บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

โปรโตคอลทีซีพี/ไอพี

1) ประวัติความเป็นมา (Carl-Michel,1993; Hunt,1992) ในปี ค.ศ. 1969 หน่วยงาน ARPA (Advance Research Project Agency) ซึ่งเป็นหน่วยงานหนึ่งในกระทรวงกลาโหมของสหรัฐอเมริกาได้ทดลองเครือข่ายแบบสลับแพกเกต (packet switching network) โดยเรียกว่า เครือข่ายอาร์พาเนต (ARPANET) ซึ่งเป็นเครือข่ายแบบสลับแพกเกตแรกในโลก เพื่อใช้สำหรับการศึกษาวิจัย รวมถึงมีบทบาทสำคัญสำหรับทางทหาร โดยเฉพาะอย่างยิ่งใน สงครามโลกครั้งที่ 1

ต่อมาในปี ค.ศ. 1975 เครือข่ายอาร์พาเนตได้ถูกใช้งานเป็น เครือข่ายปฏิบัติงานจริง โดยอยู่ในความรับผิดชอบของ หน่วยงานการสื่อสารทางทหาร (Defense Communication Agency) ซึ่งในช่วงนี้เอง โปรโตคอลทีซีพี/ไอพี (TCP/IP protocol) ได้เริ่มถูกพัฒนาขึ้นมา

โปรโตคอลทีซีพี/ไอพี ถูกกำหนดให้เป็นมาตรฐานทางทหาร (Military Standards) ในปี ค.ศ. 1983 โดยกำหนดให้ทุกเครื่องที่ต้องการเชื่อมโยงกับเครือข่ายนี้จะต้องเปลี่ยนแปลงโปรโตคอลให้เป็นตามโปรโตคอลทีซีพี/ไอพี อาร์พาเนตเดิมถูกแบ่ง เป็น 2 ส่วน คือ มิลเนต (MILNET) สำหรับการใช้งานทางทหาร และ อาร์พาเนตที่มีขนาดเล็กลง โดยคำว่าอินเตอร์เนต (Internet) ได้เป็นชื่อที่ถูกเรียกแทนทั้ง 2 ส่วนดังกล่าว และนอกจากนี้อินเตอร์เนตยังได้รวมเอาเครือข่ายต่างๆเข้ามาเป็นส่วนหนึ่งของเครือข่าย เช่น เอ็นเอฟเอสเนต (NSFnet) เป็นต้น

หลังจากที่อินเตอร์เนตถูกใช้งานเป็นเครือข่ายที่ใหญ่ที่สุดในโลก ทำให้การใช้งานโปรโตคอลทีซีพี/ไอพี แพร่หลาย และเป็นโปรโตคอลที่สำคัญและมีการใช้งานมากที่สุดโปรโตคอลหนึ่ง

2) สถาปัตยกรรมของโปรโตคอล (protocol architecture) (Hunt,1992) โปรโตคอลทีซีพี/ไอพี เป็นโปรโตคอลที่ได้รับความนิยมไม่ใช่เพราะว่าอินเตอร์เนต หรือ ทางการทหารสหรัฐใช้ แต่เนื่องจากคุณสมบัติที่ดีของโปรโตคอล ได้แก่

– เป็นโปรโตคอลมาตรฐานของระบบเปิด โปรโตคอลทีซีพี/ไอพี สามารถพัฒนาได้โดยไม่ขึ้นอยู่กับฮาร์ดแวร์ หรือ ระบบปฏิบัติการใดๆ

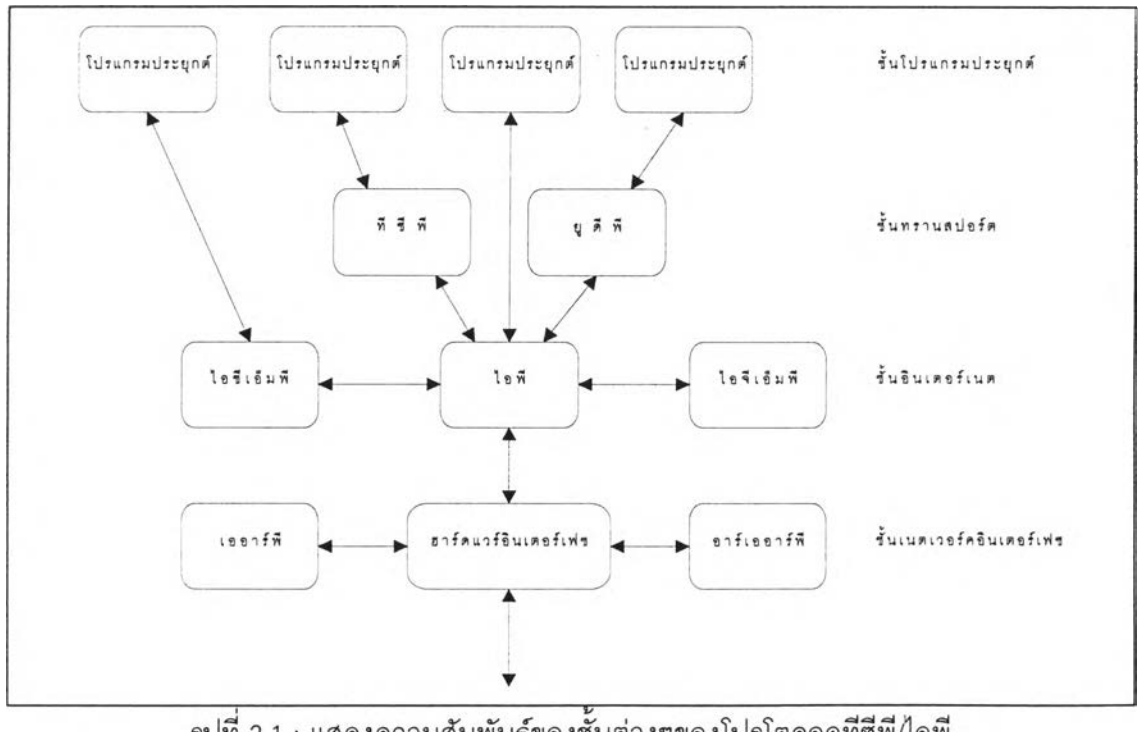
– เป็นโปรโตคอลที่เป็นอิสระจาก เครือข่ายทางกายภาพ (physical network) เนื่องจากโปรโตคอลทีซีพี/ไอพี สามารถทำงานได้บน อีเทอร์เนต (Ethernet), โทกเกนริง (Token Ring), เอฟดีดีไอ (FDDI), พอร์อนุกรมอาร์เอส232 (RS/232) หรือ เครือข่ายสลับแพกเกตเอ็กซ์25 (X.25) เป็นต้น

– การกำหนดที่อยู่ (addressing) เป็นรูปแบบที่จำแนกเครื่องต่างๆ และ เครือข่ายได้เป็นอย่างดี

– ความเป็นมาตรฐานของโปรโตคอลระดับบน ซึ่งมีการใช้งานกว้างขวาง เช่น เทลเนต (TELNET) , เอฟทีพี (FTP) เป็นต้น

ชั้น (layer) ของโปรโตคอลทีซีพี/ไอพี มี 4 ชั้น แบ่งจากชั้นบนลงล่าง ได้ดังนี้

– ชั้นโปรแกรมประยุกต์ (application layer)
– ชั้นทรานสปอร์ต (transport layer)
– ชั้นอินเตอร์เนต (internet layer)
– ชั้นเนตเวอร์คอินเตอร์เฟซ (network interface layer)



รูปที่ 2.1 : แสดงความสัมพันธ์ของชั้นต่างๆของโปรโตคอลทีซีพี/ไอพี

2.1) ชั้นโปรแกรมประยุกต์ (application layer) ชั้นโปรแกรมประยุกต์เป็นชั้นของโปรโตคอลสำหรับโปรแกรมประยุกต์บนเครือข่ายทีซีพี/ไอพี รูปแบบของโปรแกรมประยุกต์ในชั้นนี้จะเป็นลักษณะของผู้ขอรับบริการ/ผู้ให้บริการ (client/server model) นั่นคือในการใช้งานโปรแกรมประยุกต์ หนึ่งๆ จะมีโปรเซสหนึ่งทำหน้าที่เป็น โปรเซสขอรับบริการ (client process) และ โปรเซสสำหรับให้บริการ (server process) ผ่านทางเครือข่ายโดยมีโปรโตคอลสำหรับโปรแกรมประยุกต์ร่วมกัน โปรแกรมประยุกต์ต่างๆบนโปรโตคอลทีซีพี/ไอพี เช่น เทลเนต (TELNET), เอฟทีพี (FTP), เอสเอ็มทีพี (SMTP) เป็นต้น

2.2) ชั้นทรานสปอร์ต (transport layer) ชั้นการรับส่งทำหน้าที่ควบคุมการรับส่งระหว่างกัน ในชั้นนี้มีโปรโตคอลหลักๆ 2 โปรโตคอล คือ ทีซีพี (TCP) และ ยูดีพี (UDP)

ทีซีพี มีชื่อเต็มว่า " Transmission Control Protocol " เป็นโปรโตคอลแบบต้องมีการติดต่อกันก่อนของผู้รับส่งข้อมูล (connection-oriented) ซึ่งทำให้เป็นโปรโตคอลที่มีความเชื่อถือได้ (reliable) ของการรับส่ง ในรายละเอียดของโปรโตคอลนี้จะกล่าวรายละเอียดใน RFC 793 (Postel, 1981b)

ยูดีพี มีชื่อเต็มว่า " User Datagram Protocol " เป็นโปรโตคอลที่มีการทำงานง่ายกว่าทีซีพี การรับส่งข้อมูลเป็นลักษณะ ดาตาแกรม (datagram) ข้อมูลที่รับส่งไม่จำเป็นจะต้องมีการติดต่อกันก่อน (connectionless) ทำให้การรับส่งของข้อมูลขาดความเชื่อถือได้ รายละเอียดของโปรโตคอลถูกกำหนดใน RFC 768 (Postel, 1980)

2.3) ชั้นอินเตอร์เนต (internet layer) ชั้นอินเตอร์เนต มีชื่อเรียกอีกอย่างหนึ่งว่า ชั้นเครือข่าย (network layer) ในชั้นนี้โปรโตคอลที่สำคัญคือ ไอพี (IP) อันเป็นหัวใจของชุดโปรโตคอลทีซีพี/ไอพี หน้าที่สำคัญคือการส่งข้อมูล (route) ไปตามจุดต่างๆในเครือข่าย สำหรับรายละเอียดของไอพี ถูกกำหนดใน RFC 791 (Postel, 1981a)

นอกจากไอพีแล้ว ในชั้นนี้ยังมีโปรโตคอลอีก 2 โปรโตคอลคือ ไอซีเอ็มพี (ICMP : Interner Control Message Protocol) และ ไอจีเอ็มพี (IGMP : Internet Group Management Protocol) ซึ่งรายละเอียดของทั้ง 2 จะไม่กล่าวถึงในวิทยานิพนธ์นี้

2.4) ชั้นเนตเวอร์คอินเตอร์เฟซ (network interface layer) ชั้นเนตเวอร์คอินเตอร์เฟซ มีชื่อเรียกอีกอย่างหนึ่งว่า ชั้นดาตาลิงค์ (data link layer) หรือ ชั้นลิงค์ (link layer) ทำหน้าที่เป็นดี

ไวซ์ไดร์เวอร์ (device driver) ของระบบปฏิบัติการ เพื่อทำให้สามารถรับส่งข้อมูลกับฮาร์ดแวร์สำหรับการสื่อสารข้อมูลในเครือข่าย

3) การอ้างอิงที่อยู่ของไอพีและพอร์ต (IP addressing and port) ในระบบโปรโตคอลสำหรับเครือข่ายต่างๆ นั้นจำเป็นจะต้องมีวิธีการในการอ้างอิงที่อยู่ของจุด (node) ต่างๆ ซึ่ง โปรโตคอลทีซีพี/ไอพี จะใช้ ที่อยู่ของไอพี (IP address) ในการกำหนดที่อยู่ของจุดต่างๆในระบบเครือข่าย ที่อยู่ของไอพี จะเป็นตัวเลขขนาด 32 บิต โดยแสดงในรูปเลขฐาน 10 จำนวน 4 ตัว คั่นกลางด้วย "." (จุด) ตัวอย่างเช่น 192.1.1.1 , 148.1.2.35 เป็นต้น

ที่อยู่ของไอพี แบ่งออกเป็น 2 ส่วน ได้แก่

- ที่อยู่ของเครือข่าย (network address)
- ที่อยู่ของเครื่อง (host address)

ที่อยู่ของเครือข่ายจะเป็นการบ่งบอกว่าที่อยู่ของจุดนั้นๆ อยู่ที่เครือข่ายใด และ ที่อยู่ของเครื่องเป็นที่อยู่ที่บอกว่าในเครือข่ายนั้นๆ เป็นเครื่องใด

ในโปรโตคอลทีซีพี/ไอพี แบ่งระดับ (class) ของที่อยู่ของไอพี เป็น 5 ระดับ (Carl-Mitchell, 1993) ดังนี้

ระดับ	จำนวนหลักของที่อยู่ของเครือข่าย	ช่วงของที่อยู่ของเครือข่าย	จำนวนหลักของที่อยู่ของเครื่อง	ช่วงของที่อยู่ของเครื่อง
A	1	1 - 127	3	0.0.0 - 255.255.255
B	2	128.0 - 191.255	2	0.0 - 255.255
C	3	192.0.0 - 222.255.255	1	0.255
D	4	224.0.0.0 - 239.255.255.255	-	-
E	4	240.0.0.0 - 255.255.255.255	-	-

ตารางที่ 2.1: ที่อยู่ของไอพีระดับต่างๆ

ที่อยู่ของไอพีที่ถูกสำรองไว้ โดยมีความหมายอย่างอื่น ได้แก่

- ที่อยู่ไอพีที่ส่วนของที่อยู่ของเครื่องทั้งหมดเป็น 0 หมายถึง การอ้างอิงที่อยู่ของเครือข่าย
- ที่อยู่ไอพีที่ส่วนของที่อยู่ของเครื่องทั้งหมดเป็น 1 หมายถึง การอ้างอิงที่อยู่สำหรับการประกาศ (broadcast address)

- ที่อยู่ไอพี 127.0.0.1 หมายถึง ที่อยู่ภายในเครื่องนั้น
- ที่อยู่ไอพี 255.255.255.255 หมายถึง ทุกเครื่อง

ในการทำงานของโปรแกรมประยุกต์ของโปรโตคอลทีซีพี/ไอพี ทั้งที่ใช้โปรโตคอลทีซีพี และ ยูดีพี ต้องมีกระบวนการในการระบุถึงโปรเซสที่ทำการติดต่อด้วย ซึ่งทั้งทีซีพี และ ยูดีพี ใช้เลข 16 บิต เรียกว่า พอร์ต (port)

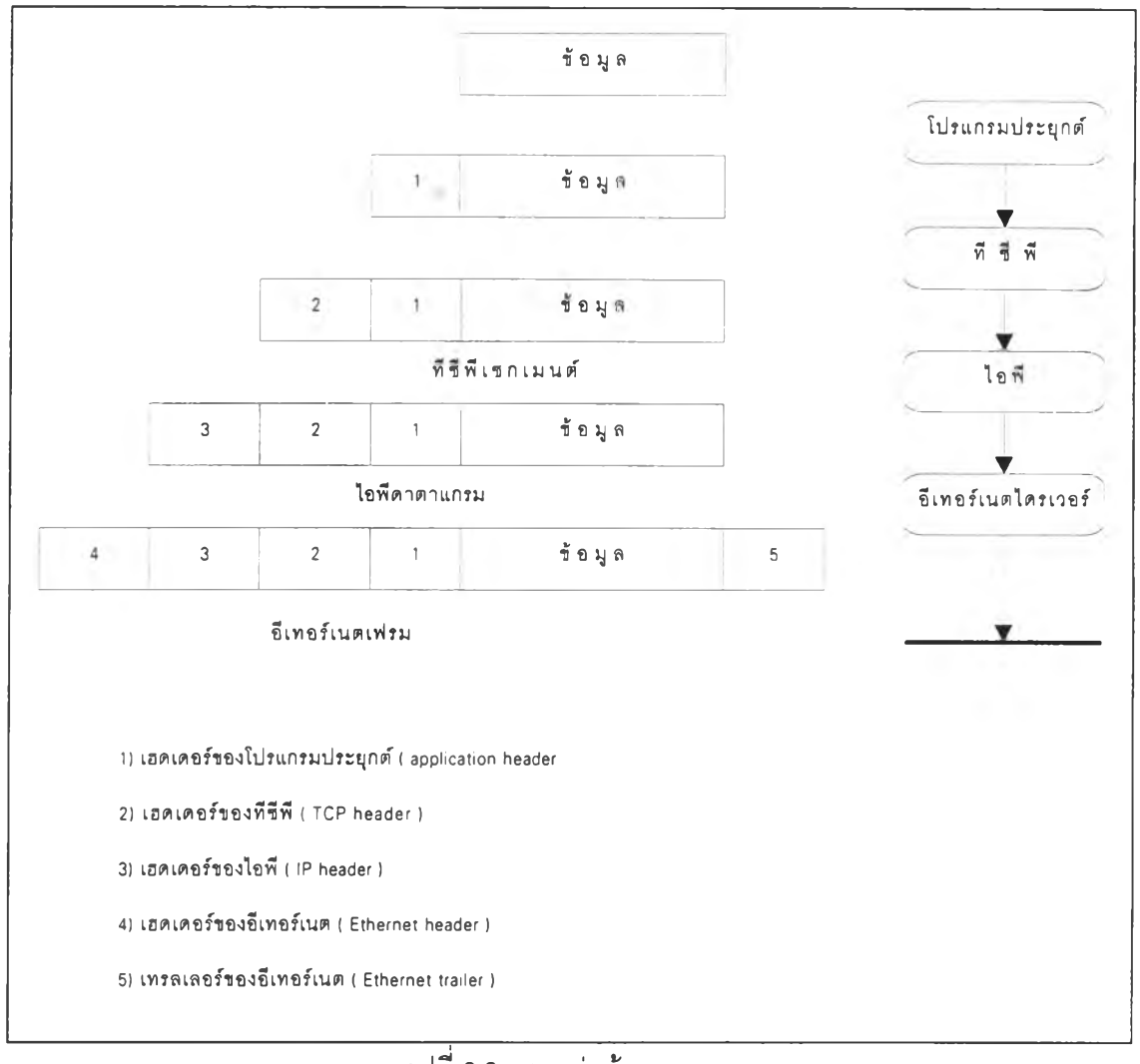
โปรเซสให้บริการจะต้องมีพอร์ตที่เป็นที่รู้จักทั่วไป (wellknown port) เพื่อให้ผู้ขอรับบริการสามารถที่จะติดต่อมาได้ เช่น โปรเซสให้บริการเทลเนต (TELNET server) จะใช้พอร์ต 23 หรือ โปรเซสให้บริการเอฟทีพี (FTP server) จะใช้พอร์ต 21

สำหรับโปรเซสขอรับบริการจะกำหนดพอร์ตขึ้นมาพอร์ตหนึ่งเพื่อใช้ติดต่อกับพอร์ตของโปรเซสให้บริการ ดังนั้นในแต่ละการติดต่อระหว่างโปรเซสขอรับบริการ และ โปรเซสให้บริการจะใช้ค่าที่อ้างอิง 5 ค่า ดังต่อไปนี้

- โปรโตคอล ทีซีพี หรือ ยูดีพี
- ที่อยู่ไอพีของตัวเอง
- พอร์ตของตัวเอง
- ที่อยู่ไอพีของอีกด้านหนึ่ง
- พอร์ตของอีกด้านหนึ่ง

4) การห่อหุ้มแพกเกต (packet encapsulation) ในการสื่อสารข้อมูลตาม โปรโตคอลทีซีพี/ไอพี ระหว่างการผ่านชั้นต่างๆจากบนลงมาล่าง ข้อมูลจะถูกส่งเป็นรูปของแพกเกต (packet) โดยในแต่ละระดับชั้นจะมีการเพิ่มส่วนหัว (header) ของ แพกเกตเพื่อใช้สื่อสารกับโปรโตคอลระดับเดียวกันของอีกด้านหนึ่ง

หน่วยของข้อมูลที่ถูกเพิ่มเฮดเดอร์ของทีซีพี เรียกว่า ทีซีพีเซกเมนต์ (TCP segment) เมื่อส่งลงไปในชั้น ไอพี มีการเพิ่มเฮดเดอร์ของไอพี เรียกว่า ไอพีดาตาแกรม (IP datagram) และเมื่อไอพีดาตาแกรมถูกส่งลงไปชั้นเนตเวอร์คอินเตอร์เฟซ เช่น อีเทอร์เนต (Ethernet) จะมีการเพิ่มเฮดเดอร์ และ เทรลเลอร์ เรียกว่า อีเทอร์เนตเฟรม (Ethernet frame)



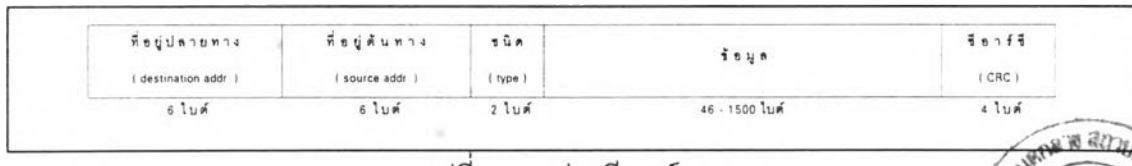
รูปที่ 2.2 : การห่อหุ้มแพกเกต

5) อีเทอร์เนตเฟรม (Ethernet frame) โปรโตคอลทีซีพี/ไอพี เป็นโปรโตคอลที่สามารถใช้งานได้กับชนิดของการรับส่งข้อมูลหลายรูปแบบเช่น อีเทอร์เนต (Ethernet), โทกเกนริง (token ring) หรือ เอฟดีดีไอ (FDDI) เป็นต้น สำหรับในวิทยานิพนธ์นี้จะกล่าวถึงรายละเอียดเฉพาะอีเทอร์เนตเท่านั้น

ประวัติความเป็นมา (Stevens , 1994) อีเทอร์เนตถูกคิดค้นขึ้นในปี ค.ศ 1982 โดย Digital Equipment Corp., Intel Corp. และ Xerox Corp. เพื่อใช้งานกับเครือข่ายท้องถิ่น (local area network) โดยการใช้เทคโนโลยีที่เรียกว่า ซีเอสเอ็มเอ/ซีดี (CSMA/CD : Carrrier Sense Multiple Access / Collision Detection) ทำงานที่ 10 เมกกะบิตต่อวินาที

หลังจากนั้น IEEE ได้กำหนดมาตรฐาน 802.3 ซึ่งมีลักษณะคล้ายคลึงกับอีเทอร์เนตแต่มีรายละเอียดที่แตกต่างออกไปซึ่งรายละเอียดจะอยู่นอกเหนือขอบเขตของวิทยานิพนธ์นี้

ลักษณะเฟรมของอีเทอร์เนตเป็นดังนี้

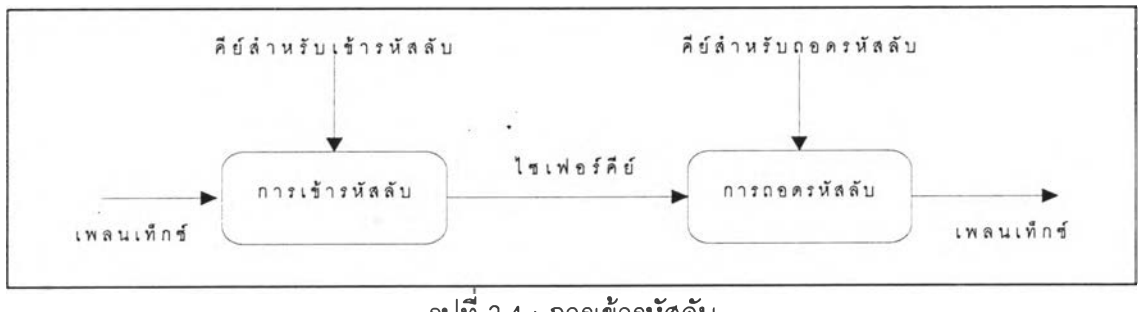


รูปที่ 2.3 : เฟรมอีเทอร์เนต

การเข้ารหัสลับข้อมูล (data encryption)

การเข้ารหัสลับข้อมูลเป็นกระบวนการเปลี่ยนแปลงข้อมูลรูปแบบที่อ่านได้ (readable message) ไปเป็นข้อมูลที่อ่านไม่เข้าใจ เพื่อป้องกันไม่ให้ผู้อื่นนำข้อมูลของเราไปใช้ ส่วนกระบวนการกลับกันเรียกว่า การถอดรหัสลับ (decryption)

ข้อมูลก่อนถูกเข้ารหัสลับเรียกว่า เพลนเท็กซ์ (plaintext) ส่วนข้อมูลที่ได้รับการเข้ารหัสลับ แล้วนั้น เรียกว่า ไซเฟอร์เท็กซ์ (ciphertext) ขั้นตอนวิธีในการเข้ารหัสลับ (encryption algorithm) จำเป็นต้องใช้คีย์ในการเข้ารหัสลับ (encryption key) และ ถอดรหัสลับ (decryption key) โดยสามารถอธิบายได้ตามภาพที่ 2.4



รูปที่ 2.4 : การเข้ารหัสลับ

1) ขั้นตอนวิธีสำหรับการเข้ารหัสลับ ขั้นตอนวิธีสำหรับการเข้ารหัสลับแบ่งเป็น 2 รูปแบบ ได้แก่ การเข้ารหัสลับแบบใช้คีย์เดียว (single key encryption) และ การเข้ารหัสลับแบบใช้คีย์สาธารณะ (public key encryption)

การเข้ารหัสลับแบบใช้คีย์เดียว เป็นรูปแบบการเข้ารหัสลับ ที่การเข้ารหัสลับ และ การถอดรหัสลับ เป็นคีย์เดียวกัน ตัวอย่างของขั้นตอนวิธีนี้ได้แก่ DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm), Lucifer, Madryga, FEAL-N, REDOC เป็นต้น

การเข้ารหัสลับแบบใช้คีย์สาธารณะ เป็น รูปแบบของการเข้ารหัสลับที่การเข้ารหัสลับ และ การถอดรหัสลับ ใช้คนละคีย์ คือ คีย์สาธารณะ (public key) ซึ่งเป็นคีย์ที่สามารถประกาศให้ผู้อื่นรับรู้ได้ และ คีย์ลับ (secret key) เป็นคีย์ที่ต้องเก็บไว้ รูปแบบความสัมพันธ์ของคีย์สาธารณะ, คีย์ลับ และ ขั้นตอนวิธีในการเข้ารหัสลับใช้กระบวนการทางคณิตศาสตร์ ซึ่งจะยกตัวอย่างให้เห็นในหัวข้อ RSA

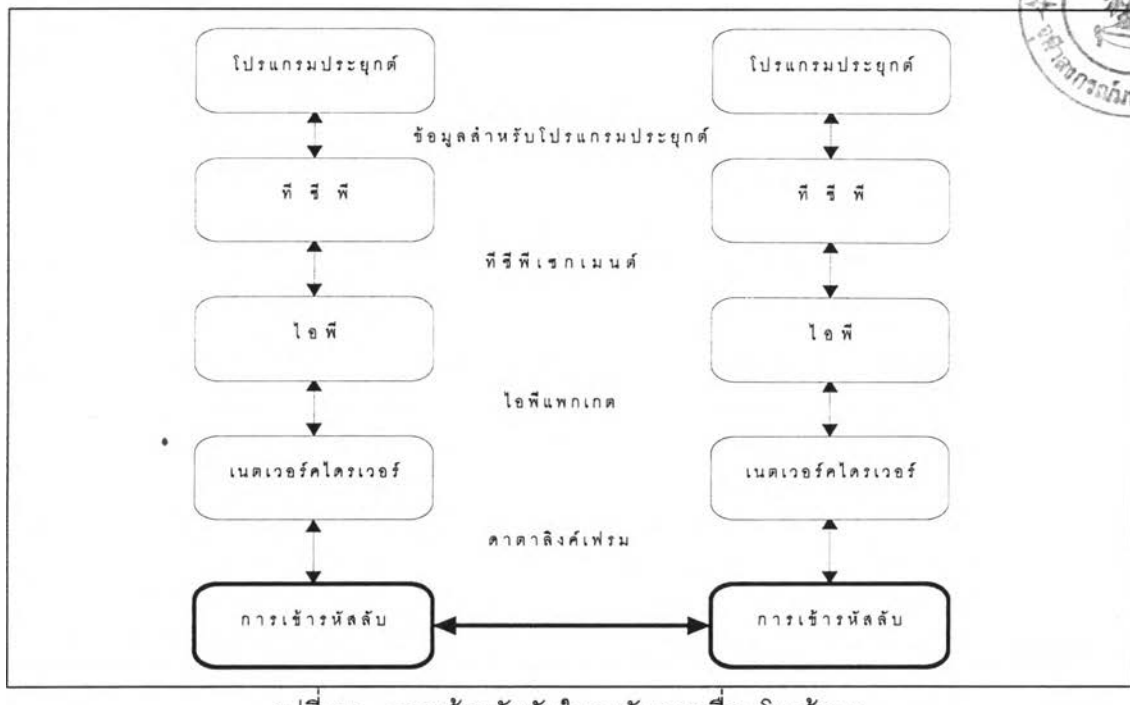
ผู้ที่จะส่งข้อมูลจะเข้ารหัสลับโดยการใช้คีย์สาธารณะ (public key) ของผู้รับไปให้ แล้วผู้รับจะถอดรหัสลับนั้นด้วยการใช้คีย์ลับ (secret key) ของตัวเอง ตัวอย่างของวิธีการเข้ารหัสแบบใช้คีย์สาธารณะได้แก่ RSA, Markle-Hellman Knapsack Algorithm, Diffie-Hellman Algorithm เป็นต้น ด้วยวิธีการเข้ารหัสแบบนี้ ทำให้การเข้ารหัสลับมีความปลอดภัยมากขึ้น เนื่องจากคีย์ที่ใช้สำหรับการถอดรหัสลับ จะเก็บไว้ที่เดียว ไม่ใช่ 2 ฝ่าย โดยวิทยานิพนธ์นี้จะกล่าวถึงเฉพาะ RSA ซึ่งเป็นวิธีการที่ใช้สำหรับการโอนย้ายเซสชั่นคีย์ (session key transferring) เป็นตัวอย่างของวิธีการเข้ารหัสลับแบบใช้คีย์สาธารณะ

2) รูปแบบการเข้ารหัสลับสำหรับการสื่อสารข้อมูลในโปรโตคอลทีซีพี/ไอพี สำหรับรูปแบบการเข้ารหัสลับสำหรับการสื่อสารข้อมูลในโปรโตคอลทีซีพี/ไอพี แบ่งออกเป็น 4 ระดับ (Cheswick, 1994) ได้แก่

- การเข้ารหัสลับระดับการเชื่อมโยงข้อมูล (link level encryption)
- การเข้ารหัสลับระดับทรานสปอร์ต (transport level encryption)
- การเข้ารหัสลับระดับเครือข่าย (network level encryption)
- การเข้ารหัสลับระดับโปรแกรมประยุกต์ (application level encryption)

2.1) การเข้ารหัสลับระดับการเชื่อมโยงข้อมูล (link level encryption) เป็นการเข้ารหัสลับในระดับดาตาลิงค์ โดยทั่วไปมักจะเป็นการใช้อุปกรณ์พิเศษที่เรียกว่า กล่องสำหรับเข้ารหัสลับ (encryption box) ซึ่งวิธีการนี้ถูกใช้ในระบบ Clipper หรือ การใช้ดีไวซ์ไดร์เวอร์

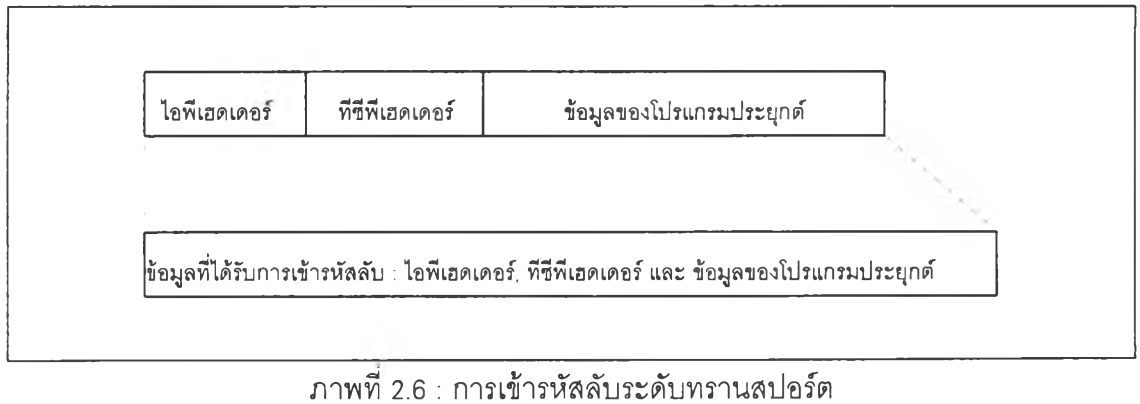
วิธีการนี้เป็นวิธีการที่ปลอดภัยที่สุด เนื่องข้อมูลถูกเข้ารหัสลับทั้งหมด นั่นคือ ทั้งข้อมูลที่ใช้งาน และ ข้อมูลที่เป็นโครงสร้างของโปรโตคอล แต่ด้วยวิธีการนี้เป็นวิธีการที่ค่อนข้างมีค่าใช้จ่ายสูง เนื่องจากอุปกรณ์สำหรับการสื่อสารข้อมูลทั้งหมด จะต้องสนับสนุนการเข้ารหัสลับนี้ด้วย



รูปที่ 2.5 : การเข้ารหัสลับในระดับการเชื่อมโยงข้อมูล

2.2) การเข้ารหัสลับระดับทรานสปอร์ต (transport level encryption) การเข้ารหัสลับระดับทรานสปอร์ต เป็นการเข้ารหัสลับใน ไอพีแพกเกต (IP packet) ลักษณะดังภาพที่ 2.6

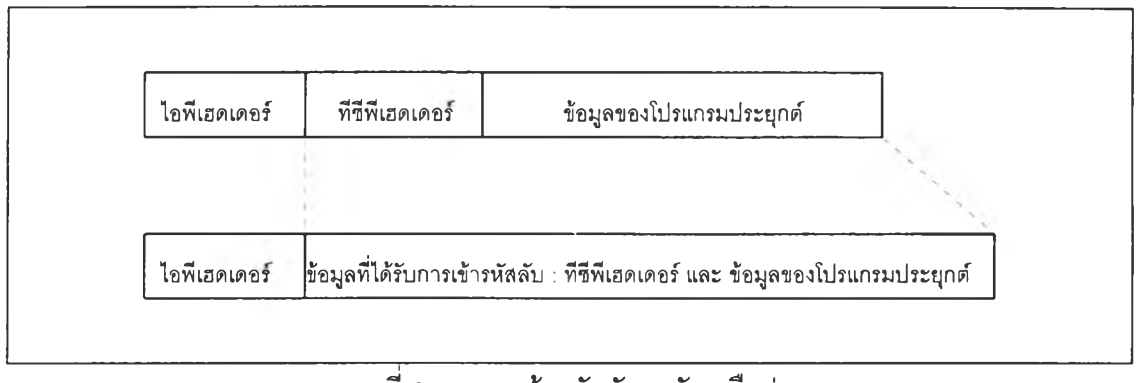
การเข้ารหัสในรูปแบบนี้ถ้าต้องการใช้งานร่วมกันระหว่างเครือข่าย (interconnection) จะต้องอาศัยอุปกรณ์ที่สนับสนุนการเข้ารหัสลับ เช่น ไอพีเราเตอร์ที่สามารถเข้ารหัสลับและถอดรหัสลับ (encryption IP router) เป็นต้น



ภาพที่ 2.6 : การเข้ารหัสลับระดับทรานสปอร์ต

2.3) การเข้ารหัสลับระดับเครือข่าย (network level encryption) การเข้ารหัสลับระดับเครือข่ายเป็นการเข้ารหัสลับ โดยเข้ารหัสในส่วนของทีซีพีเซกเมนต์ ซึ่งได้แก่ ทีซีพีเฮดเดอร์ และ ข้อมูลของโปรแกรมประยุกต์

การเข้ารหัสลับในรูปแบบตามนี้ทำให้การส่งข้อมูลไปตามเครือข่ายสามารถทำได้โดยไม่จำเป็นต้องเปลี่ยนแปลงอุปกรณ์ใด เนื่องจากโครงสร้างในส่วนของไอพีไม่มีการเปลี่ยนแปลง



ภาพที่ 2.7 : การเข้ารหัสลับระดับเครือข่าย

2.4) การเข้ารหัสลับระดับโปรแกรมประยุกต์ (application level encryption) การเข้ารหัสลับระดับโปรแกรมประยุกต์ เป็นการเข้ารหัสลับในลักษณะของ การเข้ารหัสลับแบบจุดต่อจุด (end-to-end encryption) นั่นคือ โปรแกรมประยุกต์จะเข้ารหัสลับในส่วนข้อมูลของตัวเองก่อนส่งผ่านไปยังระดับล่าง ทำให้กระบวนการในระดับล่างไม่จำเป็นต้องมีการเปลี่ยนแปลงอะไร ซึ่งวิธีการนี้ถ้าต้องการนำมาใช้กับโปรแกรมประยุกต์เดิม เช่น เทลเนต หรือ เอฟทีพี จะต้องแก้ไขในส่วนของโปรแกรมต้นฉบับ (source program) ใหม่ทั้ง โปรแกรมขอรับบริการ และ โปรแกรมให้บริการ

3) การแลกเปลี่ยนคีย์ (key exchanging) ในการเข้ารหัสลับสำหรับการสื่อสารข้อมูลระหว่างคู่ใดๆ ในแต่ละครั้งจะต้องมีการใช้ เซสชั่นคีย์ (session key) สำหรับการเข้ารหัสลับ

การแลกเปลี่ยนคีย์มีรูปแบบที่หลากหลาย และมีข้อดีข้อเสีย ต่างกัน สำหรับในวิทยานิพนธ์จะกล่าวถึงรูปแบบการแลกเปลี่ยนคีย์บางรูปแบบ เพื่อเป็นแนวทางในบทที่ 3 ต่อไป

3.1) ศูนย์กลางการกระจายคีย์ (key distribution center) ศูนย์กลางการกระจายคีย์ (key distribution center) เป็นตัวกลางสำหรับเก็บข้อมูลเซสชั่นคีย์ของเครื่องที่จะสื่อสารกันโดยการเข้ารหัสลับ หน้าที่หลักของศูนย์กลางการกระจายคีย์มีดังนี้

- สร้างเซสชั่นคีย์ (generate session key) ให้กับผู้ที่ยังไม่มีเซสชั่นคีย์ หรือต้องการสร้างใหม่
- ทำหน้าที่เก็บรักษาเซสชั่นคีย์ในฐานข้อมูลคีย์ (key database)
- ถ่ายข้อมูลคีย์ (key transferring) สำหรับผู้ที่ต้องการได้ข้อมูลเซสชั่นคีย์

3.2) การแลกเปลี่ยนคีย์แบบการเข้ารหัสลับอย่างสมมาตร การแลกเปลี่ยนคีย์แบบการเข้ารหัสลับอย่างสมมาตร (key exchange with symmetric cryptography) เป็นวิธีการแลกเปลี่ยนคีย์ ที่ใช้วิธีการเข้ารหัสลับแบบคีย์เดียวในการแลกเปลี่ยนคีย์ เพื่ออธิบายรูปแบบการทำงานของวิธีนี้ ขอยกตัวอย่างให้ (ก) ต้องการที่จะส่งเซสชั่นคีย์ ไปให้ (ข) มีขั้นตอนดังนี้

- (ก) ส่งเซสชั่นคีย์ไปให้ (ข) ซึ่งเข้ารหัสลับโดยใช้คีย์ที่ (ก) ส่งไปให้
- เมื่อ (ข) รับข้อมูลมาจาก (ก) จะเข้ารหัสลับด้วยรหัสลับของ (ข) อีกครั้ง แล้วส่งกลับไปให้ (ก)
- เมื่อ (ก) รับข้อมูลที่ (ข) เข้ารหัสลับกลับมา จะถอดรหัสลับโดยใช้คีย์ของตัวเองออกมา แล้วเหลือข้อมูลที่เป็นการเข้ารหัสลับโดยคีย์ของ (ข)ส่งกลับไปให้ (ข)
- เมื่อ (ข) รับข้อมูลกลับมาจะถอดรหัสโดยใช้คีย์ของตัวเอง ได้เป็นเซสชั่นคีย์ที่ต้องการ

3.3) การแลกเปลี่ยนคีย์โดยการใช้วิธีการเข้ารหัสแบบคีย์สาธารณะ การแลกเปลี่ยนคีย์โดยการใช้วิธีการเข้ารหัสลับแบบคีย์สาธารณะ (key exchange with public key encryption) เป็นการแลกเปลี่ยนคีย์โดยการใช้ประโยชน์จากวิธีการเข้ารหัสลับเข้ามาช่วยทำให้ขั้นตอนการแลกเปลี่ยนคีย์ลดความซับซ้อนลงไป

ในการส่งเซสชั่นคีย์จาก (ก) ไปให้ (ข) ตามตัวอย่างเดียวกันกับหัวข้อ 2.2 สามารถแสดงให้เห็นโดยวิธีการนี้ดังนี้

- (ข) ส่งคีย์สาธารณะไปให้ (ก)
- (ก) ส่งเซสชั่นคีย์ที่เข้ารหัสลับโดยใช้คีย์สาธารณะ (ข)
- (ข) ถอดรหัสลับโดยการใช้คีย์ลับของตัวเอง ได้เป็นเซสชั่นคีย์ตามต้องการ

4) RSA

4.1) ความเป็นมา (Pfleeger , 1989) RSA เป็นขั้นตอนวิธีการเข้ารหัสลับแบบการใช้คีย์สาธารณะ คิดค้นโดย Rivest, Shamir และ Adelman ในปี ค.ศ. 1978 รูปแบบของโปรโตคอลเป็นการใช้ทฤษฎีจำนวน (number theory) ทางคณิตศาสตร์ ในเรื่องของความยากในการแยกตัวประกอบของจำนวนเฉพาะ (prime number) เป็นพื้นฐานของวิธีการ

ในการทำงานของ RSA จะใช้วิธีการ modulo ทางคณิตศาสตร์ โดยในการเข้ารหัสลับจะมองค่าของเพลนเท็กซ์ เป็นเหมือนตัวเลขค่าหนึ่ง นำตัวเลขค่านั้นมาทำการคำนวณตามสูตรเพื่อให้เป็นตัวเลขอีกค่าหนึ่ง ซึ่งก็คือ ไซเฟอร์เท็กซ์

ความมั่นคงของ RSA (security of RSA) อยู่ที่การแยกตัวประกอบของตัวเลขขนาดใหญ่ ซึ่งเป็นการแก้ปัญหาแบบ NP Complete

4.2) การเข้ารหัสลับและถอดรหัสลับ การเข้ารหัสลับ จะใช้สมการดังนี้

$$C = P^e \bmod n$$

ส่วนการถอดรหัสลับ ใช้สมการดังนี้

$$P = C^d \bmod n$$

ความหมายของสัญญลักษณ์ต่างๆในสมการคือ

C หมายถึง ไซเฟอร์เท็กซ์

P หมายถึง เพลนเท็กซ์

e,d,n เป็น คีย์ของ RSA โดย *e,n* เป็นคีย์สาธารณะ และ *d,n* เป็นคีย์ลับ

4.3) การเลือกคีย์ของ RSA ความซับซ้อนของ RSA จะอยู่ที่การเลือกคีย์สาธารณะและ คีย์ลับ (*e,d,n*) ในวิทยานิพนธ์นี้จะกล่าวถึงเฉพาะสมการของการเลือกคีย์ ส่วนการพิสูจน์สมการนั้นจะอยู่นอกเหนือขอบเขตการวิจัยในวิทยานิพนธ์นี้

การเลือกคีย์ (*e,d,n*) มีขั้นตอนดังนี้

- คัดเลือก *n* เป็นจำนวนเฉพาะที่มีค่าค่อนข้างมาก

- เลือกตัวเลขปฐม 2 ค่า คือ p และ q โดยที่

$$p \times q = n$$

- เลือกตัวเลขขนาดใหญ่ e โดยที่ e เป็นจำนวนเฉพาะสัมพัทธ์ (relative prime) กับ $(p-1)(q-1)$ นั่นคือ $(p-1)(q-1)$ หารด้วย e ต้องไม่ลงตัว
- คำนวณค่า d ซึ่งเป็นอินเวอร์สกับ e จากสมการ

$$e \times d \bmod((p-1)(q-1)) = 1$$

- ใช้ e,n สำหรับเป็นคีย์สาธารณะ ส่วน d,n เป็นคีย์ลับ

4.4) ตัวอย่างการเข้ารหัสลับของ RSA

เลือก $p = 11$, $q = 13$ ดังนั้น

$n = p \times q = 143$

$(p-1) \times (q-1) = 120$

เลือก e ที่เป็นเลขปฐมสัมพัทธ์กับ $(p-1) \times (q-1)$ สมมติว่าเลือก $e = 11$

หาค่า d ที่เป็นอินเวอร์สกับ e ได้ $d = 11$

สมมติต้องการเข้ารหัสลับค่า $P = 7$

$$C = 7^{11} \bmod 143 = 106$$

ถ้าต้องการถอดรหัสลับ

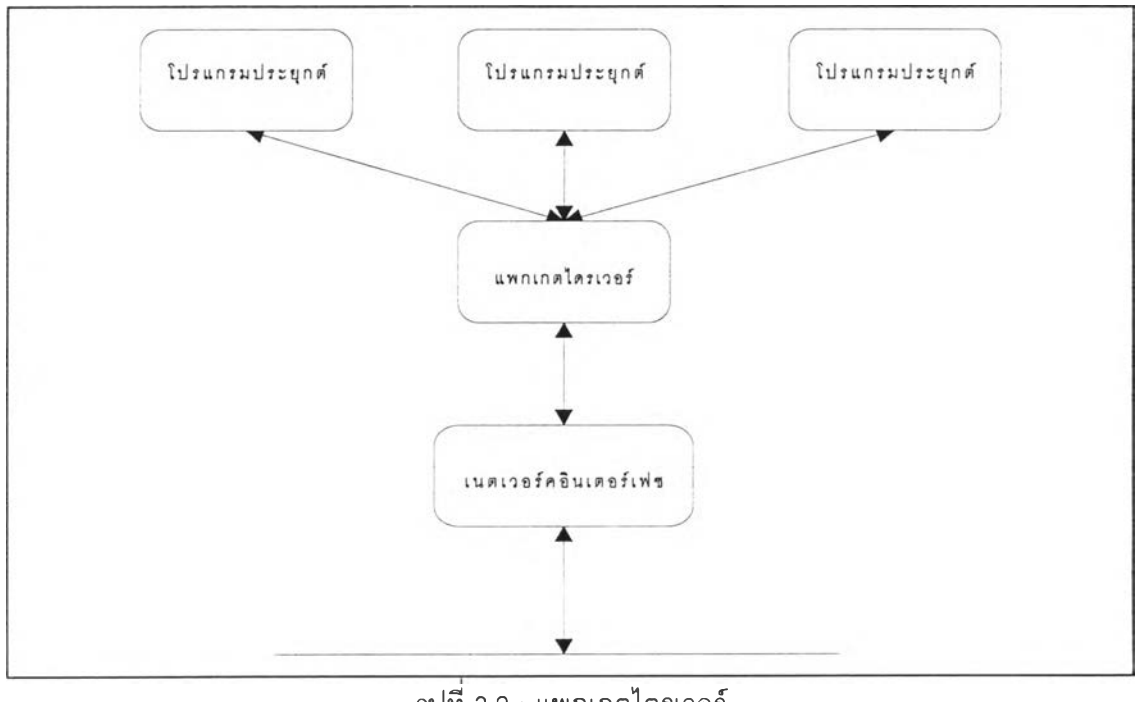
$$P = 106^{11} \bmod 143 = 7$$

แพกเกตไดรเวอร์ (packet driver)

1) ลักษณะของแพกเกตไดรเวอร์ ในการเชื่อมโยงเครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้ระบบปฏิบัติการดอสมีการใช้งานไดรเวอร์สำหรับโปรโตคอลทีซีพี/ไอพี 5 ชนิด (Carl-Mitchell,1993) ได้แก่

- แพกเกตไดรเวอร์
- NDIS (Network Driver Interface Specification)
- ASI (Adapter Support Interface)
- OSI (Open Datalink Interface)
- DLL (The DataLink Layer)

แพกเกตไดรเวอร์ เป็นไดรเวอร์ที่มีการใช้งานมากที่สุดสำหรับการใช้งานทีซีพี/ไอพีบนดอส ลักษณะของแพกเกตไดรเวอร์จะอยู่คั่นกลาง ระหว่างโปรแกรมประยุกต์และเนตเวอร์คอินเตอร์เฟซ ดังภาพ



รูปที่ 2.8 : แพกเกตไดรเวอร์

แพกเกตไดรเวอร์เป็นโปรแกรม TSR (Terminate and Stay Resident) โดยเป็นซอฟต์แวร์อินเตอร์รัพรูทีน (software interrupt service routine) มีช่วงของอินเตอร์รัพแอดเดรส (interrupt address) ระหว่าง 0x60 - 0x80

ในรูทีนของแพกเกตไดรเวอร์จะมี 3 ไบต์แรก เป็นคำสั่ง JMP ต่อจากนั้นจะเป็นตัวอักษร 'PKT DRVR' เพื่อใช้ตรวจสอบว่ามีการโหลดแพกเกตไดรเวอร์ไว้ที่อินเตอร์รัพแอดเดรสนั้นๆ

2) การใช้งานแพกเกตไดรเวอร์ ในการใช้งานแพกเกตไดรเวอร์ ให้เรียกผ่านซอฟต์แวร์อินเตอร์รัพ (INT) ที่แอดเดรสที่แพกเกตไดรเวอร์ถูกโหลดอยู่ โดยรีจิสเตอร์ AH จะบ่งบอกฟังก์ชันที่ต้องการในการเรียกบริการของแพกเกตไดรเวอร์ โดยรายละเอียดทั้งหมดของฟังก์ชันมีรายละเอียดบอกไว้ใน PC/TCP Packet Driver Specification (FTP Software Inc., 1986) สำหรับในวิทยานิพนธ์นี้จะกล่าวถึงเฉพาะฟังก์ชันที่สำคัญและเกี่ยวข้องกับการวิจัยเท่านั้น

3) การรับส่งข้อมูลผ่านทางแพกเกตไดรเวอร์ แพกเกตไดรเวอร์ จะเป็นการทำงานในระดับดาตาลิงค์ ซึ่งโปรแกรมประยุกต์ที่เรียกใช้งานต้องควบคุมการรับส่งข้อมูลให้ตรงกับโปรโตคอลทีซีพี/ไอพีเอง

ฟังก์ชันที่เกี่ยวข้องกับการรับส่งข้อมูลกับแพกเกตไดรเวอร์ได้แก่

- AH = 2 หรือ access_type()
- AH = 3 หรือ send_pkt()
- AH = 4 หรือ release_type()

access_type() AH = 2 เป็นฟังก์ชันสำหรับกำหนด การมัลติเพล็กซ์ในระดับลิงค์ (link layer muliplexing) ค่าพารามิเตอร์ที่ต้องส่งให้กับ access_type() มีดังนี้

AL เป็น ระดับของอินเตอร์เฟซ (Interface Class)

BX เป็น ชนิดของอินเตอร์เฟซ (Interface Type)

DL เป็น เลขของอินเตอร์เฟซ (Interface Number)

DS:SI เป็น ตัวชี้ (pointer) ไปยังชนิดของแพกเกต

CX เป็นความยาวของชนิดของแพกเกตไดรเวอร์

ES:DI เป็น ตัวชี้ไปยังฟังก์ชันที่ทำหน้าที่รับข้อมูล ซึ่งเป็นฟังก์ชันที่อยู่ในโปรแกรมที่เรียกใช้แพกเกตไดรเวอร์ โดยจะถูกเรียกใช้เมื่อข้อมูลที่ได้รับตรงตามข้อกำหนดข้างต้น

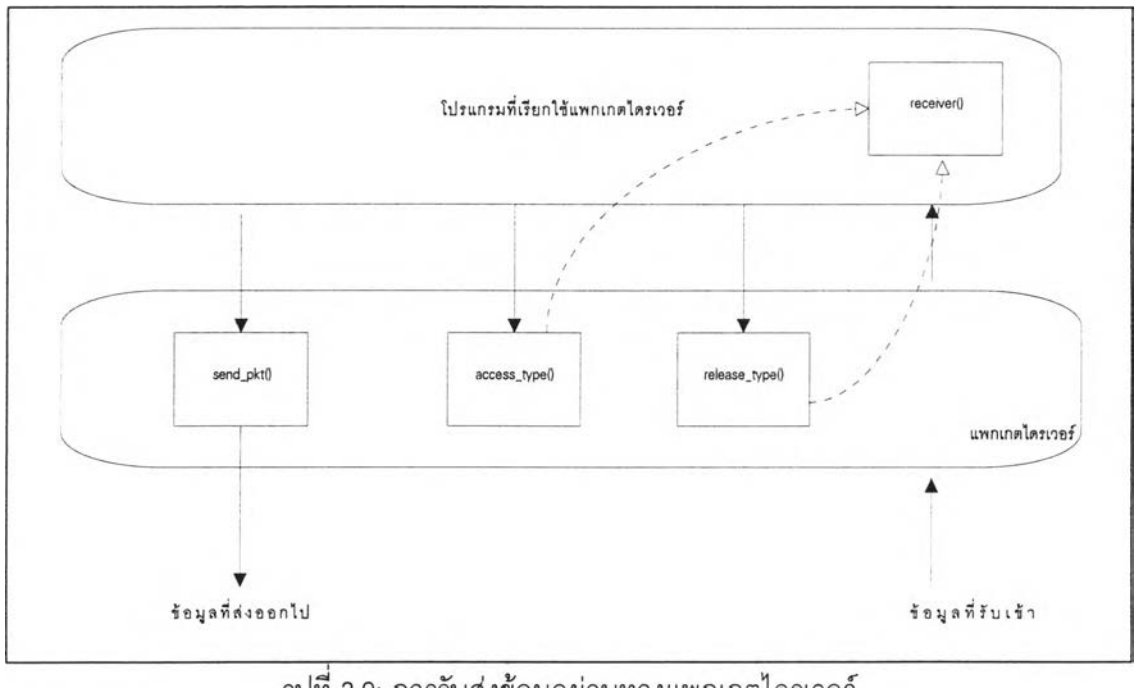
ถ้าการเรียก access_type() นี้ทำงานสำเร็จจะได้รับค่า AX เป็นค่าแฮนเดิล (handle) ซึ่งจะเป็นค่าที่จะถูกอ้างอิงในการรับส่งข้อมูลต่อไป

สำหรับฟังก์ชันในการรับข้อมูล เขียนแทนด้วย receiver() เป็นฟังก์ชันที่จะทำงานเมื่อได้รับข้อมูลโดยการทำงาน 2 ครั้ง ครั้งแรกแพกเกตไดรเวอร์จะส่ง AX = 0 มาให้ ซึ่งเป็นการขอเนื้อที่ในบัฟเฟอร์ ของการรับข้อมูล และจะส่งขนาดของข้อมูลที่ต้องการให้รับที่ CX เมื่อ receiver() มีเนื้อที่ในบัฟเฟอร์และต้องการรับข้อมูลนั้นจะส่งตัวชี้ของบัฟเฟอร์ที่ ES:DI

การทำงานครั้งที่ 2 เป็นการรับข้อมูลจริง โดย receiver() จะได้รับ AX = 1 และ ข้อมูลที่เข้ามาจะถูกชี้โดย DS:SI และ ความยาวถูกกำหนดใน CX

การส่งข้อมูลใช้ฟังก์ชัน send_pkt() AH = 4 ข้อมูลที่จะส่งจะถูกชี้โดย DS:SI และ ความยาวของข้อมูลที่จะส่งอยู่ที่ CX

การยกเลิกการรับข้อมูลที่กำหนดโดย access_type() จะใช้ฟังก์ชัน release_type() AH = 3 โดยระบุแฮนเดิลที่จะยกเลิกในรีจิสเตอร์ BX



รูปที่ 2.9: การรับส่งข้อมูลผ่านทางแพกเกตไดรเวอร์

รูปแบบของโปรเซสบนระบบปฏิบัติการยูนิกซ์

1) รูปแบบการทำงานแบบผู้ขอรับบริการ/ผู้ให้บริการ (client /server computing) ตามที่ได้กล่าวไว้ในหัวข้อของโปรโตคอลทีซีพี/ไอพี ว่าการทำงานของโปรแกรมประยุกต์ในโปรโตคอลทีซี

ฟี/ไอพี จะเป็นลักษณะการทำงานแบบผู้ขอรับบริการและผู้ให้บริการ โดยแบ่งเป็น 3 ส่วน ได้ดังนี้ (Carl-Mitchell, 1993)

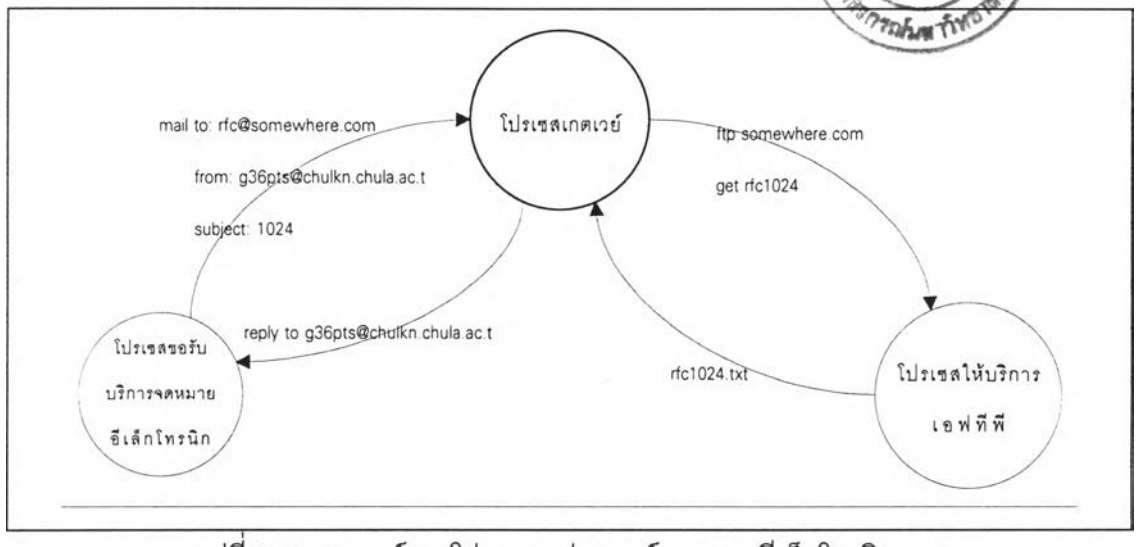
- โปรเซสขอรับบริการ ซึ่งเป็นผู้ใช้บริการ
- โปรเซสให้บริการ
- โปรโตคอลสำหรับโปรเซสขอรับบริการ และ โปรเซสให้บริการ

สำหรับรูปแบบของการทำงานแบบผู้ขอรับบริการ/ผู้ให้บริการ สามารถแบ่งชนิดของโปรเซสให้บริการเป็น 2 รูปแบบ ได้แก่ โปรเซสให้บริการแบบวน (iterative server) และ โปรเซสให้บริการแบบทำงานพร้อมกัน (concurrent server)

โปรเซสให้บริการแบบวน เป็น โปรเซสให้บริการที่มีโปรเซสเดียวให้บริการแก่โปรเซสขอรับบริการ โดยจะให้บริการในแต่ละครั้งสำหรับโปรเซสขอรับบริการเพียงโปรเซสเดียว หลังจากที่ได้ให้บริการแก่โปรเซสขอรับบริการเสร็จเรียบร้อยจึงค่อยวนไปให้บริการแก่โปรเซสขอรับบริการอื่นๆ ต่อไป ลักษณะของโปรเซสให้บริการประเภทนี้มักจะเป็นโปรเซสให้บริการแบบสั้นๆ เช่น โปรเซสให้บริการวันที่ เป็นต้น

สำหรับโปรเซสให้บริการแบบทำงานพร้อมกัน เป็น โปรเซสให้บริการที่จะมีการ fork() โปรเซสให้บริการออกมาทุกครั้งสำหรับโปรเซสขอรับบริการหนึ่งๆ ในลักษณะนี้จะทำให้สามารถจะให้บริการแก่โปรเซสขอรับบริการหลายโปรเซสพร้อมกันในเวลาเดียวกัน

2) ลักษณะของเกตเวย์ของโปรแกรมประยุกต์ (application gateway) เกตเวย์ของโปรแกรมประยุกต์ หรือ โปรเซสเกตเวย์ (gateway process) เป็นโปรเซสที่ทำหน้าที่เป็นโปรเซสคั่นกลางระหว่างโปรเซสขอรับบริการและโปรเซสให้บริการเพื่อทำหน้าที่ในการให้รับส่งข้อมูลของโปรเซสขอรับบริการและโปรเซสให้บริการที่แตกต่างกัน(Comer,1993)



รูปที่ 2.10: เกตเวย์ของโปรแกรมประยุกต์จดหมายอีเล็กโทรนิกและเอฟทีพีสำหรับขออาร์เอฟซี

สำหรับตัวอย่างของเกตเวย์ของโปรแกรมประยุกต์ ได้แก่ โปรเซสเกตเวย์สำหรับการขออาร์เอฟซี(rfc) โดยการใช้จดหมายอีเล็กโทรนิก ลักษณะเป็นดังนี้ สมมติว่าผู้ใช้ใช้งานอยู่บนเครื่องที่ไม่มีโปรแกรมขอรับบริการเอฟทีพี แต่ มีโปรแกรมสำหรับส่งจดหมายอีเล็กโทรนิก ถ้ามีโปรเซสทำหน้าที่เกตเวย์ระหว่างจดหมายอิเล็กโทรนิกและเอฟทีพี ผู้ใช้งานเครื่องที่ไม่สามารถใช้งานโปรแกรมขอรับบริการเอฟทีพี แต่สามารถใช้จดหมายอีเล็กโทรนิกได้ ผู้ใช้นั้นสามารถส่งจดหมายไปยังโปรเซสเกตเวย์เพื่อให้โปรเซสเกตเวย์ไปดึงข้อมูลอาร์เอฟซีมาให้ แล้วส่งกลับมายังผู้ใช้ในรูปจดหมายเช่นเดิม ดังภาพที่ 2.10 เป็นตัวอย่างของ g36pts ส่งจดหมายอีเล็กโทรนิกไปขอ rfc1024 ทางจดหมายอีเล็กโทรนิก โดย rfc@somewhere.com เป็นที่อยู่ที่จะส่งไปให้โปรเซสเกตเวย์ หลังจากนั้นโปรเซสเกตเวย์จะไปดึงข้อมูลในลักษณะเอฟทีพี เพื่อขออาร์เอฟซีตามที่ใน subject: ระบุไว้ในตัวอย่างเป็นการขอ rfc1024