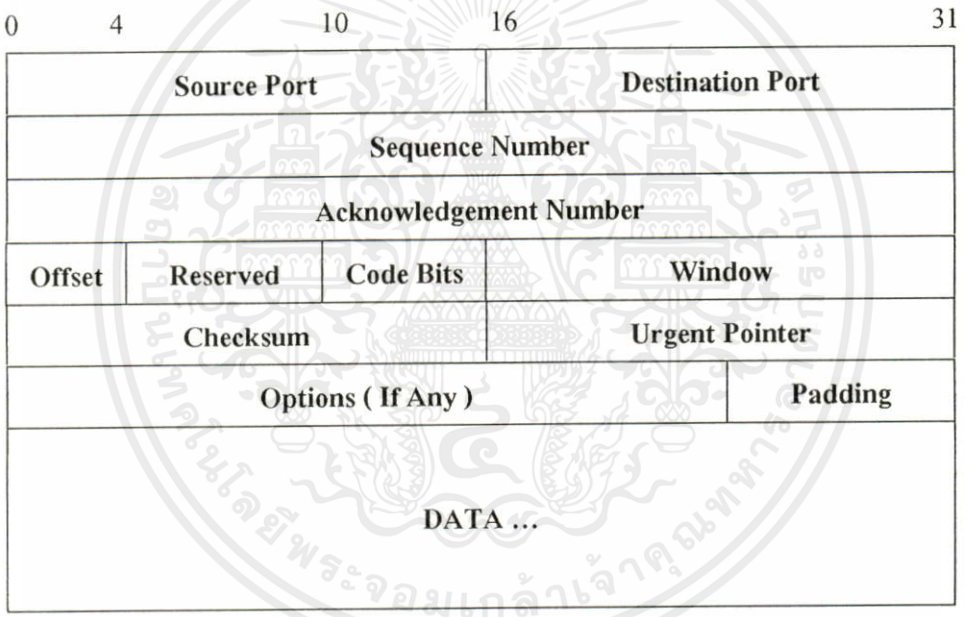
ทีซีพีจะมีส่วนหัวที่เพิ่มเข้าไปในส่วนข้อมูลอย่างน้อยที่สุด 20 อ็อกเต็ท ซึ่งจะมีส่วนที่บอกถึงหมายเลขพอร์ต (port number) และหมายเลขลำดับ (sequence number) หมายเลขพอร์ตเป็นตัวบอกกระบวนการที่จะใช้อ้างอิง ส่วนหมายเลขลำดับจะใช้เพื่อระบุถึงลำดับของข้อมูลที่จะส่ง

ส่วนของการจัดการการเชื่อมต่อ ในการที่จะให้การส่งเดต้าแกรมไปยังเป้าหมายได้ ผู้รับจะต้องส่งตัวตอบรับ (acknowledgement) ตอบกลับมาที่มีฟิลด์ของหมายเลขตอบรับ (Acknowledgement Number) ยกตัวอย่างเช่น การส่งแพ็กเก็ตโดยมีตัวตอบรับเป็นหมายเลข 1100 หมายถึงสามารถรับข้อมูลอย่างถูกต้องจนถึงอ็อกเต็ทหมายเลข 1100 ถ้าผู้ส่งยังไม่ได้รับตัวตอบรับในเวลาที่เหมาะสมก็จะส่งข้อมูลเดิมอีกครั้งหนึ่ง มีการใช้วินโดว์ (window) ในการควบคุมจำนวนข้อมูลที่จะส่งไปในแต่ละครั้ง คือจะไม่รอตัวตอบรับทุกครั้งที่จะส่งเดต้าแกรมถัดไป แต่จะส่งข้อมูลชุดต่อไปได้เลย ตามขนาดของบัฟเฟอร์ที่ยังว่างอยู่ ซึ่งกำหนดไว้ในส่วนของวินโดว์ เรียกวิธีการนี้ว่า สไลด์ดิ้งวินโดว์ (sliding window) โดยทั่วไปแล้ว ผู้ส่งข้อมูลเมื่อส่งข้อมูลไปแล้วจะต้องรอรับตัว ACK ก่อนจึงจะส่งข้อมูลชุดต่อไปได้ เรียกว่า Positive acknowledgement ซึ่งจะเสียเวลาว่างมากในการรอคอยจึงมีการใช้เทคนิคแบบ sliding window ซึ่งจะสามารถส่งข้อมูลไปเรื่อยๆ จนกระทั่ง

ข้อมูลที่ส่งมีจำนวนแพ็กเก็ตเท่ากับขนาดของวินโดว์ จึงจะหยุดส่งหรือส่งใหม่ถ้ารอการตอบรับนานเกินไป (time out) แต่โดยปกติขณะที่ส่งข้อมูลออกไปก็จะมีการใช้ ACK ตอบกลับมาทำให้สามารถส่งไปเรื่อยๆได้ เมื่อฝ่ายรับได้รับข้อมูลเรียบร้อยแล้วก็จะส่งให้การทำงานในชั้นถัดไปทำการประมวลผลต่อ

โครงสร้างส่วนหัวของทีซีพีเซ็กเมนต์

รูปที่ 5.1 โครงสร้างส่วนหัวของทีซีพีเซ็กเมนต์



ประกอบไปด้วยฟิลด์ต่างๆดังนี้

พอร์ตต้นทาง (Source Port) ขนาด 16 บิต

เป็นหมายเลขพอร์ตของผู้ส่ง

พอร์ตปลายทาง (Destination Port) ขนาด 16 บิต

เป็นหมายเลขพอร์ตผู้รับ

หมายเลขลำดับ (Sequence Number) ขนาด 32 บิต

เป็นหมายเลขลำดับของชุดข้อมูล ใช้บอกถึงลำดับข้อมูลที่ส่งให้

หมายเลขตอบรับ (Acknowledgement Number) ขนาด 32 บิต

ถ้ามีบิตควบคุมของส่วนตอบรับเซตไว้ข้อมูลในฟิลด์นี้จะมีค่าเป็นหมายเลขลำดับชุดข้อมูลถัดไปที่ผู้ส่งคาดว่าจะได้รับจากอีกฝ่าย

ดาต้าออฟเซ็ต (Data Offset) ขนาด 4 บิต

เลขขนาด 32 บิตเวิร์ด (bit word) ในส่วนหัวของทีซีพี ซึ่งใช้เป็นตัวชี้ถึงจุดเริ่มต้นของข้อมูล ซึ่งส่วนหัวของทีซีพีจะเป็นจำนวนเท่าของ 32 บิต

สงวนไว้ ขนาด 6 บิต

ฟิลด์นี้ถูกสงวนไว้ใช้ในอนาคต โดยปกติจะเป็นค่าศูนย์

คอนโทรลบิต (Code Bits) ขนาด 6 บิต (จากซ้ายไปขวา)

ยูอาร์จี (URG)	บิตบ่งบอกฟิลด์เออเจ้นท์พอยต์เตอร์ (Urgent Pointer field)
เอซีเค (ACK)	บิตบ่งบอกฟิลด์ตอบรับ (Acknowledgement field)
พีเอสเฮส (PSH)	บิตพุชฟังก์ชั่น (Push Function)
อาร์เอสที (RST)	บิตยุติการเชื่อมต่อ (Reset the connection)
เอสวายเอ็น (SYN)	บิตบอกการซิงโครไนซ์หมายเลขลำดับ
เอฟไอเอ็น (FIN)	บิตบอกว่าไม่มีข้อมูลมาจากผู้ส่ง

วินโดว์ (Window) ขนาด 16 บิต

เลขข้อมูล 16 บิต ใช้ควบคุมการไหลเวียนข้อมูล แสดงค่าที่สามารถตอบรับว่าจะรับส่งข้อมูลได้เท่าไร สามารถใช้งานได้ 2 ทิศทางพร้อมกัน เนื่องจากทีซีพีสามารถทำงานแบบ full-duplex

ผลรวมตรวจสอบ ขนาด 16 บิต

เลขข้อมูล 16 บิต มาจากการรวมข้อมูลขนาด 16 บิตเวิร์ดในส่วนหัวและส่วนข้อมูล แล้วทำวันคอมพลีเม้นท์สองครั้ง ถ้าข้อมูลมีขนาดเป็นเลขคี่ให้เพิ่มอีกอ็อกเต็ทต่อข้างหลังโดยมีค่าเป็นศูนย์ก่อนที่จะทำการคำนวนค่าตรวจสอบนี้จะต้องกำหนดให้เป็นศูนย์เสียก่อน ค่าตรวจสอบนี้จะใช้

ตรวจสอบ ที่อยู่ผู้ส่ง ที่อยู่ผู้รับ โพรโตคอล ความยาวของทีซีพีเซ็กเมนต์ ซึ่งเป็นความยาวส่วนหัวรวมกับความยาวของข้อมูลในรูปแบบอ็อกเต็ท

เออเจ้นท์พอยน์เตอร์ (Urgent Pointer) ขนาด 16 บิต

ฟิลด์นี้บ่งบอกถึงค่าปัจจุบันของตัวชี้ ในสภาพเป็นบวกจากหมายเลขลำดับข้อมูลในเซกเม้นท์ ซึ่งค่านี้ชี้ถึงลำดับของข้อมูลที่ต่อจากข้อมูลเออร์เจ้นท์ จะมีได้ก็ต่อเมื่อบิตควบคุมยูอาร์จีถูกตั้งค่าไว้เท่านั้น

ออปชั่น ขนาดเปลี่ยนแปลงได้

เปรียบเสมือนช่องว่างส่วนท้ายสุดของส่วนหัวทีซีพี มีขนาดเป็นจำนวนเท่าของ 8 บิต ฟิลด์ออปชั่นนี้จะนำมาคิดผลรวมตรวจสอบด้วย ออปชั่นมีได้ด้วยกัน 2 กรณีดังนี้

1. มีเฉพาะตัวบอกชนิด
2. มีตัวบอกชนิด ความยาวของออปชั่นขนาด 1 อ็อกเต็ทและข้อมูล

ความยาวของออปชั่นจะนับรวมฟิลด์ชนิดของออปชั่น และ ฟิลด์ความยาวของออปชั่น รวมทั้งข้อมูลของออปชั่นด้วย

ตารางที่ 5.1 ชนิดของออปชั่นของทีซีพี

ชนิด	ความยาว	ความหมาย
0	-	สิ้นสุดของรายชื่อ ออปชั่น
1	-	ไม่มีคำสั่ง (No-Operation)
2	4	ขนาดสูงสุดของเซกเม้นท์

แพดดิ้ง (Padding) ขนาดเปลี่ยนแปลงได้

เป็นส่วนที่ใช้เติมค่า 0 เพิ่มลงไปเพื่อให้ส่วนหัวของทีซีพีมีขนาดเป็นจำนวนเท่าของ 32 บิต

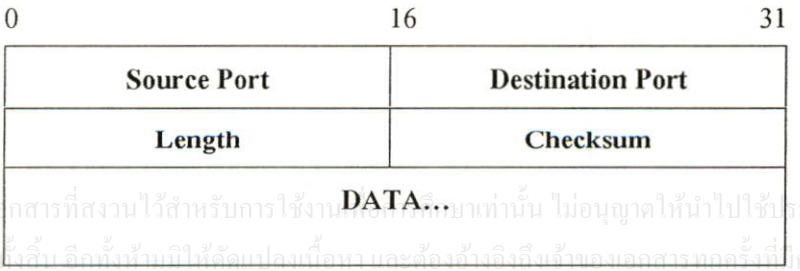
5.2 ยูสเซอร์เดต้าแกรมโพรโตคอล (UDP - User Datagram Protocol)

การทำงานบางอย่างอาจต้องการกำหนดการใช้งานของเดต้าแกรมเอง เช่น การสอบถามเบอร์อินเทอร์เนตแอดเดรสจากชื่ออินเทอร์เนตในระบบ Domain Name Service โดยปกติจะมีฐานข้อมูลสำหรับเก็บชื่อและแอดเดรส ดังนั้นผู้ใช้ก็จะส่งแพ็กเก็ตร้องถาม (query) ไปยังฐานข้อมูล คิวรีเหล่านี้จะสั้นและพอดีกับขนาดเดต้าแกรม ซึ่งไม่จำเป็นต้องใช้ทีซีพี เนื่องจากทีซีพีจะแบ่งแพ็กเก็ตข้อมูลออกเป็นเดต้าแกรม ต้องตอบรับข้อมูลทุกครั้งที่ส่งไป และถูกส่งไปอีกครั้งถ้าจำเป็น การทำงานประเภทนี้สามารถทำได้โดยไม่ต้องใช้ทีซีพีด้วยการเปลี่ยนมาใช้ยูดีพีแทน

ยูดีพีเป็นโพรโตคอลที่ทำงานแบบง่ายๆ ถูกออกแบบสำหรับการใช้งานที่ไม่จำเป็นต้องใส่ลำดับของเดต้าแกรมเข้าไป และถูกส่งไปในไอพีเหมือนกัน แต่ความสามารถของยูดีพีนั้นไม่เท่าทีซีพี คือไม่สามารถแบ่งเดต้าแกรมได้ ไม่สามารถจะตรวจสอบการส่งข้อมูลได้ ไม่มีการส่งข้อมูลซ้ำอีกครั้งเมื่อเกิดข้อผิดพลาด เปรียบได้กับการส่งจดหมาย คือส่งไปโดยที่ไม่ต้องรู้ว่าจะถึงหรือไม่ มีผู้รับหรือไม่ เหมาะกับการสื่อสารช่วงสั้นๆ ที่มีข้อมูลส่งไม่มาก เช่น การค้นหาข้อมูลในฐานข้อมูลของ Name server การส่งและรับข่าวสารสามารถหลีกเลี่ยงโอเวอร์เฮดที่ใช้ในการจัดตั้งและยกเลิกการเชื่อมต่อสื่อสารได้โดยการส่งไปเฉพาะคำร้องขอ (query) และรอผลที่ได้รับ (response) จึงเหมาะสมอย่างยิ่งในการทำระบบตรวจสอบ (monitoring) การแก้ไข (debugging) การจัดการระบบ (management) และการทดสอบโปรแกรม (testing)

เนื่องจากยูดีพีเป็นโพรโตคอลแบบง่ายๆที่ใช้ไอพีในการติดต่อสื่อสาร ดังนั้น ถ้าโปรแกรมที่ใช้งานอยู่ทำการส่งคำร้องขอโดยใช้ยูดีพีและไม่ได้รับการตอบสนองกลับมาในช่วงเวลาที่พอเหมาะ ก็จะต้องเป็นหน้าที่ของโปรแกรมนั้นเองที่จะต้องทำการจัดส่งคำร้องขอนั้นใหม่อีกครั้ง

รูปที่ 5.2 โครงสร้างของยูดีพี



พอร์ตต้นทาง (Source Port) ขนาด 16 บิต

เป็นหมายเลขพอร์ตของผู้ส่ง ถ้าไม่ใช้จะใส่ค่า "0" ไว้

พอร์ตปลายทาง (Destination Port) ขนาด 16 บิต

เป็นหมายเลขพอร์ตผู้รับ

ความยาวแพ็กเก็ต (Packet Length) ขนาด 16 บิต

เป็นความยาวของยูดีพีเดต้าแกรม ซึ่งนับรวมความยาวส่วนหัวด้วย ต้องมีค่าอย่างน้อย 8 อ็อกเต็ท

ผลรวมตรวจสอบ (Checksum) ขนาด 16 บิต

จะบวกทีละ 16 บิต ถ้าข้อมูลหารด้วย 16 ไม่ลงตัวก็จะเพิ่ม 0 ลงไปและรวมกันแบบ 1's complement ในกรณีที่ไม่ใช้ผลรวมตรวจสอบ ฟิลด์นี้จะถูกกำหนดค่าให้เป็น "0" ทุกบิตไว้ แต่ถ้าใช้และคำนวณได้ค่า "0" ก็จะแทนด้วยค่า "1" ทุกบิต

พอร์ต (Port)

โฮสต์บนระบบทีซีพี/ไอพีจะสามารถให้บริการการเชื่อมต่อได้พร้อมๆกันในเวลาเดียวกัน ยูดีพีสามารถแยกแยะการรับส่งข้อมูลดังกล่าวให้กับผู้รับได้อย่างเหมาะสมได้ ทุกๆการเชื่อมต่อที่ใช้ยูดีพีจะต้องมีการกำหนดตัวบ่งบอกขนาด 16 บิตที่เรียกกันว่าหมายเลขพอร์ต (port number) ซึ่งจะมีได้ถึง 65536 พอร์ต ค่าหมายเลขพอร์ตตั้งแต่ 0 ถึง 1023 นั้นถูกสงวนไว้สำหรับบริการที่เป็นมาตรฐาน เรียกพอร์ตมาตรฐานดังกล่าวนี้ว่า well-known port ซึ่งเบอร์พอร์ตเหล่านี้จะถูกกำหนดโดย IANA - Internet Assigned Numbers Authority เช่น บริการการแปลงจากชื่อไปเป็นแอดเดรสโดยใช้ยูดีพีจะเข้าถึงระบบทาง well-known port หมายเลข 53 ถ้าข้อมูลที่รับมาไม่มีหมายเลขพอร์ตที่จะรับจะมีการส่ง ICMP port unreachable กลับออกไปและไม่สนใจข้อมูลนั้น

ทีซีพีและยูดีพีจะใช้หมายเลขพอร์ตเช่นเดียวกันแต่จะเป็นพอร์ตคนละตัวกัน เช่น ผู้ใช้คนหนึ่งอาจจะส่งข้อมูลโดยใช้ยูดีพีพอร์ตเบอร์ 2000 ในขณะที่ผู้ใช้อีกคนหนึ่งกำลังใช้ทีซีพีพอร์ตเบอร์ 2000 เช่นเดียวกันก็ได้

ซ็อกเก็ต (Socket)

ในการสื่อสารข้อมูลจริงๆนั้นจะใช้อินเตอร์เน็ตแอดเดรสกับเบอร์พอร์ทที่ใช้มาประกอบกันเป็นโครงสร้างข้อมูลที่เรียกว่า ซ็อกเก็ต ส่วนหัวของไอพีสำหรับเดต้าแกรมจะประกอบไปด้วยอินเตอร์เน็ตแอดเดรสต้นทางและปลายทาง และส่วนหัวของยูดีพีจะประกอบไปด้วยหมายเลขพอร์ทของต้นทางและปลายทางที่ใช้ในการติดต่อ

บทที่ 6

การออกแบบระบบตรวจสอบและเฝ้าดู

ระบบตรวจสอบและเฝ้าดูที่จัดทำขึ้นนี้ใช้ดักจับแพ็กเก็ตที่ใช้ทีซีพี/ไอพีบนระบบเครือข่ายแบบอีเทอร์เน็ต ทำงานได้โดยไม่เกี่ยวข้องกับซอฟต์แวร์ที่ทำหน้าที่เน็ตเวิร์คเซอร์วิสของเครือข่ายเอง ทำให้สามารถดักจับได้เฉพาะแชร์เสตท (shared states) ที่ถูกอ้างจากเน็ตเวิร์คโพรโตคอลเท่านั้น สามารถดัดแปลงได้ง่าย ไม่ต้องติดต่อกับระบบเครือข่ายโดยตรง ข้อดีของระบบแบบนี้คือไม่ไปแย่งทรัพยากรกับเน็ตเวิร์คเซอร์วิสและโปรแกรมประยุกต์ในการเชื่อมโยงเครือข่ายเหมือนกับระบบเฝ้ามองแบบอื่น ซึ่งอาจจะทำให้ระบบเครือข่ายมีประสิทธิภาพลดลง นอกจากนี้ระบบเฝ้าดูแบบภายนอกจะมีราคาถูกกว่า การเป็นอิสระกับซอฟต์แวร์ที่เป็นเน็ตเวิร์คเซอร์วิสทำให้สามารถปรับปรุงและขยายการใช้งานได้ง่าย แต่ระบบเฝ้ามองแบบภายนอกนี้ก็มีข้อเสียคือข้อมูลที่ได้ไม่สามารถรับประกันได้ว่าแสดงสถานะจริงๆ ของส่วนประกอบที่มันกำลังเฝ้าดูอยู่ จะเป็นเพียงค่าประมาณเท่านั้น ดังนั้นหากต้องการนำมาใช้ในระบบเครือข่ายที่มีโหนดติดต่อกันอยู่มากๆแล้ว จะต้องทำระบบเฝ้ามองแบบนี้ให้มีการทำงานที่เร็วและมีประสิทธิภาพสูงมากตามไปด้วย

6.1 แพ็กเก็ตไดร์เวอร์ (Packet Driver)

เป็นโปรแกรมที่ใช้ในการเชื่อมโยงติดต่อกับการ์ดเชื่อมต่อระบบเครือข่าย ซึ่งจะมีฟังก์ชั่นการทำงานภายในตัวหลายประเภท ผู้เขียนโปรแกรมสามารถนำเอาฟังก์ชั่นเหล่านี้มาใช้ในการเขียนโปรแกรมเข้าถึงตัวการ์ดเชื่อมต่อระบบเครือข่าย ทำให้สามารถนำข้อมูลที่การ์ดเชื่อมโยงเครือข่ายมาประมวลผลต่อไปได้ แพ็กเก็ตไดร์เวอร์จะทำให้โปรแกรมที่เขียนขึ้นใช้งานการ์ดเชื่อมโยงเครือข่ายที่ดาต้าลิงค์เลเวลร่วมกันได้ และทำการแยกแพ็กเก็ตที่เข้ามาไปยังโปรแกรมโดยใช้แบบมาตรฐานของแพ็กเก็ตของตัวกลางในระบบเครือข่าย หรือผ่านเซอร์วิสแอคเซสพอยท์ฟิลด์

แพ็กเก็ตไดร์เวอร์จะมีฟังก์ชั่นให้เรียกใช้งานหลายประเภท เช่น การกำหนดค่าเริ่มต้นในการติดต่อกับรูปแบบแพ็กเก็ตที่ต้องการ การยกเลิกการติดต่อ การส่งแพ็กเก็ต การรับข้อมูลสถิติของการ์ดเชื่อมต่อระบบเครือข่าย และให้ข้อมูลเกี่ยวกับการเชื่อมต่อเครือข่าย

ข้อกำหนดของแพ็กเก็ตไดร์เวอร์จะทำให้การจัดการโพรโตคอลสแต็กเป็นอิสระต่อกันจากการ์ดจากผู้ผลิตต่างๆหรือรูปแบบของการเชื่อมต่อเครือข่าย ซึ่งที่ผ่านมาต้องใช้โพรโตคอลสแต็กแต่ละอย่างบนเครือข่ายที่แตกต่างกัน เช่น อีเทอร์เน็ต, วงแหวนแบบ 802.5 ทั้งนี้เนื่องจากความแตกต่างกันหลายประการ เช่น การแปลงโพรโตคอลไปยังฟิสิคัลแอดเดรส, รูปแบบส่วนหัว และ อื่นๆ แพ็กเก็ตไดร์เวอร์ทำให้สามารถจัดการโพรโตคอลร่วมกันได้อย่างสมบูรณ์บนเครื่องคอมพิวเตอร์ธรรมดา ผู้ใช้สามารถใช้ ทีซีพี/ไอพี เอกซ์เอ็นเอส (XNS) และ โพรโตคอลที่ใช้เฉพาะ เช่น เดคเน็ต (DECNET) และไอพีเอ็กซ์/เอสพีเอส (IPX/SPX) ได้โดยไม่ต้องไปเชื่อมต่อโดยตรงกับการ์ดอินเทอร์เฟส เพียงแต่เรียกไปยังแพ็กเก็ตไดรเวอร์เท่านั้น โปรแกรมที่ทำงานโดยใช้แพ็กเก็ตไดรเวอร์สามารถนำไปใช้ในระบบเครือข่ายอื่นที่เป็นแบบเดียวกันได้โดยไม่ต้องแก้ไขโปรแกรม เพียงแค่หาแพ็กเก็ตไดร์เวอร์อันใหม่เท่านั้น

ข้อกำหนดและการโปรแกรมแพ็กเก็ตไดรเวอร์

แพ็กเก็ตไดรเวอร์ สามารถแบ่งระดับชั้นได้ดังนี้

- ฟังก์ชั่นพื้นฐาน (Basic Function) มีฟังก์ชั่นการทำงานพื้นฐานซึ่งง่ายในการใช้งาน เนื่องจากใช้ทรัพยากรน้อย
- ฟังก์ชั่นเพิ่มเติม (Extended Function) มีฟังก์ชั่นมากกว่าแบบพื้นฐานสนับสนุนมัลติคาสท์ และเก็บสถิติในกาทำงาน
- ฟังก์ชั่นประสิทธิภาพสูง (High-performance Function) สนับสนุนการปรับปรุงประสิทธิภาพ

การทำงานของแพ็กเก็ตไดรเวอร์

แพ็กเก็ตไดรเวอร์เป็นโปรแกรมแบบฝังตัว (TSR) ใช้อินเทอร์รัปท์ในช่วง 0x60 ถึง 0x80 ในการกำหนดอินเทอร์รัปท์จะต้องสามารถเปลี่ยนแปลงได้เพื่อไม่ไปรบกวนกับการทำงานโปรแกรมอื่นๆ กลไกการจัดการ (Handler) สำหรับอินเทอร์รัพท์ กำหนดโดยใช้ค่าเริ่มต้นด้วยรหัส

คำสั่ง (executable code) 3 ไบท์ อาจจะเป็นคำสั่งกระโดด 3 ไบต์ หรือ 2 ไบต์ แล้วตามด้วยคำสั่ง NOP (No Operation) แล้วตามด้วยตัวอักษรแบบไม่มีตัวจบที่มีข้อความว่า "PKT DRVR" ในการหาอินเทอร์รัพท์ที่ถูกใช้โดยแพ็กเก็ตไดรเวอร์โปรแกรมจะต้องค้นหาตั้งแต่อินเทอร์รัพท์หมายเลข 0x60 ถึง 0x80 จนกระทั่งพบข้อความ "PKT DRVR" ใน 12 ไบต์ที่ตามหลังจุดเริ่มต้น วิธีการหาแอดเดรส ดูได้จากฟังก์ชั่น access_type()

การเรียกใช้งานฟังก์ชั่นการทำงานของแพ็กเก็ตไดรเวอร์ทุกฟังก์ชั่นจะถูกเรียกใช้งานโดยการผ่านค่าไปให้ AH รีจิสเตอร์

การระบุการ์ดเชื่อมต่อระบบเครือข่าย

การระบุการ์ดเชื่อมต่อระบบเครือข่าย กำหนดไว้ด้วยตัวเลขสามส่วน ได้แก่

คลาส (Class) ใช้บอกชนิดของสื่อที่การ์ดเชื่อมต่อระบบเครือข่ายนี้สนับสนุน เช่น ดีไอเอ็กซ์อีเทอร์เนต (DIX - DEC /Intel /Xerox) , IEEE 802.3, IEEE 802.5, ProNET-10 แอ็บเปิ้ลทอล์ค (Appletalk) , สายส่งข้อมูลแบบอนุกรม หรือสื่อแบบอื่นๆ

ชนิด (Type) ใช้กำหนดรายละเอียดของการ์ดเชื่อมต่อระบบเครือข่ายที่สนับสนุนในคลาสนั้น ๆ เช่น ในอีเทอร์เนตคลาสนั้นประกอบด้วย ชนิด 3COM, 3C503, 3C505, Interlan NI5210 Univation, BICC Data Networks ISOLAN, Ungermann-Bass NIC

หมายเลข (Number) ถ้าในเครื่องประกอบด้วยการ์ดเชื่อมต่อระบบเครือข่ายมากกว่าหนึ่งคลาสหรือหนึ่งชนิด จะต้องใช้หมายเลขในการระบุถึงความแตกต่างของการ์ดเหล่านั้น

คลาสเป็นตัวเลข 8 บิต และชนิดเป็นตัวเลข 16 บิต ซึ่งกำหนดโดยเอฟทีพีซอฟท์แวร์ (FTP Software) และชนิด 0xFFFF แทนที่ทุกชนิดซึ่งจะตรงกับทุกการเชื่อมต่อในคลาส ในหมายเลขจะไม่มีการใช้ไวลด์คาร์ด (Wildcard) ซึ่ง 0 จะหมายถึงการ์ดเชื่อมต่อระบบเครือข่ายอันแรกของคลาสและชนิด ในข้อกำหนดไม่ได้สนับสนุนมัลติเปิลการ์ดอินเทอร์เฟส (multiple interfaces card) บนแพ็กเก็ตไดรเวอร์เดียวกัน ซึ่งหมายความว่าต้องเรียกแพ็กเก็ตไดรเวอร์หนึ่งตัวต่อการ์ดเชื่อมต่อระบบเครือข่ายหนึ่งแผ่น และต้องกำหนดอินเทอร์รัพท์แต่ละเบอร์ให้แต่ละแพ็กเก็ตไดรเวอร์ ในกรณีที่ต้องใช้การ์ดเชื่อมต่อระบบเครือข่ายหลายๆแผ่น ซึ่งโปรแกรมจะต้องตรวจสอบคลาสและชนิด ซึ่งได้จากการเรียก driver_info() เพื่อให้มั่นใจว่าใช้แพ็กเก็ตไดรเวอร์ตรงกับสื่อและรูปแบบแพ็กเก็ต

6.2 ฟังก์ชั่นใช้งานของแพ็กเก็ตไดร์เวอร์

จะใช้แฮนด์เดิล (Handle) ซึ่งเป็นค่าตัวเลขที่เกี่ยวเนื่องกับแม็คลีเวลดีมัลติเพล็กซิ่ง (MAC-level demulitplexing) แต่ละชนิด ซึ่งได้มาจากการเรียกประเภทการใช้งาน (access type) ภายในแพ็กเก็ตไดรเวอร์ อาจจะเป็นพอย์นเตอร์ (Pointer) หรือ เทเบิ้ลออฟเซต (Table offset) ก็ได้

บางฟังก์ชั่นอาจจะเรียกใช้ได้ก็ต่อเมื่อการ์ดนั้นสามารถทำงานในโหมดการทำงานฟังก์ชั่นเพิ่มเติม หรือฟังก์ชั่นประสิทธิภาพสูงได้ เนื่องจากไม่จำเป็นต่อการใช้งานพื้นฐานของเน็ตเวิร์คดังนั้นจึงกำหนดเป็นส่วนเพิ่มเติม ซึ่งจะตรวจสอบว่าสามารถใช้งานได้หรือไม่ต้องตรวจสอบที่ฟังก์ชั่นชื่อ driver_info()

การเรียงตัวของบิตข้อมูลที่ส่งในระบบเครือข่ายอีเทอร์เน็ตจะตรงกันข้ามกับการจัดเก็บข้อมูลในระบบคอมพิวเตอร์ทั่วไป นั่นคือ ค่าประเภทของอีเทอร์ (Ethertype values) ซึ่งส่งไปยัง access_type() จะต้องมีการสลับค่าไบท์สูงและไบท์ต่ำกันถ้าข้อมูลนั้นเป็นแบบเวิร์ด

driver_info()

function driver_info (handle)

Input AH 1

AL 255

BX handle /*Optional*/

error return:

carry flag set

DH error code

possible errors

BAD_HANDLE

non-error return

carry flag clear

BX version

CH class

DX type

CL number

DS:SI name

AL functionality

1 == basic functions present.

2 == basic and extended present.

5 == basic and high-performance.

6 == basic, high-performance, extended.

255 == not installed.

ฟังก์ชั่นนี้ใช้เพื่อหาข้อมูลเกี่ยวกับเวอร์ชั่นของโปรแกรมแพ็กเก็ตไดร์เวอร์ที่ใช้เชื่อมต่อ ในเวอร์ชั่นก่อนหน้านี้ ค่าแฮนด์เดิลที่ส่งเข้าไปนั้นต้องมีเสมอ แต่ในเวอร์ชั่นปัจจุบันเป็นตัวเลือกซึ่งจะมีหรือไม่มีก็ได้

access_type()

function int access_type (if_class, if_type, if_number, type, typelen, receiver)

input: AH 2

AL if_class

BX if_type

DL if_number

DS:SI type

CX typelen

ES:DI receiver

error return:

carry flag set

DH error code

possible errors:

NO_CLASS

NO_TYPE

NO_NUMBER

BAD_TYPE

NO_SPACE

```
                    TYPE_INUSE
non-error return:
            carry flag clear
            AX      handle
receiver call   ( *receiver ) ( handle, flag, len [[, lah_len], buffer] )
input:      BX      handle
            AX      flag
            If AX == 0,
                    DX      lah_len
                    DS:SI   lah_buffer      /* if DX != 0    */
            Returns:
                    ES:DI   buffer          /* if 0, no buffer */
                    CX      buffer_len      /* May not equal entry CX      */
            If AX == 1,
                    DS:SI   buffer
                    CX      buffer_len      /* Bytes actually copied  */
            Error return:
                    ES:DI   0:0
            Non-error return:
                    ES:DI   buffer pointer
                    CX      buffer length
```

ฟังก์ชั่นนี้เป็นตัวเริ่มต้นในการทำงานของแพ็กเก็ตไดร์เวอร์ ซึ่งมีค่าต่างๆดังนี้

type ค่าที่ส่งผ่านไปฟังก์ชั่นนั้นต้องกำหนดโดยข้อกำหนดชนิดแพ็กเก็ต

typelen ขนาดเป็นไบต์ของฟิลด์ type ถ้าเป็น 0 หมายความว่า ตัวเรียกต้องการทุกแพ็กเก็ต

receiver คือ พอยน์เตอร์ที่ชี้ไปยังส่วนของโปรแกรมซึ่งถูกเรียกทุกครั้งเมื่อแพ็กเก็ตได้รับ เมื่อแพ็กเก็ตได้รับก็จะเรียก receiver สองครั้งด้วยกัน ครั้งแรกเรียกเพื่อขอหน่วยความจำบัฟเฟอร์จากโปรแกรมเพื่อคัดลอกแพ็กเก็ตลงไป ซึ่งค่า AX จะมีค่าเท่ากับ 0 โปรแกรมจะต้องส่งค่าพอยน์เตอร์ไปยังบัฟเฟอร์นั้นโดยใช้ ES:DI ถ้าโปรแกรมไม่มีบัฟเฟอร์จะส่งค่า 0:0 ใน ES:DI และแพ็กเก็ตไดร์เวอร์จะทิ้งแพ็กเก็ตนั้นเสีย และจะไม่มีการเรียกครั้งที่สองอีก

ความยาวของแพ็กเก็ตเป็นส่วนสำคัญมากโดยจะเก็บค่านี้ไว้ใน CX ค่านี้จะใช้ได้เมื่อค่า AX มีค่าเท่ากับ 0 ซึ่ง receiver สามารถกำหนดบัฟเฟอร์ให้เพียงพอกับขนาดได้ ค่าความยาวนี้รวมถึงส่วนหัวแม็ค (MAC header) และข้อมูลที่รับได้ แต่ไม่รวมส่วนเอฟซีเอส (FCS - Frame Check Sequence)

ในการเรียกครั้งที่สองนั้น ค่า AX มีค่าเท่ากับ 1 หมายถึงการคัดลอกแพ็กเก็ตนั้นเสร็จสิ้นแล้ว และโปรแกรมสามารถนำข้อมูลในบัฟเฟอร์ไปใช้งานได้ ซึ่งจะใช้ DS:SI เป็นตัวชี้บัฟเฟอร์ที่ใช้งานได้

release_type()

function int release_type (handle)

input: AH 3

BX handle

error return:

carry flag set

DH error code

possible errors:

BAD_HANDLE

non-error return:

carry flag clear

ฟังก์ชั่นนี้ใช้เพื่อจบการใช้งานโดยใช้แฮนด์เดิลที่ได้มาจาก access_type()

send_pkt()

function int send_pkt (buffer, length)

input: AH 4

DS:SI buffer

CX length

error return:

carry flag set

DH error code

possible errors:

CANT_SEND

non-error return:

carry flag clear

ฟังก์ชั่นนี้ใช้ส่งข้อมูลขนาดความยาว length ไบต์ในบัฟเฟอร์ ซึ่งโปรแกรมจะต้องจัดเตรียมตัวข้อมูลทั้งหมดในแพ็กเก็ตรวมทั้งส่วนหัวของเครือข่ายที่ใช้งานนั้น (local network headers)

terminate()

function terminate (handle)

input: AH 5

BX handle

error return:

carry flag set

DH error code

possible errors:

BAD_HANDLE

CANT_TERMINATE

non-error return:

carry flag clear

ฟังก์ชั่นจบการทำงานของไดรเวอร์โดยใช้แฮนเดิ้ล ถ้าเป็นไปได้ ไดรเวอร์จะคืนค่าหน่วยความจำให้กับระบบจัดการเลย

get_address()

function get_address (handle, buf, len)

input AH 6

BX handle /* Optional */

ES:DI buf

CX len

error return:

carry flag set

DH error code

possible errors:

BAD_HANDLE

NO_SPACE

non-error:

carry flag clear

CX length

ฟังก์ชั่นนี้ใช้เพื่อหาค่าฮาร์ดแวร์แอดเดรสของการ์ดเชื่อมต่อระบบเครือข่ายใส่ลงไปใน buf ที่มีความยาว len ไบต์ ซึ่งค่าตัวเลขจริงๆจะเก็บไว้ใน CX ถ้าเกิด NO_SPACE error หมายความว่า len นั้นมีขนาดไม่เพียงพอกับความยาวของฮาร์ดแวร์แอดเดรส ถ้าใช้ฟังก์ชั่น set_address() เปลี่ยนแอดเดรสใหม่ จะมีการส่งแอดเดรสใหม่คืนให้

reset_interface()

function reset_interface (handle)

input AH 7

BX handle /* Optional */

error return:

carry flag set

DH error code

possible errors:

BAD_HANDLE

CANT_RESET

non-error return:

carry flag clear

ฟังก์ชั่นนี้จะทำการรีเซ็ตการ์ดเชื่อมต่อระบบเครือข่ายใหม่อีกครั้งโดยใช้ค่าแฮนด์เดิล เพื่อต้องการทราบสถานะ (state) จะยกเลิกการส่งข้อมูลที่กำลังทำงานอยู่ และ กำหนดค่าของ reciever

ใหม่อีกครั้ง ค่าฮาร์ดแวร์แอดเดรสจะเปลี่ยนเป็นค่าเดิมที่ได้จาก ROM ของตัวการ์ดเอง ค่ามัลติคาสท์ลิสต์ (multicast list) จะถูกลบ และ โหมดการรับจะเปลี่ยนเป็น 3 (รับเฉพาะค่าแอดเดรสตัวเองและบอร์ดคาสท์) ถ้ามีแฮนด์เดิลเปิดอยู่หลายอัน ฟังก์ชั่นนี้อาจจะไปรบกวนการทำงานของโปรแกรมอื่นที่ใช้การ์ดเชื่อมต่อระบบเครือข่ายนี้ร่วมกันได้ ดังนั้นอาจจะมีการส่ง CANT_RESET คืนมาแทน

get_parameters() high-performance driver function

function get_parameters()

  AH 10

error return:

  carry flag set

  DH error code

possible errors:

  BAD_COMMAND

non error return:

  carry flag clear

  ES:DI param

```
struct param {
    unsigned char   major_rev;      /* Revision of  Packet Driver spec      */
    unsigned char   minor_rev;      /*  this driver conforms to.            */
    unsigned char   length;         /* Length of structure in bytes         */
    unsigned char   addr_len;       /* Length of a MAC-layer address        */
    unsigned short  mtu;            /* MTU, including MAC headers           */
    unsigned short  multicast_aval; /* Buffer size for multicast address    */
    unsigned short  rcv_bufs;       /* ( # of back-to-back MTU rcvs ) - 1   */
    unsigned short  xmt_bufs;       /* ( # of successive xmits ) - 1        */
    unsigned short  int_num;        /* Interrupt # to hook for post-EOI processing,
                                       0 == none                            */
};
```

โปรแกรมประยุกต์ที่จัดทำขึ้นอาจจะมีประสิทธิภาพมากขึ้นโดยการเรียกใช้ฟังก์ชั่นนี้เพื่อที่จะได้ไดรเวอร์พารามิเตอร์ (Driver parameter) ตัวอื่นๆมา

major_rev และ minor_rev เป็นค่าหมายเลขหลัก และ หมายเลขรองของรุ่นของโปรแกรมแพ็กเก็ตไดร์เวอร์นี้ ซึ่งในโปรแกรมแพ็กเก็ตไดร์เวอร์ที่ใช้นี้จะมีค่า major_rev เป็น 1 และ minor_rev เป็น 10

length ใช้เพื่อบอกขนาดของ param นี้ มีค่าเป็น 14

addr_len เป็นความยาวของแม็คแอดเดรสในขนาดไบต์

mtu เป็นค่าขนาดใหญ่ที่สุดของแม็คเลเวลแพ็กเก็ตที่ไดรเวอร์สามารถจัดการได้ ในระบบเครือข่ายแบบอีเทอร์เน็ต ค่านี้จะถูกกำหนดตายตัวเป็น 1514 ไบท์ แต่ในระบบเครือข่ายแบบอื่น เช่น 802.5 และ FDDI ค่านี้อาจจะเปลี่ยนแปลงได้

multicast_aval เป็นค่าจำนวนไบต์ที่ต้องใช้ในการเก็บมัลติคาสท์แอดเดรส โดยการเรียกใช้ฟังก์ชั่น set_multicast_list() ถ้าเป็นค่า 0 หมายถึงไม่สนับสนุนมัลติคาสท์

rcv_bufs และ xmt_bufs เป็นตัวเลขบอกค่าการรับหรือการส่งแบบแบ็คทูแบ็ค (back-to-back) ซึ่งโปรแกรมจะใช้ข้อมูลนี้เป็นตัวกำหนดการควบคุมการไหล (flow control) หรือลักษณะการส่ง (transmit strategies) การ์ดเชื่อมต่อระบบเครือข่ายที่มีบัฟเฟอร์ชุดเดียวจะให้ค่าเป็น '0' ในทั้งสองฟิลด์ ผู้สร้างไดร์เวอร์สามารถกำหนดค่า '0' ใน rcv_bufs เพื่อระบุว่า ฮาร์ดแวร์นั้นมีขีดจำกัดในการทำงาน เพื่อป้องกันจากการรับข้อมูลจากระบบอื่นที่อาจส่งมาเร็วกว่าได้ โพรโตคอลในระดับชั้นที่สูงกว่าควรจะทำการกำหนดควบคุมการไหลแบบล็อกทีละขั้น (lock-step flow control) เพื่อมิให้เกิดการสูญเสียของแพ็กเก็ต

int_num จะถูกกำหนดเป็นเบอร์อินเทอร์รัพท์ ที่โปรแกรมสามารถเรียกใช้เพื่อทำการประมวลผลอินเทอร์รัพท์ไทม์โพรโตคอล หลังจากส่ง EOI ไปยัง 8259 อินเทอร์รัพท์คอนโทรลเลอร์และการ์ดเชื่อมต่อระบบเครือข่ายพร้อมสำหรับการขัดจังหวะอื่นๆ

set_rcv_mode() **extended driver function**

function set_rcv_mode (handle, mode)

input AH 20

BX handle

CX mode

error return:

carry flag set

DH error code

possible errors:

BAD_HANDLE

BAD_MODE

non-error return:

carry flag clear

ฟังก์ชั่นนี้ใช้กำหนดการทำงานของโหมดการรับข้อมูลแพ็กเก็ตโดยใช้แฮนด์เดิล ซึ่งมีโหมดต่างๆดังนี้

1. ไม่รับแพ็กเก็ต
2. รับเฉพาะแพ็กเก็ตที่ถูกส่งมายังการ์ดเชื่อมต่อระบบเครือข่ายนี้
3. โหมด 2 รวมกับบอร์ดคาสท์เดสติเนชั่นแอดเดรส
4. โหมด 3 รวมกับมัลติคาสท์แอดเดรสที่ถูกเซ็ตโดย set_multicast_list()
5. โหมด 3 รวมกับมัลติคาสท์แพ็กเก็ตทั้งหมด
6. ทุกแพ็กเก็ต

การ์ดเชื่อมต่อระบบเครือข่ายไม่ทั้งหมดที่สามารถทำงานได้ทุกโหมดการทำงานนี้ โหมดการรับจะมีผลกับทุกแพ็กเก็ตที่รับโดยใช้การ์ดเชื่อมต่อระบบเครือข่ายนี้ ไม่เกี่ยวกับแฮนด์เดิล

ค่าปกติของโหมดการรับจะเป็นโหมด 3 และถ้า ฟังก์ชั่น set_rcv_mode() ไม่ได้กำหนดไว้จะถือว่าเป็นโหมดการรับเป็นโหมด 3 ไปตลอดตราบที่ยังมีการเรียกใช้แฮนด์เดิลอยู่

get_rcv_mode() **extended driver function**

function get_rcv_mode (handle, mode)

input AH 21

BX handle

error return:

carry flag set

DH error code

possible errors:

BAD_HANDLE

non-error return:

carry flag clear

AX mode

ฟังก์ชั่นนี้ใช้สอบถามค่าโหมดการรับปัจจุบัน

get_statistics() extended driver function

function get_statistics (handle)

input: AH 24

BX handle /* Optional */

error return:

carry flag set

DH error code

possible errors:

BAD_HANDLE

non-error return:

carry flag clear

DS:SI char far *stats

```
struct statistics {
        unsigned long   packets_in;     /* Totals across all handles       */
        unsigned long   packets_out;
        unsigned long   bytes_in;       /* Including MAC headers           */
        unsigned long   bytes_out;
        unsigned long   errors_in;      /* Totals across all error types   */
        unsigned long   errors_out;
        unsigned long   packets_lost;   /* No buffer from receiver() , card */
                                        /* out of resources, etc.          */
};
```

ฟังก์ชั่นนี้จะส่งค่าพอยน์เตอร์ที่ชี้ไปยังโครงสร้างข้อมูลแบบ struct ที่เก็บสถิติของการ์ดเชื่อมต่อระบบเครือข่ายนี้ ค่าที่เก็บอยู่ในรูปตัวเลขแบบ integer ขนาด 32 บิตในระบบ 80x86

set_address() **extended driver function**

function set_address (addr, len)

input: AH 25

ES:DI char far *addr

CX len

error return:

carry flag set

DH error code

possible errors:

CANT_SET

BAD_ADDRESS

non-error return:

carry flag clear

CX length

ฟังก์ชั่นนี้เรียกใช้เมื่อโปรแกรมประยุกต์หรือโพรโตคอลสแตกต้องการใช้ฮาร์ดแวร์แอดเดรสหมายเลขที่ระบุ

หมายเลขฟังก์ชั่นเรียกและพารามิเตอร์

แพ็กเก็ตไดรเวอร์จะกำหนดฟังก์ชั่นการทำงานได้โดยกำหนดค่าฟังก์ชั่นที่ต้องการทำงานนั้นลงในรีจิสเตอร์ AH แล้วเรียกบริการอินเทอร์รัพท์ เพื่อเรียกใช้งานฟังก์ชั่นการทำงานต่างๆของแพ็กเก็ตไดร์เวอร์

driver_info 1

access_type 2

release_type 3

send_pkt	4
terminate	5
get_address	6
reset_interface	7
+get_parameters	10
+as_send_pkt	11
+drop_pkt	13
*set_rcv_mode	20
*get_rcv_mode	21
*set_multicast_list	22
*get_multicast_list	23
*get_statistics	24
*set_address	25
*signal	29
*get_structure	30

+ หมายถึง ฟังก์ชั่นประสิทธิภาพสูง

* หมายถึง ฟังก์ชันเพิ่มเติม

ค่า AH ตั้งแต่ 128 ถึง 255 สงวนไว้ใช้สำหรับการพัฒนาอื่นๆ

รหัสข้อผิดพลาด (Error codes)

ถ้ามีข้อผิดพลาดในการเรียกฟังก์ชั่นการทำงานของแพ็กเก็ตไดรเวอร์ จะมีการเซตค่าแฟลกตัวทด (carry flag) และ ค่ารหัสผิดพลาดจะถูกกำหนดในรีจิสเตอร์ DH ซึ่งรีจิสเตอร์นี้จะไม่ถูกใช้ในการส่งค่าผ่านฟังก์ชั่น แต่จะถูกใช้เพื่อส่งรหัสผิดพลาดกลับมา ซึ่งมีการกำหนดไว้ดังนี้

1	BAD_HANDLE	หมายเลขแฮนด์เดิลผิด
2	NO_CLASS	ไม่มีคลาสที่กำหนด
3	NO_TYPE	ไม่มีชนิดที่กำหนด
4	NO_NUMBER	ไม่มีหมายเลขที่กำหนด
5	BAD_TYPE	กำหนดรูปแบบแพ็กเก็ตผิด
6	NO_MULTICAST	การ์ดเชื่อมต่อระบบเครือข่ายนี้ไม่สนับสนุนมัลติคาสท์

7	CANT_TERMINATE	ไม่สามารถสิ้นสุดการทำงานแพ็กเก็ตไดรเวอร์
8	BAD_MODE	กำหนดโหมดการรับแพ็กเก็ตผิด
9	NO_SPACE	การทำงานล้มเหลวเนื่องจากไม่มีที่ว่างพอ
10	TYPE_INUSE	ชนิด ที่กำหนดกำลังใช้งานอยู่ ไม่สามารถยกเลิกได้
11	BAD_COMMAND	คำสั่งนอกเหนือจากที่กำหนด
12	CANT_SEND	ไม่สามารถส่งแพ็กเก็ตได้ ปกติจะมีสาเหตุมาจากฮาร์ดแวร์
13	CANT_SET	ไม่สามารถเปลี่ยนฮาร์ดแวร์แอดเดรสได้เนื่องจากมีแฮนด์เดิล มากกว่า 1 เปิดอยู่
14	BAD_ADDRESS	ฮาร์ดแวร์แอดเดรสมีความยาวผิดหรือผิดรูปแบบ
15	CANT_RESET	ไม่สามารถรีเซ็ตได้เนื่องจากมีแฮนด์เดิลเปิดอยู่มากกว่า 1

6.3 ส่วนอัลกอริทึมของโปรแกรม

โปรแกรมที่จัดทำขึ้นนี้เชื่อมต่อกับระบบเครือข่ายคอมพิวเตอร์ที่เป็นแบบอีเทอร์เน็ต โดยใช้การ์ดเชื่อมต่อที่เข้ากันได้กับการ์ดแบบ NE2000 ใช้แพ็กเก็ตไดร์เวอร์ของบริษัท FTP Software ในการติดต่อกับการ์ดเชื่อมต่อ โปรแกรมที่สร้างใช้ภาษาปาสคาลร่วมกับภาษาแอสเซมบลี้ในการพัฒนาโปรแกรมให้ทำงานภายใต้ระบบจัดการแบบ MS-DOS ก่อนที่จะเริ่มสั่งให้โปรแกรมทำงานจะต้องมีการเรียกใช้แพ็กเก็ตไดร์เวอร์ให้ทำงานแบบฝังตัวเข้ากับหน่วยความจำของระบบก่อน เพื่อโปรแกรมที่จัดทำขึ้นจะสามารถเรียกใช้ฟังก์ชั่นการทำงานต่างๆของแพ็กเก็ตไดร์เวอร์ได้

การจับข้อมูล

การตรวจสอบแพ็กเก็ตจะต้องคอยระวังเรื่องการรับแพ็กเก็ตซึ่งจะมีปัญหาตรงที่ไม่สามารถรับแพ็กเก็ตมาประมวลผลได้ทัน โดยเฉพาะในระบบเครือข่ายที่มีการส่งข้อมูลอย่างหนาแน่น ทำให้ต้องสูญเสียแพ็กเก็ตบางส่วนไป ดังนั้นควรจะมีบัฟเฟอร์ความเร็วสูงเพื่อให้สามารถสำรองแพ็กเก็ตในขณะนั้นได้มากที่สุด การทำงานในส่วนนี้จะเรียกผ่านแพ็กเก็ตไดร์เวอร์ จากนั้นก็จะส่งผ่านข้อมูลไปยังโมดูลต่างๆ เพื่อรวบรวมข้อมูลมาประมวลผลต่อไป

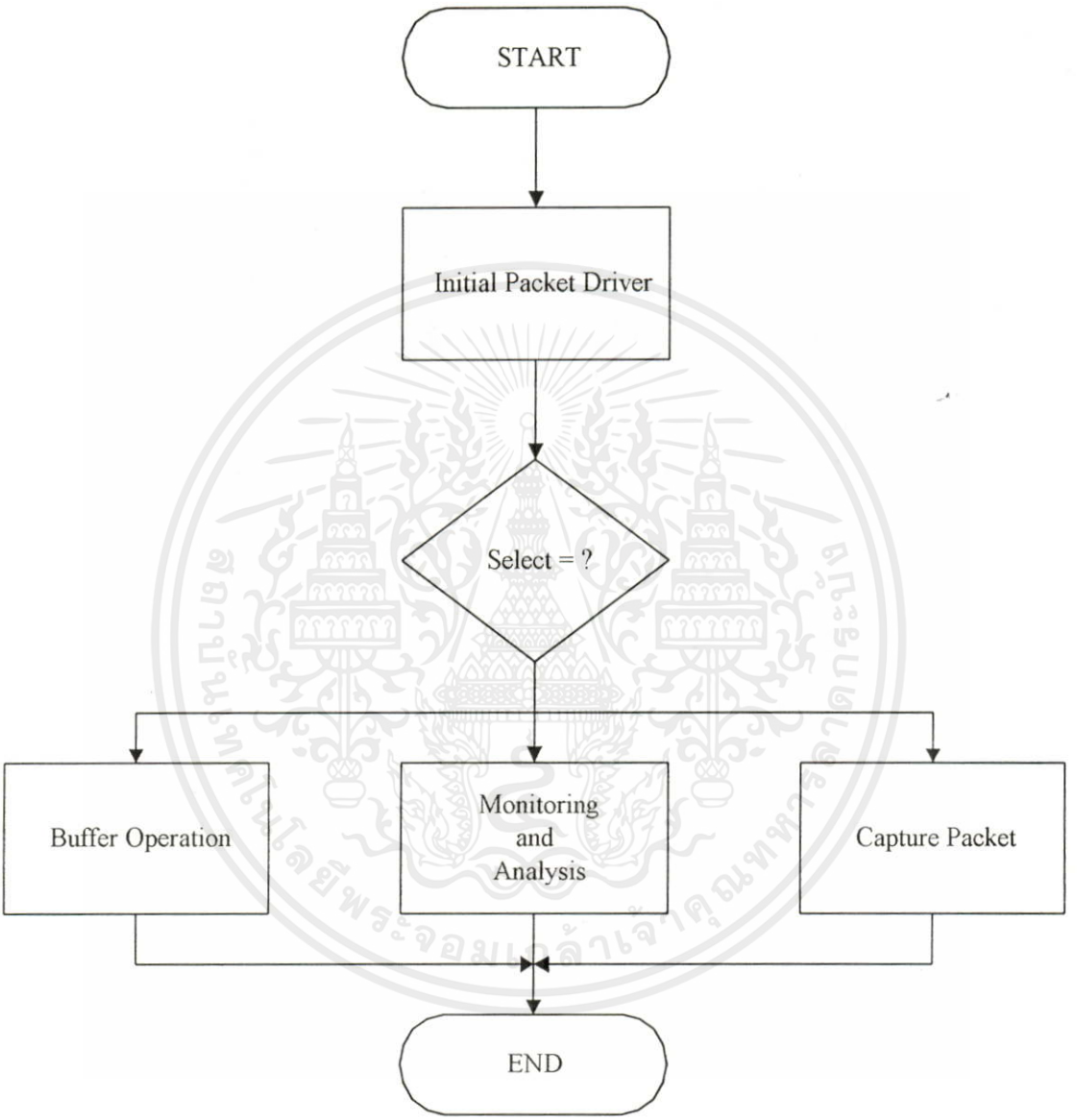
การกรองแพ็กเก็ต

หลังจากสามารถตรวจจับแพ็กเก็ตได้ก็อาจจะทำการกรองแพ็กเก็ตที่ต้องการพิจารณา ถ้าไม่ได้กำหนดการกรองจะถือว่าให้เก็บทุกแพ็กเก็ต ซึ่งอาจทำให้ข้อมูลในบัฟเฟอร์มีขนาดใหญ่และอาจมีข้อมูลที่ไม่สนใจรวมอยู่ด้วย ซึ่งจะทำให้วิเคราะห์แพ็กเก็ตที่สำคัญหรือมีปัญหาได้ลำบาก เนื่องจากมีข้อมูลที่ไม่เกี่ยวข้องมากเกินไป ดังนั้นโปรแกรมจึงควรที่จะสามารถเลือกทำการกรองอีกได้ขั้นหนึ่งเพื่อให้สามารถดูได้เฉพาะแพ็กเก็ตที่ต้องการมาวิเคราะห์ เช่น เลือกเฉพาะแพ็กเก็ตที่ส่งจากสถานีงานต้นทางหรือสถานีงานปลายทางที่กำหนดได้

ส่วนการแสดงผล

ค่าที่ได้จากการใช้แพ็กเก็ตไดร์เวอร์ทำการดักจับแพ็กเก็ตไว้ก็สามารถนำมาวิเคราะห์และแสดงผลข้อมูลต่างๆ เกี่ยวกับเครือข่าย เช่น อัตราการใช้งาน (Utilization) จำนวนแพ็กเก็ตต่อวินาที (Packet per second) จำนวนกิโลไบต์ต่อวินาที (Kilobytes per second) ข้อมูลที่อยู่ภายในตัวแพ็กเก็ต เป็นต้น ซึ่งสามารถนำข้อมูลดังกล่าวมาวิเคราะห์หาค่าพารามิเตอร์อื่นๆที่ต้องการทราบต่อได้

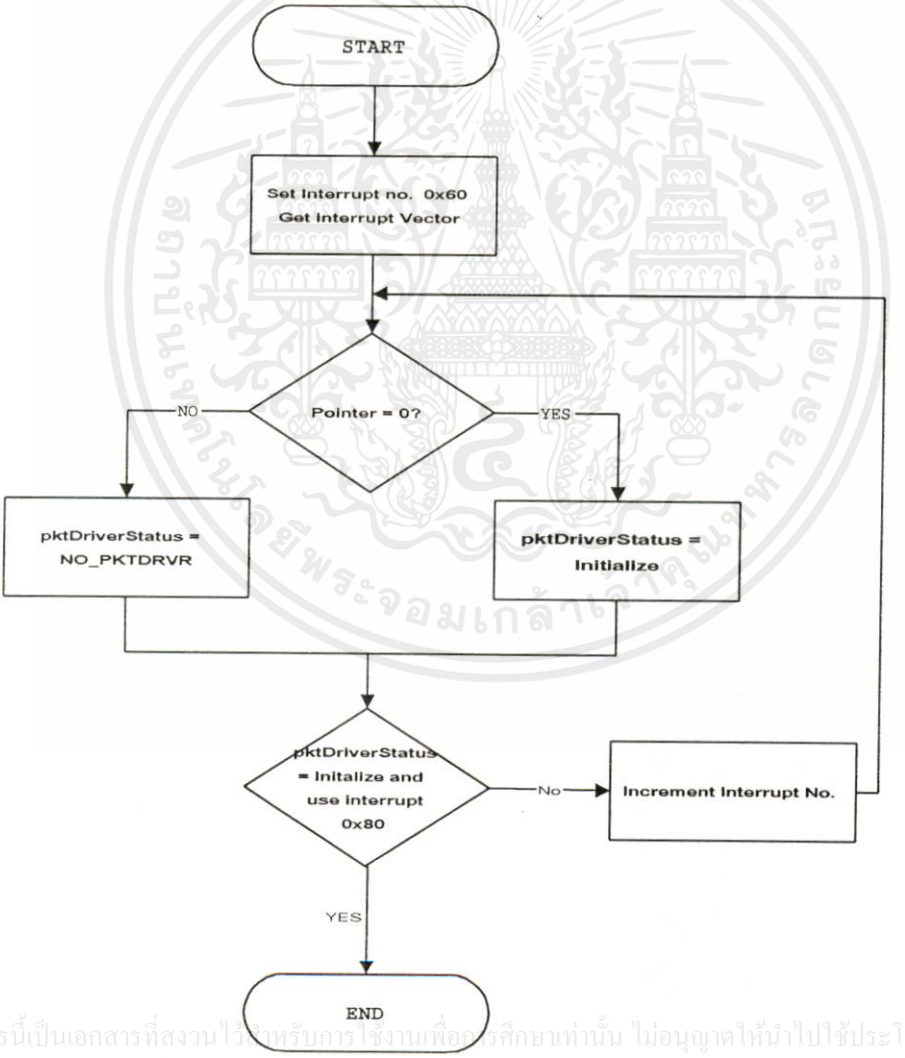
รูปที่ 6.1 ขั้นตอนการทำงานของโปรแกรมหลัก



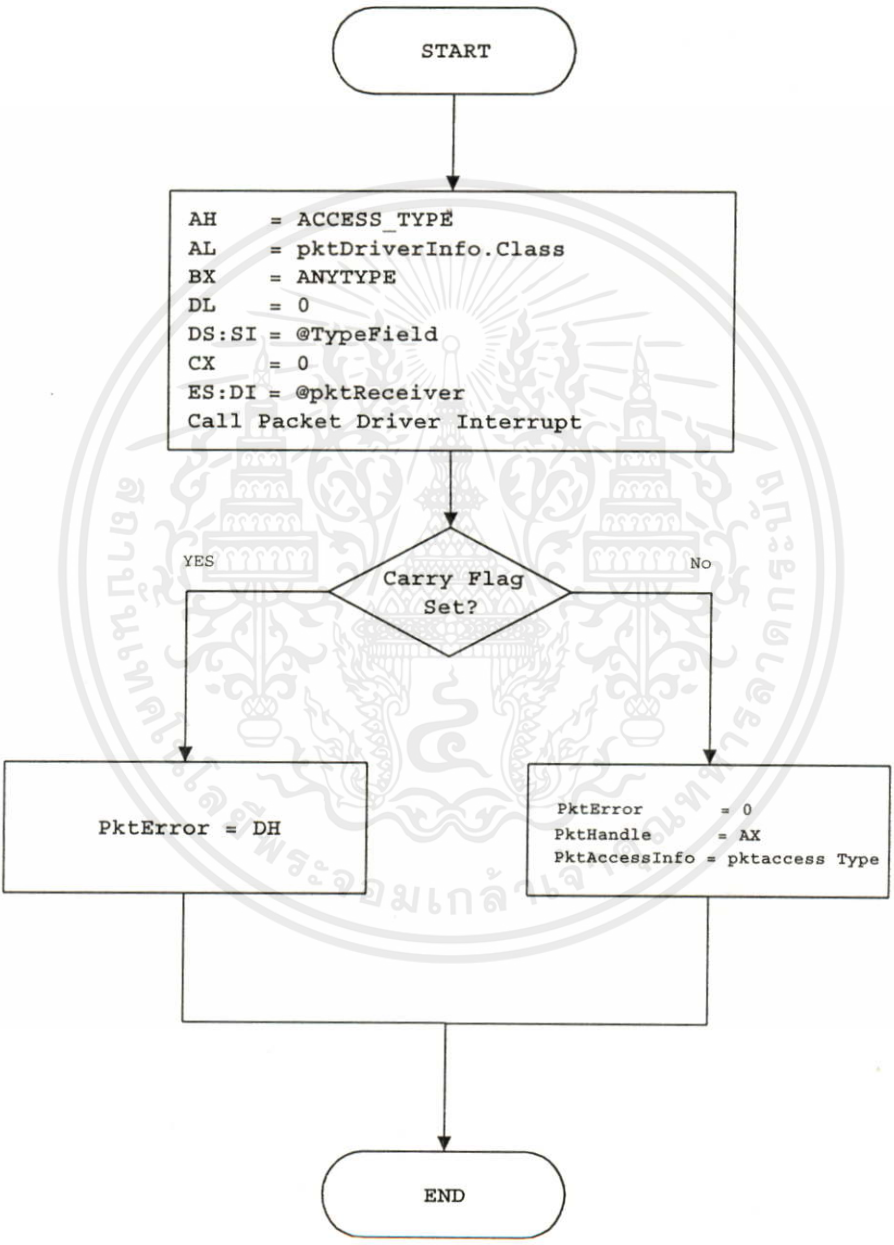
โมดูลกำหนดค่าเริ่มต้น

ทำหน้าที่กำหนดค่าเริ่มต้นในการทำงานในส่วนต่างๆ เช่นส่วนของแพ็กเก็ตไดร์เวอร์ ตัวแปร หน่วยความจำ ในส่วนการกำหนดค่าของแพ็กเก็ตไดรเวอร์จะค้นหาบริการอินเทอร์รัพท์ที่มีให้แล้วจึงตั้งค่าการทำงานของแพ็กเก็ตไดรเวอร์ต่าง ๆ เช่น วิธีการรับแพ็กเก็ตและฟังก์ชั่นที่สนับสนุนในโหมดต่างๆ

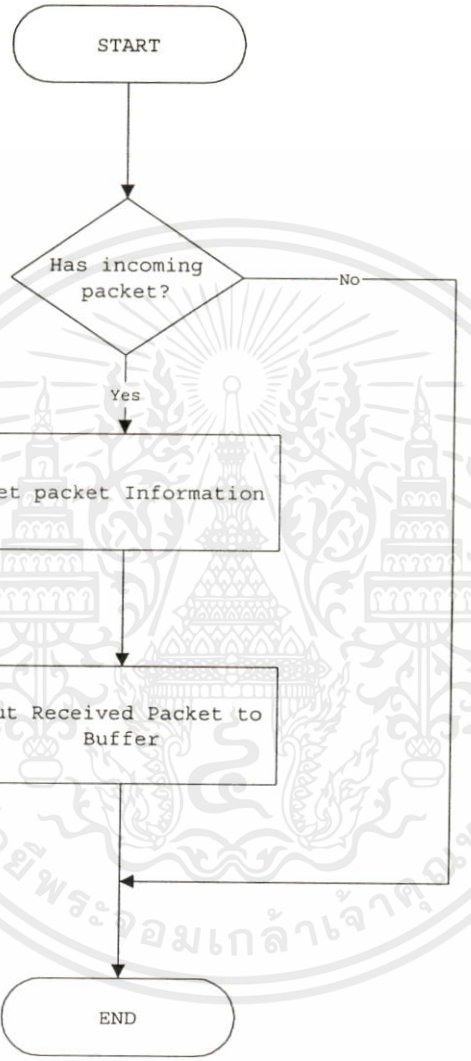
รูปที่ 6.2 ขั้นตอนการหาแพ็กเก็ตไดรเวอร์



รูปที่ 6.3 ขั้นตอนการตั้งค่าวิธีการรับแพ็กเก็ต



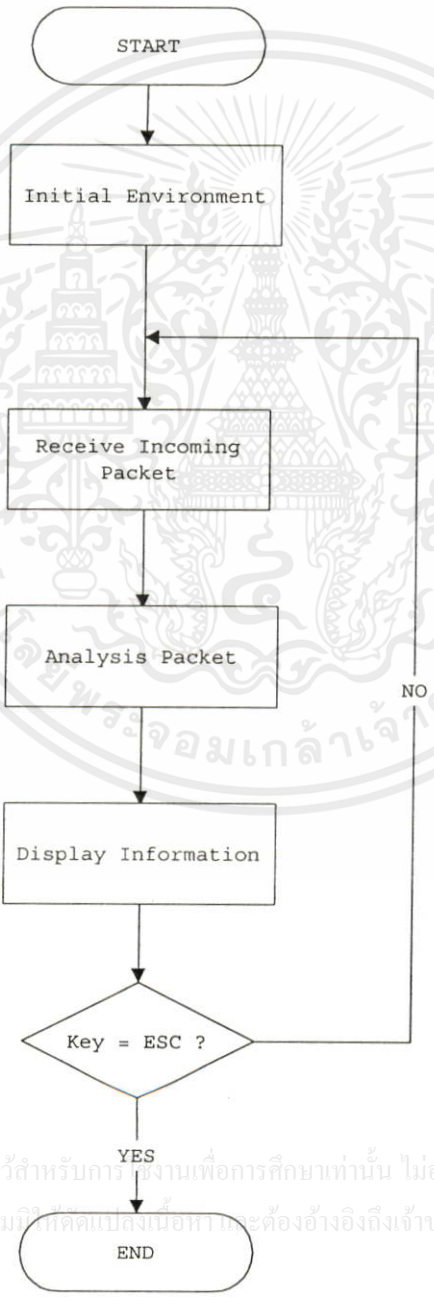
รูปที่ 6.4 ขั้นตอนการรับแพ็กเก็ต



โมดูลวิเคราะห์

ทำหน้าที่รับข้อมูลจากระบบเครือข่ายแล้วนำมาแสดงการใช้งานของระบบเครือข่าย และปริมาณข้อมูลจำนวนแพ็กเก็ตที่เกิดขึ้นในระบบเครือข่าย

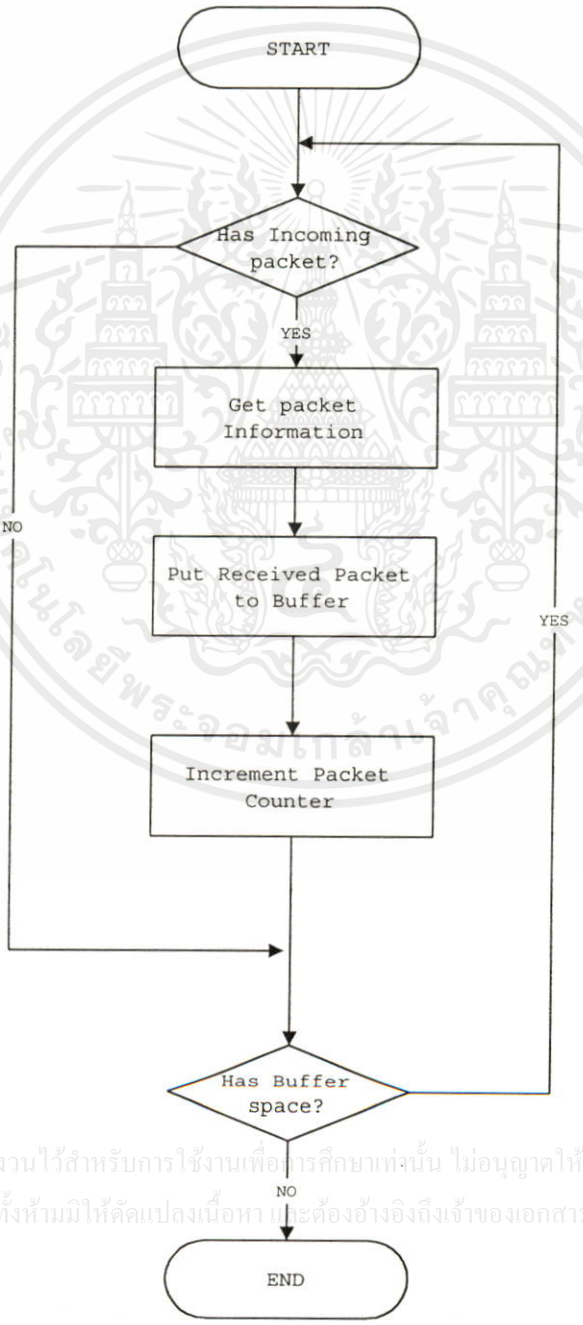
รูปที่ 6.5 ขั้นตอนการทำงานของโมดูลวิเคราะห์



โมดูลจัดเก็บแพ็กเก็ต

เป็นส่วนที่ทำการจัดเก็บข้อมูลของแพ็กเก็ตที่เกิดขึ้นในระบบเครือข่ายลงในหน่วยความจำที่เป็นบัฟเฟอร์ หรือจัดเก็บลงไฟล์ในฮาร์ดดิสก์

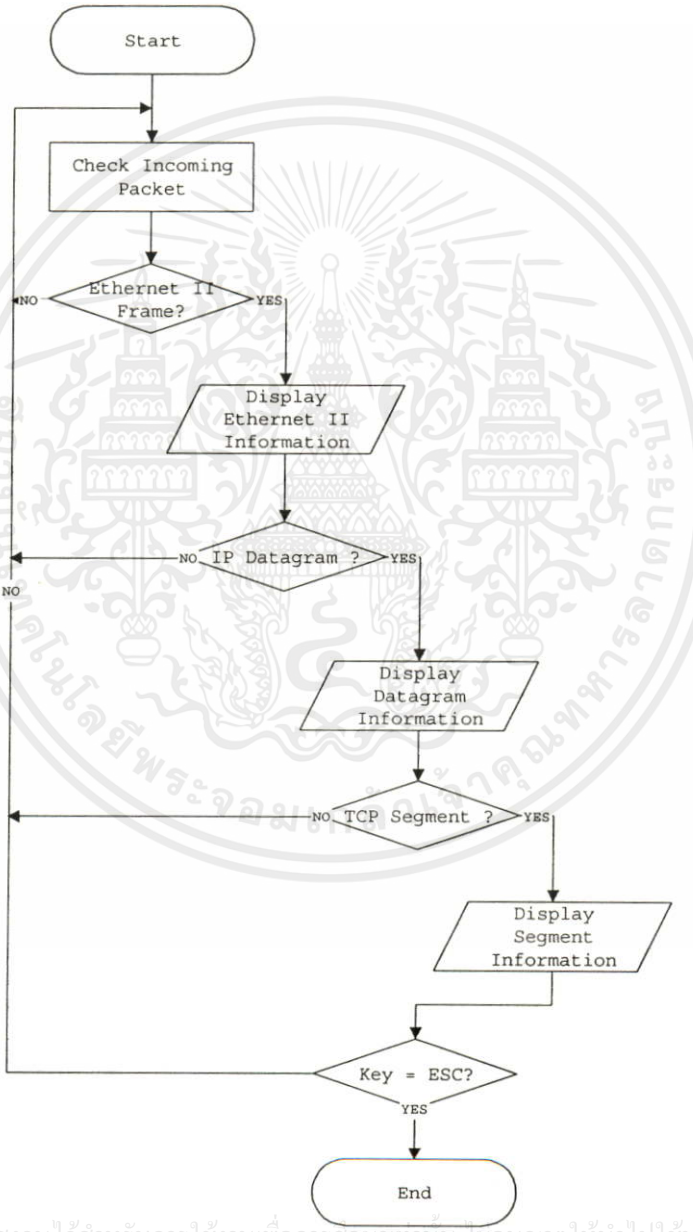
รูปที่ 6.6 ขั้นตอนการทำงานของโมดูลจัดเก็บแพ็กเก็ต



โมดูลแสดงข้อมูลแพ็กเก็ต

ทำหน้าที่แสดงข้อมูลในแพ็กเก็ต ไม่ว่าจะเป็นแพ็กเก็ตที่จับได้ในขณะนั้น หรือจากที่เก็บไว้ในไฟล์ และบัฟเฟอร์

รูปที่ 6.7 ขั้นตอนการทำงานของโมดูลตรวจสอบแพ็กเก็ต



บทที่ 7

การประยุกต์ใช้งาน

การทดสอบโปรแกรมที่จัดสร้างขึ้นนี้ได้ทำการทดสอบเป็นหลายๆแบบบนระบบเครือข่ายคอมพิวเตอร์ขนาดเล็กที่มีใช้งานอยู่จริง เช่น การตรวจสอบสภาพความหนาแน่นของการจราจรข้อมูลในระบบเครือข่าย เพื่อจะได้ทราบลักษณะการใช้งานของระบบเครือข่ายโดยดูจากจำนวนข้อมูลที่สามารถตรวจจับได้ในช่วงเวลาต่างๆ การตรวจสอบแพ็กเก็ตที่รับ-ส่งจากระบบเครือข่ายอื่น เพื่อที่จะทราบวิธีการจัดเส้นทางของข้อมูล การตรวจสอบโพรโตคอลของแพ็กเก็ตที่ตรวจจับได้เพื่อทราบลักษณะการใช้งานโพรโตคอลในระดับโปรแกรมประยุกต์ต่างๆของผู้ใช้งานในระบบเครือข่ายนั้นๆ และการวิเคราะห์ข้อมูลในแพ็กเก็ตที่สามารถดักจับได้ เพื่อสามารถศึกษาการรับส่งข้อมูลบนโพรโตคอลในระดับโปรแกรมประยุกต์บางตัวที่นิยมใช้กัน เช่น TELNET หรือ HTTP

7.1 การตรวจสอบสภาพความหนาแน่นของการจราจรข้อมูลในระบบเครือข่าย

โปรแกรมที่จัดทำขึ้นนี้สามารถอาศัยการทำงานของแพ็กเก็ตไดรเวอร์ในการดักจับแพ็กเก็ตที่วิ่งอยู่บนระบบเครือข่ายอีเทอร์เน็ต ซึ่งจะเป็นข้อมูลในแพ็กเก็ตที่เกิดขึ้นจริงในระบบเครือข่ายก่อนที่จะเลือกพิจารณาวิเคราะห์ข้อมูลในระดับชั้นที่สูงกว่าต่อไป

ในการหาความหนาแน่นของการจราจรข้อมูลในระบบเครือข่ายท้องถิ่นแบบอีเทอร์เน็ตสามารถคำนวณได้จาก

TotalPacket = ค่าผลรวมทั้งหมดของความยาวของแพ็กเก็ตข้อมูลที่สามารถตรวจจับได้ในระดับชั้นอีเทอร์เน็ตภายใน 1 วินาที มีหน่วยเป็นไบท์

MaxPacketLengt ค่าขนาดจำนวนข้อมูลสูงสุดของอีเทอร์เน็ตเฟรม มีหน่วยเป็นไบท์

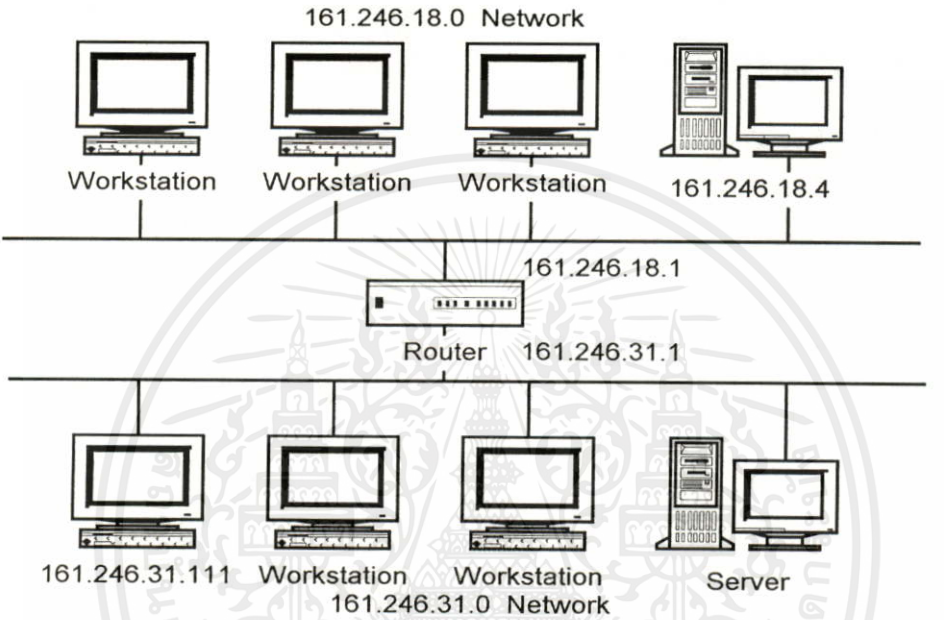
= 10,000,000 บิต ต่อวินาที

= 1,250,000 ไบท์ ต่อวินาที

ค่าความหนาแน่นการจราจร = (TotalPacket * 100) / MaxPacketLength %

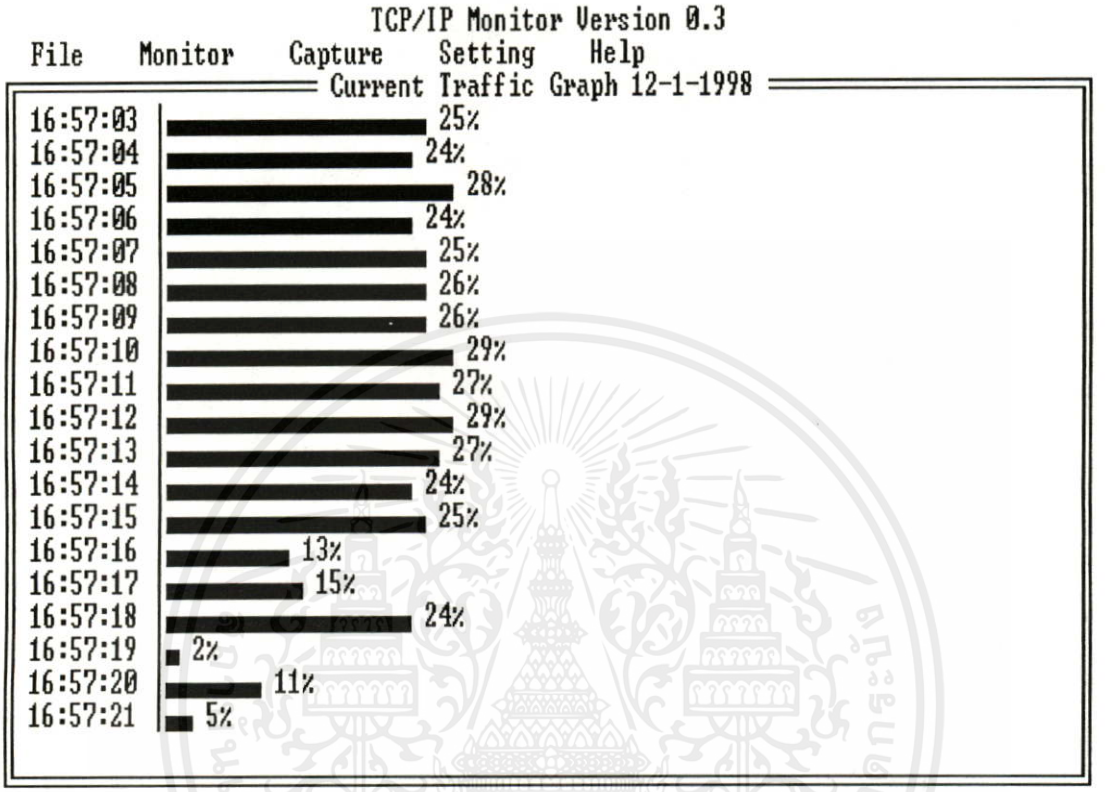
เมื่อทำการทดลองให้โปรแกรมตรวจสอบสภาพการจราจรของข้อมูลในระบบเครือข่ายที่ใช้งานอยู่จริง ในรูปที่ 7.1

รูปที่ 7.1 ระบบเครือข่ายที่ใช้ทดสอบ



ระบบเครือข่ายที่ใช้ทดสอบเป็นระบบเครือข่ายอีเทอร์เน็ตในคลาส B 161.246.0.0 แบ่งเป็นระบบเครือข่ายย่อย 2 เครือข่ายย่อย เป็นระบบเครือข่ายเบอร์ 161.246.18.0 และระบบเครือข่ายเบอร์ 161.246.31.0 ทั้งสองเครือข่ายติดต่อกันโดยใช้เราท์เตอร์ที่มีไอพีแอดเดรสเบอร์ 161.246.18.1 และ 161.246.31.1 เมื่อใช้โปรแกรมทดสอบสภาวะการจราจรบนเครือข่ายนี้โดยใช้การสุ่มเวลาในช่วงสั้นๆ ใน 1 วินาที จะได้ผลดังแสดงในรูปที่ 7.2

รูปที่ 7.2 แสดงสภาพการจราจรของข้อมูลในระบบเครือข่ายขณะนั้น



จากผลการทดสอบ สามารถแสดงสภาพความหนาแน่นของการจราจรข้อมูลในระบบเครือข่ายอีเทอร์เน็ตที่ช่วงเวลาต่างๆได้ โดยปกติแล้ว ลักษณะการรับส่งข้อมูลของระบบอีเทอร์เน็ต ค่านี้ไม่ควรมีค่าสูงเกินกว่า 50 % เนื่องจากจะทำให้ประสิทธิภาพโดยรวมของระบบลดลง ซึ่งหากขนาดการจราจรมีค่าสูงเกินกว่า 50 % เป็นระยะเวลาติดต่อกันนานๆ แสดงว่าระบบเครือข่ายนี้มีการจราจรของข้อมูลหนาแน่นผิดปกติ ซึ่งควรที่จะตรวจสอบหาสาเหตุต่อไป

7.2 การตรวจสอบแพ็กเก็ตที่รับ-ส่งจากระบบเครือข่ายอื่น

การทดสอบการทำงานในแบบนี้โดยการทำการล็อกอินระยะไกล โดยใช้โปรแกรม telnet จากสถานีงาน 161.246.31.111 ที่อยู่บนระบบเครือข่าย 161.246.31.0 ไปยังสถานีงาน 161.246.18.4 ที่อยู่บนระบบเครือข่าย 161.246.18.0 โดยใช้ระบบเครือข่ายดังรูปที่ 7.1 ซึ่งสามารถอธิบายการติดต่อได้ดังนี้

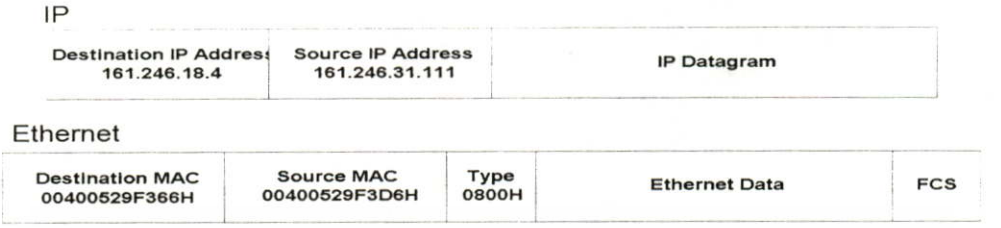
สถานีงาน 161.246.31.111 ที่อยู่บนระบบเครือข่าย 161.246.31.0 ต้องการที่จะติดต่อกับสถานีงาน 161.246.18.4 ที่อยู่บนระบบเครือข่าย 161.246.18.0 โดยใช้โพรโตคอล TELNET ในการล็อกอินระยะไกล จะมีการรับส่งเซ็กเมนต์แพ็กเก็ตของทีซีพีระหว่างสถานีงาน 161.246.31.111 กับสถานีงาน 161.246.18.4 โดยผ่านเราท์เตอร์ที่ทำการเชื่อมต่อทั้งสองระบบเครือข่ายนั้นอยู่ เราท์เตอร์ตัวนี้จะมีไอพีแอดเดรสอยู่ 2 เบอร์ คือ 161.246.18.1 ติดต่อกับระบบเครือข่าย 161.246.18.0 และเบอร์ 161.246.31.1 ติดต่อกับระบบเครือข่าย 161.246.31.0 การติดต่อสื่อสารของทีซีพี/ไอพีจะใช้ไอพีแอดเดรสในการระบุสถานีงานต้นทางและสถานีงานปลายทาง แต่เมื่อมีการส่งแพ็กเก็ตของทีซีพี/ไอพีไปในระบบเครือข่ายที่ใช้งานจริง จำเป็นต้องมีการแปลงไอพีแอดเดรสให้อยู่ในรูปฟิสิคัลแอดเดรสอ้างอิงตามระบบเครือข่ายที่ใช้งานจริงด้วย ซึ่งในที่นี้คือระบบอีเทอร์เน็ต จะมีขั้นตอนการรับส่งแพ็กเก็ตดังนี้

แพ็กเก็ตบนระบบเครือข่าย 161.246.31.0

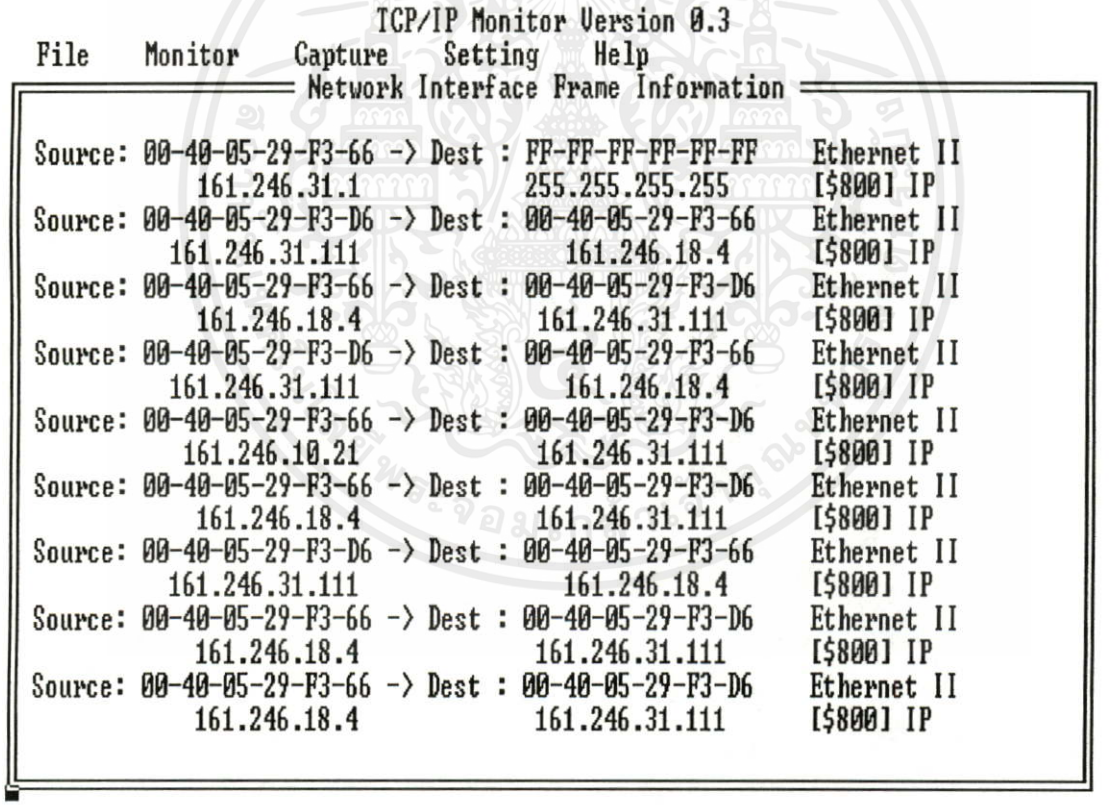
ในการที่สถานีงานต้นทางจะทำการส่งแพ็กเก็ตไปยังเครือข่ายอื่นที่ไม่ใช่เครือข่ายของตนเอง มันจะทำการส่งแพ็กเก็ตนั้นไปยังเราท์เตอร์หลัก (default router) ที่เชื่อมต่อกับระบบเครือข่ายอื่นอยู่ ซึ่งเราท์เตอร์ในที่นี้จะเชื่อมต่อกับระบบเครือข่ายทางฝั่งของสถานีต้นทางโดยใช้ไอพีแอดเดรสเบอร์ 161.246.31.1

เมื่อสถานีงานต้นทางต้องการติดต่อกับสถานีงานปลายทางใดๆก็ตามโดยใช้ไอพีแอดเดรสระบุ มันจะต้องทราบฟิสิคัลแอดเดรสหรือแมคแอดเดรสของสถานีปลายทางก่อนเสมอ โดยการใช้โพรโตคอลเออาร์พีในการแปลงข้อมูลจากไอพีแอดเดรสไปเป็นแมคแอดเดรส สถานีงานต้นทางจะทำการส่งแพ็กเก็ต ARP Request ออกไปและรอให้สถานีงานปลายทางตอบกลับมาพร้อมกับแมคแอดเดรสของสถานีงานปลายทางนั้นๆ ในที่นี้ เราท์เตอร์จะทำการตอบกลับแมคแอดเดรสของตัวมันไปให้ (00-40-05-29-F3-66H) จากนั้นสถานีงาน 161.246.31.111 จะทำการส่งข้อมูลที่เป็นเดต้าแกรมลงไปในเฟรมอีเทอร์เน็ตที่มีแมคแอดเดรสของสถานีงานปลายทางเป็นเบอร์ 00-40-05-29-F3-66H ของเราท์เตอร์ และระบุแมคแอดเดรสของตัวมันเองเป็นเบอร์ 00-40-05-29-F3-D6H พร้อมกับบอกด้วยว่าไทป์ฟิลด์เป็นเบอร์ 0800H ซึ่งเป็นเบอร์ไทป์ฟิลด์ของไอพี ซึ่งสามารถแสดงโครงสร้างของอีเทอร์เน็ตเฟรมได้ดังนี้

รูปที่ 7.3 การจัดเดต้าแกรมลงในอีเทอร์เน็ตเฟรมบนเครือข่าย 161.246.31.0



รูปที่ 7.4 แสดงทิศทางของแพ็กเก็ตในระบบเครือข่าย 161.246.31.0



แพ็กเก็ตบนระบบเครือข่าย 161.246.18.0

เมื่อเราท์เตอร์ฝั่ง 161.246.31.1 ทำการส่งแพ็กเก็ตนี้มาให้ระบบเครือข่าย 161.246.18.0 เราท์เตอร์ก็จะทราบเส้นทางต่อไปของแพ็กเก็ตนี้ได้จากตารางข่าวสารการจัดเส้นทางของตัวเราท์เตอร์เอง ซึ่งมันจะพบว่าแพ็กเก็ตนี้ต้องการส่งต่อไปยังระบบเครือข่ายในฝั่งที่มันเชื่อมต่ออยู่แล้วทางไอพีแอดเดรสเบอร์ 161.246.18.1 ดังนั้นเราท์เตอร์ก็จะทำการส่งแพ็กเก็ตนี้ไปยังสถานีงานปลายทางได้ ซึ่งจะใช้โพรโตคอลเออาร์พีทำการหาแมคแอดเดรสของไอพีแอดเดรสของสถานีงานปลายทางของแพ็กเก็ตนี้โดยการส่ง ARP Request ออกไปสอบถามกับสถานีงานที่เชื่อมต่ออยู่บนระบบเครือข่ายนี้ เมื่อเราท์เตอร์ได้แมคแอดเดรสของสถานีงานปลายทางซึ่งเป็นเบอร์ 00-00-77-02-B5-8DH มาก็จะนำแมคแอดเดรสที่ได้มานี้ไปใส่ในฟิลด์แมคแอดเดรสสถานีงานปลายทางของแพ็กเก็ตนี้ พร้อมกับระบุแมคแอดเดรสของสถานีต้นทางเบอร์ 00-00-81-11-3A-BEH ซึ่งคือตัวมัน แล้วทำการส่งแพ็กเก็ตนี้ไปให้

เมื่อสถานีงานปลายทาง (161.246.18.4) ได้รับแพ็กเก็ตมาแล้วก็จะทำการถอดรหัสข่าวสารอีเทอร์เน็ตส่วนหัวออก จากนั้นจะพิจารณาที่ไทป์ฟิลด์ซึ่งจะพบว่าแพ็กเก็ตนี้ใช้ไอพีโพรโตคอล ก็จะทำการส่งผ่านไปให้ไอพีโมดูลของมันเป็นผู้ดำเนินการต่อ ไอพีโมดูลจะทำการตีความข้อมูลภายในซึ่งก็จะพบว่าเดต้าแกรมนี้บรรจุข่าวสารของทีซีพีไว้ ก็จะส่งไปให้ทีซีพีโมดูลทำการประมวลผลต่อ ทีซีพีโมดูลจะดูหมายเลขพอร์ทที่ระบุไว้ แล้วจะส่งข้อมูลที่ได้มาไปยังโพรโตคอลที่เกี่ยวข้องในระดับโปรแกรมประยุกต์ของ TELNET ต่อไป

รูปที่ 7.5 การจัดเดต้าแกรมลงในอีเทอร์เน็ตเฟรมบนเครือข่าย 161.246.18.0

IP

Destination IP Address 161.246.18.4	Source IP Address 161.246.31.111	IP Datagram

Ethernet

Destination MAC 00007702B58DH	Source MAC 000081113ABEH	Type 0800H	Ethernet Data	FCS

รูปที่ 7.6 แสดงทิศทางของแพ็กเก็ตในระบบเครือข่าย 161.246.18.0

```
                    TCP/IP Monitor Version 0.3
File    Monitor    Capture    Setting    Help
              Network Interface Frame Information
Source: 00-00-81-06-38-91 -> Dest : 00-00-81-11-3A-BE    Ethernet II
        161.246.18.2              161.246.18.1           [$800] IP
Source: 00-00-81-11-3A-BE -> Dest : 00-40-05-29-F0-F5    Ethernet II
        202.49.47.229             161.246.18.203         [$800] IP
Source: 00-00-81-11-3A-BE -> Dest : 00-AA-00-69-76-20    Ethernet II
        204.48.18.135             161.246.18.222         [$800] IP
Source: 00-00-77-02-B5-8D -> Dest : 00-C0-DF-E0-3F-9D    Ethernet II
        161.246.18.4              161.246.18.102         [$800] IP
Source: 00-AA-00-69-76-20 -> Dest : 00-00-81-11-3A-BE    Ethernet II
        161.246.18.222            204.48.18.135          [$800] IP
Source: 00-00-81-11-3A-BE -> Dest : 00-AA-00-69-76-20    Ethernet II
        204.48.18.135             161.246.18.222         [$800] IP
Source: 00-00-81-06-38-91 -> Dest : 00-00-81-11-3A-BE    Ethernet II
        161.246.18.2              161.246.18.1           [$800] IP
Source: 00-AA-00-69-76-20 -> Dest : 00-00-81-11-3A-BE    Ethernet II
        161.246.18.222            204.48.18.135          [$800] IP
Source: 00-00-81-11-3A-BE -> Dest : 00-AA-00-69-76-20    Ethernet II
        204.48.18.135             161.246.18.222         [$800] IP
Source: 00-00-77-02-B5-8D -> Dest : 00-C0-DF-E0-3F-9D    Ethernet II
        161.246.18.4              161.246.18.102         [$800] IP
```

7.3 การตรวจสอบโพรโตคอลของแพ็กเก็ตที่ตรวจจับได้

แพ็กเก็ตที่โปรแกรมสามารถตรวจจับได้นั้น เมื่อนำมาพิจารณาเฉพาะอีเทอร์เน็ตเฟรมที่มีฟิลด์ไทป์เป็นแบบไอพีและนำมาหาค่าความหนาแน่นของการใช้งานโพรโตคอลต่างๆในระดับชั้นโฮสต์ทูโฮสต์ของทีซีพี/ไอพีจะพบว่า ข้อมูลที่รับส่งในระบบสื่อสารแบบทีซีพี/ไอพีจะใช้ข้อมูลในพอร์ตเป็นตัวกำหนดว่าต้องการติดต่อสื่อสารกับโพรโตคอลในชั้นโปรแกรมประยุกต์แบบใด

ลักษณะการสื่อสารของทีซีพี/ไอพีจะประกอบไปด้วย 2 ฝั่ง คือ ผู้ให้บริการ (server) กับผู้ขอบริการ (Client) ผู้ขอบริการจะทำการร้องขอเบอร์พอร์ตที่ว่างอยู่ที่สามารถนำมาใช้งานได้จากระบบจัดการสร้างเป็นซ็อคเก็ต เพื่อทำการติดต่อกับเบอร์พอร์ตของผู้ให้บริการซึ่งมักจะเป็นเบอร์พอร์ตมาตรฐานที่รู้จักกันดี ที่สำคัญได้แก่

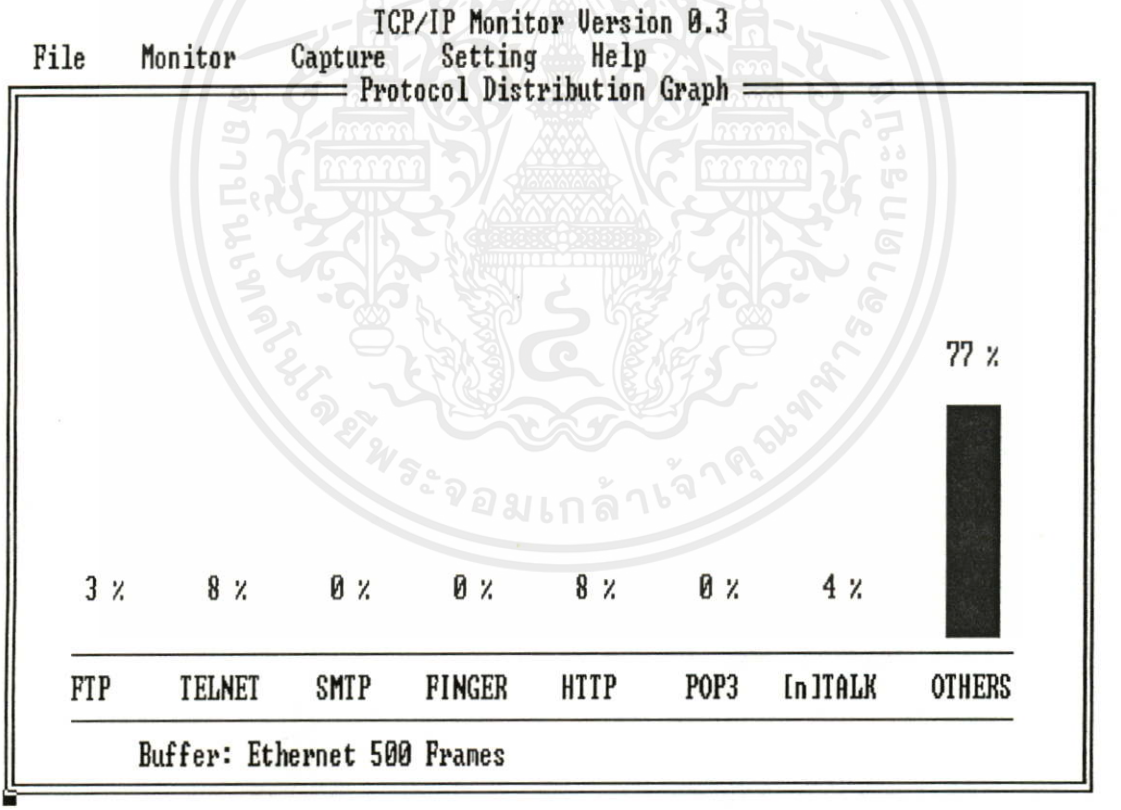
บริการล็อกอินระยะไกล ที่นิยมใช้กันบนระบบยูนิกซ์ จะใช้โพรโตคอล TELNET ซึ่งจะใช้ทีซีพีพอร์ตในการติดต่อสื่อสารเบอร์ 23

บริการไฟล์ทรานสเฟอร์ จะใช้โพรโตคอล FTP ซึ่งจะใช้ทีซีพีพอร์ทเบอร์ 20 และ 21

บริการเวิลด์ไวด์เว็บ จะใช้โพรโตคอล HTTP ซึ่งจะใช้ทีซีพีพอร์ทเบอร์ 80

ซึ่งโปรแกรมที่จัดทำขึ้นนี้สามารถรวบรวมสถิติการใช้งานพอร์ทต่างๆตามที่กำหนดไว้ได้ ว่าต้องการทราบสถิติการใช้งานของบริการโปรแกรมประยุกต์แบบใดตามเบอร์พอร์ทที่มีการกำหนดไว้ล่วงหน้าก่อนแล้ว ส่วนแพ็กเก็ตที่ติดต่อกันโดยใช้พอร์ทเบอร์อื่นๆที่ไม่ได้เป็น well-known port จะเกิดขึ้นได้นั้นจะเป็นการติดต่อกันระหว่างโปรแกรมประยุกต์บนระบบเครือข่ายโดยตรง ซึ่งจะพบว่ามักจะมีค่ามากกว่าพอร์ทที่กำหนดไว้ เนื่องจากเป็นเบอร์พอร์ทที่โปรแกรมประยุกต์ใช้ติดต่อกันชั่วคราว

รูปที่ 7.7 ความถี่ของโพรโตคอลชั้นโปรแกรมประยุกต์ในแพ็กเก็ตที่ตรวจจับได้



7.4 การแสดงข้อมูลในแพ็กเก็ตที่ตรวจจับได้ขณะนั้น

ระบบที่จัดสร้างขึ้นนี้สามารถนำข้อมูลในแพ็กเก็ตที่ตรวจจับได้มาเก็บลงไปในหน่วยความจำเพื่อใช้วิเคราะห์ข้อมูลที่เกิดขึ้นในเวลาต่อไปได้ ซึ่งอาจจะเก็บไว้ในฮาร์ดดิสก์ หรือลงไปในหน่วยความจำของเครื่องเอง แต่จะมีการเก็บข้อมูลได้จำกัดตามพื้นที่หน่วยความจำที่เหลือ ในขั้นตอนการจัดเก็บนั้นสามารถเลือกเบอร์สถานีงานที่สนใจเท่านั้นได้ด้วย โดยการกำหนดเบอร์แมคของแอดเดรสของสถานีงานนั้นๆไว้ก่อน เมื่อมีแพ็กเก็ตที่เกิดขึ้นและมีหมายเลขต้นทางหรือปลายทางตรงกับหมายเลขของสถานีงานนั้นๆก็จะทำการเก็บไว้โดยไม่จำเป็นต้องสนใจข้อมูลแพ็กเก็ตที่เกิดขึ้นจริงทั้งระบบ ทำให้สามารถเลือกเฉพาะสถานีงานที่สนใจเท่านั้นได้ ในรูปที่ 7.8 จะแสดงข้อมูลตัวอย่างที่ตรวจจับได้ซึ่งเป็นแพ็กเก็ตที่ได้จากโปรแกรม IRC (Internet Relay Chat) ส่วนรูปที่ 7.9 จะเป็นข้อมูลที่ได้จากโปรแกรม ftp

รูปที่ 7.8 แสดงข้อมูลในแพ็กเก็ตที่ตรวจจับได้

```
                    TCP/IP Monitor Version 0.3
  File    Monitor    Capture    Setting    Help
          ═══════ Real-Time Packet Information ═══════
0040   69 61 31 33 40 64 69 61   6C 33 33 2E 61 75 6E 65    ia13@dial33.aune
0050   74 2E 61 75 2E 61 63 2E   74 68 20 50 52 49 56 4D    t.au.ac.th PRIVM
0060   53 47 20 23 73 69 61 6D   20 3A 60 4A 6F 59 32 20    SG #siam :`JoY2
0070   3A 20 63 68 65 72 7E 20   2E 2E 2E 20 6D 61 69 20    : cher~ ... mai
0080   79 6F 6D 20 70 65 6E 20   66 61 6E 20 43 6F 7A 7A    yom pen fan Cozz
0090   20 62 63 75 7A 20 6D 65   65 20 46 61 6E 20 6C 61     bcuz mee Fan la
00A0   65 77 20 6E 65 65 20 61   6E 67 0D 0A 7E 29 04 0E    ew nee ang..~)..

20:52:07:30 - 60 Bytes. 161.246.18.225 -> 202.6.101.223
0000   00 00 81 11 3A BE 00 40   05 2F 20 93 08 00 45 00    ....:..@./ ...E.
0010   00 28 D4 1C 40 00 20 06   A1 F6 A1 F6 12 E1 CA 06    .(..@. .........
0020   65 DF 05 60 1A 0B 01 45   69 88 1C 7C 57 FF 50 10    e..`...Ei..|W.P.
0030   1F 21 AD 42 00 00 20 20   20 20 20 20 7A 65 6E 74    .!.B..      zent

20:52:07:80 - 60 Bytes. 161.246.18.2 -> 161.246.18.1
0000   00 00 81 11 3A BE 00 00   81 06 38 91 08 00 45 00    ....:.....8...E.
0010   00 1C 7D 03 00 00 FF 01   D6 ED A1 F6 12 02 A1 F6    ..}.............
0020   12 01 08 00 20 6B A4 1F   33 75 04 52 00 02 00 04    .... k..3u.R....
0030   54 45 4C 45 4C 41 4E 00   00 00 00 00 7A 65 6E 74    TELELAN.....zent
```

รูปที่ 7.9 แสดงข้อมูลในแพ็กเก็ตที่ตรวจจับได้

```
                    TCP/IP Monitor Version 0.3
 File    Monitor    Capture    Setting    Help
                 Real-Time Packet Information
 0070   68 6F 73 74 6D 61 73 74   65 72 C0 3A 77 17 67 C8    /.ns.thnic.net..
 0070   68 6F 73 74 6D 61 73 74   65 72 C0 3A 77 17 67 C8    hostmaster.:w.g.
 0080   00 00 54 60 00 00 01 2C   00 12 75 00 00 00 A8 C0    ..T`...,..u.....
 0090   53 09 3A 53 CE 53 93 09   93 07 01 93 01 01 74 08    S.:S.S........t.

 20:51:10:35 - 477 Bytes. 161.246.18.203 -> 202.134.225.118
 0000   00 00 81 11 3A BE 00 40   05 29 F0 F5 08 00 45 00    ....:..@.)....E.
 0010   01 CF 50 39 40 00 20 06   A8 31 A1 F6 12 CB CA 86    ..P9@. ..1......
 0020   E1 76 00 15 05 18 00 49   9E 6F 00 39 BF 04 50 18    .v.....I.o.9..P.
 0030   B2 5C CF 37 00 00 32 33   30 2D 20 2A 2A 2A 2A 2A    .\.7..230- *****
 0040   2A 2A 2A 2A 2A 2A 2A 2A   2A 2A 2A 2A 2A 2A 2A 2A    **********
 0050   2A 2A 2A 2A 2A 2A 2A 2A   2A 2A 2A 2A 2A 2A 2A 2A    **********
 0060   2A 2A 2A 2A 2A 2A 2A 2A   2A 2A 2A 2A 2A 2A 2A 2A    **********
 0070   2A 2A 2A 2A 2A 2A 2A 2A   2A 2A 2A 2A 2A 2A 2A 2A    **********
 0080   2A 2A 0D 0A 32 33 30 2D   20 20 20 20 20 20 20 20    **..230-
 0090   20 20 20 20 20 20 20 20   20 20 57 65 6C 63 6F 6D        Welcom
 00A0   65 20 74 6F 20 4E 65 77   54 79 70 65 20 46 54 50    e to NewType FTP
 00B0   20 73 65 72 76 69 63 65   0D 0A 32 33 30 2D 20 20     service..230-
 00C0   20 20 20 20 20 20 20 20   20 20 20 20 20 20 20 20
 00D0   3D 3D 3D 3D 3D 3D 3D 3D   3D 3D 3D 3D 3D 3D 3D 3D    ================
```

ข้อมูลที่ตรวจจับได้สามารถอธิบายรายละเอียดได้ดังนี้ ข้อมูลลำดับที่ 0000H-0005H เป็นแมคแอดเดรสของสถานีงานปลายทาง เบอร์ 00-00-81-11-3A-BEH ลำดับที่ 0006H-000BH เป็นแมคแอดเดรสของสถานีงานต้นทาง เบอร์ 00-40-05-29-F0-F5H ลำดับที่ 000CH-000DH แสดงชนิดของแพ็กเก็ต ในที่นี้มีค่า 0800H ซึ่งหมายถึงโพรโตคอลไอพี เมื่อทราบว่าแพกเกตนี้บรรจุเดต้าแกรมไว้แล้ว ก็สามารถวิเคราะห์ส่วนหัวไอพีได้ต่อไปดังนี้ ตำแหน่ง 4 บิทบนของข้อมูลลำดับที่ 000EH จะเป็นค่า version ของไอพี ซึ่งปัจจุบันใช้เป็นค่า 4 ตำแหน่งข้อมูล 4 บิทล่างของลำดับที่ 000EH จะเป็นค่า IHL (Internet Header Length) ซึ่งเป็นขนาดความยาวของส่วนหัวในรูปของเวิร์ดขนาด 32 บิต ค่าที่ได้เป็น 5 หมายถึงขนาดส่วนหัวนี้ยาวเท่ากับ 20 ไบท์และไม่มีส่วนของฟิลด์ออปชั่นและแพ็ดดิ้ง ลำดับที่ 000FH เป็นค่า ToS (Type of Service) มีค่าเท่ากับ 00 ลำดับที่ 0010H-0011H เป็นค่า Total Length ค่าความยาวของเดต้าแกรมในหน่วยของอ็อคเต็ท ซึ่งจะรวมความยาวส่วนหัวและขนาดข้อมูล มีค่า 01CFH ซึ่งมีค่าเป็น 463 ไบต์ ลำดับที่ 0012H-0013H เป็นค่า Identification มีค่าเท่ากับ 5039H ตำแหน่งข้อมูล 4 บิทบนของข้อมูลลำดับที่ 0014H เป็นค่า Flag มีค่าเท่ากับ 0 ตำแหน่ง 4 บิทล่างของลำดับที่ 0014H รวมกับข้อมูลลำดับที่ 0015H เป็นค่า Fragment

Offset มีค่าเท่ากับ 0 ตำแหน่งข้อมูลลำดับที่ 0016H แสดงค่า Time to Live มีค่าเป็น 20H เท่ากับ 32 ลำดับที่ 0017H เป็นค่าโพรโตคอล มีค่าเท่ากับ 6 หมายถึงโพรโตคอลทีซีพี ลำดับที่ 0018H-0019H เป็นค่า Header Checksum มีค่าเท่ากับ A831H ลำดับที่ 001AH-001DH เป็นไอพีแอดเดรสของสถานีงานต้นทาง มีค่าเท่ากับ A1F612CBH เมื่อแปลงให้อยู่ในรูปของ dotted decimal notation จะได้เป็น 161.246.18.203 ลำดับที่ 001EH-0021H เป็นไอพีแอดเดรสของสถานีงานปลายทาง มีค่าเท่ากับ CA86E176H เมื่อแปลงให้อยู่ในรูปของ dotted decimal notation จะได้เป็น 202.134.225.118 เนื่องจากเดต้าแกรมนี้บรรจุเซกเมนต์ของทีซีพี เมื่อใช้ส่วนหัวของทีซีพีอธิบายต่อไปจะได้ดังนี้ ลำดับที่ 0022H-0023H เป็นค่าทีซีพีพอร์ทต้นทาง มีค่าเป็น 15H ซึ่งเป็นพอร์ทเบอร์ 21 ของ FTP Control ลำดับที่ 0024H-0025H เป็นค่าทีซีพีพอร์ทปลายทาง มีค่าเป็น 0518H ซึ่งเป็นทีซีพีพอร์ทที่ 202.134.225.118 จองไว้ใช้งานอยู่ในขณะนี้ ลำดับที่ 0026H-0029H เป็นค่า Sequence Number มีค่าเป็น 00499E6FH ลำดับที่ 002AH-002DH เป็นค่า Acknowledgement Number มีค่าเป็น 0039BF04H ข้อมูล 4 บิทบนของลำดับที่ 002EH เป็นค่า Data Offset ซึ่งเป็นค่าความยาวของส่วนหัวทีซีพีในรูปของเวิร์ดขนาด 32 บิต มีค่าเท่ากับ 5 หมายความว่า เซกเมนต์นี้ไม่มีฟิลด์ออปชั่นและแพ็ดดิ้ง ลำดับที่ 0030H-0031H เป็นค่า window มีค่าเท่ากับ B25CH ลำดับที่ 0032H-0033H เป็นค่า Checksum มีค่าเท่ากับ CF37H ลำดับที่ 0034H-0035H เป็นค่า Urgent Pointer มีค่าเท่ากับ 0000H และข้อมูลตั้งแต่ตำแหน่ง 0036H เป็นต้นไป คือข้อมูลของโพรโตคอล FTP ในที่นี้เป็นข้อความตอบรับจากสถานีงาน 161.246.18.203 ซึ่งกำลังทำหน้าที่เป็น FTP Server

7.5 การวิเคราะห์ข้อมูลในแพ็กเก็ต

โปรแกรมนี้สามารถวิเคราะห์ข้อมูลในแพ็กเก็ตที่ใช้บริการที่เป็นที่รู้จักกันดีทั่วไป และนำมาแสดงผลได้ดังตัวอย่างต่อไปนี้

รูปที่ 7.10 แสดงข้อมูลในแพ็กเก็ตที่ใช้โพรโตคอล TELNET

```
File     Monitor     Capture     Setting     Help
                Real-Time Packet Information
                Network Interface Frame
Source: 00-40-05-29-F3-66  Dest: 00-40-05-29-F3-D6        Ethernet II [0800]
      (   161.246.18.4)          ( 161.246.31.111)
          Internet Frame                    Host-to-Host Frame
Version          : 4                 Source Port       : TELNET
IP Header Length: 5                  Dest.  Port       : 0402
Type of Service  : 00                Sequence Number   : 076187E4
Total Length     : 002F              Acknowledgment No: 000A0FD5
Identifier       : 9974              Data OffSet       : 5018
Fragment Offset  : 0                 Window            : 2200
TimeToLive       : 254               Checksum          : 1315
Protocol         : TCP               Urgent Pointer    : 0000
Header Checksum  : ADF4
Source Address   : 161.246.18.4
Destination Addr: 161.246.31.111
                        Data
login: ...DG/UX Release 5.4R3.10 (zeus.telecom.e
```

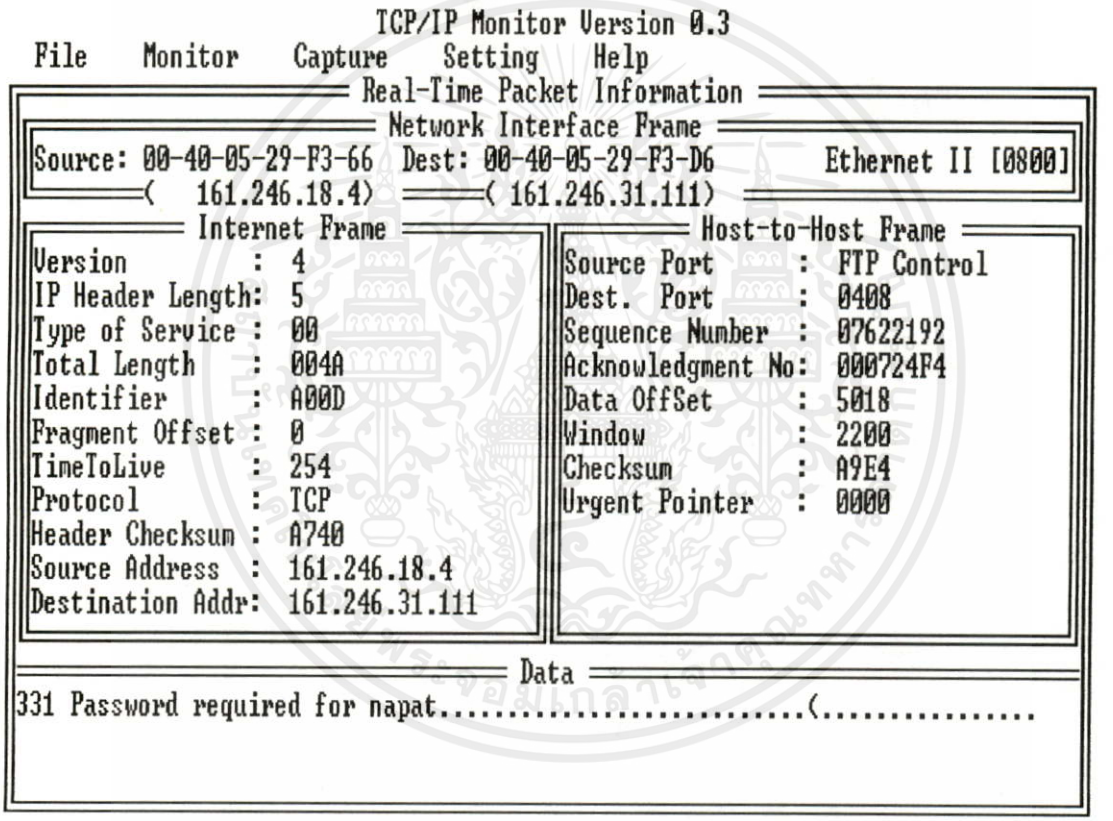
รูปที่ 7.11 แสดงข้อมูลในแพ็กเก็ตที่ใช้โพรโตคอล TELNET

```
                TCP/IP Monitor Version 0.3
File     Monitor     Capture     Setting     Help
                Real-Time Packet Information
0010   00 2E 9A 3D 00 00 FE 06   AD 2C A1 F6 12 04 A1 F6    ...=.....,......
0020   1F 6F 00 17 04 04 07 61   A7 02 00 01 13 F8 50 18    .o.....a......P.
0030   22 00 3B D2 00 00 FF FD   18 FF FD 1F 06 07 08 09    ".;.........tele

18:49:58:82 - 126 Bytes. 161.246.18.4 -> 161.246.31.111
0000   00 40 05 29 F3 D6 00 40   05 29 F3 66 08 00 45 00    .@.)...@.).f..E.
0010   00 6F 9A 42 00 00 FE 06   AC E6 A1 F6 12 04 A1 F6    .o.B............
0020   1F 6F 00 17 04 04 07 61   A7 17 00 01 14 13 50 18    .o.....a......P.
0030   22 00 AE 86 00 00 FF FB   03 FF FD 01 0D 0A 0D 0A    "...............
0040   44 47 2F 55 58 20 52 65   6C 65 61 73 65 20 35 2E    DG/UX Release 5.
0050   34 52 33 2E 31 30 20 28   7A 65 75 73 2E 74 65 6C    4R3.10 (zeus.tel
0060   65 63 6F 6D 2E 65 6E 67   2E 6B 6D 69 74 6C 2E 61    ecom.eng.kmitl.a
0070   63 2E 74 68 29 0D 0A 0D   00 0D 0A 0D 00 00 48 49    c.th).........HI

18:49:58:93 - 62 Bytes. 161.246.18.4 -> 161.246.31.111
0000   00 40 05 29 F3 D6 00 40   05 29 F3 66 08 00 45 00    .@.)...@.).f..E.
0010   00 2F 9A 46 00 00 FE 06   AD 22 A1 F6 12 04 A1 F6    ./.F....."......
0020   1F 6F 00 17 04 04 07 61   A7 64 00 01 14 1C 50 18    .o.....a.d....P.
0030   22 00 EF 54 00 00 6C 6F   67 69 6E 3A 20 00 0D 0A    "..T..login: ...
```

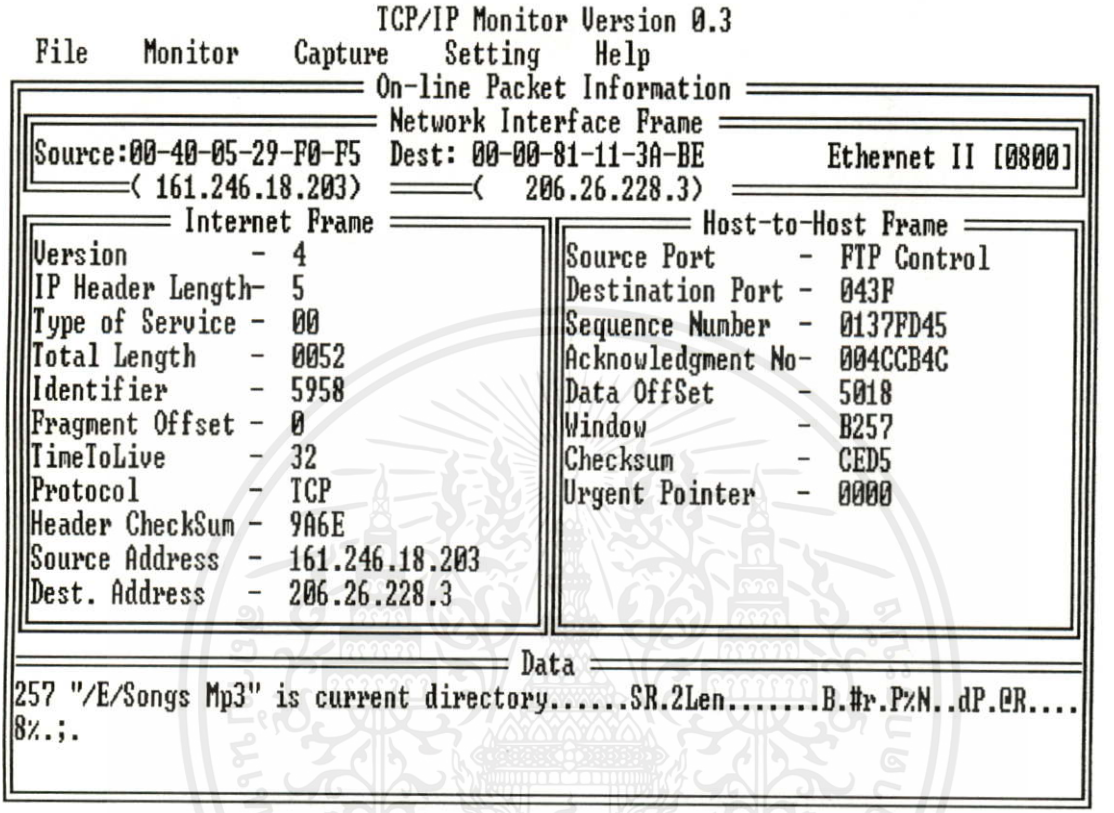
บริการรับส่งไฟล์ (File Transfer Application)

จะใช้โพรโตคอล FTP (File Transfer Protocol) ไฟล์ทรานสเฟอร์โพรโตคอลจะมีลักษณะสำคัญอย่างหนึ่ง คือ การทำงานจะใช้ทีซีพีพอร์ท 2 พอร์ทด้วยกัน คือพอร์ทหมายเลข 21 ใช้เป็นพอร์ทติดต่อในการส่งคำสั่งควบคุมของไฟล์ทรานสเฟอร์โพรโตคอล ส่วนพอร์ทหมายเลข 20 จะใช้เป็นพอร์ทรับส่งข้อมูล

รูปที่ 7.12 แสดงข้อมูลในแพ็กเก็ตที่ใช้โพรโตคอล FTP



รูปที่ 7.13 แสดงข้อมูลในแพ็กเก็ตที่ใช้โพรโตคอล FTP

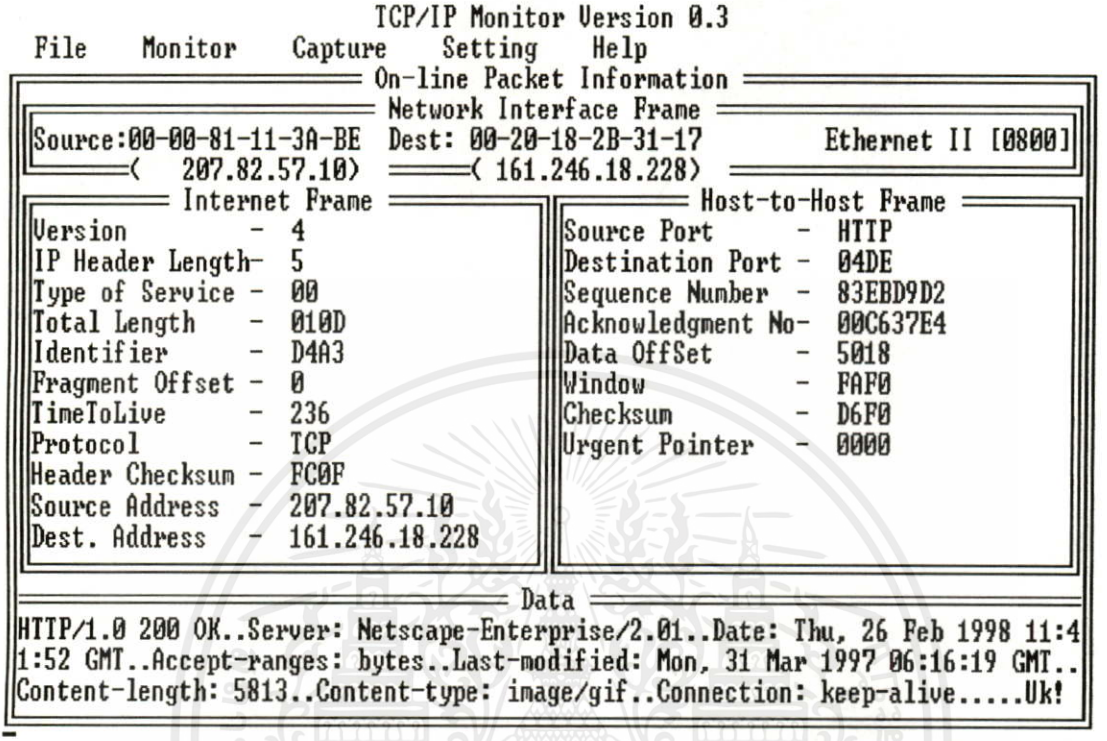


บริการเวิลด์ไวด์เว็บ (World Wide Web)

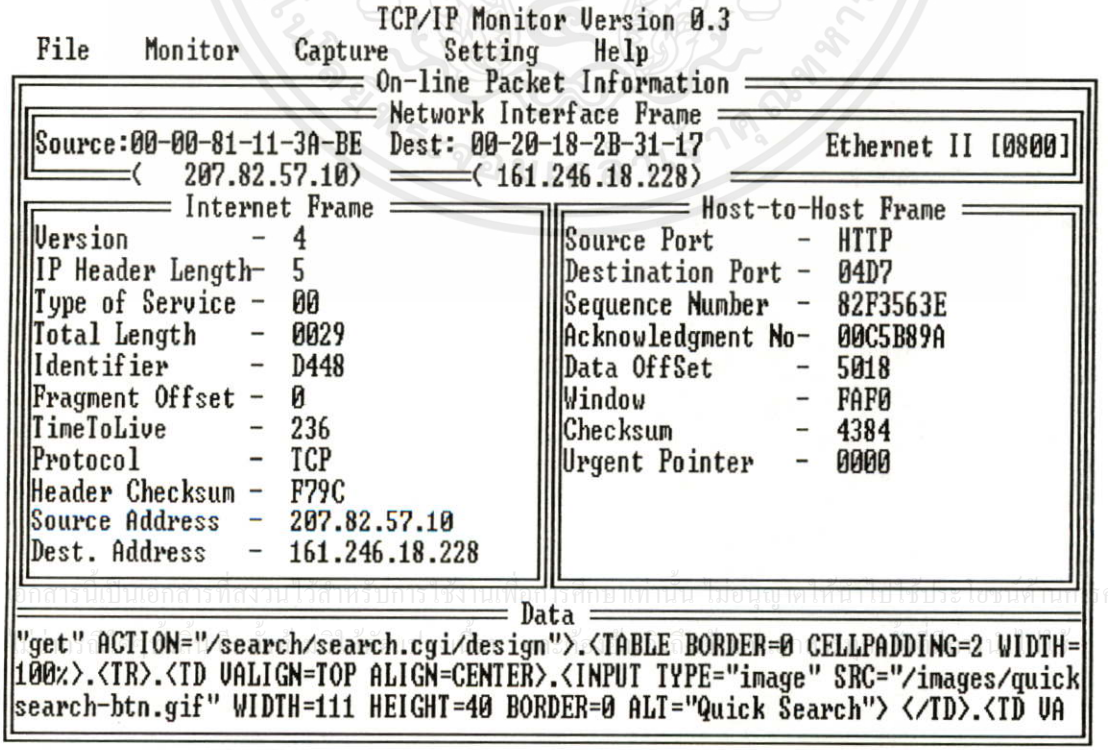
จะใช้โพรโตคอล HTTP (HyperText Transfer Protocol) ซึ่งจะใช้ทีซีพีพอร์ทเบอร์ 80 ในการติดต่อสื่อสารกันระหว่าง WWW Server กับ WWW Client ซึ่งจะใช้ WWW Browser เป็นผู้ทำการอ่านข้อมูลนั้นๆมาแสดงผลตามรูปแบบที่กำหนดในแบบไฮเปอร์มีเดียได้ ทำให้ใช้งานง่ายและเป็นที่นิยมอย่างมากในปัจจุบัน เวิลด์ไวด์เว็บสามารถใช้โพรโตคอลอื่นๆร่วมกับตัวมันได้อีกด้วย เช่น FTP หรือ NNTP ระบบเวิลด์ไวด์เว็บมีวิธีการเชื่อมต่อง่ายๆดังนี้

- Client ติดต่อไปยัง Server
- Client ส่งคำสั่งร้องขอ (request) ในระบบของเวิลด์ไวด์เว็บจะเรียกว่า วิธี (method) ซึ่งมีอยู่ 3 แบบ ได้แก่ GET, HEAD และ POST
- Server นำคำสั่งร้องขอมาประมวลผลแล้วส่งกลับไปให้

รูปที่ 7.14 แสดงข้อมูลในแพ็กเก็ตที่ใช้โพรโตคอล HTTP



รูปที่ 7.15 แสดงข้อมูลในแพ็กเก็ตที่ใช้โพรโตคอล HTTP



บทที่ 8

บทสรุปและข้อเสนอแนะ

บทสรุป

โครงงานที่จัดทำขึ้นนี้เป็นการศึกษาการทำงานภายในระบบเครือข่ายคอมพิวเตอร์ที่ใช้โพรโตคอลทีซีพี/ไอพีเพื่อให้สามารถทราบถึงการทำงานของโพรโตคอลในชุดของโพรโตคอลทีซีพี/ไอพีแต่ละเลเยอร์ และเฝ้าดูการใช้งานในระบบเครือข่ายเพื่อนำข้อมูลที่ได้มาปรับปรุงและพัฒนาระบบเครือข่าย โดยตรวจสอบแพ็กเก็ตที่ได้รับจากตัวกลางที่ใช้ส่งข้อมูลของระบบ ผ่านแพ็กเก็ตไดรเวอร์ซึ่งจะรับแพ็กเก็ตจากระบบเครือข่ายเข้ามาวิเคราะห์ถึงส่วนต่างๆที่จำเป็นต่อการทำงานและตรวจสอบระบบเครือข่าย

โปรแกรมที่จัดทำขึ้นนี้จะทำงานได้เฉพาะบนระบบเครือข่ายอีเทอร์เน็ต เนื่องจากอาศัยความสามารถของแพ็กเก็ตไดร์เวอร์อยู่ อย่างไรก็ตาม สามารถนำโปรแกรมที่จัดสร้างขึ้นนี้ไปใช้กับระบบเครือข่ายแบบอื่นได้ โดยต้องทำการหาโปรแกรมแพ็กเก็ตไดร์เวอร์ตัวใหม่และทำการแก้ไขโปรแกรมบางส่วนให้สามารถใช้งานร่วมกับแพ็กเก็ตไดร์เวอร์ตัวใหม่ได้

การตรวจจับแพ็กเก็ตนี้มีลักษณะเป็นการตรวจจับจากภายนอก ซึ่งสามารถจะทำงานบนสถานีงานใดๆก็ได้ที่เชื่อมต่อกับระบบเครือข่ายนั้นๆอยู่ แพ็กเก็ตที่ดักจับได้ในขั้นแรกจะเป็นแพ็กเก็ตในระดับอีเทอร์เน็ตเฟรมจากทุกสถานีงานในระบบเครือข่ายนั้น ซึ่งสามารถเลือกเฉพาะสถานีงานที่สนใจได้ ไม่ว่าจะเป็นสถานีงานต้นทาง หรือ สถานีงานปลายทาง เพียงแต่ต้องทราบเบอร์แมคแอดเดรสของสถานีงานนั้น หลังจากนั้นก็จะนำแพ็กเก็ตที่ได้มาถอดรหัสข่าวสารที่มีอยู่ในระดับของทีซีพี/ไอพีเพื่อวิเคราะห์ข้อมูลที่อยู่ในแพ็กเก็ตนั้นๆต่อไป

ข้อเสนอแนะ

การพัฒนาโปรแกรมที่ใช้ในการตรวจจับแพ็กเก็ตนี้ จากการทดลองทำงานสามารถตรวจสอบและดักจับแพ็กเก็ตที่ทำงานภายใต้โพรโตคอลทีซีพี/ไอพีได้ในระดับที่สามารถแสดงให้เห็นว่าการรับส่งข้อมูลบนทีซีพี/ไอพีโดยส่วนใหญ่จะไม่มีการเข้ารหัสข้อมูลไว้ ใครก็ตามที่สามารถเฝ้าดูและดักจับแพ็กเก็ตไว้ได้ก็สามารถที่จะรับรู้ข้อมูลที่กำลังรับส่งกันอยู่ในขณะนั้นได้ ทั้งนี้เป็นผลมาจากการสื่อสารบนระบบทีซีพี/ไอพี ใช้โครงสร้างการรับส่งข้อมูลของเดต้าแกรมบนระดับชั้น IP ในปัจจุบันเป็นเวอร์ชั่นที่ 4 (IP v4) ซึ่งไม่ได้ให้ความสำคัญกับการรักษาความปลอดภัยของข้อมูลที่มันทำหน้าที่บริการรับส่งข้อมูลมากนัก จึงได้มีการพัฒนาการทำงานในส่วนนี้เพิ่มขึ้นมาใน IP v6 ซึ่งยังอยู่ในช่วงการทดสอบการใช้งานกันอยู่ อย่างไรก็ตาม ในการติดต่อสื่อสารของทีซีพี/ไอพีในระดับชั้นที่สูงขึ้นก็ได้มีการพัฒนาโพรโตคอลในการรักษาความปลอดภัยเพิ่มเติมขึ้นมาหลายตัว เช่น kerberos, secure-Telnet หรือ S-HTTP เป็นต้น

บรรณานุกรม

[1] D.M. Chiu and R. Sudama, "Network Monitoring Explained Design and Application," ELLIS HORWOOD, 207p., 1992.

[2] D. E. Comer, "Internetworking with TCP/IP Volume I," Prentice-Hall International, pp. 51-425, 1991.

[3] FTP Software Inc., "PC/TCP Version 1.09 packet driver specification" FTP Software Inc., pp 5-14, 1994.

[4] C.Smythe, "Internetworking Designing the Right Architectures," Addison-Wesley, pp 95-116, 1995.

ภาคผนวก

ภาคผนวก

การกำหนดตัวเลขในอีเทอร์เนต

อีเทอร์เน็ตจะต้องมีการกำหนดค่าต่างๆเหล่านี้เพื่อให้เป็นมาตรฐานและสามารถใช้งานร่วมกันได้ ซึ่งจะมีส่วนประกอบต่างๆ ดังนี้ รหัสประเภท (Type code) , รหัสผู้ผลิต (vendor code) มัลติคาสท์ (รวมถึงบรอดคาสท์) แอดเดรส อีเทอร์เนตไทป์

ตำแหน่งอ็อกเต็ตที่ 13 และ 14 ของอีเทอร์เน็ตเฟรมเป็นส่วนของ อีเทอร์ไทป์ (Ethertype) หรือ IEEE 802.3 Length ประเภทของอีเทอร์ไทป์นั้น บริษัทซีรอก (Xerox) เป็นผู้ดูแลอยู่ บางค่าที่เป็นมาตรฐานจะใส่เครื่องหมาย "+" นอกนั้นถูกกำหนดเพื่อใช้เป็นการส่วนตัว ซึ่งข้อมูลนี้จะมีส่วนที่มาจาก ซีรอกพับลิกอีเทอร์เน็ตแพ็กเก็ตไทป์ (Xerox Public Ethernet Packet Type) , IEEE 802.3 standard , Network Manager และ ผู้ผลิต

โปรโตคอลอีเทอร์เน็ตไทป์ (Protocol Ethernet Type)

@	0000-05DC	IEEE802.3 Length Field (0.:1500.)
+	0101-01FF	Experimental
	0200	Xerox PUP (conflicts with 802.3 Length Field range)
	0201	Xerox PUP Address Translation (conflicts ...)
	0400	Nixdorf (conflicts with 802.3 Length Field)
+*	0800	DOD Internet Protocol (IP)
+	0805	X.25 Level 3
+*	0806	Address Resolution Protocol (ARP) (for IP)
	8005	HP Probe protocol
+	8019	Apollo DOMAIN
+	8035	Reverse Address Resolution Protocol (RARP)
	8037	IPX (Novell Netware?)
+	809B	EtherTalk (AppleTalk over Ethernet)
	80D5	IBM SNA Services over Ethernet
+	80F3	AppleTalk Address Resolution Protocol (AARP)

+	8137	Novell (old) NetWare IPX (ECONFIG E option)
+	8138	Novell, Inc.
	814C	SNMP over Ethernet (see RFC1089)
	817D	XTP
	86DD	IP version 6
	8888	HP LanProbe test?
+	9000	Loopback (Configuration Test Protocol)
	AAAA	DECNET? Used by VAX 6220 DEBNI
%	FF00	BBN VITAL-LanBridge cache wakeups

- ตัวเลขที่แสดงเป็นค่าเลขฐาน 16
- "*" ถูกใช้ในการทำอีเทอร์เน็ตบรอดคาสท์ ซึ่งอาจจะใช้ในมัลติคาสท์ก็ได้
- "%" อาจจะนำไปใช้เป็นการภายในโดยที่ยังไม่ได้ลงทะเบียน
- "+" โพรโตคอล ซึ่งถูกอ้างอิงโดยซีรอกในหนังสือ "COURIER (page 8-9) October 1988 issue of ในหัวข้อ publicly assigned numbers
- "@" ตามที่อธิบายใน COURIER (page 8) " ถ้ามีขนาดน้อยกว่า 600H จะเป็นแพ็กเก็ตของ 802.3 และถ้ามีขนาดมากกว่า 600H จะถือว่าเป็น flag และ Ethernet packet

รหัสผู้ผลิต (Vendor Codes)

อีเทอร์เนตฮาร์ดแวร์แอดเดรส หรือ แมคแอดเดรส เป็นตัวเลขขนาด 48 บิต หรือใช้เลขฐาน 16 (ตัวเลข 0-9 และ A-F) แสดงขนาด 12 หลัก ตัวเลขชุดแรกทางซ้าย 6 หลักจะเป็นหมายเลขของผู้ผลิต ส่วนที่เหลืออีก 6 หลัก จะเป็นซีเรียลนัมเบอร์ (serial number) ของอุปกรณ์เชื่อมต่อระบบเครือข่ายนั้น

อีเทอร์เนตแอดเดรสมักจะเขียนในรูปของเลขฐาน 16 ขนาด 2 ตำแหน่ง 6 ชุดเพื่อให้สามารถอ้างอิงได้ง่าย เช่น 00-40-05-29-F0-F4

ตัวเลขเหล่านี้เป็นฟิสิคัลสเตชั่นแอดเดรส ที่ไม่ใช้มัลติคาสท์หรือบรอดคาสท์

cisco = 00000C

Fujtsu = 00000E

NeXT = 00000F

Novell = 00001B

ATT = 00003D

Nokia = 00004B

NEC = 00004C

ATT = 000055

MIPS = 00006B

Sanyo = 0000A0 /* Sanyo Electronics */

Xerox = 0000AA

Apollo = 0000AC /* Apollo */

HP ON = 0000C6 /* H-P Intlgnt Networks Oper (EON) */

DEC = 0000F8 /* Digital Equipment Corporation */

IEE802 = 000143 /* IEEE 802 */

3com = 0020AF

3com = 00608C

3Com = 00608C

CNET = 0080AD

NET = 0080B2

IEEE = 0080C2 /* IEEE 802.1 Committee */

Intel = 00AA00

3Com = 026060

3Com = 02608C

Bridge = 080002

Apple = 080007

HP = 080009

Apollo = 08001E

Sharp = 08001F /* Sharp */

Sun = 080020

NBI = 080022

FujiXe = 080037 /* Fuji Xerox */

Motrla = 08003E /* Motorola */

Sony = 080046

IBM = 08005A /* (bit-reversed from Token-Ring) */

ATT = 08006A

Mitsu = 080070 /* Mitsubishi */

Casio = 080074 /* Casio */

SilicG = 080079 /* Silicon Graphics */

Xyplex = 080087

ATT = 09006A /* AT&T Use in smart hub */

IBM = 10005A /* (not bit-reversed from Token-Ring) */

DECnet = 1000D4 /* DEC */

ApplUX = 1000E0 /* Apple A/UX (modified addresses for licensing) */

ATT = 800010 /* AT&T */

DECnet = AA0000

DECnet = AA0003

บรอดคาสท์แอดเดรสและสเตชั่นแอดเดรส (Broadcast Addresses & station Address)

address	type	owner
FF-FF-FF-FF-FF-FF	0600	XNS packets, Hello or gateway search?
		6 packets every 15 seconds, per XNS station
FF-FF-FF-FF-FF-FF	0800	IP (e.g. RWHOD via UDP) as needed
FF-FF-FF-FF-FF-FF	0806	ARP (for IP and CHAOS) as needed
FF-FF-FF-FF-FF-FF	1600	VALID packets, Hello or gateway search?
		1 packet every 30 seconds, per VALID station
FF-FF-FF-FF-FF-FF	8035	Reverse ARP
FF-FF-FF-FF-FF-FF	809B	EtherTalk

station 09001E000000 "Apollo_DOMAIN"

station 090007FFFFFF "Atalk_Broadcast"

station 0180C2000000 "Bridge_Group_Addr"

station FFFFFFFFFFFF "Broadcast"

station 09002B230000 "DEC_Argo_Console"

station 09002B010000 "DEC_Bridges"

station AB0000020000 "DEC_Console"

station 09002B000006 "DEC_Encryption"

station AB0000040000 "DEC_END_nodes"

station 09002B040000 "DEC_LAST"

station AB0003000000 "DEC_LAT"

station 09002B00000F "DEC_LAT_Units"

station 09002B010001 "DEC_lv1_Bridges"

station AB0000030000 "DEC_lv1_Router"

station 09002B020000 "DEC_lv2_Router"

station 09002B000000 "DEC_Mumps"

station 09002B020100 "DEC_Name_Advert"

station 09002B020101 "DEC_Name_Solicit"

station 09002B000007 "DEC_Netbios"

station AB0000010000 "DEC_Pmp/Load"

station 09002B000003 "DEC_Traffic_Mon"

station 09002B000002 "DEC_VAXELN"

station C00000000100 "Ethernet Broadcast"

station 090009000004 "HP_DLC"

station 090009000001 "HP_Probe"

station 09002B000004 "ISO_END_Stns"

station 09002B000005 "ISO_Int_Stns"

station CF0000000000 "Loopback"

station 090014000101 "NCP_30_Servers"

station 030000000001 "NetBIOS"

station 090002040002 "Vtlink_Bridges"

station 09007C020005	"Vtlink_Diag"
station 09007C010001	"Vtlink_DLS"
station 09007C010004	"Vtlink_DLS/NonDLS"
station 09007C010002	"Vtlink_DLS_Hello"
station 09007C010003	"Vtlink_DLS_Inlink"
station 090002040001	"Vtlink_Printers"
station 09007C050002	"Vtlink_Validation"
station C00000000001	"Active Mon."
station C00000000100	"All Bridges"
station FFFFFFFFFFFF	"All Fs Broadcast"
station 800143000000	"Bridge Group"
station C000FFFFFFFF	"Broadcast"
station C00000000010	"Config Srv"
station C00000000008	"Error Mon."
station C00000002000	"LAN Manager"
station C00000000080	"NetBIOS"
station C00000800000	"NetWare"
station C00000000002	"Param Server"

อินเตอร์เนตโพรโตคอลนัมเบอร์

Dec	Keyword	Protocol	References
0	Reserved		[JBP]
1	ICMP	Internet Control Message	[RFC792,JBP]
2	IGMP	Internet Group Management	[RFC1112,JBP]
3	GGP	Gateway-to-Gateway	[RFC823,MB]
4	IP	IP in IP (encapsulation)	[JBP]
6	TCP	Transmission Control	[RFC793,JBP]
11	NVP-II	Network Voice Protocol	[RFC741,SC3]

17	UDP	User Datagram	[RFC768,JBP]
29	ISO-TP4	ISO Transport Protocol Class 4	[RFC905,RC77]
41	SIP	Simple Internet Protocol	[SXD]
55-60		Unassigned	[JBP]
61		any host internal protocol	[JBP]
62	CFTP	CFTP	[CFTP,HCF2]
63		any local network	[JBP]
68		any distributed file system	[JBP]
89	OSPFIGP	OSPFIGP	[RFC1583,JTM4]
91	LARP	Locus Address Resolution Protocol	[BXH]
92	MTP	Multicast Transport Protocol	[SXA]
94	IPIP	IP-within-IP Encapsulation Protocol	[JI6]
97	ETHERIP	Ethernet-within-IP Encapsulation	[RXH1]
98	ENCAP	Encapsulation Header	[RFC1241,RXB3]
99		any private encryption scheme	[JBP]
101-254		Unassigned	[JBP]
255		Reserved	[JBP]

คลาส และไทพ์ต่างๆในแพ็กเก็ตไดร์เวอร์

DEC/Intel/Xerox "Bluebook" Ethernet Class 1

3COM 3C500/3C501	1
3COM 3C505	2
Interlan Ni5010	3
BICC Data Networks 4110	4
BICC Data Networks 4117	5
MICOM-Interlan NP600	6
Ungermann-Bass PC-NIC	8
Univation NC-516	9

TRW PC-2000	10
Interlan Ni5210	11
3COM 3C503	12
3COM 3C523	13
Western Digital WD8003	14
Spider Systems S4	15
Torus Frame Level	16
10NET Communications	17
Gateway PC-bus	18
Gateway AT-bus	19
Gateway MCA-bus	20
IMC PCnic	21
IMC PCnic II	22
IMC PCnic 8bit	23
Tigan Communications	24
Micromatic Research	25
Clarkson "Multiplexor"	26
D-Link 8-bit	27
D-Link 16-bit	28
D-Link PS/2	29
Research Machines 8	30
Research Machines 16	31
Research Machines MCA	32
Radix Microsys. EXM1 16-bit	33
Interlan Ni9210	34
Interlan Ni6510	35
Vestra LANMASTER 16-bit	36
Vestra LANMASTER 8-bit	37
Allied Telesis PC/XT/AT	38
Allied Telesis NEC PC-98	39

Allied Telesis Fujitsu FMR	40
Ungermann-Bass NIC/PS2	41
Tiara LANCard/E AT	42
Tiara LANCard/E MC	43
Tiara LANCard/E TP	44
Spider Comm. SpiderComm8	45
Spider Comm. SpiderComm16	46
AT&T Starlan NAU	47
AT&T Starlan-10 NAU	48
AT&T Ethernet NAU	49
Intel smart card	50

ProNET-10 Class 2

Proteon p1300	1
Proteon p1800	2

IEEE 802.5/ProNET-4 Class 3

IBM Token ring adapter	1
Proteon p1340	2
Proteon p1344	3
Gateway PC-bus	4
Gateway AT-bus	5
Gateway MCA-bus	6

Omninet Class 4

Appletalk Class 5

Serial line Class 6

Clarkson 8250-SLIP 1

Clarkson "Multiplexor" 2

Starlan Class 7

ArcNet Class 8

Datapoint RIM 1

AX.25 Class 9

KISS Class 10

IEEE 802.3 w/802.2 hdrs Class 11

FDDI w/802.2 hdrs Class 12

Internet X.25 Class 13

Western Digital 1

Frontier Technology 2

N.T. LANSTAR (encapsulating DIX) Class 14

NT LANSTAR/8 1

NT LANSTAR/MC 2

ประวัติผู้เขียน

นายนภัทร สระเอี่ยม เกิดเมื่อวันที่ 15 มกราคม 2511 ที่จังหวัดกรุงเทพฯ สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิตจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2536 เข้ารับราชการในตำแหน่งอาจารย์ระดับ 3 สังกัดทบวงมหาวิทยาลัย ปัจจุบันดำรงตำแหน่งอาจารย์ระดับ 4 สังกัดสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง