การเชื่อมต่อระบบ LAN เข้ากับระบบ WAN
LAN TO WAN CONNECTIVITY

โดย
นายฐิติธร เสมาเงิน
นางสาววิลัสณา พูนพัฒน์พิบูลย์

ปริญญานิพนธ์นี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2534

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ ได้จัดทำขึ้นจากการศึกษาระบบ computer network ทั้ง Local Area Network (LAN) และ Wide Area Network (WAN) ในปริญญานิพนธ์ฉบับนี้นั้น ได้กล่าวถึงทฤษฎี และ มาตราฐานต่างๆ ของระบบ computer network ที่จะจำเป็นต้องใช้ และได้รวบรวมวิธีการติดตั้งระบบทั้งหมดเข้าไว้ด้วยกัน

มาตราฐานที่ได้กล่าวไว้ในปริญญานิพนธ์ฉบับนี้ ประกอบด้วย มาตราฐาน OSI (Open System Interconnection) มาตราฐาน 802 มาตราฐาน TCP/IP และ มาตราฐาน X.25 เป็นต้น นอกจากนี้ ได้รวบรวมวิธีการติดตั้งระบบของ TCP/IP ระบบ Network File System ระบบ Network Information Service ระบบ sendmail (SMTP) ระบบ Serial Line IP (SLIP) และ ระบบ X.25

Abstract

This thesis is written about the computer network, both Local Area Network (LAN) and Wide Area Network (WAN). We include the related theories as well as the standards of the computer network. Moreover, we also write about how to setup the system.

The standards that are writen in this thesis is include the Open System Interconnection standard, 802 standard, TCP/IP and X.25. Besides, we also write about the setup of TCP/IP, Network File System, Network Information Service, Sendmail (SMTP),Serial Line IP (SLIP) and X.25

คำนำ

การสื่อสารเป็นสิ่งจำเป็นสำหรับมนุษย์มาช้านาน และเมื่อคอมพิวเตอร์ได้กลายเป็นส่วนหนึ่งสำหรับชีวิตประจำวันของมนุษย์ ไม่ว่าจะเป็นในการศึกษา หรือในวงการธุรกิจ และด้วยความเจริญก้าวหน้าทางเทคโนโลยีของคอมพิวเตอร์ จึงมีการสื่อสารระหว่างคอมพิวเตอร์เกิดขึ้น และรูปแบบหนึ่งที่สำคัญสำหรับการสื่อสาร คือระบบโครงข่ายคอมพิวเตอร์ (Computer network) และสำหรับระบบโครงข่ายคอมพิวเตอร์ที่ติดต่อกันในพื้นที่ที่ไม่ไกลกันมากนัก จะเรียกว่า ระบบโครงข่ายท้องถิ่น (Local Area Network) ซึ่งเป็นรูปแบบที่นิยมใช้กันมากในปัจจุบัน ตั้งแต่ในองค์กรขนาดเล็กจนถึงองค์กรขนาดใหญ่ นอกจากนี้ ระบบ Wide Area Network ก็จำเป็นมากสำหรับระบบการสื่อสารในปัจจุบัน ซึ่งต้องมีการติดต่อกันทั่วโลก

ในโครงงานนี้ จึงได้ทำการศึกษาการเชื่อมต่อทั้งระบบ Local Area Network และ ระบบ Wide Area Network เพื่อเป็นต้นแบบ ให้แก่ผู้ที่ต้องการศึกษาเพิ่มเติม คณะผู้จัดทำ จึงหวังเป็นอย่างยิ่งว่า ปริญญานิพนธ์ฉบับนี้ จะมีคุณค่าและประโยชน์ต่อผู้ที่ศึกษาไม่มากก็น้อย

9 มีนาคม 2535

คณะผู้จัดทำ

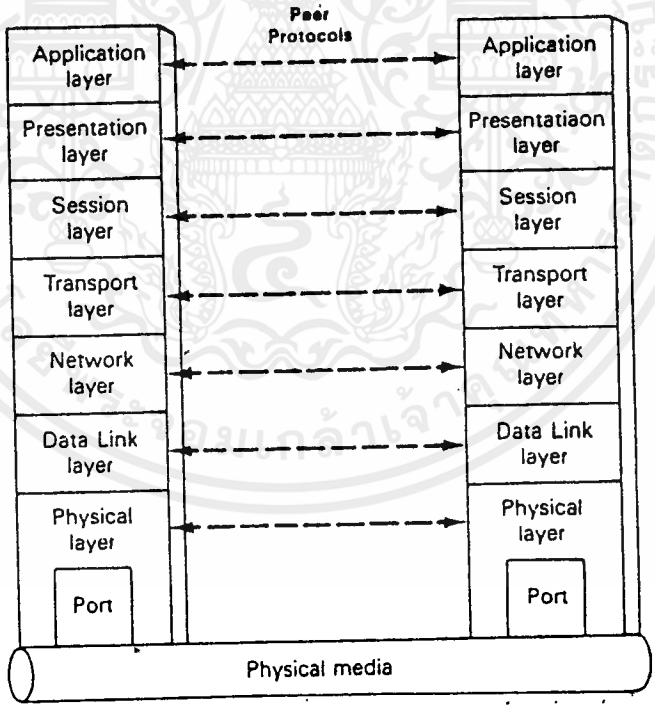
สารบัญ

บทที่ 1

มาตราฐาน OSI

(Open System Interconnection)

ในการสื่อสารข้อมูลของโครงข่ายคอมพิวเตอร์นั้น มีลักษณะเช่นเดียวกับการสื่อสารของมนุษย์ กล่าวคือจะต้องมีการกำหนดระดับ (layer) และข้อตกลง (protocol) ในการสื่อสารให้ตรงกัน รูปแบบของการแบ่งระดับการสื่อสารข้อมูล (layermodel)ที่ใช้กันอย่างแพร่หลายในปัจจุบันคือOpenSystem Intercon-nection (OSI) model ซึ่งได้รับการพัฒนาให้เป็นมาตราฐานสำหรับการสื่อสารคอมพิวเตอร์โดย International Standards Organization (ISO) ดังรูปที่ 1-1 โดยแบ่งออกเป็น 7 layers ดังนี้



รูปที่ 1-1 มาตราฐาน OSI

1. Physical Layer

เป็น layer ล่าสุด ใน layer นี้จะทำการติดต่อโดยตรงกับ phisical media จะอธิบายถึงเฉพาะการติดต่อแบบ point-to-point ระหว่างอุปกรณ์ 2 ตัวและยังกำหนดถึงว่าจะติดต่อโดยใช้ half-duplex หรือ full-duplex ผ่าน serial หรือ parrarell

2. Data Link Layer

ใน layer นี้จะทำการตรวจจับและแก้ไขข้อผิดพลาดต่างๆที่เกิดขึ้นผ่านมาจาก physical layer และยังทำการจัดแบ่งข้อมูลออกเป็น packets หรือ frame ซึ่งรวมเอา frame check sequence เข้าไปในทุกๆ frame ด้วย ในการทำ error correction นั้น เมื่อพบ error ขึ้น layer นี้จะทำการส่งสัญญาณเพื่อขอให้ฝ่ายส่งๆ มาอีกครั้งหนึ่ง

3. Network Layer

เป็น layer ที่จัดการเส้นทางในการส่งข้อมูลระหว่าง open system 2 system โดย layer นี้จะจัดให้มีการ addressing และรับประกันว่า frame ของ data

4. Transport Layer

จัดให้มีการ interface ระหว่าง data communication network กับ 3 layer ข้างบน layer นี้ถูก ออกแบบมาเพื่อแยกผู้ใช้ออกจาก physical

5. Session Layer

เป็น layer ที่อยู่ต่อจาก transport layer ใน layer นี้ จะจัดให้มีการแลกเปลี่ยนข้อมูลระหว่างผู้ใช้ โดยผู้ใช้สามารถเลือกชนิดของการ synchronization และการ control ได้จาก layer นี้

6. presentation layer

จัดให้ข้อมูลต่างๆมีไวยากรณ์ เช่นเมื่อรับข้อมูลต่างๆมาจาก Appication layer จะต้องมีชนิดของข้อมูล (เช่น integer,character) จากนั้น layer นี้จะทำการเปลี่ยนให้เป็น syntax representation (เช่น ASCII) ดังนั้นใน layer นี้จึงต้องมีตารางของ syntax อยู่ด้วย

7. Appication Layer

เป็น layer ที่สนับสนุนการทำงานของผู้ใช้ในงานที่เป็น application ข้อมูลต่างๆใน layer นี้จะต่างกับ presentation layer ที่ว่า layer นี้จะดูแลข้อมูลถึงระดับ semantic และใน layer นี้ได้จัดให้มีบริการต่างให้แก่ผู้ใช้

บทที่ 2
ระบบเครือข่ายท้องถิ่น
(Local Area Network)

ระบบเครือข่ายท้องถิ่นโดยทั่วไปแล้วหมายถึงระบบที่มีคอมพิวเตอร์ที่เชื่อมโยงกันอยู่ในพื้นที่จำกัด โดยที่สามารถติดต่อสื่อสารกันได้ ระบบเครือข่ายท้องถิ่นมีจุดประสงค์ที่จะใช้ Resource ที่มีอยู่ร่วมกัน การวางระบบเครือข่ายท้องถิ่นจะต้องมีการคำนึงถึงรูปแบบการเชื่อมต่อของคอมพิวเตอร์แต่ละเครื่องเข้าด้วยกัน ต่อไปเราจะกล่าวถึงรูปแบบการเชื่อมต่อ

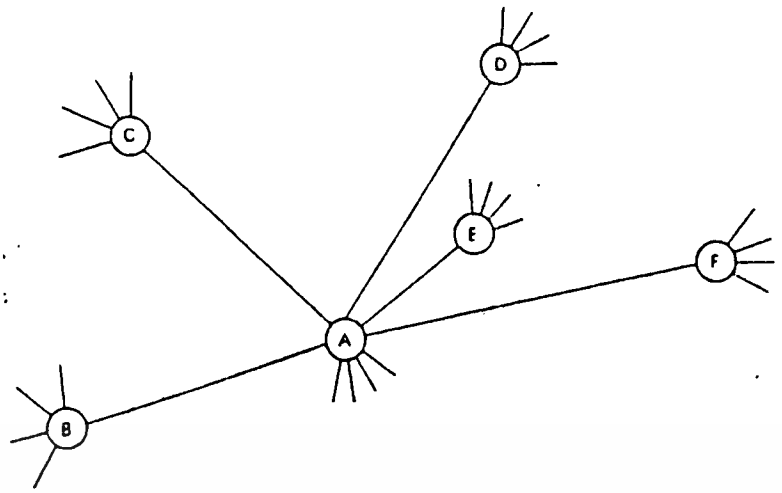
รูปแบบการเชื่อมต่อ (Topology)

1. โครงข่ายแบบดาว

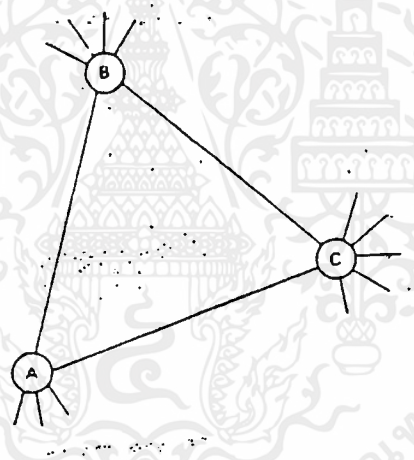
ลักษณะของโครงข่ายคือมีอุปกรณ์ตัวหนึ่ง เป็นศูนย์กลางที่มีสายไปยังอุปกรณ์อื่น การต่อสายของโครงข่ายแบบดาวนี้เราจะใช้สายต่อจากอุปกรณ์กลางไปแบบจุดต่อจุด

2. โครงข่ายแบบวงแหวน

ลักษณะของโครงข่าย คืออุปกรณ์ทั้งหมดจะถูกนำมาต่อกันในลักษณะของวงรอบ (loop) หรือวงแหวน (Ring) การต่อเช่นนี้จะทำให้แม้ว่าเส้นทางใดเกิดขัดข้องก็ยังสามารถใช้เส้นทางอื่นในการส่งข้อมูลแทนได้ ดังนั้นการต่อแบบนี้จะให้ความเชื่อถือได้มากกว่าแบบดาว



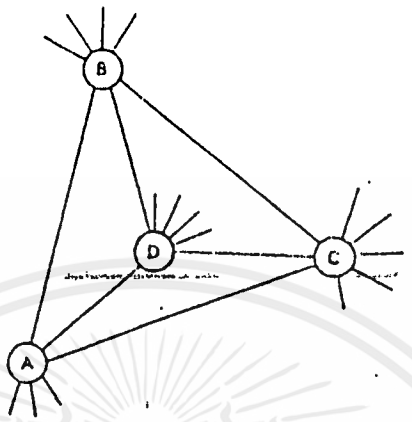
รูปที่ 2-1 โครงข่ายแบบดาว



รูปที่ 2-2 โครงข่ายแบบวงแหวน

3. โครงข่ายแบบตาข่าย

ลักษณะของโครงข่ายคือ จะต่อสายหรือทางเดินข้อมูลระหว่างอุปกรณ์ปลายตัวหนึ่ง ไปยังตัวอื่นๆ ทุกๆตัว ทำให้มีทางเดินของข้อมูลหลายทาง



รูปที่ 2-3 โครงข่ายแบบตาข่าย

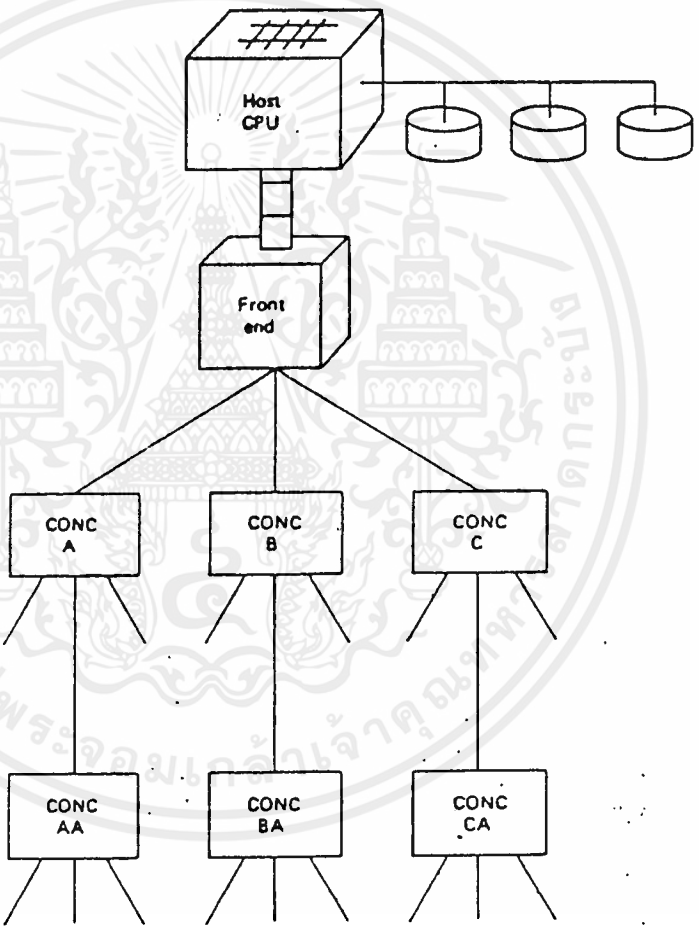
ในการเลือกใช้โครงข่ายว่าจะเป็นโครงข่ายแบบดาว แบบวงแหวน หรือแบบตาข่ายนั้น ขึ้นอยู่กับปัจจัยต่างๆคือ ราคาของสาย สภาพภูมิศาสตร์ของโครงข่าย และจำนวนของข้อมูลที่จะส่งในโครงข่ายนั้นๆ

4. โครงข่ายแบบลำดับขั้น

ลักษณะของโครงข่ายแบบลำดับขั้นแสดงในรูปที่ 1-4 โครงข่ายประกอบด้วยอุปกรณ์ของระบบคอมพิวเตอร์เป็นจำนวนมากหลายๆ ระดับต่อกันอยู่ ลักษณะการต่อแบบนี้จะมีลักษณะเช่นเดียวกับแผนภูมิโครงสร้างขององค์กรหรือหน่วยงานต่างๆ

หากพิจารณาจากโครงข่ายในแบบต่างๆ ที่กล่าวมาแล้วนั้น แบบดาว แบบวงแหวน และแบบตาข่ายนั้นจะพบว่า ความสามารถ ความเชื่อถือได้(Reliability) ของระบบจะสูงขึ้นเมื่อโครงข่ายมีความซับซ้อนมากขึ้น ในโครงข่ายแบบดาวนั้น ถ้าคอมพิวเตอร์ที่เป็นตัวกลางเกิดขัดข้อง ก็จะทำให้ไม่สามารถใช้โครงข่ายได้ ถ้าหากเราต้องการป้องกันก็ต้องเพิ่มคอมพิวเตอร์ สำรองไว้ที่จุดกลางด้วย

configuration.



รูปที่ 2-4 โครงข่ายแบบลำดับขั้น

สำหรับโครงข่ายแบบวงแหวนนั้น ในกรณีที่คอมพิวเตอร์กลางตัวใดตัวหนึ่งเกิดขัดข้องก็จะมีผบกระทบต่ออุปกรณ์ปลายเฉพาะส่วนที่ต่ออยู่กับคอมพิวเตอร์กลางตัวนั้น แต่ในส่วนอื่นของโครงข่ายก็ยังสามารถติดต่อกันได้โดยใช้เส้นทางอื่นของวงแหวน ในทำนองเดียวกัน สำหรับกรณีของโครงข่ายแบบตาข่ายซึ่งมีความแน่นอน และความเชื่อถือได้ของระบบมากกว่า เพราะมีสายต่อเชื่อมโยงภายในมากกว่า

บทที่ 3

มาตราฐาน 802

เมื่อเราทราบว่าเราจะใช้รูปแบบการเชื่อมต่ออย่างไรแล้วจะสามารถเลือกว่าเราจะต้องใช้อุปกรณ์ใด ปัจจุบันมีมาตรฐานการกำหนดอุปกรณ์ที่จะต้องใช้กับ Topology โดยมีตัวอย่างดังนี้

- มาตรฐาน IEEE802.3 CSMA/CD จะใช้กับระบบที่มี Topology แบบ Bus
- มาตรฐาน IEEE 802.4 Token Bus จะใช้กับระบบที่มี Topology แบบ Bus
- มาตรฐาน IEEE 802.5 Token Ring จะใช้กับระบบที่มี Topology แบบ Ring
- ฯลฯ

มาตรฐาน IEEE 802 จะแบ่งออกได้ดังนี้

- Physical จะเกี่ยวกับลักษณะของตัวการที่ใช้รับส่งข้อมูลเช่น พวกลักษณะทางไฟฟ้าของสัญญาณที่ใช้รับส่งข้อมูล, ลักษณะการเชื่อมต่อ เป็นต้น
- Medium Access Control (MAC)จะเกี่ยวกับลักษณะการใช้ตัวกลาง
- Logical Link Control (LLC)จะเกี่ยวกับการสร้าง Logical Link ระหว่างเครื่อง

Medium Access Control

- IEEE 802.3 Carrier Sense Multiple Access/Collision Detection

มีหลักการโดยทั่วไปว่าเครื่องจะใช้ตัวการหรือสายร่วมกัน เนื่องจากต่อในระบบ BUS หรือ Muti-Drops โดยในหนึ่งช่วงเวลาจะมีเพียง 2 เครื่องที่ทำการติดต่อกันหรือใช้สายข้อมูลอยู่ เพราะฉะนั้นก่อนที่เครื่องจะทำการส่งข้อมูลจะต้องทำการตรวจสอบก่อนว่าสายว่างอยู่หรือไม่ ถ้ายังไม่ว่างจะต้องทำการคอยจนกว่าจะว่าง แต่ว่าจะมีสองเครื่องที่ต้องการจะส่งทำการตรวจสอบสายพร้อมกันปรากฏว่าตรวจสอบได้ว่าสายว่างทั้งคู่ก็จะทำการส่งข้อมูลทั้งคู่ทำให้เกิดการชนกันของข้อมูล แต่ละเครื่องก็จะต้องทำการตรวจสอบหลังจากการส่งว่ามีการชนเกิดขึ้นหรือไม่ถ้ามีจะต้องหยุดส่งและทำการ

มาตรฐาน IEEE 802.3 และ CSMA/CD

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) เป็น media access method ชนิดหนึ่ง ที่อนุญาติให้ station 2 station หรือมากกว่าใช้ BUS ร่วมกัน สถานีส่งจะทำการรอจนกว่าสายส่งจะว่าง เมื่อพบว่าสายว่างจะทำการส่งข้อความเป็นในลักษณะของ bit-serial แต่เมื่อมีการชนกันเกิดขึ้นซึ่งเกิดเนื่องจากมีมากกว่า 1 station ส่งข้อมูลมาในเวลาเดียวกัน แต่ละ station จะทำการส่งข้อมูลบางอย่างออกมาเพื่อตรวจสอบว่ามี station อื่นตรวจพบว่ามีการชนกันเกิดขึ้นเช่นกัน ถ้าเป็นจริงจะทำการหยุดส่งเป็นระยะเวลาหนึ่งเรียกว่า Backoff โดยจะหยุดส่งเป็นระยะเวลาเท่าไรนั้นจะทำการ random ขึ้นมาและแต่ละ station จะมี function ในการ random ต่างกัน การทำเช่นนี้เป็นการป้องกันไม่ให้เกิดการชนกันซ้ำอีก มาตรฐาน IEEE 802.3 นี้สามารถใช้ได้กับ media ได้หลายชนิดที่มีความเร็วในการส่งข้อมูลตั้งแต่ 1 ถึง 20 Mb/s

CSMA/CD นั้นมักพบว่าใช้กันมากในระบบโครงข่ายการสื่อสารท้องถิ่น (Local Area Network) โดยมักใช้ใน Ethernet Specification การจัดการของ CSMA/CD นั้นยังอ้างอิงถึงConceptของ Layer Protocol อยู่ดังรูปที่ * โดย CSMA/CD จะครอบคลุม 2 Layer คือ Data Link Layer และ Physical Layer

MAC Sublayer ประกอบด้วย

Transmit Data Encapsulation

- รับข้อมูลจาร LLC (Logical Link Control)
- คำนวนหาค่า CRC แล้วใส่ให้กับ FCS

Transmit Media Access Management

- ส่ง Serail bit Stream ให้กับ physical Layer
- เลื่อนการส่งออกไป ถ้า medium busy
- Halt การส่งถ้าตรวจพบว่าเกิดการชนกันของข้อมูล
- ทำการจัดตารางการส่งใหม่ถ้าเกิดการชนกันของข้อมูล
- จัดการเพิ่ม PAD field เข้าไปกับ LLC
- Enforce การชนกันโดยส่ง jam message

Receive Data Decapsulation

- จัดให้มีการทำการตรวจสอบ CRC
- ยอมรับทุกๆ frame ที่มี address ตรงกับ station นั้น
- ส่งข้อมูลไปให้กับ LLC

Receive Media Access Management

- รับข้อมูลแบบ serial มาจาก physical layer
- ทิ้ง frame ที่มีขนาดน้อยกว่าความยาวต่ำสุด

ส่วนของ physical layer นั้นเป็น medium dependent และเช่นเดียวกับ data link layer คือ physical layer ก็จะประกอบด้วยองค์ประกอบ 2 ส่วนด้วยกันคือ data encoding/decoding entity และ transmit/recieve channel access โดยมี function ที่สำคัญดังนี้

Data Encoding/Decoding

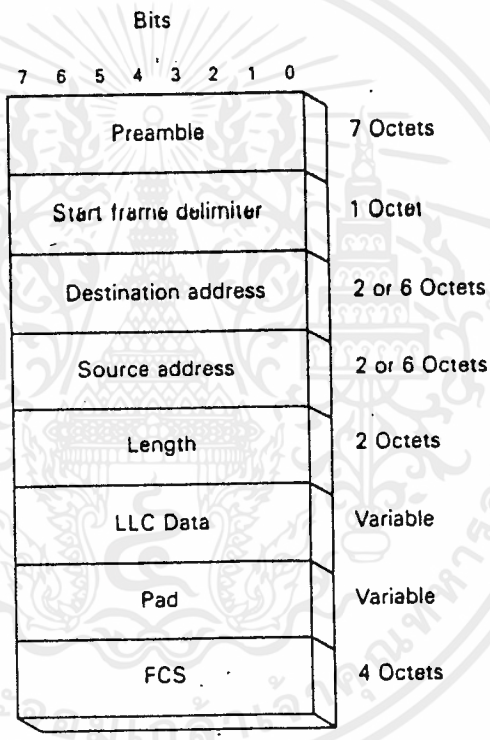
- จัดให้มีการมีการ synchronize สัญญาณระหว่าง station เรียกสัญญาณ syn signal ว่า preamble
- ทำการ encode ข้อมูลแบบ binaryให้มี self-clocking ที่ฝ่ายส่ง และทำการ decode แบบ Manchester code ไปเป็นข้อมูล binary อีกครั้งที่ฝ่ายรับ

Channel Access

- จัดให้มีการติดต่อกับ physical layer สำหรับทั้งฝ่ายส่ง และ ฝ่ายรับ
- Senses สัญญาณพาหะ (carrier) บน channel ของทั้งฝ่ายส่ง และ ฝ่ายรับ เพื่อแสดงว่า channel นั้นยังใช้งานอยู่
- ตรวจสอบการชนกันบน channel ของฝ่ายส่ง

สำหรับใน CSMA/CD network นั้นใน station เดียวกันสามารถเป็นได้ทั้งสถานีส่ง และสถานีรับ โดยจะทำหน้าที่เป็นสถานีส่งเมื่อผู้ใช้ต้องการส่งข้อมูลไปยังสถานีอื่น และทำหน้าที่เป็นสถานีรับเมื่อมีสถานีอื่นส่งข้อมูลมาให้

The CSMA/CD Frame



รูปที่ 3-1 MAC CSMA/CD Frame

ในระดับของ MAC นั้น CSMA/CD frame เป็นดังในรูปที่ 3-1 โดยเริ่มต้นด้วย Preamble และตามด้วย Start frame delimiter เพื่อบอกถึงจุดเริ่มต้นของ frame มี address field บอกถึง source และ destination station

และ length field บอกถึงความยาวของ LLC data field ถ้า data field น้อยกว่าความยาวสูงสุด PAD field จะถูกเพิ่มเข้าไปเพื่อบอกถึงความแตกต่าง นอกจากนี้ยังมี FCS(Frame Check Sequence) โดยใช้ CRC เป็นตัวคำนวน

Thin Wire (ANSI/IEEE 802.3a - 10BASE2)

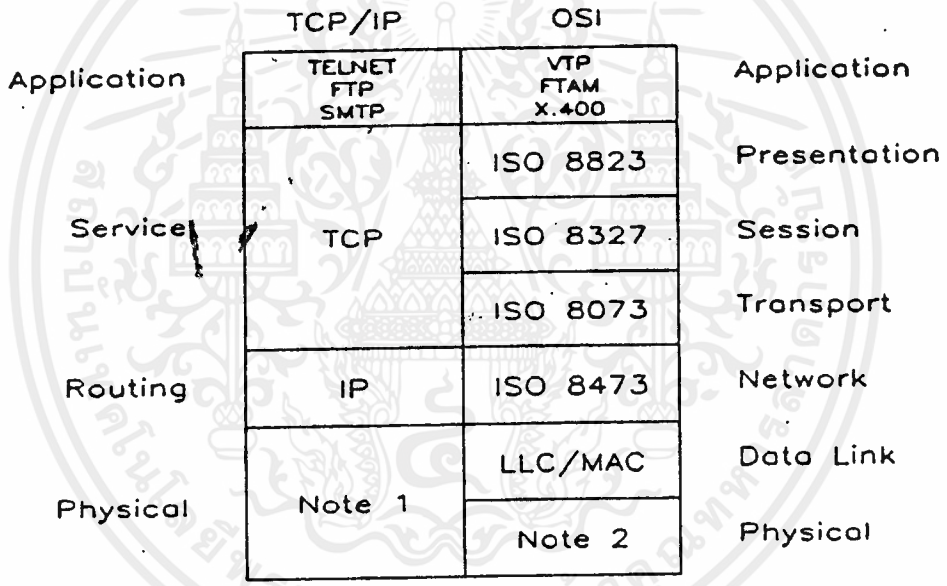
10BASE2 บางครั้งอาจเรียกได้เป็น Cheapernet หรือ Thinnet เป็นมาตรฐานของ IEEE 802.3 ที่มีราคาค่าใช้จ่ายในการติดตั้งน้อย สามารถต่อได้ 30 node ภายในระยะทาง 200 เมตร ได้โดยไม่ต้องมี repeater โดยใช้สาย RG-58 coaxial cabl

บทที่ 4

Transmission Control Protocol/Internet Protocol

(TCP/IP)

แม้ว่าจะมีมาตราฐาน OSI ที่กำหนดโดย ISO แล้วนั้น ก็ยังมีผู้กำหนดมาตราฐานอื่นขึ้นมาอีกเช่นU.S.Departmentof Defense (DoD) ได้ทำการกำหนดมาตราฐานที่เรียกว่า TCP/IP โดยมีลักษณะเป็นชั้นๆเช่นเดียวกับ OSI เช่นกัน แต่ TCP/IP จะแบ่งออกเป็น 4 ชั้นคือ Physical,Routing,Service,Appication รูปที่ 3-1 แสดงถึงการเปรียบเทียบระหว่าง OSI model กับ TCP/IP



NOTES:
1. Some writers suggest that "Ethernet" and "X.25" are the normal standards for TCP/IP. X.25 actually exists at the Network Layer while using HDLC as a Data Link protocol. X.25 and 802.3, therefore, are used only to deliver TCP/IP packets to IP (routing).
2. The Physical Layer in OSI parlance includes various modum standards, IEEE 802.x LANs, and the 8802/7 slotted ring standard.
3. X.400, message handling service; ISO 8823, connection-oriented presentation protocol; ISO 8327, connection-oriented session protocol; ISO 8073, connection-oriented transport protocol; ISO 8473, connectionless protocol; LLC, logical link control; MAC, media access control.

รูปที่ 4-1 TCP/IP - OSI architectures

จากรูปที่ 4-1 จะเห็นได้ว่า Internet Protocol (IP) สามารถเทียบได้กับ Network Layerของ OSI และ TCP อย่างน้อยสามารถเทียบได้กับ Transport Layer หรืออาจเทียบได้กับตั้งแต่ Transport Layer ถึง Presentation Layer ของ OSI และใน Appication Layer ของ TCP/IP นั้นเทียบได้กับ Appication Layer ของ OSI เช่นกัน โดยรวมเอา File Transfer Protocol (FTP) ,Simple Mail Trasfer Protocol (SMTP) และ Terminal Emulation Protocol (Telnet) เข้าไว้ด้วย ในขณะที่ International standard รวมเอา File Transfer,Access and Management (FTAM) ,the Message Handling System (X.400) และ Virtual Terminal Protocol (VTP) เข้าไว้

แม้ว่าเรามักจะพบว่า TCP/IP มักจะถูก implement อยู่บนมาตรฐานของ IEEE 802.3 แต่ความจริงแล้ว TCP/IP เป็น Protocol หนึ่งที่ไม่ยึดติดกับ Physical layer หรือ Datalink Layer ในการจัดแบ่ง Layer นั้นได้จัดเอา TCP และ IP ไว้คนละ layer ทั้งนี้เพราะ เราสามารถใช้ TCP โดยไม่ต้องใช้ IP ก็ได้ เช่นถ้า node A บน network Y ต้องการติดต่อกับ node B บน network Y จะไม่จำเป็นต้องมี router เกิดขึ้นใน IP เช่นถ้า network Y ใช้ IEEE 802.3 802.3 นี้จะทำหน้าที่ดูแลการส่ง frame จาก node A ไปยัง node B แต่ถ้า node A บน network Y ต้องการส่งข้อมูลไปยัง node B บน network X และ network X เป็น X.25 network จะต้องใช้ IP

TCP จะจัดให้มีการทำ packet sequencing,error control และบริการอื่นๆที่จำเป็นสำหรับการสื่อสารแบบ end-to-end ในขณะที่ IP จะนำ packet ที่ได้จาก TCP ผ่าน gateway ต่างๆที่จำเป็น เพื่อส่งไปยัง TCP ของฝ่ายรับที่ได้รับผ่าน IP ของฝ่ายรับเอง

วิธีการที่ข้อมูลส่งจาก node หนึ่งไปยังอีก node หนึ่งใน TCP/IP network แสดงได้ดังรูปที่ 4-2 TCP จะสื่อสารกับ application ผ่าน portที่กำหนด โดยแต่ละ

port จะมี number หรือ address ของมันเองเฉพาะ ในการส่งนั้นแต่ละ layer ต้องทำการเพิ่ม control information เข้าไปด้วยเพื่อที่จะนำไป control ที่สถานีรับ โดยการเพิ่ม header นั้นมีลำดับดังนี้

Application ----------> User Data

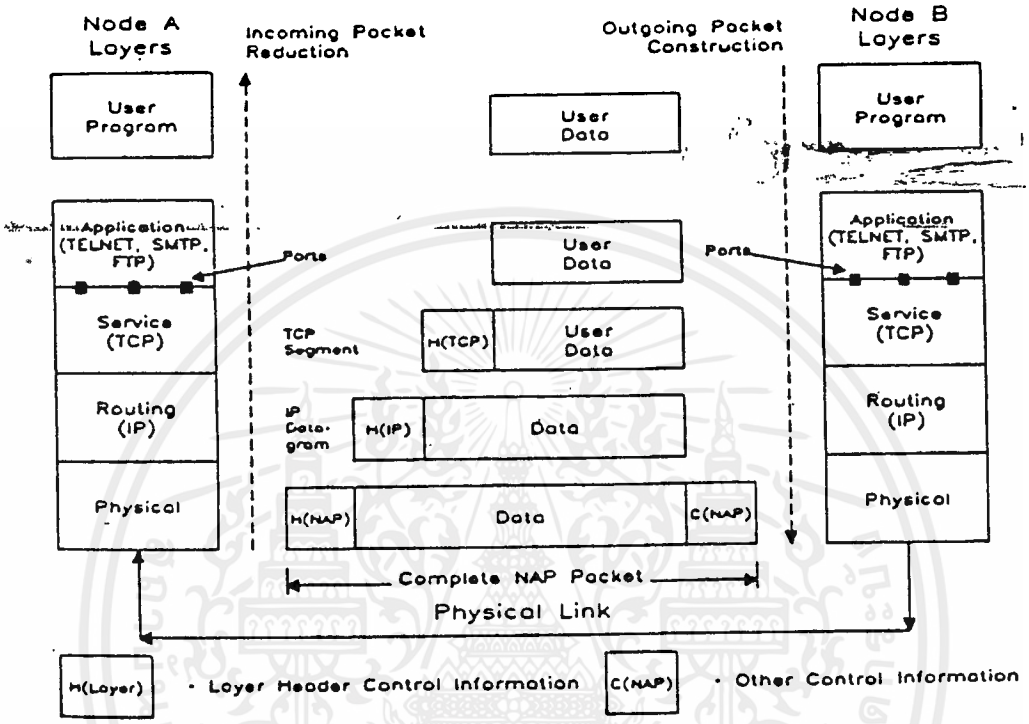
User Data + TCP Header ----------> TCP Segment

TCP Segment + IP Header ----------> IP Datagram

IP datagram + NAP Header ----------> packet

TCP จัดการ User data แล้วเพิ่ม TCP Header เข้าไป โดยรวมเอา destination port ,segment sequence number และ check sum เข้าไปด้วย รวมทั้งหมดเรียกว่า TCP Segment

เมื่อ TCP segment ถูกจัดการแล้ว จะถูกส่งผ่านไปให้กับ IP ซึ่งในชั้นนี้ก็จะมีการเพิ่ม IP header เข้าไปอีกเช่นกัน ข้อมูลที่สำคัญมากที่อยู่ใน IP header คือ Host/node Address ผลที่ได้รวมเรียกว่า IP datagram จากนั้น IP datagram จะถูกส่งไปยัง Physical Layer และที่ physical layer นี้จะมีการเพิ่ม network access protocol เข้าไป ดังนั้นจะได้ packet แล้ว packet นี้จะถูกส่งไปยัง physical medium



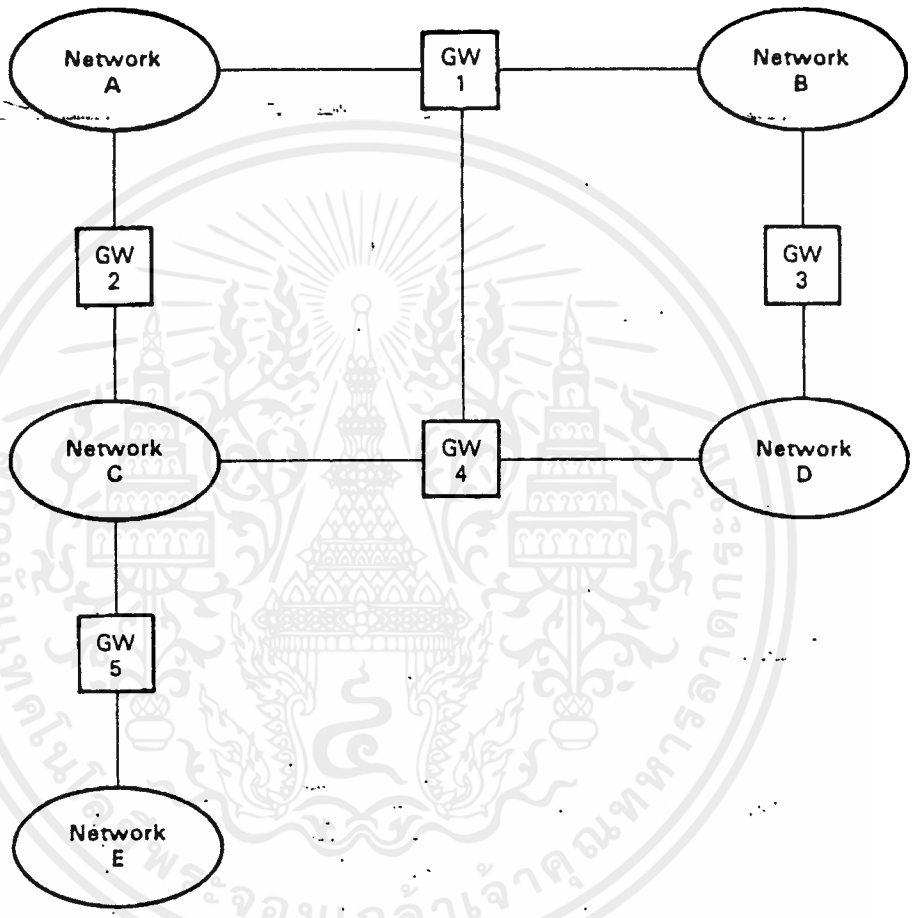
NOTES: H(Layer) denotes Layer Header Control Information [H(TCP), H(IP), H(NAP)]; C(NAP) denotes Other Control Information.

รูปที่ 4-2 Nesting of Internet Layer Protocols

The Internet Protocol (IP)

The Internet protocol (IP) เป็น networking protocol ถูกพัฒนาขึ้นโดย Department of Defense IP เป็นตัวอย่างของการสื่อสารที่ก่อนส่งไม่ต้องทำการ

prior ที่เรียกว่า ทำการ set-up ข้อมูลที่ส่งจาก station หนึ่งจะถูก encapsulate เข้าเป็น IP datagram โดย IP Header จะกำหนด address ของสถานีรับ แล้วทำการส่งให้ IP gateway หลังจากนั้น gateway จะใช้ datagram header กำหนดว่าจะทำการส่งไปทางไหนต่อไป จนกระทั่ง datagram นั้นไปถึงสถานีรับ



รูปที่ 4-3 Internet Protocol และ Gateways

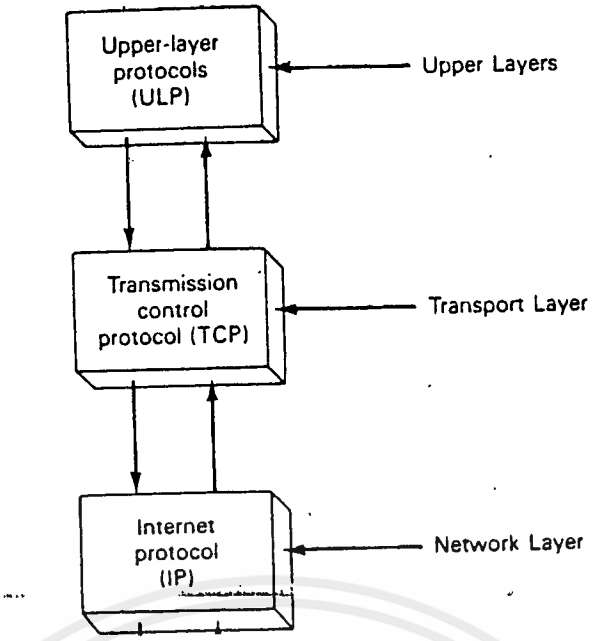
IP Routing

IP Gateway จะต้องทำการตัดสินใจด้วยว่าควรจะทำการ route ไปทางใด เช่น ถ้าสถานีรับอยู่ในคนละ network IP Gateway จะต้องทำการตัดสินว่าควรจะ route อย่างไร

เมื่อพิจารณารูปที่ 4-3 จะเห็นได้ว่า network A สามารถเลือกได้ว่าจะใช้ Gateway 1 หรือ Gateway 2 เพื่อที่จะติดต่อกับ network อื่น ถ้าสถานีรับติดต่อกับ networkserviceโดยตรงผ่านgateway จะเรียกว่า directly connected ถ้าสถานีรับอยู่ใน network ที่ติดกัน โดยติดต่อกันผ่าน gateway จะเรียกว่า neighborgateway ถ้าต้องใช้ gateway มากกว่า 1 gateway ถึงจะสามารถติดต่อกับสถานีรับได้จะเรียกว่า multiple hop sesion

Transmission control protocol (TCP)

TCP เป็น Transport Layer protocol หนึ่งที่นิยมใช้ในปัจจุบัน หน้าที่ของมันคือ ทำการรับและส่งข้อมูลข้าม network boundaries ดังแสดงในรูปที่ 4-4 TCP จะอยู่ที่ transport layer ซึ่งอยู่บน internet protocol(IP)



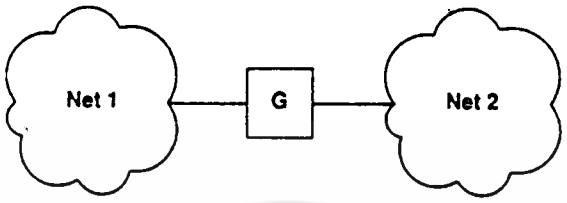
รูปที่ 4-4 Transmission Control Protocol (TCP)

Internet Architecture

เราได้เห็นการเชื่อมต่อของ machine ต่างๆใน network เดียวกันมาแล้ว แต่อาจมีคำถามขึ้นมาว่า แล้วสำหรับ network หลายๆวงจะเชื่อมต่อกันได้อย่างไร คำตอบสำหรับคำถามนี้แบ่งออกได้เป็น 2 ส่วนคือ ในทาง physical แล้ว network 2 วงจะเชื่อมต่อกันได้จะต้องเชื่อมผ่านคอมพิวเตอร์ 1 เครื่องที่ติดต่อทั้ง 2 network นั้น แต่การเชื่อมต่อกันทาง physical อย่างเดียวก็ไม่อาจรับประกันได้ว่า 2 network นั้นได้เชื่อมต่อกัน การที่เราจะให้ 2 network นี้ติดต่อกันได้นั้นต้องอาศัยส่วนสำคัญอีกส่วนหนึ่งซึ่งทำหน้าที่ผ่าน packet จาก network หนึ่งไปยังอีก network หนึ่ง สำหรับคอมพิวเตอร์ที่ทำหน้าที่ทั้ง 2 อย่างนี้เราจะเรียกว่า Internet gateway หรือ Internet router

พิจารณารูปที่ 4-5 แสดงถึง network 2 วง และมี machine G ทำหน้าที่เชื่อมต่อระหว่าง network 1 และ network 2 Machine G ซึ่งทำหน้าที่เป็น

gateway จะต้อง capture packet ของ network 1 ที่ต้องการส่งให้ machine ที่อยู่บน network 2 และทำการ transfer ข้อมูลต่างๆ และทำเช่นเดียวกันสำหรับ packet ที่ต้องการส่งจาก machine ที่อยู่บน network 2 ไปยัง network 1



รูปที่ 4-5 Internet Architecture

Interconnection ผ่าน IP gateway หรือ router

เมื่อมีการเชื่อมต่อที่ซับซ้อนขึ้นระหว่างหลายๆ network นั้น gateway จำเป็นต้องรู้ว่า topology ของ Internet ของแต่ละ network ว่า connect กันอยู่อย่างไร ดังเห็นได้ในรูป 4-6 ซึ่งมี network 3 วงต่อกันโดยใช้ 2 gateways สำหรับตัวอย่างนี้ gateway G1 จะต้องรู้ว่าจะต้องส่ง packet ของ machine 1 ที่ต้องการส่งไปยังทั้ง machine บน network 2 และ 3 จากตัวอย่างจะเป็นว่าเมื่อ network เชื่อมต่อกันเพื่อมากขึ้น ซับซ้อนมากขึ้นจะทำให้ gateway ต้องทำหน้าที่มากขึ้นในการตัดสินใจว่าจะต้องส่ง packet สำหรับ machine บน network ใดบ้าง ดังนั้นจะเห็นได้ว่า gateway จะต้องมี memory ส่วนหนึ่งไม่ว่าจะเป็น primary หรือ secondary memory ก็ตาม ที่จะใช้เก็บว่า machine ใดอยู่ที่ใดบ้างบน Internet ที่มันติดต่ออยู่ แต่สำหรับในระบบ TCP/IP แล้ว เราจะใช้ concept ที่ว่าให้ gateway route packet โดยดูจาก destination network ไม่ใช่ดูจาก destination host ดังนั้นข้อมูลต่างๆที่ gateway จะต้องรู้จึงมีเพียงเท่ากับจำนวน

23

network ใน Internet เท่านั้น ไม่ใช่ต้องรู้เท่ากับจำนวน machine ที่เชื่อมต่ออยู่ทั้งหมด

รูปที่ 4-6

Universal Identifiers

ในระบบการสื่อสารข้อมูลนั้นจะกล่าวได้ว่าสนับสนุน universal communication service ก็ต่อเมื่อมันอนุญาตให้ทุกๆ host สามารถติดต่อกันข้าม host ได้ และในการที่จะให้มันเป็นเช่นนี้ได้ เราจะต้องกำหนดวิธีการที่จะระบุว่ามีคอมพิวเตอร์ตัวใดบ้างที่ติดต่อกันอยู่

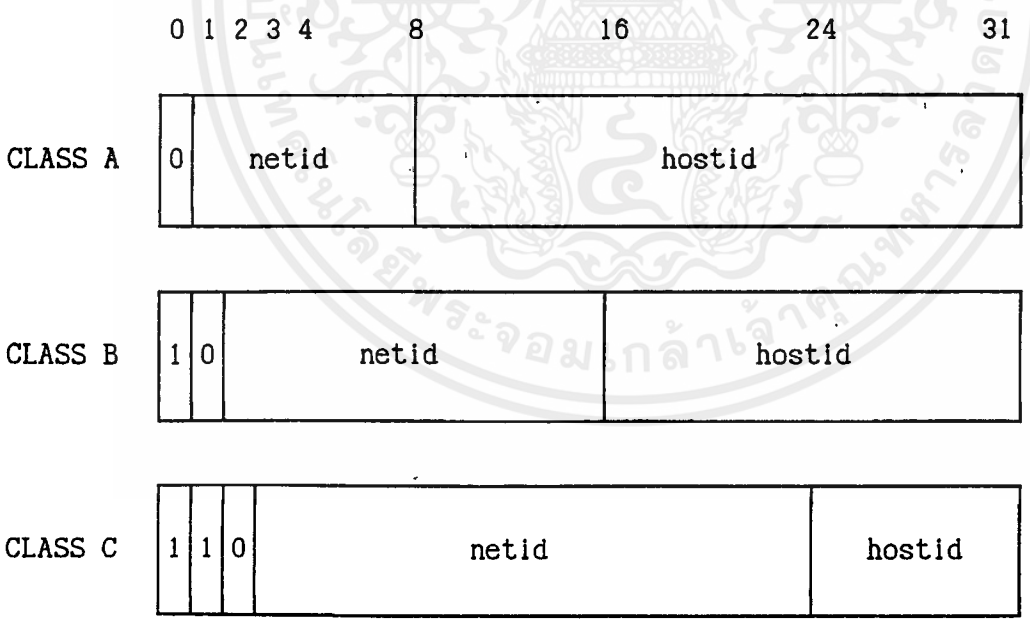
โดยปกติแล้วในการทำ host identifier จะสามารถใช้การ names,addresses หรือ routes โดยส่วนใหญ่แล้วผู้ใช้จะเลือกที่จะใช้การ names ในการ identify machine เพราะเข้าใจง่าย ในขณะที่ software จะสามารถทำงานได้ดีกว่าถ้าใช้การ identify โดย addresses ดังนั้นโดยมาตราฐานแล้วจะเลือกใช้ binary address ในการ route เพื่อให้ได้ประสิทธิภาพที่ดีกว่า

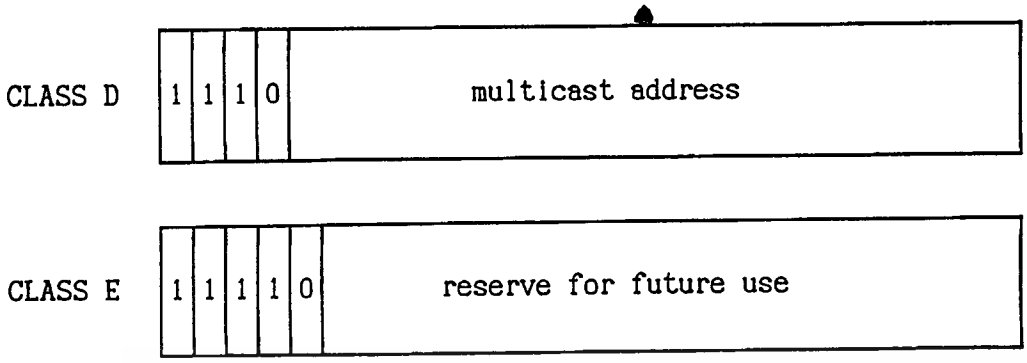
ในการทำ addresses นั้น ผู้ออกแบบ TCP/IP ได้เลือก scheme

ที่สัมพันธ์กับ physical network address สำหรับแต่ละ host ใน Internet โดยกำหนดเป็น address ที่เรียกว่า Internet address หรือ IP address ส่วนสิ่งที่พิเศษของ IP address ก็คือมันได้ถูก encode ให้ identify ทั้ง network และ unique host บน network นั้น

รายละเอียดของ IP address ถูกกำหนดโดยใช้ unique 32-bit internet address โดยแต่ละ host บน network เดียวกันจะใช้ prefix ส่วนที่เป็น network เดียวกัน หรืออาจกล่าวอีกนัยหนึ่งว่าแต่ละ address จะเป็นคู่ของ (netid,hostid) โดยที่ netid จะกำหนด network และ hostid จะกำหนด host บน network นั้น

ในทางปฏิบัติ IP address จะมีลักษณะเป็นดัง 3 class แรกใน รูปที่ 4-7





รูปที่ 4-7

Class ต่างๆ ของ IP address จะกำหนดได้จาก 3 bit แรก (ใช้เพียง 2 bit แรกก็สามารถแยก class ของ 3 class แรกได้)

- Class A จะใช้กับระบบที่มี host มากกว่า 2^{16} ในแต่ละ network โดยใช้ 7 bit สำหรับ netid และ 24 bit สำหรับ hostid
- Class B จะใช้สำหรับ network ขนาดกลางที่มี host ระหว่าง 2^{8} และ 2^{16} ในแต่ละ network โดยใช้ 14 bit สำหรับ netid และ 16 bit สำหรับ hostid
- Class C จะใช้กับระบบที่มี host น้อยกว่า 2^{8} ในแต่ละ network โดยใช้ 21 bit สำหรับ netid และ 8 bit สำหรับ hostid

ในการกำหนด IP address นี้มีข้อพึงสังเกตุคือ hostid 0 จะไม่ถูก assignให้ host ใดเลย แต่ IP address ที่มี hostid เป็น 0 จะใช้ระบุถึงตัว network ของมันเอง ซึ่งสิ่งนี้เองที่เป็นข้อได้เปรียบของ IP address เพราะสามารถ identify ได้ทั้ง host และ network

*p-3X

นอกจากนี้ยังมี broadcast address ที่จะอ้างถึงทุก host ใน network ได้ โดยให้ hostid ทุก bit เป็น 1

เนื่องจากการกำหนด address เป็นแบบ binary นั้นยากแก่การจดจำและเข้าใจของคน ดังนั้นจึงมีการกำหนด Decimal Notation สำหรับ IP address ขึ้น โดยใช้ Decimal Integer 4 ตัวแทน binary address และใช้จุด (".") แยกเลข integer ทั้ง 4 ออกจากกัน ตัวอย่างเช่น

Binary : 10000000 00001010 00000010 00011110

สามารถเขียนได้เป็น

Decimal : 128.10.2.30

สำหรับ network address 127.0.0.0 ใน class A นั้น จะถูก researve ไว้สำหรับ loopback ที่ออกแบบมาเพื่อใช้ test local machine เมื่อ program อ้างถึง loopback address ในการส่งข้อมูล protocal software จะ return data มาโดยไม่มีการส่งผ่าน network ใดๆเลย ดังนั้น packet ใน network ที่จะส่งไปยัง network 127 จึงไม่มี

บทที่ 5
การ setup ระดับพื้นฐานของระบบ TCP/IP

จะมีขั้นตอนในการทำดังต่อไปนี้

1. ทำการตรวจสอบว่า Network interface device ได้ถูกทำการติดตั้งอย่างถูกต้องอยู่บนเครื่องเสียก่อน เช่น ถ้าเป็นเครื่องที่มีโครงสร้างเป็นแบบ PC จะต้องมีการตรวจสอบว่า มี I/O port address, Interrupt request และ memory map มีการ set ถูกต้องตรงตาม ที่กำหนดอยู่ที่ใน driver ของ Network inteface device หรือไม่

2. ทำการตรวจสอบว่า สายที่ใช้ในการเชื่อมต่อถูกต่ออย่างถูกต้องหรือไม่ เช่น ถ้าใช้สายแบบ Thin Ethernet จะต้องทำการตรวจสอบดูว่ามีการต่อ Terminator อย่างถูกต้องหรือไม่

3. ทำการตรวจสอบว่า Network interface device driver ถูกเรียกขึ้นมาใช้งานได้อย่างถูกต้องหรือไม่

4. ทำการตรวจสอบว่า Internet Protocol address (IP address) ของเครื่องมีการกำหนดอย่างถูกต้องหรือไม่ IP address จะแบ่งเป็น 2 ส่วนคือ Network address และ Host address โดยปกติแล้ว IP address จะต้องถูกกำหนดโดย Network Administrator โดยจะต้องมีข้อควรระวังดังนี้

- ส่วนของ Network addrees ของเครื่องจะต้องเหมือนกับ Network addressของ Network ที่เครื่องนั้นต่ออยู่เช่นถ้าเครื่องต่ออยู่บน Network ที่มี IP address 99.0.0.0 (ซึ่งเป็น class A) เพราะฉะนั้นเครื่องที่ต่ออยู่บน Network นี้จะต้องมี Network address ของเครื่อง เป็น 99 ด้วย

- ส่วนของ Host address ของเครื่องที่อยู่บน Network เดียวกันจะไม่ซ้ำกัน โดยปกติและจะไม่ซ้ำกัน ยกเว้นเป็น IP address class D

5. ทำการตรวจสอบว่า IP address ได้ถูกกำหนดให้กับ Network interface device driver ได้ถูกต้องหรือไม่ โดยเครื่องที่เป็นระบบ UNIX ที่ใช้ TCP/IP ส่วนใหญ่จะใช้คำสั่งดังต่อไปนี้

```
ifconfig <device name>
```

เมื่อใช้คำสั่งนี้แล้วเครื่องจะแสดงถึงสถานะของ Network interface device และข้อมูลเกี่ยวกับ IP address และจะสามารถ IP address, netmask และ broadcast address ได้โดยใช้คำสั่งนี้โดย

```
ifconfig <device name> [<IP address>] [netmask <mask>]
                       [broadcast <broadcast address>]
```

เช่น ถ้าต้องการ set device en0 ให้มี IP address เป็น 99.0.0.1 มี netmask เป็น 255.0.0.0 มี broadcast address เป็น 99.255.255.255 ทำได้โดย

```
ifconfig en0 99.0.0.1 netmask 255.0.0.0
ifconfig en0 broadcast 99.255.255.255
```

เมื่อทำการ set แล้วควรจะมีการตรวจสอบอีกครั้งด้วย

6. ทำการตรวจสอบ host table โดยปกติ TCP/IP ที่อยู่บน UNIX host จะอยู่ที่ /etc/hosts ทำการตรวจสอบดูว่า hostname มีอยู่ใน file /etc/hosts หรือไม่ การตรวจ hostname ของ UNIX ทำได้โดยคำสั่ง

hostname

และสามารถกำหนด hostname ได้โดยใช้คำสั่ง

hostname ‹hostname›

เช่น จะกำหนดว่า UNIX host มีชื่อว่า kmitl จะสามารถทำได้โดย

hostname kmitl

หมายเหตุ hostname จะมีลักษณะเป็น case-sensitive

หลังที่ทราบ hostname แล้วทำการตรวจสอบ /etc/hosts ว่ามี hostname นั้นอยู่หรือไม่ โดย format ของ hostname จะเป็นดังต่อไปนี้

‹IP address› ‹hostname› ‹alias 1› ‹alias 2› ... ‹alias n›

ใน /etc/hosts จะต้องมี loopback หรือ localhost อยู่เสมอซึ่งจะมี IP address เป็น 127.0.0.1

127.0.0.1 localhost loopback

7. ทำการตรวจสอบ network โดยใช้คำสั่ง ping โดยที่คำสั่งนี้จะทดลองส่ง packet ไปยังปลายทางที่กำหนด โดยควรจะตรวจสอบดังต่อไปนี้

- ทำการตรวจสอบ localhost หรือ loopback โดยใช้คำสั่ง

ping loopback

ถ้ามีการ setup ถูกต้อง คำสั่งนี้จะมีการตอบกลับมาว่า loopback is alive ซึ่งการตอบกลับมานี้อาจจะไม่เหมือนกันแล้วแต่ระบบที่ใช้ ถ้ามีตอบว่ามี error ตอบกลับมาให้ไปดู ส่วนของ "การแก้ไขปัญหาของระบบ ICP/IP"

- ทำการตรวจสอบโดยใช้ hostname โดยใช้คำสั่ง

ping <hostname>

เช่น มี hostname เป็น kmitl ทำได้โดย

ping kmitl

- ทำการตรวจสอบ host อื่น ๆ ที่อยู่บน network เดียวกัน (มี network address เหมือนกัน) โดยใช้คำสั่งเดียวกับข้างต้น โดยเป็น <hostname> เป็น hostname ของเครื่องที่เราต้องการทดสอบ โดยที่ hostname จะต้องปรากฏอยู่ใน /etc/hosts ด้วย แต่เราสามารถสอบ host ได้อีกทางหนึ่งโดยใช้ IP address โดยใช้คำสั่ง ping <ip address> ซึ่งจะมีผลเหมือนกับการใช้ hostname

8. ถ้าในระบบมีการเชื่อมต่อหลาย network เข้าด้วยกันจะจำเป็นต้องมี bridge ซึ่งจะต้องไปตรวจสอบที่ตัว bridge ว่ามีกำหนด IP address ของ bridge ไว้อย่างถูกต้องหรือไม่ โดยที่ bridge ที่ทำการเชื่อมต่อ network เข้าด้วยกันโดยปกติแล้วจะต้องมี IP address อย่างน้อย 2 address ขึ้นอยู่กับจำนวน network ที่ bridge ตัวนั้นทำหน้าที่เชื่อมต่ออยู่ เช่น ทำการ bridge network

99.0.0.0 เข้ากับ network 89.0.0.0 ที่ bridge จะต้องมี IP address ซึ่งข้างหนึ่งอยู่บน network 99 และอีกข้างหนึ่งอยู่บน network 89 เช่น bridge อาจมี IP address เป็น 89.0.0.10 และ 99.0.0.10 เป็นต้น การ setup ของ bridge ที่ใช้จะแตกต่างไปแล้วแต่ชนิดของแต่ละตัว ใน ภาคผนวกจะมีตัวอย่างของการ setup NetWare File Server Version 3.11 ให้ทำหน้าที่เป็น bridge ของ TCP/IP

เมื่อทำการตรวจสอบบน bridge มีการ setup ที่ถูกต้องแล้วจะต้องมีการ add parameter เข้าไปที่ routing table โดยใช้คำสั่ง

route [add|delete] network gateway hop_count

network - network address ของ ปลายทางที่จะทำการติดต่อ
gateway - IP address ของตัวที่หน้าที่ส่งผ่าน packet
hop_count - จำนวน network ที่จะต้องผ่านเพื่อที่จะไปให้ถึงปลายทาง

เช่น จะต้องการ add network 89.0.0.0 เข้าไปที่ routing table ของเครื่องที่มี IP address เป็น 99.0.0.1 โดยที่ bridge มี IP address เป็น 99.0.0.10 และ 89.0.0.10 จะใช้คำสั่งต่อไปนี้

route add 89.0.0.0 99.0.0.10 1

สังเกตุว่าจะใช้ IP address ของ bridge ที่อยู่ network เดียวกับ host ที่จะทำการ add อยู่ และมี hop_count เป็น 1 เนื่องจากมีการข้าม network 1 network เมื่อเราทำการ add แล้วเราสามารถดู routing table ได้โดยใช้คำสั่ง

netstat -r

เมื่อทำการ add network เรียบร้อยแล้วสามารถทำการทดสอบได้โดยคำสั่ง ping โดยกำหนด IP address เป็น address ของ host ที่อยู่บน network ถ้ามี bridge มากกว่า 1 ตัว ก็จะทำในลักษณะเดียวกัน

หลังจากได้ทำการปฏิบัติตามข้อ 1-8 แล้วเราจะสามารถใช้คำสั่งพื้นฐานของระบบ TCP/IP ได้เช่น rlogin, ftp, telnet เป็นต้น

หมายเหตุ ถ้ามีการข้อผิดพลาดขอให้ดูในส่วนของ "การแก้ไขปัญหาของระบบ TCP/IP"

การแก้ไขปัญหาของระบบ TCP/IP

ปัญหาที่เกิดขึ้นในระบบ TCP/IP ที่พบส่วนมากมีดังต่อไปนี้

อาการ

ไม่สามารถติดต่อกับ Network interface device ได้

วิธีแก้ไข

- ทำการตรวจว่า Network interface device มีสถานะ UP อยู่หรือไม่โดยใช้คำสั่ง ถ้าไม่ ให้ทำการ UP

ifconfig <device name> [up|down]

- ทำการตรวจสอบว่า IP address, netmask, broadcast address ตรงตามที่กำหนดไว้หรือไม่ ถ้าไม่ทำการแก้ไขให้ถูกต้องโดยใช้คำสั่ง ifconfig

- ทำการทดสอบด้วยคำสั่ง ping ‹IP address¦hostname› ของ host ที่กำลังทำอยู่

อาการ

ไม่สามารถติดต่อกับ host อื่นที่อยู่ภายใน network เดียวกันได้

วิธีแก้ไข

- ทำการตรวจว่า Network interface device มีสถานะ UP อยู่หรือไม่โดยใช้คำสั่ง ถ้าไม่ ให้ทำการ UP

ifconfig ‹device name› [up¦down]

- ทำการตรวจสอบว่า IP address, netmask, broadcast address ตรงตามที่กำหนดไว้หรือไม่ ถ้าไม่ทำการแก้ไขให้ถูกต้องโดยใช้คำสั่ง ifconfig

- ทำการตรวจสอบ IP address ว่าถูกต้องหรือไม่ คือ Network address มีค่าเดียวกับ Network ที่ต่ออยู่ และ Host address จะต้องไม่ซ้ำกับ host อื่น ๆ ใน network เดียวกัน

- ทำการทดสอบด้วยคำสั่ง ping ‹IP address¦hostname› host อื่นที่อยู่บน network เดียวกัน

อาการ

ไม่สามารถติดต่อกับ host อื่นที่อยู่ใน network อื่นได้

วิธีแก้ไข

- ทำการตรวจว่า Network interface device มีสถานะ UP อยู่หรือไม่โดยใช้คำสั่ง ถ้าไม่ ให้ทำการ UP

ifconfig <device name> [up|down]

- ทำการตรวจสอบว่า IP address, netmask, broadcast address ตรงตามที่กำหนดไว้หรือไม่ ถ้าไม่ทำการแก้ไขให้ถูกต้องโดยใช้คำสั่ง ifconfig

- ทำการตรวจสอบ IP address ว่าถูกต้องหรือไม่ คือ Network address มีค่าเดียวกับ Network ที่ต่ออยู่ และ Host address จะต้องไม่ซ้ำกับ host อื่น ๆ ใน network เดียวกัน

- ทำการทดสอบด้วยคำสั่ง ping <IP address|hostname> host อื่นที่อยู่บน network เดียวกันก่อน แล้วจึงทำการ ping ไปที่ bridge ให้สังเกตุว่ามี respond จากทั้งสองข้างของ bridge หรือไม่ ถ้า bridge ทำงานได้ถูกต้องจะต้องสามารถ ping ทั้งสองข้างซึ่งต่ออยู่คนละ network ได้

- ทำการตรวจสอบ routing table ว่ามีสถานะเป็นอย่างไรโดยใช้คำสั่ง

netstat -r

35

ให้สังเกตุว่ามี network address ปลายทางอยู่ใน routing table หรือไม่ ถ้าไม่มีให้ add เพิ่มโดยใช้คำสั่ง

route add <destination network address> <gateway> hop_count

- ทำการตรวจสอบว่ามี routed ทำงานเป็น daemon process อยู่หรือไม่ โดยใช้คำสั่ง

ps -ef | grep routed

ถ้าไม่ได้ทำงานอยู่ให้เรียกขึ้นมาทำงาน

บทที่ 6

The Network File System

The Network File System(NFS) เป็นตัวช่วยในการอำนวยความสะดวกในการใช้แฟ้มข้อมูลร่วมกันระหว่างคอมพิวเตอร์หลายๆ เครื่อง หลายๆ ระบบปฏิบัติการ หรือหลายๆ โครงข่าย

บริการที่ NFS มีให้คือ อนุญาติให้ผู้ใช้ทำการ mount filesystem ผ่านโครงข่ายการสื่อสารข้อมูล จากนั้นจะดูแล remote file เป็นเหมือนกับ local file เมื่อ filesystem ได้ถูกใช้ร่วมกัน ผู้ใช้ที่แต่ละเครื่องจะสามารถเข้าถึง filesystem ได้เหมือนกับเป็น local machine นอกจากนี้ remote file สามารถถูก execute ได้ถ้ามันถูก compile สำหรับ local machine

ลักษณะการออกแบบของ NFS

ในการออกแบบ NFS จะรวมลักษณะเหล่านี้เข้าไว้ด้วย

- ผู้ใช้สามารถ access develop file โดยตรง โดยเหมือนกับไม่มีความแตกต่างระหว่างการ read และ write file ที่อยู่ใน local machine และการ read หรือ write file บน remote machine หรือกล่าวได้ว่า ข้อมูลในโครงข่ายการสื่อสารข้อมูลเป็นแบบ distributed จริงๆ

- NFS จะอนุญาติให้มีการแลกเปลี่ยนข้อมูลระหว่างเครื่องต่างชนิดกัน หรือระบบปฏิบัติการต่างชนิดกันได้ NFS ได้จัดให้มีบริการโครงข่ายข้อมูลที่อนุญาติให้ software ใหม่สามารถใช้ได้โดยไม่รบกวนต่อ software เดิม

- file server protocol ได้ถูกออกแบบมาโดย client machine สามารถทำงานต่อไปได้ แม้ว่า server จะ crash

- การบริหารงานของ network ขนาดใหญ่อาจจะซับซ้อนและเสียเวลามาก NFS สามารถถูกออกแบบให้กลุ่มของโครงข่ายการสื่อสารข้อมูลนั้นยุ่งยากในการบริหารงานน้อยกว่า กลุ่มของ local file system NFS ได้รวบรวมเอา utility บางอย่างสำหรับโครงข่ายการบริหารงาน แต่อย่างไรก็ตาม UNIX utilities ก็ยังสามารถใช้ประโยชน์ได้

NFS ได้เพิ่มความสามารถต่างๆ ที่ช่วยส่งเสริมการสื่อสารข้อมูลให้มีประสิทธิภาพมากขึ้น เช่น มีบริการ asinchronous สำหรับ multiple requests , catching ของ these blocks และ asinchronous

NFS SERVICE

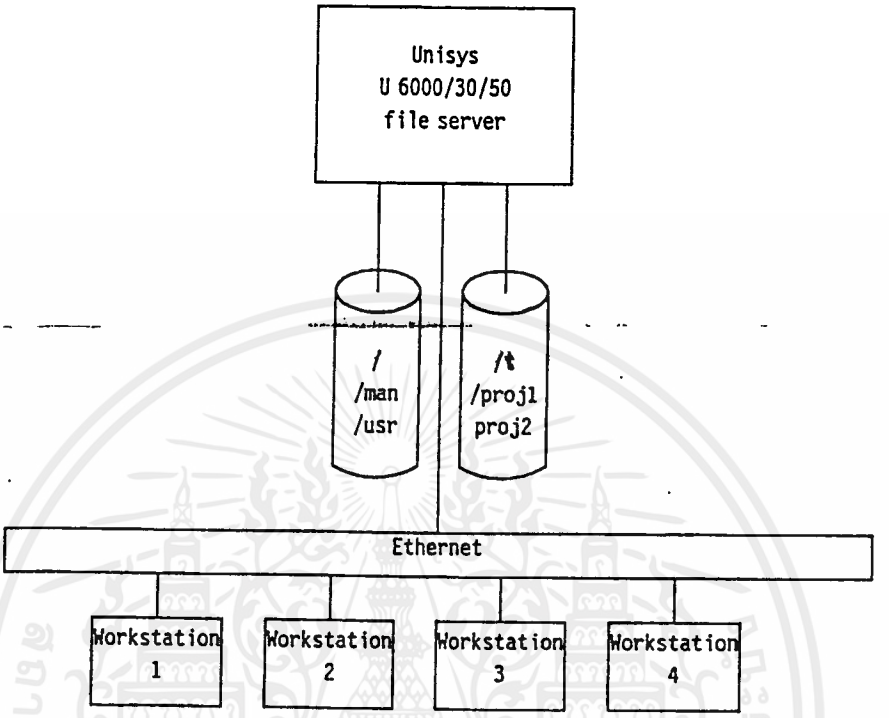
เนื่องจากการให้บริการของ NFS เป็นลักษณะของการ care file กล่าวคือ ผู้ใช้สามารถ access remote file ได้เหมือนกับเป็น local file ในการให้บริการเช่นนี้จะต้องประกอบด้วยกระบวนการ 2 อย่างคือ

1. filesystem ที่จะถูก share จะต้องถูก export ให้สำหรับ machine ที่กำหนดหรือ netgroup

2. filesystem จะถูก remote-mounted โดย machine ที่ต้องการ access file นั้น โดยอาจ access พร้อมกันหลาย machine ก็ได้

ในโครงข่ายการสื่อสารข้อมูล แต่ละ machine จะมี filesystem แตกต่างกันไป ดังรูปที่ 6-1

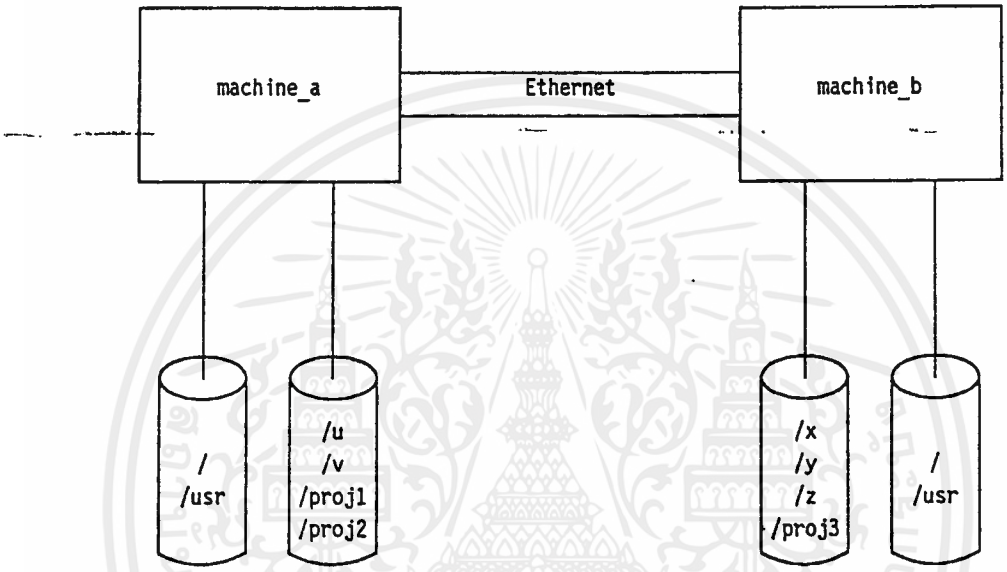
38



รูปที่ 6-1 NSF system environment

เมื่อใช้ NFS filesystem บน machine A สามารถให้ผู้ใช้บน machine B ใช้ร่วมได้ เพราะ filesystem บน machine B ก็สามารถให้ผู้ใช้ machine A ใช้ร่วมได้ เช่น ถ้า Data Base เก็บไว้ใน machine A ก็สามารถให้ผู้ใช้ machine B มาใช้ Dara Base นี้ได้

ในการเชื่อมต่อ NFS file server เพื่อให้กลุ่มของ work station ใช้ได้นั้น จะเชื่อมต่อกันผ่าน Ethernet ดังรูปที่ 6-2



รูปที่ 6-2 NFS work station environment

โดยการเชื่อมต่อแบบนี้ทำให้ work station สามารถใช้ file ได้โดยไม่ต้อง copy มา ถ้า filesystem บน server ถูก remote-mounted โดยแต่ละ work station นอกจากนี้แต่ละ work station สามารถ mount เพียงบาง file system ที่ต้องการใช้ก็ได้ เช่น mount เฉพาะ /proj2

ในการ mount file นี้สามารถทำได้ 2 วิธี วิธีที่ 1 เป็นแบบ mannually คือการใช้คำสั่ง mount หรือวิธีที่2 เป็นแบบ automatically คือการใส่

entries ไว้ใน /etc/fstab machine ที่ทำการ export file system นั้นจะเรียกว่า file server ส่วน mechine ที่ mount file system มาจะเรียกว่า client แต่ในขณะเดียวกัน machineเดียวกันสามารถเป็นทั้ง server และ client ในขณะเดียวกันได้

Exporting a Filesystem

ในการทำการ exporting สามารถทำได้โดยการแก้ไข file /etc/exports โดยการแก้ไข เพิ่มเติมชื่อ file system ที่จะอนุญาติให้ machine อื่นใช้ได้ แต่ file system ที่จะระบุได้นั้น จะต้องเป็น directory เท่านั้น เช่นถ้าต้องการ export file system /proj1 จาก machine_a ไปให้ machine_b ใช้ และต้องการ export file system /proj2 จาก machine_a ไปให้ทุกๆ machine ในโครงข่ายใช้สามารถทำได้โดยเพิ่มบรรทัดเหล่านี้เข้าไปใน file /etc/exports ของ machine_a

```
/proj1  machine_b   # for machine_b users only
/proj2              # for all network machines
```

หรืออาจ export ให้สำหรับบาง netgroup ก็ได้เช่น ถ้าต้องการ export /proj1 ให้ netgroup develop สามารถทำได้โดยเพิ่มบรรทัดข้างล่างนี้ลงไป

```
/proj1  develop     # for develop netgroup machines
```

Remote-Mounting a Filesystem

การทำ remote mount นั้นสามารถทำได้ทั้งโดย แบบ manually หรือ แบบ automatically ในการทำ automatic remote-mount นั้นผู้ทำจะต้องเป็น superuser โดยทำใน file /etc/fstab ของ client machine โดย file system จะถูก mount เวลา boot เครื่อง

ในการ mount file แบบ manually เริ่มด้วยการสร้าง directory ว่างขึ้นมา จากนั้น execute คำสั่ง mount คำสั่งนี้จะถูกยกเลิกเมื่อเครื่องถูก boot ใหม่

คำสั่ง mount มีรูปแบบการใช้งานดังนี้

```
# mount -f NFS machine_name:file system mount_point
```

เมื่อใช้ตัวอย่างเช่นเดียวกับข้างต้น ถ้าต้องการ remote-mount /proj1 file system ที่ export โดย machine_a สามารถทำได้โดยขั้นตอนดังนี้

1. สร้าง directory ว่างๆขึ้นที่ machine_b เช่นสร้าง directory /proj1 โดย directory นี้จะเป็น mount point ของ file system ที่ mount

2. Execute คำสั่งต่อไปนี้ที่ machine_b:

```
# mount -f NFS machine_a:/proj1 /proj1
```

เมื่อคำสั่งนี้ execute เรียบร้อยแล้ว user บน machine_b สา
มารถใช้ /proj1 file system ของ machine_a ได้

Network Information Service

ระบบ Network Information Service ที่จะกล่าวถึงในที่นี้คือระบบที่จะช่วยในการค้นหาข้อมูล โดยในระบบ TCP/IP มีโปรแกรมที่ใช้กันอย่างแพร่หลายและเป็นที่รู้จักกันคือ Yellow Pages ของ Sun โดยจะมีการค้นหาข้อมูลที่ไม่ได้อยู่ใน local ได้ โดยจะมีส่วนประกอบต่าง ๆ ดังต่อไปนี้

- domains
- maps
- daemons
    - ypserv (Server Process)
    - ypbind (Binding Process)
    - ypxferd (High Speed Map Transfer daemon)
- utility
    - ypcat (List data in a map)
    - ypwhich (List name of NIS server and maps served)
    - ypmatch (Match a key to its value in a map)
    - ypinit (Build and install an NIS database)
    - yppoll (Get a map's order number from a server)
    - yppush (Propagate data from master to slave NIS server)
    - ypset (Set binding to a particular server)
    - ypxfer (Transfer data from master to slave NIS server)
    - makedbm (Create dbm file for an NIS map)

ลักษณะต่าง ๆ ของ NIS

1. NIS service จะใช้ข้อมูลที่อยู่ใน NIS map ซึ่ง map เหล่านี้ได้มาจาก text file ใน /etc directory ซึ่งจะเก็บข้อมูลเกี่ยวกับ network ซึ่งใน network จะต้องมี NIS server อย่างน้อย 1 ตัวในแต่ละ domain ซึ่งจะทำการจัดการเกี่ยวกับ map สำหรับ host ตัวอื่น ๆ

2. NIS domain คือ set ของ map ที่ใช้ร่วมกัน ซึ่งแต่ละ domain จะมีชื่อ (domain name) ของตัวเอง ซึ่งถ้าเครื่องอยู่ใน domain เดียวกันจะใช้ map ร่วมกัน

3. ชนิดต่าง ๆ ของ host ใน NIS

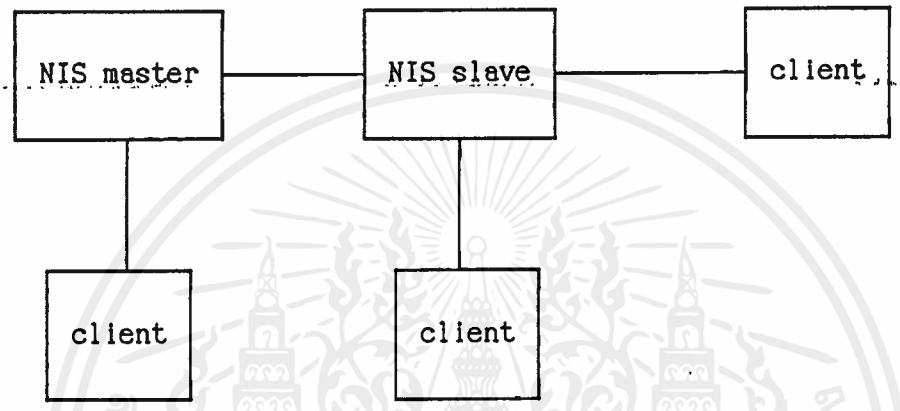
- Master Server
- Slave Server
- Client

ซึ่งตัวที่เป็น Master Server และ Slave Server จะสามารถทำตัวเป็น Client ได้ในขณะเดียวกัน

NIS Server เป็น host ที่เก็บ NIS maps เพื่อใน host อื่นใน network สามารถทำการเรียกหาข้อมูลจากมันได้ โดย NIS Server มี 2 แบบ คือ Master Server และ Slave Server ซึ่ง Master Server จะทำหน้าที่เก็บ map ต่าง ๆ ซึ่งการเปลี่ยนแปลง ควรจะทำที่ Master Server เท่านั้น ซึ่งใน แต่ละ domain จะสามารถมี Master Server ได้ 1 ตัวเท่านั้น Slave Server จะเก็บ map ซึ่งเหมือนกับที่ Master Server ซึ่งเมื่อใดก็ตามที่มีการเปลี่ยนแปลง map ที่ Master Server จะทำการส่งมาเปลี่ยนแปลงที่ Slave Server ด้วย ซึ่งในแต่ละ domain ไม่จำกัดจำนวน Slaver Server ซึ่งตัว

Slave Server จะเป็นตัวแบ่งเบา load จาก Master Server

NIS Client เป็น Process ซึ่งทำหน้าที่ request ข้อมูลจาก NIS Server ซึ่งจะไม่สนใจว่าเป็น Server ตัวใดแต่จะต้องอยู่ใน domain เดียวกัน



รูปที่ 7-1 แสดงการค้นหาข้อมูลโดยใช้ Network Information Service

บทที่ 7

Sendmail

Sendmail เป็นโปรแกรมที่ช่วยการทำงานในระบบ mail เพื่อช่วยในการ routing นอกจากนี้ยังสามารถช่วยในการทำ alias, forward mail และการหาทาง routing ไปบน network ผ่าน gateway ได้

Sendmail จะมีประโยชน์มากเมื่อระบบภายในโครงข่ายประกอบด้วยหลายๆ ระบบเช่นมีทั้ง UUCP และ SMIP โดยเราจะใช้ sendmail นี้เป็นตัว convert ระหว่างระบบดังกล่าว

เมื่อเริ่มต้นในการ startup ระบบ sendmail ถ้าต้องการให้ startup ทุกครั้งที่เปิดเครื่องให้ไปทำการ edit file /etc/init.d/bsd ดังนี้

```
#STARTDEMONS='inetd rwhod routed'
#STOPDEMONS='inetd rwhod routed'

#Use  the following parameters instead if you have sendmail
#install.

STARTDEMONS='inetd rwhod routed smtpd'
STOPDEMONS='inetd rwhod routed sendmail'
```

เมื่อมีการ startup sendmail มันจะไปอ่าน file /usr/lib /sendmail.cf เพื่อจะดู configuration ต่างๆ ดังตัวอย่าง file sendmail ที่ปรากฏต่อไปนี้

```
######################################################################
######################################################################
#####                                                            #####
#####          SENDMAIL CONFIGURATION FILE                       #####
#####                                                            #####
######################################################################
######################################################################

#####  @(#)$Revision: 16.4 $

######################################################################
######################################################################
#####                                                            #####
#####                    CAVEAT EMPTOR                           #####
#####                                                            #####
#####  This configuration file is suitable for use on            #####
#####  most HP-UX systems and can be installed,                  #####
#####  unmodified, as sendmail's configuration file,             #####
#####  /usr/lib/sendmail.cf                                      #####
#####                                                            #####
#####  Modifications that reflect this host's mail               #####
#####  environment may be needed.  HP recommends that            #####
#####  these be limited to the "Localizations" and               #####
#####  "Routing Options" described below.                        #####
#####                                                            #####
#####                                                            #####
#####  HP will provide support for this configuration            #####
#####  file if:                                                  #####
```

```
#####                                                          #####
#####   * it is left unmodified; or                            #####
#####                                                          #####
#####   * the only changes made are those described            #####
#####     below under "Localizations" and "Routing             #####
#####     Options".                                            #####
#####                                                          #####
#####  If other changes are made to this file, you             #####
#####  are on your own.                                        #####
#####                                                          #####
##################################################################
##################################################################

######################################################################
#  Localizations:
######################################################################
#
#       While this configuration file can be used unmodified, it may be
#       appropriate to make some or all of the following changes to this
#       configuration file to incorporate local information or preferences
#
#       Detailed descriptions of sendmail.cf syntax and complete list of
#       the configuration options can be found in the documentation.
#
#  Use the Nameserver                   (option I):
#
#       If this host is using the domain nameserver, the I option should
#       be set.  In most cases, sendmail will be able to detect whether
```

```
#       the nameserver is in use and will automatically do the right
#       thing.  However, in some cases it will incorrectly appear to
#       sendmail as though the nameserver is not being used, for example
#       if the nameserver has not finished loading its data and is not yet
#       responding to queries.  If this host normally uses the nameserver,
#       messages should be deferred when the nameserver is not running.
#
#       If the I option is set, if an MX record lookup or gethostbyname()
#       call fails because the nameserver is not running, sendmail will
#       defer the message.
#
#       If the I option is not set, if an MX record lookup fails, sendmail
#       will assume that the nameserver is not being used, and that there
#       are no MX records, and try to connect directly to the host.  If a
#       host name lookup fails, sendmail will assume that there is no
#       entry for the host and return an error.
#
#       By default, the I option is not set.  To set it, uncomment the
#       line:
#
#         # OI
#
# Site Hiding                           (macro Y):
#
#       It may be desirable to "hide" the name of a host generating SMTP
#       mail, so that replies need not be directed to the particular host
#       on which the message originated.  This simplifies moving a user's
#       mail home from one host to another, and gives the appearance of
```

```
#       mail being generated by a unified organization rather than a
#       collection of hosts.
#
#       By default the sender of a message routed via SMTP is identified
#       with the header line:
#
#         From: Full Name <user@host>
#
#       where "host" is the canonical name of the local host (macro w).
#
#       If the macro Y is defined, the value of macro Y will be appended
#       instead of the local host name.  The macro Y would typically be
#       defined either as the name of a local mail hub, or as the local
#       domain name.  Although the name need not be the name of an actual
#       host, this scheme requires that some host recognize the name and
#       be able to route mail to the originating user.
#
#       For example, to identify SMTP mail from the local host as
#       originating from the domain poo.bah.edu, Y should be defined by
#       replacing the line:
#
#         # DY
#
#       with:
#
#         DYpoo.bah.edu
#
#       An outgoing SMTP message from user "fred" at the host "burb"
```

```
#       would have the From: header line:
#
#         From: Fred User <fred@poo.bah.edu>
#
#       Assume that the host "porpoise.poo.bah.edu" is to accept and
#       forward mail for the domain poo.bah.edu.  In order for users at
#       other hosts to reply to such messages, there are two things that
#       must be done:
#
#          1) Replying hosts must be able to map the name poo.bah.edu to an
#             internet address or MX record.
#
#          2) Some host must be prepared to recognize mail addressed to
#             "user@poo.bah.edu" as local, and be able to route such mail
#             appropriately.
#
#       To satisfy 1), the nameserver for the domain bah.edu must have
#       either an MX record for the name "poo" directing its mail to the
#       host porpoise.poo.bah.edu, or an address record mapping the name
#       "poo" to an internet address belonging to porpoise.poo.bah.edu.
#       Do not make "poo.bah.edu" a CNAME for porpoise.poo.bah.edu,
#       because in that case a recipient may canonicalize the sender
#       address "user@poo.bah.edu" to "user@porpoise.poo.bah.edu", which
#       defeats the purpose of site hiding.  For the same reason, if a
#       replying host uses /etc/hosts or NIS (formerly known as Yellow
#       Pages) to do host name to address mapping, the entry for
#       poo.bah.edu should be a separate line, not a host name alias.
#
```

```
#      To satisfy 2), on porpoise, the name "poo.bah.edu" must be added
#      to class w (see below), and there must be mail aliases for all
#      users on systems with macro Y defined as above.  For example:
#
#        fred:  fred@burb.poo.bah.edu
#
#  Aliases for Local Host (class w):
#
#      This configuration will automatically recognize that mail to users
#      at the local host or any of its host name aliases or CNAMEs should
#      be delivered locally.  Other host names that you wish to recognize
#      as local can be added to class w as either a simple class
#      definition (Cw) or a file class (Fw).  Any names added to class w
#      must be canonical names.
#
#      For example, by default, only the server cnode of an HP-UX cluster
#      runs the sendmail daemon, and mail from the clients is sent out
#      with headers indicating that it originated on the server.
#
#      However, you might want the server also to accept mail addressed
#      to users at the clients.  You could have nameserver MX records
#      directing mail for the clients to the server, and make the server
#      recognize the clients' host names as local.  If the clients' names
#      are ick, ack, and ock in the domain urk.com, the class w could be
#      augmented as follows:
#
#        Cw ick.urk.com ack.urk.com ock.urk.com
#
```

```
#  Message-ID Header Field
#
#       Associating a unique Message-ID header line with every message can
#       be useful for tracing messages through the mail system.  This is
#       not the default because it makes the syslog file grow 50% faster.
#       If the configuration file line:
#
#         # H?M?Message-Id: <$t.$i@$w>
#
#       is uncommented, sendmail will generate a Message-ID header line
#       for each message it routes that does not already have one.  The id
#       contains the date (GMT), in the form YYMMDDhhmm, the queue ID
#       associated with the message on the local host, and the local host
#       name.
#
#       In any case, if an incoming message already has a Message-ID
#       header line, sendmail will preserve this line.
#
#  Postmaster Copy                (option P):
#
#       If this option is set to a valid address, when an undeliverable
#       message is mailed back, a copy of the message header (but not the
#       message body) is delivered to the PostMasterCopy address.  It is
#       suggested that this address be "Postmaster" since RFC 822 requires
#       that there be a valid Postmaster address on all mail systems, and
#       since sendmail also attempts to deliver undeliverable error return
#       messages to Postmaster.
#
```

```
#       Postmaster should be aliased locally to the (local or remote)
#       address of the person responsible for mail system administration.
#
#       To enable this feature, uncomment the line:
#
#         # OPPostmaster
#
#       If desired, replace "Postmaster" with the appropriate address.
#
# Dumb UUCP Hosts                (class G):
#
#       If your host has UUCP connections to hosts running "dumb",
#       i.e. non-RFC 822 compatible, UUCP mailers, you should define the
#       class G as a list of these hosts.  This will cause UUCP mail to
#       these hosts to be sent via the dumbuucp mailer rather than the
#       normal uucp mailer.  Since "dumb" mailers will not add their host
#       names to the return path in the From:  header field, the dumbuucp
#       delivery agent deletes From:  header fields from mail for such
#       hosts.  In most cases the mail system on the host where final
#       delivery is done will be able to generate a correct return path
#       from other information in the message header.
#
#       For example, if your host has UUCP connections to dumb UUCP
#       mailers on hosts rrgh and feh, the class G should be defined,
#       on the line beginning with CG, as follows:
#
#         CGrrgh feh
#
```

```
# Trusted Users                    (T):
#
#       If other hosts login to your host to perform UUCP mail transfers,
#       the login ids used by these systems must be defined as trusted
#       users.  The login id "uucp" is already defined as a trusted user.
#       If necessary, add lines of the form:
#
#         Tusername
#
#       to the Trusted Users section of this file (lines beginning with T)
#
# Logging Level                    (option L):
#
#       Logging level determines the classes of events which will be
#       logged by sendmail in /usr/spool/mqueue/syslog.  By default the
#       log level is 10, which reports successful deliveries (and the
#       mailer and host used for delivery), queue daemon startup, alias
#       database rebuilds, and various errors.  More detailed information
#       is reported with higher log levels.  In particular, log level 11
#       reports the MX host (if any) and internet address to which mail
#       was delivered.  Refer to the documentation for details.
#
#       Note that log level also affects the information reported by
#       sendmail -bv.  At log level 10 and higher, sendmail also reports
#       the mailer and host that would be used for addresses that are
#       "deliverable."
#
#       The option L is defined on the line beginning with OL.
```

```
#
#  Delivery Mode                    (option d):
#
#       The default delivery mode is "background":  the user agent
#       invoking sendmail will return immediately, and sendmail will route
#       the mail in the background.  Other options are "interactive":
#       sendmail will not return control to the program invoking it until
#       the mail has been routed; and "queue":  sendmail will put the
#       message in the mail queue and a sendmail queue process will route
#       the message later.  Delivery mode is defined on the line beginning
#       with Od.
#
#  Error Reporting Mode             (option e):
#
#       The default error reporting mode is "p":  if sendmail detects an
#       error before it finishes sending a message, error messages are
#       output to stdout; if errors occur later they are mailed back.
#       Other options are:  "w":  if the sender is logged in, any error
#       messages are written to the sender's terminal, otherwise they are
#       mailed back; "m":  error messages are always mailed; and "q":  no
#       error messages.  Error reporting mode is defined on the line
#       beginning with Oe.
#
#  Read Timeout                     (option r):
#
#       If the program transmitting a message to sendmail hangs, or if an
#       SMTP peer goes down, sendmail's read will time out after this
#       interval.  RFC 1123 section 5.3.2 discusses appropriate values for
```

```
#       this timeout.  Refer to the sendmail documentation for the syntax
#       for specifying time intervals.  The read timeout is defined on the
#       line beginning with Or.
#
#  Queue Timeout                        (option T):
#
#       Messages in the mail queue which sendmail has been unable to
#       deliver for this amount of time will be returned to the sender as
#       undeliverable.  Refer to the documentation for the syntax for
#       specifying time intervals.  Queue timeout is defined on the line
#       beginning OT.
#
#  Queue-Only Load Average              (option x):
#
#       In order to limit load on a very busy system, sendmail can be
#       configured to queue up low priority messages rather than attempt
#       delivery immediately if the five-minute load average is greater
#       than some integer value, by default 8.  This value is defined on
#       the line beginning Ox.
#
#  Refuse-Connections Load Average  (option X):
#
#       In order to limit load on a very busy system, the sendmail daemon
#       can be configured not to accept SMTP connections if the five-
#       minute load average is greater than some integer value, by
#       default 12.  This value is defined on the line beginning Ox.
#
##########################################################################
```

```
##############################################################################
#  Routing Options:
##############################################################################
#
#       The supported routing options are described below.  Options for
#       routing SMTP mail, UUCP mail, X.400 mail, and OpenMail mail are
#       described.
#
#       To implement any of these options, edit a copy of this file
#       according to the instructions for that option.  With minor
#       exceptions, noted where appropriate, the options are independent
#       of each other; you can implement any of them or none of them.
#
#       Some of these options require that you have installed and
#       configured other software not part of the ARPA/9000 product.
#
#
#       Operator precedence:
#
#       It is generally agreed that mixed addresses, being ambiguous, are
#       abhorrent.  This configuration file interprets address operators
#       in the order '@', '!', '%'.
#
#       Changes to this precedence are not supported!
#
#       Mixed addresses are resolved as follows:
#
```

```
#       Address             Mailer   Host    User                Destinat
#       -------             ------   ----    ----                --------
#       user%hostA@hostB    tcp      hostB   user%hostA@hostB    user@host
#       hostA!user@hostB    tcp      hostB   hostA!user@hostB    hostA!use
#       hostA!user%hostB    uucp     hostA   user@hostB          user@host
#
##########################################################################
#  SMTP Mail Routing:
##########################################################################
#
#       By default, all mail to addresses of the form "user@host" will be
#       routed to "host" using the SMTP protocol over TCP/IP.  If the
#       nameserver is in use, an MX record may direct sendmail to route
#       mail for that host to some other host acting as a "mail exchanger"
#       for the host in the recipient address.  If there is no MX record
#       for the host name, gethostbyname(3n) must be able to return an
#       internet address for the host or else the delivery will fail.
#
#       You can arrange to mail directly only to certain hosts.  Mail to
#       other hosts can either be rejected as an error, or be passed to a
#       "relay" host for delivery.
#
#  Direct SMTP connection within local domain only:
#
#       This option makes sense only if your site uses internet domain
#       style host names.  You can define macro L as the name of the
#       "local domain" and mail directly only to hosts whose canonical
#       names end with that name.  For example, if your local domain is
```

```
#       "envy.sins.com", you might define macro L as "envy.sins.com" and 60
#       define an SMTP relay, in which case sendmail would send directly
#       to "user@porch.envy.sins.com" but relay "user@foyer.pride.sins.com
#       Alternatively, if your host is on a closed subnet, you might
#       define macro L as "sins.com" and only relay mail (through a host
#       with an interface on an open subnet as well as on your closed
#       subnet) to domains outside sins.com .
#
#       To route only to the local domain, first define macro L.  For
#       example, to route directly to hosts in the domain sins.com, define
#       macro L, on the line beginning DL, as follows:
#
#        DLsins.com
#
#       Then uncomment the line in ruleset 0 following the comment:
#
#        # connect to hosts in local domain
#
#       Finally, comment out the line in ruleset 0 following the comment:
#
#        # try to connect to any host for user@domain
#
#  Direct SMTP connection to hosts in class S only:
#
#       You can define a file class S of hosts to connect to directly.
#       This option does not make much sense if your host runs the
#       nameserver or NIS (formerly called Yellow Pages), since it
#       requires that you keep a static table up to date.  An internal
```

```
#       hash table is generated from the file when the configuration file 61
#       is frozen or, if no frozen configuration file is in use, when
#       sendmail starts up.  Therefore, if the file from which the class
#       is generated changes, the configuration file should be re-frozen
#       and the sendmail daemon killed and restarted.
#
#       The file from which this class is defined (by default
#       /etc/hosts.smtp) should contain only host names to which your host
#       can connect via TCP, as the first word on each line.  It would be
#       appropriate for the file to include host name aliases as well as
#       official host names, and to exclude hosts that do not run an SMTP
#       server.
#
#       The file can contain internet domain style names.  The current
#       HP-UX version of sendmail can match multi-token names to class
#       members.
#
#       To define file class S, first uncomment the line defining class S
#       from /etc/hosts.smtp, following the comment:
#
#        # class S defines hosts to which you connect directly for SMTP ma
#
#       Then uncomment the line in ruleset 0 following the comment:
#
#        # connect to hosts in class S
#
#       Finally, comment out the line in ruleset 0 following the comment:
#
```

```
#       # try to connect to any host for user@domain
#
# Direct SMTP connection to nobody:
#
#       This option makes sense only if an SMTP relay is also defined.  A
#       host (for example a single user workstation) may wish to send all
#       SMTP mail through a relay.  To arrange this, comment out the line
#       in ruleset 0 following the comment:
#
#       # try to connect to any host for user@domain
#
#       Note that this option is not compatible with options 1 or 2 above.
#
# SMTP relay:
#
#       If you are limiting direct SMTP connections as described above,
#       by default mail to hosts outside the limits will be returned with
#       an error.  If you can connect to another host with wider network
#       connectivity than your host has, you can use that host as an SMTP
#       relay.  Courtesy suggests that you get permission from the
#       administrator of that host before relaying mail through it.
#
#       Any host name defined as a relay to be reached via SMTP must be an
#       official host name; in a domain naming environment this must be a
#       fully qualified domain name.  The canonicalization operator $[ $]
#       is not applied to relay names.
#
#       To set up an SMTP relay, define the macro S, on the line beginning
```

```
#       DS, as the host name (not path) of the relay host.  For example,  63
#       if you wish to relay through the host blab.bub.edu, define S as
#       follows:
#
#        DSblab.bub.edu
#
#       Then uncomment the line in Ruleset 0 following the comment:
#
#        # pass unresolved SMTP addresses to the SMTP relay
#                                          (don't relay source routes)
#
#       Note that if you are not limiting direct SMTP connections in some
#       way, SMTP mail cannot be relayed.
#
######################################################################
#  UUCP Mail Routing:
######################################################################
#
#       By default, mail to addresses of the form "host!user" is routed to
#       the uucp (or dumbuucp) mailer only if "host" is in the class U,
#       generated from the output of uuname(1).  Addresses of this form
#       referring to other host names are treated as errors.
#
#       The following UUCP routing options are supported:
#
#  pathalias external nameserver for UUCP:
#
#       You can maintain a pathalias database to provide sendmail with
```

```
#       information on how to route UUCP mail to hosts to which your host
#       does not have a direct UUCP connection.  This routing information
#       is generated by pathalias(1m).  Mkuupath(1m) creates a hashed
#       database from the routing information, and uupath(1m) is used to
#       access the database.
#
#       To make sendmail use uupath, first set up the database as
#       described above.  Then uncomment the three lines in Ruleset 0
#       following the comment:
#
#       # try to get a path to an unresolved UUCP address
#                                       from pathalias nameserver
#
#  UUCP relay (via UUCP):
#
#       If your host has a UUCP path to another host with wider UUCP
#       connectivity than your host has, it may be appropriate to route
#       UUCP mail to hosts to which your host cannot connect through
#       this other host, which will attempt to relay messages to their
#       final destination.  Courtesy suggests that you get permission from
#       the administrator of that host before relaying mail through it.
#
#       To implement UUCP relaying via UUCP, first define the macro U, on
#       the line beginning DU, as either the relay host name or the path
#       to the relay host.
#
#       For example, if you wish to relay through the host pzzz, via the
#       path rrghlfrrlfehlpzzz , define U:
```

```
#
#       DUrrgh!frr!feh!pzzz
#
#       If you have a direct UUCP connection to pzzz, define U:
#
#       DUpzzz
#
#       After defining the UUCP relay, uncomment the three lines in
#       Ruleset 0 following the comment:
#
#       # pass unresolved UUCP addresses to the UUCP relay (via UUCP)
#
# UUCP relay (via SMTP):
#
#       If you can mail via SMTP to the host you wish to use as a UUCP
#       relay, define the macro W, on the line beginning DW, as the name
#       of the UUCP relay host.
#
#       Note that in this case, the macro W must be a single host name,
#       not a path.  Note also that the name of any host defined as a
#       relay to be reached via SMTP must be an official host name; in a
#       domain naming environment this must be a fully qualified domain
#       name.  The canonicalization operator $[ $] is not applied to relay
#       names.
#
#       For example, if domain names are not in use at your site, and
#       if the relay host's official name is pzzz, define W:
#
```

```
#       DWpzzz
#
#       Then uncomment the line in Ruleset 0 following the comment:
#
#        # pass unresolved UUCP addresses to the UUCP relay (via SMTP)
#
#       Note that there is some risk of generating ambiguous (mixed)
#       addresses in message headers using this type of relaying, if you
#       are passing mail to hosts whose mail configuration is not under
#       your control.  This may cause some messages to be unreplyable.
#
#  UUCP to X.400 gateway
#
#       By default, mail from an X.400 user or gateway via UUCP may not be
#       replyable.  This is due to uucp interpreting the '/' in the name a
#       implying a file transfer.
#
#       If you are using X.400 or OpenMail on this machine, and have UUCP
#       links from this machine, then apply the following changes:
#
#       Uncomment the line in Ruleset 6 following the comment:
#
#       # Recognize mail from uucp for x400 user on this system
#
#       Uncomment the line in Ruleset 13 following the comment:
#
#       # enable uucp recipient to reply to remote x400 sender
#
```

```
#       and uncomment the line in Ruleset 13 following the comment:      67
#
#       # enable uucp recipient to reply to local x400 sender
#
#       This works by adding an extra host name 'hpx400' to the route back
#       to this system; this name is stripped by the line in Ruleset 6.
#
#       Note that this will mean that a machine called 'hpx400' will not b
#       reachable from this machine via a direct UUCP connection.
#
#
##########################################################################
#  X.400 Mail Routing:
##########################################################################
#
#       By default, mail to X.400 style addresses will be rejected as an
#       error.  The following options permit routing X.400 mail to the
#       X.400/9000 product running on this host or a remote host.
#
#  X.400 mail via X.400/9000 delivery agent on this host:
#
#       If you have installed the X.400/9000 X.400 delivery agent
#       (/usr/lib/x400/x4mailer) on this host, you must enable sendmail to
#       route X.400 mail to this mailer.  Uncomment the line in Ruleset 0
#       following the comment:
#
#        # resolve X.400 mail: local host is X.400 gateway
#
```

```
#       For example, if the name of the local host is "buh", and sendmail 68
#       is configured as described above, it would route mail to the
#       address User_Joe//SomeOrg/US/TELEMAIL////HP@buh to the X.400
#       delivery agent for further routing through the X.400 network.
#
#       See the X.400/9000 documentation for a complete explanation of
#       X.400 style addresses.
#
#       The X.400 receiving agent, also implemented by /usr/lib/x400/x4mai
#       will be able to hand incoming messages to sendmail for further
#       routing without any changes to the sendmail configuration.
#
#  X.400 Relay (X.400/9000 delivery agent on a remote host):
#
#       If the X.400/9000 X.400 delivery agent runs on a different host,
#       for example, "farfel", sendmail on farfel must be configured to
#       route X.400 mail to the X.400 delivery agent as described above.
#       Sendmail on the local host would route mail addressed, for
#       example, to User_Joe//SomeOrg/US/TELEMAIL////HP@farfel to farfel
#       via SMTP, and sendmail on farfel would hand the mail to the X.400
#       delivery agent for routing through the X.400 network.  This would
#       require no change in the local host's sendmail configuration.
#
#       So that users need not know which remote host is the X.400
#       gateway, sendmail can be configured to route mail to X.400
#       addresses via a designated X.400 gateway automatically.  Define
#       the macro X, on the line beginning DX, as the name of (not a path
#       to) the X.400 gateway host.  The X.400 relay must be accessible to
```

```
#       this host via SMTP and must be defined as an official host name.
#
#       In a domain naming environment this must be a fully qualified
#       domain name.  The canonicalization operator $[ $] is not applied
#       to relay names.
#
#       For example, if official name of the X.400 gateway is farfel,
#       define the macro X as follows:
#
#        DXfarfel
#
#       Then uncomment the line following the comment:
#
#        # resolve X.400 mail: remote host is X.400 gateway
#
#       If the local host is named "buh", mail addressed to:
#
#        User_Joe//SomeOrg/US/TELEMAIL////HP@buh
#
#       will automatically be relayed to the X.400 delivery agent on
#       farfel.
#
#       See the X.400/9000 documentation for a complete explanation of
#       X.400 style addresses.
#
##########################################################################
#  OpenMail Routing:
##########################################################################
```

```
#
#       By default, mail to OpenMail style addresses will be rejected as a
#       error.  The following options permit routing OpenMail and X.400 ma
#       to the OpenMail product running on this host or a remote host.
#
#  OpenMail to OpenMail delivery agent:
#
#       If you have installed the OpenMail internal delivery agent
#       /usr/openmail/bin/xport.in on this host, you must enable sendmail
#       to route mail from other OpenMail systems to local OpenMail users
#       via this mailer.  Uncomment the line in Ruleset 0 following the
#       comment:
#
#         # resolve mail to OpenMail from remote OpenMail system
#
#  OpenMail to X.400 delivery agent:
#
#       If you have installed the OpenMail X.400 interface option
#       (/usr/openmail/bin/x400.out) on this host, you must enable
#       sendmail to route mail from OpenMail users to the X.400 network
#       via this mailer.  Uncomment the line in Ruleset 0 following the
#       comment:
#
#         # resolve mail to X.400 from OpenMail
#
#  OpenMail delivery agent on this host:
#
#       If you have installed the OpenMail delivery agent,
```

```
#       /usr/openmail/bin/unix.in, on this host, you must enable sendmail 71
#       to route mail to OpenMail users via this mailer.  Uncomment the
#       line in Ruleset 0 following the comment:
#
#         # resolve mail to OpenMail: local host is OpenMail gateway
#
#       For example, if the name of the local host is "warsh", and
#       sendmail is configured as described above, it would route mail to
#       the address User_Joe/SomeOrgUnit@warsh to the OpenMail delivery
#       agent for routing through the OpenMail network.
#
#       See the OpenMail documentation for a complete explanation of
#       OpenMail style addresses.
#
#       If the sendmail configuration has not been changed to pass X.400
#       messages to X.400/9000, then the above change will also cause X.40
#       messages to be passed to OpenMail.  For example, the address
#       User_Joe//SomeOrg/US/TELEMAIL////HP@warsh would be passed to the
#       OpenMail delivery agent for further routing through the X.400
#       network.
#
#       The OpenMail receiving agent will be able to hand incoming message
#       to sendmail for further routing without any changes to the
#       sendmail configuration.
#
#  OpenMail Relay (OpenMail delivery agent on a remote host):
#
#       If the OpenMail delivery agent runs on a different host, for
```

```
#       example, "tortue.ordinary.com", sendmail on tortue must be
#       configured to route OpenMail mail to the OpenMail delivery agent
#       as described above.  Sendmail on the local host would route mail
#       addressed, for example, to User_Joe/SomeOrgUnit@tortue to tortue
#       via SMTP, and sendmail on tortue would hand the mail to the
#       OpenMail delivery agent for further routing through the OpenMail
#       or X.400 networks.  This would require no change to the local
#       host's sendmail configuration.
#
#       So that users need not know which remote host is the OpenMail
#       gateway, sendmail can be configured to route mail to OpenMail
#       addresses via a designated OpenMail gateway automatically.  Define
#       the macro Z, on the line beginning DZ, as the name (not path) of
#       the OpenMail gateway host.  The OpenMail relay must be accessible
#       to this host via SMTP.
#
#       In a domain naming environment this must be a fully qualified
#       domain name.  The canonicalization operator $[ $] is not applied
#       to relay names.
#
#       For example, if the official host name of the OpenMail gateway is
#       tortue.ordinary.com, define the macro Z as follows:
#
#         DZtortue.ordinary.com
#
#       Then uncomment the line following the comment:
#
#         # resolve mail to OpenMail: remote host is OpenMail gateway
```

```
#
#       If the local host is named "warsh", mail addressed to:
#
#         User_Joe//SomeOrg/US/TELEMAIL////HP@warsh
#
#       will automatically be relayed to the OpenMail delivery agent on
#       tortue.ordinary.com .
#
#       See the OpenMail documentation for a complete explanation of
#       OpenMail style addresses.
#
######################################################################

######################################################################
###                  Localizable Options                           ###
######################################################################

# logging level
OL10

# defer messages to [IPC] mailers if the nameserver is not running
# OI

# delivery mode
Odbackground

# error reporting mode
Oep
```

```
# read timeout
Or5m

# queue timeout interval
OT3d

# load average at which low priority messages are queued rather than deli
Ox8

# load average at which daemon refuses to accept connections
OX12

# postmaster address which will receive headers of undeliverable messages
# OPPostmaster

######################################################################
###        Other Options ###
######################################################################
### ###
###   HP recommends that these options not be changed. ###
### ###
######################################################################

# queue directory
OQ/usr/spool/mqueue

# Save those UN*X From_ lines
```

```
Of

# location of alias file
OA/usr/lib/aliases

# temporary file mode
OF0600

# default UID
Ou1

# default GID
Og1

# location of help file
OH/usr/lib/sendmail.hf

# recognize old style as well as new style lists in headers
Oo

# statistics file
OS/usr/lib/sendmail.st

# wait up to 5 minutes for completion of alias db initialization
Oa5

# queue up everything before starting transmission
Os
```

```
# send to me, too if in alias expansion
Om

# if the load average exceeds the x option limit, divide the q option
# value by the difference (plus one) between the current load average and
# the x option limit to find the maximum priority value (i.e. minimum
# priority) of messages to send immediately.
Oq10000

# value added to message priority per recipient
Oy1000

# message precedence factor
Oz1800

# value added to message priority per queue run
OZ9000


#####################################################################
###  Configuration-Specific Macro and Class Definitions ###
#####################################################################

# site hiding: local sender identified as user@my_site instead of user@my
DYlcad00.kmitl.ac.th

# class w defines aliases for the local host
```

```
Cwlcad00.kmitl.ac.th kmitl.ac.th lcad00

# macro L defines the "local domain" to which you connect directly for SM
DLkmitl.ac.th

# class S defines hosts to which you connect directly for SMTP mail
FS/etc/hosts.smtp %s

# class U defines known direct UUCP connections
FU!/usr/bin/uuname %s

# UUCP relay for unresolved ! addresses (via UUCP)
DU

# UUCP relay for unresolved ! addresses (via SMTP)
DW

# SMTP relay for unresolved @ addresses
DSnwg.nectec.or.th

# X.400 relay if X.400 delivery agent is not local
DX

# OpenMail relay if OpenMail delivery agent is not local
DZ

# dumb (not RFC 822 compatible) UUCP hosts
CG
```

```
# pathalias external nameserver program
DP/usr/bin/uupath


######################################################################
###                  Configuration Version                         ###
######################################################################

DV1.6.2

######################################################################
###                  Required Macro Definitions                    ###
######################################################################

# official domain name of this host for SMTP
Dj$w

# my name
DnMAILER-DAEMON

# UNIX header format
DlFrom $g $d

# delimiter (operator) characters
Do.:%@!^=/[]|

# format of a total name
```

```
Dq$?x$x <$g>$¦$g$.

# SMTP banner
De$j HP Sendmail ($v/$V) ready at $b


######################################################################
###                    Message Precedences                         ###
######################################################################

Pfirst-class=0
Pspecial-delivery=100
Pjunk=-100


######################################################################
###                       Trusted Users                            ###
######################################################################

Troot
Tdaemon
Tuucp
Tx400


######################################################################
###                    Header Field Formats                        ###
######################################################################
```

```
HReceived: $?sfrom $s $.by $w$?r with $r$.
($v/$V) id $i; $b
HResent-Date: $a
HDate: $a
HResent-From: $q
H?F?From: $q
H?x?Full-Name: $x
H?P?Return-Path: <$g>
HSubject:
# HPosted-Date: $a
# HReceived-Date: $b
# HResent-Message-Id: <$t.$i@$w>
# H?M?Message-Id: <$t.$i@$w>


######################################################################
######################################################################
#####                                                          #####
#####          Address Rewriting Rulesets                      #####
#####                                                          #####
######################################################################
######################################################################


######################################################################
###  Ruleset 1 - Sender Field Pre-rewriting                      ###
######################################################################
```

```
S1
R$+$:$>6$1strip my_host and canonicalize
R$*<@$+>$*$@$1<@$2>$3already has (remote) domain
R$+/$*/$*/$*/$*$:$1/$2/$3/$4/$5<@$w>@my_domain on local X.400 sender


##################################################################
###  Ruleset 2 - Recipient Field Pre-rewriting                 ###
##################################################################

S2
R$+$:$>6$1strip my_host and canonicalize
R$*<@$+>$*$@$1<@$2>$3already has (remote) domain
R$+/$*/$*/$*/$*$:$1/$2/$3/$4/$5<@$w>@my_domain on local X.400 recpt


##################################################################
###  Ruleset 3 - Address Internalization                       ###
##################################################################

S3
# handle "From:<>" special case
R<>$@$nnull address => MAILER-DAEMON

# basic textual canonicalization
R$*<$*<$+>$*>$*$1<$2$3$4>$5strip <> from inside
R$*<$+>$*$2strip phrase and <>
```

```
R$+ at $+$1@$2RFC 733 at => RFC 822 @

# source route <@a,@b,@c:user@d> syntax to internal form <@a>:@b:@c:user@
R@$+,$+@$1:$2change all , to :
R@$+.UUCP:$+$@<@$1.UUX>:$2.UUCP pseudo-domain in route
R@$+:$+$@<@$1>:$2focus on next hop

# The @ delimiter takes precedence.  Leave this alone.
R$+@$+$:$1<@$2>focus on domain
R$+<$+@$+>$1$2<@$3>move gaze right
R$+<@$+.UUCP>$@$1<@$2.UUX>.UUCP pseudo-domain
R$+<@$+>$@$1<@$2>already in internal form

# The ! delimiter.
R$+^$+$1!$2convert obsolete ^ to !
R$+!$+$@$2<@$1.UUX>host!user => user<@host.UUX>

# % is a low precedence @.
R$+%$+$:$1<%$2>focus on domain
R$+<$+%$+>$1$2<%$3>move gaze right
R$+<%$+>$1<@$2>user%host => user@host
R$+<@$+.UUCP>$@$1<@$2.UUX>.UUCP pseudo-domain

# miscellaneous cleanup
R$+@$@$1user@ => user
R$+%$@$1user% => user
```

```
##########################################################
###  Ruleset 4 - Final Output Post-rewriting           ###
##########################################################

S4
# special cases
R$+<@>$1null domain

# UUCP must always be presented as host!user
R$+<@$+.UUX>$@$2!$1user<@host.UUX> => host!user

# UUCP hop in source route
R<@$+.UUX>:$+<@$1.UUCP>:$2.UUX in source route => .UUCP
R<@$+>:$+.UUX$+<@$1>:$2.UUCP$3.UUX in source route => .UUCP

# defocus
R$*<$+>$*$1$2$3remove internal form <>

# don't change %s or @s in mixed addresses
R$+!$+@$+$@$1!$2@$3don't interpret it any further
R$+!$+%$+$@$1!$2%$3don't interpret it any further

# restore source route to external form
R@$+:$+:$+@$1,$2:$3all but last : => ,
R@$+$@<@$1>add <> to protect the ,s

# should be exactly one @ in user@domain style address
R$+%$+$1@$2all % => @
```

```
R$+@$+@$+$1%$2@$3all but last @ => %


##################################################################
###  Ruleset 0 - {Delivery_Agent, Host, User} Resolution  ###
##################################################################

S0
# recognize local host or canonicalize
R$+$:$>6$1anything to ruleset 6 once


# resolve domain-literals (numeric internet addresses) not canonicalized
R$*<@[$+]>$#tcp$@[$2]$:$1@[$2]user@internet address
R<@[$+]>:$*$#tcp$@[$1]$:@[$1]:$2internet address in source route

# resolve mail to dumb UUCP hosts
R$+<@$=G.UUX>$#dumbuucp$@$2$:$1user@dumb_host.UUX

# resolve mail to other known UUCP hosts
R$+<@$=U.UUX>$#uucp$@$2$:$1user@host.UUX
R<@$=U.UUX>:$+$#uucp$@$1$:$2@host.UUX in source route

# try to get a path to an unresolved UUCP address from pathalias nameserv
# R$+<@$-.UUX>$:$>5$2!$1uupath pathalias routing
# R$+<@$=G.UUX>$#dumbuucp$@$2$:$1to dumb UUCP host
# R$+<@$=U.UUX>$#uucp$@$2$:$1to other known UUCP host

# pass unresolved UUCP addresses to the UUCP relay (via SMTP)
```

```
# R$+<@$+.UUX>$#tcp$@$W$:$1<@$2.UUX>to UUCP relay via SMTP

# pass unresolved UUCP addresses to the UUCP relay (via UUCP)
# R$+<@$+.UUX>$:$>3 $U!$2!$1re-internalize with $U
# R$+<@$=G.UUX>$#dumbuucp$@$2$:$1to dumb UUCP relay
# R$+<@$+.UUX>$#uucp$@$2$:$1to UUCP relay

# other UUCP addresses are in error
R$*<@$+.UUX>$*$#error$:unable to route to UUCP host name $2
```

```
# select hosts to connect with directly for SMTP mail:

# connect to hosts in class S
R$+<@$=S>$#tcp$@$2$:$1<@$2>user@domain

# connect to hosts in local domain
# R$+<@$+.$L>$#tcp$@$2.$L$:$1<@$2.$L>user@host.localdomain

# try to connect to any host for user@domain
# R$+<@$+>$#tcp$@$2$:$1<@$2>user@domain

# try to connect to any host for source route
R<@$+>:$+$#tcp$@$1$:<@$1>:$2source route

# pass unresolved SMTP addresses to the SMTP relay (don't relay source ro
R$+<@$+>$#tcp$@$S$:$1<@$2>user@domain to SMTP relay

# other SMTP addresses are in error
```

```
R$*<@$+>$*$#error$:unable to route to domain $2


# file names, programs, and :include: must resolve to local mailer;
# explicitly distinguish these from X.400 and OpenMail syntax
R/$*$#local$:/$1to absolute file path name
R|$*$#local$:|$1to a program
R:include:$*$#local$::include:$1to :include: list


# resolve X.400 mail: local host is X.400 gateway
# R$+/$*/$*/$*/$*$#x400$@$w$:$1/$2/$3/$4/$5


# resolve X.400 mail: remote host is X.400 gateway
# R$+/$*/$*/$*/$*$#tcp$@$X$:$1/$2/$3/$4/$5<@$X>


# resolve mail to OpenMail: local host is OpenMail gateway
# R$+/$*$#openmail$@$w$:$1/$2


# resolve mail to OpenMail: remote host is OpenMail gateway
# R$+/$*$#tcp$@$Z$:$1/$2<@$Z>


# by default, reject X.400 address as error
R$+/$*/$*/$*/$*$#error$:X\.400 delivery agent not configured


# by default, reject OpenMail address as error
R$+/$*$#error$:OpenMail delivery agent not configured


# resolve mail to OpenMail from remote OpenMail system
# Ropenmail$#omxport$@$w$:openmail
```

```
# resolve mail to X.400 from OpenMail
# Ropenmailx4$#omx400$@$w$:openmailx4

# other addresses must resolve to local mailer in order for mail to Full.
# command line aliases, and quoted user names (\user) to be delivered.

# remaining names must be local
R$+$#local$:$1name
```

```
##########################################################################
##########################################################################
#####                                                              #####
#####                Special Rulesets                              #####
#####                                                              #####
##########################################################################
##########################################################################


##########################################################################
###  Ruleset 5 -    Pathalias Nameserver                            ###
##########################################################################

S5
R$+$:$<P$1uupath pathalias routing
R$+$:$>3$1re-internalize
```

```
##########################################################
###  Ruleset 6 -   Local Host Recognition              ###
##########################################################

S6
# RFC 822 does not permit hostnames to end in .
R$*<@$*.>$*$1<@$2>$3strip trailing .

# strip local host
R$+<@$w>$>3$1strip my_host and re-internalize
R<@$w>:$+$>3$1strip my_host and re-internalize

# recognize local host in UUCP syntax
R$+<@$k.UUX>$:$1<@$w>my_host in UUCP syntax
# Recognize mail from uucp for x400 user on this system
# R$+<@hpx400.UUX>$:$1<@$w>X.400 in UUCP syntax
R<@$k.UUX>:$+$:<@$w>:$1my_host.UUCP in source route

R$*<@$+.UUX>$*$@$1<@$2.UUX>$3don't canonicalize host.UUX

# canonicalize host and possibly recurse
R$*<@$+>$*$:$1<@$[$2$]>$3user@host or source route
R$*<@$=w>$*$:$>6$1<@$w>$3recurse if still to my_host

##########################################################
##########################################################
```

```
#####                                                          #####
#####        Mailer (Delivery Agent) Definitions               #####
#####                                                          #####
#########################################################################
#########################################################################

#########################################################################
###                local and program mailers                          ###
#########################################################################

Mlocal, P=/bin/rmail,F=DFMPlms,  S=10,  R=20,  A=rmail -d $u
Mprog,P=/bin/sh;  F=DFMPlshu, S=10,R=20,, A=sh -c $u

S10
R$*<@$+.UUX>$@$1<@$2.UUX>don't modify UUCP address
R$*<@$+>$*$@$1<@$2>$3already has domain
R$+$:$1<@$w>add local domain to user

S20
R$*<@$+.UUX>$@$1<@$2.UUX>don't modify UUCP address
R$*<@$+>$*$@$1<@$2>$3already has domain
R$+$:$1<@$w>add local domain to user

#########################################################################
###                  SMTP TCP/IP mailer                               ###
#########################################################################
```

```
Mtcp,  P=[IPC], F=CDFMXmu,  S=11,  R=21,  E=\r\n,  A=IPC $h


S11
R$*<@$+.UUX>$@$2!$1<@$w>add local domain to UUCP address
R<@$+>:$*$@<@$w>:@$1:$2add local domain to source route
R$+<@$+>$@$1<@$2>already has domain
R$+$:$1<@$?Y$Y$|$w$.>add local domain

S21
R$*<@$+>$*$@$1<@$2>$3already has domain
R$+$:$1<@$w>add local domain


#####################################################
###                UUCP mailer                    ###
#####################################################

Muucp,P=/usr/bin/uux, F=DFMUshu, S=13,  R=23,  A=uux - $h!rmail ($u)


S13
R$+<@$+.UUX>$@$2!$1<@$k.UUX>host!user => my_host!host!user
R$+@$+<@$+>$1%$2<@$3>all but last @ => %
# enable uucp recipient to reply to remote x400 sender
# R$*/$*<@$+>$@hpx400!$1/$2%$3<@$k.UUX>user@host => my_host!hpx400!user%h
R$+<@$+>$@$1%$2<@$k.UUX>user@host => my_host!user%host
R<@$+>:$+$@<@$k.UUCP>:@$1:$2prepend @my_host.UUCP to route
# enable uucp recipient to reply to local x400 sender
```

```
# R$*/$*$@hpx400!$1/$2<@$k.UUX>user => my_host!hpx400!user

R$+$@$1<@$k.UUX>user => my_host!user


S23


##################################################################
###                  Dumb UUCP mailer                          ###
##################################################################
###   UUCP for hosts running non-RFC 822 mailers               ###
##################################################################

Mdumbuucp,  P=/usr/bin/uux,  F=DMUshux, R=23,A=uux - $h!rmail ($u)


##################################################################
###                  X.400 mailer                              ###
##################################################################

Mx400, P=/usr/lib/x400/x4mailer, F=CDMFmn, S=14, R=24, A=x4mailer -f $g $


S14


S24


##################################################################
```

```
#########################################################

Mopenmail,  P=/usr/openmail/bin/unix.in,  F=DFLMXmnu,  S=15,  R=25,  A=un

S15

S25

Momxport,  P=/usr/openmail/bin/xport.in,  F=LMn, A=xport.in $u

Momx400,  P=/usr/openmail/bin/x400.out,  F=LMn, A=x400.out $u
```

บทที่ 8

Serial Line IP (SLIP)

SLIP เป็นการเชื่อมต่อระบบ TCP/IP เข้าด้วยกันโดยผ่าน serial link ซึ่งมักใช้กับการเชื่อมต่อในระยะไกล ซึ่งถ้าใช้สาย coax จะไม่สะดวกนัก

โดยในการศึกษาครั้งนี้ ได้ใช้โปรแกรม ppl ซึ่ง support protocol ต่างๆ คือ Serial Line Internet Protocol (SLIP), Abbreviated Serial Line Internet Protocol Client (ASLIPC), Abbrevitated Serial Line Internet Protocol SErver (ASLIPS) และ Point-to-Point (PPP)

ในการ setup ระบบ SLIP ด้วยโปรแกรม ppl นั้น จะต้องทำการ setup configuration ต่างๆ ใน file

- ppl.users
- ppl.ipool
- ppl.remotes

โดยในการทดลองนี้ได้ทำการ setup config ต่างๆดังนี้

File ppl.users :

```
#this is the ppl.users file
# format: <username> <inetaddrss or rhostname>
root cs5000
```

File ppl.remotes :

```
#  filename: ppl.remotes
#
#  file format rules:
#
# A #  or a blank in column 1 is a comment line.
# For other lines, everything after '#' is a comment
# (do NOT delete any of these lines or modify any
# text after the "#").

#
#  template for future addtions
#
cs5000          # remote host name or Internet address
slip817         # local host name or Internet address
                # Internet mask
SLIP            # protocol [SLIP] [ASLIPC] [ASLIPS] [PPP]
DIRECT          # type [DIRECT] [DIALIN] [DIALOUT] [DIALIN & DIALOUT]
                # UUCP system name
NONE            # line parity [EVEN] [ODD] [NONE]
9600            # line speed
/dev/tty0p4     # serial line
                # phone number
                # modem control available [YES] [NO]
                # log in info
                # command name
```

จาก configuration ใน file ppl.remotes จะเห็นได้ว่าเป็นการติดตั้ง SLIP ให้วิ่งระหว่าง เครื่อง cs5000 กับเครื่อง slip817 (ส่วย ip address จริงนั้น จะไป map ที่ file /etc/hosts) เป็นการเชื่อมต่อแบบ direct connect เข้ากันด้วย line ความเร็ว 9600 bps โดยผ่าน device tty0p4

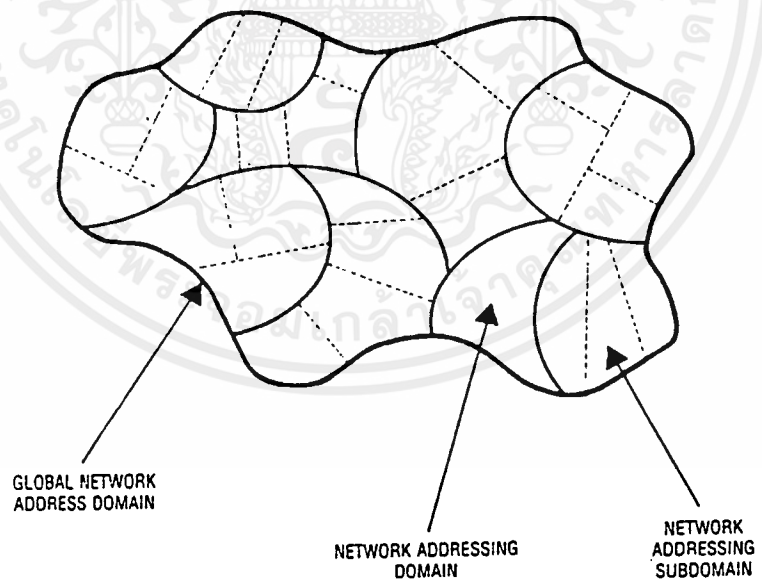
บทที่ 9

มาตราฐาน X.25 และ มาตราฐานอื่นๆที่เกี่ยวข้อง

มาตราฐาน X.121

X.121 คือมาตราฐานของ CCITT ที่จะกำหนด address ให้แก่ DTE ทุกๆ DTE ที่มาเชื่อมต่อเข้ากับ Public Data Network ทุกๆที่ทั่วโลก เราอาจเปรียบเทียบระบบ Network Address นี้ได้กับระบบการกำหนดหมายเลขโทรศัพท์ กล่าวคือ จะอนุญาตให้ destination ซึ่งมักเป็น host computer รับการเรียกจากทุกๆ คนที่รู้เลข address ของมัน

X.121 นั้นจะอนุญาตให้มี address ได้มากที่สุด 14 หลัก โดยแบ่งจาก global ลงมาตามลำดับ ลงเป็นระดับ zone หรือ domain ลงไป ในระดับของ domain นั้นจะถูกกำหนดโดย CCITT ส่วนอื่นๆจะถูกจัดสรรโดยแต่ละประเทศเข้าไปจัดสรรกันเอง ถ้ามี PDN มากกว่า 1 ในประเทศนั้นๆก็จะต้องมรการจัดเป็น subdomain ดังแสดงในรูปที่ 9-1



รูปที่ 9-1 Domain Addressing

X.121 network address สามารถแบ่งออกได้เป็น 2 ส่วน โดย 4 หลักแรกจะบ่งบอกถึง Data Network Identification Code (DNIC) ซึ่งจะถูกกำหนดโดย CCITT และยังบอกถึงแต่ละ Public Packet-Switched data network ในหลักแรกของ DNIC จะระบุ zone ต่างๆ ของโลก ดังนี้

1st DNIC digit	zone
0	Reserved
1	Reserved
2	Europe
3	North America
4	Asia
5	Oceania and Southest Asia
6	Africa
7	South America
8	Telex/TWX networks
9	Telephone networks

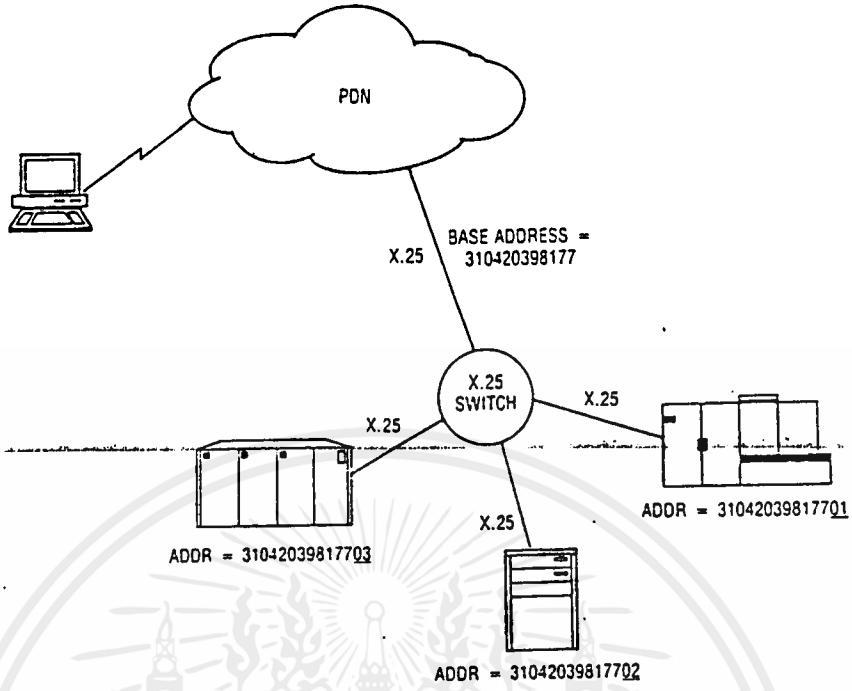
เมื่อใช้ 3 หลักแรกของ DNIC ซึ่งรวมเรียกเป็น Data Country Code (DCC) จะใช้กำหนดแต่ละประเทศ เช่น

DCC	Country
208	France
234	United Kingdom and Northern Ireland
250	Soviet Union
302	Canada
311	United State
425	Israel
440	Japan
450	Korea
454	Hong Kong
502	Malaysia
505	Australia
602	Egypt
604	Morocco
724	Brazil
730	Chile

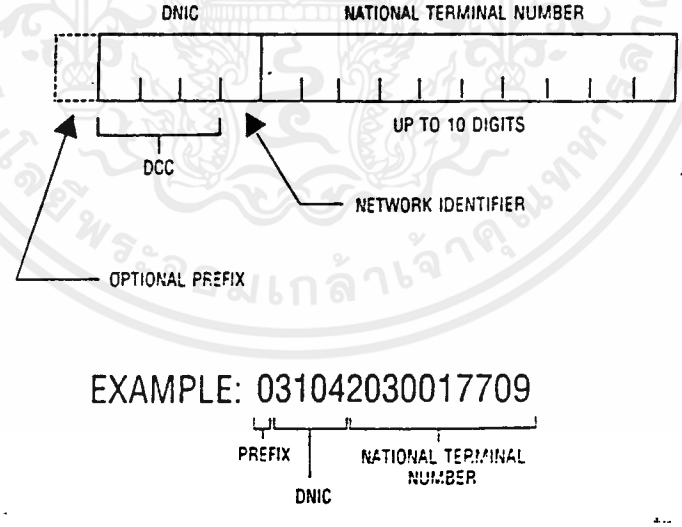
หลักที่ 4 ของ DNIC จะกำหนดสำหรับแต่ละ Public Packet Switched Network โดยใช้เพียงหลักเดียวเราก็สามารถใช้ได้เพียงพอกับจำนวน PDN ในแต่ละประเทศ คือไม่เกิน 10

สำหรับอีก 10 หลักที่เหลือของ X.121 address จะถูกกำหนด

โดยแต่ละ PDN



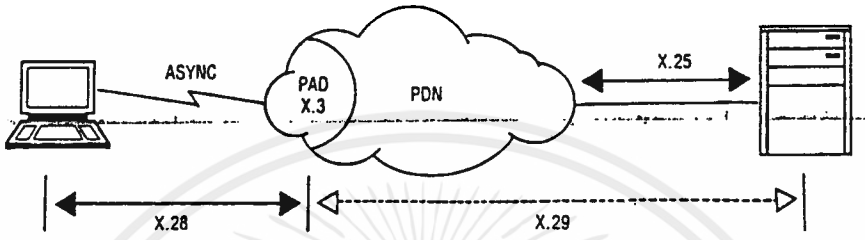
รูปที่ 9-2 การใช้ subdomain



รูปที่ 9-3 การกำหนด X.121 address

มาตราฐานของ Pad, X.3, X.28 และ X.29

ความสัมพันธ์ของ X.3, X.28 และ X.29 มีดังรูปที่ 9-4 คือ



รูปที่ 9-4 ความสัมพันธ์ของ X.3 X.28 และ X.29

X.3 จะอธิบายถึง configuration parameter ของผู้ใช้แต่ละคน ซึ่งอยู่ใน PAD แต่ละ asynchronous port ที่อาจอยู่บน PAD เดียวกัน เช่น PAD ที่มี 10 port ก็จะมีX.3 Parameter ได้ทั้งหมด 16 แบบ

คำสั่งหรือ procedure ที่ใช้ระหว่างอุปกรณ์ที่ต่ออยู่กับ X.3 จะเรียกว่า X.28 ซึ่งแบ่งออกได้เป็น 2 ส่วน คือ PAD command และ PAD response โดย PAD command คือ configuration message ที่ส่งไปยัง PAD หรืออาจเป็นคำสั่ง Call Setup/Clearing ที่ส่งมาจาก terminal และ response ที่ส่งกลับมาในทางตรงกันข้าม จะเรียกว่า PAD service signal

สำหรับ X.29 นั้นใช้ในการ control และ modify X.3 PAD

port Parameter บน network โดย remote packet-mode DTE เช่น X.25 101
host computer

X.3 Parameter

ตาม X.3 recommendation นั้น จะมีทั้งหมด 22 Parameter ดังแสดง ในรูปที่ 9-5

X.3 PAD PARAMETERS		
CATEGORY	NUMBER	FUNCTION
BREAK HANDLING	7 8	Action on Break Discard Output
DATA FORWARDING	3 4	Data Forwarding Character(s) Idle Timer
EDITING	15 16 17 18 19	Editing Character-Delete Character Buffer-Delete Character Buffer-Display Character Editing-Service Signals
FLOW CONTROL	5 12	Flow Control by PAD Flow Control by Terminal
LINE CHARACTERISTICS	11 21	Speed Parity Treatment
PAD RECALL	1	Escape From Data Transfer
DISPLAY FORMATTING	9 10 13 14 22	Carriage-Return Padding Line Folding Line-Feed Insertion Padding After Line Feed Page Wait
TERMINAL DISPLAY	2 6 20	Echo Control of PAD-Service Signals Echo Mask

รูปที่ 9-5 X.3 PAD Parameter

Parameter 1 Escape from Data Transfer

Parameter 1 เป็นตัวกำหนดว่าในระหว่างการทำงานอยู่นั้นจะยอมให้มีการสลับmode การทำงานมาเป็น PAD command หรือไม่

การส่งสัญญาน Escape from Data Transfer นั้นอาจทำได้โดยการส่ง Data link Escape sequence เช่น Ctrl-P ไปยัง PAD เมื่อกลับเข้ามาใน PAD command mode แล้ว ผู้ใช้สามารถใช้ X.28 command ได้ และยังสามารถกลับไปยัง data transfer mode ได้อีกด้วย

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่อนุญาตให้มีการ Escape from data transfer

1 อนุญาตให้มีการ Escape from data transfer ได้

Parameter 2 Echo

Parameter 2 เป็นตัวกำหนดว่า จะให้มีการแสดงตัวอักษรขึ้นบนหน้าจอเองโดย PAD หรือไม่ หรืออาจให้การแสดงตัวอักษรนี้ทำโดย remote DTE โดยใช้ X.29

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่ให้มีการ echo

1 ให้มีการ echo

Parameter 3 data forwarding Character(s)

นอกจากเราจะให้มีการ ส่ง packet เมื่อ packet เต็มแล้ว เราอาจส่งทันทีที่ได้รับอักษรที่บอกให้ทำการส่งก็ได้ โดยวิธีนี้ข้อมูลจะถูกเก็บอยู่ใน buffer จนกว่าจะมีอักษรที่ส่งให้ทำการส่ง packet ออกไป ดังนั้นอักษรที่เป็นตัวบอกให้มีการส่ง packet นั้นจะเป็นอักษรตัวสุดท้ายในทุกๆ packet

อักษรนี้ เป็นตัวที่มีบทบาทสำคัญในการกำหนดว่า ก่อนที่จะส่ง packet ออกไปนั้น แต่ละ packet มีข้อมูลอยู่เต็มหรือไม่ มากน้อยเพียงใด ถ้าต้องการที่จะลดปริมาณข้อมูลที่วิ่งอยู่บน network เราควรจะให้มีจำนวนข้อมูลอยู่ในแต่ละ packet มากที่สุด โดยทั่วๆไป อักษรที่ใช้เป็นตัวกำหนดในการส่งข้อมูลมักใช้เป็น carriage return (CR)

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0	ไม่มีการใช้อักษรใดเป็นตัวกำหนดในการส่งข้อมูล
1	ใช้อักษรที่เป็นตัว alphanumeric (A-Z, a-z, 0-9) เป็นตัวกำหนด
2	ใช้อักษร carriage return (CR) เป็นตัวกำหนด
4	ใช้อักษร ESC, BEL, ENQ, ACK เป็นตัวกำหนด
8	ใช้อักษร DEL, CAN, DC2 เป็นตัวกำหนด
16	ใช้อักษร ETX, EOT เป็นตัวกำหนด
32	ใช้อักษร HT, LF, VT, FF เป็นตัวกำหนด
64	ใช้อักษรทุกๆตัวที่เป็น ASCII ที่นอกเหนือจากที่กำหนดไว้ข้างบน (ยกเว้น DEL) เป็นตัวกำหนด

ตัวอย่างการกำหนด Parameter 3 นี้ที่เป็นไปได้เช่น (2+4) = 6 หรือ (1+2+4+8+16+32+64) = 127

Parameter 4 Idle Timer

วิธีตรงกันข้ามกับการส่งแบบ data forward ที่กำหนดโดย Parameter 3 ก็คือวิธีการใช้ Idle Timer เป็นตัวกำหนด โดยในการส่ง packet ออกไปนั้นจะขึ้นอยู่กับเวลาที่กำหนดให้ว่าข้อมูลที่จะส่งแต่ละตัวจะห่างกันนานเท่าไหร่ ค่าของ Idle time จะอยู่ระหว่าง 0 ถึง 255

สำหรับในระบบที่เป็น interactive นั้น การกำหนดให้ Idle time มีค่าต่ำๆ จะทำให้ได้ response ที่ดี แต่ในขณะเดียวกันก็เป็นการสิ้นเปลือง เพราะ packet มีขนาดเล็ก

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่ใช้ Idle Timer

1-255 Delay เป็น 20 ส่วนของวินาที (ค่า 20 = 1 วินาที)

Parameter 5 Flow Control by PAD

Parameter นี้เป็นตัวกำหนดว่าจะให้การ flow นั้นถูกควบคุมโดย PAD หรือไม่ และจะใช้วิธีการควบคุมแบบ X-ON/X-OFF หรือไม่ การควบคุมแบบนี้ไม่จำเป็นสำหรับการทำงานแบบ interactive เพราะความสามารถในการพิมพ์ของคนจะไม่มีทางล้น input buffer ของ PAD แต่ในการทำงานที่ข้อมูลมีการส่งอย่างต่อเนื่อง

เช่น การ transfer file นั้น จะต้องการการควบคุมโดยวิธีนี้ เพื่อป้องกันการสูญหายของข้อมูล

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่ใช้ X-ON/X-OFF flow control

1 ใช้ X-ON (ASCII DC1) /X-OFF (ASCII DC3) flow control

Parameter 6 Control of PAD Service Signals

PAD Service Signal เป็น network status message ที่ส่งโดย PAD ไปยัง terminal ต่างๆ เพื่อให้ข้อมูลที่จำเป็นสำหรับผู้ใช้ที่แต่ละ terminal ซึ่งเกี่ยวกับ status ของผู้ใช้ที่ทำการเรียกใช้ นอกจากนี้มันยังอาจใส่เข้าไปใน application ที่เป็น noninteractive เพื่อบอกความผิดพลาดที่เกิดขึ้นได้อีกด้วย

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่มีการส่ง service signal

1 มีการส่ง service signal

Parameter 7 Action on Break

การกำหนด Parameter นี้จะเป็นการบอกถึงสิ่งที่ PAD จะต้องทำ

เมื่อได้รับสัญญาน break โดยทั่วไปการที่ผู้ใช้ส่งสัญญาน break นั้น เพื่อบอกให้ยกเลิกการทำงานของโปรแกรมนั้นๆ ดังนั้น Parameter ที่ 7 นี้จะใช้เพื่อให้การทำเช่นนี้เร็วขึ้น หรือเพื่อไม่ให้มีการ break เกิดขึ้น

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0	ไม่มีการทำอะไรเมื่อได้รับสัญญาน break
1	PAD ส่ง X.25 Interrupt Packet
2	PAD ส่ง X.25 Reset Packet
4	PAD ส่ง Indication-of-Break message
8	ออกจาก data transfer mode
16	ไม่มีการแสดงผลออกทาง terminal

ตัวอย่างการกำหนด Parameter 7 นี้ที่เป็นไปได้เช่น (1+4) = 5 หรือ (1+4+16) = 21

Parameter 8 Discard Output

เมื่อ Parameter นี้ถูกใช้ระหว่างการเรียกใช้ ข้อมูลที่ถูก disassemble โดย PAD และส่งไปยัง terminal นั้น จะถูก discard

Parameter 8 นี้จะใช้ร่วมกับ Parameter 7 กล่าวคือ ถ้า Parameter 7 ถูกกำหนดเป็นค่า 16 จะส่งผลให้ Parameter 8 เป็น 1 คือจะ discard output

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ส่งข้อมูลปกติ

1 Discard output

Parameter 9 Carriage-Return PAdding

จำนวน padding character ที่จะส่งไปหลังจากอักษร Carriage Return (CR) จะกำหนดได้โดย Parameter ที่ 9 นี้ คุณสมบัตินี้มีไว้สำหรับ printer ที่ช้าๆ ที่ต้องใช้เวลาในการเลื่อนหัวอ่านเมื่อจะต้องขึ้นบรรทัดใหม่ padding character นั้นมักจะใช้เป็น ASCII NULL

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่ต้องส่ง padding character

1-7 จำนวน padding character ที่ต้องส่งหลัง CR

Parameter 10 Line Folding

Parameter 10 นี้ใช้สำหรับในกรณีที่จำนวนอักษรต่อ 1 บรรทัดของ host และ terminal มีขนาดไม่เท่ากัน บาง host จะส่งข้อมูลโดยมีจำนวนอักษรต่อบรรทัดเท่ากับจำนวนสูงสุดที่จะเป็นไปได้ (132 อักษรต่อบรรทัด) แต่ในขณะที่ terminal อาจแสดงได้เพียง 80 อักษรต่อบรรทัด และ terminal นั้นอาจไม่มีระบบ wraparound ดังนั้น Parameter 10 นี้จะต้องถูกกำหนดให้เป็น 80 เพื่อให้เพื่อ carriage return เข้าไปทุกๆ 80 ตัวอักษรที่ได้รับมา

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่ต้องมี line folding

1 จำนวนอักษรต่อบรรทัด

Parameter 11 Speed

Parameter 11 นี้จะไม่สามารถกำหนดได้เองโดยผู้ใช้ จะขึ้นอยู่กับว่า network ของแต่ละ port นั้นๆ สามารถรองรับได้ที่ความเร็วเท่าใด

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0	110 bit/s
1	134.5 bit/s
2	300 bit/s
3	1,200 bit/s
4	600 bit/s
5	75 bit/s
6	150 bit/s
7	1,800 bit/s
8	200 bit/s
9	100 bit/s
10	50 bit/s
11	75/1,200 bit/s (Videotex)
12	240 bit/s
13	4,800 bit/s
14	9,600 bit/s
15	19,200 bit/s
16	48,000 bit/s

17 56,000 bit/s

18 64,000 bit/s

Parameter 12 Flow Control by Terminal

เป็น Parameter ที่ใช้กำหนด Flow control ในทิศทางตรงกันข้ามกับที่กำหนดโดย Parameter 5

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่ใช้ X-ON/X-OFF flow control

1 ใช้ X-ON (ASCII DC1) /X-OFF (ASCII DC3) flow control

Parameter 13 Line-Feed Insertion

Parameter 13 ใช้ในการกำหนด line feed ที่รับและส่งกับ terminal ที่แตกต่างกันหลัง carriage-return

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่มีการเพิ่ม line-feed

1 เพิ่ม line-feed หลัง carriage-return ที่ส่งไปยัง terminal

2 เพิ่ม line-feed หลัง carriage-return ที่รับมาจาก terminal

4 เพิ่ม line-feed หลัง echo carriage-return ไปยัง terminal

ตัวอย่างการกำหนด Parameter 13 นี้ที่เป็นไปได้เช่น (1+4) = 5 หรือ (1+2+4) = 7

Parameter 14 Padding after Line-feed

Parameter 14 นี้ทำหน้าที่เหมือนกับ Parameter 9 แต่มีข้อแตกต่างกันที่ Parameter 14 นี้จะส่ง padding Character หลังจาก line-feed แทนที่จะส่ง carriage-return

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่ต้องส่ง padding character

1-7 จำนวน padding character ที่ต้องส่งหลัง LF

Parameter 15 Editing

Parameter 15 นี้จะใช้ร่วมกับ Parameter 16, 17 และ 18 ในการ edit อักษรที่อยู่ใน buffer ของ PAD ก่อนที่จะส่งไป ดังนั้นถ้ากำหนดให้การส่งเป็นแบบ Idle time จะต้องกำหนดให้ Parameter 15 นี้เป็นแบบไม่มีการ edit

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่มีการ edit

1 มีการ edit

Parameter 16 Character-delete Character

ASCII ที่กำหนดโดย Parameter 16 นี้จะเป็นตัวที่ใช้เป็น delete character ในการ edit โดยทั่วไปจะใช้ DEL (ASCII 127)

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่มีการใช้

1-127 ASCII ที่ใช้เป็น delete-character

Parameter 17 Buffer-delete Character

อักษรที่กำหนดโดย Parameter 17 นี้จะเป็นอักษรที่ใช้บอกให้มีการลบข้อมูลที่อยู่ใน buffer ของ PAD ออก โดยทั่วไปจะใช้ Ctrl-X (ASCII 24)

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่มีการใช้

1-127 ASCII ที่ใช้เป็นอักษรสำหรับส่ง buffer delete

Parameter 18 Buffer-display Character

อักษรที่กำหนดโดย Parameter 18 นี้จะเป็นอักษรที่ใช้บอกให้มี

การ redisplay ข้อมูลที่อยู่ใน buffer ของ PAD โดยทั่วไปจะใช้ Ctrl-R (ASCII 18)

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0	ไม่มีการใช้
1-127	ASCII ที่ใช้เป็นอักษรสำหรับส่ง buffer display

Parameter 19 Edit Service Signals

ในการ edit ของ hard-copy กับ video-display terminal นั้นจะแตกต่างกัน เนื่องจากในกรณีของ hard-copy นั้น เราไม่สามารถกลับไปลบส่วนที่พิมพ์ไปแล้วไม่ได้ ดังนั้นเมื่อต้องการลบนั้น เราจึงใช้วิธีส่งอักษร "\" ไปเขียนทับ แต่สำหรับกรณีของ video-display terminal เราจะใช้วิธีส่ง "BS space BS" ไปเพื่อลบข้อมูล ในกรณีของ hard-copy นั้นเมื่อลบจนหมดบรรทัดใน buffer ของ PAD แล้ว PAD จะ response "XXX",CR, LF

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0	ไม่มีการ edit ของ PAD service signal
1	Hard-copy terminal
2	video-display terminal
8	BS จะถูกส่งเป็น response สำหรับการลบ
32-126	ASCII ที่จะถูกส่งเป็น response สำหรับการลบ

Parameter 20 Echo Mask

Parameter 20 นี้ใช้ในการกำหนดว่าอักษรตัวใดจะให้มีการ echo ตัวใดไม่ให้มีการ echo

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ให้ echo ทุกๆตัวอักษร

1 ไม่ให้ echo อักษร CR

2 ไม่ให้ echo อักษร LF

4 ไม่ให้ echo อักษร VT, HT, FF

8 ไม่ให้ echo อักษร BEL, BS

16 ไม่ให้ echo อักษร ESC, ENQ

32 ไม่ให้ echo อักษร ACK,NAK,STX,SOH,EOT.ETB.ETX

64 ไม่ให้ echo อักษรที่กำหนดเป็น edit-character ใน Parameter 16,17,18

128 ไม่ให้ echo อักษร DEL และอักษรอื่นที่เป็น Control-character นอกเหนือจากที่กำหนดไว้ด้านบน

Parameter 21 Parity Treatment

Parameter 21 นี้เป็นตัวกำหนดว่าให้มีการตรวจสอบ parity หรือ การสร้าง parity ส่งไป หรืออาจทำทั้ง 2 อย่างควบคู่กันไป Parameter 21 นี้จะต้องกำหนดให้เป็น 0 ถ้ามีการส่งข้อมูลขนาด 8-bit ทั้งในกรณีของการส่งข้อมูลแบบ graphic หรือการส่งใน binary mode

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่มีการตรวจสอบ parity

1 มีการตรวจสอบ parity

2 มีการ generate parity

3 มีการตรวจสอบ และ generate parity

Parameter 22 Page Wait

ในกรณี terminal ที่เป็นแบบ scholling เราสามารถกำหนดโดย Parameter 22 นี้ได้ว่าให้หยุดทุกครั้งเมื่อจำนวน line-feed เท่ากับจำนวนบรรทัดของ full screen แล้ว

ค่าของ Parameter ที่จะเป็นไปได้มีดังนี้

0 ไม่มีการหยุดรอแต่ละหน้า

1-255 จำนวน line-feed ที่ส่งมาโดย PAD ก่อนที่จะหยุดรอแต่ละหน้า

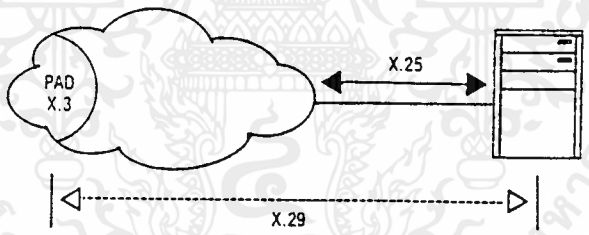
X.28 Command and response

หลังจากที่ผู้ใช้ ได้เข้าสู่ PDN node X.28 Command จะอนุญาตให้ผู้ใช้ติดต่อโดยตรงเข้ากับ PAD และทำการ Call เข้ากับ Network ได้ ดังรูปที่ 9-6 ถ้าจำเป็น คำสั่งต่าง ๆ เหล่านี้ สามารถเปลี่ยนแปลงได้ที่ X.3 Parameter โดยใช้คำสั่งทั้ง 9 คำสั่งของ PAD ดังรูป 9-7 โดยสามารถแบ่งออกได้ 3 ประเภท คือ การทำการ Call, อ่านและเปลี่ยนแปลง X.3 Parameter และตรวจสอบหรือ reset การ Call เมื่อผู้ใช้ส่ง Command ไปแล้ว PAD จะทำการตอบสนองด้วย PAD Service Signal ทั้ง 12 คำสั่งดังรูปที่ 9-8

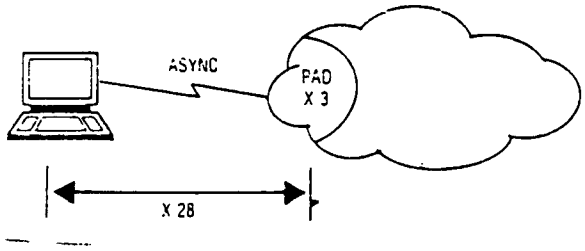
X.29 AND REMOTE PAD CONFIGURATION

นอกจากการใช้ X.28 ในการเปลี่ยนแปลง Configuration ของ PAD แล้ว เรายังสามารถอนุญาตให้ Remote Host ทำการเปลี่ยนแปลง Config ของ PAD ได้ทันทีอัตโนมัติหลังจากมีการ LOG ON โดยการใช้ X.29 Message ระหว่าง PAD และ Remote DTE หรือ Host

X.29 Message จะส่งไปบน Network ใน User Data Field ของ X.25 Packet ดังรูปที่ 9-9 Packet นี้จะถูกสังเกตได้ว่าเป็น X.29 Packet ได้ เพราะจะมี Q bit ที่หัวของ Packet เป็น 1 ดังรูปที่ 9-10 ส่วน X.29 Code นั้น จะอยู่ใน 4 บิทล่างของ octet แรกใน User Data Field



รูปที่ 9-9 X.25 และ X.29



รูปที่ 9-6 X.3 และ X.28

PAD Command signal format	Function	PAD service signal sent in response
STAT	To request status information regarding a virtual call connected to the DTE.	FREE or ENGAGED
CLR	To clear a virtual call	CLF CONF or CLR ERR (in case of local procedure error)
PAR?	To request the current values of specified parameters	PAR (list of parameter references with their current values or INV)
SET?	To request changing or setting of the current values of the specified parameters and to request the current values of specified parameters	PAR (list of parameter references with their current values or INV)
PROF	To give PAD parameters a standard set of values	Acknowledgement
RESET	To reset the virtual call	Acknowledgement
INT	To transmit an interrupt packet	Acknowledgement
SET	To set or change parameter values	Acknowledgement
Selection PAD command signal	To set up a virtual call	Acknowledgement

Source: CCITT X.28 Recommendation, Annex A

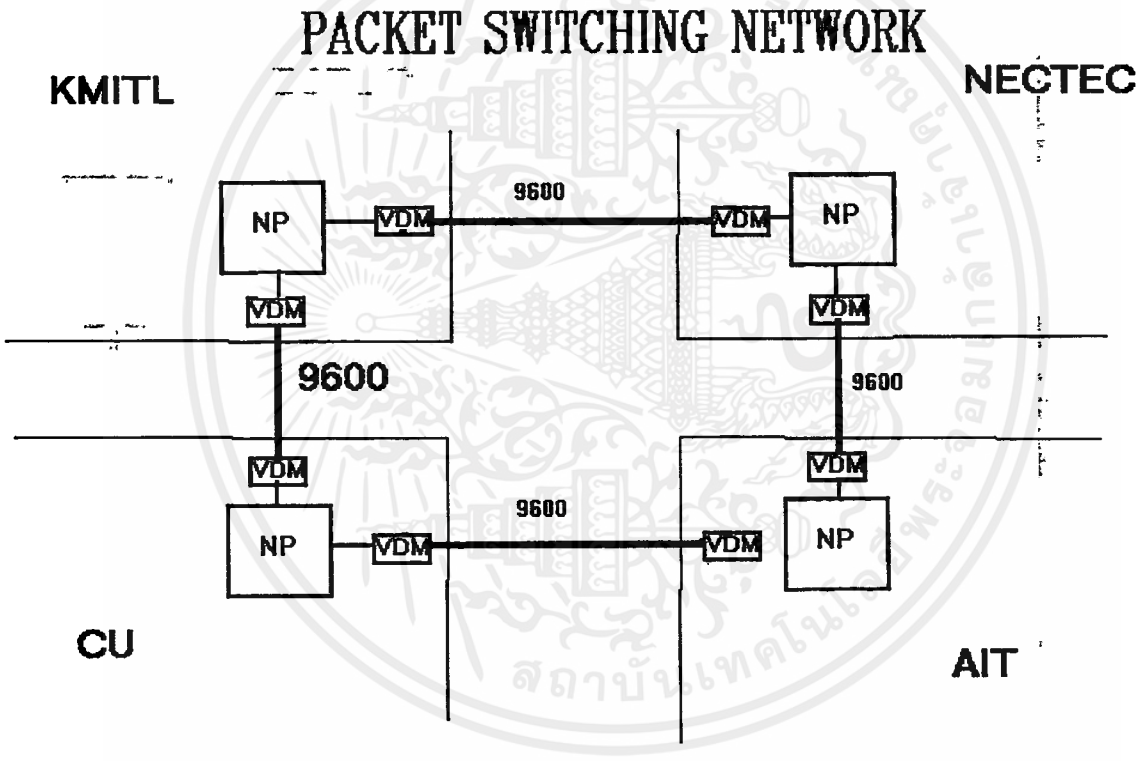
รูปที่ 9-7 PAD Command Signal

Format of the PAD service signal			Explanation
RESET	DTE	1, 2, or 3 characters which represent the decimal value of the diagnostic code	Indication that the remote DTE has reset the virtual call
	ERR		Indication of a reset of a virtual call due to a local procedure error
	NC		Indication of a reset of a virtual call due to network congestion
CLR			Indication of clearing
COM			Indication of call connected
PAD identification PAD service signal	The characters to be sent are network dependent		
ERROR	ERR		Identification that a PAD command is in error
(Network dependent)			Indication of incoming call
XXX			Indication of line delete function completed
ENGAGED			Response to status PAD command signal when a call is not established
FREE			Response to status PAD command signal when a call has been established
PAD	Decimal value of parameter: parameter value or INV		Response to set-and-read or read PAD command signal
(Network dependent)			Prompt PAD service signal
Format effector			Acknowledgement PAD service signal

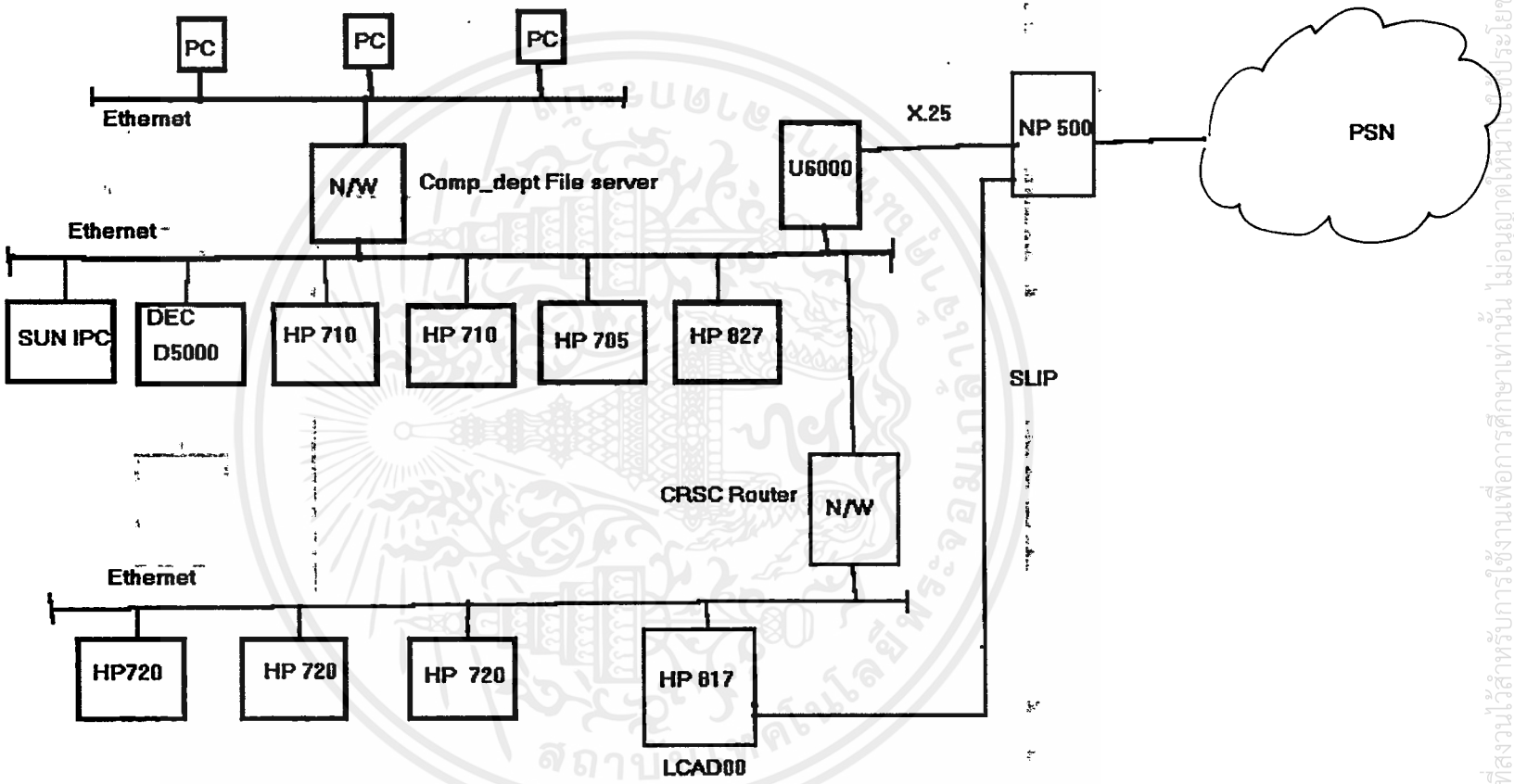
Source: CCITT X.28 Recommendation, Annex A

รูปที่ 9-8 PAD Service Signals

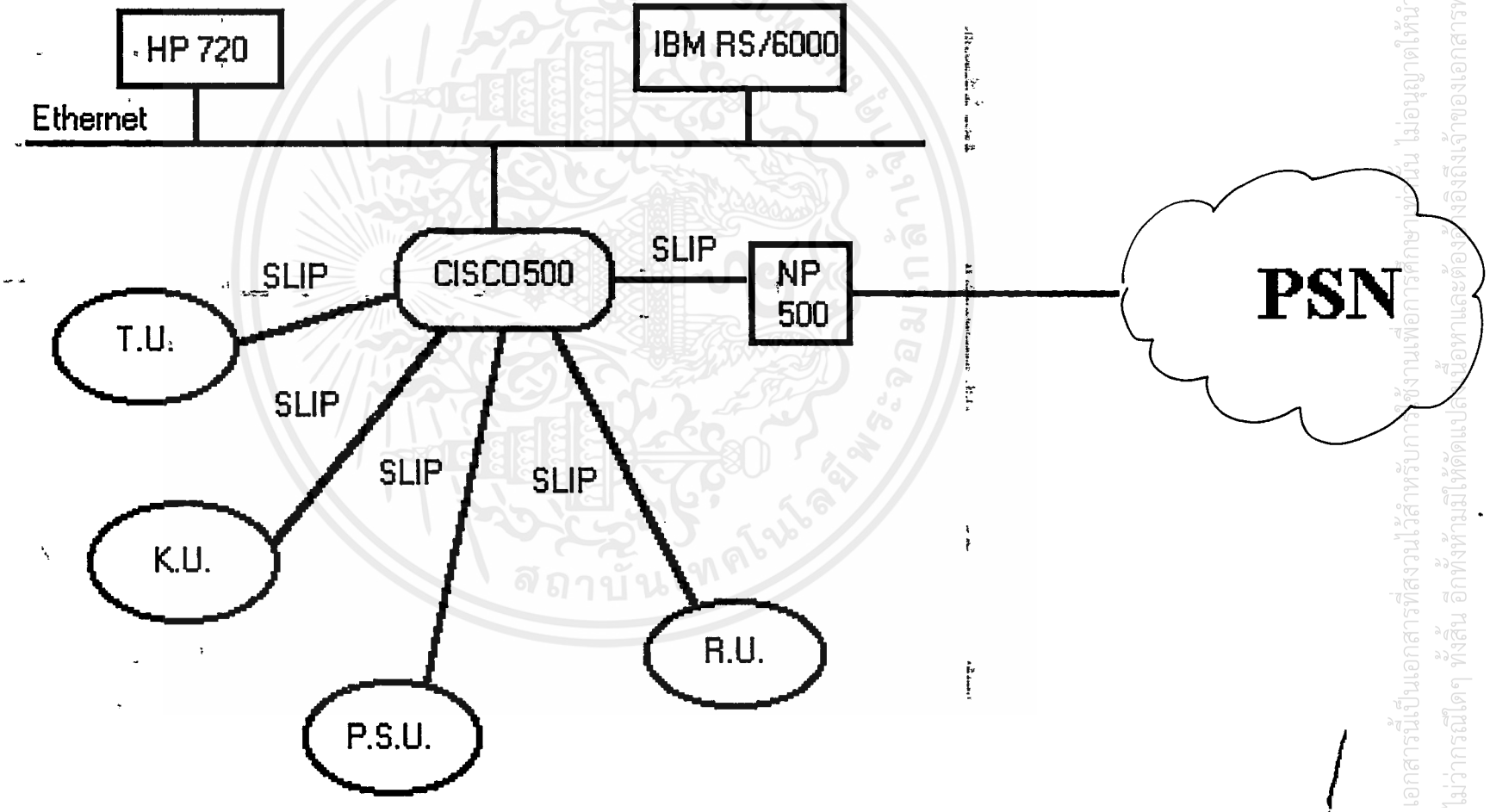
ภาคผนวก
ระบบในปัจจุบัน



KMITL NETWORK



NECTEC



E-MAIL