เทคนิคการตรวจจับโปรโตคอลในเครือข่ายท้องถิ่น

PROTOCOLS DETECTION TECHNIQUE FOR LAN

นายวิเชียร เกียรติ์ขจิตมั่น

MR. VICHIAN KIATKAJITMUN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2539

ISBN 974-621-602-3

PROTOCOLS DETECTION TECHNIQUE FOR LAN

VICHIAN KIATKAJITMUN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1996

ISBN 974-621-602-3

หัวข้อวิทยานิพนธ์	เทคนิคการตรวจจับโปรโตคอลในเครือข่ายท้องถิ่น
นักศึกษา	นายวิเชียร เกียรติ์ขจิตมั่น
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.กอบชัย เดชหาญ
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2539

บทคัดย่อ

ในระบบเครือข่ายและการสื่อสารข้อมูลนั้น จะมีโปรโตคอล หลายๆ ชนิดที่ใช้งานอยู่ภายในเครือข่าย โดยแต่ละโปรโตคอลจะมีรูปแบบและเทคนิคที่แตกต่างกันออกไป เมื่อมีปัญหาเกิดขึ้นภายในโครงข่าย ก็จำเป็นต้องมีเครื่องมือหรืออุปกรณ์ช่วยตรวจสอบเครือข่าย หรือตรวจสอบแพกเก็ต ที่อยู่ภายในเครือข่ายนั้น วิทยานิพนธ์ฉบับนี้ได้ทำการวิจัยเกี่ยวกับเทคนิคการตรวจจับแพกเก็ต ที่อยู่ภายในเครือข่าย และสามารถวิเคราะห์ถึงแพกเก็ต ว่าเป็นโปรโตคอลชนิดใด โดยใช้วิธีการเขียนโปรแกรม ติดต่อไปยังส่วนของแพกเก็ต Driver ซึ่งเป็น Driver ที่เป็นมาตรฐานของ Ethernet Card โดยผลที่ได้นั้นสามารถนำไปประกอบการวิเคราะห์ถึงปัญหาที่เกิดขึ้นในเครือข่ายได้

Thesis Title	Protocols Detection Technique for LAN
Student	Mr. Vichian Kiatkajitmun
Thesis Advisor	Assoc. Prof. Dr. Kobchai Dejhan
Level of Study	Master of Engineering in Engineering
Department	Telecommunications Engineering
	King Mongkut' s Institute of Technology Ladkrabang
Year	1996

Abstract

The networking systems have many kind of protocols. Each protocol has a different platform and technical system. As the network has a problem, it needs some tools such as protocol analyzer to analyse the packet. This thesis concerned about the packet detection technique in the network and analyses the detials of packet, what is the kind of packet protocol. The method is to write the program to access into packet driver by using the Ethernet card standard driver. The result of this experiment can be use to support analysing of the network problems.

กิติกรรมประกาศ

ในการทำวิทยานิพนธ์ฉบับนี้ได้มีอุปสรรคต่าง ๆ เกิดขึ้นมากบ้างน้อยบาง ซึ่งอุปสรรคเหล่านั้นก็สามารถถูกแก้ไข ให้สำเร็จลุล่วงไปได้ดีด้วยกำลังใจ และคำแนะนำที่ดี ซึ่งวิทยานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงไปได้เลยถ้าไม่มีบุคคลต่อไปนี้

ขอขอบพระคุณ รองศาสตราจารย์ ดร. กอบชัย เดชหาญ ที่ช่วยให้คำแนะนำ และช่วยเหลือทุก ๆ อย่างในการทำวิทยานิพนธ์

ขอขอบพระคุณ รองศาสตราจารย์ ดร. ฟูศักดิ์ ชีวสุวิทย์ และท่านอาจารย์ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาทั้งหมดให้แก่ผู้เขียน

ขอขอบคุณ คุณกิติชัย อัศวชำนาญกิจ , คุณราตรี กันหา , คุณพรทิพย์ บุษราคัมวิเศษ และคุณฐิติวรรณ กาญจนาภา ที่กรุณาช่วยในการจัดพิมพ์วิทยานิพนธ์

สุดท้ายนี้ขอขอบพระคุณ คุณพ่อ คุณแม่ พี่ๆ และเพื่อนๆ ที่ให้กำลังใจในการทำวิจัยเสมอมา จนวิทยานิพนธ์สำเร็จลงอย่างสมบูรณ์

วิเชียร เกียรติ์ขจิตมั่น

สารบัญ

หน้า

บทที่ หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญภาพ

หน้า

สารบัญภาพ(ต่อ)

หน้า

สารบัญภาพ(ต่อ)

หน้า

บทที่ 1

บทนำ

แนวความคิดของวิทยานิพนธ์

ในปัจจุบันการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์ได้รับความนิยมอย่างแพร่หลาย ซึ่งจะมีการใช้โปรโตคอล (Protocol) หลากหลายชนิดใช้ในการติดต่อสื่อสารกัน ดังนั้นในระบบเครือข่ายเองเมื่อเกิดปัญหาเกี่ยวกับโปรโตคอล ต่าง ๆ ที่ทำการติดต่อสื่อสารก็จะทำให้ผู้ดูแลระบบเองต้องทำการวิเคราะห์และทำการแก้ไขปัญหานั้น ๆ ได้ยาก และจำเป็นต้องอาศัยเครื่องมือหรืออุปกรณ์ที่ใช้ในการวิเคราะห์ปัญหาที่เกิดขึ้น ดังนั้นจึงเกิดแนวความคิดของวิทยานิพนธ์ฉบับนี้ ที่ได้ทำการวิจัยเกี่ยวกับวิธีการตรวจสอบแพกเก็ต (Packet) ของโปรโตคอล ที่เกิดขึ้นในเครือข่าย เพื่อเป็นเครื่องมือช่วยสำหรับให้ผู้ดูแลระบบเครือข่ายใช้ในการวิเคราะห์หรือแก้ไขปัญหาที่เกิดกับระบบเครือข่ายได้

วัตถุประสงค์ในการทำวิทยานิพนธ์

ในการทำวิทยานิพนธ์เรื่อง "เทคนิคการตรวจจับโปรโตคอลในเครือข่ายท้องถิ่น" (Protocols Detection Technique for LAN) ได้รวบรวมถึงความเป็นมา และรูปแบบของแพกเก็ต ของโปรโตคอล ชนิดต่าง ๆ ที่นิยมใช้งานกันอยู่ในปัจจุบัน และในวิทยานิพนธ์นี้ได้กำหนดจุดประสงค์ไว้ประการดังนี้

-เพื่อศึกษารูปแบบของ แพกเก็ต ข้อมูลบน IEEE 802.3

-เพื่อศึกษาถึงรูปแบบของ แพกเก็ต ของโปรโตคอล ต่าง ๆ ที่เป็นที่นิยมใช้งาน

-สามารถนำซอฟแวร์ที่พัฒนาขึ้นมา นำมาใช้ในการวิเคราะห์หรือแก้ไขปัญหาได้

-เพื่อเป็นแนวทางในการทำงานวิจัยในอุปกรณ์ในเครือข่าย เช่น Bridge หรือ Router

-เพื่อศึกษาค้นคว้าการเขียนซอฟแวร์เพื่อติดต่อกับ แพกเก็ตไดรเวอร์ (Packet Driver) ซึ่งเป็นไดรเวอร์ มาตรฐานชนิดหนึ่งที่นิยมใช้งาน

รายละเอียดในวิทยานิพนธ์

ในวิทยานิพนธ์ ได้แบ่งเนื้อหาออกเป็นบทได้ทั้งหมด 8 บท โดยในบทที่ 1 จะเป็นการกล่าวนำถึง แนวความคิดและวัตถุประสงค์ในการทำวิทยานิพนธ์ และได้กล่าวถึงเนื้อหาโดยย่อของแต่ละบท ซึ่งในบทอื่น ๆ จะมีเนื้อหาดังนี้

บทที่ 2 ได้กล่าวถึงทฤษฎีหลักการ และโครงสร้างของ Ethernet

บทที่ 3 กล่าวถึง Internet Protocol ซึ่งเป็นโปรโตคอล มาตรฐานชนิดหนึ่ง โดยได้แสดงถึงรูปแบบของโปรโตคอลต่าง ๆ ที่อยู่ใน Internet Protocol

บทที่ 4 กล่าวถึงกลุ่มของโปรโตคอล ที่นิยมใช้งานโดยทั่ว ๆ ไป โดยได้บอกถึงลักษณะความเป็นมา และโครงสร้างของโปรโตคอล

บทที่ 5 กล่าวถึงกลุ่มของ Routing โปรโตคอล ที่ใช้งานในเครือข่าย โดยได้บอกถึงลักษณะและโครงสร้างของ Routing โปรโตคอล

บทที่ 6 กล่าวถึงหลักการและวิธีการในการเขียนโปรแกรม ติดต่อกับ แพกเก็ตไดรเวอร์ เพื่อให้สามารถทำการตรวจจับ แพกเก็ต ในเครือข่ายได้

บทที่ 7 กล่าวถึงวิธีการทำการทดลองจากเครือข่ายที่ใช้งานอยู่จริง และสรุปปัญหาที่เกิดขึ้น

บทที่ 8 กล่าวถึงการนำซอฟแวร์ไปประยุกต์ใช้งานในงานด้านอื่น ๆ เช่น Bridge หรือ Router

ภาคผนวก ได้แสดงถึงรูปแบบชนิด (Type) ของ โปรโตคอล ชนิดต่างๆ ที่มีอยู่ และส่วนสุดท้ายเป็นประวัติผู้เขียน และผลงานวิจัยที่ได้รับการตีพิมพ์ในวารสารวิชาการ

บทที่ 2

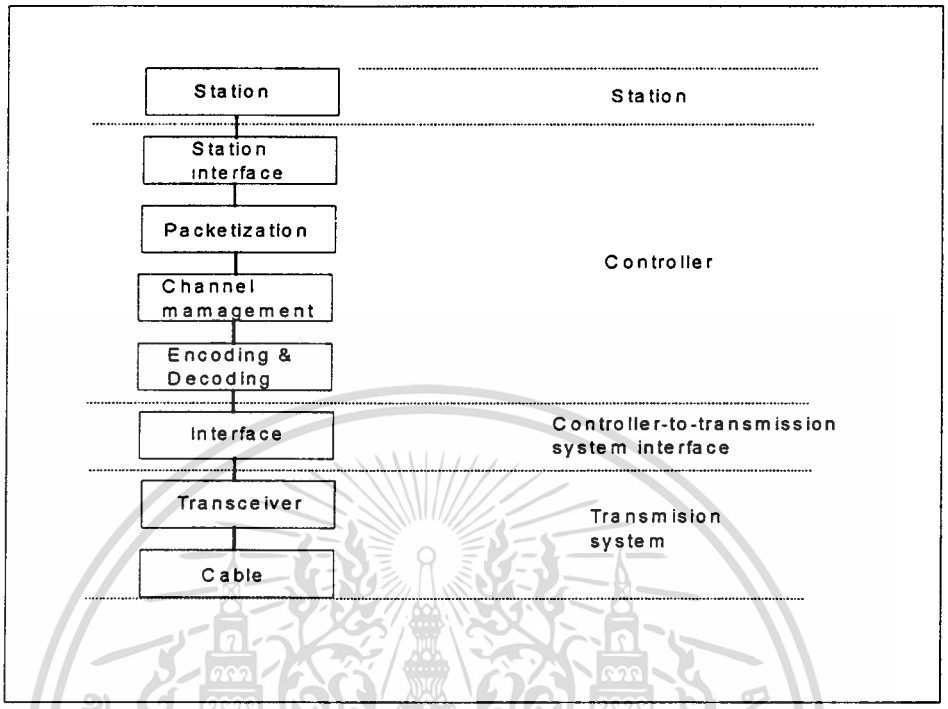
Ethernet และ IEEE 802.3

หลักการและโครงสร้างของ Ethernet

ในช่วงเริ่มแรกที่มีการผลิตสินค้าเกี่ยวกับระบบเครือข่ายท้องถิ่น ซึ่งในการผลิตไม่ได้มีรูปแบบมาตรฐาน ที่เป็นสากล ต่อมาผู้ผลิตทั้งหลายจึงได้รวมตัวกันขึ้น เพื่อทำการกำหนดมาตรฐานของ Ethernet ขึ้นเป็น IEEE Standards ซึ่งมาตรฐานของ Ethernet ได้รวมเอา Data Link Layer และ Physical Layer ของ ISO Model กับ Medium access และ Physical layer ของ IEEE 802 Model (Random access network) รูปแบบของ Ethernet เองได้ถูกพัฒนามาจากบริษัท XEROX ในช่วงต้นปี 1970 เป็นแห่งแรก โดยกำหนดรายละเอียดไว้คือ สามารถที่เชื่อมต่อกับสเตชั่น (Station) ได้เกิน 100 สเตชั่น โดยใช้กับสายสัญญาณ Coaxial ที่ความยาว 1 กิโลเมตร มี อัตราความเร็วในการส่งข้อมูลเท่ากับ 2.94 เม็กกะบิทต่อวินาที และต่อมาได้มีการพัฒนาเพิ่มเติมโดย บริษัท XEROX Intel และ Digital Equipment Coporation (DEC) กำหนดเป็นมาตรฐาน Ethernet โดยที่มาตรฐานที่ถูกกำหนดขึ้นมานี้ไม่เพียงแต่ใช้ในกลุ่มบริษัทพวกนี้เท่านั้นยังถูกกำหนดให้เป็นมาตรฐานของ IEEE Local area network standard

รูปแบบของ Ethernet นั้นได้กำหนดมาตรฐานของความยาวของ เส้นทางเดินของข้อมูล (Bus) ไว้สูงสุดที่ 2.5 กิโลเมตร สามารถเชื่อมต่อภายในเซกเมนต์ (Segment) เดียวกันได้ถึง 500 เมตร ด้วยอัตราการส่งข้อมูล 10 เมกกะบิทต่อวินาที และสามารถเชื่อมต่อสถานี (Station) ได้ถึง 1024 สถานี ในทฤษฎีของ CSMA/CD access สามารถที่ใช้การส่งกระจายการสื่อสารได้หลายชนิด ได้รวมถึง Radio, Twisted-pair, Coaxial cable หรือแม้กระทั่ง Fiber Optic ได้ อย่างไรก็ตามแล้ว Ethernet ได้ถูกออกแบบมาสำหรับ Baseband Transmission โดยใช้สายสัญญาณ Coaxial ซึ่งในภาพที่ 1 ได้แสดงถึง รูปแบบและการออกแบบ Ethernet ประกอบด้วยส่วนประกอบ 4 ส่วนที่สำคัญบน Ethernet ซึ่งประกอบด้วยสถานี (Station), ส่วนควบคุม (Controller), ส่วนรับส่ง (Transmission System) และส่วนที่ติดต่อระหว่างส่วนควบคุมกับส่วนรับส่ง (The Controller to-system interface) ซึ่งส่วนต่าง ๆ จะมีหน้าที่แตกต่างกันดังนี้

ภาพที่ 1



แสดงส่วนประกอบของ Ethernet สำหรับ 1 โหนด

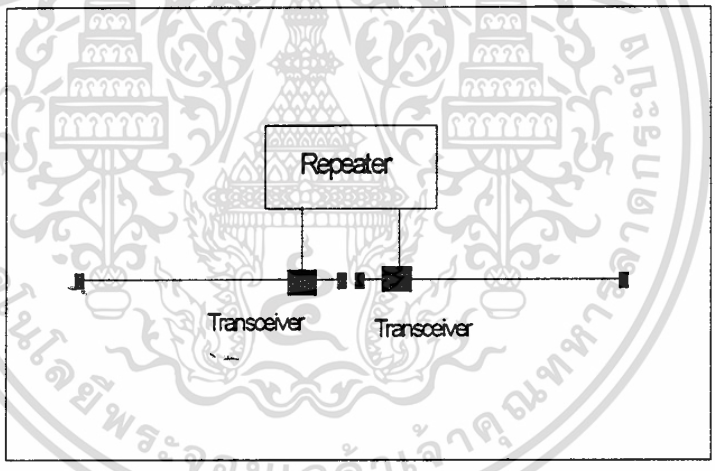
- สถานี จะเป็นชนิดของคอมพิวเตอร์ (Computer) หรือเป็นกลุ่มของเทอร์มินอล (Terminal)

- ส่วนควบคุม จะเป็นกลุ่มของฟังก์ชัน (Function) และขั้นตอนวิธี (Algorithms) ที่ต้องการ ซึ่งจะจัดการกับเครือข่าย (Network) ซึ่งจะรวมถึงการเข้ารหัส (Encoding) และการถอดรหัส (Decoding),การเปลี่ยนแบบอนุกรมไปเป็นแบบขนาน (Serial to Parallel Conversion), การจดจำแอดเดรส (Address Recognition), Signaling Conventions, Packetization, CMS/CD Channel Management, การรับรู้ความผิดพลาด (Error Detection) และการจัดเก็บ (Buffering) หน้าที่ของส่วนควบคุม จะรวมถึงฮาร์ดแวร์ (Hardware), ซอฟแวร์ (Software) และ Microcode ซึ่งขึ้นอยู่กับสถานีนั้น ๆ โดยส่วนมากแล้ว Ethernet Controllers จะถูกออกแบบมาบนชิปตัวเดียว

- ส่วนระบบรับส่ง (Transmission System) จะรวมไปถึงส่วนประกอบที่สำคัญของสำหรับ การติดต่อสื่อสารระหว่าง ส่วนควบคุม และรวมไปถึงตัวกลางในการรับส่ง (Transmission Medium) ก็คือ อุปกรณ์ส่งผ่านข้อมูล (Transceivers) และ อุปกรณ์ทวน

สัญญาณ (Repeaters) ซึ่งเป็นส่วนที่สำคัญในการขยายระยะการติดต่อสื่อสารออกไป จะใช้สายสัญญาณ Coaxial เป็นตัวกลาง ประกอบไปด้วยสายสัญญาณ (Cable) และส่วนประกอบของฮาร์ดแวร์ ที่สำคัญ เช่น Connectors, Terminators และ Taps ซึ่ง ตัว Terminator จะเป็นตัวป้องกันสัญญาณที่ส่งออกไปไม่ให้กลับมายัง bus อีก โดยใช้วิธีการ Matching Impedance ของสายสัญญาณ ตัว อุปกรณ์ส่งผ่านข้อมูล (Transceivers) จะทำหน้าที่ส่ง และรับ สัญญาณไปยังสายสัญญาณ และยังต้องรับรู้ถึงหน้าที่ของ CSMA/CD ด้วย ซึ่งตัว Transceiver จะต้องคอยตรวจจับสัญญาณบนสายสัญญาณ ก่อนที่จะเริ่มส่งสัญญาณออกไป และในขณะที่ส่งเอง ก็ต้องคอยตรวจจับสัญญาณอื่น ๆ บนสายสัญญาณ ด้วย ตัว อุปกรณ์ทวนสัญญาณ (Repeaters) จะประกอบไปด้วย อุปกรณ์ส่งผ่านข้อมูล (Transceivers) 2 ตัว ซึ่งใช้เชื่อต่อเข้ากับ Ethernet เซกเมนต์ ดังภาพที่ 2

ภาพที่ 2



แสดงการเชื่อมต่อ อุปกรณ์ทวนสัญญาณ (Repeaters)

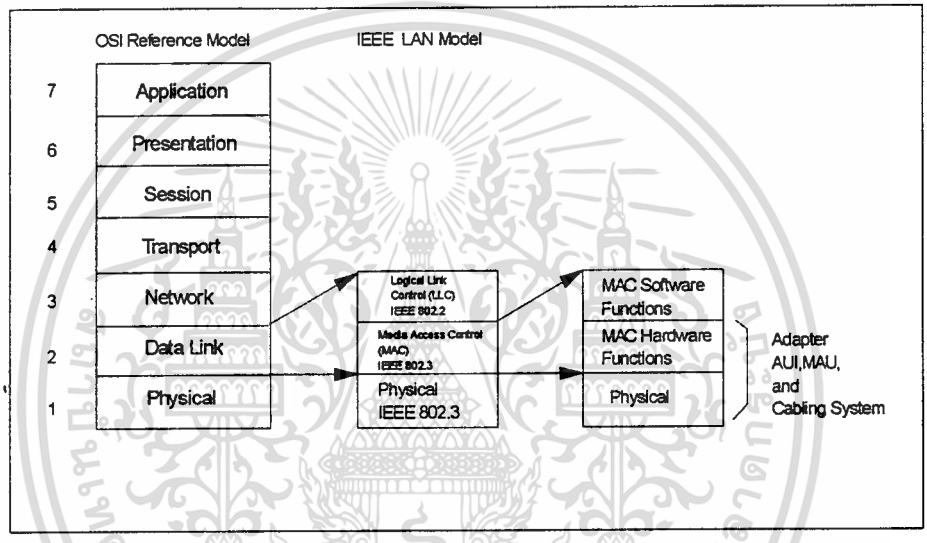
อุปกรณ์ทวนสัญญาณ (Repeaters) จะมีหน้าที่เพียงการส่งผ่านสัญญาณไปยัง Ethernet เซกเมนต์ เท่านั้น โดยจะไม่มีหน้าที่ในส่วนของการรับรู้ CSMA/CD

- Controller to transmission system interface ก็จะเป็นสายสัญญาณ ที่ใช้สำหรับเชื่อมต่อ อุปกรณ์ส่งผ่านข้อมูล (Transceivers) เข้ากับส่วน Controller

OSI Reference Model

ในภาพที่ 3 จะเป็นมาตรฐานของ IEEE 802.3 ที่ปรากฏอยู่ใน OSI Model IEEE 802.3 นั้นจะรวมเอา Physical Layer และบางส่วนใน Data Link layer ใน OSI Model ซึ่งเรียกว่า Medium Access Control (MAC) Layer

ภาพที่ 3



แสดง IEEE LAN Model สัมพันธ์ใน OSI Model

ทฤษฎีและการปฏิบัติ

Ethernet โดยทั่วไปแล้วจะมีลักษณะการใช้งานโดยเป็นการจัดการช่วงการสื่อสารร่วมกันโดยกระจายการควบคุม ซึ่งเรียกว่า CSMA/CD (Carrier Sense Multiple access with collision detection) โดยวิธีนี้เองจะไม่มีส่วนจัดการควบคุมส่วนกลางไปยังช่องสัญญาณ (Channel) ต่าง ๆ และจะไม่มีการแบ่งช่องของ Time Slot หรือ Frequency Bands เมื่อ สถานีต้องการที่จะส่งข้อมูล มันก็จะร้องขอสิทธิช่วงชิง ในการส่งข้อมูลไปยังส่วนที่ทำหน้าที่กระจาย

การสื่อสาร (Shared Communication Channel) เมื่อได้รับสิทธิช่วงชิงจากช่องการสื่อสารแล้ว สถานีก็สามารถที่จะส่งข้อมูลแพกเก็ต (Packet) ออกไปได้

ในการที่จะได้ช่องการสื่อสารนั้น สถานีจะคอยตรวจสอบเมื่อในเครือข่าย (Network) ยังไม่ว่าง คือมีการรับส่งข้อมูลอยู่ (Carrier Sense) และมันจะรอการส่ง จนกระทั่งช่องการสื่อสารนั้นว่าง คือไม่มีการสื่อสารเกิดขึ้นเมื่อสถานี สามารถที่ตรวจจับได้ว่าช่องสัญญาณ (Channel) นั้นว่าง มันก็จะทำการเริ่มส่งข้อมูลได้ และในขณะที่กำลังส่งข้อมูลอยู่นั้น มันเองก็ยังต้องคอยฟังหรือตรวจสอบว่ามีสัญญาณ Collision การชนกันของข้อมูลหรือไม่ สัญญาณ Collision ที่เกิดขึ้นนั้น จะเกิดขึ้นในช่วงเวลาสั้น ๆในการเริ่มส่งข้อมูลเท่านั้น ถ้าไม่มีสัญญาณ Collision เกิดขึ้น ผู้ส่งก็สามารถได้รับช่องสัญญาณ และสามารถส่งแพกเก็ต ออกไปได้ ถ้าในกรณีที่สถานีตรวจพบการชนกันของข้อมูลเกิดขึ้น (Collision) การส่งข้อมูลก็ถูกสั่งให้หยุดทำการส่งข้อมูลแพกเก็ต โดยทันที เพื่อที่จะให้สถานีต่าง ๆ ที่เกี่ยวข้องรับรู้การเกิดการชนกันของข้อมูลขึ้น ซึ่งก็จะรู้ทันทีว่าเกิดการติดขัด (สัญญาณ Jam) ของช่องสัญญาณขึ้น ตัวของสถานีก็จะรับรู้ถึงการการเกิดการชนกันของข้อมูลขึ้น คือจะต้องทำการส่งกลับไปใหม่ (Retransmission) โดยช่วงเวลาของการ ส่งกลับไปใหม่นั้น จะใช้เป็นช่วงเวลาของการสุ่ม (Random delay period)

มาตรฐานของ Ethernet

ข้อรวบรวมคุณลักษณะของ Ethernet

- รูปร่างระบบเครือข่าย (Topology) = BUS
- ใช้วิธีการ CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
- อัตราการรับส่งข้อมูล 10 เมกกะบิทต่อวินาที
- จำนวนสถานีที่มากที่สุดต่อเครือข่าย = 1024 สถานี
- ค่าความต้านทานภายใน ภายในสาย Coaxial = 50 โอห์ม $\pm$ 20 โอห์ม
- ระดับของสัญญาณในสาย Coaxial = 0 ถึง -2.05 โวลท์
- Thick Coaxial Cable (10 Base 5)

  ความยาวสูงสุดของสายสัญญาณในเซกเมนต์ 500 เมตร (1640 ฟุต)

  จำนวนสูงสุดของ Transceiver ในเซกเมนต์ 100 ตัว

  ช่วงระยะห่างต่ำสุดของ Transceiver 2.5 เมตร

- Thin Coaxial Cable (10 Base 2)

ความยาวสูงสุดของสายสัญญาณในเซกเมนต์ 185 เมตร (607 ฟุต)

จำนวนสูงสุดของ Transceiver ในเซกเมนต์ 30 ตัว

ช่วงระยะห่างต่ำสุดระหว่าง Transceiver 0.5 เมตร

- ชนิดของสายสัญญาณ

Thick Cable ชนิด 50 โอห์ม เป็น Ethernet Cable

Thin Cable ชนิด 50 โอห์ม RG 58 C/U หรือ RG 58 A/U

- Terminator

Thick cable ใช้ N-Series Terminator

Thin Cable ใช้ BNC Terminator

- ความยาวสูงสุดของสายสัญญาณ ที่ต่อเข้ากับ Transceiver 50 เมตร

เป็นสายชนิด 20 AWG Shielded Twisted-pair(4 pairs), 78 โอห์ม

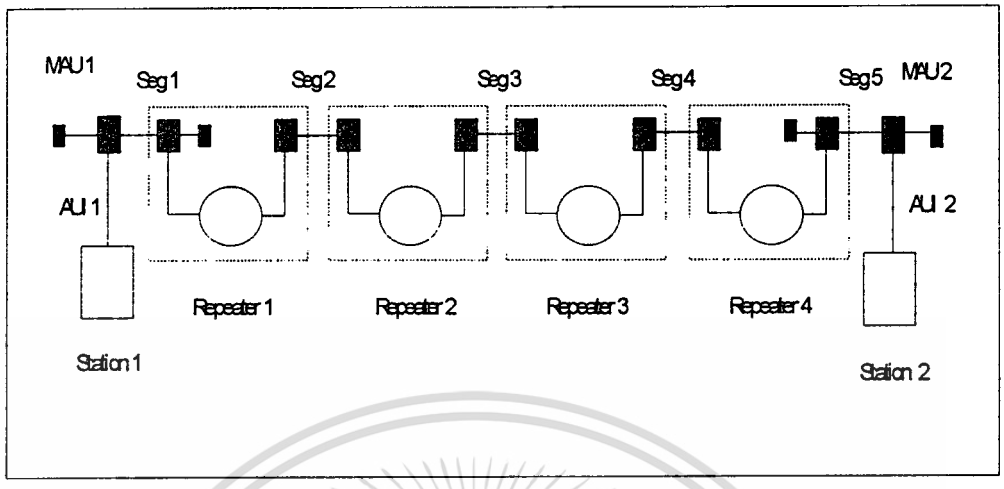
ข้อบังคับของ Ethernet

- ช่วงเวลาหน่วงเฉพาะของ end-to-end ที่ข้ามไปยังทุก ๆ เซกเมนต์ ต้องไม่เกิน 51.2 μs ความแตกต่างของตัวกลางนำพา (Media) ก็จะให้ช่วงเวลา ความเร็วเฉพาะที่แตกต่างกัน ซึ่งขึ้นอยู่กับ ระยะทาง และตัว Repeaters เองก็ตามก็รวมเอาช่วงเวลาหน่วงเฉพาะของแต่ละเซกเมนต์ ไว้ด้วย ซึ่งแต่ละเซกเมนต์ ก็จะมีค่าหน่วงของตัวเองขึ้นอยู่กับ DC Loop Resistance โดยค่า Thick Coaxial จะมีความยาวสูงสุดอยู่ที่ 500 เมตร และ Thin Coaxial มีความยาวสูงสุดอยู่ที่ 185 เมตร

- ภาพที่ 4 แสดงให้เห็นถึง ระยะที่มากที่สุดในส่งผ่านข้อมูลระหว่าง 2 สถานี ประกอบด้วย 5 เซกเมนต์ , 4 อุปกรณ์ทวนสัญญาณ (Repeaters), 2 MAUs (Multistation Access Unit เป็นอุปกรณ์ที่ใช้เป็นศูนย์รวมของสายสัญญาณซึ่งใช้เชื่อมต่อสายสัญญาณ กับสถานีต่าง ๆ ในระบบเครือข่าย) และ 2 AUI (Attachment Unit Interface เป็นสายสัญญาณที่ใช้เชื่อมต่อระหว่าง MAU กับ สถานีเครือข่าย)

- ใน Thick Coaxial Cable จำนวนความยาวมากที่สุดในการเชื่อมต่อระหว่าง เซกเมนต์ ต่าง ๆ ของ 2 MAUs เท่ากับ 2500 เมตร (5 เซกเมนต์ และเซกเมนต์ละ 500 เมตร) ส่วนใน Thin Cable จะมีความยาวมากที่สุดในการเชื่อมต่อในทำนองเดียวกันเท่ากับ 925 เมตร (5 เซกเมนต์ โดยเซกเมนต์ ละ 185 เมตร)

ภาพที่ 4



แสดงเซกเมนต์ สูงสุดของการรับส่งข้อมูล

ข้อมูลทางด้านเทคนิคของ Ethernet

Ethernet เวอร์ชั่น 1.0 สถานีต่าง ๆ ในเครือข่าย เมื่อจะรับหรือส่งข้อมูลแพกเก็ต จะต้องอยู่ในรูปแบบที่แสดงไว้ดังภาพที่ 5

ภาพที่ 5

Preamble	Destination	Source	Type	Data	CRC
8 bytes	6	6	2	46-1500	4

แสดง Ethernet เวอร์ชั่น 1.0

ขนาดสูงสุดของ Packet Size = 1526 ไบท์
(8-ไบท์ preamble + 14-ไบท์ header + 1500 data ไบท์ -4-ไบท์ CRC)
ขนาดน้อยที่สุดของ Packet Size = 72 ไบท์
(8-ไบท์ Preamble + 14-ไบท์ header + 46 data ไบท์ + 4 ไบท์ CRC)

Preamble : จะมีรูปแบบ Synchronization มีขนาดจำนวน64 บิท ซึ่งจะประกอบไปด้วยบิท 1 และบิท 0 สลับกันไป และที่ บิท ท้ายสุด 2 บิท จะเป็น 1 และ Preamble นี้จะใช้เป็นส่วนที่เริ่มต้นของแพกเก็ต

Destination Address : จะมีขนาด 48 บิท ใช้สำหรับสถานีนั้น ๆ ในการที่จะส่งแพกเก็ตไปยังสถานี อื่น ซึ่งแต่ละ สถานี ก็จะทำการตรวจสอบฟิลด์นี้ว่า ควรจะรับแพกเก็ตนี้หรือไม่ บิทแรกที่ส่งจะแสดงถึงชนิดของแอดเดรส ถ้าเป็นค่า 0 นี้จะแสดงถึง Physical address ของ Destination station และถ้าเป็นค่า 1 ก็จะแสดงถึง Logical address เรียกว่า Multicast ID ซึ่งขึ้นอยู่กับกลุ่มของผู้รับ สถานีที่สามารถรับรู้ Multicast ก็สามารถที่จะรับ Multicast Ids ที่ต้องการได้ และจะมี Multicast address ชนิดพิเศษ ซึ่งเราเรียกว่า Broadcast address ซึ่งในฟิลด์ ของ Destination Address จะประกอบไปด้วยค่าบิท 1 ทั้งหมด และ Ethernet Address จะมี ค่า แอดเดรสที่ไม่ซ้ำกัน

Source Address : จะมีขนาด 48-บิท เป็นแอดเดรส ของตัวสถานีที่ส่งเอง และแอดเดรสก็ จะไม่ซ้ำกันกับ Ethernet Address อื่น ๆ

Type Field : มีขนาด 16 บิท ใช้สำหรับแสดงถึงชนิดของโปรโตคอล ของแพกเก็ต เช่น XNS, TCP/IP, IPX, OSI เพื่อสามารถรับรู้ถึงการจัดการใน Data ฟิลด์ ได้ Data ฟิลด์ จะมีขนาดตั้งแต่ 46 บิท ถึง 1500 บิท จะเป็นฟิลด์ สำหรับบรรจุข้อมูลที่ต้องการ

Packet Check Sequence : จะมีขนาด 32 บิท เป็นฟิลด์ ที่ใช้สำหรับ Cyclic redundancy check Code ตรวจสอบข้อผิดพลาดของ แพกเก็ต ซึ่งตัว CRC นี้จะควบคุมถึง Destination และ Source Address, Type และ Data ฟิลด์ ด้วย

Minimum Packet Spacing : เป็นค่าระยะห่างของช่วงเวลาของแพกเก็ต ต่ำสุดในการส่งแพกเก็ต ออกไป มีค่า = 9.6 μs

Round Trip Delay : เป็นค่าเวลาสูงสุดในการส่ง end-to-end = 51.2μs

Collision Filtering : เมื่อได้รับแพกเก็ตแต่ค่าของจำนวน bit Sequence ที่ได้น้อยกว่าค่าต่ำสุดของแพกเก็ต ก็จะทำการยกเลิกแพกเก็ต ที่ได้รับมาทันที หรือเรียกว่า Collision fragment

Control Procedure : เป็นการกำหนดวิธีการควบคุม เมื่อ สถานี ต้องการที่จะส่งแพกเก็ตไปยังสายสัญญาณ ซึ่งเป็นวิธีการในการที่พยายามหาโอกาศช่วงชิงการส่งแพกเก็ต กับสถานีอื่นจำนวนมาก

Defer : เป็นช่วงเวลาหน่วงที่สถานี จะไม่ทำการส่งแพกเก็ต ออกไปในสายสัญญาณในขณะที่มีข้อมูลของสถานี อื่นส่งอยู่ หรือช่วงที่ Minimum Packet Spacing

Transmit : สถานี จะสามารถส่งแพกเก็ต ออกไปได้ขณะที่ไม่อยู่ในช่วงเวลาหน่วง (Deferring) และสามารถที่จะส่งแพกเก็ต ออกไปได้จนสิ้นสุด หรือจนกระทั่งเกิดการชนกันของข้อมูล (Collision)

Abort : เมื่อเกิดการชนกันของข้อมูลขึ้น การส่งแพกเก็ตก็จะต้องสิ้นสุดลง และจะมีการส่ง Jam (ข้อมูลที่มี 4-6 ไบท์) ออกไปเพื่อให้สถานี อื่นที่เกี่ยวข้องได้รับรู้ว่าเกิด เกิดการชนกันของข้อมูลขึ้น

Retransmit : หลังจากที่สถานี ได้ทำการรับรู้การเกิดการชนกันของข้อมูล จะทำการหยุดการส่งข้อมูล แล้วจะทำการรอช่วงจังหวะเวลา เพื่อที่จะพยายามส่งแพกเก็ต ออกไปใหม่ ในกรณีที่มีการส่งกลับ (Retransmit) แล้วจำนวนครั้งที่ทำการส่งกลับ เกินช่วงเวลาที่กำหนดคือ 51.2 μs ก็จะหยุดการส่งแพกเก็ตนั้น และจะรายงานข้อผิดพลาด ไปยังระดับที่อยู่สูงกว่า โดยที่ซอฟแวร์เองจะเป็นตัวตัดสินเองว่าจะทำต่อไป หรือพยายามส่งอีกครั้ง

Channel Encoding : สภาวะ Logic high = 0 V.

สภาวะ Logic Low = -2.05 V.

สภาวะไม่มีการรับส่งข้อมูล = 0 V.

Data Rate : = 10 Mbps.

Collision Detection : เมื่ออุปกรณ์ส่งผ่านข้อมูล (Transceivers) รับรู้ถึงการชนกันของข้อมูล เมื่อ DC Signal มีการเปลี่ยนแปลง ในขณะที่สัญญาณของแพกเก็ต เริ่มทำการส่งไปยังสายสัญญาณนั้นจะมีค่า DC Signal เท่ากับ -1.025 V. (ค่าเฉลี่ยที่ logic high 0 V. และ logic low -2.05 V.) และเมื่อมีแพกเก็ตอื่น ส่งเข้ามาในขณะเวลาเดียวกันก็จะทำให้ค่า DC Signal ในสายสัญญาณมีค่าสูงขึ้นซึ่งจะทำให้อุปกรณ์ส่งผ่านข้อมูล (Transceivers) ทำการรับรู้ได้ว่าเกิดการชนกันของข้อมูลขึ้น และจะรายงานกลับไปยังส่วนควบคุม (Controller)

Cyclic Redundancy Check (CRC) : ข้อมูลภายในของ Data ที่บรรจุไปด้วย Destination address, Source address, Type และ Data ฟิลด์ (ตั้งแต่ 60-1514 ไบท์) จะถูกบีบอัด (Compress) ด้วยวิธีการ CRC Algorithm แล้วใส่ลงไป 4-ไบท์ ฟิลด์ ซึ่งตัว CRC นี้จะถูกใส่ค่าก่อนที่จะทำการส่งแพกเก็ต และเมื่อทางสถานีทางปลายทางทำการรับแพกเก็ต และก็จะทำการแยก CRC ฟิลด์ออกมา แล้วจะทำการคำนวณด้วย CRC algorithm เช่นเดียวกัน เพื่อเปรียบเทียบค่าที่ได้ในกรณี ที่ค่าที่ได้แตกต่างกันก็จะทำการยกเลิกแพกเก็ตนั้น และแสดง CRC error

ข้อแตกต่างทางเวอร์ชั่นของ Ethernet

ข้อสรุประหว่าง Ethernet เวอร์ชั่น 1.0 กับเวอร์ชั่น 2.0 และ IEEE 802.3 standard

Network Architecture : มาตรฐานของ Ethernet ประกอบด้วย Physical Layer และ Datalink Layer หน้าที่ของแต่ละตัวก็จะเหมือนเช่นเดียวกับ ใน IEEE 802.3 คือ Physical และ Medium access control (MAC) Layer ซึ่งบน Ethernet ก็มีส่วนที่ติดต่อกับ Higher layer ซึ่งเรียกว่า Client Layer ตัว Client Layer นี้ก็จะให้ Logical Link ระหว่าง สถานี เช่นเดียวกับ Session connection, error control, และ end to end connection ส่วนของ IEEE 802.3 ก็จะติดต่อกับ Logical link control (LLC)

Half-Step Signaling : Ethernet เวอร์ชั่น 1.0 จะใช้ สัญญาณบน AUI เป็น full-step ส่วนใน เวอร์ชั่น 2.0 และ IEEE 802.3 จะใช้เป็น Half-step ซึ่งสามารถป้องกันสัญญาณรบกวนได้ดีกว่า

Signal Quality Error Test (SQE/Heartbeat) : SQE ถูกกำหนดโดย IEEE 802.3 เพื่อให้ซอฟแวร์ (Software) สามารถที่จะตรวจสอบ การชนกันของข้อมูล ที่เกิดขึ้นบนตัว Transceiver

Jabber Control : เป็นการควบคุมของ ส่วนควบคุม (Controller) ในกรณีที่เกิดปัญหาขึ้น ที่ส่วนควบคุมอาจมีโอกาสที่ส่งข้อมูล ที่ผิดพลาดไปในเครือข่าย โดยที่ตัวอุปกรณ์ส่งผ่านข้อมูล (Transceivers) ก็จะทำการหยุดส่งข้อมูลชั่วคราว ส่วนข้อมูลที่ผิดพลาดนั้นจะอยู่ในช่วงเวลาที่ไม่น้อยกว่า 20 ms และไม่มากกว่า 150 ms และช่วงเวลาหยุดส่งจะอยู่ในช่วง 250 ms และ 750 ms

Packet Format : Ethernet เวอร์ชั่น 1.0 และ 2.0 จะมีรูปแบบของแพกเก็ต ที่เหมือนกับส่วน IEEE 802.3 จะแตกต่างออกไปดังรูป 6

Preamble : IEEE จะแยก Ethernet Preamble มาเป็น 7-ไบท์ Preamble และอีก 1 ไบท์ ใช้สำหรับ Start frame Delimiter (SFD)

Destination Address :

Ethernet 6 ไบท์ ประกอบด้วย P/M บิท+47 บิท แอดเดรส

ค่า P/M บิท = 0 แสดงถึง Physical address

= 1 แสดงถึง Multicast address

IEEE 802.3 = 6 ไบท์ ประกอบด้วย I/G-บิท+U/L+46-บิท แอดเดรส

I/O-บิท = 0 แสดงถึง Individual address

= 1 แสดงถึง Group address

U/L-บิท = 0 แสดงถึง Universally Administered address

= 1 แสดงถึง Locally Administered address

Universal Addresses : เป็นแอดเดรส ที่ไม่ซ้ำกับแอดเดรส อื่น ๆ ในโลก

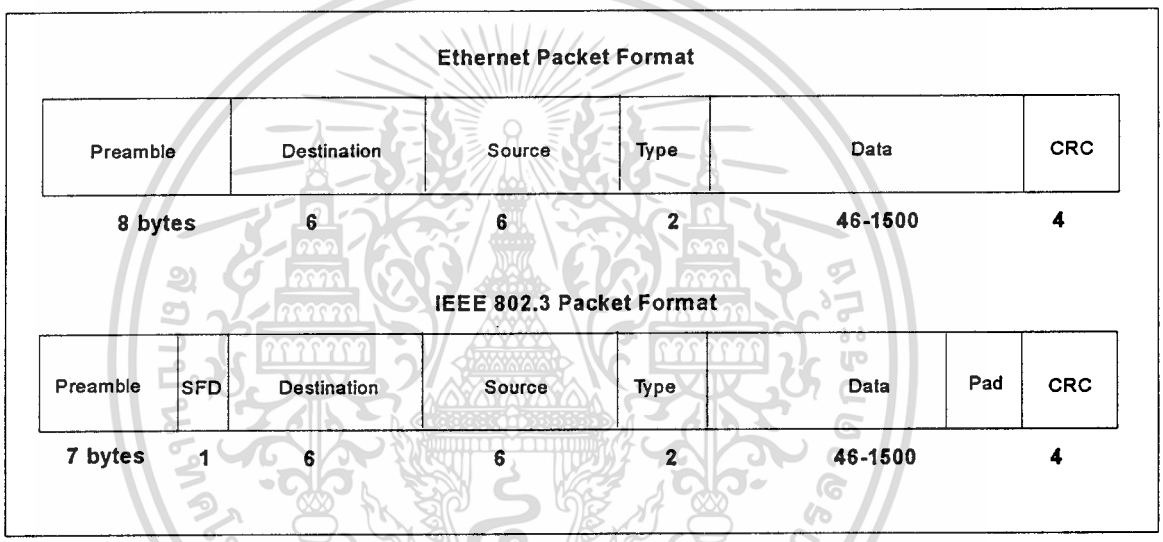
Local addresses : เป็นแอดเดรสที่ใช้ภายในและไม่ซ้ำการแอดเดรสอื่นๆ ในเครือข่าย

Source Address :

Ethernet = 6 ไบท์ ประกอบด้วยบิทเริ่มต้นจะเป็น 0 เสมอ (P/M)

IEEE 802.3 = 6 ไบท์ ประกอบด้วยบิทเริ่มต้นจะเป็น 0 เสมอ (I/G)

ภาพที่ 6



แสดงการเปรียบเทียบรูปแบบของแพกเก็ต ของ Ethernet และ IEEE 802.3

Type และ Length : ในมาตรฐานของ IEEE 802.3 จะไม่มี Type ฟิลด์ แต่ที่ Type ฟิลด์ นี้จะถูกแทนที่ด้วย Length ฟิลด์ ซึ่งมีขนาด 2-ไบท์ ซึ่งจะแสดงถึงขนาดของ Data ฟิลด์ ใน แพกเก็ต

Data : ในมาตรฐานของ IEEE 802.3 จะทำการแยกส่วนของ Data ฟิลด์ ออกเป็น Data และ Pad ฟิลด์ ซึ่งขนาดของ Data filed นั้นจะมีค่าอยู่ใน Length ฟิลด์ โดยจะรวมเอา Pad ฟิลด์ เข้าไปด้วยเพื่อใช้ในกรณีที่ Data ภายใน แพกเก็ต ไม่ได้ตามมาตรฐานขนาดของ Data ฟิลด์ จะมีขนาดระหว่าง 46-1500 ไบท์.

CRC : ของ IEEE 802.3 จะเหมือนกับ Ethernet

บทที่ 3

Internet Protocol

การจัดการเกี่ยวกับ Addressing, Subnet และ ARP

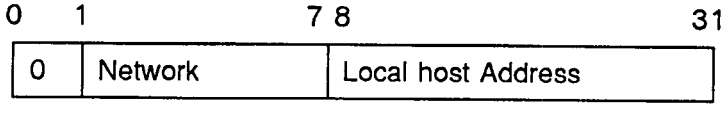
Internet Addresses

ในระบบของการสื่อสารที่เป็น Global communication โดยทั่ว ๆ ไปแล้วจะใช้ "universally - accepted method" ในการกำหนดแอดเดรสของ คอมพิวเตอร์, อุปกรณ์, หรือ Hosts ต่าง ๆ บน Internet ซึ่งอาจเป็นคอมพิวเตอร์ส่วนบุคคล, เทอร์มินอลเซิฟเวอร์, อุปกรณ์ Routers, หรือ UNIX Host และสิ่งหนึ่งที่เป็นมาตรฐานก็คือในการกำหนด Internet address อุปกรณ์บางชนิดอย่างเช่น Router ซึ่งจะมี Physical connection ที่เชื่อมต่อกันมากกว่าหนึ่งเครือข่าย ดังนั้นในการที่จะกำหนด Internet address ของแต่ละ Host นั้นต้องไม่ซ้ำกัน

Internet address นั้นจะใช้จำนวน 32-บิท แอดเดรสฟิลด์ ซึ่งบิทในแอดเดรสฟิลด์นั้นจะมีหมายเลขตั้งแต่ 0-31 และในฟิลด์ นี้จะแบ่งออกเป็น 2 ส่วนคือ ส่วนแรกแสดงถึง Host ของตัวเอง และส่วนที่เหลือจะแสดงถึง เครือข่าย ที่ Host นั้นเชื่อมต่ออยู่

รูปแบบของ Class A แอดเดรส

ภาพที่ 7

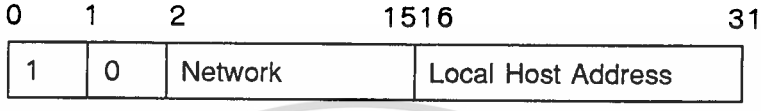


แสดงรูปแบบของ Internet Class A แอดเดรส

เครือข่ายของ Class A นั้นที่ บิทแรกนั้นจะกำหนดเป็น 0 ส่วน 7 บิท ถัดมาจะเป็นเครือข่าย และ 24-บิท ที่เหลือเป็น Host ตัวนั้นแล้วจะมี เครือข่าย Class A ได้ถึง 125 เครือข่าย และ แต่ละในเครือข่าย สามารถมีจำนวน Host ได้ถึง 16,777,214 Host

รูปแบบของ Class B แอดเดรส

ภาพที่ 8

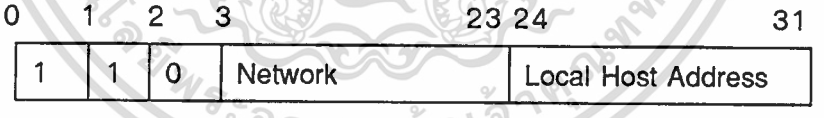


แสดงรูปแบบของ Internet Class B แอดเดรส

เครือข่ายของ Class B นั้น บิท ที่ลำดับที่ 2 บิทแรกจะถูกกำหนดเป็น 1 และ 0 ตามลำดับ 14-บิท ถัดมาเป็นเครือข่าย และ 16-บิท เป็น Host จะมีเครือข่ายใน Class A ได้ถึง 16,382 เครือข่าย และในแต่ละเครือข่ายสามารถมีจำนวน Host ได้ถึง 65,534 Host

รูปแบบของ Class C แอดเดรส

ภาพที่ 9



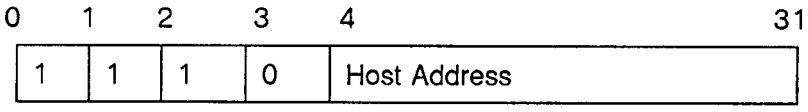
แสดงรูปแบบของ Internet Class C แอดเดรส

เครือข่ายของ Class B นั้น ตำแหน่งของ 3 บิทแรกจะถูกกำหนดเป็น 1, 1 และ 0 ตามลำดับ ส่วน 21 บิท ถัดมาเป็นเครือข่าย และ 8 บิท ท้ายสุดเป็น Host แอดเดรส ดังนั้นจะมีจำนวน เครือข่าย สามารถมีจำนวน Host ได้ถึง 254 Host

รูปแบบของ Class D แอดเดรส

เครือข่าย Class D นี้ จะถูกกำหนดไว้ใช้สำหรับ Multicast address และตำแหน่งของ 4 บิท แรกนั้นจะถูกกำหนดเป็น 1-1-1-0 ตามลำดับ Class D แอดเดรส นี้จะถูกกำหนดโดยหน่วยงาน IAB (Internet Activities Board)

ภาพที่ 10



แสดงรูปแบบของ Internet Class D แอดเดรส

ในการที่จะทำให้เข้าใจการอ่าน Internet address ได้ง่ายขึ้นจึงมีการเขียนเป็นเลขฐาน 10 จำนวน 4 ชุด ซึ่งแต่ละตัวจะถูกแบ่งด้วย dot ในรูปแบบลักษณะนี้เรียกว่า dotted decimal notation Notation นั้นจำนวน 32-บิท จะถูกแบ่งออกไปเป็น 8-บิท ต่อ 1 ฟิลด์ จะได้ทั้งหมด 4 ฟิลด์ เรียกว่า อ็อกเต็ท (Octets) และจะกำหนดค่าในแต่ละฟิลด์ ด้วยค่า Decimal number ตัวอย่างเช่น มีค่า Internet address ของ Class B ถูกกำหนดด้วยรูปแบบที่เป็นบิทดังนี้

10000001 00001111 00010001 00000011

ค่าของแต่ละ อ็อกเต็ทคือ

129 15 17 3

ค่าจะได้ค่า Internet address ในรูปของ dotted decimal

129.15.17.3

ช่วงค่าของ หมายเลขเครือข่าย ของแต่ละ Class โดยมี "hhh" จะแสดงถึง Host แอดเดรส ที่กำหนดโดยผู้ดูแลระบบเครือข่าย

Class A: 001 .hhh.hhh.hhh ถึง 126.hhh.hhh.hhh

Class B: 128.001.hhh.hhh ถึง 191.254.hhh.hhh

Class C: 192.000.001.hhh ถึง 223.255.254.hhh

Class D: 224.000.000.000 ถึง 239.255.255.255

กฎของการกำหนด แอดเดรส

1. บิทที่ใช้ในการกำหนด Host ของ Internet address นั้นจะต้องไม่เป็น บิท 1 ทั้งหมด เพราะจากมาตรฐานที่กำหนดไว้นั้น Internet address ที่มี Host ประกอบด้วย บิท 1 ทั้งหมด จะหมายถึงทุกเครื่อง "all host" ตัวอย่างเช่น แอดเดรส 128.1.225.225 จะหมายถึง Host ทุกตัวที่อยู่ใน เครือข่าย 128.1

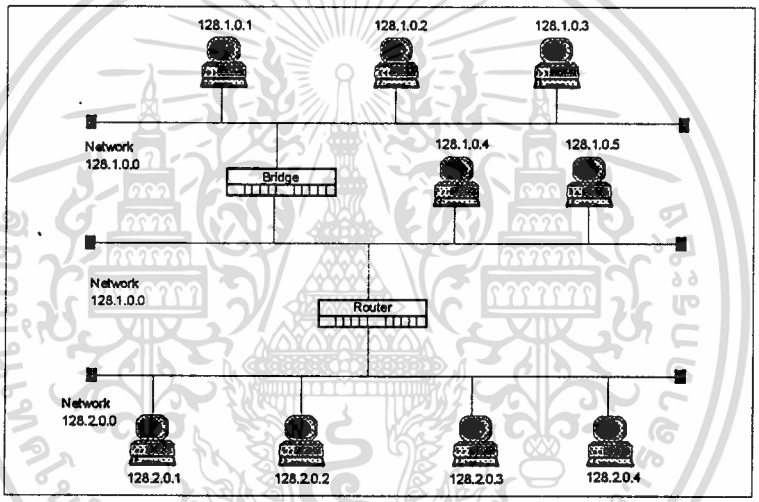
2. บิทที่ใช้ในการกำหนดเครือข่ายของ Internet address นั้นจะต้องไม่เป็น 0 ทั้งหมด เพราะจากมาตรฐานที่กำหนดไว้นั้น Host ที่ประกอบด้วย บิท 0 ทั้งหมดจะหมายถึงเฉพาะเครือข่ายนี้ "this network" ตัวอย่างเช่น แอดเดรส 0.0.0.63 จะหมายถึง Host 63 บนเครือข่ายนี้

3. ใน Class A หมายเลขเครือข่าย 127 จะถูกกำหนดไว้สำหรับ "loopbock" หมายถึงเมื่อมี ส่วนของข้อมูล ที่ส่งมาจาก Higher level protocol มายังเครือข่ายแอดเดรส 127 ก็ควรจะ Loopback ภายใน Host เอง

รูปแบบตัวอย่างของการใช้ Class B แอดเดรส

จะเห็นได้ว่า เซกเมนต์ (Segment) ที่ต่อเชื่อมโดยใช้ Bridges จะใช้เครือข่ายฟิลด์เดียวกัน และจะแตกต่างกันที่ Host ฟิลด์ ส่วนเซกเมนต์ ที่เชื่อมต่อโดยใช้ Router นั้นจะต้องมีเครือข่ายฟิลด์ ที่แตกต่างกัน

ภาพที่ 11



แสดงรูปแบบตัวอย่างการใช้ Class B แอดเดรส

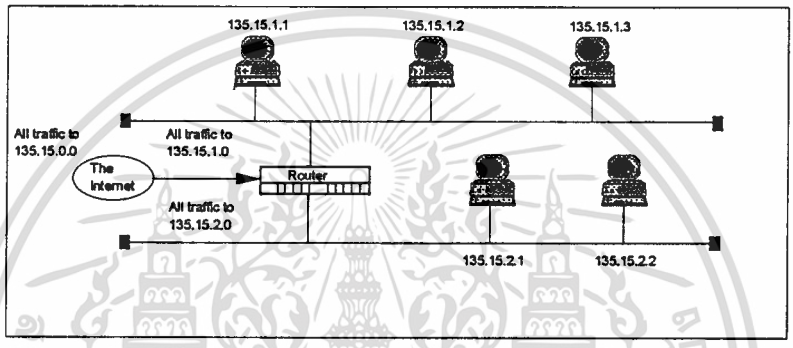
Subnetwork Addressing

Subnet จะเป็นเครือข่ายส่วนย่อยของ Internet network มีไว้สำหรับ ช่างเทคนิค หรือผู้ดูแลการระบบที่ประสงค์จะทำการจัดแบ่งเครือข่าย (Network) ให้เป็นหลายเครือข่าย โดยที่เป็นเครือข่าย ที่อิสระเชื่อมต่อมาจาก Router ในกรณีที่มี เครือข่าย TCP/IP หลาย เครือข่าย ต้องการที่จะติดต่อผ่านข้าม Router จำเป็นต้องมีการกำหนดเครือข่าย ให้แตกต่างกัน อย่างไรก็ตามถ้าเครือข่าย นั้นเป็นส่วนหนึ่งของ Internet ก็ไม่สามารถที่จะกำหนด หมายเลขเครือข่าย

(Network Number) ด้วยตัวเองได้ เพราะต้องขึ้นอยู่กับการกำหนดเครือข่าย ของ NIC Subnet นั้น สามารถกำหนดได้ในทุก ๆ Class ยกเว้น Class D ซึ่งเป็น Multicast

ในภาพ 12 แสดงให้เห็นถึงผู้ดูแลระบบได้สร้าง Subnetwork ขึ้นมา 2 Subnetwork จาก Class B Internet address และได้ใช้ค่าอ็อกเต็ท ตำแหน่งที่ 3 ของแอดเดรส ในการกำหนด Subnet ทั้งสองของเครือข่าย 135.15.0.0 และการติดต่อไปยัง Interface ถูกต้องโดยอาศัยค่า อ็อกเต็ท (Subnet) ในตำแหน่งที่ 3 ของแอดเดรส

ภาพที่ 12



แสดง Subnet addressing

Subnet mask จะใช้สำหรับให้ส่วนที่แสดงค่า Host ของ Internet address ทำการแบ่งออกเป็น 2 ส่วน ส่วนแรกใช้แสดงถึงหมายเลขของ Subnet และส่วนที่สอง ใช้สำหรับแสดงถึงค่า Host ของ Subnet

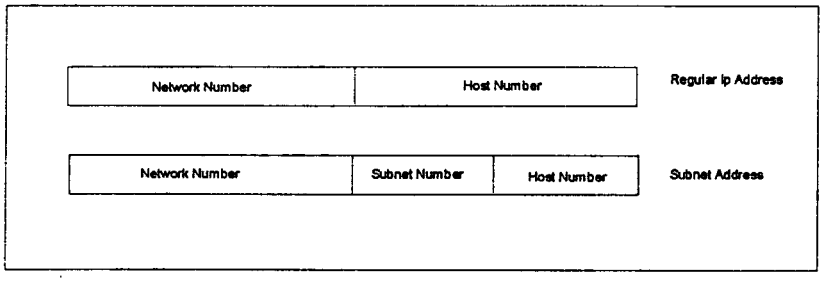
Host หรือ Router จะใช้บิท ที่นำหน้าของ IP address ในการรับรู้ถึง Class เมื่อรับรู้ถึง Class ของแอดเดรสแล้ว Host ก็จะรู้ว่าบิท ที่แสดงถึงส่วนของหมายเลขเครือข่าย ของแอดเดรส และบิทที่แสดงถึงส่วนของ Host mask จะเป็นตัวบอก ในการที่จะกำหนดหมายเลข Subnet

ตัวอย่างที่ 1 # Subnetwork

เมื่อ NIC ได้กำหนด Class B Internet address ให้คือ 128.001.0.0 ต้องการให้มี Subnet จำนวน 254 Subnet ซึ่งแต่ละ Subnet สามารถรองรับ ได้ถึง 254 Host

จาก Internet address ที่ได้นั้นตำแหน่งที่ 1 และ 2 ของอ็อกเต็ทของ IP address จะแสดงถึงเครือข่าย ตำแหน่งที่ 3 จะแสดงถึง Subnet ส่วนตำแหน่งที่ 4 จะแสดงถึง Host บน Subnet นั้น

ภาพที่ 13



ภาพแสดง Subnet Mask

วิธีการ

แสดงค่าของแอดเดรส ที่กำหนดโดย ในรูปแบบของไบนารี (binary format) จะได้ 128.1.1.1 = 1000000.00000001.00000000.00000000 ตัวเลขไบนารี ที่ขีดเส้นเอาไว้แสดงถึง เครือข่าย ใน Internet address

ไบนารีจำนวน 8 หลัก นั้นสามารถกำหนดได้ถึง 254 Subnet และสามารถที่จะมีจำนวนได้ตั้งแต่ 1-254 ส่วนค่าที่เป็น 0 และค่า 1 ทั้งหมดใน Subnet ฟิลด์ จะต้องไม่ถูกกำหนดเป็นซับเนทจริง (Actual subnets)

เลือก 8 บิท ที่สำคัญของ Host ของ Internet address ในการกำหนด Subnet ซึ่งแสดงได้จากตัวอักษรหนา

128.001.000.000 = 10000000.00000001.**00000000**.00000000

ทำการกำหนด Subnet mask โดยทุกบิท ของเครือข่าย และ Subnet ฟิลด์ ให้เป็น 1 และทุกบิท ของ Host ฟิลด์ เป็น 0

Network Number = 10000000.00000001.00000000.00000000

= 128.001.000.000

Subnet Mask = 11111111.11111111.11111111.00000000

= 255.255.255.000

ค่าของ Subnet Mask นี้ต้องกำหนดบนแต่ละ Host และ Router และควรจะต้องเป็น Mask เดียวกันด้วยเพื่อกำหนด Physical network ให้ใช้ได้ร่วมกันใน Internet address เดียวกัน

Subnet จำนวน 254 subnet แสดงได้ดังนี้

Subnet#1 100000000.00000001.00000001.00000000 = 128.001.001.000

Subnet#2 100000000.00000001.00000010.00000000 = 128.001.002.000

Subnet#3 100000000.00000001.00000011.00000000 = 128.001.003.000

:

Subnet#254 : 100000000.00000001.111111110.00000000

= 128.001.254.000

ช่วงแอดเดรส ของ Subnet #1

Subnet#1 100000000.00000001.00000001.00000000 = 128.001.001.000

Low Address : 10000000.00000001.00000001.00000001 = 128.001.001.001

High Address : 10000000.00000001.00000001.11111110 = 128.001.001.254

ตัวอย่างที่ 2

เมื่อ NIC กำหนด Internet Address ให้อยู่ใน Class B = 128.001.000.000 ต้องการ 2 Subnet และแต่ละ Subnet สามารถที่จะมี host ได้ถึง16,381 host

วิธีการ

ทำการแปลงแอดเดรส ให้อยู่ในรูป ไบนารี

128.001.000.000 = 10000000.00000001.00000000.00000000

ในส่วนเลขไบนารี ที่ขีดเส้นใต้นั้นแสดงให้เห็นถึงเครือข่ายของ Internet address ที่กำหนดจาก NIC

จำเป็นต้องใช้ไบนารี 2 digits ในการกำหนด Subnet จะได้เป็น 1-2 ค่าที่เป็น 0 ทั้งหมด และ 1 ทั้งหมดไม่ควรจะนำมากำหนดให้เป็นซับเนทจริง (actual subnet)

ตารางที่ 1

แสดงค่าของ subnet ที่เป็น 1 ถึง 2

Decimal	Binary
1	01
2	10

เลือกบิท สูงสุดที่สำคัญ 2 บิท ของ Host จาก Internet address เพื่อนำมากำหนด Subnet ซึ่งแสดงได้เป็นตัวอักษรหนา ดังนี้

128.001.000.000 = 10000000.00000001.**OO**000000.00000000

ทำการเลือก Subnet Mask โดยการกำหนดให้ทุกบิท ของเครือข่าย ฟิลด์ และ Subnet ฟิลด์ เป็น 1 ส่วน บิท ของ Host ฟิลด์ จะกำหนดให้เป็น 0

Network Number : 10000000.00000001.00000000.00000000

= 128.001.000.000

Subnet Mash : 11111111.11111111.11000000.00000000

= 255.255.192.000

ค่าของ Subnet Mask นี้ต้องทำการกำหนดให้ที่แต่ละ Host และ Router เพื่อใช้ Mask เดียวกันใน Internet adress เดียวกัน และทั้ง Subnet ทั้งสองจะมีแอดเดรส ดังนี้

Subnet #1 10000000.00000001.01000000.00000000 = 128.001.064.000

Subnet #2 10000000.00000001.10000000.00000000 = 128.001.128.000

ตัวอย่างที่ 3

NIC ได้ทำการกำหนด Internet address ให้อยู่ใน Class B คือ 128.001.000.000 ต้องการใช้ จำนวน 6 Subnetwork แต่ละ Subnet สามารถที่มี Host ได้ 8190 Host

วิธีการ

แปลง แอดเดรส ให้อยู่ในรูป ไบนารี

128.001.000.000 = 10000000.00000001.00000000.00000000

ส่วนที่ขีดเส้นใต้เอาไว้จะแสดงถึงส่วนข้อมูลเครือข่ายของ Internet address ดังนั้นจะต้องใช้ 3 ไบนารี ในการกำหนดให้ได้ 6 Subnet ซึ่งแต่ละ Subnet นั้นจะมีเลขตั้งแต่ 1-6 ส่วนค่าที่เป็น 0 และ 1 ทั้งหมดจะถูกกันไว้ใช้สำหรับซับเนทจริง

ตารางที่ 2

แสดง Subnet number ที่มีค่าตั้งแต่ 1-6

Decimal	Binary
1	001
2	010
3	011
4	100
5	101
6	110

เลือก บิท สูงสุดของส่วนที่แสดงต่อ Host ในการกำหนด Subnet ซึ่ง บิท ที่เลือกนั้นเป็นตัวหนาดังที่แสดง

128.001.000.000 = 10000000.00000001.**000**00000.00000000

และสามารถกำหนด Subnet Mask โดยการกำหนดให้ทุก บิท ของ เครือข่าย และ Future subnet ฟิลด์ เป็น 1 และทุก บิท ของ Future host ฟิลด์ เป็น 0

Network Number : 10000000.00000001.**000**00000.00000000 = 128.001.000.000

Subnet Mask : 11111111.11111111.11100000.00000000 = 255.255.224.000

Subnet Mask ที่ได้มานี้ต้องกำหนดให้แต่ละ Host และ Router ด้วยดังนั้นจะได้ Subnet จำนวน 6 Subnet ดังนี้

Subnet #1 10000000.00000001.00100000.00000000 = 128.001.032.000

#2 " 010 " = 128.001.064.000

#3 " 011 " = 128.001.094.000

#4 " 100 " = 128.001.128.000

#5 " 101 " = 128.001.160.000

#6 " 110 " = 128.001.192.000

Internet Protocol Broadcast สามารถแยกเป็น 2 ชนิดดังนี้

1. Limited Broadcast แพกเก็ต ที่ส่งไปยัง IP address 255.255.255.255 หรือ 0.0.0.0 จะถูกกำหนดไว้ให้เป็น "limited broadcast"แพกเก็ต ซึ่งมีจุดมุ่งหมายใช้ในเครือข่ายภายใน เพื่อให้เครือข่าย และ Host สามารถรับรู้ถึงแอดรแดรสปลายทาง ต่าง ๆ Limited broadcasts นี้จะไม่สามารถส่งผ่านไปยัง Router แต่สามารถที่จะส่งผ่านไปยังอุปกรณ์ทวนสัญญาณ (Repeaters) และ Bridge ได้

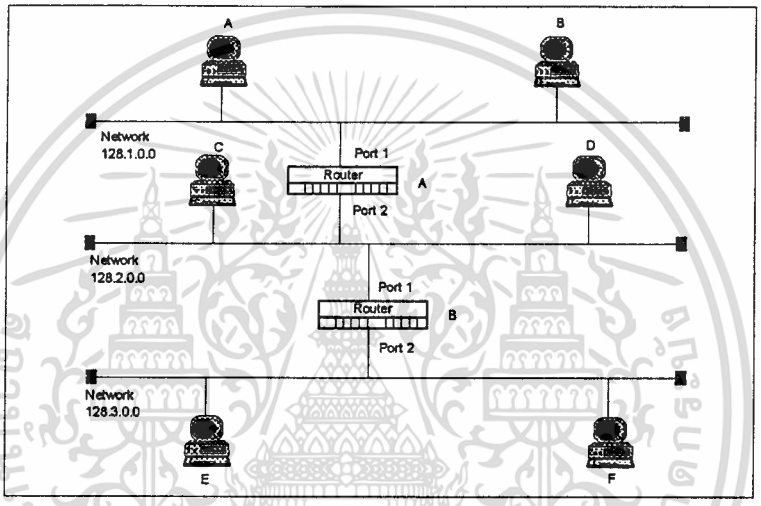
2. Directed Broadcast แพกเก็ต ที่ใช้ส่งไปยัง IP แอดเดรสปลายทาง โดยเฉพาะที่ส่วนที่แสดงค่า Host ของ IP แอดเดรส นั้นจะประกอบด้วยบิท 1 หรือบิท 0 ทั้งหมดเช่น 180.100.255.255 หรือ 180.100.0.0 เรียกว่า "Directed broadcast" แพกเก็ต และ

Directed broadcasts สามารถที่จะส่งผ่านไปยัง Router และจะ Broadcast ไปยังทุก ๆ Host ที่เป็นเครือข่ายปลายทาง

ตัวอย่าง เครือข่าย ที่ไม่มี Subnetwork

จากภาพใน 14 นั้นจะประกอบไปด้วย 3 เครือข่าย โดยที่ Router นั้นสามารถที่จะรับรู้ เครือข่าย ทั้ง 3 โดยการแลกเปลี่ยนข้อมูล Routing ซึ่งกันและกัน

ภาพที่ 14



ตัวอย่างของ IP address ที่ปราศจาก Subnets

ตารางที่ 3

แสดงถึงส่วนของข้อมูล การรับรู้ Broadcast จาก Host C.

Sender	Destination	IP Address Recipients
Host C	255.255.255.255	Host D, R_A Port2,R_B Port 1
Host C	128.1.255.255	Host A, Host B, R_A Port 1
Host C	128.2.255.255	Host D, R_A Port 2, R_B Port 1
Host C	128.3.255.255	Host E, HostF, R_B Port 2

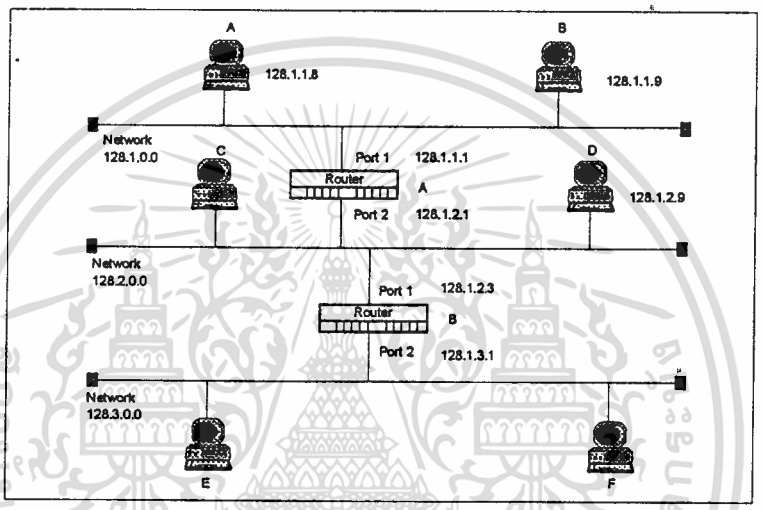
ตัวอย่าง เครือข่าย ที่ประกอบด้วย Subnetwork

Directed broadcast นั้นสามารถที่กำหนดให้ส่งไปยัง Subnet ที่ต้องการได้แต่ไม่สามารถที่จะส่งไปยังทุก Subnet

Physical address ด้วย โดย Host ทั้งสองนั้นสามารถที่จะใช้ Data Link layer Protocol ในการรับส่งส่วนข้อมูล ไปบนเครือข่ายได้

สำหรับตัวอย่างที่แสดงในภาพ 15 สมมุติให้มีค่า Subnet mask เป็น255.255.255.0 ซึ่งตัว Router แต่ละตัวนั้นสามารถที่จะรับรู้ Subnetwork ทั้ง 3 Subnetwork ได้

ภาพที่ 15



แสดงตัวอย่าง IP address ที่ประกอบด้วย Subnetwork

ตารางที่ 4

แสดงตัวอย่างการรับรู้ของการส่ง Broadcast ส่วนข้อมูล หลายชนิดจาก Host C.

Sender	Destination IP Address	Recipients
Host C	255.255.255.255	Host D, Rtr.A Port2, Rtr.B Port1
Host C	128.1.1.255	Host A, Host B, Rtr.A Port1
Host C	128.1.2.255	Host D, Rtr_A Port2, Rtr_B Port1
Host C	128.1.3.255	Host E, Host F, Rtr_B Port2

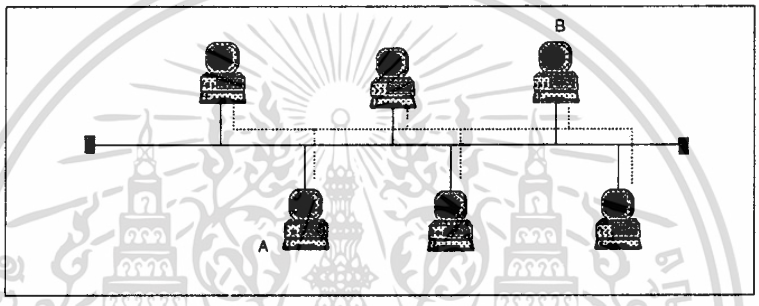
Address Resolution Protocol (ARP)

ถ้าในกรณีที่ Host ที่อยู่บนเครือข่าย มีความต้องการที่จะติดต่อสื่อสารกับ Host อีกตัวหนึ่งนั้น Host ทั้งสองจะต้องรู้มากกว่า Internet Address ของกันและกัน จำเป็นต้องรู้ถึง

ARP Request

Host A จะทำการ broadcast ที่เป็น ARP request packet ซึ่งประกอบไปด้วย Internet address ปลายทางซึ่ง Broadcast นี้ตัว Bridge สามารถที่จะส่งผ่านออกไปได้ แต่ไม่สามารถส่งผ่านจาก Router ได้ ที่ตัวHost A จะทำการส่งแพกเก็ต ไปยัง Host B โดยที่แพกเก็ต นั้นจะประกอบ Internet address กับ Ethernet address ดังนั้น Host ทุกตัวในเครือข่ายจะสามารถรับ Request จาก Host A ได้ แต่เฉพาะ Host B เท่านั้นที่จะตอบกลับ

ภาพที่ 17

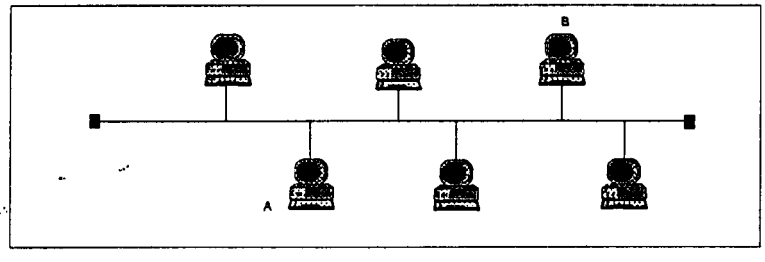


แสดง ARP Request

ARP Response

Host B จะสามารถที่จะรับรู้ Internet Address และจะทำการตอบกลับไปยัง Host A ที่ Request มา โดยการส่ง ARP reply Packet ที่มี Ethernet address ของตัวมันเองกลับไปยัง Host A ทำให้ Host A นั้นรับรู้ถึง Ethernet address ของ Host B ซึ่งเป็นส่วนที่สำคัญในการส่งแพกเก็ต ไปยัง Host B บน Local physical network

ภาพที่ 18



แสดง ARP Response

2. คุณสมบัติของ ARP คุณสมบัติของ ARP นั้นสามารถที่ทำงานได้อย่างมีประสิทธิภาพคือ

- จะทำการเก็บจำนวนของ ARP broadcast ไว้น้อยที่สุด และ host นั้นจะให้ ARP จัดการในส่วนที่เป็น Cache ที่ใช้สำหรับเปรียบเทียบ Internet-to-Ethernet ดังนั้น Host เองไม่จำเป็นต้องส่ง ARP ทุก ๆ ครั้งเมื่อต้องการที่จะส่งแพกเก็ต และเมื่อตาราง (Table) ต่าง ๆ ที่อยู่ใน Cache มีขนาดใหญ่มากขึ้น มันจะทำการนำส่วนที่ไม่ได้ใช้ออก ภายในช่วงเวลาที่กำหนดไว้ ก่อนที่ Host จะทำการส่งแพกเก็ต มันจะทำการตรวจสอบดูใน Cache ของมันเองก่อนที่จะส่ง ARP request ออกไป

- ในส่วนของการสื่อสารข้อมูลในเครือข่ายนั้น จะสามารถที่จะหลบหลีกได้โดยที่ใน ARP request นั้นจะประกอบไปด้วย Internet-to-Ethernet ซึ่งส่วนที่จะติดต่อด้วยนั้นสามารถที่จะทำการเก็บไว้ใน Cache ของตัวเอง

- ในขณะที่มีการส่ง ARP request broadcast ออกไปนั้นทุก ๆ เครื่องที่เชื่อมต่ออยู่ในเครือข่ายเดียวกันนั้นจะได้รับ Broadcast ด้วย และสามารถที่จะทำการเรียนรู้ว่า เครื่องที่ส่งมาเป็นเครื่องใด จะทำการเก็บ Interet-to-Ethernet เอาไว้ใน Cache ของตัวเอง โดยสามารถเขียนให้เป็นขั้นตอนของ ARP request ได้ดังนี้

```
Search ARP cache for binding
IF (mapping in cache) THEN
        Extract the Ethernet address
        Build the Ethernet packet
        Transmit the packet
ELSE
        store the outgoing data
        counter = 0
        ARP = received = FALSE
        DO
                counter = counter + 1
                broadcast the ARP request
                WAIT for ARP reply
                IF (ARP reply received) THEN
                        cache the mapping
                        Extract Ethernet address
                        retrieve the stored outgoing data
```

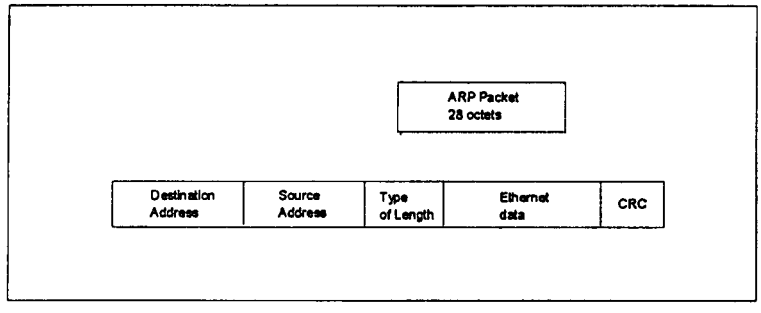
```
                    build the Ethernet packet
                    transmit the packet
                    ARP-received = TRUE
                END {IF ARP reply received...}
            UNTIL ((counter > max.value) or (ARP.received))
END {IF binding in cache...}
```

ขั้นตอนของ ARP : Reply ในการได้รับ Request จาก Host

```
Receive the broadcast ARP request
Extract senders Internet-to-Ethernet mapping
IF (mapping already in cache) THEN
        update entry by copying Ethernet Address from the packet
ELSE
        add the binding to the cache
END
IF      (local host is the target of ARP request) THEN
        build ARP reply and send with Ethernet address
ELSE
        Ignore the ARP request
END
```

3. รูปแบบและโครงสร้างของ ARP ในภาพที่ 3-13 แสดงส่วนที่ ARP แพกเก็ต ถูกบรรจุอยู่ใน Ethernet แพกเก็ต ส่งจาก Host หนึ่งไปยังอีก Host หนึ่ง

ภาพที่ 19



แสดงรูปแบบโครงสร้างของ ARP แพกเก็ต

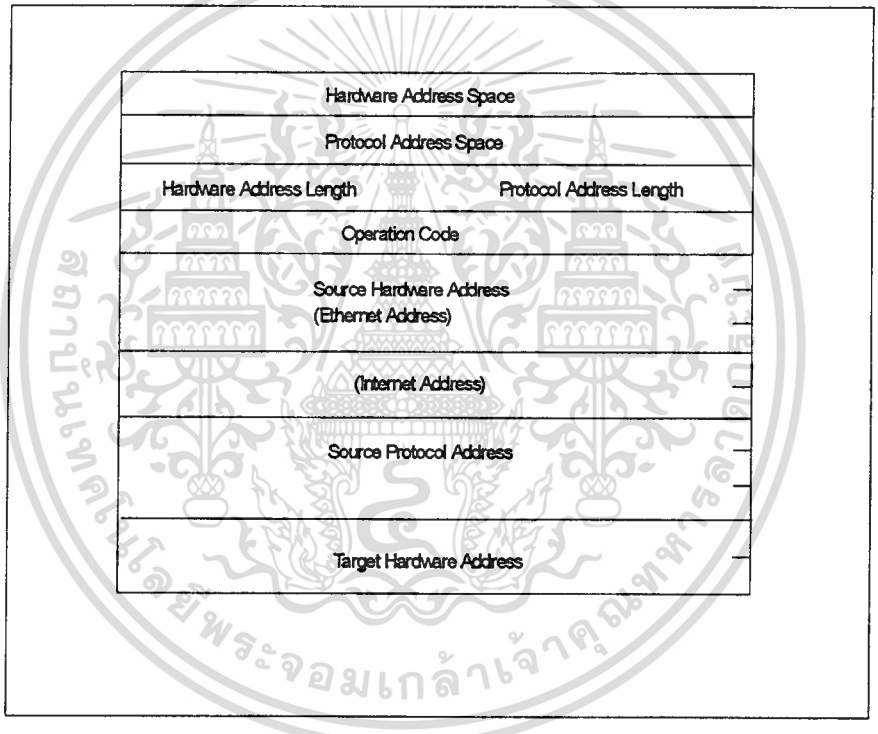
หมายเหตุ ARP แพกเก็ต จะไม่อยู่ในส่วนของ IP โปรโตคอล จึงจะไม่มีส่วนของ IP headers ค่าของ Ethernet type ที่ใช้สำหรับ ARP แพกเก็ตในการส่งผ่านฟิลด์ข้อมูล ก็คือ

- ARP request % 0806
- ARP response % 0835

รูปแบบแพกเก็ตของ ARP

ภาพที่ 20 แสดงถึงโครงสร้างของ ARP แพกเก็ตซึ่งจะถูกบรรจุอยู่ในส่วนของฟิลด์ข้อมูล ของ Ethernet แพกเก็ต

ภาพที่ 20



แสดงรูปแบบของแพกเก็ต ARP

Hardware Address Space : เป็นส่วนที่แสดงชนิดของฮาร์ดแวร์ ที่ใช้ใน Network level เช่น ถ้าเป็น Ethernet จะมีค่าเป็น 1

Protocol Address Space : แสดงถึงค่าโปรโตคอล ที่ใช้ใน Network level

Hardware Address Length : แสดงถึงขนาดของฮาร์ดแวร์แอดเดรสเป็นไบท์ สำหรับ Ethernet จะมีค่าเป็น 6

Protocol Address Space : แสดงถึงขนาดของโปรโตคอลแอดเดรสเป็นไบท์ สำหรับ TCP/IP จะมีค่าเป็น 4

Operation Code : สำหรับแสดงรายละเอียดหน้าที่ของแพกเก็ต (ARP request, ARP response, RARP request, RARP response)

Source Hardware Address : สำหรับฮาร์ดแวร์แอดเดรสของ Host ที่ส่งแพกเก็ต โดยส่วนใหญ่แล้วจะเป็น Ethernet address

Source Protocol Address : เป็น Internet address ของ Host ที่ส่งแพกเก็ต

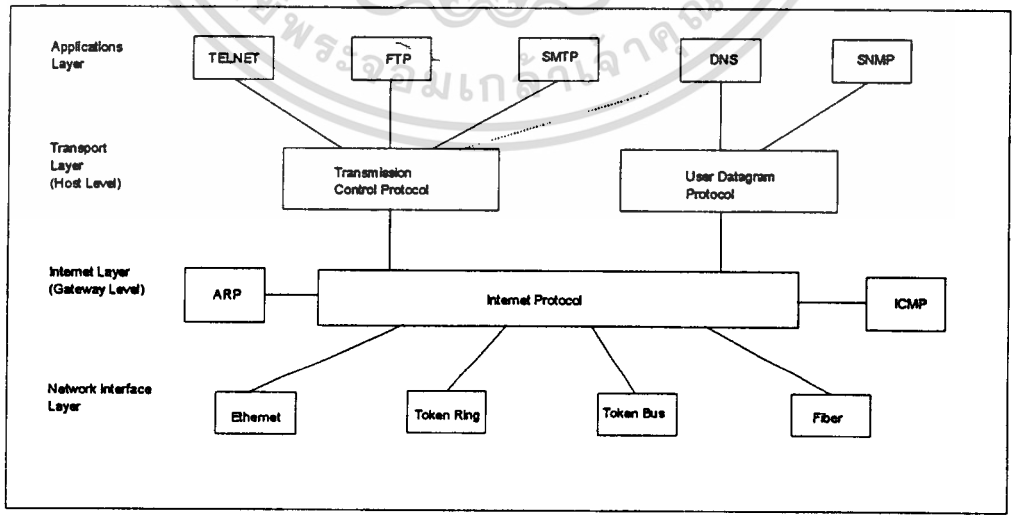
Target Hardware Address : เมื่อมีการใช้ RARP request ส่วนนี้ก็คือ Destination hardware address ซึ่งการตอบกลับมานั้นจะต้องมี Destination hardware และ Internet address

Target Protocol Address : เมื่อมีการใช้ ARP request ในส่วนนี้ก็คือ Internet address เมื่อมีการตอบกลับก็จะมี Destination hardware และ Internet address

กลุ่มของ Internet Protocol

ภาพที่ 21 ได้แสดงถึงโปรโตคอลพื้นฐาน ที่ใช้ในกลุ่มของ TCP/IP โดยที่แต่ละส่วนนั้นจะมีความสัมพันธ์กัน

ภาพที่ 21



แสดงกลุ่มของ Internet Protocol

TCP/IP นั้นเป็นกลุ่มของโปรโตคอล ซึ่งแบ่งได้ตามการใช้งานได้ 4 layers

- Application protocol เช่น การจำลองเทอร์มินอล หรืออิเล็กทรอนิกส์เมล์ Electronic mail
- Transport protocol เช่น TCP ซึ่งใช้สำหรับบริการทางการติดต่อสื่อสาร ในส่วน Application layer ต้องการใช้
- Internet layer protocol เช่น IP ซึ่งจะทำหน้าที่ในการรับส่งส่วนข้อมูล ไปยังปลายทางซึ่งต้องผ่านไปยังเครือข่ายหลาย ๆ เครือข่าย
- Network Interface protocol ซึ่งจะจัดการเกี่ยวกับทางด้าน Physical medium เช่น Ethernet หรือ point-to-point serial line (PPP)

Network Interface layer Protocol

เป็นส่วนที่มีความสัมพันธ์กับ OSI Physical และ Data Line layers network Interface layer นี้ได้มีการกำหนดกฎระเบียบในการที่ Host จะติดต่อในเครือข่ายท้องถิ่น

Internet layer Protocol

Internet level นี้จะมีความสัมพันธ์กับ OSI Network layer ซึ่งในส่วนของ Network layer นั้นจะมีการกำหนดเกี่ยวกับการส่งผ่านข้ามไปยังเครือข่ายสำหรับ Global addressing และ Routing

Internet Protocol (IP) : หน้าที่ของ IP คือการหาเส้นทางสำหรับที่จะส่งส่วนข้อมูลไปยัง IP ปลายทาง ยังมีอีกโปรโตคอลอีก 2 ชนิด ที่สามารทำงานได้อย่าง IP ที่ Internet level ซึ่งก็คือ

ARP : (Address Resolution Protocol) จะทำหน้าที่ในการ Map ให้ IP host address กับ Data link address (Ethernet address)

ICMP : (Internet Control Message Protocol) ใช้สำหรับตรวจสอบ ERROR MESSAGE

Transport layer protocol

Transport layer จะมีความสัมพันธ์กับ Transport layer ของ OSI Reference model ซึ่งหน้าที่ของ Transport layer นี้ก็คือ จะช่วยการติดต่อระหว่าง Application program ส่วนของ Internet level นั้นจะทำหน้าที่เพียงจัดส่งส่วนข้อมูล ให้ Host ที่จะส่งแพกเก็ต ไปยังปลายทางที่ถูกต้องบน Internet อย่างไรก็ตามในส่วนของ Application ก็ยังต้องการงานบริการเฉพาะ เช่น Reliability, error rate, delay หรือ characteristics ซึ่งส่วนของ Transport layer สามารถที่จะจัดการงานบริการเช่นนี้ได้

TCP/IP ได้นั้นได้แบ่งแยก levels ของงานบริการไปยัง Application โปรแกรม ออกเป็น 2 Level คือ TCP (Transmission Control Protocol) จะให้บริการทางด้าน end-to-end data stream จะประกอบด้วยขั้นตอนในการส่งผ่านข้อมูลให้ได้อย่างมีประสิทธิภาพ เช่น Checksums, Sequence numbers, Times acknowledgments, และ Retransmission procedures ส่วน Level ที่สองคือ UDP : (User Datagram Protocol) จะทำหน้าที่เป็น transaction-oriented สำหรับ Application ที่ไม่ต้องการ Reliable stream service ซึ่ง UDP สามารถที่จะทำเหมือน Multiplexing ไปยังแอดเดรส ต่าง ๆ ได้จะมีเพียงการตรวจสอบ CRC เท่านั้น

Application Layer Protocol

ในส่วนที่เป็น 3 Layer บนสุดของ ISO Layer นั้นประกอบด้วย Application, Presentation, และ Session Layer นั้นสามารถที่จะรวมไว้เป็น Level Layer สูงสุดของ TCP/IP ได้ซึ่ง Application นี้จะเป็นส่วนที่ติดต่อกับ Transport level protocols เพื่อใช้ในการรับส่งลำดับ (Sequence) ของข้อความ หรือ Stream ของไบท์ ในบาง Application โปรโตคอลที่มาจาก TCP/IP นั้นจะรวมไปถึง

Telnet สำหรับเทอร์มินอลที่อยู่ห่างไกลที่จะติดต่อกับ Host หรือติดต่อกับเทอร์มินอลเซิฟเวอร์

FTP (File Transfer Protocol) จะใช้ในส่วนของการส่งไฟล์ จาก Host หนึ่งไปยังอีก Host หนึ่ง โดยจะมีความเชื่อถือจาก TCP ที่ทำให้แน่ใจว่าไฟล์ ที่ได้สมบูรณ์

SMTP (Simple mail transfer protocol) ประกอบไปด้วยวิธีการสำหรับจดหมายอิเลคทรอนิกส์ จะใช้ TCP ในการส่งผ่านข้อความจดหมายเหมือนกัน

DNS (Domain name system) ใช้สำหรับกำหนดชื่อไปยัง แอดเดรส เพื่อใช้สำหรับไม่ต้องรู้ถึง Internet address ของแต่ละ Internet host

SNMP (Simple Network Management Protocol) เป็นโปรโตคอลมาตรฐาน สำหรับการควบคุมเครือข่าย (Network management) ในการควบคุม Host และ Router ที่เป็น TCP/IP

<u>การติดต่อสื่อสารโดยใช้ TCP/IP</u>

ตัวอย่างเช่นต้องการจะส่งข้อความจดหมาย ไปยังอีก Host หนึ่งบนเครือข่าย สามารถแบ่งเป็นขั้นตอนของแต่ละ level ได้ดังนี้

- Application level

จะใช้ applications-level protocol สำหรับการส่งจดหมาย โดยมีการกำหนดให้ส่งจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง และมีทั้งผู้ส่ง และผู้รับรวมถึงข้อความในจดหมาย แสดงส่วนของข้อมูลที่อยู่ใน ข้อความจดหมาย

YourMailMessage

Stream ของข้อมูลนี้จะถูกส่งโดย TCP Module

- Transport Service level

TCP จะทำการแบ่งแยกไฟล์ ของจดหมาย ที่ขนาดใหญ่ออกเป็นหลาย ๆ เซกเมนต์

Your Mail Message

จากนั้นจะทำการใส่ Header ไปยังด้านหน้าของแต่ละเซกเมนต์ ซึ่งให้ Header นั้นจะประกอบไปด้วย Source port, Destination port และ Sequence number ถ้ากำหนดให้ TCP header เป็น "T" จะได้

(T) Your (T) Mail (T) Message

ส่วนข้อมูลนี้ จะส่งผ่านลงไปยัง Internet level สำหรับ ขั้นตอนการส่งโดยใช้ IP

- Internet Level

ส่วนที่ติดต่อระหว่าง TCP กับ IP นั้น จะมีความสัมพันธ์กัน โดยที่ TCP นั้นจะจัดการเกี่ยวกับ ส่วนข้อมูล ที่มีค่าปลายทาง ไปยัง IP โดยที่ IP นั้นจะไม่รู้ได้เลยว่า ส่วนข้อมูลนั้นมีความสัมพันธ์กับส่วนข้อมูล อื่นๆ อย่างไร

หน้าที่ของ IP ก็คือ จะทำการหาเส้นทางสำหรับส่วนข้อมูล ที่จะส่งไปยังปลายทาง เพื่อให้ ที่ปลายทางนั้นสามารถที่จะรับ ส่วนข้อมูล ต่าง ๆ ได้ และสามารถนำกลับมาเป็นไฟล์ของจดหมาย ที่สมบูรณ์ได้

ในการส่งส่วนข้อมูลไปยัง ปลายทางนั้น IP ก็จะทำการใส่ Header ของแต่ละ เซกเมนต์ ด้วย ซึ่งจะประกอบด้วย Source Internet address, Destination Internet address, protocol number และ ท้ายสุดจะเป็น Checksum ถ้ากำหนดให้ IP header เป็น "I" จะได้

(I) (T) Your (I) (T) Mail (I) (T) Message

แต่ละส่วนข้อมูล จะถูกผ่านลงไปยัง Network level สำหรับที่จะส่งต่อไปยัง Physical Network

- Network Interface level

ในส่วนของ physical network ก็จะทำการใส่ Header ลงไปในแต่ละส่วนข้อมูลด้วย สมมติว่าจะส่งไปยัง Ethernet network ถ้ากำหนดให้ Ethernet header เป็น "E" และ Ethernet Checksum เป็น "C" จะได้เป็น

(E)(I)(T) Your (C) (E)(I)(T)Mail(C) (E)(I)(T)Message (C)

- Destination Station

เมื่อสเตชั่นปลายทาง ได้รับแพกเก็ต ต่าง ๆ แล้วส่วนของข้อมูล ก็จะถูกจัดการตาม layers ต่าง ๆ ของโปรโตคอล

Ethernet interface ตรวจสอบชนิดของ Ethernet และส่งผ่าน ส่วนข้อมูล ไปยัง IP

IP จะตรวจสอบโปรโตคอลฟิลด์ และส่งผ่าน ส่วนข้อมูลไปยัง TCP

TCP ตรวจสอบหมายเลขลำดับ (Sequence number) และข้อมูลอื่น ๆ เพื่อจะทำการรวบรวม ส่วนข้อมูล ให้เป็นไฟล์ของจดหมายที่สมบูรณ์ ซึ่งไฟล์ ที่ได้ก็จะส่งไปยัง Mail Application

โครงสร้างของข้อมูล IP

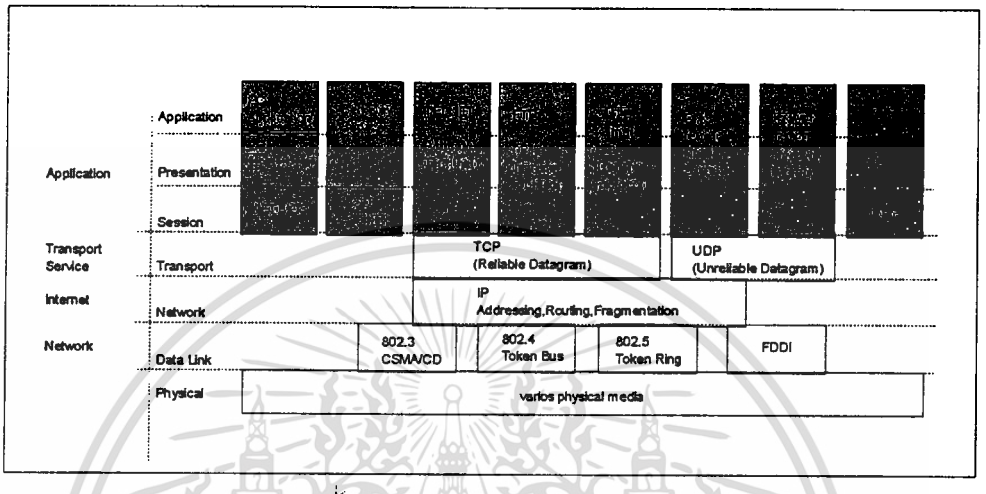
Internet Protocol (IP) จะมีหน้าที่สำหรับส่งส่วนข้อมูล ผ่านไปยังเครือข่าย ซึ่ง IP นั้น จะรับส่วนข้อมูลนั้นมาจากระดับของโปรโตคอลที่สูงกว่าเช่น TCP หรือ UDP

รูปแบบของ IP Routing

ในภาพที่ 25 แสดงถึงเครือข่ายขนาดเล็ก ซึ่งจะมีด้วยกัน 4 เครือข่าย และมี 3 Router ในส่วนที่จะไม่ได้แสดงถึง Hosts ต่าง ๆ ที่ต่ออยู่ในเครือข่าย เพราะว่าใน Router แต่ละตัวนั้นจะเป็นตัวส่งแพกเก็ต ต่าง ๆ ซึ่งขึ้นอยู่กับ หมายเลขเครือข่าย นั้น ๆ เอง และตัว Rquter

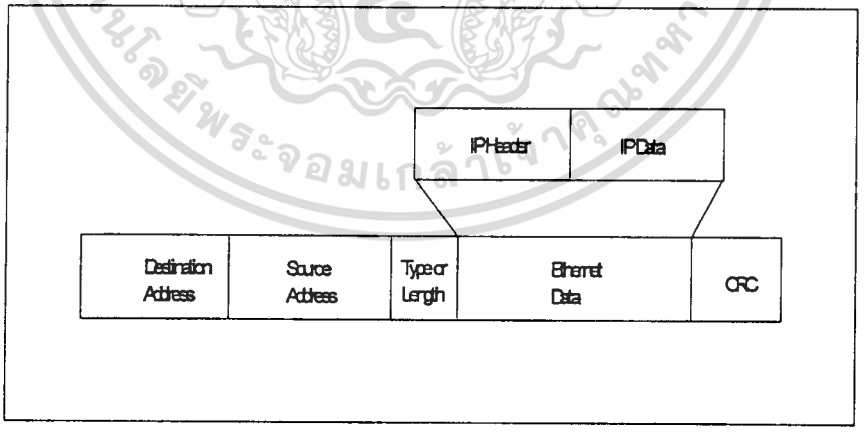
เองก็จะใช้ ARP ในการค้นหา Physical address ที่สัมพันธ์กับ Internet address สำหรับแต่ละ Host หรือ Router ที่เชื่อมต่ออยู่ในเครือข่าย

ภาพที่ 22



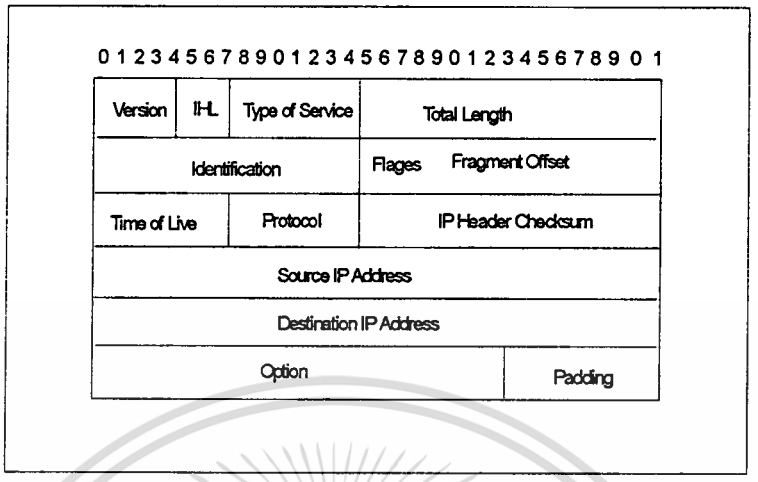
แสดงถึงการเปรียบเทียบระหว่าง IP และ OSI Reference Model

ภาพที่ 23



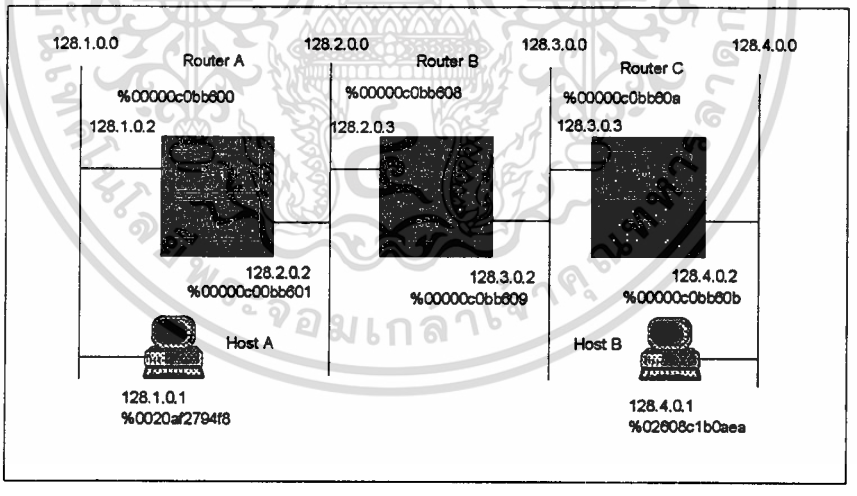
แสดงถึง IP datagram ที่อยู่ใน Data portion ของ Ethernet

ภาพที่ 24



แสดงถึงรูปแบบของ IP Header Format

ภาพที่ 25



แสดงตัวอย่าง Internet เครือข่าย ขนาดเล็ก

ค่าของตาราง Routing ของแต่ละ Router ในภาพที่ 25 แสดงได้ในตารางที่ 5 ถึง 7 ในตาราง Routing สามารถที่จะบอกถึง Destination IP Network, IP address ของ Router ตัวถัดไปและค่า Metric ที่แสดงถึงเส้นทางที่จะไปยังเครือข่ายปลายทาง (Destination Network)

ตารางที่ 5

แสดงตาราง Routing ของ Router A

Destination Network	Next Hop Router	Metric (Hops)
128.1.0.0	Direct Port1	0
128.2.0.0	Direct Port2	0
128.3.0.0	128.2.0.3	1
128.4.0.0	128.2.0.3	2

ตารางที่ 6

แสดงตาราง Routing ของ Router B

Destination Network	Next Hop Router	Metric (Hops)
128.1.0.0	128.2.0.2	1
128.2.0.0	Direct Port1	0
128.3.0.0	Direct Port2	0
128.4.0.0	128.3.0.3	1

ตารางที่ 7

แสดงตาราง Routing ของ Router C

Destination Network	Next Hop Router	Metric (Hops)
128.1.0.0	128.3.0.2	2
128.2.0.0	128.3.0.2	1
128.3.0.0	Direct Port1	0
128.4.0.0	Direct Port2	0

Internet Control Message Protocol (ICMP)

ICMP นี้เป็นโปรโตคอล ส่วนหนึ่งที่ใช้ในการติดต่อกันใน Internet Protocol (IP) หมายถึงว่าทั้ง Host หรือ Router ต่าง ที่ใช้ IP ในการติดต่อสื่อสารก็จำเป็นต้องใช้ ICMP เป็น

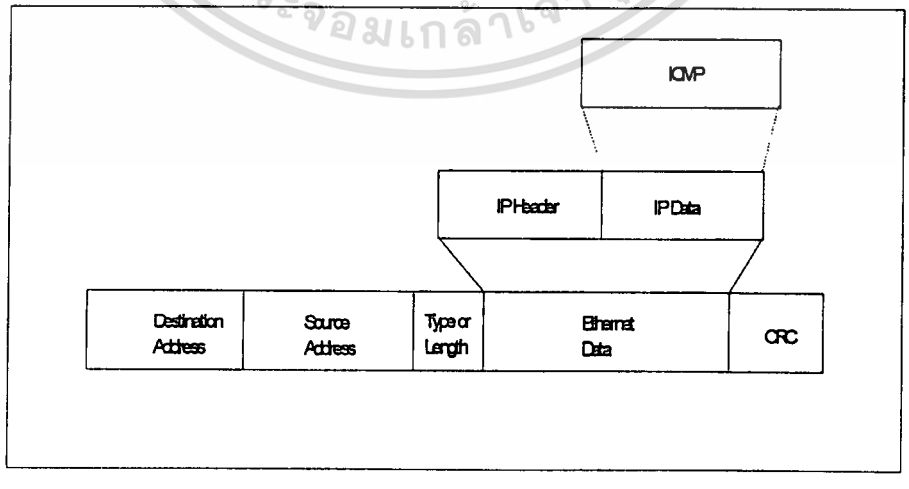
ส่วนประกอบด้วย หน้าที่ของ ICMP ก็คือ เมื่อ Router หรือ Host ปลายทาง ต้องการรายงานความผิดพลาดขั้นตอนของ ส่วนข้อมูล ไปยังต้นทางที่แท้จริงของแพกเก็ตนั้น โดยจะให้ ICMP เป็นตัวจัดการ ตัวอย่างของ ICMP Message ที่ใช้

- เมื่อ Router ต้องการที่จะยกเลิกส่วนข้อมูล เพราะว่าหมดเวลาของ Time-To-Live Counter
- เมื่อ Router ไม่มีส่วนที่รองรับเพียงพอในการส่งส่วนข้อมูล
- เมื่อ Host หรือ Router รับรู้ถึง Syntax error ใน IP header
- เมื่อ Router ไม่มีเส้นทางของ เครือข่ายปลายทางในตาราง Routing
- เมื่อ Router ตอบกลับไปที่ Host ต้นทางว่าให้ ใช้ Router ตัวอื่นเพราะว่ามีเส้นทางที่ใกล้กว่า

IP นั้นไม่ได้ถูกออกแบบมาให้การบริการรับส่งเป็นไปอย่างสมบูรณ์ ดังนั้นหน้าที่โดยหลักของ ICMP ก็คือจะทำหน้าที่ Feedback กลับเมื่อเกิดปัญหาในส่วนของการติดต่อสื่อสาร ICMP message นั้นถูกบรรจุอยู่ในส่วนข้อมูลของ IP datagram ก็เปรียบเสมือน IP datagram ธรรมดา ดังนั้นแล้ว การส่งที่เป็น ICMP message นั้นก็ไม่ได้รับประกันว่าทุกข้อความ ที่ส่งไปในจะต้องถึงผู้รับเสมอ

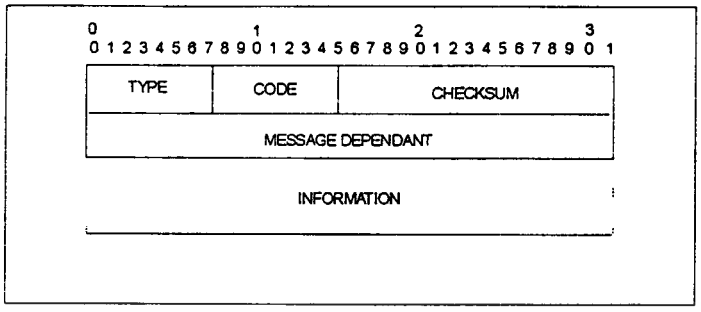
ICMP นั้นจัดเป็นสมาชิกตัวหนึ่งที่อยู่ใน Internet ของ TCP/IP โปรโตคอล เหตุผลที่ให้ ICMP ถูกบรรจุอยู่ในส่วนข้อมูลของ IP ก็คือ ICMP นั้นต้องสามารถที่ส่งผ่านไปยัง Router และ เครือข่าย ต่าง ๆ ไปยัง Host ปลายทางได้ ดังนั้นการใช้ Data Link layer protocol encapsulation จึงไม่เพียงพอที่จะทำให้ข้อความ ส่งผ่านข้าม Router ไปได้

ภาพที่ 26



แสดงถึงโครงสร้างของ ICMP

ภาพที่ 27



แสดงถึงรูปแบบของ ICMP Message Format

TYPE (8 บิท) เป็นส่วนที่บอกชนิดของข้อความ ซึ่งประกอบด้วย 13 ข้อความที่ ICMP กำหนดไว้คือ

ตารางที่ 8

แสดงส่วนของ TYPE ของ ICMP

TYPE Field Decimal	Message Type
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

Code (8 บิท) : สำหรับข้อมูลเพิ่มเติมของ message type

Checksum (16 บิท) : ตรวจสอบ Checksum สำหรับ ICMP message

Message Dependent (32 บิท) : ส่วนใหญ่แล้ว ICMP Message จะไม่ได้ใช้ แต่ส่วนนี้จะสำรองไว้ สำหรับขยายเพิ่มเติม(เมื่อใช้ส่งจะเป็น 0)

Information (variable) : จะรวมไปด้วย IP header และ 64 บิท แรกของฟิลด์ข้อมูล

User Datagram Protocol (UDP)

UDP นี้เป็นหนึ่งใน 2 ส่วนที่สำคัญของ Transport layer protocol ซึ่งอยู่เหนือ Internet Protocol (IP) โดยจะใช้วิธี transaction oriented เป็นส่วนบริการสำหรับ applications ทั่ว ๆ ไปที่ไม่ต้องการความสมบูรณ์ ของ Data stream มากนัก ตัวอย่างของ Application ที่ใช้ UDP

- Network File System (NFS)
- Domain Name Service (DNS)
- Trivial File Transfer (TFTP)
- Simple Network Management Protocol (SNMP)

UDP นั้นจะมีความสามารถ เหมือนกับ Internet Protocol (IP) ซึ่งจะมีส่วนของ flow control ในการจัดการแลกเปลี่ยนข้อมูลระหว่าง Host โดยที่ UDP นั้น จะไม่มีการรับหรือส่ง Acknowledgements ในการรับประกันว่า การส่งข้อมูลนั้นสมบูรณ์ และจะไม่มีการจัดการเกี่ยวกับลำดับของแพกเก็ต UDP นั้นมีคุณสมบัติเพิ่มเติม 2 ส่วนจาก IP คือ

- UDP มีความสามารถในการ Demultiplex Data จากส่วนของ Application process โดยขึ้นอยู่กับ หมายเลขของ Port ปลายทาง

- UDP header นั้นจะรวม checksum เพื่อตรวจสอบข้อผิดพลาด ที่เกิดขึ้นในขณะที่ส่งข้อมูลจาก Host ต้นทาง ไปยัง Host ปลายทาง

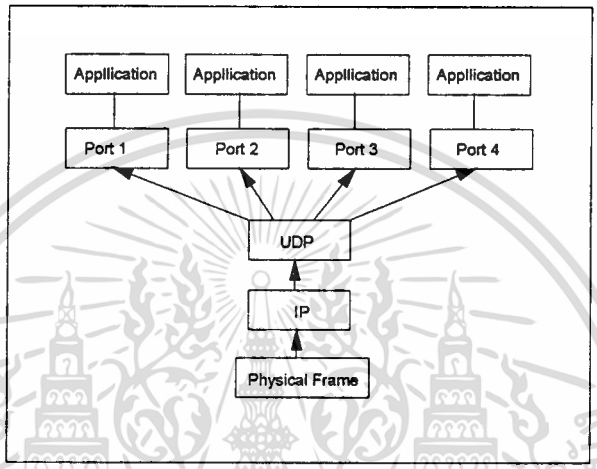
การ Demultiplexing Based บน Port Number

ในขณะที่ IP ทำหน้าที่เกี่ยวกับ Rounting function ทำให้เกิดการติดต่อกันระหว่าง Host ทั้งสองผ่านไปยัง Internet ก็จะมีวิธีการในการแบ่งแยก ไปยังส่วนต่างๆของ Host ได้ ทำให้ Application ต่าง ๆ ที่ทำงานอยู่บน Host นั้นจะสามารถรับส่งส่วนข้อมูลที่เป็นอิสระต่อกันได้

UDP นั้นจะมีส่วนการติดต่อของ Transactions ที่แยกกันระหว่าง Requester process และ Server process โดยที่ส่วนที่เป็น Requester process จะเป็น Active client ของ UDP ใน

ขณะที่ Server process เป็น Passive client การทำงานนั้น Active client จะทำการขอการบริการ (Service) จาก Remote passive client เมื่อได้รับการร้องขอแล้ว Passive client ที่จัดการกับ Listen socket อยู่จะรับรู้ถึงหมายเลขของ well_known port จะตอบกลับไปยังการบริการ (Service) นั้น ๆ

ภาพที่ 28



แสดงการ Demultiplexing

ตัวอย่างเช่น การใช้ UDP เกี่ยวกับการควบคุมและจัดการเครือข่าย โดยจะมี SNMP agents (the server process) ที่อยู่บน Host ที่ต้องคอยติดต่อกับ การร้องขอของเครื่องที่เป็นส่วนควบคุมและจัดการ (Network management stations (the requester process)) แต่ละ SNMP agent ต้องคอยติดต่อจาก management ซึ่งมี well-known UDP port ที่ 161 ถ้า SNMP manager ต้องการข้อมูลเกียวกับการจัดการ (Management information)ก็จะทำการร้องขอไปยัง UDP port ที่ 161 ของ Host ปลายทาง

ตารางที่ 9

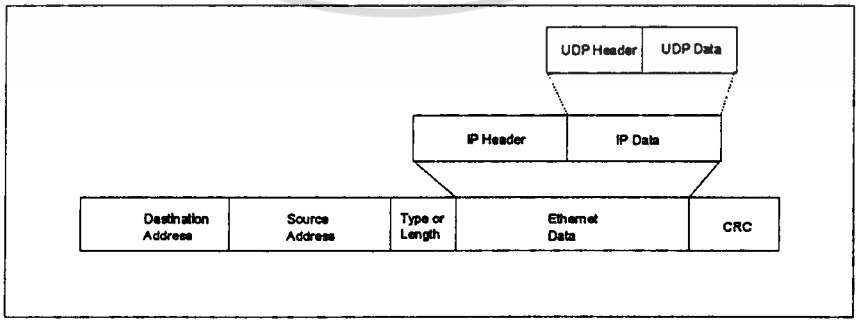
แสดงตัวอย่าง UDP Port Number

Decimal	Description
5	Remote Job Entry
7	Echo
9	Discard

ตารางที่ 9 (ต่อ)

11	Active User
13	Daytime
15	Who is up or Netstat
17	Quote of the Day
19	Character Generator
37	Time
39	Resouce Location Protocol
42	Host Name Server
43	Who is
53	Domain Name Server
67	Bootstrap Protocol Server
68	Bootstrap Protocol Client
69	Trivial Filter Transfer
79	Finger
111	Sun Microsystems' RPC
123	Network Time Protocol
161	SNMP Message
162	SNMP Trap

ภาพที่ 29



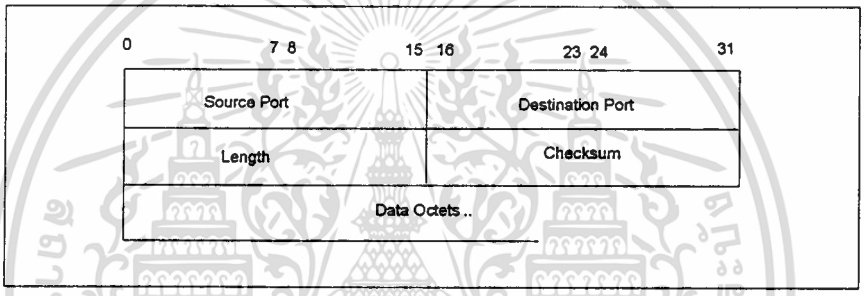
แสดง UDP Message ที่บรรจุอยู่ใน IP datagram

Ethernet layer : มีหน้าที่สำหรับการส่งผ่านข้อมูลระหว่าง Host หรือ Router บน Physical network เดียวกัน

IP layer : มีหน้าที่สำหรับส่งผ่านข้อมูล ผ่านไปยัง Router ระหว่าง Host ที่เชื่อมต่ออยู่ ต่างเครือข่ายกัน

UDP layer : มีหน้าที่สำหรับ Demultiplexing ไปยัง Multiple process ที่ อยู่บน Host ปลายทาง

ภาพที่ 30



แสดงถึง UDP Header Format

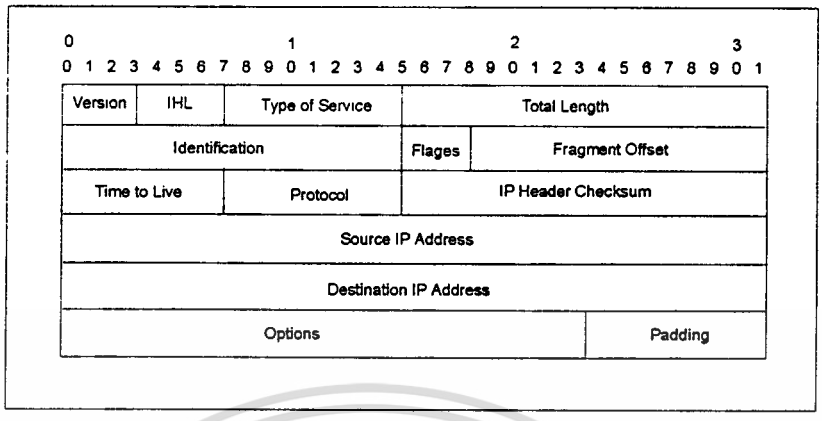
Source Port : (Option) แสดงถึง Port ของผู้ที่ส่งมา ใช้ในกรณีที่ต้องการให้ตอบกลับมา ถ้ากรณีที่ Host ผู้ส่งไม่ได้ให้ Source port ดังนั้น ฟิลด์ นี้ก็ควรกำหนดเป็น 0

Destination Port : เป็นส่วนที่ Demultiplexes datagram ไปยัง Processes

Length : จะเป็นส่วนที่แสดงถึงผลรวมของ UDP header และ Data อยู่ในรูปของอ็อกเต็ท

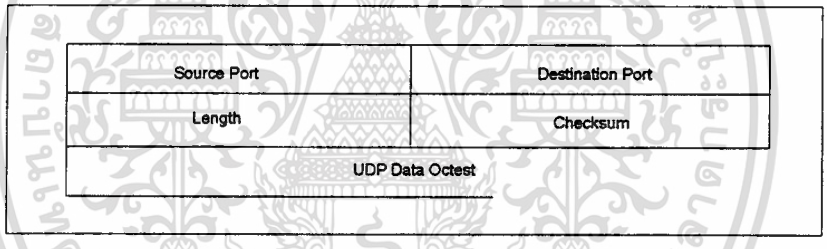
Checksum : (option) ควรจะกำหนดเป็น 0 เมื่อไม่ได้มีการคำนวณ checksum จากการที่ IP ไม่ได้คำนวณ checksum ในส่วนของส่วนข้อมูล (IP จะคำนวณ Checksum รวมถึง IP header เท่านั้น) UDP checksum ก็มีไว้สำหรับให้ตรวจสอบว่าข้อมูลที่ได้รับนั้นไม่ผิดพลาด

ภาพที่ 31



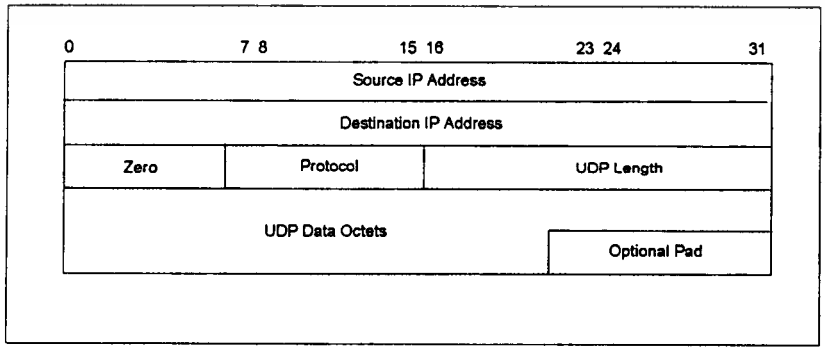
แสดงมาตรฐานของ IP Header

ภาพที่ 32



แสดงมาตรฐานของ UDP Header

ภาพที่ 33



แสดงถึง UDP Checksum Fields: Pseudo Header and Data Octets

Transmission Control Protocol (TCP)

TCP นั้นจะคุณสมบัติทางด้าน Reliable, Byte-stream-oriented, Virtual circuit protocol สามารถที่จะทำการการติดต่อสื่อสารระหว่าง Host ไปยัง Host บนเครือข่าย เป็นไปได้อย่างถูกต้องมีประสิทธิภาพดีเยี่ยม ซึ่งตัว TCP มีหน้าที่หลักดังนี้

- ทำหน้าที่ในการ Establishment และ termination ของ connection
- ดูแลและจัดการเกี่ยวกับ แพกเก็ต ที่รับส่งให้มีประสิทธิภาพ
- มีการจัดเรียงลำดับก่อนหลังของ แพกเก็ต ที่รับส่ง
- มี Flow control ในการป้องกัน Data over flow ไปยัง host
- มีการ Error recovery สำหรับ แพกเก็ต ที่สูญหาย (Lost) หรือที่ซ้ำกัน (Duplicated)
- สามารถแบ่งแยก (Demultiplex) ไปยัง Applications ต่าง ๆ ที่อยู่บน Host ได้

ตัวอย่าง application layer ที่ใช้ TCP

- Telnet
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)

TCP จะเป็น client ของ Internet Protocol(IP) โดยที่ IP นั้นจะเป็นตัวที่นำพาการรับส่งของข้อมูลต่าง ๆ ของ TCP หน้าที่หลักของ IP ที่กระทำต่อ TCP นั้นจะรวมถึง

- เกี่ยวกับ แอดเดรส ในการวิเคราะห์ถึง สเตชั่นต้นทางและ สเตชั่นปลายทางที่อยู่ต่างเครือข่ายกัน
- ความสามารถในการส่งผ่านส่วนข้อมูล ข้ามไปยัง Internet
- ความสามารถในการแบ่งแยก ส่วนข้อมูล สำหรับส่งผ่านไปยังเครือข่าย ที่มีแพกเก็ตขนาดเล็ก
- Time-to-live field ซึ่งเป็นส่วนที่จำกัดช่วงของเวลาของส่วนข้อมูล ที่อยู่บน Internet
- Type-of-Service field แสดงถึงลักษณะของการบริการที่ Router ควรจะจัดการให้

ส่วน Interface

ส่วนของ TCP interface นั้นจะอยู่ระหว่าง Application process และ Network layer protocol เมื่อ Application ต้องการที่จะส่งผ่านข้อมูล โดยการเรียก (Call) ไปยัง TCP และส่งผ่านบัฟเฟอร์ ของ โดยที่ข้อมูลของ TCP จะทำการรวบรวมข้อมูลที่รับมาไปเป็น เซกเมนต์ และจะทำการเรียกไปยัง IP module เพื่อส่งแต่ละ เซกเมนต์ ไปยัง Host ปลายทาง สำหรับผู้รับเมื่อได้รับแล้วก็จะนำข้อมูลออกจากเซกเมนต์ ส่งต่อไปยังบัฟเฟอร์ และเตือน (Notifies) ไปยัง Destination application process

ส่วน TCP Application Interface

ส่วนที่ติดต่อระหว่าง Application process และ TCP นั้นจะประกอบด้วยกลุ่มของ Function Calls ที่ถูกกำหนดมาอย่างดีแล้ว โดยจะเป็นมาตรฐานของระบบในการที่จะเปิด (Open), ปิด(Close), อ่าน(Read) หรือเขียน(Write) ไปยังไฟล์ function ต่าง ๆ เหล่านี้ สามารถทำให้ Application เอง สามารถที่จะเปิด หรือปิด Connection, ส่ง หรือ รับข้อมูล หรือ เก็บรักษา สถานะ (Status) ของ Connection ไว้ได้

ส่วนTCP Internet Interface

ส่วนที่สำหรับติดต่อระหว่าง TCP และ Lower level protocol ไม่ได้ถูกกำหนดไว้เป็นมาตรฐาน แต่ก็มีเพียงแต่วิธีการที่จะทำให้ levels ทั้งสองสามารถที่จะส่งผ่านข้อมูลระหว่างกันได้ การใช้ Function Call ในการส่ง และ รับส่วนข้อมูล ไปยัง Remote TCP Modules ที่อยู่บน Host บน Internet system ซึ่ง Function นั้นจะต้องมีข้อมูลต่าง ๆ สำหรับการส่งแอดเดรส, Type ของการบริการ (Service), Precedence, ความปลอดภัย (Security) และ ข้อมูลที่ใช้ในการควบคุมอื่น ๆ อีก โดยทั่วไปแล้วส่วนที่ติดต่อกับ Physical network นั้นจะถูกควบคุมด้วย Device driver โดยที่TCP เองไม่สามารถที่จะเรียก ไปยัง Network device driver ได้โดยตรง

TCP นั้นสามารถที่จะเรียก ผ่านไปยัง IP module เท่านั้น ส่วน IP module เองจะเป็นตัวที่เรียก ตรงไปยัง Device driver ได้

<u>หน้าที่หลักที่สำคัญของ TCP คือ</u>

1. Basic Data Transfer ส่วนที่เป็นพื้นฐานในการส่งผ่าน TCP Software ของ Host ทั้งสอง นั้นเรียกว่า เซกเมนต์ "Segment" TCP นั้นแสดงส่วนของข้อมูลเหมือนกับส่วนที่เรียงกันไปของไบท์ หรืออ็อกเต็ท ซึ่งจะถูกรวมอยู่ใน เซกเมนต์ นั้น สำหรับส่งผ่านข้อมูล แต่ละเซกเมนต์ นั้นจะส่งผ่านข้ามไปยัง Internet อยู่ในรูปของฟิลด์ข้อมูล ของ แต่ละ IP Datagram

2. Reliability TCP สามารถที่จะนำข้อมูลกลับมา (Recover Data) จากการถูกทำลาย, การสูญหาย, และการซ้ำกันของข้อมูล โดยที่ TCP สามารถกำหนดลำดับของ แต่ละอ็อกเต็ท (ไบท์) ที่ส่งออกไป และต้องการรับทราบการตอบกลับจาก TCP ปลายทาง ถ้าในกรณีที่ไม่ได้รับตอบกลับภายในช่วงเวลาที่กำหนดไว้ ข้อมูลก็จะถูกส่งกลับไปอีกครั้งโดยสถานีต้นทาง ส่วนสถานีปลายทาง ก็จะใช้หมายเลขลำดับ ในการตรวจสอบ เซกเมนต์ ที่ถูกต้อง และสามารถแยกแยะหรือกำจัด เซกเมนต์ ที่ซ้ำกันได้ โดยที่ข้อมูลที่ผิดพลาดสามารถตรวจสอบได้จาก ส่วน Checksum ของ แต่ละเซกเมนต์ เมื่อตรวจสอบเจอก็จะทำการปฏิเสธ เซกเมนต์ นั้น โดยจะไม่มีการตอบกลับไปยังสถานีต้นทาง ดังนั้นที่สถานีต้นทาง ก็จะทำการทวนการส่งกลับไปอีกครั้ง

3. Flow Control TCP มีวิธีการในการควบคุมข้อมูล จำนวนมากที่ส่งมาจาก สถานีต้นทาง มายัง สถานีปลายทาง โดยจะใช้ช่องหน้าต่างการรับ (receive window) เป็นตัวที่จะส่งค่าตอบกลับ (ACK) กลับไปเพื่อบอกว่าที่ปลายทาง สามารถที่จะบรรจุค่าอ็อกเต็ทได้เท่าไร ซึ่งก็เหมือนกับส่วนรองรับ (Receive buffer fill)

4. Multiplexing จะเหมือนกับ UDP โดยที่ TCP นั้นจะรวมกับ Port ในการที่จะไปถึงยังจุดหมายปลายทางที่อยู่ในเครื่องนั้น TCP นั้นจะให้กลุ่มของ Port ที่อยู่ในแต่ละ Host ใช้ในการทำงานเป็นหลาย ๆขบวนการ (multiple process) ภายใน Host ตัวเดียวกัน โดยการใช้การบริการของ TCP ในการติดต่อสื่อสาร

ส่วน Socket นั้น จะถูกสร้างขึ้นมาโดยการรวมกันระหว่าง Internet address ของ Host กับหมายเลข Port ของตัว Socket นั้นสามารถที่จะแยกแยะจุดหมายปลายทางของการรับส่ง TCP ทั้งหมด โดยคู่ของ Socket ที่เกิดขึ้นนั้นจะต้องไม่ซ้ำกัน ของแต่ละ ส่วนการติดต่อ

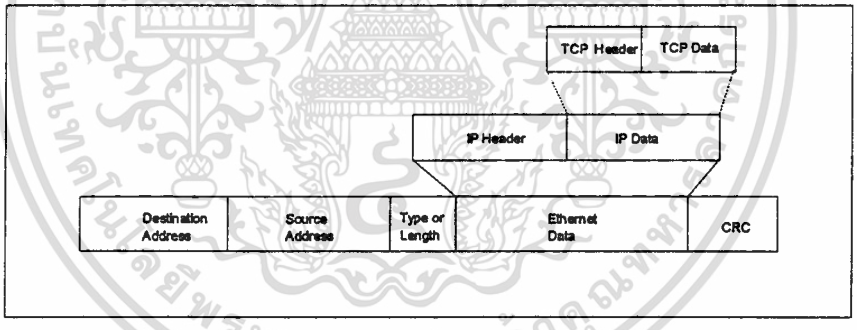
(Connection) อย่างไรก็ตาม Socket นั้นสามารถที่จะทำงานมากกว่าหนึ่ง ส่วนการติดต่อ (Connection)

5. Connection เป็นส่วนที่รวบรวมของข้อมูลต่าง ๆ คือ Socket number, Sequence number และ ส่วนที่ใหญ่สุดของช่องหน้าต่าง (window) โดยแต่ละ Connection นั้นจะต้องไม่ซ้ำกัน โดยที่แต่ละคู่ของ Socket จะแสดงถึงด้าน (Side) ที่เป็นของ Virtual circuit

เมื่อมี ขบวนการ (Process) 2 ขบวนการ ต้องการที่จะติดต่อกัน ขบวนการ TCP ของ Host แต่ละ Host ครั้งแรกนั้นจะต้องทำการสร้างส่วนการติดต่อขึ้นมา (Establish connection) ซึ่งกันและกัน โดยที่ขบวนการของการติดต่อ (Connection process) นี้จะเป็นส่วนที่ทำให้ เกิดการรับรู้ สถานะของข้อมูล สำหรับแต่ละด้าน ของ Virtual circuit หลังจากการแลกเปลี่ยนข้อมูล สิ้นสุดลง ส่วนของการติดต่อ (Connection) จะทำการตัดการติดต่อตัวเองลงเพื่อคืนสภาวะให้สำหรับการงานอื่น ๆ

แสดง TCP Message ที่สมบูรณ์ ซึ่งประกอบด้วย Header และข้อมูลที่อยู่ใน IP datagram โดยสามารถผ่านข้ามไปบน Internet

ภาพที่ 34



แสดงถึงโครงสร้างของ Ethernet Frame

Ethernet layer: แสดงถึงส่วนที่ผ่านข้อมูลระหว่าง Host ทั้งสอง หรือ Router บน Physical network เดียวกัน

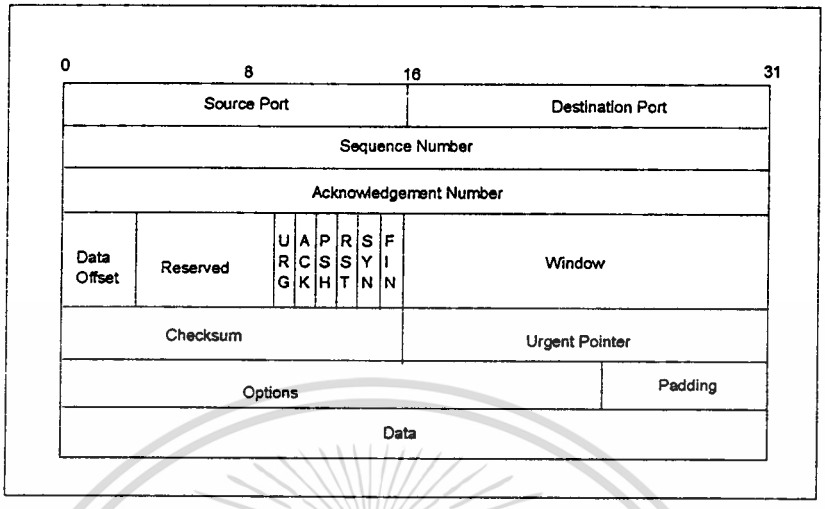
IP Layer: แสดงถึงส่วนที่ส่งผ่านข้อมูลข้าม Router ระหว่าง Host บน Internet

TCP Layer: แสดงถึงส่วนที่ทำหน้าที่ให้ความถูกต้องแม่นยำในการส่งไปยัง Application Layer

Source Port: แสดงถึงส่วนของ Application บน Host ที่เป็นส่วนแรก ในการส่งผ่านข้อมูล

Destination Port: แสดงถึงส่วนของ Application บน Host ที่ต้องส่งผ่านข้อมูลไปให้ถึง

ภาพที่ 35



แสดงถึง TCP Header Format

Sequence Number: เป็นส่วนแรกของข้อมูลอ็อกเต็ท ใน เซกเมนต์ (ยกเว้นเมื่อมี SYN) เมื่อมีค่าของ SYN ตัวหมายเลขลำดับ (Sequence number) นี้จะเป็นหมายเลขลำดับเริ่ม (initial sequence number (ISN)) และส่วนของข้อมูลอ็อกเต็ท ส่วนแรกจะเป็น ISN + 1

Acknowledgement number : จะมีก็ต่อเมื่อส่วนของ ACK Control บิท มีการกำหนด เอาไว้ ค่าของหมายเลข ACK จะอยู่ถัดจากหมายเลขลำดับ (Sequnce number) ใน เซกเมนต์ ของผู้ส่ง

Data offset : จะมีการกำหนดไว้ 32 บิท ใน TCP Header เพื่อแสดงถึงตำแหน่งเริ่มต้นของข้อมูลใน เซกเมนต์ ที่ต้องใช้ค่านี้ก็เพราะว่าในส่วนของ Option Field นั้นจะมีขนาดเปลี่ยนแปลง

Reserved: จะมีค่าเป็น 6 บิท ตามหลัง Data offset ฟิลด์ ใช้สำรองเอาไว้ โดยปกติมีค่าเป็น 0

Control Bits : เป็นส่วนทำงานร่วมกันสำหรับตรวจสอบใน หน้าที่พิเศษ

- URG - แสดงถึง Urgent Pointer ฟิลด์ เป็นส่วนที่สำคัญ
- ACK - แสดงถึง Acknowledgment ฟิลด์ เป็นส่วนที่สำคัญ
- PSH - Push Function
- RST - Reset the connection
- SYN - Synchronize sequence number
- FIN - No more data from sender

Window : เป็นหมายเลขของ อ็อกเต็ท ที่กำหนดโดย ผู้ส่ง

Checksum : ใช้ในการตรวจสอบ เซกเมนต์ ที่ทำการส่งโดยไม่มีข้อผิดพลาด ในกรณีข้อผิดพลาดเกิดขึ้น เซกเมนต์ นั้นจะไม่ถูกนำมาใช้อีก

Urgent Pointer : เป็นส่วนสำคัญใช้ในกรณีที่ URG Control bit ถูกกำหนดเอาไว้ค่านี้จะเป็น Positive offset จากหมายเลขลำดับ และแสดงถึงจุดสิ้นสุดของ Urgent data

Option : เป็นส่วนที่เป็น Option ของ TCP อาจใช้ในการเช่น แสดงค่าขนาดของเซกเมนต์สูงสุดของผู้ส่ง

Padding : เป็นหมายเลข 0 บิท สำหรับให้มั่นใจว่าเป็นส่วนสิ้นสุดของ Header บน 32-บิท Boundary

ตารางที่ 10

แสดงหมายเลข Port ต่าง ๆ

Decimal	Description
1	TCP Multiplexer
5	Remote Job Entry
7	Echo
9	Discard
11	Active Users
13	Daytime
15	Who is up ?
17	Quote of the day
19	Character Generator
20	File Transfer Protocol (data)
21	File Transfer Protocol (control)
23	Telnet
25	Simple Mail Transport Protocol
37	Time
39	Resource Location Protocol
42	Host Number Server
43	Who is
53	Domain Name Server
67	Bootstrap Protocol Server
68	Bootstrap Protocol Client

ตารางที่ 10 (ต่อ)

69	Trivial Filter Transfer
75	Any private dial-out server
77	Any private RJE Service
79	Who is on system
101	NIC Host Name Server
102	ISO-TSAP
103	X.400 Mail Service
104	X.400 Mail Sending
111	Sun Remote Procedure Call
113	Authentication Service
139	NETBIOS Session Service

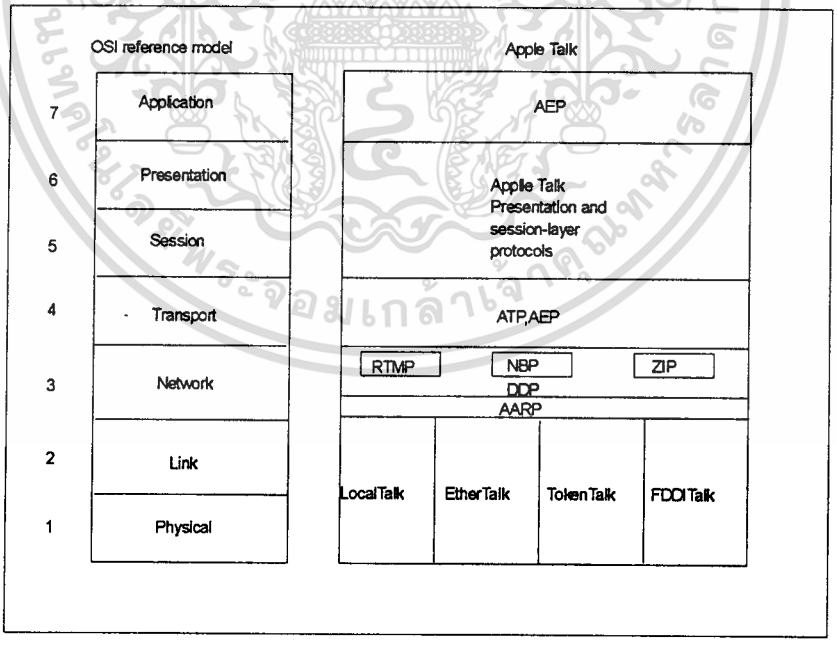
บทที่ 4

กลุ่มของโปรโตคอลที่ใช้งานทั่วไป

โปรโตคอล Apple Talk

Apple Talk นั้นได้มีส่วนที่สัมพันธ์กับ OSI model ซึ่งในภาพที่ 36 ได้แสดงถึงการเปรียบเทียบในระดับต่าง ๆ กับ OSI Layers

ภาพที่ 36



แสดงถึง Apple Talk และ OSI Reference Model

Media Access

Apple นั้นได้ถูกออกแบบมา โดยสามารถที่จะติดต่อกับ link layer หลายชนิดด้วยกัน ซึ่งรวมถึง Ethernet, Token Ring, FDDI และ Local Talk โดยที่ Apple จะทำการอ้างถึง Ethernet ไปเป็น EtherTalk, ส่วนของ Token Ring ก็จะเป็น Token Talk และ FDDI ก็จะเป็น FDDITalk

LocalTalk นั้นจะเป็นรูปแบบเฉพาะของ Apple เองที่พัฒนาขึ้นมาสำหรับ media-access system โดยทำงานบนสาย Shielded twisted-pair ที่ความเร็ว 230.4 กิโลบิทต่อวินาที และในช่วงความยาวใน เซกเม็นต์ นั้นสามารถยาวได้ถึง 300 เมตร สามารถรองรับจำนวนโนด (Node) มากที่สุดถึง 32 โนด

Network Layer

ในส่วนนี้จะแสดงถึงรายละเอียดและวิธีการของส่วน Network-layer ของ Apple Talk ซึ่งจะรวมถึงส่วนของ โปรโตคอล ในการกำหนด Address, การกำหนดเครือข่าย และการทำงานของโปรโตคอล Apple Talk นั้นที่เทียบ OSI Reference model ใน layer 3

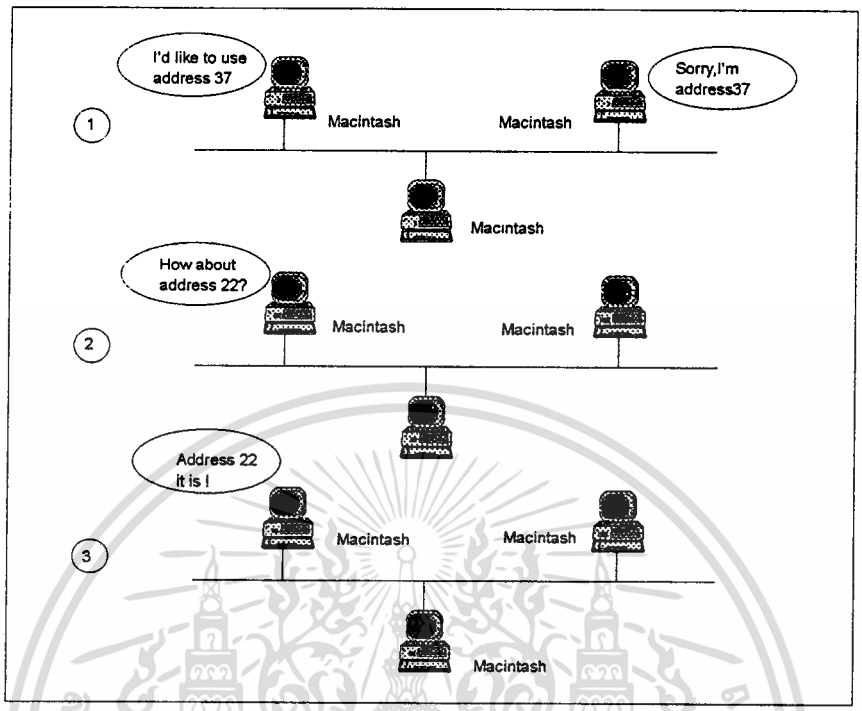
โปรโตคอลที่ใช้สำหรับการกำหนด Address

เมื่อเครื่อง Macintosh เริ่มทำงานนั้น Apple Talk ก็จะเริ่มทำการเลือก Address โดยการตรวจสอบ address ที่ตัวเองใช้กับ Address อื่น ๆ ใน Network ถ้าไม่ซ้ำกับเครื่องอื่นก็ถือว่า Address นั้นสมบูรณ์ แต่ถ้าในกรณีที่ Address นั้นมีอยู่แล้วในเครือข่าย โนดที่เป็นเจ้าของ Address ก็จะทำการส่งข้อความกลับไป เพื่อให้โนด ใหม่นั้นทำการเลือก Address ใหม่ ซึ่งรูปแบบของการกำหนด Address นั้นสามารถแสดงได้ในภาพที่ 37 โดยที่จะใช้ Apple Talk Address Resolution Protocol (AARP) เป็น โปรโตคอล สำหรับการกำหนด Address

โปรโตคอล Decnet

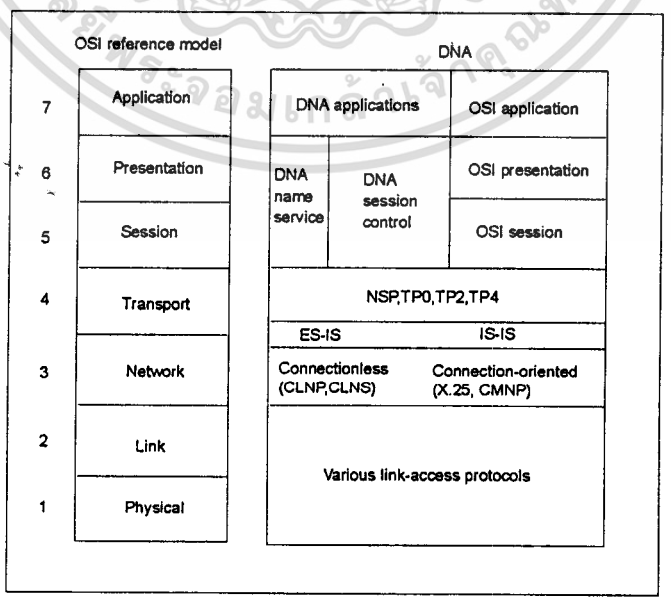
เป็นโปรโตคอลที่ทางบริษัทดิจิตอลได้พัฒนาขึ้นมาเพื่อใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์ ในเวอร์ชั่นแรกของ DECnet นั้นได้ออกมาในปี 1975 ต่อมาได้มีการพัฒนาใหม่เรียกเป็น Phase V และสามารถแสดงรูปแบบของ Digital Network Architecture (DNA) เปรียบเทียบกับ OSI Reference Model

ภาพที่ 37



แสดงขั้นตอนของการเลือก Address ของ Apple Talk

ภาพที่ 38



แสดงการเปรียบเทียบระหว่าง DNA และ OSI Reference Model

โปรโตคอล Netware

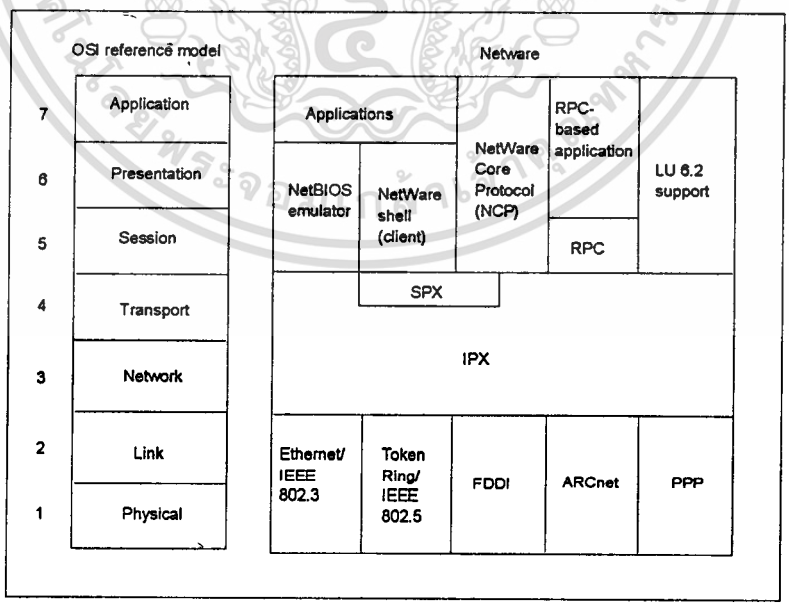
Netware นั้นก็คือ Network operating system (NOS) ตัวหนึ่งและสามารถที่จะรองรับการบริการในกลุ่มได้ Netware นี้ได้ถูกสร้างขึ้นมาโดยบริษัท Novell และนำออกสู่ท้องตลาดในช่วงต้นปี 1980 ซึ่งในขณะนั้นเองยังมีกลุ่มผู้ที่ใช้ Network นั้นยังเป็นกลุ่มเล็ก ๆ ต่อมาจึงทำให้การติดต่อสื่อสารบนคอมพิวเตอร์ส่วนบุคคลเป็นที่นิยมมากขึ้น เทคโนโลยีของ Netware ส่วนใหญ่แล้วถูกพัฒนามาจาก Xerox Network System (XNS)

ในส่วนของ NOS นั้น Netware ได้ถูกกำหนดไว้เป็นจำนวน 5 Layers ของ OSI reference model โดยที่สามารถที่จะบริการเกี่ยวกับการใช้แฟ้มข้อมูลและปริ้นเตอร์ร่วมกันได้ สามารถใช้งานกับ Applications ได้หลายอย่าง เช่น จดหมายอิเลคทรอนิกส์ การจัดการฐานข้อมูล และบริการอื่น ๆ Netware ได้ถูกออกแบบโดยมีการทำงานเป็น Client-server architecture

Network Layer

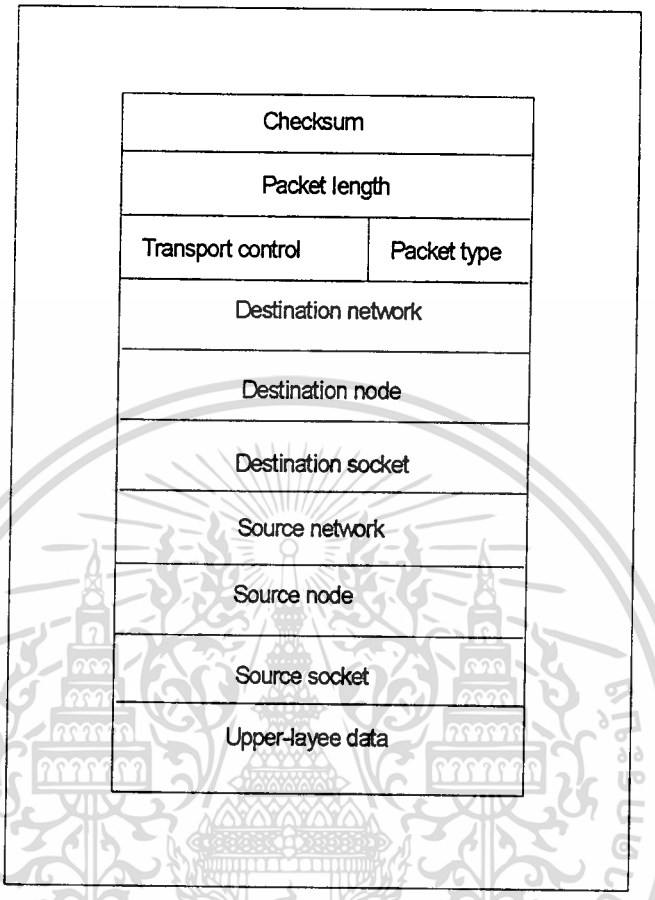
Internet Packet Exchange (IPX) เป็นโปรโตคอล สำหรับ Novell เมื่ออุปกรณ์ต้องการที่จะติดต่อสื่อสารผ่านไปยังเครือข่าย อื่นๆ ที่อยู่ต่างกันโปรโตคอล IPX ก็จะเป็นตัวผ่านข้อมูลต่าง ๆ ไปยัง เครือข่ายอื่น

ภาพที่ 39



แสดงถึงความสัมพันธ์กันระหว่าง โปรโตคอล Netware
เมื่อเทียบกับ OSI Reference model

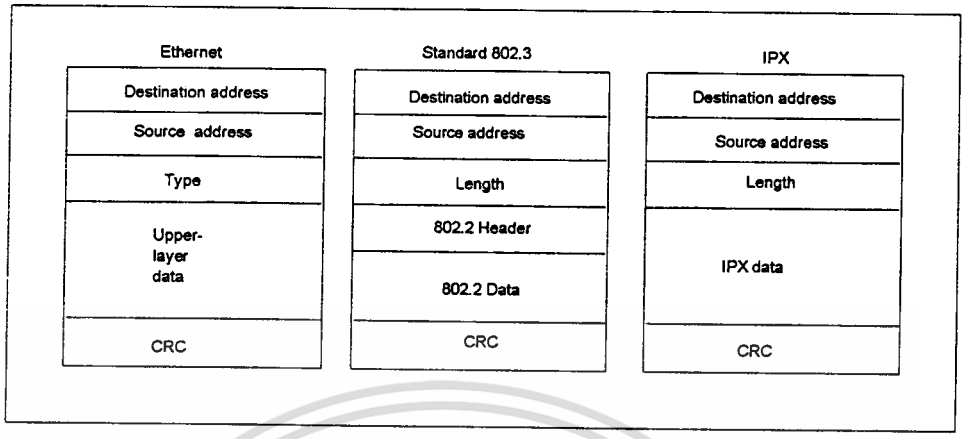
ภาพที่ 40



แสดงถึงรูปแบบของ IPX แพกเก็ต

Checksum	มีขนาดเท่ากับ 16-bit
Length	มีขนาด 16 บิท เพื่อแสดงถึงขนาดของส่วนของข้อมูล IPX
Transport control	มีขนาด 8 บิท แสดงถึงจำนวนของ Router ที่ แพกเก็ต นั้นผ่านมา
Packet type	มีขนาด 8 bit แสดงถึงชนิดของการรับข้อมูล ถ้าค่าเป็น 5 จะแสดงถึง Sequenced packet exchange (SPX) ถ้าค่าเป็น 17 จะแสดงถึง Netware core protocol (NCP)
Destination address	นั้นจะประกับไปด้วย 3 ฟิลด์ จะบอกถึงเครือข่ายปลายทาง (Destination network), Host, และ Socket
Source address	ก็จะแสดงถึงเครือข่ายต้นทาง(Source network), Host และ Socket
Data field	ก็จะแสดงถึงข้อมูลสำหรับ Upper-layer processes

ภาพที่ 41

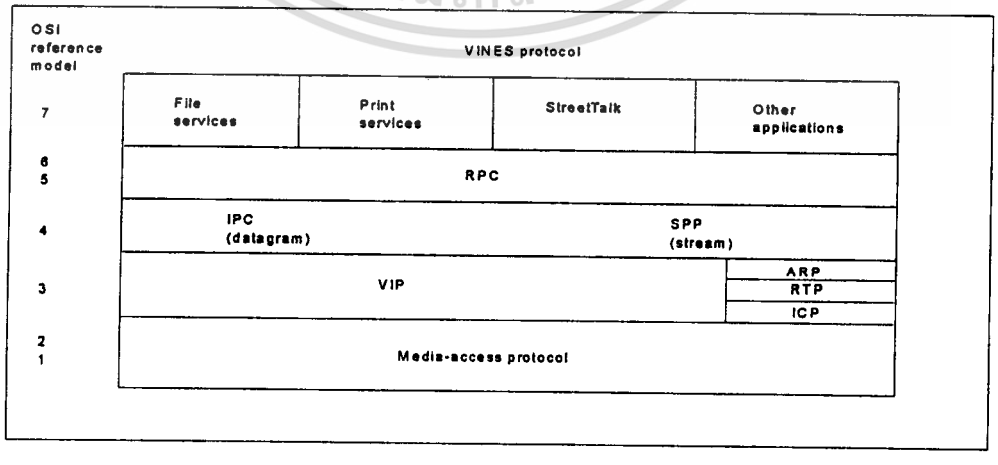


แสดงถึงการเปรียบเทียบรูปแบบของ Ethernet, IEEE 802.3
และ IPX Encapsulation Format

โปรโตคอล Banyan VINES

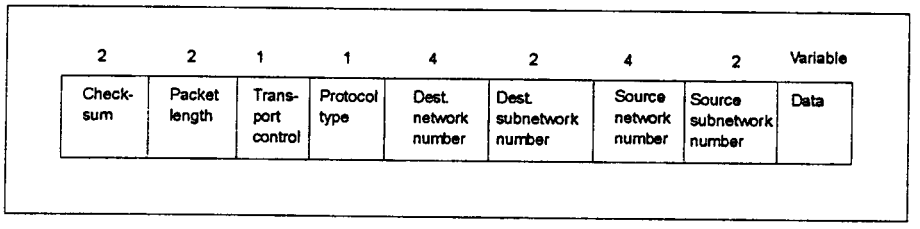
ระบบเครือข่ายของ Banyan Virtual Network System ได้ถูกพัฒนาขึ้นมาจาก โปรโตคอล Xerox's Network System (XNS) โดยใช้ระบบการกระจาย (Distributed System) ที่ยินยอมให้สามารถแลกเปลี่ยนข้อมูลข่าวสารระหว่างตัว Clients กับตัว Servers ได้

ภาพที่ 42



แสดงความสัมพันธ์กันระหว่าง VINES Protocol Stack
เมื่อเทียบกับ OSI Reference Model

ภาพที่ 43



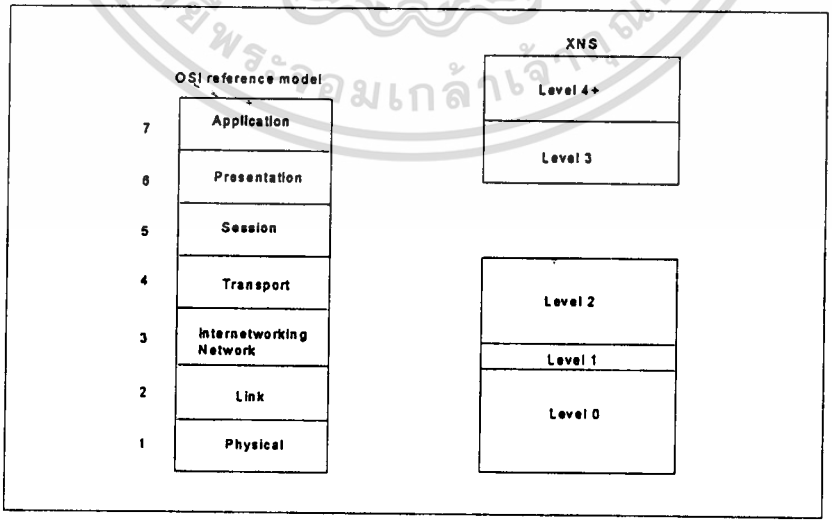
แสดงถึงรูปแบบ แพกเก็ต ของ VIP

โปรโตคอล Xerox Network System (XNS)

โปรโตคอล Xerox Network System (XNS) ได้ถูกพัฒนาขึ้นมาจากบริษัท Xerox เองในปลายปี 1980 และ XNS เองได้ถูกเลือกเป็นรูปแบบของโปรโตคอลเป็นมาตราฐาน ที่บริษัทต่าง ๆ ได้นำไปพัฒนาเพิ่มเติม เช่น บริษัท Novell, Ungermann-Bass และ 3Com

การออกแบบของ XNS นั้นได้มีบางส่วนที่แตกต่างไปจาก OSI reference model โดยสามารถเปรียบเทียบได้จากภาพที่ 44

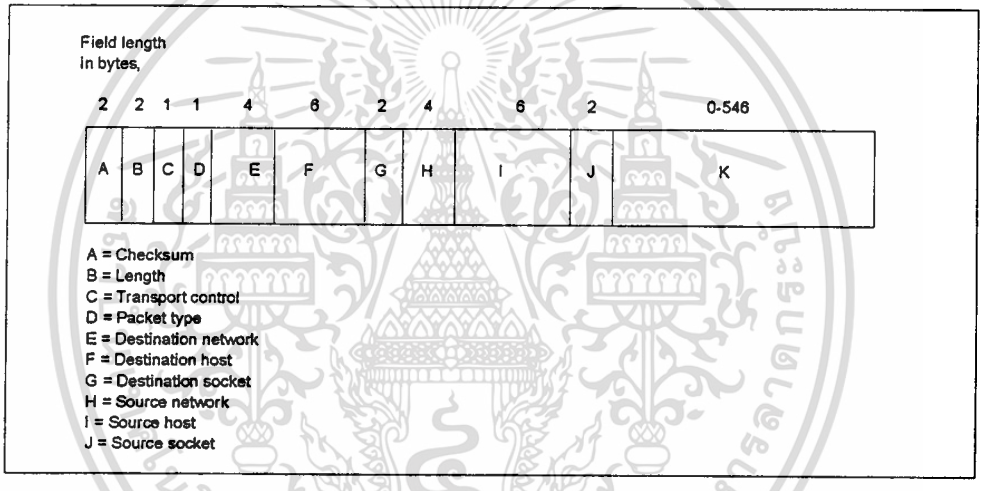
ภาพที่ 44



แสดงการเปรียบเทียบระหว่าง XNS กับ OSI Reference Model

จากรูปนั้นจะเป็นได้ว่า XNS จะประกอบด้วยจำนวน 5 ระดับ (Level) โดยที่ระดับ 0 นั้นจะเทียบได้กับ OSI ในชั้นที่ 1 และ 2 ซึ่งจะจัดการเกี่ยวกับ Link access กับ Bit stream ส่วนระดับที่ 1 นั้นจะเทียบได้กับ OSI ในชั้นที่ 3 โดยทำหน้าที่จัดการเกี่ยวกับการจราจรของเครือข่าย ระดับที่ 2 จะเทียบได้กับ OSI ในชั้นที่ 3 โดยจัดการเกี่ยวกับเส้นทางเครือข่ายภายนอก (Internetwork routing) และ OSI ในชั้นที่ 4 นั้นจะจัดการเกี่ยวกับการติดต่อสื่อสาร ส่วนระดับที่ 3 และ 4 จะอยู่เหนือขึ้นไป จะจัดการเกี่ยวกับโครงสร้างของข้อมูล, ขบวนการต่าง ๆ และเกี่ยวกับทางด้าน Application โดยที่ XNS จะไม่มีส่วนที่จัดการเหมือนกับ OSI ในชั้นที่ 5 XNS นั้นได้มีชื่อเรียกอีกอย่างหนึ่งว่า Internet Datagram Protocol (IDP)

ภาพที่ 45



แสดงถึงรูปแบบของแพกเก็ต IDP

บทที่ 5

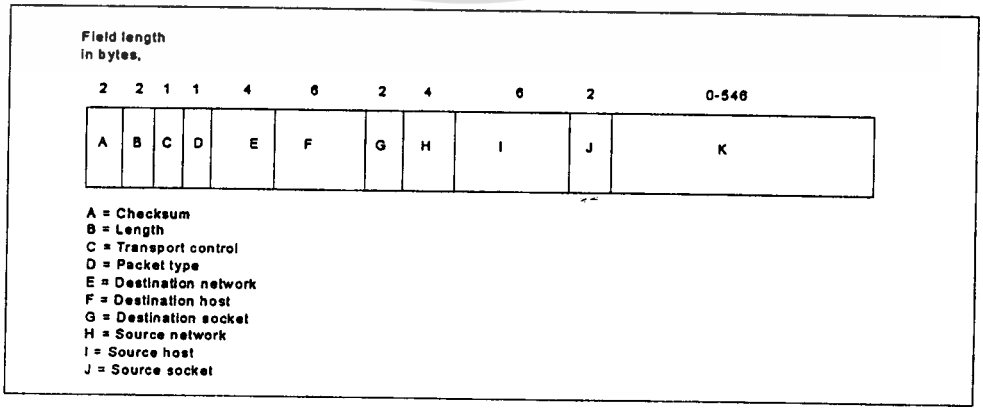
กลุ่มของ Routing โปรโตคอล

โปรโตคอล RIP (Routing Information Protocol)

Routing Information Protocol (RIP) เป็นโปรแกรมที่ใช้ในการส่งผ่านไปตามระยะทาง "routed" ซึ่งพัฒนามาจาก 4.3 Berkeley UNIX ซึ่งในขณะนั้นยังไม่มีมาตราฐานเดียวกับ Routing โปรโตคอล จึงทำให้ RIP กลายมาเป็นโปรโตคอลมาตรฐาน โดย RIP นั้นจะใช้วิธีการที่เรียกว่า "Distance-Vector Algorithms" ซึ่งเป็นวิธีการเดียวกับที่ใช้ใน Interior Gateway Protocol (IGP)

RIP นั้นจะถูกสร้างขึ้นมาโดยอาศัยการบริการของ USER Datagram Protocol (UDP) โดยที่แต่ละ Router จะใช้ RIP ในการรับส่งผ่านข้อมูลอยู่บนส่วน UDP หมายเลขที่ 520 ข้อความที่ RIP ใช้ส่งนั้นรวมถึง Header และ ข้อมูล (Data) จะถูกบรรจุอยู่ในส่วนที่เป็นส่วนข้อมูล ของ แพกเก็ต UDP โดยที่ส่วนของ UDP นั้นจะรวมถึง Header และ ข้อมูล นั้นจะถูกบรรจุอยู่ในส่วนข้อมูล IP (IP datagram) อีกที โดยจะใช้ Ethernet layer เป็นส่วนที่ใช้สำหรับส่งผ่านข้อมูลระหว่าง Host หรือ Router บน Physical Network เดียวกัน

ภาพที่ 46



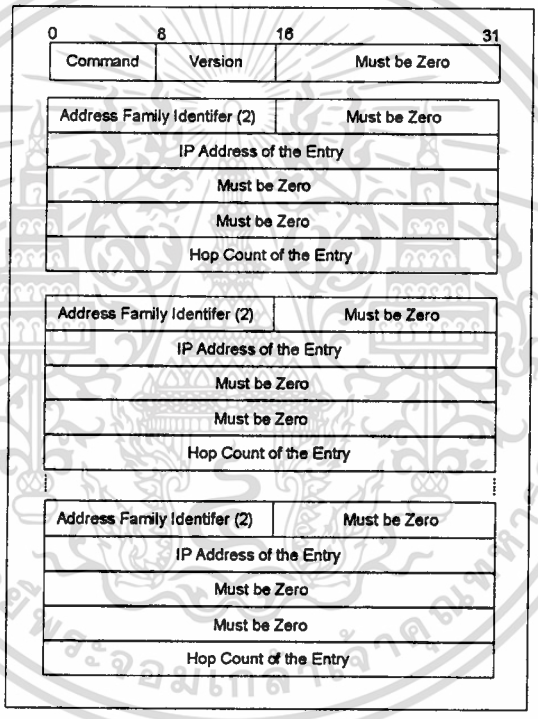
แสดงรูปแบบของ RIP Frame Encapsulation

โครงสร้างและรูปแบบของ RIP Header Format

ข้อความ (Message) ของ RIP นั้น จะประกอบไปด้วย Header ซึ่งจะเป็นรายละเอียดของ เครือข่าย ต่าง ๆ โดยอาจมีการส่งมาจาก Router เอง

ค่าสัดส่วน (Portion) ของส่วนของข้อมูลจากฟิลด์ ของ "Address Family Identifier" ไปยัง ฟิลด์ ของ "Hop Count of Entry" นั้นอาจมีถึง 25 ครั้ง ซึ่งจะยอมให้แต่ละข้อความ ของ RIP นั้นสามารถบรรจุได้ถึง 25 Address จะมีขนาด 500 อ็อกเต็ท โดยที่ขนาดที่มากที่สุดของส่วนของข้อมูล (Datagram) ของ RIP นั้นจะเป็น 512 อ็อกเต็ท โดยไม่รวมถึง Header ของ IP และ UDP

ภาพที่ 47



แสดงรูปแบบของ RIP Header Format

ค่าของฟิลด์ ใน RIP ประกอบไปด้วย

Command: ฟิลด์นี้แสดงถึงจุดมุ่งหมายของ ส่วนของข้อมูล โดยที่ Command นี้จะสร้างทั้ง RIP Request(1), หรือ RIP Response(2) ส่วนที่เป็นชนิดอื่น ๆ นั้นถูกยกเลิกการใช้งานแล้ว

Version : จะแสดงถึงเวอร์ชั่นของโปรโตคอล ที่ใช้งาน

Address Family Identifier : แสดงถึงความแตกต่างของโปรโตคอล อย่างเช่น ถ้าเป็น IP จะมีค่าเท่ากับ 2

IP Address : จะแสดงถึง IP Address อาจจะเป็น หมายเลขเครือข่ายของ Host (Host address network number), หมายเลขเครือข่ายย่อย (Subnet number) หรือ Zero เพื่อแสดงค่าเส้นทางหลัก (Default route) โดยตัว RIP แพกเก็ตนั้นไม่สามารถที่จะแยกชนิดต่าง ๆ ของ Addresses ได้

Must Be Zero : จะเป็นส่วนหนึ่งของ Address ฟิลด์ แต่จะไม่ได้ใช้เพราะ IP address นั้น จะมีจำนวนเพียง 4 อ็อกเต็ท ดังนั้นส่วนที่ไม่ได้ใช้ ก็ถูกแทนด้วยค่า 0

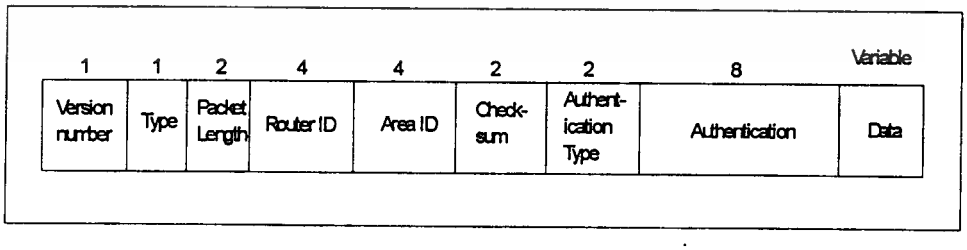
Hop Count : แสดงถึงค่าระยะห่าง (Metric) ปัจจุบันของ Network address ถ้าเป็นค่า 16 จะแสดงถึงค่าเครือข่ายที่ไม่สามารถติดต่อได้

โปรโตคอล OSPF (OPEN SHORTEST PATH FIRST)

OSPF เป็น Routing โปรโตคอล ที่พัฒนาขึ้นมาสำหรับ Internet Protocol (IP) เหตุผลที่ทำให้มีการพัฒนา OSPF ขึ้นมาก็คือ Routing Information Protocol (RIP) ซึ่งถูกพัฒนามาในช่วงกลางของปี 1980 นั้น ในปัจจุบันไม่สามารถที่จะรองรับ เครือข่ายภายนอก (Internetwork) ที่มีขนาดใหญ่ได้

โครงสร้างของ OSPF แพกเก็ตนั้น จะประกอบด้วย ส่วนหัว จำนวน 24-ไบท์ แสดงได้ในภาพที่ 48

ภาพที่ 48

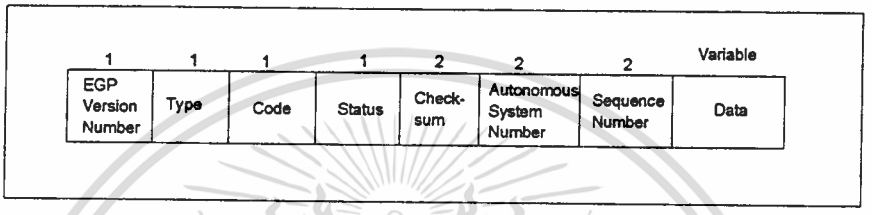


แสดงรูปแบบของ OSPF Header Format

โปรโตคอล EGP (Exterior Gateway Protocol)

EGP เป็น Routing โปรโตคอล อีกชนิดขึ้นที่ถูกพัฒนาขึ้นมาจาก ARPANET มีรูปแบบของแพกเก็ต ดังนี้

ภาพที่ 49

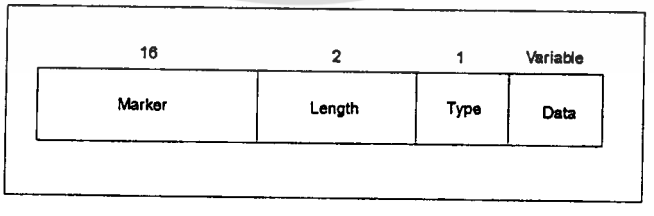


แสดงถึง EGP Packet Format

โปรโตคอล BGP (Border Gateway Protocol)

BGP เป็น Rounting โปรโตคอล ที่พัฒนาเพิ่มเติมจาก EGP มีรูปแบบของ แพกเก็ต แสดงดังภาพที่ 50 ซึ่งจะประกอบด้วย ส่วนหัว จำนวน 19 ไบท์

ภาพที่ 50



แสดงถึงรูปแบบของ BGP Packet Format

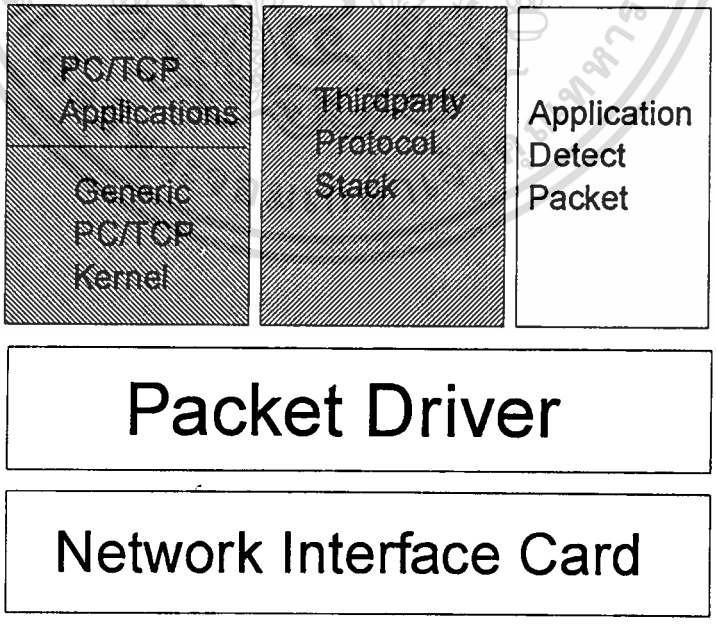
บทที่ 6

การออกแบบโปรแกรม

การใช้งาน แพกเก็ต Driver

แพกเก็ต Driver เป็นมาตรฐานหนึ่งที่บอกถึงการที่จะ Load Software driver ภายใต้ DOS ได้โดยจะเป็นตัวจัดการเกี่ยวกับซอฟท์แวร์ ที่จะติดต่อผ่านไปยังฮาร์ดแวร์ เช่น Ethernet Card, IEEE 802.5 Token-Ring Card ซึ่งจะเป็นไปตามมาตรฐานของแพกเก็ต Driver ที่กำหนดเอาไว้ และการใช้แพกเก็ต Driver นี้ สามารถที่ใช้โปรโตคอล Stack ทำให้โปรโตคอลแต่ละตัวสามารถทำงานได้โดยเป็นอิสระภายใต้ Network Interface Card เดียวกันดังภาพที่ 51

ภาพที่ 51



แสดงการใช้ แพกเก็ต Driver กับ Protocol Stack

แพกเก็ต Driver นั้นจะทำงานเช่นเดียวกับ TSR Module หรือ DOS Device Driver โดยที่จะ Load ไปยังหน่วยความจำ และมี Software Interrupt Vector อยู่ในช่วง OX60-OX80 และเมื่อ ต้องการที่จะติดต่อกับแพกเก็ต Driver ก็สามารถที่จะเรียกผ่าน Interrupt Vector ที่กำหนดไว้ได้ ซึ่งตัว Networking Application เมื่อต้องการจะติดต่อกับ Network ต่าง ๆ ในการรับเข้า (Input) หรือ การส่งออก (Output) ก็ตาม ก็จะให้แพกเก็ต Driver เป็นตัวจัดการให้ใน Application Program ต้องการรู้เพียงโครงสร้างของ Frame สำหรับ ชนิดของ Protocol Stack นั้น ๆ เช่น IPX, IP, XNS หรือ OSI แต่ไม่ต้องรู้ถึง ข้อกำหนดของ Interface Card ที่ใช้ เช่น I/O Port Hardware Interrupt หรือ รายละเอียด อื่น ๆ ของ Hardware ดังนั้น ตัว แพกเก็ต จะเป็นตัวจัดการเอง

Programming Interface

ในการที่จะเขียนโปรแกรมติดต่อกับ Packet Driver ได้นั้น ต้องรู้จัก ฟังก์ชั่น มาตราฐานต่าง ๆที่กำหนดไว้ โดยที่ ฟังก์ชั่น นั้นสามารถเรียกใช้ผ่านไปยัง Interrupt Vactor ที่ใช้งาน ตัวPacket Driver จะส่งค่างต่าง ๆ ที่ต้องการกลับมาให้ ส่วนภาษาที่ใช้ในการเขียน Program นั้นจะใช้ Pascal เป็นส่วนที่ทำหน้าที่ แสดงผลข้อมูลโดยที่จะมีส่วนของ Assembly ที่ใช้ในการเรียกใช้ ฟังก์ชั่น ต่าง ๆ ของแพกเก็ต Driver

รายละเอียดของฟังก์ชั่น ที่ใช้ในการตรวจจับ แพกเก็ต

1. Driver_info( ) จะเป็น ฟังก์ชั่น เมื่อเรียกใช้ จะให้ รายละเอียดเกี่ยวกับ Interface กลับมาโดยเราจะต้อง กำหนดให้ Register AH = 1 และ AL = 255

```
driver_info (handle)        AH == 1, AL==255
                int         handle: BX              /* Option */
        error return
                carry flag set
                error code              DH
                possible errors :
```

BAD_HANDLE /* older drivers only */

non_error return :

carry flag clear

version BX

class CH

type CL

name DS : SI

functionality AL

1 == basic functions present

2 == basic and extended present

5 == basic and high-performance

6 == basic, high-performance, extended

225 == not installed

2. Access_type ( ) เป็น ฟังก์ชั่น ที่ทำหน้าที่ในการกำหนดค่าเริ่มแรกที่ จะเริ่มใช้ Packet Driver ซึ่งจะมีการกำหนดค่าผ่านฟังชันที่ AH = 2

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

AH == 2

int	if_class ;	AL
int	if_type ;	BX
int	if_number ;	DL
char far	*type ;	DS : S1
00unsigned	typelen ;	CX
int	(for *receiver) ( ) ;	ES : DI

error return :

carry flag set

error code DH

possible errors :

NO_CLASS

NO_TYPE

NO_NUMBER

```
        BAD_TYPE
        NO_SPACE
        TYPE_INUSE
non_error return :
        carry flag clear
        handle                  AX
receiver call :
        (*receiver)(handle, flag, len [,buffer])
                int     handle ; BX
                int     flag ;          AX
                unsigned len ;          CX
        if AX == 1
                char for * buffer ; DS:SI
```

- type จะเป็นค่าของตำแหน่ง pointer ที่ระบุถึง packet type ส่วนค่าตัวแปร typelen จะบอกถึงขนาดความยาวของ type

- receiver จะแสดงตำแหน่ง pointer ของโปรแกรมย่อย ซึ่งจะทำงานเมื่อมีการรับ แพกเก็ต และในกรณีที่กำหนดค่าใน typelen = 0 จะเป็นการบอกถึงว่าต้องการรับทุก แพกเก็ต ที่เข้ามา ในการรับ แพกเก็ต นั้น โปรแกรมย่อย receiver จะมีการเรียกใช้งาน 2 ครั้ง ซึ่งครั้งแรกนั้นจะเรียกใช้งานพร้อมกับให้ค่า Register AH = 0 และค่า Register CX จะบอกถึงขนาดของ Frame ซึ่งรวมค่าของ destination, source และ type field เข้าไปด้วย เมื่อได้รับค่า AH = 0 แล้วตัว Application ต้องส่งค่าตำแหน่งของ Pointer ของ Buffer ไปยัง ES:D1 เพื่อเตรียมรับข้อมูลแต่ถ้าไม่ต้องการรับข้อมูลนี้ก็ให้ส่งค่า 0:0 ไปยัง ES:DS และตัว Packet Driver ก็จะไม่ทำครั้งที่สอง

การเรียกใช้งานครั้งที่ 2 นั้นจะให้ค่า AX = 1 ให้เพื่อบอกให้ทราบว่าขณะนี้ ได้ทำการ Copy ข้อมูลไปยัง Buffer เรียบร้อยแล้ว และ Application สามารถนำข้อมูลใน Buffer ไปใช้งานได้

3. Set_rcv_mode ( ) เป็น ฟังก์ชั่น ที่กำหนด Mode ในการรับข้อมูล โดยมีการกำหนด AH = 20

```
extended driver function
set_rcv_mode(handle, mode) AH == 20
        int     handle ;        BX
```

int mode ; CX

error return :

carry flag set

error code DH

possible erors :

BAD_HANDLE

BAD_MODE

non_error return :

carry flag clear

mode meaning

1. turn off receiver
2. receive only packets sent to this interface
3. mode 2 plus broadcast packets
4. mode 3 plus limited multicast packets
5. mode 3 plus all multicast packets
6. all packets

4. Terminate ( )เป็น ฟังก์ชั่น ที่ใช้เมื่อสิ้นสุดการทำงานและจะส่ง Memory ที่ใช้งานคืนสู่ระบบเดิม และจะมีค่า AH = 5

terminate(handle) AH == 5

int handle ; BX

error return :

carry flag set

error code DH

possible errors :

BAD_HANDLE

CANT_TERMINATE

non_error return :

carry flag clear

5. Release_type( ) ฟังก์ชั่นนี้จะเป็นการสิ้นสุดการติดต่อกับแพกเก็ต โดยจะมีความสัมพันธ์กับค่าของ handle ที่มาจากฟังชั่นของ access_type()

int release_type(handle) AH == 3

```
    int handle;              BX
error return :
    carry flag set
    error code               DX
    possible errors :
      BAD_HANDLE
    non-error  return:
      carry flag clear
```

6. Send_pkt( ) ในการส่งข้อมูลนั้นขนาดของจำนวนไบท์ ของข้อมูลนั้นจะเริ่มที่ส่วนสำรอง (Buffer) และตัว Application ต้องเป็นตัวจัดการข้อมูลใน แพกเก็ต นี้ ซึ่งจะรวมถึงส่วนหัว (Header) ของ เครือข่ายภายในด้วย

```
int send_pkt(Buffer,length)  AH == 4
    char far  *buffer;       DS:SI
    unsigned   length;        CX
error return
    carry flag set
    error code                DH
    possible errors:
       CANT_SEND
non=error return
    carry flag clear
```

7. Get_address( ) หน้าที่ของฟังก์ชั่นจะเป็นการคัดลอก address ของส่วนติดต่อ (Interface) ไปยัง Buf ขนาดของไบท์ ใน Buf จะถูกส่งมาแสดงใน CX ถ้าในกรณีที่มีการผิดพลาดคือแสดง no_space ก็จะแสดงถึงขนาดของค่า Len ไม่เพียงพอที่จะรองรับ Address ของส่วนติดต่อได้ ดังนั้นถ้าต้องการเปลี่ยน Address ก็สามารถใช้ฟังก์ชั่นของ Set_address( ) ทำการเปลี่ยนแปลงแก้ไขได้

```
get_address(handle,buf,len)   AH == 6
  int handle;                  BX
  char far   *buf;             ES:DI
  int          len;            CX
```

error return:
carry flag set
error code DH
possible errors:
BAD_HANDLE
NO_SPACE
non-error retrun:
carry flag clear
length CX

8. Reset_interface( ) ฟังก์ชั่นของการจัดค่าใหม่ของส่วนติดต่อ โดยจะมีความสัมพันธ์กับค่า Handle จะทำให้ยกเลิกขบวนการรับส่งข้อมูล และจะมีการกำหนดค่าเดิมของส่วนรับข้อมูล (Receiver) โดยจะกำหนดโหมดของส่วนรับข้อมูลใหม่เป็น 3 (Own address & broadcast)

reset_interface(handle) AH == 7
int handle BX
error return:
carry flag set
error code DX
possible errors:
BAD_HANDLE
CANT_RESET
non-error return:
carry flag clear

9. Get_rcv_mode( )เป็นฟังก์ชั่นที่จะให้ค่าของโหมดการรับข้อมูลโดยจะมีความสัมพันธ์กับ Handle

get_rcv_mode(handle,mode) AH == 21
int handle; BX
error return:
carry flag set
error code DH

```
        possible error:
            BAD_HANDLE
    non-error  return:
        carry flag clear
        mode                    AX
```

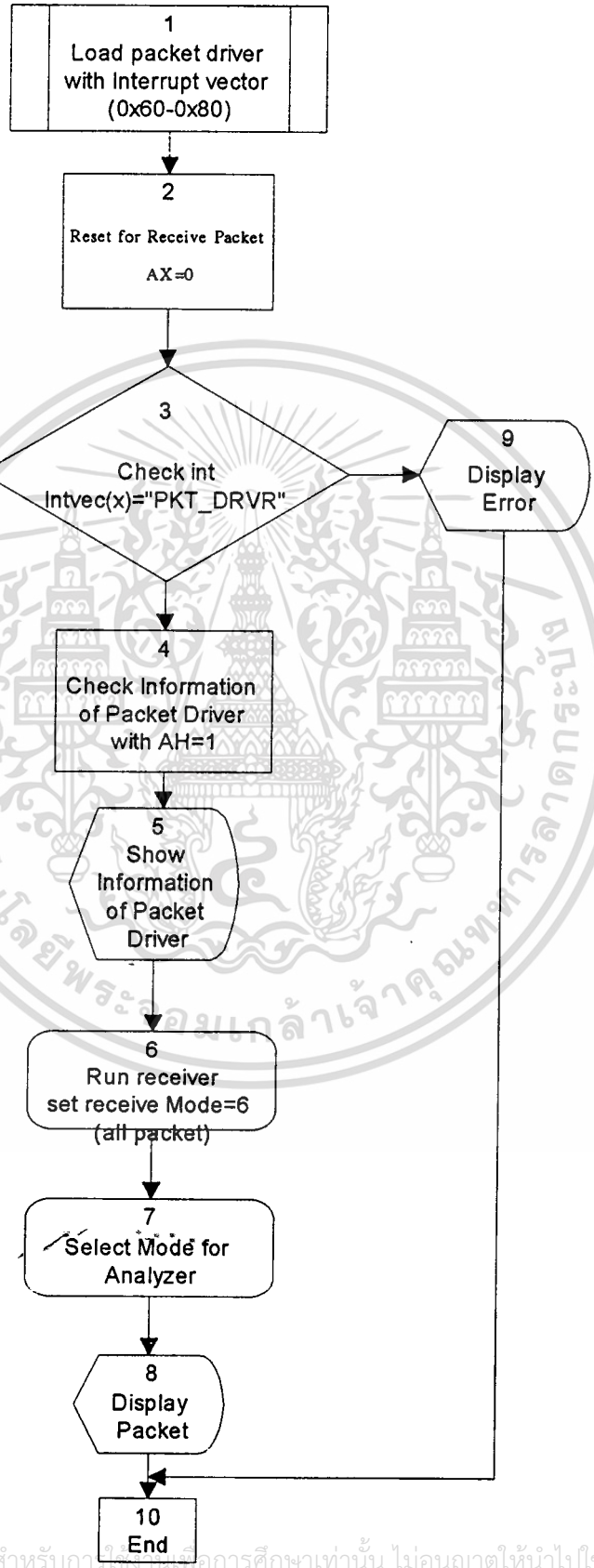
10. Set_address( ) ฟังก์ชั่นนี้จะใช้เมื่อในส่วนของ Application หรือตัว โปรโตคอล ต้องการที่จะใช้ Address พิเศษ เช่น โปรโตคอล DECnet ในกรณีที่มีข้อผิดพลาดมีการแสดงค่า BAD_ADDRESS จะหมายถึงตัว Packet Driver นั้นไม่สามารถทำงานได้ ซึ่ง Address นั้นอาจมีขนาดมากไปหรือน้อยไปก็ได้

```
set_address(addr,len)          AH == 25
    char far *addr;            SE:DI
    int len;                   CX
error return
    carry flag set
    error code                 DX
    possible errors:
      CANT_SET
      BAD_ADDRESS
non-error return:
    carry flag clear
    length                     CX
```

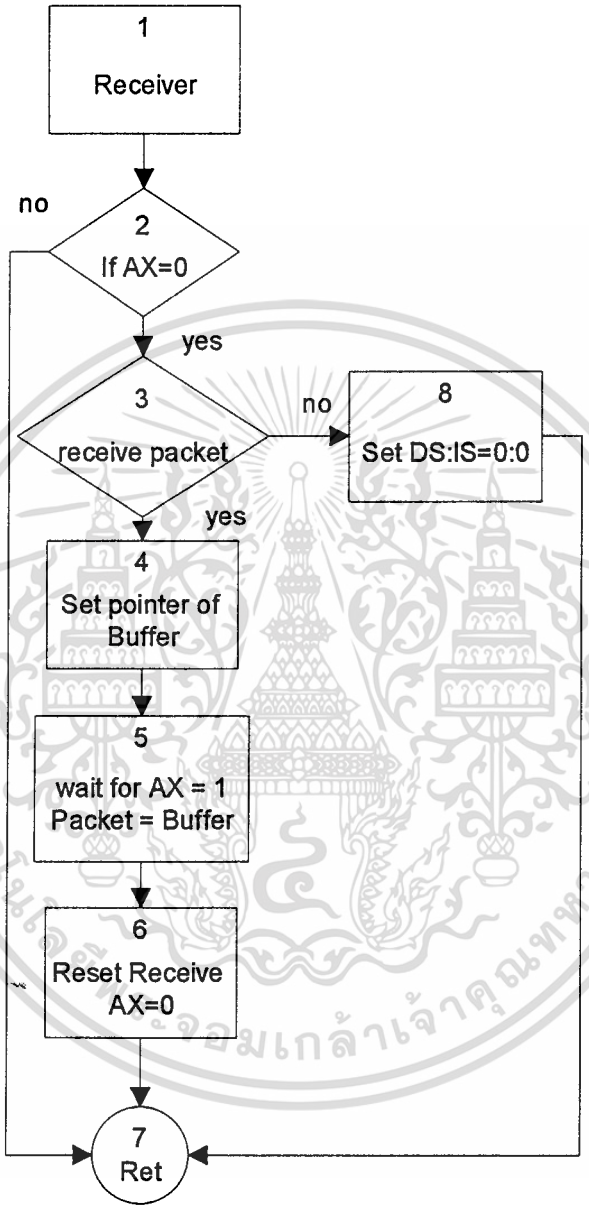

โครงสร้างของโปรแกรมหลัก

ในส่วนของโปรแกรมหลัก นั้นจะใช้ภาษาปาสคาล เป็นตัวพัฒนา โดยจะมีการเรียกฟังก์ชั่น ต่าง ๆ ที่กล่าวไว้ ตามผังงาน ที่แสดงในภาพที่ 52

ภาพที่ 52



ภาพที่ 52 (ต่อ)



แสดงผังงานส่วนของการรับข้อมูลแพกเก็ต

การใช้งานโปรแกรม

การใช้งานซอฟท์แวร์สำหรับวิเคราะห์เครือข่าย มีพารามิเตอร์ที่ต้องใช้งานร่วมดังนี้

LANANZ.EXE [/M 1:2] [/S (Source Address)] [/D (Destination Address)]
[/T (Type:Length)] [/P (Pointer:"nnnnnn")]

/M : แสดงถึงค่าโหมดของการทำงาน ค่า 1 แสดงถึงการทำงานสำหรับตรวจจับข้อมูลในเครือข่าย โดยจะแสดงรายงานผลเป็นกราฟแท่ง จะบอกถึงจำนวนข้อมูลที่เกิดขึ้นเป็นเปอร์เซ็นต์ต่อทุก ๆ วินาที และค่า 2 แสดงถึงการเลือกตรวจจับเฉพาะข้อมูลแพกเก็ตที่ต้องการ โดยขึ้นอยู่กับพารามิเตอร์ที่กำหนด กรณีที่ไม่ได้ระบุพารามิเตอร์จะแสดงถึงการเลือกทุกแพกเก็ต

/S : เป็นพารามิเตอร์ที่แสดงเงื่อนไข ให้เลือกตรวจจับแพกเก็ต ที่มี Source Address เป็นไปตามค่าที่กำหนด มีค่าจำนวน 6 ไบท์

/D : เป็น พารามิเตอร์ที่แสดงเงื่อนไข ให้เลือกตรวจจับแพกเก็ต ที่มี Destination Address เป็นไปตามค่าที่กำหนด มีค่าจำนวน 6 ไบท์

/T : เป็นพารามิเตอร์ที่แสดงเงื่อนไข ให้เลือกตรวจจับแพกเก็ต ที่มี Type หรือ Length เป็นไปตามค่าที่กำหนด มีค่าจำนวน 2 ไบท์

/P : เป็นพารามิเตอร์ที่แสดงเงื่อนไข ให้เลือกตรวจจับแพกเก็ต โดยระบุตำแหน่งของข้อมูลที่ต้องการ Pointer หมายถึงตำแหน่งมีค่าเป็น 2 ไบท์ ส่วนค่า "nnnnnn" แสดงถึงข้อมูลที่ต้องการเลือกในแพกเก็ต

บทที่ 7

การทดสอบประยุกต์ใช้งาน และบทสรุป

บทนำ

ในการทดสอบใช้งานนั้น ได้แบ่งทำการทดสอบออกเป็นหลายวิธีด้วยกัน โดยทดสอบกับเครือข่ายที่ใช้งานจริง เช่นการตรวจสอบสภาพความหนาแน่นการจราจรของข้อมูลในเครือข่าย เพื่อให้ทราบถึงจำนวนของข้อมูลที่รับส่งภายในเครือข่าย การทดสอบโดยการส่งข้อมูลแพกเก็ตข้ามไปยังเครือข่ายที่อยู่ต่างกัน โดยมีตัว Router เป็นตัวเชื่อมต่อในเครือข่าย เพื่อให้ทราบถึงเส้นทางต่าง ๆ ของข้อมูลแพกเก็ต ที่ส่งออกไป การทดสอบเลือกตรวจจับเฉพาะชนิดของโปรโตคอลที่ต้องการ เพื่อต้องการเลือกโปรโตคอลที่สนใจ การทดสอบการตรวจจับข้อมูลทั่วไปและการวิเคราะห์ข้อมูลในแพกเก็ต เพื่อให้ทราบรายละเอียดโครงสร้างภายในแพกเก็ต

การตรวจสอบสภาพความหนาแน่นการจราจรของข้อมูลในเครือข่าย

ในการทดสอบนี้ได้ใช้ ซอฟแวร์สำหรับวิเคราะห์เครือข่าย ให้ทำงานในโหมดการตรวจสอบสภาพความหนาแน่นการจราจรของข้อมูล โดยจะแสดงสภาวะความหนาแน่นของข้อมูลทุกๆ 1 วินาที และได้แบ่งระดับความหน้าแน่นของข้อมูลออกเป็น 3 ระดับด้วยกันคือ ระดับปกติ (Normal จะมีช่วงตั้งแต่ 0-45 % ถ้าสภาพการรับส่งข้อมูลอยู่ในระดับนี้ก็แสดงถึงอยู่ในสภาวะปกติ ระดับสูง (High) จะมีช่วงตั้งแต่ 46-70 % ถ้าสภาพการรับส่งข้อมูลอยู่ในระดับนี้แสดงถึงสภาพการจราจรของข้อมูลอยู่ในสภาวะหนาแน่นมีผลทำให้การรับส่งข้อมูลอาจไม่ดีเท่าที่ควร ผู้ดูแลระบบเครือข่ายควรเริ่มตรวจดูระบบเครือข่าย และระดับสุดท้ายคือ ระดับสูงมาก (Very High) จะมีช่วงตั้งแต่ 71-100 % ถ้าสภาพการรับส่งข้อมูลอยู่ในระดับนี้ แสดงถึงสภาพการจราจรของข้อมูลอยู่ในสภาวะที่หนาแน่นมาก มีผลทำให้การรับส่งข้อมูลเกิดการติดขัดมาก มีความผิดผลาดของข้อมูลสูง ผู้ดูแลระบบเครือข่ายต้องทำการตรวจสอบและวิเคราะห์ปัญหาที่

เกิด เพื่อทำการแก้ไขให้อยู่ในสภาวะการรับส่งข้อมูลให้เป็นปกติ โดยที่มีวิธีการคำนวนจากสูตรดังต่อไปนี้

กำหนดให้

SUM = ผลรวมขนาดของความยาวในแต่ละแพกเก็ตที่เกิดขึ้นภายใน 1 วินาที เป็นไบท์

MAX = ค่าจำนวน MAXIMUM ที่มาตรฐานกำหนดไว้ เป็นไบท์

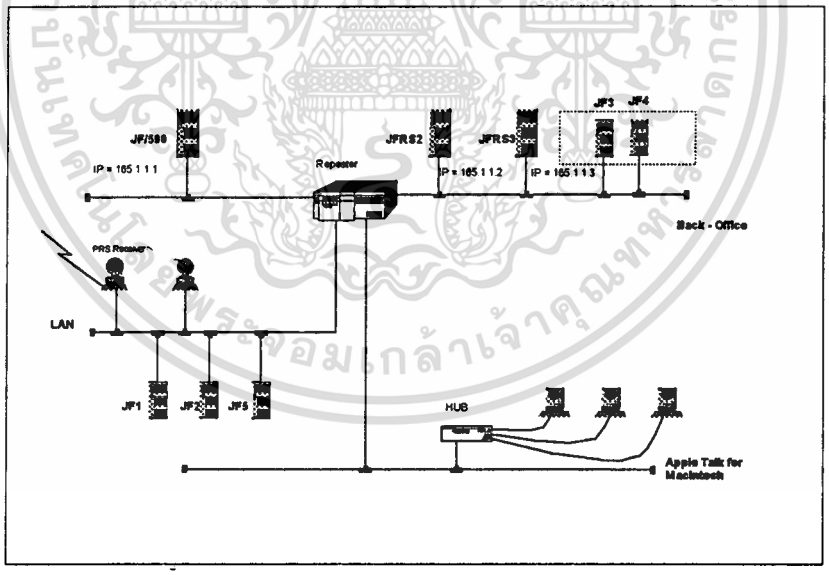
= 10 เมกกะบิทต่อวินาที

= 1,250,000 ไบท์ต่อวินาที

TRAFFIC INTENSITY = (SUM x 100) / MAX %

การทดสอบได้นำซอฟแวร์สำหรับวิเคราะห์เครือข่าย ให้ทำงานในโหมดตรวจสอบการจราจรของข้อมูลบนเครื่องคอมพิวเตอร์ส่วนบุคคล โดยเชื่อมต่อกับเครือข่ายที่ใช้งานอยู่จริง ในเครือข่ายที่ทดสอบประกอบไปด้วย ไฟล์เซิฟเวอร์จำนวน 6 เครื่อง ยูนิกส์เซิฟเวอร์จำนวน 3 เครื่อง และเครื่องคอมพิวเตอร์ในเครือข่ายอีก 210 เครื่อง ดังแสดงในภาพที่ 53

ภาพที่ 53



แสดงเครือข่ายจริงที่ใช้ทำการทดสอบความหนาแน่นการจราจรของข้อมูล

ขั้นตอนการเลือกโหมดการทำงาน ในการตรวจสอบความหนาแน่นการจราจรข้อมูล มีขั้นตอนดังต่อไปนี้

- โหลด Packet Driver ของ Ethernet Card พร้อมกับเลือก Interrup Vector ที่ต้องการ อยู่ในช่วง 0x60 - 0x80 ในการทดสอบนี้ใช้ Ethernet Card ของบริษัท 3Com

C:\ 3C503.COM 0x60

- โหลดซอฟแวร์ Lan Analyzer เพื่อทำการทดสอบ

C:\LANANZ.EXE

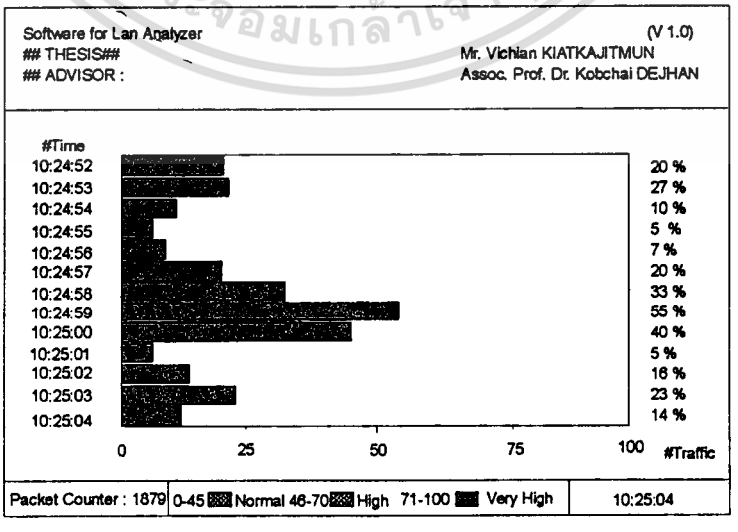
ที่หน้าจอคอมพิวเตอร์แสดงผลดังต่อไปนี้

```
********     Lan Analyzer     ********
          1. Traffic Monitor
          2. Filter Packet
          3. Display Packet
      Select Mode Monitor [ 1 - 3 ]  1
***********************************
```

เลือกข้อ 1. เพื่อตรวจสอบสภาพความหนาแน่นของข้อมูลที่เกิดขึ้นภายในเครือข่าย ผลที่ได้สามารถแสดงได้ใน ภาพที่ 54

จากผลการทดสอบสามารถเห็นความเปลี่ยนแปลง ของความหนาแน่นการจราจรของข้อมูลบนเครือข่าย ตามช่วงเวลาต่าง ๆ ได้ โดยบางช่วงเวลาการรับส่งข้อมูลมีขนาดสูงถึง 50 % แต่จากโดยรวมแล้วการรับส่งข้อมูลในเครือข่ายนี้ถืออยู่ในระดับปกติ ถ้าในกรณีที่ขนาดการรับส่งข้อมูลมีค่า สูงถึง 50 % เป็นเวลาติดต่อกัน ก็สามารถบอกได้ว่าเครือข่ายนี้มีการจราจรของข้อมูลหนาแน่น ผู้ดูแลระบบควรเข้ามาตรวจสอบหรือ ดูสาเหตุที่เกิดขึ้น

ภาพที่ 54



แสดงหน้าจอแสดงผลของ ซอฟแวร์สำหรับวิเคราะห์เครือข่าย

การทดสอบการตรวจจับข้อมูลโดยการส่งข้อมูลแพกเก็ตข้ามไปยังเครือข่ายที่อยู่ต่างกัน

ขั้นตอนของการทดสอบนั้นจะใช้ซอฟแวร์สำหรับวิเคราะห์เครือข่าย ที่ได้พัฒนาขึ้นมา ให้ทำงานในโหมดเลือกตรวจจับชนิดของโปรโตคอล ที่เป็น IP เท่านั้นคือ 0800h โดย Run บนเครื่องคอมพิวเตอร์ที่เชื่อมต่ออยู่กับเครือข่ายต่าง ๆ และทำการทดสอบโดยการส่งผ่านข้อมูลจาก Host หนึ่งไปยังอีก Host หนึ่ง โดยผ่านไปยังเครือข่ายต่างๆ แสดงได้ในภาพ 56 ซึ่งจะเป็นตัวอย่างของการทดลอง จะประกอบด้วย Host ต้นทาง คือ Host A และ Host ปลายทางคือ Host B, ตัวกลางที่เป็น Router 3 เครื่อง และ เครือข่ายอีกจำนวน 4 เครือข่าย โดยได้มีการแสดงถึง ค่าของ Internet และ Ethernet address ของแต่ละ Host รวมถึง Port แต่ละ Port ของ Router และสามารถสรุปเป็นวิธีการ และขั้นตอนของการส่งผ่านข้อมูลได้ดังนี้

- โหลด Packet Driver ของ Ethernet Card พร้อมกับเลือก Interrup Vector ที่ต้องการอยู่ในช่วง 0x60 - 0x80 ในการทดสอบนี้ใช้ Ethernet Card ของบริษัท 3Com

```
C:\ 3C503.COM  0x60
```

- โหลดซอฟแวร์ Lan Analyzer เพื่อทำการทดสอบ

```
C:\LANANZ.EXE
```

ที่หน้าจอคอมพิวเตอร์แสดงผลดังต่อไปนี้

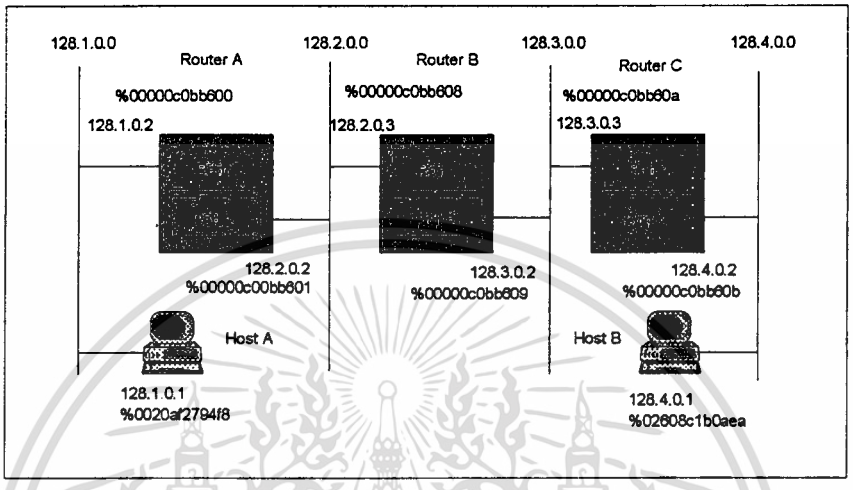
```
********    Lan Analyzer     ********
        1. Traffic Monitor
        2. Filter Packet
        3. Display Packet
    Select Mode Monitor [ 1 - 3 ]  2
**********************************
```

เลือกข้อ 2. เพื่อตรวจสอบเฉพาะตำแหน่งข้อมูล Packet ที่ต้องการ

- Select the Destination Address [Y/N] : ตอบ N
- Select the Source Address [Y/N] : ตอบ N
- Select the Type of Packet [Y/N] : ตอบ Y
- Enter the Type of Packet 2 Byte [xxxx] : 0800 (เป็นค่าที่เลือกเฉพาะ IP)
- Select the Pointer of Packet [Y/N] : ตอบ N
- Do you want to capture file [Y/N] : ตอบ N

ทำเช่นเดียวกันกับเครื่องที่ใช้สำหรับ Analyzer บนเครือข่าย 128.1.0.0 128.2.0.0 128.3.0.0 และ 128.4.0.0

ภาพที่ 55



แสดงการทดสอบการส่งข้อมูลข้ามไปยังเครือข่ายต่าง ๆ

Host A

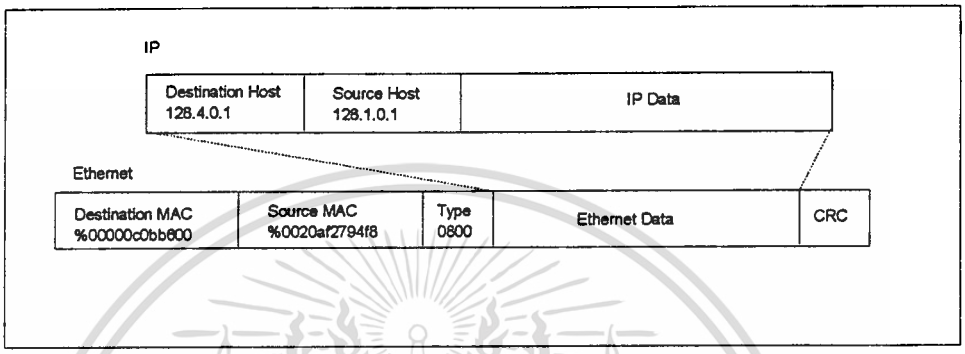
Host A ซึ่งอยู่บนเครือข่าย 128.1.0.0 ต้องการที่จะติดต่อกับ Host B ซึ่งอยู่บนเครือข่าย 128.4.0.0 โดยใช้โปรโตคอล Telnet ดังนั้นแพกเก็ตจาก Host A ไปยัง Host B นั้นจะต้องผ่าน Router หนึ่งไปยังอีก Router หนึ่ง จะเห็นได้ว่าค่าของ IP Header ที่กำหนดไว้จาก Host จะไม่เปลี่ยนแปลง แต่ส่วนที่จะเปลี่ยนแปลงก็เฉพาะ source และ destination ของ Ethernet address เท่านั้น ซึ่งเป็นขั้นตอนดังนี้

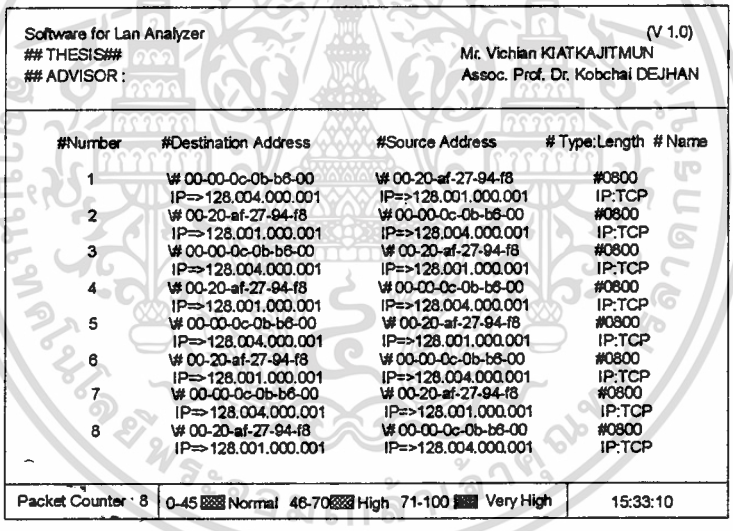
แพกเก็ตบนเครือข่าย 128.1.0.0

เมื่อ Host A และ Host B อยู่ต่างเครือข่ายกันนั้น Host A จะต้องอาศัย IP Router โดยที่การกำหนดเริ่มแรกนั้น Host A จะทำการเรียนรู้ว่า IP address ของ เส้นทางหลัก (Default Gatway) คือ 128.1.0.2 และที่ Router นี้แต่จะเป็นตัวที่ Host A จะทำการส่งแพกเก็ต ออกไปเมื่อแพกเก็ต นั้นเป็นแพกเก็ต ของ Internet address ที่ต่างเครือข่ายกัน ถ้าไป Host A ยังไม่มี ARP Chache ของ (Router) 128.1.0.2 มันก็จะทำการส่ง ARP Request และรอให้ Router A ตอบกลับมา และเมื่อได้ address mapping ที่สมบูรณ์แล้ว Host A ก็จะส่ง Ethernet Frame พร้อมด้วย Destination MAC address ของ %00000

c0bb600 (Router A), และ Source MAC address ของ %0020af2794f8 (Host A), และ type field ของ 0800h (IP) ซึ่งโครงสร้างของแพกเก็ตบนเครือข่าย 128.1.0.0 สามารถที่จะแสดงได้ภาพที่ 56

ภาพที่ 56





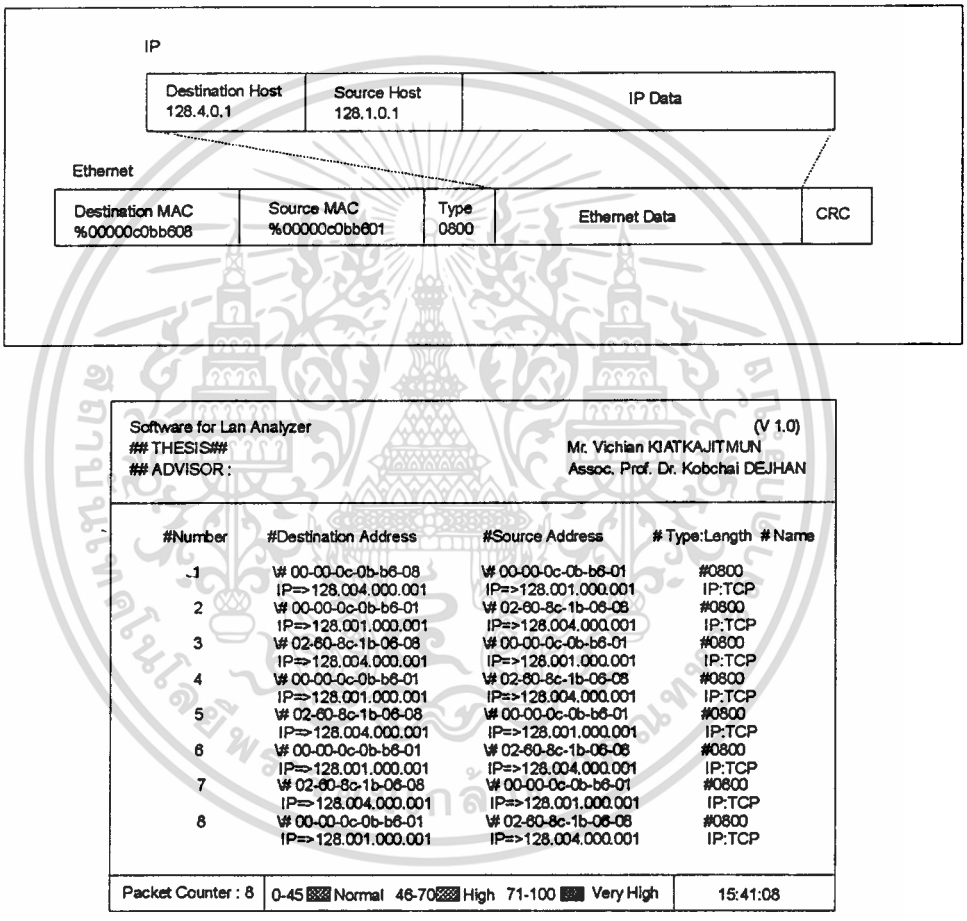
แสดงแพกเก็ตบนเครือข่าย 128.1.0.0

แพกเก็ตบนเครือข่าย 128.2.0.0

เมื่อ Router A ได้รับแพกเก็ต มาเรียบร้อยแล้ว ก็จะนำเอา Ethernet header ออก จะเหมือนเฉพาะส่วนของข้อมูลนำไปยัง IP process ซึ่ง IP process นั้นจะรู้ถึงหมายเลขเครือข่าย ซึ่งจะอยู่ใน IP header และเรียนรู้ตำแหน่งของ เครือข่าย 128.4.0.0 ในตาราง Routing โดยที่ Router A จะรู้ว่าเครือข่ายปลายทาง นั้นจะอยู่ถัดไป 2 hops และจะต้องส่งส่วนของข้อมูลไปยัง Router B ที่ IP address 128.2.0.3 โดยที่ Router A จะส่ง ARP

Request และรอให้ Router B ตอบกลับ (ในกรณีที่ไม่มี address mapping ใน ARP cache ของ Router A) จากนั้น Router A ก็สามารถส่ง Ethernet Frame บน Port2 ซึ่งจะประกอบด้วย Destination MAC address ของ %00000c0bb608 (Router B), source MAC address ของ %00000c0bb601 (Port 2 ของ Router A) และ type field ของ 0800h (IP) ซึ่งโครงสร้างของแพกเก็ต บนเครือข่าย 128.2.0.0 แสดงได้ในภาพที่ 57

ภาพที่ 57



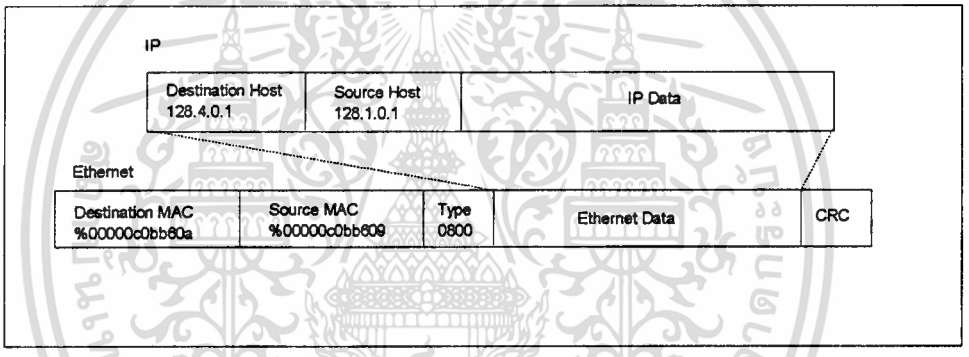
แสดงแพกเก็ตบนเครือข่าย 128.2.0.0

แพกเก็ตบนเครือข่าย 128.3.0.0

เมื่อ Router B ได้รับแพกเก็ตจาก Router A แล้วก็จะนำเอา Ethernet herder ออก จะเหลือเพียงส่วนที่เป็นข้อมูล ส่งต่อไปยัง IP Process ซึ่ง IP Process นั้นจะรู้ว่าหมายเลขเครือข่าย อยู่ใน IP herder ทำการเรียนรู้ตำแหน่งของ เครือข่าย 128.4.0.0 ที่อยู่ในตาราง

Routing โดยที่ Router B จะรู้ว่าเครือข่ายปลายทาง จะอยู่ถัดไปอีก 1 Hop และจะต้องส่งข้อมูล ไปยัง Router C ที่ IP address 128.3.0.3 ในกรณีที่ Router B ยังไม่มี Address mapping นั้น Router B ก็จะทำการส่ง ARP Request ออกไปและจะรอ Router C ตอบกลับมา เมื่อได้ address mapping แล้ว Router B ก็สามารถที่จะส่ง Ethernet frame ไปบน Port2 ซึ่งประกอบไปด้วย Destination MAC address ของ %00000c0bb60a (Router C), source MAC address ของ %00000c0bb609 (Port2 ของ Router B) และ type field ของ 0800h (P) ซึ่งโครงสร้างของแพกเก็ตบนเครือข่าย 128.3.0.0 แสดงได้ในภาพที่ 59

ภาพที่ 58



Software for Lan Analyzer (V 1.0)
THESIS##
ADVISOR :
Mr. Vichian KIATKAJITMUN
Assoc. Prof. Dr. Kobchai DEJHAN

#Number	#Destination Address	#Source Address	# Type:Length # Name
1	\# 00-00-0c-0b-b6-0a IP=>128.004.000.001	\# 00-00-0c-0b-b6-09 IP=>128.001.000.001	#0800 IP:TCP
2	\# 00-00-0c-0b-b6-09 IP=>128.001.000.001	\# 02-60-8c-1b-06-0a IP=>128.004.000.001	#0800 IP:TCP
3	\# 02-60-8c-1b-06-0a IP=>128.004.000.001	\# 00-00-0c-0b-b6-09 IP=>128.001.000.001	#0800 IP:TCP
4	\# 00-00-0c-0b-b6-09 IP=>128.001.000.001	\# 02-60-8c-1b-06-0a IP=>128.004.000.001	#0800 IP:TCP
5	\# 02-60-8c-1b-06-0a IP=>128.004.000.001	\# 00-00-0c-0b-b6-09 IP=>128.001.000.001	#0800 IP:TCP
6	\# 00-00-0c-0b-b6-09 IP=>128.001.000.001	\# 02-60-8c-1b-06-0a IP=>128.004.000.001	#0800 IP:TCP
7	\# 02-60-8c-1b-06-0a IP=>128.004.000.001	\# 00-00-0c-0b-b6-09 IP=>128.001.000.001	#0800 IP:TCP
8	\# 00-00-0c-0b-b6-09 IP=>128.001.000.001	\# 02-60-8c-1b-06-0a IP=>128.004.000.001	#0800 IP:TCP

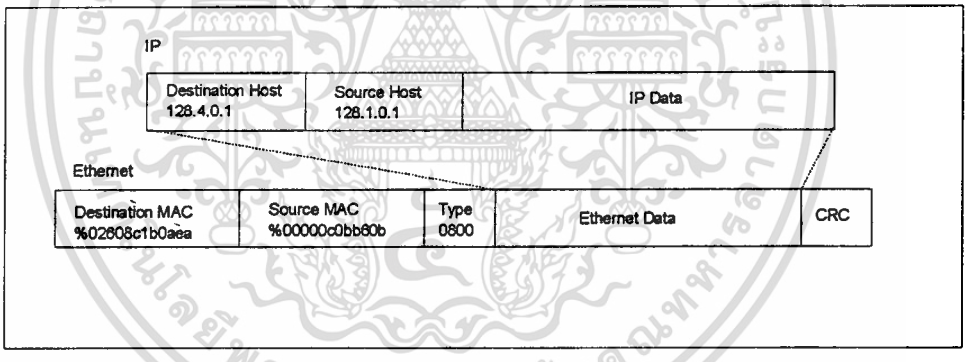
Packet Counter : 8 | 0-45 Normal 46-70 High 71-100 Very High | 15:32:23

แสดงแพกเก็ตบนเครือข่าย 128.3.0.0

แพกเก็ตบนเครือข่าย 128.4.0.0

เมื่อ Router C ได้รับแพกเก็ตจาก Router B แล้ว ก็จะนำเอา Ethernet headre ออก และเหลือเพียงส่วนของขอมูล ส่งต่อไปยัง IP Process ซึ่ง IP Process จะรู้ว่าหมายเลขของ เครือข่าย อยู่ใน IP herder ทำการเรียนรู้ตำแหน่งของเครือข่าย 128.4.0.0 ที่อยู่ในตาราง Routing โดยที่ Router C จะรู้ว่าเครือข่ายปลายทางนั้นต่ออยู่ในครงข่าย เดียวกันที่ Post2 ดังนั้นจึงไม่จำเป็นที่ต้องส่งส่วนของข้อมูลไปยัง Router ตัวอื่น Router C นั้นสามารถส่งส่วนข้อมูลตรงไปยังปลายทางได้โดยตรง และเช่นเดียวกันในกรณีที่ Router C ยังไม่มี address mapping นั้น Router C ก็จะส่ง ARP Request ออกไปและรอ Host B ตอบกลับเมื่อได้ adress mapping แล้ว Router C ก็สามารถที่จะส่ง Ethernet Frame ไปบน Port2 ซึ่งประกอบไปด้วย Destination MAC address ของ %02608c1b0aea (Host B), Source MAC address ของ %00000c0bb60b (Port2 ของ Router ( ), และ Type field ของ 800h (IP) ซึ่งโครงสร้างของแพกเก็ตบนเครือข่าย 128.4.0.0 แสดงได้ในภาพที่ 60

ภาพที่ 59



Software for Lan Analyzer (V 1.0)
THESIS## Mr. Vichian KIATKAJITMUN
ADVISOR : Assoc. Prof. Dr. Kobchai DEJHAN

#Number	#Destination Address	#Source Address	# Type:Length # Name
1	\# 02-60-8c-1b-06-ea	\# 00-00-0c-0b-b6-0b	#0800
	IP=>128.004.000.001	IP=>128.001.000.001	IP:TCP
2	\# 00-00-0c-0b-b6-0b	\# 02-60-8c-1b-06-ea	#0800
	IP=>128.001.000.001	IP=>128.004.000.001	IP:TCP
3	\# 02-60-8c-1b-06-ea	\# 00-00-0c-0b-b6-0b	#0800
	IP=>128.004.000.001	IP=>128.001.000.001	IP:TCP
4	\# 00-00-0c-0b-b6-0b	\# 02-60-8c-1b-06-ea	#0800
	IP=>128.001.000.001	IP=>128.004.000.001	IP:TCP
5	\# 02-60-8c-1b-06-ea	\# 00-00-0c-0b-b6-0b	#0800
	IP=>128.004.000.001	IP=>128.001.000.001	IP:TCP
6	\# 00-00-0c-0b-b6-0b	\# 02-60-8c-1b-06-ea	#0800
	IP=>128.001.000.001	IP=>128.004.000.001	IP:TCP
7	\# 02-60-8c-1b-06-ea	\# 00-00-0c-0b-b6-0b	#0800
	IP=>128.004.000.001	IP=>128.001.000.001	IP:TCP
8	\# 00-00-0c-0b-b6-0b	\# 02-60-8c-1b-06-ea	#0800
	IP=>128.001.000.001	IP=>128.004.000.001	IP:TCP

Packet Counter : 8 | 0-45 Normal 46-70 High 71-100 Very High | 15:31:04

แสดงแพกเก็ตบนเครือข่าย 128.4.0.0

Host B

เมื่อ Host B รับแพกเก็ตมาได้ ก็จะนำเอา Ethernet header ออก แล้วผ่านไปให้ IP module ซึ่งจะมี IP process ทำการตรวจสอบข้อมูลซึ่งเป็น address ของ Local host จากนั้น นำเอา IP header ออกไปแล้วส่งผ่านไปให้ TCP เพื่อจัดการต่อไป TCP จะทำการตรวจสอบหมายเลขของ Port และส่งผ่านข้อมูลไปยังส่วน Input queue สำหรับ Telnet process

การทดสอบการตรวจับข้อมูลทั่วไป และการวิเคราะห์ข้อมูลในแพกเก็ต

ในการทดสอบนี้จะเลือกโหมดการทำงานเป็นข้อ 2. Filter Packet โดยที่จะสามารถตรวจรับข้อมูลที่เกิดขึ้นในเครือข่าย และสามารถนำข้อมูลที่รับมาไปเก็บใน Hard Disk ได้ สำหรับเครือข่ายที่ใช้ทดสอบนั้นเป็นเครือข่ายเดิมแสดงได้ในภาพที่ 53มีขั้นตอนดังต่อไปนี้

- โหลด Packet Driver ของ Ethernet Card พร้อมกับเลือก Interrup Vector ที่ต้องการอยู่ในช่วง 0x60 - 0x80 ในการทดสอบนี้ใช้ Ethernet Card ของบริษัท 3Com

```
C:\-3C503.COM  0x60
```

- โหลดซอฟแวร์ Lan Analyzer เพื่อทำการทดสอบ

```
C:\LANANZ.EXE
```

ที่หน้าจอคอมพิวเตอร์แสดงผลดังต่อไปนี้

```
********      Lan Analyzer      ********
        1. Traffic Monitor
        2. Filter Packet
        3. Display Packet
    Select Mode Monitor [ 1 - 3 ]  2
***********************************
```

เลือกข้อ 2. เพื่อตรวจสอบเฉพาะตำแหน่งข้อมูล Packet ที่ต้องการ

- Select the Destination Address [Y/N] : ตอบ N
- Select the Source Address [Y/N] : ตอบ N
- Select the Type of Packet [Y/N] : ตอบ N
- Select the Pointer of Packet [Y/N] : ตอบ N

- Do you want to capture file [Y/N] : Y (เพื่อนำข้อมูลแพกเก็ตที่รับเข้ามาได้ไปเก็บไว้ที่ Hard Disk)

- Enter file name to capture : CAPTURE.DAT (ชื่อไฟล์ที่ต้องการเก็บข้อมูล๗

ตัวอย่างของข้อมูลแพกเก็ตที่เกิดขึ้นในเครือข่ายแสดงได้ในภาพที่ 60

ภาพที่ 60

Software for Lan Analyzer (V 1.0)
THESIS## Mr. Vichian KIATKAJITMUN
ADVISOR : Assoc. Prof. Dr. Kobchai DEJHAN

#Number	#Destination Address	#Source Address	# Type:Length # Name
45	00-40-8c-10-6f-54	00-00-0c-0b-b6-08	0026 60
46	00-00-0c-0b-b6-08	00-40-8c-10-6f-54	002e 60
47	00-40-8c-10-6f-54	00-00-0c-0b-b6-08	0026 60
48	02-60-8c-a6-e1-bd	00-0a-24-2f-4f-0a	0800 60
49	00-00-0c-0b-b6-08	00-40-8c-10-cd-1b	002e 60
50	00-40-8c-10-cd-1b	00-00-0c-0b-b6-08	0026 60
51	00-00-0c-0b-b6-08	00-40-8c-10-cd-1b	002e 60
52	00-40-8c-10-cd-1b	00-00-0c-0b-b6-08	0026 60
53	00-00-0c-0b-b6-08	00-40-8c-16-21-b6	002e 60
54	00-0a-24-2f-4f-0a	02-60-8c-a6-e1-bd	0800 108
55	00-20-af-d8-df-7d	00-00-0c-0b-b6-08	0800 227
56	02-60-8c-a6-e1-bd	00-a0-24-2f-4f-0a	0800 72
57	00-00-0c-0b-b6-08	00-40-8c-10-6f-54	002e 60

Packet Counter : 57 | 0-45 Normal 46-70 High 71-100 Very High | 18:01:08

แสดงการตรวจจับข้อมูลทั่วไป

เมื่อได้รับข้อมูลเรียบร้อยแล้วก็ออกจากโปรแกรม โดยการกดคีย์ Esc จากนั้นโหลดซอฟแวร์ Lan Analyzer อีกครั้ง ที่จอคอมพิวเตอร์แสดงผลดังต่อไปนี้

******** Lan Analyzer ********

1. Traffic Monitor
2. Filter Packet
3. Display Packet

Select Mode Monitor [1 - 3] 3

****

เลือกข้อ 3. เพื่อนำข้อมูลที่เก็บไว้มาแสดงผล และวิเคราะห์

- Enter your file name :CAPTURE.DAT (เป็นการใส่ชื่อแฟ้มข้อมูลที่ได้เก็บข้อมูลไว้)

เลือกตัวอย่างข้อมูลแพกเก็ตในรายการที่ 55 เพื่อนำมาวิเคราะห์แสดงได้ภาพที่ 61

ภาพที่ 61

```
Select The Packet Number [ 0 = Exit ] : 55
Packet Number          : 55
Time Received          : 21:43:04
Destination Address    : 00-20-AF-D8-DF-7D
Source Address         : 00-00-0C-B0-B6-08
Type:Length            : 0800
Length of Packet       : 227

      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0000: 00 20 AF D8 DF 7D 00 00 0C 0B B6 08 08 00 45 00
0010: 00 D5 1E F0 00 00 3B 06 13 43 A5 02 01 04 A5 01
0020: 01 E1 00 17 3F 03 0A AA B5 CC 07 F0 20 8A 50 18
0030: 40 00 01 BA 00 00 20 20 20 66 69 73 76 77 20 32
0040: 31 37 36 39 20 32 38 34 32 34 20 20 20 37 20 31
0050: 36 3A 33 38 3A 30 35 20 20 70 74 73 2F 33 20 20
0060: 30 3A 30 30 20 2D 6B 73 68 20 0D 0A 20 20 20 66
0070: 69 73 76 77 20 32 33 38 30 33 20 20 20 20 20 31
0080: 20 20 20 30 20 20 20 41 70 72 20 30 32 20 20 20
0090: 20 20 20 2D 20 20 31 3A 31 32 20 6C 6F 61 64 61
00A0: 76 67 64 20 32 31 31 32 20 0D 0A 20 20 20 66 69
00B0: 73 76 77 20 33 39 34 36 39 20 32 31 37 36 39 20
00C0: 20 20 34 20 31 36 3A 34 35 3A 35 39 20 20 70 74
00D0: 73 2F 33 20 20 30 3A 30 30 20 70 73 20 2D 65 66
00E0: 20 0D 0A
```

แสดงตัวอย่างรายละเอียดของข้อมูลแพกเก็ตที่ตรวจรับเข้ามาได้ในลำดับที่ 55

สามารถวิเคราะห์รายละเอียดได้ดังนี้ ข้อมูลลำดับตั้งแต่ 0000-0005 จะแสดงถึง Destination Address คือ 00-20-AF-D8-DF-7D ลำดับตั้งแต่ 0006-000B จะแสดงค่าถึง

Source Address คือ 00-00-0C-0B-B6-08 ลำดับตั้งแต่ 000C-000D จะแสดงค่าถึงชนิดของแพกเก็ตซึ่งมีค่าเท่ากับ 0800h หมายถึง IP โปรโตคอล เมื่อทราบว่าเป็น IP โปรโตคอลแล้ว สามารถทำการวิเคราะห์ถึงค่า IP Header ได้ดังต่อไปนี้ ข้อมูล 4 บิทสูง ของลำดับที่ 000E จะแสดงค่าถึง Version เป็นรูปแบบของ Internet Header และที่ใช้อยู่ในปัจจุบันนี้มีค่าเป็น 4 ต่อไปข้อมูล 4 บิทต่ำ ของลำดับที่ 000E จะแสดงค่าถึง IHL (Internet Header Lengh) เป็นขนาดความยาวของ Internet Header และค่าที่ได้มีค่าเท่ากับ 5 แสดงถึงขนาดของ Internet Header มีค่าเท่ากับ 20 ไบท์และจะไม่มีค่าของ Option กับ Padding ลำดับที่ 000F จะแสดงค่าถึง TOS (Type of Service) มีค่าเท่ากับ 00 ลำดับที่ 0010-0011 จะแสดงค่าถึง Total Length เป็นขนาดของจำนวนข้อมูล โดยจะรวมส่วนของ IP Header และ Data มีค่าเป็นไบท์ จะแสดงอยู่ในรูปอ็อกเต็ท มีค่าเท่ากับ 00D5 เมื่อแปลงเป็นค่าอ็อกเต็ทจะได้เท่ากับ 213 ไบท์ ลำดับตั้งแต่ 0012-0013 จะแสดงค่าถึง Identification มีค่าเท่ากับ 1EF0h ตำแหน่งข้อมูล 4 บิทสูง ของลำดับที่ 0014 แสดงค่าถึง Flage มีค่าเท่ากับ 0 ตำแหน่งข้อมูล 4 บิทต่ำของลำดับที่ 0014 รวมกับข้อมูลลำดับที่ 0015 แสดงถึงค่า Fragment Offset มีค่าเท่ากับ 0 ตำแหน่งข้อมูลลำดับที่ 0016 แสดงค่าถึง Time of Live มีค่าเท่ากับ 3Bh เมื่อแปลงอยู่ในรูปของอ็อกเต็ทมีค่าเท่ากับ 59 ตำแหน่งข้อมูลลำดับที่ 0017 จะแสดงค่าถึงโปรโตคอล มีค่าเท่ากับ 6 หมายถึงโปรโตคอล TCP ตำแหน่งข้อมูลลำดับที่ 0018-0019 จะแสดงค่าถึง IP Header Checksum มีค่าเท่ากับ 1343h ตำแหน่งข้อมูลลำดับที่ 001A-001D จะแสดงค่าถึง Source IP Address มีค่าเท่ากับ A5-02-01-04 เมื่อแปลงค่าให้ IP Address อยู่ในรูปอ็อกเต็ท จะได้เท่ากับ 165.2.1.4 ตำแหน่งข้อมูลลำดับที่ 001E-0021 จะแสดงค่าถึง Destination IP Address มีค่าเท่ากับ A5-01-01-E9 เมื่อแปลงค่าให้ IP Address อยู่ในรูปอ็อกเต็ท จะได้เท่ากับ 165.1.1.233 ตำแหน่งข้อมูลลำดับที่ 0022-0023 จะแสดงค่าถึง Source Port ของ TCP มีค่าเท่ากับ 0017h แปลงให้อยู่ในรูปอ็อกเต็ทจะได้เท่ากับ 23 หมายถึง Telnet ตำแหน่งข้อมูลลำดับที่ 0024-0025 จะแสดงค่าถึง Destination Port ของ TCP มีค่าเท่ากับ 3F03h แปลงให้อยู่ในรูปอ็อกเต็ทจะได้เท่ากับ 16131 ตำแหน่งข้อมูลลำดับที่ 0026-0029 แสดงถึงค่า Sequence Number มีค่าเท่ากับ 0AAAB5CCh ตำแหน่งข้อมูลลำดับที่ 002A-002D แสดงถึงค่า Ackowledgement Number มีค่าเท่ากับ 07F0208Ah ตำแหน่งข้อมูล 4 บิทสูงของลำดับที่ 002E แสดงค่าถึง Data Offset มีค่าเท่ากับ 5 แสดงว่าข้อมูล TCP Header จะไม่มี Option และ Padding ตำแหน่งข้อมูลลำดับที่ 0030-0031 แสดงถึงค่า Window มีค่าเท่ากับ 4000h แปลงเป็นค่าอ็อกเต็ทได้เท่ากับ 16384 ตำแหน่งข้อมูลลำดับที่ 0032-0033 แสดงถึงค่า Checksum มีค่าเท่ากับ 01BAh ตำแหน่งข้อมูลลำดับที่ 0034-0035 แสดงถึงค่า Urgent Pointer มีค่าเท่ากับ 0000h และข้อมูลชุดสุดท้ายลำดับที่ 0036-00E2 แสดงถึงส่วนของข้อมูล

บทที่ 8

บทสรุปและข้อเสนอแนะ

บทสรุป

จากการทำวิทยานิพนธ์นี้ ทำให้รับทราบถึงรูปแบบต่าง ๆ ของแพกเก็ตข้อมูลบน IEEE 802.3 และมีโปรโตคอลเป็นที่นิยมใช้งานอยู่หลายชนิด แต่ละชนิดของโปรโตคอลก็มีความสามารถแตกต่างกันออกไป ได้ทำการศึกษาและเรียนรู้เกี่ยวกับมาตรฐานของแพกเก็ตไดรว์เวอร์ เพื่อที่จะสามารถเขียนโปรแกรม ติดต่อขอข้อมูลที่ต้องการได้ โดยที่แต่ละฟังก์ชันการทำงานของแพกเก็ตไดรว์เวอร์ ก็มีความสามารถแตกต่างกันออกไป และนำข้อมูลที่ได้ไปประยุกต์ใช้งานได้

ซอฟท์แวร์สำหรับวิเคราะห์เครือข่ายท้องถิ่นนี้ การใช้งานนั้นต้องทำงานบนเครื่องคอมพิวเตอร์ ที่เชื่อมต่ออยู่ในเครือข่าย ซึ่งตัวซอฟท์แวร์นี้มีความสามารถในการตรวจจับแพกเก็ตข้อมูลต่าง ๆ ที่ทำการรับส่งอยู่บนเครือข่ายได้ โดยจะทำงานอยู่ในระดับของชั้น Data Link Layer ของ OSI Layer และสามารถแสดงผลรายละเอียดของข้อมูลได้ ในกรณีที่มีจำนวนของข้อมูล หรือจำนวนของโปรโตคอลที่ใช้งานอยู่ในเครือข่ายมาก ซอฟท์แวร์นี้สามารถที่จะเลือก หรือทำการกรองเฉพาะข้อมูลโปรโตคอลที่ต้องการได้ โดยการกำหนดค่าพารามิเตอร์ตามที่ต้องการ เช่น แอดเดรสทางต้นทาง (Source Address) แอดเดรสทางปลายทาง (Destination Address) ชนิดของโปรโตคอล (Type or Length) และรูปแบบต่าง ๆ ของข้อมูลได้

ข้อเสนอแนะ

จากการพัฒนาซอฟท์แวร์สำหรับวิเคราะห์เครือข่ายนี้ ได้รับทราบถึงวิธีการตรวจจับข้อมูลแพกเก็ตของโปรโตคอลต่าง ๆ ได้จากการติดต่อผ่านแพกเก็ตไดรว์เวอร์ หรืออาจทำการศึกษาการติดต่อผ่านไดรว์เวอร์ชนิดอื่น เช่น ODI, NDIS, และ NETBIOS และสามารถนำหลักการและวิธีการนี้ นำไปพัฒนาเพื่อสร้างเครื่องมือ หรืออุปกรณ์สำหรับใช้ในเครือข่ายได้ เช่น Bridge จะเป็นอุปกรณ์ที่ใช้สำหรับการเชื่อมต่อระหว่าง ระบบโครงข่างสองระบบเข้าด้วยกัน โดยที่จะทำงานที่ระดับของชั้น Data Link Layer ของ OSI Layer และ Router จะเป็นอุปกรณ์ที่ใช้สำหรับการเชื่อมต่อระหว่าง ระบบโครงข่ายสองระบบเข้าด้วยกัน โดยที่จะทำงานที่ระดับของ

ชั้น Network Layer ของ OSI Layer จะมีความสามารถที่จะสร้างหรือแยกแพกเก็ตข้อมูล สามารถจัดระดับความสำคัญของข้อมูล และที่เป็นความสามารถที่สำคัญของ Router ก็คือ การควบคุมเส้นทางการเดินทางของข้อมูลไปในเครือข่ายต่าง ๆ

หนังสืออ้างอิง

[1] FTP Software, Inc. "PC/TCP Version 1.09 Packet Driver Specification" pp.5-14, September 14,1989

[2] D. E. Comer, "Internetworking with TCP/IP Volume I," pp. 51-200, 1991

[3] D.E. Comer, "Internetworking with TCP/IP Volume II," pp. 27-192, 1991

[4] Cisco Systems, "Internetworking Technology Overview," pp. 2-1 to 5-20, September 1992

[5] W. Stallings,"Handbook of Computer Communications Standard Volume 2 (Local Area Network)," 1987

[6] P.H Jesty, "Networking with Microcomputers," 1992

[7] T. McGoven, "Data Communications Concepts and Applications," 1988

ภาคผนวก

1. ข้อมูล Ethernet Type Fields

Hex	Information
0000-05DC	IEEE802.3 Length Field (0.:5000.)
0200	Xerox PUP (Conflicts with IEEE802.3 Length Field range)
0201	Xerox PUP Address Translation (conflicts ...) (see 0A01)
0600	Xerox NS IDP *
0800	DOD Internet Protocol (IP) *#
0801	X.75 Internet
0802	NBS Internet
0803	ECMA Internet
0804	CHAOSnet
0805	X.25 Level 3
0806	Address Resolution Protocol (ARP) * (For IP and for CHAOS)
0807	XNS Compatibility
081C	Symbolics Private
0888-088A	Xyplex
0900	Ungermann-Bass network debugger
0A00	Xerox IEEE 802.3 PUP
0A01	Xerox IEEE 802.3 PUP Address Translation
0BAD	Banyan Systems
1000	Berkeley Trailer negotiation
1001-100F	Berkeley Trailer encapsulation
1600	VALID-machine protocol?*
5208	BBN Simnet Private %
6000	DEC unassigned
6001	DEC Maintenance Operation Protocol (MOP) Dump/Load Assistace
6002	DEC Maintenance Operation Protocol (MOP) Remote Console

Hex	Information
6003	DecNET Phase IV
6004	DEC Local Area Transport (LAT)
6005	DEC diagnostic protocol (at Interface initialization?)
6006	DEC Customer protocol
6007	DEC Local Area VAX Cluster (LAVC)
6008	DEC unasstgned
6009	DEC unasstgned
6010-6014	3Com
7000	Ungermann-Bass download
7002	Ungermann-Bass diagnostic/loopback
7020-7029	LRT
7030	Proteon
8003	Cronus VLN
8004	Cronus Direct
8005	HP Probe Protocol
8006	Nestar
8008	AT&T
8010	Excelan
8013	Silicon Graphics diagnostic
8014	Silicon Graphics network garnes
8015	Silicon Graphics reserved
8016	Silicon Graphics XNS Nameserver,bounce server
8019	Apollo DOMAIN
802E	Tymshare
802F	Tigan
8035	Reversse Address Resolution Protocol (RARP)
8036	Aeonic Systems
8038	DEC Lan Bridge Management
8039	DEC unassigned
803A	DEC unassigned
803B	DEC unassigned

Hex	Information
803C	DEC unassigned
803D	DEC Ethernet Encryption Protocol
803E	DEC unassigned
803F	DEC LAN Traffic Monitor Protocol
8040	DEC unassigned
8041	DEC unassigned
8042	DEC unassigned
8044	Planning research Co.
8046	AT&T
8047	AT&T
8049	ExperData
805B	Stanford V kernel, experimental
805C	Stanford V kernal, production
805D	Evans and Sutherland
8060	Little Machines
8062	Counterpoint Computers
8065	University of Massachusetts, Amherst
8066	University of Massachusetts, Amherst
8067	Veeco Integrated Automation
8068	General Dynamics
8069	AT&T
806A	Autophon
806C	ComDesign
806D	Compugraphic
806E-8077	Landmark Graphics
807A	Marra
807B	Dansk Data Elektronik
807C	Merit Intermodel
807D-807F	Vitalink
8080	Vitalink TransLAN III Management

Hex	Information
8081-8083	Counterpoint Computers
809B	EtherTalk(Appletalk over Ethernet
809C-809E	Datability
809F	Spider Systems
80A3	Nixdorf Computers
80A4-80B3	Siemens Gammasonics
80C6	Pacer Software
80C7	Applitek
80C8-80CC	Intergraph
80CD-80D2	Harris
80CF-80D2	Taylor Instrument
80D3-80D4	Rosemount
80DD	Varian
80E0-80E3	Allen-Bradley
80E4-80F0	Datability
80F2	Retix
80F3	AppleTalk Address Resolution Protocol (AARP)
80F4-80F5	Kinetics
80F7	Apollo
80FF-8103	Wellfleet
8107	Symbolics Private
8108	Symbolics Private
8109	Symbolics Private
8137	Novell (old)
8138	Novell
9000	Loopback(Configuration Test Protocol)
9001	Bridge Communications XNS Systems Management
9002	Bridge Communications TCP/IP Systems Management
9003	Bridge Communications
FF00	BBN VITAL-LanBridge cache wakeups %

2. ข้อมูล Ethernet Vendor Addresses

Address	Vendor
00002A	TRW
00005A	S & Koch
000065	Network General
000089	Cayman Systems Gatorbox
000093	Proteon
00009F	Ameristar Technology
0000A9	Network Systems
0000AA	Xerox, Xerox machines
0000B3	CIMLinc
0000C0	Western Digital?
0000DD	Gould
0000E2	Acer Counterpoint
000102	BBN, BBN internal usage (not registered)
001700	Kabel
00AA00	Intel
00DD00	Ungermann-Bass
00DD01	Ungermann-Bass
020701	MICOM/Interian , UNIBUS or QBUS machines,Apollo
020406	BBN,BBN internal usage (not registered)
02608C	3Com, IBM PC,Imagen; Valid
02CF1F	CMC, Masscomp, Silicon Graphics
080002	Bridge
080003	ACC (Advanced Computer Communications)
080005	Symbolics, Symbolics LISP machines
080008	BBN
080009	Hewlett-Packard
08000A	Nestar Systems
080010	AT&T

Address	Vendor
080014	Exceian, BBN Butterfly, Masscomp,Silicon Graphics
080017	NSC
08001A	Data General
08001B	Data General
08001E	Apollo
080020	Sun,Sun machines
080022	NBI
080025	CDC
080028	TI, Explorer
08002B	DEC,UNIBUS or QBUS machines, VAXen , LANBridges(DEUNA, DEQNA, DELUA)
080036	Intergraph,CAE stations
080039	Spider Systems
080047	Sequent
080049	Univation
08004C	Encore
08004E	BICC
08005A	IBM
080067	Comdesign
080068	Ridge
080069	Silicon Graphics
08006E	Excelan
080075	DDE(Danish Data Elektronik A/S)
08007C	Vitalink, TransLAN III
080080	XIOS
080086	Imagen/QMS
080089	Kinetics, Apple Talk-Ethernet interface
08008B	Pyramid
08008D	XyVision, Xy Vision machines

Address	Vendor
AA0003	DEC, Global physical address for some DEC machines
AA0004	DEC, Local logical address for systems running , DECnet

3. ข้อมูล Broadcast Address

Broadcast Address	Type Field	Usage
FF-FF-FF-FF-FF-FF	0600	XNS packets, Hello or gateway search?6 packets every 15 secounds, per XNS station
FF-FF-FF-FF-FF-FF	0800	IP (such as RWHOD via UDP) as needed
FF-FF-FF-FF-FF-FF	0804	CHAOS
FF-FF-FF-FF-FF-FF	0806	IP (Such as RWHOD via UDP) as needed
FF-FF-FF-FF-FF-FF	0BAD	Banyan
FF-FF-FF-FF-FF-FF	1600	VALID packets, Hello or Gateway search?1 packets every 30 seconds, per VALID station
FF-FF-FF-FF-FF-FF	8035	Reverse ARP
FF-FF-FF-FF-FF-FF	807C	Merit Intermodel (INP)
FF-FF-FF-FF-FF-FF	809B	EtherTalk

4. ข้อมูล Ethernet Multicast Addresses

Ethernet Address	Type Field	Usage
Multicast Addresses:		
09-00-09-00-00-01	8005	HP Probe
09-00-2B-00-00-03	8035	DEC Lanbridge Traffic Monitor (LTM)
09-00-2B-00-00-0F	6004	DEC Local Area Transport (LAT)
09-00-2B-01-00-00	8038	DEC LanBridge Copy Packets
09-00-2B-01-00-01	8038	DEC LanBridge Hello packets, 1 packet per second,sent by the designated, LanBridge
AB-00-00-01-00-00	6001	DEC Maintenance Operation protocol(MOP), Remote Console -1 System ID packet every 8-10 minutes,by every:DEC LanBridge,DEC DEUNA interface, DELUA interface, DEC DEQNA interface (in a certain mode)
AB-00-00-03-00-00	6003	DECNET Phase IV end node Hello packets, 1 packet every 15 seconds, sent by the DECNET router
AB-00-00-05-00-00 AB-00-03-FF-FF-FF	????	Reserved DEC through
AB-00-03-00-00-00	6004	DEC Local Area Transport (LAT) - old
AB-00-04-01-xx-yy	6007	DEC Local Area VAX Cluster groups
CF-00-00-00-00-00	9000	Ethernet Configuration Test protocol (Loopback)

ประวัติผู้เขียน

นาย วิเชียร เกียรติ์ขจิตมั่น เกิดเมื่อวันที่ 22 มีนาคม 2512 ที่จังหวัดชลบุรี สำเร็จการศึกษา คณะวิศวกรรมศาสตร์ ได้รับปริญญาอุตสาหกรรมศาสตรบัณฑิต สาขาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2534 และเข้าศึกษาต่อที่บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2537 ปัจจุบัน ทำงานที่บริษัทหลักทรัพย์ เจเอฟ ธนาคม จำกัด ตำแหน่ง วิศวกรระบบอาวุโส

ผลงานทางวิชาการที่ได้รับการตีพิมพ์

[1] วิเชียร เกียรติ์ขจิตมั่น, กอบชัย เดชหาญ, พรพรรณ์ ดุลยกาญจน์, ปิติกันต์ รักราชการ, บัณฑูร พิทักษ์วงศ์ "การตรวจจับ Packet บน IEEE 802 Bus Network" วารสารคอมพิวเตอร์ สมาคมคอมพิวเตอร์แห่งประเทศไทยในพระบรมราชูปถัมภ์, ปีที่ 21 ฉบับที่ 133, หน้า 74-81, พฤษภาคม-มิถุนายน 2538.

[2] วิเชียร เกียรติ์ขจิตมั่น, กอบชัย เดชหาญ "การตรวจจับ อินเตอร์เน็ทโปรโตคอลแพกเก็ต" ได้รับการตอบรับให้ลงตีพิมพ์ใน วารสารคอมพิวเตอร์ สมาคมคอมพิวเตอร์แห่งประเทศไทยในพระบรมราชูปถัมภ์